RaqForum
c.raqsoft.com

Desktop and Excel Data Processing Cases

Scudata Inc. | Raqsoft Inc.

Table of contents

Preface

With the advantages of easy-to-use, rich data processing functions, strong chart drawing capability, fast and accurate calculation, etc., Excel has become an indispensable utility software for data processing in daily office work, and very popular among office workers.

Nevertheless, there are still quite a few complex tasks that are difficult to handle with Excel, such as the processing of ordered set, the conversion of data structure, the alternate performing of grouping, filtering and aggregating, the association and comparison between tables, and the merging and splitting of multiple files. Consequently, these complex tasks often make the office workers so head-scratching that they have to work overtime to cope. Although Excel comes with the VBA language, it provides most employees with little substantial help for the reason that VBA language is insufficient in set orientation, relatively high difficulty to learn and high complexity in use.

SPL, as an open-source programming language, provides Excel with add-in and clipboard, and enhances the functions like programming, which has the following characteristics: i) SPL is easy to learn, seamlessly integrated with Excel, and can be used as one function of Excel; ii) SPL provides a wealth of ordered operations, allowing you to easily achieve the positioning and search; iii) SPL provides rich structured data processing functions, which allow you to perform not only various complex calculations but many conversions on data structure; iv) SPL naturally supports the grouped intermediate results, and is able to filter and aggregate the grouped results, thereby allowing you no longer worry about the complex situation that grouping, filtering and aggregating occur alternatively; v) Association and comparison between tables are the forte of SPL, which can be implemented easily, and done efficiently. In terms of other minor functions, SPL has also made a lot of supplements to Excel, such as the data generation with special requirements.

With the help of SPL, the office workers can easily solve the deficiencies of Excel in the functions mentioned above, thereby greatly improving their daily work efficiency.

To learn SPL, please visit: http://c.raqsoft.com/article/1634722432114, where you will find it easy to learn as long as you have a high school education background.

This book collects a large number of Excel problems that may be encountered in your daily work from the Internet, and the number of such problems is up to several hundreds, with a wide coverage. All you have to do is to search for the chapters that interest you and find the appropriate cases and formulas, and you will be able to apply them to your own work after minor modifications.

In short, SPL can help you avoid working overtime on Excel data processing, and make your dream of finishing work in advance come true!

Chapter 1 **Reading and writing files and common computing**

1.1 **Text file**

1.1.1 **Structured text file**

The format of structured text is relatively regular, that is, there is one piece of data in each line, and the columns are separated by separators. SPL can use the function import/export to read and write the structured text.

For example: the file ordersNT.txt stores the order information, and the columns are separated by the tab. The business meanings of every column sequentially are: order ID, customer NO, sales ID, order amount and order date. Part of the data is as follows:

```
26 TAS 1 2142.4 2009-08-05
33 DSGC 1 613.2 2009-08-14
84 GC 1 88.5 2009-10-16
133 HU 1 1419.8 2010-12-12
32 JFS 3 468.0 2009-08-13
39 NR 3 3016.0 2010-08-21
43 KT 3 2169.0 2009-08-27
…
```

The following is the partial result after the table is processed in a way that first sort the orders in ascending order alphabetically by customer NO, and then sort the orders with same customer NO in descending order by order amount, and finally save it to a new file in the original format:

```
136 ARO 25 899.0 2009-12-16
16 BDR 27 2464.8 2009-07-23
81 BDR 29 1168.0 2010-10-14
108 BDR 12 480.0 2010-11-15
139 BDR 30 166.0 2010-12-18
93 BON 6 2564.4 2010-10-29
106 BSF 27 10741.6 2009-11-13
…
```

SPL script:

	A
1	=file("D:/data/ordersNT.txt").import()
2	=A1.sort(_2,-_4)
3	=file("D:/data/ordersNT_sort.txt").export(A2)

A1, A3: Read in and write out structured text file.

A2: The function sort is to sort; _2 and _4 represent the 2nd and 4th columns respectively; By default, they are sorted in ascending direction, and the negative sign represents the descending direction.

SPL can also process the text file with **column name (title)**. For example, the first line of orders.txt is the column name, and part of the data are as follows:

OrderID Client SellerId Amount OrderDate
26 TAS 1 2142.4 2009-08-05
33 DSGC 1 613.2 2009-08-14
84 GC 1 88.5 2009-10-16
133 HU 1 1419.8 2010-12-12
32 JFS 3 468.0 2009-08-13
39 NR 3 3016.0 2010-08-21
43 KT 3 2169.0 2009-08-27
…

Likewise, sort this file and write the result to a new file together with column name:

	A
1	=file("D:/data/orders.txt").import@t()
2	=A1.sort(Client,-Amount)
3	=file("D:/data/orders_sort.txt").export@t(A2)

A1, A3: The option @t means the text file is read and written together with column name.

A2: Sort by column names rather than sequence number.

The default **separator** of the function import/export is tab, and the option @c means that the comma is used as the separator (usually used in csv files). If other special separators are encountered, SPL can also handle.

For example, the orders_semi.txt uses || as the separator:

	A
1	=file("D:/data/orders_semi.txt").import@t(;,"||")
2	=A1.select(Amount>=1000 && Amount<2010)
3	=file("D:/data/orders_semi_select.txt").export@t(A2;"||")

The default function of the function export is to write the data to a new file, or to overwrite the file with the same name, but sometimes we need to **append** new data with the same structure to the original file, in this case, we can use the option @a:

=file("D:/data/orders_semi_select.txt").export@at(A2;"||")

1.1.2 String sequence

The format of some text files is not regular, and it cannot directly perform the structured calculation. Such files, however, can be read as a sequence of strings. The formats of such semi-structured data are numerous, let's take multi-line data as an example to illustrate the general method of reading and writing the string sequence in SPL.

The first 2 lines of every 3 lines in file 3lines.txt correspond to one piece of data, and the 3rd line is useless. Part of the data is as follows:

```
26 TAS 1 2142.4
2009-08-05
some comment
33 DSGC 1 613.2
2009-08-14
some comment
27 TAS 1 2142.4
2009-08-05
some comment
```

Remove the useless line from the file and write the result to a new file:

	A
1	=file("D:/data/3lines.txt").read@n()
2	=A1.step(3,1,2)
3	=file("D:/3lines_reuslt.txt").write(A5)

A1: Read the text file. @n means reading as a sequence by line, and each member of the sequence corresponds to a line.

A2: Take the first member and the second member for every three members of sequence A1.

A3: Write the sequence to a text file, and each member of the sequence corresponds to a line.

1.2 Excel file

1.2.1 Structured tables

The structured Excel table is relatively regular, and SPL reads and writes it with the xlsimport/xlsexport function.

For example: the business meanings of every column of the first sheet in ordersNT.xlsx sequentially are: order ID, customer NO, sales ID, order amount and order date. Part of the data is as follows:

	A	B	C	D	E
1	26	TAS	1	2142.4	2009/8/5
2	33	DSGC	1	613.2	2009/8/14
3	84	GC	1	88.5	2009/10/16
4	133	HU	1	1419.8	2010/12/12
5	32	JFS	3	468	2009/8/13
6	39	NR	3	3016	2010/8/21
7	43	KT	3	2169	2009/8/27

The following is the partial result after the table is processed in a way that first sort the orders in ascending order alphabetically by customer NO, and then sort the orders with same customer NO in descending order by order amount, and finally save it to a new Excel in the original format:

SPL script:

	A
1	=file("D:/data/ordersNT.xlsx").xlsimport()
2	=A1.sort(_2,-_4)
3	=file("D:/data/ordersNT_sort.xlsx").xlsexport(A2)

A1, A3: read and write the first sheet of Excel. If you want to read the specified sheet, you can use:

xlsimport(;Sheet number or Sheet name)

If you want to write to the specified sheet, you can use:

xlsexport(A2; Sheet number or Sheet name)

SPL can also process the structured table with **column name (title)** in a similar way to process text files. For example, partial data of orders.xlsx is as follows:

	A	B	C	D	E
1	OrderID	Client	SellerId	Amount	OrderDate
2	26	TAS	1	2142.4	2009/8/5
3	33	DSGC	1	613.2	2009/8/14
4	84	GC	1	88.5	2009/10/16
5	133	HU	1	1419.8	2010/12/12
6	32	JFS	3	468	2009/8/13
7	39	NR	3	3016	2010/8/21
8	43	KT	3	2169	2009/8/27

Sort this file and write the result to a new file together with column name:

	A
1	=file("D:/data/orders.xlsx").xlsimport@t()
2	=A1.sort(Client,-Amount)
3	=file("D:/data/orders_sort.xlsx").xlsexport@t(A2)

Sometimes the first few rows of the table are useless and need to be **skipped,** for example:

	A	B	C	D	E
1	orders from 2009 to 2010				
2					
3					
4	OrderID	Client	SellerId	Amount	OrderDate
5	26	TAS	1	2142.4	2009/8/5
6	33	DSGC	1	613.2	2009/8/14
7	84	GC	1	88.5	2009/10/16
8	133	HU	1	1419.8	2010/12/12
9	32	JFS	3	468	2009/8/13

Reading from row 4:

```
=file("D:/data/ orders.xlsx").xlsimport@t(;,4)
```

Sometimes we need to **append** new data with the same structure to the original table. In this case, we can use the option @a:

```
=file(""D:/data/orders_sort.xlsx").xlsexport@at(A2)
```

If the appearance attribute is set for the last non-blank row of the original table, the appended data will inherit the style of this row. For example, the display format of column D of original table is #,##0.00, and the style of column E is mmm-dd-yyyy, as shown in the table below:

	A	B	C	D	E
1	**OrderID**	**Client**	**SellerId**	**Amount**	**OrderDate**
2	26	TAS	1	2,142.40	Aug-05-2009
3	33	DSGC	1	613.20	Aug-14-2009
4	84	GC	1	88.50	Oct-16-2009

After appending the data, the result is as follows:

	A	B	C	D	E
1	**OrderID**	**Client**	**SellerId**	**Amount**	**OrderDate**
2	26	TAS	1	2,142.40	Aug-05-2009
3	33	DSGC	1	613.20	Aug-14-2009
4	84	GC	1	88.50	Oct-16-2009
5	133	HU	1	1,419.80	Dec-12-2010
6	32	JFS	3	468.00	Aug-13-2009

The style attribute has been set for the first blank row after the last non-blank row of original table, the appended data will inherit the style attribute of this row. Using this feature, we can achieve the data output from scratch in the specified format. For example, first create a blank Excel, and set the display format of column D in row 2 to #,##0.00, and that of column E to mmm-dd-yyyy.

	A	B	C	D	E
1	**OrderID**	**Client**	**SellerId**	**Amount**	**OrderDate**
2					

Then append data to the blank table, and the result is as follows:

	A	B	C	D	E
1	**OrderID**	**Client**	**SellerId**	**Amount**	**OrderDate**
2	133	HU	1	1,419.80	Dec-12-2010
3	32	JFS	3	468.00	Aug-13-2009

1.2.2 Two-dimensional string sequence

Some Excel tables are not regular in format and do not have clear column attributes. In this case, they can be processed as a two-dimensional string sequence.

For example, the following key-value data:

	A	B	C	D	E
1			A=123	B=456	C=789
2	A=678	B=783	A=900	U=89	
3	A=330	Y=67	B=890	C=311	F=19

Now, we want to split the above data into 2 columns by key and value, and sort them by key and value, and finally write them to a new Excel. The result is as follows:

	A	B
1	A	123
2	A	330
3	A	678
4	A	900
5	B	456
6	B	783
7	B	890
8	C	311
9	C	789
10	F	19
11	U	89
12	Y	67

SPL script:

	A
1	=file("D:/data/keyvalue.xlsx").xlsimport@w()
2	=A1.conj().select(~)
3	=A2.(~.split("="))
4	=A3.sort(~(1),~(2))
5	=file("D:/data/keyvalue_result.xlsx").xlsexport@w(A4)

A1: read in the Excel, @w means reading as a two-dimensional string sequence. The whole is a large sequence, and each row is not only a member of the large sequence, but also a small sequence; each cell in the row is a member of the small sequence.

A2: Concatenate the 2D sequence into 1D sequence and remove possible blank cells such as A1, B1, E2.

A3: Split the string sequence into key and value.

A4: Sort by key and value.

A5: Write the result to a new Excel, @w means writing a sequence of sequences.

1.2.3 **Cells**

Sometimes we need to read and write the Excel cells accurately.

For example: the table below has the editor and edit date in row 1.

	A	B	C	D	E
1	editor:emily				date:Dec-30-2011
2	**OrderID**	**Client**	**SellerId**	**Amount**	**OrderDate**
3	26	TAS	1	2,142.40	Aug-05-2009
4	33	DSGC	1	613.20	Aug-14-2009
5	84	GC	1	88.50	Oct-16-2009
6					
7					

Now we want to copy the editor and edit date to the corresponding position on row 7, the result is as follows:

	A	B	C	D	E
1	editor:emily				date:Dec-30-2011
2	OrderID	Client	SellerId	Amount	OrderDate
3	26	TAS	1	2,142.40	Aug-05-2009
4	33	DSGC	1	613.20	Aug-14-2009
5	84	GC	1	88.50	Oct-16-2009
6					
7	editor:emily				date:Dec-30-2011

SPL script:

	A	B
1	=file("D:/data/cell.xlsx")	
2	=A1.xlsopen()	
3	=str=A2.xlscell("A1")	=A2.xlscell("A7";str)
4	=str=A2.xlscell("E1")	=A2.xlscell("E7";str)
5	=A1.xlswrite(A2)	

A2: Open the Excel file as an object.

A3: Read the cell A1 and assign the variable str. By default, it is to read from the first sheet. If you want to read the cell A1 in the specified sheet, you can use:

A2.xlscell("A1",Sheet number or Sheet name)

B3: Write contents of cell A1 to cell A7. Similarly, if you want to write to cell A7 of the specified sheet, you can use:

A2.xlscell("A7",Sheet number or Sheet name;str)

A4-B4: Read the contents of cell E1, and write to cell E7.

A5: Write Excel object to Excel file.

In the above example, A1-E1 that need to be read are consecutive cells, and A7-E7 that need to be written are also consecutive cells. For the reading and writing of such consecutive cells, SPL can implement with a more simplified code:

	A
1	=file("cell.xlsx")
2	=A1.xlsopen()
3	=arry=A2.xlscell@w("A1":"E1")
4	=A2.xlscell("A7":"E7";arry)
5	=A1.xlswrite(A2)

A3: Read consecutive cells in sequence format

A4: Write the sequence to consecutive cells, and each member of the sequence corresponds to one cell. You can either use a sequence to write data to consecutive cells or use a string separated by TAB (\t) or Enter (\r), where TAB means horizontal (column) separation and Enter means vertical (row) separation.

1.2.4 **Multi-sheet processing**

Using Excel objects, not only can read and write cells but also process multiple sheets.

A certain Excel uses multiple sheets to store the order tables. Each sheet has the same format, but the number and name are not the same. Now we want to sort these orders into a new Excel to make each sheet store one year's data.

SPL script:

	A
1	=file("orders_sheet.xlsx").xlsopen()
2	=A1.(stname).(A1.xlsimport@t(;~)).conj()
3	=A2.group(string(year(OrderDate)):name;~:content)
4	=file("orders_result.xlsx").xlsopen@w()
5	=A3.(A4.xlsexport@t(~.content;string(~.name)))
6	=A4.xlsclose()

A1: Open the source Excel file as an object.

A2: Traverse each sheet, read the order of each sheet, and concatenate all the orders. A1.(stname) means taking out all sheet names from the Excel object A1.

A3: Group the orders by year.

A4: Open the target Excel file as an object. @w means write mode, and a new file will be created if the file does not exist.

A5: Traverse each group (yearly) orders of A3 and write them to the new sheet of A4 in turn.

A6: The Excel object opened in @w mode must be closed with the function xlsclose.

1.3 Files and directories

1.3.1 Parse file name

The function filename can parse out different parts of the file name:

	A
1	=filename("D://file/test.splx")
2	=filename@e("D://file/test.splx")
3	=filename@n("D://file/test.splx")
4	=filename@d("D://file/test.splx")

A1: File name with extension: test.splx

A2: Extension: splx

A3: File name without extension: test

A4: Path: D://file

Having known every part of the filename, you can use the concat function to piece together the full path, for example:

	A
1	=concat("D://file/","test",".splx")

A1: Full path: D://file/test.spx

1.3.2 Traverse the files

There are many Excel files under a certain directory, and the first Sheet of these files stores the order data and has the same structure. Now we want to concatenate these orders into a new Excel.

SPL script:

	A
1	=direcotory@p("d:/data/*.xlsx")
2	=A1.conj(file(~).xlsimport@t())
3	=file("d:/result.xlsx").xlsexport@t(A2)

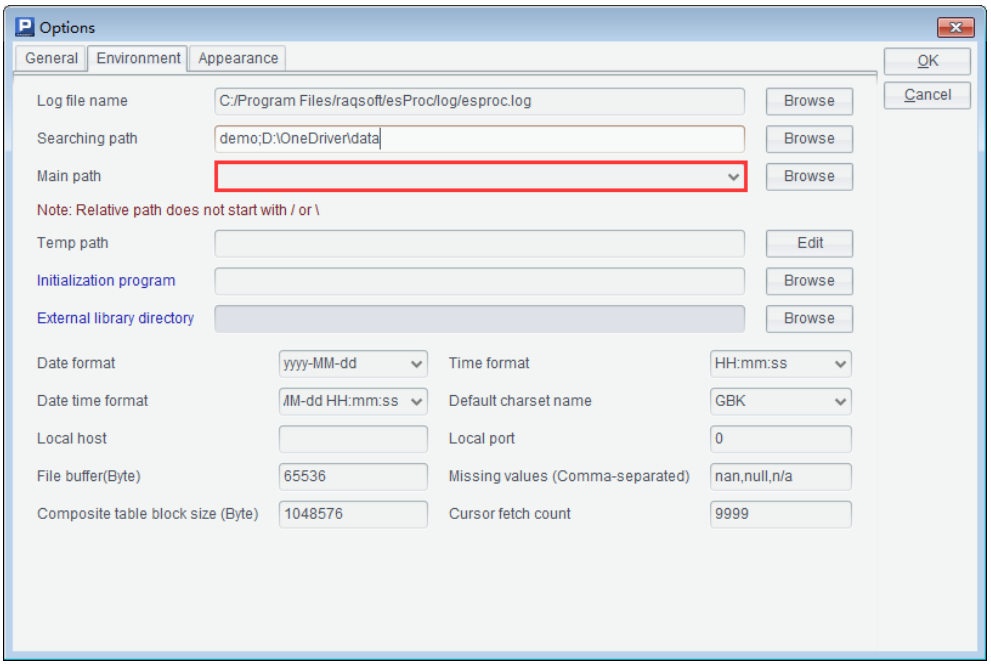
A1: Search for all file names whose extension is xlsx in the directory. @p means returning the full path.

A2: Loop through the file names to read Excel, and then concatenate the data.

The function directory has more functions, such as using the @s option to search for subdirectories recursively, using the @d option to list subdirectories, using @r to delete directories, and using @m to create directories.

1.3.3 System directory

In the previous example, we use the full path to access the data file. If the main path in the system directory is configured, it can be used as the root directory. In this case, we can use the relative path to access the data file. See the figure below for the specific configuration interface:



For example, when the main path is not configured, the script is:

```
file("d:/data/p/orders.xlsx").xlsimport()
```

When configuring main path=d:\data, the script can be written as:

```
file("p/orders.xlsx").xlsimport()
```

If the main path is not configured, and a direct relative path is used, the actual main path is the directory where esProc is started. When esProc is started directly (or via shortcut), the directory is [esProc installation directory\bin]. When double-clicking the .splx file to start esProc indirectly, the directory is the directory where the script file is located. Using the following script can obtain the current main path:

```
filename@p("")
```

In addition to the main path, esProc has other important system directories.

temp: It's the directory where the calculation engine stores temporary files. If it is not set, the operating system temporary directory will be used by default.

searching path: It's the root directory of the script file, including the main path. Multiple directories can be set, and the directories are separated by semicolon.

1.4 General data table operations

1.4.1 T function and E function

When using the import()/xlsimport function, it needs to define the file object first. Since the operation for reading and writing of the structured files is very common, SPL provides a simpler T() function, which can automatically take different actions based on the file extension.

	A
1	=T("data.txt")
2	=T("data.csv")
3	=T("data.txt";”|”)
4	=T@b("data.csv")
5	=T("data.xls")
6	=T("data.xlsx")
7	=T("data.xlsx";"sheet2")
8	=T@b("data.xlsx")

A1: With title, columns to be separated by TAB

A2: With title, columns to be separated by comma

A3: With title, columns to be separated by |

A4: Without title, separated by comma

A5: With title

A6: With title

A7: With title, specify sheet

A8: Without title

T() function has also other parameters, which allow you to choose to read partial columns, and support writing. Because it is not very common, no examples are given here.

The data in Excel table often appears in the form of two-dimensional sequence, and it will be more convenient to convert the sequence to a table sequence when processing. SPL provides a short E function to process the said conversion:

	A
1	=file("data.xlsx").xlsimport@w()
2	=E(A1)
3	=E@b(A1)
4	=E(A2)
5	=E@b(A2)

A1: Read the Excel as a 2D sequence

A2: Convert two-dimensional sequence to table sequence, the first row is the title

A3: Without title

A4: Convert the table sequence to two-dimensional sequence, the first row is the title

A5: Ignore the title

E() function has other options, which allow you to convert the table sequence and TAB/Enter separated strings to each other. You can refer the documents to do experiments by yourself.

1.4.2 **Filtering**

Filter out the row that meets the condition from the data table.

Example: We want to filter out the student scores of Class 10 from the student score table Students_scores.txt. The first row in the file is the column name, and the data starts from the second row, as shown in the figure below.

CLASS	NAME	English	Chinese	Math
1	Adams Brooke	63	31	69
1	Adams Hannah	89	85	79
1	Adams Jonathan	88	87	91
1	Allen Ashley	98	97	97

	A
1	=T(“E:/txt/Students_scores.txt”).select(CLASS==10)

A1: Read the data in the file and then select the rows of class 10. The T function will automatically select the appropriate separator based on the file extension.

1.4.3 **Summary**

Summarize the data in the data table.

Example: Calculate the average score in Chinese, the highest score in math, and the total score in English for all students in the student score table.

	A
1	=T("E:/txt/Students_scores.txt")
2	=A1.avg(Chinese)
3	=A1.max(Math)
4	=A1.sum(English)

A1: read the data in the file.

A2: calculate the average score in Chinese

A3: calculate the highest score in Math

A4: calculate the total score in English

1.4.4 Cross-column calculation

Perform cross-column calculation on the data in the data table.

Example: calculate the total score of each student in the student score table.

	A
1	=T("E:/txt/Students_scores.txt")
2	=A1.derive(English+Chinese+Math:total_score)

A1: read the data in the file.

A2: add a column total_score in A1, and the value of this new column is the sum of English, Chinese and Math columns

The results in A2 are as follows:

Index	CLASS	NAME	English	Chinese	Math	total_score
1	1	Adams Bro...	63	31	69	163
2	1	Adams Han...	89	85	79	253
3	1	Adams Jon...	88	87	91	266
4	1	Allen Ashley	98	97	97	292
5	1	Allen Brand...	93	76	78	247

1.4.5 Sorting

Sort the data in ascending/descending order.

Example: sort the student score table in ascending order by class number, and in descending order by total score.

	A
1	=T("E:/txt/Students_scores.txt")
2	=A1.sort(CLASS)
3	=A1.sort(CLASS,-Math)

A1: read the data in the file.

A2: sort in ascending order by class number

A3: first sort in ascending order by class number, and then sort in descending order by math score within the class

1.4.6 Grouping and aggregating

Group and aggregate the data in the data table.

Example: query the lowest score in English, the highest score in Chinese and the total score in Math for each Class.

	A
1	=T("E:/txt/Students_scores.txt")
2	=A1.groups(CLASS;min(English),max(Chinese),sum(Math))

A1: read the data in the file.

A2: group by class, and calculate the lowest score in English, highest score in Chinese, and total score in math for each class.

1.4.7 Filter after grouping

Filter the data after they are grouped and aggregated.

Example: find the classes with an average English score below 70.

	A
1	=T("E:/txt/Students_scores.txt")
2	=A1.groups(CLASS;avg(English):avg_En)
3	=A2.select(avg_En<70)

A1: read the data in the file.

A2: group by class, and calculate the average English score of each class and name the new column as avg_En

A3: select those with an average English score of below 70 from A2

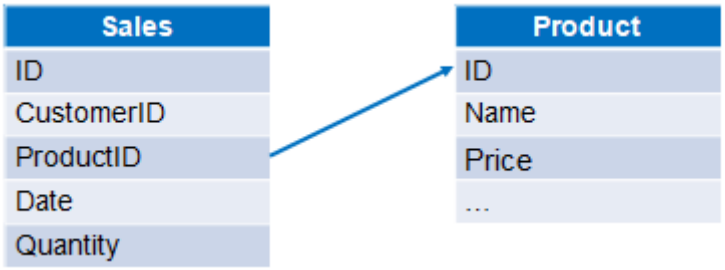
The query results in A3 are as follows:

Index	CLASS	avg_En
1	4	69.6046511627907
2	7	69.86

1.4.8 **Association**

- Perform the associative calculation on the data in two data tables.

Example: The sales order information and product information are stored in two Excel files, respectively, and now we want to calculate the sales of each order. The data structure of the two files is as follows:



	A
1	=T(“e:/orders/sales.xlsx”)
2	=T(“e:/orders/product.xlsx”).keys(ID)
3	=A1.join(ProductID,A2,Name,Price)
4	=A3.derive(Quantity*Price:amount)

A1: read the sales order data.

A2: read the product information data, and set ID as the primary key

A3: associate A1 with the primary key in A2 according to the ProductID, and join the data of Name and Price columns at the same time

A4: add a column amount in A3, and its value is product of the sales Quantity and product Price

- Perform the associative query on the data in two data tables.

Example: We still use the above-mentioned two files to query the sales orders with product price greater than 20 dollars.

	A
1	=T(“e:/orders/sales.xlsx”)
2	=T(“e:/orders/product.xlsx”).select(Price>20).keys(ID)
3	=A1.switch@i(ProductID,A2)

A1: read the sales order data.

A2: read the product information data to select the product information with a price greater than 20, and then set the ID as the primary key.

A3: associate A1 with the primary key in A2 according to ProductID, the option @i means that when a product ID that matches the ProductID cannot be found in A2, this row will be deleted.

- Perform the associative query on the data in primary table and detail table.

Example: Part of data in the employee information table (employee.xlsx) and employee family member table (family.xlsx) are as follows. Now we want to query the information of employees whose family has the elderly person over 70 years old.

	A	B	C	D	E	F	G
1	Eid	IDCard	Name	Sex	Birthday	Phone	Depart
2	1	510121198206253000	San Zhang	M	1982-6-25	13612345678	Research
3	2	110114198907286000	Si Li	M	1989-7-28	13818022624	Sales
4	3	310503198803243000	Xiaoyu Zhao	F	1988-3-24	13852416325	HR
5	4	211506199202133000	Wu Wang	M	1992-2-13	13785265498	Research

	A	B	C	D	E
1	Eid	Name	Relation	Birthday	Phone
2	1	Zhou Zhang	farther	1945-6-4	15313231568
3	1	Yin Hu	wife	1990-8-20	13718826593
4	1	Wuji Zhang	son	2016-3-16	
5	2	Dasuan Li	farther	1961-7-13	13625689532
6	2	Haixia Liu	mother	1964-9-24	13924689512
7	3	Darong Luo	husband	1986-12-15	13598325647
8	3	Xiaolu Luo	daughter	2014-10-9	

	A
1	=T("e:/work/employee.xlsx")
2	=T("e:/work/family.xlsx").select(age(Birthday)>=70)
3	=join(A1:employee,Eid;A2:family,Eid)
4	=A3.conj(employee)

A1: read the employee information data

A2: read the employee family member data and select members over the age of 70

A3: associate A1 and A2 according to Eid, and filter to delete unmatched row, and name A1 as employee and A2 as family

A4: take out the employee column in A3 and concatenate as a table sequence

Chapter 2 Use Excel Add-in

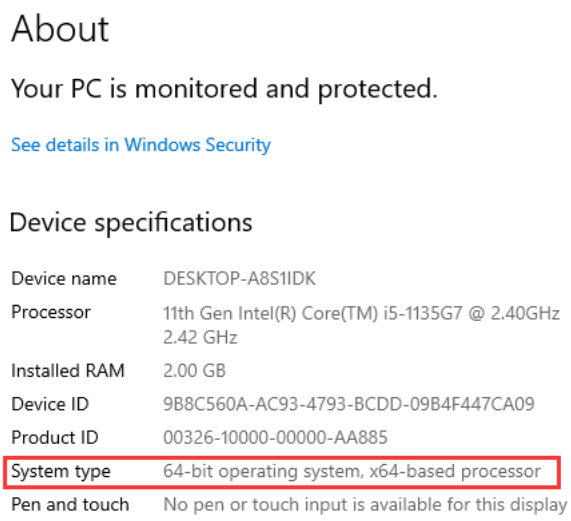
2.1 Installation and configuration

2.1.1 Environment

The **64-bit** (not 32-bit) version of both Windows and Excel is required, otherwise an exception will occur.

To check whether it is 64-bit:

1. For Windows: Click: Settings > System > About, to find the information.



2. For Excel: Open Excel and click: File > Account > About Excel, a pop-up dialog box will display the current Excel version, as shown below:

About Microsoft® Excel® 2019

Microsoft® Excel® 2019MSO (Version 2202 Build 16.0.14931.20116) 64Bit

Product ID: 00405-32554-86889-AAOEM

Session ID: 5DCA85DD-F5BE-4570-8CE4-4D56EF93273D

About language: Make sure the esProc version you are installing has the same language as your operating system. That is, install an English-version esProc if you are using an English-version OS, and a Chinese-version esProc if your OS is Chinese version, otherwise problems like messy code will occur.

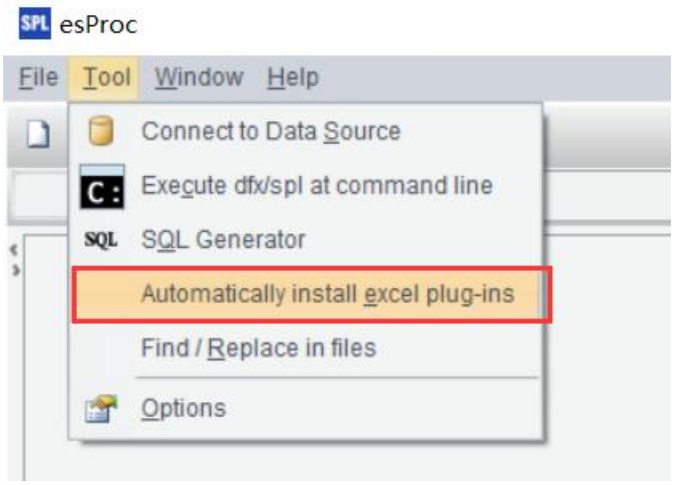
If messy code or other problems occur after a Chinese-version esProc is installed under a Chinese-version OS, check OS configurations in:

Settings > Time & language > Language > Administrative language settings > Change system locale

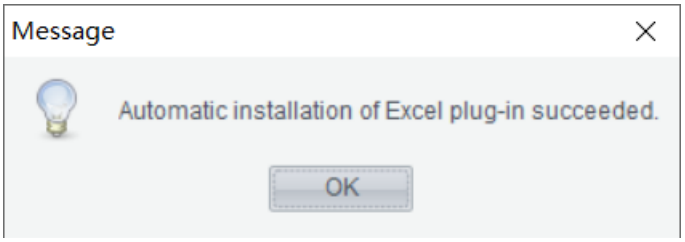
Check whether the current selected language is Chinese, if not, change it to **Chinese**.

2.1.2 **Installation**

Start the esProc **as administrator**, click Tool > Automatically install excel plug-ins:

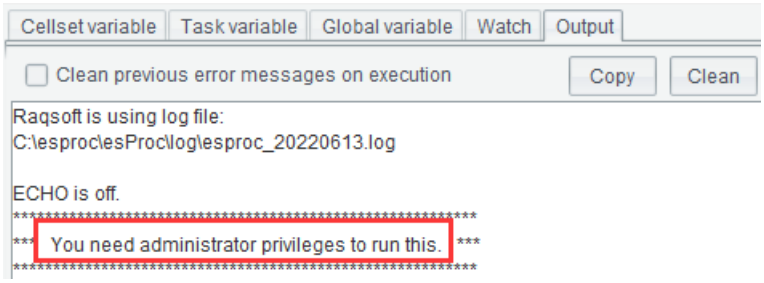


After that, a message will appear on the screen:



It indicates that the Excel plug-in has been installed successfully.

If the above message does not appear, you can find the reason by clicking Output in the right-bottom of the designer:



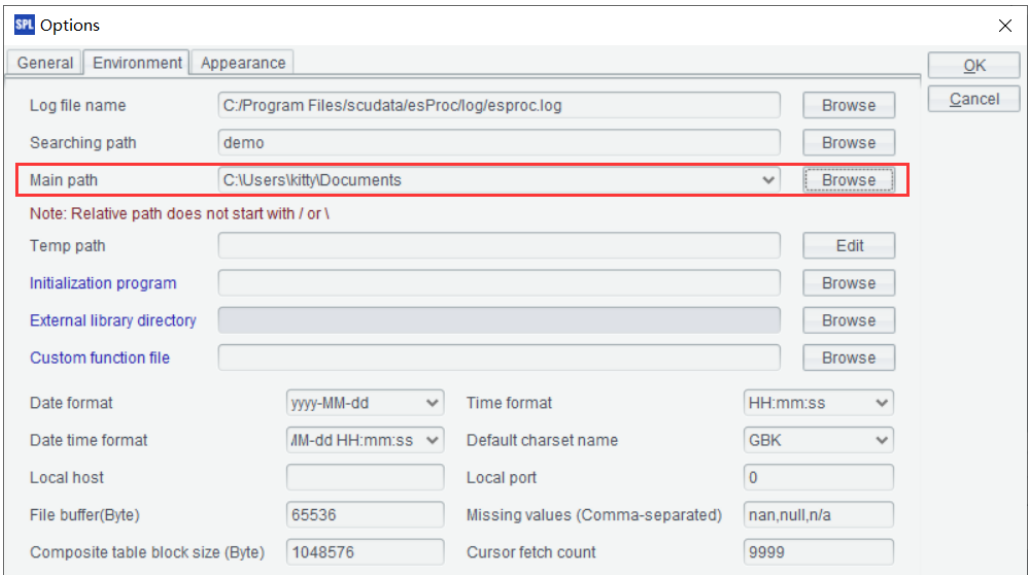
Generally, if you do not start esProc as administrator, the installation will fail.

The **spl()** function will be added to Excel after loading successfully.

2.1.3 Configure esProc main path

To configure the esProc main path, the script file (.splx) to be invoked should be placed in the target directory in advance, otherwise the search will fail. To set the main path, follow the directions below:

Open esProc, click Tool > Options, and select Environment to find Main path



Restart the esProc and Excel to make the configuration come into effect.

2.1.4 Logs

You can view exceptions in the log file. The file's path is %appdata%\esproc\tmp\Excel.log.

2.2 Using spl() function

Once the add-in is installed and configured, the spl() function will be available in Excel for executing the SPL script in Excel cell and obtaining the calculated result set.

The spl() function has two parameter formats:

1. spl(exp, arg1, …)

exp	It is an expression string in SPL syntax, starting with =. In the string, the question mark is used to represent the referenced parameter; ?1 corresponds to the first parameter, and ?2 corresponds to the second one, and so on.
arg1,…	It is the parameter value. There can be none or multiple parameters (30 at most). You can fill in the constants, or reference the current Excel cell (a single cell or a range of cells are allowed).

For complex computing requirements, execute the script file coded in SPL in the format of:

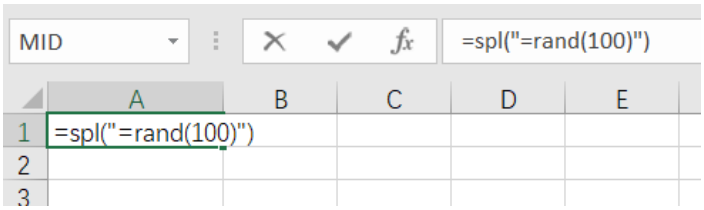
2. spl(splx(…), …)

splx()	It is the script file name without an extension. .splx will be added mandatorily, and must be followed by parentheses (). If the script has parameters, it also needs to write ?1,?2,.. etc. in () to indicate these parameters.
arg1,…	It is the parameter required by corresponding to the script file in turn. You can reference a single Excel cell or a range of cells.

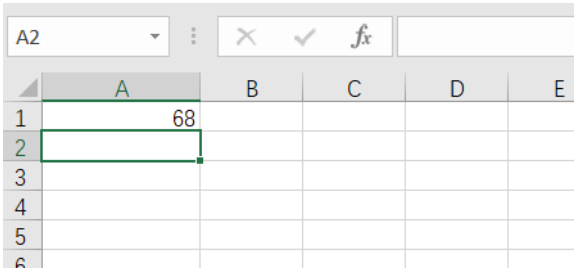
The spl() function may return a single value, a one-dimensional array or a two-dimensional array.

2.2.1 No parameters

To generate random integers within 100:

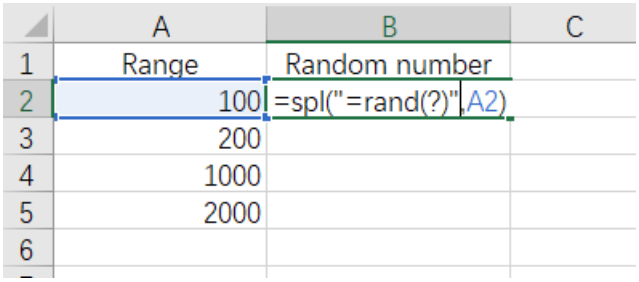


Calculation result:

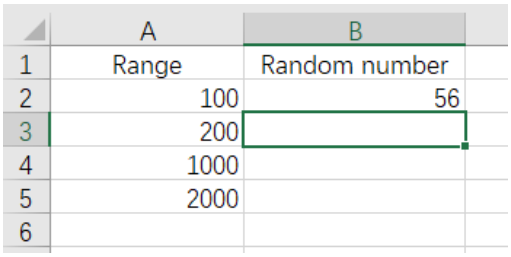


2.2.2 A single parameter

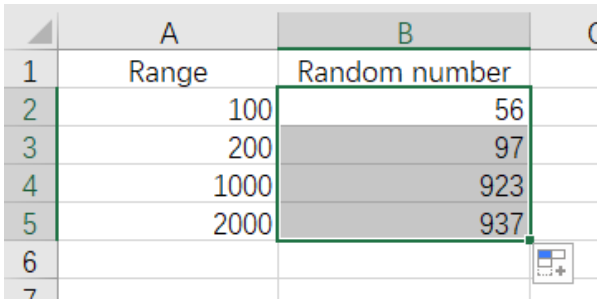
In the following Excel file, the first column is the range of random integers, and the second column is the number of the integers randomly generated according to the range given in the first column:



Calculation result:



Drag B2 down to every relevant cell to obtain the results:



2.2.3 Double quotation marks exist in expression

Here's an Excel file containing date data:

	A	B
1	Date	
2	20020101	
3	20220207	
4	20120103	
5	20150304	
6	20220108	
7	20090106	
8	20220103	
9	20080108	
10	20060115	
11	19990110	
12		

These data are not in date format and cannot be recognized by Excel. Instead, they are regarded as common string of numbers. Now we need to convert them to date type data.

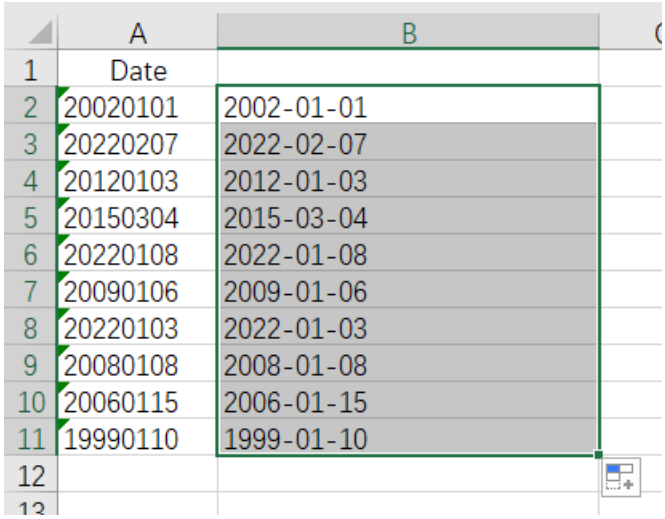
SPL provides a simple writing method: =date(A2,"yyyyMMdd"), which can parse A2 to the date format in the form of "yyyyMMdd". Such SPL script itself, however, contains double quotation marks that, when writing into the spl() function, need to be escaped according to the Excel rule by writing " as "", as shown below:

	A	B	C
1	Date		
2	20020101	=spl("=date(?,""yyyyMMdd"")",A2)	
3	20220207		
4	20120103		
5	20150304		
6	20220108		
7	20090106		
8	20220103		
9	20080108		
10	20060115		
11	19990110		
12			

The result is as follows:

	A	B
1	Date	
2	20020101	2002-01-01
3	20220207	
4	20120103	
5	20150304	
6	20220108	
7	20090106	
8	20220103	
9	20080108	
10	20060115	
11	19990110	
12		

Drag B2 down to every relevant cell to get the results:



	A	B
1	Date	
2	20020101	2002-01-01
3	20220207	2022-02-07
4	20120103	2012-01-03
5	20150304	2015-03-04
6	20220108	2022-01-08
7	20090106	2009-01-06
8	20220103	2022-01-03
9	20080108	2008-01-08
10	20060115	2006-01-15
11	19990110	1999-01-10
12		

Notes:

You need to first set the type of cell where the date data is located to text type.

2.2.4 **Multiple parameters**

In the following Excel file, the r, g, b, a represent the red, green, blue and transparency component respectively:

	A	B	C	D
1	**r**	**g**	**b**	**a**
2	194	254	200	7
3	5	150	52	53
4	208	46	44	184
5	41	82	99	194
6	238	182	212	153
7	139	87	49	171
8	79	182	227	120
9	124	118	62	20
10	27	127	133	210
11	165	206	163	114
12	220	161	131	107
13	37	1	190	110
14	211	103	171	90

Now we want to add a column on the right to compute the color value combined by the color components of each row. Enter the SPL code as follows:

	A	B	C	D	E	F	G
1	r	g	b	a	Color		
2	194	254	200	7	=spl("=rgb(?1,?2,?3,?4)",A2,B2,C2,D2)		
3	5	150	52	53			
4	208	46	44	184			
5	41	82	99	194			
6	238	182	212	153			
7	139	87	49	171			
8	79	182	227	120			
9	124	118	62	20			
10	27	127	133	210			
11	165	206	163	114			
12	220	161	131	107			
13	37	1	190	110			
14	211	103	171	90			

In this code, ?1,?2,?3,?4 respectively corresponds to the first parameter A2, the second parameter B2, the third parameter C2 and the fourth parameter D2.

After entering the above expression, the color value of the current row can be calculated as follows:

	A	B	C	D	E
1	r	g	b	a	Color
2	194	254	200	7	-312
3	5	150	52	53	
4	208	46	44	184	
5	41	82	99	194	
6	238	182	212	153	
7	139	87	49	171	
8	79	182	227	120	
9	124	118	62	20	
10	27	127	133	210	
11	165	206	163	114	
12	220	161	131	107	
13	37	1	190	110	
14	211	103	171	90	

Then drag E2 down to every relevant cell to obtain all results:

	A	B	C	D	E
1	r	g	b	a	Color
2	194	254	200	7	-312
3	5	150	52	53	-27084
4	208	46	44	184	11820
5	41	82	99	194	21091
6	238	182	212	153	-18732
7	139	87	49	171	22321
8	79	182	227	120	-18717
9	124	118	62	20	30270
10	27	127	133	210	32645
11	165	206	163	114	-12637
12	220	161	131	107	-24189
13	37	1	190	110	446
14	211	103	171	90	26539
15					

2.2.5 **A single-row range parameter**

According to the following Excel file, we want to calculate the median of each row after removing the largest value and the smallest value in each row, and then fill the result in the rightmost column:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Sampling Data								Median			
2	90	61	79	5	33	76	69	91	=spl("=?.conj().sort().m(2:-2).median()",A2:H2)			
3	55	11	80	25	14	99	27	54				
4	98	80	38	54	1	29	54	43				
5	10	83	58	2	35	26	94	35				
6	4	32	89	3	11	89	92	49				
7	6	82	51	62	80	64	8	67				
8	86	11	83	87	29	83	20	10				
9	20	97	43	57	90	49	33	49				
10	43	97	10	49	16	52	22	53				
11	60	94	89	99	64	8	3	50				
12	83	67	32	82	90	88	23	61				
13	35	23	33	41	2	98	8	73				
14	62	44	78	28	98	30	17	71				
15	92	5	78	29	93	0	76	6				
16	38	51	49	90	10	72	82	33				
17	67	63	40	62	15	12	58	82				
18	35	87	93	24	87	56	42	6				
19	47	95	44	36	88	28	35	63				
20	87	80	61	45	23	77	89	79				
21	7	14	62	87	27	32	30	98				

Enter =spl("=?.conj().sort().m(2,-2).median()",A2:H2) in I2, and Excel will pass the array combined by the range of cells A2:H2 to SPL expression "=?.conj().sort().m(2,-2).median()" to replace the parameter ? in it, and return the median with **the largest and the smallest values** removed.

Below is the result:

	A	B	C	D	E	F	G	H	I
1	Sampling Data								Median
2	90	61	79	5	33	76	69	91	72.5
3	55	11	80	25	14	99	27	54	
4	98	80	38	54	1	29	54	43	
5	10	83	58	2	35	26	94	35	
6	4	32	89	3	11	89	92	49	
7	6	82	51	62	80	64	8	67	
8	86	11	83	87	29	83	20	10	
9	20	97	43	57	90	49	33	49	
10	43	97	10	49	16	52	22	53	
11	60	94	89	99	64	8	3	50	
12	83	67	32	82	90	88	23	61	
13	35	23	33	41	2	98	8	73	
14	62	44	78	28	98	30	17	71	
15	92	5	78	29	93	0	76	6	
16	38	51	49	90	10	72	82	33	
17	67	63	40	62	15	12	58	82	
18	35	87	93	24	87	56	42	6	
19	47	95	44	36	88	28	35	63	
20	87	80	61	45	23	77	89	79	
21	7	14	62	87	27	32	30	98	
22									

Notes:

1. Excel will automatically convert the array parameter to a **two-dimensional array**, even if there is actually only one row (or one column). Therefore, there is a need to use **conj()** in the SPL script to concatenate into a single-level sequence.

At his point, dragging I2 down to every relevant row can automatically compute the median of every row, as shown in the figure blow:

	A	B	C	D	E	F	G	H	I
1	Sampling Data								Median
2	90	61	79	5	33	76	69	91	72.5
3	55	11	80	25	14	99	27	54	40.5
4	98	80	38	54	1	29	54	43	48.5
5	10	83	58	2	35	26	94	35	35
6	4	32	89	3	11	89	92	49	40.5
7	6	82	51	62	80	64	8	67	63
8	86	11	83	87	29	83	20	10	56
9	20	97	43	57	90	49	33	49	49
10	43	97	10	49	16	52	22	53	46
11	60	94	89	99	64	8	3	50	62
12	83	67	32	82	90	88	23	61	74.5
13	35	23	33	41	2	98	8	73	34
14	62	44	78	28	98	30	17	71	53
15	92	5	78	29	93	0	76	6	52.5
16	38	51	49	90	10	72	82	33	50
17	67	63	40	62	15	12	58	82	60
18	35	87	93	24	87	56	42	6	49
19	47	95	44	36	88	28	35	63	45.5
20	87	80	61	45	23	77	89	79	78
21	7	14	62	87	27	32	30	98	31
22									

2.2.6 A single-column range parameter

According to the following Excel file, we want to calculate the median of each column after **removing the largest value and the smallest value** in each column, and then fill the result in the bottom row:

	A	B	C	D	E	F	G	H	I
1	Sampling Data	90	61	79	5	33	76	69	91
2		55	11	80	25	14	99	27	54
3		98	80	38	54	1	29	54	43
4		10	83	58	2	35	26	94	35
5		4	32	89	3	11	89	92	49
6		6	82	51	62	80	64	8	67
7		86	11	83	87	29	83	20	10
8		20	97	43	57	90	49	33	49
9		43	97	10	49	16	52	22	53
10		60	94	89	99	64	8	3	50
11		83	67	32	82	90	88	23	61
12		35	23	33	41	2	98	8	73
13		62	44	78	28	98	30	17	71
14	Median	=spl("=?.conj().sort().m(2:-2).median()",B1:B13)							

Enter =spl("=?.conj().sort().m(2,-2).median()",B1:B13) in A14, and Excel will pass the array combined by the range of cells B1:B13 to SPL expression "=?.conj().sort().m(2,-2).median()" to replace the parameter ? in it, and return the median with the largest and the smallest values removed.

Below is the result:

	A	B	C	D	E	F	G	H	I
1	Sampling Data	90	61	79	5	33	76	69	91
2		55	11	80	25	14	99	27	54
3		98	80	38	54	1	29	54	43
4		10	83	58	2	35	26	94	35
5		4	32	89	3	11	89	92	49
6		6	82	51	62	80	64	8	67
7		86	11	83	87	29	83	20	10
8		20	97	43	57	90	49	33	49
9		43	97	10	49	16	52	22	53
10		60	94	89	99	64	8	3	50
11		83	67	32	82	90	88	23	61
12		35	23	33	41	2	98	8	73
13		62	44	78	28	98	30	17	71
14	Median	55							
15									

Notes:

1. Excel will automatically convert the array parameter to a **two-dimensional array**, even if there is actually only one row (or one column). Therefore, there is a need to use **conj()** in the SPL script to concatenate into a single-level sequence.

At this point, dragging B14 right to every relevant column can automatically compute the median of every column on the right, as shown in the figure blow:

	A	B	C	D	E	F	G	H	I
1	Sampling Data	90	61	79	5	33	76	69	91
2		55	11	80	25	14	99	27	54
3		98	80	38	54	1	29	54	43
4		10	83	58	2	35	26	94	35
5		4	32	89	3	11	89	92	49
6		6	82	51	62	80	64	8	67
7		86	11	83	87	29	83	20	10
8		20	97	43	57	90	49	33	49
9		43	97	10	49	16	52	22	53
10		60	94	89	99	64	8	3	50
11		83	67	32	82	90	88	23	61
12		35	23	33	41	2	98	8	73
13		62	44	78	28	98	30	17	71
14	Median	55	67	58	49	33	64	23	53
15									

2.2.7 **Multi-row, multi-column range parameter**

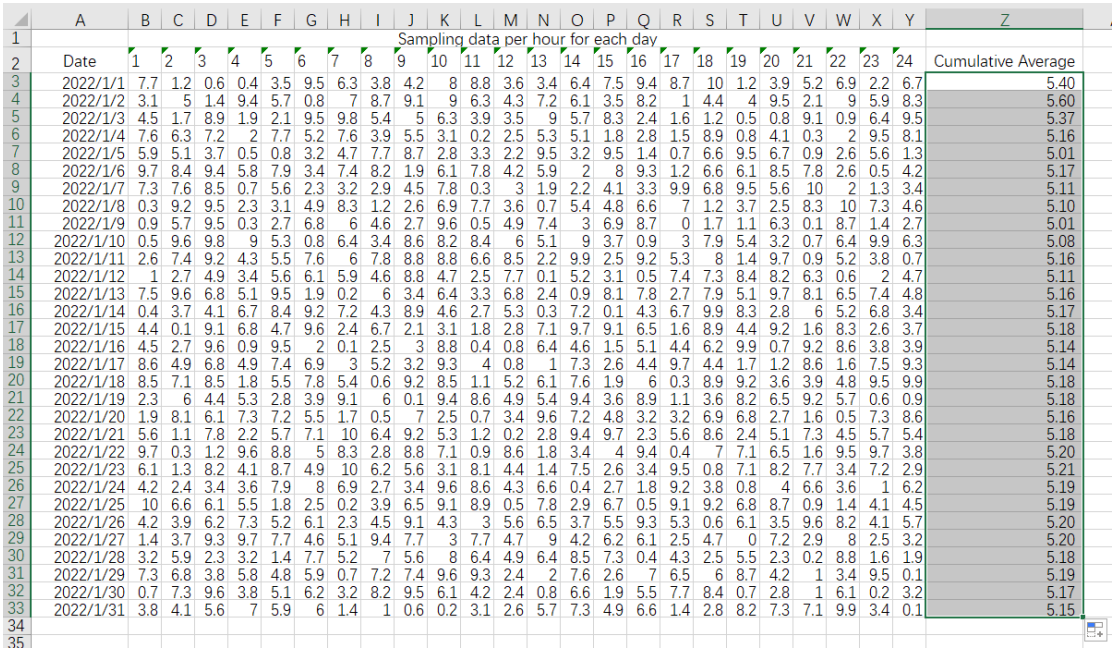
According to the following Excel file:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1		Sampling data per hour for each day																							
2	Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
3	2022-01-01	9	4	8	6	1	9	1	0	2	4	6	5	9	1	2	7	3	9	6	8	5	0	0	4
4	2022-01-02	1	1	8	7	3	7	3	5	1	8	8	7	5	1	9	8	1	8	6	6	5	5	5	4
5	2022-01-03	6	5	5	3	8	4	4	9	0	9	1	1	6	5	1	1	9	0	5	3	1	0	6	8
6	2022-01-04	8	3	0	7	6	1	9	0	9	6	2	6	1	5	6	5	3	2	3	6	8	2	1	3
7	2022-01-05	3	6	6	0	3	7	8	3	4	6	6	5	1	8	4	2	1	9	2	4	1	1	1	0
8	2022-01-06	6	8	2	1	6	2	0	5	6	4	2	1	4	8	8	9	5	2	2	2	2	6	5	4
9	2022-01-07	7	7	1	2	4	8	1	9	3	6	6	8	1	4	6	9	4	8	7	6	1	1	6	1
10	2022-01-08	5	7	8	4	7	0	3	9	9	2	4	0	5	9	3	6	5	9	5	5	1	8	0	9
11	2022-01-09	3	7	0	8	4	3	0	4	3	5	7	5	8	5	4	0	5	2	9	4	6	0	5	2
12	2022-01-10	3	2	7	9	1	1	8	6	5	8	5	8	5	2	7	8	4	5	6	8	7	6	4	9
13	2022-01-11	9	7	8	9	7	8	4	1	5	7	7	0	2	8	2	1	2	9	3	4	2	9	3	1
14	2022-01-12	5	2	0	1	7	5	5	1	8	3	1	4	8	1	3	0	3	8	1	5	7	8	1	2
15	2022-01-13	1	6	6	7	7	4	6	0	2	9	8	9	6	1	4	4	2	4	5	7	9	0	3	5
16	2022-01-14	5	1	5	3	9	4	3	0	4	6	9	7	0	2	5	8	8	0	5	2	9	5	8	8
17	2022-01-15	9	9	5	6	2	5	2	2	6	0	5	3	2	6	8	6	3	0	7	6	3	7	7	5
18	2022-01-16	5	6	6	9	4	1	7	5	5	3	4	3	0	9	4	2	0	7	4	8	1	4	8	4
19	2022-01-17	0	6	2	0	5	7	7	8	4	7	0	9	5	1	1	9	2	5	6	2	0	1	9	1
20	2022-01-18	1	3	6	0	2	8	4	4	8	8	6	6	3	6	3	0	9	5	9	4	1	9	0	2
21	2022-01-19	3	3	2	9	3	0	1	4	1	8	8	9	6	4	7	5	6	6	8	3	8	4	7	3
22	2022-01-20	3	1	2	0	6	8	1	1	9	5	3	8	5	8	3	8	9	0	3	5	3	5	3	7
23	2022-01-21	2	7	8	4	6	1	5	0	7	6	7	8	9	2	3	6	9	2	8	5	8	7	3	0
24	2022-01-22	3	9	0	8	5	6	0	9	0	6	3	9	1	5	9	1	6	9	9	5	1	0	8	4
25	2022-01-23	4	6	1	1	2	1	2	2	2	2	0	1	0	8	5	9	7	8	2	0	6	8	5	5
26	2022-01-24	1	0	1	7	3	7	4	7	6	7	0	6	9	6	6	6	8	0	1	4	1	8	3	
27	2022-01-25	0	8	4	0	2	9	0	7	9	7	6	4	6	5	5	4	8	4	8	3	3	1	5	7
28	2022-01-26	9	5	0	1	3	0	0	6	3	4	4	1	2	7	6	4	9	1	8	8	3	0	6	9
29	2022-01-27	8	0	4	6	1	0	2	8	8	4	5	7	7	6	5	7	5	5	1	1	4	8	1	5
30	2022-01-28	2	1	3	0	2	3	0	4	7	3	5	5	3	5	4	9	4	7	5	1	4	8	9	1
31	2022-01-29	6	9	0	4	6	5	0	0	2	1	6	7	8	8	1	2	6	1	0	0	2	2	6	6
32	2022-01-30	3	4	2	2	9	1	9	8	2	4	0	6	8	5	7	4	5	7	1	1	7	3	7	3
33	2022-01-31	0	2	6	7	6	9	0	2	7	8	9	0	4	7	7	9	7	5	1	3	9	1	4	2
34																									

We want to add a column on the right to calculate the cumulative average of sampling data. That is, for the row of the first day, compute the average of the first day; for the row of the second day, compute the average of the first two days; for the records of the third day, compute the average

of the first three days, and so on. Likewise, we **remove the largest and the smallest values** before computing the average.

Below is the result:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1		Sampling data per hour for each day																								
2	Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Cumulative Average
3	2022/1/1	7.7	1.2	0.6	0.4	3.5	9.5	6.3	3.8	4.2	8	8.8	3.6	3.4	6.4	7.5	9.4	8.7	10	1.2	3.9	5.2	6.9	2.2	6.7	5.40
4	2022/1/2	3.1	5	1.4	9.4	5.7	0.8	7	8.7	9.1	9	6.3	4.3	7.2	6.1	3.5	8.2	1	4.4	4	9.5	2.1	9	5.9	8.3	5.60
5	2022/1/3	4.5	1.7	8.9	1.9	2.1	9.5	9.8	5.4	5	6.3	3.9	3.5	9	5.7	8.3	2.4	1.6	1.2	0.5	0.8	9.1	0.9	6.4	9.5	5.37
6	2022/1/4	7.6	6.3	7.2	2	7.7	5.2	7.6	3.9	5.5	3.1	0.2	2.5	5.3	5.1	1.8	2.8	1.5	8.9	0.8	4.1	0.3	2	9.5	8.1	5.16
7	2022/1/5	5.9	5.1	3.7	0.5	0.8	3.2	4.7	7.7	8.7	2.8	3.3	2.2	9.5	3.2	9.5	1.4	0.7	6.6	9.5	6.7	0.9	2.6	5.6	1.3	5.01
8	2022/1/6	9.7	8.4	9.4	5.8	7.9	3.4	7.4	8.2	1.9	6.1	7.8	4.2	5.9	2	8	9.3	1.2	6.6	6.1	8.5	7.8	2.6	0.5	4.2	5.17
9	2022/1/7	7.3	7.6	8.5	0.7	5.6	2.3	3.2	2.9	4.5	7.8	0.3	3	1.9	2.2	4.1	3.3	9.9	6.8	9.5	5.6	10	2	1.3	3.4	5.11
10	2022/1/8	0.3	9.2	9.5	2.3	3.1	4.9	8.3	1.2	2.6	6.9	7.7	3.6	0.7	5.4	4.8	6.6	7	1.2	3.7	2.5	8.3	10	7.3	4.6	5.10
11	2022/1/9	0.9	5.7	9.5	0.3	2.7	6.8	6	4.6	2.7	9.6	0.5	4.9	7.4	3	6.9	8.7	0	1.7	1.1	6.3	0.1	8.7	1.4	2.7	5.01
12	2022/1/10	0.5	9.6	9.8	9	5.3	0.8	6.4	3.4	8.6	8.2	8.4	6	5.1	9	3.7	0.9	3	7.9	5.4	3.2	0.7	6.4	9.9	6.3	5.08
13	2022/1/11	2.6	7.4	9.2	4.3	5.5	7.6	6	7.8	8.8	8.8	6.6	8.5	2.2	9.9	2.5	9.2	5.3	8	1.4	9.7	0.9	5.2	3.8	0.7	5.16
14	2022/1/12	1	2.7	4.9	3.4	5.6	6.1	5.9	4.6	8.8	4.7	2.5	7.7	0.1	5.2	3.1	0.5	7.4	7.3	8.4	8.2	6.3	0.6	2	4.7	5.11
15	2022/1/13	7.5	9.6	6.8	5.1	9.5	1.9	0.2	6	3.4	6.4	3.3	6.8	2.4	0.9	8.1	7.8	2.7	7.9	5.1	9.7	8.1	6.5	7.4	4.8	5.16
16	2022/1/14	0.4	3.7	4.1	6.7	8.4	9.2	7.2	4.3	8.9	4.6	2.7	5.3	0.3	7.2	0.1	4.3	6.7	9.9	8.3	2.8	6	5.2	6.8	3.4	5.17
17	2022/1/15	4.4	0.1	9.1	6.8	4.7	9.6	2.4	6.7	2.1	3.1	1.8	2.8	7.1	9.7	9.1	6.5	1.6	8.9	4.4	9.2	1.6	8.3	2.6	3.7	5.18
18	2022/1/16	4.5	2.7	9.6	0.9	9.5	2	0.1	2.5	3	8.8	0.4	0.8	6.4	4.6	1.5	5.1	4.4	6.2	9.9	0.7	9.2	8.6	3.8	3.9	5.14
19	2022/1/17	8.6	4.9	6.8	4.9	7.4	6.9	3	5.2	3.2	9.3	4	0.8	1	7.3	2.6	4.4	9.7	4.4	1.7	1.2	8.6	1.6	7.5	9.3	5.14
20	2022/1/18	8.5	7.1	8.5	1.8	5.5	7.8	5.4	0.6	9.2	8.5	1.1	5.2	6.1	7.6	1.9	6	0.3	8.9	9.2	3.6	3.9	4.8	9.5	9.9	5.18
21	2022/1/19	2.3	6	4.4	5.3	2.8	3.9	9.1	6	0.1	9.4	8.6	4.9	5.4	9.4	3.6	8.9	1.1	3.6	8.2	6.5	9.2	5.7	0.6	0.9	5.18
22	2022/1/20	1.9	8.1	6.1	7.3	7.2	5.5	1.7	0.5	7	2.5	0.7	3.4	9.6	7.2	4.8	3.2	3.2	6.9	6.8	2.7	1.6	0.5	7.3	8.6	5.16
23	2022/1/21	5.6	1.1	7.8	2.2	5.7	7.1	10	6.4	9.2	5.3	1.2	0.2	2.8	9.4	9.7	2.3	5.6	8.6	2.4	5.1	7.3	4.5	5.7	5.4	5.18
24	2022/1/22	9.7	0.3	1.2	9.6	8.8	5	8.3	2.8	8.8	7.1	0.9	8.6	1.8	3.4	4	9.4	0.4	7	7.1	6.5	1.6	9.5	9.7	3.8	5.20
25	2022/1/23	6.1	1.3	8.2	4.1	8.7	4.9	10	6.2	5.6	3.1	8.1	4.4	1.4	7.5	2.6	3.4	9.5	0.8	7.1	8.2	7.7	3.4	7.2	2.9	5.21
26	2022/1/24	4.2	2.4	3.4	3.6	7.9	8	6.9	2.7	3.4	9.6	8.6	4.3	6.6	0.4	2.7	1.8	9.2	3.8	0.8	4	6.6	3.6	1	6.2	5.19
27	2022/1/25	10	6.6	6.1	5.5	1.8	2.5	0.2	3.9	6.5	9.1	8.9	0.5	7.8	2.9	6.7	0.5	9.1	9.2	6.8	8.7	0.9	1.4	4.1	4.5	5.19
28	2022/1/26	4.2	3.9	6.2	7.3	5.2	6.1	2.3	4.5	9.1	4.3	3	5.6	6.5	3.7	5.5	9.3	5.3	0.6	6.1	3.5	9.6	8.2	4.1	5.7	5.20
29	2022/1/27	1.4	3.7	9.3	9.7	7.7	4.6	5.1	9.4	7.7	3	7.7	4.7	9	4.2	6.2	6.1	2.5	4.7	0	7.2	2.9	8	2.5	3.2	5.20
30	2022/1/28	3.2	5.9	2.3	3.2	1.4	7.7	5.2	7	5.6	8	6.4	4.9	6.4	8.5	7.3	0.4	4.3	2.5	5.5	2.3	0.2	8.8	1.6	1.9	5.18
31	2022/1/29	7.3	6.8	3.8	5.8	4.8	5.9	0.7	7.2	7.4	9.6	9.3	2.4	2	7.6	2.6	7	6.5	6	8.7	4.2	1	3.4	9.5	0.1	5.19
32	2022/1/30	0.7	7.3	9.6	3.8	5.1	6.2	3.2	8.2	9.5	6.1	4.2	2.4	0.8	6.6	1.9	5.5	7.7	8.4	0.7	2.8	1	6.1	0.2	3.2	5.17
33	2022/1/31	3.8	4.1	5.6	7	5.9	6	1.4	1	0.6	0.2	3.1	2.6	5.7	7.3	4.9	6.6	1.4	2.8	8.2	7.3	7.1	9.9	3.4	0.1	5.15
34																										
35																										

Enter the expression: =spl("=?.conj().sort().m(2:-2).avg()",$B$3:Y3) in Z3 and get the result 5.40.

As can be seen from the expression that the parameter passed in is $B$3:Y3, where the first cell B3 is added with two symbols $, the reason is that we should always keep accumulating from cell B3 when copying the expression down since what we want to calculate is the cumulative average.

Drag Z3 down to every relevant row to get the cumulative average of every row.

2.2.8 **Mixed parameters**

The following Excel file contains the sampling data per hour for each day:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1		Sampling data per hour for each day																									
2	Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	n	Average of cumulative topn
3	2022/1/1	5.4	13	4.6	14	8.1	11	1.7	3.3	1	8.5	15	0.6	1.6	15	7.3	9.5	11	7.2	6.2	7.4	13	13	4.9	2.8	5	
4	2022/1/2	9.4	15	6.8	0.9	12	6.3	8.2	6.1	8.7	2.6	5.5	6.8	4.5	12	3.4	6.9	9.7	8.9	14	8.9	15	13	8.4	14	5	
5	2022/1/3	11	11	12	10	0.2	11	2.8	9.8	6	7.2	11	12	13	9.5	3.2	13	1.1	8	2	6.3	2.6	10	8.1	4.4	5	
6	2022/1/4	14	1.7	8	14	14	13	2	4.5	10	2.5	15	6.7	0.6	13	14	11	13	12	3.7	12	3.7	4.5	4.7	10	5	
7	2022/1/5	14	8.5	14	5	8.4	13	5.8	7.5	4.4	9.1	13	9.3	2.1	13	4.9	15	14	11	9.2	5	8.6	13	0.6	5.9	5	
8	2022/1/6	2.8	12	3.5	0.2	1.6	15	8.7	12	5.1	0.9	8	9.5	8.1	15	8.9	6.4	8.7	9.2	15	15	15	13	3.7	15	5	
9	2022/1/7	12	8.5	14	12	14	12	9.1	13	11	1.8	6.6	4	3.4	2.6	9.9	11	6.1	2.5	12	8.8	0.2	6	3.1	6.1	5	
10	2022/1/8	3.7	8.4	4.6	12	13	1	12	11	7.7	5.3	3.3	2.4	1.8	3.4	8.4	9.3	12	3.1	11	2.2	5.8	11	14	5.3	5	
11	2022/1/9	10	12	14	11	0.9	12	9.9	9.7	11	7	0.4	14	6.8	2	15	7.3	8	3.9	9.8	8.4	13	13	9	2.9	5	
12	2022/1/10	2.3	5.5	12	9.6	15	14	7.9	9	3.2	4.7	10	0.5	4.8	4.9	3.9	11	14	10	12	4.3	13	4.4	5.4	3.7	5	
13	2022/1/11	5.1	13	1.8	7	15	3.7	6.7	8.7	6.2	2.9	7	4.1	14	5.3	7.8	6.3	12	14	11	3.4	0.6	8.6	6.4	4.7	5	
14	2022/1/12	4.3	4	7.1	14	7.6	5.2	3.8	14	3.1	6.4	14	14	7.2	11	12	2.4	14	5	14	15	13	2.8	9.1	4.5	5	
15	2022/1/13	5.6	2.3	8.2	6.3	13	10	13	4	4.9	1.6	7	11	14	11	7.1	10	5.2	1.8	0.2	0.2	8.1	12	0.1	9.1	5	
16	2022/1/14	0.8	6.4	1.4	5.4	1.8	12	5.5	13	7.9	12	4.4	5	13	7.4	12	14	8.5	9.6	2.5	0	12	13	9.7	12	5	
17	2022/1/15	6.1	0.1	6	2.5	5.3	5.6	13	4.3	6.9	5.8	12	7.6	7	6.3	0.6	14	0.1	7	4.2	11	0.3	14	5.2	2.1	5	
18	2022/1/16	0.7	2.2	4.9	13	4.2	8.1	8.5	13	12	12	6.4	3.8	6.7	13	6.2	5.6	14	0.9	2.9	4.9	14	4.1	15	0.9	10	
19	2022/1/17	3.3	3.6	13	0.9	11	10	11	14	0.7	5.2	4	10	12	6.3	3.7	12	9.5	8	2.7	9.5	9.9	1.3	7.2	15	10	
20	2022/1/18	9.6	0.6	14	10	15	5.7	5.8	13	4.5	0.8	11	14	13	9.5	0.2	8.9	12	10	6.9	8.5	5.9	8	2.6	1.1	10	
21	2022/1/19	9.8	3.5	1.1	14	9.2	7.9	9.8	8.8	5.4	13	14	0.8	7.8	7.5	12	4.5	0.4	7.1	0	0.7	14	11	10	1.3	10	
22	2022/1/20	0.4	8.3	12	14	8.2	14	7.3	11	8.6	9.3	1.5	14	7.4	14	8.2	8.3	5.2	2.1	2.5	4.9	5.9	15	15	11	10	
23	2022/1/21	2.8	6.5	14	11	11	3.7	8.2	7.5	6.7	3.6	4.2	13	6.1	4	0.9	1.6	9.8	9.5	5.3	1.7	0.7	5.5	2.9	11	10	
24	2022/1/22	1	14	7.7	9.5	6.3	11	13	0.8	14	7.2	10	6.9	6.6	12	14	14	4.5	5.9	4	8.2	15	3	9.1	14	10	
25	2022/1/23	7	9.1	14	2.2	11	4.3	4.8	3.5	11	1.4	4.9	12	2.8	15	2.4	2.8	9.9	2.8	1.3	12	3.5	5.8	2.8	2.2	10	
26	2022/1/24	0	11	11	12	12	14	14	3.7	15	8.8	7.9	6.3	2.8	5.2	12	2.9	13	12	7.9	2.1	9.3	15	13	3.7	10	
27	2022/1/25	10	13	14	14	11	13	2.8	9.2	0.8	9.9	13	13	4.9	13	2.9	14	3.2	9.2	8.5	8	8.9	14	6.9	14	10	
28	2022/1/26	4.5	0.7	12	8.8	7	11	3.2	10	15	0.2	1.8	9.7	15	4.1	1.7	6.9	8.3	1	10	0.2	7.1	14	12	8.1	10	
29	2022/1/27	3.1	0.4	14	9.5	3.8	13	12	0.1	7.5	11	4.5	6.5	15	13	1.1	11	7.9	5.5	15	4.4	15	7.9	6.5	7.9	10	
30	2022/1/28	9.9	12	8.9	8.7	11	13	8.1	10	12	12	12	14	5.9	6	7.3	14	6.1	4.5	12	14	2.4	11	14	4.7	10	
31	2022/1/29	1.2	5.1	13	11	9.1	12	14	5.6	7.6	9.2	1.5	11	1.3	15	13	9.2	5.4	8.8	0.9	9.4	13	14	3	12	10	
32	2022/1/30	8.2	11	14	8.1	7.8	5.2	1	2.6	4.2	7.9	9.4	11	12	12	8.6	4.1	14	9.9	12	13	10	4.4	9.7	6.1	10	
33	2022/1/31	12	5	14	10	4.2	1.5	5.7	1.5	6.3	7.1	15	7.6	6.9	5.4	4	11	7.6	4.9	8.4	7	11	13	8.6	6.7	10	

Now we want to add a column on the right to calculate the average of the largest n values in all sampling data of this month as of the current day, and n is determined by the data in column Z.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1		Sampling data per hour for each day																										
2	Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	n	Average of cumulative topn	
3	2022/1/1	5.4	13	4.6	14	8.1	11	1.7	3.3	1	8.5	15	0.6	1.6	15	7.3	9.5	11	7.2	6.2	7.4	13	13	4.9	2.8	5	=spl("=?1.top@2(-?2).avg()",$B$3:Y3,Z3)	
4	2022/1/2	9.4	15	6.8	0.9	12	6.3	8.2	6.1	8.7	2.6	5.5	6.8	4.5	12	3.4	6.9	9.7	8.9	14	8.9	15	13	8.4	14	5		
5	2022/1/3	11	11	12	10	0.2	11	2.8	9.8	6	7.2	11	12	13	9.5	3.2	13	1.1	8	2	6.3	2.6	10	8.1	4.4	5		
6	2022/1/4	14	1.7	8	14	14	13	2	4.5	10	2.5	15	6.7	0.6	13	14	11	13	12	3.7	12	3.7	4.5	4.7	10	5		
7	2022/1/5	14	8.5	14	5	8.4	13	5.8	7.5	4.4	9.1	13	9.3	2.1	13	4.9	15	14	11	9.2	5	8.6	13	0.6	5.9	5		
8	2022/1/6	2.8	12	3.5	0.2	1.6	15	8.7	12	5.1	0.9	8	9.5	8.1	15	8.9	6.4	8.7	9.2	15	15	15	13	3.7	15	5		
9	2022/1/7	12	8.5	14	12	14	12	9.1	13	11	1.8	6.6	4	3.4	2.6	9.9	11	6.1	2.5	12	8.8	0.2	6	3.1	6.1	5		
10	2022/1/8	3.7	8.4	4.6	12	13	1	12	11	7.7	5.3	3.3	2.4	1.8	3.4	8.4	9.3	12	3.1	11	2.2	5.8	11	14	5.3	5		
11	2022/1/9	10	12	14	11	0.9	12	9.9	9.7	11	7	0.4	14	6.8	2	15	7.3	8	3.9	9.8	8.4	13	13	9	2.9	5		
12	2022/1/10	2.3	5.5	12	9.6	15	14	7.9	9	3.2	4.7	10	0.5	4.8	4.9	3.9	11	14	10	12	4.3	13	4.4	5.4	3.7	5		
13	2022/1/11	5.1	13	1.8	7	15	3.7	6.7	8.7	6.2	2.9	7	4.1	14	5.3	7.8	6.3	12	14	11	3.4	0.6	8.6	6.4	4.7	5		
14	2022/1/12	4.3	4	7.1	14	7.6	5.2	3.8	14	3.1	6.4	14	14	7.2	11	12	2.4	14	5	14	15	13	2.8	9.1	4.5	5		
15	2022/1/13	5.6	2.3	8.2	6.3	13	10	13	4	4.9	1.6	7	11	14	11	7.1	10	5.2	1.8	0.2	0.2	8.1	12	0.1	9.1	5		
16	2022/1/14	0.8	6.4	1.4	5.4	1.8	12	5.5	13	7.9	12	4.4	5	13	7.4	12	14	8.5	9.6	2.5	0	12	13	9.7	12	5		
17	2022/1/15	6.1	0.1	6	2.5	5.3	5.6	13	4.3	6.9	5.8	12	7.6	7	6.3	0.6	14	0.1	7	4.2	11	0.3	14	5.2	2.1	5		
18	2022/1/16	0.7	2.2	4.9	13	4.2	8.1	8.5	13	12	12	6.4	3.8	6.7	13	6.2	5.6	14	0.9	2.9	4.9	14	4.1	15	0.9	10		
19	2022/1/17	3.3	3.6	13	0.9	11	10	11	14	0.7	5.2	4	10	12	6.3	3.7	12	9.5	8	2.7	9.5	9.9	1.3	7.2	15	10		
20	2022/1/18	9.6	0.6	14	10	15	5.7	5.8	13	4.5	0.8	11	14	13	9.5	0.2	8.9	12	10	6.9	8.5	5.9	8	2.6	1.1	10		
21	2022/1/19	9.8	3.5	1.1	14	9.2	7.9	9.8	8.8	5.4	13	14	0.8	7.8	7.5	12	4.5	0.4	7.1	0	0.7	14	11	10	1.3	10		
22	2022/1/20	0.4	8.3	12	14	8.2	14	7.3	11	8.6	9.3	1.5	14	7.4	14	8.2	8.3	5.2	2.1	2.5	4.9	5.9	15	15	11	10		
23	2022/1/21	2.8	6.5	14	11	11	3.7	8.2	7.5	6.7	3.6	4.2	13	6.1	4	0.9	1.6	9.8	9.5	5.3	1.7	0.7	5.5	2.9	11	10		
24	2022/1/22	1	14	7.7	9.5	6.3	11	13	0.8	14	7.2	10	6.9	6.6	12	14	14	4.5	5.9	4	8.2	15	3	9.1	14	10		
25	2022/1/23	7	9.1	14	2.2	11	4.3	4.8	3.5	11	1.4	4.9	12	2.8	15	2.4	2.8	9.9	2.8	1.3	12	3.5	5.8	2.8	2.2	10		
26	2022/1/24	0	11	11	12	12	14	14	3.7	15	8.8	7.9	6.3	2.8	5.2	12	2.9	13	12	7.9	2.1	9.3	15	13	3.7	10		
27	2022/1/25	10	13	14	14	11	13	2.8	9.2	0.8	9.9	13	13	4.9	13	2.9	14	3.2	9.2	8.5	8	8.9	14	6.9	14	10		
28	2022/1/26	4.5	0.7	12	8.8	7	11	3.2	10	15	0.2	1.8	9.7	15	4.1	1.7	6.9	8.3	1	10	0.2	7.1	14	12	8.1	10		
29	2022/1/27	3.1	0.4	14	9.5	3.8	13	12	0.1	7.5	11	4.5	6.5	15	13	1.1	11	7.9	5.5	15	4.4	15	7.9	6.5	7.9	10		
30	2022/1/28	9.9	12	8.9	8.7	11	13	8.1	10	12	12	12	14	5.9	6	7.3	14	6.1	4.5	12	14	2.4	11	14	4.7	10		
31	2022/1/29	1.2	5.1	13	11	9.1	12	14	5.6	7.6	9.2	1.5	11	1.3	15	13	9.2	5.4	8.8	0.9	9.4	13	14	3	12	10		
32	2022/1/30	8.2	11	14	8.1	7.8	5.2	1	2.6	4.2	7.9	9.4	11	12	12	8.6	4.1	14	9.9	12	13	10	4.4	9.7	6.1	10		
33	2022/1/31	12	5	14	10	4.2	1.5	5.7	1.5	6.3	7.1	15	7.6	6.9	5.4	4	11	7.6	4.9	8.4	7	11	13	8.6	6.7	10		

Enter the SPL expression in AA3 as follows:

=spl("=?1.top@2(-?2).avg()", $B$3:Y3, Z3)

The first parameter **=?1.top@2(-?2).avg()** is the SPL expression, which means that the top n values are calculated for parameter ?1, and n is determined by parameter ?2; Then calculate the average of top n values.

The second parameter **$B$3:Y3** is the parameter passed to SPL expression and corresponds to ?1

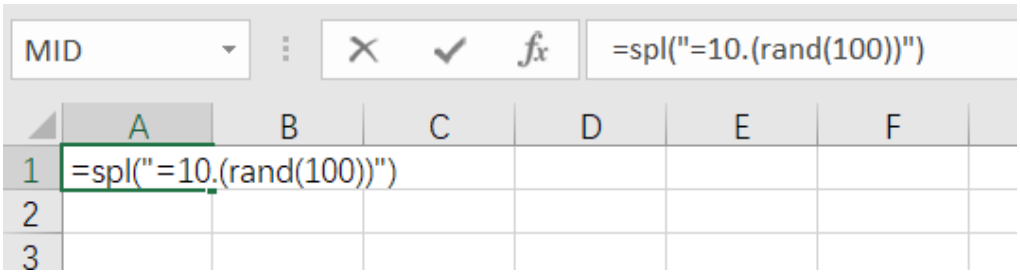
The third parameter **Z3** is the parameter passed to SPL expression and corresponds to ?2.

Drag AA3 down to every relevant row to get the final result:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1		Sampling data per hour for each day																									
2	Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	n	Average of cumulative topn
3	2022/1/1	5.4	13	4.6	14	8.1	11	1.7	3.3	1	8.5	15	0.6	1.6	15	7.3	9.5	11	7.2	6.2	7.4	13	13	4.9	2.8	5	13.94
4	2022/1/2	9.4	15	6.8	0.9	12	6.3	8.2	6.1	8.7	2.6	5.5	6.8	4.5	12	3.4	6.9	9.7	8.9	14	8.9	15	13	8.4	14	5	14.56
5	2022/1/3	11	11	12	10	0.2	11	2.8	9.8	6	7.2	11	12	13	9.5	3.2	13	1.1	8	2	6.3	2.6	10	8.1	4.4	5	14.56
6	2022/1/4	14	1.7	8	14	14	13	2	4.5	10	2.5	15	6.7	0.6	13	14	11	13	12	3.7	12	3.7	4.5	4.7	10	5	14.70
7	2022/1/5	14	8.5	14	5	8.4	13	5.8	7.5	4.4	9.1	13	9.3	2.1	13	4.9	15	14	11	9.2	5	8.6	13	0.6	5.9	5	14.72
8	2022/1/6	2.8	12	3.5	0.2	1.6	15	8.7	12	5.1	0.9	8	9.5	8.1	15	8.9	6.4	8.7	9.2	15	15	15	13	3.7	15	5	14.88
9	2022/1/7	12	8.5	14	12	14	12	9.1	13	11	1.8	6.6	4	3.4	2.6	9.9	11	6.1	2.5	12	8.8	0.2	6	3.1	6.1	5	14.88
10	2022/1/8	3.7	8.4	4.6	12	13	1	12	11	7.7	5.3	3.3	2.4	1.8	3.4	8.4	9.3	12	3.1	11	2.2	5.8	11	14	5.3	5	14.88
11	2022/1/9	10	12	14	11	0.9	12	9.9	9.7	11	7	0.4	14	6.8	2	15	7.3	8	3.9	9.8	8.4	13	13	9	2.9	5	14.94
12	2022/1/10	2.3	5.5	12	9.6	15	14	7.9	9	3.2	4.7	10	0.5	4.8	4.9	3.9	11	14	10	12	4.3	13	4.4	5.4	3.7	5	14.94
13	2022/1/11	5.1	13	1.8	7	15	3.7	6.7	8.7	6.2	2.9	7	4.1	14	5.3	7.8	6.3	12	14	11	3.4	0.6	8.6	6.4	4.7	5	14.96
14	2022/1/12	4.3	4	7.1	14	7.6	5.2	3.8	14	3.1	6.4	14	14	7.2	11	12	2.4	14	5	14	15	13	2.8	9.1	4.5	5	14.98
15	2022/1/13	5.6	2.3	8.2	6.3	13	10	13	4	4.9	1.6	7	11	14	11	7.1	10	5.2	1.8	0.2	0.2	8.1	12	0.1	9.1	5	14.98
16	2022/1/14	0.8	6.4	1.4	5.4	1.8	12	5.5	13	7.9	12	4.4	5	13	7.4	12	14	8.5	9.6	2.5	0	12	13	9.7	12	5	14.98
17	2022/1/15	6.1	0.1	6	2.5	5.3	5.6	13	4.3	6.9	5.8	12	7.6	7	6.3	0.6	14	0.1	7	4.2	11	0.3	14	5.2	2.1	5	14.98
18	2022/1/16	0.7	2.2	4.9	13	4.2	8.1	8.5	13	12	12	6.4	3.8	6.7	13	6.2	5.6	14	0.9	2.9	4.9	14	4.1	15	0.9	10	14.87
19	2022/1/17	3.3	3.6	13	0.9	11	10	11	14	0.7	5.2	4	10	12	6.3	3.7	12	9.5	8	2.7	9.5	9.9	1.3	7.2	15	10	14.90
20	2022/1/18	9.6	0.6	14	10	15	5.7	5.8	13	4.5	0.8	11	14	13	9.5	0.2	8.9	12	10	6.9	8.5	5.9	8	2.6	1.1	10	14.91
21	2022/1/19	9.8	3.5	1.1	14	9.2	7.9	9.8	8.8	5.4	13	14	0.8	7.8	7.5	12	4.5	0.4	7.1	0	0.7	14	11	10	1.3	10	14.91
22	2022/1/20	0.4	8.3	12	14	8.2	14	7.3	11	8.6	9.3	1.5	14	7.4	14	8.2	8.3	5.2	2.1	2.5	4.9	5.9	15	15	11	10	14.92
23	2022/1/21	2.8	6.5	14	11	11	3.7	8.2	7.5	6.7	3.6	4.2	13	6.1	4	0.9	1.6	9.8	9.5	5.3	1.7	0.7	5.5	2.9	11	10	14.92
24	2022/1/22	1	14	7.7	9.5	6.3	11	13	0.8	14	7.2	10	6.9	6.6	12	14	14	4.5	5.9	4	8.2	15	3	9.1	14	10	14.93
25	2022/1/23	7	9.1	14	2.2	11	4.3	4.8	3.5	11	1.4	4.9	12	2.8	15	2.4	2.8	9.9	2.8	1.3	12	3.5	5.8	2.8	2.2	10	14.93
26	2022/1/24	0	11	11	12	12	14	14	3.7	15	8.8	7.9	6.3	2.8	5.2	12	2.9	13	12	7.9	2.1	9.3	15	13	3.7	10	14.93
27	2022/1/25	10	13	14	14	11	13	2.8	9.2	0.8	9.9	13	13	4.9	13	2.9	14	3.2	9.2	8.5	8	8.9	14	6.9	14	10	14.93
28	2022/1/26	4.5	0.7	12	8.8	7	11	3.2	10	15	0.2	1.8	9.7	15	4.1	1.7	6.9	8.3	1	10	0.2	7.1	14	12	8.1	10	14.94
29	2022/1/27	3.1	0.4	14	9.5	3.8	13	12	0.1	7.5	11	4.5	6.5	15	13	1.1	11	7.9	5.5	15	4.4	15	7.9	6.5	7.9	10	14.94
30	2022/1/28	9.9	12	8.9	8.7	11	13	8.1	10	12	12	12	14	5.9	6	7.3	14	6.1	4.5	12	14	2.4	11	14	4.7	10	14.94
31	2022/1/29	1.2	5.1	13	11	9.1	12	14	5.6	7.6	9.2	1.5	11	1.3	15	13	9.2	5.4	8.8	0.9	9.4	13	14	3	12	10	14.94
32	2022/1/30	8.2	11	14	8.1	7.8	5.2	1	2.6	4.2	7.9	9.4	11	12	12	8.6	4.1	14	9.9	12	13	10	4.4	9.7	6.1	10	14.94
33	2022/1/31	12	5	14	10	4.2	1.5	5.7	1.5	6.3	7.1	15	7.6	6.9	5.4	4	11	7.6	4.9	8.4	7	11	13	8.6	6.7	10	14.95

2.2.9 **Return a one-dimensional array**

To generate 10 random integers within 100 and fill them in the first row in sequence:



After Entering the expression in A1, press Ctrl-Enter to execute the macro defined in esproc_template.xla, and fill the calculation result into adjacent cell:

	A	B	C	D	E	F	G	H	I	J
1	71	58	27	57	37	87	79	91	64	14
2										
3										

When the expression returns a one-dimensional array, pressing Ctrl-Enter will fill all members of the array into a row of cells from left to right beginning from the current cell. If you don't use the hotkey, only the first member is filled into the cell holding the expression.

2.2.10 **Return a two-dimensional array**

We have an Employee.xlsx file:

EID	NAME	SURNAME	GENDER	DEPT	SALARY	BIRTHDAY	HIREDATE	STATE
1	Rebecca	Moore	F	R&D	7000	1974/11/20	2005/03/11	California
2	Ashley	Wilson	F	Finance	11000	1980/07/19	2008/03/16	New York
3	Rachel	Johnson	F	Sales	9000	1970/12/17	2010/12/01	New Mexico
4	Emily	Smith	F	HR	7000	1985/03/07	2006/08/15	Texas
5	Ashley	Smith	F	R&D	16000	1975/05/13	2004/07/30	Texas
6	Matthew	Johnson	M	Sales	11000	1984/07/07	2005/07/07	California
7	Alexis	Smith	F	Sales	9000	1972/08/16	2002/08/16	Illinois
8	Megan	Wilson	F	Marketing	11000	1979/04/19	1984/04/19	California
9	Victoria	Davis	F	HR	3000	1983/12/07	2009/12/07	Texas
10	Ryan	Johnson	M	R&D	13000	1976/03/12	2006/03/12	Pennsylvania
11	Jacob	Moore	M	Sales	12000	1974/12/16	2004/12/16	Texas
12	Jessica	Davis	F	Sales	7000	1980/09/11	2008/09/11	New York
13	Daniel	Davis	M	Finance	10000	1982/05/14	2010/05/14	Florida
14	Alyssa	Wilson	F	Sales	4000	1977/12/24	2005/12/24	Florida
15	Alexis	Smith	F	Sales	8000	1983/07/10	2006/07/10	New York
16	Christopher	Hernandez	M	Production	9000	1979/06/27	2007/06/27	Florida
17	Hannah	Johnson	F	Marketing	4000	1980/07/19	2006/07/19	Texas
18	Jonathan	Moore	M	Administratior	7000	1971/03/07	2000/03/07	Florida
19	Samantha	Williams	F	Production	10000	1974/03/29	2004/03/29	Pennsylvania
20	Alexis	Allen	F	Administratior	16000	1977/08/07	2007/08/07	Florida
21	Jacob	Moore	M	Marketing	10000	1975/09/17	2005/09/17	Pennsylvania
22	Jacob	Davis	M	R&D	16000	1985/05/07	2001/05/07	Texas
23	Joseph	Turner	M	Finance	6000	1983/08/27	2003/08/27	California
24	Chloe	Smith	F	Finance	10000	1978/08/20	2008/08/20	Florida

We want to count the number of employees in each department by the column DEPT. Enter the SPL script in A1 as follows:

	A	B	C	D	E	F
1	=spl("=T(""Employee.xlsx"").groups(DEPT;count(1):'The number of people')")					
2						
3						

This Excel file is preferably stored under esProc main path, so that it can be located without writing the path name, otherwise its full path name needs to be written in the expression.

Press Ctrl-Enter to fill the calculation result into adjacent cell:

	A	B
1	DEPT	The number of people
2	Administration	4
3	Finance	24
4	HR	19
5	Marketing	99
6	Production	91
7	R&D	29
8	Sales	187
9	Technology	47
10		
11		

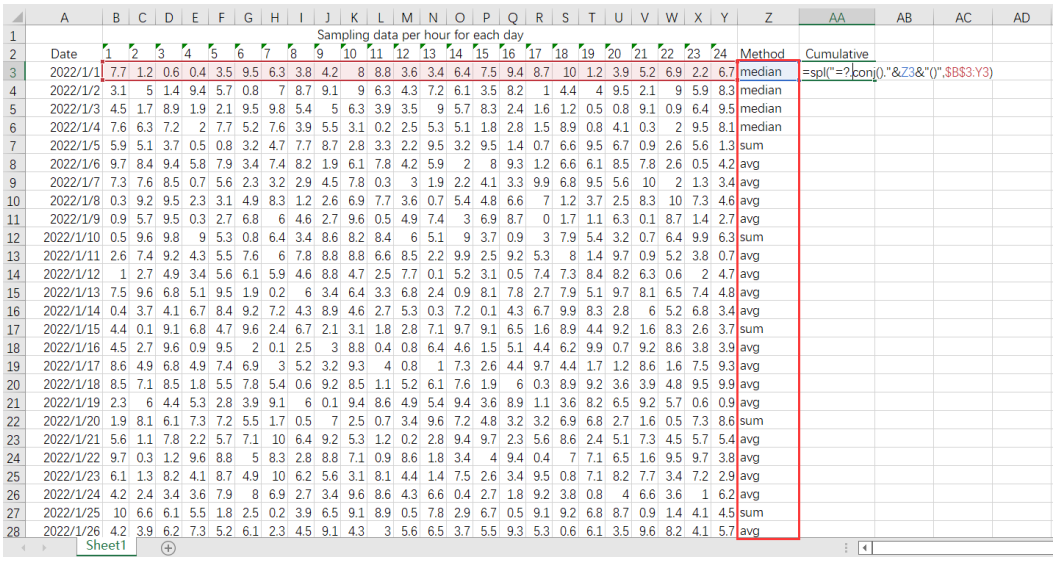
Notes:

1. You need to press **Ctrl-Enter** to trigger the fill action when a **two-dimensional array** is returned. Values will be filled **rightward and downward to a range of cells** beginning from the current cell. Only the first member of the array is filled into the cell holding the expression if the hotkey isn't used.

2. If there is " in the SPL script, it needs to be escaped to "" according to the Excel rule.

2.2.11 **Concatenate SPL expression dynamically**

According to the following Excel file:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1		Sampling data per hour for each day																												
2	Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Method	Cumulative			
3	2022/1/1	7.7	1.2	0.6	0.4	3.5	9.5	6.3	3.8	4.2	8	8.8	3.6	3.4	6.4	7.5	9.4	8.7	10	1.2	3.9	5.2	6.9	2.2	6.7	median	=spl("=?.conj()."&Z3&"()",$B$3:Y3)			
4	2022/1/2	3.1	5	1.4	9.4	5.7	0.8	7	8.7	9.1	9	6.3	4.3	7.2	6.1	3.5	8.2	1	4.4	4	9.5	2.1	9	5.9	8.3	median				
5	2022/1/3	4.5	1.7	8.9	1.9	2.1	9.5	9.8	5.4	5	6.3	3.9	3.5	9	5.7	8.3	2.4	1.6	1.2	0.5	0.8	9.1	0.9	6.4	9.5	median				
6	2022/1/4	7.6	6.3	7.2	2	7.7	5.2	7.6	3.9	5.5	3.1	0.2	2.5	5.3	5.1	1.8	2.8	1.5	8.9	0.8	4.1	0.3	2	9.5	8.1	median				
7	2022/1/5	5.9	5.1	3.7	0.5	0.8	3.2	4.7	7.7	8.7	2.8	3.3	2.2	9.5	3.2	9.5	1.4	0.7	6.6	9.5	6.7	0.9	2.6	5.6	1.3	sum				
8	2022/1/6	9.7	8.4	9.4	5.8	7.9	3.4	7.4	8.2	1.9	6.1	7.8	4.2	5.9	2	8	9.3	1.2	6.6	6.1	8.5	7.8	2.6	0.5	4.2	avg				
9	2022/1/7	7.3	7.6	8.5	0.7	5.6	2.3	3.2	2.9	4.5	7.8	0.3	3	1.9	2.2	4.1	3.3	9.9	6.8	9.5	5.6	10	2	1.3	3.4	avg				
10	2022/1/8	0.3	9.2	9.5	2.3	3.1	4.9	8.3	1.2	2.6	6.9	7.7	3.6	0.7	5.4	4.8	6.6	7	1.2	3.7	2.5	8.3	10	7.3	4.6	avg				
11	2022/1/9	0.9	5.7	9.5	0.3	2.7	6.8	6	4.6	2.7	9.6	0.5	4.9	7.4	3	6.9	8.7	0	1.7	1.1	6.3	0.1	8.7	1.4	2.7	avg				
12	2022/1/10	0.5	9.6	9.8	9	5.3	0.8	6.4	3.4	8.6	8.2	8.4	6	5.1	9	3.7	0.9	3	7.9	5.4	3.2	0.7	6.4	9.9	6.3	sum				
13	2022/1/11	2.6	7.4	9.2	4.3	5.5	7.6	6	7.8	8.8	8.8	6.6	8.5	2.2	9.9	2.5	9.2	5.3	8	1.4	9.7	0.9	5.2	3.8	0.7	avg				
14	2022/1/12	1	2.7	4.9	3.4	5.6	6.1	5.9	4.6	8.8	4.7	2.5	7.7	0.1	5.2	3.1	0.5	7.4	7.3	8.4	8.2	6.3	0.6	2	4.7	avg				
15	2022/1/13	7.5	9.6	6.8	5.1	9.5	1.9	0.2	6	3.4	6.4	3.3	6.8	2.4	0.9	8.1	7.8	2.7	7.9	5.1	9.7	8.1	6.5	7.4	4.8	avg				
16	2022/1/14	0.4	3.7	4.1	6.7	8.4	9.2	7.2	4.3	8.9	4.6	2.7	5.3	0.3	7.2	0.1	4.3	6.7	9.9	8.3	2.8	6	5.2	6.8	3.4	avg				
17	2022/1/15	4.4	0.1	9.1	6.8	4.7	9.6	2.4	6.7	2.1	3.1	1.8	2.8	7.1	9.7	9.1	6.5	1.6	8.9	4.4	9.2	1.6	8.3	2.6	3.7	sum				
18	2022/1/16	4.5	2.7	9.6	0.9	9.5	2	0.1	2.5	3	8.8	0.4	0.8	6.4	4.6	1.5	5.1	4.4	6.2	9.9	0.7	9.2	8.6	3.8	3.9	avg				
19	2022/1/17	8.6	4.9	6.8	4.9	7.4	6.9	3	5.2	3.2	9.3	4	0.8	1	7.3	2.6	4.4	9.7	4.4	1.7	1.2	8.6	1.6	7.5	9.3	avg				
20	2022/1/18	8.5	7.1	8.5	1.8	5.5	7.8	5.4	0.6	9.2	8.5	1.1	5.2	6.1	7.6	1.9	6	0.3	8.9	9.2	3.6	3.9	4.8	9.5	9.9	avg				
21	2022/1/19	2.3	6	4.4	5.3	2.8	3.9	9.1	6	0.1	9.4	8.6	4.9	5.4	9.4	3.6	8.9	1.1	3.6	8.2	6.5	9.2	5.7	0.6	0.9	avg				
22	2022/1/20	1.9	8.1	6.1	7.3	7.2	5.5	1.7	0.5	7	2.5	0.7	3.4	9.6	7.2	4.8	3.2	3.2	6.9	6.8	2.7	1.6	0.5	7.3	8.6	sum				
23	2022/1/21	5.6	1.1	7.8	2.2	5.7	7.1	10	6.4	9.2	5.3	1.2	0.2	2.8	9.4	9.7	2.3	5.6	8.6	2.4	5.1	7.3	4.5	5.7	5.4	avg				
24	2022/1/22	9.7	0.3	1.2	9.6	8.8	5	8.3	2.8	8.8	7.1	0.9	8.6	1.8	3.4	4	9.4	0.4	7	7.1	6.5	1.6	9.5	9.7	3.8	avg				
25	2022/1/23	6.1	1.3	8.2	4.1	8.7	4.9	10	6.2	5.6	3.1	8.1	4.4	1.4	7.5	2.6	3.4	9.5	0.8	7.1	8.2	7.7	3.4	7.2	2.9	avg				
26	2022/1/24	4.2	2.4	3.4	3.6	7.9	8	6.9	2.7	3.4	9.6	8.6	4.3	6.6	0.4	2.7	1.8	9.2	3.8	0.8	4	6.6	3.6	1	6.2	avg				
27	2022/1/25	10	6.6	6.1	5.5	1.8	2.5	0.2	3.9	6.5	9.1	8.9	0.5	7.8	2.9	6.7	0.5	9.1	9.2	6.8	8.7	0.9	1.4	4.1	4.5	sum				
28	2022/1/26	4.2	3.9	6.2	7.3	5.2	6.1	2.3	4.5	9.1	4.3	3	5.6	6.5	3.7	5.5	9.3	5.3	0.6	6.1	3.5	9.6	8.2	4.1	5.7	avg				

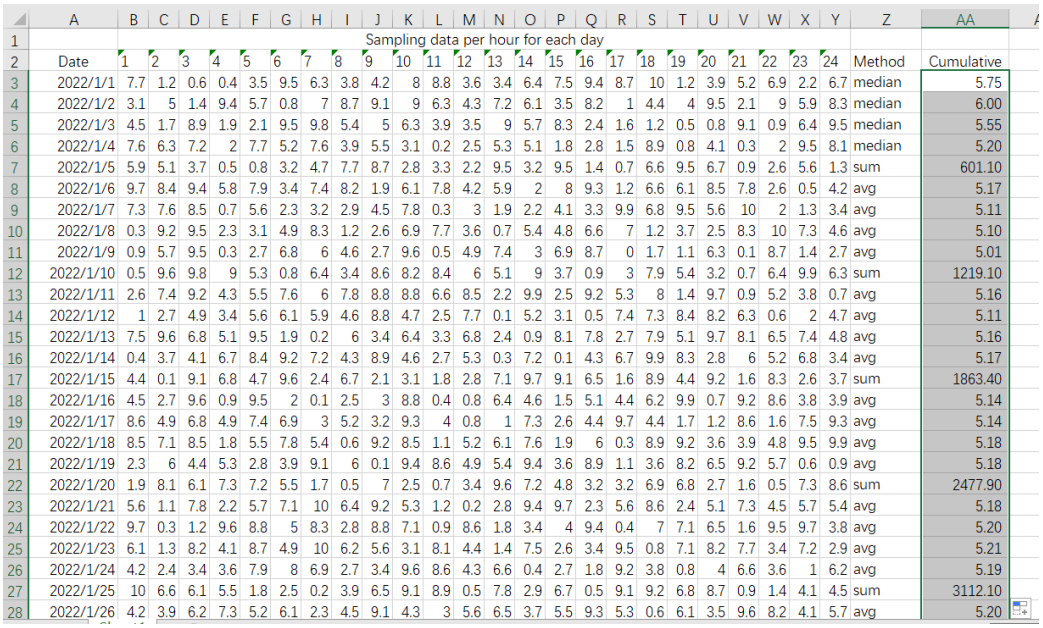
Sheet1

Now we want to add a column on the right to compute the cumulative aggregate of sampling data. The aggregate function is determined by the function name given in column Z. If it is median, compute the median; if it is avg, compute the average; if it is sum, compute the sum, and so on.

Enter the expression: =spl("=?.conj()."&Z3&"()",$B$3:Y3) in AA3, where the first parameter "=?.conj()."&Z3&"()" is a dynamically concatenated SPL expression; the second parameter $B$3:Y3 is to pass in an array consisting of values in the range of B3:Y3.

In the pass-in parameter $B$3:Y3, the first cell is added with two symbols $, the reason is that we should always keep accumulating from cell B3 when copying the expression down since what we want to calculate is the cumulative aggregate, and this cell should remain unchanged.

Drag AA3 down to every relevant row to get the cumulative aggregate of every row.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1		Sampling data per hour for each day																									
2	Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Method	Cumulative
3	2022/1/1	7.7	1.2	0.6	0.4	3.5	9.5	6.3	3.8	4.2	8	8.8	3.6	3.4	6.4	7.5	9.4	8.7	10	1.2	3.9	5.2	6.9	2.2	6.7	median	5.75
4	2022/1/2	3.1	5	1.4	9.4	5.7	0.8	7	8.7	9.1	9	6.3	4.3	7.2	6.1	3.5	8.2	1	4.4	4	9.5	2.1	9	5.9	8.3	median	6.00
5	2022/1/3	4.5	1.7	8.9	1.9	2.1	9.5	9.8	5.4	5	6.3	3.9	3.5	9	5.7	8.3	2.4	1.6	1.2	0.5	0.8	9.1	0.9	6.4	9.5	median	5.55
6	2022/1/4	7.6	6.3	7.2	2	7.7	5.2	7.6	3.9	5.5	3.1	0.2	2.5	5.3	5.1	1.8	2.8	1.5	8.9	0.8	4.1	0.3	2	9.5	8.1	median	5.20
7	2022/1/5	5.9	5.1	3.7	0.5	0.8	3.2	4.7	7.7	8.7	2.8	3.3	2.2	9.5	3.2	9.5	1.4	0.7	6.6	9.5	6.7	0.9	2.6	5.6	1.3	sum	601.10
8	2022/1/6	9.7	8.4	9.4	5.8	7.9	3.4	7.4	8.2	1.9	6.1	7.8	4.2	5.9	2	8	9.3	1.2	6.6	6.1	8.5	7.8	2.6	0.5	4.2	avg	5.17
9	2022/1/7	7.3	7.6	8.5	0.7	5.6	2.3	3.2	2.9	4.5	7.8	0.3	3	1.9	2.2	4.1	3.3	9.9	6.8	9.5	5.6	10	2	1.3	3.4	avg	5.11
10	2022/1/8	0.3	9.2	9.5	2.3	3.1	4.9	8.3	1.2	2.6	6.9	7.7	3.6	0.7	5.4	4.8	6.6	7	1.2	3.7	2.5	8.3	10	7.3	4.6	avg	5.10
11	2022/1/9	0.9	5.7	9.5	0.3	2.7	6.8	6	4.6	2.7	9.6	0.5	4.9	7.4	3	6.9	8.7	0	1.7	1.1	6.3	0.1	8.7	1.4	2.7	avg	5.01
12	2022/1/10	0.5	9.6	9.8	9	5.3	0.8	6.4	3.4	8.6	8.2	8.4	6	5.1	9	3.7	0.9	3	7.9	5.4	3.2	0.7	6.4	9.9	6.3	sum	1219.10
13	2022/1/11	2.6	7.4	9.2	4.3	5.5	7.6	6	7.8	8.8	8.8	6.6	8.5	2.2	9.9	2.5	9.2	5.3	8	1.4	9.7	0.9	5.2	3.8	0.7	avg	5.16
14	2022/1/12	1	2.7	4.9	3.4	5.6	6.1	5.9	4.6	8.8	4.7	2.5	7.7	0.1	5.2	3.1	0.5	7.4	7.3	8.4	8.2	6.3	0.6	2	4.7	avg	5.11
15	2022/1/13	7.5	9.6	6.8	5.1	9.5	1.9	0.2	6	3.4	6.4	3.3	6.8	2.4	0.9	8.1	7.8	2.7	7.9	5.1	9.7	8.1	6.5	7.4	4.8	avg	5.16
16	2022/1/14	0.4	3.7	4.1	6.7	8.4	9.2	7.2	4.3	8.9	4.6	2.7	5.3	0.3	7.2	0.1	4.3	6.7	9.9	8.3	2.8	6	5.2	6.8	3.4	avg	5.17
17	2022/1/15	4.4	0.1	9.1	6.8	4.7	9.6	2.4	6.7	2.1	3.1	1.8	2.8	7.1	9.7	9.1	6.5	1.6	8.9	4.4	9.2	1.6	8.3	2.6	3.7	sum	1863.40
18	2022/1/16	4.5	2.7	9.6	0.9	9.5	2	0.1	2.5	3	8.8	0.4	0.8	6.4	4.6	1.5	5.1	4.4	6.2	9.9	0.7	9.2	8.6	3.8	3.9	avg	5.14
19	2022/1/17	8.6	4.9	6.8	4.9	7.4	6.9	3	5.2	3.2	9.3	4	0.8	1	7.3	2.6	4.4	9.7	4.4	1.7	1.2	8.6	1.6	7.5	9.3	avg	5.14
20	2022/1/18	8.5	7.1	8.5	1.8	5.5	7.8	5.4	0.6	9.2	8.5	1.1	5.2	6.1	7.6	1.9	6	0.3	8.9	9.2	3.6	3.9	4.8	9.5	9.9	avg	5.18
21	2022/1/19	2.3	6	4.4	5.3	2.8	3.9	9.1	6	0.1	9.4	8.6	4.9	5.4	9.4	3.6	8.9	1.1	3.6	8.2	6.5	9.2	5.7	0.6	0.9	avg	5.18
22	2022/1/20	1.9	8.1	6.1	7.3	7.2	5.5	1.7	0.5	7	2.5	0.7	3.4	9.6	7.2	4.8	3.2	3.2	6.9	6.8	2.7	1.6	0.5	7.3	8.6	sum	2477.90
23	2022/1/21	5.6	1.1	7.8	2.2	5.7	7.1	10	6.4	9.2	5.3	1.2	0.2	2.8	9.4	9.7	2.3	5.6	8.6	2.4	5.1	7.3	4.5	5.7	5.4	avg	5.18
24	2022/1/22	9.7	0.3	1.2	9.6	8.8	5	8.3	2.8	8.8	7.1	0.9	8.6	1.8	3.4	4	9.4	0.4	7	7.1	6.5	1.6	9.5	9.7	3.8	avg	5.20
25	2022/1/23	6.1	1.3	8.2	4.1	8.7	4.9	10	6.2	5.6	3.1	8.1	4.4	1.4	7.5	2.6	3.4	9.5	0.8	7.1	8.2	7.7	3.4	7.2	2.9	avg	5.21
26	2022/1/24	4.2	2.4	3.4	3.6	7.9	8	6.9	2.7	3.4	9.6	8.6	4.3	6.6	0.4	2.7	1.8	9.2	3.8	0.8	4	6.6	3.6	1	6.2	avg	5.19
27	2022/1/25	10	6.6	6.1	5.5	1.8	2.5	0.2	3.9	6.5	9.1	8.9	0.5	7.8	2.9	6.7	0.5	9.1	9.2	6.8	8.7	0.9	1.4	4.1	4.5	sum	3112.10
28	2022/1/26	4.2	3.9	6.2	7.3	5.2	6.1	2.3	4.5	9.1	4.3	3	5.6	6.5	3.7	5.5	9.3	5.3	0.6	6.1	3.5	9.6	8.2	4.1	5.7	avg	5.20

2.3 **Editing SPL code**

For certain computations, the SPL expression may be very complex, even more than one line, which makes it inconvenient to edit directly in Excel. To solve this problem, esProc provides the **Excel Copy/Paste** feature to copy the edited code to Excel.

2.3.1 **Generate complex SPL script with esProc**

We have an Excel file as follows:

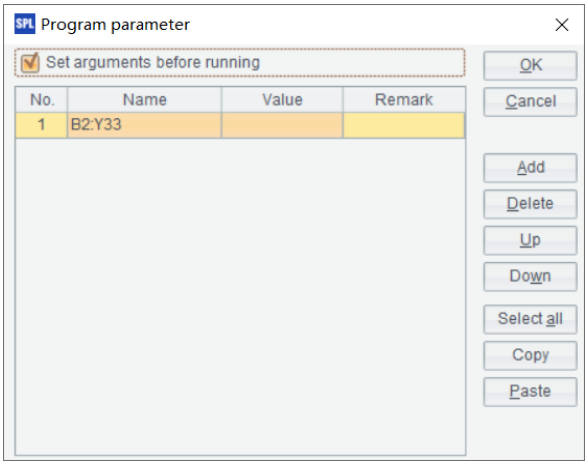
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1		Sampling data per hour for each day																							
2	Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
3	2022-01-01	9	4	8	6	1	9	1	0	2	4	6	5	9	1	2	7	3	9	6	8	5	0	0	4
4	2022-01-02	1	1	8	7	3	7	3	5	1	8	8	7	5	1	9	8	1	8	6	6	5	5	5	4
5	2022-01-03	6	5	5	3	8	4	4	9	0	9	1	1	6	5	1	1	9	0	5	3	1	0	6	8
6	2022-01-04	8	3	0	7	6	1	9	0	9	6	2	6	1	5	6	5	3	2	3	6	8	2	1	3
7	2022-01-05	3	6	6	0	3	7	8	3	4	6	6	5	1	8	4	2	1	9	2	4	1	1	1	0
8	2022-01-06	6	8	2	1	6	2	0	5	6	4	2	1	4	8	8	9	5	2	2	2	2	6	5	4
9	2022-01-07	7	7	1	2	4	8	1	9	3	6	6	8	1	4	6	9	4	8	7	6	1	1	6	1
10	2022-01-08	5	7	8	4	7	0	3	9	9	2	4	0	5	9	3	6	5	9	5	5	1	8	0	9
11	2022-01-09	3	7	0	8	4	3	0	4	3	5	7	5	8	5	4	0	5	2	9	4	6	0	5	2
12	2022-01-10	3	2	7	9	1	1	8	6	5	8	5	8	5	2	7	8	4	5	6	8	7	6	4	9
13	2022-01-11	9	7	8	9	7	8	4	1	5	7	7	0	2	8	2	1	2	9	3	4	2	9	3	1
14	2022-01-12	5	2	0	1	7	5	5	1	8	3	1	4	8	1	3	0	3	8	1	5	7	8	1	2
15	2022-01-13	1	6	6	7	7	4	6	0	2	9	8	9	6	1	4	4	2	4	5	7	9	0	3	5
16	2022-01-14	5	1	5	3	9	4	3	0	4	6	9	7	0	2	5	8	8	0	5	2	9	5	8	8
17	2022-01-15	9	9	5	6	2	5	2	2	6	0	5	3	2	6	8	6	3	0	7	6	3	7	7	5
18	2022-01-16	5	6	6	9	4	1	7	5	5	3	4	3	0	9	4	2	0	7	4	8	1	4	8	4
19	2022-01-17	0	6	2	0	5	7	7	8	4	7	0	9	5	1	1	9	2	5	6	2	0	1	9	1
20	2022-01-18	1	3	6	0	2	8	4	4	8	8	6	6	3	6	3	0	9	5	9	4	1	9	0	2
21	2022-01-19	3	3	2	9	3	0	1	4	1	8	8	9	6	4	7	5	6	6	8	3	8	4	7	3
22	2022-01-20	3	1	2	0	6	8	1	1	9	5	3	8	5	8	3	8	9	0	3	5	3	5	3	7
23	2022-01-21	2	7	8	4	6	1	5	0	7	6	7	8	9	2	3	6	9	2	8	5	8	7	3	0
24	2022-01-22	3	9	0	8	5	6	0	9	0	6	3	9	1	5	9	1	6	9	9	5	1	0	8	4
25	2022-01-23	4	6	1	1	2	1	2	2	2	2	0	1	0	8	5	9	7	8	2	0	6	8	5	5
26	2022-01-24	1	0	1	7	3	7	4	7	6	7	0	6	9	6	6	6	6	8	0	1	4	1	8	3
27	2022-01-25	0	8	4	0	2	9	0	7	9	7	6	4	6	5	5	4	8	4	8	3	3	1	5	7
28	2022-01-26	9	5	0	1	3	0	0	6	3	4	4	1	2	7	6	4	9	1	8	8	3	0	6	9
29	2022-01-27	8	0	4	6	1	0	2	8	8	4	5	7	7	6	5	7	5	5	1	1	4	8	1	5
30	2022-01-28	2	1	3	0	2	3	0	4	7	3	5	5	3	5	4	9	4	7	5	1	4	8	9	1
31	2022-01-29	6	9	0	4	6	5	0	0	2	1	6	7	8	8	1	2	6	1	0	0	2	2	6	6
32	2022-01-30	3	4	2	2	9	1	9	8	2	4	0	6	8	5	7	4	5	7	1	1	7	3	7	3
33	2022-01-31	0	2	6	7	6	9	0	2	7	8	9	0	4	7	7	9	7	5	1	3	9	1	4	2
34																									

Now we want to add a row at the bottom to calculate the average of the sampling data for each hour in January. We would write one expression that can calculate the average of every cell in the new row without copying the expression to each cell.

The calculation result is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1		Sampling data per hour for each day																							
2	Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
3	2022/1/1	1	8	6	7	6	0	8	1	0	4	4	3	1	4	8	4	3	5	9	5	6	3	7	0
4	2022/1/2	2	8	6	5	8	0	1	0	0	5	0	6	4	1	9	1	0	6	2	2	8	1	7	8
5	2022/1/3	0	9	8	4	7	0	1	3	8	7	8	8	4	3	7	2	9	1	8	6	4	5	2	7
6	2022/1/4	5	7	6	8	8	3	0	1	1	3	8	7	5	6	3	1	6	2	8	7	9	5	1	0
7	2022/1/5	1	5	6	3	5	5	6	3	4	1	1	6	1	1	1	2	0	1	2	5	9	8	0	8
8	2022/1/6	4	5	7	3	7	2	3	8	2	7	8	8	7	8	8	4	6	2	6	1	9	4	7	3
9	2022/1/7	2	3	4	1	0	6	7	9	8	9	3	6	6	0	9	7	2	0	1	7	5	0	3	9
10	2022/1/8	4	9	4	5	9	5	5	1	3	1	2	5	8	4	6	6	5	4	6	0	1	7	0	7
11	2022/1/9	9	5	3	5	7	0	8	6	1	8	4	9	6	1	9	7	1	3	3	1	1	9	0	0
12	2022/1/10	7	7	9	8	3	0	2	7	3	0	3	3	9	4	9	1	6	7	0	3	4	2	2	7
13	2022/1/11	6	2	5	2	7	3	0	9	4	3	3	3	4	2	7	4	8	0	5	9	9	7	2	1
14	2022/1/12	7	3	2	3	3	4	5	2	1	6	8	5	7	6	6	5	1	4	1	8	9	8	9	9
15	2022/1/13	4	6	8	6	4	0	0	2	0	2	6	3	0	6	1	3	7	8	3	4	0	8	8	1
16	2022/1/14	9	5	1	9	8	0	6	7	2	1	0	5	6	7	2	1	5	5	4	3	7	5	3	7
17	2022/1/15	5	6	5	3	2	6	6	7	2	4	7	1	5	4	6	7	2	0	7	4	9	3	2	2
18	2022/1/16	2	4	1	5	9	7	7	7	6	5	6	2	1	0	2	7	8	6	2	8	1	2	5	5
19	2022/1/17	2	2	3	8	6	7	6	1	5	0	4	3	7	2	1	2	9	7	5	7	4	5	9	0
20	2022/1/18	3	3	1	9	4	3	7	9	7	6	3	5	0	2	8	7	3	8	9	7	7	2	4	1
21	2022/1/19	8	0	8	8	6	3	6	7	7	8	4	5	8	3	7	3	9	5	0	8	0	1	6	0
22	2022/1/20	4	3	9	5	1	1	0	6	2	0	3	6	4	4	3	2	0	1	6	1	3	7	9	6
23	2022/1/21	8	9	1	8	2	9	0	6	2	2	4	3	1	5	0	8	2	1	4	4	4	9	6	3
24	2022/1/22	2	5	4	0	3	7	8	7	7	9	6	6	7	4	7	1	4	4	1	2	5	4	8	4
25	2022/1/23	2	9	7	1	3	0	0	9	6	5	6	2	5	5	5	9	8	6	5	0	6	3	5	7
26	2022/1/24	7	5	1	1	2	7	1	2	7	1	6	8	4	1	8	0	6	6	4	6	9	4	6	7
27	2022/1/25	0	1	2	8	8	7	4	7	1	6	5	3	0	9	1	0	6	6	0	5	9	0	2	8
28	2022/1/26	8	7	4	0	4	1	8	5	1	4	0	8	8	8	2	9	0	5	8	3	1	7	4	8
29	2022/1/27	5	5	4	3	0	0	6	3	6	3	9	9	5	7	3	0	8	4	8	7	1	5	9	7
30	2022/1/28	3	8	8	3	5	0	8	9	9	9	7	2	0	5	4	2	4	7	8	7	3	7	2	0
31	2022/1/29	1	6	8	9	8	3	3	2	1	1	7	1	1	5	4	4	5	4	7	1	9	1	3	6
32	2022/1/30	3	8	6	4	1	4	7	0	1	7	1	0	7	2	9	6	3	8	9	3	2	1	4	7
33	2022/1/31	7	7	9	8	9	9	0	8	7	6	6	5	9	3	8	1	3	8	4	6	9	4	5	8
34	Average	4.23	5.48	5.03	4.9	5	3.29	4.16	4.97	3.68	4.29	4.58	4.71	4.52	3.94	5.26	3.74	4.48	4.32	4.68	4.52	5.26	4.42	4.52	4.71
35																									

In this example, it needs to take B2:Y33 as the parameter. Open the esProc, and define the parameter as follows:



Note: The parameter name here must be **B2:Y33**, which is the same as the parameter passed from Excel.

First, edit the script:

	A
1	=E('B2:Y33')
2	=to(24).("avg("'+string(~)+"'):"'+string(~)+"'")
3	=A1.groups(;${A2.concat@c()})
4	return A3(1).array()

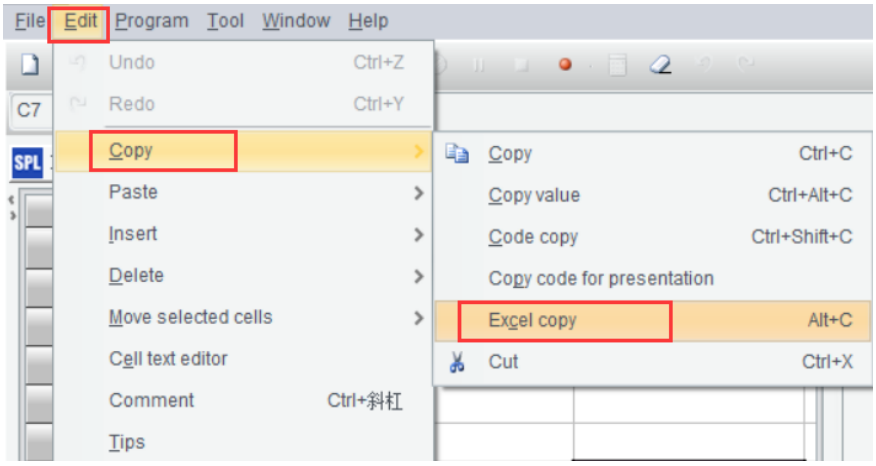
A1: Convert the passed-in two-dimensional array parameter **'B2:Y33'** to a table sequence, with the first row as column name; The parameter name needs to be enclosed in single quotation marks to distinguish it from the cell of esProc.

A2: Dynamically piece together an aggregate expression for 24 columns.

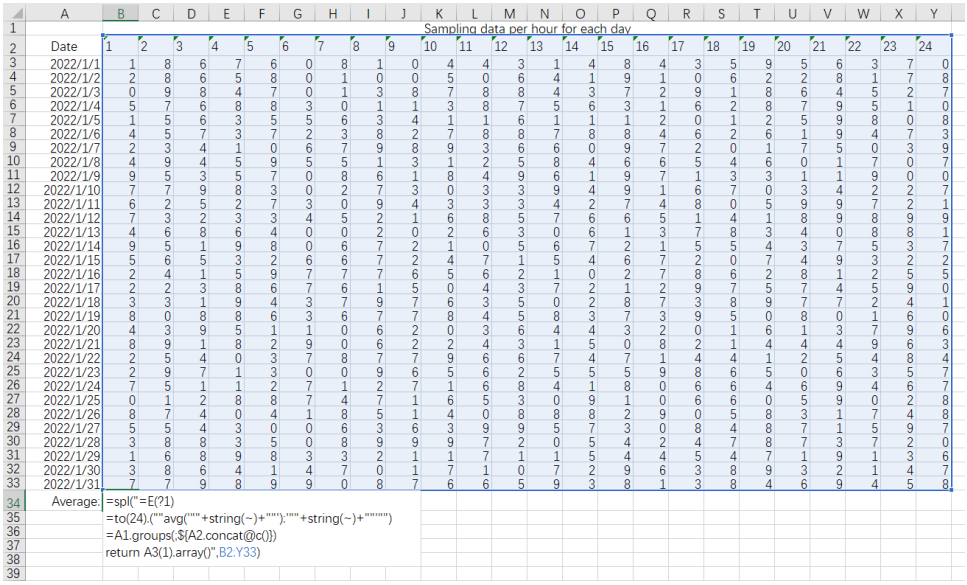
A3: Concatenate the aggregate expression into the groups function to calculate the average of every column.

A4: Convert the result to a single-level sequence and return.

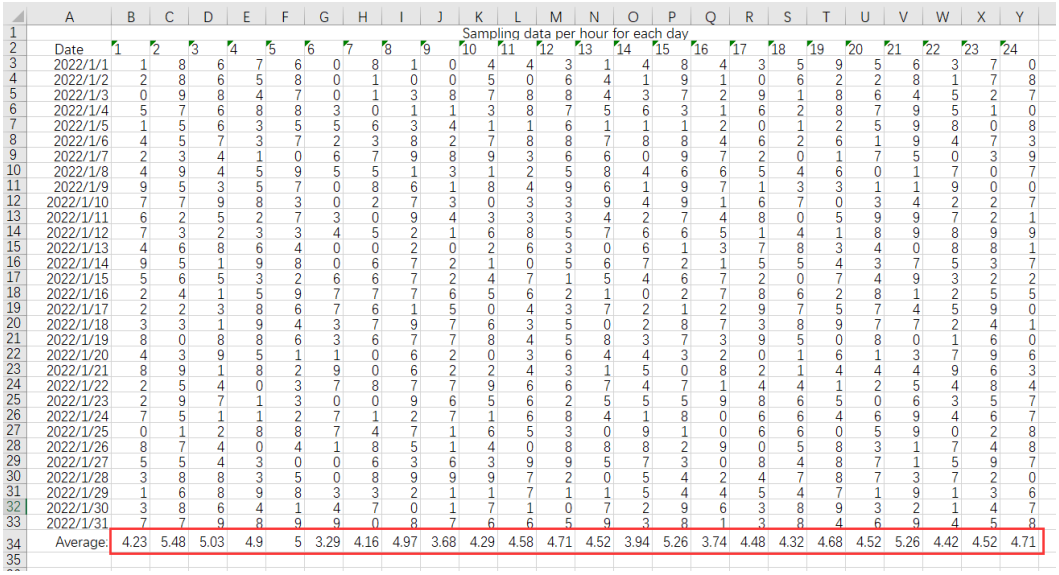
Then, Click Edit > Copy > Excel Copy, at this point, this code can be copied to the clipboard.



Now, go back to the Excel file, select cell B34, and press Ctrl-V to paste the code in, as shown below:



This code will return a one-dimensional array, press Ctrl-Enter to fill the returned array values rightward into every cell of the row in sequence. The final result is as follows:



Notes:

1. When the returned result is a one-dimensional array, the result set will be filled rightward into the cells in sequence after pressing Ctrl-Enter.

2. If the parameter passed to SPL script is a range of cells (a single row, a single column, multi-row and multi-column), it will automatically form a two-level sequence. In this case, it can be converted to a table sequence in the SPL script using **E()** function;

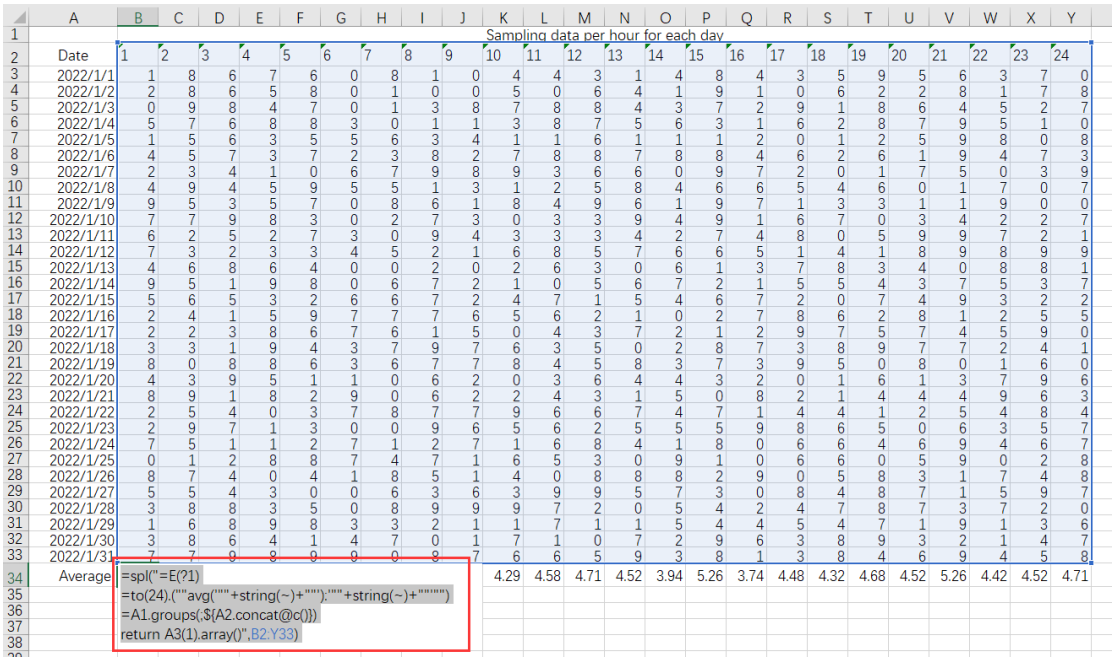
3. The rule for parameter is the same as that for a one-line code. That is, use ?1 to represent the first parameter, ?2 to represent the second, and so on.

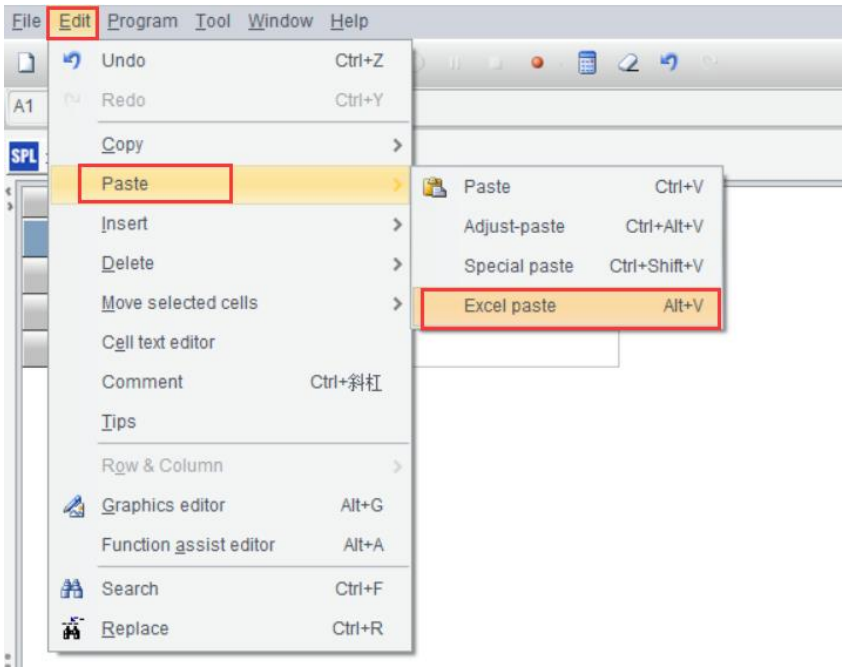
4. In the SPL script, the Excel cell (or a range of cells), as parameter, is enclosed in single quotation marks.

2.3.2 **Back to esProc to modify script**

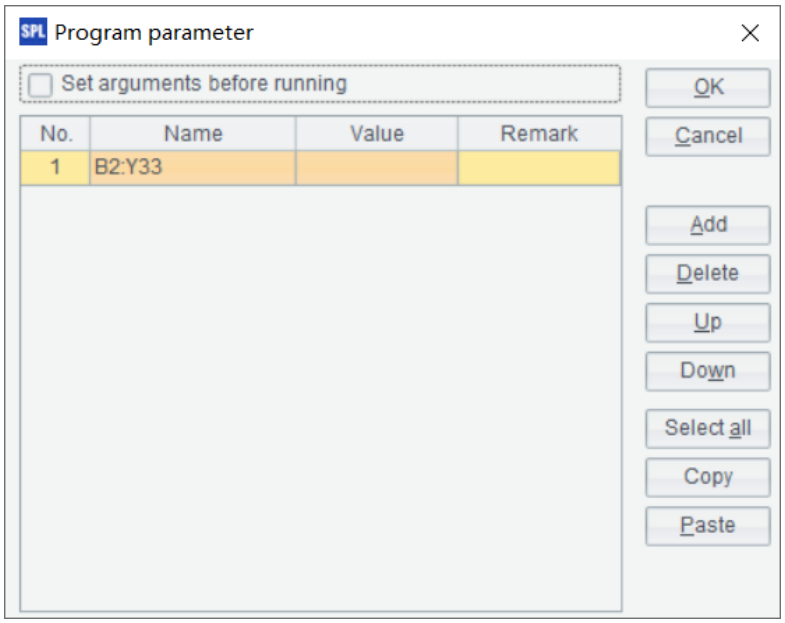
In the above example, after the code is copied to Excel, we might close the esProc and didn't save the code. In this case, if the code needs to be modified, it is inconvenient to modify it directly in Excel. To solve this problem, we can use the Excel Paste feature of esProc to copy SPL script back to esProc for modifying. The specific method is as follows:

Select the whole SPL code in Excel, and press Ctrl-C to copy it to the clipboard. Then open esProc and click Edit > Paste > Excel Paste, at his point, the original code can be restored to esProc, as shown below:





	A
1	=E('B2:Y33')
2	=to(24).("avg('"+string(~)+"'):'"+string(~)+"'")
3	=A1.groups(;${A2.concat@c()})
4	return A3(1).array()

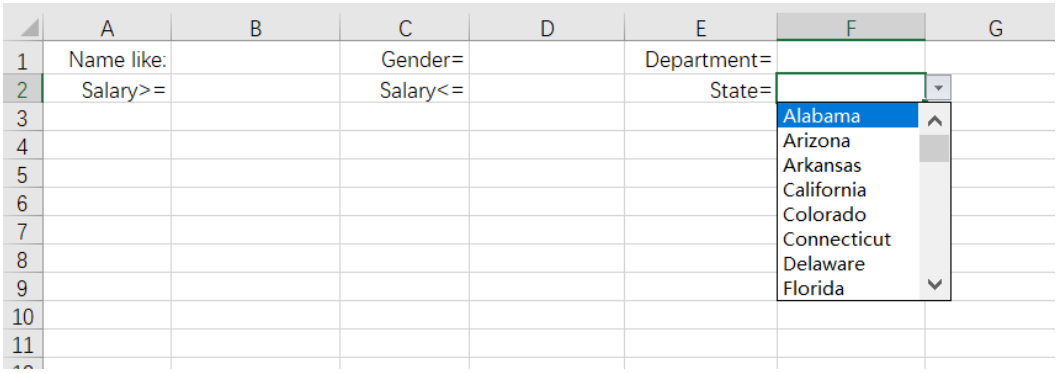


2.3.3 **Extremely long SPL script**

We have an Employee.xlsx file as follows (the file must be stored under esProc main path):

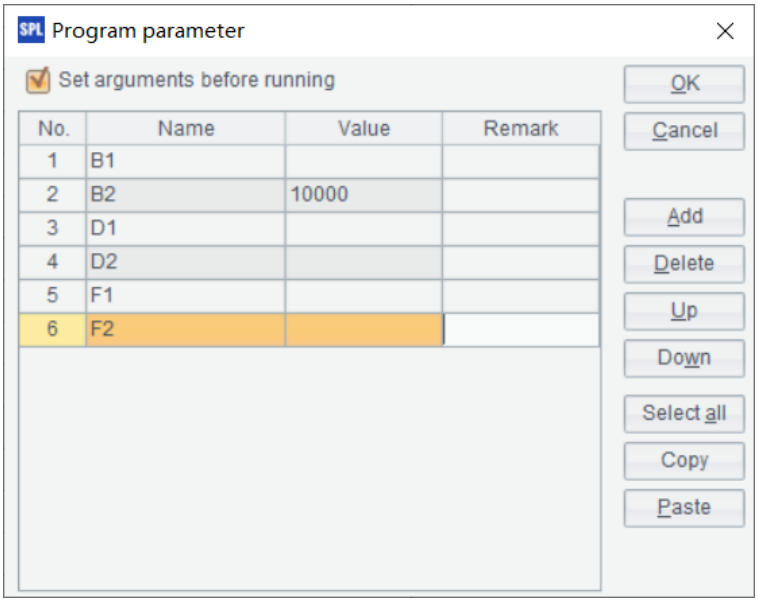
	A	B	C	D	E	F	G	H	I
1	EID	NAME	SURNAME	GENDER	DEPT	SALARY	BIRTHDAY	HIREDATE	STATE
2	1	Rebecca	Moore	F	R&D	7000	1974/11/20	2005/03/11	California
3	2	Ashley	Wilson	F	Finance	11000	1980/07/19	2008/03/16	New York
4	3	Rachel	Johnson	F	Sales	9000	1970/12/17	2010/12/01	New Mexico
5	4	Emily	Smith	F	HR	7000	1985/03/07	2006/08/15	Texas
6	5	Ashley	Smith	F	R&D	16000	1975/05/13	2004/07/30	Texas
7	6	Matthew	Johnson	M	Sales	11000	1984/07/07	2005/07/07	California
8	7	Alexis	Smith	F	Sales	9000	1972/08/16	2002/08/16	Illinois
9	8	Megan	Wilson	F	Marketing	11000	1979/04/19	1984/04/19	California
10	9	Victoria	Davis	F	HR	3000	1983/12/07	2009/12/07	Texas
11	10	Ryan	Johnson	M	R&D	13000	1976/03/12	2006/03/12	Pennsylvania
12	11	Jacob	Moore	M	Sales	12000	1974/12/16	2004/12/16	Texas
13	12	Jessica	Davis	F	Sales	7000	1980/09/11	2008/09/11	New York
14	13	Daniel	Davis	M	Finance	10000	1982/05/14	2010/05/14	Florida
15	14	Alyssa	Wilson	F	Sales	4000	1977/12/24	2005/12/24	Florida
16	15	Alexis	Smith	F	Sales	8000	1983/07/10	2006/07/10	New York
17	16	Christopher	Hernandez	M	Production	9000	1979/06/27	2007/06/27	Florida
18	17	Hannah	Johnson	F	Marketing	4000	1980/07/19	2006/07/19	Texas
19	18	Jonathan	Moore	M	Administratior	7000	1971/03/07	2000/03/07	Florida
20	19	Samantha	Williams	F	Production	10000	1974/03/29	2004/03/29	Pennsylvania
21	20	Alexis	Allen	F	Administratior	16000	1977/08/07	2007/08/07	Florida
22	21	Jacob	Moore	M	Marketing	10000	1975/09/17	2005/09/17	Pennsylvania
23	22	Jacob	Davis	M	R&D	16000	1985/05/07	2001/05/07	Texas
24	23	Joseph	Turner	M	Finance	6000	1983/08/27	2003/08/27	California
25	24	Chloe	Smith	F	Finance	10000	1978/08/20	2008/08/20	Florida

We want to query the file to display the following columns only in the query result: EID, NAME, GENDER, DEPT, SALARY, BIRTHDAY, HIREDATE and STATE, and the display of column SURNAME is not required. The query condition is shown as below:

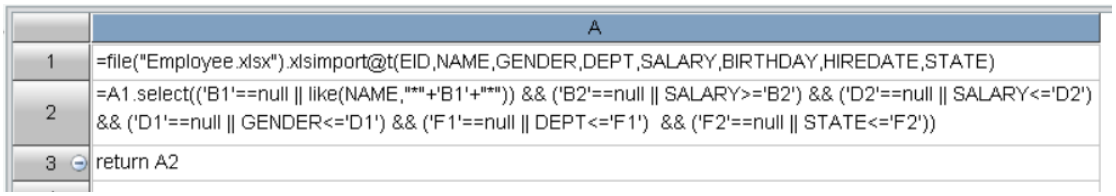


In this example, the parameters to be passed to SPL script include **B1, B2, D1, D2, F1 and F2**.

First, Open the esProc, and define the following parameters:

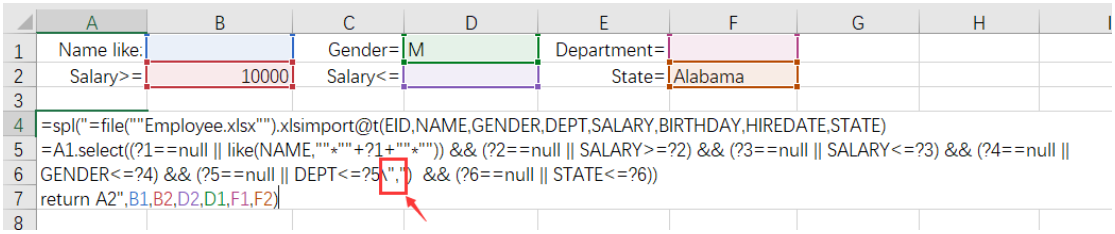


Then write the script as follows:



Next, click Edit > Copy > Excel Copy to copy the code to clipboard.

Now open the Excel file, select cell A4, and press Ctrl-V to paste the code in, as shown below:



	A	B	C	D	E	F	G	H
1	Name like:		Gender=	M	Department=			
2	Salary>=	10000	Salary<=		State=	Alabama		
3								

```
=spl(""=file(""Employee.xlsx"").xlsimport@t(EID,NAME,GENDER,DEPT,SALARY,BIRTHDAY,HIREDATE,STATE)
=A1.select((?1==null || like(NAME,""*""+?1+""*"")) && (?2==null || SALARY>=?2) && (?3==null || SALARY<=?3) && (?4==null ||
GENDER<=?4) && (?5==null || DEPT<=?5\","")  && (?6==null || STATE<=?6))
return A2",B1,B2,D2,D1,F1,F2)
```

Due to the restriction on the length of string in Excel expression, extremely-long code will be automatically split into multiple parts when it is copied to Excel, with each part not longer than 240 characters. If a part of code ends with a slash \, it indicates that the code has not ended, and the next parameter is still the code.

Finally, press Ctrl-Enter to get the final result:

	A	B	C	D	E	F	G	H
1	Name like:		Gender=	M	Department=			
2	Salary>=	10000	Salary<=		State=	Alabama		
3								
4	EID	NAME	GENDER	DEPT	SALARY	BIRTHDAY	HIREDATE	STATE
5	102	Christian	M	Sales	12000	1972/7/25	2000/10/1	Alabama
6								

2.3.4 **Invoke SPL script directly**

For longer SPL code, you can write the SPL script to a file and invoke it directly in the spl() function.

For example, in the above example, we save the script file as Example15.splx under esProc main path, and then write the following expression in cell A4 of Excel:

	A	B	C	D	E	F
1	Name like:		Gender=	M	Department=	
2	Salary>=	10000	Salary<=		State=	Alabama
3						
4	=spl("Example15(?1,?2,?3,?4,?5,?6)",B1,B2,D1,D2,F1,F2)					
5						

As you can see from the expression that the syntax for directly invoking the script file is:

=spl("script file name (?n……)",parameter n….)

The number of question marks to be written in parentheses depends on that of parameters in the script. The corresponding relationship rule with passed-in parameters is the same as that for directly writing code, that is, ?1 represents the first parameter, ?2 represents the second one, and so on.

Now press Ctrl-Enter to get the final result:

	A	B	C	D	E	F	G	H
1	Name like:		Gender=	M	Department=			
2	Salary>=	10000	Salary<=		State=	Alabama		
3								
4	EID	NAME	GENDER	DEPT	SALARY	BIRTHDAY	HIREDATE	STATE
5	102	Christian	M	Sales	12000	1972/7/25	2000/10/1	Alabama
6								

Chapter 3 **Using the clipboard**

3.1 **Basic usage**

esProc provides the clipboard() function, which can exchange data with the clipboard.

Let's take the example of “finding the top 3 students in each subject”.

The following is the source data in Excel, where column A contains the student names, and column B-D contain the scores in math, English and physics respectively.

	A	B	C	D
1	name	math	english	physics
2	lily	97	100	99
3	Joshua	100	99	100
4	Sarah	98	99	96
5	Bertram	94	95	85
6	Paula	91	88	91
7	Sophia	92	81	76
8	Ben	87	80	76
9	Ruth	92	91	87
10	Pag	95	87	87

Calculation objective: find the top 3 students in score in each subject, and append their names to the end of score column of the corresponding subject.

To achieve the objective, we need to use some features such as the row set TopN, join by sequence number. However, it’s hard to solve in Excel, we use SPL to make it easy.

Operation steps: i) select the source data area (A1:D10) in Excel; ii) press Ctrl+C to copy it to system clipboard; iii) open the esProc to write and execute the following script:

	A
1	=clipboard().import@t()
2	=A1.top(-3;math).(name)
3	=A1.top(-3;english).(name)
4	=A1.top(-3;physics).(name)
5	=join@p(A2;A3;A4).export()
6	=clipboard(A5)

A1: Read the data from clipboard

A2: Get the names of top 3 students in math

A5: Join the names of these students to form a two-dimensional table and convert it to a string

A6: Write the string to clipboard

In this code, the clipboard() function is used in two ways. When it is called without parameter, the string in the clipboard will be taken out, such as the cell A1; When it is called using a variable or cell name as the parameter, the string will be written to the clipboard, such as the clipboard(…) in A6.

After executing the above script, select the cell B11 in Excel, and press Ctrl+V to copy the data from clipboard to B11-D13, as shown below:

	A	B	C	D
	…	…	…	…
10	Pag	95	87	87
11		Joshua	Lily	Joshua
12		Sarah	Sarah	Lily
13		lily	Joshua	Sarah

3.2 **Edit the script at will**

When editing and debugging the script, it is very likely to use the copy and paste functions, which will overwrite the content in the clipboard. Consequently, the returned content will be the last copied code rather than the source data in Excel while executing the clipboard() the next time. In this case, the calculation will fail, and you have to go back to Excel to re-copy the data, which will give you some trouble.

To solve this problem, SPL provides the clipboard@e() function, where the option @e means that it will always return the data copied from Excel for the first time to the clipboard. Let's have a try:

In the example in the previous section, suppose that we have copied the data from Excel and written the script. At this point, if we edit the script, and cut A3 and A4 and paste them into B2 and B3, A2 will report an error when the script is executed again, because the clipbaord() in A1 gets wrong data. To avoid this problem, we need to modify the code, and use the just mentioned clipboard@e() in A1. By doing so, the script will still be executed normally after moving the code. The edited code is as follows:

	A	B	C
1	=clipboard@e().import@t()		
2	=A1.top(-3;math).(name)	=A1.top(-3;english).(name)	=A1.top(-3;physics).(name)
3	=join@p(A2;B2;B3).export( )		
4	=clipboard(A3)		

A1: Read data from clipboard

A3: Join the names of these students to form a two-dimensional table and convert it to a string

A4: Write the string to clipboard

3.3 **Multiple result data areas**

The clipboard(...) can only return one result, but some complex operations may need to return multiple results, what should we do?

We can directly copy multiple cell values (or variable values) in the result display area of esProc to the clipboard separately, and return them to Excel in turn.

Let's take the example of "finding the top 3 students in each subject and the goal of each person to surpass".

Calculation objective: 1) append the names of the top 3 students in each subject to the end of corresponding subject based on the student score table; 2) add a new column "target" to calculate out three students for everyone, and each of the three students has a total score higher than and is close to the one to be calculated, as his/her target to surpass. Note that not all students have 3 targets, and the target students should be concatenated by the greater than sign (>).

Operation steps: i) select the source data area (A1:D10) in Excel; ii) press Ctrl+C to copy to system's clipboard; iii) open the esProc to write and execute the following script:

	A	B	C
1	=clipboard@e().import@t()		
2	=A1.top(-3;math).(name)	=A1.top(-3;english).(name)	=A1.top(-3;physics).(name)
3	=join@p(A2;B2;C2)		
4	=A1.derive(sum(math,english,physics):subtotal)		
5	=A4.derive(t=subtotal,A4.select(subtotal>t):beforeMe)		
6	=A5.new(beforeMe.top(3;subtotal).(name).concat(">"):target)		

A1: Get the data from clipboard

A3: Data area 1: top 3 students in each subject

A4: Total score of each student

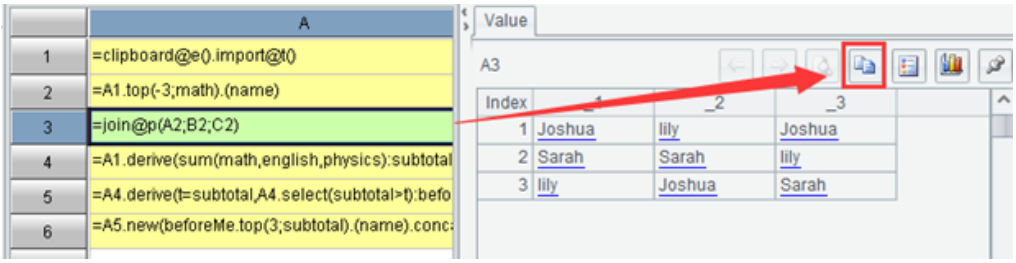
A5: 3 students whose total score is higher than the one to be calculated

A6: Data area 2: 3 students whose total score is close to the one to be calculated

A3 stores the result data area 1, that is, the top 3 students in score in each subject; A4 stores the result data area 2, i.e., the targets of each student to surpass. It should also be noted that export () and clipboard (...) are not needed in the code.

Now we return the results to Excel.

Click on the data area 1, and click the corresponding "copy data" button on the right, as shown below:



Then, select the cell B11 in Excel, and press Ctrl+V to copy the data area 1 to B11-D13, as shown as below:

	A	B	C	D
1	name	math	english	physics
2	Lily	97	100	99
3	Joshua	100	99	100
4	Sarah	98	99	96
5	Bertram	94	95	85
6	Paula	91	88	91
7	Sophia	92	81	76
8	Ben	87	80	76
9	Ruth	92	91	87
10	Pag	95	87	87
11		Joshua	lily	Joshua
12		Sarah	Sarah	lily
13		lily	Joshua	Sarah

Next, click on the data area 2 in the script, and hold down the Shift key and click the corresponding "copy data" button on the right. After that, select the cell E1 in Excel, and press Ctrl+V to paste the data area 2 together with the column name to E1-E10. The result is as follows:

	A	B	C	D	E
1	name	math	english	physics	Target
2	Lily	97	100	99	Joshua
3	Joshua	100	99	100	
4	Sarah	98	99	96	lily>Joshua
5	Bertram	94	95	85	Sarah>lily>Joshua
6	Paula	91	88	91	Bertram>Sarah>lily
7	Sophia	92	81	76	Pag>Ruth>Paula
8	Ben	87	80	76	Sophia>Pag>Ruth
9	Ruth	92	91	87	Bertram>Sarah>lily
10	Pag	95	87	87	Ruth>Paula>Bertram
11		Joshua	lily	Joshua	
12		Sarah	Sarah	lily	
13		lily	Joshua	Sarah	

We can use Shift key to control whether the calculation result has a column name.

3.4 **Multiple source data areas**

Having solved the multiple-target problem, it is easy to think of the multiple-source problem. In practice, the calculation may need to use multiple source data areas in Excel. However, the clipboard() function only holds the most-recently copied data area. How to solve this problem?

We can copy the clipboard content directly to a cell in the cellset.

The following is an example of "querying the orders that meet the specified conditions".

There are two sheets in Excel, order details and employee list, where the order details table is as follows:

	A	B	C	D	E
1	OrderID	Client	SellerId	Amount	OrderDate
2	1	WVF Vip	1	440	2014-11-03
3	2	UFS Com	1	1863	1/1/2015
4	3	SWFR	2	1813	11/1/2014
5	4	JFS Pep	2	671	1/1/2015
6	5	DSG	1	3730	1/1/2015
7	6	JFE	1	1445	1/1/2015
8	7	OLF	3	625	1/1/2015
9	8	PAER	3	2490	1/1/2015

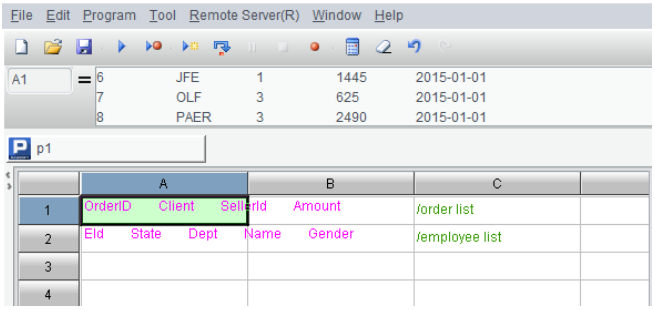
The employee list is as follows:

	A	B	C	D	E	F	G
1	Eid	State	Dept	Name	Gender	Salary	Birthday
2	2	New York	Marketing	Ashley	F	11001	7/19/1980
3	3	New Mexico	Sales	Rachel	F	9000	12/17/1970
4	4	Texas	HR	Emily	F	7000	3/7/1985
5	5	Texas	R&D	Ashley	F	16000	5/13/1975
6	6	California	Sales	Matthew	M	11000	7/7/1984
7	7	Illinois	Sales	Alexis	F	9000	8/16/1972
8	8	California	Marketing	Megan	F	11000	4/19/1979
9	1	Texas	HR	Victoria	F	3000	12/7/1983

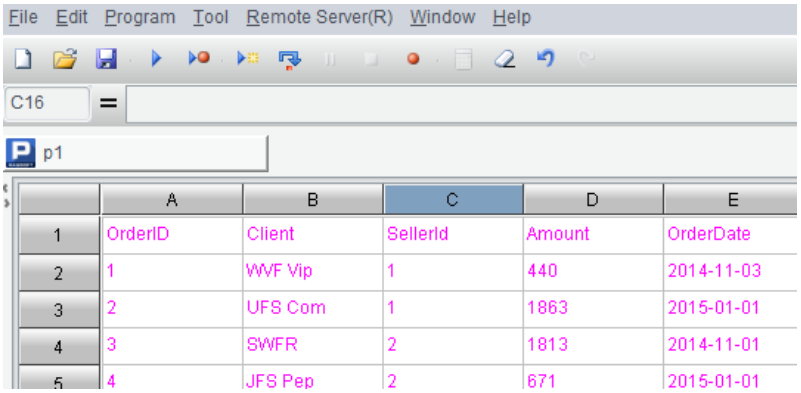
Calculation objective: query the order data in the last *days*, or the order data belonging to the department list *depts*. The required columns include the OrderID, OrderDate and Amount in the order details, and the Name and Dept in the employee list. In these conditions, *days* is an external parameter, and you can enter a different value for each execution, for example, entering 30 means

querying the order data in the last 30 days; *depts* is also an external parameter, such as ["Markeding","Finance"]. This calculation objective involves dynamic query and multi-key-value query. Such queries are hard to be achieved in Excel, but is easy to get it done with SPL.

First, select the data area of "Order Details" sheet in Excel, and copy and paste it to the cell A1 in SPL cellset together with the column names; Likewise, copy the data area of "Employee list" sheet to cell A2 in SPL cellset. See the figure below:



Note: the paste action should be performed when the cell A1 is in the editing state, otherwise the content of the clipboard will be filled into a large range of cells, as shown below:



This figure shows a wrong result. Since it occupies a too large area, the code layout will be affected.

If the operation is correct, only a small part of data is displayed on the appearance of cell A1/A2, and the complete data will be presented only when clicking A1/A2.This is the unique feature of cell-style code, which is suitable for copying a large range of data without affecting reading and code layout.

After copying the two source data areas, edit the script, during which we can run the script many times and perform the copy and paste actions manually. The final code is as follows:

	A	B
1	OrderID Client…	=A1.import@t()
2	EId State…	=A2.import@t()
3	=B1.switch(SellerId,B2:EId)	
4	=A3.select(OrderDate>=after(date(now()),days*-1)|| depts.pos(SellerId.Dept))	
5	=A4.new(OrderID,OrderDate,Amount,SellerId.Name:Name,SellerId.Dept:Dept)	

After the calculation is finished, we can use the "copy data" button to copy the calculation result in A5 back to Excel. It should be noted that the data in the cell A1/A2 should be cleared when saving the code, otherwise all source data will be saved.

More examples

Chapter 4 **Merge Excel files**

In daily work, we often need to merge the data of multiple Excel files together for convenience of performing various statistical analysis.

4.1 **Merge by row - same name and number of columns**

The simplest and most common operation is to merge several files with the same name, number and order of columns by the row. For example:

Before merging:

Fruits.xlsx

	A	B
1	Name	UnitPrice
2	Apple	1.69
3	Banana	0.69
4	Peach	0.88
5	Strawberry	1.97

and

Meats.xlsx

	A	B
1	Name	UnitPrice
2	Mutton	7.69
3	Pork	4.58
4	Chicken	5.77
5	Duck	6.89
6	Beef	7.96

After merging:

	A	B
1	Name	UnitPrice
2	Apple	1.69
3	Banana	0.69
4	Peach	0.88
5	Strawberry	1.97
6	Mutton	7.69
7	Pork	4.58
8	Chicken	5.77
9	Duck	6.89
10	Beef	7.96

Script:

	A
1	=file("Fruits.xlsx").xlsimport@t()
2	=file("Meats.xlsx").xlsimport@t()
3	=A1|A2
4	=file("Foods.xlsx").xlsexport@t(A3)

4.2 **Merge by column - same name and number of rows**

We often need to merge some Excel files with the same number and name of rows by the column. For example:

Before merging:

Fruits.xlsx

	A	B
1	Name	UnitPrice
2	Apple	1.69
3	Banana	0.69
4	Peach	0.88
5	Strawberry	1.97

and

FruitStock.xlsx

	A	B	C
1	Name	Stock	MaximumStock
2	Apple	1000	3000
3	Banana	800	1500
4	Peach	2000	2000
5	Strawberry	1500	1500

After merging:

	A	B	C	D
1	Name	UnitPrice	Stock	MaximumStock
2	Apple	1.69	1000	3000
3	Banana	0.69	800	1500
4	Peach	0.88	2000	2000
5	Strawberry	1.97	1500	1500

Script:

	A
1	=file("Fruits.xlsx").xlsimport@t()
2	=file("FruitStock.xlsx").xlsimport@t()
3	=A1.new(Name,UnitPrice,A2(#).Stock,A2(#).MaximumStock)
4	=file("FruitsPriceStock.xlsx").xlsexport@t(A3)

4.3 **Merge by row - different name and number of columns - keep all columns**

Before merging:

FruitsPriceStock.xlsx

	A	B	C	D
1	Name	UnitPrice	Stock	MaximumStock
2	Apple	1.69	1000	3000
3	Banana	0.69	800	1500
4	Peach	0.88	2000	2000
5	Strawberry	1.97	1500	1500

MeatsPriceStock.xlsx

	A	B	C	D
1	Name	Stock	MinimumStock	UnitPrice
2	Beef	3600	2000	7.96
3	Pork			4.58
4	Mutton	5000	3000	7.69
5	Duck	2500	2000	
6	Chicken	4000	3000	5.77

After merging:

	A	B	C	D	E
1	Name	UnitPrice	Stock	MaximumStock	MinimumStock
2	Apple	1.69	1000	3000	
3	Banana	0.69	800	1500	
4	Peach	0.88	2000	2000	
5	Strawberr	1.97	1500	1500	
6	Beef	7.96	3600		2000
7	Pork	4.58			
8	Mutton	7.69	5000		3000
9	Duck		2500		2000
10	Chicken	5.77	4000		3000

Script:

	A
1	=file("FruitsPriceStock.xlsx").xlsimport@t()
2	=file("MeatsPriceStock.xlsx").xlsimport@t()
3	=create(${(A1.fname()&A2.fname()).concat@c()})
4	=A3.insert@f(0:A1)
	=A3.insert@f(0:A2)
	=file("FoodsPriceStock.xlsx").xlsexport@t(A3)

A3: All columns need to be kept, so use the union of column names

4.4 Merge by row - different name and number of columns - keep only duplicate columns

Before merging:

FruitsPriceStock.xlsx

	A	B	C	D
1	Name	UnitPrice	Stock	MaximumStock
2	Apple	1.69	1000	3000
3	Banana	0.69	800	1500
4	Peach	0.88	2000	2000
5	Strawberry	1.97	1500	1500

MeatsPriceStock.xlsx

	A	B	C	D
1	Name	Stock	MinimumStock	UnitPrice
2	Beef	3600	2000	7.96
3	Pork			4.58
4	Mutton	5000	3000	7.69
5	Duck	2500	2000	
6	Chicken	4000	3000	5.77

After merging:

	A	B	C
1	Name	UnitPrice	Stock
2	Apple	1.69	1000
3	Banana	0.69	800
4	Peach	0.88	2000
5	Strawberry	1.97	1500
6	Beef	7.96	3600
7	Pork	4.58	
8	Mutton	7.69	5000
9	Duck		2500
10	Chicken	5.77	4000

Script:

	A
1	=file("FruitsPriceStock.xlsx").xlsimport@t()
2	=file("MeatsPriceStock.xlsx").xlsimport@t()
3	=create(${(A1.fname()^A2.fname()).concat@c()})
4	=A3.insert@f(0:A1)
	=A3.insert@f(0:A2)
	=file("FoodsPriceStock.xlsx").xlsexport@t(A3)

A3: Only duplicate columns need to be kept, so use the intersection of column names

4.5 Merge by row - different name and number of columns - keep only columns of the first file

Before merging:

FruitsPriceStock.xlsx

	A	B	C	D
1	Name	UnitPrice	Stock	MaximumStock
2	Apple	1.69	1000	3000
3	Banana	0.69	800	1500
4	Peach	0.88	2000	2000
5	Strawberry	1.97	1500	1500

MeatsPriceStock.xlsx

	A	B	C	D
1	Name	Stock	MinimumStock	UnitPrice
2	Beef	3600	2000	7.96
3	Pork			4.58
4	Mutton	5000	3000	7.69
5	Duck	2500	2000	
6	Chicken	4000	3000	5.77

After merging:

	A	B	C	D
1	Name	UnitPrice	Stock	MaximumStock
2	Apple	1.69	1000	3000
3	Banana	0.69	800	1500
4	Peach	0.88	2000	2000
5	Strawberry	1.97	1500	1500
6	Beef	7.96	3600	
7	Pork	4.58		
8	Mutton	7.69	5000	
9	Duck		2500	
10	Chicken	5.77	4000	

Script:

	A
1	=file("FruitsPriceStock.xlsx").xlsimport@t()
2	=file("MeatsPriceStock.xlsx").xlsimport@t()
3	=A1.insert**@f**(0:A2)
4	=file("FoodsPriceStock.xlsx").xlsexport@t(A3)

A3: @f option means inserting the data of the same fields in A2 into A1

4.6 **Merge by column - different name and number of rows - keep all rows**

Before merging:

Meats.xlsx

	A	B
1	Name	UnitPrice
2	Mutton	7.69
3	Pork	4.58
4	Chicken	5.77
5	Beef	7.96

and

MeatStock.xlsx

	A	B	C
1	Name	Stock	MinimumStock
2	Chicken	4000	3000
3	Duck	2500	2000
4	Beef	3600	2000
5	Mutton	5000	3000

After merging:

	A	B	C	D
1	Name	Stock	MinimumStock	UnitPrice
2	Beef	3600	2000	7.96
3	Pork			4.58
4	Mutton	5000	3000	7.69
5	Duck	2500	2000	
6	Chicken	4000	3000	5.77

Script:

	A
1	=file("Meats.xlsx").xlsimport@t()
2	=file("MeatStock.xlsx").xlsimport@t()
3	=join**@f**(A1:Price,Name;A2:Stock,Name)
4	=A3.new(**[Price.Name,Stock.Name].ifn()**:Name,Stock.Stock,Stock.MinimumStock,Price.UnitPrice)
5	=file("MeatsPriceStock.xlsx").xlsexport@t(A4)

A3: @f option means full join

A4: Bold code means selecting the non-null Name values

4.7 Merge by column - different name and number of rows - keep only duplicate rows

Before merging:

Meats.xlsx

	A	B
1	Name	UnitPrice
2	Mutton	7.69
3	Pork	4.58
4	Chicken	5.77
5	Beef	7.96

and

MeatStock.xlsx

	A	B	C
1	Name	Stock	MinimumStock
2	Chicken	4000	3000
3	Duck	2500	2000
4	Beef	3600	2000
5	Mutton	5000	3000

After merging:

	A	B	C	D
1	Name	Stock	MinimumStock	UnitPrice
2	Beef	3600	2000	7.96
3	Mutton	5000	3000	7.69
4	Chicken	4000	3000	5.77

Script:

	A
1	=file("Meats.xlsx").xlsimport@t()
2	=file("MeatStock.xlsx").xlsimport@t()
3	=join(A1:Price,Name;A2:Stock,Name)
4	=A3.new(Stock.Name,Stock.Stock,Stock.MinimumStock,Price.UnitPrice)
5	=file("MeatsPriceStock.xlsx").xlsexport@t(A4)

A3: Inner join

4.8 Merge by column - different name, number and order of rows - keep only rows of the first file and align the rows

Before merging:

Meats.xlsx

	A	B
1	Name	UnitPrice
2	Mutton	7.69
3	Pork	4.58
4	Chicken	5.77
5	Beef	7.96

and

MeatStock.xlsx

	A	B	C
1	Name	Stock	MinimumStock
2	Chicken	4000	3000
3	Duck	2500	2000
4	Beef	3600	2000
5	Mutton	5000	3000

After merging:

	A	B	C	D
1	Name	Stock	MinimumStock	UnitPrice
2	Beef	3600	2000	7.96
3	Pork			4.58
4	Mutton	5000	3000	7.69
5	Chicken	4000	3000	5.77

Script:

	A
1	=file("Meats.xlsx").xlsimport@t()
2	=file("MeatStock.xlsx").xlsimport@t()
3	=join**@1**(A1:Price,Name;A2:Stock,Name)
4	=A3.new(**[Price.Name,Stock.Name].ifn()**:Name,Stock.Stock,Stock.MinimumStock,Price.UnitPrice)
5	=file("MeatsPriceStock.xlsx").xlsexport@t(A4)

A3: @1 option is left join, note that here is the number “1” rather than the letter “l”

A4: ifn() means selecting non-null Name values

4.9 **Merge by row - convert file names to column values - unfixed number of files**

Before merging:

Apple.xlsx

	A	B
1	Name	Amount
2	Peter	6
3	Mark	3
4	Alice	7
5	Lily	8
6	John	9
7	Cindy	4
8	Maggie	5
9	Leon	2

Bread.xlsx

	A	B
1	Name	Amount
2	Peter	29
3	Mark	59
4	Alice	140
5	Lily	120
6	John	240
7	Cindy	19
8	Maggie	38
9	Leon	29

Pork.xlsx

	A	B
1	Name	Amount
2	Peter	7
3	Mark	11
4	Alice	23
5	Lily	25
6	John	41
7	Cindy	3
8	Maggie	7
9	Leon	8

After merging:

	A	B	C
1	Name	Amount	Commodity
2	Peter	6	Apple
3	Mark	3	Apple
4	Alice	7	Apple
5	Lily	8	Apple
6	John	9	Apple
7	Cindy	4	Apple
8	Maggie	5	Apple
9	Leon	2	Apple
10	Peter	29	Bread
11	Mark	59	Bread
12	Alice	140	Bread
13	Lily	120	Bread
14	John	240	Bread
15	Cindy	19	Bread
16	Maggie	38	Bread
17	Leon	29	Bread
18	Peter	7	Pork
19	Mark	11	Pork
20	Alice	23	Pork
21	Lily	25	Pork
22	John	41	Pork
23	Cindy	3	Pork
24	Maggie	7	Pork
25	Leon	8	Pork

Script:

	A
1	=directory@p("tmp/*.xlsx")
2	=A1.conj((fn=filename@n(~),T(~).derive(fn:Commodity)))
3	=file("Amount.xlsx").xlsexport@t(A2)

A1: List all files in the directory, which can be used to process unfixed number of files

4.10 **Merge by column - convert file names to column names**

Before merging:

Apple.xlsx

	A	B
1	Name	Amount
2	Peter	6
3	Mark	3
4	Alice	7
5	Lily	8
6	John	9
7	Cindy	4
8	Maggie	5
9	Leon	2

Bread.xlsx

	A	B
1	Name	Amount
2	Peter	29
3	Mark	59
4	Alice	140
5	Lily	120
6	John	240
7	Cindy	19
8	Maggie	38
9	Leon	29

Pork.xlsx

	A	B
1	Name	Amount
2	Peter	7
3	Mark	11
4	Alice	23
5	Lily	25
6	John	41
7	Cindy	3
8	Maggie	7
9	Leon	8

After merging:

	A	B	C	D
1	Name	Apple	Bread	Pork
2	Peter	6	29	7
3	Mark	3	59	11
4	Alice	7	140	23
5	Lily	8	120	25
6	John	9	240	41
7	Cindy	4	19	3
8	Maggie	5	38	7
9	Leon	2	29	8

Script:

	A
1	=directory@p("tmp/*.xlsx")
2	=A1.(filename@n(~))
3	=A1.(T(~))
4	=A3(1).new(Name,Amount:${A2(1)},A3(2)(#).Amount:${A2(2)},A3(3)(#).Amount:${A2(3)})
5	=file("Amount.xlsx").xlsexport@t(A4)

A1: List all file names in the directory

A2: Obtain file names without extension

A3: Read files as a table sequence

A4: Convert Amount fields of the original table sequence to corresponding file names while generating a new table sequence

4.11 **Merge by column - one to many - copy data**

Before merging:

Types.xlsx

	A	B
1	Type	Description
2	Fruits	Edible plant fruit which is succulent and tastes mainly sweet and sour
3	Meats	Edible subcutaneous tissue and muscle of animals

Foods.xlsx

	A	B	C
1	Type	Name	UnitPrice
2	Fruits	Apple	1.69
3	Fruits	Banana	0.69
4	Fruits	Peach	0.88
5	Fruits	Strawberry	1.97
6	Meats	Mutton	7.69
7	Meats	Pork	4.58
8	Meats	Chicken	5.77
9	Meats	Duck	6.89
10	Meats	Beef	7.96

After merging:

	A	B	C	D
1	Type	Name	UnitPrice	Description
2	Fruits	Apple	1.69	Edible plant fruit which is succulent and tastes mainly sweet and sour
3	Fruits	Peach	0.88	Edible plant fruit which is succulent and tastes mainly sweet and sour
4	Fruits	Strawberry	1.97	Edible plant fruit which is succulent and tastes mainly sweet and sour
5	Fruits	Banana	0.69	Edible plant fruit which is succulent and tastes mainly sweet and sour
6	Meats	Mutton	7.69	Edible subcutaneous tissue and muscle of animals
7	Meats	Chicken	5.77	Edible subcutaneous tissue and muscle of animals
8	Meats	Beef	7.96	Edible subcutaneous tissue and muscle of animals
9	Meats	Duck	6.89	Edible subcutaneous tissue and muscle of animals
10	Meats	Pork	4.58	Edible subcutaneous tissue and muscle of animals

Script:

	A
1	=T("Types.xlsx")
2	=T("Foods.xlsx")
3	=join**@f**(A1:Type,Type;A2:Food,Type)
4	=A3.new(Food.Type,Food.Name,Food.UnitPrice,Type.Description)
5	=T("FoodsDescription.xlsx",A4)

A3: @f means full join

4.12 Merge by column - one to many - leave subsequent rows empty

Before merging:

Types.xlsx

	A	B
1	Type	Description
2	Fruits	Edible plant fruit which is succulent and tastes mainly sweet and sour
3	Meats	Edible subcutaneous tissue and muscle of animals

Foods.xlsx

	A	B	C
1	Type	Name	UnitPrice
2	Fruits	Apple	1.69
3	Fruits	Banana	0.69
4	Fruits	Peach	0.88
5	Fruits	Strawberry	1.97
6	Meats	Mutton	7.69
7	Meats	Pork	4.58
8	Meats	Chicken	5.77
9	Meats	Duck	6.89
10	Meats	Beef	7.96

After merging:

	A	B	C	D
1	Type	Name	UnitPrice	Description
2	Fruits	Apple	1.69	Edible plant fruit which is succulent and tastes mainly sweet and sour
3	Fruits	Banana	0.69	
4	Fruits	Peach	0.88	
5	Fruits	Strawberry	1.97	
6	Meats	Mutton	7.69	Edible subcutaneous tissue and muscle of animals
7	Meats	Pork	4.58	
8	Meats	Chicken	5.77	
9	Meats	Duck	6.89	
10	Meats	Beef	7.96	

Script:

	A
1	=T("Types.xlsx")
2	=T("Foods.xlsx")
3	=A1.**align**(A2:Type,Type)
4	=A2.new(Type,Name,UnitPrice,A3(#).Description)
5	=T("FoodsDescription.xlsx",A4)

A3: Align means A1 is aligned to A2 with alignment conditions as Type field of A2 and Type field of A1; only the first row is aligned if there are duplicate data in A2

4.13 Merge and de-duplicate by row - duplicate whole row of data

If the whole row of data is duplicated when merging by row, only one of the same rows will be kept during the merge. For example:

Before merging:

	A	B
1	Name	Times
2	Peter	5
3	Mark	9
4	Alice	8
5	Lily	6
6	Cindy	2

and

	A	B
1	Name	Times
2	John	7
3	Cindy	2
4	Maggie	4
5	Leon	3
6	Lily	6

As can be seen from the above figures that the whole row of data in Cindy and Lily rows are duplicated. The merged result is as follows:

	A	B
1	Name	Times
2	Alice	8
3	Cindy	2
4	John	7
5	Leon	3
6	Lily	6
7	Maggie	4
8	Mark	9
9	Peter	5

Script:

	A
1	=file("Customer1.xlsx").xlsimport@t().**sort**(Name,Times)
2	=file("Customer2.xlsx").xlsimport@t().**sort**(Name,Times)
3	=[A1,A2].merge@u(Name,Times)
4	=file("CustomerTimes.xlsx").xlsexport@t(A3)

A1: The original data need to be sorted because of merge

A3: merge@u means the union, using Name and Times as criteria for judging duplication, therefore, if the whole row is used as the criteria, then all the field names should be added

4.14 Merge and de-duplicate by row - duplicate row headers - keep the data that firstly appear

When merging multiple Excel files by row, we may use only the row headers or one/several key columns as the criteria for judging whether data are duplicated. As shown in the following example where Name is used as the criterion for judging duplication:

Before merging:

	A	B
1	Name	Times
2	Peter	5
3	Mark	9
4	Alice	8
5	Lily	6
6	Cindy	2

and

	A	B
1	Name	Times
2	John	7
3	Cindy	2
4	Maggie	4
5	Leon	3
6	Lily	6

From the above figures, Cindy and Lily are the duplicate data in the Name column, and the merged result is as follows:

	A	B
1	Name	Times
2	Alice	8
3	Cindy	2
4	John	7
5	Leon	3
6	Lily	6
7	Maggie	4
8	Mark	9
9	Peter	5

Script:

	A
1	=file("Customer1.xlsx").xlsimport@t().**sort**(Name,Times)
2	=file("Customer2.xlsx").xlsimport@t().**sort**(Name,Times)
3	=[A1,A2].**merge@u**(**Name**)
4	=file("CustomerTimes.xlsx").xlsexport@t(A3)

A1: The original data need to be sorted because of merge

A3: merge@u means the union, using Name as criteria for judging duplication

4.15 Merge and de-duplicate by row - duplicate row headers - keep non-null data

Customer3.xlsx

	A	B
1	Name	Quantity
2	Peter	5
3	Mark	10
4	Alice	24
5	Lily	34
6	Cindy	

Customer4.xlsx

	A	B
1	Name	Quantity
2	John	15
3	Cindy	18
4	Maggie	22
5	Leon	28
6	Lily	

We can see from the above figures that Cindy and Lily rows are duplicated, and the row with null Quantity value will be removed during the merge. The result is as follows:

	A	B
1	Name	Quantity
2	Peter	5
3	Mark	10
4	Alice	24
5	Lily	34
6	John	15
7	Cindy	18
8	Maggie	22
9	Leon	28

Script:

	A
1	=file("Customer3.xlsx").xlsimport@t().select(Quantity!=null)
2	=file("Customer4.xlsx").xlsimport@t().select(Quantity!=null)
3	=A1|A2
4	=file("CustomerQuantity.xlsx").xlsexport@t(A3)

4.16 Merge and de-duplicate by row - duplicate row headers - delete all duplicate data

CustomerTotal.xlsx

	A	B	C	D
1	Name	Times	Quantity	Amount
2	Peter	5	5	30
3	Mark	2	10	60
4	Alice	6	24	144
5	Lily	6	24	144
6	John	7	42	252
7	Cindy	2	2	12
8	Maggie	6	6	36
9	Leon	3	6	36

Customer.xlsx

	A	B	C
1	Name	Times	Quantity
2	Alice	6	24
3	Lily	6	24
4	Peter	5	5
5	Mark	2	10

Since the same key columns will be considered as duplicate data, as a key column, the duplicate records of Name column in Customer.xlsx need to be deleted from CustomerTotal.xlsx, and the result after de-duplication is shown as follows:

	A	B	C	D
1	Name	Times	Quantity	Amount
2	Cindy	2	2	12
3	John	7	42	252
4	Leon	3	6	36
5	Maggie	6	6	36

Script:

	A
1	=file("CustomerTotal.xlsx").xlsimport@t().**sort**(Name)
2	=file("Customer.xlsx").xlsimport@t().**sort**(Name)
3	=[A1,A2].**merge@d**(Name)
4	=file("CustomerTotalNew.xlsx").xlsexport@t(A3)

A1: The original data need to be sorted because of merge

A3: option @d means deleting the data that appear in subsequent table sequence from the first table sequence

4.17 Merge and de-duplicate by column - duplicate column names - keep data in columns that appear later

Before merging:

CustomerFruits.xlsx

	A	B	C	D	E
1	Name	Apple	Strawberry	Bread	Peach
2	Peter	5	5	30	10
3	Mark	2	10	60	15
4	Alice	6	24	144	18
5	Lily	6	24	144	20
6	John	7	42	252	24
7	Cindy	2	2	12	22
8	Maggie	6	6	36	16
9	Leon	3	6	36	13

and

CustomerMeats.xlsx

	A	B	C	D	E
1	Name	Mutton	Pork	Bread	Duck
2	Peter	6	7	29	12
3	Mark	3	11	59	17
4	Alice	7	23	140	19
5	Lily	8	25	120	22
6	John	9	41	240	23
7	Cindy	4	3	19	28
8	Maggie	5	7	38	15
9	Leon	2	8	29	14

It can be seen from the figures that the Bread columns are duplicated, and we want to keep the Bread column of the second file and delete the Bread field in the first file after merging. The result is as follows:

	A	B	C	D	E	F	G	H
1	Name	Apple	Strawberry	Peach	Mutton	Pork	Bread	Duck
2	Peter	5	5	10	6	7	29	12
3	Mark	2	10	15	3	11	59	17
4	Alice	6	24	18	7	23	140	19
5	Lily	6	24	20	8	25	120	22
6	John	7	42	24	9	41	240	23
7	Cindy	2	2	22	4	3	19	28
8	Maggie	6	6	16	5	7	38	15
9	Leon	3	6	13	2	8	29	14

Script:

	A
1	=file("CustomerFruits.xlsx").xlsimport@t()
2	=file("CustomerMeats.xlsx").xlsimport@t()
3	=A1.new(Name,Apple,Strawberry,Peach,A2(#).Mutton,A2(#).Pork,A2(#).Bread,A2(#).Duck)
4	=file("CustomerFoods.xlsx").xlsexport@t(A3)

4.18 Merge by row and column simultaneously - keep data that firstly appear

Before merging:

CustomerFruits1.xlsx

	A	B	C	D	E
1	Name	Apple	Strawberry	Bread	Peach
2	Peter	5	5	30	10
3	Mark	2	10	60	15
4	Alice	6	24	144	18
5	Lily	6	24	144	20
6	John	7	42	252	24

CustomerMeats1.xlsx

	A	B	C	D	E
1	Name	Mutton	Pork	Bread	Duck
2	Lily	8	25	120	22
3	John	9	41	240	23
4	Cindy	4	3	19	28
5	Maggie	5	7	38	15
6	Leon	2	8	29	14

According to the order of CustomerFruits1.xlsx first and CustomerMeats1.xlsx later, the duplicate records that appear in CustomerFruits1.xlsx first are kept. The result after merging is:

	A	B	C	D	E	F	G	H
1	Name	Apple	Bread	Duck	Mutton	Peach	Pork	Strawberry
2	Alice	6	144			18		24
3	Cindy		19	28	4		3	
4	John	7	252	23	9	24	41	42
5	Leon		29	14	2		8	
6	Lily	6	144	22	8	20	25	24
7	Maggie		38	15	5		7	
8	Mark	2	60			15		10
9	Peter	5	30			10		5

Script:

	A
1	=file("CustomerFruits1.xlsx").xlsimport@t()
2	=file("CustomerMeats1.xlsx").xlsimport@t()
3	=A1.pivot@r(Name;col,val)
4	=A2.pivot@r(Name;col,val)
5	=(A3|A4).**group@1**(Name,col)
6	=A5.pivot(Name;col,val)
7	=file("CustomerFoods1.xlsx").xlsexport@t(A6)

A3: Transpose the data of original cross layout to a list

A5: Select the record that appears firstly after grouping

A6: Transpose the data back to cross layout

4.19 Format conversion - merge multiple card-style files to form one row-based table

There are multiple card-style files as follows:

Andrew.Fuller.xlsx：

	A	B	C	D	E
1	ID	2	Gender	Male	
2	FirstName	Andrew	LastName	Fuller	
3	Title	Vice President, Sales			
4	Birthday	1952-02-19			
5	HomePhone	(206) 555-9482			
6	Address	908 W. Capital Way			
7	PostalCode				

Janet.Leverling.xlsx：

	A	B	C	D	E
1	ID	3	Gender	Female	
2	FirstName	Janet	LastName	Leverling	
3	Title	Sales Representative			
4	Birthday	1963-08-30			
5	HomePhone	(206) 555-3412			
6	Address	722 Moss Bay Blvd.			
7	PostalCode				

Margaret.Peacock.xlsx：

	A	B	C	D	E
1	ID	4	Gender	Female	
2	FirstName	Margaret	LastName	Peacock	
3	Title	Sales Representative			
4	Birthday	1937-09-19			
5	HomePhone	(206) 555-8122			
6	Address	4110 Old Redmond Rd.			
7	PostalCode				

Nancy.Davolio.xlsx：

	A	B	C	D	E
1	ID	1	Gender	Female	
2	FirstName	Nancy	LastName	Davolio	
3	Title	Sales Representative			
4	Birthday	1948-12-08			
5	HomePhone	(206) 555-9857			
6	Address	507 - 20th Ave. E.			
7	PostalCode				

…

Now we want to merge them to form one row-based table. The merged format is as follows:

Employee.xlsx：

	A	B	C	D	E	F	G	H	I
1	ID	FirstName	LastName	Gender	Title	Birthday	HomePhone	PostalCode	Address
2	1	Nancy	Davolio	Female	Sales Representative	1948-12-08 00:00:00	(206) 555-9857	98122	507 - 20th Ave. E.Apt. 2A
3	2	Andrew	Fuller	Male	Vice President, Sales	1952-02-19 00:00:00	(206) 555-9482	98401	908 W. Capital Way
4	3	Janet	Leverling	Female	Sales Representative	1963-08-30 00:00:00	(206) 555-3412	98033	722 Moss Bay Blvd.
5	4	Margaret	Peacock	Female	Sales Representative	1937-09-19 00:00:00	(206) 555-8122	98052	4110 Old Redmond Rd.

Script:

	A	B
1	=directory@p("Cards/*.*.xlsx")	
2	=create(ID,FirstName,LastName,Gender,Title,Birthday,HomePhone,PostalCode,Address)	
3	[B1,B2,D2,D1,B3,B4,B5,B7,B6]	
4	for A1	=file(A4).xlsopen()
5		=A3.(B4.xlscell(~))
6		>A2.record(B5)
7	=T("Cards/Employee.xlsx",A2)	

A1: List all files named by Name in the Cards directory

A2: Create a result table sequence

A3: List the cell names in the card-style files to be read

A4: Loop through the files in A1

B4: Open the files

B5: Read the data of the cells listed in A3 in the file

B6: Insert the data in B5 into A2

A7: Write the table sequence in A2 to Employee.xlsx

4.20 Format conversion - merge multiple primary-sub table files to form two row-based tables

There are multiple primary-sub tables. Now we want to separate the primary tables to form a row-based table, and separate the sub-tables to form another row-based table.

An example of the primary-sub table files is as follows:

Orders1.xlsx:

	A	B	C	D	E	F
1	OrderID	10248		OrderDate	1996-07-04 00:00:00	
2	CustomerID	Vins et alcools Chevalier		EmployeeID	Steven.Buchanan	
3	Consignee	Wolski Zajazd				
4	ShipAddress	59 rue de l'Abbaye				
6	ProductID	ProductName	UnitPrice	Quantity	Discount	ExtendedPrice
7	11	Queso Cabrales	14.0000	12	0	168.0000
8	42	Singaporean Hokkien Fried Mee	9.8000	10	0	98.0000
9	72	Mozzarella di Giovanni	34.8000	5	0	174.0000

Orders2.xlsx:

	A	B	C	D	E	F
1	OrderID	10249		OrderDate	1996-07-05 00:00:00	
2	CustomerID	Toms Spezialit?ten		EmployeeID	Michael.Suyama	
3	Consignee	Ana Trujillo Emparedados y helados				
4	ShipAddress	Luisenstr. 48				
6	ProductID	ProductName	UnitPrice	Quantity	Discount	ExtendedPrice
7	14	Tofu	18.6000	9	0	167.4000
8	51	Manjimup Dried Apples	42.4000	40	0	1696.0000

Orders3.xlsx:

	A	B	C	D	E	F
1	OrderID	10250		OrderDate	1996-07-08 00:00:00	
2	CustomerID	Hanari Carnes		EmployeeID	Margaret.Peacock	
3	Consignee	Antonio Moreno Taqueria				
4	ShipAddress	Rua do Pa?o, 67				
6	ProductID	ProductName	UnitPrice	Quantity	Discount	ExtendedPrice
7	41	Jack's New England Clam Chowder	7.7000	10	0	77.0000
8	51	Manjimup Dried Apples	42.4000	35	0.15	1261.4000
9	65	Louisiana Fiery Hot Pepper Sauce	16.8000	15	0.15	214.2000

The results after separating are as follows:

Orders.xlsx：

	A	B	C	D	E	F
1	OrderID	OrderDate	CustomerID	EmployeeID	Consignee	ShipAddress
2	10248	1996-07-04	Vins et alcools Chevalier	Steven.Buchanan	Wolski Zajazd	59 rue de l'Abbaye
3	10249	1996-07-05	Toms Spezialit?ten	Michael.Suyama	Ana Trujillo Emparedados y helados	Luisenstr. 48
4	10250	1996-07-08	Hanari Carnes	Margaret.Peacock	Antonio Moreno Taquería	Rua do Pa?o, 67

OrderDetails.xlsx：

	A	B	C	D	E	F	G
1	OrderID	ProductID	ProductName	UnitPrice	Quantity	Discount	ExtendedPrice
2	10248	11	Queso Cabrales	14	12	0	168
3	10248	42	Singaporean Hokkien Fried Mee	9.8	10	0	98
4	10248	72	Mozzarella di Giovanni	34.8	5	0	174
5	10249	14	Tofu	18.6	9	0	167.4
6	10249	51	Manjimup Dried Apples	42.4	40	0	1696
7	10250	41	Jack's New England Clam Chowder	7.7	10	0	77
8	10250	51	Manjimup Dried Apples	42.4	35	0.15	1261.4
9	10250	65	Louisiana Fiery Hot Pepper Sauce	16.8	15	0.15	214.2

Script:

	A	B
1	=directory@p("Orders/Order*.xlsx").(file(~).xlsopen())	
2	=create(OrderID,OrderDate,CustomerID,EmployeeID,Consignee,ShipAddress)	
3	=create(OrderID,ProductID,ProductName,UnitPrice,Quantity,Discount,ExtendedPrice)	
4	for A1	=A4.xlsimport@t(;1,6).select(ProductID).derive(A4.xlscell("B1"):OrderID)
5		>A2.insert(0,A4.xlscell("B1"):OrderID,A4.xlscell("E1"):OrderDate,A4.xlscell("B2"):CustomerID, A4.xlscell("E2"):EmployeeID,A4.xlscell("B3"):Consignee,A4.xlscell("B4"):ShipAddress)
6		>A3.insert@f(0:B4)
7	=T("Orders/Orders.xlsx",A2)	
8	=T("Orders/OrderDetails.xlsx",A3)	

A1: List the .xlsx files starting with Order in the Orders directory and open them one by one

A2: Create the table sequence of primary tables

A3: Create the table sequence of sub tables

A4: Loop through the files in A1 one by one

B4: Read the data starting from the sixth row in the file, return them to the table sequence, and add the column OrderID

B5: Read the data of the primary tables in the file one by one, and insert them into A2

B6: Insert the data of sub tables read in B4 into A3

A7: Write the table sequence of primary tables created in A2 to the file Orders.xlsx

A8: Write the table sequence of sub tables created in A3 to the file OrderDetails.xlsx

4.21 Aggregate files - same rows and columns

In practical business, sometimes we need to aggregate the data while merging multiple Excel files, for example:

Apple.xlsx

	A	B
1	Name	Amount
2	Peter	6
3	Mark	3
4	Alice	7
5	Lily	8
6	John	9
7	Cindy	4
8	Maggie	5
9	Leon	2

Bread.xlsx

	A	B
1	Name	Amount
2	Peter	29
3	Mark	59
4	Alice	140
5	Lily	120
6	John	240
7	Cindy	19
8	Maggie	38
9	Leon	29

Pork.xlsx

	A	B
1	Name	Amount
2	Peter	7
3	Mark	11
4	Alice	23
5	Lily	25
6	John	41
7	Cindy	3
8	Maggie	7
9	Leon	8

Now we need to aggregate the Amount fields to form a column “total amount”, and store it to a new file. The result is as follows:

	A	B
1	Name	TotalAmount
2	Peter	42
3	Mark	73
4	Alice	170
5	Lily	153
6	John	290
7	Cindy	26
8	Maggie	50
9	Leon	39

Script:

	A
1	=file("Apple.xlsx").xlsimport@t()
2	=file("Bread.xlsx").xlsimport@t()
3	=file("Pork.xlsx").xlsimport@t()
4	=A1.new(Name,Amount+A2(#).Amount+A3(#).Amount:TotalAmount)
5	=file("TotalAmount.xlsx").xlsexport@t(A4)

4.22 Aggregate files - merge by row and column simultaneously - aggregate duplicate records

Before merging:

CustomerFruits1.xlsx

	A	B	C	D	E
1	Name	Apple	Strawberry	Bread	Peach
2	Peter	5	5	30	10
3	Mark	2	10	60	15
4	Alice	6	24	144	18
5	Lily	6	24	144	20
6	John	7	42	252	24

CustomerMeats1.xlsx

	A	B	C	D	E
1	Name	Mutton	Pork	Bread	Duck
2	Lily	8	25	120	22
3	John	9	41	240	23
4	Cindy	4	3	19	28
5	Maggie	5	7	38	15
6	Leon	2	8	29	14

The final result after aggregating duplicate records and merging is:

	A	B	C	D	E	F	G	H
1	Name	Apple	Bread	Duck	Mutton	Peach	Pork	Strawberry
2	Alice	6	144			18		24
3	Cindy		19	28	4		3	
4	John	7	492	23	9	24	41	42
5	Leon		29	14	2		8	
6	Lily	6	264	22	8	20	25	24
7	Maggie		38	15	5		7	
8	Mark	2	60			15		10
9	Peter	5	30			10		5

Script:

	A
1	=file("CustomerFruits1.xlsx").xlsimport@t()
2	=file("CustomerMeats1.xlsx").xlsimport@t()
3	=A1.pivot@r(Name;col,val)
4	=A2.pivot@r(Name;col,val)
5	=(A3|A4).groups(Name,col;sum(val):val)
6	=A5.pivot(Name;col,val)
7	=file("CustomerFoods2.xlsx").xlsexport@t(A6)

A3: Transpose the data of original cross layout to a list

A5: Grouping and aggregating

A6: Transpose back to cross layout

4.23 Aggregate files - aggregate by cell positions - unfixed number of files

The head office receives the balance sheets from each branch, one of the tables is shown below (there are 37 rows in total, but only 14 rows are shown in the table):

	A	B	C	D	E	F
1	Balance Sheet					
2						
3	Assets	Beginning balance	Ending balance	Liability & Equity	Beginning balance	Ending balance
4	Current assets			Current liabilities		
5	Cash	56,554,453.04	84,682,835.11	Short-term loans		
6	Short term investments			Tradable financial liabilities		
7	Notes receivable			Notes payable		
8	Accounts receivable	1,694,252.02	31,947,059.00	Accounts payable	23,732,995.19	22,472,155.41
9	Accounts prepaid	31,937,029.69	2,300,880.00	Advances from customers	82,621,644.00	69,292,831.00
10	Interest receivable			Accrued payroll		5,165.34
11	Dividend receivable			Taxes payable	19,694,272.18	9,199,386.78
12	Other receivables	65,340,103.71	6,701,354.86	Interest payable		
13	Inventories	184,772,072.22	195,167,858.72	Dividend payable		
14	Prepaid and deferred ex	3,342,812.95	2,893,135.32	Other creditors	37,875,543.35	3,575,071.48

Now we want to aggregate these balance sheets to generate the balance sheet of head office.

Script:

	A	B	C
1	=directory@p("zc*.xlsx")		
2	=A1.(file(~).xlsopen())		
3	=to(4,37)	[B,C,E,F]	=A3.(B3.(~/A3.~)).conj()
4	for C3	>v=null	
5		for A2	>v+=number(B5.xlscell(A4,1))
6		>A2(1).xlscell(A4,1;string(v))	
7	=file("total.xlsx").xlswrite(A2(1))		

A1: List all the to-be-aggregated balance sheets whose file names begin with zc in the folder, and @p option means listing the full path of the file

A2: Open the files listed in A1 as Excel objects

A3: Specify the row number range of to-be-aggregated numeric cells: 4-37

B3: Specify the column numbers of to-be-aggregated numeric cells: B,C,E,F

C3: Concatenate the row numbers in A3 and column numbers in B3 to form the name for every to-be-aggregated numeric cell

A4: Loop through all to-be-aggregated numeric cells in C3

B4: Define the aggregation value variable v

B5: Loop through the balance sheet of every branch

C5: Read the value of current aggregation cell from the balance sheet of current branch, convert it to a number and add it to v

B6: Save the added v to the balance sheet of the first branch

A7: Save the balance sheet of the first branch to the balance sheet of head office total.xlsx

4.24 Aggregate files - append and aggregate

There is a statistical table for daily purchase and delivery of goods:

	A	B	C	D
1	Date	Goods	Purchase	Delivery
2	2020-08-03	milk	18	12
3	2020-08-03	vine	25	33
4	2020-08-03	sauce	30	26

There is also a summary table for daily purchase, delivery and inventory of goods:

	A	B	C	D	E
1	Date	Goods	Purchase	Delivery	Inventory
2	2020-08-01	milk	24	15	45
3	2020-08-01	vine	52	63	84
4	2020-08-01	sauce	48	39	67
5	2020-08-02	milk	32	18	59
6	2020-08-02	vine	12	37	59
7	2020-08-02	sauce	31	49	49

Now we want to append the daily purchase and delivery data to the summary table to calculate the latest inventory: inventory of the previous day + purchase - delivery. The aggregation result is:

	A	B	C	D	E
1	Date	Goods	Purchase	Delivery	Inventory
2	2020-08-01	milk	24	15	45
3	2020-08-01	vine	52	63	84
4	2020-08-01	sauce	48	39	67
5	2020-08-02	milk	32	18	59
6	2020-08-02	vine	12	37	59
7	2020-08-02	sauce	31	49	49
8	2020-08-03	milk	18	12	65
9	2020-08-03	vine	25	33	51
10	2020-08-03	sauce	30	26	53

Script:

	A
1	=T("20200803.xlsx").derive(Inventory)
2	=T("total.xlsx")
3	=A1.run(Inventory=A2.select@z1(Goods==A1.Goods).Inventory+Purchase-Delivery)
4	=file("total.xlsx").xlsexport@a(A3)

A1: Read the current day data to be appended and aggregated and add a new “Inventory column.

A2: Read the data of summary table

A3: Loop through every row in A1 so that the value of Inventory is the Inventory of the last good in summary table plus the current Purchase and minus the current Delivery. @z1 option means selecting the first row that satisfies the condition from back to front

A4: Append and save the result of A3 to the file total.xlsx, and @a option means appending the data

4.25 Aggregate files - cumulate and aggregate

There are statistical tables for the daily sales of some goods in current month, one table per day. Now we want to add the cumulative value to the column “monthly cumulative sales” of these files.

Before merging:

20220101.xlsx

	A	B	C
1	Name	DailySales	MonthlyCumulativeSales
2	Apple	1419	
3	Banana	4753	
4	Peach	2107	
5	Strawberry	4634	

20220102.xlsx

	A	B	C
1	Name	DailySales	MonthlyCumulativeSales
2	Apple	6233	
3	Banana	9233	
4	Peach	799	
5	Strawberry	1162	

20220103.xlsx

	A	B	C
1	Name	DailySales	MonthlyCumulativeSales
2	Apple	1314	
3	Banana	1162	
4	Peach	7917	
5	Strawberry	9393	

Files of other dates are omitted.

After merging:

20220101.xlsx

	A	B	C
1	Name	DailySales	MonthlyCumulativeSales
2	Apple	1419	1419
3	Banana	4753	4753
4	Peach	2107	2107
5	Strawberry	4634	4634

20220102.xlsx

	A	B	C
1	Name	DailySales	MonthlyCumulativeSales
2	Apple	6233	7652
3	Banana	9233	13986
4	Peach	799	2906
5	Strawberry	1162	5796

20220103.xlsx

	A	B	C
1	Name	DailySales	MonthlyCumulativeSales
2	Apple	1314	8966
3	Banana	1162	15148
4	Peach	7917	10823
5	Strawberry	9393	15189

Files of other dates are omitted.

Script:

	A	B
1	2022-01-01	2022-01-31
2	=periods(A1,B1).(string(~,"yyyyMMdd")+".xlsx")	
3	=A2.(T(~))	
4	>A3(1).run(MonthlyCumulativeSales=DailySales)	
5	for A3.to(2,)	=A5.run(MonthlyCumulativeSales=DailySales+A3(#A5).select@1(Name==A5.Name). MonthlyCumulativeSales)
6	=A3.run(T(A2(#),~))	

4.26 Aggregate files - insert aggregation sheet

A shopping mall complies a purchase summary table of key customers for 12 months of the year in the format shown below:

Jan.xlsx：

	A	B	C	D	E
1	CustomerName	Apple	Banana	Peach	Strawberry
2	Peter	69	21	80	19
3	Mark	20	0	45	10
4	Alice	1	64	19	26
5	Lily	25	55	88	47

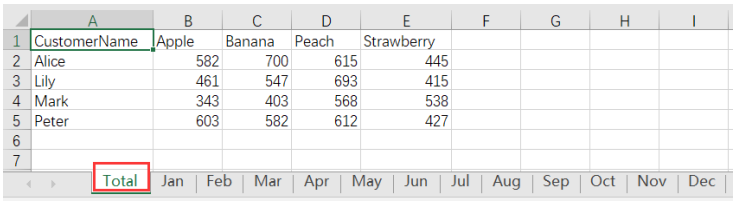
Feb.xlsx：

	A	B	C	D	E
1	CustomerName	Apple	Banana	Peach	Strawberry
2	Peter	74	27	49	10
3	Mark	24	10	85	66
4	Alice	54	72	5	15
5	Lily	22	68	79	5

Files of other months are omitted.

Now we want to aggregate these Excel files to different sheets of one Excel file with file name as the sheet name, and insert an aggregation sheet named Total on the home page.

The aggregated Excel is as follows:



Script:

	A	B
1	[Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec]	
2	=A1.(T(~+".xlsx"))	
3	=A2.conj().groups(CustomerName;sum(Apple):Apple, sum(Banana):Banana,sum(Peach):Peach,sum(Strawberry):Strawberry)	
4	=T("Total.xlsx",A3;"Total")	
5	for A2	=file("Total.xlsx").xlsexport@at(A5;A1(#A5))

A3: Aggregate data

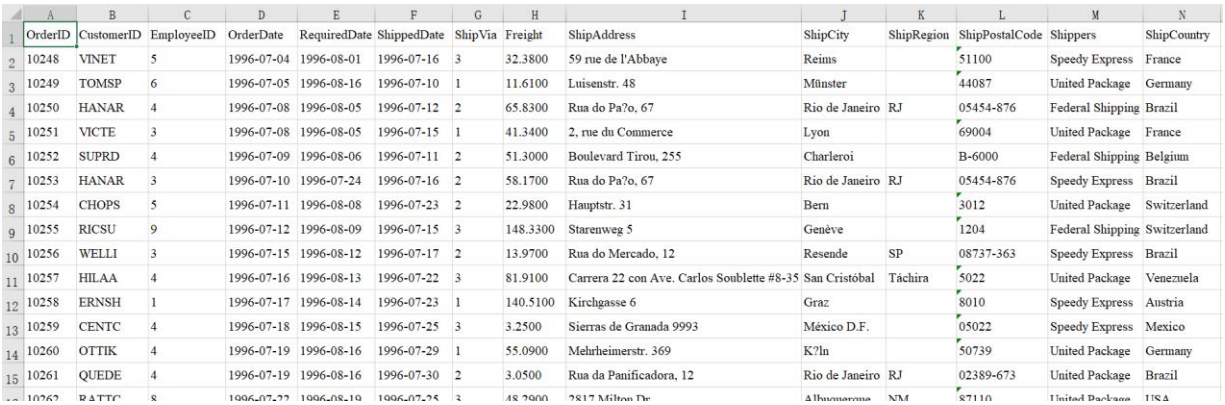
A4: Export A3 to the first sheet of Excel, and name it as Total

B5: Append the original data to the subsequent sheets of Excel and name them with file name; @a option means appending the data

Chapter 5 **Split Excel file**

5.1 **Split by row - by number of rows**

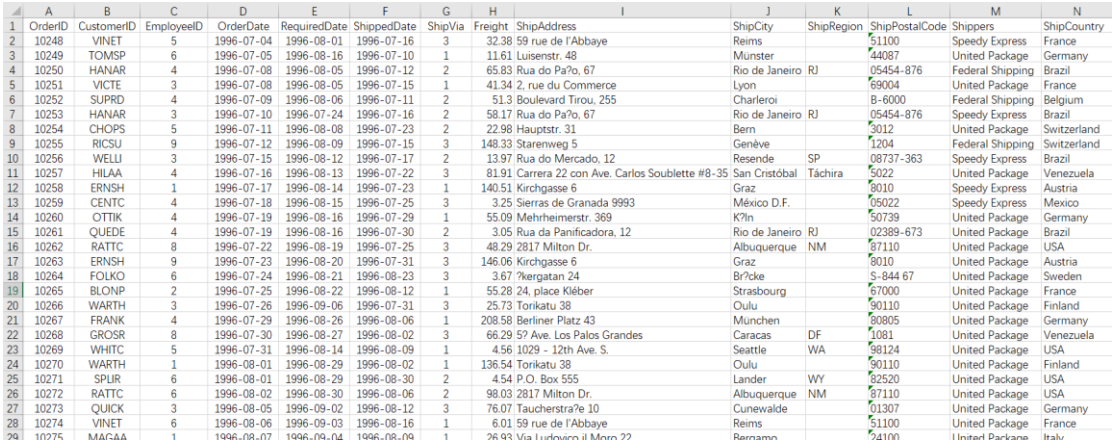
We have an order table file Orders.xls, and part of the data is shown in the figure below. The first row contains the column titles, and the data start from the second row with one piece of data per row.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10248	VINET	5	1996-07-04	1996-08-01	1996-07-16	3	32.3800	59 rue de l'Abbaye	Reims		51100	Speedy Express	France
3	10249	TOMSP	6	1996-07-05	1996-08-16	1996-07-10	1	11.6100	Luisenstr. 48	Münster		44087	United Package	Germany
4	10250	HANAR	4	1996-07-08	1996-08-05	1996-07-12	2	65.8300	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Federal Shipping	Brazil
5	10251	VICTE	3	1996-07-08	1996-08-05	1996-07-15	1	41.3400	2, rue du Commerce	Lyon		69004	United Package	France
6	10252	SUPRD	4	1996-07-09	1996-08-06	1996-07-11	2	51.3000	Boulevard Tirou, 255	Charleroi		B-6000	Federal Shipping	Belgium
7	10253	HANAR	3	1996-07-10	1996-07-24	1996-07-16	2	58.1700	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Speedy Express	Brazil
8	10254	CHOPS	5	1996-07-11	1996-08-08	1996-07-23	2	22.9800	Hauptstr. 31	Bern		3012	United Package	Switzerland
9	10255	RICSU	9	1996-07-12	1996-08-09	1996-07-15	3	148.3300	Starenweg 5	Genève		1204	Federal Shipping	Switzerland
10	10256	WELLI	3	1996-07-15	1996-08-12	1996-07-17	2	13.9700	Rua do Mercado, 12	Resende	SP	08737-363	Speedy Express	Brazil
11	10257	HILAA	4	1996-07-16	1996-08-13	1996-07-22	3	81.9100	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	5022	United Package	Venezuela
12	10258	ERNSH	1	1996-07-17	1996-08-14	1996-07-23	1	140.5100	Kirchgasse 6	Graz		8010	Speedy Express	Austria
13	10259	CENTC	4	1996-07-18	1996-08-15	1996-07-25	3	3.2500	Sierras de Granada 9993	México D.F.		05022	Speedy Express	Mexico
14	10260	OTTIK	4	1996-07-19	1996-08-16	1996-07-29	1	55.0900	Mehrheimerstr. 369	K?ln		50739	United Package	Germany
15	10261	QUEDE	4	1996-07-19	1996-08-16	1996-07-30	2	3.0500	Rua da Panificadora, 12	Rio de Janeiro	RJ	02389-673	United Package	Brazil
16	10262	RATTC	8	1996-07-22	1996-08-19	1996-07-25	3	48.2900	2817 Milton Dr.	Albuquerque	NM	87110	United Package	USA

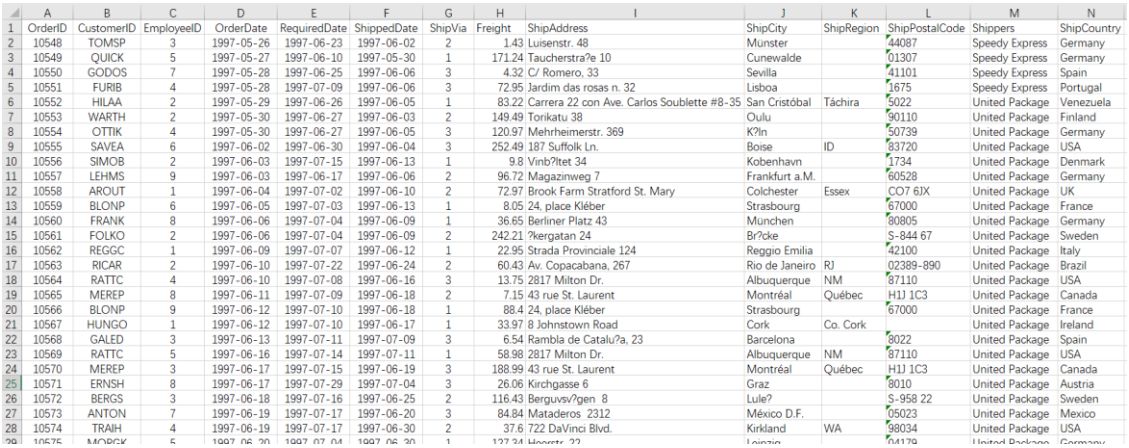
Now we need to split this file into several small files by the specified number of rows. The split result is shown as follows:

Orders1.xlsx:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10248	VINET	5	1996-07-04	1996-08-01	1996-07-16	3	32.38	59 rue de l'Abbaye	Reims		51100	Speedy Express	France
3	10249	TOMSP	6	1996-07-05	1996-08-16	1996-07-10	1	11.61	Luisenstr. 48	Münster		44087	United Package	Germany
4	10250	HANAR	4	1996-07-08	1996-08-05	1996-07-12	2	65.83	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Federal Shipping	Brazil
5	10251	VICTE	3	1996-07-08	1996-08-05	1996-07-15	1	41.34	2, rue du Commerce	Lyon		69004	United Package	France
6	10252	SUPRD	4	1996-07-09	1996-08-06	1996-07-11	2	51.3	Boulevard Tirou, 255	Charleroi		B-6000	Federal Shipping	Belgium
7	10253	HANAR	3	1996-07-10	1996-07-24	1996-07-16	2	58.17	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Speedy Express	Brazil
8	10254	CHOPS	5	1996-07-11	1996-08-08	1996-07-23	2	22.98	Hauptstr. 31	Bern		3012	United Package	Switzerland
9	10255	RICSU	9	1996-07-12	1996-08-09	1996-07-15	3	148.33	Starenweg 5	Genève		1204	Federal Shipping	Switzerland
10	10256	WELLI	3	1996-07-15	1996-08-12	1996-07-17	2	13.97	Rua do Mercado, 12	Resende	SP	08737-363	Speedy Express	Brazil
11	10257	HILAA	4	1996-07-16	1996-08-13	1996-07-22	3	81.91	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	5022	United Package	Venezuela
12	10258	ERNSH	1	1996-07-17	1996-08-14	1996-07-23	1	140.51	Kirchgasse 6	Graz		8010	Speedy Express	Austria
13	10259	CENTC	4	1996-07-18	1996-08-15	1996-07-25	3	3.25	Sierras de Granada 9993	México D.F.		05022	Speedy Express	Mexico
14	10260	OTTIK	4	1996-07-19	1996-08-16	1996-07-29	1	55.09	Mehrheimerstr. 369	K?ln		50739	United Package	Germany
15	10261	QUEDE	4	1996-07-19	1996-08-16	1996-07-30	2	3.05	Rua da Panificadora, 12	Rio de Janeiro	RJ	02389-673	United Package	Brazil
16	10262	RATTC	8	1996-07-22	1996-08-19	1996-07-25	3	48.29	2817 Milton Dr.	Albuquerque	NM	87110	United Package	USA
17	10263	ERNSH	9	1996-07-23	1996-08-20	1996-07-31	3	146.06	Kirchgasse 6	Graz		8010	United Package	Austria
18	10264	FOLKO	6	1996-07-24	1996-08-21	1996-08-23	3	3.67	?kergatan 24	Br?cke		S-844 67	United Package	Sweden
19	10265	BLONP	2	1996-07-25	1996-08-22	1996-08-12	1	55.28	24, place Kléber	Strasbourg		67000	United Package	France
20	10266	WARTH	3	1996-07-26	1996-09-06	1996-07-31	3	25.73	Torikatu 38	Oulu		90110	United Package	Finland
21	10267	FRANK	4	1996-07-29	1996-08-26	1996-08-06	1	208.58	Berliner Platz 43	München		80805	United Package	Germany
22	10268	GROSR	8	1996-07-30	1996-08-27	1996-08-02	3	66.29	5? Ave. Los Palos Grandes	Caracas	DF	1081	United Package	Venezuela
23	10269	WHITC	5	1996-07-31	1996-08-14	1996-08-09	1	4.56	1029 - 12th Ave. S.	Seattle	WA	98124	United Package	USA
24	10270	WARTH	1	1996-08-01	1996-08-29	1996-08-02	1	136.54	Torikatu 38	Oulu		90110	United Package	Finland
25	10271	SPLIR	6	1996-08-01	1996-08-29	1996-08-30	2	4.54	P.O. Box 555	Lander	WY	82520	United Package	USA
26	10272	RATTC	6	1996-08-02	1996-08-30	1996-08-06	2	98.03	2817 Milton Dr.	Albuquerque	NM	87110	United Package	USA
27	10273	QUICK	3	1996-08-05	1996-09-02	1996-08-12	3	76.07	Taucherstra?e 10	Cunewalde		01307	United Package	Germany
28	10274	VINET	6	1996-08-06	1996-09-03	1996-08-16	1	6.01	59 rue de l'Abbaye	Reims		51100	United Package	France
29	10275	MAGAA	1	1996-08-07	1996-09-04	1996-08-09	1	26.93	Via Ludovico il Moro 22	Bergamo		24100	United Package	Italy

Orders2.xlsx:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10548	TOMSP	3	1997-05-26	1997-06-23	1997-06-02	2	1.43	Luisenstr. 48	Münster		44087	Speedy Express	Germany
3	10549	QUICK	5	1997-05-27	1997-06-10	1997-05-30	1	171.24	Taucherstra?e 10	Cunewalde		01307	Speedy Express	Germany
4	10550	GODOS	7	1997-05-28	1997-06-25	1997-06-06	3	4.32	C/ Romero, 33	Sevilla		41101	Speedy Express	Spain
5	10551	FURIB	4	1997-05-28	1997-07-09	1997-06-06	3	72.95	Jardim das rosas n. 32	Lisboa		1675	Speedy Express	Portugal
6	10552	HILAA	2	1997-05-29	1997-06-26	1997-06-05	1	83.22	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	5022	United Package	Venezuela
7	10553	WARTH	2	1997-05-30	1997-06-27	1997-06-03	2	149.49	Torikatu 38	Oulu		90110	United Package	Finland
8	10554	OTTIK	4	1997-05-30	1997-06-27	1997-06-05	3	120.97	Mehrheimerstr. 369	K?ln		50739	United Package	Germany
9	10555	SAVEA	6	1997-06-02	1997-06-30	1997-06-04	3	252.49	187 Suffolk Ln.	Boise	ID	83720	United Package	USA
10	10556	SIMOB	2	1997-06-03	1997-07-15	1997-06-13	1	9.8	Vinb?ltet 34	Kobenhavn		1734	United Package	Denmark
11	10557	LEHMS	9	1997-06-03	1997-06-17	1997-06-06	2	96.72	Magazinweg 7	Frankfurt a.M.		60528	United Package	Germany
12	10558	AROUT	1	1997-06-04	1997-07-02	1997-06-10	2	72.97	Brook Farm Stratford St. Mary	Colchester	Essex	CO7 6JX	United Package	UK
13	10559	BLONP	6	1997-06-05	1997-07-03	1997-06-13	1	8.05	24, place Kléber	Strasbourg		67000	United Package	France
14	10560	FRANK	8	1997-06-06	1997-07-04	1997-06-09	1	36.65	Berliner Platz 43	München		80805	United Package	Germany
15	10561	FOLKO	2	1997-06-06	1997-07-04	1997-06-09	2	242.21	?kergatan 24	Br?cke		S-844 67	United Package	Sweden
16	10562	REGGC	1	1997-06-09	1997-07-07	1997-06-12	1	22.95	Strada Provinciale 124	Reggio Emilia		42100	United Package	Italy
17	10563	RICAR	2	1997-06-10	1997-07-22	1997-06-24	2	60.43	Av. Copacabana, 267	Rio de Janeiro	RJ	02389-890	United Package	Brazil
18	10564	RATTC	4	1997-06-10	1997-07-08	1997-06-16	3	13.75	2817 Milton Dr.	Albuquerque	NM	87110	United Package	USA
19	10565	MEREP	8	1997-06-11	1997-07-09	1997-06-18	2	7.15	43 rue St. Laurent	Montréal	Québec	H1J 1C3	United Package	Canada
20	10566	BLONP	9	1997-06-12	1997-07-10	1997-06-18	1	88.4	24, place Kléber	Strasbourg		67000	United Package	France
21	10567	HUNGO	1	1997-06-12	1997-07-10	1997-06-17	1	33.97	8 Johnstown Road	Cork	Co. Cork		United Package	Ireland
22	10568	GALED	3	1997-06-13	1997-07-11	1997-07-09	3	6.54	Rambla de Catalu?a, 23	Barcelona		8022	United Package	Spain
23	10569	RATTC	5	1997-06-16	1997-07-14	1997-07-11	1	58.98	2817 Milton Dr.	Albuquerque	NM	87110	United Package	USA
24	10570	MEREP	3	1997-06-17	1997-07-15	1997-06-19	3	188.99	43 rue St. Laurent	Montréal	Québec	H1J 1C3	United Package	Canada
25	10571	ERNSH	8	1997-06-17	1997-07-29	1997-07-04	3	26.06	Kirchgasse 6	Graz		8010	United Package	Austria
26	10572	BERGS	3	1997-06-18	1997-07-16	1997-06-25	2	116.43	Berguvsv?gen 8	Lule?		S-958 22	United Package	Sweden
27	10573	ANTON	7	1997-06-19	1997-07-17	1997-06-20	3	84.84	Mataderos 2312	México D.F.		05023	United Package	Mexico
28	10574	TRAIH	4	1997-06-19	1997-07-17	1997-06-30	2	37.6	722 DaVinci Blvd.	Kirkland	WA	98034	United Package	USA
29	10575	MORGK	5	1997-06-20	1997-07-04	1997-06-30	1	127.34	Heerstr. 22	Leipzig		04179	United Package	Germany

Orders3.xlsx:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10848	CONSH	7	1998-01-23	1998-02-20	1998-01-29	2	38.24	Berkeley Gardens 12 Brewery	London		WX1 6LT	Federal Shipping	UK
3	10849	KOENE	9	1998-01-23	1998-02-20	1998-01-30	2	0.56	Maubelstr. 90	Brandenburg		14776	Federal Shipping	Germany
4	10850	VICTE	1	1998-01-23	1998-03-06	1998-01-30	1	49.19	2, rue du Commerce	Lyon		69004	Federal Shipping	France
5	10851	RICAR	5	1998-01-26	1998-02-23	1998-02-02	1	160.55	Av. Copacabana, 267	Rio de Janeiro	RJ	02389-890	Federal Shipping	Brazil
6	10852	RATTC	8	1998-01-26	1998-02-09	1998-01-30	1	174.05	2817 Milton Dr.	Albuquerque	NM	87110	Federal Shipping	USA
7	10853	BLAUS	9	1998-01-27	1998-02-24	1998-02-03	2	53.83	Forsterstr. 57	Mannheim		68306	Federal Shipping	Germany
8	10854	ERNSH	3	1998-01-27	1998-02-24	1998-02-05	2	100.22	Kirchgasse 6	Graz		8010	Federal Shipping	Austria
9	10855	OLDWO	3	1998-01-27	1998-02-24	1998-02-04	1	170.97	2743 Bering St.	Anchorage	AK	99508	United Package	USA
10	10856	ANTON	3	1998-01-28	1998-02-25	1998-02-10	2	58.43	Mataderos 2312	México D.F.		05023	United Package	Mexico
11	10857	BERGS	8	1998-01-28	1998-02-25	1998-02-06	2	188.85	Berguvsv?gen 8	Lule?		S-958 22	United Package	Sweden
12	10858	LACOR	2	1998-01-29	1998-02-26	1998-02-03	1	52.51	67, avenue de l'Europe	Versailles		78000	United Package	France
13	10859	FRANK	1	1998-01-29	1998-02-26	1998-02-02	2	76.1	Berliner Platz 43	München		80805	United Package	Germany
14	10860	FRANR	3	1998-01-29	1998-02-26	1998-02-04	3	19.26	54, rue Royale	Nantes		44000	United Package	France
15	10861	WHITC	4	1998-01-30	1998-02-27	1998-02-17	2	14.93	1029 - 12th Ave. S.	Seattle	WA	98124	United Package	USA
16	10862	LEHMS	8	1998-01-30	1998-03-13	1998-02-02	2	53.23	Magazinweg 7	Frankfurt a.M.		60528	United Package	Germany
17	10863	HILAA	4	1998-02-02	1998-03-02	1998-02-17	2	30.26	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	5022	United Package	Venezuela
18	10864	AROUT	4	1998-02-02	1998-03-02	1998-02-09	2	3.04	Brook Farm Stratford St. Mary	Colchester	Essex	CO7 6JX	United Package	UK
19	10865	QUICK	2	1998-02-02	1998-02-16	1998-02-12	1	348.14	Taucherstra?e 10	Cunewalde		01307	United Package	Germany
20	10866	BERGS	5	1998-02-03	1998-03-03	1998-02-12	1	109.11	Berguvsv?gen 8	Lule?		S-958 22	United Package	Sweden
21	10867	LONEP	6	1998-02-03	1998-03-17	1998-02-11	1	1.93	89 Chiaroscuro Rd.	Portland	OR	97219	United Package	USA
22	10868	QUEEN	7	1998-02-04	1998-03-04	1998-02-23	2	191.27	Alameda dos Canários, 891	Sao Paulo	SP	05487-020	United Package	Brazil
23	10869	SEVES	5	1998-02-04	1998-03-04	1998-02-09	1	143.28	90 Wadhurst Rd.	London		OX15 4NB	United Package	UK
24	10870	WOLZA	5	1998-02-04	1998-03-04	1998-02-13	3	12.04	ul. Filtrowa 68	Warszawa		01-012	United Package	Poland
25	10871	BONAP	9	1998-02-05	1998-03-05	1998-02-10	2	112.27	12, rue des Bouchers	Marseille		13008	United Package	France
26	10872	GODOS	5	1998-02-05	1998-03-05	1998-02-09	2	175.32	C/ Romero, 33	Sevilla		41101	United Package	Spain
27	10873	WILMK	4	1998-02-06	1998-03-06	1998-02-09	1	0.82	Keskuskatu 45	Helsinki		21240	United Package	Finland
28	10874	GODOS	5	1998-02-06	1998-03-06	1998-02-11	2	19.58	C/ Romero, 33	Sevilla		41101	United Package	Spain

Script:

	A	B
1	=T**@c**("Orders.xlsx")	
2	for A1,300	=T("Orders"/#A2/".xlsx",A2)

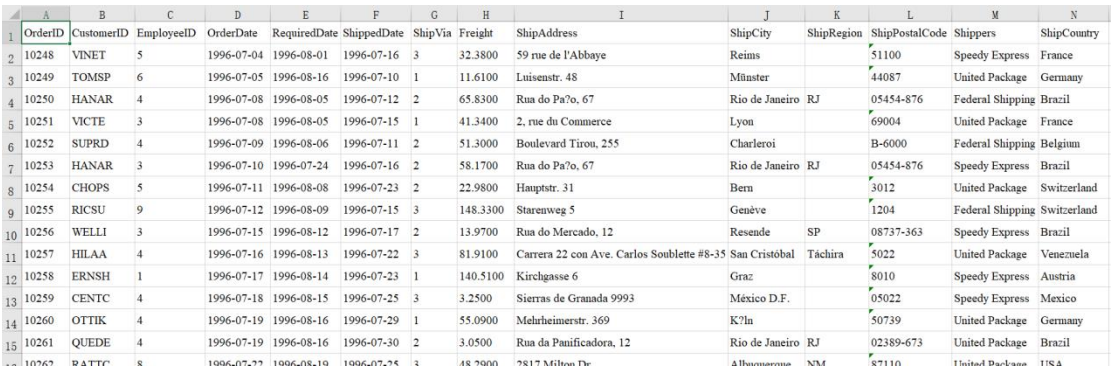
A1: Read the data of Orders.xlsx file as a cursor, the @c option means reading as a cursor.

A2: Loop through the data in A1, fetching 300 pieces of data each time.

B2: Take Orders + loop sequence number as the file name, and write the data in A2 to the file.

5.2 **Split by row - group by data - split into multiple Sheets**

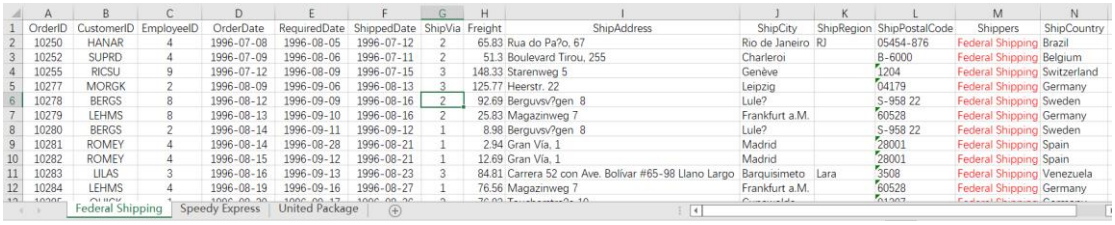
We have an order table file orders. xlsx, and part of the data is shown in the figure below. The first row contains the column titles, and the data start from the second row with one piece of data per row.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10248	VINET	5	1996-07-04	1996-08-01	1996-07-16	3	32.3800	59 rue de l'Abbaye	Reims		51100	Speedy Express	France
3	10249	TOMSP	6	1996-07-05	1996-08-16	1996-07-10	1	11.6100	Luisenstr. 48	Münster		44087	United Package	Germany
4	10250	HANAR	4	1996-07-08	1996-08-05	1996-07-12	2	65.8300	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Federal Shipping	Brazil
5	10251	VICTE	3	1996-07-08	1996-08-05	1996-07-15	1	41.3400	2, rue du Commerce	Lyon		69004	United Package	France
6	10252	SUPRD	4	1996-07-09	1996-08-06	1996-07-11	2	51.3000	Boulevard Tirou, 255	Charleroi		B-6000	Federal Shipping	Belgium
7	10253	HANAR	3	1996-07-10	1996-07-24	1996-07-16	2	58.1700	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Speedy Express	Brazil
8	10254	CHOPS	5	1996-07-11	1996-08-08	1996-07-23	2	22.9800	Hauptstr. 31	Bern		3012	United Package	Switzerland
9	10255	RICSU	9	1996-07-12	1996-08-09	1996-07-15	3	148.3300	Starenweg 5	Genève		1204	Federal Shipping	Switzerland
10	10256	WELLI	3	1996-07-15	1996-08-12	1996-07-17	2	13.9700	Rua do Mercado, 12	Resende	SP	08737-363	Speedy Express	Brazil
11	10257	HILAA	4	1996-07-16	1996-08-13	1996-07-22	3	81.9100	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	5022	United Package	Venezuela
12	10258	ERNSH	1	1996-07-17	1996-08-14	1996-07-23	1	140.5100	Kirchgasse 6	Graz		8010	Speedy Express	Austria
13	10259	CENTC	4	1996-07-18	1996-08-15	1996-07-25	3	3.2500	Sierras de Granada 9993	México D.F.		05022	Speedy Express	Mexico
14	10260	OTTIK	4	1996-07-19	1996-08-16	1996-07-29	1	55.0900	Mehrheimerstr. 369	K?ln		50739	United Package	Germany
15	10261	QUEDE	4	1996-07-19	1996-08-16	1996-07-30	2	3.0500	Rua da Panificadora, 12	Rio de Janeiro	RJ	02389-673	United Package	Brazil

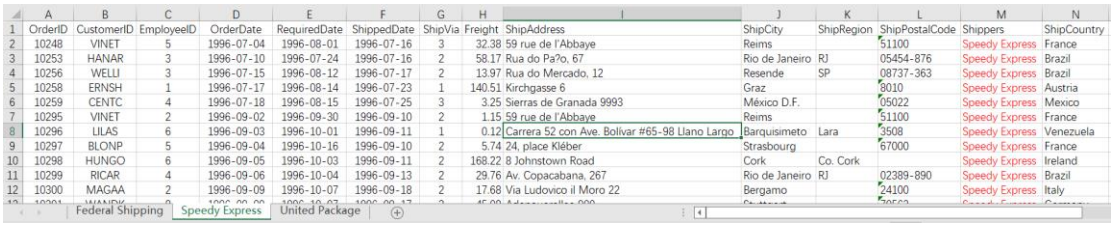
Now we want to group the data in this table by Shippers, and store each grouped data to one separate Sheet, with the Shippers name as the Sheet name. The split results are as follows:

Ordersm.xlsx:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10250	HANAR	4	1996-07-08	1996-08-05	1996-07-12	2	65.83	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Federal Shipping	Brazil
3	10252	SUPRD	4	1996-07-09	1996-08-06	1996-07-11	2	51.3	Boulevard Tirou, 255	Charleroi		B-6000	Federal Shipping	Belgium
4	10255	RICSU	9	1996-07-12	1996-08-09	1996-07-15	3	148.33	Starenweg 5	Genève		1204	Federal Shipping	Switzerland
5	10277	MORGK	2	1996-08-09	1996-09-06	1996-08-13	3	125.77	Heerstr. 22	Leipzig		04179	Federal Shipping	Germany
6	10278	BERGS	8	1996-08-12	1996-09-09	1996-08-16	2	92.69	Berguvsv?gen 8	Lule?		S-958 22	Federal Shipping	Sweden
7	10279	LEHMS	8	1996-08-13	1996-09-10	1996-08-16	2	25.83	Magazinweg 7	Frankfurt a.M.		60528	Federal Shipping	Germany
8	10280	BERGS	2	1996-08-14	1996-09-11	1996-09-12	1	8.98	Berguvsv?gen 8	Lule?		S-958 22	Federal Shipping	Sweden
9	10281	ROMEY	4	1996-08-14	1996-08-28	1996-08-21	1	2.94	Gran Vía, 1	Madrid		28001	Federal Shipping	Spain
10	10282	ROMEY	4	1996-08-15	1996-09-12	1996-08-21	1	12.69	Gran Vía, 1	Madrid		28001	Federal Shipping	Spain
11	10283	LILAS	3	1996-08-16	1996-09-13	1996-08-23	3	84.81	Carrera 52 con Ave. Bolívar #65-98 Llano Largo	Barquisimeto	Lara	3508	Federal Shipping	Venezuela
12	10284	LEHMS	4	1996-08-19	1996-09-16	1996-08-27	1	76.56	Magazinweg 7	Frankfurt a.M.		60528	Federal Shipping	Germany

Federal Shipping | Speedy Express | United Package



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10248	VINET	5	1996-07-04	1996-08-01	1996-07-16	3	32.38	59 rue de l'Abbaye	Reims		51100	Speedy Express	France
3	10253	HANAR	3	1996-07-10	1996-07-24	1996-07-16	2	58.17	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Speedy Express	Brazil
4	10256	WELLI	3	1996-07-15	1996-08-12	1996-07-17	2	13.97	Rua do Mercado, 12	Resende	SP	08737-363	Speedy Express	Brazil
5	10258	ERNSH	1	1996-07-17	1996-08-14	1996-07-23	1	140.51	Kirchgasse 6	Graz		8010	Speedy Express	Austria
6	10259	CENTC	4	1996-07-18	1996-08-15	1996-07-25	3	3.25	Sierras de Granada 9993	México D.F.		05022	Speedy Express	Mexico
7	10295	VINET	2	1996-09-02	1996-09-30	1996-09-10	2	1.15	59 rue de l'Abbaye	Reims		51100	Speedy Express	France
8	10296	LILAS	6	1996-09-03	1996-10-01	1996-09-11	1	0.12	Carrera 52 con Ave. Bolívar #65-98 Llano Largo	Barquisimeto	Lara	3508	Speedy Express	Venezuela
9	10297	BLONP	5	1996-09-04	1996-10-16	1996-09-10	2	5.74	24, place Kléber	Strasbourg		67000	Speedy Express	France
10	10298	HUNGO	6	1996-09-05	1996-10-03	1996-09-11	2	168.22	8 Johnstown Road	Cork	Co. Cork		Speedy Express	Ireland
11	10299	RICAR	4	1996-09-06	1996-10-04	1996-09-13	2	29.76	Av. Copacabana, 267	Rio de Janeiro	RJ	02389-890	Speedy Express	Brazil
12	10300	MAGAA	2	1996-09-09	1996-10-07	1996-09-18	2	17.68	Via Ludovico il Moro 22	Bergamo		24100	Speedy Express	Italy

Federal Shipping | Speedy Express | United Package

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10249	TOMSP	6	1996-07-05	1996-08-16	1996-07-10	1	11.61	Luisenstr. 48	Munster		44087	United Package	Germany
3	10251	VICTE	3	1996-07-08	1996-08-05	1996-07-15	1	41.34	2, rue du Commerce	Lyon		69004	United Package	France
4	10254	CHOPS	5	1996-07-11	1996-08-08	1996-07-23	2	22.98	Hauptstr. 31	Bern		3012	United Package	Switzerland
5	10257	HILAA	4	1996-07-16	1996-08-13	1996-07-22	3	81.91	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	5022	United Package	Venezuela
6	10260	OTTIK	4	1996-07-19	1996-08-16	1996-07-29	1	55.09	Mehrheimerstr. 369	K?ln		50739	United Package	Germany
7	10261	QUEDE	4	1996-07-19	1996-08-16	1996-07-30	2	3.05	Rua da Panificadora, 12	Rio de Janeiro	RJ	02389-673	United Package	Brazil
8	10262	RATTC	8	1996-07-22	1996-08-19	1996-07-25	3	48.29	2817 Milton Dr.	Albuquerque	NM	87110	United Package	USA
9	10263	ERNSH	9	1996-07-23	1996-08-20	1996-07-31	3	146.06	Kirchgasse 6	Graz		8010	United Package	Austria
10	10264	FOLKO	6	1996-07-24	1996-08-21	1996-08-23	3	3.67	?kergatan 24	Br?cke		S-844 67	United Package	Sweden
11	10265	BLONP	2	1996-07-25	1996-08-22	1996-08-12	1	55.28	24, place Kléber	Strasbourg		67000	United Package	France
12	10266	WARTH	3	1996-07-26	1996-09-06	1996-07-31	3	25.73	Torikatu 38	Oulu		90110	United Package	Finland

Federal Shipping | Speedy Express | United Package

Script:

	A	B
1	=T("orders.xlsx")	=A1.group(Shippers)
2	for B1	=file("Ordersm.xlsx").xlsexport@ta(A2;A2.Shippers)

A1: Read the data in orders.xlsx.

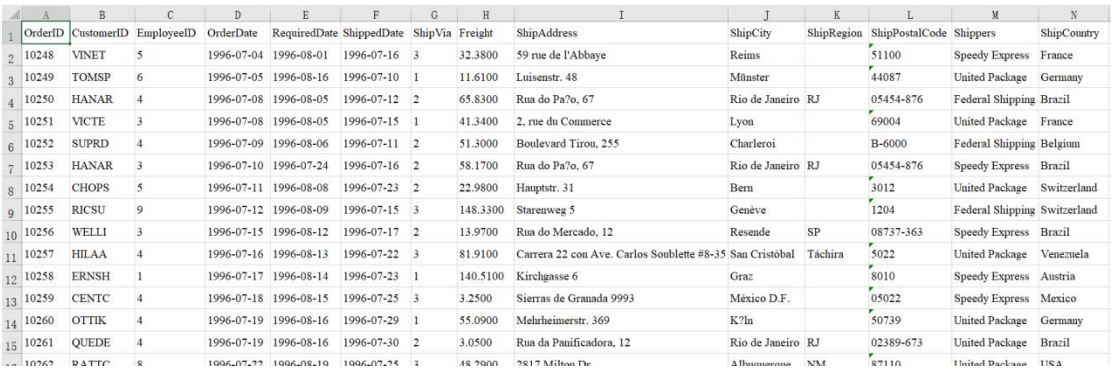
B1: Group by Shippers.

A2: Loop each Shippers group.

B2: Take the Shippers name as the Sheet name, and write the grouped data in A2 to the Sheet.

5.3 **Split by row - group by data - split into multiple files**

We have an order table file orders.xlsx, and part of the data is shown in the figure below. The first row contains the column titles, and the data start from the second row with one piece of data per row.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10248	VINET	5	1996-07-04	1996-08-01	1996-07-16	3	32.3800	59 rue de l'Abbaye	Reims		51100	Speedy Express	France
3	10249	TOMSP	6	1996-07-05	1996-08-16	1996-07-10	1	11.6100	Luisenstr. 48	Münster		44087	United Package	Germany
4	10250	HANAR	4	1996-07-08	1996-08-05	1996-07-12	2	65.8300	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Federal Shipping	Brazil
5	10251	VICTE	3	1996-07-08	1996-08-05	1996-07-15	1	41.3400	2, rue du Commerce	Lyon		69004	United Package	France
6	10252	SUPRD	4	1996-07-09	1996-08-06	1996-07-11	2	51.3000	Boulevard Tirou, 255	Charleroi		B-6000	Federal Shipping	Belgium
7	10253	HANAR	3	1996-07-10	1996-07-24	1996-07-16	2	58.1700	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Speedy Express	Brazil
8	10254	CHOPS	5	1996-07-11	1996-08-08	1996-07-23	2	22.9800	Hauptstr. 31	Bern		3012	United Package	Switzerland
9	10255	RICSU	9	1996-07-12	1996-08-09	1996-07-15	3	148.3300	Starenweg 5	Genève		1204	Federal Shipping	Switzerland
10	10256	WELLI	3	1996-07-15	1996-08-12	1996-07-17	2	13.9700	Rua do Mercado, 12	Resende	SP	08737-363	Speedy Express	Brazil
11	10257	HILAA	4	1996-07-16	1996-08-13	1996-07-22	3	81.9100	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	5022	United Package	Venezuela
12	10258	ERNSH	1	1996-07-17	1996-08-14	1996-07-23	1	140.5100	Kirchgasse 6	Graz		8010	Speedy Express	Austria
13	10259	CENTC	4	1996-07-18	1996-08-15	1996-07-25	3	3.2500	Sierras de Granada 9993	México D.F.		05022	Speedy Express	Mexico
14	10260	OTTIK	4	1996-07-19	1996-08-16	1996-07-29	1	55.0900	Mehrheimerstr. 369	K?ln		50739	United Package	Germany
15	10261	QUEDE	4	1996-07-19	1996-08-16	1996-07-30	2	3.0500	Rua da Panificadora, 12	Rio de Janeiro	RJ	02389-673	United Package	Brazil
16	10262	RATTC	8	1996-07-22	1996-08-19	1996-07-25	3	48.2900	2817 Milton Dr.	Albuquerque	NM	87110	United Package	USA

Now we want to group the data in this table by Shippers, and store each grouped data to one separate Excel file, with the Shippers name as the file name. The split results are as follows:

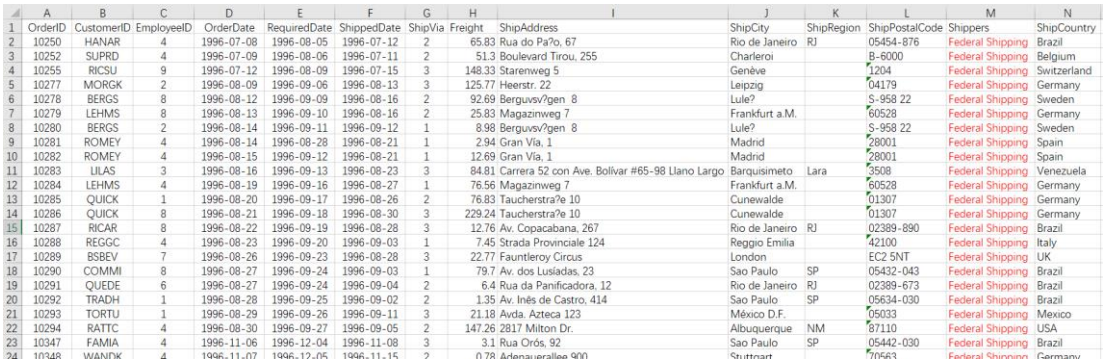
Speedy Express.xlsx:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10248	VINET	5	1996-07-04	1996-08-01	1996-07-16	3	32.38	59 rue de l'Abbaye	Reims		51100	Speedy Express	France
3	10253	HANAR	3	1996-07-10	1996-07-24	1996-07-16	2	58.17	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Speedy Express	Brazil
4	10256	WELLI	3	1996-07-15	1996-08-12	1996-07-17	2	13.97	Rua do Mercado, 12	Resende	SP	08737-363	Speedy Express	Brazil
5	10258	ERNSH	1	1996-07-17	1996-08-14	1996-07-23	1	140.51	Kirchgasse 6	Graz		8010	Speedy Express	Austria
6	10259	CENTC	4	1996-07-18	1996-08-15	1996-07-25	3	3.25	Sierras de Granada 9993	México D.F.		05022	Speedy Express	Mexico
7	10295	VINET	2	1996-09-02	1996-09-30	1996-09-10	2	1.15	59 rue de l'Abbaye	Reims		51100	Speedy Express	France
8	10296	LILAS	6	1996-09-03	1996-10-01	1996-09-11	1	0.12	Carrera 52 con Ave. Bolívar #65-98 Llano Largo	Barquisimeto	Lara	3508	Speedy Express	Venezuela
9	10297	BLONP	5	1996-09-04	1996-10-16	1996-09-10	2	5.74	24, place Kléber	Strasbourg		67000	Speedy Express	France
10	10298	HUNGO	6	1996-09-05	1996-10-03	1996-09-11	2	168.22	8 Johnstown Road	Cork	Co. Cork		Speedy Express	Ireland
11	10299	RICAR	4	1996-09-06	1996-10-04	1996-09-13	2	29.76	Av. Copacabana, 267	Rio de Janeiro	RJ	02389-890	Speedy Express	Brazil
12	10300	MAGAA	2	1996-09-09	1996-10-07	1996-09-18	2	17.68	Via Ludovico il Moro 22	Bergamo		24100	Speedy Express	Italy
13	10301	WANDK	8	1996-09-09	1996-10-07	1996-09-17	2	45.08	Adenauerallee 900	Stuttgart		70563	Speedy Express	Germany
14	10302	SUPRD	4	1996-09-10	1996-10-08	1996-10-09	2	6.27	Boulevard Tirou, 255	Charleroi		B-6000	Speedy Express	Belgium
15	10303	GODOS	7	1996-09-11	1996-10-09	1996-09-18	2	107.83	C/ Romero, 33	Sevilla		41101	Speedy Express	Spain
16	10304	TORTU	1	1996-09-12	1996-10-10	1996-09-17	2	63.79	Avda. Azteca 123	México D.F.		05033	Speedy Express	Mexico
17	10305	OLDWO	8	1996-09-13	1996-10-11	1996-10-09	3	257.62	2743 Bering St.	Anchorage	AK	99508	Speedy Express	USA
18	10306	ROMEY	1	1996-09-16	1996-10-14	1996-09-23	3	7.56	Gran Vía, 1	Madrid		28001	Speedy Express	Spain
19	10384	BERGS	3	1996-12-16	1997-01-13	1996-12-20	3	168.64	Berguvsv?gen 8	Lule?		S-958 22	Speedy Express	Sweden
20	10385	SPLIR	1	1996-12-17	1997-01-14	1996-12-23	2	30.96	P.O. Box 555	Lander	WY	82520	Speedy Express	USA
21	10386	FAMIA	9	1996-12-18	1997-01-01	1996-12-25	3	13.99	Rua Orós, 92	Sao Paulo	SP	05442-030	Speedy Express	Brazil
22	10387	SANTG	1	1996-12-18	1997-01-15	1996-12-20	2	93.63	Erling Skakkes gate 78	Stavern		4110	Speedy Express	Norway
23	10388	SEVES	2	1996-12-19	1997-01-16	1996-12-20	1	34.86	90 Wadhurst Rd.	London		OX15 4NB	Speedy Express	UK
24	10389	BOTTM	4	1996-12-20	1997-01-17	1996-12-24	2	47.42	23 Tsawassen Blvd.	Tsawassen	BC	T2F 8M4	Speedy Express	Canada
25	10390	ERNSH	6	1996-12-23	1997-01-20	1996-12-26	1	126.38	Kirchgasse 6	Graz		8010	Speedy Express	Austria
26	10391	DRACD	3	1996-12-23	1997-01-20	1996-12-31	3	5.45	Walserweg 21	Aachen		52066	Speedy Express	Germany
27	10392	PICCO	2	1996-12-24	1997-01-21	1997-01-01	3	122.46	Geislweg 14	Salzburg		5020	Speedy Express	Austria

United Package.xlsx:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10249	TOMSP	6	1996-07-05	1996-08-16	1996-07-10	1	11.61	Luisenstr. 48	Münster		44087	United Package	Germany
3	10251	VICTE	3	1996-07-08	1996-08-05	1996-07-15	1	41.34	2, rue du Commerce	Lyon		69004	United Package	France
4	10254	CHOPS	5	1996-07-11	1996-08-08	1996-07-23	2	22.98	Hauptstr. 31	Bern		3012	United Package	Switzerland
5	10257	HILAA	4	1996-07-16	1996-08-13	1996-07-22	3	81.91	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Táchira	5022	United Package	Venezuela
6	10260	OTTIK	4	1996-07-19	1996-08-16	1996-07-29	1	55.09	Mehrheimerstr. 369	K?ln		50739	United Package	Germany
7	10261	QUEDE	4	1996-07-19	1996-08-16	1996-07-30	2	3.05	Rua da Panificadora, 12	Rio de Janeiro	RJ	02389-673	United Package	Brazil
8	10262	RATTC	8	1996-07-22	1996-08-19	1996-07-25	3	48.29	2817 Milton Dr.	Albuquerque	NM	87110	United Package	USA
9	10263	ERNSH	9	1996-07-23	1996-08-20	1996-07-31	3	146.06	Kirchgasse 6	Graz		8010	United Package	Austria
10	10264	FOLKO	6	1996-07-24	1996-08-21	1996-08-23	3	3.67	?kergatan 24	Br?cke		S-844 67	United Package	Sweden
11	10265	BLONP	2	1996-07-25	1996-08-22	1996-08-12	1	55.28	24, place Kléber	Strasbourg		67000	United Package	France
12	10266	WARTH	3	1996-07-26	1996-09-06	1996-07-31	3	25.73	Torikatu 38	Oulu		90110	United Package	Finland
13	10267	FRANK	4	1996-07-29	1996-08-26	1996-08-06	1	208.58	Berliner Platz 43	München		80805	United Package	Germany
14	10268	GROSR	8	1996-07-30	1996-08-27	1996-08-02	3	66.29	5? Ave. Los Palos Grandes	Caracas	DF	1081	United Package	Venezuela
15	10269	WHITC	5	1996-07-31	1996-08-14	1996-08-09	1	4.56	1029 - 12th Ave. S.	Seattle	WA	98124	United Package	USA
16	10270	WARTH	1	1996-08-01	1996-08-29	1996-08-02	1	136.54	Torikatu 38	Oulu		90110	United Package	Finland
17	10271	SPLIR	6	1996-08-01	1996-08-29	1996-08-30	2	4.54	P.O. Box 555	Lander	WY	82520	United Package	USA
18	10272	RATTC	6	1996-08-02	1996-08-30	1996-08-06	2	98.03	2817 Milton Dr.	Albuquerque	NM	87110	United Package	USA
19	10273	QUICK	3	1996-08-05	1996-09-02	1996-08-12	3	76.07	Taucherstra?e 10	Cunewalde		01307	United Package	Germany
20	10274	VINET	6	1996-08-06	1996-09-03	1996-08-16	1	6.01	59 rue de l'Abbaye	Reims		51100	United Package	France
21	10275	MAGAA	1	1996-08-07	1996-09-04	1996-08-09	1	26.93	Via Ludovico il Moro 22	Bergamo		24100	United Package	Italy
22	10276	TORTU	8	1996-08-08	1996-08-22	1996-08-14	3	13.84	Avda. Azteca 123	México D.F.		05033	United Package	Mexico
23	10307	LONEP	2	1996-09-17	1996-10-15	1996-09-25	2	0.56	89 Chiaroscuro Rd.	Portland	OR	97219	United Package	USA
24	10308	ANATR	7	1996-09-18	1996-10-16	1996-09-24	3	1.61	Avda. de la Constitución 2222	México D.F.		05021	United Package	Mexico

Federal Shipping.xlsx:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Shippers	ShipCountry
2	10250	HANAR	4	1996-07-08	1996-08-05	1996-07-12	2	65.83	Rua do Pa?o, 67	Rio de Janeiro	RJ	05454-876	Federal Shipping	Brazil
3	10252	SUPRD	4	1996-07-09	1996-08-06	1996-07-11	2	51.3	Boulevard Tirou, 255	Charleroi		B-6000	Federal Shipping	Belgium
4	10255	RICSU	9	1996-07-12	1996-08-09	1996-07-15	3	148.33	Starenweg 5	Genève		1204	Federal Shipping	Switzerland
5	10277	MORGK	2	1996-08-09	1996-09-06	1996-08-13	3	125.77	Heerstr. 22	Leipzig		04179	Federal Shipping	Germany
6	10278	BERGS	8	1996-08-12	1996-09-09	1996-08-16	2	92.69	Berguvsv?gen 8	Lule?		S-958 22	Federal Shipping	Sweden
7	10279	LEHMS	8	1996-08-13	1996-09-10	1996-08-16	2	25.83	Magazinweg 7	Frankfurt a.M.		60528	Federal Shipping	Germany
8	10280	BERGS	2	1996-08-14	1996-09-11	1996-09-12	1	8.98	Berguvsv?gen 8	Lule?		S-958 22	Federal Shipping	Sweden
9	10281	ROMEY	4	1996-08-14	1996-08-28	1996-08-21	1	2.94	Gran Vía, 1	Madrid		28001	Federal Shipping	Spain
10	10282	ROMEY	4	1996-08-15	1996-09-12	1996-08-21	1	12.69	Gran Vía, 1	Madrid		28001	Federal Shipping	Spain
11	10283	LILAS	3	1996-08-16	1996-09-13	1996-08-23	3	84.81	Carrera 52 con Ave. Bolívar #65-98 Llano Largo	Barquisimeto	Lara	3508	Federal Shipping	Venezuela
12	10284	LEHMS	4	1996-08-19	1996-09-16	1996-08-27	1	76.56	Magazinweg 7	Frankfurt a.M.		60528	Federal Shipping	Germany
13	10285	QUICK	1	1996-08-20	1996-09-17	1996-08-26	2	76.83	Taucherstra?e 10	Cunewalde		01307	Federal Shipping	Germany
14	10286	QUICK	8	1996-08-21	1996-09-18	1996-08-30	3	229.24	Taucherstra?e 10	Cunewalde		01307	Federal Shipping	Germany
15	10287	RICAR	8	1996-08-22	1996-09-19	1996-08-28	3	12.76	Av. Copacabana, 267	Rio de Janeiro	RJ	02389-890	Federal Shipping	Brazil
16	10288	REGGC	4	1996-08-23	1996-09-20	1996-09-03	1	7.45	Strada Provinciale 124	Reggio Emilia		42100	Federal Shipping	Italy
17	10289	BSBEV	7	1996-08-26	1996-09-23	1996-08-28	3	22.77	Fauntleroy Circus	London		EC2 5NT	Federal Shipping	UK
18	10290	COMMI	8	1996-08-27	1996-09-24	1996-09-03	1	79.7	Av. dos Lusíadas, 23	Sao Paulo	SP	05432-043	Federal Shipping	Brazil
19	10291	QUEDE	6	1996-08-27	1996-09-24	1996-09-04	2	6.4	Rua da Panificadora, 12	Rio de Janeiro	RJ	02389-673	Federal Shipping	Brazil
20	10292	TRADH	1	1996-08-28	1996-09-25	1996-09-02	2	1.35	Av. Inês de Castro, 414	Sao Paulo	SP	05634-030	Federal Shipping	Brazil
21	10293	TORTU	1	1996-08-29	1996-09-26	1996-09-11	3	21.18	Avda. Azteca 123	México D.F.		05033	Federal Shipping	Mexico
22	10294	RATTC	4	1996-08-30	1996-09-27	1996-09-05	2	147.26	2817 Milton Dr.	Albuquerque	NM	87110	Federal Shipping	USA
23	10347	FAMIA	4	1996-11-06	1996-12-04	1996-11-08	3	3.1	Rua Orós, 92	Sao Paulo	SP	05442-030	Federal Shipping	Brazil
24	10348	WANDK	4	1996-11-07	1996-12-05	1996-11-15	2	0.78	Adenauerallee 900	Stuttgart		70563	Federal Shipping	Germany

Script:

	A	B
1	=T("orders.xlsx")	=A1.group(Shippers)
2	for B1	=T(A2.Shippers+".xlsx",A2)

A1: Read the data in orders.xlsx.

B1: Group by Shippers.

A2: Loop each Shippers group.

B2: Take the Shippers name as the file name, and write the grouped data in A2 to the file.

5.4 **Split by row - segment by data (by filtering condition)**

There is an order detail data file OrderDetailExtended.xlsx as follows:

	A	B	C	D	E	F
1	OrderID	ProductID	UnitPrice	Quantity	Discount	ExtendedPrice
2	10248	11	14.0000	12	0	168.0000
3	10248	42	9.8000	10	0	98.0000
4	10248	72	34.8000	5	0	174.0000
5	10249	14	18.6000	9	0	167.4000
6	10249	51	42.4000	40	0	1696.0000
7	10250	41	7.7000	10	0	77.0000
8	10250	51	42.4000	35	0.15	1261.4000
9	10250	65	16.8000	15	0.15	214.2000
10	10251	22	16.8000	6	0.05	95.7600
11	10251	57	15.6000	15	0.05	222.3000
12	10251	65	16.8000	20	0	336.0000
13	10252	20	64.8000	40	0.05	2462.4000
14	10252	33	2.0000	25	0.05	47.5000
15	10252	60	27.2000	40	0	1088.0000
16	10253	31	10.0000	20	0	200.0000

Now we want to split the data of this table into three segments by the value in the column of ExtendedPrice (<500, 500-2000, >2000), and save the segmented data as three Excel files. The split results are as follows:

lt500.xlsx:

	A	B	C	D	E	F
1	OrderID	ProductID	UnitPrice	Quantity	Discount	ExtendedPrice
2	10248	11	14	12	0	168
3	10248	42	9.8	10	0	98
4	10248	72	34.8	5	0	174
5	10249	14	18.6	9	0	167.4
6	10250	41	7.7	10	0	77
7	10250	65	16.8	15	0.15	214.2
8	10251	22	16.8	6	0.05	95.76
9	10251	57	15.6	15	0.05	222.3
10	10251	65	16.8	20	0	336
11	10252	33	2	25	0.05	47.5
12	10253	31	10	20	0	200
13	10254	24	3.6	15	0.15	45.9
14	10254	55	19.2	21	0.15	342.72
15	10254	74	8	21	0	168
16	10255	2	15.2	20	0	304

Sheet1

Mt2000.xlsx:

	A	B	C	D	E	F
1	OrderID	ProductID	UnitPrice	Quantity	Discount	ExtendedPrice
2	10252	20	64.8	40	0.05	2462.4
3	10267	59	44	70	0.15	2618
4	10305	29	99	25	0.1	2227.5
5	10316	62	39.4	70	0	2758
6	10324	63	35.1	80	0.15	2386.8
7	10329	38	210.8	20	0.05	4005.2
8	10339	17	31.2	70	0.05	2074.8
9	10345	8	32	70	0	2240
10	10351	38	210.8	20	0.05	4005.2
11	10353	38	210.8	50	0.2	8432
12	10359	60	27.2	80	0.05	2067.2
13	10360	29	99	35	0	3465
14	10360	38	210.8	10	0	2108
15	10372	38	210.8	40	0.25	6324
16	10398	55	19.2	120	0.1	2073.6
17	10400	29	99	21	0	2079

Sheet1

500-2000.xlsx:

	A	B	C	D	E	F
1	OrderID	ProductID	UnitPrice	Quantity	Discount	ExtendedPrice
2	10249	51	42.4	40	0	1696
3	10250	51	42.4	35	0.15	1261.4
4	10252	60	27.2	40	0	1088
5	10253	39	14.4	42	0	604.8
6	10253	49	16	40	0	640
7	10255	59	44	30	0	1320
8	10257	27	35.1	25	0	877.5
9	10258	2	15.2	50	0.2	608
10	10258	5	17	65	0.2	884
11	10260	57	15.6	50	0	780
12	10263	16	13.9	60	0.25	625.5
13	10263	30	20.7	60	0.25	931.5
14	10264	2	15.2	35	0	532
15	10265	17	31.2	30	0	936
16	10267	40	14.7	50	0	735
17	10268	29	99	10	0	990

Sheet1

Script:

	A	B
1	=T("OrderDetailsExtended.xlsx")	
2	=A1.group(if(ExtendedPrice<500:"lt500.xlsx",ExtendedPrice>2000:"mt2000.xlsx";"500-2000.xlsx"):fileName;~:data)	
3	for A2	=T(A3.fileName,A3.data)

A1: Read the data in OrderDetailsExtended.xlsx.

A2: Divide A1 into three groups by the value in the column of ExtendedPrice (i.e., value less than 500, value greater than 2000, and value between 500 and 2000), and take such values as the name of corresponding file.

A3: Loop by A2.

B3: Write the group data in each row to the corresponding file.

5.5 Split by row - generate one card per row

There is an employee information table Employee.xlsx, and part of the data is as follows:

	A	B	C	D	E	F	G	H	I
1	ID	FirstName	LastName	Gender	Title	Birthday	HomePhone	PostalCode	Address
2	1	Nancy	Davolio	Female	Sales Representative	1948-12-08 00:00:00	(206) 555-9857	98122	507 - 20th Ave. E.Apt. 2A
3	2	Andrew	Fuller	Male	Vice President, Sales	1952-02-19 00:00:00	(206) 555-9482	98401	908 W. Capital Way
4	3	Janet	Leverling	Female	Sales Representative	1963-08-30 00:00:00	(206) 555-3412	98033	722 Moss Bay Blvd.
5	4	Margaret	Peacock	Female	Sales Representative	1937-09-19 00:00:00	(206) 555-8122	98052	4110 Old Redmond Rd.

Now we want to split the data to generate one card-style table for each employee in the following format, and the empty card-style file is named Card.xlsx:

	A	B	C	D	E
1	ID		Gender		
2	FirstName		LastName		
3	Title				
4	Birthday				
5	HomePhone				
6	Address				
7	PostalCode				

The generated card-style file is named after employee name, such as:

Andrew.Fuller.xlsx：

	A	B	C	D	E
1	ID	2	Gender	Male	
2	FirstName	Andrew	LastName	Fuller	
3	Title	Vice President, Sales			
4	Birthday	1952-02-19			
5	HomePhone	(206) 555-9482			
6	Address	908 W. Capital Way			
7	PostalCode				

Janet.Leverling.xlsx：

	A	B	C	D	E
1	ID	3	Gender	Female	
2	FirstName	Janet	LastName	Leverling	
3	Title	Sales Representative			
4	Birthday	1963-08-30			
5	HomePhone	(206) 555-3412			
6	Address	722 Moss Bay Blvd.			
7	PostalCode				

Margaret.Peacock.xlsx：

	A	B	C	D	E
1	ID	4	Gender	Female	
2	FirstName	Margaret	LastName	Peacock	
3	Title	Sales Representative			
4	Birthday	1937-09-19			
5	HomePhone	(206) 555-8122			
6	Address	4110 Old Redmond Rd.			
7	PostalCode				

Nancy.Davolio.xlsx：

	A	B	C	D	E
1	ID	1	Gender	Female	
2	FirstName	Nancy	LastName	Davolio	
3	Title	Sales Representative			
4	Birthday	1948-12-08			
5	HomePhone	(206) 555-9857			
6	Address	507 - 20th Ave. E.			
7	PostalCode				

…

Script:

	A	B	C
1	=T("Employee.xlsx")		[B1,B2,D2,D1,B3,B4,B5,B7,B6]
2	for A1	=file("Card.xlsx").xlsopen()	
3		for C1	=B2.xlscell(B3,1;A2.field(#B3))
4		=file(A2.FirstName+"."+A2.LastName+".xlsx").xlswrite(B2)	

A1: Read the data of employee information table.

C1: Define the sequence of cell names where each column of the employee information table is to be written on the card.

A2: Loop through every employee.

B2: Open the empty card-style file as an Excel object.

B3: Loop through each cell to be filled in.

C3: Fill the content of the corresponding column number in the current employee A2 into the current to-be-written cell.

B4: Store the Excel object of B2 to the Excel file named after the employee's name.

5.6 Split by row - split multiple cards to make one card generate one file

There is a multi-card style table Cards.xlsx as follows:

	A	B	C	D	E
1	ID	1	Gender	Female	
2	FirstName	Nancy	LastName	Davolio	
3	Title	Sales Representative			
4	Birthday	17875			
5	HomePhone	(206) 555-9857			
6	Address	507 - 20th Ave. E.			
7	PostalCode	Apt. 2A			
8					
9	ID	4	Gender	Female	
10	FirstName	Margaret	LastName	Peacock	
11	Title	Sales Representative			
12	Birthday	13777			
13	HomePhone	(206) 555-8122			
14	Address	4110 Old Redmond Rd.			
15	PostalCode	98052			
16					
17	ID	3	Gender	Female	
18	FirstName	Janet	LastName	Leverling	
19	Title	Sales Representative			
20	Birthday	23253			
21	HomePhone	(206) 555-3412			
22	Address	722 Moss Bay Blvd.			

Sheet1

Now we need to split it to make one card generate one file, with person's name as the file name. The splitting results are as follows:

Andrew.Fuller.xlsx：

	A	B	C	D	E
1	ID	2	Gender	Male	
2	FirstName	Andrew	LastName	Fuller	
3	Title	Vice President, Sales			
4	Birthday	19043			
5	HomePhone	(206) 555-9482			
6	Address	908 W. Capital Way			
7	PostalCode	98401			

Janet.Leverling.xlsx：

	A	B	C	D	E
1	ID	3	Gender	Female	
2	FirstName	Janet	LastName	Leverling	
3	Title	Sales Representative			
4	Birthday	23253			
5	HomePhone	(206) 555-3412			
6	Address	722 Moss Bay Blvd.			
7	PostalCode	98033			

Margaret.Peacock.xlsx：

	A	B	C	D	E
1	ID	4	Gender	Female	
2	FirstName	Margaret	LastName	Peacock	
3	Title	Sales Representative			
4	Birthday	13777			
5	HomePhone	(206) 555-8122			
6	Address	4110 Old Redmond Rd.			
7	PostalCode	98052			

Nancy.Davolio.xlsx：

	A	B	C	D	E
1	ID	1	Gender	Female	
2	FirstName	Nancy	LastName	Davolio	
3	Title	Sales Representative			
4	Birthday	17875			
5	HomePhone	(206) 555-9857			
6	Address	507 - 20th Ave. E.			
7	PostalCode	Apt. 2A			

…

Operation steps:

1. Create an empty card-style file Card.xlsx, as the initial file format:

	A	B	C	D	E
1	ID		Gender		
2	FirstName		LastName		
3	Title				
4	Birthday				
5	HomePhone				
6	Address				
7	PostalCode				

2. Script:

	A	B	C
1	=file("Card.xlsx").xlsopen()		
2	=file("Cards.xlsx").xlsopen()		
3	[B,D,B,D,B,B,B,B]	[1,1,2,2,3,4,5,6,7]	
4	=A3.(~/B3(#))		
5	for	=A3.(~/B3(#)).(A2.xlscell(~))	
6		if B5(1)==""	break
7		for A4	=A1.xlscell(B7,1;B5(#B7))
8		=file(B5(3)+"."+B5(4)+".xlsx").xlswrite(A1)	
9		>B3=B3.(~+8)	

A1: Open the empty card-style template Card.xlsx.

A2: Open the multi-card style table Cards.xlsx.

A3: List the column number of the cells to be read.

B3: List the row number of the cells to be read.

A4: Piece together the cell names of the first card.

A5: Loop.

B5: Piece together the cell names of the current card, and read the data in it.

B6: Exit loop if cards have been read.

B7: Loop through A4 and write the data in B5 into A1 one by one.

B8: Write A1 to the file named after the current card.

B9: Plus 8 after row number, because one card has 8 rows.

5.7 Format conversion - split tables with primary-sub relationship into cards

There are two tables with primary-sub relationship as follows:

Orders.xlsx：

	A	B	C	D	E	F
1	OrderID	OrderDate	CustomerID	EmployeeID	Consignee	ShipAddress
2	10248	1996-07-04	Vins et alcools Chevalier	Steven.Buchanan	Wolski Zajazd	59 rue de l'Abbaye
3	10249	1996-07-05	Toms Spezialit?ten	Michael.Suyama	Ana Trujillo Emparedados y helados	Luisenstr. 48
4	10250	1996-07-08	Hanari Carnes	Margaret.Peacock	Antonio Moreno Taquería	Rua do Pa?o, 67

OrderDetails.xlsx：

	A	B	C	D	E	F	G
1	OrderID	ProductID	ProductName	UnitPrice	Quantity	Discount	ExtendedPrice
2	10248	11	Queso Cabrales	14	12	0	168
3	10248	42	Singaporean Hokkien Fried Mee	9.8	10	0	98
4	10248	72	Mozzarella di Giovanni	34.8	5	0	174
5	10249	14	Tofu	18.6	9	0	167.4
6	10249	51	Manjimup Dried Apples	42.4	40	0	1696
7	10250	41	Jack's New England Clam Chowder	7.7	10	0	77
8	10250	51	Manjimup Dried Apples	42.4	35	0.15	1261.4
9	10250	65	Louisiana Fiery Hot Pepper Sauce	16.8	15	0.15	214.2

Now we need to associate them by OrderID, and generate one card with primary-sub relationship for each order, with OrderID as the file name, as shown below:

Orders10248.xlsx:

	A	B	C	D	E	F
1	OrderID	10248		OrderDate	1996-07-04 00:00:00	
2	CustomerID	Vins et alcools Chevalier		EmployeeID	Steven.Buchanan	
3	Consignee	Wolski Zajazd				
4	ShipAddress	59 rue de l'Abbaye				
6	ProductID	ProductName	UnitPrice	Quantity	Discount	ExtendedPrice
7	11	Queso Cabrales	14.0000	12	0	168.0000
8	42	Singaporean Hokkien Fried Mee	9.8000	10	0	98.0000
9	72	Mozzarella di Giovanni	34.8000	5	0	174.0000

Orders10249.xlsx:

	A	B	C	D	E	F
1	OrderID	10249		OrderDate	1996-07-05 00:00:00	
2	CustomerID	Toms Spezialit?ten		EmployeeID	Michael.Suyama	
3	Consignee	Ana Trujillo Emparedados y helados				
4	ShipAddress	Luisenstr. 48				
6	ProductID	ProductName	UnitPrice	Quantity	Discount	ExtendedPrice
7	14	Tofu	18.6000	9	0	167.4000
8	51	Manjimup Dried Apples	42.4000	40	0	1696.0000

Orders10250.xlsx:

	A	B	C	D	E	F
1	OrderID	10250		OrderDate	1996-07-08 00:00:00	
2	CustomerID	Hanari Carnes		EmployeeID	Margaret.Peacock	
3	Consignee	Antonio Moreno Taqueria				
4	ShipAddress	Rua do Pa?o, 67				
6	ProductID	ProductName	UnitPrice	Quantity	Discount	ExtendedPrice
7	41	Jack's New England Clam Chowder	7.7000	10	0	77.0000
8	51	Manjimup Dried Apples	42.4000	35	0.15	1261.4000
9	65	Louisiana Fiery Hot Pepper Sauce	16.8000	15	0.15	214.2000

Operation steps:

1. Create an empty card-style template.

Order.xlsx

	A	B	C	D	E	F
1	OrderID			OrderDate		
2	CustomerID			EmployeeID		
3	Consignee					
4	ShipAddress					
6	ProductID	ProductName	UnitPrice	Quantity	Discount	ExtendedPrice
7						
8						
9						
10						

2. Script:

	A	B
1	=T("Orders/Orders.xlsx")	
2	=T("Orders/OrderDetails.xlsx")	
3	=A2.align@a(A1:OrderID,OrderID)	
4	=file("Orders/Order.xlsx").xlsopen()	
5	=A2.alter(;OrderID)	
6	for A1	>A4.xlscell("B1",1;A6.OrderID),A4.xlscell("E1",1;A6.OrderDate),A4.xlscell("B2",1;A6.CustomerID), A4.xlscell("E2",1;A6.EmployeeID),A4.xlscell("B3",1;A6.Consignee),A4.xlscell("B4",1;A6.ShipAddress)
7		=A4.xlscell("A7",1;A3(#A6))
8		=file("Orders/Order"+A6.OrderID+".xlsx").xlswrite(A4)

A1: Read the data of Orders/Orders.xlsx.

A2: Read the data of Orders/OrderDetails.xlsx.

A3: Align A2 to A1, @a means many-to-one, and the associated column is OrderID.

A4: Open the empty template file Orders/Order.xlsx.

A5: Delete the column OrderID from A2, because this column does not need to be displayed in the sub table when outputting.

A6: Loop by primary table A1.

B6: Write the primary table data to the corresponding position of the A4 template.

B7: Write the sub-table data at the corresponding position in A3 to the cell starting with A7 of A4 template.

B8: Output the final generated primary-sub card-style table to the file named with the order number.

5.8 Split by column - by column - take column name as file name

There is a file Amount.xlsx as shown below:

	A	B	C	D
1	Name	Apple	Bread	Pork
2	Peter	6	29	7
3	Mark	3	59	11
4	Alice	7	140	23
5	Lily	8	120	25
6	John	9	240	41
7	Cindy	4	19	3
8	Maggie	5	38	7
9	Leon	2	29	8

This is an amount summary table for certain products purchased by some key customers. Now we need to split it by product, and save each product as a file with the product name as the file name and Amount as the column name. The split results are as follows:

Apple.xlsx：

	A	B
1	Name	Amount
2	Peter	6
3	Mark	3
4	Alice	7
5	Lily	8
6	John	9
7	Cindy	4
8	Maggie	5
9	Leon	2

Bread.xlsx：

	A	B
1	Name	Amount
2	Peter	29
3	Mark	59
4	Alice	140
5	Lily	120
6	John	240
7	Cindy	19
8	Maggie	38
9	Leon	29

Pork.xlsx：

	A	B
1	Name	Amount
2	Peter	7
3	Mark	11
4	Alice	23
5	Lily	25
6	John	41
7	Cindy	3
8	Maggie	7
9	Leon	8

Script:

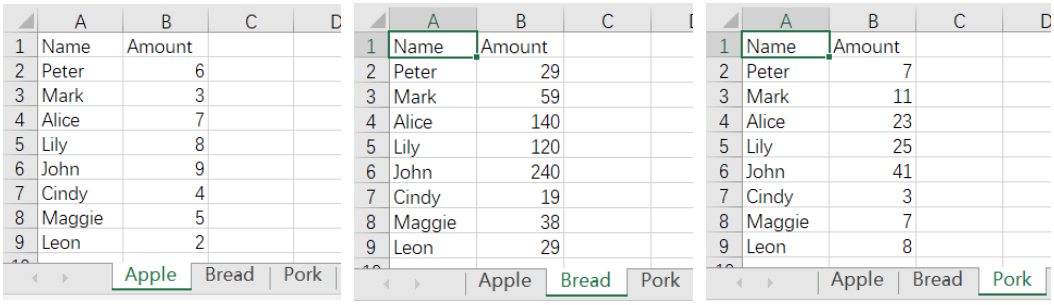
	A	B
1	=T("Amount.xlsx")	=A1.fname()\"Name"
2	for B1	=A1.new(Name,${A2}:Amount)
3		=T(A2+".xlsx",B2)

5.9 Split by column - by column - take column name as Sheet name

There is a file Amount.xlsx as shown below:

	A	B	C	D
1	Name	Apple	Bread	Pork
2	Peter	6	29	7
3	Mark	3	59	11
4	Alice	7	140	23
5	Lily	8	120	25
6	John	9	240	41
7	Cindy	4	19	3
8	Maggie	5	38	7
9	Leon	2	29	8

This is an amount summary table for certain products purchased by some key customers. Now we need to split it by product, and save each product to a Sheet with the product name as the Sheet name and Amount as the column name. The split results are as follows:

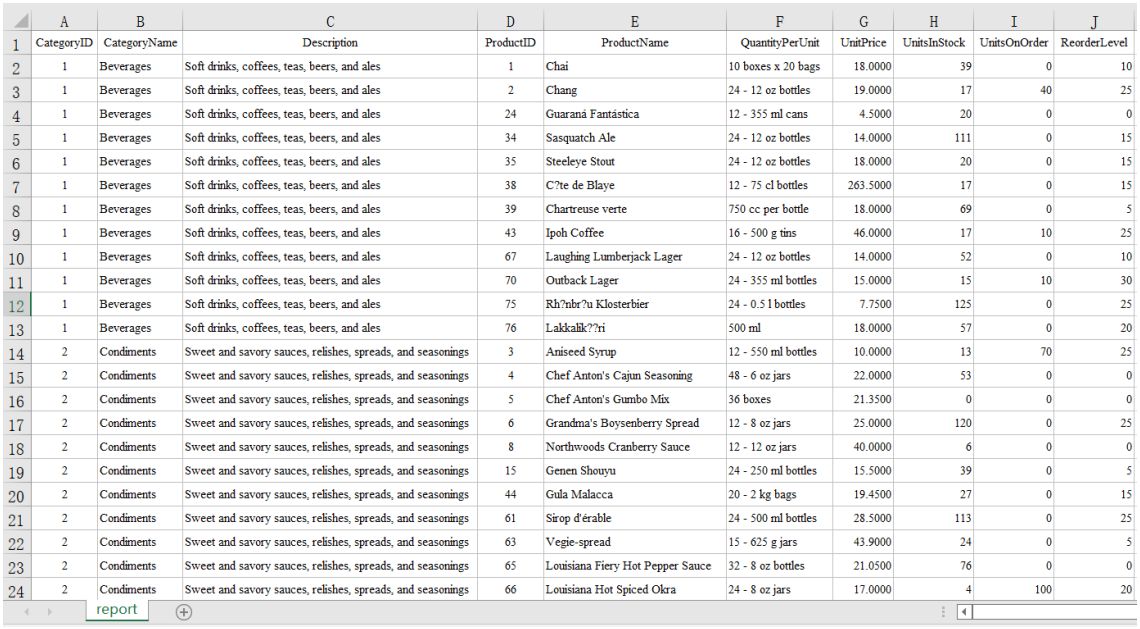


Script:

	A	B
1	=T("Amount.xlsx")	=A1.fname()\"Name"
2	for B1	=A1.new(Name,${A2}:Amount)
3		=file("Amounts.xlsx").xlsexport@ta(B2;A2)

5.10 **Split by column - merge duplicate rows after splitting**

There is a product data table ProductCategories.xls:



	A	B	C	D	E	F	G	H	I	J
1	CategoryID	CategoryName	Description	ProductID	ProductName	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel
2	1	Beverages	Soft drinks, coffees, teas, beers, and ales	1	Chai	10 boxes x 20 bags	18.0000	39	0	10
3	1	Beverages	Soft drinks, coffees, teas, beers, and ales	2	Chang	24 - 12 oz bottles	19.0000	17	40	25
4	1	Beverages	Soft drinks, coffees, teas, beers, and ales	24	Guaraná Fantástica	12 - 355 ml cans	4.5000	20	0	0
5	1	Beverages	Soft drinks, coffees, teas, beers, and ales	34	Sasquatch Ale	24 - 12 oz bottles	14.0000	111	0	15
6	1	Beverages	Soft drinks, coffees, teas, beers, and ales	35	Steeleye Stout	24 - 12 oz bottles	18.0000	20	0	15
7	1	Beverages	Soft drinks, coffees, teas, beers, and ales	38	C?te de Blaye	12 - 75 cl bottles	263.5000	17	0	15
8	1	Beverages	Soft drinks, coffees, teas, beers, and ales	39	Chartreuse verte	750 cc per bottle	18.0000	69	0	5
9	1	Beverages	Soft drinks, coffees, teas, beers, and ales	43	Ipoh Coffee	16 - 500 g tins	46.0000	17	10	25
10	1	Beverages	Soft drinks, coffees, teas, beers, and ales	67	Laughing Lumberjack Lager	24 - 12 oz bottles	14.0000	52	0	10
11	1	Beverages	Soft drinks, coffees, teas, beers, and ales	70	Outback Lager	24 - 355 ml bottles	15.0000	15	10	30
12	1	Beverages	Soft drinks, coffees, teas, beers, and ales	75	Rh?nbr?u Klosterbier	24 - 0.5 l bottles	7.7500	125	0	25
13	1	Beverages	Soft drinks, coffees, teas, beers, and ales	76	Lakkalik??ri	500 ml	18.0000	57	0	20
14	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	3	Aniseed Syrup	12 - 550 ml bottles	10.0000	13	70	25
15	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	22.0000	53	0	0
16	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	5	Chef Anton's Gumbo Mix	36 boxes	21.3500	0	0	0
17	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	6	Grandma's Boysenberry Spread	12 - 8 oz jars	25.0000	120	0	25
18	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	8	Northwoods Cranberry Sauce	12 - 12 oz jars	40.0000	6	0	0
19	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	15	Genen Shouyu	24 - 250 ml bottles	15.5000	39	0	5
20	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	44	Gula Malacca	20 - 2 kg bags	19.4500	27	0	15
21	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	61	Sirop d'érable	24 - 500 ml bottles	28.5000	113	0	25
22	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	63	Vegie-spread	15 - 625 g jars	43.9000	24	0	5
23	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	65	Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	21.0500	76	0	0
24	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	66	Louisiana Hot Spiced Okra	24 - 8 oz jars	17.0000	4	100	20

report

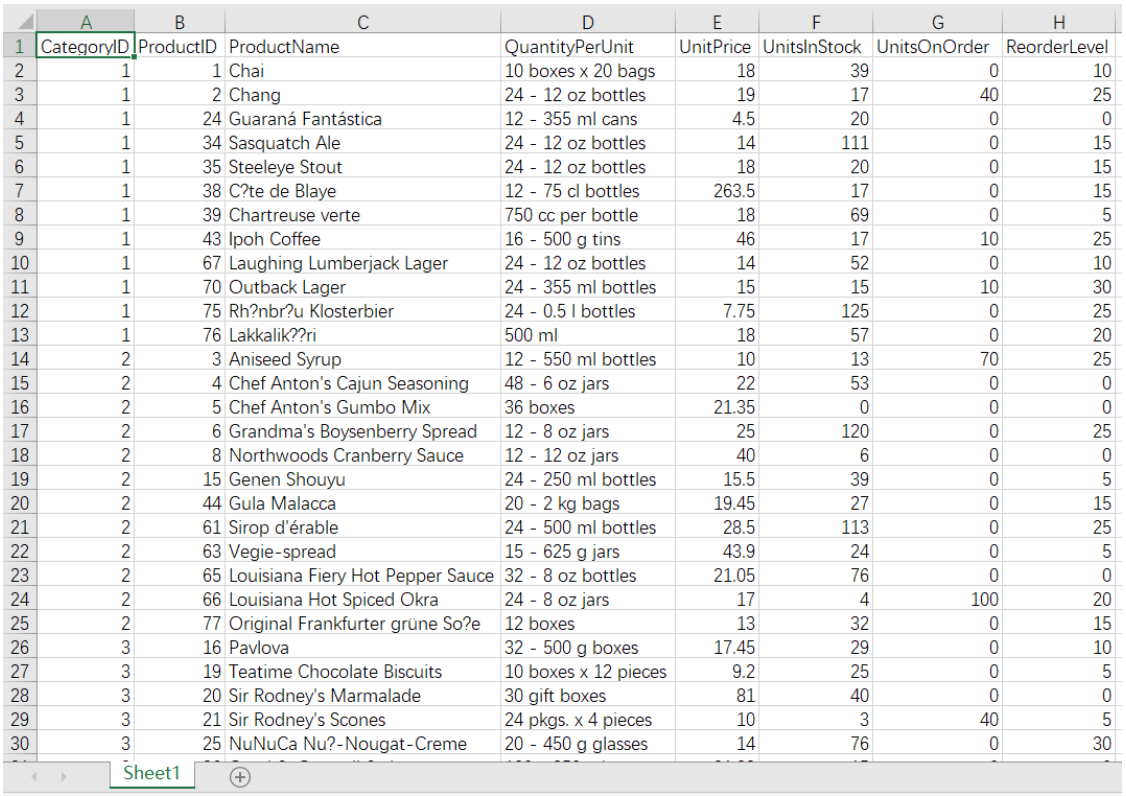
This table contains the category information of products, such as CategoryID, CategoryName, Description, etc. Now we need to separate the category information columns to form one Categories table, and take the remaining product columns as the Products table. Since there are many products under one category field, and many of them are duplicate after splitting, deduplication needs to be performed.

The results after splitting are shown as below:

Categories.xlsx：

	A	B	C
1	CategoryID	CategoryName	Description
2	1	Beverages	Soft drinks, coffees, teas, beers, and ales
3	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
4	3	Confections	Desserts, candies, and sweet breads
5	4	Dairy Products	Cheeses
6	5	Grains/Cereals	Breads, crackers, pasta, and cereal
7	6	Meat/Poultry	Prepared meats
8	7	Produce	Dried fruit and bean curd
9	8	Seafood	Seaweed and fish

Products.xlsx：



	A	B	C	D	E	F	G	H
1	CategoryID	ProductID	ProductName	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel
2	1	1	Chai	10 boxes x 20 bags	18	39	0	10
3	1	2	Chang	24 - 12 oz bottles	19	17	40	25
4	1	24	Guaraná Fantástica	12 - 355 ml cans	4.5	20	0	0
5	1	34	Sasquatch Ale	24 - 12 oz bottles	14	111	0	15
6	1	35	Steeleye Stout	24 - 12 oz bottles	18	20	0	15
7	1	38	C?te de Blaye	12 - 75 cl bottles	263.5	17	0	15
8	1	39	Chartreuse verte	750 cc per bottle	18	69	0	5
9	1	43	Ipoh Coffee	16 - 500 g tins	46	17	10	25
10	1	67	Laughing Lumberjack Lager	24 - 12 oz bottles	14	52	0	10
11	1	70	Outback Lager	24 - 355 ml bottles	15	15	10	30
12	1	75	Rh?nbr?u Klosterbier	24 - 0.5 l bottles	7.75	125	0	25
13	1	76	Lakkalik??ri	500 ml	18	57	0	20
14	2	3	Aniseed Syrup	12 - 550 ml bottles	10	13	70	25
15	2	4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	22	53	0	0
16	2	5	Chef Anton's Gumbo Mix	36 boxes	21.35	0	0	0
17	2	6	Grandma's Boysenberry Spread	12 - 8 oz jars	25	120	0	25
18	2	8	Northwoods Cranberry Sauce	12 - 12 oz jars	40	6	0	0
19	2	15	Genen Shouyu	24 - 250 ml bottles	15.5	39	0	5
20	2	44	Gula Malacca	20 - 2 kg bags	19.45	27	0	15
21	2	61	Sirop d'érable	24 - 500 ml bottles	28.5	113	0	25
22	2	63	Vegie-spread	15 - 625 g jars	43.9	24	0	5
23	2	65	Louisiana Fiery Hot Pepper Sauce	32 - 8 oz bottles	21.05	76	0	0
24	2	66	Louisiana Hot Spiced Okra	24 - 8 oz jars	17	4	100	20
25	2	77	Original Frankfurter grüne So?e	12 boxes	13	32	0	15
26	3	16	Pavlova	32 - 500 g boxes	17.45	29	0	10
27	3	19	Teatime Chocolate Biscuits	10 boxes x 12 pieces	9.2	25	0	5
28	3	20	Sir Rodney's Marmalade	30 gift boxes	81	40	0	0
29	3	21	Sir Rodney's Scones	24 pkgs. x 4 pieces	10	3	40	5
30	3	25	NuNuCa Nu?-Nougat-Creme	20 - 450 g glasses	14	76	0	30

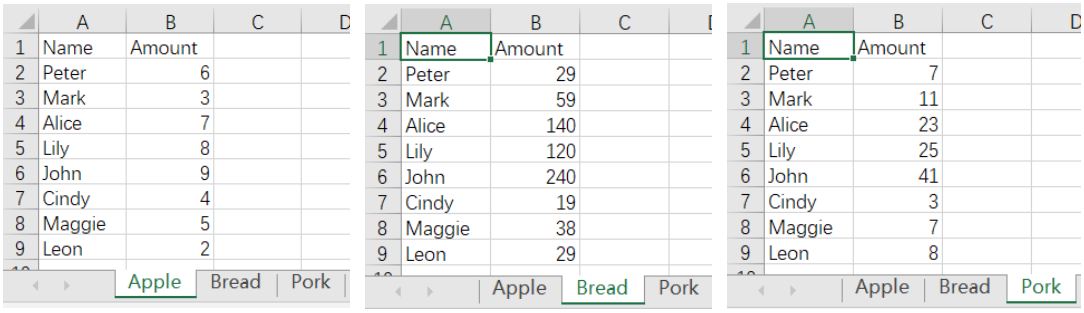
Sheet1

Script:

	A
1	=T("ProductCategories.xlsx")
2	=A1.groups(CategoryID,CategoryName,Description)
3	=A1.new(CategoryID,ProductID,ProductName,QuantityPerUnit,UnitPrice,UnitsInStock,UnitsOnOrder,ReorderLevel)
4	=T("Categories.xlsx",A2)
5	=T("Products.xlsx",A3)

5.11 Split multi-Sheet file into multiple files - unfixed number of Sheets

There is a multi-Sheet file Amount.xlsx as follows:



	A	B
1	Name	Amount
2	Peter	6
3	Mark	3
4	Alice	7
5	Lily	8
6	John	9
7	Cindy	4
8	Maggie	5
9	Leon	2

Apple | Bread | Pork

	A	B
1	Name	Amount
2	Peter	29
3	Mark	59
4	Alice	140
5	Lily	120
6	John	240
7	Cindy	19
8	Maggie	38
9	Leon	29

Apple | Bread | Pork

	A	B
1	Name	Amount
2	Peter	7
3	Mark	11
4	Alice	23
5	Lily	25
6	John	41
7	Cindy	3
8	Maggie	7
9	Leon	8

Apple | Bread | Pork

The number of Sheets is not fixed, now we need to separate each Sheet to form one file, and take the Sheet name as the file name. The results after splitting are shown as below:

Apple.xlsx：

	A	B
1	Name	Amount
2	Peter	6
3	Mark	3
4	Alice	7
5	Lily	8
6	John	9
7	Cindy	4
8	Maggie	5
9	Leon	2

Bread.xlsx：

	A	B
1	Name	Amount
2	Peter	29
3	Mark	59
4	Alice	140
5	Lily	120
6	John	240
7	Cindy	19
8	Maggie	38
9	Leon	29

Pork.xlsx：

	A	B
1	Name	Amount
2	Peter	7
3	Mark	11
4	Alice	23
5	Lily	25
6	John	41
7	Cindy	3
8	Maggie	7
9	Leon	8

Script:

	A	B
1	=file("Amounts.xlsx").xlsopen()	
2	for A1	=A1.xlsimport@t(;A2.stname)
3		=T(A2.stname+".xlsx",B2)

Chapter 6 **Searching, positioning and filtering**

6.1 **Search for the nth, the nth from last**

Here below is a sales statistical table:

	A	B	C
1	CategoryName	ProductName	ProductSales
2	Beverages	C?te de Blaye	46563.09
3	Beverages	Chai	4887
4	Beverages	Chang	7038.55
5	Beverages	Chartreuse verte	4475.7
6	Beverages	Guaraná Fantástica	1553.63
7	Beverages	Ipoh Coffee	11069.9
8	Beverages	Lakkalik??ri	7883.1
9	Beverages	Laughing Lumberjack Lager	910
10	Beverages	Outback Lager	5828.4
11	Beverages	Rh?nbr?u Klosterbier	4191.04
12	Beverages	Sasquatch Ale	1967
13	Beverages	Steeleye Stout	5706.9
14	Condiments	Aniseed Syrup	1724
15	Condiments	Chef Anton's Cajun Seasoning	5214.88
16	Condiments	Chef Anton's Gumbo Mix	373.63
17	Condiments	Genen Shouyu	1474.83
18	Condiments	Grandma's Boysenberry Spread	2500
19	Condiments	Gula Malacca	6543.45
20	Condiments	Louisiana Fiery Hot Pepper Sauce	9331.08
21	Condiments	Louisiana Hot Spiced Okra	2958
22	Condiments	Northwoods Cranberry Sauce	4260
23	Condiments	Original Frankfurter grüne So?e	4906.98
24	Condiments	Sirop d'érable	9091.5
25	Condiments	Vegie-spread	6899.25
26	Confections	Chocolade	1282.01
27	Confections	Gumb?r Gummib?rchen	11225.66
28	Confections	Maxilaku	3060.6
29	Confections	NuNuCa Nu?-Nougat-Creme	1551.9
30	Confections	Pavlova	7180.17
31	Confections	Schoggi Schokolade	10974

Sheet1

Now we want to find out the product ranked tenth by sales and its sales, and the product ranked tenth from last by sales and its sales. The results are as follows:

The 10th:

```
=spl("=E(?1).sort(ProductSales:-1)(10)",A1:C78)
```

CategoryName	ProductName	ProductSales
Dairy Products	Gudbrandsdalsost	14041.8

The 10th from last:

```
=spl("=E(?1).sort(ProductSales:-1).m(-10)",A1:C78)
```

CategoryName	ProductName	ProductSales
Beverages	Guaraná Fantástica	1553.63

6.2 **Search for top N, last N**

Here below is a sales statistical table:

	A	B	C
1	CategoryName	ProductName	ProductSales
2	Beverages	C?te de Blaye	46563.09
3	Beverages	Chai	4887
4	Beverages	Chang	7038.55
5	Beverages	Chartreuse verte	4475.7
6	Beverages	Guaraná Fantástica	1553.63
7	Beverages	Ipoh Coffee	11069.9
8	Beverages	Lakkalik??ri	7883.1
9	Beverages	Laughing Lumberjack Lager	910
10	Beverages	Outback Lager	5828.4
11	Beverages	Rh?nbr?u Klosterbier	4191.04
12	Beverages	Sasquatch Ale	1967
13	Beverages	Steeleye Stout	5706.9
14	Condiments	Aniseed Syrup	1724
15	Condiments	Chef Anton's Cajun Seasoning	5214.88
16	Condiments	Chef Anton's Gumbo Mix	373.63
17	Condiments	Genen Shouyu	1474.83
18	Condiments	Grandma's Boysenberry Spread	2500
19	Condiments	Gula Malacca	6543.45
20	Condiments	Louisiana Fiery Hot Pepper Sauce	9331.08
21	Condiments	Louisiana Hot Spiced Okra	2958
22	Condiments	Northwoods Cranberry Sauce	4260
23	Condiments	Original Frankfurter grüne So?e	4906.98
24	Condiments	Sirop d'érable	9091.5
25	Condiments	Vegie-spread	6899.25
26	Confections	Chocolade	1282.01
27	Confections	Gumb?r Gummib?rchen	11225.66
28	Confections	Maxilaku	3060.6
29	Confections	NuNuCa Nu?-Nougat-Creme	1551.9
30	Confections	Pavlova	7180.17
31	Confections	Schoggi Schokolade	10974

Sheet1

Now we want to find out the products ranked in the top 10 by sales and their sales, and the last 10 products and their sales. The results are shown as below:

Top 10:

```
=spl("=E(?1).sort(ProductSales:-1).to(10)",A1:C78)
```

CategoryName	ProductName	ProductSales
Beverages	C?te de Blaye	46563.09
Dairy Products	Raclette Courdavault	33616.55
Meat/Poultry	Thüringer Rostbratwurst	33109.51
Grains/Cereals	Gnocchi di nonna Alice	32604
Produce	Manjimup Dried Apples	23550.02
Confections	Tarte au sucre	20762.82
Dairy Products	Camembert Pierrot	20652.28
Meat/Poultry	Alice Mutton	16580.85
Seafood	Carnarvon Tigers	15950
Dairy Products	Gudbrandsdalsost	14041.8

Last 10:

```
=spl("=E(?1).sort(ProductSales:-1).to(-10)",A1:C78)
```

CategoryName	ProductName	ProductSales
Beverages	Guaraná Fantástica	1553.63
Confections	NuNuCa Nu?-Nougat-Creme	1551.9
Condiments	Genen Shouyu	1474.83
Confections	Chocolade	1282.01
Beverages	Laughing Lumberjack Lager	910
Produce	Longlife Tofu	888
Dairy Products	Geitost	786
Seafood	Konbu	758.94
Seafood	Gravad lax	629.2
Condiments	Chef Anton's Gumbo Mix	373.63

6.3 **Filter by position**

We have a statistical table for daily sales of January 2022:

	A	B
1	Date	Sales
2	2022-01-01 00:00:00	98.0000
3	2022-01-02 00:00:00	174.0000
4	2022-01-03 00:00:00	168.0000
5	2022-01-04 00:00:00	167.4000
6	2022-01-05 00:00:00	1696.0000
7	2022-01-06 00:00:00	214.2000
8	2022-01-07 00:00:00	77.0000
9	2022-01-08 00:00:00	1261.4000
10	2022-01-09 00:00:00	336.0000
11	2022-01-10 00:00:00	95.7600
12	2022-01-11 00:00:00	222.3000
13	2022-01-12 00:00:00	2462.4000
14	2022-01-13 00:00:00	47.5000
15	2022-01-14 00:00:00	1088.0000
16	2022-01-15 00:00:00	640.0000
17	2022-01-16 00:00:00	604.8000
18	2022-01-17 00:00:00	200.0000
19	2022-01-18 00:00:00	342.7200
20	2022-01-19 00:00:00	168.0000
21	2022-01-20 00:00:00	45.9000
22	2022-01-21 00:00:00	1320.0000
23	2022-01-22 00:00:00	304.0000
24	2022-01-23 00:00:00	486.5000
25	2022-01-24 00:00:00	380.0000
26	2022-01-25 00:00:00	393.0000
27	2022-01-26 00:00:00	124.8000

Now we want to filter out the sales on even-numbered days:

```
=spl("=E(?1).select(#%2==0)",A1:B32)
```

The symbol # represents the current row number.

Date	Sales
2022-01-02 00:00:00	174
2022-01-04 00:00:00	167.4
2022-01-06 00:00:00	214.2
2022-01-08 00:00:00	1261.4
2022-01-10 00:00:00	95.76
2022-01-12 00:00:00	2462.4
2022-01-14 00:00:00	1088
2022-01-16 00:00:00	604.8
2022-01-18 00:00:00	342.72
2022-01-20 00:00:00	45.9
2022-01-22 00:00:00	304
2022-01-24 00:00:00	380
2022-01-26 00:00:00	124.8
2022-01-28 00:00:00	86.4
2022-01-30 00:00:00	608

6.4 Search for position of a certain value, take the value by position

We have a statistical table for daily sales of January 2022:

	A	B
1	Date	Sales
2	2022-01-01 00:00:00	98.0000
3	2022-01-02 00:00:00	174.0000
4	2022-01-03 00:00:00	168.0000
5	2022-01-04 00:00:00	167.4000
6	2022-01-05 00:00:00	1696.0000
7	2022-01-06 00:00:00	214.2000
8	2022-01-07 00:00:00	77.0000
9	2022-01-08 00:00:00	1261.4000
10	2022-01-09 00:00:00	336.0000
11	2022-01-10 00:00:00	95.7600
12	2022-01-11 00:00:00	222.3000
13	2022-01-12 00:00:00	2462.4000
14	2022-01-13 00:00:00	47.5000
15	2022-01-14 00:00:00	1088.0000
16	2022-01-15 00:00:00	640.0000
17	2022-01-16 00:00:00	604.8000
18	2022-01-17 00:00:00	200.0000
19	2022-01-18 00:00:00	342.7200
20	2022-01-19 00:00:00	168.0000
21	2022-01-20 00:00:00	45.9000
22	2022-01-21 00:00:00	1320.0000
23	2022-01-22 00:00:00	304.0000
24	2022-01-23 00:00:00	486.5000
25	2022-01-24 00:00:00	380.0000
26	2022-01-25 00:00:00	393.0000
27	2022-01-26 00:00:00	124.8000

Now we want to find out the day with the largest sales. Enter the following in cell C2:

```
=spl("=E(?1).pmax(Sales)",A1:B32)
```

The returned result is 12.

Next, find out the sales of 5 days before and after the day with the largest sales:

```
=spl("=E(?1).to(?2-5,?2+5)",A1:B32,C2)
```

Date	Sales
2022/1/7	77
2022/1/8	1261.4
2022/1/9	336
2022/1/10	95.76
2022/1/11	222.30
2022/1/12	2462.40
2022/1/13	47.50
2022/1/14	1088.00
2022/1/15	640
2022/1/16	604.8
2022/1/17	200

6.5 Search for row number that satisfies the condition

We have a statistical table for daily sales of January 2022:

	A	B
1	Date	Sales
2	2022-01-01 00:00:00	98.0000
3	2022-01-02 00:00:00	174.0000
4	2022-01-03 00:00:00	168.0000
5	2022-01-04 00:00:00	167.4000
6	2022-01-05 00:00:00	1696.0000
7	2022-01-06 00:00:00	214.2000
8	2022-01-07 00:00:00	77.0000
9	2022-01-08 00:00:00	1261.4000
10	2022-01-09 00:00:00	336.0000
11	2022-01-10 00:00:00	95.7600
12	2022-01-11 00:00:00	222.3000
13	2022-01-12 00:00:00	2462.4000
14	2022-01-13 00:00:00	47.5000
15	2022-01-14 00:00:00	1088.0000
16	2022-01-15 00:00:00	640.0000
17	2022-01-16 00:00:00	604.8000
18	2022-01-17 00:00:00	200.0000
19	2022-01-18 00:00:00	342.7200
20	2022-01-19 00:00:00	168.0000
21	2022-01-20 00:00:00	45.9000
22	2022-01-21 00:00:00	1320.0000
23	2022-01-22 00:00:00	304.0000
24	2022-01-23 00:00:00	486.5000
25	2022-01-24 00:00:00	380.0000
26	2022-01-25 00:00:00	393.0000
27	2022-01-26 00:00:00	124.8000

The following code is to find out the date with daily sales greater than 1,000:

```
=spl("=E(?1).pselect@a(Sales>1000)",A1:B32)
```

pselect@a() means returning the sequence number of all rows that meet the condition, and pselect() means returning only the sequence number of the first row that meets the condition.

5
8
12
14
21

6.6 **Search for row that satisfies the condition**

We have a statistical table for daily sales of January 2022:

	A	B
1	Date	Sales
2	2022-01-01 00:00:00	98.0000
3	2022-01-02 00:00:00	174.0000
4	2022-01-03 00:00:00	168.0000
5	2022-01-04 00:00:00	167.4000
6	2022-01-05 00:00:00	1696.0000
7	2022-01-06 00:00:00	214.2000
8	2022-01-07 00:00:00	77.0000
9	2022-01-08 00:00:00	1261.4000
10	2022-01-09 00:00:00	336.0000
11	2022-01-10 00:00:00	95.7600
12	2022-01-11 00:00:00	222.3000
13	2022-01-12 00:00:00	2462.4000
14	2022-01-13 00:00:00	47.5000
15	2022-01-14 00:00:00	1088.0000
16	2022-01-15 00:00:00	640.0000
17	2022-01-16 00:00:00	604.8000
18	2022-01-17 00:00:00	200.0000
19	2022-01-18 00:00:00	342.7200
20	2022-01-19 00:00:00	168.0000
21	2022-01-20 00:00:00	45.9000
22	2022-01-21 00:00:00	1320.0000
23	2022-01-22 00:00:00	304.0000
24	2022-01-23 00:00:00	486.5000
25	2022-01-24 00:00:00	380.0000
26	2022-01-25 00:00:00	393.0000
27	2022-01-26 00:00:00	124.8000

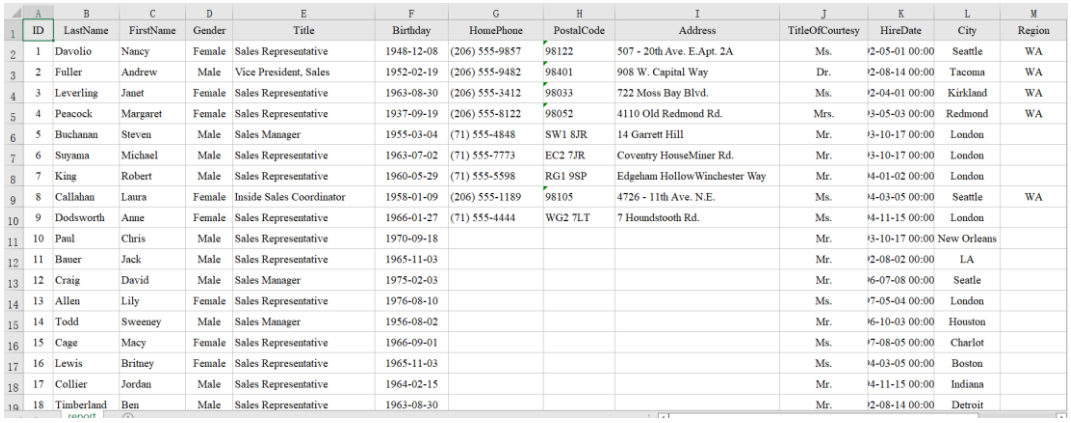
The following code is to find out the data with daily sales greater than 1,000:

```
=spl("=E(?1).select(Sales>1000)",A1:B32)
```

Date	Sales
2022-01-05 00:00:00	1696
2022-01-08 00:00:00	1261.4
2022-01-12 00:00:00	2462.4
2022-01-14 00:00:00	1088
2022-01-21 00:00:00	1320

6.7 **Filter by multiple conditions**

There is an employee information table:



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ID	LastName	FirstName	Gender	Title	Birthday	HomePhone	PostalCode	Address	TitleOfCourtesy	HireDate	City	Region
2	1	Davolio	Nancy	Female	Sales Representative	1948-12-08	(206) 555-9857	98122	507 - 20th Ave. E.Apt. 2A	Ms.	92-05-01 00:00	Seattle	WA
3	2	Fuller	Andrew	Male	Vice President, Sales	1952-02-19	(206) 555-9482	98401	908 W. Capital Way	Dr.	92-08-14 00:00	Tacoma	WA
4	3	Leverling	Janet	Female	Sales Representative	1963-08-30	(206) 555-3412	98033	722 Moss Bay Blvd.	Ms.	92-04-01 00:00	Kirkland	WA
5	4	Peacock	Margaret	Female	Sales Representative	1937-09-19	(206) 555-8122	98052	4110 Old Redmond Rd.	Mrs.	93-05-03 00:00	Redmond	WA
6	5	Buchanan	Steven	Male	Sales Manager	1955-03-04	(71) 555-4848	SW1 8JR	14 Garrett Hill	Mr.	93-10-17 00:00	London	
7	6	Suyama	Michael	Male	Sales Representative	1963-07-02	(71) 555-7773	EC2 7JR	Coventry HouseMiner Rd.	Mr.	93-10-17 00:00	London	
8	7	King	Robert	Male	Sales Representative	1960-05-29	(71) 555-5598	RG1 9SP	Edgeham HollowWinchester Way	Mr.	94-01-02 00:00	London	
9	8	Callahan	Laura	Female	Inside Sales Coordinator	1958-01-09	(206) 555-1189	98105	4726 - 11th Ave. N.E.	Ms.	94-03-05 00:00	Seattle	WA
10	9	Dodsworth	Anne	Female	Sales Representative	1966-01-27	(71) 555-4444	WG2 7LT	7 Houndstooth Rd.	Ms.	94-11-15 00:00	London	
11	10	Paul	Chris	Male	Sales Representative	1970-09-18				Mr.	93-10-17 00:00	New Orleans	
12	11	Bauer	Jack	Male	Sales Representative	1965-11-03				Mr.	92-08-02 00:00	LA	
13	12	Craig	David	Male	Sales Manager	1975-02-03				Mr.	96-07-08 00:00	Seatle	
14	13	Allen	Lily	Female	Sales Representative	1976-08-10				Ms.	97-05-04 00:00	London	
15	14	Todd	Sweeney	Male	Sales Manager	1956-08-02				Mr.	96-10-03 00:00	Houston	
16	15	Cage	Macy	Female	Sales Representative	1966-09-01				Ms.	97-08-05 00:00	Charlot	
17	16	Lewis	Britney	Female	Sales Representative	1965-11-03				Ms.	94-03-05 00:00	Boston	
18	17	Collier	Jordan	Male	Sales Representative	1964-02-15				Mr.	94-11-15 00:00	Indiana	
19	18	Timberland	Ben	Male	Sales Representative	1963-08-30				Mr.	92-08-14 00:00	Detroit	

Now we want to find out the employees who are Female and were born before 1970:

```
=spl("=E(?1).select(Gender==""Female"" && Birthday<""1970-01-01"")",A1:O32)
```

ID	LastName	FirstName	Gender	Title	Birthday	HomePhone	PostalCode	Address	TitleOfCourtesy	HireDate	City	Region	Country	Extension
1	Davolio	Nancy	Female	Sales Representative	1948-12-08	(206) 555-...	98122	507 - 20th Ave. E.Apt. 2A	Ms.	1992-05-0...	Seattle	WA	USA	5467
3	Leverling	Janet	Female	Sales Representative	1963-08-30	(206) 555-...	98033	722 Moss Bay Blvd.	Ms.	1992-04-0...	Kirkland	WA	USA	3355
4	Peacock	Margaret	Female	Sales Representative	1937-09-19	(206) 555-...	98052	4110 Old Redmond Rd.	Mrs.	1993-05-0...	Redmond	WA	USA	5176
8	Callahan	Laura	Female	Inside Sales Coordinator	1958-01-09	(206) 555-...	98105	4726 - 11th Ave. N.E.	Ms.	1994-03-0...	Seattle	WA	USA	2344
9	Dodsworth	Anne	Female	Sales Representative	1966-01-27	(71) 555-4...	WG2 7LT	7 Houndstooth Rd.	Ms.	1994-11-1...	London	(null)	UK	452
15	Cage	Macy	Female	Sales Representative	1966-09-01	(null)	(null)	(null)	Ms.	1997-08-0...	Charlot	(null)	USA	243
16	Lewis	Britney	Female	Sales Representative	1965-11-03	(null)	(null)	(null)	Ms.	1994-03-0...	Boston	(null)	USA	567
19	Carl	Donna	Female	Sales Representative	1960-05-29	(null)	(null)	(null)	Mrs.	1993-10-1...	Dalas	(null)	USA	122
21	Carlos	Betty	Female	Sales Representative	1955-03-04	(null)	(null)	(null)	Ms.	1994-11-1...	Atlanta	(null)	USA	3879

6.8 **Search by adjacent rows**

We have a statistical table for daily sales of January 2022:

	A	B
1	Date	Sales
2	2022-01-01 00:00:00	98.0000
3	2022-01-02 00:00:00	174.0000
4	2022-01-03 00:00:00	168.0000
5	2022-01-04 00:00:00	167.4000
6	2022-01-05 00:00:00	1696.0000
7	2022-01-06 00:00:00	214.2000
8	2022-01-07 00:00:00	77.0000
9	2022-01-08 00:00:00	1261.4000
10	2022-01-09 00:00:00	336.0000
11	2022-01-10 00:00:00	95.7600
12	2022-01-11 00:00:00	222.3000
13	2022-01-12 00:00:00	2462.4000
14	2022-01-13 00:00:00	47.5000
15	2022-01-14 00:00:00	1088.0000
16	2022-01-15 00:00:00	640.0000
17	2022-01-16 00:00:00	604.8000
18	2022-01-17 00:00:00	200.0000
19	2022-01-18 00:00:00	342.7200
20	2022-01-19 00:00:00	168.0000
21	2022-01-20 00:00:00	45.9000
22	2022-01-21 00:00:00	1320.0000
23	2022-01-22 00:00:00	304.0000
24	2022-01-23 00:00:00	486.5000
25	2022-01-24 00:00:00	380.0000
26	2022-01-25 00:00:00	393.0000
27	2022-01-26 00:00:00	124.8000

Now we want to find out the data of date where the previous and next day's sales are both more than 500 lower than that of the date:

```
=spl("=E(?1).select(Sales[-1]<Sales-500 && Sales[1]<Sales-500)",A1:B32)
```

Sales[-1] represents the value of the column Sales of the previous row, and Sales[1] represents the value of the column Sales of the next row.

Date	Sales
2022-01-05 00:00:00	1696
2022-01-08 00:00:00	1261.4
2022-01-12 00:00:00	2462.4
2022-01-21 00:00:00	1320
2022-01-27 00:00:00	877.5

6.9 Take values of adjacent rows in same group (search & filter within adjacent intervals)

There is an Excel table:

	A	B	C	D	E
1	OrderDate	ProductName	Sales	PreviousDailySales	NextDailySales
2	1996-07-04	Aniseed Syrup	3044.0000		
3	1996-07-04	Chai	12788.1000		
4	1996-07-04	Chang	16355.9600		
5	1996-07-04	Chef Anton's Cajun Seasoning	8567.9000		
6	1996-07-04	Chef Anton's Gumbo Mix	5347.2100		
7	1996-07-05	Aniseed Syrup	3044.0000		
8	1996-07-05	Chai	12788.1000		
9	1996-07-05	Chang	16355.9600		
10	1996-07-05	Chef Anton's Cajun Seasoning	8567.9000		
11	1996-07-05	Chef Anton's Gumbo Mix	5347.2100		
12	1996-07-08	Aniseed Syrup	6088.0000		
13	1996-07-08	Chai	25576.2000		
14	1996-07-08	Chang	32711.9200		
15	1996-07-08	Chef Anton's Cajun Seasoning	17135.8000		
16	1996-07-08	Chef Anton's Gumbo Mix	10694.4200		
17	1996-07-09	Aniseed Syrup	3044.0000		
18	1996-07-09	Chai	12788.1000		
19	1996-07-09	Chang	16355.9600		

Now we want to add two columns, PreviousDailySales and NextDailySales, to fill in the sales of current product on the previous selling day and the next selling day, respectively.

Analysis of the problem: The data is sorted by date first, and then by product. If the rows with same product are regarded as a group, the problem will change to taking the values of the previous row and the next row in the same group. The difficulty of the problem is how to find the previous and next rows in the same group without changing the order of rows.

There are two ideas to solve this problem (fill in the code in cell D1):

1. Search and filter in adjacent intervals: directly search forward and backward without changing the order of data. Once the row of the first product with the same name is found, it is the sales of the previous selling day/next selling day.

	A
1	=E('A1:C2401')
2	=A1.derive(~[:-1].select@1z(ProductName==A1.ProductName).Sales:PreviousDailySales, ~[1:].select@1(ProductName==A1.ProductName).Sales:NextDailySales)
3	return A2.new(PreviousDailySales,NextDailySales)

A2: ~[:-1] represents the set of all rows from the beginning to the previous row, and ~[1:] represents the set of all rows from the next row to the end.

2. Take the value of adjacent rows within the same group: group the data by product, and take the value of the previous row/the next row within the group directly, which is the sales of the previous selling day/next selling day.

	A
1	=E('A1:C2401').derive(:PreviousDailySales,:NextDailySales)
2	=A1.group(ProductName).run(~.run(PreviousDailySales=Sales[-1], NextDailySales=Sales[1]))
3	return A1.new(PreviousDailySales,NextDailySales)

A2: Sales[-1] represents the value of column Sales of the previous row, Sales[1] represents the value of column Sales of the next row.

	A	B	C	D	E
1	OrderDate	ProductName	Sales	PreviousDailySales	NextDailySales
2	1996-07-04	Aniseed Syrup	51919.0000		22783
3	1996-07-04	Chai	21169.0000		85616
4	1996-07-04	Chang	23154.0000		11653
5	1996-07-04	Chef Anton's Cajun Seasoning	20307.0000		22862
6	1996-07-04	Chef Anton's Gumbo Mix	97636.0000		72832
7	1996-07-05	Aniseed Syrup	22783.0000	51919	30295
8	1996-07-05	Chai	85616.0000	21169	61361
9	1996-07-05	Chang	11653.0000	23154	1383
10	1996-07-05	Chef Anton's Cajun Seasoning	22862.0000	20307	80644
11	1996-07-05	Chef Anton's Gumbo Mix	72832.0000	97636	68157
12	1996-07-08	Aniseed Syrup	30295.0000	22783	32772
13	1996-07-08	Chai	61361.0000	85616	72080
14	1996-07-08	Chang	1383.0000	11653	85152
15	1996-07-08	Chef Anton's Cajun Seasoning	80644.0000	22862	63940
16	1996-07-08	Chef Anton's Gumbo Mix	68157.0000	72832	3425
17	1996-07-09	Aniseed Syrup	32772.0000	30295	94325
18	1996-07-09	Chai	72080.0000	61361	95866
19	1996-07-09	Chang	85152.0000	1383	93226
20	1996-07-09	Chef Anton's Cajun Seasoning	63940.0000	80644	84700
21	1996-07-09	Chef Anton's Gumbo Mix	3425.0000	68157	26448
22	1996-07-10	Aniseed Syrup	94325.0000	32772	55325
23	1996-07-10	Chai	95866.0000	72080	96662
24	1996-07-10	Chang	93226.0000	85152	76960
25	1996-07-10	Chef Anton's Cajun Seasoning	84700.0000	63940	93287
26	1996-07-10	Chef Anton's Gumbo Mix	26448.0000	3425	80649
27	1996-07-11	Aniseed Syrup	55325.0000	94325	10565

6.10 **Filter by group's aggregation value**

We have an Excel table for the daily sales of products:

	A	B	C
1	OrderDate	ProductName	Sales
2	1996-07-04	Aniseed Syrup	51919.0000
3	1996-07-04	Chai	21169.0000
4	1996-07-04	Chang	23154.0000
5	1996-07-04	Chef Anton's Cajun Seasoning	20307.0000
6	1996-07-04	Chef Anton's Gumbo Mix	97636.0000
7	1996-07-05	Aniseed Syrup	22783.0000
8	1996-07-05	Chai	85616.0000
9	1996-07-05	Chang	11653.0000
10	1996-07-05	Chef Anton's Cajun Seasoning	22862.0000
11	1996-07-05	Chef Anton's Gumbo Mix	72832.0000
12	1996-07-08	Aniseed Syrup	30295.0000
13	1996-07-08	Chai	61361.0000
14	1996-07-08	Chang	1383.0000
15	1996-07-08	Chef Anton's Cajun Seasoning	80644.0000
16	1996-07-08	Chef Anton's Gumbo Mix	68157.0000
17	1996-07-09	Aniseed Syrup	32772.0000
18	1996-07-09	Chai	72080.0000
19	1996-07-09	Chang	85152.0000
20	1996-07-09	Chef Anton's Cajun Seasoning	63940.0000
21	1996-07-09	Chef Anton's Gumbo Mix	3425.0000
22	1996-07-10	Aniseed Syrup	94325.0000
23	1996-07-10	Chai	95866.0000
24	1996-07-10	Chang	93226.0000
25	1996-07-10	Chef Anton's Cajun Seasoning	84700.0000
26	1996-07-10	Chef Anton's Gumbo Mix	26448.0000
27	1996-07-11	Aniseed Syrup	55325.0000

We want to find out the date when the total daily sales amount exceeds 300,000:

```
=spl("=E(?1).groups(OrderDate;sum(Sales):TotalSales).select(TotalSales>300000)",A1:C2401)
```

OrderDate	TotalSales
1996/7/10	394565
1996/7/11	402883
1996/7/22	331978
1996/8/1	304470
1996/8/6	306196
1996/8/12	333358
1996/8/16	323308
1996/8/20	326687
1996/8/26	305450
1996/8/28	309841

6.11 Use group's aggregation value when filtering

We have an aggregation table for the sales of grouped products:

	A	B	C
1	CategoryName	ProductName	ProductSales
2	Beverages	C?te de Blaye	46563.09
3	Beverages	Chai	4887
4	Beverages	Chang	7038.55
5	Beverages	Chartreuse verte	4475.7
6	Beverages	Guaraná Fantástica	1553.63
7	Beverages	Ipoh Coffee	11069.9
8	Beverages	Lakkalik??ri	7883.1
9	Beverages	Laughing Lumberjack Lager	910
10	Beverages	Outback Lager	5828.4
11	Beverages	Rh?nbr?u Klosterbier	4191.04
12	Beverages	Sasquatch Ale	1967
13	Beverages	Steeleye Stout	5706.9
14	Condiments	Aniseed Syrup	1724
15	Condiments	Chef Anton's Cajun Seasoning	5214.88
16	Condiments	Chef Anton's Gumbo Mix	373.63
17	Condiments	Genen Shouyu	1474.83
18	Condiments	Grandma's Boysenberry Spread	2500
19	Condiments	Gula Malacca	6543.45
20	Condiments	Louisiana Fiery Hot Pepper Sauce	9331.08
21	Condiments	Louisiana Hot Spiced Okra	2958
22	Condiments	Northwoods Cranberry Sauce	4260
23	Condiments	Original Frankfurter grüne So?e	4906.98
24	Condiments	Sirop d'érable	9091.5
25	Condiments	Vegie-spread	6899.25
26	Confections	Chocolade	1282.01
27	Confections	Gumb?r Gummib?rchen	11225.66
28	Confections	Maxilaku	3060.6
29	Confections	NuNuCa Nu?-Nougat-Creme	1551.9
30	Confections	Pavlova	7180.17
31	Confections	Schoggi Schokolade	10974

Sheet1

Now we want to find out the product whose sales is greater than the average in its group:

=spl("=E(?1).group(CategoryName).((a=~.avg(ProductSales),~.select(ProductSales>a))).conj()",A1:C78)

To solve this problem, we can first calculate group's average sales and assign it to the temporary variable a, and then use ProductSales>a as the filter condition to filter the group. Since it is a two-step calculation, an extra layer of parentheses is required.

CategoryName	ProductName	ProductSales
Beverages	C?te de Blaye	46563.09
Beverages	Ipoh Coffee	11069.9
Condiments	Chef Anton's Cajun Seasoning	5214.88
Condiments	Gula Malacca	6543.45
Condiments	Louisiana Fiery Hot Pepper Sauce	9331.08
Condiments	Original Frankfurter grüne So?e	4906.98
Condiments	Sirop d'érable	9091.5
Condiments	Vegie-spread	6899.25
Confections	Gumb?r Gummib?rchen	11225.66
Confections	Pavlova	7180.17
Confections	Schoggi Schokolade	10974
Confections	Sir Rodney's Marmalade	7314.3
Confections	Tarte au sucre	20762.82
Dairy Products	Camembert Pierrot	20652.28
Dairy Products	Gudbrandsdalsost	14041.8
Dairy Products	Mozzarella di Giovanni	11838.6
Dairy Products	Raclette Courdavault	33616.55
Grains/Cereals	Gnocchi di nonna Alice	32604
Meat/Poultry	Alice Mutton	16580.85
Meat/Poultry	Thüringer Rostbratwurst	33109.51
Produce	Manjimup Dried Apples	23550.02
Produce	R?ssle Sauerkraut	12854.28
Seafood	Boston Crab Meat	9796.33
Seafood	Carnarvon Tigers	15950
Seafood	Ikura	8819.5
Seafood	Inlagd Sill	6894.15
Seafood	Nord-Ost Matjeshering	5884.29

6.12 Filter by maximum or minimum value within a group (find out one for each group)

We have an Excel table for the daily sales of products:

	A	B	C
1	OrderDate	ProductName	Sales
2	1996-07-04	Aniseed Syrup	3044.0000
3	1996-07-04	Chai	12788.1000
4	1996-07-04	Chang	16355.9600
5	1996-07-04	Chef Anton's Cajun Seasoning	8567.9000
6	1996-07-04	Chef Anton's Gumbo Mix	5347.2100
7	1996-07-05	Aniseed Syrup	3044.0000
8	1996-07-05	Chai	12788.1000
9	1996-07-05	Chang	16355.9600
10	1996-07-05	Chef Anton's Cajun Seasoning	8567.9000
11	1996-07-05	Chef Anton's Gumbo Mix	5347.2100
12	1996-07-08	Aniseed Syrup	6088.0000
13	1996-07-08	Chai	25576.2000
14	1996-07-08	Chang	32711.9200
15	1996-07-08	Chef Anton's Cajun Seasoning	17135.8000
16	1996-07-08	Chef Anton's Gumbo Mix	10694.4200
17	1996-07-09	Aniseed Syrup	3044.0000
18	1996-07-09	Chai	12788.1000
19	1996-07-09	Chang	16355.9600

The code below is to find out the product with the largest daily sales and its sales:

=spl("=E(?1).group(OrderDate).(~.maxp(Sales))",A1:C2401)

OrderDate	ProductName	Sales
1996/7/4	Chef Anton's Gumbo Mix	97636
1996/7/5	Chai	85616
1996/7/8	Chef Anton's Cajun Seasoning	80644
1996/7/9	Chang	85152
1996/7/10	Chai	95866
1996/7/11	Chai	96662
1996/7/12	Chef Anton's Cajun Seasoning	59143
1996/7/15	Chef Anton's Gumbo Mix	95177
1996/7/16	Aniseed Syrup	82713
1996/7/17	Aniseed Syrup	98232
1996/7/18	Chai	68975
1996/7/19	Chang	73880
1996/7/22	Aniseed Syrup	81548
1996/7/23	Chai	55341
1996/7/24	Chef Anton's Cajun Seasoning	92312
1996/7/25	Chef Anton's Gumbo Mix	87198
1996/7/26	Aniseed Syrup	96086
1996/7/29	Chef Anton's Gumbo Mix	41919
1996/7/30	Chai	74750
1996/7/31	Chef Anton's Cajun Seasoning	96910
1996/8/1	Aniseed Syrup	92147
1996/8/2	Chang	87156
1996/8/5	Chef Anton's Cajun Seasoning	85283
1996/8/6	Chef Anton's Cajun Seasoning	99526
1996/8/7	Chang	67402
1996/8/8	Aniseed Syrup	73662

6.13 Find out interval in which a certain condition occurs continuously

We have a statistical table for daily sales:

	A	B
1	Date	Sales
2	2022-01-01 00:00:00	98.0000
3	2022-01-02 00:00:00	174.0000
4	2022-01-03 00:00:00	168.0000
5	2022-01-04 00:00:00	167.4000
6	2022-01-05 00:00:00	1696.0000
7	2022-01-06 00:00:00	214.2000
8	2022-01-07 00:00:00	77.0000
9	2022-01-08 00:00:00	1261.4000
10	2022-01-09 00:00:00	336.0000
11	2022-01-10 00:00:00	95.7600
12	2022-01-11 00:00:00	222.3000
13	2022-01-12 00:00:00	2462.4000
14	2022-01-13 00:00:00	47.5000
15	2022-01-14 00:00:00	1088.0000
16	2022-01-15 00:00:00	640.0000
17	2022-01-16 00:00:00	604.8000
18	2022-01-17 00:00:00	200.0000
19	2022-01-18 00:00:00	342.7200
20	2022-01-19 00:00:00	168.0000
21	2022-01-20 00:00:00	45.9000
22	2022-01-21 00:00:00	1320.0000
23	2022-01-22 00:00:00	304.0000
24	2022-01-23 00:00:00	486.5000
25	2022-01-24 00:00:00	380.0000
26	2022-01-25 00:00:00	393.0000
27	2022-01-26 00:00:00	124.8000

Find out the date when the sales rises for three consecutive days or more:

	A
1	=0
2	=E('A1:B32').group@o(if(Sales>Sales[-1], A1,A1=A1+1)).select(~.len()>=3).conj()

A2: Take A1 as a temporary value. A1 remains unchanged when the sales rises, and plus 1 when the sales falls, and then group according to this temporary value. In this way, the rows of consecutive rise are put into the same group.

Date	Sales
2022/1/10	95.76
2022/1/11	222.3
2022/1/12	2462.4
2022/1/28	86.4
2022/1/29	156
2022/1/30	608
2022/1/31	884

Chapter 7 **Calculate cell value and aggregation value**

7.1 **Simple column-wise aggregation**

Here below is a data table:

	A	B	C	D	E
1	Class	Name	Math	English	PE
2	one	Jack	89	78	83
3	one	Tom	90	60	99
4	one	Jerry	76	88	62
5	one	Kate	66	90	85
6	two	Jim	87	60	76
7	two	Alice	100	99	97
8	two	Rebecca	75	76	88
9	three	Cindy	63	80	72
10	three	Kitty	82	50	74
11	three	Lucy	40	100	63

To count the following information for each class: total number of students, the average score of individual total score, the code is as follows:

```
=spl("=E(?1).groups(Class;count(1):TotalNumber,avg(Math+English+PE):AverageScore)",A1:E11)
```

Class	TotalNumber	AverageScore
one	4	241.5
three	3	208
two	3	252.7

7.2 **Conditional aggregation**

Here below is a data table:

	A	B	C
1	Class	Name	Score
2	one	Jack	89
3	one	Tom	90
4	one	Jerry	76
5	one	Kate	66
6	one	Jim	82
7	one	Alice	73
8	two	Rebecca	86
9	two	Cindy	87
10	two	Kitty	100
11	two	Lucy	75
12	two	Peter	76
13	two	June	89
14	three	Leon	94
15	three	Mark	63
16	three	Maggie	82
17	three	Julia	40
18	three	Sunny	90
19	three	Lily	80

To count the following information for each class: number of students with score above 90, the number of failed students, and the average score after removing the highest and lowest scores, the code is as follows:

```
=spl("=E(?1).group(Class).new(Class,~.count(Score>=90):'ScoreAbove90',~.count(Score<60):Failed,~.sort(Score).m(2:-2).avg(Score):AverageScore)", A1:C19)
```

where **m(2:-2)** represents the set of members from the second to penultimate member of a sequence.

Class	ScoreAbove90	Failed	AverageScore
one	1	1	80
three	2	1	78.75
two	1	1	84.25

7.3 Fill aggregation value in the first row of the same group of data

Here below is a data table:

	A	B	C	D	E	F
1	Class	Name	Score			
2	one	Jack	89			
3	one	Tom	90			
4	one	Jerry	76			
5	one	Kate	56			
6	one	Jim	82			
7	one	Alice	73			
8	two	Rebecca	86			
9	two	Cindy	87			
10	two	Kitty	100			
11	two	Lucy	75			
12	two	Peter	46			
13	two	June	89			
14	three	Leon	94			
15	three	Mark	63			
16	three	Maggie	82			
17	three	Julia	40			
18	three	Sunny	90			
19	three	Lily	80			
20						

Now we want to count the following information for each class: number of students with score above 90, number of failed students, and the average score after removing the highest and lowest scores, and then fill the results in the first row of each class (calculate in cell D1):

	A
1	=E('A1:C19').derive(:ScoreAbove90,:Failed,:AverageScore)
2	=A1.group(Class).run(~(1).ScoreAbove90=~.select(Score>=90).len(),~(1).Failed=~.select(Score<60).len(),~(1).AverageScore=~.sort(Score).m(2:-2).avg(Score))
3	return A1.new(ScoreAbove90,Failed,AverageScore)

	A	B	C	D	E	F
1	Class	Name	Score	ScoreAbove90	Failed	AverageScore
2	one	Jack	89	1	1	80
3	one	Tom	90			
4	one	Jerry	76			
5	one	Kate	56			
6	one	Jim	82			
7	one	Alice	73			
8	two	Rebecca	86	1	1	84.25
9	two	Cindy	87			
10	two	Kitty	100			
11	two	Lucy	75			
12	two	Peter	46			
13	two	June	89			
14	three	Leon	94	2	1	78.75
15	three	Mark	63			
16	three	Maggie	82			
17	three	Julia	40			
18	three	Sunny	90			
19	three	Lily	80			

7.4 **Split aggregation value and fill them in detail rows**

We have an annual and monthly water consumption data table for water meter, and part of the data is shown as below:

	A	B	C	D
1	Year	Month	Water	Water leakage
2	2016	6	9739	
3	2016	7	5611	
4	2016	8	6005	
5	2016	9	6325	
6	2016	10	7435	
7	2016	11	5957	
8	2016	12	5366	
9	2017	1	8343	
10	2017	2	8136	
11	2017	3	8066	
12	2017	4	10828	
13	2017	5	7781	
14	2017	6	4155	
15	2017	7	4420	
16	2017	8	9953	
17	2017	9	6153	
18	2017	10	5827	
19	2017	11	6727	
20	2017	12	5000	
21	2018	1	10446	
22	2018	2	8415	

We also have a statistical table for annual water leakage amount:

	A	B
1	Year	Water leakage
2	2016	42828.2
3	2017	33252.6
4	2018	33104.8
5	2019	55849.9

Now we want to assign the annual water leakage amount to the Water leakage column of the first table according to the proportion of monthly water consumption in the total water consumption of the year (calculate in cell D1):

	A
1	=E('A1:C44').derive(:'Water leakage')
2	=E('Sheet2!A1:B5')
3	=A1.group(Year)
4	=A3.run(a=A2.select@1(Year==A3.Year).'Water leakage',s=~.sum(Water),~.run('Water leakage'=Water*a/s))
5	return A1.new('Water leakage')

A3: Group the data in A1 by Year.

A4: Loop through every group in A3; the variable a is the leakage loss of the corresponding year selected from A2; the variable s is the total Water consumption in this year, and then loop through all rows in this year; assign the Water leakage column as: Water*a/s.

A5: return to A1.

Year	Month	Water	Water leakage
2016	6	9739	8981.950984
2016	7	5611	5174.835915
2016	8	6005	5538.208816
2016	9	6325	5833.334015
2016	10	7435	6857.04955
2016	11	5957	5493.940036
2016	12	5366	4948.880684
2017	1	8343	3248.971668
2017	2	8136	3168.360721
2017	3	8066	3141.10098
2017	4	10828	4216.692464
2017	5	7781	3030.114893
2017	6	4155	1618.060324
2017	7	4420	1721.257914
2017	8	9953	3875.945705
2017	9	6153	2396.131209
2017	10	5827	2269.178702
2017	11	6727	2619.661083
2017	12	5000	1947.124337
2018	1	10446	3944.032172
2018	2	8415	3177.199954

7.5 **Simple accumulation**

There is a sales data table:

	A	B	C
1	Date	Sales	Cumulative sales
2	2021-01-01	21479	
3	2021-01-02	144136	
4	2021-01-03	65552	
5	2021-01-04	187506	
6	2021-01-05	3434	
7	2021-01-06	13146	
8	2021-01-07	192799	
9	2021-01-08	85490	
10	2021-01-09	74014	
11	2021-01-10	138232	
12	2021-01-11	126249	
13	2021-01-12	132568	
14	2021-01-13	91789	
15	2021-01-14	183192	
16	2021-01-15	75045	
17	2021-01-16	24870	
18	2021-01-17	96934	
19	2021-01-18	135606	
20	2021-01-19	156364	
21	2021-01-20	178160	
22	2021-01-21	116533	
23	2021-01-22	101199	
24	2021-01-23	82975	
25	2021-01-24	181258	
26	2021-01-25	91504	
27	2021-01-26	18056	
28	2021-01-27	39243	
29	2021-01-28	25590	
30	2021-01-29	183589	

We want to calculate the cumulative sales volume and fill them in column C. First calculate in cell C2:

```
=spl("=?1+?2",B2,C1)
```

Then drag C2 down to every relevant row:

	A	B	C
1	Date	Sales	Cumulative sales
2	2021-01-01	21479	21479
3	2021-01-02	144136	165615
4	2021-01-03	65552	231167
5	2021-01-04	187506	418673
6	2021-01-05	3434	422107
7	2021-01-06	13146	435253
8	2021-01-07	192799	628052
9	2021-01-08	85490	713542
10	2021-01-09	74014	787556
11	2021-01-10	138232	925788
12	2021-01-11	126249	1052037
13	2021-01-12	132568	1184605
14	2021-01-13	91789	1276394
15	2021-01-14	183192	1459586
16	2021-01-15	75045	1534631
17	2021-01-16	24870	1559501
18	2021-01-17	96934	1656435
19	2021-01-18	135606	1792041
20	2021-01-19	156364	1948405
21	2021-01-20	178160	2126565
22	2021-01-21	116533	2243098
23	2021-01-22	101199	2344297
24	2021-01-23	82975	2427272
25	2021-01-24	181258	2608530
26	2021-01-25	91504	2700034
27	2021-01-26	18056	2718090

7.6 Accumulate data in each group

There is a sales data table:

	A	B	C	D	E
1	Name	Sales	Date		
2	Tom	4343	2021/6/1		
3	Tom	23000	2021/6/2		
4	Tom	24619	2021/6/3		
5	Tom	13993	2021/6/4		
6	Tom	26323	2021/6/5		
7	Tom	4750	2021/6/6		
8	Tom	29202	2021/6/7		
9	John	15318	2021/6/1		
10	John	13114	2021/6/2		
11	John	16283	2021/6/3		
12	John	4562	2021/6/4		
13	John	22782	2021/6/5		
14	John	4572	2021/6/6		
15	John	25740	2021/6/7		
16	Kate	16888	2021/6/1		
17	Kate	14568	2021/6/2		
18	Kate	19250	2021/6/3		
19	Kate	10260	2021/6/4		
20	Kate	23908	2021/6/5		
21	Kate	10980	2021/6/6		
22	Kate	16136	2021/6/7		
23					

To calculate the cumulative sales of each person, the code is as follows (calculate in cell D1):

```
=spl("=E(?1).new(cum(~~+Sales;Name):'Cumulative Sales')",A1:C22)
```

cum(x; Gi,…) is an iteration function, used in the loop function; x is an expression containing ~~; ~~ represents the cumulative value; Gi is an ordered column, and group by Gi and accumulate within the group.

Name	Sales	Date	Cumulative Sales
Tom	4343	2021/6/1	4343
Tom	23000	2021/6/2	27343
Tom	24619	2021/6/3	51962
Tom	13993	2021/6/4	65955
Tom	26323	2021/6/5	92278
Tom	4750	2021/6/6	97028
Tom	29202	2021/6/7	126230
John	15318	2021/6/1	15318
John	13114	2021/6/2	28432
John	16283	2021/6/3	44715
John	4562	2021/6/4	49277
John	22782	2021/6/5	72059
John	4572	2021/6/6	76631
John	25740	2021/6/7	102371
Kate	16888	2021/6/1	16888
Kate	14568	2021/6/2	31456
Kate	19250	2021/6/3	50706
Kate	10260	2021/6/4	60966
Kate	23908	2021/6/5	84874
Kate	10980	2021/6/6	95854
Kate	16136	2021/6/7	111990

7.7 Filter by Accumulation

There is a sales data table:

	A	B	C
1	Name	Sales	Date
2	Tom	4343	2021/6/1
3	Tom	23000	2021/6/2
4	Tom	24619	2021/6/3
5	Tom	13993	2021/6/4
6	Tom	26323	2021/6/5
7	Tom	4750	2021/6/6
8	Tom	29202	2021/6/7
9	John	15318	2021/6/1
10	John	13114	2021/6/2
11	John	16283	2021/6/3
12	John	4562	2021/6/4
13	John	22782	2021/6/5
14	John	4572	2021/6/6
15	John	25740	2021/6/7
16	Kate	16888	2021/6/1
17	Kate	14568	2021/6/2
18	Kate	19250	2021/6/3
19	Kate	10260	2021/6/4
20	Kate	23908	2021/6/5
21	Kate	10980	2021/6/6
22	Kate	16136	2021/6/7

To count the date when the sales volume of each person reaches 50,000, the code is as follows:

```
=spl("=E(?1).group(Name).new(Name,~.select@1(Sales[:0].sum()>50000).Date:Date)",A1:C22)
```

Name	Date
John	2021/6/5
Kate	2021/6/3
Tom	2021/6/3

7.8 Early-terminated accumulation

Here below is an inventory data table:

	A	B	C	D
1	Code	WarehouseID	ShelfID	Quantity
2	ANU400662P0039	1220	9092	3
3	ANU400662P0039	1220	9092	2
4	ANU400662P0039	2000	1890	3
5	ANU400662P0039	2000	1890	1
6	ANU400662P0039	1220	9001	19
7	ANU400662P0039	1010	1801	1
8	ANU400662P0039	1220	9001	11
9	ANU400662P0039	1220	9001	1
10	ANU400662P0039	1220	9001	1
11	ANU400662P0039	1220	9001	1
12	ANU400662P0039	1220	9001	1
13	ANU400662P0039	1220	9001	16
14	ANU400662P0039	2000	1889	20
15	ANU400662P0039	1220	9001	98
16	ANU400662P0039	1220	9001	1
17	ANU400662P0039	1220	9001	1
18	ANU400662P0039	1220	9001	1

We've known that the quantity of this product sold today is 50, and want to calculate the new inventory data (subtract the inventory quantity in turn according to the order in the table until 50 are subtracted in total, and keep only the rows with inventory quantity greater than 0):

	A
1	=E('A1:D18')
2	=A1.iterate((a=min(Quantity,~~),Quantity-=a,~~-a),50,~~==0)
3	return A1.select(Quantity>0)

A2: Use iterate to loop the iteration, the ~~ in the loop represents the result of the last iteration, and its initial value is set to 50. Take the minimum value of the quantity of current row and ~~, and assign the value to the variable a; subtract a from the quantity of current row, and take ~~-a as the result of this iteration; When the iteration result ~~ is 0, terminate the iteration.

A3: Select the rows with quantity>0 in A1 after iteration.

Code	WarehouseID	ShelfID	Quantity
ANU400662P0039	1220	9001	10
ANU400662P0039	2000	1889	20
ANU400662P0039	1220	9001	98
ANU400662P0039	1220	9001	1
ANU400662P0039	1220	9001	1
ANU400662P0039	1220	9001	1

7.9 Accumulation for continuous occurrence of a certain condition

There is a statistical table for daily sales:

	A	B
1	Date	Sales
2	2022-01-01 00:00:00	98.0000
3	2022-01-02 00:00:00	174.0000
4	2022-01-03 00:00:00	168.0000
5	2022-01-04 00:00:00	167.4000
6	2022-01-05 00:00:00	1696.0000
7	2022-01-06 00:00:00	214.2000
8	2022-01-07 00:00:00	77.0000
9	2022-01-08 00:00:00	1261.4000
10	2022-01-09 00:00:00	336.0000
11	2022-01-10 00:00:00	95.7600
12	2022-01-11 00:00:00	222.3000
13	2022-01-12 00:00:00	2462.4000
14	2022-01-13 00:00:00	47.5000
15	2022-01-14 00:00:00	1088.0000
16	2022-01-15 00:00:00	640.0000
17	2022-01-16 00:00:00	604.8000
18	2022-01-17 00:00:00	200.0000
19	2022-01-18 00:00:00	342.7200
20	2022-01-19 00:00:00	168.0000
21	2022-01-20 00:00:00	45.9000
22	2022-01-21 00:00:00	1320.0000
23	2022-01-22 00:00:00	304.0000
24	2022-01-23 00:00:00	486.5000
25	2022-01-24 00:00:00	380.0000
26	2022-01-25 00:00:00	393.0000
27	2022-01-26 00:00:00	124.8000

We want to add a column to the right to calculate the cumulative days of consecutive rising (calculate in cell C1):

	A
1	=0
2	return E('A1:B32').new(if(Sales>Sales[-1], A1=A1+1,A1=0):RiseDayCount)

A2: Take A1 as a temporary value. A1 plus 1 when the sales volume rises, and assign A1 to 0 when the sales volume falls.

	A	B	C
1	Date	Sales	RiseDayCount
2	2022-01-01 00:00:00	98.0000	1
3	2022-01-02 00:00:00	174.0000	2
4	2022-01-03 00:00:00	168.0000	0
5	2022-01-04 00:00:00	167.4000	0
6	2022-01-05 00:00:00	1696.0000	1
7	2022-01-06 00:00:00	214.2000	0
8	2022-01-07 00:00:00	77.0000	0
9	2022-01-08 00:00:00	1261.4000	1
10	2022-01-09 00:00:00	336.0000	0
11	2022-01-10 00:00:00	95.7600	0
12	2022-01-11 00:00:00	222.3000	1
13	2022-01-12 00:00:00	2462.4000	2
14	2022-01-13 00:00:00	47.5000	0
15	2022-01-14 00:00:00	1088.0000	1
16	2022-01-15 00:00:00	640.0000	0
17	2022-01-16 00:00:00	604.8000	0
18	2022-01-17 00:00:00	200.0000	0
19	2022-01-18 00:00:00	342.7200	1
20	2022-01-19 00:00:00	168.0000	0
21	2022-01-20 00:00:00	45.9000	0
22	2022-01-21 00:00:00	1320.0000	1
23	2022-01-22 00:00:00	304.0000	0
24	2022-01-23 00:00:00	486.5000	1
25	2022-01-24 00:00:00	380.0000	0
26	2022-01-25 00:00:00	393.0000	1
27	2022-01-26 00:00:00	124.8000	0
28	2022-01-27 00:00:00	877.5000	1
29	2022-01-28 00:00:00	86.4000	0
30	2022-01-29 00:00:00	156.0000	1
31	2022-01-30 00:00:00	608.0000	2
32	2022-01-31 00:00:00	884.0000	3

7.10 Calculate using adjacent row/interval when data of the same group is continuous (link relative ratio and YOY)

Here below is an annual and quarterly sales data table:

	A	B	C	D	E
1	Year	Quarter	Sales		
2	2018	First	397824		
3	2018	Second	102109		
4	2018	Third	558042		
5	2018	Fourth	1003743		
6	2019	First	768595		
7	2019	Second	343989		
8	2019	Third	695316		
9	2019	Fourth	1174501		
10	2020	First	983170		
11	2020	Second	601321		
12	2020	Third	1041197		
13	2020	Fourth	210749		
14	2021	First	134960		
15	2021	Second	879064		
16	2021	Third	114080		
17	2021	Fourth	1061119		
18	2022	First	1226876		
19	2022	Second	348215		
20	2022	Third	108836		
21	2022	Fourth	731940		
22					

Now we want to calculate LRR and YOY (calculate in cell D1):

	A
1	=E('A1:C21')
2	=A1.(Sales-Sales[-1])
3	=A1.group(Year)
4	=A3.(~.(Sales-A3.~[-1](#).Sales)).conj()
5	return A2.new(~:LinkRelative,A4(#):YOY)

A2: Calculate LRR, [-1] represents the previous row.

A3: Group by Year.

A4: Calculate YOY. The symbol ~ represents the current member in the loop function, and # represents the sequence number of current member in loop function.

Year	Quarter	Sales	LinkRelative	YOY
2018	First	397824	397824	397824
2018	Second	102109	-295715	102109
2018	Third	558042	455933	558042
2018	Fourth	1003743	445701	1003743
2019	First	768595	-235148	370771
2019	Second	343989	-424606	241880
2019	Third	695316	351327	137274
2019	Fourth	1174501	479185	170758
2020	First	983170	-191331	214575
2020	Second	601321	-381849	257332
2020	Third	1041197	439876	345881
2020	Fourth	210749	-830448	-963752
2021	First	134960	-75789	-848210
2021	Second	879064	744104	277743
2021	Third	114080	-764984	-927117
2021	Fourth	1061119	947039	850370
2022	First	1226876	165757	1091916
2022	Second	348215	-878661	-530849
2022	Third	108836	-239379	-5244
2022	Fourth	731940	623104	-329179

7.11 Calculate using adjacent row/interval when data of the same group is discontinuous (LRR/YOY in the case of missing data)

Here below is an annual and quarterly sales data table:

	A	B	C	D	E
1	Year	Quarter	Sales	LinkRelative	YOY
2	2018	First	397824		
3	2018	Second	102109		
4	2018	Third	558042		
5	2018	Fourth	1003743		
6	2019	First	768595		
7	2019	Second	343989		
8	2019	Third	695316		
9	2019	Fourth	1174501		
10	2020	Second	601321		
11	2020	Third	1041197		
12	2020	Fourth	210749		
13	2021	First	134960		
14	2021	Second	879064		
15	2021	Third	114080		
16	2021	Fourth	1061119		
17	2022	First	1226876		
18	2022	Second	348215		
19	2022	Third	108836		
20	2022	Fourth	731940		

In this table, the data of the first quarter in 2020 is missing. When the data of this quarter is used to calculate LRR, skip this quarter directly, and use the data of the fourth quarter in 2019; when the data of this quarter is used to calculate YOY, regard it as zero (calculate in the cell D1):

```
=spl("=E(?1).new(Sales-Sales[-1]:LinkRelative,Sales-~[:-1].select@z1(Year==get(1,Year)-1 &&
Quarter==get(1,Quarter)).Sales:YOY)",A1:C20)
```

Where, get(1,Year) means taking the value in the column Year of current member of previous-layer function.

~[:-1] represents the set from the first member to the previous member.

Year	Quarter	Sales	LinkRelative	YOY
2018	First	397824	397824	397824
2018	Second	102109	-295715	102109
2018	Third	558042	455933	558042
2018	Fourth	1003743	445701	1003743
2019	First	768595	-235148	370771
2019	Second	343989	-424606	241880
2019	Third	695316	351327	137274
2019	Fourth	1174501	479185	170758
2020	Second	601321	-573180	257332
2020	Third	1041197	439876	345881
2020	Fourth	210749	-830448	-963752
2021	First	134960	-75789	134960
2021	Second	879064	744104	277743
2021	Third	114080	-764984	-927117
2021	Fourth	1061119	947039	850370
2022	First	1226876	165757	1091916
2022	Second	348215	-878661	-530849
2022	Third	108836	-239379	-5244
2022	Fourth	731940	623104	-329179

7.12 **Merge data of the same group**

Here below is an annual and quarterly sales data table:

	A	B	C
1	Year	Quarter	Sales
2	2018	First	397824
3	2018	Second	102109
4	2018	Third	558042
5	2018	Fourth	1003743
6	2019	First	768595
7	2019	Second	343989
8	2019	Third	695316
9	2019	Fourth	1174501
10	2020	First	983170
11	2020	Second	601321
12	2020	Third	1041197
13	2020	Fourth	210749
14	2021	First	134960
15	2021	Second	879064
16	2021	Third	114080
17	2021	Fourth	1061119
18	2022	First	1226876
19	2022	Second	348215
20	2022	Third	108836
21	2022	Fourth	731940

In this table, column A is ordered and has duplicate values. Now we want to clear the data from all rows in the same group except the first row.

```
=spl("=E(?1).group(#1).(~.run(if(#==1,,#1=null))).conj()",A1:C21)
```

Year	Quarter	Sales
2018	First	397824
	Second	102109
	Third	558042
	Fourth	1003743
2019	First	768595
	Second	343989
	Third	695316
	Fourth	1174501
2020	First	983170
	Second	601321
	Third	1041197
	Fourth	210749
2021	First	134960
	Second	879064
	Third	114080
	Fourth	1061119
2022	First	1226876
	Second	348215
	Third	108836
	Fourth	731940

7.13 String concatenation and aggregation

There is a data table:

	A	B	C
1	Name	ID	
2	Jack	1	
3	Jack	2	
4	Jack	3	
5	Emily	4	
6	Emily	5	
7	John	6	
8	John	7	
9	John	8	
10	Lauren	9	
11	Lauren	10	
12	Lauren	11	
13	Lauren	12	

Now we want to concatenate the ID numbers for each group of names by | to form a string, and put the result in column C (calculate in cell C1):

```
=spl("=E(?1).new(cum(~~+""|""+ID;Name):IDs)",A1:B13)
```

	A	B	C
1	Name	ID	IDs
2	Jack	1	|1
3	Jack	2	|1|2
4	Jack	3	|1|2|3
5	Emily	4	|4
6	Emily	5	|4|5
7	John	6	|6
8	John	7	|6|7
9	John	8	|6|7|8
10	Lauren	9	|9
11	Lauren	10	|9|10
12	Lauren	11	|9|10|11
13	Lauren	12	|9|10|11|12

7.14 Calculate proportion using aggregation information of data of the same group

Here below is an annual and quarterly sales data table:

	A	B	C
1	Year	Quarter	Sales
2	2018	First	397824
3	2018	Second	102109
4	2018	Third	558042
5	2018	Fourth	1003743
6	2019	First	768595
7	2019	Second	343989
8	2019	Third	695316
9	2019	Fourth	1174501
10	2020	First	983170
11	2020	Second	601321
12	2020	Third	1041197
13	2020	Fourth	210749
14	2021	First	134960
15	2021	Second	879064
16	2021	Third	114080
17	2021	Fourth	1061119
18	2022	First	1226876
19	2022	Second	348215
20	2022	Third	108836
21	2022	Fourth	731940

The following code is to calculate the proportion of each quarterly sales to the total sales of the year (calculate in cell D1):

```
=spl("=E(?1).group(Year).((a=~.sum(Sales),~.(Sales/a))).conj().new(~:Proportion)",A1:C21)
```

	A	B	C	D
1	Year	Quarter	Sales	Proportion
2	2018	First	397824	19.30%
3	2018	Second	102109	4.95%
4	2018	Third	558042	27.07%
5	2018	Fourth	1003743	48.68%
6	2019	First	768595	25.77%
7	2019	Second	343989	11.53%
8	2019	Third	695316	23.31%
9	2019	Fourth	1174501	39.38%
10	2020	First	983170	34.66%
11	2020	Second	601321	21.20%
12	2020	Third	1041197	36.71%
13	2020	Fourth	210749	7.43%
14	2021	First	134960	6.16%
15	2021	Second	879064	40.15%
16	2021	Third	114080	5.21%
17	2021	Fourth	1061119	48.47%
18	2022	First	1226876	50.78%
19	2022	Second	348215	14.41%
20	2022	Third	108836	4.51%
21	2022	Fourth	731940	30.30%

7.15 Generate number in each group

Here below is the student data table:

	A	B	C
1	Class	Name	ID
2	001	Jack	
3	001	Tom	
4	001	Jerry	
5	001	Kate	
6	001	Jim	
7	002	Alice	
8	002	Rebecca	
9	002	Cindy	
10	002	Kitty	
11	002	Lucy	
12	003	Peter	
13	003	June	
14	003	Leon	
15	003	Mark	
16	003	Maggie	
17	004	Julia	
18	004	Sunny	
19	004	Lily	
20	004	Emily	
21	004	John	

Now we want to create a student ID for every student. The student ID should be: class number + student's serial number in the class, and the serial number should be two-digit (calculate in cell C1):

```
=spl("=E(?1).group(Class).(~.(Class+string(#,""00""))).conj().new(~:ID)",A1:B21)
```

	A	B	C
1	Class	Name	ID
2	001	Jack	00101
3	001	Tom	00102
4	001	Jerry	00103
5	001	Kate	00104
6	001	Jim	00105
7	002	Alice	00201
8	002	Rebecca	00202
9	002	Cindy	00203
10	002	Kitty	00204
11	002	Lucy	00205
12	003	Peter	00301
13	003	June	00302
14	003	Leon	00303
15	003	Mark	00304
16	003	Maggie	00305
17	004	Julia	00401
18	004	Sunny	00402
19	004	Lily	00403
20	004	Emily	00404
21	004	John	00405

Chapter 8 Operation on sets and judgment of belongingness

8.1 Intersection, union and difference in the case of simple members - two sets

The following table lists the top 10 products by sales in January and February:

	A	B	C
1	Ranking	Jan	Feb
2	1	Sasquatch Ale	Northwoods Cranberry Sauce
3	2	Steeleye Stout	Original Frankfurter grüne So?e
4	3	Aniseed Syrup	Chef Anton's Cajun Seasoning
5	4	Chef Anton's Cajun Seasoning	Genen Shouyu
6	5	Chef Anton's Gumbo Mix	Louisiana Fiery Hot Pepper Sauce
7	6	Genen Shouyu	Sasquatch Ale
8	7	Grandma's Boysenberry Spread	Valkoinen suklaa
9	8	Gula Malacca	Zaanse koeken
10	9	Louisiana Fiery Hot Pepper Sauce	Camembert Pierrot
11	10	Louisiana Hot Spiced Okra	Flotemysost

Find out the products that make the top 10 in both January and February:

```
=spl("=?1^?2",B2:B11,C2:C11)
```

Sasquatch Ale
Chef Anton's Cajun Seasoning
Genen Shouyu
Louisiana Fiery Hot Pepper Sauce

Find out the products that make the top 10 once or more:

```
=spl("=?1&?2",B2:B11,C2:C11)
```

Sasquatch Ale
Steeleye Stout
Aniseed Syrup
Chef Anton's Cajun Seasoning
Chef Anton's Gumbo Mix
Genen Shouyu
Grandma's Boysenberry Spread
Gula Malacca
Louisiana Fiery Hot Pepper Sauce
Louisiana Hot Spiced Okra
Northwoods Cranberry Sauce
Original Frankfurter grüne So?e
Valkoinen suklaa
Zaanse koeken
Camembert Pierrot
Flotemysost

Find out the products that make the top 10 in January but fail to make the top 10 in February:

=spl("=?1\?2",B2:B11,C2:C11)

Steeleye Stout
Aniseed Syrup
Chef Anton's Gumbo Mix
Grandma's Boysenberry Spread
Gula Malacca
Louisiana Hot Spiced Okra

Find out the products that make the top 10 in February but fail to make the top 10 in January:

=spl("=**?2\?1**",B2:B11,C2:C11)

Northwoods Cranberry Sauce
Original Frankfurter grüne So?e
Valkoinen suklaa
Zaanse koeken
Camembert Pierrot
Flotemysost

8.2 Intersection, union and difference in the case of simple members - multiple sets

The following table lists the top 10 products by sales in the first couple of months of the year (the number of months will increase over time):

	A	B	C	D	E
1	Ranking	Jan	Feb	Mar	Apr
2	1	Sasquatch Ale	Northwoods Cranberry Sauce	Aniseed Syrup	Chef Anton's Gumbo Mix
3	2	Steeleye Stout	Original Frankfurter grüne So?e	Chef Anton's Cajun Seasoning	Sasquatch Ale
4	3	Aniseed Syrup	Chef Anton's Cajun Seasoning	Chef Anton's Gumbo Mix	Gustaf's Kn?ckebr?d
5	4	Chef Anton's Cajun Seasoning	Genen Shouyu	Sasquatch Ale	Ravioli Angelo
6	5	Chef Anton's Gumbo Mix	Louisiana Fiery Hot Pepper Sauce	Valkoinen suklaa	Singaporean Hokkien Fried Mee
7	6	Genen Shouyu	Sasquatch Ale	Zaanse koeken	Tofu
8	7	Grandma's Boysenberry Spread	Chef Anton's Gumbo Mix	Queso Manchego La Pastora	Uncle Bob's Organic Dried Pears
9	8	Gula Malacca	Zaanse koeken	Raclette Courdavault	Boston Crab Meat
10	9	Louisiana Fiery Hot Pepper Sauce	Camembert Pierrot	Filo Mix	Carnarvon Tigers
11	10	Louisiana Hot Spiced Okra	Flotemysost	Gnocchi di nonna Alice	Escargots de Bourgogne

Find out the products that make the top 10 in every month of these months:

```
=spl("=transpose(?1).isect()",B2:E11)
```

Sasquatch Ale	Chef Anton's Gumbo Mix

Find out the products that make the top 10 once or more:

```
=spl("=transpose(?1).union()",B2:E11)
```

The calculation result is not given here because it is very long.

Find out the products that make the top 10 in January but fail to make the top 10 in any of other months:

```
=spl("=transpose(?1).diff()",B2:E11)
```

Steeleye Stout	Grandma's Boysenberry Spread	Gula Malacca	Louisiana Hot Spiced Okra

8.3 Intersection, union and difference in the case of row-based data - two sets - by key column

The following tables list the sales data of the top 10 products by sales in January and February:

	A	B	C
1	Ranking	ProductName	Sales
2	1	Sasquatch Ale	774372
3	2	Steeleye Stout	736366
4	3	Aniseed Syrup	706337
5	4	Chef Anton's Cajun Seasoning	637046
6	5	Chef Anton's Gumbo Mix	566448
7	6	Genen Shouyu	508177
8	7	Grandma's Boysenberry Spread	320791
9	8	Gula Malacca	227088
10	9	Louisiana Fiery Hot Pepper Sauce	218566
11	10	Louisiana Hot Spiced Okra	209321
12			

Jan Feb

	A	B	C
1	Ranking	ProductName	Sales
2	1	Northwoods Cranberry Sauce	730428
3	2	Original Frankfurter grüne So?e	569142
4	3	Chef Anton's Cajun Seasoning	561817
5	4	Genen Shouyu	325843
6	5	Louisiana Fiery Hot Pepper Sauce	306817
7	6	Sasquatch Ale	276420
8	7	Valkoinen suklaa	246383
9	8	Zaanse koeken	244454
10	9	Camembert Pierrot	213016
11	10	Flotemysost	175908
12			

Jan Feb

Find out the sales data of the products that make the top 10 in both January and February (the ranking and sales in January are required only):

```
=spl("=[E(?1),E(?2)].merge@oi(ProductName)",Jan!A1:C11,Feb!A1:C11)
```

	A	B	C
1	Ranking	ProductName	Sales
2	1	Sasquatch Ale	774372
3	4	Chef Anton's Cajun Seasoning	637046
4	6	Genen Shouyu	508177
5	9	Louisiana Fiery Hot Pepper Sauce	218566

Find out the sales data of products that make the top 10 once or more (the ranking and sales that appear for the first time are required only):

=spl("=[E(?1),E(?2)].merge@ou(ProductName)",Jan!A1:C11,Feb!A1:C11)

	A	B	C
1	Ranking	ProductName	Sales
2	1	Sasquatch Ale	774372
3	2	Steeleye Stout	736366
4	3	Aniseed Syrup	706337
5	4	Chef Anton's Cajun Seasoning	637046
6	5	Chef Anton's Gumbo Mix	566448
7	6	Genen Shouyu	508177
8	7	Grandma's Boysenberry Spread	320791
9	8	Gula Malacca	227088
10	9	Louisiana Fiery Hot Pepper Sauce	218566
11	10	Louisiana Hot Spiced Okra	209321
12	1	Northwoods Cranberry Sauce	730428
13	2	Original Frankfurter grüne So?e	569142
14	7	Valkoinen suklaa	246383
15	8	Zaanse koeken	244454
16	9	Camembert Pierrot	213016
17	10	Flotemysost	175908

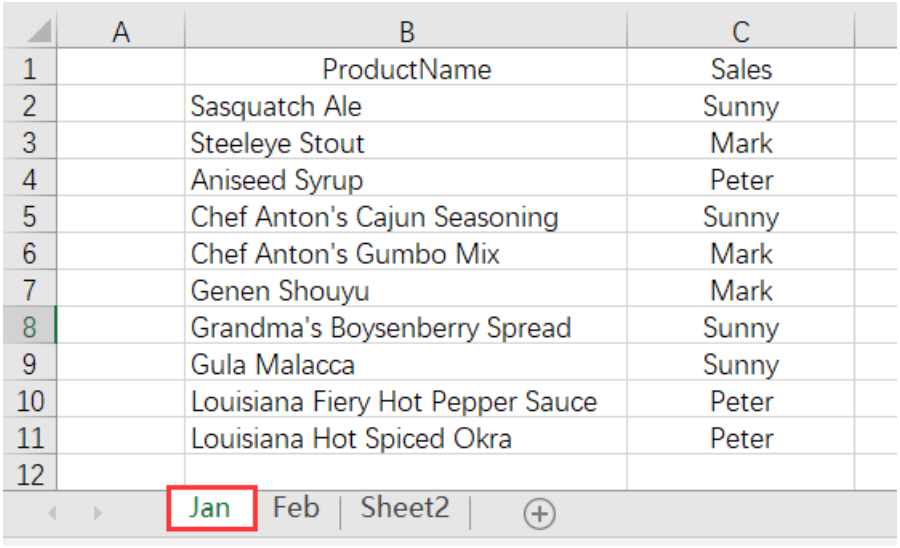
Find out the sales data of products that make the top 10 in January but fail to make the top 10 in February:

=spl("=[E(?1),E(?2)].merge@od(ProductName)",Jan!A1:C11,Feb!A1:C11)

	A	B	C
1	Ranking	ProductName	Sales
2	2	Steeleye Stout	736366
3	3	Aniseed Syrup	706337
4	5	Chef Anton's Gumbo Mix	566448
5	7	Grandma's Boysenberry Spread	320791
6	8	Gula Malacca	227088
7	10	Louisiana Hot Spiced Okra	209321

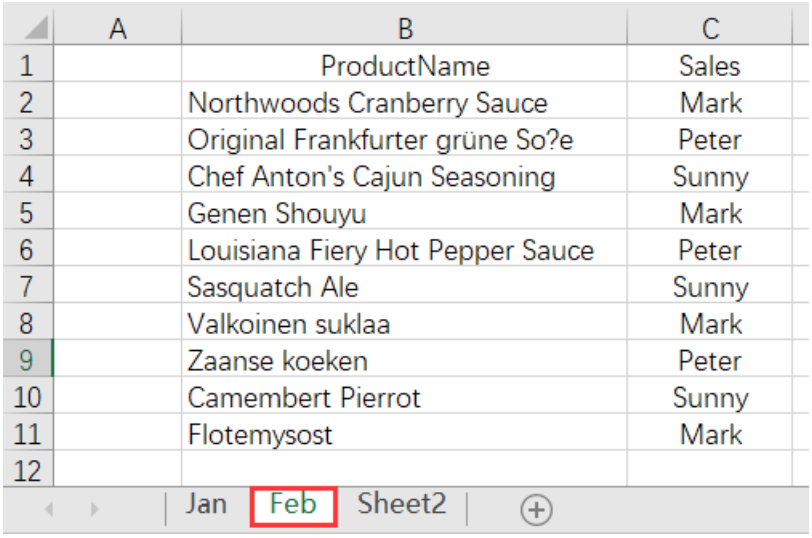
8.4 Intersection, union and difference in the case of row-based data - two sets - by whole row

The following tables list the data of the products and salespersons that make the top 10 by sales in January and February:



	A	B	C
1		ProductName	Sales
2		Sasquatch Ale	Sunny
3		Steeleye Stout	Mark
4		Aniseed Syrup	Peter
5		Chef Anton's Cajun Seasoning	Sunny
6		Chef Anton's Gumbo Mix	Mark
7		Genen Shouyu	Mark
8		Grandma's Boysenberry Spread	Sunny
9		Gula Malacca	Sunny
10		Louisiana Fiery Hot Pepper Sauce	Peter
11		Louisiana Hot Spiced Okra	Peter
12			

Jan | Feb | Sheet2



	A	B	C
1		ProductName	Sales
2		Northwoods Cranberry Sauce	Mark
3		Original Frankfurter grüne So?e	Peter
4		Chef Anton's Cajun Seasoning	Sunny
5		Genen Shouyu	Mark
6		Louisiana Fiery Hot Pepper Sauce	Peter
7		Sasquatch Ale	Sunny
8		Valkoinen suklaa	Mark
9		Zaanse koeken	Peter
10		Camembert Pierrot	Sunny
11		Flotemysost	Mark
12			

Jan | Feb | Sheet2

Find out the data of products and salespersons that make the top 10 in both January & February.

```
=spl("=[E(?1),E(?2)].merge@oi()",Jan!B1:C11,Feb!B1:C11)
```

	A	B
1	ProductName	Sales
2	Sasquatch Ale	Sunny
3	Chef Anton's Cajun Seasoning	Sunny
4	Genen Shouyu	Mark
5	Louisiana Fiery Hot Pepper Sauce	Peter

Find out the data of the products and salespersons that make the top 10 once or more.

=spl("=[E(?1),E(?2)].merge@ou()",Jan!B1:C11,Feb!B1:C11)

	A	B
1	ProductName	Sales
2	Sasquatch Ale	Sunny
3	Steeleye Stout	Mark
4	Aniseed Syrup	Peter
5	Chef Anton's Cajun Seasoning	Sunny
6	Chef Anton's Gumbo Mix	Mark
7	Genen Shouyu	Mark
8	Grandma's Boysenberry Spread	Sunny
9	Gula Malacca	Sunny
10	Louisiana Fiery Hot Pepper Sauce	Peter
11	Louisiana Hot Spiced Okra	Peter
12	Northwoods Cranberry Sauce	Mark
13	Original Frankfurter grüne So?e	Peter
14	Valkoinen suklaa	Mark
15	Zaanse koeken	Peter
16	Camembert Pierrot	Sunny
17	Flotemysost	Mark

Find out the data of products and salespersons that make the top 10 in January but fail to make the top 10 in February:

=spl("=[E(?1),E(?2)].merge@od()",Jan!B1:C11,Feb!B1:C11)

	A	B
1	ProductName	Sales
2	Steeleye Stout	Mark
3	Aniseed Syrup	Peter
4	Chef Anton's Gumbo Mix	Mark
5	Grandma's Boysenberry Spread	Sunny
6	Gula Malacca	Sunny
7	Louisiana Hot Spiced Okra	Peter

Notes:

The merge() function without parameter means the whole row will be taken as the matching criterion, and the merge() function with parameter means the parameter value will be taken as the matching criterion.

8.5 Intersection, union and difference in the case of row-based data - multiple sets

There is a file top10Sales.xlsx, which lists the sales data of the top 10 products by sales in the first couple of months of the year (the number of months will increase over time):

	A	B	C
1	Ranking	ProductName	Sales
2	1	Sasquatch Ale	774372
3	2	Steeleye Stout	736366
4	3	Aniseed Syrup	706337
5	4	Chef Anton's Cajun Seasoning	637046
6	5	Chef Anton's Gumbo Mix	566448
7	6	Genen Shouyu	508177
8	7	Grandma's Boysenberry Spread	320791
9	8	Gula Malacca	227088
10	9	Louisiana Fiery Hot Pepper Sauce	218566
11	10	Louisiana Hot Spiced Okra	209321

Jan | Feb | Mar | Apr

	A	B	C
1	Ranking	ProductName	Sales
2	1	Northwoods Cranberry Sauce	730428
3	2	Original Frankfurter grüne So?e	569142
4	3	Chef Anton's Cajun Seasoning	561817
5	4	Genen Shouyu	325843
6	5	Louisiana Fiery Hot Pepper Sauce	306817
7	6	Sasquatch Ale	276420
8	7	Chef Anton's Gumbo Mix	246383
9	8	Zaanse koeken	244454
10	9	Camembert Pierrot	213016
11	10	Flotemysost	175908

Jan | Feb | Mar | Apr

	A	B	C
1	Ranking	ProductName	Sales
2	1	Aniseed Syrup	889521
3	2	Chef Anton's Cajun Seasoning	838437
4	3	Chef Anton's Gumbo Mix	812739
5	4	Sasquatch Ale	553979
6	5	Valkoinen suklaa	374837
7	6	Zaanse koeken	371306
8	7	Queso Manchego La Pastora	199903
9	8	Raclette Courdavault	114641
10	9	Filo Mix	64488
11	10	Gnocchi di nonna Alice	47198

Jan | Feb | Mar | Apr

	A	B	C
1	Ranking	ProductName	Sales
2	1	Chef Anton's Gumbo Mix	974543
3	2	Sasquatch Ale	960785
4	3	Gustaf's Kn?ckebr?d	939094
5	4	Ravioli Angelo	839508
6	5	Singaporean Hokkien Fried Mee	679543
7	6	Tofu	655362
8	7	Uncle Bob's Organic Dried Pears	510849
9	8	Boston Crab Meat	495085
10	9	Carnarvon Tigers	356539
11	10	Escargots de Bourgogne	37989

Jan | Feb | Mar | Apr

Find out the products that make the top 10 in every month of these months:

	A
1	=file("top10Sales.xlsx").xlsopen()
2	=A1.(A1.xlsimport@t(;stname)).merge@oi(ProductName)

Index	Ranking	ProductName	Sales
1	1	Sasquatch Ale	774372
2	5	Chef Anton's Gumbo Mix	566448

Find out the products that make the top 10 once or more:

	A
1	=file("top10Sales.xlsx").xlsopen()
2	=A1.(A1.xlsimport@t(;stname)).merge@ou(ProductName)

Index	Ranking	ProductName	Sales
1	1	Sasquatch Ale	774372
2	2	Steeleye Stout	736366
3	3	Aniseed Syrup	706337
4	4	Chef Anton's Cajun Seasoning	637046
5	5	Chef Anton's Gumbo Mix	566448
6	6	Genen Shouyu	508177
7	7	Grandma's Boysenberry Spread	320791
8	8	Gula Malacca	227088
9	9	Louisiana Fiery Hot Pepper Sauce	218566
10	10	Louisiana Hot Spiced Okra	209321
11	1	Northwoods Cranberry Sauce	730428
12	2	Original Frankfurter grüne So?e	569142
13	8	Zaanse koeken	244454
14	9	Camembert Pierrot	213016
15	10	Flotemysost	175908
16	5	Valkoinen suklaa	374837
17	7	Queso Manchego La Pastora	199903
18	8	Raclette Courdavault	114641
19	9	Filo Mix	64488
20	10	Gnocchi di nonna Alice	47198
21	3	Gustaf's Kn?ckebr?d	939094
22	4	Ravioli Angelo	839508
23	5	Singaporean Hokkien Fried Mee	679543
24	6	Tofu	655362

Find out the products that make the top 10 in January but fail to make the top 10 in any of other months:

	A
1	=file("top10Sales.xlsx").xlsopen()
2	=A1.(A1.xlsimport@t(;stname)).merge@od(ProductName)

Ranking	ProductName	Sales
2	Steeleye Stout	736366
7	Grandma's Boysenberry Spread	320791
8	Gula Malacca	227088
10	Louisiana Hot Spiced Okra	209321

8.6 Judge equality of sets when order is considered

Here below is a monthly sales ranking table for the first half of the year:

	A	B	C	D
1	Month	Ranking	ProductName	Sales
2	Jan	1	Sasquatch Ale	774372
3	Jan	2	Steeleye Stout	736366
4	Jan	3	Aniseed Syrup	706337
5	Feb	1	Steeleye Stout	730428
6	Feb	2	Aniseed Syrup	569142
7	Feb	3	Sasquatch Ale	561817
8	Mar	1	Aniseed Syrup	889521
9	Mar	2	Sasquatch Ale	838437
10	Mar	3	Steeleye Stout	812739
11	Apr	1	Sasquatch Ale	974543
12	Apr	2	Aniseed Syrup	960785
13	Apr	3	Steeleye Stout	939094
14	May	1	Steeleye Stout	889521
15	May	2	Sasquatch Ale	838437
16	May	3	Aniseed Syrup	812739
17	June	1	Aniseed Syrup	974543
18	June	2	Steeleye Stout	960785
19	June	3	Sasquatch Ale	939094

Find out the month whose products ranking by sales is the same with the products ranking by total sales in the first half of the year:

	A
1	=E('A1:D19')
2	=A1.groups(ProductName;sum(Sales):Sales).sort(Sales:-1).(ProductName)
3	return A1.group@o(Month).select(~.(ProductName)==A2).id(Month)

A1: Convert the passed-in data to a two-dimensional table sequence.

A2: Calculate the products ranking by total sales in the first half of the year.

A3: Find out the month whose products ranking by sales is the same with the products ranking by total sales in the first half of the year through the operation for judgement of the equality of sets.

Result: May

8.7 Judge belongingness of sets when order is considered

Here below is a monthly sales ranking table for the first quarter:

	A	B	C	D
1	Month	Ranking	ProductName	Sales
2	Jan	1	Sasquatch Ale	774372
3	Jan	2	Steeleye Stout	736366
4	Jan	3	Chef Anton's Cajun Seasoning	706337
5	Jan	4	Aniseed Syrup	637046
6	Jan	5	Chef Anton's Gumbo Mix	566448
7	Feb	1	Northwoods Cranberry Sauce	730428
8	Feb	2	Original Frankfurter grüne So?e	569142
9	Feb	3	Chef Anton's Cajun Seasoning	561817
10	Feb	4	Genen Shouyu	325843
11	Feb	5	Louisiana Fiery Hot Pepper Sauce	306817
12	Mar	1	Aniseed Syrup	889521
13	Mar	2	Chef Anton's Cajun Seasoning	838437
14	Mar	3	Chef Anton's Gumbo Mix	812739
15	Mar	4	Sasquatch Ale	553979
16	Mar	5	Valkoinen suklaa	374837

Find out the month that contains the top 3 products by total sales in the first quarter (same ranking order is required):

	A
1	=E('A1:D16')
2	=A1.groups(ProductName;sum(Sales):Sales).sort(Sales:-1).(ProductName)(to(3))
3	return A1.group@o(Month).select(~.(ProductName).pos@c(A2)!=null).id(Month)

A1: Convert the passed-in data to a two-dimensional table sequence.

A2: Calculate the top 3 products by total sales in the first quarter.

A3: When the pos@c() judges the belongingness of sets, the order of members should be the same.

Result: Jan

8.8 Judge equality of sets when order is ignored

Here below is a monthly sales ranking table for the first quarter:

	A	B	C	D
1	Month	Ranking	ProductName	Sales
2	Jan	1	Sasquatch Ale	774372
3	Jan	2	Steeleye Stout	736366
4	Jan	3	Aniseed Syrup	706337
5	Feb	1	Northwoods Cranberry Sauce	730428
6	Feb	2	Original Frankfurter grüne So?e	569142
7	Feb	3	Chef Anton's Cajun Seasoning	561817
8	Mar	1	Aniseed Syrup	889521
9	Mar	2	Chef Anton's Cajun Seasoning	838437
10	Mar	3	Chef Anton's Gumbo Mix	812739

Find out the month that contains the top 3 products by total sales in the first quarter:

	A
1	=E('A1:D10')
2	=A1.groups(ProductName;sum(Sales):Sales).sort(Sales:-1).(ProductName)(to(3))
3	return A1.group@o(Month).select(~.(ProductName).pos(A2)!=null).id(Month)

A1: Convert the passed-in data to a two-dimensional table sequence.

A2: Calculate the top 3 products by total sales in the first quarter.

A3: When the number of set members is the same, using the pos function can judge the equality of sets in the case that the order is ignored.

Result: Mar

8.9 Judging belongingness of sets when order is ignored

Here below is a monthly sales ranking table for the first quarter:

	A	B	C	D
1	Month	Ranking	ProductName	Sales
2	Jan	1	Sasquatch Ale	774372
3	Jan	2	Steeleye Stout	736366
4	Jan	3	Aniseed Syrup	706337
5	Jan	4	Chef Anton's Cajun Seasoning	637046
6	Jan	5	Chef Anton's Gumbo Mix	566448
7	Feb	1	Northwoods Cranberry Sauce	730428
8	Feb	2	Original Frankfurter grüne So?e	569142
9	Feb	3	Chef Anton's Cajun Seasoning	561817
10	Feb	4	Genen Shouyu	325843
11	Feb	5	Louisiana Fiery Hot Pepper Sauce	306817
12	Mar	1	Aniseed Syrup	889521
13	Mar	2	Chef Anton's Cajun Seasoning	838437
14	Mar	3	Chef Anton's Gumbo Mix	812739
15	Mar	4	Sasquatch Ale	553979
16	Mar	5	Valkoinen suklaa	374837

Find out the month that contains the top 3 products by total sales in the first quarter:

	A
1	=E('A1:D16')
2	=A1.groups(ProductName;sum(Sales):Sales).sort(Sales:-1).(ProductName)(to(3))
3	return A1.group@o(Month).select(~.(ProductName).pos(A2)!=null).id(Month)

A1: Convert the passed-in data to a two-dimensional table sequence.

A2: Calculate the top 3 products by total sales in the first quarter.

A3: Find out the month that contains the top 3 products in the first quarter through the operation for judgement of belongingness of sets.

Result: Jan, Mar

Chapter 9 **Judgment, counting and deleting of duplicate data**

9.1 **Judge duplication of simple members**

Here below a data table of names:

	A	B
1	Name	Dup
2	Rebecca	
3	Ashley	
4	Rachel	
5	Emily	
6	Ashley	

Judge whether there are duplicate names, if so, fill in 1 in corresponding Dup field, otherwise fill in 0. Enter the formula in cell B2:

```
=spl("=if(?1.conj().select(~==?2).count()>1,1,0)",A$2:A$6,A2)
```

Then drag B2 down to every relevant row:

	A	B
1	Name	Dup
2	Rebecca	0
3	Ashley	1
4	Rachel	0
5	Emily	0
6	Ashley	1
7		

9.2 **Judge duplication of row-based data - by key column**

Here below is a personnel data table:

	A	B	C	D	E
1	ID	Name	Gender	Birthday	Dup
2	1	Rebecca	Male	1972/11/19	
3	2	Ashley	Female	1979/1/22	
4	3	Rachel	Male	1978/10/13	
5	4	Emily	Female	1982/11/24	
6	5	Ashley	Male	1992/12/5	
7	6	Rachel	Male	1978/10/13	

Judge whether there are duplicate data, if so, fill in 1 in corresponding Dup field, otherwise fill in 0 (the judging basis is that the data in the Name field are the same). Enter the formula in cell E2:

```
=spl("=if(?1.conj().select(~==?2).count()>1,1,0)",B$2:B$7,B2)
```

Then drag E2 down to every relevant row:

	A	B	C	D	E
1	ID	Name	Gender	Birthday	Dup
2	1	Rebecca	Male	1972/11/19	0
3	2	Ashley	Female	1979/1/22	1
4	3	Rachel	Male	1978/10/13	1
5	4	Emily	Female	1982/11/24	0
6	5	Ashley	Male	1992/12/5	1
7	3	Rachel	Male	1978/10/13	1
8					

9.3 **Judge duplication of row-based data - by whole row**

Here below is a personnel data table:

	A	B	C	D	E
1	ID	Name	Gender	Birthday	
2	1	Rebecca	Male	1972/11/19	
3	2	Ashley	Female	1979/1/22	
4	3	Rachel	Male	1978/10/13	
5	4	Emily	Female	1982/11/24	
6	5	Ashley	Male	1992/12/5	
7	3	Rachel	Male	1978/10/13	

Judge whether there are duplicate whole-row data, if so, fill in 1 in corresponding Dup field, otherwise fill in 0. Calculate in the cell E1:

	A
1	=E('A1:D7').derive(:Dup)
2	=A1.group(ID,Name,Gender,Birthday).run(a=if(~.len()>1,1,0),~.run(Dup=a))
3	return A1.new(Dup)

	A	B	C	D	E
1	ID	Name	Gender	Birthday	Dup
2	1	Rebecca	Male	1972/11/19	0
3	2	Ashley	Female	1979/1/22	0
4	3	Rachel	Male	1978/10/13	1
5	4	Emily	Female	1982/11/24	0
6	5	Ashley	Male	1992/12/5	0
7	3	Rachel	Male	1978/10/13	1

9.4 Count number of repetitions of simple members

Here below a data table of names:

	A	B
1	Name	Num
2	Rebecca	
3	Ashley	
4	Rachel	
5	Emily	
6	Ashley	

To count the number of duplicate names, enter the formula in cell B2:

```
=spl("=?1.conj().select(~==?2).count()",A$2:A$6,A2)
```

Then drag B2 down to every relevant row:

	A	B	C
1	Name	Num	
2	Rebecca	1	
3	Ashley	2	
4	Rachel	1	
5	Emily	1	
6	Ashley	2	
7			

9.5 Count number of repetitions of row-based data - by key column

Here below is a personnel data table:

	A	B	C	D	E
1	ID	Name	Gender	Birthday	Num
2	1	Rebecca	Male	1972/11/19	
3	2	Ashley	Female	1979/1/22	
4	3	Rachel	Male	1978/10/13	
5	4	Emily	Female	1982/11/24	
6	5	Ashley	Male	1992/12/5	
7	3	Rachel	Male	1978/10/13	

To count the number of duplicate data (the judging basis is that the data in the Name field are the same), enter the formula in the cell E2:

```
=spl("=?1.conj().select(~==?2).count()",B$2:B$7,B2)
```

Then drag E2 down to every relevant row:

	A	B	C	D	E
1	ID	Name	Gender	Birthday	Num
2	1	Rebecca	Male	1972/11/19	1
3	2	Ashley	Female	1979/1/22	2
4	3	Rachel	Male	1978/10/13	2
5	4	Emily	Female	1982/11/24	1
6	5	Ashley	Male	1992/12/5	2
7	3	Rachel	Male	1978/10/13	2
8					

9.6 Count number of repetitions of row-based data - by whole row

Here below is a personnel data table:

	A	B	C	D	E
1	ID	Name	Gender	Birthday	
2	1	Rebecca	Male	1972/11/19	
3	2	Ashley	Female	1979/1/22	
4	3	Rachel	Male	1978/10/13	
5	4	Emily	Female	1982/11/24	
6	5	Ashley	Male	1992/12/5	
7	3	Rachel	Male	1978/10/13	

To count the number of duplicate whole-row data, calculate in the cell E1:

	A
1	=E('A1:D7').derive(:Num)
2	=A1.group(ID,Name,Gender,Birthday).run(a=~.len(),~.run(Num=a))
3	return A1.new(Num)

	A	B	C	D	E
1	ID	Name	Gender	Birthday	Num
2	1	Rebecca	Male	1972/11/19	1
3	2	Ashley	Female	1979/1/22	1
4	3	Rachel	Male	1978/10/13	2
5	4	Emily	Female	1982/11/24	1
6	5	Ashley	Male	1992/12/5	1
7	3	Rachel	Male	1978/10/13	2

9.7 **Deduplication of simple data**

There is a data table:

	A
1	Name
2	Rebecca
3	Ashley
4	Rachel
5	Emily
6	Ashley
7	Nancy
8	Andrew
9	Janet
10	Margaret
11	Steven
12	Michael
13	Robert
14	Laura
15	Anne
16	Margaret
17	Nancy

Deduplicate the data:

```
=spl("=?1.id()",A2:A17)
```

Andrew
Anne
Ashley
Emily
Janet
Laura
Margaret
Michael
Nancy
Rachel
Rebecca
Robert
Steven

9.8 **Deduplication of row-based data - by key column**

There is a data table:

	A	B	C	D
1	ID	Name	Gender	Birthday
2	1	Rebecca	Male	1972/11/19
3	2	Ashley	Female	1979/1/22
4	3	Rachel	Male	1978/10/13
5	4	Emily	Female	1982/11/24
6	5	Ashley	Male	1992/12/5
7	3	Rachel	Male	1978/10/13

Find out the data that are not duplicate and the first duplicate data in the Name column:

=spl("=E(?1).group(Name).(~(1))",A1:D7)

ID	Name	Gender	Birthday
2	Ashley	Female	1979/1/22
4	Emily	Female	1982/11/24
3	Rachel	Male	1978/10/13
1	Rebecca	Male	1972/11/19

9.9 **Deduplication of row-based data - by whole row**

There is a data table:

	A	B	C	D
1	ID	Name	Gender	Birthday
2	1	Rebecca	Male	1972/11/19
3	2	Ashley	Female	1979/1/22
4	3	Rachel	Male	1978/10/13
5	4	Emily	Female	1982/11/24
6	5	Ashley	Male	1992/12/5
7	3	Rachel	Male	1978/10/13

Find out the whole-row data that are not duplicate and the first duplicate whole-row data:

=spl("=E(?1).group(ID,Name,Gender,Birthday).(~(1))",A1:D7)

ID	Name	Gender	Birthday
1	Rebecca	Male	1972/11/19
2	Ashley	Female	1979/1/22
3	Rachel	Male	1978/10/13
4	Emily	Female	1982/11/24
5	Ashley	Male	1992/12/5

9.10 **Deduplication of simple data - keeping order**

There is a data table:

	A
1	Name
2	Rebecca
3	Ashley
4	Rachel
5	Emily
6	Ashley
7	Nancy
8	Andrew
9	Janet
10	Margaret
11	Steven
12	Michael
13	Robert
14	Laura
15	Anne
16	Margaret
17	Nancy

Duplicate the data, and keep the original order:

```
=spl("=?1.id@u()",A2:A17)
```

Rebecca
Ashley
Rachel
Emily
Nancy
Andrew
Janet
Margaret
Steven
Michael
Robert
Laura
Anne

9.11 **Deduplication of row-based data - by key column - keeping order**

There is a data table:

	A	B	C	D
1	ID	Name	Gender	Birthday
2	1	Rebecca	Male	1972/11/19
3	2	Ashley	Female	1979/1/22
4	3	Rachel	Male	1978/10/13
5	4	Emily	Female	1982/11/24
6	5	Ashley	Male	1992/12/5
7	3	Rachel	Male	1978/10/13

Find out the data that are not duplicate and the first duplicate data in the Name column:

=spl("=E(?1).group**@u**(Name).(~(1))",A1:D7)

ID	Name	Gender	Birthday
1	Rebecca	Male	1972/11/19
2	Ashley	Female	1979/1/22
3	Rachel	Male	1978/10/13
4	Emily	Female	1982/11/24

9.12 **Deduplication of row-based data - by whole row - keeping order**

There is a data table:

	A	B	C	D
1	ID	Name	Gender	Birthday
2	1	Rebecca	Male	1972/11/19
3	2	Ashley	Female	1979/1/22
4	3	Rachel	Male	1978/10/13
5	4	Emily	Female	1982/11/24
6	5	Ashley	Male	1992/12/5
7	3	Rachel	Male	1978/10/13

Find out the whole-row data that are not duplicate, and the first duplicate whole-row data:

=spl("=E(?1).group**@u**(ID,Name,Gender,Birthday).(~(1))",A1:D7)

ID	Name	Gender	Birthday
1	Rebecca	Male	1972/11/19
2	Ashley	Female	1979/1/22
3	Rachel	Male	1978/10/13
4	Emily	Female	1982/11/24
5	Ashley	Male	1992/12/5

9.13 Filter by number of repetitions

The following table lists the sales data of the top 5 products by monthly sales:

	A	B	C	D
1	Month	Ranking	ProductName	Sales
2	Jan	1	Sasquatch Ale	774372
3	Jan	2	Steeleye Stout	736366
4	Jan	3	Aniseed Syrup	706337
5	Jan	4	Chef Anton's Cajun Seasoning	637046
6	Jan	5	Chef Anton's Gumbo Mix	566448
7	Feb	1	Northwoods Cranberry Sauce	730428
8	Feb	2	Original Frankfurter grüne So?e	569142
9	Feb	3	Chef Anton's Cajun Seasoning	561817
10	Feb	4	Genen Shouyu	325843
11	Feb	5	Louisiana Fiery Hot Pepper Sauce	306817
12	Mar	1	Aniseed Syrup	889521
13	Mar	2	Chef Anton's Cajun Seasoning	838437
14	Mar	3	Chef Anton's Gumbo Mix	812739
15	Mar	4	Sasquatch Ale	553979
16	Mar	5	Valkoinen suklaa	374837
17	Apr	1	Chef Anton's Gumbo Mix	974543
18	Apr	2	Sasquatch Ale	960785
19	Apr	3	Gustaf's Kn?ckebr?d	939094
20	Apr	4	Ravioli Angelo	839508
21	Apr	5	Singaporean Hokkien Fried Mee	679543

Find out the data of products that appears three or more times:

```
=spl("=E(?1).group(ProductName).select(~.len()>2).conj()",A1:D21)
```

Month	Ranking	ProductName	Sales
Jan	4	Chef Anton's Cajun Seasoning	637046
Feb	3	Chef Anton's Cajun Seasoning	561817
Mar	2	Chef Anton's Cajun Seasoning	838437
Jan	5	Chef Anton's Gumbo Mix	566448
Mar	3	Chef Anton's Gumbo Mix	812739
Apr	1	Chef Anton's Gumbo Mix	974543
Jan	1	Sasquatch Ale	774372
Mar	4	Sasquatch Ale	553979
Apr	2	Sasquatch Ale	960785

9.14 **Delete data that can be paired**

Here below is a data table:

	A	B	C	D
1	ID	Name	Item	Value
2	1001	Alice	eye	60
3	1001	Alice	eye	-60
4	1002	Tom	nose	30
5	1003	Jerry	stomach	70
6	1003	Jerry	stomach	-70
7	1003	Jerry	hand	50
8	1004	Bob	arm	100
9	1005	Jack	leg	25
10	1005	Jack	leg	-25

Delete the rows that have the same data in the fields of ID, Name and Item, and whose positive and negative values in Value field can be completely offset.

```
=spl("=E(?1).group(ID,Name,Item).select(~.sum(Value)!=0).conj()",A1:D10)
```

ID	Name	Item	Value
1002	Tom	nose	30
1003	Jerry	hand	50
1004	Bob	arm	100

Chapter 10 **Ranking and Sorting**

10.1 **Sorting of simple members**

Here below is a data table:

	A	B
1	58	
2	5	
3	19	
4	14	
5	81	
6	45	
7	7	
8	32	
9	27	
10	36	
11		

Sort in ascending order:

```
=spl("=?.sort()",A1:A10)
```

5
7
14
19
27
32
36
45
58
81

Sort in descending order:

```
=spl("=?.sort(~:-1)",A1:A10)
```

81
58
45
36
32
27
19
14
7
5

10.2 **Sorting of row-based data**

Here below is a data table:

	A	B	C
1	Class	Name	TotalScore
2	1	Tom	256
3	2	John	267
4	2	Joan	238
5	1	Rocky	265
6	2	Ham	220
7	1	Kate	232
8	1	Rose	238
9	2	Nomy	239
10	1	Neil	271
11	2	Jack	232
12	1	Joe	229

Sort in ascending order by TotalScore:

```
=spl("=E(?).sort(TotalScore)",A1:C12)
```

Class	Name	TotalScore
2	Ham	220
1	Joe	229
1	Kate	232
2	Jack	232
2	Joan	238
1	Rose	238
2	Nomy	239
1	Tom	256
1	Rocky	265
2	John	267
1	Neil	271

Sort in descending order by TotalScore:

```
=spl("=E(?).sort(TotalScore:-1)",A1:C12)
```

Class	Name	TotalScore
1	Neil	271
2	John	267
1	Rocky	265
1	Tom	256
2	Nomy	239
2	Joan	238
1	Rose	238
1	Kate	232
2	Jack	232
1	Joe	229
2	Ham	220

10.3 **Sorting of row-based data - by combination of multiple columns**

Here below is a data table:

	A	B	C	D	E
1	Class	Name	Maths	English	PE
2	1	Tom	85	97	74
3	2	John	92	80	95
4	2	Joan	86	67	85
5	1	Rocky	95	95	75
6	2	Ham	92	75	53
7	1	Kate	83	99	50
8	1	Rose	95	62	81
9	2	Nomy	78	91	70
10	1	Neil	91	83	97
11	2	Jack	84	75	73
12	1	Joe	98	71	60

Sort in reverse order by the order combination of three columns (Maths, English, PE):

```
=spl("=E(?).sort(Maths:-1,English:-1,PE:-1)",A1:E12)
```

Class	Name	Maths	English	PE
1	Joe	98	71	60
1	Rocky	95	95	75
1	Rose	95	62	81
2	John	92	80	95
2	Ham	92	75	53
1	Neil	91	83	97
2	Joan	86	67	85
1	Tom	85	97	74
2	Jack	84	75	73
1	Kate	83	99	50
2	Nomy	78	91	70

10.4 **Sorting of row-based data - by expression**

Here below is a data table:

	A	B	C	D	E
1	Class	Name	Maths	English	PE
2	1	Tom	85	97	74
3	2	John	92	80	95
4	2	Joan	86	67	85
5	1	Rocky	95	95	75
6	2	Ham	92	75	53
7	1	Kate	83	99	50
8	1	Rose	95	62	81
9	2	Nomy	78	91	70
10	1	Neil	91	83	97
11	2	Jack	84	75	73
12	1	Joe	98	71	60

Sort in reverse order by the sum of three columns, Maths, English and PE:

```
=spl("=E(?).sort(Maths+English+PE:-1)",A1:E12)
```

Class	Name	Maths	English	PE
1	Neil	91	83	97
2	John	92	80	95
1	Rocky	95	95	75
1	Tom	85	97	74
2	Nomy	78	91	70
2	Joan	86	67	85
1	Rose	95	62	81
1	Kate	83	99	50
2	Jack	84	75	73
1	Joe	98	71	60
2	Ham	92	75	53

10.5 **Sort in group**

Here below is a data table:

	A	B	C
1	Class	Name	Maths
2	1	Tom	85
3	2	John	92
4	2	Joan	86
5	1	Rocky	95
6	2	Ham	92
7	1	Kate	83
8	1	Rose	95
9	2	Nomy	78
10	1	Neil	91
11	2	Jack	84
12	1	Joe	98

Sort the data (in ascending order) by Class, (in descending order) by Score:

```
=spl("=E(?1).sort(Class,Maths:-1)",A1:C12)
```

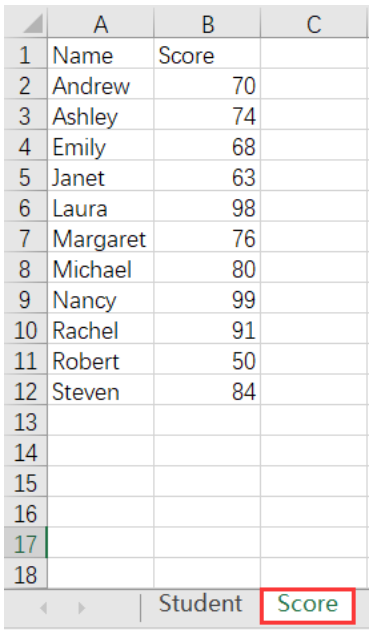
Class	Name	Maths
1	Joe	98
1	Rocky	95
1	Rose	95
1	Neil	91
1	Tom	85
1	Kate	83
2	John	92
2	Ham	92
2	Joan	86
2	Jack	84
2	Nomy	78

10.6 **Sort by specified order**

Here below is a data table:

	A	B	C
1	ID	Name	
2	1	Rebecca	
3	2	Ashley	
4	3	Rachel	
5	4	Emily	
6	5	Nancy	
7	6	Alice	
8	7	Andrew	
9	8	Janet	
10	9	Sunny	
11	10	Margaret	
12	11	Steven	
13	12	Michael	
14	13	Robert	
15	14	Laura	
16	15	Anne	
17	16	Peter	
18			

Student | Score

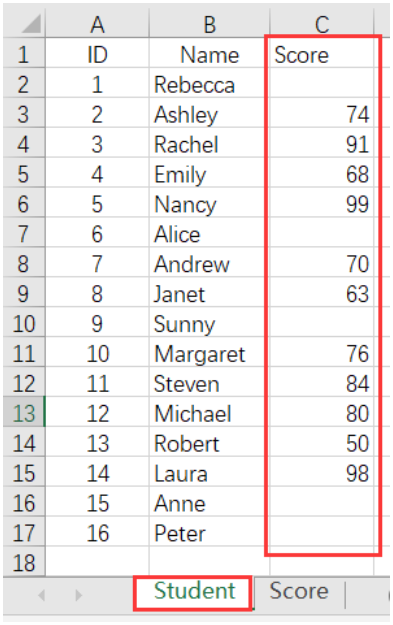


	A	B	C
1	Name	Score	
2	Andrew	70	
3	Ashley	74	
4	Emily	68	
5	Janet	63	
6	Laura	98	
7	Margaret	76	
8	Michael	80	
9	Nancy	99	
10	Rachel	91	
11	Robert	50	
12	Steven	84	
13			
14			
15			
16			
17			
18			

Student | Score

Please add one column Score on the right of the column Name in the Student sheet, and fill in the data of column Score of the Score sheet in the added column. Calculate in cell C1 of Student sheet:

	A
1	=E('A1:B17')
2	return E('Score!A1:B12').align(A1:Name,Name).new(~.Score)

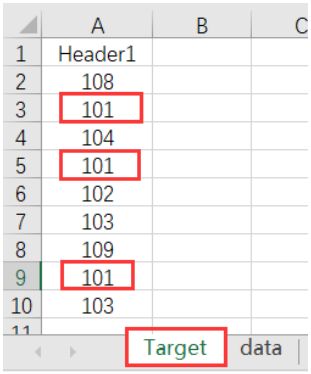


	A	B	C
1	ID	Name	Score
2	1	Rebecca	
3	2	Ashley	74
4	3	Rachel	91
5	4	Emily	68
6	5	Nancy	99
7	6	Alice	
8	7	Andrew	70
9	8	Janet	63
10	9	Sunny	
11	10	Margaret	76
12	11	Steven	84
13	12	Michael	80
14	13	Robert	50
15	14	Laura	98
16	15	Anne	
17	16	Peter	
18			

Student | Score

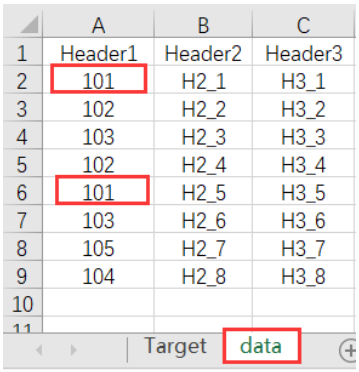
10.7 Sort by specified order in which duplicate values exist

There is a data table, and some index numbers are stored in the column Header1. In these index numbers, there may be duplicate values, as shown in the red boxes in the figure below, there are multiple 101s:



	A	B	C
1	Header1		
2	108		
3	101		
4	104		
5	101		
6	102		
7	103		
8	109		
9	101		
10	103		

Target data



	A	B	C
1	Header1	Header2	Header3
2	101	H2_1	H3_1
3	102	H2_2	H3_2
4	103	H2_3	H3_3
5	102	H2_4	H3_4
6	101	H2_5	H3_5
7	103	H2_6	H3_6
8	105	H2_7	H3_7
9	104	H2_8	H3_8
10			

Target data

Fill in the data of Header2 and Header3 of the data sheet in columns B and C of the Target sheet respectively. The first 101 of Target sheet corresponds to the data row of the first 101 of data sheet, and the second 101 corresponds to the data row of the second 101. Since there isn't the third 101 in data sheet, the third 101 in Target sheet will correspond to null value. Calculate in cell B1:

	A
1	=E('A1:A10').derive(:key)
2	=E('data!A1:C9').derive(:key)
3	=A1.group(Header1).run(~.run(key=Header1/"_"/#))
4	=A2.group(Header1).run(~.run(key=Header1/"_"/#))
5	return A2.align(A1:key,key).new(~.Header2,~.Header3)

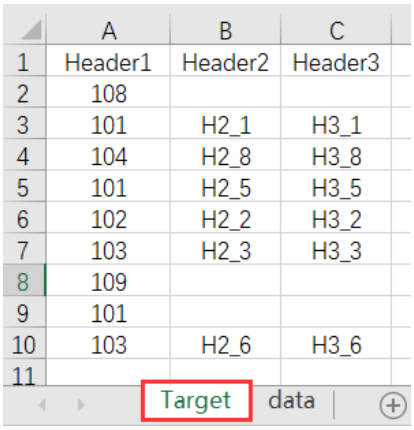
A1: Convert the passed-in data to a two-dimensional table sequence and add one column key to generate the key value used for alignment.

A2: Convert the passed-in data to a two-dimensional table sequence and add one column key to generate the key value used for alignment.

A3: Group A1 by Header1, loop through every group, and loop through every row in the group. Set the key value to: Header1 + underline + its serial number in the group.

A4: Same as A3.

A5: Align the keys in A1 by the order of the keys in A2, to generate a new data set to return.



	A	B	C
1	Header1	Header2	Header3
2	108		
3	101	H2_1	H3_1
4	104	H2_8	H3_8
5	101	H2_5	H3_5
6	102	H2_2	H3_2
7	103	H2_3	H3_3
8	109		
9	101		
10	103	H2_6	H3_6
11			

Target data

10.8 **Shuffle the data**

Here below is a data table:

	A	B	C	D
1	1	2	3	4
2	5	6	7	8
3	9	10	11	12
4	13	14	15	16
5	17	18	19	20

Shuffle the data of each row:

```
=spl("=clipboard(E(?1.(~.sort(rand())))",A1:D5)
```

2	3	1	4
8	7	6	5
10	12	9	11
15	16	14	13
18	19	17	20

Shuffle the data of each column:

```
=spl("=clipboard(E(transpose(transpose(?1).(~.sort(rand())))))",A1:D5)
```

13	2	11	20
9	6	15	16
5	18	3	4
17	14	7	12
1	10	19	8

Shuffle the data of the whole area:

```
=spl("=clipboard(E(?1.conj().sort(rand()).group(#%5)))",A1:D5)
```

7	13	1	11
5	3	2	8
9	18	20	4
16	14	19	12
6	15	17	10

Notes:

Only when Ctrl-Enter is pressed will the set returned by SPL() be filled in multiple cells, while other methods that trigger the calculation will only fill in the first member. In this example, the data are shuffled in a random way, and each calculation result is different, therefore, in this code, the result is copied to the clipboard, which can be pasted into the target cell after calculation.

10.9 **Ranking of simple members**

Here below is a data table:

	A	B
1	58	
2	5	
3	19	
4	14	
5	81	
6	45	
7	7	
8	32	
9	27	
10	36	

To calculate the ranking of members (in descending order), and fill in the results in corresponding cell of column B, enter in cell B1:

```
=spl("=?1.conj().rank@z(?2)",A$1:A$10,A1)
```

	A	B
1	58	2
2	5	
3	19	
4	14	
5	81	
6	45	
7	7	
8	32	
9	27	
10	36	

Drag B1 down to every relevant row:

	A	B
1	58	2
2	5	10
3	19	7
4	14	8
5	81	1
6	45	3
7	7	9
8	32	5
9	27	6
10	36	4
11		
12		

10.10 **Ranking of row-based data**

Here below is a data table:

	A	B	C	D
1	Class	Name	TotalScore	Rank
2	1	Tom	256	
3	2	John	267	
4	2	Joan	238	
5	1	Rocky	265	
6	2	Ham	220	
7	1	Kate	232	
8	1	Rose	238	
9	2	Nomy	239	
10	1	Neil	271	
11	2	Jack	232	
12	1	Joe	229	

To calculate the ranking of students by TotalScore (in descending order), and fill in the results in corresponding cell of column D, enter in cell D2:

```
=spl("=?1.conj().rank@z(?2)",C$2:C$12,C2)
```

	A	B	C	D
1	Class	Name	TotalScore	Rank
2	1	Tom	256	4
3	2	John	267	
4	2	Joan	238	
5	1	Rocky	265	
6	2	Ham	220	
7	1	Kate	232	
8	1	Rose	238	
9	2	Nomy	239	
10	1	Neil	271	
11	2	Jack	232	
12	1	Joe	229	

Drag D2 down to every relevant row:

	A	B	C	D
1	Class	Name	TotalScore	Rank
2	1	Tom	256	4
3	2	John	267	2
4	2	Joan	238	6
5	1	Rocky	265	3
6	2	Ham	220	11
7	1	Kate	232	8
8	1	Rose	238	6
9	2	Nomy	239	5
10	1	Neil	271	1
11	2	Jack	232	8
12	1	Joe	229	10
13				

10.11 **Ranking of row-based data - by combination of multiple columns**

Here below is a data table:

	A	B	C	D	E	F
1	Class	Name	Maths	English	PE	Rank
2	1	Tom	85	97	74	
3	2	John	92	80	95	
4	2	Joan	86	67	85	
5	1	Rocky	95	95	75	
6	2	Ham	92	75	53	
7	1	Kate	83	99	50	
8	1	Rose	95	62	81	
9	2	Nomy	78	91	70	
10	1	Neil	91	83	97	
11	2	Jack	84	75	73	
12	1	Joe	98	71	60	

To rank (in descending order) by the order combination of three columns Maths, English, and PE, enter in cell F2:

```
=spl("=?1.rank@z(?2.conj())",C$2:E$12,C2:E2)
```

	A	B	C	D	E	F
1	Class	Name	Maths	English	PE	Rank
2	1	Tom	85	97	74	8
3	2	John	92	80	95	
4	2	Joan	86	67	85	
5	1	Rocky	95	95	75	
6	2	Ham	92	75	53	
7	1	Kate	83	99	50	
8	1	Rose	95	62	81	
9	2	Nomy	78	91	70	
10	1	Neil	91	83	97	
11	2	Jack	84	75	73	
12	1	Joe	98	71	60	

Drag F2 down to every relevant row:

	A	B	C	D	E	F	G
1	Class	Name	Maths	English	PE	Rank	
2	1	Tom	85	97	74	8	
3	2	John	92	80	95	4	
4	2	Joan	86	67	85	7	
5	1	Rocky	95	95	75	2	
6	2	Ham	92	75	53	5	
7	1	Kate	83	99	50	10	
8	1	Rose	95	62	81	3	
9	2	Nomy	78	91	70	11	
10	1	Neil	91	83	97	6	
11	2	Jack	84	75	73	9	
12	1	Joe	98	71	60	1	
13							

10.12 **Ranking of row-based data - by expression**

Here below is a data table:

	A	B	C	D	E	F
1	Class	Name	Maths	English	PE	Rank
2	1	Tom	85	97	74	
3	2	John	92	80	95	
4	2	Joan	86	67	85	
5	1	Rocky	95	95	75	
6	2	Ham	92	75	53	
7	1	Kate	83	99	50	
8	1	Rose	95	62	81	
9	2	Nomy	78	91	70	
10	1	Neil	91	83	97	
11	2	Jack	84	75	73	
12	1	Joe	98	71	60	

To calculate the ranking (in descending order) by the sum of three columns Maths, English and PE, enter in cell F2:

```
=spl("=E(?1).(Maths+English+PE).rank@z(?2)",C$1:E$12,C2+D2+E2)
```

	A	B	C	D	E	F
1	Class	Name	Maths	English	PE	Rank
2	1	Tom	85	97	74	4
3	2	John	92	80	95	
4	2	Joan	86	67	85	
5	1	Rocky	95	95	75	
6	2	Ham	92	75	53	
7	1	Kate	83	99	50	
8	1	Rose	95	62	81	
9	2	Nomy	78	91	70	
10	1	Neil	91	83	97	
11	2	Jack	84	75	73	
12	1	Joe	98	71	60	

Drag F2 down to every relevant row:

	A	B	C	D	E	F	G
1	Class	Name	Maths	English	PE	Rank	
2	1	Tom	85	97	74	4	
3	2	John	92	80	95	2	
4	2	Joan	86	67	85	6	
5	1	Rocky	95	95	75	3	
6	2	Ham	92	75	53	11	
7	1	Kate	83	99	50	8	
8	1	Rose	95	62	81	6	
9	2	Nomy	78	91	70	5	
10	1	Neil	91	83	97	1	
11	2	Jack	84	75	73	8	
12	1	Joe	98	71	60	10	
13							

10.13 **Concatenate members with the same ranking**

Here below is a data table:

	A	B
1	Name	Maths
2	Tom	85
3	John	92
4	Joan	86
5	Rocky	95
6	Ham	92
7	Kate	83
8	Rose	95
9	Nomy	78
10	Neil	92
11	Jack	86
12	Joe	98

Rank the students by math score, and concatenate the names of students having the same score with comma:

```
=spl("=E(?1).group(Maths).(~.(Name).concat@c()).rvs().new(#:Rank,~:Name)",A1:B12)
```

Rank	Name
1	Joe
2	Rocky,Rose
3	John,Ham,Neil
4	Joan,Jack
5	Tom
6	Kate
7	Nomy

10.14 **Rank in group**

Here below is a data table:

	A	B	C	D	E
1	ID	Class	Name	Maths	
2	20220001	1	Tom	85	
3	20220002	2	John	92	
4	20220003	2	Joan	86	
5	20220004	1	Rocky	95	
6	20220005	2	Ham	92	
7	20220006	1	Kate	83	
8	20220007	1	Rose	95	
9	20220008	2	Nomy	78	
10	20220009	1	Neil	91	
11	20220010	2	Jack	84	
12	20220011	1	Joe	98	

To calculate the ranking of students in the class, and fill in the results in corresponding cell of column E, enter in cell E1:

	A
1	=E('B1:D12').derive(:RankInClass)
2	=A1.group(Class).run(a=~.ranks@z(Maths),~.run(RankInClass=a(#)))
3	return A1.new(RankInClass)

A1: Convert the data to a two-dimensional table sequence, and add one column RankInClass.

A2: Group by class, calculate ranking within each group, and assign the value to RankInClass.

A3: Return RankInClass.

	A	B	C	D	E
1	ID	Class	Name	Maths	RankInClass
2	20220001	1	Tom	85	5
3	20220002	2	John	92	1
4	20220003	2	Joan	86	3
5	20220004	1	Rocky	95	2
6	20220005	2	Ham	92	1
7	20220006	1	Kate	83	6
8	20220007	1	Rose	95	2
9	20220008	2	Nomy	78	5
10	20220009	1	Neil	91	4
11	20220010	2	Jack	84	4
12	20220011	1	Joe	98	1

Chapter 11 **Grouping and aggregating**

11.1 **Simple grouping**

Here below is a data table:

	A	B	C	D	E	F
1	Class	Name	Maths	English	PE	
2	1	Tom	85	97	74	
3	2	John	92	80	95	
4	2	Joan	86	67	85	
5	1	Rocky	95	95	75	
6	2	Ham	92	75	53	
7	1	Kate	83	99	50	
8	1	Rose	95	62	81	
9	2	Nomy	78	91	70	
10	1	Neil	91	83	97	
11	2	Jack	84	75	73	
12	1	Joe	98	71	60	
13						
14						

We want to group by Class, and calculate the average score of each subject of each class:

=spl("=E(?1).groups(Class;avg(Maths):Maths,avg(English):English,avg(PE):PE)",A1:E12)

Class	Maths	English	PE
1	91.2	84.5	72.8
2	86.4	77.6	75.2

11.2 **Group by combination of multiple columns**

Here below is a data table:

	A	B	C	D
1	Year	Quarter	Month	Sales
2	2020	1	1	7,588,420
3	2020	1	2	8,953,172
4	2020	1	3	6,276,185
5	2020	2	4	1,281,280
6	2020	2	5	2,686,043
7	2020	2	6	5,851,210
8	2020	3	7	292,254
9	2020	3	8	8,216,267
10	2020	3	9	5,143,192
11	2020	4	10	3,206,181
12	2020	4	11	5,536,126
13	2020	4	12	6,176,479
14	2021	1	1	7,319,085
15	2021	1	2	2,392,104
16	2021	1	3	446,324
17	2021	2	4	5,137,729
18	2021	2	5	9,663,806
19	2021	2	6	4,265,516
20	2021	3	7	8,447,330
21	2021	3	8	5,781,475
22	2021	3	9	1,917,614
23	2021	4	10	3,979,820
24	2021	4	11	4,250,158
25	2021	4	12	8,215,770

We want to group by Year and Quarter, and count the total sales of each quarter:

```
=spl("=E(?1).groups(Year,Quarter;sum(Sales):Sales)",A1:D25)
```

Year	Quarter	Sales
2020	1	22,817,777
2020	2	9,818,533
2020	3	13,651,713
2020	4	14,918,786
2021	1	10,157,513
2021	2	19,067,051
2021	3	16,146,419
2021	4	16,445,748

11.3 **Group by expression**

Here below is a data table:

	A	B	C
1	ID	Name	Sales
2	010010001	Tom	7,588,420
3	010010002	John	8,953,172
4	010010003	Joan	6,276,185
5	020010004	Rocky	1,281,280
6	020010005	Ham	2,686,043
7	020010006	Kate	5,851,210
8	020010007	Rose	292,254
9	030020008	Nomy	8,216,267
10	030020009	Neil	5,143,192
11	030020010	Jack	3,206,181
12	030020011	Joe	5,536,126
13	030020012	Peter	6,176,479
14	040020013	Sunny	7,319,085
15	040020014	Tiger	2,392,104
16	040020015	Alice	446,324
17	040020016	Cindy	5,137,729
18	040020017	Leon	9,663,806
19	050030018	Lily	4,265,516
20	050030019	Kevin	8,447,330
21	050030020	Shelly	5,781,475
22	050030021	Panzy	1,917,614
23	050030022	Maggie	3,979,820
24	050030023	Mark	4,250,158
25	050030024	Aileen	8,215,770

We want to group and aggregate by the first two digits of ID, and calculate the total sales of each group:

```
=spl("=E(?1).groups(left(ID,2):ID;sum(Sales):Sales)",A1:C25)
```

ID	Sales
01	22,817,777
02	10,110,787
03	28,278,245
04	24,959,048
05	36,857,683

11.4 **Group by segment**

Here below is a data table:

	A	B	C
1	ID	Name	Score
2	010010001	Tom	80
3	010010002	John	80
4	010010003	Joan	74
5	020010004	Rocky	86
6	020010005	Ham	90
7	020010006	Kate	70
8	020010007	Rose	63
9	030020008	Nomy	84
10	030020009	Neil	82
11	030020010	Jack	63
12	030020011	Joe	64
13	030020012	Peter	71
14	040020013	Sunny	61
15	040020014	Tiger	70
16	040020015	Alice	98
17	040020016	Cindy	93
18	040020017	Leon	85
19	050030018	Lily	75
20	050030019	Kevin	92
21	050030020	Shelly	77
22	050030021	Panzy	63
23	050030022	Maggie	57
24	050030023	Mark	68
25	050030024	Aileen	80

We want to count the number of people in each of the following score segments respectively: below 50, 50-60, 60-70, 70-80, 80-90 and above 90.

```
=spl("=E(?1).groups(if(Score<50:""<50"",Score>=50 && Score<60:""50-60"",Score>=60 && Score<70:""60-70"",Score>=70 && Score<80:""70-80"",Score>=80 && Score<90:""80-90"";"">=90""):Score;count(1):Num)",A1:C25)
```

Score	Num
50-60	1
60-70	6
70-80	6
80-90	7
>=90	4

11.5 **Enumeration grouping**

Here below is a data table:

	A	B
1	Goods	Sales
2	Apple	111417
3	Banana	66845
4	Orange	244168
5	Pork	326980
6	Beef	227928
7	Mutton	525107
8	Tissue	363607
9	Detergent	907357
10	Bread	939477
11	Milk	511058
12	Juice	40363
13	Cauliflower	227928
14	eggplant	525107

The task is to put Apple, Banana and Orange in a group and name it Fruits, put Pork, Beef and Mutton in a group and name it Meats, and put the rest of the goods in a group and name it Others, and then count the total sales of each group:

```
=spl("=E(?1).groups(if([""Apple"",""Banana"",""Orange""].pos(Goods)!=null:""Fruits"",
[""Pork"",""Beef"",""Mutton""].pos(Goods)!=null:""Meats""; ""Others""):Goods;
sum(Sales):Sales)", A1:B14)
```

Goods	Sales
Fruits	422430
Meats	1080015
Others	3514897

11.6 **Put every N members in a group**

Here below is a data table:

	A	B
1	01.01.2017 00:15	1480611
2	01.01.2017 00:30	2288469
3	01.01.2017 00:45	213178
4	01.01.2017 01:15	1499191
5	01.01.2017 01:30	944919
6	01.01.2017 01:45	1035472
7	01.01.2017 02:15	1579507
8	01.01.2017 02:30	316392
9	01.01.2017 02:45	1209184
10	01.01.2017 03:15	1385223
11	01.01.2017 03:30	678256
12	01.01.2017 03:45	1059992

We want to put every 4 rows in order in a group, and calculate the serial number of group and the sum of the values in the second column of each group:

```
=spl("=E@b(?1).groups((#-1)\4+1:ID;sum(#2):Total)",A1:B12)
```

ID	Total
1	5481449
2	3876290
3	4332655

To put every 4 rows in a group, we can group by the result of (#-1)\4+1, that is, subtract 1 from row number # to get the difference, divide the difference by 4 to get the quotient, and add 1 to the quotient. After grouping, we can count the sum of the values in the second column of each group.

11.7 Convert one-dimensional array to two-dimensional array

Here below is a data table:

	A	B	C	D	E	F	G
1	a	Column1	Column2	Column3	Column4	Column5	
2	b						
3	c						
4	d						
5	e						
6	f						
7	g						
8	h						
9	i						
10	j						
11	k						
12	l						
13	m						
14	n						
15	o						
16	p						
17	q						
18	r						
19	s						
20	t						
21	u						
22	v						
23	w						
24	x						
25	y						
26	z						
27							

We want to convert the data to a 5-column & N-row table with header, in the order of horizontal arrangement first and then vertical arrangement. To achieve this, enter the formula in cell B2:

```
=spl("=?1.conj().group((#-1)\5)",A1:A26)
```

	A	B	C	D	E	F	G
1	a	Column1	Column2	Column3	Column4	Column5	
2	b	a	b	c	d	e	
3	c	f	g	h	i	j	
4	d	k	l	m	n	o	
5	e	p	q	r	s	t	
6	f	u	v	w	x	y	
7	g	z					
8	h						
9	i						
10	j						
11	k						
12	l						
13	m						
14	n						
15	o						
16	p						
17	q						
18	r						
19	s						
20	t						
21	u						
22	v						
23	w						
24	x						
25	y						
26	z						
27							

Subtract 1 from serial number # to get the difference, and divide the difference by 5 to get the quotient. Grouping by this quotient can put every 5 numbers in a group.

11.8 Take adjacent data as grouping criteria

Here below is a data table:

	A	B
1	51	55.74
2	55	61.47
3	56	44.53
4	62	142.11
5	75	162.74
6	81	142.11
7	85	44.53
8	92	52.89
9	95	118.37
10	96	354.8
11	97	6.49
12	99	44.53
13		

The task is to group the data in the first column, and calculate the sum of the values in the second column of each group. Grouping criteria: when there is a difference of 5 or more between the current data and previous one, create a new group:

```
=spl("=E@b(?1).group@i(#1-#1[-1]>=5).new(#1:ID,~.sum(#2):Total)",A1:B12)
```

ID	Total
51	161.74
62	142.11
75	162.74
81	186.64
92	577.08

11.9 Group when meeting blank row

Here below is a data table:

	A	B	C	D
1	ID	Value		
2	1	3204		
3	2	5644		
4	3	5026		
5	4	4675		
6				
7	5	7111		
8	6	2380		
9	7	9132		
10				
11	10	4312		
12	11	4049		
13	12	6967		
14	13	4190		
15				
16	15	6602		
17	16	5074		
18	17	8110		
19				
20	19	8187		
21				

The data consist of multiple segments, and each segment includes N consecutive rows and 1 blank row. Now we want to calculate the sum of the values of each segment, and fill in the result in the cell of the blank row, column C, and keep the rest cells in column C null. To achieve this, enter the formula in cell C1:

```
=spl("=?1.conj().group@i(~[-1]==null).([null]*(~.len()-1)|~.sum()).conj().new(~:Total)",B2:B21)
```

	A	B	C
1	ID	Value	Total
2	1	3204	
3	2	5644	
4	3	5026	
5	4	4675	
6			18549
7	5	7111	
8	6	2380	
9	7	9132	
10			18623
11	10	4312	
12	11	4049	
13	12	6967	
14	13	4190	
15			19518
16	15	6602	
17	16	5074	
18	17	8110	
19			19786
20	19	8187	
21			8187

When the previous member is null, create a new group.

Loop through every group, calculate the sum of the values of each group, and make up null values in the cells above the sum cell of each group (the number of nulls should be the number of members minus 1).

11.10 Group when meeting non-null value

We have an annual and monthly water-consumption data table, and the year is filled in only on the left of the first month of each year. Part of the data is shown as below:

	A	B	C	D
1	Year	Month	Water	
2	2016	6	9714	
3		7	7025	
4		8	8216	
5		9	9484	
6		10	7113	
7		11	8337	
8		12	6869	
9	2017	1	6608	
10		2	9550	
11		3	5538	
12		4	7961	
13		5	9137	
14		6	5058	
15		7	10284	
16		8	7432	
17		9	5473	
18		10	9716	
19		11	10919	
20		12	7894	

The task is to calculate the total water consumption of each year. To achieve this, enter the formula in cell D1:

```
=spl("=E(?1).derive(:TotalWater).group@i(Year!=null).run(~(1).TotalWater=~.sum(Water)).conj(
).new(TotalWater)",A1:C20)
```

	A	B	C	D
1	Year	Month	Water	TotalWater
2	2016	6	9714	56758
3		7	7025	
4		8	8216	
5		9	9484	
6		10	7113	
7		11	8337	
8		12	6869	
9	2017	1	6608	95570
10		2	9550	
11		3	5538	
12		4	7961	
13		5	9137	
14		6	5058	
15		7	10284	
16		8	7432	
17		9	5473	
18		10	9716	
19		11	10919	
20		12	7894	
21				

When meeting a non-null value in the column Year, create a new group. Option @i means creating a new group when the condition is satisfied. Assign the sum(Water) of the current group to the first row of each group, and create a new data set and take out the TotalWater.

11.11 Group by interval of data values

Here below is a data table:

	A	B	C
1	Date	Value	
2	2000/1/1	61.47	
3	2000/1/1	19.57	
4	2000/1/1	44.53	
5	2000/1/4	65.98	
6	2000/1/5	142.11	
7	2000/1/6	25.87	
8	2000/1/6	142.11	
9	2000/1/8	44.53	
10	2000/1/9	873.2	
11	2000/1/10	52.89	
12	2000/1/11	82.4	
13	2000/1/11	118.37	
14	2000/1/11	354.8	
15	2000/1/14	90.6	
16	2000/1/15	6.49	
17	2000/1/16	44.53	
18	2000/1/17	33.76	
19	2000/1/18	162.74	
20			

The task is to put every 5 days in a group, and calculate the average of the values in each group:

	A
1	=E('A1:B19')
2	=A1.Date
3	=A1.group@i(if(Date-A2==5,(A2=Date,true),false))
4	=A3.new(Date,~.avg(Value):Avg)

Date	Avg
2000/1/1	66.73
2000/1/6	227.72
2000/1/11	130.53
2000/1/16	80.34

A2: Let the start variable of loop be the Date of the first row.

A3: Put every 5 days in a group.

A4: Calculate the average of the values in each group, and the Date means the first date of each group.

11.12 Concatenate data within group into text

Here below is a data table:

	A	B
1	Dept	Name
2	R&D	Tom
3	Finance	John
4	Sales	Joan
5	HR	Rocky
6	R&D	Ham
7	Finance	Kate
8	Sales	Rose
9	HR	Nomy
10	R&D	Neil
11	Finance	Jack
12	Sales	Joe

The task is to group by Dept, and concatenate the names within the same group into the same row:

```
=spl("=E(?1).group(Dept;~.(Name).concat@c():Name)",A1:B12)
```

Dept	Name
Finance	John,Kate,Jack
HR	Rocky,Nomy
R&D	Tom,Ham,Neil
Sales	Joan,Rose,Joe

11.13 Auto-aggregating in the case of multiple columns - unfixed number

Here below is a data table:

	A	B	C	D	E	F
1	1	a	72	39	70	73
2	2	b	79	69	49	73
3	3	c	59	100	15	49
4		sum:				
5	4	d	15	77	4	59
6	5	e	96	90	71	84
7		sum:				
8	6	f	22	51	93	93
9	7	g	23	7	27	31
10	8	h	72	8	95	37
11	9	i	3	3	95	33
12		sum:				
13	10	j	5	91	40	70
14	11	k	42	64	98	63
15		sum:				

In this table, the number of value columns is unfixed, and may vary from time to time. Currently, there are four columns C~F, and we want to calculate the aggregation value of each column, and fill in the result in the blank cell below each group of data:

	A	B
1	='A1:F15'.group@i(~[-1](1)==null)	
2	for A1	=A2.select(~(1))
3		=B2|[A2.m(-1)(to(2))|transpose(B2.(~.to(3,))).(~.sum())]
4		>B1|=B3
5	return B1	

1	a	72	39	70	73
2	b	79	69	49	73
3	c	59	100	15	49
	sum:	210	208	134	195
4	d	15	77	4	59
5	e	96	90	71	84
	sum:	111	167	75	143
6	f	22	51	93	93
7	g	23	7	27	31
8	h	72	8	95	37
9	i	3	3	95	33
	sum:	120	69	310	194
10	j	5	91	40	70
11	k	42	64	98	63
	sum:	47	155	138	133

A1: Group whenever the previous cell in the first column is null.

A2: Loop through every group.

B2: Filter the aggregation row.

B3: Append aggregation result to B2.

B4: Append the result of each group to B1.

Chapter 12 **Association and comparison**

12.1 **Use formulas to handle association**

Here below is an Excel file *book1.xlsx*, which stores employees' meal data. Part of the data is as follows:

	A	B	C
1	**Name**	**Meal type**	**Meal charge**
2	Tom	breakfast	
3	John	lunch	
4	Joan	supper	
5	Rocky	breakfast	
6	Ham	lunch	
7	Kate	supper	
8	Rose	breakfast	
9	Nomy	lunch	
10	Neil	supper	
11	Jack	breakfast	
12	Joe	lunch	
13	Peter	supper	
14	Sunny	breakfast	
15	Tiger	lunch	
16	Alice	breakfast	
17	Cindy	lunch	
18	Leon	supper	
19	Lily	breakfast	
20	Kevin	lunch	
21	Shelly	supper	
22	Panzy	breakfast	
23	Maggie	lunch	
24	Mark	supper	
25	Aileen	breakfast	
26			

The task is to calculate the values in column C according to the following pricing rules:

	A	B
1	**Meal type**	**Meal charge**
2	breakfast	10
3	lunch	15
4	supper	20

Since there are only a few rows in this table, we can enumerate them in the formula directly. Enter in cell C2:

```
=spl("=case(?1,""breakfast"":10,""lunch"":15,""supper"":20)",B2)
```

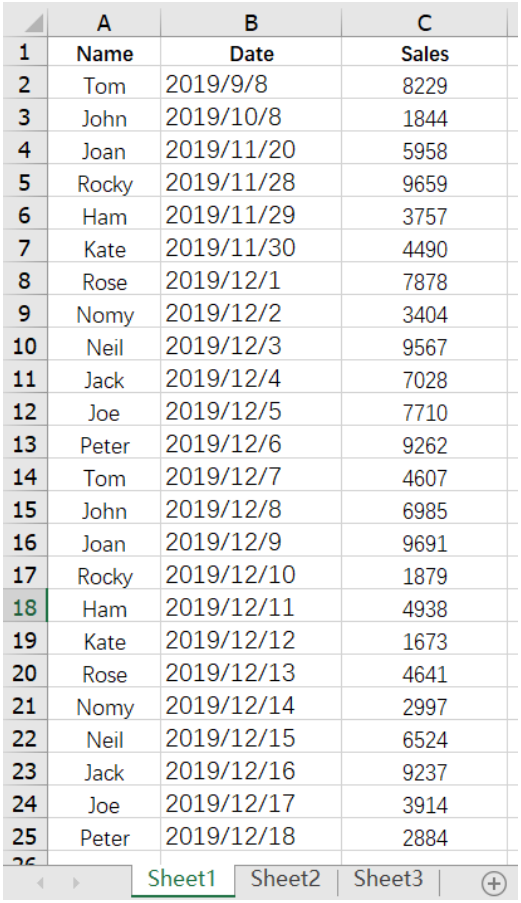
	A	B	C
1	Name	Meal type	Meal charge
2	Tom	breakfast	10
3	John	lunch	
4	Joan	supper	
5	Rocky	breakfast	
6	Ham	lunch	
7	Kate	supper	
8	Rose	breakfast	
9	Nomy	lunch	
10	Neil	supper	
11	Jack	breakfast	
12	Joe	lunch	
13	Peter	supper	
14	Sunny	breakfast	
15	Tiger	lunch	
16	Alice	breakfast	
17	Cindy	lunch	
18	Leon	supper	
19	Lily	breakfast	
20	Kevin	lunch	
21	Shelly	supper	
22	Panzy	breakfast	
23	Maggie	lunch	
24	Mark	supper	
25	Aileen	breakfast	

Then drag C2 down to every relevant row:

	A	B	C
1	Name	Meal type	Meal charge
2	Tom	breakfast	10
3	John	lunch	15
4	Joan	supper	20
5	Rocky	breakfast	10
6	Ham	lunch	15
7	Kate	supper	20
8	Rose	breakfast	10
9	Nomy	lunch	15
10	Neil	supper	20
11	Jack	breakfast	10
12	Joe	lunch	15
13	Peter	supper	20
14	Sunny	breakfast	10
15	Tiger	lunch	15
16	Alice	breakfast	10
17	Cindy	lunch	15
18	Leon	supper	20
19	Lily	breakfast	10
20	Kevin	lunch	15
21	Shelly	supper	20
22	Panzy	breakfast	10
23	Maggie	lunch	15
24	Mark	supper	20
25	Aileen	breakfast	10
26			

12.2 **Single column association**

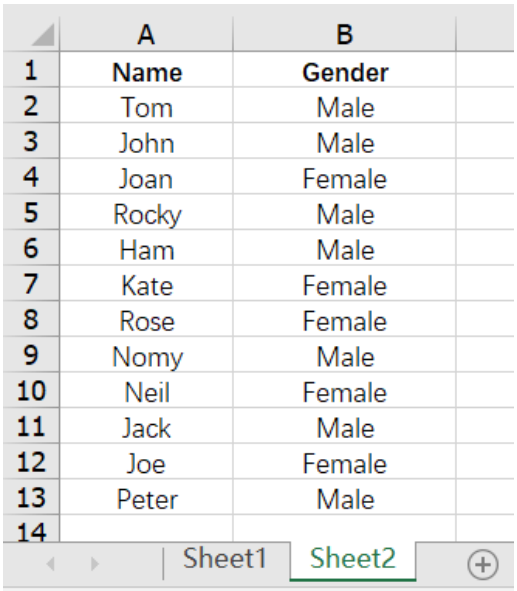
Here below is a data table that stores the sales details:



	A	B	C
1	Name	Date	Sales
2	Tom	2019/9/8	8229
3	John	2019/10/8	1844
4	Joan	2019/11/20	5958
5	Rocky	2019/11/28	9659
6	Ham	2019/11/29	3757
7	Kate	2019/11/30	4490
8	Rose	2019/12/1	7878
9	Nomy	2019/12/2	3404
10	Neil	2019/12/3	9567
11	Jack	2019/12/4	7028
12	Joe	2019/12/5	7710
13	Peter	2019/12/6	9262
14	Tom	2019/12/7	4607
15	John	2019/12/8	6985
16	Joan	2019/12/9	9691
17	Rocky	2019/12/10	1879
18	Ham	2019/12/11	4938
19	Kate	2019/12/12	1673
20	Rose	2019/12/13	4641
21	Nomy	2019/12/14	2997
22	Neil	2019/12/15	6524
23	Jack	2019/12/16	9237
24	Joe	2019/12/17	3914
25	Peter	2019/12/18	2884

Sheet1 Sheet2 Sheet3

The following table stores the salespersons information:

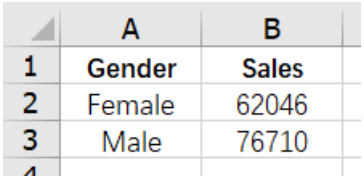


	A	B
1	Name	Gender
2	Tom	Male
3	John	Male
4	Joan	Female
5	Rocky	Male
6	Ham	Male
7	Kate	Female
8	Rose	Female
9	Nomy	Male
10	Neil	Female
11	Jack	Male
12	Joe	Female
13	Peter	Male
14		

Sheet1 Sheet2

Now we want to calculate the total sales of male and female salespersons respectively:

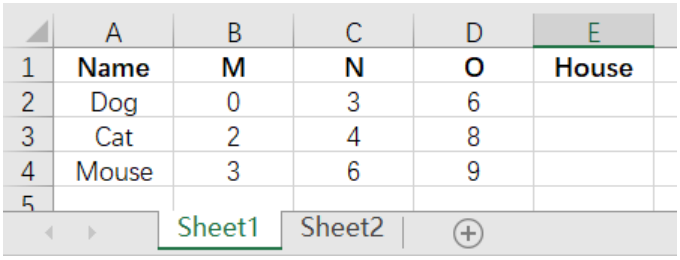
	A
1	=E('Sheet1!A1:C25')
2	=E('Sheet2!A1:B13')
3	=A1.join(Name,A2:Name,Gender)
4	=A3.groups(Gender;sum(Sales):Sales)



	A	B
1	Gender	Sales
2	Female	62046
3	Male	76710

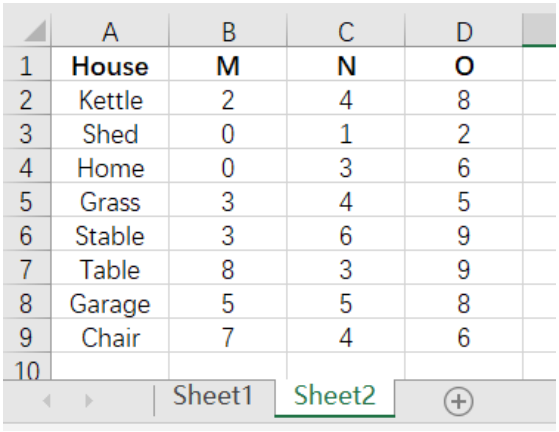
12.3 **Multiple columns association**

Here below is a data table:



	A	B	C	D	E
1	**Name**	**M**	**N**	**O**	**House**
2	Dog	0	3	6	
3	Cat	2	4	8	
4	Mouse	3	6	9	
5					

Sheet1 Sheet2

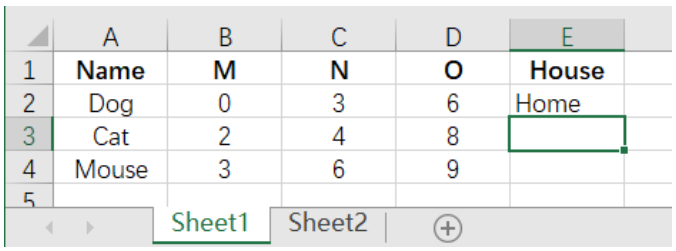


	A	B	C	D
1	**House**	**M**	**N**	**O**
2	Kettle	2	4	8
3	Shed	0	1	2
4	Home	0	3	6
5	Grass	3	4	5
6	Stable	3	6	9
7	Table	8	3	9
8	Garage	5	5	8
9	Chair	7	4	6
10				

Sheet1 Sheet2

The task is to compare the values of M, N and O of each row in Sheet1 with the corresponding values in Sheet2. When the row that has the same values is found, return the House value of this row, and fill in this value in column E of Sheet1. To achieve this, enter the following formula in cell E2:

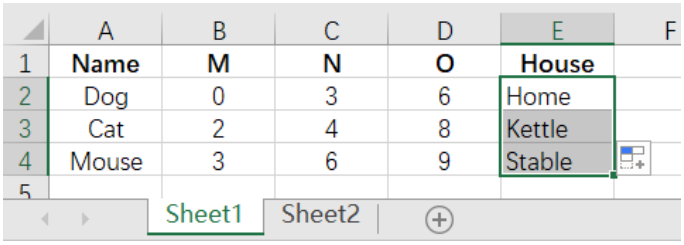
```
=spl("=E(?1).keys(M,N,O).find([?2,?3,?4]).House",Sheet2!A$1:D$9,B2,C2,D2)
```



	A	B	C	D	E
1	**Name**	**M**	**N**	**O**	**House**
2	Dog	0	3	6	Home
3	Cat	2	4	8	
4	Mouse	3	6	9	
5					

Sheet1 Sheet2

Then drag E2 down to every relevant row:



	A	B	C	D	E	F
1	**Name**	**M**	**N**	**O**	**House**	
2	Dog	0	3	6	Home	
3	Cat	2	4	8	Kettle	
4	Mouse	3	6	9	Stable	
5						

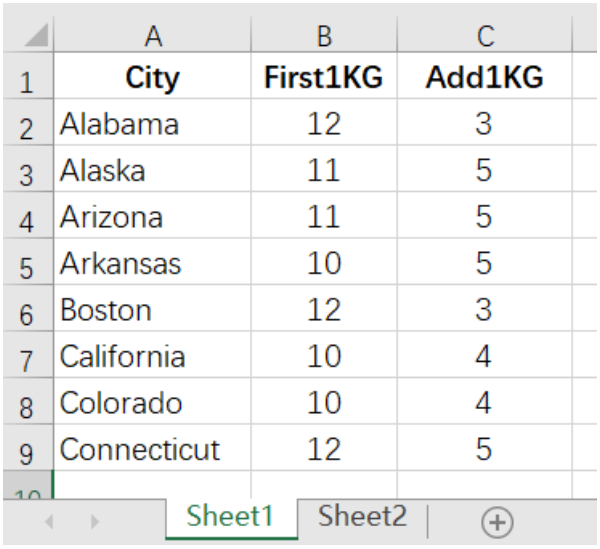
Sheet1 Sheet2

Convert the data of Sheet2 to a table sequence, and specify M, N and O as key column;

Loop through each row of Sheet2, use the values of M, N and O to correspond the values of key columns of Sheet2 to search for the row with the same values. When such row is found, return the House value of the row.

12.4 **Reference multi-column data from association table**

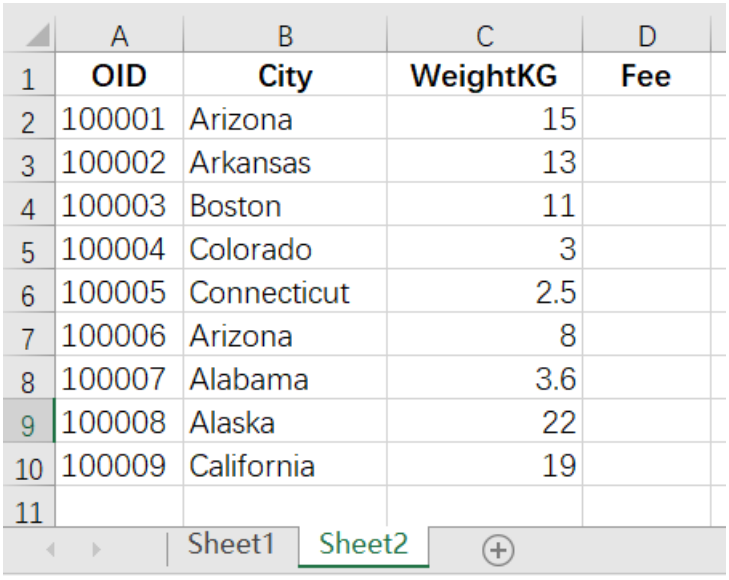
Here below is a table that lists the freight charging standards:



	A	B	C
1	City	First1KG	Add1KG
2	Alabama	12	3
3	Alaska	11	5
4	Arizona	11	5
5	Arkansas	10	5
6	Boston	12	3
7	California	10	4
8	Colorado	10	4
9	Connecticut	12	5

Sheet1 Sheet2

The following table stores the shipping order information:



	A	B	C	D
1	OID	City	WeightKG	Fee
2	100001	Arizona	15	
3	100002	Arkansas	13	
4	100003	Boston	11	
5	100004	Colorado	3	
6	100005	Connecticut	2.5	
7	100006	Arizona	8	
8	100007	Alabama	3.6	
9	100008	Alaska	22	
10	100009	California	19	
11				

Sheet1 Sheet2

Now we want to calculate the actual freight based on the first table. To achieve this, write in cell D2:

	A
1	=spl("=E(?1).select@1(City==?2)
2	=A1.First1KG+A1.Add1KG*(ceil(?3)-1)",Sheet1!A$1:C$9,B2,C2)

	A	B	C	D
1	OID	City	WeightKG	Fee
2	100001	Arizona	15	81
3	100002	Arkansas	13	
4	100003	Boston	11	
5	100004	Colorado	3	
6	100005	Connecticut	2.5	
7	100006	Arizona	8	
8	100007	Alabama	3.6	
9	100008	Alaska	22	
10	100009	California	19	
11				

Sheet1 Sheet2

Drag D2 down to every relevant row:

	A	B	C	D
1	OID	City	WeightKG	Fee
2	100001	Arizona	15	81
3	100002	Arkansas	13	70
4	100003	Boston	11	42
5	100004	Colorado	3	18
6	100005	Connecticut	2.5	22
7	100006	Arizona	8	46
8	100007	Alabama	3.6	21
9	100008	Alaska	22	116
10	100009	California	19	82
11				

Sheet1 Sheet2

12.5 Use formulas to handle interval association

Here below is a data table:

	A	B
1	Quantity	Price
2	221	
3	87	
4	33	
5	73	
6	162	
7	227	
8	403	
9	288	
10	78	
11	213	
12	374	
13	152	
14		

We want to calculate the price value in column B according to the quantity in column A, and the calculation should follow the rule: different quantity intervals correspond to different prices, as shown in the table below:

	A	B
1	Quantity	Price
2	30-50	15
3	50-100	13.75
4	100-300	13
5	300-500	12.5

Enter in cell B2:

```
=spl("=[15,13.75,13,12.5]([30,50,100,300,500].pseg@r(?1))",A2)
```

	A	B
1	Quantity	Price
2	221	13
3	87	
4	33	
5	73	
6	162	
7	227	
8	403	
9	288	
10	78	
11	213	
12	374	
13	152	
14		

Then drag B2 down to every relevant row:

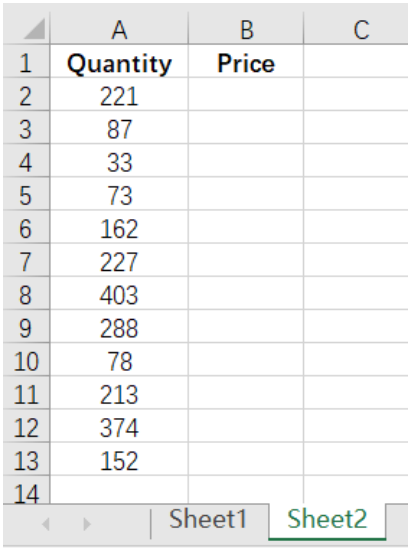
	A	B
1	Quantity	Price
2	221	13
3	87	13.75
4	33	15
5	73	13.75
6	162	13
7	227	13
8	403	12.5
9	288	13
10	78	13.75
11	213	13
12	374	12.5
13	152	13
14		

The idea of the calculation is to use the pseg function to calculate which interval the quantity value belongs to [30, 50, 100, 300, 500], and then take out the price of the corresponding interval from the price sequence [15, 13.75, 13, 12.5] and return.

12.6 Use association table to handle interval association

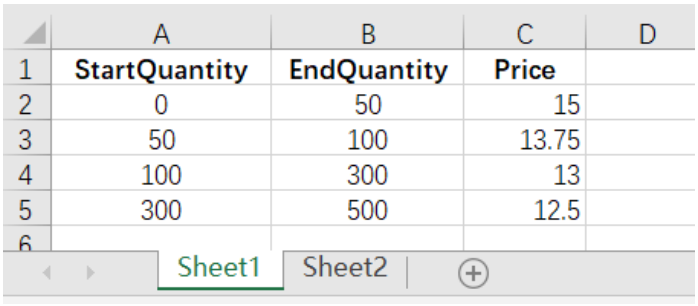
Example 1:

Here below is a data table:



	A	B	C
1	Quantity	Price	
2	221		
3	87		
4	33		
5	73		
6	162		
7	227		
8	403		
9	288		
10	78		
11	213		
12	374		
13	152		
14			

Sheet1 Sheet2



	A	B	C	D
1	StartQuantity	EndQuantity	Price	
2	0	50	15	
3	50	100	13.75	
4	100	300	13	
5	300	500	12.5	
6				

Sheet1 Sheet2

The task is to calculate the values in column B of Sheet2 according to the rule: search Sheet1 with the quantity value of Sheet2, if the value is greater than StartQuantity and less than or equal to EndQuantity of a certain row, return the price of this row. To achieve this task, enter in cell B2:

```
=spl("=E(?1).segp@r(StartQuantity,?2).Price",Sheet1!A$1:C$5,A2)
```

	A	B	C
1	Quantity	Price	
2	221	13	
3	87		
4	33		
5	73		
6	162		
7	227		
8	403		
9	288		
10	78		
11	213		
12	374		
13	152		
14			

Sheet1 Sheet2

Then drag B2 down to every relevant row:

	A	B	C
1	Quantity	Price	
2	221	13	
3	87	13.75	
4	33	15	
5	73	13.75	
6	162	13	
7	227	13	
8	403	12.5	
9	288	13	
10	78	13.75	
11	213	13	
12	374	12.5	
13	152	13	
14			

Sheet1 Sheet2

The idea of the calculation is to use the segp function to query which segment number of interval formed by StartQuantity of Sheet1 the quantity value is in, and then take the price of the row corresponding to the segment number and return. The option @r means forming a left-open and right-closed interval. For example, the number 50 should be counted in the interval where the first row is located.

Example 2:

Here below is a car charging data table:

	A	B	C	D	E
1	CardId	Starttime	Endtime	Quantity	Price
2	0000012541	2018/4/30 17:11:28	2018/4/30 18:20:17	47.32	
3	0000012541	2018/4/30 15:04:54	2018/4/30 16:07:07	42.7	
4	0000012541	2018/4/30 14:49:22	2018/4/30 14:49:45	0	
5	0000012541	2018/4/30 13:56:16	2018/4/30 14:46:00	35.88	
6	0000012541	2018/4/24 18:00:19	2018/4/24 18:00:38	0	
7	0000012541	2018/4/24 11:31:54	2018/4/24 11:33:05	0.13	
8	0000012541	2018/4/24 11:31:00	2018/4/24 11:31:54	0	
9	0000012541	2018/4/24 11:04:45	2018/4/24 11:05:45	0.13	
10	0000012541	2018/4/21 17:34:06	2018/4/21 17:35:00	0.13	
11	0000012541	2018/4/21 17:32:32	2018/4/21 17:33:37	0.17	
12	0000012541	2018/4/21 17:27:25	2018/4/21 17:28:38	0.35	
13	0000012882	2018/5/31 19:20:05	2018/5/31 20:22:39	45.92	
14	0000012881	2018/5/28 6:46:01	2018/5/28 7:19:06	24.17	
15	0000012881	2018/5/24 6:52:40	2018/5/24 7:23:08	22.22	
16	0000012881	2018/5/24 6:50:33	2018/5/24 6:51:32	0	
17	0000012881	2018/5/19 15:08:47	2018/5/19 15:44:21	25.74	
18	0000012881	2018/5/19 15:07:14	2018/5/19 15:08:04	0.13	
19	0000012882	2018/5/18 14:36:07	2018/5/18 14:36:22	0	
20	0000012882	2018/5/18 14:23:44	2018/5/18 14:23:57	0	
21	0000012882	2018/5/18 14:22:29	2018/5/18 14:23:44	0	
22	0000012882	2018/5/18 8:51:31	2018/5/18 9:32:28	29.6	
23	0000012882	2018/5/18 8:47:20	2018/5/18 8:48:38	0.44	
24	0000012882	2018/5/18 8:46:01	2018/5/18 8:47:20	0	
25	0000012882	2018/5/18 8:40:14	2018/5/18 8:43:48	2.05	
26	0000012882	2018/5/18 8:39:23	2018/5/18 8:40:14	0	
27	0000012882	2018/5/18 8:36:52	2018/5/18 8:37:26	0	
28	0000012882	2018/5/18 8:23:55	2018/5/18 8:26:14	0.8	
29	0000014529	2018/5/15 15:45:25	2018/5/15 15:47:09	0.58	
30	0000014529	2018/5/15 15:44:04	2018/5/15 15:44:42	0	

Sheet1 | Sheet2 | Sheet3 | Sheet4

The following table lists the electricity price data at different charging time intervals:

	A	B	C	D	E	F
1	StartHour	6	8	12	16	22
2	EndHour	8	12	16	22	6
3	Price	0.9094	1.0354	0.6574	0.9094	0.4054
4						

Sheet1 | Sheet2 | Sheet3 | Sheet4

The task is to calculate the values in Price column of Sheet3 according to the rule: search Sheet4 for the time interval where the hour number of Starttime is located, and take the price. To achieve this, enter in cell E2:

```
=spl("=a=E@b(?1),a(3).array().to(2,)(a(1).array().to(2,).pseg(?2))", Sheet4!A$1:F$3, HOUR(B2))
```

	A	B	C	D	E
1	CardId	Starttime	Endtime	Quantity	Price
2	0000012541	2018/4/30 17:11:28	2018/4/30 18:20:17	47.32	0.9094
3	0000012541	2018/4/30 15:04:54	2018/4/30 16:07:07	42.7	
4	0000012541	2018/4/30 14:49:22	2018/4/30 14:49:45	0	
5	0000012541	2018/4/30 13:56:16	2018/4/30 14:46:00	35.88	
6	0000012541	2018/4/24 18:00:19	2018/4/24 18:00:38	0	
7	0000012541	2018/4/24 11:31:54	2018/4/24 11:33:05	0.13	
8	0000012541	2018/4/24 11:31:00	2018/4/24 11:31:54	0	
9	0000012541	2018/4/24 11:04:45	2018/4/24 11:05:45	0.13	
10	0000012541	2018/4/21 17:34:06	2018/4/21 17:35:00	0.13	
11	0000012541	2018/4/21 17:32:32	2018/4/21 17:33:37	0.17	
12	0000012541	2018/4/21 17:27:25	2018/4/21 17:28:38	0.35	
13	0000012882	2018/5/31 19:20:05	2018/5/31 20:22:39	45.92	
14	0000012881	2018/5/28 6:46:01	2018/5/28 7:19:06	24.17	
15	0000012881	2018/5/24 6:52:40	2018/5/24 7:23:08	22.22	
16	0000012881	2018/5/24 6:50:33	2018/5/24 6:51:32	0	
17	0000012881	2018/5/19 15:08:47	2018/5/19 15:44:21	25.74	
18	0000012881	2018/5/19 15:07:14	2018/5/19 15:08:04	0.13	
19	0000012882	2018/5/18 14:36:07	2018/5/18 14:36:22	0	
20	0000012882	2018/5/18 14:23:44	2018/5/18 14:23:57	0	
21	0000012882	2018/5/18 14:22:29	2018/5/18 14:23:44	0	
22	0000012882	2018/5/18 8:51:31	2018/5/18 9:32:28	29.6	
23	0000012882	2018/5/18 8:47:20	2018/5/18 8:48:38	0.44	
24	0000012882	2018/5/18 8:46:01	2018/5/18 8:47:20	0	
25	0000012882	2018/5/18 8:40:14	2018/5/18 8:43:48	2.05	
26	0000012882	2018/5/18 8:39:23	2018/5/18 8:40:14	0	
27	0000012882	2018/5/18 8:36:52	2018/5/18 8:37:26	0	
28	0000012882	2018/5/18 8:23:55	2018/5/18 8:26:14	0.8	
29	0000014529	2018/5/15 15:45:25	2018/5/15 15:47:09	0.58	
30	0000014529	2018/5/15 15:44:04	2018/5/15 15:44:42	0	

Sheet1 Sheet2 Sheet3 Sheet4

Drag E2 down to every relevant row:

	A	B	C	D	E
1	CardId	Starttime	Endtime	Quantity	Price
2	0000012541	2018/4/30 17:11:28	2018/4/30 18:20:17	47.32	0.9094
3	0000012541	2018/4/30 15:04:54	2018/4/30 16:07:07	42.7	0.6574
4	0000012541	2018/4/30 14:49:22	2018/4/30 14:49:45	0	0.6574
5	0000012541	2018/4/30 13:56:16	2018/4/30 14:46:00	35.88	0.6574
6	0000012541	2018/4/24 18:00:19	2018/4/24 18:00:38	0	0.9094
7	0000012541	2018/4/24 11:31:54	2018/4/24 11:33:05	0.13	1.0354
8	0000012541	2018/4/24 11:31:00	2018/4/24 11:31:54	0	1.0354
9	0000012541	2018/4/24 11:04:45	2018/4/24 11:05:45	0.13	1.0354
10	0000012541	2018/4/21 17:34:06	2018/4/21 17:35:00	0.13	0.9094
11	0000012541	2018/4/21 17:32:32	2018/4/21 17:33:37	0.17	0.9094
12	0000012541	2018/4/21 17:27:25	2018/4/21 17:28:38	0.35	0.9094
13	0000012882	2018/5/31 19:20:05	2018/5/31 20:22:39	45.92	0.9094
14	0000012881	2018/5/28 6:46:01	2018/5/28 7:19:06	24.17	0.9094
15	0000012881	2018/5/24 6:52:40	2018/5/24 7:23:08	22.22	0.9094
16	0000012881	2018/5/24 6:50:33	2018/5/24 6:51:32	0	0.9094
17	0000012881	2018/5/19 15:08:47	2018/5/19 15:44:21	25.74	0.6574
18	0000012881	2018/5/19 15:07:14	2018/5/19 15:08:04	0.13	0.6574
19	0000012882	2018/5/18 14:36:07	2018/5/18 14:36:22	0	0.6574
20	0000012882	2018/5/18 14:23:44	2018/5/18 14:23:57	0	0.6574
21	0000012882	2018/5/18 14:22:29	2018/5/18 14:23:44	0	0.6574
22	0000012882	2018/5/18 8:51:31	2018/5/18 9:32:28	29.6	1.0354
23	0000012882	2018/5/18 8:47:20	2018/5/18 8:48:38	0.44	1.0354
24	0000012882	2018/5/18 8:46:01	2018/5/18 8:47:20	0	1.0354
25	0000012882	2018/5/18 8:40:14	2018/5/18 8:43:48	2.05	1.0354
26	0000012882	2018/5/18 8:39:23	2018/5/18 8:40:14	0	1.0354
27	0000012882	2018/5/18 8:36:52	2018/5/18 8:37:26	0	1.0354
28	0000012882	2018/5/18 8:23:55	2018/5/18 8:26:14	0.8	1.0354
29	0000014529	2018/5/15 15:45:25	2018/5/15 15:47:09	0.58	0.6574
30	0000014529	2018/5/15 15:44:04	2018/5/15 15:44:42	0	0.6574

Sheet1 | Sheet2 | Sheet3 | Sheet4

The idea of calculation is to use the StartHour sequence starting from the 1st row and the 2nd column of Sheet4 to form the time intervals, and search for the interval where the hour number of Starttime of Sheet3 is located, and take the price in the corresponding 3rd row of Sheet4 and return.

12.7 Use a two-dimensional association table

We have a scoring data table for children's height. In this table, the cells A1, B1 and C1 are different age ranges, and below them are the height data (cm); column D lists the scoring data.

	A	B	C	D
1	6-7	8-9	10-11	Score
2	130	140	155	100
3	125	132	145	90
4	120	128	140	80
5	115	120	132	70
6				

Sheet1 Sheet2

The following table stores the children's information:

	A	B	C	D
1	Name	Age	Height	Score
2	Tom	9	135	
3	Jack	6	122	
4	Joan	10	136	
5	Kate	8	142	
6				

Sheet1 Sheet2

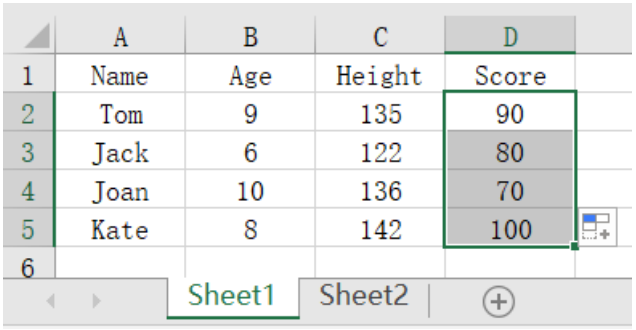
Now we want to find out the scoring data in Sheet2 according to Age and Height in Sheet1, and fill in the results in column D. To achieve this, enter in cell D2:

	A
1	=E('Sheet2!A$1:D$5').rvs()
2	=(A1.fno()-1).(int(A1.fname(~).split("-")(1))).pseg('B2')
3	=A1.segp(~.field(A2),'C2').Score

	A	B	C	D
1	Name	Age	Height	Score
2	Tom	9	135	90
3	Jack	6	122	
4	Joan	10	136	
5	Kate	8	142	
6				

Sheet1 Sheet2

Then drag D2 down to every relevant row:



	A	B	C	D
1	Name	Age	Height	Score
2	Tom	9	135	90
3	Jack	6	122	80
4	Joan	10	136	70
5	Kate	8	142	100
6				

Sheet1 Sheet2

A1: Convert the data of Sheet2 to a table sequence, and use the rvs function to reverse the order to make the heigh data arranged in ascending order

A2: Take out the column names of A1, and remove the last column name; Split each column name with a minus sign, and then convert the first one to an integer to form a sequence, i.e., [6,8,10]. In this sequence, find out the segment number where the current Age is located.

A3: Use the Height value to search the sequence interval formed by the values of A2-th column of A1 to find its corresponding row, and take the score of this row and return

12.8 Use interval range to perform retroactive searching of association table

Here below is a data table:

	A	B	C	D
1	ID	Start	End	Value
2	1001	1	54	
3	1001	55	100	
4	1001	101	149	
5	1001	150	200	
6	1002	1	10	
7	1002	11	24	
8	1003	1	2	
9	1003	3	6	
10	1003	7	8	
11	1003	9	21	
12	1003	22	30	

Sheet1 Sheet2

	A	B	C
1	ID	Num	Value
2	1001	5	2
3	1001	77	5
4	1002	9	1
5	1003	11	4

Sheet1 Sheet2

The task is to calculate the values in column D of Sheet1 according to the following requirements:

1. The ID column of Sheet2 is the same as that of Sheet1.

2. When the first criterion is met, judge whether the Num of Sheet2 falls into the interval between start and end of Sheet1.

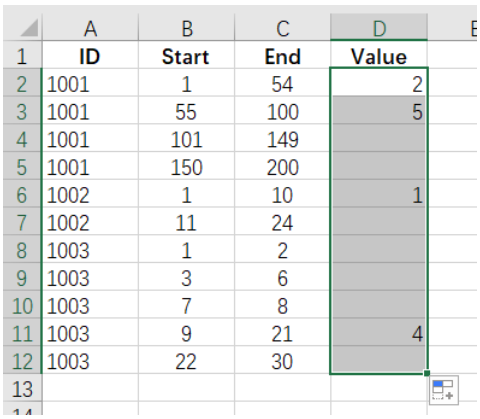
3. When the above two criteria are met at the same time, the value in Sheet1 is the corresponding value in Sheet2.

Enter in cell D2:

```
=spl("=E(?1).select@1(ID==?2 && Num>?3 && Num
<=?4).Value",Sheet2!A$1:C$5,A2,B2,C2)
```

	A	B	C	D
1	ID	Start	End	Value
2	1001	1	54	2
3	1001	55	100	
4	1001	101	149	
5	1001	150	200	
6	1002	1	10	
7	1002	11	24	
8	1003	1	2	
9	1003	3	6	
10	1003	7	8	
11	1003	9	21	
12	1003	22	30	

Then drag D2 down to every relevant row:



	A	B	C	D
1	**ID**	**Start**	**End**	**Value**
2	1001	1	54	2
3	1001	55	100	5
4	1001	101	149	
5	1001	150	200	
6	1002	1	10	1
7	1002	11	24	
8	1003	1	2	
9	1003	3	6	
10	1003	7	8	
11	1003	9	21	4
12	1003	22	30	
13				

12.9 Associate multiple rows of data

Here below is a data table:

	A	B	C
1	aid	bid	cid
2	XD19010576	SCD181202515	XO18120141
3	XD19010577	SCD181202515	XO18120141
4	XD19010578	SCD181202514	XO18120142
5	XD19010579	SCD181202612	XO18120253
6	XD19010580	SCD181202619	XO18120254
7	XD19010581	SCD181202614	XO18120255
8	XD19020009	SCD181202619	XO18120254
9	XD19020010	SCD181202614	XO18120255
10	XD19020011	SCD181202514	XO18120142
11	XD19020012	SCD181202612	XO18120253
12	XD19020013	SCD181202514	XO18120142
13	XD19020014	SCD181202612	XO18120253
14	XD19020015	SCD181202619	XO18120254
15	XD19020016	SCD181202614	XO18120255

Sheet1 Sheet2

	A	B	C	D	E	F
1	bid	cid	aids			
2	SCD181202515	XO18120141				
3	SCD181202514	XO18120142				
4	SCD181202612	XO18120253				
5	SCD181202619	XO18120254				
6	SCD181202614	XO18120255				

Sheet1 Sheet2

The task is to associate the bid and cid of Sheet1 with the corresponding bid and cid of Sheet2, and calculate the values of aid in Sheet2. Enter the formula in cell C2:

	A
1	=E('Sheet1!A$1:C$15').group(bid,cid;~.(aid):aid)
2	=E('A1:B6')
3	=A1.align(A2:[bid,cid],[bid,cid])
4	=A3.(aid)

	A	B	C	D	E
1	bid	cid	aids		
2	SCD181202515	XO18120141	XD19010576	XD19010577	
3	SCD181202514	XO18120142	XD19010578	XD19020011	XD19020013
4	SCD181202612	XO18120253	XD19010579	XD19020012	XD19020014
5	SCD181202619	XO18120254	XD19010580	XD19020009	XD19020015
6	SCD181202614	XO18120255	XD19010581	XD19020010	XD19020016

Sheet1 Sheet2

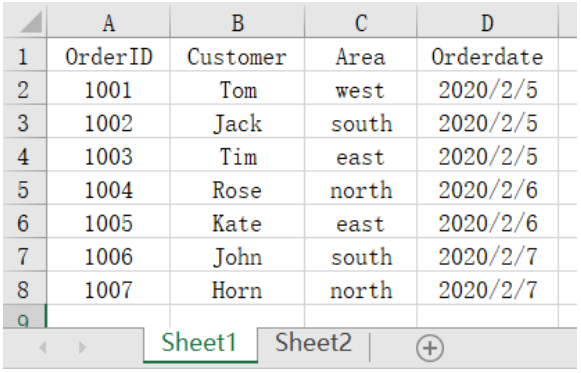
A1: Convert the data of Sheet1 to a table sequence, and then group by bid and cid, with multiple aids under each group

A3: Associate the bid and cid of Sheet1 with the corresponding bid and cid of Sheet2

A4: Return the aids in the result

12.10 Associate with detail table

Here below is a data table that stores the customer order information:



	A	B	C	D
1	OrderID	Customer	Area	Orderdate
2	1001	Tom	west	2020/2/5
3	1002	Jack	south	2020/2/5
4	1003	Tim	east	2020/2/5
5	1004	Rose	north	2020/2/6
6	1005	Kate	east	2020/2/6
7	1006	John	south	2020/2/7
8	1007	Horn	north	2020/2/7
9				

Sheet1 Sheet2

The following data table stores the order details:

	A	B	C	D	E
1	OrderID	Index	Product	Price	Quantity
2	1001	1	p1	12.5	32
3	1001	2	p2	14.3	55
4	1001	3	p3	15.7	14
5	1002	1	p1	12.5	10
6	1002	2	p3	15.7	8
7	1003	1	p1	12.5	36
8	1003	2	p4	8.8	15
9	1003	3	p6	62.3	85
10	1004	1	p2	14.3	24
11	1005	1	p3	15.7	12
12	1005	2	p4	8.8	27
13	1006	1	p1	12.5	18
14	1006	2	p4	8.8	92
15	1006	3	p6	62.3	10
16	1007	1	p3	15.7	6
17	1007	2	p5	24.8	8

Sheet1 Sheet2

Example 1:

Find out the customer order information with a total order amount greater than 1000:

	A
1	=E('Sheet1!A1:D8')
2	=E('Sheet2!A1:E17')
3	=A2.groups(OrderID;sum(Price*Quantity):Amount).select(Amount>1000)
4	=A1.join@i(OrderID,A3:OrderID)

A3: Group A2 by OrderID; calculate the total order Amount of each group; select the group with Amount>1000

A4: Join A1 and A3 on OrderID. The option @i means discarding the unmatched row in A1

	A	B	C	D
1	**OrderID**	**Customer**	**Area**	**Orderdate**
2	1001	Tom	west	2020-02-05
3	1003	Tim	east	2020-02-05
4	1006	John	south	2020-02-07

Example 2:

Find out the order details in the north area.

Script:

	A
1	=E('Sheet1!A1:D8')
2	=E('Sheet2!A1:E17')
3	=A1.select(Area=="north")
4	=A2.join@i(OrderID,A3:OrderID)

A3: Find out the order information in the north area from A1

A4: Join A2 and A3 on OrderID. The option @i means discarding the unmatched rows in A2

	A	B	C	D	E
1	**OrderID**	**Index**	**Product**	**Price**	**Quantity**
2	1004	1	p2	14.3	24
3	1007	1	p3	15.7	6
4	1007	2	p5	24.8	8
5					

12.11 **Find changes through comparison**

When comparing the contents of the key columns of two two-dimensional tables, we can first read the table data as a table sequence, and then take out the values of key column to form a set, and finally perform the operation on the two sets to obtain the result.

Here below is a data table, which stores the sales order data of 2018 and 2019 in Sheet1 and Sheet2 respectively, these two sheets have the same column structure:

	A	B	C	D	E
1	OrderID	CustomerID	ProductID	OrderDate	Amount
2	10248	VINET	11	2018-07-04	168
3	10248	VINET	42	2018-07-04	98
4	10248	VINET	72	2018-07-04	174
5	10249	TOMSP	14	2018-07-05	167.4
6	10249	TOMSP	51	2018-07-05	1696
7	10250	HANAR	41	2018-07-08	77
8	10250	HANAR	51	2018-07-08	1484
9	10250	HANAR	65	2018-07-08	252
10	10251	VICTE	22	2018-07-08	100.8
11	10251	VICTE	57	2018-07-08	234
12	10251	VICTE	65	2018-07-08	336
13	10252	SUPRD	20	2018-07-09	2592
14	10252	SUPRD	33	2018-07-09	50
15	10252	SUPRD	60	2018-07-09	1088
16	10253	HANAR	31	2018-07-10	200
17	10253	HANAR	39	2018-07-10	604.8
18	10253	HANAR	49	2018-07-10	640
19	10254	CHOPS	24	2018-07-11	54
20	10254	CHOPS	55	2018-07-11	403.2
21	10254	CHOPS	74	2018-07-11	168

Y2018 Y2019

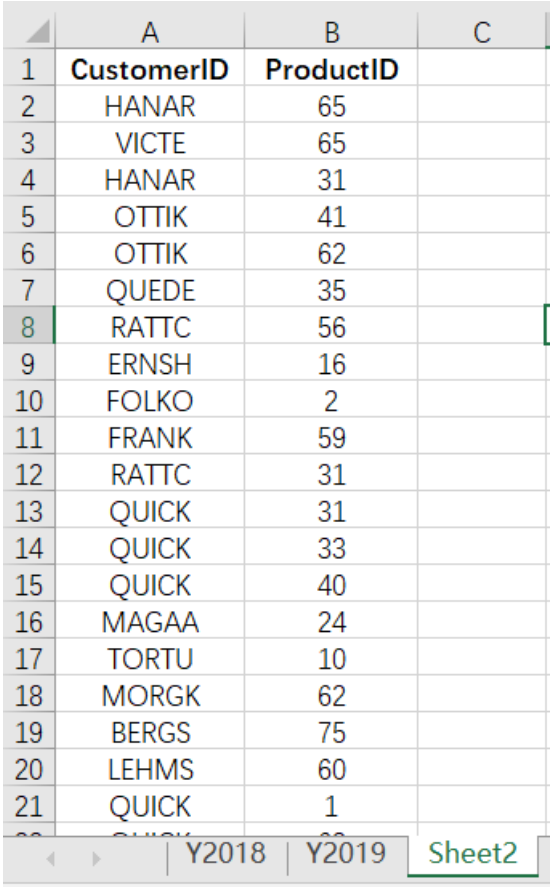
	A	B	C	D	E
1	OrderID	CustomerID	ProductID	OrderDate	Amount
2	10400	EASTC	29	2019-01-01	2079
3	10400	EASTC	35	2019-01-01	504
4	10400	EASTC	49	2019-01-01	480
5	10401	RATTC	30	2019-01-01	372.6
6	10401	RATTC	56	2019-01-01	2128
7	10401	RATTC	65	2019-01-01	336
8	10401	RATTC	71	2019-01-01	1032
9	10402	ERNSH	23	2019-01-02	432
10	10402	ERNSH	63	2019-01-02	2281.5
11	10403	ERNSH	16	2019-01-03	291.9
12	10403	ERNSH	48	2019-01-03	714
13	10404	MAGAA	26	2019-01-03	747
14	10404	MAGAA	42	2019-01-03	448
15	10404	MAGAA	49	2019-01-03	480
16	10405	LINOD	3	2019-01-06	400
17	10406	QUEEN	1	2019-01-07	144
18	10406	QUEEN	21	2019-01-07	240
19	10406	QUEEN	28	2019-01-07	1528.8
20	10406	QUEEN	36	2019-01-07	76
21	10406	QUEEN	40	2019-01-07	29.4

Y2018 Y2019

Example 1: Find the similarities

Find out the CustomerID and ProductID that the same product is purchased in both years:

```
=spl("=[E(?1),E(?2)].merge@io()",Y2018!B1:C406,Y2019!B1:C1060)
```



	A	B	C
1	CustomerID	ProductID	
2	HANAR	65	
3	VICTE	65	
4	HANAR	31	
5	OTTIK	41	
6	OTTIK	62	
7	QUEDE	35	
8	RATTC	56	
9	ERNSH	16	
10	FOLKO	2	
11	FRANK	59	
12	RATTC	31	
13	QUICK	31	
14	QUICK	33	
15	QUICK	40	
16	MAGAA	24	
17	TORTU	10	
18	MORGK	62	
19	BERGS	75	
20	LEHMS	60	
21	QUICK	1	

Y2018 | Y2019 | Sheet2

Merge the data of two years, @i means returning the common rows of the two years.

Example 2: Find the differences

Find out the order information of the new customers in 2019:

	A
1	=E('Y2018!A1:E406')
2	=E('Y2019!A1:E1060')
3	=A2.id(CustomerID)\A1.id(CustomerID)
4	=A2.select(A3.contain(CustomerID))

A3: Subtract the CustomerIDs of 2018 from all CustomerIDs of 2019 to get new CustomerIDs

A4: Filter out the orders of new CustomerIDs from 2019 order data table

	A	B	C	D	E
1	OrderID	CustomerID	ProductID	OrderDate	Amount
2	10405	LINOD	3	2019-01-06	400
3	10408	FOLIG	37	2019-01-08	208
4	10408	FOLIG	54	2019-01-08	35.4
5	10408	FOLIG	62	2019-01-08	1379
6	10409	OCEAN	14	2019-01-09	223.2
7	10409	OCEAN	21	2019-01-09	96
8	10422	FRANS	26	2019-01-22	49.8
9	10423	GOURL	31	2019-01-23	140
10	10423	GOURL	59	2019-01-23	880
11	10435	CONSH	2	2019-02-04	152
12	10435	CONSH	22	2019-02-04	201.6
13	10435	CONSH	72	2019-02-04	278
14	10448	RANCH	26	2019-02-17	149.4
15	10448	RANCH	40	2019-02-17	294
16	10462	CONSH	13	2019-03-03	4.8
17	10462	CONSH	23	2019-03-03	151.2
18	10480	FOLIG	47	2019-03-20	228
19	10480	FOLIG	59	2019-03-20	528
20	10482	LAZYK	40	2019-03-21	147
21	10485	LINOD	2	2019-03-25	304

Y2018 | Y2019 | Sheet2 | Sheet1

Example 3: Find all lost CustomerIDs in 2019:

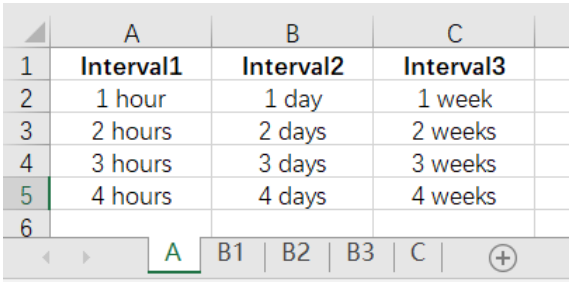
```
=spl("=E(?1)\E(?2)",'Y2018'!B1:B406,'Y2019'!B1:B1060)
```

	A	B	C	D	E
1	CustomerID				
2	VINET				
3	VINET				
4	VINET				
5	TOMSP				
6	TOMSP				
7	HANAR				
8	HANAR				
9	HANAR				
10	VICTE				
11	VICTE				
12	VICTE				
13	SUPRD				
14	SUPRD				
15	SUPRD				
16	HANAR				
17	HANAR				
18	HANAR				
19	CHOPS				
20	CHOPS				
21	CHOPS				

Y2018 | Y2019 | Sheet2 | Sheet1 | Sheet3

12.12 **Dynamic association operation**

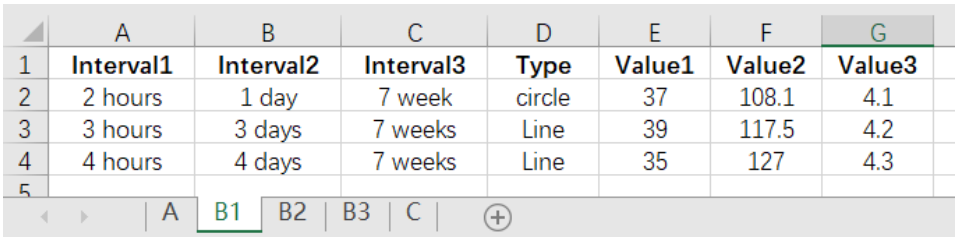
The sheets of the following Excel file are divided into three types, of which Sheet A is the basic table, part of the data is as follows:



	A	B	C
1	**Interval1**	**Interval2**	**Interval3**
2	1 hour	1 day	1 week
3	2 hours	2 days	2 weeks
4	3 hours	3 days	3 weeks
5	4 hours	4 days	4 weeks
6			

A | B1 | B2 | B3 | C

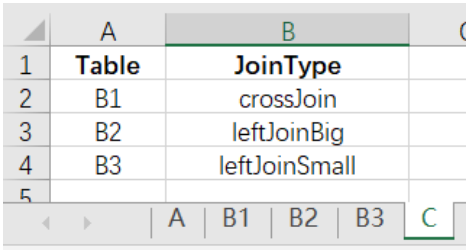
Sheet B1\B2…Bn are the association table, and all of these sheets have the same format, and have the same columns (Interval1, Interval2 and Interval3) as Sheet A. The figure below shows the data of Sheet B1:



	A	B	C	D	E	F	G
1	**Interval1**	**Interval2**	**Interval3**	**Type**	**Value1**	**Value2**	**Value3**
2	2 hours	1 day	7 week	circle	37	108.1	4.1
3	3 hours	3 days	7 weeks	Line	39	117.5	4.2
4	4 hours	4 days	7 weeks	Line	35	127	4.3
5							

A | B1 | B2 | B3 | C

Sheet C is used to describe the Join type between A and B1\B2..Bn. There are totally 3 types of joins, of which the **cross Join** represents Cartesian product, and both the **leftJoinBig** and **leftJoinSmall** represent the left association, and the associated columns are Interval1 and Interval2 respectively. See the following figure for details:



	A	B
1	**Table**	**JoinType**
2	B1	crossJoin
3	B2	leftJoinBig
4	B3	leftJoinSmall
5		

A | B1 | B2 | B3 | C

Calculation objective: associate sheet A with Sheet B1\B2..Bn according to the join type in sheet C, take the column Interva1 from sheet A, and take other columns from sheet B, and finally form a new two-dimensional table.

Let’s take the above sheet B as an example (actually each B should be different), if JoinType==crossJoin, the association result should be:

	A	B	C	D	E	F	G
1	Interval1	Interval2	Interval3	Type	Value1	Value2	Value3
2	1 hour	1 day	7 week	Circle	37	108.1	4.1
3	1 hour	3 days	7 weeks	Line	39	117.5	4.2
4	1 hour	4 days	7 weeks	Line	35	127	4.3
5	2 hours	1 day	7 week	circle	37	108.1	4.1
6	2 hours	3 days	7 weeks	Line	39	117.5	4.2
7	2 hours	4 days	7 weeks	Line	35	127	4.3
8	3 hours	1 day	7 week	circle	37	108.1	4.1
9	3 hours	3 days	7 weeks	Line	39	117.5	4.2
10	3 hours	4 days	7 weeks	Line	35	127	4.3
11	4 hours	1 day	7 week	circle	37	108.1	4.1
12	4 hours	3 days	7 weeks	Line	39	117.5	4.2
13	4 hours	4 days	7 weeks	Line	35	127	4.3

If joinType==leftJoinBig, the association result should be:

	A	B	C	D	E	F	G
1	Interval1	Interval2	Interval3	Type	Value1	Value2	Value3
2	1 hour						
3	2 hours	1 day	7 week	circle	37	108.1	4.1
4	3 hours	3 days	7 weeks	line	39	117.5	4.2
5	4 hours	4 days	7 weeks	line	35	127	4.3

If joinType==leftJoinSmall, the association result should be:

	A	B	C	D	E	F	G
1	Interval1	Interval2	Interval3	Type	Value1	Value2	Value3
2	1 hour						
3	2 hours						
4	3 hours	3 days	7 weeks	line	39	117.5	4.2
5	4 hours	4 days	7 weeks	line	35	127	4.3

Since this calculation needs to loop through sheet C and outputs multiple two-dimensional tables, it can only be implemented by using script instead of formula.

Script:

	A	B
1	=file("data.xlsx").xlsopen()	
2	=A1.xlsimport@t(;"C")	
3	=tableA=A1.xlsimport@t(;"A")	
4	for A2	=tableB=A1.xlsimport@t(;A4.Table)
5		=case(A4.JoinType, "crossJoin":xjoin(tableA:A;tableB:B), "leftJoinBig":xjoin@1 (tableA:A;tableB:B,A.Interval1==Interval1), "leftJoinSmall":xjoin@1(tableA:A;tableB:B,A.Interval1==Interval1 && A.Interval2==Interval2))
6		=B5.new(A.Interval1,B.Interval2,B.Interval3,B.Type,B.Value1,B.Value2,B.Value3)
7		=file(A4.Table+A4.JoinType+".xlsx").xlsexport@t(B6)

The script function case can judge the Join type, xjoin calculates the Cartesian product, and @1 represents the left association.

Chapter 13 **Conversion between rows and columns**

13.1 **Row-to-column conversion for fixed columns**

Here below is a data table:

	A	B	C	D
1	ID	Name	Subject	Score
2	110210	Lorry	Chinese	80
3	110210	Lorry	Maths	60
4	110210	Lorry	English	86
5	110211	Tom	Chinese	81
6	110211	Tom	Maths	72
7	110211	Tom	English	67
8	110212	Curry	Chinese	97
9	110212	Curry	Maths	91
10	110212	Curry	English	87
11	110213	Joan	Chinese	86
12	110213	Joan	Maths	69
13	110213	Joan	English	73

Now we want to convert the table to the form as shown in the figure below, and list the scores in the order of Chinese, Maths and English:

	A	B	C	D	E
1	ID	Name	Chinese	Maths	English
2	110210	Lorry	80	60	86
3	110211	Tom	81	72	67
4	110212	Curry	97	91	87
5	110213	Joan	86	69	73

Script:

```
=spl("=E(?1).pivot(ID,Name;Subject,Score;
""Chinese"",""Maths"",""English"")",Sheet1!A1:D13)
```

Perform the row-to-column conversion based on columns ID and Name. The values in the Subject column are transferred and used as the new column names, the values in the Score column are transferred and used as the values in the new columns, and the new column names are arranged in the order of "Chinese", "Maths", "English".

13.2 Convert row-based table to crosstab

The following data table records the daily sales of products:

	A	B	C
1	SaleDate	Product	Amount
2	2022/1/1	Bread	575
3	2022/1/1	Milk	701
4	2022/1/1	Cookie	877
5	2022/1/2	Milk	27
6	2022/1/3	Bread	203
7	2022/1/3	Milk	524
8	2022/1/3	Cookie	4
9	2022/1/4	Cookie	616
10	2022/1/5	Bread	766
11	2022/1/5	Milk	887
12	2022/1/5	Cookie	519
13	2022/1/6	Bread	679
14	2022/1/7	Bread	443
15	2022/1/7	Milk	592
16	2022/1/7	Cookie	197
17			

Sheet1

Now we want to create a crosstab, with the SaleDate as the left header of crosstab, the Products as the upper headers of crosstab, like this:

	A	B	C	D
1	**SaleDate**	**Bread**	**Cookie**	**Milk**
2	2022/1/1	507	458	254
3	2022/1/2			259
4	2022/1/3	341	151	900
5	2022/1/4		215	
6	2022/1/5	866	103	355
7	2022/1/6	619		
8	2022/1/7	488	121	166
9				

Script:

```
=spl("=E(?1).pivot(SaleDate;Product,Amount)",Sheet1!A1:C16)
```

Perform the row-to-column conversion based on SaleDate column, the values in the Product column are transferred and used as the new column names, and the values in the Amount column are transferred and used as the values in new columns.

13.3 Convert crosstab to row-based table

We have a crosstab that stores product's style data (width and length) and price information. The width data are stored in the first row, and the length data are stored in the first column:

Style	12	16	20	25	30	35	40	50	60	70
50		5.8	7.3	9.1	10.9					
55		6.4	8.0	10.0	12.0	14.0				
60		7.0	8.7	10.9	13.1	15.2				
65		7.5	9.4	11.8	14.1	16.5				
70		8.1	10.2	12.7	15.2	17.8	20.3	25.4		
75		8.7	10.9	13.6	16.3	19.0	21.8	27.2	32.6	
80		9.3	11.6	14.5	17.4	20.3	23.2	29.0		
85		9.9	12.3	15.4	18.5	21.6	24.7	30.8		
90		10.4	13.1	16.3	19.6	22.8	26.1	32.6		
95		11.0	13.8	17.2	20.7	24.1	27.6	34.4		
100		11.6	14.5	18.1	21.8	25.4	29.0	36.3		
105		12.2	15.2	19.0	22.8	26.6	30.5	38.1		
110		12.8	16.0	19.9	23.9	27.9	31.9	39.9		
115		13.3	16.7	20.8	25.0	29.2	33.4	41.7		
120		13.9	17.4	21.8	26.1	30.5	34.8	43.5		

Now we want to convert this table to a row-based table, and show product style as width*length, like this:

Style	Price
16*100	11.6
16*105	12.2
16*110	12.8
16*115	13.3
16*120	13.9
16*50	5.8
16*55	6.4
16*60	7.0
16*65	7.5
16*70	8.1
16*75	8.7
16*80	9.3
16*85	9.9
16*90	10.4
16*95	11.0
20*100	14.5
20*105	15.2
20*110	16.0
20*115	16.7
20*120	17.4
20*50	7.3
20*55	8.0

Script:

```
=spl("=E(?1).pivot@r(Style:Length;Width,Price).select(Price).sort(Width,Length).new(Width/""*
""/Length:Style,Price)",Sheet1!A1:K16)
```

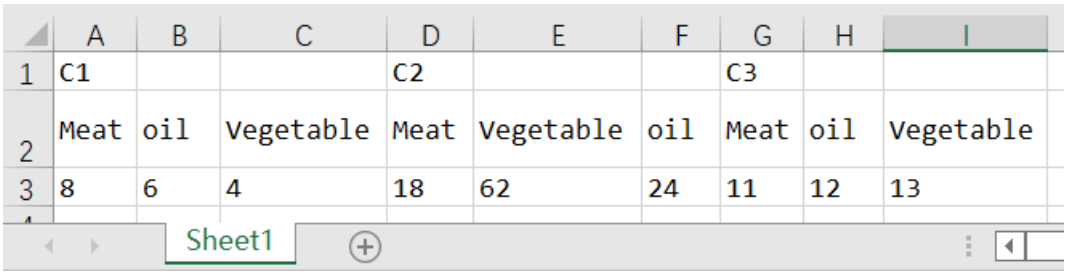
Perform the column-to-row conversion based on Style column, and give it a new name Length; the option @r means column-to-row conversion; the original column names are transferred and

used as the values in the new column Width, and the original values in the cross cells are transferred and used as the values in the new column Price.

After that, select the rows whose value in column Price is nonnull, and sort them by Width and Length; create a new dataset, with "Width*Length" as the values in the new column Style, and take the Price column as the price column of the new dataset.

13.4 Interconversion of upper layer groups for rows and columns - column-to-row

Here below is a data table:



	A	B	C	D	E	F	G	H	I
1	C1			C2			C3		
2	Meat	oil	Vegetable	Meat	Vegetable	oil	Meat	oil	Vegetable
3	8	6	4	18	62	24	11	12	13

Sheet1

The first row is the country code of each piece of data, the second row is other column names of each piece of data, and the third row is the value corresponding to each column. Now we want to rearrange the data into standard row-based data. The result is as follows:

	A	B	C	D
1	**Country**	**Meat**	**Oil**	**Vegetable**
2	C1	8	6	4
3	C2	18	24	62
4	C3	11	12	13

Script:

	A
1	=transpose('Sheet1!A1:I3').run(if(~(1)==null,~(1)=~[-1](1)))
2	=create(Country,Cate,Value)
3	>A1.(A2.record(~))
4	=A2.pivot(Country;Cate,Value)

A1: Transpose the sequence of sequences, and complement the countries in the first column

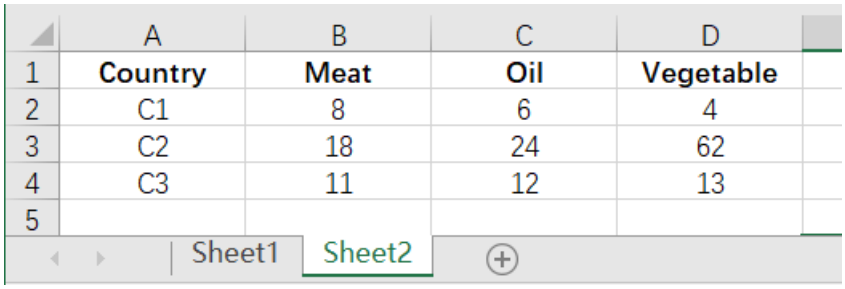
A2: Create an empty table sequence (Country,Cate,Value)

A3: Fill in the data of A1 in the table sequence one by one

A4: Use the pivot function to perform row-to-column conversion

13.5 Interconversion of upper layer groups for rows and columns - row-to-column

Here below is a data table:



	A	B	C	D
1	Country	Meat	Oil	Vegetable
2	C1	8	6	4
3	C2	18	24	62
4	C3	11	12	13
5				

Sheet1 Sheet2

The first row is the column name, including the Country column and other information columns. Now we want to rearrange the data into the following form:

	A	B	C	D	E	F	G	H	I
1	C1			C2			C3		
2	Meat	Oil	Vegetable	Meat	Vegetable	Oil	Meat	Oil	Vegetable
3	8	6	4	18	62	24	11	12	13

Script:

	A
1	=E('Sheet2!A1:D4').pivot@r(Country;Cate,Value;Meat,Vegetable,Oil)
2	=A1.group(Country).(~.run(Country=if(#==1,Country,""))).conj()
3	=transpose(A2.(#1|#2|#3))

A1: Perform the column-to-row conversion, and take the column names Meat, Vegetable and Oil as the values of Cate, and take the values in original Meat, Vegetable and Oil columns as the data in the Value column

A2: Group by Country, for each group, set the country values of non-first row as empty and then concatenate

A3: Merge the columns to become a sequence of sequences, transpose the sequence of sequences and return

13.6 **Put data in a group horizontally into columns**

Here below is a data table:

	A	B
1	**Name**	**Score**
2	Tom	87
3	John	99
4	Joan	98
5	Rocky	85
6	Ham	74
7	Kate	80
8	Rose	72
9	Nomy	73
10	Neil	100
11	Jack	74
12	Joe	76
13	Peter	84
14	Sunny	88
15	Tiger	80
16	Alice	72
17	Cindy	87
18	Leon	82
19	Lily	97
20	Kevin	72
21	Shelly	70
22	Panzy	92
23	Maggie	96
24	Mark	88
25	Aileen	75
26		

Sheet1

The task is to rank the students by score, and arrange the names with the same score into the same row. The result is as follows:

	A	B	C	D
1	1	Neil		
2	2	John		
3	3	Joan		
4	4	Lily		
5	5	Maggie		
6	6	Panzy		
7	7	Sunny	Mark	
8	8	Tom	Cindy	
9	9	Rocky		
10	10	Peter		
11	11	Leon		
12	12	Kate	Tiger	
13	13	Joe		
14	14	Aileen		
15	15	Ham	Jack	
16	16	Nomy		
17	17	Rose	Alice	Kevin
18	18	Shelly		

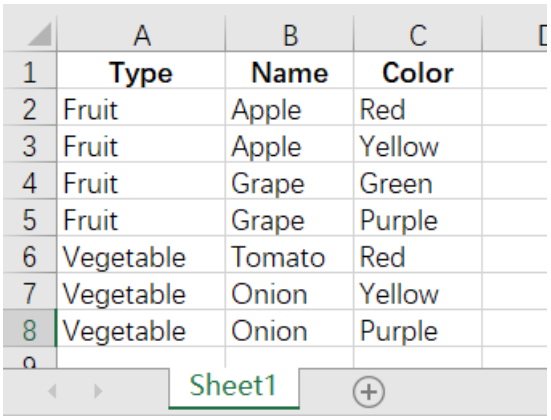
Script:

```
=spl("=E(?1).group(-Score).([#]|~.(Name))",Sheet1!A1:B25)
```

Group by Score and sort in reverse order; loop through each group, and merge the group number and the name of students in the group into a sequence.

13.7 Re-group or sort when filling grouped data into columns

There is a data table. In this table, the products with the same name may have multiple colors.



	A	B	C
1	**Type**	**Name**	**Color**
2	Fruit	Apple	Red
3	Fruit	Apple	Yellow
4	Fruit	Grape	Green
5	Fruit	Grape	Purple
6	Vegetable	Tomato	Red
7	Vegetable	Onion	Yellow
8	Vegetable	Onion	Purple

Sheet1

The task is to convert the data to the form as shown below, that is, arrange each type of product in one row, and then list the Name and Color of each product in turn.

	A	B	C	D	E	F	G
1	Fruit	Apple	Red	Yellow	Grape	Green	Purple
2	Vegetable	Onion	Yellow	Purple	Tomato	Red	
3							

Script:

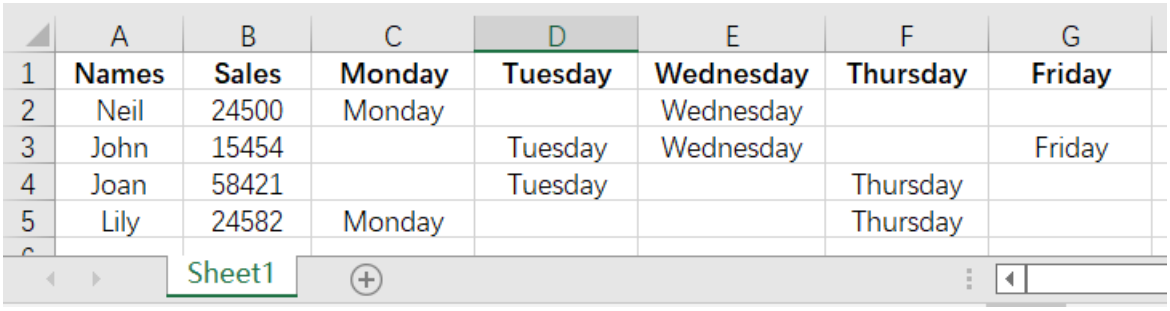
```
=spl("=E(?1).group(Type).(~.group(Name)).((~.Type|~.(Name|~.(Color))).conj@r())",Sheet1!A1:C8)
```

First group by Type, and then group by Name in each group.

Loop through groups of Type. First take Type, and then add the sequence composed of Name and Color of each group in its Name subgroups, and finally use conj to concatenate the sequence. Option @r means recursive calculating until all members are no longer a sequence.

13.8 Convert certain columns of the same row, as group members, to multiple rows

Here below is a data table:



	A	B	C	D	E	F	G
1	**Names**	**Sales**	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
2	Neil	24500	Monday		Wednesday		
3	John	15454		Tuesday	Wednesday		Friday
4	Joan	58421		Tuesday		Thursday	
5	Lily	24582	Monday			Thursday	

Sheet1

Now we want to convert it to a row-based table, like this:

	A	B	C
1	**Names**	**Sales**	**Day**
2	Neil	24500	Monday
3	Neil	24500	Wednesday
4	John	15454	Tuesday
5	John	15454	Wednesday
6	John	15454	Friday
7	Joan	58421	Tuesday
8	Joan	58421	Thursday
9	Lily	24582	Monday
10	Lily	24582	Thursday

Script:

```
=spl("=E(?1).pivot@r(Names,Sales;Weekday,Day).new(Names,Sales,Day).select(Day)",Sheet1!A1:G5)
```

Perform the row-column transposition on the data. The option @r means column-to-row. The Names and Sales columns are used as the base column, other column names are transferred and used as the values in the Weekday column, and the values in other columns are transferred and used as the values in column Day.

Select the columns Names, Sales and Day, and the rows whose value in column Day is nonnull.

13.9 Convert group formed by every N columns to multiple rows

Here below is a data table. In this table, the data are stored from the second column, and every two columns form a pair of columns (i.e., each even column and each odd column except the first column form a pair of columns, such as the second and third columns). There are a total of 4 pairs of such columns:

	A	B	C	D	E	F	G	H	I
1	**Micro**	**Group**	**Series**	**Group1**	**Series1**	**Group2**	**Series2**	**Group3**	**Series3**
2	1	Back	3	Biceps	1	Delts	3	Traps	4
3	1	Chest	2	Triceps	4	Delts	1		
4	1	Biceps	1						
5	1	Cuads	3	Glut	4				
6	2	Back	4	Biceps	4	Delts	3	Traps	4
7	2	Chest	5	Triceps	5	Delts	2		
8	2	Biceps	1						
9	2	Cuads	3	Glut	3				
10									

Sheet1

Now we want to group by the first column and all even columns, and then aggregate the odd columns in the group. The result should be as follows:

	A	B	C
1	**Micro**	**Group**	**Series**
2	1	Back	3
3	1	Biceps	2
4	1	Chest	2
5	1	Cuads	3
6	1	Delts	4
7	1	Glut	4
8	1	Traps	4
9	1	Triceps	4
10	2	Back	4
11	2	Biceps	5
12	2	Chest	5
13	2	Cuads	3
14	2	Delts	5
15	2	Glut	3
16	2	Traps	4
17	2	Triceps	5

Script:

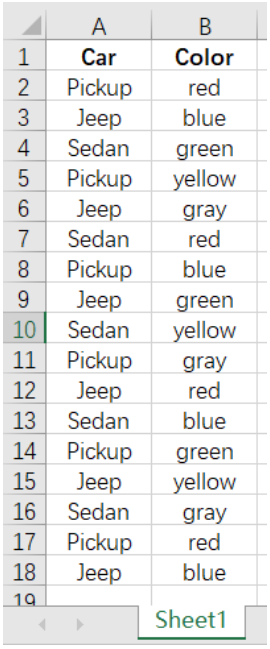
	A
1	='Sheet1!A2:I9'
2	=A1.news(~.len()\2;A1.~(1):Micro,A1.~(#*2):Group,A1.~(#*2+1):Series).select(Group!=null)
3	=A2.groups(Micro,Group;sum(Series):Series)

A2: Expand each row of A1, and the number of rows expanded is the quotient of the number of members of the row divided by 2. In the expanded row, the first column Micro is the first member of A1, the second column Group is the #*2(th) member of A1, and the third column Series is the #*2+1(th) member of A1. The symbol # here represents the row number expanded by the row.

A3: Group A2 by Micro and Group, and calculate the sum of Series and name it Series column.

13.10 Convert groups to columns after grouping

Here below is a data table with two columns (Car and Color):



	A	B
1	**Car**	**Color**
2	Pickup	red
3	Jeep	blue
4	Sedan	green
5	Pickup	yellow
6	Jeep	gray
7	Sedan	red
8	Pickup	blue
9	Jeep	green
10	Sedan	yellow
11	Pickup	gray
12	Jeep	red
13	Sedan	blue
14	Pickup	green
15	Jeep	yellow
16	Sedan	gray
17	Pickup	red
18	Jeep	blue
19		

Sheet1

Now we want to convert the data to the following form with the car type as column name, and list all colors of each type of car and remove the duplicate color:

	A	B	C
1	**Jeep**	**Pickup**	**Sedan**
2	blue	blue	blue
3	gray	gray	gray
4	green	green	green
5	red	red	red
6	yellow	yellow	yellow

Script:

	A
1	=E('Sheet1!A1:B18')
2	=A1.group(Car).(Car|~.id(Color))
3	=A2.max(~.len())
4	=A2.(~.pad(null,A3))
5	=transpose(A4)

A2: Group by Car. In each group, form a sequence with car name and its distinct colors. ~.id(Color) means taking the color that is unique in this group

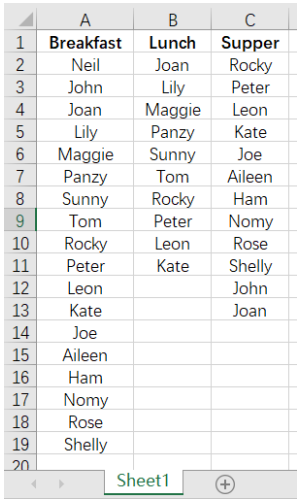
A3: Calculate the maximum length of all grouped sequences

A4: Use null value to complement the sequence of each group to the maximum length for transposing

A5: Transpose rows and columns of A4

13.11 Rearrange multiple columns into a cross-tab

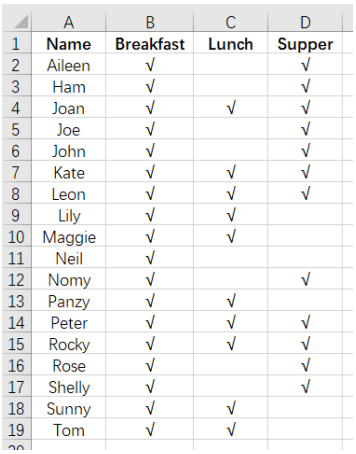
Here below is a data table:



	A	B	C
1	**Breakfast**	**Lunch**	**Supper**
2	Neil	Joan	Rocky
3	John	Lily	Peter
4	Joan	Maggie	Leon
5	Lily	Panzy	Kate
6	Maggie	Sunny	Joe
7	Panzy	Tom	Aileen
8	Sunny	Rocky	Ham
9	Tom	Peter	Nomy
10	Rocky	Leon	Rose
11	Peter	Kate	Shelly
12	Leon		John
13	Kate		Joan
14	Joe		
15	Aileen		
16	Ham		
17	Nomy		
18	Rose		
19	Shelly		
20			

Sheet1

The task is to convert the data to the following form:



	A	B	C	D
1	**Name**	**Breakfast**	**Lunch**	**Supper**
2	Aileen	√		√
3	Ham	√		√
4	Joan	√	√	√
5	Joe	√		√
6	John	√		√
7	Kate	√	√	√
8	Leon	√	√	√
9	Lily	√	√	
10	Maggie	√	√	
11	Neil	√		
12	Nomy	√		√
13	Panzy	√	√	
14	Peter	√	√	√
15	Rocky	√	√	√
16	Rose	√		√
17	Shelly	√		√
18	Sunny	√	√	
19	Tom	√	√	

Script:

	A
1	='Sheet1!A1:C19'
2	=create(Meal,Name,Flag)
3	=A1.to(2,).run(~.run(if(~!=null,A2.record([A1(1)(#),~ , "√"]))))
4	=A2.pivot(Name;Meal,Flag)

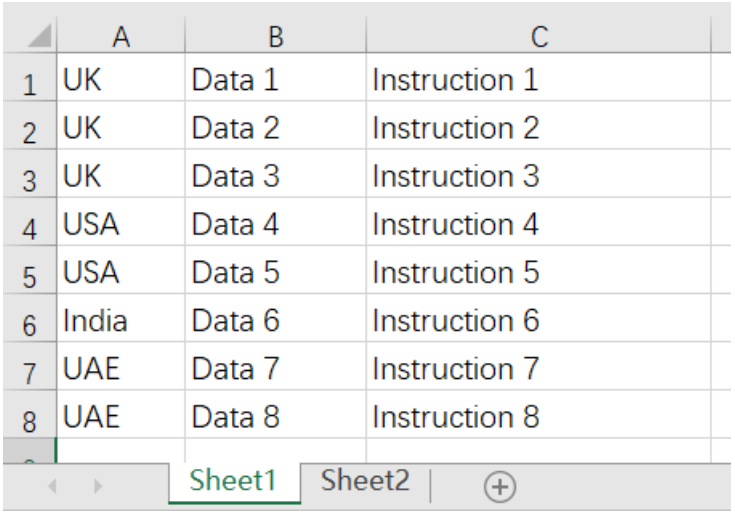
A2: Create a table sequence having three columns Meal, Name and Flag

A3: Loop through every row starting from row 2 of A1, and then loop through the data sequence of each row; If the sequence member is not empty, insert its corresponding column name and itself into the table sequence of A2 in turn, and set Flag to √

A4: Group A2 by Name to perform row-to-column conversion, with the Meal values as new column names, and Flag as the new column values

13.12 Interconversion of rows and columns within a group

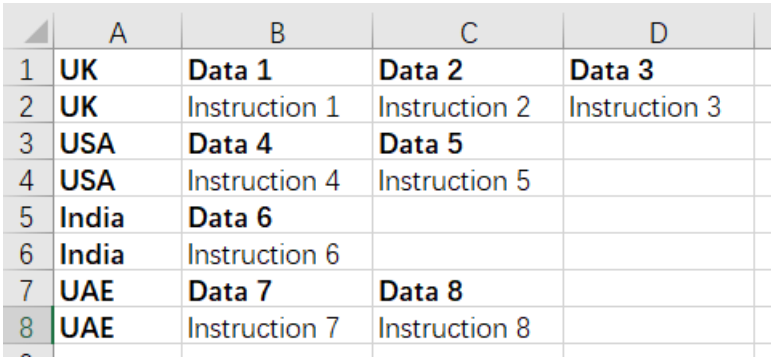
Here below is a data table:



	A	B	C
1	UK	Data 1	Instruction 1
2	UK	Data 2	Instruction 2
3	UK	Data 3	Instruction 3
4	USA	Data 4	Instruction 4
5	USA	Data 5	Instruction 5
6	India	Data 6	Instruction 6
7	UAE	Data 7	Instruction 7
8	UAE	Data 8	Instruction 8

Sheet1 Sheet2

Now we want to convert the data to the following form:



	A	B	C	D
1	**UK**	**Data 1**	**Data 2**	**Data 3**
2	**UK**	Instruction 1	Instruction 2	Instruction 3
3	**USA**	**Data 4**	**Data 5**	
4	**USA**	Instruction 4	Instruction 5	
5	**India**	**Data 6**		
6	**India**	Instruction 6		
7	**UAE**	**Data 7**	**Data 8**	
8	**UAE**	Instruction 7	Instruction 8	

Script:

	A
1	='Sheet1!A1:C8'
2	=A1.group@u(~(1))
3	=A2.(transpose(~.(~.to(2,))))
4	=A3.(~.(A2.(~(1)(1))(A3.#)|~)).conj()

A2: Group by the first column (Countries)

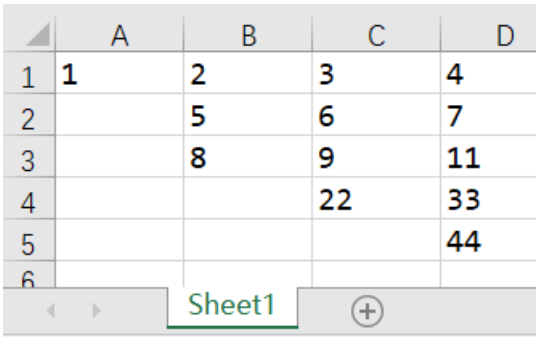
A3: Transpose the data in each group (except the country column)

A4: Concatenate the country and merge

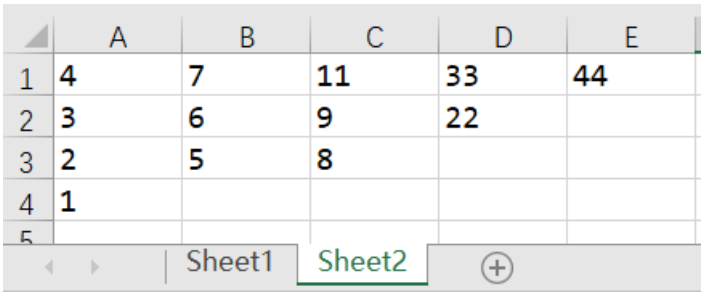
The processing method described above is used for column-to-row conversion, this method works for row-to-column conversion as well.

13.13 Interconversion of rows and columns in reverse order

Here below is a data table:



Now we want to transpose the M columns of a two-dimensional table to M rows, and the transposition order should be: the M-th column, M-1 column, M-2...2, 1. The results are as follows:



Script:

```
=spl("=transpose(?1).rvs()",Sheet1!A1:D5)
```

Reverses the order of the columns after transposing.

Script for reverse transposing:

```
=spl("=transpose(?1).(~.rvs())",Sheet2!A1:E4)
```

It should be noted what is reversed here is the order of rows.

Chapter 14 **Expand and complement**

14.1 **Generate continuous array**

There is a column of serial numbers (No), as shown in the figure below:

	A
1	**No**
2	1-5
3	8-12

Now we want to use the two numbers in each row to expand into continuous number intervals, like this:

No
1
2
3
4
5
8
9
10
11
12

Script:

```
=spl("=E(?1).news((a=No.split(""-"").(int(~)),to(a(1),a(2)));~:No)",A1:A3)
```

Loop through each row, and split No with the minus sign, and then convert the split numbers to an integer sequence and assign it to the variable a, and finally form a continuous sequence of numbers with a(1) and a(2). The function news means expanding each row into multiple rows according to the number of members of the number sequence, and the numeric member is the value of No in the new row.

14.2 Generate continuous array - concatenate results into a string

The values in columns A and B are natural numbers, representing the start and end points of the array respectively:

	A	B	C
1	Start	End	Result
2	1	10	
3	9	3	
4	2	8	

The task is to use the values in columns A and B to generate a string composed of natural number array, and fill in the result in column C, as follows:

	A	B	C
1	Start	End	Result
2	1	10	1,2,3,4,5,6,7,8,9,10
3	9	3	9,8,7,6,5,4,3
4	2	8	2,3,4,5,6,7,8

Enter in cell C2:

```
=spl("=to(?1,?2).concat@c()",A2,B2)
```

	A	B	C
1	Start	End	Result
2	1	10	1,2,3,4,5,6,7,8,9,10
3	9	3	
4	2	8	

Then drag C2 down to every relevant row:

	A	B	C
1	Start	End	Result
2	1	10	1,2,3,4,5,6,7,8,9,10
3	9	3	9,8,7,6,5,4,3
4	2	8	2,3,4,5,6,7,8
5			

14.3 Expand one row into multiple rows based on value

There is a data table, which stores some information of products. The first column is the serial number, and the second column is the remaining quantity. The initial data is as follows:

	A	B	C
1	ItemID	QtyRemaining	InvoiceNo
2	00001	2	404040
3	00002	9	505050
4	00003	0	606060

Now we want to copy each value in column ItemID n times according to the remaining quantity of product (i.e., the number in Qty Remaining column). The requirements are: 1) Except for the first row (as the original row), the other copied rows only retain the value in column ItemID; 2) The remaining quantity of product 00003 is 0, this row will be no longer retained in the new table. The expected result is as follows:

ItemID	QtyRemaining	InvoiceNo
1	2	404040
1		
2	9	505050
2		
2		
2		
2		
2		
2		
2		
2		

Script:

```
=spl("=E(?1).news(#2;'ItemID',if(#==1,'QtyRemaining',null):'QtyRemaining',if(#==1,'InvoiceNo',
null):'InvoiceNo')",A1:C4)
```

Copy each row n times according to the value in the second column to generate a new table sequence, where the values in the second and third columns use the if expression. If it is the first row, take the original value, otherwise take the null value.

14.4 Expand one row into multiple rows after splitting text

The following is a data table, in which columns D and E have multiple lines of text, the number of lines is the same, and such lines are in one-to-one correspondence. For example, F corresponds to Fail, as shown below:

	A	B	C	D	E
1	Names	Class	Year	Grades	Comment
2	Mark	2nd	2012	A F C	Very Good, Needs Improvement Fail Satisfactory
3	Alice	4th	2012	F F	Fail Fail
4	Sunny	5th	2012	A B C	Very Good, Needs Improvement Good, Needs Improvement Satisfactory

Now we want to split the values in column D, E by line break, and expand into multiple rows to make the result look like this:

Names	Class	Year	Grades	Comment
Mark	2nd	2012	A	Very Good, Needs Improvement
Mark	2nd	2012	F	Fail
Mark	2nd	2012	C	Satisfactory
Alice	4th	2012	F	Fail
Alice	4th	2012	F	Fail
Sunny	5th	2012	A	Very Good, Needs Improvement
Sunny	5th	2012	B	Good, Needs Improvement
Sunny	5th	2012	C	Satisfactory

Script:

```
=spl("=E(?1).run(Grades=Grades.split(""\n""),Comment=Comment.split(""\n"")).news(Grades.len
        ();Names,Class,Year,Grades(#):Grades,Comment(#):Comment)",A1:E4)
```

Loop through each row, split Grades and Comment into a string sequence by \n respectively, and then expand each row into multiple rows, the number of rows is the number of members of Grades sequence. In each new row, take the original Names, Class, and Year columns, the #th member of the Grades sequence is the Grades in new column, and the #th member of the Comment sequence is the Comment in new column, where # represents the row number expanded from original row.

14.5 Make up missing parts to make data continuous

The following table records the daily sales data of products, and some dates are missing due to no sales data:

	A	B	C
1	SaleDate	Product	Amount
2	2022-04-01	Milk	105
3	2022-04-02	Bread	159
4	2022-04-03	Cookie	197
5	2022-04-05	Milk	199
6	2022-04-06	Bread	200
7	2022-04-10	Cookie	142
8	2022-04-11	Milk	107
9	2022-04-14	Bread	140
10	2022-04-15	Cookie	150
11	2022-04-17	Milk	154

Now we want to list the daily sales data in the order of date, and make up the dates missed in the original table, as shown in the following figure:

SaleDate	Product	Amount
2022-04-01	Milk	105
2022-04-02	Bread	159
2022-04-03	Cookie	197
2022-04-04		
2022-04-05	Milk	199
2022-04-06	Bread	200
2022-04-07		
2022-04-08		
2022-04-09		
2022-04-10	Cookie	142
2022-04-11	Milk	107
2022-04-12		
2022-04-13		
2022-04-14	Bread	140
2022-04-15	Cookie	150
2022-04-16		
2022-04-17	Milk	154

Script:

	A
1	=E('A1:C11')
2	=A1.min(SaleDate)
3	=A1.max(SaleDate)
4	=A2|(A3-A2).(A2+~)
5	=A1.align(A4,SaleDate)
6	=A5.new(A4(#):SaleDate,Product,Amount)

A2: Find the minimum date in A1

A3: Find the maximum date in A1

A4: Concatenate all dates from the minimum date to the maximum date in order into a sequence

A5: Align the SaleDate of A1 in the order of A4

A6: Use the function new to reassign each row of A5, # represents the current row number of A5, take the date value of the same row number in A4 as SaleDate, and then take the Product and Amount columns of the current row of A5

14.6 **Add several blank rows every N rows**

There is a data table that stores the detailed data of student examination room and seat number, and part of the data is as follows:

	A	B	C	D
1	**Id**	**Name**	**Seat no**	**Classroom**
2	110201	Mike	1	101
3	110202	Joan	2	101
4	110203	Kate	3	101
5	110204	Tim	4	101
6	110205	Jack	5	101
7	110206	Lorry	6	101
8	110207	Jim	7	101
9	110208	John	8	101
10	110209	Tom	9	101
11	110210	Rose	10	101
12	110211	Jordan	11	101
13	110212	Leon1	12	101
14	110213	Leon2	13	101
15	110214	Leon3	14	102
16	110215	Leon4	15	102
17	110216	Leon5	16	102
18	110217	Leon6	17	102
19	110218	Leon7	18	102
20	110219	Leon8	19	102
21	110220	Leon9	20	102
22	110221	Leon10	21	102
23	110222	Leon11	22	102
24	110223	Leon12	23	102
25	110224	Leon13	24	102
26	110225	Leon14	25	102
27	110226	Leon15	26	102
28	110227	Leon16	27	103
29	110228	Leon17	28	103
30	110229	Leon18	29	103
31	110230	Leon19	30	103
32	110231	Leon20	31	103
33	110232	Leon21	32	103
34	110233	Leon22	33	103
35	110234	Leon23	34	103
36	110235	Leon24	35	103
37	110236	Leon25	36	103

Now we want to add two blank rows after every 13 rows. The result is as follows:

Id	Name	Seat no	Classroom
110201	Mike	1	101
110202	Joan	2	101
110203	Kate	3	101
110204	Tim	4	101
110205	Jack	5	101
110206	Lorry	6	101
110207	Jim	7	101
110208	John	8	101
110209	Tom	9	101
110210	Rose	10	101
110211	Jordan	11	101
110212	Leon1	12	101
110213	Leon2	13	101
110214	Leon3	14	102
110215	Leon4	15	102
110216	Leon5	16	102
110217	Leon6	17	102
110218	Leon7	18	102
110219	Leon8	19	102
110220	Leon9	20	102
110221	Leon10	21	102
110222	Leon11	22	102
110223	Leon12	23	102
110224	Leon13	24	102
110225	Leon14	25	102
110226	Leon15	26	102
110227	Leon16	27	103
110228	Leon17	28	103
110229	Leon18	29	103
110230	Leon19	30	103
110231	Leon20	31	103
110232	Leon21	32	103

Script:

	A
1	=E('A1:D152')
2	=row=A1.create().insert(0)
3	=A1.group((#-1)\13).(~|row|row).conj()

A2: Copy the data structure of A1, insert one blank row, and assign this row to the variable row

A3: Take every 13 students into a group, add two blank rows after each group, and then concatenate each group into a data set

14.7 **Insert row after specific row**

Here below is a data table:

	A
1	1
2	2
3	100
4	2
5	3
6	6
7	100
8	5
9	100

The task is to insert one row after the number 100, and fill in a001, a002, ... in turn, like this:

1
2
100
a001
2
3
6
100
a002
5
100
a003

Script:

```
=spl("=E@b(?1).group@i(~[-1].#1==100).(~|new(string(#,""a000""):_1)).conj()",A1:A9)
```

Group the data. When the value in column A of the previous row is 100, create a new group.

Loop through each group, insert a record at the end of the group, whose value of the first column is the current group number #, formatted with "a000".

14.8 **Insert blank row when meeting with data change**

Here below is a data table:

	A
1	AHMAD MAULUD
2	AHMAD MAULUD
3	AHMAD MAULUD
4	DOLLY INDRA SIREGAR
5	DOLLY INDRA SIREGAR
6	DOLLY INDRA SIREGAR
7	MANCHESTER
8	MANCHESTER
9	MANCHESTER

We want to insert one blank row when the values in two adjacent rows change, the result is as follows:

AHMAD MAULUD
AHMAD MAULUD
AHMAD MAULUD
DOLLY INDRA SIREGAR
DOLLY INDRA SIREGAR
DOLLY INDRA SIREGAR
MANCHESTER
MANCHESTER
MANCHESTER

Script:

```
=spl("=?1.conj().group@o(~).(~|[null]).conj().new(~:_1)",A1:A9)
```

14.9 Expand into multiple columns horizontally

The following data table stores a variety of parts and their metal subparts in an orderly manner. When Level=2, it indicates that this row is the part row (aggregation row), and when Level=3, it indicates this row is the subpart row. The columns Proportion and Material store the proportion and name of metal for the subparts respectively.

	A	B	C	D
1	Name	Level	Proportion	Material
2	Part 1	2		
3	Part 1 A	3	0.37	Ally
4	Part 1 B	3	0.4	Ally
5	Part 1 C	3	0.04	Copper
6	Part 1 D	3	0.01	Titainium
7	Part 1 E	3	0.04	Steel
8	Part 1 F	3	0.07	Titainium
9	Part 1 G	3	0.07	Copper
10	Part 2	2		
11	Part 2 A	3	0.5	Steel
12	Part 2 B	3	0.5	Ally

The task is to put the total proportion of each metal on the right side of aggregation row:

	A	B	C	D	E	F	G	H
1	Name	Level	Proportion	Material	Ally	Copper	Steel	Titainium
2	Part 1	2			0.77	0.11	0.04	0.08
3	Part 1 A	3	0.37	Ally				
4	Part 1 B	3	0.4	Ally				
5	Part 1 C	3	0.04	Copper				
6	Part 1 D	3	0.01	Titainium				
7	Part 1 E	3	0.04	Steel				
8	Part 1 F	3	0.07	Titainium				
9	Part 1 G	3	0.07	Copper				
10	Part 2	2			0.5		0.5	
11	Part 2 A	3	0.5	Steel				
12	Part 2 B	3	0.5	Ally				

Enter in cell E1:

	A
1	=E('A1:D12')
2	=A1.id(Material).select(~)
3	=A1.derive(${A2.concat@c()})
4	=A3.group@i(Level==2)
5	=A4.run(~.to(2,).groups(Material;sum(Proportion):value).run(A4.~(1).field(Material,value)))
6	=A3.new(${A2.concat@c()})

A2: Find out the type of Material that is unique and nonnull

A3: Append one column for each type of Material found in A2

A4: Group A3, and create a new group when Level is 2. The option @i means creating a new group when the condition is met

A5: Loop through each group of A4, and count the sum of proportions of each material by the Material group from the second row to the last row, and name it the value column, and loop through each group, and assign Material column of the first row of the current group in A4 to value

A6: Select the required columns from the results in A3 and return

14.10 Expand into multiple N-column horizontally

There is a registration table, which records the entering and leaving time of customers in a certain bathroom on a certain day, and part of the data is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1						18:00-19:00		19:00-20:00		20:00-21:00		21:00-22:00	
2	Name	Sex	Enter	Leave									
3	Mike	M	18:05	19:40									
4	Joan	F	19:00	20:50									
5	Kate	F	18:36	21:28									
6	Tim	M	18:15	21:00									
7	Jack	M	19:08	21:10									
8	Lorry	M	19:32	20:55									
9	Jim	F	19:45	21:40									
10	John	M	20:00	21:36									
11	Tom	M	20:08	22:00									
12	Rose	F	20:15	21:00									
13	Jordan	M	20:20	21:42									
14	Lily	F	20:27	21:38									
15	Susan	F	20:30	21:58									

The task is to count the situation of customers in each one-hour time period from 18:00 to 22:00. If the customer is in the bathroom during a time period, fill in 1, otherwise leave it blank, as shown in the following figure:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1						18:00-19:00		19:00-20:00		20:00-21:00		21:00-22:00	
2	Name	Sex	Enter	Leave		Male	Female	Male	Female	Male	Female	Male	Female
3	Mike	M	18:05	19:40		1		1					
4	Joan	F	19:00	20:50					1		1		
5	Kate	F	18:36	21:28			1		1		1		1
6	Tim	M	18:15	21:00		1		1		1			
7	Jack	M	19:08	21:10				1		1		1	
8	Lorry	M	19:32	20:55				1		1			
9	Jim	F	19:45	21:40					1		1		1
10	John	M	20:00	21:36						1		1	
11	Tom	M	20:08	22:00						1		1	
12	Rose	F	20:15	21:00							1		
13	Jordan	M	20:20	21:42						1		1	
14	Lily	F	20:27	21:38							1		1
15	Susan	F	20:30	21:58							1		1

Enter in cell F2:

	A
1	=E('A2:D15')
2	=create(${(["Male","Female"]*4).concat@c()})
3	=interval@s(time("00:00","HH:mm"),time("18:00","HH:mm"))/86400
4	=A1.run(A2.insert(0),4.run(t1=A3+(~-1)*3600/86400,t2=A3+~*3600/86400,k=if(A1.Sex=="M",~*2-1,~*2),if(A1.Enter<t2 && A1.Leave>t1,A2(A1.#).field(k,1))))
5	return A2

A2: Create a table sequence using 4 groups of Male and Female as its columns

A3: Convert the start time 18:00 to the value as which Excel store

A4: Loop through each row of A1, and append a new row in A2; loop through 4 time periods, and calculate the start time t1 and end time t2 of each time period; calculate the column number k

to be filled in according to the gender of the customer. If the customer is still in the bathroom during a period, then fill in 1 in the kth column of the current row of A2

14.11 Generate permutations and combinations

Here below is a data table, in which the data are separated by commas:

	A	B	C
1	A,B	1,2,3	0,1,2
2			

Now we want to list the permutations and combinations of these data, one combination per row, as shown in the following figure:

	A	B	C
1	A,B	1,2,3	0,1,2
2	A	1	0
3	A	1	1
4	A	1	2
5	A	2	0
6	A	2	1
7	A	2	2
8	A	3	0
9	A	3	1
10	A	3	2
11	B	1	0
12	B	1	1
13	B	1	2
14	B	2	0
15	B	2	1
16	B	2	2
17	B	3	0
18	B	3	1
19	B	3	2

Enter in cell A2:

	A
1	='A1:C1'.conj().(~.split@c())
2	=A1.("A1("/#/")").concat(";")
3	=xjoin(${A2})

A1: Split the data in each cell with comma, and the option @c means splitting with comma

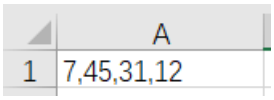
A2: Loop through each member of A1 and concatenate them into a string A1(1);A1(2);A1(3)

A3: Convert the string concatenated in A2 to the parameter of the function xjoin in macro form, that is, perform full cross-product on the sequences formed by split data in each cell

Chapter 15 **Operations on text**

15.1 **Split string - separate by comma – automatic parsing of data type**

Here is a comma-separated number string. To find the largest number:



	A
1	7,45,31,12

Script:

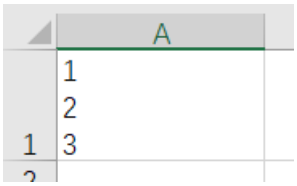
```
=spl("=?1.split@cp().max()",A1)
```

Result: 45

The function split is used to split the string into a sequence; the option @c means splitting by comma (you don't have to use this option, you can use the parameter "," instead); the option @p means automatically parsing the data type, for example, the integer string can be parsed into integers.

15.2 **Split string - separate by carriage return (CR) - automatic parsing of data type**

Here is a CR-separated number string. To find the largest number:



	A
	1
	2
1	3

Script:

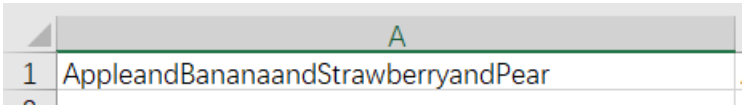
```
=spl("=?1.split@np().max()",A1)
```

Result: 3

The function split is used to split the string into a sequence; the option @n means splitting by CR (you don't have to use this option, you can use the parameter "\n" instead); the option @p means automatically parsing the data type, for example, the integer string can be parsed into integers.

15.3 Split string - separate by multi-character separator

Here is a string separated by multi-character separator. To split this string into multiple strings by the separator “and”:



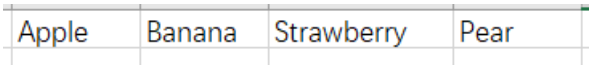
	A
1	AppleandBananaandStrawberryandPear

Script:

```
=spl("=?1.split@(""and"")",A1)
```

The function split is used to split the string into a sequence; the parameter "and" is used as the separator.

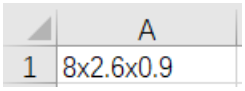
Result:



Apple	Banana	Strawberry	Pear

15.4 Concatenate into string

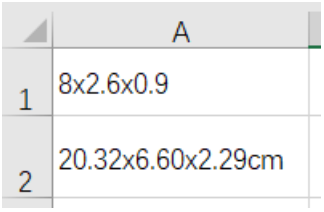
The following is the size data in inches.



	A
1	8x2.6x0.9

We want to convert the data to the one in centimeters and write the result to A2:

```
=spl("=?1.split@p(""x"").(string(~*2.54,""#.00"")).concat(""x"")+""cm""",A1)
```



	A
1	8x2.6x0.9
2	20.32x6.60x2.29cm

Use the function split to split the string into a sequence by the separator x, the option @p means automatically parsing the data type; convert the data in inches in the sequence to the data in centimeters, and convert the converted data to a string sequence and retain two decimal places; use the function concat to concatenate the string sequence into a string with the separator x, and concatenate the centimeter unit "cm".

15.5 Parse and extract numbers

Below is a table with a column of data that is a mixture of characters and numbers. The numbers appear in multiple places irregularly:

	A
1	aBc1d5_x9
2	12:2&0
3	ABC123
4	ABCD4578124YUIOH
5	888ABC123
6	
7	
8	

Now we want to extract all the numbers from each row and put them in column B. The expected result is as follows:

	A	B
1	aBc1d5_x9	159
2	12:2&0	1220
3	ABC123	123
4	ABCD4578124YUIOH	4578124
5	888ABC123	888123
6		

Enter in cell B1:

```
=spl("=?1.words@d().concat()",A1)
```

Then drag B1 down to every relevant row

Use the words@d to split the string in each row to extract all numbers to form a sequence, and then use the concat to concatenate every number together.

15.6 **Parse and extract dates**

The following table stores the event memo information:

	A
1	6.5.18 - no int in zone 1 but int in zone 2. 18.7.19 - summer update.
2	12.7.19 - may have French investor. 11.9.19 - invests in series c firms
3	4.5.17 - James interested in purchase. 14.3.18 - only invest in passed ventures. 20.11.18 - sent material. 24.7.19 - summer update.
4	Zone IV 5.9.19 - not available

For the convenience of future statistics, we want to extract all the date data from each row, and put them in the next column after separating by semicolon, like this:

	A	B
1	6.5.18 - no int in zone 1 but int in zone 2. 18.7.19 - summer update.	2018/05/06;2019/07/18
2	12.7.19 - may have French investor. 11.9.19 - invests in series c firms	2019/07/12;2019/09/11
3	4.5.17 - James interested in purchase. 14.3.18 - only invest in passed ventures. 20.11.18 - sent material. 24.7.19 - summer update.	2017/05/04;2018/03/14;2018/11/20;2019/07/24
4	Zone IV 5.9.19 - not available	2019/09/05

Enter in cell B1:

```
=spl("=?1.split("" "").(date(~,""dd.MM.yy"")).select(ifdate(~)).concat("";"")",A1)
```

Then drag B1 down to every relevant row.

Split the string with spaces to form a sequence;

Convert the text string in the sequence to the date type data according to the specified format;

Select the date type data and concatenate them into a string with semicolon.

15.7 Take out different types of characters

The following data table stores multiple types of attendance records of employees on a certain day. For example, A means late arrival, and the number after it represents how long he/she is late; B means early leave, and the number after it represents how long he/she leaves early; C means asking for leave, the number after it represents how long he/she asks for leave, and so on. Now we want to summarize the attendance situation, and then count the sum of the numbers after each letter, and finally fill in the results in the black box.

	A	B	C	D	E	F	G
1	A1.5B3.28C6.01D8.9E9.55F0.36	B6.6D3.7	A0.69D3.9F20	E1.5C5.9	F0.3	A3.6B0.5	F3.9B6.8C11.8
2							
3							
4		Type	Total				
5		A	5.79				
6		B	17.18				
7		C	23.71				
8		D	16.5				
9		E	11.05				
10		F	24.56				

Enter in cell B4:

	A
1	='A1:G1'.conj()
2	=create(Type,Value)
3	=A1.(~.words@wp().run(if(#%2==1,A2.record([~,number(~[1])]))))
4	=A2.groups(Type;sum(Value):Total)

A2: Create a table sequence having two columns: Type and Value.

A3: Loop through the string in each cell, and split the string into words. The option @w means that the string will be thoroughly split in such a way that the Chinese characters/symbols are split into single character, and the English words/numbers are split into single word; the option p means that the beginning of the data will be recognized as a numeric value or a date according to the data type, and split into a whole. After splitting, loop through the split sequence, if the current member's number is odd, take the current member and its next member and convert them into a value to form a row of data and save it in A2

A4: Group A2 by Type and calculate the sum of Value

15.8 **Take out words**

There is a text data table:

	A
1	Guo Weimin(郭卫民), spokesperson of the fourth session of the 13th CPPCC National Committee, whose age is 50, will brief journalists on the session and take questions.

To take out the number strings in the text:

```
=spl("=?1.words@d()",A1)
```

50	13

To take out the English words and the number strings in the text:

```
=spl("=?1.words@a()",A1)
```

Guo	Weimin	aged	50	and	spokesperson	of	the	fourth	session	of	the	13	th	CPPCC	National	Committee	will	brief	journalists	on	the	session	and	take	questions

To take out all the characters: split Chinese characters/symbols into single character, and split English words/numbers into single word:

```
=spl("=?1.words@w()",A1)
```

Guo	Weimin	(| 郭 | 卫 | 民 |)	,	aged	50	and	spokesperson	of	the	fourth	session	of	the	13	th	CPPCC	National	Committee	,	will	brief	journalists	on	the	session	and	take	questions	.

15.9 Parse and extract Key-Value pair

Data 1:

	A
1	A=123 B=456 C=789 D=INV E=67

Script 1-1:

```
=spl("=?1.property()",A1)
```

The function property is used to read the attribute value from a KV string. When there is no parameter, it means to return all attributes to form a table sequence.

Result 1-1:

name	value
A	123
E	67
D	INV
C	789
B	456

Script 1-2:

```
=spl("=?1.property@v(""A"")",A1)
```

The function property is used to read the attribute value of A from a KV string. @v means that it will be parsed into a value after reading it out.

Result 1-2:

123

Data 2:

	A
1	color:red;size:20;price:500

Script 2:

```
=spl("=?1.property@vcj()",A1)
```

The function property is used to read the attribute value from a KV string. When there is no parameter, it means to return all attributes to form a table sequence. @v means that the string is read first and then parsed into the value; the option @c means that the sectioned strings are separated by a comma or semicolon.

Result 2:

name	value
color	red
size	20
price	500

Data 3:

	A
1	{S1}ADD1;{S2}ADD2;{S3}ADD17;{S8}AD234

Script 3:

```
=spl("=replace(replace(?1,""{"","""""),""}"","""="").property@vc()",A1)
```

Change the format of KV string to K=V.

The function property is used to read the attribute value from a KV string. When there is no parameter, it means to return all attributes to form a table sequence. @v means that the string is read first and then parsed into the value; the option @j means that the name and value of byte strings are separated by comma or semicolon.

Result 3:

name	value
S3	ADD17
S2	ADD2
S1	ADD1
S8	AD234

Chapter 16 **Operations on date and time**

16.1 **Count date by year and month**

Here below is a data table:

	A	B
1	Date	Val
2	2018/1/1	51
3	2019/1/1	52
4	2019/2/1	37
5	2019/2/5	32
6	2019/3/5	95
7	2019/3/10	2
8	2019/3/19	9
9	2019/3/19	85
10	2019/4/15	40
11	2019/4/29	6

Now we want to calculate the sum of values in column Val by the year and month in column Date, the result is as follows:

YM	Total
201801	51
201901	52
201902	69
201903	191
201904	46

Script:

```
=spl("=E(?1).run(Date=date(""18991230"",""yyyyMMdd"")+Date).groups(month@y(Date):
YM;sum(Val):Total)",A1:B11)
```

Group the dates in column Date by year and month, and then calculate the sum of the corresponding data in column Val; the option @y following the month means that the parameter includes the year data, and a 6-digit number will be returned.

16.2 Calculate time repeat interval

There is a registration table, which records the entering and leaving time of customers in a certain bathroom on a certain day:

	A	B	C	D	E	F	G
1	**Name**	**Enter**	**Leave**	**18:00-19:00**	**19:00-20:00**	**20:00-21:00**	**21:00-22:00**
2	Mike	18:05	19:40				
3	Joan	19:00	20:50				
4	Kate	18:36	21:28				
5	Tim	18:15	21:00				
6	Jack	19:08	21:10				
7	Lorry	19:32	20:55				
8	Jim	19:45	21:40				
9	John	20:00	21:36				
10	Tom	20:08	22:00				
11	Rose	20:15	21:00				
12	Jordan	20:20	21:42				
13	Lily	20:27	21:38				
14	Susan	20:30	21:58				

The task is to calculate the number of minutes each customer stays in each time period, as shown in the following figure:

	A	B	C	D	E	F	G
1	Name	Enter	Leave	18:00-19:00	19:00-20:00	20:00-21:00	21:00-22:00
2	Mike	18:05	19:40	55	40		
3	Joan	19:00	20:50		60	50	
4	Kate	18:36	21:28	24	60	60	28
5	Tim	18:15	21:00	45	60	60	
6	Jack	19:08	21:10		52	60	10
7	Lorry	19:32	20:55		28	55	
8	Jim	19:45	21:40		15	60	40
9	John	20:00	21:36			60	36
10	Tom	20:08	22:00			52	60
11	Rose	20:15	21:00			45	
12	Jordan	20:20	21:42			40	42
13	Lily	20:27	21:38			33	38
14	Susan	20:30	21:58			30	58

Enter in cell D2:

	A
1	='D$1'.split("-").(interval@s(time("00:00","HH:mm"),time(~,"HH:mm"))/60)
2	=min('$C2'*1440,A1(2))-max('$B2'*1440,A1(1))
3	=if(A2>0,A2,null)

A1: Split D1 into two values by -, and then convert them to the number of minutes from 00:00

A2: The time stored in this Excel table is obtained by dividing the seconds from 00:00 to the present by 86400, so multiplying the time by 1440 is the number of minutes from 00:00 to the present. Calculate the larger value of customer's entering time and the start time, and the smaller value of customer's leaving time and the end time, and then calculate the time difference (minutes) between the two values.

Then drag D2 to every relevant row and column.

16.3 Generate a time sequence with the same time interval – one day

Script: =spl("=periods(date(""2020-01-01"",""yyyy-MM-dd""),date(""2020-01-10"",""yyyy-MM-dd""),1)")

Note: for the function periods(s,e,i), s represents the start date, e represents the end date, and i represents the time interval. The default unit for time interval is day. The option @m means the time interval unit is month, @y means year, @q means quarter, @t means ten days, and @s means second.

2020-01-01
2020-01-02
2020-01-03
2020-01-04
2020-01-05
2020-01-06
2020-01-07
2020-01-08
2020-01-09
2020-01-10

16.4 Generate a time sequence with the same time interval – two days

Script: =spl("=periods(date(""2020-01-01"",""yyyy-MM-dd""),date(""2020-01-10"",""yyyy-MM-dd""),2)")

Note: when the third parameter of periods is set to 2, it means two interval units.

2020-01-01
2020-01-03
2020-01-05
2020-01-07
2020-01-09
2020-01-10

16.5 Generate a time sequence with the same time interval – two hours

Script: =spl("=periods@s(""08:00:00"",""20:00:00"",7200)")

Note: The option @s means the time interval unit is second.

	A	B
1	8:00:00	
2	10:00:00	
3	12:00:00	
4	14:00:00	
5	16:00:00	
6	18:00:00	
7	20:00:00	

16.6 Generate a time sequence with the same time interval – one month

Script: =spl("=periods@mox(""2018-03-31"",""2019-03-01"",1)")

Note: this code is to generate a date sequence with an interval of 1 month between 2018-03-31 and 2019-03-01; the option @m means the time interval unit is month; x means the back endpoint (2019-03-01) is not included; and o means that you do not need to adjust the time to the starting point of the time unit, if this option was omitted, the time sequence would be adjusted to the 1st day of each month from the second time point.

	A
1	2018-3-31
2	2018-4-30
3	2018-5-31
4	2018-6-30
5	2018-7-31
6	2018-8-31
7	2018-9-30
8	2018-10-31
9	2018-11-30
10	2018-12-31
11	2019-1-31
12	2019-2-28

16.7 Generate a time sequence with the same time interval - Sunday

To generate a sequence of Sundays from 2020-02-01 to 2020-04-30, like this:

2020-02-02
2020-02-09
2020-02-16
2020-02-23
2020-03-01
2020-03-08
2020-03-15
2020-03-22
2020-03-29
2020-04-05
2020-04-12
2020-04-19
2020-04-26

Script:

	A
1	=date("2020-02-01")
2	=pdate@w(A1)
3	=if(A2<A1,A2+7,A2)
4	=periods@x(A3,"2020-04-30",7)

A1: The start date 2020-02-01

A2: Find the Sunday of the week of the start date, the option @w means to get the Sunday of the week of the specified date

A3: If A2 is earlier than the start date, let A3 be A2+7, i.e., the next Sunday, otherwise let A3 be A2

A4: Generate a date sequence with a 7-day interval between the first Sunday (A3) and the end date (2020-04-30), that is, a sequence of consecutive Sundays within the time period of specifying the start and end times. The option @x means that the backend date 2020-04-30 is not included.

16.8 The first Friday of a certain month/quarter/year

Script:

	A
1	>n=5
2	=pdate@w(pdate@m(date("2021-05-01"))+6-n)+n
3	=pdate@w(pdate@q(date("2021-05-01"))+6-n)+n
4	=pdate@w(pdate@y(date("2021-05-01"))+6-n)+n

A1: n represents the day of the week, and Sunday is 0, followed by +1

A2: Use the pdate@m to find the first day of the month, then add 6-n days to this day, and find the Sunday of the week of the date, and finally add n days to obtain result: 2021-05-07

A3: Use the pdate@q to find the first day of the quarter, and then add 6-n days to this day, and find the Sunday of the week of the date, and finally add n days to obtain result: 2021-04-02

A4: Use the pdate@y to find the first day of the year, and then add 6-n days to this day, and find the Sunday of the week of the date, and finally add n days to obtain result: 2021-01-01