

有序



目录

CONTENTS

1

有序集合

2

有序循环

3

迭代函数

4

定位计算

5

排序与对齐

01

有序集合

序列	要求	表达式	结果
[2,3,5,10,15,22,4,56]	取第三个成员	A1(3)	5
	取第1和第4个成员	A1([1,4])	[2,10]
	取前3个成员	A1(to(3))	[2,3,5]
	取第二个成员	A1.m(2)	3
	取第2到第4个成员	A1.m(2:4)	[3,5,10]
	取最后一个成员	A1.m(-1)	56
	取前两个, 第5个, 第7个到最后的成员	A1.m(:2,5,7:)	[2,3,15,4,56]

序表数据

Index	EID	NAME	GENDER	SALARY
1	1	Rebecca	F	7000
2	2	Ashley	F	11000
3	3	Rachel	F	9000
4	4	Emily	F	7000
5	5	Ashley	F	16000
6	6	Matthew	M	11000
7	7	Alexis	F	9000
8	8	Megan	F	11000
9	9	Victoria	F	3000
10	10	Ryan	M	13000

要求	表达式
取第三个成员	A1(3)
取第1和第4个成员	A1([1,4])
取前3个成员	A1(to(3))
取第二个成员	A1.m(2)
取第2到第4个成员	A1.m(2:4)
取最后一个成员	A1.m(-1)
取前两个, 第5个, 第7个到最后的成员	A1.m(:2,5,7:)

The diagram illustrates the results of various list operations on the original list A1. Arrows point from the expressions in the requirements table to the corresponding sub-tables:

- A1(3)**: Points to a sub-table with one row: EID 3, NAME Rachel, GENDER F, SALARY 9000.
- A1([1,4])**: Points to a sub-table with two rows: (1, 1, Rebecca, F, 7000) and (2, 4, Emily, F, 7000).
- A1(to(3))**: Points to a sub-table with three rows: (1, 1, Rebecca, F, 7000), (2, 2, Ashley, F, 11000), and (3, 3, Rachel, F, 9000).
- A1.m(2)**: Points to a sub-table with one row: EID 2, NAME Ashley, GENDER F, SALARY 11000.
- A1.m(2:4)**: Points to a sub-table with three rows: (1, 2, Ashley, F, 11000), (2, 3, Rachel, F, 9000), and (3, 4, Emily, F, 7000).
- A1.m(-1)**: Points to a sub-table with one row: EID 500, NAME Joseph, GENDER M, SALARY 12000.
- A1.m(:2,5,7:)**: Points to a sub-table with ten rows: (1, 1, Rebecca, F, 7000), (2, 2, Ashley, F, 11000), (3, 5, Ashley, F, 16000), (4, 7, Alexis, F, 9000), (5, 8, Megan, F, 11000), (6, 9, Victoria, F, 3000), (7, 10, Ryan, M, 13000), (8, 11, Jacob, M, 12000), (9, 12, Jessica, F, 7000), and (10, 13, Daniel, M, 10000).

	A	B
1	<code>=file("E:/txt/students_score.txt").import@t()</code>	
2	<code>=A1.(English).sort()</code>	/英语成绩排序
3	<code>=A2.len()+1</code>	/集合成员数量
4	<code>=A2([A3\2,(A3+1)\2]).avg()</code>	/偶数取中间两个的均值, 奇数取中间数

A1~A4的结果

Index	Name	Math	Chinese	English
1	Natalie	84	90	84
2	Jessica	87	88	78
3	Brianna	89	90	75
4	Emma	88	84	94
5	Zachary	75	81	85
6	Sophia	74	86	93
7	Hannah	90	76	95
8	Christopher	71	81	86
9	Sean	98	86	81
10	Tyler	87	78	93

Index	Member
1	75
2	78
3	81
4	84
5	85
6	86
7	93
8	93
9	94
10	95

Value
11

Value
85.5

$A.pos(x)$, 其中 x 是序列, 获取 x 成员在 A 中的位置

Index	Member
1	16
2	22
3	5
4	6
5	2
6	7

$x=[22,6,2]$

$A.pos(x)$



2
4
5

$A.pos(x)$, 相当于 $A(p)=x$ 的逆运算。

查询 "Tyler" 和 "Sean" 的总成绩名次

	A	B
1	<code>=file("E:/txt/students_score.txt").import@t()</code>	
2	<code>=A1.sort(-(Chinese+Math+English))</code>	/按照总成绩从大到小排序
3	<code>=A2.(Name).pos(["Tyler","Sean"])</code>	/查询 "Tyler" 和 "Sean" 的总成绩名次

A1~A3结果

Index	Name	Math	Chinese	English
1	Natalie	84	90	84
2	Jessica	87	88	78
3	Brianna	89	90	75
4	Emma	88	84	94
5	Zachary	75	81	85
6	Sophia	74	86	93
7	Hannah	90	76	95
8	Christopher	71	81	86
9	Sean	98	86	81
10	Tyler	87	78	93

Index	Name	Math	Chinese	English
1	Emma	88	84	94
2	Sean	98	86	81
3	Hannah	90	76	95
4	Natalie	84	90	84
5	Tyler	87	78	93
6	Brianna	89	90	75
7	Jessica	87	88	78
8	Sophia	74	86	93
9	Zachary	75	81	85
10	Christopher	71	81	86

Index	Member
1	5
2	2

A	
1	=file("E:/txt/EMPLOYEE_nan.txt").import@t()
2	=A1.select(~.array().pos(null)>0)

结果

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	1.0	Rebecca	Moore	F	California	1974-11-20	2005-03-11	R&D	7000.0
2	2.0	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000.0
3	3.0	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000.0
4	4.0	Emily	Smith	F	Texas	1985-03-07	2006-08-15	HR	7000.0
5	5.0	Ashley	Smith	F	Texas	1975-05-13	2004-07-30	R&D	16000.0
6	6.0	Matthew	Johnson	M	California	1984-07-07	2005-07-07	Sales	11000.0
7	7.0	Alexis	Smith	F	Illinois	1972-08-16	2002-08-16	Sales	9000.0
8	8.0	Megan	Wilson	F	California	1979-04-19	1984-04-19	Marketing	11000.0
9	9.0	Victoria	Davis	F	Texas	1983-12-07	2009-12-07	HR	3000.0
10	10.0	Ryan	Johnson	M	Pennsylva...	1976-03-12	2006-03-12	R&D	13000.0

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	16.0	Christopher	(null)	M	Florida	1979-06-27	2007-06-27	Production	9000.0
2	17.0	Hannah	Johnson	F	Texas	(null)	2006-07-19	Marketing	4000.0
3	(null)	Joseph	(null)	M	California	1983-08-27	2003-08-27	Finance	6000.0
4	27.0	Alexis	Jones	F	California	1983-12-27	(null)	Marketing	10000.0
5	(null)	Olivia	Harris	F	California	1979-08-27	2009-08-27	Sales	8000.0
6	35.0	Justin	Smith	M	Texas	1978-08-20	(null)	R&D	7000.0
7	41.0	Emily	Davis	F	Illinois	1984-07-07	(null)	Production	9000.0
8	43.0	Joshua	Williams	M	California	1980-11-25	2000-11-25	(null)	12000.0
9	55.0	Olivia	Anderson	F	(null)	1971-08-27	2001-08-27	Technology	13000.0
10	65.0	Michael	Smith	(null)	Connecticut	1971-03-03	2004-03-01	Sales	8000.0

员工的部分数据表

Index	EID	NAME	GENDER	DEPT	SALARY
1	1	Rebecca	F	R&D	7000
2	2	Ashley	F	Finance	11000
3	3	Rachel	F	Sales	9000
4	4	Emily	F	HR	7000
5	5	Ashley	F	R&D	16000
6	6	Matthew	M	Sales	11000
7	7	Alexis	F	Sales	9000
8	8	Megan	F	Marketing	11000
9	9	Victoria	F	HR	3000
10	10	Ryan	M	R&D	13000

A.select(GENDER == "M")

Index	EID	NAME	GENDER	DEPT	SALARY
1	6	Matthew	M	Sales	11000
2	10	Ryan	M	R&D	13000
3	11	Jacob	M	Sales	12000
4	13	Daniel	M	Finance	10000
5	16	Christopher	M	Production	9000

A.select(DEPT == "Sales")

Index	EID	NAME	GENDER	DEPT	SALARY
1	3	Rachel	F	Sales	9000
2	6	Matthew	M	Sales	11000
3	7	Alexis	F	Sales	9000
4	11	Jacob	M	Sales	12000
5	12	Jessica	F	Sales	7000

A2&A3

Index	EID	NAME	GENDER	DEPT	SALARY
1	6	Matthew	M	Sales	11000
2	10	Ryan	M	R&D	13000
3	11	Jacob	M	Sales	12000
4	13	Daniel	M	Finance	10000
5	16	Christopher	M	Production	9000

A.select(GENDER == M || DEPT == "Sales")

Index	EID	NAME	GENDER	DEPT	SALARY
1	3	Rachel	F	Sales	9000
2	6	Matthew	M	Sales	11000
3	7	Alexis	F	Sales	9000
4	10	Ryan	M	R&D	13000
5	11	Jacob	M	Sales	12000

序列	计算	描述	结果
A1=[1,2,3] A2=[1,3,2] A3=[3,1,4,5]	A1==A2	判断序列是否相等	false
	A1^A3	A1与A3的交集	[1,3]
	A3^A1	A3与A1的交集	[3,1]
	A1&A3	A1与A3的并集	[1,2,3,4,5]
	A3&A1	A3与A1的并集	[3,1,4,5,2]

判断成员相同??

序列	计算	描述	结果
A1=[1,2,3] A2=[1,3,2] A3=[3,1,4,5]	A1.eq(A2)	判断成员是否相同	true
	A1.eq(A3)		false

A.eq(B)

true

序表

A2&A3

EID	NAME	GENDER	DEPT	SALARY
2	Matthew	M	Sales	11000
6	Ryan	M	R&D	13000
7	Jacob	M	Sales	12000
3	Alexis	F	Sales	9000

A.select(GENDER==M||DEPT=="Sales")

EID	NAME	GENDER	DEPT	SALARY
2	Matthew	M	Sales	11000
3	Alexis	F	Sales	9000
6	Ryan	M	R&D	13000
7	Jacob	M	Sales	12000

02

有序循环

员工数据

Index	EID	NAME	GENDER	SALARY
1	1	Rebecca	F	7000
2	2	Ashley	F	11000
3	3	Rachel	F	9000
4	4	Emily	F	7000
5	5	Ashley	F	16000
6	6	Matthew	M	11000
7	7	Alexis	F	9000
8	8	Megan	F	11000
9	9	Victoria	F	3000
10	10	Ryan	M	13000
11	11	Jacob	M	12000
12	12	Jessica	F	7000

取偶数位置的记录(A.select(%2==0))

Index	EID	NAME	GENDER	SALARY
1	2	Ashley	F	11000
2	4	Emily	F	7000
3	6	Matthew	M	11000
4	8	Megan	F	11000
5	10	Ryan	M	13000
6	12	Jessica	F	7000
7	14	Alyssa	F	4000
8	16	Christopher	M	9000
9	18	Jonathan	M	7000
10	20	Alexis	F	16000
11	22	Jacob	M	16000
12	24	Chloe	F	10000

序列	函数	表达式	结果	描述
[2,3,5,6]	A.(~)	~	[2,3,5,6]	返回原序列
	A.(~[-1])	~[-1]	[null,2,3,5]	返回和当前成员距离-1的成员
	A.(~[2])	~[2]	[5,6,null,null]	返回和当前成员距离2的成员

函数	表达式	描述
A.(prod)	prod	prod字段
A.(prod[-1])	prod[-1]	和prod字段成员距离-1的成员
A.(prod[:])	prod[:]	包含prod所有成员的序列组成的序列
A.(prod[:0])	prod[:0]	包含prod从开始到当前位置的成员组成的序列

数据

Index	prod	month	sales
1	1	1	2018.2
2	2	1	1814.8
3	3	1	400.0
4	4	1	0.0
5	5	1	0.0
6	6	1	0.0
7	7	1	0.0
8	8	1	0.0
9	9	1	1897.6
10	10	1	0.0

Index	Member
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Index	Member
1	(null)
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9

Index	Member
1	[1,2,3, ...]
2	[1,2,3, ...]
3	[1,2,3, ...]
4	[1,2,3, ...]
5	[1,2,3, ...]
6	[1,2,3, ...]
7	[1,2,3, ...]
8	[1,2,3, ...]
9	[1,2,3, ...]
10	[1,2,3, ...]

序号	成员
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Index	Member
1	[1]
2	[1,2]
3	[1,2,3]
4	[1,2,3, ...]
5	[1,2,3, ...]
6	[1,2,3, ...]
7	[1,2,3, ...]
8	[1,2,3, ...]
9	[1,2,3, ...]
10	[1,2,3, ...]

序	成员
1	1
2	2
3	3

找出1001股价连续上涨的最大天数

有序循环

	A
1	=file("E:/txt/stock1001_price.txt").import@t()
2	=a=0,A1.max(a=if(CL>CL[-1],a+1,0))

A1、A2结果

Index	stockid	DT	CL
1	1001	2009-01-01	4.0
2	1001	2009-01-02	3.64
3	1001	2009-01-05	3.95
4	1001	2009-01-06	3.68
5	1001	2009-01-07	3.53
6	1001	2009-01-08	3.59
7	1001	2009-01-09	3.9
8	1001	2009-01-12	3.56
9	1001	2009-01-13	3.22
10	1001	2009-01-14	3.17

Value
3

计算1001股价连续5个交易日的平均值

有序循环

A	
1	<code>=file("E:/txt/stock1001_price.txt").import@t()</code>
2	<code>=A1.derive(CL[-5:0].avg():5_avg)</code>

A1、A2结果

Index	stockid	DT	CL
1	1001	2009-01-01	4.0
2	1001	2009-01-02	3.64
3	1001	2009-01-05	3.95
4	1001	2009-01-06	3.68
5	1001	2009-01-07	3.53
6	1001	2009-01-08	3.59
7	1001	2009-01-09	3.9
8	1001	2009-01-12	3.56
9	1001	2009-01-13	3.22
10	1001	2009-01-14	3.17

Index	stockid	DT	CL	5_avg
1	1001	2009-01-01	4.0	4.0
2	1001	2009-01-02	3.64	3.8200000...
3	1001	2009-01-05	3.95	3.8633333...
4	1001	2009-01-06	3.68	3.8175
5	1001	2009-01-07	3.53	3.7600000...
6	1001	2009-01-08	3.59	3.7316666...
7	1001	2009-01-09	3.9	3.715
8	1001	2009-01-12	3.56	3.7016666...
9	1001	2009-01-13	3.22	3.58
10	1001	2009-01-14	3.17	3.4949999...

	A
1	=file("E:/txt/prod1.txt").import@t()
2	=A1.derive(sum(sales[:0]):add_up)

A1、A2结果

Index	prod	month	sales
1	1	1	2018.2
2	1	2	0.0
3	1	3	672.0
4	1	4	2657.8
5	1	5	1344.0
6	1	6	838.45
7	1	7	1140.0
8	1	8	450.0
9	1	9	0.0
10	1	10	630.0
11	1	11	1850.0
12	1	12	0.0

Index	prod	month	sales	add_up
1	1	1	2018.2	2018.2
2	1	2	0.0	2018.2
3	1	3	672.0	2690.2
4	1	4	2657.8	5348.0
5	1	5	1344.0	6692.0
6	1	6	838.45	7530.45
7	1	7	1140.0	8670.45
8	1	8	450.0	9120.45
9	1	9	0.0	9120.45
10	1	10	630.0	9750.45
11	1	11	1850.0	11600.45
12	1	12	0.0	11600.45

找出各个产品比上月多10%的销售额的记录

有序循环

	A	B
1	=file("E:/txt/prod.txt").import@t()	
2	=A1.sort(prod,month)	/按照prod和month排序
3	=A2.select(if(prod==prod[-1],sales/sales[-1]>1.1))	/找到结果

A1~A3
结果

Index	prod	month	sales
1	1	1	2018.2
2	2	1	1814.8
3	3	1	400.0
4	4	1	0.0
5	5	1	0.0
6	6	1	0.0
7	7	1	0.0
8	8	1	0.0
9	9	1	1897.6
10	10	1	0.0

Index	prod	month	sales
1	1	1	2018.2
2	1	2	0.0
3	1	3	672.0
4	1	4	2657.8
5	1	5	1344.0
6	1	6	838.45
7	1	7	1140.0
8	1	8	450.0
9	1	9	0.0
10	1	10	630.0

Index	prod	month	sales
1	1	3	672.0
2	1	4	2657.8
3	1	7	1140.0
4	1	10	630.0
5	1	11	1850.0
6	2	3	1125.5
7	2	4	1388.5
8	2	6	0.0
9	2	7	0.0
10	2	8	605.0

计算各个产品销售额三天内的平均值

有序循环

A	
1	=file("E:/txt/prod.txt").import@t()
2	=A1.sort(prod,month)
3	=A2.derive(if(prod==prod[-1]&&prod==prod[1],sales[-1:1].avg()):moving_avg)

A1~A3
结果

Index	prod	month	sales
1	1	1	2018.2
2	2	1	1814.8
3	3	1	400.0
4	4	1	0.0
5	5	1	0.0
6	6	1	0.0
7	7	1	0.0
8	8	1	0.0
9	9	1	1897.6
10	10	1	0.0

Index	prod	month	sales
1	1	1	2018.2
2	1	2	0.0
3	1	3	672.0
4	1	4	2657.8
5	1	5	1344.0
6	1	6	838.45
7	1	7	1140.0
8	1	8	450.0
9	1	9	0.0
10	1	10	630.0

Index	prod	month	sales	moving_avg
1	1	1	2018.2	(null)
2	1	2	0.0	896.733333...
3	1	3	672.0	1109.93333...
4	1	4	2657.8	1557.93333...
5	1	5	1344.0	1613.41666...
6	1	6	838.45	1107.48333...
7	1	7	1140.0	809.483333...
8	1	8	450.0	530.0
9	1	9	0.0	360.0
10	1	10	630.0	826.666666...

A	
1	=file("E:/txt/prod.txt").import@t()
2	=A1.sort(prod,month)
3	=A2.derive((x=if(prod!=prod[-1],#,x),sum(sales[x-#:0])):add_up)

A1~A3结果

Index	prod	month	sales
1	1	1	2018.2
2	2	1	1814.8
3	3	1	400.0
4	4	1	0.0
5	5	1	0.0
6	6	1	0.0
7	7	1	0.0
8	8	1	0.0
9	9	1	1897.6
10	10	1	0.0
11	11	1	1194.0
12	12	1	0.0
13	13	1	0.0
14	14	1	319.2
15	15	1	0.0
16	16	1	1005.9
17	17	1	102.4
18	18	1	0.0

Index	prod	month	sales
1	1	1	2018.2
2	1	2	0.0
3	1	3	672.0
4	1	4	2657.8
5	1	5	1344.0
6	1	6	838.45
7	1	7	1140.0
8	1	8	450.0
9	1	9	0.0
10	1	10	630.0
11	1	11	1850.0
12	1	12	0.0
13	2	1	1814.8
14	2	2	631.6
15	2	3	1125.5
16	2	4	1388.5
17	2	5	0.0
18	2	6	0.0

Index	prod	month	sales	add_up
1	1	1	2018.2	2018.2
2	1	2	0.0	2018.2
3	1	3	672.0	2690.2
4	1	4	2657.8	5348.0
5	1	5	1344.0	6692.0
6	1	6	838.45	7530.45
7	1	7	1140.0	8670.45
8	1	8	450.0	9120.45
9	1	9	0.0	9120.45
10	1	10	630.0	9750.45
11	1	11	1850.0	11600.45
12	1	12	0.0	11600.45
13	2	1	1814.8	1814.8
14	2	2	631.6	2446.4
15	2	3	1125.5	3571.9
16	2	4	1388.5	4960.4
17	2	5	0.0	4960.4
18	2	6	0.0	4960.4

03

迭代函数

SUM: 先设置一个初始值 0，然后遍历集合的每个成员，每次将成员值加到初始值上，直到成员被遍历完。

MAX: 设置初始值为无穷小，遍历集合成员，每碰比初始值更大的成员值则替换初值，直到遍历完。

MIN: 和 MAX 一样，只是初始值和比较方向是反的。

`a`=初始值, `~`=当前成员, `~~`=当前初始值
这样就可以完成完成这些运算。

`sum()`的迭代过程:

3
5
6
2
1
9
8

	<code>~</code>	<code>~~</code>
初始化		0
第1步	3	3
第2步	5	8
第3步	6	14
第 <i>i</i> 步	<code>A(i)</code>	<code>~~+A(i)</code>
第7步	8	34



结果: 34

我们用iterate函数来实现这些迭代函数

	A	B
1	[2,4,6,11]	
2	=A1.sum()	=A1.iterate(~~=~~+~,0)
3	=A1.min()	=A1.iterate(if(~<~~,~,~~),inf())
4	=A1.max()	=A1.iterate(if(~>~~,~,~~),-inf())
5	=demo.query("select * from EMPLOYEE")	
6	=A5.maxp@a(EID)	=A5.iterate(if(!~ ~.EID>~~.EID,[~],if(~.EID==~~.EID,~~ ~,~~)),null)
7	=A5.maxp@a(SALARY)	=A5.iterate(if(!~ ~.SALARY>~~.SALARY,[~],if(~.SALARY==~~.SALARY,~~ ~,~~)),null)

同行结果相同，2,3,4,6,7行结果如下

Value
23

Value
2

Value
11

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	500	Joseph	Smith	M	Pennsylva...	1972-06-04	2003-12-01	Production	12000

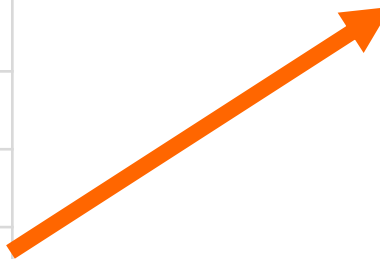
Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	5	Ashley	Smith	F	Texas	1975-05-13	2004-07-30	R&D	16000
2	20	Alexis	Allen	F	Florida	1977-08-07	2007-08-07	Administrat...	16000
3	22	Jacob	Davis	M	Texas	1985-05-07	2001-05-07	R&D	16000
4	47	Elizabeth	Brown	F	Pennsylva...	1971-08-27	2001-08-27	Marketing	16000

	A
1	=10.iterate([~~(2),(~~(1)+~~(2))],[1,1])

	~	~~
初始化		[1,1]
第1步	1	[1,2]
第2步	2	[2,3]
第3步	3	[3,5]
第i步	i	[~~(2),~~(1)+~~(2)]
第10步	10	[89,144]

结果

Index	Member
1	89
2	144



二分法求 $\ln x + 2x - 6 = 0$ 的近似解

	A	B
1	func	return $\ln(A1) + 2 * A1 - 6$
	>y=1	1e-10
2	=1000.iterate((x=(~~(1)+~~(2))/2,y=func(A1,x),if(y>0,[~~(1),x],[x,~~(2)])),[0,3],abs(y)<B2)	
3	=x	
4	=func(A1,x)	

A2的迭代过程:

	~~	$x=(~~(1)+~~(2))/2$	$y=\ln(x)+2*x-6$	abs(y)
初始化	[0,3]			
第1步	[0,3]	1.5	-2.5945	2.5945
第2步	[1.5,3]	2.25	-0.6891	0.6891
第3步	[2.25,3]	2.625	0.2151	0.2151
第4步	[2.25,2.625]	2.4375	-0.234	0.234
第i步	~~	$(~~(1)+~~(2))/2$	$\ln(x)+2*x-6$	abs(y)
第n步	~~	$(~~(1)+~~(2))/2$	$\ln(x)+2*x-6$	if abs(y)<1e-10 break

Value
2.534919132012874

Value
-2.6577851031106547E-11

计算某个销售员的累计销售额

迭代函数

	A	B
1	=demo.query("select ORDERID,SELLERID,AMOUNT,ORDERDATE from SALES ")	
2	=A1.select(SELLERID==10).sort(ORDERDATE)	/筛选10号销售员并按照ORDERDATE排序
3	=A2.derive(iterate(~~+AMOUNT,0):cum_sum)	/使用迭代函数计算累计值

A1

Index	ORDERID	SELLERID	AMOUNT	ORDERDATE
1	1	17	392.0	2012-11-02 ...
2	2	6	4802.0	2012-11-09 ...
3	3	16	13500.0	2012-11-05 ...
4	4	9	26100.0	2012-11-08 ...
5	5	11	4410.0	2012-11-12 ...
6	6	18	6174.0	2012-11-07 ...
7	7	2	17800.0	2012-11-06 ...
8	8	7	2156.0	2012-11-09 ...
9	9	14	17400.0	2012-11-12 ...
10	10	19	19200.0	2012-11-12 ...

A2

Index	ORDERID	SELLERID	AMOUNT	ORDERDA...
1	33	10	21800.0	2012-12-03...
2	29	10	29500.0	2012-12-05...
3	76	10	20700.0	2013-01-20...
4	99	10	11800.0	2013-02-11...
5	110	10	11000.0	2013-02-24...
6	142	10	8232.0	2013-03-28...
7	156	10	7448.0	2013-04-08...
8	182	10	23700.0	2013-05-05...
9	259	10	25600.0	2013-07-18...
10	271	10	4900.0	2013-07-29...

A3

Index	ORDERID	SELLERID	AMOUNT	ORDERDA...	cum_sum
1	33	10	21800.0	2012-12-0...	21800.0
2	29	10	29500.0	2012-12-0...	51300.0
3	76	10	20700.0	2013-01-2...	72000.0
4	99	10	11800.0	2013-02-1...	83800.0
5	110	10	11000.0	2013-02-2...	94800.0
6	142	10	8232.0	2013-03-2...	103032.0
7	156	10	7448.0	2013-04-0...	110480.0
8	182	10	23700.0	2013-05-0...	134180.0
9	259	10	25600.0	2013-07-1...	159780.0
10	271	10	4900.0	2013-07-2...	164680.0

结果

计算所有销售员的累计销售额

	A	B
1	=demo.query("select CLIENT,SELLERID,AMOUNT,ORDERDATE from SALES ")	
2	=A1.sort(SELLERID,ORDERDATE)	/按照SELLERID和订单日期排序
3	=A2.derive(iterate(~~=~~+AMOUNT,0;SELLERID):cum_sum)	
4	=A2.derive(cum(AMOUNT;SELLERID):cum_sum)	
	/A3: 使用迭代函数计算累计值; A4: cum(x;G)当G变化时, x重新开始计算。	

结果

A1

Index	CLIENT	SELLERID	AMOUNT	ORDERDATE
1	UJRNP	17	392.0	2012-11-02 1...
2	SJCH	6	4802.0	2012-11-09 1...
3	UJRNP	16	13500.0	2012-11-05 1...
4	PWQ	9	26100.0	2012-11-08 1...
5	PWQ	11	4410.0	2012-11-12 1...
6	HANAR	18	6174.0	2012-11-07 1...
7	EGU	2	17800.0	2012-11-06 1...
8	VILJX	7	2156.0	2012-11-09 1...
9	JAYB	14	17400.0	2012-11-12 1...
10	JAXE	19	19200.0	2012-11-12 1...

A2

Index	CLIENT	SELLERID	AMOUNT	ORDERDATE
1	JAYB	1	7644.0	2012-11-16 1...
2	HANAR	1	13200.0	2013-01-17 1...
3	YZ	1	11600.0	2013-01-20 1...
4	AVU	1	21800.0	2013-02-05 1...
5	HL	1	26400.0	2013-02-18 1...
6	PWQ	1	17500.0	2013-02-21 1...
7	FHYBR	1	16000.0	2013-03-03 1...
8	HP	1	13600.0	2013-03-15 1...
9	DNEDL	1	26100.0	2013-05-13 1...
10	EGU	1	14000.0	2013-05-20 1...

A3、A4

Index	CLIENT	SELLERID	AMOUNT	ORDERDATE	cum_sum
1	JAYB	1	7644.0	2012-11-16...	7644.0
2	HANAR	1	13200.0	2013-01-17...	20844.0
3	YZ	1	11600.0	2013-01-20...	32444.0
4	AVU	1	21800.0	2013-02-05...	54244.0
5	HL	1	26400.0	2013-02-18...	80644.0
6	PWQ	1	17500.0	2013-02-21...	98144.0
7	FHYBR	1	16000.0	2013-03-03...	114144.0
8	HP	1	13600.0	2013-03-15...	127744.0
9	DNEDL	1	26100.0	2013-05-13...	153844.0
10	EGU	1	14000.0	2013-05-20...	167844.0

	A	B
1	<code>=file("E:/txt/students_c.txt").import@t()</code>	
2	<code>=A1.select(CLASS==1).sort(-SCORE)</code>	/按照SCORE逆序排序
3	<code>=A2.derive(iterate((x=x+1,if(SCORE==SCORE[-1],~~,x)),(x=0)):RANK)</code>	
	/初始化x=0, 迭代一次x+1, 当SCORE相同时, 排名不变, 不同时排名为x。	

结果

A1

Index	CLASS	NAME	SCORE
1	1	Adams Bro...	607
2	1	Adams Ha...	620
3	1	Adams Jon...	611
4	1	Allen Ashley	609
5	1	Allen Bran...	612
6	1	Baker Dani...	620
7	1	Brown Am...	616
8	1	Brown Lau...	618
9	1	Carter Mich...	621
10	1	Clark Justin	615

A2

Index	CLASS	NAME	SCORE
1	1	Smith Willi...	629
2	1	Garcia Bryan	628
3	1	Jones Justin	628
4	1	Lee Rachel	628
5	1	Moore Mich...	628
6	1	Smith Reb...	628
7	1	Lewis Gabr...	627
8	1	Martin Jose...	627
9	1	Moore Jon...	627
10	1	Williams N...	627

A3

Index	CLASS	NAME	SCORE	RANK
1	1	Smith Willi...	629	1
2	1	Garcia Bryan	628	2
3	1	Jones Justin	628	2
4	1	Lee Rachel	628	2
5	1	Moore Mich...	628	2
6	1	Smith Reb...	628	2
7	1	Lewis Gabr...	627	7
8	1	Martin Jos...	627	7
9	1	Moore Jon...	627	7
10	1	Williams N...	627	7

	A	B
1	<code>=file("E:/txt/students_c.txt").import@t()</code>	
2	<code>=A1.sort(CLASS,-SCORE)</code>	/按照CLASS和SCORE逆序排序
3	<code>=A2.derive(iterate((x=x+1,if(SCORE==SCORE[-1],~~,x)),(x=0);CLASS):RANK)</code>	
4	<code>=A2.derive(rank(SCORE;CLASS):RANK)</code>	
	<p>/A3和A4的结果相同, A3:初始化x=0, 迭代一次x+1, 当SCORE相同时, 排名不变, 不同时排名为x。 , 当CLASS变化时, x重新初始化 A4:SCORE相同时, 排名不变, 不同时排名等于当前排名+上个排名数量, CLASS变化时, 从1开始排。</p>	

结果

A1

Index	CLASS	NAME	SCORE
1	1	Adams Bro...	607
2	1	Adams Ha...	620
3	1	Adams Jon...	611
4	1	Allen Ashley	609
5	1	Allen Bran...	612
6	1	Baker Dani...	620
7	1	Brown Am...	616
8	1	Brown Lau...	618
9	1	Carter Mich...	621
10	1	Clark Justin	615

A2

Index	CLASS	NAME	SCORE
1	1	Smith Willi...	629
2	1	Garcia Bryan	628
3	1	Jones Justin	628
4	1	Lee Rachel	628
5	1	Moore Mich...	628
6	1	Smith Reb...	628
7	1	Lewis Gab...	627
8	1	Martin Jos...	627
9	1	Moore Jon...	627
10	1	Williams N...	627

A3、A4

Index	CLASS	NAME	SCORE	RANK
1	1	Smith Willi...	629	1
2	1	Garcia Bryan	628	2
3	1	Jones Justin	628	2
4	1	Lee Rachel	628	2
5	1	Moore Mich...	628	2
6	1	Smith Reb...	628	2
7	1	Lewis Gab...	627	7
8	1	Martin Jos...	627	7
9	1	Moore Jon...	627	7
10	1	Williams N...	627	7

04

定位计算

序号	序列	表达式	判断	选项	结果
1	7		False		
2	15		True	无	2
3	8		False		
4	9	~ > 10	False		
5	23		True	@a	[2,5,7]
6	5		False		
7	22		True		

A.pselec(GENDER:" M" ,DEPT:" Sales")

=A.pselect(GENDER==" M" &&DEPT==" Sales")

数据

Index	EID	NAME	GENDER	SALARY
1	1	Rebecca	F	7000
2	2	Ashley	F	11000
3	3	Rachel	F	9000
4	4	Emily	F	7000
5	5	Ashley	F	16000
6	6	Matthew	M	11000
7	7	Alexis	F	9000
8	8	Megan	F	11000
9	9	Victoria	F	3000
10	10	Ryan	M	13000

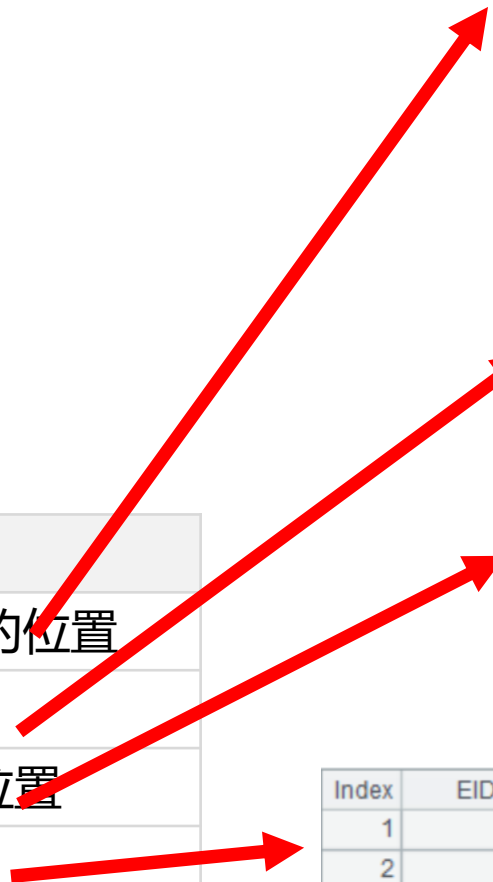
Index	Member
1	11
2	12
3	13
4	14
5	15
6	16
7	17
8	18
9	19
10	20

Value
500

Index	Member
1	9
2	39
3	45

函数	描述
A.pselect@a(EID>10&&EID<=20)	取EID大于10小于等于20的位置
A.pmax(EID)	获得EID的最大值的位置
A.pmin@a(SALARY)	获得SALARY的最小值的位置
A(A.pmin@a(SALARY))	得到最小值对应的记录

Index	EID	NAME	GENDER	SALARY
1	9	Victoria	F	3000
2	39	Andrew	M	3000
3	45	Kayla	F	3000



计算出某个月中股价超过100元的交易日的当日涨幅 (1号记录当日价格)

定位计算

	A	B
1	=file("E:/txt/stock_price.txt").import@t()	
2	=A1.sort(stockid)	/按stockid排序
3	=A2.pselect@a(CL>100)	/找出股价大于100的记录位置
4	=A2.calc(A3,if(day(DT)==1,~.CL,~.CL-~[-1].CL))	/利用位置计算涨幅

A1~A4结果

Index	stockid	DT	CL
1	1001	2009-01-01	4.0
2	1026	2009-01-01	2.13
3	1028	2009-01-01	20.09
4	1070	2009-01-01	14.95
5	1107	2009-01-01	3.74
6	1134	2009-01-01	10.68
7	1137	2009-01-01	42.31
8	1147	2009-01-01	19.61
9	1206	2009-01-01	14.01
10	1213	2009-01-01	40.94

Index	stockid	DT	CL
1	1001	2009-01-01	4.0
2	1001	2009-01-02	3.64
3	1001	2009-01-05	3.95
4	1001	2009-01-06	3.68
5	1001	2009-01-07	3.53
6	1001	2009-01-08	3.59
7	1001	2009-01-09	3.9
8	1001	2009-01-12	3.56
9	1001	2009-01-13	3.22
10	1001	2009-01-14	3.17

Index	Member
1	243
2	244
3	245
4	246
5	247
6	248
7	257
8	260
9	353
10	354

Index	Member
1	134.61
2	-10.620000000000019
3	-3.8900000000000006
4	-12.009999999999991
5	4.079999999999998
6	-9.600000000000009
7	9.409999999999997
8	4.670000000000002
9	231.91
10	-10.719999999999999

计算出1001股票某月首次最高价的当日增长率(1号记录当日股价)

定位计算

	A	B
1	=file("E:/txt/stock1001_price.txt").import@t()	
2	=A1.pmax(CL)	/找到最高价的位置
3	=A1.calc(A2,if(day(DT)==1,A1.CL,A1.CL/A1.CL[-1]-1))	/计算当日增长率

A1~A3结果

Index	stockid	DT	CL
1	1001	2009-01-01	4.0
2	1001	2009-01-02	3.64
3	1001	2009-01-05	3.95
4	1001	2009-01-06	3.68
5	1001	2009-01-07	3.53
6	1001	2009-01-08	3.59
7	1001	2009-01-09	3.9
8	1001	2009-01-12	3.56
9	1001	2009-01-13	3.22
10	1001	2009-01-14	3.17

Value
18

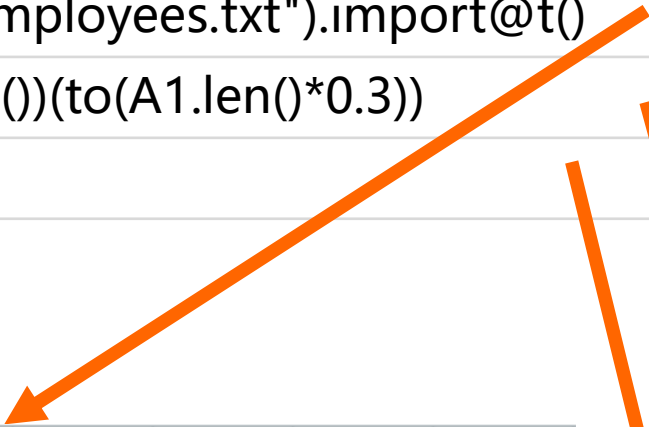
Value
0.09691629955947145

05


排序与对齐

随机的将数据分成30%和70%，作为训练集和测试集

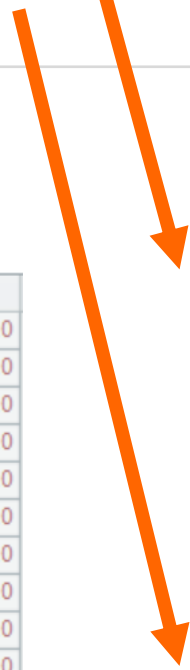
	A	B
1	=file("E:/txt/Employees.txt").import@t()	
2	=A1.sort(rand()(to(A1.len()*0.3)))	/将数据顺序打乱, 取30%
3	=A1\A2	/差集



Index	ID	Name	Gender	Post	Birthday	AccountNo	BasePay
1	1	Mike	Female	Sale	1968-12-0...	536936891...	5600.0
2	2	Jake	Male	Vice Presid...	1962-02-1...	964107677...	2500.0
3	3	Lucy	Female	Sale	1973-08-3...	665248245...	10800.0
4	4	Andy	Male	Sales Man...	1968-09-1...	650028860...	7500.0
5	5	Jim	Male	Sales Man...	1965-03-0...	441380247...	4700.0
6	6	David	Male	Sale	1967-07-0...	860916757...	9300.0
7	7	Jessica	Male	Sale	1960-05-2...	568899664...	2900.0
8	8	Lily	Female	Inside Sale...	1969-01-0...	428844667...	3500.0
9	9	Mary	Female	Sale	1969-07-0...	962447951...	7200.0
10	10	Tiger	Male	General M...	1970-10-1...	255490864...	4300.0



Index	ID	Name	Gender	Post	Birthday	AccountNo	BasePay
1	4	Andy	Male	Sales Man...	1968-09-1...	650028860...	7500.0
2	23	Larry	Female	R&D Leader	1980-07-2...	841933302...	6800.0
3	10	Tiger	Male	General M...	1970-10-1...	255490864...	4300.0
4	17	Frank	Male	Engineer	1983-10-2...	128284974...	8900.0
5	16	Ed	Male	R&D Leader	1983-02-1...	652548359...	5700.0
6	14	Charlie	Female	Engineer	1979-01-0...	394274763...	11000.0



Index	ID	Name	Gender	Post	Birthday	AccountNo	BasePay
1	1	Mike	Female	Sale	1968-12-0...	536936891...	5600.0
2	2	Jake	Male	Vice Presid...	1962-02-1...	964107677...	2500.0
3	3	Lucy	Female	Sale	1973-08-3...	665248245...	10800.0
4	5	Jim	Male	Sales Man...	1965-03-0...	441380247...	4700.0
5	6	David	Male	Sale	1967-07-0...	860916757...	9300.0
6	7	Jessica	Male	Sale	1960-05-2...	568899664...	2900.0
7	8	Lily	Female	Inside Sale...	1969-01-0...	428844667...	3500.0
8	9	Mary	Female	Sale	1969-07-0...	962447951...	7200.0
9	11	Kate	Female	Human Re...	1980-04-0...	683984106...	7600.0
10	12	Al	Male	Sale	1980-03-0...	739029848...	7500.0

序列A

序号	1	2	3	4	5	6	7
成员	7	15	8	9	23	5	22

序列B

序号	1	2	3	4	5	6	7
成员	7	15	8	9	23	5	22

序列C

序号	1	2	3	4	5	6	7
成员	7	15	8	9	23	5	22

sort

psort

sort是对序列成员排序返回序列成员;

psort是对序列成员排序返回成员位置序列;

所以A(C)=B

计算1001股票三个最高价的当日涨幅

	A	B
1	=file("E:/txt/stock1001_price.txt").import@t()	
2	=A1.psort@z(CL)	/逆序排序后返回序号序列
3	=A2(to(3))	/取前三个
4	=A1.calc(A3,if(day(DT))=1,CL,CL/CL[-1]-1))	/计算涨幅

A1~A4结果

Index	stockid	DT	CL
1	1001	2009-01-01	4.0
2	1001	2009-01-02	3.64
3	1001	2009-01-05	3.95
4	1001	2009-01-06	3.68
5	1001	2009-01-07	3.53
6	1001	2009-01-08	3.59
7	1001	2009-01-09	3.9
8	1001	2009-01-12	3.56
9	1001	2009-01-13	3.22
10	1001	2009-01-14	3.17

Index	Member
1	18
2	22
3	21
4	19
5	17
6	20
7	16
8	1
9	3
10	14

Index	Member
1	18
2	22
3	21

Index	Member
1	0.09691629955947145
2	0.061833688699360234
3	0.08816705336426933



用psort函数可以复用排序运算结果，一次排序就可以得到排序和排名。

	A	B
1	[7,15,8,9,23,5,22,5,5,5]	
2	=A1.psort()	/逆序排序后返回序号序列
3	=to(A1.len())	
4	>A2.run(if(A1(~)!=A1(~[-1]),A3(~)=#,A3(~)=A3(~[-1])))	/如果重复, 则排名不变
5	=A1.ranks()	/ranks计算排名

A1

Index	Member
1	7
2	15
3	8
4	9
5	23
6	5
7	22
8	5
9	5
10	5

A2

Index	Member
1	6
2	8
3	9
4	10
5	1
6	3
7	4
8	2
9	7
10	5

A3

Index	Member
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

A4

Index	Member
1	5
2	8
3	6
4	7
5	10
6	1
7	9
8	1
9	1
10	1

A5

Index	Member
1	5
2	8
3	6
4	7
5	10
6	1
7	9
8	1
9	1
10	1

A3执行时的A3

A4执行后的A3

使用psort函数建立索引。查看Name为“Andy”的信息。

	A	B
	1 =file("E:/txt/employees.txt").import@t(Name,Gender,BasePay)	/员工表
建立索引	2 =A1.psort(Name)	/返回排序后的序号
	3 =A1(A2)	/返回排序后的排列
索引查找	4 =A3.pselect@b(Name: "Andy")	/二分法查找 "Andy"
	5 =A2(A4)	/索引位置
	6 =A1(A5)	/返回所需数据

原序表

索引表

Index	Name	Gender	BasePay
1	Mike	Female	5600.
2	Jake	Male	2500.
3	Lucy	Female	10800.
4	Andy	Male	7500.
5	Jim	Male	4700.
6	David	Male	9300.
7	Jessica	Male	2900.

Index	Member
1	12
2	4
3	13
4	14
5	6
6	15
7	16

Index	Name	Gender	BasePay
1	Al	Male	7500.0
2	Andy	Male	7500.0
3	Ben	Male	10700.0
4	Charlie	Female	11000.0
5	David	Male	9300.0
6	David	Male	5700.0
7	Ed	Male	5700.0

索引表位置

原序表位置

结果

Value
2

Value
4

Name	Gender	BasePay
Andy	Male	7500.0

将人口排名表和GDP排名表按照指定的省份顺序进行排序

	A	B
1	[beijing,tianjin,heilongjiang,jilin,liaoning,neimenggu,xinjiang,ningxia,gansu,qinghai,shaanxi,xizang,sichuan,chongqing,guizhou,yunnan,shanxi,hebei,shandong,henan,anhui,jiangsu,shanghai,hubei,hunan,jiangxi,zhejiang,fujian,taiwan,guangxi,guangdong,hainan,xianggang,aomen]	/各省按照地域排序
2	=file("E:/txt/population.txt").import@t()	/各省人口排名表
3	=file("E:/txt/GDP.txt").import@t()	/各省GDP排名表
4	=A1.(A2.select@1(province==A1.~))	/按指定顺序筛选人口表
5	=A1.(A3.select@1(province==A1.~))	/按指定顺序筛选GDP表
6	=A2.align(A1,province)	
7	=A3.align(A1,province)	

A1~A7结果

Index	Member
1	beijing
2	tianjin
3	heilongjiang
4	jilin
5	liaoning
6	neimenggu
7	xinjiang

Index	province	Population
1	guangdong	10430.03
2	shandong	9579.31
3	henan	9402.36
4	sichuan	8041.82
5	jiangsu	7865.99
6	hebei	7185.42
7	hunan	6568.37

Index	province	2018_GDP
1	guangdong	9.73
2	jiangsu	9.26
3	shandong	7.63
4	zhejiang	5.54
5	henan	4.81
6	sichuan	4.07
7	hubei	3.94

Index	province	Population
1	beijing	1961.2
2	tianjin	1293.82
3	heilongjiang	3831.22
4	jilin	2746.22
5	liaoning	4374.63
6	neimenggu	2470.63
7	xinjiang	2181.33

Index	Member
1	[beijing,3.03]
2	[tianjin,1.88]
3	[heilongjiang,1.67]
4	[jilin,1.6]
5	[liaoning,2.53]
6	[neimenggu,1.73]
7	[xinjiang,1.15]

province	2018_GDP
beijing	3.03

Index	Member
28	[fujian,3.58]
29	(null)
30	[guangxi,2.04]
31	[guangdong,9.73]
32	[hainan,0.48]
33	(null)
34	(null)

同A4

同A5

	A	
1	[北京,天津,黑龙江,吉林,辽宁,内蒙古,新疆,宁夏,甘肃,青海,陕西,西藏,四川,重庆,贵州,云南,山西,河北,山东,河南,安徽,江苏,上海,湖北,湖南,江西,浙江,福建,台湾,广西,广东,海南,香港,澳门]	/各省按照地域排序
2	=file("E:/txt/population.txt").import@t()	/各省人口排名表
3	=file("E:/txt/GDP.txt").import@t()	/各省GDP排名表
4	=A2.align(A1,province)	/按指定顺序筛选人口表
5	=A3.align(A1,province)	/按指定顺序筛选GDP表
6	=A1.new(~:province,A5(#).#2/A4(#).#2*10000:Per_capita_GPD)	/对位计算, 得到结果

结果

Index	province	Per_capita_GPD
1	beijing	15.449724658372425
2	tianjin	14.53061476866952
3	heilongjiang	4.358924833342904
4	jilin	5.826190181412997
5	liaoning	5.78334624871132
6	neimenggu	7.002262580799228
7	xinjiang	5.272012946230052

A3缺少港澳台的GDP数据, 所以结果中包含null

Index	province	Per_capita_GPD
28	fujian	10.07882882882883
29	taiwan	(null)
30	guangxi	4.432219629518583
31	guangdong	9.328832227711713
32	hainan	5.535374502681196
33	xianggang	(null)
34	aomen	(null)

THANKS

感谢聆听 批评指导