向量运算

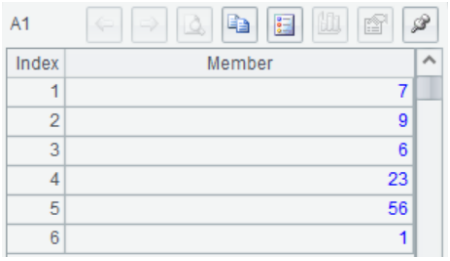
基本运算：

向量的生成

在 SPL 中用序列来表示向量，向量的元素被称为成员，成员和成员之间用逗号间隔。
比如：[7,9,6,23,56,1]表示一个有 6 个元素的向量，也就是有 6 个成员的序列。 SPL 代码和返回值如下：

	A
1	[7,9,6,23,56,1]

A1 输入一个向量，返回下图结果



向量的索引：

在 SPL 中可以通过位置引用向量元素

	A
1	[7,9,6,23,56,1]
2	=A1(3)
3	=A1.to(2)
4	=A1.to(3:5)
5	=A1.to(3:)
6	=A1.m(-2)
7	=A1.m([-2,-4])
8	=A1.m(-2:-4)

A1 输入一个向量
A2 取 A1 的第 3 位成员，返回 6。在 SPL 中所有的索引都是从 1 开始的。
A3 取前 2 位，返回 [1,2]
A4 取第 3 位到第 5 位，返回 [6,23,56]
A5 从第 3 位开始取到末尾，返回 [6,23,56,1]
A6 取 A1 的倒数第 2 位，返回 56。m 表示倒取

A7 取倒数第 2 位和倒数第 4 位，返回[56,6]
A8 从倒数第 2 位开始娶到倒数第 4 位，返回[56,23,6]

非零元素查找

mfind(A,n)，在 *A* 中查找前 *n* 个非 0 成员位置

	A
1	[0,6,0,23,56,1,0]
2	=mfind(A1,5)
3	=mfind(A1)
4	=mfind(A1,20)

A1 输入一个包含零元素的向量
A2 查找 A1 的前 3 个非零元素的位置。A1 中前 3 个非零元素为 6，23，56，对应的位置为 2，4，5，返回位置序列[2,4,5]
A3 *n* 省略时查找第 1 个位置，返回 2
A4 n 大于元素个数时，返回全部非零元素的位置序列[2,4,5,6]

向量的模

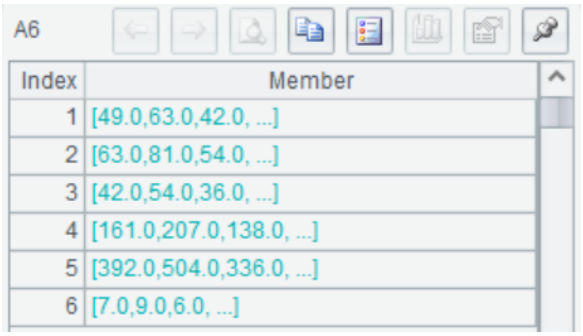
	A
1	[7,9,6,23,56,1]
2	=sqrt(sum(A1.(~*~)))

A1 输入向量
A2 计算向量 A1 的模，返回 61.903

向量的加减，数乘，点乘，叉乘

	A	B
1	[7,9,6,23,56,1]	[7,9,6,23,56,1]
2	=A1++B1	
3	=A1--B1	
4	=A1.(~*5)	
5	=A1.(~*B1(#)).sum()	
6	=mul(A1,B1)	

A2 A1 和 B1 相加，返回[14,18,12,46,112,2]
A3 A1 和 B1 相减，返回[0,0,0,0,0,0]
A4 向量的数乘，返回[35,45,30,115,280,5]
A5 计算 A1 和 B1 的点积，返回 3832
A6 计算 A1 和 B1 的叉乘，返回叉乘结果。使用 mul()函数做向量叉乘时，位于右边的向量 B1 会自动转置

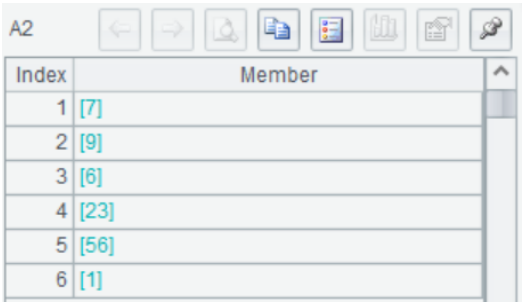


向量转置

	A
1	[7,9,6,23,56,1]
2	=transpose(A1)

A1 输入一个向量

A2 对 A1 进行转置，返回一个列向量

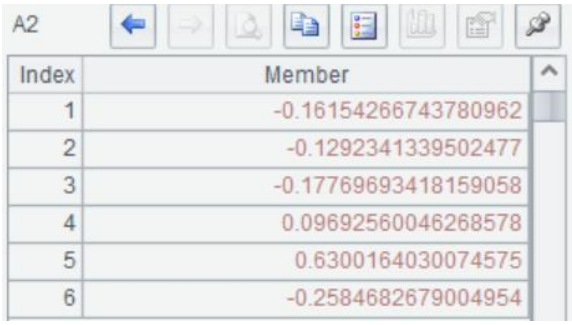


向量归一化

norm()可以用来对向量或矩阵进行归一化操作

	A
1	[7,9,6,23,56,1]
2	=norm(A1)
3	=norm@0(A1)
4	=norm@s(A1)

A2 对向量 A1 进行归一化，归一化方法为减去每行的均值后除以欧几里得范数（L2 范数）



A3 只减平均值，不做长度归一，归一化后均值为 0

A4 进行 0-1 归一化，归一化后均值为 0，标准差为 1

向量的统计指标

计算向量的最大最小值、平均值、中位数、众数、四分位数、极差、方差、标准差、偏度

	A
1	[7,9,6,23,56,1,6]
2	=A1.max()
3	=A1.min()
4	=A1.avg()
5	=A1.mode()
6	=A1.median()
7	=A1.median(1:4)
8	=A1.median(3:4)
9	=var@s(A1)
10	=sqrt(A9)

A2 计算 A1 的最大值
A3 计算 A1 的最小值
A4 计算 A1 的平均值
A5 计算 A1 的众数
A6 计算 A1 的中位数
A7 计算 A1 的下四分位数
A8 计算 A1 的上四分位数
A9 计算 A1 的方差
A10 计算 A1 的标准差

计算偏度的公式稍微有些复杂，我们写成脚本的形式保存为 skew_calc.dfx 方便调用

	A
1	=seq.avg()
2	=seq.count()
3	=seq.sum(power((~-A1),3))/A2
4	=power(seq.sum(power((~-A1),2))/A2,1.5)
5	=sqrt(A2*(A2-1))/(A2-2)
6	return A5*A3/A4

seq 是脚本的参数名，传入需要计算偏度的向量，比如：

	A
1	[7,9,6,23,56,1,6]
2	=call("skew_calc.dfx",A1)

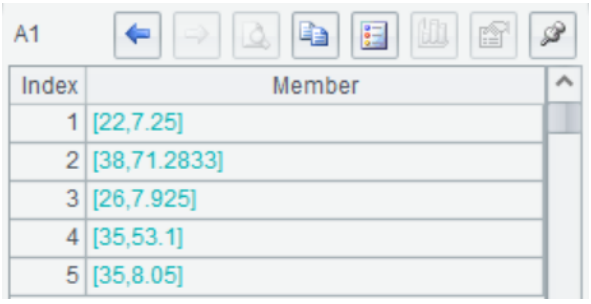
A2 调用脚本，计算 A1 的偏度

矩阵运算

基本运算

矩阵的生成

在 SPL 中，矩阵以二级序列的形式存在，比如[[22,7.25],[38,71.2833],[26,7.925],[35,53.1],[35,8.05]]，每一个子序列表示一行，通常是一个样本记录；子序列中同一位置的元素表示一列,如[22,38,26,35,35]为矩阵的第一列，通常是一个变量



矩阵的索引

可以用序列操作对矩阵元素进行引用

	A
1	[[22,7.25],[38,71.2833],[26,7.925],[35,53.1],[35,8.05]]
2	=A1(1)
3	=A1.(~(1))
4	=A1(3)(1)
5	=A1.to(2)
6	=A1.to(3:5)
7	=A1.to(3:)
8	=A1.m(-2)
9	=A1.m([-2,-4])
10	=A1.m(-2:-4)

A1 输入一个矩阵
A2 取矩阵的第 1 行
A3 取矩阵的第 1 列
A4 取矩阵的第 3 行第 1 个元素
A5 取矩阵的前 2 行
A6 取矩阵的第 3 行到第 5 行
A7 从第 3 行开始取到末尾

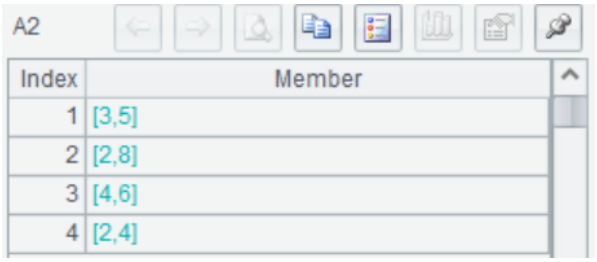
A8 取 A1 的倒数第 2 行。m 表示倒取
A9 取倒数第 2 行和倒数第 4 行
A10 从倒数第 2 行开始取到倒数第 4 行

矩阵合并

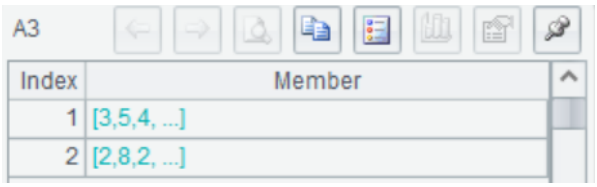
将矩阵$\begin{bmatrix}3 & 5\\2 & 8\end{bmatrix}$、$\begin{bmatrix}4 & 6\\2 & 4\end{bmatrix}$进行上下合并和左右合并

	A	B
1	[[3,5],[2,8]]	[[4,6],[2,4]]
2	=A1|B1	
3	=transpose([A1,B1]).(~.conj())	

A2 矩阵上下合并，返回 4 行 2 列的矩阵



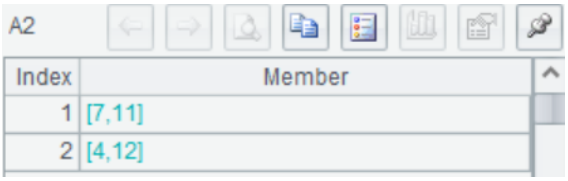
A3 矩阵左右合并，返回 2 行 4 列的矩阵



矩阵的加法和减法

	A	B
1	[[3,5],[2,8]]	[[4,6],[2,4]]
2	=A1.(~++B1(#))	
3	=A1.(~--B1(#))	

A2 A1 和 B1 的元素相加，返回[[7,11],[4,12]]



A3 A1 和 B1 的元素相减，返回[[-1,-1],[0,4]]

A3

Index	Member
1	[-1,-1]
2	[0,4]

数乘和点乘

	A	B
1	[[3,5],[2,8]]	[[4,6],[2,4]]
2	=A1.(~.(~*5))	
3	=A1.(~.(~*B1(A1.#)(#)))	

A2 数乘，A1 的每个元素都乘以 5，返回[[15,25],[10,40]]

A2

Index	Member
1	[15,25]
2	[10,40]

A3 点乘，矩阵 A1 和矩阵 B1 的对应元素相乘，返回[[12,30],[4,32]]

A3

Index	Member
1	[12,30]
2	[4,32]

矩阵相乘

mul(A,X)函数将两个矩阵相乘，例如$\begin{bmatrix}3 & 5\\2 & 8\end{bmatrix}$和 $\begin{bmatrix}4 & 6\\2 & 4\end{bmatrix}$相乘

	A	B
1	[[3,5],[2,8]]	[[4,6],[2,4]]
2	=mul(A1,B1)	

A2 将 A1 和 B1 相乘，返回矩阵

A2

Index	Member
1	[22.0,38.0]
2	[24.0,44.0]

矩阵行列式

当矩阵为方阵时可计算行列式，使用 det(A)函数

	A
1	[[3,5],[2,8]]
2	=det(A1)

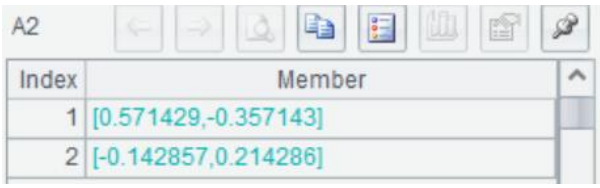
A2 计算 A1 的行列式，返回 14

矩阵的逆

inverse(A)计算方阵的逆，没有逆时返回空

	A
1	[[3,5],[2,8]]
2	=inverse(A1)

A2 求 A1 的逆矩阵，返回



多维矩阵计算

矩阵的维度

1 维矩阵：其实就是向量，比如[1,2,3]
2 维矩阵：最常见的矩阵，比如[[11,12,13],[21,22,23],[31,32,33]]
3 维矩阵：比如
[[[111,112,113],[121,122,123],[131,132,133]],[[211,212,213],[221,222,223],[231,232,233]]]
…
以此类推
在一些矩阵运算函数中，通常会用**维度层数 n** 来描述参与计算的成员，比在 2 维矩阵[[11,12,13],[21,22,23],[31,32,33]]中，第 1 层维度成员指 3 个向量成员[11,12,13]、[21,22,23]、[31,32,33]，第 2 层维度指每个向量的成员如第一个向量包含 11、12、13。

Index	Member
1	[11,12,13]
2	[21,22,23]
3	[31,32,33]

Index	Member
1	11
2	12
3	13

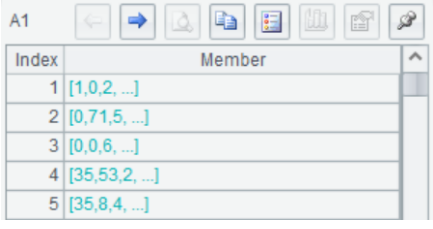
同理在 3 维向量
[[[111,112,113],[121,122,123],[131,132,133]],[[211,212,213],[221,222,223],[231,232,233]]]中，n=1 时，指两个 3*3 的矩阵
n=2 时，指每个矩阵的向量成员
n=3 时，指每个向量的成员

矩阵求和

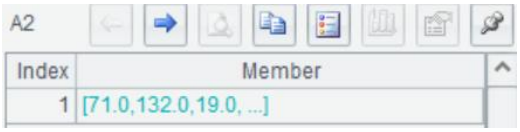
msum(*A, n*)表示在矩阵或多维矩阵中汇总求和，其中 n 为汇总的维度层数
比如一个 5*4 的矩阵[[1,0,2,3],[0,71,5,6],[0,0,6,5],[35,53,2,3],[35,8,4,2]]，可以通过设置 n 来实现行或列求和

	A
1	[[1,0,2,3],[0,71,5,6],[0,0,6,5],[35,53,2,3],[35,8,4,2]]
2	=msum(A1,1)
3	=msum(A1,2)
4	=msum@a(A1)

A1 输入矩阵



A2 n=1，在第 1 层维度上汇总求和，将每个成员向量对应位置的元素相加，即返回每列的和



A3 n=2, 在第 2 层维度上汇总求和，将每个向量内的元素相加，即返回每一行的和

A3

序号	成员
1	6.0
2	82.0
3	11.0
4	93.0
5	49.0

A4 @a 表示对 A1 全部元素求和

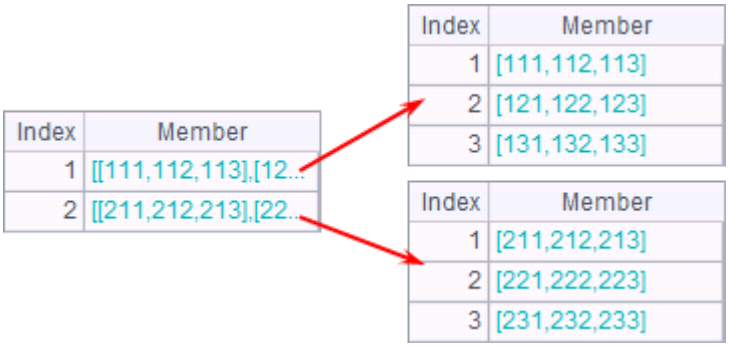
A4

Value
241.0

再比如，在一个 3 维矩阵中同样可以通过设置 n 在不同维度求和

	A
1	[[[111,112,113],[121,122,123],[131,132,133]],[[211,212,213],[221,222,223],[231,232,233]]]
2	=msum(A1, 1)
3	=msum(A1, 2)
4	=msum(A1, 3)
5	=msum@a(A1)

A1 输入一个 3 维矩阵，有 2 个成员，均为为 3*3 的矩阵



A2 n=1，在第 1 层维度上汇总求和，即两个成员矩阵对位相加

A2

Index	Member
1	[322.0,324.0,326.0]
2	[342.0,344.0,346.0]
3	[362.0,364.0,366.0]

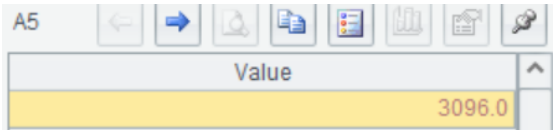
A3 n=2，在第 2 层维度上汇总求和，即每个成员矩阵内的向量对位相加

A3

Index	Member
1	[363.0,366.0,369.0]
2	[663.0,666.0,669.0]

A4 n=3，在第 3 层维度上汇总求和，即每个向量内的成员相加求和

A5 @a 对多维矩阵中的所有元素求和，返回数值

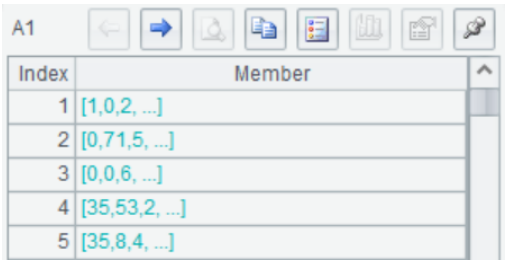


矩阵元素累计求和

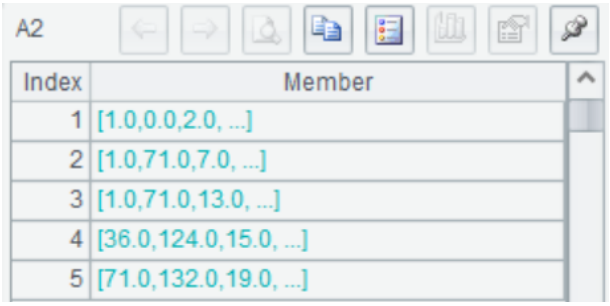
mcumsum(*A*, *n*)对矩阵 A 的元素在第 n 层维度累计求和

	A
1	[[1,0,2,3],[0,71,5,6],[0,0,6,5],[35,53,2,3],[35,8,4,2]]
2	=mcumsum(A1,1)
3	=mcumsum(A1,2)
4	=mcumsum@z(A1,1)

A1 输入一个 2 维矩阵，有 5 个成员向量



A2 n=1，在第 1 层维度上累计求和，将每个成员向量对位累加，即返回每列的累计求和，例如第一列元素累计求和的结果为 1，1，1，36，71



A3 n=2，在第 2 层维度上累计求和，将每个向量内的成员累计求和，即返回每行的累计和

A3

Index	Member
1	[1.0,1.0,3.0, ...]
2	[0.0,71.0,76.0, ...]
3	[0.0,0.0,6.0, ...]
4	[35.0,88.0,90.0, ...]
5	[35.0,43.0,47.0, ...]

A4 对矩阵 A1 的第一层维度逆向累计求和，@z 表示逆向累计

A4

Index	Member
1	[71.0,132.0,19.0, ...]
2	[70.0,132.0,17.0, ...]
3	[70.0,61.0,12.0, ...]
4	[70.0,61.0,6.0, ...]
5	[35.0,8.0,4.0, ...]

矩阵元素求均值

mmean(*A*, *n*)对矩阵 A 在 n 层维度求均值

	A
1	[[1,0,2,3],[0,71,5,6],[0,0,6,5],[35,53,2,3],[35,8,4,2]]
2	=mmean(A1,1)
3	=mmean(A1,2)
4	=mmean@a(A1)

A1 输入一个 2 维矩阵，有 5 个成员向量

A1

Index	Member
1	[1,0,2, ...]
2	[0,71,5, ...]
3	[0,0,6, ...]
4	[35,53,2, ...]
5	[35,8,4, ...]

A2 n=1，在第 1 层维度上计算，将每个成员向量对位求均值，即返回每列的均值

A2

Index	Member
1	[14.2,26.4,3.8, ...]

A3 n=2，在第 2 层维度上计算，将每个向量内的球员求均值，即返回每行的均值

A3

Index	Member
1	1.5
2	20.5
3	2.75
4	23.25
5	12.25

A4 对矩阵 A1 的所有元素求均值，@a 表示汇总全部元素

A4

Value
12.05

矩阵元素求标准差

mstd(*A*, *n*)对矩阵 A 在 n 层维度方向上求标准差

	A
1	[[1,0,2,3],[0,71,5,6],[0,0,6,5],[35,53,2,3],[35,8,4,2]]
2	=mstd(A1,1)
3	=mstd(A1,2)
4	=mstd@s(A1,2)

A1 输入一个 2 维矩阵，有 5 个成员向量

A1

Index	Member
1	[1,0,2,...]
2	[0,71,5,...]
3	[0,0,6,...]
4	[35,53,2,...]
5	[35,8,4,...]

A2 n=1，在第 1 层维度上计算，将每个成员向量对位求标准差，即返回每列的标准差

A2

Index	Member
1	16.98705389406886
2	29.763064358362026
3	1.5999999999999999
4	1.469693845669907

A3 n=2，在第 1 层维度上计算，将每个向量内的成员汇总求标准差，即返回每行的标准差

A3

Index	Member
1	1.118033988749895
2	29.244657631779518
3	2.7726341266023544
4	21.706853756359994
5	13.311179511974137

A4 采用统计学算法对矩阵 A1 的每一行求标准差，@s 统计学计算，除以 n-1

A4

Index	Member
1	1.2909944487358056
2	33.768821912132694
3	3.2015621187164243
4	25.064915718988566
5	15.370426148939398

非零元素查找

mfind(A,n)，在 *A* 中按列依次查找前 *n* 个非 0 成员位置

	A
1	[[1,0],[0,71.2833],[0,0],[35,53.1],[35,8.05]]
2	=mfind(A1,4)
3	=mfind(A1)
4	=mfind(A1,100)

A1 输入一个包含零元素的矩阵

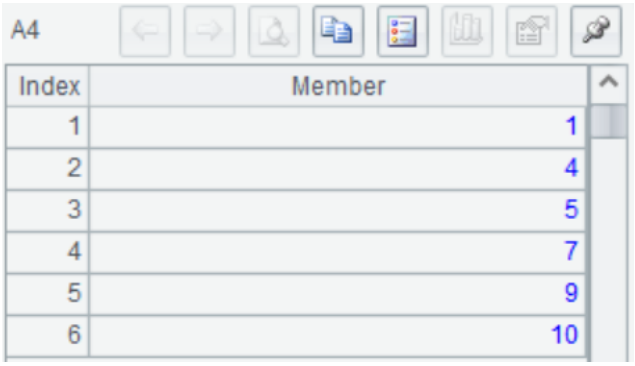
A2 按列查找 A1 的前 4 个非零元素的位置。A1 中前 4 个非零元素为 1，35，35，71.2833 对应的位置为 1，4，5，7，因此返回位置序列[1,4,5,7]

A2

Index	Member
1	1
2	4
3	5
4	7

A3 *n* 省略时查找第 1 非零元素个位置，返回 1

A3 n 大于元素个数时，返回全部非零元素的位置序列


A4

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

矩阵变换

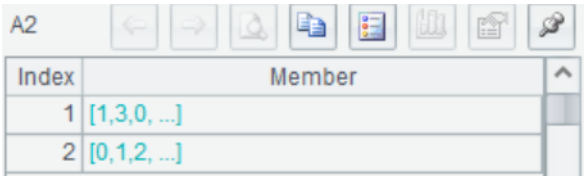
矩阵的转置

transpose(A)可对矩阵进行转置

例如将$\begin{bmatrix}1 & 0\\3 & 1\\0 & 2\\4 & 8\end{bmatrix}$进行转置

	A
1	[[1,0],[3,1],[0,2],[4,8]]
2	=transpose(A1)

A2 计算 A1 的转置矩阵，返回 2 行 4 列的矩阵


A2

Index	Member
1	[1,3,0, ...]
2	[0,1,2, ...]

矩阵求秩

rankm(A)返回矩阵 A 的秩

	A
1	[[1,0],[3,1],[0,2],[4,8]]
2	=rankm(A1)

A2 计算 A1 的秩，返回 2

矩阵行变换化简（A 的行最简形式）
矩阵的特征多项式
矩阵的特征值
矩阵的特征向量
矩阵的对角化

矩阵归一化

使用 mnorm(A, n)可以对矩阵在不同维度方向上进行归一化

例如，将矩阵$\begin{bmatrix}8 & 1 & 6\\3 & 5 & 7\\4 & 9 & 2\end{bmatrix}$归一化

	A
1	[[8,1,16],[3,5,7],[4,9,2]]
2	=mnorm(A1,1)
3	=mnorm(A1,2)
4	=mnorm@s(A1,1)

A2 将矩阵 A1 的每一列进行标准化，标准化后每列均值为 0,标准差为 1

A3 将矩阵 A1 的每一行进行标准化，标准化后每行均值为 0,标准差为 1

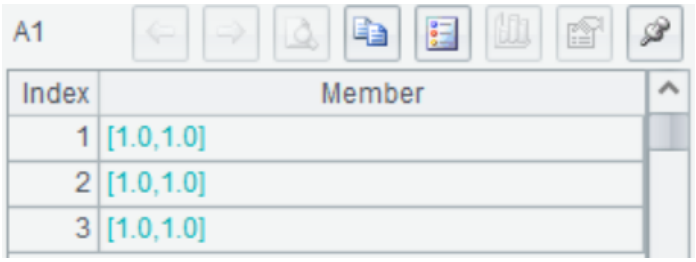
A4 采用统计学概念标准化，@s 表示统计学算法，除以 n-1

特殊矩阵生成

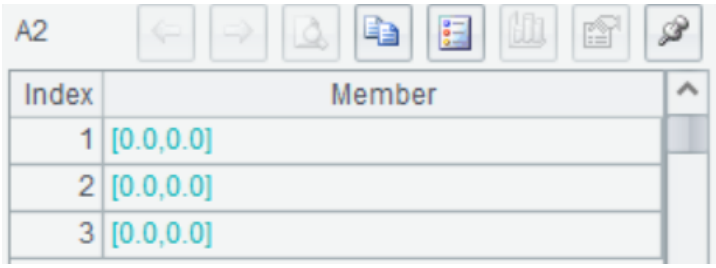
全“1”矩阵，零矩阵，主对角线为”1”的矩阵，单位矩阵

	A
1	=ones(3,2)
2	=zeros(3,2)
3	=eye(3,4)
4	=I(3)

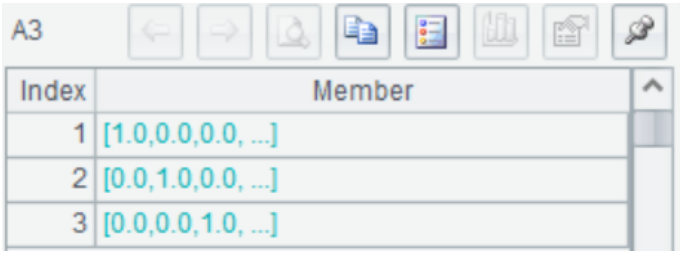
A1 生成 3 行 2 列的全”1”矩阵



A2 生成 3 行 2 列的零矩阵

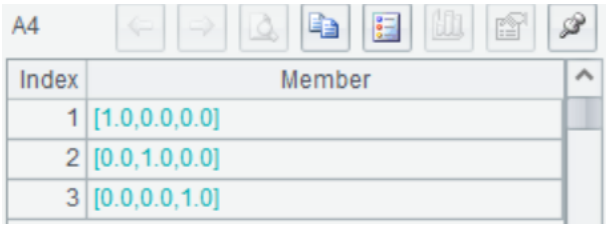


A3 生成 3 行 4 列，主对角线为“1”的矩阵


A3

Index	Member
1	[1.0,0.0,0.0, ...]
2	[0.0,1.0,0.0, ...]
3	[0.0,0.0,1.0, ...]

A4 生成 3 阶单位矩阵


A4

Index	Member
1	[1.0,0.0,0.0]
2	[0.0,1.0,0.0]
3	[0.0,0.0,1.0]

变量相关性评估

协方差

cov(A,B)，用来计算两个向量的协方差

	A	B
1	=[7,9,6,23,56,1,6]	=[6,1,56,23,6,9,7]
2	=cov(A1,B1)	

A2 计算 A1 和 B1 的协方差

协方差矩阵

covm(A)用来计算矩阵的协方差矩阵

	A
1	[[8,1,16],[3,5,7],[4,9,2]]
2	=covm(A1)

A2 计算 A1 的协方差矩阵

相关系数

SPL 中提供 pearson()和 spearman()函数，来评估两变量之间的相关程度

	A	B
1	=[7,9,6,23,56,1,6]	=[6,1,56,23,6,9,7]
2	=pearson(A1,B1)	
3	=spearman(A1,B1)	

A2 计算 A1 和 B1 的 pearson 相关系数

A2 计算 A1 和 B1 的 spearman 相关系数

相关系数矩阵

相关系数矩阵反映的是两两变量之间的相关程度，一般是计算皮尔逊相关系数。如下表，有 x1，x2，x3，x4 四个变量，计算其相关系数矩阵，评估变量相关性。

	x1	x2	x3	x4
1	7	26	6	60
2	1	29	15	52
3	11	56	8	20
4	11	31	8	47
5	7	52	6	33
6	11	55	9	22

SPL 代码如下：

	A	B
1	[[7,26,6,60],[1,29,15,52],[11,56,8,20],[11,31,8,47],[7,52,6,33],[11,55,9,22],[3,71,17,6],[1,31,22,44],[2,54,18,22],[21,47,4,26],[1,40,23,34],[11,66,9,12],[10,68,8,12]]	[[x1,x2,x3,x4]]
2	=transpose(A1)	
3	for A2	=A2.(pearson(A3,~))
4		>B1=B1|[B3]
5	=transpose(B1)	
6	=[B1(1).insert(1,"")]|A5	

A1 输入样本数据

B1 定义一个序列，保存计算结果

A2 将 A1 转置

A3:B4 循环 A2, 计算变量两两之间的相关系数，结果存入 B1

A5 转置 B1

A6 加入变量名称

上述代码执行完毕后，可生成相关系数矩阵如下表：

	x1	x2	x3	x4
x1	1	0.228579	-0.82413	-0.24545
x2	0.228579	1	-0.13924	-0.97295
x3	-0.82413	-0.13924	1	0.029537
x4	-0.24545	-0.97295	0.029537	1

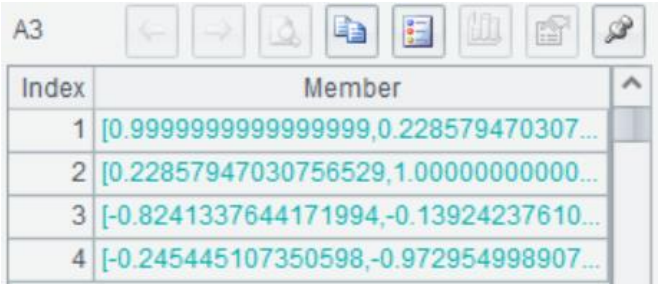
除了上述方法，相关系数矩阵还可以通过计算标准化后的协方差矩阵来获得。

	A
1	[[7,26,6,60],[1,29,15,52],[11,56,8,20],[11,31,8,47],[7,52,6,33],[11,55,9,22],[3,71,17,6],[1,31,22,44],[2,54,18,22],[21,47,4,26],[1,40,23,34],[11,66,

	9,12],[10,68,8,12]]
2	=mnorm@s(A1,1)
3	=covm(A2)

A2 将 A1 按列标准化，标准化后中心值为 0，标准差为 1。@表示采用统计学计算除以 n-1

A2 计算标准化后协方差矩阵，结果即为相关系数矩阵。



线性方程组求解

求逆法

对于 AX=Y，若 A 的逆矩阵存在，可用求逆法解方程组

$x1 + 2 * x2 = 8$

$2 * x1 + 3 * x2 = 13$

求解代码如下：

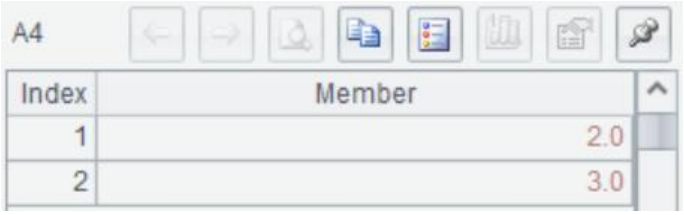
	A
1	[[1,2],[2,3]]
2	[8,13]
3	=inverse(A1)
4	=mul(A3,A2)

A1 输入系数矩阵

A2 输入 Y

A3 求 A1 的逆矩阵

A4 A3 和 A2 矩阵相乘，求解出$x1 = 2$，$x2 = 3$



最小二乘法求解

$x1 - 2 * x2 + 3 * x3 = -10$

$2 * x1 + 3 * x2 + 4 * x3 = 5$

$3*x1+5*x2+7*x3=7$

使用 linefit()可采用最小二乘法求解方程组

	A
1	[[1,-2, 3],[2,3,4],[3,5,7]]
2	[-10,5,7]
3	=linefit(A1,A2)

A1 输入系数矩阵

A2 输入 Y

A3 最小二乘法求解方程组，求解出$x1=2$，$x2=3$，$x3=-2$

A3

Index	Member
1	[2.0]
2	[3.0]
3	[-2.0]

参数估计和假设检验

参数估计

在参数估计和假设检验中，常常需要构造分布函数，在 SPL 中提供了常见分布的逆累积分布函数（ICDF）。SPL 中的逆累积分布函数有：norminv()，tinv()，chi2inv()，finv()，分别表示正态分布，T 分布，卡方分布，F 分布。

例如，有一批产品寿命的变量服从正态分布，均值 mu=4，标准差 sigma=1.3，单位是年，计算 10%的产品失效所需时间

	A
1	=norminv(0.1,4,1.3)

A1 采用正态逆累积分布函数，带入概率值、均值和标准差，返回 2.334，即 10%得产品失效所需时间为 2.334 年

A1

Value
2.333982964792019

使用逆累积分布函数还可以进行区间估计

例如，使用金球测定引力常数（单位：$10^{-11}m^3\cdot kg^{-1}\cdot s^{-2}$），测定观察值为

6.683，6.681，6.676，6.678，6.679，6.672

设测定值总体为 N(μ,σ^2)，μ，σ^2均未知，求μ的置信度为 0.9 的置信区间，并求σ^2的置信度为 0.9 的置信区间。

当μ，σ^2均未知时，μ的置信区间公式为（$\bar{X}-\frac{S}{\sqrt{n}}t_{\alpha/2}(n-1),\bar{X}+\frac{S}{\sqrt{n}}t_{\alpha/2}(n-1)$）

其中$\bar{X}$为样本均值，S 为样本标准差，n 为样本容量

在本例中 1-α=0.9，n=6，带入公式

用 SPL 代码计算μ的区间估计

	A	B	C
1	[6.683,6.681,6.676,6.678,6.679,6.672]	0.9	6
2	=tinv(1-(1-B1)/2,C1-1)		
3	=sqrt(var@s(A1))/sqrt(C1)*A2		
4	=avg(A1)		
5	=[A4-A3,A4+A3]		

A1 输入样本数据

B1 置信度，即 1-α

C1 样本容量 n

A2 使用 T 分布逆累积函数 tinv(p, nu)计算$t_{\alpha/2}(n-1)$，p 表示累积概率，nu 表示自由度，即 p=1-$\alpha/2=0.95$，nu=n-1=5

A3 计算$\frac{s}{\sqrt{n}}t_{\alpha/2}(n-1)$

A4 计算样本均值$\bar{X}$

A5 带入 A3 和 A4 的结果，计算置信区间，根据返回值得到总体均值μ，在置信度为 0.9 时的置信区间为（6.675，6.681）

A5

Index	Member
1	6.674984137487131
2	6.681349195846201

同理，方差σ^2的置信区间公式为（$\frac{(n-1)S^2}{\chi^2_{\alpha/2}(n-1)},\frac{(n-1)S^2}{\chi^2_{1-\alpha/2}(n-1)}$），$S^2$为样本方差

SPL 计算σ^2的区间估计

	A	B	C
1	[6.683,6.681,6.676,6.678,6.679,6.672]	0.9	6
2	=var@s(A1)		
3	=chi2inv(1-(1-B1)/2,C1-1)		
4	=chi2inv((1-B1)/2,C1-1)		
5	=[(C1-1)*A2/A3,(C1-1)*A2/A4]		

A1 输入样本数据

B1 置信度，即 1-α

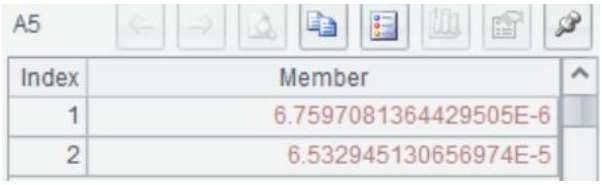
C1 样本容量 n

A2 计算样本方差

A3 使用卡方分布逆累积函数 chi2inv(p, v)计算$\chi^2_{\alpha/2}(n-1)$，p 表示累积概率，v 表示自由度，即 p=1-$\alpha/2=0.95$，nu=n-1=5

A4 计算$\chi^2_{1-\alpha/2}(n-1)$

A5计算置信区间，根据返回值得到总体方差σ^2，在置信度为0.9时的置信区间为（6.75×10^{-6}，6.533×10^{-5}）



回归模型的假设检验

在多元线性回归模型中，因变量 y 和自变量x_1，x_2……x_m之间是否存在线性关系以及回归系数的显著性是需要检验的，比如在下面的例子中我们可以通过 F 检验和 t 检验来分析判断自变量和因变量之间的关系。

表中的数据为某养猪场 25 头育肥猪 4 个胴体性状的数据资料，需要进行瘦肉量 y 对眼肌面积x_1、腿肉量x_2、腰肉量x_3的多元回归分析

序号	眼肌面积x_1	腿肉量x_2	腰肉量x_3	瘦肉量 y
1	23.73	5.49	1.21	15.02
2	22.34	4.32	1.35	12.62
3	28.84	5.04	1.92	14.86
4	27.67	4.72	1.49	13.98
5	20.83	5.35	1.56	15.91
6	22.27	4.27	1.5	12.47
7	27.57	5.25	1.85	15.8
8	28.01	4.62	1.51	14.32
9	24.79	4.42	1.46	13.76
10	28.96	5.3	1.66	15.18
11	25.77	4.87	1.64	14.2
12	23.17	5.8	1.9	17.07
13	28.57	5.22	1.66	15.4
14	23.52	5.18	1.98	15.94
15	21.86	4.86	1.59	14.33
16	28.95	5.18	1.37	15.11
17	24.53	4.88	1.39	13.81
18	27.65	5.02	1.66	15.58
19	27.29	5.55	1.7	15.85
20	29.07	5.26	1.82	15.28
21	32.47	5.18	1.75	16.4
22	29.65	5.08	1.7	15.02
23	22.11	4.9	1.81	15.73
24	22.43	4.65	1.82	14.75
25	20.04	5.08	1.53	14.35

要求：（1）求 y 关于x_1，x_2，x_3的线性回归方程

$$y = c_0 + c_1x_1 + c_2x_2 + c_3x_3$$

计算c_0，c_1，c_2，c_3的估计值

(2)对上述回归模型和回归系数进行检验，显著性水平α取 0.05

分析：

（1）线性回归可以采用最小二乘法进行拟合，使用 linefit()，返回拟合回归系数c_0，c_1，c_2，c_3

（2）使用 F 检验 y 和x_1，x_2，x_3之间是否存在线性关系，如通过检验，则继续使用 t 检验回归系数的显著性，需要计算的统计量有：F，$F_{1-\frac{\alpha}{2}}(m,n-m-1)$，$F_{\frac{\alpha}{2}}(m,n-m-1)$，t，$t_{\frac{\alpha}{2}}(n-m-1)$。

计算公式如下：

F=$\frac{SSR/m}{SSE/(n-m-1)}$

其中：m 表示分子自由度，n 表示样本数量

回归平方和：SSR=$\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2$

残差平方和：SSE=$\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$

$$t_j=\frac{c_{j/\sqrt{c_{jj}}}}{\sqrt{SSE/(n-m-1)}}$$

其中c_{jj}为$(X^TX)^{-1}$中的第（j+1,j+1）个元素

$F_{\frac{\alpha}{2}}(m,n-m-1)$，$t_{\frac{\alpha}{2}}(n-m-1)$可使用逆累积函数 finv()和 tinv()计算

用 SPL 实现计算：

（1） 采用最小二乘法，拟合回归方程

$$\begin{bmatrix}1 & \cdots & x_{1,3}\\ \vdots & \ddots & \vdots\\ 1 & \cdots & x_{25,3}\end{bmatrix}\begin{bmatrix}c_0\\ c_1\\ c_3\\ c_4\end{bmatrix}=\begin{bmatrix}y_1\\ \vdots\\ \vdots\\ y_{25}\end{bmatrix}$$

求解出c_0，c_1，c_2，c_3的估计值

	A	B	C	D
1	[[1,23.73,5.49,1.21],[1,22.34,4.32,1.35],[1,28.84,5.04,1.92],[1,27.67,4.72,1.49],[1,20.83,5.35,1.56],[1,22.27,4.27,1.5],[1,27.57,5.25,1.85],[1,28.01,4.62,1.51],[1,24.79,4.42,1.46],[1,28.96,5.3,1.66],[1,25.77,4.87,1.64],[1,23.17,5.8,1.9],[1,28.57,5.22,1.66],[1,23.52,5.18,1.98],[1,21.86,4.86,1.59],[1,28.95,5.18,1.37],[1,24.53,4.88,1.39],[1,27.65,5.02,1.66],[1,27.29,5.55,1.7],[1,29.07,5.26,1.82],[1,32.47,5.18,1.75],[1,29.65,5.08,1.7],[1,22.11,4.9,1.81],[1,22.43,4.65,1.82],[1,20.04,5.08,1.53]]	3	25	0.
2	[15.02,12.62,14.86,13.98,15.91,12.47,15.8,14.32,13.76,15.18,14.2,17.07,15.4,15.94,14.33,15.11,13.81,15.58,15.85,15.28,16.4,15.02,15.73,14.75,14.35]			
3	=linefit(A1,A2)			

A1 输入矩阵 X

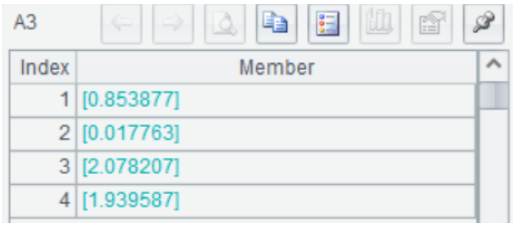
B1 输入 m 值，本例中 m=3

C1 输入 n 值，本例中 n=25

D1 输入α，本例中$\alpha = 0.05$

A2 输入 Y

A3 最小二乘拟合方程组，返回回归系数c_0，c_1，c_2，c_3的估计值



（2） 使用 F 检验因变量 y 与自变量x_1，x_2，x_3之间是否存在线性关系。令原假设为

H_0:c_j=0， j=1，2，3。在显著性水平α下，若$F_{1-\frac{\alpha}{2}}(m,n-m-1) < F < F_{\frac{\alpha}{2}}(m,n-m-1)$，

则接受H_0，否则拒绝

	A	B	C	D
…	…			
4	=mul(A1,A3).conj()			
5	=(A2--A4).sum(~*~)			
6	=A2.avg()			
7	=A4.sum((~-A6)*(~-A6))			
8	=(A7/B1)/(A5/(C1-B1-1))			
9	=finv((D1/2),B1,(C1-B1-1))			
10	=finv((1-D1/2),B1,(C1-B1-1))			
11	=if(A8>A9&&A8<A10,"accept H0,model do not pass","refuse H0,model pass")			

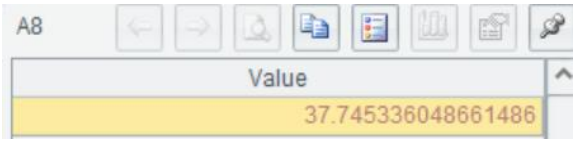
A4 将 X 与回归系数相乘，得到预测值

A5 计算残差平方和 SSE

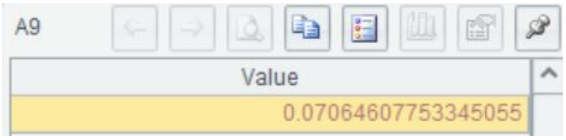
A6 计算$\bar{y}$

A7 计算回归平方和 SSR

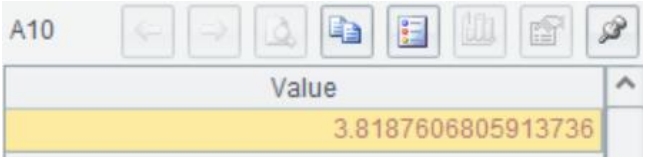
A8 计算统计量 F，F=37.753



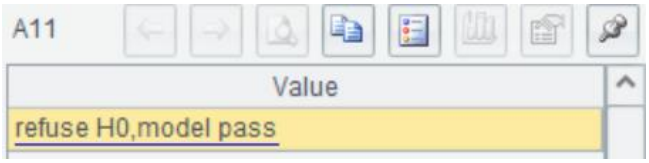
A9 使用 F 逆累积分布函数 finv()计算上 1-$\alpha/2$分位数



A10 使用 F 逆累积分布函数 finv()计算上$\alpha/2$分位数



A11 比较 F 和$F_{1-\frac{\alpha}{2}}(m,n-m-1)$，$F_{\frac{\alpha}{2}}(m,n-m-1)$，返回检验结果



（3） 当H_0被拒绝时，c_j不全为 0，但不排除其中若干个等于 0，所以应进一步做 m+1 个 t 检验：

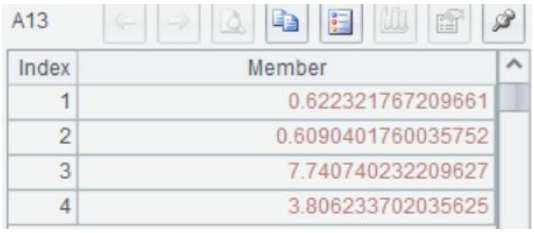
$$H_0^{(j)}: c_j = 0，\ j = 0，\ 1，\ \ldots，\ m$$

对于给定的α，若$|t_j|<t_{\frac{\alpha}{2}}(n-m-1)$，则接受$H_0^{(j)}$，否则拒绝

	A	B	C	D
…	…			
12	=inverse(mul(transpose(A1),A1))			
13	=A3.conj().(~/sqrt(A12(#)(#))/sqrt(A5/(C1-B1-1)))			
14	=tinv((1-D1/2),(C1-B1-1))			
15	=A13.(if(abs(~)<A14,"accept c"/(#-1)/"=0","refuse c"/(#-1)/"=0"))			

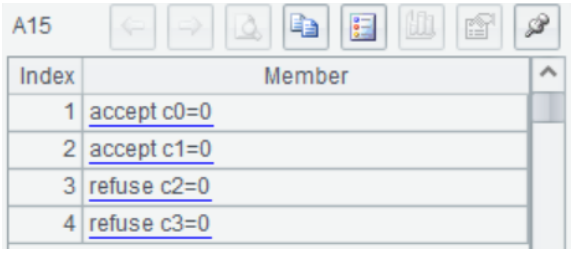
A12 计算$（X^TX）^{-1}$

A13 计算t_j，从上到下依次为t_0，t_1，t_2，t_3



A14 使用 t 逆累积分布函数 tinv()计算 t 分布的上$\alpha/2$分位数

A15 比较$|t_j|$和$t_{\frac{\alpha}{2}}(n-m-1)$，返回检验结果



由结果可知变量 x1 对模型的影响是不显著的，建立线性模型时可以不使用 x1

样本聚类

样本间统计距离

在一些聚类分析中，经常需要度量样本间的相似性，在 SPL 中可以通过计算欧式距离或马氏距离来评估样本间的相似性

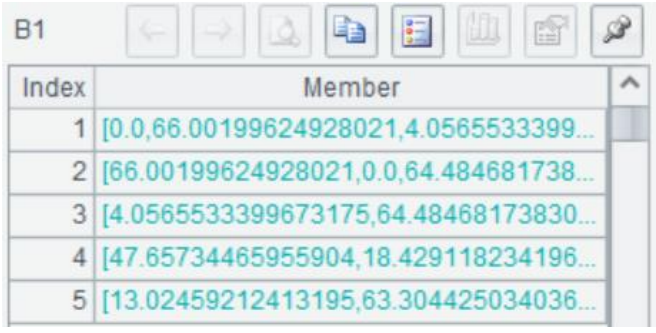
欧式距离

欧式距离即直线距离，使用函数 dis(A,B)可计算向量 A 和向量 B 之间的欧式距离。例如有下表中 5 个样本，求两两样本之间的欧式距离

	X	Y
1	22	7.25
2	38	71.2833
3	26	7.925
4	35	53.1
5	35	8.05

	A	B
1	[[22,7.25],[38,71.2833],[26,7.925],[35,53.1],[35,8.05]]	[]
2	for A1	=A1.(dis(A2,~))
3		>B1=B1|[B2]

A1 输入样本数据
B1 定义一个空序列，用来保存计算结果
A2:B3 循环 A1 中的样本，计算两两样本之间的欧式距离，结果存入 B1
上述代码执行完毕后，B1 返回样本之间的欧式距离矩阵



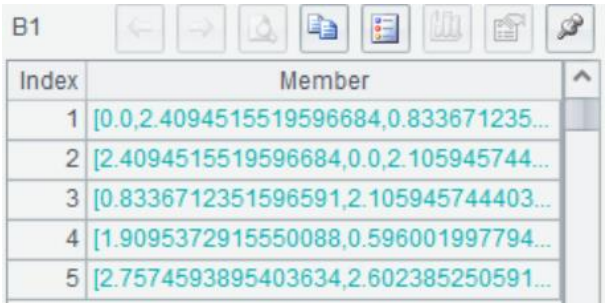
马氏距离

马氏距离计算观测样本在总体样本中的距离，不受量纲的影响。dism(X,Y,C) 用来计算向量 X 与向量 Y 在协方差矩阵 C 下的马氏距离。样本数要大于维数。例如同样上述 5 个样本，计

算两两之间的马氏距离，代码如下：

	A	B
1	[[22,7.25],[38,71.2833],[26,7.925],[35,53.1],[35,8.05]]	[]
2	=covm(A1)	
3	for A1	=A1.(dism(A3,~,A2))
4		>B1=B1|[B3]

A1 输入样本数据
B1 定义一个空序列，用来保存计算结果
A2 计算样本空间 A1 的协方差矩阵
A3:B4 循环 A1 中的样本，计算两两样本之间的马氏距离，结果存入 B1
上述代码执行完毕后，B1 返回样本之间的马氏距离矩阵



聚类方法

数学建模

建模准备

pca 降维

主成分分析是一种常用的降维方法，能将许多相关性很高的变量转化成少数几个彼此相互独立或不相关的变量。例如，有一份 6 个样本，4 个变量的数据

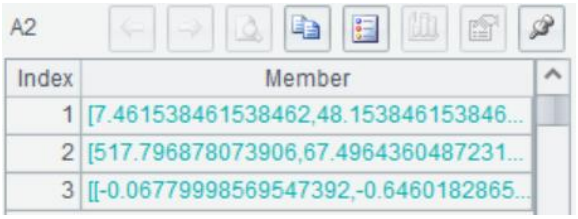
	x1	x2	x3	x4
1	7	26	6	60
2	1	29	15	52
3	11	56	8	20
4	11	31	8	47
5	7	52	6	33
6	11	55	9	22

对 x1,x2,x3,x4 降维，代码如下：

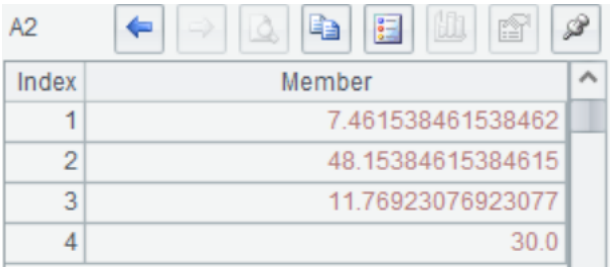
	A
1	[[7,26,6,60],[1,29,15,52],[11,56,8,20],[11,31,8,47],[7,52,6,33],[11,55,9,22],[3,71,17,6],[1,31,22,44],[2,54,18,22],[21,47,4,26],[1,40,23,34],[11,66,9,12],[10,68,8,12]]
2	=pca(A1)
3	=A2(2).((~)/sum(A2(2)))
4	=A3.(cum(~))
5	=pca@r(A1,2)

A1 矩阵的行对应样本，列对应变量

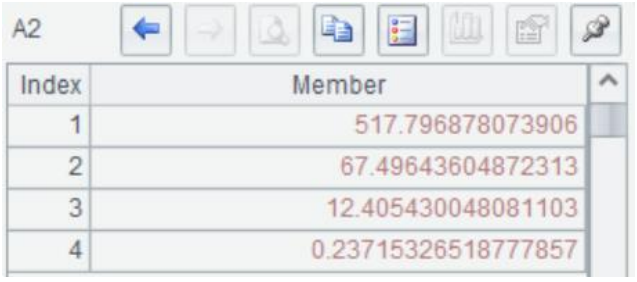
A2 对 A1 进行主成分分析，返回降维用的信息



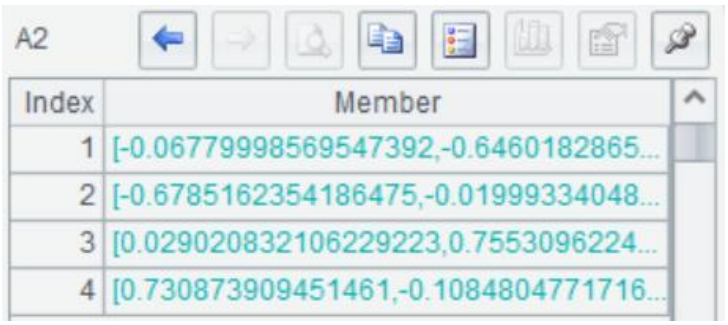
A2(1) 返回每一列的平均值



A2(2) 主成分方差，也就是各特征向量对应的特征值，从大到小进行排列



A1(3) 主成分系数，也就是样本协方差矩阵的特征向量矩阵



A3 计算四个特征值的信息贡献率

A3

Index	Member
1	0.8659738950184315
2	0.11288239481550631
3	0.020747090283889557
4	3.966198821725747E-4

A4 计算累计贡献率，可以看到前两个特征之和已经达到 97%以上，因此可以取前两个主成分。

A4

Index	Member
1	0.8659738950184315
2	0.9788562898339378
3	0.9996033801178273
4	0.9999999999999999

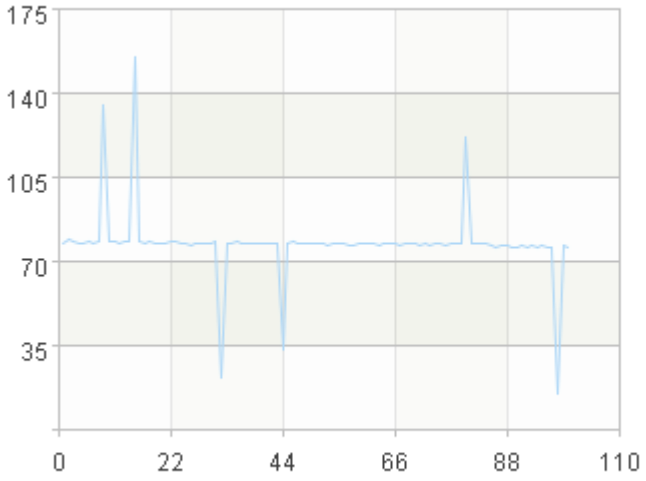
A5 对 A1 进行降维，n 值取 2，返回降维结果。

A5

Index	Member
1	[36.821825999449686,-6.87087815422737]
2	[29.60727342071096,4.610881963526308]
3	[-12.981775719737616,-4.204913183175939]
4	[23.71472572091802,-6.634052554708715]
5	[-0.5531916766245959,-4.4617321231786935]
6	[-10.812490833309818,-3.646571174544059]
7	[-32.58816660881793,8.979846284936055]
8	[22.606395499005586,10.725906457369453]
9	[-9.262587237675842,8.985373347478786]
10	[-3.283969329640679,-14.157277337500906]
11	[9.220031117829375,12.386080787220456]

sg 数据平滑(多项式平滑)

对于噪声比较多的数据通常需要先将数据平滑处理来去噪
比如，有一组含有噪声的数据如下图，使用 sg()函数可以将数据进行多项式平滑处理



	A
1	=file("sgdata.xlsx").xlsimport@w()
2	=A1.(~(1))
3	=A1.(~(2))
4	=sg(A3,1,15).conj()
5	=canvas()
6	=A5.plot("NumericAxis","name":"x")
7	=A5.plot("NumericAxis","name":"y","location":2)
8	=A5.plot("Line","markerStyle":0,"axis1":"x","data1":A2,"axis2":"y","data2":A3)
9	=A5.plot("Line","markerStyle":0,"lineColor ":-65536,"axis1":"x","data1":A2,"axis2":"y", "data2":A4)
10	=A5.draw@p(600,600)

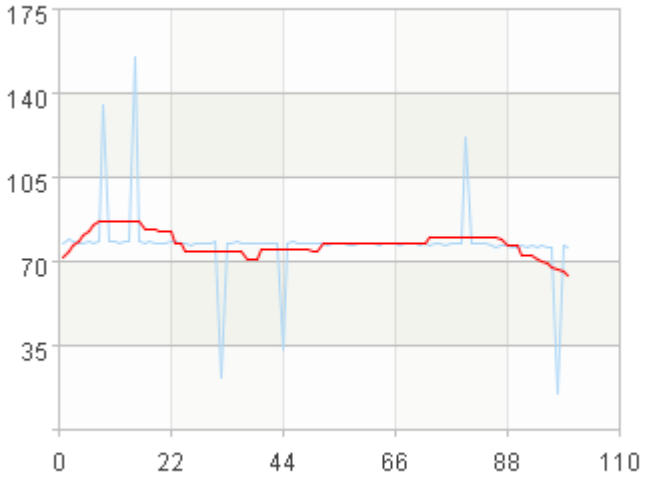
A1 将数据读成二级序列

A2 数据序号

A3 原始数据

A4 对 A3 数据进行 15 点 1 次平滑（n=1,m=15），返回平滑后数据，如图中红线。sg(A,n,m,d)中，参数 d 省略时不求导表示仅平滑；d=1，表示求一阶导数平滑结果，可消除常数项误差；d=2 二阶导数平滑，可消除一次项误差

A5-A10 将平滑前后数据画图，观察平滑效果。如效果不理想，可调整 n、m、d 的值，直至理想。



曲线拟合

最小二乘法线性拟合

最小二乘数据拟合的步骤为：

给定一组数据点(x1,y1)，(x2,y2)……(xm,ym)

（1）确定用于拟合数据的曲线类型，例如，y=a_1x+a_2

（2）将数据点代入曲线，得到系统 AX=Y

（3）使用 linefit()线性最小二乘法求解 AX=Y

例如，用最小二乘法拟合下表中的数据

x	19	25	31	38	44
y	19	32.3	49	73.3	97.8

SPL 代码如下：

	A
1	[19,25,31,38,44]
2	[19,32.3,49,73.3,97.8]
3	=canvas()
4	=A3.plot("NumericAxis","name":"x")
5	=A3.plot("NumericAxis","name":"y","location":2)

6	=A3.plot("Dot","lineWeight":0,"lineColor":-16776961,"markerWeight":1,"axis1":"x","data1":A1,"axis2":"y","data2": A2)
7	=A3.draw(800,400)
8	[[19,1],[25,1],[31,1],[38,1],[44,1]]
9	=linefit(A8,A2).conj()
10	=to(10,50)
11	=A10.([~,A9(1)*~+A9(2)])
12	=A3.plot("Line","markerStyle":0,"axis1":"x","data1":A11.(~(1)),"axis2":"y", "data2":A11.(~(2)))
13	=A3.draw(800,400)

A1-A7 导入数据，画出散点图，确定曲线类型

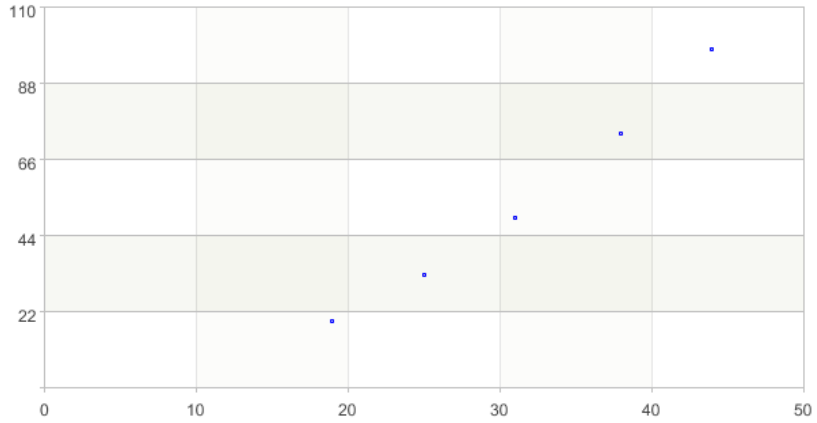
A3 生成画布

A4 绘制横轴”x”

A5 绘制纵轴”y”

A6 绘制点图元，x 轴数据取 A1，y 轴数据取 A2

A7 画图，观察点的分布呈直线型，选取拟合曲线类型为 y=a_1x+a_2



A8-A9 得到系统 AX=Y，并求解拟合曲线

将数据点带入曲线并写成矩阵形式，得到$\begin{bmatrix}19 & 1\\25 & 1\\31 & 1\\38 & 1\\44 & 1\end{bmatrix}\begin{bmatrix}a1\\a2\end{bmatrix}=\begin{bmatrix}19\\32.3\\49\\73.3\\97.8\end{bmatrix}$，即 AX=Y 形式。

将矩阵参数传入 linefit(A,Y)，求解出a_1，a_2的值，a_1=3.157452，a_2=-44.8639981

A9

Index	Member
1	3.157452
2	-44.863998

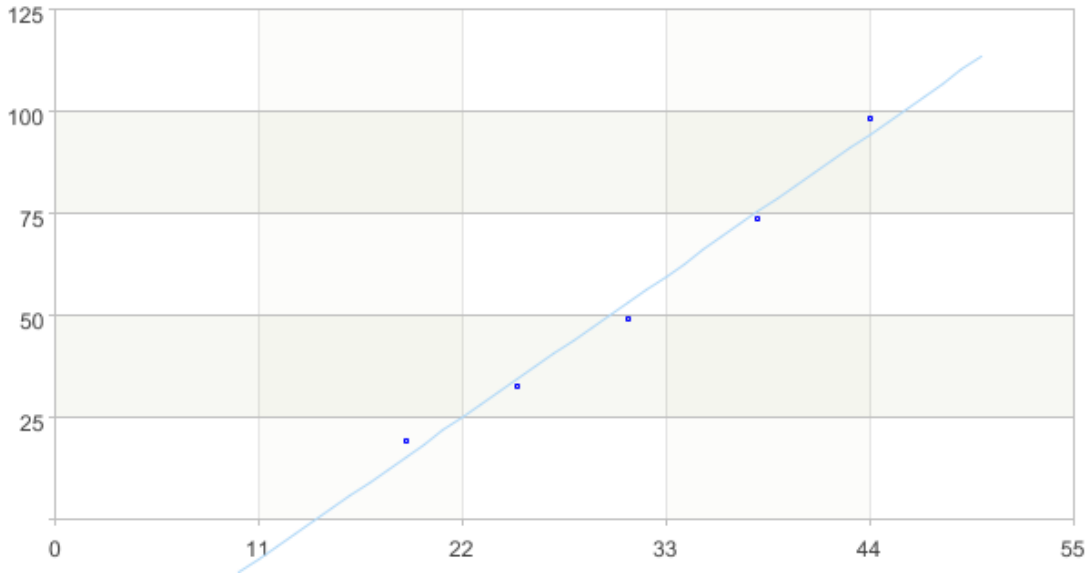
A10-A13 将拟合曲线画到散点图上，直观对比

A10 根据点的分布，选取 10-50 的区间作图

A11 区间内循环取数，生成 x 轴和 y 轴数据

A12 绘制线图元

A13 画图



可以用线性最小二乘法 linefit()拟合的曲线类型有多种，常用的曲线有：

（1） 直线 y=a_1x+a_2

（2） 多项式 y=a_1x^m+……+a_mx+a_{m+1}

（3） 双曲线（一支）y=$\frac{a_1}{x}$+a_2

（4） 指数曲线 y=$a_1e^{a_2x}$

对于双曲线和指数曲线，需做变量代换，转化为对a_1，a_2的线性函数

多项式拟合

对于多项式拟合，SPL 提供了现成的函数 polyfit()
例如还是上一小节中的数据，我们采用 y=a_1x^2+a_2x+a_3的形式拟合

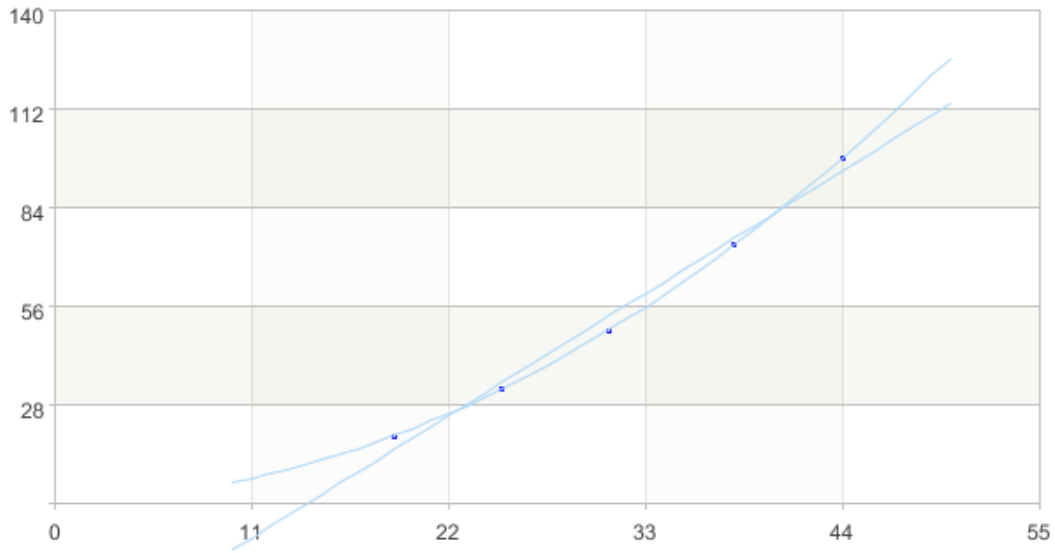
	A
……	……
14	=polyfit(A1,A2,2).conj()
15	=A10.([~,A14(3)*~*~+A14(2)*~+A14(1)])
16	=A3.plot("Line","markerStyle":0,"lineColor":-65536,"axis1":"x","data1":A15.(~(1)),"axis2":"y","data2":A15.(~(2)))
17	=A3.draw(800,400)

A14 将 x 和 y 数据传入函数 polyfit(X,Y,n)，n 为多项式次数，这里取 2，返回拟合系数

A14

Index	Member
1	0.688179
2	0.019301
3	0.049733

从下到上降次排列，即$a_1 = 0.049733$，$a_2 = 0.019301$，$a_3 = 0.688179$
A15-A17 将二次函数拟合曲线画图



偏最小二乘法拟合

偏最小二乘回归提供一种多对多线性回归建模方法，特别当两组变量的个数很多，且都存在多重相关性，而观测数据的样本量又较少时，用偏最小二乘法回归建立的模型具有传统的经

典回归分析等方法多没有的有点。

例如下图是关于体能训练的数据，被测变量分为两组。第一组是身体特征指标 X，包括体重（x1）、腰围（x2）、脉搏（x3）。第二组变量是训练结果指标 Y，包括单杠（y1）、弯曲（y2）、跳高（y3）。通过偏最小二乘法建立 Y 和 X 之间的回归模型。

NO.	Weight	Waist	Pulse	Horizontal bar	Bending	High jump
1	191	36	50	5	162	60
2	189	37	52	2	110	60
3	193	38	58	12	101	101
4	162	35	62	12	105	37
5	189	35	46	13	155	58
6	182	36	56	4	101	42
7	211	38	56	8	101	38
8	167	34	60	6	125	40
9	176	31	74	15	200	40
10	154	33	56	17	251	250
11	169	34	50	17	120	38
12	166	33	52	13	210	115
13	154	34	64	14	215	105
14	247	46	50	1	50	50
15	193	36	46	6	70	31
16	202	37	62	12	210	120
17	176	37	54	4	60	25
18	157	32	52	11	230	80
19	156	33	54	15	225	73

	A
1	=file("plsdata.xlsx").xlsimport@w()
2	=transpose(A1.to(2:)).to(2:)
3	=transpose(A2.to(:3))
4	=transpose(A2.to(4:))
5	=pls(A3,A4,3)
6	=transpose(pls(A3,A5))
7	=canvas()
8	=A7.plot("NumericAxis","name":"y_pre","title":"prediction data")
9	=A7.plot("NumericAxis","name":"y","location":2,"title":"observation data","titleAngle":270)
10	=A7.plot("Dot","lineWeight":0,"lineColor":-16776961,"markerWeight":1,"axis1":"y_pre","data1":A6(1),"axis2":"y","data2": A2(4))
11	=to(20)
12	=A7.plot("Line","markerStyle":0,"axis1":"y_pre","data1":A11,"axis2":"y","data2":A11)
13	=A7.plot("Legend","legendText":"Horizontal bar prediction","iconWidth":0)
14	=A7.draw(600,600)

A1 导入数据，读成二级序列

A2 去掉序号和标题，选出数据矩阵

A3 选出自变量数据 X

A4 选出因变量数据 Y

A5 采用偏最小二乘法进行拟合，返回拟合模型表达式系数

A5

Index	Member
1	[47.968412908226725,623.2817463113163,179.886...
2	[0.07884384006295048,0.7276599817135389,-0.537...
3	[-1.455842560448941,-17.38722056498598,0.23378...
4	[-0.018950019671607393,0.13931887620587113,-0....

比如数据的第一列表示：

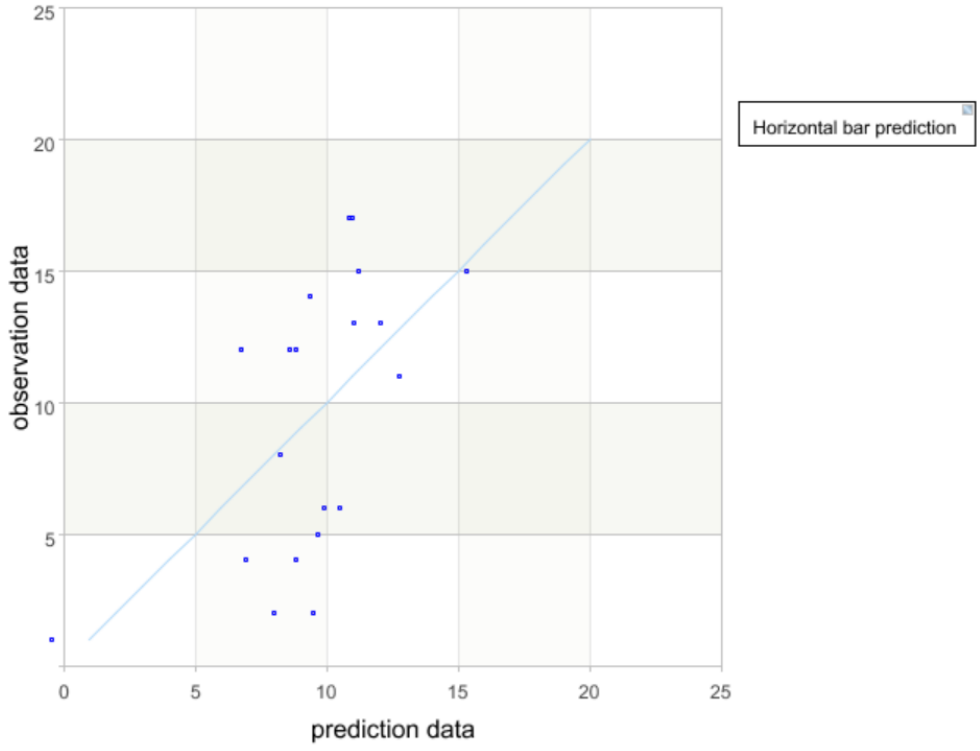
y1=47.9684+0.0788x1-1.4558x2-0.0190x3

A6 传入要预测的数据，和 A5 建好的模型，进行预测，返回预测结果。即 y1、y2、y3 的值。

A6

Index	Member
1	[9.669753200508026,8.018322920589966,6.764155...
2	[143.2908062894005,124.72690351339918,111.086...
3	[66.14118853024561,66.67351500848045,62.42426...

A7-A14 以单杠 y1 为例画图，评估模型效果。横轴表示每个样本单杠数据的预测值，纵轴表示观测数据真实值。如果所有点都能在对角线附近均匀分布，则方程的拟合值与原值差异很小，方程的拟合效果就好。同理可画出 y2 和 y3 的预测效果图。



A7 生成画布

A8 绘制横轴，预测数据

A9 绘制纵轴，观测数据真实值
A10 绘制点图元，横轴取单杠的预测值，纵轴取单杠的真实值
A11 取[1,20]的固定区间
A12 绘制线图元，y=x
A13 绘制图例
A14 画图

Lasso 回归

有一组给定的 x1，x2 和 y 的值如下表，用 lasso 回归拟合数据

x1	1.1	1.4	1.7	1.7	1.8	1.8	1.9	2.0	2.3	2.4
x2	1.1	1.5	1.8	1.7	1.9	1.8	1.8	2.1	2.4	2.5
y	16.3	16.8	19.2	18	19.5	20.9	21.1	20.9	20.3	22

代码如下：

	A
1	[[1.1,1.1],[1.4,1.5],[1.7,1.8],[1.7,1.7],[1.8,1.9],[1.8,1.8],[1.9,1.8],[2.0,2.1],[2.3,2.4],[2.4,2.5]]
2	[16.3,16.8,19.2,18,19.5,20.9,21.1,20.9,20.3,22]
3	=lasso(A1,A2,0.01,10000)
4	=lasso(A1,A3)
5	=canvas()
6	=A5.plot("NumericAxis","name":"y_pre","autoCalcValueRange":false,"autoRangeFromZero":false,"maxValue":25,"minValue":15,"title":" prediction data ")
7	=A5.plot("NumericAxis","name":"y","location":2,"autoCalcValueRange":false,"autoRangeFromZero":false,"maxValue":25,"minValue":15,"title":"observation data","titleAngle":270)
8	=A5.plot("Dot","lineWeight":0,"lineColor":-16776961,"markerWeight":1,"axis1":"y_pre","data1":A4.conj(),"axis2":"y","data2": A2)
9	=to(15:25)
10	=A5.plot("Line","markerStyle":0,"axis1":"y_pre","data1":A9,"axis2":"y","data2":A9)
11	=A5.draw(600,600)

A1 输入数据 x
A2 输入数据 y
A3 用 lasso 拟合数据，0.01 表示学习率，10000 表示迭代次数，返回模型信息。
A3(1)系数矩阵
A3(2)截距

A3

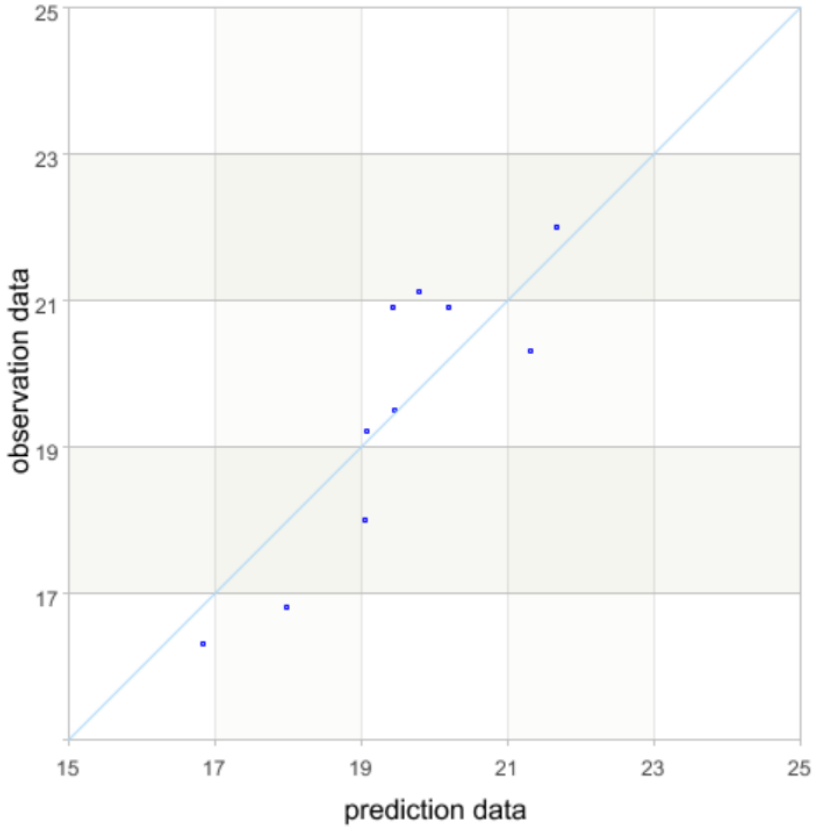
Index	Member
1	[[3.4818651254572965],[0.22088180...
2	12.77662618331654

A4 用 A3 拟合的模型在 A1 上进行预测，返回预测结果

A4

Index	Member
1	[16.84964780651063]
2	[17.982560066035482]
3	[19.093384145088415]
4	[19.0712959646165]
5	[19.463658838106 06]
6	[19.441570657634145]
7	[19.789757170179875]
8	[20.20420822414135]
9	[21.315032303194286]
10	[21.685306996211928]

A5-A11 画图对比预测值和真实值，评估模型效果。横轴表示每个样本数据的预测值，纵轴表示观测数据真实值。如果所有点都能在对角线附近均匀分布，则方程的拟合值与原值差异很小，方程的拟合效果就好。如效果不理想可通过调节学习率和迭代次数进行优化。



A5 生成画布

A6 绘制横轴，预测数据

A7 绘制纵轴，观测数据真实值

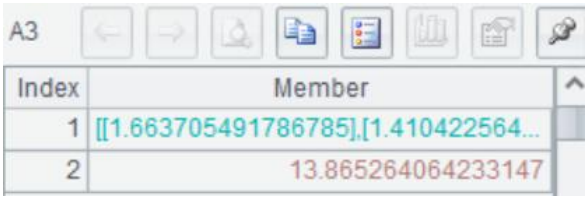
A8 绘制点图元，横轴取预测值，纵轴取真实值
A9 取[15,25]的固定区间
A10 绘制线图元，y=x
A11 画图

Rideg 回归

还是上一小节的样本数据，用岭回归进行拟合

	A
1	[[1.1,1.1],[1.4,1.5],[1.7,1.8],[1.7,1.7],[1.8,1.9],[1.8,1.8],[1.9,1.8],[2.0,2.1],[2.3,2.4],[2.4,2.5]]
2	[16.3,16.8,19.2,18,19.5,20.9,21.1,20.9,20.3,22]
3	=ridge(A1,A2,0.01,10000)
4	=ridge(A1,A3)
5	=canvas()
6	=A5.plot("NumericAxis","name":"y_pre","autoCalcValueRange":false,"autoRangeFromZero":false,"maxValue":25,"minValue":15,"title":"prediction data")
7	=A5.plot("NumericAxis","name":"y","location":2,"autoCalcValueRange":false,"autoRangeFromZero":false,"maxValue":25,"minValue":15,"title":"observation data","titleAngle":270)
8	=A5.plot("Dot","lineWeight":0,"lineColor":-16776961,"markerWeight":1,"axis1":"y_pre","data1":A4.conj(),"axis2":"y","data2": A2)
9	=to(15:25)
10	=A5.plot("Line","markerStyle":0,"axis1":"y_pre","data1":A9,"axis2":"y","data2":A9)
11	=A5.draw(600,600)

A1 输入数据 x
A2 输入数据 y
A3 用 ridge 拟合数据，0.01 表示学习率，10000 表示迭代次数，返回模型信息。
A3(1)系数矩阵
A3(2)截距

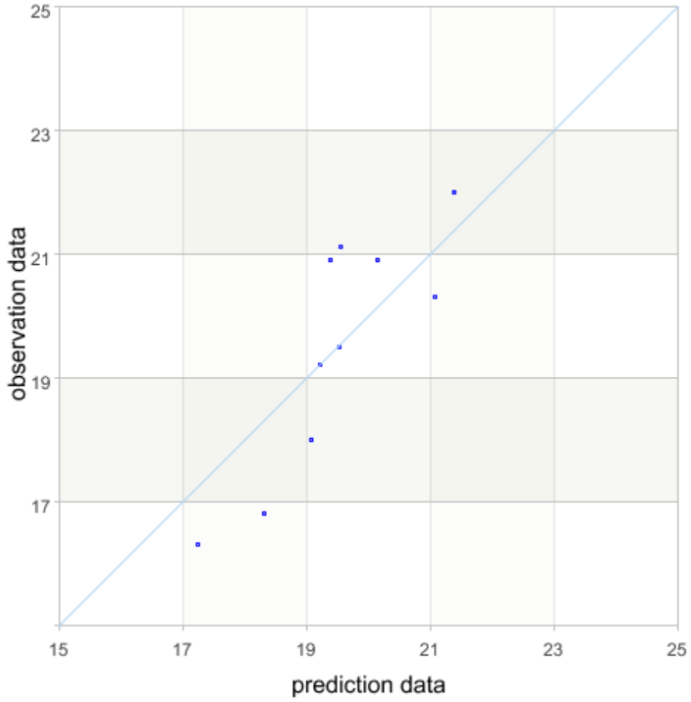


A4 用 A3 拟合的模型在 A1 上进行预测，返回预测结果

A4

Index	Member
1	[17.246804925879736]
2	[18.310085599118]
3	[19.232324015930708]
4	[19.09128175950515]
5	[19.539736821534945]
6	[19.398694565109388]
7	[19.565065114288064]
8	[20.154562432743415]
9	[21.076800849556122]
10	[21.38421365516036]

A5-A11 画图对比预测值和真实值，评估模型效果。横轴表示每个样本数据的预测值，纵轴表示观测数据真实值。如果所有点都能在对角线附近均匀分布，则方程的拟合值与原值差异很小，方程的拟合效果就好。如效果不理想可通过调节学习率和迭代次数进行优化。



A5 生成画布
A6 绘制横轴，预测数据
A7 绘制纵轴，观测数据真实值
A8 绘制点图元，横轴取预测值，纵轴取真实值
A9 取[15,25]的固定区间
A10 绘制线图元，y=x
A11 画图

Elastic Net 回归

继续使用上 lasso 回归里的样本数据，用弹性网络回归进行拟合

	A

1	[[1.1,1.1],[1.4,1.5],[1.7,1.8],[1.7,1.7],[1.8,1.9],[1.8,1.8],[1.9,1.8],[2.0,2.1],[2.3,2.4],[2.4,2.5]]
2	[16.3,16.8,19.2,18,19.5,20.9,21.1,20.9,20.3,22]
3	=elasticnet(A1,A2,0.01,10000,0.9,0.1)
4	=elasticnet(A1,A3)
5	=canvas()
6	=A5.plot("NumericAxis","name":"y_pre","autoCalcValueRange":false,"autoRangeFromZero":false,"maxValue":25,"minValue":15,"title":"prediction data")
7	=A5.plot("NumericAxis","name":"y","location":2,"autoCalcValueRange":false,"autoRangeFromZero":false,"maxValue":25,"minValue":15,"title":"observation data","titleAngle":270)
8	=A5.plot("Dot","lineWeight":0,"lineColor":-16776961,"markerWeight":1,"axis1":"y_pre","data1":A4.conj(),"axis2":"y","data2": A2)
9	=to(15:25)
10	=A5.plot("Line","markerStyle":0,"axis1":"y_pre","data1":A9,"axis2":"y","data2":A9)
11	=A5.draw(600,600)

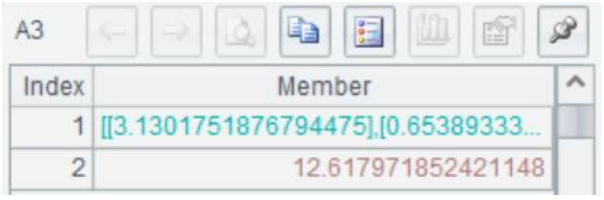
A1 输入数据 x

A2 输入数据 y

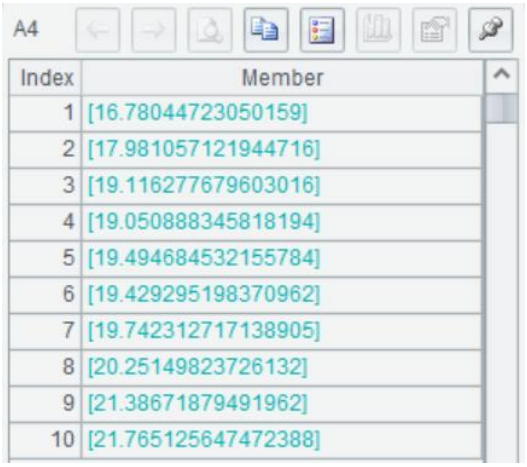
A3 用 elasticnet 拟合数据，0.01 表示学习率，10000 表示迭代次数，0.9 表示 L1 系数，0.1 表示 L2 系数，返回模型信息。

A3(1)系数矩阵

A3(2)截距

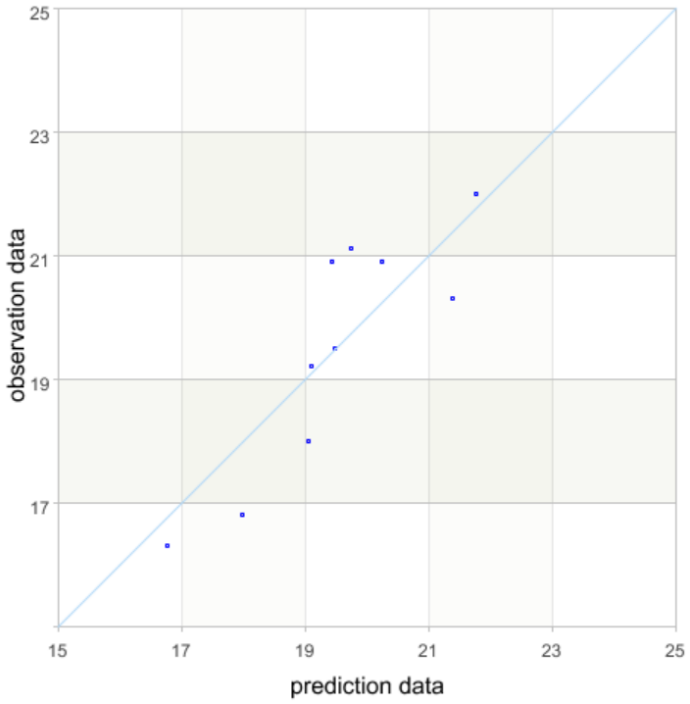


A4 用 A3 拟合的模型在 A1 上进行预测，返回预测结果



A5-A11 画图对比预测值和真实值，评估模型效果。横轴表示每个样本数据的预测值，纵轴表示观测数据真实值。如果所有点都能在对角线附近均匀分布，则方程的拟合值与原值差异

很小，方程的拟合效果就好。如效果不理想可通过调节学习率和迭代次数进行优化。



A5 生成画布
A6 绘制横轴，预测数据
A7 绘制纵轴，观测数据真实值
A8 绘制点图元，横轴取预测值，纵轴取真实值
A9 取[15,25]的固定区间
A10 绘制线图元，y=x
A11 画图

误差评估：

MSE, RMSE, MAE, MAPE, R^2

Excel 文件里有两列数据，分别是某地区房屋价格的预测值和真实值，对其进行误差评估

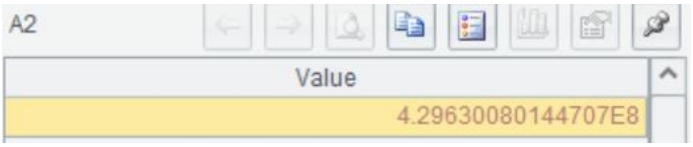
SalePrice_predictvalue	SalePrice
205578.0852	208500
176323.4192	181500
217021.2719	223500
161173.8857	140000
281214.8513	250000
…	…

	A
1	=T("houseprice_result.xls")

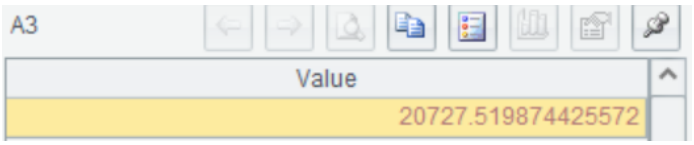
2	=mse(A1.(SalePrice_predictvalue),A1.(SalePrice))
3	=sqrt(A2)
4	=mae(A1.(SalePrice_predictvalue),A1.(SalePrice))
5	=A1.(abs((SalePrice_predictvalue-SalePrice)*100/SalePrice)).avg()
6	=1-A2/var(A1.(SalePrice))

A1 将文件读为序表

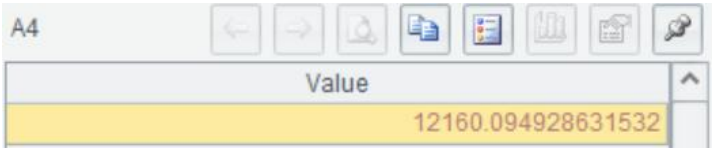
A2 计算 MSE，均方误差



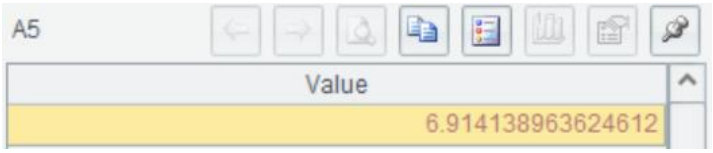
A3 计算 RMSE，均方根误差



A4 计算 MAE，平均绝对误差



A5 计算 MAPE，平均绝对百分比误差



A6 计算 R^2，拟合优度