1 奇异值分解

奇异值分解将数据矩阵 $\mathbf{X}_{n\times p}$ （秩为 $r \le min(n,p)$）分解为

$$X_{n\times p} = U_{n\times r} D_{r\times r} V_{r\times p}^{\top}, \quad U^{\top}U = I_r, \quad V^{\top}V = I_r, \quad D = diag(\sqrt{\lambda_1},..., \sqrt{\lambda_r}), \tag{1}$$

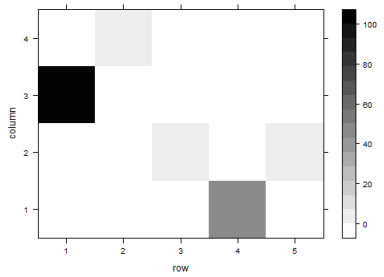
其中 $\lambda_1 \ge ... \ge \lambda_r > 0$ 为 $X^{\top}X$ 或 $XX^{\top}$ 的非 0 特征根, V 的第 k 个列向量是对应 λ_k 的 $X^{\top}X$ 的特征向量(刻画 X 的行向量)，U 的第 k 个列向量是 $XX^{\top}$ 对应 λ_k 的特征向量（刻画 X 的列向量)。R 中的奇异值分解函数 svd 的调用方式为：

```
svd( x, nu = min(n, p), nv = min(n, p) )
 # x: any n by p matrix
```

其中 x 是数据矩阵，svd 函数输出三部分: "d","u","v", 分别对应于公式 (1) 中的 D,U,V，选项中的 nu,nv 分别指定需要输出的 u,v 的列数（通常我们并不需要完整的 D,U,V），缺省值为 $\min(n,p)$。如果指定 nu $=r$, nv $=r$ (r=rank(x)), 就是公式 (1) 的情况，即完整的 SVD 分解。

例 1 (svd). 我们知道，奇异值分解的特征向量（U,V 的列）分别刻画矩阵的列向量特征和行向量特征，但具体情况则通常难以详细描述。对于下面的简单矩阵 A，我们考察其其奇异值分解。

```
> x=c(0,100,0,0,1,0,0,0,0,0,1,0,0,0,0,50,0,0,1,0)
> ( A=matrix(x,4,5) )
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    1    0    0    0
[2,]  100    0    0    0    0
[3,]    0    0    1    0    1
[4,]    0    0    0   50    0
## 用 levelplot 画格子点（不连续的）热图
library(lattice)
levelplot(A,col.regions = gray((10:1)/10) )
```



A 是一个 4×5 矩阵，大部分元素为 0 或 1，有两个较大的元素 100，50。按列来看，100 位于第一列的 4×1 向量的第 2 个位置，50 位于第 4 个位置，我们认为列向量的第 2、4 位置是比较重要的。下面观察奇异值分解的特征向量:

```
> svd(A,nu=2,nv=2)
$d
[1] 100.000000  50.000000   1.414214   1.000000
$u                                  $v
     [,1] [,2]                           [,1] [,2]
[1,]    0    0                      [1,]    1    0
[2,]    1    0                      [2,]    0    0
[3,]    0    0                      [3,]    0    0
[4,]    0    1                      [4,]    0    1
                                    [5,]    0    0
```

A 的奇异值分解的第一个（最重要的）u,v 特征向量是

$$\mathbf{u}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$\mathbf{u}_1$ 第 2 个位置非 0，其它全为 0，说明 A 的列向量最重要的特征是 “第 2 位置最大，其它为 0 或较小”，这正描述了数字 100 在列向量中的位置。$\mathbf{v}_1$ 说明 A 的行向量最重要的特征是第 1 个位置，这正是数字 100 在行向量中的位置。因此 $\mathbf{u}_1, \mathbf{v}_1$ 分别刻画了矩阵 A 的最主要的列向量和行向量特征，两者一起决定了 “$(2,1)$ 元素是 A 的最重要的元素”。类似地，第 2 特征向量 $\mathbf{u}_2 = (0,0,0,1)^\top$, $\mathbf{v}_2 = (0,0,0,1,0)^\top$ 刻画了 A 的第二重要特征 (即 50 所在位置)。

练习 1. 截取 A 奇异值分解的前两阶，即 $A_2 = \sqrt{\lambda_1}\mathbf{u}_1\mathbf{v}_1^\top + \sqrt{\lambda_2}\mathbf{u}_2\mathbf{v}_2^\top$, 画出 A_2 的热图.

练习 2（图像压缩）. 数据集 monalisa.txt **是一个** 18x29 **的黑白图像数据。**

```
monalisa=read.table("http://staff.ustc.edu.cn/~ynyang/vector/data/monalisa.txt")
monalisa=as.matrix( monalisa)
#  连续热图:   image()
image(monalisa, col=gray.colors(256),axes=F )
```

求数据集 monalisa **的** 5 **阶** SVD **逼近，画出热图，**

例 2 (奇异值分解 SVD **与主成分分析** PCA). 当矩阵 X 是中心化矩阵时，PCA⇔SVD。假设中心化矩阵 X 的奇异值分解为 $X = UDV^\top$, 则 V 是 PCA 的载荷矩阵，$Y = XV = UD$ 为所有主成分 (y_{ik} 为第 i 个样本点的第 k 个主成分)。因此, 主成分分析也可由 svd 函数完成，PC 散点图的横轴为第一主成分，即 $XV = UD$ 的第一列；纵轴为第二主成分，即 UD 的第二列。利用奇异值分解做 PCA 与 R 的专用函数 princomp，prcomp 结果相同，后者会提供方差贡献率等内容，而奇异值分解不提供（但从给出的奇异值很容易计算得到）。奇异值分解做 PCA 的优势是计算效率较高（特别是对于大型数据），更高效的 SVD 函数还有：

```
rsvd (in package:rsvd),  fastsvd (in package: Bootsvd)
```

练习 3 （基因数据主成分分析）. 数据集

`http://staff.ustc.edu.cn/~ynyang/vector/data/genedata1.txt`

包含 $n = 800$ 个人的 $p = 1000$ 个基因的数据，第一列为每个人的种族（race: EUR **欧洲，**EAS **东亚，**AMR **美洲，**SAS **南亚，**AFR **非洲），**2-1001 **列为基因数据（取值** 0,1,2**）。应用** svd **函数求前两个主成分，画出其散点图，并把不同的种群** race **标记不同的颜色。各个种群在二维图上是否能较好地分开？（参见第** 12 **讲例** 1**）**

例 3 (eigenface). SVD 在图像处理中的一个应用是所谓 eigenface（特征脸）的求解。该方法将每一个图像的像素灰度矩阵按列排成向量 (不再像练习 2 那样当作矩阵)，比如 92×112 像素的黑白图像排成长度为 10304(=92*112) 的向量。数据集 faces40.txt 是 40 个人的 92*112 图像数据，该数据有 40 行 (40 人), 10304 列 (代表 10304 个像素).

对该矩阵进行 SVD 分解，得到的 d,u,v 三部分, 其中 v 称为 eigenfaces, 是特征脸谱或代表性的脸谱（特征向量或基）。特征脸谱加权平均即可得到各个人脸图像的逼近。

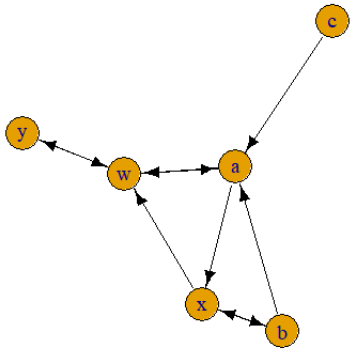
```
> faces40 = read.table("http://staff.ustc.edu.cn/~ynyang/vector/data/faces40.txt")
> dim(faces40)
[1]    40 10304
> faces40=as.matrix(faces40)
> face1 = faces40[1,]  # 第一个人的图像数据（向量）
> face1=matrix(face1, 92,112) # 向量还原成 92x112 像素矩阵
> image(face1,  col=gray((1:256)/256), axes=F)  # 画图

> svd(faces40)->faces40.svd
> d=faces40.svd[[1]] #40x1
> u=faces40.svd[[2]] #40x 40
> v=faces40.svd[[3]] #10304x 40

> eigenface1 =v[,1] # 第一个 eigenface/特征向量，长度为 10304
> eigenface1=matrix( eigenface1, 92,112) # 向量还原成 92x112 像素矩阵
> image(eigenface1, col=gray((1:256)/256), axes=F) # 第一个 eigenface
> image(matrix(v[,2],92,112), col=gray((1:256)/256), axes=F) # 第二个 eigenface
```

例 4 (**网页排序**) . 网页排序算法与奇异值分解有关。假设 n 个网页，定义所有网页的权威性 (authority) 指标为 $\mathbf{x} = (x_1,...,x_n)^\top$, 一个网页的权威性指标越大，越容易被别的网页链接。定义所有网页的中心性 (hub) 指标为 $\mathbf{y} = (y_1,...,y_n)^\top$, 一个网页的中心性指标越大，它越容易连接到其它网页。假设网页之间的邻接矩阵为 $n \times n$ 矩阵 A, 其 (i,j) 元 $a_{ij} = 1$, 若网页 i 链接到网页 j, 否则 $a_{ij} = 0$. HITS 模型假设

$$\mathbf{y} = cA\mathbf{x}, \quad \mathbf{x} = dA^\top \mathbf{y}.$$

HITS 算法使用迭代方法，每次迭代时可以适当地将 $\mathbf{x}, \mathbf{y}$ 单位化，直到迭代收敛:

```
x=x.ini=rep(1,n)  #x 初值
repeat{
 y=A%*%x;  y=y/sqrt(sum(y^2))
 x.new=t(A)%*%y; x.new=x.new/sqrt(sum(x.new^2))
 error=sum((x.new-x)^2)
 if (error<1e-5) break
 x=x.new
}
```



练习 4. 上图网络中两个网页之间的单箭头表示有链接（例如 b 连接到 a），双箭头表示相互有链接；(a) 写出邻接矩阵 A, 应用 HITS 迭代算法求各个网页的权威性指标并对网页排序; (b) 检查 HITS 算法收敛得到的权威性指标向量是否等于 $A^\top A$ 的最大特征向量（注意可能相差一个正负号）。

2 典则相关分析（CCA）

典则相关分析的 R 函数 cancor 只能处理样本 CCA，而不能处理仅有方差协方差矩阵的情况（即总体 CCA），cancor 也不计算每个样本的典则变量。我们使用自编函数 cca()。

```
#读入cca函数
source("http://staff.ustc.edu.cn/~ynyang/vector/lab/cca.txt")
#调用方法:
cca(covmat=S,xlab=1:2) #总体cca, covmat为协方差矩阵
#xlab=1:2指定S的第一二行或列为x, 其它为y
cca(x,y) #样本cca, x,y: data frame
```

例 5 (总体 CCA，第 16 讲例 1) . 四门考试成绩 $(x_1, x_2, y_1, y_2)^\top$ 的相关系数如下

$$R = \begin{pmatrix} 1.00 & 0.63 & 0.24 & 0.06 \\ & 1.00 & -0.06 & 0.07 \\ & & 1.00 & 0.42 \\ & & & 1.00 \end{pmatrix}$$

其中 x_1, x_2, y_1, y_2 分别是阅读速度、阅读能力、运算速度、运算能力的测试成绩。CCA 考察阅读 $\mathbf{x} = (x_1, x_2)^\top$ 与算数 $\mathbf{y} = (y_1, y_2)$ 的相关关系。

```
## 第 16 讲例 1（总体 cca）
tmp=c(0.63, 0.24,0.06, -0.06, 0.07, 0.42)
s =matrix(0, 4,4)
s[lower.tri(s)]=tmp
s=s+t(s)
diag(s)=1
```

```
cca(covmat=s,xlab=1:2)
$cor.canonical
[1] 0.39 0.07 # 第一和第二典则相关系数

$xcoef  # x 的第一和第二典则变量的系数
        canon_v1  canon_v2
        1 -1.247798 0.3179603
        2  1.033039 0.7687192

$ycoef  # y 的第一和第二典则变量的系数
        canon_v1     canon_v2
        3 -1.1018763 -0.007089979
        4  0.4563537  1.002957091
```

练习 5. 求出第 16 讲例 3 的三对典则变量，并画出散点图。

```
#######
# 第 16 讲例 3:（样本 cca）

read.table("http://staff.ustc.edu.cn/~ynyang/vector/data/T9-12.DAT")->X
X=as.matrix(X)
colnames(X)=c("sale_growth", "sale_profit","sale_newacc","creativity","m_reasoning","a_reasoning","math")
apply(X,2, mean)->m
t(X)-m->X
X=t(X)
y=X[,1:3]
x=X[,4:7]

mycca=cca(x,y)
xcca=mycca$xcca # 三列分别是 50 个人的 x 的 3 个典则变量
ycca =mycca$ycca # 三列分别是 50 个人的 y 的 3 个典则变量
```

3 对应分析（CA：correspondence analysis）

假设 $p \times q$ 矩阵 W 是一个列联表，其各行代表某个属性变量的各个类别 (category)，各列代表第二个属性的各个类别。记行和，列和分别为

$$\mathbf{r} = W\mathbf{1}_q, \quad \mathbf{c} = W^\top \mathbf{1}_p$$

以及以它们为对角元的对角阵

$$D_r = \mathrm{diag}(\mathbf{r}), \quad D_c = \mathrm{diag}(\mathbf{c})$$

X 的 “中心标准化” 矩阵定义为：

$$Z = D_r^{-1/2}\left(W - \mathbf{r}\mathbf{c}^\top/n\right)D_c^{-1/2} = D_r^{-1/2} W_c D_c^{-1/2}$$

其中 $n = \sum w_{ij}$, 对应分析对 Z 进行奇异值分解，

$$Z = UDV^\top$$

并令

$$F = D_r^{-1/2} UD, \quad G = D_c^{-1/2} VD$$

对应分析的双标图同时画 $F[,1:2]$ 和 $G[,1:2]$。互相靠近的 F 坐标点表示行类别接近，互相靠近的 G 坐标点表示列类别接近，互相靠近的 F 和 G 坐标点表示响应的行类别与列类别关联性较强。

例 6 (对应分析). 对应分析研究计数矩阵（称为列联表）的行与列之间的对应关系. 列联表各行代表某个属性变量的各个类别 (category)，各列代表第二个属性的各个类别。比如下表是考古数据的统计表，各行代表 6 个考古地点 (site1-6)，不同的地点可能代表不同的时期，各列代表 7 种燧石类型 (A-G)。

	a	b	c	d	e	f	g
1	20	3	4	42	18	0	13
2	85	3	12	0	0	0	0
3	26	40	8	0	0	26	0
4	20	1	4	13	58	0	4
5	67	10	23	0	0	0	0
6	26	29	8	3	0	33	1

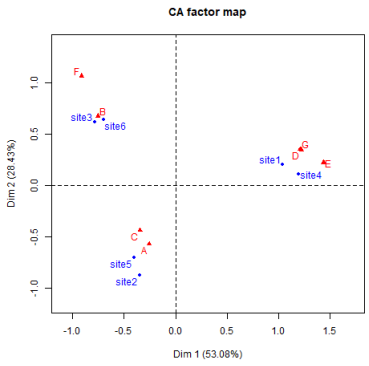
我们感兴趣的问题包括

- 哪些考古地点类似？哪些不同？这是行之间比较的问题，但比较之前，需要将每一行归一化（除非各行总和相同），即每行的各个计数除以行的总数。
- 哪些燧石类型是类似的？这是列比较的问题，但比较之前需要列归一化，即每列的元素都出一该列的总数。
- 每个地点主要有哪些类型的燧石，或者每种燧石主要产出于那个考古地点？这是行列之间的关联问题。
- 对应分析给出的主坐标刻画了地点之间、燧石之间的距离或相近关系，所以主坐标特别是第一主坐标可能包含了年代次序信息，这具有重要的考古意义。

R 软件包 FactoMineR 提供的 CA 函数用来对列联表数据进行对应分析。**R** 命令：

```
install.packages("FactoMineR")
library(FactoMineR)
myca=CA(X) #X: 列联表
```

对考古数据应用 CA 函数，得到下图所示的双标图可以看到 site1、4 相近（相似），D、G、E 相似. 而 site1、4 与 D、G、E 靠近，说明 site1,4 出土 DGE 较多。



练习 6. 利用第一主坐标对燧石和考古地点的时间排序。

练习 7. bac 数据集 (http://staff.ustc.edu.cn/~ynyang/vector/data/bac.xls) 包含 1976 年法国 22 个大区的 202100 个毕业大学生的专业统计如下 (专业以 A-H 表示)：

	A	B	C	D	E	F	G	H
ILDF	9724	5650	8679	9432	839	3353	5355	83
CHAM	924	464	567	984	132	423	736	12
PICA	1081	490	830	1222	118	410	743	13
HNOR	1135	587	686	904	83	629	813	13
CENT	1482	667	1020	1535	173	629	989	26
BNOR	1033	509	553	1063	100	433	742	13
BOUR	1272	527	861	1116	219	769	1232	13
NOPC	2549	1141	2164	2752	587	1660	1951	41
LORR	1828	681	1364	1741	302	1289	1683	15
ALSA	1076	443	880	1121	145	917	1091	15
FRAC	827	333	481	892	137	451	618	18
PAYL	2213	809	1439	2623	269	990	1783	14
BRET	2158	1271	1633	2352	350	950	1509	22
PCHA	1358	503	639	1377	164	495	959	10
AQUI	2757	873	1466	2296	215	789	1459	17
MIDI	2493	1120	1494	2329	254	855	1565	28
LIMO	551	297	386	663	67	334	378	12
RHOA	3951	2127	3218	4743	545	2072	3018	36
AUVE	1066	579	724	1239	126	476	649	12
LARO	1844	816	1154	1839	156	469	993	16
PROV	3944	1645	2415	3616	343	1236	2404	22
CORS	327	31	85	178	9	27	79	0

```
A Philosophy-Letters,
B Economics and Social Sciences,
C Mathematics and Physics,
D Mathematics and Natural Sciences,
E Mathematics and Techniques,
F Industrial Techniques,
G Economic Techniques,
H Computer Techniques.
```

22 个本土大区列表如下：

```
NOPC:  Nord Pas-de-Calais: 北部-加莱海峡
PICA:  Picardie: 皮卡第
HNOR: Normandie: 诺曼底
ILDF: Ile de France: 法兰西岛
CHAM: Champagne Ardenne: 香槟-阿登
LORR: Lorraine: 洛林
ALSA: Alsace: 阿尔萨斯
BRET: Bretagne: 布列塔尼
PAYL: Pays de loire: 卢瓦尔河地区
CENT: Centre Val de loire: 中央－卢瓦河山谷
BOUR: Bourgogne: 勃艮第
FRAC: Franche comté: 弗朗什孔泰
PCHA: Poitou Charentes: 普瓦图-夏朗德
LIMO: Limousin: 利穆赞
AUVE: Auvergne: 奥弗涅
RHOA: Rhone Alpes: 罗讷-阿尔卑斯
Alpes Pays de savoie: 阿尔卑斯－萨瓦地区
AQUI: Aquitaine: 阿基坦
MIDI: Midi pyrénées: 南部-比利牛斯
PROV: Provence Alpes Cote d'azur: 普罗旺斯－阿尔卑斯南部，蓝色海岸
LARO: Languedoc Roussilon: 朗格多克-鲁西永
CORS: corse: 科西嘉
```

试对 bac 数据做对应分析（必要时删除 CORS 岛的数据），根据得到的双标图，回答问题：

1. 根据双标图分析专业分布的特点。
2. 巴黎大区及其附近有什么专业特点？
3. 法国南部有什么特点？
4. 法国东北部有什么特点？

4 (选) 结构方程模型 SEM

R 程序包 lavvan 和 sem 的提供了结构方程模型的分析工具。lavaan 语法简洁，但不提供路径图的画法；sem 语法较为复杂，但提供了路径图的绘制函数。

例 **7.** 我们以第十五讲例 1 的确认性因子分析 (CFA) 为例，学习使用 R 程序包 lavvan 和 sem 的用法。该数据是 220 个学生 6 门课程成绩的的相关系数矩阵，如下：

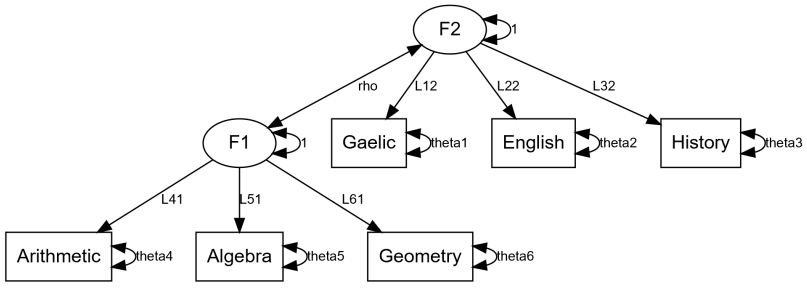
$$\mathbf{R} = \begin{array}{c} \begin{array}{cccccc} \text{Gaelic} & \text{English} & \text{History} & \text{Arithmetic} & \text{Algebra} & \text{Geometry} \end{array} \\ \begin{bmatrix} 1.0 & .439 & .410 & .288 & .329 & .248 \\ & 1.0 & .351 & .354 & .320 & .329 \\ & & 1.0 & .164 & .190 & .181 \\ & & & 1.0 & .595 & .470 \\ & & & & 1.0 & .464 \\ & & & & & 1.0 \end{bmatrix} \end{array}$$

```
      #相关系数矩阵 r:
r=matrix(0,6,6)
r[lower.tri(r)]=c(0.439,0.410,0.288,0.329,0.248,
0.351, 0.354,0.320,0.329, 0.164,0.190,0.181,0.595,0.470,0.464)
r= r+t(r)
diag(r)=1
rownames(r)=colnames(r)=c("Gaelic","English","History",
"Arithmetic","Algebra","Geometry")
```

假设如下结构方程模型（确认性因子分析模型）：

$$\begin{aligned} Gaelic &= L_{12}F_2 + \epsilon_1 \\ English &= L_{22}F_2 + \epsilon_2 \\ History &= L_{32}F_2 + \epsilon_3 \qquad \text{其中} \epsilon_i, i = 1,...,6 \text{独立}, var(\epsilon_i) = \theta_i \\ Arithmetic &= L_{41}F_1 + \epsilon_4 \qquad \epsilon's \text{与} (F_1, F_2) \text{独立} \\ Algebra &= L_{51}F_1 + \epsilon_5 \qquad var(F_1) = var(F_2) = 1, cov(F_1, F_2) = \rho \\ Geometry &= L_{61}F_1 + \epsilon_6 \end{aligned}$$

该模型以如下路径图表示 (绘图工具参见最后一页)：



4.1 lavaan 程序包

在 lavaan 中, 路径图模型结构以两个单引号 之间的一串命令指定，其中因子与显变量的关系以 =~ 指定，相关关系以 ~~ 指定，回归关系以 ~ 指定 (参见下面代码中的 mymodel='...')。结构方差模型的主要函数是 sem()

```
                    ### MODEL SPECIFICATION:
> mymodel= '
 F=~variable1+variable2+... ##definition of factors
 F1~~F2 ## F1 and F2 are correlated
 F1~F2 # F1=a+b*F2+error ##regression
                    '
 ### FIT THE SEM MODEL SPECIFIED BY mymodel
> sem(model = mymodel, data = , sample.cov = , sample.nobs =, ...)
 # model:model structure, data:original data,
 # sample.cov:covariance matrix or correlation matrix,
 # sample.nobs: sample size n
```

使用 lavann 程序包分析例 3 数据的代码如下：

```
        ## 例 3
> library(lavaan)
        # 指定模型:
> model = '
        # latent variable definitions
        F1 =~ Arithmetic+Algebra+Geometry # “=~”read as “is manifested by”
        F2 =~ Gaelic+English+History
        # residual correlations （方差，协方差）
        F1~~F2  # F1 and F2 are correlated
        '
> fit <- sem(model=model, sample.cov=r, sample.nobs=220, std.lv=TRUE )
        #sample.cov 指定协方差矩阵或相关系数矩阵，同时需要指定 sample.nobs 样本量
        # 也可以用 data = .. 指定原始数据
        #std.lv=T 因子的方差假设为 1
        > summary(fit)
```

练习 8. 例 3 中假设 F_1, F_2 是相关的, 试修改上述代码中 model 指定的最后一行的相关性假设修改为回归关系：$F_1 = \rho F_2 + error$ ，并运行 sem 分析.

4.2 sem 程序包

sem 程序包同样需要先指定模型结构，函数为 specifyModel()，结构方程分析的主要函数和 lavaan 一样也是 sem()（注意：在同一个 R 环境中如果同时加载 lavaan 和 sem 程序包，主函数 sem() 会相互冲突）。路径图的绘制函数为 pathDiagram()。细节如下：

```
### MODEL SPECIFICATION:
> mod=c("f1 -> variable1, parameter, initial.value",...)
> mymodel= specifyModel(text=mod)
> mysem = sem(model=mymodel, S=r, N= 220)
 #S: covariance or correlation matrix, N: sample size
> pathDiagram(mysem) # draw the path diagram
```

在 sem 程序包中，上页的路径图所代表的模型如下指定：

```
library(sem)   # 载入 sem
mod <- c(  "F2   -> Gaelic, L12, NA",
 # 表示 Gaelic = L12*F2+error, 最后一个是 L12 的初值，这里 NA 表示不设初值
 "F2   -> English, L22, NA",
 "F2   -> History, L32, NA",
 "F1 -> Arithmetic, L41, NA",
 "F1 -> Algebra, L51, NA",
 "F1 -> Geometry, L61, NA",
 "F2   <-> F1, rho, NA", #CFA
 "F1   <-> F1, NA, 1",  # 表示 F1 的方差已知，为 1
 "F2   <-> F2, NA, 1",
 "Gaelic     <-> Gaelic, theta1, NA", #Gaelic 的方差为 theta1, 不赋初值 (NA)
 "English    <-> English, theta2, NA",
 "History    <-> History, theta3, NA",
 "Arithmetic <-> Arithmetic, theta4, NA",
 "Algebra    <-> Algebra, theta5, NA",
 "Geometry   <-> Geometry, theta6, NA")
```

其中每一项描述路径图中的一个单向箭头或双向箭头及其相关的参数, 包含逗号分隔的 3 各部分，分别表示箭头、参数和参数初值：

- 箭头：单箭头 $A \to B$ 表示回归方程 $B = \beta A + error$。双箭头 $A \leftrightarrow B$ 表示 A,B 相关。箭头后面第一个位置为变量 A，第二个位置为参数 β，第三个位置为参数 β 的初值（NA 表示不赋初值）
- 参数：当箭头是单向时，参数为载荷/回归系数 (β), 系数/载荷 β 会标在箭头附近；当箭头是双向时，参数为协方差（$A \leftrightarrow B$）或方差 ($A \leftrightarrow A$), 协方差或方差会标记在双箭头附近。

使用 sem 程序包画出例 3 的路径图，代码如下：

```
# 描述/指定模型:
> mymodel =specifyModel(text=mod)
# 运行 sem
> mysem <- sem(model=mymodel, S=r, N= 220)
# 画路径图:
> pathDiagram(mysem,ignore.double=FALSE,edge.labels="both",rank.direction="TB")
 #ignore.double: 是否不画双向箭头，箭头标号 edge.labels 可选"name","value","both";
 #rank.direction 可选"TB" (从上到下）或"LR"（从左到右）
```