

结构方程模型 (sem) 的 R 程序包

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 1. 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 门课程成绩的相关系数矩阵，如下：

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

```

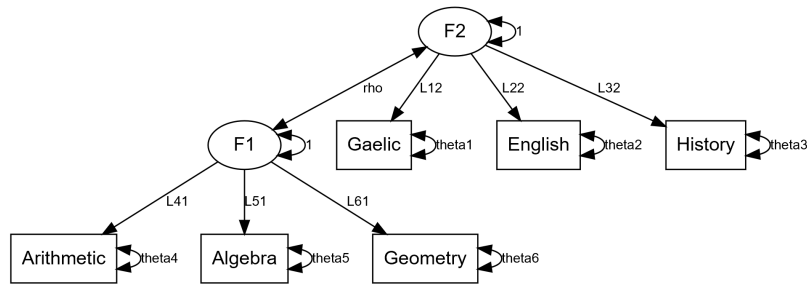
#相关系数矩阵 r:
r=matrix(0,6,6)
r[lower.tri(r)]=c(0.439,0.410,0.288,0.329,0.248,
0.351, 0.354,0.320,0.329, 0.164,0.190,0.181,0.595,0.470,0.464)
r= r+t(r)
diag(r)=1
rownames(r)=colnames(r)=c("Gaelic","English","History",
"Arithmetic","Algebra","Geometry")
    
```

假设如下结构方程模型（确认性因子分析模型）：

$$\begin{aligned}
 \text{Gaelic} &= L_{12}F_2 + \epsilon_1 \\
 \text{English} &= L_{22}F_2 + \epsilon_2 \\
 \text{History} &= L_{32}F_2 + \epsilon_3 \\
 \text{Arithmetic} &= L_{41}F_1 + \epsilon_4 \\
 \text{Algebra} &= L_{51}F_1 + \epsilon_5 \\
 \text{Geometry} &= L_{61}F_1 + \epsilon_6
 \end{aligned}$$

其中 $\epsilon_i, i = 1, \dots, 6$ 独立, $\text{var}(\epsilon_i) = \theta_i$
 ϵ_i 's 与 (F_1, F_2) 独立
 $\text{var}(F_1) = \text{var}(F_2) = 1, \text{cov}(F_1, F_2) = \rho$

该模型以如下路径图表示 (绘图工具参见最后一页)：



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 $=\sim$ 指定, 相关关系以 $\sim\sim$ 指定, 回归关系以 \sim 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 lavaan 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. 例 1 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.

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 $A \rightarrow B$ 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 1 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)

```
