中国石油大学(北京)克拉玛依校区
CHINA UNIVERSITY OF PETROLEUM - BEIJING AT KARAMAY

Factor Analysis

因子分析

肖 磊，2026 年 5 月 14 日

已学知识点 (Recap)

第 12 章 因子分析

- 12.1 正交因子模型
  - 因子分析模型旨在描述数据集中原始的 p 个变量如何依赖于少量潜在因子 $(k < p)$，即该模型假设 $\boldsymbol{X} = \boldsymbol{QF} + \boldsymbol{U} + \boldsymbol{\mu}$. 其中，($k$ 维) 随机向量 $\boldsymbol{F}$ 包含公因子，(p 维) 向量 $\boldsymbol{U}$ 包含特殊因子，而矩阵 $Q\,(p \times k)$ 则包含因子载荷.
  - 假设公因子 $\boldsymbol{F}$ 与特殊因子 $\boldsymbol{U}$ 互不相关且均值均为 $\mathbf{0}$，即 $\boldsymbol{F} \sim \left(\mathbf{0}, \mathscr{I}\right)$，$\boldsymbol{U} \sim \left(\mathbf{0}, \boldsymbol{\psi}\right)$，其中 $\boldsymbol{\psi}$ 为对角矩阵，且 $\operatorname{Cov}(\boldsymbol{F}, \boldsymbol{U}) = \mathbf{0}$.
  - 由此得出协方差结构为 $\boldsymbol{\Sigma} = QQ^{\mathrm{T}} + \boldsymbol{\psi}$.
  - 因子 $\boldsymbol{F}$ 的解释可通过相关矩阵 $P_{\boldsymbol{XF}} = D^{-1/2}Q$ 得到.
  - 标准化分析可通过模型 $P = QQ^{\mathrm{T}} + \boldsymbol{\psi}$ 实现. 因子的解释可由载荷矩阵 Q：$P_{\boldsymbol{XF}} = Q$ 直接给出.
  - 因子分析模型具有尺度不变性. 因子载荷不唯一 (仅在与一个正交矩阵相乘的范围内唯一).
  - 模型是否存在唯一解，由自由度 $d = \dfrac{(p-k)^2}{2} - \dfrac{p+k}{2}$ 决定. 若 $d \geq 0$，模型可识别 (存在唯一解或有限个解)；若 $d < 0$，模型不可识别 (存在无限个解).

因子模型的估计 (Estimation of the Factor Model)

- 在实际应用中，我们使用 $\mathcal{S}$，即 $\mathcal{X}$ 的经验协方差，来估计载荷 $\mathcal{Q}$ 和特定方差 Ψ:

$$\mathcal{S} = \widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}} + \widehat{\Psi}$$

Model: $\Sigma = \mathcal{Q}\mathcal{Q}^{\mathrm{T}} + \Psi$

  - 给定一个估计值 $\widehat{\mathcal{Q}}$，很自然会设定

$$\widehat{\psi}_{jj} = s_{X_jX_j} - \sum_{\ell=1}^{k} \hat{q}_{j\ell}^2$$

$$\sigma_{X_jX_j} = \sum_{\ell=1}^{k} q_{j\ell}^2 + \psi_{jj}$$

  - 对公因子方差 $h_j^2 = \sum_{\ell=1}^{k} q_{j\ell}^2$，我们有

$$\hat{h}_j^2 = \sum_{\ell=1}^{k} \hat{q}_{j\ell}^2$$

  - 在 $d = 0$ 的理想情况下，存在精确解.
  - 通常 $d > 0$，我们需要求得 $\widehat{\mathcal{Q}}$ 和 $\widehat{\Psi}$，以使 $\mathcal{S}$ 可以由 $\widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}} + \widehat{\Psi}$ 近似.
  - 这使得计算标准化模型的载荷和特殊方差更为容易.

因子模型的估计 (Estimation of the Factor Model)

- 在实际应用中，我们使用 $\mathcal{S}$，即 $\mathcal{X}$ 的经验协方差，来估计载荷 $\mathcal{Q}$ 和特定方差 Ψ:

Model: $\Sigma = \mathcal{Q}\mathcal{Q}^{\mathrm{T}} + \Psi$

$$\mathcal{S} = \widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}} + \widehat{\Psi}$$

$$\mathcal{S} = \frac{1}{n}\mathcal{X}^{\mathrm{T}}\mathcal{H}\mathcal{X}$$

- 设

$$\mathcal{Y} = \mathcal{H}\mathcal{X}\mathcal{D}^{-1/2}$$

$\mathcal{X}$ 的标准化

$$\mathcal{D} = \begin{pmatrix} s_{X_1X_1} & & & \\ & s_{X_2X_2} & & \\ & & \ddots & \\ & & & s_{X_pX_p} \end{pmatrix}$$

$$\mathcal{H} = \mathcal{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}$$

- 估计得到的 $\mathcal{Y}$ 的因子载荷矩阵 $\widehat{\mathcal{Q}}_Y$ 以及特殊方差 $\widehat{\Psi}_Y$ 为

$$\widehat{\mathcal{Q}}_Y = \mathcal{D}^{-1/2}\widehat{\mathcal{Q}}_X, \quad \widehat{\Psi}_Y = \mathcal{D}^{-1}\widehat{\Psi}_X$$

- 于是，对 $\mathcal{X}$ 的相关矩阵 $\mathcal{R}$，我们有 $\mathcal{R} = \widehat{\mathcal{Q}}_Y\widehat{\mathcal{Q}}_Y^{\mathrm{T}} + \widehat{\Psi}_Y$.
- 因子的解释是依据对因子载荷矩阵 $\widehat{\mathcal{Q}}_Y$ 的分析得出的.

因子模型的估计 (Estimation of the Factor Model)

- 例: 对于汽车数据，三个变量 (价格、安全性和操控便利性) 得到以下相关矩阵:

$$\mathscr{R} = \begin{pmatrix} 1 & 0.975 & 0.613 \\ 0.975 & 1 & 0.620 \\ 0.613 & 0.620 & 1 \end{pmatrix}$$

  - 我们寻找一个因子，即 $k = 1$.

$$d = \frac{1}{2}(p-k)^2 - \frac{1}{2}(p+k) = 0$$

$$\hat{h}_i^2 = \hat{q}_i^2$$

communalities: $h_j^2 = \sum_{\ell=1}^{k} q_{j\ell}^2$

  - 这意味着存在一个精确解!

$$\begin{pmatrix} 1 & r_{X_1X_2} & r_{X_1X_3} \\ r_{X_1X_2} & 1 & r_{X_2X_3} \\ r_{X_1X_3} & r_{X_2X_3} & 1 \end{pmatrix} = \mathscr{R} = \begin{pmatrix} \hat{q}_1^2 + \widehat{\psi}_{11} & \hat{q}_1\hat{q}_2 & \hat{q}_1\hat{q}_3 \\ \hat{q}_1\hat{q}_2 & \hat{q}_2^2 + \widehat{\psi}_{22} & \hat{q}_2\hat{q}_3 \\ \hat{q}_1\hat{q}_3 & \hat{q}_2\hat{q}_3 & \hat{q}_3^2 + \widehat{\psi}_{33} \end{pmatrix}$$

$$\Longrightarrow \quad \hat{q}_1^2 = \frac{r_{X_1X_2}r_{X_1X_3}}{r_{X_2X_3}}, \quad \hat{q}_2^2 = \frac{r_{X_1X_2}r_{X_2X_3}}{r_{X_1X_3}}, \quad \hat{q}_3^2 = \frac{r_{X_1X_3}r_{X_2X_3}}{r_{X_1X_2}}$$

$$\widehat{\psi}_{11} = 1 - \hat{q}_1^2, \quad \widehat{\psi}_{22} = 1 - \hat{q}_2^2, \quad \widehat{\psi}_{33} = 1 - \hat{q}_3^2$$

因子模型的估计 (Estimation of the Factor Model)

- 例: 对于汽车数据，三个变量 (价格、安全性和操控便利性) 得到以下相关矩阵:

$$\mathscr{R} = \begin{pmatrix} 1 & 0.975 & 0.613 \\ 0.975 & 1 & 0.620 \\ 0.613 & 0.620 & 1 \end{pmatrix}$$

  - 我们得到以下解

$$\hat{q}_1 = 0.982 \qquad \hat{q}_2 = 0.993 \qquad \hat{q}_3 = 0.624$$

$$\hat{\psi}_{11} = 0.035 \qquad \hat{\psi}_{22} = 0.014 \qquad \hat{\psi}_{33} = 0.610$$

  - 因为前两个公因子方差 $\left(\hat{h}_1^2 = \hat{q}_1^2\,,\ \hat{h}_2^2 = \hat{q}_2^2\right)$ 接近于 1，我们可以得出结论，前两个变量，即价格和安全性，能被该单一因子很好地解释.
  - 这个因子可以被解释为 “价格+安全性” 因子.

因子模型的估计 (Estimation of the Factor Model)

- 极大似然方法
  - 来自总体 $\boldsymbol{X} \sim N_p\left(\boldsymbol{\mu}, \Sigma\right)$ 的一个数据矩阵 $\mathscr{X}$ 的对数似然函数 ℓ 为：

$$\ell\left(\mathscr{X};\ \boldsymbol{\mu}, \Sigma\right) = -\frac{n}{2}\log\left|2\pi\Sigma\right| - \frac{1}{2}\sum_{i=1}^{n}\left(\boldsymbol{x}_i - \boldsymbol{\mu}\right)\Sigma^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}\right)^{\mathrm{T}}$$

$$= -\frac{n}{2}\log\left|2\pi\Sigma\right| - \frac{n}{2}\mathrm{tr}\left(\Sigma^{-1}\mathscr{S}\right) - \frac{n}{2}\left(\bar{\boldsymbol{x}} - \boldsymbol{\mu}\right)\Sigma^{-1}\left(\bar{\boldsymbol{x}} - \boldsymbol{\mu}\right)^{\mathrm{T}}$$

  - 它可以改写为

$$\ell\left(\mathscr{X};\ \hat{\boldsymbol{\mu}}, \Sigma\right) = -\frac{n}{2}\left[\log\left|2\pi\Sigma\right| + \mathrm{tr}\left(\Sigma^{-1}\mathscr{S}\right)\right]$$

  - 代入 $\Sigma = \mathscr{Q}\mathscr{Q}^{\mathrm{T}} + \Psi$ 后，该式变为

$$\ell\left(\mathscr{X};\ \hat{\boldsymbol{\mu}}, \mathscr{Q}, \Psi\right) = -\frac{n}{2}\left\{\log\left|2\pi\left(\mathscr{Q}\mathscr{Q}^{\mathrm{T}} + \Psi\right)\right| + \mathrm{tr}\left[\left(\mathscr{Q}\mathscr{Q}^{\mathrm{T}} + \Psi\right)^{-1}\mathscr{S}\right]\right\}$$

  - 即使在单个因子 $(k = 1)$ 的情形，要针对 $\widehat{\mathscr{Q}}$ 和 $\widehat{\Psi}$ 求解这些方程亦相当复杂.
  - 必须使用迭代数值算法求解 (详见 Mardia 等人 1979 年的著作，第 263 页).

因子模型的估计 (Estimation of the Factor Model)

- 公因子个数的似然比检验
  - 运用第 7 章中的检验方法，我们可以检验因子分析模型的适用性

$$\begin{cases} H_0: \text{因子分析模型是正确的} \\ H_1: \text{对协方差矩阵无任何约束} \end{cases}$$

  - 假设 $\widehat{\mathcal{Q}}$ 和 $\widehat{\Psi}$ 是相应于下式的极大似然估计

$$\ell\left(\mathcal{X};\ \widehat{\boldsymbol{\mu}}\ ,\ \mathcal{Q}\ ,\ \Psi\right) = -\frac{n}{2}\left\{\log\left|2\pi\left(\mathcal{Q}\mathcal{Q}^{\mathrm{T}}+\Psi\right)\right| + \operatorname{tr}\left[\left(\mathcal{Q}\mathcal{Q}^{\mathrm{T}}+\Psi\right)^{-1}\mathcal{S}\right]\right\}$$

$$\Longrightarrow \quad -2\log\left(\frac{\text{maximized likelihood under } H_0}{\text{maximized likelihood}}\right) = n\log\left(\frac{\left|\widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}}+\widehat{\Psi}\right|}{\left|\mathcal{S}\right|}\right)$$

$$\sim \chi^2_{\frac{1}{2}\left[(p-k)^2-p-k\right]}\ , \qquad n\to\infty$$

因子模型的估计 (Estimation of the Factor Model)

- 公因子个数的似然比检验
  - Bartlett (1954) 通过用 $n-1-\dfrac{2p+4k+5}{6}$ 代替 n 改进了 χ^2 近似.
  - 显著水平 α 时，如果下式成立我们会拒绝因子分析模型

$$\left(n-1-\frac{2p+4k+5}{6}\right)\log\left(\frac{\left|\widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}}+\widehat{\Psi}\right|}{|\mathcal{S}|}\right)>\chi^2_{\frac{(p-k)^2-p-k}{2}}(\alpha)$$

上 α 分位数

足够大

k 满足 $\dfrac{(p-k)^2-p-k}{2}$ 为正

$$\Longrightarrow\quad -2\log\left(\frac{\text{maximized likelihood under } H_0}{\text{maximized likelihood}}\right)=n\log\left(\frac{\left|\widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}}+\widehat{\Psi}\right|}{|\mathcal{S}|}\right)$$

$$\sim\chi^2_{\frac{1}{2}[(p-k)^2-p-k]},\qquad n\to\infty$$

因子模型的估计 (Estimation of the Factor Model)

- 主因子方法
  - 主因子法 (method of principal factors) 主要聚焦于相关矩阵 $\mathscr{R}$ 或协方差矩阵 $\mathscr{S}$ 的分解.
  - 为简便起见，这里仅讨论针对相关矩阵 $\mathscr{R}$ 的方法.
  - 假设我们知道确切的 Ψ，那么这些约束条件 $\mathscr{D} = \mathscr{I}$

$$\mathscr{Q}^{\mathrm{T}}\Psi^{-1}\mathscr{Q} \quad \text{为对角矩阵，} \quad \text{或} \quad \mathscr{Q}^{\mathrm{T}}D^{-1}\mathscr{Q} \quad \text{为对角矩阵.}$$

$$\Longrightarrow \mathscr{Q} \text{ 的列正交}$$

$$\mathscr{R} = \mathscr{Q}\mathscr{Q}^{\mathrm{T}} + \Psi \quad \Longrightarrow \quad \mathscr{Q}\mathscr{Q}^{\mathrm{T}} = \mathscr{R} - \Psi$$

$$\mathscr{Q} \triangleq (\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_k)$$

$$\Longrightarrow \mathscr{Q}\mathscr{Q}^{\mathrm{T}}\boldsymbol{q}_i = (\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_k)\begin{pmatrix} \boldsymbol{q}_1^{\mathrm{T}} \\ \boldsymbol{q}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{q}_k^{\mathrm{T}} \end{pmatrix}\boldsymbol{q}_i = (\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_k)\begin{pmatrix} 0 \\ \vdots \\ \boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{q}_i \\ \vdots \\ 0 \end{pmatrix} = \left(\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{q}_i\right)\boldsymbol{q}_i$$

- $\mathscr{Q}$ 的列是 $\mathscr{Q}\mathscr{Q}^{\mathrm{T}} = \mathscr{R} - \Psi$ 的特征向量.

因子模型的估计 (Estimation of the Factor Model)

- 主因子方法
  - 此外，假设 $\mathcal{Q}\mathcal{Q}^{\mathrm{T}} = \mathcal{R} - \Psi$ 的前 k 个特征值为正.
  - 在这种情况下，我们可以通过对 $\mathcal{Q}\mathcal{Q}^{\mathrm{T}} = \mathcal{R} - \Psi$ 进行谱分解来计算 $\mathcal{Q}$.
  - k 就是因子数.
  - 当然，我们必须先找到一种方法来估计 Ψ.

因子模型的估计 (Estimation of the Factor Model)

- 主因子方法
  - 主因子方法基于对公因子方差 $h_j^2 \ (j = 1, 2, \ldots, p)$ 良好的初步估计量 $\hat{h}_j^2$.

$$1 = \sigma_{X_j X_j} = \sum_{\ell=1}^{k} q_{j\ell}^2 + \psi_{jj} = h_j^2 + \psi_{jj}$$

  - 有两种传统的方案：
    - $\hat{h}_j^2$，定义为 X_j 与 $X_l \ (l \neq j)$ 的复相关系数的平方.

$$\hat{h}_j^2 = \rho^2\left(V, \boldsymbol{W}\hat{\boldsymbol{\beta}}\right)$$

V：X_j；$\boldsymbol{W} = (X_1 \ \cdots \ X_{j-1} \ X_{j+1} \ \cdots \ X_p)$

变量 V 关于 $\boldsymbol{W}$ 回归的最小二乘回归参数

    - $\hat{h}_j^2 = \max_{\ell \neq j} \left| r_{X_j X_\ell} \right|$，其中 $\mathscr{R} = \left(r_{X_j X_\ell}\right)$ 是 $\mathscr{X}$ 的相关矩阵. ✓ 简便易行

因子模型的估计 (Estimation of the Factor Model)

- 主因子方法
  - 给定 $\widetilde{\psi}_{jj} = 1 - \widetilde{h}_j^2$，我们可以构建简化相关矩阵 (reduced correlation matrix)，$\mathscr{R} - \widetilde{\Psi}$.
  - 谱分解定理表明

$$\mathscr{R} - \widetilde{\Psi} = \sum_{\ell=1}^{p} \lambda_\ell \, \boldsymbol{\gamma}_\ell \, \boldsymbol{\gamma}_\ell^{\mathrm{T}}$$

$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$ 为特征值

  - 假设前 k 个特征值 $\lambda_1, \lambda_2, \ldots, \lambda_k$ 为正，且相比其它特征值足够大.
  - 则我们令 $\hat{\boldsymbol{q}}_\ell = \sqrt{\lambda_\ell}\, \boldsymbol{\gamma}_\ell, \quad \ell = 1, 2, \ldots, k$ —in matrix form→ $\widehat{\mathscr{Q}} = \Gamma_1 \Lambda_1^{1/2}$

$$\Gamma_1 = (\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_k), \quad \Lambda_1 = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_k)$$

  - 下一步，我们令

$$\widehat{\psi}_{jj} = 1 - \sum_{\ell=1}^{k} \hat{q}_{j\ell}^2, \quad j = 1, 2, \ldots, p$$

因子模型的估计 (Estimation of the Factor Model)

- 主因子方法
  - 注意：该过程可以迭代进行.
  - 由 $\widehat{\psi}_{jj}$ 我们可以计算新的简化相关矩阵 $\mathscr{R}-\widetilde{\Psi}$，然后进行相同的处理.
  - 迭代过程一般当 $\widehat{\psi}_{jj}$ 收敛到某一稳定值时终止.

因子模型的估计 (Estimation of the Factor Model)

- 例: 考虑汽车品牌数据.

X_1 : 经济性

X_2 : 服务

X_3 : 保值性

X_4 : 价格，每辆廉价汽车记为 1

X_5 : 设计

X_6 : 运动型

X_7 : 安全性

X_8 : 操控简便

```
> (x = carmean2)
     Economy Service Value Price Design Sporty Safety Easy
A100     3.9     2.8   2.2   4.2    3.0    3.1    2.4  2.8
BMW3     4.8     1.6   1.9   5.0    2.0    2.5    1.6  2.8
CiAX     3.0     3.8   3.8   2.7    4.0    4.4    4.0  2.6
Ferr     5.3     2.9   2.2   5.9    1.7    1.1    3.3  4.3
FiUn     2.1     3.9   4.0   2.6    4.5    4.4    4.4  2.2
FoFi     2.3     3.1   3.4   2.6    3.2    3.3    3.6  2.8
Hyun     2.5     3.4   3.2   2.2    3.3    3.3    3.3  2.4
Jagu     4.6     2.4   1.6   5.5    1.3    1.6    2.8  3.6
Lada     3.2     3.9   4.3   2.0    4.3    4.5    4.7  2.9
Mazd     2.6     3.3   3.7   2.8    3.7    3.0    3.7  3.1
M200     4.1     1.7   1.8   4.6    2.4    3.2    1.4  2.4
Mits     3.2     2.9   3.2   3.5    3.1    3.1    2.9  2.6
NiSu     2.6     3.3   3.9   2.1    3.5    3.9    3.8  2.4
OpCo     2.2     2.4   3.0   2.6    3.2    4.0    2.9  2.4
OpVe     3.1     2.6   2.3   3.6    2.8    2.9    2.4  2.4
P306     2.9     3.5   3.6   2.8    3.2    3.8    3.2  2.6
Re19     2.7     3.3   3.4   3.0    3.1    3.4    3.0  2.7
Rove     3.9     2.8   2.6   4.0    2.6    3.0    3.2  3.0
ToCo     2.5     2.9   3.4   3.0    3.2    3.1    3.2  2.8
Trab     3.6     4.7   5.5   1.5    4.1    5.8    5.9  3.1
VWGo     3.8     2.3   1.9   4.2    3.1    3.6    1.6  2.4
VWPa     3.1     2.2   2.1   3.2    3.5    3.5    2.8  1.8
Wart     3.7     4.7   5.5   1.7    4.8    5.2    5.5  4.0
```

```
# clear variables and close windows
rm(list = ls(all = TRUE))
graphics.off()
options(digits = 3)
# load data 载入数据
load("~/Desktop/2023_Applied Multivariate Statistical Analysis/R Codes with data/MVAexercise/data/carmean2.rda")
(x = carmean2)
```

因子模型的估计 (Estimation of the Factor Model)

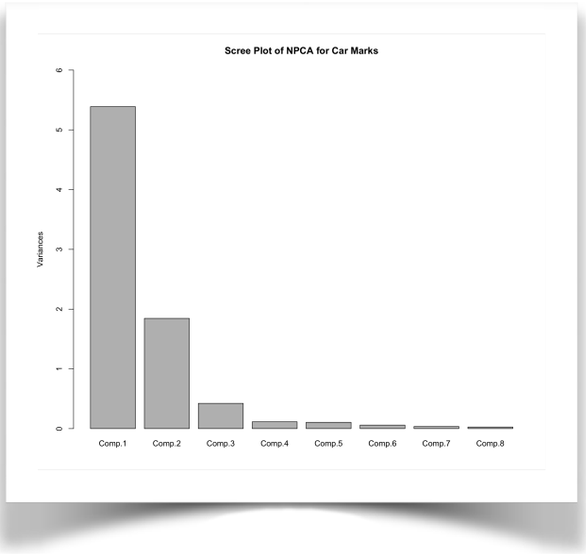
- 例: 考虑汽车品牌数据.
  - 首先，我们作标准化主成分分析 (NPCA).

```
NPCA.car = princomp(x, cor = TRUE, scores = TRUE, fix_sign = TRUE)
summary(NPCA.car)
# 主成分的标准差
NPCA.car$sdev
# 主成分在原变量的载荷（权重）
NPCA.car$loadings
# 碎石图
screeplot(NPCA.car, ylim = c(0, 6), main = 'Scree Plot of NPCA for Carmeans')
```

```
> summary(NPCA.car)
Importance of components:
                        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  Comp.6  Comp.7 Comp.8
Standard deviation      2.321  1.358 0.6505 0.3385 0.3255 0.24372 0.18902 0.1649
Proportion of Variance  0.674  0.231 0.0529 0.0143 0.0132 0.00743 0.00447 0.0034
Cumulative Proportion   0.674  0.904 0.9571 0.9715 0.9847 0.99213 0.99660 1.0000
```

```
> # 主成分的标准差
> NPCA.car$sdev
Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
 2.321  1.358  0.650  0.339  0.325  0.244  0.189  0.165
```

```
> # 主成分在原变量的载荷（权重）
> NPCA.car$loadings

Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
Economy  0.268  0.469  0.681         0.169         0.460
Service -0.382  0.285 -0.122  0.309  0.385 -0.681         0.217
Value   -0.410  0.181        -0.304 -0.137 -0.135  0.207 -0.790
Price    0.409  0.170         0.423  0.129 -0.121 -0.582 -0.496
Design  -0.403 -0.112  0.222  0.725 -0.421  0.230  0.140
Sporty  -0.382 -0.109  0.628 -0.316        -0.135 -0.546  0.154
Safety  -0.371  0.325 -0.129         0.518  0.654 -0.207
Easy            0.712 -0.222        -0.578        -0.208  0.239
```



因子模型的估计 (Estimation of the Factor Model)

- 例: 考虑汽车品牌数据.
  - 首先,

```
# 观测值在各主成分
NPCA.car$scores
```

```
> # 观测值在各主成分的投影
> NPCA.car$scores
      Comp.1  Comp.2  Comp.3  Comp.4   Comp.5    Comp.6  Comp.7   Comp.8
A100   1.548  0.0392  0.4672  0.4334 -0.07896 -0.239560 -0.0103  0.19512
BMW3   3.793  0.1371  0.9096 -0.3507 -0.17345  0.143244  0.2383 -0.29832
CiAX  -1.896 -0.1097  0.3648  0.3392  0.19944 -0.163386 -0.1264 -0.00295
Ferr   3.710  3.6964 -0.5830  0.3574  0.12605  0.080299  0.0412 -0.07478
FiUn  -2.741 -1.0194 -0.1459  0.7902  0.38551  0.036192 -0.3597 -0.28232
FoFi  -0.708 -0.4658 -1.0182 -0.2491 -0.13754  0.138473 -0.1252  0.08822
Hyun  -0.822 -0.9573 -0.7059 -0.0715  0.25657 -0.196482  0.3918  0.31022
Jagu   3.955  1.9157 -0.5320 -0.2123  0.43147  0.084386 -0.2755  0.18273
Lada  -2.723  0.5623  0.3695  0.1730  0.00205  0.268028  0.1891  0.09366
Mazd  -0.900  0.2348 -0.9898  0.2993 -0.49059  0.175053  0.1271 -0.09948
M200   3.007 -0.9967  1.0296 -0.2394 -0.23603 -0.052678 -0.0790 -0.12001
Mits   0.418 -0.3696 -0.1664  0.0580  0.01808 -0.110856  0.1594 -0.31368
NiSu  -1.541 -0.7779 -0.2783 -0.3528  0.20906  0.091617  0.2385 -0.10474
OpCo  -0.299 -1.6407 -0.2926 -0.5733 -0.41936  0.264001 -0.4082  0.11805
OpVe   1.293 -1.0152 -0.2198  0.0606  0.12199 -0.115153  0.0847  0.11914
P306  -0.784 -0.3557 -0.1473 -0.2063  0.15024 -0.515628  0.0493 -0.04454
Re19  -0.329 -0.4166 -0.5459 -0.1361 -0.05269 -0.445360 -0.0131 -0.05380
Rove   1.296  0.6375  0.0893 -0.0913  0.22712  0.114160 -0.0716  0.08228
ToCo  -0.196 -0.4646 -0.8704 -0.1244 -0.32180  0.072775 -0.0475 -0.17032
Trab  -4.414  1.7019  1.0139 -0.6359  0.60578 -0.000799 -0.0910 -0.05204
VWGo   1.884 -1.0680  1.0710  0.3072 -0.35679 -0.303647 -0.1144  0.16926
VWPa   0.686 -2.0675  0.5725  0.3436  0.31948  0.635403  0.1589  0.13269
Wart  -4.237  2.7998  0.6081  0.0813 -0.78563  0.039920  0.0436  0.12562
```

因子模型的估计 (Estimation of the Factor Model)

- 例: 考虑汽车品牌数据.
  - 首先，我们作标准化主成分分析 (NPCA).

$$\hat{\boldsymbol{q}}_\ell = \sqrt{\lambda_\ell}\,\boldsymbol{\gamma}_\ell, \quad \ell = 1, 2, \ldots, p$$

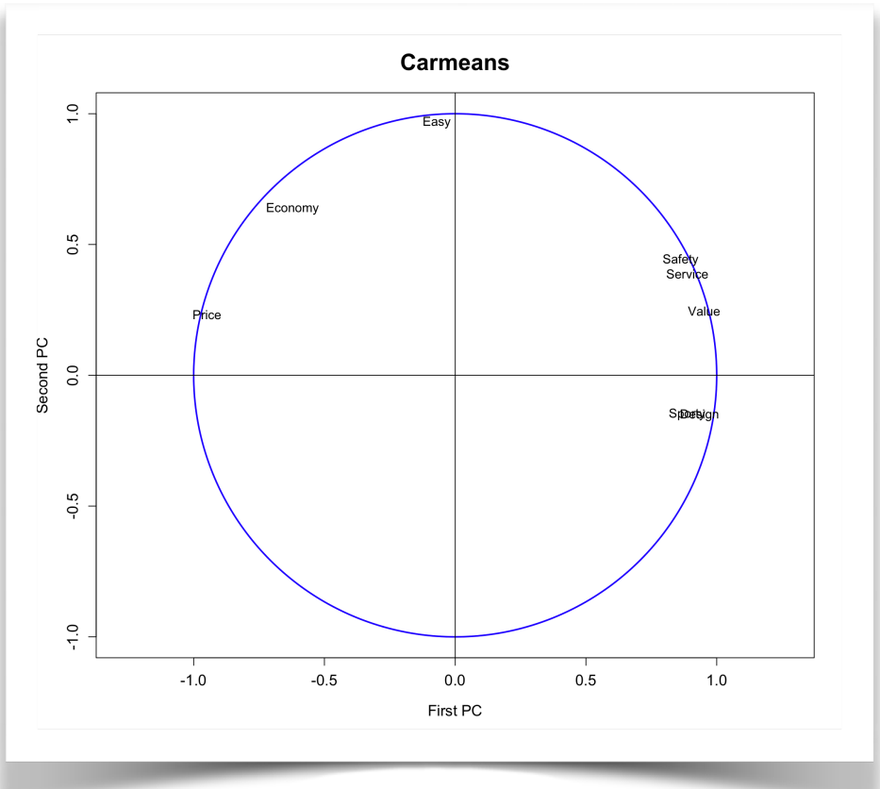
```
# 各变量与第一主成分的相关系数
cor.x_NPC1 = NPCA.car$sdev[1] * NPCA.car$loadings[, 1]
# 各变量与第二主成分的相关系数
cor.x_NPC2 = NPCA.car$sdev[2] * NPCA.car$loadings[, 2]
# 各变量方差由前两个主成分解释的比例
NPC2.prop = cor.x_NPC1^2 + cor.x_NPC2^2
cbind(NPC1 = cor.x_NPC1, NPC2 = cor.x_NPC2, Sum_Squares = NPC2.prop)
```

```
> cbind(NPC1 = cor.x_NPC1, NPC2 = cor.x_NPC2, Sum_Squares = NPC2.prop)
            NPC1   NPC2 Sum_Squares
Economy   0.6218  0.637       0.793
Service -0.8877  0.387       0.938
Value    -0.9520  0.246       0.967
Price     0.9489  0.231       0.954
Design   -0.9344 -0.153       0.896
Sporty   -0.8867 -0.148       0.808
Safety   -0.8620  0.441       0.937
Easy      0.0703  0.967       0.940
```

因子模型的估计 (Estimation of the Factor Model)

- 例: 考虑汽车品牌数据.
  - 首先，我们作标准化主成分分析 (NPCA).

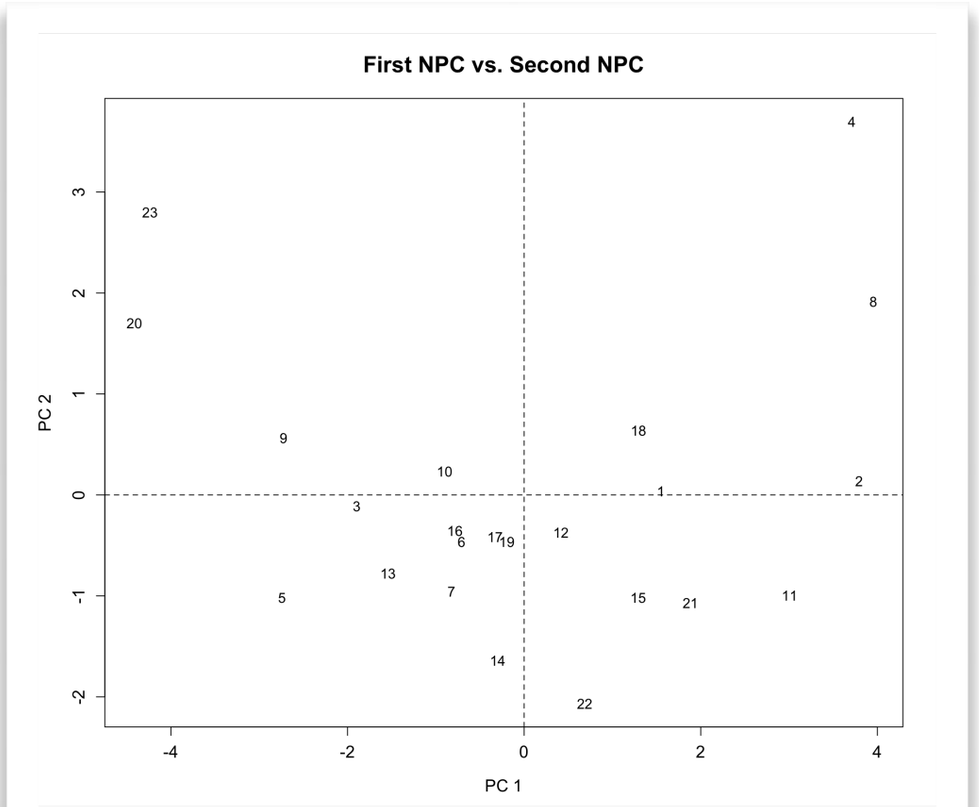
```
# 变量投影在前两个主成分上的散点图
graphics.off()
ucircle = cbind(cos((0:360)/180 * pi), sin((0:360)/180 * pi))
plot(ucircle, type = "l", lty = "solid", col = "blue", xlab = "First PC", ylab = "Second PC",
    main = "Carmeans", cex.lab = 1.2, cex.axis = 1.2, cex.main = 1.8, lwd = 2, asp = 1)
abline(h = 0, v = 0)
# label = c("X1", "X2", "X3", "X4", "X5", "X6", "X7", "X8")
label = row.names(cor.NPC.car.2)
text(-cor.x_NPC1, cor.x_NPC2, label, cex = 1)
```



因子模型的估计 (Estimation of the Factor Model)

- 例: 考虑汽车品牌数据.
  - 首先，我们作标准化主成分分析 (NPCA).

```
# 观测值在前两个主成分的散点图
graphics.off()
plot(cbind(NPCA.car$scores[, 1], NPCA.car$scores[, 2]), type = "n", xlab = "PC 1", ylab = "PC 2",
    main = "First NPC vs. Second NPC", cex.lab = 1.2, cex.axis = 1.2, cex.main = 1.6)
abline(h = 0, v = 0, lty = 2)
label = 1:NPCA.car$n.obs
text(NPCA.car$scores[, 1], NPCA.car$scores[, 2], label)
```



因子模型的估计 (Estimation of the Factor Model)

- 例: 考虑汽车品牌数据.
  - 现在，我们使用主因子法来确定因子的数量.

```
# correlation matrix 计算相关矩阵
(r = cor(x))
# 将相关矩阵主对角元素变为零，记为矩阵 m
m = r
for (i in 1:ncol(r)) {
  m[i, i] = r[i, i] - 1
}
m
```

```
> (r = cor(x))
        Economy Service  Value  Price Design Sporty Safety   Easy
Economy   1.000  -0.335 -0.448  0.758 -0.619 -0.476 -0.284  0.583
Service  -0.335   1.000  0.928 -0.737  0.758  0.688  0.938  0.297
Value    -0.448   0.928  1.000 -0.858  0.829  0.801  0.917  0.180
Price     0.758  -0.737 -0.858  1.000 -0.887 -0.856 -0.714  0.270
Design   -0.619   0.758  0.829 -0.887  1.000  0.883  0.712 -0.217
Sporty   -0.476   0.688  0.801 -0.856  0.883  1.000  0.658 -0.251
Safety   -0.284   0.938  0.917 -0.714  0.712  0.658  1.000  0.348
Easy      0.583   0.297  0.180  0.270 -0.217 -0.251  0.348  1.000
```

```
> m
        Economy Service  Value  Price Design Sporty Safety   Easy
Economy   0.000  -0.335 -0.448  0.758 -0.619 -0.476 -0.284  0.583
Service  -0.335   0.000  0.928 -0.737  0.758  0.688  0.938  0.297
Value    -0.448   0.928  0.000 -0.858  0.829  0.801  0.917  0.180
Price     0.758  -0.737 -0.858  0.000 -0.887 -0.856 -0.714  0.270
Design   -0.619   0.758  0.829 -0.887  0.000  0.883  0.712 -0.217
Sporty   -0.476   0.688  0.801 -0.856  0.883  0.000  0.658 -0.251
Safety   -0.284   0.938  0.917 -0.714  0.712  0.658  0.000  0.348
Easy      0.583   0.297  0.180  0.270 -0.217 -0.251  0.348  0.000
```

因子模型的估计 (Estimation of the Factor Model)

- 例: 考虑汽车品牌数据.
  - 现在，我们使用主因子法来确定因子的数量.

$$\widetilde{\psi}_{jj} = 1 - \widetilde{h}_j^2 , \quad \widehat{h}_j^2 = \max_{\ell \neq j} \left| r_{X_j X_\ell} \right| , \quad j = 1, 2, \ldots, p$$

```
# 计算 Ψ̂_jj
psi = matrix(0, 8, 8)
for (i in 1:8) {
  psi[i, i] = 1 - max(abs(m[, i]))
}
psi
```

```
> psi
      [,1]   [,2]   [,3]  [,4]  [,5]  [,6]   [,7]  [,8]
[1,] 0.242 0.0000 0.0000 0.000 0.000 0.000 0.0000 0.000
[2,] 0.000 0.0624 0.0000 0.000 0.000 0.000 0.0000 0.000
[3,] 0.000 0.0000 0.0723 0.000 0.000 0.000 0.0000 0.000
[4,] 0.000 0.0000 0.0000 0.113 0.000 0.000 0.0000 0.000
[5,] 0.000 0.0000 0.0000 0.000 0.113 0.000 0.0000 0.000
[6,] 0.000 0.0000 0.0000 0.000 0.000 0.117 0.0000 0.000
[7,] 0.000 0.0000 0.0000 0.000 0.000 0.000 0.0624 0.000
[8,] 0.000 0.0000 0.0000 0.000 0.000 0.000 0.0000 0.417
```

因子模型的估计 (Estimation of the Factor Model)

- 例: 考虑汽车品牌数据.
  - 现在，我们使用主因子法来确定因子的数量.

```
# 对 R − Ψ 作谱分解 spectral decomposition
eig = eigen(r - psi)
ee  = eig$values
ee # 特征值: 前两个特征值为正，且与其它相比较大，所以取 k = 2 个因子即可
  ee  = eig$values[1:2] # 取前两个大的特征值
  ee
```

```
> ee
[1]  5.28764  1.57179  0.25108  0.01535 -0.00595 -0.02901 -0.09125 -0.19792
```

```
> ee
[1] 5.29 1.57
```

因子模型的估计 (Estimation of the Factor Model)

- 例: 考虑汽车品牌数据.
  - 现在，我们使用主因

```
> vv
        [,1]   [,2]    [,3]    [,4]     [,5]    [,6]    [,7]    [,8]
[1,]  0.2593 -0.485 -0.5962 -0.0223  0.00273  0.0196 -0.3354  0.4785
[2,] -0.3862 -0.310  0.1711  0.5630  0.37824 -0.4812  0.0394  0.1799
[3,] -0.4129 -0.188  0.0541 -0.4232  0.45738  0.4685  0.3207  0.2837
[4,]  0.4073 -0.201 -0.1372  0.3278 -0.07928  0.1322  0.8020 -0.0224
[5,] -0.4013  0.138 -0.2438  0.5670 -0.22339  0.6011 -0.1579 -0.0545
[6,] -0.3806  0.133 -0.7014 -0.1925  0.03131 -0.3468  0.2353 -0.3631
[7,] -0.3752 -0.355  0.1822 -0.1877 -0.76378 -0.1424  0.1628  0.1865
[8,]  0.0263 -0.656  0.0942 -0.0470  0.08488  0.1709 -0.1817 -0.6986
```

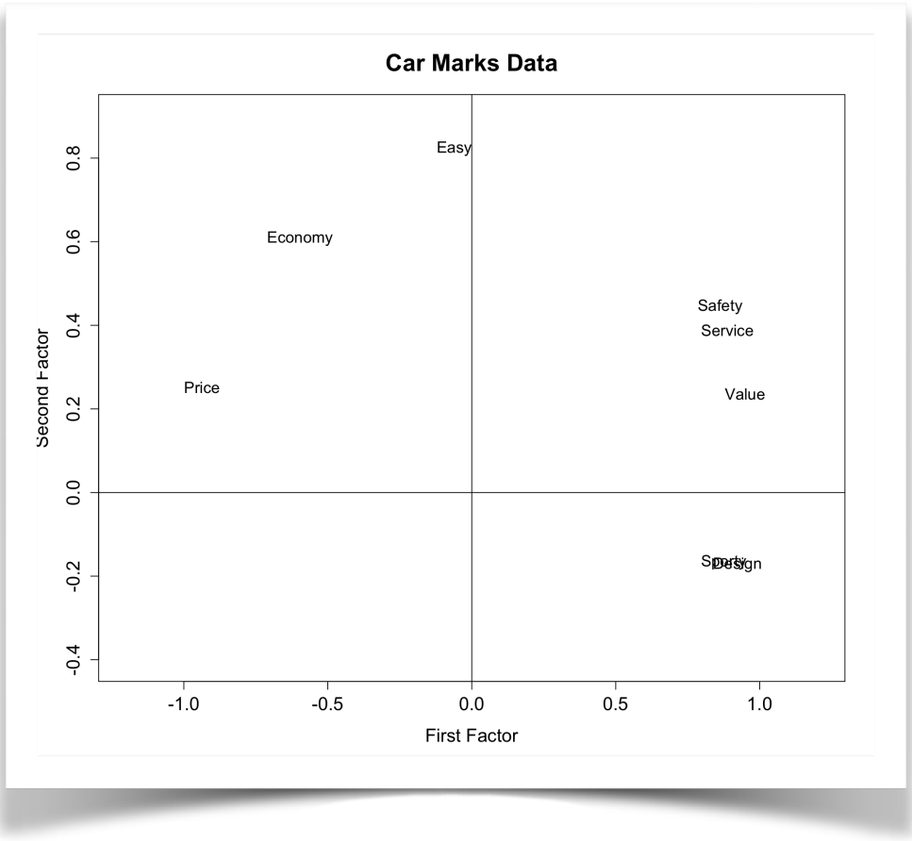
```
# 特征向量
vv  = eig$vectors
vv
vv  = eig$vectors[, 1:2] # 前两个特征向量
vv  = t(t(vv[, 1:2]) * sign(vv[2, 1:2])) # 调整特征向量的符号
q1  = sqrt(ee[1]) * vv[, 1] # 变量在第一个主方向上的投影
q2  = sqrt(ee[2]) * vv[, 2] # 变量在第二个主方向上的投影
(q   = cbind(q1, q2))
```

```
> (q   = cbind(q1, q2))
          q1     q2
[1,]  0.5962 -0.608
[2,] -0.8880 -0.389
[3,] -0.9495 -0.236
[4,]  0.9365 -0.252
[5,] -0.9227  0.173
[6,] -0.8751  0.167
[7,] -0.8627 -0.445
[8,]  0.0606 -0.823
```

因子模型的估计 (Estimation of the Factor Model)

- 例: 考虑汽车品牌数据.
  - 现在，我们使用主因子法来确定因子的数量.

```
# plot 变量在前两个因子上的投影散点图
plot(q, type = "n", xlab = "First Factor", ylab = "Second Factor", main = "Car Marks Data",
     xlim = c(-1.2, 1.2), ylim = c(-0.4, 0.9), cex.lab = 1.4, cex.axis = 1.4, cex.main = 1.8)
text(q, colnames(x), cex = 1.2, xpd = NA)
abline(v = 0)
abline(h = 0)
```



因子模型的估计 (Estimation of the Factor Model)

- 主成分方法
  - 主成分方法 (The Principal Component Method) 从因子载荷矩阵 $\mathcal{Q}$ 的一个近似值 $\widehat{\mathcal{Q}}$ 开始.
  - 样本协方差矩阵： $\mathcal{S} = \Gamma\Lambda\Gamma^{\mathrm{T}}$.
  - 然后保留前 k 个特征向量来构建

$$\widehat{\mathcal{Q}} = \left(\sqrt{\lambda_1}\boldsymbol{\gamma}_1 , \sqrt{\lambda_2}\boldsymbol{\gamma}_2 , \cdots , \sqrt{\lambda_k}\boldsymbol{\gamma}_k\right)$$

$\mathcal{S} = \widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}} + \widehat{\Psi}$

  - 特殊方差的估计值由矩阵 $\mathcal{S} - \widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}}$ 的对角元素给出，

$$\widehat{\Psi} = \begin{pmatrix} \widehat{\psi}_{11} & & & \\ & \widehat{\psi}_{22} & & \\ & & \ddots & \\ & & & \widehat{\psi}_{pp} \end{pmatrix} \quad \text{with} \quad \widehat{\psi}_{jj} = s_{X_jX_j} - \sum_{\ell=1}^{k} \hat{q}_{j\ell}^2$$

因子模型的估计 (Estimation of the Factor Model)

- 主成分方法
  - 由定义

$$\mathcal{S} = \widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}} + \widehat{\Psi}$$

  - $\mathcal{S}$ 的主对角元素与 $\widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}} + \widehat{\Psi}$ 的主对角元素相等.
  - 这种近似的效果如何?
  - 考虑主成分求解得到的残差矩阵

$$\mathcal{S} - \left(\widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}} + \widehat{\Psi}\right)$$

  - 从分析角度来看，我们有

$$\sum_{i,\,j}\left(\mathcal{S} - \widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}} - \widehat{\Psi}\right)_{ij}^{2} \le \lambda_{k+1}^{2} + \cdots + \lambda_{p}^{2}$$

  - 这意味着被忽略的特征值数值较小，会使得近似误差也较小.

因子模型的估计 (Estimation of the Factor Model)

- 主成分方法
  - 一种用于选择因子数量的启发式方法是，考虑第 j 个因子在样本总方差中所占的比例.
  - 这个量通常等于
    - 对于协方差矩阵 $\mathcal{S}$ 的因子分析，$\dfrac{\lambda_j}{\sum_{j=1}^{p} s_{jj}}$.
    - 对于相关矩阵 $\mathcal{R}$ 的因子分析，$\dfrac{\lambda_j}{p}$.

因子模型的估计 (Estimation of the Factor Model)

- 例: 要求客户对一款新产品的几个属性进行评分. 将结果制成表格，得到了相关矩阵 $\mathscr{R}$ 如下:

变量 1 和 3 高度相关.

属性 (变量) :

味道	1
物有所值	2
口感	3
适合作零食	4
提供大量能量	5

$$\begin{pmatrix} 1.00 & 0.02 & 0.96 & 0.42 & 0.01 \\ 0.02 & 1.00 & 0.13 & 0.71 & 0.85 \\ 0.96 & 0.13 & 1.00 & 0.50 & 0.11 \\ 0.42 & 0.71 & 0.50 & 1.00 & 0.79 \\ 0.01 & 0.85 & 0.11 & 0.79 & 1.00 \end{pmatrix} = \mathscr{R}$$

变量 2 和 5 高度相关.

变量 4 与变量 2 和 5 的相关性更强.

```
x = c(1.00, 0.02, 0.96, 0.42, 0.01, 0.02, 1.00, 0.13, 0.71, 0.85,
      0.96, 0.13, 1.00, 0.50, 0.11, 0.42, 0.71, 0.50, 1.00, 0.79,
      0.01, 0.85, 0.11, 0.79, 1.00)
R = matrix(x, nrow = 5, byrow = TRUE)
R
```

  - 因此，包含 2 (或 3) 个因子的模型似乎是合理的.

因子模型的估计 (Estimation of the Factor Model)

- 例: 要求客户对一款新产品的几个属性进行评分. 将结果制成表格，得到了相关矩阵 $\mathscr{R}$ 如下:

```
R.eig = eigen(R) # 谱分解
R.eig$values # 特征值
sum(R.eig$values[1:2]) / 5 # 两个公共因子的累积方差占总方差的比例
```

```
> R.eig$values
[1] 2.8531 1.8063 0.2045 0.1024 0.0337
```

  - 矩阵 $\mathscr{R}$ 的前两个特征值是仅有的大于 1 的特征值.

$$\lambda_1 = 2.8531\ , \quad \lambda_2 = 1.8063$$

```
> sum(R.eig$values[1:2]) / 5
[1] 0.932
```

  - $k = 2$ 个公共因子累计占总 (标准化) 样本方差的 93.2%.

$$\frac{\lambda_1 + \lambda_2}{p} = \frac{2.8531 + 1.8063}{5} = 0.932$$

因子模型的估计 (Estimation of the Factor Model)

- 例: 要求客户对一款新产品的几个属性进行评分. 将结果制成表格，得到了相关矩阵 $\mathscr{R}$ 如下:
  - 因子载荷的估计值为

$$\widehat{\mathscr{Q}} = \left(\sqrt{\lambda_1}\,\gamma_1\,,\sqrt{\lambda_2}\,\gamma_2\,,\ \cdots\ ,\sqrt{\lambda_k}\,\gamma_k\right)$$

```
# 公共因子的载荷 (k=2)
load.F = R.eig$vectors %*% diag(sqrt(R.eig$values))
round(load.F[, 1:2], digits = 2)
```

```
> round(load.F[, 1:2], digits = 2)
     [,1]  [,2]
[1,] 0.56 -0.82
[2,] 0.78  0.52
[3,] 0.65 -0.75
[4,] 0.94  0.10
[5,] 0.80  0.54
```

  - 公因子方差的估计值为

$$\hat{h}_j^2 = \sum_{\ell=1}^{k} \hat{q}_{j\ell}^2$$

```
# 公共因子方差 (k=2)
h = load.F^2
h = h[, 1] + h[, 2]
round(h, digits = 2)
```

```
> round(h, digits = 2)
[1] 0.98 0.88 0.98 0.89 0.93
```

因子模型的估计 (Estimation of the Factor Model)

- 例: 要求客户对一款新产品的几个属性进行评分. 将结果制成表格，得到了相关矩阵 $\mathscr{R}$ 如下:
  - 特殊因子方差的估计值为

$$\widehat{\psi}_{jj} = 1 - \hat{h}_j^2$$

```
# 特殊因子方差 (k=2)
Psi = 1 - h
round(Psi, digits = 2)
```

```
> round(Psi, digits = 2)
[1] 0.02 0.12 0.02 0.11 0.07
```

```
# 放在一个数据框中
FA = cbind(Load_F1 = load.F[, 1], Load_F2 = load.F[, 2], Communalities = h, S_Variances = Psi)
round(FA, digits = 2)
```

```
> round(FA, digits = 2)
     Load_F1 Load_F2 Communalities S_Variances
[1,]    0.56   -0.82          0.98         0.02
[2,]    0.78    0.52          0.88         0.12
[3,]    0.65   -0.75          0.98         0.02
[4,]    0.94    0.10          0.89         0.11
[5,]    0.80    0.54          0.93         0.07
```

属性 (变量) :			
味道	1		
物有所值	2		
口感	3		
适合作零食	4		
提供大量能量	5		
特征值		2.85	1.81
累积百分比		0.571	0.932

因子模型的估计 (Estimation of the Factor Model)

- 例: 要求客户对一款新产品的几个属性进行评分. 将结果制成表格，得到了相关矩阵 $\mathscr{R}$ 如下:
  - 我们来看 $\widehat{\mathscr{Q}}\,\widehat{\mathscr{Q}}^{\mathrm{T}}+\widehat{\mathscr{\Psi}}$:

$$\widehat{\mathscr{Q}}\,\widehat{\mathscr{Q}}^{\mathrm{T}}+\widehat{\mathscr{\Psi}}=\begin{pmatrix}0.56 & -0.82\\ 0.78 & 0.52\\ 0.65 & -0.75\\ 0.94 & 0.10\\ 0.80 & 0.54\end{pmatrix}\begin{pmatrix}0.56 & 0.78 & 0.65 & 0.94 & 0.80\\ -0.82 & 0.52 & -0.75 & 0.10 & 0.54\end{pmatrix}+\begin{pmatrix}0.02 & & & & \\ & 0.12 & & & \\ & & 0.02 & & \\ & & & 0.11 & \\ & & & & 0.07\end{pmatrix}$$

```
# 验证估计的效果
load.F[, 1:2] %*% t(load.F[, 1:2]) + diag(Psi)
```

这几乎重现了相关矩阵 $\mathscr{R}$.

$$=\begin{pmatrix}1.00 & 0.01 & 0.97 & 0.44 & 0.00\\ 0.01 & 1.00 & 0.11 & 0.78 & 0.91\\ 0.97 & 0.11 & 1.00 & 0.53 & 0.11\\ 0.44 & 0.78 & 0.53 & 1.00 & 0.81\\ 0.00 & 0.91 & 0.11 & 0.81 & 1.00\end{pmatrix}\qquad \mathscr{R}=\begin{pmatrix}1.00 & 0.02 & 0.96 & 0.42 & 0.01\\ 0.02 & 1.00 & 0.13 & 0.71 & 0.85\\ 0.96 & 0.13 & 1.00 & 0.50 & 0.11\\ 0.42 & 0.71 & 0.50 & 1.00 & 0.79\\ 0.01 & 0.85 & 0.11 & 0.79 & 1.00\end{pmatrix}$$

  - 由于因子载荷的非唯一性，通过旋转可能会改进解释效果.

因子模型的估计 (Estimation of the Factor Model)

- 因子旋转 (Rotation)
  - 约束条件

$$\mathcal{Q}^{\mathrm{T}}\Psi^{-1}\mathcal{Q} \quad \text{为对角矩阵,} \quad \text{或} \quad \mathcal{Q}^{\mathrm{T}}D^{-1}\mathcal{Q} \quad \text{为对角矩阵.}$$

    是出于数学上的方便而给出的 (以得到唯一解).

  - 但它们会使因子的解释问题变得更为复杂.
  - 如果变量能被划分为互不相交的集合，每个集合与一个因子相关联，那么对载荷的解释就会非常简单.

因子模型的估计 (Estimation of the Factor Model)

- 因子旋转 (Rotation)
  - Kaiser (1985 年) 提出的**方差最大旋转法** (varimax rotation method)，是一种著名的用于旋转载荷的解析算法.
  - 对于最简单的情形 $k = 2$ 个因子，此时的旋转矩阵 $\mathcal{G}$ 由下式给出

$$\mathcal{G}(\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

  - 它表示坐标轴顺时针旋转角度 θ.
  - 相应的载荷旋转通过以下方式计算

$$\widehat{\mathcal{Q}}^* = \widehat{\mathcal{Q}}\ \mathcal{G}(\theta)$$

因子模型的估计 (Estimation of the Factor Model)

- 因子旋转 (Rotation)
  - 方差最大旋转法的思想：确定角度 θ，使矩阵 $\widehat{Q}^*$ 每列中平方载荷 $\hat{q}_{ij}$ 的方差之和达到最大.
  - 更准确地说，定义

$$\tilde{q}_{ij}^* = \frac{\hat{q}_{ij}^*}{\hat{h}_j^*}$$

  - 方差最大化准则选择使下式子达到最大的 θ:

$$\mathcal{V} = \frac{1}{p}\sum_{\ell=1}^{k}\left\{\sum_{j=1}^{p}\left(\tilde{q}_{j\ell}^*\right)^4 - \frac{1}{p}\left[\sum_{j=1}^{p}\left(\tilde{q}_{j\ell}^*\right)^2\right]\right\}$$

因子模型的估计 (Estimation of the Factor Model)

- 例: 顾客对一款新产品的几个属性进行评分.

```
> round(FA, digits = 2)
     Load_F1 Load_F2 Communalities S_Variances
[1,]    0.56   -0.82          0.98          0.02
[2,]    0.78    0.52          0.88          0.12
[3,]    0.65   -0.75          0.98          0.02
[4,]    0.94    0.10          0.89          0.11
[5,]    0.80    0.54          0.93          0.07
```

属性 (变量) :		
味道		
物有所值		
口感		
适合作零食		
提供大量能量		
特征值	2.85	1.81
累积百分比	0.571	0.932

- 第一个因子和第二个因子的基本因子载荷几乎相同，这使得对这些因子的解释变得困难.

因子模型的估计 (Estimation of the Factor Model)

- 例: 顾客对一款新产品的几个属性进行评分.

  - 应用方差最大旋转法，我们得到因子载荷

	$\tilde{q}_1$	$\tilde{q}_2$
味道	0.02	**0.99**
物有所值	**0.94**	−0.01
口感	0.13	**0.98**
适合作零食	**0.84**	0.43
提供大量能量	**0.97**	−0.02

  - 变量 2、4、5 定义了因子 1，即营养因子.
  - 变量 1 和 3 定义了因子 2，即口味因子.

因子得分与策略 (Factor Scores and Strategies)

- 因子得分 (factor scores) 是每个观测值 $\boldsymbol{x}_i$, $(i = 1, 2, \ldots, n)$ 对应的不可观测随机向量 F_l, $(l = 1, 2, \ldots, k)$ 的估计值.
- Johnson 和 Wichern (1998 年) 介绍了三种方法，在实际应用中，这些方法得出的结果非常相似.
- 我们介绍回归的方法，其优点是简单且易于实施.
- 其思路是考虑 $\left(\boldsymbol{X} - \boldsymbol{\mu}\right)$ 与 $\boldsymbol{F}$ 的联合分布，然后进行回归分析.

因子得分与策略 (Factor Scores and Strategies)

$$\boldsymbol{X} = \mathcal{Q}\,\boldsymbol{F} + \boldsymbol{U} + \boldsymbol{\mu}$$

$\mathcal{Q}_{p\times k}$ 公因子载荷

$\boldsymbol{F}_{k\times 1}$ 公共因子

$\boldsymbol{U}_{p\times 1}$ 特殊因子

- 模型假设：

$$\mathrm{E}(\boldsymbol{F}) = \boldsymbol{0}$$

$$\mathrm{Var}(\boldsymbol{F}) = \mathcal{I}_k$$

$$\mathrm{E}(\boldsymbol{U}) = \boldsymbol{0}$$

$$\mathrm{Cov}\left(U_i\,,\,U_j\right) = 0\,,\quad i \neq j$$

$$\mathrm{Var}(\boldsymbol{U}) \triangleq \boldsymbol{\Psi} = \begin{pmatrix} \psi_{11} & & & \\ & \psi_{22} & & \\ & & \ddots & \\ & & & \psi_{pp} \end{pmatrix}$$

$$\mathrm{Cov}\left(\boldsymbol{F}\,,\,\boldsymbol{U}\right) = \boldsymbol{0}$$

$$\mathrm{Var}\begin{pmatrix}\boldsymbol{X}-\boldsymbol{\mu}\\ \boldsymbol{F}\end{pmatrix} = \mathrm{Var}\begin{pmatrix}\mathcal{Q}\,\boldsymbol{F}+\boldsymbol{U}\\ \boldsymbol{F}\end{pmatrix} = \mathrm{Var}\left[\begin{pmatrix}\mathcal{Q}\\ \mathcal{I}_k\end{pmatrix}\boldsymbol{F} + \begin{pmatrix}\boldsymbol{U}\\ \boldsymbol{0}\end{pmatrix}\right] = \mathrm{Var}\left[\begin{pmatrix}\mathcal{Q}\\ \mathcal{I}_k\end{pmatrix}\boldsymbol{F}\right] + \mathrm{Var}\left[\begin{pmatrix}\boldsymbol{U}\\ \boldsymbol{0}\end{pmatrix}\right]$$

$$= \begin{pmatrix}\mathcal{Q}\\ \mathcal{I}_k\end{pmatrix}\mathrm{Var}(\boldsymbol{F})\begin{pmatrix}\mathcal{Q}^{\mathrm{T}} & \mathcal{I}_k\end{pmatrix} + \begin{pmatrix}\boldsymbol{\Psi} & \\ & \boldsymbol{0}\end{pmatrix} = \begin{pmatrix}\mathcal{Q}\mathcal{Q}^{\mathrm{T}}+\boldsymbol{\Psi} & \mathcal{Q}\\ \mathcal{Q}^{\mathrm{T}} & \mathcal{I}_k\end{pmatrix}$$

$$= \begin{pmatrix}\boldsymbol{\Sigma} & \mathcal{Q}\\ \mathcal{Q}^{\mathrm{T}} & \mathcal{I}_k\end{pmatrix}_{(p+k)\times(p+k)}$$

$\mathcal{I}_k$

$\boldsymbol{\Sigma} = \mathcal{Q}\mathcal{Q}^{\mathrm{T}} + \boldsymbol{\Psi}$

因子得分与策略 (Factor Scores and Strategies)

- 在联合分布为正态分布的假设下，

$$\mathrm{Var}\begin{pmatrix} X-\mu \\ F \end{pmatrix} = \begin{pmatrix} \Sigma & \mathcal{Q} \\ \mathcal{Q}^{\mathrm{T}} & \mathcal{I}_k \end{pmatrix}_{(p+k)\times(p+k)} \Longrightarrow \begin{pmatrix} X \\ F \end{pmatrix} \sim N_{p+k}\left(\begin{pmatrix} \mu_{p\times 1} \\ \mathbf{0}_{k\times 1} \end{pmatrix}, \begin{pmatrix} \Sigma & \mathcal{Q} \\ \mathcal{Q}^{\mathrm{T}} & \mathcal{I}_k \end{pmatrix} \right)$$

定理 5.1 设 $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{pmatrix} \sim N_p(\boldsymbol{\mu}, \Sigma)$，$\boldsymbol{X}_1 \in \mathbb{R}^r$，$\boldsymbol{X}_2 \in \mathbb{R}^{p-r}$. 利用协方差矩阵的分块

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

定义

$$\boldsymbol{X}_{2\cdot 1} = \boldsymbol{X}_2 - \Sigma_{21}\Sigma_{11}^{-1}\boldsymbol{X}_1$$

则

$$\boldsymbol{X}_1 \sim N_r(\boldsymbol{\mu}_1, \Sigma_{11})$$
$$\boldsymbol{X}_{2\cdot 1} \sim N_{p-r}(\boldsymbol{\mu}_{2\cdot 1}, \Sigma_{22\cdot 1})$$

相互独立，其中

$$\boldsymbol{\mu}_{2\cdot 1} = \boldsymbol{\mu}_2 - \Sigma_{21}\Sigma_{11}^{-1}\boldsymbol{\mu}_1, \quad \Sigma_{22\cdot 1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$$

定理 5.3 给定 $\boldsymbol{X}_1 = \boldsymbol{x}_1$ 时 $\boldsymbol{X}_2$ 的条件分布亦是正态分布，其均值向量为 $\boldsymbol{\mu}_2 + \Sigma_{21}\Sigma_{11}^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1)$，协方差矩阵为 $\Sigma_{22\cdot 1}$，即

$$\left(\boldsymbol{X}_2 \,\middle|\, \boldsymbol{X}_1 = \boldsymbol{x}_1\right) \sim N_{p-r}\left(\boldsymbol{\mu}_2 + \Sigma_{21}\Sigma_{11}^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1),\ \Sigma_{22\cdot 1}\right)$$

- $\boldsymbol{F} \,|\, \boldsymbol{X}$ 的条件分布为多元正态分布，且

$$\mathrm{E}\left(\boldsymbol{F} \,\middle|\, \boldsymbol{X} = \boldsymbol{x}\right) = \mathcal{Q}^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{X} - \boldsymbol{\mu})$$

$$\mathrm{Var}\left(\boldsymbol{F} \,\middle|\, \boldsymbol{X} = \boldsymbol{x}\right) = \mathcal{I}_k - \mathcal{Q}^{\mathrm{T}}\Sigma^{-1}\mathcal{Q}$$

$$\Longrightarrow \quad \left(\boldsymbol{F} \,\middle|\, \boldsymbol{X} = \boldsymbol{x}\right) \sim N_k\left(\mathcal{Q}^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{X} - \boldsymbol{\mu}),\ \mathcal{I}_k - \mathcal{Q}^{\mathrm{T}}\Sigma^{-1}\mathcal{Q}\right)$$

因子得分与策略 (Factor Scores and Strategies)

- 在实际应用中，我们用相应的估计值替代未知的 $\mathcal{Q}$、Σ 和 $\boldsymbol{\mu}$，从而得到观测数据因子得分的估计值：

$$\hat{\boldsymbol{f}}_i = \hat{\mathcal{Q}}^{\mathrm{T}} \mathcal{S}^{-1} \left(\boldsymbol{x}_i - \bar{\boldsymbol{x}} \right)$$

  - 我们更倾向于使用原始样本协方差矩阵 $\mathcal{S}$ 作为 Σ 的估计量，而非因子分析的近似值 $\hat{\mathcal{Q}}\,\hat{\mathcal{Q}}^{\mathrm{T}} + \hat{\Psi}$.
  - 其目的是在面对因子数量的错误判定时，具备更强的稳健性.

$$\Longrightarrow \quad \left(\boldsymbol{F} \,\middle|\, \boldsymbol{X} = \boldsymbol{x} \right) \sim N_k \left(\mathcal{Q}^{\mathrm{T}} \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu}) \,,\, \mathcal{I}_k - \mathcal{Q}^{\mathrm{T}} \Sigma^{-1} \mathcal{Q} \right)$$

因子得分与策略 (Factor Scores and Strategies)

- 使用 $\mathscr{R}$ 而非 $\mathscr{S}$ 时，也可遵循同样的规则.
  - 标准化变量

$$\boldsymbol{Z} = \mathscr{D}_{\Sigma}^{-1/2}\left(\boldsymbol{X} - \boldsymbol{\mu}\right), \qquad \mathscr{D}_{\Sigma} = \begin{pmatrix} \sigma_{11} & & & \\ & \sigma_{22} & & \\ & & \ddots & \\ & & & \sigma_{pp} \end{pmatrix}$$

  - 此时，因子由下式给出

$$\hat{\boldsymbol{f}}_i = \hat{\mathscr{Q}}^{\mathrm{T}} \mathscr{R}^{-1} \boldsymbol{z}_i$$

$\hat{\mathscr{Q}}$ 通过相关矩阵 $\mathscr{R}$ 得到的载荷

$$\mathscr{D}_S = \begin{pmatrix} s_{11} & & & \\ & s_{22} & & \\ & & \ddots & \\ & & & s_{pp} \end{pmatrix}$$

$$\boldsymbol{z}_i = \mathscr{D}_S^{-1/2}\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right)$$

  - 如果因子通过正交矩阵 $\mathscr{G}$ 进行旋转，那么因子得分也必须相应地旋转，即

$$\hat{\boldsymbol{f}}_i^{*} = \hat{\mathscr{G}}^{\mathrm{T}} \hat{\boldsymbol{f}}_i$$

因子得分与策略 (Factor Scores and Strategies)

- 实用建议
  - 在因子分析的实际应用中，没有哪种方法绝对优于其它方法.
  - 然而，通过应用试错调整过程，数据的因子分析视角可以得到稳定.
  - 这就引出了以下步骤：

① 根据数据的相关结构和 (或) 特征值的碎石图，确定一个合理的因子数量，比如 $k = 2$ 或 3.

② 执行几种上述介绍的方法，包括旋转操作. 比较各个结果中的载荷、公因子方差和因子得分.

③ 如果结果显示出显著偏差，(根据因子得分) 检查是否存在异常值，并考虑改变因子数量 k.

因子得分与策略 (Factor Scores and Strategies)

- 实用建议
  - 对于大数据集，建议使用交叉验证方法.
    - 将样本数据拆分为训练集和验证集.
    - 使用训练样本和适当的方法估计因子模型.
    - 使用获得的参数来预测验证数据集当中的因子得分.
    - 预测的因子得分应当与仅使用验证数据集得出的因子得分具有可比性.
  - 这一稳定性标准也可适用于因子载荷和公因子方差.

因子得分与策略 (Factor Scores and Strategies)

- 因子分析与主成分分析的对比
  - 因子分析和主成分分析采用相同的数学工具(谱分解、投影……等).
  - 乍看起来，人们可能会得出结论，认为它们的视角和策略相同，因此会产生非常相似的结果.
  - 这两种数据分析方法之间存在显著差异.

并非如此！

因子得分与策略 (Factor Scores and Strategies)

- 因子分析与主成分分析的对比
  - 主成分分析 (PCA) 和因子分析 (Factor Analysis) 之间最大的区别源于模型的理念.
    - 因子分析设定了由固定数量的公共 (潜在) 因子构成的严格结构.
    - 主成分分析 (PCA) 按照重要性递减的顺序确定 p 个因子.
    - 在主成分分析中，最重要的因子是使投影方差最大化的那个因子.
    - 在因子分析中，最重要的因子是 (经过旋转后) 解释能力最强的那个因子. 这通常与第一主成分的方向不同.

因子得分与策略 (Factor Scores and Strategies)

- 因子分析与主成分分析的对比
  - 从实施角度来看：
    - 主成分分析 (PCA) 基于一种定义明确、独一无二的算法 (谱分解).
    - 拟合一个因子分析模型涉及多种数值计算方法.
  - 因子分析过程的不唯一性为进行主观解读提供了空间，因而会产生一系列不同的结果.
  - 这种数据分析的理念使得因子分析变得困难.
    - 特别是当模型设定涉及交叉验证以及基于数据驱动来选择因子数量时尤其困难.

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

X_1 : 人均犯罪率

X_2 : 划定用于大片住宅用地的比例

X_3 : 非零售商业用地的比例

X_4 : 查尔斯河 (与河相邻为 1，否则为 0)

X_5 : 一氧化氮浓度

X_6 : 每套住宅的平均房间数

X_7 : 1940年之前建造的自住房屋比例

X_8 : 到波士顿五个就业中心的加权距离

X_9 : 辐射状公路可达性指数

X_{10} : 每 10000 美元房产的全额税率

X_{11} : 学生与教师的比例

X_{12} : $1000(B-0.63)^2\ \mathrm{I}(B<0.63)$ 其中 B 是非洲裔美国人的比例

X_{13} : 人口中较低社会地位群体的占比

X_{14} : 自有住房的中位价值 (单位：1000 美元)

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 对这些变量采用此前作过的一系列变换.

$$\widetilde{X}_1 = \log X_1, \qquad \widetilde{X}_8 = \log X_8$$
$$\widetilde{X}_2 = \frac{X_2}{10}, \qquad \widetilde{X}_9 = \log X_9$$
$$\widetilde{X}_3 = \log X_3, \qquad \widetilde{X}_{10} = \log X_{10}$$
$$\widetilde{X}_4 = X_4, \qquad \widetilde{X}_{11} = \frac{\mathrm{e}^{0.4\times X_{11}}}{100}$$
$$\widetilde{X}_5 = \log X_5, \qquad \widetilde{X}_{12} = \frac{X_{12}}{100}$$
$$\widetilde{X}_6 = \log X_6, \qquad \widetilde{X}_{13} = \sqrt{X_{13}}$$
$$\widetilde{X}_7 = \frac{X_7^{2.5}}{10000}, \qquad \widetilde{X}_{14} = \log X_{14}$$

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 对这些变量采用此前作过的一系列变换.
  - 同样，变量 X_4 (Charles 河示性变量) 将被排除在外.
  - 与之前一样，使用标准化变量，分析基于相关矩阵进行.

```
# 读入数据
rm(list = ls(all = TRUE))
library(MASS)
data = Boston
head(data)
```

```
> head(data)
     crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat medv
1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98 24.0
2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14 21.6
3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03 34.7
4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94 33.4
5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33 36.2
6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21 28.7
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# transform data 变量数据的变换
xt = data
xt[, 1]  = log(data[, 1])
xt[, 2]  = data[, 2]/10
xt[, 3]  = log(data[, 3])
xt[, 5]  = log(data[, 5])
xt[, 6]  = log(data[, 6])
xt[, 7]  = (data[, 7]^(2.5))/10000
xt[, 8]  = log(data[, 8])
xt[, 9]  = log(data[, 9])
xt[, 10] = log(data[, 10])
xt[, 11] = exp(0.4 * data[, 11])/1000
xt[, 12] = data[, 12]/100
xt[, 13] = sqrt(data[, 13])
xt[, 14] = log(data[, 14])
data = xt[, -4] # 剔除变量 X_4
round(head(data), digits = 4)
```

```
> round(head(data), digits = 4)
     crim  zn  indus     nox     rm    age    dis    rad    tax ptratio  black  lstat   medv
1 -5.0640 1.8 0.8372 -0.6199 1.8833 3.4326 1.4085 0.0000 5.6904  0.4549 3.9690 2.2316 3.1781
2 -3.6005 0.0 1.9559 -0.7572 1.8596 5.5296 1.6028 0.6931 5.4889  1.2365 3.9690 3.0232 3.0727
3 -3.6012 0.0 1.9559 -0.7572 1.9720 2.9181 1.6028 0.6931 5.4889  1.2365 3.9283 2.0075 3.5467
4 -3.4305 0.0 0.7793 -0.7809 1.9456 1.4196 1.8021 1.0986 5.4027  1.7722 3.9463 1.7146 3.5086
5 -2.6729 0.0 0.7793 -0.7809 1.9667 2.1627 1.8021 1.0986 5.4027  1.7722 3.9690 2.3087 3.5891
6 -3.5116 0.0 0.7793 -0.7809 1.8610 2.6399 1.8021 1.0986 5.4027  1.7722 3.9412 2.2825 3.3569
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# rename variables 重新命名变量
colnames(data) = c("X1", "X2", "X3", "X5", "X6", "X7", "X8", "X9", "X10", "X11", "X12", "X13", "X14")
round(head(data), digits = 4)
```

```
> round(head(data), digits = 4)
       X1  X2     X3      X5     X6     X7     X8     X9    X10    X11    X12    X13    X14
1 -5.0640 1.8 0.8372 -0.6199 1.8833 3.4326 1.4085 0.0000 5.6904 0.4549 3.9690 2.2316 3.1781
2 -3.6005 0.0 1.9559 -0.7572 1.8596 5.5296 1.6028 0.6931 5.4889 1.2365 3.9690 3.0232 3.0727
3 -3.6012 0.0 1.9559 -0.7572 1.9720 2.9181 1.6028 0.6931 5.4889 1.2365 3.9283 2.0075 3.5467
4 -3.4305 0.0 0.7793 -0.7809 1.9456 1.4196 1.8021 1.0986 5.4027 1.7722 3.9463 1.7146 3.5086
5 -2.6729 0.0 0.7793 -0.7809 1.9667 2.1627 1.8021 1.0986 5.4027 1.7722 3.9690 2.3087 3.5891
6 -3.5116 0.0 0.7793 -0.7809 1.8610 2.6399 1.8021 1.0986 5.4027 1.7722 3.9412 2.2825 3.3569
```

```
# standardize variables 变量标准化
da  = scale(data)
round(head(da), digits = 4)
```

```
> round(head(da), digits = 4)
       X1      X2      X3      X5     X6      X7     X8      X9     X10     X11    X12     X13    X14
1 -1.9813  0.2845 -1.7027 -0.0490 0.4577 -0.4565 0.4087 -2.1349 -0.6081 -1.2428 0.4406 -1.2015 0.3512
2 -1.3043 -0.4872 -0.2630 -0.7302 0.2467  0.1314 0.7688 -1.3426 -1.1163 -0.6698 0.4406 -0.3996 0.0934
3 -1.3047 -0.4872 -0.2630 -0.7302 1.2476 -0.6008 0.7688 -1.3426 -1.1163 -0.6698 0.3960 -1.4285 1.2531
4 -1.2257 -0.4872 -1.7772 -0.8480 1.0128 -1.0209 1.1381 -0.8791 -1.3339 -0.2770 0.4158 -1.7252 1.1597
5 -0.8753 -0.4872 -1.7772 -0.8480 1.2003 -0.8126 1.1381 -0.8791 -1.3339 -0.2770 0.4406 -1.1234 1.3567
6 -1.2632 -0.4872 -1.7772 -0.8480 0.2592 -0.6788 1.1381 -0.8791 -1.3339 -0.2770 0.4102 -1.1499 0.7887
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# correlation matrix 计算相关矩阵
dat = cor(da)
round(dat, digits = 4)
```

```
> round(dat, digits = 4)
        X1      X2      X3      X5      X6      X7      X8      X9     X10     X11     X12     X13     X14
X1  1.0000 -0.5171  0.7396  0.8070 -0.3242  0.6968 -0.7439  0.8389  0.8100  0.4539 -0.4788  0.6223 -0.5672
X2 -0.5171  1.0000 -0.6559 -0.5685  0.3094 -0.5263  0.5907 -0.3506 -0.3059 -0.3501  0.1755 -0.4522  0.3633
X3  0.7396 -0.6559  1.0000  0.7505 -0.4296  0.6581 -0.7303  0.5805  0.6593  0.4547 -0.3311  0.6214 -0.5539
X5  0.8070 -0.5685  0.7505  1.0000 -0.3183  0.7831 -0.8600  0.6129  0.6683  0.3437 -0.3793  0.6094 -0.5153
X6 -0.3242  0.3094 -0.4296 -0.3183  1.0000 -0.2767  0.2807 -0.2134 -0.3064 -0.3208  0.1297 -0.6394  0.6104
X7  0.6968 -0.5263  0.6581  0.7831 -0.2767  1.0000 -0.7960  0.4687  0.5409  0.3778 -0.2859  0.6371 -0.4821
X8 -0.7439  0.5907 -0.7303 -0.8600  0.2807 -0.7960  1.0000 -0.5421 -0.5996 -0.3217  0.3248 -0.5555  0.4057
X9  0.8389 -0.3506  0.5805  0.6129 -0.2134  0.4687 -0.5421  1.0000  0.8205  0.3982 -0.4113  0.4612 -0.4345
X10 0.8100 -0.3059  0.6593  0.6683 -0.3064  0.5409 -0.5996  0.8205  1.0000  0.4763 -0.4279  0.5335 -0.5572
X11 0.4539 -0.3501  0.4547  0.3437 -0.3208  0.3778 -0.3217  0.3982  0.4763  1.0000 -0.2047  0.4338 -0.5082
X12 -0.4788 0.1755 -0.3311 -0.3793  0.1297 -0.2859  0.3248 -0.4113 -0.4279 -0.2047  1.0000 -0.3610  0.4024
X13 0.6223 -0.4522  0.6214  0.6094 -0.6394  0.6371 -0.5555  0.4612  0.5335  0.4338 -0.3610  1.0000 -0.8250
X14 -0.5672 0.3633 -0.5539 -0.5153  0.6104 -0.4821  0.4057 -0.4345 -0.5572 -0.5082  0.4024 -0.8250  1.0000
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 在上一节中，我们介绍了因子分析的实际应用.
  - 基于主成分，我们选取了三个因子，然后运用以下方法来进行因子分析
    - MLM: 极大似然方法
    - PFM: 主因子方法
    - PCM: 主成分方法
  - 为便于说明，将分别进行有方差最大化旋转和无方差最大化旋转的极大似然方法.

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# R 中的因子分析
?factanal
```

Factor Analysis

Description

Perform maximum-likelihood factor analysis on a covariance matrix or data matrix.

Usage

```
factanal(x, factors, data = NULL, covmat = NULL, n.obs = NA,
         scores = c("none", "regression", "Bartlett"),
         rotation = "varimax", ...)
```

Arguments

```
x    A formula or a numeric matrix or an object that can be coerced to a
     numeric matrix.
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# R 中的因子分析
?factanal
```

Factor Analysis

Description

Perform maximum-likelihood factor analysis on a covariance matrix or data matrix.

Usage

```
factanal(x, factors, data = NULL, covmat = NULL, n.obs = NA,
         scores = c("none", "regression", "Bartlett"),
         rotation = "varimax", ...)
```

Arguments

```
factors   The number of factors to be fitted.
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# R 中的因子分析
?factanal
```

Factor Analysis

Description

Perform maximum-likelihood factor analysis on a covariance matrix or data matrix.

Usage

```
factanal(x, factors, data = NULL, covmat = NULL, n.obs = NA,
         scores = c("none", "regression", "Bartlett"),
         rotation = "varimax", ...)
```

Arguments

```
data      An optional data frame, used only if x is a formula. By default
          the variables are taken from environment(formula).
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# R 中的因子分析
?factanal
```

Factor Analysis

Description

Perform maximum-likelihood factor analysis on a covariance matrix or data matrix.

Usage

```
factanal(x, factors, data = NULL, covmat = NULL, n.obs = NA,
         scores = c("none", "regression", "Bartlett"),
         rotation = "varimax", ...)
```

Arguments

```
covmat    A covariance matrix, or a covariance list as returned by cov.wt.
          Of course, correlation matrices are covariance matrices.
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# R 中的因子分析
?factanal
```

Factor Analysis

Description

Perform maximum-likelihood factor analysis on a covariance matrix or data matrix.

Usage

```
factanal(x, factors, data = NULL, covmat = NULL, n.obs = NA,
         scores = c("none", "regression", "Bartlett"),
         rotation = "varimax", ...)
```

Arguments

```
n.obs    The number of observations, used if covmat is a covariance matrix.
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# R 中的因子分析
?factanal
```

Factor Analysis

Description

Perform maximum-likelihood factor analysis on a covariance matrix or data matrix.

Usage

```
factanal(x, factors, data = NULL, covmat = NULL, n.obs = NA,
         scores = c("none", "regression", "Bartlett"),
         rotation = "varimax", ...)
```

Arguments

```
scores   Type of scores to produce, if any. The default is
         none, "regression" gives Thompson's scores, "Bartlett" given
         Bartlett's weighted least-squares scores.
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# R 中的因子分析
?factanal
```

Factor Analysis

Description

Perform maximum-likelihood factor analysis on a covariance matrix or data matrix.

Usage

```
factanal(x, factors, data = NULL, covmat = NULL, n.obs = NA,
         scores = c("none", "regression", "Bartlett"),
         rotation = "varimax", ...)
```

Arguments

```
rotation character. "none" or the name of a function to be used to rotate
         the factors: it will be called with first argument the loadings
         matrix, and should return a list with component loadings giving
         the rotated loadings, or just the rotated loadings.
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 不进行方差最大化旋转的极大似然因子分析.

```
# Maximum Likelihood Factor Analysis without varimax rotation: factanal 极大似然法的因子分析，未作 varimax 旋转
mlm  = factanal(da, 3, rotation = "none", covmat = dat)
str(mlm)
```

```
> str(mlm)
List of 10
$ converged   : logi TRUE
$ loadings    : 'loadings' num [1:13, 1:3] 0.929 -0.582 0.819 0.879 -0.445 ...
 ..- attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:13] "X1" "X2" "X3" "X5" ...
 .. ..$ : chr [1:3] "Factor1" "Factor2" "Factor3"
$ uniquenesses: Named num [1:13] 0.0964 0.5752 0.3091 0.1439 0.5188 ...
 ..- attr(*, "names")= chr [1:13] "X1" "X2" "X3" "X5" ...
$ correlation : num [1:13, 1:13] 1 -0.517 0.74 0.807 -0.324 ...
 ..- attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:13] "X1" "X2" "X3" "X5" ...
 .. ..$ : chr [1:13] "X1" "X2" "X3" "X5" ...
$ criteria    : Named num [1:3] 0.616 30 30
 ..- attr(*, "names")= chr [1:3] "objective" "counts.function" "counts.gradient"
$ factors     : num 3
$ dof         : num 42
$ method      : chr "mle"
$ n.obs       : logi NA
$ call        : language factanal(x = da, factors = 3, covmat = dat, rotation = "none")
- attr(*, "class")= chr "factanal"
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# estimated factor loadings 因子载荷的估计值
load = mlm$loadings
load
```

```
> load

Loadings:
    Factor1 Factor2 Factor3
X1   0.929   0.165   0.111
X2  -0.582           0.290
X3   0.819          -0.138
X5   0.879          -0.272
X6  -0.445   0.531
X7   0.784          -0.355
X8  -0.829  -0.157   0.411
X9   0.795   0.306   0.405
X10  0.826   0.140   0.291
X11  0.505  -0.185   0.155
X12 -0.470          -0.163
X13  0.760  -0.506
X14 -0.694   0.591  -0.180

                Factor1 Factor2 Factor3
SS loadings       6.998   1.099   0.818
Proportion Var    0.538   0.085   0.063
Cumulative Var    0.538   0.623   0.686
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# the estimated factor loadings matrix 因子载荷矩阵
ld = cbind(load[, 1], load[, 2], load[, 3])
round(ld, digits = 4)
```

```
> round(ld, digits = 4)
       [,1]    [,2]    [,3]
X1   0.9295  0.1653  0.1107
X2  -0.5823  0.0378  0.2903
X3   0.8192 -0.0296 -0.1379
X5   0.8789  0.0988 -0.2719
X6  -0.4447  0.5310 -0.0378
X7   0.7836 -0.0148 -0.3554
X8  -0.8294 -0.1571  0.4110
X9   0.7955  0.3062  0.4053
X10  0.8262  0.1400  0.2906
X11  0.5051 -0.1851  0.1552
X12 -0.4701  0.0227 -0.1627
X13  0.7601 -0.5058 -0.0072
X14 -0.6942  0.5906 -0.1797
```

$$X = \mathcal{Q}\,F + U + \mu$$

$$= \widehat{\mathcal{Q}}$$

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# communalities are calculated 计算公共因子方差
com = diag(ld %*% t(ld))
round(com, digits = 4)
```

公因子方差: $h_j^2 = \sum_{\ell=1}^{k} q_{j\ell}^2$

```
> round(com, digits = 4)
    X1     X2     X3     X5     X6     X7     X8     X9    X10    X11    X12    X13    X14
0.9036 0.4248 0.6909 0.8561 0.4812 0.7406 0.8815 0.8908 0.7867 0.3135 0.2480 0.8337 0.8630
```

```
# specific variances are calculated 计算特殊因子方差
psi  = diag(dat) - diag(ld %*% t(ld))
round(psi, digits = 4)
```

特殊因子方差: $\psi_{jj} = \sigma_{jj} - h_j^2$

```
> round(psi, digits = 4)
    X1     X2     X3     X5     X6     X7     X8     X9    X10    X11    X12    X13    X14
0.0964 0.5752 0.3091 0.1439 0.5188 0.2594 0.1185 0.1092 0.2133 0.6865 0.7520 0.1663 0.1370
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# 因子载荷、公共因子方差、特殊因子方差的矩阵
tbl  = cbind(L_F1 = load[, 1], L_F2 = load[, 2], L_F3 = load[, 3], com, psi)
round(tbl, digits = 4)
```

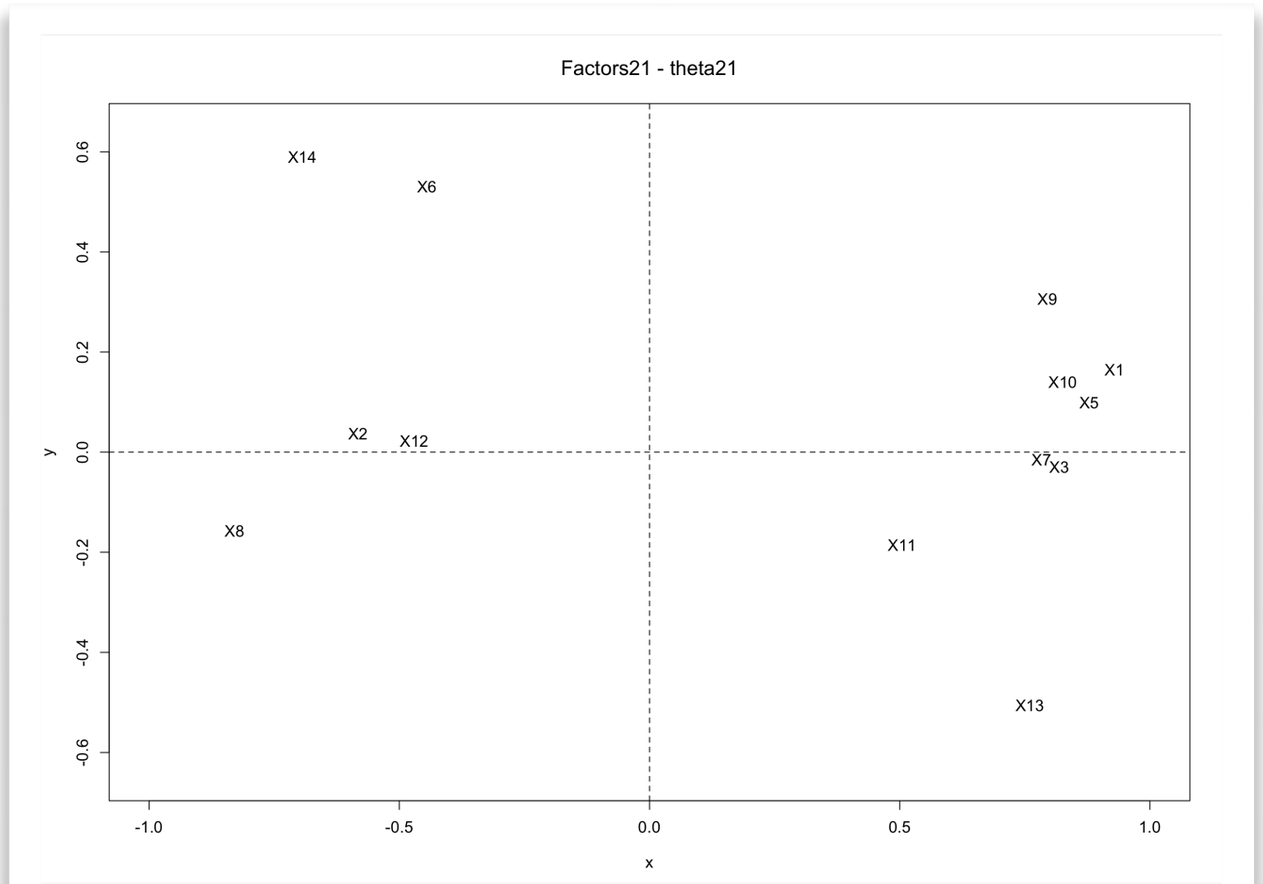
```
> round(tbl, digits = 4)
       L_F1    L_F2    L_F3    com    psi
X1   0.9295  0.1653  0.1107 0.9036 0.0964
X2  -0.5823  0.0378  0.2903 0.4248 0.5752
X3   0.8192 -0.0296 -0.1379 0.6909 0.3091
X5   0.8789  0.0988 -0.2719 0.8561 0.1439
X6  -0.4447  0.5310 -0.0378 0.4812 0.5188
X7   0.7836 -0.0148 -0.3554 0.7406 0.2594
X8  -0.8294 -0.1571  0.4110 0.8815 0.1185
X9   0.7955  0.3062  0.4053 0.8908 0.1092
X10  0.8262  0.1400  0.2906 0.7867 0.2133
X11  0.5051 -0.1851  0.1552 0.3135 0.6865
X12 -0.4701  0.0227 -0.1627 0.2480 0.7520
X13  0.7601 -0.5058 -0.0072 0.8337 0.1663
X14 -0.6942  0.5906 -0.1797 0.8630 0.1370
```

$\hat{q}_1$ $\hat{q}_2$ $\hat{q}_3$ $\hat{h}_j^2$ $\hat{\psi}_{jj}$

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

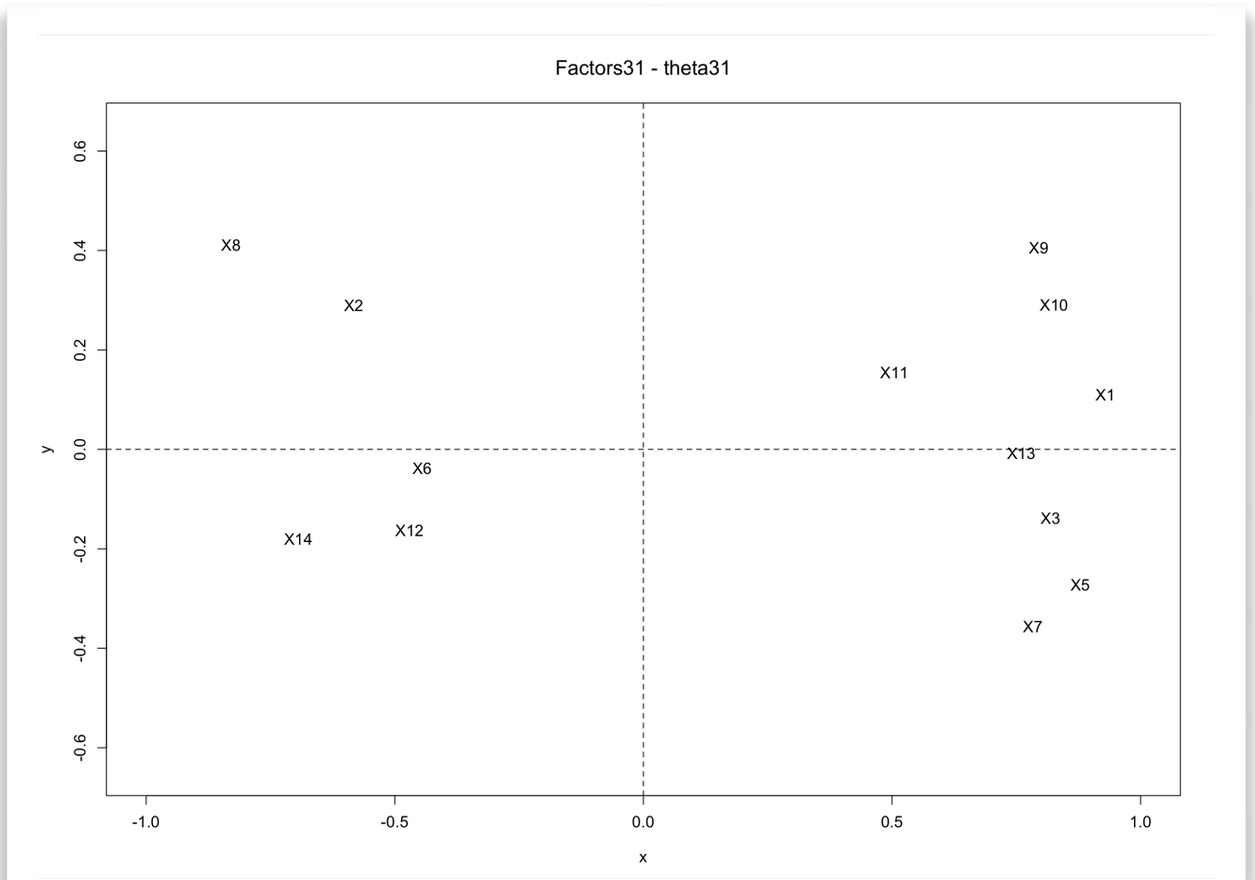
```
# plot first factor against second 变量在因子 1 和因子 2 平面上的散点图
graphics.off()
plot(load[, 1], load[, 2], type = "n", xlab = "x", ylab = "y", main = "Factors21 - theta21",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1.0, 1.0), ylim = c(-0.6, 0.6), asp = 1)
text(load[, 1], load[, 2], colnames(data), cex = 1.1)
abline(h = 0, v = 0, lty = 2)
```



Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

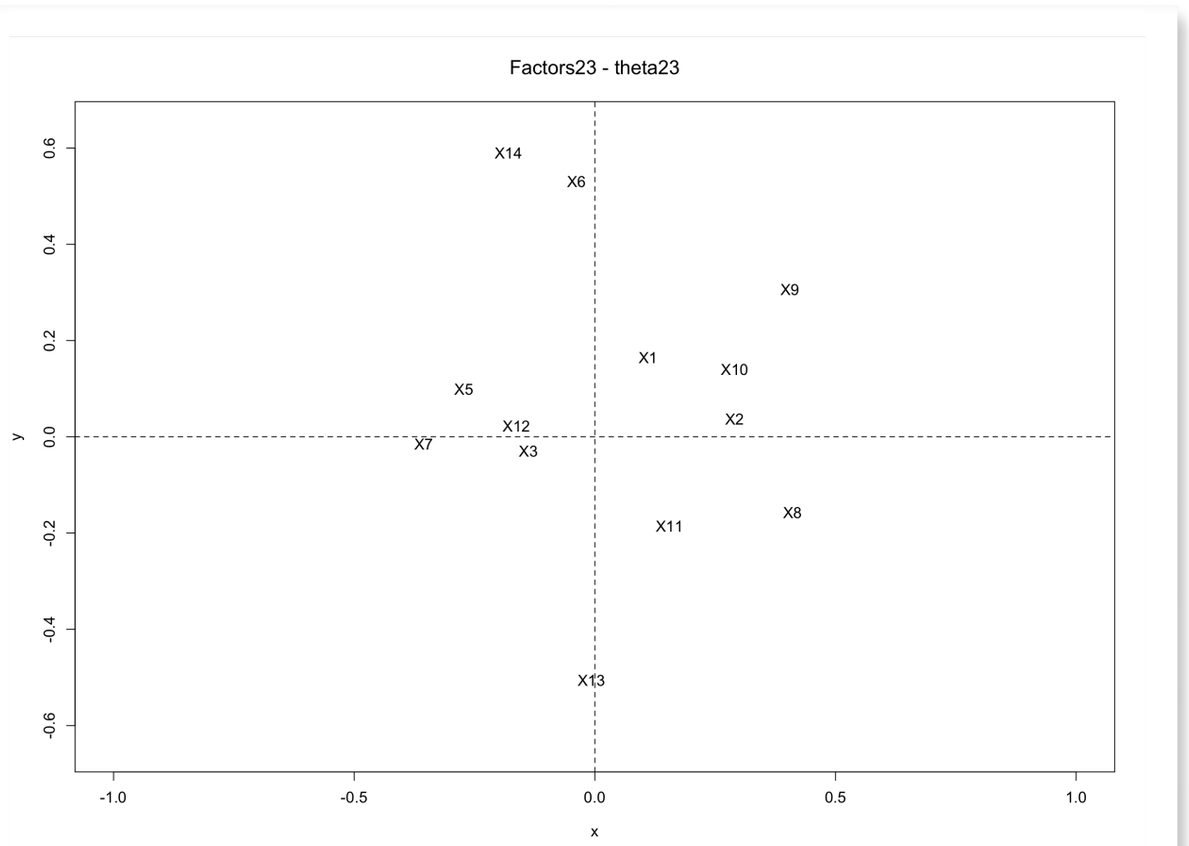
```
# plot first factor against third 变量在因子 1 和因子 3 平面上的散点图
graphics.off()
plot(load[, 1], load[, 3], type = "n", xlab = "x", ylab = "y", main = "Factors31 - theta31",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1.0, 1.0), ylim = c(-0.6, 0.6), asp = 1)
text(load[, 1], load[, 3], colnames(data), cex = 1.1)
abline(h = 0, v = 0, lty = 2)
```



Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

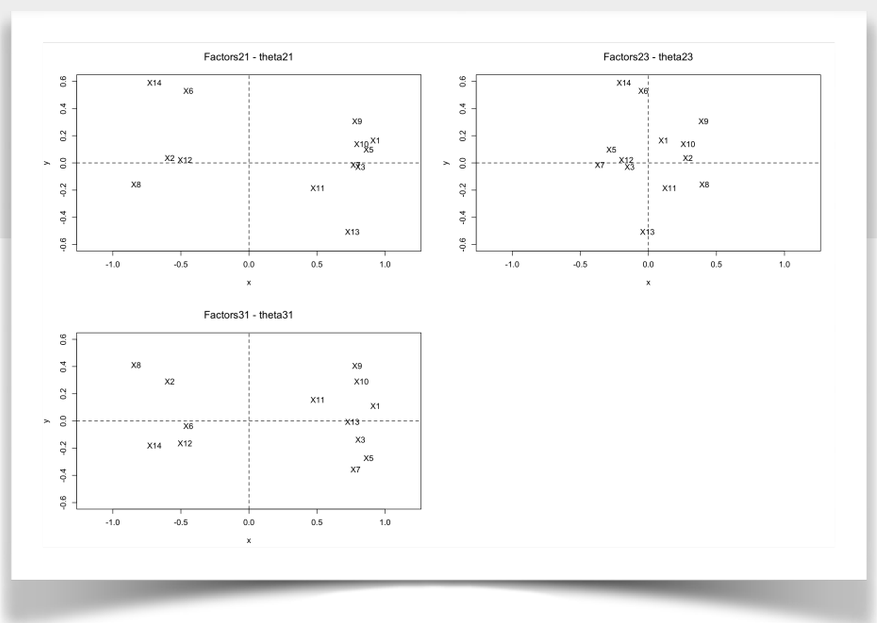
```
# plot second factor against third 变量在因子 2 和因子 3 平面上的散点图
graphics.off()
plot(load[, 3], load[, 2], type = "n", xlab = "x", ylab = "y", main = "Factors23 - theta23",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1.0, 1.0),
    ylim = c(-0.6, 0.6), asp = 1)
text(load[, 3], load[, 2], colnames(data), cex = 1.1)
abline(h = 0, v = 0, lty = 2)
```



Boston Housing

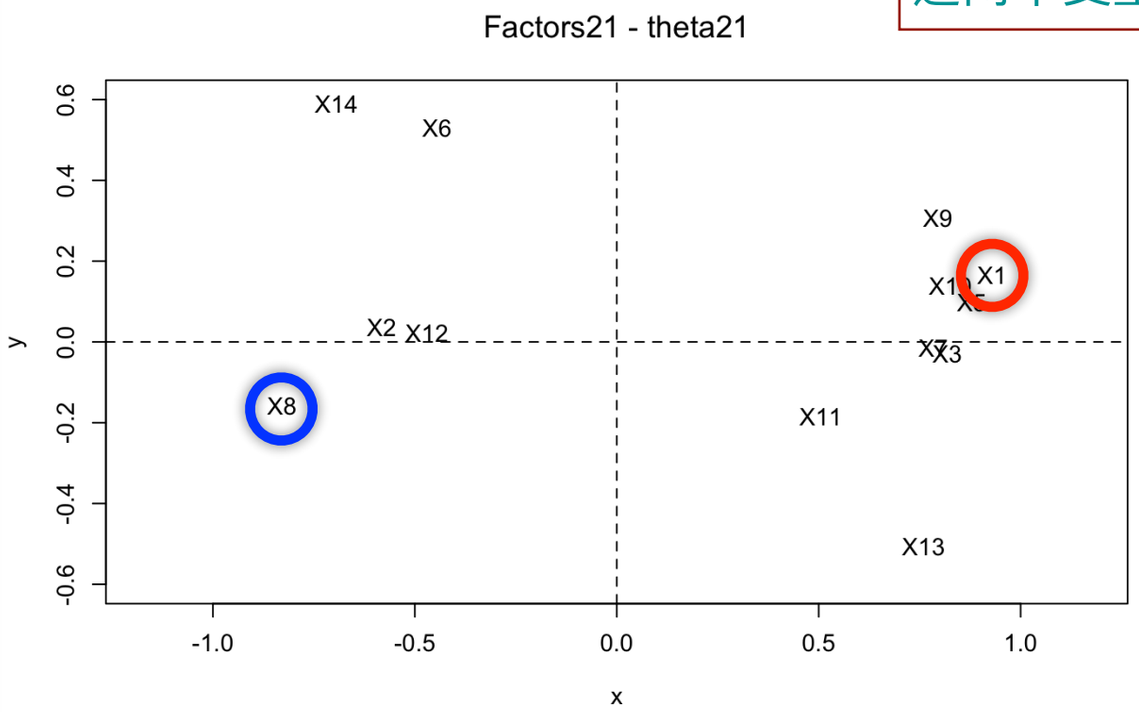
- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

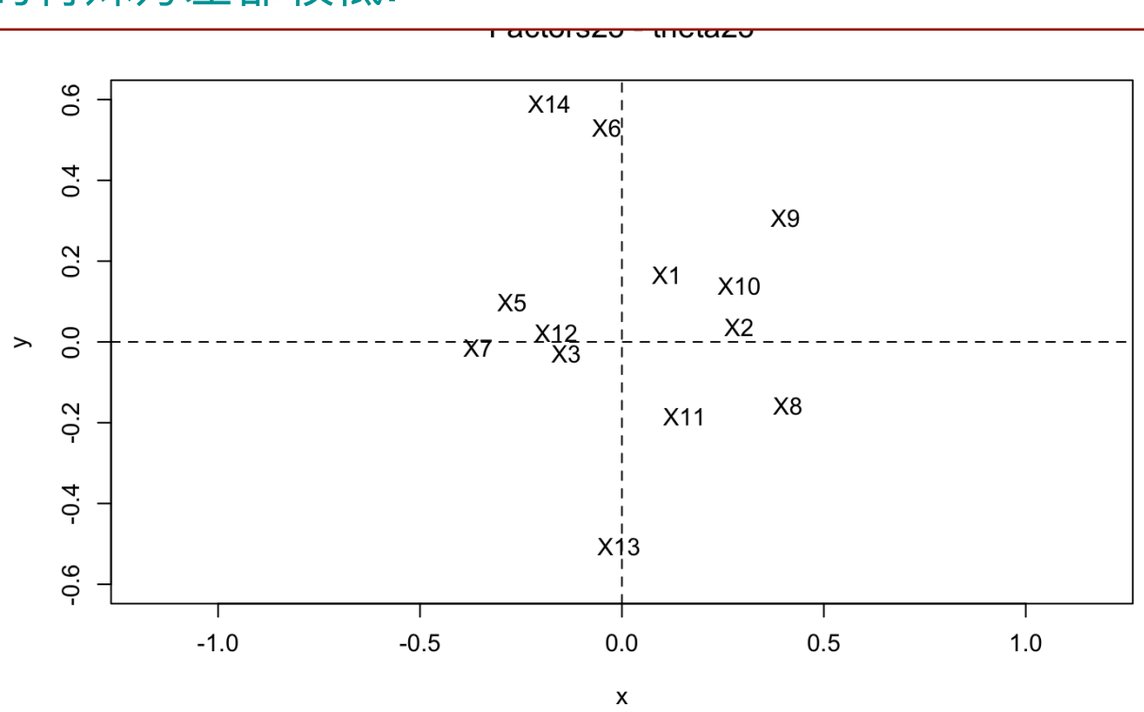
```
# 三张图放在一起
graphics.off()
par(mfcol = c(2, 2))
plot(load[, 1], load[, 2], type = "n", xlab = "x", ylab = "y", main = "Factors21 - theta21",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1.0, 1.0), ylim = c(-0.6, 0.6), asp = 1)
text(load[, 1], load[, 2], colnames(data), cex = 1.1)
abline(h = 0, v = 0, lty = 2)

plot(load[, 1], load[, 3], type = "n", xlab = "x", ylab = "y", main = "Factors31 - theta31",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1.0, 1.0), ylim = c(-0.6, 0.6), asp = 1)
text(load[, 1], load[, 3], colnames(data), cex = 1.1)
abline(h = 0, v = 0, lty = 2)

plot(load[, 3], load[, 2], type = "n", xlab = "x", ylab = "y", main = "Factors23 - theta23",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1.0, 1.0),
    ylim = c(-0.6, 0.6), asp = 1)
text(load[, 3], load[, 2], colnames(data), cex = 1.1)
abline(h = 0, v = 0, lty = 2)
```

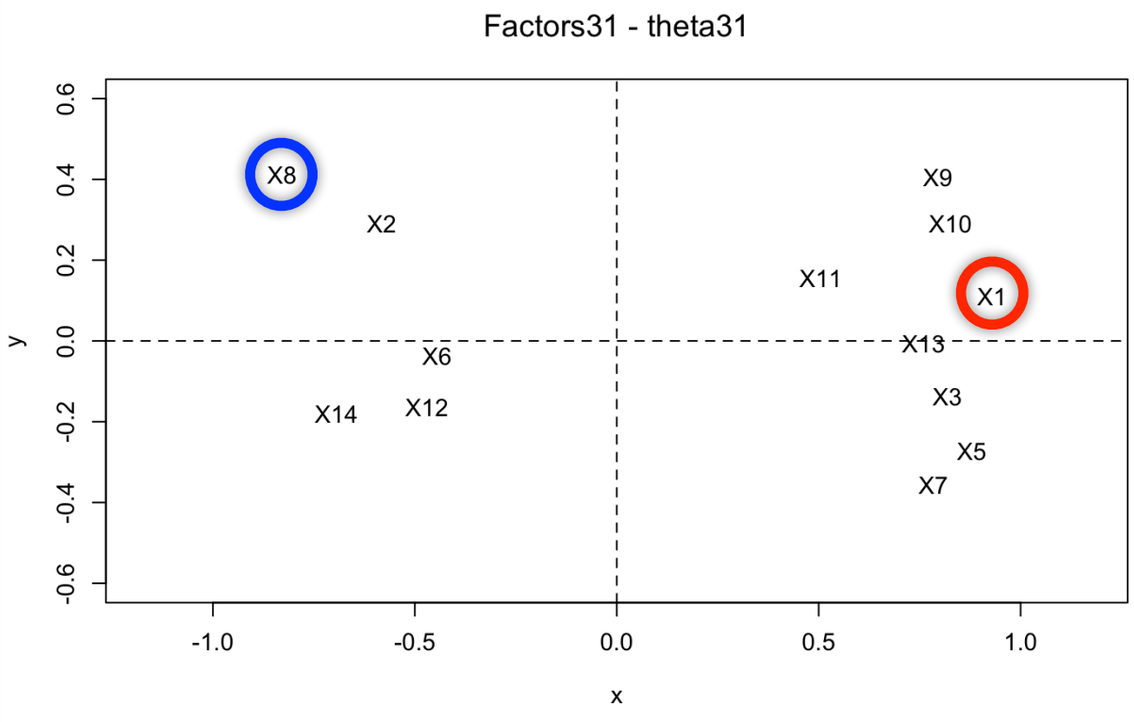


Boston Housing

因子 1 大致可解释为“生活质量因子”，因为它与诸如 X_1 (人均犯罪率) 等变量呈正相关，与 X_8 (到波士顿五个就业中心的加权距离) 呈负相关，且这两个变量的特殊方差都较低.



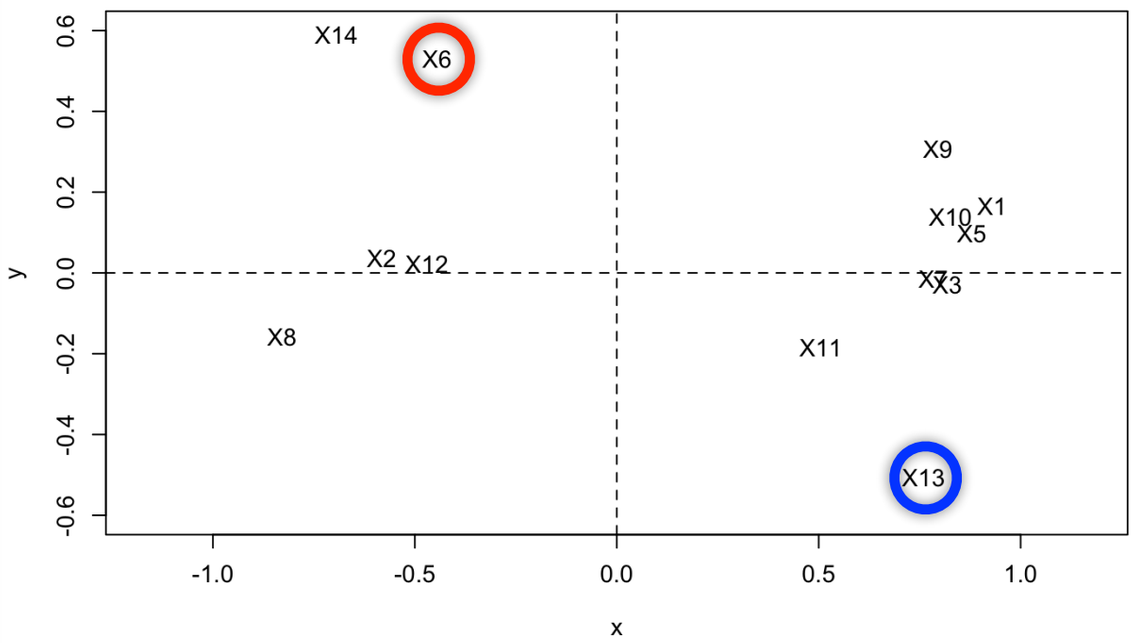


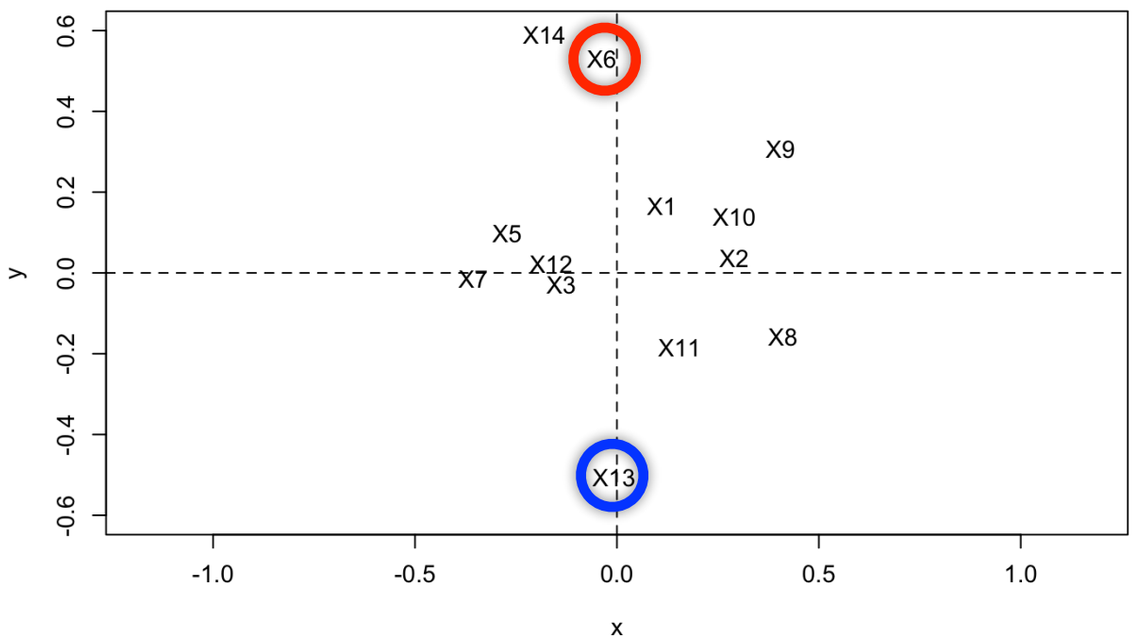


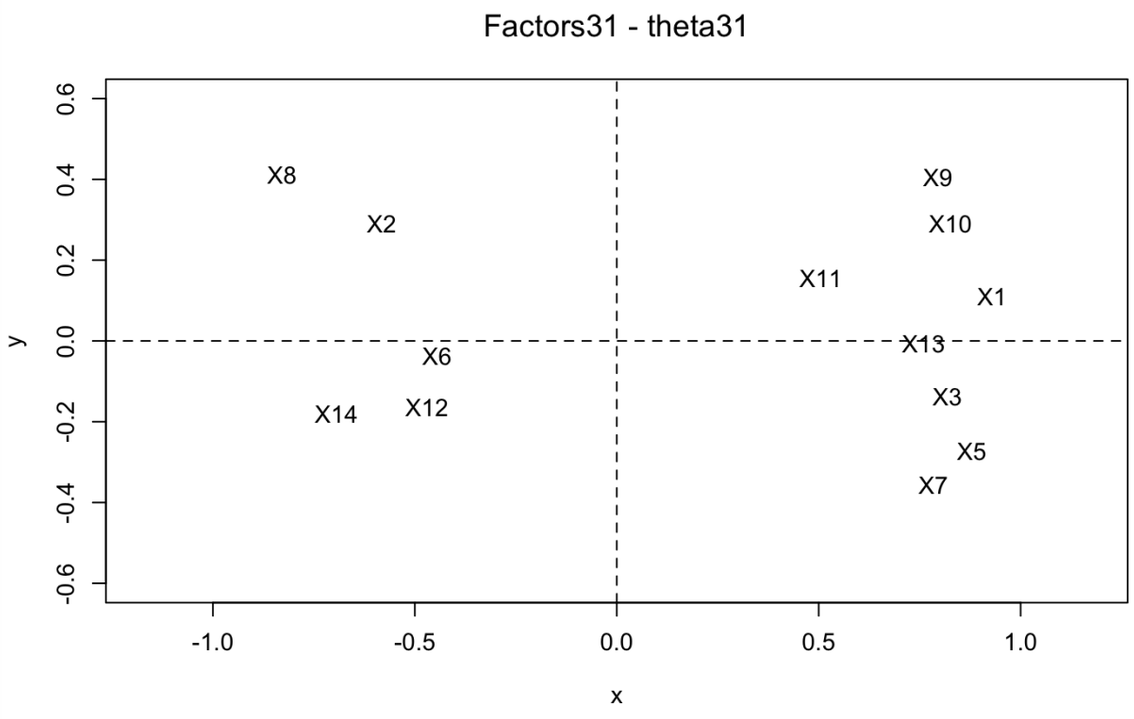

```
> round(tbl, digits = 4)
       L_F1    L_F2    L_F3    com    psi
X1   0.9295  0.1653  0.1107 0.9036 0.0964
X2  -0.5823  0.0378  0.2903 0.4248 0.5752
X3   0.8192 -0.0296 -0.1379 0.6909 0.3091
X5   0.8789  0.0988 -0.2719 0.8561 0.1439
X6  -0.4447  0.5310 -0.0378 0.4812 0.5188
X7   0.7836 -0.0148 -0.3554 0.7406 0.2594
X8  -0.8294 -0.1571  0.4110 0.8815 0.1185
X9   0.7955  0.3062  0.4053 0.8908 0.1092
X10  0.8262  0.1400  0.2906 0.7867 0.2133
X11  0.5051 -0.1851  0.1552 0.3135 0.6865
X12 -0.4701  0.0227 -0.1627 0.2480 0.7520
X13  0.7601 -0.5058 -0.0072 0.8337 0.1663
X14 -0.6942  0.5906 -0.1797 0.8630 0.1370
```

Boston Housing

第二个因子可被解释为 “居住因子”，因为它与变量 X_6 (每套住宅的平均房间数) 和 X_{13} (人口中较低社会地位群体的占比) 高度相关.








```
> round(tbl, digits = 4)
       L_F1    L_F2    L_F3    com    psi
X1   0.9295  0.1653  0.1107 0.9036 0.0964
X2  -0.5823  0.0378  0.2903 0.4248 0.5752
X3   0.8192 -0.0296 -0.1379 0.6909 0.3091
X5   0.8789  0.0988 -0.2719 0.8561 0.1439
X6  -0.4447  0.5310 -0.0378 0.4812 0.5188
X7   0.7836 -0.0148 -0.3554 0.7406 0.2594
X8  -0.8294 -0.1571  0.4110 0.8815 0.1185
X9   0.7955  0.3062  0.4053 0.8908 0.1092
X10  0.8262  0.1400  0.2906 0.7867 0.2133
X11  0.5051 -0.1851  0.1552 0.3135 0.6865
X12 -0.4701  0.0227 -0.1627 0.2480 0.7520
X13  0.7601 -0.5058 -0.0072 0.8337 0.1663
X14 -0.6942  0.5906 -0.1797 0.8630 0.1370
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# R 中的因子旋转
?varimax
```

Rotation Methods for Factor Analysis

Description

The function 'rotate' loading matrices in factor analysis.

Usage

```
varimax(x, normalize = TRUE, eps = 1e-5)
```

Arguments

x	A loadings matrix, with p rows and $k < p$ columns
normalize	logical. Should Kaiser normalization be performed? If so the rows of x are re-scaled to unit length before rotation, and scaled back afterwards.
eps	The tolerance for stopping: the relative change in the sum of singular values.

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# rotates the factor loadings matrix 因子载荷矩阵的旋转
var = varimax(ld)
str(var)
```

```
> str(var)
List of 2
 $ loadings: 'loadings' num [1:13, 1:3] 0.7247 -0.1587 0.4105 0.4141 -0.0799 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:13] "X1" "X2" "X3" "X5" ...
  .. ..$ : NULL
 $ rotmat  : num [1:3, 1:3] 0.627 0.421 0.656 -0.43 0.889 ...
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# estimated factor loadings after varimax 旋转后的因子载荷
load = var$loadings
round(load, digits = 4)
```

```
# 旋转后的因子载荷矩阵
vl   = cbind(load[, 1], load[, 2], load[, 3])
round(vl, digits = 4)
```

```
# 旋转矩阵
rm = var$rotmat
round(rm, digits = 4)
```

```
> round(rm, digits = 4)
       [,1]    [,2]    [,3]
[1,] 0.6267 -0.4300 -0.6498
[2,] 0.4210  0.8886 -0.1820
[3,] 0.6557 -0.1595  0.7379
```

```
> round(vl, digits = 4)
       [,1]    [,2]    [,3]
X1   0.7247 -0.2705 -0.5525
X2  -0.1587  0.2377  0.5858
X3   0.4105 -0.3566 -0.6287
X5   0.4141 -0.2468 -0.7898
X6  -0.0799  0.6691  0.1644
X7   0.2518 -0.2934 -0.7688
X8  -0.3164  0.1515  0.8709
X9   0.8932 -0.1347 -0.2736
X10  0.7673 -0.2772 -0.3480
X11  0.3405 -0.4065 -0.1800
X12 -0.3917  0.2483  0.1813
X13  0.2587 -0.7752 -0.4072
X14 -0.3043  0.8520  0.2111
```

```
> round(load, digits = 4)

Loadings:
    [,1]   [,2]   [,3]
X1   0.725 -0.270 -0.552
X2  -0.159  0.238  0.586
X3   0.410 -0.357 -0.629
X5   0.414 -0.247 -0.790
X6          0.669  0.164
X7   0.252 -0.293 -0.769
X8  -0.316  0.152  0.871
X9   0.893 -0.135 -0.274
X10  0.767 -0.277 -0.348
X11  0.340 -0.406 -0.180
X12 -0.392  0.248  0.181
X13  0.259 -0.775 -0.407
X14 -0.304  0.852  0.211

                 [,1]  [,2]  [,3]
SS loadings     2.876 2.523 3.516
Proportion Var  0.221 0.194 0.270
Cumulative Var  0.221 0.415 0.686
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

```
# communalities are calculated 计算公共因子方差
com  = diag(vl %*% t(vl))
round(com, digits = 4)
```

```
> round(com, digits = 4)
    X1     X2     X3     X5     X6     X7     X8     X9    X10    X11    X12    X13    X14
0.9036 0.4248 0.6909 0.8561 0.4812 0.7406 0.8815 0.8908 0.7867 0.3135 0.2480 0.8337 0.8630
```

```
# specific variances are calculated 计算特殊因子方差
psi  = diag(dat) - diag(vl %*% t(vl))
round(psi, digits = 4)
```

```
> round(psi, digits = 4)
    X1     X2     X3     X5     X6     X7     X8     X9    X10    X11    X12    X13    X14
0.0964 0.5752 0.3091 0.1439 0.5188 0.2594 0.1185 0.1092 0.2133 0.6865 0.7520 0.1663 0.1370
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

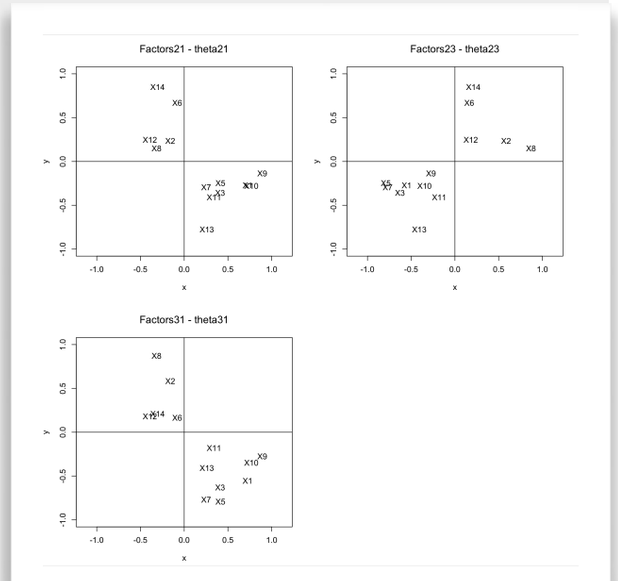
```
# 旋转后的因子载荷、公共因子方差、特殊因子方差的矩阵
tbl  = cbind(RL_F1 = load[, 1], RL_F2 = load[, 2], RL_F3 = load[, 3], com, psi)
round(tbl, digits = 4)
```

```
> round(tbl, digits = 4)
      RL_F1   RL_F2   RL_F3    com    psi
X1   0.7247 -0.2705 -0.5525 0.9036 0.0964
X2  -0.1587  0.2377  0.5858 0.4248 0.5752
X3   0.4105 -0.3566 -0.6287 0.6909 0.3091
X5   0.4141 -0.2468 -0.7898 0.8561 0.1439
X6  -0.0799  0.6691  0.1644 0.4812 0.5188
X7   0.2518 -0.2934 -0.7688 0.7406 0.2594
X8  -0.3164  0.1515  0.8709 0.8815 0.1185
X9   0.8932 -0.1347 -0.2736 0.8908 0.1092
X10  0.7673 -0.2772 -0.3480 0.7867 0.2133
X11  0.3405 -0.4065 -0.1800 0.3135 0.6865
X12 -0.3917  0.2483  0.1813 0.2480 0.7520
X13  0.2587 -0.7752 -0.4072 0.8337 0.1663
X14 -0.3043  0.8520  0.2111 0.8630 0.1370
```

Boston Housing

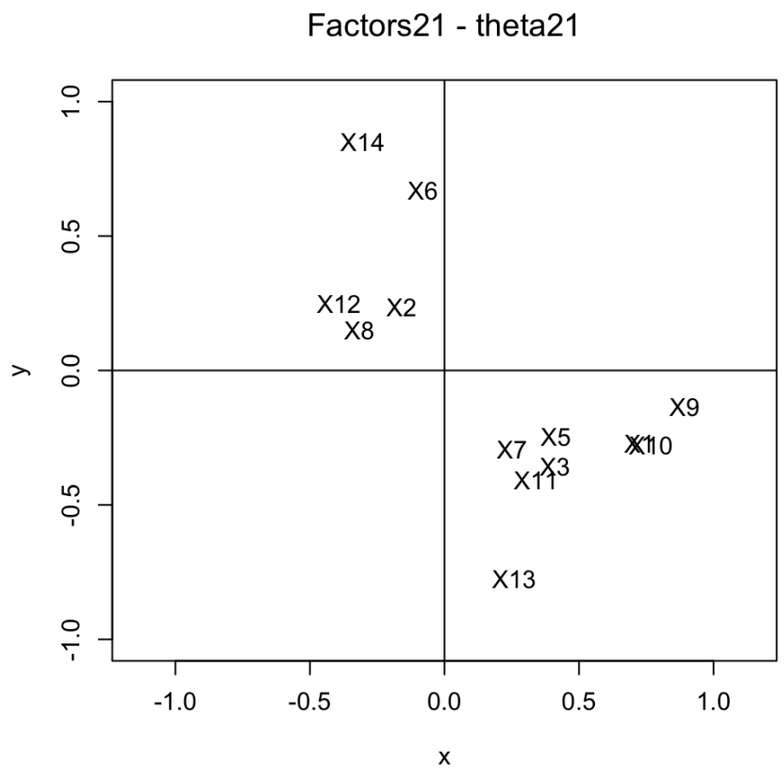
- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.

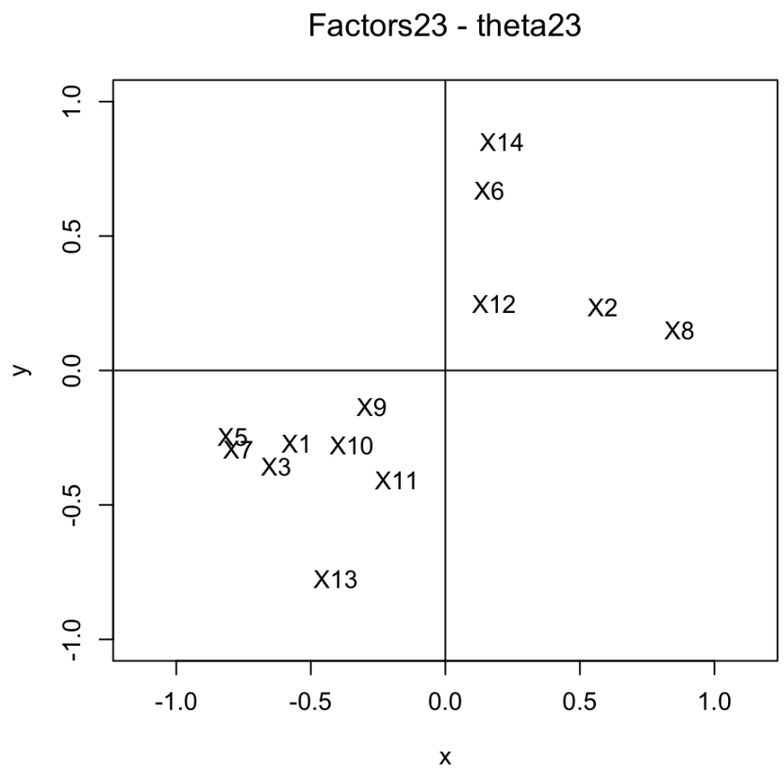
```
graphics.off()
par(mfcol = c(2, 2))
# plot first factor against second 旋转后变量在第一、第二公共因子的散点图
plot(load[, 1], load[, 2], type = "n", xlab = "x", ylab = "y", main = "Factors21 - theta21",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1, 1), ylim = c(-1, 1), asp = 1)
text(load[, 1], load[, 2], colnames(data), cex = 1.1)
abline(h = 0, v = 0)

# plot first factor against third 旋转后变量在第一、第三公共因子的散点图
plot(load[, 1], load[, 3], type = "n", xlab = "x", ylab = "y", main = "Factors31 - theta31",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1, 1), ylim = c(-1, 1), asp = 1)
text(load[, 1], load[, 3], colnames(data), cex = 1.1)
abline(h = 0, v = 0)

# plot second factor against third 旋转后变量在第二、第三公共因子的散点图
plot(load[, 3], load[, 2], type = "n", xlab = "x", ylab = "y", main = "Factors23 - theta23",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1, 1), ylim = c(-1, 1), asp = 1)
text(load[, 3], load[, 2], colnames(data), cex = 1.1)
abline(h = 0, v = 0)
```

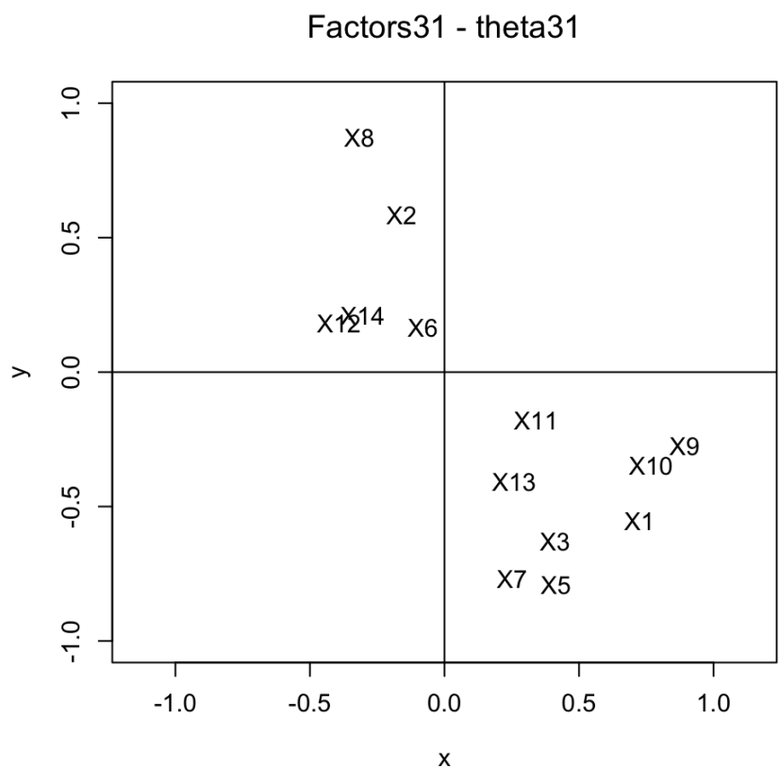


Boston Housing

我们可以看到，方差最大化旋转并没有显著改变通过极大似然法 (MLM) 得到的因子的解释.





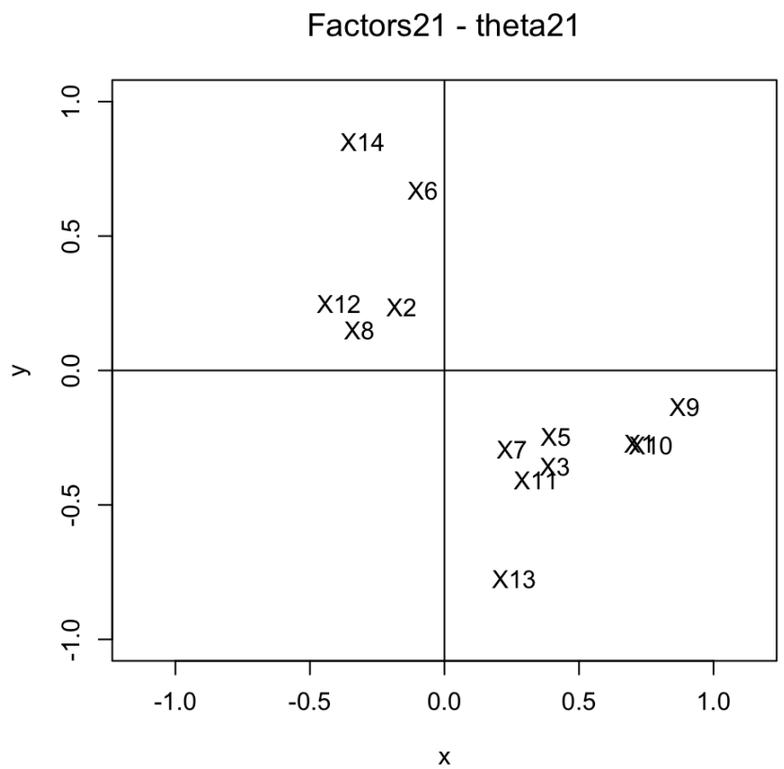


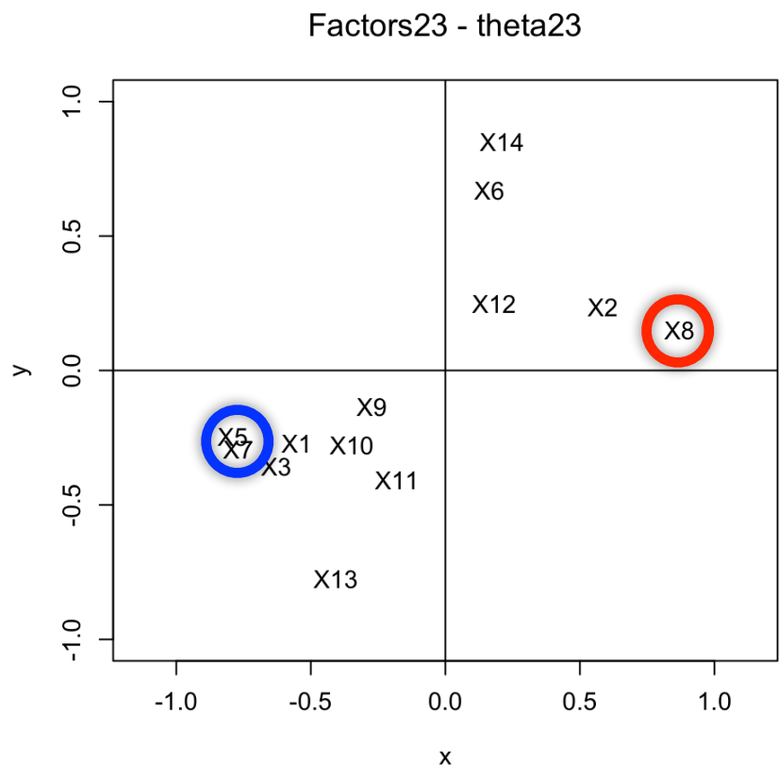

```
> round(tbl, digits = 4)
      RL_F1   RL_F2   RL_F3    com    psi
X1   0.7247 -0.2705 -0.5525 0.9036 0.0964
X2  -0.1587  0.2377  0.5858 0.4248 0.5752
X3   0.4105 -0.3566 -0.6287 0.6909 0.3091
X5   0.4141 -0.2468 -0.7898 0.8561 0.1439
X6  -0.0799  0.6691  0.1644 0.4812 0.5188
X7   0.2518 -0.2934 -0.7688 0.7406 0.2594
X8  -0.3164  0.1515  0.8709 0.8815 0.1185
X9   0.8932 -0.1347 -0.2736 0.8908 0.1092
X10  0.7673 -0.2772 -0.3480 0.7867 0.2133
X11  0.3405 -0.4065 -0.1800 0.3135 0.6865
X12 -0.3917  0.2483  0.1813 0.2480 0.7520
X13  0.2587 -0.7752 -0.4072 0.8337 0.1663
X14 -0.3043  0.8520  0.2111 0.8630 0.1370
```

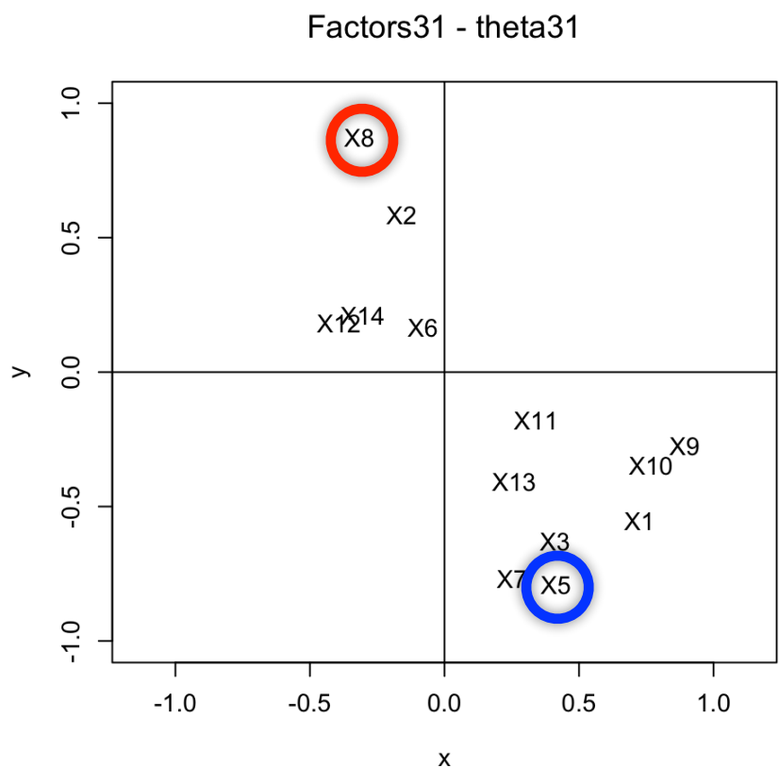
lysis

我们可以将因子 3 解释为就业因子，因为我们观察到它与 X_8 和 X_5 有高度相关性.

在极大似然法 (MLM) 经方差最大化旋转的版本中，各变量有了明显的区分.








```
> round(tbl, digits = 4)
      RL_F1   RL_F2   RL_F3    com    psi
X1   0.7247 -0.2705 -0.5525 0.9036 0.0964
X2  -0.1587  0.2377  0.5858 0.4248 0.5752
X3   0.4105 -0.3566 -0.6287 0.6909 0.3091
X5   0.4141 -0.2468 -0.7898 0.8561 0.1439
X6  -0.0799  0.6691  0.1644 0.4812 0.5188
X7   0.2518 -0.2934 -0.7688 0.7406 0.2594
X8  -0.3164  0.1515  0.8709 0.8815 0.1185
X9   0.8932 -0.1347 -0.2736 0.8908 0.1092
X10  0.7673 -0.2772 -0.3480 0.7867 0.2133
X11  0.3405 -0.4065 -0.1800 0.3135 0.6865
X12 -0.3917  0.2483  0.1813 0.2480 0.7520
X13  0.2587 -0.7752 -0.4072 0.8337 0.1663
X14 -0.3043  0.8520  0.2111 0.8630 0.1370
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转谱分解后的主成分法.

```
# 相关矩阵的谱分解
e = eigen(dat)

# 特征值
eigval.all = e$values # 全部特征值
round(eigval.all, digits = 4)
eigval = e$values[1:3] # 前三个最大特征值
round(eigval, digits = 4)
```

```
> round(eigval.all, digits = 4)
 [1] 7.2852 1.3517 1.1266 0.7802 0.6359 0.5290 0.3397 0.2628 0.1936 0.1547 0.1405 0.1100 0.0900
```

```
> round(eigval, digits = 4)
[1] 7.2852 1.3517 1.1266
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转谱分解后的主成分法.

```
# 特征向量
eigvec.all = e$vectors # 全部特征向量
round(eigvec.all, digits = 4)
eigvec = e$vectors[, 1:3] # 前三个特征向量
round(eigvec, digits = 4)
```

```
> round(eigvec.all, digits = 4)
        [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]    [,8]    [,9]   [,10]   [,11]   [,12]   [,13]
 [1,]  0.3363  0.1933  0.1373  0.0459  0.1084 -0.0882 -0.0861  0.2047  0.0154 -0.2098  0.3262  0.2388  0.7430
 [2,] -0.2371 -0.0251  0.4765 -0.0981  0.4405  0.5190  0.3205 -0.1946  0.0851 -0.2847  0.0774  0.0877 -0.0244
 [3,]  0.3179  0.0352 -0.1738  0.0753 -0.0466 -0.2477  0.1635 -0.7175  0.4110 -0.2527  0.1132  0.0663 -0.0824
 [4,]  0.3237  0.2057 -0.1677 -0.1356  0.1037  0.1222  0.1289 -0.0560 -0.5291 -0.1539  0.3923 -0.5251 -0.1669
 [5,] -0.1891  0.6052  0.0819  0.0961 -0.2884  0.3207 -0.5312 -0.2899 -0.0475 -0.1109 -0.1150 -0.0290  0.0350
 [6,]  0.2964  0.1338 -0.2778 -0.1141  0.0235  0.5206 -0.0298  0.2278  0.5005  0.3668  0.2521  0.0748 -0.1599
 [7,] -0.3060 -0.2498  0.2810  0.1206 -0.0558 -0.1563 -0.2993 -0.0452  0.2982  0.1296  0.5670 -0.4502  0.0376
 [8,]  0.2790  0.2457  0.3584  0.2227  0.2489 -0.2888 -0.2165  0.3307  0.1270 -0.2297 -0.0102  0.0967 -0.5511
 [9,]  0.3006  0.1415  0.3459  0.1959  0.2685 -0.0474  0.0666 -0.2473 -0.0411  0.6317 -0.3017 -0.2612  0.1778
[10,]  0.2102 -0.2294  0.1411  0.6836 -0.4683  0.3196  0.2405  0.1023 -0.0565 -0.1361 -0.0158 -0.0646 -0.0073
[11,] -0.1818 -0.0895 -0.4870  0.5618  0.5757  0.0831 -0.2277 -0.0606 -0.0827 -0.0349  0.0168  0.0374  0.0495
[12,]  0.2963 -0.3658 -0.0236 -0.2109  0.0963  0.1796 -0.3610  0.0659  0.2089 -0.3542 -0.4397 -0.4157  0.1481
[13,] -0.2729  0.4438 -0.1646  0.1096  0.0181 -0.1465  0.4272  0.2746  0.3613 -0.1633 -0.1939 -0.4360  0.1573
```

```
> round(eigvec, digits = 4)
        [,1]    [,2]    [,3]
 [1,]  0.3363  0.1933  0.1373
 [2,] -0.2371 -0.0251  0.4765
 [3,]  0.3179  0.0352 -0.1738
 [4,]  0.3237  0.2057 -0.1677
 [5,] -0.1891  0.6052  0.0819
 [6,]  0.2964  0.1338 -0.2778
 [7,] -0.3060 -0.2498  0.2810
 [8,]  0.2790  0.2457  0.3584
 [9,]  0.3006  0.1415  0.3459
[10,]  0.2102 -0.2294  0.1411
[11,] -0.1818 -0.0895 -0.4870
[12,]  0.2963 -0.3658 -0.0236
[13,] -0.2729  0.4438 -0.1646
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转谱分解后的主成分法.

```
# the estimated factor loadings matrix 计算因子载荷矩阵的估计值
Q = eigvec %*% sqrt(diag(eigval))
round(Q, digits = 4)
```

$$\widehat{\mathcal{Q}} = \left(\sqrt{\lambda_1}\boldsymbol{\gamma}_1, \sqrt{\lambda_2}\boldsymbol{\gamma}_2, \cdots, \sqrt{\lambda_k}\boldsymbol{\gamma}_k\right)$$

```
> round(Q, digits = 4)
        [,1]    [,2]    [,3]
 [1,]  0.9076  0.2247  0.1457
 [2,] -0.6399 -0.0292  0.5058
 [3,]  0.8580  0.0409 -0.1845
 [4,]  0.8737  0.2391 -0.1780
 [5,] -0.5104  0.7037  0.0869
 [6,]  0.7999  0.1556 -0.2949
 [7,] -0.8259 -0.2904  0.2982
 [8,]  0.7531  0.2857  0.3804
 [9,]  0.8114  0.1645  0.3672
[10,]  0.5674 -0.2667  0.1498
[11,] -0.4906 -0.1041 -0.5170
[12,]  0.7996 -0.4253 -0.0251
[13,] -0.7366  0.5160 -0.1747
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转谱分解后的主成分法.

```
# rotates the factor loadings matrix 因子载荷矩阵的旋转
pcm = varimax(Q)
load = pcm$loadings # estimated factor loadings after varimax 旋转后的因子载荷
ld   = cbind(load[, 1], load[, 2], load[, 3]) # 旋转后的因子载荷矩阵
round(ld, digits = 4)
```

```
> round(ld, digits = 4)
        [,1]    [,2]    [,3]
 [1,]  0.6034 -0.2456  0.6864
 [2,] -0.7722  0.2631  0.0247
 [3,]  0.7183 -0.3701  0.3449
 [4,]  0.7936 -0.2043  0.4250
 [5,] -0.1601  0.8585  0.0218
 [6,]  0.7895 -0.2375  0.2670
 [7,] -0.8562  0.1318 -0.3240
 [8,]  0.3681 -0.1268  0.8012
 [9,]  0.3744 -0.2604  0.7825
[10,]  0.1982 -0.5124  0.3372
[11,] -0.0382  0.1647 -0.7002
[12,]  0.4141 -0.7564  0.2781
[13,] -0.2111  0.8131 -0.3657
```

```
rm = pcm$rotmat # 旋转矩阵
round(rm, digits = 4)
```

```
> round(rm, digits = 4)
        [,1]    [,2]   [,3]
[1,]  0.6823 -0.4813 0.5503
[2,]  0.3469  0.8757 0.3358
[3,] -0.6435 -0.0382 0.7645
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转谱分解后的主成分法.

```
# communalities are calculated 计算公共因子方差
com = diag(ld %*% t(ld))
names(com) = row.names(dat)
round(com, digits = 4)
```

$$\hat{h}_j^2 = \sum_{\ell=1}^{k} \hat{q}_{j\ell}^2$$

```
> round(com, digits = 4)
    X1     X2     X3     X5     X6     X7     X8     X9    X10    X11    X12    X13    X14
0.8955 0.6661 0.7719 0.8521 0.7632 0.7510 0.8554 0.7935 0.8203 0.4155 0.5188 0.8209 0.8394
```

```
# specific variances are calculated 计算特殊因子方差
psi = diag(dat) - diag(ld %*% t(ld))
round(psi, digits = 4)
```

$$\widehat{\psi}_{jj} = s_{X_j X_j} - \sum_{\ell=1}^{k} \hat{q}_{j\ell}^2$$

```
> round(psi, digits = 4)
    X1     X2     X3     X5     X6     X7     X8     X9    X10    X11    X12    X13    X14
0.1045 0.3339 0.2281 0.1479 0.2368 0.2490 0.1446 0.2065 0.1797 0.5845 0.4812 0.1791 0.1606
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转谱分解后的主成分法.

```
# 旋转后的因子载荷、公共因子方差、特殊因子方差的矩阵
tbl = cbind(LF1 = load[, 1], LF2 = load[, 2], LF3 = load[, 3], com, psi)
round(tbl, digits = 4)
```

$$\mathcal{S} = \widehat{\mathcal{Q}}\,\widehat{\mathcal{Q}}^{\mathrm{T}} + \widehat{\Psi}$$

```
> round(tbl, digits = 4)
        LF1     LF2     LF3    com    psi
X1   0.6034 -0.2456  0.6864 0.8955 0.1045
X2  -0.7722  0.2631  0.0247 0.6661 0.3339
X3   0.7183 -0.3701  0.3449 0.7719 0.2281
X5   0.7936 -0.2043  0.4250 0.8521 0.1479
X6  -0.1601  0.8585  0.0218 0.7632 0.2368
X7   0.7895 -0.2375  0.2670 0.7510 0.2490
X8  -0.8562  0.1318 -0.3240 0.8554 0.1446
X9   0.3681 -0.1268  0.8012 0.7935 0.2065
X10  0.3744 -0.2604  0.7825 0.8203 0.1797
X11  0.1982 -0.5124  0.3372 0.4155 0.5845
X12 -0.0382  0.1647 -0.7002 0.5188 0.4812
X13  0.4141 -0.7564  0.2781 0.8209 0.1791
X14 -0.2111  0.8131 -0.3657 0.8394 0.1606
```

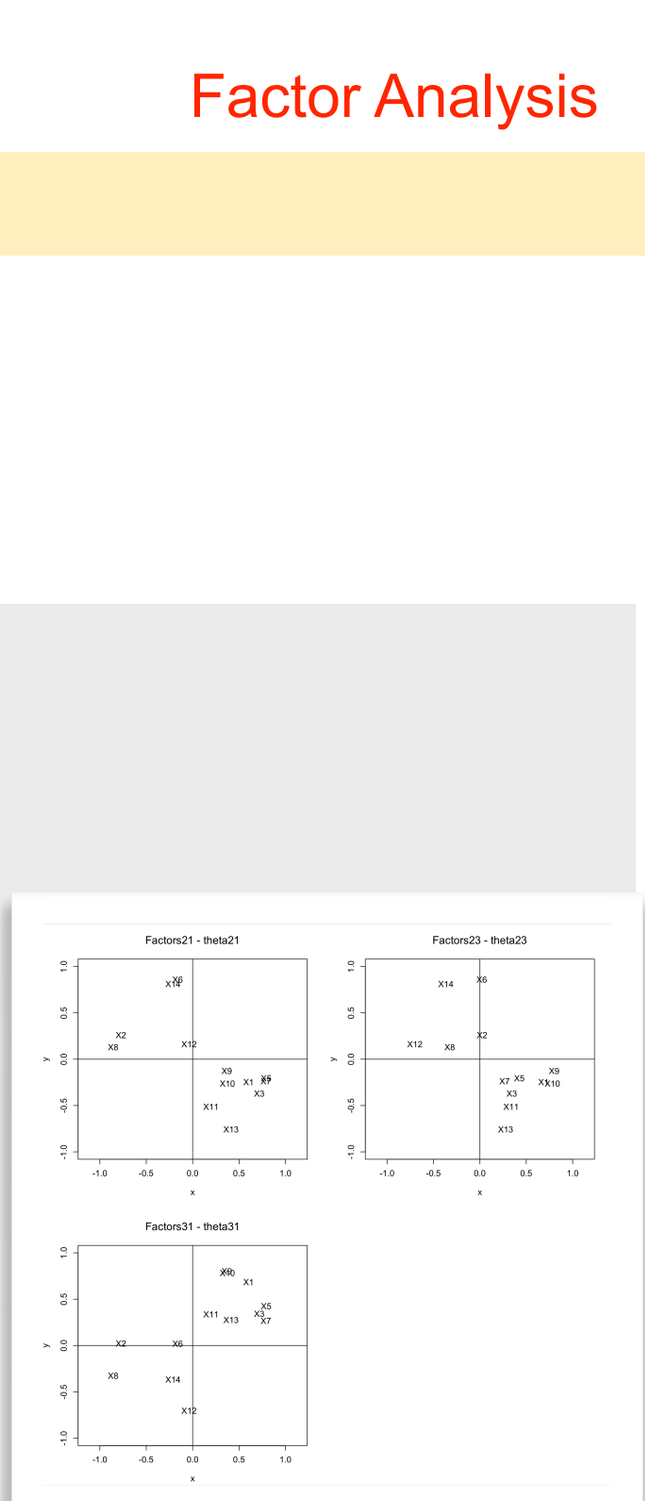
$\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$

$\widehat{\boldsymbol{q}}_1$ $\widehat{\boldsymbol{q}}_2$ $\widehat{\boldsymbol{q}}_3$ $\widehat{h}_j^2$ $\widehat{\psi}_{jj}$

Boston Housing

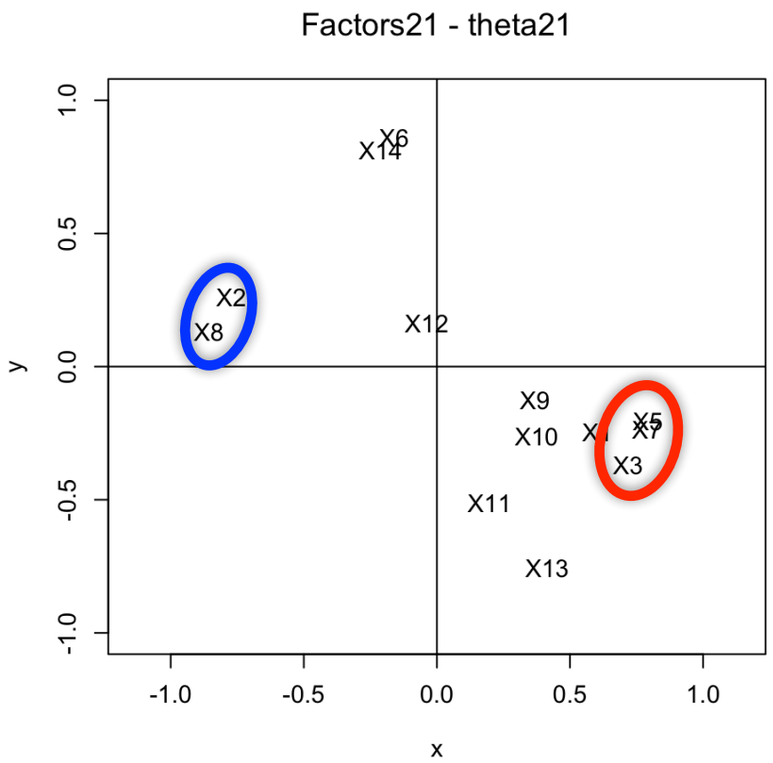
- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转谱分解后的主成分法.

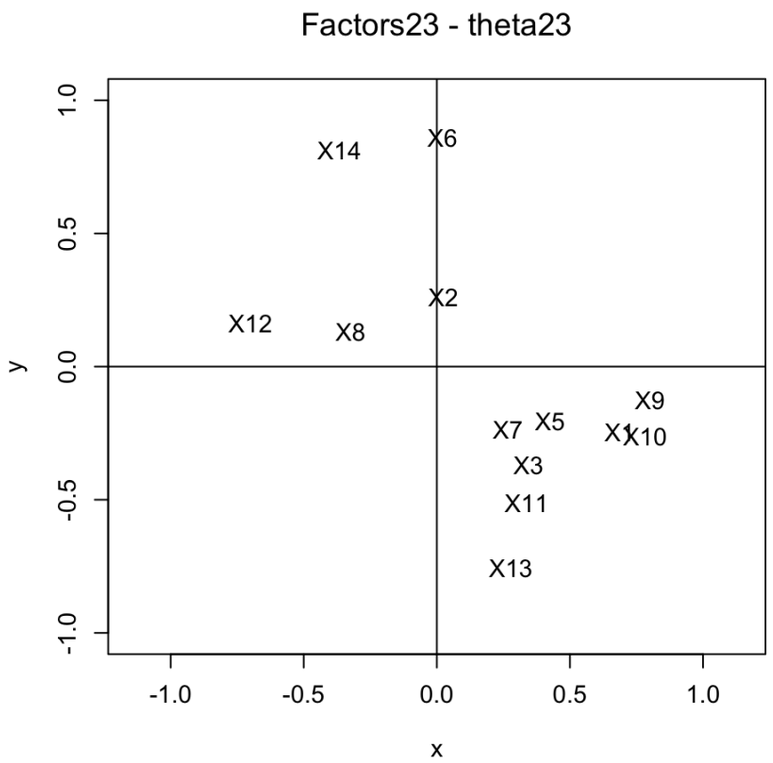
```
graphics.off()
par(mfcol = c(2, 2))
# plot first factor against second 旋转后变量在第一、第二公共因子的散点图
plot(load[, 1], load[, 2], type = "n", xlab = "x", ylab = "y", main = "Factors21 - theta21", xlim = c(-1, 1),
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, ylim = c(-1, 1), asp = 1)
text(load[, 1], load[, 2], colnames(data), cex = 1.1)
abline(h = 0, v = 0)
# plot first factor against third 旋转后变量在第一、第三公共因子的散点图
plot(load[, 1], load[, 3], type = "n", xlab = "x", ylab = "y", main = "Factors31 - theta31",  xlim = c(-1, 1),
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, ylim = c(-1, 1), asp = 1)
text(load[, 1], load[, 3], colnames(data), cex = 1.1)
abline(h = 0, v = 0)
# plot second factor against third 旋转后变量在第二、第三公共因子的散点图
plot(load[, 3], load[, 2], type = "n", xlab = "x", ylab = "y", main = "Factors23 - theta23",  xlim = c(-1, 1),
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, ylim = c(-1, 1), asp = 1)
text(load[, 3], load[, 2], colnames(data), cex = 1.1)
abline(h = 0, v = 0)
```

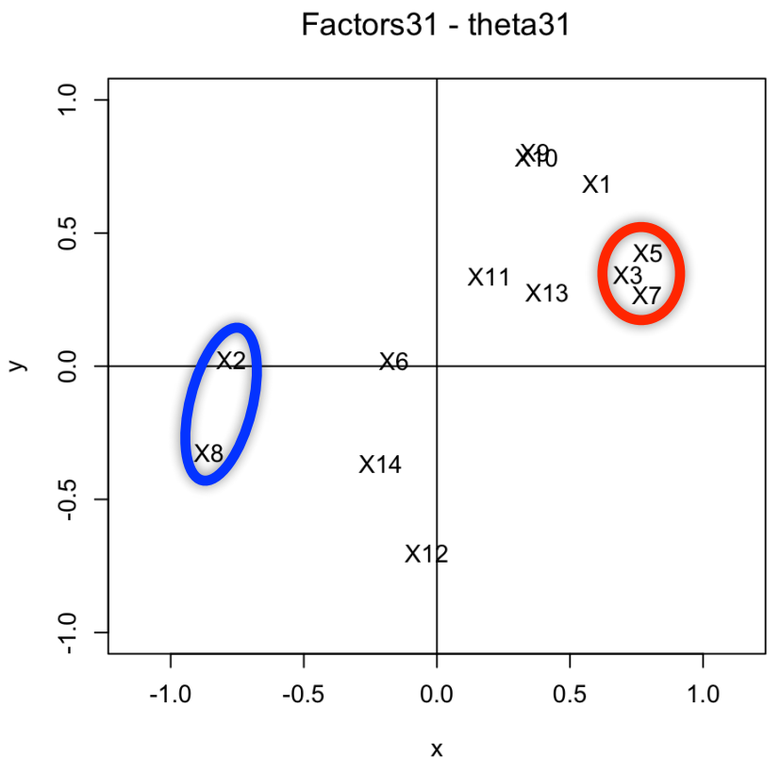


Boston Housing

因子 1 仍然是一个“生活质量因子”，它与诸如 X_5, X_3, X_7 等变量呈正相关，与 X_2, X_8 呈负相关.

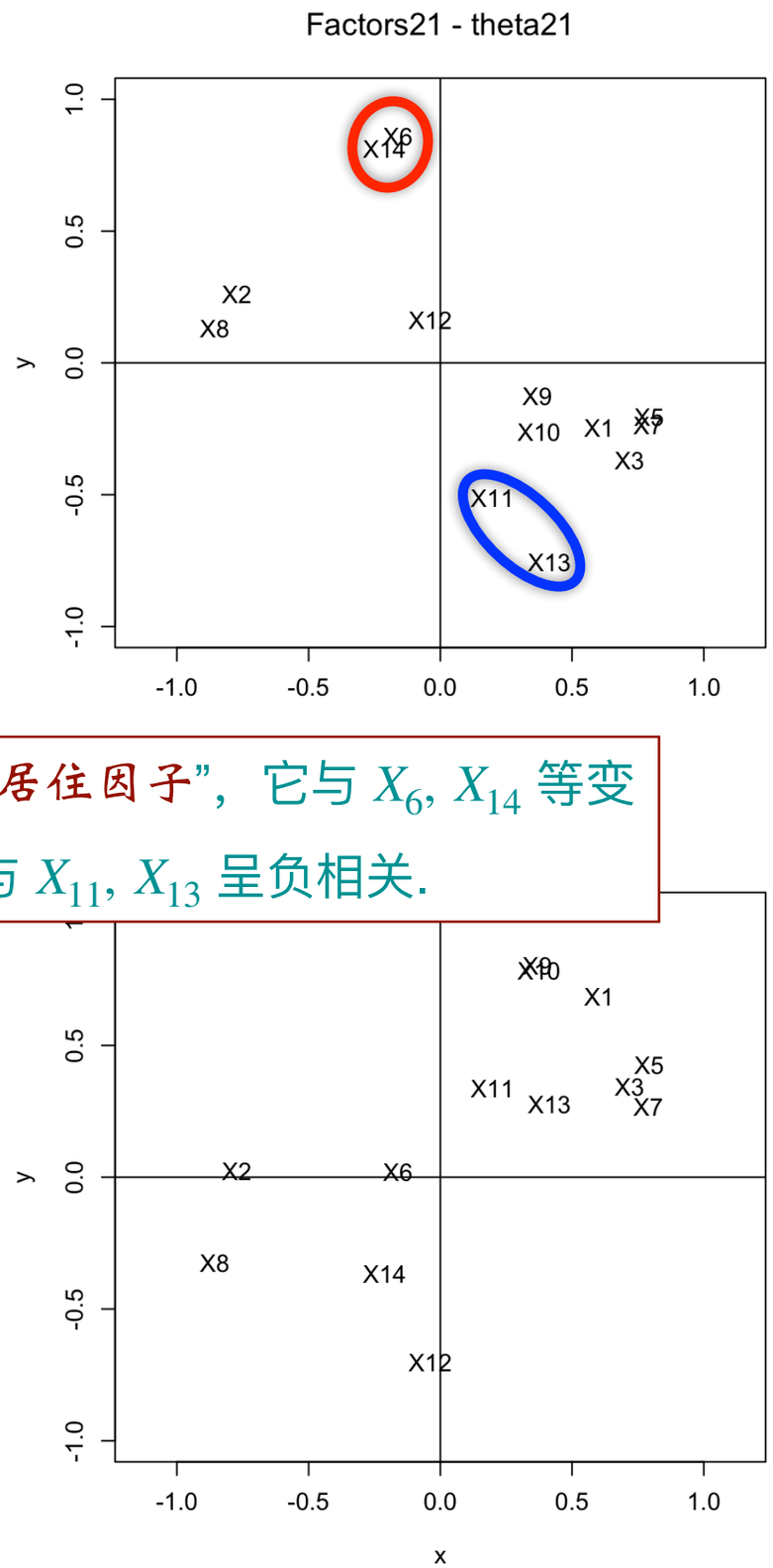


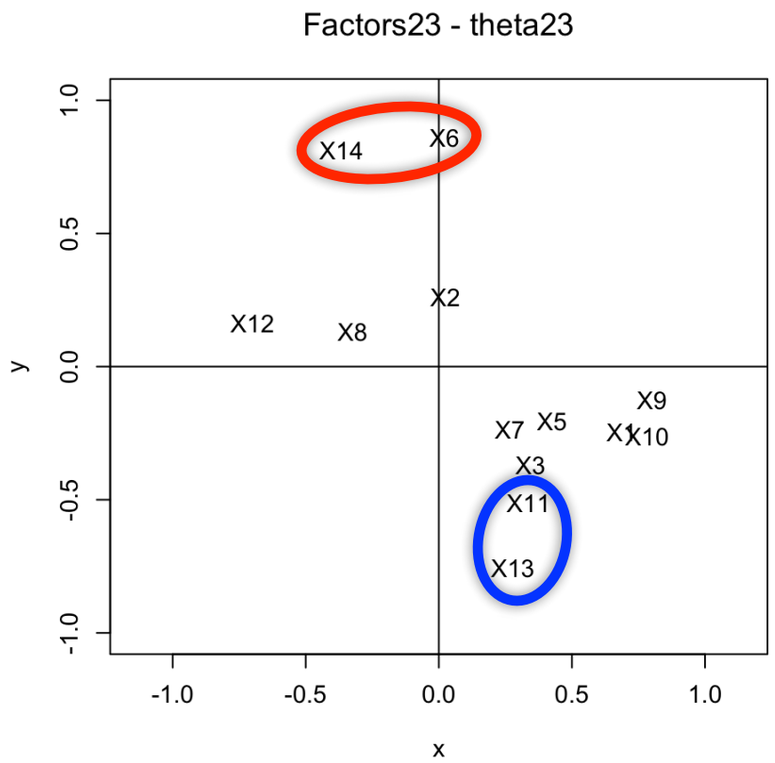





```
> round(tbl, digits = 4)
        LF1     LF2     LF3    com    psi
X1   0.6034 -0.2456  0.6864 0.8955 0.1045
X2  -0.7722  0.2631  0.0247 0.6661 0.3339
X3   0.7183 -0.3701  0.3449 0.7719 0.2281
X5   0.7936 -0.2043  0.4250 0.8521 0.1479
X6  -0.1601  0.8585  0.0218 0.7632 0.2368
X7   0.7895 -0.2375  0.2670 0.7510 0.2490
X8  -0.8562  0.1318 -0.3240 0.8554 0.1446
X9   0.3681 -0.1268  0.8012 0.7935 0.2065
X10  0.3744 -0.2604  0.7825 0.8203 0.1797
X11  0.1982 -0.5124  0.3372 0.4155 0.5845
X12 -0.0382  0.1647 -0.7002 0.5188 0.4812
X13  0.4141 -0.7564  0.2781 0.8209 0.1791
X14 -0.2111  0.8131 -0.3657 0.8394 0.1606
```

Boston Housing



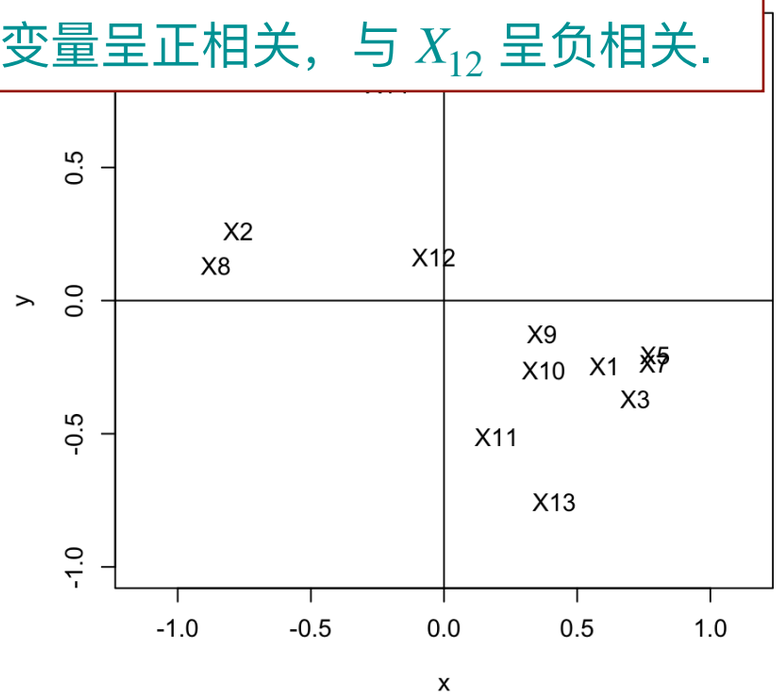


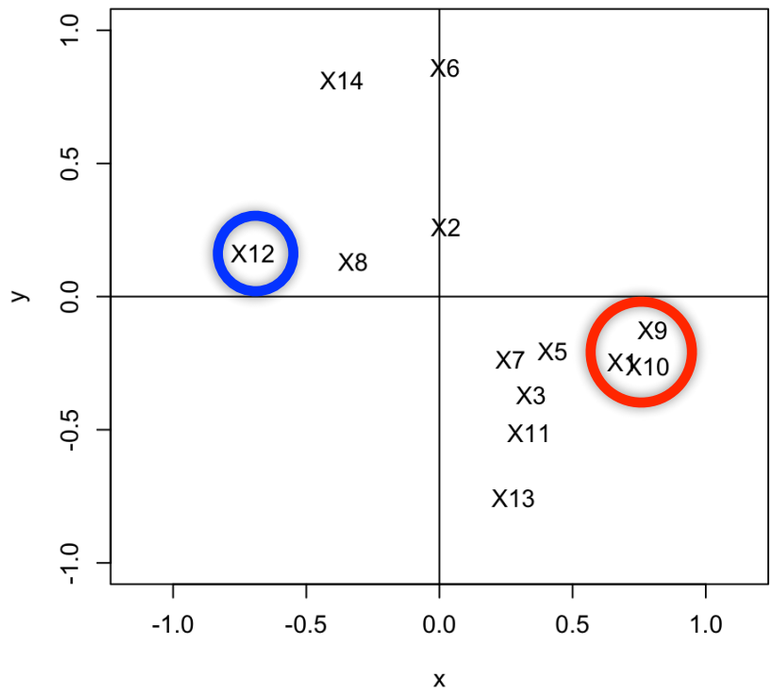
因子 2 仍然是“居住因子”，它与 X_6, X_{14} 等变量呈正相关，与 X_{11}, X_{13} 呈负相关.

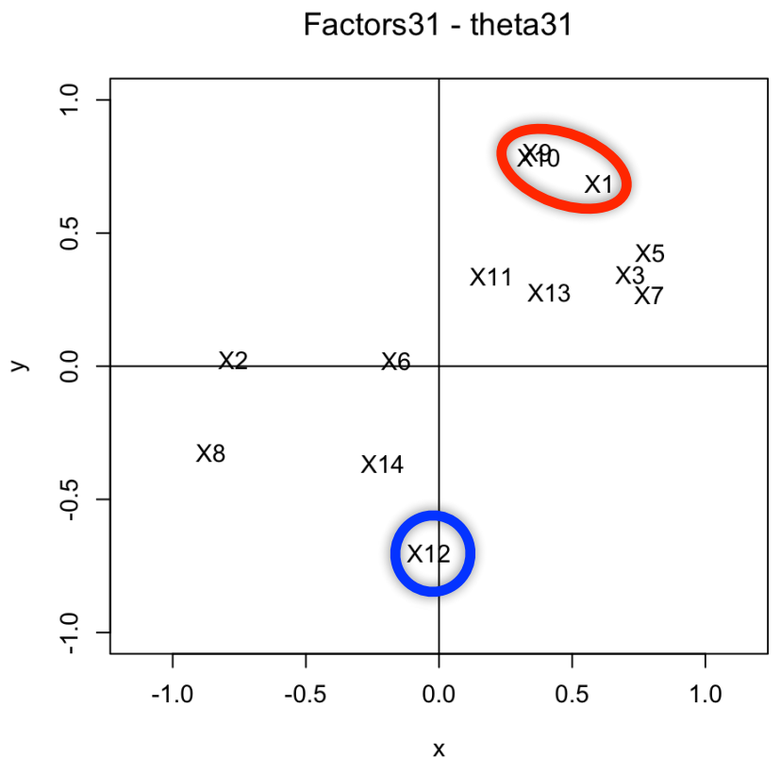

```
> round(tbl, digits = 4)
        LF1     LF2     LF3    com    psi
X1   0.6034 -0.2456  0.6864 0.8955 0.1045
X2  -0.7722  0.2631  0.0247 0.6661 0.3339
X3   0.7183 -0.3701  0.3449 0.7719 0.2281
X5   0.7936 -0.2043  0.4250 0.8521 0.1479
X6  -0.1601  0.8585  0.0218 0.7632 0.2368
X7   0.7895 -0.2375  0.2670 0.7510 0.2490
X8  -0.8562  0.1318 -0.3240 0.8554 0.1446
X9   0.3681 -0.1268  0.8012 0.7935 0.2065
X10  0.3744 -0.2604  0.7825 0.8203 0.1797
X11  0.1982 -0.5124  0.3372 0.4155 0.5845
X12 -0.0382  0.1647 -0.7002 0.5188 0.4812
X13  0.4141 -0.7564  0.2781 0.8209 0.1791
X14 -0.2111  0.8131 -0.3657 0.8394 0.1606
```

Boston Housing

因子 3 更难解释，因为所有的载荷都较小. 它与诸如 X_9, X_{10}, X_1 等变量呈正相关，与 X_{12} 呈负相关.








```
> round(tbl, digits = 4)
       LF1     LF2     LF3    com    psi
X1  0.6034 -0.2456  0.6864 0.8955 0.1045
X2 -0.7722  0.2631  0.0247 0.6661 0.3339
X3  0.7183 -0.3701  0.3449 0.7719 0.2281
X5  0.7936 -0.2043  0.4250 0.8521 0.1479
X6 -0.1601  0.8585  0.0218 0.7632 0.2368
X7  0.7895 -0.2375  0.2670 0.7510 0.2490
X8 -0.8562  0.1318 -0.3240 0.8554 0.1446
X9  0.3681 -0.1268  0.8012 0.7935 0.2065
X10 0.3744 -0.2604  0.7825 0.8203 0.1797
X11 0.1982 -0.5124  0.3372 0.4155 0.5845
X12 -0.0382 0.1647 -0.7002 0.5188 0.4812
X13 0.4141 -0.7564  0.2781 0.8209 0.1791
X14 -0.2111 0.8131 -0.3657 0.8394 0.1606
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转后基于相关矩阵逆的主因子法.

```
#### Principal Factor Method after varimax rotation inverse of the correlation matrix
#### 主因子方法的因子分析，旋转后，从相关矩阵的逆矩阵出发
f = solve(dat) # 相关矩阵的逆矩阵
# preliminary estimate of psi - (psi 的) 初始值
psiini = diag(1/f[row(f) == col(f)])
psi = psiini
round(psi, digits = 4)
```

```
> round(psi, digits = 4)
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]  [,11]  [,12]  [,13]
 [1,] 0.1254 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
 [2,] 0.0000 0.4623 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
 [3,] 0.0000 0.0000 0.2704 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
 [4,] 0.0000 0.0000 0.0000 0.1751 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
 [5,] 0.0000 0.0000 0.0000 0.0000 0.5061 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
 [6,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.2649 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
 [7,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.1929 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
 [8,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.2029 0.0000 0.0000 0.0000 0.0000 0.0000
 [9,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.2103 0.0000 0.0000 0.0000 0.0000
[10,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.6305 0.0000 0.0000 0.0000
[11,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.7084 0.0000 0.0000
[12,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.2053 0.0000
[13,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.2393
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转后基于相关矩阵逆的主因子法.

```
# 迭代求解因子载荷矩阵
for (i in 1:10) {
  ee = eigen(dat - psi) # ℛ − Ψ̃ 的谱分解
  eigval = ee$values[1:3] # 取前三个特征值
  eigvec = ee$vectors[, 1:3] # 取前三个特征向量
  EE = matrix(eigval, nrow(dat), ncol = 3, byrow = T)
  QQ = sqrt(EE) * eigvec # 计算载荷矩阵
  psiold = psi
  psi = diag(as.vector(1 - t(colSums(t(QQ * QQ))))) # 利用载荷矩阵计算 psi 的估计值
  i = i + 1
  z = psi - psiold
  convergence = z[row(z) == col(z)]
}
round(QQ, digits = 4) # 未旋转的因子载荷矩阵
```

```
> round(QQ, digits = 4)
        [,1]    [,2]    [,3]
 [1,]  0.9128  0.2159 -0.1767
 [2,] -0.6028 -0.0259 -0.3185
 [3,]  0.8397  0.0342  0.1427
 [4,]  0.8708  0.2184  0.1946
 [5,] -0.4824  0.5088 -0.0590
 [6,]  0.7816  0.1191  0.2824
 [7,] -0.8262 -0.2849 -0.3272
 [8,]  0.7479  0.2642 -0.4318
 [9,]  0.8074  0.1486 -0.3980
[10,]  0.5221 -0.1601 -0.1038
[11,] -0.4484 -0.0230  0.2043
[12,]  0.7940 -0.4457  0.0638
[13,] -0.7336  0.5522  0.1366
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转后基于相关矩阵逆的主因子法.

```
# rotates the factor loadings matrix 因子载荷矩阵的旋转
pfm  = varimax(QQ)
# estimated factor loadings after varimax 旋转后的因子载荷矩阵
load = pfm$loadings
ld   = cbind(load[, 1], load[, 2], load[, 3])
round(ld, digits = 4)
```

```
> round(ld, digits = 4)
        [,1]    [,2]    [,3]
 [1,]  0.5477 -0.2558 -0.7387
 [2,] -0.6148  0.2668  0.1281
 [3,]  0.6523 -0.3761 -0.3996
 [4,]  0.7723 -0.2296 -0.4412
 [5,] -0.1732  0.6783  0.0699
 [6,]  0.7390 -0.2723 -0.2909
 [7,] -0.8565  0.1485  0.3395
 [8,]  0.2855 -0.1359 -0.8460
 [9,]  0.3062 -0.2656 -0.8174
[10,]  0.2116 -0.3943 -0.3297
[11,] -0.1604  0.1994  0.4217
[12,]  0.4005 -0.7743 -0.2706
[13,] -0.1885  0.8400  0.3473
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转后基于相关矩阵逆的主因子法.

```
# communalities are calculated 计算公共因子方差
com  = diag(ld %*% t(ld))
round(com, digits = 4)
```

```
> round(com, digits = 4)
 [1] 0.9111 0.4655 0.7266 0.8438 0.4950 0.7048 0.8709 0.8157 0.8324 0.3089 0.2433 0.8331 0.8617
```

```
# specific variances are calculated 计算特殊因子方差
psi  = diag(dat) - diag(ld %*% t(ld))
round(psi, digits = 4)
```

```
> round(psi, digits = 4)
    X1     X2     X3     X5     X6     X7     X8     X9    X10    X11    X12    X13    X14
0.0889 0.5345 0.2734 0.1562 0.5050 0.2952 0.1291 0.1843 0.1676 0.6911 0.7567 0.1669 0.1383
```

Boston Housing

- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转后基于相关矩阵逆的主因子法.

```
# 旋转后的因子载荷、公共因子方差、特殊因子方差的矩阵
tbl  = cbind(L_P1 = load[, 1], L_P2 = load[, 2], L_P3 = load[, 3], com, psi)
round(tbl, digits = 4)
```

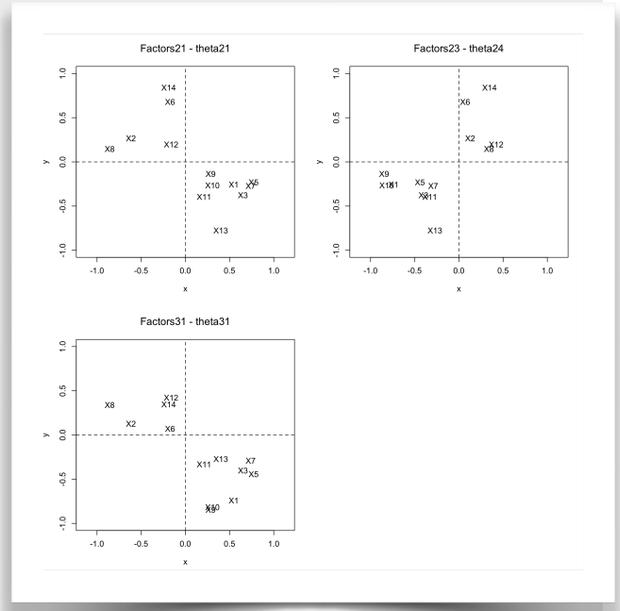
```
> round(tbl, digits = 4)
       L_P1    L_P2    L_P3    com    psi
X1   0.5477 -0.2558 -0.7387 0.9111 0.0889
X2  -0.6148  0.2668  0.1281 0.4655 0.5345
X3   0.6523 -0.3761 -0.3996 0.7266 0.2734
X5   0.7723 -0.2296 -0.4412 0.8438 0.1562
X6  -0.1732  0.6783  0.0699 0.4950 0.5050
X7   0.7390 -0.2723 -0.2909 0.7048 0.2952
X8  -0.8565  0.1485  0.3395 0.8709 0.1291
X9   0.2855 -0.1359 -0.8460 0.8157 0.1843
X10  0.3062 -0.2656 -0.8174 0.8324 0.1676
X11  0.2116 -0.3943 -0.3297 0.3089 0.6911
X12 -0.1604  0.1994  0.4217 0.2433 0.7567
X13  0.4005 -0.7743 -0.2706 0.8331 0.1669
X14 -0.1885  0.8400  0.3473 0.8617 0.1383
```

$\hat{q}_1$ $\hat{q}_2$ $\hat{q}_3$ $\hat{h}_j^2$ $\hat{\psi}_{jj}$

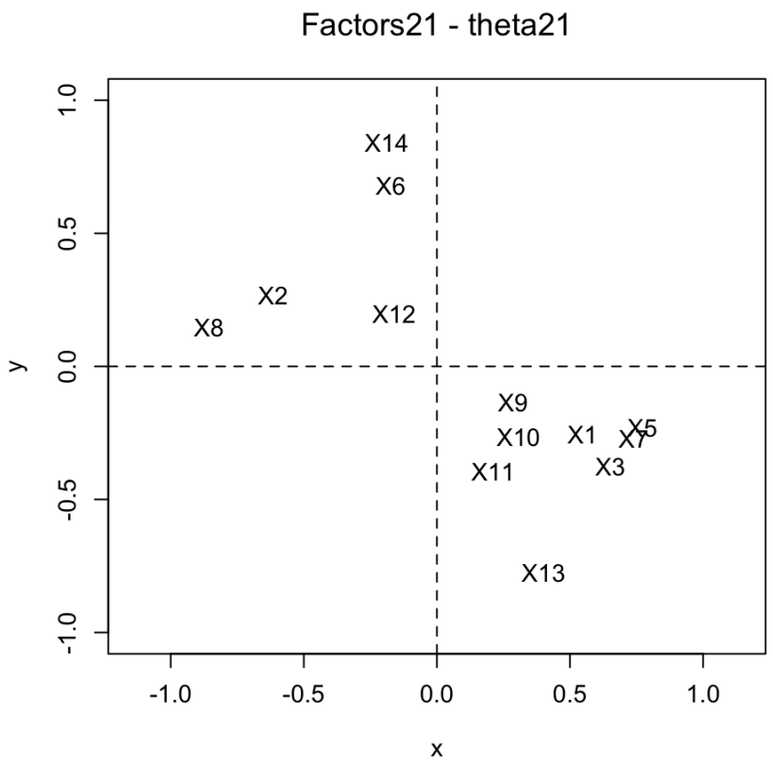
Boston Housing

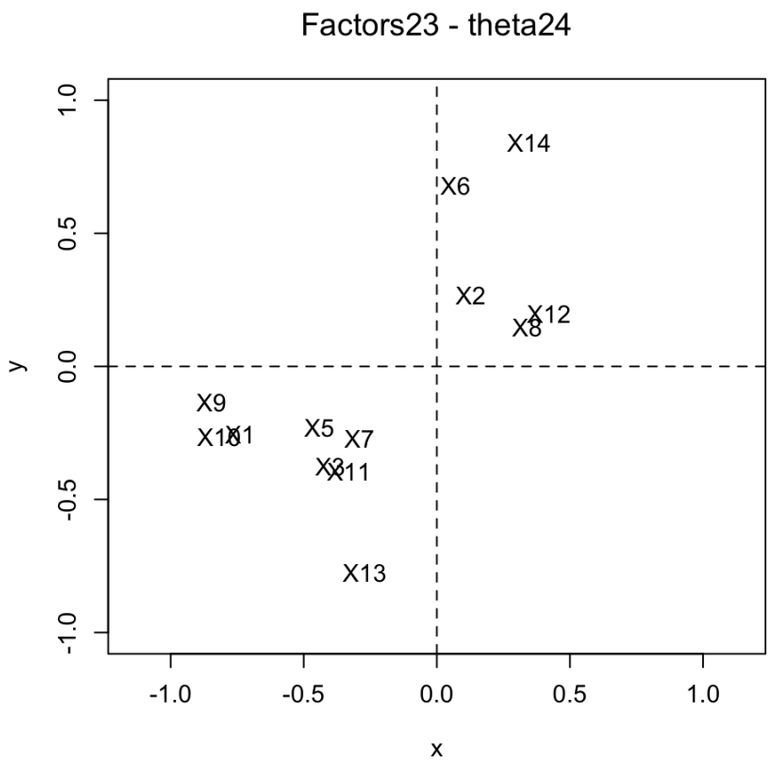
- 例: 我们使用 Boston 房价数据集来说明如何实施因子分析.
  - 方差最大化旋转后基于相关矩阵逆的主因子法.

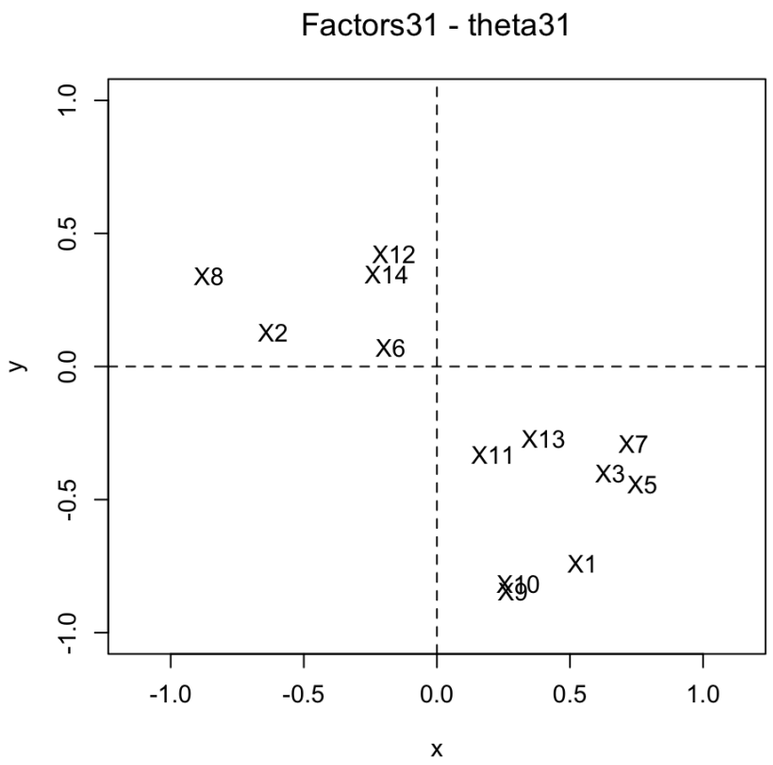
```
graphics.off()
par(mfcol = c(2, 2))
# plot first factor against second 旋转后变量在第一、第二公共因子的散点图
plot(load[, 1], load[, 2], type = "n", xlab = "x", ylab = "y", main = "Factors21 - theta21",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1, 1), ylim = c(-1, 1), asp = 1)
text(load[, 1], load[, 2], colnames(data), cex = 1.1)
abline(h = 0, v = 0, lty = 2)
# plot first factor against third 旋转后变量在第一、第三公共因子的散点图
plot(load[, 1], load[, 3], type = "n", xlab = "x", ylab = "y", main = "Factors31 - theta31",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1, 1), ylim = c(-1, 1), asp = 1)
text(load[, 1], load[, 3], colnames(data), cex = 1.1)
abline(h = 0, v = 0, lty = 2)
# plot second factor against third 旋转后变量在第二、第三公共因子的散点图
plot(load[, 3], load[, 2], type = "n", xlab = "x", ylab = "y", main = "Factors23 - theta24",
    font.main = 1, cex.lab = 1.1, cex.axis = 1.1, cex.main = 1.4, xlim = c(-1, 1), ylim = c(-1, 1), asp = 1)
text(load[, 3], load[, 2], colnames(data), cex = 1.1)
abline(h = 0, v = 0, lty = 2)
```



Boston Housing








```
> round(tbl, digits = 4)
       L_P1    L_P2    L_P3    com    psi
X1   0.5477 -0.2558 -0.7387 0.9111 0.0889
X2  -0.6148  0.2668  0.1281 0.4655 0.5345
X3   0.6523 -0.3761 -0.3996 0.7266 0.2734
X5   0.7723 -0.2296 -0.4412 0.8438 0.1562
X6  -0.1732  0.6783  0.0699 0.4950 0.5050
X7   0.7390 -0.2723 -0.2909 0.7048 0.2952
X8  -0.8565  0.1485  0.3395 0.8709 0.1291
X9   0.2855 -0.1359 -0.8460 0.8157 0.1843
X10  0.3062 -0.2656 -0.8174 0.8324 0.1676
X11  0.2116 -0.3943 -0.3297 0.3089 0.6911
X12 -0.1604  0.1994  0.4217 0.2433 0.7567
X13  0.4005 -0.7743 -0.2706 0.8331 0.1669
X14 -0.1885  0.8400  0.3473 0.8617 0.1383
```

能力与智力测试 (Ability and Intelligence Tests)

- 例: 对 112 人进行了六项测试，这里给出了计算得出的协方差矩阵.
  - 普通测试: 使用 Cattell 文化公平测试的非言语形式的一般智力测量
  - 画图: 图片补全测试
  - 积木: 积木图案
  - 迷宫: 迷宫游戏
  - 阅读: 阅读理解
  - 词汇: 词汇量

```
# clear variables and close windows
rm(list = ls(all = TRUE))
graphics.off()
options(digits = 2)
?ability.cov
ability.cov
```

```
> ability.cov
$cov
        general picture  blocks   maze reading   vocab
general  24.641   5.991  33.520  6.023  20.755  29.701
picture   5.991   6.700  18.137  1.782   4.936   7.204
blocks   33.520  18.137 149.831 19.424  31.430  50.753
maze      6.023   1.782  19.424 12.711   4.757   9.075
reading  20.755   4.936  31.430  4.757  52.604  66.762
vocab    29.701   7.204  50.753  9.075  66.762 135.292

$center
[1] 0 0 0 0 0 0

$n.obs
[1] 112
```

能力与智力测试 (Ability and Intelligence Tests)

- 例: 对 112 人进行了六项测试，这里给出了计算得出的协方差矩阵.

```
# 利用函数 cov2cor 计算相关矩阵
x = ability.cov$cov # 读入协方差矩阵数据
cor_x = cov2cor(x) # 利用协方差矩阵计算相关矩阵
cor_x
```

```
> cor_x
        general picture blocks maze reading vocab
general    1.00    0.47   0.55 0.34    0.58  0.51
picture    0.47    1.00   0.57 0.19    0.26  0.24
blocks     0.55    0.57   1.00 0.45    0.35  0.36
maze       0.34    0.19   0.45 1.00    0.18  0.22
reading    0.58    0.26   0.35 0.18    1.00  0.79
vocab      0.51    0.24   0.36 0.22    0.79  1.00
```

能力与智力测试 (Ability and Intelligence Tests)

- 例: 对 112 人进行了六项测试，这里给出了计算得出的协方差矩阵.
  - 确定需要提取的公因子的数量.

```
library(psych)
?fa.parallel # fa.parallel 的帮助文件: 数据或相关矩阵的碎石图
fa.parallel(cor_x, n.obs = 112, fa = "both", n.iter = 100, main = "Scree plots with parallel analysis")
```

公因子个数: $k = 2$.

fa.parallel {psych} R Documentation

Scree plots of data or correlation matrix compared to random “parallel" matrices

Description

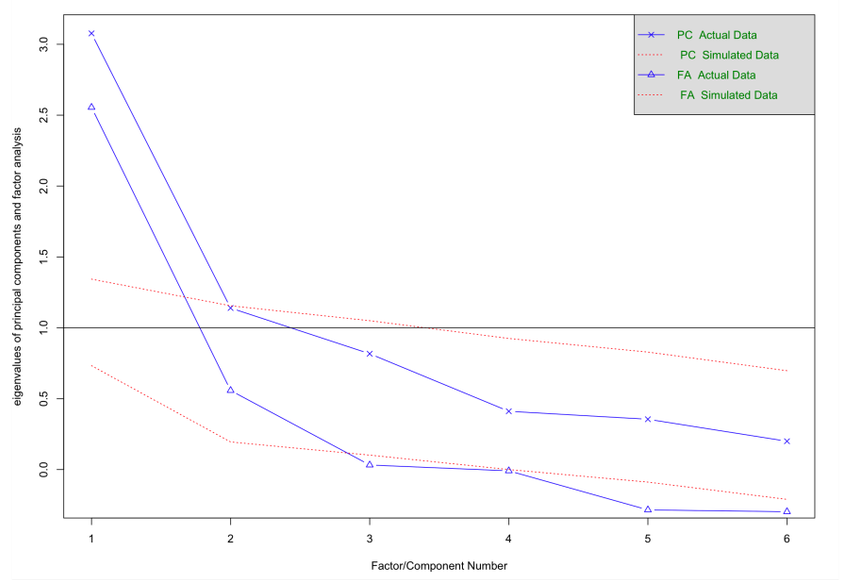
One way to determine the number of factors or components in a data matrix or a correlation matrix is to examine the “scree" plot of the successive eigenvalues. Sharp breaks in the plot suggest the appropriate number of components or factors to extract. “Parallel" analyis is an alternative technique that compares the scree of factors of the observed data with that of a random data matrix of the same size as the original. This may be done for continuous , dichotomous, or polytomous data using Pearson, tetrachoric or polychoric correlations.

Usage

```
fa.parallel(x,n.obs=NULL,fm="minres",fa="both",nfactors=1,
main="Parallel Analysis Scree Plots",
n.iter=20,error.bars=FALSE,se.bars=FALSE,SMC=FALSE,ylabel=NULL,show.legend=TRUE,
sim=TRUE,quant=.95,cor="cor",use="pairwise",plot=TRUE,correct=.5)
fa.parallel.poly(x ,n.iter=10,SMC=TRUE,  fm = "minres",correct=TRUE,sim=FALSE,
       fa="both",global=TRUE)   #deprecated
## S3 method for class 'poly.parallel'
plot(x,show.legend=TRUE,fa="both",...)
```

Arguments

x	A data.frame or data matrix of scores. If the matrix is square, it is assumed to be a correlation matrix. Otherwise, correlations (with pairwise deletion) will be found
n.obs	n.obs=0 implies a data matrix/data.frame. Otherwise, how many cases were used to find the correlations.
fm	What factor method to use. (minres, ml, uls, wls, gls, pa) See fa for details.
fa	show the eigen values for a principal components (fa="pc") or a principal axis factor analysis (fa="fa") or both principal components and principal factors (fa="both")
nfactors	The number of factors to extract when estimating the eigen values. Defaults to 1, which was the prior value used.
main	a title for the analysis
n.iter	Number of simulated analyses to perform



能力与智力测试 (Ability and Intelligence Tests)

- 例: 对 112 人进行了六项测试，这里给出了计算得出的协方差矩阵.
  - 使用 fa( ) 函数提取公因子.

```
?fa # fa 函数的帮助文件：因子分析
```

fa {psych} R Documentation

Exploratory Factor analysis using MinRes (minimum residual) as well as EFA by Principal Axis, Weighted Least Squares or Maximum Likelihood

Description

Among the many ways to do latent variable exploratory factor analysis (EFA), one of the better is to use Ordinary Least Squares (OLS) to find the minimum residual (minres) solution. This produces solutions very similar to maximum likelihood even for badly behaved matrices. A variation on minres is to do weighted least squares (WLS). Perhaps the most conventional technique is principal axes (PAF). An eigen value decomposition of a correlation matrix is done and then the communalities for each variable are estimated by the first n factors. These communalities are entered onto the diagonal and the procedure is repeated until the sum(diag(r)) does not vary. Yet another estimate procedure is maximum likelihood. For well behaved matrices, maximum likelihood factor analysis (either in the fa or in the factanal function) is probably preferred. Bootstrapped confidence intervals of the loadings and interfactor correlations are found by fa with n.iter > 1.

Usage

```
fa(r,nfactors=1,n.obs = NA,n.iter=1, rotate="oblimin", scores="regression",
residuals=FALSE, SMC=TRUE, covar=FALSE,missing=FALSE,impute="median",
min.err = 0.001,  max.iter = 50,symmetric=TRUE, warnings=TRUE, fm="minres",
 alpha=.1,p=.05,oblique.scores=FALSE,np.obs=NULL,use="pairwise",cor="cor",
 correct=.5,weight=NULL,n.rotations=1,hyper=.15,...)
```

能力与智力测试 (Ability and Intelligence Tests)

- 例: 对 112 人进行了六项测试，这里给出了计算得出的协方差矩阵.
  - 使用 fa( ) 函数提取公因子.

拟提取的因子数

```
fa_x = fa(cor_x, nfactors = 2, n.obs = 112, rotate = "none", fm = "pa", scores = "regression" )
fa_x
```

提取因子的方法："minres" 极小残差方法，"wls" 加权最小二乘方法，"gls" 广义加权最小二乘方法，"pa" 主因子方法，"ml" 极大似然方法.

相关矩阵或原始数据矩阵

观测值的个数

旋转方法："none", "varimax", "quartimax", "bentlerT", "equamax", "varimin", "geominT" and "bifactor" 是正交旋转. "Promax", "promax", "oblimin", "simplimax", "bentlerQ, "geominQ" and "biquartimin" and "cluster" 是斜交旋转.

因子得分的计算方法："regression" , "Thurstone", "Anderson" and "Bartlett".

```
> fa_x
Factor Analysis using method =  pa
Call: fa(r = cor_x, nfactors = 2, n.obs = 112, rotate = "none", scores = "regression",
    fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
        PA1   PA2   h2    u2 com
general 0.75  0.07 0.57 0.432 1.0
picture 0.52  0.32 0.38 0.623 1.7
blocks  0.75  0.52 0.83 0.166 1.8
maze    0.39  0.22 0.20 0.798 1.6
reading 0.81 -0.51 0.91 0.089 1.7
vocab   0.73 -0.39 0.69 0.313 1.5

                       PA1  PA2
SS loadings           2.75 0.83
Proportion Var        0.46 0.14
Cumulative Var        0.46 0.60
Proportion Explained  0.77 0.23
Cumulative Proportion 0.77 1.00

Mean item complexity =  1.5
Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are  15  and the objective function was  2.5 with Chi Square of  268
The degrees of freedom for the model are 4  and the objective function was  0.07

The root mean square of the residuals (RMSR) is  0.03
The df corrected root mean square of the residuals is  0.06

The harmonic number of observations is  112 with the empirical chi square  3.5  with prob <  0.48
The total number of observations was  112  with Likelihood Chi Square =  7.2  with prob <  0.13

Tucker Lewis Index of factoring reliability =  0.95
RMSEA index =  0.083  and the 90 % confidence intervals are  0 0.18
BIC =  -12
Fit based upon off diagonal values = 0.99
Measures of factor score adequacy
                                                PA1  PA2
Correlation of (regression) scores with factors   0.96 0.92
Multiple R square of scores with factors          0.93 0.84
Minimum correlation of possible factor scores     0.86 0.68
```

```
> fa_x
Factor Analysis using method =  pa
Call: fa(r = cor_x, nfactors = 2, n.obs = 112, rotate = "none", scores = "regression",
    fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
          PA1   PA2   h2    u2 com
general 0.75  0.07 0.57 0.432 1.0
picture 0.52  0.32 0.38 0.623 1.7
blocks  0.75  0.52 0.83 0.166 1.8
maze    0.39  0.22 0.20 0.798 1.6
reading 0.81 -0.51 0.91 0.089 1.7
vocab   0.73 -0.39 0.69 0.313 1.5

                       PA1  PA2
SS loadings           2.75 0.83
Proportion Var        0.46 0.14
Cumulative Var        0.46 0.60
Proportion Explained  0.77 0.23
Cumulative Proportion 0.77 1.00

Mean item complexity =  1.5
Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are  15  and the objective function was  2.5 with Chi Square of  268
The degrees of freedom for the model are 4  and the objective function was  0.07

The root mean square of the residuals (RMSR) is  0.03
The df corrected root mean square of the residuals is  0.06

The harmonic number of observations is  112 with the empirical chi square  3.5  with prob <  0.48
The total number of observations was  112  with Likelihood Chi Square =  7.2  with prob <  0.13

Tucker Lewis Index of factoring reliability =  0.95
RMSEA index =  0.083  and the 90 % confidence intervals are  0 0.18
BIC =  -12
Fit based upon off diagonal values = 0.99
Measures of factor score adequacy
                                                PA1  PA2
Correlation of (regression) scores with factors   0.96 0.92
Multiple R square of scores with factors          0.93 0.84
Minimum correlation of possible factor scores     0.86 0.68
```

因子载荷

因子载荷的意义不好解释，考虑因子旋转！

两个因子解释了 60% 的方差.

能力与智力测试 (Ability and Intelligence Tests)

- 例: 对 112 人进行了六项测试，这里给出了计算得出的协方差矩阵.
  - 正交旋转

```
# 正交旋转
fa_x.varimax = fa(cor_x, nfactors = 2, n.obs = 112, rotate = "varimax", fm = "pa", scores = "regression" )
fa_x.varimax
```

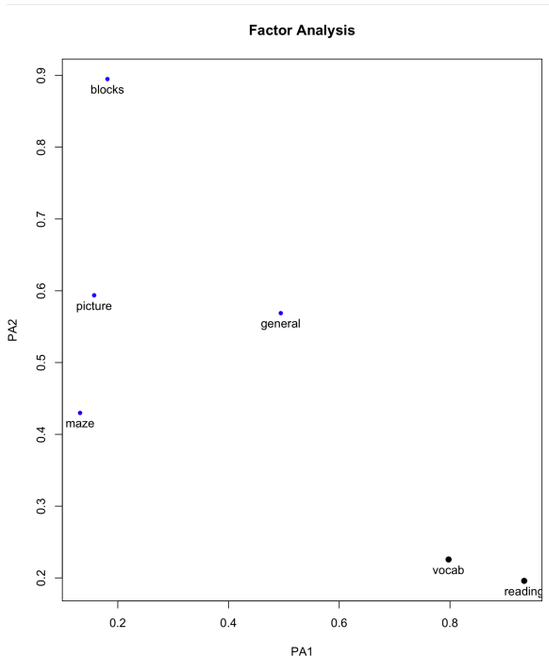
```
> fa_x.varimax
Factor Analysis using method =  pa
Call: fa(r = cor_x, nfactors = 2, n.obs = 112, rotate = "varimax",
    scores = "regression", fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
         PA1  PA2   h2    u2 com
general 0.49 0.57 0.57 0.432 2.0
picture 0.16 0.59 0.38 0.623 1.1
blocks  0.18 0.89 0.83 0.166 1.1
maze    0.13 0.43 0.20 0.798 1.2
reading 0.93 0.20 0.91 0.089 1.1
vocab   0.80 0.23 0.69 0.313 1.2

                       PA1  PA2
SS loadings           1.83 1.75
Proportion Var        0.30 0.29
Cumulative Var        0.30 0.60
Proportion Explained  0.51 0.49
Cumulative Proportion 0.51 1.00

Mean item complexity =  1.3
Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are  15  and the objective function was  2.5 with Chi Square of  268
The degrees of freedom for the model are 4  and the objective function was  0.07

The root mean square of the residuals (RMSR) is  0.03
The df corrected root mean square of the residuals is  0.06

The harmonic number of observations is  112 with the empirical chi square  3.5  with prob <  0.48
The total number of observations was  112  with Likelihood Chi Square =  7.2  with prob <  0.13

Tucker Lewis Index of factoring reliability =  0.95
RMSEA index =  0.083  and the 90 % confidence intervals are  0 0.18
BIC =  -12
Fit based upon off diagonal values = 0.99
Measures of factor score adequacy
                                                 PA1  PA2
Correlation of (regression) scores with factors   0.96 0.92
Multiple R square of scores with factors          0.91 0.85
Minimum correlation of possible factor scores     0.82 0.71
```

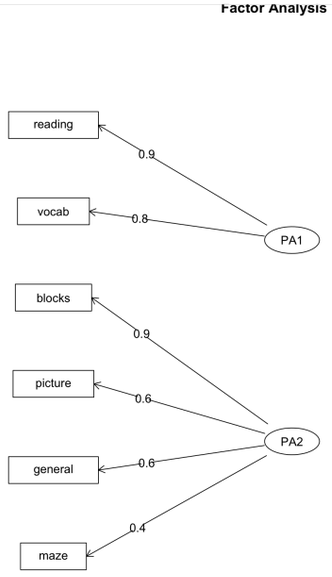
```
> fa_x.varimax
Factor Analysis using method =  pa
Call: fa(r = cor_x, nfactors = 2, n.obs = 112, rotate = "varimax",
    scores = "regression", fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
         PA1  PA2   h2    u2 com
general 0.49 0.57 0.57 0.432 2.0
picture 0.16 0.59 0.38 0.623 1.1
blocks  0.18 0.89 0.83 0.166 1.1
maze    0.13 0.43 0.20 0.798 1.2
reading 0.93 0.20 0.91 0.089 1.1
vocab   0.80 0.23 0.69 0.313 1.2

                       PA1  PA2
SS loadings           1.83 1.75
Proportion Var        0.30 0.29
Cumulative Var        0.30 0.60
Proportion Explained  0.51 0.49
Cumulative Proportion 0.51 1.00

Mean item complexity =  1.3
Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are  15  and the objective function was  2.5 with Chi Square of  268
The degrees of freedom for the model are 4  and the objective function was  0.07

The root mean square of the residuals (RMSR) is  0.03
The df corrected root mean square of the residuals is  0.06

The harmonic number of observations is  112 with the empirical chi square  3.5  with prob <  0.48
The total number of observations was  112  with Likelihood Chi Square =  7.2  with prob <  0.13

Tucker Lewis Index of factoring reliability =  0.95
RMSEA index =  0.083  and the 90 % confidence intervals are  0 0.18
BIC =  -12
Fit based upon off diagonal values = 0.99
Measures of factor score adequacy
                                                 PA1  PA2
Correlation of (regression) scores with factors 0.96 0.92
Multiple R square of scores with factors        0.91 0.85
Minimum correlation of possible factor scores   0.82 0.71
```

能力与智力测试 (Ability and Intelligence Tests)

- 例: 对 112 人进行了六项测试，这里给出了计算得出的协方差矩阵.
  - 正交旋转

```
# 正交旋转效果图
graphics.off()
par(mfrow = c(1, 2))
factor.plot(fa_x.varimax, labels = rownames(fa_x.varimax$loadings))
fa.diagram(fa_x.varimax, simple = TRUE)
```





能力与智力测试 (Ability and Intelligence Tests)