

统计I



- 数据集aqi_combine.csv描述：AQI指数（空气质量指数）AQI的指数的取值范围为0~500，其中0~50、51~100、101~200、201~300和大于300，分别对应国家空气质量标准中日均值的I级、II级、III级、IV级和V级标准的污染物浓度限定数值。
 - ➔ I级：空气质量评估为优，对人体健康无影响；
 - ➔ II级：空气质量评估为良，对人体健康无显著影响；
 - ➔ III级：为轻度污染，健康人群出现刺激症状；
 - ➔ IV级：中度污染，健康人群普遍出现刺激症状；
 - ➔ V级：严重污染，健康人群出现严重刺激症状。
- 主要污染物
 - ➔ 六项污染物质的浓度：其中PM2.5（粒径小于等于 $2.5\mu\text{m}$ 的颗粒物，也称细颗粒物），PM10（粒径小于等于 $10\mu\text{m}$ 的颗粒物，也称可吸入颗粒物），SO2（二氧化硫），NO2（二氧化氮）以及CO（一氧化碳）的浓度全部为24小时平均值，O3浓度值为8小时的滑动平均值。
- 时间跨度：2015年1月1日至2017年6月30日，共有912条记录。

- 使用ggplot2里的画图函数完成以下的练习：
 - ➔ 载入绘图相关数据包并加载数据集文件aqi_combine.csv，打印数据概况
 - ➔ 污染等级的频率和频数表
 - ➔ AQI指数的频数直方图
 - ➔ 数据集的第一列日期，请提取其中的年份，并转换成因子类型，画出分年份AQI密度曲线,设置主题为theme_bw
 - ➔ 主要污染物的频数统计，以及分污染等级对主要污染物进行频数统计
 - ➔ AQI指数与各类污染物的矩阵散点图
 - ➔ 是否下雨条件下分组AQI密度曲线,设置主题为theme_bw
 - ➔ 日均温度-AQI散点图和拟合曲线，分是否下雨情况下的日均温度箱线图
 - ➔ 温度-各类污染物散点图，设置布局为两行三列，是否下雨-各污染物浓度分组箱线图，布局为一行六列

基本统计

CH01

```
mean( x,  
      trim = 0,  
      na.rm = FALSE  
    )
```

```
median( x,  
        na.rm = FALSE  
    )
```

```
quantile( x,  
          props = seq(0,1,0.25),  
          na.rm = FALSE,  
          name = TRUE,  
          type = 7,  
          ...  
    )
```

```
weighted.mean( x,  
               w,  
               trim = 0,  
               na.rm = FALSE  
    )
```

length
min, max, range, sum
fivenum, IQR(四分位)
var, sd

见教材RiA的88页和help

密度函数	d
分布函数	p
分位数函数	q
随机数函数	r

dorm
pnorm
qnorm
rnorm

binom	二项分布
norm	正态分布
unit	均匀分布
beta	Beta分布
exp	指数分布
pois	泊松分布
t	t分布
chisq	卡方分布
logis	Logistic分布
...	...

`set.seed(1234)`: 设定随机数种子

随机采样函数

```
sample( x,
        size,
        replace = FALSE,
        prob = NULL
    )
```

见教材RiA的90页和help

```
> vars <- c("mpg", "hp", "wt")  
> head(mtcars[vars])
```

	mpg	hp	wt
Mazda RX4	19.4	110	2.620
Mazda RX4 Wag	19.4	110	2.875
Datsun 710	27.0	93	2.320
Hornet 4 Drive	20.9	110	3.215
Hornet Sportabout	12.2	175	3.440
Valiant	10.7	105	3.460

```
> summary(mtcars[vars])
```

	mpg	hp	wt
Min.	: 11.699	Min. : 52.0	Min. : 1.513
1st Qu.:	56.635	1st Qu.: 96.5	1st Qu.: 2.581
Median :	135.895	Median : 123.0	Median : 3.325
Mean :	262.350	Mean : 146.7	Mean : 3.217
3rd Qu.:	270.234	3rd Qu.: 180.0	3rd Qu.: 3.610
Max. :	1320.684	Max. : 335.0	Max. : 5.424

```
mystats <- function(x, na.omit = FALSE) {  
  if (na.omit)  
    x <- x[!is.na(x)]  
  m <- mean(x)  
  n <- length(x)  
  s <- sd(x)  
  skew <- sum((x - m)^3/s^3)/n  
  kurt <- sum((x - m)^4/s^4)/n - 3  
  return(c(n = n, mean = m, stdev = s, skew = skew, kurtosis = kurt))  
}
```

偏度
峰度

```
> sapply(mtcars[vars], mystats)
```

	mpg	hp	wt
n	3.200000e+01	32.00000000	32.00000000
mean	2.623503e+05	146.6875000	3.21725000
stdev	3.284077e+05	68.5628685	0.97845744
skew	1.841288e+00	0.7260237	0.42314646
kurtosis	2.466317e+00	-0.1355511	-0.02271075


```
> describe(mtcars[vars])
```

```
mtcars[vars]
```

describe(mtcars[vars])

```
3 Variables      32 Observations
```

```
mpg
```

n	missing	unique	Info	Mean	.05	.10	.25	.50	.75
32	0	25	1	262350	22474	42304	56635	135895	270234
.90	.95								
824210	965638								

```
lowest : 11699 31290 41816 46695 50625
highest: 456976 555457 854072 1101996 1320684
```

```
hp
```

n	missing	unique	Info	Mean	.05	.10	.25	.50	.75
32	0	22	1	146.7	63.65	66.00	96.50	123.00	180.00
.90	.95								
243.50	253.55								

```
lowest : 52 62 65 66 91, highest: 215 230 245 264 335
```

```
wt
```

n	missing	unique	Info	Mean	.05	.10	.25	.50	.75
32	0	29	1	3.217	1.736	1.956	2.581	3.325	3.610
.90	.95								
4.048	5.293								

```
lowest : 1.513 1.615 1.835 1.935 2.140, highest: 3.845 4.070 5.250 5.345 5.424
```

```
> stat.desc(mtcars[vars])
```

	mpg	hp	wt
nbr.val	3.200000e+01	32.0000000	32.0000000
nbr.null	0.000000e+00	0.0000000	0.0000000
nbr.na	0.000000e+00	0.0000000	0.0000000
min	1.169859e+04	52.0000000	1.5130000
max	1.320684e+06	335.0000000	5.4240000
range	1.308985e+06	283.0000000	3.9110000
sum	8.395209e+06	4694.0000000	102.9520000
median	1.358954e+05	123.0000000	3.3250000
mean	2.623503e+05	146.6875000	3.2172500
SE.mean	5.805483e+04	12.1203173	0.1729685
CI.mean.0.95	1.184036e+05	24.7195501	0.3527715
var	1.078516e+11	4700.8669355	0.9573790
std.dev	3.284077e+05	68.5628685	0.9784574
coef.var	1.251791e+00	0.4674077	0.3041285

```
> library(psych)
```

```
Warning message:
```

```
package 'psych' was built under R version 3.4.4
```

```
> vars <- c("mpg", "hp", "wt")
```

```
> describe(mtcars[vars])
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis
mpg	1	32	20.09	6.03	19.20	19.70	5.41	10.40	33.90	23.50	0.61	-0.37
hp	2	32	146.69	68.56	123.00	141.19	77.10	52.00	335.00	283.00	0.73	-0.14
wt	3	32	3.22	0.98	3.33	3.15	0.77	1.51	5.42	3.91	0.42	-0.02

```
se
```

```
mpg 1.07
```

```
hp 12.12
```

```
wt 0.17
```

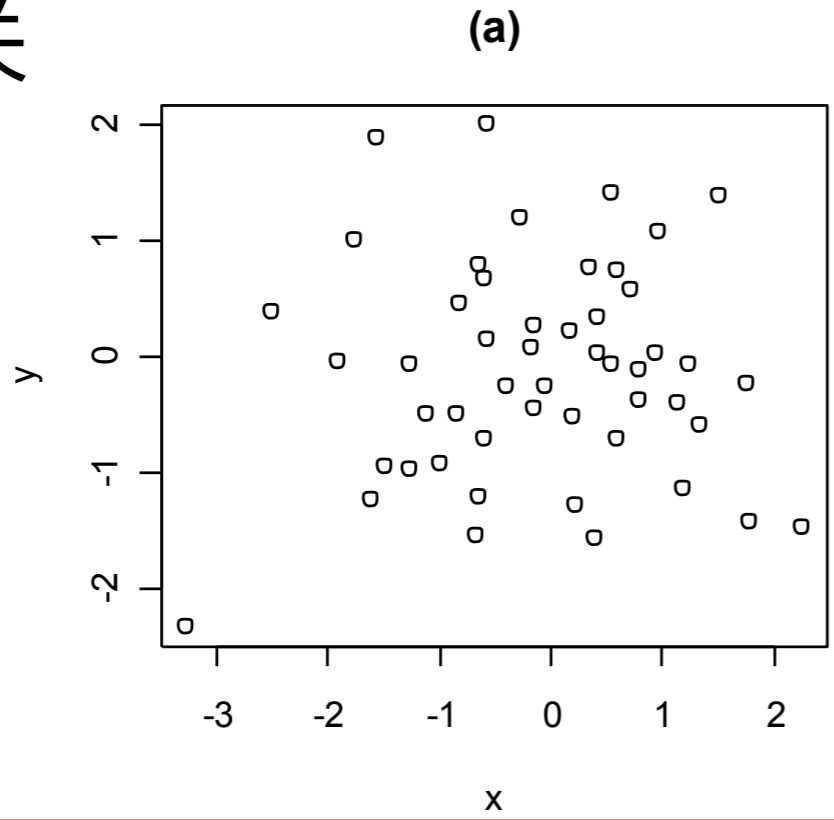
回归分析

CH08

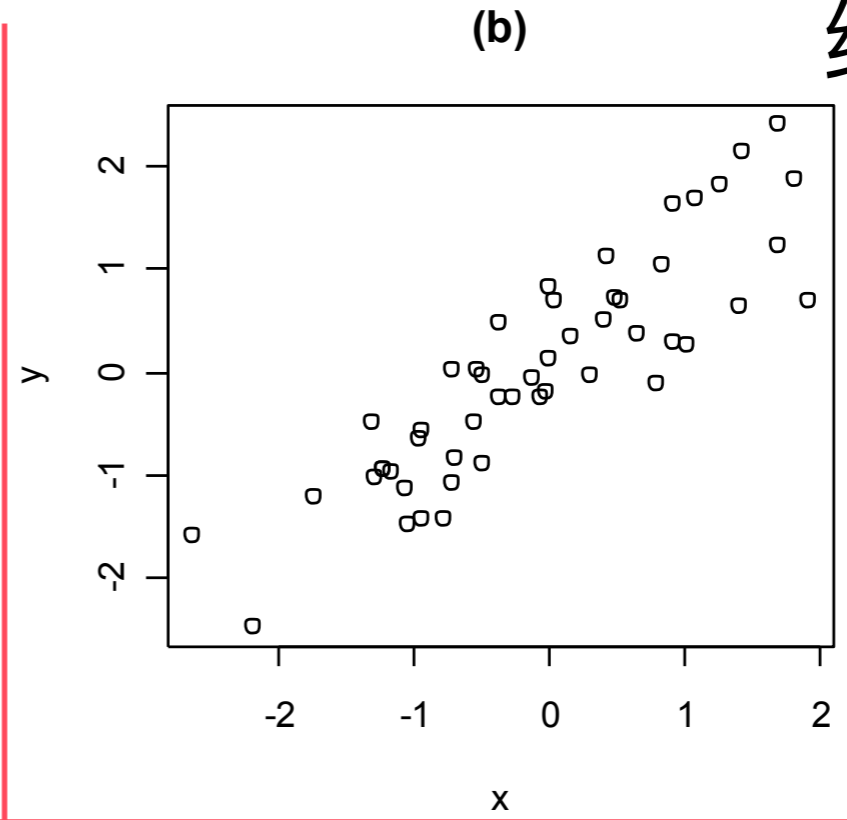
- 发现变量之间的统计关系，并且用此规律来帮助我们进行决策才是统计实践的最终目的。
- 一般来说，统计可以根据目前所拥有的信息（数据）来建立人们所关心的变量和其他有关变量的关系。这种关系一般称为模型 (model)
- 假如用 Y 表示感兴趣的变量，用 X 表示其他可能与 Y 有关的变量（ X 也可能是若干变量组成的向量）。则所需要的是建立一个函数关系 $Y=f(X)$ 。
- 这里 Y 称为因变量或响应变量(dependent variable, response variable)，而 X 称为自变量，也称为解释变量或协变量(independent variable, explanatory variable, covariate)。建立这种关系的过程就叫做回归(regression)

- 一旦建立了回归模型，除了对变量的关系有了进一步的定量理解之外，还可以利用该模型（函数）通过自变量对因变量做预测（prediction）。
- 这里所说的预测，是用已知的自变量的值通过模型对未知的因变量值进行估计

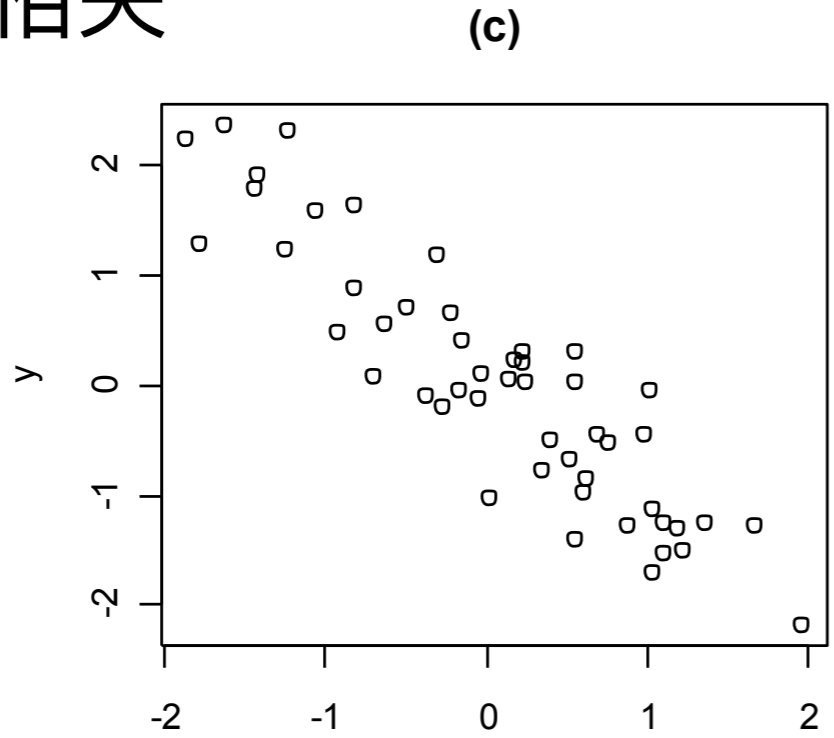
不相关



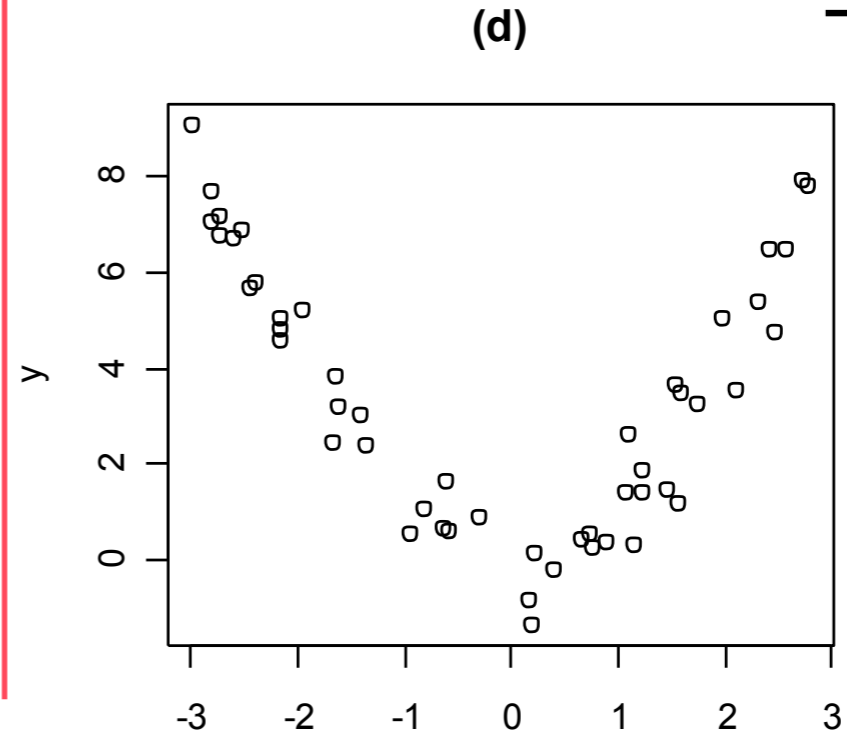
线性正相关



线性负相关



非线性相关



- Pearson相关系数 (Pearson's correlation coefficient) 又叫相关系数或线性相关系数。它一般用字母 r 表示。它是由两个变量的样本取值得到，这是一个描述线性相关强度的量，取值于-1和1之间。当两个变量有很强的线性相关时，相关系数接近于1（正相关）或-1（负相关），而当两个变量不那么线性相关时，相关系数就接近0
- Kendall τ 相关系数 (Kendall's τ) 这里的度量原理是把所有的样本点配对（如果每一个点由 x 和 y 组成的坐标 (x,y) 代表，一对点就是诸如 (x_1,y_1) 和 (x_2,y_2) 的点对），然后看每一对中的 x 和 y 的观测值是否同时增加（或减少）。比如由点对 (x_1,y_1) 和 (x_2,y_2) ，可以算出乘积 $(x_2-x_1)(y_2-y_1)$ 是否大于0；如果大于0，则说明 x 和 y 同时增长或同时下降，称这两点协同 (concordant)；否则就是不协同。如果样本中协同的点数目多，两个变量就更加相关一些；如果样本中不协同 (discordant) 的点数目多，两个变量就不很相关

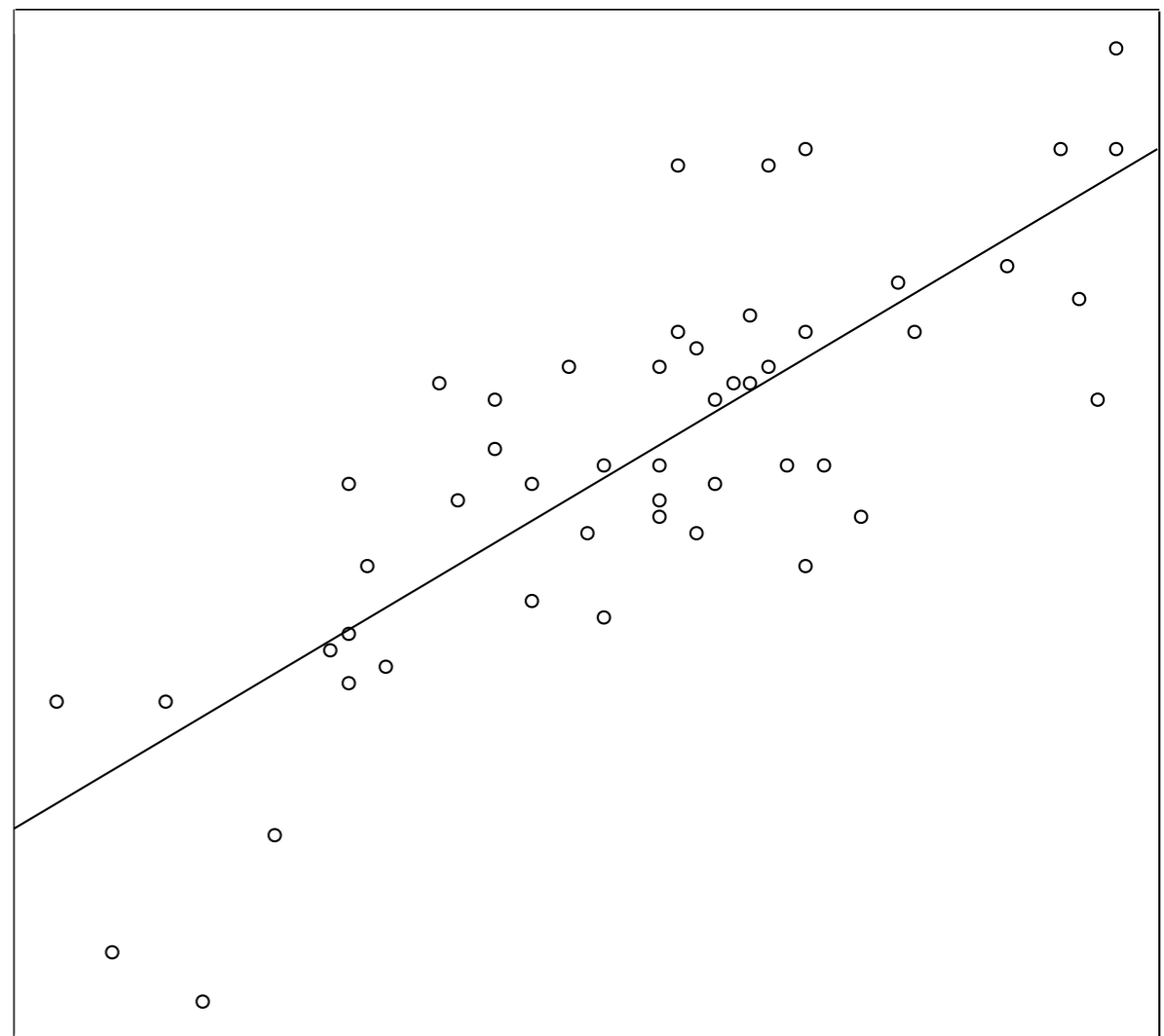
- Spearman 秩相关系数 (Spearman rank correlation coefficient 或 Spearman's ρ) 它和 Pearson 相关系数定义有些类似, 只不过在定义中把点的坐标换成各自样本的秩 (即样本点大小的“座次”)。Spearman 相关系数也是取值在 -1 和 1 之间, 也有类似的解释。通过它也可以进行不依赖于总体分布的非参数检验。

- 两个变量的数据进行线性回归，就是要找到一条直线来适当地代表那些点的趋势。
- 首先需要确定选择这条直线的标准。这里介绍最小二乘回归（least squares regression）。古汉语“二乘”是平方的意思。
- 这就是寻找一条直线，使得所有点到该直线的垂直距离的平方和最小。用数据寻找一条直线的过程也叫做拟合（fit）一条直线

$$y = \beta_0 + \beta_1 x + \varepsilon$$

$$y = 26.44 + 0.65x$$

截距=26.444; 斜率=0.651



回归类型	用途
简单线性	用一个量化的解释变量预测一个量化的响应变量
多项式	用一个量化的解释变量预测一个量化的响应变量，模型的关系是n阶多项式
多元线性	用两个或多个量化的解释变量预测一个量化的响应变量
多变量	用一个或多个解释变量预测多个响应变量
Logistic	用一个或多个解释变量预测一个类别型响应变量
泊松	用一个或多个解释变量预测一个代表频数的响应变量
Cox比例风险	用一个或多个解释变量预测一个事件（死亡、失败或旧病复发）发生的时间
时间序列	对误差项相关的时间序列数据建模
非线性	用一个或多个量化的解释变量预测一个量化的响应变量，不过模型是非线性的
非参数	用一个或多个量化的解释变量预测一个量化的响应变量，模型的形式源自数据形式，不事先设定
稳健	用一个或多个量化的解释变量预测一个量化的响应变量，能抵御强影响点的干扰

- **lm**(format, data)
- $y \sim x_1 + x_2 + \dots + x_k$

符 号	用 途
~	分隔符号，左边为响应变量，右边为解释变量。例如，要通过x、z和w预测y，代码为 $y \sim x + z + w$
+	分隔预测变量
:	表示预测变量的交互项。例如，要通过x、z及x与z的交互项预测y，代码为 $y \sim x + z + x:z$
*	表示所有可能交互项的简洁方式。代码 $y \sim x * z * w$ 可展开为 $y \sim x + z + w + x:z + x:w + z:w + x:z:w$
^	表示交互项达到某个次数。代码 $y \sim (x + z + w)^2$ 可展开为 $y \sim x + z + w + x:z + x:w + z:w$
.	表示包含除因变量外的所有变量。例如，若一个数据框包含变量x、y、z和w，代码 $y \sim .$ 可展开为 $y \sim x + z + w$
-	减号，表示从等式中移除某个变量。例如， $y \sim (x + z + w)^2 - x:w$ 可展开为 $y \sim x + z + w + x:z + z:w$
-1	删除截距项。例如，表达式 $y \sim x - 1$ 拟合y在x上的回归，并强制直线通过原点
I()	从算术的角度来解释括号中的元素。例如， $y \sim x + (z + w)^2$ 将展开为 $y \sim x + z + w + z:w$ 。相反，代码 $y \sim x + I((z + w)^2)$ 将展开为 $y \sim x + h$ ，h是一个由z和w的平方和创建的新变量
function	可以在表达式中用的数学函数。例如， $\log(y) \sim x + z + w$ 表示通过x、z和w来预测 $\log(y)$

函 数	用 途
<code>summary()</code>	展示拟合模型的详细结果
<code>coefficients()</code>	列出拟合模型的模型参数（截距项和斜率）
<code>confint()</code>	提供模型参数的置信区间（默认95%）
<code>fitted()</code>	列出拟合模型的预测值
<code>residuals()</code>	列出拟合模型的残差值
<code>anova()</code>	生成一个拟合模型的方差分析表，或者比较两个或更多拟合模型的方差分析表
<code>vcov()</code>	列出模型参数的协方差矩阵
<code>AIC()</code>	输出赤池信息统计量
<code>plot()</code>	生成评价拟合模型的诊断图
<code>predict()</code>	用拟合模型对新的数据集预测响应变量值


```
> fit <- lm(weight ~ height, data = women)
> summary(fit)
```

```
Call:
lm(formula = weight ~ height, data = women)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-1.7333 -1.1333 -0.3833  0.7417  3.1167
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -87.51667    5.93694  -14.74 1.71e-09 ***
height       3.45000    0.09114   37.85 1.09e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.525 on 13 degrees of freedom
Multiple R-squared:  0.991,    Adjusted R-squared:  0.9903
F-statistic: 1433 on 1 and 13 DF,  p-value: 1.091e-14
```

```
> head(women)
  height weight
1     58    115
2     59    117
3     60    120
4     61    123
5     62    126
6     63    129
```

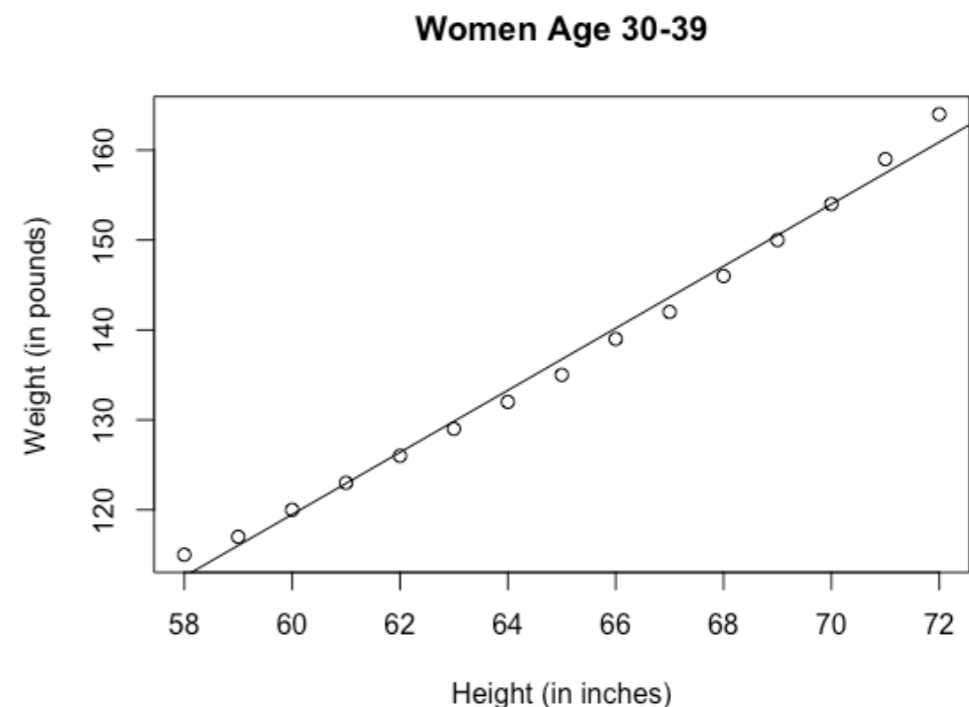
$$\widehat{\text{Weight}} = -87.52 + 3.45 \times \text{Height}$$

```

> women$weight
 [1] 115 117 120 123 126 129 132 135 139 142 146 150 154 159 164
> fitted(fit)
      1      2      3      4      5      6      7      8      9     10
112.5833 116.0333 119.4833 122.9333 126.3833 129.8333 133.2833 136.7333 140.1833 143.6333
     11     12     13     14     15
147.0833 150.5333 153.9833 157.4333 160.8833
> residuals(fit)
      1      2      3      4      5      6      7
 2.41666667  0.96666667  0.51666667  0.06666667 -0.38333333 -0.83333333 -1.28333333
      8      9     10     11     12     13     14
-1.73333333 -1.18333333 -1.63333333 -1.08333333 -0.53333333  0.01666667  1.56666667
     15
 3.11666667
    
```

```

plot(women$height,
     women$weight,
     main = "Women Age 30-39",
     xlab = "Height (in inches)",
     ylab = "Weight (in pounds)")
abline(fit)
    
```



```
> fit2 <- lm(weight ~ height + I(height^2), data = women)
> summary(fit2)
```

```
Call:
lm(formula = weight ~ height + I(height^2), data = women)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-0.50941 -0.29611 -0.00941  0.28615  0.59706
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	261.87818	25.19677	10.393	2.36e-07 ***
height	-7.34832	0.77769	-9.449	6.58e-07 ***
I(height^2)	0.08306	0.00598	13.891	9.32e-09 ***

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

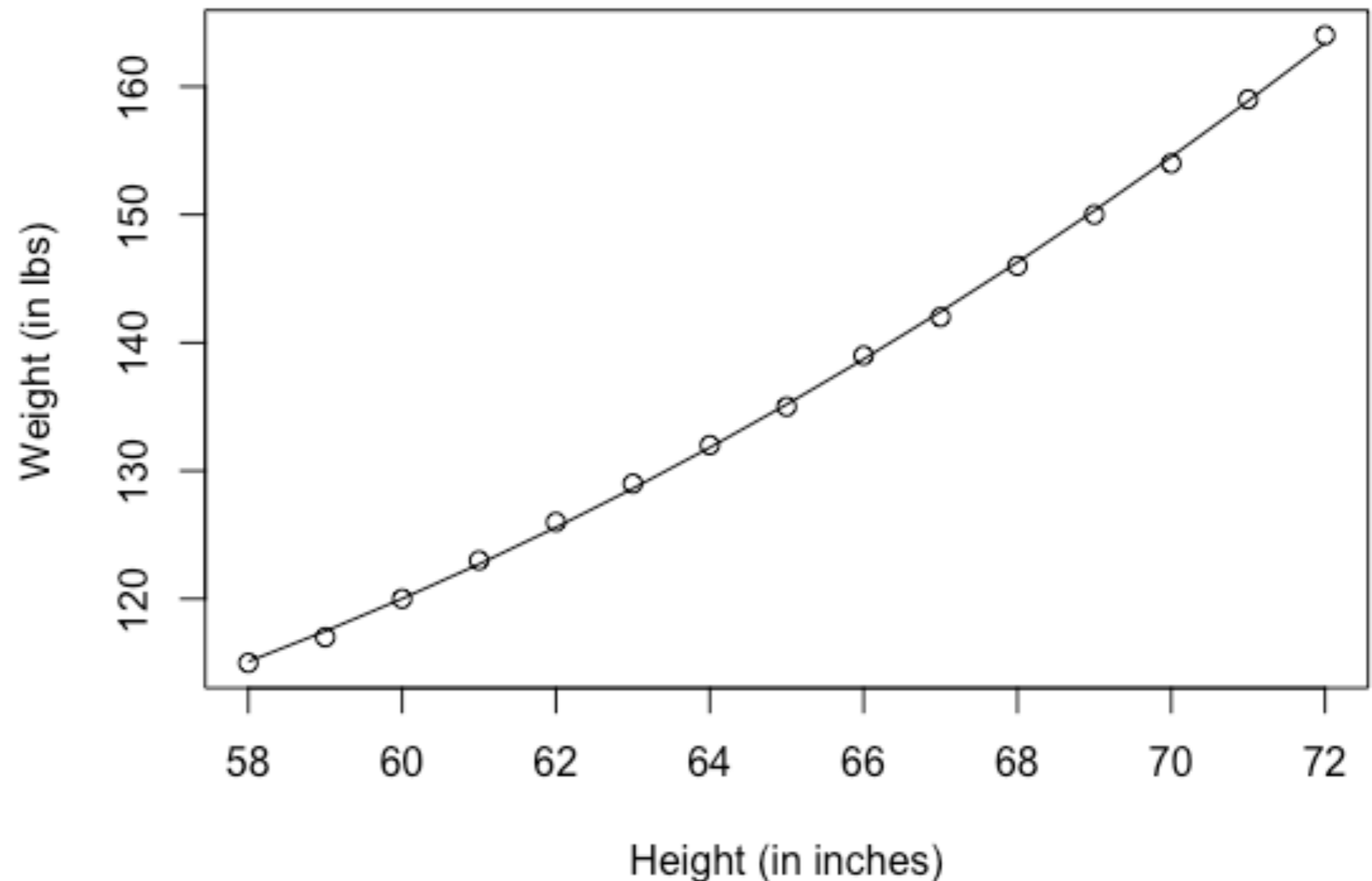
```
Residual standard error: 0.3841 on 12 degrees of freedom
Multiple R-squared:  0.9995,    Adjusted R-squared:  0.9994
F-statistic: 1.139e+04 on 2 and 12 DF,  p-value: < 2.2e-16
```

$$\widehat{\text{Weight}} = 261.88 - 7.35 \times \text{Height} + 0.083 \times \text{Height}^2$$


```
plot(women$height, women$weight, main = "Women Age 30-39",  
     xlab = "Height (in inches)", ylab = "Weight (in lbs)")
```

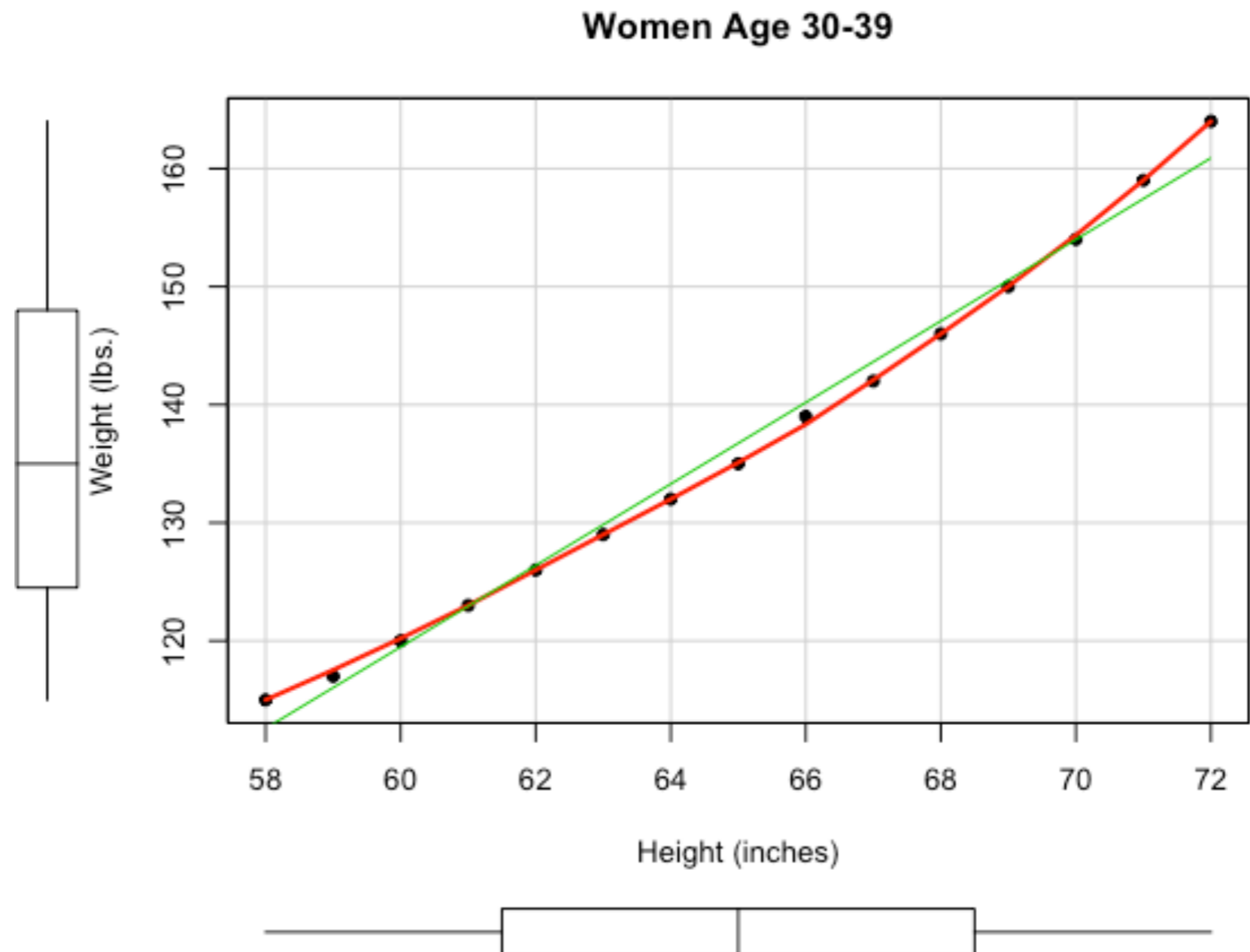
```
lines(women$height, fitted(fit2))
```

Women Age 30-39



```
library(car)
```

```
scatterplot(weight ~ height, data = women, spread = FALSE,  
            ty.smooth = 2, pch = 19, main = "Women Age 30-39",  
            xlab = "Height (inches)", ylab = "Weight (lbs.)")
```



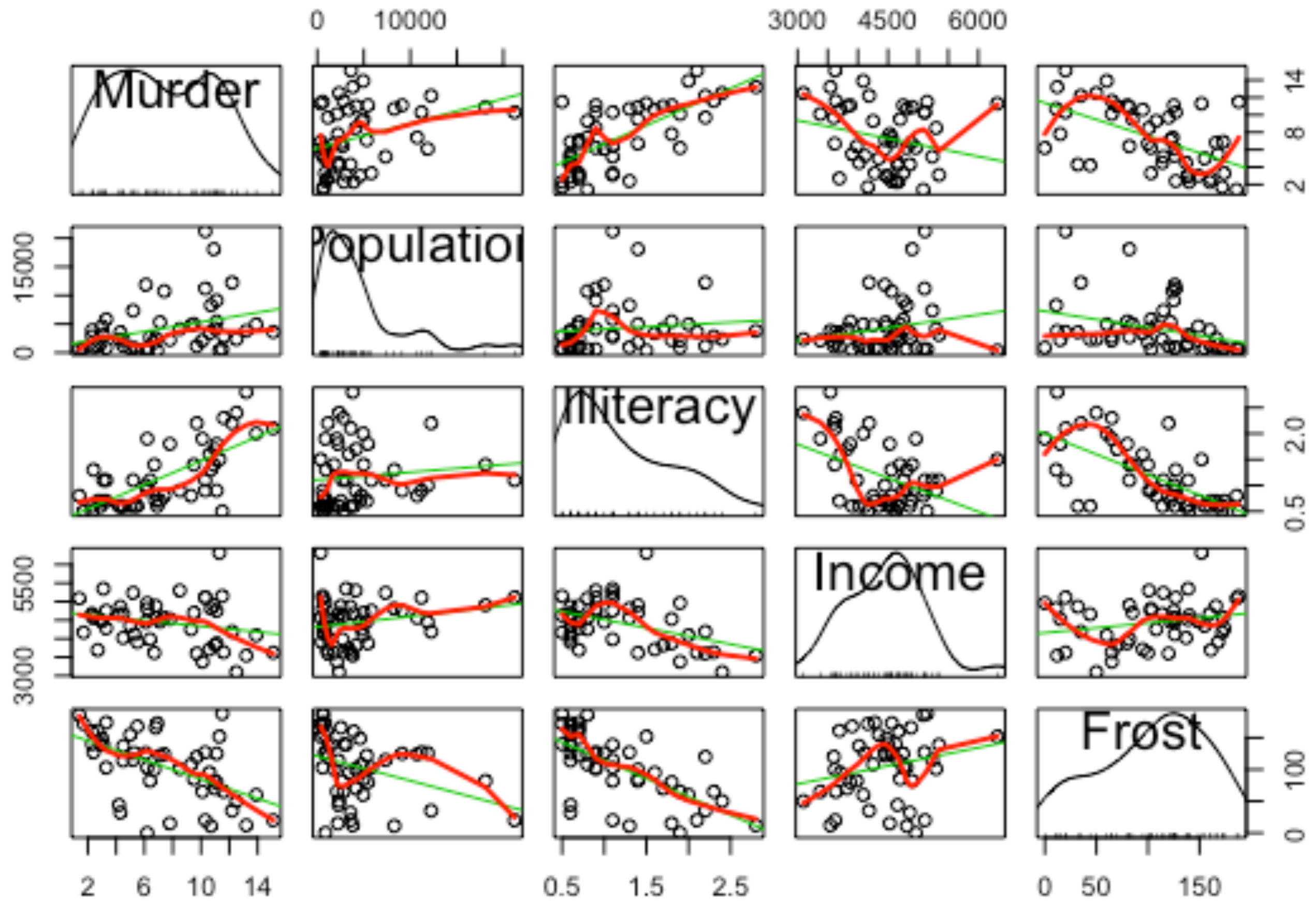
```
> states <- as.data.frame(state.x77[, c("Murder", "Population",  
+   "Illiteracy", "Income", "Frost")])  
>  
> cor(states)
```

	Murder	Population	Illiteracy	Income	Frost
Murder	1.0000000	0.3436428	0.7029752	-0.2300776	-0.5388834
Population	0.3436428	1.0000000	0.1076224	0.2082276	-0.3321525
Illiteracy	0.7029752	0.1076224	1.0000000	-0.4370752	-0.6719470
Income	-0.2300776	0.2082276	-0.4370752	1.0000000	0.2262822
Frost	-0.5388834	-0.3321525	-0.6719470	0.2262822	1.0000000

```
library(car)
```

```
scatterplotMatrix(states, spread = FALSE, lty.smooth = 2,  
                  main = "Scatterplot Matrix")
```

Scatterplot Matrix




```
> fit <- lm(Murder ~ Population + Illiteracy + Income +
+ Frost, data = states)
> summary(fit)
```

Call:

```
lm(formula = Murder ~ Population + Illiteracy + Income + Frost,
    data = states)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-4.7960 -1.6495 -0.0811  1.4815  7.6210
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	1.235e+00	3.866e+00	0.319	0.7510	
Population	2.237e-04	9.052e-05	2.471	0.0173	*
Illiteracy	4.143e+00	8.744e-01	4.738	2.19e-05	***
Income	6.442e-05	6.837e-04	0.094	0.9253	
Frost	5.813e-04	1.005e-02	0.058	0.9541	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.535 on 45 degrees of freedom

Multiple R-squared: 0.567, Adjusted R-squared: 0.5285

F-statistic: 14.73 on 4 and 45 DF, p-value: 9.133e-08

```
> fit <- lm(mpg ~ hp + wt + hp:wt, data = mtcars)
> summary(fit)
```

Call:

```
lm(formula = mpg ~ hp + wt + hp:wt, data = mtcars)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.0632	-1.6491	-0.7362	1.4211	4.5513

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	49.80842	3.60516	13.816	5.01e-14	***
hp	-0.12010	0.02470	-4.863	4.04e-05	***
wt	-8.21662	1.26971	-6.471	5.20e-07	***
hp:wt	0.02785	0.00742	3.753	0.000811	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.153 on 28 degrees of freedom

Multiple R-squared: 0.8848, Adjusted R-squared: 0.8724

F-statistic: 71.66 on 3 and 28 DF, p-value: 2.981e-13

因子分析

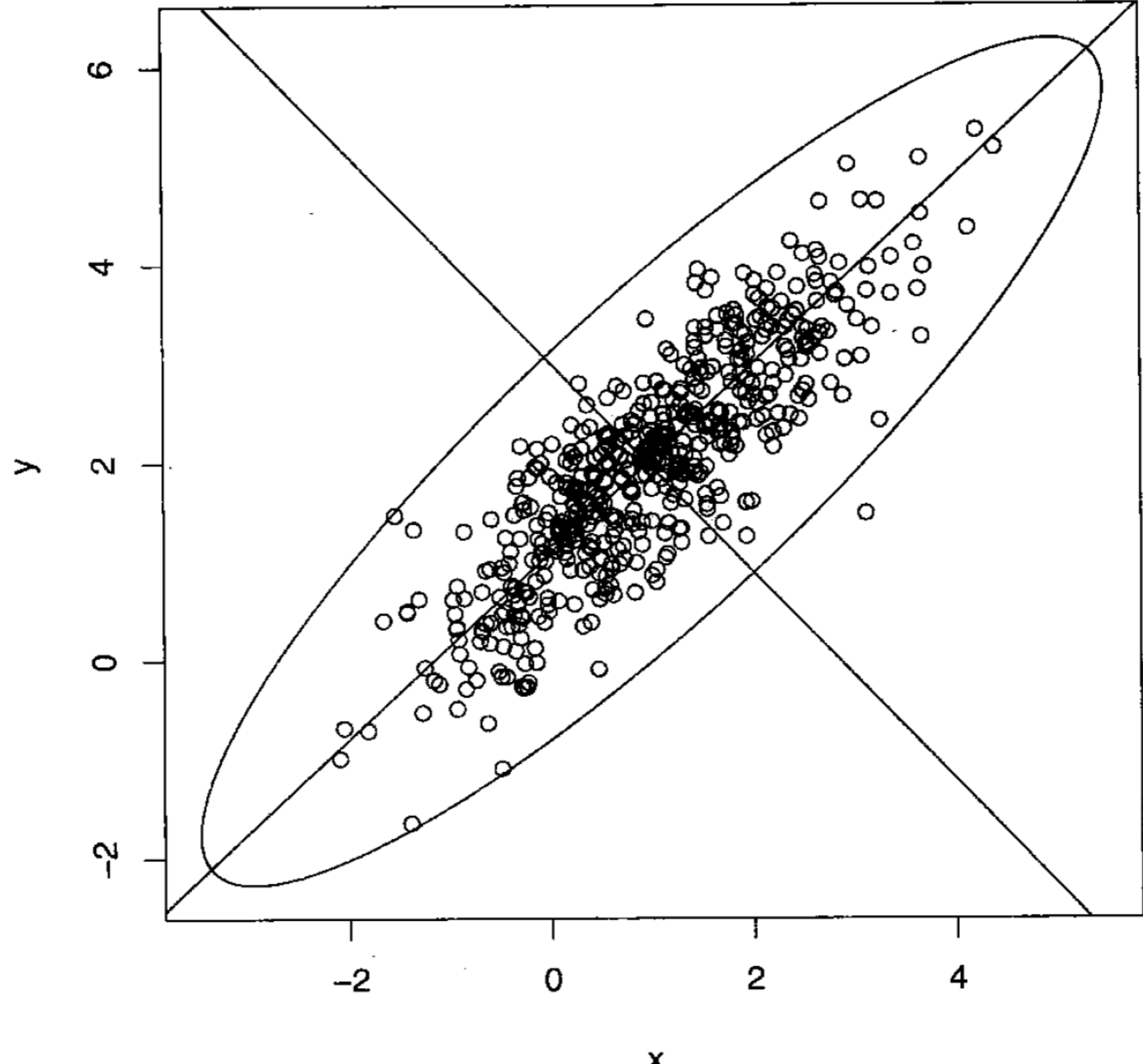
主成分分析

- 假定你是一个公司的财务经理，掌握了公司的所有数据，这包括众多的变量，如：固定资产、流动资金、借贷的数额和期限、各种税费、工资支出、原料消耗、产值、利润、折旧、职工人数、分工和教育程度等等
- 如果让你向上级或有关方面介绍公司状况，你能够把这些指标和数字都原封不动地摆出去吗？
- 在如此多的变量之中，有很多是相关的。人们希望能够找出它们的少数“代表”来对它们进行描述。
- 需要把这种有很多变量的数据进行高度概括。
- 如果每个变量都是独立的，因子分析就没有意义

- 两种把变量维数降低以便于描述、理解和分析的方法：主成分分析（principal component analysis）和因子分析（factor analysis）
- 实际上主成分分析可以说是因子分析的一个特例
- 两个方法的目的的一样，都是寻找众多相关变量的少数代表，这些代表变量称为成分或因子，都是原来变量的线性组合，由于代表变量的数目显著的小于原来变量数目，数据维度也就显著降低了
- 主成分分析发展早，因子分析发展晚，但是结果更理想

- 假设数据只有两个变量的观测值，即二维数据，如果两个变量特别由横轴和纵轴所代表
- 当坐标轴和椭圆的长短轴平行，那么代表长轴的变量就描述了数据的主要变化，而代表短轴的变量就描述了数据的次要变化。
- 但是，坐标轴通常并不和椭圆的长短轴平行。因此，需要寻找椭圆的长短轴，并进行变换，使得新变量和椭圆的长短轴平行。
- 如果长轴变量代表了数据包含的大部分信息，就用该变量代替原先的两个变量（舍去次要的一维），降维就完成了。

二维空间主成分分析



- 多维变量的情况和二维类似，也有高维的椭球，只不过不那么直观罢了
- 首先把高维椭球的主轴找出来，再用代表大多数数据信息的最长的几个轴作为新变量；这样，主成分分析就基本完成了
- 正如二维椭圆有两个主轴，三维椭球有三个主轴一样，有几个变量，就有几个主轴。
- 和二维情况类似，高维椭球的主轴也是互相垂直的。
- 这些互相正交的新变量是原先变量的线性组合，叫做主成分(principal component)。

- 选择越少的主成分，降维就越好。什么是标准呢？
- 那就是这些被选的主成分所代表的主轴的长度之和占了主轴长度总和的大部分。
- 有些文献建议，所选的主轴总长度占有所有主轴长度之和的大约85%即可，其实，这只是一个大体的说法；具体选几个，要看实际情况而定。

- 162个国家和地区的10个变量组成的数据，变量情况如下：
 - x1: 青少年生育率 (%)
 - x2: 人均国家收入
 - x3: 女小学生入学率 (%)
 - x4: 男小学生入学率
 - x5: 人口增长率 (%)
 - x6: 城镇人口比率 (%)
 - x7: 年龄中位数 (%)
 - x8: 60岁以上比例 (%)
 - x9: 15岁以下比例 (%)
 - x10: 每女性生育数
- 能不能把这10个变量用1-2个综合变量来表示，这1-2个综合变量包含多少原来变量信息，如何解释

- `w <- read.table("01_who.txt",sep=" ",header=T)`
- `b <- eigen (cor(w))`
- `data.frame(bva,bva/sum(b$va),cumsum(b$va)/sum(b$va))`

```
> w <- read.table("01_who.txt",sep=" ",header=T)
> b <- eigen (cor(w))
> data.frame(b$va,b$va/sum(b$va),cumsum(b$va)/sum(b$va))
```

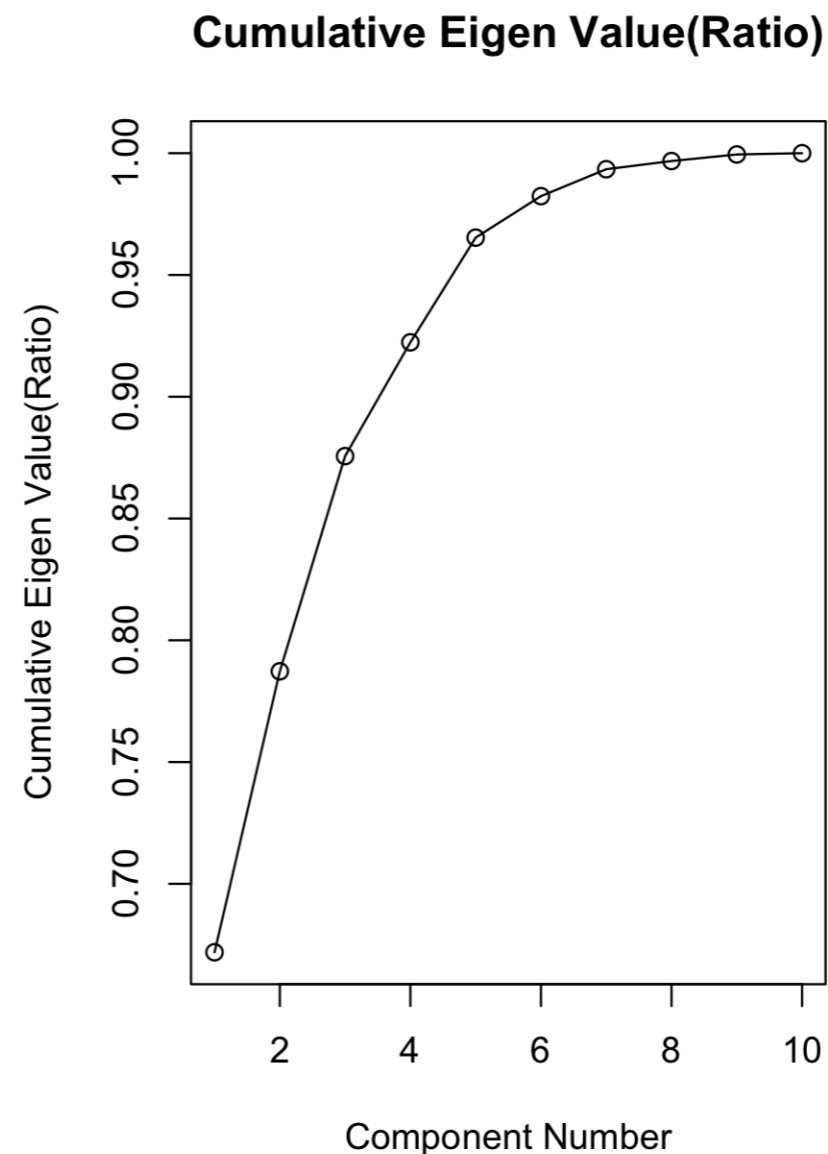
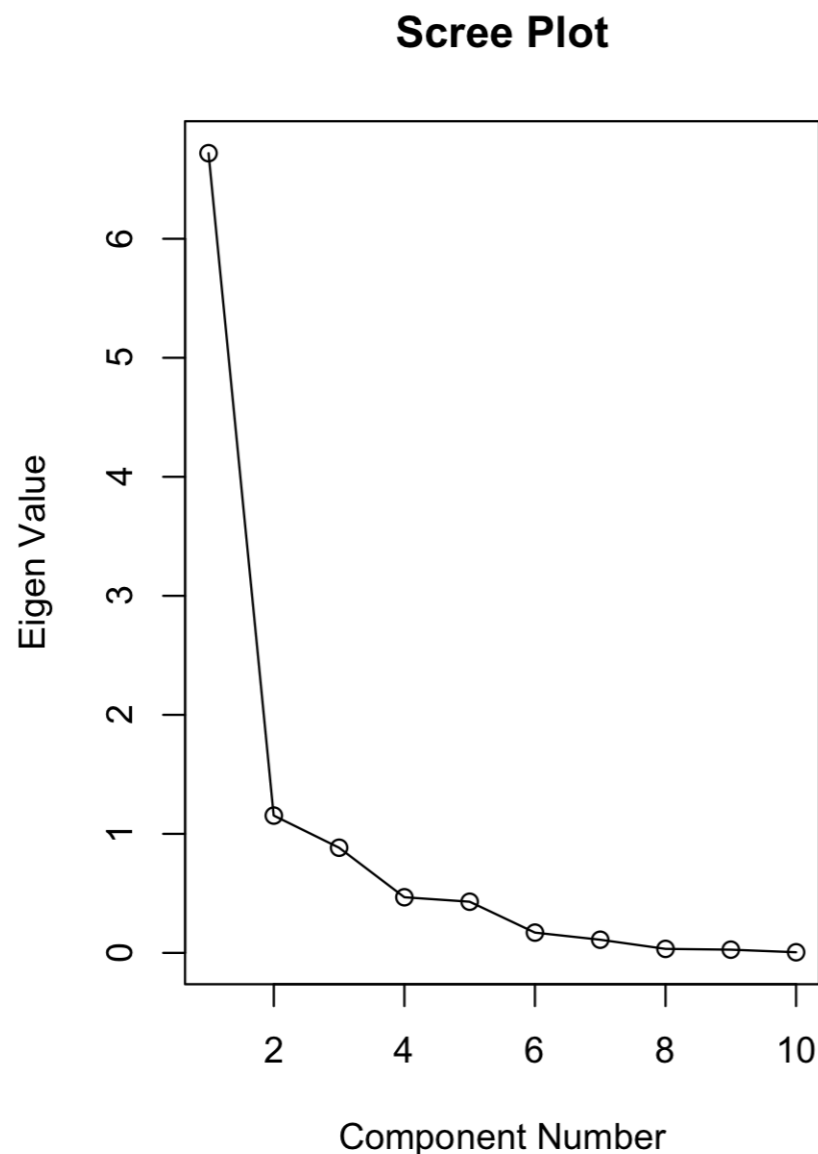
	b.va	b.va.sum.b.va.	cumsum.b.va..sum.b.va.
1	6.718991161	0.6718991161	0.6718991
2	1.153587902	0.1153587902	0.7872579
3	0.883542757	0.0883542757	0.8756122
4	0.467350145	0.0467350145	0.9223472
5	0.429855650	0.0429855650	0.9653328
6	0.170309196	0.0170309196	0.9823637
7	0.110557931	0.0110557931	0.9934195
8	0.033578573	0.0033578573	0.9967773
9	0.027010327	0.0027010327	0.9994784
10	0.005216358	0.0005216358	1.0000000

- **cor**(x, use=, method=)
- **cor.test**(x, y, alternative= , method=)

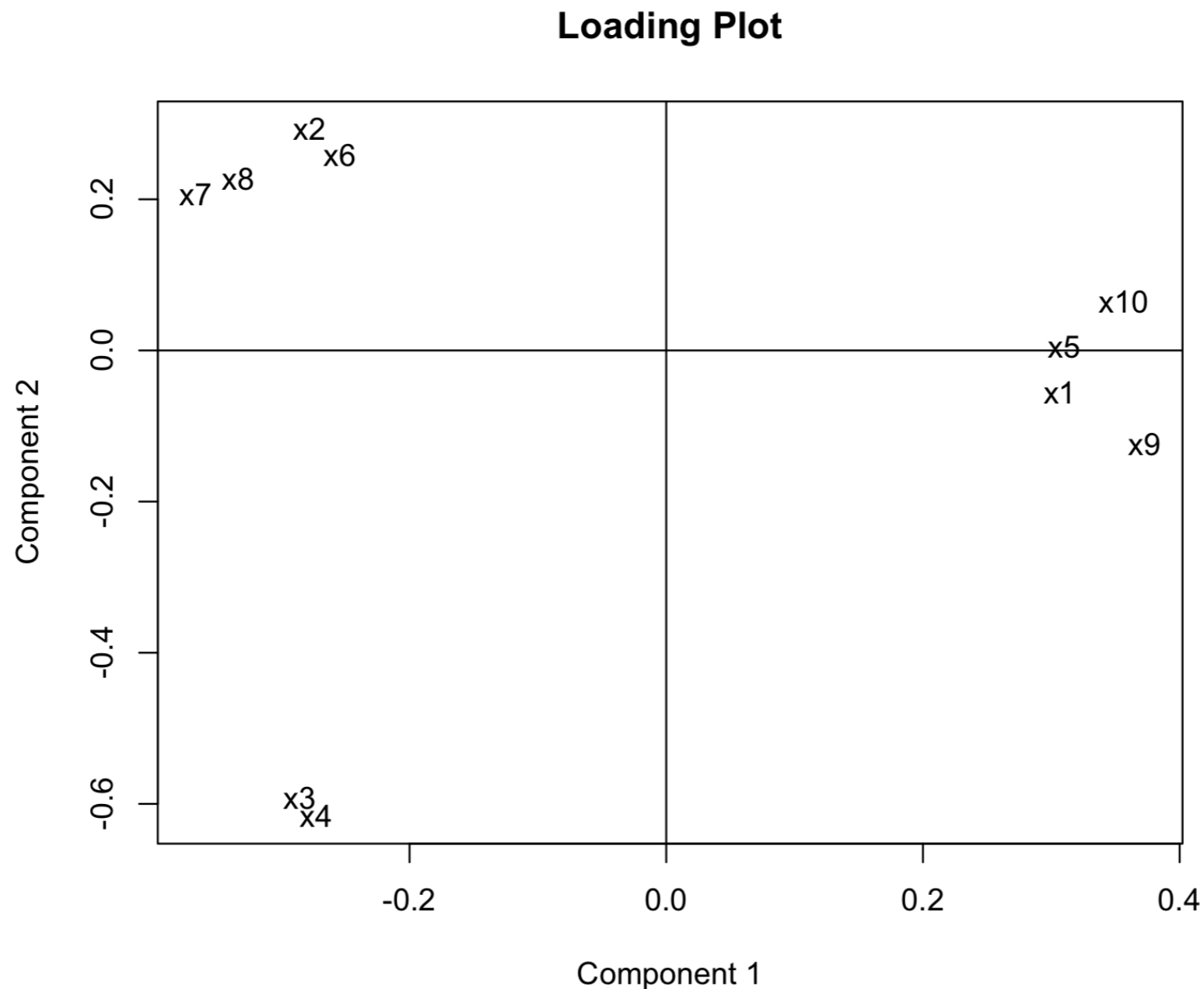
参 数	描 述
x	矩阵或数据框
use	指定缺失数据的处理方式。可选的方式为all.obs（假设不存在缺失数据——遇到缺失数据时将报错）、everything（遇到缺失数据时，相关系数的计算结果将被设为missing）、complete.obs（行删除）以及pairwise.complete.obs（成对删除，pairwise deletion）
method	指定相关系数的类型。可选类型为pearson、spearman或kendall

见教材RiA的146页

- `par(mfrow=c(1,2))`
- `plot(b$va,type="o",main="Scree Plot",xlab="Component Number",ylab="Eigen Value")`
- `plot(cumsum(b$va)/sum(b$va),type="o", main="Cumulative Eigen Value(Ratio)", xlab="Component Number", ylab="Cumulative Eigen Value(Ratio)")`



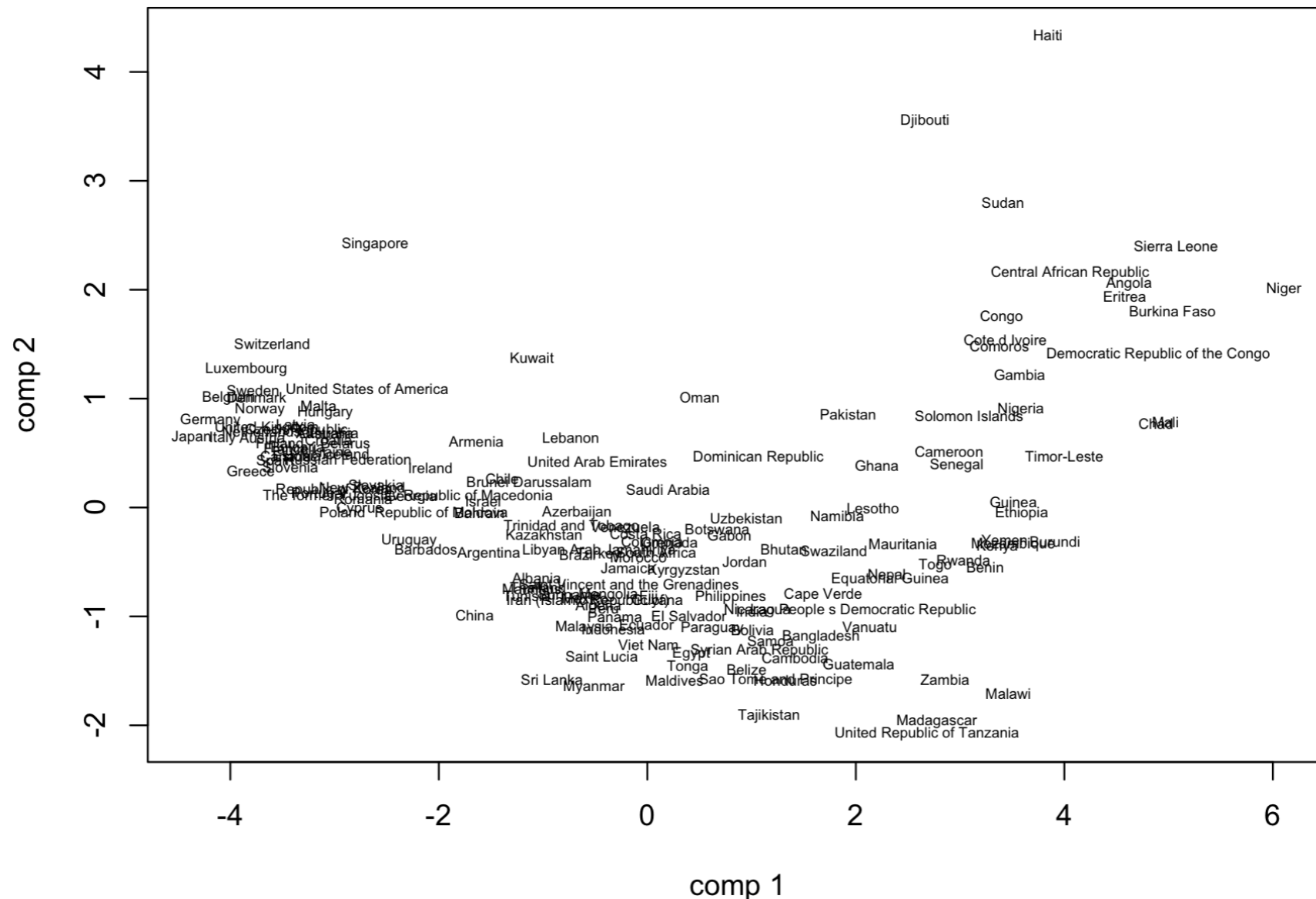
- `plot(b$ve[, 1:2], type="n", main="Loading Plot", xlab="Component 1", ylab="Component 2")`
- `abline(h=0); abline(v=0); text(b$ve[, 1:2], names(w))`



- `(loading <- sweep(b$ve,2,sqrt(b$va),"*"))`

```
> (loading <- sweep(b$ve,2,sqrt(b$va),"*"))
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]
[1,]  0.7950001 -0.060144065  0.008697677  0.56507852 -0.135066198  0.143678554 -0.07787578  0.0007345938 -0.0037926986
[2,] -0.7208566  0.314824979 -0.475837423  0.07682501  0.309396920  0.210140824  0.09473734  0.0023668077  0.0068380772
[3,] -0.7406579 -0.636943492 -0.160185149  0.05713586  0.014479830 -0.003239975  0.02145828 -0.0992238977 -0.0792299571
[4,] -0.7067703 -0.661901672 -0.176245805  0.10130616  0.058068620 -0.056390300 -0.00204842  0.1011608888  0.0651573982
[5,]  0.8042734  0.004766166 -0.507466756 -0.08369953  0.176395939 -0.097088710 -0.21565387 -0.0340943603  0.0195477392
[6,] -0.6595830  0.276611798 -0.507240181  0.04462954 -0.467825127 -0.095992853  0.03252340  0.0042275839 -0.0008730082
[7,] -0.9502772  0.220799909  0.093096669  0.10903561  0.093429548 -0.050609487 -0.10008490  0.0327552657 -0.0497531293
[8,] -0.8648319  0.242879593  0.193619843  0.28717471  0.154519908 -0.200829636  0.01477382 -0.0685856110  0.0612307083
[9,]  0.9659989 -0.134339443 -0.128011327  0.02461353 -0.005201777 -0.028266696  0.14930443 -0.0552491310  0.0597440017
[10,] 0.9236953  0.068687733 -0.151546515  0.15698910  0.172326059 -0.199956059  0.12231945  0.0590242908 -0.0790538720
      [,10]
[1,] -0.0014715092
[2,] -0.0038134883
[3,] -0.0012919074
[4,]  0.0011586092
[5,] -0.0007199536
[6,] -0.0006228413
[7,]  0.0530022135
[8,] -0.0153927042
[9,]  0.0454607638
[10,] -0.0091039202
```

- `wl <- as.matrix(scale(w))`
- `plot(wl %*% b$ve[, 1:2], type="n", xlab="comp 1", ylab="comp 2")`
- `text(wl %*% b$ve[, 1:2], row.names(w), cex=0.5)`



- 主成分分析从原理上是寻找椭球的所有主轴，有几个变量就有几个主成分，因子分析需要事前确定要找多少成分（也叫因子）
-
- `w <- read.table("01_who.txt",sep=" ",header=T)`
 - `a <- factanal(w,2,scores="regression")`
 - `a$loadings`

```
> a$loadings
```

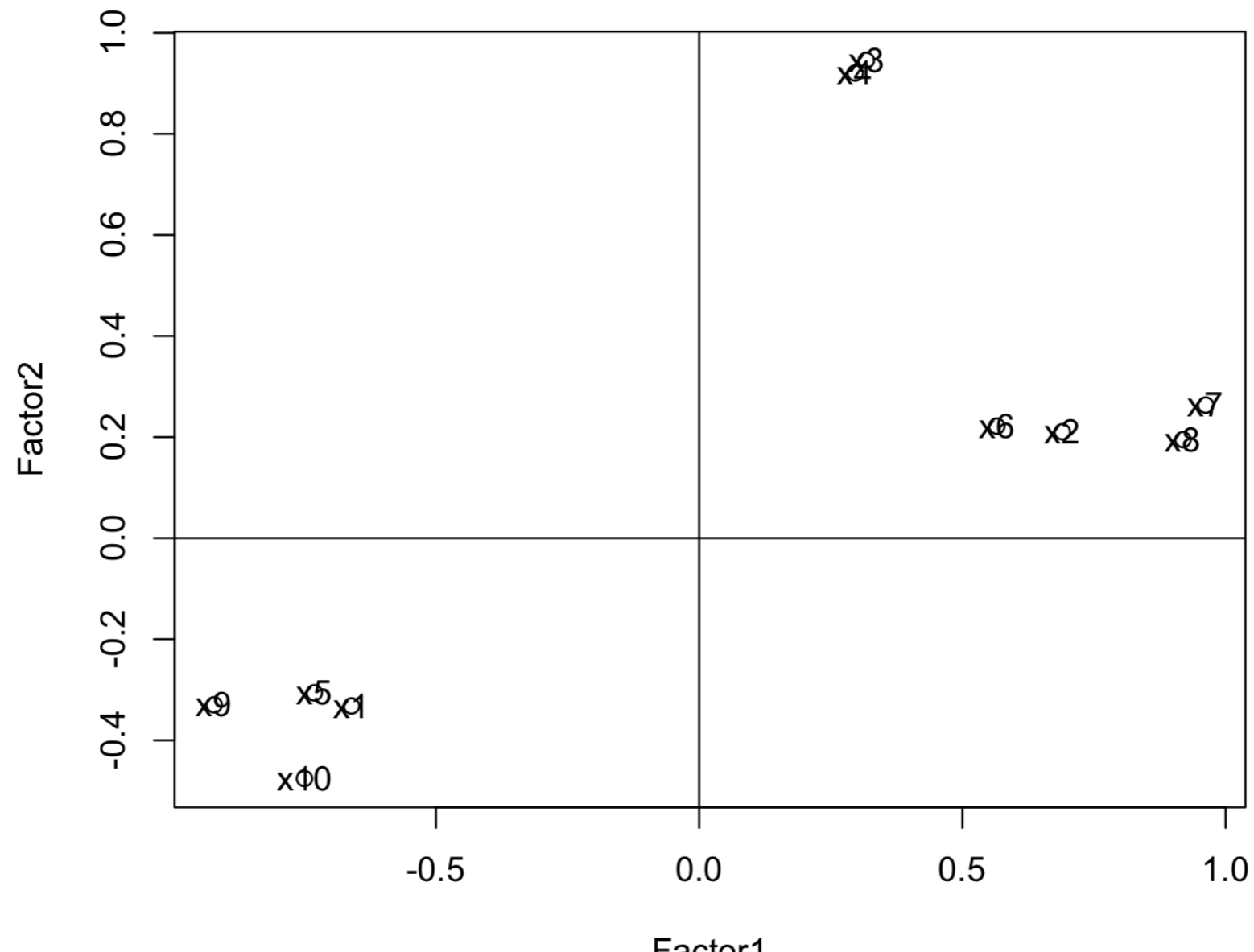
Loadings:

	Factor1	Factor2
x1	-0.660	-0.332
x2	0.690	0.211
x3	0.318	0.946
x4	0.296	0.920
x5	-0.731	-0.306
x6	0.565	0.222
x7	0.962	0.263
x8	0.919	0.195
x9	-0.921	-0.330
x10	-0.750	-0.476

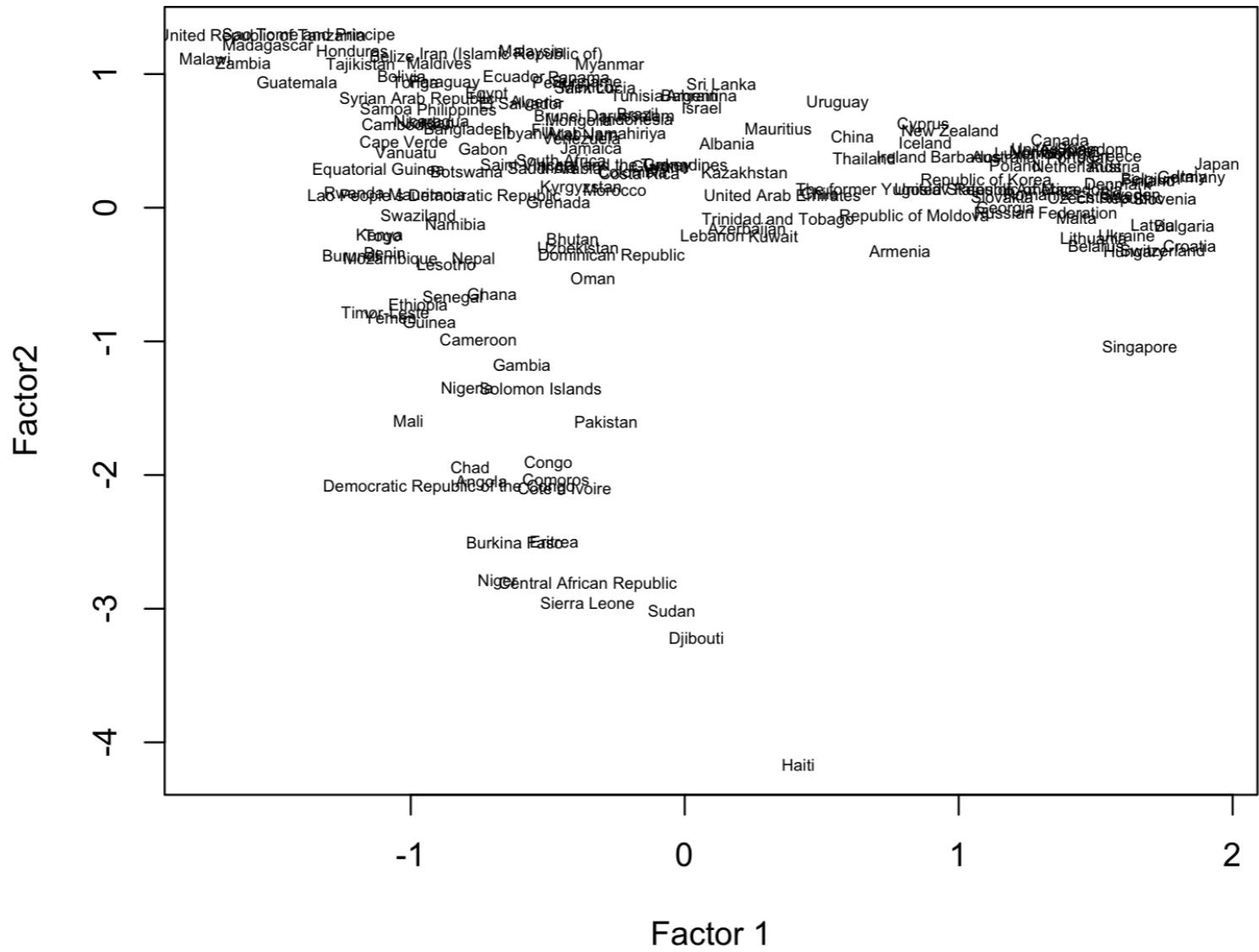
	Factor1	Factor2
SS loadings	5.136	2.481
Proportion Var	0.514	0.248
Cumulative Var	0.514	0.762

$$\begin{aligned}
 x_1 &= -0.6604421f_1 - 0.3320026f_2 \\
 x_2 &= 0.6897585f_1 + 0.2105372f_2 \\
 x_3 &= 0.3175013f_1 + 0.9456340f_2 \\
 x_4 &= 0.2964161f_1 + 0.9204082f_2 \\
 x_5 &= -0.7311456f_1 - 0.3061111f_2 \\
 x_6 &= 0.5654772f_1 + 0.2217057f_2 \\
 x_7 &= 0.9621764f_1 + 0.2633326f_2 \\
 x_8 &= 0.9189977f_1 + 0.1949305f_2 \\
 x_9 &= -0.9212964f_1 - 0.3297795f_2 \\
 x_{10} &= -0.7497948f_1 - 0.4757015f_2
 \end{aligned}$$

- `plot(a$loadings)`
- `abline(h=0); abline(v=0); text(a$loadings[,1:2],names(w))`



- `plot(a$scores,type="n",xlab="Factor 1",ylab="Factor2")`
- `text(a$scores,row.names(w),cex=0.5)`



在某中学随机抽取某年级 30 名学生，测量其身高 (X_1)、体重 (X_2)、胸围 (X_3) 和坐高 (X_4)，数据如表 9.1 所示。试对这 30 名中学生身体四项指标数据做主成分分析。

- **princomp()**
- **summary()**
- **predict()**
- **screeplot()**
- **biplot()**

表 9.1: 30 名中学生身体四项指标数据

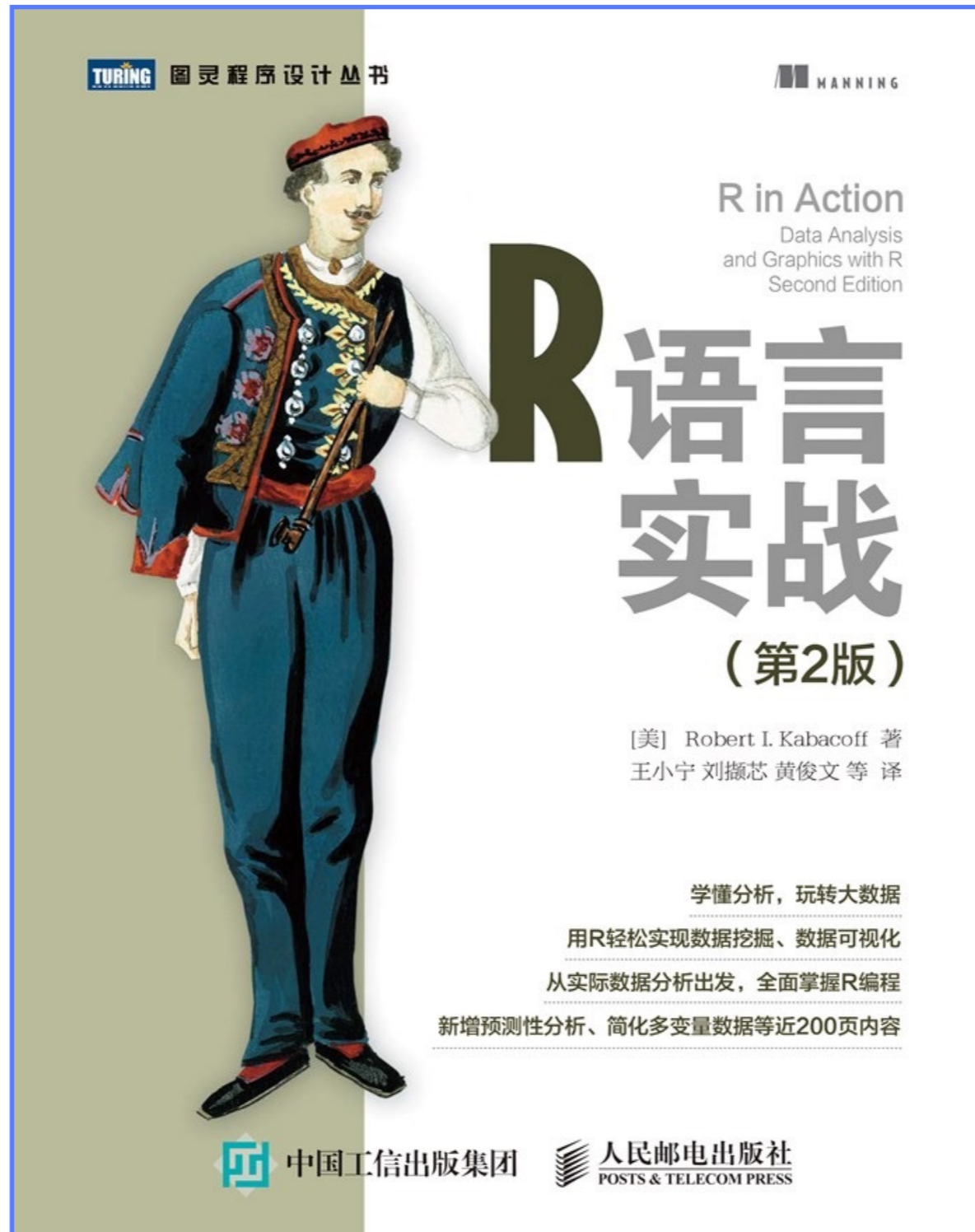
序号	X_1	X_2	X_3	X_4	序号	X_1	X_2	X_3	X_4
1	148	41	72	78	16	152	35	73	79
2	139	34	71	76	17	149	47	82	79
3	160	49	77	86	18	145	35	70	77
4	149	36	67	79	19	160	47	74	87
5	159	45	80	86	20	156	44	78	85
6	142	31	66	76	21	151	42	73	82
7	153	43	76	83	22	147	38	73	78
8	150	43	77	79	23	157	39	68	80
9	151	42	77	80	24	147	30	65	75
10	139	31	68	74	25	157	48	80	88
11	140	29	64	74	26	151	36	74	80
12	161	47	78	84	27	144	36	68	76
13	158	49	78	83	28	141	30	67	76
14	140	33	67	77	29	139	32	68	73
15	137	31	66	73	30	148	38	70	78

提问时间!

孙惠平

sunhp@ss.pku.edu.cn

练习



第8章(8.1和8.2)

第14章

INTERACTIVE COURSE

Introduction to Regression in R

[Start Course For Free](#) [Bookmark](#)

🕒 4 hours ▶ 14 Videos <> 52 Exercises 👤 9,590 Participants 📊 4,050 XP

提交方式和上节课一样!

<https://www.datacamp.com/courses>

INTERACTIVE COURSE

Factor Analysis in R

[Start Course For Free](#) [Bookmark](#)

🕒 4 hours ▶ 13 Videos <> 45 Exercises 👤 6,263 Participants 📊 3,600 XP

可选

谢谢!

孙惠平

sunhp@ss.pku.edu.cn