Regression and Correlation Methods

Linear Regression

- Regression: $y \sim x$

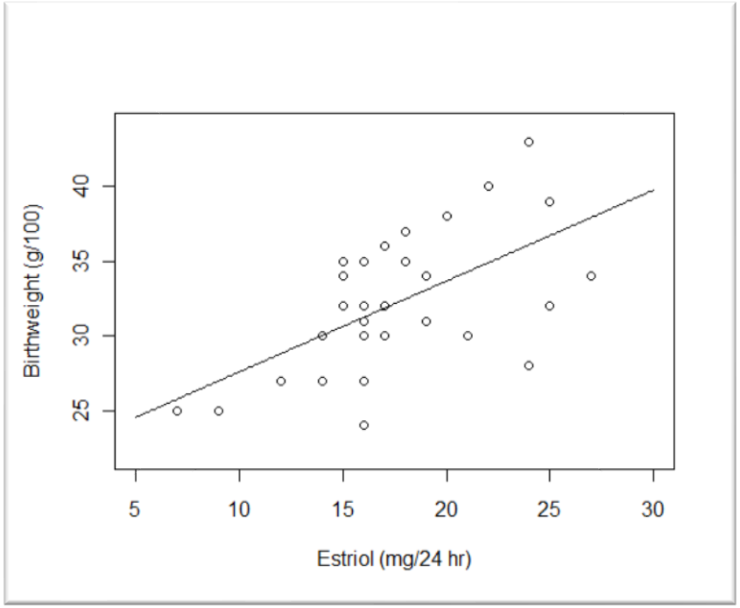
- Simple linear regression

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

dependent variable

independent variable

$$e \sim \mathcal{N}(0, \sigma^2)$$



- Multiple regression model

$$y = \alpha + \sum_{j=1}^{k} \beta_j x_j + e, e \sim N(0, \sigma^2)$$

Estimation

- In the early days of Statistics, regression analysis started off by using the absolute value norm:

$$L_1(\alpha, \beta) = \sum |d_i| = \sum |y_i - \beta_0 - \beta_1 x_i|$$

- But, it does not yield analytical solution.

- Alternatively, the least squares approach is used:

$$L_1(\alpha, \beta) = \sum {d_i}^2 = \sum (y_i - \beta_0 - \beta_1 x_i)^2$$

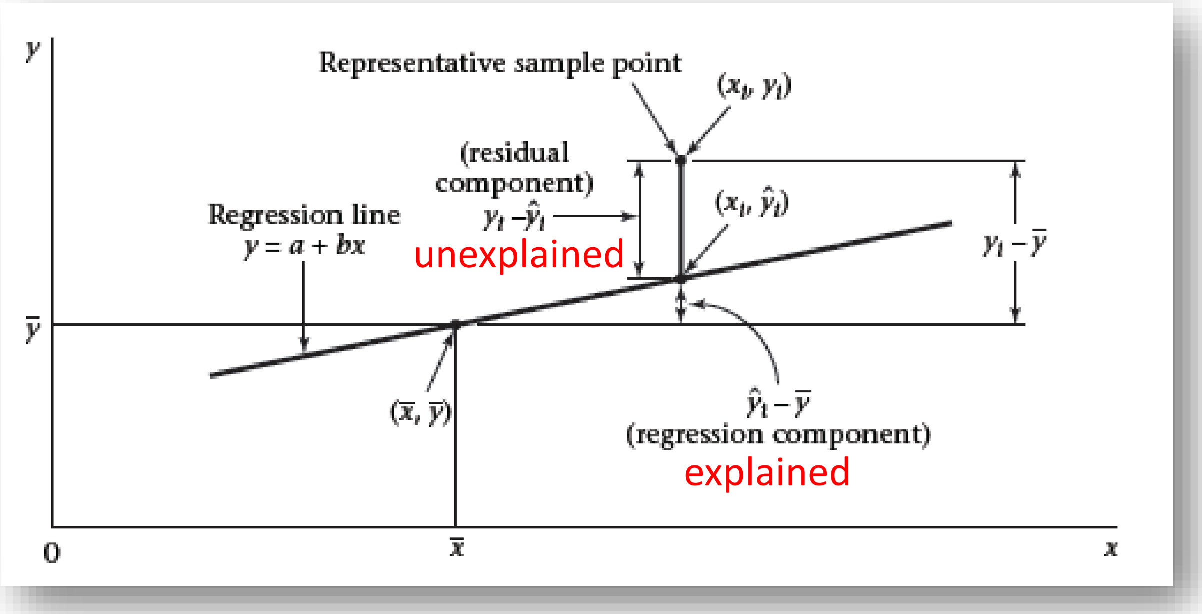
- MLE: maximum likelihood estimation
- Hypothesis testing on parameters

Variance decomposition

$$\sum_{i}^{n}(y_i - \bar{y})^2 = \sum_{i}^{n}(\hat{y}_i - \bar{y})^2 + \sum_{i}^{n}(y_i - \hat{y})^2$$

total sum of square regression sum of square residual sum of square

TSS = Reg SS + Res SS



Issues in MLR

- Collinear: two or more predictor variables in a multiple regression model are highly correlated.
  - The collinearity can make it impossible to identify the specific effects of each variable
  - Omit the redundant parameters

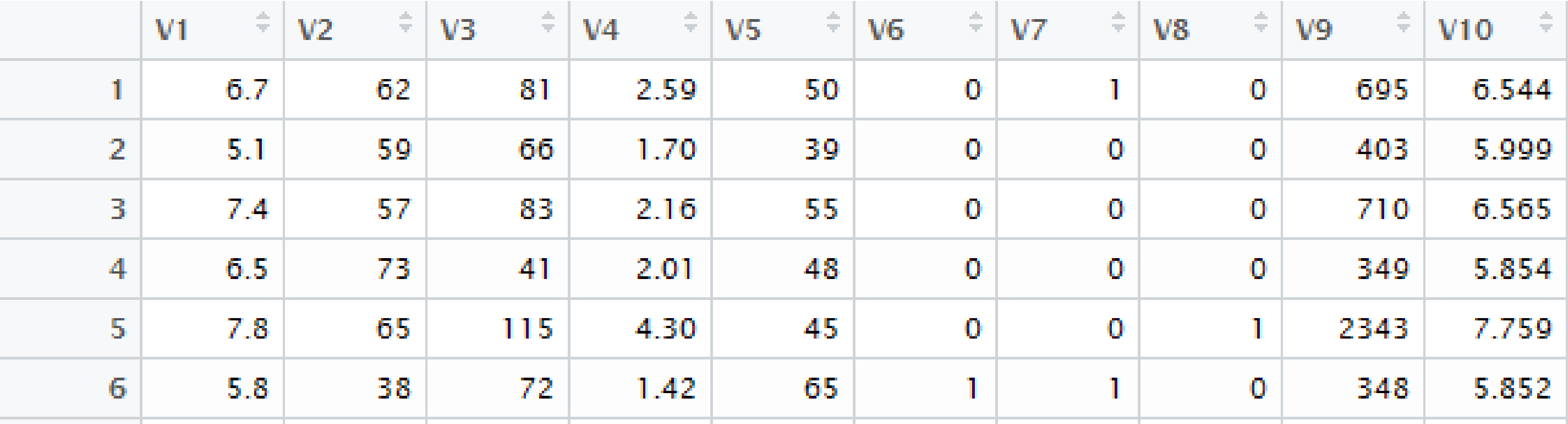
- High dimensional data: dimension reduction or variable selection

- Model complexity

例子：Surgical Unit

背景：一个医院的外科手术小组对一群正在接受一种特殊类型的肝脏手术的病人的存活时间感兴趣。

数据描述：在这个例子中，共收集到54个样本，10个变量，在这里我们只考虑其中的6个变量



	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
1	6.7	62	81	2.59	50	0	1	0	695	6.544
2	5.1	59	66	1.70	39	0	0	0	403	5.999
3	7.4	57	83	2.16	55	0	0	0	710	6.565
4	6.5	73	41	2.01	48	0	0	0	349	5.854
5	7.8	65	115	4.30	45	0	0	1	2343	7.759
6	5.8	38	72	1.42	65	1	1	0	348	5.852

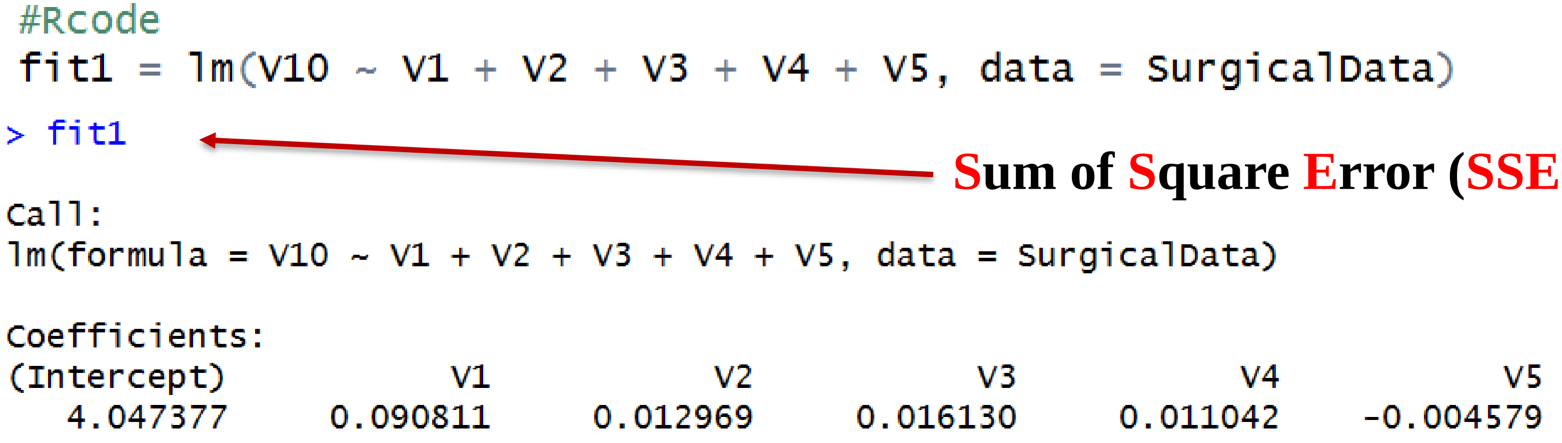
例子：Surgical Unit

V1	Blood clotting （凝血）score
V2	Prognostic index（预后指数）
V3	Enzyme function test score（酶功能测试分数）
V4	Liver function test score（肝功能测试分数）
V5	Age （年龄）
V10 (Y)	Survival time（存活时间） (log-form)

我们现在用**线性回归模型**来拟合这个数据

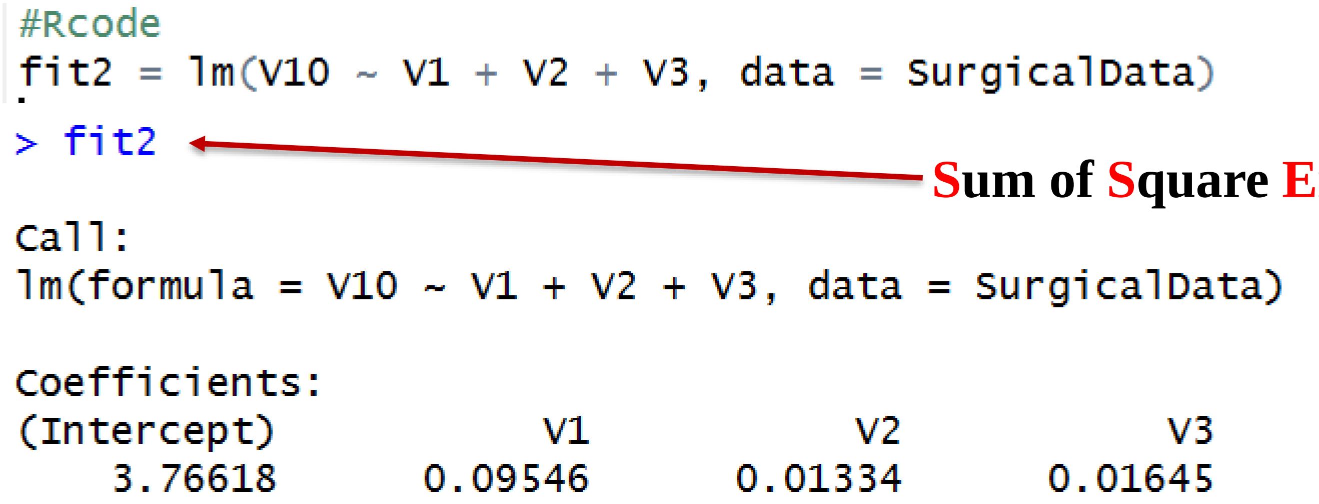
1. 模型1：我们用V10对变量V1,V2,V3,V4,V5做回归
2. 模型2：我们用V10对V1,V2,V3做回归：

例子：Surgical Unit



```
#Rcode
fit1 = lm(V10 ~ V1 + V2 + V3 + V4 + V5, data = SurgicalData)
> fit1

Call:
lm(formula = V10 ~ V1 + V2 + V3 + V4 + V5, data = SurgicalData)

Coefficients:
(Intercept)           V1           V2           V3           V4           V5  
   4.047377     0.090811     0.012969     0.016130     0.011042    -0.004579  
```

Sum of Square Error (SSE):2.9575



```
#Rcode
fit2 = lm(V10 ~ V1 + V2 + V3, data = SurgicalData)
> fit2

Call:
lm(formula = V10 ~ V1 + V2 + V3, data = SurgicalData)

Coefficients:
(Intercept)           V1           V2           V3  
    3.76618      0.09546      0.01334      0.01645  
```

Sum of Square Error (SSE): 3.1085

例子：Surgical Unit

比较以上两个回归模型：

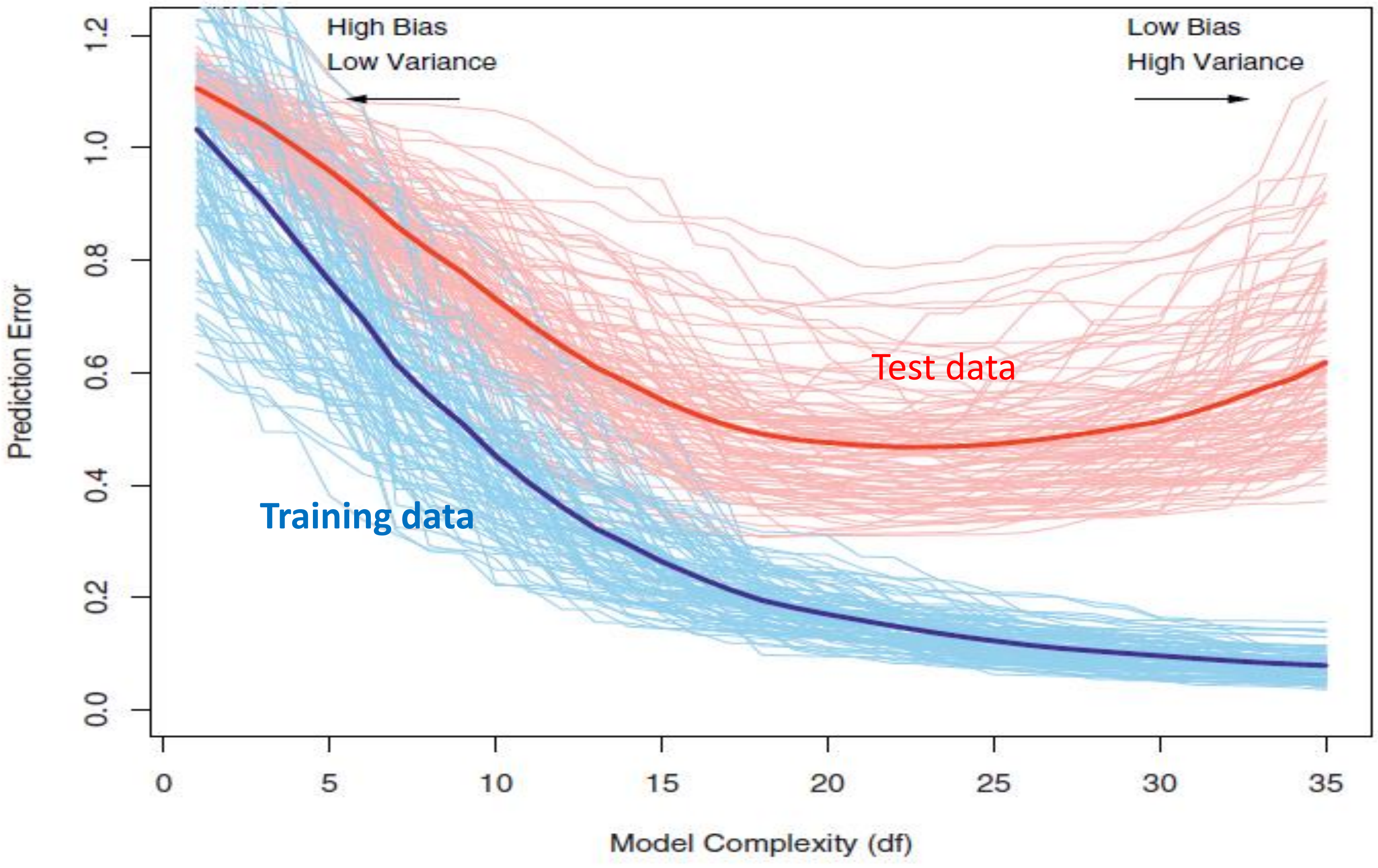
	模型 1	模型 2
预测变量个数	5	3
SSE	2.9575	3.1085

第二个模型更简单，但是它的SSE并没有增加太多（仅增加了$\frac{3.1085-2.9575}{2.9575}$=5.1%），一定程度上反应了变量V4和V5的大部分信息已经包含在前三个变量中。

对所有变量进行回归并不一定是最好的，虽然增加变量会提高拟合效果，但会存在信息冗余，甚至造成过拟合的现象。

在这些情况下，我们可以通过**选择一部分重要的变量做回归**从而获得更稳定的结果。

Relationship between model complexity and prediction error



https://ml.berkeley.edu/blog/2017/07/13/tutorial-4/

- **为什么一个简单的模型有时候比复杂的模型有更好的效果？**
- 假设我们有一个模型$Y = f(X) + \epsilon$, $E(\epsilon|X) = 0$, $var(\epsilon|X) = \sigma_\epsilon^2$，那么对于一个新的观测点$x_0$的估计偏差可做如下分解：

$$Err(x_0) = E\left[\left(Y - \hat{f}(X)\right)^2 | X = x_0\right] = E\left[\left(f(x_0) + \epsilon - \hat{f}(x_0)\right)^2\right]$$
$$= \sigma_\epsilon^2 + \left[E\hat{f}(x_0) - f(x_0)\right]^2 + E\left[\hat{f}(x_0) - E\hat{f}(x_0)\right]^2$$
$$= \boldsymbol{\sigma_\epsilon^2 + Bias^2\left(\hat{f}(x_0)\right) + Var\left(\hat{f}(x_0)\right)}$$

随着模型复杂度的增加，Bias降低，但var增加，导致prediction error增加

如何定义最好的模型？

定义一个评价准则，在备选模型中选择最好的一个。

评价准则的一般形式如下：

$$\boxed{\frac{SSE_p}{n}} + \boxed{\lambda(p)}$$

训练误差　惩罚项

其中: SSE_p是含p个自变量的模型的残差平方和；$\lambda(p)$表示关于p的惩罚项函数。

一些常用的评价准则：

Adjusted R^2, C_p, MSE, $PRESS$, AIC and BIC

If there are too many predictive variables, …

Stepwise methods

- Forward-stepwise:从空模型开始，逐个添加最能提高拟合效果的预测变量。
- Backward-stepwise: 从全模型开始，逐个删除对拟合影响最小的预测量。
- Stepwise-regression: 结合前两种方法，在每一步分别进行向前和向后回归。

R functions: step, stepAIC (MASS)

Shrinkage Method

最小二乘估计量满足：

$$\hat{\beta}_{(ols)} = \underset{\beta}{argmin}(Y - X\beta)^T(Y - X\beta)$$

Shrinkage method在平方损失函数中添加惩罚项：

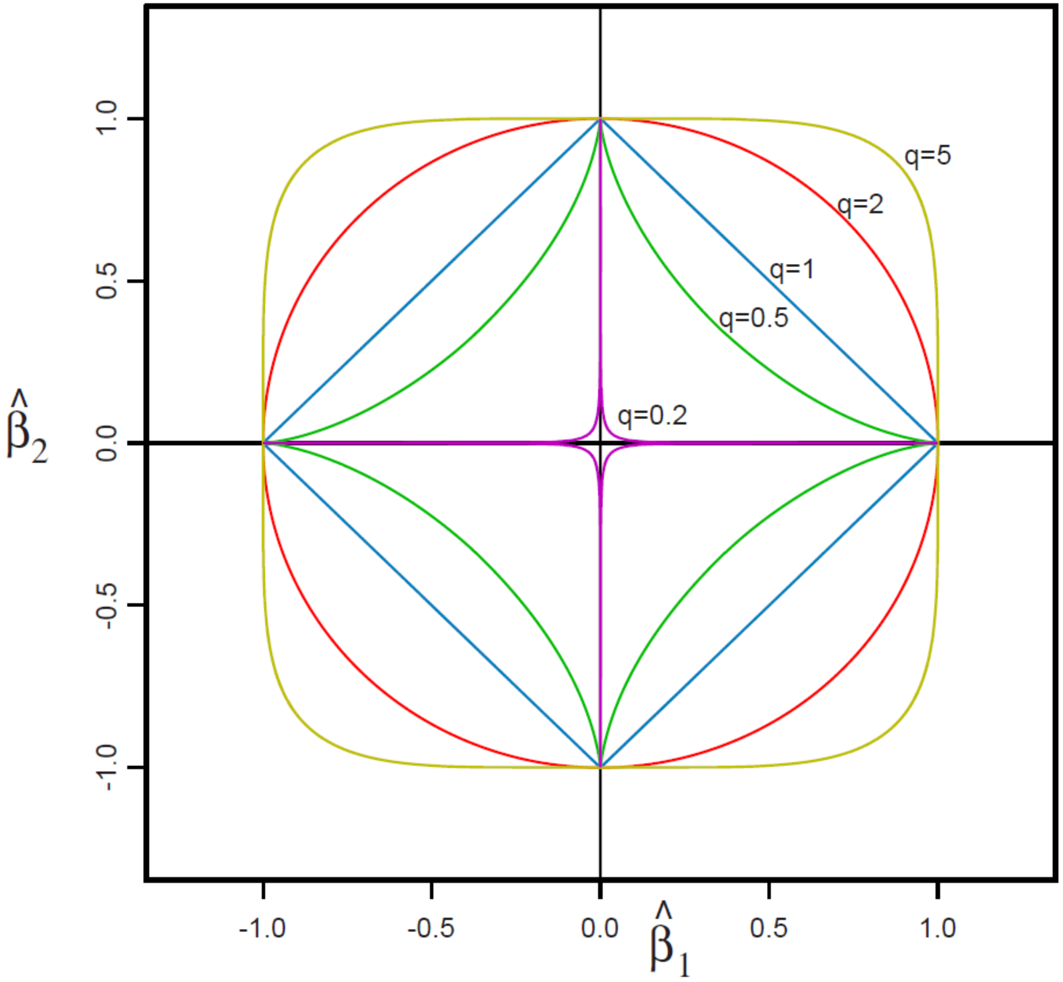
$$\phi(\beta) = (Y - X\beta)^T(Y - X\beta) + \boldsymbol{\lambda p(\beta)}$$

其中，λ 是正则化参数，$p(\cdot)$是惩罚函数。我们需要找到使得 $\phi(\beta)$ 最小时的$\hat{\beta}$。

1. 当$\lambda = 0$, $\hat{\beta}$ 是最小二乘估计
2. 当$\lambda \neq 0$时，
   - $p(\beta) = ||\beta||_2^2$,该方法是岭回归
   - $p(\beta) = ||\beta||_1$,该方法是lasso回归
   - $p(\beta) = ||\beta||_0$, 该方法是子集选择

Shrinkage Methods

对于 $p(\beta) = ||\beta||_q^q$ 中不同的q有下图（β为二维）：



压缩估计方法—岭回归(Ridge regression, L_2)

$$\hat{\beta}_{ridge} = (X^T X + \lambda I)^{-1} X^T Y$$

which is equivalent to

$$\hat{\beta}_{ridge} = \arg min_{\beta} \{(Y - X\beta)^T (Y - X\beta) + \lambda p(\beta)\}, \lambda > 0$$

with the penalty function $p(\beta) = \left|\left|\beta\right|\right|_2^2 = \beta^T \beta$, and

$$\hat{\beta}_{ridge} = \arg min_{\beta} \{(Y - X\beta)^T (Y - X\beta)\}$$

Subject to $\sum_{i=1}^{p} \beta_j^2 \leq t$

压缩估计方法——Lasso (L_1)

$$p(\beta) = ||\beta||_1 = \sum_{i=1}^{p} |\beta_i|$$

The Lasso estimator is defined by

$\hat{\beta}_{Lasso} \ = \arg min_{\beta} \{(Y - X\beta)^T (Y - X\beta) + \lambda \sum_{i=1}^{p} |\beta_i|\}$, where λ>0.

Equivalently,

$$\hat{\beta}_{ridge} \ = \arg min_{\beta} \{(Y - X\beta)^T (Y - X\beta)\}$$

Subject to$\sum_{i=1}^{p} |\beta_i| \leq t$

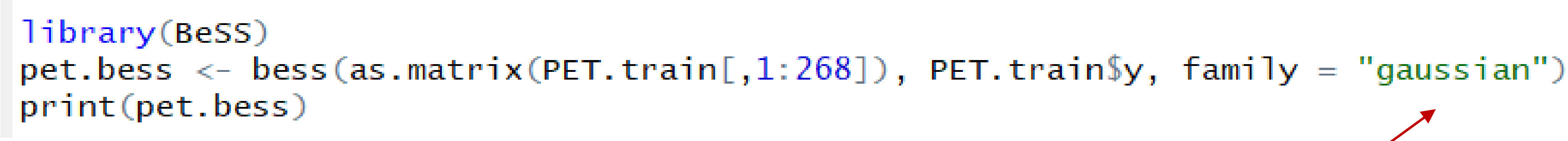
Algorithm: LARS, coordinate descent

R package: glmnet, ranger

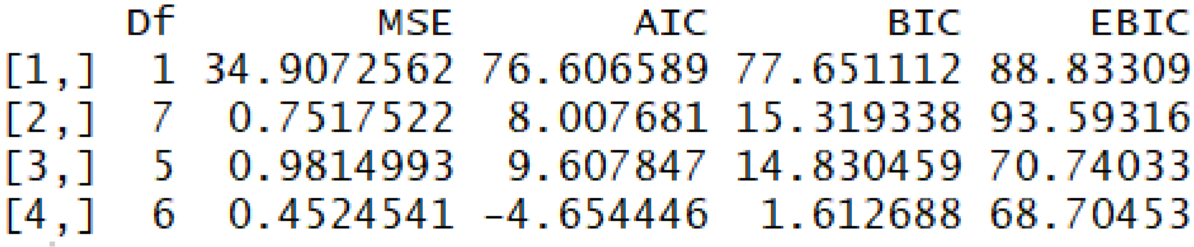
压缩估计方法——L_0

考虑惩罚函数$p(\beta) = ||\beta||_0 = \sum_{i=1}^{p} I_{[\beta_i \neq 0]}$

对应最优子集选择方法，可以用R语言中BeSS软件包求解



```
library(BeSS)
pet.bess <- bess(as.matrix(PET.train[,1:268]), PET.train$y, family = "gaussian")
print(pet.bess)
```



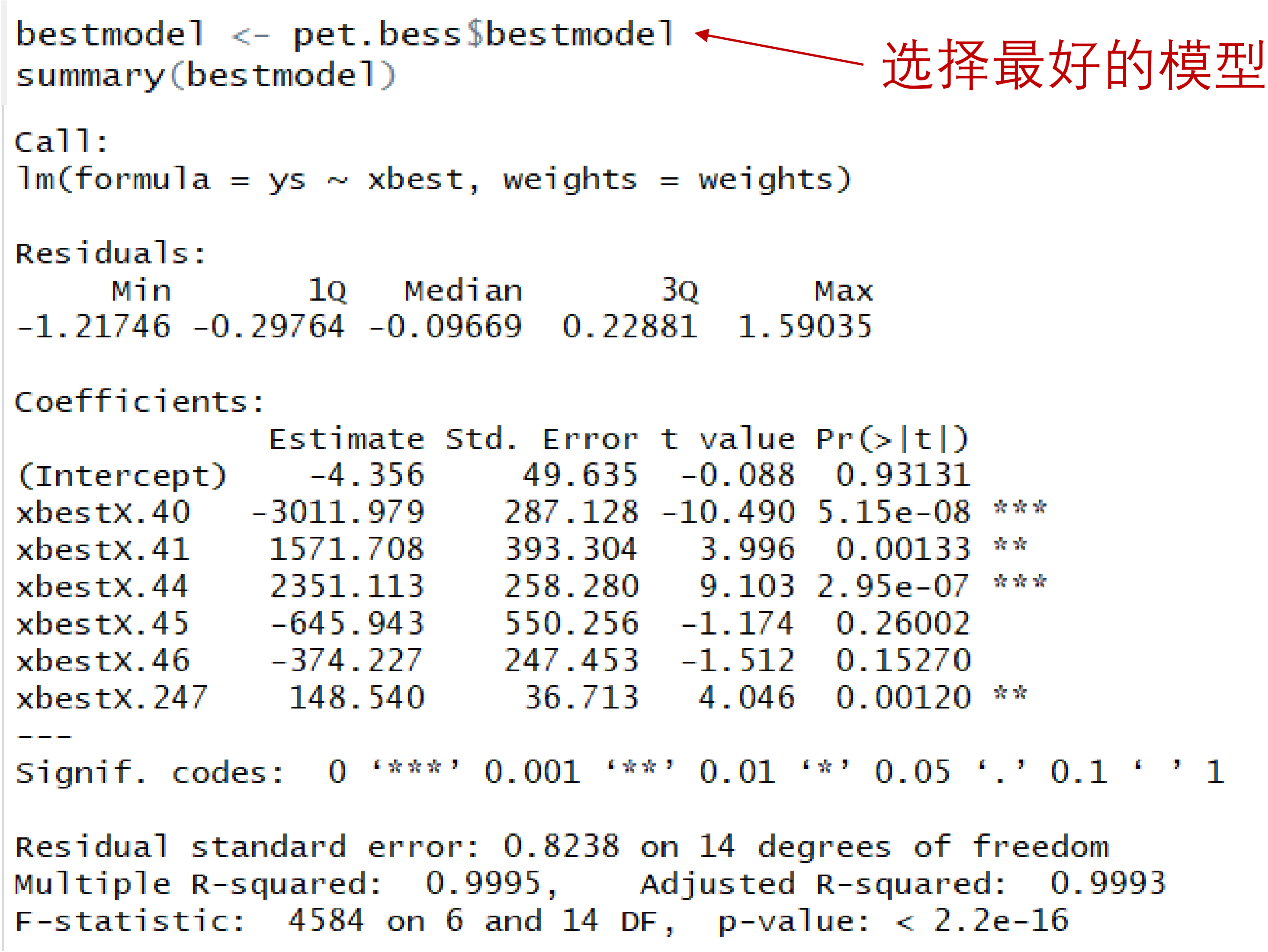
```
     Df        MSE       AIC       BIC     EBIC
[1,]  1 34.9072562 76.606589 77.651112 88.83309
[2,]  7  0.7517522  8.007681 15.319338 93.59316
[3,]  5  0.9814993  9.607847 14.830459 70.74033
[4,]  6  0.4524541 -4.654446  1.612688 68.70453
```

表示用线性回归

Df： 表示选进模型的变量个数

也可以直接用aic(pet.bess), bic(pet.bess), ebic(pet.bess)
来获得候选模型的评价值

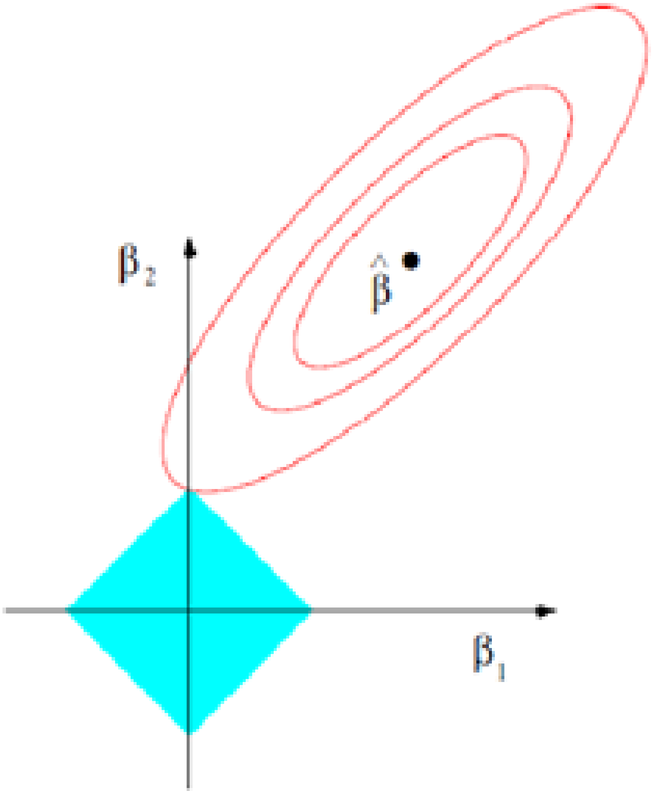
压缩估计方法——L_0



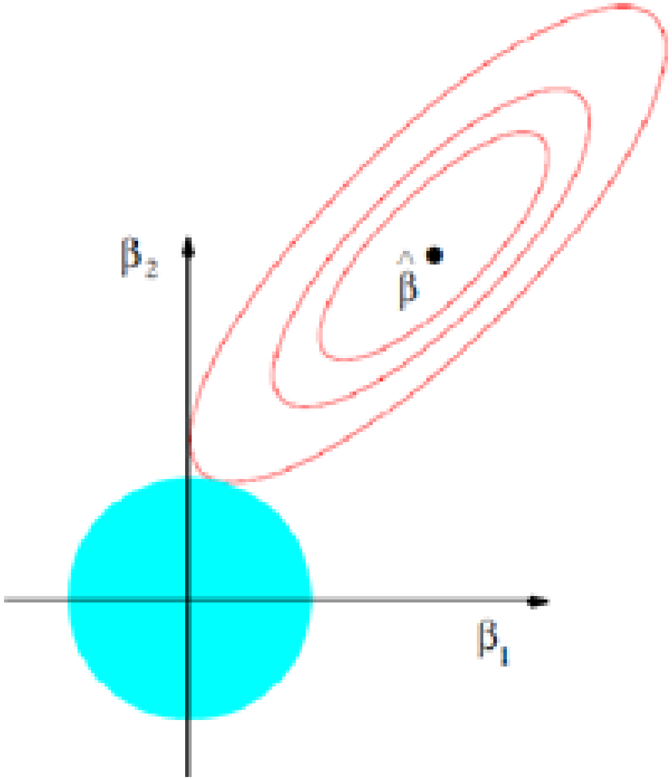
```
bestmodel <- pet.bess$bestmodel
summary(bestmodel)

Call:
lm(formula = ys ~ xbest, weights = weights)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.21746 -0.29764 -0.09669  0.22881  1.59035 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -4.356     49.635  -0.088  0.93131    
xbestX.40   -3011.979    287.128 -10.490 5.15e-08 ***
xbestX.41    1571.708    393.304   3.996  0.00133 ** 
xbestX.44    2351.113    258.280   9.103 2.95e-07 ***
xbestX.45    -645.943    550.256  -1.174  0.26002    
xbestX.46    -374.227    247.453  -1.512  0.15270    
xbestX.247    148.540     36.713   4.046  0.00120 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8238 on 14 degrees of freedom
Multiple R-squared:  0.9995,	Adjusted R-squared:  0.9993 
F-statistic:  4584 on 6 and 14 DF,  p-value: < 2.2e-16
```

选择最好的模型

压缩估计方法比较：岭回归 VS Lasso （二维例子）



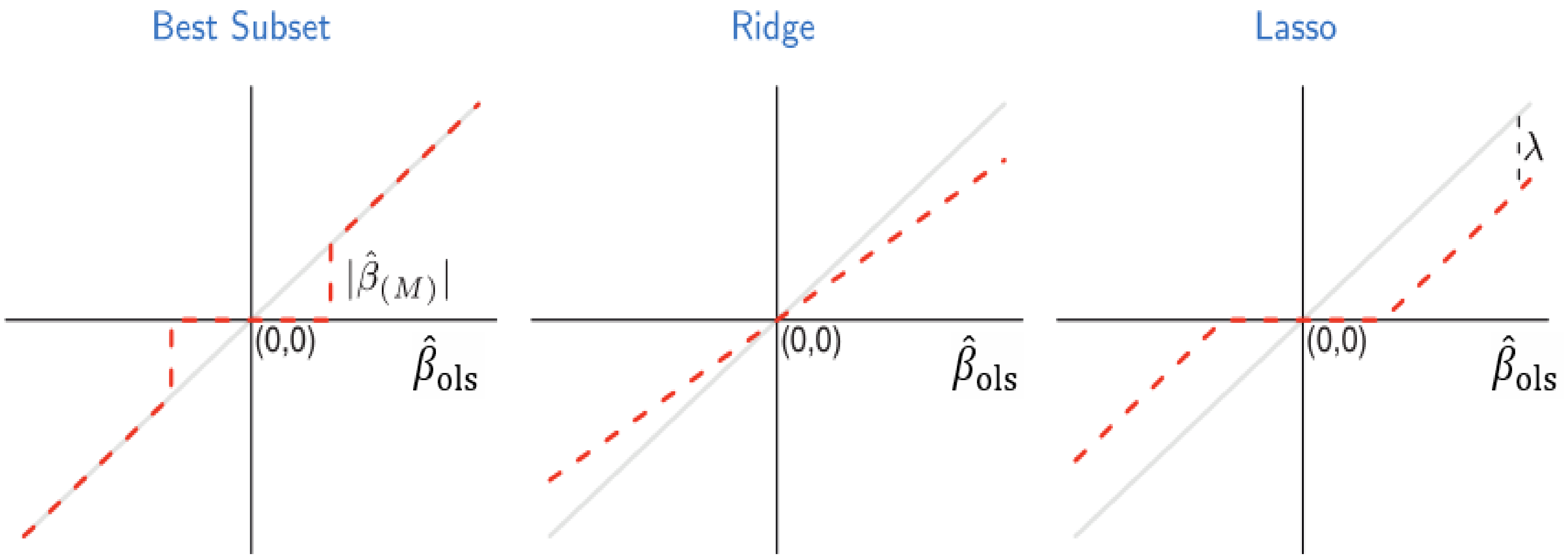
Lasso回归



岭回归

压缩估计方法比较：

Best Subset VS 岭回归 VS Lasso



压缩估计方法——思考

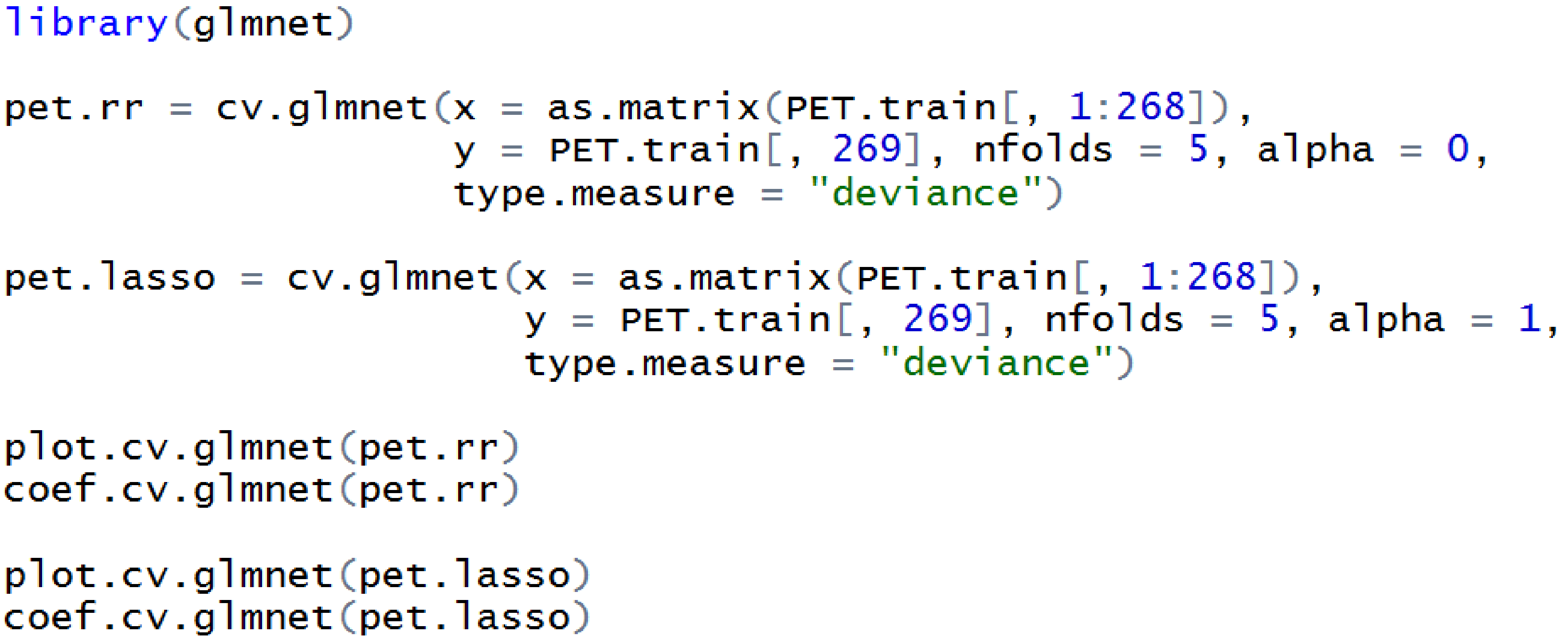
问题： 在岭回归和lasso回归中, 我们如何选择正则化参数λ?

- 给定 λ, 通过交叉验证(cross-validation) 的方法，估计模型的泛化误差;
- 对不同的λ, 我们可以得到它们对应的泛化误差估计结果, 选择使得泛化误差达到最小的λ。

R 应用：

应用glmnet 包中的cv.glmnet, plot.cv.glmnet, coef.cv.glmnet

压缩估计方法——R实现



```r
library(glmnet)

pet.rr = cv.glmnet(x = as.matrix(PET.train[, 1:268]),
                   y = PET.train[, 269], nfolds = 5, alpha = 0,
                   type.measure = "deviance")

pet.lasso = cv.glmnet(x = as.matrix(PET.train[, 1:268]),
                      y = PET.train[, 269], nfolds = 5, alpha = 1,
                      type.measure = "deviance")

plot.cv.glmnet(pet.rr)
coef.cv.glmnet(pet.rr)

plot.cv.glmnet(pet.lasso)
coef.cv.glmnet(pet.lasso)
```

Note:

alpha=0 means ridge regression,

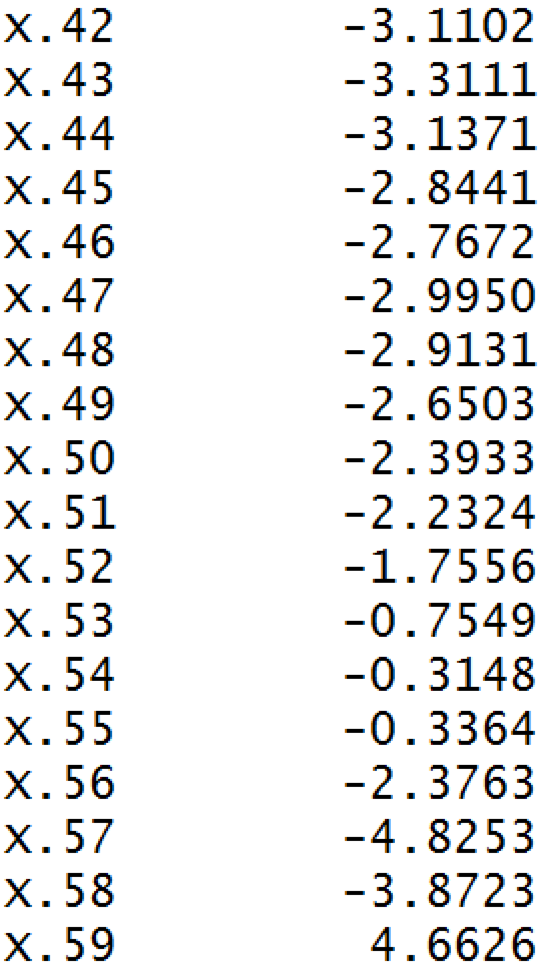
alpha=1 means lasso regression,

nfolds=5 means using 5-fold cross-validation

压缩估计方法——R实现

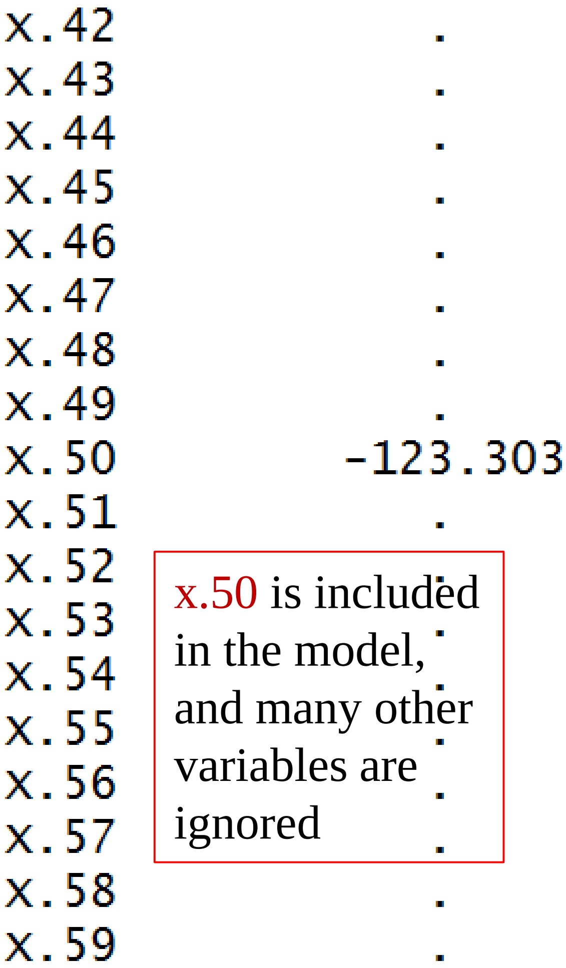
部分回归系数:

岭回归



```
x.42          -3.1102
x.43          -3.3111
x.44          -3.1371
x.45          -2.8441
x.46          -2.7672
x.47          -2.9950
x.48          -2.9131
x.49          -2.6503
x.50          -2.3933
x.51          -2.2324
x.52          -1.7556
x.53          -0.7549
x.54          -0.3148
x.55          -0.3364
x.56          -2.3763
x.57          -4.8253
x.58          -3.8723
x.59           4.6626
```

Lasso回归

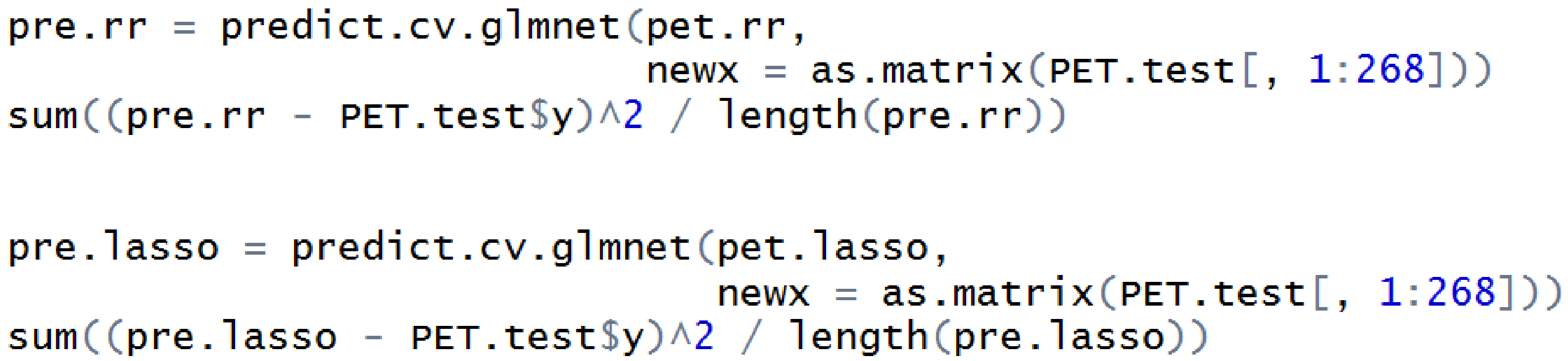


```
x.42              .
x.43              .
x.44              .
x.45              .
x.46              .
x.47              .
x.48              .
x.49              .
x.50       -123.303
x.51              .
x.52              .
x.53              .
x.54              .
x.55              .
x.56              .
x.57              .
x.58              .
x.59              .
```

x.50 is included in the model, and many other variables are ignored

压缩估计方法——R实现

对新数据进行预测并计算均方误差



```
pre.rr = predict.cv.glmnet(pet.rr,
                           newx = as.matrix(PET.test[, 1:268]))
sum((pre.rr - PET.test$y)^2 / length(pre.rr))


pre.lasso = predict.cv.glmnet(pet.lasso,
                              newx = as.matrix(PET.test[, 1:268]))
sum((pre.lasso - PET.test$y)^2 / length(pre.lasso))
```

$$MSE_{ridge} = 2.33,$$
$$MSE_{lasso} = 0.234$$

Model Validation

Collection of New Data to Check Model

It is the best means of model validation.

- Re-estimate the chosen model using the New data. The estimated regression coefficients and various characteristics of the fitted model are then compared for consistency to those of the regression model obtained on the earlier data.
- Check the **predictive capability of the selected model**, that is

$$MSPR = \frac{\sum_{i=1}^{n^*}(Y_i - \hat{Y}_i)^2}{n^*}$$

- If $MSPR$(mean squared prediction error) is fairly close to MSE based on the regression fit to the model-building data set, it gives an appropriate indication of the predictive ability of the model.

Model Validation

Data Splitting

- When the data set is large enough, we can split the data into model-building set (**training data**) and validation set (**test data**). Model-building set is used to develop the model and the validation set is used to evaluate the reasonableness and predictive ability of the selected model.
- Cross-validation (CV)

Correlation coefficient

1. Pearson’s *r* in linear regression
2. Spearman’s *rho*
3. Kendall’s *tau*

Correlation coefficient

- The sample (Pearson) correlation coefficient (*r*) is defined by

$$r = L_{xy} \,/\, \sqrt{L_{xx} L_{yy}}$$

where $L_{xy} = \sum(x_i - \bar{x})(y_i - \bar{y})$, $L_{xx} = \sum(x_i - \bar{x})^2$, $L_{yy} = \sum(y_i - \bar{y})^2$

- Also

$$r = \frac{s_{xy}}{\sqrt{s_{xx} s_{yy}}} = \frac{\text{sample covariance between x and y}}{(\text{sample sd of } x)(\text{sample sd of } y)}$$

where $s_{xy} = \mathrm{L_{xy}}/(\mathrm{n} - 1)$ is the sample covariance

- R: `> cor(x,y)`

Relationship Between b and r

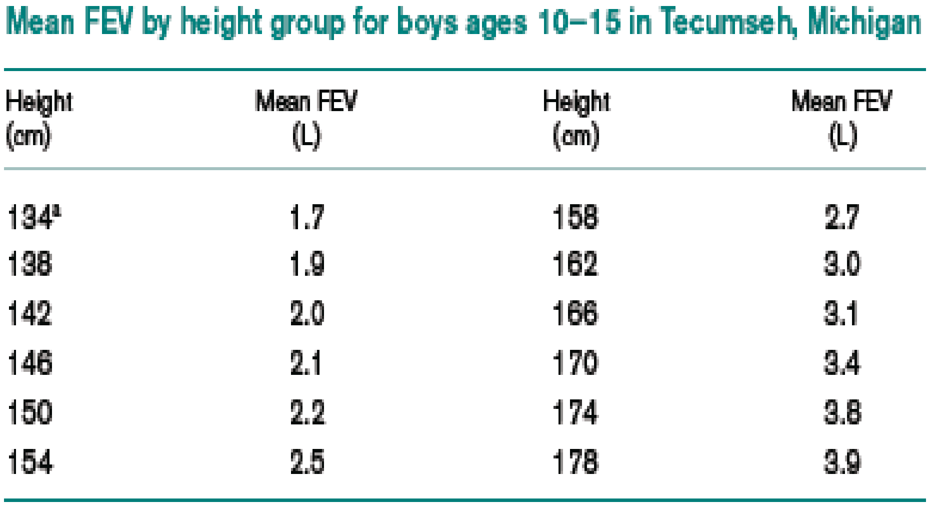
- The sample regression coefficient (b): $b = L_{xy}/L_{xx}$
- The sample correlation coefficient (r): $r = L_{xy} / \sqrt{L_{xx}L_{yy}}$

$$b = \sqrt{\frac{L_{yy}}{L_{xx}}} \times r = \frac{s_y}{s_x} \times r$$

- Thus, b can be interpreted as a rescaled version of r, where the scale factor is the ratio of the sd of *y* to that of *x*.

- r does not change if we change units of measurement, while b depends on the unit of measurement.

R codes



Mean FEV by height group for boys ages 10–15 in Tecumseh, Michigan

Height (cm)	Mean FEV (L)	Height (cm)	Mean FEV (L)
134[a]	1.7	158	2.7
138	1.9	162	3.0
142	2.0	166	3.1
146	2.1	170	3.4
150	2.2	174	3.8
154	2.5	178	3.9

```
> height <- seq(134,178,4)
> fev <- c(1.7,1.9,2.0,2.1,2.2,2.5,2.7,3.0,3.1,3.4,3.8,3.9)
> print( r <- cor(fev, height)) # Pearson coefficient

[1] 0.988159
> coef(fev.lm <- lm(fev ~ height))[[2]]
[1] 0.05131119
> sd(fev)/sd(height)*r # relation between r and b
[1] 0.05131119
```

b vs r

- When should the regression coefficient or the correlation coefficient be used?

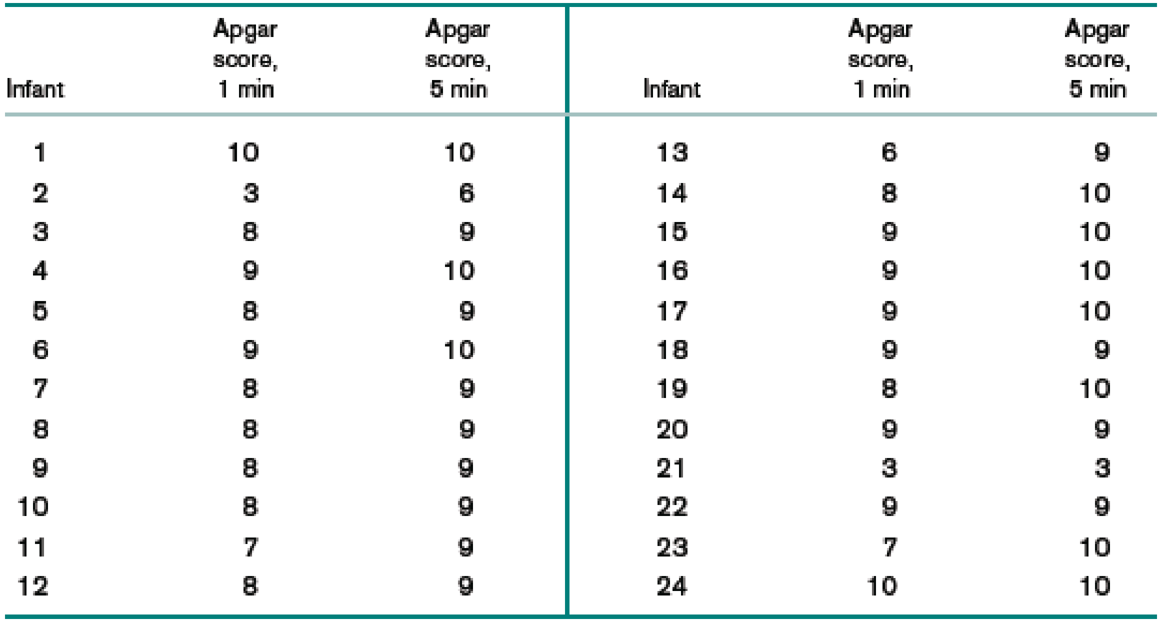
- predict one variable from another => regression coefficient
  - How does Y change with one unit of X? ⇒ (b)

- describe the linear relationship between two variables

  => correlation coefficient
  - How does Y change (in standard deviation) with one standard deviation of X? ⇒ (bs_x/s_y)

Example
Obstetrics

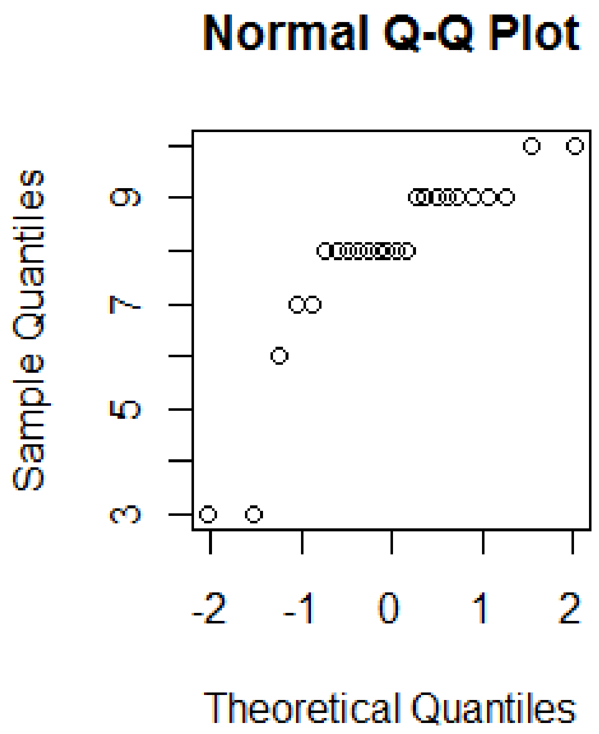
- The Apgar score is obtained by summing five components, each of which is rated as 0, 1, or 2 and represents different aspects of the condition of an infant at birth.
- as a measure of the physical condition of an infant at 1 and 5 minutes after birth .

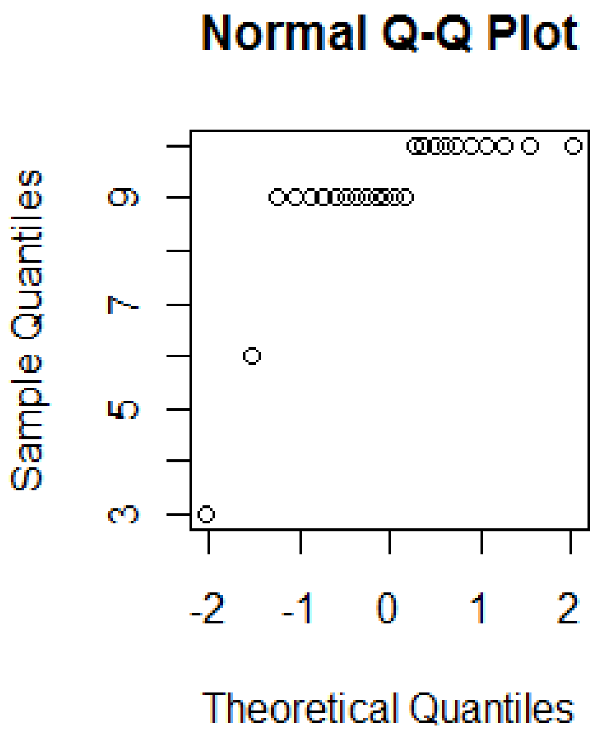


Infant	Apgar score, 1 min	Apgar score, 5 min	Infant	Apgar score, 1 min	Apgar score, 5 min
1	10	10	13	6	9
2	3	6	14	8	10
3	8	9	15	9	10
4	9	10	16	9	10
5	8	9	17	9	10
6	9	10	18	9	9
7	8	9	19	8	10
8	8	9	20	9	9
9	8	9	21	3	3
10	8	9	22	9	9
11	7	9	23	7	10
12	8	9	24	10	10

Contd.

- Pearson correlation shouldn’t be used since the scores didn’t follow a normal distribution.





Rank correlation

- The spearman rank correlation coefficient

$$r_s = \frac{L_{xy}}{\sqrt{L_{xx} \times L_{yy}}}$$

- where the L's are computed from the ranks rather than from the actual scores

- R codes: `cor(x,y, method = "spearman")`

One sample t test

- To test $H_0: \rho = 0$ vs. $H_1: \rho \neq 0$

- The test statistic is

$$t_s = \frac{r_s\sqrt{n-2}}{\sqrt{1-{r_s}^2}} \sim t_{n-2} \text{ under } H_0$$

- The t test is similar to those for Pearson correlation coefficient.

R code

Obstetrics

```
> as1 <- c(10,3,8,9,8,9,8,8,8,8,7,8,
+            6,8,9,9,9,9,8,9,3,9,7,10)
> as5 <- c(10,6,9,10,9,10,9,9,9,9,9,9,
+            9,10,10,10,10,9,10,9,3,9,10,10)
> cor(as1,as5) # the Pearson coefficient
[1] 0.8448476
> cor(as1,as5, method = "s") # the Spearman coefficient
[1] 0.5927144
> cor.test(as1,as5,method="s") # t test

Spearman's rank correlation rho

data: as1 and as5 S = 936.7569, p-value = 0.002272
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
 0.5927144
```

kendall's Tau

- based on number of concordant and discordant pairs

$$\tau = \frac{\#\{\text{concordant pairs}\} - \#\{\text{discordant pairs}\}}{\frac{1}{2}n(n-1)}$$

- Any pair of observations (x_i, y_i) and (x_j, y_j) are said to be concordant if the ranks for both elements agree
  - if both $x_i \ > x_j$ and $y_i \ > y_j$ or if both $x_i \ < x_j$ and $y_i \ < y_j$.
- They are said to be discordant:
  - If $x_i \ > x_j$ and $y_i < y_j$ or if both $x_i \ < x_j$ and $y_i > y_j$

kendall’s Tau

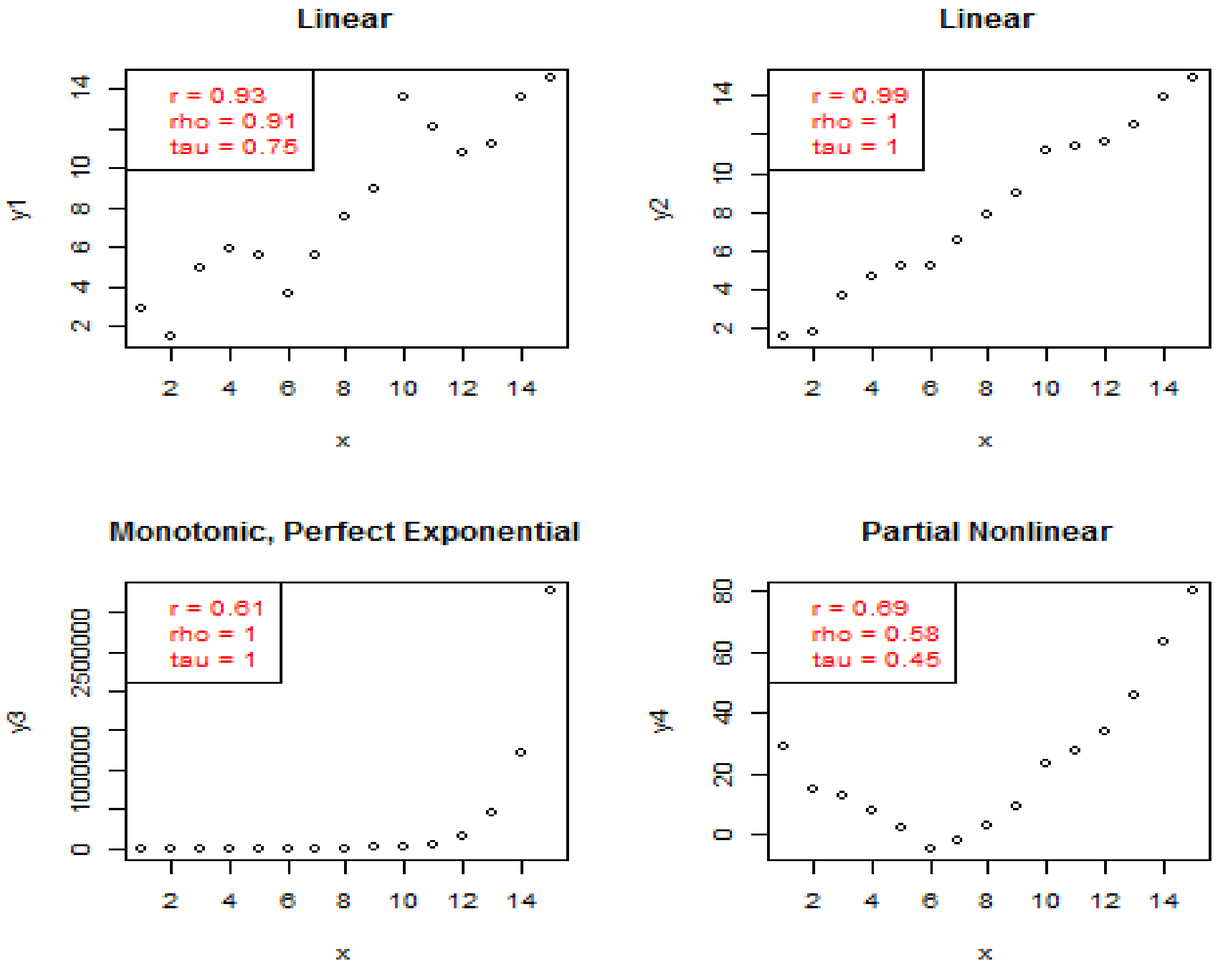
- to measure the degree of correspondence between two rankings and assessing the significance of this correspondence.

- In other words, it measures the strength of association of the cross tabulations.

- Which type of correlation coefficients should you use?
  - first look scatter plot

- nonlinear correlation measurement
  - MIC, distance correlation, …

Comparison using four types of curve


Linear
r = 0.93
rho = 0.91
tau = 0.75
y1
x
Linear
r = 0.99
rho = 1
tau = 1
y2
x
Monotonic, Perfect Exponential
r = 0.61
rho = 1
tau = 1
y3
x
Partial Nonlinear
r = 0.69
rho = 0.58
tau = 0.45
y4
x