B05

統計模型與迴歸分析

吳漢銘

國立政治大學 統計學系

http://www.hmwu.idv.tw

本章大綱與學習目標

- 統計模型配適
  - 解釋變數(X)，反應變數(Y)，Model Formula
- 簡單線性迴歸 (Simple Linear Regression)
  - 最小平方法、ANOVA Table、信賴區間
- Extract Information from Model Objects.
- 統計模型檢測(Model Checking in R)
  - Residual Plots， Normal QQ-plot，A Scale-Location Plot， Cook's Distance vs Row Labels，Residuals vs Leverages，Cook's Distance vs Leverage.
- 逐步迴歸變數篩選
- 範例: Linear Regression, Logistic Regression
- 共線性 (Collinearity)
  - 變異數膨脹因子(The Variance Inflation Factors)

統計模型配適 (Statistical Modeling)

四個問題:

1. Which of your variables is the **response variable** (反應變數, Y)?

2. Which are the **explanatory variable** (解釋變數, $\boldsymbol{X}$)?

3. Are the explanatory variables **continuous** (連續) or **categorical** (類別), or a **mixture** (混合) of both?

4. What kind of response variable do you have: **continuous** measurement, a **count**, a **proportion**, a **time** at death, or **category**?

配適統計模型的目的

- To determine the values of the **parameters** in a specific model that lead to the best fit of the model to the data.

解釋變數, X

The Explanatory Variable (X)

- **All x's are continuous**: Regression

例如:

$$\text{Simple linear regression: } y = \beta_0 + \beta_1 x + \epsilon$$
$$\text{Multiple linear regression: } y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$
$$\text{Polynomial regression: } y = \beta_0 + \beta_1 x + +\beta_2 x^2 + \cdots + \beta_d x^d + \epsilon$$
$$\text{Nonlinear regression: } y = \theta_0 + \theta_1 (1 - e^{\theta_2 x}) + \epsilon$$

- **All x's are categorical**: Analysis of Variance (ANOVA, 變異數分析)

例如:

$$y_{ij} = \mu + \alpha_j + \epsilon_{ij}$$
$$\boldsymbol{y} = \boldsymbol{A\theta} + \boldsymbol{\epsilon}$$

- **x's are both continuous and categorical:** Analysis of Covariance (ANCOVA)

例如:

$$y = \beta_0 + \beta_1 x + \theta z + \epsilon, \;\; z = \{0, 1\}$$

反應變數, Y (1)

The Response Variable (y)

- **Continuous**: Normal Regression, ANOVA or ANCOVA
- **Binary**: Binary Logistic Analysis

例如:

$$P(y_i = 0) = 1 - \pi_i, \;\; P(y_i = 1) = \pi_i$$

$$\text{Logistic link function: } g(\pi) = \log(\frac{\pi}{1-\pi})$$

$$\text{Logistic regression: } \log(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

- **Ordinal**: proportional-odds model

例如:

$$\gamma_j(\mathbf{x}) = P(Y \leq j|\mathbf{x}), \quad \log\left(\frac{\gamma_j(\mathbf{x})}{1-\gamma_j(\mathbf{x})}\right) = \boldsymbol{\beta}^T\mathbf{x}$$

反應變數, Y (2)

- **Count**: Log-Linear Models

例如:

$$Y \sim Poisson(\mu),\ \mu = E(Y), \quad \log \mu = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

- **Time at death**: Survival Analysis

- T: survival time with a density function $f(t)$.
- $1 - F(t)$: survival function (i.e., $F(t) = \int_{-\infty}^{t} f(s)\, ds$).
- $h(t) = \dfrac{f(t)}{1 - F(t)}$: hazard function.
- $h(t)\delta t$: the probability of dying in the next small interval δt given survival to time t
- Proportional-hazards model: $h(t; \mathbf{x}) = \lambda(t) \exp(\beta^T \mathbf{x})$

模式寫法 (Model Formulae in R)

- The structure of the model: `response.variable ~ explanatory.variables`
  - Example: `fm <- formula(y ~ x)`
  - Example: `lm(fm)`, `lm(y ~ x)`; `aov(y ~ x)`; `glm(y ~ x)`
- ~: "is modelled as a function of"
  - Example: `lm(y ~ x)`
- + : inclusion of an explanatory variable in the model (not addition);
  - Example: `lm(y ~ x1 + x2)`
- – : deletion of an explanatory variable from the model (not subtraction);
  - Example: `lm(y ~ x1 - 1)`
- * : inclusion of explanatory variables and interactions (not multiplication);
  - Example: `lm(y ~ x1 * x2)`
- /: nesting of explanatory variables in the model (not division);
  - Example: `lm(y ~ x1 / x2)` *# x1因子的各分類下，再細分出x2因子的分類*

Examples

```
> y <- rnorm(50)
> x1 <- rnorm(50)
> x2 <- rnorm(50)
> x3 <- rnorm(50)
> lm(y ~ x1 + x2)
Call:
lm(formula = y ~ x1 + x2)

Coefficients:
(Intercept)           x1
   -0.13024      0.05576
         x2
    0.02093
> lm(y ~ x1 - 1)
Call:
lm(formula = y ~ x1 - 1)

Coefficients:
     x1
0.03885
> lm(y ~ x1 * x2)
Call:
lm(formula = y ~ x1 * x2)

Coefficients:
(Intercept)           x1
   -0.05122     -0.03178
         x2        x1:x2
    0.05614      0.26850
```

```
> y <- rnorm(50)
> school <- as.factor(sample(c("a", "b", "c"), 50, replace=T))
> gender <- as.factor(sample(c("f", "m"), 50, replace=T))
> table(school, gender)
      gender
school  f  m
     a 10 12
     b  4  9
     c  6  9
> lm(y ~ school / gender)
Call:
lm(formula = y ~ school/gender)

Coefficients:
    (Intercept)          schoolb          schoolc
         0.1198           0.1504           1.0190
schoola:genderm  schoolb:genderm  schoolc:genderm
         0.1192          -0.0647          -1.3472

> lm(y ~ gender / school)

Call:
lm(formula = y ~ gender/school)

Coefficients:
    (Intercept)          genderm  genderf:schoolb
         0.1198           0.1192           0.1504
genderm:schoolb  genderf:schoolc  genderm:schoolc
        -0.0335           1.0190          -0.4475
```

模式寫法 (Model Formulae in R)

- |: indicates conditioning (not 'or'), so that y ~ x | z is read as 'y as a function of x given z". Example: `lm(y ~ x1 | x2)`
- " : " : a colon denotes an interaction
  - `A:B` means the two-way interaction between `A` and `B`
  - `N:P:K:Mg` means the four-way interaction between `N, P, K` and `Mg`.

```
> lm(y ~ x1 | x2)

Call:
lm(formula = y ~ x1 | x2)

Coefficients:
(Intercept)  x1 | x2TRUE
    -0.1216           NA
```

```
> lm(y ~ x1:x2:x3)

Call:
lm(formula = y ~ x1:x2:x3)

Coefficients:
(Intercept)     x1:x2:x3
   -0.08602     -0.20145
```

```
> ##Create a formula for a model with a large number of variables:
> xnam <- paste("x", 1:25, sep="")
> (fmla <- as.formula(paste("y ~ ", paste(xnam, collapse= "+"))))
y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 +
    x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x20 + x21 +
    x22 + x23 + x24 + x25
```

模式寫法 (Model Formulae in R)

- `A*B*C` is the same as `A+B+C+A:B+A:C+B:C+A:B:C`
- `A/B/C` is the same as `A+B%in%A+C%in%B%in%A`
- `(A+B+C)^3` is the same as `A*B*C`
- `(A+B+C)^2` is the same as `A*B*C - A:B:C`

```
> y <- rnorm(50)
> A <- rnorm(50)
> B <- rnorm(50)
> C <- rnorm(50)
```

```
> lm(y ~ A*B*C)

Call:
lm(formula = y ~ A * B * C)

Coefficients:
(Intercept)            A            B
    0.20776      -0.04336      0.01105
          C          A:B          A:C
   -0.06969      0.14857     -0.02269
        B:C        A:B:C
   -0.06689      0.08850

> lm(y ~ A/B/C)

Call:
lm(formula = y ~ A/B/C)

Coefficients:
(Intercept)            A          A:B
    0.21586      -0.06219      0.12840
      A:B:C
    0.07229
```

```
> lm(y ~ (A+B+C)^3)

Call:
lm(formula = y ~ (A + B + C)^3)

Coefficients:
(Intercept)            A            B            C
    0.20776      -0.04336      0.01105     -0.06969
        A:B          A:C          B:C        A:B:C
    0.14857      -0.02269     -0.06689      0.08850

> lm(y ~ (A+B+C)^2)

Call:
lm(formula = y ~ (A + B + C)^2)

Coefficients:
(Intercept)            A            B            C
    0.21990      -0.03953      0.02210     -0.05622
        A:B          A:C          B:C
    0.15181      -0.05379     -0.03787
```

Model Formula 例子1

Table 9.3. Examples of R model formulae. In a model formula, the function I case i) stands for 'as is' and is used for generating sequences I(1:10) or calculating quadratic terms I(x^2).

Model	Model formula	Comments
Null	y ~ 1	1 is the intercept in regression models, but here it is the overall mean *y*
Regression	y ~ x	*x* is a continuous explanatory variable
Regression through origin	y ~ x-1	Do not fit an intercept `y ~ 0 + x`
One-way ANOVA	y ~ sex	sex is a two-level categorical variable
One-way ANOVA	y ~ sex-1	as above, but do not fit an intercept (gives two means rather than a mean and a difference)
Two-way ANOVA	y ~ sex + genotype	genotype is a four-level categorical variable
Factorial ANOVA	y ~ N * P * K	*N*, *P* and *K* are two-level factors to be fitted along with all their interactions

Source: Crawley, M. J. , 2007, The R Book, Wiley.

Model Formula 例子2

Table 9.3. (Continued)

Model	Model formula	Comments
Three-way ANOVA	y ~ N*P*K – N:P:K	As above, but don't fit the three-way interaction
Analysis of covariance	y ~ x + sex	A common slope for *y* against *x* but with two intercepts, one for each sex
Analysis of covariance	y ~ x * sex	Two slopes and two intercepts
Nested ANOVA	y ~ a/b/c	Factor *c* nested within factor *b* within factor *a*
Split-plot ANOVA	y ~ a*b*c+Error(a/b/c)	A factorial experiment but with three plot sizes and three different error variances, one for each plot size
Multiple regression	y ~ x + z	Two continuous explanatory variables, flat surface fit
Multiple regression	y ~ x * z	Fit an interaction term as well (x + z + x:z)

Source: Crawley, M. J. , 2007, The R Book, Wiley.

Model Formula 例子 3

Table 9.3. (Continued)

Model	Model formula	Comments
Multiple regression	y ~ x + I(x^2) + z + I(z^2)	Fit a quadratic term for both *x* and *z*
Multiple regression	y <- poly(x,2) + z	Fit a quadratic polynomial for *x* and linear *z*
Multiple regression	y ~ (x + z + w)^2	Fit three variables plus all their interactions up to two-way
Non-parametric model	y ~ s(x) +s(z)	*y* is a function of smoothed *x* and *z* in a generalized additive model
Transformed response and explanatory variables	log(y) ~ I(1/x) + sqrt(z)	All three variables are transformed in the model

the function I case i) stands
for 'as is' and is used for generating sequences I(1:10)
or calculating quadratic terms I(x^2).

Source: Crawley, M. J. , 2007, The R Book, Wiley.

Statistical Models in R

lm	fits a linear model with normal errors and constant variance; generally this is used for regression analysis using continuous explanatory variables.
aov	fits analysis of variance with normal errors, constant variance and the identity link; generally used for categorical explanatory variables or ANCOVA with a mix of categorical and continuous explanatory variables.
glm	fits generalized linear models to data using categorical or continuous explanatory variables, by specifying one of a family of **error structures** (e.g. Poisson for count data or binomial for proportion data) and a particular **link function**.
gam	fits generalized additive models
lme	and lmer fit linear mixed-effects models
nls	fits a non-linear regression model via least squares
nlme	fits a specified non-linear function in a mixed-effects model
loess	fits a local regression model
tree	fits a regression tree model using binary recursive partitioning

Source: Crawley, M. J. , 2007, The R Book, Wiley.

簡單線性迴歸 (Simple Linear Regression)

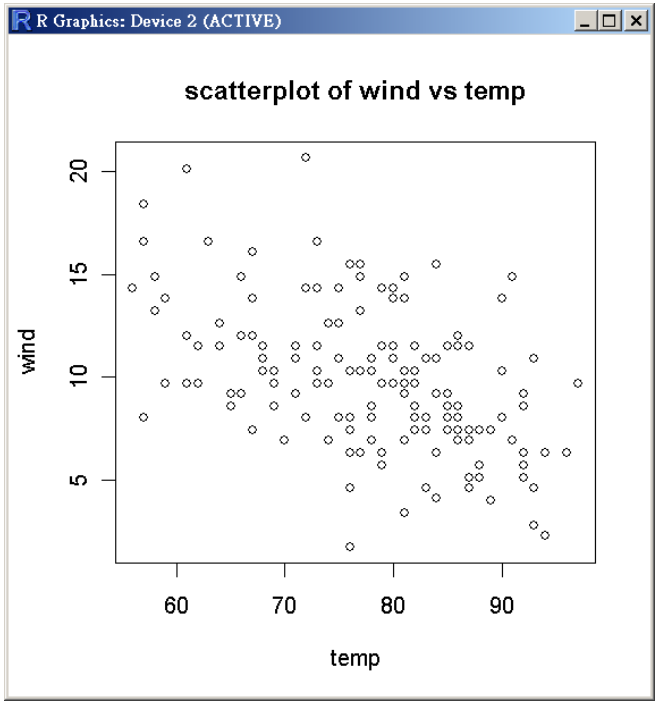
$$y = \beta_0 + \beta_1 x + \epsilon$$

$$E(\epsilon) = 0$$

$$Var(\epsilon) = \sigma^2$$

$$E(y|x) = \beta_0 + \beta_1 x$$

$$Var(y|x) = Var(\beta_0 + \beta_1 x + \epsilon) = \sigma^2$$




```
> wind <- airquality$Wind
> temp <- airquality$Temp
> plot(temp, wind, main="scatterplot of wind vs temp")
```

- beta_0 (intercept), beta_1 (slope): parameters to be estimated from observed data.
- Random errors (epsilon): mean zero and unknown variance (sigma^2).
- The variance in y is constant (i.e. the variance does not change as y gets bigger).

參數估計: 最小平方法

$$(y_1, x_1), \cdots, (y_n, x_n)$$

$$S(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

```
> y <- airquality$Wind
> x <- airquality$Temp
> xbar <- mean(x) ; xbar
[1] 77.88235
> ybar <- mean(y) ; ybar
[1] 9.957516

> beta1.num <- sum((x-xbar)*(y-ybar))
> beta1.den <- sum((x-xbar)^2)
> (beta1.hat <- beta1.num/beta1.den)
[1] -0.1704644

> (beta0.hat <- ybar-beta1.hat*xbar)
[1] 23.23369
> yhat <- beta0.hat + beta1.hat * x
```

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}$$

$$S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

$$S_{yy} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$S_{xy} = \sum_{i=1}^{n} y_i (x_i - \bar{x})$$

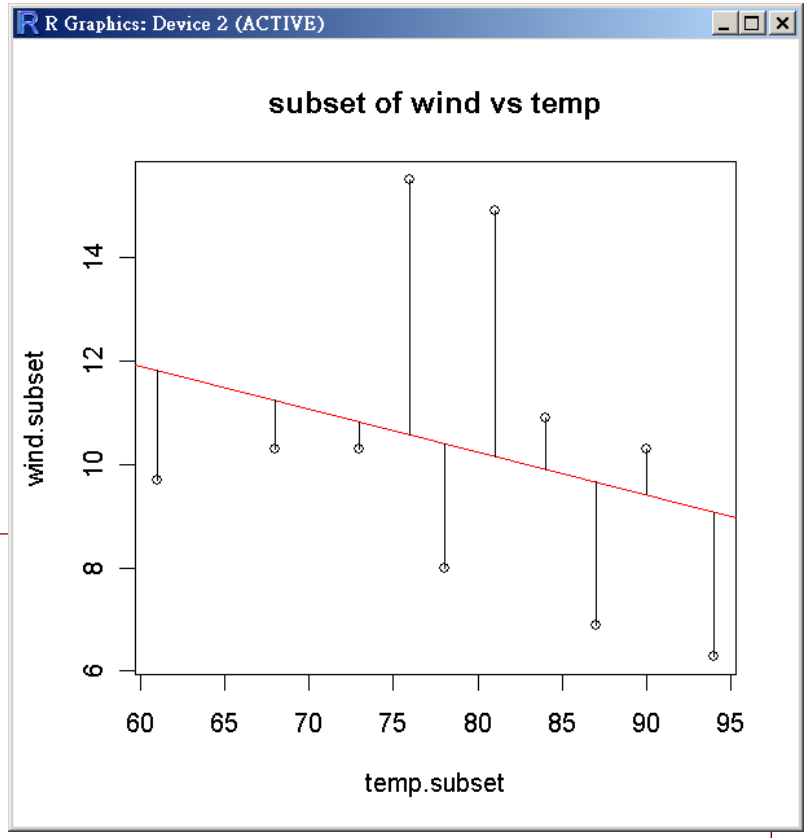
```
> Sxy <- sum(y*(x-xbar)) ; Sxy
[1] -2321.365
> Sxx <- sum((x-xbar)^2) ; Sxx
[1] 13617.88
> Syy <- sum((y-ybar)^2) ; Syy
[1] 1886.554
> beta1.hat2 <- Sxy/Sxx ; beta1.hat2
[1] -0.1704644
```

最小平方法

$$S(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

$$e_i = y_i - \hat{y}_i$$

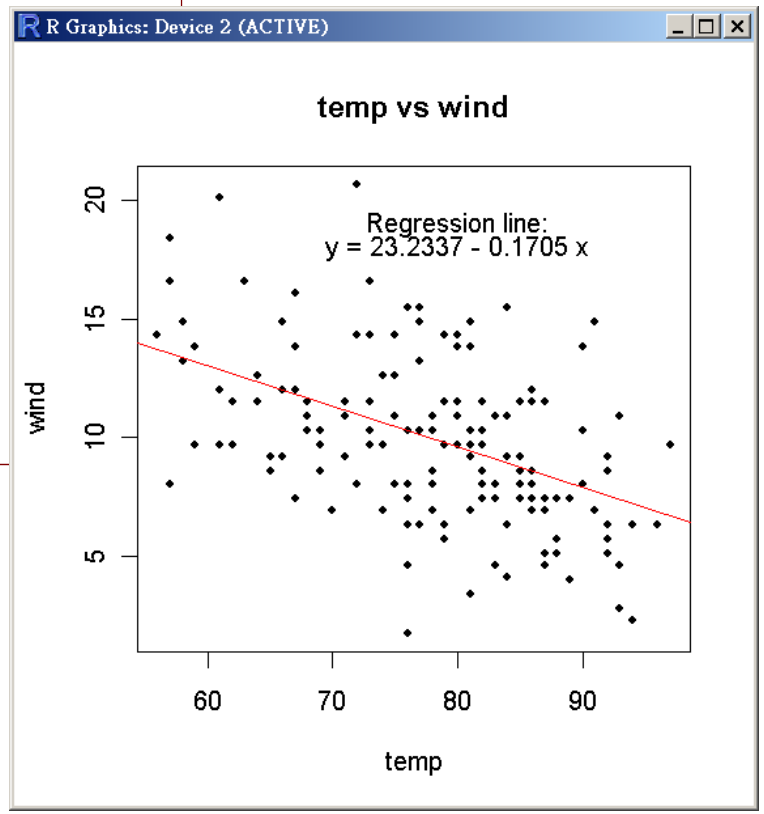
和 **summary(lm(y~x))** 比較




```
> wind <- airquality$Wind
> temp <- airquality$Temp

> n <- length(wind)
> index <- sample(1:n, 10)
> wind.subset <- wind[index]
> temp.subset <- temp[index]

> plot(wind.subset~temp.subset, main="subset of wind vs temp")
> subset.lm <- lm(wind.subset~temp.subset)
> abline(subset.lm, col="red")
> segments(temp.subset, fitted(subset.lm), temp.subset, wind.subset)
```

Find the Least Squares Fit

```
> model.fit <- lsfit(temp, wind)
> ls.print(model.fit)
Residual Standard Error=3.1422
R-Square=0.2098
F-statistic (df=1, 151)=40.0795
p-value=0

          Estimate Std.Err t-value Pr(>|t|)
Intercept  23.2337  2.1124 10.9987        0
X          -0.1705  0.0269 -6.3308        0


> plot(temp, wind, main="temp vs wind", pch=20)
> abline(model.fit, col="red")
> text(80,19, "Regression line:")
> text(80,18, "y = 23.2337 - 0.1705 x")
```



Assume a Distributional Form for the Errors ε

- Up till now, we haven't found it necessary to assume any distributional form for the errors ε. However, if we want to make any confidence intervals or perform any hypothesis tests, we will need to do this.

$$\varepsilon \sim N(0, \sigma^2 I) \qquad y = X\beta + \varepsilon,$$

$$y \sim N(X\beta, \sigma^2 I)$$

$$\hat{\beta} = (X^T X)^{-1} X^T y \sim N(\beta, (X^T X)^{-1} \sigma^2)$$

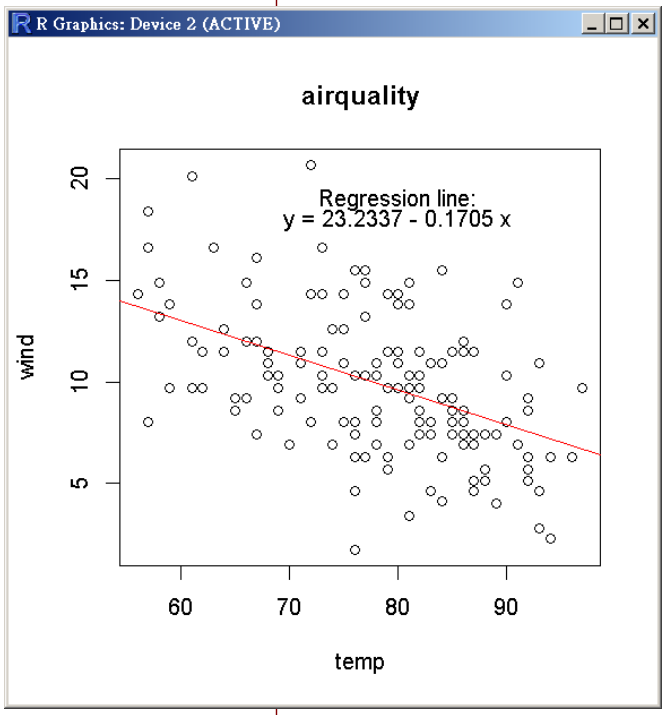
Testing just one predictor $\quad H_0 : \beta_i = 0. \qquad t_i = \hat{\beta}_i / se(\hat{\beta}_i)$

Test of all predictors $\quad H_0 : \beta_1 = \ldots \beta_{p-1} = 0 \qquad F = \dfrac{(\mathrm{SYY} - \mathrm{RSS})/(p-1)}{\mathrm{RSS}/(n-p)}$

Fit A Linear Model: lm

```
> my.model <- lm(wind ~ temp)
> my.model

Call:
lm(formula = wind ~ temp)

Coefficients:
(Intercept)         temp
    23.2337      -0.1705

> summary(my.model)

Call:
lm(formula = wind ~ temp)

Residuals:
    Min      1Q  Median      3Q     Max
-8.5784 -2.4489 -0.2261  1.9853  9.7398

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 23.23369    2.11239  10.999  < 2e-16 ***
temp        -0.17046    0.02693  -6.331 2.64e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.142 on 151 degrees of freedom
Multiple R-squared: 0.2098,     Adjusted R-squared: 0.2045
F-statistic: 40.08 on 1 and 151 DF,  p-value: 2.642e-09
```

```
> plot(wind ~ temp, main="airquality")
> abline(my.model, col="red")
> text(80,19, "Regression line:")
> text(80,18, "y = 23.2337 - 0.1705 x")
```



Test of all Predictors and ANOVA Table

$$e_i = y_i - \hat{y}_i$$

$$SS_E = \sum_{i=1}^{n} e_i^2 \qquad MS_E = \frac{SS_E}{n-2} = \hat{\sigma}^2$$

$$SS_R = \hat{\beta}_1 S_{xy} \qquad MS_R = SS_R/1 \qquad F_0 = MS_R/MS_E$$

```
> n <- length(wind)
> e <- y-yhat
> SSE <- sum(e^2) ; SSE
[1] 1490.844
> MSE <- SSE/(n-2) ; MSE
[1] 9.873137
> SSR <- beta1.hat*Sxy ; SSR
[1] 395.7101
> MSR <- SSR/1 ; MSR
[1] 395.7101
> SST <- SSR + SSE ; SST
[1] 1886.554
> Syy
[1] 1886.554
> F0 <- MSR/MSE; F0
[1] 40.07947
```

```
> my.aov <- aov(my.model)
> summary(my.aov)
            Df  Sum Sq Mean Sq F value    Pr(>F)
temp         1  395.71  395.71  40.080 2.642e-09 ***
Residuals  151 1490.84    9.87
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.
```

The ANOVA Table for Regression

Source	SS *(Sum of Squares, the numerator of the variance)*	DF *(the denominator)*	MS *(Mean Square, the variance)*	F
Regression *(or Model)*	$SSR = \sum_{i=1}^{n} ((\hat{\beta}_0 + \hat{\beta}_1 x_i) - \bar{y})^2$	*2-1=1*	$MSR = \frac{SSR}{1}$	$F = \frac{MSR}{MSE}$
Error	$SSE = \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$	*n-2*	$MSE = \frac{SSE}{n-2}$	
Total	$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$	*n-1*		

$$H_0 : \beta_1 = \ldots \beta_{p-1} = 0$$

課堂練習1: 估計量

- 用R寫出以下估計量，並與上述例子的答案比較。

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} y_i x_i - \frac{1}{n}(\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i)}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{2} x_i)^2}$$

$$SS_E = SS_T - SS_R$$

$$SS_E = S_{yy} - \hat{\beta}_1 S_{xy}$$

$$R^2 = \frac{SS_R}{S_{yy}} = 1 - \frac{SS_E}{S_{yy}}$$

決定系數 Coefficient of Determination

參數估計之信賴區間

$100(1-\alpha)\%$ confident interval on the intercept β_0.

$$E(\hat{\beta}_0) = \beta_0 \qquad se(\hat{\beta}_0) = \sqrt{MS_E(1/n + \bar{x}^2/S_{xx})}$$

$$\hat{\beta}_0 - t_{\alpha/2,n-1}se(\hat{\beta}_0) \le \beta_0 \le \hat{\beta}_0 + t_{\alpha/2,n-1}se(\hat{\beta}_0)$$

$100(1-\alpha)\%$ confident interval on the slope β_1.

$$E(\hat{\beta}_1) = \beta_1 \qquad se(\hat{\beta}_1) = \sqrt{MS_E/S_{xx}}$$

$$\hat{\beta}_1 - t_{\alpha/2,n-1}se(\hat{\beta}_1) \le \beta_1 \le \hat{\beta}_1 + t_{\alpha/2,n-1}se(\hat{\beta}_1)$$

```
> alpha <- 0.05
> se.beta0 <- sqrt(MSE*(1/n+xbar^2/Sxx)) ; se.beta0
[1] 2.112395
> tstar <- qt(alpha/2, n-1)* se.beta0
> CI.beta0 <- beta0.hat + c(-tstar*se.beta0, tstar*se.beta0) ; CI.beta0
[1] 32.04965 14.41772
```

```
> se.beta1 <- sqrt(MSE/Sxx) ; se.beta1
[1] 0.02692606
> tstar <- qt(alpha/2, n-1)* se.beta1
> CI.beta1 <- beta1.hat + c(-tstar*se.beta0, tstar*se.beta1); CI.beta1
[1] -0.0580900 -0.1718968
```

課堂練習2: 信賴區間

- 用R寫出以下估計量，並用以上的例子算出答案。

$100(1-\alpha)\%$ confident interval on σ^2.

$$\frac{(n-2)MS_E}{\chi^2_{\alpha/2,n-2}} \leq \sigma^2 \leq \frac{(n-2)MS_E}{\chi^2_{1-\alpha/2,n-2}}$$

$100(1-\alpha)\%$ confident interval on

the mean response at the point $x = x_0$.

$$\hat{y}_0 - t_{\alpha/2,n-2}\sqrt{MS_E(\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}})} \leq E(y|x_0) \leq \hat{y}_0 + t_{\alpha/2,n-2}\sqrt{MS_E(\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}})}$$

Generic Functions

```
> my.model <- lm(wind ~ temp)
> summary(my.model)
```

- **summary**: produces parameter estimates and standard errors from **lm**, and ANOVA tables from **aov**.
- **plot**: produces diagnostic plots for model checking, including residuals against fitted values, influence tests, etc.
- **update**: is used to modify the last model fit; it saves both typing effort and computing time.
- **predict**: uses information from the fitted model to produce smooth functions for plotting a line through the scatterplot of your data.
- **fitted**: gives the fitted values, predicted by the model for the values of the explanatory variables included.
- **resid**: gives the residuals.

Extract Information from Model Objects

方法一: by functions

```
> my.model <- lm(wind ~ temp)
> summary(my.model)

Call:
lm(formula = wind ~ temp)

Residuals:
    Min      1Q  Median      3Q     Max
-8.5784 -2.4489 -0.2261  1.9853  9.7398

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 23.23369    2.11239  10.999  < 2e-16 ***
temp        -0.17046    0.02693  -6.331 2.64e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.142 on 151 degrees of freedom
Multiple R-squared: 0.2098,     Adjusted R-squared: 0.2045
F-statistic: 40.08 on 1 and 151 DF,  p-value: 2.642e-09
```

```
> coef(my.model)
(Intercept)        temp
 23.2336881  -0.1704644

> vcov(my.model)
             (Intercept)          temp
(Intercept)  4.46221130 -0.0564656925
temp        -0.05646569  0.0007250127
```

Extract Information from Model Objects

```
> summary(my.model)[[1]]   # my.model formula
lm(formula = wind ~ temp)
> summary(my.model)[[2]]   # attributes of the objects
wind ~ temp
attr(,"variables")
list(wind, temp)
attr(,"factors")
     temp
wind    0
temp    1
attr(,"term.labels")
[1] "temp"
attr(,"order")
[1] 1
attr(,"intercept")
[1] 1
attr(,"response")
[1] 1
attr(,".Environment")
<environment: R_GlobalEnv>
attr(,"predvars")
list(wind, temp)
attr(,"dataClasses")
     wind      temp
"numeric" "numeric"
```

方法二: with list subscripts

```
> length(summary(my.model))
[1] 11
> names(summary(my.model))
 [1] "call"  "terms"  "residuals" "coefficients"
 [5] "aliased"  "sigma" "df"  "r.squared"
 [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
> summary(my.model)$sigma
[1] 3.142155
> summary(my.model)[[6]]
[1] 3.142155
> length(summary(my.model)[[1]])
[1] 2
> length(summary(my.model)[[2]])
[1] 3
> length(summary(my.model)[[3]])
[1] 153
```

Extract Information from Model Objects

方法二: with list subscripts

```
> summary(my.model)[[3]]  # residuals for data points
          1           2           3           4           5           6
-4.41257055 -2.96024835  1.98068054 -1.16489276  0.61232059  2.91696501
...
145         146         147         148         149         150
-1.93071279  0.87393162 -1.17164167  4.10557168 -4.40117723  3.09207386
        151         152         153
 3.85114498 -2.27839058 -0.14210611

> summary(my.model)[[4]]  # parameters table
              Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 23.2336881 2.11239468 10.998744 4.901351e-21
temp        -0.1704644 0.02692606 -6.330835 2.641597e-09

> summary(my.model)[[4]][[1]]  # intercept
[1] 23.23369

> summary(my.model)[[4]][[2]]  # slope,.... summary(my.model)[[4]][[28]]
[1] -0.1704644
```

```
> str(summary(my.model)[[4]])
 num [1:2, 1:4] 23.2337 -0.1705  2.1124  0.0269 10.9987 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:2] "(Intercept)" "temp"
  ..$ : chr [1:4] "Estimate" "Std. Error" "t value" "Pr(>|t|)"
```

Extract Information from Model Objects

方法二: with list subscripts

```
> summary(my.model)[[5]]  # whether the fit should be returned.
(Intercept)        temp
      FALSE       FALSE
> summary(my.model)[[6]]  # residual standard error
[1] 3.142155
> summary(my.model)[[7]]  # the number of rows in the summary.lm table.
[1]   2 151   2
> summary(my.model)[[8]]  # r square, the fraction of the total variation
   in the response variable that is explained by the my.model.
[1] 0.2097529
> summary(my.model)[[9]]  # adjusted r square
[1] 0.2045195
> summary(my.model)[[10]]  # F ratio information
    value     numdf     dendf
 40.07947   1.00000 151.00000
> summary(my.model)[[11]]  # correlation matrix of the parameter estimates.
              (Intercept)          temp
(Intercept)  0.451954754 -5.719124e-03
temp        -0.005719124  7.343286e-05
```

Extract Information from Model Objects

方法三: using $

```
> my.model <- lm(wind ~ temp)
> names(my.model)
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"

> model$coefficients
> model$fitted.values
> model$residuals
```

依此類推...

```
> summary.aov(my.model)
> summary.aov(my.model)[[1]][[1]]~
> summary.aov(my.model)[[1]][[5]]
```

Extract Information from Model Objects

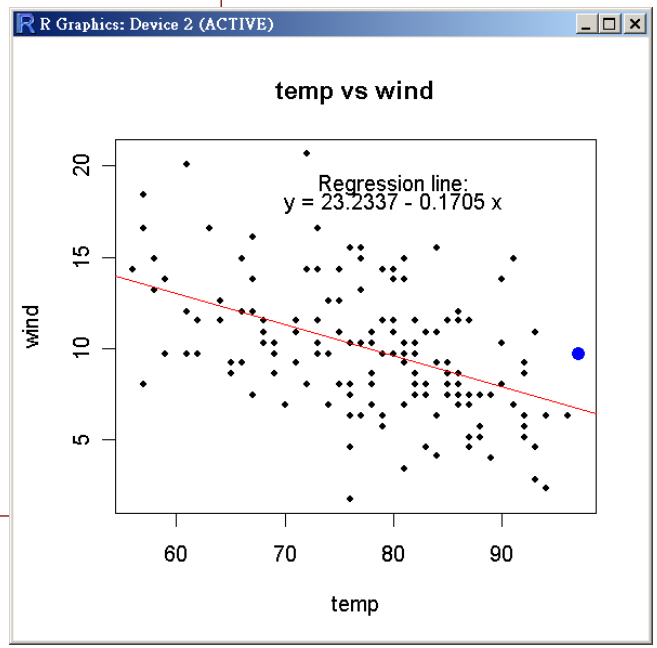
方法四: using ["names"]

```
> (iris.aov <- aov(iris[,1]~iris[,5]))
Call:
   aov(formula = iris[, 1] ~ iris[, 5])

Terms:
                iris[, 5] Residuals
Sum of Squares   63.21213  38.95620
Deg. of Freedom        2       147

Residual standard error: 0.5147894
Estimated effects may be unbalanced
> (iris.sum.aov <- summary(iris.aov))
             Df Sum Sq Mean Sq F value Pr(>F)
iris[, 5]     2  63.21  31.606   119.3 <2e-16 ***
Residuals   147  38.96   0.265
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> (iris.sum.aov2 <- unlist(iris.sum.aov))
         Df1          Df2       Sum Sq1       Sum Sq2      Mean Sq1      Mean Sq2      F value1
2.000000e+00 1.470000e+02 6.321213e+01 3.895620e+01 3.160607e+01 2.650082e-01 1.192645e+02
    F value2       Pr(>F)1       Pr(>F)2
          NA 1.669669e-31            NA
> names(iris.sum.aov2)
 [1] "Df1"      "Df2"      "Sum Sq1"  "Sum Sq2"  "Mean Sq1" "Mean Sq2" "F value1"
 [8] "F value2" "Pr(>F)1"  "Pr(>F)2"
> iris.sum.aov2["Pr(>F)1"]
     Pr(>F)1
1.669669e-31
```

使用子集合做分析

- Investigate how much a influence point affected the parameter estimates and their standard error.
- Repeat the statistical modeling but leave out the point in question, using subset.

```
> new.model <- update(my.model, subset=(temp!=max(temp)))
> summary(new.model)

Call:
lm(formula = wind ~ temp, subset = (temp != max(temp)))

Residuals:
    Min      1Q  Median      3Q     Max
-8.5663 -2.3871 -0.2027  1.9662  9.7344

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  23.5529     2.1382  11.015  < 2e-16 ***
temp         -0.1748     0.0273  -6.403 1.85e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.143 on 150 degrees of freedom
Multiple R-squared: 0.2147,     Adjusted R-squared: 0.2094
F-statistic:    41 on 1 and 150 DF,  p-value: 1.847e-09
```

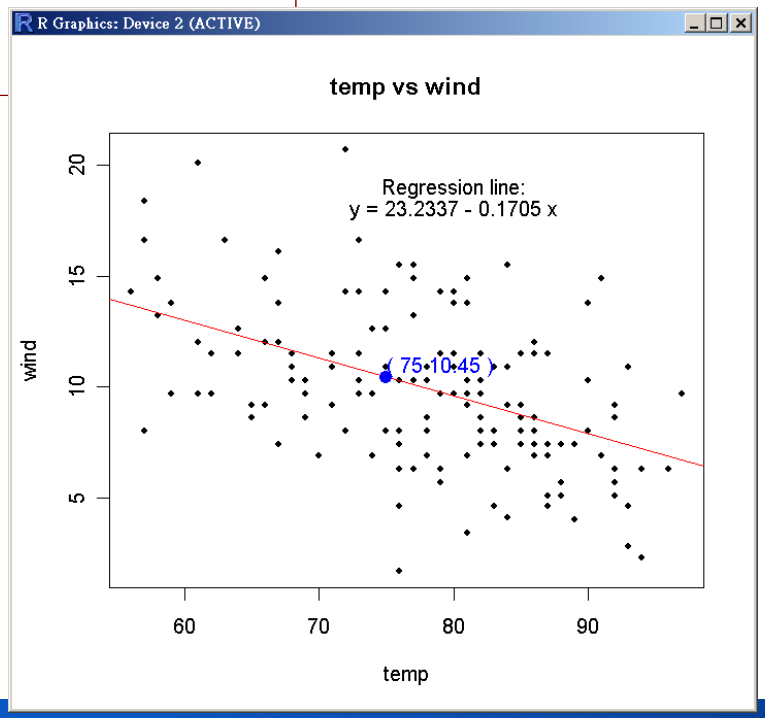


課堂練習：

- 將要刪除的點在二維散佈圖上標出來。
- 更新二維散佈圖及Regression Fit。

預測 (Prediction)

```
> summary(wind)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.700   7.400   9.700   9.958  11.500  20.700
> summary(temp)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  56.00   72.00   79.00   77.88   85.00   97.00

> predict(my.model, list(temp=75))
[1] 10.44886
> predict(my.model, list(temp=c(66, 80,100)))
        1         2         3
11.983035  9.596533  6.187244
```



課堂練習:

- 將predict出來的值在二維散佈圖上標出來。

統計模型檢測 (Model Checking in R)

- After fitting a model to data we need to investigate how well the model describes the data to see if there are any systematic trends in the goodness of fit.

- We hope that $\varepsilon \sim N(0,\sigma^2 I)$, but
  - Errors may be heterogeneous (unequal variance).
  - Errors may be correlated.
  - Errors may not be normally distributed. (less serious, the βhat's will tend to normality due to the power of the central limit theorem. With larger datasets, normality of the data is not much of a problem.

George Box (1919-2013),
Professor Emeritus of Statistics,
University of Wisconsin-Madison

"Essentially, all models are wrong, but some are useful"
https://en.wikipedia.org/wiki/All_models_are_wrong

Box married Joan Fisher,
the second of R.A. Fisher (1890-1962) five daughters.

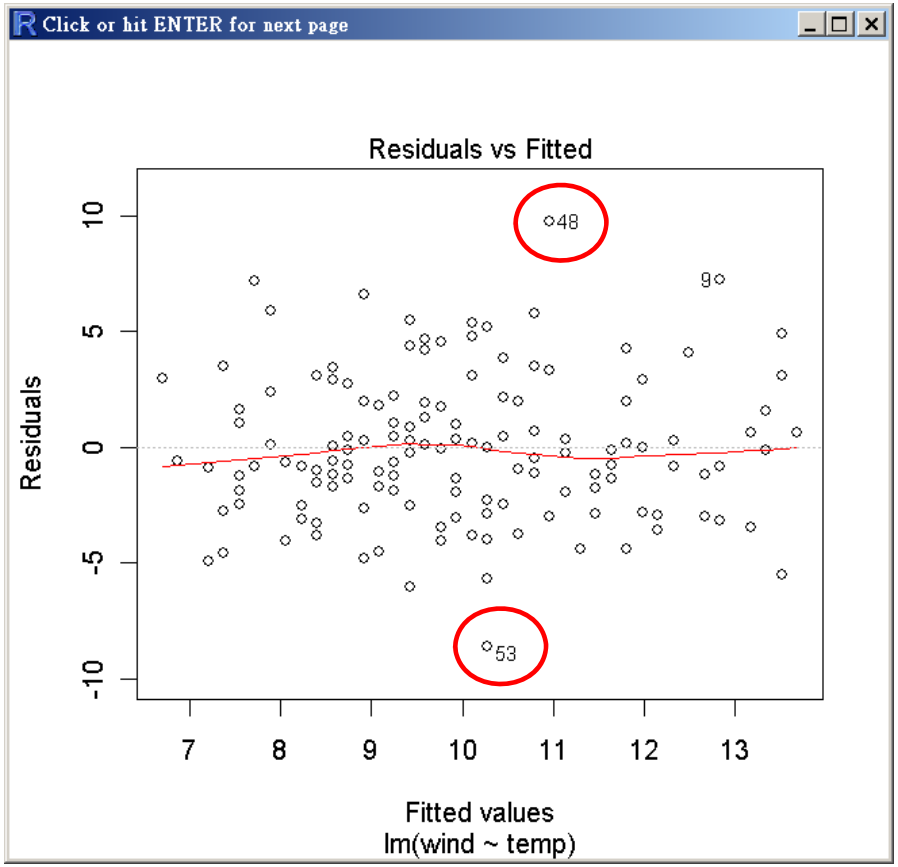
1. 殘差vs. 估計值: Residual Plots

```
> ?plot.lm
```

default

This plot should be
with no pattern of any sort.

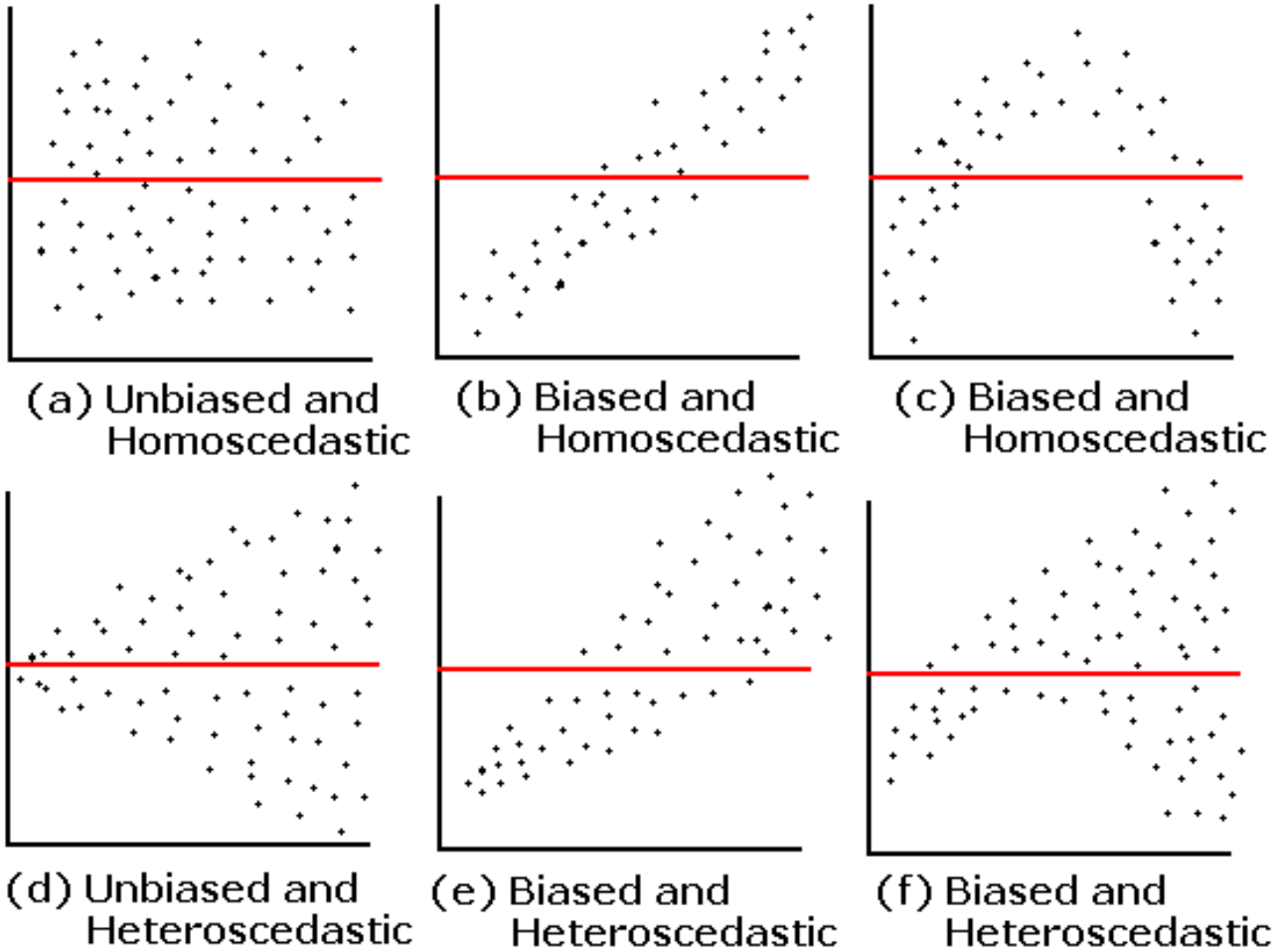
```
> wind <- airquality$Wind
> temp <- airquality$Temp
> my.model <- lm(wind ~ temp)
> plot(my.model, which=1:6)
Waiting to confirm page change...
```




```
> plot(fitted(my.model), residuals(my.model),xlab="Fitted values",
+  ylab="Residuals")
> abline(h=0, lty=2)
```

課堂練習: 將Residuals大於±6的點標出來(顏色為紅色)。

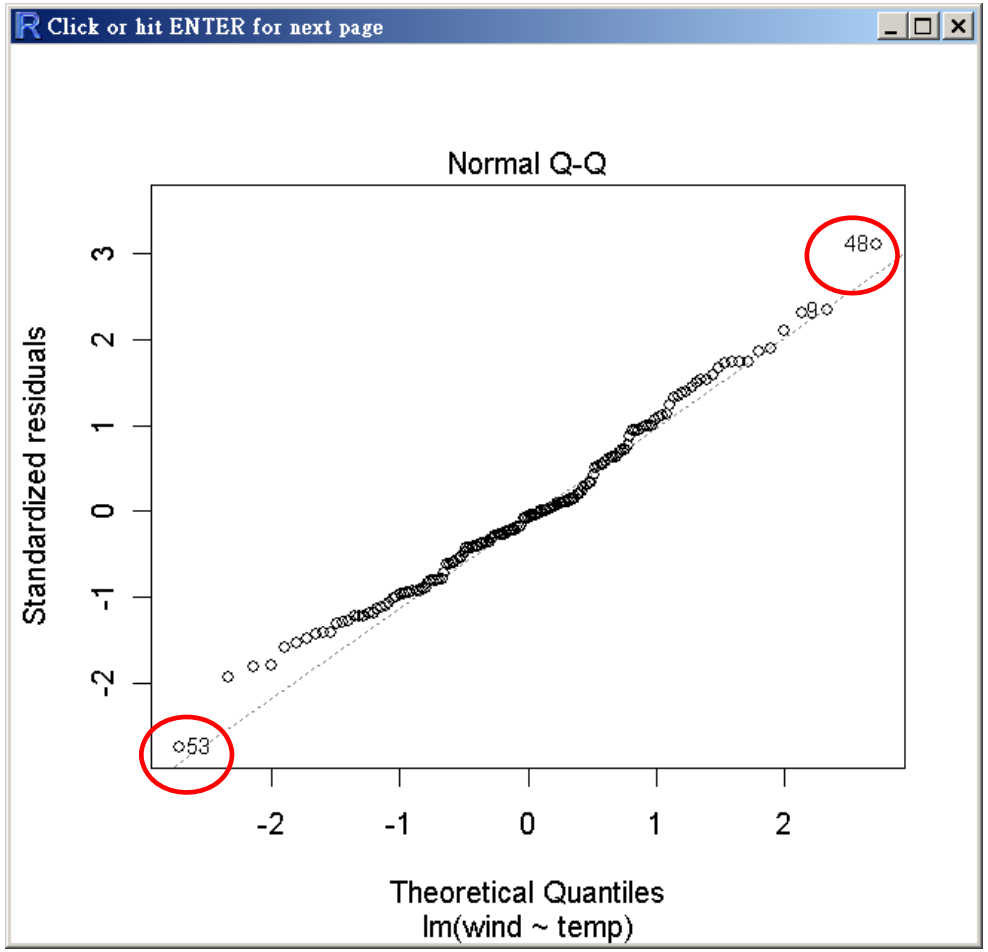
殘差圖 Residual Plots



https://www.r-bloggers.com/model-validation-interpreting-residual-plots/

2. 常態QQ圖 (Normal QQ-plot)

default

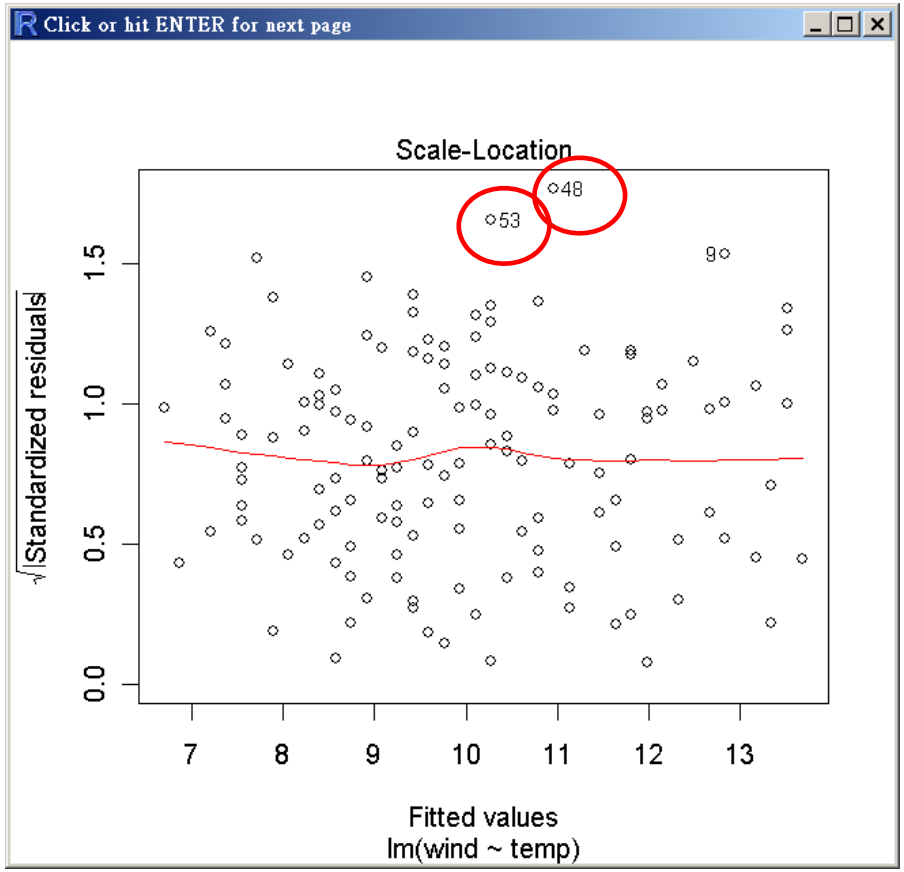



```
> qqnorm(residuals(my.model))
> qqline(residuals(my.model))
```

3. 尺度-位置圖 (A Scale-Location Plot)

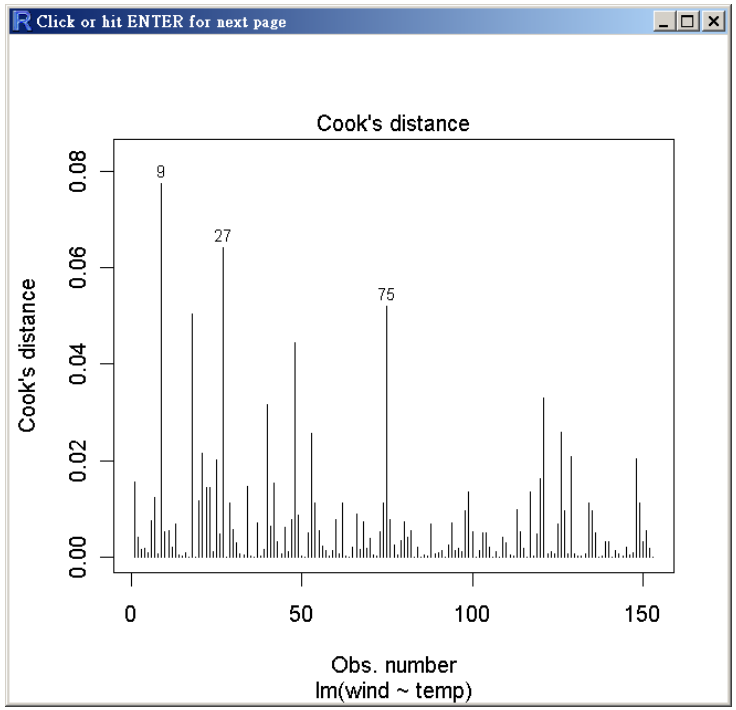
- A scale-loaction plot of sqrt(abs(residuals)) against fitted values.
- This is like a positive-valued version of the first graph; it is good for detecting non-constancy of variance (heteroscedasticity).

default



4. Plot of Cook's Distance vs Row Labels

- Cook's distance measures the effect of deleting a given observation.
- Cook's distance is a measure of the squared distance between the least square estimate based on all n points β and the estimate obtained by deleting the *i*th points $\beta_{(i)}$.
- Points with a Cook's distance of 1 or more are considered to be influential.



$$D_i = \frac{\sum_{j=1}^{n}\left(\hat{y}_j - \hat{y}_{j(i)}\right)^2}{pMS_E}$$

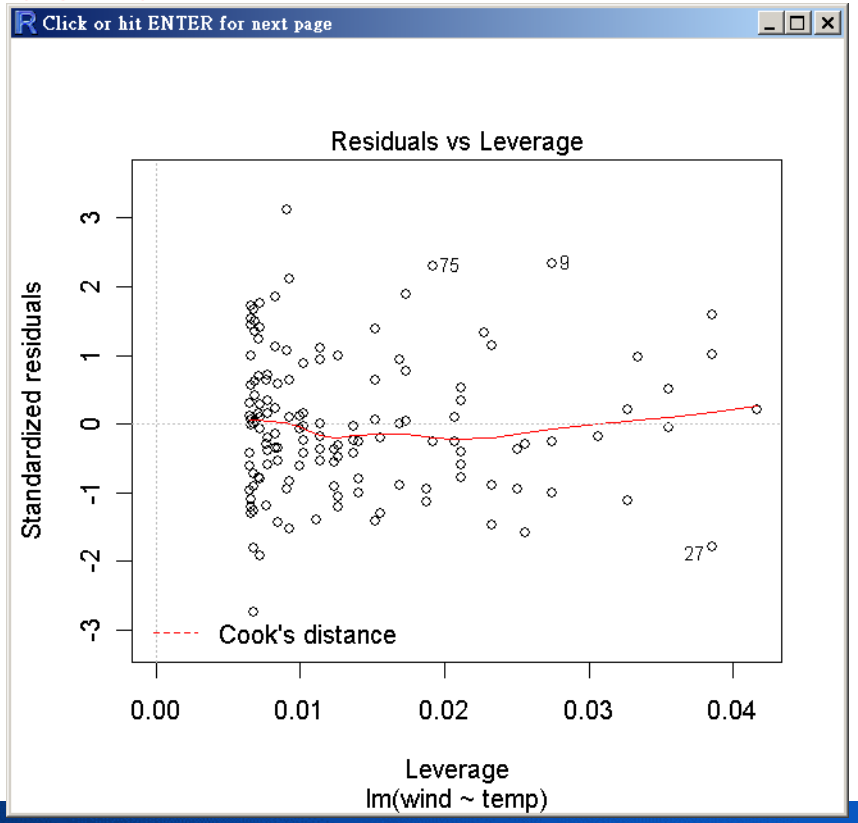
課堂練習:

- 算出Cook's Distance。
- 畫出Cook's Distance vs. Row Labels的散佈圖。
- 標出前三大Cook's Distance值所在位置。

5. Plot of Residuals vs Leverages

- Outliers in the response variable are called outliers.
- Outliers with respect to the predictors are called leverage points.
- For the regression, it is the points that have large leverage are important.
- Points that have small leverage "do not count" in the regression – we could move them or remove them from the data and the regression line does not change very much.

default



$$\mathrm{Le}_i = \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{n}(x_j - \bar{x})^2}$$

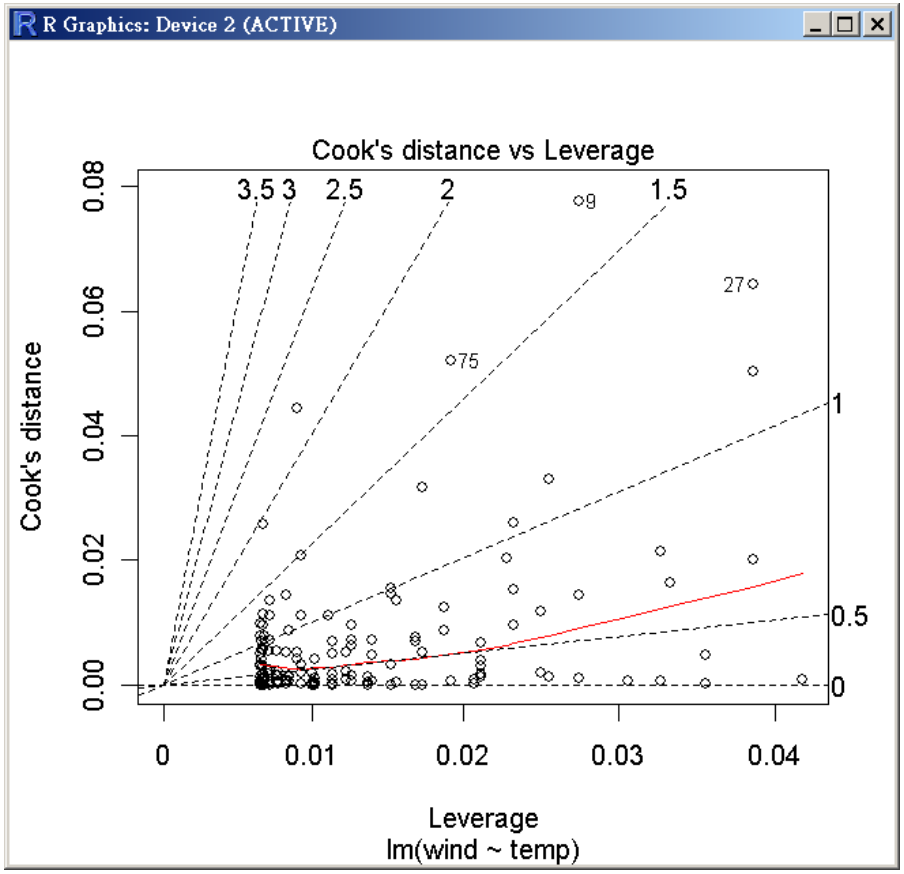
$$\hat{\beta}_1 = \sum_{i=1}^{n} \mathrm{Le}_i \frac{(y_i - \bar{y})}{(x_i - \bar{x})}$$

課堂練習：

- 算出Leverages 。
- 將Residuals標準化。
- 畫出Residuals標準化 vs. Leverages 的散佈圖。
- 標出前三大Leverages值所在位置。

6. Cook's Distance vs Leverage

- In the Cook's distance vs leverage/(1-leverage) plot, contours of standardized residuals that are equal in magnitude are lines through the origin.



範例: 模型選取/變數選取

Swiss Fertility and Socioeconomic Indicators (1888) Data

```
> head(swiss)
             Fertility Agriculture Examination Education Catholic Infant.Mortality
Courtelary        80.2        17.0          15        12     9.96             22.2
Delemont          83.1        45.1           6         9    84.84             22.2
Franches-Mnt      92.5        39.7           5         5    93.40             20.2
Moutier           85.8        36.5          12         7    33.77             20.3
Neuveville        76.9        43.5          17        15     5.16             20.6
Porrentruy        76.1        35.3           9         7    90.57             26.6
```

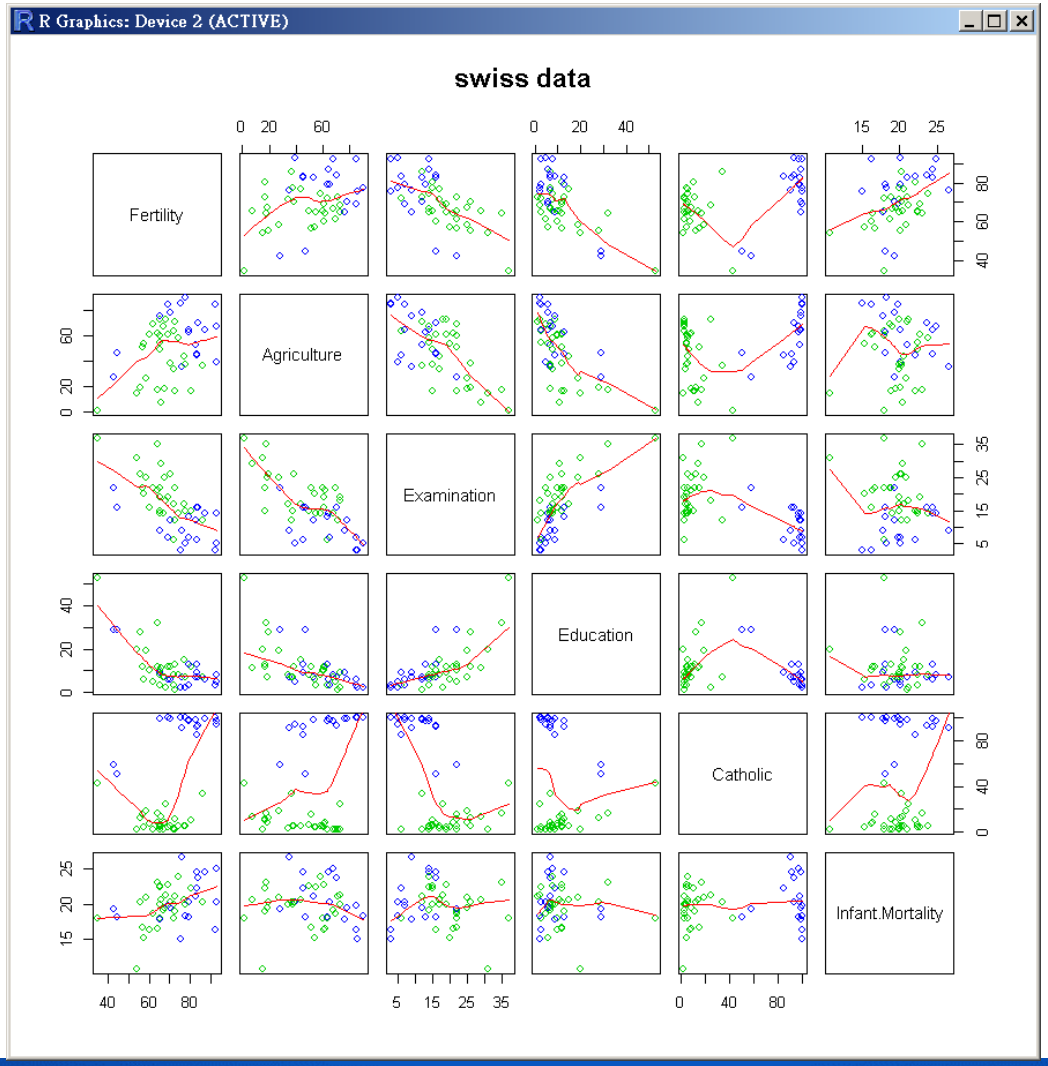
A data frame with 47 observations on 6 variables, each of which is in percent, i.e., in [0, 100].

[,1]	Fertility	Ig, ‘common standardized fertility measure’
[,2]	Agriculture	% of males involved in agriculture as occupation
[,3]	Examination	% draftees receiving highest mark on army examination
[,4]	Education	% education beyond primary school for draftees.
[,5]	Catholic	% ‘catholic’ (as opposed to ‘protestant’).
[,6]	Infant.Mortality	live births who live less than 1 year.

All variables but ‘Fertility’ give proportions of the population.

散佈圖矩陣

```
> pairs(swiss, panel = panel.smooth, main = "swiss data",
+       col = 3 + (swiss$Catholic > 50))
```



配適多重迴歸模型: lm

```
> summary(my.lm <- lm(Fertility ~ ., data = swiss))

Call:
lm(formula = Fertility ~ ., data = swiss)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.2743  -5.2617   0.5032   4.1198  15.3213 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
Agriculture      -0.17211    0.07030  -2.448  0.01873 *  
Examination      -0.25801    0.25388  -1.016  0.31546    
Education        -0.87094    0.18303  -4.758 2.43e-05 ***
Catholic          0.10412    0.03526   2.953  0.00519 ** 
Infant.Mortality  1.07705    0.38172   2.822  0.00734 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.165 on 41 degrees of freedom
Multiple R-squared:  0.7067,    Adjusted R-squared:  0.671 
F-statistic: 19.76 on 5 and 41 DF,  p-value: 5.594e-10
```

step()：逐步迴歸變數篩選

AIC (Akaike information criterion)常用來作為模型選取的準則。其值越小，代表模型的解釋能力越好(用的變數越少，或是誤差平方和越小)。

語法:

```
step(object, scope, scale = 0, direction = c("both", "backward", "forward"),
trace = 1, keep = NULL, steps = 1000, k = 2, ...)
```

```
> smy.lm <- step(my.lm)
Start:  AIC=190.69
Fertility ~ Agriculture + Examination + Education + Catholic +
    Infant.Mortality

                   Df Sum of Sq    RSS    AIC
- Examination       1     53.03 2158.1 189.86
<none>                          2105.0 190.69
- Agriculture       1    307.72 2412.8 195.10
- Infant.Mortality  1    408.75 2513.8 197.03
- Catholic          1    447.71 2552.8 197.75
- Education         1   1162.56 3267.6 209.36

Step:  AIC=189.86
Fertility ~ Agriculture + Education + Catholic + Infant.Mortality

                   Df Sum of Sq    RSS    AIC
<none>                          2158.1 189.86
- Agriculture       1    264.18 2422.2 193.29
- Infant.Mortality  1    409.81 2567.9 196.03
- Catholic          1    956.57 3114.6 205.10
- Education         1   2249.97 4408.0 221.43
```

$$AIC = \ln\left(\frac{ESS}{n}\right) + \frac{2p}{n}, \quad ESS = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

最後選取的模型

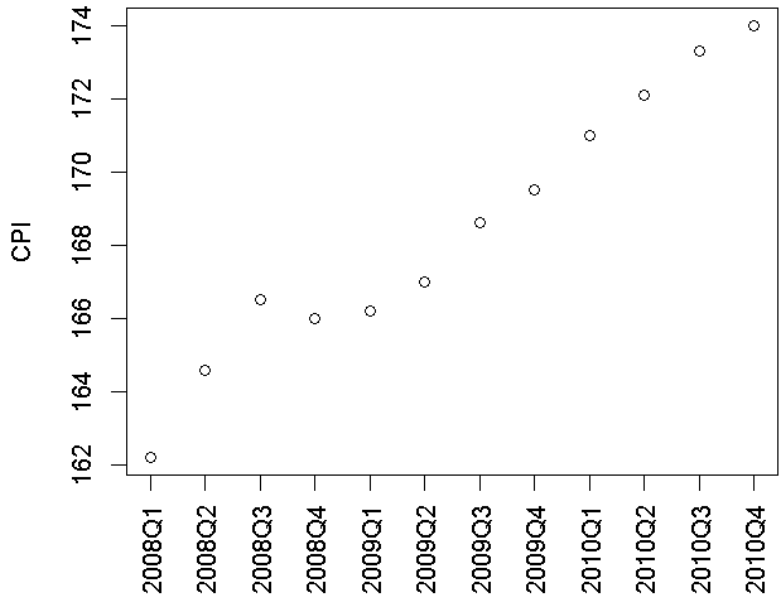
```
> summary(smy.lm)

Call:
lm(formula = Fertility ~ Agriculture + Education + Catholic +
    Infant.Mortality, data = swiss)

Residuals:
     Min       1Q   Median       3Q      Max
-14.6765  -6.0522   0.7514   3.1664  16.1422

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      62.10131    9.60489   6.466 8.49e-08 ***
Agriculture      -0.15462    0.06819  -2.267  0.02857 *
Education        -0.98026    0.14814  -6.617 5.14e-08 ***
Catholic          0.12467    0.02889   4.315 9.50e-05 ***
Infant.Mortality  1.07844    0.38187   2.824  0.00722 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.168 on 42 degrees of freedom
Multiple R-squared:  0.6993,    Adjusted R-squared:  0.6707
F-statistic: 24.42 on 4 and 42 DF,  p-value: 1.717e-10
```

範例: Linear Regression

- Australian CPI (Consumer Price Index) data, which are quarterly CPIs from 2008 to 2010. From Australian Bureau of Statistics (http://www.abs.gov.au)

```
> year <- rep(2008:2010, each=4)
> quarter <- rep(1:4, 3)
> cpi <- c(162.2, 164.6, 166.5, 166.0,
+          166.2, 167.0, 168.6, 169.5,
+          171.0, 172.1, 173.3, 174.0)
> cbind(cpi, year, quarter)
        cpi year quarter
 [1,] 162.2 2008       1
 [2,] 164.6 2008       2
 [3,] 166.5 2008       3
 [4,] 166.0 2008       4
 [5,] 166.2 2009       1
 [6,] 167.0 2009       2
 [7,] 168.6 2009       3
 [8,] 169.5 2009       4
 [9,] 171.0 2010       1
[10,] 172.1 2010       2
[11,] 173.3 2010       3
[12,] 174.0 2010       4
> plot(cpi, xaxt="n", ylab="CPI", xlab="")
> axis(1, labels=paste(year, quarter, sep="Q"),
at=1:12, las=3) # las=3: vertical text.
```

```
> cor(year, cpi)
[1] 0.9096316
> cor(quarter, cpi)
[1] 0.3738028
```



Source: R and Data Mining: Examples and Case Studies, Chapter 5: Regression

Modeling

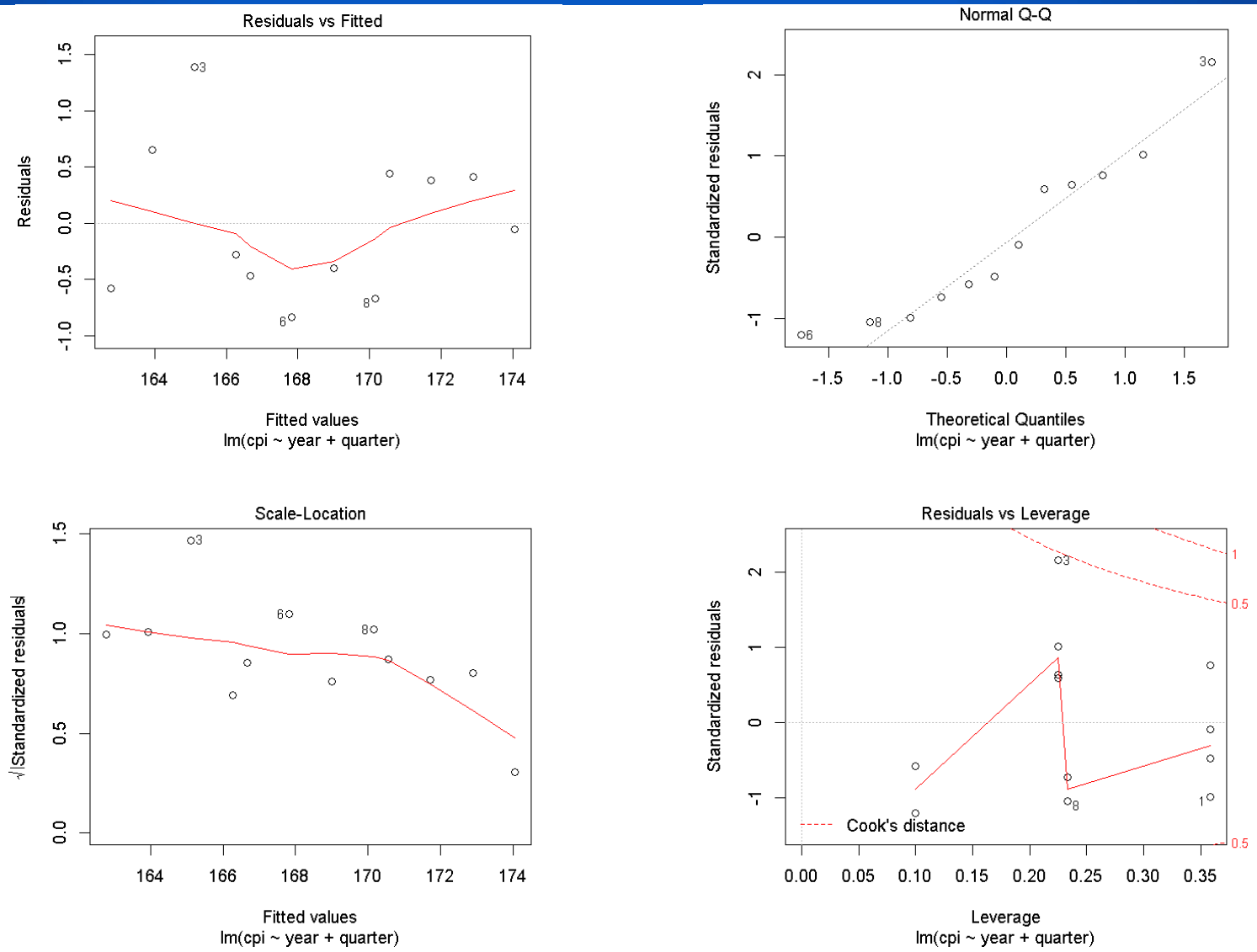
```
> fit <- lm(cpi ~ year + quarter)
> fit

Call:
lm(formula = cpi ~ year + quarter)

Coefficients:
(Intercept)         year      quarter
  -7644.488        3.888        1.167

> attributes(fit)
$names
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"

$class
[1] "lm"

> fit$coefficients
 (Intercept)         year      quarter
-7644.487500     3.887500     1.166667
> residuals(fit)
          1           2           3           4           5           6
-0.57916667  0.65416667  1.38750000 -0.27916667 -0.46666667 -0.83333333
          7           8           9          10          11          12
-0.40000000 -0.66666667  0.44583333  0.37916667  0.41250000 -0.05416667
```

Summary of Fit

```
> summary(fit)

Call:
lm(formula = cpi ~ year + quarter)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.8333 -0.4948 -0.1667  0.4208  1.3875 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -7644.4875   518.6543 -14.739 1.31e-07 ***
year            3.8875     0.2582  15.058 1.09e-07 ***
quarter         1.1667     0.1885   6.188 0.000161 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7302 on 9 degrees of freedom
Multiple R-squared:  0.9672,    Adjusted R-squared:  0.9599 
F-statistic: 132.5 on 2 and 9 DF,  p-value: 2.108e-07
```

Model Diagnostic

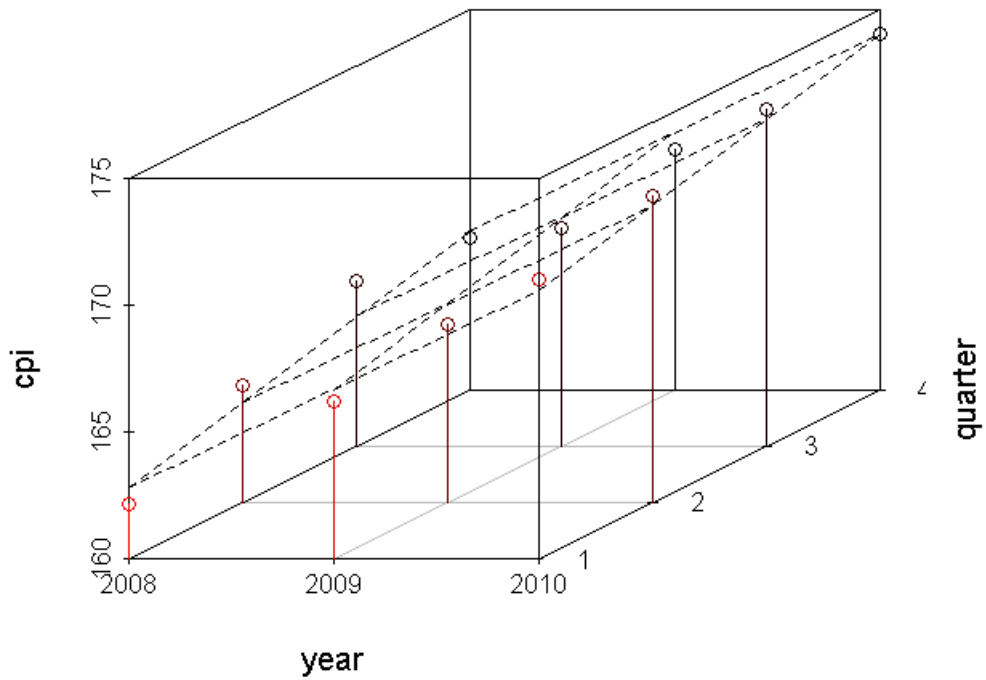
```
> plot(fit)
```



R and Data Mining: Examples and Case Studies, Chapter 5: Regression

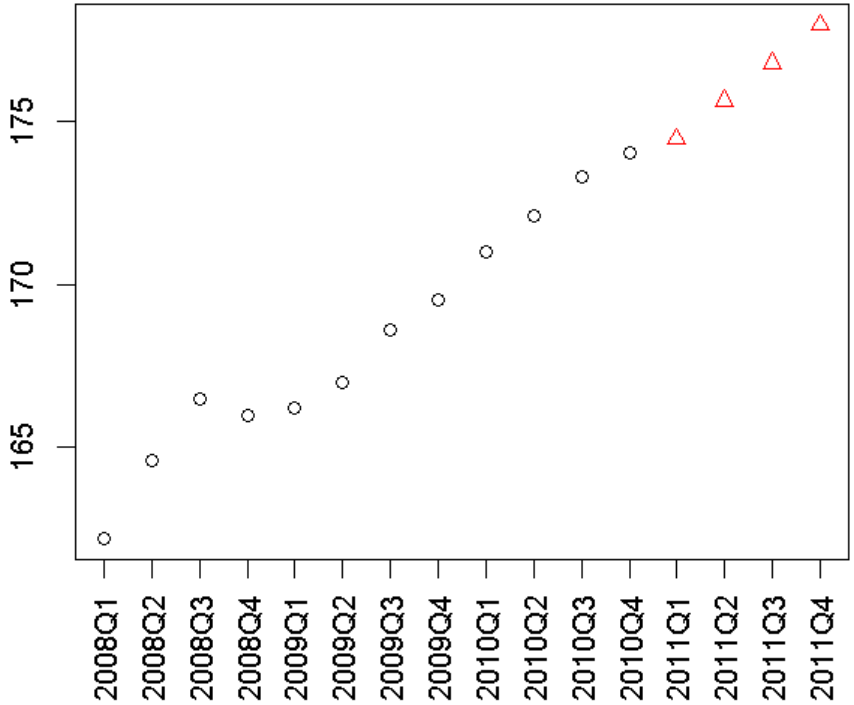
Plot of Fit

```
> library(scatterplot3d)
> s3d <- scatterplot3d(year, quarter, cpi, highlight.3d=T, type="h", lab=c(2,3))
> s3d$plane3d(fit)
```



Prediction

```
> data2011 <- data.frame(year=2011, quarter=1:4)
> cpi2011 <- predict(fit, newdata=data2011)
> cpi2011
       1        2        3        4
174.4417 175.6083 176.7750 177.9417
> style <- c(rep(1,12), rep(2,4))
> plot(c(cpi, cpi2011), xaxt="n", ylab="CPI", xlab="", pch=style, col=style)
> axis(1, at=1:16, las=3, labels=c(paste(year, quarter, sep="Q"), "2011Q1", "2011Q2",
"2011Q3", "2011Q4"))
```



範例: Logistic Regression

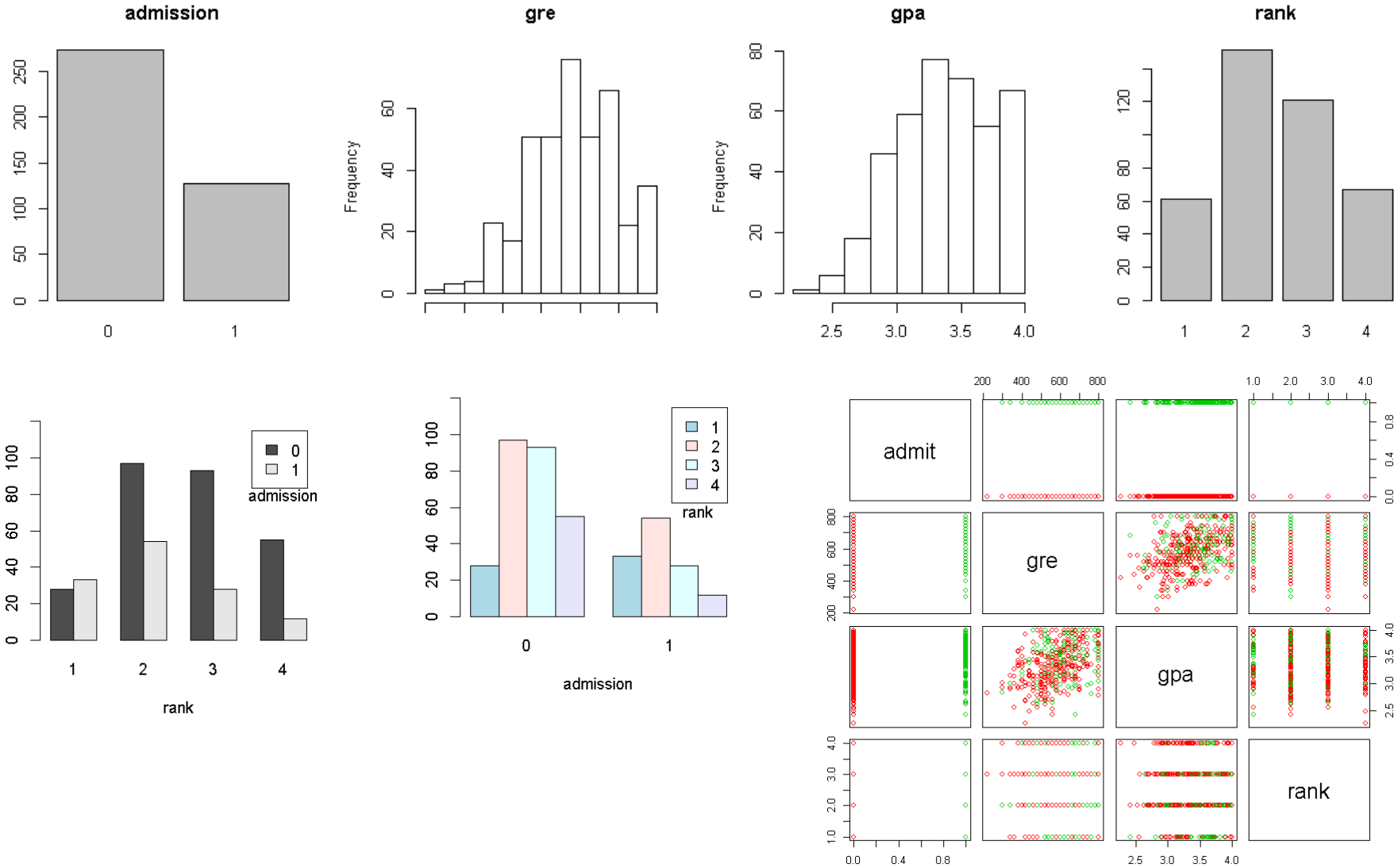
- A researcher is interested in how variables, such as **GRE** (Graduate Record Exam scores), **GPA** (grade point average) and prestige of the undergraduate institution (**rank**=1,2,3,4, 1 =highest prestige, 4 = the lowest), effect **admission** into graduate school.
- The response variable, admit/don't admit, is a binary variable.

```
> mydata <- read.csv("http://www.ats.ucla.edu/stat/data/binary.csv")
> dim(mydata)
[1] 400   4
> head(mydata)
  admit gre  gpa rank
1     0 380 3.61    3
2     1 660 3.67    3
3     1 800 4.00    1
4     1 640 3.19    4
5     0 520 2.93    4
6     1 760 3.00    2
> summary(mydata)
     admit             gre             gpa             rank
 Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000
 1st Qu.:0.0000   1st Qu.:520.0   1st Qu.:3.130   1st Qu.:2.000
 Median :0.0000   Median :580.0   Median :3.395   Median :2.000
 Mean   :0.3175   Mean   :587.7   Mean   :3.390   Mean   :2.485
 3rd Qu.:1.0000   3rd Qu.:660.0   3rd Qu.:3.670   3rd Qu.:3.000
 Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000
> sapply(mydata, sd)
      admit         gre         gpa        rank
  0.4660867 115.5165364   0.3805668   0.9444602
```

```
> xtabs(~ admit + rank, data = mydata)
     rank
admit  1  2  3  4
    0 28 97 93 55
    1 33 54 28 12
> mydata$rank <- factor(mydata$rank)
```

UCLA Statistical Consulting Group. http://www.ats.ucla.edu/stat/r/dae/logit.htm

Statistical Graphics



Modeling and Summary of Fit

```
> mylogit <- glm(admit ~ gre + gpa + rank, data = mydata, family = "binomial")
> summary(mylogit)

Call:
glm(formula = admit ~ gre + gpa + rank, family = "binomial",
    data = mydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6268  -0.8662  -0.6388   1.1490   2.0790

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.989979   1.139951  -3.500 0.000465 ***
gre          0.002264   0.001094   2.070 0.038465 *
gpa          0.804038   0.331819   2.423 0.015388 *
rank2       -0.675443   0.316490  -2.134 0.032829 *
rank3       -1.340204   0.345306  -3.881 0.000104 ***
rank4       -1.551464   0.417832  -3.713 0.000205 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 458.52  on 394  degrees of freedom
AIC: 470.52

Number of Fisher Scoring iterations: 4
```

- For every one unit change in **gre**, the log odds of admission (versus non-admission) increases by 0.002.
- For a one unit increase in **gpa**, the log odds of being admitted to graduate school increases by 0.804.

Having attended an undergraduate institution with rank of 2, versus an institution with a rank of 1, changes the log odds of admission by -0.675.

UCLA Statistical Consulting Group. http://www.ats.ucla.edu/stat/r/dae/logit.htm

Wald Test for Model Coefficients

```
wald.test(Sigma, b, Terms = NULL, L = NULL, H0 = NULL, df = NULL, verbose = FALSE)
```

Sigma: the variance covariance matrix of the error terms, **b**: the coefficients, **Terms**: terms in the model are to be tested, in this case, terms 4, 5, and 6, are the three terms for the levels of rank.

```
> library(aod) #aod: Analysis of Overdispersed Data
> wald.test(b = coef(mylogit), Sigma = vcov(mylogit), Terms = 4:6)
Wald test:
----------

Chi-squared test:
X2 = 20.9, df = 3, P(> X2) = 0.00011
```

To contrast two terms, we multiply one of them by 1, and the other by -1. The other terms in the model are not involved in the test, so they are multiplied by 0.
Test the difference (subtraction) of the terms for **rank=2** and **rank=3** (i.e., the 4th and 5th terms in the model). **L=l**: base the test on the vector **l** (rather than using the Terms option).

```
> l <- cbind(0,0,0,1,-1,0)
> wald.test(b = coef(mylogit), Sigma = vcov(mylogit), L = l)
Wald test:
----------

Chi-squared test:
X2 = 5.5, df = 1, P(> X2) = 0.019
```

The difference between the coefficient for **rank=2** and the coefficient for **rank=3** is statistically significant.

UCLA Statistical Consulting Group. http://www.ats.ucla.edu/stat/r/dae/logit.htm

Interpret Coefficients as Odds-ratios

```
> exp(cbind(OR = coef(mylogit), confint(mylogit)))
Waiting for profiling to be done...
                    OR       2.5 %    97.5 %
(Intercept) 0.0185001 0.001889165 0.1665354
gre         1.0022670 1.000137602 1.0044457
gpa         2.2345448 1.173858216 4.3238349
rank2       0.5089310 0.272289674 0.9448343
rank3       0.2617923 0.131641717 0.5115181
rank4       0.2119375 0.090715546 0.4706961
```

- For a one unit increase in **gpa**, the odds of being admitted to graduate school (versus not being admitted) increase by a factor of 2.23.
- For more information on interpreting odds ratios see our FAQ page How do I interpret odds ratios in logistic regression? http://www.ats.ucla.edu/stat/mult_pkg/faq/general/odds_ratio.htm
- Note that while R produces it, the odds ratio for the intercept is not generally interpreted.

http://www.ats.ucla.edu/stat/r/dae/logit.htm

Predicted Probabilities

- Predicted probabilities can be computed for both categorical and continuous predictor variables.
- Want to calculate the predicted probability of admission at each value of **rank**, holding gre and **gpa** at their means.

```
> newdata1 <- with(mydata, data.frame(gre = mean(gre), gpa = mean(gpa), rank = factor(1:4)))
> newdata1
    gre    gpa rank
1 587.7 3.3899    1
2 587.7 3.3899    2
3 587.7 3.3899    3
4 587.7 3.3899    4
> newdata1$rankP <- predict(mylogit, newdata = newdata1, type = "response")
> newdata1
    gre    gpa rank     rankP
1 587.7 3.3899    1 0.5166016
2 587.7 3.3899    2 0.3522846
3 587.7 3.3899    3 0.2186120
4 587.7 3.3899    4 0.1846684
```

- The predicted probability of being accepted into a graduate program is 0.52 for students from the highest prestige undergraduate institutions (**rank=1**), and 0.18 for students from the lowest ranked institutions (**rank=4**), holding **gre** and **gpa** at their means.

http://www.ats.ucla.edu/stat/r/dae/logit.htm

Create a Table of Predicted Probabilities

```
> newdata2 <- with(mydata,
+   data.frame(gre = rep(seq(from = 200, to = 800, length.out = 100), 4),
+   gpa = mean(gpa), rank = factor(rep(1:4, each = 100))))
> dim(newdata2)
[1] 400   3
> head(newdata2)
       gre    gpa rank
1 200.0000 3.3899    1
2 206.0606 3.3899    1
3 212.1212 3.3899    1
4 218.1818 3.3899    1
5 224.2424 3.3899    1
6 230.3030 3.3899    1
>
> newdata3 <- cbind(newdata2, predict(mylogit, newdata = newdata2, type="link", se=TRUE))
> newdata3 <- within(newdata3, {
+   PredictedProb <- plogis(fit)
+   LL <- plogis(fit - (1.96 * se.fit))
+   UL <- plogis(fit + (1.96 * se.fit))
+ })
> head(newdata3)
       gre    gpa rank        fit    se.fit residual.scale        UL        LL PredictedProb
1 200.0000 3.3899    1 -0.8114870 0.5147714              1 0.5492064 0.1393812     0.3075737
2 206.0606 3.3899    1 -0.7977632 0.5090986              1 0.5498513 0.1423880     0.3105042
3 212.1212 3.3899    1 -0.7840394 0.5034491              1 0.5505074 0.1454429     0.3134499
4 218.1818 3.3899    1 -0.7703156 0.4978239              1 0.5511750 0.1485460     0.3164108
5 224.2424 3.3899    1 -0.7565919 0.4922237              1 0.5518545 0.1516973     0.3193867
6 230.3030 3.3899    1 -0.7428681 0.4866494              1 0.5525464 0.1548966     0.3223773
```

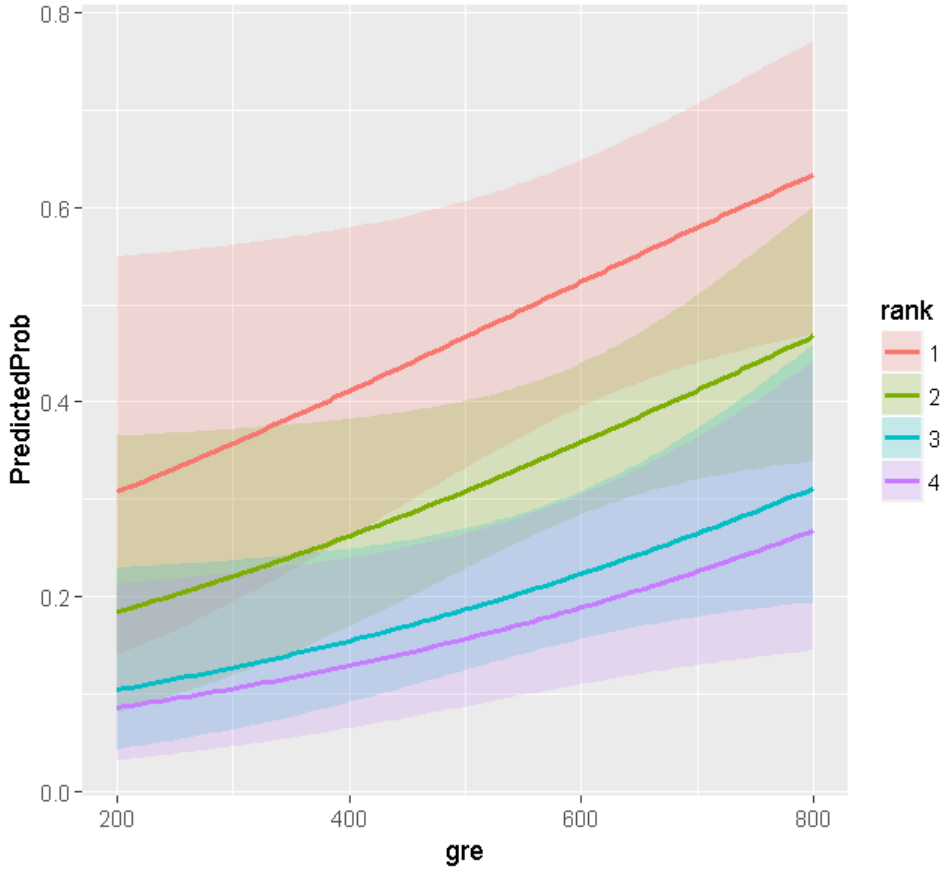
Create 100 values of **gre** between 200 and 800, at each value of **rank** (i.e., 1, 2, 3, and 4) and plot.

http://www.ats.ucla.edu/stat/r/dae/logit.htm

Plot the Predicted Probabilities

```
> library(ggplot2)
> ggplot(newdata3, aes(x = gre, y = PredictedProb)) +
+   geom_ribbon(aes(ymin = LL, ymax = UL, fill = rank), alpha = .2) +
+   geom_line(aes(colour = rank), size=1)
```

Plot the predicted probabilities and 95% confidence intervals to understand and/or present the model.



http://www.ats.ucla.edu/stat/r/dae/logit.htm

Measure the Model Fits

```
> # the difference in deviance for the two models (i.e., the test statistic)
> with(mylogit, null.deviance - deviance)
[1] 41.45903
> with(mylogit, df.null - df.residual)
[1] 5
> #the p-value
> with(mylogit, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
[1] 7.578194e-08
> # the model's log likelihood
> logLik(mylogit)
'log Lik.' -229.2587 (df=6)
```

- One measure of model fit is the significance of the overall model: whether the model with predictors fits significantly better than a model with just an intercept (i.e., a null model).
- The test statistic is the difference between the residual deviance for the model with predictors and the null model.
- The chi-square of 41.46 with 5 degrees of freedom and an associated p-value of less than 0.001 tells us that our model as a whole fits significantly better than an empty model.

http://www.ats.ucla.edu/stat/r/dae/logit.htm

Analysis of Deviance Table

```
> anova(mylogit, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: admit

Terms added sequentially (first to last)

     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                    399     499.98
gre   1  13.9204       398     486.06 0.0001907 ***
gpa   1   5.7122       397     480.34 0.0168478 *
rank  3  21.8265       394     458.52 7.088e-05 ***
---
Signif. codes:  0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1
```

Use **anova** function to give an analysis of deviance table, or the **drop1** function to try dropping each factor.

```
> drop1(mylogit, test="Chisq")
Single term deletions

Model:
admit ~ gre + gpa + rank
       Df Deviance    AIC     LRT  Pr(>Chi)
<none>      458.52 470.52
gre     1   462.88 472.88  4.3578   0.03684 *
gpa     1   464.53 474.53  6.0143   0.01419 *
rank    3   480.34 486.34 21.8265 7.088e-05 ***
---
Signif. codes:  0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1
```

Things to Consider

- **Empty cells or small cells**: check the crosstab between categorical predictors and the outcome variable. If a cell has very few cases (a small cell), the model may become unstable or it might not run at all.

- **Separation or quasi-separation** (also called perfect prediction), a condition in which the outcome does not vary at some levels of the independent variables. See
  http://www.ats.ucla.edu/stat/mult_pkg/faq/general/complete_separation_logit_models.htm

- **Sample size**: Both logit and probit models require more cases than OLS regression because they use maximum likelihood estimation techniques.

- **Pseudo-R-squared**: none of psuedo-R-squared measures can be interpreted exactly as R-squared in OLS regression is interpreted. See
  http://www.ats.ucla.edu/stat/mult_pkg/faq/general/Psuedo_RSquareds.htm

- **Diagnostics**: The diagnostics for logistic regression are different from those for OLS regression. See Hosmer and Lemeshow (2000, Chapter 5).

http://www.ats.ucla.edu/stat/r/dae/logit.htm

共線性 (Collinearity)

- **What is the multicollinearity (collinearity)**
  - it is a statistical phenomenon in which two or more predictor variables in a multiple regression model are highly correlated.
  - one predictor can be linearly predicted from the others with a non-trivial degree of accuracy.

- **How problematic is multicollinearity?**
  - Moderate multicollinearity may not be problematic.
  - Severe multicollinearity can increase the variance of the coefficient estimates and make the estimates very sensitive to minor changes in the model:
    - the coefficient estimates are unstable (may be to switch signs) and difficult to interpret, or
    - parameter estimates may include substantial amounts of uncertainty,
    - forward or backward selection of variables could produce inconsistent results,
    - variance partitioning analyses may be unable to identify unique sources of variation.

變異數膨脹因子
The Variance Inflation Factors

$$X_j = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \varepsilon. \qquad VIF_j = \frac{1}{1 - R_j^2}$$

- A VIF for a single explanatory variable is obtained using the R-squared value of the regression of that variable X_j against all other explanatory variables.

- A VIF measures how much the variance of the estimated regression coefficients are inflated as compared to when the predictor variables are not linearly related.

VIF	Status of predictors
VIF = 1	Not correlated
1 < VIF < 5	Moderately correlated
VIF > 5 to 10	Highly correlated

vif in R

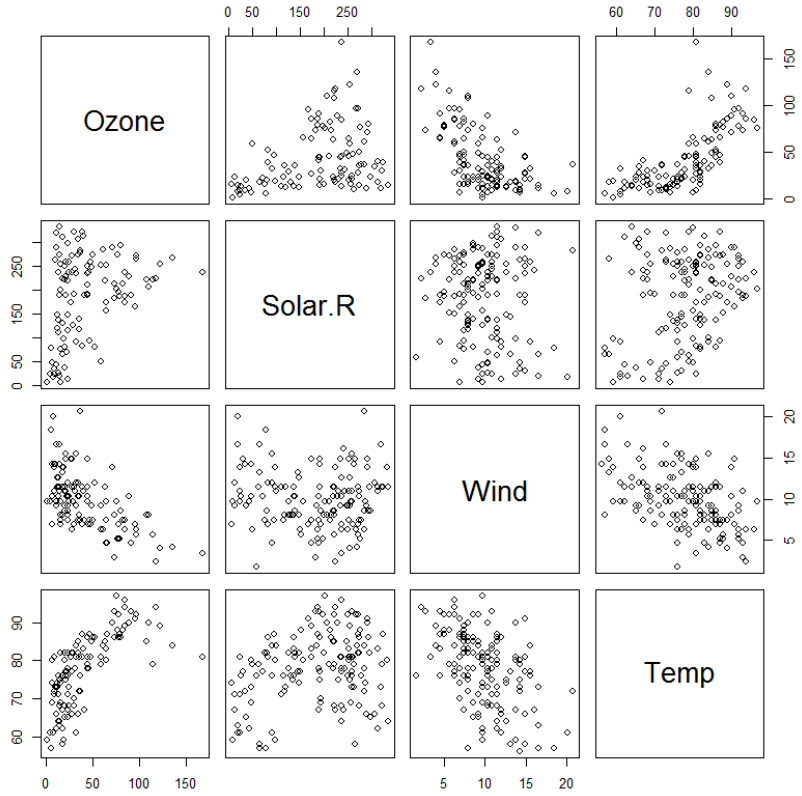
- R packages:
  vif{faraway}, vif{HH}, vif{car}, VIF{fmsb}, vif{VIF}
  - **faraway**: Functions and Datasets for Books by Julian Faraway
  - **HH**: Statistical Analysis and Data Display: Heiberger and Holland
  - **car**: Companion to Applied Regression
  - **fmsb**: Functions for Medical Statistics Book with some Demographic Data
  - **VIF**: A Fast Regression Algorithm For Large Data

```
> head(airquality)
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
>
> model0 <- lm(Ozone ~ Wind + Temp + Solar.R, data=airquality)
```

An Example

```
> cor(airquality[,1:4], use = "pairwise")
             Ozone     Solar.R        Wind       Temp
Ozone    1.0000000  0.34834169 -0.60154653  0.6983603
Solar.R  0.3483417  1.00000000 -0.05679167  0.2758403
Wind    -0.6015465 -0.05679167  1.00000000 -0.4579879
Temp     0.6983603  0.27584027 -0.45798788  1.0000000
> pairs(airquality[,1:4])
```

```
> library(car)
> vif(model0)
    Wind     Temp  Solar.R
1.329070 1.431367 1.095253
```



The Stepwise VIF Selection

```
> summary(model0)
Call:
lm(formula = Ozone ~ Wind + Temp + Solar.R, data = airquality)

Residuals:
    Min      1Q  Median      3Q     Max 
-40.485 -14.219  -3.551  10.097  95.619 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -64.34208   23.05472  -2.791  0.00623 ** 
Wind         -3.33359    0.65441  -5.094 1.52e-06 ***
Temp          1.65209    0.25353   6.516 2.42e-09 ***
Solar.R       0.05982    0.02319   2.580  0.01124 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.18 on 107 degrees of freedom
  (42 observations deleted due to missingness)
Multiple R-squared:  0.6059,	Adjusted R-squared:  0.5948 
F-statistic: 54.83 on 3 and 107 DF,  p-value: < 2.2e-16
```

```
> library(fmsb)
> model1 <- lm(Wind ~ Temp + Solar.R, data=airquality)
> model2 <- lm(Temp ~ Wind + Solar.R, data=airquality)
> model3 <- lm(Solar.R ~ Wind + Temp, data=airquality)
> # checking multicolinearity for independent variables.
> VIF(model0)
[1] 2.537392
> sapply(list(model1, model2, model3), VIF)
[1] 1.267492 1.367450 1.089300
```

Linear model: : large p small n

The linear model

The standard linear regression model can be written as

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_j x_{ji} + \varepsilon_i \qquad (1)$$

Where

y_i =Dependent variable value for subject i

x_{ji} =Independent variable j value for subject i

β_0 =Intercept

β_j =Coefficient for independent variable j

ε_i =Error for subject i

least squares (LS) estimator is typically used.

$$e^2 = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

What happen if $n << p$?

$$y \sim x_1 + x_2 + \ \ldots + x_p, \qquad if\ n = 10, 100, 1000$$

Statistical challenges of high-dimensional data

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A

Phil. Trans. R. Soc. A (2009) **367**, 4237–4253
doi:10.1098/rsta.2009.0159

INTRODUCTION

Statistical challenges of high-dimensional data

BY IAIN M. JOHNSTONE[1] AND D. MICHAEL TITTERINGTON[2,*]

[1]*Department of Statistics, Stanford University, Stanford, CA 94305, USA*
[2]*Department of Statistics, University of Glasgow, Glasgow G12 8QQ, UK*

$$\{x_i, y_i; i = 1, \ldots, n\}$$

$$y = X\beta + \epsilon \qquad \epsilon_i \sim N(0, \sigma^2)$$

$$E(y_i | x_i) = \beta_1 + \beta_2 x_i$$

$n \times p$ so-called design matrix X

$$\hat{\beta} = \arg\min_{\beta} \sum_i (y_i - \beta_1 - \beta_2 x_i)^2$$

$$\hat{\beta} = \arg\min_{\beta} \| y - X\beta \|_2^2.$$

$y \sim N_n(X\beta, \sigma^2 I)$

N_n now denotes an n-variate multi-variate Gaussian distribution

$$p(y|X, \beta) = \{\sqrt{(2\pi\sigma^2)}\}^{-n/2} \exp\left\{ -\frac{\| y - X\beta \|_2^2}{2\sigma^2} \right\}.$$

Regression: large p small n

$$p(y|X,\beta) = \{\sqrt{(2\pi\sigma^2)}\}^{-n/2} \exp\left\{-\frac{\|y - X\beta\|_2^2}{2\sigma^2}\right\}.$$

$$\hat{\beta} = \arg\max_{\beta} p(y|X,\beta).$$

$$\hat{\beta} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y, \quad \textit{provided that } X^{\mathrm{T}}X \textit{ can be inverted.}$$

if the model is correct,

$$\hat{\beta} \sim N_p(\beta, \sigma^2(X^{\mathrm{T}}X)^{-1}),$$

we must have $p \le n$, otherwise $(X^{\mathrm{T}}X)$ is singular and the parameters in the regression model cannot be uniquely estimated.

What if $p > n$ or even $p \gg n$

to side-stepping the singularity of $(X^{\mathrm{T}}X)$

$$S_{\lambda_2} = (X^{\mathrm{T}}X + \lambda_2 I)^{-1}$$

$$\hat{\beta}_{\mathrm{R}} = S_{\lambda_2} X^{\mathrm{T}} y,$$

λ_2 is called a ridge parameter

regularization.

ridge regression

(Hoerl & Kennard 1970)

penalized least squares

penalized maximum likelihood.

Furthermore, in the general maximum-likelihood contexts, the asymptotic theory breaks down.

Regression: large p small n

However, although invertible,

$X^{\mathrm{T}}X + \lambda_2 I$ is $p \times p$ and still potentially a very large matrix.

to seek a solution for β in which

many of the elements are zero. Lasso (Tibshirani 1996).

sparsity

(i) $\hat{\beta}_{\mathrm{L}} = \arg\min_{\beta}\{\|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1\}$, for some λ_1,
(ii) $\hat{\beta}_{\mathrm{L}}$ minimizes $\|y - X\beta\|_2^2$ subject to $\|\beta\|_1 \le c_1(\lambda_1)$, and
(iii) $\hat{\beta}_{\mathrm{L}}$ minimizes $\|\beta\|_1$ subject to $\|y - X\beta\|_2^2 \le b_1(\lambda_1)$.

Regression: large p small n

"LASSO" stands for Least Absolute Shrinkage and Selection Operator

J. R. Statist. Soc. B (1996)
58, *No.* 1, *pp.* 267–288

Regression Shrinkage and Selection via the Lasso

By ROBERT TIBSHIRANI†
University of Toronto, Canada

[Received January 1994. Revised January 1995]

Regression shrinkage and selection via the lasso
R Tibshirani - Journal of the Royal Statistical Society: Series B …, 1996 - Wiley Online Library
We propose a new method for estimation in linear models. The 'lasso'minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. Because of the nature of this constraint it tends to produce some coefficients that …
☆ 99 被引用 37573 次 相關文章 全部共 55 個版本

Open Access Medical Statistics
Dovepress
open access to scientific and medical research
Open Access Full Text Article
REVIEW

High-dimensional data and linear models: a review

This article was published in the following Dove Press journal:
Open Access Medical Statistics
6 August 2014
Number of times this article has been viewed

M Brimacombe
Department of Biostatistics,
University of Kansas Medical Center,
Kansas City, KS, USA

Abstract: The need to understand large database structures is an important
and medical science. This review paper is aimed at quantitative medical re
for guidance in modeling large numbers of variables in medical research, h
standard linear models and the geometry that underlies their analysis. Issues

Practical Assessment, Research & Evaluation
A peer-reviewed electronic journal.

Volume 21, Number 7, May 2016 ISSN 1531-7714

Regularization Methods for Fitting Linear Models with Small Sample Sizes: Fitting the Lasso Estimator using R

W. Holmes Finch, *Ball State University*
Maria E. Hernandez Finch, *Ball State University*

High Dimensional Data and Linear Model

- when $n \ll p$:
  - parameter estimates: can not converge, high variance.
  - **reduce power**: conclude that one or more of x variables are not related to the y, when in fact they are.
  - **collinearity**, or very strong relationship among x variables, leading to biased parameter estimators.
  - not possible to obtain LS estimators.

How to deal with HD data

- **Variable selection methods** (e.g. stepwise regression, best subsets regression)
  - the variable selection methods can produce estimates with inflated standard errors for the coefficients (Hastie, Tibshirani, & Friedman, 2009)
- **Dimension reduction techniques** (e.g. principal components regression, supervised principal components regression, and partial least squares regression).
  - Dimension reduction models combine the independent variables into a small number of linear combinations, making interpretation of results for individual variables somewhat more difficult, and creating an extra layer of complexity in the model as a whole (Finch, Hernandez Finch, & Moss, 2014).

Regularization, or Shrinkage techniques: alternative parameter estimation algorithms

- **Regularization methods** identify optimal values of the β_j such that the most important independent variables receive higher values, and the least important are assigned coefficients at or near 0.

2. THE LASSO

2.1. *Definition*

Suppose that we have data $(\mathbf{x}^i, y_i)$, $i = 1, 2, \ldots, N$, where $\mathbf{x}^i = (x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$ are the predictor variables and y_i are the responses. As in the usual regression set-up, we assume either that the observations are independent or that the y_is are conditionally independent given the x_{ij}s. We assume that the x_{ij} are standardized so that $\Sigma_i x_{ij}/N = 0$, $\Sigma_i x_{ij}^2/N = 1$.

Letting $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)^{\mathrm{T}}$, the lasso estimate $(\hat{\alpha}, \hat{\beta})$ is defined by

$$(\hat{\alpha}, \hat{\beta}) = \arg\min \left\{ \sum_{i=1}^{N} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 \right\} \qquad \text{subject to} \sum_j |\beta_j| \leqslant t. \qquad (1)$$

Here $t \geqslant 0$ is a tuning parameter. Now, for all t, the solution for α is $\hat{\alpha} = \bar{y}$. We can assume without loss of generality that $\bar{y} = 0$ and hence omit α.

Computation of the solution to equation (1) is a quadratic programming problem with linear inequality constraints. We describe some efficient and stable algorithms for this problem in Section 6.

The lasso

$$e^2 = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p}\left|\hat{\beta}_j\right| \quad (3)$$

- **Regularization** methods have in common the application of a penalty to the LS estimator in regression model.
- Tuning parameter λ: control the amount of shrinkage (i.e. the degree to which the relationship of the independent variables to the dependent variable are down weighted or removed from the model).
  - Larger λ values: greater shrinkage of the model; i.e. a greater reduction in the number of independent variables that are likely to be included in the final model.
  - A λ of 0 leads to the LS estimator.

the optimal λ is the one that minimizes the leave-one-out MSE value calculated using equation (4).

$$MSE_{k\lambda} = \frac{\sum_{i=1}^{N}(y_{ik} - \hat{y}_{ik\lambda})^2}{N_k} \quad (4)$$

Where

y_i =Dependent variable value for subject i in test set k

$\hat{y}_{ik\lambda}$ = Model predicted dependent variable value for subject i in test set k using λ

The lasso

- The least squares estimator is known to have **low bias** in many situations, but can also have relatively **large variance**, particularly in the context of high dimensional data; i.e. relatively many predictors and few observations (Loh & Wainwright, 2012).

- The **lasso** has been found to have somewhat greater bias than the standard least squares estimator, but with lower variance, particularly in the high dimensional case (Hastie, Tibshirani, & Wainwright, 2015).

Lasso Regression & Ridge Regression

Lasso regression:

- 可同時進行變數篩選與複雜度調整(正規化,避免overfitting) 。
- 懲罰項為L1 norm。
- 適合資料特徵為高維度稀疏資料。

Ridge regression:

- 使用L2 周norm 正則化，避免過度配適。
- 適合資料特徵為低維度稠密資料。

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \Big)^2,$$

$$\text{subject to} \sum_{j=1}^{p} \beta_j^2 \le t,$$

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{N} \big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \big)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \Big\}.$$

R packages for lasso

- **`biglass`**: Extending Lasso Model Fitting to Big Data
- islasso: The Induced Smoothed Lass
- **`HDCI`**: High Dimensional Confidence Interval Based on Lasso and Bootstrap
- **`glmnet`**: Lasso and Elastic-Net Regularized Generalized Linear Models
- **`lars`**: Least Angle Regression, Lasso and Forward Stagewise

lasso Approach for Fitting Linear Models

- The data were collected on 10 adults with autism (自閉症) who were clients of an autism research and service provision center at a large Midwestern university.
- Adults identified with autism represent a particularly difficult population from which to sample, meaning that quite frequently sample sizes are small.
- **Sample**: 10 adults (9 males), with a mean age of 20 years, 2 months (SD=1 year, 9.6 months).
- **Interest**: the relationship between executive functioning (16 independent variables) as measured by the Delis-Kaplan Executive Functioning System (DKEFS; Delis, Kaplan, & Kramer, 2001) and the full scale intelligence score (FSIQ) (dependent variable)on the Wechsler Adult Intelligence Scale, 4th edition (WAIS-IV; Wechsler, 2008).

Finch and Finch: Regularization Methods for Fitting Linear Models with Small Sampl

Practical Assessment, Research & Evaluation

A peer-reviewed electronic journal.

Copyright is retained by the first or sole author, who grants right of first publication to *Practical Assessment, Research & Evaluation*. Permission is granted to distribute this article for nonprofit, educational purposes if it is copied in its entirety and the journal is credited. PARE has the right to authorize third party reproduction of this article in print, electronic and database forms.

Volume 21, Number 7, May 2016 ISSN 1531-7714

Regularization Methods for Fitting Linear Models with Small Sample Sizes: Fitting the Lasso Estimator using R

W. Holmes Finch, *Ball State University*
Maria E. Hernandez Finch, *Ball State University*

Independent Variables

Variable
Visual scanning
Number sequencing
Letter sequencing
Number-letter sequencing
Motor speed
Letter fluency
Category fluency
Category switching
Category switching accuracy
Filled dots
Empty dots
Dots switching
Color naming
Word reading
Inhibition
Inhibition/switching

More simulation examples:
https://www4.stat.ncsu.edu/~post/josh/LASSO_Ridge_Elastic_Net_-_Examples.html

Boston Housing Data (BostonHousing, 506x14, y=medv)

```
> keras::dataset_boston_housing()  #Boston housing price regression dataset
> MASS::Boston #Housing Values in Suburbs of Boston
> mlbench::BostonHousing #Boston Housing Data
```

資料原調查目的: 估計波士頓居民為了提高空氣品質而願意支付額外費用的傾向

The original data are 506 observations on 14 variables, medv being the target variable:

- **crim**: per capita crime rate by town (人均犯罪率/鎮)
- **zn**: proportion of residential land zoned for lots over 25,000 sq.ft
- **indus**: proportion of non-retail business acres per town (非零售商業用地所佔的百分比/鎮)
- **chas**: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- **nox**: nitric oxides concentration (parts per 10 million) (氮氧化合物濃度)
- **rm**: average number of rooms per dwelling (平均房間數/寓所)
- **age**: proportion of owner-occupied units built prior to 1940 (1940年前建的自有住房所
- **dis**: weighted distances to five Boston employment centres (到達5個波士就業中心的平
- **rad**: index of accessibility to radial highways
- **tax**: full-value property-tax rate per USD 10,000 (全額不動產稅率)
- **ptratio**: pupil-teacher ratio by town (學生教師比/鎮)
- **b**: 1000(B - 0.63)^2 where B is the proportion of blacks by town
- **lstat**: percentage of lower status of the population (低社會地位人口百分比)
- **medv**: median value of owner-occupied homes in USD 1000's (自有住房數的中位數價格)

```
dataset_boston_housing(
  path = "boston_housing.npz",
  test_split = 0.2,
  seed = 113L
)
```

Value: Lists of training and test data: `train$x, train$y, test$x, test$y`.

The corrected data set (BostonHousing2) has the following additional columns:

- **cmedv**: corrected median value of owner-occupied homes in USD 1000's
- **town**: name of town
- **tract**: census tract (人口普查區)
- **lon**: longitude of census tract
- **lat**: latitude of census tract

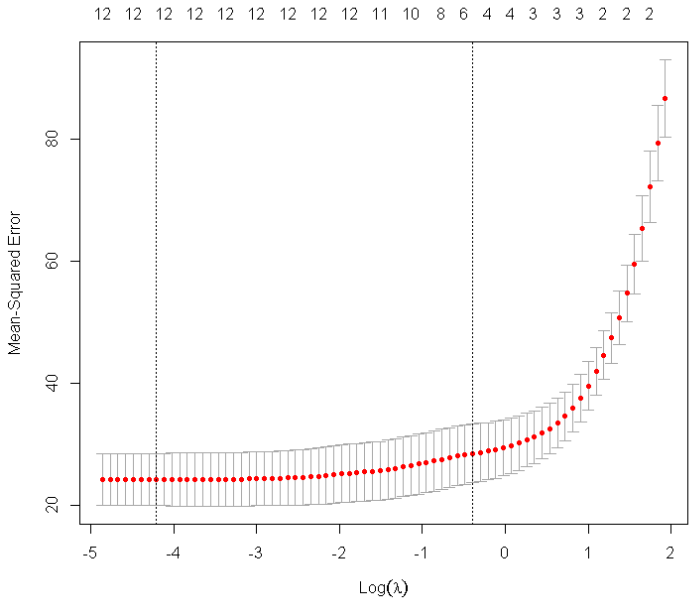
```
> BHdata <- dataset_boston_housing()
> data(Boston, package="MASS")
> head(Boston)
```

Boston Housing Data (BostonHousing, 506x14, y=medv)

```
> library(glmnet)
> library(caret)
> set.seed(123)
> mydata <- MASS::Boston
> head(mydata)
     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
...
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
> y.var <- "medv"
> x.var <- names(mydata)[! names(mydata) == y.var]
> # prepare training/testing dataset
> id <- createDataPartition(mydata[ , y.var], p=0.8, list=F)
> xtrain <- as.matrix(mydata[id, x.var])
> ytrain <- as.matrix(mydata[id, y.var])
> xtest <- as.matrix(mydata[-id, x.var])
> ytest <-as.matrix( mydata[-id, y.var])
> lapply(list(xtrain, ytrain, xtest, ytest), dim)
[[1]]
[1] 407  13

[[2]]
[1] 407   1

[[3]]
[1] 99 13

[[4]]
[1] 99  1
```

lasso: Boston Housing Data

```
> # Cross-validation to determine optimal value of lambda
> lasso.cv <- cv.glmnet(xtrain, ytrain, family="gaussia
> coef(lasso.cv)
14 x 1 sparse Matrix of class "dgCMatrix"
                      1
(Intercept) 13.995490487
crim         .
zn           .
indus        .
chas         0.416641462
nox          .
rm           4.238545370
age          .
dis          .
rad          .
tax          .
ptratio     -0.731155293
black        0.005107381
lstat       -0.498953947
> # The vertical lines show the locations of λmin and λ1se.
> # The two different values of λ reflect two common choices for λ.
> # The λmin is the one which minimizes out-of-sample loss in CV.
> # The λ1se is the one which is the largest λ value within 1
> #     standard error of λmin.
> # The numbers across the top are the  #nonzero coefficient estimates.
> plot(lasso.cv)
>
> best.lambda <- lasso.cv$lambda.min
> best.lambda
[1] 0.01488213
```



lasso: Boston Housing Data

```
> lasso.model <- glmnet(xtrain, ytrain, family = "gaussian",
+                       alpha = 1, lambda = best.lambda)
>
> coef(lasso.model)
14 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept)  36.438819403
crim         -0.089569526
zn            0.047441561
indus        -0.009331238
chas          2.276674852
nox         -16.510578478
rm            3.826013498
age           .
dis          -1.574095619
rad           0.274331319
tax          -0.011933631
ptratio      -0.945572072
black         0.009593514
lstat        -0.529185530
>
> # prediction
> yhat <- predict(lasso.model, xtest)
>
> # evaluation
> mse <- mean((ytest - yhat) ^ 2)
> mae <- MAE(ytest, yhat)
> rmse <- RMSE(ytest, yhat)
> r2 <- R2(ytest, yhat, form = "traditional")
```

```
> cat(" MAE:", mae, "\n", "MSE:", mse, "\n",
+     "RMSE:", rmse, "\n", "R-squared:", r2)
 MAE: 3.714517
 MSE: 24.87917
 RMSE: 4.987902
 R-squared: 0.5543279
>
> # plot prediction
> plot(yhat, ytest,
+ main="medv vs pred-medv", asp=1)
> abline(a=0, b=0.5, col="blue")
> abline(lm(ytest ~ yhat), col="red")
```