# Regression

May 5, 2016

## Contents

ALL MODELS ARE WRONG, BUT SOME MODELS ARE USEFUL!

In many ways, regression analysis lives at the heart of statistics. It's a broad term for a set of methodologies used to predict a response variable (also called a dependent, criterion, or outcome variable) from one or more predictor variables (also called independent or explanatory variables). In general, regression analysis can be used to identify the explanatory variables that are related to a response variable, to describe the form of the relationships involved, and to provide an equation for predicting the response variable from the explanatory variables.

# 1 OLS regression

## Linear versus nonlinear models

Are the following models linear regression models? Why or why not?

$$
\begin{aligned}
Y_i &= e^{\beta_1+\beta_2 X_i+\varepsilon_i} && (1)\\
Y_i &= \frac{1}{e^{\beta_1+\beta_2 X_i+\varepsilon_i}} && (2)\\
Y_i &= \beta_1 + (0.75-\beta_1)e^{-\beta_2(X_i-2)} + \varepsilon_i && (3)\\
Y_i &= \beta_1 + \beta_2^3 X_i + \varepsilon_i && (4)\\
Y_i &= \beta_1 + \beta_2(\frac{1}{X_i}) + \varepsilon_i && (5)
\end{aligned}
$$

## 1.1 Simple linear regression

### 1.1.1 Methodology

Consider the following model:

$$y_i = \beta_0 + \beta_1 x_i + u_i, \quad i = 1, 2 \ldots, n$$

### 1.1.2 CLRM

The Gaussian, standard, or classical linear regression model (CLRM), which is the cornerstone of most econometric theory, makes 10 assumptions.

- Assumption 1: Linear regression model. The regression model is linear in the parameters

$$y_i = \beta_0 + \beta_1 x_i + u_i$$

- Assumption 2: X values are fixed in repeated sampling. Values taken by the regressor X are considered fixed in repeated samples. More technically, X is assumed to be nonstochastic.

- Assumption 3: Zero mean value of disturbance $u_i$. Given the value of X, the mean, or expected, value of the random disturbance term ui is zero. Technically, the conditional mean value of $u_i$ is zero. Symbolically, we have

$$E(u_i|X_i) = 0$$

- Assumption 4: Homoscedasticity or equal variance of $u_i$. Given the value of X, the variance of $u_i$ is the same for all observations. That is, the conditional variances of $u_i$ are identical. Symbolically, we have

$$\text{var}(u_i|X_i) = E[u_i - E(u_i|X_i)]^2 = E(u_i^2|X_i) = \sigma^2$$

- Assumption 5: No autocorrelation between the disturbances. Given any two X values, $X_i$ and $X_j (i = j)$, the correlation between any two $u_i$ and $u_j (i = j)$ is zero. Symbolically,

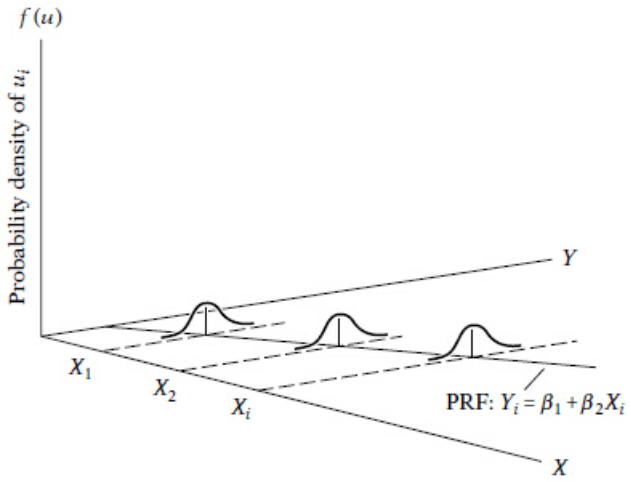



Figure 1: Homoscedasticity

$$\text{cov}(u_i, u_j|X_i, X_j) = E(u_i|X_i)(u_j|X_j) = 0$$

- Assumption 6: Zero covariance between $u_i$ and $X_i$, or $E(u_i X_i) = 0$. Formally,

$$\text{cov}(u_i, X_i) = E([u_i - E(u_i)][X_i - E(X_i)]) = E(u_i X_i) = 0$$

- Assumption 7: The number of observations n must be greater than the number of parameters to be estimated. Alternatively, the number of observations n must be greater than the number of explanatory variables.
- Assumption 8: Variability in X values. The X values in a given sample must not all be the same. Technically, var (X) must be a finite positive number.
- Assumption 9: The regression model is correctly specified. Alternatively, there is no specification bias or error in the model used in empirical analysis.
- Assumption 10: There is no perfect multicollinearity. That is, there are no perfect linear relationships among the explanatory variables.

### 1.1.3 Empirical analysis using R

```
options(digits=3)
fit <- lm(weight ~ height, data = women)
summary(fit)
```

```
##
## Call:
## lm(formula = weight ~ height, data = women)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.733 -1.133 -0.383  0.742  3.117
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -87.5167     5.9369   -14.7  1.7e-09 ***
```

```
## height        3.4500     0.0911    37.9  1.1e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.53 on 13 degrees of freedom
## Multiple R-squared:  0.991,Adjusted R-squared:  0.99
## F-statistic: 1.43e+03 on 1 and 13 DF,  p-value: 1.09e-14
```

```
coefficients(fit)  #gives the coefficients (estimated parameters) from the model
```

```
## (Intercept)      height
##      -87.52        3.45
```

```
fitted(fit)
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
## 113 116 119 123 126 130 133 137 140 144 147 151 154 157 161
```

```
residuals(fit)     #the residuals, that is response minus fitted values
```

```
##       1       2       3       4       5       6       7       8       9
##  2.4167  0.9667  0.5167  0.0667 -0.3833 -0.8333 -1.2833 -1.7333 -1.1833
##      10      11      12      13      14      15
## -1.6333 -1.0833 -0.5333  0.0167  1.5667  3.1167
```

```
deviance(fit)
```

```
## [1] 30.2
```

```
confint(fit,level=0.99)
```

```
##               0.5 % 99.5 %
## (Intercept) -105.40 -69.63
## height         3.18   3.72
```

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


The plot aboved suggests that you might be able to improve your prediction using a regression with a quadratic term (that is, $X^2$).

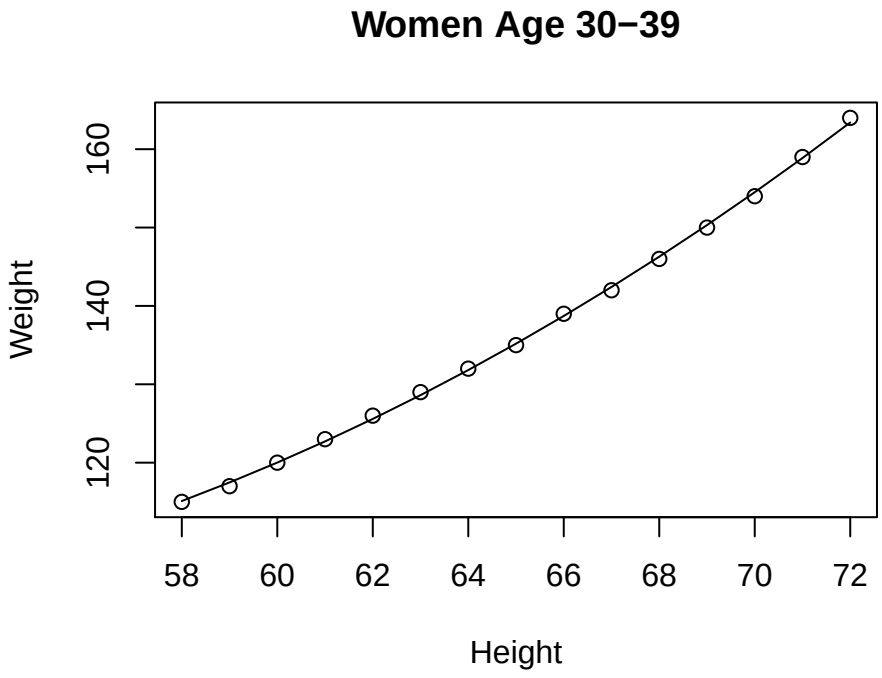
| | |
|---|---|
| `~` | Separates response variables on the left from the explanatory variables on the right. For example, a prediction of `y` from `x`, `z`, and `w` would be coded `y ~ x + z + w`. |
| `+` | Separates predictor variables. |
| `:` | Denotes an interaction between predictor variables. A prediction of `y` from `x`, `z`, and the interaction between `x` and `z` would be coded `y ~ x + z + x:z`. |
| `*` | A shortcut for denoting all possible interactions. The code `y ~ x * z * w` expands to `y ~ x + z + w + x:z + x:w + z:w + x:z:w`. |
| `^` | Denotes interactions up to a specified degree. The code `y ~ (x + z + w)^2` expands to `y ~ x + z + w + x:z + x:w + z:w`. |
| `.` | A place holder for all other variables in the data frame except the dependent variable. For example, if a data frame contained the variables `x`, `y`, `z`, and `w`, then the code `y ~ .` would expand to `y ~ x + z + w`. |
| `-` | A minus sign removes a variable from the equation. For example, `y ~ (x + z + w)^2 - x:w` expands to `y ~ x + z + w + x:z + z:w`. |
| `-1` | Suppresses the intercept. For example, the formula `y ~ x -1` fits a regression of `y` on `x`, and forces the line through the origin at `x=0`. |
| `I()` | Elements within the parentheses are interpreted arithmetically. For example, `y ~ x + (z + w)^2` would expand to `y ~ x + z + w + z:w`. In contrast, the code `y ~ x + I((z + w)^2)` would expand to `y ~ x + h`, where `h` is a new variable created by squaring the sum of `z` and `w`. |
| *function* | Mathematical functions can be used in formulas. For example, `log(y) ~ x + z + w` would predict `log(y)` from `x`, `z`, and `w`. |

Figure 2: Symbols commonly used in R formulas

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

```
##
## Call:
## lm(formula = weight ~ height + I(height^2), data = women)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5094 -0.2961 -0.0094  0.2862  0.5971
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 261.87818   25.19677   10.39  2.4e-07 ***
## height       -7.34832    0.77769   -9.45  6.6e-07 ***
## I(height^2)   0.08306    0.00598   13.89  9.3e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.384 on 12 degrees of freedom
## Multiple R-squared:  0.999,Adjusted R-squared:  0.999
## F-statistic: 1.14e+04 on 2 and 12 DF,  p-value: <2e-16
```

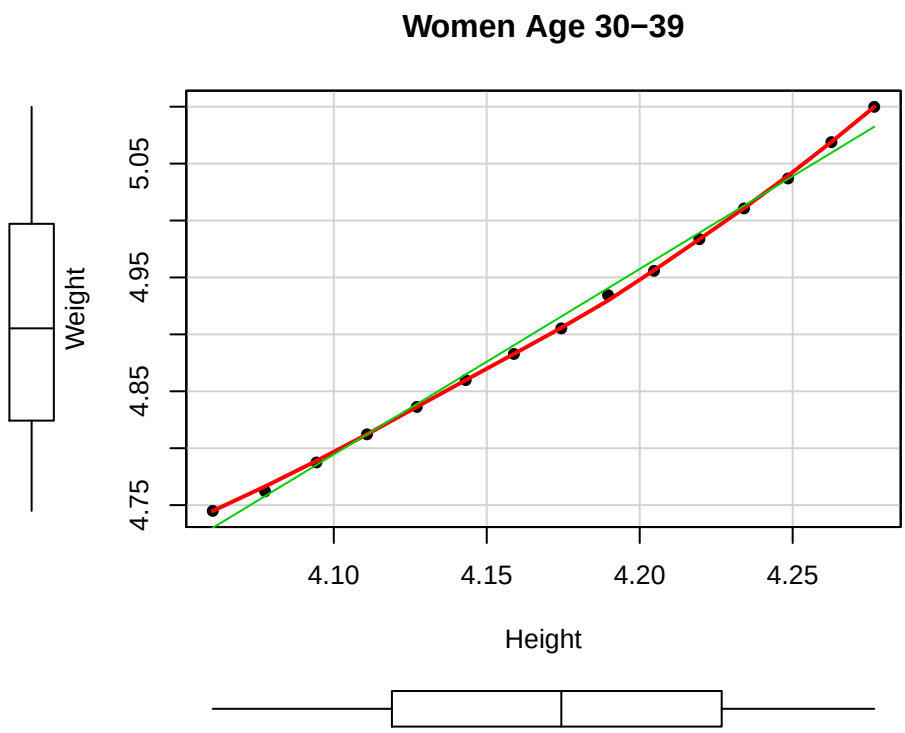
```
plot(women$height, women$weight, main = "Women Age 30-39",
     xlab = "Height", ylab = "Weight")
lines(women$height, fitted(fit2))
```




The scatterplot() function in the car package provides a simple and convenient method of plotting a bivariate relationship. install.packages("car")

```
library(car)
scatterplot(log(weight) ~ log(height), data = women, spread=FALSE,
            pch = 19, main = "Women Age 30-39", xlab = "Height",
```

```
            ylab = "Weight")
```




## 1.2 Outlier

In some applications, especially, but not only, with small data sets, the OLS estimates are sensitive to the inclusion of one or several observations. OLS is susceptible to outlying observations because it minimizes the sum of squared residuals: large residuals (positive or negative) receive a lot of weight in the least squares minimization problem. If the estimates change by a practically large amount when we slightly modify our sample, we should be concerned.

### 1.2.1 An Example

```
Anscombe<-data.frame(
  X =c(10.0, 8.0, 13.0, 9.0, 11.0, 14.0, 6.0, 4.0, 12.0, 7.0, 5.0),
  Y1=c(8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68),
  Y2=c(9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74),
  Y3=c(7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.44, 5.73),
  X4=c(rep(8,7), 19, rep(8,3)),
  Y4=c(6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.50, 5.56, 7.91, 6.89)
)
summary(lm(Y1~X, data=Anscombe))

##
## Call:
## lm(formula = Y1 ~ X, data = Anscombe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1.92127 -0.45577 -0.04136  0.70941  1.83882
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0001     1.1247   2.667  0.02573 *
## X             0.5001     0.1179   4.241  0.00217 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6665,Adjusted R-squared:  0.6295
## F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217
```

```
summary(lm(Y2~X, data=Anscombe))
```

```
##
## Call:
## lm(formula = Y2 ~ X, data = Anscombe)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9009 -0.7609  0.1291  0.9491  1.2691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.001      1.125   2.667  0.02576 *
## X              0.500      0.118   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6662,Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002179
```

```
summary(lm(Y3~X, data=Anscombe))
```

```
##
## Call:
## lm(formula = Y3 ~ X, data = Anscombe)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1586 -0.6159 -0.2325  0.1510  3.2407
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0075     1.1244   2.675  0.02542 *
## X             0.4994     0.1179   4.237  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.666,Adjusted R-squared:  0.6289
## F-statistic: 17.95 on 1 and 9 DF,  p-value: 0.002185
```

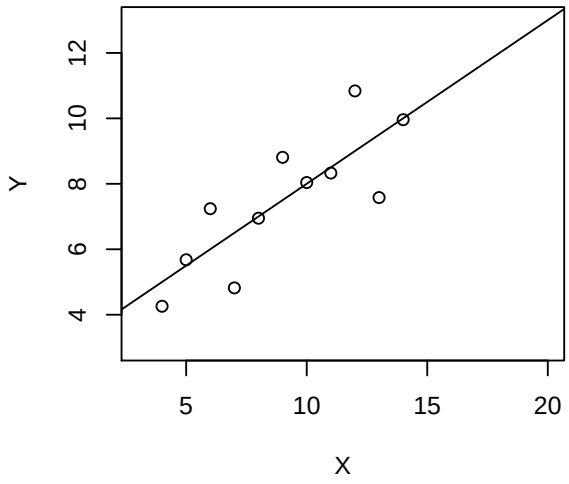
```r
summary(lm(Y4~X4,data=Anscombe))
```

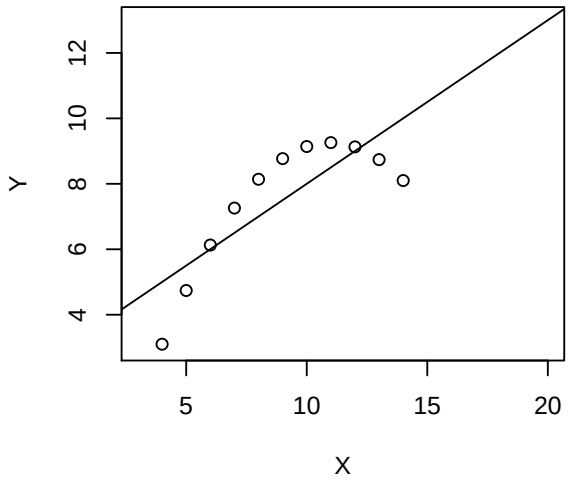
```
##
## Call:
## lm(formula = Y4 ~ X4, data = Anscombe)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.751 -0.831  0.000  0.809  1.839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0017     1.1239   2.671  0.02559 *
## X4            0.4999     0.1178   4.243  0.00216 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6667,Adjusted R-squared:  0.6297
## F-statistic:    18 on 1 and 9 DF,  p-value: 0.002165
```

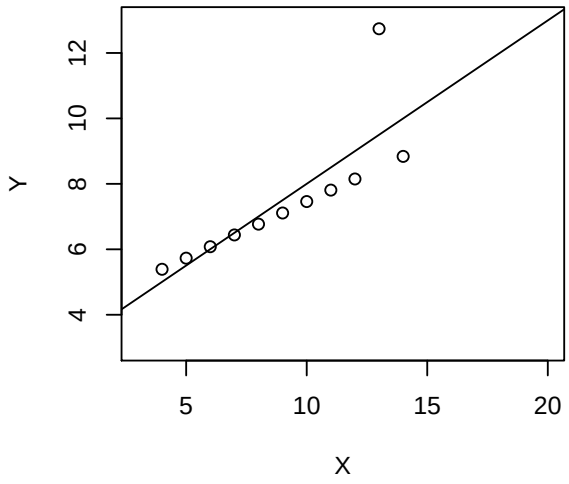
```r
head(Anscombe)
```

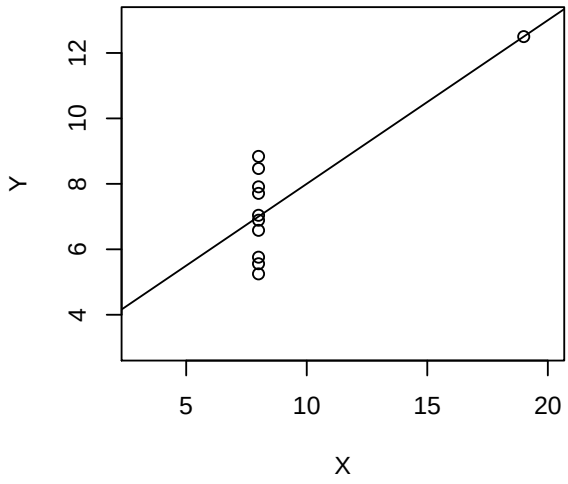
```
##    X   Y1   Y2    Y3 X4   Y4
## 1 10 8.04 9.14  7.46  8 6.58
## 2  8 6.95 8.14  6.77  8 5.76
## 3 13 7.58 8.74 12.74  8 7.71
## 4  9 8.81 8.77  7.11  8 8.84
## 5 11 8.33 9.26  7.81  8 8.47
## 6 14 9.96 8.10  8.84  8 7.04
```

```r
attach(Anscombe)
par(mfrow = c(2,2))
plot(c(3,20), c(3,13), type="n", xlab = "X", ylab = "Y"); points(X,Y1); abline(lm(Y1~X))
plot(c(3,20), c(3,13), type="n", xlab = "X", ylab = "Y"); points(X,Y2); abline(lm(Y2~X))
plot(c(3,20), c(3,13), type="n", xlab = "X", ylab = "Y"); points(X,Y3); abline(lm(Y3~X))
plot(c(3,20), c(3,13), type="n", xlab = "X", ylab = "Y"); points(X4,Y4); abline(lm(Y4~X4))
```



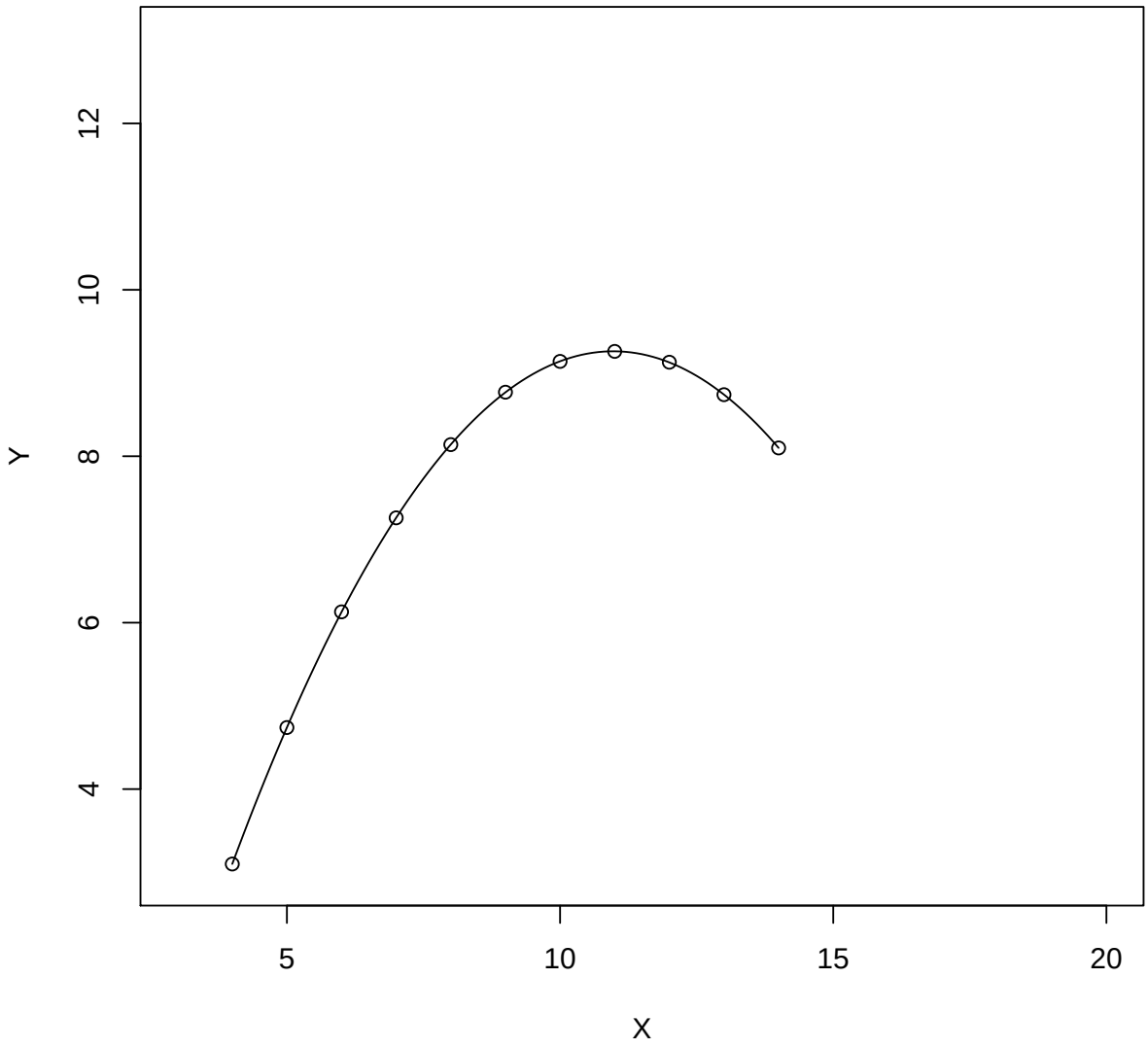




```r
#1 is good
par(mfrow = c(1,1))
#2
X2<-X^2
lm2.sol<-lm(Y2~X+X2)
summary(lm2.sol)

##
## Call:
## lm(formula = Y2 ~ X + X2)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0013287 -0.0011888 -0.0006294  0.0008741  0.0023776
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.9957343  0.0043299   -1385   <2e-16 ***
## X            2.7808392  0.0010401    2674   <2e-16 ***
## X2          -0.1267133  0.0000571   -2219   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001672 on 8 degrees of freedom
## Multiple R-squared:      1,Adjusted R-squared:      1
## F-statistic: 7.378e+06 on 2 and 8 DF,  p-value: < 2.2e-16
```

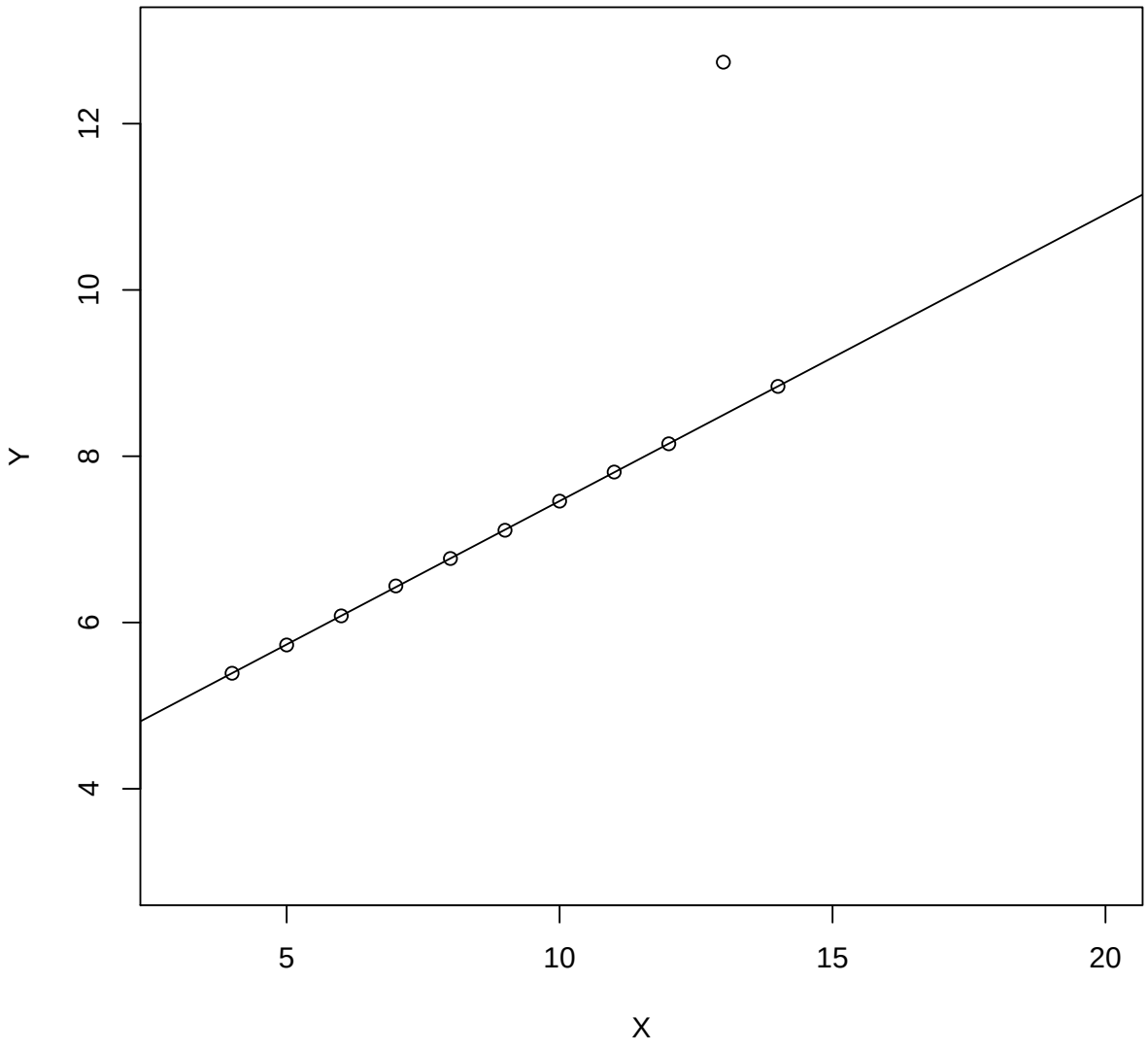
```
x<-seq(min(X), max(X), by=0.1)
b<-coef(lm2.sol)
y<-b[1]+b[2]*x+b[3]*x^2
plot(c(3,20), c(3,13), type="n", xlab = "X", ylab = "Y")
points(X,Y2)
lines(x,y)
```




```r
#3
i<-1:11; Y31<-Y3[i!=3]; X3<-X[i!=3]
lm3.sol<-lm(Y31~X3)
summary(lm3.sol)

##
## Call:
## lm(formula = Y31 ~ X3)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0060173 -0.0012121 -0.0010173 -0.0008225  0.0140693
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.0106277  0.0057115   702.2   <2e-16 ***
## X3          0.3450433  0.0006262   551.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006019 on 8 degrees of freedom
## Multiple R-squared:      1,Adjusted R-squared:      1
## F-statistic: 3.036e+05 on 1 and 8 DF,  p-value: < 2.2e-16
```

```
plot(c(3,20), c(3,13), type="n", xlab = "X", ylab = "Y")
points(X,Y3)
abline(lm3.sol)
```



```
detach(Anscombe)
```

### 1.2.2 Test outlier

#### Dffits

DFFITS is a diagnostic meant to show how influential a point is in a statistical regression. It was proposed in 1980.

$$\text{DFFITS} = \frac{\widehat{y_i} - \widehat{y_{i(i)}}}{s_{(i)}\sqrt{h_{ii}}}\sqrt{h_{ii}/(1-h_{ii})}$$

where $\widehat{y_i}$ and $\widehat{y_{i(i)}}$ are the prediction for point i with and without point i included in the regression, $s_{(i)}$ is the standard error estimated without the point in question, and $h_{ii}$ is the leverage for the point $(\partial\widehat{y_i}/\partial y_i)$.

The authors suggest investigating those points with DFFITS greater than $2\sqrt{\frac{p+1}{n}}$.

```
attach(Anscombe)
p<-1; n<-length(X);d<-dffits(lm(Y3~X, data=Anscombe))
cf<-1:n; cf[d>2*sqrt((p+1)/n)]

## [1] 3

detach(Anscombe)
```

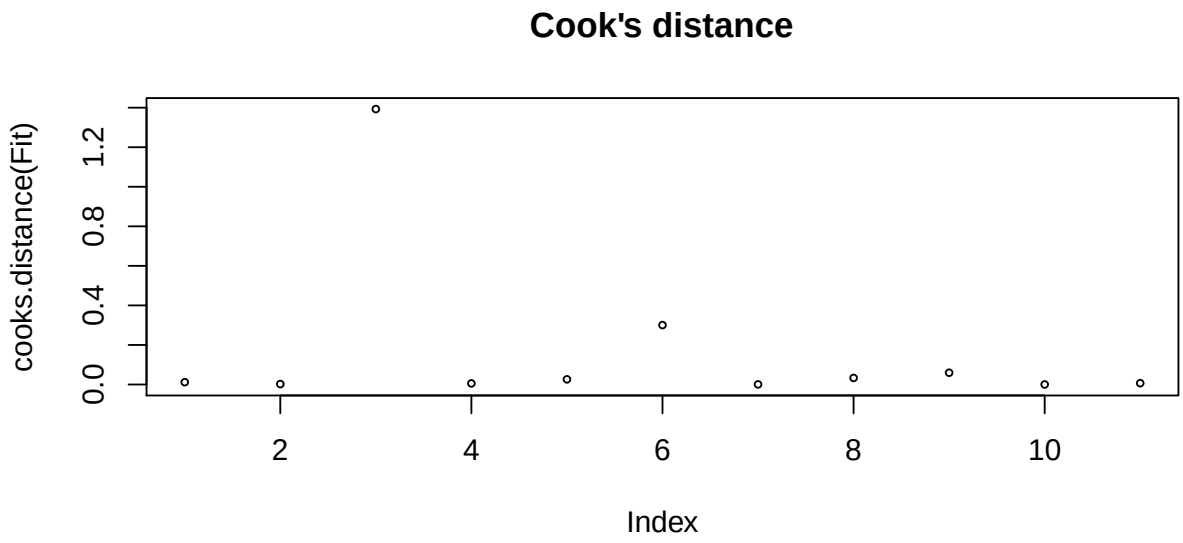
#### Cook's distance

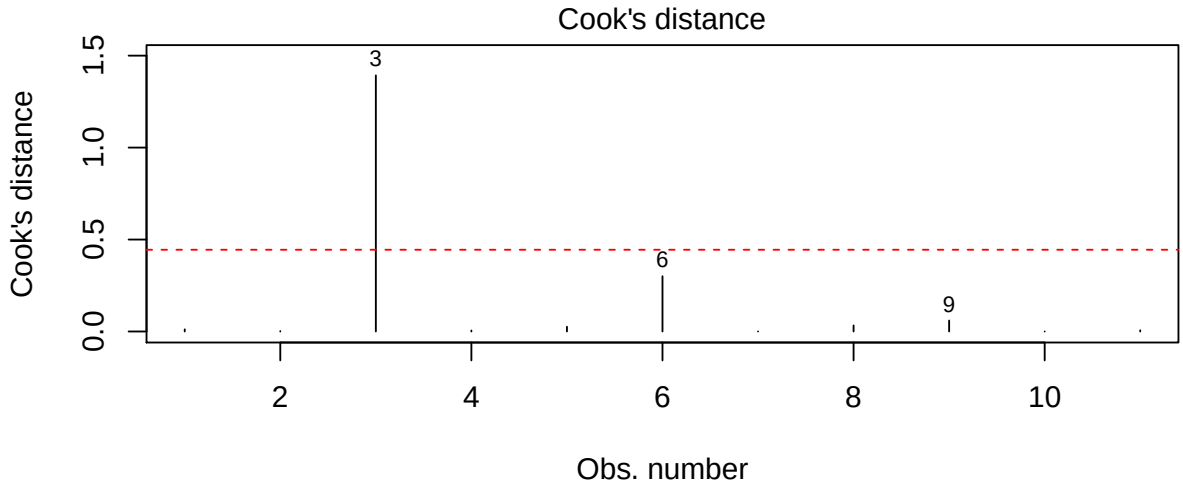
Cook's distance measures the effect of deleting a given observation. Data points with large residuals (outliers) and/or high leverage may distort the outcome and accuracy of a regression. Points with a large Cook's distance are considered to merit closer examination in the analysis.

$$D_i = \frac{(\hat{\beta} - \hat{\beta}^{(-i)})^T (X^T X)(\hat{\beta} - \hat{\beta}^{(-i)})}{(1+p)s^2}$$

where $h_{ii}$ is the leverage, i.e., the i-th diagonal element of the hat matrix $\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$; $e_i$ is the residual (i.e., the difference between the observed value and the value fitted by the proposed model).

```
Fit<-lm(Y3~X, data=Anscombe)
cooks.distance(Fit)

##            1            2            3            4            5
## 0.0118305891 0.0021827101 1.3928277909 0.0055254398 0.0260716064
##            6            7            8            9           10
## 0.3006335925 0.0004804045 0.0331943873 0.0596504117 0.0002176290
##           11
## 0.0067519721

par(mfrow=c(2,1))
plot(cooks.distance(Fit),main="Cook's distance",cex=0.5)
Np<-length(coefficients(Fit))-1
N<-length(fitted(Fit))
CutLevel<-4/(N-Np-1)
plot(Fit,which=4)
abline(CutLevel,0,lty=2,col="red")
```




and so on (COVRATIO)[1].

**Summary**

```
influence.measures(lm(Y3~X, data=Anscombe))

## Influence measures of
##   lm(formula = Y3 ~ X, data = Anscombe) :
##
##       dfb.1_     dfb.X    dffit    cov.r   cook.d    hat inf
## 1  -4.64e-03 -4.43e-02  -0.1468 1.34e+00 0.011831 0.1000
## 2  -3.75e-02  1.88e-02  -0.0624 1.39e+00 0.002183 0.1000
## 3  -1.83e+02  2.69e+02 342.7851 7.36e-10 1.392828 0.2364   *
```

[1]See the details ?covratio, ?hatvalues.

```
## 4  -3.31e-02 -2.66e-18  -0.0997 1.36e+00 0.005525 0.0909
## 5   4.92e-02 -1.17e-01  -0.2197 1.34e+00 0.026072 0.1273
## 6   4.90e-01 -6.67e-01  -0.7898 1.36e+00 0.300634 0.3182
## 7   2.60e-02 -2.01e-02   0.0292 1.53e+00 0.000480 0.1727
## 8   2.39e-01 -2.07e-01   0.2449 1.80e+00 0.033194 0.3182   *
## 9   1.38e-01 -2.32e-01  -0.3365 1.34e+00 0.059650 0.1727
## 10 -1.54e-02  1.05e-02  -0.0197 1.45e+00 0.000218 0.1273
## 11  1.04e-01 -8.62e-02   0.1098 1.64e+00 0.006752 0.2364
```

## 1.3 Maximum likelihood

The probability density function, or pdf, for a random variable, y, conditioned on a set of parameters, $\vartheta$, is denoted f (y | $\vartheta$).1 This function identifies the data-generating process that underlies an observed sample of data and, at the same time, provides a mathematical description of the data that the process will produce. The joint density of n independent and identically distributed (i.i.d.) observations from this process is the product of the individual densities.

$$f(y_1, \ldots y_n|\theta) = \Pi_{i=1}^{n} f(y_i|\theta) = L(\theta|y).$$

This joint density is the likelihood function, defined as a function of the unknown parameter vector, $\vartheta$, where y is used to indicate the collection of sample data. It is usually simpler to work with the log of the likelihood function:

$$\ln L(\theta|y) = \sum \ln f(y_i|\theta).$$

Consider the following model:

$$y_i = \beta_0 + \beta_1 x_i + u_i, \quad i = 1, 2 \ldots, n$$

The following function is called a likelihood function, denoted by $\mathrm{LF}(\beta_1, \beta_2, \sigma^2)$

$$\mathrm{LF}(\beta_1, \beta_2, \sigma^2) = f(Y_1, Y_2, \ldots, Y_n|\beta_1 + \beta_2 X_i, \sigma^2) = \frac{1}{\sigma^n(\sqrt{2\pi})^n} \exp(\frac{1}{2} \sum \frac{(Y_i - \beta_1 - \beta_2 X_i)^2}{\sigma^2}) \tag{6}$$

where $\beta_1, \beta_2, \sigma^2$ are not known. The method of maximum likelihood, as the name indicates, consists in estimating the unknown parameters in such a manner that the probability of observing the given Y's is as high (or maximum) as possible. Therefore, we have to find the maximum of the function 6. For differentiation it is easier to express 6 in the log term as follows:

$$\begin{aligned} \ln LF &= -n \ln \sigma - \frac{n}{2} \ln(2\pi) - \frac{1}{2} \sum \frac{(Y_i - \beta_1 - \beta_2 X_i)^2}{\sigma^2} \\ &= -\frac{n}{2} \ln \sigma^2 - \frac{n}{2} \ln(2\pi) - \frac{1}{2} \sum \frac{(Y_i - \beta_1 - \beta_2 X_i)^2}{\sigma^2} \end{aligned} \tag{7}$$

Differentiating 7 partially with respect to $\beta_1$, $\beta_2$, and $\sigma^2$, we can obtain the ML estimators. install.packages(maxLik)

```
library(maxLik)
indfood<-read.csv(file="C:\\Users\\XXXHHF\\Documents\\R\\workfile\\#R lecture\\10\\Indfood.csv")
#indfood<-read.csv(file.choose())
foodexp<-indfood[,1]
totalexp<-indfood[,2]
lm_r <- lm(foodexp~totalexp)
summary(lm_r)
```

```
##
## Call:
## lm(formula = foodexp ~ totalexp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -153.766  -46.613    7.748   37.696  171.586
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 94.20878   50.85635   1.852   0.0695 .
## totalexp     0.43681    0.07832   5.577 8.45e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.86 on 53 degrees of freedom
## Multiple R-squared:  0.3698,Adjusted R-squared:  0.3579
## F-statistic:  31.1 on 1 and 53 DF,  p-value: 8.451e-07
```

```
loglik=function (para){
        N=length(foodexp)
        e=foodexp-para[1]-para[2]*totalexp
        ll=-0.5*N*log(2*pi)-0.5*N*log(para[3]^2)-0.5*sum(e^2/para[3]^2)
        return(ll)
}
mle1=maxLik(loglik,start=c(0.1,1,1))
coef(mle1)
```

```
## [1] 94.2090103  0.4368099 65.6268050
```

## 1.4 Multiple linear regression

### 1.4.1 Methodology

The essence of regression analysis is using sample data to estimate parameter values and their standard errors. First, however, we need to select a model which describes the relationship between the response variable and the explanatory variable(s). The simplest of all is the linear model

$$y = \mathbf{X}\beta + \varepsilon, \tag{8}$$

where

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix}, \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix},$$

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}, \hat{\beta} = \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_n \end{bmatrix}, e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix},$$

To properly interpret the coefficients of the OLS model, you must satisfy a number of statistical assumptions:

- Linearity —The dependent variable is linearly related to the independent variables.
- Strict exogeneity —The errors in the regression should have conditional mean zero.

$$E[\varepsilon|X] = 0$$

- No linear dependence —The regressors in X must all be linearly independent.

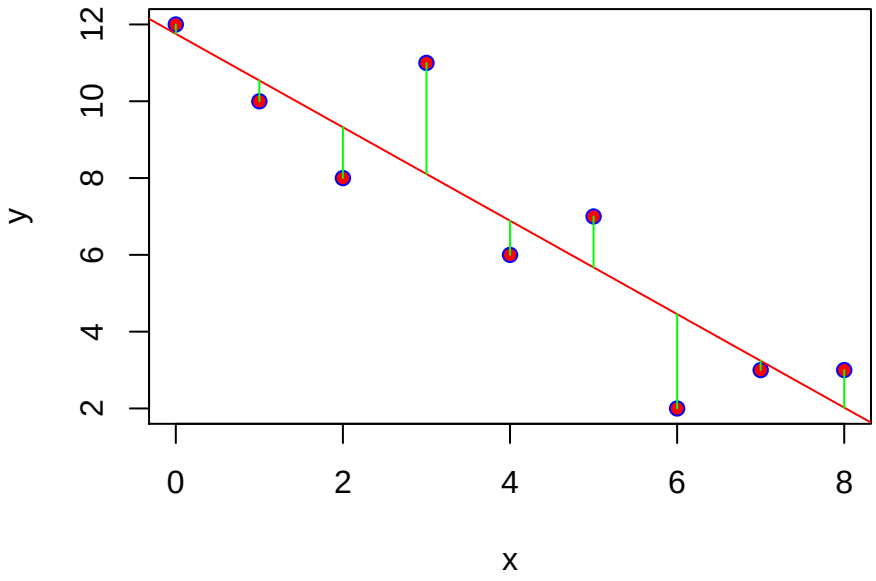
$$Pr[\text{rank}(X) = p] = 1$$

- Spherical errors
  - Homoscedasticity —The variance in y is constant (i.e. the variance does not change as y gets bigger).
  - Nonautocorrelation —The errors are uncorrelated between observations.

$$Var[\varepsilon|X] = \sigma^2 I_n$$

- Normality —For fixed values of the independent variables, the dependent variable is normally distributed.

$$\varepsilon|X \sim N(0, \sigma^2 I_n)$$




Our goal is to select model parameters (intercept and slopes) that minimize the difference between actual response values and those predicted by the model. Specifically, model parameters are selected to minimize the sum of squared residuals

$$\begin{aligned} \text{RSS} &= (Y - \mathbf{X}b)'(Y - \mathbf{X}b) \\ &= y'y - 2\hat{\beta}'\mathbf{X}'y + \hat{\beta}'\mathbf{X}'\mathbf{X}\hat{\beta}, \end{aligned} \tag{9}$$

differentiating with respect to $\hat{\beta}$

$$\frac{\partial \text{RSS}}{\partial \hat{\beta}} = -2\mathbf{X}'y + 2\mathbf{X}'\mathbf{X}\hat{\beta}, \tag{10}$$

we derive the OLS estimator as following.

$$b = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'y, \tag{11}$$

Substituting Equation (8) in Equation (11), we have

$$b = (\mathbf{X}^{'}\mathbf{X})^{-1}\mathbf{X}^{'}(\mathbf{X}\beta + \varepsilon) = \beta + (\mathbf{X}^{'}\mathbf{X})^{-1}\mathbf{X}^{'}\varepsilon. \tag{12}$$

Noting that $E[\varepsilon] = 0$, $E[\varepsilon\varepsilon^{'}] = \sigma^2\mathbf{I_n}$, and $(\mathbf{X}^{'}\mathbf{X})^{-1}$ is an idempotent matrix, where $\mathbf{I_n}$ is the $n \times n$ indentity matrix, the mean vector and the variance-covariance matrix of $b$ are

$$E[b] = \beta + (\mathbf{X}^{'}\mathbf{X})^{-1}\mathbf{X}^{'}E[\varepsilon] = \beta, \tag{13}$$

and

$$\begin{aligned} V[b] &= E[(b - E[b])(b - E[b])^{'}] = E[(b-\beta)(b-\beta)^{'}] \\ &= E[(\mathbf{X}^{'}\mathbf{X})^{-1}\mathbf{X}^{'}\varepsilon\varepsilon^{'}\mathbf{X}(\mathbf{X}^{'}\mathbf{X})^{-1}] = \sigma^2(\mathbf{X}^{'}\mathbf{X})^{-1}, \end{aligned} \tag{14}$$

respectively. The trace of the variance-covariance matrix is the total variance. Thus, the total variance is

$$TV[b] = \sigma^2 \text{Trace}(\mathbf{X}^{'}\mathbf{X})^{-1} = \sigma^2 \sum_{i=1}^{k} \frac{1}{\lambda_i} \tag{15}$$

where $\lambda_i$ are the positive eignevalues of $\mathbf{X}^{'}\mathbf{X}$. Since the normal distribution is reproductive, $b \sim N(\beta, \sigma^2(\mathbf{X}^{'}\mathbf{X})^{-1})$.

Further, the bias and the MSE of $b$ are

$$\text{bias}[b] = E[b] - \beta = 0, \tag{16}$$

and

$$\begin{aligned} \text{MSE}[b] &= E[(b-\beta)^{'}(b-\beta)] = E[\varepsilon^{'}\mathbf{X}(\mathbf{X}^{'}\mathbf{X})^{-1}(\mathbf{X}^{'}\mathbf{X})^{-1}\mathbf{X}^{'}\varepsilon] \\ &= E[\text{Trace}((\mathbf{X}^{'}\mathbf{X})^{-1}(\mathbf{X}^{'}\mathbf{X})^{-1}\mathbf{X}^{'}\mathbf{X}\varepsilon\varepsilon^{'})] = \sigma^2\text{Trace}(\mathbf{X}^{'}\mathbf{X})^{-1}. \end{aligned} \tag{17}$$

**Theorem 1.1 (Gauss-Markov)**: *The OLS estimator is more efficient than other linear unbiased estimator in terms of variance-covariance.*

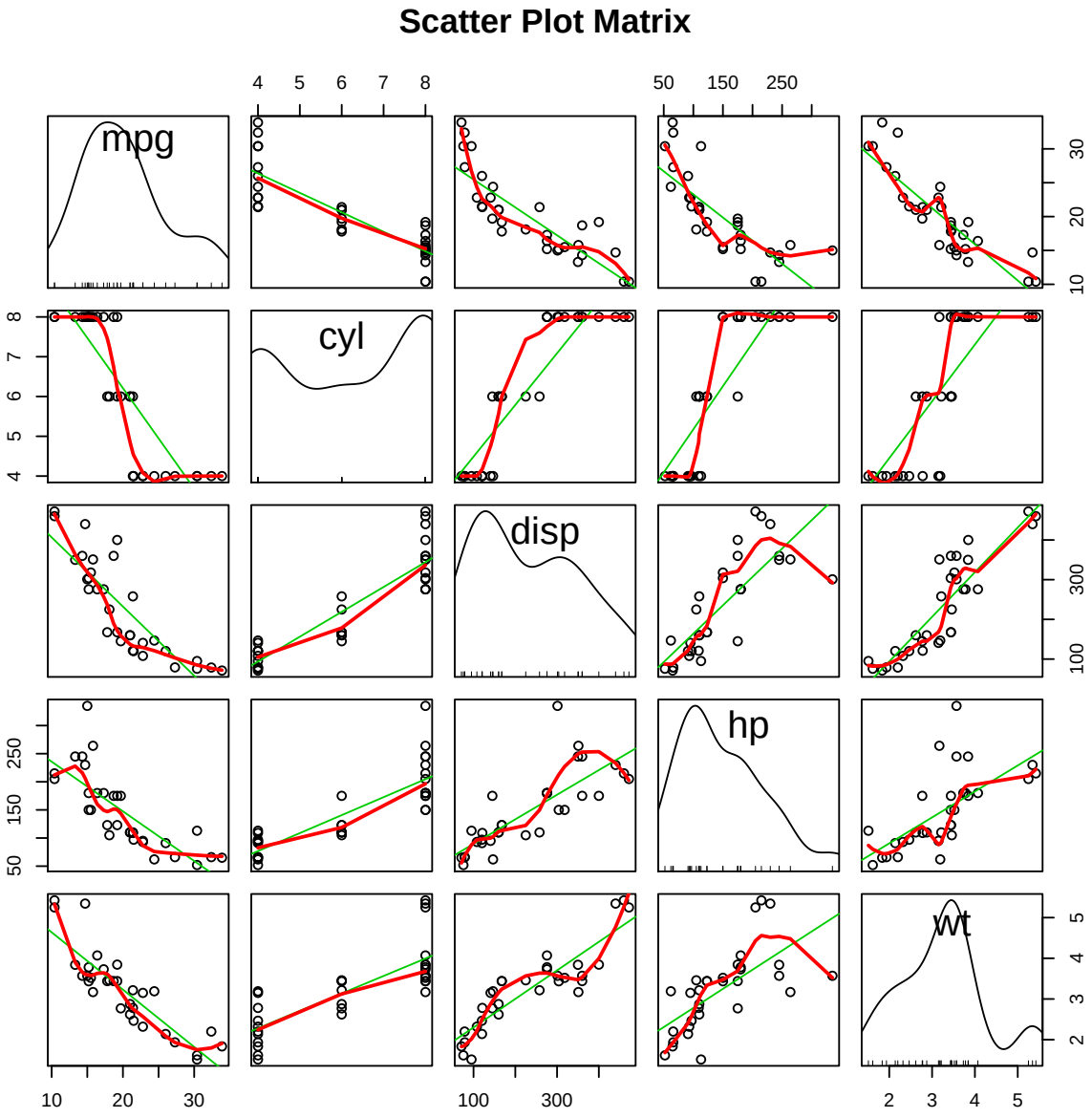
Since the bias of the OLS estimator is zero, the OLS estimator is an unbiased estimator.

### 1.4.2 Empirical analysis using R

When there's more than one predictor variable, simple linear regression becomes multiple linear regression.

```
class(mtcars)

## [1] "data.frame"

mtcar <- as.data.frame(mtcars[,c("mpg", "cyl",
        "disp", "hp", "wt")])
cor(mtcar)
```

```
##              mpg        cyl       disp         hp         wt
## mpg    1.0000000 -0.8521620 -0.8475514 -0.7761684 -0.8676594
## cyl   -0.8521620  1.0000000  0.9020329  0.8324475  0.7824958
## disp -0.8475514  0.9020329  1.0000000  0.7909486  0.8879799
## hp    -0.7761684  0.8324475  0.7909486  1.0000000  0.6587479
## wt    -0.8676594  0.7824958  0.8879799  0.6587479  1.0000000

library(car)
scatterplotMatrix(mtcar, spread=FALSE, main="Scatter Plot Matrix")
```

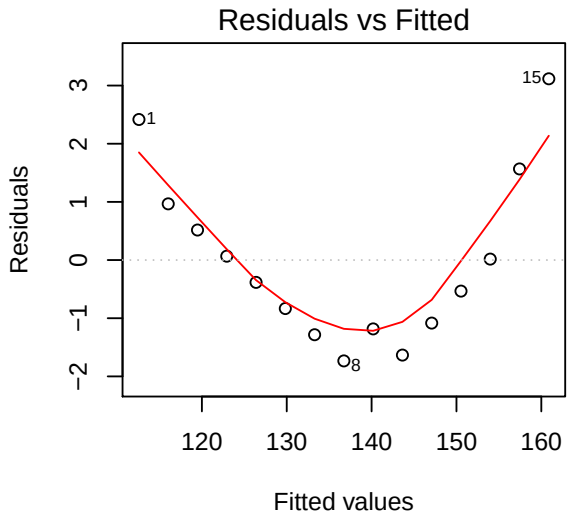


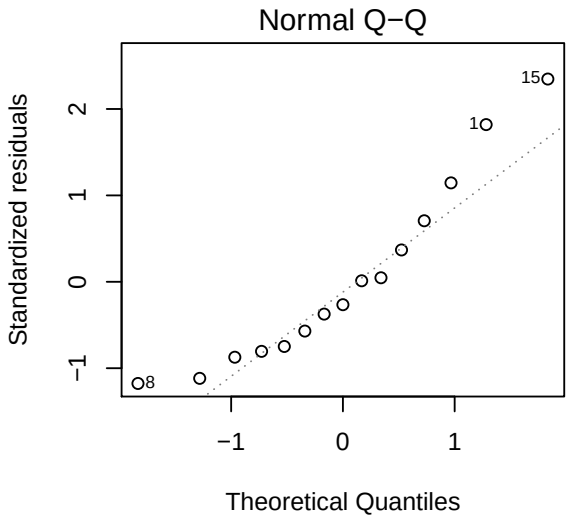

```
fit3 <- lm(mpg ~ hp + wt + hp:wt, data = mtcar)
#fit4 <- lm(mpg ~ cyl + disp + hp + wt, data=mtcar)
#as same as the following command
#fit5 <- lm(mpg ~ ., data=mtcar)
```

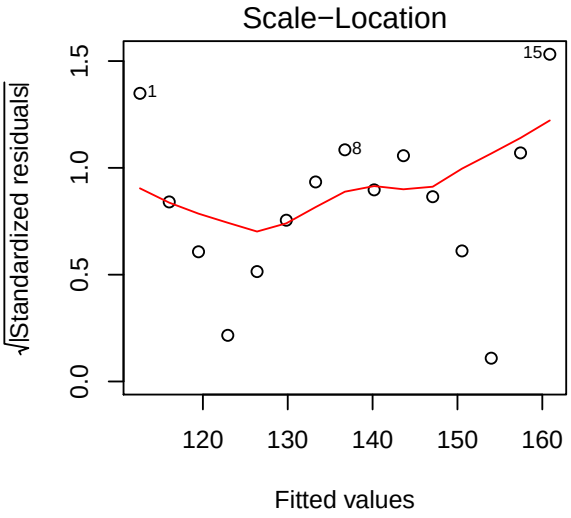
```
summary(fit3)

##
## Call:
## lm(formula = mpg ~ hp + wt + hp:wt, data = mtcar)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0632 -1.6491 -0.7362  1.4211  4.5513
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 49.80842    3.60516  13.816 5.01e-14 ***
## hp          -0.12010    0.02470  -4.863 4.04e-05 ***
## wt          -8.21662    1.26971  -6.471 5.20e-07 ***
## hp:wt        0.02785    0.00742   3.753 0.000811 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.153 on 28 degrees of freedom
## Multiple R-squared:  0.8848,Adjusted R-squared:  0.8724
## F-statistic: 71.66 on 3 and 28 DF,  p-value: 2.981e-13
```

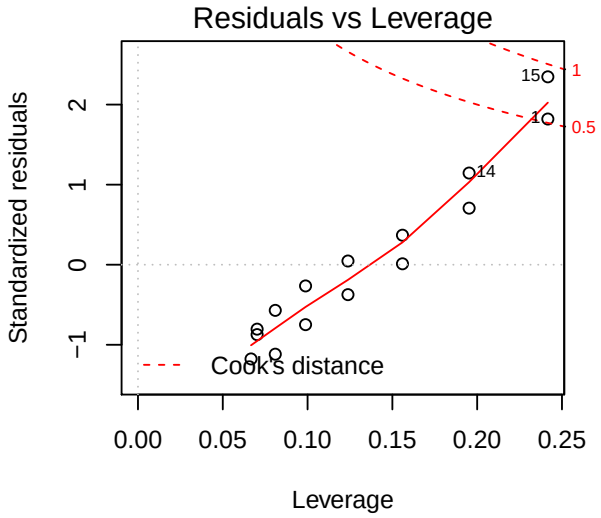
R's base installation provides numerous methods for evaluating the statistical assumptions in a regression analysis. The most common approach is to apply the plot() function to the object returned by the lm().

```
fit <- lm(weight ~ height, data=women)
par(mfrow=c(2,2))
plot(fit)
```

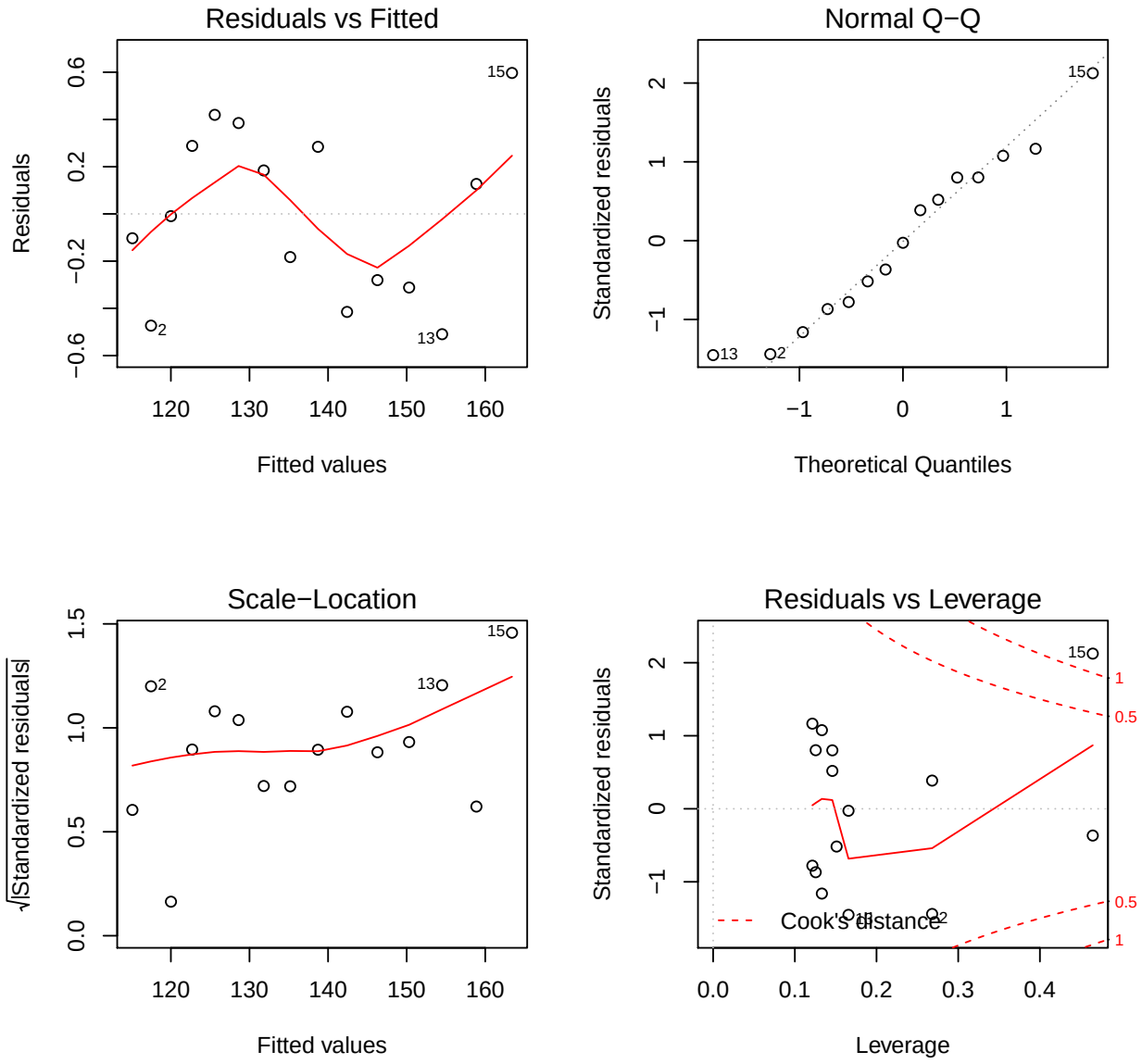



```
fit2 <- lm(weight ~ height + I(height^2), data=women)
plot(fit2)
```




- Normality —If the dependent variable is normally distributed for a fixed set of predictor values, then the residual values should be normally distributed with a mean of 0. The Normal Q-Q plot (upper right) is a probability plot of the standardized residuals against the values that would be expected under normality. If you've met the normality assumption, the points on this graph should fall on the straight 45-degree line. Because they don't, you've clearly violated the normality assumption.

- Homoscedasticity —If you've met the constant variance assumption, the points in the Scale-Location graph (bottom left) should be a random band around a horizontalline.

- Finally, the Residual versus Leverage graph (bottom right) provides information on individual observations that you may wish to attend to. The graph identifies outliers, high-leverage points, and influential observations. Specifically:

    1. An outlier is an observation that isn't predicted well by the fitted regression model (that is, has a large positive or negative residual).

2. An observation with a high leverage value has an unusual combination of predictor values. That is, it's an outlier in the predictor space. The dependent variable value isn't used to calculate an observation's leverage.
3. An influential observation is an observation that has a disproportionate impact on the determination of the model parameters. Influential observations are identified using a statistic called Cook's distance, or Cook's D.

### 1.4.3 Explanation of coefficients

In order to explain the meaning of coefficients ,we have the following step.

Regression model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + u_i$$

Step 1:

$$w_i = y_i - \hat{\alpha}_0 - \hat{\alpha}_1 x_{i2}$$

Step 2:

$$v_i = x_{i1} - \hat{b}_0 - \hat{b}_1 x_{i2}$$

Step 3:

$$\bar{\beta}_1 = \frac{\sum v_i w_i}{\sum v_i^2}$$

```
mtcar <- as.data.frame(mtcars[,c("mpg", "cyl", "disp", "hp", "wt")])
fit <- lm(mpg~wt+disp, data=mtcar)
summary(fit)

##
## Call:
## lm(formula = mpg ~ wt + disp, data = mtcar)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4087 -2.3243 -0.7683  1.7721  6.3484
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.96055    2.16454  16.151 4.91e-16 ***
## wt          -3.35082    1.16413  -2.878  0.00743 **
## disp        -0.01773    0.00919  -1.929  0.06362 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.917 on 29 degrees of freedom
## Multiple R-squared:  0.7809,Adjusted R-squared:  0.7658
## F-statistic: 51.69 on 2 and 29 DF,  p-value: 2.744e-10

fit1 <- lm(mpg~disp, data=mtcar)
fit2 <- lm(wt~disp, data=mtcar)
fit3 <- lm(fit1$residuals~fit2$residuals-1)
summary(fit3)
```

```
##
## Call:
## lm(formula = fit1$residuals ~ fit2$residuals - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4087 -2.3243 -0.7683  1.7721  6.3484
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## fit2$residuals   -3.351      1.126  -2.976  0.00562 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.821 on 31 degrees of freedom
## Multiple R-squared:  0.2222,Adjusted R-squared:  0.1971
## F-statistic: 8.857 on 1 and 31 DF,  p-value: 0.00562
```

### 1.4.4 Confidence intervals for regression

With the normality assumption for $u_i$, the OLS estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ are themselves normally distributed with means and variances given therein.

$$Z = \frac{\hat{\beta}_1 - \beta_1}{se(\hat{\beta}_1)} = \frac{(\hat{\beta}_1 - \beta_1)\sqrt{\sum(x_i - \bar{x})^2}}{\sigma} \tag{18}$$

If $\sigma^2$ is known, an important property of a normally distributed variable with mean $\mu$ and variance $\sigma^2$ is that the area under the normal curve between $\mu \pm \sigma$ is about 68 percent, that between the limits $\mu \pm 2\sigma$ is about 95 percent, and that between $\mu \pm 3\sigma$ is about 99.7 percent.

But $\sigma^2$ is rarely known, and in practice it is determined by the unbiased estimator $\hat{\sigma}^2$. If we replace $\sigma$ by $\hat{\sigma}$ , 18 may be written as

$$Z = \frac{\hat{\beta}_1 - \beta_1}{se(\hat{\beta}_1)} = \frac{(\hat{\beta}_1 - \beta_1)\sqrt{\sum(x_i - \bar{x})^2}}{\hat{\sigma}}$$

where the $se(\hat{\beta}_1)$ now refers to the estimated standard error. It can be shown that the $t$ variable thus defined follows the $t$ distribution with $n-2$ df. Therefore, instead of using the normal distribution, we can use the $t$ distribution to establish a confidence interval for $\beta_2$ as follows:

$$Pr(-t_{\alpha/2} \leq t \leq t_{\alpha/2}) = 1 - \alpha \tag{19}$$

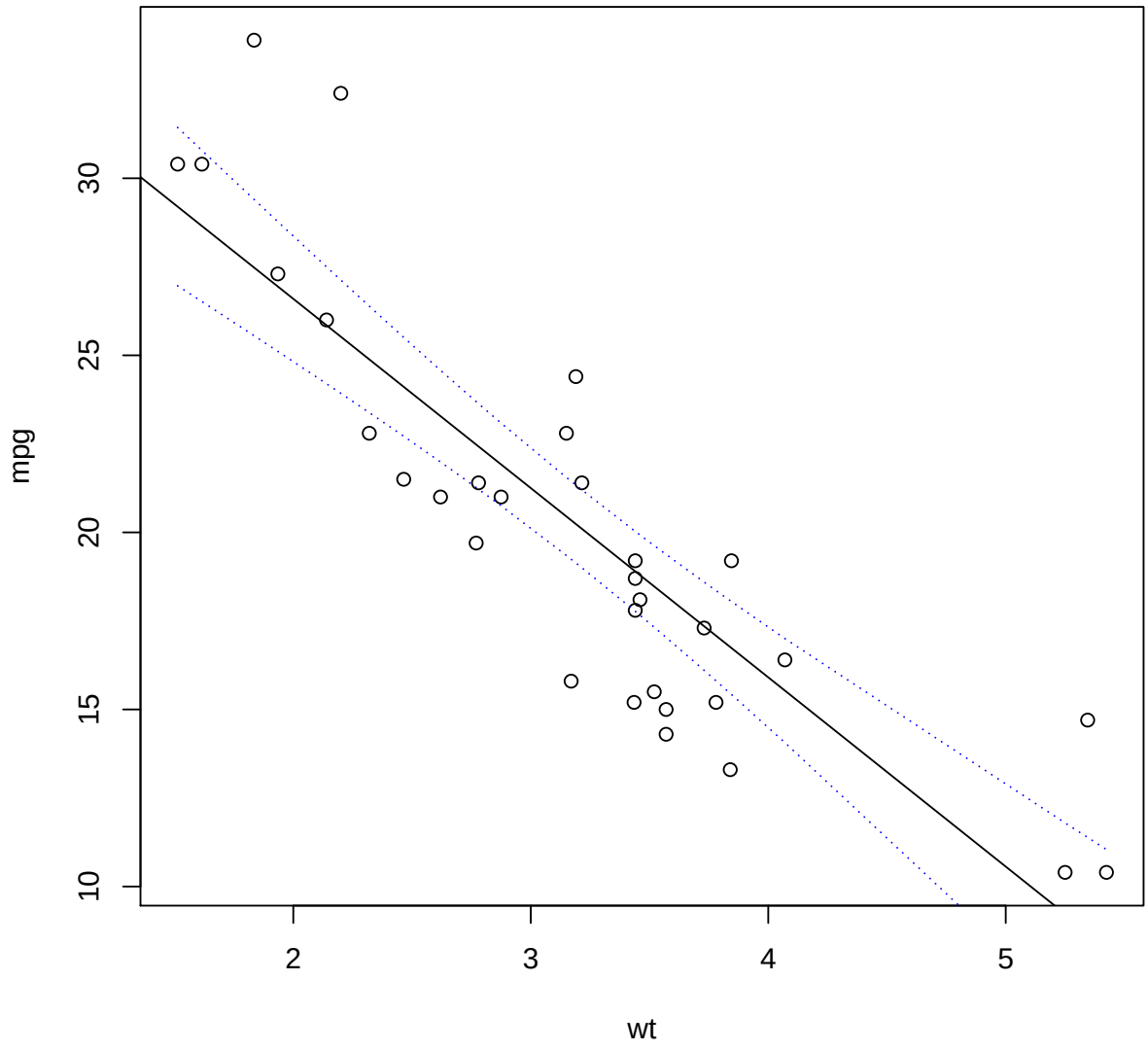
Rearranging 19, we obtain

$$Pr[\hat{\beta}_2 - t_{\alpha/2}se(\hat{\beta}_1) \leq \beta_1 \leq \hat{\beta}_2 + t_{\alpha/2}se(\hat{\beta}_1)] = 1 - \alpha \tag{20}$$

Equation 20 provides a $100(1-\alpha)$ percent confidence interval for $\beta_1$.

```
mtcar <- as.data.frame(mtcars[,c("mpg", "cyl", "disp", "hp", "wt")])
mtcarn<-mtcar[order(mtcar$wt),]
fit <- lm(mpg~wt, data=mtcarn)
conf=predict(fit,interval="confidence",level=0.95)
conf
```

```
##                          fit       lwr      upr
## Lotus Europa        29.198941 26.963760 31.43412
## Honda Civic         28.653805 26.519252 30.78836
## Toyota Corolla      27.478021 25.554415 29.40163
## Fiat X1-9           26.943574 25.112491 28.77466
## Porsche 914-2       25.847957 24.198041 27.49787
## Fiat 128            25.527289 23.927797 27.12678
## Datsun 710          24.885952 23.383008 26.38890
## Toyota Corona       24.111004 22.715653 25.50635
## Mazda RX4           23.282611 21.988668 24.57655
## Ferrari Dino        22.480940 21.268498 23.69338
## Volvo 142E          22.427495 21.219818 23.63517
## Mazda RX4 Wag       21.919770 20.752751 23.08679
## Merc 230            20.450041 19.347720 21.55236
## Ford Pantera L      20.343151 19.242185 21.44412
## Merc 240D           20.236262 19.136179 21.33635
## Hornet 4 Drive      20.102650 19.003004 21.20230
## AMC Javelin         18.926866 17.799465 20.05427
## Hornet Sportabout   18.900144 17.771469 20.02882
## Merc 280            18.900144 17.771469 20.02882
## Merc 280C           18.900144 17.771469 20.02882
## Valiant             18.793255 17.659216 19.92729
## Dodge Challenger    18.472586 17.319886 19.62529
## Duster 360          18.205363 17.034274 19.37645
## Maserati Bora       18.205363 17.034274 19.37645
## Merc 450SL          17.350247 16.104455 18.59604
## Merc 450SLC         17.083024 15.809403 18.35664
## Camaro Z28          16.762355 15.452833 18.07188
## Pontiac Firebird    16.735633 15.423002 18.04826
## Merc 450SE          15.533127 14.064349 17.00190
## Cadillac Fleetwood   9.226650  6.658271 11.79503
## Chrysler Imperial    8.718926  6.052112 11.38574
## Lincoln Continental  8.296712  5.547468 11.04596

plot(mpg~wt, data=mtcarn)
abline(fit)
lines(mtcarn$wt,conf[,2],lty=3,col="blue")
lines(mtcarn$wt,conf[,3],lty=3,col="blue")
```




### 1.4.5 Hypothesis testing

The standard regression output as provided by summary() only indicates individual significance of each regressor and joint significance of all regressors in the form of t and F statistics, respectively. Often it is necessary to test more general hypotheses.

```
mtcar <- as.data.frame(mtcars[,c("mpg", "cyl",
        "disp", "hp", "wt")])
library(car)
fit <- lm(mpg ~ hp + wt, data = mtcar)
summary(fit)

##
## Call:
```

```
## lm(formula = mpg ~ hp + wt, data = mtcar)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.941 -1.600 -0.182  1.050  5.854
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.22727    1.59879  23.285  < 2e-16 ***
## hp          -0.03177    0.00903  -3.519  0.00145 **
## wt          -3.87783    0.63273  -6.129 1.12e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.593 on 29 degrees of freedom
## Multiple R-squared:  0.8268,Adjusted R-squared:  0.8148
## F-statistic: 69.21 on 2 and 29 DF,  p-value: 9.109e-12
```

```
linearHypothesis(fit, "hp = 0")
```

```
## Linear hypothesis test
##
## Hypothesis:
## hp = 0
##
## Model 1: restricted model
## Model 2: mpg ~ hp + wt
##
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     30 278.32
## 2     29 195.05  1    83.274 12.381 0.001451 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(fit, "hp = -0.5")
```

```
## Linear hypothesis test
##
## Hypothesis:
## hp = - 0.5
##
## Model 1: restricted model
## Model 2: mpg ~ hp + wt
##
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1     30 18280
## 2     29   195  1     18085 2688.8 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(fit, "hp - wt= 0")
```

```
## Linear hypothesis test
##
## Hypothesis:
```

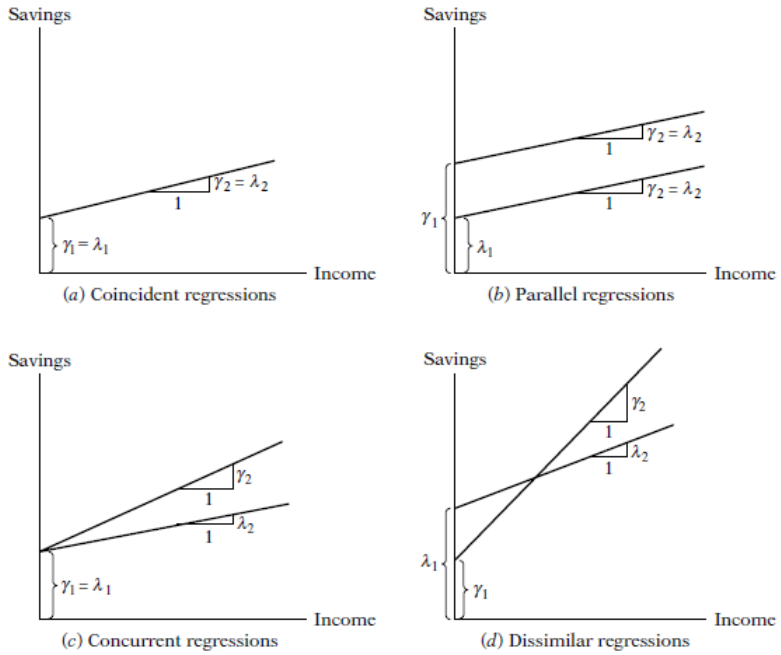



Figure 3: Dummy

```
## hp - wt = 0
##
## Model 1: restricted model
## Model 2: mpg ~ hp + wt
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     30 438.92
## 2     29 195.05  1    243.87 36.259 1.501e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 1.5 Dummy variable regression model

In regression analysis the dependent variable, or regressand, is frequently influenced not only by ratio scale variables (e.g., income, output, prices, costs, height, temperature) but also by variables that are essentially qualitative, or nominal scale, in nature, such as sex, race, color, religion, nationality, etc. Variables that assume such 0 and 1 values are called dummy variables.

1. Both the intercept and the slope coefficients are the same in the two regressions. This, the case of coincident regressions.

2. Only the intercepts in the two regressions are different but the slopes are the same. This is the case of parallel regressions.

3. The intercepts in the two regressions are the same, but the slopes are different. This is the situation of concurrent regressions.

4. Both the intercepts and slopes in the two regressions are different. This is the case of dissimilar regressions.

- INTERACTION EFFECTS USING DUMMY VARIABLES
- DUMMY VARIABLES IN SEASONAL ANALYSIS
- PIECEWISE LINEAR REGRESSION

### Predictors with Only Two Levels

Suppose that we wish to investigate differences in credit card balance between males and females, ignoring the other variables for the moment. If a qualitative predictor (also known as a factor) only has two levels, or possible values, then incorporating it into a regression model is very simple. We simply create an indicator or dummy variable that takes on two possible numerical values.

### Qualitative Predictors with More than Two Levels

When a qualitative predictor has more than two levels, a single dummy variable cannot represent all possible values. In this situation, we can create additional dummy variables. For example, for the ethnicity variable we create two dummy variables. The first could be

$$D_{i1} = \begin{cases} 1 & \text{if ith person is Asian} \\ 0 & \text{if ith person is not Asian} \end{cases}$$

and the second could be

$$D_{i2} = \begin{cases} 1 & \text{if ith person is Caucasian} \\ 0 & \text{if ith person is not Caucasian} \end{cases}$$

Then both of these variables can be used in the regression equation, in order to obtain the model

$$y_i = \beta_0 + \beta_1 D_{i1} + \beta_2 D_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if ith person is Asian} \\ \beta_0 + \beta_2 + \epsilon_i & \text{if ith person is Caucasian} \\ \beta_0 + \epsilon_i & \text{if ith person is African Americ} \end{cases}$$

Now $\beta_0$ can be interpreted as the average credit card balance for African Americans, $\beta_1$ can be interpreted as the difference in the average balance between the Asian and African American categories, and $\beta_2$ can be interpreted as the difference in the average balance between the Caucasian and African American categories. There will always be one fewer dummy variable than the number of levels. The level with no dummy variable — African American in this example — is known as the baseline.

### How to generate a Dummy?

```
con_China<-read.csv("consumption_China.csv")
names(con_China)<-c("year","c","y")
names(con_China)

## [1] "year" "c"    "y"

con_China$year

##  [1] 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991
## [15] 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005
## [29] 2006

Dummy<-con_China$year>=1992
Dummy
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [23]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE

Dummy<-as.numeric(Dummy)
Dummy

##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

### An Exercises

```
wage1<-read.csv("wage1.csv",header=T)
head(wage1)

##   wage educ exper tenure nonwhite female married numdep smsa northcen
## 1 3.10   11     2      0        0      1       0      2    1        0
## 2 3.24   12    22      2        0      1       1      3    1        0
## 3 3.00   11     2      0        0      0       0      2    0        0
## 4 6.00    8    44     28        0      0       1      0    1        0
## 5 5.30   12     7      2        0      0       1      1    0        0
## 6 8.75   16     9      8        0      0       1      0    1        0
##   south west construc ndurman trcommpu trade services profserv profocc
## 1     0    1        0       0        0     0        0        0       0
## 2     0    1        0       0        0     0        1        0       0
## 3     0    1        0       0        0     1        0        0       0
## 4     0    1        0       0        0     0        0        0       0
## 5     0    1        0       0        0     0        0        0       0
## 6     0    1        0       0        0     0        0        1       1
##   clerocc servocc    lwage expersq tenursq
## 1       0       0 1.131402       4       0
## 2       0       1 1.175573     484       4
## 3       0       0 1.098612       4       0
## 4       1       0 1.791759    1936     784
## 5       0       0 1.667707      49       4
## 6       0       0 2.169054      81      64

lm.wage<-lm(lwage~ female+educ+exper+expersq+tenure+tenursq,data=wage1)
summary(lm.wage)

##
## Call:
## lm(formula = lwage ~ female + educ + exper + expersq + tenure +
##     tenursq, data = wage1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.83160 -0.25659 -0.02126  0.25500  1.13370
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.4166909  0.0989279   4.212 2.98e-05 ***
## female      -0.2965110  0.0358055  -8.281 1.04e-15 ***
## educ         0.0801967  0.0067573  11.868  < 2e-16 ***
```

```
## exper        0.0294324  0.0049752   5.916 6.00e-09 ***
## expersq     -0.0005827  0.0001073  -5.431 8.65e-08 ***
## tenure       0.0317139  0.0068452   4.633 4.56e-06 ***
## tenursq     -0.0005852  0.0002347  -2.493    0.013 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3998 on 519 degrees of freedom
## Multiple R-squared:  0.4408,Adjusted R-squared:  0.4343
## F-statistic: 68.18 on 6 and 519 DF,  p-value: < 2.2e-16

#
attach(wage1)
marrmale<-as.numeric(married==1 & female==0)
marrfemale<-as.numeric(married==1 & female==1)
singmale<-as.numeric(married==0 & female==0)
singfem<-as.numeric(married==0 & female==1)
lm.wage1<-lm(lwage~marrmale+marrfemale+singfem+educ+exper +tenure+expersq+tenursq)
summary(lm.wage1)

##
## Call:
## lm(formula = lwage ~ marrmale + marrfemale + singfem + educ +
##     exper + tenure + expersq + tenursq)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89697 -0.24060 -0.02689  0.23144  1.09197
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.3213780  0.1000090   3.213 0.001393 **
## marrmale     0.2126756  0.0553572   3.842 0.000137 ***
## marrfemale  -0.1982677  0.0578355  -3.428 0.000656 ***
## singfem     -0.1103502  0.0557421  -1.980 0.048272 *
## educ         0.0789103  0.0066945  11.787  < 2e-16 ***
## exper        0.0268006  0.0052428   5.112 4.50e-07 ***
## tenure       0.0290875  0.0067620   4.302 2.03e-05 ***
## expersq     -0.0005352  0.0001104  -4.847 1.66e-06 ***
## tenursq     -0.0005331  0.0002312  -2.306 0.021531 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3933 on 517 degrees of freedom
## Multiple R-squared:  0.4609,Adjusted R-squared:  0.4525
## F-statistic: 55.25 on 8 and 517 DF,  p-value: < 2.2e-16

#drop the sinmale automatically
lm.wage2<-lm(lwage~marrmale+marrfemale+singfem+educ+exper+tenure+expersq+tenursq+singmale)
summary(lm.wage2)

##
## Call:
## lm(formula = lwage ~ marrmale + marrfemale + singfem + educ +
##     exper + tenure + expersq + tenursq + singmale)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89697 -0.24060 -0.02689  0.23144  1.09197
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.3213780  0.1000090   3.213 0.001393 **
## marrmale     0.2126756  0.0553572   3.842 0.000137 ***
## marrfemale  -0.1982677  0.0578355  -3.428 0.000656 ***
## singfem     -0.1103502  0.0557421  -1.980 0.048272 *
## educ         0.0789103  0.0066945  11.787  < 2e-16 ***
## exper        0.0268006  0.0052428   5.112 4.50e-07 ***
## tenure       0.0290875  0.0067620   4.302 2.03e-05 ***
## expersq     -0.0005352  0.0001104  -4.847 1.66e-06 ***
## tenursq     -0.0005331  0.0002312  -2.306 0.021531 *
## singmale            NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3933 on 517 degrees of freedom
## Multiple R-squared:  0.4609,Adjusted R-squared:  0.4525
## F-statistic: 55.25 on 8 and 517 DF,  p-value: < 2.2e-16
```

```
#
fema<-female*married
lm.wage3<-lm(lwage~female+married+fema+educ+exper+tenure+expersq+tenursq)
summary(lm.wage3)    #interceprt of married==1 and female==0
```

```
##
## Call:
## lm(formula = lwage ~ female + married + fema + educ + exper +
##     tenure + expersq + tenursq)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89697 -0.24060 -0.02689  0.23144  1.09197
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.3213780  0.1000090   3.213 0.001393 **
## female      -0.1103502  0.0557421  -1.980 0.048272 *
## married      0.2126756  0.0553572   3.842 0.000137 ***
## fema        -0.3005931  0.0717670  -4.188 3.30e-05 ***
## educ         0.0789103  0.0066945  11.787  < 2e-16 ***
## exper        0.0268006  0.0052428   5.112 4.50e-07 ***
## tenure       0.0290875  0.0067620   4.302 2.03e-05 ***
## expersq     -0.0005352  0.0001104  -4.847 1.66e-06 ***
## tenursq     -0.0005331  0.0002312  -2.306 0.021531 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3933 on 517 degrees of freedom
## Multiple R-squared:  0.4609,Adjusted R-squared:  0.4525
## F-statistic: 55.25 on 8 and 517 DF,  p-value: < 2.2e-16
```

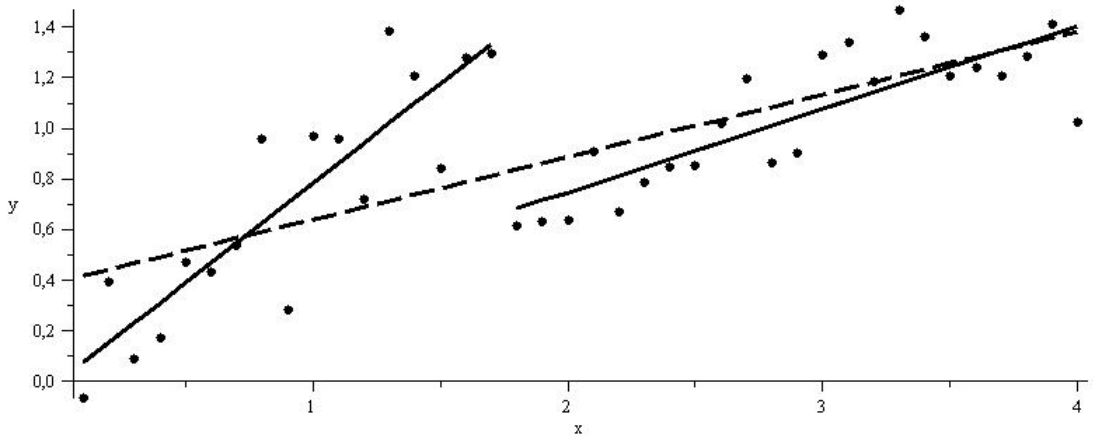



Figure 4: structural break

```
detach(wage1)
```

## 1.6 Chow Test

The Chow test is a statistical and econometric test of whether the coefficients in two linear regressions on different data sets are equal. The Chow test was invented by economist Gregory Chow in 1960. In econometrics, the Chow test is most commonly used in time series analysis to test for the presence of a structural break.

The chow test of regressions from sample sizes $n_1$ and $n_2$ is then carried out using the following steps.

1. Run a regression on the combined sample with size $n = n_1 + n_2$ and obtain within group sum of squares called $S_1$. The number of degrees of freedom is $n_1 + n_2 - k$, with $k$ being the number of parameters estimated, including the intercept.

2. Run two regressions on the two individual samples with sizes $n_1$ and $n_2$, and obtain their within group sums of square $S_2 + S_3$, with $n_1 + n_2 - 2k$ degrees of freedom.

3. Conduct an $F_{(k,n_1+n_2-2k)}$ test defined by

$$F = \frac{[S_1 - (S_2 + S_3)]/k}{[(S_2 + S_3)/(n_1 + n_2 - 2k)]}$$

If the F statistic exceeds the critical F, we reject the null hypothesis that the two regressions are equal.

```
source("C:\\Users\\XXXHHF\\Documents\\LYX\\10. Linear Regression\\data\\Chow_test.R")
con_China<-read.csv("C:\\Users\\XXXHHF\\Documents\\LYX\\10. Linear Regression\\data\\consumption_Chin
#View(con_China)
names(con_China)<-c("year","con","GDP")
attach(con_China)
con1<-con[year<=1991]
con2<-con[year>1991]
GDP1<-GDP[year<=1991]
GDP2<-GDP[year>1991]
dat1<-cbind(GDP1,con1)
dat2<-cbind(GDP2,con2)
```

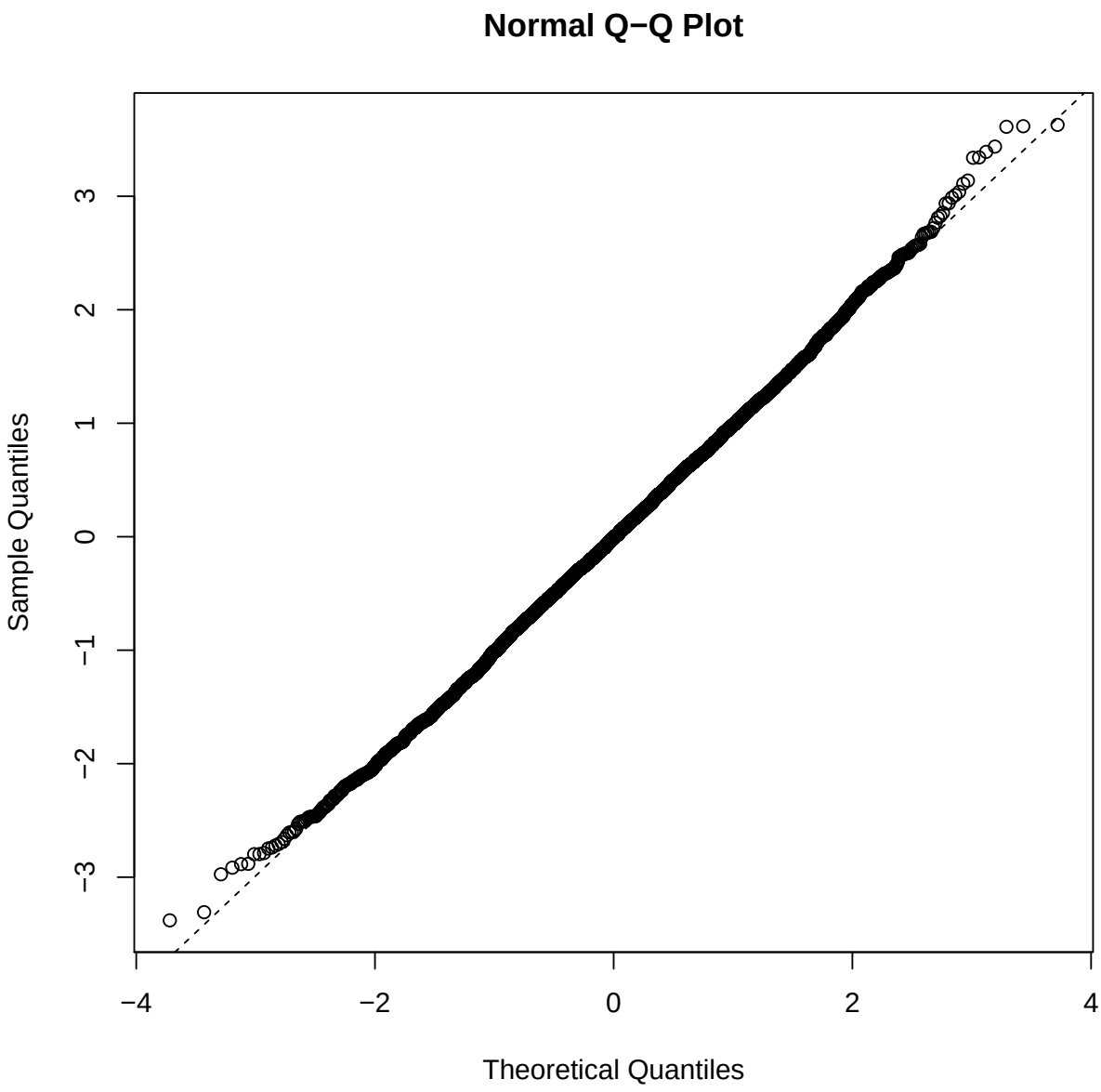
```
chow(dat1,dat2)

##
##  Chow TEST
##
## data:  dat1 and dat2
## F = 13.558, df1 = 2, df2 = 25, p-value = 0.0001028
```

## 1.7 Normality

### Testing normality

The simplest test of normality (and in many ways the best) is the 'quantile–quantile plot'. This plots the ranked samples from our distribution against a similar number of ranked quantiles taken from a normal distribution. If our sample is normally distributed then the line will be straight. Departures from normality show up as various sorts of non-linearity (e.g. S-shapes or banana shapes). The functions you need are qqnorm and qqline (quantile–quantile plot against a normal distribution)

```
y <- rnorm(5000)
shapiro.test(y)

##
##  Shapiro-Wilk normality test
##
## data:  y
## W = 0.9994, p-value = 0.09815

qqnorm(y)
qqline(y,lty=2)
```



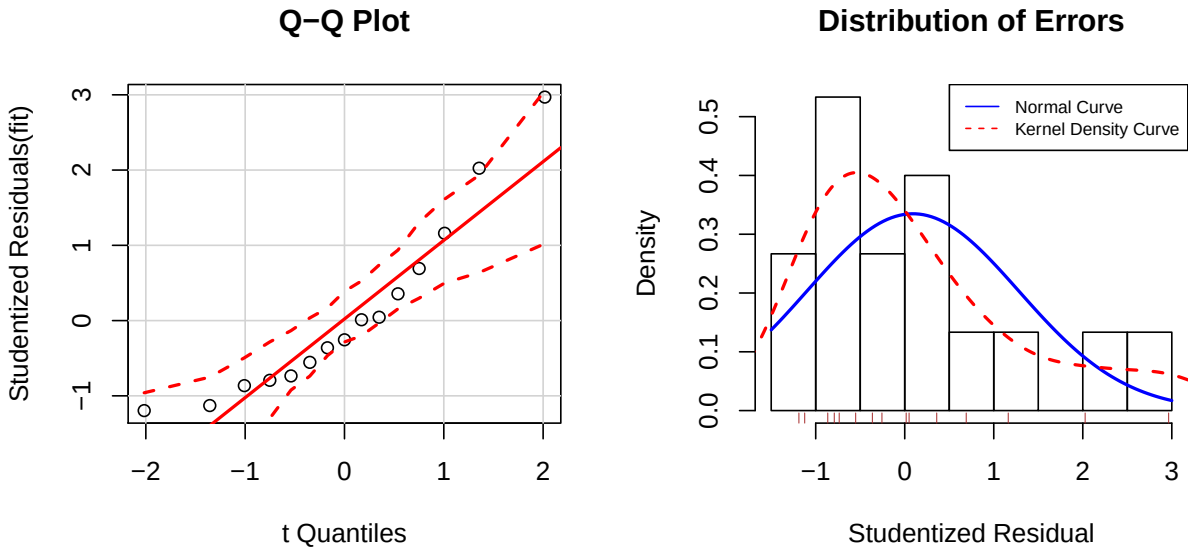

The car package provides a number of functions that significantly enhance your ability to fit and evaluate regression models. (See help file of car package for further details)

```
shapiro.test(runif(100,min=2,max=4))

##
##  Shapiro-Wilk normality test
##
## data:  runif(100, min = 2, max = 4)
## W = 0.93271, p-value = 7.118e-05
```

The qqPlot() function provides a more accurate method of assessing the normality assumption than provided by the plot() function in the base package. It plots the studentized residuals (also called studentized deleted residuals or jackknifed residuals) against a t distribution with n-p-1 degrees of freedom, where n is the sample size and p is the number of regression parameters (including the intercept). The residplot() function

generates a histogram of the studentized residuals and superimposes a normal curve, kernel density curve, and rug plot. It doesn't require the car package.

```
library(car)
fit <- lm(weight ~ height, data=women)
residplot <- function(fit, nbreaks=10) {
    z <- rstudent(fit)
    hist(z, breaks=nbreaks, freq=FALSE,
    xlab="Studentized Residual",
    main="Distribution of Errors")
    rug(jitter(z), col="brown")
    curve(dnorm(x, mean=mean(z), sd=sd(z)),
        add=TRUE, col="blue", lwd=2)
    lines(density(z)$x, density(z)$y,
        col="red", lwd=2, lty=2)
    legend("topright",
        legend = c( "Normal Curve", "Kernel Density Curve"),
        lty=1:2, col=c("blue","red"), cex=.7) }
par(mfrow=c(1,2))
qqPlot(fit,labels=row.names(women),id.method="identify", simulate=TRUE, main="Q-Q Plot")
residplot(fit)
```




### Jarque–Bera (JB) Test of Normality

The JB test of normality is an asymptotic, or large-sample, test. It is also based on the OLS residuals. This test first computes the skewness and kurtosis measures of the OLS residuals and uses the following test statistic:

$$\text{JB} = n[\frac{S^2}{6} + \frac{(K-3)^2}{24}]$$

where n = sample size, S = skewness coefficient, and K = kurtosis coefficient. For a normally distributed variable, S = 0 and K = 3. Therefore, the JB test of normality is a test of the joint hypothesis that S and K are 0 and 3, respectively. In that case the value of the JB statistic is expected to be 0.

Under the null hypothesis that the residuals are normally distributed, Jarque and Bera showed that asymptotically (i.e., in large samples) the JB statistic follows the chi-square distribution with 2 df. If the computed p value of the JB statistic in an application is sufficiently low, which will happen if the value of the statistic is very different from 0, one can reject the hypothesis that the residuals are normally distributed. But if the p value is reasonably high, which will happen if the value of the statistic is close to zero, we do not reject the normality assumption.

```
library("ccgarch")
jb.test(fit$residuals)

##            series 1
## test stat 1.6595731
## p-value   0.4361424
```

## 1.8 Nonlinear regression

### 1.8.1 Type 1 (log-log model)

REAL GROSS PRODUCT, LABOR DAYS, AND REAL CAPITAL INPUT IN THE AGRICULTURAL SECTOR OF TAIWAN, 1958–1972

| Year | Real gross product (millions of NT $)*, $Y$ | Labor days (millions of days), $X_2$ | Real capital input (millions of NT $), $X_3$ |
|---|---|---|---|
| 1958 | 16,607.7 | 275.5 | 17,803.7 |
| 1959 | 17,511.3 | 274.4 | 18,096.8 |
| 1960 | 20,171.2 | 269.7 | 18,271.8 |
| 1961 | 20,932.9 | 267.0 | 19,167.3 |
| 1962 | 20,406.0 | 267.8 | 19,647.6 |
| 1963 | 20,831.6 | 275.0 | 20,803.5 |
| 1964 | 24,806.3 | 283.0 | 22,076.6 |
| 1965 | 26,465.8 | 300.7 | 23,445.2 |
| 1966 | 27,403.0 | 307.5 | 24,939.0 |
| 1967 | 28,628.7 | 303.7 | 26,713.7 |
| 1968 | 29,904.5 | 304.7 | 29,957.8 |
| 1969 | 27,508.2 | 298.6 | 31,585.9 |
| 1970 | 29,035.5 | 295.5 | 33,474.5 |
| 1971 | 29,281.5 | 299.0 | 34,821.8 |
| 1972 | 31,535.8 | 288.1 | 41,794.3 |

*Source:* Thomas Pei-Fan Chen, "Economic Growth and Structural Change in Taiwan—1952–1972, A Production Function Approach," unpublished Ph.D. thesis, Dept. of Economics, Graduate Center, City University of New York, June 1976, Table II.

*New Taiwan dollars.

We can convert nonlinear relationships into linear ones so that we can work within the framework of the classical linear regression model. The Cobb–Douglas production function, in its stochastic form, may be expressed as

$$Y_i = \beta_1 + X_2^{\beta_2} X_3^{\beta_3} e^{\varepsilon_i} \tag{21}$$

where $Y$ = output, $X_2$ = labor input, $X_3$ = capital input, $\varepsilon$ = stochastic disturbance term, $e$ = base of natural logarithm.

From 21 it is clear that the relationship between output and the two inputs is nonlinear. However, if we log-transform this model, we obtain:

$$\ln Y_i = \ln \beta_1 + \beta_2 \ln X_2 + \beta_3 \ln X_3 + \varepsilon_i \tag{22}$$

where $\beta_2$ is the (partial) elasticity of output with respect to the labor input, that is, it measures the percentage change in output for, say, a 1 percent change in the labor input, holding the capital input constant. Likewise, $\beta_3$ is the (partial) elasticity of output with respect to the capital input, holding the labor input constant.

To see this, differentiate 22 partially with respect to the log of each X variable. Therefore, $\partial \ln Y/\partial \ln X_2 = (\partial Y/\partial X_2)(X_2/Y) = \beta_2$, which, by definition, is the elasticity of Y with respect to $X_2$ and $\partial \ln Y/\partial \ln X_3 = (\partial Y/\partial X_3)(X_3/Y) = \beta_3$, which is the elasticity of Y with respect to $X_3$.

```
cobb<-read.csv("C:\\Users\\XXXHHF\\Documents\\R\\workfile\\Cobb.csv",header=T)
cobb

##     obs       Y      X3    X2
## 1  1958 16607.7 17803.7 275.5
## 2  1959 17511.3 18096.8 274.4
## 3  1960 20171.2 18271.8 269.7
## 4  1961 20932.9 19167.3 267.0
## 5  1962 20406.0 19647.6 267.8
## 6  1963 20831.6 20803.5 275.0
## 7  1964 24806.3 22076.6 283.0
## 8  1965 26465.8 23445.2 300.7
## 9  1966 27403.0 24939.0 307.5
## 10 1967 28628.7 26713.7 303.7
## 11 1968 29904.5 29957.8 304.7

lny<-log(cobb$Y)
lnX2<-log(cobb$X2)
lnX3<-log(cobb$X3)
summary(lm(lny~lnX2+lnX3))

##
## Call:
## lm(formula = lny ~ lnX2 + lnX3)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.100540 -0.052948  0.001617  0.058639  0.067641
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.10543    2.70993  -0.408  0.69403
## lnX2         0.07285    0.87550   0.083  0.93573
## lnX3         1.07430    0.28582   3.759  0.00556 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06998 on 8 degrees of freedom
## Multiple R-squared:  0.9023,Adjusted R-squared:  0.8779
## F-statistic: 36.94 on 2 and 8 DF,  p-value: 9.116e-05
```

we see that in the Taiwanese agricultural sector for the period 1958–1968 the output elasticities of labor and capital were 0.07285 and 1.07430, respectively. In other words, over the period of study, holding the capital input constant, a 1 percent increase in the labor input led on the average to about a 0.07 percent increase in the output. Similarly, holding the labor input constant, a 1 percent increase in the capital input led on the average to about a 1.07 percent increase in the output. Adding the two output elasticities, we obtain 1.14, which gives the value of the returns to scale parameter.

**log-linear model**

$$\ln Y_i = \beta_1 + \beta_2 X_i + \varepsilon_i$$

**linear-log model**

$$Y_i = \beta_1 + \beta_2 \ln X_i + \varepsilon_i$$

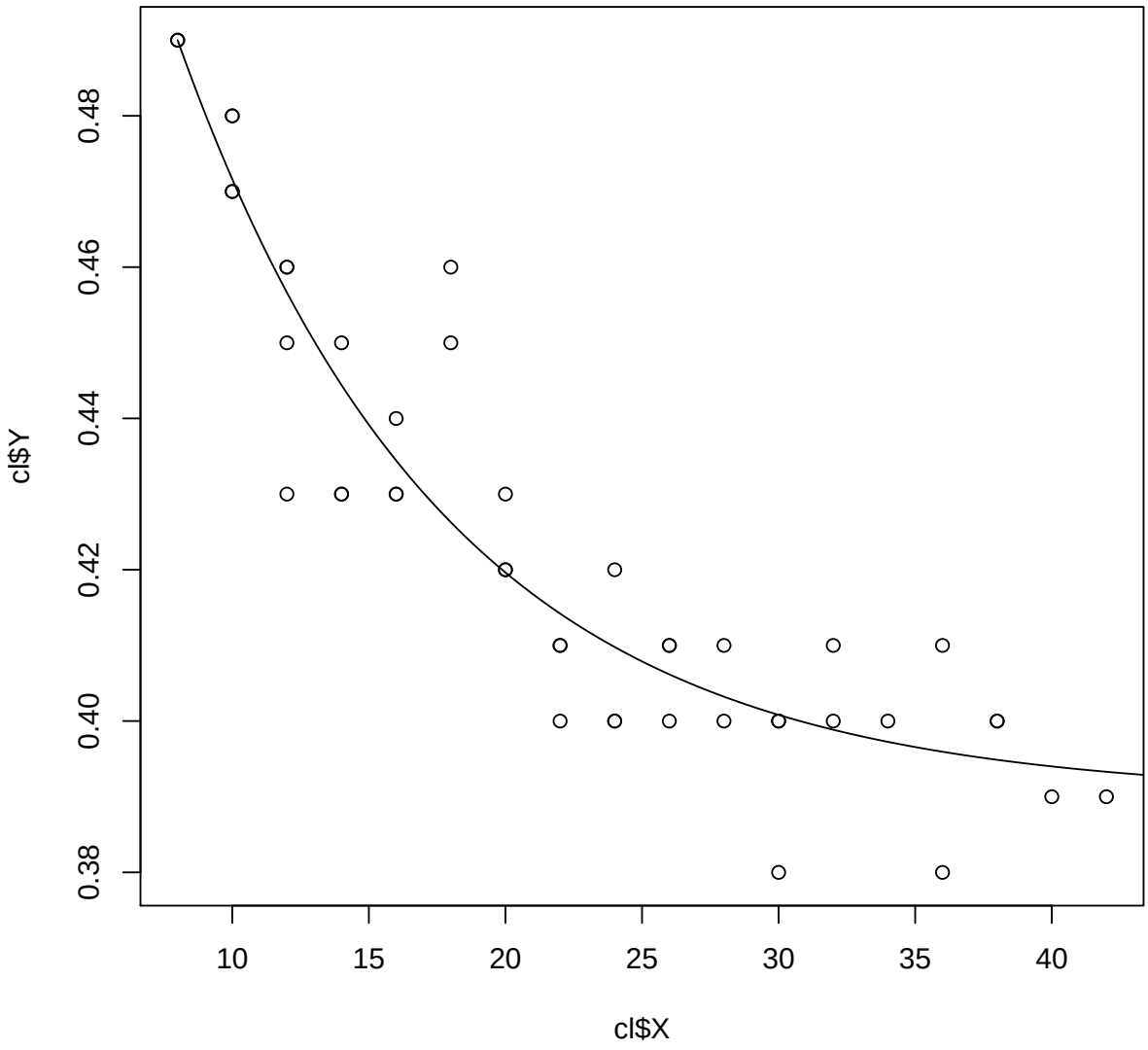
### 1.8.2 Type 2

Sometimes you know the form of a model, even if the model is extremely nonlinear. To fit nonlinear models (minimizing least squares error), you can use the nls function.

For example, we have the nonlinear regression:

$$Y = \alpha + (0.49 - \alpha)\exp(-\beta(X - 8)) + \varepsilon$$

```
cl<-data.frame(
   X=c(rep(2*4:21, c(2, 4, 4, 3, 3, 2, 3, 3, 3, 3, 2,
       3, 2, 1, 2, 2, 1, 1))),
   Y=c(0.49, 0.49, 0.48, 0.47, 0.48, 0.47, 0.46, 0.46,
        0.45, 0.43, 0.45, 0.43, 0.43, 0.44, 0.43, 0.43,
        0.46, 0.45, 0.42, 0.42, 0.43, 0.41, 0.41, 0.40,
        0.42, 0.40, 0.40, 0.41, 0.40, 0.41, 0.41, 0.40,
        0.40, 0.40, 0.38, 0.41, 0.40, 0.40, 0.41, 0.38,
        0.40, 0.40, 0.39, 0.39)
)
#nonlinear regression
nls.sol<-nls(Y~a+(0.49-a)*exp(-b*(X-8)), data=cl,
             start = list( a= 0.1, b = 0.01 ))
nls.sum<-summary(nls.sol); nls.sum

##
## Formula: Y ~ a + (0.49 - a) * exp(-b * (X - 8))
##
## Parameters:
##   Estimate Std. Error t value Pr(>|t|)
## a 0.390140   0.005045  77.333  < 2e-16 ***
## b 0.101633   0.013360   7.607 1.99e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01091 on 42 degrees of freedom
##
## Number of iterations to convergence: 19
## Achieved convergence tolerance: 1.365e-06

#plot the fitted line and scatter plot
xfit<-seq(8,44,len=200)
yfit<-predict(nls.sol, data.frame(X=xfit))
plot(cl$X, cl$Y)
lines(xfit,yfit)
```




# 2 Subset Selection

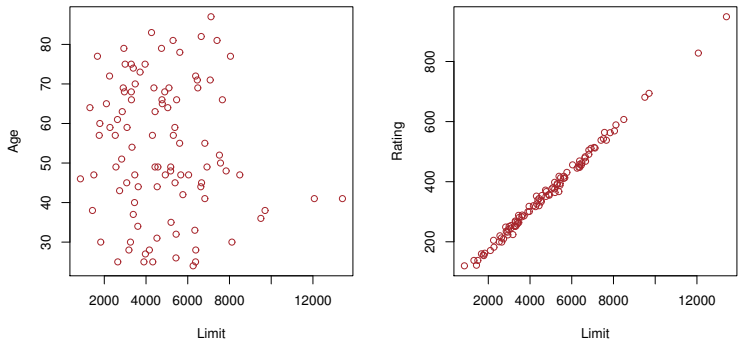
In this section we consider some methods for selecting subsets of predictors. These include best subset and stepwise model selection procedures.

## 2.1 Multicollinearity

### 2.1.1 Problem of multicollinearity

Multicollinearity refers to the situation in which two or more predictor variables are closely related to one another. The concept of multicollinearity is illustrated in following Figure using the Credit data set. In the left-hand panel of Figure, the two predictors limit and age appear to have no obvious relationship. In contrast, in the right-hand panel of Figure, the predictors limit and rating are very highly correlated with each other, and we say that they are collinear. The presence of multicollinearity can pose problems in the

regression context, since it can be difficult to separate out the individual effects of collinear variables on the response. In other words, since limit and rating tend to increase or decrease together, it can be difficult to determine how each one separately is associated with the response, balance.




In many situations which economic or financial data are used, the independent variables of regression are not orthogonal. When the linear relationship are too strong, the least squares solution is unstable. The estimates of the coefficients may change radically. Hoerl and Kennard's (1970a, b) proposed ridge regression to deal with most serious ill-effects of multicollinearity. Consider the standard model for multiple linear regression

$$Y = \mathbf{X}\beta + \varepsilon, \;\; \mathrm{E}(\varepsilon) = 0, \;\; \mathrm{E}(\varepsilon'\varepsilon) = \sigma^2\mathbf{I_n} \tag{23}$$

where $Y$ is an $(n \times 1)$ vector of observations on the dependent variable, $\mathbf{X}$ is an $(n \times p)$ matrix of observations on the explanatory variables and is of full rank $p$ $(p \leq n)$, $\beta$ is a $(p \times 1)$ unknown vector of regression coefficients. $\mathbf{X}'\mathbf{X}$ is in the form of a correlation matrix, and the vector $\mathbf{X}'Y$ is the vector of correlation coefficients of the dependent variable with each explanatory variable.

To minimize the residual sum of squares, the least squares estimate of $\beta$ is obtained by

$$b = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y \tag{24}$$

$b$ is an unbiased estimator, has minimum variance among unbiased linear eatimators, and follows the multivariate normal $(\beta, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$ distribution. However, if there exsits a strong linear relationship among the independent variables, then the variance becomes a large value. As a interpretation,

For example, a model that contains two explanatory variables and a constant.

$$\mathrm{Var}[b_1] = \frac{\sigma^2}{(1 - r_{12}^2)\Sigma_{i=1}^{n}(x_{i1} - \bar{x}_1)^2} \tag{25}$$

Figure [Farrar and Glauber (1967)]
Following the Johnston (1972) [p.160.], the effects of multicollinearity on the least squares estimates of the regression coefficients are as follows:

1. The precision of estimation falls so that it becomes very difficult, if not impossible, to disentangle the relative influences of various $x$ variables. This loss of precision has three aspects:a. Specific estimates may have very large errors;b. these errors may be highly correlated, one with another; b. and the sampling variances of the coefficients will be very large.

2. Investigators are sometimes led to drop variables incorrectly from an analysis because their coefficients are not significantly different from zero, but the ture situation may be not that a variable has no effect but simply that the set of sample data has not enabled us to pick it up.

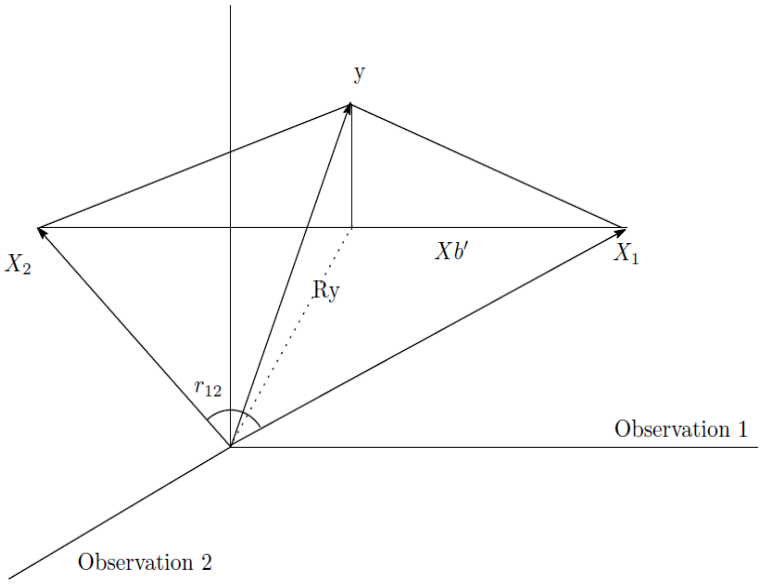


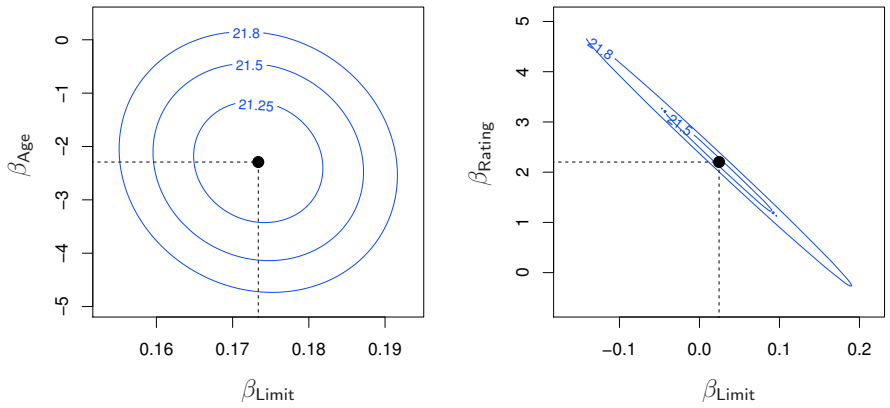

Figure 5: Image of the multicollinearity.

3. Estimates of coefficients become very sensitive to particular sets of sample data, and the addition of a few more observations can sometimes produce dramatic shift in some of the coefficients.

Following Greene (2002) [p.57.], the problem faced by applied researchers when regressors are highly, although not perfectlly, correlated included the following symptoms:

1. Small changes in the data produce wide seings in the parameter estimates.
2. Coefficients may have very high standard errors and low significance levels even though they are jointly significant and the $R^2$ for the regression is quite high.
3. Coefficients may have the "worng" sign or implausible magnitudes.

The following Figure illustrates some of the difficulties that can result from multicollinearity. The left-hand panel of Figure is a contour plot of the RSS associated with different possible coefficient estimates for the regression of balance on limit and age. Each ellipse represents a set of coefficients that correspond to the same RSS, with ellipses nearest to the center taking on the lowest values of RSS. The black dots and associated dashed lines represent the coefficient estimates that result in the smallest possible RSS — in other words, these are the least squares estimates. We see that the true limit coefficient is almost certainly somewhere between 0.15 and 0.20. In contrast, the right-hand panel of Figure displays contour plots of the RSS associated with possible coefficient estimates for the regression of balance onto limit and rating, which we know to be highly collinear. Now the contours run along a narrow valley; there is a broad range of values for the coefficient estimates that result in equal values for RSS. Hence a small change in the data could cause the pair of coefficient values that yield the smallest RSS — that is, the least squares estimates — to move anywhere along this valley. This results in a great deal of uncertainty in the coefficient estimates.




### 2.1.2 Measure of the multicollinearity

**Variance Inflation Factor (VIF)**

The VIF diagnostic statistic is defined as

$$\text{VIF} = \frac{1}{(1 - R_k^2)} \tag{26}$$

where $R_k^2$ is the $R^2$ in the regression of $x_k$ on all the other variables.

**Eigenvalue of X′X**

The total mean squared error of $b$ is given by

$$E[(b - \beta)'(b - \beta)] = \sigma^2 \sum_{j=1}^{p} \lambda_j^{-1} \tag{27}$$

If the original set of predictors were orthogonal, the total mean squared error would be $p\sigma^2$. The ratio

$$L^* = \sum_{j=1}^{p} (1/\lambda_j)/p \tag{28}$$

is the total mean squared error in $\hat{\beta}$, relative to what it would be if the independent variable were orthogonal. With this interpretation, $L^*$ is also an indicator of multicollinearity. [See Price, (1977)]

**Condition Number**

One way to measure the magnification factor is by means of the quantity called the condition number [Liu (2003)]. It is defined by

$$\kappa = (\text{largest eigenvalue}/\text{smallest eigenvalue})^{1/2} \tag{29}$$

A high condition number implies that $\mathbf{X'X}$ is ill conditioned. The effect of collinearity on least squares estimator is most severe when the condition number of $\mathbf{X'X}$ is high and the signal to noise ratio $\beta^{'}\beta/\sigma^2$ is small.

## 2.2 Ridge regression

### 2.2.1 Methodology

The remedy suggested by Hoerl and Kennard (1970a) is to accept some bias in trade for a reduction in variance. The ridge regression estimator is given as

$$\hat{\beta} = (\mathbf{X}'\mathbf{X} + \mathbf{K})^{-1}\mathbf{X}'Y \tag{30}$$

where the elements of elements of $\mathbf{X}'\mathbf{X}$ and $\mathbf{X}'Y$ are correlations,

$$\mathbf{K} = \begin{pmatrix} k_1 & 0 & \dots & 0 \\ 0 & k_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & k_p \end{pmatrix} \tag{31}$$

and the $k_i$ are small positive numbers. Since that $\mathbf{X}'e = 0$ (or $\mathbf{X}'Y = (\mathbf{X}'\mathbf{X})b$), Equation (30) can be shown as

$$\hat{\beta} = (\mathbf{X}'\mathbf{X} + \mathbf{K})^{-1}\mathbf{X}'\mathbf{X}b \tag{32}$$

As the elements of $\mathbf{K}$ grow, $\hat{\beta}$ approaches zero. The variance of $\hat{\beta}$ is $(\mathbf{X}'\mathbf{X} + \mathbf{K})^{-1}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X} + \mathbf{K})^{-1}\sigma^2$. The bias increases as $k_i$ increases. However, the variance decreases with $\mathbf{K}$ increases.

Ridge regression's advantage over least squares is rooted in the bias-variance trade-off . As $k$ increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias.

Also, the total variance of $\hat{\beta}$ is

$$TV[\hat{\beta}] = \sigma^2 \sum_{i=1}^{k} \frac{\lambda_i}{(\lambda_i + \Bbbk)^2} \tag{33}$$

where $\lambda_i$ is the eigenvalues of $\mathbf{X}'\mathbf{X}$.

Since Equation (33) is less than Equation (15), the Ridge estimator has a smaller total variance than the OLS estimator. Further, as is clearly shown in the previous, the Ridge estimator is not an unbiased estimator, and has a larger bias (absolute value) than the OLS estimator (i.e., bias$[\hat{\beta}] \neq 0$).

Furthermore, following Hoerl and Kennard (1970), if $L$ denotes the distance between $\hat{\beta}$ and $\beta$, then the MSE of $\hat{\beta}$ is

$$\begin{aligned} MSE[\hat{\beta}] = E[L'L(\Bbbk)] &= E[(\hat{\beta} - \beta)'(\hat{\beta} - \beta)] \\ &= \Bbbk^2 \beta'(\mathbf{X}'\mathbf{X} + \Bbbk\mathbf{I}_{\mathbf{k}})^{-2}\beta + \sigma^2 \sum_{i=1}^{k} \lambda_i/(\lambda_i + \Bbbk)^2 \\ &= \gamma_1(\Bbbk) + \gamma_2(\Bbbk) \end{aligned} \tag{34}$$

Then $\gamma_1(\Bbbk)$ can be considered as the square of a bias, and $\gamma_2(\Bbbk)$ can be indicated as the sum of the total variance of the Ridge estimator.

Differentiating $\gamma_1(\Bbbk)$ and $\gamma_2(\Bbbk)$ with respect to $\Bbbk$, and put $\Bbbk \to 0^+$, we obtain

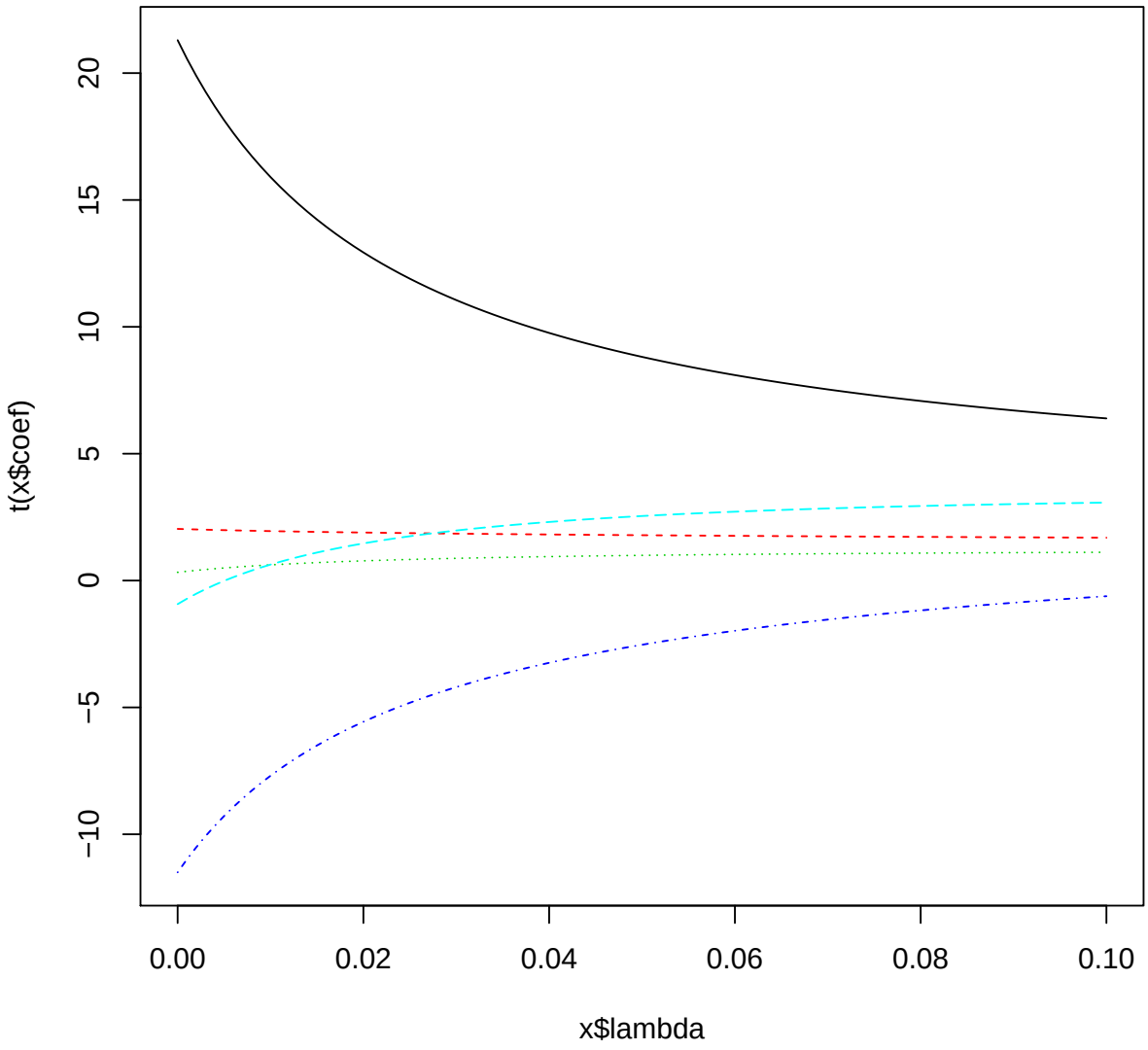
$$\begin{aligned} \lim_{\Bbbk \to 0^+} (d\gamma_1/d\Bbbk) &= 0 \\ \lim_{\Bbbk \to 0^+} (d\gamma_2/d\Bbbk) &= -2\sigma^2 \Sigma_{i=1}^{k} (1/\lambda_i^2) \end{aligned}$$

$\gamma_2(\Bbbk)$has a negative derivative which attains $-2\Bbbk\sigma^2$ when $\Bbbk \to 0^+$ for an orthogonal $\mathbf{X}'\mathbf{X}$, and attains $-\infty$ when $\mathbf{X}'\mathbf{X}$ becomes ill-conditioned. These properties indicate that it is possible to move to $\Bbbk > 0$ (close to

zero), take a small bias, and substantially reduce the variance, finally improving the MSE of estimation and prediction. Namely, although the ridge regression estimator is biased, if we accept some bias in trade for a reduction in variance, the ridge regression estimator may have smaller MSE than the OLS estimator even if multicollinearity does not exist. From this viewpoint, many recent studies on the small sample properties of the ridge regression estimators have been made. [Also see, for example, Dwivedi et al. (1980) and Kozumi and Ohtani (1994).]

### 2.2.2 An Example

```
#install.packages("MASS")
library("MASS")

## Warning:  package 'MASS' was built under R version 3.2.5

library("car")
load("longley.RData")
lm.normal<-lm(y~.,data=longley)
summary(lm.normal)

##
## Call:
## lm(formula = y ~ ., data = longley)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.07502 -0.43246  0.08478  0.50481  1.55564
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  225.800117  81.873096   2.758  0.02020 *
## GNP            0.221305   0.060903   3.634  0.00458 **
## Unemployed     0.022501   0.008190   2.747  0.02057 *
## Armed.Forces   0.004825   0.007800   0.619  0.55000
## Population    -1.707504   0.644747  -2.648  0.02438 *
## Employed      -0.273425   0.746137  -0.366  0.72166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.148 on 10 degrees of freedom
## Multiple R-squared:  0.9925,Adjusted R-squared:  0.9887
## F-statistic: 263.2 on 5 and 10 DF,  p-value: 2.836e-10

vif(lm.normal)   #>10

##          GNP   Unemployed Armed.Forces   Population     Employed
##   417.195026     6.668525     3.354687   229.002459    78.175038

mean(vif(lm.normal))   #>2

## [1] 146.8791

XX<-cor(longley[2:6])
kappa(XX,exact=T)   #>1000

## [1] 2300.046

lm.r<-lm.ridge(y ~ ., longley)
plot(lm.ridge(y ~ ., longley, lambda = seq(0,0.1,0.001)))
```




```
select(lm.ridge(y ~ ., longley, lambda = seq(0,0.1,0.0001)))

## modified HKB estimator is 0.006688179
## modified L-W estimator is 0.03647802
## smallest value of GCV  at 0.0028
```

### 2.2.3 Monte Carlo Simulation

## An Example

```
library("MASS")
#1
set.seed(1234)
```

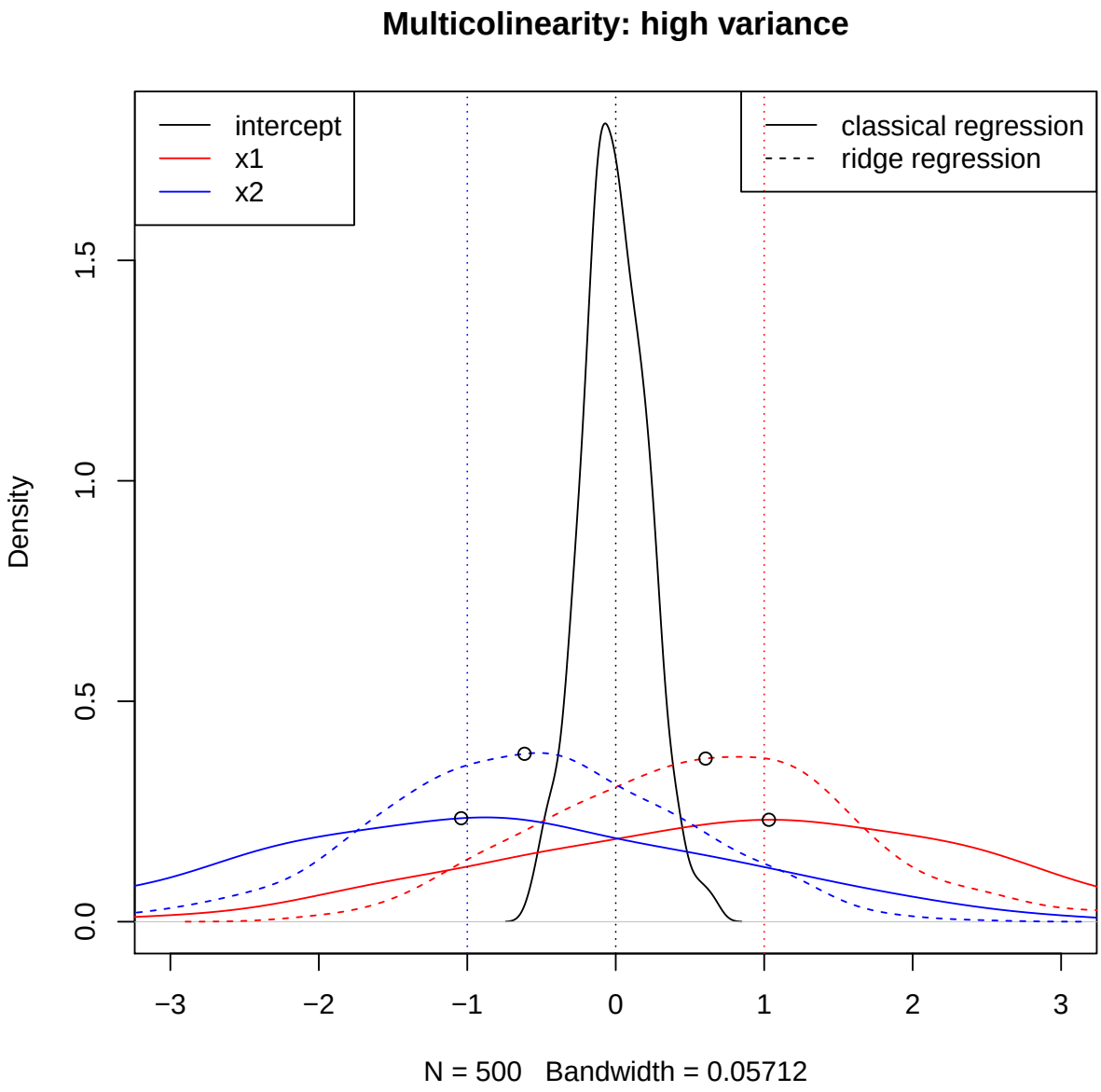
```r
x1 <- rnorm(20)
x2 <- rnorm(20,mean=x1,sd=.01)
y <- rnorm(20,mean=3+x1+x2)
lm(y~x1+x2)$coef
```

```
## (Intercept)          x1          x2
##     2.16879    50.38647   -48.78379
```

```r
lm.ridge(y~x1+x2,lambda=1)
```

```
##                   x1        x2
## 2.4710161 0.8605031 0.8062424
```

```r
#2
my.sample <- function (n=20) {
        x <- rnorm(n)
        x1 <- x + .1*rnorm(n)
        x2 <- x + .1*rnorm(n)
        y <- 0 + x1 - x2 + rnorm(n)
        cbind(y, x1, x2)
}
n <- 500
r <- matrix(NA, nr=n, nc=3)
s <- matrix(NA, nr=n, nc=3)
corr <- matrix(NA, nr=n, nc=3)
for (i in 1:n) {
        m <- my.sample()
        corr[i,] <- cor(m[,2],m[,3])
        r[i,] <- lm(m[,1]~m[,-1])$coef
        s[i,2:3] <- lm.ridge(m[,1]~m[,-1], lambda=.1)$coef
        s[i,1] <- mean(m[,1])
}
plot( density(r[,1]), xlim=c(-3,3),
      main="Multicolinearity: high variance")
abline(v=0, lty=3)
lines( density(r[,2]), col="red" )
lines( density(s[,2]), col="red", lty=2 )
abline(v=1, col="red", lty=3)
lines( density(r[,3]), col="blue" )
lines( density(s[,3]), col="blue", lty=2 )
abline(v=-1, col="blue", lty=3) # We give the mean, to show that it is biased
evaluate.density <- function (d,x, eps=1e-6) {
        density(d, from=x-eps, to=x+2*eps, n=4)$y[2]
}
x<-mean(r[,2]); points( x, evaluate.density(r[,2],x) )
x<-mean(s[,2]); points( x, evaluate.density(s[,2],x) )
x<-mean(r[,3]); points( x, evaluate.density(r[,3],x) )
x<-mean(s[,3]); points( x, evaluate.density(s[,3],x) )
legend( par("usr")[1], par("usr")[4], yjust=1,
        c("intercept", "x1", "x2"),
        lwd=1, lty=1,
        col=c(par("fg"), "red", "blue"))
legend( par("usr")[2], par("usr")[4], yjust=1, xjust=1,
        c("classical regression", "ridge regression"),
        lwd=1, lty=c(1,2),
        col=par("fg") )
```

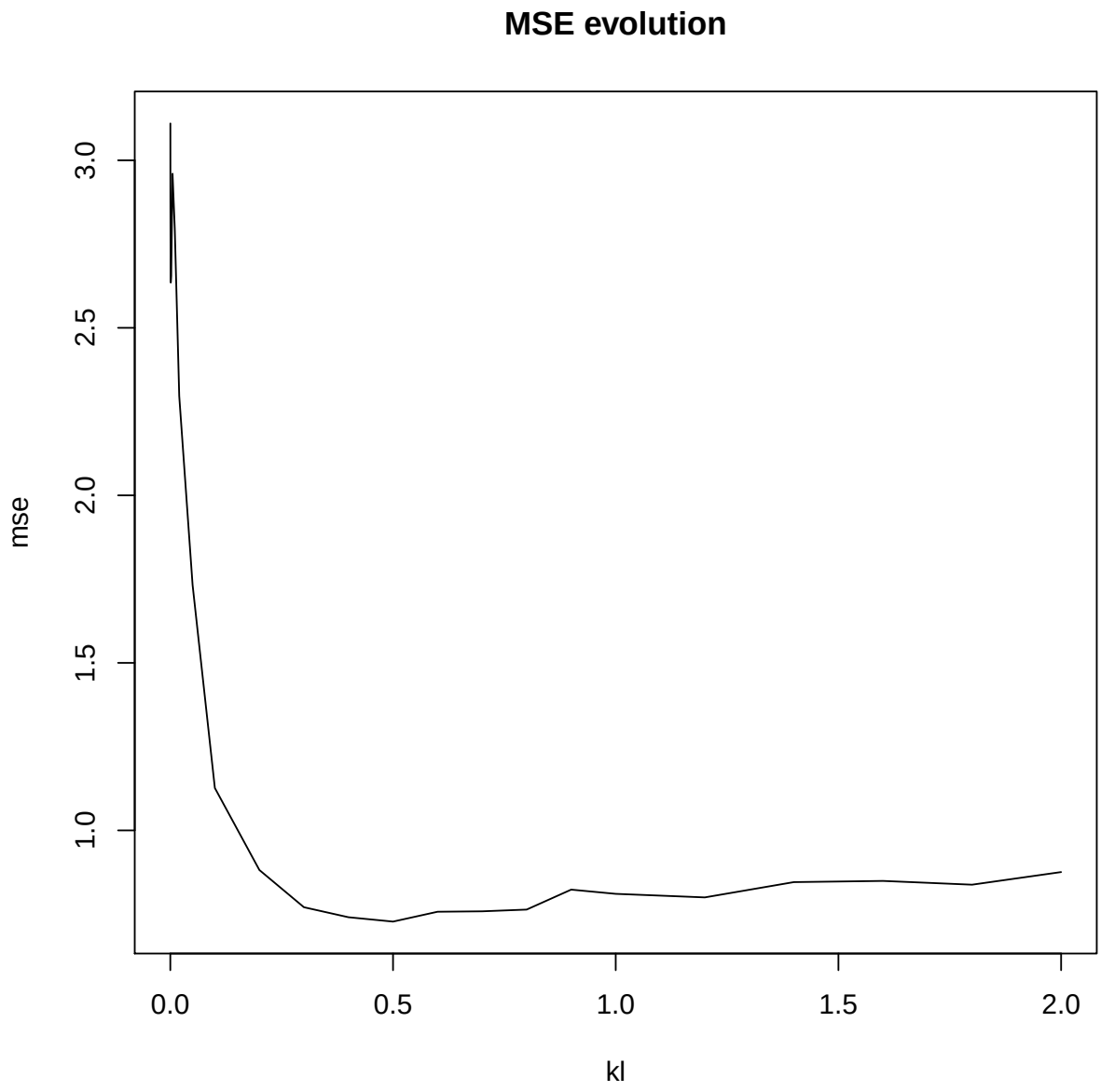



We can compute the mean and variance of our estimators:

```
# LS regression
apply(r,2,mean)
## [1] -0.006073365  1.031650733 -1.041272053
apply(r,2,var)
## [1] 0.04837863 3.26642031 3.26225379
# Ridge regression
apply(s,2,mean)
## [1] -0.007218683  0.605057959 -0.614373473
apply(s,2,var)
## [1] 0.04305713 1.09458938 1.08522375
```

But we cannot compare them that way; to do so, we prefer using the Mean Square Error (MSE): it is the mean of the squares of the differences of the real value of the looked-for parameter.

```
n <- 500
v <- matrix(c(0,1,-1), nr=n, nc=3, byrow=T)
mse <- NULL
kl <- c(1e-4, 2e-4, 5e-4, 1e-3, 2e-3, 5e-3, 1e-2, 2e-2, 5e-2, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1,
for (k in kl) {
r <- matrix(NA, nr=n, nc=3)
for (i in 1:n) {
m <- my.sample()
r[i,2:3] <- lm.ridge(m[,1]~m[,-1], lambda=k)$coef
r[i,1] <- mean(m[,1])
}
mse <- append(mse, apply( (r-v)^2, 2, mean )[2])
}
plot( mse ~ kl, type="l" )
title(main="MSE evolution")
```




On this example, we see that $k = 0.5$ is a good value (but according to the literature, it is VERY high: they say you should not go beyond 0.1). You can also choose k from a plot: say, the parameters (or something that depends on the parameters, such as $R^2$) with respect to $k$.

## 2.3 Model selection criteria

Several criteria are used for this purpose. In particular, we discuss these criteria: (1) $R^2$, (2) adjusted $R^2$ ($=\bar{R}^2$), (3) Akaike information criterion (AIC), (4) Schwarz information criteria (SIC). All these criteria aim at minimizing the residual sum of squares (RRS) (or increasing the $R^2$ value). However, except for the first criterion, criteria (2), (3), and (4) impose a penalty for including an increasingly large number of regressors. Thus there is a tradeoff between goodness of fit of the model and its complexity (as judged by the number of regressors).

**The $R^2$ Criterion**

Consider the following model:

$$y_i = \beta_0 + \beta_1 x_i + u_i, \quad i = 1, 2 \ldots, n$$

We want to minimize

$$RSS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \tag{35}$$

Differentiating 35 with respect to $\hat{\beta}$

$$\begin{aligned} \frac{\partial RSS}{\partial \hat{\beta}_0} &= -2 \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0 \\ \frac{\partial RSS}{\partial \hat{\beta}_1} &= -2 \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) x_i = 0 \end{aligned}$$

We have

$$\sum_{i=1}^{n} e_i = 0$$

$$\sum_{i=1}^{n} x_i e_i = 0$$

$$\sum_{i=1}^{n} \hat{y}_i e_i = 0$$

Total movement can be expressed as

$$\begin{aligned} \sum_{i=1}^{n} (y_i - \bar{y})^2 &= \sum_{i=1}^{n} (\hat{y}_i + e_i - \bar{y})^2 \\ &= \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n} e_i^2 + 2 \sum_{i=1}^{n} (\hat{y}_i - \bar{y}) e_i \end{aligned}$$

Since third item of right hand is zero, we have

$$\sum_{i=1}^{n} (y_i - \bar{y})^2 = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n} e_i^2 + 2 \sum_{i=1}^{n} (\hat{y}_i - \bar{y}) e_i$$

$R^2$ can be defined as

$$0 \leq R^2 = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} = \frac{\text{ESS}}{\text{TSS}} = 1 - \frac{\sum_{i=1}^{n} e_i^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} = 1 - \frac{\text{RSS}}{\text{TSS}} \leq 1$$

But there are problems with $R^2$. First, it measures in-sample goodness of fit in the sense of how close an estimated Y value is to its actual value in the given sample. There is no guarantee that it will forecast well out-of-sample observations. Second, in comparing two or more $R^2$'s, the dependent variable, or regress and, must be the same. Third, and more importantly, an $R^2$ cannot fall when more variables are added to the model. Therefore, there is every temptation to play the game of "maximizing the $R^2$" by simply adding more variables to the model. Of course, adding more variables to the model may increase $R^2$ but it may also increase the variance of forecast error.

### Adjust $R^2$

As a penalty for adding regressors to increase the $R^2$ value, Henry Theil developed the adjusted $R^2$, denoted by $\bar{R}^2$.

$$\bar{R}^2 = 1 - \frac{\text{RSS}/(n-k)}{\text{TSS}/(n-1)} = 1 - (1-R^2)\frac{n-1}{n-k}$$

the adjusted $R^2$ will increase only if the absolute t value of the added variable is greater than 1.

```
slm<-summary(lm)
lm$r.squared
lm$adj.r.quared
```

## $C_p$ value

For a fitted least squares model containing $k$ predictors, the $C_p$ estimate of test MSE is computed using the equation

$$C_p = \frac{1}{n}(\text{RSS} + 2k\hat{\sigma}^2)$$

where $\hat{\sigma}^2$ is an estimate of the variance of the error ǫ associated with each response measurement. Essentially, the $C_p$ statistic adds a penalty of $2k\hat{\sigma}^2$ to the training RSS in order to adjust for the fact that the training error tends to underestimate the test error. Clearly, the penalty increases as the number of predictors in the model increases; this is intended to adjust for the corresponding decrease in training RSS.

### Akaike Information Criterion (AIC)

The AIC criterion is defined for a large class of models fit by maximum likelihood, which is defined as:

$$\text{AIC} = (\frac{2k}{n}) + \ln(\frac{\text{RSS}}{n})$$

where $k$ is the number of regressors (including the intercept), $n$ is the number of observations, $2k/n =$ penalty factor. In comparing two or more models, the model with the lowest value of AIC is preferred.

```
mod1 = lm(y ~ x1 + ... + xk, data=ds)
AIC(mod1)
```

### Bayesian information criterion (BIC)

Similar in spirit to the AIC, the BIC criterion is defined as:

$$\ln \text{SIC} = \frac{k}{n}\ln n + \ln(\frac{\text{RSS}}{n})$$

where $[(k/n)\ln n]$ is the penalty factor. BIC imposes a harsher penalty than AIC. Like AIC, the lower the value of BIC, the better the model. The BIC statistic generally places a heavier penalty on models with many variables, and hence results in the selection of smaller models than $C_p$.

```
library(nlme)
mod1 = lm(y ~ x1 + ... + xk, data=ds)
BIC(mod1)
```

## 2.4 Best Subset Selection

To perform best subset selection, we fit a separate least squares regression for each possible combination of the $k$ predictors. That is, we fit all $k$ models that contain exactly one predictor, all $\begin{pmatrix} k \\ 2 \end{pmatrix} = k(k-1)/2$ models that contain exactly two predictors, and so forth. We then look at all of the resulting models, with the goal of identifying the one that is best.

The problem of selecting the best model from among the $2^p$ possibilities considered by best subset selection is not trivial. This is usually broken up into two stages.

### Algorithm: Best Subset Selection

For $p = 1, 2, ...k$:

1. Fit all $\begin{pmatrix} k \\ p \end{pmatrix}$ models that contain exactly $p$ predictors.
2. Pick the best among these $\begin{pmatrix} k \\ p \end{pmatrix}$ models. Here best is defined as having the smallest RSS, or equivalently largest $R^2$.

While best subset selection is a simple and conceptually appealing approach, it suffers from computational limitations. The number of possible models that must be considered grows rapidly as $k$ increases. In general, there are $2^k$ models that involve subsets of $k$ predictors. So if $k = 10$, then there are approximately one thousand possible models to be considered, and if $k = 20$, then there are over one million possibilities! Consequently, best subset selection becomes computationally infeasible for values of $k$ greater than around 40, even with extremely fast modern computers.

#### An Example

```
library(ISLR)

## Warning:  package 'ISLR' was built under R version 3.2.4

names(Hitters)

##  [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "RBI"
##  [6] "Walks"     "Years"     "CAtBat"    "CHits"     "CHmRun"
## [11] "CRuns"     "CRBI"      "CWalks"    "League"    "Division"
## [16] "PutOuts"   "Assists"   "Errors"    "Salary"    "NewLeague"

dim(Hitters)

## [1] 322  20

sum(is.na(Hitters$Salary))

## [1] 59

Hitters=na.omit(Hitters)
dim(Hitters)

## [1] 263  20

sum(is.na(Hitters))

## [1] 0
```

```r
library(leaps)
regfit.full=regsubsets(Salary~.,data=Hitters,nvmax=19)
#summary(regfit.full)
reg.summary=summary(regfit.full)
names(reg.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```r
reg.summary$adjr2
```

```
##  [1] 0.3188503 0.4208024 0.4450753 0.4672734 0.4808971 0.4972001 0.5007849
##  [8] 0.5137083 0.5180572 0.5222606 0.5225706 0.5217245 0.5206736 0.5195431
## [15] 0.5178661 0.5162219 0.5144464 0.5126097 0.5106270
```

```r
reg.summary$cp
```

```
##  [1] 104.281319  50.723090  38.693127  27.856220  21.613011  14.023870
##  [7]  13.128474   7.400719   6.158685   5.009317   5.874113   7.330766
## [13]   8.888112  10.481576  12.346193  14.187546  16.087831  18.011425
## [19]  20.000000
```

```r
reg.summary$bic
```

```
##  [1]  -90.84637 -128.92622 -135.62693 -141.80892 -144.07143 -147.91690
##  [7] -145.25594 -147.61525 -145.44316 -143.21651 -138.86077 -133.87283
## [13] -128.77759 -123.64420 -118.21832 -112.81768 -107.35339 -101.86391
## [19]  -96.30412
```

```r
par(mfrow=c(2,2))
plot(reg.summary$rss,xlab="Number of Variables",ylab="RSS",type="l")
plot(reg.summary$adjr2,xlab="Number of Variables",ylab="Adjusted RSq",type="l")
which.max(reg.summary$adjr2)
```

```
## [1] 11
```

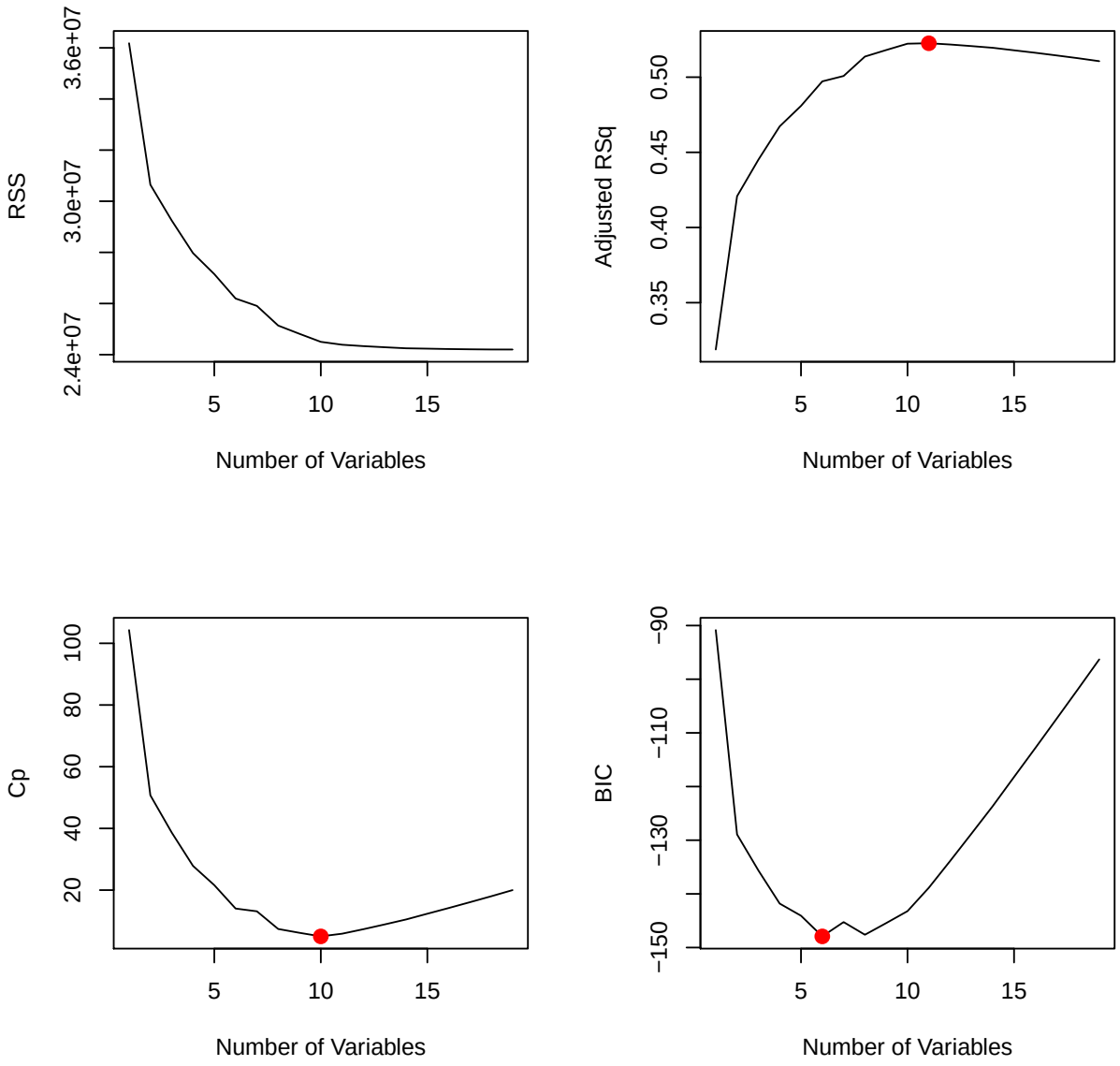
```r
points(11,reg.summary$adjr2[11], col="red",cex=2,pch=20)
plot(reg.summary$cp,xlab="Number of Variables",ylab="Cp",type='l')
which.min(reg.summary$cp)
```

```
## [1] 10
```

```r
points(10,reg.summary$cp[10],col="red",cex=2,pch=20)
which.min(reg.summary$bic)
```

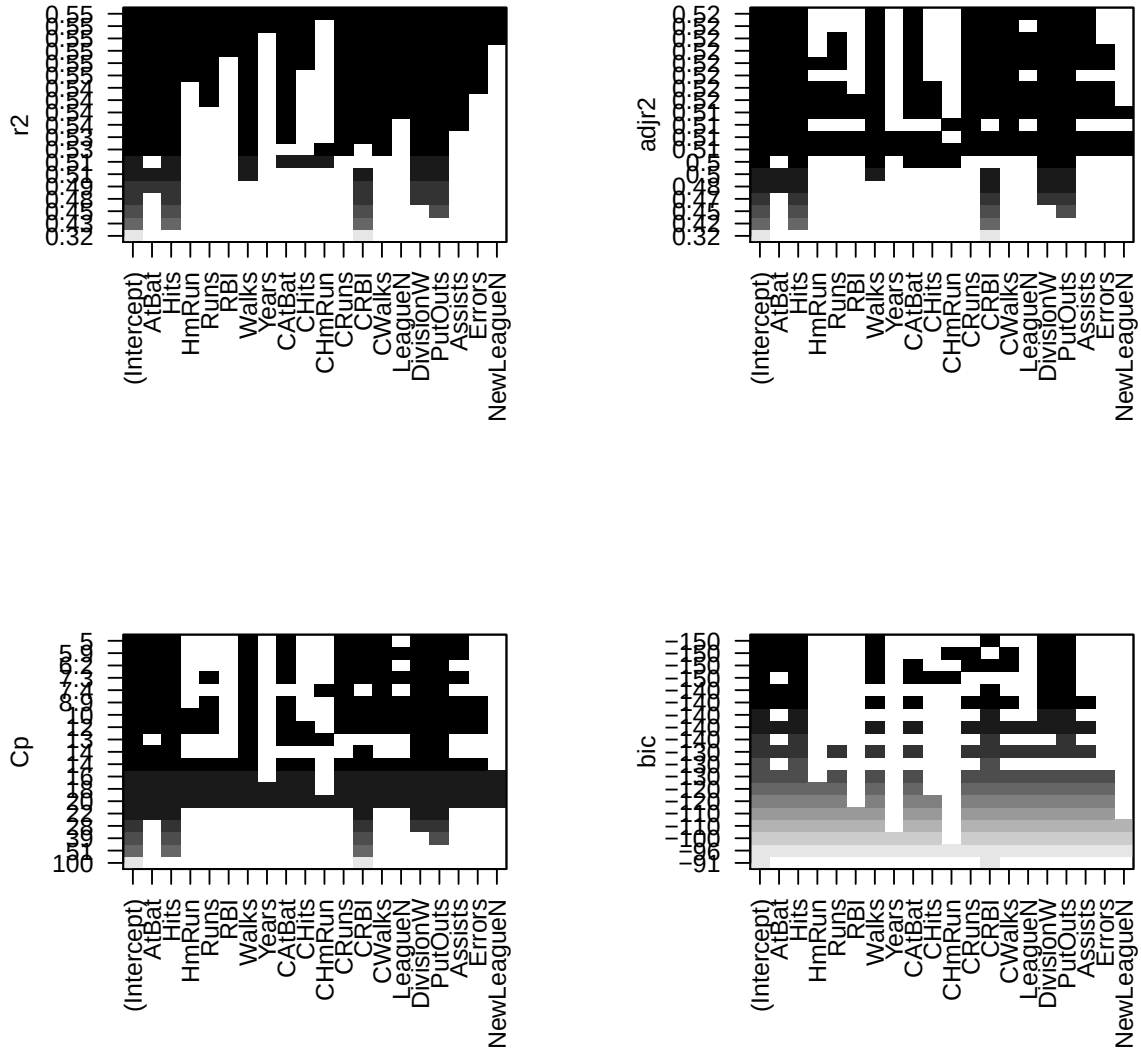
```
## [1] 6
```

```r
plot(reg.summary$bic,xlab="Number of Variables",ylab="BIC",type='l')
points(6,reg.summary$bic[6],col="red",cex=2,pch=20)
```




The regsubsets() function has a built-in plot() command which can be used to display the selected variables for the best model with a given number of predictors, ranked according to the BIC, $C_p$, adjusted $R^2$, or AIC.

```
par(mfrow=c(2,2))
plot(regfit.full,scale="r2")
plot(regfit.full,scale="adjr2")
plot(regfit.full,scale="Cp")
plot(regfit.full,scale="bic")
```




The top row of each plot contains a black square for each variable selected according to the optimal model associated with that statistic. For instance, we see that several models share a BIC close to −150. However, the model with the lowest BIC is the six-variable model that contains only AtBat, Hits, Walks, CRBI, DivisionW, and PutOuts. We can use the coef() function to see the coefficient estimates associated with this model.

```
coef(regfit.full,6)

##  (Intercept)        AtBat         Hits        Walks         CRBI
##   91.5117981   -1.8685892    7.6043976    3.6976468    0.6430169
##    DivisionW      PutOuts
## -122.9515338    0.2643076
```

## 2.5 Stepwise

**Selecting algorithm:**

Stepwise regression, a method for feature selection that involves selecting features according to some selection criterion by either adding or subtracting features to a regression model in a systematic way. There are three primary methods of stepwise regression: forward selection, backward elimination, and a combined approach (forward and backward).

**Forward selection:**

In forward selection you start with a regression model with no features, and gradually add one feature at a time according to which feature improves the model the most based on a selection criterion. This looks like this: build all possible regression models with a single predictor. Pick the best. Now try all possible models that include that best predictor and a second predictor. Pick the best of those. You keep adding one feature at a time, and you stop when your selection criterion no longer improves, but instead gets worse.

**Backward elimination:**

In backward elimination you start with a regression model that includes all the features, and you gradually remove one feature at a time according to the feature whose removal makes the biggest improvement in the selection criterion. You stop removing features when removing the feature makes the selection criterion get worse.

**Combined approach:**

Most subset methods are capturing some flavor of minimumredundancy- maximum-relevance. So, for example, you could have a greedy algorithm that starts with the best feature, takes a few more highly ranked, removes the worst, and so on. This a hybrid approach with a filter method.

**An Example**

```
cement<-data.frame( X1=c( 7, 1, 11, 11, 7, 11, 3, 1, 2, 21, 1, 11, 10),
X2=c(26, 29, 56, 31, 52, 55, 71, 31, 54, 47, 40, 66, 68),
X3=c( 6, 15, 8, 8, 6, 9, 17, 22, 18, 4, 23, 9, 8),
X4=c(60, 52, 20, 47, 33, 22, 6, 44, 22, 26, 34, 12, 12),
Y =c(78.5, 74.3, 104.3, 87.6, 95.9, 109.2, 102.7, 72.5, 93.1,115.9, 83.8, 113.3, 109.4)
)
lm.sol<-lm(Y ~ X1+X2+X3+X4, data=cement)
summary(lm.sol)

##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4, data = cement)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1750 -1.6709  0.2508  1.3783  3.9254
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  62.4054    70.0710   0.891   0.3991
## X1            1.5511     0.7448   2.083   0.0708 .
## X2            0.5102     0.7238   0.705   0.5009
## X3            0.1019     0.7547   0.135   0.8959
```

```
## X4           -0.1441     0.7091  -0.203   0.8441
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.446 on 8 degrees of freedom
## Multiple R-squared:  0.9824,Adjusted R-squared:  0.9736
## F-statistic: 111.5 on 4 and 8 DF,  p-value: 4.756e-07

lm.step<-step(lm.sol)

## Start:  AIC=26.94
## Y ~ X1 + X2 + X3 + X4
##
##        Df Sum of Sq    RSS    AIC
## - X3    1    0.1091 47.973 24.974
## - X4    1    0.2470 48.111 25.011
## - X2    1    2.9725 50.836 25.728
## <none>              47.864 26.944
## - X1    1   25.9509 73.815 30.576
##
## Step:  AIC=24.97
## Y ~ X1 + X2 + X4
##
##        Df Sum of Sq    RSS    AIC
## <none>              47.97 24.974
## - X4    1      9.93  57.90 25.420
## - X2    1     26.79  74.76 28.742
## - X1    1    820.91 868.88 60.629

summary(lm.step)

##
## Call:
## lm(formula = Y ~ X1 + X2 + X4, data = cement)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0919 -1.8016  0.2562  1.2818  3.8982
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  71.6483    14.1424   5.066 0.000675 ***
## X1            1.4519     0.1170  12.410 5.78e-07 ***
## X2            0.4161     0.1856   2.242 0.051687 .
## X4           -0.2365     0.1733  -1.365 0.205395
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.309 on 9 degrees of freedom
## Multiple R-squared:  0.9823,Adjusted R-squared:  0.9764
## F-statistic: 166.8 on 3 and 9 DF,  p-value: 3.323e-08

drop1(lm.step)

## Single term deletions
```

```
##
## Model:
## Y ~ X1 + X2 + X4
##        Df Sum of Sq    RSS    AIC
## <none>               47.97 24.974
## X1      1    820.91 868.88 60.629
## X2      1     26.79  74.76 28.742
## X4      1      9.93  57.90 25.420

lm.opt<-lm(Y ~ X1+X2, data=cement); summary(lm.opt)

##
## Call:
## lm(formula = Y ~ X1 + X2, data = cement)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.893 -1.574 -1.302  1.363  4.048
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 52.57735    2.28617   23.00 5.46e-10 ***
## X1           1.46831    0.12130   12.11 2.69e-07 ***
## X2           0.66225    0.04585   14.44 5.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.406 on 10 degrees of freedom
## Multiple R-squared:  0.9787,Adjusted R-squared:  0.9744
## F-statistic: 229.5 on 2 and 10 DF,  p-value: 4.407e-09
```

### Update the liner models

```
x1=rnorm(100)
x2=rnorm(100)
x3=rnorm(100)
y=rnorm(100)
fm2 <- lm(y ~ x1 + x2)
fm3 <- update(fm2, . ~ . + x3)   #add x3 to the fit model
smf3 <- update(fm3, I(y^2) ~ .)   #change y to y^2
```

# Appendix

## Mean Squared Error (MSE)

Mean squared error of estimator are defined as follow:

$$\begin{aligned} MSE(\hat{\theta}) &= E[(\hat{\theta}-\theta)^2] \\ &= E[\{(\hat{\theta}-E[\hat{\theta}])+(E[\hat{\theta}]-\theta)\}^2] \\ &= E[(\hat{\theta}-E[\hat{\theta}])^2] \\ &\quad +2E[(\hat{\theta}-E[\hat{\theta}])(E[\hat{\theta}]-\theta)]+(E[\hat{\theta}]-\theta)^2] \\ &= E[(\hat{\theta}-E[\hat{\theta}])^2]+2(E[\hat{\theta}]-E[\hat{\theta}])(E[\hat{\theta}]-\theta)]+(E[\hat{\theta}]-\theta)^2 \\ &= V(\hat{\theta})+(E[\hat{\theta}]-\theta)^2 \\ &= variance+Bias^2 \end{aligned} \tag{36}$$