R Lab Manual for Biostatistics II

Spring 2012

Department of Biostatistics

Fay W. Boozman College of Public Health

University of Arkansas for Medical Sciences

Josh Callaway, MPH

D. Keith Williams, PhD

Zoran Bursac, PhD

Table of Contents

INTRODUCTION

The purpose of this manual is to provide a continuing R education for those who already possess a fairly firm foundation in Introductory Statistics and how to implement introductory level data analysis in R. It is assumed here that the students can implement R in the following statistical areas: data input, graphs and descriptive statistics, probability and probability distributions, binomial and poisson distributions, normal distributions, sampling distributions, large sample estimation, large sample hypothesis testing, small sample inference, and categorical data. Therefore, this handbook will not contain an in depth discussion of how to import datasets into R nor discuss any of the above-mentioned topics in detail. This manual also assumes a basic understanding of what R is and why it is useful, so this contains no detailed description of R. Rather, this manual focuses on analyzing data at the next level. Therefore, one of the main purposes here is to teach the student how to build models, interpret the models, and select the best fit models. Areas of focus include: multiple regression, randomized block designs, modeling with interaction terms, introduction to logistic regression and generalized linear models, and analysis of longitudinal data. This manual follows the Biostatistics II syllabus so students can follow it throughout the semester. Most of this manual contains text that is verbatim from *Introduction to Probability and Statistics 13th Edition* (Mendenhall, Beaver, & and Beaver, 2009). However, all of the examples are explained how to analyze in R instead of MINITAB. The logistic regression supplements are from Hosmer supplements. Repeated Measures text is reproduced exactly from *Modeling Longitudinal Data* (Sudha Purohit). To create a user-friendly approach for students, this manual highlights R script code in `red` and output text in `blue`. Some output is strictly one or more figures, so it is not followed by `blue` output text but by one or more figures. It is also important to note here for the reader that all script entered in the R Console is preceded by the "`>`" symbol. However, when showing script in this manual, the "`>`" symbol is omitted. In addition, when a new line of the same command is entered into the R console, it is preceded by the "`+`" symbol. The manual omits any use of this symbol as well.

CHAPTER 12: LINEAR REGRESSION AND CORRELATION

12.2 A SIMPLE LINEAR PROBABILISTIC MODEL (p. 503 in text)

This model describes a deterministic relationship between the variable of interest y, sometimes called the **response variable**, and the independent variable x, often called the **predictor variable**. A particular response y is described using the **probabilistic model**:

$$y = \alpha + \beta x + \varepsilon$$

Table 12.1 displays the mathematics achievement test scores for a random sample of n=10 college freshmen, along with their final calculus grades.

Table 12.1: Mathematics Achievement Test Scores and Final Calculus Grades for College Freshmen

Student	Mathematics Achievement Test Score	Final Calculus Grade
1	39	65
2	43	78
3	21	52
4	64	82
5	57	92
6	47	89
7	28	73
8	75	98
9	34	56
10	52	75

To reproduce this table in R, we simply input the data into variable names and store them using `data.frame`:

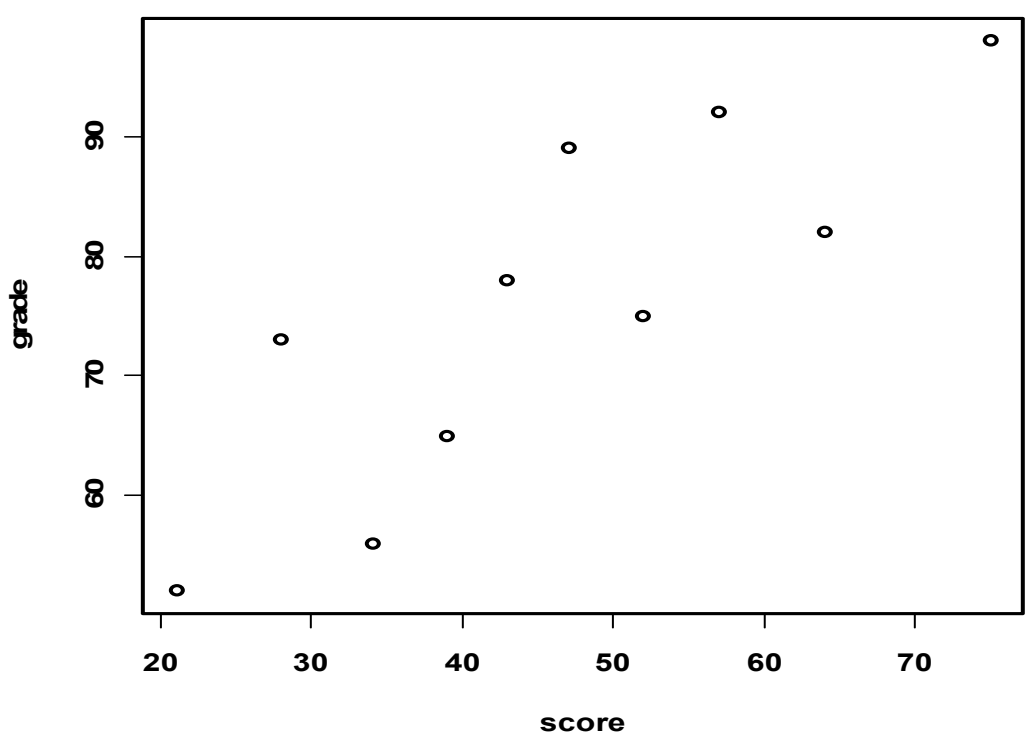
```
score = c(39,43,21,64,57,47,28,75,34,52)
grade = c(65,78,52,82,92,89,73,98,56,75)
freshmen = data.frame(score,grade)
freshmen
```

```
   score grade
1     39    65
2     43    78
3     21    52
4     64    82
5     57    92
6     47    89
7     28    73
8     75    98
9     34    56
10    52    75
```

To reproduce Figure 12.2 (p. 505) from the text, we can choose one of two functions, `plot` (which is a low-power function) or `scatterplot` (which is a high-power function). If a function is higher-power, it basically means it can give you more output.

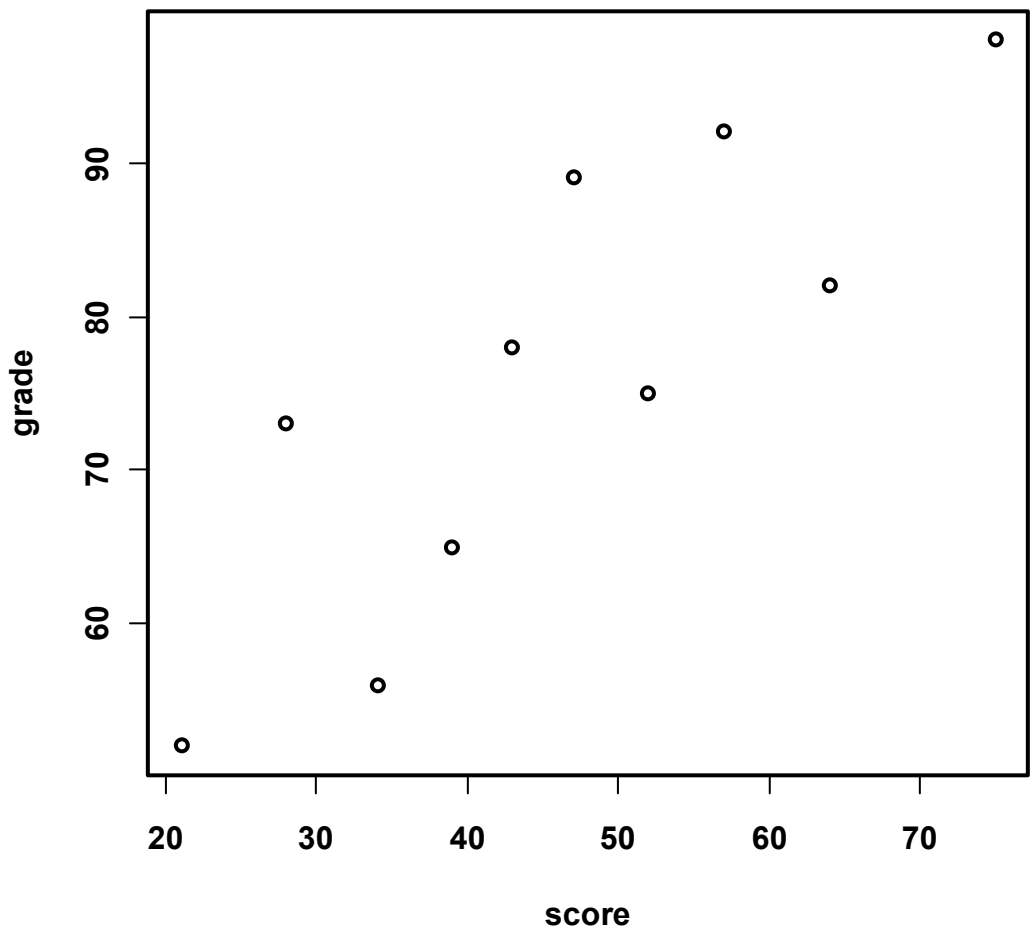
```
par(font=2, font.axis =2, font.lab=2)

plot(score,grade)
```

Figure 12.2: Scatterplot of the data in Table 12.1 (using `plot`)



```
par(font=2, font.axis =2, font.lab=2)

scatterplot(grade~score,smooth=FALSE,reg.line=FALSE,
            boxplots=FALSE,grid=FALSE,data=freshmen)
```

Figure 12.2: Scatterplot of the data in Table 12.1 (using `scatterplot`)



Notice that in order to produce the same figure from `plot` using `scatterplot`, we have to set several within function commands equal to `FALSE` due to the high-power capabilities of `scatterplot`. To explore within function commands for both `plot` and `scatterplot`, simply type: `?plot` or `?scatterplot` (assuming you are connected to the internet).

12.3 THE METHOD OF LEAST SQUARES (p. 506 in text)

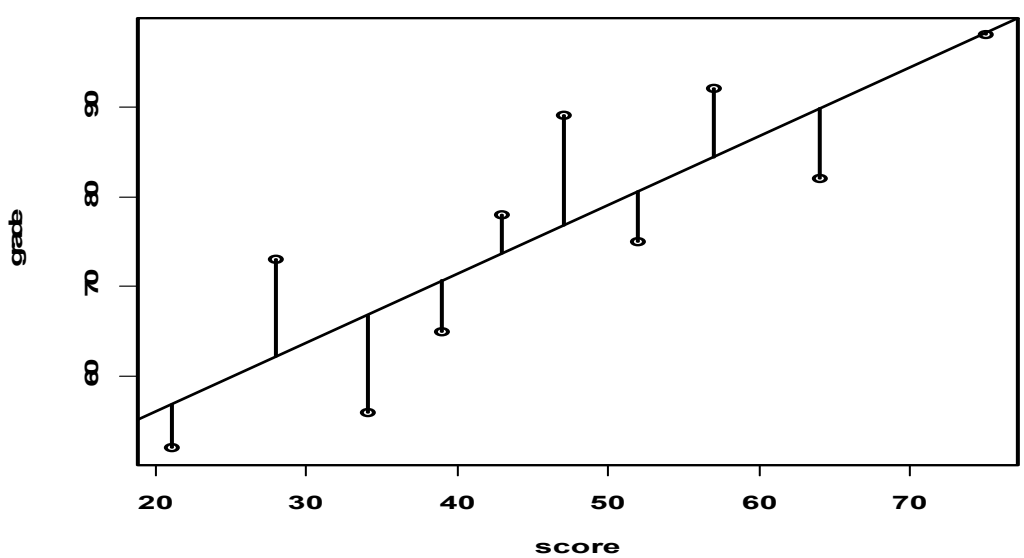
The formula for the best-fitting line is:

$$\hat{y} = a + bx$$

where a and b are the estimates of the intercept and slope parameters α and β, respectively. To give a crude reproduction of Figure 12.4 (p. 506) in R, we first need to store our data into a linear model using the `lm` function. Then we re-use the `plot` command, followed by `abline` for the best-fitting line, and finally use `segments` to draw lines representing the residual deviance of each data point from the best-fitting line. The `par` function simply adjusts graphical visual parameters for our choosing: font gives overall font for text, `font.axis` gives font for numbers on the x and y axes, `font.lab` gives the font for x and y labels, and `lwd` gives the line width. To learn more ways to tinker with these parameters, simply run `?par`.

```
fit <- lm(grade~score,data=freshmen)
par(font=2,font.axis=2,font.lab=2, lwd=2)
plot(score,grade)
abline(fit, lwd=2)
segments(score,fitted(fit),score,grade)
```

Figure 12.4: Method of Least Squares



Principle of Least Squares

The line that minimizes the sum of squares of the deviations of the observed values of y from those predicted is the **best-fitting line**. The sum of squared deviations is commonly called the **sum of squares for error** (SSE) and defined as:

$$SSE = \sum y_i - \hat{y}_i = \sum (y_i - a - bx_i)^2$$

Least-Squares Estimators of α and β

$b = \frac{S_{xy}}{S_{xx}}$ and $a = \bar{y} - b\bar{x}$

where the quantities S_{xy} and S_{xx} are defined as

$$S_{xy} = \sum (x_i - \bar{x})^2 (y_i - \bar{y}) = \sum x_i\, y_i - \frac{(\sum x_i)(\sum y_i)}{n}$$

and

$$S_{xx} = \sum (x_i - \bar{x})^2 = \sum x_i^2 - \frac{(\sum x_i)^2}{n}$$

Example 12.1

Find the least-squares prediction line for the calculus grade data in Table 12.1.

Solution Use the data in Table 12.2 and the data entry method in R to find the following sums of squares:

$$S_{xx} = \sum x_i^2 - \frac{(\sum x_i)^2}{n} = 23{,}634 - \frac{(460)^2}{10} = 2474$$

$$S_{xy} = \sum x_i y_i - \frac{(\sum x_i)(\sum y_i)}{n} = 36{,}854 - \frac{(460)(760)}{10} = 1894$$

$$\bar{y} = \frac{\sum y_i}{n} = \frac{760}{10} = 76 \qquad \bar{x} = \frac{\sum x_i}{n} = \frac{460}{10} = 46$$

Table 12.2 Calculations for the Data in Table 12.1

	y_i	x_i	x_i^2	$x_i y_i$	y_i^2
	65	39	1521	2535	4225
	78	43	1849	3354	6084
	52	21	441	1092	2704
	82	64	4096	5248	6724
	92	57	3249	5244	8464
	89	47	2209	4183	7921
	73	28	784	2044	5329
	98	75	5625	7350	9604
	56	34	1156	1904	3136
	75	52	2704	3900	5625
Sum	760	460	23,634	36,854	59,816

In order to reproduce these results in R, we need to first define all of the variables in the equations. We have done so as follows:

x_i^2	`x.squared`
y_i^2	`y.squared`
$x_i y_i$	`x.y`
$\sum x_i^2$	`sigma1`
$(\sum x_i)^2$	`sigma2`
$\sum x_i y_i$	`sigma3`
$\sum x_i$	`sigma4`
$\sum y_i$	`sigma5`
n	`n`
S_{xx}	`Sxx`
S_{xy}	`Sxy`
$\bar{y}$	`y.bar`
$\bar{x}$	`x.bar`
a	`a`
b	`b`
$\hat{y}$	`y.hat`

```
x.squared = score^2
y.squared = grade^2
sigma.y = sum(y.squared)
sum.x.squared = (sum(score))^2
sigma1 = sum(x.squared)
sigma2 = sum.x.squared
n=length(score)
Sxx = sigma1 - (sigma2/n)

x.y = score*grade
sigma3 = sum(x.y)
sigma4 = sum(score)
sigma5 = sum(grade)
Sxy = sigma3 - (sigma4*sigma5/n)

y.bar = sigma5/n
x.bar = sigma4/n

b = Sxy/Sxx
a = y.bar - b*x.bar

y.hat = a + b*score
y.hat.sum = sum(y.hat)

calculations = data.frame(grade,score,x.squared,x.y,y.squared,y.hat)
calculations[11,] = c(sigma5,sigma4,sigma1,sigma3,sigma.y,y.hat.sum)
calculations
```

	grade	score	x.squared	x.y	y.squared	y.hat
1	65	39	1521	2535	4225	70.64107
2	78	43	1849	3354	6084	73.70331
3	52	21	441	1092	2704	56.86095
4	82	64	4096	5248	6724	89.78011
5	92	57	3249	5244	8464	84.42118
6	89	47	2209	4183	7921	76.76556
7	73	28	784	2044	5329	62.21989
8	98	75	5625	7350	9604	98.20129
9	56	34	1156	1904	3136	66.81326
10	75	52	2704	3900	5625	80.59337
11	760	460	23634	36854	59816	760.00000

```
calculations[11,]
       grade      score      x.squared      x.y       y.squared      y.hat
 11     760        460         23634       36854       59816          760
```

The code `calculations[11,]` gives us the last row of our data frame, which is all of the sums. If we wanted to make these equations generalizable to any bivariate dataset, then we could make a function. First, let's define the template for a function:

```
function.name = function(parameters, data) {
        arguments
        }
```

The `function.name` is anything we would like to call our function so long as it is not already used for a built-in function of R. In the statement `function(parameters, data)`, we enter the parameters we wish to measure and the name of the dataset from which those parameters came. In `{arguments}` we code everything we want to do with these parameters and data. Here is an example of one for sums of squares parameters:

```
sums.of.squares = function(x,y,data) {
   x.squared = x^2
   y.squared = y^2
   sigma.y = sum(y.squared)
   sum.x.squared = (sum(x))^2
   sigma1 = sum(x.squared)
   sigma2 = sum.x.squared
   n=length(x)
   Sxx = sigma1 - (sigma2/n)
   x.y = x*y
   sigma3 = sum(x.y)
   sigma4 = sum(x)
   sigma5 = sum(y)
   Sxy = sigma3 - (sigma4*sigma5/n)
   y.bar = sigma5/n
   x.bar = sigma4/n
   b = Sxy/Sxx
   a = y.bar - b*x.bar
   y.hat = a + b*x
   y.hat.sum = sum(y.hat)
   calculations = data.frame(y,x,x.squared,x.y,y.squared,y.hat)
   calculations[n+1,] = c(sigma5,sigma4,sigma1,sigma3,sigma.y,y.hat.sum)
   calculations
   }
```

Now that we have stored the function, we can use it on our specific bivariate data:

```
sums.of.squares(score,grade,freshmen)
```

and presto!

```
        x       y     x.squared       x.y      y.squared       y.hat
 1      65      39     1521           2535       4225       70.64107
 2      78      43     1849           3354       6084       73.70331
 3      52      21      441           1092       2704       56.86095
 4      82      64     4096           5248       6724       89.78011
 5      92      57     3249           5244       8464       84.42118
 6      89      47     2209           4183       7921       76.76556
 7      73      28      784           2044       5329       62.21989
 8      98      75     5625           7350       9604       98.20129
 9      56      34     1156           1904       3136       66.81326
10      75      52     2704           3900       5625       80.59337
11     760     460    23634          36854      59816      760.00000
```

For a quick-and-easy way to get the best fitting line, we just return our fit linear model from earlier:

```
fit <- lm(grade~score,data=freshmen)
fit
Call:
lm(formula = grade ~ score, data = freshmen)

Coefficients:
(Intercept)        score
    40.7842       0.7656
```

If we want to see all of the output, then:

```
summary(fit)

     Call:        lm(formula = grade ~ score, data = freshmen)
     Residuals:     Min       1Q       Median      3Q         Max
                  -10.813   -5.629     -2.531     6.758      12.234
     Coefficients:   Estimate    Std. Error    t value    Pr(>|t|)
       (Intercept)   40.7842      8.5069        4.794     0.00137  **
            score     0.7656      0.1750        4.375     0.00236  **
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 8.704 on 8 degrees of freedom
Multiple R-squared: 0.7052,     Adjusted R-squared: 0.6684
F-statistic: 19.14 on 1 and 8 DF,  p-value: 0.002365
```

We could also use the built-in R function `simple.lm`. First, if we have not already done so, we should install the **UsingR** package by clicking on "packages" at the top of the graphical user interface (RGui). Click "Install package(s)" and choose a CRAN mirror (it doesn't matter which one). Find **UsingR** and click on it. Then, use the following code:

```
library(UsingR)
simple.lm(score,grade)
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
    40.7842       0.7656
```

12.4 AN ANALYSIS OF VARIANCE FOR LINEAR REGRESSION (p. 509 in text)

The total variation in the response variable y, given by

$$Total\ SS = S_{yy} = \sum (y_i - \bar{y})^2 = \sum y_i^2 - \frac{(\sum y_i)^2}{n}$$

is divided into two portions:

- **SSR** (sum of squares for regression) measures the amount of variation explained by using the regression line with one independent variable x
- **SSE** (sum of squares for error) measures the "residual" variation in the data that is not explained by the independent variable x

so that

$$Total\ SS = SSR + SSE$$

To give a reproduction of Figure 12.5 (p. 509) in R, with first re-apply the `plot` command and overlay an `abline` using our model stored to `fit`. Next, we need to overlay $\bar{y}$, so we store the mean of grade repeated 10 times (10 is length of dataset) using the `rep` command to a name of our choosing. Then, we use the `lines` function to add the line. The command `lty` gives line type, and `col` gives the color. Finally, we re-apply the `segments` function using different colors for the line from $\hat{y}$ to $\bar{y}$ and the line from $\hat{y}$ to y. To label the SSR and SSE portions of the segments, simply use `text()`. The `locator()` command gives you the ability to click anywhere on the graph you wish to place the text. Inputting `1` means you get 1 click. The command `cex` give character expansion.

```
par(font=2,font.axis=2,font.lab=2, lwd=2)
plot(score,grade)
abline(fit,lwd=2)
length(score)
y.bar.line=c(rep(mean(grade),10))
lines(score,y.bar.line,lty=3,col="red")
segments(score,fitted(fit),score,y.bar.line,col="blue",lty=3)
segments(score,fitted(fit),score,grade,col="green",lty=4)
text(locator(1),"SSE",cex=.8,lwd=2)
text(locator(1),"SSR",cex=.8,lwd=2)
text(locator(1),"y mean",cex=.8,lwd=2)
text(locator(1),"y = a + bx",cex=.8,lwd=2)
```

Figure 12.5: Deviations from the fitted line

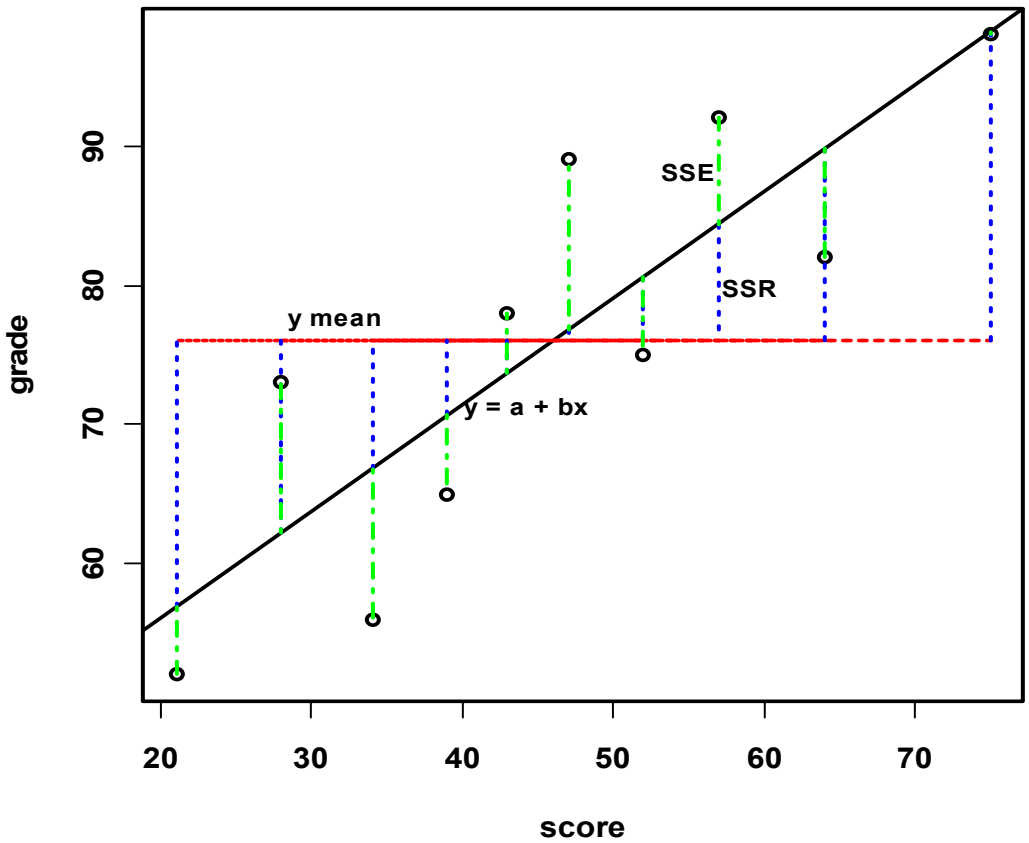


Table 12.3: Analysis of Variance for Linear Regression

Source	*df*	SS	MS
Regression	1	$\frac{(S_{xy})^2}{S_{xx}}$	MSR
Error	n-2	$S_{yy} - \frac{(S_{xy})^2}{S_{xx}}$	MSE
Total	n-1	S_{yy}	

Using R to develop an analysis of variance (ANOVA) for the data in Table 12.1, we can use a variety of methods, some of which give varying but similar output:

```
anova(fit)

Analysis of Variance Table

Response: grade

            Df      Sum Sq      Mean Sq      F value      Pr(>F)

    score    1       1449.97     1449.97      19.141      0.002365 **

  Residuals  8       606.03       75.75

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

aov(fit)

Call:         aov(formula = fit)

Terms:                  score        Residuals

     Sum of Squares    1449.9741     606.0259

     Deg. of Freedom      1             8

Residual standard error: 8.703633

Estimated effects may be unbalanced

summary(aov(fit))

            Df        Sum Sq       Mean Sq       F value        Pr(>F)

 score      1         1449.97      1449.97       19.141       0.002365 **

 Residuals  8         606.03        75.75

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

12.5 TESTING THE USEFULNESS OF THE LINEAR REGRESSION MODEL (p. 514 in text)

In considering linear regression, we may ask two questions:

- Is the independent variable x useful in predicting the response variable y?
- If so, how well does it work?

Test of Hypothesis Concerning the Slope of a Line

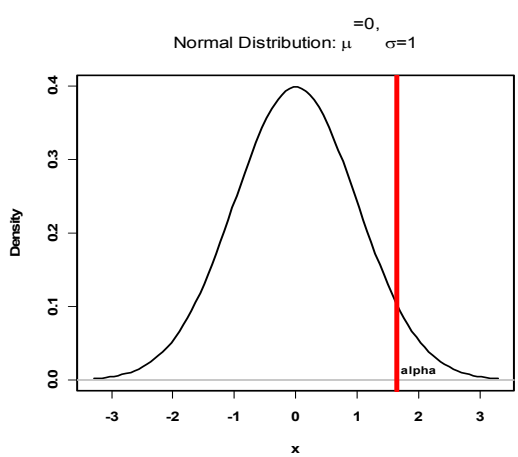
1. Null hypothesis: H_0: $\beta = \beta_0$
2. Alternative hypothesis:

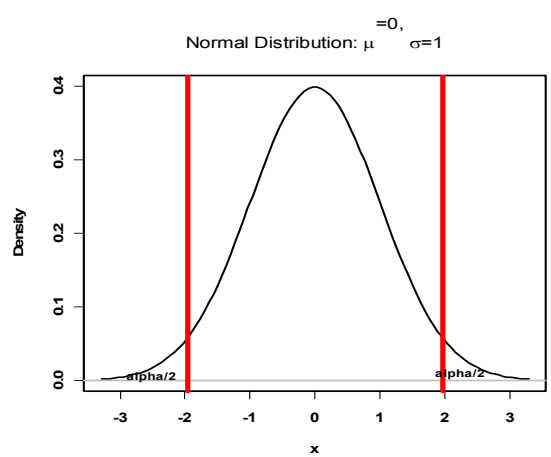
One-Tailed Test	**Two-Tailed Test**
H_a: $\beta > \beta_0$	H_a: $\beta \neq \beta_0$
(or $\beta < \beta_0$)	

3. Test statistic: $t = \frac{b-\beta_0}{\sqrt{\text{MSE}/S_{xx}}}$

   When the assumptions are satisfied, the test statistic will have a Student's t distribution with (n - 2) degrees of freedom.
4. Rejection region: Reject H_0 when

One-Tailed Test	**Two-Tailed Test**
$t > t_\alpha$	$t > t_{\alpha/2}$ or $t < -t_{\alpha/2}$
(or $t < -t_\alpha$ when the alternative hypothesis is H_a: $\beta < \beta_0$)	
or when p-value < α	





To produce the above two graphs:

```
.x <- seq(-3.291, 3.291, length.out=100)
plot(.x, dnorm(.x, mean=0, sd=1), xlab="x", ylab="Density",
      main=expression(paste("Normal Distribution: ",mu,"=0, ",sigma,"=1")),
      type="l")
abline(h=0, col="gray")
remove(.x)
abline(v=1.645, col="red",lwd=5)
text(locator(1),"alpha",cex=.8,lwd=2)

.x <- seq(-3.291, 3.291, length.out=100)
plot(.x, dnorm(.x, mean=0, sd=1), xlab="x", ylab="Density",
      main=expression(paste("Normal Distribution: ",mu,"=0,",sigma,"=1")),
      type="l")
abline(h=0, col="gray")
remove(.x)
abline(v=1.96, col="red",lwd=5)
abline(v=-1.96, col="red",lwd=5)
text(locator(2),"alpha/2",cex=.8,lwd=2)
```

Example 12.2

Determine whether there is a significant linear relationship between the calculus grades and test scores listed in Table 12.1. Test at the 5% level of significance.

Solution The hypotheses to be tested are

H_0: β=0 versus H_a: $\beta \neq 0$

and the observed value of the test statistic is calculated as

$$t = \frac{b-0}{\sqrt{\mathrm{MSE}/S_{\mathrm{xx}}}} = \frac{.7656-0}{\sqrt{75.7532/2474}} = 4.38$$

with (n - 2) = 8 degrees of freedom. R can answer this in a much simpler fashion. Simply re-call all the information using:

```
summary(fit)

  Call:    lm(formula = grade ~ score, data = freshmen)
  Residuals:    Min       1Q        Median       3Q         Max
             -10.813    -5.629     -2.531       6.758      12.234
  Coefficients:    Estimate     Std. Error    t value    Pr(>|t|)
    (Intercept)    40.7842       8.5069        4.794     0.00137   **
        score       0.7656       0.1750        4.375     0.00236   **
      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 8.704 on 8 degrees of freedom
Multiple R-squared: 0.7052,     Adjusted R-squared: 0.6684
F-statistic: 19.14 on 1 and 8 DF,  p-value: 0.002365
```

The slope is under the `Estimate` column in the `score` row. By looking at the far right of the output, we see that there are two asterisks (`**`). Checking the `Signif. codes`, we see that `**` denotes significance at the 0.01 level. We also see `4.378` similar to the book's 4.38 under the `t value` column, and under `Signif. codes` the `Residual standard error` and `degrees of freedom`. To pull out only the information we need about the slope, we simply pull the second row from the `Coefficients` output using:

```
coefficients(summary(fit))[2,]

       Estimate       Std. Error         t value          Pr(>|t|)

     0.765561843     0.174984967      4.375014926     0.002364532
```

Example 12.3

Find a 95% confidence interval estimate of the slope β for the calculus grade data in Table 12.1.

Solution Substituting previously calculated values into

$$b \pm t_{.025}\sqrt{\frac{\text{MSE}}{S_{xx}}}$$

you have

$$.766 \pm 2.306\sqrt{\frac{75.7532}{2474}}$$

$$.766 \pm .404$$

The resulting 95% confidence interval is .362 to 1.170. Since the interval does not contain 0, you can conclude that the true value of β is not 0, and you can reject the null hypothesis H_0: β=0 in favour of H_a: $\beta \neq 0$, a conclusion that agrees with the findings in Example 12.2. Since the summary from the model output gives us all the parameters we need to produce a confidence interval, we can use R to retrieve them and produce these same results from the book:

```
beta=coefficients(summary(fit))[2,1]
alpha=.05
n=length(fit)
t.star=qt(1-alpha/2,n-2)
SE=coefficients(summary(fit))[2,2]
conf.int=c(beta-t.star*SE,beta+t.star*SE)
[1] 0.3756710   1.1554526
```

Or we could produce a function:

```
conf.int = function (model,alpha) {
    beta=coefficients(summary(model))[2,1]
    n=length(model)
    t.star=qt(1-alpha/2,n-2)
    SE=coefficients(summary(model))[2,2]
    c(beta-t.star*SE,beta+t.star*SE)
   }
```

Simply input the name of our model and alpha corresponding to our desired confidence level:

```
conf.int(fit,.05)
```

and presto!

```
[1] 0.3756710   1.1554526
```

Coefficient of Determination

Definition The coefficient of determination r^2 can be interpreted as the percent reduction in the total variation in the experiment obtained by using the regression line $\hat{y} = a + bx$, instead of ignoring x and using the sample mean $\bar{y}$ to predict the response variable y. For the calculus grade data, a reduction of $r^2 = .705$ or 70.5% is substantial. This value can also be found in R under `summary(fit)` at the bottom of the output labelled `Multiple R-squared`.

12.6 DIAGNOSTIC TOOLS FOR CHECKING THE REGRESSION ASSUMPTIONS (p. 522 in text)

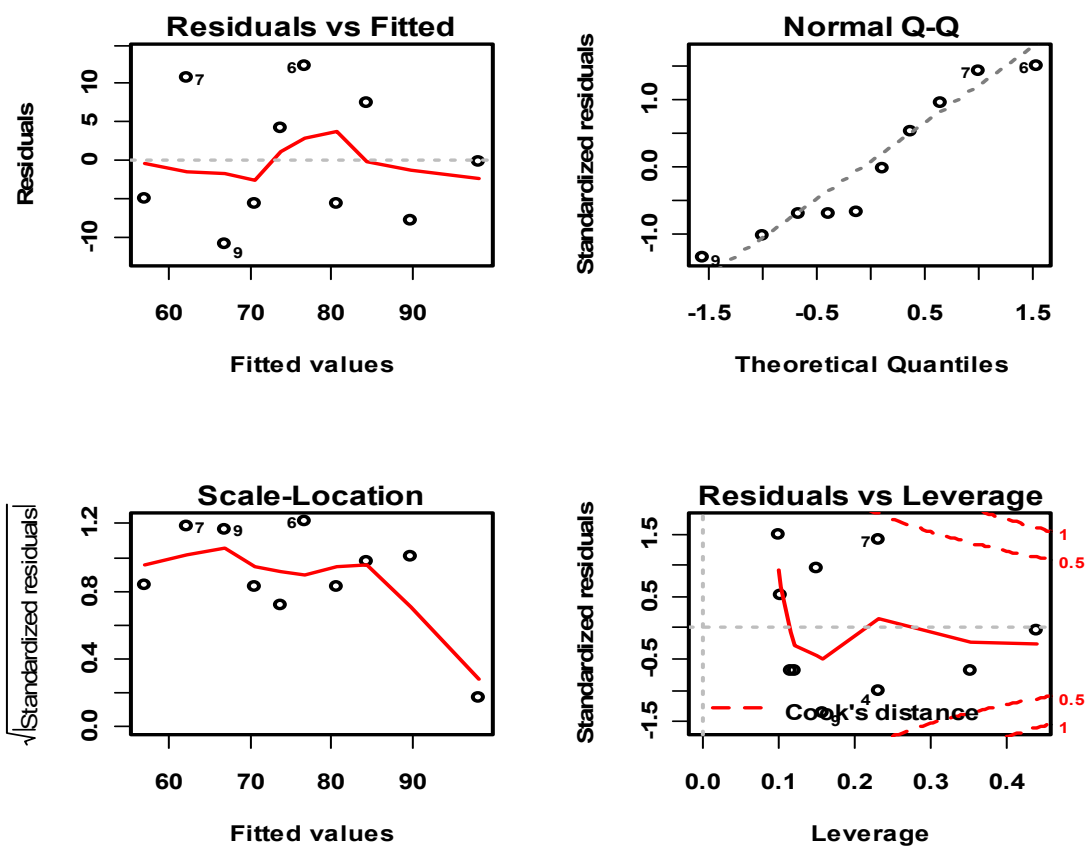
Even though you have determined – using the t-test for the slope (or the ANOVA F-test) and the value of r^2 – that x is useful in predicting the value of y, the results of a regression analysis are valid only when the data satisfy the necessary regression assumptions.

Regression Assumptions

- The relationship between y and x must be linear, given by the model
$$y = \alpha + \beta x + \varepsilon$$
- The values of the random error term ε (1) are independent, (2) have a mean of 0 and a common variance σ^2, independent of x, and (3) are normally distributed.

In R, when we use the plot command on our fitted model it returns us four plots. Therefore, we need to set our graphing window parameters to `c(2,2)` so it will plot two graphs by two graphs in one window. We set it back using `c(1,1)` as follows:

```
par(mfrow=c(2,2))
plot(fit)
par(mfrow=c(1,1))
```



Interpretation of Diagnostic Plots (obtained from *Modelling in R: Lesson One* by Sudha Purohit (c) Sudha Purohit and statistics.com p. 9)

The upper left figure of the `plot(lm())` function returns us the residuals plotted against the fitted values for the model. If there is no violation of our model assumptions, then these points will be approximately randomly distributed with a constant variance above and below the fitted line. If there appears to be a pattern in this figure, then we may need to fit a different model (i.e. quadratic, cubic, log transformation).

The lower left figure returns the scale location plot. Here the square root of the estimated standard errors of the raw residuals is plotted against the fitted values. The square root reduces the skewness in the distribution, so it is easier to validate the constant variance assumption.

The upper right figure is a quantile-quantile (q-q) plot of the residuals against the normal distribution. If a straight line is revealed, the normality is not violated. Otherwise, there is a violation.

The lower right figure shows the standardized residuals against the leverage of the points. It essentially determines whether or not any particular point has a significant influence on the overall model. This influence is measured by the Cook's Distance value, which tests for outliers. If this value is very large for a point, it is an indication that this particular datum exerts a large amount of influence on the overall model. Hence, if one or more points exhibit a very large Cook's Distance value, we can conclude that the point(s) is outlier. Therefore, it may be wise to re-compute the model without the point(s).

12.7 ESTIMATION AND PREDICTION USING THE FITTED LINE (p. 527 in text)

Now that you have

- tested the fitted regression line, $\bar{y} = a + bx$, to make sure that it is useful for prediction and
- used the diagnostic tools to make sure that none of the regression assumptions have been violated

you are ready to use the line for one of its two purposes:

- Estimating the average value of y for a given value of x
- Predicting a particular value of y for a given value of x

(1 – α) 100% Confidence and Prediction Intervals

- For estimating the average value of y when $x = x_0$:

$$\hat{y} \pm t_{\alpha/2}\sqrt{\text{MSE}\left[\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}}\right]}$$

- For predicting a particular value of y when $x = x_0$:

$$\hat{y} \pm t_{\alpha/2}\sqrt{\text{MSE}\left[1+\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}}\right]}$$

where $t_{\alpha/2}$ is the value of t with (n - 2) degrees of freedom and area $\alpha/2$ to its right.

Example 12.4

Use the information in Example 12.1 to estimate the average calculus grade for students whose achievement score is 50, with a 95% confidence interval.

Solution The point estimate of $E(y|x_0 = 50)$, the average calculus grade for students whose achievement score is 50, is

$$\hat{y} = 40.78424 + .76556(50) = 79.06$$

The standard error of $\hat{y}$ is

$$\sqrt{\text{MSE}\left[\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}}\right]} = \sqrt{\text{MSE}\left[\frac{1}{n}+\frac{(x_0-\bar{x})^2}{S_{xx}}\right]} = 2.840$$

and the 95% confidence interval is

79.06 ± 2.306(2.840)

79.06 ± 6.55

Our results indicate that the average calculus grade for students who score 50 on the achievement test will lie between 72.51 and 85.61. To obtain this result in R, we simply use the `predict` function. Input the model (in this case `fit`), then format with `data.frame` and enter the value for the x variable, specify the confidence level, and ask for what type of interval you are seeking.

```
predict(fit,data.frame(score=50),level=.95, interval="confidence")
                 fit            lwr             upr
      1        79.06225       72.51334        85.61115
```

Example 12.5

A student took the achievement test and scored 50 but has not yet taken the calculus test. Using the information in Example 12.1, predict the calculus grade for this student with a 95% prediction interval.

Solution The predicted value of y is $\bar{y} = 79.06$, as in Example 12.4. However, the error in prediction is measured by $\text{SE}(y - \hat{y})$, and the 95% prediction interval is

$$79.06 \pm 2.306\sqrt{75.7532\left[1 + \frac{1}{10} + \frac{(50-46)^2}{2474}\right]}$$
$$79.06 \pm 2.306(9.155)$$
$$79.06 \pm 21.11$$

or from 57.95 to 100.17. The prediction interval is *wider* than the confidence interval in Example 12.4 because of the extra variability in predicting the actual value of the response y. To produce this in R, we use the same function `predict`, only specifying `"prediction"` as our interval type.

```
predict(fit,data.frame(score=50),level=.95, interval="prediction")
             fit         lwr             upr
     1     79.06225    57.95022        100.1743
```

One particular point on the line of means is often of interest to experimenters, the ***y*-intercept *α*** – the average value of y when $x_0 = 0$.

Example 12.6

Prior to fitting a line to the calculus grade-achievement score data, you may have thought that a score of 0 on the achievement test would predict a grade of 0 on the calculus test. This implies that we should fit a model with α equal to 0. Do the data support the hypothesis of a 0 intercept?

Solution You can answer this question by constructing a 95% confidence interval for the y-intercept α, which is the average value of y when $x = 0$. The estimate of α is

$$\hat{y} = 40.784 + .76556(0) = 40.784 = \alpha$$

and the 95% confidence interval is

$$\hat{y} \pm t_{\alpha/2}\sqrt{\text{MSE}\left[1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right]}$$
$$40.784 \pm 2.306\sqrt{75.7532\left[\frac{1}{10} + \frac{(0-46)^2}{2474}\right]}$$
$$40.784 \pm 19.617$$

Or from 21.167 to 60.401, an interval that does not contain the value $\alpha = 0$. Hence, it is unlikely that the y-intercept is 0. To reproduce this in R, simply use `predict` again.

```
predict(fit,data.frame(score=0),level=.95, interval="confidence")

              fit              lwr              upr

       1    40.78416         21.16730         60.40101
```

The test for the 0 intercept given in Figure 12.14 of the text is again found in the summary of the fitted model in the line labeled `(Intercept)`. The coefficient given as 40.7842 is a, with standard error given in the column labeled `Std. Error` as 8.5069, which agrees with the value calculated in Example 12.6. The `t value` 4.794 is found by dividing a by its standard error with p-value = .001.

```
summary(fit)

  Call:        lm(formula = grade ~ score, data = freshmen)
  Residuals:    Min      1Q       Median       3Q         Max
               -10.813  -5.629    -2.531      6.758      12.234
  Coefficients:   Estimate      Std. Error    t value     Pr(>|t|)
    (Intercept)   40.7842        8.5069        4.794      0.00137  **
         score     0.7656        0.1750        4.375      0.00236  **
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 8.704 on 8 degrees of freedom
Multiple R-squared: 0.7052,     Adjusted R-squared: 0.6684
F-statistic: 19.14 on 1 and 8 DF,  p-value: 0.002365
```

To return the standard error of the fit for the new observation, we simply call for it using the `predict` function again.

```
predict(fit,data.frame(score=50),se.fit=TRUE)

          $fit        $se.fit        $df        $residual.scale

        79.06225     2.839936        8            8.703633
```

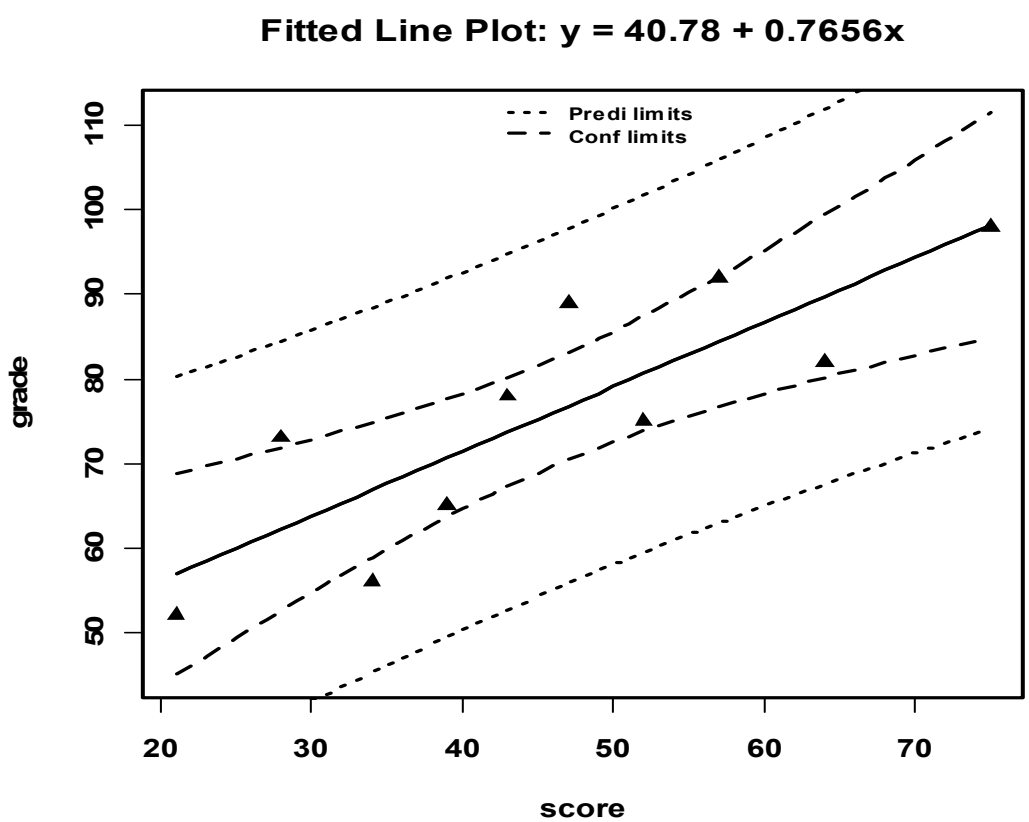
It is found under `$se.fit` as `2.839936`.

To reproduce Figure 12.16 from the text, we use the `matplot` function.

```
range(freshmen$score)

 [1] 21 75
```

```
predict.frame <- data.frame(score=21:75)

b1<-predict(fit,interval="prediction", newdata=predict.frame)

b2<-predict(fit,interval="confidence", newdata=predict.frame)

par(font=2,font.axis=2,font.lab=2, lwd=2)

matplot(score,grade,"p",pch=17,ylim=range(grade,b2),

        main="Fitted Line Plot: y = 40.78 + 0.7656x")

pred.score <- predict.frame$score

matlines(pred.score,b2,type="l",lty=c(1,2,2), col="black",lwd=2)

matlines(pred.score,b1,type="l",lty=c(1,3,3), col="black",lwd=2)

legend("top",legend=c("Predi limits","Conf
limits"),lty=c(3,2),cex=0.7,bty="n")
```

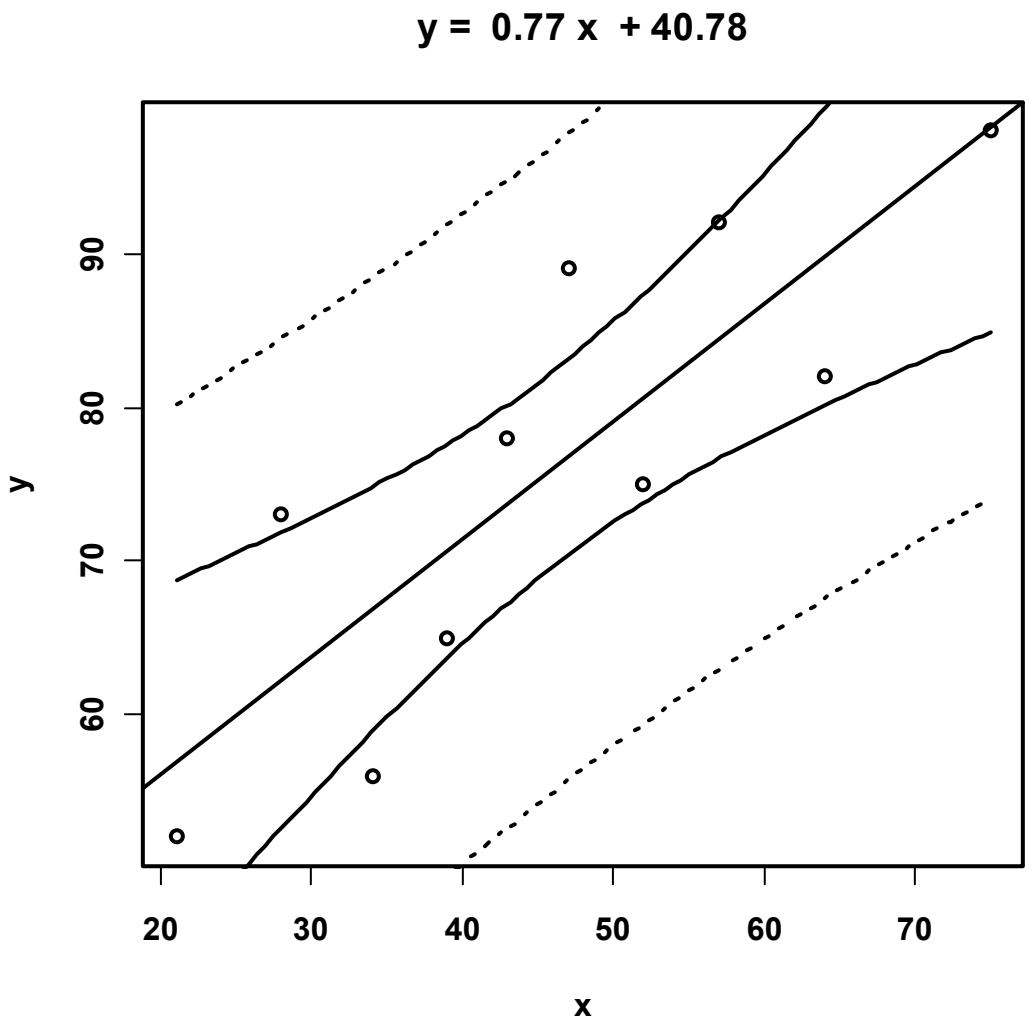
Figure 12.16: Confidence and prediction intervals for the data in Table 12.1 (using `matplot` and `matlines`)



This can also be done using `simple.lm` after loading the package **UsingR**.

```
library(UsingR)

simple.lm(score,grade,show.ci=TRUE,conf.level=0.95)
```

Figure 12.16: Confidence and prediction intervals for the data in Table 12.1 (using `simple.lm`)



12.8 CORRELATION ANALYSIS (p. 533 in text)

The correlation coefficient, r – is formally called the **Pearson product moment sample coefficient of correlation**.

Pearson Product Moment Coefficient of Correlation

$r = \frac{s_{xy}}{s_x s_y} = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}}$ for $-1 \le r \le 1$

Example 12.7

The heights and weights of n = 10 offensive backfield football players are randomly selected from a county's football all-stars. Calculate the correlation coefficient for the heights (in inches) and weights (in pounds) given in Table 12.4.

Table 12.4: Heights and Weights of n = 10 Backfield All-Stars

Player	Height, x	Weight, y
1	73	185
2	71	175
3	75	200
4	72	210
5	72	190
6	75	195
7	67	150
8	69	170
9	71	180
10	69	175

Solution You should use the appropriate data entry method of R to verify the calculations for the sums of squares and cross-products

$$S_{xy} = 328 \quad S_{xx} = 60.4 \quad S_{yy} = 2610$$

then

$$r = \frac{328}{\sqrt{(60.4)(2610)}} = .8261$$

or r = .83. This value of r is fairly close to 1, the largest possible value of r, which indicates a fairly strong positive linear relationship between height and weight. To reproduce this is R, we first need to input the data:

```
height=c(73,71,75,72,72,75,67,69,71,69)
weight=c(185,175,200,210,190,195,150,170,180,175)
football=data.frame(height,weight)
```

Then reproduce Table 12.4:

```
football
    height weight
1       73    185
2       71    175
3       75    200
4       72    210
5       72    190
6       75    195
7       67    150
8       69    170
9       71    180
10      69    175
```

Then, we can use a similar function to `sums.of.squares` that we created for Sums of Squares earlier. However, now we only need S_{xy}, S_{xx}, S_{yy}, and r. We change the name of our function to `corr.anal`.

x_i^2	`x.squared`
y_i^2	`y.squared`
$x_i y_i$	`x.y`
$\sum x_i^2$	`sigma.x.square`
$\sum y_i^2$	`sigma.y.square`
$(\sum x_i)^2$	`sum.x.squared`
$(\sum y_i)^2$	`sum.y.squared`
$\sum x_i y_i$	`sigma.xy`
$\sum x_i$	`sigma.x`
$\sum y_i$	`sigma.y`
n	`n`
S_{xx}	`Sxx`
S_{xy}	`Sxy`

```
corr.anal = function(x,y,data) {
   x.squared = x^2
   y.squared = y^2
   sigma.x.square = sum(x.squared)
   sigma.y.square = sum(y.squared)
   sum.x.squared = (sum(x))^2
   sum.y.squared = (sum(y))^2
   n=length(x)
   Sxx = sigma.x.square - (sum.x.squared/n)
   Syy = sigma.y.square - (sum.y.squared/n)

   x.y = x*y
   sigma.xy = sum(x.y)
   sigma.x = sum(x)
   sigma.y = sum(y)
   Sxy = sigma.xy - (sigma.x*sigma.y/n)

   r = Sxy / sqrt(Sxx*Syy)
   measures = data.frame(Sxy,Sxx,Syy,r)
   measures
   }

corr.anal(height,weight,football)
          Sxy        Sxx          Syy          r
     1    328        60.4         2610       0.8261048
```

Or, we could just use the built-in `cor` function of R:

```
cor(height,weight)
          [1] 0.8261048
cor(football)
                       height          weight
       height        1.0000000       0.8261048
       weight        0.8261048       1.0000000
cor(height,weight,method="pearson")
          [1] 0.8261048
cor(football,method="pearson")
                      height          weight
       height       1.0000000       0.8261048
       weight       0.8261048       1.0000000
```

By default, `method="pearson"`. There is a direct relationship between the correlation coefficient r and the slope of the regression line b. Since the numerator of both quantities is S_{xy}, both r and b have the same sign. Therefore, the correlation coefficient has these general properties:

- When $r = 0$, the slope $b = 0$, and there is no linear relationship between x and y.
- When r is positive, so is b, and there is a positive linear relationship between x and y.
- When r is negative, so is b, and there is a negative linear relationship between x and y.
- If there is no random variation and all the points fall on the regression line, then SSE = 0 and $r^2 = 1$.
- If the points are randomly scattered and there is no variation explained by regression, then SSR = 0 and $r^2 = 0$.

Test of Hypothesis Concerning the Correlation Coefficient ρ

1. Null hypothesis: H_0: $\rho = 0$
2. Alternative hypothesis:

One-Tailed Test	**Two-Tailed Test**
H_a: $\rho > 0$ (or $\rho < 0$)	H_a: $\rho \neq 0$

3. Test statistic: $t = r\sqrt{\frac{n-2}{1-r^2}}$

   When the assumptions are satisfied, the test statistic will have a Student's t distribution with (n - 2) degrees of freedom.
4. Rejection region: Reject H_0 when

One-Tailed Test	**Two-Tailed Test**
$t > t_\alpha$ (or $t < -t_\alpha$ when the alternative hypothesis is H_a: $\beta < \beta_0$) or when p-value < α	$t > t_{\alpha/2}$ or $t < -t_{\alpha/2}$

Example 12.8

Refer to the height and weight data in Example 12.7. The correlation of height and weight was calculated to be $r = .8261$. Is this correlation significantly different from 0?

Solution To test the hypotheses

$H_0: \rho = 0$ versus $H_a: \rho \neq 0$

the value of the test statistic is

$$t = r\sqrt{\frac{n-2}{1-r^2}} = .8261\sqrt{\frac{10-2}{1-(.8261)^2}} = 4.15$$

which for $n = 10$ has a t distribution with 8 degrees of freedom. Since this value is greater than $t_{.005} = 3.355$, the two-tailed p-value is less than $2(.005) = .01$, and the correlation is declared significant at the 1% level ($P < .01$). The value $r^2 = .8261^2 = .6824$ means that about 68% of the variation in one of the variables is explained by the other. To produce this data in R, we simply use the `cor.test` function. By default, `method="pearson"` and `conf.level=0.95`.

```
cor.test(height,weight)

        Pearson's product-moment correlation

data:  height and weight
t = 4.1464, df = 8, p-value = 0.003225
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4094472 0.9576313
sample estimates:
      cor
0.8261048
```

CHAPTER 13: MULTIPLE REGRESSION ANALYSIS

13.2 THE MULTIPLE REGRESSION MODEL (p. 552 in text)

The **general linear model** for a multiple regression analysis describes a particular response y using the model given next.

General Linear Model and Assumptions

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon$$

where

- y is the **response variable** that you want to predict.
- $\beta_0, \beta_1, \beta_2, \ldots, \beta_k$ are unknown constants.
- $x_1, x_2, \ldots, x_k$ are independent **predictor variables** that are measured without error.
- ε is the random error, which allows each response to deviate from the average value of y by the amount ε. You must assume that the values of ε (1) are independent; (2) have a common variance σ^2 for any set $x_1, x_2, \ldots, x_k$; and (3) are normally distributed.

Example 13.1

Suppose you want to relate a random variable y to two independent variables x_1 and x_2. The multiple regression model is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$$

with the mean value of y given as

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

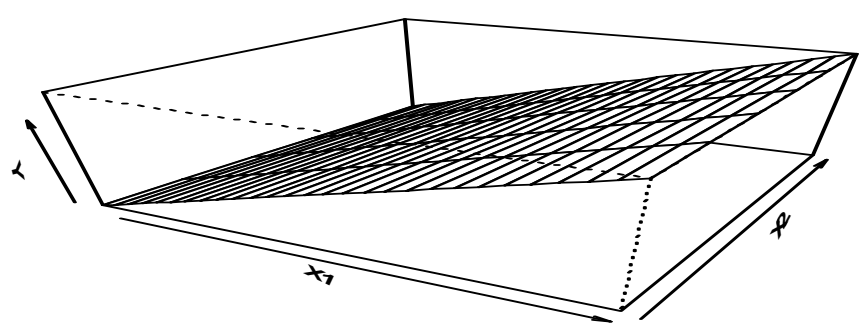
This equation is a three-dimensional extension of the **line of means** from Chapter 12 and traces a **plane** in three-dimensional space (see Figure 13.1 reproduced in R using the perspective plot function `persp` with x_1 and x_2 from Example 13.2).

```
price=c(169.0,218.5,216.5,225.0,229.9,235.0,239.9,247.9,260.0,269.9,234.9,
        55.0,269.9,294.5,309.9)
area=c(6,10,10,11,13,13,13,17,19,18,13,18,17,20,21)
floors=c(1,1,1,1,1,2,1,2,2,1,1,1,2,2,2)
bed=c(2,2,3,3,3,3,3,3,3,3,4,4,4,4,4)
bath=c(1,2,2,2,1.7,2.5,2,2.5,2,2,2,2,3,3,3)
condos=data.frame(price,area,floors,bed,bath)
condos
```

```
        price      area       floors     bed     bath
1       169.0        6           1        2      1.0
2       218.5       10           1        2      2.0
3       216.5       10           1        3      2.0
4       225.0       11           1        3      2.0
5       229.9       13           1        3      1.7
6       235.0       13           2        3      2.5
7       239.9       13           1        3      2.0
8       247.9       17           2        3      2.5
9       260.0       19           2        3      2.0
10      269.9       18           1        3      2.0
11      234.9       13           1        4      2.0
12      255.0       18           1        4      2.0
13      269.9       17           2        4      3.0
14      294.5       20           2        4      3.0
15      309.9       21           2        4      3.0
```

```
lm.condos <- lm(price ~ area + floors, data=condos)
range(area)
X1<-seq(6,61,2)
range(floors)
X2<-seq(1,2,by=.2)
Y<-outer(X1,X2,function(X1,X2)predict.lm(lm.condos,
         newdata=data.frame(area=X1,floors=X2),type="response"))
par(font=2,font.axis=2,font.lab=2, lwd=2)
persp(X1,X2,Y,theta=30,phi=30,expand=0.5)
```

Figure 13.1: Plane of means for Example 13.1



13.3 A MULTIPLE REGRESSION ANALYSIS (p. 553 in text)

A multiple regression analysis involves estimation, testing, and diagnostic procedures designed to fit the multiple regression model

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

to a set of data.

Example 13.2

How do real estate agents decide on the asking price for a newly listed condominium? A computer database in a small community contains the listed selling price y (in thousands of dollars), the amount of living area x_1 (in hundreds of square feet), and the numbers of floors x_2, bedrooms x_3, and bathrooms x_4, for n = 15 randomly selected condos currently on the market. The data are shown in Table 13.1.

Table 13.1: Data on 15 Condominiums

Observation	List Price, y	Living Area, x_1	Floors, x_2	Bedrooms, x_3	Baths, x_4
1	169.0	6	1	2	1
2	218.5	10	1	2	2
3	216.5	10	1	3	2
4	225.0	11	1	3	2
5	229.9	13	1	3	1.7
6	235.0	13	2	3	2.5
7	239.9	13	1	3	2
8	247.9	17	2	3	2.5
9	260.0	19	2	3	2
10	269.9	18	1	3	2
11	234.9	13	1	4	2
12	255.0	18	1	4	2
13	269.9	17	2	4	3
14	294.5	20	2	4	3
15	309.9	21	2	4	3

To reproduce in R, we first enter our data.

```
price=c(169.0,218.5,216.5,225.0,229.9,235.0,239.9,247.9,260.0,269.9,234.9,
        55.0,269.9,294.5,309.9)
area=c(6,10,10,11,13,13,13,17,19,18,13,18,17,20,21)
floors=c(1,1,1,1,1,2,1,2,2,1,1,1,2,2,2)
bed=c(2,2,3,3,3,3,3,3,3,3,4,4,4,4,4)
bath=c(1,2,2,2,1.7,2.5,2,2.5,2,2,2,2,3,3,3)
condos=data.frame(price,area,floors,bed,bath)
condos
```

```
        price        area        floors        bed        bath
  1     169.0         6            1            2         1.0
  2     218.5        10            1            2         2.0
  3     216.5        10            1            3         2.0
  4     225.0        11            1            3         2.0
  5     229.9        13            1            3         1.7
  6     235.0        13            2            3         2.5
  7     239.9        13            1            3         2.0
  8     247.9        17            2            3         2.5
  9     260.0        19            2            3         2.0
 10     269.9        18            1            3         2.0
 11     234.9        13            1            4         2.0
 12     255.0        18            1            4         2.0
 13     269.9        17            2            4         3.0
 14     294.5        20            2            4         3.0
 15     309.9        21            2            4         3.0
```

To obtain the regression analysis, we first fit a linear model using the `lm` function just as we did in simple linear regression but with the additional covariates. We then use the `summary` function to obtain the coefficients, standard errors, *t*-values, *p*-values, *R*-squared, and *F*-test.

```
lm.condos.full<-lm(price~area+floors+bed+bath,data=condos)

summary(lm.condos.full)
```

Figure 13.2: R Regression Analysis Printout for Example 13.2

```
Call:    lm(formula = price ~ area + floors + bed + bath, data = condos)

Residuals:    Min        1Q         Median        3Q         Max

            -12.700    -1.616        0.984       2.510      11.759

Coefficients:    Estimate      Std. Error      t value     Pr(>|t|)

   (Intercept)   118.7633        9.2074        12.899      1.48e-07  ***

        area       6.2698        0.7252         8.645      5.93e-06  ***

       floors    -16.2033        6.2121        -2.608      0.02611    *

         bed      -2.6730        4.4939        -0.595      0.56519

        bath      30.2705        6.8487         4.420      0.00129   **

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

                Residual standard error: 6.849 on 10 degrees of freedom

                Multiple R-squared: 0.9714,     Adjusted R-squared: 0.9599

                F-statistic:  84.8 on 4 and 10 DF,  p-value: 1.128e-07
```

The Analysis of Variance for Multiple Regression

We call for the ANOVA for multiple regression just as we did with simple regression using the `anova` function. This table gives us the sum of squares, mean squares, *F*-value, and *p*-value for each covariate and for the residuals.

```
anova(lm.condos.full)
```

Figure 13.3: R ANOVA Printout for Example 13.2

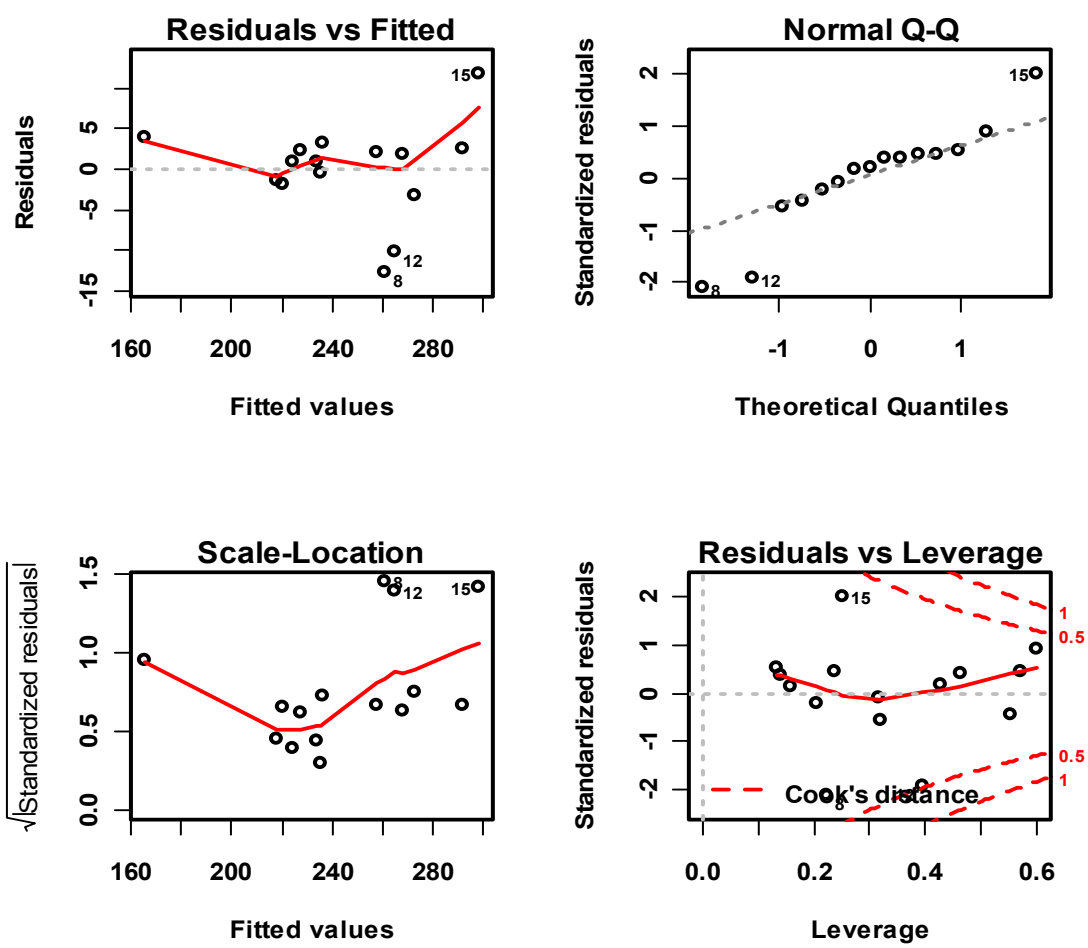
```
   Analysis of Variance Table         Response: price

              Df      Sum Sq       Mean Sq        F value      Pr(>F)

     area      1      14829.3       14829.3       316.1025     6.76e-09 ***

   floors      1          0.9           0.9         0.0184     0.894652

      bed      1        166.4         166.4         3.5472     0.089023 .

     bath      1        916.5         916.5        19.5356     0.001294 **

Residuals     10        469.1          46.9

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

Checking the Regression Assumptions

Before using the regression model for its main purpose – estimation and prediction of *y* – you should look at computer-generated **residual plots** to make sure that all the regression assumptions are valid. These are shown in Figure 13.5 for the real estate data. There appear to be three observations that do not fit the general pattern. You can see them as outliers in the *Residuals vs. Fitted* and *Scale-Location* plots. These three observations should probably be investigated; however, they do not provide strong evidence that the assumptions are violated.

```
par(font=2,font.axis=2,font.lab=2, lwd=2)
par(mfrow=c(2,2))
plot(lm.condos.full)
par(mfrow=c(1,1))
```

Figure 13.5: R diagnostic plots



Using the Regression Model for Estimation and Prediction

Finally, once you have determined that the model is effective in describing the relationship between y and the predictor variables $x_1, x_2, \ldots, x_k$, the model can be used for these purposes:

- Estimating the average value of y – $E(y)$ – for given values of $x_1, x_2, \ldots, x_k$
- Predicting a particular value of y for given values of $x_1, x_2, \ldots, x_k$

Let's see how well our prediction works for the real estate data, using another house from the computer database – a house with 1000 square feet of living area, one floor, three bedrooms, and two baths, which was listed at $221,500. The printout in Figure 13.6 shows the confidence and prediction intervals for these values.

```
predict(lm.condos.full,data.frame(area=10,floors=1,bed=3,bath=2),
        level=.95,interval="confidence",se.fit=TRUE)
```

Figure 13.6: Confidence interval for Example 13.2

```
$fit
           fit              lwr               upr
1        217.7797         210.8587          224.7006
$se.fit
[1] 3.10617
$df
[1] 10
$residual.scale
[1] 6.849303
```

```
predict(lm.condos.full,data.frame(area=10,floors=1,bed=3,bath=2),
        level=.95,interval="prediction",se.fit=TRUE)
```

Figure 13.6: Prediction interval for Example 13.2

```
$fit
            fit                lwr               upr
1         217.7797           201.0224          234.5369

$se.fit
[1] 3.10617

$df
[1] 10

$residual.scale
[1] 6.849303
```

13.4 A POLYNOMIAL REGRESSION MODEL (p. 559 in text)

In Section 13.3, we explained in detail the various portions of the multiple regression printout. When you perform a multiple regression analysis, you should use a step-by-step approach:

1. Obtain the fitted prediction model.
2. Use the analysis of variance F-test and R^2 to determine how well the model fits the data.
3. Check the t-tests for the partial regression coefficients to see which ones are contributing significant information in the presence of the others.
4. If you choose to compare several different models, use R^2(adj) to compare their effectiveness.
5. Use computer-generated residual plots to check for violation of the regression assumptions.

Example 13.3

In a study of variables that affect productivity in the retail grocery trade, W.S. Good uses value added per work-hour to measure the productivity of retail grocery outlets. He defines "value added" as "the surplus [money generated by the business] available to pay for labor, furniture and fixtures, and equipment." Data consistent with the relationship between value added per work-hour y and the size x of a grocery outlet described in Good's article are shown in Table 13.2 for 10 fictitious grocery outlets. Choose a model to relate y to x.

Table 13.2: Data on Store Size and Value Added

Store	Value Added per Work-Hour, y	Size of Store (thousand square feet), x
1	$4.08	21.0
2	3.40	12.0
3	3.51	25.2
4	3.09	10.4
5	2.92	30.9
6	1.94	6.8
7	4.11	19.6
8	3.16	14.5
9	3.75	25.0
10	3.60	19.1

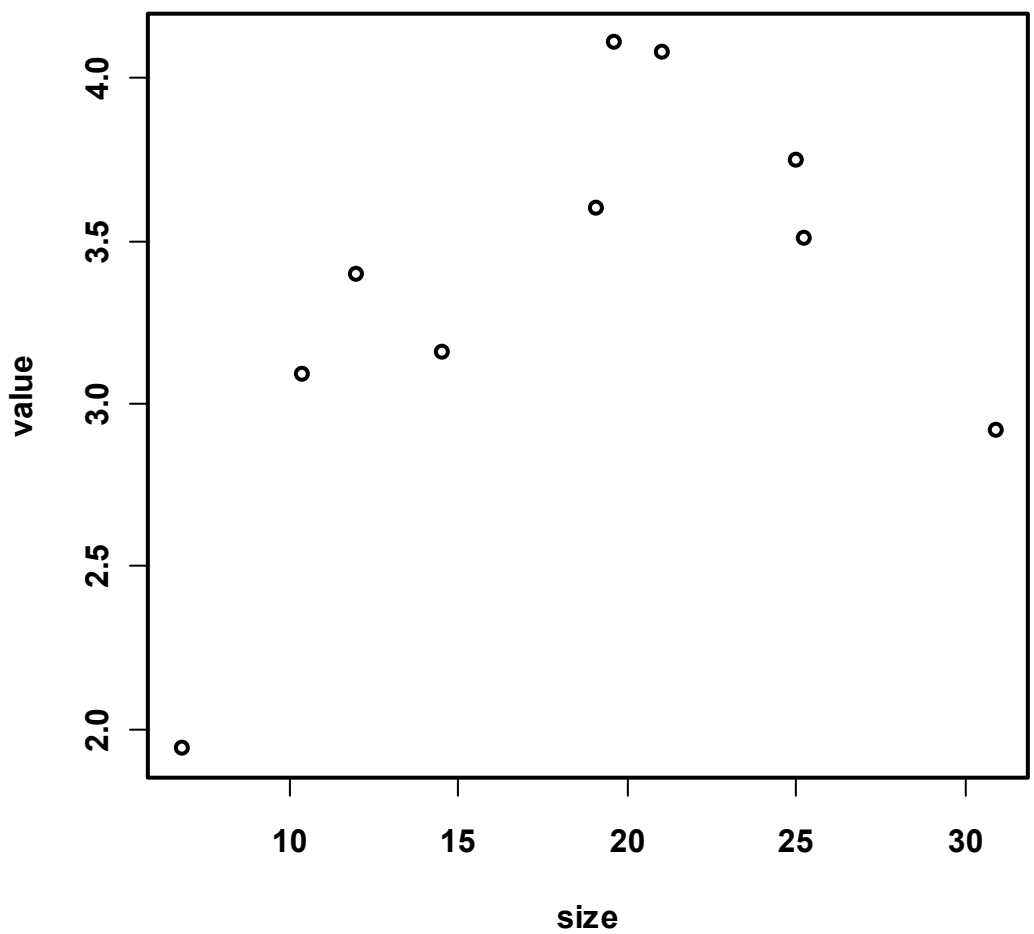
Solution You can investigate the relationship between y and x by looking at the plot of the data points in Figure 13.7. The graph suggests that productivity, y, increases as the size of the grocery outlet, x, increases until an optimal size is reached. Above that size, productivity tends to decrease. The relationship appears to be *curvilinear*, and a quadratic model,

$$E(y) = \beta_0 + \beta_1 x + \beta_2 x^2$$

may be appropriate. Remember that, in choosing to use this model, we are not saying that the true relationship is quadratic, but only that it may provide more accurate estimations and predictions than, say, a linear model.

```
value=c(4.08,3.40,3.51,3.09,2.92,1.94,4.11,3.16,3.75,3.60)

size=c(21.0,12.0,25.2,10.4,30.9,6.8,19.6,14.5,25.0,19.1)

grocery=data.frame(size,value)

grocery

           size       value

    1      21.0       4.08

    2      12.0       3.40

    3      25.2       3.51

    4      10.4       3.09

    5      30.9       2.92

    6       6.8       1.94

    7      19.6       4.11

    8      14.5       3.16

    9      25.0       3.75

   10      19.1       3.60

par(font=2,font.axis=2,font.lab=2, lwd=2)

plot(size,value)
```

Figure 13.7: Plot of store size *x* and value added *y* for Example 13.3



Example 13.4

Refer to the data on grocery outlet productivity and outlet size in Example 13.3. R is used to fit a quadratic model to the data and to graph the quadratic prediction curve, along with the plotted points. Discuss the adequacy of the fitted model.

Solution From the printout in Figure 13.8, you can see that the regression equation is

$$\hat{y} = -.159 + .392x - .00949x^2$$

The graph of this quadratic equation together with the data points is shown in Figure 13.9.

```
lm.grocery <- lm(value ~ size + I(size^2))
summary(lm.grocery)
```

Figure 13.8: R Regression Analysis printout for Example 13.4

```
Call:      lm(formula = value ~ size + I(size^2))
Residuals:     Min        1Q       Median       3Q         Max
            -0.36736   -0.16497    0.03989    0.19918     0.23504
Coefficients:   Estimate      Std. Error     t value    Pr(>|t|)
  (Intercept)  -0.159356       0.500580      -0.318     0.759512
       size     0.391931       0.058006       6.757     0.000263  ***
    I(size^2)  -0.009495       0.001535      -6.188     0.000451  ***
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
         Residual standard error: 0.2503 on 7 degrees of freedom
         Multiple R-squared: 0.8794,     Adjusted R-squared: 0.845
         F-statistic: 25.53 on 2 and 7 DF,  p-value: 0.0006085
```

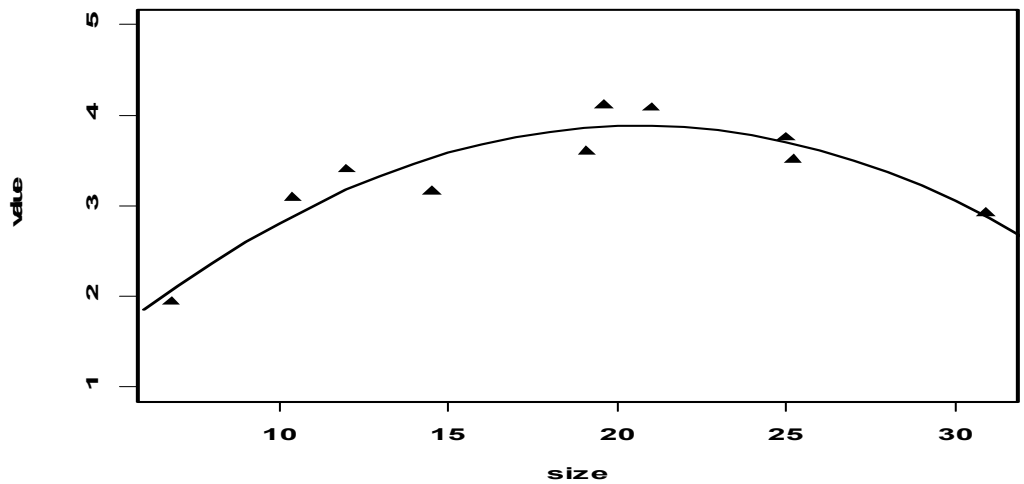
```
anova(lm.grocery)
```

Figure 13.8: R ANOVA printout for Example 13.4

```
Analysis of Variance Table

Response: value

              Df       Sum Sq       Mean Sq       F value     Pr(>F)

  size        1        0.80032      0.80032        12.774    0.0090466 **

I(size^2)     1        2.39858      2.39858        38.286    0.0004507 ***

Residuals     7        0.43855      0.06265

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

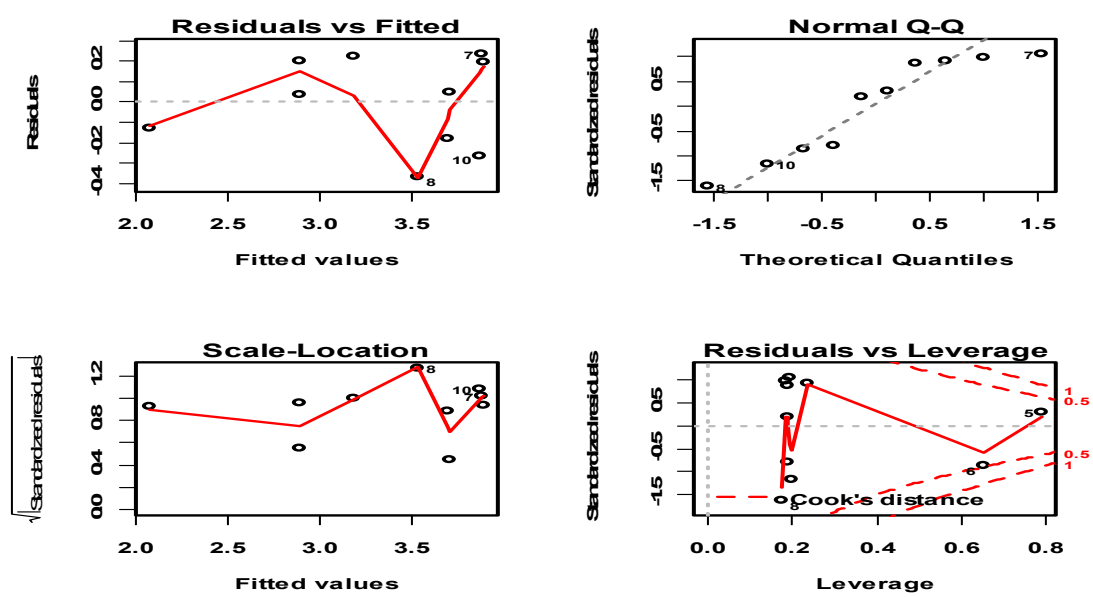
```
range(size)

[1]  6.8 30.9

predict.frame <- data.frame(size=6:40)

b<-predict(lm.grocery,newdata=predict.frame)

par(font=2,font.axis=2,font.lab=2, lwd=2)

matplot(size,value,"p",pch=17,ylim=range(1,5))

pred.size <- predict.frame$size

matlines(pred.size,b,type="l",col="black",lwd=2)
```

Figure 13.9: Fitted quadratic regression line for Example 13.4




```
par(font=2,font.axis=2,font.lab=2, lwd=2)

par(mfrow=c(2,2))

plot(lm.grocery)

par(mfrow=c(1,1))
```

Figure 13.10: R diagnostic plots for Example 13.4



13.5 USING QUANTITATIVE AND QUALITATIVE PREDICTOR VARIABLES IN A REGRESSION MODEL (p. 566 in text)

One reason multiple regression models are very flexible is that they allow for the use of both *qualitative* and *quantitative* predictor variables. A **quantitative variable** x can be entered as a linear term, x, or to some higher power such as x^2 or x^3, as in the quadratic model in Example 13.3. When more than one quantitative variable is necessary, the interpretation of the possible models becomes more complicated. For example, with two quantitative variables x_1 and x_2, you could use a **first-order model** such as

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

which traces a plane in three-dimensional space (see Figure 13.1). However, it may be that one of those variables – say, x_2 – is not related to y in the same way when $x_1 = 1$ as it is when $x_1 = 2$. To allow x_2 to behave differently depending on the value of x_1, we add an **interaction term**, $x_1 x_2$, and allow the two-dimensional plane to *twist*. The model is now a **second-order model**:

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$$

In contrast to quantitative predictor variables, **qualitative predictor variables** are entered into a regression model through **dummy** or **indicator variables**.

Example 13.6

Random samples of six female and six male assistant professors were selected from among the assistant professors in a college of arts and sciences. The data on salary and years of experience are shown in Table 13.3. Note that each of the two samples (male and female) contained two professors with 3 years of experience, but no male professor had 2 years of experience.

Table 13.3: Salary versus Gender and Years of Experience

Years of Experience, x_1	Salary for Men, y	Salary for Women, y
1	60710	59510
2	-	60440
3	63160	61340
3	63210	61760
4	64140	62750
5	65760	63200
5	65590	-

Solution The R code is shown for the data in Table 13.3 and printout in Figure 13.12.

```
years=c(1,1,2,2,3,3,3,3,4,4,5,5,5,5)
salary=c(60710,59510,NA,60440,63160,61340,63210,61760,64140,62750,65760,
         63200,65590,NA)
sex=as.factor(c("M","F","M","F","M","F","M","F","M","F","M","F","M","F"))
university=data.frame(years,salary,sex)
university
```

	years	salary	sex
1	1	60710	M
2	1	59510	F
3	2	NA	M
4	2	60440	F
5	3	63160	M
6	3	61340	F
7	3	63210	M
8	3	61760	F
9	4	64140	M
10	4	62750	F
11	5	65760	M
12	5	63200	F
13	5	65590	M
14	5	NA	F

```
lm.university <- lm(salary ~ years + sex + years:sex)
summary(lm.university)
```

Figure 13.12: R Regression Analysis output for Example 13.6

```
Call:     lm(formula = salary ~ years + sex + years:sex)

Residuals:      Min        1Q        Median       3Q         Max

             -238.000   -108.250     -1.232     85.833     281.000

Coefficients:    Estimate     Std. Error    t value     Pr(>|t|)

   (Intercept)   58593.00      207.95       281.769     < 2e-16 ***

        years      969.00       63.67        15.219    3.44e-07 ***

         sexM      866.71      305.26         2.839    0.0218 *

    years:sexM     260.13       87.06         2.988    0.0174 *

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

    Residual standard error: 201.3 on 8 degrees of freedom

    (2 observations deleted due to missingness)

    Multiple R-squared: 0.9924,     Adjusted R-squared: 0.9895

    F-statistic: 346.2 on 3 and 8 DF,  p-value: 8.372e-09
```

```
anova(lm.university)
```

Figure 13.12: R ANOVA output for Example 13.6

```
Analysis of Variance Table

Response: salary

            Df       Sum Sq       Mean Sq      F value      Pr(>F)

   years    1       33294036      33294036     821.2774    2.377e-09 ***

    sex     1        8452797       8452797     208.5085    5.174e-07 ***

 years:sex  1         361944        361944       8.9282    0.01739 *

 Residuals  8         324315         40539

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

salary = b0 + b1years + b2sex + b3years*sex

salary for female = b0 + b1years + b2(0) + b3(0)

= b0 + b1years

= 58593 + 969years

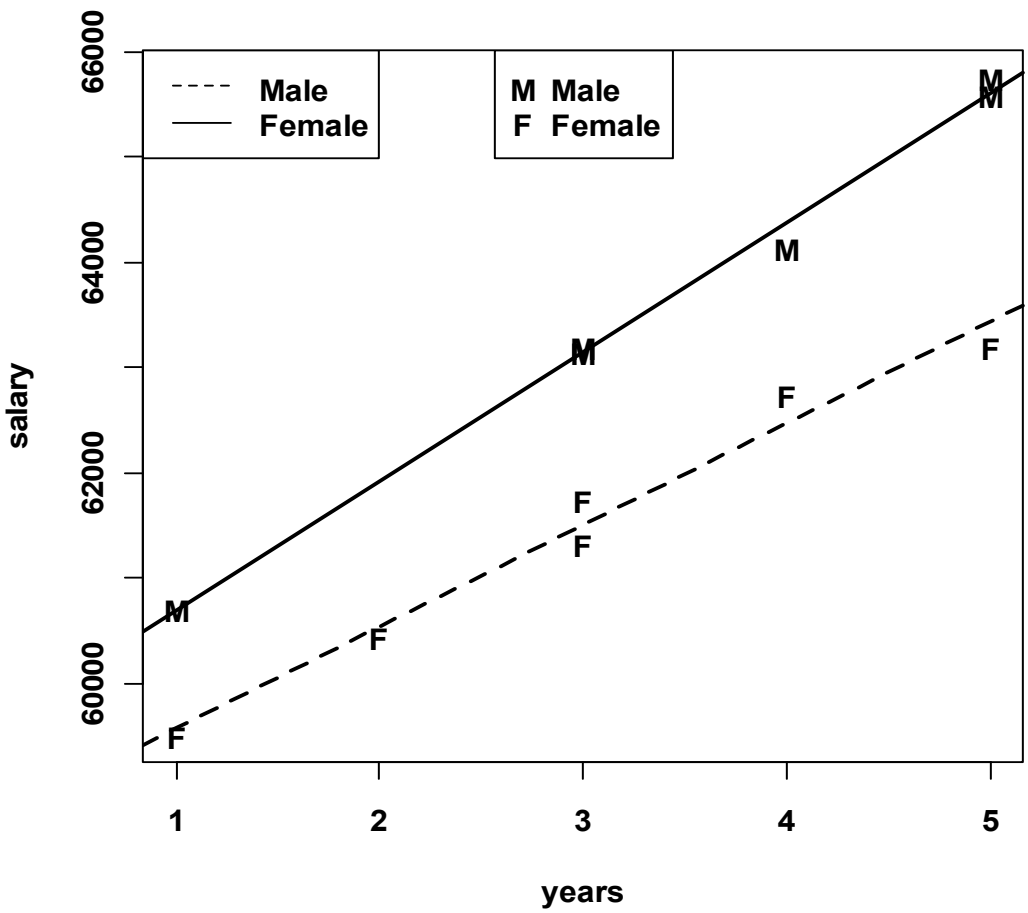
E(Y) for Male = b0 + b1years + b2(1) + b3(1)years

= 58593 + 969years + 866.71 + 260.13years

= 59459.71 + 1229.13years

```
par(font=2, font.axis =2, font.lab=2)

plot(salary~years,pch=ifelse(university$sex=="M","M","F"))

legend("top",legend=c("Male","Female"),pch=c("M","F"))

abline(58593,969,lwd=2,lty=2)

abline(59459.71,1229.13,lwd=2)

legend("topleft",legend=c("Male","Female"),lty=2:1)
```

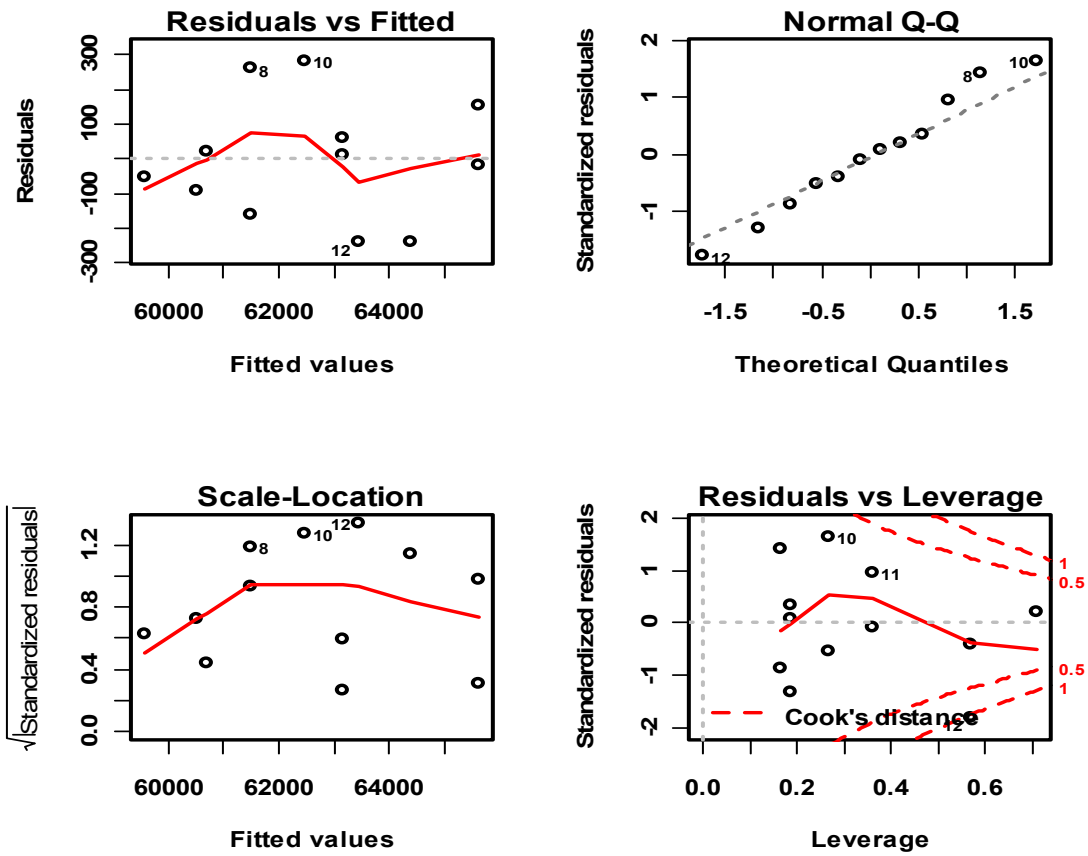
Figure 13.13: A graph of the faculty salary prediction lines for Example 13.6



Finally, check the residual plots to make sure that there are no strong violations of the regression assumptions. These plots, which behave as expected for a properly fit model, are shown in Figure 13.14.

```
par(font=2, font.axis =2, font.lab=2)

par(mfrow=c(2,2))

plot(lm.university)

par(mfrow=c(1,1))
```

Figure 13.14: R residual plots for Example 13.6



Example 13.7

Refer to Example 13.6. Do the data provide sufficient evidence to indicate that the annual rate of increase in male junior faculty salaries exceeds the annual rate of increase in female junior faculty salaries? That is, do the data provide sufficient evidence to indicate that the slope of the men's faculty salary line is greater than the slope of the women's faculty salary line?

Solution Since β_3 measures the difference in slopes, the slopes of the two lines will be identical if $\beta_3 = 0$. Therefore, you want to test the null hypothesis

$$H_0 : \beta_3 = 0$$

- that is, the slopes of the two lines are identical – versus the alternative hypothesis

$$H_a : \beta_3 > 0$$

- that is, the slope of the men's faculty salary line is greater than the slope of the women's faculty salary line.

The calculated value of t corresponding to β_3, shown in the row labeled `years:sexM` in Figure 13.12, is `2.988`. Since the R regression summary output provides p-values for two-tailed significance tests, the p-value in the printout, `0.0174 *`, is *twice* what it would be for a one-tailed test. For this one-tailed test, the p-value is 0.0174/2 = 0.0087, and the null hypothesis is rejected. There is sufficient evidence to indicate that the annual rate of increase in men's faculty salaries exceeds the rate for women. The `*` indicates significance at the 0.05 level.

13.6 TESTING SETS OF REGRESSION COEFFICIENTS (p. 575 in text)

In the preceding sections, you have tested the complete set of partial regression coefficients using the F-test for the overall fit of the model, and you have tested the partial regression coefficients individually using the Student's t-test. Besides these two important tests, you might want to test hypotheses about some subsets of these regression coefficients.

Example 13.8

Refer to the real estate data of Example 13.2 that relate the listed selling price y to the square feet of living area x_1, the number of floors x_2, the number of bedrooms x_3, and the number of bathrooms, x_4. The realtor suspects that the square footage of living area is the most important predictor variable and that the other variables might be eliminated from the model without loss of much prediction information. Test this claim with α = .05.

Solution The hypothesis to be tested is $H_0 : \beta_2 = \beta_3 = \beta_4 = 0$ versus the alternative hypothesis that at least one of β_2, β_3, or β_4 is different from 0. The **complete model 2**, given as

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \varepsilon$$

was fitted in Example 13.2. A portion of the R printout from Figure 13.3 is reproduced in Figure 13.15 along with a portion of the R printout for the simple linear regression analysis of the **reduced model 1**, given as

$$y = \beta_0 + \beta_1 x_1 + \varepsilon$$

```
lm.full <- lm(price ~ area + floors + bed + bath, data=condos)

lm.reduced <- lm(price ~ area, data=condos)

summary(lm.full)
```

Figure 13.15: Portion of the R regression printout for complete model for Example 13.8

```
Residual standard error: 6.849 on 10 degrees of freedom

Multiple R-squared: 0.9714,     Adjusted R-squared: 0.9599

F-statistic:  84.8 on 4 and 10 DF,  p-value: 1.128e-07
```

```
summary(lm.reduced)
```

Figure 13.15: Portion of the R regression printout for reduced model for Example 13.8

```
Residual standard error: 10.93 on 13 degrees of freedom

Multiple R-squared: 0.9052,     Adjusted R-squared: 0.8979

F-statistic: 124.1 on 1 and 13 DF,  p-value: 5.061e-08
```

```
anova(lm.reduced,lm.full)

Analysis of Variance Table

Model 1: price ~ area

Model 2: price ~ area + floors + bed + bath

  Res.Df       RSS       Df      Sum of Sq      F       Pr(>F)

1     13       1552.88

2     10        469.13      3       1083.8      7.7004   0.00589  **

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

The critical value of F with $\alpha = .05$, $df_1 = 3$, and $df_2 = n - (k + 1) = 15 - (4 + 1) = 10$ is $F_{.05} = 3.71$. Hence, H_0 I rejected. There is evidence to indicate that at least one of the three variables – number of floors, bedrooms, or bathrooms – is contributing significant information for predicting the listed selling price. This is supported by Akaike's Information Criterion (AIC) for which smaller values exhibit a better fit.

```
AIC(lm.full,lm.reduced)

                df          AIC

lm.full         6         106.2106

lm.reduced      3         118.1654
```

LOGISTIC REGRESSION (Hosmer Supplements)

The goal of logistic regression is to find the best fitting, simplest model possible describing the relationship between an outcome (dependent or response) variable and a set of independent (predictor or explanatory) variables. What distinguishes a logistic regression from the linear regression model is that the outcome variable is binary (or dichotomous). The techniques used in linear regression analysis will provide the motivation for our approach to logistic regression.

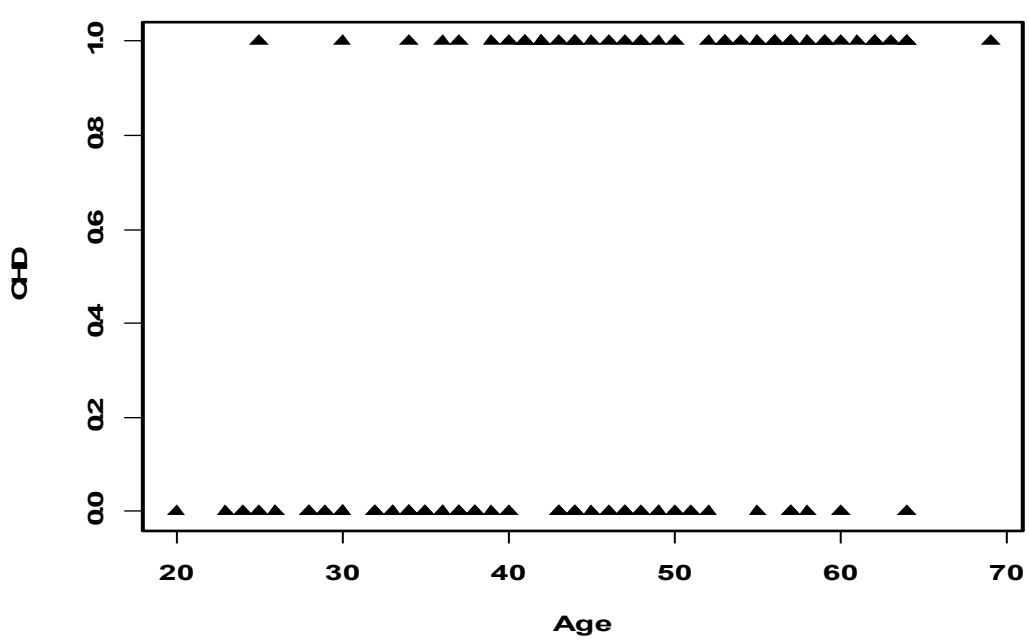
Example

AGE (yrs) and presence or absence of evidence of significant coronary hearts disease (CHD) were recorded for 100 subjects selected to participate in a study.

Let us explore the relationship between AGE and presence or absence of CHD. Had our outcome variable been continuous rather than binary we would probably have begun by creating a scatter plot of the dependent vs. the independent variable.

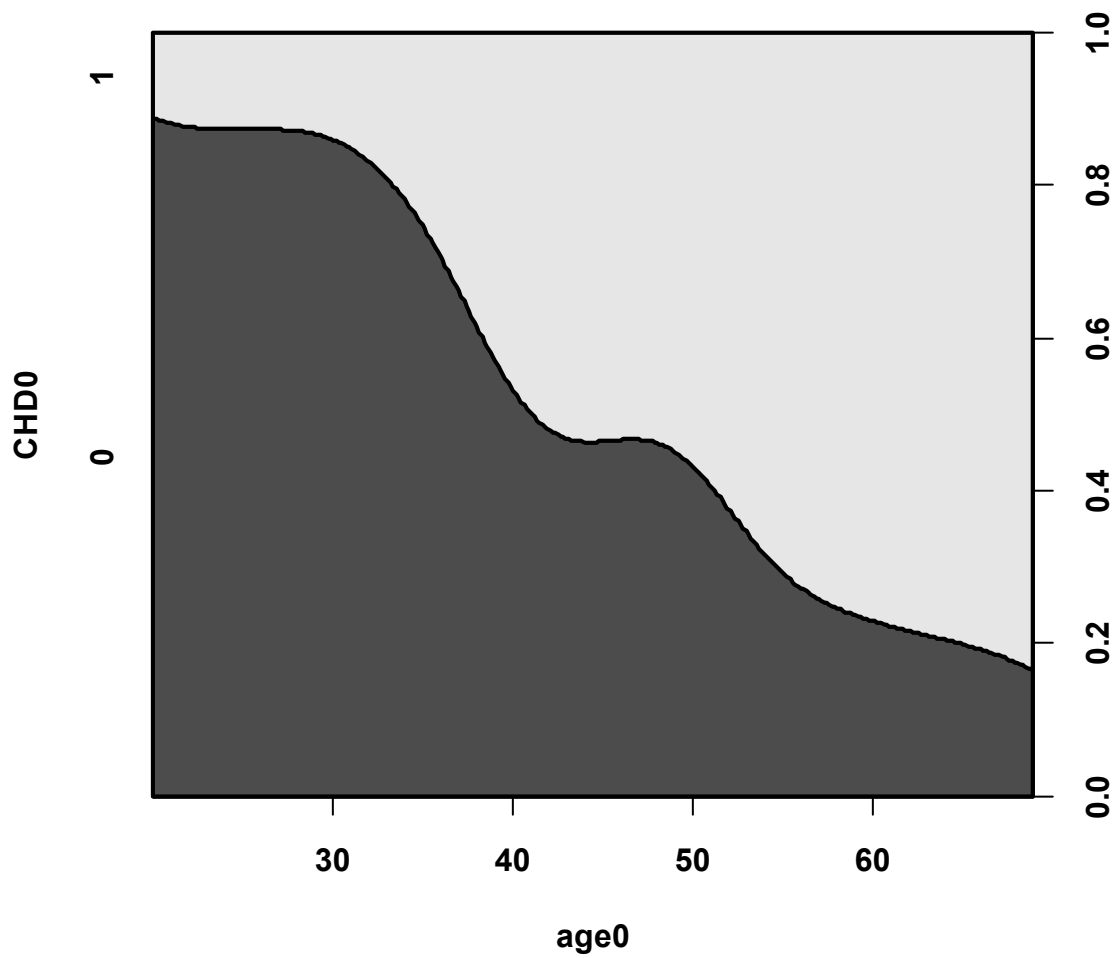
```
age0=c(20,23,24,25,25,26,26,28,28,29,30,30,30,30,30,30,32,32,33,33,34,34,34,
       4,34,35,35,36,36,36,37,37,37,38,38,39,39,40,40,41,41,42,42,42,42,43,
       43,43,44,44,44,44,45,45,46,46,47,47,47,48,48,48,49,49,49,50,50,51,52,
       52,53,53,54,55,55,55,56,56,56,57,57,57,57,57,57,58,58,58,59,59,60,60,
       61,62,62,63,64,64,64,69)
CHD0=as.factor(c(rep(0,4),1,rep(0,10),1,rep(0,6),1,rep(0,5),1,0,0,1,rep(0,4),
               1,0,1,rep(1,5),1,0,0,1,0,0,1,1,0,1,0,1,0,0,1,0,1,1,0,0,1,0,1,
               0,0,1,1,1,1,0,1,1,1,1,1,0,0,1,1,1,1,0,1,1,1,1,0,1,1,1,1,1,0,1,
               1,1))
plot(age0,CHD0,"p",pch=17,xlab="Age",ylab="CHD", lwd=2)
```

This plot would help us assess the relationship between *x* and *y*.



Clearly, in this scatterplot, all points fall on one of two parallel lines representing CHD = 0 and CHD = 1.
We can see that there is some tendency for the individuals with no evidence of CHD (y = 0) to be younger than those with CHD (y = 1).
While this plot does depict the dichotomous nature of the outcome variable quite clearly, it does not provide a clear picture of the nature of the relationship between CHD and AGE.
Next, we look at conditional density plots of the response variable given the explanatory variable with the `cdplot` command. This plot describes how the conditional distribution of the categorical variable CHD changes as the numerical variable age changes.

```
cdplot(CHD0 ~ age0)
```



To better explore this relationship let us create intervals for the independent variable and compute the mean of the outcome variable within each group.

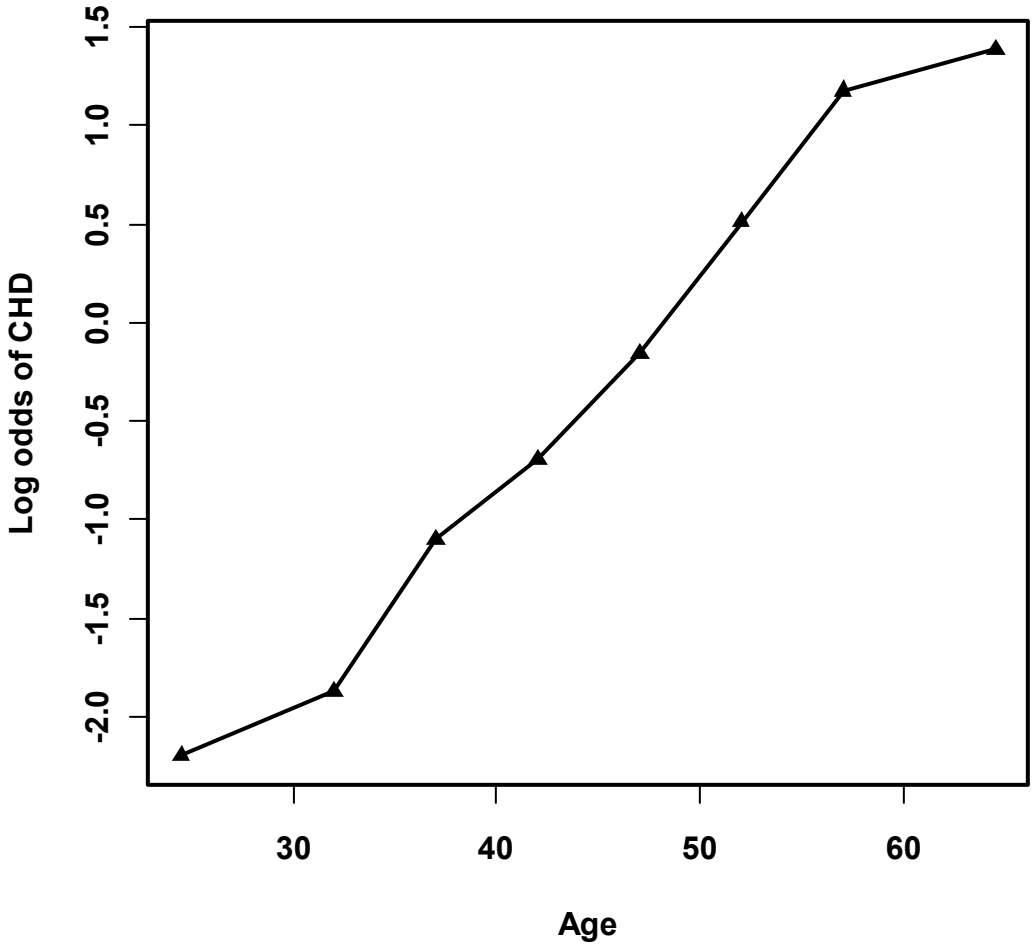
Age Group	n	CHD Absent	CHD Present	Mean Present
20-29	10	9	1	0.10
30-34	15	13	2	0.13
35-39	12	9	3	0.25
40-44	15	10	5	0.33
45-49	13	7	6	0.46
50-54	8	3	5	0.63
55-59	17	4	13	0.76
60-69	10	2	8	0.80
	100	57	43	

Here we see that as age increases, the proportion of individuals with evidence of CHD increases. To reproduce this table in R:

```
n=c(10,15,12,15,13,8,17,10)

y=c(1,2,3,5,6,5,13,8)

p=y/n

age1<-c(mean(c(20,29)),mean(c(30,34)),mean(c(35,39)),mean(c(40,44)),

        mean(c(45,49)),mean(c(50,54)),mean(c(55,59)),mean(c(60,69)))

CHD=data.frame(n,y,p)

CHD

      n      y          p

1    10      1     0.1000000

2    15      2     0.1333333

3    12      3     0.2500000

4    15      5     0.3333333

5    13      6     0.4615385

6     8      5     0.6250000

7    17     13     0.7647059

8    10      8     0.8000000
```

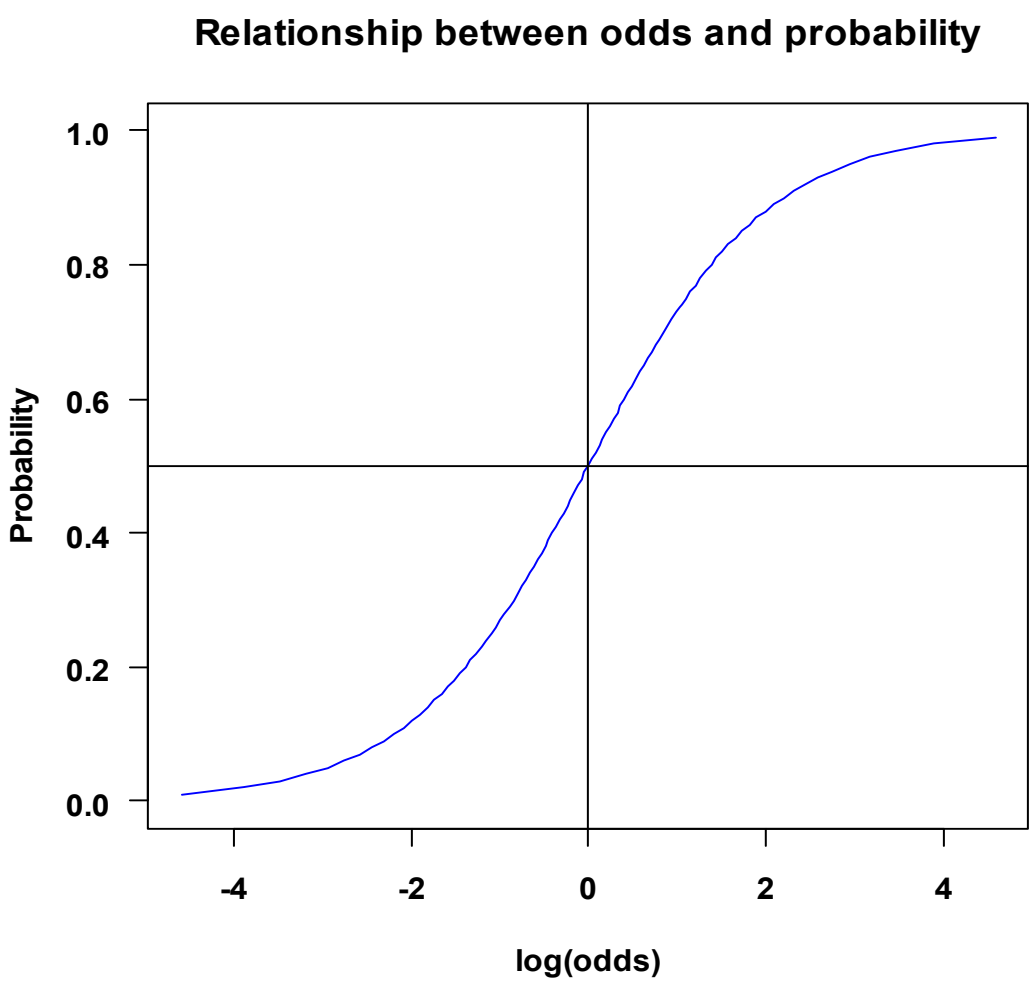
For a reproduction of the graphic showing the proportion of individuals with evidence of CHD increases in R:

```
par(font=2,font.axis=2,font.lab=2)

odds<- p/(1-p)

lnodds<-log(odds)

plot(age1,lnodds,"p",pch=17,xlab="Age",ylab="Log odds of CHD", lwd=2)

lines(lnodds~age1)
```



To get a clear picture of a textbook sigmoidal logistic curve, we create our own sequence from 0 to 1 by increments of .01. Notice the graph for the CHD data looks very similar.

```
p <- seq(from=0, to=1, by=.01)
odds <- p/(1-p)
plot(log(odds), p, type="l", col="blue", ylab="Probability",
          main="Relationship between odds and probability", las=1)
abline(h=.5)
abline(v=0)
```



Now, we want to rearrange the form in which we input our data so we account for the age brackets we have created. Then we can apply a generalized linear model using the `glm` function and specifying `family=binomial`.

```
CHD1=c(1,rep(0,9),rep(1,2),rep(0,13),rep(1,3),rep(0,9),rep(1,5),rep(0,10),
       rep(1,6),rep(0,7),rep(1,5),rep(0,3),rep(1,13),rep(0,4),rep(1,8),
       rep(0,2))
age1<-c(rep(mean(c(20,29)),10),rep(mean(c(30,34)),15),rep(mean(c(35,39)),12),
        rep(mean(c(40,44)),15),rep(mean(c(45,49)),13),rep(mean(c(50,54)),8),
        rep(mean(c(55,59)),17),rep(mean(c(60,69)),10))
glm1<-glm(CHD1~age1,family=binomial)
```

To reproduce in R the confidence intervals as shown on page 10 of the first Hosmer PDF:

```
confint(glm1)
Waiting for profiling to be done...
                     2.5 %          97.5 %
     (Intercept)   -7.34905895     -3.0568963
          age1      0.06276942      0.1539715
```

To receive a summary of the model, we simply apply the `summary` command just as we did with simple linear and multiple regression.

```
summary(glm1)
   Call:   glm(formula = CHD1 ~ age1, family = binomial)
   Deviance Residuals:     Min       1Q       Median       3Q        Max
                        -1.9483   -0.9250    -0.4039     0.8094    2.2569
   Coefficients:    Estimate     Std. Error    z value       Pr(>|z|)
      (Intercept)   -5.03822      1.08626      -4.638       3.52e-06  ***
          age1       0.10502      0.02308       4.551       5.35e-06  ***
           Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
           (Dispersion parameter for binomial family taken to be 1)
    Null deviance: 136.66  on 99  degrees of freedom
Residual deviance: 108.49  on 98  degrees of freedom
AIC: 112.49             Number of Fisher Scoring iterations: 4
```

Thus, $$logit = -5.038 + 0.105age$$

Notice the number of iterations is the same as shown on page 10 of the first Hosmer PDF.

To tease out the odds ratio for the *age* variable we simply apply the `exp` function in R:

```
exp(coefficients(summary(glm1))[2,1])
```

```
[1] 1.110732
```

To retrieve a covariance matrix of the coefficients similar to the one page 16 of the first Hosmer PDF, we apply the `vcov` command to our model:

```
vcov(glm1)
```

```
                  (Intercept)         age1
  (Intercept)     1.17996637      -0.0244761778
        age1     -0.02447618       0.0005325724
```

To receive a calculation of the log-likelihood estimate as shown on page 10 of the first Hosmer PDF, we first need to install and load the **epicalc** package and then apply the `logistic.display` function to our model. Notice that this also provides us with an odds ratio estimate (along with a 95% confidence interval) and a Wald's test *p*-value.

```
library(epicalc)

logistic.display(glm1)
```

```
         Logistic regression predicting CHD1

                       OR       (95%CI)       P(Wald's test)    P(LR-test)

 age1 (cont. var.)     1.11     (1.06,1.16)      < 0.001          < 0.001

     Log-likelihood = -54.2428

     No. of observations = 100

     AIC value = 112.4856
```

Next, we input our data from that provided on page 30 of the second Hosmer PDF and perform the same type of analysis:

		Age		
		≥ 55 (1)	< 55 (0)	
CHD	1	21	22	43
	0	6	51	57
		27	73	100

```
CHD2=c(rep(1,43),rep(0,57))

age2=c(rep(1,21),rep(0,22),rep(1,6),rep(0,51))

glm2<-glm(CHD2~age2,family=binomial)

summary(glm2)

   Call:   glm(formula = CHD2 ~ age2, family = binomial)

   Deviance Residuals:   Min       1Q        Median       3Q         Max

                        -1.734    -0.847      -0.847      0.709      1.549

   Coefficients:   Estimate      Std. Error     z value     Pr(>|z|)

      (Intercept)  -0.8408         0.2551       -3.296      0.00098  ***

           age2     2.0935         0.5285        3.961      7.46e-05 ***

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

              (Dispersion parameter for binomial family taken to be 1)

    Null deviance: 136.66  on 99  degrees of freedom

Residual deviance: 117.96  on 98  degrees of freedom

AIC: 121.96

Number of Fisher Scoring iterations: 4
```

Thus, $logit = -0.841 + 2.094age$

```
logistic.display(glm2)

                      Logistic regression predicting CHD2

                                                    OR          (95%CI)
P(Wald's test)     P(LR-test)

                 age2: 1 vs 0            8.11      (2.88,22.86)            <
0.001                  < 0.001

     Log-likelihood = -58.9796

     No. of observations = 100

     AIC value = 121.9591
```

Next we show in R how to analyze the data using the polytomous independent variable from page 33 of the second Hosmer PDF. Suppose now that *x* has three or more levels in which there are a fixed number of outcomes and the scale of measurement is nominal. We must form a set of design variables to represent the categories of the variable.

	CHD		
Race	**Present**	**Absent**	**Total**
White	5	20	25
Black	20	10	30
Hispanic	15	10	25
Other	10	10	20
Total	50	50	100

Let us use White as the reference group.

```
CHD3=c(rep(1,5),rep(0,20),rep(1,20),rep(0,10),rep(1,15),rep(0,10),rep(1,10),

       rep(0,10))

race=as.factor(c(rep("1White",25),rep("Black",30),rep("Hispanic",25),

               rep("Other",20)))

glm3=glm(CHD3~race,family=binomial)

summary(glm3)
```

```
Call:   glm(formula = CHD3 ~ race, family = binomial)

Deviance Residuals:   Min        1Q        Median       3Q         Max

                    -1.4823    -1.1774     0.1162      1.0108    1.7941

Coefficients:   Estimate     Std. Error      z value      Pr(>|z|)

  (Intercept)   -1.3863        0.5000        -2.773      0.00556   **

   raceBlack     2.0794        0.6325         3.288      0.00101   **

 raceHispanic    1.7918        0.6455         2.776      0.00551   **

   raceOther     1.3863        0.6708         2.067      0.03878    *

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

             (Dispersion parameter for binomial family taken to be 1)

    Null deviance: 138.63  on 99  degrees of freedom

Residual deviance: 124.59  on 96  degrees of freedom

AIC: 132.59

Number of Fisher Scoring iterations: 4


logistic.display(glm3)

        Logistic regression predicting CHD3

                        OR       (95%CI)       P(Wald's test)     P(LR-test)

race: ref.=1White

        Black           8       (2.32,27.63)        0.001           0.003

      Hispanic          6       (1.69,21.26)        0.006

        Other           4        (1.07,14.9)        0.039

                 Log-likelihood = -62.2937

                 No. of observations = 100

                 AIC value = 132.5874
```

CHAPTER 11: THE ANALYSIS OF VARIANCE

11.5 THE ANALYSIS OF VARIANCE FOR A COMPLETELY RANDOMIZED DESIGN (p. 451 in text)

Suppose you want to compare k population means, $\mu_1, \mu_2, \ldots, \mu_k$, base on independent random samples of size $n_1, n_2, \ldots, n_k$ from normal populations with a common variance σ^2. That is, each of the normal populations has the same shape, but their locations might be different.

ANOVA Table For *k* Independent Random Samples: Completely Randomized Design

Source	***df***	**SS**	**MS**	***F***
Treatments	$k - 1$	SST	MST = SST/(k – 1)	MST/MSE
Error	n – k	SSE	MSE = SSE/(n – k)	
Total	$n - 1$	Total SS		

where

$$Total\ SS = \sum x_{ij}^2 - CM$$

= (Sum of squares of all x-values) – CM

with

$$CM = \frac{(\sum x_{ij})^2}{n} = \frac{G^2}{n}$$

$$SST = \sum \frac{T_i^2}{n_i} - CM$$

$$MST = \frac{SST}{k-1}$$

$$SSE = Total\ SS - SST$$

$$MSE = \frac{SSE}{n-k}$$

and

G = Grand total of all n observations

T_i = Total of all observations in sample i

n_i = Number of observations in sample i

$n = n_1 + n_2 + \ldots + n_k$

Example 11.4

In an experiment to determine the effect of nutrition on the attention spans of elementary school students, a group of 15 students were randomly assigned to each of three meal plans: no breakfast, light breakfast, and full breakfast. Their attention spans (in minutes) were recorded during a morning reading period and are shown in Table 11.1. Construct the analysis of variance table for this experiment.

Table 11.1 Attention Spans of Students After Three Meal Plans

No Breakfast	Light Breakfast	Full Breakfast
8	14	10
7	16	12
9	12	16
13	17	15
10	11	12
$T_1 = 47$	$T_2 = 70$	$T_3 = 65$

Solution To use the calculational formulas, you need the $k = 3$ treatment totals together with $n_1 = n_2 = n_3 = 5$, $n = 15$, and $\sum x_{ij} = 182$. Then

$$CM = \frac{(182)^2}{15} = 2208.2667$$

Total SS = $(8^2 + 7^2 + \ldots + 12^2) - \text{CM} = 2338 - 2208.2667 = 129.7333$ with $(n - 1) =$

$(15 - 1) = 14$ degrees of freedom,

$$SST = \frac{47^2 + 70^2 + 65^2}{5} - CM = 2266.8 - 2208.2667 = 58.5333$$

with $(k - 1) = (3 - 1) = 2$ degrees of freedom, and by subtraction,

SSE = Total SS – SST = 129.7333 – 58.5333=71.2

with $(n - k) = (15 - 3) = 12$ degrees of freedom. These three sources of variation, their degrees of freedom, sums of squares, and mean squares are shown in the following Figure 11.3 output generated by R. First, we input the data and reproduce Table 11.1 in R.

```
meal1=c(8,7,9,13,10)

meal2=c(14,16,12,17,11)

meal3=c(10,12,16,15,12)

spans=data.frame(meal1, meal2, meal3)

spans

              values         ind

     1        8              meal1

     2        7              meal1

     3        9              meal1

     4        13             meal1

     5        10             meal1

     6        14             meal2

     7        16             meal2

     8        12             meal2

     9        17             meal2

     10       11             meal2

     11       10             meal3

     12       12             meal3

     13       16             meal3

     14       15             meal3

     15       12             meal3

spans=stack(spans)

oneway.test(values ~ ind, data= spans, var.equal=T)
```

Figure 11.3: R overall output for Example 11.4

```
One-way analysis of means

 data:  values and ind

 F = 4.9326, num df = 2, denom df = 12,p-value = 0.02733
```

```
lm.spans <- lm(values ~ ind, data=spans)

anova(lm.spans)
```

Figure 11.3: R ANOVA output for Example 11.4

```
Analysis of Variance Table

Response: values

          Df        Sum Sq      Mean Sq      F value      Pr(>F)

  ind      2        58.533      29.2667      4.9326      0.02733 *

Residuals 12        71.200       5.9333

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

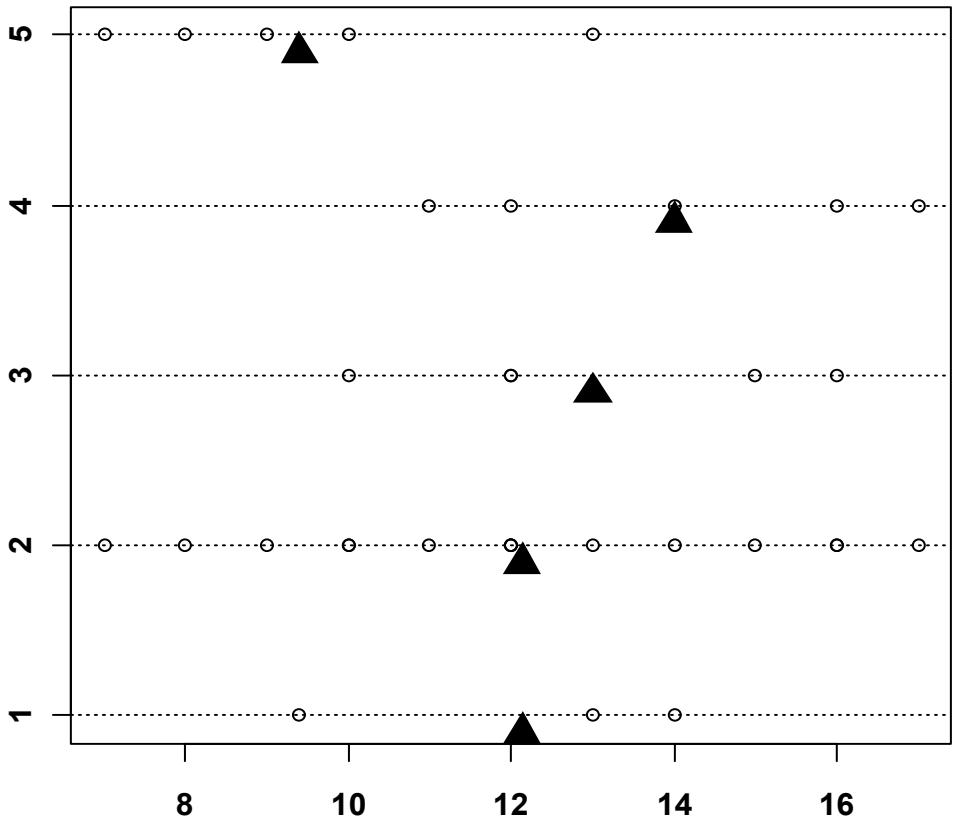
```
summary(lm.spans)
```

Figure 11.3: R summary output for Example 11.4

```
Call:      lm(formula = values ~ ind, data = spans)

Residuals:   Min        1Q        Median       3Q          Max

            -3.0       -1.7        -0.4        2.0         3.6

Coefficients:   Estimate     Std. Error     t value      Pr(>|t|)

  (Intercept)    9.400        1.089         8.629       1.72e-06  ***

    indmeal2     4.600        1.541         2.986       0.0114    *

    indmeal3     3.600        1.541         2.337       0.0376    *

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

            Residual standard error: 2.436 on 12 degrees of freedom

            Multiple R-squared: 0.4512,     Adjusted R-squared: 0.3597

            F-statistic: 4.933 on 2 and 12 DF,  p-value: 0.02733
```

To wrap our brain around this data visually, we use the function `simple.fancy.stripchart` from the **UsingR** package.

```
library(UsingR)

simple.fancy.stripchart(list(c(mean(meal1),mean(meal2),mean(meal3)),

                              spans$values,meal3,meal2,meal1))
```



The first line of the stripchart shows the overall mean (triangle) compared to the means for the three treatment groups. The second line shows the overall mean compared to every data point. The last three lines reveal the means for each treatment groups along with their respective data points. Notice the triangles for the last three lines match up with their means from the first line.

Testing the Equality of the Treatment Means

The *mean squares* in the analysis of variance table can be used to test the null hypothesis

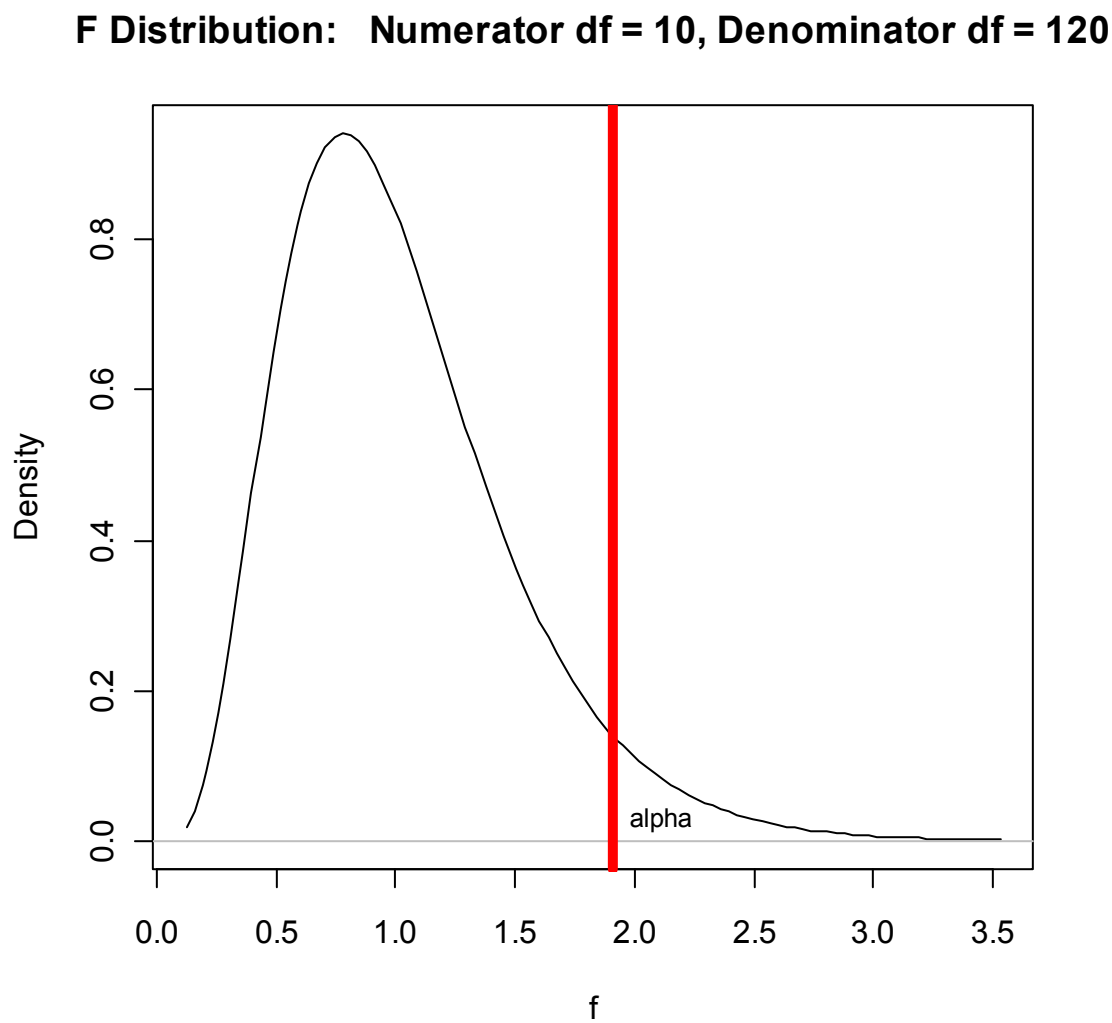
$$H_0: \mu_1 = \mu_2 = \cdots = \mu_k$$

versus the alternative hypothesis

$$H_{\mathrm{a}}: \text{At least one of the means is different from the others}$$

F Test For Comparing *k* Population Means

1. Null hypothesis: H_0: $\mu_1 = \mu_2 = \ldots = \mu_k$
2. Alternative hypothesis: H_a: One or more pairs of population means differ
3. Test statistic: F = MST/MSE, where F is based on $df_1 = (k - 1)$ and $df_2 = (n - k)$
4. Rejection region: Reject H_0 if $F > F_\alpha$, where F_α lies in the upper tail of the F distribution (with $df_1 = k - 1$ and $df_2 = n - k$) or if the p-value $< \alpha$



Assumptions

- The samples are randomly and independently selected from their respective populations.
- The populations are normally distributed with means $\mu_1, \mu_2, \ldots, \mu_k$ and equal variances, $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_3^2 = \sigma^2$.

The R code to create the *F*-distribution above was:

```
.x <- seq(0.122, 3.533, length.out=100)
plot(.x, df(.x, df1=10, df2=100), xlab="f", ylab="Density",
         main="F Distribution:   Numerator df = 10, Denominator df = 100",
type="l")
abline(h=0, col="gray")
remove(.x)
abline(v=1.91, col="red",lwd=5)
text(locator(1),"alpha",cex=.8,lwd=2)
```

Example 11.5

Do the data in Example 11.4 provide sufficient evidence to indicate a difference in the average attention spans depending on the type of breakfast eaten by the student?

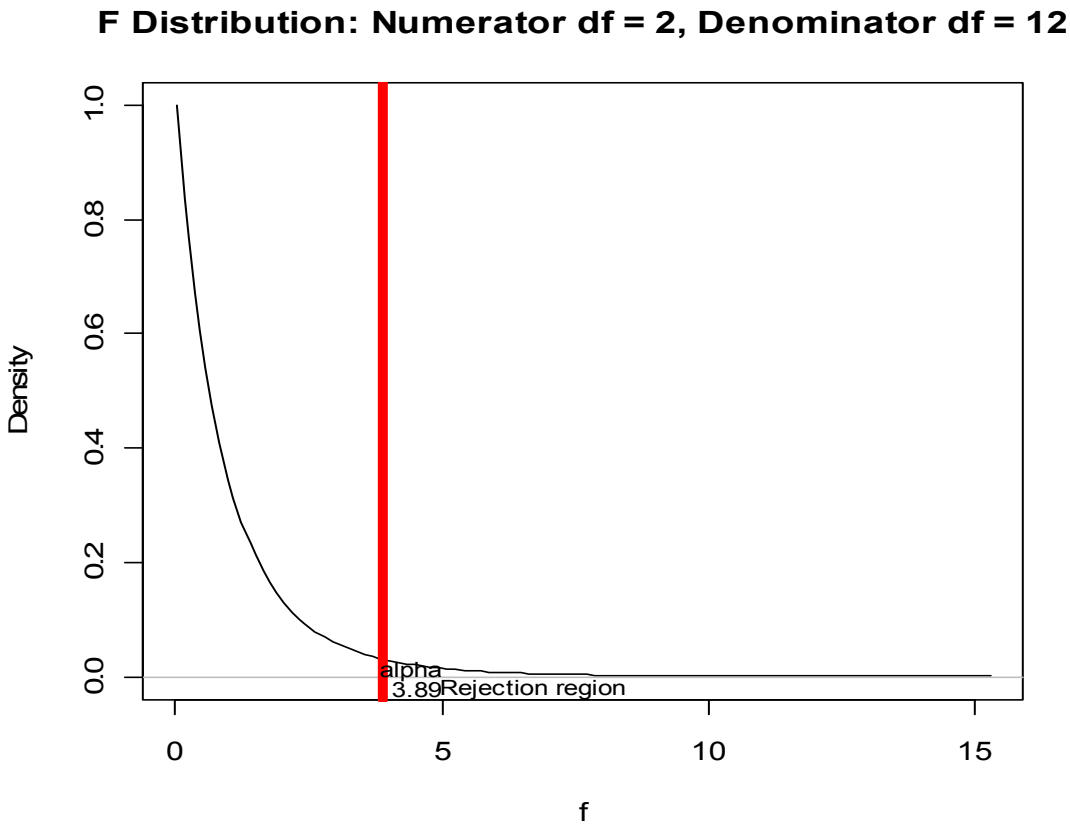
Solution To test $H_0: \mu_1 = \mu_2 = \mu_3$ versus the alternative hypothesis that the average attention span is different for at least one of the three treatments, you use the analysis of variance *F* statistic, calculated as

$$F = \frac{MST}{MSE} = \frac{29.2667}{5.9333} = 4.93$$

and shown as `F` in Figure 11.3: R overall output for Example 11.4, in the `F value` column of Figure 11.3: R ANOVA output for Example 11.4, and as `F-statistic` at the bottom of Figure 11.3: R summary output for Example 11.4. The corresponding exact p-values are found under `p-value` in Figure 11.3: R overall output for Example 11.4, the `Pr(>|t|)` column in Figure 11.3: R ANOVA output for Example 11.4, and `p-value` at the bottom of Figure 11.3: R summary output for Example 11.4. The test statistic MST/MSE calculated above has an *F* distribution with $df_1 = 2$ and $df_2 = 12$ degrees of freedom. Using the critical value approach with $\alpha = .05$, you can reject H_0 if $F > F_{.05} = 3.89$ (see Figure 11.5). Since the observed value, $F = 4.93$, exceeds the critical value, you reject H_0. There is sufficient evidence to indicate that at least one of the three average attention spans is different from at least one of the others. For a reproduction in R of Figure 11.5 in the book:

```
.x <- seq(0.001, 15.297, length.out=100)
plot(.x, df(.x, df1=2, df2=12), xlab="f", ylab="Density",
         main="F Distribution: Numerator df = 2, Denominator df = 12",
type="l")
abline(h=0, col="gray")
remove(.x)
abline(v=3.89, col="red",lwd=5)
text(locator(1),"alpha",cex=.8,lwd=2)
text(locator(1),"3.89",cex=.8,lwd=2)
text(locator(1),"Rejection region",cex=.8,lwd=2)
```

Figure 11.5: Rejection region for Example 11.5



Completely Randomized Design: (1 – α)100% Confidence Intervals for a Single Treatment Mean and the Difference Between Two Treatment Means

Single treatment mean:

$$\bar{x}_i \pm t_{\alpha/2}\left(\frac{s}{\sqrt{n_i}}\right)$$

Difference between two treatment means:

$$\left(\bar{x}_i - \bar{x}_j\right) \pm t_{\alpha/2}\sqrt{s^2\left(\frac{1}{n_i}+\frac{1}{n_j}\right)}$$

with

$$s = \sqrt{s^2} = \sqrt{MSE} = \sqrt{\frac{SSE}{n-k}}$$

where $n = n_1 + n_2 + \ldots + n_k$ and $t_{\alpha/2}$ is based on $(n - k)$ *df.*

Example 11.6

The researcher in Example 11.4 believes that students who have no breakfast will have significantly shorter attention spans but that there may be no difference between those who eat a light or a full breakfast. Find a 95% confidence interval for the average attention span for students who eat no breakfast, as well as a 95% confidence interval for the difference in the average attention spans for light versus full breakfast eaters.

Solution For $s^2 = MSE = 5.9333$ so that $s = \sqrt{5.9333} = 2.436$ with $df = (n - k) = 12$, you can calculate the two confidence intervals:

- For no breakfast:

$$\bar{x}_1 \pm t_{\alpha/2}\left(\frac{s}{\sqrt{n_1}}\right)$$
$$9.4 \pm 2.179\left(\frac{2.436}{\sqrt{5}}\right)$$
$$9.4 \pm 2.37$$

- For light versus full breakfast:

$$(\bar{x}_2 - \bar{x}_3) \pm t_{\alpha/2}\sqrt{s^2\left(\frac{1}{n_2} + \frac{1}{n_3}\right)}$$
$$(14 - 13) \pm 2.179\sqrt{5.9333\left(\frac{1}{5} + \frac{1}{5}\right)}$$
$$1 \pm 3.36$$

To reproduce this in R, we can just apply a one-sample t-test to the no breakfast category and a two-sample t-test for light versus full breakfast. Both of these functions contain a 95% confidence interval that is slightly different from those shown in the book. We see that the two-sample confidence interval does not indicate a difference in the average attention spans for students who ate light versus full breakfasts, as the researcher suspected.

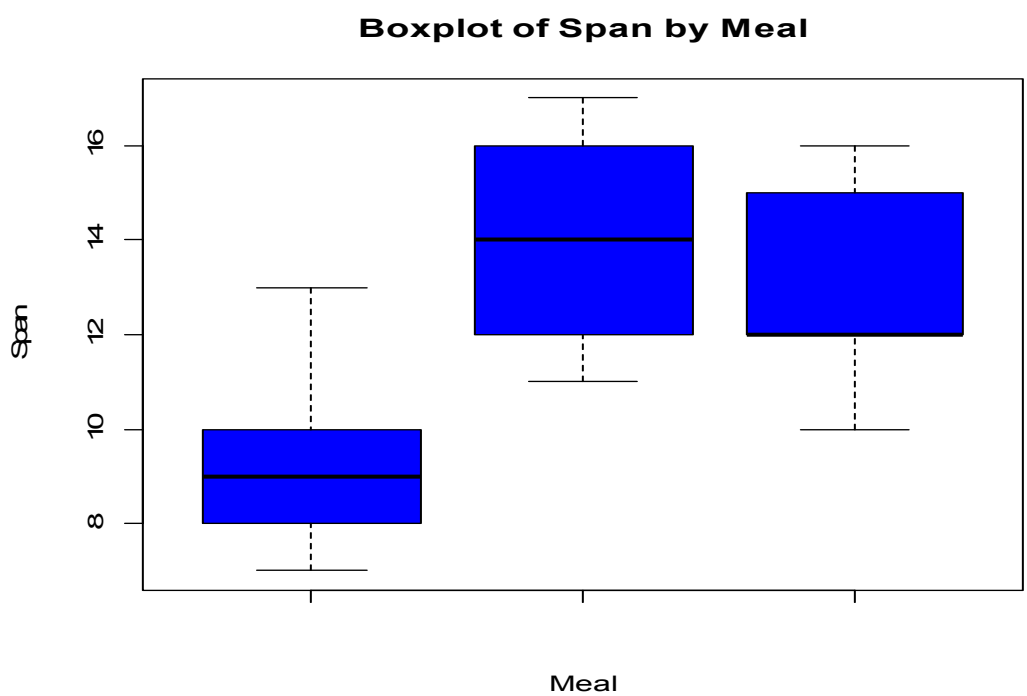
```
t.test(meal1)

        One Sample t-test

data:  meal1

t = 9.1301, df = 4, p-value = 0.0007985

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

  6.541475 12.258525

sample estimates:

mean of x :       9.4
```

```
t.test(meal2,meal3,var.equal=TRUE)

        Two Sample t-test

data:  meal2 and meal3

t = 0.6325, df = 8, p-value = 0.5447

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

 -2.646113  4.646113

sample estimates:

mean of x mean of y

       14        13
```

For a reproduction of Figure 11.7 in R:

```
boxplot(meal1,meal2,meal3,main="Boxplot of Span by Meal",xlab="Meal",

                 ylab="Span",col="blue")
```

Figure 11.7: Box plots for Example 11.6



11.6 RANKING POPULATION MEANS (p. 462 in text)

A simple way to avoid the high risk of declaring differences when they do not exist is to use the **studentized range,** the difference between the smallest and the largest in a set of k sample means, as the yardstick for determining whether there is a difference in a pair of population means. This method, often called **Tukey's method for paired comparisons**, makes the probability of declaring that a difference exists between at least one pair in a set of k treatment means, when no difference exists, equal to α.

Yardstick for Making Paired Comparisons

$$\omega = q_\alpha(k, df)\left(\frac{s}{\sqrt{n_t}}\right)$$

where

k = Number of treatments

s^2 = MSE = Estimator of the common variance σ^2 and $s = \sqrt{s^2}$

df = Number of degrees of freedom for s^2

n_t = Common sample size – that is, the number of observations in each of the k treatment means

$q_\alpha(k, df)$ = Tabulated value from Tables 11(a) and 11(b) in Appendix I, for α = .05 and .01, respectively, and for various combinations of k and df

Example 11.7

Refer to Example 11.4, in which you compared the average attention spans for students given three different "meal" treatments in the morning: no breakfast, a light breakfast, or a full breakfast. The ANOVA F-test in Example 11.5 indicated a significant difference in the population means. Use Tukey's method for paired comparisons to determine which of the three population means differ from the others.

Solution For this example, there are k = 3 treatment means, with $s = \sqrt{MSE} = 2.436$. Tukey's method can be used, with each of the three samples containing $n_t = 5$ measurements and (n – k) = 12 degrees of freedom. The calculated "yardstick" is

$$\omega = q_{.05}(3, 12)\left(\frac{s}{\sqrt{n_t}}\right) = 3.77\left(\frac{2.436}{\sqrt{5}}\right) = 4.11$$

which is greater than $q_{.05}(k, df) = q_{.05}(3, 12) = 3.77$. We can perform this analysis in R using two different methods, `TukeyHSD` or the `glht` function from the `multcomp` package. We will designate the output from the `multcomp` package as modified output for Figure 11.8 from the text and the `TukeyHSD` output as Figure 11.9.

```
library(multcomp)
amod<-aov(values~ind, data=spans)
summary(glht(amod, linfct = mcp(ind = "Tukey")))
         Simultaneous Tests for General Linear Hypotheses
Multiple Comparisons of Means: Tukey Contrasts
Fit: aov(formula = values ~ ind, data = spans)
Linear Hypotheses:    Estimate     Std. Error     t value     Pr(>|t|)
 meal2 - meal1 == 0     4.600        1.541         2.986       0.0287 *
 meal3 - meal1 == 0     3.600        1.541         2.337       0.0887 .
 meal3 - meal2 == 0    -1.000        1.541        -0.649       0.7964
  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
  (Adjusted p values reported -- single-step method)
ci.glht<- confint(tuk)
plot(ci.glht)
```

Figure 11.8: Ranked means for Example 11.7

```
TukeyHSD(aov(values ~ ind, data=spans))
```

Figure 11.9: R output for Example 11.7

```
  Tukey multiple comparisons of means

    95% family-wise confidence level

  Fit: aov(formula = values ~ ind, data = spans)

$ind

                 diff          lwr            upr         p adj

meal2-meal1      4.6        0.4899889       8.710011     0.0284289

meal3-meal1      3.6       -0.5100111       7.710011     0.0886624

meal3-meal2     -1.0       -5.1100111       3.110011     0.7963670
```

11.8 THE ANALYSIS OF VARIANCE FOR A RANDOMIZED BLOCK DESIGN (p. 467 in text)

The randomized block design identifies two factors: **treatments** and **blocks** – both of which affect the response.

Partitioning the Total Variation in the Experiment

This is partitioned into *three* (rather than two) parts in such a way that

Total SS = SSB + SST + SSE

where

- SSB (sum of squares for blocks) measures the variation among the block means.
- SST (sum of squares for treatments) measures the variation among the treatment means.
- SSE (sum of squares for error) measures the variation of the differences among the treatment observations *within* blocks, which measures the experimental error.

Calculating the Sums of Squares for a Randomized Block Design, *k* Treatments in *b* Blocks

$$CM = \frac{G^2}{n}$$

where

$$G = \sum x_{ij} = Total\ of\ all\ n = bk\ observations$$

$$Total\ SS = \sum x_{ij}^2 - CM$$

$$= (Sum\ of\ squares\ of\ all\ x - values) - CM$$

$$SST = \sum \frac{T_i^2}{b} - CM$$

$$SSB = \sum \frac{B_j^2}{k} - CM$$

$$SSE = Total\ SS - SST - SSB$$

where

$$T_i = Total\ of\ all\ observations\ receiving\ treatment\ i, i = 1,2, \dots, k$$

$$B_j = Total\ of\ all\ observations\ in\ block\ j, j = 1,2, \dots, b$$

ANOVA Table for a Randomized Block Design, *k* Treatments and *b* Blocks

Source	***df***	**SS**	**MS**	***F***
Treatments	$k - 1$	SST	MST = SST/(k – 1)	MST/MSE
Blocks	$b - 1$	SSB	MSB = SSB/(b – 1)	MSB/MSE
Error	$(b - 1)(k - 1)$	SSE	MSE = SSE/(n – k)	
Total	$n - 1 = bk - 1$	Total SS		

Example 11.8

The cellular phone industry is involved in a fierce battle for customers, with each company devising its own complex pricing plan to lure customers. Since the cost of a cell phone minute varies drastically depending on the number of minutes per month used by the customer, a consumer watchdog group decided to compare the average costs for four cellular phone companies using three different usage levels as blocks. The monthly costs (in dollars) computed by the cell phone companies for peak-time callers at low (20 minutes per month), middle (150 minutes per month), and high (1000 minutes per month) usage levels are given in Table 11.3. Construct the analysis of variance table for this experiment.

Table 11.3: Monthly Phone Costs of Four Companies at Three Usage Levels

	Company				
Usage Level	**A**	**B**	**C**	**D**	**Totals**
Low	27	24	31	23	$B_1 = 105$
Middle	68	76	65	67	$B_2 = 276$
High	308	326	312	300	$B_3 = 1246$
Totals	$T_1 = 403$	$T_2 = 426$	$T_3 = 408$	$T_4 = 390$	$G = 1627$

Solution The experiment is designed as a randomized block design with b = 3 usage levels (blocks) and k = 4 companies (treatments), so there are n = bk = 12 observations and G = 1627. Then

$$CM = \frac{G^2}{n} = \frac{1627^2}{12} = 220{,}594.0833$$

$$Total\ SS = (27^2 + 24^2 + \cdots + 300^2) - CM = 189{,}798.9167$$

$$SST = \frac{403^2 + \cdots + 390^2}{3} - CM = 222.25$$

$$SSB = \frac{105^2 + 276^2 + 1246^2}{4} - CM = 189{,}335.1667$$

and by subtraction,

$$SSE = Total\ SS - SST - SSB = 241.5$$

These four sources of variation, degrees of freedom, sums of squares, and mean squares are shown in the analysis of variance table, generated by R and given in Figure 11.10.

```
cost=c(27,24,31,23,68,76,65,67,308,326,312,300)
usage=c(rep("Low",4),rep("Middle",4),rep("High",4))
company=c(rep(c("A","B","C","D"),3))
phone=data.frame(cost,usage,company)
lm.phone <- lm(cost ~ usage + company, data=phone)
anova(lm.phone)
```

Figure 11.10: R output for Example 11.8

```
Analysis of Variance Table
Response: cost
           Df        Sum Sq      Mean Sq     F value      Pr(>F)
usage      2         189335       94668     2351.9896    2.067e-09  ***
company    3            222          74        1.8406    0.2404
Residuals  6            242          40
    Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We also use R as before to obtain a summary:

```
summary(lm.phone)
   Call:     lm(formula = cost ~ usage + company, data = phone)
   Residuals:     Min        1Q       Median        3Q          Max
                -8.6667   -2.7917     0.4167      2.6458       8.0833
   Coefficients:  Estimate     Std. Error     t value      Pr(>|t|)
    (Intercept)   310.250       4.486         69.158      6.15e-10  ***
     usageLow    -285.250       4.486        -63.585      1.02e-09  ***
    usageMiddle  -242.500       4.486        -54.056      2.69e-09  ***
      companyB      7.667       5.180          1.480      0.189
      companyC      1.667       5.180          0.322      0.759
      companyD     -4.333       5.180         -0.837      0.435
       Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
            Residual standard error: 6.344 on 6 degrees of freedom
            Multiple R-squared: 0.9987,     Adjusted R-squared: 0.9977
            F-statistic: 941.9 on 5 and 6 DF,  p-value: 1.35e-08
```

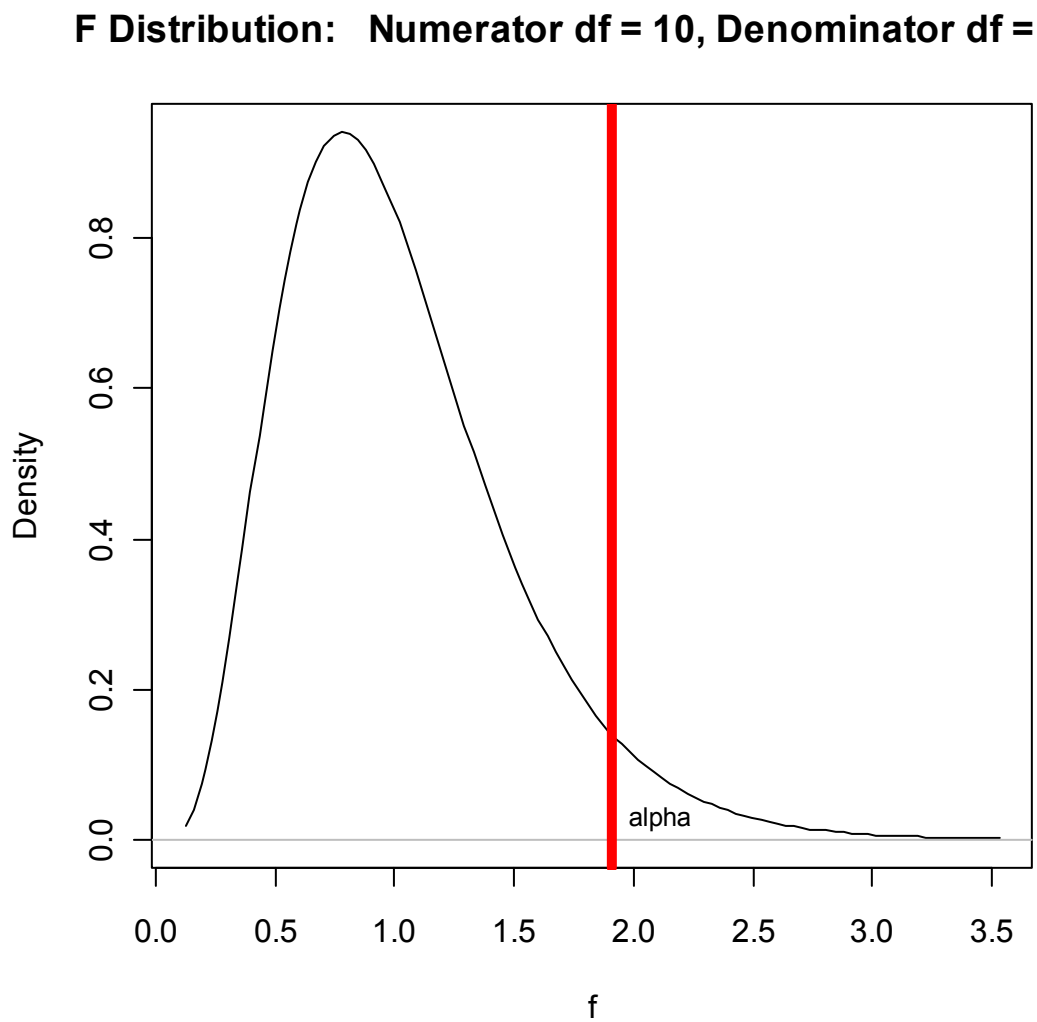
Tests for a Randomized Block Design

For comparing treatment means:

1. Null hypothesis: H_0: The treatment means are equal
2. Alternative hypothesis: H_a: At least two of the treatment means differ
3. Test statistic: F = MST/MSE, where F is based on $df_1 = (k - 1)$ and $df_2 = (b - 1)(k - 1)$
4. Rejection region: Reject if $F > F_\alpha$, where F_α lies in the upper tail of the F distribution (see the figure), or when the p-value $< \alpha$

For comparing block means:

1. Null hypothesis: H_0: The block means are equal
2. Alternative hypothesis: H_a: At least two of the block means differ
3. Test statistic: F = MSB/MSE, where F is based on $df_1 = (b - 1)$ and $df_2 = (b - 1)(k - 1)$
4. Rejection region: Reject if $F > F_\alpha$, where F_α lies in the upper tail of the F distribution (see the figure), or when the p-value $< \alpha$




```
.x <- seq(0.122, 3.533, length.out=100)
plot(.x, df(.x, df1=10, df2=100), xlab="f", ylab="Density",
         main="F Distribution:   Numerator df = 10, Denominator df = 100",
type="l")
abline(h=0, col="gray")
remove(.x)
abline(v=1.91, col="red",lwd=5)
text(locator(1),"alpha",cex=.8,lwd=2)
```

Example 11.9

Do the data in Example 11.8 provide sufficient evidence to indicate a difference in the average monthly cell phone cost depending on the company the customer uses?

Solution The cell phone companies represent the *treatments* in this randomized block design, and the differences in their average monthly costs are of primary interest to the researcher. To test

$$H_0: No\ difference\ in\ the\ average\ cost\ among\ companies$$

versus the alternative that the average cost is different for at least one of the four companies, you use the analysis of variance F statistic, calculated as

$$F = \frac{MST}{MSE} = \frac{74.1}{40.3} = 1.84$$

and shown in the column marked `F value` and the row marked `company` in Figure 11.10. The exact p-value is found in the column marked `Pr(>F)` and the row marked `company` in Figure 11.10 as `0.2404`, which is too large to allow rejection of H_0. The results do not show a significant difference in the treatment means. That is, there is insufficient evidence to indicate a difference in the average monthly costs for the four companies.

Comparing Treatment and Block Means

Tukey's yardstick for comparing block means:

$$\omega = q_\alpha(b, df)\left(\frac{s}{\sqrt{k}}\right)$$

Tukey's yardstick for comparing treatment means:

$$\omega = q_\alpha(b, df)\left(\frac{s}{\sqrt{b}}\right)$$

(1 – α)100% confidence interval for the difference in two block means:

$$\left(\bar{B}_i - \bar{B}_j\right) \pm t_{\alpha/2}\sqrt{s^2\left(\frac{1}{k} + \frac{1}{k}\right)}$$

where $\bar{B}_i$ is the average of all observations in block i

(1 – α)100% confidence interval for the difference in two treatment means:

$$\left(\bar{T}_i - \bar{T}_j\right) \pm t_{\alpha/2}\sqrt{s^2\left(\frac{1}{b} + \frac{1}{b}\right)}$$

where $\bar{T}_i$ is the average of all observations in treatment i.

Example 11.10

Identify the nature of any differences you found in the average monthly cell phone costs from Example 11.8.

Solution Since the F-test did not show any significant differences in the average costs for the four companies, there is no reason to use Tukey's method of paired comparisons. Suppose, however, that you are an executive for company B and your major competitor is company C. Can you claim a significant difference in the two average costs? Using a 95% confidence interval, you can calculate

$$(\bar{T}_2 - \bar{T}_3) \pm t_{.025}\sqrt{MSE\left(\frac{2}{b}\right)}$$

$$\left(\frac{426}{3} - \frac{408}{3}\right) \pm 2.447\sqrt{40.3\left(\frac{2}{3}\right)}$$

$$6 \pm 12.68$$

so the difference between the two average costs is estimated as between -$6.68 and $18.68. Since 0 is contained in the interval, you do not have evidence to indicate a significant difference in your average costs. These values can be obtained in R using the following code:

```
B1=mean(cost[company=="B"])
B2=mean(cost[company=="C"])
diff=B1-B2
alpha=.05
df=anova(lm.phone)['Residuals','Df']
t.star=qt(1-alpha/2,df)
s2=anova(lm.phone)['Residuals','Mean Sq']
b=anova(lm.phone)['company','Df']
tukey.int=c(diff-t.star*sqrt(s2*(2/b)),diff+t.star*sqrt(s2*(2/b)))
tukey.int
       [1] -6.675224 18.675224
```

11.9 THE *a* x *b* FACTORIAL EXPERIMENT: A TWO-WAY CLASSIFICATION (p. 478 in text)

Suppose the manager of a manufacturing plant suspects that the output (in number of units produced per shift) of a production line depends on two factors:

- Which of two supervisors is in charge of the line
- Which of three shifts – day, swing, or night – is being measured

You need to investigate not only the average output for the two supervisors and the average output for the three shifts, but also the **interaction** or relationship between the two factors.

Example 11.11

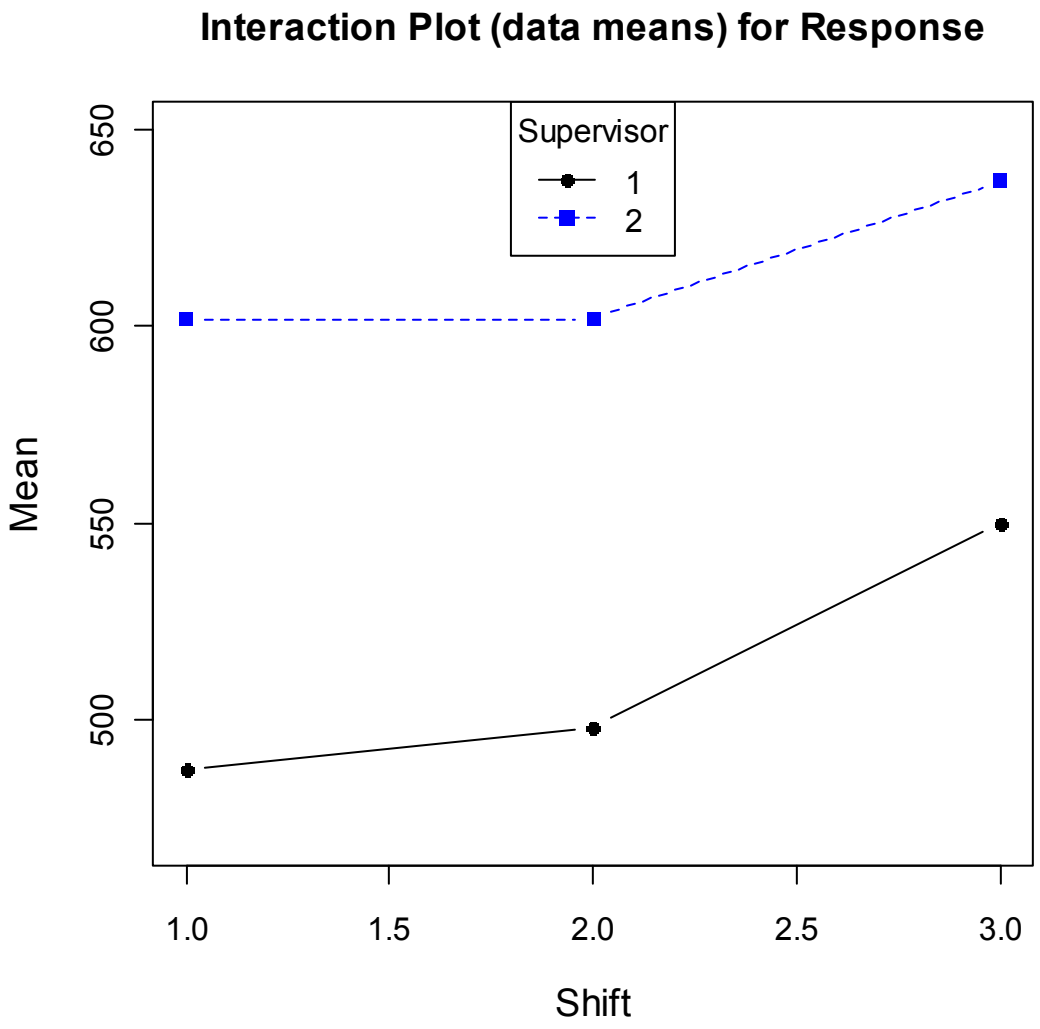
Suppose that the two supervisors are each observed on three randomly selected days for each of the three different shifts. The average outputs for the three shifts are shown in Table 11.4 for each of the supervisors. Look at the relationship between the two factors in the line chart for these means, shown in Figure 11.11. Notice that supervisor 2 always produces a higher output, regardless of the shift. The two factors behave *independently;* that is, the output is always about 100 units higher for supervisor 2, no matter which shift you look at.

Table 11.4: Average Outputs for Two Supervisors on Three Shifts

	Shift		
Supervisor	**Day**	**Swing**	**Night**
1	487	498	550
2	602	602	637

```
output1=c(487,498,550,602,602,637)
supervisor1=c(rep(1,3),rep(2,3))
shift1=(c(rep(c(1,2,3),2)))
data1=data.frame(output,supervisor,shift)
plot(shift[supervisor==1],output[supervisor==1],type="b",ylim=c(470,650),
     pch=16,xlab="Shift",ylab="Mean",
     main="Interaction Plot (data means) for Response",cex.lab=1.2,
     bg="lightblue")
points(shift[supervisor==2],output[supervisor==2],type="b", col="blue",
       pch=15,lty=2)
legend("top",title="Supervisor", legend=c(1,2),lty=1:2,
       col=c("black","blue"),pch=16:15)
```

Figure 11.11: Interaction plot for means in Table 11.4



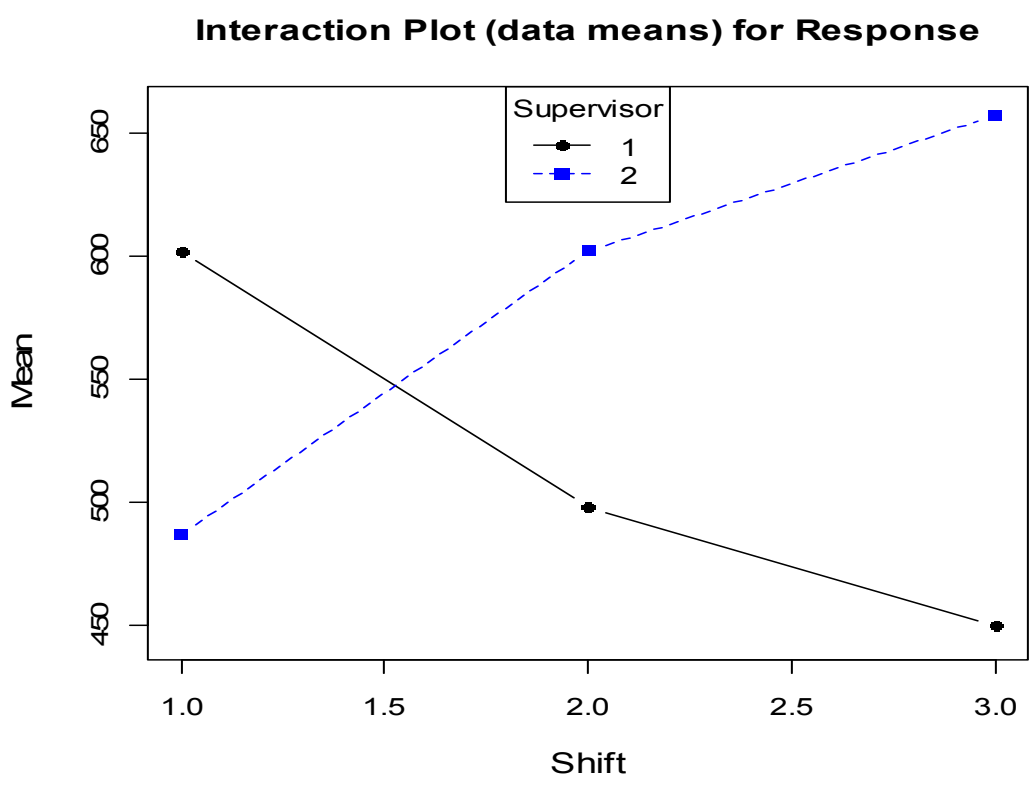
Now consider another set of data for the same situation, shown in Table 11.5. There is a definite difference in the results, depending on which shift you look at, and the *interaction* can be seen in the crossed lines of the chart in Figure 11.12.

Table 11.5: Average Outputs for Two Supervisors on Three Shifts

	Shift		
Supervisor	**Day**	**Swing**	**Night**
1	602	498	450
2	487	602	657

```
output2=c(602,498,450,487,602,657)
supervisor2=c(rep(1,3),rep(2,3))
shift2=(c(rep(c(1,2,3),2)))
data2=data.frame(output,supervisor,shift)
plot(shift2[supervisor2==1],output2[supervisor2==1],type="b",
     ylim=c(445,660), pch=16,xlab="Shift",ylab="Mean",
     main="Interaction Plot (data means) for Response",cex.lab=1.2,
     bg="lightblue")
points(shift2[supervisor2==2],output2[supervisor2==2],type="b",col="blue",
       pch=15,lty=2)
legend("top",title="Supervisor",legend=c(1,2),lty=1:2,col=c("black","blue"),
       pch=16:15)
```

Figure 11.12: Interaction plot for means in Table 11.5



11.10 THE ANALYSIS OF VARIANCE FOR AN *a* x *b* FACTORIAL EXPERIMENT (p. 480 in text)

An analysis of variance for a two-factor factorial experiment replicated *r* times follows the same pattern as the previous designs and is partitioned into *four* parts:

$$Total\ SS = SSA + SSB + SS(AB) + SSE$$

ANOVA Table for *r* Replications of a Two-Factor Experiment: Factor A at *a* Levels and Factor B at *b* Levels

Source	***df***	**SS**	**MS**	***F***
A	$a - 1$	SSA	MSA = SSA/$(a - 1)$	MSA/MSE
B	$b - 1$	SSB	MSB = SSB/$(b - 1)$	MSB/MSE
AB	$(a - 1)(b - 1)$	SS(AB)	MS(AB) = SS(AB)/$(a - 1)(b - 1)$	MS(AB)/MSE
Error	$ab(r - 1)$	SSE	MSE = SSE/ $ab(r - 1)$	
Total	$abr - 1$	Total SS		

Tests for a Factorial Experiment

- **For interaction:**

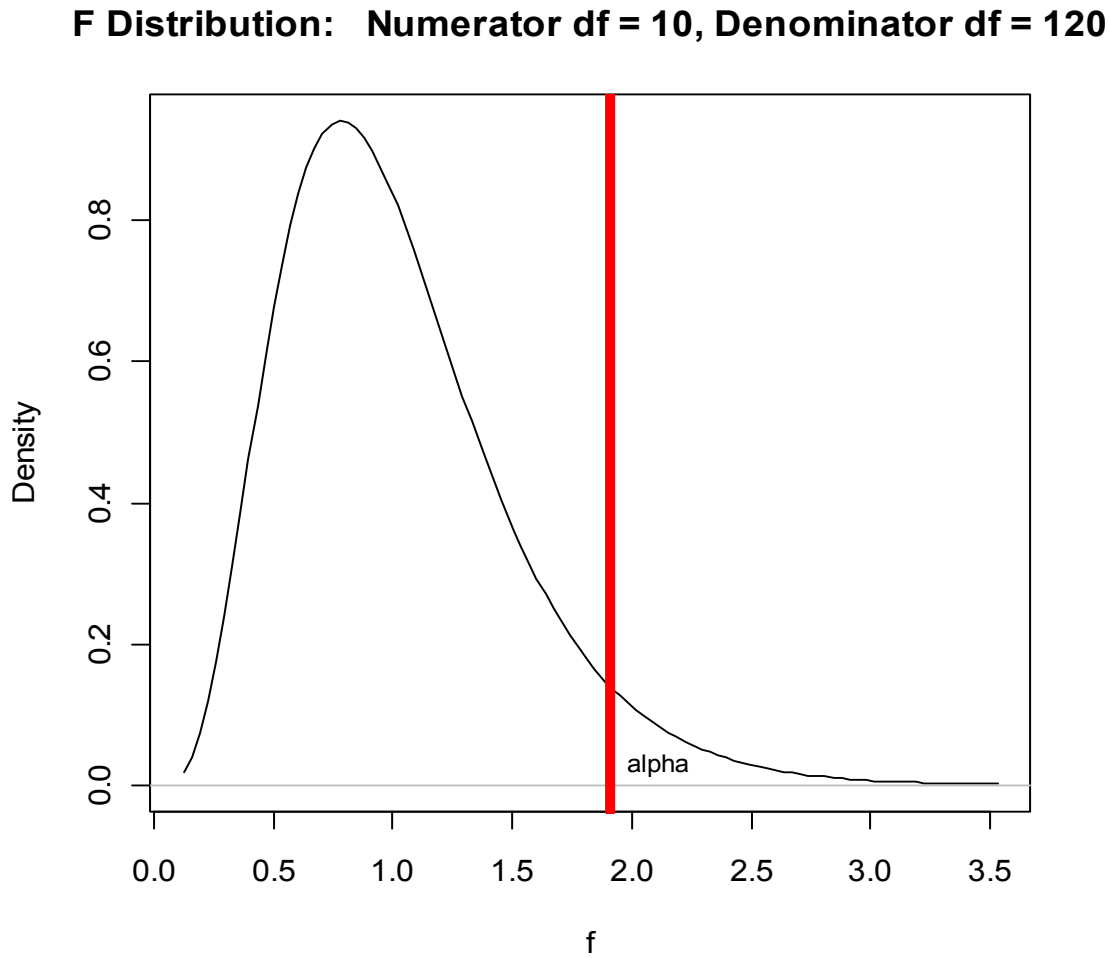
1. Null hypothesis: H_0: Factors A and B do not interact
2. Alternative hypothesis: H_a: Factors A and B interact
3. Test statistic: F = MS(AB)/MSE, where F is based on $df_1 = (a - 1)(b - 1)$ and $df_2 = ab(r - 1)$
4. Rejection region: Reject H_0 when $F > F_\alpha$, where F_α lies in the upper tail of the F distribution (see the figure), or when the p-value $< \alpha$

- **For main effects, factor A:**

1. Null hypothesis: H_0: There are no differences among the factor A means
2. Alternative hypothesis: H_a: At least two of the factor a means differ
3. Test statistic: F = MSA/MSE, where F is based on $df_1 = (a - 1)$ and $df_2 = ab(r - 1)$
4. Rejection region: Reject H_0 when $F > F_\alpha$ (see the figure) or when the p-value $< \alpha$

- **For main effects, factor B:**

1. Null hypothesis: H_0: There are no differences among the factor B means
2. Alternative hypothesis: H_a: At least two of the factor B means differ
3. Test statistic: F = MSB/MSE, where F is based on $df_1 = (b - 1)$ and $df_2 = ab(r - 1)$
4. Rejection region: Reject H_0 when $F > F_\alpha$ (see the figure) or when the p-value $< \alpha$




```
.x <- seq(0.122, 3.533, length.out=100)
plot(.x, df(.x, df1=10, df2=100), xlab="f", ylab="Density",
     main="F Distribution: Numerator df = 10, Denominator df = 100",
     type="l")
abline(h=0, col="gray")
remove(.x)
abline(v=1.91, col="red",lwd=5)
text(locator(1),"alpha",cex=.8,lwd=2)
```

Example 11.12

Table 11.6 shows the original data used to generate Table 11.5 in Example 11.11. That is, the two supervisors were each observed on three randomly selected days for each of the three different shifts, and the production outputs were recorded. Analyze these data using the appropriate analysis of variance procedure.

Table 11.6: Outputs for Two Supervisors on Three Shifts

	Shift		
Supervisor	**Day**	**Swing**	**Night**
1	571	480	470
	610	474	430
	625	540	450
2	480	625	630
	516	600	680
	465	581	661

Solution The R output is shown in Figure 11.13.

```
output=c(571,480,470,610,474,430,625,540,450,480,625,630,516,600,680,465,
         581,66)
supervisor=as.factor(c(rep(1,9),rep(2,9)))
shift=as.factor(c(rep(c("Day","Swing","Night"),6)))
data=data.frame(output,supervisor,shift)
lm.data <- lm(output~supervisor+shift+supervisor:shift,data=data)
summary(lm.data)
```

Figure 11.13: R summary output for Example 11.12

```
Call:lm(formula = output ~ supervisor+shift+supervisor:shift,data = data)
Residuals:   Min       1Q       Median       3Q         Max
           -31.00   -20.75     -1.00       22.25      42.00
Coefficients:           Estimate  Std. Error t value     Pr(>|t|)
  (Intercept)             602.00     15.49    38.859    5.44e-14 ***
  supervisor2            -115.00     21.91    -5.249    0.000205 ***
   shiftNight            -152.00     21.91    -6.938    1.57e-05 ***
   shiftSwing            -104.00     21.91    -4.747    0.000475 ***
supervisor2:shiftNight    322.00     30.98    10.393    2.36e-07 ***
supervisor2:shiftSwing    219.00     30.98     7.068    1.30e-05 ***
      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
               Residual standard error: 26.83 on 12 degrees of freedom
               Multiple R-squared: 0.9209,     Adjusted R-squared: 0.8879
               F-statistic: 27.94 on 5 and 12 DF,  p-value: 3.234e-06
```

```
anova(lm.data)
```

Figure 11.13: R ANOVA output for Example 11.12

```
Analysis of Variance Table

Response: output

                    Df    Sum Sq    Mean Sq   F value     Pr(>F)

supervisor           1     19208     19208    26.6778    0.0002351 ***

   shift             2       247       123     0.1715    0.8444061

supervisor:shift     2     81127     40564    56.3382    7.95e-07  ***

   Residuals        12      8640       720

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

REPEATED MEASURES (Supplements)

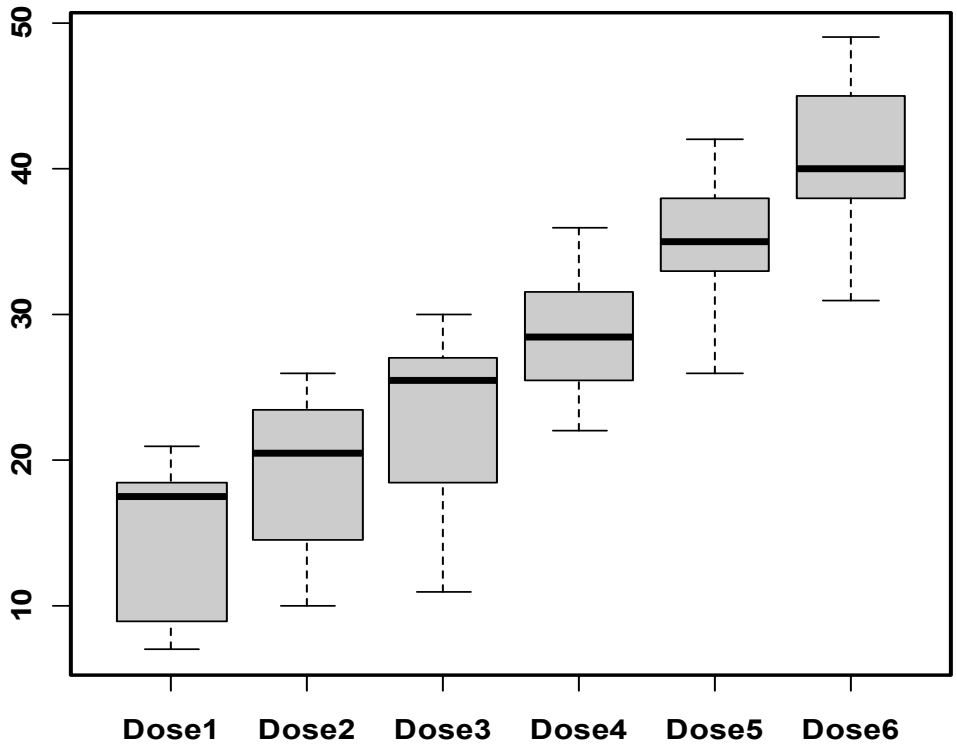
Longitudinal data are defined as data resulting from the observations of subjects (human beings, animals, laboratory samples, etc.), which are measured repeatedly over time. The purpose of conducting a longitudinal study is to look at change across the time span. When change itself is the object of study, the only way to investigate it is by collecting repeated measurements. For example, in medicine, patients may be assigned to different treatments at the start of a study so the investigators can determine at intervals (by week or by year, for example) any effects of the treatments assigned. The advantage of longitudinal study is the information that emerges about individual change. That is, by collecting data longitudinally, changes over time that may occur for an individual sample can be separated from differences between individuals at baseline. Thus, longitudinal studies give tremendous information about their subjects. For such data, mixed-effects models provide a useful and flexible framework in which population characteristics are modeled as fixed effects, and individual variation is modeled as random effects, and within-subject variations are accounted for by an error process.

Here, we use the rabbit data example in which blood pressure was measured for 12 rabbits at six doses in which the dose increased in an ascending manner. First we input our variables in R as usual. The `rep` command simply repeats a value or variable name as many times as we need.

```
bp=c(21.0,21.0,23.0,35.0,36.0,48.0,19.0,24.0,27.0,36.0,36.0,46.0,12.0,25.0,

     27.0,26.0,33.0,40.0,9.0,17.0,18.0,27.0,34.0,39.0,7.0,10.0,19.0,25.0,

     31.0,38.0,18.0,26.0,26.0,29.0,39.0,44.0,9.0,12.0,17.0,22.0,33.0,40.0,

     20.0,20.0,30.0,30.0,38.0,41.0,18.0,18.0,27.0,31.0,42.0,49.0,8.0,12.0,

     11.0,24.0,26.0,31.0,18.0,22.0,25.0,32.0,38.0,38.0,17.0,23.0,26.0,28.0,

     34.0,35.0)

rabbit=as.factor(c(rep(1,6),rep(2,6),rep(3,6),rep(4,6),rep(5,6),rep(6,6),

                   rep(7,6),rep(8,6),rep(9,6),rep(10,6),rep(11,6),rep(12,6)))

dose=as.factor(c(rep(c(1,2,3,4,5,6),12)))

set=data.frame(bp,rabbit,dose)
```

Here, we partition blood pressure into six groups corresponding to the six doses in order to compare the doses with boxplots.

```
bp.1=set$bp[dose==1]
bp.2=set$bp[dose==2]
bp.3=set$bp[dose==3]
bp.4=set$bp[dose==4]
bp.5=set$bp[dose==5]
bp.6=set$bp[dose==6]
bp.level=data.frame(bp.1,bp.2,bp.3,bp.4,bp.5,bp.6)
par(mfrow=c(1,1),font=2,font.lab=2,font.axis=2)
boxplot(bp.1,bp.2,bp.3,bp.4,bp.5,bp.6,col=gray(0.8),
        names=c("Dose1","Dose2","Dose3","Dose4","Dose5","Dose6"),
        pch=16,lwd=1.5)
```



```
bp.level

   bp.1 bp.2 bp.3 bp.4 bp.5 bp.6

1    21   21   23   35   36   48

2    19   24   27   36   36   46

3    12   25   27   26   33   40

4     9   17   18   27   34   39

5     7   10   19   25   31   38

6    18   26   26   29   39   44

7     9   12   17   22   33   40

8    20   20   30   30   38   41

9    18   18   27   31   42   49

10    8   12   11   24   26   31

11   18   22   25   32   38   38

12   17   23   26   28   34   35

bp.unlist=unlist(bp.level)
```

With repeated measures, it is often the case that we need to restructure the data into long form. Here is the manual way to do this with our data:

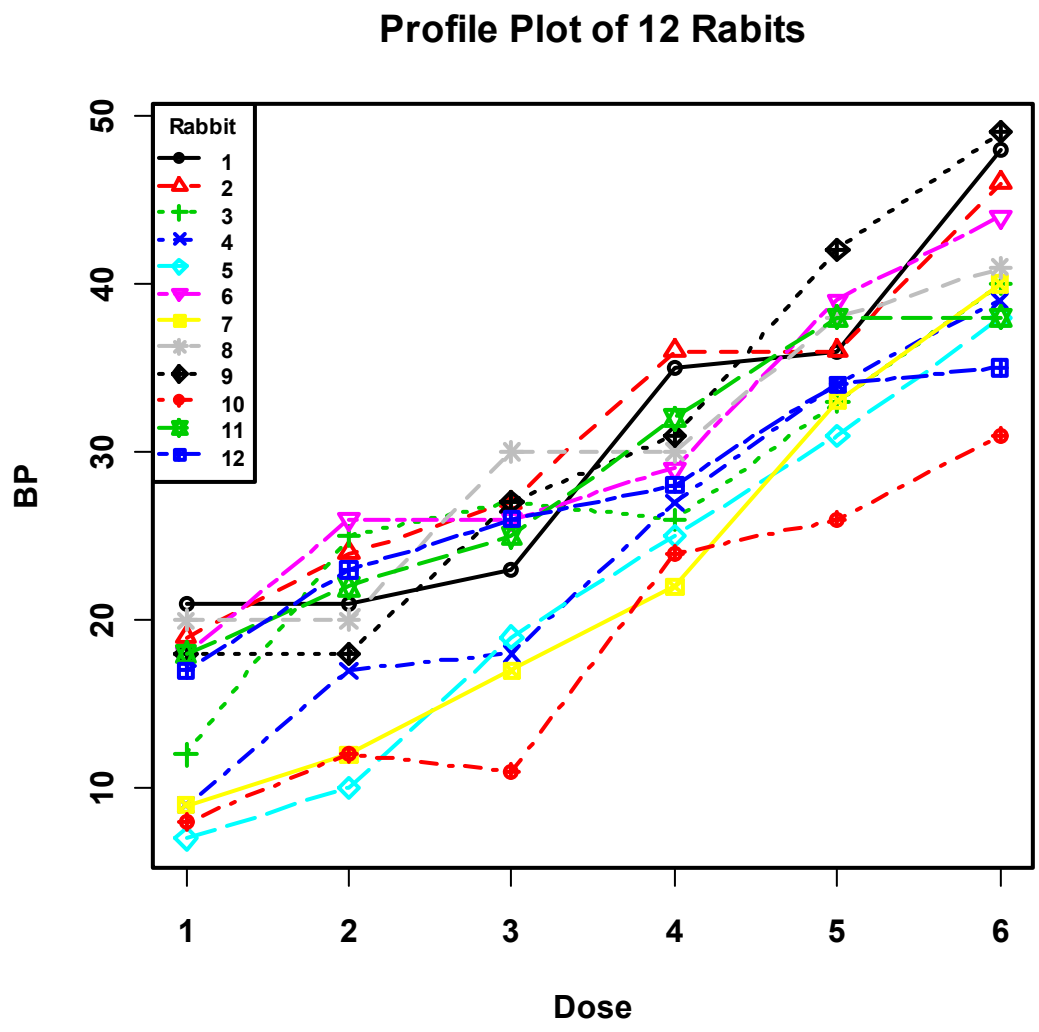
```
rabbit=rep(1:nrow(bp.level),6)

dose=rep(1:6,rep(nrow(bp.level),6))

bp.long<-data.frame(rabbit,dose,bp)

bp.long

   rabbit dose bp

1       1    1 21

2       2    1 21

3       3    1 23

……————————————

72     12    6 35
```

Or, we can obtain the same result by using the `direction= "long"` command in the `reshape` function.

```
bp.long1<-reshape(bp.level,idvar="rabbit",
          varying=c("bp.1","bp.2","bp.3","bp.4","bp.5","bp.6"),
          direction="long")
bp.long1
   rabbit dose bp
1       1    1 21
2       2    1 21
3       3    1 23
4       4    1 35
5       5    1 36
.............................
72     12    6 35
```

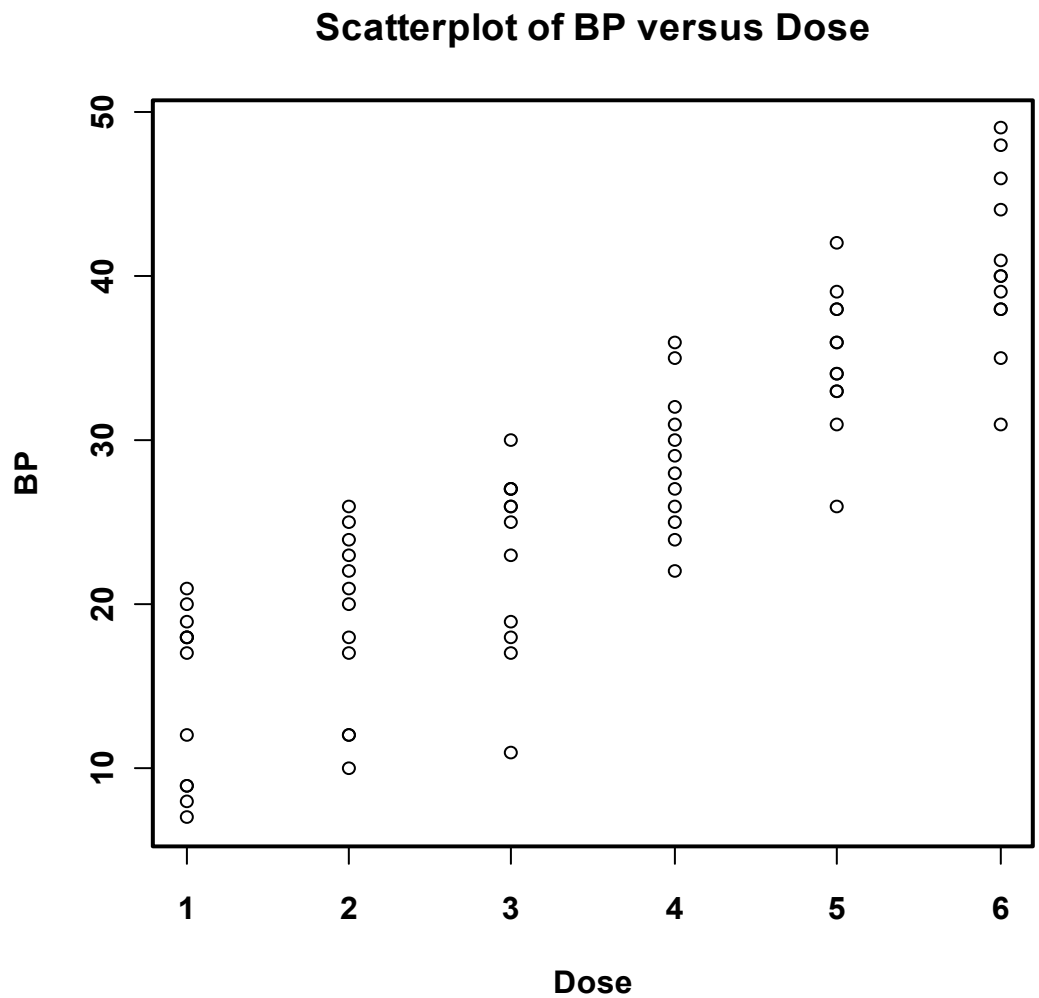
A profile plot is obtained by the following code:

```
par(mfrow=c(1,1), font=2,font.axis=2,font.lab=2)
y<-1:6
matplot(y, t(bp.level), type = "o", pch=1:12,
        main = "Profile Plot of 12 Rabits", xlab = "Dose",
        ylab="BP",col=1:12,lty=1:12,lwd=2)
legend("topleft",title="Rabbit",legend=1:12,lty=1:12,
       col=1:12,pch=1:12,cex=.7)
```



A scatterplot of blood pressure against dose reveals the repeated nature of the data:

```
x<-c(rep(1,12),rep(2,12),rep(3,12),rep(4,12),rep(5,12),rep(6,12))

y<-c(bp.1,bp.2,bp.3,bp.4,bp.5,bp.6)

frame<-data.frame(x,y)

par(font=2,font.axis=2,font.lab=2, lwd=2)

matplot(x,y,xlab="Dose",ylab="BP",pch=1,main="Scatterplot of BP versus Dose")
```



Next, we obtain a correlation and covariance matrix for blood pressure:

```
cor(bp.level)
```

	bp.1	bp.2	bp.3	bp.4	bp.5	bp.6
bp.1	1.0000000	0.7362178	0.7944226	0.8556488	0.7658162	0.6195511
bp.2	0.7362178	1.0000000	0.7777343	0.6155840	0.5575903	0.3859129
bp.3	0.7944226	0.7777343	1.0000000	0.6177764	0.7866943	0.5609202
bp.4	0.8556488	0.6155840	0.6177764	1.0000000	0.6241806	0.6639864
bp.5	0.7658162	0.5575903	0.7866943	0.6241806	1.0000000	0.7629271
bp.6	0.6195511	0.3859129	0.5609202	0.6639864	0.7629271	1.0000000

```
cov(bp.level)
```

	bp.1	bp.2	bp.3	bp.4	bp.5	bp.6
bp.1	27.33333	20.87879	23.09091	19.18182	16.72727	17.09091
bp.2	20.87879	29.42424	23.45455	14.31818	12.63636	11.04545
bp.3	23.09091	23.45455	30.90909	14.72727	18.27273	16.45455
bp.4	19.18182	14.31818	14.72727	18.38636	11.18182	15.02273
bp.5	16.72727	12.63636	18.27273	11.18182	17.45455	16.81818
bp.6	17.09091	11.04545	16.45455	15.02273	16.81818	27.84091

For an ANOVA and for differences of least squares means, we can run just a linear model:

```
lm.rabbits<-lm(bp~dose,data=set)

anova(lm.rabbits)

Analysis of Variance Table

Response: bp

            Df        Sum Sq       Mean Sq      F value       Pr(>F)

 dose        5         5826.3       1165.26      46.195      < 2.2e-16 ***

 Residuals  66         1664.8         25.22

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

```
summary(lm.rabbits)

Call:  lm(formula = bp ~ dose, data = set)

Residuals:      Min       1Q       Median      3Q         Max

            -12.0000  -3.0000     0.5417     3.4583     8.2500

Coefficients:   Estimate     Std. Error     t value     Pr(>|t|)

  (Intercept)    14.667        1.450        10.116      4.74e-15  ***

      dose2       4.500        2.050         2.195      0.031709

      dose3       8.333        2.050         4.064      0.000131  ***

      dose4      14.083        2.050         6.869      2.77e-09  ***

      dose5      20.333        2.050         9.917      1.05e-14  ***

      dose6      26.083        2.050        12.721      < 2e-16   ***

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

       Residual standard error: 5.022 on 66 degrees of freedom

       Multiple R-squared: 0.7778,     Adjusted R-squared: 0.7609

       F-statistic: 46.19 on 5 and 66 DF,  p-value: < 2.2e-16
```

Here, we fit a random intercept model using the **lme4** package from R:

```
library(lme4)

lmer.rabbits<-lmer(bp~dose+(1|rabbit),data=bp.long)

summary(lmer.rabbits)

Linear mixed model fit by REML

Formula: bp ~ dose + (1 | rabbit)

   Data: bp.long

      AIC      BIC      logLik         deviance        REMLdev

     478.8    487.9     -235.4           474.4          470.8

Random effects:     Groups      Name       Variance     Std.Dev.

                    rabbit   (Intercept)    84.570      9.1962

                    Residual                26.528      5.1506

Number of obs: 72, groups: rabbit, 12

Fixed effects:     Estimate      Std. Error     t value

      Intercept)   28.6389         2.9937        9.566

          dose     -0.5000         0.3554       -1.407

Correlation of Fixed Effects:

           (Intr)

    dose   -0.416
```