Intermediate Analytics

Fatemeh Ahmadi

ALY 6015

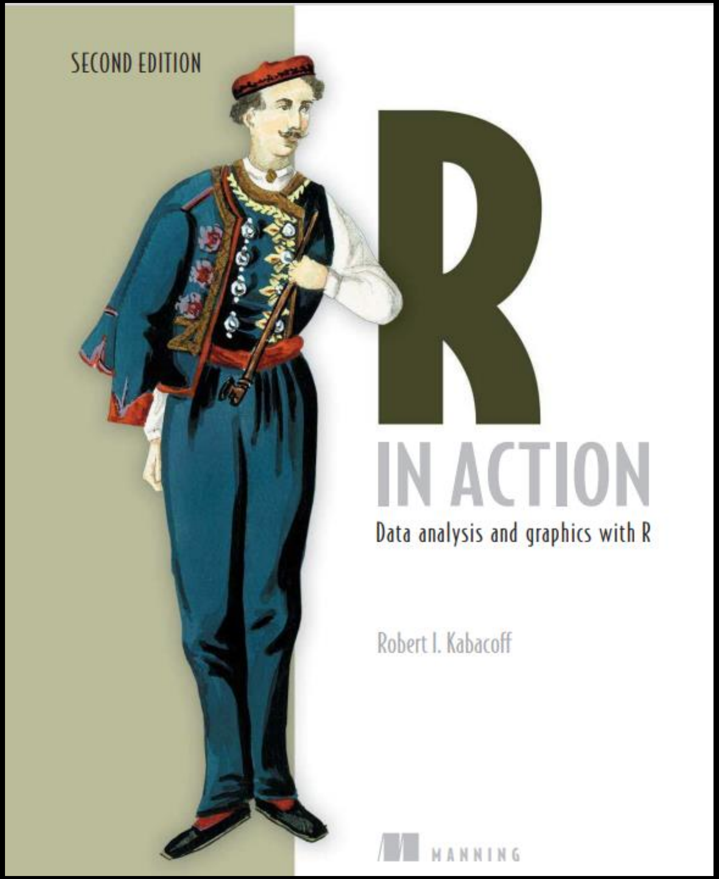
Analysis of Variance

Slides are mainly borrowed from the textbook:

- ***R in Action, 2nd Edition, R. Kabacoff, Manning***

You will learn in this course:

- Using R to model basic experimental designs
- Fitting and interpreting ANOVA type models
- Evaluating model assumptions



Introduction

In chapter 7, we looked at regression models for predicting a quantitative response variable from quantitative predictor variables. But there's no reason that we couldn't have included nominal or ordinal factors as predictors as well.

When factors are included as explanatory variables, our focus usually shifts from prediction to understanding group differences, and the methodology is referred to as **analysis of variance (ANOVA)**. ANOVA methodology is used to analyze a wide variety of experimental and quasi-experimental designs.

This chapter provides an overview of *R* functions for analyzing common research designs. First, we'll look at design terminology, followed by a general discussion of *R*'s approach to fitting ANOVA models. Then we'll explore several examples that illustrate the analysis of common designs. Along the way, you'll treat anxiety disorders, lower blood cholesterol levels, help pregnant mice have fat babies, assure that pigs grow long in the tooth, facilitate breathing in plants, and learn which grocery shelves to avoid.

Introduction

In addition to the base installation, you’ll be using the:

✓car,

✓*gplots*,

✓*HH*,

✓*rrcov*,

✓*multcomp*,

✓*effects*,

✓*MASS*,

✓*mvoutlier*

packages in the examples. Be sure to install them before trying out the sample code.

A Crash Course on Terminology

Experimental design in general, and analysis of variance in particular, have its own language. Before discussing the analysis of these designs, we’ll quickly review some important terms. We’ll use a series of increasingly complex study designs to introduce the most significant concepts.

Say you’re interested in studying the treatment of anxiety. Two popular therapies for anxiety are cognitive behavior therapy (*CBT*) and eye movement desensitization and reprocessing (*EMDR*). You recruit 10 anxious individuals and randomly assign half of them to receive five weeks of *CBT* and half to receive five weeks of *EMDR*. At the conclusion of therapy, each patient is asked to complete the State-Trait Anxiety Inventory (*STAI*), a self-report measure of anxiety. The design is outlined in table 9.1

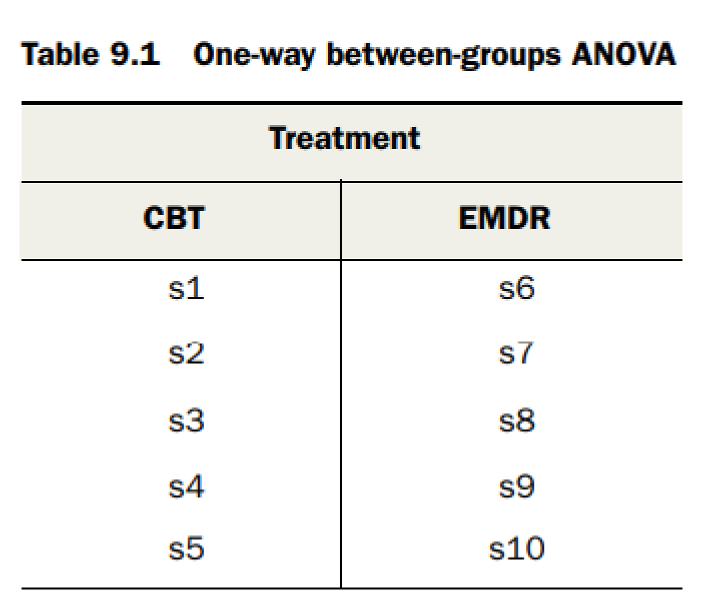


Table 9.1 One-way between-groups ANOVA

Treatment	
CBT	**EMDR**
s1	s6
s2	s7
s3	s8
s4	s9
s5	s10

A Crash Course on Terminology

In this design, *Treatment* is a *between-groups factor* with two levels (*CBT*, *EMDR*). It's called a between-groups factor because patients are assigned to one and only one group. No patient receives both *CBT* and *EMDR*. The *s* characters represent the subjects (patients). *STAI* is the dependent variable, and *Treatment* is the *independent variable*. Because there is an equal number of observations in each treatment condition, you have a *balanced design*. When the sample sizes are unequal across the cells of a design, you have an *unbalanced design*.

The statistical design in table 9.1 is called a one-way ANOVA because there's a single classification variable. Specifically, it's a **one-way between-groups ANOVA**. Effects in ANOVA designs are primarily evaluated through F-tests. If the F-test for *Treatment* is significant, you can conclude that the mean *STAI* scores for the two therapies differed after five weeks of treatment.

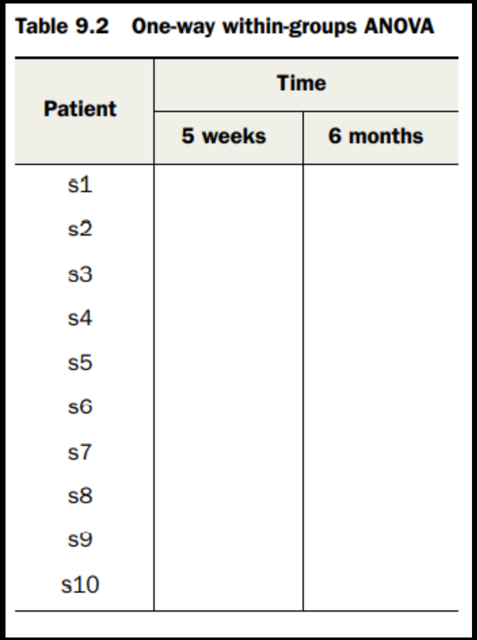


Table 9.2 One-way within-groups ANOVA

Patient	Time	
	5 weeks	6 months
s1		
s2		
s3		
s4		
s5		
s6		
s7		
s8		
s9		
s10		

A Crash Course on Terminology

By including both *Therapy* and *Time* as factors, you’re able to examine the impact of *Therapy* (averaged across *Time*), *Time* (averaged across *Therapy* type), and the interaction of *Therapy* and *Time*. The first two are called the *main effects*, whereas the interaction is (not surprisingly) called an *interaction effect*.

When you cross two or more factors, as is done here, you have a factorial ANOVA design. Crossing two factors produce a two-way ANOVA, crossing three factors produces a three-way ANOVA, and so forth. When a factorial design includes both between-groups and within-groups factors, it’s also called a mixed-model ANOVA. The current design is a two-way mixed-model factorial ANOVA.

Fitting ANOVA Models

Although ANOVA and regression methodologies developed separately, functionally they’re both special cases of the general linear model. You could analyze ANOVA models using the same *lm()* function used for regression in chapter 7.

But you’ll primarily use the *aov*() function in this chapter. The results of *lm*() and *aov*() are equivalent, but the *aov*() function presents these results in a format that’s more familiar to ANOVA methodologists.

The aov() Function

The syntax of the *aov*() function is *aov(formula, data=dataframe)*. Table 9.4 describes special symbols that can be used in the formulas. In this table, *y* is the dependent variable and the letters *A*, *B*, and *C* represent factors.

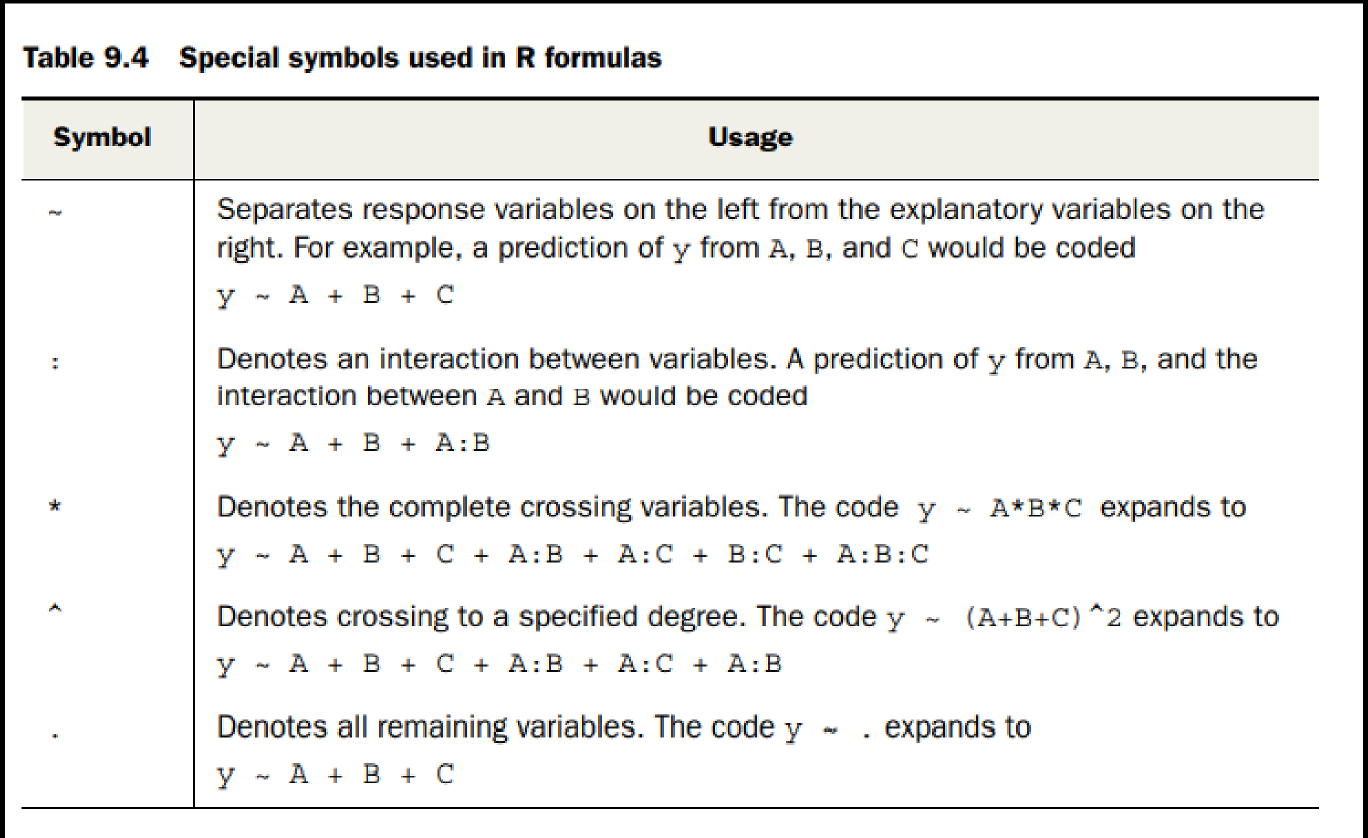


Table 9.4 Special symbols used in R formulas

Symbol	Usage
~	Separates response variables on the left from the explanatory variables on the right. For example, a prediction of `y` from `A`, `B`, and `C` would be coded `y ~ A + B + C`
:	Denotes an interaction between variables. A prediction of `y` from `A`, `B`, and the interaction between `A` and `B` would be coded `y ~ A + B + A:B`
*	Denotes the complete crossing variables. The code `y ~ A*B*C` expands to `y ~ A + B + C + A:B + A:C + B:C + A:B:C`
^	Denotes crossing to a specified degree. The code `y ~ (A+B+C)^2` expands to `y ~ A + B + C + A:B + A:C + A:B`
.	Denotes all remaining variables. The code `y ~ .` expands to `y ~ A + B + C`

The aov() Function

Table 9.5 provides formulas for several common research designs. In this table, lowercase letters are quantitative variables, uppercase letters are grouping factors, and subject is a unique identifier variable for subjects.

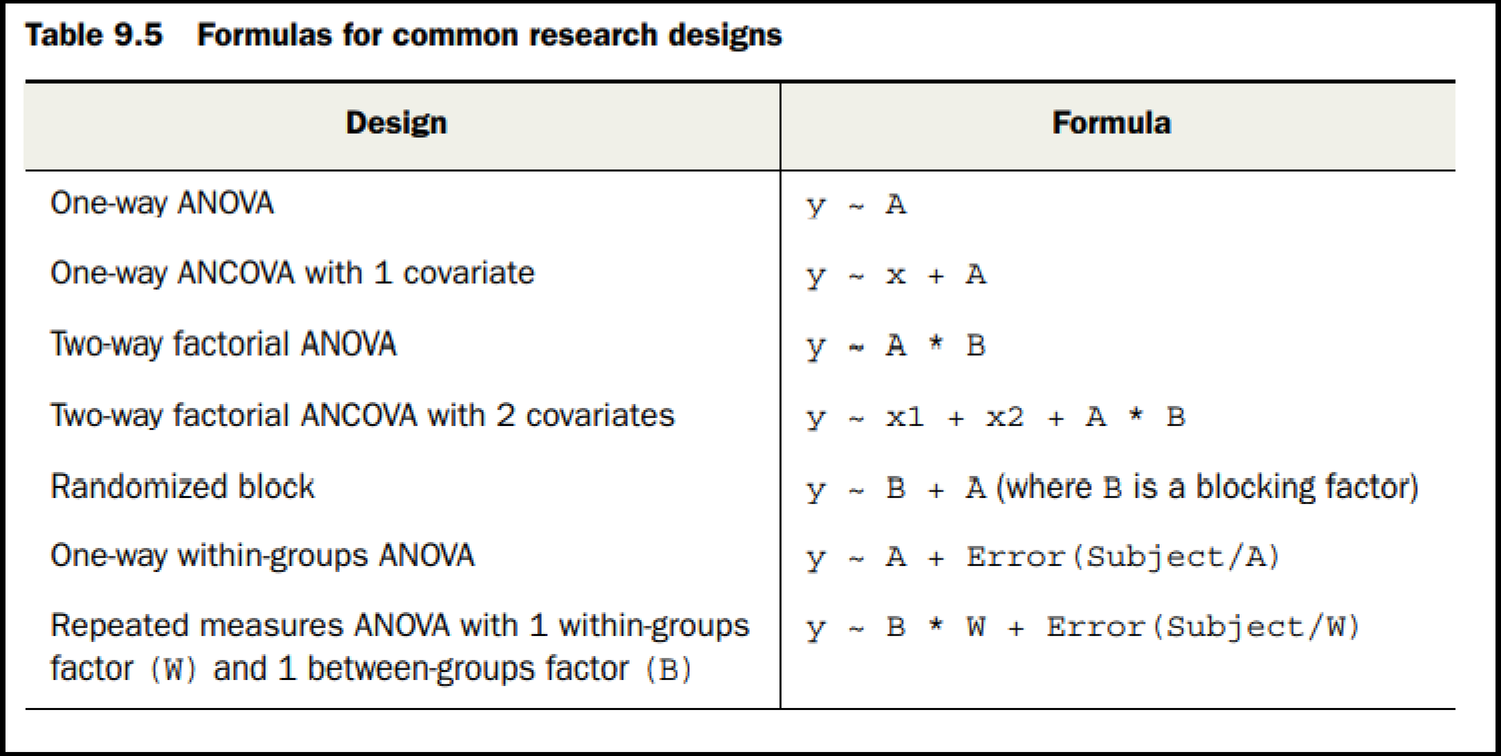


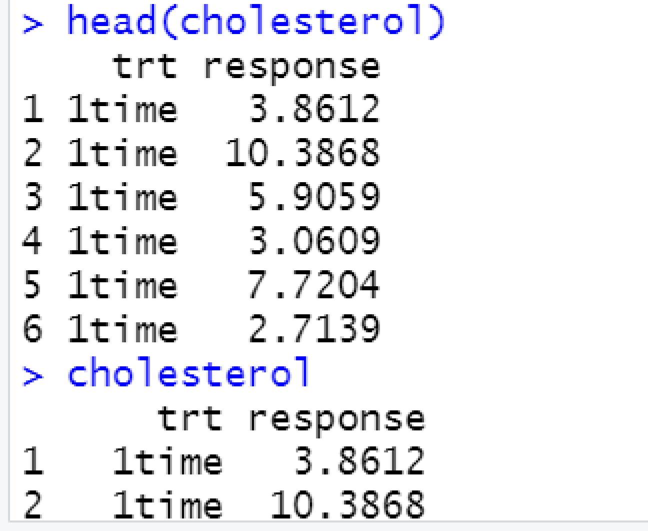
Table 9.5 Formulas for common research designs

Design	Formula
One-way ANOVA	`y ~ A`
One-way ANCOVA with 1 covariate	`y ~ x + A`
Two-way factorial ANOVA	`y ~ A * B`
Two-way factorial ANCOVA with 2 covariates	`y ~ x1 + x2 + A * B`
Randomized block	`y ~ B + A` (where `B` is a blocking factor)
One-way within-groups ANOVA	`y ~ A + Error(Subject/A)`
Repeated measures ANOVA with 1 within-groups factor (`W`) and 1 between-groups factor (`B`)	`y ~ B * W + Error(Subject/W)`

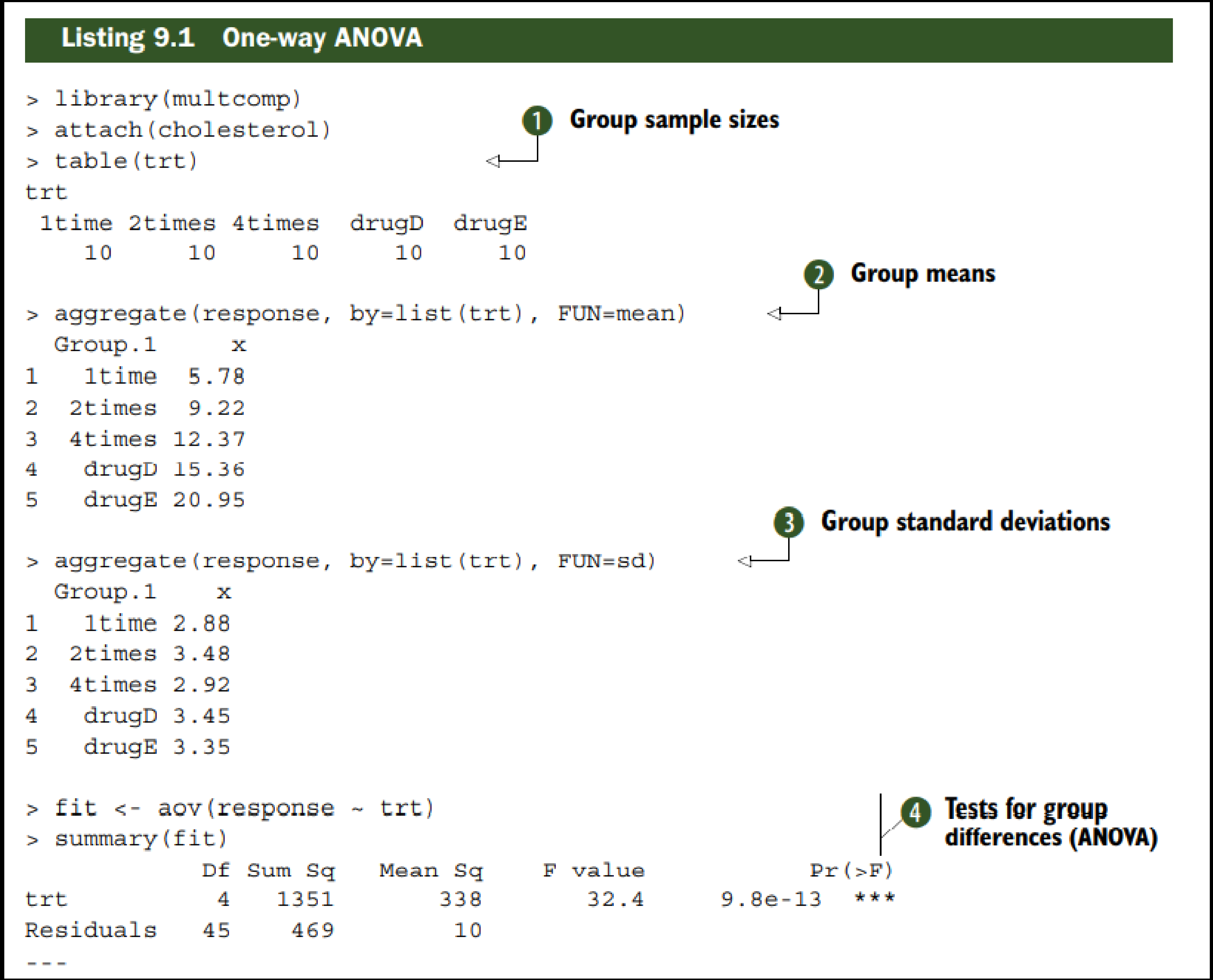
One-Way ANOVA

- ✓ In a one-way ANOVA, you’re interested in comparing the dependent variable means of two or more groups defined by a categorical grouping factor.
- ✓ This example comes from the cholesterol dataset in the *multcomp* package, taken from *Westfall*, *Tobia*, *Rom*, & *Hochberg* (1999).
- ✓ Fifty patients received one of five cholesterol-reducing drug regimens (*trt*). Three of the treatment conditions involved the same drug administered as *20mg* once per day (1 time), *10mg* twice per day (2 times), or *5mg* four times per day (4 times).
- ✓ The two remaining conditions (*drugD* and *drugE*) represented competing drugs.
- ✓ Which drug regimen produced the greatest cholesterol reduction (response)? The analysis is provided in the following listing.

One-Way ANOVA



```
> head(cholesterol)
    trt response
1 1time   3.8612
2 1time  10.3868
3 1time   5.9059
4 1time   3.0609
5 1time   7.7204
6 1time   2.7139
> cholesterol
      trt response
1   1time   3.8612
2   1time  10.3868
```



Listing 9.1 One-way ANOVA

```
> library(multcomp)
> attach(cholesterol)
> table(trt)
trt
 1time 2times 4times  drugD  drugE
    10     10     10     10     10

> aggregate(response, by=list(trt), FUN=mean)
  Group.1     x
1   1time  5.78
2  2times  9.22
3  4times 12.37
4   drugD 15.36
5   drugE 20.95

> aggregate(response, by=list(trt), FUN=sd)
  Group.1    x
1   1time 2.88
2  2times 3.48
3  4times 2.92
4   drugD 3.45
5   drugE 3.35

> fit <- aov(response ~ trt)
> summary(fit)
            Df  Sum Sq   Mean Sq    F value           Pr(>F)
trt          4    1351       338       32.4     9.8e-13  ***
Residuals   45     469        10
---
```

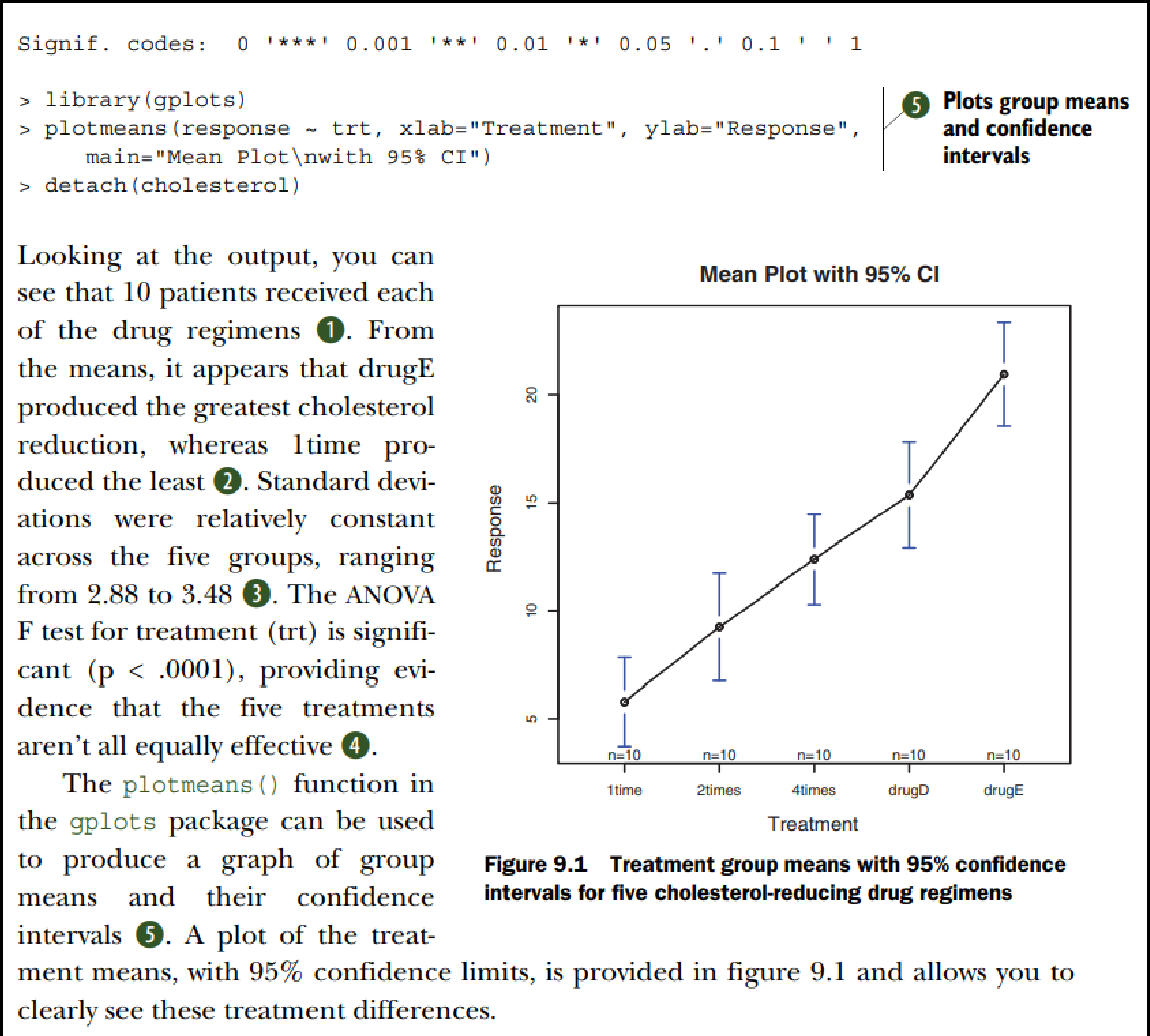
1 Group sample sizes

2 Group means

3 Group standard deviations

4 Tests for group differences (ANOVA)

One-Way ANOVA



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

> library(gplots)
> plotmeans(response ~ trt, xlab="Treatment", ylab="Response",
     main="Mean Plot\nwith 95% CI")
> detach(cholesterol)
```

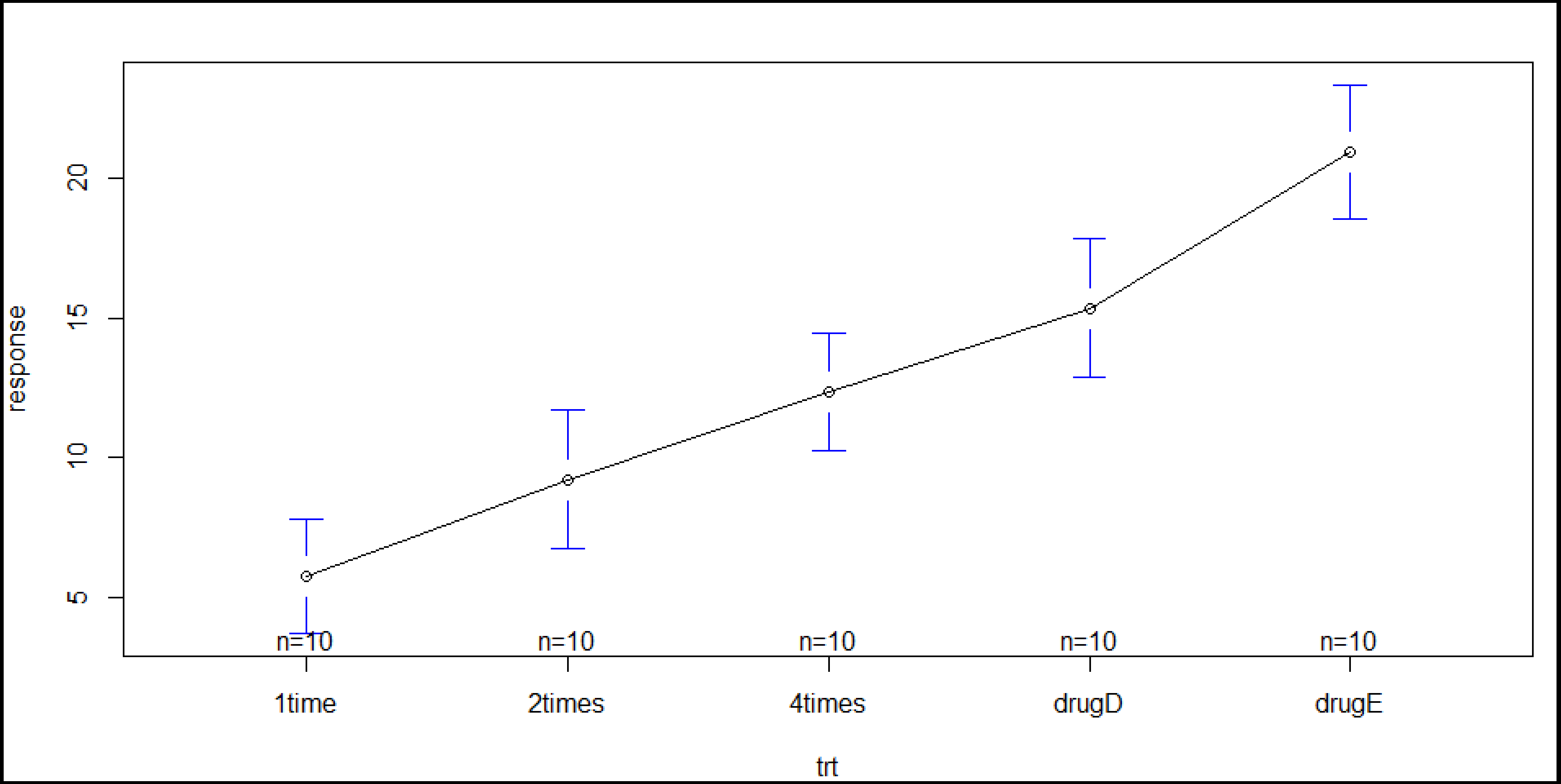
❺ Plots group means and confidence intervals

Looking at the output, you can see that 10 patients received each of the drug regimens ❶. From the means, it appears that drugE produced the greatest cholesterol reduction, whereas 1time produced the least ❷. Standard deviations were relatively constant across the five groups, ranging from 2.88 to 3.48 ❸. The ANOVA F test for treatment (trt) is significant (p < .0001), providing evidence that the five treatments aren't all equally effective ❹.

The plotmeans() function in the gplots package can be used to produce a graph of group means and their confidence intervals ❺. A plot of the treatment means, with 95% confidence limits, is provided in figure 9.1 and allows you to clearly see these treatment differences.

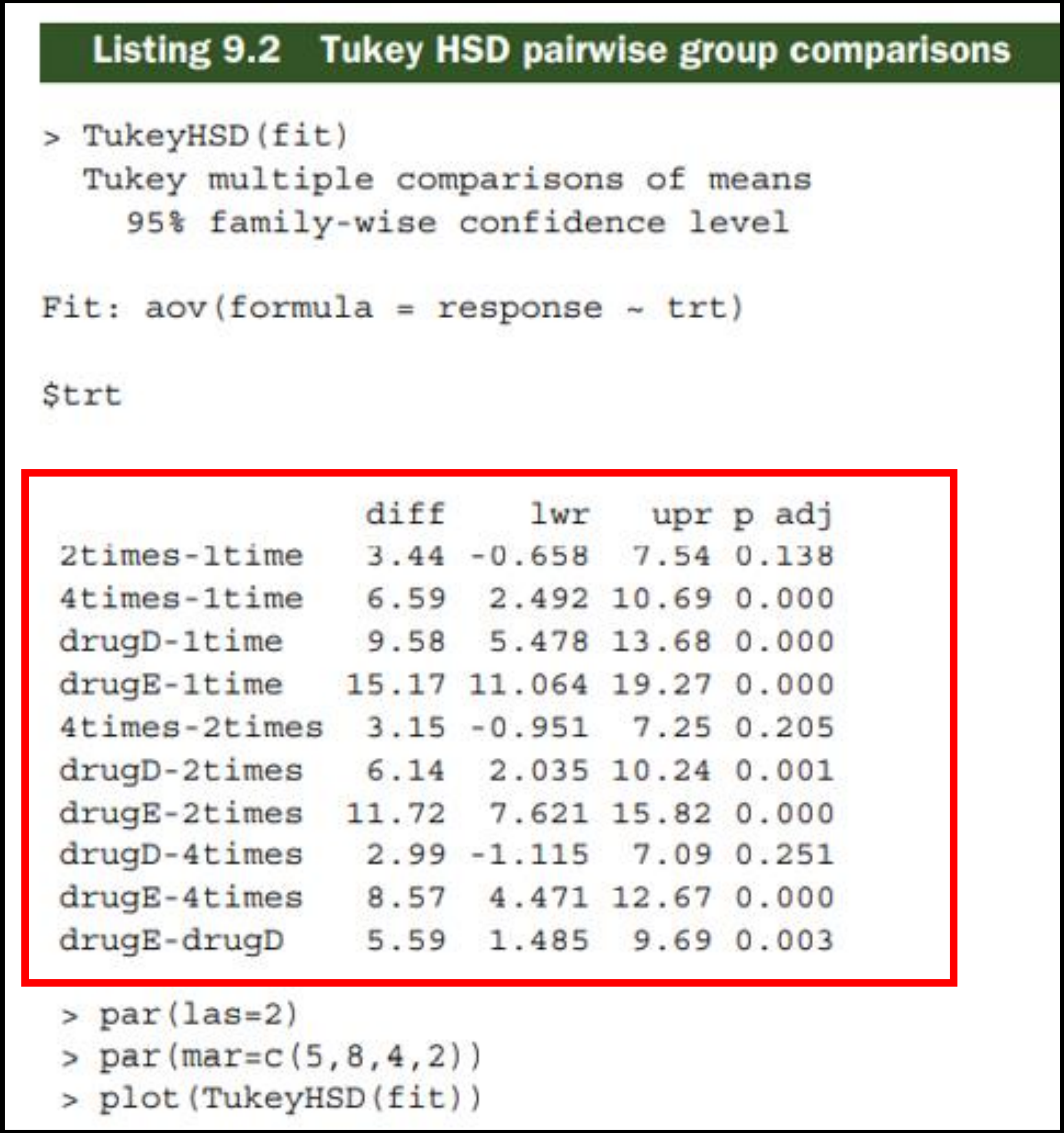
Figure 9.1 Treatment group means with 95% confidence intervals for five cholesterol-reducing drug regimens

One-Way ANOVA



Multiple Comparisons

- The ANOVA F-test for treatment tells you that the five drug regimens aren’t equally effective, but it doesn’t tell you which treatments differ from one another.
- You can use a multiple comparison procedure to answer this question. For example, the *TukeyHSD*() function provides a test of all pairwise differences between group means, as shown next (Since all groups have n=10, we can use the *Tukey* test.)



Listing 9.2 Tukey HSD pairwise group comparisons

```
> TukeyHSD(fit)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = response ~ trt)

$trt

                diff    lwr   upr p adj
 2times-1time   3.44 -0.658  7.54 0.138
 4times-1time   6.59  2.492 10.69 0.000
 drugD-1time    9.58  5.478 13.68 0.000
 drugE-1time   15.17 11.064 19.27 0.000
 4times-2times  3.15 -0.951  7.25 0.205
 drugD-2times   6.14  2.035 10.24 0.001
 drugE-2times  11.72  7.621 15.82 0.000
 drugD-4times   2.99 -1.115  7.09 0.251
 drugE-4times   8.57  4.471 12.67 0.000
 drugE-drugD    5.59  1.485  9.69 0.003
 > par(las=2)
 > par(mar=c(5,8,4,2))
 > plot(TukeyHSD(fit))
```

Multiple Comparisons

par(mar=c()): Sets the bottom, left, top and right margins respectively of the plot region in the number of lines of text.

For example, the mean cholesterol reductions for 1 time and 2 times aren’t significantly different from each other (p = 0.138), whereas the difference between 1 time and 4 times is significantly different (p<.001). The pairwise comparisons are plotted in figure 9.2.

The first *par* statement rotates the axis labels, and the second one increases the left margin area so that the labels fit (*par* options are covered in chapter 3). In this graph, confidence intervals that include 0 indicate treatments that aren’t significantly different (p > 0.5).

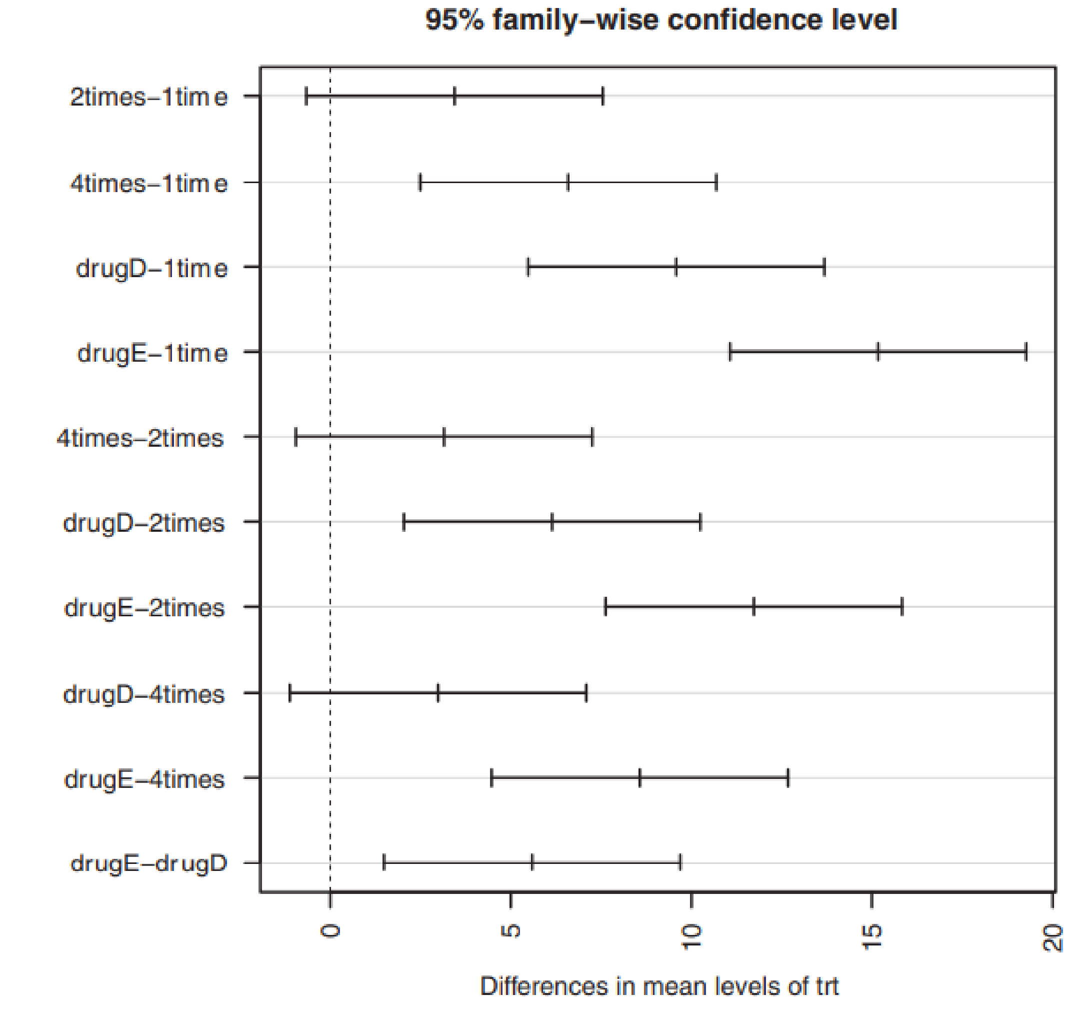
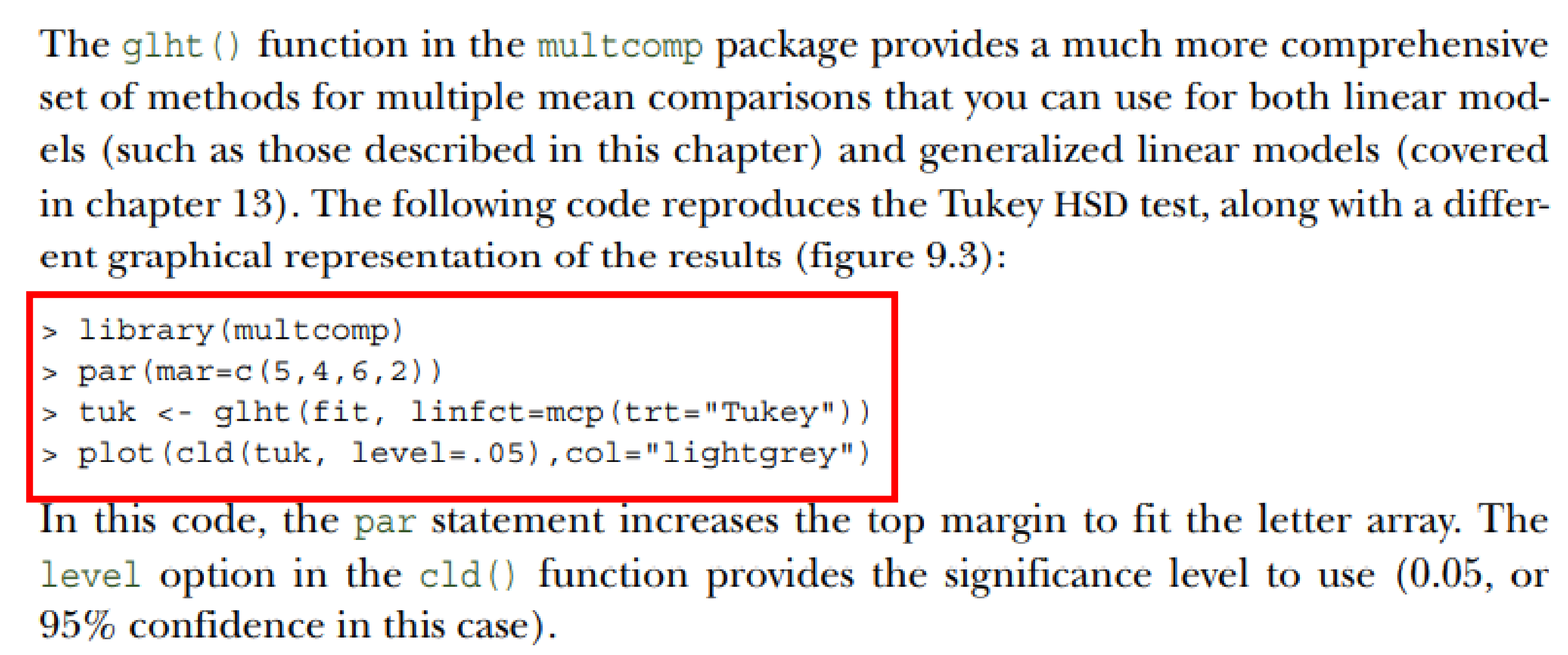


Figure 9.2 **Plot of Tukey HSD pairwise mean comparisons**

Multiple Comparisons

glht: General Linear Hypotheses Test



The glht() function in the multcomp package provides a much more comprehensive set of methods for multiple mean comparisons that you can use for both linear models (such as those described in this chapter) and generalized linear models (covered in chapter 13). The following code reproduces the Tukey HSD test, along with a different graphical representation of the results (figure 9.3):

```
> library(multcomp)
> par(mar=c(5,4,6,2))
> tuk <- glht(fit, linfct=mcp(trt="Tukey"))
> plot(cld(tuk, level=.05),col="lightgrey")
```

In this code, the par statement increases the top margin to fit the letter array. The level option in the cld() function provides the significance level to use (0.05, or 95% confidence in this case).

***cld*: Set up a compact letter display of all pair-wise comparisons**

Linfct: a specification of the linear hypotheses to be tested. Linear functions can be specified by either the matrix of coefficients or by symbolic descriptions of one or more linear hypotheses. Multiple comparisons in ANOVA models are specified by objects returned from function **mcp**.

Multiple Comparisons

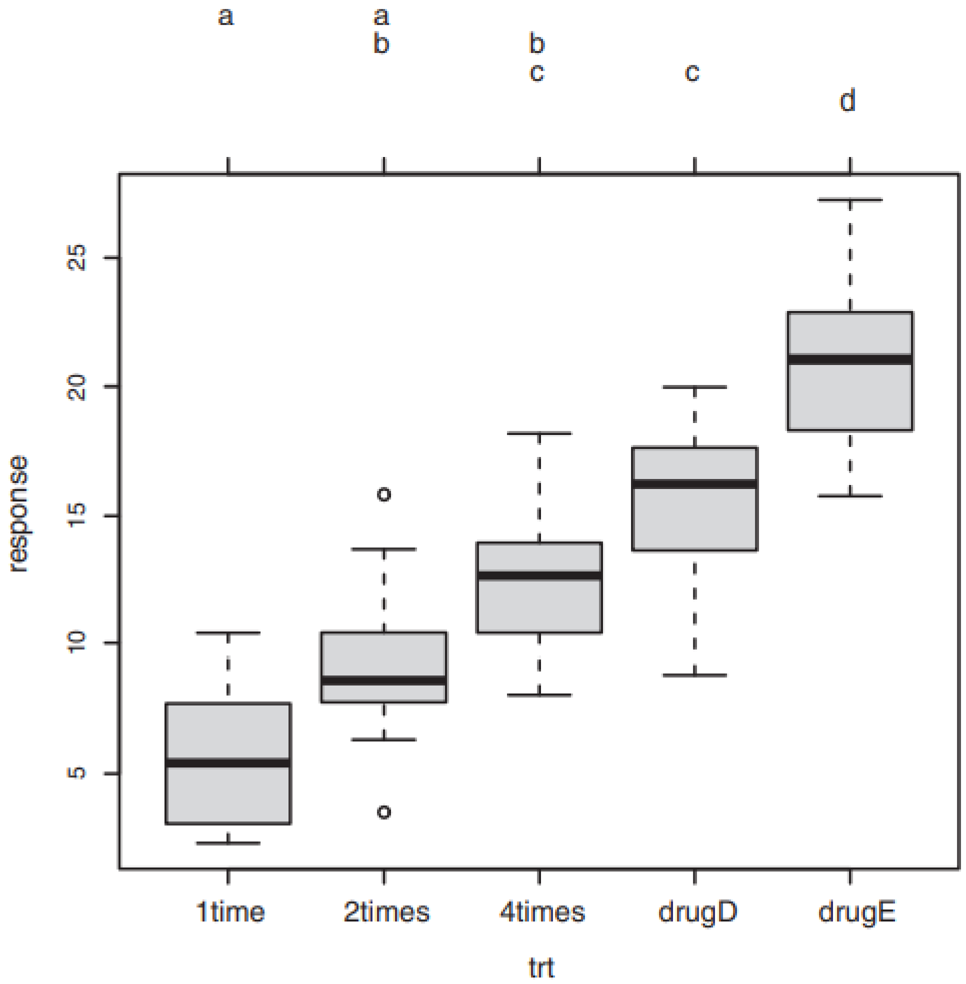
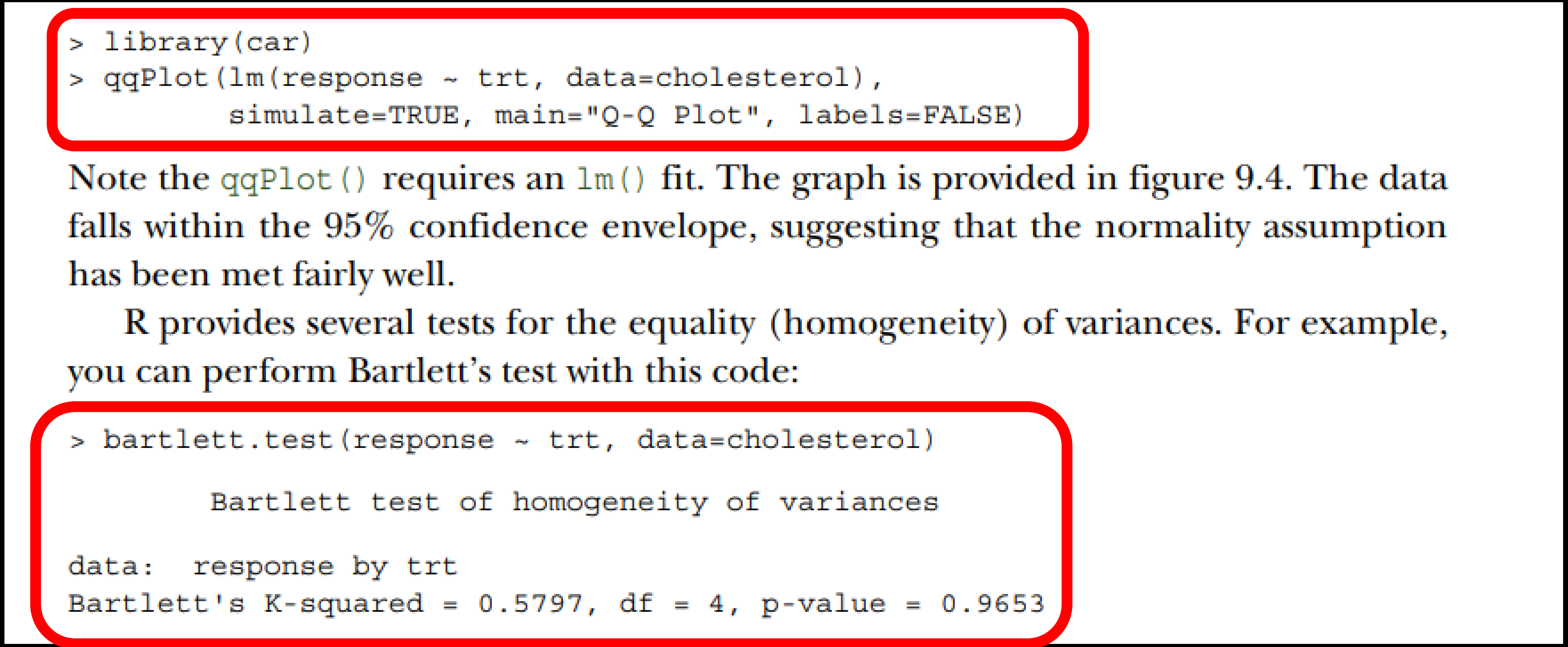


Figure 9.3 Tukey HSD tests provided by the `multcomp` package

Q–Q plot (quantile-quantile plot)

Assessing Test Assumptions

As you saw in the previous chapter, confidence in results depends on the degree to which your data satisfies the assumptions underlying the statistical tests. In a one-way ANOVA, the dependent variable is assumed to be normally distributed and have equal variance in each group. You can use a *Q-Q plot* to assess the normality assumption:



```
> library(car)
> qqPlot(lm(response ~ trt, data=cholesterol),
         simulate=TRUE, main="Q-Q Plot", labels=FALSE)
```

Note the `qqPlot()` requires an `lm()` fit. The graph is provided in figure 9.4. The data falls within the 95% confidence envelope, suggesting that the normality assumption has been met fairly well.

R provides several tests for the equality (homogeneity) of variances. For example, you can perform Bartlett's test with this code:

```
> bartlett.test(response ~ trt, data=cholesterol)

        Bartlett test of homogeneity of variances

data:  response by trt
Bartlett's K-squared = 0.5797, df = 4, p-value = 0.9653
```

Assessing Test Assumptions

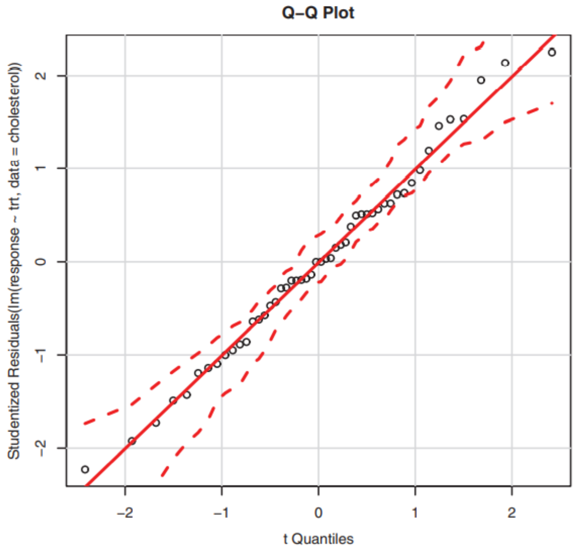
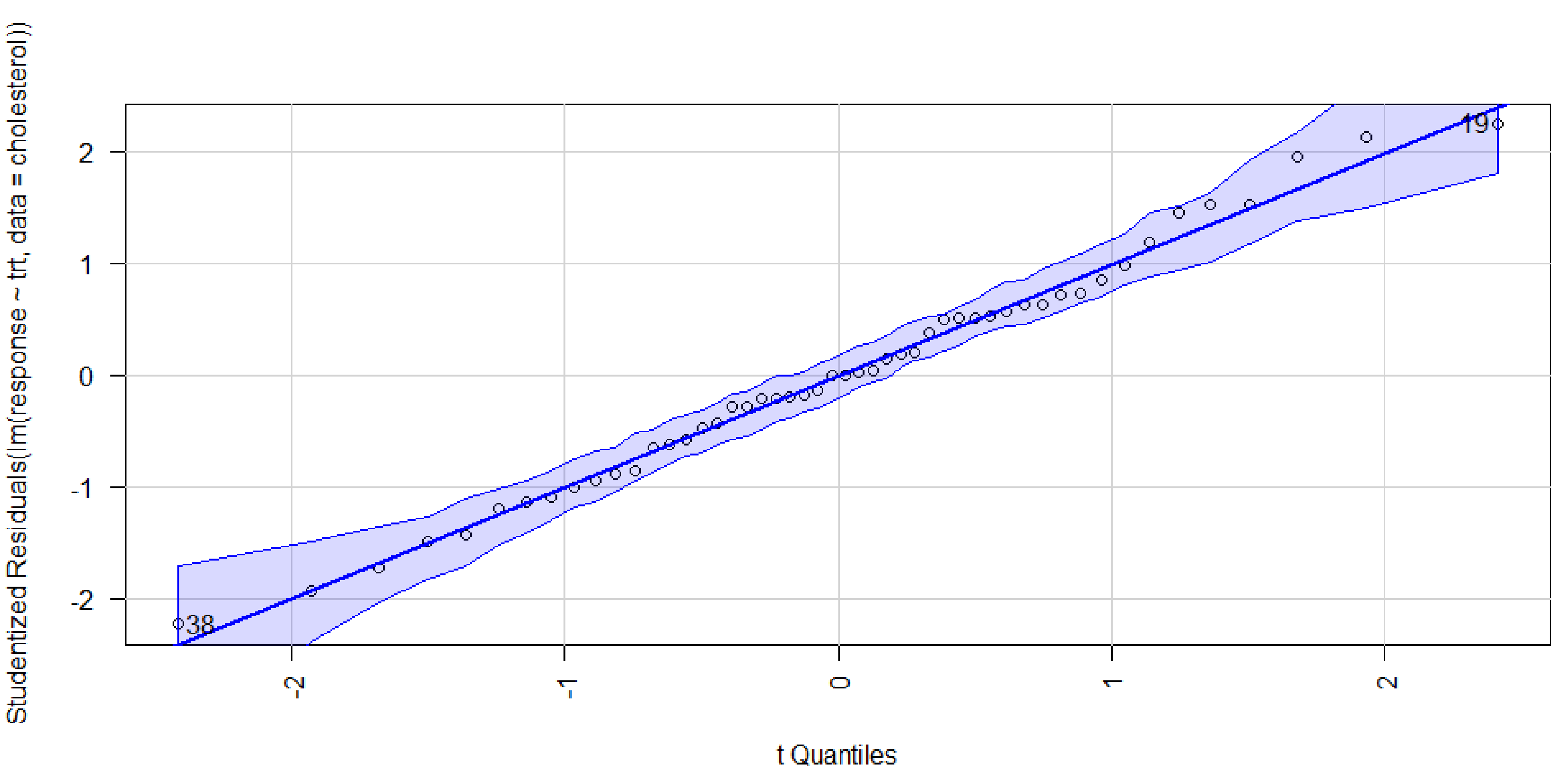
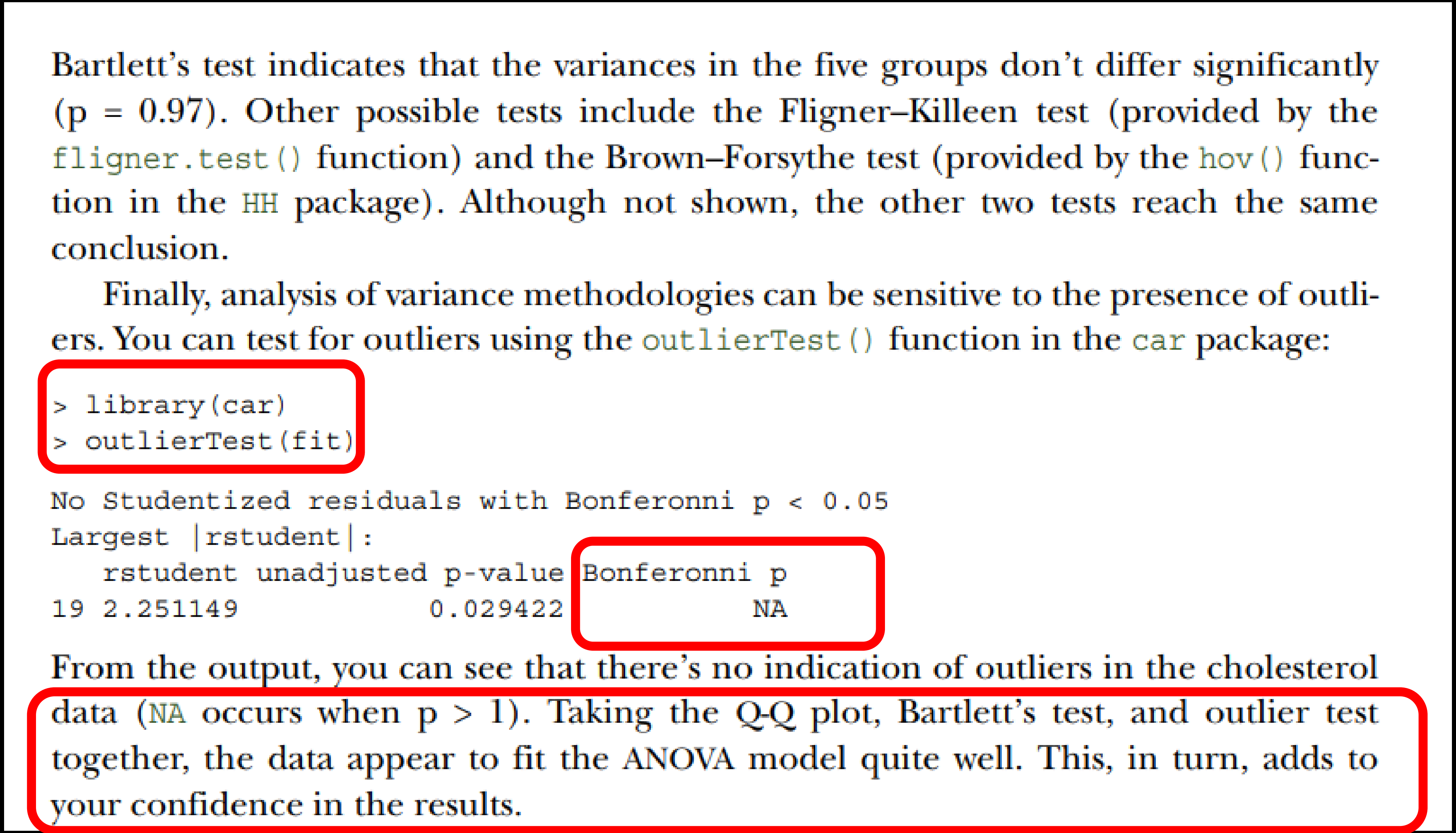


Figure 9.4
Test of normality



Assessing Test Assumptions



Bartlett's test indicates that the variances in the five groups don't differ significantly (p = 0.97). Other possible tests include the Fligner–Killeen test (provided by the `fligner.test()` function) and the Brown–Forsythe test (provided by the `hov()` function in the `HH` package). Although not shown, the other two tests reach the same conclusion.

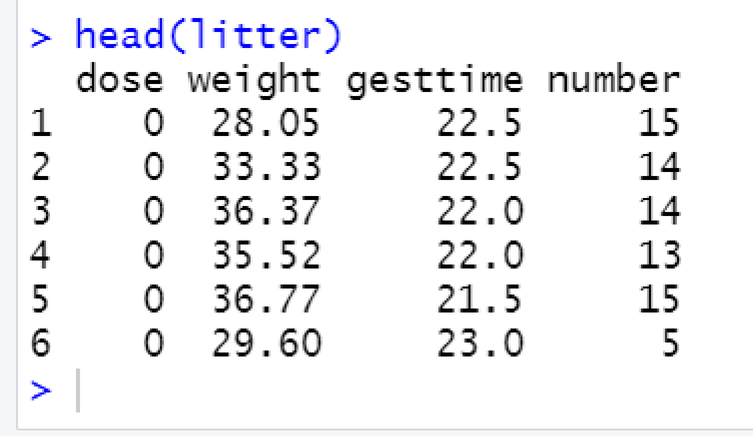
Finally, analysis of variance methodologies can be sensitive to the presence of outliers. You can test for outliers using the `outlierTest()` function in the `car` package:

```
> library(car)
> outlierTest(fit)

No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
   rstudent unadjusted p-value Bonferonni p
19 2.251149           0.029422           NA
```

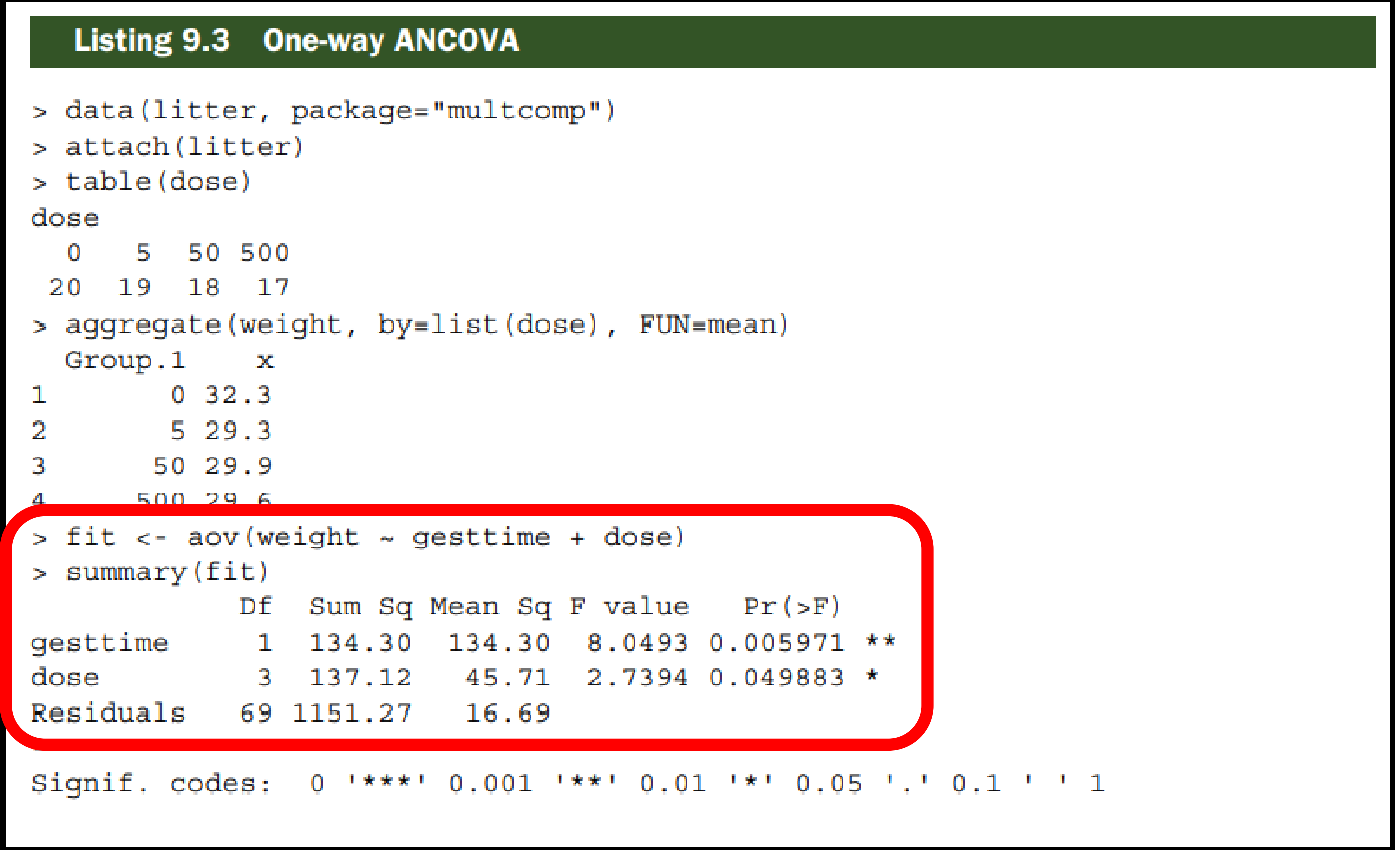
From the output, you can see that there's no indication of outliers in the cholesterol data (`NA` occurs when p > 1). Taking the Q-Q plot, Bartlett's test, and outlier test together, the data appear to fit the ANOVA model quite well. This, in turn, adds to your confidence in the results.

One-Way ANCOVA

A one-way analysis of covariance (ANCOVA) extends the one-way ANOVA to include one or more quantitative covariates. This example comes from the *litter* dataset in the *multcomp* package. Pregnant mice were divided into four treatment groups; each group received a different dose of a drug (0, 5, 50, or 500). The mean **post-birth weight** for each litter was the dependent variable, and gestation time was included as a covariate. The analysis is given in the following listing.



```
> head(litter)
  dose weight gesttime number
1    0  28.05     22.5     15
2    0  33.33     22.5     14
3    0  36.37     22.0     14
4    0  35.52     22.0     13
5    0  36.77     21.5     15
6    0  29.60     23.0      5
>
```



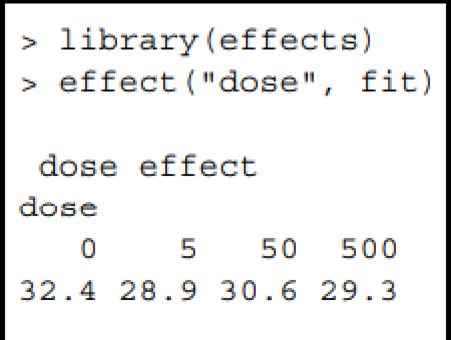
Listing 9.3 One-way ANCOVA

```
> data(litter, package="multcomp")
> attach(litter)
> table(dose)
dose
  0   5  50 500
 20  19  18  17
> aggregate(weight, by=list(dose), FUN=mean)
  Group.1    x
1       0 32.3
2       5 29.3
3      50 29.9
4     500 29.6
> fit <- aov(weight ~ gesttime + dose)
> summary(fit)
            Df  Sum Sq Mean Sq F value   Pr(>F)
gesttime     1  134.30  134.30  8.0493 0.005971 **
dose         3  137.12   45.71  2.7394 0.049883 *
Residuals   69 1151.27   16.69

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

One-Way ANCOVA

From the *table*() function, you can see that there is **an unequal number of litters at each dosage level**, with 20 litters at zero dosage (no drug) and 17 litters at dosage 500. Based on the group means provided by the *aggregate*() function, the no-drug group had the highest mean litter weight (32.3).

The ANCOVA F-tests indicate that (a) gestation time was related to birth weight, and (b) drug dosage was related to birth weight after controlling for gestation time. The mean birth weight isn’t the same for each of the drug dosages, after controlling for gestation time. Because you’re using a covariate, you may want to obtain adjusted group means—that is, the group means obtained after parting out the effects of the covariate. You can use the *effect*() function in the *effects* library to calculate adjusted means:



```
> library(effects)
> effect("dose", fit)

 dose effect
dose
   0    5   50  500
32.4 28.9 30.6 29.3
```

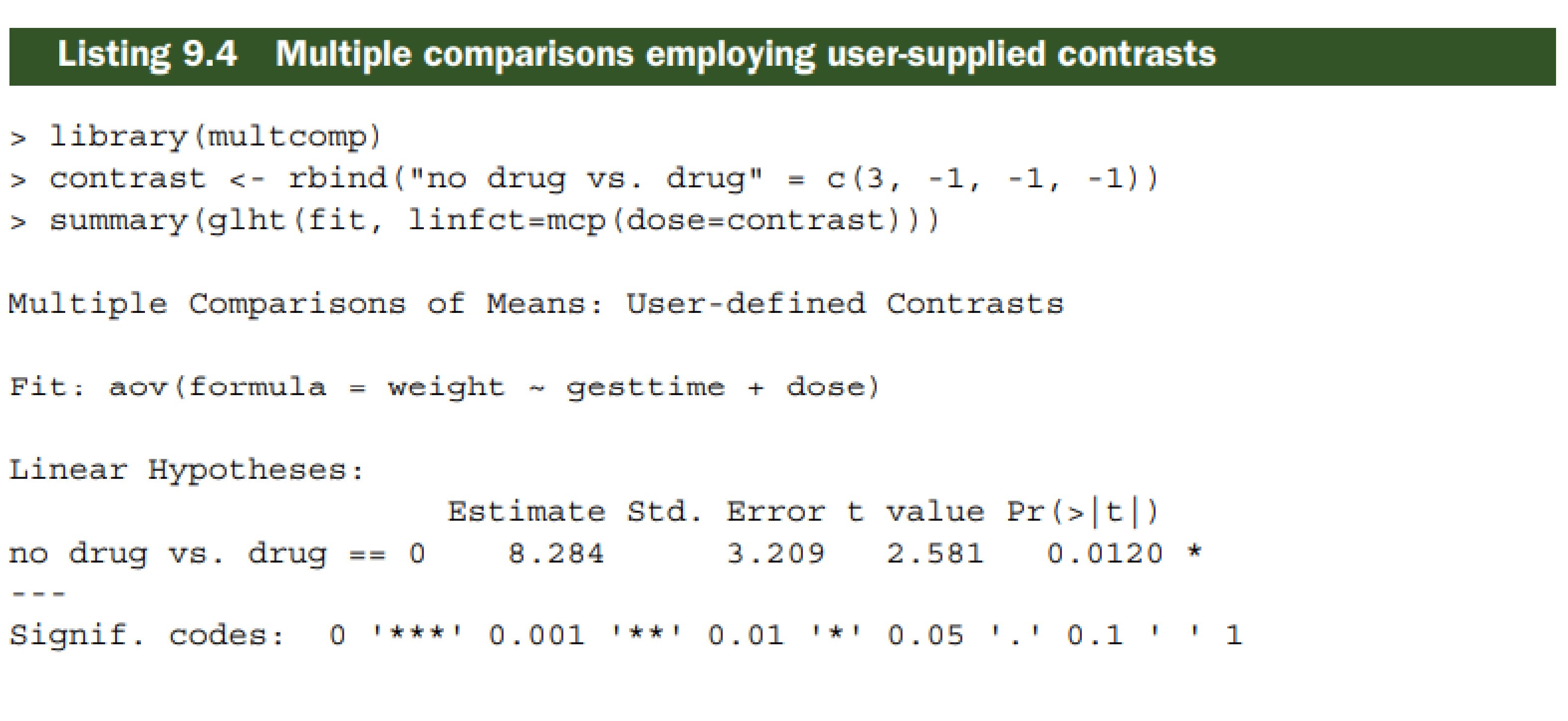
One-Way ANCOVA

In this case, the adjusted means are similar to the unadjusted means produced by the *aggregate*() function, but this won’t always be the case.

The *effects* package provides a powerful method of obtaining adjusted means for complex research designs and presenting them visually. See the package documentation on *CRAN* for more details. As with the one-way ANOVA example in the last section, the F-test for *dose* indicates that the treatments don’t have the same mean *birth weight*, but it doesn’t tell you which means differ from one another.

Again you can use the multiple comparison procedures provided by the *multcomp* package to compute all pairwise mean comparisons.
Additionally, the *multcomp* package can be used to test specific user-defined hypotheses about the means. Suppose you’re interested in whether the no-drug condition differs from the three-drug condition. The code in the following listing can be used to test this hypothesis.

One-Way ANCOVA



Listing 9.4 Multiple comparisons employing user-supplied contrasts

```
> library(multcomp)
> contrast <- rbind("no drug vs. drug" = c(3, -1, -1, -1))
> summary(glht(fit, linfct=mcp(dose=contrast)))

Multiple Comparisons of Means: User-defined Contrasts

Fit: aov(formula = weight ~ gesttime + dose)

Linear Hypotheses:
                      Estimate Std. Error t value Pr(>|t|)
no drug vs. drug == 0    8.284      3.209   2.581   0.0120 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The *contrast c(3, -1, -1, -1)* specifies a comparison of the first group with the average of the other three. The hypothesis is tested with a *t* statistic (2.581 in this case), which is significant at the p < 0.05 level. Therefore, you can conclude that the no-drug group has a higher *birth weight* than drug conditions. Other contrasts can be added to the *rbind*() function (see help(*glht*) for details).

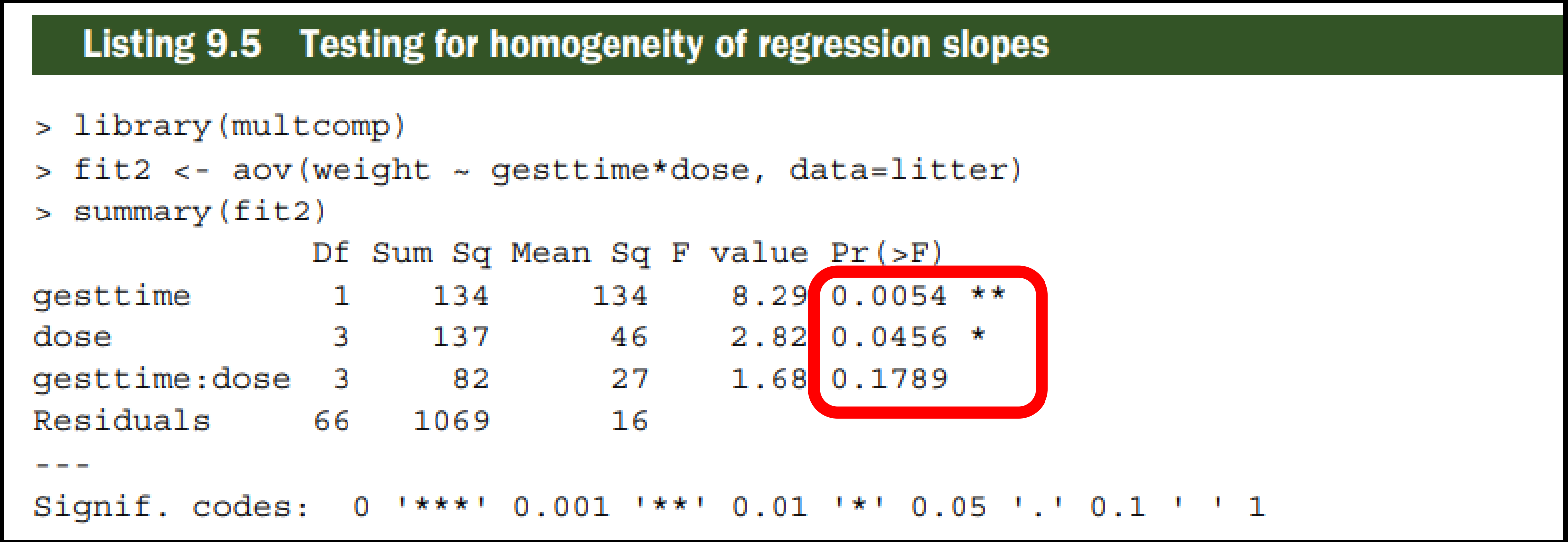
Assessing Test Assumptions

ANCOVA designs make the same normality and homogeneity of variance assumptions described for ANOVA designs, and you can test these assumptions using the same procedures described in section 9.3.2.

In addition, standard ANCOVA designs assume homogeneity of regression slopes. In this case, it's assumed that the regression slope for predicting birth weight from gestation time is the same in each of the four treatment groups. A test for the homogeneity of regression slopes can be obtained by including a gestation × dose interaction term in your ANCOVA model.

A significant interaction would imply that the relationship between gestation and birth weight depends on the level of the dose variable. The code and results are provided in the following listing.

Assessing Test Assumptions

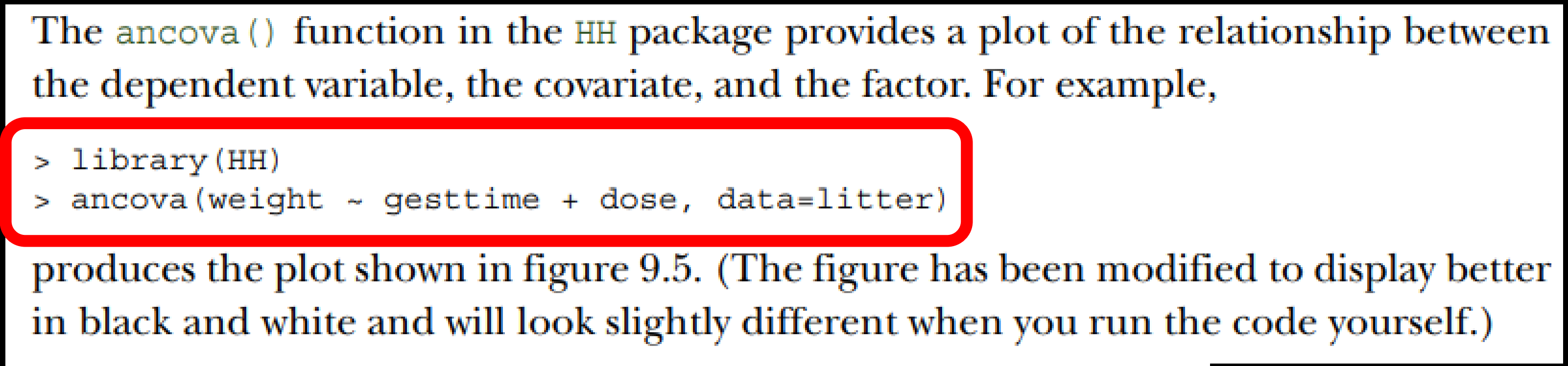


Listing 9.5 Testing for homogeneity of regression slopes

```
> library(multcomp)
> fit2 <- aov(weight ~ gesttime*dose, data=litter)
> summary(fit2)
              Df Sum Sq Mean Sq F value Pr(>F)
gesttime       1    134     134    8.29 0.0054 **
dose           3    137      46    2.82 0.0456 *
gesttime:dose  3     82      27    1.68 0.1789
Residuals     66   1069      16
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interaction is nonsignificant, supporting the assumption of the equality of slopes. If the assumption is untenable, you could try transforming the covariate or dependent variable, using a model that accounts for separate slopes, or employing a nonparametric ANCOVA method that doesn’t require homogeneity of regression slopes. See the *sm.ancova*() function in the *sm* package for an example of the latter.

Visualizing the Results



The ancova() function in the HH package provides a plot of the relationship between the dependent variable, the covariate, and the factor. For example,

```
> library(HH)
> ancova(weight ~ gesttime + dose, data=litter)
```

produces the plot shown in figure 9.5. (The figure has been modified to display better in black and white and will look slightly different when you run the code yourself.)

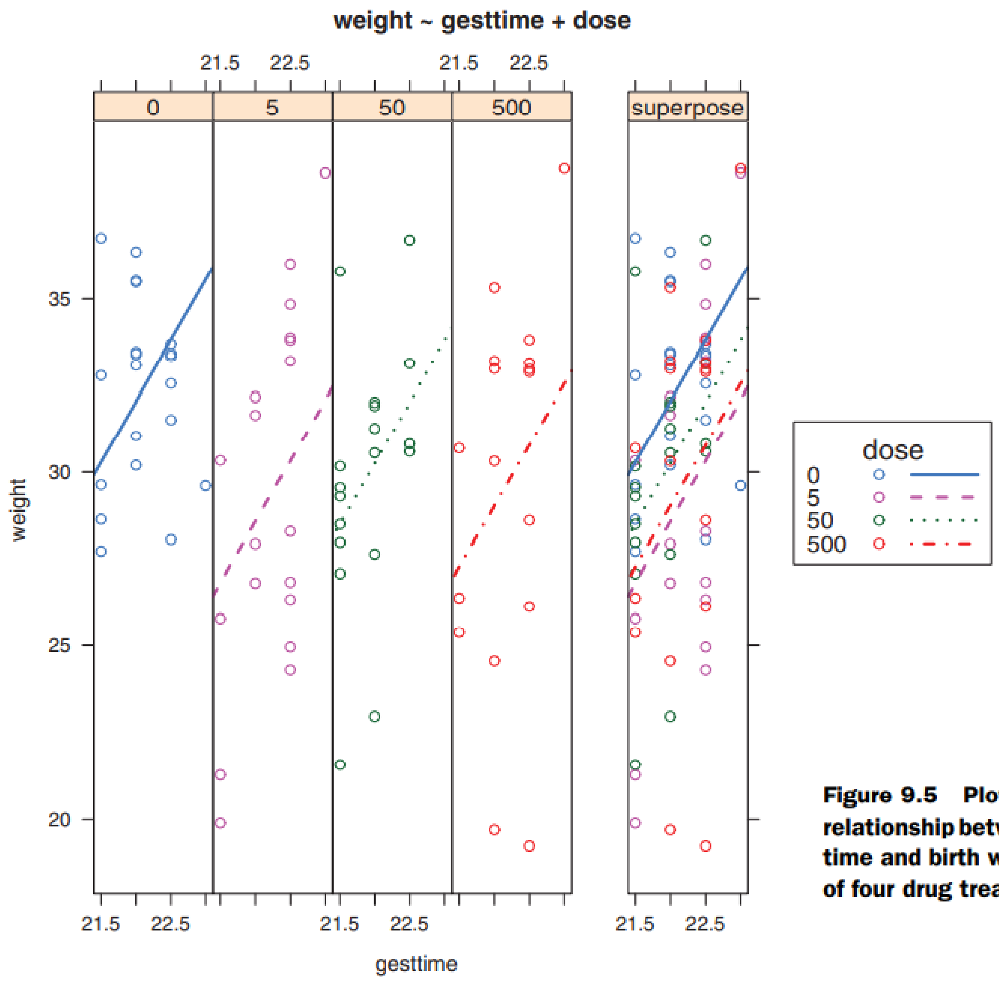


Figure 9.5 Plot of the relationship between gestation time and birth weight for each of four drug treatment groups

Visualizing the Results

Here you can see that the regression lines for predicting birth weight from gestation time are parallel in each group but have different intercepts.

As gestation time increases, birth weight increases. Additionally, you can see that the zero-dose group has the largest intercept and the five-dose group has the lowest intercept.

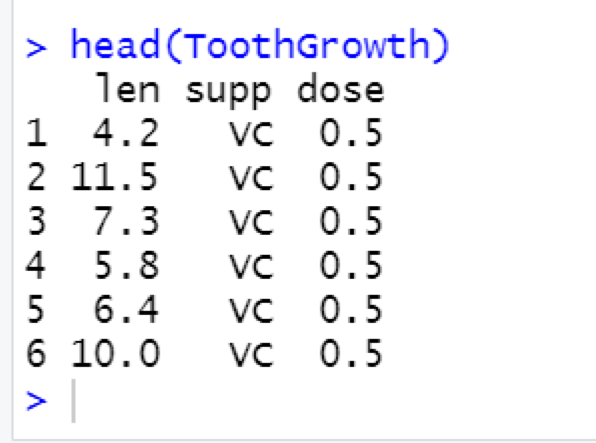
The lines are parallel because they’ve been specified to be. If you used the statement *ancova(weight ~ gesttime*dose)* instead, you’d generate a plot that allows both the slopes and intercepts to vary by group. This approach is useful for visualizing the case where the homogeneity of regression slopes doesn’t hold.

Two-way Factorial ANOVA

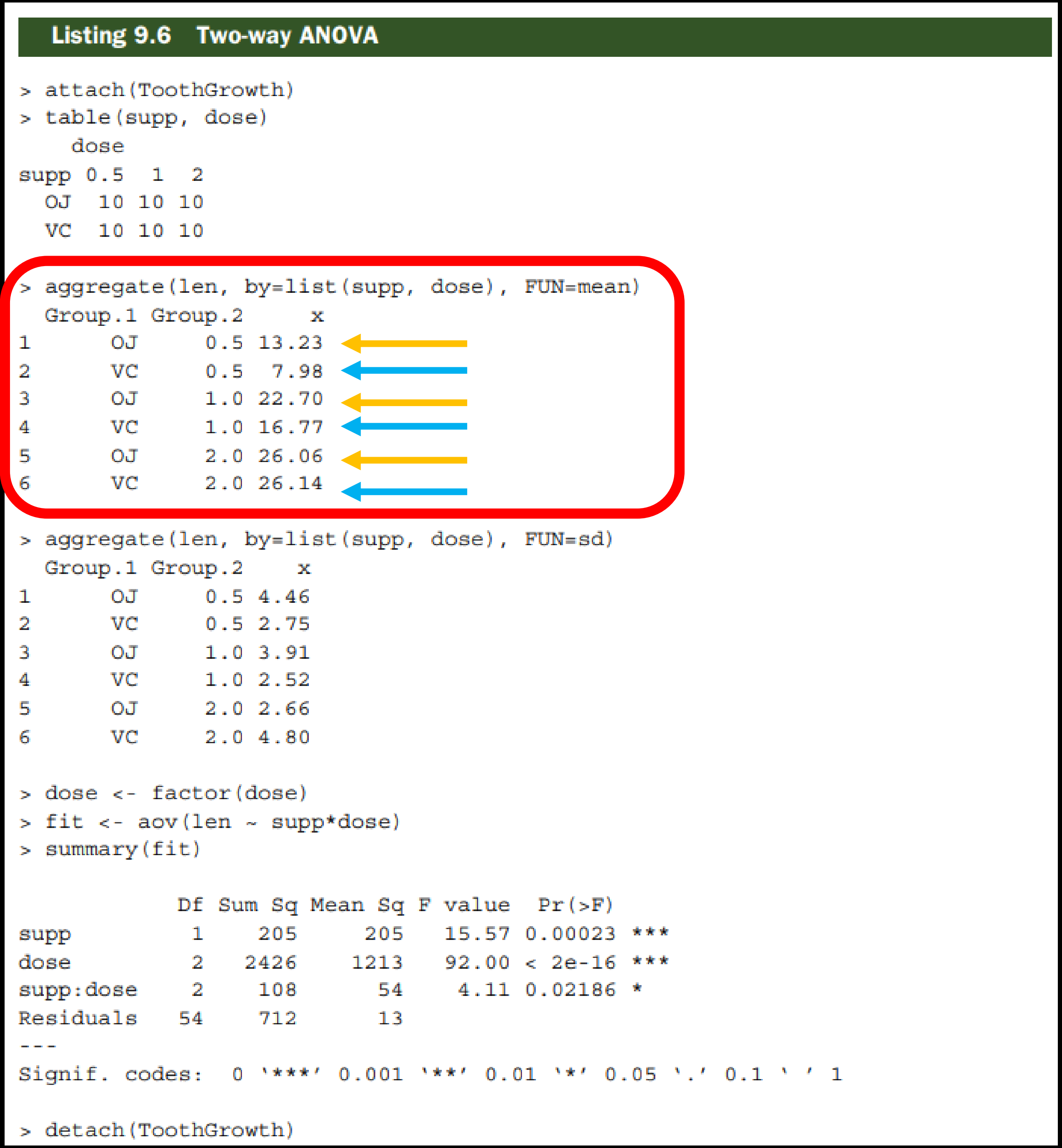
In a two-way factorial ANOVA, subjects are assigned to groups that are formed from the cross-classification of two factors. This example uses the *ToothGrowth* dataset in the *base* installation to demonstrate a two-way between-groups ANOVA. Sixty guinea pigs are randomly assigned to receive one of three levels of ascorbic acid (0.5, 1, or 2 mg) and one of two delivery methods (orange juice or Vitamin C), under the restriction that each treatment combination has 10 guinea pigs. The dependent variable is tooth length. The following listing shows the code for the analysis.

The table statement indicates that you have a balanced design (equal sample sizes in each cell of the design), and the aggregate statements provide the cell means and standard deviations. The dose variable is converted to a factor so that the *aov*() function will treat it as a grouping variable, rather than a numeric covariate. The ANOVA table provided by the *summary*() function indicates that both main effects (*supp* and *dose*) and the interaction between these factors are significant.

Two-way Factorial ANOVA



```
> head(ToothGrowth)
   len supp dose
1  4.2   VC  0.5
2 11.5   VC  0.5
3  7.3   VC  0.5
4  5.8   VC  0.5
5  6.4   VC  0.5
6 10.0   VC  0.5
> 
```

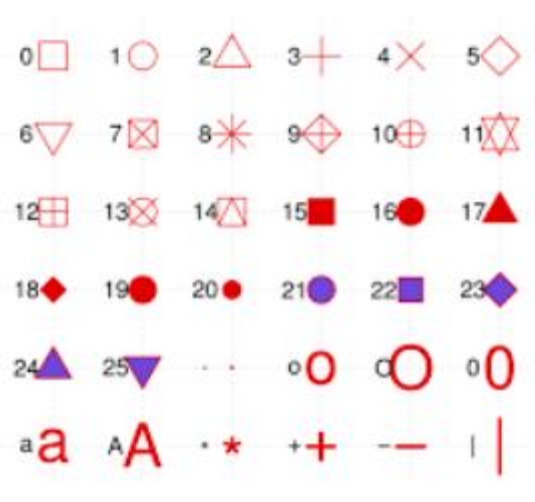


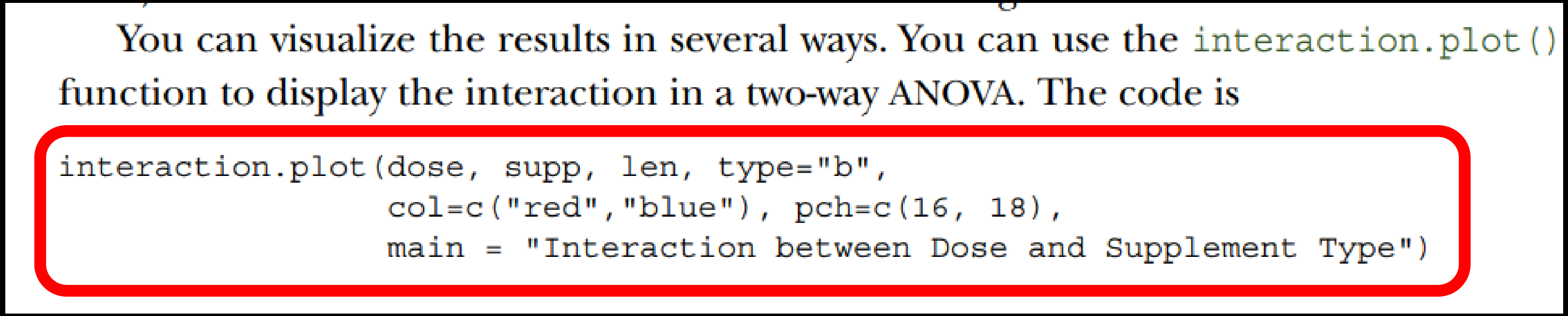
Listing 9.6 Two-way ANOVA

```
> attach(ToothGrowth)
> table(supp, dose)
    dose
supp 0.5  1  2
  OJ  10 10 10
  VC  10 10 10

> aggregate(len, by=list(supp, dose), FUN=mean)
  Group.1 Group.2     x
1      OJ     0.5 13.23
2      VC     0.5  7.98
3      OJ     1.0 22.70
4      VC     1.0 16.77
5      OJ     2.0 26.06
6      VC     2.0 26.14

> aggregate(len, by=list(supp, dose), FUN=sd)
  Group.1 Group.2    x
1      OJ     0.5 4.46
2      VC     0.5 2.75
3      OJ     1.0 3.91
4      VC     1.0 2.52
5      OJ     2.0 2.66
6      VC     2.0 4.80

> dose <- factor(dose)
> fit <- aov(len ~ supp*dose)
> summary(fit)

            Df Sum Sq Mean Sq F value   Pr(>F)
supp         1    205     205   15.57  0.00023 ***
dose         2   2426    1213   92.00  < 2e-16 ***
supp:dose    2    108      54    4.11  0.02186 *
Residuals   54    712      13
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> detach(ToothGrowth)
```

Two-way Factorial ANOVA

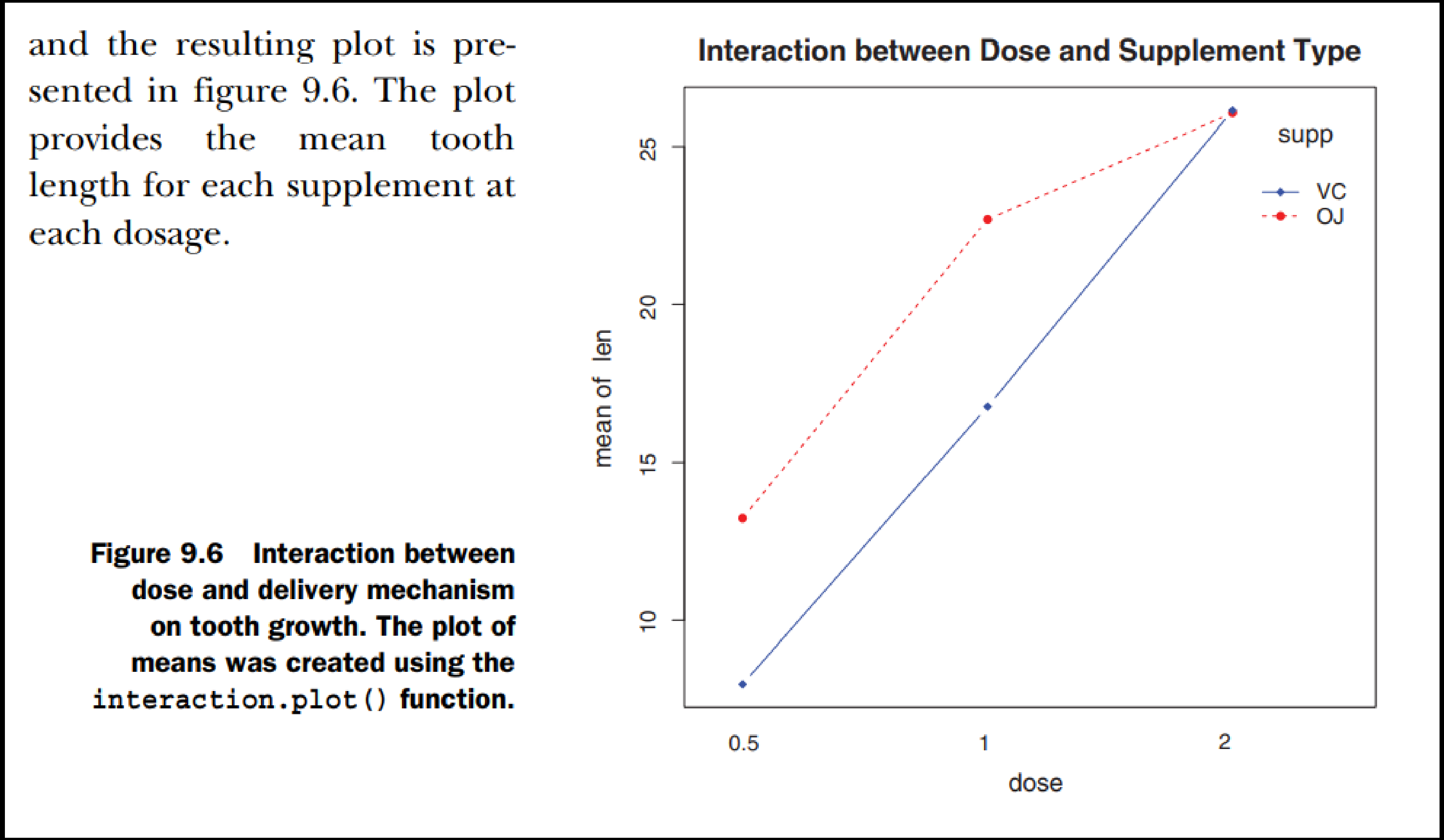
The *pch* stands for **plot character**. The *pch* option in R is used to define the point symbols in the functions plot () and lines (). The *pch* contains numeric values ranging from 0 to 25 or character symbols (“+”, “.”, “;”, etc.) specifying the point symbols (or shapes).





You can visualize the results in several ways. You can use the interaction.plot() function to display the interaction in a two-way ANOVA. The code is

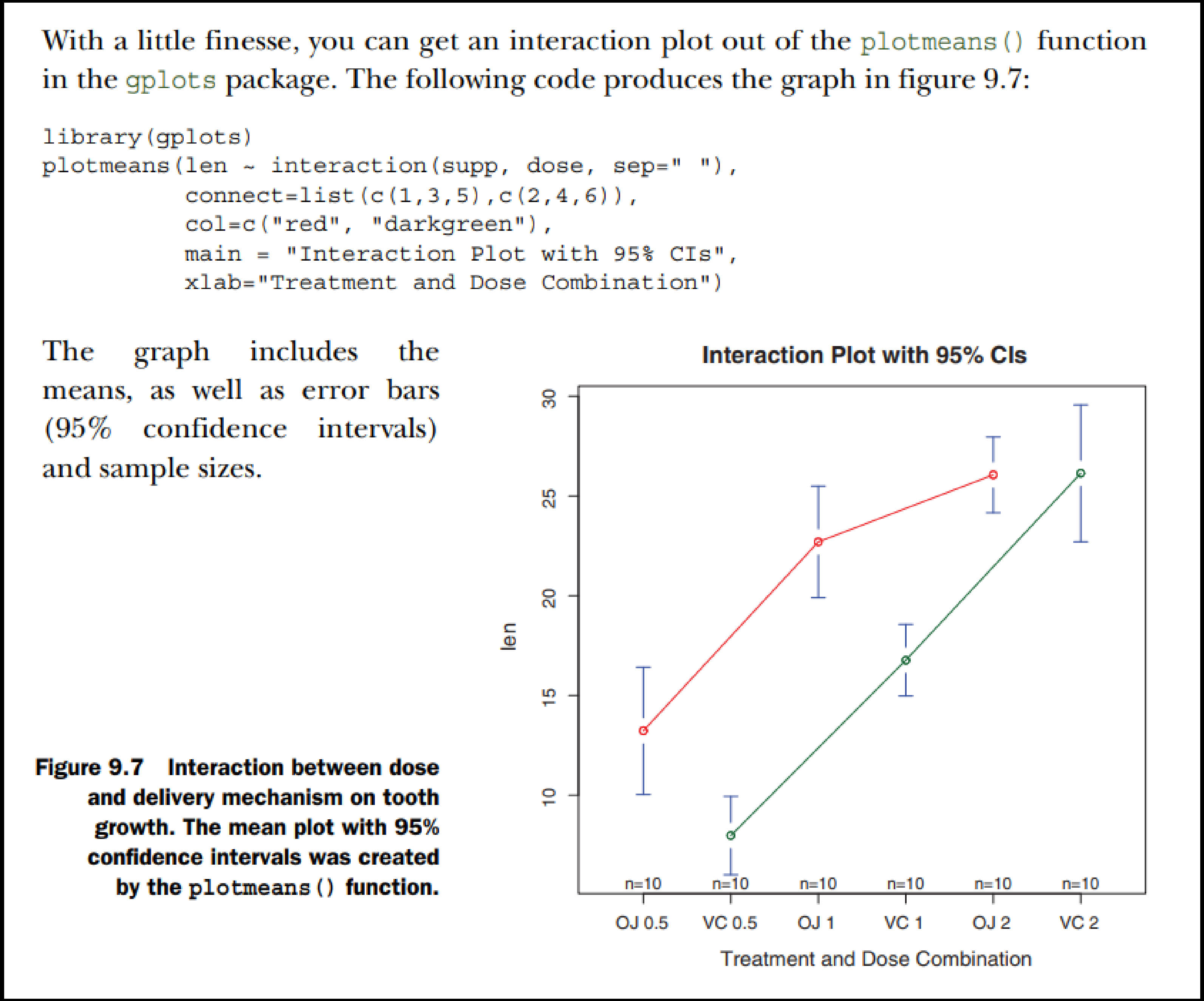
```
interaction.plot(dose, supp, len, type="b",
                 col=c("red","blue"), pch=c(16, 18),
                 main = "Interaction between Dose and Supplement Type")
```



and the resulting plot is presented in figure 9.6. The plot provides the mean tooth length for each supplement at each dosage.

Figure 9.6 Interaction between dose and delivery mechanism on tooth growth. The plot of means was created using the interaction.plot() function.

Two-way Factorial ANOVA



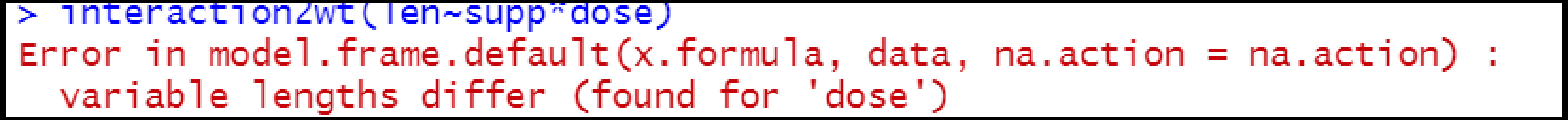
With a little finesse, you can get an interaction plot out of the plotmeans() function in the gplots package. The following code produces the graph in figure 9.7:

```
library(gplots)
plotmeans(len ~ interaction(supp, dose, sep=" "),
          connect=list(c(1,3,5),c(2,4,6)),
          col=c("red", "darkgreen"),
          main = "Interaction Plot with 95% CIs",
          xlab="Treatment and Dose Combination")
```

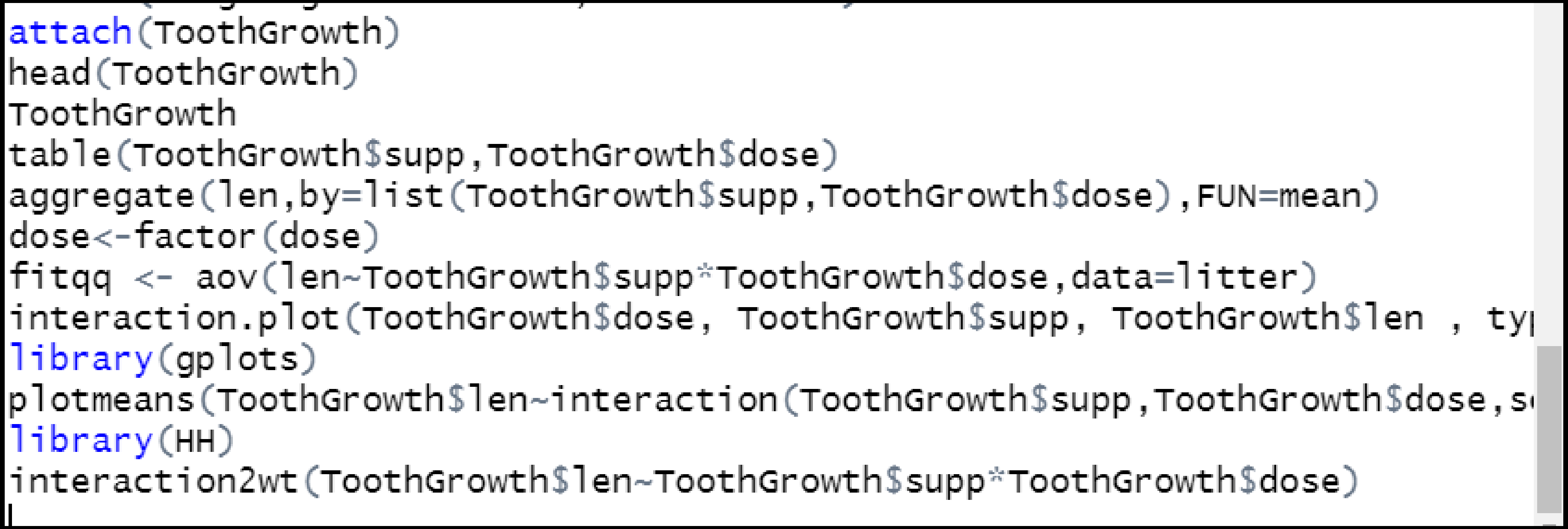
The graph includes the means, as well as error bars (95% confidence intervals) and sample sizes.

Figure 9.7 Interaction between dose and delivery mechanism on tooth growth. The mean plot with 95% confidence intervals was created by the plotmeans() function.

Two-way Factorial ANOVA



```
> interaction2wt(len~supp*dose)
Error in model.frame.default(x.formula, data, na.action = na.action) :
  variable lengths differ (found for 'dose')
```



```
attach(ToothGrowth)
head(ToothGrowth)
ToothGrowth
table(ToothGrowth$supp,ToothGrowth$dose)
aggregate(len,by=list(ToothGrowth$supp,ToothGrowth$dose),FUN=mean)
dose<-factor(dose)
fitqq <- aov(len~ToothGrowth$supp*ToothGrowth$dose,data=litter)
interaction.plot(ToothGrowth$dose, ToothGrowth$supp, ToothGrowth$len , ty
library(gplots)
plotmeans(ToothGrowth$len~interaction(ToothGrowth$supp,ToothGrowth$dose,s
library(HH)
interaction2wt(ToothGrowth$len~ToothGrowth$supp*ToothGrowth$dose)
```

Two-way Factorial ANOVA

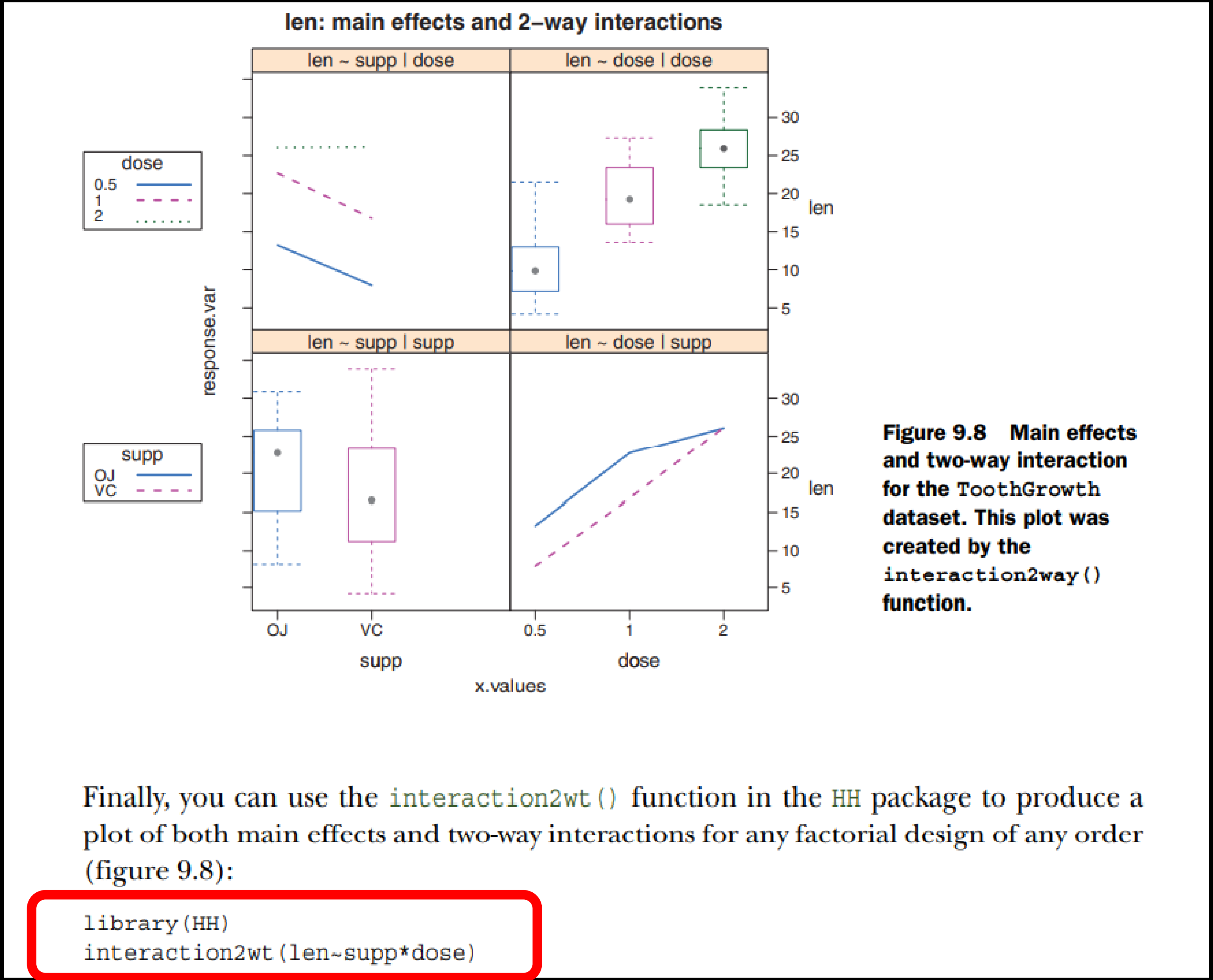

len: main effects and 2−way interactions
len ~ supp | dose
len ~ dose | dose
len ~ supp | supp
len ~ dose | supp
dose
0.5
1
2
supp
OJ
VC
response.var
len
len
5
10
15
20
25
30
5
10
15
20
25
30
OJ
VC
0.5
1
2
supp
dose
x.values

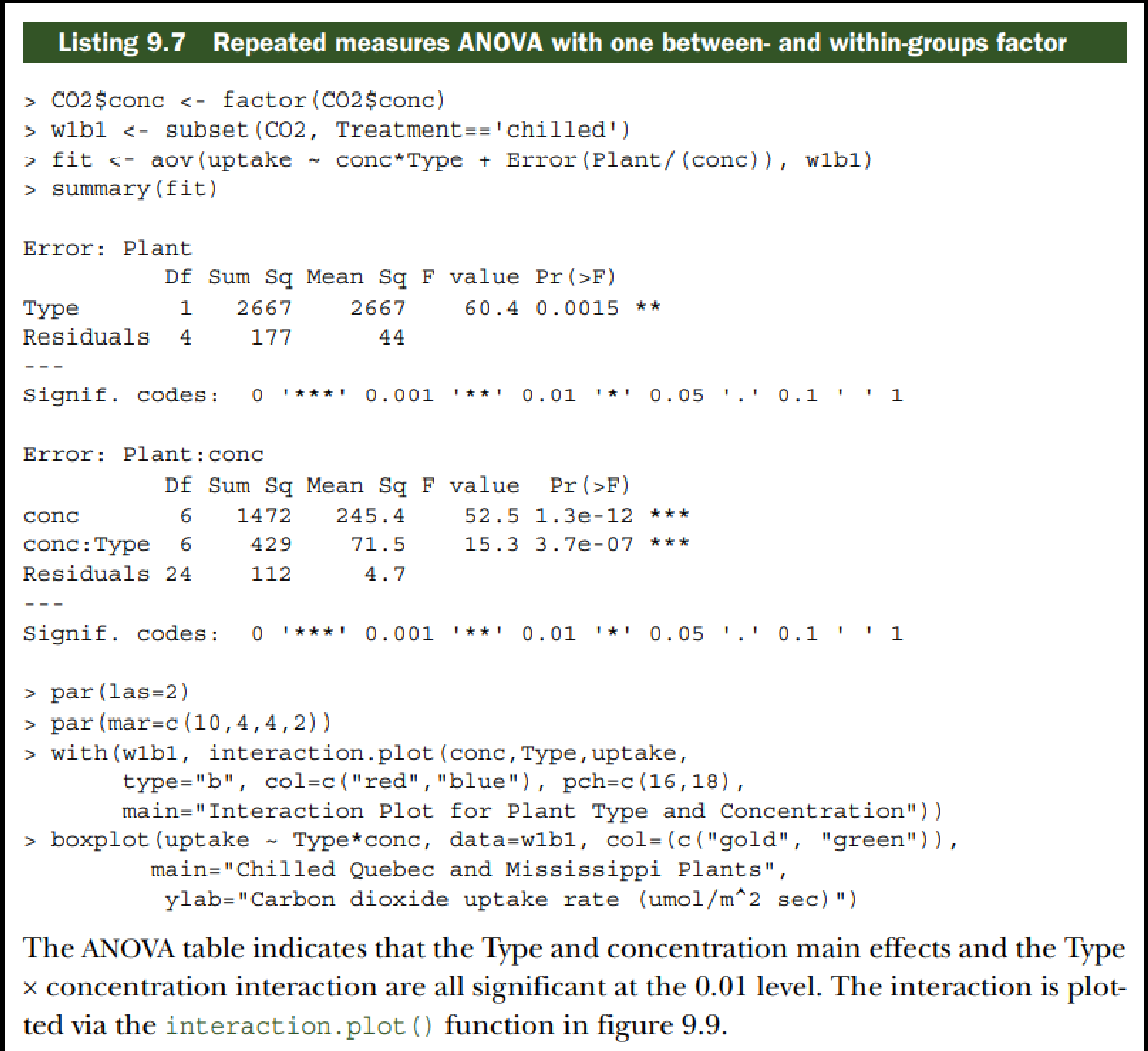
Figure 9.8 Main effects and two-way interaction for the `ToothGrowth` dataset. This plot was created by the `interaction2way()` function.

Finally, you can use the `interaction2wt()` function in the `HH` package to produce a plot of both main effects and two-way interactions for any factorial design of any order (figure 9.8):

```
library(HH)
interaction2wt(len~supp*dose)
```

Repeated Measures ANOVA

In repeated measures ANOVA, subjects are measured more than once. This section focuses on repeated measures ANOVA with one within-groups and one between-groups factor (a common design). We’ll take our example from the field of physiological ecology. Physiological ecologists study how the physiological and biochemical processes of living systems respond to variations in environmental factors (a crucial area of study given the realities of global warming). The *CO2* dataset included in the *base* installation contains the results of a study of cold tolerance in Northern and Southern plants of the grass species *Echinochloa* crus-galli (Potvin, Lechowicz, & Tardif, 1990). The photosynthetic rates of chilled plants were compared with the photosynthetic rates of nonchilled plants at several ambient *CO2* concentrations. Half the plants were from *Quebec*, and half were from *Mississippi*.



Listing 9.7 Repeated measures ANOVA with one between- and within-groups factor

```
> CO2$conc <- factor(CO2$conc)
> w1b1 <- subset(CO2, Treatment=='chilled')
> fit <- aov(uptake ~ conc*Type + Error(Plant/(conc)), w1b1)
> summary(fit)

Error: Plant
          Df Sum Sq Mean Sq F value Pr(>F)
Type       1   2667    2667    60.4 0.0015 **
Residuals  4    177      44
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Plant:conc
          Df Sum Sq Mean Sq F value  Pr(>F)
conc       6   1472   245.4    52.5 1.3e-12 ***
conc:Type  6    429    71.5    15.3 3.7e-07 ***
Residuals 24    112     4.7
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> par(las=2)
> par(mar=c(10,4,4,2))
> with(w1b1, interaction.plot(conc,Type,uptake,
      type="b", col=c("red","blue"), pch=c(16,18),
      main="Interaction Plot for Plant Type and Concentration"))
> boxplot(uptake ~ Type*conc, data=w1b1, col=(c("gold", "green")),
        main="Chilled Quebec and Mississippi Plants",
         ylab="Carbon dioxide uptake rate (umol/m^2 sec)")
```

The ANOVA table indicates that the Type and concentration main effects and the Type × concentration interaction are all significant at the 0.01 level. The interaction is plotted via the `interaction.plot()` function in figure 9.9.

Repeated Measures ANOVA

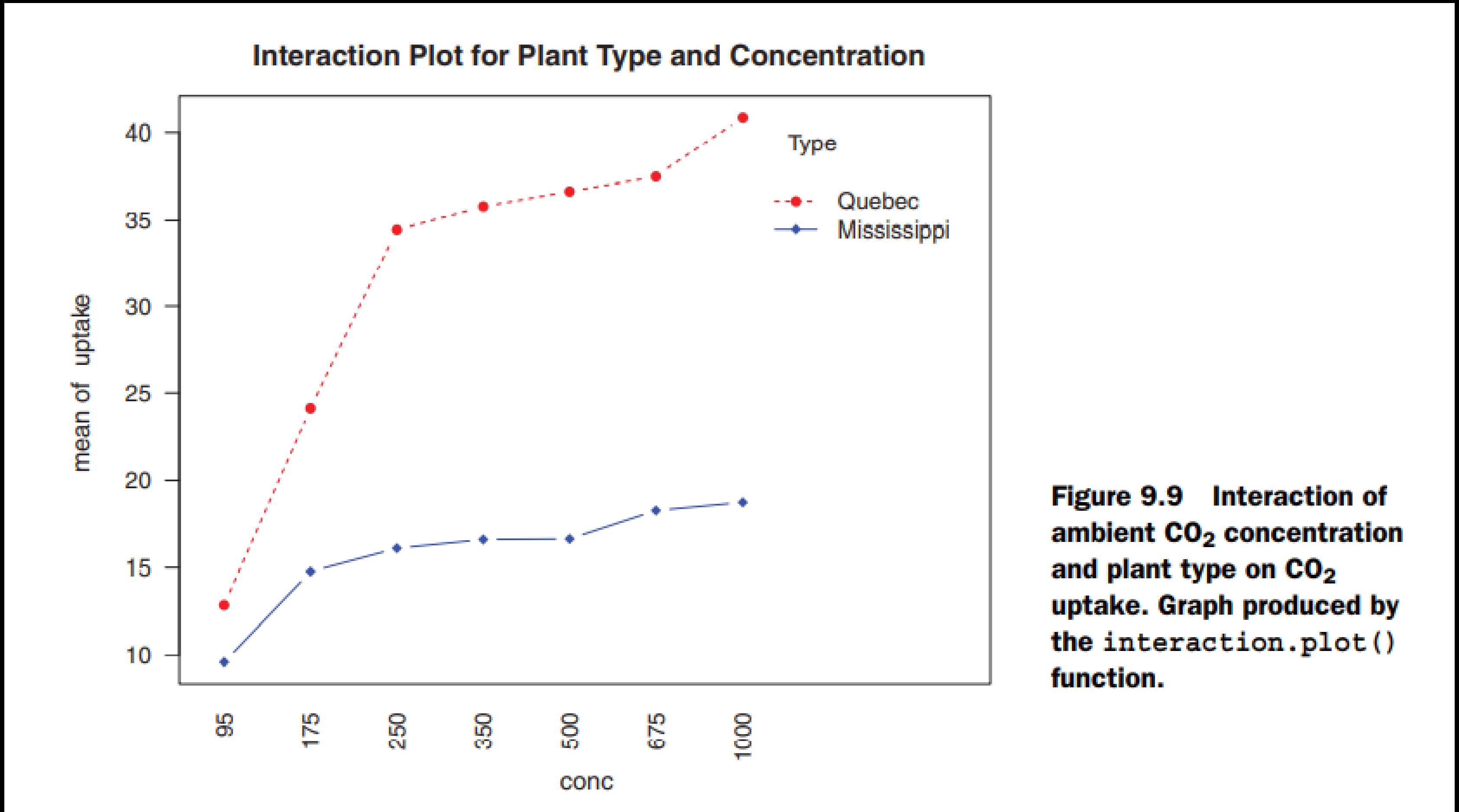
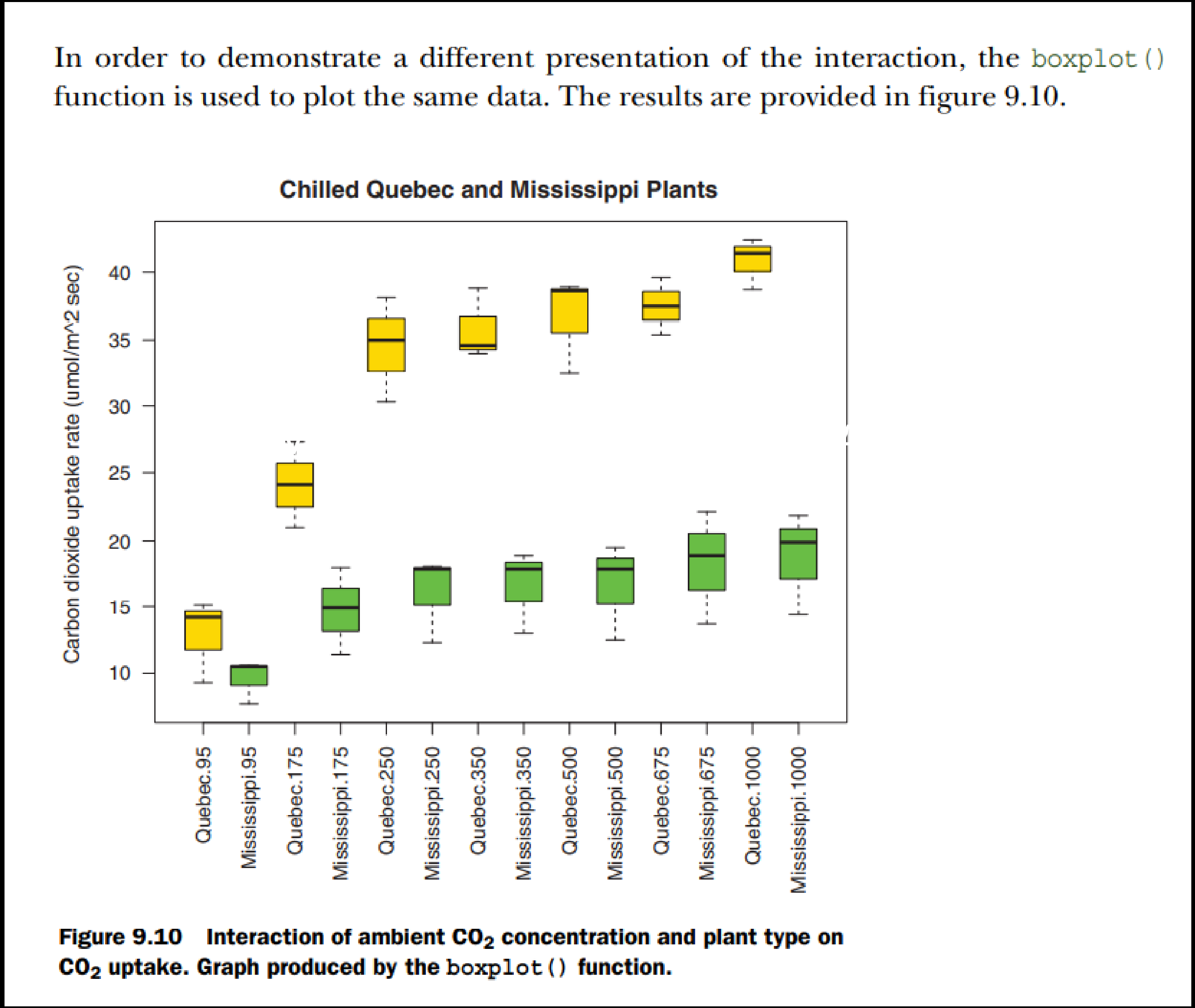


Figure 9.9 Interaction of ambient CO_2 concentration and plant type on CO_2 uptake. Graph produced by the `interaction.plot()` function.

Repeated Measures ANOVA



In order to demonstrate a different presentation of the interaction, the `boxplot()` function is used to plot the same data. The results are provided in figure 9.10.

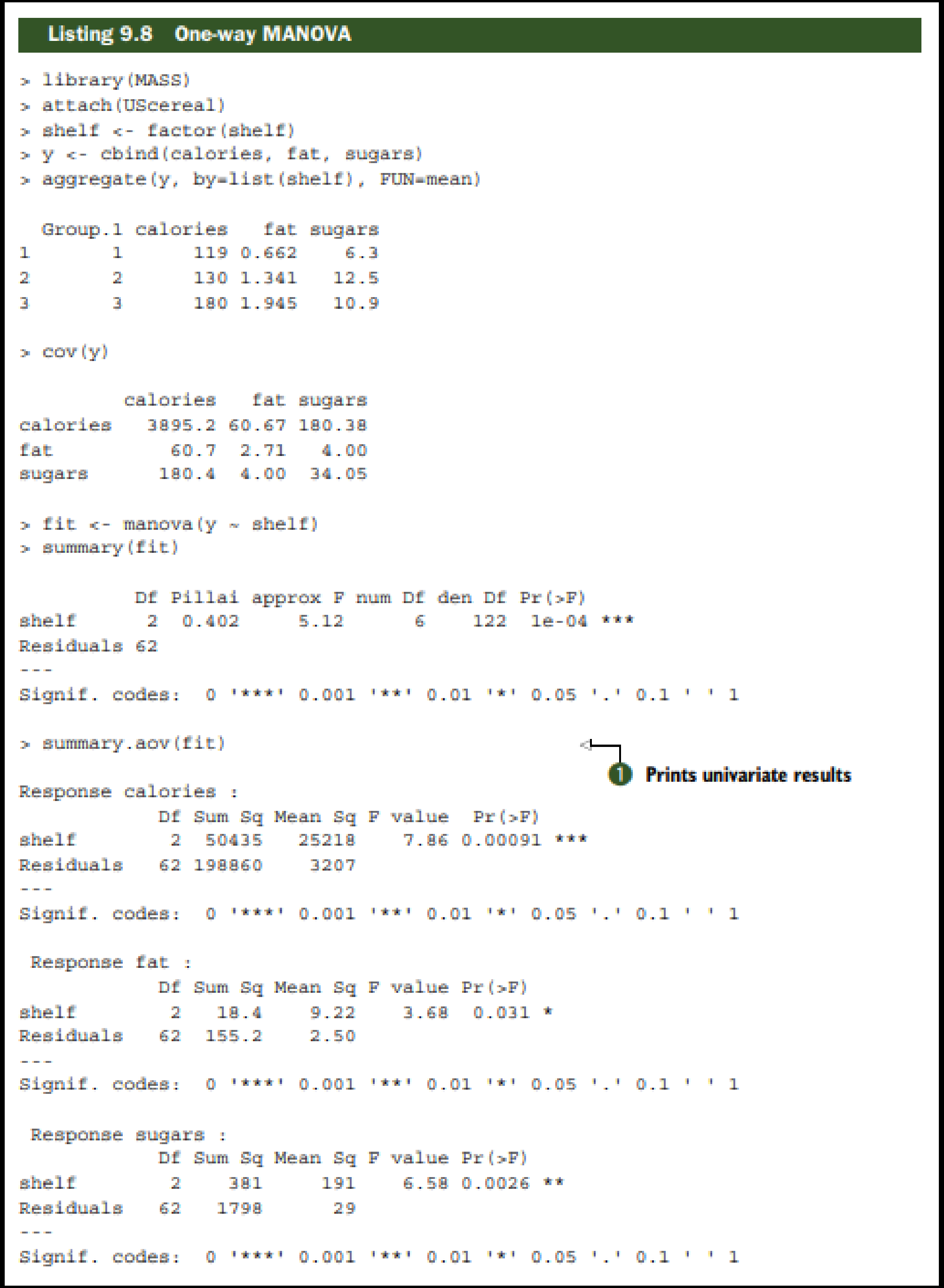
Figure 9.10 Interaction of ambient CO_2 concentration and plant type on CO_2 uptake. Graph produced by the `boxplot()` function.

Multivariate Analysis of Variance (MANOVA)

If there’s more than one dependent (outcome) variable, you can test them simultaneously using a multivariate analysis of variance (MANOVA). The following example is based on the UScereal dataset in the MASS package. The dataset comes from Venables & Ripley (1999). In this example, you’re interested in whether the calories, fat, and sugar content of US cereals vary by store shelf, where 1 is the bottom shelf, 2 is the middle shelf, and 3 is the top shelf. Calories, fat, and sugars are the dependent variables, and shelf is the independent variable, with three levels (1, 2, and 3). The analysis is presented in the following listing.

Multivariate Analysis of Variance (MANOVA)

First, the shelf variable is converted to a factor so that it can represent a grouping variable in the analyses. Next, the *cbind*() function is used to form a matrix of the three dependent variables (calories, fat, and sugars). The *aggregate*() function provides the shelf means, and the *cov*() function provides the variance and the covariances across cereals. The *manova*() function provides the multivariate test of group differences. The significant F value indicates that the three groups differ on the set of nutritional measures. Note that the shelf variable was converted to a factor so that it can represent a grouping variable.

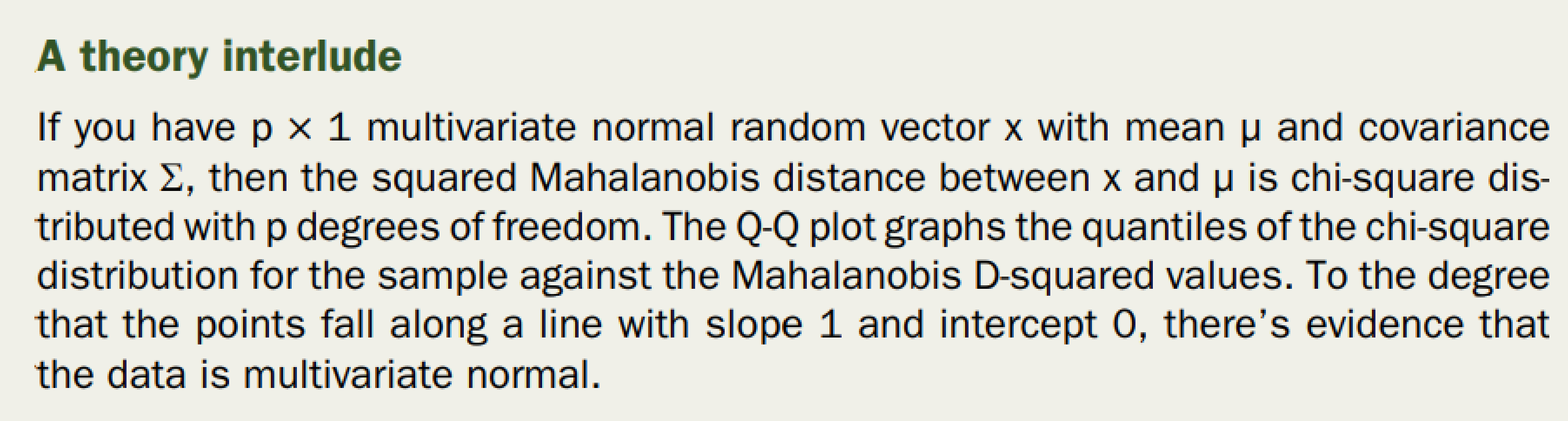


Listing 9.8 One-way MANOVA

```
> library(MASS)
> attach(UScereal)
> shelf <- factor(shelf)
> y <- cbind(calories, fat, sugars)
> aggregate(y, by=list(shelf), FUN=mean)

  Group.1 calories   fat sugars
1       1      119 0.662    6.3
2       2      130 1.341   12.5
3       3      180 1.945   10.9

> cov(y)

         calories   fat sugars
calories   3895.2 60.67 180.38
fat          60.7  2.71   4.00
sugars      180.4  4.00  34.05

> fit <- manova(y ~ shelf)
> summary(fit)

          Df Pillai approx F num Df den Df Pr(>F)
shelf      2  0.402     5.12      6    122  1e-04 ***
Residuals 62
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary.aov(fit)

Response calories :
            Df Sum Sq Mean Sq F value  Pr(>F)
shelf        2  50435   25218    7.86 0.00091 ***
Residuals   62 198860    3207
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

 Response fat :
            Df Sum Sq Mean Sq F value Pr(>F)
shelf        2   18.4    9.22    3.68  0.031 *
Residuals   62  155.2    2.50
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

 Response sugars :
            Df Sum Sq Mean Sq F value Pr(>F)
shelf        2    381     191    6.58 0.0026 **
Residuals   62   1798      29
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Assessing Test Assumptions

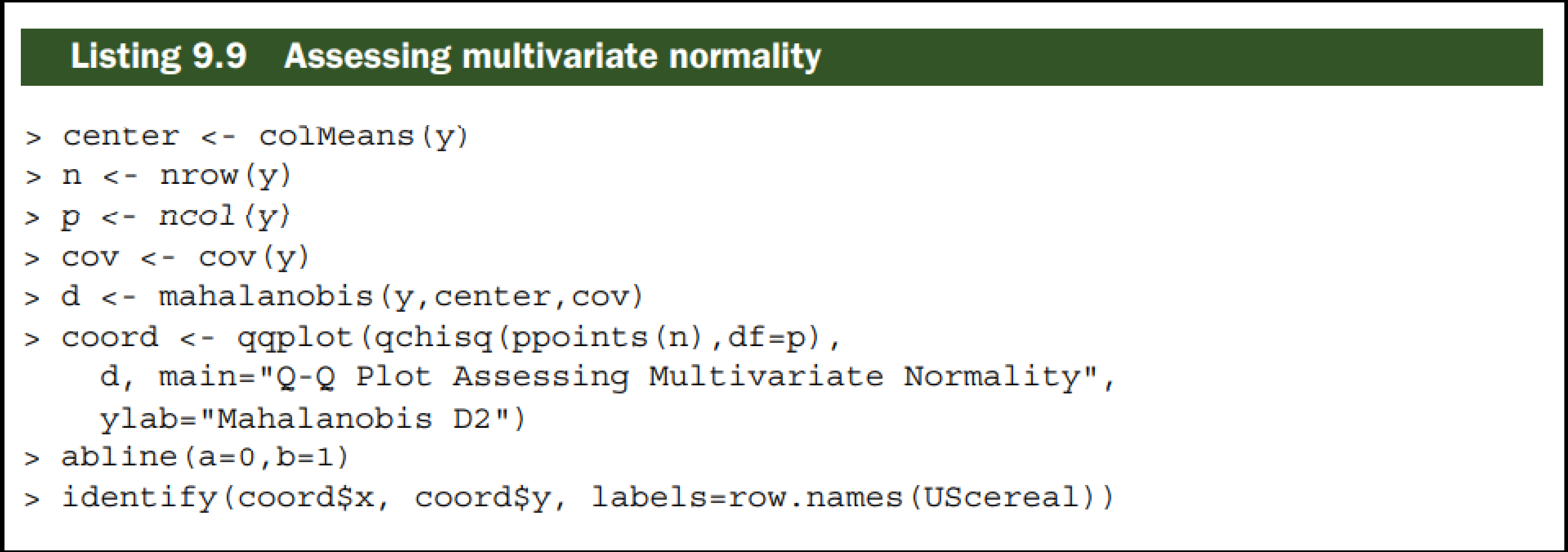
The two assumptions underlying a one-way MANOVA are multivariate normality and homogeneity of variance-covariance matrices. The first assumption states that the vector of dependent variables jointly follows a multivariate normal distribution. You can use a Q-Q plot to assess this assumption (see the sidebar “A theory interlude” for a statistical explanation of how this works).



A theory interlude

If you have p × 1 multivariate normal random vector x with mean μ and covariance matrix Σ, then the squared Mahalanobis distance between x and μ is chi-square distributed with p degrees of freedom. The Q-Q plot graphs the quantiles of the chi-square distribution for the sample against the Mahalanobis D-squared values. To the degree that the points fall along a line with slope 1 and intercept 0, there’s evidence that the data is multivariate normal.

Assessing Test Assumptions



Listing 9.9 Assessing multivariate normality

```
> center <- colMeans(y)
> n <- nrow(y)
> p <- ncol(y)
> cov <- cov(y)
> d <- mahalanobis(y,center,cov)
> coord <- qqplot(qchisq(ppoints(n),df=p),
    d, main="Q-Q Plot Assessing Multivariate Normality",
    ylab="Mahalanobis D2")
> abline(a=0,b=1)
> identify(coord$x, coord$y, labels=row.names(UScereal))
```

Assessing Test Assumptions

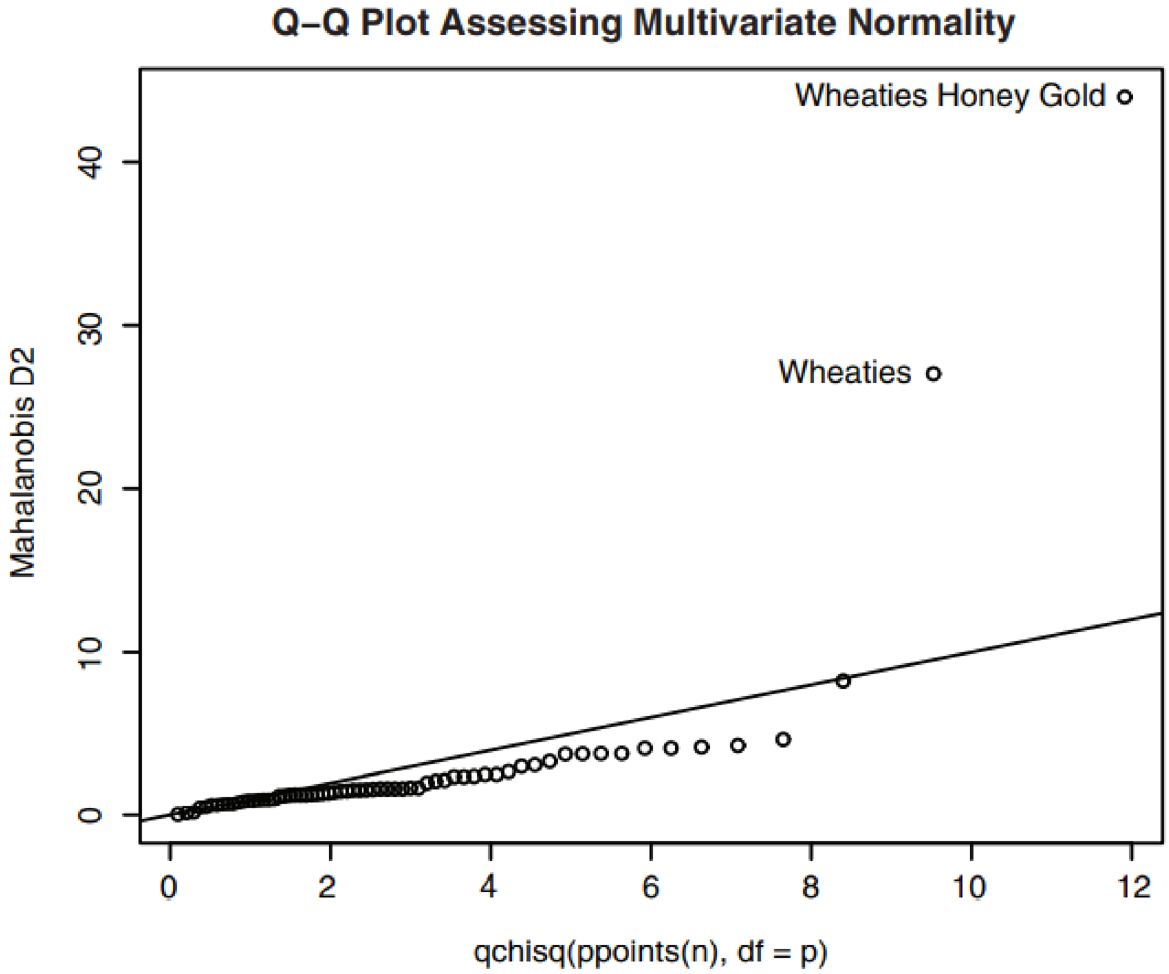
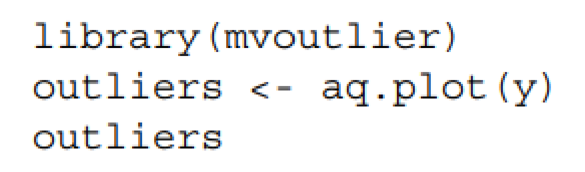


Figure 9.11 A Q-Q plot for assessing multivariate normality

Assessing Test Assumptions



```
library(mvoutlier)
outliers <- aq.plot(y)
outliers
```

If the data follow a multivariate normal distribution, then points will fall on the line. The *identify*() function allows you to interactively identify points in the graph. (The *identify*() function is covered in section 16.4.) Here, the dataset appears to violate multivariate normality, primarily due to the observations for Wheaties Honey Gold and Wheaties.

You may want to delete these two cases and rerun the analyses.

The homogeneity of variance-covariance matrices assumption requires that the covariance matrix for each group is equal. The assumption is usually evaluated with a Box’s M test. R doesn’t include a function for Box’s M, but an internet search will provide the appropriate code.

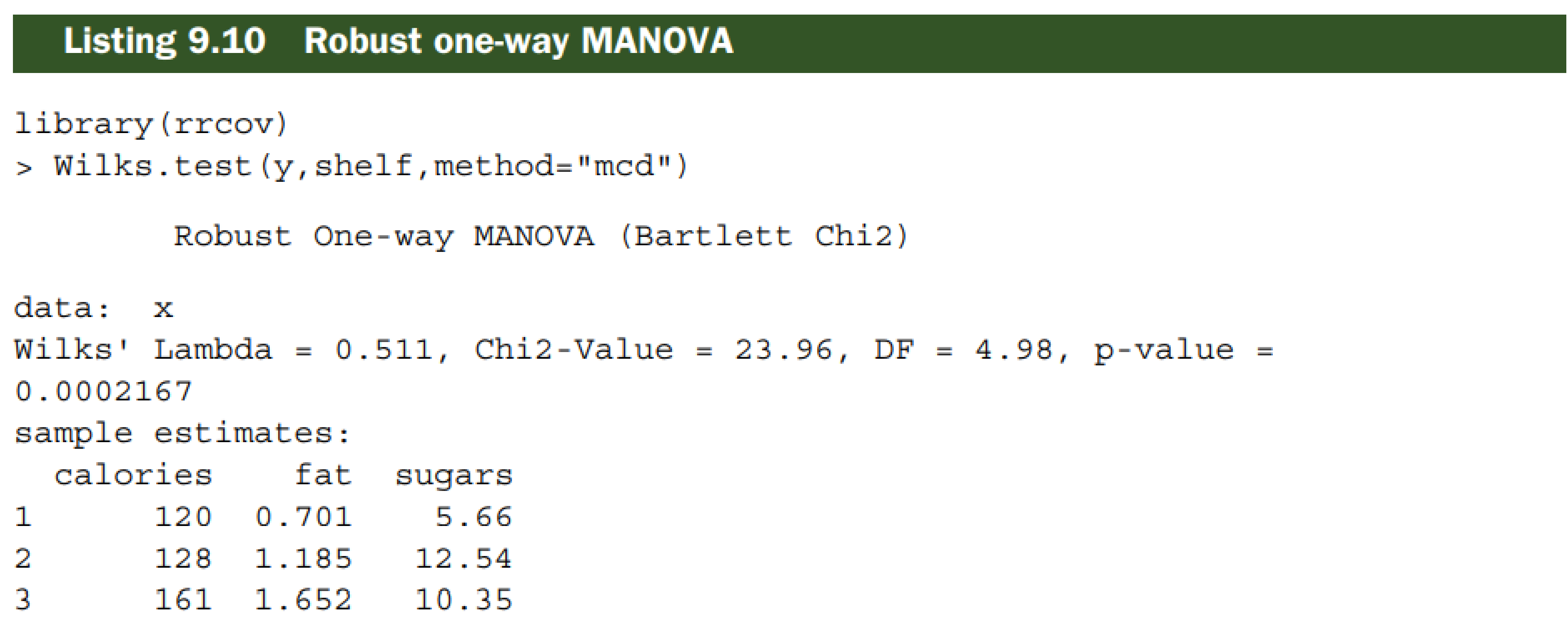
Unfortunately, the test is sensitive to violations of normality, leading to rejection in most typical cases. This means that we don’t yet have a good working method for evaluating this important assumption (but see Anderson [2006] and Silva et al. [2008] for interesting alternative approaches not yet available in R). Finally, you can test for multivariate outliers using the aq.plot() function in the mvoutlier package. The code in this case looks like this:

Robust MANOVA

If the assumptions of multivariate normality or homogeneity of variance-covariance matrices are untenable, or if you’re concerned about multivariate outliers, you may want to consider using a robust or nonparametric version of the MANOVA test instead. A robust version of the one-way MANOVA is provided by the *Wilks.test*() function in next slide.

From the results, you can see that using a robust test that’s insensitive to both outliers and violations of MANOVA assumptions still indicates that the cereals on the top, middle, and bottom store shelves differ in their nutritional profiles.

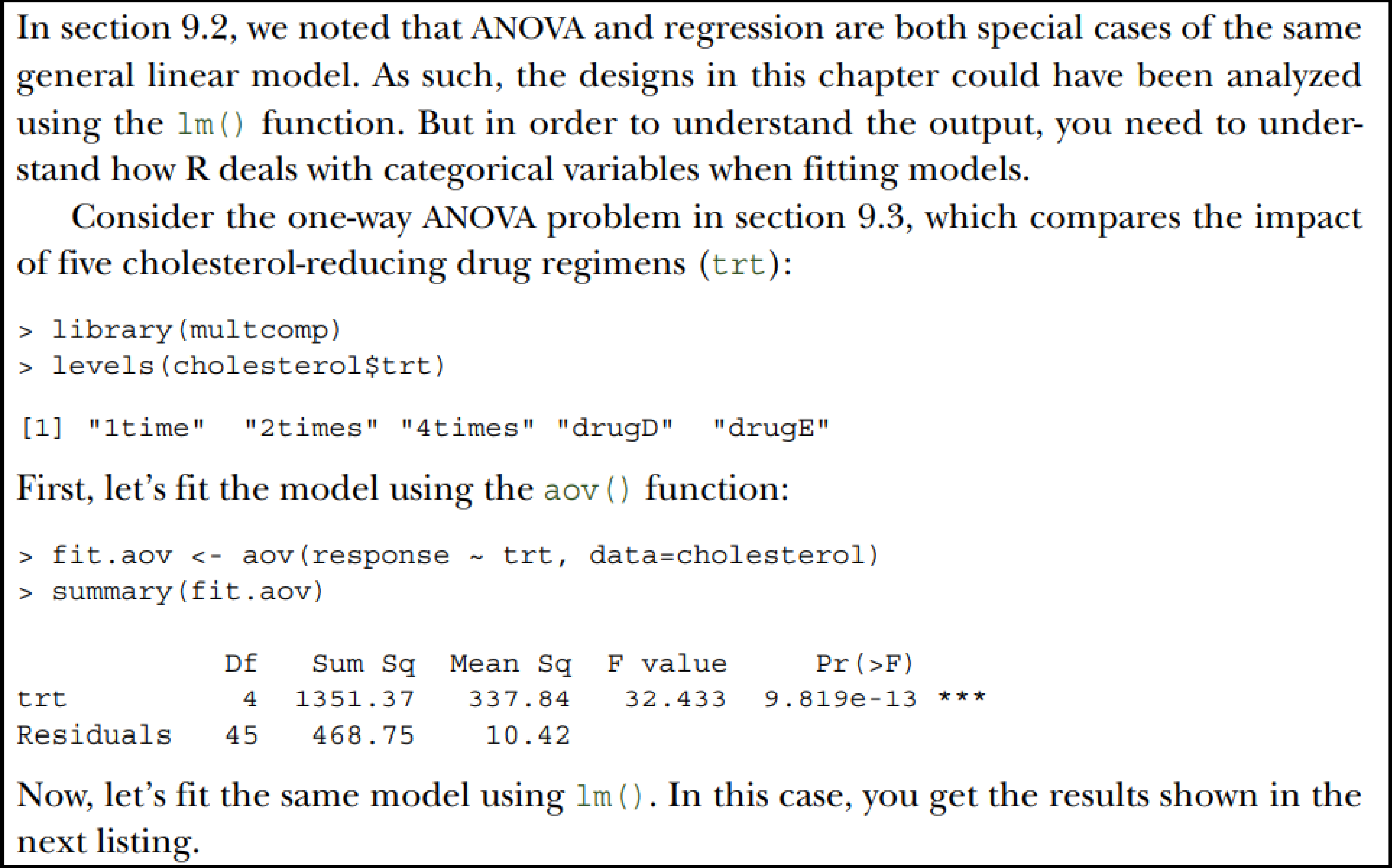
Robust MANOVA



Listing 9.10 Robust one-way MANOVA

```
library(rrcov)
> Wilks.test(y,shelf,method="mcd")

        Robust One-way MANOVA (Bartlett Chi2)

data:  x
Wilks' Lambda = 0.511, Chi2-Value = 23.96, DF = 4.98, p-value =
0.0002167
sample estimates:
  calories   fat  sugars
1      120 0.701    5.66
2      128 1.185   12.54
3      161 1.652   10.35
```

ANOVA as Regression



In section 9.2, we noted that ANOVA and regression are both special cases of the same general linear model. As such, the designs in this chapter could have been analyzed using the `lm()` function. But in order to understand the output, you need to understand how R deals with categorical variables when fitting models.

Consider the one-way ANOVA problem in section 9.3, which compares the impact of five cholesterol-reducing drug regimens (`trt`):

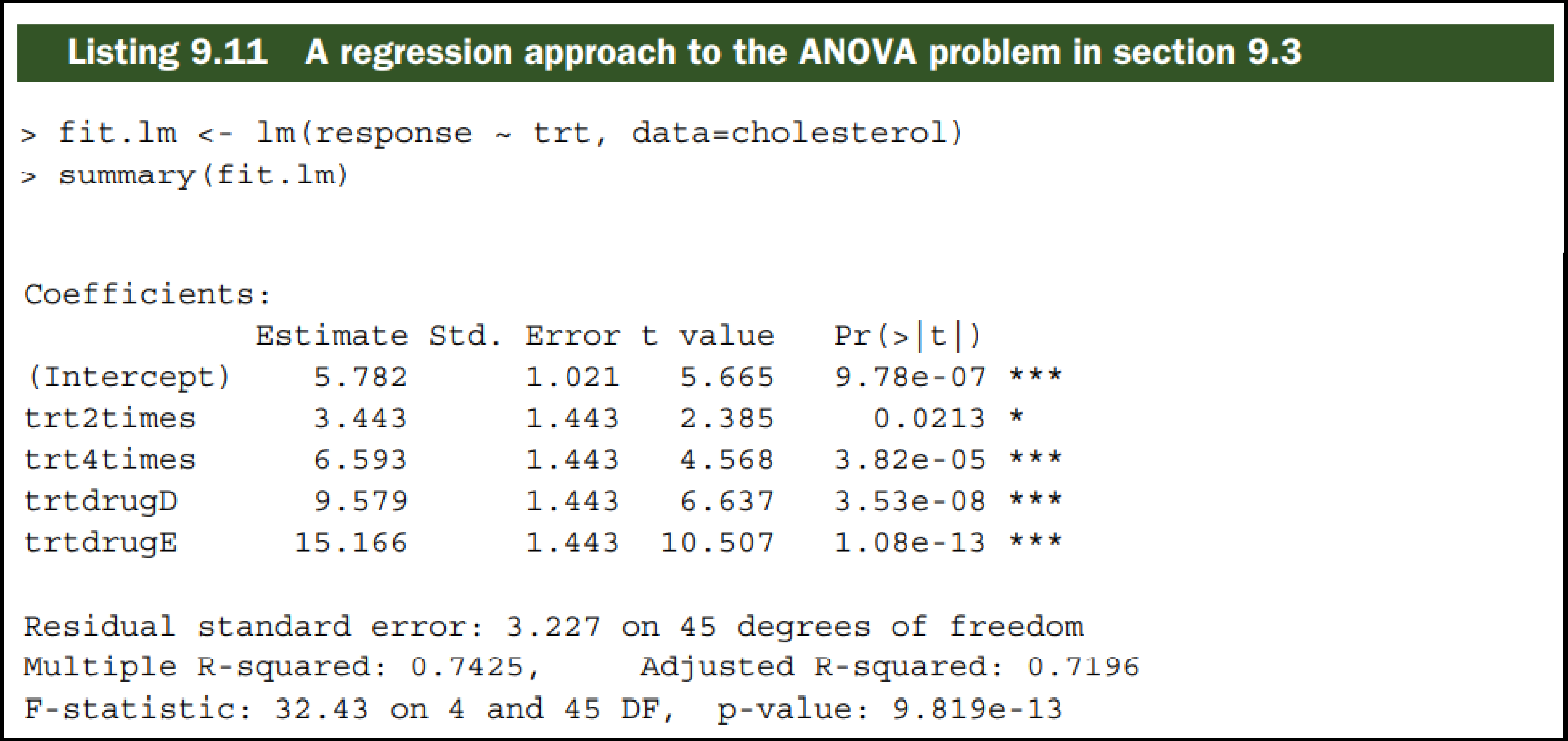
```
> library(multcomp)
> levels(cholesterol$trt)

[1] "1time"  "2times" "4times" "drugD"  "drugE"
```

First, let's fit the model using the `aov()` function:

```
> fit.aov <- aov(response ~ trt, data=cholesterol)
> summary(fit.aov)

            Df  Sum Sq  Mean Sq  F value     Pr(>F)
trt          4 1351.37   337.84   32.433  9.819e-13 ***
Residuals   45  468.75    10.42
```

Now, let's fit the same model using `lm()`. In this case, you get the results shown in the next listing.

ANOVA as Regression



Listing 9.11 A regression approach to the ANOVA problem in section 9.3

```
> fit.lm <- lm(response ~ trt, data=cholesterol)
> summary(fit.lm)


Coefficients:
            Estimate Std. Error t value   Pr(>|t|)
(Intercept)    5.782      1.021   5.665   9.78e-07 ***
trt2times      3.443      1.443   2.385     0.0213 *
trt4times      6.593      1.443   4.568   3.82e-05 ***
trtdrugD       9.579      1.443   6.637   3.53e-08 ***
trtdrugE      15.166      1.443  10.507   1.08e-13 ***

Residual standard error: 3.227 on 45 degrees of freedom
Multiple R-squared: 0.7425,     Adjusted R-squared: 0.7196
F-statistic: 32.43 on 4 and 45 DF,  p-value: 9.819e-13
```

ANOVA as Regression

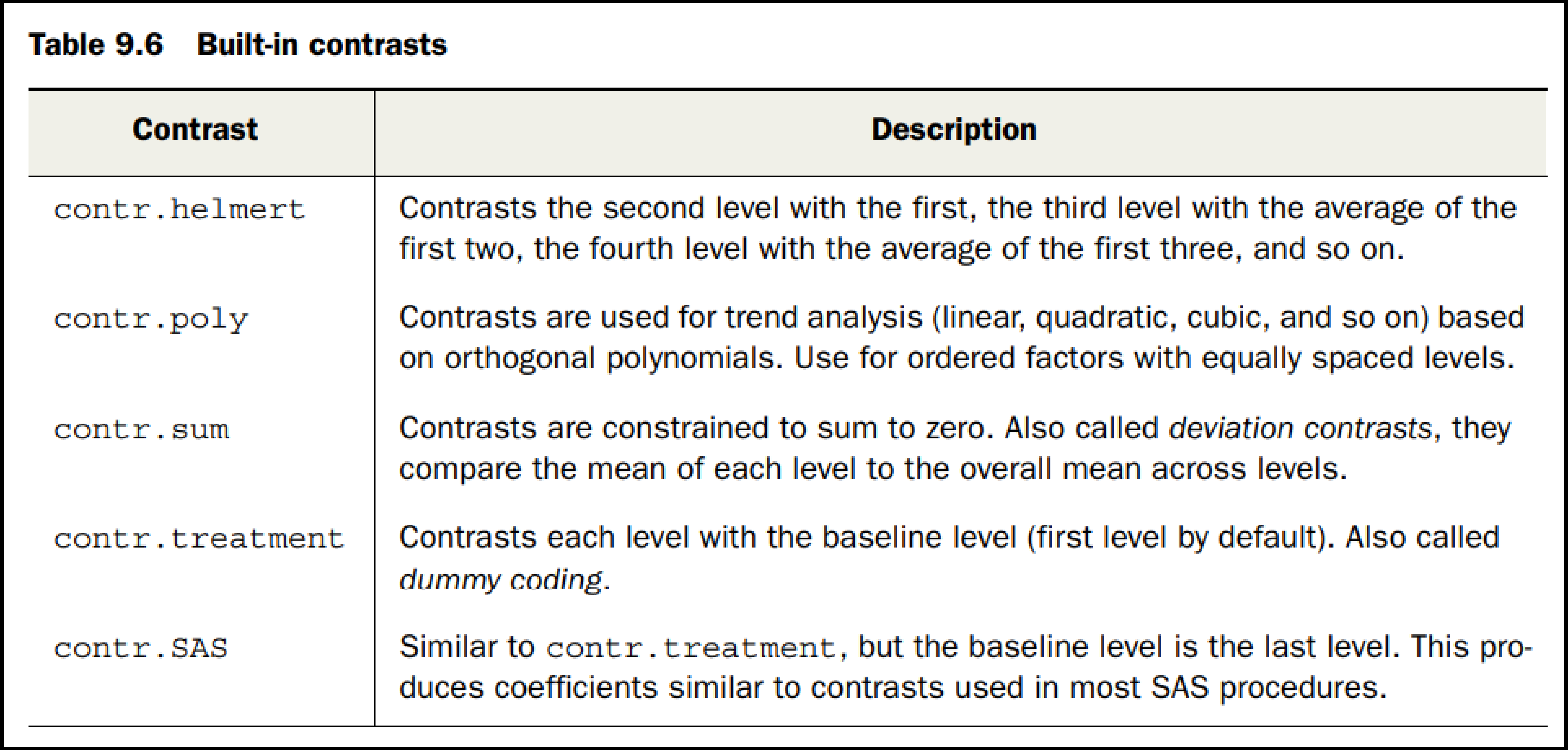
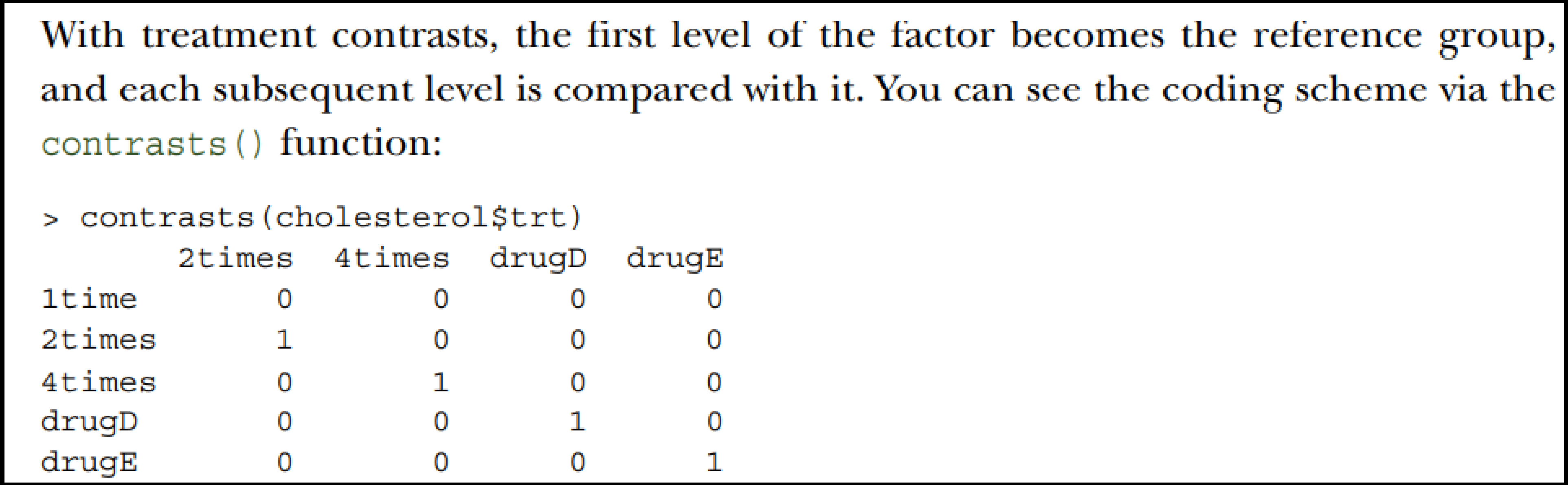


Table 9.6 Built-in contrasts

Contrast	Description
`contr.helmert`	Contrasts the second level with the first, the third level with the average of the first two, the fourth level with the average of the first three, and so on.
`contr.poly`	Contrasts are used for trend analysis (linear, quadratic, cubic, and so on) based on orthogonal polynomials. Use for ordered factors with equally spaced levels.
`contr.sum`	Contrasts are constrained to sum to zero. Also called *deviation contrasts*, they compare the mean of each level to the overall mean across levels.
`contr.treatment`	Contrasts each level with the baseline level (first level by default). Also called *dummy coding*.
`contr.SAS`	Similar to `contr.treatment`, but the baseline level is the last level. This produces coefficients similar to contrasts used in most SAS procedures.

ANOVA as Regression

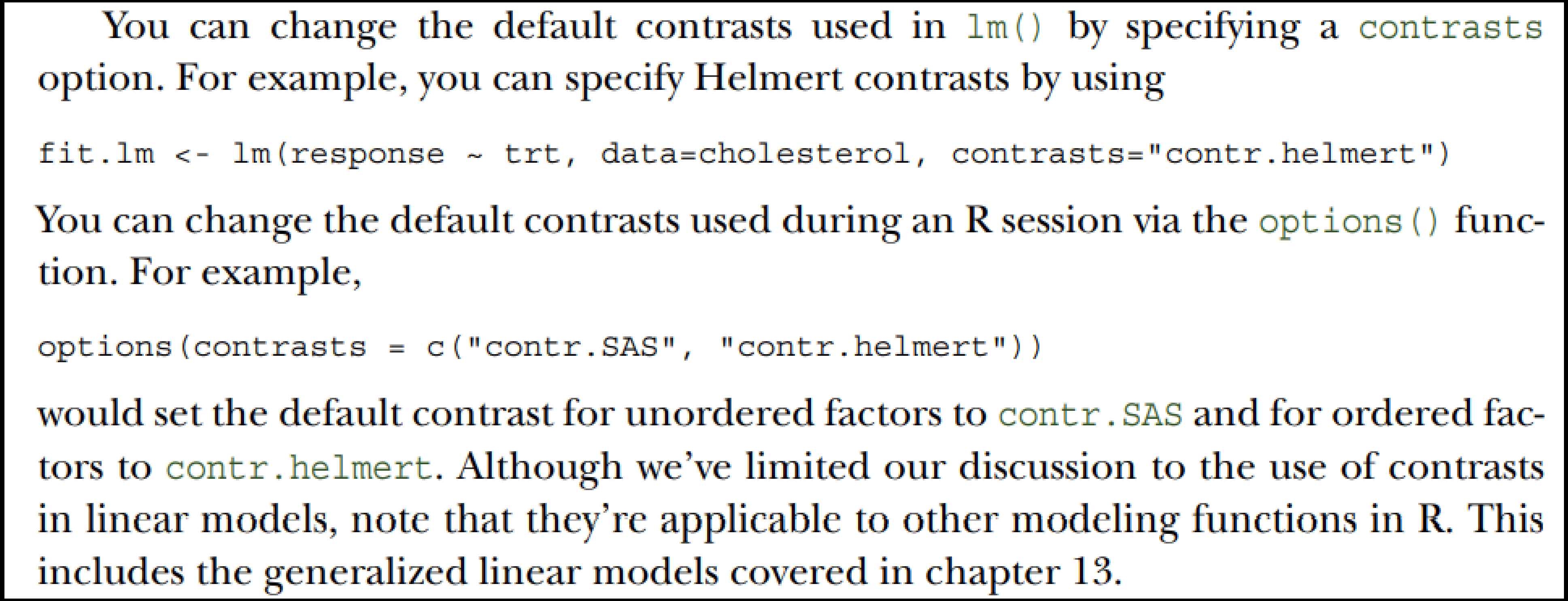


With treatment contrasts, the first level of the factor becomes the reference group, and each subsequent level is compared with it. You can see the coding scheme via the `contrasts()` function:

```
> contrasts(cholesterol$trt)
       2times  4times  drugD  drugE
1time       0       0      0      0
2times      1       0      0      0
4times      0       1      0      0
drugD       0       0      1      0
drugE       0       0      0      1
```

If a patient is in the *drugD* condition, then the variable drugD equals 1, and the variables 2times, 4times, and *drugE* each equal zero. You don’t need a variable for the first group, because a zero on each of the four indicator variables uniquely determines that the patient is in the 1times condition. In listing 9.11, the variable trt2times represents a contrast between the levels 1time and 2time. Similarly, trt4times is a contrast between 1time and 4times, and so on. You can see from the probability values in the output that each drug condition is significantly different from the first (1time)

ANOVA as Regression



You can change the default contrasts used in `lm()` by specifying a `contrasts` option. For example, you can specify Helmert contrasts by using

```
fit.lm <- lm(response ~ trt, data=cholesterol, contrasts="contr.helmert")
```

You can change the default contrasts used during an R session via the `options()` function. For example,

```
options(contrasts = c("contr.SAS", "contr.helmert"))
```

would set the default contrast for unordered factors to `contr.SAS` and for ordered factors to `contr.helmert`. Although we've limited our discussion to the use of contrasts in linear models, note that they're applicable to other modeling functions in R. This includes the generalized linear models covered in chapter 13.

References

- R in Action, R. Kabacoff, 2nd edition, Manning, ISBN 978-1-617-29138-8, Chapter 9.