

Homework 3 Solutions

BY MAREK RYCHLIK

Copyright © Marek Rychlik, 2009
March 26, 2009

Problem 3

Reading/displaying data

Create a temporary file, so that we do not have to rely upon the knowledge of the paths of the underlying filesystem.

```
> file = tempfile() # Creates a temporary file
```

```
> cat("Premolt Fasting Bran60g Bran80g LayingMash  
94.09 98.81 197.18 102.93 83.14  
90.45 103.55 207.31 117.51 89.59  
99.38 115.23 177.50 119.92 87.76  
73.56 129.06 226.05 112.01 96.43  
74.39 117.61 222.74 101.10 82.94", file=file)
```

```
>
```

Then, we read the data back. We note that the data is in a “wide” format, i.e. the factor levels are used as column labels.

```
> (data = read.table(file, header=T))
```

	Premolt	Fasting	Bran60g	Bran80g	LayingMash
1	94.09	98.81	197.18	102.93	83.14
2	90.45	103.55	207.31	117.51	89.59
3	99.38	115.23	177.50	119.92	87.76
4	73.56	129.06	226.05	112.01	96.43
5	74.39	117.61	222.74	101.10	82.94

```
>
```

Computing analysis of variance

Before we compute analysis of variance, we transform the data to the standard model form (the “long” form). The transformation is done easily using the `stack` function. The more sophisticated function `reshape` could be possibly used, but it requires full understanding of what it does. We settled for `stack`.

```
> data.stacked = stack(data)
```

```
> colnames(data.stacked) = c("Serum T3", "Treatment")
```

```
> data.stacked
```

	Serum T3	Treatment
1	94.09	Premolt
2	90.45	Premolt
3	99.38	Premolt
4	73.56	Premolt
5	74.39	Premolt
6	98.81	Fasting
7	103.55	Fasting
8	115.23	Fasting
9	129.06	Fasting
10	117.61	Fasting
11	197.18	Bran60g
12	207.31	Bran60g
13	177.50	Bran60g
14	226.05	Bran60g
15	222.74	Bran60g
16	102.93	Bran80g
17	117.51	Bran80g
18	119.92	Bran80g

```

19  112.01  Bran80g
20  101.10  Bran80g
21   83.14  LayingMash
22   89.59  LayingMash
23   87.76  LayingMash
24   96.43  LayingMash
25   82.94  LayingMash

```

```
> data.aov = aov('Serum T3' ~ Treatment, data.stacked)
```

```
> summary(data.aov)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Treatment	4	48569	12142	78.08	6.482e-12 ***
Residuals	20	3110	156		

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

```
>
```

Calculating the means

```
> (mt = model.tables(data.aov,type="means",se=T))
```

```
Tables of means
```

```
Grand mean
```

```
120.8096
```

```
Treatment
```

```
Treatment
```

Bran60g	Bran80g	Fasting	LayingMash	Premolt
206.16	110.69	112.85	87.97	86.37

```
Standard errors for differences of means
```

```
Treatment
```

```
replic.      7.887
            5
```

```
>
```

Picking model tables apart

The following illustrates picking apart a typical R data structure. As usual, functions like `names` and `attributes` are helpful.

```
> names(mt)
```

```
[1] "tables" "n"      "se"
```

```
> mt$n
```

```
Treatment
      5
```

```
> mt$tables
```

```
$'Grand mean'
```

```
[1] 120.8096
```

```
$Treatment
```

```
Treatment
```

Bran60g	Bran80g	Fasting	LayingMash	Premolt
206.156	110.694	112.852	87.972	86.374

```
> mt$tables$`Grand mean`
```

```
[1] 120.8096
```

```
> mt$tables$Treatment["Bran60g"]
```

```
Bran60g
```

```
206.156
```

```
>
```

Confidence intervals for the means

According to the built-in function `confint`, which is concerned with **effects** rather than means:


```
> confint(data.aov, level = 0.95, type="means")
```

	2.5 %	97.5 %
(Intercept)	194.5228	217.78922
TreatmentBran80g	-111.9139	-79.01014
TreatmentFasting	-109.7559	-76.85214
TreatmentLayingMash	-134.6359	-101.73214
TreatmentPremolt	-136.2339	-103.33014

```
>
```

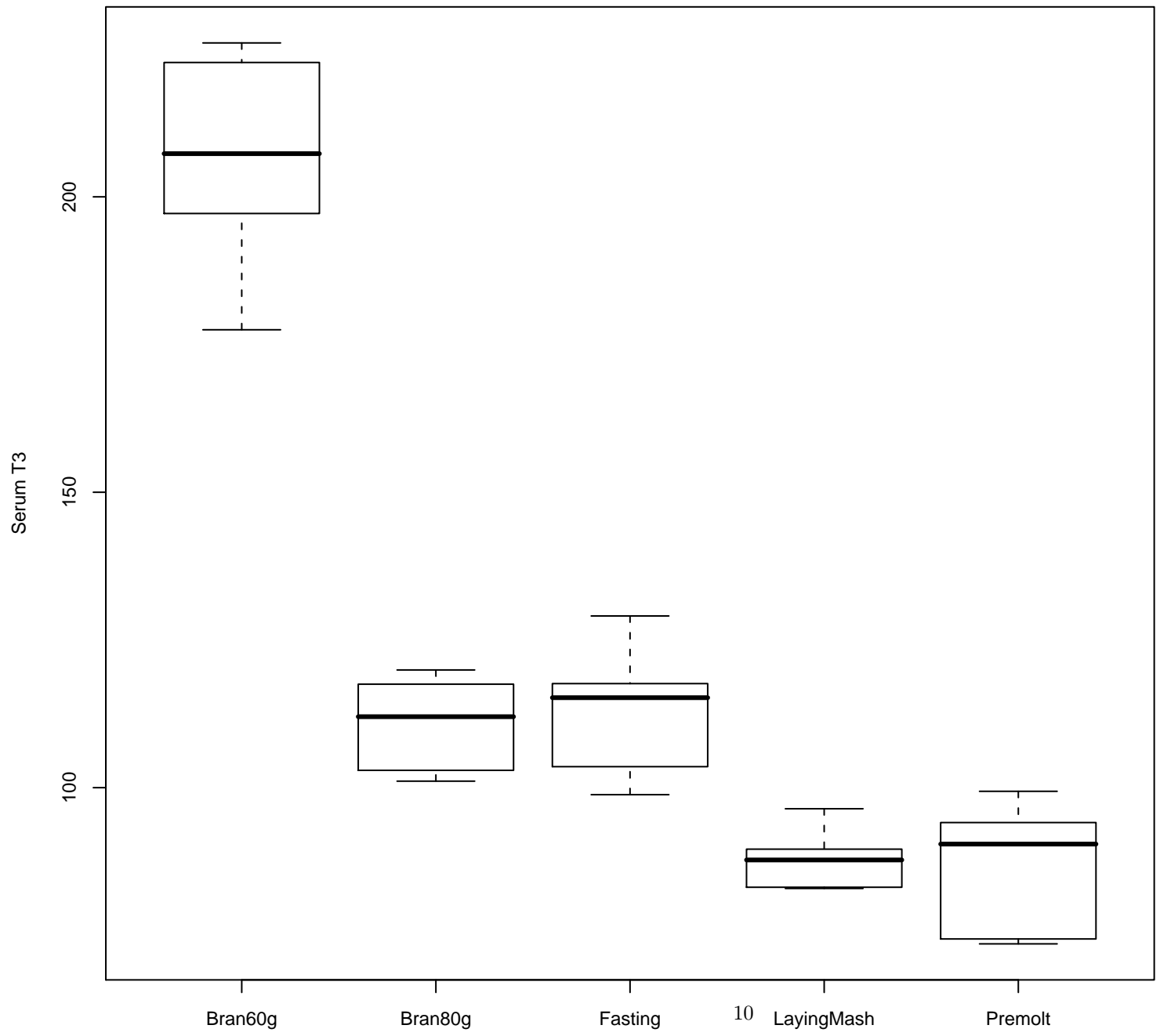
In order to get confidence intervals as in the book, we need to use the formulas directly.

Plots

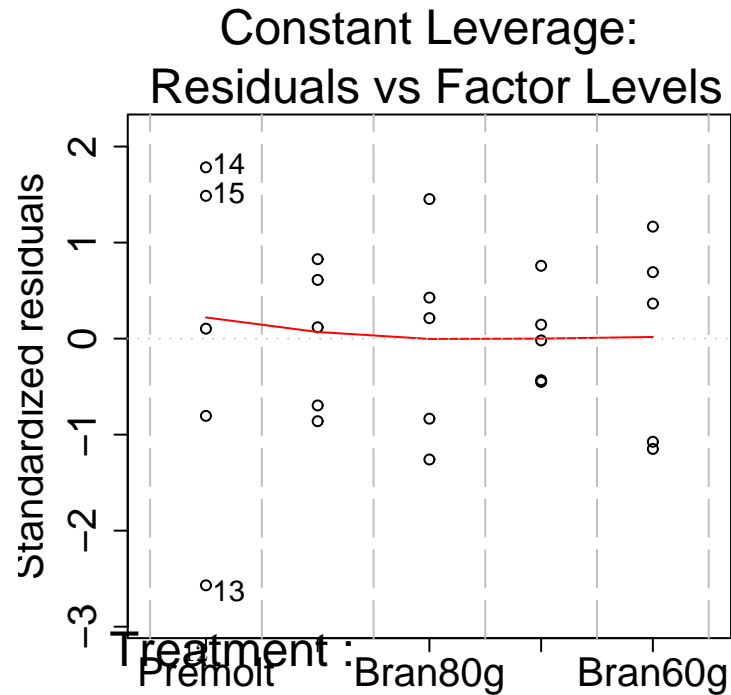
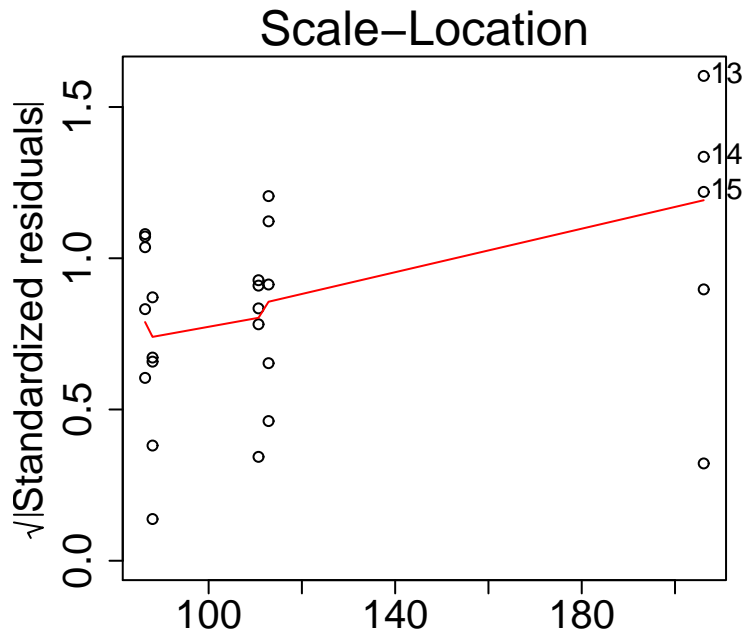
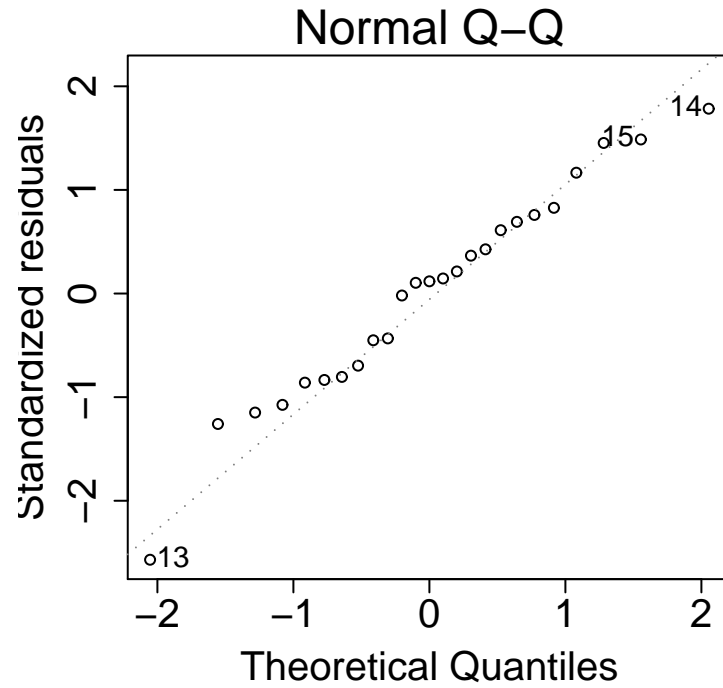
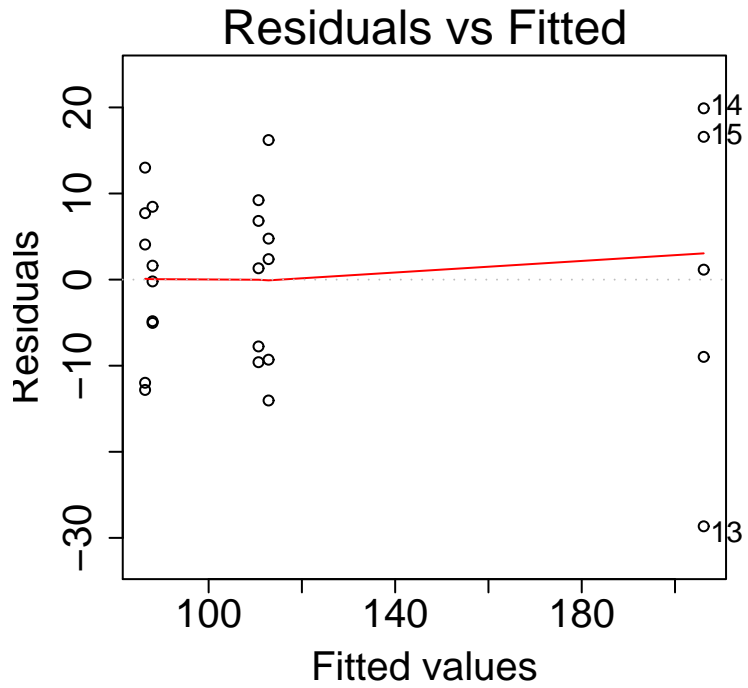
```
> opts=options(); opts$texmacs$width=8; opts$texmacs$height=8;  
opts$nox11=F; options(opts)
```

```
> X11(pointsize=6, width=3, height=3)
```

```
> plot('Serum T3' ~ Treatment, data.stacked); v()
```



```
> par(mfrow=c(2,2)); plot(data.aov);v()
```



>