Solution to Exercise 1 in Chapter 3

Marek Rychlik

March 23, 2009

Statement of the problem

Use the data on traffic delay in Excercise 2.1.

a) Conduct an analysis of variance for the data, and estimate the folliwing contrasts and their standard errors:

   i. a contrast betwee the pretimed and the average of the semi- and fully actuated signals

   ii. a contrast between the sem- and fully actuated signals

b) Compute the sum of squares of each contrast, and show that their sum is equal to the treatment sum of squares in the analysis of variance.

c) Test the null hypothesis for each contrast, H_0: $C = 0$, with the student t test at the 0.05 level of significance.

d) Test the null hypothesis in part (c) with the F test at the 0.05 level of significance.

e) What is the relathionship between the two tests in parts (c) and (d)?

Analysis of variance using R

Entering data and contrasts

We decided to enter the small amount of data as code instead of relying upon an external source of data such as a text file. We used the function `gl` to set up the factor (the traffic light type). We asssigned the contrasts directly to the factor, as indicated by part (b) of the exercise.

```
> Delay = c(36.6, 39.2, 30.4, 37.1, 34.1, 17.5, 20.6, 18.7, 25.7,
  22.0, 15.0, 10.4, 18.9, 10.5, 15.2)
```

```
> Type = gl(3, 5, labels = c("Pretimed", "Semi-Actuated", "Fully-
  Actuated"))
```

```
> contrasts(Type) = cbind(PretimedVsActuated=c(2,-1,-1),
  FullyVsSemi=c(0,1,-1))
```

```
> traffic.data = data.frame(Type, Delay)
```

```
> traffic.data
```

```
             Type Delay
1        Pretimed  36.6
2        Pretimed  39.2
3        Pretimed  30.4
4        Pretimed  37.1
5        Pretimed  34.1
6   Semi-Actuated  17.5
7   Semi-Actuated  20.6
8   Semi-Actuated  18.7
9   Semi-Actuated  25.7
10  Semi-Actuated  22.0
11 Fully-Actuated  15.0
12 Fully-Actuated  10.4
13 Fully-Actuated  18.9
14 Fully-Actuated  10.5
15 Fully-Actuated  15.2
```

```
>
```

We recall that the measured quantity is the delay in seconds of a car waiting at an intersection for a green light. Three types of traffic lights are used (pretimed, semi-actuated and fully-actuated). This is an observational study, and each result in the `Delay` column is a mean of a random sample of cars. Assumptions such as normality, parity of information (equal number of cars representing each mean), homoskedasticity, etc. are assumed.

Plotting wizardry - changing plot size for TeXmacs

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

```
>
```

We would like to note that width and height are specified in inches. The logic of the above code is consistent with how global options should be changed in R. The steps are:

- Call `options` to get the list of options

- Assign the options that we would like to change, in our case, the `texmacs` options, which are added by the TeXmacs plugin at startup

- Finally, we call `options` with one argument, the new list of options

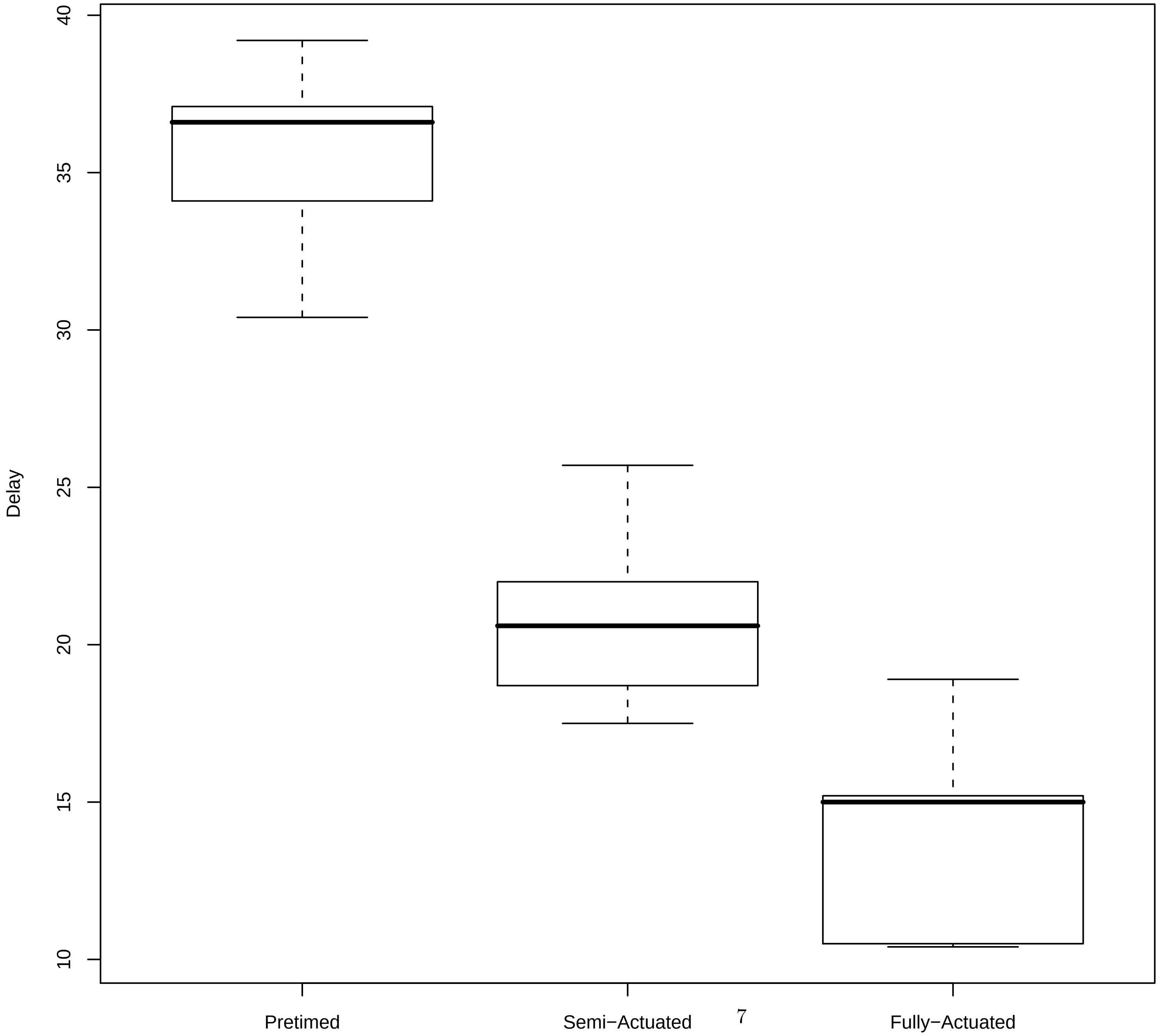
The following command is standard R. Function `X11` controls the Unix X Windows window system (works under Cygwin within Windows and on the Mac if X Windows are used as the display driver). In our case, we change the fontsize to 8 points (point is 1/72 of an inch). We also set width and height of the X Windows window to 3 inches (does not appear to have impact on the graphics which will appear in TeXmacs). We only use the X11 window to provide feedback that the plotting is actually done.

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

```
>
```

The standard plot of data frame

```
> plot(traffic.data);v()
```


Delay
40
35
30
25
20
15
10
Pretimed
Semi−Actuated
Fully−Actuated

```
>
```

Fitting the model

```
> traffic.aov = aov(Delay ~ Type, traffic.data, qr = T,
  projections = T)
```

```
> summary(traffic.aov)
```

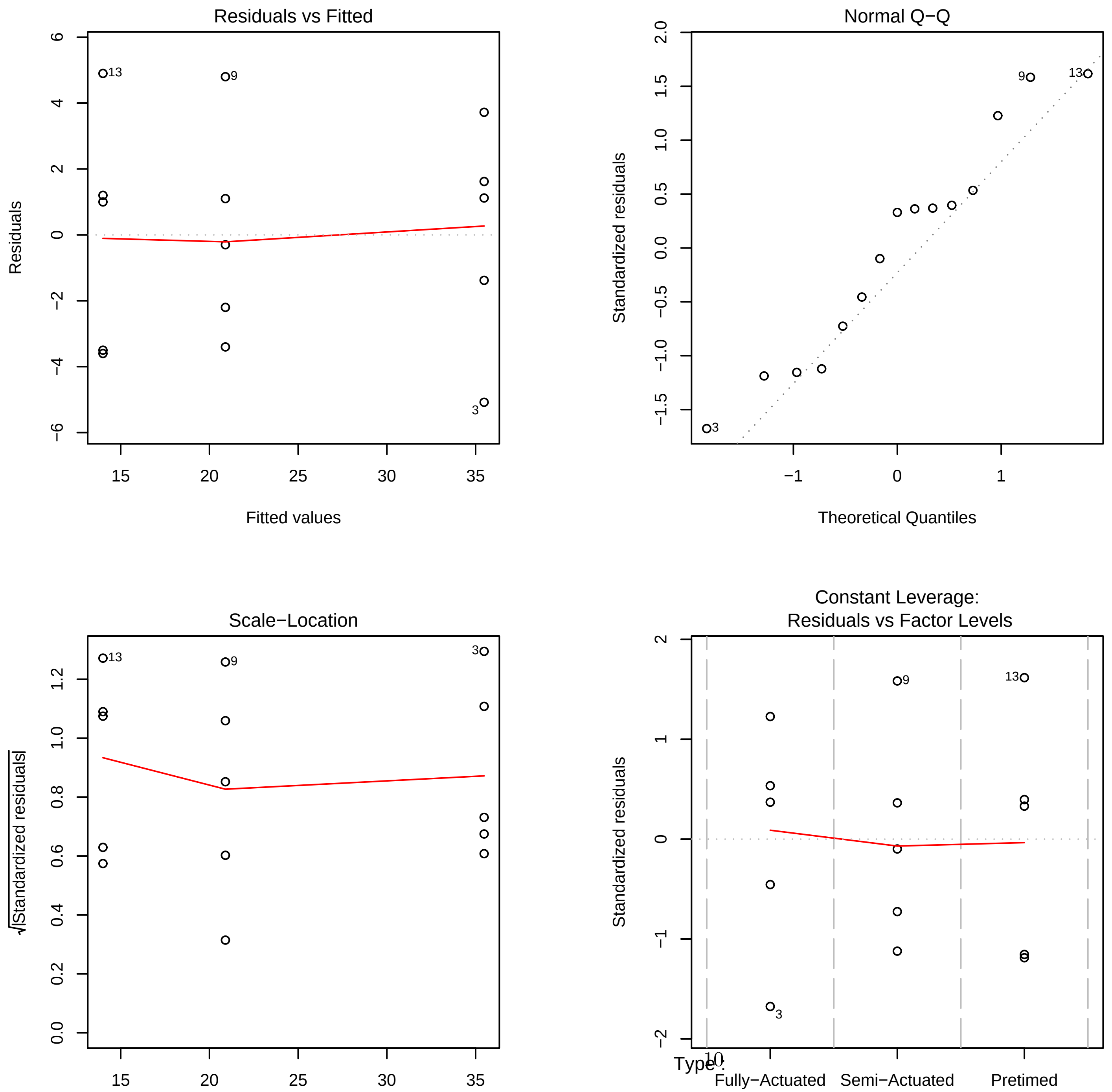
```
             Df  Sum Sq Mean Sq F value    Pr(>F)
Type          2 1202.63  601.31  52.353 1.182e-06 ***
Residuals    12  137.83   11.49
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05
âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
>
```

Various plots produced by the `plot` function

The `plot` function produces four standarized plots related to the analysis of variance.

```
> par(mfrow=c(2,2),pty="s");plot(traffic.aov); v();
```


Residuals vs Fitted
Residuals
Fitted values
13
9
3
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Constant Leverage:
Residuals vs Factor Levels
Type :
Fully−Actuated
Semi−Actuated
Pretimed

```
>
```

Computing the sum of squares of contrast in R

Standard structures produced by `aov` do not appear to contain the sum of squares directly. There may be packages on the Web which do have this functionality. There are many ways to compute the sum of squares, ranging from simulating essentially by-hand computation, to using parts of the structures returned by R and performing some simple postprocessing.

Method 1 — Doing it directly

Group Means

```
> Group.Means = unlist(lapply(split(Delay, Type), mean))
```

```
> Group.Means
```

```
      Pretimed  Semi-Actuated Fully-Actuated
         35.48          20.90          14.00
```

```
>
```

Contrast Values

```
> Contrast.Values = crossprod(contrasts(Type), Group.Means)
```

```
> Contrast.Values
```

```
                   [,1]
PretimedVsActuated 36.06
```

```
FullyVsSemi         6.90
```

```
>
```

Sum of Squares of a Contrast

```
> Contrast.NormalizingFactor =
  as.matrix(diag(crossprod(contrasts(Type), contrasts(Type))))
```

```
> Contrast.NormalizingFactor
```

```
                   [,1]
PretimedVsActuated    6
FullyVsSemi           2
```

```
> r = 5; t = 3; N = r * t;
```

```
> SSC = r * Contrast.Values^2 / Contrast.NormalizingFactor
```

```
> SSC
```

```
                        [,1]
PretimedVsActuated 1083.603
FullyVsSemi         119.025
```

```
>
```

Sum of Squares of Error

```
> Fitted.Values = as.vector(t(replicate(r, Group.Means)))
```

```
> Fitted.Values
```

```
 [1] 35.48 35.48 35.48 35.48 35.48 20.90 20.90 20.90 20.90 20.90
14.00 14.00
[13] 14.00 14.00 14.00
```

```
> SSE = sum((Delay-Fitted.Values)^2)
```

```
> SSE
```

```
[1] 137.828
```

```
> MSC = SSC; MSE = SSE / (N - t)
```

```
> MSC
```

```
                        [,1]
PretimedVsActuated 1083.603
FullyVsSemi         119.025
```

```
> MSE
```

```
[1] 11.48567
```

```
>
```

The F-statistic for each Contrast

```
> F.statistic = MSC / MSE
```

```
> F.statistic
```

```
                        [,1]
PretimedVsActuated 94.34394
FullyVsSemi        10.36292
```

```
>
```

The Student t-Statistic for each Contrast

```
> t.statistic = sqrt(r) * Contrast.Values /
  sqrt(Contrast.NormalizingFactor) / sqrt(MSE)
```

```
> t.statistic
```

```
                     [,1]
PretimedVsActuated 9.713081
FullyVsSemi        3.219148
```

```
>
```

Checking the identity $t^2 = F$

```
> t.statistic^2 - F.statistic
```

```
               [,1]
PretimedVsActuated    0
```

```
FullyVsSemi            0
```

```
>
```

Verification of the Partition of squares

```
> mu = mean(Delay)
> SST = sum((Fitted.Values - mu)^2)
[1] 1202.628
> SSC.sum = sum(SSC); SSC.sum
[1] 1202.628
> SST - SSC.sum
[1] 2.273737e-13
>
```

The Student t-Test

```
> df = N - t; df
```

```
[1] 12
```

```
> pt(t.statistic, df = df, lower.tail = F)
```

```
                         [,1]

PretimedVsActuated 2.450307e-07

FullyVsSemi        3.683055e-03
```

```
>
```

The F-test

```
> pf(F.statistic, df1 = 1, df2 = df, lower.tail = F)
```

```
                         [,1]
PretimedVsActuated 4.900614e-07
FullyVsSemi        7.366110e-03
```

Method 2 - Obtaining information from R

Group means

```
> fitted(traffic.aov)
```

```
    1     2     3     4     5     6     7     8     9    10    11
   12    13
35.48 35.48 35.48 35.48 35.48 20.90 20.90 20.90 20.90 20.90 14.00
14.00 14.00
   14    15
14.00 14.00
```

Sums of Squares

The quickest way to obtain sums of squares for orthogonal contrasts is to examin the result of the `proj` function, with keyword argument `onedf` set to `TRUE`. The returned structure contains (amongst other things) the projections of the pooled experimental data (vector X in other notes), onto all columns of the design matrix (returned by `model.matrix`). We note that for orthogonal contrasts the design matrix is also orthogonal in the standard sense (when the treatment groups have equal size). The design matrix is built from the coefficients of the contrasts, except for the first column (consisting of 1's). Thus, it is not difficult to check that **all** sums of squares (of contrasts and of error) can be obtained by summing the squares of the columns of this structure. The detailed calculation follows.

```
> p = proj(traffic.aov, onedf = T); p
```

```
   (Intercept) TypePretimedVsActuated TypeFullyVsSemi Residuals
1        23.46                  12.02    6.661338e-16      1.12
2        23.46                  12.02   -8.534522e-17      3.72
3        23.46                  12.02    8.675368e-17     -5.08
4        23.46                  12.02    8.675368e-17      1.62
5        23.46                  12.02    8.675368e-17     -1.38
6        23.46                  -6.01    3.450000e+00     -3.40
7        23.46                  -6.01    3.450000e+00     -0.30
8        23.46                  -6.01    3.450000e+00     -2.20
9        23.46                  -6.01    3.450000e+00      4.80
10       23.46                  -6.01    3.450000e+00      1.10
11       23.46                  -6.01   -3.450000e+00      1.00
12       23.46                  -6.01   -3.450000e+00     -3.60
```

```
13        23.46                    -6.01   -3.450000e+00       4.90
14        23.46                    -6.01   -3.450000e+00      -3.50
15        23.46                    -6.01   -3.450000e+00       1.20
attr(,"df")
            (Intercept) TypePretimedVsActuated
TypeFullyVsSemi
                      1                        1
1
              Residuals
                     12
attr(,"formula")
Delay ~ Type
attr(,"onedf")
[1] TRUE
attr(,"factors")
attr(,"factors")$`(Intercept)`
[1] "(Intercept)"

attr(,"factors")$<NA>
```

```
NULL

attr(,"factors")$<NA>
NULL

attr(,"factors")$Residuals
[1] "Type"   "Within"

attr(,"call")
aov(formula = Delay ~ Type, data = traffic.data, projections = T,
    qr = T)
attr(,"t.factor")
      Type
Delay    0
Type     1
attr(,"class")
[1] "aovproj"
```

```
> as.matrix(apply(p[,2:4]^2, 2, sum))
```

```
                         [,1]
TypePretimedVsActuated 1083.603
TypeFullyVsSemi         119.025
Residuals               137.828
```

```
>
```