

STAT 8004 – Statistical Methods II
Spring 2015
Homework Assignment 5 (not to be collected)

- Reading assignment
 - R&S Chapter 9.
 - R&S Chapter 14 for balanced two-way ANOVA.
 - R&S Chapter 15.3 for un-balanced two-way ANOVA.
 - The following are for exercises only, and are not to be collected.
1. Consider the scenario of Problem 15.19 of R&S (the data are in Table 15.1). In a study to examine the effects of $I = 2$ levels of Protein (call this factor A) and $J = 3$ Meat Types (call this factor B), weight gains (y , in gm) were observed for male rats.

- (a) Create three vectors in R of length $n = 47$. The first should contain y values, the second Protein Level (1,2), and the third Meat Type (1-3). Call these vectors respectively “ y ”, “protein”, and “meat”. Then create and print out an R data frame using the commands:

```
> d<-data.frame(y,protein,meat)
> d
```

- (b) Turn the numerical variables `protein` and `meat` into variables that R will recognize as levels of qualitative factors by issuing the commands

```
> d$protein<-as.factor(d$protein)
> d$meat<-as.factor(d$meat)
```

Then compute and print out the cell means by typing

```
> means<-tapply(d$y,list(d$meat,d$protein),mean)
> means
```

You may find out more about the function `tapply` by typing

```
> ?tapply
```

- (c) Make a crude interaction plot by doing the following. First type

```
> x.axis<-unique(d$meat)
```

to set up horizontal plotting positions for the sample means. Then make a “matrix plot” with lines connecting points by issuing the commands

```
> matplot(c(1,3),c(70,110),type="n",xlab="Meat",ylab="Mean
Response",main="Weight Gain")
> matlines(x.axis,means,type="b")
```

The first of these commands sets up the axes and makes a dummy plot with invisible points “plotted” at (1,70) and (3,110) . The second puts the lines and identifying protein levels (as plotting symbols) on the plot.

- (d) Set the default for the restriction used to create a full rank model matrix, run the linear models routine and find both sets of “Type I” sums of squares by issuing the following commands

```
> options(contrasts=c("contr.sum","contr.sum"))
> lm.out1<-lm(y~protein*meat,data=d)
> summary.aov(lm.out1,ssType=1)
> lm.out2<-lm(y~meat*protein,data=d)
> summary.aov(lm.out2,ssType=1)
```

See if anything changes if you ask R to compute “Type III” sums of squares by issuing the command

```
> summary.aov(lm.out1,ssType=3)
```

- (e) Start over with this problem, doing the calculations “from scratch” using your basic linear models knowledge and matrix calculations in R. Compute all of Type I, Type II and Type III sums of squares here, using the sum restriction in the first two cases (and the order of factors A,B). Then compute Type I and Type II sums of squares using the SAS baseline restriction.