Computer Exercise 3 (R)

1. Read the data in **C3Grass.txt**. These are the proportions of plant functional types used in the lecture. For more on multiple regression see section 10.1.3 (pp. 489 – 497) in the R Book. Begin by computing a transformed y-variable: logC3 = $\log_{10}$(C3 + 0.1). By doing `?log` in R, you can learn that the base 10 log function in R is called `log10`, so that you get the transformed variable as

R console
```
dat$logC3 <- with(dat, log10(C3 + 0.1))
```

Also, instead of the LAT and LONG variables given in the data file, use centered variables: compute new variables cLAT and cLONG by subtracting the means from LAT and LONG, for instance

R console
```
dat$cLat <- with(dat, LAT - (mean(LAT))
```

Use the `lm` function in R to get the regression of logC3 on cLAT and cLONG, for instance

R console
```
fm <- lm(logC3 ~ cLAT + cLONG, data=dat)

summary(fm)
```

and write down the regression equation (the summary output contains the intercept and slopes of the regression). How does this regression equation differ from the equation used in the lecture?

Next, check the significance test for the effects of cLAT and cLONG in the multiple regression. In R, you get *t*-values and *p*-values from the summary output. To learn more about sums of squares and the fitting of linear models, you will now do the test in a different way. Begin by loading the car package

R console
```
library(car)
```

The name stands for "Companion to Applied Regression", which is a book by John Fox. The car package contains functions that let you perform certain analyses that many people want, but which some of the "R gurus" think are unsuitable: they have to do with the so-called type III sums of squares. Next, you get an analysis of variance table for your regression with the command

R console
```
Anova(fm)
```

This gives the so-called type II sums of squares, but for this example they are the same as type III sums of squares (you will learn more about using type III sums of squares in R later on). Write down

the sums of squares used in the tests of cLAT and cLONG and the residual sum of squares (note that the p-values are the same as in the summary output; it is in fact an equivalent test). You are now about to check how the cLAT and cLONG sums of squares come about through the comparison of the fit of different models. First investigate the cLAT sum of squares, by performing a simple regression of logC3 on cLONG, e.g.

R console

```
fm1 <- lm(logC3 ~ cLONG, data=dat)

Anova(fm1)
```

and computing the difference between the residual SS for that model and the residual SS for the model with both cLAT and cLONG included (you can also use `anova(fm1,fm)`). Then do the same for the sums of squares in the test of cLAT (`anova(fm2,fm)`).

Finally, compare the regression coefficients in the simple regressions of logC3 on either cLAT or cLONG with the corresponding coefficients in the multiple regressions. Are they the same?

2. Read the data in **Timber.txt**, containing the diameter (in inches), height (in feet), and merchantable volume of wood, in cubic feet, of 31 trees. Your job is to use multiple regression to find a good formula to predict the volume of wood, given measurements that can be obtained before the tree is processed. Since diameter is easy to measure (can be done before the tree is cut down), investigate whether it is worthwhile to also measure the height, given that the diameter is known. Give the estimated regressions of volume on only diameter and volume on both diameter and height. Assuming you read into the data frame dat, you can use commands like

R console

```
fm1 <- lm(Volume ~ Diam, data=dat)

fm2 <- lm(Volume ~ Diam + Height, data=dat)
```

Then use the summary function to get the information you need. Why is there a change in the regression coefficient for diameter between these cases? Try to find the tolerance values for diameter and height in the regression (see p. 128 in Q&K for an explanation of what tolerance is). You can easily compute the tolerance of Diam as one minus the R^2 you get in a regression of Diam on Height, and vice versa for Height (since the R^2 will be the same for these, the tolerance values will also be the same). There is a function in the car package that calculates the Variance Inflation Factor (VIF) for explanatory variables in linear models. VIF is simply equal to 1/tolerance. This function can come in handy when your model contains more than only two explanatory variables. Try comparing your calculated value for tolerance to the VIF:

```
vif (fm2)
```

Based on the idea that volume should be roughly proportional to height x (diameter)^2, also construct three new variables containing the logarithms of diameter, height, and volume. Go through the same

procedure with these variables. Can you explain more variation using the logarithms? What is your final and best predictive formula?

3. Read the data in **Pupw1.txt** (this is the same data you used before). A quick way of making Sex into a factor is the command

R console
```
dat$Sex <- as.factor(dat$Sex)
```

assuming you read into the data frame dat. Last week you tested whether there was sexual size dimorphism, i.e. whether there were sex differences in mean pupal weight, using a *t*-test. Now do it using analysis of variance, for instance as follows

R console
```
fm <- lm(PupW ~ Sex, data=dat)

Anova(fm)
```

The function `lm` can fit either regression or an analysis of variance model, or any other general linear model, depending on the *x*-variables appearing after the ~ symbol. In this case the *x*-variable is a factor, which makes it an anova. Write down the *F* value and the *p* value. Compare with what you got from the *t*-test (with the equal variances assumption) last week. Check that you get the exact same *p*-value, and that the statistics are related as follows: $F = t^2$. So, a *t*-test (assuming equal variances in the two groups) is actually a special case of anova.

4. Read the data in **HouseFly.txt**. These data were used in the lecture to illustrate only random as compared to real group differences. The variable WingLen1 corresponds to no real differences and WingLen2 to real differences. Do an anova (using the lm function) on each of these two variables and compare the *F* values and *p* values. It is important that you make the Group variable into a factor before doing the analysis, for instance

R console
```
dat$Group <- as.factor(dat$Group)
```

otherwise, the lm function will do a linear regression of wing length on group number, which is not what you want.

5. Read the data in **Foodplnt.txt**. The variable AduDryW is the dry weight (in mg) of adult green-veined whites (*Pieris napi*), grown on different kinds of host plants. Four types of plants were used, coded as follows: 1 = high quality *Alliaria petiolata*, 2 = low quality *A. petiolata*, 3 = high quality

Berteroa incana, 4 = low quality *B. incana*. For the analysis, you need to make the variable Foodpl to a factor, perhaps as follows

R console

```
dat$Foodpl <- factor(dat$Foodpl, levels=c(1, 2, 3, 4),
                labels=c("AlHi", "AlLo", "BeHi", "BeLo"))
```

assuming `dat` is the data frame. You might also assign levels to the sexes (1 = male, 2 = female)

R console

```
dat$Sex <- factor(dat$Sex, levels=c(1, 2),
            labels=c("male", "female"))
```

You can confine the analysis to one of the sexes (pick either male or female, or do both, one after the other). For instance, if you choose females, you could make a new data frame

```
datf  <- subset(dat, Sex=="female")
```

(If you enter this in the R commander window, you have to activate `datf` instead of `dat` before analyzes)

First, inspect the data graphically, perhaps in a boxplot of AduDryW

R console

```
boxplot(AduDryW ~ Foodpl, data=datf, col="lightgray")
```

Next, perform an anova to see if host plant affects adult weight, using the lm function to fit the model. In this connection, certain planned comparisons seem natural: Are there differences between high and low quality for *A. petiolata*, and similarly for *B. incana*, and, are there differences between the two host plant species? These questions can be addressed by giving suitable contrasts for the analysis. Define the contrast matrix

```
cntr <- cbind(c(1,-1,0,0), c(0,0,1,-1), c(1,1,-1,-1))
```

which has three columns. The first column has 1, -1, 0, 0 and corresponds to a comparison of high quality *A. petiolata* with low quality *A. petiolata*, the second column is the same thing for *B. incana*, and the third column compares the two plant species. If you now do the analysis like

R console

```
fmf <- lm(AduDryW ~ Foodpl, data=datf, contrasts=list(Foodpl=cntr))

 summary(fmf)
```

you will get a test of significance for each of the three contrasts.

Finally, to show that you can do the same things in many ways using R, repeat the contrasts using the package emmeans:

R console

```
Install.packages("emmeans")

library("emmeans")
```

First specify your linear model like you did the first time, and then make it into the format used by the package emmeans, specifying the variable which you want to use for the contrasts ("Foodpl"):

R console

```
fmf2 <- lm(AduDryW ~ Foodpl, data=datf)

mm1 <- emmeans(fmf2, "Foodpl")
```

Then you can specify the factor levels in a contrast-friendly manner:

R console

```
AlHi <- c(1,0,0,0)
AlLo <- c(0,1,0,0)
BeHi <- c(0,0,1,0)
BeLo <- c(0,0,0,1)
Al <- c(1,1,0,0)
Be <- c(0,0,1,1)
```

Where "AlHi" represents high-quality *A. petiolata*, "AlLo" represents low-quality *A. petiolata*, etc., and "Al" represents both high- and low-quality *A. petiolata*, while "Be" represents both high- and low-quality *B. incana*. If you need help visualizing what the numbers represent, take a look at the boxplot.

Now you can contrast the factor levels against each other by subtracting one with the other. For example, high quality *A. petiolata* contrasted agains low quality *B. incana* would be specified as follows: [1,0,0,0] – [0,0,0,1] = [1,0,0,-1]. Note that the formulation on the right-hand side is how you would write our contrasts using the function inside lm(), like you did before. Now try contrasting the same factor levels as you did before, but this time using emmeans!

R console

```
contrast(mm1, method=list("AlHi - AlLo" = AlHi-AlLo, "BeHi -
BeLo" = BeHi - BeLo, "Al - Be" = Al - Be))
```

Did it change your conclusions?

6. Read the data in **Melon.txt,** consisting of yields from different varieties of melon. First perform an anova, to test if there are differences in yield. Then perform some post-hoc tests of differences in yield between pairs of varieties. One possibility in R is to load the multcomp package

R console

```
library(multcomp)
```

and then to use the `glht` function in that package, as follows

```
fm <- lm(Yield ~ Melon, data=dat)
```

Be sure to to use the above specified command rather than writing: *lm(dat$Yield ~ dat$Melon),* which, in most cases would be perfectly fine but in this case creates an incompatibility with the glht-function used below.

```
mc <- glht(fm, linfct = mcp(Melon = "Tukey"))

summary(mc)
```

also try

```
plot (mc)
```

In this case you used the Tukey method to adjust *p*-values. Compare this with what you get without any adjustment

```
pairwise.t.test(dat$Yield, dat$Melon, p.adj="none")
```

and with the Holm method (which is the same as sequential Bonferroni). Read more about multiple comparisons on pp. 533-534 in the R Book.

```
pairwise.t.test(dat$Yield, dat$Melon, p.adj="holm")
```

Last, repeat the unplanned comparisons using the emmeans package that you used for the previous question:

R console

```
fm <- lm(Yield~Melon, data=dat)
mm1 <- emmeans(fm, "Melon")
pairs(mm1)
```

Here, the function pairs uses Tukey's p-value correction to compare all the groups.

Which pairs do you regard as significantly different in yield? Compare with what you see in a graphical display of the effects, perhaps one made in the following way (inspired from pp. 134-135 in Dalgaard)

```
xbar = tapply(dat$Yield, dat$Melon, mean)

s = tapply(dat$Yield, dat$Melon, sd)

n = tapply(dat$Yield, dat$Melon, length)

se = s/sqrt(n)

stripchart(dat$Yield ~ dat$Melon, method="jitter",
           jit=0.05, pch=16, vert=T)
```

```
arrows(1:4, xbar + se, 1:4, xbar - se, angle=90, code=3,
length=0.1, col="red", lwd=2)
lines(1:4, xbar, pch=4, type="b", cex=2, col="blue", lwd=2)
```

The figure gives the data points ("jittered" in order not to fall on top of each other) and the means with their standard errors.