Computer Exercise 2 (R)

1. Read the data in **Pupw1.txt** (these are the *P. napi* pupal weights you saw in the lecture). You can use the command

R console
`dat1 <- read.delim("Pupw1.txt")`

to read into a data frame `dat1`. Use the command `str(dat1)` in the *R console* to look at the structure of the data. There are two variables, Sex and PupW, and the sexes are coded as 1=male, 2=female. However, Sex is a qualitative or categorical variable used to define grouping of the data. This kind of variable is called a *factor* in R and the values of a factor are called *levels* of the factor. You can change the variable Sex to a factor with the levels male and female as follows

R console

```
dat1$Sex <- factor(dat1$Sex, levels=c(1, 2), labels=c("male",
            "female"))
```

(for the notation c(1, 2) and c("male", "female"), see pp. 35-37 for numeric vectors and p. 87 for text based vectors (factors) in the R Book). Check what you have done by looking at the data, using `str(dat1)` in the *R console*

Next, read the data in **Pupw2.txt**, using the command

R console
`dat2 <- read.delim("Pupw2.txt")`

These are the same data as in **Pupw1.txt**, but males and females are placed in separate variables, PupWM and PupWF. Since there are fewer females than males in the material, there are a number of "missing values" (coded as NA in R) at the end of the PupWF variable. When data are in this form, with no special relationship e.g. between the first male and the first female in the data file, there really is no good reason to keep the data in an R data frame. You can move the data to two variables using the commands

```
PupWM <- dat2$PupWM

PupWF <- dat2$PupWF[!is.na(dat2$PupWF)]
```

where the second command also gets rid of the missing values in PupWF. Section 2.5.1 (pp. 33 – 35) in the R Book describes this in more detail. Make histograms for the two samples (perhaps using the function `norm.fit` in the file NormFit.R from the previous exercise), and find and write down the sample averages and the standard errors. There is no readymade function for the standard error in R, but you can use the command

```
sd(PupWM)/sqrt(length(PupWM))
```

instead. If you think this is inconvenient, perhaps you can write a function that computes the standard error?

Now do the same analysis using the data in the `dat1` data frame. The way to select the males or the females when computing e.g. the mean pupal weights is

R console

```
mean(dat1$PupW[dat1$Sex=="male"])
mean(dat1$PupW[dat1$Sex=="female"])
```

See pp. 36, 172-176, 552 in the R Book for information on this way of selecting subsets of the data. The hard brackets select subsets of a data frame or vector. The selection is: [row(s), column(s)] for data frames and [vector index] (number(s) along the vector) for vectors. Look at p 22. in the R Book for more on logical operators, such as '==' meaning 'equals' and '!' meaning 'not'.

Finally, in the formula for a confidence interval that you saw in the lecture, use the sample averages and standard errors you wrote down to determine 95% confidence intervals for the mean, for both males and females, using the `qt` function to get critical values (see **6**. in Computer Exercise 1). Check your result with what you get from a t-test (see **4**. in Computer Exercise 1).

This exercise shows two different ways of arranging a sample that is structured into subgroups. The arrangement in **Pupw1.txt** is to be preferred: each row of the data file contains data from one sampling unit, and the columns correspond to the properties measured on a unit. A data frame in R is meant to represent this kind of data. In the file **Pupw1.txt**, males and females were coded using the numbers 1 and 2, but it is actually better to write male and female directly in the data file. In such a case, R would automatically make the variable Sex into a factor when it is read into a data frame (you will see many examples of this in future exercises).

2. Continue using the data in **Pupw1.txt** from the previous exercise. Find the sample variance of PupW for each sex separately (male is coded as 1 and female as 2 in the data file, but you may have converted Sex to a factor with levels male and female in the previous exercise). For instance, the commands

```
var(dat1$PupW[dat1$Sex=="female"])

length(dat1$PupW[dat1$Sex=="female"])
```

give the variance and the sample size of female pupal weights, assuming you have converted Sex to a factor. Because a sample variance is actually a mean square, you can construct an *F*-ratio from the two mean squares and test the null hypothesis that the true variance is the same for the two sexes. Have a look at pp. 120-121 as well as pp. 286-287

R console

```
var.test(PupW ~ Sex, data=dat1)
```

If you find that the variances are the same in the two sexes, go on to test whether there is sexual size dimorphism, i.e. whether there are sex differences in mean pupal weight.

R console

You can use:

```
t.test(dat1$PupW[dat1$Sex=="male"],
       dat1$PupW[dat1$Sex=="female"], var.equal=T)
```

but in this case the alternative command

```
  t.test(dat1$PupW ~ dat1$Sex, var.equal=T)

or: t.test(PupW ~ Sex, var.equal=T, data=dat1)
```

is more convenient (write ?t.test to learn more).

3. Read the data in **Deer.txt**. These data were obtained in an experiment on food choice in fallow deer. There were 10 individuals (hand reared females) in the experiment, and each individual was presented with bowls with different types of pellets, some with lower and some with higher concentrations of (bitter tasting) tannin. The variables AmntLow and AmntHigh give the amounts (in grams, averaged over 8 trials) eaten from the different types of bowls. Your job is to find out if the deer differentiated between the two types of food. Which test will you use and what do you find?

4. Continue using the data in **Deer.txt**. In addition to the amounts eaten from the different types of bowls, the average time (in seconds) spent eating was also recorded. Your job is to investigate if any of the variables AmntLow, AmntHigh, TimeLow and TimeHigh are correlated. Which test do you use and what do you conclude? (See pp. 376-377 in the R Book) Should you take into account that you are doing multiple tests on the same data? If so, how will you do that? (Check on pp. 531 – 353 in the R Book).

5. Read the data in **Spider1.txt** into a data frame. The data are from staged fights between pairs of males of the bowl and doily spider (*Frontinella pyramitela*), in the presence of a female. Each row corresponds to a fight between a different pair of males. The variable Asymm is a measure of the size difference between the males (it is log(Len1/Len2), where Len1 and Len2 are the lengths in mm of the two males). Grapple is the duration in seconds of the fight. According to certain theories of fighting behaviour, one ought to expect longer fights when the opponents are of similar size, whereas if one spider is much smaller than the other, that spider might give up the fight fairly quickly. Your job is to test this theory.

Begin by looking of a scatter plot of Grapple against Asymm. Then compute a new variable, called AbsAsymm, which is the absolute value of Asymm. You need this kind of variable because the theory predicts shorter contests both for negative and positive values of Asymm. Assuming that you read into the data frame `dat`, a command for adding AbsAsymm to `dat` is

R console

```
dat$AbsAsymm <- abs(dat$Asymm)
```

Use the command `str(dat)` to see what happened. Now look at a plot of Grapple against AbsAsymm, for instance

R console

```
plot(dat$AbsAsymm, dat$Grapple, col="red")
```

How should you go about testing if Grapple depends on AbsAsymm? One possibility is to transform the Grapple variable, to achieve an approximately linear relationship. For instance, try a log transformation of Grapple, adding a variable LogGrapp to the data frame. Perform a linear regression of LogGrapp on AbsAsymm, using the command

R console

```
    fm <- lm(LogGrapp ~ AbsAsymm, data=dat)
```

The results have been put in a variable called `fm`. The way to think about this is that a linear statistical model of LogGrapp as a function of AbsAsymm has been fitted and the result put in the object fm (I used the name fm to denote fitted model; see also pp. 450-464 in the R Book). You extract information about the model by applying functions to the fm object. For instance, you can determine the equation for the regression line using the command:

R console

```
    summary(fm)
```

which also gives you tests of significance. This gives you the same results as those shown in R commander. Do you think this regression is OK? You can look at a plot with the regression line

R console

```
plot(AbsAsymm, LogGrapp, col="red", cex=2)
abline(fm, col="blue", lwd=3)
```

and you can get several diagnostic plots with the commands

```
    plot(fm)
```

Finally, what do you conclude about the fighting theory?

6. Read the data in **Mcinxia1.txt** into a data frame. The file contains pupal weights, adult weights, and head width for a sample of *Melitaea cinxia* butterflies, reared on the roof of the biology building. Your job is to determine the allometric relationships between adult weight and pupal weight, and between head width and adult weight, for at least one of the sexes (limit the analysis to either females or males).

Allometric relationships are linear relationships between log body size measures, so you need to log transform the size measurements. Perform linear regression of log adult weight on log pupal weight, and of log head width on log adult weight. For instance, for the regression of log adult weight on log pupal weight for males is put into `fm1` with the command

```
 fm1 <- lm(log(adulwei) ~ log(pupawei), data=dat, subset=(sex=="m"))
```

and you can deal with log head width in a similar way. Use the summary function and plots to determine if the regressions are OK. Are the slopes significant? What are the allometric relationships (i.e. the regression equations)?