Packages

Packages are bundles of data and ways to process data (functions) that have been provided by helpful individuals and organizations. They greatly extend the functionality of R, especially for us as we study time series. In this lecture we will work to

- Understand the R environment with respect to packages
- Be able to download and use Packages, e.g. faraway
- Access and analyze the data available in packages

Once you have R up and running, look at the GUI toolbar and you will find a button labeled *Packages*. Take a moment to select *Install Package(s)*, after which a window labelled *CRAN mirror* will pop up. Find a geographically close and safe site, select it, and then click on *OK*.

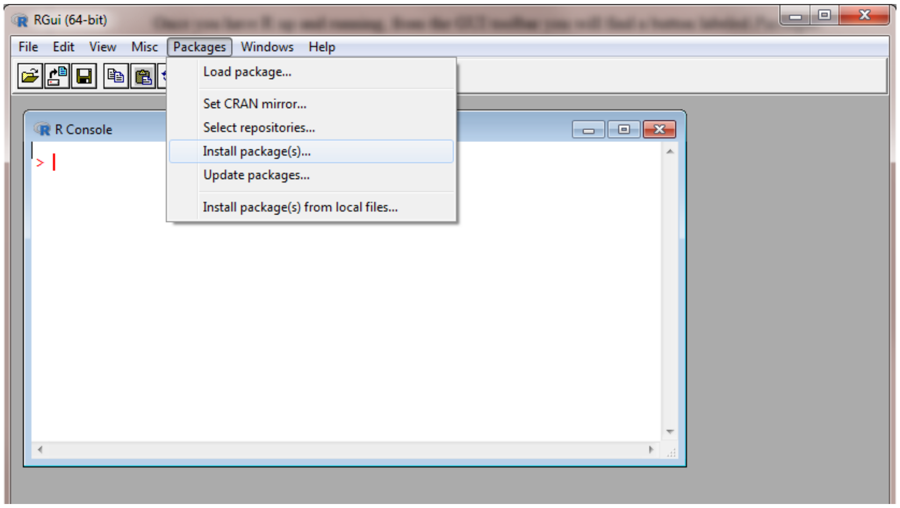

Figure 1 Screen Capture of R GUI for loading packages

When the pop-up labelled *Packages* presents itself, scroll down and select *faraway*, then double click or press *OK* to install the package *faraway*. This package will allow us, among other things, to access data sets that will help us to get started.

There are other ways to install packages once we get into the scripting environment, but “clicking” is probably the easiest way to get started. We can discuss the concept of packages in more detail later, but for now it is important to just get access to some data and begin to explore what R can do for us.

In the *R GUI*, you should have a window available labelled *R Console*. By typing *ls()* you can see that we currently do not have any data sets immediately available to us in the workspace. In order to see what data sets Faraway has made available to us, type

```
data(package='faraway')
```

As you can see, there are many data sets at your disposal within this package. Obviously, other packages have other data sets- R is a very rich environment! If you'd like, you can also see which data sets come with the basic installation of R just by typing *data().*

Right now we wish to use a data set obtained from an experiment where animals were fed a variety of diets. After they became accustomed to the diet, blood coagulation times were measured. These data are derived from the book from *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building* (Box, Hunter, & Hunter, 1978). At the R Console, type

```
data(coagulation, package='faraway')
```

to "load" the data. You will be underwhelmed with what you see, which is essentially nothing. If you type *ls()* again, you will see that coagulation is readily available. To see what this data set holds you need to inquire. Type *coagulation* just by itself to see the actual data values. You should see 24 rows of data (we call these *cases*) on your console:

```
          coag      diet
 1         62        A
 2         60        A
 ⋮          ⋮         ⋮
24         59        D
```

Evidently each case has three variables associated with it: a case number (that is, which individual animal we are talking about), the coagulation times in seconds for this individual animal, and finally which diet the animal was on.

If you like working with spreadsheets, you would probably organize the data differently:

Diet	*Coagulation Times*							
A	62	60	63	59				
B	63	67	71	64	65	66		
C	68	66	71	67	68	68		
D	56	62	60	61	63	64	63	59

In R, these data are stored as a *data frame*. We can discuss data frames later. For now, just understand that many data sets are organized so that an individual (called a *case*) has several aspects measured on it (we call these *variables*) and that we often wish to store each individual’s readings together. So, the first animal was fed diet A and had a blood coagulation time of 62 seconds. The second animal was fed diet A and had a blood coagulation time of 60 seconds. The sixth animal was fed diet B and had a coagulation time of 67 seconds, etc.

Data frames allow us to do some tasks very naturally. R knows what types of plotting and analysis most people would choose to do for the most common situations. For example, recall from your elementary statistics courses that we often like to use box plots to summarize and present data sets visually. Since we have 4 different diets, we would probably want to obtain a graphical representation “side by side” to see whether there are obvious differences. To obtain your box plot, type the following in the console window:

```
plot( coag ~ diet, data=coagulation)
```

Unpacking this, R will plot “coagulation on diet” (with “*coag ~ diet”*). This groups according to diet on the horizontal and then boxplots the coagulation times on the vertical. As you can see from the plot below, R is making assumptions about your data set and about how a reasonable person (such as yourself) would like to plot the data.

Notice that your data values are “protected” by the data frame (this is the ***data=coagulation*** part***)***. The variables *coag* and *diet* are only available if you indicate that the data frame in play is the *coagulation* data frame. This can be especially handy when you have multiple data frames open at the same time.

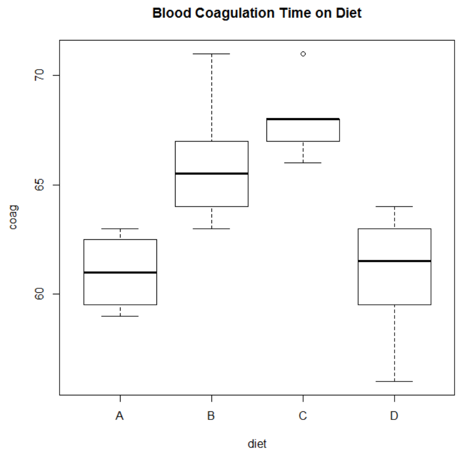

Figure 2 Box plot representation of blood coagulation times (in seconds) for animals given 4 different diets.

Not to belabor the point, but if you had just typed *plot(coag ~ diet)* you would obtain an error. R would claim to not know what data you are talking about. I tried this and all I got for my efforts was *Error in eval(expr, envir, enclos) : object 'coag' not found.*

Looking at the boxplot, it seems natural to ask whether the average coagulation times are independent of diet. This is a problem typically approached with ANOVA (Analysis of Variance), one of the topics we assume you have seen, at least a little. You may have done some simple ANOVA in an introductory statistics course. If you type *summary(coagulation)* you obtain the output seen below. Note that this "bunches up" or aggregates the coagulation data into 24 coagulation times and 24 diet types and tells useful information for each variable individually, rather than associating coagulation time with diet like the last plot. Since *coag* represents times, it is useful to know the "5 number summary" (i.e. the quartiles) as well as the average, or mean. Diet divides our cases into 4 groups, and you might want to know how many cases are in each group.

```
           coag              diet
      Min.    56.00          A:4
   1st Qu.:   61.75          B:6
  Median :    63.50          C:6
      Mean    64.00          D:8
   3rd Qu.:   67.00
      Max.    71.00
```

If you want to access each variable, you can use the operator “$” as in:

```
coagulation$diet
```

which returns

```
[1] A A A A B B B B B B C C C C C C D D D D D D D D
Levels: A B C D
```

and

```
coagulation$coag
```

which returns

```
[1] 62 60 63 59 63 67 71 64 65 66 68 66 71 67 68 68 56 62 60 61 63 64 63 59
```

If you are timid you may lightly skim the next page or so. If you are intrepid I’ll quickly note that one way to see summary data for each class (which, admittedly is a little premature at this point in our discussion, but will get us oriented in the right direction) is as follows:

```
ourModel = lm(coag~diet-1, coagulation)
summary(ourModel)
```

You should see the following print out:

```
Call:
lm(formula = coag ~ diet - 1, data = coagulation)
Residuals:
    Min         1Q            Median        3Q            Max
-5.000e+00   -1.250e+00    1.743e-16     1.250e+00     5.000e+00
Coefficients:
       Estimate      Std. Error     t value      Pr(>|t|)
dietA  61.0000       1.1832         51.55        <2e-16 ***
dietB  66.0000       0.9661         68.32        <2e-16 ***
dietC  68.0000       0.9661         70.39        <2e-16 ***
dietD  61.0000       0.8367         72.91        <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 2.366 on 20 degrees of freedom
Multiple R-squared: 0.9989,     Adjusted R-squared: 0.9986
F-statistic:  4399 on 4 and 20 DF,  p-value: < 2.2e-16
```

If you have never done regression, this is a bit much to take right now!!! But don't worry, and *never panic*. Our job in this course is to

- know why a rational person would make such a call and
- to understand all of this printout.

For now just note, consistent with our box plots, the averages for each of the 4 diets are: 61, 66, 68 and 61 seconds, respectively. (Recall, however, that a box plot shows the ***median***, not the ***mean***).

An Exercise: It's time for some rats!

a. Load in the data set on poisons, treatments, and rats (effect of toxic agents on rats) with the command:

   data(rats, package='faraway')

   We can find out about all of these data sets by downloading and reading through

   http://cran.r-project.org/web/packages/faraway/faraway.pdf

b. Write a paragraph explaining what variables are recorded in the data set and say where Faraway got the data. Swashbucklers will also take a glance at (Box & Cox, 1964)

c. Obtain box plots for survival times by poisons and for survival times on treatments. For instance, the box plot for survival times on treatments may be obtained as

   plot(time~treat,data=rats)