

To Enter Raw Data into R

In general, you want to create “vectors” of data.

For example: `> myvariable1=c(1, 2, 3, 4, 5)`

If you prefer entering them in more of a spreadsheet window, one way to convince R to let you do this is to type

```
> data.entry(1)
```

This opens up a Data Editor window (with a name of 1 and a first value of 1, but you will just overwrite those). You can name the columns by just typing into the header cells (also specify the mode here as numeric or character). You can enter data by just typing in values and hitting return or tab. You can also use the up and down arrows to navigate.

When you are done, just choose File > Close. If you type `ls()` you should now see the variable names you created. To change any of the data values, you can then use

```
> data.entry(variablename)
```

(you can list any number of variable names here). If you click on the name in the Data Editor window, you can change the variable mode from numeric to character and vice versa. You must close the Data Entry window before you can enter new commands in the R Console.

Alternatively, you can enter the data in Excel, preferably with column names. When you are done, chose File > Save As. Use the “Save as type” pull-down menu to select .csv (comma delimited) or .txt (Text (tab delimited)) and then give the file a name and press Save. Make sure you keep track of where this file is stored (which folder or directory). Then in the R Console you need to read in the data specifying the entire path name for where the file is located. For example:

```
> mydata = read.table("C:/My Documents/Times.csv",  
+   header=TRUE,                               if you have column names, FALSE if not  
+   sep=" , ",                                  to match .csv format, use "\t" for tab delimited  
)
```

Or you can open a dialog window to find the file:

```
> mydata=read.table(file=file.choose())
```

You can use the same header and separator options.

You can also copy the data in Excel to the clipboard (just select and use ctrl-C to copy)

```
> mydata = read.table("clipboard", header=TRUE, sep="\t")
```

Then to access individual variables, you can use \$, for example: `mydata$var1`

Or use the with command e.g., `> with(mydata, do this ...)`

It might even be easier to cast these to other variables, e.g.,

```
> time = mydata$time
```

Just keep in mind if you edit one, you don't edit the other.