Math 10 - Spring 2013
Handout - Getting Data into R

Numbers have an important story to tell. They rely on you to give them a voice.— Stephen Few

First, check that your data is in the right format to be read into R. The data you are interested in should be organized into columns, so that each column represents a unique variable, and each row represents a different data point or measurement of that variable. Also make sure that each column has a title (preferably without using spaces.) Save the data as a csv (Comma separated value) file if it is not already in that format. (This is an option under format in the save-as dialog box in excel for example.)

Now, you will need to tell R where to find the data on your computer. At any time, R has a *working directory*, the current folder where it will look for files and save files. You can find out what directory this is by typing:

```
getwd()
```

This may not be the directory where you're data is stored, however. You can change this directory using the `setwd` command. For example, to change the working directory to
`/Users/nathan/Desktop` you would type:

```
setwd("/Users/nathan/Desktop")
```

If you want to list the contents of the current directory, you can do so using the command

```
dir()
```

Now you want to load your data into R. Assuming you have saved your data in the file `data.csv` and your working directory is the directory where this file is located, you can use the command

```
dataframe <- read.csv("data.csv")
```

which creates a dataframe named `dataframe` (you can give it any name you want) and fills it with the data from your csv files, with column names taken from the first row of the file. You can view the contents of your data by simply typing

```
dataframe
```

If you file contains a lot of data you may find it hard to read all of your data at once. You can use the commands

```
head(dataframe)
```

and

```
tail(dataframe)
```

to view just the first (respectively last) 7 lines of your data. In some cases you may need to load an excel file rather than a csv file. R cannot do that on its own, however you can install a package `gdata` command which can. The command `install.packages("gdata")` will download and install the package, which you can then load with `library(gdata)`. You can then use the command `read.xls` in the same way as `read.csv`.

Cleaning up your data

In many cases you may find that your data needs to be cleaned in some way. Note that it may be easier to do some of this in excel before loading the data into R. First, remove any columns that you don't need from your data, for example if your data has a column called `other`, you can remove it from the dataframe by running the command

```
dataframe$other <- NULL
```

Even after cleaning out unwanted columns however, there may still be entries without any data in them. Whenever R encounters an empty cell in a dataframe, it replaces it with `"NA"`, meaning there is no data in that position. The problem with this is anytime you compute a statistic using data that is incomplete you will get `"NA"` for an answer.

One way to solve this using the command `complete.cases` which returns true for every row that does not have any missing data. You can use this to create a new dataframe which consists only of complete rows in the first dataframe like so

```
newdataframe <- dataframe[complete.cases(dataframe), ]
```

In many cases, you may simply want to fill in a zero for empty cells, which can be done using the following command, to replace each instance of `NA` in the column `height` of dataframe with a zero.

```
dataframe$height[is.na(dataframe$height)] <- 0
```