R Tutorial Supplement
(addition to Section 2.3 of ISLR)

(In lab tutorial, Monday July 21, 2014)
Note: These notes are supplementary to the R tutorial in Section 2.3 of ISLR. Information is organized according to the same section headings as ISLR.

2.3 Lab: Introduction to R

RStudio: After installing R, I recommend that you install the add-on user interface "RStudio". Go to www.rstudio.com and download and install the "desktop" version. As of this writing I'm using version v0.98.976.

2.3.1 Basic Commands

The RStudio interface
There are several different windows inside RStudio. These include:

- *Console*, where you type R commands and get output
- *Environment / History* window, where you see R objects and past commands
- *Files / Plots / Packages / Help / Viewer*, for various output and information
- *Editor*, where you edit R commands

When entering commands in R, you have the option of typing them directly in the console or typing them in the Editor. As a general rule, use the console for commands you want to try out, and the editor for commands you want to save and re-use. There are buttons and keyboard shortcuts that let you execute line(s) of code from the editor in the console. The same principle of editing a file and running commands from it can be used in "plain R".

As with any editor program, you need to be careful about the folder / directory in which you save your file with R commands. You can also set the R working directory, and a good strategy is to set it to be the same place as the R file you are editing.

Note that in both R and RStudio, you can copy commands from other sources and paste them into the console. "Other sources" includes pdf documents.

As we work through the commands in this section you will notice that the *Environment* window displays objects and their contents as they're created.

2.3.2 Graphics

In the RStudio "Plots" window, the "export" button provides options for saving a copy of a plot. You will find it useful to "copy to clipboard" and then insert the graphic into your favorite word processor. This will let you make up solutions to assignment questions in a word processor. Other export options, including saving as pdf or jpg files, and those explained in the ISLR Ch 2 tutorial, are available.

RStudio's "Plots" window has a "history" (back and forward arrows) that lets you page through plots generated in your R session.

The contour plot on p.46 uses "outer" which is efficient but confusing. You can also use loops, which are slower but easier to understand:

```
f = matrix(0,50,50)
for (i in 1:50) {
  for (j in 1:50) {
    f[i,j] = cos(y[j])/(1+x[i]^2)
  }
}
```

(Notice how I use a fixed width font, like courier, to display R code. The equal width of all characters makes code easier to read. Please do this for your assignments.)

2.3.3 Indexing Data

Clicking on the "spreadsheet icon" () beside a matrix (like "f") will open a spreadsheet-style viewer.

2.3.4 Loading Data

The "Auto" data used in this section is part of the "ISLR" package. To access the data we have to install the package and load it. In RStudio, go to "Tools > Install Packages" and type "ISLR" (in capitals) in the "Packages" box. When you click "Install", RStudio will download the ISLR package from the web, and install it. You only need to do this once.

After you have downloaded and installed a package, you still need to load it. Type

```
library(ISLR)
```

to load the package. You must do this in every R session where you want to use functions or data from the ISLR package. See also ISLR Section 3.6.1.

Once you have loaded the ISLR package with the "library" command, you do not need to use the "read.table" command to load the "Auto" data. It is already loaded in R. You can view the file using

```
fix(Auto)
```

as indicated in the text. BUT BE CAREFUL! The "fix" function lets you edit entries. Any edits you make will be saved in a "personal" copy of the data frame "Auto", and you can go back to the original by simply deleting your copy, with "`rm(Auto)`"

The ISLR tutorial gives several ways to read data into R. You can locate a file on your computer and read it from there. Sometimes the file is online and you can read it directly from the internet. We'll read from the internet here, because it is easiest. Because the commands are slightly different than in ISLR, I reproduce them below.

```
Auto = read.table('http://www-bcf.usc.edu/~gareth/ISL/Auto.data')
```

It's complicated to work through the examples as they are given in the textbook. You need to save a local copy of "Auto.data" in a folder in your computer, then either change R's working directory to the folder where you saved it, or give the full path to your file to "read.table".

```
Auto = read.table('http://www-bcf.usc.edu/~gareth/ISL/Auto.data',
  header=TRUE,na.strings='?')
```

Footnote: Word processors tend to get "fancy" with quotes (look closely at the quotes in this sentence). Cutting and pasting from a word processor to R or RStudio may generate errors if matching "..." or '....` are used. This is another good reason why using the built-in editor for your commands is better.

2.3.5 Additional Graphical and Numeric Summaries

WARNING: The `attach()` function needs to be used with caution (or not at all!). It is easy to lose track of what assignments are being made to what variables. For example in this section when you type

```
cylinders = as.factor(cylinders),
```

you're not changing a column of `Auto`, but instead you are making a new variable called "cylinders" that has nothing to do with `Auto`. You can check this via `summary(Auto)`.

Without `attach`, you would make the above change directly to `Auto` as follows:

```
Auto$cylinders = as.factor(Auto$cylinders)
```

If you must use `attach`, then you should use detach a data frame afterwards:

```
detach(Auto)
```

The `q()` function, for quitting R, will ask you if you want to save the workspace image. Answering "yes" will save all the objects you have created in R. While this can be helpful, it is an even better habit to instead save all the commands you used to generate the results. This will give a "permanent record" of all the steps you followed to make the objects, and enable you (or others) to reproduce your results exactly.