Stat 6021: R Tutorial for Module 1

Today, we will learn some of the basics of the statistical software R. We will learn how to read data in from a text file, how to handle variables, how to plot data, and how to save these plots.

1. Directories. Create a directory within your "home directory" titled "stat6021". Create a directory within this "stat6021" directory titled "module1".

2. Launch R. You should always make sure you know what the working directory is. There are a few ways to set the working directory.

   - Go to File >> Change dir...
   - Use command "setwd(dir)", where dir is replaced by the path of the working directory you want, e.g.

     ```
     setwd("C:/Users/Owner/Documents/stat6021/module1")
     ```

   The command "getwd()" will display the current working directory.

3. Scripts. Scripts, which are almost like text files, contain the commands that you will enter on the command line in R. I highly recommend using scripts instead of typing commands directly to the command line. It's much easier to edit commands this way and you can save your commands for future reference. Use File >> New script to open a window to type in your script. To save, use File >> Save as, and make sure to save the script with a .R extension.

4. Download files. Obtain the data file, "purity.txt" and save this file in "module1". The data set records the purity of oxygen produced by a fractional distillation process, which is believed to be influenced by the percentage of hydrocarbons in the main condenser of the processing unit, from 20 samples. We want to import this data into R. One way to do this is to use the `read.table()` command; simply type

   ```
   read.table("purity.txt", header=TRUE ,sep="")
   ```

`read.table()` is used when the data are to be treated as a data frame, which is the case with most data sets in regression. We need to set TRUE for header when the file contains the names of the variables as its first line. By default, this is set as FALSE.

R has read the data frame and output them to the screen. We would like to store the data frame so that we can work with it. To do this, we declare a variable `data` and set it equal to the output of the `read.table()` command, i.e. the list of numbers. Type

```
data<-read.table("purity.txt", header=TRUE ,sep="")
```

5. Checking variables in data frame:

   (a) Type `names(data)`. This command lists all the variables present in data frame named `data`.

   (b) To check the contents of a variable, simply input `data$hydro` if you want to check the contents of the variable hydro.

   (c) You can also define `data` as the default dataset using the command `attach(data)`.

   (d) After attaching `data`, you can check the variables just by typing the variable name, e.g. `hydro`.

   (e) The command `detach(data)` will remove `data` as the default data frame.

   (f) Now, type `hydro`. What do you see?

6. (Re-attach `data`) Easy plotting. Making attractive plots is one of R's strong suits.

   (a) Create a scatter plot of the `purity` against `hydro` using `plot(hydro,purity)` or `plot(purity~hydro)`.

   (b) We can change the labels for the axes and add a title.

   ```
   plot(hydro,purity,xlab="Percent of Hydrocarbons", ylab="Purity of Oxygen",
   main="Plot of Purity of Oxygen against Percent of Hydrocarbons")
   ```

   Note: For all work, all plots must be labeled to obtain full credit.

   (c) We now want side-by-side plots of each variable. To produce more than one plot in a window, we need to split up the plot "device". We do this with

   ```
   par(mfrow=c(1,2))
   plot(hydro, main="Plot of Percent of Hydrocarbons for each Sample")
   plot(purity, main="Plot of Purity of Oxygen for each Sample")
   ```

7. Saving plots. We can save this plot in any format that we would like (pdf, postscript, jpg, etc). For example, we want to save this graphic as "joint.jpg".

```
jpeg("joint.jpg")
par(mfrow=c(1,2))
plot(hydro, main="Plot of Percent of Hydrocarbons for each Sample")
plot(purity, main="Plot of Purity of Oxygen for each Sample")
dev.off()
```

Note that you won't actually see the plot. The dev.off() command will produce the completed file. Without it you may get nothing at all. To save as pdf, replace `jpeg("joint.jpg")` with `pdf("joint.pdf")`.

8. Performing regression. Use `lm(purity~hydro)`. To store the results from fitting the linear model, create a variable, e.g. `result<-lm(purity~hydro)`. Typing `summary(result)` produces even more information. Where are the values of $\hat{\beta}_1$, $\hat{\beta}_0$, R^2, and $\hat{\sigma}^2$?

9. Obtaining the ANOVA table. Use `anova(lm(purity~hydro))` to obtain the ANOVA table. Note that in R, the total sum of squares is not provided. What is the total sum of squares for this data set?