Some helpful R commands

Entering data to R Studio

To read in data from an Excel csv file called *excel_data.csv* to R Studio and name it `mydata`, first use the drop down menus in R Studio **Session > Set Working Directory > Choose Directory** to indicate the location of *excel_data.csv* on your computer. The following code will then read the data into R Studio:

```
mydata<-read.csv("excel_data.csv")
```

`attach(mydata)` — this adds the variable names

At the end of the analysis remember to use `detach(mydata)` to disassociate the variable names.

(a) Graphics

<u>If you have the numeric variables X and Y:</u>

`hist(X, main= "Title",xlab="x-axis label",ylab="Frequency")` — this produces a histogram of the variable named X, it adds a title and axis labels

`boxplot(Y, main="Title", ylab="y-axis label" )` — produces a boxplot of the numerical variable Y

`boxplot(X,Y, main="Title", xlab="x-axis label", ylab="y-axis label", names=c("X","Y"))` — produces a comparative boxplot of the numerical variables X and Y

`plot(X,Y, main="Scatterplot of Y on X",xlab="x-axis label",ylab="y-axis label")` — produces a scatterplot of Y on X

<u>If you have the categorical variable X:</u>

`table(X)` — computes the number of observations in each level of the categorical variable X

`pie(table(X), main="Title")` — this gives a simple pie chart of the categories in variable X with the specified title

`barplot(table(X), main="Title", xlab="x-axis label", ylab="Frequency")` — this gives a bar chart of the categorical variable X with the required title and axis labels

(b) Descriptive Statistics

`mean(X)` — computes the mean of the numerical variable X

`sd(X)` — computes the standard deviation of the numerical variable X

`summary(X)` — computes the mean, median, minimum, maximum and upper and lower quartiles of the numerical variable X

`IQR(X)` — computes the interquartile range of the numerical variable X

`prop.table(table(X))` — returns the proportion of observations in each level of the categorical variable X

`prop.table(table(X))*100` — returns the percentage of observations in each level of the categorical variable X

`table(X,Y)` — produces a cross-tabulation between the two categorical variables X and Y

(c) Correlation and Regression

`cor.test(X,Y)` — computes the correlation between X and Y and performs a test of the null hypothesis of zero correlation

`lm(Y~X)` — fits a linear regression line to the data (lm command stands for linear model)

`abline(lm(Y~X))` — adds the least squares linear regression line to an existing scatterplot of Y on X

`summary(lm(Y~X))` — displays the coefficient of determination (R-squared)

<u>To predict with your Linear Model:</u>

`predict(lm(Y ~ X), newdata=data.frame(X=C),interval = "pred")` — computes the predicted value of Y when X=C along with a 95% prediction interval

(d) Hypothesis Testing

`t.test(X,Y)` — performs a two-sample t-test between X and Y

`t.test(X,Y,paired=TRUE)` — performs a paired t-test between X and Y

`prop.test(x = c(a, b), n = c(n1, n2))` — performs a 2-sample test for equality of proportions