**Basic Commands for RStudio**

**Entering data**

Set working directory : Session>Set Working Directory>Choose Directory

**Read in .csv file:** MyData <- read.csv(file.choose(), header = TRUE) (Use browser to find and select

.csv file you want.)

**Read data into variables and combine into a data frame.** Example:

X <- c(5, 7, 9 , 12, 12, 14, 18, 20)

Y <- c(20, 18, 16, 14, 12, 13, 11, 10)

W <- c(1, 2, 2, 2, 1, 1, 2, 1)

MyData <- data.frame(cbind(X, Y, W))

str(MyData) structure of variables in the data frame.

MyData (This will print out **all** of the data in the data frame.)

head(MyData) (This will print out the first 6 rows in MyData.)

**Change W into a factor, label 1s Blue and 2s Red (do not include missing value levels or labels):**

MyData$W <- factor(MyData$W, levels = c(1, 2),labels = c("Blue", "Red"))

**NOTE: Let me repeat:** If there are missing values, such as the -9s from MTF data, do not

include in levels or labels. R will treat as NA.)

**Read in two-way table** (such as one from a textbook exercise)

MyTable <- as.table(rbind(c(enter row 1 entries separated by commas), c(enter row 2), etc.))

MyTable (prints table without variable headings)

dimnames(MyTable) <- list(*RowVariableName* = c("first outcome row variable ", "second

outcome row variable", etc.), *ColumnVariableName* = c("first outcome column variable”,

“second outcome column variable”, etc)

MyTable (prints out table with row/column variable names, row/column outcome labels.)

**Commands attach and detach (allows access to variables without using the data frame name)**

attach(data frame name)

detach(data frame name)

**Descriptive Statistics**

Sample mean and standard deviation:

mean(MyData$X)

sd(MyData$X)

Five number summary (including mean)

summary(MyData$X)

Interquartile range:

IQR(MyData$X)

Correlation of X and Y:

cor(MyData$X, MyData$Y)

**Graphic Displays**

Histogram: hist(MyData$X)

hist(MyData$X, breaks = 5, xlab = “X”) Controls number of class intervals, adds label

to horizontal axis

Box-and-whisker plot: boxplot(MyData$X, horizontal = TRUE)

boxplot(MyData$X, horizontal = TRUE, xlab = “variable name”, main = “title for graph”)

Box-and-whisker plot for side-by-side comparative boxplots (X quantitative, W categorical):

boxplot(MyData$X ~ MyData$W, horizontal = TRUE)

Stem-and-leaf plot:

stem(MyData$X)

stem(MyData$X,2);  **expands stem by factor of 2.**

Dotplot: stripchart(MyData$X, xlab = "X Values", method = "stack")

**Tables Commands, Chi-Square test, and Graphic Displays (Categorical Data)**

**Univariate Analysis (One Variable)**

**Frequency, Percent Tables**

Table1 <- table(MyData$W) #Makes the table and stores it in Table1.

Table1 #Prints table of frequencies (or counts).

(Table1 <- table(MyData$W)) #Makes table, stores table as Table 1, prints Table1.

Table1\_Per <- prop.table(Table1)\*100 #Converts Table1 from counts to percentages.

Table1\_Per #Prints table of percentages.

**Graphic Displays of Frequency Tables**

pie(Table1\_Per) #Notice that the input for pie (to make a piechart) must be a table.

barplot(Table1\_Per, col = c(“red”, “blue”), ylab = “Percent”) # For this table, the vertical

scale will be percent. I’ve added different colors for the bars.

**Bivariate Analysis – Two Categorical Variables – we’ll call them Cat$A and Cat$B**

**Frequency Tables**

TableAB <- table(Cat$A, Cat$B) #Makes the two-way table of frequencies (counts).

TableAB #Prints the two-way table.

TableAB\_margins <- addmargins(TableAB) #Adds marginal totals to table. Do not store

as Table AB or you will overwrite your original table.

TableAB\_margins #Prints table with marginal totals.

**Percents Tables**

options(digits = 4) #Limits number of digits in table for all tables printed after this

command is run.

TableAB\_joint <- prop.table(TableAB)\*100 #Makes a two-way table of the **joint**

percents ; percents sum to 100.

TableAB\_joint #Prints joint distribution; one distribution; sum of percents = 100.

TableAB\_row <- prop.table(TableAB,1)\*100 #Makes a table of row percents. This gives

conditional distributions of the *column*

*variable* for each level of the *row var-*

*iable*. Sum each row = 100. One

distribution per row. (Here row variable is the explanatory variable and column variable is the response variable.)

TableAB\_row #Prints row percents table.

TableAB\_col <- prop.table(TableAB,2)\*100 #Makes a table of column percents. This gives conditional distributions of the row

variable for each level of the column variable. Sum each column = 100. One

distribution per column.

TableAB\_col #Prints column percents table.

**Pearson’s** **Chi-Square Test**

(Xsq <- chisq.test(MyTable, **correct = FALSE**)) #Computes Pearson’s chi-square test;

stores and prints results.

Xsq$expected # Prints table of expected counts.

Xsq$p.value #Prints the *p*-value.

**Clustered Bar Graphs (Example with row percents)**

barplot(TableAB\_row, beside = TRUE) #Makes a barchart of the row percents, which

means that the column variable is viewed as the response variable.

barplot(TableAB\_row, beside = TRUE, xlab = “name of column variable”,

ylab = “Percent”, legend = rownames(TableAB \_row)) #Adds axes labels and a

legend.

Warning! Always check that you have an appropriate graphic display and that your labels for the axes make sense.

**Example from Lab 3, Question 6 – Education (rows) and Voting (columns)**

Entered Table

U <- as.table(rbind(c(57,64), c(227,163), c(271,93), c(303,51))) #Entered one row at a time

dimnames(U) <- list(Education = c("Not HS Grad", "HS Grad/No Col",

"Some Col", "BA or Higher"),

Voted = c("Yes", "No")) #Labeled Row Variable and its outcomes; Labeled column

variable and its outcomes.

Calculated Table of Row Percents

U\_Row\_Perc <- prop.table(U,1)\*100 #Created conditional distributions of column variable for each level

of row variable.

Created Bar Chart (with legend and percents on graph)

bp\_Ed\_Vote <- barplot(U\_Row\_Perc, ylab = "Percents",xlab = "Vote", #Note: xlab comes from columns.

main = "Voting Status By Education",

beside = TRUE, col = c("cadetblue1", "cadetblue2",

"cadetblue3", "cadetblue4")) #Note: Number of colors is same as number rows.

legend("topright",legend = rownames(U\_Row\_Perc),

c("Not HS Grad", "HS Grad/No Col","Some Col", "BA or Higher"),

fill = c("cadetblue1", "cadetblue2","cadetblue3", "cadetblue4")) #Note: colors match colors above.

text(bp\_Ed\_Vote,0,round(U\_Row\_Perc, digits = 1),cex = 1, pos = 3)