# In these notes we will begin the process of analyzing data.
# We will begin with descriptive statistics for one and two variables.
# Including Visualizations.

callsurvey <- read.csv("Fallsurvey.csv", header = TRUE)

str(callsurvey)

#~~~~~~~~~~~~~~~
# code chunk 1  #
# Basic Descriptive Stats#
# Quantitative Vars  #
#~~~~~~~~~~~~~~~#

attach(callsurvey)

#You can start by running the equivalent of a Proc means on the whole dataframe by calling Summary:
#this also provides the 5 number summary:

summary(callsurvey)

summary(callsurvey[,5:7])
#instead of identifying the order of the columns, we can identify the names of the columns:


print(mysummary)
str(fallsurvey)

#this can be a little overwhelming. To surgically identify the mean/median/std:

mean(Adj.GPA)

#You generated "NA"...what does this mean? Very simply it means that there were missing
#values in the data. So, you want to run the analysis on the valid observations.

mean(Adj.GPA,na.rm=TRUE)
median(Adj.GPA,na.rm=TRUE)
sd(Adj.GPA,na.rm=TRUE)

#you can also impute the missing values with the mean or median...

fallsurvey$Adj.GPA[is.na(fallsurvey$Adj.GPA)] <- round(median(fallsurvey$Adj.GPA, na.rm = TRUE))

mean(fallsurvey$Adj.GPA)

#this is an option (omits all missing)...but consider the implications:

?na.omit

newdata <- na.omit(fallsurvey)
dim(newdata)

attach(newdata)

?round

#Confidence Intervals are important, but require a bit more effort:

alpha<-.05
n<-sum(!is.na(Adj.GPA))
lclm<-round(mean(Adj.GPA,na.rm=TRUE)-qt(1-alpha/2,n-1)*sd(Adj.GPA,na.rm=TRUE)/sqrt(n),digits=3)
uclm<-round(mean(Adj.GPA,na.rm=TRUE)+qt(1-alpha/2,n-1)*sd(Adj.GPA,na.rm=TRUE)/sqrt(n),digits=3)
mean<-round(mean(Adj.GPA,na.rm=TRUE),digits=3)
limits<-cbind(n,mean,lclm,uclm)
limits

?is.na

#an important part of "data discovery" is the visualization of the data.
#for quantitative variables, this is the histogram and the boxplot.
#The graphics in R are pretty good.

?hist

#simple histogram - gets the job done - but not great:
hist(Adj.GPA)
hist(Adj.GPA, freq=FALSE, col="Blue", main="Figure i: Histogram of Adjusted GPA", xlab="Adjusted GPA", ylab="Percentage", xlim= c(0,4))

boxplot(Adj.GPA, col="Red", main="Figure i: Boxplot of Adjusted GPA", xlab="Adjusted GPA")

#If you want to create a picture that you can use in other packages:

jpeg("figure1.jpg")

hist(fallsurvey$Adj.GPA, freq=FALSE, col="Blue", main="Figure i: Histogram of Adjusted GPA", xlab="Adjusted GPA", ylab="Percentage", xlim= c(0,4))

dev.off()

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
# code chunk 2  #
# Basic Descriptive Stats#
# Qualitative Vars  #
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#

#For Qualitative Variables, we typically want to see frequencies and relative percentages.

summary(fallsurvey[,c("Beatles.vs.Elvis","Parent.s.Music")])

table(Beatles.vs.Elvis)

#There are three blanks in there...how do we get rid of those?

test<-fallsurvey[which(fallsurvey$Beatles.vs.Elvis != " "),]

table(test$Beatles.vs.Elvis)
levels(test$Beatles.vs.Elvis)

?droplevels

fallsurvey1 <- subset(fallsurvey, Beatles.vs.Elvis != ""

table(fallsurvey1$Beatles.vs.Elvis)

table(droplevels(fallsurvey1)$Beatles.vs.Elvis)

# we can also use the apply function...

?apply

apply(fallsurvey[,c("Beatles.vs.Elvis","Parent.s.Music")],2,table)

# question - think about what would happen if we set the 2 to a 1?

mytable< table(droplevels(fallsurvey1)$Beatles.vs.Elvis)

mytable

prop.table(mytable)

mypercents<round(prop.table(mytable)*100,2)

mypercents

# The typical visualizations for a single qualitative variable tends to be the pie chart:

BvE< table(droplevels(fallsurvey1)$Beatles.vs.Elvis)

pie(BvE)
?pie

# we can add some color:

colors<-c("Red", "Blue")

pie(BvE,label=c("Beatles","Elvis"), col=colors, main="Beatles versus Elvis")

mylabels<-paste(names(mytable),", n = ",mytable, ", mypercents,"%",sep="")

pie(mytable,labels = mylabels, col=colors,

  main="Pie Chart of Beatles and Elvis")

# or the bar chart:

BvE<-table(Beatles.vs.Elvis)

barplot(BvE)

?barplot

barplot(table(Beatles.vs.Elvis), col=colors, main = "Figure i: Bar Chart of Beatles and Elvis",

  xlab = "Number of Votes")

# we may want to rotate the barplot:

?barplot

barplot(table(Beatles.vs.Elvis), col=colors, horiz=TRUE, main = "Figure i: Bar Chart of Beatles and Elvis",

  xlab = "Number of Votes")