```
## Sneheil##
# Set working directory and import dataframe
setwd("C:\\Users\\Sneheil Saxena\\Desktop\\Math 189\\Case Study 2")
followUp <- read.table("videoMultiple.txt", header=TRUE)
survey <- read.table("videodata.txt", header=TRUE)

# Preprocess the data
followUp <- na.exclude (followUp)

# Get number of rows for data processing
row <- nrow (followUp)

# Create a "dictionary" to store all the proportions of genres
prop <- vector(mode="list", length=5)
names(prop) <- c(names(followUp[, 1:5]))

# Loop through the genres in the dataframe
for (i in names(followUp[, 1:5])) {
  prop[[i]] <- sum(followUp[[i]] == 1)/row
}

# Scenario 2

# Cleaning the 99s

# Creating a vector of the indices where freq < 99
# and removing the invalid entries
validIndices <- which(survey['freq'] < 99)
survey <- survey[validIndices, ]

validIndices <- which(survey['busy'] < 99)
survey <- survey[validIndices, ]

avgHrsPlayed <- numeric (length = 4)

# Getting total number of hours played last week grouped by reported frequency
for (i in 1:4) {
  # getting hours for freq == i, where i is the reported frequency
  validIndices <- which(survey['freq'] == i)
  surveyTemp <- survey[validIndices, ]
  avgHrsPlayed[i] = sum(surveyTemp$time)/nrow(surveyTemp)
}
```

```
#get data of nonbusy students
validIndices <- which(survey['busy'] == 0)
nonbusy <- survey[validIndices,]
#vector to store numeric populations of each frequency grouping
nonbusyPop <- numeric(length = 4)

#getting avg number of hours played by nonbusy students last week grouped by reported frequency
for (i in 1:4) {
  # getting hours for freq = i, where i is the frequency category
  validIndices <- which(nonbusy['freq'] == i)
  nonbusyTemp <- nonbusy[validIndices,]
  avgHrsPlayed[i] <- sum(nonbusyTemp$time/nrow(nonbusyTemp))
  #store the population of the frequency grouping
  nonbusyPop[i] <- nrow(nonbusyTemp)
}

nonBusyPopHrs <- avgHrsPlayed

#get data of busy students
validIndices <- which(survey["busy"] == 1)
busy <- survey[validIndices,]
#vector to store numeric populations of each frequency grouping
busyPop <- numeric(length = 4)

#getting avg number of hours played by busy students last week grouped by reported frequency
for (i in 1:4) {
  # getting hours for freq = i, where i is the frequency category
  validIndices <- which(busy['freq'] == i)
  busyTemp <- busy[validIndices,]
  avgHrsPlayed[i] <- sum(busyTemp$time/nrow(busyTemp))
  #store the population of the frequency grouping
  busyPop[i] <- nrow(busyTemp)
}

busyPopHrs <- avgHrsPlayed

busyDF <- data.frame("Time"=busyPopHrs, "Frequency"=c("1","2","3","4"))

ggplot(busyDF, aes(x=Frequency, y = Time)) +
   geom_bar(stat = "identity", alpha = 0.75, fill = "pink", width = 0.7) +
   labs(title="           Avg. Time Spent Playing During Week Prior To Survey
                      Grouped By Frequency of Playing
```

```
          (For students who play even when they are busy)",
     x="Frequency of playing (Categorical Variable)", y = "Time (In Hrs)")

nonBusyDF <- data.frame("Time"=nonBusyPopHrs, "Frequency"=c("1","2","3","4"))

ggplot(nonBusyDF, aes(x=Frequency, y = Time)) +
 geom_bar(stat = "identity", alpha = 0.75, fill = "pink", width = 0.7) +
 labs(title="          Avg. Time Spent Playing During Week Prior To Survey
                    Grouped By Frequency of Playing
          (For students who do not play when they are busy)",
     x="Frequency of playing (Categorical Variable)", y = "Time (In Hrs)")

# Additional Investigation
followUp <- read.table("videoMultiple.txt", header=TRUE)
survey <- read.table("videodata.txt", header=TRUE)

# Preprocessing
validIndices <- which(survey["grade"] < 5)
survey <- survey[validIndices,]

# Extract the busy vs nonbusy data for investigation
validIndices <- which(survey["busy"] == 1)
busy <- survey[validIndices,]
validIndices <- which(survey["busy"] == 0)
nonbusy <- survey[validIndices,]

# Calculate the averages of the expected grades
busyAvg = mean(busy$grade)
nonbusyAvg = mean(nonbusy$grade)

# Create bar graph
gradeComparisonDF <- data.frame("Averages"=c(busyAvg, nonbusyAvg),
                "Category"=c("Play When Busy","Don't Play When Busy"))

ggplot(gradeComparisonDF, aes(x=Category, y = Averages)) +
 geom_bar(stat = "identity", alpha = 0.85, fill = "pink", width = 0.3, col = "pink") +
 labs(title="        Avg. Expected Grade (numeric representation)
      Based on Whether Students Played When Busy",
     x="", y = "Avg. Grade") +
 ylim(0, 4)

######Scenario 2 Code JOSH#######
```

```
#cleaning the data of invalid answers
validIndices <- which(survey['freq'] < 99)
survey <- survey[validIndices, ]

#create vector to store means of total number of hours
avgHoursPlayed <- numeric(length = 4)

#getting average number of hours played last week grouped by reported frequency
for (i in 1:4) {
  # getting hours for freq = i, where i is the frequency category
  validIndices <- which(survey['freq'] == i)
  surveyTemp <- survey[validIndices,]
  avgHoursPlayed[i] = sum(surveyTemp$time/nrow(surveyTemp))
}

#get data of nonbusy students
validIndices <- which(survey['busy'] == 0)
nonbusy <- survey[validIndices,]
#vector to store numeric populations of each frequency grouping
nonbusyPop <- numeric(length = 4)

#getting avg number of hours played by nonbusy students last week grouped by reported frequency
for (i in 1:4) {
  # getting hours for freq = i, where i is the frequency category
  validIndices <- which(nonbusy['freq'] == i)
  nonbusyTemp <- nonbusy[validIndices,]
  avgHoursPlayed[i] <- sum(nonbusyTemp$time/nrow(nonbusyTemp))
  #store the population of the frequency grouping
  nonbusyPop[i] <- nrow(nonbusyTemp)
}

#get data of busy students
validIndices <- which(survey["busy"] == 1)
busy <- survey[validIndices,]
#vector to store numeric populations of each frequency grouping
busyPop <- numeric(length = 4)

#getting avg number of hours played by busy students last week grouped by reported frequency
for (i in 1:4) {
  # getting hours for freq = i, where i is the frequency category
  validIndices <- which(busy['freq'] == i)
  busyTemp <- busy[validIndices,]
  avgHoursPlayed[i] <- sum(busyTemp$time/nrow(busyTemp))
```

```
  #store the population of the frequency grouping
  busyPop[i] <- nrow(busyTemp)
}

######SCENARIO 6 CODE JOSH######
#get the expected grades from the data
grades <- survey$grade
#put the grades into a table
#gradesT <- table(grades)
#barplot(gradesT, main = "Expected Grade Distribution", xlab = "Expected Grades",
#        ylab = "Quantity of Students", col = "cornflowerblue", names = c("C", "B", "A"))

#create a vector to hold the quantity of each grade category from the grades data
proportion <- c(0,0,0,0,.1,.4,.3,.2 )

#increment the count per grade category according to the expected grades
for (i in grades) {
  if (i == 1){
    proportion[1] <- proportion[1] + 1
  }
  else if (i == 2) {
    proportion[2] <- proportion[2] + 1
  }
  else if (i == 3) {
    proportion[3] <- proportion[3] + 1
  }
  else {
    proportion[4] <- proportion[4] + 1
  }
}

#find the proportion of the grade category population per category
for (i in 1:4) {
  proportion[i] <- proportion[i] / row
}

#create a dataframe of the expected and target grades to plot
dist <- data.frame(char = rep( c("Expected", "Target"), each = 4),
                   grade = rep(c("D/F", "C", "B", "A"), 2),
                   proportion = rep(proportion))

#plot the dataframe
library(ggplot2)
```

```
ggplot(data=dist, aes(x=grade, y=proportion, fill=char)) +
  geom_bar(stat="identity", position=position_dodge()) +
  ggtitle("Expected vs Target Grade Distributions") + xlab("Grade") + ylab("Proportion")


==================Kevin's section================

#
ggplot(data=survey, aes(survey$time)) +
  geom_histogram(breaks=seq(0, 40, by = 1),
                 col="black",
                 fill="pink",
                 alpha = .8) +
  labs(title="                Frequency of people playing video games in varying lengths of time") +
  labs(x="Amount of Time", y="Frequency in the Student Population") + ylim(0,60) +
geom_vline(aes(xintercept=mean(time)), color="black",
                                                                  linetype="dashed")

standardSkewness <- c()
standardKurtosis <- c()
videoSkewness <- c()
videoKurtosis <- c()
sampleMean <- c()

for(i in 1:1000) {
  #use MonteCarlo to obtain normal
  standardSkewness <- c(standardSkewness, skewness(rnorm(nrow(survey))))
  standardKurtosis <- c(standardKurtosis, kurtosis(rnorm(nrow(survey))))

  #bootstrap
  videoSkewness <- c(videoSkewness, skewness(sample(survey$time,
                                          size=nrow(survey),
                                          replace=TRUE)))
  videoKurtosis <- c(videoKurtosis, kurtosis(sample(survey$time,
                                          size=nrow(survey),
                                          replace=TRUE)))
  sampleMean <- c(sampleMean, mean(sample(survey$time,
                                  size=nrow(survey),
                                  replace=TRUE)))
}

hist(main='Distribution of 1000 Bootstrapped Sample Means',
    sampleMean,
```

```
    col="blue",
    ylim=c(0,250),
    alpha = .6,
    xlab = "Amount of Time")

sk <- NULL
kur <- NULL
for(i in 1:1000) {
  sk <- c(sk, skewness(rnorm(91)))
  kur <- c(kur, kurtosis(rnorm(91)))
}

hist(main='Distribution of 1000 Bootstrapped Sample Kurtosis',
    kur,
    col="cornflower blue",
    ylim=c(0,550),
    xlab = "Amount of Time")

hist(main='Distribution of 1000 Bootstrapped Sample Skewness',
    sk,
    col="cornflower blue",
    ylim=c(0,350),
    xlab = "Normal Skewness")



means <- c()
pop <- rep(survey$time > 0, length.out = 314)

for(i in 1:1000){
  means <- c(means, sum(sample(pop, size = 91, replace = FALSE)) / 91)
}


hist(means, main = 'Distribution of 1000 Bootstrapped Proportion Sample Means', ylim=c(0,250),
xlab = 'Proportion of Students who Played Video Games', col = "cornflower blue")

play <- survey$time
frequency <- survey$freq
boxplot(play~frequency, data = survey, main = 'Time playing video games based on frequency',
xlab = 'Frequency of play', ylab = 'Number of hours during the week prior to survey', names =
c('Daily (1)','Weekly (2)','Monthly (3)','Semesterly (4)'))
```

```
like.data <- survey[which(survey$like != 1), ]
like.data$like[like.data$like == 2 | like.data$like == 3] <- "Like"
like.data$like[like.data$like == 4 | like.data$like == 5] <- "Dislike"
#remove those who did not answer who worked
like.data <- like.data[which(!is.na(like.data$work)),]

#change name of "sex" value
like.data$sex[like.data$sex == 0] <- "Female"
like.data$sex[like.data$sex == 1] <- "Male"
#change name of "work" value
like.data$work[like.data$work > 0] <- "Work"
like.data$work[like.data$work == 0] <- "No work"
#change name of "own" value
like.data$own[like.data$own == 0] <- "Does not own PC"
like.data$own[like.data$own == 1] <- "Own PC"

numcount <- table(like.data$sex, like.data$like)
barplot(numcount, main = 'Preference towards Video Games by Sex', xlab = 'Video game
preference ', ylab = 'Frequency', col = c('light yellow', 'light blue'), legend = rownames(numcount),
beside = TRUE)


numcount2 <- table(like.data$work, like.data$like)
barplot(numcount2, main = 'Preference towards Video Games by Work', xlab = 'Video game
preference ', ylab = 'Frequency', col = c('light yellow', 'light blue'), legend = rownames(numcount2),
beside = TRUE)

numcount3 <- table(like.data$own, like.data$like)
barplot(numcount3, main = 'Preference towards Video Games by Ownership of a PC', xlab =
'Video game preference ', ylab = 'Frequency', col = c('light yellow', 'light blue'), legend =
rownames(numcount3), beside = TRUE)


CrossTable(like.data$like, like.data$sex)
CrossTable(like.data$like, like.data$work)
CrossTable(like.data$like, like.data$own)


##############Tony Code################
library(moments)

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

```
data.population = 314  #N
data.sample = 91   #n
#clean data, set unanswered results to NA
data[data == 99] <- NA

#Scenario 1
#number of players in the last week
players <- length(which(data$time > 0))
players.proportion <- players / data.sample

#95% confidence interval, not accounting for non iid
players.error_margin1 <- qnorm(0.975) * sqrt(players.proportion * (1 - players.proportion) /
data.sample)
players.standard_lower <- players.proportion - players.error_margin1
players.standard_upper <- players.proportion + players.error_margin1
data.players_95CI_standard <- c(players.standard_lower, players.standard_upper)
#95% confidence interval, accounting for non iid
players.error_margin2 <- qnorm(0.975) * sqrt(((players.proportion * (1 - players.proportion)) /
                                (data.sample-1))*(data.population-data.sample) / data.population)
players.corrected_lower <- players.proportion - players.error_margin2
players.corrected_upper <- players.proportion + players.error_margin2
data.players_95CI_corrected <- c(players.corrected_lower, players.corrected_upper)

#bootstrap
set.seed(0)
means <- c()
pop <- rep(data$time > 0, length.out = 314)

for(i in 1:1000){
  means <- c(means, sum(sample(pop, size = 91, replace = FALSE)) / 91)
}

hist(means, main = 'Distribution of 1000 Bootstrapped Proportion Sample Means', ylim=c(0,250),
xlab = 'Proportion of Students who Played Video Games', col = "cornflower blue")
abline(v=mean(means),col="red")


#Scenario 3
time.mean <- mean(data$time)

#95% confidence interval, standard
time.error_margin1 <- qnorm(0.975) * sd(data$time) / sqrt(data.sample)
time.standard_lower <- time.mean - time.error_margin1
```

```
time.standard_upper <- time.mean + time.error_margin1
data.time_95CI_standard <- c(time.standard_lower, time.standard_upper)
#95% confidence interval, corrected
time.error_margin2 <- qnorm(0.975) * sd(data$time) / sqrt(data.sample) * sqrt((data.population -
data.sample)/data.population)
time.corrected_lower <- time.mean - time.error_margin2
time.corrected_upper <- time.mean + time.error_margin2
data.time_95CI_corrected <- c(time.corrected_lower, time.corrected_upper)
#95% confidence interval using simulation study, bootstrapping
time.bootstrap <- c()
time.bootstrap_population <- rep(data$time, length.out = data.population)
for(i in 1:1000) {
  time.bootstrap <- c(time.bootstrap, mean(
                    sample(time.bootstrap_population, size=nrow(data),replace=FALSE)))
}
hist(time.bootstrap,
    main='Distribution of 1000 Bootstrapped Sample Means',
    col="cornflowerblue",
    ylim=c(0,250),
    xlab = "Amount of Time")

time.bootstrap_mean <- mean(time.bootstrap)
time.error_margin3 <- qnorm(0.975) * sd(time.bootstrap) / sqrt(data.sample)
time.bootstrap_lower <- time.bootstrap_mean - time.error_margin3
time.bootstrap_upper <- time.bootstrap_mean + time.error_margin3
data.time_95CI_bootstrap <- c(time.bootstrap_lower, time.bootstrap_upper)

#observe difference in skewness
skewness(data$time)
kurtosis(data$time)

skewness(time.bootstrap)
kurtosis(time.bootstrap)

#compare distributions
normal_test <- rnorm(data.sample)
#normalize to standard units
time.bootstrap <-(time.bootstrap - time.bootstrap_mean) / sd(time.bootstrap)
time.su = (data$time - mean(data$time)) / sd(data$time)

ks.test(time.su, normal_test)
ks.test(time.bootstrap, normal_test)
```

```
#Scenario 5
#clean data, remove if never played video games,
#group somewhat liking/very much like to Like
#group not really liking/not like at all to Dislike
like.data <- data[which(data$like != 1), ]
like.data$like[like.data$like == 2 | like.data$like == 3] <- "Like"
like.data$like[like.data$like == 4 | like.data$like == 5] <- "Dislike"
#remove those who did not answer who worked
like.data <- like.data[which(!is.na(like.data$work)),]

#change name of "sex" value
like.data$sex[like.data$sex == 0] <- "Female"
like.data$sex[like.data$sex == 1] <- "Male"
#change name of "work" value
like.data$work[like.data$work > 0] <- "Work"
like.data$work[like.data$work == 0] <- "No work"
#change name of "own" value
like.data$own[like.data$own == 0] <- "Does not own PC"
like.data$own[like.data$own == 1] <- "Own PC"

library(gmodels)
CrossTable(like.data$like, like.data$sex)
CrossTable(like.data$like, like.data$Work)
CrossTable(like.data$like, like.data$own)


######################


# Get raw numbers
# Figure out proportions
custom1 <- matrix(c((12/38), (26/38), (8/48), (40/48)),ncol=2,byrow=TRUE)
colnames(custom1) <- c("Dislike","Like")
rownames(custom1) <- c("Female","Male")
custom1 <- as.table(custom1)

# plot proportions
barplot(custom1, main = 'Preference towards Video Games by Sex',
        xlab = 'Video game preference ', ylab = 'Proportion',
        col = c('light yellow', 'light blue'),
        legend = rownames(numcount),
        beside = TRUE)
```

```
numcount2 <- table(like.data$work, like.data$like)

# Figure out proportions
custom2 <- matrix(c((14/44), (30/44), (6/42), (36/42)),ncol=2,byrow=TRUE)
colnames(custom2) <- c("Dislike","Like")
rownames(custom2) <- c("No work","Work")
custom2 <- as.table(custom2)

barplot(custom2, main = 'Preference towards Video Games by Work',
      xlab = 'Video game preference ', ylab = 'Proportion',
      col = c('light yellow', 'light blue'),
      legend = rownames(numcount2), beside = TRUE)

numcount3 <- table(like.data$own, like.data$like)

# Figure out proportions
custom3 <- matrix(c((3/23), (20/23), (17/63), (46/63)),ncol=2,byrow=TRUE)
colnames(custom3) <- c("Dislike","Like")
rownames(custom3) <- c("No work","Work")
custom3 <- as.table(custom2)
```