```r
# set working directory
setwd("D:/STORAGE/College Work/year 4/Y4Q3/MATH 189/hw/hw2")

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

# replace 99 with NA
data[data == 99] <- NA




# scenario 1
# What proportion of students played a videogame a during the week before an exam
# Counting those that play daily and weekly while excluding NA values

data$freq[data$freq == 1] <- 1
data$freq[data$freq == 2] <- 1
data$freq[data$freq != 1] <- 0
data$freq[is.na(data$freq)] <- 0

freq.percentage = mean(data$freq)

shuffle.ind=sample(1:nrow(data))

# Creates a simulated population based on the single sample VIA bootstrap
boot.population <- rep(data$freq[shuffle.ind], length.out = 314)
print(boot.population)
# In the population replace all those who play games daily or monthly with a 1 and all those
# who don't with a 0. This allows us to later take the mean to easily get a proportion...

B = 500 # the number of bootstrap samples we want

# Runs 500 times to sample
boot.sample <- array(dim = c(B, 91))
for (i in 1:B) {
  boot.sample[i, ] <- sample(boot.population, size = 91, replace = FALSE)
}

# Calculate the means for each individual bootstrap
boot.mean <- apply(X = boot.sample, MARGIN = 1, FUN = mean)

hist(boot.mean, breaks = 20, probability = TRUE, density = 20, col = 3, border = 3)
lines(density(boot.mean, adjust = 2), col = 2)

# Identifying if the distribution is normal
par(pty = 's')
qqnorm(boot.mean)
qqline(boot.mean)
ks.test((boot.mean - mean(boot.mean))/sd(boot.mean), pnorm)

# The distribution is relatively normal so I can continue to construct confiedence intervals
boot.sd <- sd(boot.mean)
interval <- freq.percentage + c(-1, 1)*1.96*boot.sd

print(freq.percentage)
print(interval)


# scenario 2
# 1hr/day = 7 hrs
# 1hr/week = 1 hrs
# 1hr/month = 15 mins
mean(data$time[data$freq == 1], na.rm=TRUE)
mean(data$time[data$freq == 2], na.rm=TRUE)
mean(data$time[data$freq == 3], na.rm=TRUE)
mean(data$time[data$freq == 4], na.rm=TRUE)


# scenario 3
data$time[is.na(data$time)] <- 0

like.avg = mean(data$time, na.rm = TRUE)

print(like.avg)
shuffle.ind=sample(1:nrow(data))

# Creates a simulated population based on the single sample VIA bootstrap
boot.population <- rep(data$time[shuffle.ind], length.out = 314)
# In the population replace all those who play games daily or monthly with a 1 and all those
# who don't with a 0. This allows us to later take the mean to easily get a proportion...

mean(boot.population)
hist(boot.population, breaks = 30, probability = TRUE, density = 30, col = 3, border = 3)
```

```
B = 500 # the number of bootstrap samples we want

# Runs 500 times to sample
boot.sample <- array(dim = c(B, 91))
for (i in 1:B) {
  boot.sample[i, ] <- sample(boot.population, size = 91, replace = FALSE)
}

# Calculate the means for each individual bootstrap
boot.mean <- apply(X = boot.sample, MARGIN = 1, FUN = mean)

hist(boot.mean, breaks = 20, probability = TRUE, density = 20, col = 3, border = 3)
lines(density(boot.mean, adjust = 2), col = 2)

# Identifying if the distribution is normal
par(pty = 's')
qqnorm(boot.mean)
qqline(boot.mean)
ks.test((boot.mean - mean(boot.mean))/sd(boot.mean), pnorm)

# The distribution is relatively normal so I can continue to construct confiedence intervals
boot.sd <- sd(boot.mean)
interval <- like.avg + c(-1, 1)*1.96*boot.sd

print(like.avg)
print(interval)


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

data[data == 99] <- NA

data$time[is.na(data$time)] <- 0

k <- 100
tab <- table(cut(data$time, breaks = seq(0, 10, length.out = k+1), include.lowest = TRUE))
head(tab, 10)
counts <- as.vector(tab)
head(counts, 10)
# Poisson simulation
hist(counts, breaks = 15, col = rgb(1,0,0,0.5), probability = TRUE, xlab = "number of points inside an interval",
ylim = c(0,1))
lines(density(counts, adjust = 2), col = rgb(1,0,0,0.5))
Pois <- rpois(1000, lambda = mean(counts))
hist(Pois, breaks = 15, col = rgb(0,0,1,0.5), probability = TRUE, add = TRUE)
lines(density(Pois, adjust = 2), col = rgb(0,0,1,0.5))
legend(x = 14, y = 0.15, legend = c("sample", "Poisson"), lty = c(1,1), col = c(rgb(1,0,0,0.5), rgb(0,0,1,0.5)))

time_mean <- mean(data$time, na.rm = TRUE)
time_sd <- sd(data$time, na.rm = TRUE)
interval <- c(time_mean - 2*time_sd,
              time_mean + 2*time_sd)
interval


# scenario 4
# Do students enjoy playing video games?
mult_data <- read.table("videoMultiple.txt", header=TRUE)
response_count <- nrow(mult_data)

# Filter out people that don't play video games
game_types <- c("action", "adv", "sim", "sport", "strategy")
game_types_played <- rowSums(mult_data[game_types] == 0)
game_players <- mult_data[(game_types_played != 0) & !(is.na(game_types_played)),]
gamers_count <- nrow(game_players)

# Out of control dislikes:
# too much time, costs too much, friends don't play
out_of_cont_dislikes <- colSums(game_players[c("time", "cost", "friends")], na.rm = TRUE) / gamers_count * 100
# Workable dislikes:
# frustrating, lonely, too many rules, boring, it is pointless
in_cont_dislikes <- colSums(game_players[c("frust", "lonely", "rules", "boring", "point")], na.rm = TRUE) /
gamers_count * 100

# Of people that play games, these people dislike them for external reasons
# A large percentage of people are in this category
out_of_cont_dislikes
# of people that play games, these people dislike them for game-related reasons
# A smaller amount of people are in these categories
in_cont_dislikes

gamers_data <- data[which((data$like != 1) & !(is.na(data$like))),]
```

```
sum(gamers_data$like <= 3)
sum(gamers_data$like > 3)


### SCENARIO 5 ###

# Cross-tables and plots
#install.packages("gmodels")
library(gmodels)

# Collapse the range of responses?
data.clean <- data[which(data$like != 1),] # Clean out the "Never played" data
# Regroup the 'like' value
data.clean$like[data.clean$like == 2 | data.clean$like == 3] <- "Like"
data.clean$like[data.clean$like == 4 | data.clean$like == 5] <- "Dislike"

## Like vs Sex

# Regroup the 'sex' value
data.clean$sex[data.clean$sex == 0 ] <- "Female"
data.clean$sex[data.clean$sex == 1] <- "Male"

# Cross tabulations between like and sex
CrossTable(data.clean$like, data.clean$sex)
chisq.test(table(data.clean$like, data.clean$sex))

# Bar Graph
counts <- table(data.clean$like, data.clean$sex)
barplot(counts, main = "Game Preference by Gender",
        xlab='Respondents', ylab = 'Frequency Count',
        col=c('black','white'), legend = rownames(counts), beside=TRUE)

## Like vs Work

# Regroup the 'work' value
data.clean$work[data.clean$work > 0 ] <- "Work"
data.clean$work[data.clean$work == 0] <- "No Work"

# Cross tabulations between like and work
CrossTable(data.clean$like, data.clean$work)
chisq.test(table(data.clean$like, data.clean$work))

# Bar Graph
counts <- table(data.clean$like, data.clean$work)
barplot(counts, main = "Game Preference by Employment Status",
        xlab='Respondents Who...', ylab = 'Frequency Count',
        col=c('black','white'), legend = rownames(counts), beside=TRUE)

#Boxplot "hrs worked vs like playing"
boxplot(data$work~data$like, main = "Hours Worked vs Like Playing",
        xlab = "Like Playing Category",
        ylab = "Number of Hours Worked",
        names = c("Never", "Very Much", "Somewhat", "Not Really", "Not at All"))

## Like vs Own

# Regroup the 'own' value
data.clean$own[data.clean$own == 0 ] <- "No PC"
data.clean$own[data.clean$own == 1] <- "Own PC"

# Cross tabulations between like and own
CrossTable(data.clean$like, data.clean$own)
chisq.test(table(data.clean$like, data.clean$own))

# Bar Graph
counts <- table(data.clean$like, data.clean$own)
barplot(counts, main = "Game Preference by PC Ownership",
        xlab='Respondents Who...', ylab = 'Frequency Count',
        col=c('black','white'), legend = rownames(counts), beside=TRUE)


### SCENARIO 6 ###

table<-table(data$grade)
barplot(table, main = "Number of Students by Expected Grade",
        xlab = "Grade Expected", ylab = "# Students",
        name=c("C","B","A"))
table # gives counts
#proportion_C = length(which(data$grade==2))/sample
#proportion_B = length(which(data$grade==3))/sample
#proportion_A = length(which(data$grade==4))/sample

numA = sum(data$grade == 4 & !is.na(data$grade))
numB = sum(data$grade == 3 & !is.na(data$grade))
numC = sum(data$grade == 2 & !is.na(data$grade))
```

```
numD = sum(data$grade == 1 & !is.na(data$grade))
numNA = sum(is.na(data$grade)) #0
Tgrade <- matrix(c(numA,numB,numC,numD,numA/91,numB/91,numC/91,numD/91),ncol = 4, byrow = TRUE)
colnames(Tgrade) <- c("Expected A","Expected B","Expected C","Expected D")
rownames(Tgrade) <- c("Counts","Percentage")
Tgrade <- as.table(Tgrade)
Tgrade
```