StatsLab Code

Justin Glommen (Scenarios 1,4)

Victor Angulo (Scenarios 3,6)

Atharva Fulay (Scenario 2)

Peter Yao (Scenario 5)

2/5/2017

## Download necessary packages

# Install gmodels for cross-tabulation  
install.packages('gmodels',repos="http://cran.rstudio.com/")

# Data Management

Loading data from current directory

data <- read.table("videodata.txt", header=TRUE)  
data.population <- 314 # True population  
data.samples <- 91 # Number of samples  
head(data)

## time like where freq busy educ sex age home math work own cdrom email  
## 1 2.0 3 3 2 0 1 0 19 1 0 10 1 0 1  
## 2 0.0 3 3 3 0 0 0 18 1 1 0 1 1 1  
## 3 0.0 3 1 3 0 0 1 19 1 0 0 1 0 1  
## 4 0.5 3 3 3 0 1 0 19 1 0 0 1 0 1  
## 5 0.0 3 3 4 0 1 0 19 1 1 0 0 0 1  
## 6 0.0 3 2 4 0 0 1 19 0 0 12 0 0 0  
## grade  
## 1 4  
## 2 2  
## 3 3  
## 4 3  
## 5 3  
## 6 3

summary(data)

## time like where freq   
## Min. : 0.000 Min. : 1.000 Min. : 1.00 Min. : 1.00   
## 1st Qu.: 0.000 1st Qu.: 2.000 1st Qu.: 3.00 1st Qu.: 2.00   
## Median : 0.000 Median : 3.000 Median : 3.00 Median : 3.00   
## Mean : 1.243 Mean : 4.077 Mean :21.97 Mean :16.46   
## 3rd Qu.: 1.250 3rd Qu.: 3.000 3rd Qu.: 5.00 3rd Qu.: 4.00   
## Max. :30.000 Max. :99.000 Max. :99.00 Max. :99.00   
## busy educ sex age   
## Min. : 0.00 Min. : 0.00 Min. :0.0000 Min. :18.00   
## 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.:0.0000 1st Qu.:19.00   
## Median : 0.00 Median : 1.00 Median :1.0000 Median :19.00   
## Mean :12.15 Mean :14.55 Mean :0.5824 Mean :19.52   
## 3rd Qu.: 1.00 3rd Qu.: 1.00 3rd Qu.:1.0000 3rd Qu.:20.00   
## Max. :99.00 Max. :99.00 Max. :1.0000 Max. :33.00   
## home math work own   
## Min. :0.0000 Min. : 0.000 Min. : 0.00 Min. :0.0000   
## 1st Qu.:1.0000 1st Qu.: 0.000 1st Qu.: 0.00 1st Qu.:0.0000   
## Median :1.0000 Median : 0.000 Median : 5.00 Median :1.0000   
## Mean :0.7582 Mean : 1.407 Mean :10.37 Mean :0.7363   
## 3rd Qu.:1.0000 3rd Qu.: 1.000 3rd Qu.:14.50 3rd Qu.:1.0000   
## Max. :1.0000 Max. :99.000 Max. :99.00 Max. :1.0000   
## cdrom email grade   
## Min. : 0.000 Min. :0.0000 Min. :2.000   
## 1st Qu.: 0.000 1st Qu.:1.0000 1st Qu.:3.000   
## Median : 0.000 Median :1.0000 Median :3.000   
## Mean : 5.604 Mean :0.7912 Mean :3.253   
## 3rd Qu.: 0.000 3rd Qu.:1.0000 3rd Qu.:4.000   
## Max. :99.000 Max. :1.0000 Max. :4.000

## Cleaning Data

Replacing 99 values (the unanswered/improper results) with NAs

data[data == 99] <- NA  
numSamples <- NROW(data)  
head(data)

## time like where freq busy educ sex age home math work own cdrom email  
## 1 2.0 3 3 2 0 1 0 19 1 0 10 1 0 1  
## 2 0.0 3 3 3 0 0 0 18 1 1 0 1 1 1  
## 3 0.0 3 1 3 0 0 1 19 1 0 0 1 0 1  
## 4 0.5 3 3 3 0 1 0 19 1 0 0 1 0 1  
## 5 0.0 3 3 4 0 1 0 19 1 1 0 0 0 1  
## 6 0.0 3 2 4 0 0 1 19 0 0 12 0 0 0  
## grade  
## 1 4  
## 2 2  
## 3 3  
## 4 3  
## 5 3  
## 6 3

summary(data)

## time like where freq   
## Min. : 0.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.: 0.000 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:2.000   
## Median : 0.000 Median :3.000 Median :3.000 Median :3.000   
## Mean : 1.243 Mean :3.022 Mean :2.973 Mean :2.705   
## 3rd Qu.: 1.250 3rd Qu.:3.000 3rd Qu.:4.000 3rd Qu.:4.000   
## Max. :30.000 Max. :5.000 Max. :6.000 Max. :4.000   
## NA's :1 NA's :18 NA's :13   
## busy educ sex age   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :18.00   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:19.00   
## Median :0.0000 Median :0.0000 Median :1.0000 Median :19.00   
## Mean :0.2125 Mean :0.4744 Mean :0.5824 Mean :19.52   
## 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:20.00   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :33.00   
## NA's :11 NA's :13   
## home math work own   
## Min. :0.0000 Min. :0.0000 Min. : 0.000 Min. :0.0000   
## 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.: 0.000 1st Qu.:0.0000   
## Median :1.0000 Median :0.0000 Median : 1.000 Median :1.0000   
## Mean :0.7582 Mean :0.3222 Mean : 7.352 Mean :0.7363   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:13.250 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000 Max. :55.000 Max. :1.0000   
## NA's :1 NA's :3   
## cdrom email grade   
## Min. :0.0000 Min. :0.0000 Min. :2.000   
## 1st Qu.:0.0000 1st Qu.:1.0000 1st Qu.:3.000   
## Median :0.0000 Median :1.0000 Median :3.000   
## Mean :0.1744 Mean :0.7912 Mean :3.253   
## 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:4.000   
## Max. :1.0000 Max. :1.0000 Max. :4.000   
## NA's :5

# Scenario 1

## Sample Proportion of Students Who Played a Video Game in the Last Week

The individual variables measured here are Bernoulli since time is being converted to a binary 'did' or 'did not' play.

# Create 'numPlayers' variable to count number of players in the last week.  
# This is done by counting the number of people with time spent over 0, which represents the  
# people who played something in the last week since they spent time on it. 0 indicates no time   
# spent.  
numPlayers <- NROW(which(data$time > 0))  
paste("Number of players:", numPlayers, sep=" ")

## [1] "Number of players: 34"

# Sample proportion is the ratio of numPlayers to total students (rows in data)  
data.playersSampleProportion <- (numPlayers/numSamples)  
paste("Sample proportion:", data.playersSampleProportion, sep=" ")

## [1] "Sample proportion: 0.373626373626374"

## Players Sample Proportion Confidence Interval

Since the sample Bernoulli variables are NOT identically independentally distributed, the confidence interval itself will be computed utilizing the finite population correction factor.

# Sample proportion is nearly Binomial, except not iid.  
playersCorrectionFactor <- sqrt((data.population - numSamples)/data.population)  
# Binomial standard error formula without correction  
playersIndepStandardError <- (sqrt(data.playersSampleProportion\*(1-data.playersSampleProportion))/sqrt(numSamples - 1))  
# Standard error with finite population correction  
data.playersStandardErrorEstimate <- playersIndepStandardError\*playersCorrectionFactor  
paste("Corrected Standard Error:", data.playersStandardErrorEstimate, sep=" ")

## [1] "Corrected Standard Error: 0.0429736108569751"

# Since the sample proportion follows a normal distribution by the Central Limit Theorem,  
# we need to multiply the corrected standard error by 1.96 to generate the interval.  
data.playersMarginOfError <- 1.96\*data.playersStandardErrorEstimate  
paste("Margin of Error: ", data.playersMarginOfError, sep="")

## [1] "Margin of Error: 0.0842282772796712"

# Therefore, the confidence interval:  
playersLowerBound <- data.playersSampleProportion - data.playersMarginOfError  
playersUpperBound <- data.playersSampleProportion + data.playersMarginOfError  
data.playersSampleProportionConf95 <- c(playersLowerBound, playersUpperBound)  
paste("Player Proportion 95% CI: ", "(",playersLowerBound, ", ", playersUpperBound,")", sep="")

## [1] "Player Proportion 95% CI: (0.289398096346702, 0.457854650906045)"

# Scenario 2

smalltime.ind <- which(data$time < 6)  
data.smalltime <- data[smalltime.ind,]  
  
zerohours.ind <- which(data.smalltime$time ==0)  
data.zerohours <- data[zerohours.ind, ]  
mean(data.zerohours$freq, na.rm=TRUE)

## [1] 3

fewhours.ind <- which(data.smalltime$time > 0 & data.smalltime$time <=5 )  
data.fewhours <- data[fewhours.ind, ]  
mean(data.fewhours$freq, na.rm=TRUE)

## [1] 2.206897

manyhours.ind <- which(data$time > 6)  
data.manyhours <- data[manyhours.ind, ]  
summary(data.manyhours$freq, na.rm=TRUE)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 1.000 1.333 1.500 2.000

daily.ind <- which(data$freq == 1)  
weekly.ind <- which(data$freq == 2)  
monthly.ind <- which(data$freq == 3)  
semester.ind <- which(data$freq == 4)  
  
data.daily <- data[daily.ind, ]  
data.weekly <- data[weekly.ind, ]  
data.monthly <- data[monthly.ind, ]  
data.semester <- data[semester.ind, ]  
  
mean(data.daily$time)

## [1] 4.444444

mean(data.weekly$time)

## [1] 2.539286

mean(data.monthly$time)

## [1] 0.05555556

mean(data.semester$time)

## [1] 0.04347826

busy.ind <- which(data$busy == 1)  
data.busy <- data[busy.ind, ]  
  
notbusy.ind <- which(data$busy == 0)  
data.notbusy <- data[notbusy.ind, ]  
  
mean(data.busy$time)

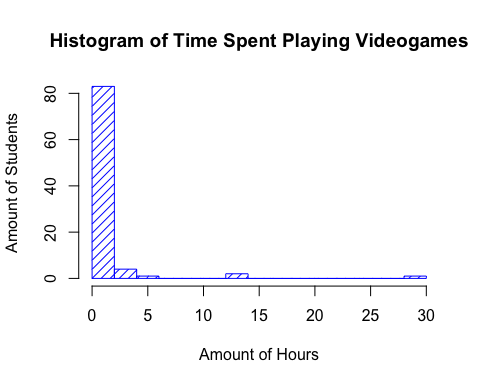
## [1] 4.705882

mean(data.notbusy$time)

## [1] 0.5095238

# Scenario 3

#First we calculate the estimate for the # of students that played a video game:  
nogame.ind <- which(data['time'] == 0.0) #Identify those who did not play video games the week prior  
data.nogame <- data[nogame.ind,] #Create a data frame with no gamers  
n1 <- length(data.nogame$time) #Calculates the # of students that played video games  
prop.nogame <- (n1)/91 #Calculates the proportion (# that don't play/sample size)  
sd.prop.nogame <- sqrt( (.6263736)\*(1-.6263736)/90 )\*sqrt((314-91)/314 ) #Calculates the sd of those that don't game  
prop.nogame.ci <- prop.nogame + c(-1, 1)\*2\*sd.prop.nogame #Creates the CI  
  
#Histogram of sample time spent playing  
hist(data$time, main = "Histogram of Time Spent Playing Videogames", xlab = "Amount of Hours", ylab = "Amount of Students",   
 col = 4, density = 15, breaks = 15)



#Here we do Bootstrap  
boot.population <- rep(data$time, length.out = 314) #Creates the population  
sample1 <- sample(boot.population, size = 91, replace = FALSE) #creates the sample populations  
B = 500 # the number of bootstrap samples we want  
boot.sample <- array(dim = c(B, 91))  
for (i in 1:B)  
 {  
 boot.sample[i, ] <- sample(boot.population, size = 91, replace = FALSE)  
}  
boot.mean <- apply(X = boot.sample, MARGIN = 1, FUN = mean) #Here we take the sample mean of each sample  
ci.boot <- c(quantile(boot.mean,0.025), quantile(boot.mean, 0.975))

# Scenario 4

Getting proportion who likes games.

# Initializing variables corresponding to responses from students on the survey  
likeVeryMuch <- 2  
likeSomewhat <- 3  
# Fetching all students who responded with positive game likeness  
data.likeColumns <- which(data$like == likeVeryMuch)   
data.likeColumns <- c(data.likeColumns, which(data$like == likeSomewhat))  
# Calculating percentage  
numOfLikes <- NROW(data.likeColumns)  
proportionLike <- numOfLikes/data.samples  
paste("Proportion of Like: ", proportionLike, sep="")

## [1] "Proportion of Like: 0.758241758241758"

# Scenario 5

# Using gmodels library  
library(gmodels)  
  
#Cross-Tabulation for owning a computer/like playing games  
CrossTable(data$like, data$own)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | Chi-square contribution |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 90   
##   
##   
## | data$own   
## data$like | 0 | 1 | Row Total |   
## -------------|-----------|-----------|-----------|  
## 1 | 0 | 1 | 1 |   
## | 0.267 | 0.097 | |   
## | 0.000 | 1.000 | 0.011 |   
## | 0.000 | 0.015 | |   
## | 0.000 | 0.011 | |   
## -------------|-----------|-----------|-----------|  
## 2 | 5 | 18 | 23 |   
## | 0.209 | 0.076 | |   
## | 0.217 | 0.783 | 0.256 |   
## | 0.208 | 0.273 | |   
## | 0.056 | 0.200 | |   
## -------------|-----------|-----------|-----------|  
## 3 | 16 | 30 | 46 |   
## | 1.136 | 0.413 | |   
## | 0.348 | 0.652 | 0.511 |   
## | 0.667 | 0.455 | |   
## | 0.178 | 0.333 | |   
## -------------|-----------|-----------|-----------|  
## 4 | 1 | 12 | 13 |   
## | 1.755 | 0.638 | |   
## | 0.077 | 0.923 | 0.144 |   
## | 0.042 | 0.182 | |   
## | 0.011 | 0.133 | |   
## -------------|-----------|-----------|-----------|  
## 5 | 2 | 5 | 7 |   
## | 0.010 | 0.003 | |   
## | 0.286 | 0.714 | 0.078 |   
## | 0.083 | 0.076 | |   
## | 0.022 | 0.056 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 24 | 66 | 90 |   
## | 0.267 | 0.733 | |   
## -------------|-----------|-----------|-----------|  
##   
##

#Cross-Tabulation for working/like playing games  
CrossTable(data$like, data$work==0)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | Chi-square contribution |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 87   
##   
##   
## | data$work == 0   
## data$like | FALSE | TRUE | Row Total |   
## -------------|-----------|-----------|-----------|  
## 1 | 1 | 0 | 1 |   
## | 0.518 | 0.506 | |   
## | 1.000 | 0.000 | 0.011 |   
## | 0.023 | 0.000 | |   
## | 0.011 | 0.000 | |   
## -------------|-----------|-----------|-----------|  
## 2 | 14 | 9 | 23 |   
## | 0.609 | 0.596 | |   
## | 0.609 | 0.391 | 0.264 |   
## | 0.326 | 0.205 | |   
## | 0.161 | 0.103 | |   
## -------------|-----------|-----------|-----------|  
## 3 | 22 | 21 | 43 |   
## | 0.026 | 0.026 | |   
## | 0.512 | 0.488 | 0.494 |   
## | 0.512 | 0.477 | |   
## | 0.253 | 0.241 | |   
## -------------|-----------|-----------|-----------|  
## 4 | 3 | 10 | 13 |   
## | 1.826 | 1.785 | |   
## | 0.231 | 0.769 | 0.149 |   
## | 0.070 | 0.227 | |   
## | 0.034 | 0.115 | |   
## -------------|-----------|-----------|-----------|  
## 5 | 3 | 4 | 7 |   
## | 0.061 | 0.060 | |   
## | 0.429 | 0.571 | 0.080 |   
## | 0.070 | 0.091 | |   
## | 0.034 | 0.046 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 43 | 44 | 87 |   
## | 0.494 | 0.506 | |   
## -------------|-----------|-----------|-----------|  
##   
##

#Cross-Tabultion for sex/like playing games  
CrossTable(data$like, data$sex)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | Chi-square contribution |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 90   
##   
##   
## | data$sex   
## data$like | 0 | 1 | Row Total |   
## -------------|-----------|-----------|-----------|  
## 1 | 0 | 1 | 1 |   
## | 0.422 | 0.309 | |   
## | 0.000 | 1.000 | 0.011 |   
## | 0.000 | 0.019 | |   
## | 0.000 | 0.011 | |   
## -------------|-----------|-----------|-----------|  
## 2 | 5 | 18 | 23 |   
## | 2.285 | 1.670 | |   
## | 0.217 | 0.783 | 0.256 |   
## | 0.132 | 0.346 | |   
## | 0.056 | 0.200 | |   
## -------------|-----------|-----------|-----------|  
## 3 | 21 | 25 | 46 |   
## | 0.128 | 0.094 | |   
## | 0.457 | 0.543 | 0.511 |   
## | 0.553 | 0.481 | |   
## | 0.233 | 0.278 | |   
## -------------|-----------|-----------|-----------|  
## 4 | 8 | 5 | 13 |   
## | 1.149 | 0.840 | |   
## | 0.615 | 0.385 | 0.144 |   
## | 0.211 | 0.096 | |   
## | 0.089 | 0.056 | |   
## -------------|-----------|-----------|-----------|  
## 5 | 4 | 3 | 7 |   
## | 0.369 | 0.270 | |   
## | 0.571 | 0.429 | 0.078 |   
## | 0.105 | 0.058 | |   
## | 0.044 | 0.033 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 38 | 52 | 90 |   
## | 0.422 | 0.578 | |   
## -------------|-----------|-----------|-----------|  
##   
##

# Scenario 6

#Chi-square test  
observed <- c(31, 52, 8, 0)  
expected <- c(.2, .33, .4, .1)  
chisq.test(observed, p = expected, rescale.p = TRUE)

##   
## Chi-squared test for given probabilities  
##   
## data: observed  
## X-squared = 57.942, df = 3, p-value = 1.617e-12