Problem Set #2 - DATASCI W241

Greg Ceccarelli

Sept 30, 2015

1. FE, exercise 3.6

```
setwd("/Users/ceccarelli/MIDS/DATASCI_W241/Async Material and Sample Files/Chapter 3")
#clear variables
rm( list = ls() )
#read in tabular data
data <- read.csv("Clingingsmith_et_al_QJE_2009dta.csv", sep=",", header = TRUE)
#create shorthand reference
d<-data
##replicate the table on 3.2 in the book
prop.table(table(d$views,d$success),2)*100
```

```
##
##                0          1
##   -12  0.0000000  0.1960784
##   -9   0.2232143  0.0000000
##   -8   0.0000000  0.1960784
##   -6   0.4464286  0.1960784
##   -5   0.0000000  0.1960784
##   -4   0.4464286  0.5882353
##   -3   0.0000000  0.1960784
##   -2   1.1160714  0.9803922
##   -1   1.5625000  2.7450980
##   0   27.2321429 18.6274510
##   1   18.3035714 13.1372549
##   2   24.3303571 25.2941176
##   3    8.4821429 10.9803922
##   4    5.8035714  9.6078431
##   5    3.3482143  3.9215686
##   6    3.7946429  7.2549020
##   7    2.2321429  2.5490196
##   8    0.8928571  1.3725490
##   9    0.2232143  0.7843137
##   10   0.4464286  0.0000000
##   11   0.6696429  0.1960784
##   12   0.4464286  0.9803922
```

```
#Create two vectors to sample from based on the data
Y <- d$views #outcome
Z <- d$success #treatment

# number of RI iterations
numiter <- 10000
set.seed(1234567) # set random number seed (so that results can be replicated)
```

```
#compute ATE prior to randomization
denom <- var(Z)
ate <- cov(Y,Z)/denom

#iterate
tauRI <- rep(NA,numiter)

for (i in 1:numiter) {

Zri <- sample(Z)
tauRI[i] <- cov(Y,Zri)/denom

if (i %% round(numiter/10) == 0) cat("Iteration",i,"of",numiter,"\n")
}
```

```
## Iteration 1000 of 10000
## Iteration 2000 of 10000
## Iteration 3000 of 10000
## Iteration 4000 of 10000
## Iteration 5000 of 10000
## Iteration 6000 of 10000
## Iteration 7000 of 10000
## Iteration 8000 of 10000
## Iteration 9000 of 10000
## Iteration 10000 of 10000
```

How many of the simulated random assignments generate an ATE that is at least as large as the actual estimate of the ATE?

```
nrow(subset(as.data.frame(tauRI),tauRI>=ate))
```

```
## [1] 26
```

What is the implied one-tailed p-value?

```
pvalue <- round(nrow(subset(as.data.frame(tauRI),tauRI>=ate))/numiter,3)
pvalue
```

```
## [1] 0.003
```

How many of the simulated random assignments generate an ATE that is at least as large in absolute value as the actual estimate of the ATE?

```
#wrap tauRI in abs function
nrow(subset(as.data.frame(tauRI),abs(tauRI)>=ate))
```

```
## [1] 42
```

What is the implied one-tailed p-value?

```
#wrap tauRI in abs function
pvalue.twotailed <- round(nrow(subset(as.data.frame(tauRI),abs(tauRI)>=ate))/numiter,3)
pvalue.twotailed
```

```
## [1] 0.004
```

2. FE, exercise 3.8

```
setwd("/Users/ceccarelli/MIDS/DATASCI_W241/Async Material and Sample Files/Chapter 3")
#clear variables
rm( list = ls() )
require("foreign")
```

```
## Loading required package: foreign
```

```
#read in tabular data
data <- read.dta("Titiunik_WorkingPaper_2010.csv.dta")
#create shorthand reference
t<-data
```

a. For each state, estimate the effect of having a two-year term on the number of bills introduced?

```
#Create two vectors for Texas
t.Y <- unlist(subset(t,t$texas0_arkansas1==0)[2]) #outcome
t.Z <- unlist(subset(t,t$texas0_arkansas1==0)[1])#treatment

#compute ATE for Texas
t.denom <- var(t.Z)
t.ate <- cov(t.Y,t.Z)/t.denom
cat("Texas ATE = ", t.ate)
```

```
## Texas ATE =  -16.74167
```

```
##done another way
#mean(unlist(subset(t,t$texas0_arkansas1==0 & t$term2year==0)[2]))-mean(unlist(subset(t,t$texas0_arkans

#Create two vectors for Texas
a.Y <- unlist(subset(t,t$texas0_arkansas1==1)[2]) #outcome
a.Z <- unlist(subset(t,t$texas0_arkansas1==1)[1])#treatment

#compute ATE for Arkansas
a.denom <- var(a.Z)
a.ate <- cov(a.Y,a.Z)/a.denom
cat("Arkansas ATE = ", a.ate)
```

```
## Arkansas ATE =  -10.09477
```

b. For each state, estimate the standard error of the ATE?

```
##texas
t.obs.0 <- unlist(subset(t,t$texas0_arkansas1==0 & t$term2year==0)[2])
t.obs.1 <- unlist(subset(t,t$texas0_arkansas1==0 & t$term2year==1)[2])

##calculate t.Var.hat.1 manually
sum((t.obs.1 - ((sum(t.obs.1)/length(t.obs.1))))^2)/(length(t.obs.1)-1)
```

```
## [1] 413.6952
```

```
##matches
var(t.obs.1)
```

```
## [1] 413.6952
```

```
t.se.hat <- sqrt((var(t.obs.0)/length(t.obs.0)) + (var(t.obs.1)/length(t.obs.1)))
cat("Texas STE of ATE = ", t.se.hat)
```

```
## Texas STE of ATE =  9.345871
```

```
##arkansas
a.obs.0 <- unlist(subset(t,t$texas0_arkansas1==1 & t$term2year==0)[2])
a.obs.1 <- unlist(subset(t,t$texas0_arkansas1==1 & t$term2year==1)[2])

a.se.hat <- sqrt((var(a.obs.0)/length(a.obs.0)) + (var(a.obs.1)/length(a.obs.1)))
cat("Arkansas STE of ATE = ", a.se.hat)
```

```
## Arkansas STE of ATE =  3.395979
```

c. Estimate overall of ATE for both states combined

```
##
t.pe.count<-length(t.obs.0)+length(t.obs.1)
a.pe.count<-length(a.obs.0)+length(a.obs.1)

pooled.ate <- a.ate*(a.pe.count/sum(t.pe.count,a.pe.count))+t.ate*(t.pe.count/sum(t.pe.count,a.pe.count
cat("Pooled ATE = ", pooled.ate)
```

```
## Pooled ATE =  -13.2168
```

d. Explain why pooling the data leads to biased estimates of ATE? Quite simply, there are not the same number of observations in the treatment and control in each state

e. Insert the estimated standard errors into equation 3.12 to estiamte standard error for overall ATE

```
pooled.se <- sqrt(a.se.hat^2*(a.pe.count/sum(t.pe.count,a.pe.count))^2+a.se.hat^2*(t.pe.count/sum(t.pe.
cat("Pooled SE for Overall ATE = ", pooled.se)
```

```
## Pooled SE for Overall ATE =  2.405726
```

not quite; error in the weighting

f. use randomization inference to test sharp null

```r
numiter <- 10000 # number of RI iterations. Use more for greater precision, fewer for greater speed.
set.seed(1234567) # set random number seed (so that results can be replicated)

##texas
t.tauRI <- rep(NA,numiter)

for (i in 1:numiter) {

t.Zri <- sample(t.Z)
t.tauRI[i] <- cov(t.Y,t.Zri)/t.denom

if (i %% round(numiter/10) == 0) cat("Iteration",i,"of",numiter,"\n")
}
```

```
## Iteration 1000 of 10000
## Iteration 2000 of 10000
## Iteration 3000 of 10000
## Iteration 4000 of 10000
## Iteration 5000 of 10000
## Iteration 6000 of 10000
## Iteration 7000 of 10000
## Iteration 8000 of 10000
## Iteration 9000 of 10000
## Iteration 10000 of 10000
```

```r
t.pvalue <- round(sum(t.tauRI >= t.ate)/numiter,3)
t.pvalue
```

```
## [1] 0.957
```

```r
##arkansas
a.tauRI <- rep(NA,numiter)

for (i in 1:numiter) {

a.Zri <- sample(a.Z)
a.tauRI[i] <- cov(a.Y,a.Zri)/a.denom

if (i %% round(numiter/10) == 0) cat("Iteration",i,"of",numiter,"\n")
}
```

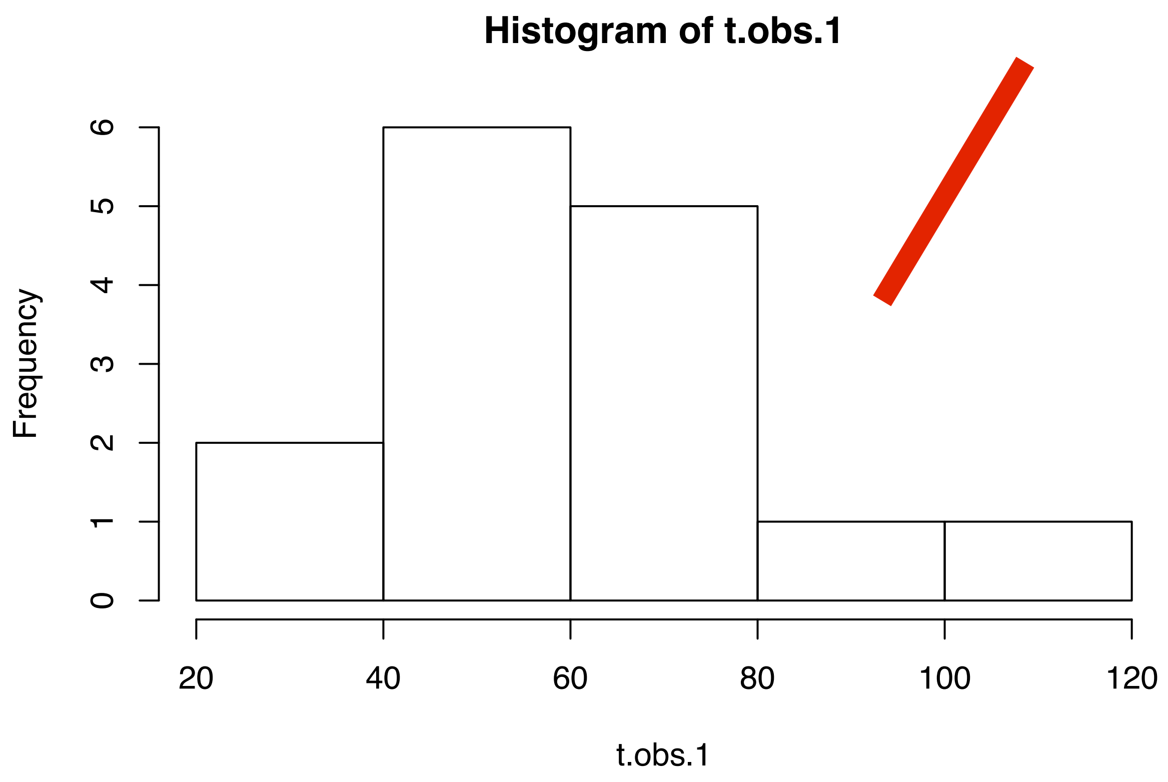
```
## Iteration 1000 of 10000
## Iteration 2000 of 10000
## Iteration 3000 of 10000
## Iteration 4000 of 10000
## Iteration 5000 of 10000
## Iteration 6000 of 10000
## Iteration 7000 of 10000
## Iteration 8000 of 10000
## Iteration 9000 of 10000
## Iteration 10000 of 10000
```

```
a.pvalue <- round(sum(a.tauRI >= a.ate)/numiter,3)
a.pvalue
```

```
## [1] 0.998
```

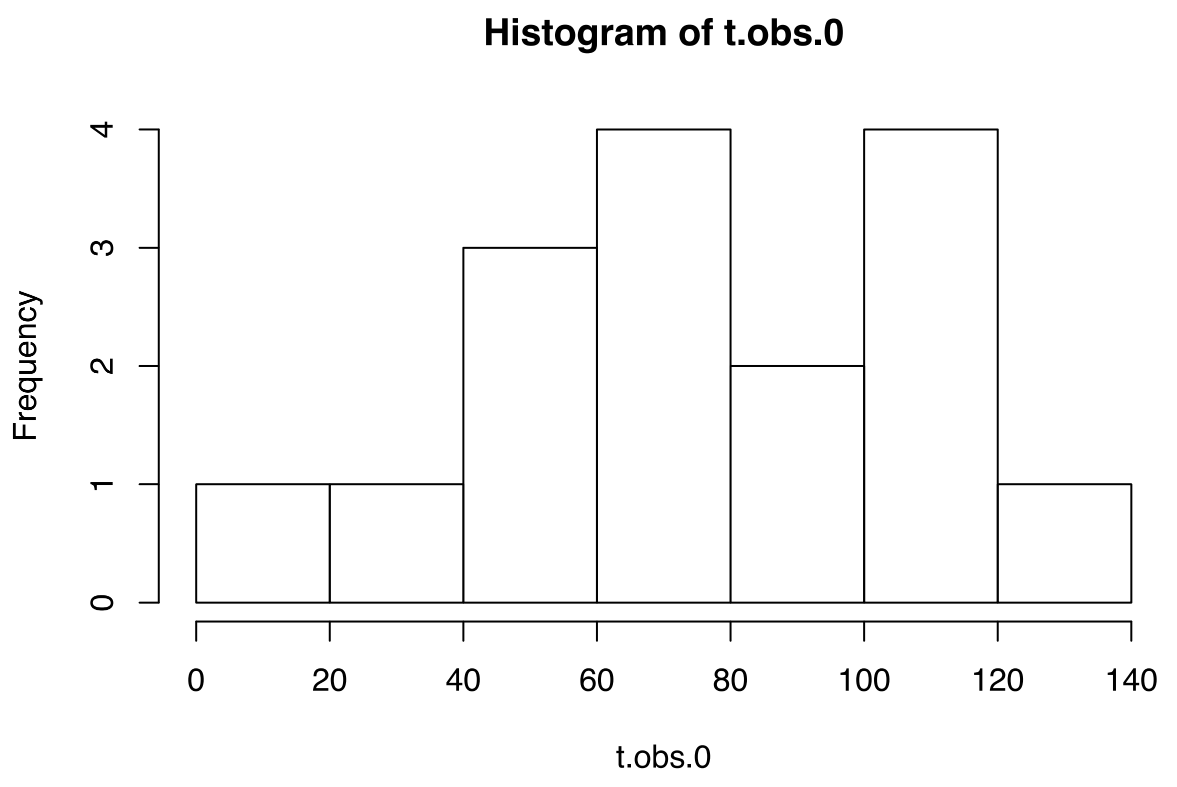
g. plotting histograms per instruction

```
#texas treatment & control
hist(t.obs.1)
```

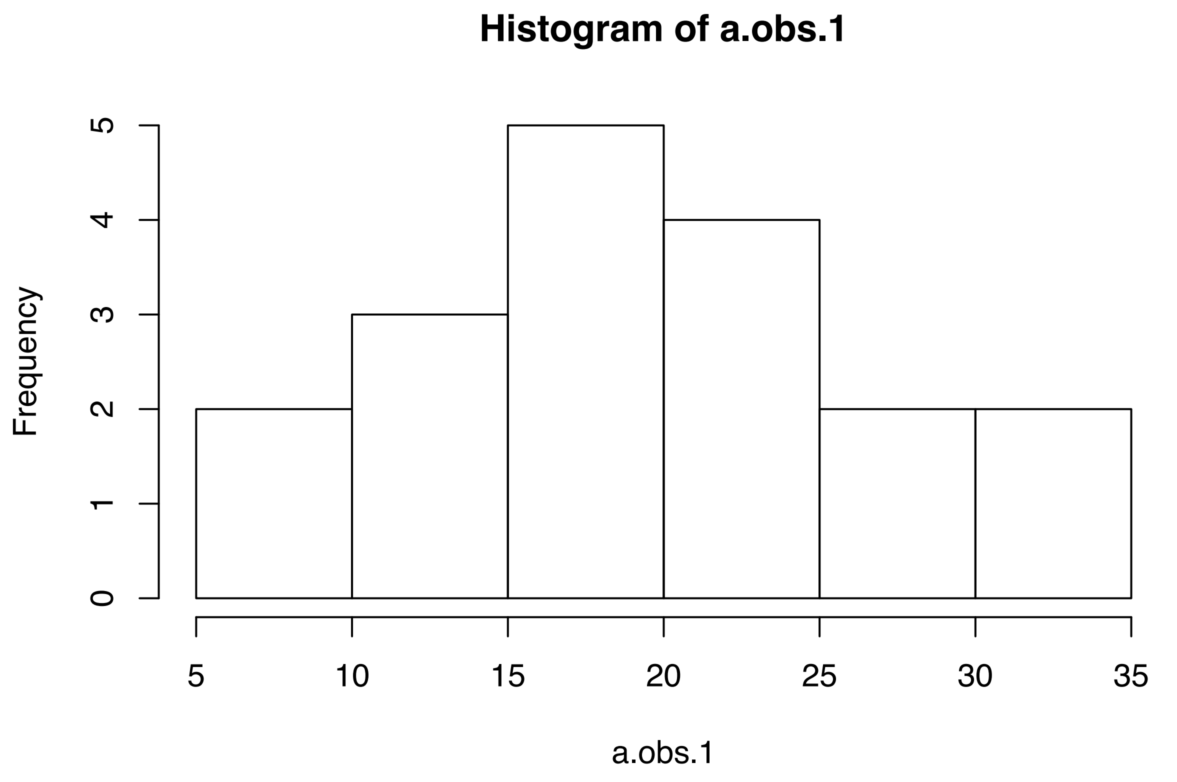


```
hist(t.obs.0)
```

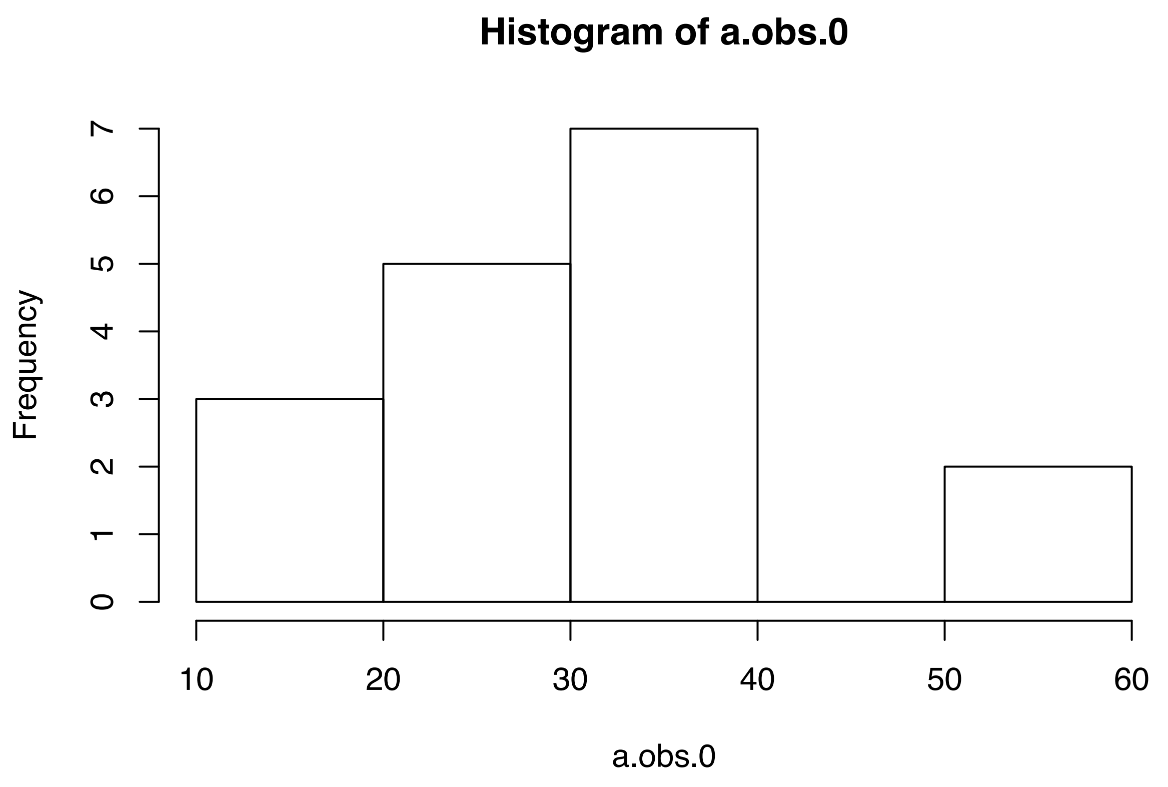
consider locating all hists on a common x-range



```
#arkansas treatment & control
hist(a.obs.1)
```



```r
hist(a.obs.0)
```



3. FE, exercise 3.11

a.

```r
setwd("/Users/ceccarelli/MIDS/DATASCI_W241/Async Material and Sample Files/Chapter 3")
#clear variables
rm( list = ls() )
#read in tabular data
data <- read.csv("GerberGreenBook_Chapter3_Table_3_3.csv",header = TRUE)
#create shorthand reference
c<-data
##View(c)
c$Y
```

```
##  [1]  0  1  2  4  4  6  6  9 14 15 16 16 17 18
```

```r
c$D
```

```
##  [1]  0  0  1  2  0  0  2  3 12  9  8 15  5 17
```

```r
cluster.ids<-c(1,1,2,2,3,3,4,4,5,5,6,6,7,7)
cluster.ids
```

```
##  [1] 1 1 2 2 3 3 4 4 5 5 6 6 7 7
```

```r
all.clusters <- unique(cluster.ids)

randomize.clustered <- function(){
  treat.cluster.ids <- sample(all.clusters, length(all.clusters)*(3/7))
  return(
    as.numeric(cluster.ids %in% treat.cluster.ids)
  )
}
randomize.clustered()
```

```
##  [1] 1 1 0 0 0 0 1 1 1 1 0 0 0 0
```

```r
cl1<-as.data.frame(cbind(c$Y,c$D,randomize.clustered()))
names(cl1)[1] <- "Y"
names(cl1)[2] <- "D"
names(cl1)[3] <- "Cluster"

cl1.0 <- unlist(subset(cl1,cl1$Cluster==0)[1])
cl1.1 <- unlist(subset(cl1,cl1$Cluster==0)[2])

se.ate <- sqrt((1/(7-1))*((4*var(cl1.0)/(7-4))+(3*var(cl1.1)/(4))+(2*cov(cl1.0,cl1.1))))
cat("SE for Cluster Assingment 1 = ", se.ate)
```

```
## SE for Cluster Assingment 1 =  4.487085
```

a.

```r
setwd("/Users/ceccarelli/MIDS/DATASCI_W241/Async Material and Sample Files/Chapter 3")
#clear variables
rm( list = ls() )
#read in tabular data
data <- read.csv("GerberGreenBook_Chapter3_Table_3_3.csv",header = TRUE)
#create shorthand reference
c<-data
##View(c)
c$Y
```

```
##  [1]  0  1  2  4  4  6  6  9 14 15 16 16 17 18
```

```r
c$D
```

```
##  [1]  0  0  1  2  0  0  2  3 12  9  8 15  5 17
```

```r
cluster.ids<-c(1,2,3,4,5,6,7,7,6,5,4,3,2,1)
cluster.ids
```

```
##  [1] 1 2 3 4 5 6 7 7 6 5 4 3 2 1
```

```r
all.clusters <- unique(cluster.ids)

randomize.clustered <- function(){
  treat.cluster.ids <- sample(all.clusters, length(all.clusters)*(3/7))
  return(
    as.numeric(cluster.ids %in% treat.cluster.ids)
  )
}
randomize.clustered()
```

```
##  [1] 1 0 1 0 0 1 0 0 1 0 0 1 0 1
```

```r
cl1<-as.data.frame(cbind(c$Y,c$D,randomize.clustered()))
names(cl1)[1] <- "Y"
names(cl1)[2] <- "D"
names(cl1)[3] <- "Cluster"

cl1.0 <- unlist(subset(cl1,cl1$Cluster==0)[1])
cl1.1 <- unlist(subset(cl1,cl1$Cluster==0)[2])

se.ate <- sqrt((1/(7-1))*((4*var(cl1.0)/(7-4))+(3*var(cl1.1)/(4))+(2*cov(cl1.0,cl1.1))))
cat("SE for Cluster Assingment 2 = ", se.ate)
```

hmm… this isn't hitting it.

```
## SE for Cluster Assingment 2 =  4.699053
```

c. They lead to different standard errors because there is variability in cluster level means across clusters. For large expeirments its crucial to look for ways to INCREASE the number of clusters to reduce this penalty.

4. Question 4

```r
#problem set up
cws <-  1
advertising <- .10
users <- 1:1000000
profit <- 100
conversion <- .005
```

by and large, i'm not following the steps you've made in this question.

a. By how much does the ad campaign need to increase the probability of purchase in order to be "worth it" and a positive ROI?

```r
options("scipen"=100, "digits"=10)
average.sales<- length(users)*conversion*profit
average.sales
```

```
## [1] 500000
```

```r
cost<-length(users)*.10
required.conversion <- (average.sales+cost+1)/(length(users)*profit)
required.increase <- (required.conversion - conversion)/conversion*100
required.increase
```

```
## [1] 20.0002
```

b.

```r
measuredeffect<-.002
size<-length(users)/2
treatment <- (length(users)/2)*(measuredeffect+conversion)
control <- (length(users)/2)*(conversion)
#prop.test(x=c(treatment,control),n=c(size,size))
p<-(treatment+control)/(size+size)
p
```

```
## [1] 0.006
```

```r
se <- sqrt(p*(1-p)*((1/size)+(1/size)))*size
se
```

```
## [1] 77.22693831
```

```r
#confidence interval
conf<- c(treatment-se*1.96, treatment+se*1.96)
conf
```

```
## [1] 3348.635201 3651.364799
```

c. Is this confidence interval precise enough that you would recommend running this experiment? Why or why not? Yes, not only is it statistically significant (per a prop.test) but the interval does not dip behind the average expectation

d. What would be the width of the confidence interval for this experiment if only 1% of users were placed in the control group?

```r
measuredeffect<-.002
size.treatment<-length(users)*(.99)
size.control<-length(users)*(.01)
treatment <- (size.treatment)*(measuredeffect+conversion)
control <- (size.control)*(conversion)
#prop.test(x=c(treatment,control),n=c(size.treatment,size.control))
p<-(treatment+control)/(size.treatment+size.control)
p
```

```
## [1] 0.00698
```

```
se <- sqrt(p*(1-p)*((1/size.treatment)+(1/size.control)))*size.treatment
se
```

```
## [1] 828.3698935
```

```
#confidence interval
conf<- c(treatment-se*1.96, treatment+se*1.96)
conf
```

```
## [1] 5306.395009 8553.604991
```

5. Question 5

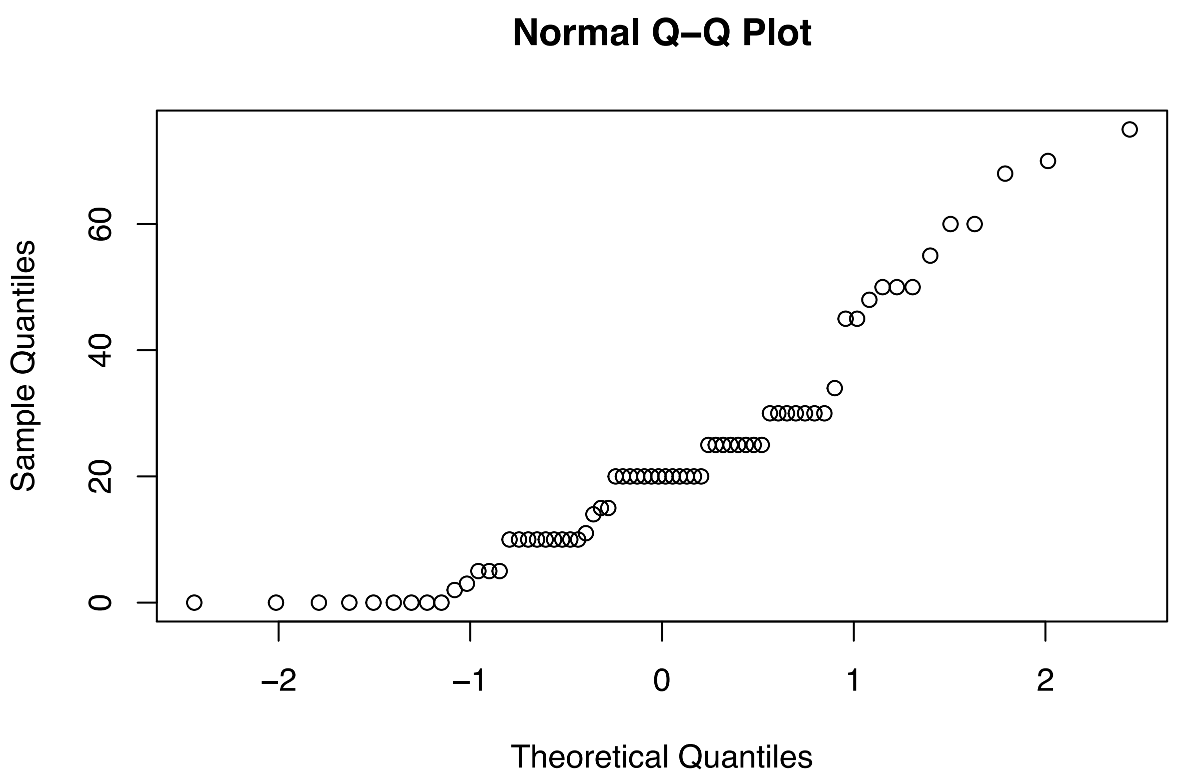
```
setwd("/Users/ceccarelli/MIDS/DATASCI_W241/Async Material and Sample Files/Chapter 3")
#clear variables
rm( list = ls() )
#read in tabular data
data <- read.csv("PS 2 data - list_luckingreiley_auction_data.csv",header = TRUE)
#create shorthand reference
r<-data

# The means look different between groups
by(r$bid, r$uniform_price_auction, mean, na.rm = TRUE)
```

```
## r$uniform_price_auction: 0
## [1] 28.82352941
## ---------------------------------------------------------
## r$uniform_price_auction: 1
## [1] 16.61764706
```

a. Compute a 95% confidence interval for the difference between the treatment mean and the control mean, using analytic formulas for a two-sample t-test from your earlier statistics course.

```
#Check if means statistically different
# From the qqplot, it's not clear if loggdp is normally distributed, doesn't look like it
qqnorm(r$bid)
```



```
# The Shapiro test suggests that it's not
shapiro.test(r$bid)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  r$bid
## W = 0.90139907, p-value = 0.00005581924
```

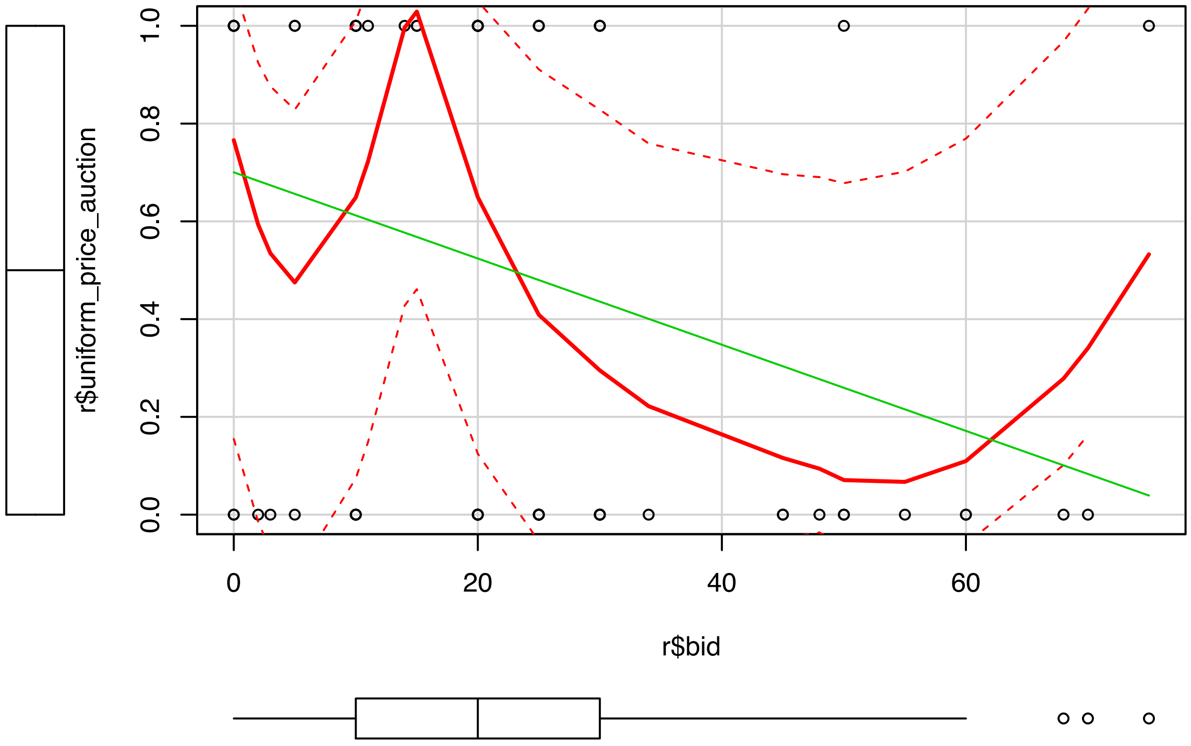
```
# But we have a large sample size, so we can rely on
# the central limit theorem and use a regular t.test
t.test(r$bid ~ r$uniform_price_auction, r, alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  r$bid by r$uniform_price_auction
## t = 2.8211439, df = 61.982867, p-value = 0.006420778
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   3.557140637 20.854624069
## sample estimates:
## mean in group 0 mean in group 1
##     28.82352941     16.61764706
```

b. In plain language, what does this confidence interval mean? The confidence interval (3.557140637 20.854624069) is the range of values that is likely to contain the population mean in this case.

c. Calculate the 95% confidence interval you get from the regression.

```
library(car)
#look at a quick scatter
scatterplot(r$bid,r$uniform_price_auction)
```



```
# Build regression model
model = lm(bid~uniform_price_auction, data = r)
summary(model)
```

```
##
## Call:
## lm(formula = bid ~ uniform_price_auction, data = r)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -28.823529 -11.617647  -3.220588   8.382353  58.382353
##
## Coefficients:
##                        Estimate Std. Error  t value             Pr(>|t|)
## (Intercept)           28.823529   3.059348  9.42146 0.000000000000078066
## uniform_price_auction -12.205882   4.326572 -2.82114            0.0063148
##
## (Intercept)           ***
## uniform_price_auction **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.83891 on 66 degrees of freedom
## Multiple R-squared:  0.1076119,  Adjusted R-squared:  0.09409087
## F-statistic: 7.958853 on 1 and 66 DF,  p-value: 0.006314796
```

```
estimate <- as.vector(model$coefficients[1])
standard.error <- abs(as.vector(model$coefficients[2]))

lower.bound <- estimate - standard.error * 1.96
upper.bound <- estimate + standard.error * 1.96

cat("Conf Interval = ", lower.bound, upper.bound)
```

```
## Conf Interval =  4.9 52.74705882
```

d. On to p-values. What p-value does the regression report?

```
cat("p-value = ", summary(model)$coefficients[,4])
```

```
## p-value =  0.0000000000007806564729 0.006314795951
```

e. Now compute the same p-value using randomization inference.

```
numiter <- 10000 # number of RI iterations. Use more for greater precision, fewer for greater speed.
set.seed(12) # set random number seed (so that results can be replicated)

# Compute RI Distribution

Y <- r$bid
Z <- r$uniform_price_auction

denom <- var(Z)
tau <- cov(Y,Z)/denom

tauRI <- rep(NA,numiter)

for (i in 1:numiter) {

Zri <- sample(Z)
tauRI[i] <- cov(Y,Zri)/denom

if (i %% round(numiter/10) == 0) cat("Iteration",i,"of",numiter,"\n")
}
```

```
## Iteration 1000 of 10000
## Iteration 2000 of 10000
## Iteration 3000 of 10000
## Iteration 4000 of 10000
## Iteration 5000 of 10000
## Iteration 6000 of 10000
## Iteration 7000 of 10000
## Iteration 8000 of 10000
## Iteration 9000 of 10000
## Iteration 10000 of 10000
```

```
pvalue <- round(sum(tauRI >= abs(tau))/numiter,3)
pvalue
```

```
## [1] 0.003
```

this is exactly 1/2 what it should be; this isn't a small sample problem; its a code problem

f. Compute the same p-value again using analytic formulas for a two-sample t-test from your earlier statistics course. (Also see part (a).)

```
# compare with traditional t-tests
  t.test(Y~Z,var.equal=FALSE,alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  Y by Z
## t = 2.8211439, df = 61.982867, p-value = 0.006420778
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   3.557140637 20.854624069
## sample estimates:
## mean in group 0 mean in group 1
##     28.82352941     16.61764706
```

g. Compare the two p-values in parts (e) and (f). Are they much different? Why or why not? How might your answer to this question change if the sample size were different?

The t-tests aren't very much different in real terms. They would converge as the sample size increased.