Homework 2

Joshua Oswari - A14751270

4/20/2019

Problem 1

```
#install.packages("MASS")
require(MASS)
```

```
## Loading required package: MASS
```

```
options(warn=-1)
# to run back warning use: options(warn=0)
```

```
chisq.normal.test = function(x, B = 2000){
  n=length(x)
  #calculating MLE
  fit = fitdistr(x, "normal")
  mu = fit$est[1]
  sigma = fit$est[2]

  # histogram for automatically selecting the bins
  H = hist(x, 30, col="grey", freq=FALSE)

  # expected counts for these particular bins, and original statistic
  P = pnorm(H$breaks, mean=mu, sd=sigma)
  exp.counts = n*diff(P)
  obs.counts = table(cut(x, H$breaks))
  test = chisq.test(obs.counts, p=exp.counts, rescale.p=TRUE)
  D = as.numeric(test$stat)

  D.boot = numeric(B)
  count = 0
  for (b in 1:B) {
  dat.boot = rnorm(n, mean=mu, sd=sigma)
  mu.boot = mean(dat.boot)
    sigma.boot = sd(dat.boot)
    P = pnorm(H$breaks, mean=mu.boot, sd=sigma.boot)
    exp.counts = n*diff(P)
    obs.counts = table(cut(dat.boot, H$breaks))
    test = chisq.test(obs.counts, p=exp.counts, rescale.p=TRUE)
    D.boot[b] = test$stat
    }
  return((count+1)/(B+1))
}
```

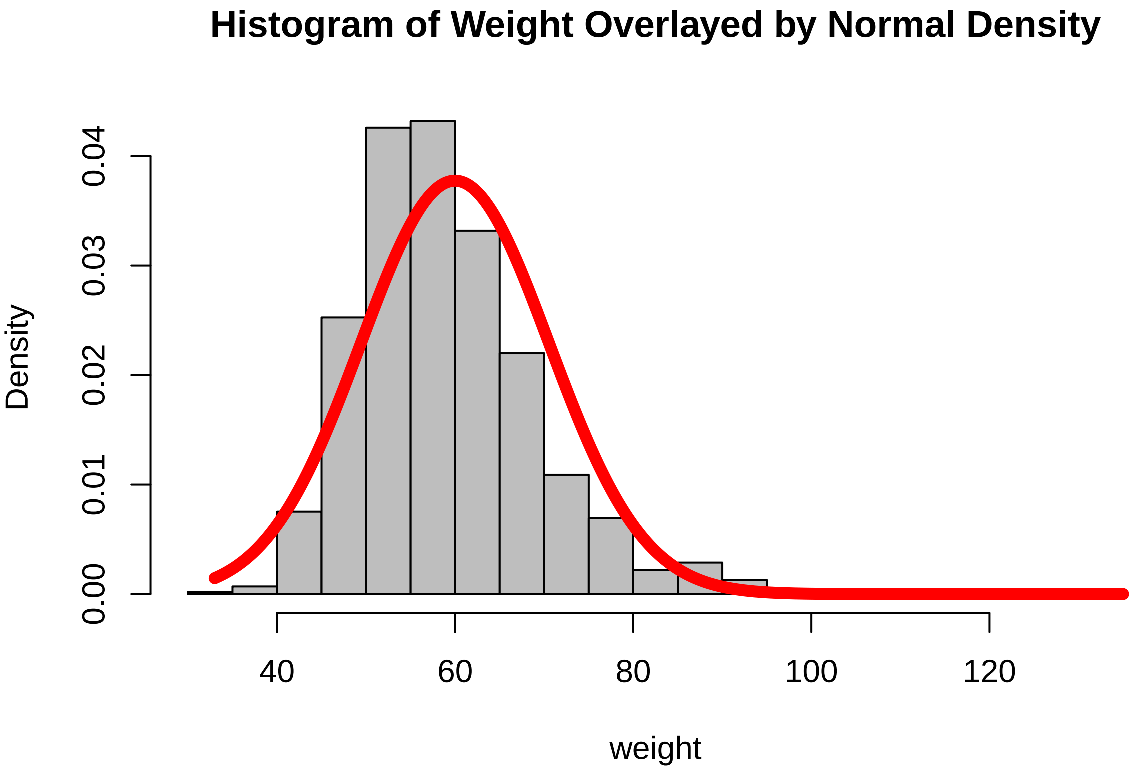
Problem 2 Note: This example is done for the polulation together; the code for each individual country is identical with the only change being w

```
#install.packages("readxl")
library(readxl)
```

```
xdf = read_excel("~/Documents/Math185/S1Dataset.xlsx", sheet=2)
w = xdf$Weight
w = as.numeric(na.omit(w))
```

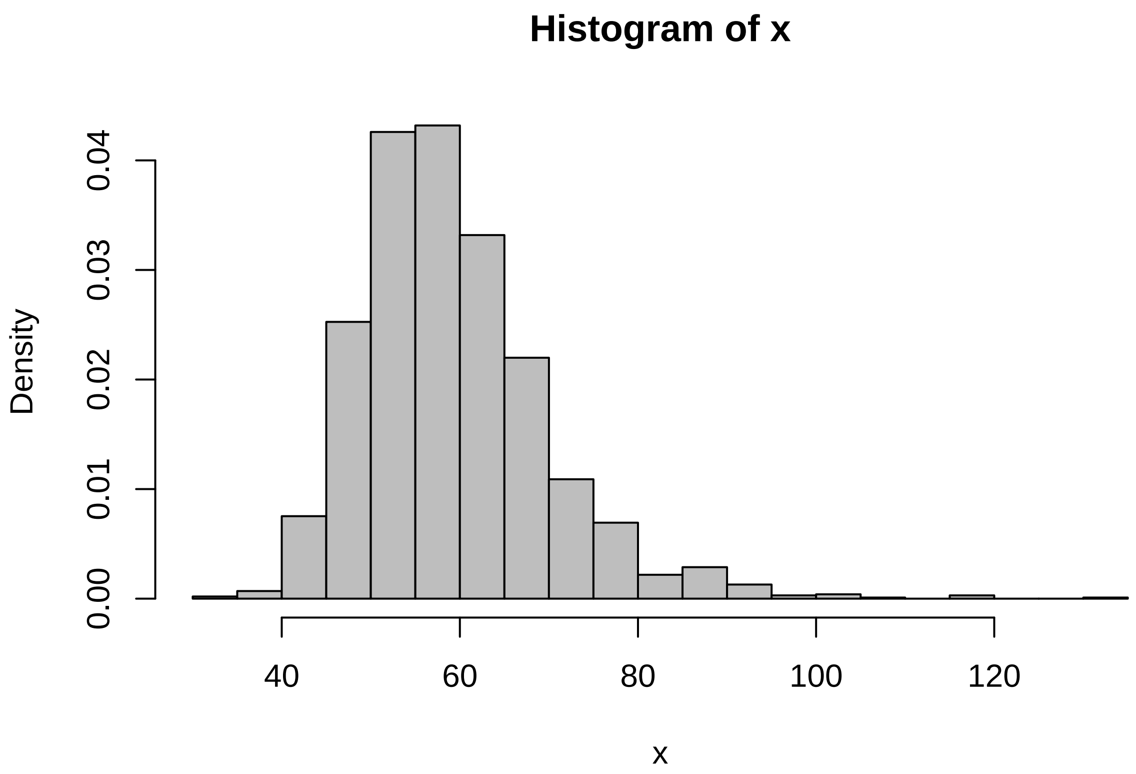
Histogram and overlay the normal distribution

```
dat = w
fit = fitdistr(dat, "normal")
mu = fit$est[1]
sigma = fit$est[2]

# comparing histogram with fitted density
H = hist(dat, 30, col="grey", freq=FALSE, main="Histogram of Weight Overlayed by Normal Density", xlab=
# 33 and 135 comes from the summary of w; you could do this for each individual country
t = seq(33, 135, len = 1000)
lines(t, dnorm(t, mean=mu, sd=sigma), col=2, lwd = 6)
```



Test for normality of the all country(together)'s weight using function in Problem 1

```
chisq.normal.test(w, B = 2000)
```



```
## [1] 0.0004997501
```

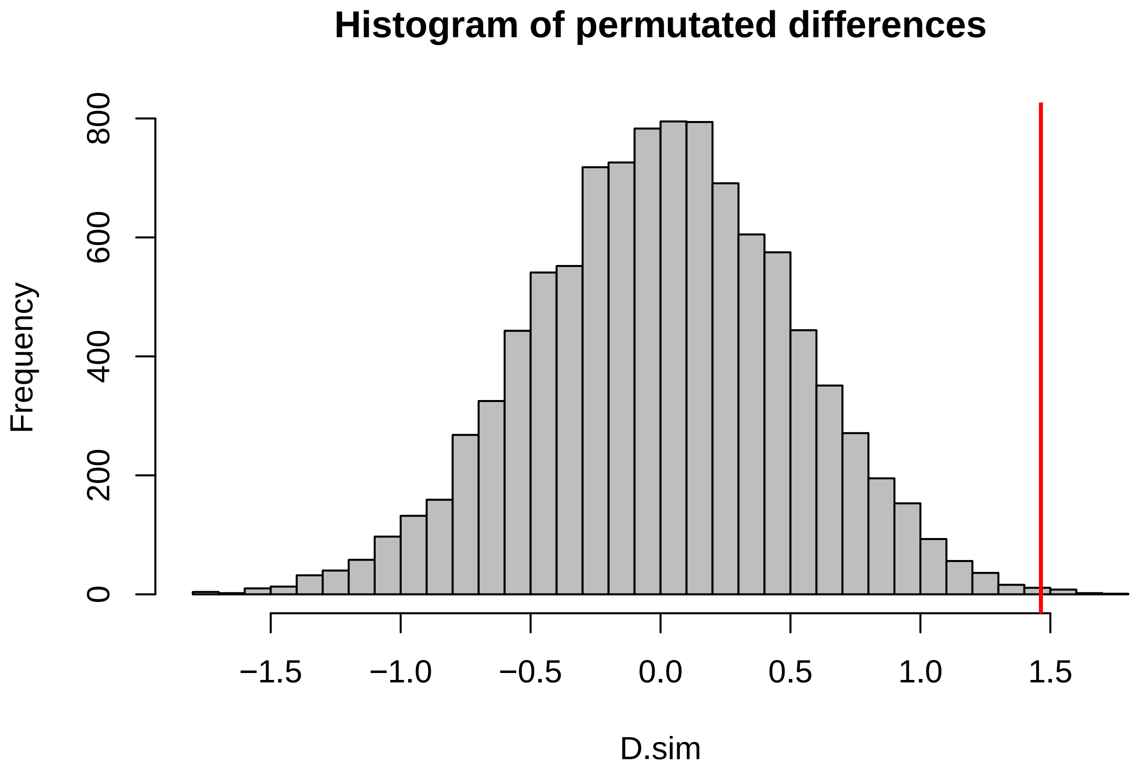
Problem 3

```
brazil = xdf[which(xdf$Country==1), ]
portugal =  xdf[which(xdf$Country==2), ]
mozambique = xdf[which(xdf$Country==3), ]

brazilweight = brazil$Weight
brazilweight=as.numeric(na.omit(brazilweight))

portugalweight = portugal$Weight
portugalweight=as.numeric(na.omit(portugalweight))

mozambiqueweight = mozambique$Weight
mozambiqueweight=as.numeric(na.omit(mozambiqueweight))
```

One possible Way is to write a function pertest that does permutation test of the following hypothesis: H0: X ~ Y H1: X stochastically dominates Y

Once the function is written, you could use the following two lines

```
pertest = function(X, Y, B) {
m = length(X)
n = length(Y)
D = mean(X) - mean(Y)
Z = c(X, Y)
D.sim = numeric(B)
for (b in 1:B) {
    Zperm = sample(Z)
    D.sim[b] = mean(Zperm[1:m]) - mean(Zperm[(m+1):(m+n)])
    }
p.val = (sum(D.sim >= D)+1)/(B+1)
hist(D.sim, breaks=50, main='Histogram of permutated differences', col='grey')
abline(v=D, col=2, lwd=2)
}

pertest(brazilweight, portugalweight, 10000)
```



Wilcoxen Rank Sum Test/Mann Whitney U Test

```
wilcox.test(brazilweight, portugalweight, alternative="greater")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  brazilweight and portugalweight
## W = 401190, p-value = 0.009522
## alternative hypothesis: true location shift is greater than 0
```