HW2.R

riserate

Wed Jul 5 17:41:44 2017

dat <- read.csv("femaleMiceWeights.csv")  
dat[1:12,2]

## [1] 21.51 28.14 24.04 23.45 23.68 19.79 28.40 20.98 22.51 20.10 26.91  
## [12] 26.25

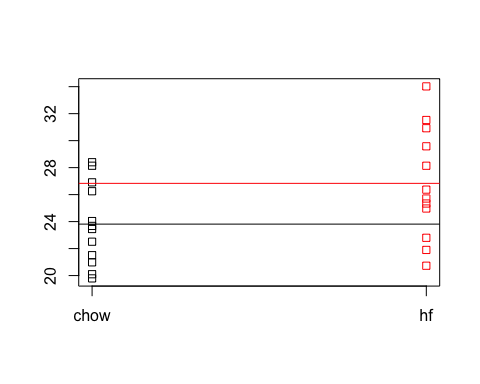
dat[13:24,2]

## [1] 25.71 26.37 22.80 25.34 24.97 28.14 29.58 30.92 34.02 21.90 31.53  
## [12] 20.73

mean(dat[13:24,2]) - mean(dat[1:12,2])

## [1] 3.020833

s = split(dat[,2], dat[,1])  
stripchart(s, vertical=TRUE, col=1:2)  
  
abline(h=sapply(s, mean), col=1:2)



#new vector for data set  
  
highfat = s[["hf"]]  
chow = s[["chow"]]  
  
sample(highfat, 6, replace=TRUE) # replaces values with TRUE

## [1] 25.71 25.34 20.73 24.97 20.73 20.73

as.numeric(highfat > 30) #converts TRUE to 0 or 1

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

sum(highfat > 30)

## [1] 3

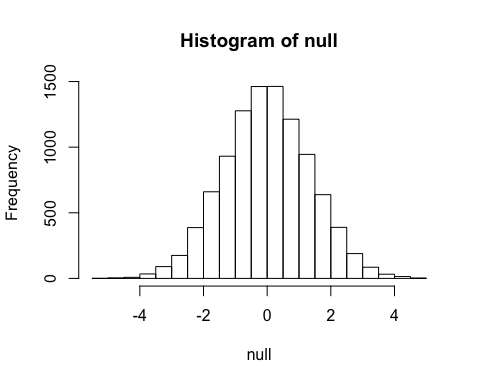
mean(as.numeric(highfat > 30))

## [1] 0.25

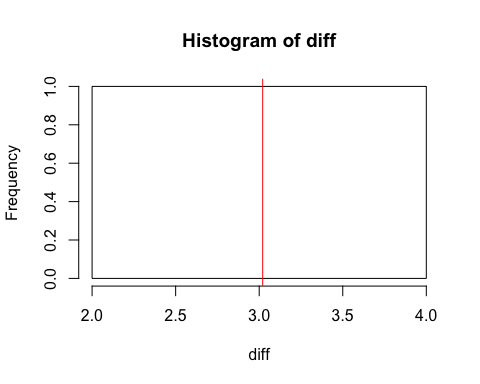
population <- read.csv("femaleControlsPopulation.csv") #import data  
  
control <- sample(population[,1],12)  
mean(control)

## [1] 25.24417

n <- 10000  
null <- vector("numeric",n)  
for(i in 1:n){  
 control <- sample(population[,1],12)  
 treatment <- sample(population[,1],12)  
 null[i] <- mean(control) - mean(treatment)  
}  
  
hist(null)



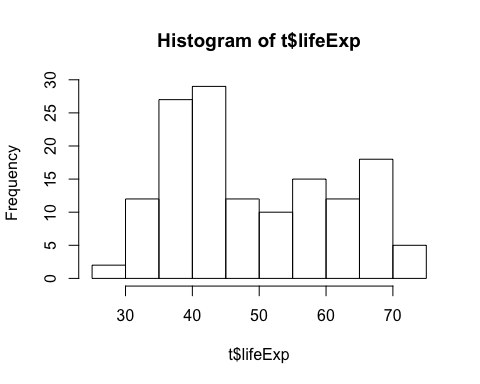
#sampleMean = replicate(10000, mean(sample(population, 12)))  
#mean(sample(population, 12))  
#head(sampleMean)  
  
  
  
diff = mean(dat[13:24,2]) - mean(dat[1:12,2])  
hist(diff)  
abline(v=diff, col="red")  
abline(v=-diff, col="red")



#HW2 Distro II  
  
library(devtools)  
library(devtools)  
library(gapminder)  
data(gapminder)  
head(gapminder)

## # A tibble: 6 x 6  
## country continent year lifeExp pop gdpPercap  
## <fctr> <fctr> <int> <dbl> <int> <dbl>  
## 1 Afghanistan Asia 1952 28.801 8425333 779.4453  
## 2 Afghanistan Asia 1957 30.332 9240934 820.8530  
## 3 Afghanistan Asia 1962 31.997 10267083 853.1007  
## 4 Afghanistan Asia 1967 34.020 11537966 836.1971  
## 5 Afghanistan Asia 1972 36.088 13079460 739.9811  
## 6 Afghanistan Asia 1977 38.438 14880372 786.1134

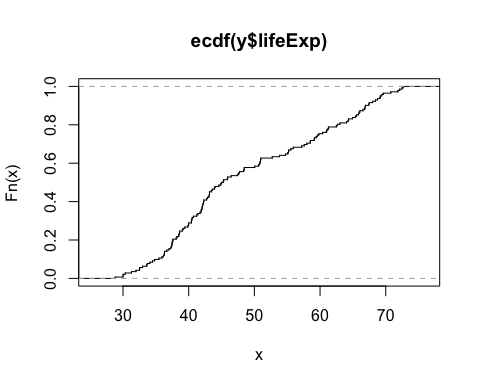
#vector of Country and lifeExp 1952  
t = gapminder[gapminder$year <= 1952,c(1,4)] #Both country and lifeExp  
x = gapminder[gapminder$year <= 1952,c(1)] #country  
y = gapminder[gapminder$year <= 1952,c(4)] #lifeExp  
hist(t$lifeExp)



stem(t$lifeExp)

##   
## The decimal point is at the |  
##   
## 28 | 8  
## 30 | 0033  
## 32 | 055067  
## 34 | 15859  
## 36 | 2333703445566  
## 38 | 1255660149  
## 40 | 004455724799  
## 42 | 0011233791122469  
## 44 | 06903399  
## 46 | 5568  
## 48 | 455  
## 50 | 1688990  
## 52 | 78  
## 54 | 712269  
## 56 | 26  
## 58 | 05512468  
## 60 | 40123  
## 62 | 560  
## 64 | 03494699  
## 66 | 1689945  
## 68 | 0481246  
## 70 | 89  
## 72 | 157

plot(ecdf(y$lifeExp), do.points=FALSE, verticals=TRUE)



mean(y <=40) #What is the proportion of countries in 1952 that have a life expectancy less than or equal to 40?

## [1] 0.2887324

mean(y <=60) - mean(y <=40)

## [1] 0.4647887

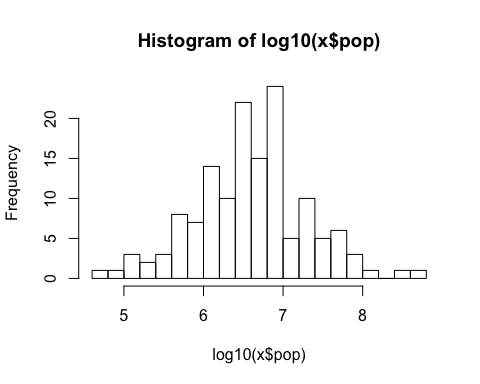
mean(y >= 40)

## [1] 0.7183099

#vector of Country and Population in 1952  
t = gapminder[gapminder$year <= 1952,c(1,5)] #Both country and Population  
head(t)

## # A tibble: 6 x 2  
## country pop  
## <fctr> <int>  
## 1 Afghanistan 8425333  
## 2 Albania 1282697  
## 3 Algeria 9279525  
## 4 Angola 4232095  
## 5 Argentina 17876956  
## 6 Australia 8691212

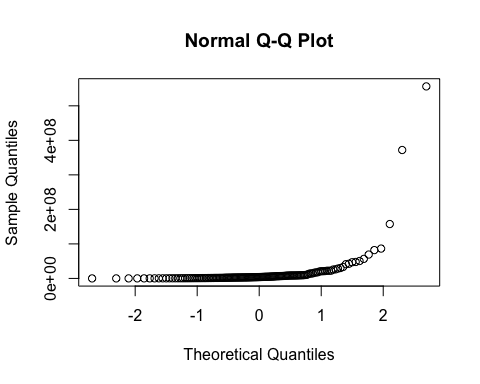
x = gapminder[gapminder$year <= 1952,c(5)] #population  
hist(log10(x$pop), breaks=20)



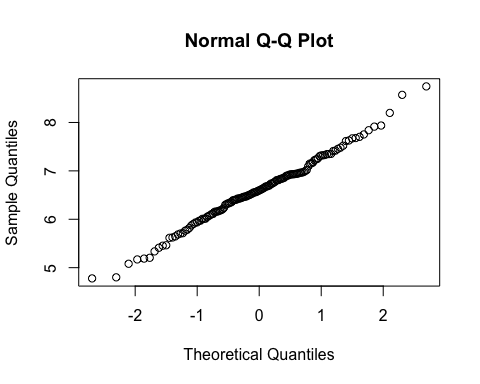
sd(log10(x$pop)) #Standard Dev

## [1] 0.7070292

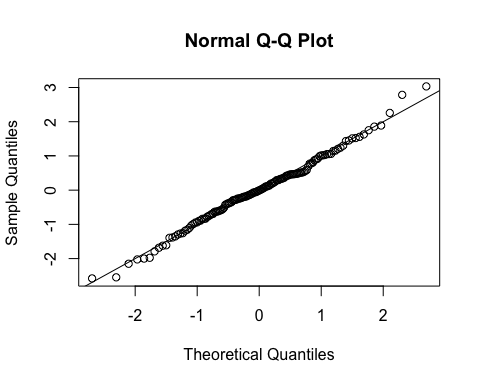
qqnorm(x$pop) #Q-Q Plot



#2.2 Standardize the population  
#vector of x of Log10 of 1952 populations  
x = gapminder[gapminder$year <= 1952, c(5)]  
x <- log10(x) #new Log10 vector  
qqnorm(x$pop) #Q-Q Plot



z <- ((x$pop - mean(x$pop))/sd(x$pop)) #standardize the population vector  
qqnorm(z) #Q-Q Plot  
abline(0,1)



max(z) #z - score

## [1] 3.03194

# 2.3 Normal Distrobution Approximation  
x = gapminder[gapminder$year <= 1952, c(5)]  
x <- log10(floor(x$pop)) #new Log10 vector  
F = function(q) pnorm(q, mean=mean(x), sd=sd(x))  
n = length(x) #number of countries  
(F(7) - F(6)) \* n

## [1] 73.27622

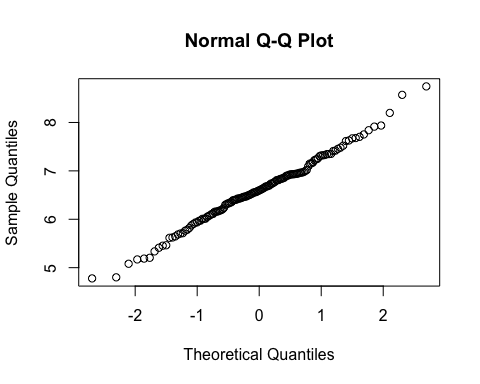
mean(z)

## [1] -2.410962e-16

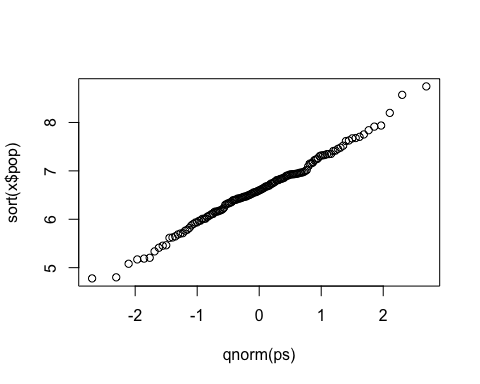
head(pnorm(z))

## [1] 0.6766035 0.2425973 0.6976099 0.5140721 0.8212934 0.6834280

#Problem 2.3  
x = gapminder[gapminder$year <= 1952, c(5)]  
x <- log10(x) #new Log10 vector  
qqnorm(x$pop) #Q-Q Plot



n = length(x$pop)  
ps = ((1:n) - 0.5)/n  
plot(qnorm(ps), sort(x$pop))



qnorm(ps)

## [1] -2.69483954 -2.30571146 -2.10592040 -1.96602455 -1.85651433  
## [6] -1.76558923 -1.68728119 -1.61813363 -1.55595703 -1.49927303  
## [11] -1.44703428 -1.39847048 -1.35299778 -1.31016257 -1.26960504  
## [16] -1.23103477 -1.19421375 -1.15894434 -1.12506054 -1.09242148  
## [21] -1.06090650 -1.03041145 -1.00084571 -0.97212993 -0.94419419  
## [26] -0.91697655 -0.89042181 -0.86448058 -0.83910846 -0.81426535  
## [31] -0.78991491 -0.76602408 -0.74256270 -0.71950311 -0.69681993  
## [36] -0.67448975 -0.65249095 -0.63080350 -0.60940878 -0.58828945  
## [41] -0.56742932 -0.54681325 -0.52642702 -0.50625727 -0.48629142  
## [46] -0.46651758 -0.44692449 -0.42750151 -0.40823849 -0.38912579  
## [51] -0.37015419 -0.35131491 -0.33259950 -0.31399988 -0.29550827  
## [56] -0.27711717 -0.25881932 -0.24060773 -0.22247560 -0.20441633  
## [61] -0.18642349 -0.16849080 -0.15061213 -0.13278149 -0.11499296  
## [66] -0.09724075 -0.07951913 -0.06182245 -0.04414511 -0.02648156  
## [71] -0.00882627 0.00882627 0.02648156 0.04414511 0.06182245  
## [76] 0.07951913 0.09724075 0.11499296 0.13278149 0.15061213  
## [81] 0.16849080 0.18642349 0.20441633 0.22247560 0.24060773  
## [86] 0.25881932 0.27711717 0.29550827 0.31399988 0.33259950  
## [91] 0.35131491 0.37015419 0.38912579 0.40823849 0.42750151  
## [96] 0.44692449 0.46651758 0.48629142 0.50625727 0.52642702  
## [101] 0.54681325 0.56742932 0.58828945 0.60940878 0.63080350  
## [106] 0.65249095 0.67448975 0.69681993 0.71950311 0.74256270  
## [111] 0.76602408 0.78991491 0.81426535 0.83910846 0.86448058  
## [116] 0.89042181 0.91697655 0.94419419 0.97212993 1.00084571  
## [121] 1.03041145 1.06090650 1.09242148 1.12506054 1.15894434  
## [126] 1.19421375 1.23103477 1.26960504 1.31016257 1.35299778  
## [131] 1.39847048 1.44703428 1.49927303 1.55595703 1.61813363  
## [136] 1.68728119 1.76558923 1.85651433 1.96602455 2.10592040  
## [141] 2.30571146 2.69483954