Assignment2\_final.R

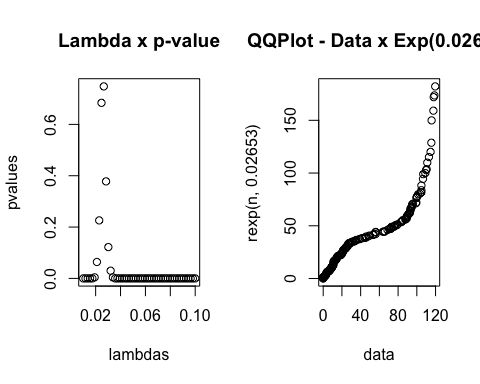
sarasal

Tue Feb 27 17:56:54 2018

### Exercise 1  
  
# Question 1.1  
data <- read.table(file = "telephone.txt", header = TRUE)  
data = unlist(data, use.names = FALSE)   
  
# The statistic to be used is the median - t = median for the given population's sample  
t=median(data)   
  
numLambdas=50  
B=1000;   
tstar=numeric(B)  
n=length(data)  
  
#Initially the interval for lambda will be split into equidistant values for lambda. Afterwards, the p-value  
#will be calculated for each of them.  
  
#Spliting the lambda interval in 50 different points:  
lambdas=numeric(numLambdas)  
pvalues=numeric(numLambdas)  
  
lambdas=seq(0.01,0.1,length=50)  
  
for (j in 1:numLambdas) {  
 # Creating Xstars and surrogate Tstars  
 tstar=numeric(B) #Creating a Tstar vector to be used in the Bootstrap test  
 for (i in 1:B){  
 xstar=rexp(n,lambdas[j]) # generating simulated samples  
 tstar[i]=median(xstar) #generating surrogated ts  
 }  
  
 pl=sum(tstar<t)/B  
 pr=sum(tstar>t)/B  
 pvalues[j]=2\*min(pl,pr)  
 print(j);print(lambdas[j]);print(pvalues[j])  
}

## [1] 1  
## [1] 0.01  
## [1] 0  
## [1] 2  
## [1] 0.01183673  
## [1] 0  
## [1] 3  
## [1] 0.01367347  
## [1] 0  
## [1] 4  
## [1] 0.0155102  
## [1] 0  
## [1] 5  
## [1] 0.01734694  
## [1] 0  
## [1] 6  
## [1] 0.01918367  
## [1] 0.004  
## [1] 7  
## [1] 0.02102041  
## [1] 0.064  
## [1] 8  
## [1] 0.02285714  
## [1] 0.226  
## [1] 9  
## [1] 0.02469388  
## [1] 0.684  
## [1] 10  
## [1] 0.02653061  
## [1] 0.748  
## [1] 11  
## [1] 0.02836735  
## [1] 0.378  
## [1] 12  
## [1] 0.03020408  
## [1] 0.122  
## [1] 13  
## [1] 0.03204082  
## [1] 0.03  
## [1] 14  
## [1] 0.03387755  
## [1] 0.004  
## [1] 15  
## [1] 0.03571429  
## [1] 0  
## [1] 16  
## [1] 0.03755102  
## [1] 0  
## [1] 17  
## [1] 0.03938776  
## [1] 0  
## [1] 18  
## [1] 0.04122449  
## [1] 0  
## [1] 19  
## [1] 0.04306122  
## [1] 0  
## [1] 20  
## [1] 0.04489796  
## [1] 0  
## [1] 21  
## [1] 0.04673469  
## [1] 0  
## [1] 22  
## [1] 0.04857143  
## [1] 0  
## [1] 23  
## [1] 0.05040816  
## [1] 0  
## [1] 24  
## [1] 0.0522449  
## [1] 0  
## [1] 25  
## [1] 0.05408163  
## [1] 0  
## [1] 26  
## [1] 0.05591837  
## [1] 0  
## [1] 27  
## [1] 0.0577551  
## [1] 0  
## [1] 28  
## [1] 0.05959184  
## [1] 0  
## [1] 29  
## [1] 0.06142857  
## [1] 0  
## [1] 30  
## [1] 0.06326531  
## [1] 0  
## [1] 31  
## [1] 0.06510204  
## [1] 0  
## [1] 32  
## [1] 0.06693878  
## [1] 0  
## [1] 33  
## [1] 0.06877551  
## [1] 0  
## [1] 34  
## [1] 0.07061224  
## [1] 0  
## [1] 35  
## [1] 0.07244898  
## [1] 0  
## [1] 36  
## [1] 0.07428571  
## [1] 0  
## [1] 37  
## [1] 0.07612245  
## [1] 0  
## [1] 38  
## [1] 0.07795918  
## [1] 0  
## [1] 39  
## [1] 0.07979592  
## [1] 0  
## [1] 40  
## [1] 0.08163265  
## [1] 0  
## [1] 41  
## [1] 0.08346939  
## [1] 0  
## [1] 42  
## [1] 0.08530612  
## [1] 0  
## [1] 43  
## [1] 0.08714286  
## [1] 0  
## [1] 44  
## [1] 0.08897959  
## [1] 0  
## [1] 45  
## [1] 0.09081633  
## [1] 0  
## [1] 46  
## [1] 0.09265306  
## [1] 0  
## [1] 47  
## [1] 0.0944898  
## [1] 0  
## [1] 48  
## [1] 0.09632653  
## [1] 0  
## [1] 49  
## [1] 0.09816327  
## [1] 0  
## [1] 50  
## [1] 0.1  
## [1] 0

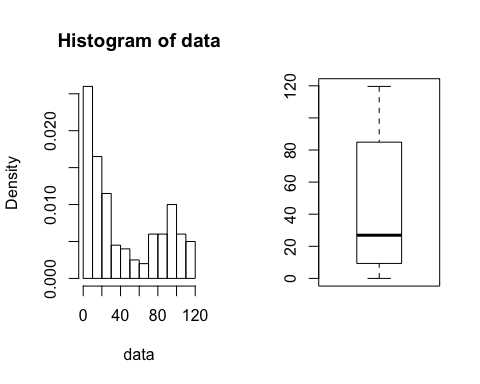
par(mfrow=c(1,2))  
plot(lambdas,pvalues, main="Lambda x p-value")  
qqplot(data, rexp(n,0.02653), main="QQPlot - Data x Exp(0.02653)")



# Question 1.2:  
# We generate the bootstrap interval, using the median as location estimator.  
  
TstarBootstrapInterval = numeric(B)  
for(i in 1:B){  
 Xstar = sample(data,replace=TRUE) #Generates a ramdon permutation of 'size of data' elements  
 TstarBootstrapInterval[i]=mean(Xstar) # Computes the statistic for the sample.  
}  
Tstar25=quantile(TstarBootstrapInterval,0.025)  
Tstar975=quantile(TstarBootstrapInterval,0.975)  
  
T1 = mean(data)  
c(2\*T1-Tstar975,2\*T1-Tstar25) # ==> mean is in this interval

## 97.5% 2.5%   
## 38.34240 48.80156

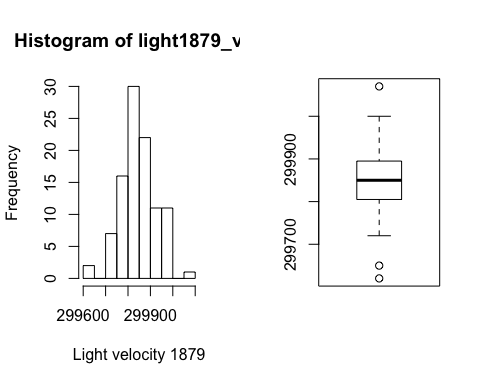
# 97.5% 2.5%   
# 38.12714 48.77792 Bootstrap interval for the mean of this population with 95% confidence.  
  
par(mfrow=c(1,2))  
hist(data, prob=T)  
boxplot(data)



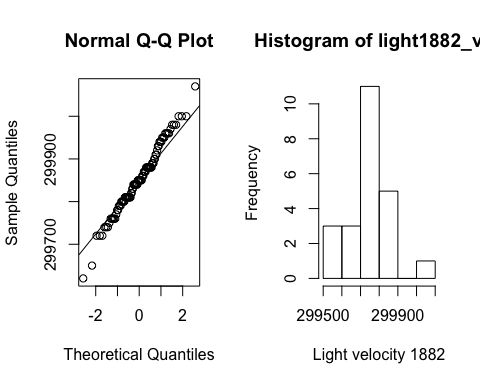
### Exercise 2  
light1879\_dataframe=read.table("light1879.txt", header = FALSE)  
light1882\_dataframe=read.table("light1882.txt", fill = TRUE)  
light\_dataframe= read.table("light.txt", header = FALSE)  
  
light1879\_vec = unlist(light1879\_dataframe, use.names = FALSE)  
light1879\_vec = light1879\_vec + 299000  
  
light1882\_vec = unlist(light1882\_dataframe, use.names = FALSE)  
light1882\_vec <- light1882\_vec[!is.na(light1882\_vec)]  
light1882\_vec = light1882\_vec + 299000  
  
light\_vec = unlist(light\_dataframe, use.names = FALSE)  
light\_vec = 7.442 / (((light\_vec/1000) + 24.8)/1000000)  
  
# Question 2.1  
par(mfrow=c(1,2))  
  
median(light1879\_vec)

## [1] 299850

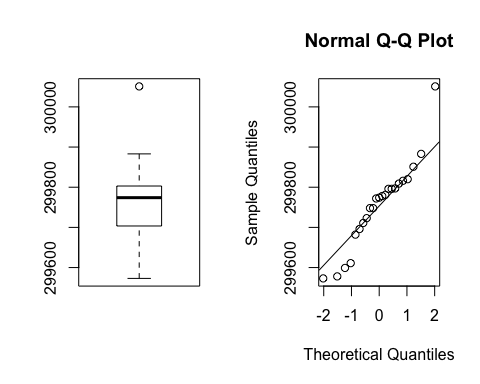
hist(light1879\_vec, xlab = "Light velocity 1879")  
boxplot(light1879\_vec)



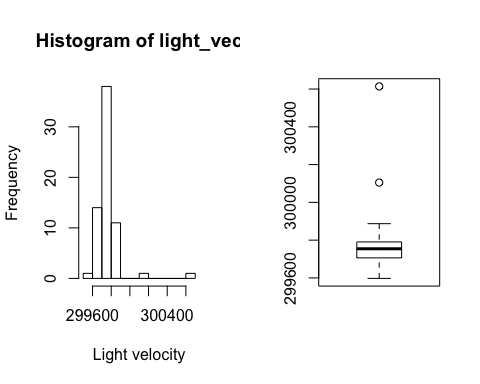
qqnorm(light1879\_vec)  
qqline(light1879\_vec)  
  
hist(light1882\_vec, xlab = "Light velocity 1882")



boxplot(light1882\_vec)  
qqnorm(light1882\_vec)  
qqline(light1882\_vec)



hist(light\_vec, xlab = "Light velocity")  
boxplot(light\_vec)



qqnorm(light\_vec)  
qqline(light\_vec)  
  
# Question 2.2  
# Dataset 1879  
#### MEAN   
B=1000  
Tstar=numeric(B)  
for(i in 1:B) {  
 Xstar=sample(light1879\_vec, replace=TRUE)  
 Tstar[i]=mean(Xstar)  
}  
Tstar25=quantile(Tstar,0.025)  
Tstar975=quantile(Tstar,0.975)  
Tmean= mean(light1879\_vec)  
c(2\*Tmean-Tstar975,2\*Tmean-Tstar25)

## 97.5% 2.5%   
## 299835.1 299867.9

#### Median  
Tstar=numeric(B)  
for(i in 1:B) {  
 Xstar=sample(light1879\_vec, replace=TRUE)  
 Tstar[i]=median(Xstar)  
}  
Tstar25=quantile(Tstar,0.025)  
Tstar975=quantile(Tstar,0.975)  
Tmedian = median(light1879\_vec)  
c(2\*Tmedian-Tstar975,2\*Tmedian-Tstar25)

## 97.5% 2.5%   
## 299830 299860

# Dataset 1882  
#### MEAN   
B=1000  
Tstar=numeric(B)  
for(i in 1:B) {  
 Xstar=sample(light1882\_vec, replace=TRUE)  
 Tstar[i]=mean(Xstar)  
}  
Tstar25=quantile(Tstar,0.025)  
Tstar975=quantile(Tstar,0.975)  
Tmean= mean(light1882\_vec)  
c(2\*Tmean-Tstar975,2\*Tmean-Tstar25)

## 97.5% 2.5%   
## 299711.1 299800.2

#### Median  
Tstar=numeric(B)  
for(i in 1:B) {  
 Xstar=sample(light1882\_vec, replace=TRUE)  
 Tstar[i]=median(Xstar)  
}  
Tstar25=quantile(Tstar,0.025)  
Tstar975=quantile(Tstar,0.975)  
Tmedian = median(light1882\_vec)  
c(2\*Tmedian-Tstar975,2\*Tmedian-Tstar25)

## 97.5% 2.5%   
## 299752 299825

# Dataset light.txt  
#### MEAN   
B=1000  
Tstar=numeric(B)  
for(i in 1:B) {  
 Xstar=sample(light\_vec, replace=TRUE)  
 Tstar[i]=mean(Xstar)  
}  
Tstar25=quantile(Tstar,0.025)  
Tstar975=quantile(Tstar,0.975)  
Tmean= mean(light\_vec)  
c(2\*Tmean-Tstar975,2\*Tmean-Tstar25)

## 97.5% 2.5%   
## 299730.5 299790.5

#### Median  
Tstar=numeric(B)  
for(i in 1:B) {  
 Xstar=sample(light\_vec, replace=TRUE)  
 Tstar[i]=median(Xstar)  
}  
Tstar25=quantile(Tstar,0.025)  
Tstar975=quantile(Tstar,0.975)  
Tmedian = median(light\_vec)  
c(2\*Tmedian-Tstar975,2\*Tmedian-Tstar25)

## 97.5% 2.5%   
## 299742.2 299766.4

# The confidence interval for μ with 95% confidence is measured as below  
sd\_light=sd(light\_vec)  
len\_sample=length(light\_vec)  
c(Tmean-2\*sd\_light/sqrt(len\_sample), Tmean+2\*sd\_light/sqrt(len\_sample))

## [1] 299731.9 299795.9

# Question 2.4  
# 1879 dataset is in normal distribution  
accurate\_light\_velocity = 299792.458  
t.test(light1879\_vec, mu=accurate\_light\_velocity, conf.level=0.95)

##   
## One Sample t-test  
##   
## data: light1879\_vec  
## t = 7.5866, df = 99, p-value = 1.824e-11  
## alternative hypothesis: true mean is not equal to 299792.5  
## 95 percent confidence interval:  
## 299836.7 299868.1  
## sample estimates:  
## mean of x   
## 299852.4

# 1882 and light.txt is not in normal distribution  
wilcox.test(light1882\_vec, mu= accurate\_light\_velocity)

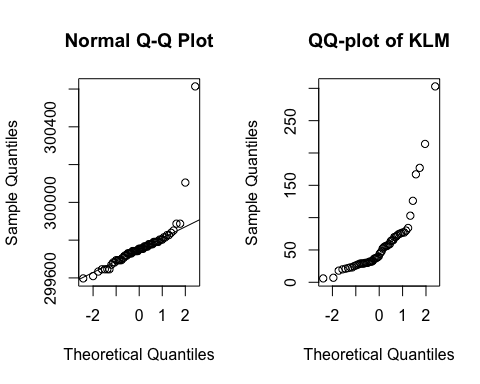
## Warning in wilcox.test.default(light1882\_vec, mu =  
## accurate\_light\_velocity): cannot compute exact p-value with ties

##   
## Wilcoxon signed rank test with continuity correction  
##   
## data: light1882\_vec  
## V = 83, p-value = 0.09736  
## alternative hypothesis: true location is not equal to 299792.5

wilcox.test(light\_vec, mu= accurate\_light\_velocity)

##   
## Wilcoxon signed rank test with continuity correction  
##   
## data: light\_vec  
## V = 387, p-value = 4.451e-06  
## alternative hypothesis: true location is not equal to 299792.5

### Exercise 3  
  
# Question 3.1  
klm=scan("klm.txt")  
qqnorm(klm,main="QQ-plot of KLM")



m=sum(klm>31)  
n= length(klm)  
binom.test(m,n,p=0.5)

##   
## Exact binomial test  
##   
## data: m and n  
## number of successes = 40, number of trials = 60, p-value = 0.01349  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.5331273 0.7831306  
## sample estimates:  
## probability of success   
## 0.6666667

# Question 3.2  
m=sum(klm>72)  
n= length(klm)  
binom.test(m,n,p=0.1)

##   
## Exact binomial test  
##   
## data: m and n  
## number of successes = 13, number of trials = 60, p-value =  
## 0.007478  
## alternative hypothesis: true probability of success is not equal to 0.1  
## 95 percent confidence interval:  
## 0.1207160 0.3419562  
## sample estimates:  
## probability of success   
## 0.2166667

### Exercise 4   
  
# Question 4.1  
par(mfrow=c(1,4))  
cloud\_data = read.table("clouds.txt", header=TRUE)  
  
summary(cloud\_data$seeded)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4.10 98.12 221.60 441.98 406.02 2745.60

sd(cloud\_data$seeded)

## [1] 650.7872

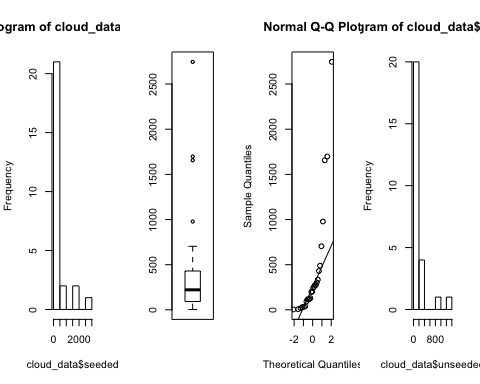
hist(cloud\_data$seeded)  
boxplot(cloud\_data$seeded)  
qqnorm(cloud\_data$seeded)  
qqline(cloud\_data$seeded)  
  
summary(cloud\_data$unseeded)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.01 24.82 44.20 164.56 159.20 1202.60

sd(cloud\_data$unseeded)

## [1] 278.4462

hist(cloud\_data$unseeded)



qqnorm(cloud\_data$unseeded)  
qqline(cloud\_data$unseeded)  
  
t.test(cloud\_data$seeded, cloud\_data$unseeded)

##   
## Welch Two Sample t-test  
##   
## data: cloud\_data$seeded and cloud\_data$unseeded  
## t = 1.9984, df = 33.856, p-value = 0.05375  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -4.740491 559.585876  
## sample estimates:  
## mean of x mean of y   
## 441.9846 164.5619

wilcox.test(cloud\_data$seeded, cloud\_data$unseeded)

## Warning in wilcox.test.default(cloud\_data$seeded, cloud\_data$unseeded):  
## cannot compute exact p-value with ties

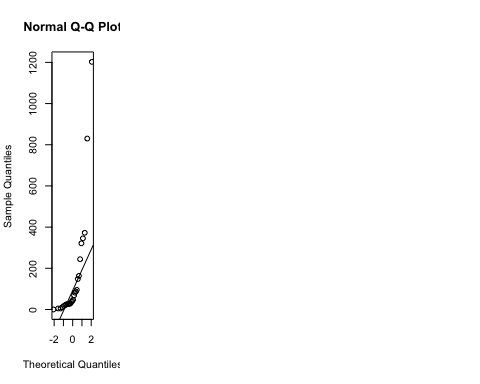
##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: cloud\_data$seeded and cloud\_data$unseeded  
## W = 473, p-value = 0.01383  
## alternative hypothesis: true location shift is not equal to 0

ks.test(cloud\_data$seeded, cloud\_data$unseeded)

## Warning in ks.test(cloud\_data$seeded, cloud\_data$unseeded): cannot compute  
## exact p-value with ties

##   
## Two-sample Kolmogorov-Smirnov test  
##   
## data: cloud\_data$seeded and cloud\_data$unseeded  
## D = 0.42308, p-value = 0.01905  
## alternative hypothesis: two-sided

# Question 4.2  
par(mfrow=c(1,4))



square\_root\_data = sqrt(cloud\_data)  
  
boxplot(square\_root\_data$seeded)  
summary(square\_root\_data$seeded)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.025 9.894 14.870 17.068 20.122 52.398

sd(square\_root\_data$seeded)

## [1] 12.51776

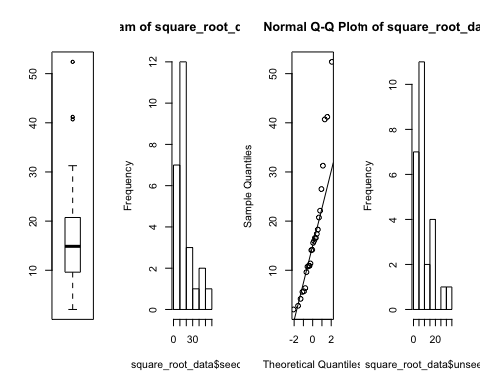
hist(square\_root\_data$seeded)  
qqnorm(square\_root\_data$seeded)  
qqline(square\_root\_data$seeded)  
  
summary(square\_root\_data$unseeded)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.100 4.982 6.644 9.931 12.615 34.679

sd(square\_root\_data$unseeded)

## [1] 8.28058

hist(square\_root\_data$unseeded)



qqnorm(square\_root\_data$unseeded)  
qqline(square\_root\_data$unseeded)  
  
t.test(square\_root\_data$seeded, square\_root\_data$unseeded)

##   
## Welch Two Sample t-test  
##   
## data: square\_root\_data$seeded and square\_root\_data$unseeded  
## t = 2.4246, df = 43.363, p-value = 0.01956  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 1.202087 13.071300  
## sample estimates:  
## mean of x mean of y   
## 17.068014 9.931321

wilcox.test(square\_root\_data$seeded, square\_root\_data$unseeded)

## Warning in wilcox.test.default(square\_root\_data$seeded, square\_root\_data  
## $unseeded): cannot compute exact p-value with ties

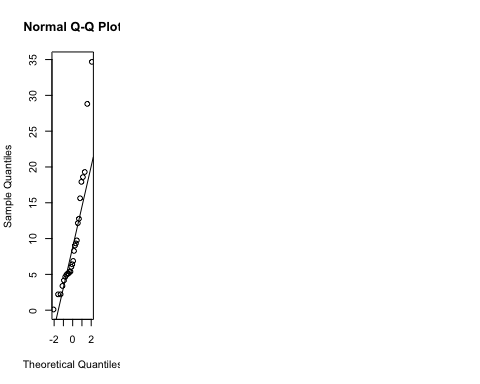
##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: square\_root\_data$seeded and square\_root\_data$unseeded  
## W = 473, p-value = 0.01383  
## alternative hypothesis: true location shift is not equal to 0

ks.test(square\_root\_data$seeded, square\_root\_data$unseeded)

## Warning in ks.test(square\_root\_data$seeded, square\_root\_data$unseeded):  
## cannot compute exact p-value with ties

##   
## Two-sample Kolmogorov-Smirnov test  
##   
## data: square\_root\_data$seeded and square\_root\_data$unseeded  
## D = 0.42308, p-value = 0.01905  
## alternative hypothesis: two-sided

# Question 4.3  
par(mfrow=c(1,4))



square\_root\_of\_square\_root\_data = sqrt(square\_root\_data)  
summary(square\_root\_of\_square\_root\_data$seeded)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.423 3.145 3.855 3.879 4.484 7.239

sd(square\_root\_of\_square\_root\_data$seeded)

## [1] 1.449941

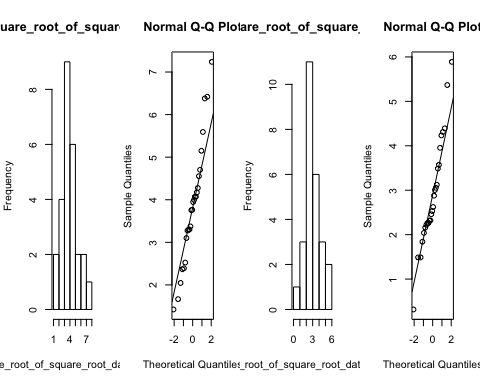
hist(square\_root\_of\_square\_root\_data$seeded)  
qqnorm(square\_root\_of\_square\_root\_data$seeded)  
qqline(square\_root\_of\_square\_root\_data$seeded)  
  
  
summary(square\_root\_of\_square\_root\_data$unseeded)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.3162 2.2320 2.5772 2.9073 3.5515 5.8888

sd(square\_root\_of\_square\_root\_data$unseeded)

## [1] 1.240097

hist(square\_root\_of\_square\_root\_data$unseeded)  
qqnorm(square\_root\_of\_square\_root\_data$unseeded)  
qqline(square\_root\_of\_square\_root\_data$unseeded)



t.test(square\_root\_of\_square\_root\_data$seeded, square\_root\_of\_square\_root\_data$unseeded)

##   
## Welch Two Sample t-test  
##   
## data: square\_root\_of\_square\_root\_data$seeded and square\_root\_of\_square\_root\_data$unseeded  
## t = 2.5968, df = 48.826, p-value = 0.0124  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.2196477 1.7236468  
## sample estimates:  
## mean of x mean of y   
## 3.878988 2.907340

wilcox.test(square\_root\_of\_square\_root\_data$seeded, square\_root\_of\_square\_root\_data$unseeded)

## Warning in wilcox.test.default(square\_root\_of\_square\_root\_data$seeded,  
## square\_root\_of\_square\_root\_data$unseeded): cannot compute exact p-value  
## with ties

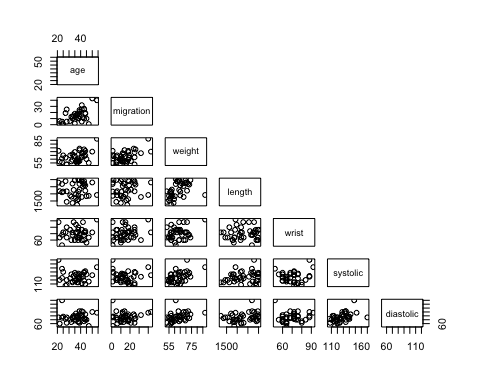
##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: square\_root\_of\_square\_root\_data$seeded and square\_root\_of\_square\_root\_data$unseeded  
## W = 473, p-value = 0.01383  
## alternative hypothesis: true location shift is not equal to 0

ks.test(square\_root\_of\_square\_root\_data$seeded, square\_root\_of\_square\_root\_data$unseeded)

## Warning in ks.test(square\_root\_of\_square\_root\_data$seeded,  
## square\_root\_of\_square\_root\_data$unseeded): cannot compute exact p-value  
## with ties

##   
## Two-sample Kolmogorov-Smirnov test  
##   
## data: square\_root\_of\_square\_root\_data$seeded and square\_root\_of\_square\_root\_data$unseeded  
## D = 0.42308, p-value = 0.01905  
## alternative hypothesis: two-sided

###Exercise 5  
  
# Question 5.1  
peruvians=read.table("peruvians.txt",header=TRUE)  
peruvians = peruvians[,-c(5,6,7)]  
pairs(peruvians, upper.panel=NULL)



# Question 5.2  
attach(peruvians)   
  
# Test 5.2.1 (migration x age)  
peruvians[,c(1,2)]

## age migration  
## 1 21 1  
## 2 22 6  
## 3 24 5  
## 4 24 1  
## 5 25 1  
## 6 27 19  
## 7 28 5  
## 8 28 25  
## 9 31 6  
## 10 32 13  
## 11 33 13  
## 12 33 10  
## 13 34 15  
## 14 35 18  
## 15 35 2  
## 16 36 12  
## 17 36 15  
## 18 37 16  
## 19 37 17  
## 20 38 10  
## 21 38 18  
## 22 38 11  
## 23 38 11  
## 24 39 21  
## 25 39 24  
## 26 39 14  
## 27 41 25  
## 28 41 32  
## 29 41 5  
## 30 42 12  
## 31 43 25  
## 32 43 26  
## 33 43 10  
## 34 44 19  
## 35 44 18  
## 36 45 10  
## 37 47 1  
## 38 50 43  
## 39 54 40

cor.test(migration, age,method="spearman")

## Warning in cor.test.default(migration, age, method = "spearman"): Cannot  
## compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: migration and age  
## S = 5176.6, p-value = 0.002189  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.4760575

# Test 5.2.2 (migration x weight)  
cor.test(migration, weight,method="spearman")

## Warning in cor.test.default(migration, weight, method = "spearman"): Cannot  
## compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: migration and weight  
## S = 6415.1, p-value = 0.02861  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.3506956

# Test 5.2.3 (migration x length)  
cor.test(migration, length,method="spearman")

## Warning in cor.test.default(migration, length, method = "spearman"): Cannot  
## compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: migration and length  
## S = 9044.3, p-value = 0.6087  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.08458432

# Test 5.2.4 (migration x wrist)  
cor.test(migration, wrist,method="spearman")

## Warning in cor.test.default(migration, wrist, method = "spearman"): Cannot  
## compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: migration and wrist  
## S = 7712.8, p-value = 0.1797  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.2193498

# Test 5.2.5 (migration x diastolic)  
cor.test(migration, diastolic,method="spearman")

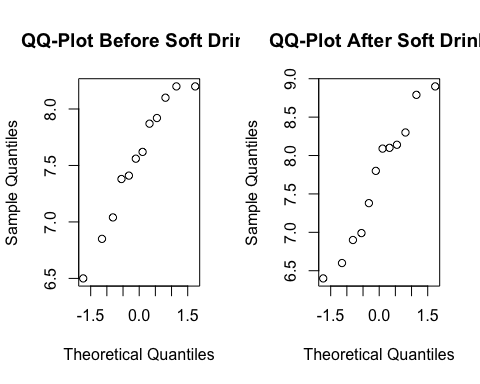
## Warning in cor.test.default(migration, diastolic, method = "spearman"):  
## Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: migration and diastolic  
## S = 9137.6, p-value = 0.6494  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.07514098

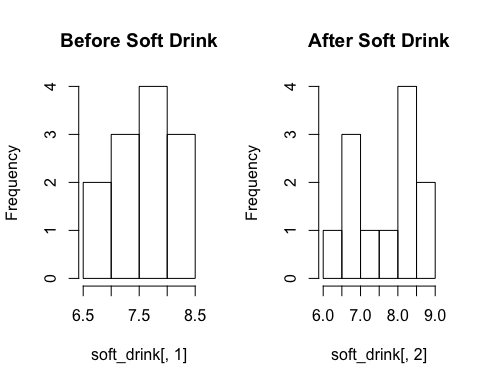
### Exercise 6  
  
# Question 6.1  
run=read.table("run.txt")  
run

## before after drink  
## 1 7.92 6.99 lemo  
## 2 7.56 8.14 lemo  
## 3 7.62 8.09 lemo  
## 4 6.85 6.40 lemo  
## 5 7.87 8.79 lemo  
## 6 7.41 6.60 lemo  
## 7 7.38 7.38 lemo  
## 8 7.04 7.80 lemo  
## 9 8.10 8.90 lemo  
## 10 6.50 6.90 lemo  
## 11 8.20 8.10 lemo  
## 12 8.20 8.30 lemo  
## 13 7.33 7.11 energy  
## 14 7.70 8.30 energy  
## 15 8.37 8.46 energy  
## 16 7.70 7.60 energy  
## 17 7.80 7.20 energy  
## 18 7.60 7.51 energy  
## 19 7.70 7.30 energy  
## 20 7.60 7.00 energy  
## 21 8.00 7.70 energy  
## 22 7.65 7.51 energy  
## 23 8.04 7.95 energy  
## 24 7.30 7.30 energy

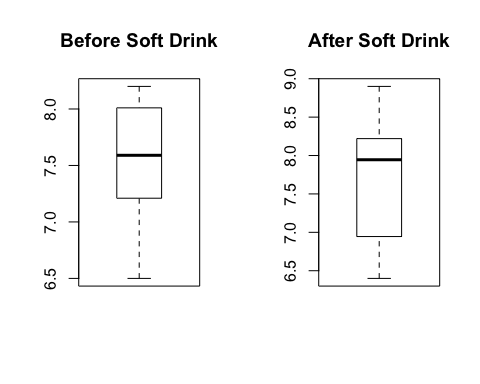
soft\_drink=run[run$drink=='lemo', 1:2]  
energy=run[run$drink=='energy', 1:2]  
  
par(mfrow=c(1,2))  
qqnorm(soft\_drink[,1],main = 'QQ-Plot Before Soft Drink')  
qqnorm(soft\_drink[,2],main = 'QQ-Plot After Soft Drink')



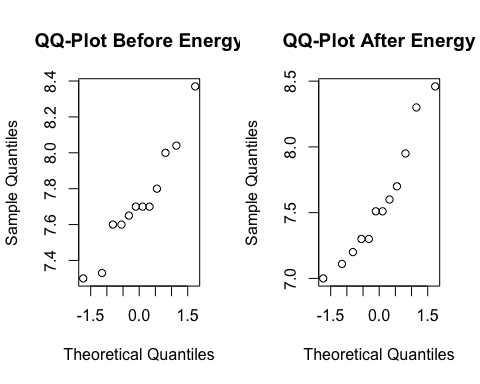
hist(soft\_drink[,1],main = 'Before Soft Drink')  
hist(soft\_drink[,2],main = 'After Soft Drink')



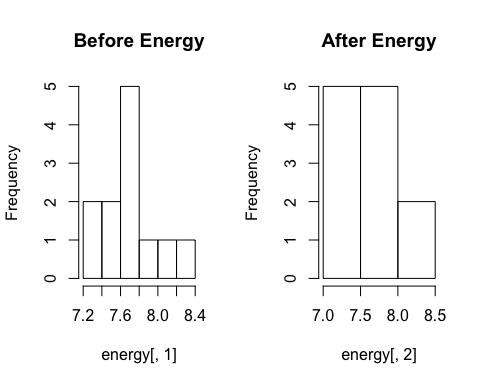
boxplot(soft\_drink[,1],main = 'Before Soft Drink')  
boxplot(soft\_drink[,2],main = 'After Soft Drink')



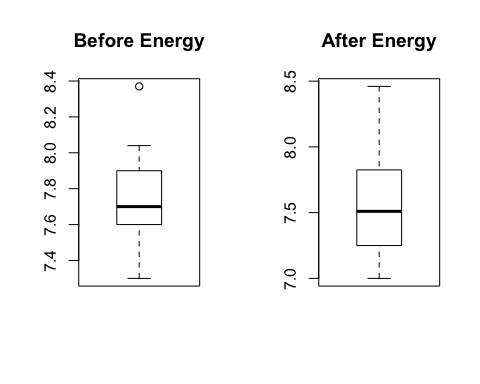
qqnorm(energy[,1],main = 'QQ-Plot Before Energy')  
qqnorm(energy[,2],main = 'QQ-Plot After Energy')



hist(energy[,1],main = 'Before Energy')  
hist(energy[,2],main = 'After Energy')



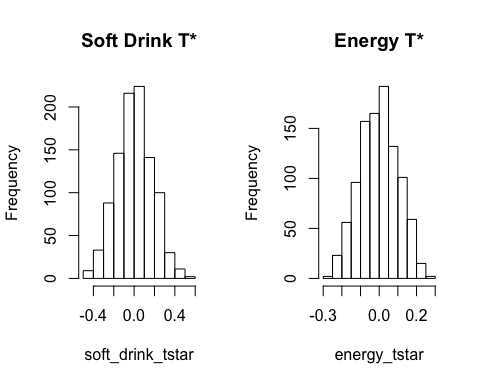
boxplot(energy[,1],main = 'Before Energy')  
boxplot(energy[,2],main = 'After Energy')



# Question 6.2  
soft\_drink=run[run$drink=='lemo', 1:2]  
energy=run[run$drink=='energy', 1:2]  
mystat=function(x,y) {mean(x-y)}  
B=1000  
  
#soft drink  
soft\_drink\_tstar=numeric(B)  
for (i in 1:B){  
 soft\_drink\_star=t(apply(cbind(soft\_drink[,1],soft\_drink[,2]),1,sample))  
 soft\_drink\_tstar[i]=mystat(soft\_drink\_star[,1],soft\_drink\_star[,2])  
}  
soft\_drink\_myt=mystat(soft\_drink\_star[,1],soft\_drink\_star[,2])  
hist(soft\_drink\_tstar, main = 'Soft Drink T\*')  
pl=sum(soft\_drink\_tstar<soft\_drink\_myt)/B  
pr=sum(soft\_drink\_tstar>soft\_drink\_myt)/B  
p\_soft\_drink=2\*min(pl,pr)  
p\_soft\_drink

## [1] 0.52

#energy drink  
energy\_tstar=numeric(B)  
for (i in 1:B){  
 energy\_star=t(apply(cbind(energy[,1],energy[,2]),1,sample))  
 energy\_tstar[i]=mystat(energy\_star[,1],energy\_star[,2])  
}  
energy\_myt=mystat(energy\_star[,1],energy\_star[,2])  
hist(energy\_tstar, main = 'Energy T\*')



pl=sum(energy\_tstar<energy\_myt)/B  
pr=sum(energy\_tstar>energy\_myt)/B  
p\_energy=2\*min(pl,pr)  
p\_energy

## [1] 0.24

# Question 6.3  
n=nrow(run)  
diff\_time <- data.frame(numeric(n), character(n))  
diff\_time[,1]= run[,2]- run[,1]  
diff\_time[,2]= run[,3]  
wilcox.test(diff\_time[1:12,1],diff\_time[13:24,1])

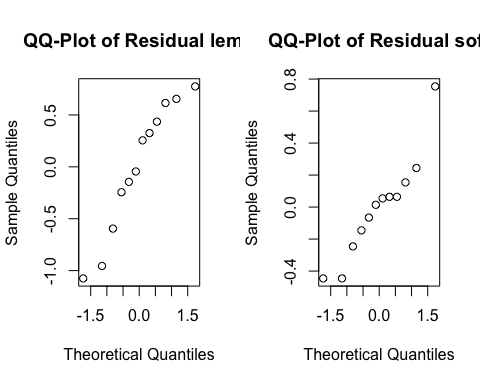
## Warning in wilcox.test.default(diff\_time[1:12, 1], diff\_time[13:24, 1]):  
## cannot compute exact p-value with ties

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: diff\_time[1:12, 1] and diff\_time[13:24, 1]  
## W = 98.5, p-value = 0.1332  
## alternative hypothesis: true location shift is not equal to 0

# Question 6.6  
par(mfrow=c(1,2))  
as.matrix(diff\_time)

## numeric.n. character.n.  
## [1,] "-0.93" "lemo"   
## [2,] " 0.58" "lemo"   
## [3,] " 0.47" "lemo"   
## [4,] "-0.45" "lemo"   
## [5,] " 0.92" "lemo"   
## [6,] "-0.81" "lemo"   
## [7,] " 0.00" "lemo"   
## [8,] " 0.76" "lemo"   
## [9,] " 0.80" "lemo"   
## [10,] " 0.40" "lemo"   
## [11,] "-0.10" "lemo"   
## [12,] " 0.10" "lemo"   
## [13,] "-0.22" "energy"   
## [14,] " 0.60" "energy"   
## [15,] " 0.09" "energy"   
## [16,] "-0.10" "energy"   
## [17,] "-0.60" "energy"   
## [18,] "-0.09" "energy"   
## [19,] "-0.40" "energy"   
## [20,] "-0.60" "energy"   
## [21,] "-0.30" "energy"   
## [22,] "-0.14" "energy"   
## [23,] "-0.09" "energy"   
## [24,] " 0.00" "energy"

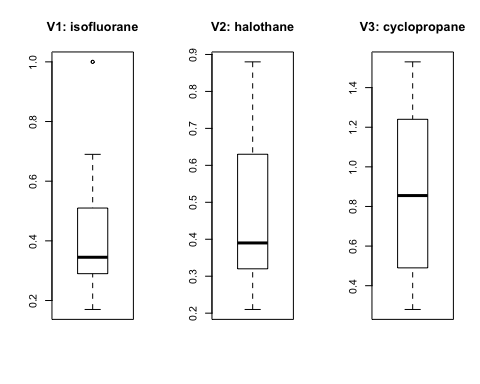
diff\_frame= data.frame(time\_diff=as.vector(diff\_time[,1]), drink\_type=factor(rep(1:2,each=12)))  
diff\_time\_aov= lm(time\_diff~drink\_type,data = diff\_frame)  
time\_residual=residuals(diff\_time\_aov)  
  
qqnorm(time\_residual[1:12],main = 'QQ-Plot of Residual lemo')  
qqnorm(time\_residual[13:24],main = 'QQ-Plot of Residual soft')



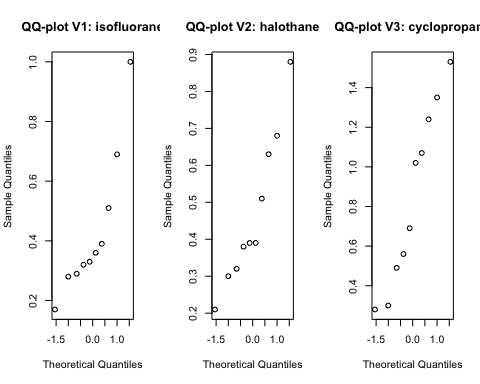
#t.test(diff\_time[1:12,1],diff\_time[13:24,1],paired = TRUE)  
  
  
### Exercise 7  
  
# Question 7.1  
dogs=read.table("dogs.txt",header = TRUE)  
dogs

## isofluorane halothane cyclopropane  
## 1 0.28 0.30 1.07  
## 2 0.51 0.39 1.35  
## 3 1.00 0.63 0.69  
## 4 0.39 0.68 0.28  
## 5 0.29 0.38 1.24  
## 6 0.36 0.21 1.53  
## 7 0.32 0.88 0.49  
## 8 0.69 0.39 0.56  
## 9 0.17 0.51 1.02  
## 10 0.33 0.32 0.30

par(mfrow=c(1,3))  
boxplot(dogs[,1], main = "V1: isofluorane")  
boxplot(dogs[,2], main = "V2: halothane")  
boxplot(dogs[,3], main = "V3: cyclopropane")



qqnorm(dogs[,1], main = "QQ-plot V1: isofluorane")  
qqnorm(dogs[,2], main = "QQ-plot V2: halothane")  
qqnorm(dogs[,3], main = "QQ-plot V3: cyclopropane")



# Question 7.2  
dogsframe= data.frame(plasma=as.vector(as.matrix(dogs)), drugs=factor(rep(1:3,each=10)))  
dogsaov= lm(plasma~drugs,data = dogsframe)  
anova(dogsaov)

## Analysis of Variance Table  
##   
## Response: plasma  
## Df Sum Sq Mean Sq F value Pr(>F)   
## drugs 2 1.0808 0.54040 5.355 0.011 \*  
## Residuals 27 2.7247 0.10092   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(dogsaov)

##   
## Call:  
## lm(formula = plasma ~ drugs, data = dogsframe)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.5730 -0.1608 -0.0790 0.2000 0.6770   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.4340 0.1005 4.320 0.000189 \*\*\*  
## drugs2 0.0350 0.1421 0.246 0.807266   
## drugs3 0.4190 0.1421 2.949 0.006504 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3177 on 27 degrees of freedom  
## Multiple R-squared: 0.284, Adjusted R-squared: 0.231   
## F-statistic: 5.355 on 2 and 27 DF, p-value: 0.011

drug1 = 0.4340  
drug2 = drug1 + 0.0350  
drug3 = drug1 + 0.4190  
drug1; drug2; drug3

## [1] 0.434

## [1] 0.469

## [1] 0.853

# Question 7.3  
attach(dogsframe)  
kruskal.test(plasma,drugs)

##   
## Kruskal-Wallis rank sum test  
##   
## data: plasma and drugs  
## Kruskal-Wallis chi-squared = 5.6442, df = 2, p-value = 0.05948