

Assignment 1

Chang Liu, Kai Zhang, Sherida van den Bent

2020/02/20

Exercise 1

First, we construct a simple function that will return the power of the t-test using the given parameters (n,m,mu,nu,sd).

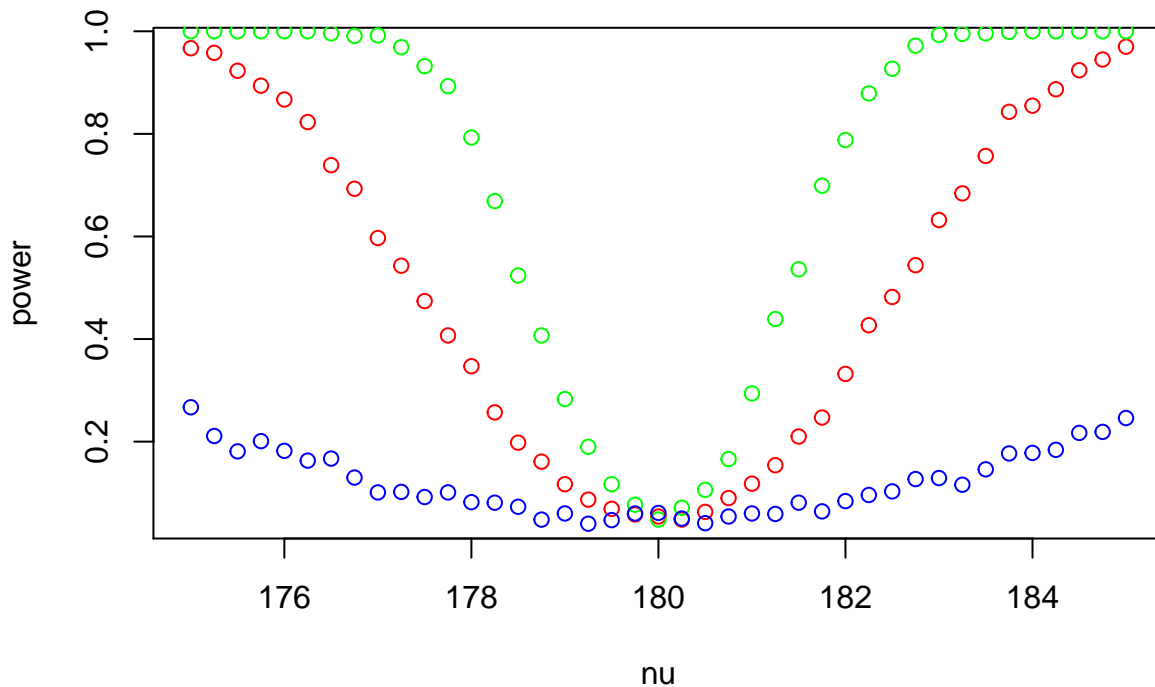
```
powerOfTtest <- function(n, m, mu, nu, sd)
{
  B = 1000; p = numeric(B); power = numeric(length(nu))
  for (i in 1:length(nu))
  {
    for (b in 1:B)
    {
      x = rnorm(n, mu,sd); y = rnorm(m, nu[i],sd)
      p[b] = t.test(x,y,var.equal=TRUE)[[3]]
    }
    power[i] = mean(p<0.05)
  }
  power
}
```

a) & b) & c) Using the above function, we can now easily calculate the power of the t-test with the given parameters. This is plotted for all sets of parameters, with red for subquestion a, green for subquestion b, and blue for subquestion c.

```
nu = seq(175,185,by=0.25); mu = 180 # these parameters stay equal in all sets
# compute and plot power function with parameters : n = m = 30, and sd = 5
n = m = 30; sd = 5
power = powerOfTtest(n, m, mu, nu, sd)
plot(nu, power, col = 'red')

# compute and plot power function with parameters : n = m = 100, and sd = 5.
n = m = 100; sd = 5
power = powerOfTtest(n, m, mu, nu, sd)
points(nu, power, col = 'green')

# compute power function with parameters : n = m = 30, and sd = 15.
n = m = 30; sd = 15
power = powerOfTtest(n, m, mu, nu, sd)
points(nu, power, col = 'blue')
```



d) First, with fixed n, m, sd and μ . As the second sample's mean of the sampling distribution (ν) goes closer to μ , the power of p-value could be rather low which suggests t-test tends to NOT reject the null hypothesis and gives the right result. Second, comparing plot from problem a and b, b's up-side-down bell-shaped plot is thinner than a's plot, indicating as sample size increase t-test becomes more strict, t-test will NOT reject null hypothesis only when two means fairly close to each other. The third conclusion comes from problem c, when standard deviation becomes larger, its plot became less smooth and t-test's performance became unstable. Because higher sample sizes yield higher power, increasing sample size may solve this problem.

Exercise 2

a) To investigate the normality for all three data sets, I choose to use Shapiro-Wilk normality test with an alpha level of 0.05. The null-hypothesis of this test is that the population is normally distributed.

```
light = 7.442 / ((scan('light.txt', quiet=TRUE) / 1000 + 24.8) / 10^6)
light1879 = scan('light1879.txt', quiet=TRUE) + 299000
light1882 = scan('light1882.txt', quiet=TRUE) + 299000
par(mfrow=c(1,3))
shapiro.test(light)[[2]]
```

```
## [1] 2.72356e-12
```

```
shapiro.test(light1879)[[2]]
```

```
## [1] 0.5137039
```

```
shapiro.test(light1882)[[2]]
```

```
## [1] 0.1111188
```

After computing all three data sets' p-value, the data from light1879 and light1882 have a p-value greater than 0.05. So I would deduce 'light1879' and 'light1882' is normally distributed, the data set 'light' is not normally distributed.

b) We can use t-distribution to calculate distribution confidence intervals even the distribution is not a normal distribution. `t.test` in R would give me confidence intervals directly.

```
lightCi = t.test(light)[[4]]
light1879Ci = t.test(light1879)[[4]]
light1882Ci = t.test(light1882)[[4]]
c(lightCi[1], lightCi[2])
```

```
## [1] 299731.9 299795.8
```

```
c(light1879Ci[1], light1879Ci[2])
```

```
## [1] 299836.7 299868.1
```

```
c(light1882Ci[1], light1882Ci[2])
```

```
## [1] 299709.9 299802.5
```

c) t-test will perform less accurately when the distribution is not normal, so I choose Wilcoxon signed rank test to find the first sample's p-value.

```
lightSpeed = 299792.458
p = wilcox.test(light,mu=lightSpeed)[[3]]
p1879 = t.test(light1879, mu=lightSpeed)[[3]]
p1882 = t.test(light1882,mu=lightSpeed)[[3]]
p; p1879; p1882;
```

```
## [1] 4.450593e-06
```

```
## [1] 1.823745e-11
```

```
## [1] 0.1189198
```

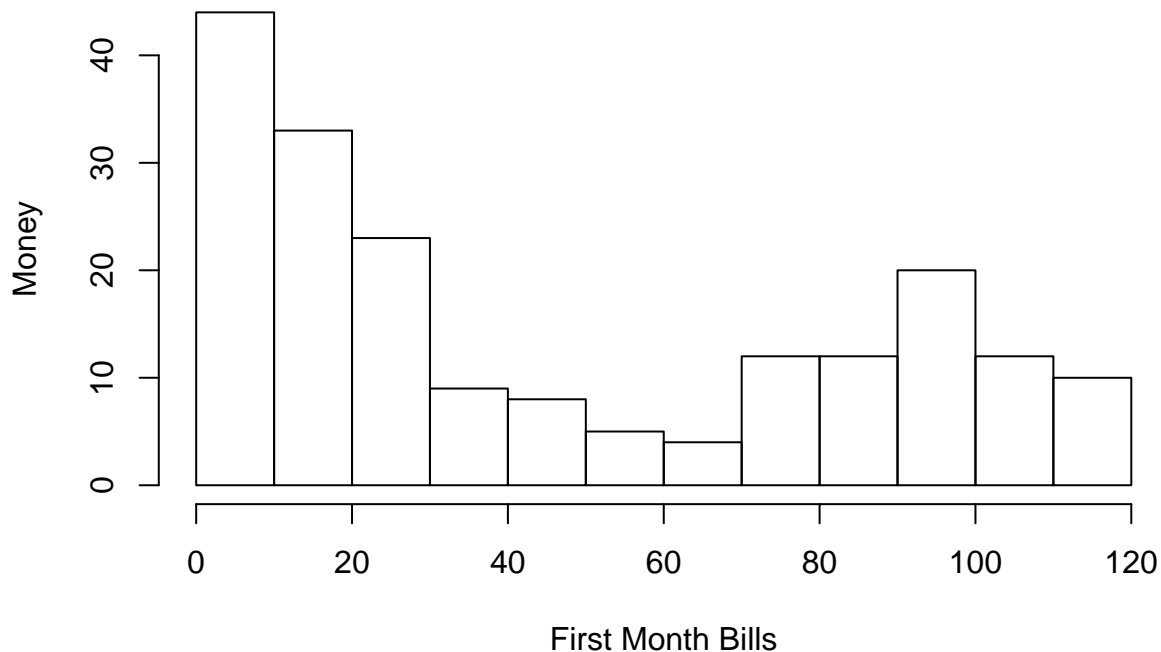
As result, only light1882's p-value ($0.1189198 > 0.05$), which makes it most accurate sample.

Exercise 3

a)

```
telephoneBills = read.table('telephone.txt', header=TRUE)
telephone = telephoneBills[telephoneBills$Bills!=0, ]
hist(telephone, xlab='First Month Bills', ylab='Money', main='Histogram of New Subscribers')
```

Histogram of New Subscribers



I think best plot to represent the distribution of subscribers is histogram. The strategy that the manager should adopt is to increase the preferential activities in the price range of 30-70 to promote low consumption users to increase consumption, and ultimately increase users in this price range. There is one inconsistencies in the data. It contains zero value which is inappropriate since the survey should target customers who have already spent. Hence zero-cost subscribers shouldn't be included.

b) I make a sequence between 0.01 and 0.1, and use bootstrap-test to test every single one of them to evaluate whether data fits exponential distribution.

```
lambdas=seq(0.01, 0.1, by=0.01)
B=1000
t=median(telephone)
n=length(telephone)
p=numeric(length(lambdas))
for (i in 1:length(lambdas)) {
  tstar=numeric(B)
  for (b in 1:B) {
    xstar=rexp(n, lambdas[i])
    tstar[b]=median(xstar)
  }
  pl=sum(tstar<t)/B
  pr=sum(tstar>t)/B
  p[i]=2*min(pl,pr)
}
p
```

```
## [1] 0.00 0.09 0.02 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

As result, when lambda equals 0.02, p-value = 0.074 > 0.05, so the data fit Exp(0.02)

c) To construct a 95% bootstrap confidence interval, first I should do bootstrap simulation to generate 1000 groups (X_1^*, \dots, X_N^*) and compute $T_i^* = \text{median}(X_1^*, \dots, X_N^*)$. With the formula for the bootstrap

confidence interval with confidence $1 - 2\alpha$: $[2T - T_{(1-\alpha)}^*, 2T - T_{(\alpha)}^*]$, I can now construct a 95% bootstrap confidence interval.

```
B = 1000
medians = numeric(B)
for (b in 1:B) {
  xstar=sample(telephone, size=length(telephone), replace=TRUE)
  medians[b] = median(xstar)
}
Tstar25 = quantile(medians, 0.025)
Tstar975 = quantile(medians, 0.975)
T1 = median(telephone)
c(2*T1-Tstar975, 2*T1-Tstar25)
```

```
## 97.5% 2.5%
## 16.430 36.575
```

d) For an exponential distribution, we have $E[X] = \frac{1}{\lambda}$. When applying bootstrap to simulate central limit theorem to an exponential distribution, we could expect $\hat{\lambda} = \frac{1}{\bar{X}}$.

```
B = 1000
sample_means = numeric(B)
medians = numeric(B)
for (b in 1:B) {
  xstar = sample(telephone, size=length(telephone), replace=TRUE)
  sample_means[b] = mean(xstar)
  medians[b] = median(xstar)
}

central = mean(sample_means)
lambda = 1 / central
lambda
```

```
## [1] 0.02195547
```

```
Tstar25 = quantile(medians, 0.025)
Tstar975 = quantile(medians, 0.975)
T1 = median(telephone)
c(2*T1-Tstar975, 2*T1-Tstar25)
```

```
## 97.5% 2.5%
## 17.515 36.460
```

So we have $\lambda = 0.022$, and CI for population median is $[16.12500, 36.56688]$.

e) I choose sign test to verify wheter the median is bigger or equal to 40, and wheter the probability less than 10 is at most 25%.

```
binom.test(sum(telephone>=40),length(telephone),p=0.5)[[3]]
```

```
## [1] 0.07091956
```

```
binom.test(sum(telephone<10),length(telephone),p=0.25, alternative='greater')[[3]]
```

```
## [1] 0.7714992
```

the p-value of first test $0.07091956 > 0.05$. Conclusion: H_0 is not rejected, median bill is bigger or equal to 40 euro. the p-value of first test $0.7715 > 0.05$. Conclusion: H_0 is not rejected, the fraction of bills less than 10 euro is at most 25%.

Exercise 4

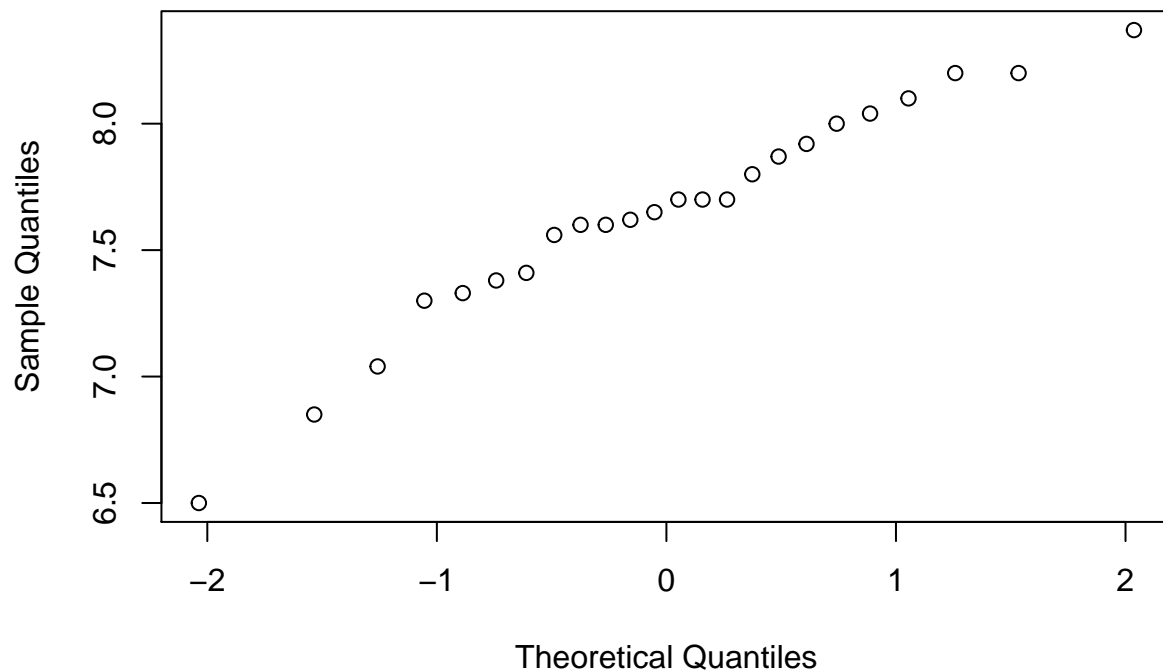
a)

```
run = read.table('run.txt')  
cor.test(run$before, run$after)
```

```
##  
## Pearson's product-moment correlation  
##  
## data: run$before and run$after  
## t = 3.8944, df = 22, p-value = 0.00078  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.3171271 0.8286612  
## sample estimates:  
## cor  
## 0.638803
```

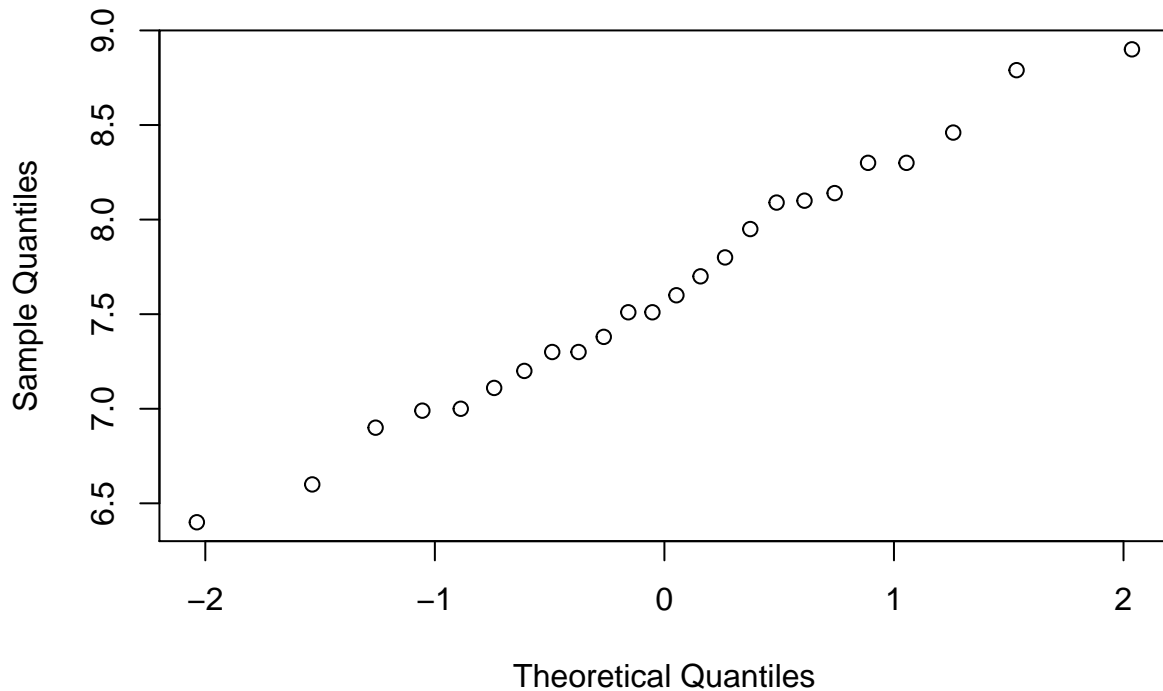
```
qqnorm(run$before)
```

Normal Q-Q Plot



```
qqnorm(run$after)
```

Normal Q-Q Plot



```
shapiro.test(run$before)[[2]]
```

```
## [1] 0.4168152
```

```
shapiro.test(run$after)[[2]]
```

```
## [1] 0.9463846
```

For both two columns' data are normally distributed, we could use Pearson's to test whether run times before drink and after are correlated, and finally we got p-value = 0.00078, which is much smaller than 0.05.

Moreover, we ran Shapiro-Wilk normality test and drew QQ-Plot to check both the "before running data" and the "after running data"'s normality.

As both data Shapiro test p-value results are bigger than 0.05 and QQ-Plots' dots are both nearly in a line, both two columns' data are normally distributed.

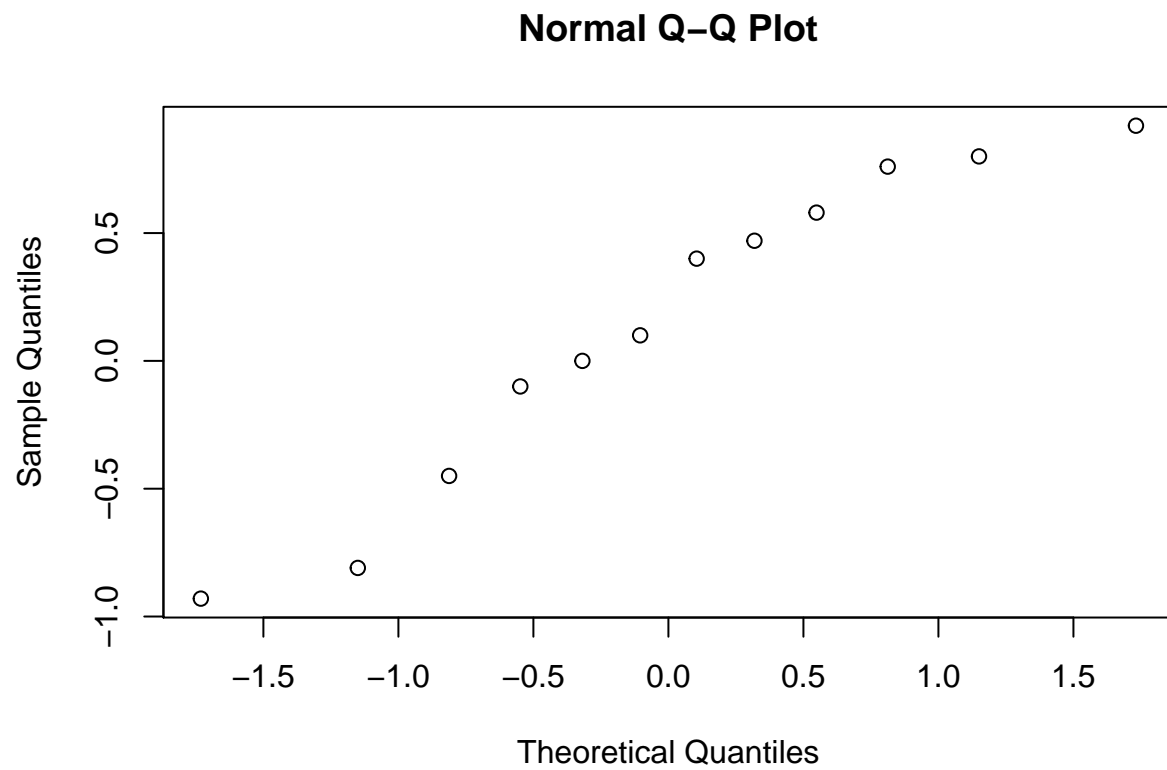
Conclusion: there is significant correlation, given dataset's normality.

b) For difference in speed test we will use t-test in both softdrink and the energy drink conditions. The test null hypothesis: $H_0 : H_{before} = H_{after}$, and alternative hypothesis: $H_{before} \neq H_{after}$.

```
lemo = run[run$drink=='lemon', ]
t.test(lemo$before, lemo$after, paired = TRUE)[[3]]
```

```
## [1] 0.4373423
```

```
qqnorm(lemo$after - lemo$before)
```

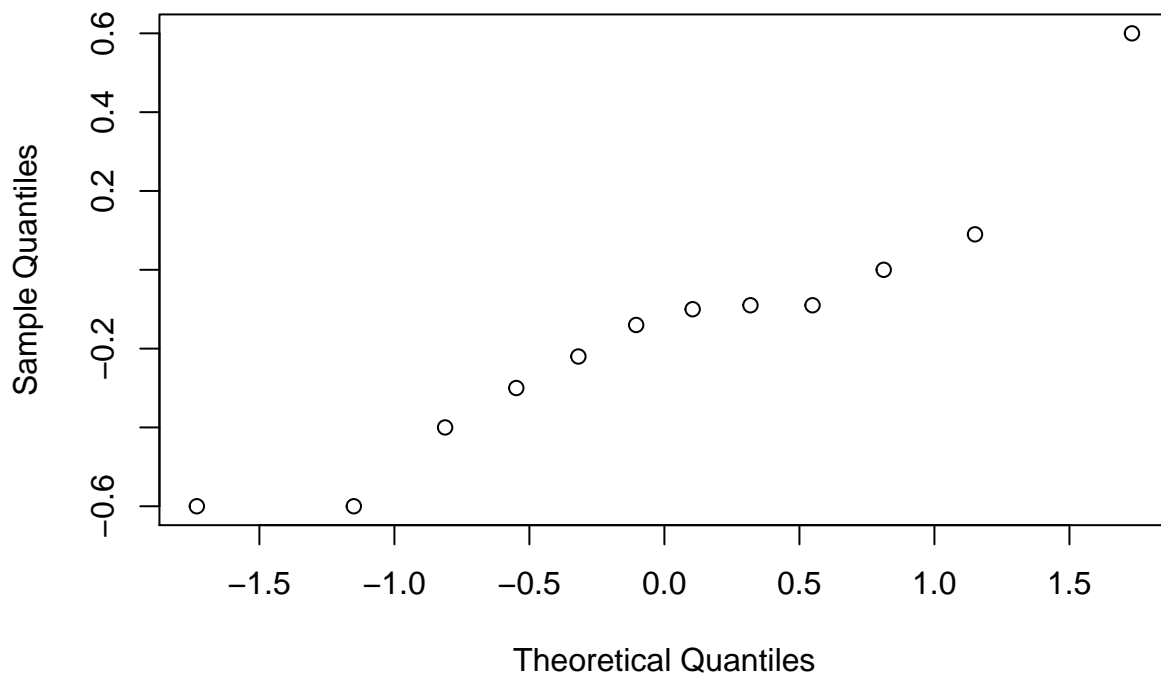


```
energy = run[run$drink=='energy', ]  
t.test(energy$before, energy$after, paired = TRUE)[[3]]
```

```
## [1] 0.1263962
```

```
qqnorm(energy$after - energy$before)
```


Normal Q-Q Plot



From the paired t-test's requirement, difference between run before and run after should be normally distributed, however from the QQ-plot it shows that neither soft drink' nor energy drink's differences are normally distributed. So we will try to use permutation tests, which normality of data differencee is not required.

```
mystat=function(x,y) {mean(x-y)}
B=1000
tstar=numeric(B)
for (i in 1:B)
{
  lemonstar=t(apply(cbind(lemo$before,lemo$after),1,sample))
  tstar[i]=mystat(lemonstar[,1],lemonstar[,2])
}
myt=mystat(lemo$before,lemo$after)
pl=sum(tstar<myt)/B
pr=sum(tstar>myt)/B
p=2*min(pl,pr)
p

## [1] 0.47

mystat=function(x,y) {mean(x-y)}
B=1000
tstar=numeric(B)
for (i in 1:B)
{
  lemonstar=t(apply(cbind(energy$before,energy$after),1,sample))
  tstar[i]=mystat(lemonstar[,1],lemonstar[,2])
}
myt=mystat(energy$before,energy$after)
pl=sum(tstar<myt)/B
pr=sum(tstar>myt)/B
```

```
p=2*min(pl,pr)
p
```

```
## [1] 0.118
```

Both two permutation tests results (softdrink: 0.434 and energy: 0.136) are bigger than 0.05, which means there are no big speed difference before drink and after, no matter we test on which drink.

A more interesting outcome is t.test of softdrink's p-value = 0.4373423 > 0.05 and t.test of energy's p-value = 0.1263962 > 0.05, which means t-test results also show no big speed difference before drink and after, both in softdrink condition and energy drink condition.

c) We chose permutation test and $T_i^* = \text{mean}(X^* - Y^*)$ to test whether these time differences are effected by the type of drink.

```
lemo$difference = lemo$before - lemo$after
energy$difference = energy$before - energy$after
mystat=function(x,y) {mean(x-y)}
B=1000
tstar=numeric(B)
for (i in 1:B) {
  adiffstar=t(apply(cbind(lemo$difference,energy$difference),1,sample))
  tstar[i]=mystat(adiffstar[,1],adiffstar[,2])
}
myt=mystat(lemo$difference,energy$difference)
pl=sum(tstar<myt)/B
pr=sum(tstar>myt)/B
p=2*min(pl,pr)
p
```

```
## [1] 0.18
```

p-value = 0.188 > 0.05, hence there is no difference between the two types of drinks.

d) Whether drinking the energy drink speeds up the running have another important attribute, that is the test time after drinking. Maybe energy drink will have great influence just in a very short time, maybe 5 min, after it is drunk. So more test cases on test time are needed. This is the similar objection to the experiment design in c).

Exercise 5

a)

```
meatmeal = chickwts[chickwts$feed == 'meatmeal', ]$weight
sunflower = chickwts[chickwts$feed == 'sunflower', ]$weight
t.test(meatmeal, sunflower)[[3]]
```

```
## [1] 0.04441462
```

```
wilcox.test(meatmeal, sunflower)[[3]]
```

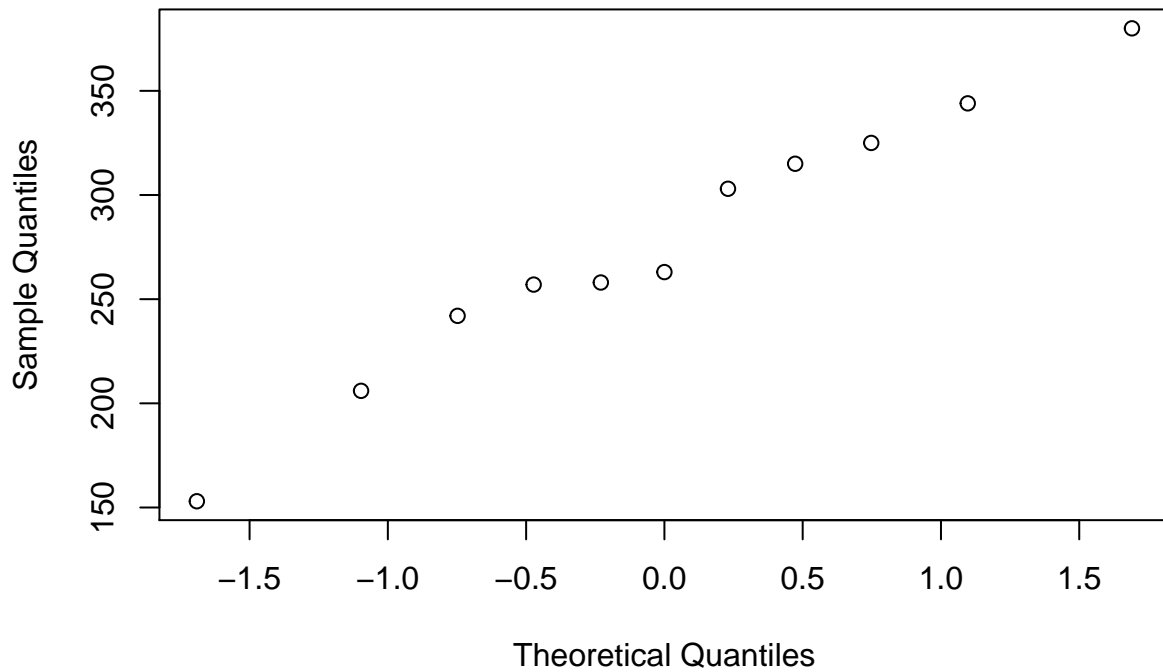
```
## [1] 0.06881704
```

```
ks.test(meatmeal, sunflower)[[2]]
```

```
## [1] 0.108496
```

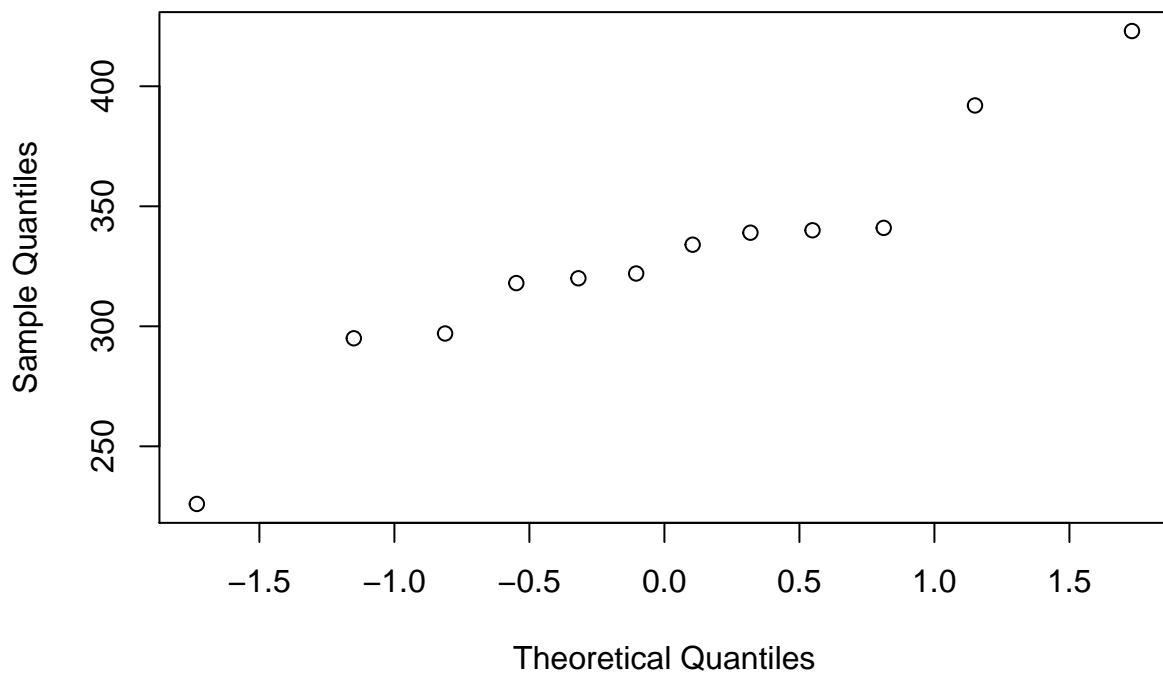
```
qqnorm(meatmeal)
```

Normal Q-Q Plot



```
qqnorm(sunflower)
```

Normal Q-Q Plot



For paired t-test argues that the two outcomes are measured on the same experimental unit, but it is not in this case, so the data is not paired.

After we ran the tests, we got t-test result: 0.04441462, Mann-Whitney test result: 0.06881704, Kolmogorov-

Smirnov test result: 0.108496.

From, t-test result, we can assume that meatmeal condition and sunflower condition has great difference of mean weight, but this assumption is not correct. This is because from QQ-plot of sunflower we can find that the data is not normally distributed.

Both Mann-Whitney test and Kolmogorov-Smirnov test show that there are no big difference between meatmeal condition and sunflower condition.

But why MW got bigger p-value than KS's p-value? The KS test is sensitive to any differences in the two distributions. Substantial differences in shape, spread or median will result in a small P value. In contrast, the MW test is mostly sensitive to changes in the median. This means in this question we'd better take MW's p-value.

b)

```
chickwtsaov=lm(weight~feed,data=chickwts)
anova(chickwtsaov)[[5]][1]

## [1] 5.93642e-10

summary(chickwtsaov)

##
## Call:
## lm(formula = weight ~ feed, data = chickwts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -123.909  -34.413    1.571   38.170  103.091
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    323.583     15.834   20.436 < 2e-16 ***
## feedhorsebean -163.383     23.485   -6.957 2.07e-09 ***
## feedlinseed   -104.833     22.393   -4.682 1.49e-05 ***
## feedmeatmeal   -46.674     22.896   -2.039 0.045567 *
## feedsoybean    -77.155     21.578   -3.576 0.000665 ***
## feedsunflower    5.333     22.393    0.238 0.812495
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.85 on 65 degrees of freedom
## Multiple R-squared:  0.5417, Adjusted R-squared:  0.5064
## F-statistic: 15.36 on 5 and 65 DF,  p-value: 5.936e-10
```

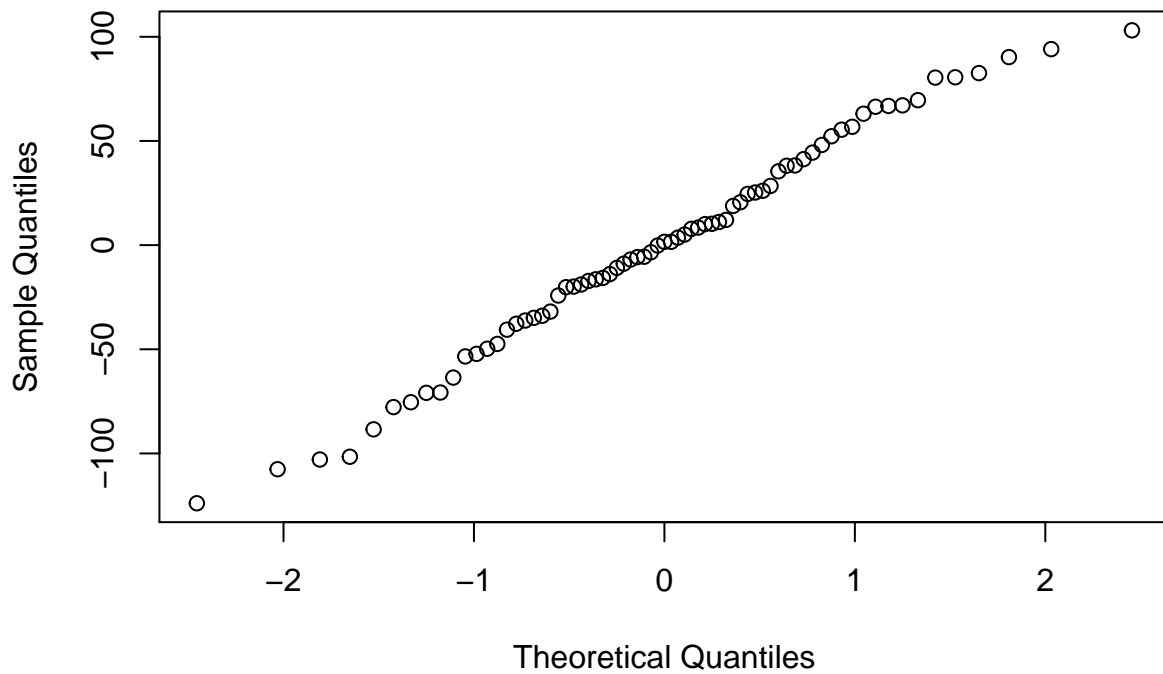
After ran the script, we got one-way ANOVA's p-value=5.93642e-10 < 0.05, so we can reach a conclusion that there do have difference between each group.

The summary of ANOVA shows estimated chick weights for each feed supplements are 323.583(casein), -163.383+323.583=160.2(horsebean), 218.75(linseed), 276.909(meatmeal), 246.428(soybean) and 328.916(sunflower).

In conclusion sunflower is the best feed supplement.

```
par(mfrow=c(1,1)); qqnorm(residuals(chickwtsaov))
```

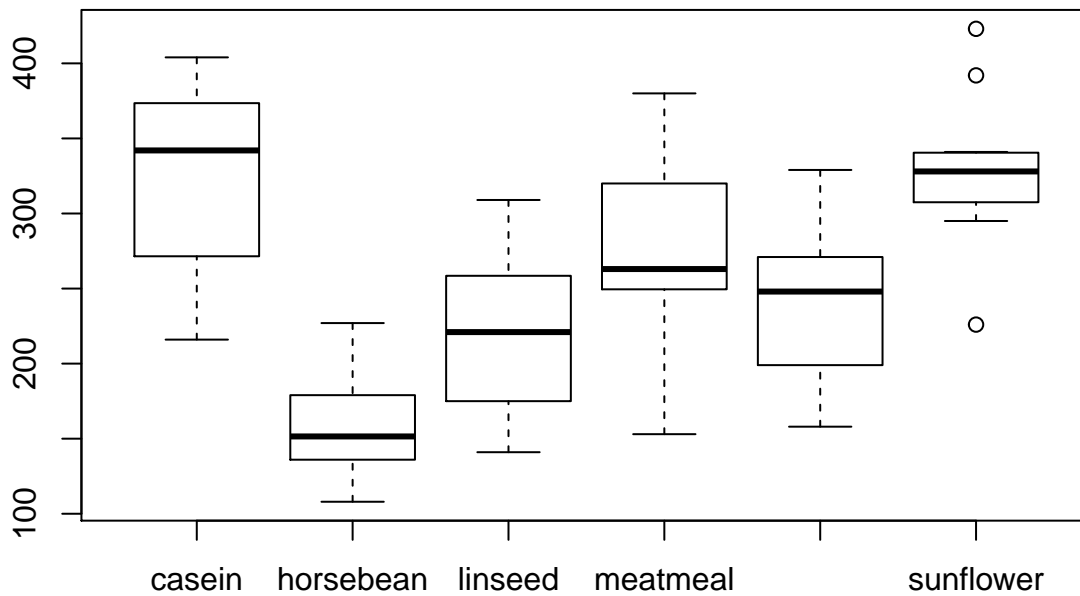
Normal Q-Q Plot



nally we need to check the assumption of normality of the populations. We used residuals data to draw QQ-plots, and all data are in a line, which is normality.

c)

```
X <- split(chickwts$weight, chickwts$feed)
boxplot(X)
```



```
stripchart(X, vertical=TRUE)
```



boxplot and strip to check the ANOVA model assumptions. In boxplot graph both casein and sunflower condition got the highest median. From the strip chart we found that casein condition distributed more widely, in contrast sunflower condition's data is more concentrated. These mean out conclusion in b) is correct.

d)

```
attach(chickwts)
kruskal.test(weight,feed)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: weight and feed
## Kruskal-Wallis chi-squared = 37.343, df = 5, p-value = 5.113e-07
```

The Kruskal-Wallis test p-value is $5.113e-07 < 0.05$, so we can reach a conclusion that there do have difference between each group. Furthermore, this conclusion is the same as b).