Excercise 1

DATA VISUALIZATION

```r
set.seed(1234)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## v purrr   0.3.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

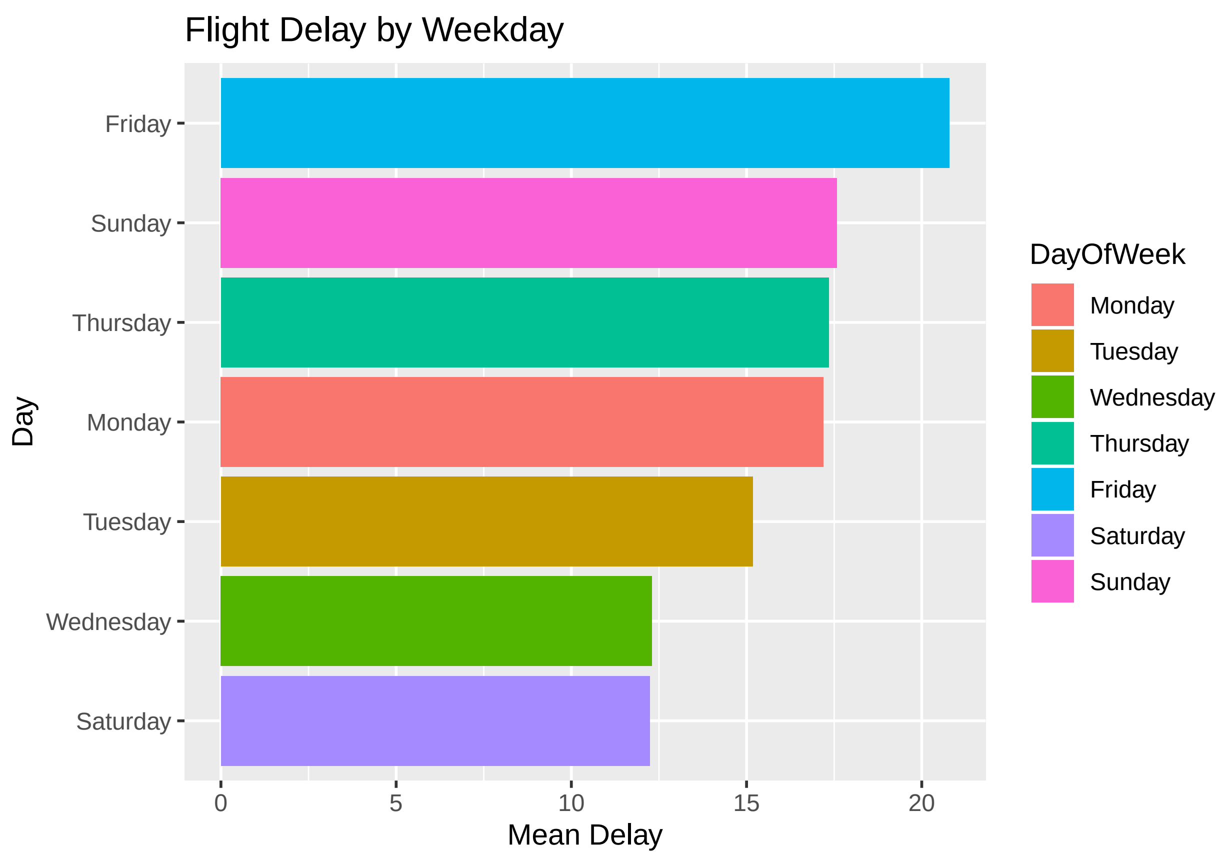
```r
data <- read.csv("~/Desktop/SDS323-master/data/ABIA.csv")

# clean the data
data$Year <- NULL
data$Month <- as.factor(data$Month)
data$DayofMonth <- as.factor(data$DayofMonth)
data$DayOfWeek <- as.factor(data$DayOfWeek)
data$Cancelled <- as.factor(data$Cancelled)
data$Diverted <- as.factor(data$Diverted)
data$Delay <- data$DepDelay + data$ArrDelay
summary(data)
```

```
##      Month         DayofMonth    DayOfWeek    DepTime       CRSDepTime
##  6      : 9090   18     : 3346   1:14798   Min.   :   1   Min.   :  55
##  5      : 9021   21     : 3336   2:14803   1st Qu.: 917   1st Qu.: 915
##  7      : 8931   11     : 3334   3:14841   Median :1329   Median :1320
##  3      : 8921   14     : 3333   4:14774   Mean   :1329   Mean   :1320
##  1      : 8726   10     : 3318   5:14768   3rd Qu.:1728   3rd Qu.:1720
##  8      : 8553   7      : 3315   6:11454   Max.   :2400   Max.   :2346
##  (Other):46018   (Other):79278   7:13822   NA's   :1413
##     ArrTime       CRSArrTime   UniqueCarrier     FlightNum       TailNum
##  Min.   :   1   Min.   :   5   WN     :34876   Min.   :   1          : 1104
##  1st Qu.:1107   1st Qu.:1115   AA     :19995   1st Qu.: 640   N678CA :  195
##  Median :1531   Median :1535   CO     : 9230   Median :1465   N511SW :  180
##  Mean   :1487   Mean   :1505   YV     : 4994   Mean   :1917   N526SW :  176
##  3rd Qu.:1903   3rd Qu.:1902   B6     : 4798   3rd Qu.:2653   N528SW :  172
##  Max.   :2400   Max.   :2400   XE     : 4618   Max.   :9741   N520SW :  168
##  NA's   :1567                  (Other):20749                  (Other):97265
##  ActualElapsedTime CRSElapsedTime     AirTime          ArrDelay
##  Min.   : 22.0     Min.   : 17.0   Min.   :  3.00   Min.   :-129.000
##  1st Qu.: 57.0     1st Qu.: 58.0   1st Qu.: 38.00   1st Qu.:  -9.000
##  Median :125.0     Median :130.0   Median :105.00   Median :  -2.000
```

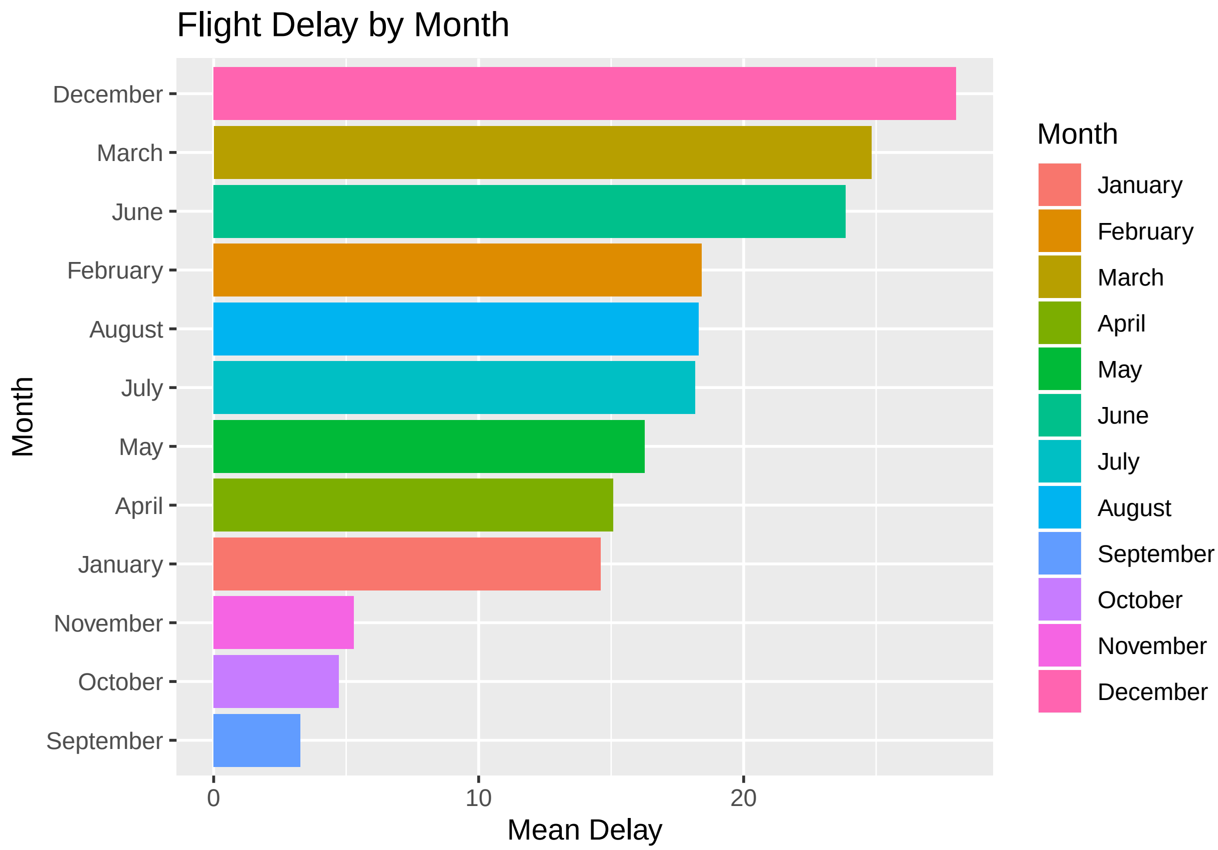
```
##  Mean   :120.2     Mean   :122.1   Mean   : 99.81   Mean   :   7.065
##  3rd Qu.:164.0     3rd Qu.:165.0   3rd Qu.:142.00   3rd Qu.:  10.000
##  Max.   :506.0     Max.   :320.0   Max.   :402.00   Max.   : 948.000
##  NA's   :1601      NA's   :11      NA's   :1601     NA's   :1601
##     DepDelay           Origin           Dest          Distance
##  Min.   :-42.000   AUS    :49623   AUS    :49637   Min.   :  66
##  1st Qu.: -4.000   DAL    : 5583   DAL    : 5573   1st Qu.: 190
##  Median :  0.000   DFW    : 5508   DFW    : 5506   Median : 775
##  Mean   :  9.171   IAH    : 3704   IAH    : 3691   Mean   : 705
##  3rd Qu.:  8.000   PHX    : 2786   PHX    : 2783   3rd Qu.:1085
##  Max.   :875.000   DEN    : 2719   DEN    : 2673   Max.   :1770
##  NA's   :1413      (Other):29337   (Other):29397
##      TaxiIn            TaxiOut        Cancelled CancellationCode Diverted
##  Min.   :  0.000   Min.   :  1.00   0:97840    :97840          0:99079
##  1st Qu.:  4.000   1st Qu.:  9.00   1: 1420   A:  719          1:  181
##  Median :  5.000   Median : 12.00             B:  605
##  Mean   :  6.413   Mean   : 13.96             C:   96
##  3rd Qu.:  7.000   3rd Qu.: 16.00
##  Max.   :143.000   Max.   :305.00
##  NA's   :1567      NA's   :1419
##   CarrierDelay     WeatherDelay       NASDelay      SecurityDelay
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00
##  Median :  0.00   Median :  0.00   Median :  2.00   Median :  0.00
##  Mean   : 15.39   Mean   :  2.24   Mean   : 12.47   Mean   :  0.07
##  3rd Qu.: 16.00   3rd Qu.:  0.00   3rd Qu.: 16.00   3rd Qu.:  0.00
##  Max.   :875.00   Max.   :412.00   Max.   :367.00   Max.   :199.00
##  NA's   :79513    NA's   :79513    NA's   :79513    NA's   :79513
##  LateAircraftDelay     Delay
##  Min.   :  0.00    Min.   :-139.0
##  1st Qu.:  0.00    1st Qu.: -12.0
##  Median :  6.00    Median :  -2.0
##  Mean   : 22.97    Mean   :  16.2
##  3rd Qu.: 30.00    3rd Qu.:  16.0
##  Max.   :458.00    Max.   :1823.0
##  NA's   :79513     NA's   :1601
```

```r
# change the levels to make the data more readable
levels <- levels(data$DayOfWeek)
levels <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
levels(data$DayOfWeek) <- levels

levels <- levels(data$Month)
levels <- c("January", "February", "March", "April", "May", "June", "July", "August", "September", "Octo
levels(data$Month) <- levels
```

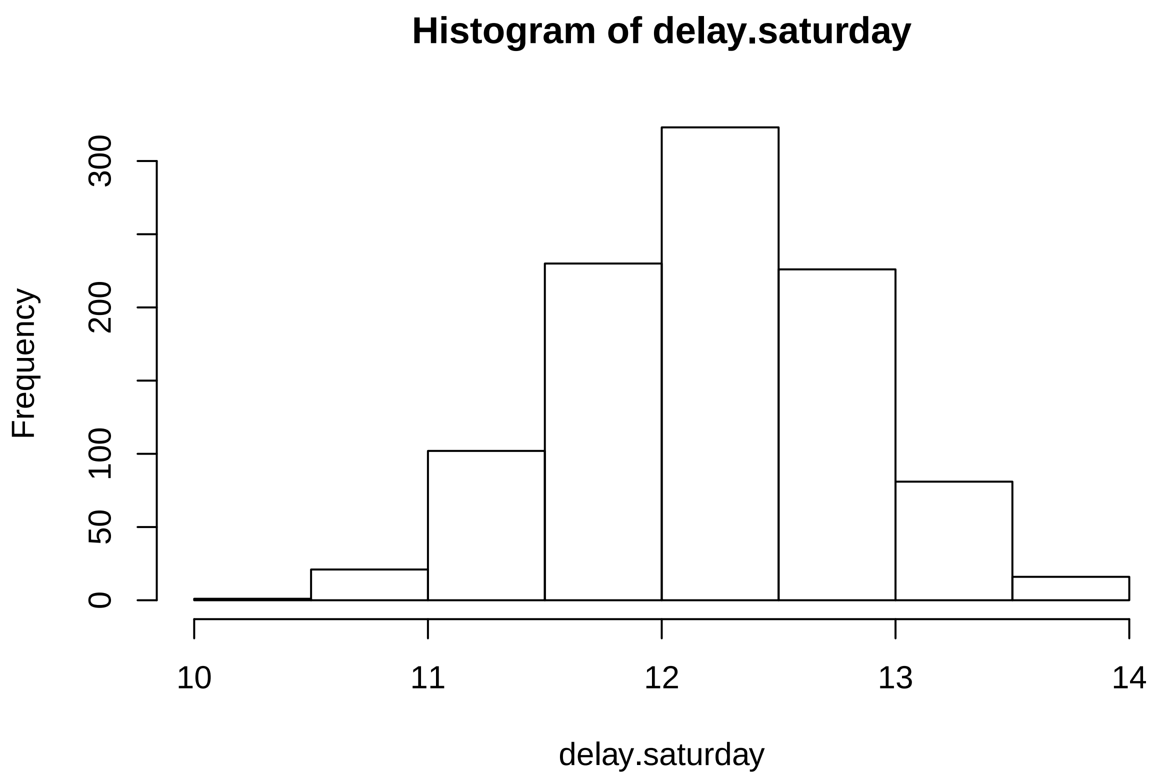
```r
set.seed(1234)

# group data by DayOfWeek
delay_summ <- data %>% group_by(DayOfWeek) %>% summarize(sum_delay.mean = mean(DepDelay + ArrDelay, na.r

ggplot(data = delay_summ, aes(x = reorder(DayOfWeek, sum_delay.mean), y = sum_delay.mean, fill=DayOfWeek
```



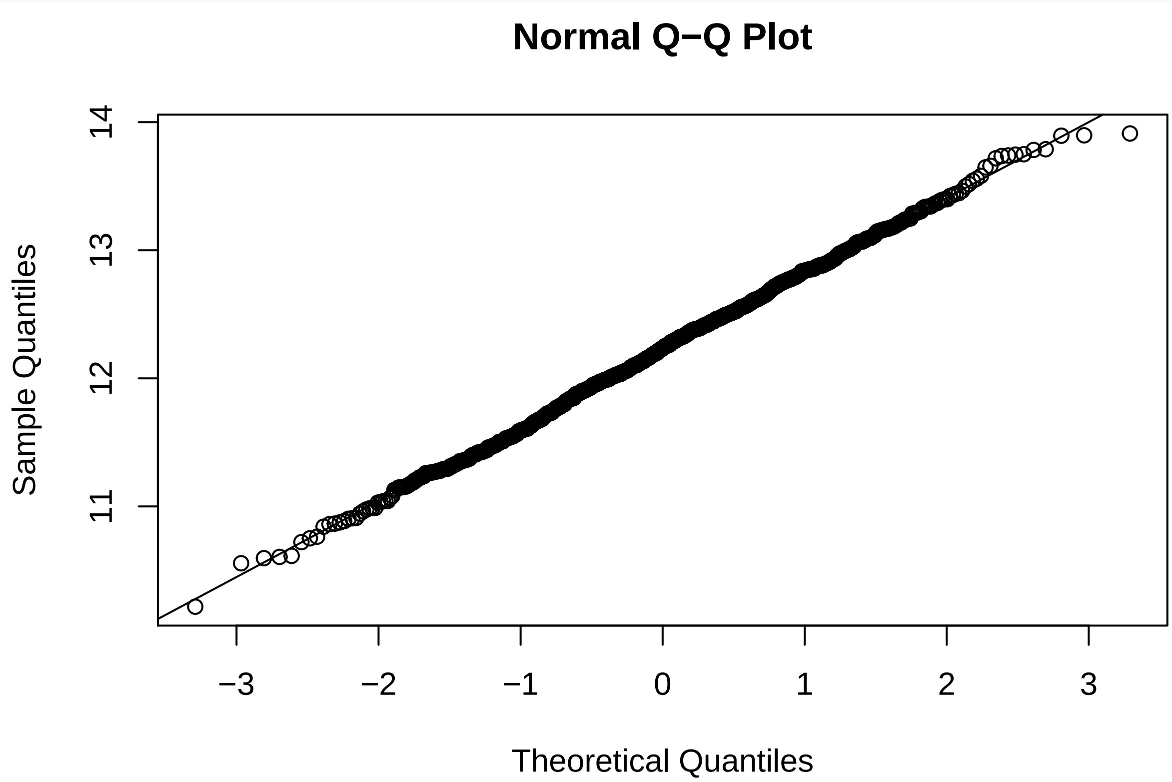

```
# group data by Month
month <- data %>% group_by(Month) %>% summarize(delay = mean(DepDelay + ArrDelay, na.rm = TRUE))

ggplot(data = month, aes(x = reorder(Month, delay), y = delay, fill=Month)) + geom_bar(stat = "identity
```



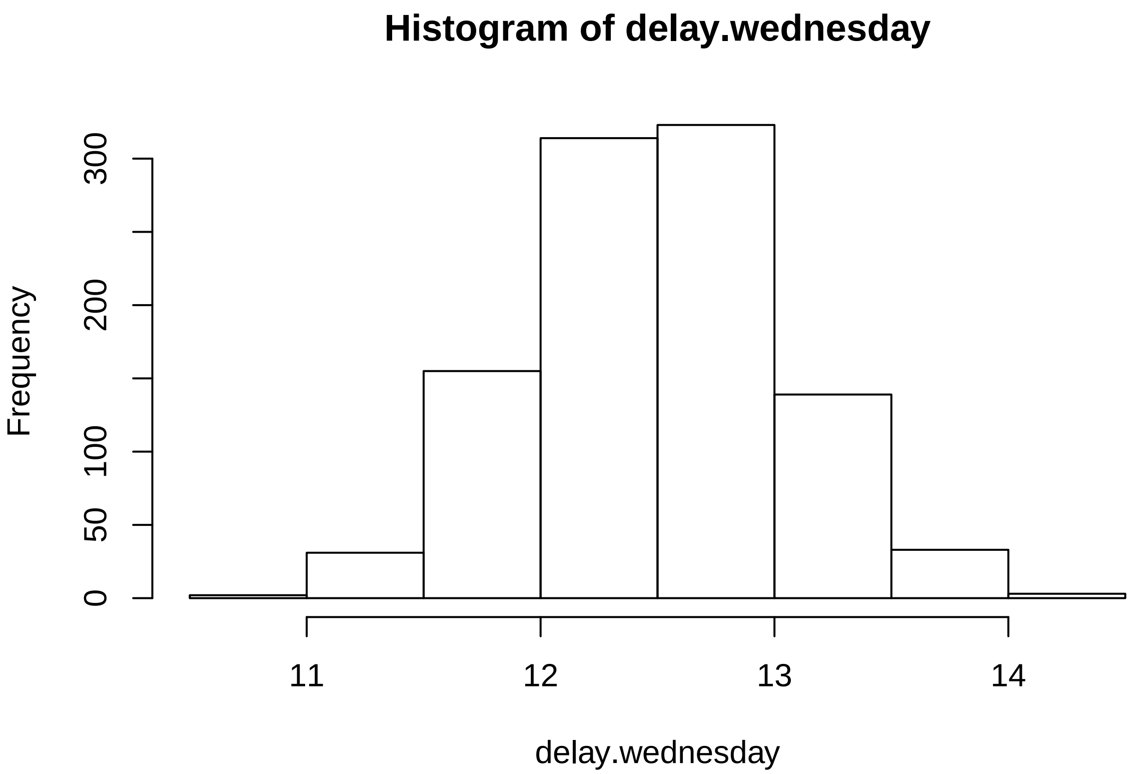

```r
saturday <- subset(data, data$DayOfWeek=="Saturday")
wednesday <- subset(data, data$DayOfWeek=="Wednesday")

# bootstrap for mean delay on saturday and wednesday
delay.saturday <- c()
delay.wednesday <- c()
for(i in 1:1000) {
  x <- saturday[sample(nrow(saturday), replace = TRUE),]
  delay.saturday[i] <- mean(x$DepDelay + x$ArrDelay, na.rm = TRUE)
  y <- wednesday[sample(nrow(saturday), replace = TRUE),]
  delay.wednesday[i] <- mean(y$DepDelay + y$ArrDelay, na.rm = TRUE)
}

hist(delay.saturday)
```



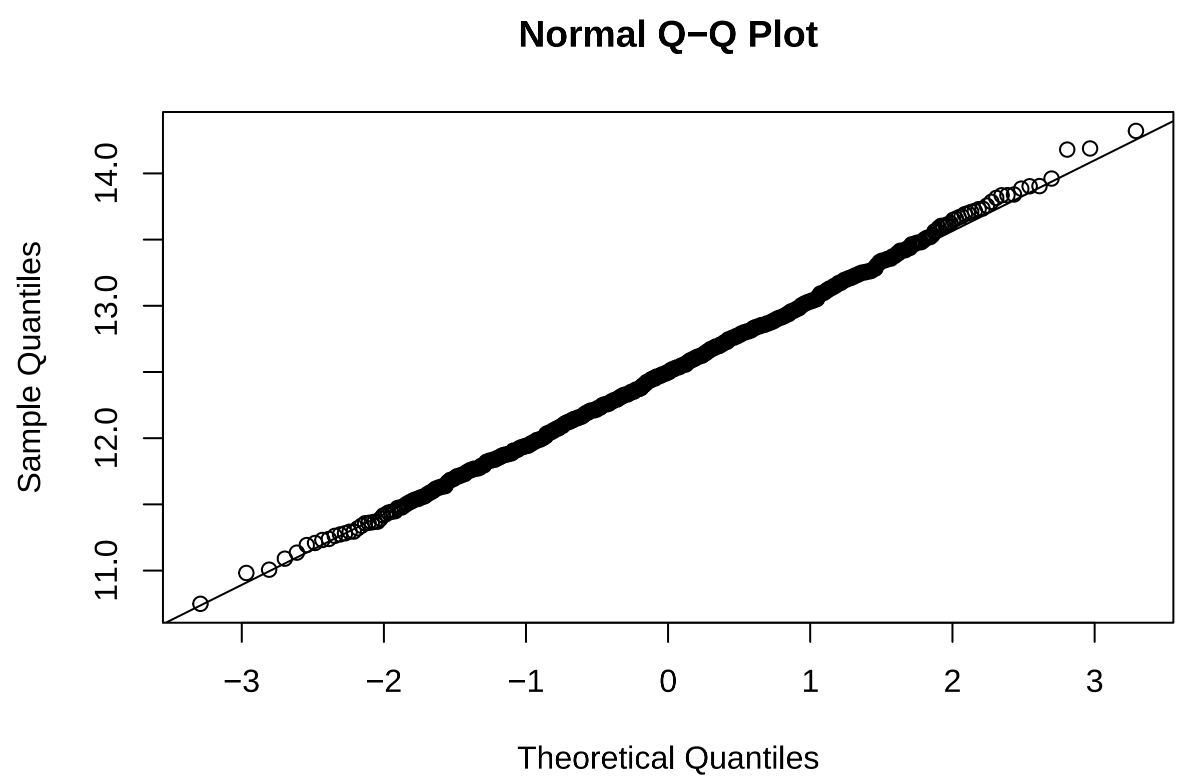

```r
qqnorm(delay.saturday)
qqline(delay.saturday)
```



```r
hist(delay.wednesday)
```




```r
qqnorm(delay.wednesday)
qqline(delay.wednesday)
```



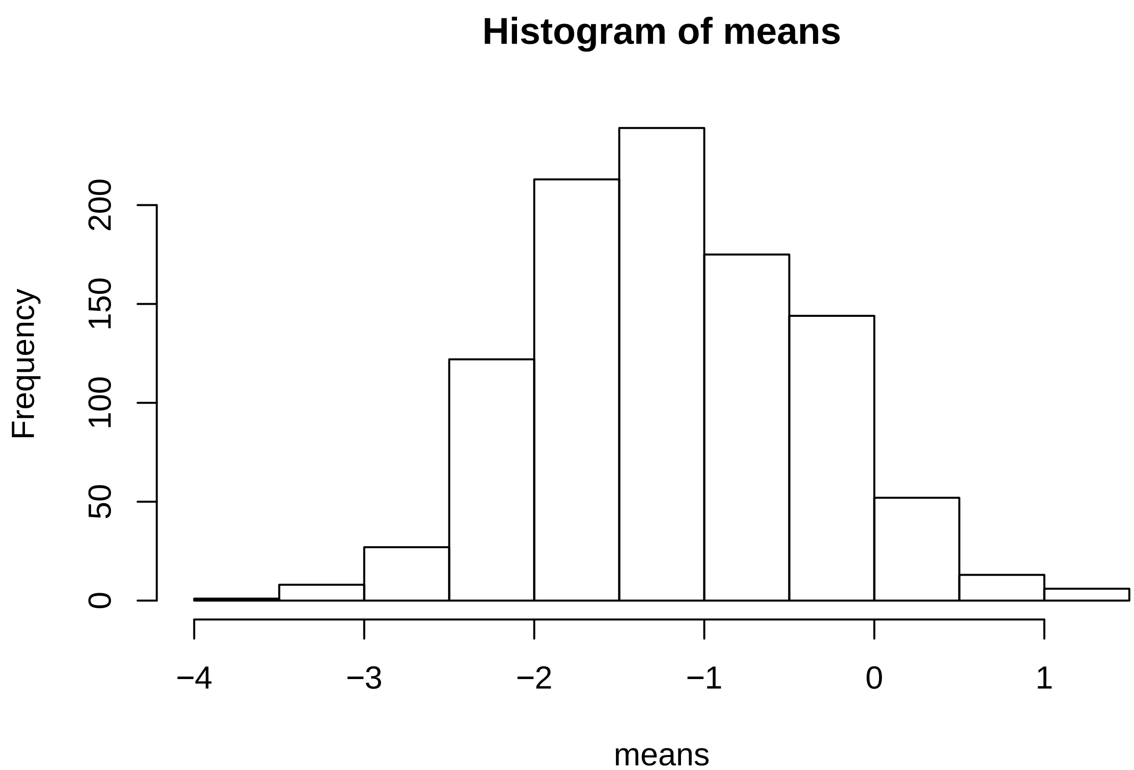
```
# the bootstrap distributions are approximately normal

# 95% confidence intervals
quantile(delay.saturday, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 11.04180 13.38497
```

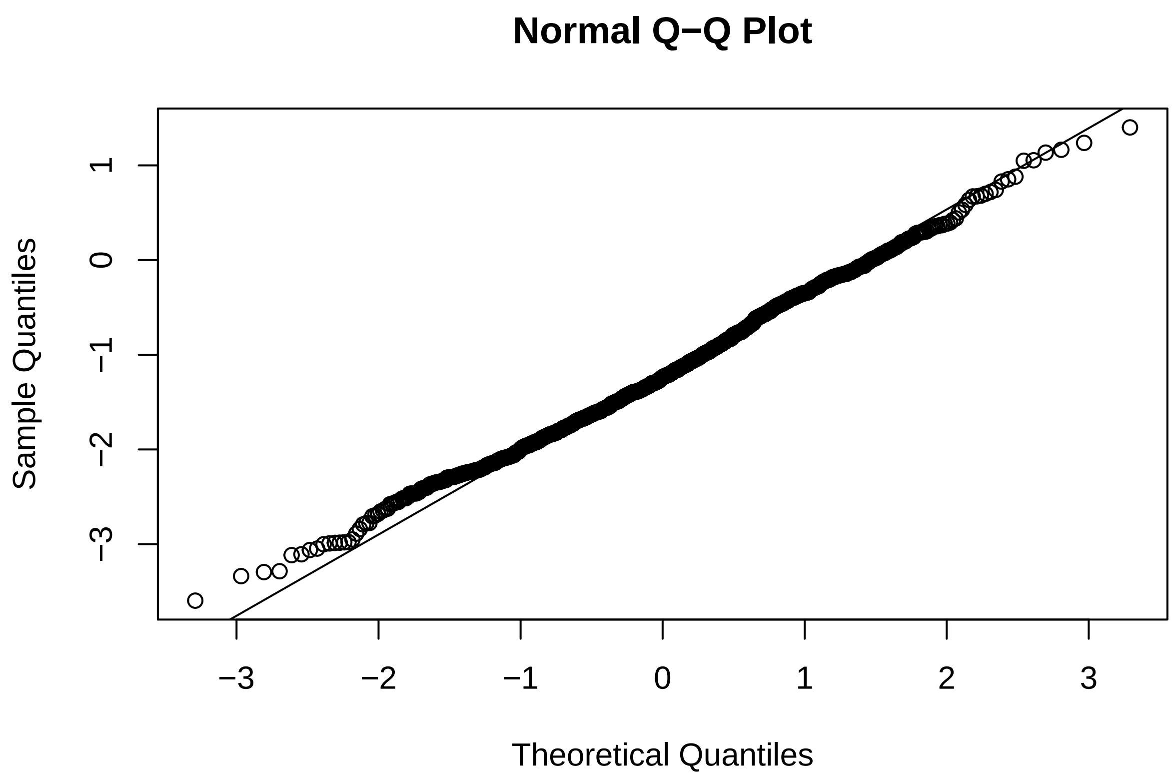
```
quantile(delay.wednesday, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 11.44269 13.60888
```

```
best <- subset(data, data$Month=="September" & (data$DayOfWeek=="Saturday" | data$DayOfWeek=="Wednesday

# bootstrap for "best" travel days
means <- replicate(1000, mean(sample(best$Delay, nrow(best), replace = TRUE), na.rm = TRUE))

hist(means)
```




```
qqnorm(means)
qqline(means)
```




```
# the bootstrap distribution is approximately normal

# 95% confidence interval
quantile(means, c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -2.6278043  0.3682786
```

I think the plots showcase my point without explanation, but I have also provided a more detailed explanation. To better analyze the data I defined delay as the sum of departure and arrival delay. When flying on a Saturday, we can say with 95% confidence that the average delay (a sum of departure and arrival delay) will be between 11.04 and 13.38 minutes. We know this because the bootstrap distribution is approximately normal. For wednesday, with 95% confidence the average delay will be between 11.44 and 13.61 minues. When analyzing delay by month, 3 distinct groups appear: September, October, and November easily have the shortest average delays; January, April, May, July, August, and February; and June, March, and December. June is the begining of summer and the end of the school year, March has spring break for UT Austin and other nearby universities (when many students will be flying in and out of AUS on the same day), and December is the worst travel month of the year because of Christmas and winter break. The best days to fly out of AUS are Wednesday and Saturday, and the best months are September, October, and November. If we had all of the freedom in the world to plan our flight we would choose to fly in and out of AUS on Wednesday and Saturday of September. Flying out on one of these ideal days, with 95% confidence we can expect our delay to be between -2.61 and 0.34 minutes. Meaning we will likely have no delays, and even more so our flights will be shorter than advertised!

REGRESSION PRACTICE

Import the raw data

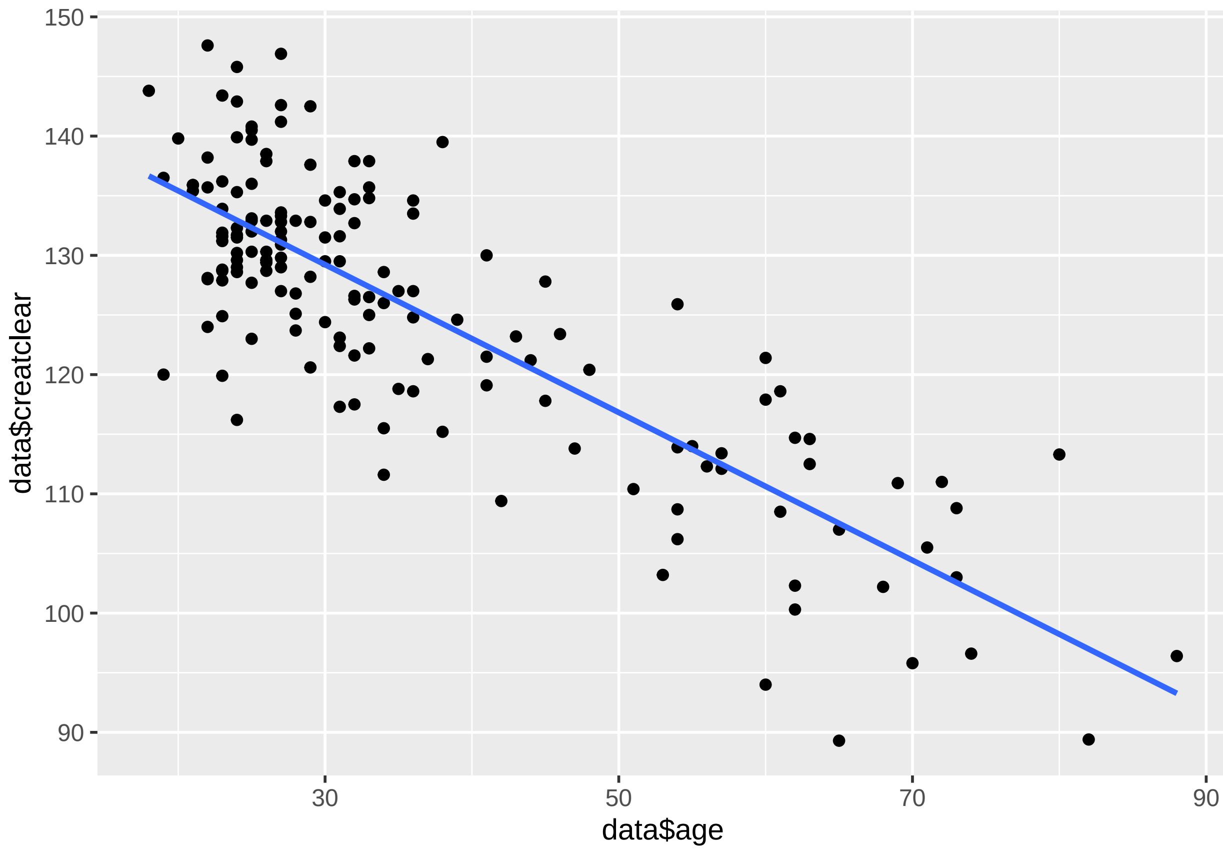
```
data <- read.csv("~/Desktop/SDS323-master/data/creatinine.csv")
summary(data)
```

```
##       age          creatclear
```

```
##  Min.   :18.00   Min.   : 89.3
##  1st Qu.:25.00   1st Qu.:118.6
##  Median :31.00   Median :128.0
##  Mean   :36.39   Mean   :125.3
##  3rd Qu.:43.00   3rd Qu.:133.3
##  Max.   :88.00   Max.   :147.6
```

Plot the data

```r
library(ggplot2)

linear <- lm(data = data, creatclear ~ age)
ggplot(data=data, aes(x=data$age, y = data$creatclear)) + geom_point() + geom_smooth(method = 'lm', se =
```




```r
summary(linear)
```

```
##
## Call:
## lm(formula = creatclear ~ age, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.2249  -4.6175   0.2221   4.7212  15.8221
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 147.81292    1.37965  107.14   <2e-16 ***
## age          -0.61982    0.03475  -17.84   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.911 on 155 degrees of freedom
## Multiple R-squared:  0.6724, Adjusted R-squared:  0.6703
## F-statistic: 318.2 on 1 and 155 DF,  p-value: < 2.2e-16
```

A linear regression seems reasonable for this data set given the shape and spread of the data. Here we will predict the creatine clearance rate for a 55 year old.

```
age <- c(40, 55, 60)
df <- data.frame(age)
pred <- predict(linear, df)
print(pred)
```

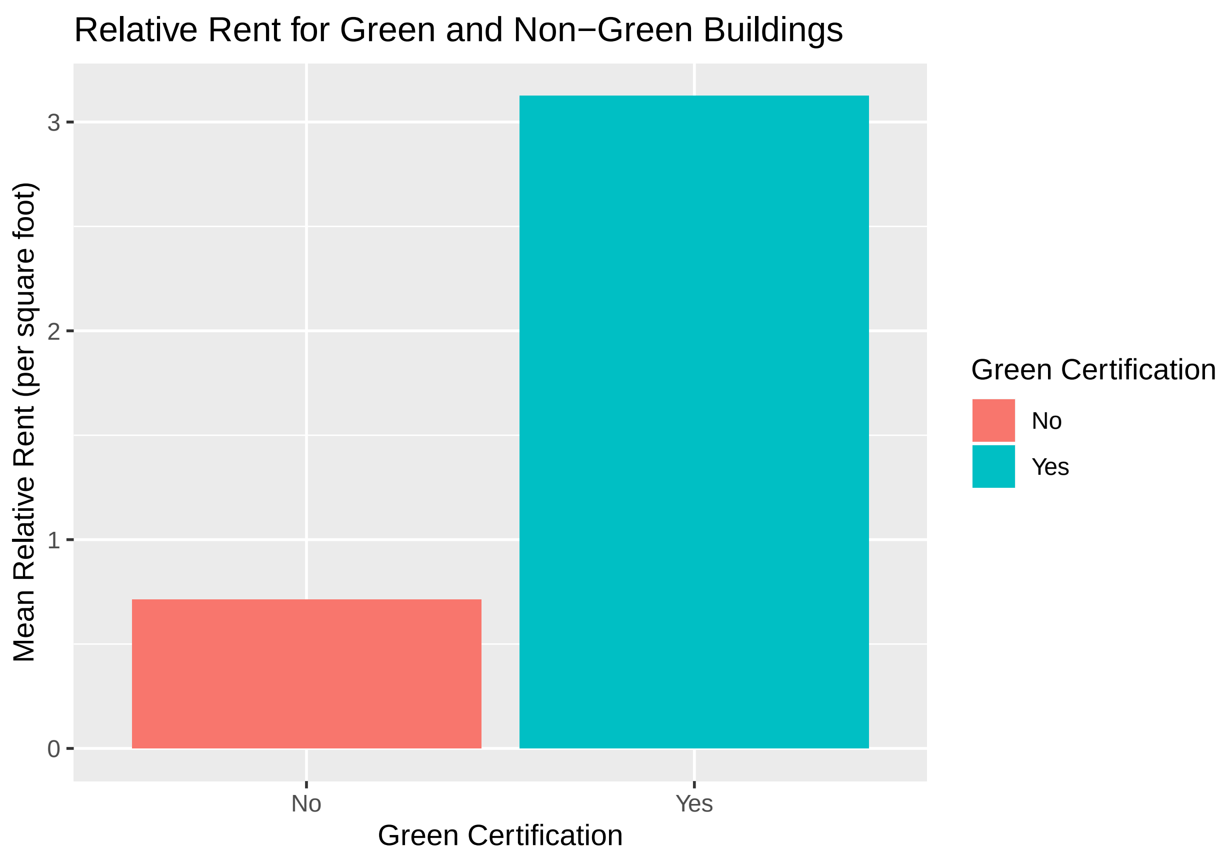
```
##        1        2        3
## 123.0203 113.7230 110.6240
```

We should expect a creatine clearance rate of 113.7 (the raw data is rounded to the 10th decimal place). Creatine clear rate changes by -0.6 ml/minute per year. The 40 year old with a creatine clearance rate of 135 is healthier than a 60 year olf with 112, because the average creatine clearance rates are 123.0 and 110.6 for 40 and 60 year olds respectively. Therefore the 40 year old is much healthier.

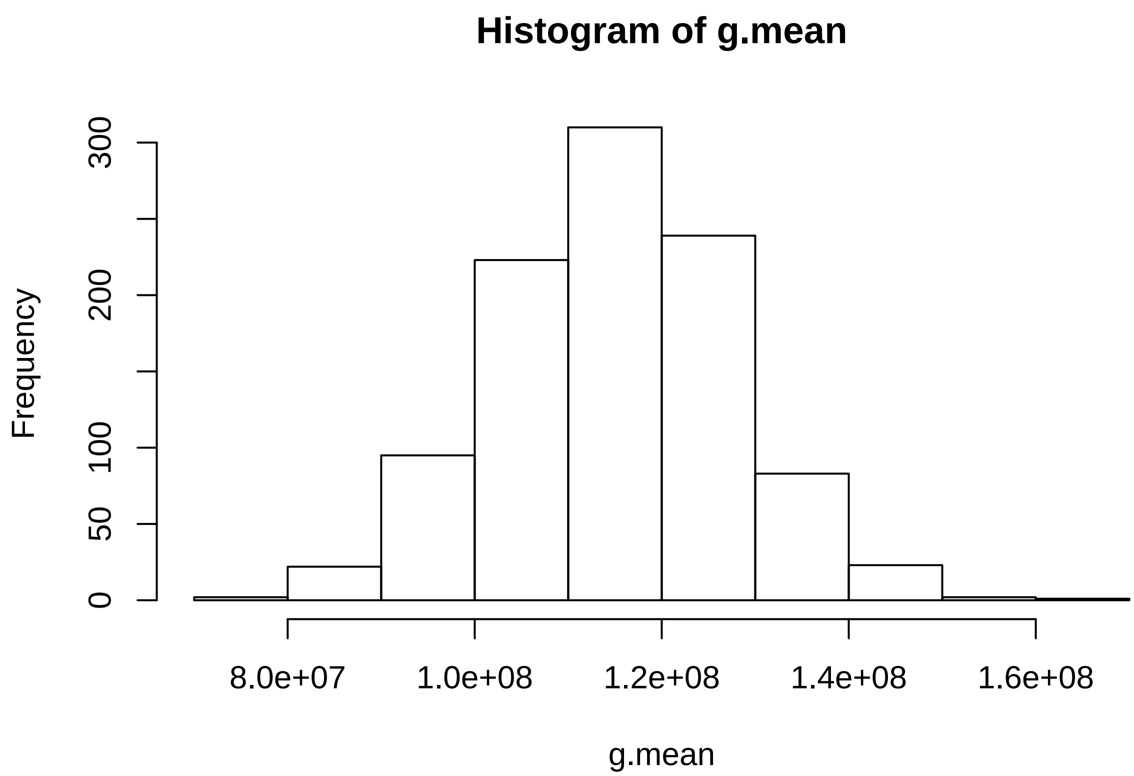
GREEN BUILDINGS

I disagree with the way the "excel guru" analyzed the data. Scrapping buildings with < 10% occupancy is a poor decision because green certification could have a large role in occupancy. For example, it's possible that because green buildings are more expensive to construct that they then must charge more in rent, leading to lower occupancy. This may not be a problem for small occupancy loses, but buildings with < 10% are likely losing a substantial abount of money, and are therefore important for our analysis. Subtracting median rent for green and non-green buildings is too simple and is likely ignoring compounding variables. For example, because green buildings tend to cost more to build, it is likely they are more common in wealthier areas where they can charge more rent to compensate for the building costs. Because this may not be true in other areas, it is important to compare green and non-green building rent within their clusters.

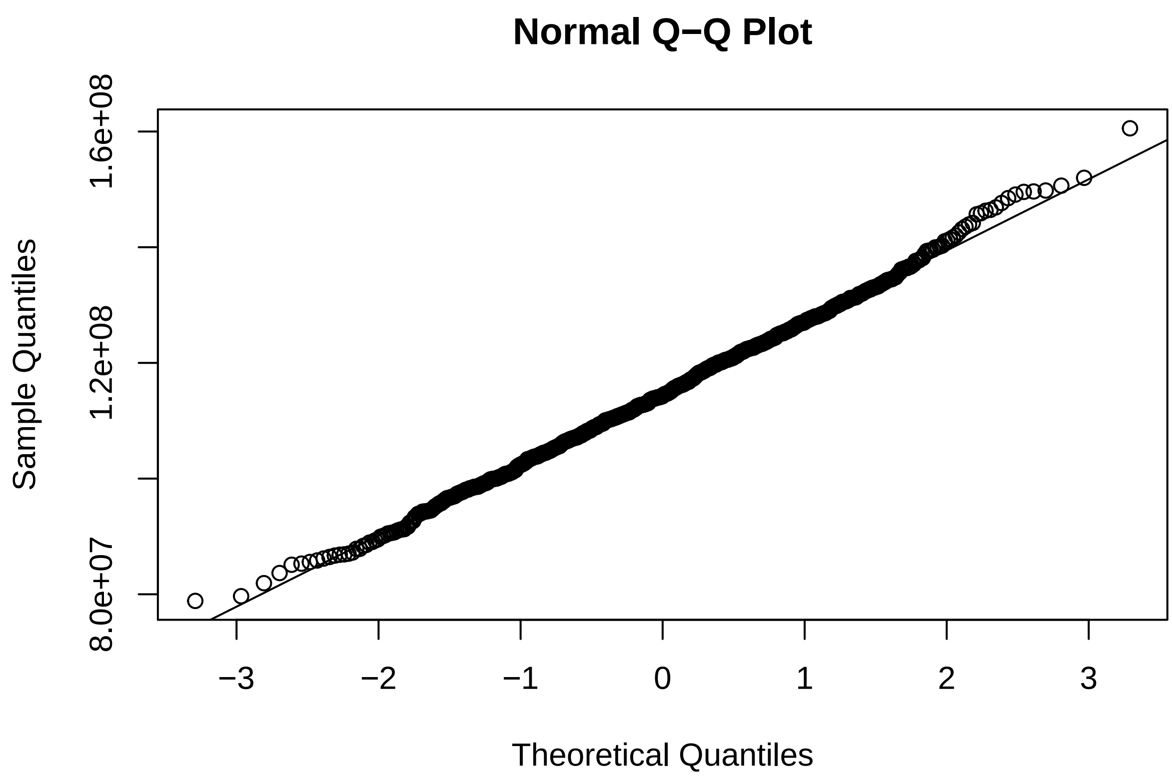
```
library(tidyverse)
library(knitr)

green <- read.csv("~/Documents/R/SDS 323/SDS323-master/data/greenbuildings.csv")

# create new variables which represent the setting better
green$RelativeRent <- green$Rent - green$cluster_rent
green$TotalRent <- green$Rent*green$size*green$leasing_rate
green$RelativeTotalRent <- green$RelativeRent*green$size*green$leasing_rate

# make green rating a factor
green$green_rating <- factor(green$green_rating)
levels <- levels(green$green_rating)
levels <- c("No", "Yes")
levels(green$green_rating) <- levels

# group data by green rating
d1 <- green %>% group_by(green_rating) %>% summarize(mean = mean(RelativeRent), sd = sd(RelativeRent))

# plot mean RelativeRent vs green_rating
ggplot(data = d1) + geom_bar(mapping = aes(x = green_rating, y = mean, fill = green_rating), stat = "id
```




```
# the plot seems to indicate that green buildings are more profitable, but these distributions have lar

# bootstrap
g <- subset(green, green$green_rating=="Yes")
g.mean <- c()
r <- subset(green, green$green_rating=="No")
r.mean <- c()
for(i in 1:1000) {
  x <- g[sample(nrow(g), replace = TRUE),]
  g.mean[i] <- mean(x$RelativeTotalRent)
  y <- r[sample(nrow(r), replace = TRUE),]
  r.mean[i] <- mean(y$RelativeTotalRent)
}

hist(g.mean)
```



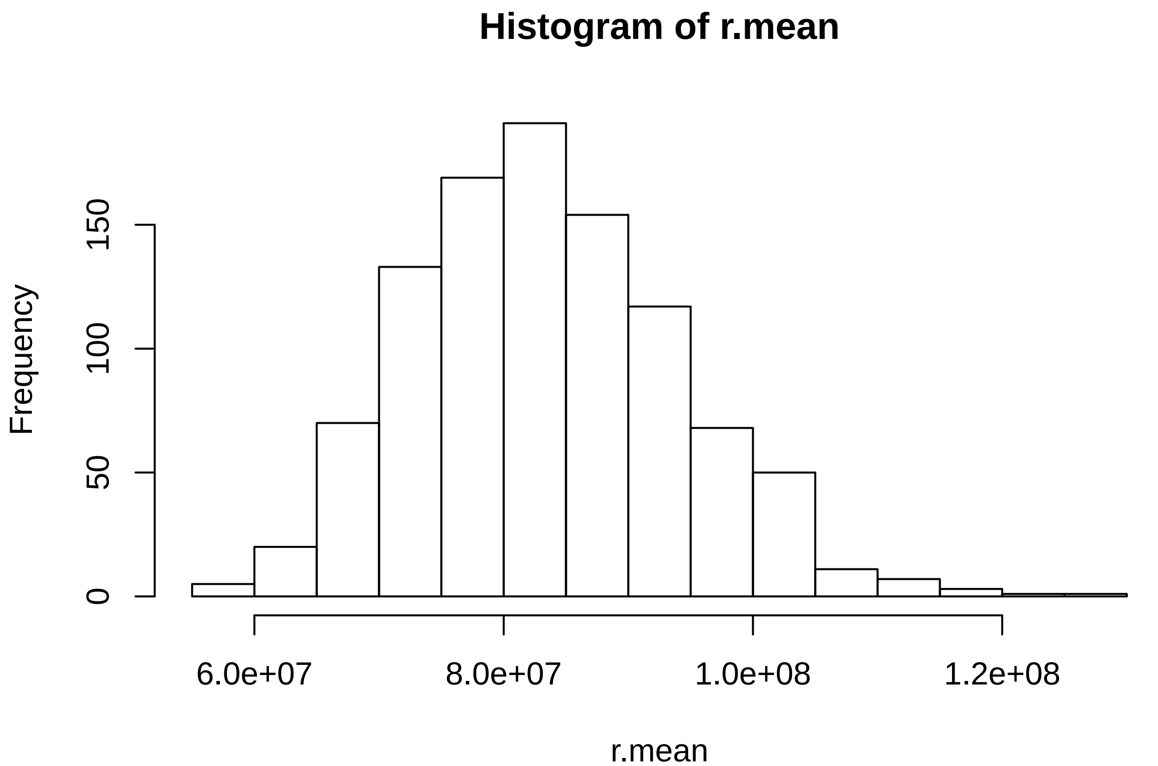

```r
qqnorm(g.mean)
qqline(g.mean)
```



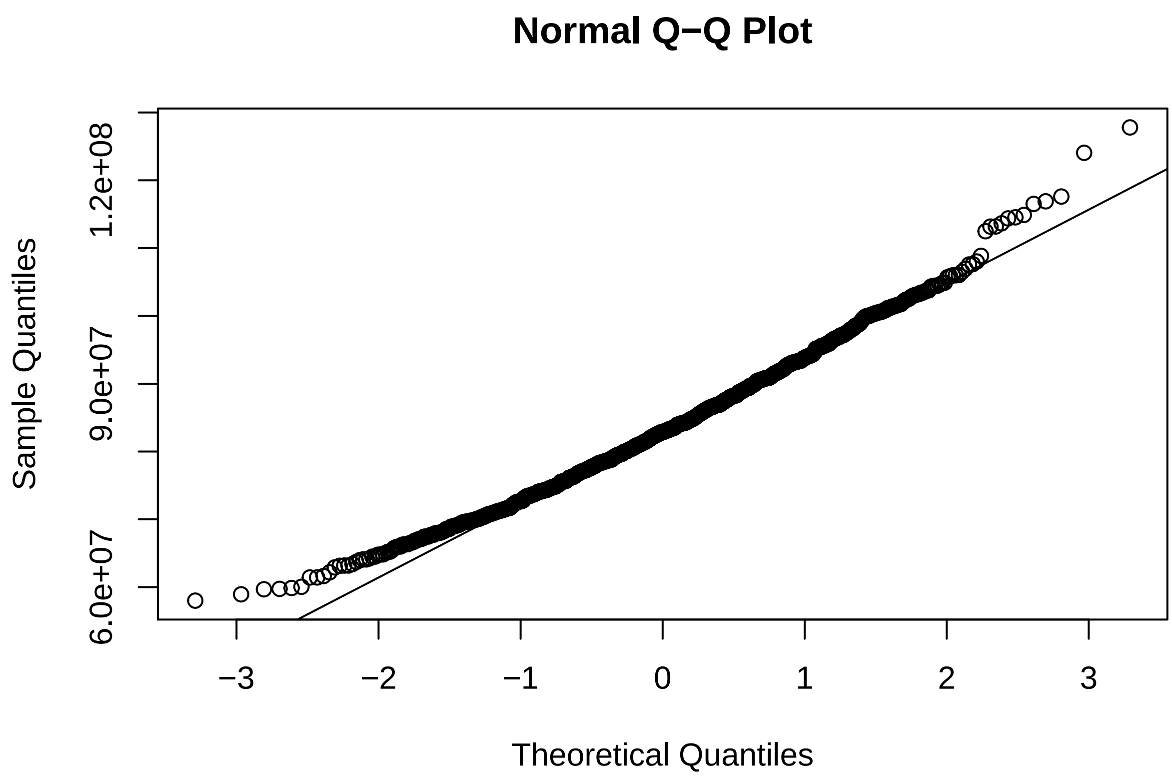
```r
quantile(g.mean, c(0.025, 0.975))
```

```
##      2.5%     97.5%
##  90237916 140131392
```

```r
hist(r.mean)
```




```r
qqnorm(r.mean)
qqline(r.mean)
```




```r
quantile(r.mean, c(0.025, 0.975))
```

```
##      2.5%     97.5%
##  65025117 104637581
```

```r
# both of the bootstrap distributions are approximately normal

sd(g.mean)
```

```
## [1] 12519631
```

```r
sd(r.mean)
```

```
## [1] 10566880
```

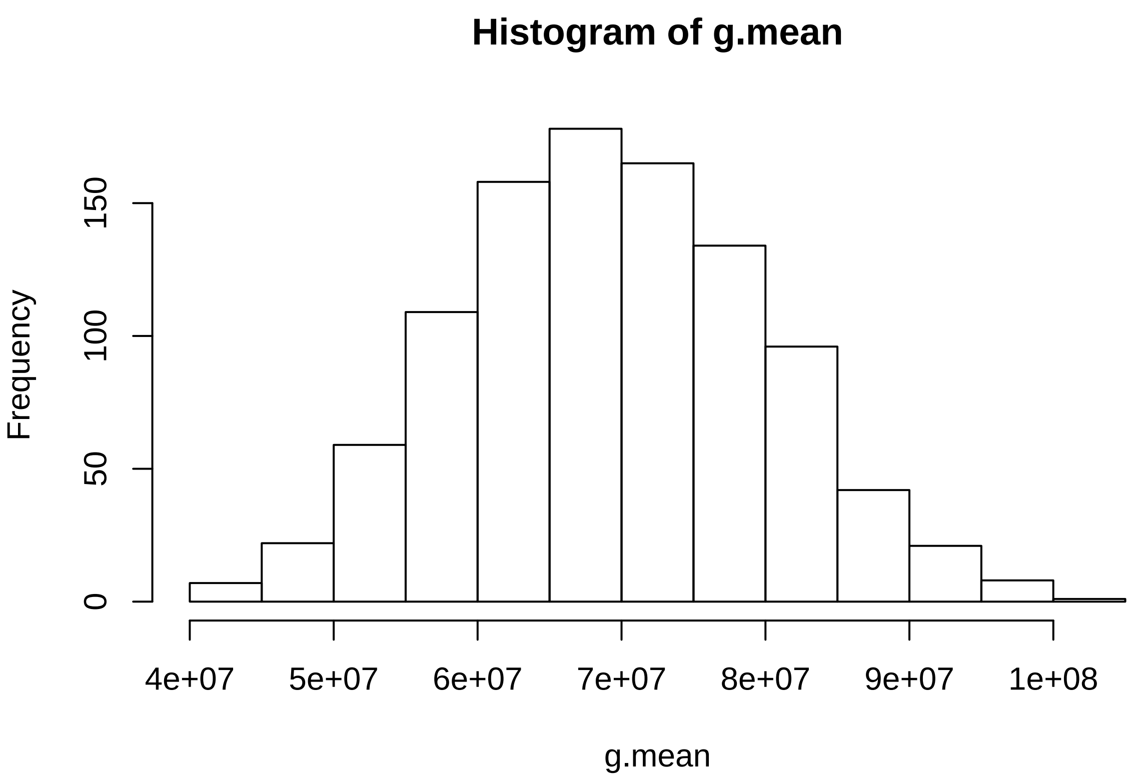
```r
# the two distributions have different sd => populations have different variances

# compute two sided t-test for the bootstrap distributions with different variances
t.test(x = g.mean, y = r.mean, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  g.mean and r.mean
## t = 60.598, df = 1943.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  30378383 32410460
## sample estimates:
## mean of x mean of y
## 114827887  83433466
```

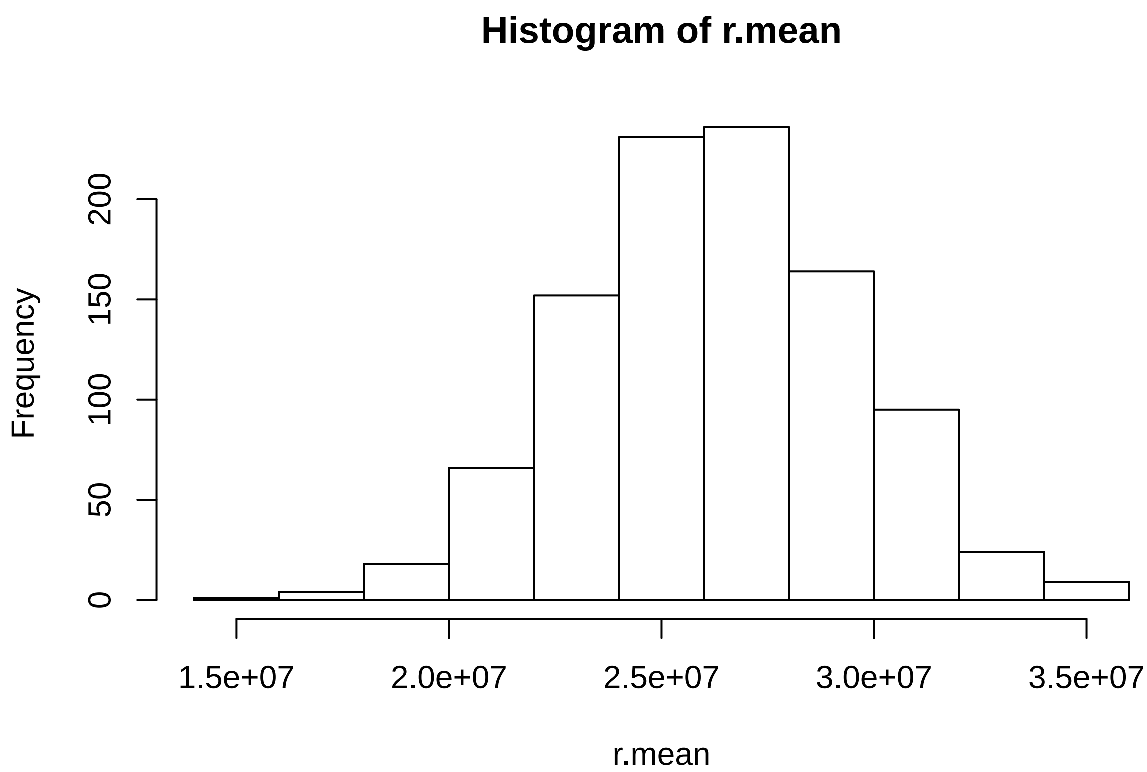
With 95% confidence we can say there is a difference in means between RelativeTotalRent for green and non-green buildings, with the means being 114,720,650 and 84,097,855 respectively. The 95% CI for the difference in means is (29,633,275, 31,612,314).

In order to relate this result to our problem, I will investigate if this relation follows for buildings more like the one we want to build. Specifically, buildings that have between 10 and 20 stories.

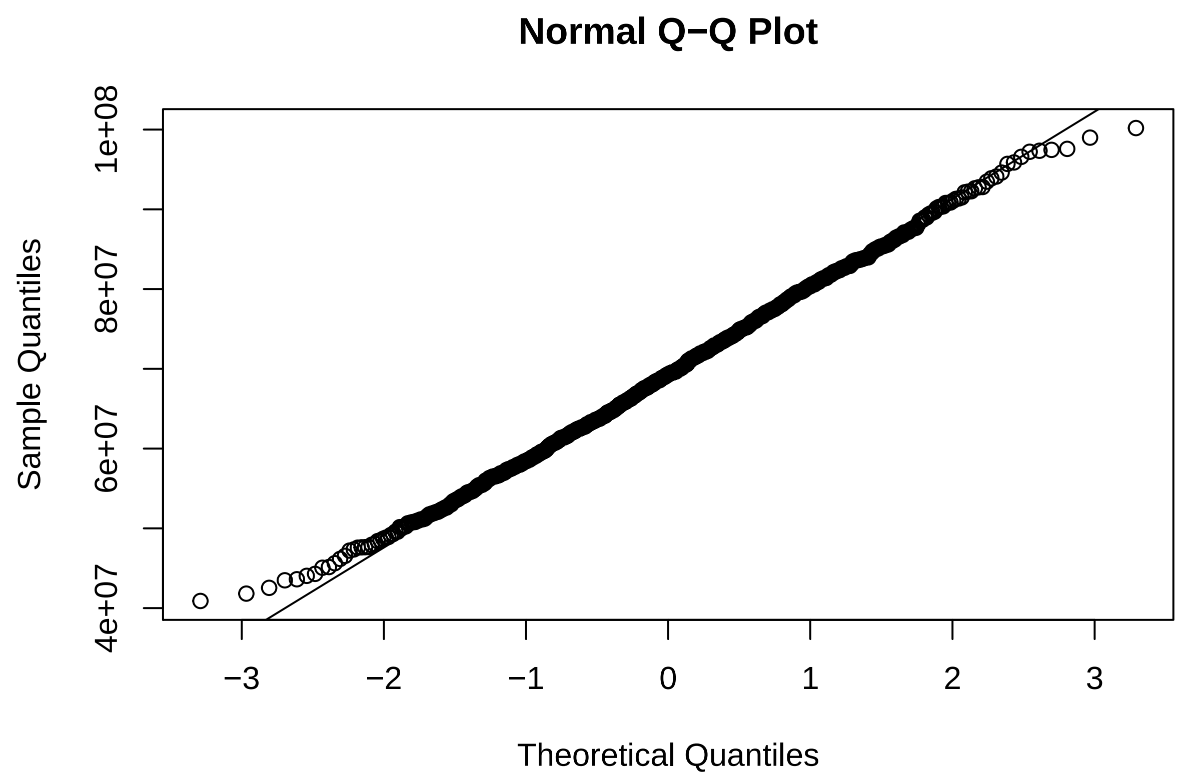
```
# bootstrap with different subset of buildings
tall <- subset(green, subset = green$stories >= 10 & green$stories < 20)
g <- subset(tall, tall$green_rating=="Yes")
r <- subset(tall, tall$green_rating=="No")
g.mean <- c()
r.mean <- c()
for(i in 1:1000) {
  x <- g[sample(nrow(g), replace = TRUE),]
  y <- r[sample(nrow(r), replace = TRUE),]
  g.mean[i] <- mean(x$RelativeTotalRent)
  r.mean[i] <- mean(y$RelativeTotalRent)
}

hist(g.mean)
```



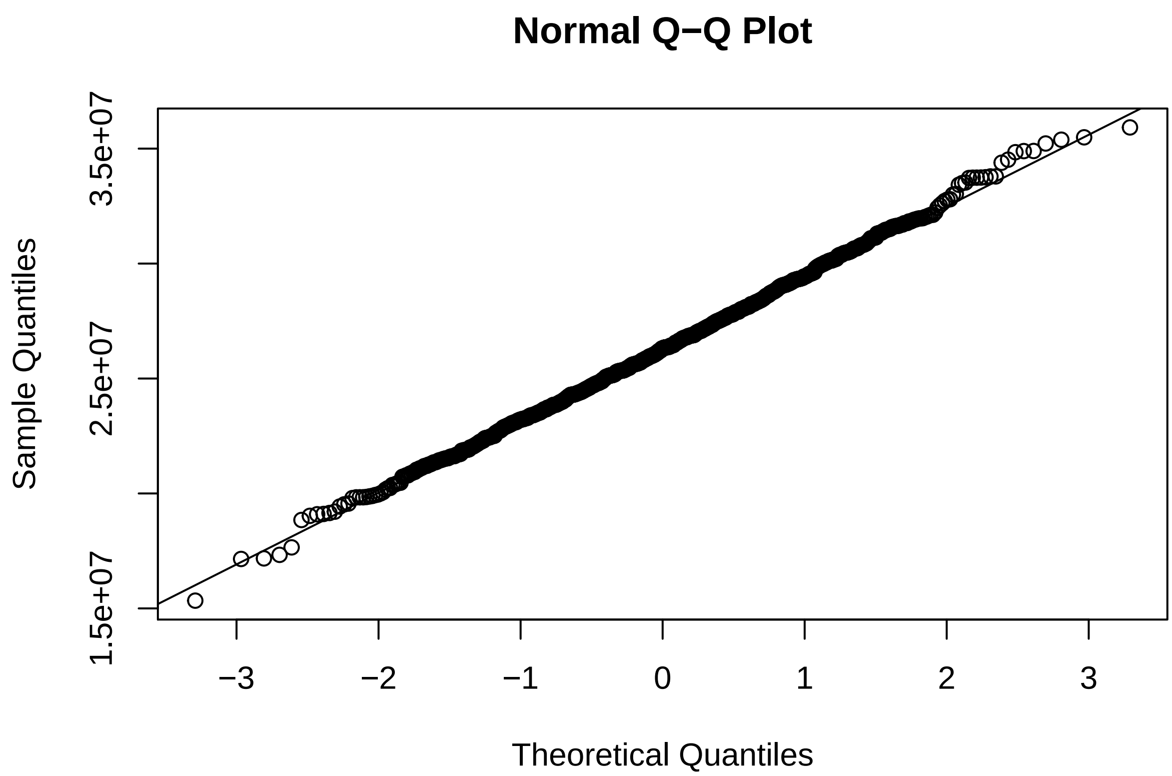

```
hist(r.mean)
```




```r
qqnorm(g.mean)
qqline(g.mean)
```




```r
qqnorm(r.mean)
qqline(r.mean)
```




```r
# bootstrap distributions are approximately normal

sd(g.mean)
```

```
## [1] 10615595
```

```r
sd(r.mean)
```

```
## [1] 3173080
```

```r
# the standard deviations are substantially different => different variances

# expected difference in means
mean(g.mean - r.mean)
```

```
## [1] 43098647
```

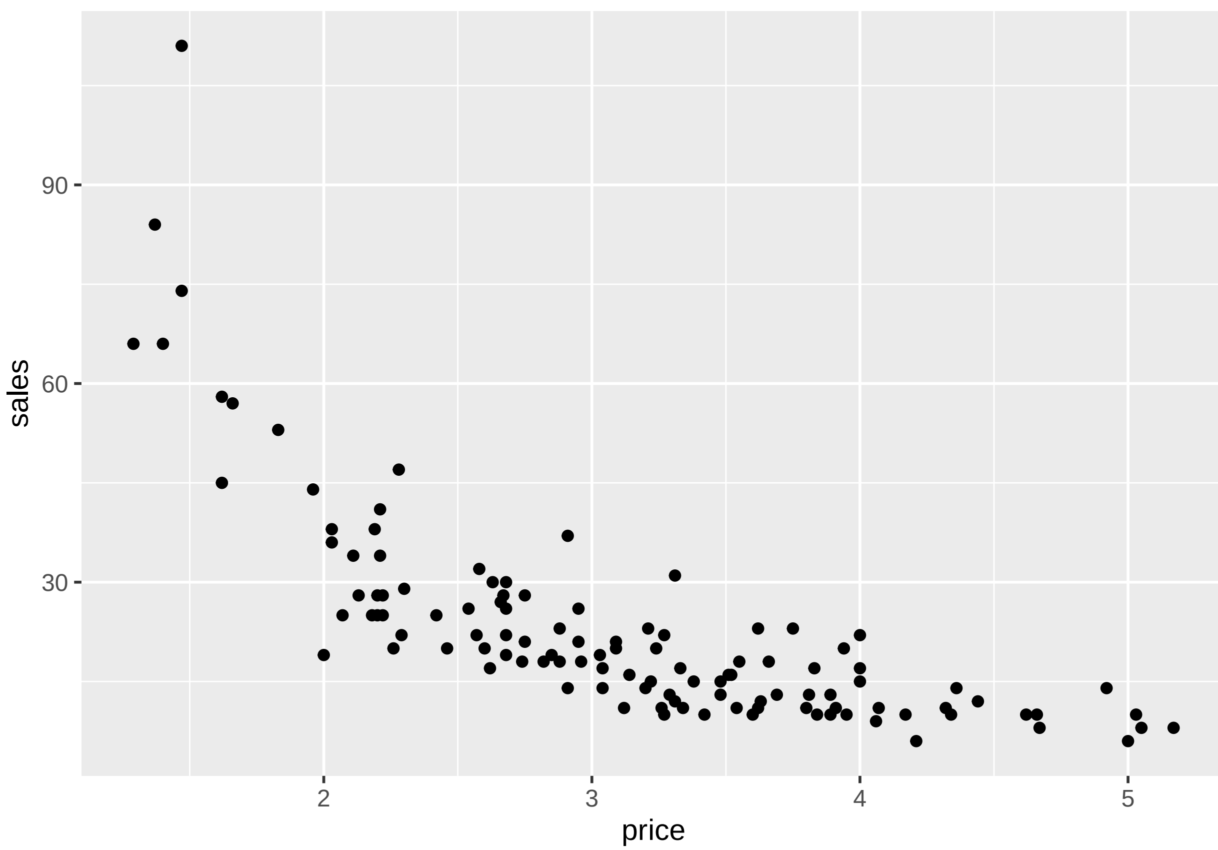
```r
# two sided t-test for difference in means with different variances
t.test(x = g.mean, y = r.mean, var.equal = FALSE, conf.level = 0.9999)
```

```
##
##  Welch Two Sample t-test
##
## data:  g.mean and r.mean
## t = 123.01, df = 1176.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 99.99 percent confidence interval:
##  41730809 44466485
## sample estimates:
## mean of x mean of y
##  69401519  26302872
```

This is a large investment for any company so I decided to use a 99.99% confidence level for our t-test for difference in means. We will see that we would come to the same conclusions with practically any confidence level. With 99.99% confidence we can expect the RelativeTotalRent for buildings with 10 to 20 stories, to be within the interval (42,379,397, 45,086,090). Where the expected difference is centered about 43,732,744 per year. According to the assignment, constructing a green building is expected to cost an additional 5 million dollars. So with an expected 43 million dollars in revenue per year, we would expect to recoup the green ceritification costs within the first year and begin increasing our profit. Because of this I would say that investing in constructing a green building is a wise decision. If I knew which clusters were near the I-35/East Cesar Chavez clusters (where our building is being built), I would redo the previous analysis with similar clusters. However we know that the difference in means is statistically significant (p-value < 2.2e-16).

MILK PRICES

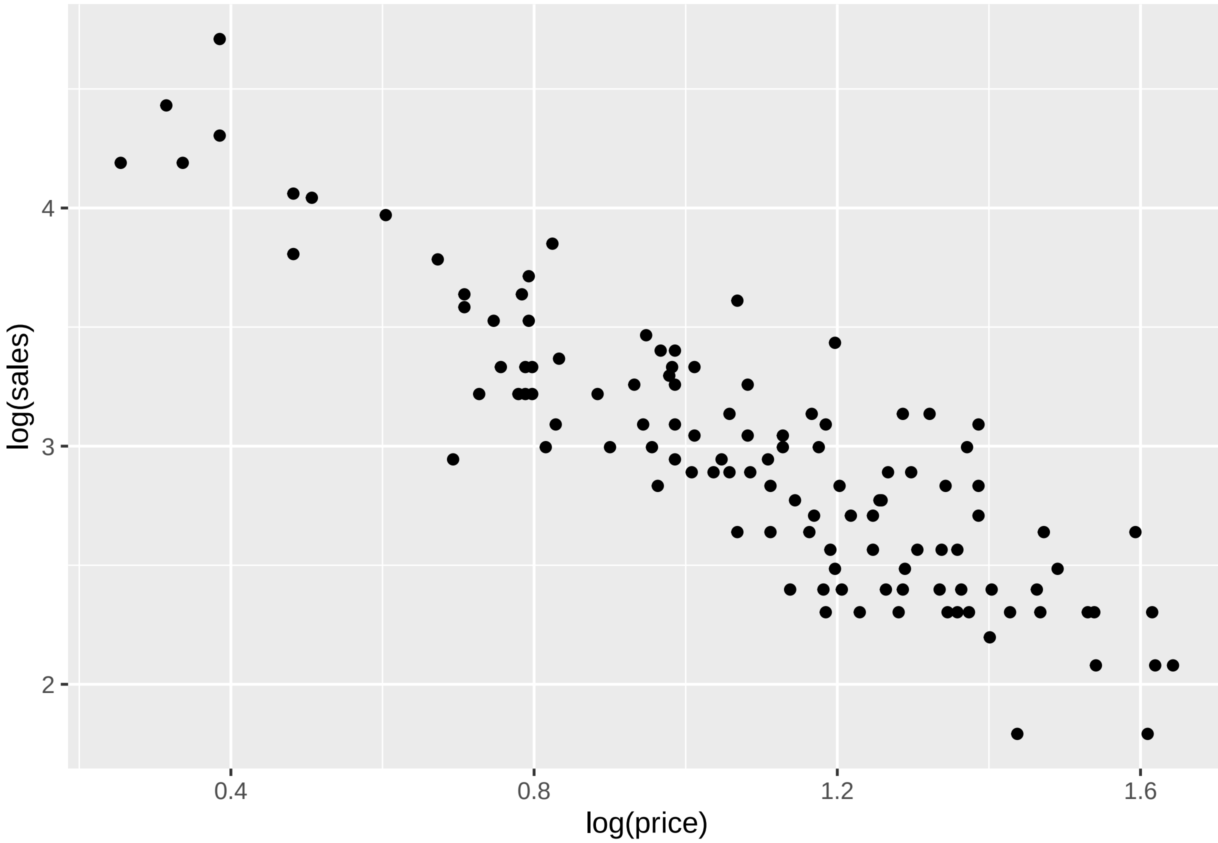
```
library(ggplot2)

milk <- read.csv("~/Documents/R/SDS 323/SDS323-master/data/milk.csv")

ggplot(data = milk, aes(x = price, y = sales)) + geom_point()
```



The data appears to be quadratic or an inverse power (i.e y = x^-a)

```
model1 <- glm(data = milk, sales ~ price)
model2 <- glm(data = milk, sales ~ poly(x = price, degree = 2))
model3 <- glm(data = milk, sales ~ poly(x = price, degree = 3))
model4 <- glm(data = milk, sales ~ poly(x = price, degree = 4))
f1 <- function(x) 133.4321 - 60.0686*x + 7.2914*x^2
f2 <- function(x) 236.6667 - 171.5551*x + 44.3760*x^2 - 3.8469*x^3
f3 <- function(x) exp(4.7206) * x^(-1.6186)

# plotting log(sales) vs log(price) indicates a roughly linear relationship between the two, this is mo
```

```r
ggplot(data = milk, aes(x = log(price), y = log(sales))) + geom_point()
```




```r
model5 <- glm(data = milk, log(sales) ~ log(price))

# LOOCV
library(boot)
cv.glm(data = milk, model1)$delta[1]
```

```
## [1] 125.4126
```

```r
cv.glm(data = milk, model2)$delta[1]
```

```
## [1] 74.68748
```

```r
cv.glm(data = milk, model3)$delta[1]
```

```
## [1] 59.16259
```

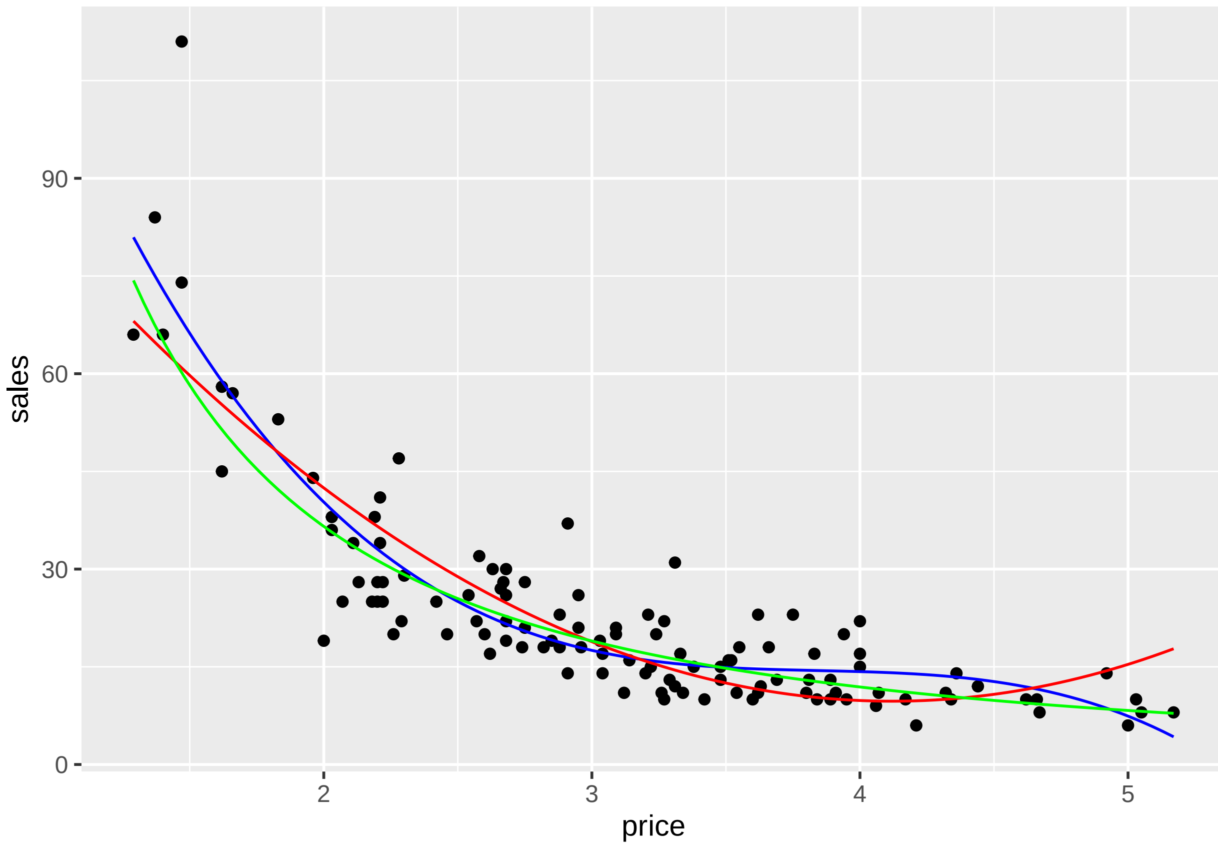
```r
cv.glm(data = milk, model4)$delta[1]
```

```
## [1] 59.32089
```

```r
cv.glm(data = milk, model5)$delta[1]
```

```
## [1] 0.07347467
```

```r
# of the polynomial models, model3 has the smallest MSE, just barely smaller than model4. However with

ggplot(data = milk, aes(x = price, y = sales)) + geom_point() + stat_function(fun = f2, color = "blue")
```



We can see that the red line tails upwards as price increases which is most likely a failure of the model rather than representative of the actual data. Hence why the blue line (model 3) is the more accurate and appropriate polynomial fit. However the green line appears to fit the data even better, and the LOOCV supports this, so we will use the power model to fit our data.

```
# N - net profit
# c - whole sale cost per carton (given)
# P - per unit price
# Q - units sold

# N = (P-c)*Q
# Q = exp(4.7206) * P^(-1.6186)
# N = (P-c)*(exp(4.7206) * P^(-1.6186))
# N'(P) = c*181.664*x^(-2.6186)-69.4289*x^(-1.6186)

library(rootSolve)

# c can be set to any number >= 0
c <- 1
# interval to test over, may need to be expanded for larger values of c
interval <- c(0, 10)

# our functions for net profit
n <- function(P) (P-c)*(exp(4.7206)*P^(-1.6186))
n.prime <- function(P) c*181.664*P^(-2.6186)-69.4289*P^(-1.6186)

# there will only ever be one critical point with this function
root <- uniroot.all(n.prime, interval)
```

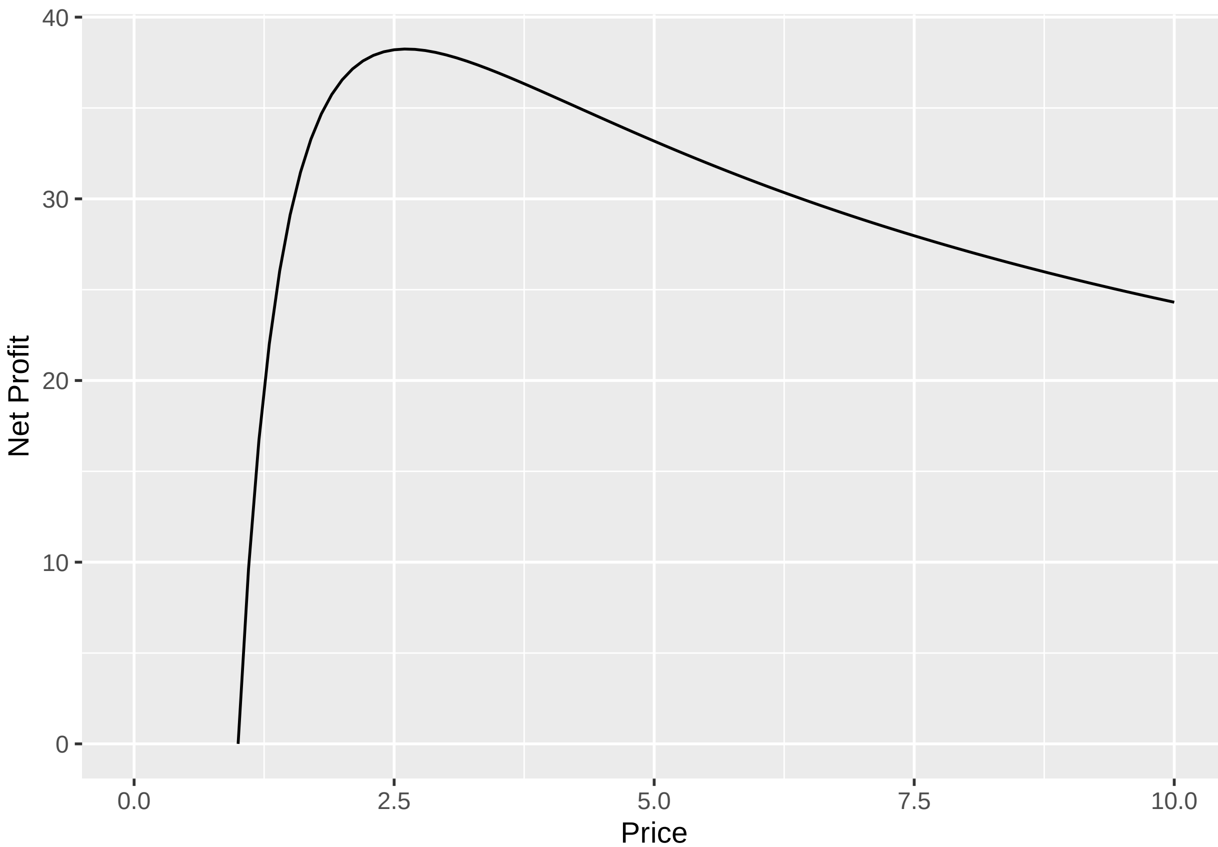
```r
# print x = max and f(max)
print(root)
```

```
## [1] 2.616571
```

```r
n(root)
```

```
## [1] 38.24577
```

```r
# plot net profit vs price given c
ggplot(data = data.frame(x=0), mapping = aes(x = x)) +
  scale_x_continuous(limits = interval) +
  ylim(0, NA) +
  stat_function(fun = n) +
  xlab("Price") +
  ylab("Net Profit")
```



Given c >= P, this graph plots net profit vs price of milk. For c = 1, we can see that maximum profit occurs around P = 2.5 and net profit seems to be slightly less thatn 40. Solving the function directly corroborates this as the maximum occurs at P = 2.62 and f(P) = 38.25. So for a wholesale cost of 1 dollar we would maximize our profit by charging 2.62 dollars per unit of milk. By varying c we can easily find the maximum net profit for any c.