Exercise 1 - SDS323

Alt Nayani and Conor McKinley

2/14/2020

Question 1

Load Data

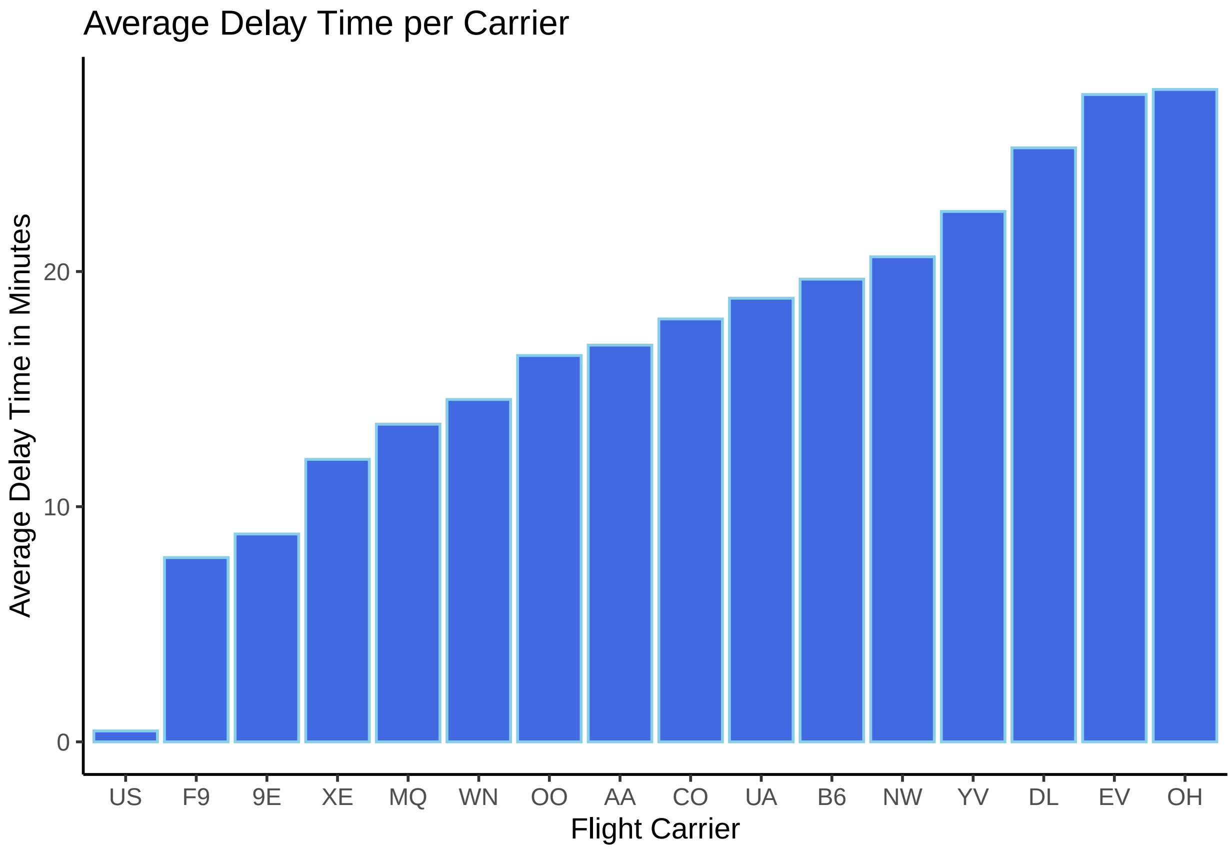
```
Question1url <- "https://raw.githubusercontent.com/jgscott/SDS323/master/data/ABIA.csv"
ABIA <- read_csv(url(Question1url))
```

Question: Which flight carrier, on average, has the longest delay time?

```
ABIA2 <- ABIA %>%
  filter(!is.na(UniqueCarrier)) %>%
  group_by(UniqueCarrier) %>%
  summarise(
    arr_delay_mean = mean(ArrDelay, na.rm = TRUE),
    dep_delay_mean = mean(DepDelay, na.rm = TRUE),
    )

ABIA2 <- ABIA2 %>%
  mutate(total_delay_mean = arr_delay_mean + dep_delay_mean)
ABIA2
```

```
## # A tibble: 16 x 4
##    UniqueCarrier arr_delay_mean dep_delay_mean total_delay_mean
##    <chr>                  <dbl>          <dbl>            <dbl>
##  1 9E                     3.14            5.70             8.84
##  2 AA                     8.14            8.73            16.9
##  3 B6                     8.34           11.3             19.7
##  4 CO                     8.81            9.17            18.0
##  5 DL                    13.8            11.5             25.3
##  6 EV                    10.9            16.6             27.5
##  7 F9                     3.48            4.35             7.84
##  8 MQ                     6.82            6.69            13.5
##  9 NW                    12.6             8.02            20.6
## 10 OH                    14.9            12.9             27.7
## 11 OO                     7.46            8.97            16.4
## 12 UA                     8.84           10.0             18.9
## 13 US                    -0.542           1.00             0.463
## 14 WN                     5.00            9.56            14.6
## 15 XE                          5.57            6.44           12.0
## 16 YV                         10.9            11.6            22.6
```

```
ggplot(data = ABIA2, mapping = aes(x = reorder(UniqueCarrier, total_delay_mean), y = total_delay_mean))
  geom_col(fill = "royalblue", colour = "skyblue") +
  labs(title = "Average Delay Time per Carrier",
       y = "Average Delay Time in Minutes",
       x = "Flight Carrier") +
  theme_classic()
```



Question 2

Load Data

```
Question2url <- "https://raw.githubusercontent.com/jgscott/SDS323/master/data/creatinine.csv"
Creatinine <- read.csv(url(Question2url))
```

```
### Regression of Clearance Rate on Age.
CreatineRegression <- lm(creatclear ~ age, data = Creatinine)
CreatineRegression$coefficients
```

```
## (Intercept)         age
## 147.8129158  -0.6198159
```

As age increases by a unit, Clearance Rate decrease by 0.62 on average.

```
### The actual Clearance Rate
CreatinineMean <- Creatinine %>%
  group_by(age) %>%
  summarise(average = mean(creatclear, nm.rm = TRUE))
CreatinineMean[34,]
```

```
## # A tibble: 1 x 2
##     age average
##   <int>   <dbl>
## 1    55     114
```

```
### The expected/predicted Clearance Rate
Age55Expected <- data.frame(age = 55)
Prediction55 <- predict(CreatineRegression, Age55Expected)
Prediction55
```

```
##       1
## 113.723
```

The actual clearance rate for the 55 year old was 114, compared to the predicted clearance rate of 113.72 from the modeled regression.

```
### Determine residual of age 40 and 60 year old
Age40 <- data.frame(age = 40)
Prediction40 <- predict(CreatineRegression, Age40)
Residual40 <- 135-Prediction40
Residual40
```

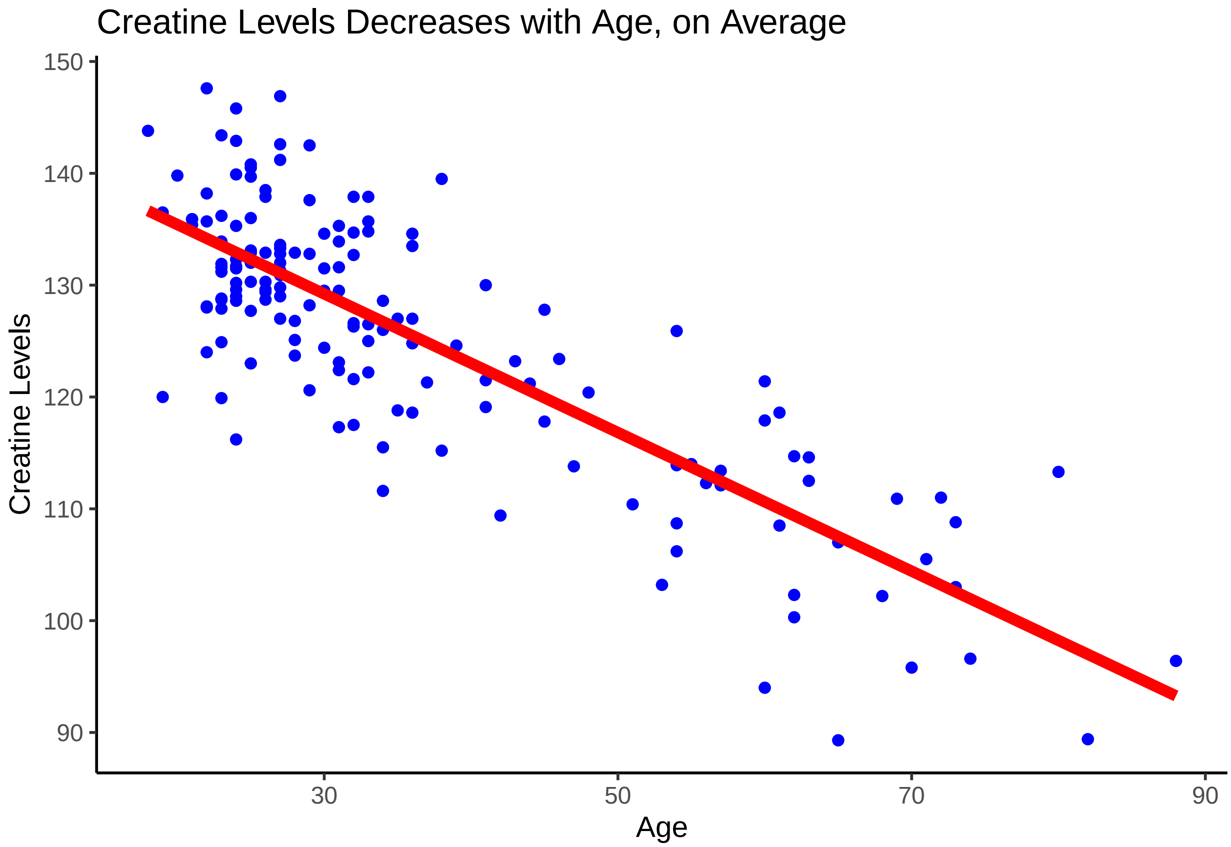
```
##        1
## 11.97972
```

```
Age60 <- data.frame(age = 60)
Prediction60 <- predict(CreatineRegression, Age60)
Residual60 <- 112-Prediction60
Residual60
```

```
##        1
## 1.376035
```

After accounting for age, the 40 year-old is healthier based on their Clearance Rate since the residual of the 40-year old is greater than that of the 60-year old.

```
### Plot of Clearance Rate vs. Age
Creatinine$Predicted <- fitted(CreatineRegression)

ggplot(data = Creatinine) +
  geom_point(mapping = aes(x = age, y = creatclear), color = "blue") +
  geom_line(mapping = aes(x = age, y = Predicted), color = "red", size = 2) +
  labs(title = "Creatine Levels Decreases with Age, on Average",
       x = "Age",
       y = "Creatine Levels") +
  theme_classic()
```



Question 3

Load Data

```
Question3url <- "https://raw.githubusercontent.com/jgscott/SDS323/master/data/greenbuildings.csv"
greenbuildings <- read_csv(url(Question3url))
```

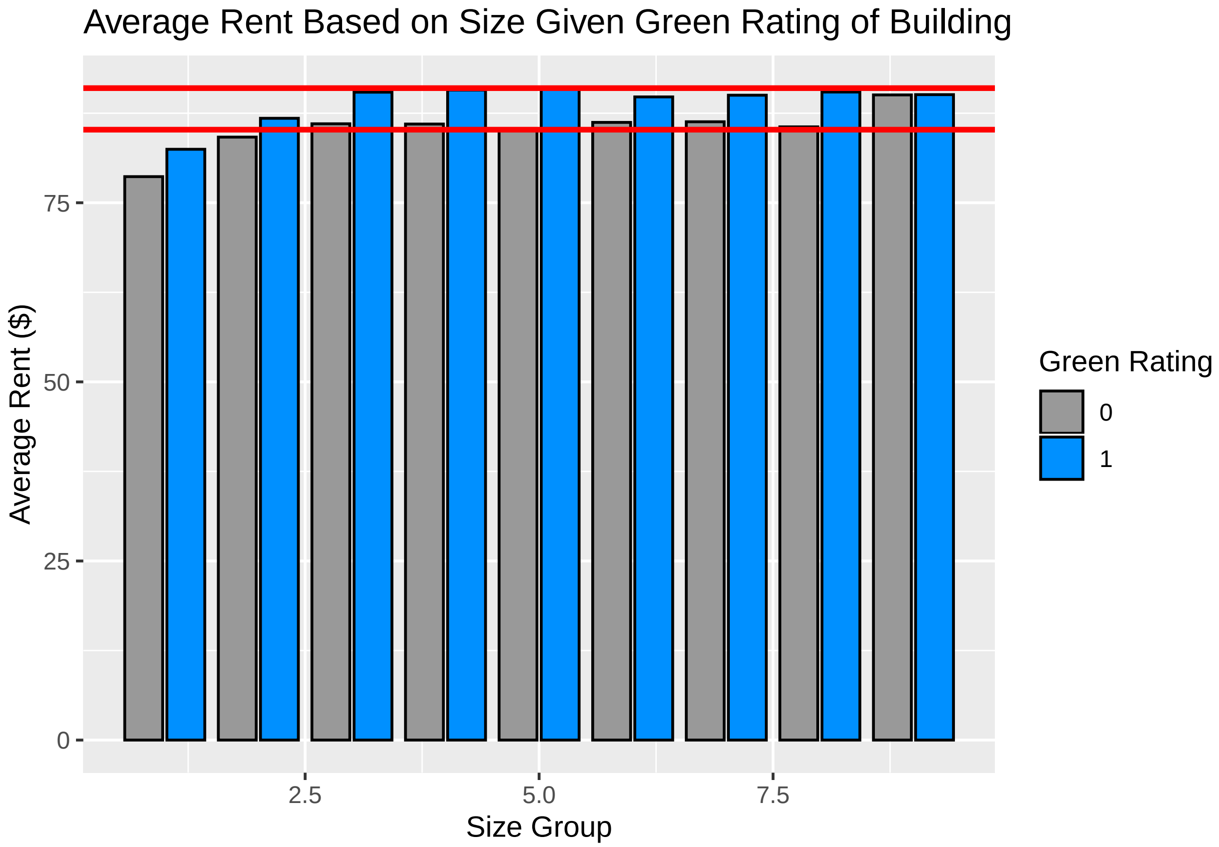
```
### Define Global Variables
BuildingSqFeet = 250000
GreenCertificationCost = 5000000


### Scrub below 10% occupancy and select key variables
greenbuildings <- greenbuildings %>%
  filter(leasing_rate > 10) %>%
  select(size, cluster, green_rating, leasing_rate, Rent) %>%
  mutate(green_rating = factor(green_rating, levels=c(0,1), labels=c("0", "1")))

### Split Price and Size (sq.ft) into groups
sizecuts <- c(-Inf, 50000, 100000, 150000, 200000, 250000, 300000, 350000, 400000, Inf)
sizenames <- c("0 to 50k","50k to 100k", "100k to 150k", "150k to 200k", "200k to 250k", "250k to 300k"

greenbuildings$sizegroup <- as.numeric(cut(greenbuildings$size, breaks = sizecuts, labels = sizenames))
```

```
greenbuildings <- greenbuildings %>%
  group_by(sizegroup, green_rating) %>%
  summarise(averagerent = mean(leasing_rate, na.rm = TRUE),
            medianrent = median(leasing_rate, na.rm = TRUE))
```

The Excel guru's analysis is not wrong but it can be improved. There are a large number of confounding variables that can alter the difference in rent prices for green buildings v. non-green buildings. Some key variables to determine the actual economic impact of going green is the location of the building (cluster) and the size of the building.

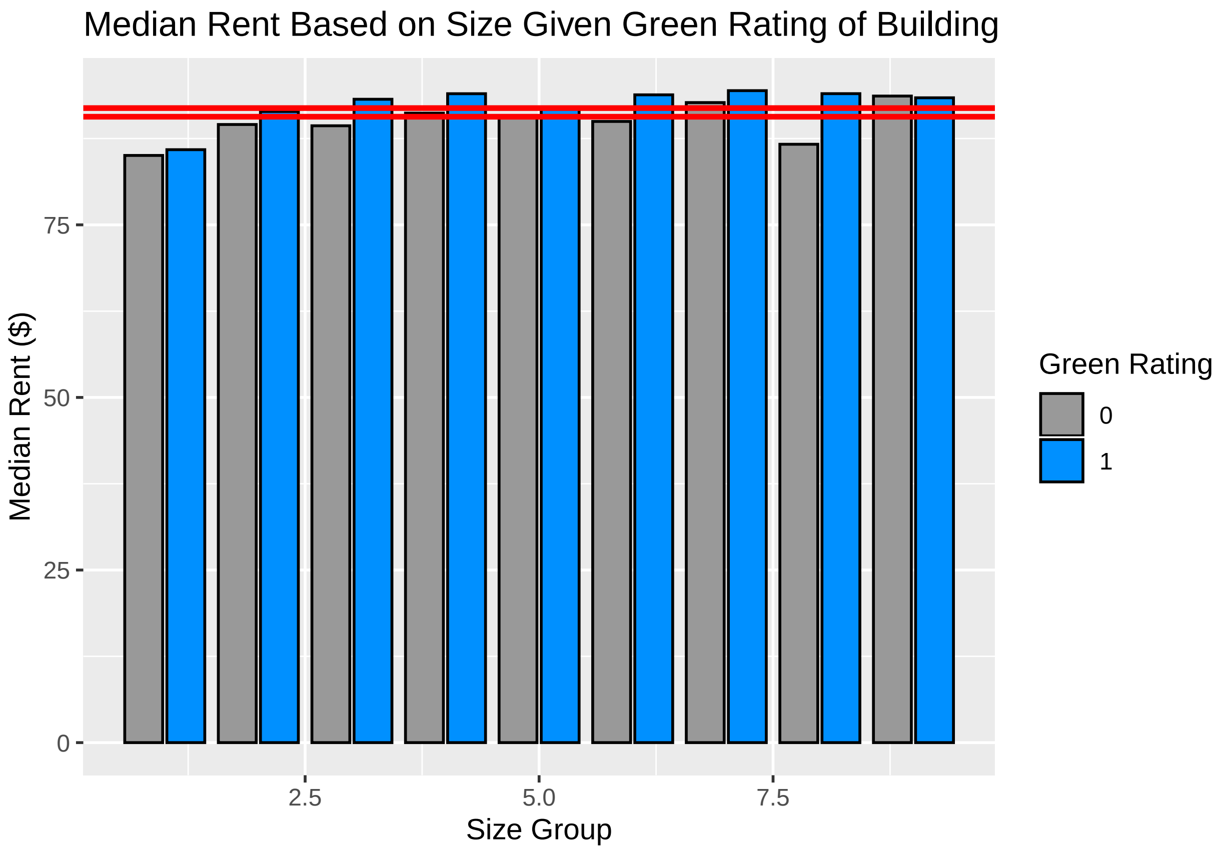
A potential problem of the data set is we are unsure the specific buildings in the cluster and cannot determine where the building would lie within the clusters; therefore, only the size was taken a look at.

```
ggplot(greenbuildings, aes(x = sizegroup, y = averagerent, fill = green_rating)) +
  geom_col(position = position_dodge2(), colour = "black") +
  scale_fill_manual(values = c("#999999", "#0090ff")) +
  labs(title = "Average Rent Based on Size Given Green Rating of Building",
       x = "Size Group",
       y = "Average Rent ($)",
       fill = "Green Rating") +
  geom_hline(yintercept = 85.2, size = 1, color = "red") +
  geom_hline(yintercept = 91, size = 1, color = "red")
```



It could be seen through the previous graph, that for the size of the East Cesar Chavez Building would fall into the fifth group. In the fifth group (which contains buildings from 250k to 300k sq. ft) the average rent for a green building was 91.00 while the average rent for a non-green building was 85.27. Thus, a green building would provide, on average, 6.73 benfit per sq. foot.

```
ggplot(greenbuildings, aes(x = sizegroup, y = medianrent, fill = green_rating)) +
  geom_col(position = position_dodge2(), colour = "black") +
  scale_fill_manual(values = c("#999999", "#0090ff")) +
  labs(title = "Median Rent Based on Size Given Green Rating of Building",
       x = "Size Group",
       y = "Median Rent ($)",
       fill = "Green Rating") +
  geom_hline(yintercept = 90.66, size = 1, color = "red") +
  geom_hline(yintercept = 91.91, size = 1, color = "red")
```

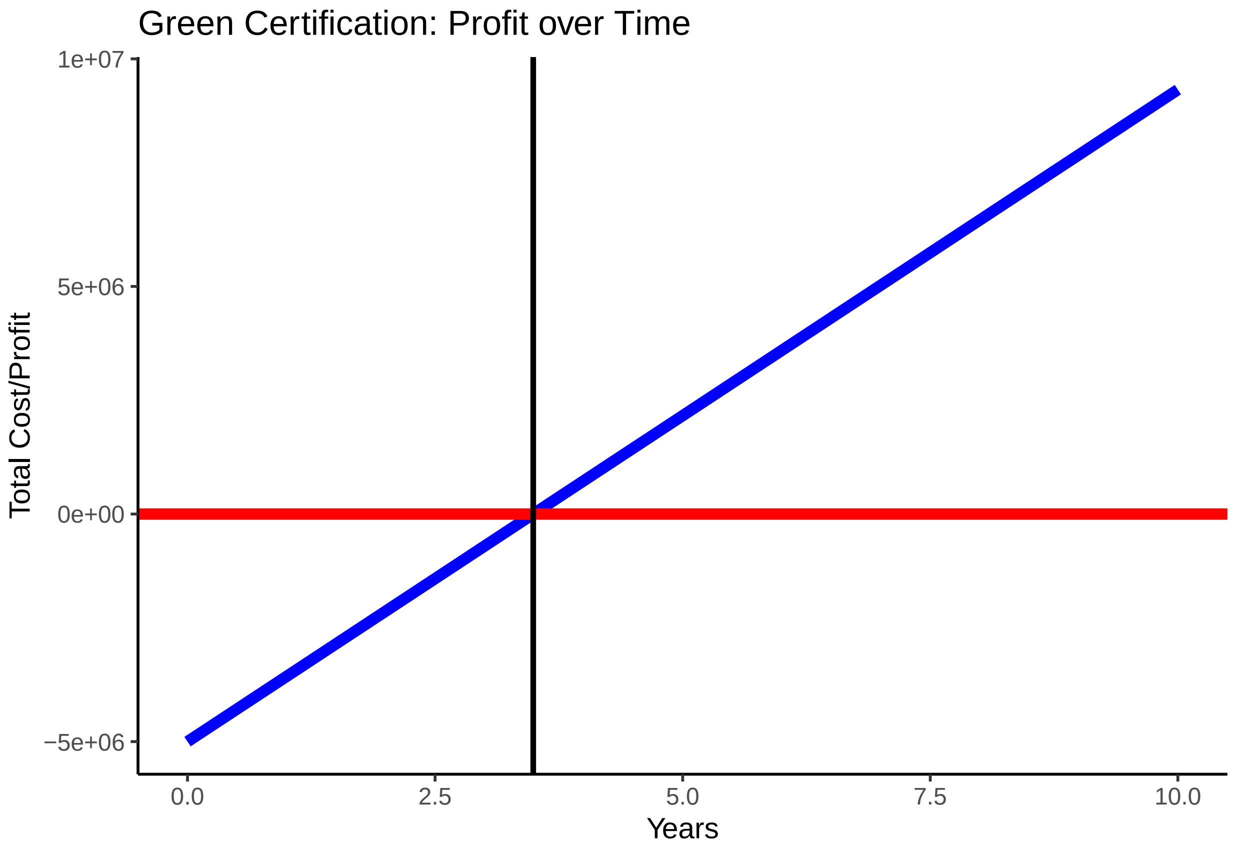


Furthermore, the economic benefit for the green buildings v. non-green building for the fifth group using median was only 1.25. Green buildings had a median rent of 91.91 and non-green buildings had a median rent of 90.66.

We can also determine the repayment time (payback period) of the Green Certificate, originally costing 5M by using the incremental revenue gain of going green. The expected yearly profit would be 1,432,330 using the average premium of a green building vs. a non-green building. The average was used in this analysis, because while it can be skewed it gives a true weighted distribution of what the actual benefits of having a green building could be.

Using the Green Certificate cost and dividing it by the average premium of a green building, it was determined the repayment time was approximaltely 3.5 years.

```r
ExpectedYearlyProfit = (greenbuildings$averagerent[10] - greenbuildings$averagerent[9])*BuildingSqFeet
RepaymentLength = GreenCertificationCost/ExpectedYearlyProfit

summedprofit = matrix(0, 11, 2)
summedprofit[1,2] = -GreenCertificationCost
summedprofit[,1] = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

for (i in 2:11) {
  summedprofit[i,2] = summedprofit[i-1,2] + ExpectedYearlyProfit
}
summedprofitdf = data.frame(summedprofit)

ggplot(summedprofitdf) +
  geom_line(mapping = aes(x = summedprofitdf$X1, y = summedprofitdf$X2), color = "blue", size = 2) +
  geom_hline(yintercept = 0, size = 2, color = "red") +
  geom_vline(xintercept = RepaymentLength, size = 1, color = "black") +
  labs(title = "Green Certification: Profit over Time",
       x = "Years",
       y = "Total Cost/Profit") +
  theme_classic()
```



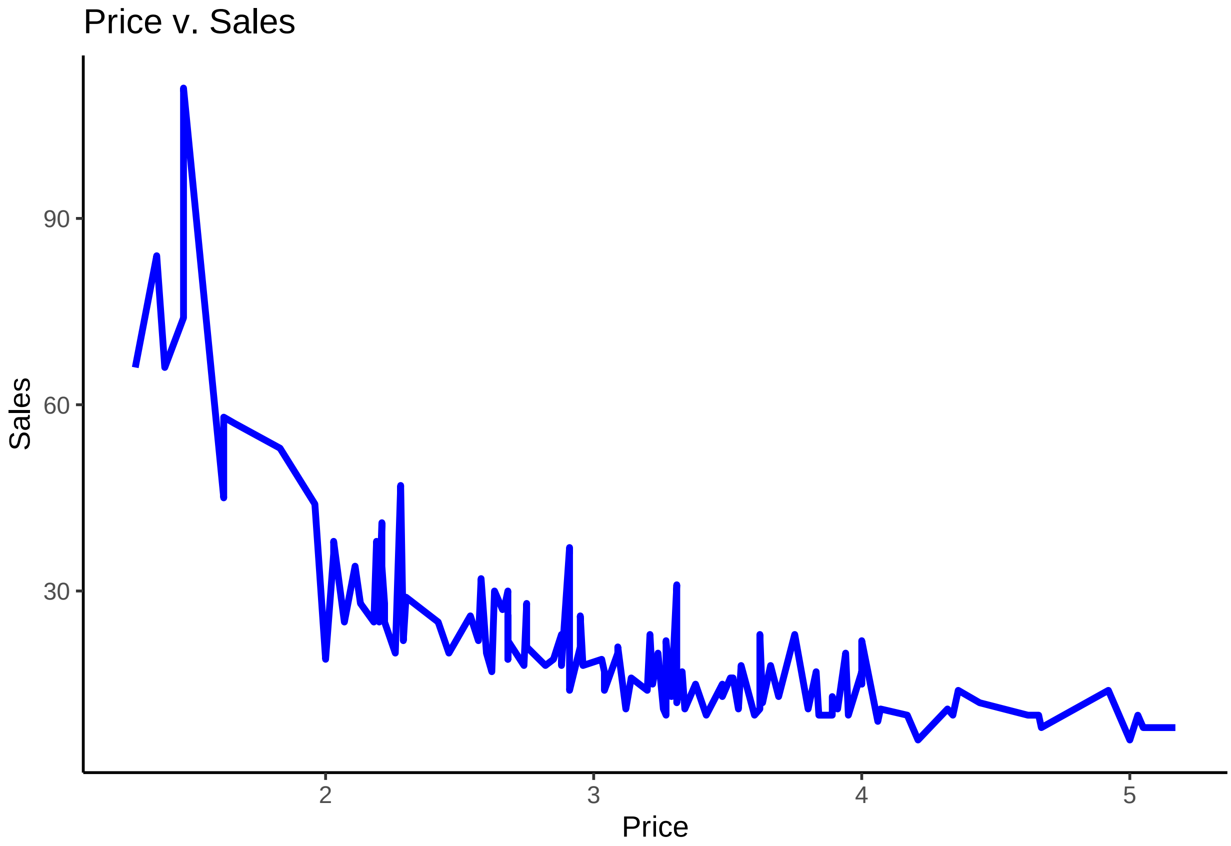
Question 4

Load Data

```
milk <- read.csv("milk.csv")
```

To determine what price to charge for a gallon of a milk, it is a simple optimization problem where you maximize Quantity * (Price - Cost). However, the problem is that Quantity is a function of the price, where charging a higher price leads to a lower quantity demanded and vice versa. Therefore, we need to determine the shape of the demand curve.

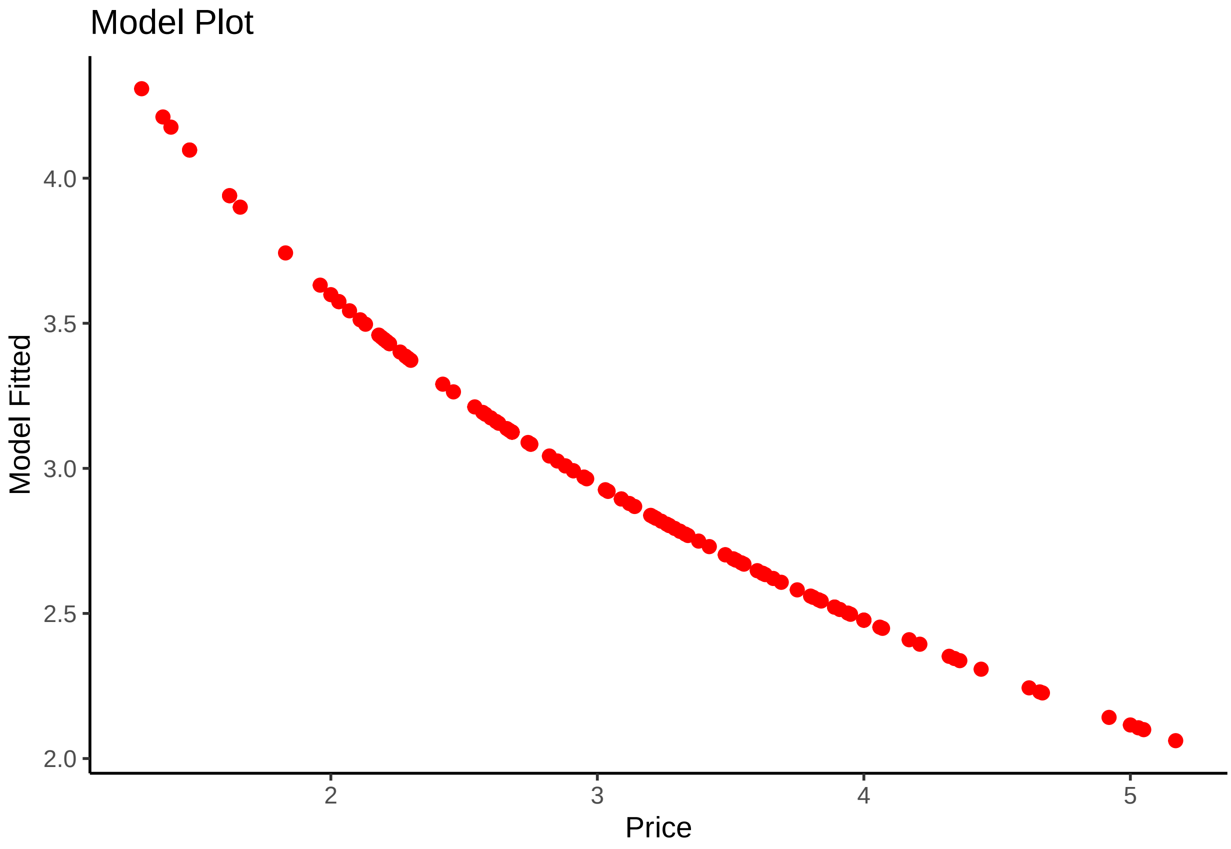
```
ggplot(milk) +
  geom_line(mapping = aes(x = price, y = sales), colour = "blue", size = 1.2) +
  labs(title = "Price v. Sales",
       x = "Price",
       y = "Sales") +
  theme_classic()
```



The relationship seems to be a power law. Therefore, we cannot run a simple linear regression, but rather need to take the log of the price and sales to run a regression.

The following is a plotted model of the logged linear regression.

```
demandcurve = lm(log(sales) ~ log(price), data = milk)
fitteddata = data.frame(
  x = milk$price,
  y.fit = fitted(demandcurve))

ggplot(fitteddata)+
  geom_point(mapping = aes(x = x, y = y.fit), size = 2, colour = "red") +
  labs(title = "Model Plot",
       x = "Price",
       y = "Model Fitted") +
  theme_classic()
```



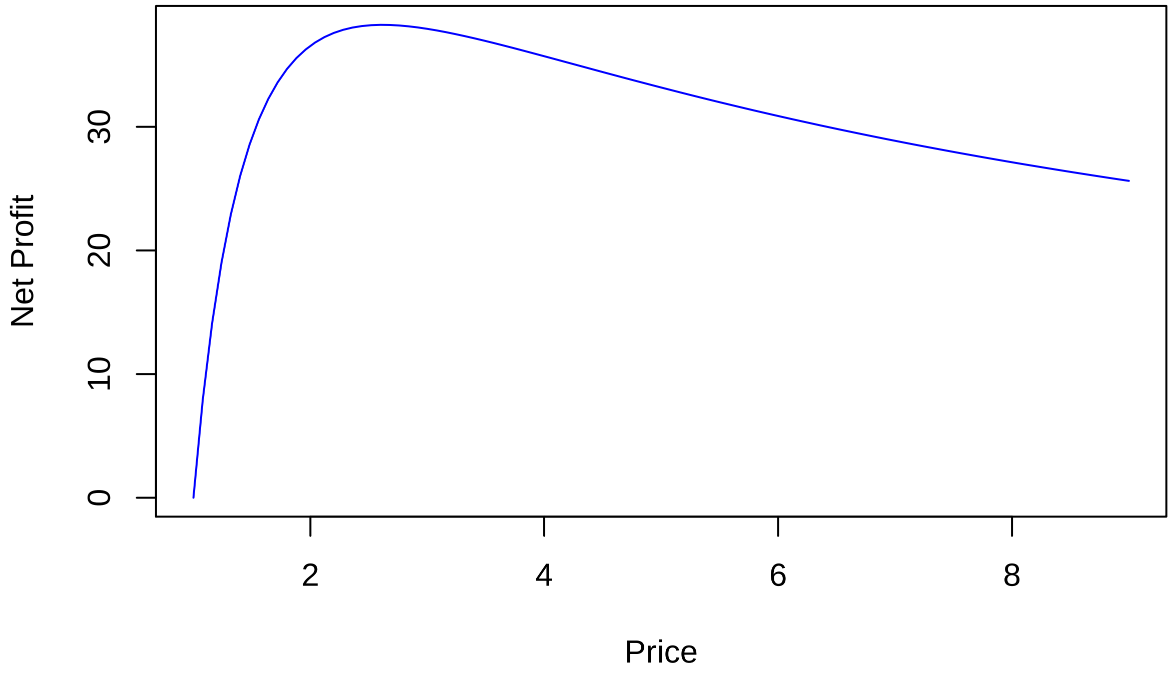
```
Alpha = demandcurve$coefficients[1]
PriceElasticity = demandcurve$coefficients[2]
Perunitcost = 1
```

Next we must convert the coefficients back into a form that can be used with the original price units.

We can now determine the net profit by the function: (Price - Cost) * f(p), where f(p) = exp(alpha) * Price ˆ (PriceElasticity). In this case, alpha was 4.7 and the price elascity was -1.62.

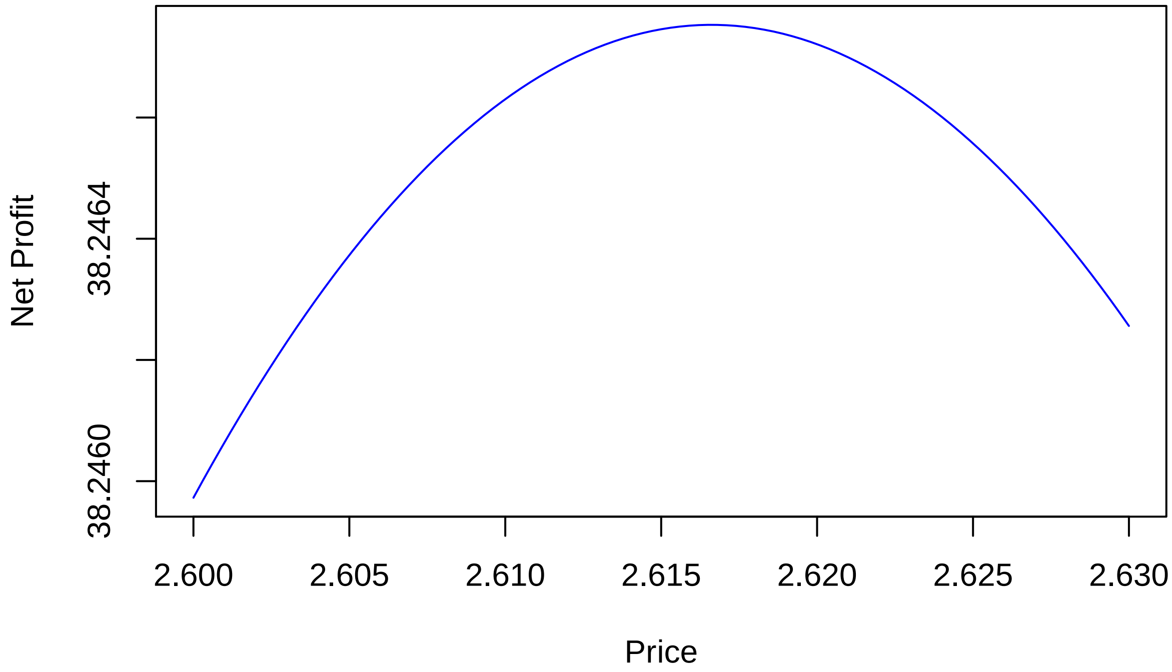
The following is a plot of net profit as price changes.

```
curve((x-Perunitcost)*exp(Alpha)*x^(PriceElasticity), from = 1, to = 9,
      xlab = "Price",
      ylab = "Net Profit",
      col = "blue")
```



We can zoom into to find a specific price to charge.

```
curve((x-Perunitcost)*exp(Alpha)*x^(PriceElasticity), from = 2.6, to = 2.63,
      xlab = "Price",
      ylab = "Net Profit",
      col = "blue")
```



It can be determined, we should charge $2.62 per gallon of milk.