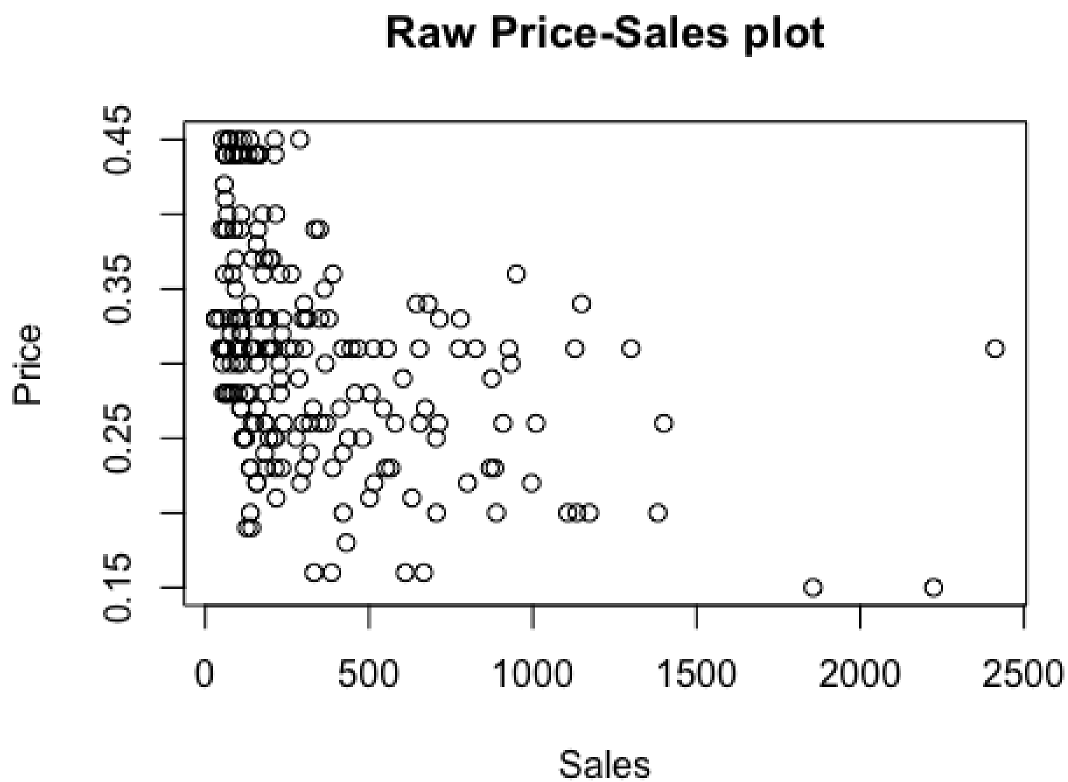
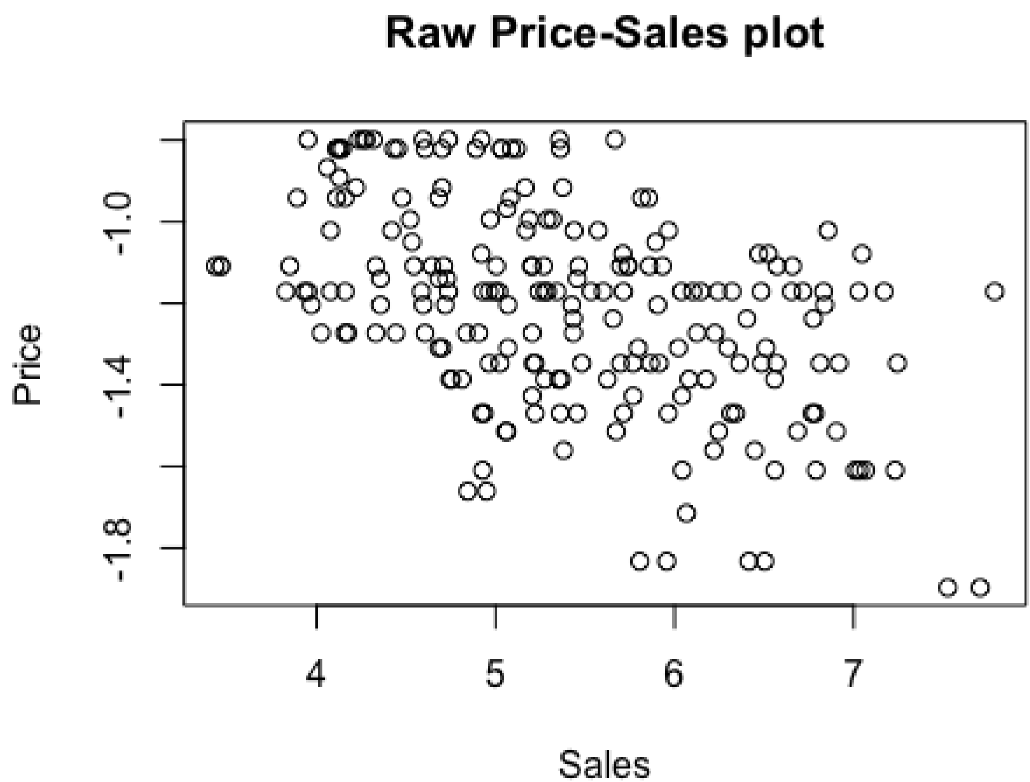
Group HW 1

2024-10-21

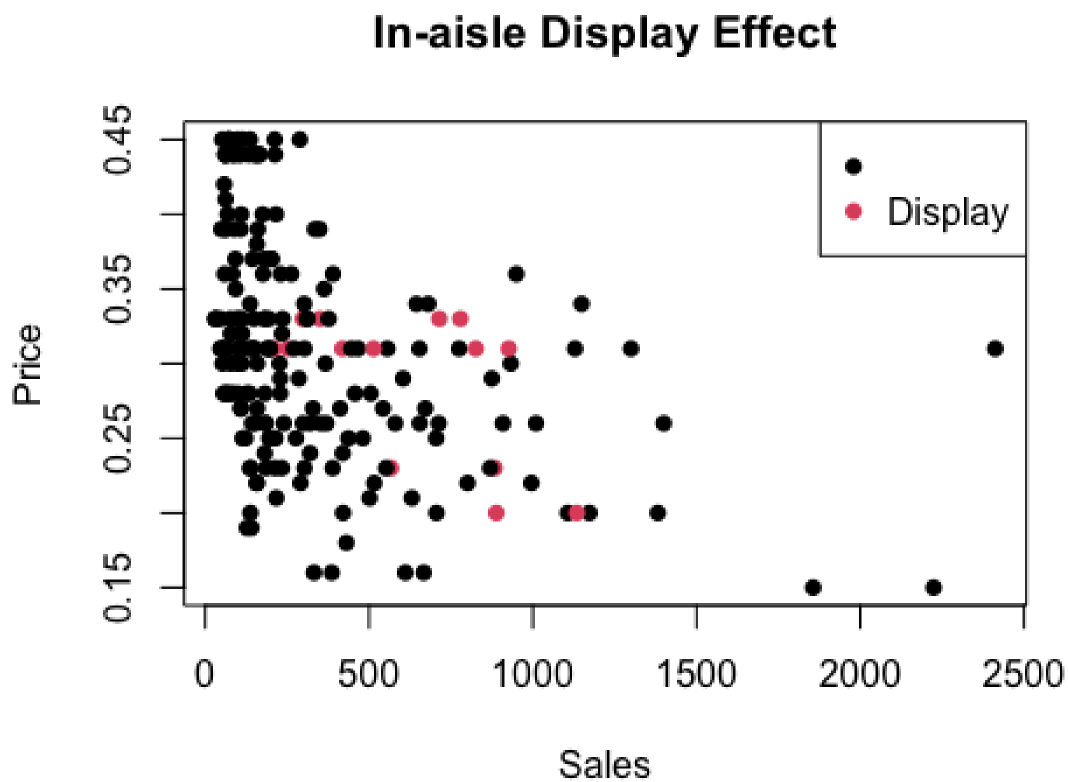
```r
df = read.csv("oj_data.csv")

nrow(df)

## [1] 208

head(df, 15)

##    Product_id Store_id Week Sales Price Holiday Display
## 1           1        2  121   189  0.31
## 2           1        2  122   111  0.33
## 3           1        2  123   161  0.39
## 4           1        2  124   108  0.39
## 5           1        2  125    62  0.41
## 6           1        2  126    93  0.35
## 7           1        2  127    98  0.31
## 8           1        2  128  1129  0.31
## 9           1        2  129   181  0.33
## 10          1        2  130    92  0.37
## 11          1        2  131   303  0.31
## 12          1        2  132   149  0.33
## 13          1        2  133    83  0.36 Holiday
## 14          1        2  134   227  0.30
## 15          1        2  135   194  0.33

tail(df)

##     Product_id Store_id Week Sales Price Holiday Display
## 203          3      137  167   159  0.27
## 204          3      137  168   389  0.23 Holiday
## 205          3      137  169  1010  0.26
## 206          3      137  170   543  0.27
## 207          3      137  171  2224  0.15
## 208          3      137  172   611  0.16 Holiday

#Plot the data
plot(x = df$Sales,  ## x-coordinates
     y = df$Price,  ## y-coordinates
     type = "p",    ## type of the graph ("p"= points, "l" = line)
     cex=1,         ## Size of the point
     col = "black", ## color of the point
     xlab = "Sales",  ##  label on x-axis
     ylab = "Price",       ##  label on y-axis
     main = "Raw Price-Sales plot")
```



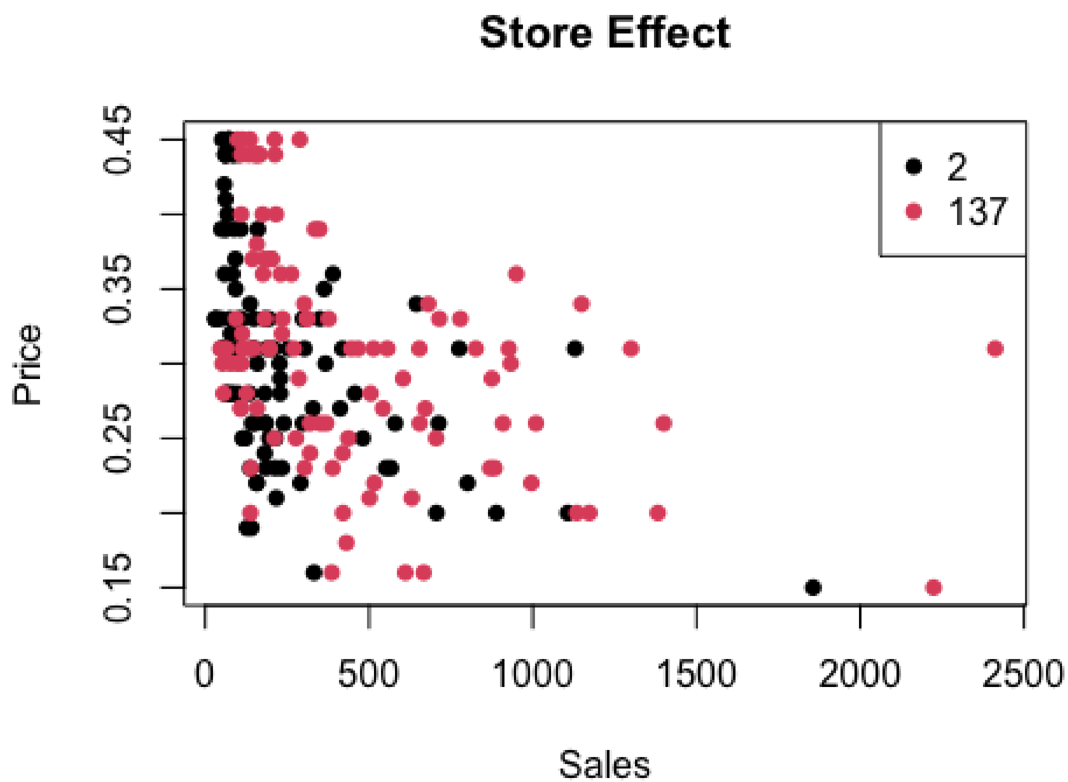

```
## Add logged variables to the data frame
df$logSales = log(df$Sales)
df$logPrice = log(df$Price)

#Plot the data
plot(x = df$logSales,  ## x-coordinates
     y = df$logPrice,  ## y-coordinates
     type = "p",    ## type of the graph ("p"= points, "l" = line)
     cex=1,         ## Size of the point
     col = "black", ## color of the point
     xlab = "Sales",  ##  label on x-axis
     ylab = "Price",       ##  label on y-axis
     main = "Raw Price-Sales plot")
```



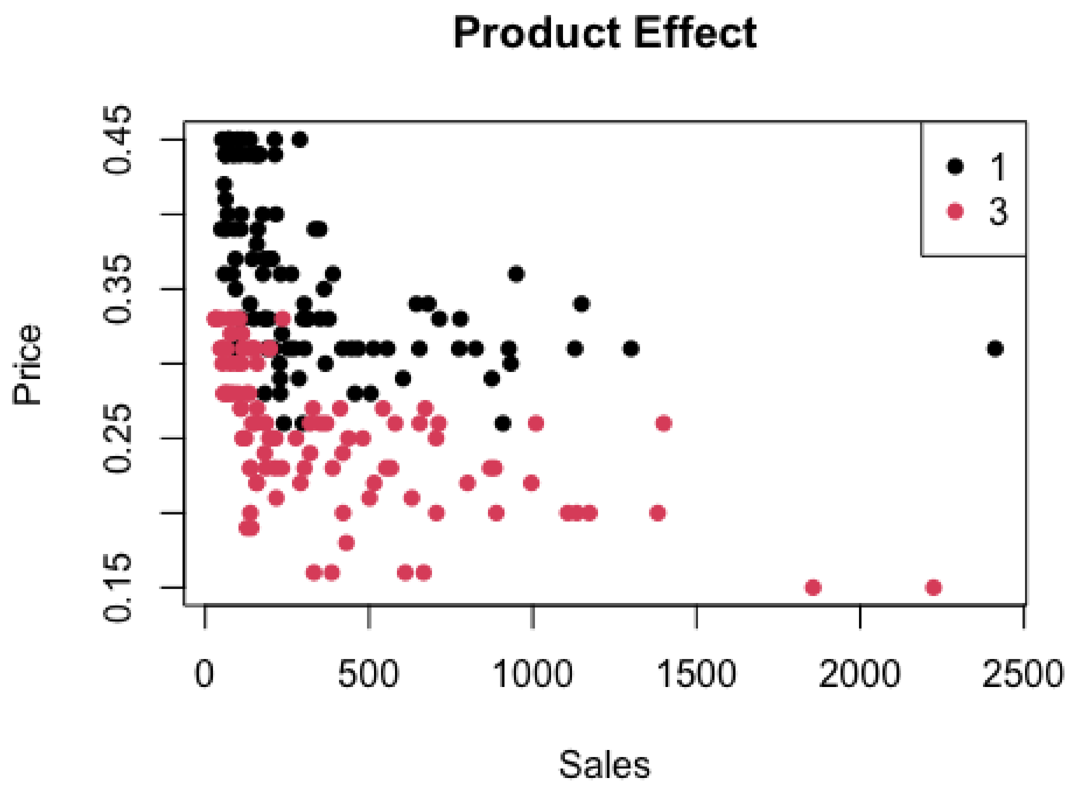

```
df$Display_cat= factor(df$Display) #Categorizing display variable
df$Holiday_cat= factor(df$Holiday) #Categorizing display variable

## Now play with colors - Color by product
plot(x = df$Sales,
     y = df$Price,
     type = "p",     ## type of the graph ("p"= points, "l" = line)
     cex = 1,      ## shape of the point. filled circle
     pch = 16, #filled circles
     col = df$Display_cat,  ## color will differ depending on whether there
is a promotional display
     xlab = "Sales",  ##  label on x-axis
     ylab = "Price",       ##  label on y-axis
     main = "In-aisle Display Effect")

legend("topright", ### location of legend
       legend = unique(df$Display_cat),
       col=1:length(df$Display_cat),
       pch=16)
```




```
# Creating a numerical dummy variable
#Making a display dummy variable
df$DisplayDummy = 1*(df$Display == "Display")

df$HolidayDummy = 1*(df$Holiday == "Holiday")

#Show the Store Effect
df$Store_cat= factor(df$Store_id)
## Now play with colors - Color by product
plot(x = df$Sales,
     y = df$Price,
     type = "p",    ## type of the graph ("p"= points, "l" = line)
     cex = 1,     ## shape of the point. filled circle
     pch = 16, #filled circles
     col = df$Store_cat,  ## color will differ depending on whether there is
a promotional display
     xlab = "Sales",  ##  label on x-axis
     ylab = "Price",       ##  label on y-axis
     main = "Store Effect")
legend("topright", ### location of legend
       legend = unique(df$Store_cat),
       col=1:length(df$Store_cat),
       pch=16)
```




```
#Show the Product Effect
df$Product_cat= factor(df$Product_id)
## Now play with colors - Color by product
plot(x = df$Sales,
     y = df$Price,
     type = "p",    ## type of the graph ("p"= points, "l" = line)
     cex = 1,     ## shape of the point. filled circle
     pch = 16, #filled circles
     col = df$Product_cat,  ## color will differ depending on whether there
is a promotional display
     xlab = "Sales",  ##  label on x-axis
     ylab = "Price",      ##  label on y-axis
     main = "Product Effect")
legend("topright", ### location of legend
       legend = unique(df$Product_cat),
       col=1:length(df$Product_cat),
       pch=16)
```




```
# visual inspection tells pricing is probably different for products - maybe
premium and regular brands

## Make a dummy variable for store and product
df$Product1 = 1*(df$Product_id == 1)
df$Store2 = 1*(df$Store_id == 2)
```

Question 1 and 2

```
df$Product1logPrice = df$Product1 * df$logPrice #interaction term with
product dummy

out_reg_hw = lm(logSales ~ logPrice + Product1logPrice + Product1 + Store2 +
DisplayDummy + HolidayDummy, data=df)
summary(out_reg_hw)

##
## Call:
## lm(formula = logSales ~ logPrice + Product1logPrice + Product1 +
##     Store2 + DisplayDummy + HolidayDummy, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.58222 -0.38231 -0.06522  0.29952  1.61680
```

```
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.28031    0.43259   2.960 0.003450 **
## logPrice          -3.24906    0.30696 -10.585  < 2e-16 ***
## Product1logPrice -0.21527    0.48671  -0.442 0.658748
## Product1           0.83374    0.58079   1.436 0.152691
## Store2            -0.64772    0.08337  -7.770 3.97e-13 ***
## DisplayDummy       0.61024    0.17217   3.544 0.000489 ***
## HolidayDummy      -0.38695    0.12354  -3.132 0.001994 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6005 on 201 degrees of freedom
## Multiple R-squared:  0.6002, Adjusted R-squared:  0.5883
## F-statistic:  50.3 on 6 and 201 DF,  p-value: < 2.2e-16
```

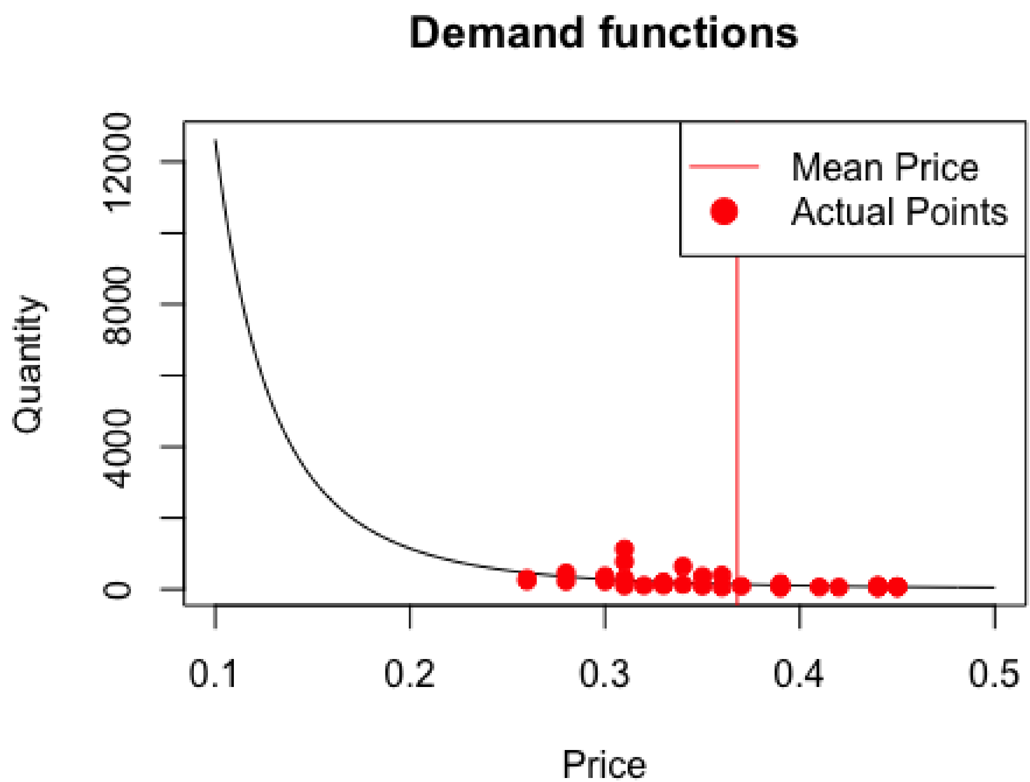
Question 3

```
demand_results_hw = out_reg_hw$coefficients # estimated coefficients from the
model
out_reg_hw$coefficients
```

```
##      (Intercept)         logPrice Product1logPrice         Product1
##        1.2803089       -3.2490640       -0.2152722        0.8337358
##           Store2     DisplayDummy     HolidayDummy
##       -0.6477201        0.6102362       -0.3869468
```

```
demand_oj_hw = function(Price, Product1, Store2) {
  ## demand function baesd on the regression results
  ## Which equation does this refer to?
  ## Which data variables is this function using?
  Q = exp(demand_results_hw[1] + demand_results_hw[2]*log(Price) +
demand_results_hw[3]*Product1*log(Price)+ demand_results_hw[4]*Product1 +
demand_results_hw[5]*Store2)
  Q[which(Q < 0)] = 0 # to make sure that demand is not negative
  return(as.numeric(Q))
}

# creating a vector of prices in increments of .001
## plot predicted demand
price_grid_hw = seq(
  from = 0.1,
  to = .5,
  by = .001) ## create a grid of prices

## what is the demand for these prices for Product 1 at Store 2
demand_grid_hw = demand_oj_hw(price_grid_hw, 1, 1)

plot(x = price_grid_hw,
     y = demand_grid_hw,
     xlab = "Price",
```

```r
     ylab = "Quantity",
     type ="l",# linetype. 1 = solid line. Default option is 1.
     lty = 1,
     main = "Demand functions") ## now type is line

Price_p1s2 = df[df$Product1==1 & df$Store2 == 1 & df$HolidayDummy == 0 &
df$DisplayDummy == 0,]$Price

Sales_p1s2 = df[df$Product1==1 & df$Store2 == 1 & df$HolidayDummy == 0 &
df$DisplayDummy == 0,]$Sales

MeanPrice_p1s2 = mean(Price_p1s2)

points(x = Price_p1s2, y = Sales_p1s2,lty=2,col="red",pch=19)

abline(v=MeanPrice_p1s2, col="red")

# Add a legend
legend("topright", # Location of the legend
       legend = c("Mean Price", "Actual Points"), # Text for the legend
       col = c("red", "red"), # Colors of the lines and points
       lty = c(1, NA), # Line type for the mean price (solid line)
       pch = c(NA, 19), # Points for the actual data points
       pt.cex = 1.5) # Size of the points in the legend
```


Demand functions
Mean Price
Actual Points
Quantity
0
4000
8000
12000
0.1
0.2
0.3
0.4
0.5
Price

##

Question 4

Equation: Question 4 and 5 – solving by hand.

$$\ln(\text{Sales}) = \ln(\alpha) + \beta \ln(\text{Price}) + \beta_2 \ln(\text{Price}) * D_{\text{Product}} + \delta D_{\text{Store}} + \mu D_{\text{Product}} + \lambda D_{\text{disp}} + \theta D_{\text{holiday}} + \varepsilon$$

↑ ↑

Q4. Recommended Price:

$$= \frac{\beta}{1+\beta} \times C$$

$$= \left(\frac{-3.24906 - 0.21527}{-3.24906 - 0.21527 + 1} \right) \times 0.1 = \$0.1406$$

Q5. $\frac{\beta}{1+\beta} \times 2c = 0.1406 \times 2 = 0.28$ USD

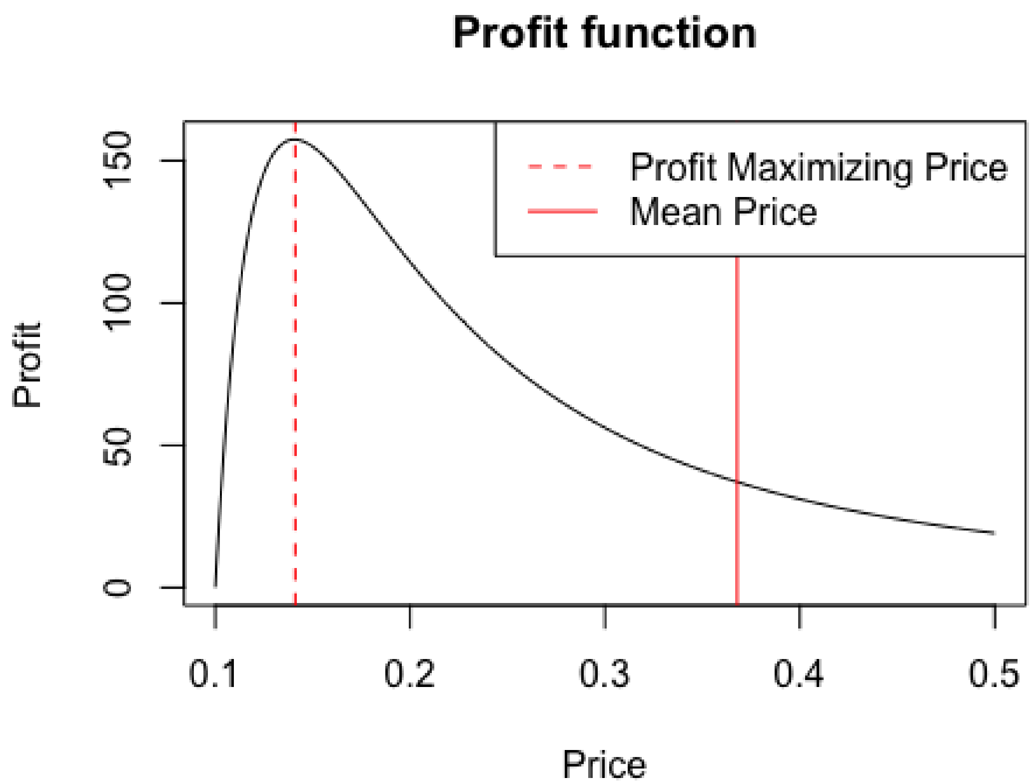
```r
## Define profit function
profit_oj_full <- function(Price, Product1, Store2, unit_cost, fixed_cost) {
  quantity = demand_oj_hw(Price, Product1, Store2) ## compute the demand given price
  profit = quantity * Price - quantity * unit_cost - fixed_cost ## compute profit
  return(profit)
}

profit_grid = profit_oj_full(P = price_grid_hw,
                             Product1 = 1,
                             Store2 = 1,
                             unit_cost = .1,
                             fixed_cost = 0)

df_profit <- data.frame(price_grid_hw, profit_grid)
head(df_profit)
```

```
##   price_grid_hw profit_grid
## 1         0.100     0.00000
## 2         0.101    12.19501
## 3         0.102    23.57160
## 4         0.103    34.18233
## 5         0.104    44.07616
## 6         0.105    53.29863
```

```
df_profit[which.max(df_profit$profit_grid),] #this returns 0.141 which is the
max profit on the profit grid
```

```
##    price_grid_hw profit_grid
## 42         0.141    157.3952
```

```
## Plot the optimal price on the profit function graph
plot(x = price_grid_hw,
     y = profit_grid,
     xlab = "Price",
     ylab = "Profit",
     type ="l",
     main = "Profit function")

abline(v = 0.141, ## x-intercept of the vertical line #profit maximizing
price
       lty = 2, ## linetype. 2 = dash
       col = 'red')

abline(v=MeanPrice_p1s2, col="red")

legend("topright", # Location of the legend
       legend = c("Profit Maximizing Price", "Mean Price"), # Text for the
legend
       col = "red", # Color for both lines
       lty = c(2, 1), # Dashed line for profit-maximizing price, solid line
for mean price
       pt.cex = 1.5) # Size of the points in the legend
```




```
MeanPrice_p1s2 <- 0.368
RecommendedPrice <- 0.141

# profit for the mean price (0.368)
mean_profit_row <- df_profit[which(df_profit$price_grid_hw ==
MeanPrice_p1s2), ]
mean_profit <- mean_profit_row$profit_grid

# Calculate the profit difference
recommended_profit = 157.3952
profit_difference <- recommended_profit - mean_profit

# Output the profit difference
print(profit_difference)

## [1] 120.3271

profit_grid2 = profit_oj_full(P = price_grid_hw,
                              Product1 = 1,
                              Store2 = 1,
                              unit_cost = .1,
                              fixed_cost = 25)
```

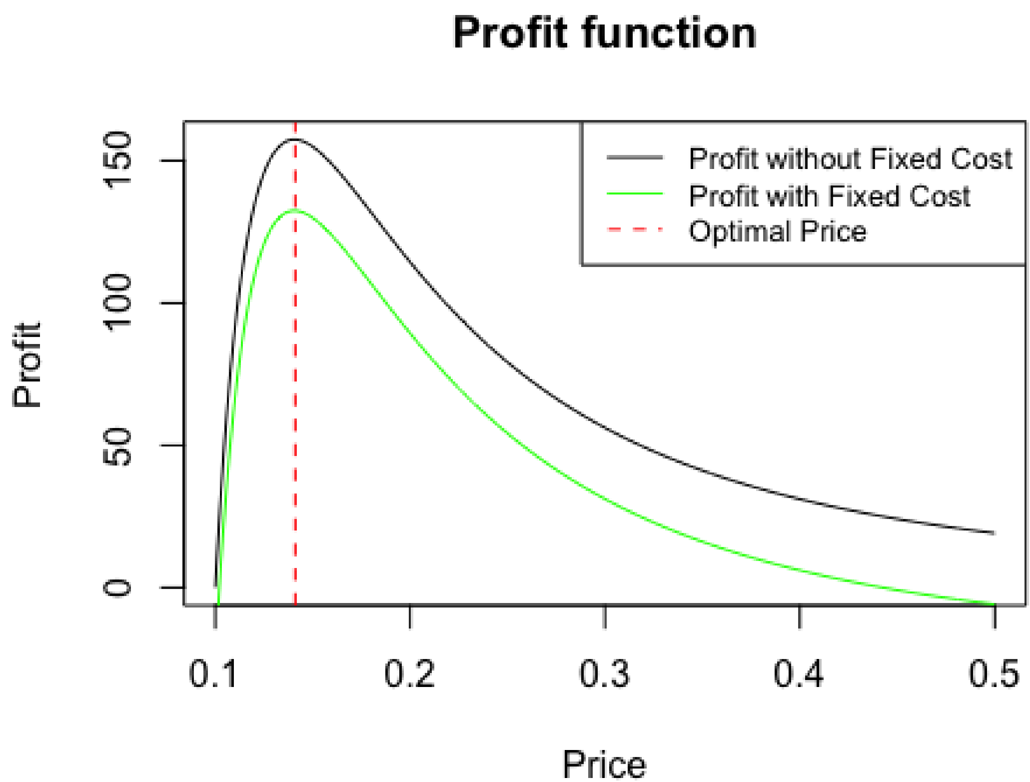
```r
df_profit2 <- data.frame(price_grid_hw, profit_grid2)
head(df_profit2)
```

```
##   price_grid_hw profit_grid2
## 1         0.100   -25.000000
## 2         0.101   -12.804988
## 3         0.102    -1.428401
## 4         0.103     9.182335
## 5         0.104    19.076156
## 6         0.105    28.298634
```

```r
df_profit2[which.max(df_profit2$profit_grid2),]
```

```
##    price_grid_hw profit_grid2
## 42         0.141     132.3952
```

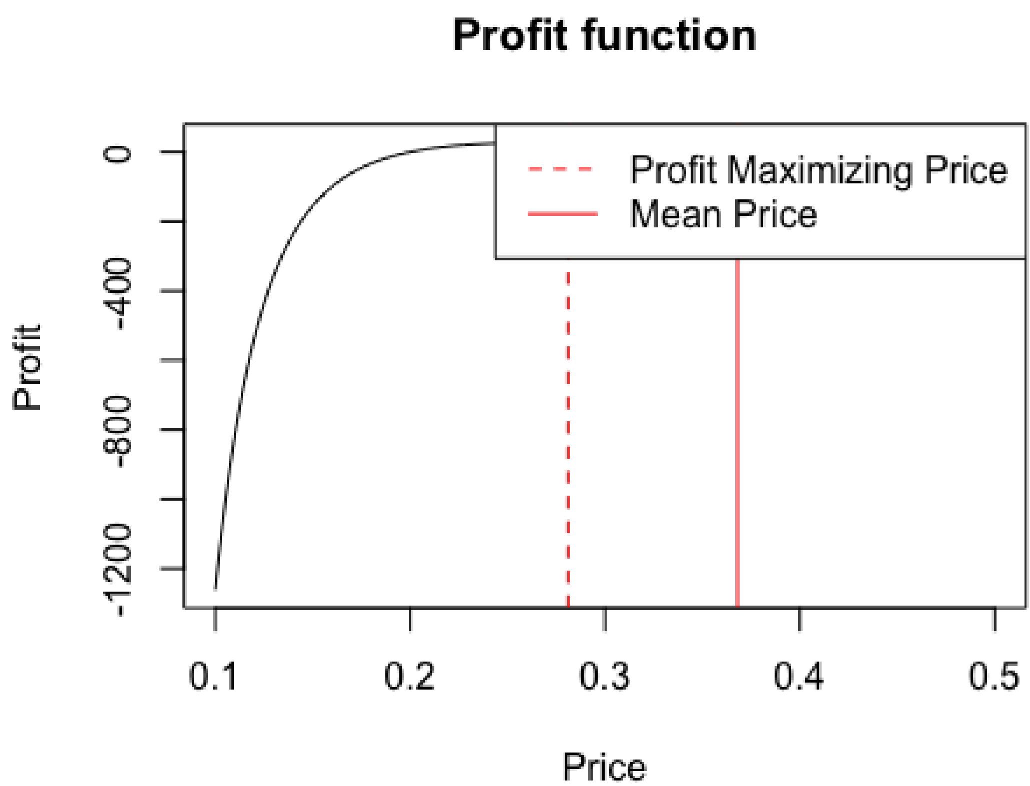
```r
## Plot the optimal price on the profit function graph
plot(x = price_grid_hw,
     y = profit_grid,
     xlab = "Price",
     ylab = "Profit",
     type ="l",
     main = "Profit function")

lines(x = price_grid_hw,
      y = profit_grid2,
      col = "green")

abline(v = 0.141, ## x-intercept of the vertical line #profit maximizing price
       lty = 2, ## linetype. 2 = dash
       col = 'red')

legend("topright",
       legend = c("Profit without Fixed Cost", "Profit with Fixed Cost", "Optimal Price"),
       col = c("black", "green", "red"),
       lty = c(1, 1, 2),
       cex = 0.8)
```



question 5

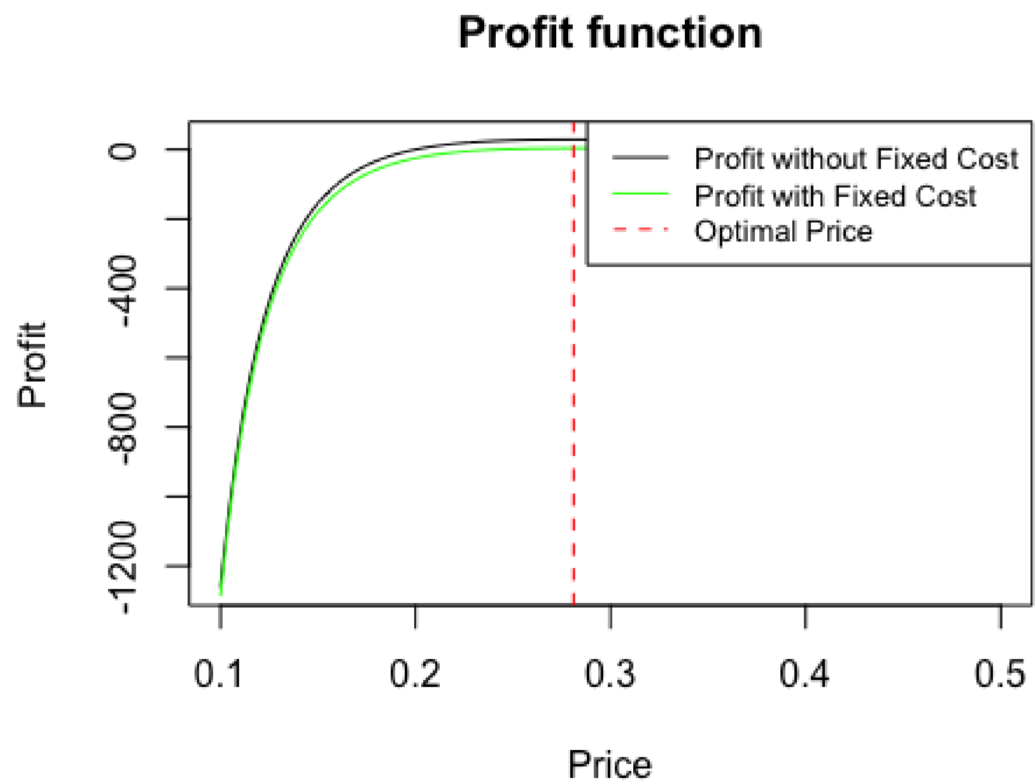
```
profit_grid3 = profit_oj_full(P = price_grid_hw,
                             Product1 = 1,
                             Store2 = 1,
                             unit_cost = .2,
                             fixed_cost = 0)

df_profit3 <- data.frame(price_grid_hw, profit_grid3)
df_profit3[which.max(df_profit3$profit_grid3),]

##     price_grid_hw profit_grid3
## 182         0.281     28.52124

plot(x = price_grid_hw,
     y = profit_grid3,
     xlab = "Price",
     ylab = "Profit",
     type ="l",
     main = "Profit function")

abline(v = 0.281, ## x-intercept of the vertical line #profit maximizing
price
       lty = 2, ## linetype. 2 = dash
```

```r
       col = 'red')

abline(v=MeanPrice_p1s2, col="red")

legend("topright",                        # Location of the legend
       legend = c("Profit Maximizing Price", "Mean Price"), # Labels
       col = "red",                       # Color for both lines
       lty = c(2, 1),                     # Dashed line for profit-maximizing
price, solid line for mean price
       pt.cex = 1.5)
```




```r
MeanPrice_p1s2 <- 0.368
RecommendedPrice <- 0.281

# profit for the mean price (0.368)
mean_profit_row <- df_profit3[which(df_profit3$price_grid_hw ==
MeanPrice_p1s2), ]
mean_profit <- mean_profit_row$profit_grid

# Calculate the profit difference
recommended_profit = 28.521
profit_difference <- recommended_profit - mean_profit
```

```r
# Output the profit difference
print(profit_difference)

## [1] 5.284265

profit_grid4 = profit_oj_full(P = price_grid_hw,
                              Product1 = 1,
                              Store2 = 1,
                              unit_cost = .2,
                              fixed_cost = 25)

df_profit4 <- data.frame(price_grid_hw, profit_grid4)

df_profit4[which.max(df_profit4$profit_grid4),]

##     price_grid_hw profit_grid4
## 182         0.281     3.521239

## Plot the optimal price on the profit function graph
plot(x = price_grid_hw,
     y = profit_grid3,
     xlab = "Price",
     ylab = "Profit",
     type ="l",
     main = "Profit function")

lines(x = price_grid_hw,
      y = profit_grid4,
      col = "green")

abline(v = 0.281, ## x-intercept of the vertical line #profit maximizing
price
       lty = 2, ## linetype. 2 = dash
       col = 'red')

legend("topright",
       legend = c("Profit without Fixed Cost", "Profit with Fixed Cost",
"Optimal Price"),
       col = c("black", "green", "red"),
       lty = c(1, 1, 2),
       cex = 0.8)
```



Optimal price is a function of price elasticity and marginal cost, and when the unit cost doubles, optimal price doubles as well. If the retailer is insistent on keeping the price same, as a manager, we will emphasis the risk of missing out on maximizing profits and will perhaps, show the above data driven approach, to drive home the point.