Assignment 1 - SafeBabies Analytics

Fabrizio Fiorini

11/4/2021

INTRODUCTION

SafeBabies is a large company who is producing car seats for babies and toddlers. They sell their products all over the US and abroad. The management team has hired you as a Business Analytics consultant to help them maximize their profit.

Your primary task is to determine:

1- The optimal price for selling the car seats at those stores where the shelve location is “good” (i.e., the product is highly visible)?

2- The optimal price for selling the car seats at those stores where the shelve location is “bad” (i.e., the product is poorly visible)?

For part 1, and 2, you have been told that the cost of producing each car seat is $55.0.

For part 3 (see below, you need to vary the cost of the production from $40 to $85 in $5 increaments.

3- Plot the optimal price for selling the car seats at those stores where the shelve location is “good” and separately for those stores where the shelve location is “bad” when varying the production costs from $40 to $85 (in $5 increments).

1.SETTING AND DATA EXPLORATION

First, let us load the necessary libraries. We also need to load the “Carseats” dataset and select only the three attributes we will need for our analysis: Sales (in thousands) of each location, Price charged at each location, and ShelveLoc that indicates the quality of the shelving location at the store. Our subset will be called “SafeBabies”.

```
#libraries
library(ISLR)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
#Loading dataset
SafeBabies <- Carseats %>% select("Sales", "Price", "ShelveLoc")
```

As a common practice for this type of analysis, we are going to performe some data exploration so that we have a better understanding of the data we are working on.

```
#data exploration on the dataset
head(SafeBabies)
```

```
##   Sales Price ShelveLoc
## 1  9.50   120       Bad
## 2 11.22    83      Good
## 3 10.06    80    Medium
## 4  7.40    97    Medium
## 5  4.15   128       Bad
## 6 10.81    72       Bad
```

```
str(SafeBabies)
```

```
## 'data.frame':    400 obs. of  3 variables:
##  $ Sales    : num  9.5 11.22 10.06 7.4 4.15 ...
##  $ Price    : num  120 83 80 97 128 72 108 120 124 124 ...
##  $ ShelveLoc: Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3 ...
```

```
levels(SafeBabies$ShelveLoc)
```

```
## [1] "Bad"    "Good"   "Medium"
```

```
anyNA(SafeBabies)
```

```
## [1] FALSE
```

```
colMeans(is.na(SafeBabies))
```

```
##     Sales     Price ShelveLoc
##         0         0         0
```

```
SafeBabies %>% count(ShelveLoc, sort = TRUE)
```

```
##   ShelveLoc   n
## 1    Medium 219
## 2       Bad  96
## 3      Good  85
```

```
SafeBabies %>% group_by(ShelveLoc) %>% summarize(
  "Store Avg Sales" = round(mean(Sales), digits=1),
  "Avg Price" = round(mean(Price), digits=1))
```

```
## # A tibble: 3 x 3
##   ShelveLoc `Store Avg Sales` `Avg Price`
## * <fct>                 <dbl>       <dbl>
## 1 Bad                     5.5         114.
## 2 Good                   10.2         118.
## 3 Medium                  7.3         116.
```

```
summary(SafeBabies)
```

```
##      Sales            Price        ShelveLoc
##  Min.   : 0.000   Min.   : 24.0   Bad   : 96
##  1st Qu.: 5.390   1st Qu.:100.0   Good  : 85
##  Median : 7.490   Median :117.0   Medium:219
##  Mean   : 7.496   Mean   :115.8
##  3rd Qu.: 9.320   3rd Qu.:131.0
##  Max.   :16.270   Max.   :191.0
```

Let us begin with the head() function that shows us the first 6 observations of the SafeBabies dataset. As we can see, we have the number of car seats sold for each store, as well as the price and the type of shelve location.

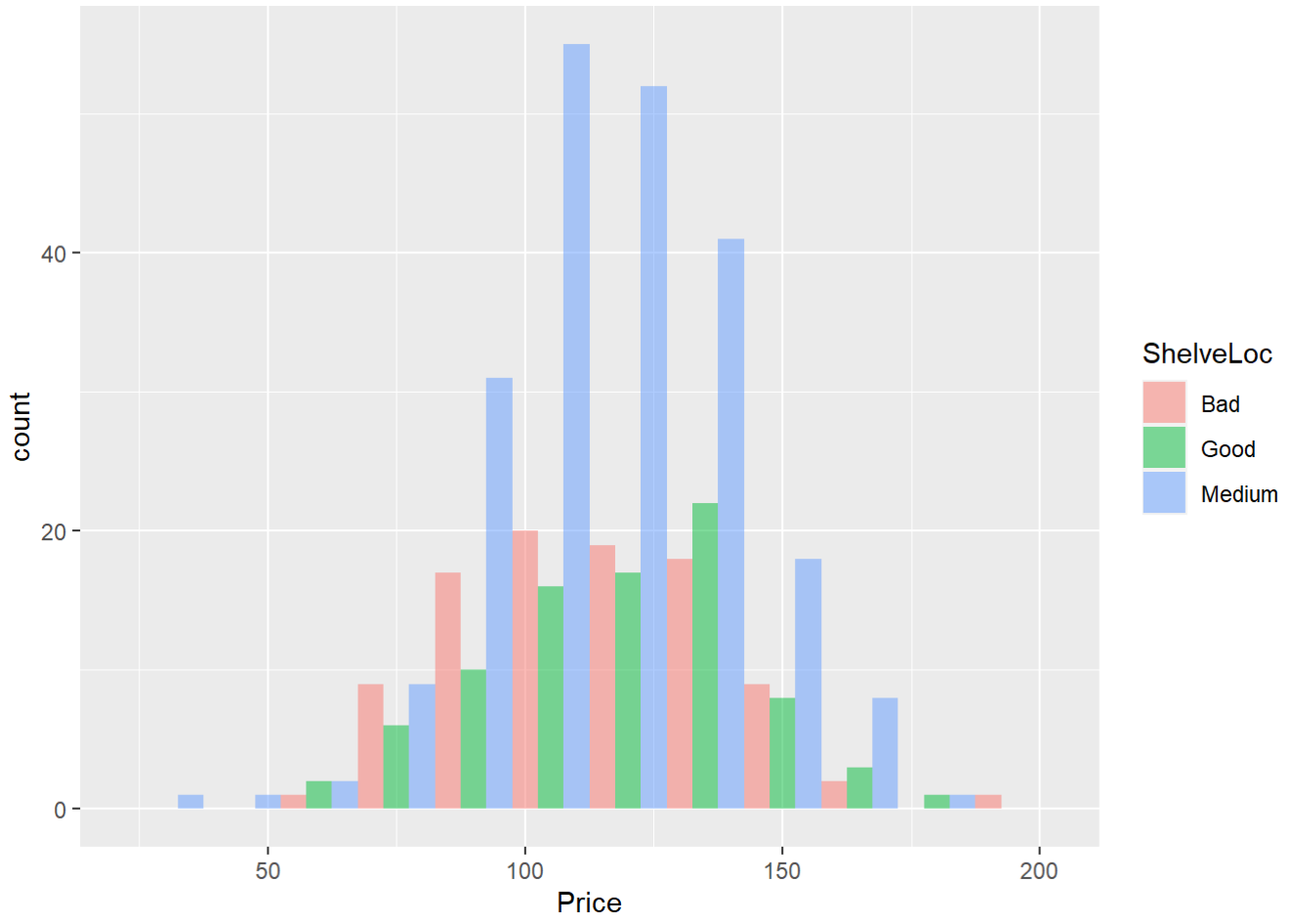
With str() function we know that there are 400 observations (stores) and the three attributes we selected, of which the first two are numerical variables and the third one is a factor containing 3 levels. Let us check this with the level() function: store location can be Good, Medium or Bad.

Then, we want to check for possible missing value. We can use either anyNA() or colMeans() combined with is.na() function. In both cases, there are not missing value.

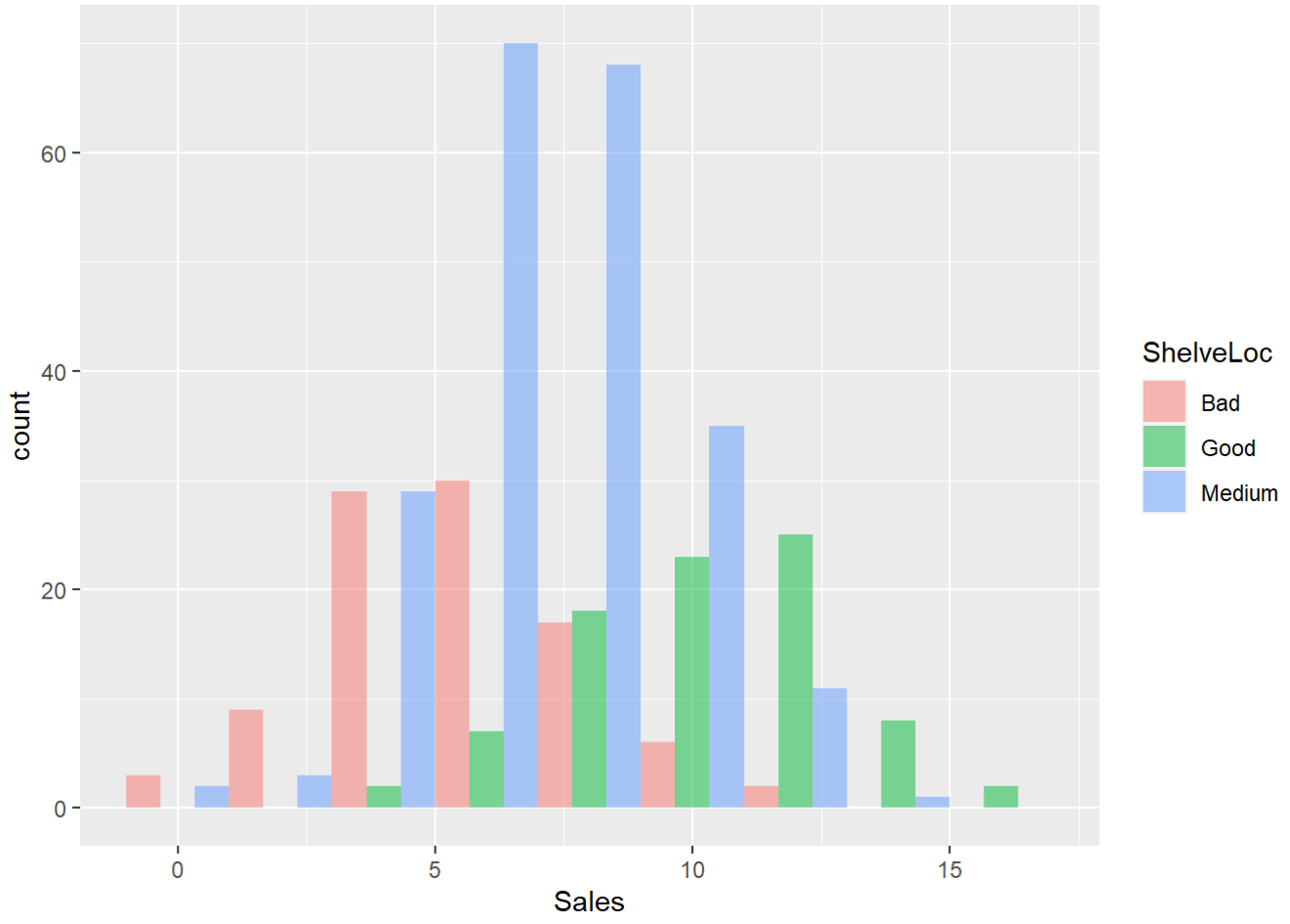
Now, we need to dive deep into the ShelveLoc types. Thanks to the count() function we can see that most of the stores have a Medium location, while we have 96 and 85 stores considered, respectively, as Bad and Good. From the table we created, we know the store average sales and the average price charged according with the attribute ShelveLoc. For example, car seats in store where they benefit from a Good location are sold on average more than twice the amount sold in stores with Bad location (10,200 versus 5,500), and the average price is slightly higher (117.9 versus 114.3).

Finally, much of the information we retrieved and more can be obtained by the summary() function. For example, from the output we see that considering all the 400 observations, the average sales amount is about 7,500 (minimum of 0 and maximum of more than 16,200) and the average price is 115.8 dollars (minimum of 24 and maximum of 191).

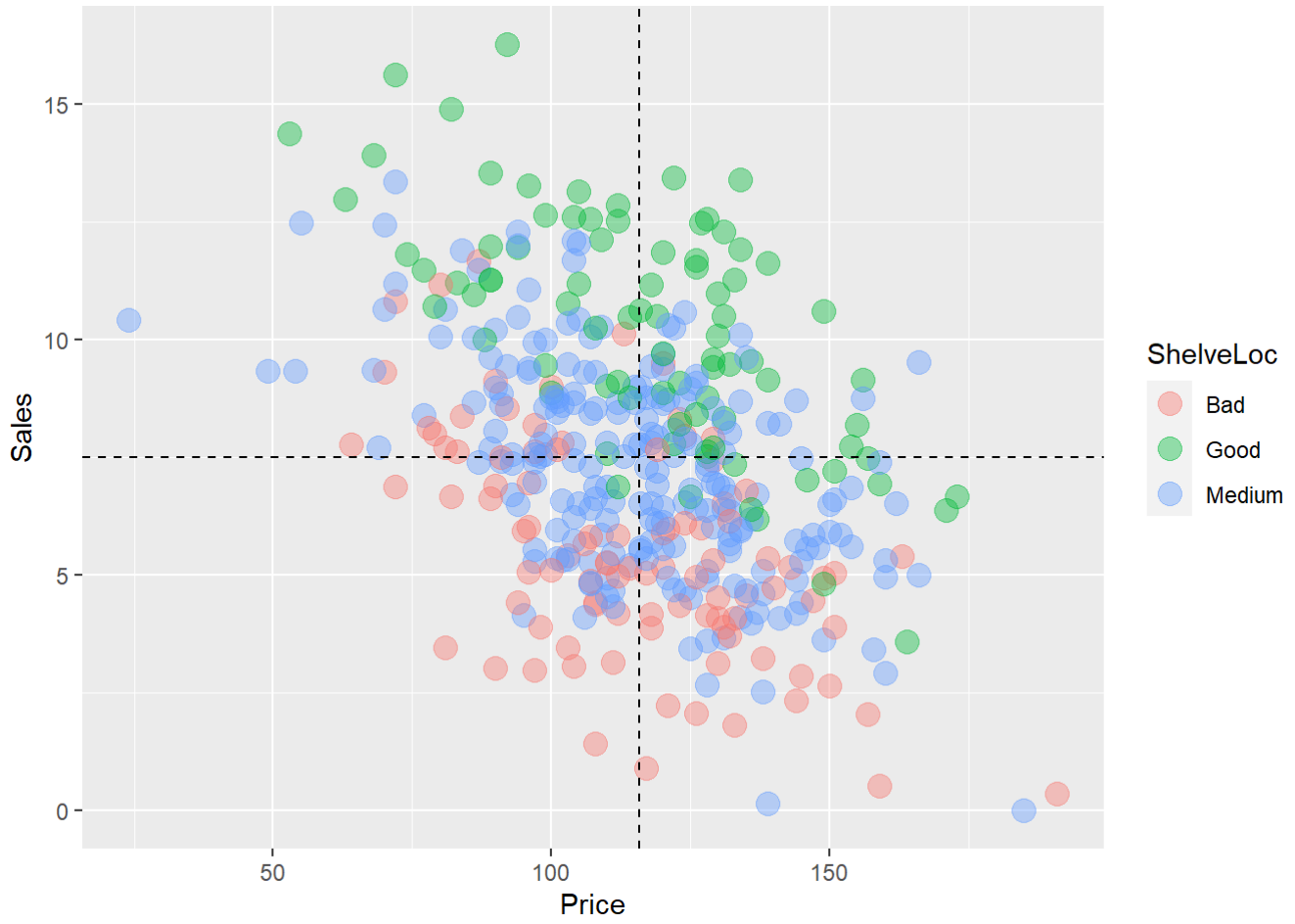
```
#plotting
SafeBabies %>% ggplot(aes(Price, fill=ShelveLoc)) + geom_histogram(binwidth = 15, alpha = 0.5
, position = "dodge")
```




```
SafeBabies %>% ggplot(aes(Sales, fill=ShelveLoc)) + geom_histogram(binwidth = 2, alpha = 0.5,
position = "dodge")
```



```
SafeBabies %>% ggplot(aes(x=Price, y=Sales, fill=ShelveLoc, color=ShelveLoc)) +
  geom_point(shape = 21, size = 4, alpha = 0.4) +
  geom_vline(xintercept = mean(SafeBabies$Price), linetype = "dashed", color = "black") +
  geom_hline(yintercept = mean(SafeBabies$Sales), linetype = "dashed", color = "black")
```



By plotting the data, we can visualize the impact that the location of our product in different stores makes. While in stores where the car seats are kept in a bad location, the price charged is usually in the range 80-130, in stores where they are kept in good location, the price is usually within the range 110-140. The highest prices are charged indifferently in stores with bad, medium and good location, as we can see from the far right end of the histogram.

The Sales histogram tells us that there is a greater difference between “Bad” and “Good” stores: the first ones have lower sales, generally around 3 to 5 thousands seats sold, while the second ones generate sales for 10 to 12 thousands. Also at the extremities of the chart we have an exceptional performance (around 15 thousands seats sold) in “Good” locations and poor performance (less than a thousands) from “Bad” stores.

Finally, in the scatter plot we have a vertical and an horizontal lines indicating respectively the overall average of Price and Sales for the entire data, and we can see the position of the individual datapoints with a different color according to the attribute ShelveLoc. Clearly, there are more green stores in the upper-right section, indicating a higher-than-average Price and higher-than-average Sales. The opposite can be said for the “Bad” stores. Notice a red outlier in the bottom right section, meaning that a store applies a very high price to the car seats and push down the sales.

2.DATA PREPARATION

In order to provide the insight that the management team asked, we are going to create a simple linear regression model and then solve the problem aiming at maximizing the total profit. For our analysis, it can be helpful to extract from the subset two different sets, one containing only the observations for Good location and

one containing only stores where location is Bad.

```
#subsetting data
GoodLoc <- SafeBabies %>% filter(ShelveLoc == "Good")
str(GoodLoc)
```

```
## 'data.frame':    85 obs. of  3 variables:
##  $ Sales    : num  11.2 11.8 12 11 11.2 ...
##  $ Price    : num  83 120 94 86 118 110 131 68 109 82 ...
##  $ ShelveLoc: Factor w/ 3 levels "Bad","Good","Medium": 2 2 2 2 2 2 2 2 2 2 ...
```

```
summary(GoodLoc)
```

```
##      Sales           Price        ShelveLoc
##  Min.   : 3.58   Min.   : 53.0   Bad   : 0
##  1st Qu.: 8.33   1st Qu.:103.0   Good  :85
##  Median :10.50   Median :122.0   Medium: 0
##  Mean   :10.21   Mean   :117.9
##  3rd Qu.:11.96   3rd Qu.:132.0
##  Max.   :16.27   Max.   :173.0
```

```
BadLoc <- SafeBabies %>% filter(ShelveLoc == "Bad")
str(BadLoc)
```

```
## 'data.frame':    96 obs. of  3 variables:
##  $ Sales    : num  9.5 4.15 10.81 9.01 10.14 ...
##  $ Price    : num  120 128 72 100 113 97 102 138 126 124 ...
##  $ ShelveLoc: Factor w/ 3 levels "Bad","Good","Medium": 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(BadLoc)
```

```
##      Sales            Price         ShelveLoc
##  Min.   : 0.370   Min.   : 64.00   Bad   :96
##  1st Qu.: 4.053   1st Qu.: 96.75   Good  : 0
##  Median : 5.210   Median :113.50   Medium: 0
##  Mean   : 5.523   Mean   :114.27
##  3rd Qu.: 7.463   3rd Qu.:130.00
##  Max.   :11.670   Max.   :191.00
```

As we saw earlier, we have 85 stores with Good ShelveLoc and 96 stores with Bad ShelveLoc. From the summary we can check that the two sets contains only observation of either Good or Bad locations. Moreover, we can check the values of average sales and prices we measured during the data exploration phase. Here, we can also see the minimum and maximum Sales and Price for both Sales and Price.

Our goal is to find the optimal prices that provide the maximum profit. Profit can be expressed as unit profit on the sale by calculating the difference between price and unit cost, or as store profit by multiplying the unit profit for the sales amount. Therefore, we need to use the production cost which is set to 55 dollars and is not included in the initial dataset.

```
#storing production cost value
ProdCost = 55
```

3.1.LINEAR REGRESSION MODEL FOR GOOD

As we said, we have Sales, Price, ProdCost, and we also know how to measure profit. What we still do not know is the relationship between Sales and Price. Therefore, here we create a simple linear regression model for both Good and Bad sets. When we know how to express Sales as a function of the Price, we will be able to calculate the profit.

```
#simple regression model
set.seed(12)
GoodLocReg <- GoodLoc[,-3]
lregrmodel1.G <- lm(Sales ~ Price, data = GoodLocReg)
summary(lregrmodel1.G)
```

```
##
## Call:
## lm(formula = Sales ~ Price, data = GoodLocReg)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.721 -1.351 -0.098  1.483  4.353
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.968864   0.988008  18.187  < 2e-16 ***
## Price       -0.065785   0.008199  -8.023 5.85e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.888 on 83 degrees of freedom
## Multiple R-squared:  0.4368, Adjusted R-squared:   0.43
## F-statistic: 64.37 on 1 and 83 DF,  p-value: 5.848e-12
```

As we could expect, our model reached a 0.4368 R-squared since we are trying to predict Sales with only one variable, Price. Our model is clearly underfitting. However, we now have the relationship between the two that is:

Sales = 17.9689 - 0.0658*Price

Notice that Sales is negatively correlated to the Price, when the latter increases, the former is penalized. Specifically, for every dollar that the price increases, sales goes down by almost 66.

Let us now try to improve the model by partitioning the Good set in training set and test set, providing for them respectively 65% and 35%.

```
#data partitioning for the model
set.seed(123)
GoodLocReg <- GoodLoc[,-3]
subset.G = createDataPartition(GoodLocReg$Sales, p = 0.65, list = FALSE)
GoodTrain = GoodLocReg[subset.G, ]
GoodTest = GoodLocReg[-subset.G, ]
str(GoodTrain)
```

```
## 'data.frame':    57 obs. of  2 variables:
##  $ Sales: num  12 11.2 12.3 13.9 13.6 ...
##  $ Price: num  94 118 131 68 89 137 128 134 99 128 ...
```

```
str(GoodTest)
```

```
## 'data.frame':    28 obs. of  2 variables:
##  $ Sales: num  11.22 11.85 10.96 7.58 12.13 ...
##  $ Price: num  83 120 86 110 109 82 131 100 149 104 ...
```

We see that the training set contains 57 observations while the test set only 28. Now, we create a new regression model that is trained only with the training set and then apply it to the test set.

```
#creating a linear regression model for "Good" dataset
set.seed(100)
lregrmodel2.G <- lm(Sales ~ Price, data = GoodTrain)
summary(lregrmodel2.G)
```

```
##
## Call:
## lm(formula = Sales ~ Price, data = GoodTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7590 -1.5390 -0.0904  1.5254  4.3151
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.00808    1.38613   12.99  < 2e-16 ***
## Price       -0.06580    0.01152   -5.71 4.71e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.994 on 55 degrees of freedom
## Multiple R-squared:  0.3722, Adjusted R-squared:  0.3607
## F-statistic:  32.6 on 1 and 55 DF,  p-value: 4.711e-07
```

```
predictions.G <- predict(lregrmodel2.G, newdata = GoodTest)
error.G <- sqrt((sum((GoodTest$Sales - predictions.G)^2))/nrow(GoodTest))
error.G
```

```
## [1] 1.663784
```

```
Model.G = predictions.G
Actual.G = GoodTest$Sales
Delta.G = (Actual.G - Model.G)/Actual.G
GoodComparison <- cbind(Model.G, Actual.G, Delta.G)
GoodComparison[1:10,]
```

```
##      Model.G Actual.G    Delta.G
## 2  12.547042    11.22 -0.1182747
## 8  10.112601    11.85  0.1466159
## 14 12.349655    10.96 -0.1267933
## 17 10.770558     7.58 -0.4209180
## 22 10.836354    12.13  0.1066485
## 26 12.612838    14.90  0.1535008
## 27  9.388849     8.33 -0.1271127
## 37 11.428515     8.89 -0.2855473
## 50  8.204526    10.61  0.2267176
## 74 11.165332    12.61  0.1145652
```

```
min(Delta.G)
```

```
## [1] -1.016087
```

```
max(Delta.G)
```

```
## [1] 0.2267176
```

Due to the extremely small amount of data, the model performed worst in terms of R-squared: the benefits coming from the training phase is offset by the fact that with fewer observation the model is not able to learn the trend of the data. Nonetheless, the relationship between Sales and Price is the following:

Sales = 18.0081 - 0.0658*Price

From the output we can see a table showing, for the first 10 observations, the model prediction, the actual value contained in the test set, and the percentage variation between the two values. The worst predictions were -101% and +23%.

However, considering the poor performance of this model, for the next steps we will use the function generated from the linear regression analysis run on the entire dataset.

Recall that the store profit is equal to:

StoreProfit = UnitProfit * Sales

StoreProfit = (Price - ProdCost) * Sales

Now we can substitute Sales with the equation we obtained from the regression analysis so that the only unknown variable is Price. With a little bit of maths we calculate the following equation for the Profit:

StoreProfit = a*Price^2* + *b*Price + c

When we plot this function we get a parabola in which the Profit initially increases for small prices, reaches a maximum point, and then starts to decrease for high prices. We can find the maximum point by calculating the derivative of the function and set it equal to zero.

2a*Price + b = 0

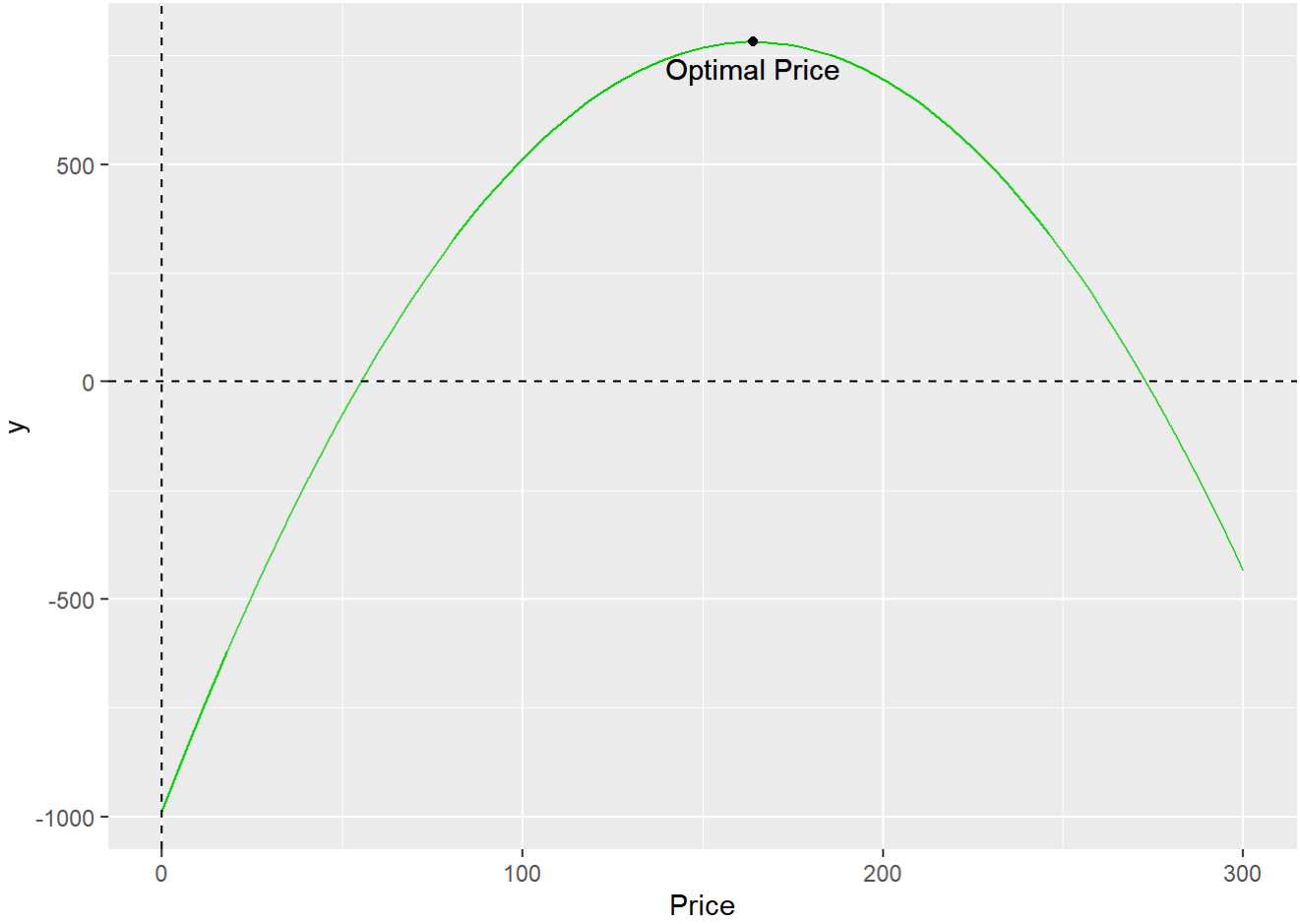
The optimal price is:

Price = - b/2a

```
#Finding maximum point of profit function
Price.G <- GoodLocReg$Price
StoreProfit.G = (Price.G - ProdCost) * (17.969 - 0.0658*Price.G)
StoreProfit.G <- -0.0658*Price.G^2 + 21.588*Price.G - 988.295
OptPrice.G = 21.588/0.1316
OptSales.G = 17.969 - 0.0658*OptPrice.G
equation.G <- function(Price){-0.0658*Price^2 + 21.588*Price - 988.295}
print(cbind(OptPrice.G, OptSales.G, StoreProfit.G = equation.G(OptPrice.G)))
```

```
##      OptPrice.G OptSales.G StoreProfit.G
## [1,]   164.0426      7.175      782.3803
```

```
#plotting the curve and the optimal price
ggplot(data.frame(Price=c(0, 300)), aes(x=Price)) +
  stat_function(fun=equation.G, color = "green3") +
  geom_vline(xintercept = 0, linetype = "dashed", color = "black") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  geom_point(aes(x=OptPrice.G, y=equation.G(OptPrice.G))) +
  geom_text(aes(x=OptPrice.G, y=equation.G(OptPrice.G)-60), label = "Optimal Price", color =
"black")
```



The optimal price for stores where car seats are in a Good location on the shelves is equal to 164 dollars. It would generate sales for more than 7,100 and a total profit per store of 782,380 dollars.

3.2.LINEAR REGRESSION MODEL FOR BAD

Let us now focus on the set of stores where the car seats are located in a bad area on the shelves. We are going to repeat the process as we did for the GoodLoc dataset.

```
#simple regression model
set.seed(100)
BadLocReg <- BadLoc[,-3]
lregrmodel1.B <- lm(Sales ~ Price, data = BadLocReg)
summary(lregrmodel1.B)
```

```
##
## Call:
## lm(formula = Sales ~ Price, data = BadLocReg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4622 -1.0617 -0.2014  1.2050  4.6412
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.832984   0.990317  11.949  < 2e-16 ***
## Price       -0.055220   0.008486  -6.507  3.7e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.967 on 94 degrees of freedom
## Multiple R-squared:  0.3105, Adjusted R-squared:  0.3032
## F-statistic: 42.34 on 1 and 94 DF,  p-value: 3.702e-09
```

This time, our model reached a 0.3105 R-squared and we have underfitting. The relationship between Sales and Price is:

Sales = 11.833 - 0.055*Price

According to this function, for every dollar that the price increases, sales goes down by 55.

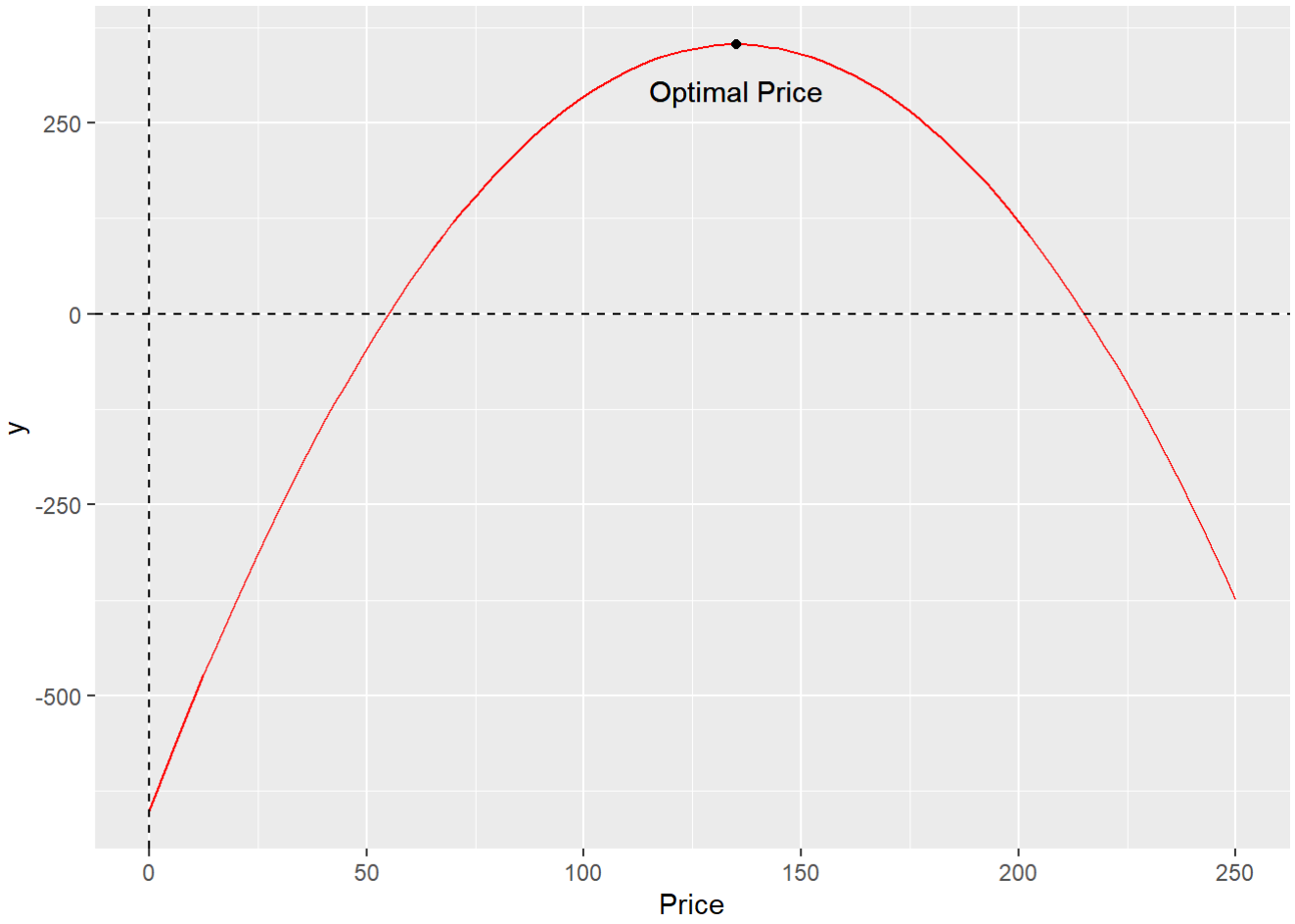
We can find the optimal price for these stores by calculating the derivative of the StoreProfit equation and set it equal to zero. Again, the optimal price is:

Price = - b/2a

```
#Finding maximum point of profit function
Price.B <- BadLocReg$Price
StoreProfit.B = (Price.B - ProdCost) * (11.833 - 0.055*Price.B)
StoreProfit.B = -0.055*Price.B^2 + 14.858*Price.B - 650.815
OptPrice.B = 14.858/0.11
OptSales.B = 11.833 - 0.055*OptPrice.B
equation.B <- function(Price){-0.055*Price^2 + 14.858*Price - 650.815}
print(cbind(OptPrice.B, OptSales.B, StoreProfit.B = equation.B(OptPrice.B)))
```

```
##      OptPrice.B OptSales.B StoreProfit.B
## [1,]   135.0727      4.404      352.6403
```

```
ggplot(data.frame(Price=c(0, 250)), aes(x=Price)) +
  stat_function(fun=equation.B, color = "red") +
  geom_vline(xintercept = 0, linetype = "dashed", color = "black") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  geom_point(aes(x=OptPrice.B, y=equation.B(OptPrice.B))) +
  geom_text(aes(x=OptPrice.B, y=equation.B(OptPrice.B)-60), label = "Optimal Price", color =
"black")
```



Here we have an optimal price equal to 135 dollars that would generate sales for than 4,400 and a store profit of 352,640 dollars.

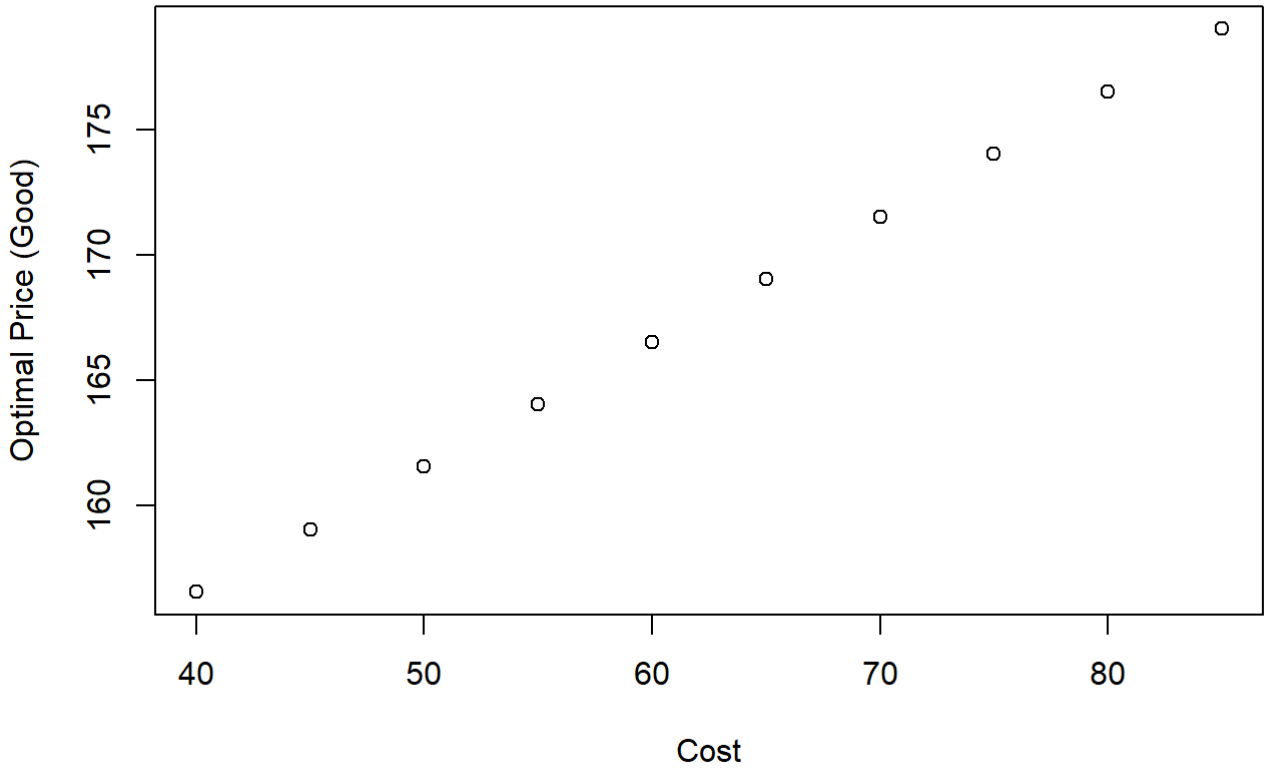
4.PLOTTING OPTIMAL PRICE WHEN PRODUCTION COST CHANGES

For the last section of this assignment, we are lifting the assumption that the production cost is fixed. Let us consider different levels of cost ranging from 40 to 85 dollars with increasing of 5. For both the subset GoodLoc and BadLoc we can now write the equation for StoreProfit as a function of both Price and Cost. We then calculate the derivative for the variable Price and set it equal to zero (remember that when we calculate the derivative of an equation for one variable, the others are considered constants). We now know the relationship between Price and Cost that is given by the equation for Price as a function of Cost. When we substitute the Cost with the different level we set earlier, we get the relative optimal prices for each level of production cost.

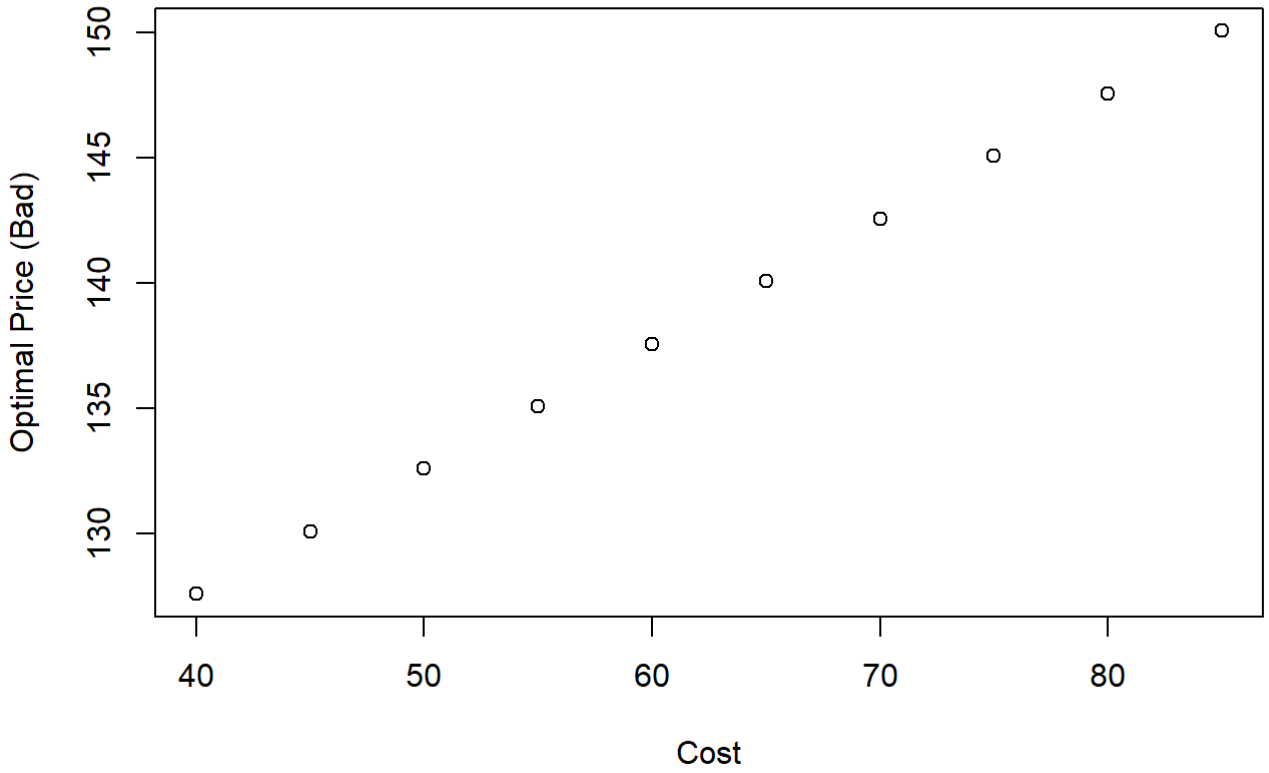
```
#plotting cost-price
Cost = seq(40,85, by = 5)
Cost
```

```
##  [1] 40 45 50 55 60 65 70 75 80 85
```

```
PriceVar.G = 136.543 + 0.5*Cost
plot(Cost, PriceVar.G, ylab = "Optimal Price (Good)")
```




```
PriceVar.B = 107.573 + 0.5*Cost
plot(Cost, PriceVar.B, ylab = "Optimal Price (Bad)")
```



Notice that as we can expect, when production cost increases, also the optimal price increases.