hw04-01

Weijun Zhu

December 4, 2019

Contents

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.2
## 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

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

```r
library(ggplot2)
```

1.

R/Python Project: We wish to examine the relation between price and sales of two editions of a textbook. The two editions are hard cover and paperback. The paperback has a blue cover, and the hardcover version an orange one. Each data item is collected from a different bookstore. Each bookstore can carry only one of either paperback and hardcover versions of the text. The data contains the amount of weekly sales from bookstore across country, the price at which the text was sold, and the type(paperback or hardcover).

1a).

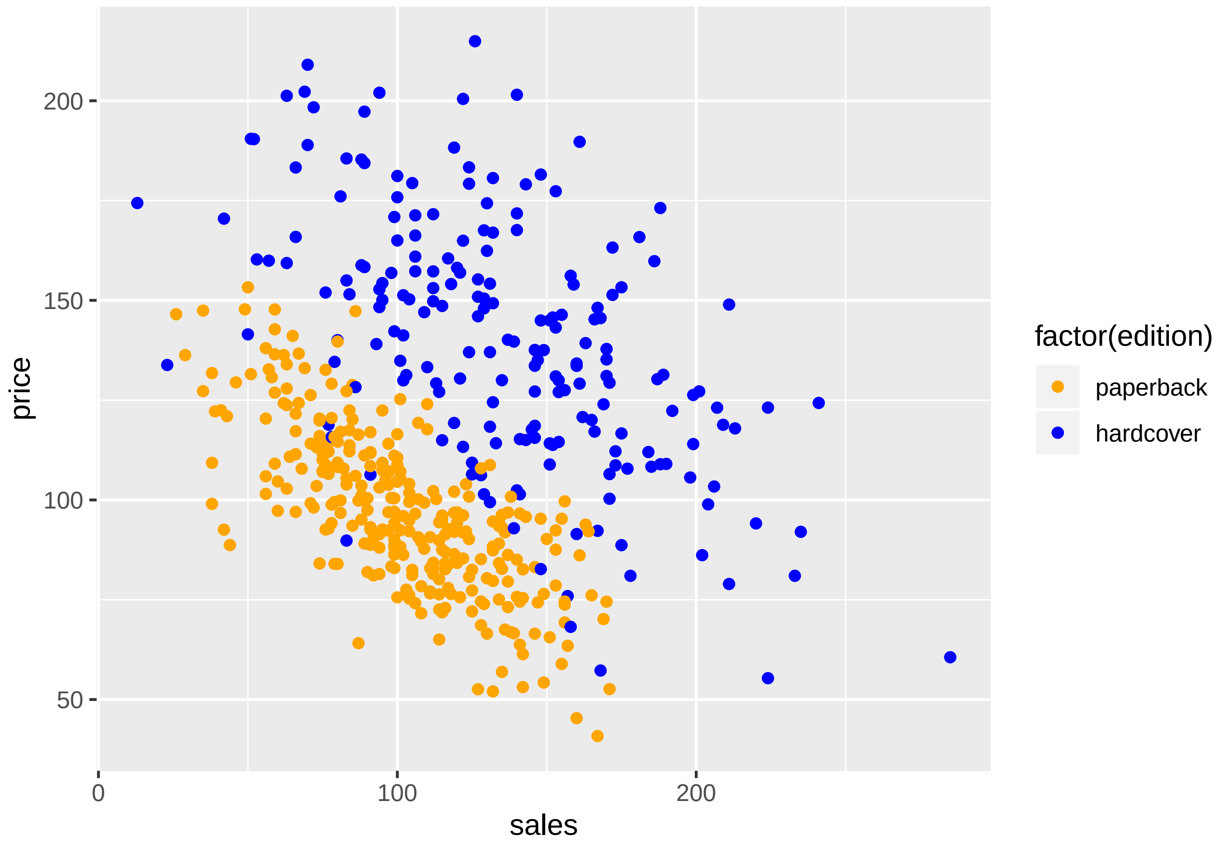
Read the file textbookSales.csv and put it in the data frame called priceData. Plot the scatter plot of the data, coloring paperback points blue and hardcover points orange.

```
priceData <- read.csv('./data/textbookSales.csv', header = T)
```

```
head(priceData)
```

```
##    price sales   edition
## 1 117.46    84 paperback
## 2 111.73    91 paperback
## 3  96.73    81 paperback
## 4 114.86    78 paperback
## 5  84.23   120 paperback
## 6  90.01    99 paperback
```

```
graph1 <- ggplot(priceData, aes(x = sales, y = price)) +
  geom_point(aes(color = factor(edition))) +
  scale_color_manual(breaks = c('paperback','hardcover'), values = c('blue','orange'))
graph1
```



1b).

Assuming the relationship between price and sales is linear for both editions, set up a linear model with sales as the target (response) variable, and price and edition as the independent variables. Call the linear models modelA. Print a summary of the model.

```
# # The 1st way to factorize:
# priceData$edition <- as.factor(priceData$edition)
# priceData$edition <- as.numeric(priceData$edition)
# # "2" represents "paperback", and "1" represents "hardcover"

# The 2nd way to factorize:
priceData$edition <- factor(priceData$edition,
                            levels=c("hardcover","paperback"), labels=c(1,2))
priceData$edition <- as.numeric(priceData$edition)
```

```
modelA <- lm(sales ~ price + edition, data=priceData)
summary(modelA)
```

```
##
## Call:
## lm(formula = sales ~ price + edition, data = priceData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -117.630  -15.838   -0.513   16.960   91.729
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 334.40142   10.16460   32.90   <2e-16 ***
## price        -0.90963    0.05051  -18.01   <2e-16 ***
## edition     -72.06294    3.37896  -21.33   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.78 on 497 degrees of freedom
## Multiple R-squared:  0.4963, Adjusted R-squared:  0.4943
## F-statistic: 244.9 on 2 and 497 DF,  p-value: < 2.2e-16
```

1c).

How much of the variation in sales is determined by this model?

Answer: R-squared looks good. From previous problem, we can see both of price and sales are significant, and the p-value are really small. We can say sales determine this model much.

1d).

What is the relationship between price and sales for the hardcover edition, and what is this relationship for the paperback version? Extract this information from the model you derived in question 1b.

```
# "2" represents "paperback", and "1" represents "hardcover"
hardcover <- priceData %>%
  filter(edition==1)

modelB <- lm(edition ~ price + sales, data=hardcover)
summary(modelB)
```

```
##
## Call:
## lm(formula = edition ~ price + sales, data = hardcover)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.493e-14  6.060e-17  1.307e-16  1.828e-16  4.333e-16
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  1.000e+00  1.015e-15  9.851e+14   <2e-16 ***
## price        1.402e-18  4.818e-18  2.910e-01    0.771
## sales       -1.320e-18  3.425e-18 -3.860e-01    0.700
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.783e-15 on 197 degrees of freedom
## Multiple R-squared:  0.4988, Adjusted R-squared:  0.4938
## F-statistic: 98.04 on 2 and 197 DF,  p-value: < 2.2e-16
```

```r
# "2" represents "paperback", and "1" represents "hardcover"
paperback <- priceData %>%
  filter(edition==2)

modelC <- lm(edition ~ price + sales, data=paperback)
summary(modelC)
```

```
##
## Call:
## lm(formula = edition ~ price + sales, data = paperback)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.375e-13  1.450e-16  4.410e-16  7.460e-16  1.760e-15
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  2.000e+00  5.122e-15  3.904e+14   <2e-16 ***
## price       -2.747e-17  3.320e-17 -8.270e-01    0.409
## sales       -4.729e-18  2.199e-17 -2.150e-01    0.830
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.007e-15 on 297 degrees of freedom
## Multiple R-squared:  0.5006, Adjusted R-squared:  0.4972
## F-statistic: 148.8 on 2 and 297 DF,  p-value: < 2.2e-16
```

Answer: Whatever paperback or hardcover, We can see p-values of both of price and sales are big, and they are not significant. So there are no relationships between price and sales for the hardcover and paperback edition,

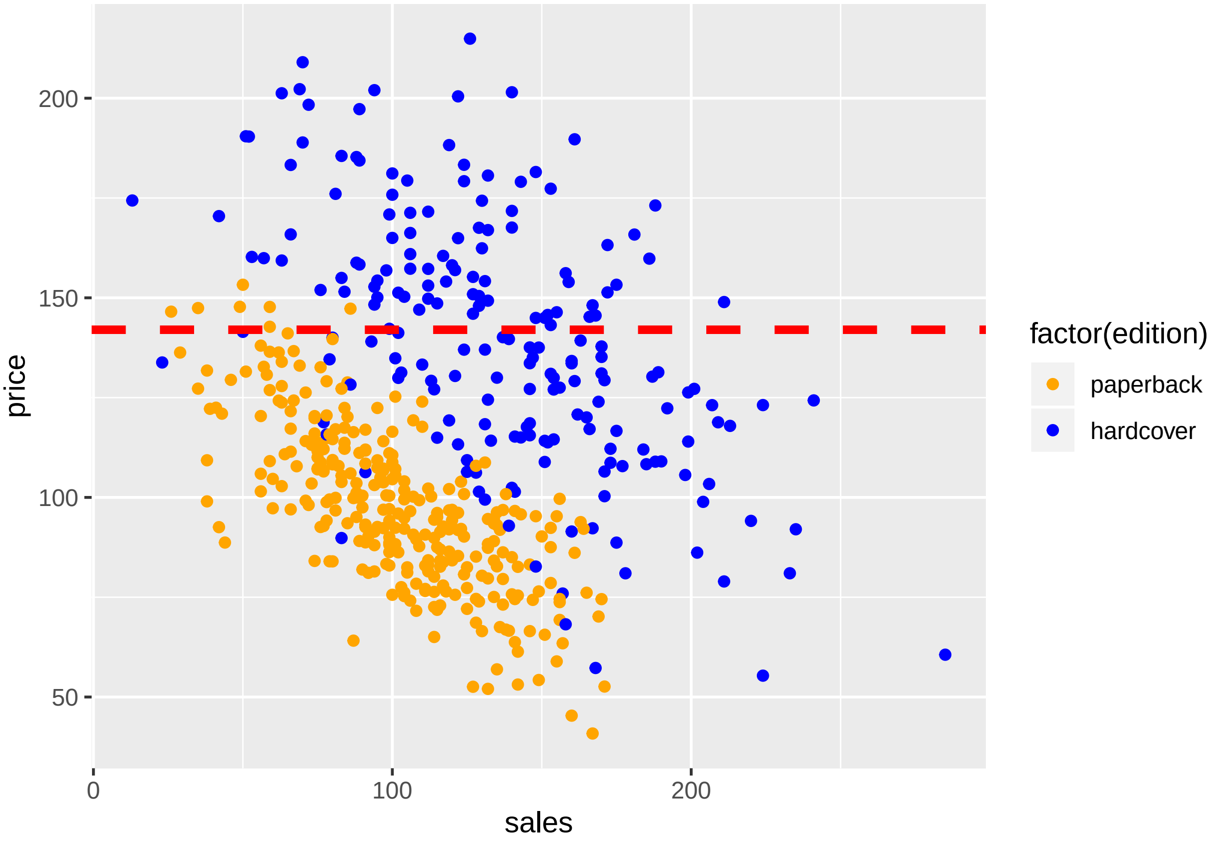
1e).

A bookstore reports sales of 123 copies of the text at the price of 142, but the edition is not reported. According to the Bayes rule which edition is more likely to have been sold by this bookstore? Clearly Justify your answer.

```
table(priceData$edition)
```

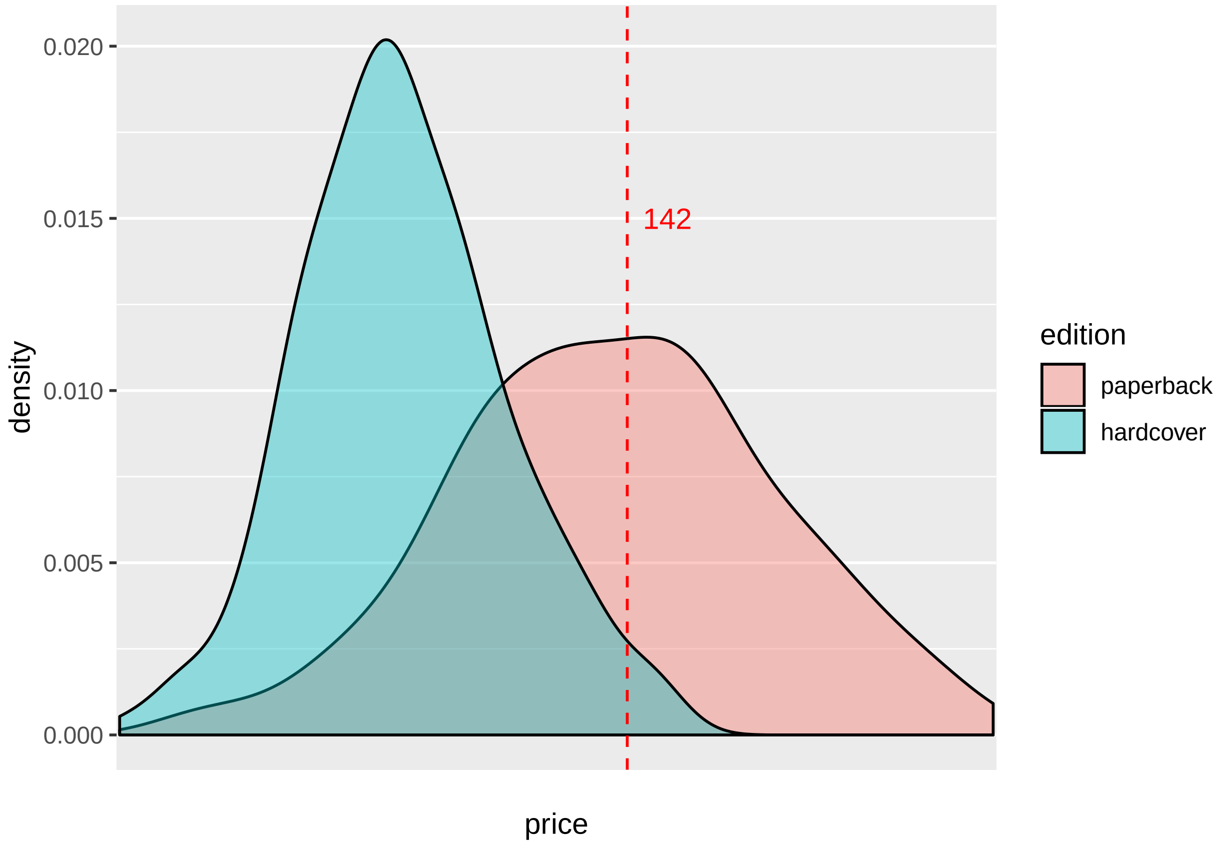
```
##
##   1   2
## 200 300
```

```
# add a line which value of price equal 142
graph1 +
  geom_hline(yintercept=142, linetype="dashed", color="red", size=1.5)
```



We can not get any information from above graph, so we should do further.Then I plot the density graph.

```
# "2" represents "paperback", and "1" represents "hardcover"
# Rename legend labels and change the order of items
ggplot(priceData, aes(x=price, fill=factor(edition)))+
  geom_density(alpha=0.4)+
  # Change the order of legend items: scale_x_discrete()
  # Edit legend title and text labels: scale_fill_discrete()
  scale_x_discrete(limits=c("2", "1"))+
  scale_fill_discrete(name = "edition", labels = c("paperback", "hardcover"))+
  geom_vline(aes(xintercept=142), color="red", linetype="dashed")+
  annotate(geom="text", x=150, y=0.015, label="142",color="red")
```



Conclusion: We can see at price of 142, the more likely is hardcover which is represented by 1 in above graph.

1f).

Now suppose the data were presented to you without the information about which edition was sold. You are to use the Expectation-Maximization technique to separate the set of books. We assume that we know there are two types of books, but pretend we do not know price/sales data corresponds to which edition. The objective is to use the EM algorithm to figure this out. Here is the outline of the EM algorithm in this case. The instructions are for R users; for Python, figure out the equivalent procedure:

1. In R make sure you have installed the EMCluster package (Figure out what the equivalent package is in Python.)
2. Use the init.EM function to initialize the EM process. As the input matrix you should pass the two-column matrix made up of price and sales data of pricedata (so no information about the edition feature is passed to EMcluster functions.) Save the output of init.EM in an object called em1.
3. Using em1 run the emcluster function to estimate the latent variable. Assign the output of the emcluster function to an object called em.
4. Use the assign.class function, and again pass the price and sales columns of pricedata as a matrix. Assign the output to an object called c.

5. Plot the scatter plot of pricedata, price vs sales, but this time color with respect to classes produced by the assign.class function in c.
6. Using the table function compare the classes produced by the EM method and the original classes in the Edition column of pricedata. How many are misclassified? What is the misclassification rate?
7. Run the linear regression model again, but this time instead of edition use the classes pro- duced by the EM algorithm, and stored in c. Compare the slope and intercept for each class, with the slope and intercept for the original classes paperback and hardcover.

Answer: step by step

1. In R make sure you have installed the EMCluster package (Figure out what the equivalent package is in Python.)

```
library(EMCluster)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
##
## Attaching package: 'EMCluster'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

2. Use the init.EM function to initialize the EM process. As the input matrix you should pass the two-column matrix made up of price and sales data of pricedata (so no information about the edition feature is passed to EMcluster functions.) Save the output of init.EM in an object called em1.

```
priceDataEM <- priceData %>%
  # There is a conflict between EMcluster and tidyverse.
  dplyr::select(-edition) # select the columns except the "edition"

em1 <- init.EM(priceDataEM,nclass=2)
```

3. Using em1 run the emcluster function to estimate the latent variable. Assign the output of the emcluster function to an object called em.

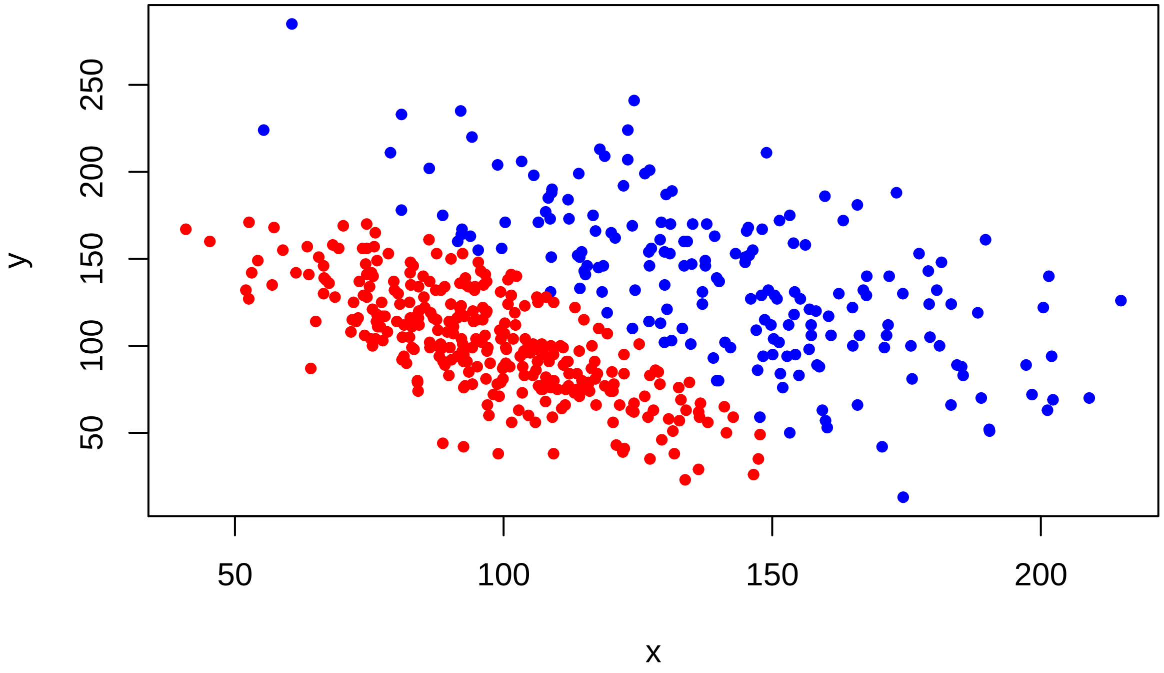
```
em <- emcluster(priceDataEM,em1)
```

4. Use the assign.class function, and again pass the price and sales columns of pricedata as a matrix. Assign the output to an object called c.

```
c<-assign.class(priceDataEM,em)
```

5. Plot the scatter plot of pricedata, price vs sales, but this time color with respect to classes produced by the assign.class function in c.

```
plot(priceDataEM[c$class==1,1],priceDataEM[c$class==1,2],pch=20,col="blue",
          xlab="x",ylab="y",
          xlim=c(min(priceDataEM[,1]),max(priceDataEM[,1])),
          ylim=c(min(priceDataEM[,2]),max(priceDataEM[,2])))
points(priceDataEM[c$class==2,1],priceDataEM[c$class==2,2],pch=20,col="red")
points(priceDataEM[c$class==3,1],priceDataEM[c$class==3,2],pch=20,col="maroon")
```



6. Using the table function compare the classes produced by the EM method and the original classes in the Edition column of pricedata. How many are misclassified? What is the misclassification rate?

```r
print('Original dataset: ')
```

```
## [1] "Original dataset: "
```

```r
table(priceData$edition)
```

```
##
##   1   2
## 200 300
```

```r
cat('\n\n')
```

```r
print('EM method: ')
```

```
## [1] "EM method: "
```

```r
table(c$class)
```

```
##
##   1   2
## 188 312
```

Conclusion: There are 114 values misclassified. The misclassification rate is 0.228.

7. Run the linear regression model again, but this time instead of edition use the classes pro- duced by the EM algorithm, and stored in c. Compare the slope and intercept for each class, with the slope and intercept for the original classes paperback and hardcover.

```r
priceDataEM['edition'] <- c$class
modelC <- lm(c$class ~ sales + price, data=priceDataEM)
summary(modelC)
```

```
##
## Call:
## lm(formula = c$class ~ sales + price, data = priceDataEM)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62472 -0.14341  0.02952  0.16560  0.60507
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.7138511  0.0545084   68.13   <2e-16 ***
## sales       -0.0067670  0.0002699  -25.08   <2e-16 ***
## price       -0.0114758  0.0003329  -34.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2396 on 497 degrees of freedom
## Multiple R-squared:  0.7568, Adjusted R-squared:  0.7558
## F-statistic: 773.2 on 2 and 497 DF,  p-value: < 2.2e-16
```

```r
hardcoverEM <- priceDataEM %>%
  filter(edition==1)

modelD <- lm(edition ~ price + sales, data=hardcoverEM)
summary(modelD)
```

```
## 
## Call:
## lm(formula = edition ~ price + sales, data = hardcoverEM)
## 
## Residuals:
##        Min         1Q     Median         3Q        Max 
## -4.931e-16 -2.237e-16 -1.474e-16 -6.880e-17  2.717e-14 
## 
## Coefficients:
##               Estimate Std. Error   t value Pr(>|t|)    
## (Intercept)  1.000e+00  1.395e-15  7.17e+14   <2e-16 ***
## price       -2.833e-18  6.455e-18 -4.39e-01    0.661    
## sales        7.811e-19  4.353e-18  1.79e-01    0.858    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.006e-15 on 185 degrees of freedom
## Multiple R-squared:  0.5007, Adjusted R-squared:  0.4953 
## F-statistic: 92.76 on 2 and 185 DF,  p-value: < 2.2e-16
```

```r
paperbackEM <- priceDataEM %>%
  filter(edition==2)

modelD <- lm(edition ~ price + sales, data=paperbackEM)
summary(modelD)
```

```
## 
## Call:
## lm(formula = edition ~ price + sales, data = paperbackEM)
## 
## Residuals:
##        Min         1Q     Median         3Q        Max 
## -7.012e-14  6.600e-17  2.260e-16  3.870e-16  8.220e-16 
## 
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)    
## (Intercept)  2.000e+00  2.596e-15  7.704e+14   <2e-16 ***
## price       -1.469e-17  1.692e-17 -8.680e-01    0.386    
## sales       -2.791e-18  1.096e-17 -2.550e-01    0.799    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4.003e-15 on 309 degrees of freedom
## Multiple R-squared:  0.5004, Adjusted R-squared:  0.4971 
## F-statistic: 154.7 on 2 and 309 DF,  p-value: < 2.2e-16
```

Conclusion: Intercepts of both of the paperback and hardcover do not change. The slops change smaller after we use EM method.