Marketing Analytics Homework 1

Gor Yeghiazaryan

2025-10-05

Project Gutenberg Open Audiobook Collection

Project Gutenberg is the oldest digital library, started in 1971 to make e-books more accessible. But CEO Greg Newby says it "isn't great at either creating or distributing." So Microsoft and MIT teamed up to make the Open Audiobook Collection, using text-to-speech tech to turn 5,000 books into free, synthetically narrated audiobooks, now available on Spotify. The software fueling the project was also released at no charge.

Data for Bass model

The company I have chosen as look-alike innovation is Harper Collins. HarperCollins is today the second-largest consumer book publisher in the world. The company publishes approximately 10,000 new books every year in 16 languages and has a print and digital catalog of more than 200,000 titles.

The innovation and this company are very similar to each other. Project Gutenberg Open Audiobook Collection uses modern neural text-to-speech technology to make literature more available to readers.

I think those 2 companies are similar to each other as they both produce audiobooks using different approaches. Using the data from HarperCollins I will find the estimates.

The data includes essential details such as the year, revenue and change. It includes the data starting from 2013 up to 2022.

This is the data from Harper Collins. It shows different statistics but the focus is on the revenue and the year.

```r
library(knitr)
knitr::include_graphics("harpercollins.pdf")
```

Fiscal year	**Revenue ($ million)**	**Change (%)**	**EBITDA ($ million)**	**Change (%)**
2022	2,191	10.38%	306	0.99%
2021	1,985	19.15%	303	41.59%
2020	1,666	-5.02%	214	-15.08%
2019	1,754	-0.23%	252	5.44%
2018	1,758	7.46%	239	20.10%
2017	1,636	-0.61%	199	7.57%
2016	1,646	-1.26%	185	-16.29%
2015	1,667	16.25%	221	12.18%
2014	1,434	4.75%	197	38.73%
2013	1,369		142	

Upload all the necessary libraries.

```r
library(ggplot2)
library(ggpubr)
library(knitr)
library(diffusion)
```

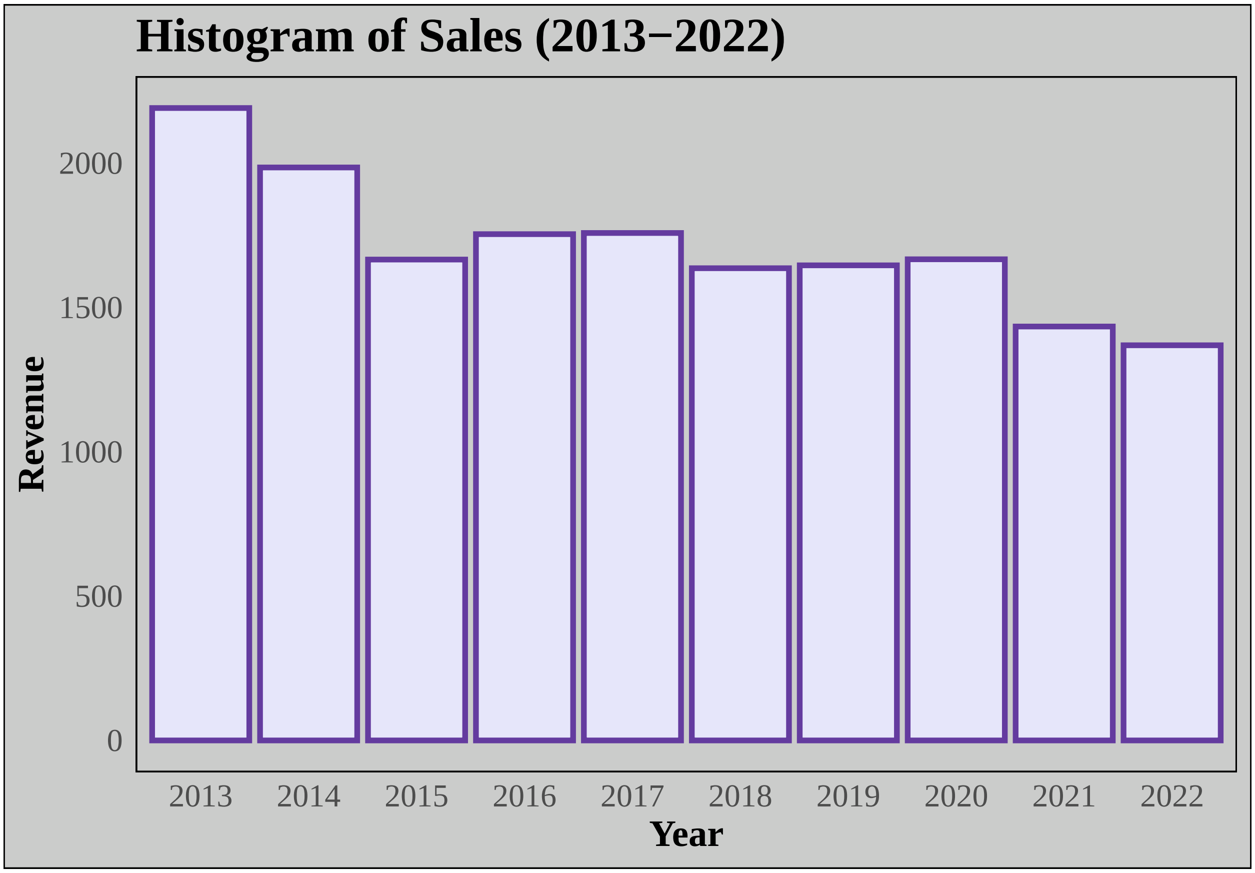
Approach 1: Take the whole data

The distribution shows that the rate of 2013 was the highest in the period 2013-2022, meaning that the p innovation rate is higher than q, which does not result in a bell shaped distribution.

One solution is to slice the data and estimate p and q based on the new dataset. Starting from 2015 the sales are decreasing, and based on that fact the second approach is to create a new dataset of sales starting from 2015.

```r
sales_data_2013 <- data.frame(
  year = 2013:2022,
  sales = c(2191, 1985, 1666, 1754, 1758, 1636, 1646, 1667, 1434, 1369)
)

ggplot(sales_data_2013, aes(x = as.factor(year), y = sales)) +
  geom_bar(stat = "identity", fill = "#E6E6FA", color = "#643b9f", linewidth = 1) +
  labs(title = "Histogram of Sales (2013-2022)",
       x = "Year",
       y = "Revenue") +
  theme_minimal() +
  theme(
    plot.title = element_text(family = "serif", face = "bold", size = 18),
    axis.title = element_text(family = "serif", face = "bold", size = 14),
    axis.text = element_text(family = "serif", size = 12),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "#cbcccb"),
    plot.background = element_rect(fill = "#cbcccb")
  )
```



The distribution shows that the revenue of the company reached its peak in the very beginning of the data in 2013, and after that it started to decline almost every year. Overall the pattern of the data when compared to 2013 is decreasing.

Define the bass functions for further use.

```
bass.f <- function(t, p, q) {
  ((p + q)^2 / p) * exp(-(p + q) * t) / (1 + (q / p) * exp(-(p + q) * t))^2
}

bass.F <- function(t, p, q) {
  (1 - exp(-(p + q) * t)) / (1 + (q / p) * exp(-(p + q) * t))
}
```

Based on the look-alike product's data predictions of p,q and m

Estimation 1: The first approach is to do with the use of the library diffusion. This way returns a q value almost equal to 0, meaning that this approach is not convienent for the estimations.

```r
sales_2013 = c(2191, 1985, 1666, 1754, 1758, 1636, 1646, 1667, 1434, 1369)
diff_m = diffusion(sales_2013)
p_dif=round(diff_m$w,4)[1]
q_dif=round(diff_m$w,4)[2]
m_dif=round(diff_m$w,4)[3]
diff_m
```

```
## bass model
##
## Parameters:
##     Estimate p-value
## m 38721.2591      NA
## p     0.0530      NA
## q     0.0221      NA
##
## sigma: 102.7059
```

Estimation 2: The second approach is using NLS(Non-linear Least Squares) method. But with the given data p is larger than q, and as the step above showed the estimation of q is a very small number. For that reason it returns an error related to the minimum number of iterations.

Estimation 3: Here is another approach in which initial parameters are given than they are optimized. The function bass_model has the argument params which are the parameters to calculate the bass model: p, q and m. The time period represents t and the data is called sales_data_2013.

Inside the function the aim of optimization is to minimize the sum of squares. Then initial parameters are given to understand what the model and the data represent. During the research, I found the optim function which minimizes the objective function in this case the sum of squared differences and the other arguments are passed to the function. The results are represented below. Without lower and upper bounds the function returned q as a negative number, so the function also needs lower bounds that p, q and m cannot be negative. The loer ower bound is p=0, q=0 and m=0. The lower bound works in case of the method specification called L-BFGS-B. The results are the following:

p = 0.07760767 q = 0.2399192 m = 17106

```r
t <- 1:length(sales_2013)

bass_model <- function(params, t, sales) {
  p <- params[1]
  q <- params[2]
  m <- params[3]

  predicted_sales <- ((p + q)^2 / p) * exp(-(p + q) * t) / (1 + (q / p) * exp(-(p + q) * t))^2 * m
  sum((predicted_sales - sales_2013)^2)
}

initial_params <- c(p = 0.02, q = 0.2, m = sum(sales_2013))

optimization <- optim(par = initial_params, fn = bass_model, t = t, sales = sales_2013,
                      lower = c(0, 0, 0), method = "L-BFGS-B")
```

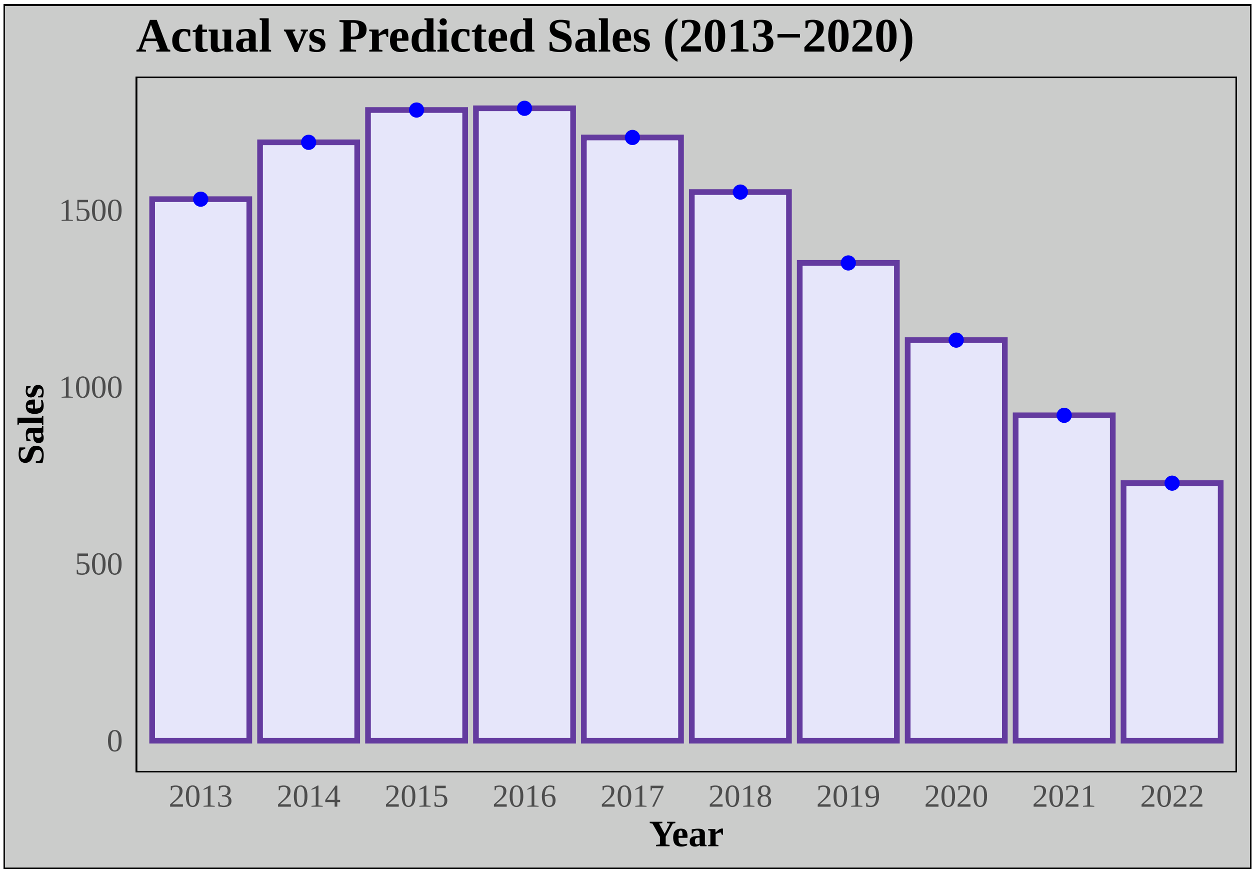
```r
params_optim <- optimization$par
print(params_optim)
```

```
##            p            q            m
## 7.760767e-02 2.399192e-01 1.710600e+04
```

Prediction for 2013 data

```r
pred_sales <- bass.f(1:10, p = 7.760767e-02, q = 2.399192e-01) * 1.710600e+04

ggplot(sales_data_2013, aes(x = as.factor(year), y = pred_sales)) +
  geom_bar(stat = "identity", fill = "#E6E6FA", color = "#643b9f", linewidth = 1) +
  geom_point(aes(y = pred_sales), color = 'blue', size = 2) +
  labs(title = "Actual vs Predicted Sales (2013-2020)",
       x = "Year",
       y = "Sales") +
  theme_minimal() +
  theme(
    plot.title = element_text(family = "serif", face = "bold", size = 18),
    axis.title = element_text(family = "serif", face = "bold", size = 14),
    axis.text = element_text(family = "serif", size = 12),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "#cbcccb"),
    plot.background = element_rect(fill = "#cbcccb")
  )
```



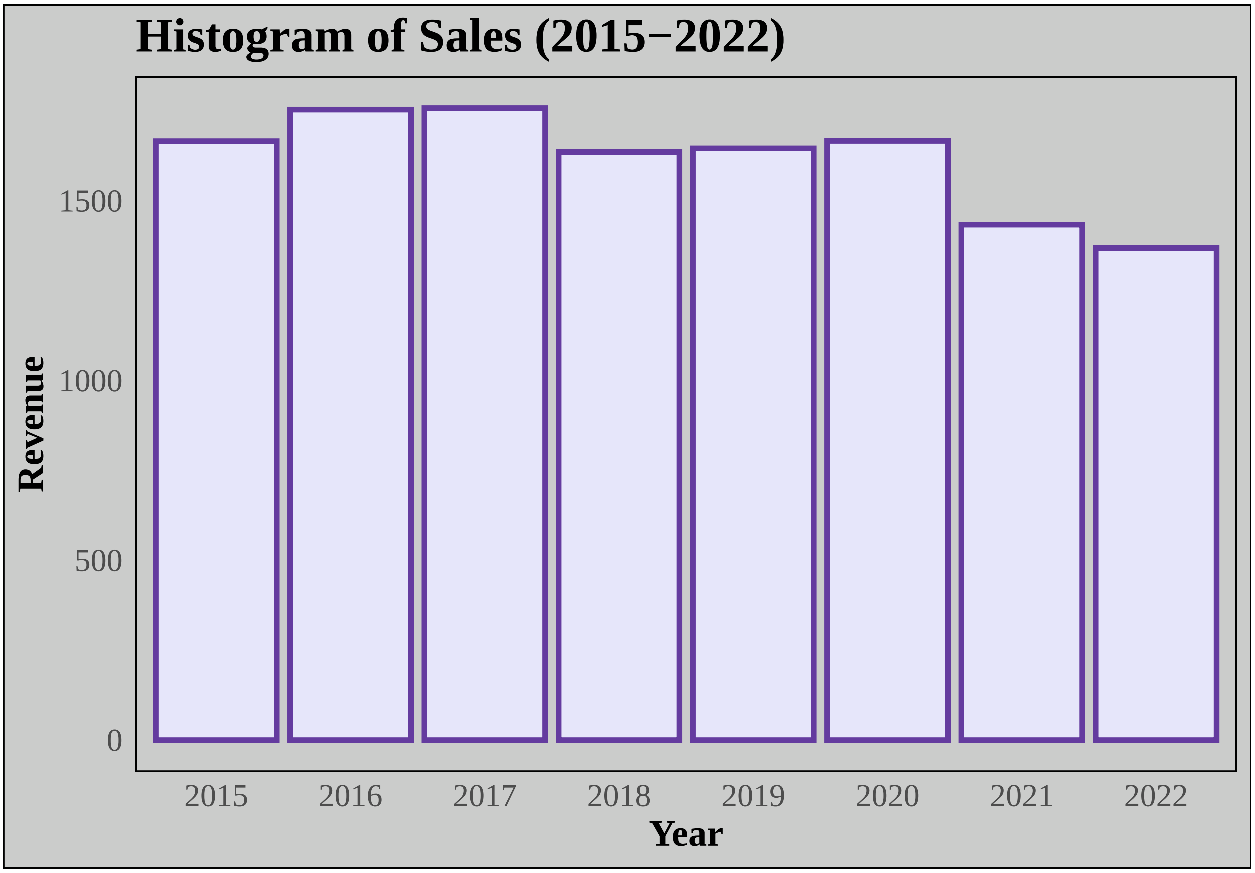
This values of p and q give the most exact predictions.

Approach 2: Take the data starting from 2015

```
sales_data_2015 <- data.frame(
  year = 2015:2022,
  sales = c(1666, 1754, 1758, 1636, 1646, 1667, 1434, 1369)
)

ggplot(sales_data_2015, aes(x = as.factor(year), y = sales)) +
  geom_bar(stat = "identity", fill = "#E6E6FA", color = "#643b9f", size = 1) +
  labs(title = "Histogram of Sales (2015-2022)",
       x = "Year",
       y = "Revenue") +
  theme_minimal() +
  theme(
    plot.title = element_text(family = "serif", face = "bold", size = 18),
    axis.title = element_text(family = "serif", face = "bold", size = 14),
    axis.text = element_text(family = "serif", size = 12),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "#cbcccb"),
    plot.background = element_rect(fill = "#cbcccb")
  )
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



From the graph it is obvious that p is smaller than the subsequent years. Here are the 3 estimations for the data starting from 2015:

Estimation 1: NLS works here, because q is not 0.

```r
t <- 1:length(sales_data_2015$sales)

bass_model <- function(t, m, p, q) {
  ((p + q)^2 / p) * exp(-(p + q) * t) / (1 + (q / p) * exp(-(p + q) * t))^2 * m
}

bass_m <- nls(sales ~ bass_model(t, m, p, q),
              data = sales_data_2015,
              start = c(m = sum(sales_data_2015$sales), p = 0.02, q = 0.4))

summary(bass_m)
```

```
##
## Formula: sales ~ bass_model(t, m, p, q)
##
```

```
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## m 2.288e+04  2.092e+03   10.94 0.000111 ***
## p 7.061e-02  4.472e-03   15.79 1.85e-05 ***
## q 1.202e-01  2.803e-02    4.29 0.007792 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58.84 on 5 degrees of freedom
##
## Number of iterations to convergence: 6
## Achieved convergence tolerance: 3.507e-06
```

M=22880, p=0.07061, q=0.1202

Estimation 2: Using optim function.

```
t <- 1:length(sales_data_2015$sales)

bass_model <- function(params, t, sales) {
  p <- params[1]
  q <- params[2]
  m <- params[3]

  predicted_sales <- ((p + q)^2 / p) * exp(-(p + q) * t) / (1 + (q / p) * exp(-(p + q) * t))^2 * m
  sum((predicted_sales - sales_data_2015$sales)^2)
}

initial_params <- c(p = 0.02, q = 0.2, m = sum(sales_data_2015$sales))

optimization <- optim(par = initial_params, fn = bass_model, t = t, sales = sales_data_2015$sales,
                      lower = c(0, 0, 0), method = "L-BFGS-B")

params_optim <- optimization$par
print(params_optim)
```

```
##            p            q            m
## 7.214352e-02 3.731970e-01 1.293000e+04
```

m= 12930 p=0.07214352 q=0.3731970

Estimation 3: Diffusion library

```
sales_2015 = c(1666, 1754, 1758, 1636, 1646, 1667, 1434, 1369)
diff_m = diffusion(sales_2015)
p_dif=round(diff_m$w,4)[1]
q_dif=round(diff_m$w,4)[2]
m_dif=round(diff_m$w,4)[3]
diff_m
```

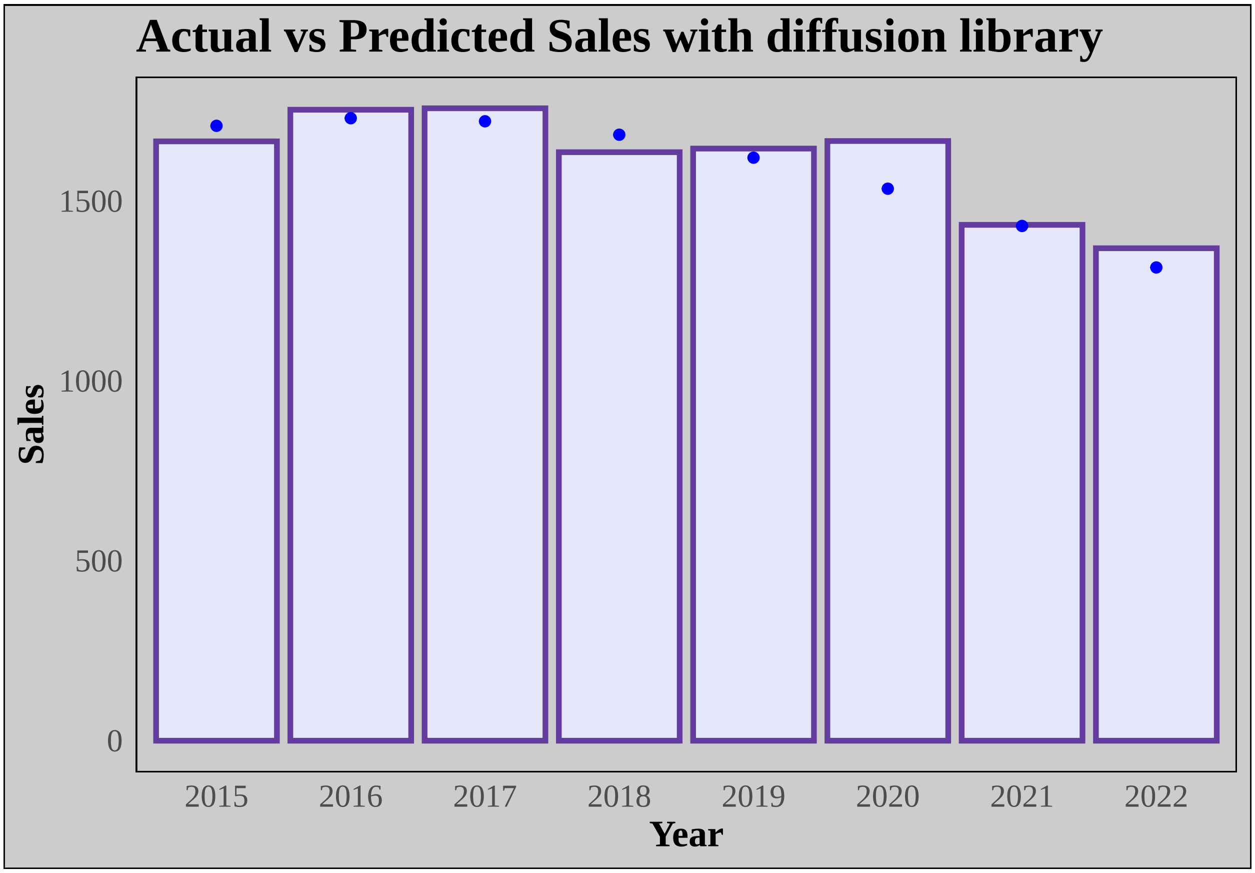
```
## bass model
##
## Parameters:
```

```
##     Estimate p-value
## m 21796.1283      NA
## p     0.0757      NA
## q     0.1192      NA
##
## sigma: 46.5993
```

m=22465 p=0.0739 q=0.1114

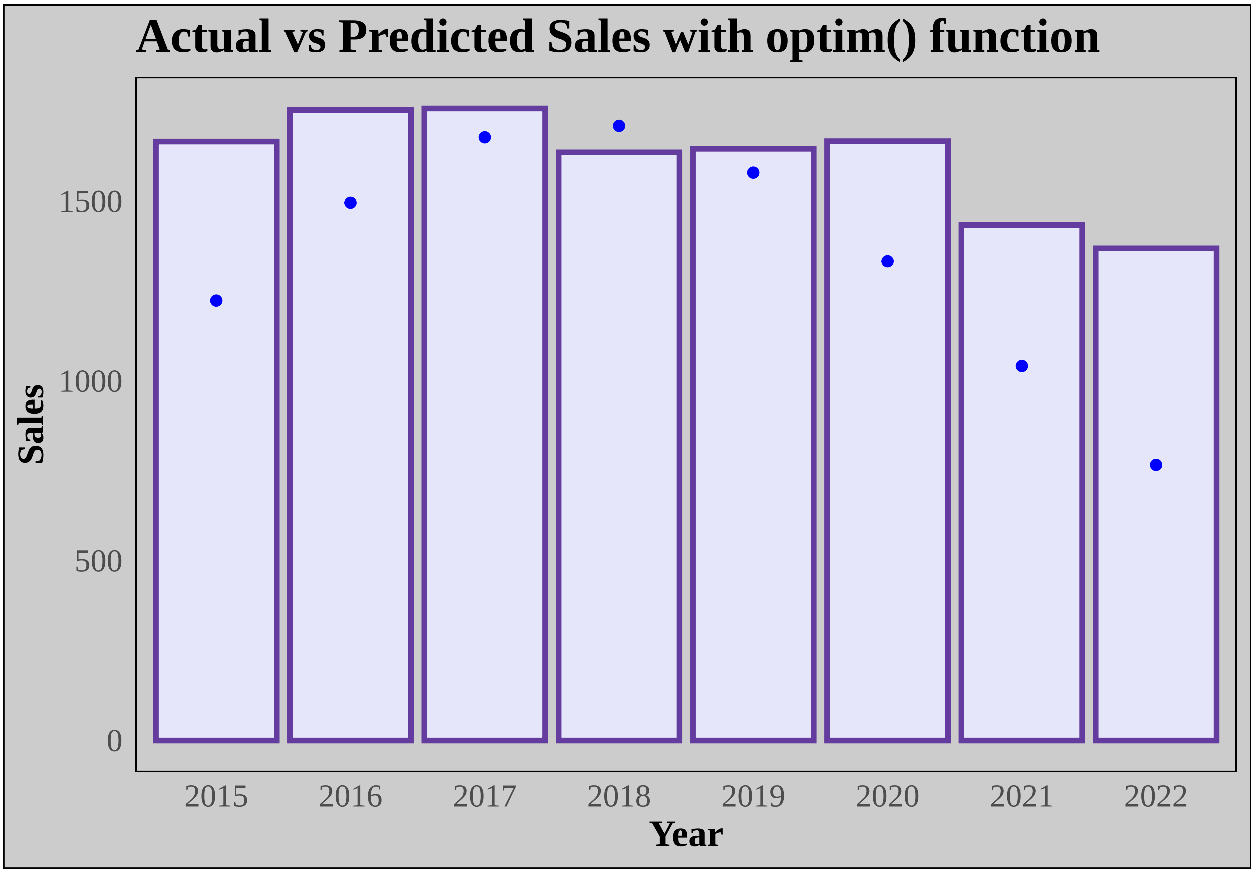
```
sales_data_2015$pred_sales <- bass.f(1:8, p = 0.0739, q = 0.1114) * 22465

ggplot(sales_data_2015, aes(x = as.factor(year), y = sales_2015)) +
  geom_bar(stat = "identity", fill = "#E6E6FA", color = "#643b9f", size = 1) +
  geom_point(aes(y = pred_sales), color = 'blue') +
  labs(title = "Actual vs Predicted Sales with diffusion library",
       x = "Year",
       y = "Sales") +
  theme_minimal() +
  theme(
    plot.title = element_text(family = "serif", face = "bold", size = 18),
    axis.title = element_text(family = "serif", face = "bold", size = 14),
    axis.text = element_text(family = "serif", size = 12),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "#cbcccb"),
    plot.background = element_rect(fill = "#cbcccb")
  )
```



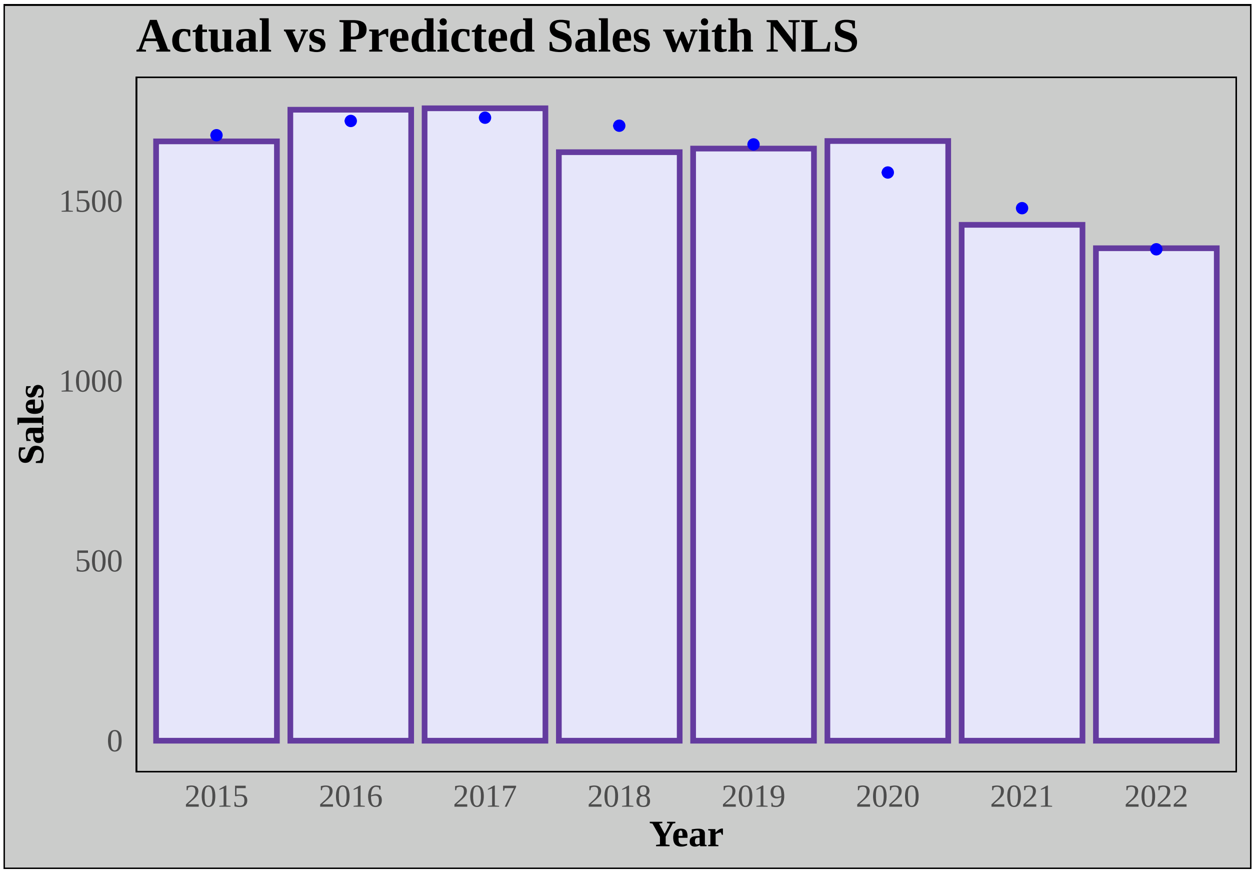
```r
sales_data_2015$pred_sales <- bass.f(1:8, p = 0.07214352, q = 0.3731970) * 12930

ggplot(sales_data_2015, aes(x = as.factor(year), y = sales_2015)) +
  geom_bar(stat = "identity", fill = "#E6E6FA", color = "#643b9f", size = 1) +
  geom_point(aes(y = pred_sales), color = 'blue') +
  labs(title = "Actual vs Predicted Sales with optim() function",
       x = "Year",
       y = "Sales") +
  theme_minimal() +
  theme(
    plot.title = element_text(family = "serif", face = "bold", size = 18),
    axis.title = element_text(family = "serif", face = "bold", size = 14),
    axis.text = element_text(family = "serif", size = 12),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "#cbcccb"),
    plot.background = element_rect(fill = "#cbcccb")
  )
```



```r
sales_data_2015$pred_sales <- bass.f(1:8, p=0.07061, q = 0.1202) * 22880

ggplot(sales_data_2015, aes(x = as.factor(year), y = sales_2015)) +
  geom_bar(stat = "identity", fill = "#E6E6FA", color = "#643b9f", size = 1) +
  geom_point(aes(y = pred_sales), color = 'blue') +
  labs(title = "Actual vs Predicted Sales with NLS",
       x = "Year",
       y = "Sales") +
  theme_minimal() +
  theme(
    plot.title = element_text(family = "serif", face = "bold", size = 18),
    axis.title = element_text(family = "serif", face = "bold", size = 14),
    axis.text = element_text(family = "serif", size = 12),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "#cbcccb"),
    plot.background = element_rect(fill = "#cbcccb")
  )
```

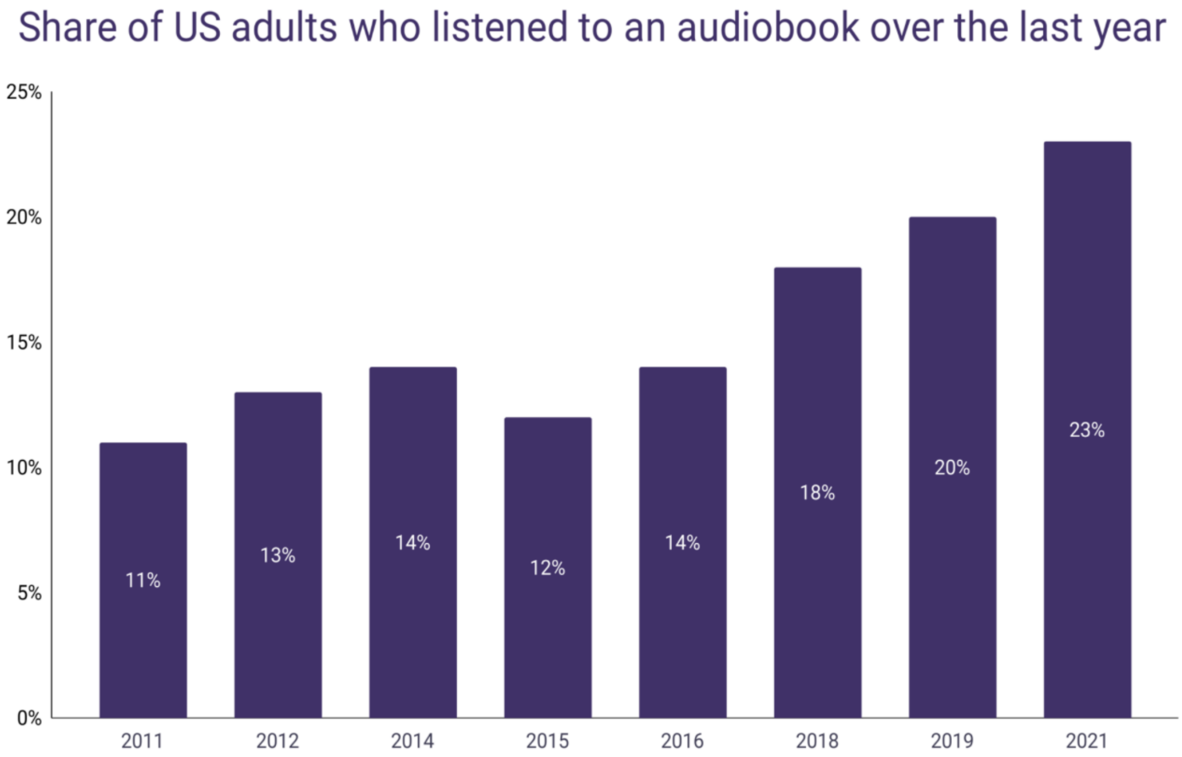


After gaining understanding about the values of p and q, the next step is the market estimation, to understand how many people will use the product and later on categorize them using the diffusion of innovation theory.

In the above estimations, M gave different values which are not very far from each other. As there is no exact number, taking the average of these numbers will be one possible solution. Therefore, m is 25542 for the look-alike innovation.

Fermi estimation for our innovation

```r
library(knitr)
knitr::include_graphics("listener_info.pdf")
```



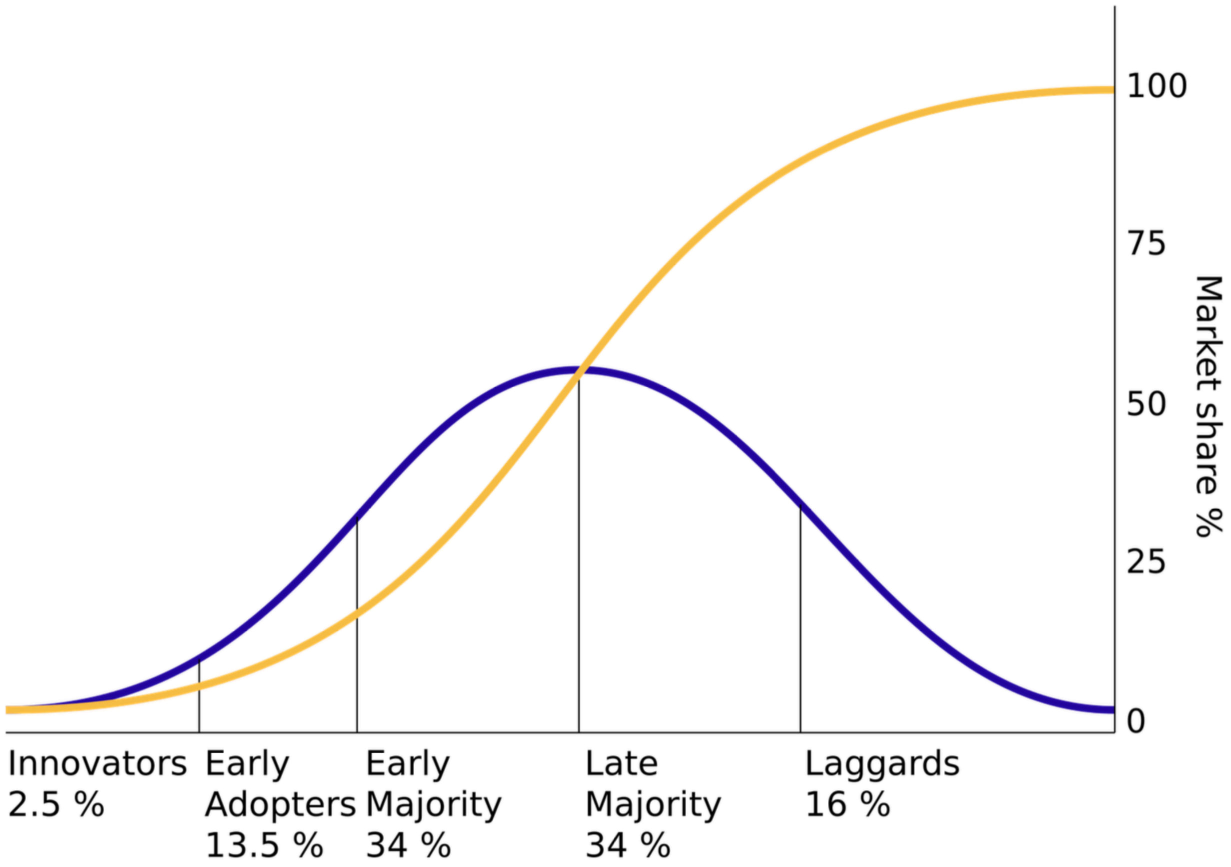
This data suggests that in 2021 23% of the adults population used audiobooks in the US. The population in 2021 was 338,289,857, and with calculations we get that approximately 77 million people used audiobooks.

The number of users is decreasing over time. The current population of the United States of America is 341,183,724. The data is for adults and now in the USA there are 258,300,000 adults.The proportion of adults has increased by 10 percent compared to 2010. Consider 10 percent of the USA adults population now uses audiobooks. It makes 25 million from the population uses audiobooks. Suppose that for the start 10 percent of the population will be influenced by the innovation with the help of innovators and early adopters. And out of 250000 people 10 percent will use the product. As a result of all those calculations, the upper bound of market potential for one year (t is a time period and the estimation if for years) will be approximately 250 thousand people.

The market share (m) is 250 thousand people.

Now, based on the Diffusion of Innovation, here are the calculations:

```r
library(knitr)
knitr::include_graphics("diffusion.pdf")
```



Innovators = 0.025 * 250.000 = 6250

Early Adopters = 0.135 * 250.000 = 33.750

Early Majority = 0.34 * 250.000 = 85.000

Late Majority = 0.34 * 250.000 = 85.000

Laggards = 0.16 * 250.000 = 40.000

Conclusion

The data for the look-alike product is the same as the innovation but using different approaches. The data is for the US only. After doing experiments, the best one is using the data starting from 2015 to avoid having p big and q almost 0. The estimations are for 3 different p q and m estimations. As the plots show, NLS starting from 2015 shows the best predictions for the data we have.

References

Innovation of Project Gutenberg Open Audiobook Collection

https://time.com/collection/best-inventions-2023/6324762/project-gutenberg-open-audiobook-collection/

Data Source: Harper Collins' data for revenue and year https://wordsrated.com/harpercollins-sales-statistics/

Audiobook users information

https://wordsrated.com/audiobook-statistics/#:~:text=Audiobook%20listening%20habits&text=Over%2023%25%20of%20A

Popoulation of USA in 2021 and 2024

https://wordsrated.com/audiobook-statistics/#:~:text=Audiobook%20listening%20habits&text=Over%2023%25%20of%20A