Marketing Hw1

2024-02-23

Before starting the homework itself let's install the necesary packages

```
libs<-c('ggplot2','ggpubr','knitr','diffusion', 'inline')
load_libraries<-function(libs){
  new_libs <- libs[!(libs %in% installed.packages()[,"Package"])]
  if(length(new_libs)>0) {install.packages(new_libs)}
  lapply(libs, library, character.only = TRUE)
}
load_libraries(libs)
```

```
## [[1]]
## [1] "ggplot2"   "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[2]]
## [1] "ggpubr"    "ggplot2"   "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
##
## [[3]]
##  [1] "knitr"     "ggpubr"    "ggplot2"   "stats"     "graphics"  "grDevices"
##  [7] "utils"     "datasets"  "methods"   "base"
##
## [[4]]
##  [1] "diffusion" "knitr"     "ggpubr"    "ggplot2"   "stats"     "graphics"
##  [7] "grDevices" "utils"     "datasets"  "methods"   "base"
##
## [[5]]
##  [1] "inline"    "diffusion" "knitr"     "ggpubr"    "ggplot2"   "stats"
##  [7] "graphics"  "grDevices" "utils"     "datasets"  "methods"   "base"
```

Every year, Time magazine publishes a list of the best 100 innovations of that year. Go to the list, choose an innovation, and put the link of the selected product.

I selected a product, which is a "A Longer-lasting Laptop" Framework Laptop 16. It is a new laptop that can be continually modified by swapping out its parts, which will let users upgrade the CPU and graphics—and keep older machines out of landfills. This has a better battery health and it is an upgraded version of the previous laptops in the market. To be honest, I could not find a lot of differences in this innovation and I would call it "an upgraded laptop" if I had a chance. Yet a very good and useful inovation especially for those who work in IT sphere.

Thinking about look-alike innovation from the past I decided to take simple laptops which for the past years where the boom in the market and changed a lot of things in humans' lives. For this reason, I found a data related to revenue of the laptops industry worldwide from 2019-2028, but I will use just the subset of it and make predictions based on that. I guess the innovation of the laptop was world-changing and year after year even more interesting and inovative sort of it is produced and introduced to the market.

The reason why I chose this innovation for my further analysis is that while reading about this product I understood that the market potential of this product is just a subset of the one of the laptops'. The reason is the product has a relatively higher price and the ones who have enough money and purpose of using it will get it. If I personally had a thought to launch this type of product I would analyse the laptop market as well. So here I am for doing just the little part of it based on Bass Model Analysis.

```
library(readxl)
excel_file <- "statistic_id1181717_revenue-of-the-laptops-industry-worldwide-2019-2028.xlsx"
data <- read_excel(excel_file, sheet = "Data", skip = 4, col_names = c("year", "revenue"))

data
```

```
## # A tibble: 10 x 2
##    year  revenue
##    <chr>   <dbl>
##  1 2019     117.
##  2 2020     129.
##  3 2021     131.
##  4 2022     126.
##  5 2023     128.
##  6 2024     128.
##  7 2025     132.
##  8 2026     137.
##  9 2027     142
## 10 2028     147.
```

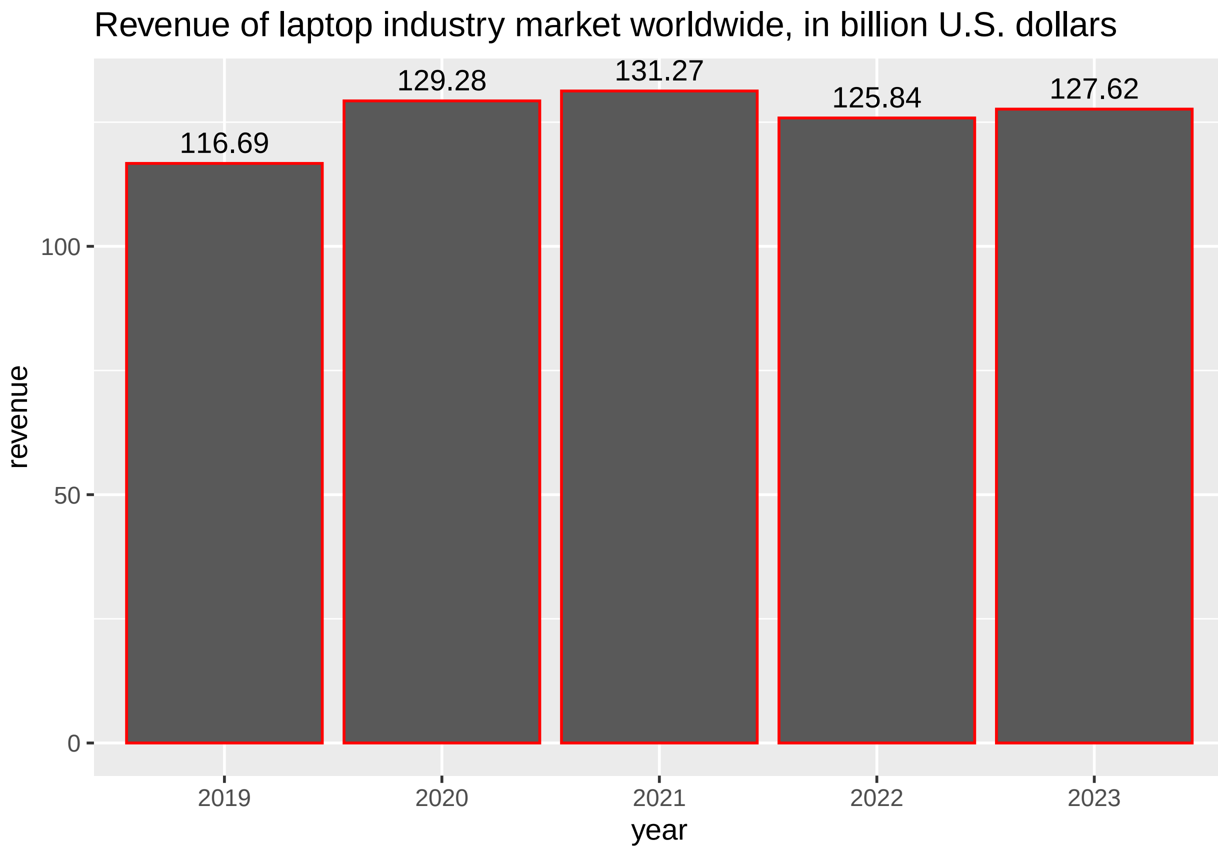
I specified the column names to make it more understandable of what we will be working with, now I will filter the data to use just the subset of it we need. To make it more clear, I have a worldwide data of laptop revenue from worldwide market.

```
filtered_data <- data[data$year <= 2023, ]
filtered_data
```

```
## # A tibble: 5 x 2
##   year  revenue
##   <chr>   <dbl>
## 1 2019     117.
## 2 2020     129.
## 3 2021     131.
## 4 2022     126.
## 5 2023     128.
```

Now I have the data I will later on work with for this homework specifically. Later on I promise to compare the value I got for the next year and the ones in the excel_sheet. Not promising the alike values yet)

```
ggplot(data = filtered_data, aes(x = year, y = revenue)) +
  geom_bar(stat = 'identity', color = 'red') +
  geom_text(aes(label = revenue), vjust = -0.5, size = 4) +
  ggtitle('Revenue of laptop industry market worldwide, in billion U.S. dollars')
```

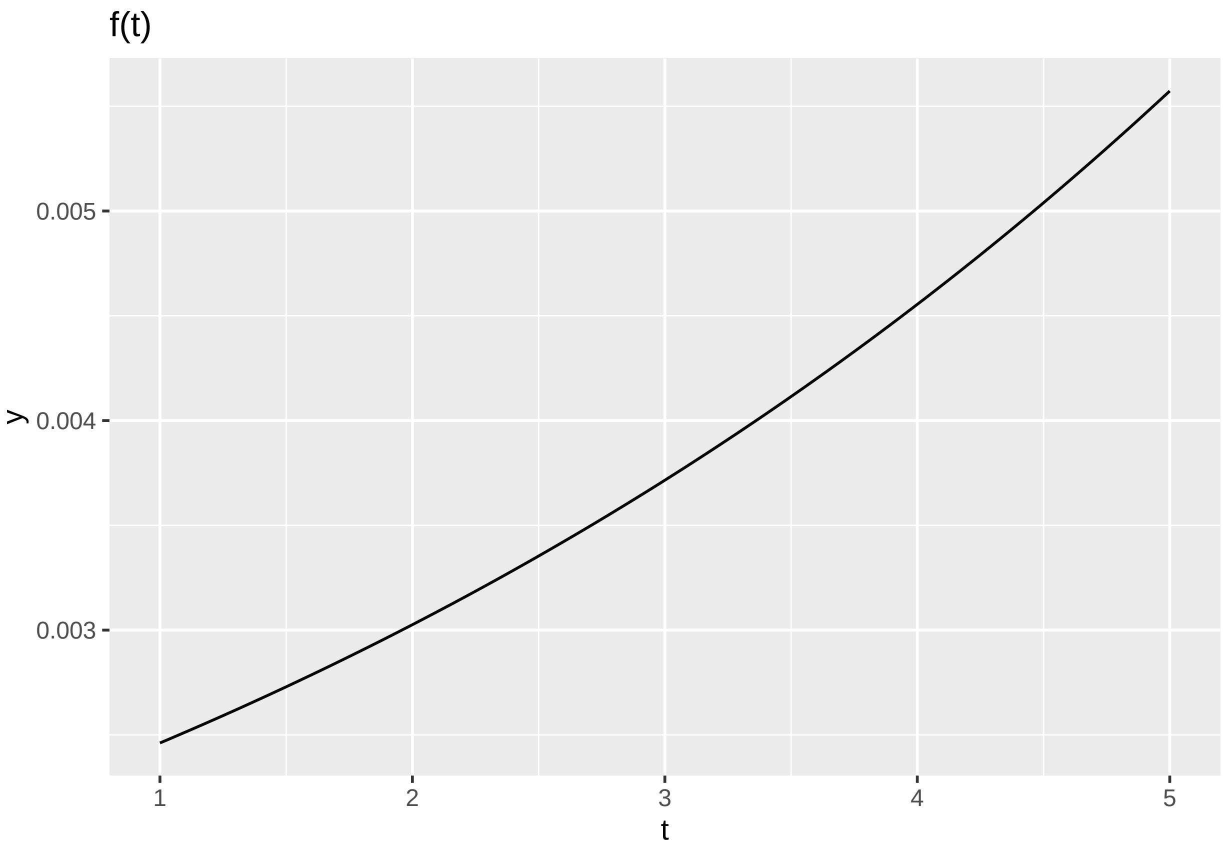


As we can see, there are no big drops or incrases of the sales since the revenue of the worldwide market of laptops was more or less the same through past 5 years. The slight raise can be detected between 2019-2020 and 2020-2021 yet then it dropped by 5.43 billion dollars.

Let's now difine the bass functions for the later use. You will see them throughout the code a lot. They are just implemented versions of the formulas.

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

```
time_ad <- ggplot(data.frame(t = c(1:5)), aes(t)) +
  stat_function(fun = bass.f, args = c(p = 0.002, q = 0.21)) +
  labs(title = 'f(t)')

time_ad
```



In this case, the time_ad represents the plot of bass model based on time, p and q in this very case where given randomly. Through the output we can easily detect the increasing trend of the plot.

Task 3: Estimate Bass model parameters for the look-alike innovation. Firstly let me define the function of bass model to make the latter happen.

Now I will use the bass.f I have already defined above.

```
bass_model <- function(params, t) {
  p <- params[1]
  q <- params[2]
  m <- max(filtered_data$revenue)
  m / (1 + ((m / p) * exp(-(p + q) * t)))
}
```

The model is defined. Let's have a look if we have right type of variables to work with.

```
summary(filtered_data)
```

```
##       year              revenue
##  Length:5           Min.   :116.7
##  Class :character   1st Qu.:125.8
##  Mode  :character   Median :127.6
##                     Mean   :126.1
##                     3rd Qu.:129.3
##                     Max.   :131.3
```

```
typeof
```

```
## function (x)
```

```
## .Internal(typeof(x))
## <bytecode: 0x7f95a39d3398>
## <environment: namespace:base>
```

Everything is okey!

```r
# Defined the Bass diffusion model function
bass_model <- function(params, year) {
  p <- params[1]
  q <- params[2]
  pqt <- 1 - exp(-(p + q) * as.numeric(year))
  return((p + q) / p * pqt - q)
}

sum_squared_errors <- function(params, year, revenue) {
  predicted_revenue <- bass_model(params, year)
  return(sum((revenue - predicted_revenue)^2))
}

initial_params <- c(p = 0.01, q = 0.1) #random

# Optimize the sum of squared errors using the Nelder-Mead algorithm
opt_result <- optim(par = initial_params, fn = sum_squared_errors, year = filtered_data$year, revenue =


opt_params <- opt_result$par
opt_params
```

```
##           p           q
## 0.001012374 0.126822009
```

Let's recall what p and q show and try to elaborate on the scores we got. P shows the innovation rate and q shows the imitation rate. We see that p is relatively slower while the q is higher The affect of internel influence is higher. Now p = 0.001012374, q = 0.126822009 are my new optimized estimators of Bass model.

#TODO: make this point swork

Task 4: Make predictions of the diffusion of the innovation

```r
library(diffusion)

revenue <- filtered_data$revenue

diffusion_results <- diffusion(revenue)#estimated my parameteres
estimated_parameters <- round(diffusion_results$w, 4)
p_estimate <- estimated_parameters[1]
q_estimate <- estimated_parameters[2]
m_estimate <- estimated_parameters[3]

diffusion_results
```

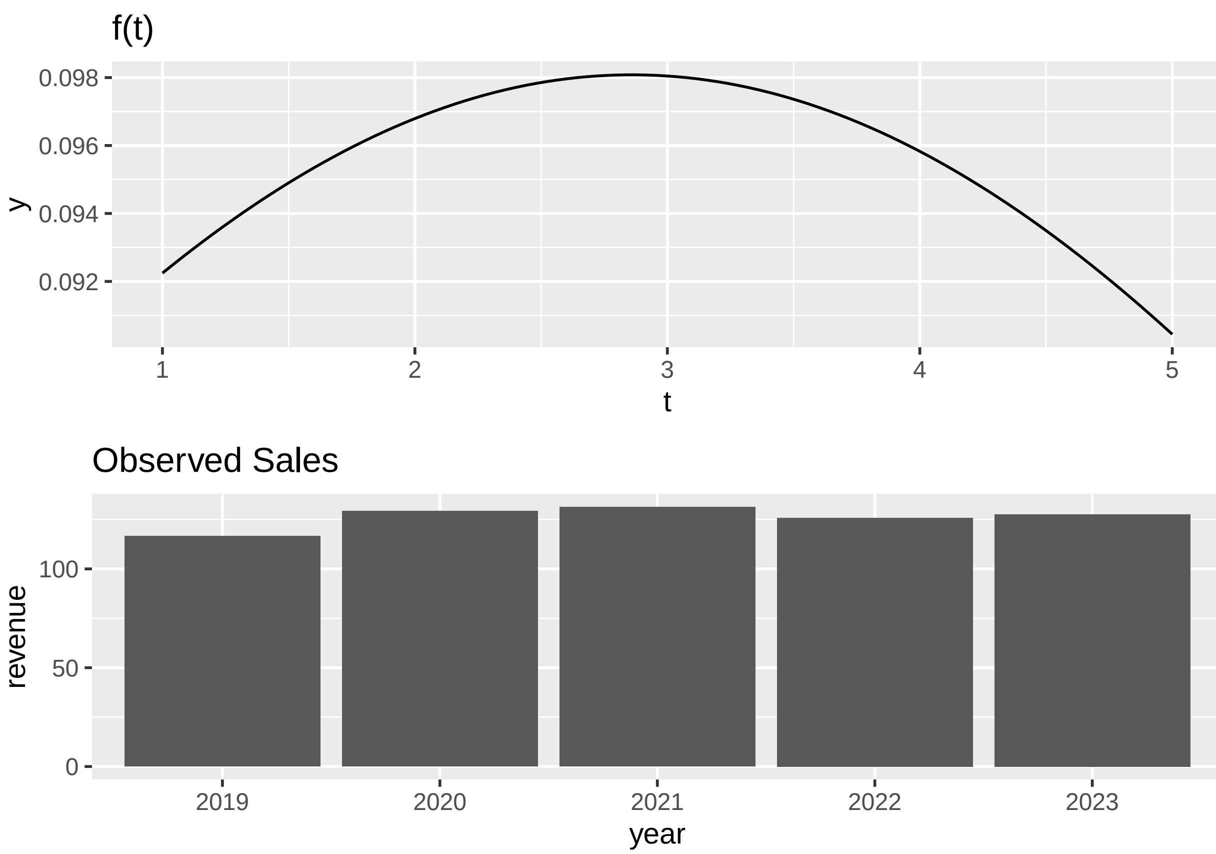
```
## bass model
##
## Parameters:
##                                  Estimate p-value
## p - Coefficient of innovation    0.0850      NA
```

```
## q - Coefficient of imitation     0.1828      NA
## m - Market potential          1335.2133      NA
##
## sigma: 2.6887
```

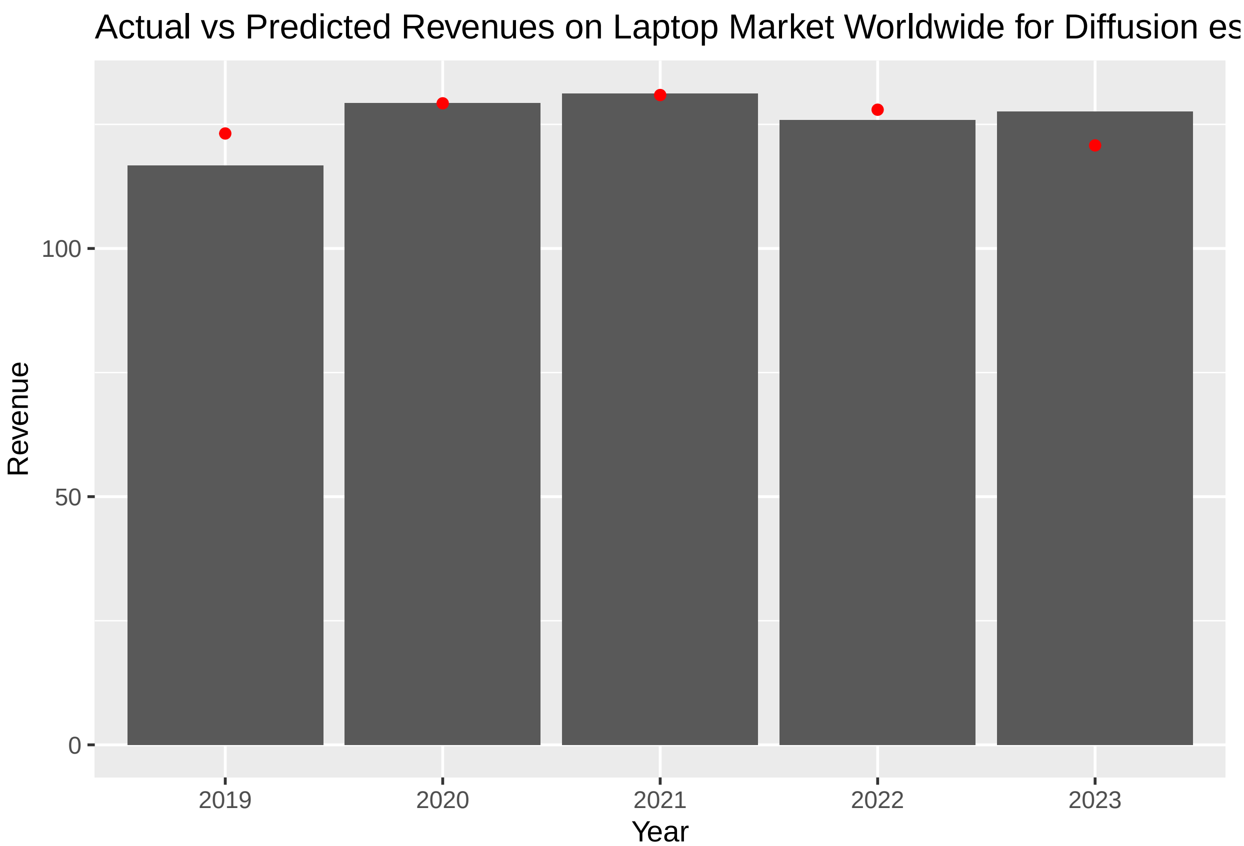
Let me explain the results we have got:

p - Coefficient of innovation is estimated to be 0.0850 which shows the rate of innovator-customers who adopt the product. q - Coefficinet of imitation is estimated to be 0.1828 which shows the rate of immitator-customers who adopt the product. m - Market potential is estimated to be 1335.2133 which shows the potential number of customers who will eventually buy the product. sigma - std of 2.6887 which measures inaccuracy of the estimation.

```r
library(ggplot2)
library(ggpubr)

time_df <- data.frame(t = 1:5)

time_ad <- ggplot(time_df, aes(t)) +
  stat_function(fun = bass.f, args = c( p = 0.0850, q = 0.1828)) +
  labs(title = 'f(t)')


ggarrange(time_ad, ggplot(filtered_data, aes(x = year, y = revenue)) +
            geom_bar(stat = 'identity') +
            labs(title = 'Observed Sales'),
          ncol = 1, nrow = 2)
```



```r
pred_sales <- bass.f(1:5, p = 0.0850, q = 0.1828) * 1335.2133

ggplot(filtered_data, aes(x = as.factor(year), y = revenue)) +
  geom_bar(stat = "identity") +
  geom_point(color = 'red', y = pred_sales) +
  labs(title = "Actual vs Predicted Revenues on Laptop Market Worldwide for Diffusion estimation",
       x = "Year", y = "Revenue")
```



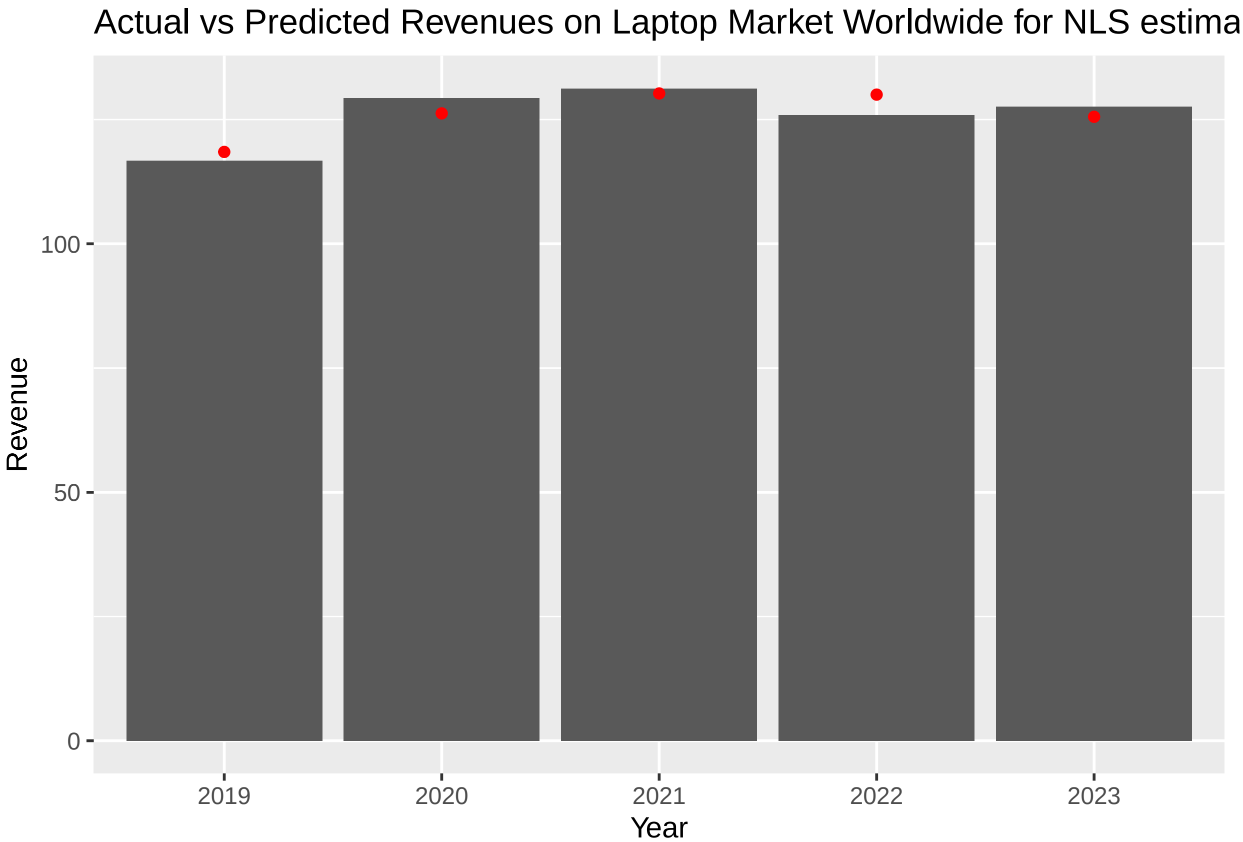
The red dots show our estimation, for this case we can see that it perfectly fits the 2020-2021 years, yet the values of 2019, 2022 and 2023 are very close to the actual ones. We are on the right ways.

```r
revenue = filtered_data$revenue
t = 1:length(revenue)
bass_m = nls(revenue  ~ m*(((p+q)**2/p)*exp(-(p+q)*t))/
               (1+(q/p)*exp(-(p+q)*t))**2,
              start=c(list(m=sum(revenue),p = 0.0850, q = 0.1828)))
bass_m
```

```
## Nonlinear regression model
##   model: revenue ~ m * (((p + q)^2/p) * exp(-(p + q) * t))/(1 + (q/p) *     exp(-(p + q) * t))^2
##    data: parent.frame()
##         m         p         q 
## 1.435e+03 7.519e-02 1.830e-01 
##  residual sum-of-squares: 35.18
## 
## Number of iterations to convergence: 5 
## Achieved convergence tolerance: 1.941e-06
```

If we look at the reuslts we got closer, we would see the estimated parameters, which are m, p and q. For m we have maximum market size of 1435, the rate of innovators is 0.07519 and for the imitators we have 0.183. The convergence tolerance achieved during the optimization process is 1.941e-06, indicating a high level of precision in estimating the parameters.The sum-of-squares is 35.18, suggesting that the model explains a significant portion of the variance in the data.Good scores!

```
pred_sales <- bass.f(1:5, p = 0.07519, q = 0.183) * 1435

ggplot(filtered_data, aes(x = as.factor(year), y = revenue)) +
  geom_bar(stat = "identity") +
  geom_point(color = 'red', y = pred_sales) +
  labs(title = "Actual vs Predicted Revenues on Laptop Market Worldwide for NLS estimation",
       x = "Year", y = "Revenue")
```



After the previous plot we can detect the values of 2019 and 2023 getting more concrete. This means tha it was a good idea to use the previous estimated p and q values for the non linear estimation as well.

```
# I do not know if I understood the assignment right so I wrote this code and asked some of my friends

calc_adopters <- function(p, q, m, t) {
  adopters <- m * ((p + q)**2 / q) * (1 - exp(-(p + q) * t))
  return(adopters)
}

p <- 0.07519
q <- 0.183
m <- 1435


years <- 1:5
```

```r
adopters <- calc_adopters(p, q, m, years)
print(adopters)
```

```
## [1] 118.9486 210.8303 281.8042 336.6278 378.9763
```

Year Excel My estimation 2024 127.51 118.9486
2025 132.09 210.8303
2026 136.66 281.8042
2027 142.00 336.6278
2028 147.45 378.9763

Here the estimation of my adopters increased which shoes the time_ad as well. Yet the values are not too close, the the trend of excel file for 2024-2028 and my estimations using the bass model are identical) I am happy for that since it is what I was leading for.

```r
percent_innovators <- 0.025
percent_early_adopters <- 0.135
percent_early_majority <- 0.34
percent_late_majority <- 0.34
percent_laggards <- 0.16

innovators <- percent_innovators * m
early_adopters <- percent_early_adopters * m
early_majority <- percent_early_majority * m
late_majority <- percent_late_majority * m
laggards <- percent_laggards * m

cat("Estimated number of adopters for each category:\n")
```

```
## Estimated number of adopters for each category:
```

```r
cat("Innovators:", innovators, "\n")
```

```
## Innovators: 35.875
```

```r
cat("Early Adopters:", early_adopters, "\n")
```

```
## Early Adopters: 193.725
```

```r
cat("Early Majority:", early_majority, "\n")
```

```
## Early Majority: 487.9
```

```r
cat("Late Majority:", late_majority, "\n")
```

```
## Late Majority: 487.9
```

```r
cat("Laggards:", laggards, "\n")
```

```
## Laggards: 229.6
```

The numbers we got are real-like. This is the suggestion of my friend to estimate the proprotions in numbers as well. Interesting.

Overall, I really liked working with the real product and real data and use the knowledge we have just earned to put in practice. Thank you !

** References

The innovation I chose: https://time.com/collection/best-inventions-2023/6323599/framework-laptop-16/ The worldwide data I got will be in the repository as well.