# Homework 1

Ruzanna Margaryan

2024-10-18

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(minpack.lm)
```

The innovation "Virtual Fauna" shares similarities with AR (Augmented Reality) due to their common focus on blending the physical and digital worlds. Virtual Fauna, like AR, utilizes technology to create interactive digital creatures that can inhabit real-world environments through devices like smartphones or AR glasses. Both innovations enhance user experiences by overlaying digital elements onto reality.

In terms of market impact, AR revolutionized industries like gaming (e.g., Pokémon GO) and retail (virtual try-ons), while Virtual Fauna can similarly impact entertainment, education, and wildlife simulation, offering immersive interactions with virtual animals.

```r
data <- read.csv('Data.csv', stringsAsFactors = FALSE, sep = ";")
data$AR.Hardware <- as.numeric(data$AR.Hardware)
```

```
## Warning: NAs introduced by coercion
```

```r
data$AR.Software <- as.numeric(data$AR.Software)
data$AR.Hardware[is.na(data$AR.Hardware)] <- 0
filtered_data <- data %>% filter(Year <= 2023) %>%
  select(Year, AR.Software, AR.Hardware)
filtered_data <- filtered_data %>%
  mutate(Total.AR.Adoption = AR.Software + AR.Hardware)
filtered_data
```

```
##   Year AR.Software AR.Hardware Total.AR.Adoption
## 1 2017      137.20        0.00            137.20
## 2 2018      143.50        0.00            143.50
```

```
## 3 2019      154.71        0.00            154.71
## 4 2020      172.96        9.05            182.01
## 5 2021      199.01       13.58            212.59
## 6 2022      230.94       21.92            252.86
## 7 2023      263.36       49.45            312.81
```

```r
bass_model_cumulative <- function(t, p, q, M) {
  adoption <- M * (1 - exp(-(p + q) * t)) / (1 + (q / p) * exp(-(p + q) * t))
  return(adoption)
}

time_periods <- 1:nrow(filtered_data)
total_adoption <- filtered_data$Total.AR.Adoption

initial_guess <- c(p = 0.03, q = 0.38, M = max(total_adoption) * 2)

fit <- nlsLM(Total.AR.Adoption ~ bass_model_cumulative(time_periods, p, q, M),
             data = filtered_data, start = initial_guess)

params <- coef(fit)
p <- params['p']
q <- params['q']
M <- params['M']
```

```r
cat("Estimated p (Coefficient of innovation):", p, "\n")
```

```
## Estimated p (Coefficient of innovation): 0.2200149
```

```r
cat("Estimated q (Coefficient of imitation):", q, "\n")
```

```
## Estimated q (Coefficient of imitation): -0.2200872
```

```r
cat("Estimated M (Market potential):", M, "\n")
```

```
## Estimated M (Market potential): 449.5019
```

```r
future_years <- 1:(nrow(filtered_data) + 12)

predicted_cumulative_adoption <- bass_model_cumulative(future_years, p, q, M)
predicted_yearly_adoption <- c(0, diff(predicted_cumulative_adoption))

prediction_df <- data.frame(
  Year = 2020:(2020 + length(future_years) - 1),
  Cumulative_Adoption = predicted_cumulative_adoption,
  Yearly_Adoption = predicted_yearly_adoption
)

print(prediction_df)
```

```
##    Year Cumulative_Adoption Yearly_Adoption
```

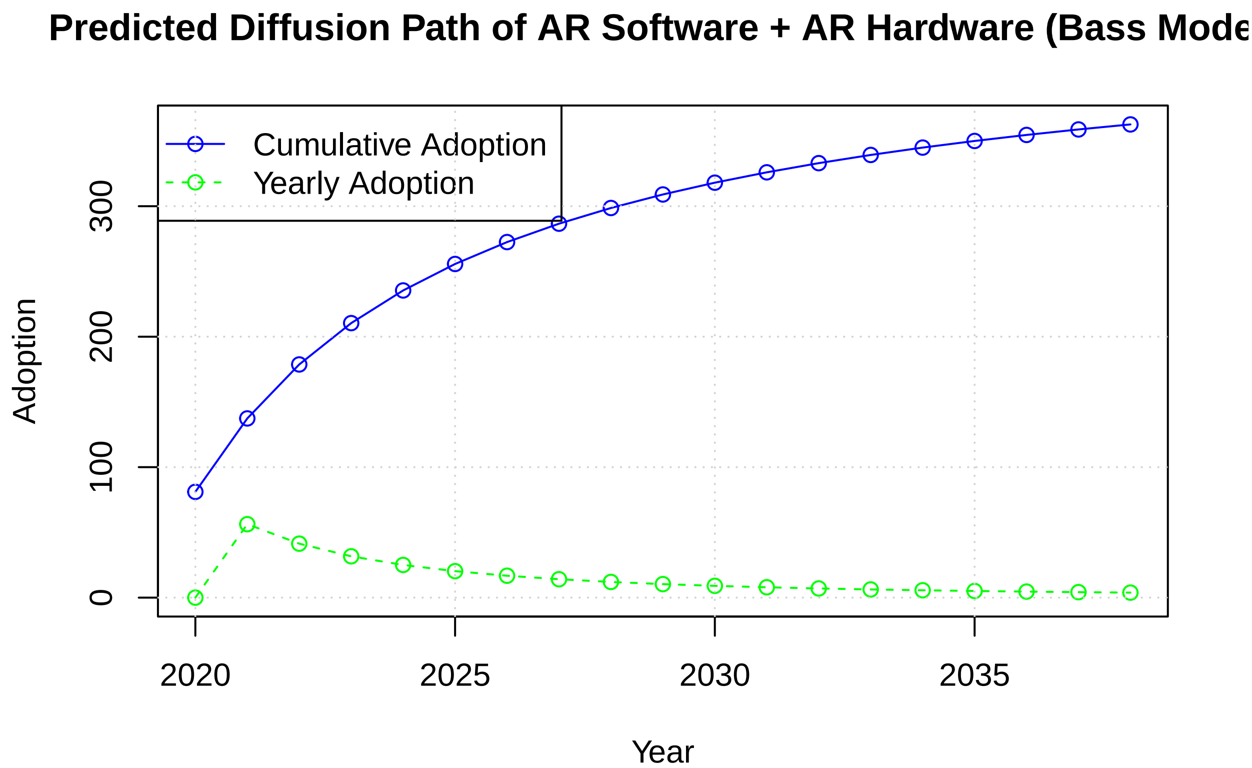
```
## 1  2020             81.0598        0.000000
## 2  2021            137.3473       56.287534
## 3  2022            178.7132       41.365861
## 4  2023            210.3965       31.683294
## 5  2024            235.4406       25.044156
## 6  2025            255.7346       20.293927
## 7  2026            272.5127       16.778123
## 8  2027            286.6158       14.103124
## 9  2028            298.6364       12.020622
## 10 2029            309.0041       10.367666
## 11 2030            318.0378        9.033724
## 12 2031            325.9795        7.941666
## 13 2032            333.0158        7.036343
## 14 2033            339.2933        6.277479
## 15 2034            344.9284        5.635097
## 16 2035            350.0149        5.086520
## 17 2036            354.6293        4.614330
## 18 2037            358.8342        4.204975
## 19 2038            362.6820        3.847780
```

```r
plot(prediction_df$Year, prediction_df$Cumulative_Adoption, type = "o", col = "blue",
     xlab = "Year", ylab = "Adoption", main = "Predicted Diffusion Path of AR Software + AR Hardware (Ba
     ylim = range(c(prediction_df$Cumulative_Adoption, prediction_df$Yearly_Adoption)))

lines(prediction_df$Year, prediction_df$Yearly_Adoption, type = "o", col = "green", lty = 2)

legend("topleft", legend = c("Cumulative Adoption", "Yearly Adoption"), col = c("blue", "green"),
       lty = c(1, 2), pch = c(1, 1))

grid()
```




As my data is related to only Italy I decided to analyze the diffusion focusing on a country-specific analysis.

```r
bass_model_cumulative <- function(t, p, q, M) {
  adoption <- M * (1 - exp(-(p + q) * t)) / (1 + (q / p) * exp(-(p + q) * t))
  return(adoption)
}
future_years <- 1:15

predicted_cumulative_adoption <- bass_model_cumulative(future_years, p, q, M)

predicted_yearly_adoption <- c(0, diff(predicted_cumulative_adoption))

adoption_df <- data.frame(
  Year = 2020:(2020 + length(future_years) - 1),
  Cumulative_Adoption = predicted_cumulative_adoption,
  Yearly_Adoption = predicted_yearly_adoption
)

print(adoption_df)
```

```
##    Year Cumulative_Adoption Yearly_Adoption
## 1  2020             81.0598        0.000000
## 2  2021            137.3473       56.287534
## 3  2022            178.7132       41.365861
## 4  2023            210.3965       31.683294
## 5  2024            235.4406       25.044156
```

```
## 6  2025             255.7346       20.293927
## 7  2026             272.5127       16.778123
## 8  2027             286.6158       14.103124
## 9  2028             298.6364       12.020622
## 10 2029             309.0041       10.367666
## 11 2030             318.0378        9.033724
## 12 2031             325.9795        7.941666
## 13 2032             333.0158        7.036343
## 14 2033             339.2933        6.277479
## 15 2034             344.9284        5.635097
```