Marketing Analytics Homework 1 Khachatryan Ela

2023-09-27

R Markdown

#importing necessary libraries

```
libs <- c('ggplot2', 'knitr', 'diffusion', 'ggpubr')

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

Getting data for the look-alike innovation

```
tesla <- read.csv("Tesla sales by year.csv", fileEncoding="UTF-8-BOM", sep = ";")
tesla
```

```
##   year   sales
## 1 2014  31.655
## 2 2015  50.658
## 3 2016  76.285
```

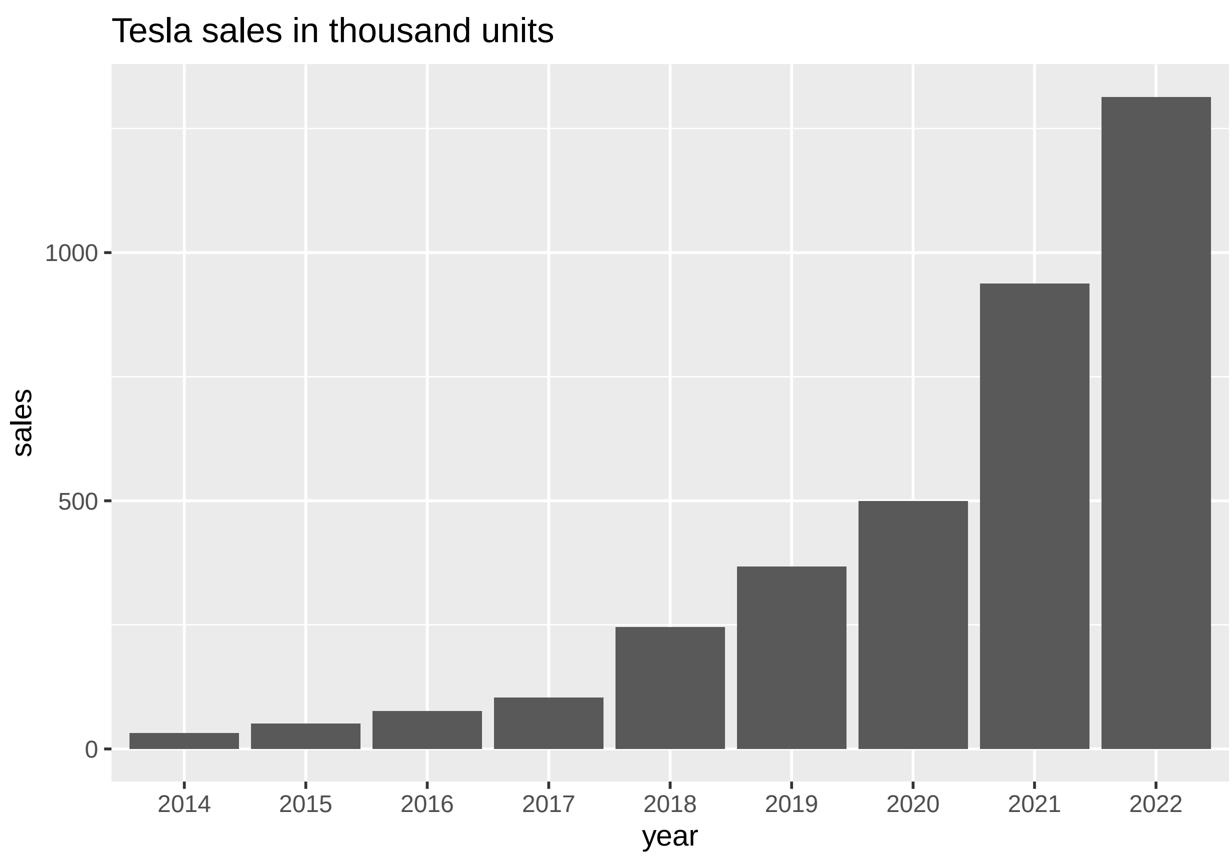
```
## 4 2017  103.181
## 5 2018  245.240
## 6 2019  367.500
## 7 2020  499.550
## 8 2021  936.950
## 9 2022 1313.581
```

Visualizing Tesla sales

```r
tesla$year <- factor(tesla$year)

plot_tesla = ggplot(data = tesla, aes(x = year, y = sales)) +
  geom_bar(stat = 'identity') +
  ggtitle('Tesla sales in thousand units')

plot_tesla
```



4. Estimating Bass model parameters for the look-alike innovation.

```r
sales_tesla = tesla$sales

t = 1:length(sales_tesla)
```

```r
bass_m = nls(sales_tesla ~ m*(((p+q)**2/p)*exp(-(p+q)*t))/
               (1+(q/p)*exp(-(p+q)*t))**2,
              start=c(list(m=sum(sales_tesla),p=0.02,q=0.4)))

options(scipen = 999, digits = 4) ## to avoid scientific notations
summary(bass_m)
```

```
##
## Formula: sales_tesla ~ m * (((p + q)^2/p) * exp(-(p + q) * t))/(1 + (q/p) *
##     exp(-(p + q) * t))^2
##
## Parameters:
##       Estimate   Std. Error t value Pr(>|t|)
## m 17339.947568 10123.648862    1.71   0.1376
## p     0.000847     0.000246    3.44   0.0138 *
## q     0.540172     0.067223    8.04   0.0002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.7 on 6 degrees of freedom
##
## Number of iterations to convergence: 38
## Achieved convergence tolerance: 0.00000409
```

5. Make predictions of the diffusion of the innovation you chose at stage 1.

```r
diff_m = diffusion(sales_tesla)
p=round(diff_m$w,4)[1]
q=round(diff_m$w,4)[2]
m=round(diff_m$w,4)[3]
diff_m
```

```
## bass model
##
## Parameters:
##                                Estimate p-value
## p - Coefficient of innovation     0.0006      NA
## q - Coefficient of imitation      0.4256      NA
## m - Market potential          57222.0298      NA
##
## sigma: 62.4626
```

Period when the sales will reach to the peak.

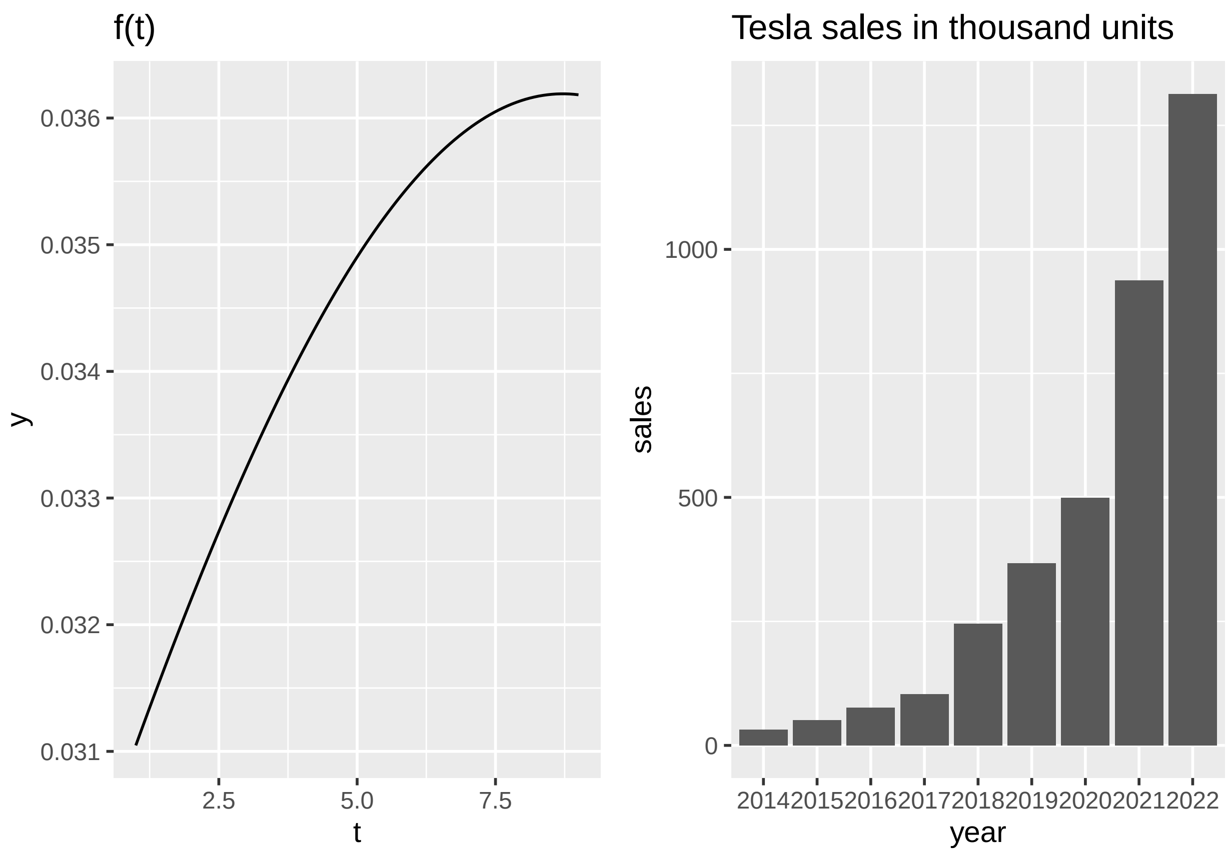
```r
data.frame(Predicted=log(q/p)/(p+q),
                 Actual=which.max(tesla$sales))
```

```
##   Predicted Actual
## q      15.4      9
```

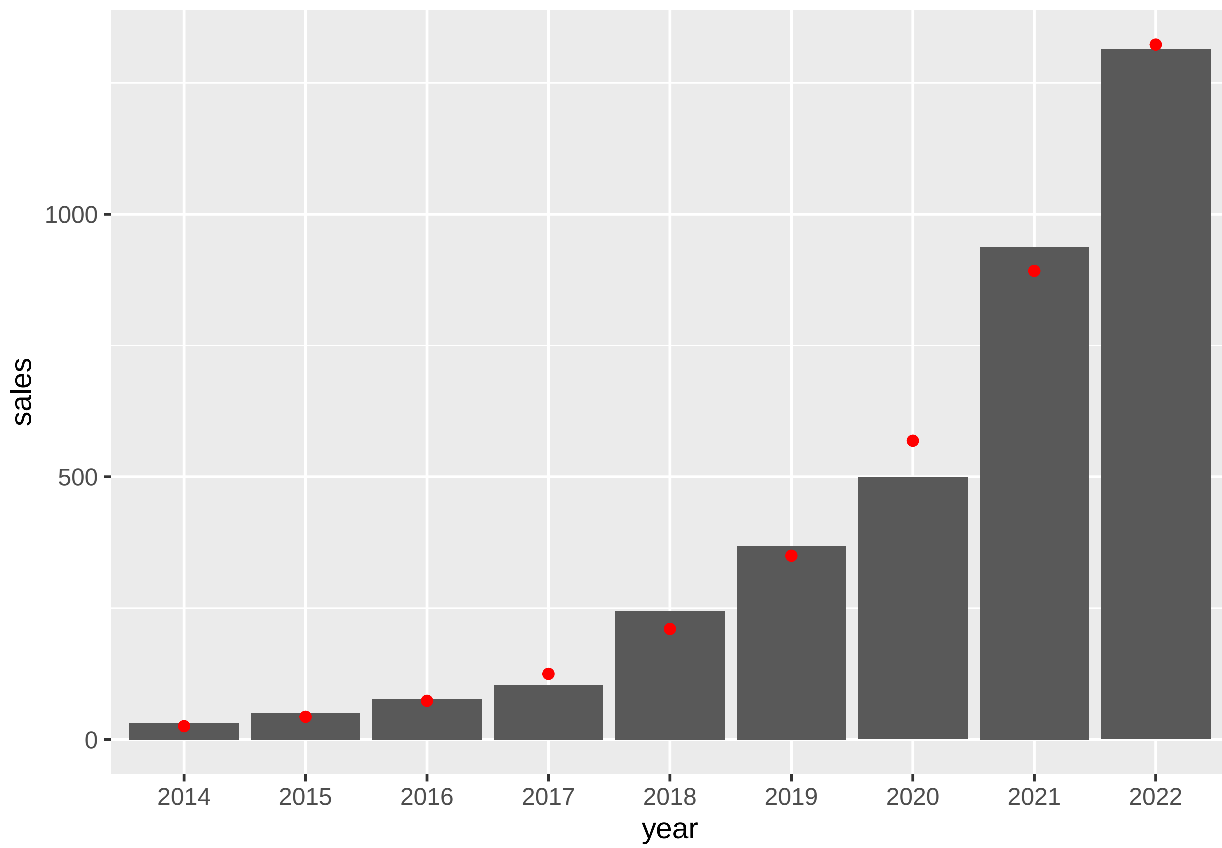
```r
## Defining bass function
bass.f <- function(t,p,q){ ((p+q)**2/p)*exp(-(p+q)*t)/
    (1+(q/p)*exp(-(p+q)*t))**2
  }
```

6. Estimate the number of adopters by time period. Thus, you will need to estimate the potential market share. You can use Fermi's logic here.

```r
## Modeling f(t) and visualizing it
time_ad = ggplot(data.frame(t = c(1:9)), aes(t)) +
  stat_function(fun = bass.f, args = c(p=0.0298, q=0.073)) +
  labs(title = 'f(t)')
ggarrange(time_ad, plot_tesla)
```




```r
## Predicting sales
tesla$pred_sales = bass.f(1:9, p = 0.000847, q = 0.540172)*17339.9
ggplot(data = tesla, aes(x = year, y = sales)) +
  geom_bar(stat = 'identity') +
  geom_point(mapping = aes(x=year, y=pred_sales), color = 'red')
```


1000
500
0
sales
2014
2015
2016
2017
2018
2019
2020
2021
2022
year