Marketing Analytics - Homework 1

Khoren Movsisyan

2024-02-24

1. Chosen Invention: Robot Coworker - "Sanctuary AI Phoenix"

https://time.com/collection/best-inventions-2023/6326928/sanctuary-ai-phoenix/

2. Look-alike innovation from the past: Amazon Echo - "A smart speaker with voice control."

For a product like the Sanctuary AI Phoenix, the market is relatively new and there is not much data that would be enough for this task. The only related data I was able to found were not for a single product or product line, but rather the whole robotics industry. That is why I decided to take a Amazon Echo, for which there is plenty of data, as a look-alike product.

Justification: While Robot Coworker and Amazon Echo are not the same product and have a lot of differences in terms of their specific functions, they share some key underlying principles. In a sense, Amazon Echo and similar products have paved the way for innovations like Sanctuary AI Phoenix in the future. Let me elaborate on the shared characteristics.

1) Both Amazon Echo and Sanctuary AI Phoenix have voice interaction as a primary mode of communication with the user.
2) Both products have AI incorporated in their core, and use it to function, to understand and respond to user commands, among other tasks.
3) The success and the adoption of voice assistant speakers like Amazon Echo demonstrate users' willingness to this kind of technologies. Users might be more open to embracing Robot Coworkers if they offer similar ease of use and clear benefits.

The connection between these two innovations can help us understand the potential impact of Robot Coworkers and how lessons learned from past inventions can influence their development and future success.

3. Data for the look-alike innovation:

The data chosen for the look-alike innovation is the number of units sold worldwide. The data includes sales from 2014 until 2023. Originally the data contained predicted sales for the years 2024 and 2025, however I have removed them from the calculation. There are two columns in the data, the first being the year and the second one being the units sold in millions of units.

```
data <- read_excel("amazon_echo_data.xlsx", sheet = "Data")
data <- head(data, n = -2)

data_with_predictions <- read_excel("amazon_echo_data.xlsx", sheet = "Data")
#This will be needed to compare the outputs of the 2 methods used

data
```

```
## # A tibble: 9 x 2
##    '2014' '0.6'
##    <chr>  <dbl>
## 1 2015      2.2
## 2 2016      8.5
## 3 2017     26.2
## 4 2018     32
## 5 2019     42
## 6 2020     53.9
## 7 2021     65
## 8 2022     82
## 9 2023     94
```

```r
colnames(data) <- c("Year", "Units")

data$Year <- as.numeric(data$Year)

data
```
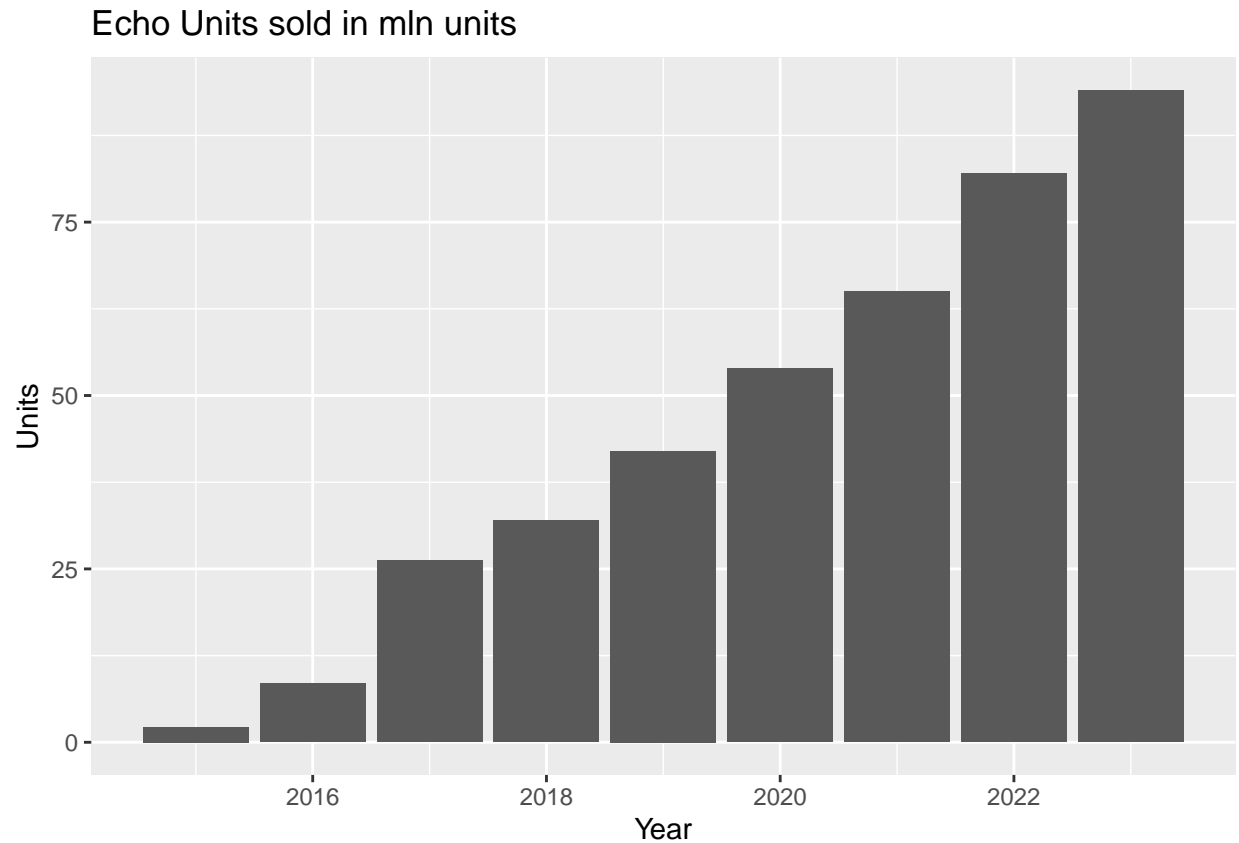
```
## # A tibble: 9 x 2
##     Year Units
##    <dbl> <dbl>
## 1  2015   2.2
## 2  2016   8.5
## 3  2017  26.2
## 4  2018  32
## 5  2019  42
## 6  2020  53.9
## 7  2021  65
## 8  2022  82
## 9  2023  94
```

Here with the bar plot we can visually see the distribution of the sold units by each year.

```r
units_sales <- ggplot(data, aes(Year, Units)) +
  geom_bar(stat='identity') +
  ggtitle("Echo Units sold in mln units")
units_sales
```

Echo Units sold in mln units

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

Here I have used two methods for the model predictions, first by using the nonlinear least squares method, and then using the diffusion library.

Method 1 - NLS:

```r
units <- data$Units
t <- 1:length(units)

estimation_by_nls <- nls(units ~ m * (((p+q)^2/p) * exp(-(p+q)*t)) / (1+(q/p) * exp(-(p+q)*t)) ^ 2,
            start = c(list(m=sum(units), p=0.02, q=0.4)))
summary(estimation_by_nls)
```

```
##
## Formula: units ~ 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 9.357e+02  2.136e+02    4.381 0.004662 **
## p 6.967e-03  1.028e-03    6.775 0.000505 ***
## q 3.986e-01  5.590e-02    7.130 0.000383 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.966 on 6 degrees of freedom
##
## Number of iterations to convergence: 10
## Achieved convergence tolerance: 1.882e-06
```

Method 2 - 'diffusion' library

```
estimation_by_diff = diffusion(units)

p=round(estimation_by_diff$w,4)[1]
q=round(estimation_by_diff$w,4)[2]
m=round(estimation_by_diff$w,4)[3]
estimation_by_diff
```

```
## bass model
##
## Parameters:
##                                Estimate p-value
## p - Coefficient of innovation   0.0083      NA
## q - Coefficient of imitation    0.4534      NA
## m - Market potential          760.6330      NA
##
## sigma: 4.4884
```
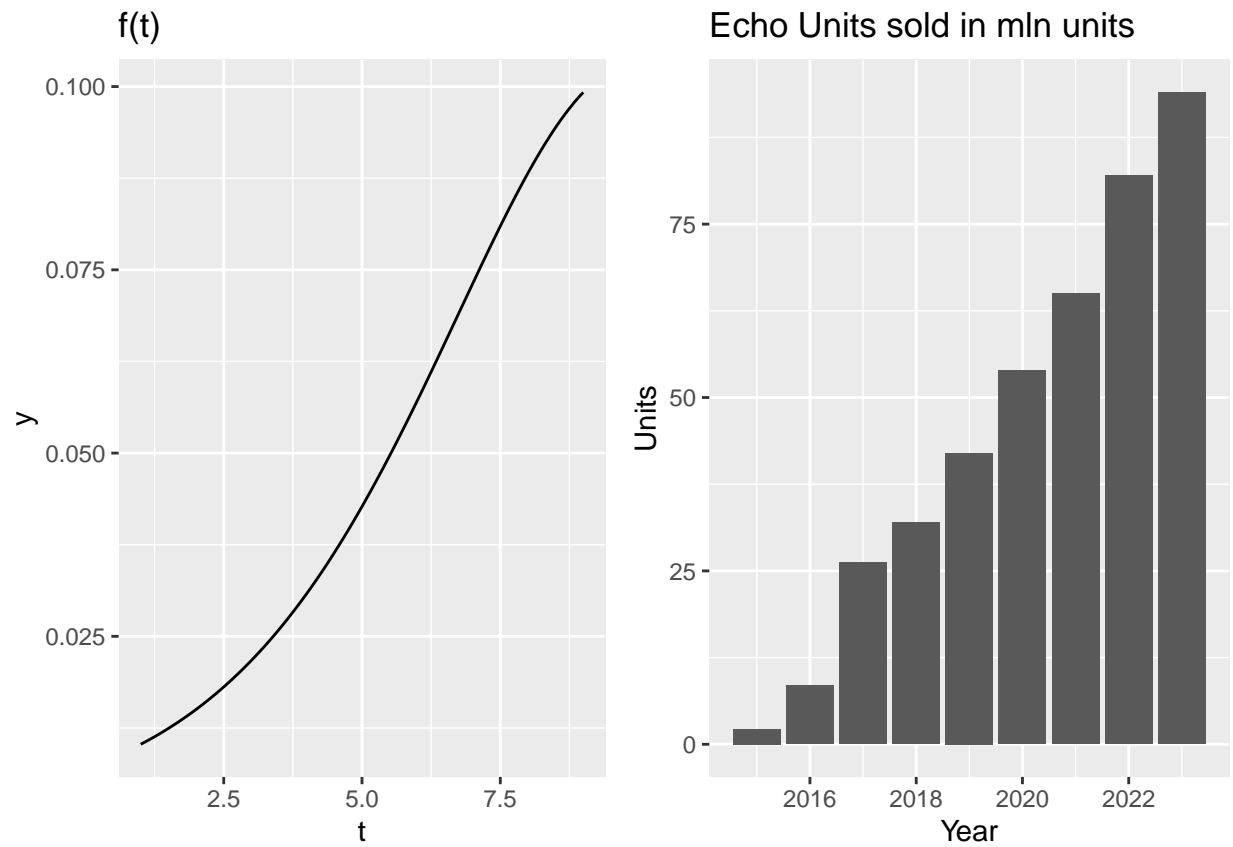
5. Parameter estimation and prediction:

```
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))
}
```
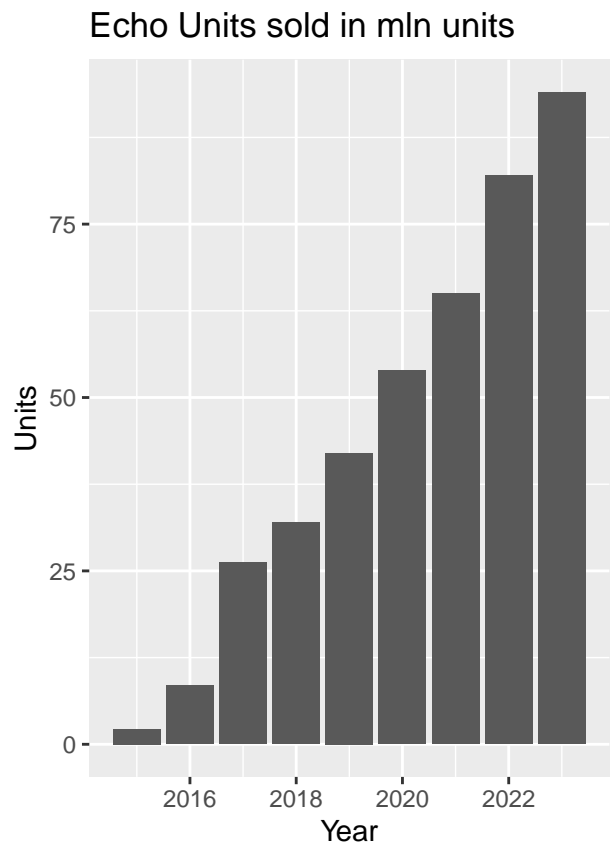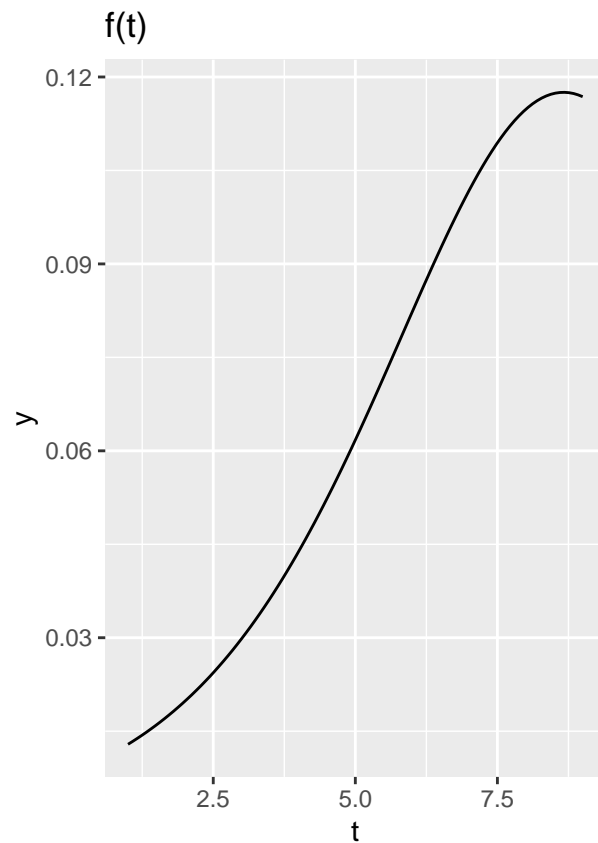
Parameters from nls():

```
ft_nls <- ggplot(data.frame(t=c(1:9)), aes(t)) +
  stat_function(fun = bass.f, args = c(p=0.006967, q=0.3986)) +
  ggtitle('f(t)')
ggarrange(ft_nls, units_sales )
```

f(t)

Echo Units sold in mln units

Parameters from the diffusion library Bass model:

```r
ft_diff <- ggplot(data.frame(t=c(1:9)), aes(t)) +
  stat_function(fun = bass.f, args = c(p=0.0083, q=0.4534)) +
  ggtitle('f(t)')
ggarrange(ft_diff, units_sales)
```

Now going back to the original data, where we had predicted sales for the years 2024 and 2025, let's visualize it.
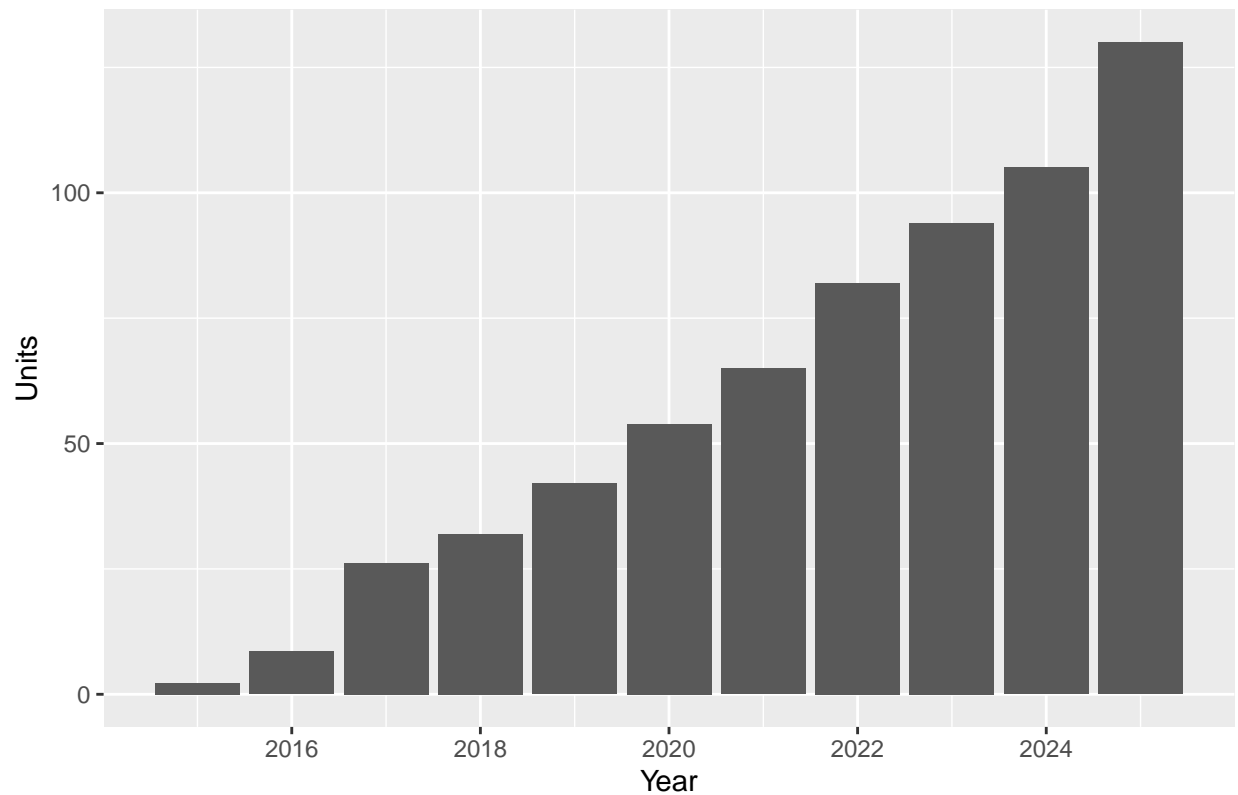
```r
colnames(data_with_predictions) <- c("Year", "Units")

data_with_predictions$Year <- as.numeric(data_with_predictions$Year)

unit_sales_with_prediction <- ggplot(data_with_predictions, aes(Year, Units)) +
  geom_bar(stat='identity') +
  ggtitle("Echo Units sold in mln units with 2024 and 2025 predictions")

unit_sales_with_prediction
```

Echo Units sold in mln units with 2024 and 2025 predictions

It becomes apparent then that with the parameters derived by the Nonlinear Least Squares method, we are able to better approximate the given data. That said, I will use those parameters, which are p=0.006967, q=0.3986 and M = 935.7 to do the prediction.

We can also vizualize the cumulative adoptions of the Echo units:

```
cum_adoptions = ggplot(data.frame(t = c(1, 9)), aes(t)) +
stat_function(fun = bass.F, args = c(p=0.006967, q=0.3986)) +
labs(title = 'Cumulative adoptions of Echo units')
cum_adoptions
```
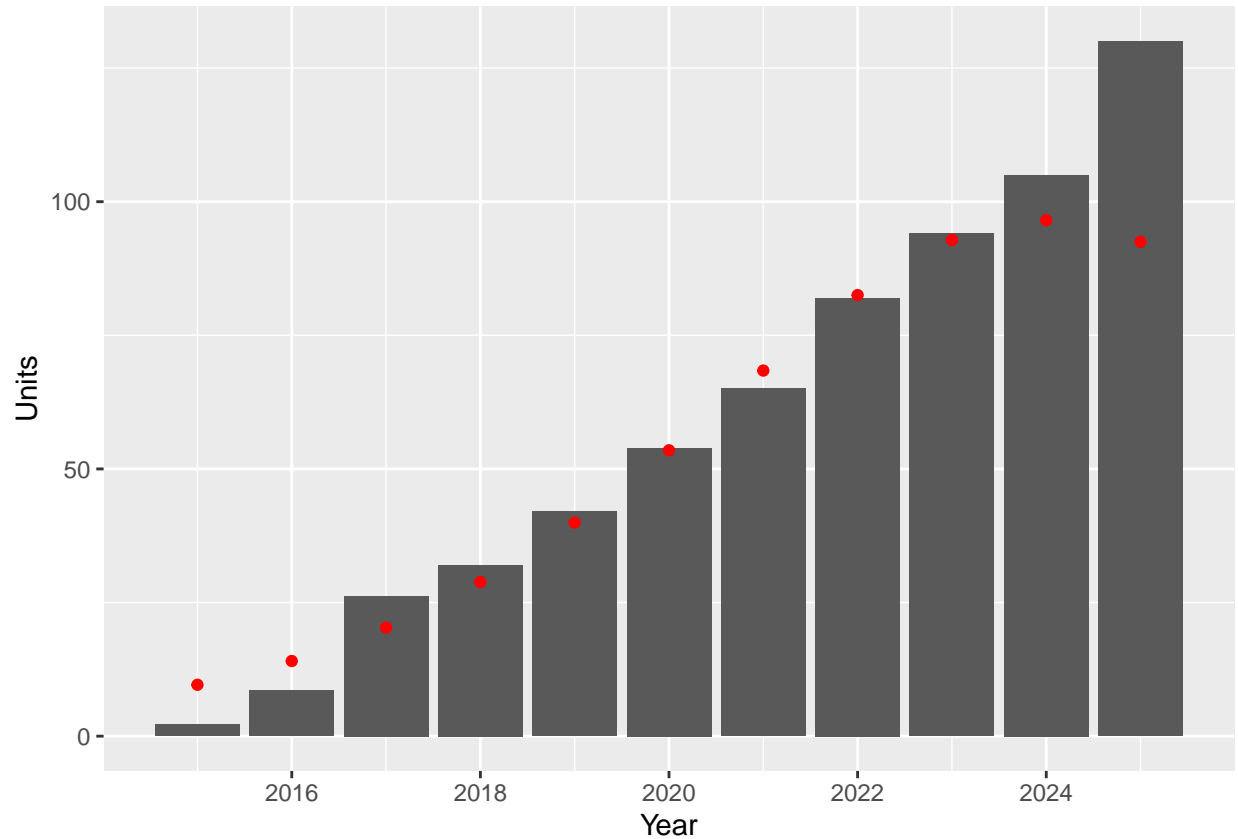
Cumulative adoptions of Echo units
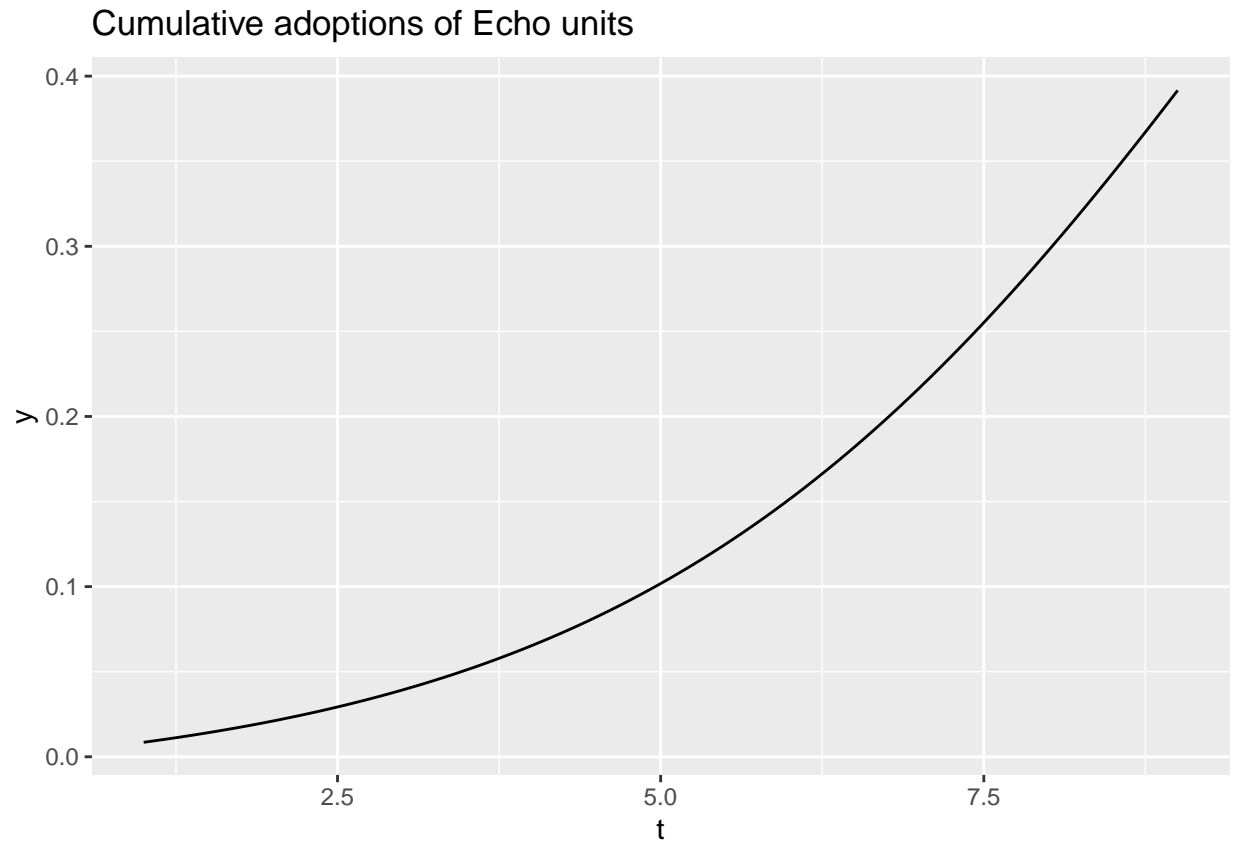
Prediction:

```
data_with_predictions$Pred_sales = bass.f(1:11, p=0.006967, q=0.3986) * 935.7

ggplot(data_with_predictions, aes(Year, Units)) + geom_bar(stat = 'identity') +
  geom_point(mapping = aes(Year, Pred_sales), color = 'red')
```

Here is my approach to estimate the number of adopters by period for the industrial robots market.

We know that in 2023 there were about 4.5 million robot units sold, and also that from 2016-2023 a total of 28.74 mln industrial and service robots have been sold.

There are about 334 million companies worldwide. We can assume, hypothetically, that the number of companies that will want to sometime in the future use robot coworkers to be about 20% of the total number. We get 66.8mln companies that could use industrial robots. We will also need to assume/predict an Average Adoption Rate per Adopter. Let's assume each potential adopter purchases an average of 10 industrial robots.

So our final Market Potential would be 66.8 mln * 10 = 668 mln units.

Innovators: 668,000,000 * 0.025 = 16,700,000 Early Adopters: 668,000,000 * 0.135 = 90,180,000 Early Majority: 668,000,000 * 0.34 = 227,120,000 Late Majority: 668,000,000 * 0.34 = 227,120,000 Laggards: 668,000,000 * 0.16 = 106,880,000

References

https://time.com/collection/best-inventions-2023/6326928/sanctuary-ai-phoenix/

https://www.statista.com/forecasts/1388096/global-robotics-volume-by-category

https://www.statista.com/statistics/1022701/worldwide-amazon-echo-unit-shipment/