Practical Session 1 - Modèles de Régression Linéaire

Richard CHEAM & Menghor THUO

03-10-2023

Contents

IV. Application: study your own data using a linear model with transformed data

a) Retrieve data

- To make sure the retrieved data is reliable, two different sites were referred to in order to compare each value obtained, precisely, year and revenue. As a result, the data were identical to one another.
- The data represents the growth of Amazon over the last 19 years, latest in 2022, based on the total amount of its income generated by the sale of goods and services (revenue) in $bn (billion-dollar).
- Amazon annual revenue data were retrieved from:

(n.d.). Business of Apps - Connecting the app industry. https://www.businessofapps.com/data/amazon-statistics/

Amazon annual net sales 2022. (2023, February 3). Statista. https://www.statista.com/statistics/266282/annual-net-revenue-of-amazoncom/

b) Organize data

```
#saving data in a vector
year <- c(2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013,
          2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022)
revenue <- c(6.92, 8.49, 10.71, 14.84, 19.17, 24.51, 34.2, 48.08, 61.09, 74.45,
             88.99, 107.01, 135.99, 177.87, 232.89, 280.52, 386.06, 469.82, 513.98)

#using data.frame() to obtain a table and write it using write.table()
df_ <- data.frame(year, revenue)
write.table(df_, file='amazon.txt')

#read data using read.table()
tab_ <- read.table(file = 'amazon.txt', sep = " ", header = TRUE)

tab_
```

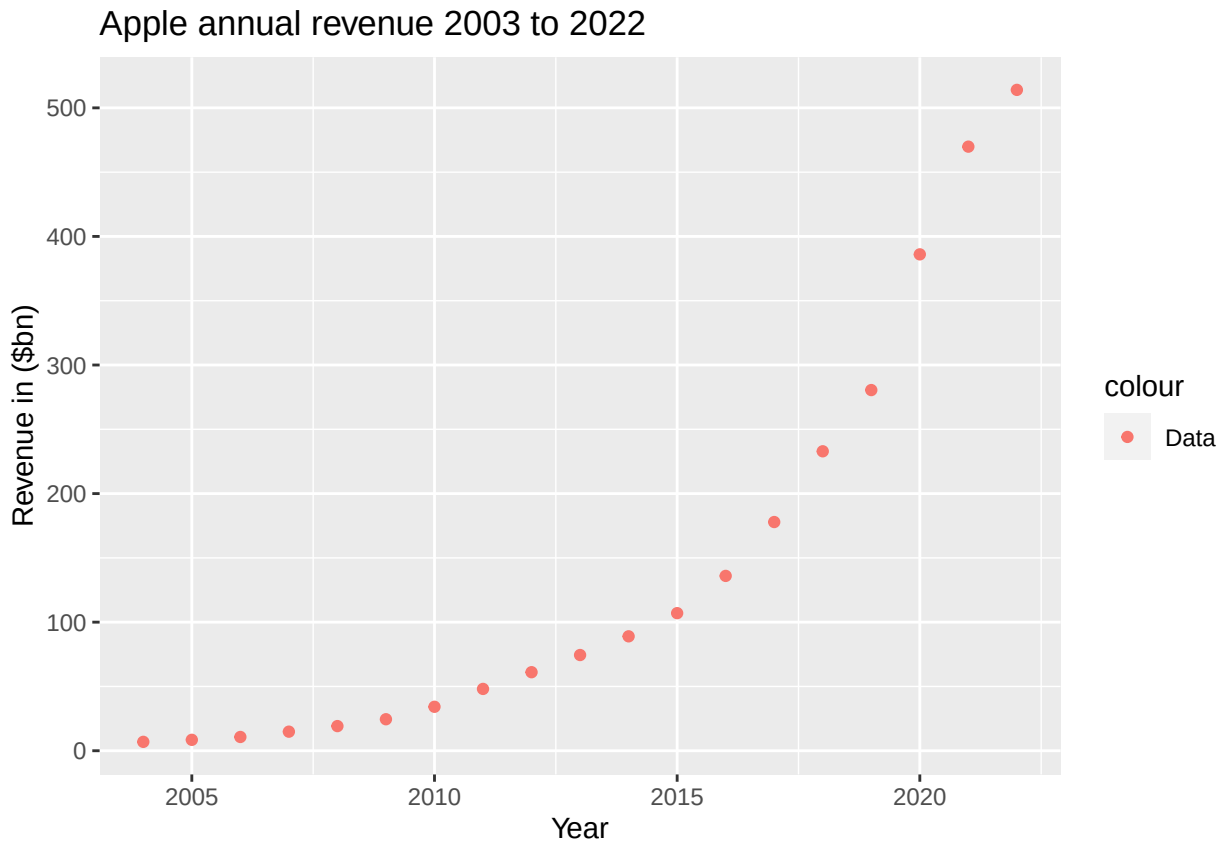
```
##    year revenue
## 1  2004    6.92
## 2  2005    8.49
## 3  2006   10.71
## 4  2007   14.84
## 5  2008   19.17
## 6  2009   24.51
## 7  2010   34.20
## 8  2011   48.08
## 9  2012   61.09
## 10 2013   74.45
## 11 2014   88.99
## 12 2015  107.01
## 13 2016  135.99
## 14 2017  177.87
```

```
## 15 2018  232.89
## 16 2019  280.52
## 17 2020  386.06
## 18 2021  469.82
## 19 2022  513.98
```

c) Visualize data

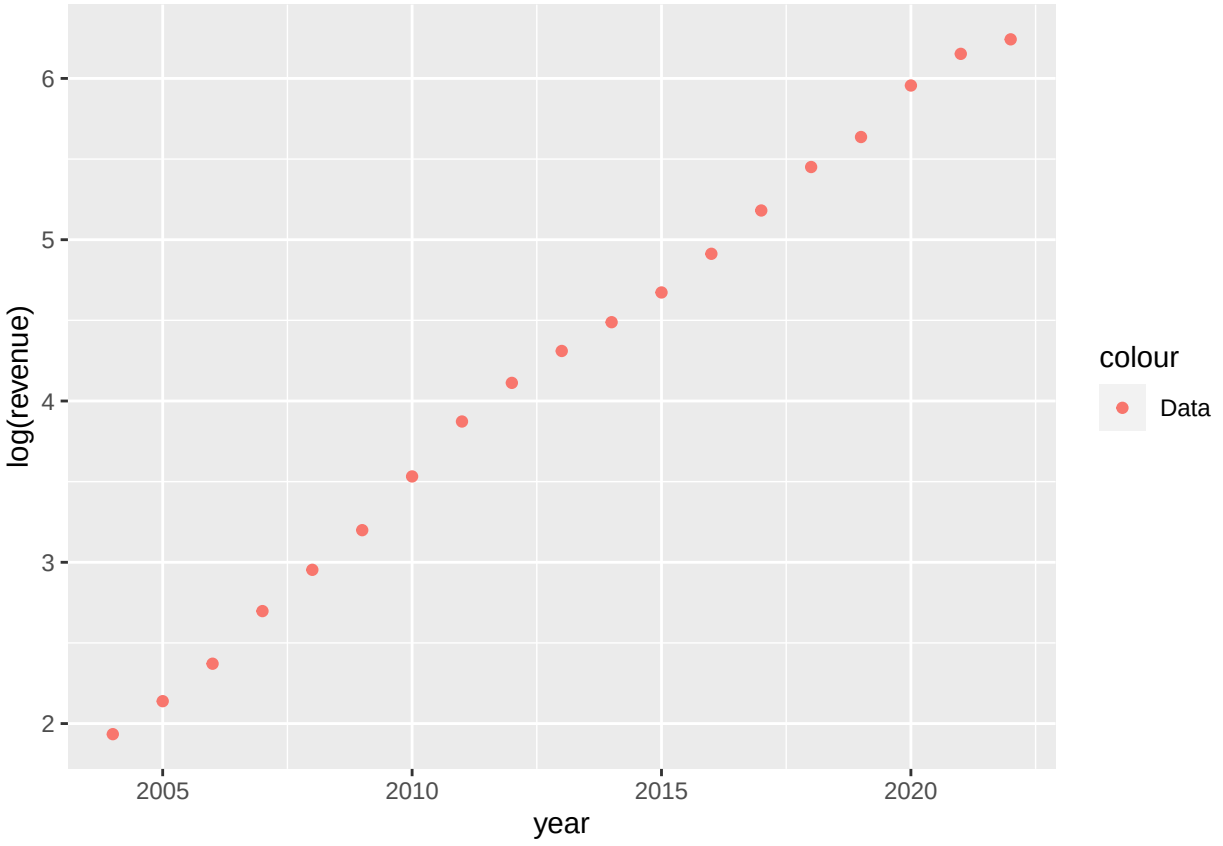
- This step is crucial in order to have a good visual on the data set, its trend, pattern, and so on, so that it facilitates on the choice of model selection based on the underlying pattern of the data. Such approach can save time and prevent errors in the modeling process.

```
ggplot(tab_, aes(x = year, y = revenue, color = "Data")) +
  geom_point() +
  labs(title = "Apple annual revenue 2003 to 2022", x = "Year", y = "Revenue in ($bn)")
```



- The graphic above shows the non-linear pattern (exponential growth), however, it is still appropriate to apply the logarithmic transformation. In this exercise, we will first fit the linear regression model, and then log-linear will be applied by taking the natural logarithm on the dependent variable (y) to turn the data into the linear pattern.

```
ggplot(tab_, aes(x = year, y = log(revenue), color = "Data")) +
  geom_point()
```



d) Linear model (bad model, just to see the difference)

```
# using revenue y-dependent variable and year as x-explanary variable
model_ <- lm(revenue~., tab_)
summary(model_)
```

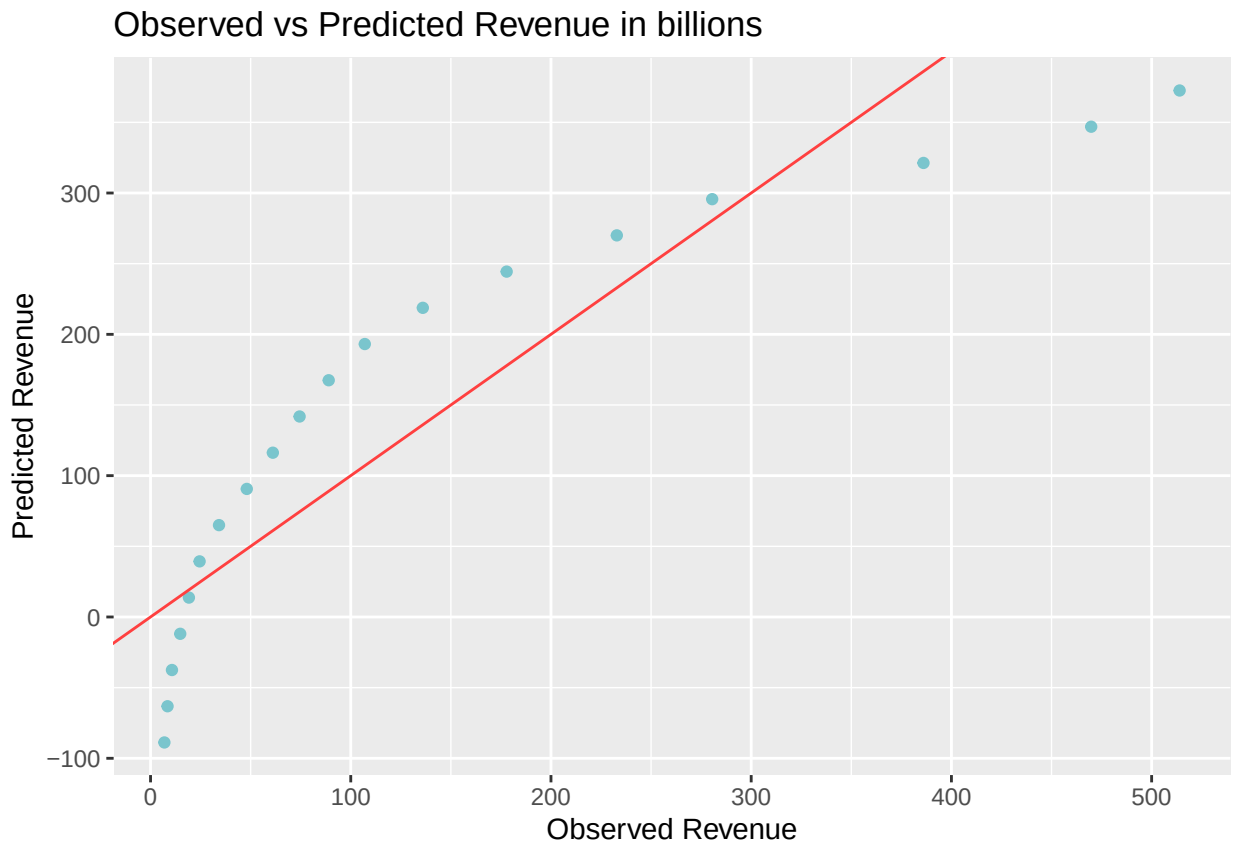
```
##
## Call:
## lm(formula = revenue ~ ., data = tab_)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -86.12 -60.84 -15.13  56.51 141.44
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -51450.54    6259.92  -8.219 2.52e-07 ***
## year            25.63       3.11   8.242 2.43e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 74.24 on 17 degrees of freedom
## Multiple R-squared:  0.7998, Adjusted R-squared:  0.7881
## F-statistic: 67.93 on 1 and 17 DF,  p-value: 2.428e-07
```

- The "year" coefficient has three asterisks (***) next to it. This indicates that it is highly statistically significant, with a very small p-value of 2.58e-13, which is much less than 0.001.
- While the R-squared value (approaching 1) suggests a good relation between the explanatory variable (year) and independent variable (revenue), the residual standard error indicates a bad fit to the data set.

```
observed_ <- tab_$revenue
predicted_ <- predict(model_, tab_)
```

d.1) Prediction study

```
ggplot(data.frame(observed_, predicted_), aes(x = observed_, y = predicted_)) +
  geom_point(color = "cadetblue3") +
  geom_abline(intercept = 0, slope = 1, color = "brown1") +
  labs(title = "Observed vs Predicted Revenue in billions", x = "Observed Revenue",
       y = "Predicted Revenue")
```



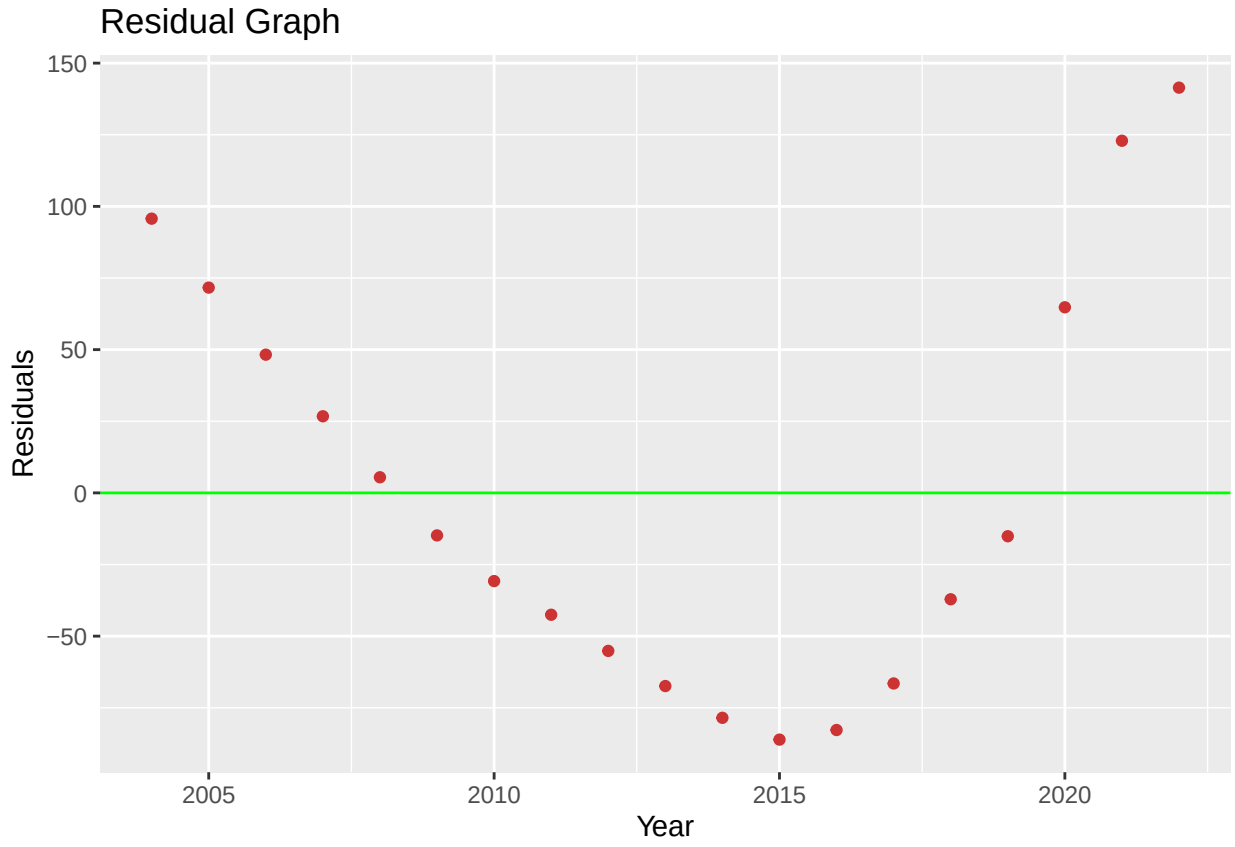
Visually, the data are not located on the bisector line indicating a great difference in value between the observed and predicted revenue in which we can conclude that the model is not good.

d.2) Residual study

- In the case of a multiple regression model, since there are multiple predictors, we plot the residuals $\epsilon = y - y_i$ versus the predicted values $\hat{y}_i$. But, since this is a simple linear regression model, we instead

plot the residuals versus the predictor x_i.

```
residuals_ <- model_$res
ggplot(data.frame(tab_$year, residuals_), aes(x = year, y = residuals_)) +
  geom_point(color = "brown3") +
  geom_hline(yintercept = 0, color = "green") +
  labs(title = "Residual Graph", x = "Year", y = "Residuals")
```




```
sqrt((1/length(observed_)) * sum((observed_ - predicted_)^2))
```

```
## [1] 70.22768
```

- There is a systematic curvature in the residuals (U-shaped) which indicates that the assumed linear relationship between Y (revenue) and X (year) is not reasonable since there is still some information in the residuals. In addition, the RMSE of 70.22768 is quite high as well, hence, the model needs to be changed.
- So we will now look into another model which is a lot better for this trend.

e) Log-linear model

- By applying the logarithmic transformation, we then obtained the following function:

$$ln(Y) = \beta_0 + \beta_1 X + \epsilon$$

- where X is the explanatory variable (year) and Y is the response variable (revenue), and ϵ is the residual.

```
#transform the y-dependent variable (revenue) using logarithm
logTab_ <- tab_
logTab_$revenue <- log(tab_$revenue)
#fit log-linear model using lm() function where year is the explanatory variable
logModel_ <- lm(revenue~., logTab_)
summary(logModel_)
```

```
##
## Call:
## lm(formula = revenue ~ ., data = logTab_)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.17130 -0.03378 -0.01814  0.03806  0.16381
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.907e+02  7.346e+00  -66.80   <2e-16 ***
## year         2.459e-01  3.649e-03   67.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08712 on 17 degrees of freedom
## Multiple R-squared:  0.9963, Adjusted R-squared:  0.996
## F-statistic:  4539 on 1 and 17 DF,  p-value: < 2.2e-16
```

- The "year" coefficient once again has a high statistical significance with an extremely small p-value $< (2)10^{-16}$, which is a lot less than 0.001.
- According to the summary of the model, it can be written as:

$$ln(Y) = -4.908(10^2) + 2.459(10^{-1})X + \epsilon$$

- For this model, we have a better $R^2 = 0.9963 \approx 1$ as it corresponds to the cosinus of the angle between the vector of the predicted value and the vector of the observed value, so the closer the angle approaches 0 (integer solution, $R^2 = cos^2\omega = 0.9963 \Rightarrow \omega \approx 0$), the better the model is. Plus, the residual standard is really low (0.08712) which suggests a high predictive accuracy.

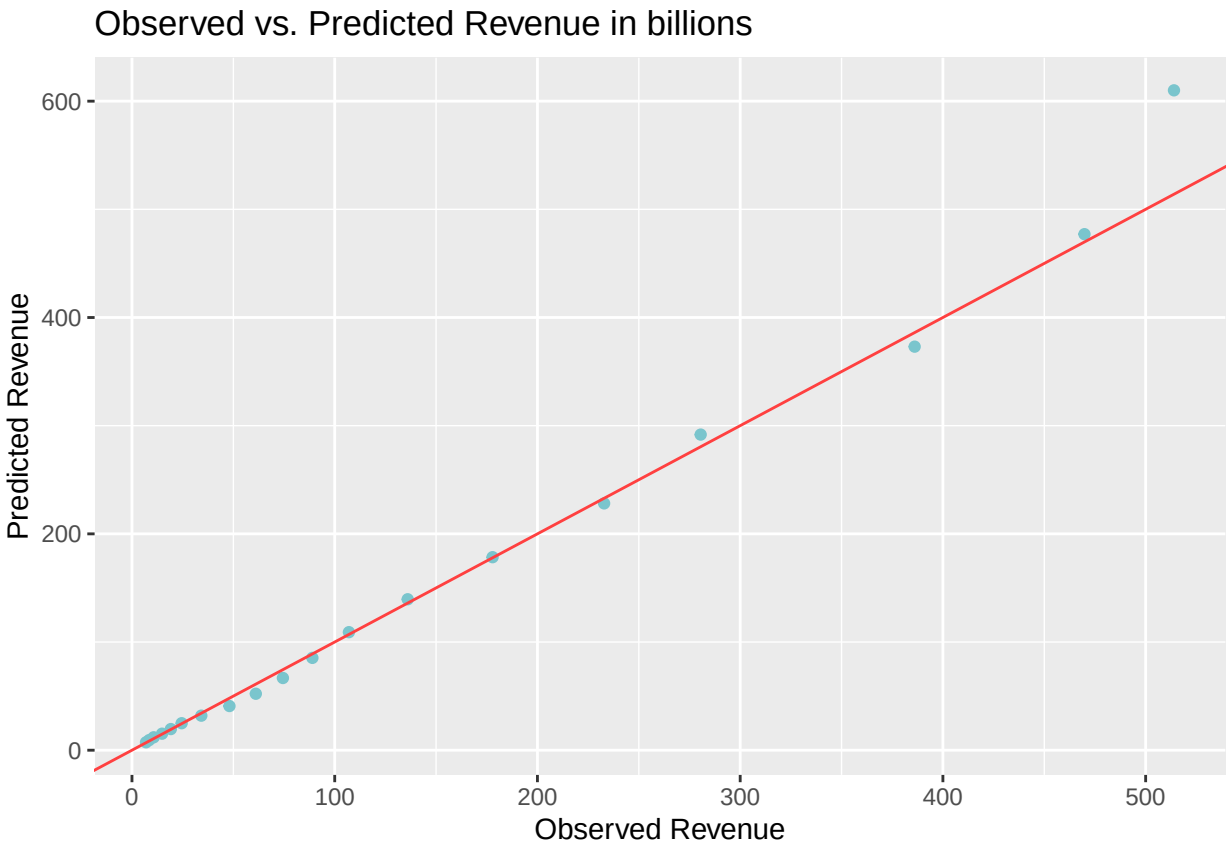
e.1) Prediction study

- Since this is log-linear model, we then would have a prediction function $\hat{y}$ as below:

$$\hat{Y} = exp(\beta_0 + \beta_1 X)$$

```
logPredicted_ <- exp(predict(logModel_))

ggplot(data.frame(observed_, logPredicted_), aes(x = observed_, y = logPredicted_)) +
  geom_point(color = "cadetblue3") +
  geom_abline(intercept = 0, slope = 1, color = "brown1") +
  labs(title = "Observed vs. Predicted Revenue in billions", x = "Observed Revenue",
       y = "Predicted Revenue")
```

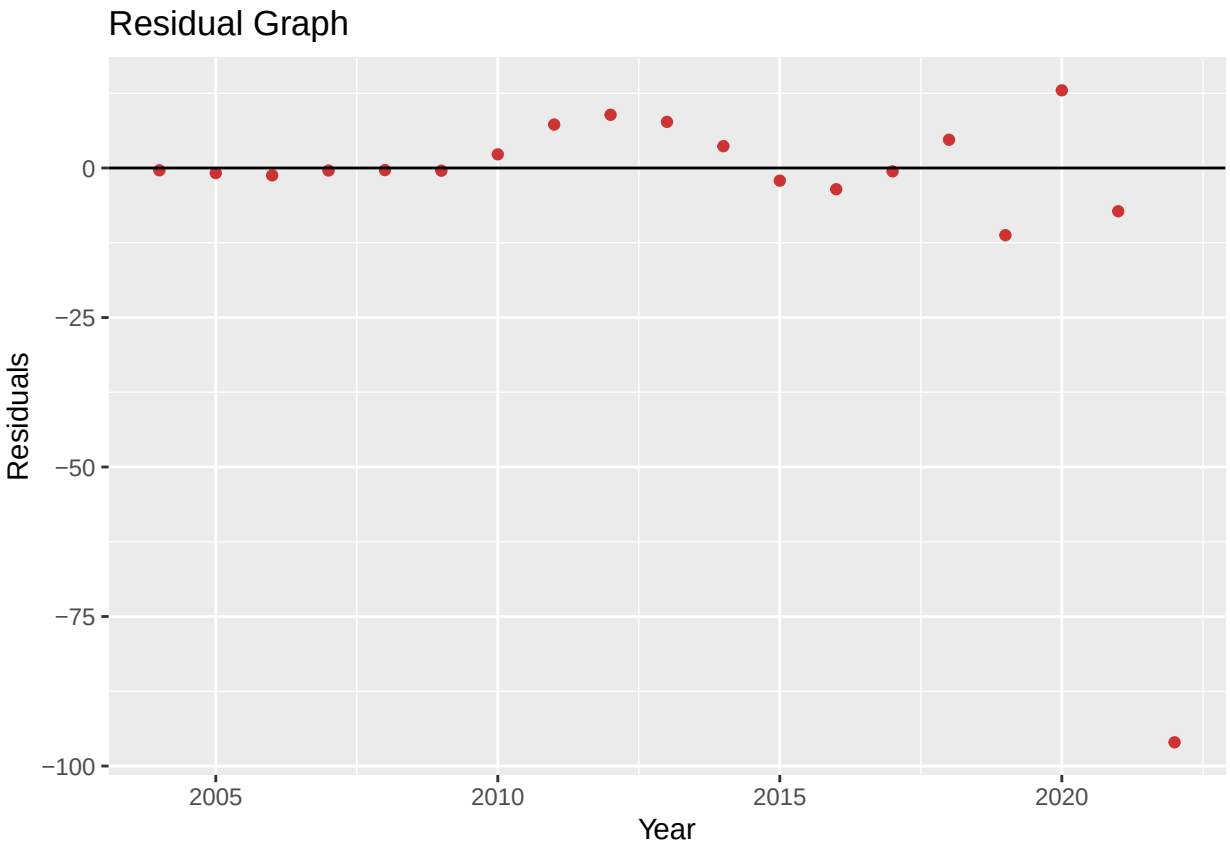


- In conclusion, in accordance to the graphic above, we can see that the log-linear model, basically a logarithmic transformation of linear model, fitted the model better as the scatter points lie on the bisector line, which represents more accurately the growth trend of Amazon over the last 19 years. However, an outlier appeared in which we will see in the residual study below.

e.2) Residual study

```
logResidual_ <- observed_ - logPredicted_

ggplot(data.frame(tab_$year, logResidual_), aes(x = year, y = logResidual_)) +
  geom_point(color = "brown3") +
  geom_hline(yintercept = 0) +
  labs(title = "Residual Graph", x = "Year", y = "Residuals")
```



- The points remain mostly on or near the horizontal line at zero (with one outlier), this can indicate that this model is doing a reasonably good job at capturing the relationship between the explanatory variable and the response variable, and on average, this model's predictions are reasonably close to the actual values.

```
logRMSE <- sqrt((1/length(observed_)) * sum((observed_ - logPredicted_)^2))
logRMSE
```

```
## [1] 22.73713
```

- With RMSE of 22.73713 and the presence of the single isolation in the residual plot, it suggests that this model has relatively good predictive performance but may make occasional errors.

V. Cookies Study

```
dataset <- read.csv("cookies.csv", header = TRUE)
dim(dataset)
```

```
## [1]  32 701
```

- Indicating 32 observations with a y-dependent variable and $x_{i=1,2,...,700}$-independent variables.

a) Features extraction

```
#separate response variable from explanatory variables
Y <- as.vector(dataset$fat)
X <- as.matrix(dataset[2:701])
```

```
#calculate mean of each cookie by applying mean() function on each row of matrix X
mean_values <- apply(X , 1, mean)
mean_values
```

```
##  [1] 0.9851499 1.0355417 1.0010620 1.0280481 1.0655011 1.0840236 1.0872053
##  [8] 1.1780192 1.1456231 1.0535643 1.1713458 1.0352136 1.1147738 1.1765116
## [15] 1.0834190 1.1122142 1.0728825 1.0127377 1.0029307 1.0676309 0.9929406
## [22] 1.0346131 1.0775659 1.1374148 1.0786413 1.1381555 1.0310771 1.1354353
## [29] 1.0280111 1.0069282 1.1306734 0.9419162
```

```
#calculate standard deviation of each cookie
sd_values <- apply(X , 1, sd)
sd_values
```

```
##  [1] 0.4111868 0.4123933 0.4025158 0.4040351 0.4158252 0.4262425 0.4572639
##  [8] 0.4822934 0.5025094 0.4610220 0.4641904 0.4531590 0.4642941 0.4871777
## [15] 0.4576513 0.4675242 0.4685757 0.4279332 0.4255500 0.4547905 0.4256206
## [22] 0.4366263 0.4368955 0.4730211 0.4594111 0.4657774 0.4457069 0.4599680
## [29] 0.4439346 0.4243892 0.4720336 0.4006358
```

```
#calculate the minimum value of each cookie
minimun <- apply(X , 1, min)
minimun
```

```
##  [1] 0.259270 0.266864 0.251654 0.277777 0.288328 0.284625 0.275583 0.302558
##  [9] 0.295653 0.259807 0.297032 0.259899 0.295240 0.300989 0.276969 0.287420
## [17] 0.270242 0.264721 0.257326 0.277993 0.255322 0.269356 0.282209 0.295327
## [25] 0.273962 0.291417 0.262233 0.293434 0.258019 0.260089 0.285933 0.245734
```

```
#calculate the maximum of each cookie
maximum <- apply(X , 1, max)
maximum
```

```
##  [1] 1.73946 1.66273 1.60960 1.63881 1.70320 1.74356 2.03196 2.15614 2.50716
## [10] 2.11086 2.02262 2.08371 2.05526 2.25429 2.04356 2.12978 2.15661 1.88404
## [19] 1.86586 2.03417 1.87919 1.91287 1.89490 2.08916 2.06176 2.08172 2.01232
## [28] 2.06373 1.92981 1.84227 2.11972 1.73909
```

```
#calculate slope for each cookie
slopes <- numeric(32)
x <- seq(1,700,1)
for (i in 1:32){
  mod <- lm(unlist(dataset[i, 2:701])~x)
slopes[i] <- coef(mod)[2]
}
slopes
```

```
##  [1] 0.001914311 0.001898164 0.001860203 0.001861782 0.001910926 0.001967228
##  [7] 0.002126801 0.002235741 0.002310536 0.002141247 0.002136442 0.002104337
## [13] 0.002155329 0.002254186 0.002120034 0.002167335 0.002171707 0.001987932
## [19] 0.001981711 0.002118449 0.001984380 0.002022981 0.002030155 0.002192669
## [25] 0.002125721 0.002155873 0.002069382 0.002127916 0.002064303 0.001969204
## [31] 0.002189921 0.001870085
```

b) Regression model

```
#create a data frame with 5 independent variables and fit linear model
data_tab <- data.frame(Y, mean_values, sd_values, minimun, maximum, slopes)
modreg <- lm(Y~., data=data_tab)
summary(modreg)
```

```
##
## Call:
## lm(formula = Y ~ ., data = data_tab)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.37145 -0.52481  0.03787  0.53121  1.22305
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.728e+00  5.739e+00  -0.475  0.63851
## mean_values  9.563e+00  1.623e+01   0.589  0.56092
## sd_values    3.358e+02  1.404e+02   2.392  0.02430 *
## minimun      5.956e-01  3.321e+01   0.018  0.98583
## maximum     -3.338e+00  5.166e+00  -0.646  0.52381
## slopes      -6.614e+04  2.340e+04  -2.826  0.00893 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7837 on 26 degrees of freedom
## Multiple R-squared:  0.7167, Adjusted R-squared:  0.6622
## F-statistic: 13.15 on 5 and 26 DF,  p-value: 1.939e-06
```

- The "slopes" coefficient has two asterisks (**) next to it. This indicates that it is moderately statistically significant, with a p-value of 0.00893, which is less than 0.01. While, the "standard deviation" has one asterisk (*) next to it meaning it is marginally statistically significant, with a p-value of 0.02430, which is less than less than 0.05. Therefore, the "slopes" variable is statistically significant at a significance level of 0.01, while the "sd" variable is at the significance level of 0.05.
- The model can be written as:

$$Y = -2.728 + 9.563X_1 + 335.8X_2 + 0.5956X_3 - 3.338X_4 - 66140X_5 + \epsilon$$

```
r_sqrt <- summary(modreg)$r.squared
r_sqrt
```

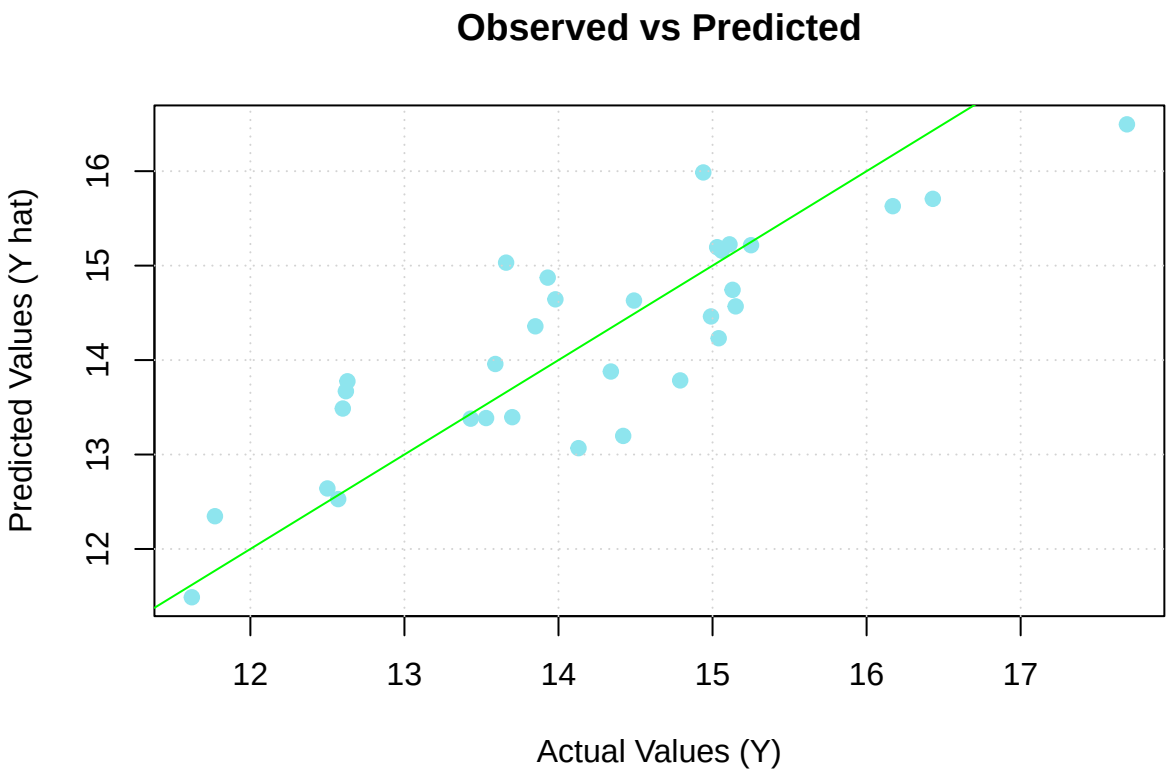
```
## [1] 0.7166928
```

- The value $R^2 = 0.7166928$ is a positive indicator for this model in which it the predictor variables explain roughly 71.67% of the variance in the outcome we are trying to predict. However, other factors should still be considered in order to conclude whether this model fits well with the data set.

b.1) Prediction study

```
Y_hat <- predict(modreg, data_tab)

plot(Y, Y_hat,  pch = 19, col = 'cadetblue2', xlab = "Actual Values (Y)",
     ylab = "Predicted Values (Y hat)", main = "Observed vs Predicted")
grid()
abline(lm(Y ~ Y_hat), col = 'green')
```




```
rmse <- sqrt(mean((Y - Y_hat)^2))
rmse
```

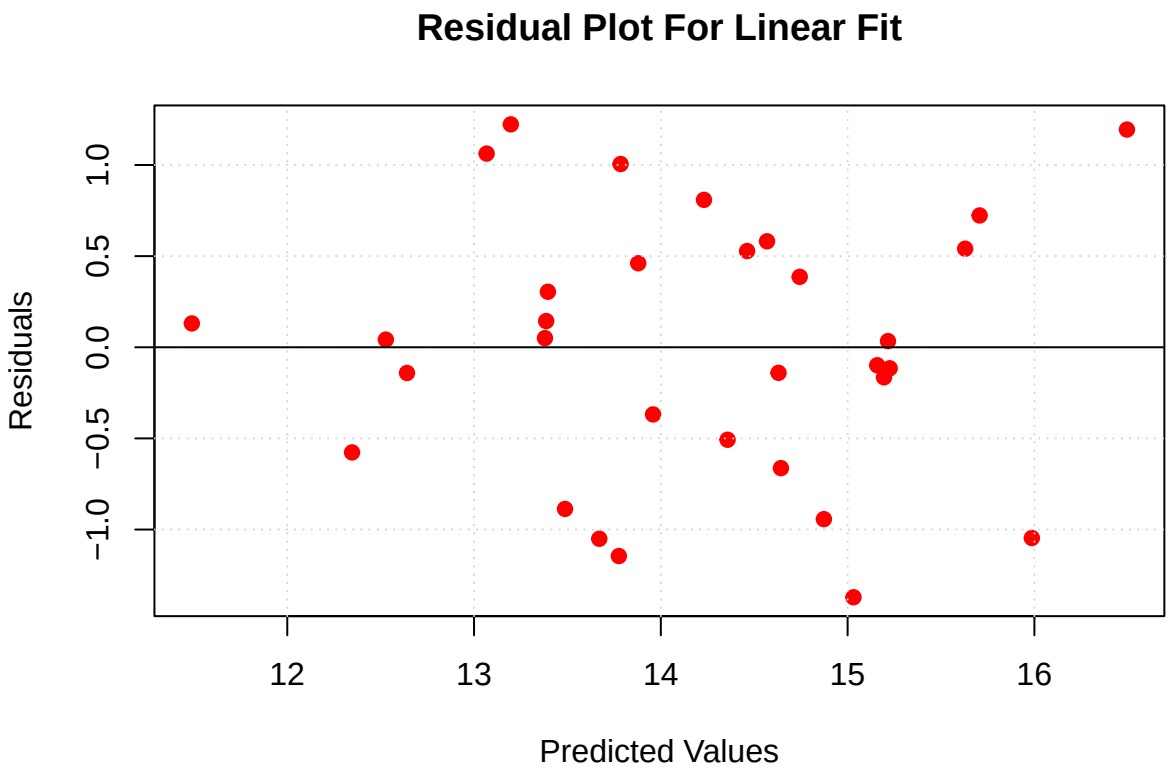
```
## [1] 0.7064409
```

- The RMSE value suggests a relatively small error which once again shows a good sign within the application of linear regression model. Nevertheless, based on the $(Y, \hat{Y})$ graph above, it seems like there are quite a decent amount of data points in which they are fairly far away from the bisector line. In the final analysis, the difference between the actual values and the predicted values demonstrate an inadequate quality of this model which we can conclude that this model is not that good.

b.2) Residual study

```
epsilon <- modreg$res

plot(Y_hat, epsilon, pch = 19, col = "red", xlab = "Predicted Values",
     ylab = "Residuals", main = "Residual Plot For Linear Fit")
grid()
abline(0, 0)
```



- The $(\hat{\epsilon}, \hat{y})$ plot shows no discernible pattern. Thus, there is no information to be captured for such random distribution of residual values.