

An introductory guide to Ordinary Least Squares: Applying CAPM to Amazon stock

Juan David Rincón Mora

2023

1. Introduction.

The main objective of this notebook is to estimate a model of economic interest using the **Ordinary Least Squares (OLS)** method. The process of matrix construction, the estimation of statistics and variables, and the interpretation of the results obtained will be described in detail.

It is important to note that this notebook focuses on the development of the different estimators and statistics necessary for the estimation of the econometric model in terms of the programming framework of R. It will not go into the mathematical development behind them, beyond presenting the corresponding equations, formulas, and interpretations.

Finally, it is relevant to mention that the mathematical, econometric, and statistical content presented in this notebook is largely based on the approach and methodology proposed by Judge, Hill, Griffiths, Lütkepohl, and Lee (1988).

2. Model.

In this notebook, it will be estimated the **Capital Asset Pricing Model (CAPM)**, whose objective is to explain the profitability of a financial asset as a function of the market risk it assumes. According to Berk and DeMarzo (2020), the simplest mathematical model of the CAPM is expressed as follows:

$$r_i = r_f + \beta(E[R_{mkt}] - r_f)$$

Where r_i represents the return on the asset to be valued, r_f is the risk-free rate, β is the sensitivity of the asset's return to changes in the market portfolio return, therefore, it represents a measure of systemic risk, and $E[R_{mkt}]$ is the expected value of the market return, from which the risk-free rate is subtracted, which corresponds to the market risk premium.

To estimate the model econometrically, first, a small transformation is made to the mathematical expression. The risk-free rate term r_f is moved to the left-hand side of the equation so that the dependent variable is the risk premium generated by the asset itself in relation to the risk-free rate. This results in a new parameter α , which represents the intercept and whose interpretation is the natural risk premium of the asset. In addition, the perturbation term ε_t is added, which collects all the unobservable information that affects the model. Thus, the equation of the econometric model is as follows:

$$(r_t^i - r_t^f) = \alpha + \beta(r_t^{mkt} - r_t^f) + \varepsilon_t$$

It is important to note that the model is built from **time series data**, which explains that each variable is subindexed to a particular time t . The superscripts indicate to which type of return each component corresponds according to the initial notation of the mathematical model.

2.1. Variables.

In this case, **Amazon** has been chosen as the asset r^i to value, identified by its ticker AMZN. Since this stock is mainly listed on the **NASDAQ Composite index**, this index has been taken as the market return r^{mkt} . For the risk-free rate r^f was used the monthly market rate of the **10-year U.S. Treasury Bond**, represented by the UST10 indicator.

It has taken the values corresponding to the periods from **January 2015 to December 2022**. It is important to clarify that in the case of AMZN and NASDAQ, prices have not been considered directly, but the monthly yield, i.e. the percentage change with respect to the last price of each month.

The database used for this exercise is available at the following link: [*Amazon CAPM data*](#). The first values of the database are presented below:

```
#install.packages("readxl")
library(readxl)

# Import data.
file1 <- file.choose()
DATA <- as.data.frame(read_excel(file1, sheet = 1))

# First values of data.
head(DATA)
```

##	Date	AMZN	NASDAQ	UST10y	AMZN Rtrn	NASDAQ Rtrn	UST10 mv
## 1	2015-01-30	17.73	4635.24	0.01639	0.14239691	-0.021285671	0.001355679
## 2	2015-02-27	19.01	4963.53	0.01996	0.07219402	0.070824812	0.001648308
## 3	2015-03-31	18.61	4900.88	0.01927	-0.02104156	-0.012622065	0.001591823
## 4	2015-04-30	21.09	4941.42	0.02035	0.13326169	0.008271984	0.001680219
## 5	2015-05-29	21.46	5070.02	0.02128	0.01754386	0.026024908	0.001756269
## 6	2015-06-30	21.70	4986.87	0.02349	0.01118360	-0.016400330	0.001936736

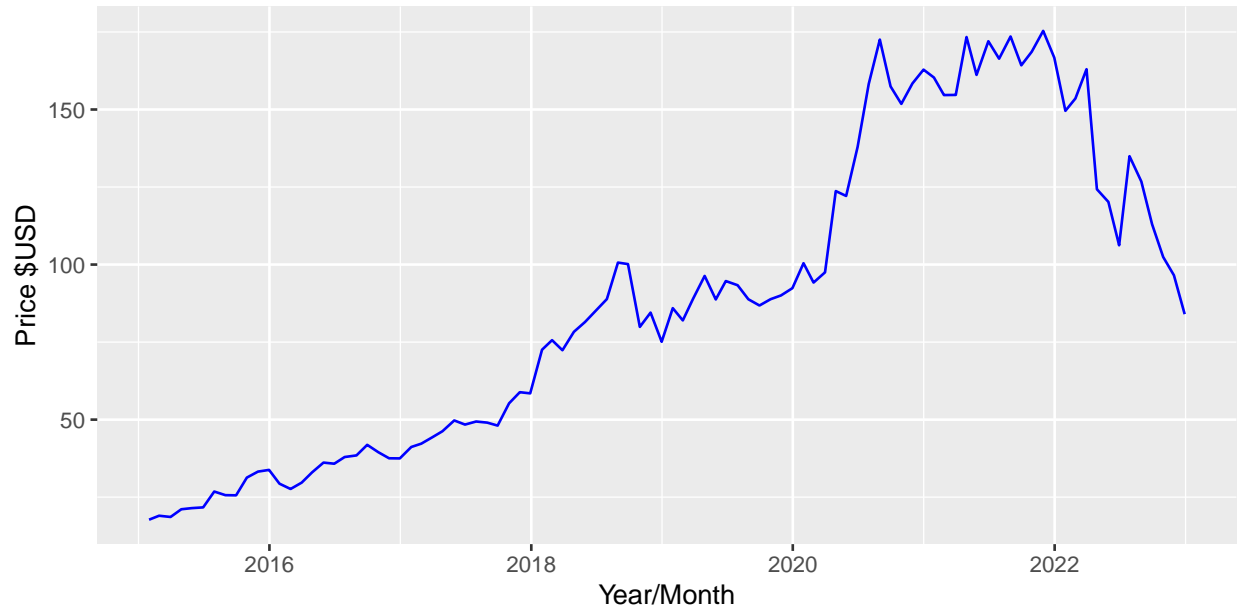
2.1.1. Graphical representation.

```
#install.packages("ggplot2")
library(ggplot2)

# Amazon Prices Graph.
ggplot(data=DATA, aes(x = Date, y = AMZN)) +
  geom_line(color = "blue") +
  labs(x = "Year/Month", y = "Price $USD",
       title = "Amazon latest monthly price (AMZN)",
       subtitle = "January 2015 - December 2022")
```

Amazon latest monthly price (AMZN)

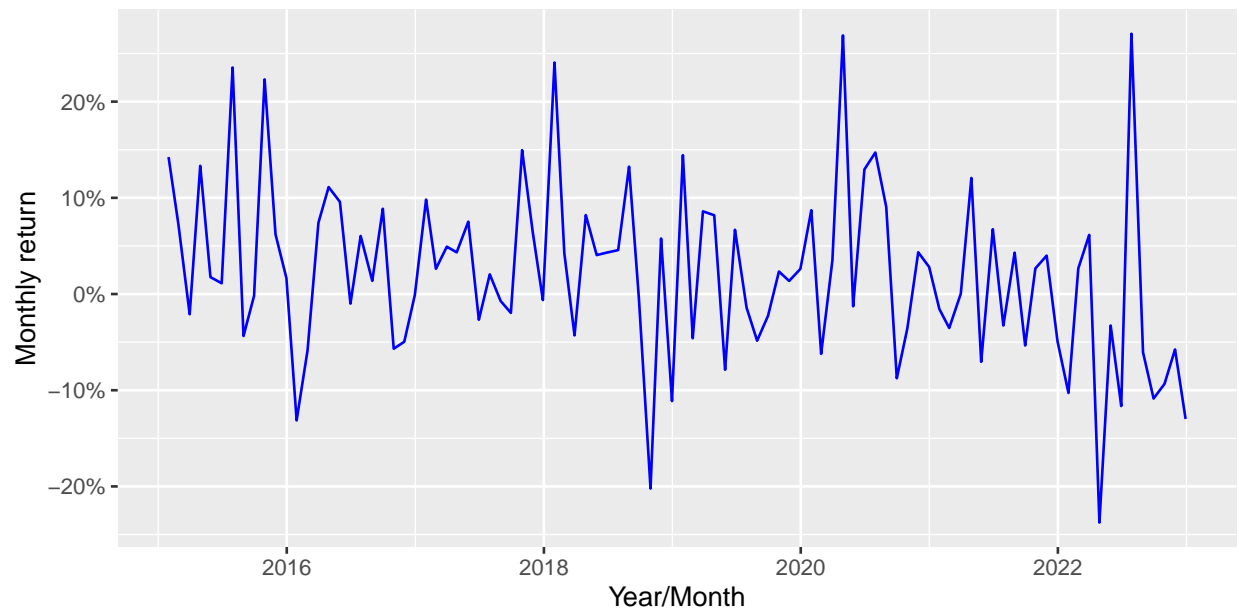
January 2015 – December 2022



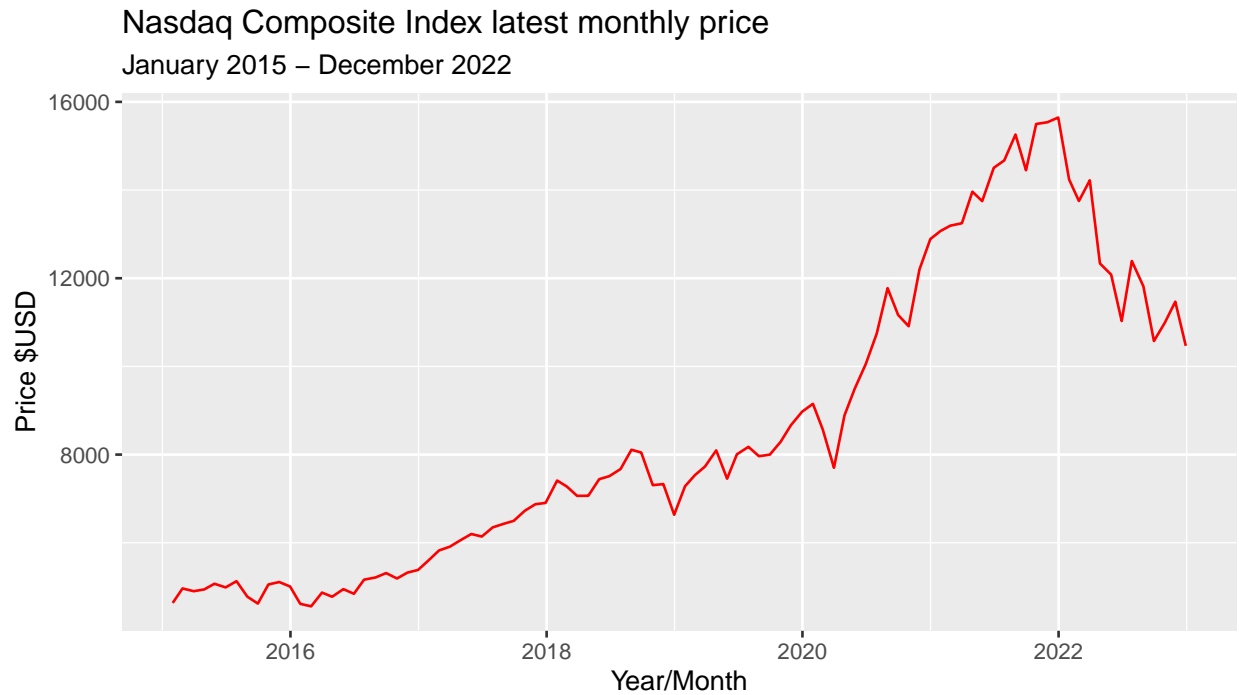
```
# Amazon Returns Graph.  
ggplot(data=DATA, aes(x = Date, y = `AMZN Rtrn`)) +  
  geom_line(color = "blue") +  
  labs(x = "Year/Month", y = "Monthly return",  
       title = "Amazon Monthly Return (AMZN)",  
       subtitle = "January 2015 - December 2022") +  
  scale_y_continuous(labels = scales::percent)
```

Amazon Monthly Return (AMZN)

January 2015 – December 2022



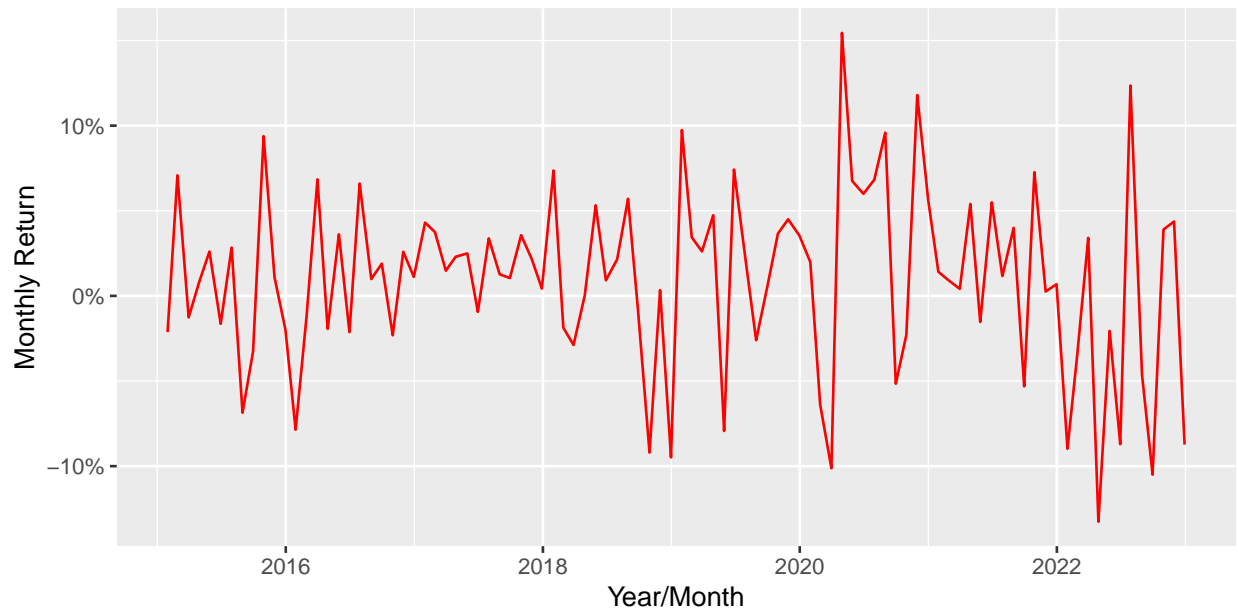
```
# NASDAQ Prices Graph.
ggplot(data=DATA, aes(x = Date, y = NASDAQ)) +
  geom_line(color = "red") +
  labs(x = "Year/Month", y = "Price $USD",
       title = "Nasdaq Composite Index latest monthly price",
       subtitle = "January 2015 - December 2022")
```



```
# NASDAQ Returns Graph.
ggplot(data=DATA, aes(x = Date, y = `NASDAQ Rtrn`)) +
  geom_line(color = "red") +
  labs(x = "Year/Month", y = "Monthly Return",
       title = "Nasdaq Composite Index Monthly Return",
       subtitle = "January 2015 - December 2022") +
  scale_y_continuous(labels = scales::percent)
```

Nasdaq Composite Index Monthly Return

January 2015 – December 2022



```
# 10-year Treasury Bonds Graph.
ggplot(data=DATA, aes(x = Date, y = `UST10 mv`)) +
  geom_line(color = "black") +
  labs(x = "Year/Month", y = "Monthly Return",
       title = "Monthly rate 10-year U.S. Treasury Bonds",
       subtitle = "January 2015 - December 2022") +
  scale_y_continuous(labels = scales::percent)
```

Monthly rate 10-year U.S. Treasury Bonds

January 2015 – December 2022



2.1.2. Construction of model variables.

First, each variable is extracted in individual vectors:

```
# Amazon, Ri.
ri <- DATA$`AMZN Rtrn`

# Nasdaq, Rmkt.
rmkt <- DATA$`NASDAQ Rtrn`

# UST 10y Bonds, Rf.
rf <- DATA$`UST10 mv`
```

Then the variables that will actually be used in the model are constructed, i.e. the asset's own risk premium and the market risk premium:

```
# Assets Risk Premium, Ri-Rf.
ri_rf <- ri - rf

# Market Risk Premium, Rmkt-Rf.
rmkt_rf <- rmkt - rf
```

3. OLS Estimation.

The following model is estimated using the Ordinary Least Squares (OLS) method in matrix form:

$$Y = X\beta + e$$

Where Y is a matrix of size $T \times 1$ that contains the values of the dependent variable, X is a design matrix of size $T \times K$ that contains the independent variables, β is a matrix of size $K \times 1$ that contains the parameters that relate the explanatory variables to the explained variable, and e is a matrix of size $T \times 1$ that contains the unobservable disturbances.

3.1. Design matrices.

More specifically, the Y and X matrices, from which the information and data are available, are constructed as follows:

$$Y = \begin{bmatrix} r_1^i - r_1^f \\ r_2^i - r_2^f \\ \vdots \\ r_T^i - r_T^f \end{bmatrix}_{T \times 1} \quad X = \begin{bmatrix} 1 & r_1^{mkt} - r_1^f \\ 1 & r_2^{mkt} - r_2^f \\ \vdots & \vdots \\ 1 & r_T^{mkt} - r_T^f \end{bmatrix}_{T \times 2}$$

```
# Y Matrix (Dependent variable).
Y <- as.matrix(ri_rf)

# X Matrix (Independent variable).
X <- cbind(1, rmkt_rf)

# Number of observations and variables.
N <- dim(X)[1]
K <- dim(X)[2]
```

3.2. Ordinary Least Squares.

The Ordinary Least Squares estimator for the coefficients (betas) of the model, in matrix form, is as follows:

$$\hat{\beta} = (X'X)^{-1} X'Y$$

```
# OLS Estimation Function.
Beta_OLS.f <- function(X,Y){

  # Formula.
  beta_ht <- solve(t(X)%*%X)%*%t(X)%*%Y

  # Result.
  return(beta_ht)
}

# Estimated Betas.
B_ht <- Beta_OLS.f(X,Y)
B_ht
```

```
##           [,1]
##      0.009957433
## rmkt_rf 1.268456732
```

Interpretations of the estimated betas are as follows:

1. Amazon's natural risk premium, represented by the $\hat{\alpha}$ intercept, is approximately **0.9957%** per month, holding all else constant. This coefficient indicates how much additional return the asset generates no matter what happens to market returns.
2. Since $\hat{\beta}$ is greater than 1, this indicates that Amazon is more volatile than the market, i.e., it reacts in an amplified manner to positive or negative movements in the NASDAQ. In particular, the coefficient value indicates that given a 1% increase (0.01 in the magnitude of the data) in the market risk premium, Amazon's risk premium increases by approximately **1.2685%** (0.012685), holding all else constant.

3.3. Dependent variable estimation.

To find the estimated dependent variable, the matrix X of independent variables is multiplied by the vector $\hat{\beta}$ of coefficients estimated by OLS.

$$\hat{Y} = X\hat{\beta}$$

```
# Estimated dependent variable.
Y_ht <- X%*%B_ht
head(Y_ht)
```

```
##           [,1]
## [1,] -0.018762139
## [2,]  0.097704835
## [3,] -0.008072269
## [4,]  0.018318802
## [5,]  0.040741152
## [6,] -0.013302341
```

3.4. Errors estimation.

To estimate the errors, which in principle are not observable, the actual values of the dependent variable and the values estimated by the model are subtracted.

$$\hat{e} = Y - \hat{Y}$$

```
# Estimated errors.
e_ht <- Y - Y_ht
head(e_ht)
```

```
##           [,1]
## [1,]  0.15980337
## [2,] -0.02715912
## [3,] -0.01456111
## [4,]  0.11326267
## [5,] -0.02495356
## [6,]  0.02254920
```

3.5. Estimated variance of errors.

The unbiased estimator for the variance of the OLS method errors is calculated as follows:

$$\hat{\sigma}^2 = \frac{\hat{e}' \cdot \hat{e}}{T - K}$$

```
# Estimated Variance Function.
sigma2_ht.f <- function(errors, T_obs, K_var){

  # Formula.
  sigma2_ht <- as.numeric((t(errors)%*%errors)/(T_obs-K_var))

  # Result.
  return(sigma2_ht)
}
```

```
# Estimated Variance.
sigma2_ht <- sigma2_ht.f(e_ht, N, K)
sigma2_ht
```

```
## [1] 0.003867785
```

```
# Estimated Standard Deviation.
sigma_ht <- sqrt(sigma2_ht)
sigma_ht
```

```
## [1] 0.06219152
```

3.6. Variance-Covariance matrix of betas.

The variance-covariance matrix of the betas estimated by OLS is calculated as follows:

$$\sum_{\hat{\beta}} = \text{VarCov}(\hat{\beta}) = \hat{\sigma}^2 (X'X)^{-1}$$

For the number of regressors in this model, the matrix contains the following information:

$$\sum_{\hat{\beta}} = \text{VarCov}(\hat{\beta}) = \begin{bmatrix} \text{Var}(\hat{\alpha}) & \text{Cov}(\hat{\alpha}, \hat{\beta}) \\ \text{Cov}(\hat{\beta}, \hat{\alpha}) & \text{Var}(\hat{\beta}) \end{bmatrix}_{2 \times 2}$$

```
# Variance-Covariance Matrix.
varcov_beta_ht <- sigma2_ht*solve(t(X)%*%X)
varcov_beta_ht
```

```
##                                rmkt_rf
##          4.119061e-05 -0.0001122052
## rmkt_rf -1.122052e-04  0.0139704374
```

```
# Standard deviation of betas.
sd_beta_ht <- sqrt(diag(varcov_beta_ht))
sd_beta_ht
```

```
##                                rmkt_rf
## 0.006417991 0.118196605
```

3.7. Coefficient of determination.

To measure the goodness of fit of the model, the coefficient of determination R^2 is used, which represents the percentage of variability of the dependent variable that is explained by the variability of the explanatory variables of the model. There are two ways to calculate this statistic:

$$R^2 = \frac{\text{SSR}}{\text{SST}} = \frac{\sum_{t=1}^T (\hat{Y}_t - \bar{Y})^2}{\sum_{t=1}^T (Y_t - \bar{Y})^2}$$

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}} = 1 - \frac{\sum_{t=1}^T (\hat{e}_t)^2}{\sum_{t=1}^T (Y_t - \bar{Y})^2}$$

Where SSR represents the sum of squares explained by regression, SSE represents the sum of squared errors, and SST represents the total sum of squares. These three components are defined exactly as shown in the corresponding formulas.

```
# R2 function.
R2.f <- function(Y, Y_ht){

  # Sum of squares explained.
  ssr <- sum((Y_ht-mean(Y))^2)

  # Total sum of squares.
  sst <- sum((Y-mean(Y))^2)

  # Formula.
  R2 <- as.numeric(ssr/sst)

  # Result.
  return(R2)
}
```

```
# Coefficient of determination R2.
R2 <- R2.f(Y, Y_ht)
R2
```

```
## [1] 0.5506059
```

This result indicates that approximately **55.1%** of the variability of Amazon's risk premium is explained by the variability of the NASDAQ market risk premium.

3.7.1. Adjusted coefficient of determination.

One of the drawbacks of R^2 as a goodness-of-fit measure is that its value tends to increase as new explanatory variables are added to the model. This can lead to the misinterpretation that a model with more variables is always better. In reality, some of these additional variables may lack a consistent economic relationship with the model or may not be statistically significant.

To address this problem, the adjusted coefficient of determination R^2_{adj} is introduced, which penalizes the increase in R^2 as more variables are added to the model. This coefficient is defined as follows:

$$R^2_{adj} = 1 - \frac{T-1}{T-k-1}(1 - R^2)$$

Where R^2 is the normal coefficient of determination, T is the total number of observations, and k is the number of regressors without the intercept.

It should be clarified that in this particular case, since the model has only one regressor other than the intercept, it is not necessary to use the adjusted coefficient of determination. However, to provide a more complete understanding and for educational purposes, the result of the calculation of the R^2_{adj} is presented below.

```
# Adjusted R2 function.
R2adj.f <- function(R2, T_obs, K_var){

  # Formula.
  R2adj <- 1-((1-R2)*((T_obs-1)/(T_obs-(K_var-1)-1)))

  # Result.
  return(R2adj)
}

# Adjusted R2 Coefficient of determination.
R2adj <- R2adj.f(R2, N, K)
R2adj
```

```
## [1] 0.5458251
```

This result indicates that, adjusted under the number of regressors, approximately **54.6%** of the variability of Amazon's risk premium is explained by the variability of the NASDAQ market risk premium.

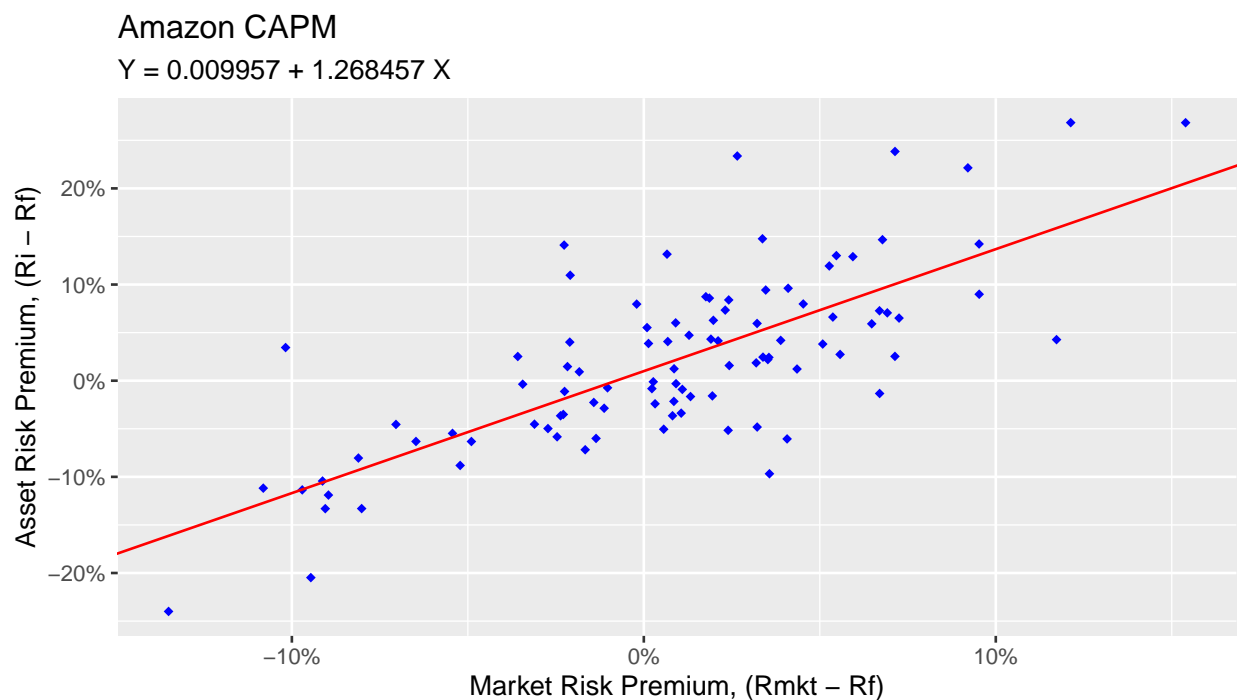
4. Use of the model.

```

# Provisional DataFrame.
df1 <- as.data.frame(cbind(ri_rf, rmkt_rf))

# Graph.
ggplot(df1, aes(x=rmkt_rf, y=ri_rf)) +
  geom_point(color="blue", pch=18) +
  geom_abline(color="red", intercept = B_ht[1], slope = B_ht[2]) +
  labs(x = "Market Risk Premium, (Rmkt - Rf)",
       y = "Asset Risk Premium, (Ri - Rf)",
       title = "Amazon CAPM",
       subtitle = paste("Y =", round(B_ht[1],6), "+", round(B_ht[2],6), "X")) +
  scale_y_continuous(labels = scales::percent) +
  scale_x_continuous(labels = scales::percent)

```



4.1. Forecast.

Once the linear regression coefficients have been obtained, they can be used to predict and evaluate Amazon's future profitability. This process is simple and is achieved by substituting any desired value of the market premium into the functional form of the regression model, or by multiplying a transposed vector of information by the matrix of estimated coefficients.

$$\hat{y}_t = \hat{\alpha} + x_t \hat{\beta} \quad \text{or} \quad \hat{y}_t = x_t' \hat{\beta}$$

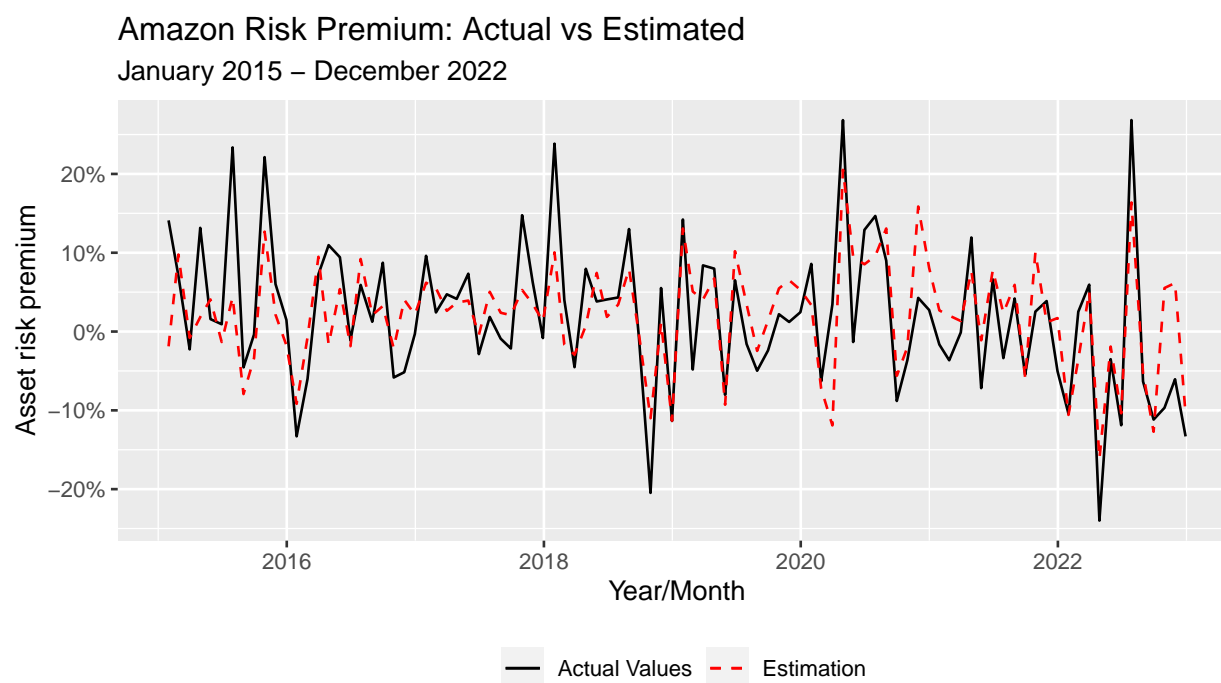
In this way, an estimate of Amazon's future profitability can be obtained using historical data and patterns found in the linear regression model. This approach allows investors to anticipate possible outcomes and make informed decisions about investing in Amazon stock in the future.

4.1.1. In-sample forecast.

From the estimated dependent variables \hat{Y} , they can be compared with the actual realized values, in this case $y_t = r_t^i - r_t^f$.

```
# Provisional Data Frame.
df2 <- data.frame(Date = DATA$Date, Y_ht = Y_ht, ri_rf = ri_rf)

# In-sample forecast graph.
ggplot(df2, aes(x=Date)) +
  geom_line(aes(y=ri_rf, linetype="Actual Values", color="Actual Values")) +
  geom_line(aes(y=Y_ht, linetype="Estimation", color="Estimation")) +
  labs(title = "Amazon Risk Premium: Actual vs Estimated",
       subtitle = "January 2015 - December 2022",
       x = "Year/Month", y = "Asset risk premium",
       linetype = "", color = "") +
  scale_linetype_manual(values = c("Actual Values"="solid", "Estimation"="dashed")) +
  scale_color_manual(values = c("Actual Values"="black", "Estimation"="red")) +
  scale_y_continuous(labels = scales::percent) +
  theme(legend.position = "bottom")
```



When analyzing the series, it can be noted that at several points the estimate resembles the actual realization, at some points the estimate is very close and at others, it is far from the actual realization. However, in some cases, the direction of the estimate is opposite to that of the realization.

To measure the average estimation error in-sample, the **Mean Absolute Error (MAE)** is calculated, which consists of the arithmetic average of the absolute value of the estimated errors \hat{e} of the sample of size n to be evaluated. The average of the residuals is not taken as it is because, due to estimation properties, this average will always be equal to zero. Therefore, the absolute value is used to avoid this problem and to obtain a more accurate measure of the estimation error. The way to calculate the MAE is as follows:

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} = \frac{\sum_{i=1}^n |\hat{e}_i|}{n}$$

```
# Mean absolute error (In-sample).
mae <- sum(abs(e_ht))/N
mae
```

```
## [1] 0.04730879
```

The result obtained indicates that, considering the sample data, **the model is expected to have an average error of approximately 4.73%**, either positive or negative, when estimating Amazon's risk premium.

To evaluate the direction of estimation, the percentage of coincidences will be calculated, i.e., it will be counted those estimations whose positive or negative direction coincides with the direction of the variable's actual realization. In this way, the ability of the model to predict the appropriate direction of the Amazon risk premium can be determined more accurately.

```
# Vector of matches.
dMatch <- (Y > 0) == (Y_ht > 0)

# Percentage of matches (In-sample).
sum(dMatch)/N
```

```
## [1] 0.7604167
```

The result obtained indicates that **76.04% of the estimates coincide in direction with the actual variable**, while the remaining 23.96% are estimates whose direction is opposite to the true one.

4.1.2. Out-of-sample forecast.

To evaluate the out-of-sample model forecast, taking into account that the post-sample comparison window is limited, two types of analysis will be performed. First, pre-sample data will be estimated and compared, specifically from **January 2013 to December 2014**. Second, the most recent post-sample data will be compared, covering only the months of January, February, March, and April 2023. It is important to note that the data sample ends in December 2022.

```
# Import out-of-sample data.
DATA0 <- as.data.frame(read_excel(file1, sheet = 2))

# First values of data.
head(DATA0)
```

```
##      Date      AMZN      AMZN Rtrn  NASDAQ  NASDAQ Rtrn  UST10y  UST10 mv
## 1 2013-01-31 13.2750  0.058317057 3142.13  0.040609196 0.01985 0.001639305
## 2 2013-02-28 13.2135 -0.004632768 3160.19  0.005747712 0.01888 0.001559881
## 3 2013-03-28 13.3245  0.008400499 3267.52  0.033963173 0.01852 0.001530386
## 4 2013-04-30 12.6905 -0.047581523 3328.79  0.018751230 0.01675 0.001385231
## 5 2013-05-31 13.4600  0.060635909 3455.91  0.038188012 0.02164 0.001785691
## 6 2013-06-28 13.8845  0.031537890 3403.25 -0.015237640 0.02478 0.002041911
```

```
# Initial variables.
ri_0 <- DATA0$`AMZN Rtrn`
rmkt_0 <- DATA0$`NASDAQ Rtrn`
```

```

rf_0 <- DATA0$`UST10 mv`

# Model variables.
ri_rf_0 <- ri_0 - rf_0
rmkt_rf_0 <- rmkt_0 - rf_0

# Design matrices.
Y0 <- as.matrix(ri_rf_0)
X0 <- cbind(1, rmkt_rf_0)

# Forecast.
Y0_ht <- X0%*%B_ht

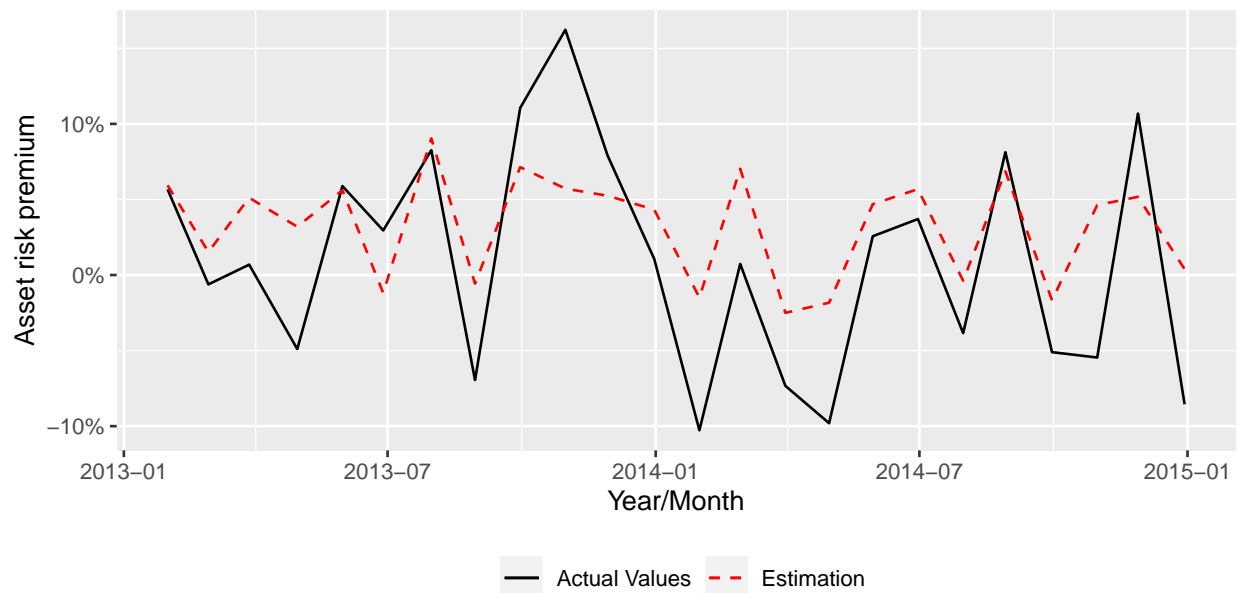
# Provisional Data Frame.
df3 <- data.frame(Date = DATA0$Date, Y0_ht = Y0_ht, ri_rf_0 = ri_rf_0)

# Out-of-sample forecast graph.
ggplot(df3, aes(x=Date)) +
  geom_line(aes(y=ri_rf_0, linetype="Actual Values", color="Actual Values")) +
  geom_line(aes(y=Y0_ht, linetype="Estimation", color="Estimation")) +
  labs(title = "Amazon Risk Premium: Actual vs Estimation",
       subtitle = "January 2013 - December 2014",
       x = "Year/Month", y = "Asset risk premium",
       linetype = "", color = "") +
  scale_linetype_manual(values = c("Actual Values"="solid", "Estimation"="dashed")) +
  scale_color_manual(values = c("Actual Values"="black", "Estimation"="red")) +
  scale_y_continuous(labels = scales::percent) +
  theme(legend.position = "bottom")

```

Amazon Risk Premium: Actual vs Estimation

January 2013 – December 2014



```
# Mean absolute error (Out-of-sample).
e0_ht <- Y0 - Y0_ht
mae0 <- mean(abs(e0_ht))
mae0
```

```
## [1] 0.04647277
```

```
# Vector of matches (Out-of-sample).
dMatch0 <- (Y0 > 0) == (Y0_ht > 0)

# Percentage of matches (Out-of-sample).
sum(dMatch0)/length(Y0)
```

```
## [1] 0.7916667
```

From these results, it can be seen that the forecasting ability of the model does not differ significantly when evaluated with out-of-sample data. Amazon's estimated profitability series, plotted along with the actual realization of profitability during 2013 and 2014, shows similar behavior at most points. Specifically, for this comparison case, the mean absolute estimation error (positive or negative) was found to be **4.65%**, and for the direction estimate, the percentage of matches was approximately **79.17%**.

Next, the forecast quality of the model will be evaluated using the most recent post-sample data. For this purpose, NASDAQ monthly return and risk-free rate data for the period January to April 2023 will be used.

	NASDAQ Monthly Return	UST10Y Monthly Risk-free rate
January	10.68%	0.2894%
February	-1.11%	0.3206%
March	6.69%	0.2866%
April	0.04%	0.2832%

With this information, vectors are constructed individually with the data corresponding to each month and the forecast is made.

```
# Vectors of information.
xjan <- c(1, 0.1068 - 0.002894)
xfeb <- c(1, -0.0111 - 0.003206)
xmar <- c(1, 0.0669 - 0.002866)
xapr <- c(1, 0.0004 - 0.002832)

# Forecast.
frcstM <- data.frame(
  Month = c("January", "February", "March", "April"),
  Estimate = round(c(xjan%*%B_ht, xfeb%*%B_ht, xmar%*%B_ht, xapr%*%B_ht), 6)
)
frcstM
```

```
##      Month Estimate
## 1 January  0.141758
## 2 February -0.008189
## 3 March   0.091182
## 4 April   0.006873
```

	Actual Amazon risk premium	Estimated Amazon risk premium	Estimation error
January	22.48%	14.18%	8.3%
February	-8.95%	-0.82%	-8.13%
March	9.33%	9.12%	0.2%
April	2.09%	0.69%	1.40%

The results indicate that for these four months, the mean absolute error of estimation is **4.51%** and that in all forecasts made, the model was correct in the direction of the percentage change or profitability.

5. Conclusions.

The objective of this document was to build a CAPM model to value and estimate the monthly return of Amazon (AMZN), using the monthly return of the NASDAQ market and the risk-free rate measured as the market rate converted to monthly of 10-year U.S. Treasury bonds. The exercise was carried out for pedagogical purposes to review step by step the estimation process and its application in terms of programming.

Regarding the estimated coefficients of the model, it was found that Amazon generates on average a 1% risk premium, holding all else constant, and that its sensitivity to market movements is amplified, 1.27 times the changes in the NASDAQ market premium.

As for the quality and usefulness of the model in terms of forecasting, it was evaluated with in-sample and out-of-sample data, and it is concluded that the estimation error is between 4.5% to 5%, and that the model accurately predicts the direction of Amazon's percentage change, with a match rate of 75% to 80%. The model is considered to be very useful for making decisions to buy or sell Amazon shares and the predicted value can be used as a take profit or target value. However, it is important to note that when the forecast is around 0%, the variance of residuals and the average estimation error indicates that the asset can go in either a positive or negative direction.

It is noted that this model is constructed in a simple manner and is only based on past price information, which implies that it may not fully capture other aspects that affect Amazon's profitability. Furthermore, in terms of model utilization, it is noted that the NASDAQ monthly risk premium and risk-free rate, which are necessary for forecasting, are not known until the end of the month, just when Amazon's monthly return value is also known. Therefore, as a complement to this model, it would be interesting to build another model that allows forecasting at least one month ahead of the NASDAQ prices and the risk-free rate. In this way, the present CAPM could be fully utilized and the profitability of Amazon's stock could be predicted in a timely manner in order to make investment decisions that generate value.

References.

- Berk, J. & DeMarzo, P. (2020). Optimal Portfolio Choice and the Capital Asset Pricing Model. *Corporate Finance* (pp. 393-443). Pearson Education.
- Judge G., Hill R., Griffiths W., Lütkepohl H. & Lee T. (1988). Introduction to the Theory and Practice of Econometrics. John Wiley & Sons.