

ACTL1101 Assignment Part B

Xing Chen

2024 T2

CAPM Analysis

Introduction

In this assignment, you will explore the foundational concepts of the Capital Asset Pricing Model (CAPM) using historical data for AMD and the S&P 500 index. This exercise is designed to provide a hands-on approach to understanding how these models are used in financial analysis to assess investment risks and returns.

Background

The CAPM provides a framework to understand the relationship between systematic risk and expected return, especially for stocks. This model is critical for determining the theoretically appropriate required rate of return of an asset, assisting in decisions about adding assets to a diversified portfolio.

Objectives

1. **Load and Prepare Data:** Import and prepare historical price data for AMD and the S&P 500 to ensure it is ready for detailed analysis.
2. **CAPM Implementation:** Focus will be placed on applying the CAPM to examine the relationship between AMD's stock performance and the overall market as represented by the S&P 500.
3. **Beta Estimation and Analysis:** Calculate the beta of AMD, which measures its volatility relative to the market, providing insights into its systematic risk.
4. **Results Interpretation:** Analyze the outcomes of the CAPM application, discussing the implications of AMD's beta in terms of investment risk and potential returns.

Instructions

Step 1: Data Loading

- We are using the `quantmod` package to directly load financial data from Yahoo Finance without the need to manually download and read from a CSV file.
- `quantmod` stands for “Quantitative Financial Modelling Framework”. It was developed to aid the quantitative trader in the development, testing, and deployment of statistically based trading models.
- Make sure to install the `quantmod` package by running `install.packages("quantmod")` in the R console before proceeding.

```

# Set start and end dates
start_date <- as.Date("2019-05-20")
end_date <- as.Date("2024-05-20")

# Load data for AMD, S&P 500, and the 1-month T-Bill (DTB4WK)
amd_data <- getSymbols("AMD", src = "yahoo", from = start_date, to = end_date, auto.assign = FALSE)
gspc_data <- getSymbols("^GSPC", src = "yahoo", from = start_date, to = end_date, auto.assign = FALSE)
rf_data <- getSymbols("DTB4WK", src = "FRED", from = start_date, to = end_date, auto.assign = FALSE)

# Convert Adjusted Closing Prices and DTB4WK to data frames
amd_df <- data.frame(Date = index(amd_data), AMD = as.numeric(Cl(amd_data)))
gspc_df <- data.frame(Date = index(gspc_data), GSPC = as.numeric(Cl(gspc_data)))
rf_df <- data.frame(Date = index(rf_data), RF = as.numeric(rf_data[,1])) # Accessing the first column

# Merge the AMD, GSPC, and RF data frames on the Date column
df <- merge(amd_df, gspc_df, by = "Date")
df <- merge(df, rf_df, by = "Date")

```

```
colSums(is.na(df))
```

Data Processing

```
## Date  AMD GSPC  RF
##    0    0    0    9
```

```

# Fill N/A RF data
df <- df %>%
  fill(RF, .direction = "down")

```

Step 2: CAPM Analysis

The Capital Asset Pricing Model (CAPM) is a financial model that describes the relationship between systematic risk and expected return for assets, particularly stocks. It is widely used to determine a theoretically appropriate required rate of return of an asset, to make decisions about adding assets to a well-diversified portfolio.

The CAPM Formula The formula for CAPM is given by:

$$E(R_i) = R_f + \beta_i(E(R_m) - R_f)$$

Where:

- $E(R_i)$ is the expected return on the capital asset,
- R_f is the risk-free rate,
- β_i is the beta of the security, which represents the systematic risk of the security,
- $E(R_m)$ is the expected return of the market.

CAPM Model Daily Estimation

- **Calculate Returns:** First, we calculate the daily returns for AMD and the S&P 500 from their adjusted closing prices. This should be done by dividing the difference in prices between two consecutive days by the price at the beginning of the period.

$$\text{Daily Return} = \frac{\text{Today's Price} - \text{Previous Trading Day's Price}}{\text{Previous Trading Day's Price}}$$

```
df <- df %>%
  mutate(
    AMD_Daily_Return = (AMD - lag(AMD)) / lag(AMD),
    GSPC_Daily_Return = (GSPC - lag(GSPC)) / lag(GSPC)
  )
df$AMD_Daily_Return[1] <- 0
df$GSPC_Daily_Return[1] <- 0
```

- **Calculate Risk-Free Rate:** Calculate the daily risk-free rate by conversion of annual risk-free Rate. This conversion accounts for the compounding effect over the days of the year and is calculated using the formula:

$$\text{Daily Risk-Free Rate} = \left(1 + \frac{\text{Annual Rate}}{100}\right)^{\frac{1}{360}} - 1$$

```
df <- df %>%
  mutate(
    Daily_RF_Rate = ((1 + RF / 100) ^ (1 / 360)) - 1
  )
```

- **Calculate Excess Returns:** Compute the excess returns for AMD and the S&P 500 by subtracting the daily risk-free rate from their respective returns.

```
df <- df %>%
  mutate(
    AMD_excess_returns = AMD_Daily_Return - Daily_RF_Rate,
    GSPC_excess_returns = GSPC_Daily_Return - Daily_RF_Rate
  )
df$AMD_excess_returns[1] <- 0
df$GSPC_excess_returns[1] <- 0
```

- **Perform Regression Analysis:** Using linear regression, we estimate the beta (β) of AMD relative to the S&P 500. Here, the dependent variable is the excess return of AMD, and the independent variable is the excess return of the S&P 500. Beta measures the sensitivity of the stock's returns to fluctuations in the market.

```
df_regression <- lm(AMD_excess_returns ~ GSPC_excess_returns, data = df)
summary_df <- summary(df_regression)
summary_df
```

```
##
## Call:
## lm(formula = AMD_excess_returns ~ GSPC_excess_returns, data = df)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.095780 -0.014728 -0.001134  0.012270  0.173633
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.0011032  0.0007237   1.524   0.128
## GSPC_excess_returns 1.5700013  0.0540439  29.050 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02566 on 1257 degrees of freedom
## Multiple R-squared:  0.4017, Adjusted R-squared:  0.4012
## F-statistic: 843.9 on 1 and 1257 DF,  p-value: < 2.2e-16
```

Interpretation What is your β ? Is AMD more volatile or less volatile than the market?

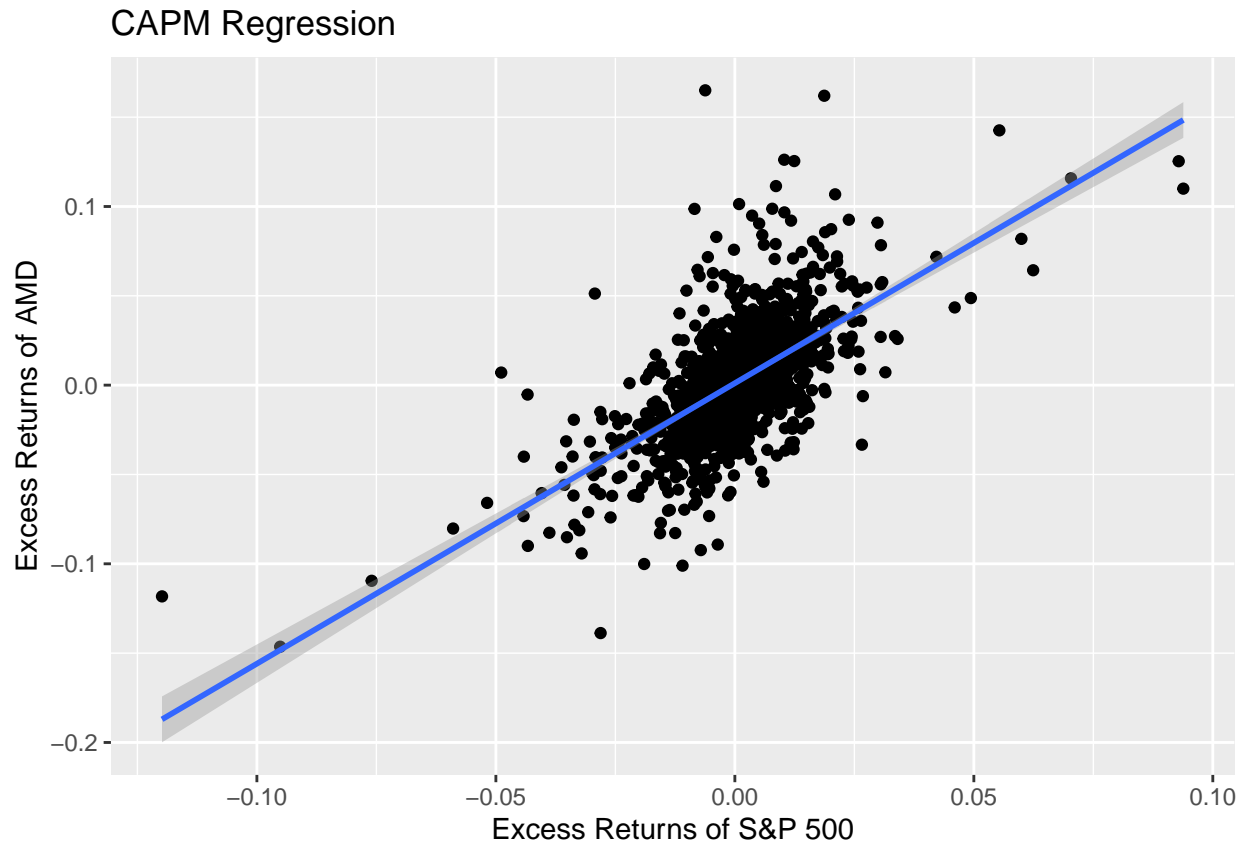
My β value is 1.5700013. We can infer from this value of β that AMD's stock is more volatile than the market. If market's return shifts 1% outwards (inwards), AMD's returns would increase (decrease) by 1.5700% according to the regression summary. We know that if the return of a given stock return is more susceptible to change, it is implied that there would naturally be a higher risk level despite its potential to generate higher rates of return in the market. If the market is performing well, this would generate higher returns for the stock, conversely, if the market were to perform poorly then said stock would also incur a greater loss as a result of the added risk involved.

Answer:

Plotting the CAPM Line Plot the scatter plot of AMD vs. S&P 500 excess returns and add the CAPM regression line.

```
attach(df) #Omitting the need for referring to the 'df' file
#Plotting the graph and labell
ggplot(df,aes(x=GSPC_excess_returns,y=AMD_excess_returns))+
  geom_point()+ #Plotting the points of data to make a scatter plot
  labs(x='Excess Returns of S&P 500', y='Excess Returns of AMD',title='CAPM Regression')+
  #Changing the label to excess returns
  geom_smooth(method='lm', se=TRUE) #Making the CAPM regression line
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



Step 3: Predictions Interval

Suppose the current risk-free rate is 5.0%, and the annual expected return for the S&P 500 is 13.3%. Determine a 90% prediction interval for AMD's annual expected return.

Hint: Calculate the daily standard error of the forecast (s_f), and assume that the annual standard error for prediction is $s_f \times \sqrt{252}$. Use the simple return average method to convert daily stock returns to annual returns if needed.

Note:

$$E(R_i) = R_f + \beta_i(E(R_m) - R_f)$$

Where:

- $E(R_i)$ is the expected return on the capital asset,
- R_f is the risk-free rate,
- β_i is the beta of the security, which represents the systematic risk of the security,
- $E(R_m)$ is the expected return of the market.

CAPM Model Daily Estimation Answer:

```
Current_RFR<-0.05
Annual_Market_ER<-0.133 #Annual Market expected return
alpha<-0.1
```

```

t<-qt(1-alpha/2, 1257, lower.tail=TRUE)
#Finding critical t value with both tails on a 5% level of significance on both sides.
n<- length(df$GSPC_Daily_Return) #Number of observations
MSE<-mean(summary_df$residuals^2)
std_error<-sqrt(MSE)
GSPC_mean<- mean(df$GSPC_Daily_Return)
slope<- coef(df_regression)["GSPC_excess_returns"] #Stating slope found from summary
Predicted_Ret_AMD<-Current_RFR+slope*(Annual_Market_ER-Current_RFR)
#Finding the forecasted(expected) return on AMD stock
var_ret<-sum(((df$GSPC_Daily_Return)-GSPC_mean)^2)
#Variance of independent variable, of which is S&P500
Daily_SEF<-std_error*sqrt((1+1/n+(Annual_Market_ER/252-GSPC_mean)^2/var_ret))

#Finding the daily standard error of forecast
Annual_SEF<-sqrt(252)*Daily_SEF
#Finding the annual standard error of the forecast
Up_bound_Pred_Int<-Predicted_Ret_AMD+(t*Annual_SEF)
Low_bound_Pred_Int<-Predicted_Ret_AMD-(t*Annual_SEF)
cat("The estimated annual return of our AMD stock based off our model is"

,paste(round(Predicted_Ret_AMD,2)), "\n")

```

The estimated annual return of our AMD stock based off our model is 0.18

```

#Stating our expected AMD return
cat("This outcome has a 90% probability of lying within the prediction interval of [",
paste(round(Low_bound_Pred_Int*100, 2)), "% ,",
paste(round(Up_bound_Pred_Int*100, 2)), "%]\n")

```

This outcome has a 90% probability of lying within the prediction interval of [-48.99 % , 85.05 %]

```

#Stating our 90% prediction interval

```