ACTL1101 Assignment Part B

Michael Liu

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.

```r
# 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)
```

```r
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")
```

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

Data Processing

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

```r
# 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.

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

```r
getSymbols("AMD", from = "2020-01-01", to = "2024-01-01", src = "yahoo")
```

```
## [1] "AMD"
```

```r
getSymbols("^GSPC", from = "2020-01-01", to = "2024-01-01", src = "yahoo")
```

```
## [1] "GSPC"
```

```r
amd_adj_close <- Cl(AMD)
spy_adj_close <- Cl(GSPC)

amd_returns <- dailyReturn(amd_adj_close)
spy_returns <- dailyReturn(spy_adj_close)
```

- **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:

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

```r
daily_rf_rate <- (1+5/100)^(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.

```r
amd_excess_returns <- amd_returns - daily_rf_rate
spy_excess_returns <- spy_returns - daily_rf_rate
```

- **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.

```r
capm_model <- lm(amd_excess_returns ~ spy_excess_returns)
summary(capm_model)
```

```
##
## Call:
## lm(formula = amd_excess_returns ~ spy_excess_returns)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.097245 -0.015482 -0.001227  0.012754  0.173456
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.0009955  0.0008083   1.232    0.218
## spy_excess_returns 1.5129891  0.0557942  27.117   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02563 on 1004 degrees of freedom
## Multiple R-squared:  0.4228, Adjusted R-squared:  0.4222
## F-statistic: 735.3 on 1 and 1004 DF,  p-value: < 2.2e-16
```

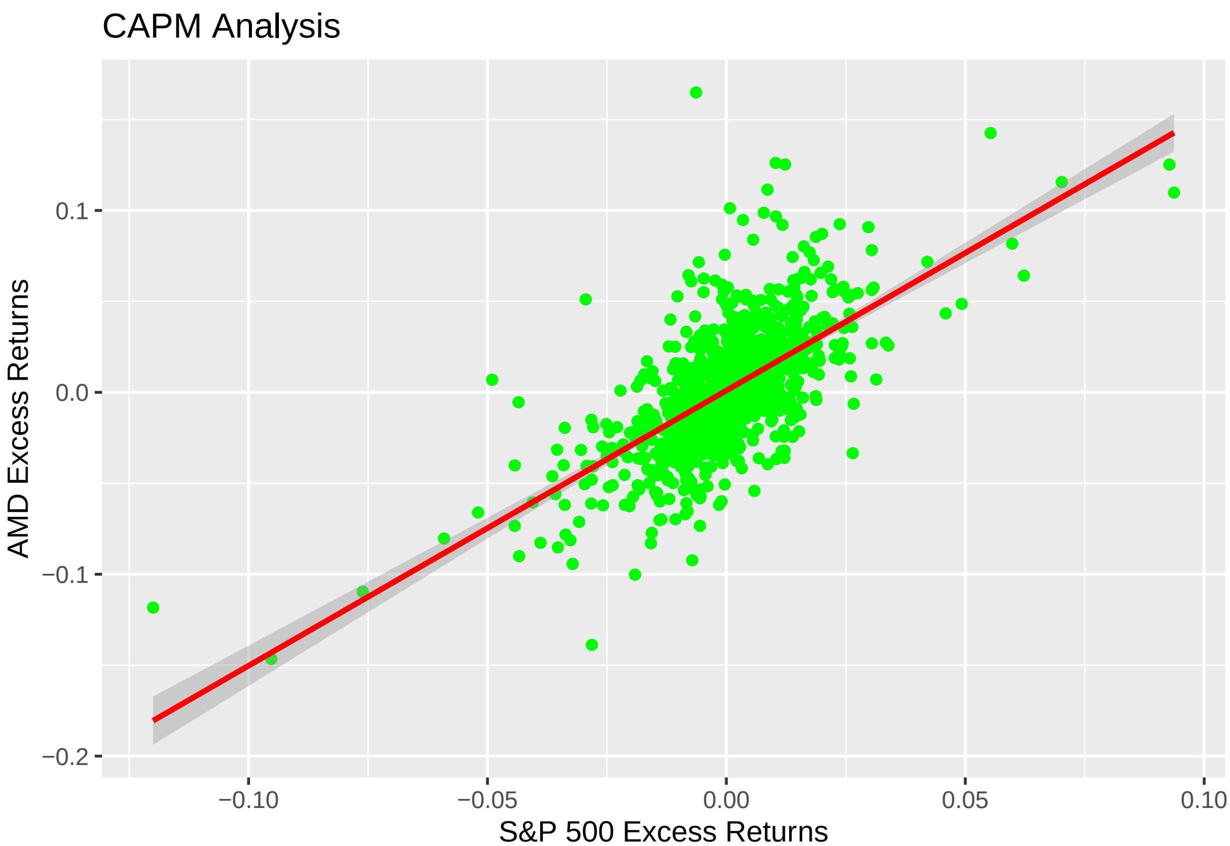
Interpretation What is your β? Is AMD more volatile or less volatile than the market?

Answer: The beta estimate that my code produced was 1.5129891 which suggests that AMD is significantly more volatile than the market. This figure implies that for every 1% the S&P500 moves that AMD's excess returns change by approximately, 1.513%. Hence, demonstrating that AMD's stock price tends to have larger fluctuations compared to the market.

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

```r
ggplot() +
  geom_point(aes(x = spy_excess_returns, y = amd_excess_returns), colour = "green") +
  geom_smooth(aes(x = spy_excess_returns, y = amd_excess_returns), method = "lm", color = "red")+
  labs(x = "S&P 500 Excess Returns", y = "AMD Excess Returns", title = "CAPM Analysis")
```

```
## Don't know how to automatically pick scale for object of type <xts/zoo>.
## Defaulting to continuous.
## Don't know how to automatically pick scale for object of type <xts/zoo>.
## Defaulting to continuous.
## `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.

Answer: Based on the CAPM model, the 90% prediction interval for AMD's annual expected return is devised. This interval provides a range which AMD's annual return is expected to lie with 90% confidence. In this case the range from [-49.36352 %, 84.47914 %] indicates that AMD could offer significant returns but also carries risks of significant losses. This also means that 90% of the outcomes will fall within that range and there is a 5% chance it could be above a -49.4% loss and a 5% chance it could supercede an 84.5% gain.

```r
s_f <- summary(capm_model)$sigma
annual_s_f <- s_f*sqrt(252)
annual_market_return <-0.133
```

```r
annual_amd_expected_return <- (5/100)+coef(capm_model)[2]*(annual_market_return-(5/100))

z_score <- qnorm(0.95)
lower_bound <- annual_amd_expected_return - z_score*annual_s_f
upper_bound <- annual_amd_expected_return + z_score*annual_s_f
cat("90% prediction interval for AMD's annual expected return: [", lower_bound * 100, "%, ", upper_boun
```

```
## 90% prediction interval for AMD's annual expected return: [ -49.36352 %,  84.47914 %]
```