ACTL1101 Assignment Part B

Jacob Nath

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 fir
st column of rf_data

# 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")
```

Data Processing

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

```
## 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$AMD_daily_return <- NA
df$GSPC_daily_return <- NA

for (i in 2:nrow(df)) {
  df$AMD_daily_return[i] <- (df$AMD[i] - df$AMD[i-1])/df$AMD[i-1]
  df$GSPC_daily_return[i] <- (df$GSPC[i] - df$GSPC[i-1])/df$GSPC[i-1]

}
```

- **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$rf_daily <- NA

for (i in 1:nrow(df)) {
  df$rf_daily[i] <- (1 + df$RF[i]/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$AMD_excess_returns <- NA
df$GSPC_excess_returns <- NA

for (i in 2:nrow(df)) {
  df$AMD_excess_returns[i] <- df$AMD_daily_return[i] - df$rf_daily[i]
  df$GSPC_excess_returns[i] <- df$GSPC_daily_return[i] - df$rf_daily[i]
}

head(df)
```

	Date <date>	AMD <dbl>	GSPC <dbl>	RF <dbl>	AMD_daily_return <dbl>	GSPC_daily_return <dbl>	rf_daily <dbl>
1	2019-05-20	26.68	2840.23	2.35	NA	NA	6.452465e-05
2	2019-05-21	27.35	2864.36	2.33	0.025112446	0.008495836	6.398177e-05
3	2019-05-22	27.41	2856.27	2.32	0.002193765	-0.002824396	6.371028e-05
4	2019-05-23	26.36	2822.24	2.34	-0.038307160	-0.011914150	6.425322e-05
5	2019-05-24	26.44	2826.06	2.33	0.003034898	0.001353559	6.398177e-05

	Date	AMD	GSPC	RF	AMD_daily_return	GSPC_daily_return	rf_daily
	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
6	2019-05-28	29.05	2802.39	2.31	0.098714019	-0.008375677	6.343877e-05

6 rows | 1-8 of 10 columns

- **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.

```
CAPM_model <- lm(AMD_excess_returns ~ GSPC_excess_returns, data = df)

summary(CAPM_model)
```

```
##
## Call:
## lm(formula = AMD_excess_returns ~ GSPC_excess_returns, data = df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.095781 -0.014735 -0.001152  0.012276  0.173632
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.0011041  0.0007243   1.524    0.128
## GSPC_excess_returns 1.5699987  0.0540654  29.039   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02567 on 1256 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4017, Adjusted R-squared:  0.4012
## F-statistic: 843.3 on 1 and 1256 DF,  p-value: < 2.2e-16
```

Interpretation

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

Answer: The β value attained when running the linear regression model was 1.5699987. The β value represents the gradient of the linearly-fitted line, and in the context of the CAPM model in question, represents the sensitivity of the AMD's stock return as compared to S&P500 (GSPC), which can be considered the returns of the market (a measure of volatility for AMD's stock). Thus we are essentially measuring the systematic risk compared to the market, exhibiting a value of 1.5699987 which suggests AMD's stock is more volatile as we see generally larger movement in the price of the stock of AMD (indicated by β being greater than 1). The reason for this is because the β value tells us how AMD's stock behaves compared to the market, where if the market were to go up by about 1%, then AMD is predicted to go up about 1.5699987% and vice versa if the market stock goes down.

Since the extent to which AMD varies is greater than the market, it is thus defined as more volatile than the market.

Plotting the CAPM Line

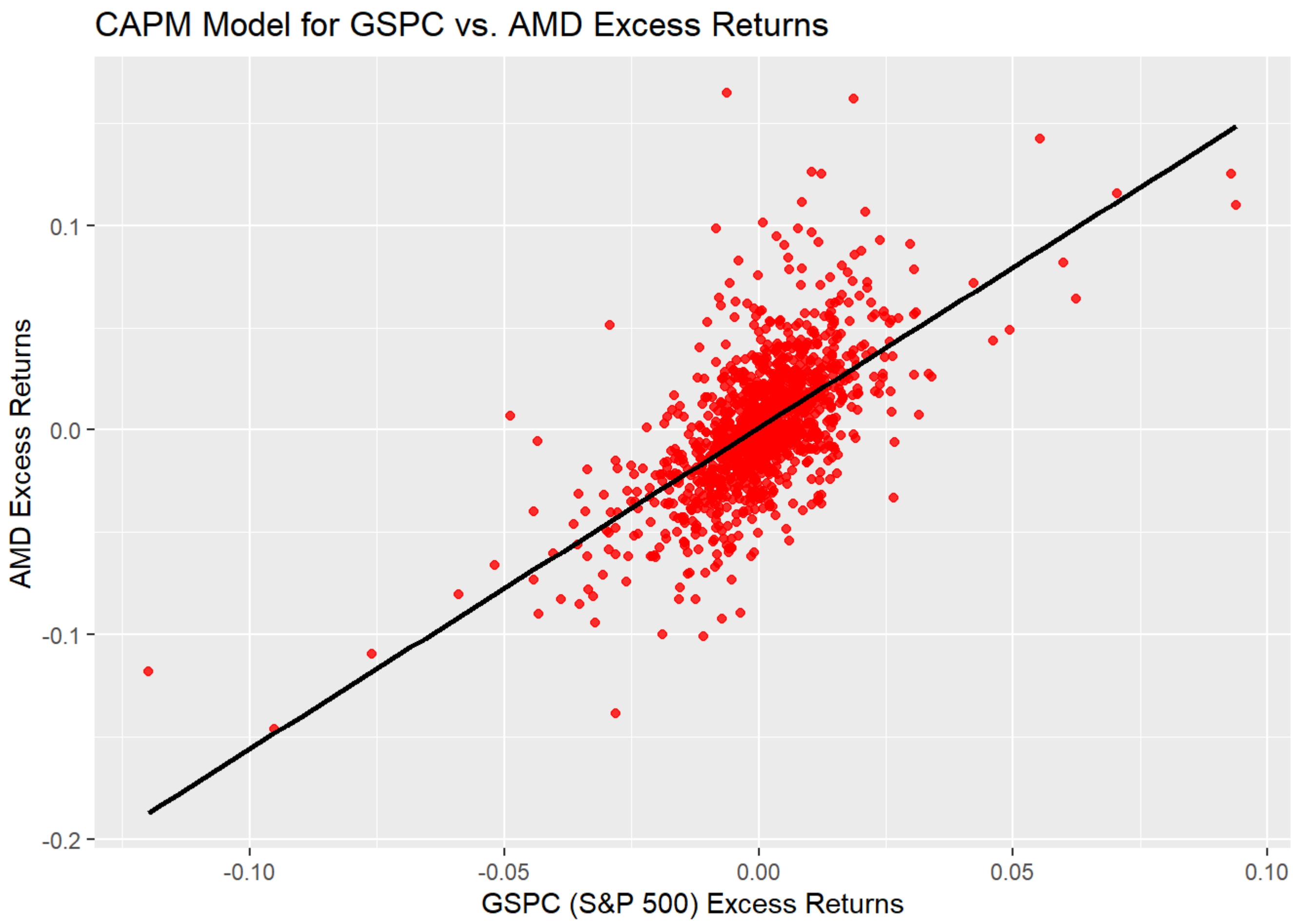
Plot the scatter plot of AMD vs. S&P 500 excess returns and add the CAPM regression line.

```
ggplot(data = df, aes(x = GSPC_excess_returns, y = AMD_excess_returns)) +
  geom_point(color = "red", alpha = 0.8) + # Scatter plot
  geom_smooth(method = "lm", color = "black", se = FALSE) + # Regression line
  labs(title = "CAPM Model for GSPC vs. AMD Excess Returns",
       x = "GSPC (S&P 500) Excess Returns",
       y = "AMD Excess Returns")
```

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

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



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: When calculating the prediction interval of 90%, we must first consider the forecasted value of the dependent variable (AMD expected return), via the forecasted independent variable (S&P 500 returns), given by:

$$\hat{Y}_f = \hat{\beta}_0 + \hat{\beta}_1 X_f.$$

We also calculate the standard error of the forecast, using

$$s_f = s_e\sqrt{1 + \frac{1}{n} + \frac{(X_f - \bar{X})^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}}.$$

Where s_e is the standard error of the estimate (given by 0.02567 in the linear regression summary table), n is the number of observations (given by 1256 from the linear regression summary table), and lastly X_i, X_f, and $\bar{X}$ are the S&P 500 excess returns, the forecasted value of S&P 500 returns, and the mean of S&P 500 market returns, respectively.

After this, given that s_f is calculated, we get the critical t values via the `qt` function, using the appropriate arguments. Lastly, the prediction interval is calculated using

$$\hat{Y}_f \pm t_{\text{critical for } \alpha/2} s_f.$$

The code section below uses these steps to calculate the prediction interval.

```
#List of all the quantities required
rf_daily <- (1 + 0.05)^(1/360) - 1 # Use the conversion of annual risk free rate to the daily
risk free rate
rmarket_daily <- 0.133/252
rmarket_annual <- 0.133
rf_annual <- 0.05
beta <- coef(CAPM_model)[2] # Takes the value of the beta coefficient from linear regression
table
n <- nrow(df)
s_e <- 0.02567 # The residual standard error from linear regression table

# Calculation of quantities to convert the forecasted error from the standard error
X_bar <- mean(df$GSPC_excess_returns, na.rm = TRUE)
Xf <- (rmarket_daily - rf_daily)
X_i <- df$GSPC_excess_returns
SSX <- sum((X_i - X_bar)^2, na.rm = TRUE)

s_f_daily <- s_e * sqrt(1 + 1/n + (Xf - X_bar)^2 / SSX)

s_f_annual <- s_f_daily * sqrt(252)

# The argument of the qt function is 0.95, as it is 1 - α/2 where α = 0.1, and we subtract 2
from nrow(df) due to 2 tails
t_crit <- qt(0.95, df = nrow(df) - 2)

# Use the expected returns of the forecasted stock formula provided for CAPM model
expected_AMD_return_annual <- rf_annual + beta * (rmarket_annual - rf_annual)

# Use upper and lower bound formulae
lower_bound <- expected_AMD_return_annual - t_crit * s_f_annual
upper_bound <- expected_AMD_return_annual + t_crit * s_f_annual

print(t_crit)
```

```
## [1] 1.646067
```

```
print(lower_bound)
```

```
## GSPC_excess_returns
##          -0.4907264
```

```
print(upper_bound)
```

```
## GSPC_excess_returns
##           0.8513462
```

Interpretation of the Results

It can be observed that the prediction interval is [-49.07264%, 85.13462%] using the above steps. This appears to be a relatively wide range, and thus insinuates that there is a large uncertainty for the forecasted annual return of AMD. This is actually somewhat expected, drawing from the response in part 2 where it was mentioned that the AMD stock was more volatile compared to the market. Some other factors which could have lead to this large interval is that the linear regression model did not take into account all the factors that may have affected (and will affect) AMD’s returns.

One other thing that is to be noted that the critical t-value of 1.646067 is greater than the t-value of 1.524 when running the linear regression model, which can be interpreted as that the coefficient is not statistically significant at the 90% confidence interval (which also means we cannot reject the null hypothesis). Hence, the predictor variable (the market returns) is not a very strong predictor of what we returns we will obtain for AMD. Hence, the reliability of our coefficients is to be questioned when using this to forecast AMD returns.