Full name: Nguyen Nhut Vu Truong

Student ID: 413707008

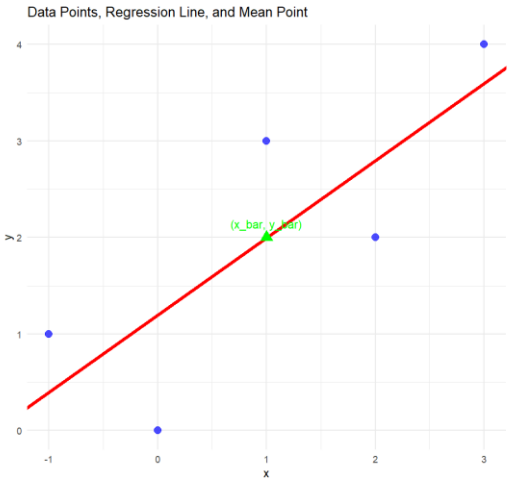
Course: Financial Econometrics

HW0224

Question 1

```
> # Part a
> cat("a. Table Results and Sample Means:\n")
a. Table Results and Sample Means:
> print(answers$a_table)
   x y x_minus_xbar x_minus_xbar_sq y_minus_ybar x_y_diff_prod
1  3 4            2               4            2             4
2  2 2            1               1            0             0
3  1 3            0               0            1             0
4 -1 1           -2               4           -1             2
5  0 0           -1               1           -2             2
> cat("\nSums:\n")

Sums:
> print(answers$a_sums)
  sum_x sum_y sum_x_minus_xbar sum_x_minus_xbar_sq sum_y_minus_ybar sum_x_y_diff_prod
1     5    10                0                  10                0                 8
> cat("Sample means: x_bar =", answers$a_means["x_bar"], ", y_bar =", answers$a_means["y_bar"], "\n\n")
Sample means: x_bar = 1 , y_bar = 2

>
> # Part b
> cat("b. OLS Estimates:\n")
b. OLS Estimates:
> cat("b2 (slope) =", answers$b_coeffs["b2"], "\n")
b2 (slope) = 0.8
> cat("b1 (intercept) =", answers$b_coeffs["b1"], "\n")
b1 (intercept) = 1.2
> cat(answers$b_regression, "\n\n")
y_hat = 1.2 + 0.8 * x


> # Part c
> cat("c. Sums and Identity Verifications:\n")
c. Sums and Identity Verifications:
> cat("Sum x_i^2:", answers$c_sums["sum_x_sq"], "\n")
Sum x_i^2: 15
> cat("Sum y_i^2:", answers$c_sums["sum_y_sq"], "\n")
Sum y_i^2: 30
> cat("Sum x_i * y_i:", answers$c_sums["sum_x_y"], "\n")
Sum x_i * y_i: 18
> cat("Sum (x_i - x_bar)^2 check:", answers$c_identities["sum_x_minus_xbar_sq_check"], " (should match", answers$c_identities["should_match_x"], ")\n")
Sum (x_i - x_bar)^2 check: 10  (should match 10 )
> cat("Sum (x_i - x_bar)(y_i - y_bar) check:", answers$c_identities["sum_x_y_diff_prod_check"], " (should match", answers$c_identities["should_match_x
y"], ")\n\n")
Sum (x_i - x_bar)(y_i - y_bar) check: 8  (should match 8 )

> # Part d
> cat("d. Fitted Values, Residuals, and Statistics:\n")
d. Fitted Values, Residuals, and Statistics:
> print(answers$d_table)
   x y y_hat    e e_sq  x_e
1  3 4   3.6  0.4 0.16  1.2
2  2 2   2.8 -0.8 0.64 -1.6
3  1 3   2.0  1.0 1.00  1.0
4 -1 1   0.4  0.6 0.36 -0.6
5  0 0   1.2 -1.2 1.44  0.0
> cat("\nSums:\n")

Sums:
> print(answers$d_sums)
  sum_x sum_y sum_y_hat         sum_e sum_e_sq        sum_x_e
1     5    10        10 -2.220446e-16      3.6 -1.332268e-15
> cat("Statistics:\n")
Statistics:
> cat("Sample variance of y (s_y^2):", answers$d_stats["s_y_sq"], "\n")
Sample variance of y (s_y^2): 2.5
> cat("Sample variance of x (s_x^2):", answers$d_stats["s_x_sq"], "\n")
Sample variance of x (s_x^2): 2.5
> cat("Sample covariance of x and y (s_xy):", answers$d_stats["s_xy"], "\n")
Sample covariance of x and y (s_xy): 2
> cat("Sample correlation (r_xy):", answers$d_stats["r_xy"], "\n")
Sample correlation (r_xy): 0.8
> cat("Coefficient of variation of x (CV_x, %):", answers$d_stats["CV_x"], "\n")
Coefficient of variation of x (CV_x, %): 158.1139
> cat("Median (50th percentile) of x:", answers$d_stats["median_x"], "\n\n")
Median (50th percentile) of x: 1
```

```
> # Part e
> cat("e. Plot of Data Points and Regression Line:\n")
e. Plot of Data Points and Regression Line:
> print(answers$e_plot)
>
> # Part f
> cat("\nf. Regression Line and Mean Point:\n")

f. Regression Line and Mean Point:
> cat(answers$f_check, "\n\n")
The regression line passes through the mean point ( 1 , 2 ): TRUE
```




```
> # Part g
> check_g <- b1 + b2 * x_bar
> check_g == y_bar
[1] TRUE
>
> # Part h
> check_h <- sums_d$sum_y_hat/N
> check_h == y_bar
[1] TRUE
>
> # Part i
> cat("i. Estimated Variance (sigma^2):\n")
i. Estimated Variance (sigma^2):
> cat("sigma^2 =", answers$i_sigma_sq, "\n\n")
sigma^2 = 1.2

>
> # Part j
> cat("j. Variance and Standard Error of b2:\n")
j. Variance and Standard Error of b2:
> cat("Variance of b2 | x:", answers$j_stats["var_b2"], "\n")
Variance of b2 | x: 0.12
> cat("Standard error of b2:", answers$j_stats["se_b2"], "\n")
Standard error of b2: 0.3464102
```

Question 14

```
> # Print results for part (a)
> print(paste("Mean years of education in the rural area:", round(mean_educ_rural, 3)))
[1] "Mean years of education in the rural area: 13.678"
> print(paste("Elasticity of wages in the rural area:", round(elasticity_rural, 3)))
[1] "Elasticity of wages in the rural area: 1.247"

> # Print result for part (b)
> print(paste("Mean wage in urban area:", round(mean_wage_urban, 3)))
[1] "Mean wage in urban area: 22.893"
> print(paste("Standard error of elasticity in urban area:", round(se_elasticity_urban, 3)))
[1] "Standard error of elasticity in urban area: 0.096"

> # Print predicted wages for part (c)
> print("Predicted Wages for 12 and 16 Years of Education:")
[1] "Predicted Wages for 12 and 16 Years of Education:"
> print(predicted_wages)
  Education_Years Urban_Wage Rural_Wage
1              12      18.76      16.72
2              16      28.60      23.92
```

Question 16

```
> # a. Explain why the econometric model above is a simple regression model like those discussed in this chapter.
```

The CAPM econometric model is a simple regression model because it expresses a dependent variable (excess return of security) as a linear function of a single independent variable (market risk premium), with an intercept, a slope coefficient, and an error term. This structure perfectly matches the standard simple regression framework discussed in the chapter, while providing parameters with meaningful financial interpretations.

b. Estimate the CAPM model for each firm, and comment on their estimated beta values.

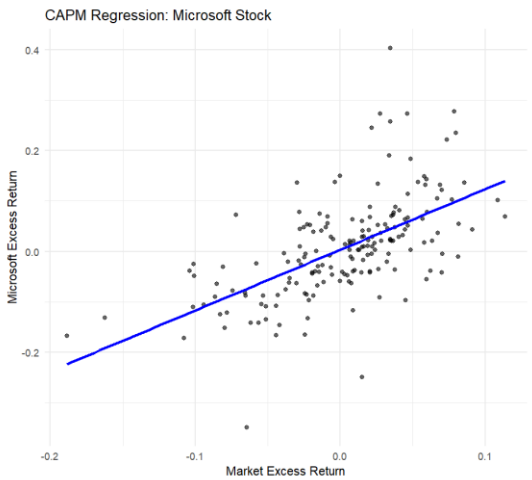
```
> # Display the results
> print(capm_results)
# A tibble: 12 × 6
   stock      term          estimate std.error statistic  p.value
   <chr>      <chr>            <dbl>     <dbl>     <dbl>    <dbl>
 1 GE         (Intercept)  -0.000959   0.00442    -0.217 8.29e- 1
 2 GE         market_excess  1.15      0.0895     12.8   4.47e-27
 3 IBM        (Intercept)   0.00605    0.00483     1.25  2.12e- 1
 4 IBM        market_excess 0.977      0.0978      9.98  6.37e-19
 5 Ford       (Intercept)   0.00378    0.0102      0.370 7.12e- 1
 6 Ford       market_excess 1.66       0.207       8.03  1.27e-13
 7 Microsoft  (Intercept)   0.00325    0.00604     0.538 5.91e- 1
 8 Microsoft  market_excess 1.20       0.122       9.84  1.63e-18
 9 Disney     (Intercept)   0.00105    0.00468     0.224 8.23e- 1
10 Disney     market_excess 1.01       0.0946     10.7   6.50e-21
11 ExxonMobil (Intercept)   0.00528    0.00354     1.49  1.37e- 1
12 ExxonMobil market_excess 0.457      0.0716      6.38  1.48e- 9
```

```
> print(beta_values)
# A tibble: 6 × 7
  stock      term          estimate std.error statistic  p.value type
  <chr>      <chr>            <dbl>     <dbl>     <dbl>    <dbl> <chr>
1 Ford       market_excess    1.66     0.207       8.03 1.27e-13 aggressive
2 Microsoft  market_excess    1.20     0.122       9.84 1.63e-18 aggressive
3 GE         market_excess    1.15     0.0895     12.8  4.47e-27 aggressive
4 Disney     market_excess    1.01     0.0946     10.7  6.50e-21 aggressive
5 IBM        market_excess    0.977    0.0978      9.98 6.37e-19 defensive
6 ExxonMobil market_excess    0.457    0.0716      6.38 1.48e- 9 defensive
```

```
> print(most_aggressive)  # Highest beta
# A tibble: 1 × 6
  stock term          estimate std.error statistic  p.value
  <chr> <chr>            <dbl>     <dbl>     <dbl>    <dbl>
1 Ford  market_excess     1.66     0.207      8.03 1.27e-13
> print(most_defensive)  # Lowest beta
# A tibble: 1 × 6
  stock      term          estimate std.error statistic       p.value
  <chr>      <chr>            <dbl>     <dbl>     <dbl>         <dbl>
1 ExxonMobil market_excess    0.457    0.0716      6.38 0.00000000148
```

c. The p-values of all alphas are larger than 0.1, which is insignificant.

```
> # Check if alpha is significantly different from zero (other options: 0.1 and 0.01)
> significant_alphas <- alpha_values %>%
+   filter(p.value < 0.05)
>
> # Display stocks with significant alphas
> print(significant_alphas)
# A tibble: 0 × 5
# ℹ 5 variables: stock <chr>, estimate <dbl>, std.error <dbl>, statistic <dbl>, p.value <dbl>
>
> # Conclusion based on significance
> if (nrow(significant_alphas) == 0) {
+   print("All intercept estimates are not significantly different from zero, supporting CAPM theory.")
+ } else {
+   print("Some intercepts are significantly different from zero, suggesting additional factors influence returns.")
+ }
[1] "All intercept estimates are not significantly different from zero, supporting CAPM theory."
```



d. Beta values remain similar, it indicates that the intercept does not strongly influence the estimates.

```
> # Display beta comparison
> print(beta_comparison)
# A tibble: 6 × 3
  stock      beta_with_intercept beta_without_intercept
  <chr>                    <dbl>                  <dbl>
1 GE                       1.15                   1.15 
2 IBM                      0.977                  0.984
3 Ford                     1.66                   1.67 
4 Microsoft                1.20                   1.21 
5 Disney                   1.01                   1.01 
6 ExxonMobil               0.457                  0.463
```