## PSET 2

### 1)

OLS regression model: $y = x\beta + e$, where $x$ is the matrix of predictors, $\beta$ is the vector of coefficients and $e$ is the error term.

1. Divide all the X data by 2:

The new coefficients will be twice the original coefficients. This is because each unit change in X now corresponds to half the change it previously did. So, in this case, if the original $\beta' = [1\ 2\ 3]$, the new coefficients would be $\beta' = [2\ 4\ 6]$.

2. Multiply all the Y data by 5:

In this case, all the coefficients will be multiplied by 5, because the relationship between the predictors X and the response Y has been scaled up by a factor of 5. So, if the original $\beta' = [1\ 2\ 3]$, the new coefficients would be $\beta' = [5\ 10\ 15]$.

3. Multiply the X and Y data both by 2:

In this situation, the coefficients will remain the same. This is because multiplying both the predictors and the response by the same constant factor will not change the relationship between them, so the regression coefficients will be unchanged. Therefore, if the original $\beta' = [1\ 2\ 3]$, the new coefficients would still be $\beta' = [1\ 2\ 3]$.

### 2)

Based on the information provided, we can conduct hypothesis tests for the three scenarios. However, note that an exact p-value calculation would require computing t-statistics which need standard errors of the coefficient estimates. The variance-covariance matrix of the coefficients is provided, so we can take square root of the diagonal elements to get standard errors.

Firstly, let's calculate standard errors for $\beta_1$, $\beta_2$ and $\beta_3$:

$$SE(\beta_1) = \sqrt{3}$$
$$SE(\beta_2) = \sqrt{4} = 2$$
$$SE(\beta_3) = \sqrt{3}$$

**a.** The test of $\beta_2 = 0$:

The t-statistic would be calculated as

$$t = (\hat{\beta}_2 - 0)/SE(\beta_2) = (3 - 0)/2 = 1.5$$

The t-value is 1.5, corresponding to a p-value of 0.1388

**b**. The test of $\beta_1 + 2\beta_2 = 5$:

$$SE(\beta 1 + 2\beta 2) = \sqrt{1^2 * 3 + 2^2 * 4 + 2 * 1 * 2 * (-2)}$$
$$= 3.3166$$

Then, calculate the t-statistic:

$$t = \frac{\hat{\beta}_1 + 2\hat{\beta}_2 - 5}{SE(\beta_1 + 2\beta_2)}$$
$$= \frac{2 + 2*3 - 5}{3.3166}$$
$$= 0.9045$$

.

The t-value is 0.9045, corresponding to a p-value of 0.3693

c. The test of $\beta_1 - \beta_2 + 2\beta_3 = 5$:

$$SE(\beta 1 + -\beta_2 + 2\beta_3) = \sqrt{1^2 * 3 + 1^2 * 4 + 2^2 * 3 + 2 * 1 * 1 * (-2) + 2 * 1 * 2 * 1 + 2 * 1 * 2 * 0}$$
$$= 5.196$$

$$t = \frac{\beta_1 - \beta_2 + 2\beta_3 - 5}{SE(\beta_1 - \beta_2 + 2\beta_3)}$$
$$= \frac{2 - 3 + 2 * 1 - 5}{5.196}$$
$$= -0.7698$$

The t-value is -0.7698, corresponding to a p-value of 0.4444

3) ***All the code and the output of STATA are shown in Appendix***

**a .** For Microsoft, we estimated a beta of approximately 1.133 and for Tesla, a beta of approximately 1.392. These estimates indicate that both stocks tend to amplify the overall market's movements, with Tesla having a greater sensitivity to the market.

Next, we tested the null hypothesis that these betas are equal to one or a half. For both stocks, we found strong evidence to reject the null hypotheses. This means that, statistically, the betas for both Microsoft and Tesla are significantly different from 1 and 0.5. The fact that the beta is statistically significantly different from one implies that the stock does not move perfectly with the market.

Hence, the betas of Microsoft and Tesla are not equal to one or half, suggesting that both companies have a higher risk compared to the market portfolio, as they tend to overreact to the movements of the market.

**b.** In this analysis, we first tested for the presence of heteroskedasticity in the residuals of our CAPM regressions using the Breusch–Pagan/Cook–Weisberg test. For both Microsoft and Tesla, we found statistically significant evidence of heteroskedasticity. This implies that the variance of the residuals is not constant across different levels of the independent variable, violating one of the key assumptions of ordinary least squares regression.

To correct for heteroskedasticity, we estimated robust standard errors, which are valid regardless of whether the assumption of homoskedasticity holds. The robust standard errors provide more accurate standard error estimates when there is heteroskedasticity.

Next, we re-tested the null hypotheses that the betas are equal to one or a half, but this time using the robust standard errors. We found strong evidence to reject the null hypotheses for both stocks. This indicates that the betas for Microsoft and Tesla are statistically significantly different from 1 and 0.5, even after correcting for heteroskedasticity.

For Microsoft, the robust standard errors lead to a smaller t-value (53.36 versus 67.14) but still highly significant, which indicates that Microsoft's return is still strongly related to the market return. For Tesla, the robust standard errors lead to a smaller t-value (20.47 versus 27.09) but still highly significant, which indicates that Tesla's return is also strongly related to the market return.

In summary, the test results using robust standard errors confirm our previous findings: the betas of Microsoft and Tesla are not equal to one or half, implying that these stocks do not move perfectly with the market, and they both tend to overreact to the movements of the market. The use of robust standard errors provides more reliable standard error estimates and hence more reliable test results in the presence of heteroskedasticity.

c. First, by using GLS (Generalized Least Squares) to correct the standard errors, we see that the beta of Microsoft's excess returns on S&P 500 is approximately 1.133335, and the beta for Tesla's excess returns on S&P 500 is approximately 0.0177807. Both of these betas are significantly different from 1 or 0.5, suggesting that the excess returns of both stocks do not perfectly move one-to-one or one-to-half with the S&P 500, contrary to the CAPM model's assumption.

Next, a Hausman test was performed to compare the coefficients from the OLS and GLS regressions. For Microsoft, the test statistic is close to 0, with a p-value of 0.9997, indicating that we fail to reject the null hypothesis that the coefficients are the same in both models. This implies that the choice between OLS and GLS does not substantially affect the coefficient estimates.

However, for Tesla, the test statistic is 405.43, with a p-value close to 0. This suggests that we can reject the null hypothesis that the coefficients are the same in both models. In this case, the choice between OLS and GLS does make a substantial difference for the coefficient estimates.

Thus, based on the above results, the conclusion would be that for Microsoft, OLS or GLS could be used interchangeably, but for Tesla, the GLS estimation might be more reliable considering the heteroscedasticity correction. Furthermore, neither of the betas for Microsoft or Tesla is close to 1 or 0.5, challenging the CAPM assumptions for these two stocks.

d.
For Microsoft:
In the first regression model, we obtain a beta value of 1.13 with a p-value of 0.000 indicating that it is statistically significant at a 95% confidence level. This implies that for every 1%

change in SPX, Microsoft's returns change by approximately 1.13%.

However, when we interact the year with the SPX returns in the second regression model, the coefficient of ret_spx decreases to 0.888. This reduction suggests that the beta might have been varying over time. Furthermore, we can see from the confidence intervals that several interaction terms are significant at the 95% confidence level.

For Tesla:
In the first regression model, Tesla's beta value is 1.392 with a p-value of 0.000, which is significant at a 95% confidence level. This means for every 1% change in SPX, Tesla's returns change by approximately 1.39%.

Similar to Microsoft, in the second regression model, the beta value decreases to 1.124 when the interaction term is included, suggesting potential variation over time. Some of the interaction terms are statistically significant at the 95% confidence level, reinforcing the hypothesis that beta may have been varying.

Therefore, it can be concluded that there is evidence to suggest that the betas for both Microsoft and Tesla have varied over the given time period. This suggests that the systematic risk associated with these stocks relative to the market has not remained constant and has likely evolved with market conditions, company developments, and overall economic changes.

4)
The uncertainty in returns of private equity or illiquid assets that are not publicly traded/marked to market recalls the concept of heteroskedasticity in econometrics. In the context of econometric modeling, heteroskedasticity refers to the condition where the variability of the error term, or residual, is not constant across all levels of the independent variables. Here, the error term could potentially embody the uncertainty and unique risks associated with investing in these illiquid assets.

Running a CAPM-style regression with private equity or illiquid assets could result in inefficient and biased estimates of beta. This is primarily because CAPM is based on the premise of efficient markets where assets are regularly traded, providing a steady stream of price information. In contrast, private equity investments are marked to market infrequently, potentially leading to "stale" or "smoothed" return data. This could cause beta, a measure of market risk, to be underestimated since the model might not fully capture the unique risks and volatilities of these illiquid assets.

This could certainly make an investor reconsider the role of alternatives in their portfolio for diversification purposes. While the low beta resulting from a standard CAPM regression might suggest low market risk and therefore effective diversification, the actual risk associated with these illiquid assets might be understated due to factors not captured by beta, such as liquidity risk.

In terms of a correction, one potential approach might be to incorporate a measure of liquidity risk into the model. For instance, an illiquidity-adjusted CAPM could be used where a liquidity premium is added to the traditional CAPM to account for the increased risk associated with illiquid investments. Alternatively, we could apply more sophisticated techniques like GARCH models to account for time-varying volatility, a common feature in illiquid markets. However, it's important to bear in mind that these solutions are not without their limitations and could introduce additional complexities to the model.

```
 1 . do "/var/folders/2w/hhdffn2j7vg0q3zxtrj96n6h0000gn/T//SD00622.000000"

 2 . clear

 3 . capture log close

 4 . cd "/Users/yimingzhang/Desktop"
   /Users/yimingzhang/Desktop

 5 . log using "PS2 example.log", replace
  ------------------------------------------------------------------------------
       name:  <unnamed>
        log:  /Users/yimingzhang/Desktop/PS2 example.log
    log type:  text
   opened on:  24 Jul 2023, 23:04:55

 6 .
 7 . * Load the data
 8 . insheet using "pst2_data.csv", comma names
   (4 vars, 5,922 obs)

 9 .
10 . * Convert the date to Stata's date format
11 . gen stata_date = date(date, "YMD")

12 .
13 . * Set the time series variable
14 . tsset stata_date

   Time variable: stata_date, 14613 to 23210, but with gaps
           Delta: 1 unit

15 .
16 . * Check for heteroskedasticity
17 . tsline ret_spx

18 .
19 . * a. Estimate a CAPM style regression for Microsoft
20 . reg ret_msft ret_spx

      Source |       SS           df       MS      Number of obs   =     3,288
-------------+----------------------------------   F(1, 3286)      =   4507.28
       Model |  .515916339         1  .515916339   Prob > F        =    0.0000
    Residual |  .376125349     3,286  .000114463   R-squared       =    0.5784
-------------+----------------------------------   Adj R-squared   =    0.5782
       Total |  .892041687     3,287  .000271385   Root MSE        =     .0107
```

| ret_msft | Coefficient | Std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| ret_spx | **1.132816** | **.0168734** | **67.14** | **0.000** | **1.099732** | **1.165899** |
| _cons | **.0004576** | **.0001868** | **2.45** | **0.014** | **.0000914** | **.0008238** |

```
21 .
22 . * Test that the regression beta is equal to one
23 . test ret_spx == 1

 ( 1)  ret_spx = 1

       F(  1,  3286) =   61.96
            Prob > F =    0.0000

24 . test ret_spx == 0.5

 ( 1)  ret_spx = .5

       F(  1,  3286) = 1406.53
            Prob > F =    0.0000

25 .
26 . * Repeat the above steps for Tesla
27 . reg ret_tsla ret_spx
```

| Source | SS | df | MS | Number of obs | = | 3,288 |
|---|---|---|---|---|---|---|
| | | | | F(1, 3286) | = | 733.63 |
| Model | .779474666 | 1 | .779474666 | Prob > F | = | 0.0000 |
| Residual | 3.49135943 | 3,286 | .001062495 | R-squared | = | 0.1825 |
| | | | | Adj R-squared | = | 0.1823 |
| Total | 4.27083409 | 3,287 | .001299311 | Root MSE | = | .0326 |

| ret_tsla | Coefficient | Std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| ret_spx | **1.392421** | **.0514083** | **27.09** | **0.000** | **1.291626** | **1.493217** |
| _cons | **.001497** | **.0005691** | **2.63** | **0.009** | **.0003812** | **.0026127** |

```
28 . test ret_spx == 1

 ( 1)  ret_spx = 1

       F(  1,  3286) =   58.27
            Prob > F =    0.0000
```

```
29 . test ret_spx == 0.5

    ( 1)  ret_spx = .5

          F(  1,  3286) =  301.35
               Prob > F =    0.0000

30 .
31 . * b. Test for heteroskedasticity for Microsoft
32 . reg ret_msft ret_spx
```

| Source | SS | df | MS | Number of obs | = | 3,288 |
|---|---|---|---|---|---|---|
| | | | | F(1, 3286) | = | 4507.28 |
| Model | .515916339 | 1 | .515916339 | Prob > F | = | 0.0000 |
| Residual | .376125349 | 3,286 | .000114463 | R-squared | = | 0.5784 |
| | | | | Adj R-squared | = | 0.5782 |
| Total | .892041687 | 3,287 | .000271385 | Root MSE | = | .0107 |

| ret_msft | Coefficient | Std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| ret_spx | 1.132816 | .0168734 | 67.14 | 0.000 | 1.099732 | 1.165899 |
| _cons | .0004576 | .0001868 | 2.45 | 0.014 | .0000914 | .0008238 |

```
33 . estat hettest

   Breusch-Pagan/Cook-Weisberg test for heteroskedasticity
   Assumption: Normal error terms
   Variable: Fitted values of ret_msft

   H0: Constant variance

       chi2(1) =   9.79
   Prob > chi2 = 0.0018

34 .
35 . * Correct the standard errors
36 . reg ret_msft ret_spx, robust

   Linear regression                               Number of obs     =      3,288
                                                   F(1, 3286)        =    2847.74
                                                   Prob > F          =     0.0000
                                                   R-squared         =     0.5784
                                                   Root MSE          =      .0107

   ------------------------------------------------------------------------------
                |               Robust
```

| ret_msft | Coefficient | std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| ret_spx | **1.132816** | **.021228** | **53.36** | **0.000** | **1.091194** | **1.174437** |
| _cons | **.0004576** | **.0001862** | **2.46** | **0.014** | **.0000924** | **.0008227** |

```
37 .
38 . * Test that the corrected beta is equal to one
39 . test ret_spx == 1

 ( 1)  ret_spx = 1

       F(  1,  3286) =   39.15
            Prob > F =    0.0000

40 .
41 . * Repeat the above steps for Tesla
42 . reg ret_tsla ret_spx
```

| Source | SS | df | MS | Number of obs | = | 3,288 |
|---|---|---|---|---|---|---|
| Model | **.779474666** | **1** | **.779474666** | F(1, 3286) | = | **733.63** |
| Residual | **3.49135943** | **3,286** | **.001062495** | Prob > F | = | **0.0000** |
| Total | **4.27083409** | **3,287** | **.001299311** | R-squared | = | **0.1825** |
| | | | | Adj R-squared | = | **0.1823** |
| | | | | Root MSE | = | **.0326** |

| ret_tsla | Coefficient | Std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| ret_spx | **1.392421** | **.0514083** | **27.09** | **0.000** | **1.291626** | **1.493217** |
| _cons | **.001497** | **.0005691** | **2.63** | **0.009** | **.0003812** | **.0026127** |

```
43 . estat hettest

   Breusch-Pagan/Cook-Weisberg test for heteroskedasticity
   Assumption: Normal error terms
   Variable: Fitted values of ret_tsla

   H0: Constant variance

       chi2(1) =  19.74
   Prob > chi2 = 0.0000

44 . reg ret_tsla ret_spx, robust

   Linear regression                               Number of obs     =      3,288
                                                   F(1, 3286)        =     418.95
```

```
                                                Prob > F          =     0.0000
                                                R-squared         =     0.1825
                                                Root MSE          =      .0326

------------------------------------------------------------------------------
             |               Robust
     ret_tsla | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
      ret_spx |   1.392421   .0680287    20.47   0.000     1.259039    1.525804
        _cons |    .001497   .0005666     2.64   0.008      .000386     .002608
------------------------------------------------------------------------------

45 . test ret_spx == 1

 ( 1)  ret_spx = 1

       F(  1,  3286) =   33.28
            Prob > F =    0.0000

46 .
47 . * c. Attempt to correct the standard errors through GLS for Microsoft
48 . qui reg ret_msft ret_spx, robust

49 . predict ehat, resid
(2,634 missing values generated)

50 . gen ehat_sq = ehat^2
(2,634 missing values generated)

51 . gen ret_spx_gls = ret_spx / sqrt(ehat_sq)
(2,634 missing values generated)

52 . gen ret_msft_gls = ret_msft / sqrt(ehat_sq)
(2,634 missing values generated)

53 . gen cons_gls = 1 / sqrt(ehat_sq)
(2,634 missing values generated)

54 . reg ret_msft_gls cons_gls ret_spx_gls, noc

      Source |       SS           df       MS      Number of obs   =     3,288
-------------+----------------------------------   F(2, 3286)      >  99999.00
       Model |  8163324.07         2  4081662.03   Prob > F        =    0.0000
    Residual |  3285.63651     3,286  .999889383   R-squared       =    0.9996
-------------+----------------------------------   Adj R-squared   =    0.9996
       Total |   8166609.7     3,288  2483.76208   Root MSE        =    .99994

------------------------------------------------------------------------------
```

| ret_msft_gls | Coefficient | Std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| cons_gls | .0004567 | 9.65e-07 | 473.49 | 0.000 | .0004548 | .0004586 |
| ret_spx_gls | 1.133335 | .0004048 | 2799.66 | 0.000 | 1.132541 | 1.134128 |

```
55 .
56 . * Repeat the above steps for Tesla
57 . qui reg ret_tsla ret_spx, robust

58 . predict ehat2, resid
(2,634 missing values generated)

59 . gen ehat_sq2 = ehat2^2
(2,634 missing values generated)

60 . gen ret_tsla_gls = ret_tsla / sqrt(ehat_sq2)
(2,634 missing values generated)

61 . reg ret_tsla_gls cons_gls ret_spx_gls, noc
```

| Source | SS | df | MS | | |
|---|---|---|---|---|---|
| Model | 1959.75313 | 2 | 979.876565 | Prob > F | 0.8072 |
| Residual | 15034748.3 | 3,286 | 4575.3951 | R-squared | 0.0001 |
| Total | 15036708.1 | 3,288 | 4573.20805 | Root MSE | 67.642 |

| | Number of obs = | 3,288 |
|---|---|---|
| | F(2, 3286) = | 0.21 |
| | Adj R-squared = | -0.0005 |

| ret_tsla_gls | Coefficient | Std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| cons_gls | 3.87e-06 | .0000653 | 0.06 | 0.953 | -.0001241 | .0001318 |
| ret_spx_gls | .0177807 | .0273836 | 0.65 | 0.516 | -.03591 | .0714713 |

```
62 .
63 . * Conduct a Hausman Test for Microsoft
64 . qui reg ret_msft ret_spx

65 . est store ols

66 .
67 . qui reg ret_msft_gls cons_gls ret_spx_gls, noc

68 . est store gls

69 . suest ols gls
```

Simultaneous results for ols, gls  Number of obs = **3,288**

| | Coefficient | Robust std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| **ols_mean** | | | | | | |
| ret_spx | **1.132816** | **.0212248** | **53.37** | **0.000** | **1.091216** | **1.174416** |
| _cons | **.0004576** | **.0001862** | **2.46** | **0.014** | **.0000926** | **.0008225** |
| **ols_lnvar** | | | | | | |
| _cons | **-9.075259** | **.0672333** | **-134.98** | **0.000** | **-9.207034** | **-8.943484** |
| **gls_mean** | | | | | | |
| cons_gls | **.0004567** | **2.65e-07** | **1721.01** | **0.000** | **.0004562** | **.0004572** |
| ret_spx_gls | **1.133335** | **.0002816** | **4024.81** | **0.000** | **1.132783** | **1.133886** |
| **gls_lnvar** | | | | | | |
| _cons | **-.0001106** | **.000686** | **-0.16** | **0.872** | **-.0014552** | **.001234** |

```
70 . *test if constants and slopes are the same in the two specifications
71 . test ([ols_mean]_cons=[gls_mean]cons_gls) ([ols_mean]ret_spx=[gls_mean]ret_spx_g
   > ls)

 ( 1)  [ols_mean]_cons - [gls_mean]cons_gls = 0
 ( 2)  [ols_mean]ret_spx - [gls_mean]ret_spx_gls = 0

           chi2(  2) =    0.00
         Prob > chi2 =    0.9997

72 .
73 . * Repeat the above steps for Tesla
74 . qui reg ret_tsla ret_spx

75 . est store ols

76 . qui reg ret_tsla_gls cons_gls ret_spx_gls, noc

77 . est store gls

78 . suest ols gls
```

Simultaneous results for ols, gls  Number of obs = **3,288**

| | Coefficient | Robust std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| **ols_mean** | | | | | | |
| ret_spx | **1.392421** | **.0680183** | **20.47** | **0.000** | **1.259108** | **1.525735** |
| _cons | **.001497** | **.0005666** | **2.64** | **0.008** | **.0003866** | **.0026074** |
| **ols_lnvar** | | | | | | |
| _cons | **-6.847135** | **.0504303** | **-135.77** | **0.000** | **-6.945977** | **-6.748294** |
| **gls_mean** | | | | | | |
| cons_gls | **3.87e-06** | **3.86e-06** | **1.00** | **0.315** | **-3.69e-06** | **.0000114** |
| ret_spx_gls | **.0177807** | **.0093345** | **1.90** | **0.057** | **-.0005146** | **.036076** |
| **gls_lnvar** | | | | | | |
| _cons | **8.428448** | **.7042744** | **11.97** | **0.000** | **7.048096** | **9.808801** |

```
79 . *test if constants and slopes are the same in the two specifications
80 . test ([ols_mean]_cons=[gls_mean]cons_gls) ([ols_mean]ret_spx=[gls_mean]ret_spx_g
   > ls)

 ( 1)  [ols_mean]_cons - [gls_mean]cons_gls = 0
 ( 2)  [ols_mean]ret_spx - [gls_mean]ret_spx_gls = 0

           chi2(  2) =  405.43
         Prob > chi2 =    0.0000

81 .
82 . * d. Check the stability of the betas through the sample for Microsoft
83 .
84 . gen year = substr(date, 1, 4)

85 . destring year, replace
   year: all characters numeric; replaced as int

86 .
87 . reg ret_msft ret_spx, robust

   Linear regression                               Number of obs     =      3,288
                                                   F(1, 3286)        =    2847.74
                                                   Prob > F          =     0.0000
                                                   R-squared         =     0.5784
                                                   Root MSE          =      .0107
```

| ret_msft | Coefficient | Robust std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| ret_spx | **1.132816** | **.021228** | **53.36** | **0.000** | **1.091194** | **1.174437** |

| | | | | | | |
|---|---|---|---|---|---|---|
| _cons | **.0004576** | **.0001862** | **2.46** | **0.014** | **.0000924** | **.0008227** |

```
88 . xi i.year*ret_spx
   i.year            _Iyear_2000-2023    (naturally coded; _Iyear_2000 omitted)
   i.year*ret_spx    _IyeaXret_#         (coded as above)

89 . drop _Iyear*

90 . reg ret_msft ret_spx _I*, robust
   note: _IyeaXret_2014 omitted because of collinearity.
   note: _IyeaXret_2015 omitted because of collinearity.
   note: _IyeaXret_2016 omitted because of collinearity.
   note: _IyeaXret_2017 omitted because of collinearity.
   note: _IyeaXret_2018 omitted because of collinearity.
   note: _IyeaXret_2019 omitted because of collinearity.
   note: _IyeaXret_2020 omitted because of collinearity.
   note: _IyeaXret_2021 omitted because of collinearity.
   note: _IyeaXret_2022 omitted because of collinearity.
   note: _IyeaXret_2023 omitted because of collinearity.

   Linear regression                               Number of obs     =      3,288
                                                   F(14, 3273)       =     323.94
                                                   Prob > F          =     0.0000
                                                   R-squared         =     0.5929
                                                   Root MSE          =     .01053
```

| ret_msft | Coefficient | Robust std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| ret_spx | **.8883665** | **.0648695** | **13.69** | **0.000** | **.7611776** | **1.015555** |
| _IyeaXret_2001 | **-.0558922** | **.0740158** | **-0.76** | **0.450** | **-.2010141** | **.0892297** |
| _IyeaXret_2002 | **.1221236** | **.1013222** | **1.21** | **0.228** | **-.0765377** | **.3207849** |
| _IyeaXret_2003 | **.0774573** | **.1471983** | **0.53** | **0.599** | **-.2111529** | **.3660674** |
| _IyeaXret_2004 | **.2840773** | **.1325609** | **2.14** | **0.032** | **.0241665** | **.5439881** |
| _IyeaXret_2005 | **.3295029** | **.0959274** | **3.43** | **0.001** | **.141419** | **.5175867** |
| _IyeaXret_2006 | **.2371338** | **.0888875** | **2.67** | **0.008** | **.0628531** | **.4114145** |
| _IyeaXret_2007 | **.5471484** | **.1181707** | **4.63** | **0.000** | **.3154523** | **.7788445** |
| _IyeaXret_2008 | **.5565243** | **.0888458** | **6.26** | **0.000** | **.3823253** | **.7307233** |
| _IyeaXret_2009 | **.2061779** | **.0825341** | **2.50** | **0.013** | **.0443542** | **.3680016** |
| _IyeaXret_2010 | **.393908** | **.1032026** | **3.82** | **0.000** | **.1915597** | **.5962563** |
| _IyeaXret_2011 | **.3192717** | **.0779272** | **4.10** | **0.000** | **.1664806** | **.4720627** |
| _IyeaXret_2012 | **.4392827** | **.0868361** | **5.06** | **0.000** | **.2690241** | **.6095412** |
| _IyeaXret_2013 | **.5919075** | **.4338843** | **1.36** | **0.173** | **-.2588048** | **1.44262** |
| _IyeaXret_2014 | **0** | (omitted) | | | | |
| _IyeaXret_2015 | **0** | (omitted) | | | | |
| _IyeaXret_2016 | **0** | (omitted) | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| _IyeaXret_2017 | **0** | (omitted) | | | | |
| _IyeaXret_2018 | **0** | (omitted) | | | | |
| _IyeaXret_2019 | **0** | (omitted) | | | | |
| _IyeaXret_2020 | **0** | (omitted) | | | | |
| _IyeaXret_2021 | **0** | (omitted) | | | | |
| _IyeaXret_2022 | **0** | (omitted) | | | | |
| _IyeaXret_2023 | **0** | (omitted) | | | | |
| _cons | **.0004532** | **.0001838** | **2.47** | **0.014** | **.0000928** | **.0008136** |

```
91 .
92 . * Repeat the above steps for Tesla
93 . reg ret_tsla ret_spx, robust
```

Linear regression — Number of obs = **3,288**; F(1, 3286) = **418.95**; Prob > F = **0.0000**; R-squared = **0.1825**; Root MSE = **.0326**

| ret_tsla | Coefficient | Robust std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| ret_spx | **1.392421** | **.0680287** | **20.47** | **0.000** | **1.259039** | **1.525804** |
| _cons | **.001497** | **.0005666** | **2.64** | **0.008** | **.000386** | **.002608** |

```
94 . xi i.year*ret_spx
i.year            _Iyear_2000-2023    (naturally coded; _Iyear_2000 omitted)
i.year*ret_spx    _IyeaXret_#         (coded as above)

95 . drop _Iyear*

96 . reg ret_tsla ret_spx _I*, robust
note: _IyeaXret_2014 omitted because of collinearity.
note: _IyeaXret_2015 omitted because of collinearity.
note: _IyeaXret_2016 omitted because of collinearity.
note: _IyeaXret_2017 omitted because of collinearity.
note: _IyeaXret_2018 omitted because of collinearity.
note: _IyeaXret_2019 omitted because of collinearity.
note: _IyeaXret_2020 omitted because of collinearity.
note: _IyeaXret_2021 omitted because of collinearity.
note: _IyeaXret_2022 omitted because of collinearity.
note: _IyeaXret_2023 omitted because of collinearity.

Linear regression                               Number of obs     =      3,288
                                                F(14, 3273)       =      54.96
```

| | |
|---|---|
| Prob > F | **0.0000** |
| R-squared | **0.1950** |
| Root MSE | **.03241** |

| ret_tsla | Coefficient | Robust std. err. | t | P>\|t\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| ret_spx | **1.124074** | **.2672483** | **4.21** | **0.000** | **.6000834** | **1.648065** |
| _IyeaXret_2001 | **.1318168** | **.2950776** | **0.45** | **0.655** | **-.4467386** | **.7103722** |
| _IyeaXret_2002 | **.0982042** | **.3529913** | **0.28** | **0.781** | **-.593902** | **.7903104** |
| _IyeaXret_2003 | **.7076956** | **.4085712** | **1.73** | **0.083** | **-.0933854** | **1.508777** |
| _IyeaXret_2004 | **.0966198** | **.3206588** | **0.30** | **0.763** | **-.5320924** | **.725332** |
| _IyeaXret_2005 | **.1881156** | **.3089784** | **0.61** | **0.543** | **-.4176949** | **.7939261** |
| _IyeaXret_2006 | **-.2784742** | **.3234865** | **-0.86** | **0.389** | **-.9127307** | **.3557822** |
| _IyeaXret_2007 | **.3635275** | **.3362665** | **1.08** | **0.280** | **-.2957867** | **1.022842** |
| _IyeaXret_2008 | **.1637751** | **.3402845** | **0.48** | **0.630** | **-.503417** | **.8309671** |
| _IyeaXret_2009 | **-.0174276** | **.3244096** | **-0.05** | **0.957** | **-.6534939** | **.6186387** |
| _IyeaXret_2010 | **1.184405** | **.4157287** | **2.85** | **0.004** | **.3692902** | **1.99952** |
| _IyeaXret_2011 | **.7936406** | **.3076077** | **2.58** | **0.010** | **.1905175** | **1.396764** |
| _IyeaXret_2012 | **.5121882** | **.3136637** | **1.63** | **0.103** | **-.1028088** | **1.127185** |
| _IyeaXret_2013 | **2.080545** | **1.019302** | **2.04** | **0.041** | **.0820114** | **4.079079** |
| _IyeaXret_2014 | **0** | (omitted) | | | | |
| _IyeaXret_2015 | **0** | (omitted) | | | | |
| _IyeaXret_2016 | **0** | (omitted) | | | | |
| _IyeaXret_2017 | **0** | (omitted) | | | | |
| _IyeaXret_2018 | **0** | (omitted) | | | | |
| _IyeaXret_2019 | **0** | (omitted) | | | | |
| _IyeaXret_2020 | **0** | (omitted) | | | | |
| _IyeaXret_2021 | **0** | (omitted) | | | | |
| _IyeaXret_2022 | **0** | (omitted) | | | | |
| _IyeaXret_2023 | **0** | (omitted) | | | | |
| _cons | **.0014121** | **.0005669** | **2.49** | **0.013** | **.0003005** | **.0025237** |

```
97 .
   end of do-file

98 .
```