MINI PROJECT 2 REPORT

STOCK CORRELATION

submitted by

WONGSATHON MATHONGSA

(63010835)

guided by

Asst. Prof. Dr. Tulya Limpiti

Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang

Contents

Mini Project Report

1.Project Goal

1.The goal of this project is trying to use the historical data of one stock to predict the future price of the second stock.

2. Study and review the knowledge previously learned.

3. Practice coding and using a MATLAB program.

2.The highest value of Pearson's correlation coefficient

Pearson's correlation coefficient is the correlation coefficient is a measure of linear correlation between two sets of data.

Pearson's correlation coefficient Equation For a sample is

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{l=1}^{n}(y_i - \bar{y})^2}} = \frac{1}{n}\sum_{i=L}^{n} z_{x_i} \cdot z_{y_i}$$

From **Table 1** in next page

The highest value of Pearson's correlation coefficient between them.

The first one (X) is **GOOG** from Alphabet Inc., and the second one (Y) is **IYW** from iShares U.S. Technology ETF. And value of Pearson's correlation coefficient of them is 0.9611

```
r =

    0.9611
```

Code to find Pearson's correlation coefficient

```
a = importdata('GOOG.csv');   % GOOG data from June 1, 2021 to Oct 31, 2021.
b = importdata('IYW.csv');    % IYW data from June 1, 2021 to Oct 31, 2021.
x = a.data(:,2);              % the daily high price from data a
y = b.data(:,2);              % the daily high price from data b
zx = zscore(x,1);             % Z-score of x
zy = zscore(y,1);             % Z-score of y
r = mean(zx.*zy)              % r = The Pearson's correlation coefficient between x and y using Z-score to calculate
```

(Use Z-score to find Pearson's correlation coefficient)

Table 1: the value of Pearson's correlation coefficient between 2 stocks using the data from June 1, 2021, to Oct 31, 2021.

1st Stock	2nd Stock	value of Pearson's correlation coefficient
AAPL (Apple Inc.)	TECB (iShares U.S. Tech Breakthrough Multisector ETF)	0.9219
	SPMO (Invesco S&P 500 Momentum ETF)	0.9162
	MID (American Century Mid Cap Growth Impact ETF)	0.9197
	ADBE (Adobe Inc.)	0.9278
INTC (Intel Corporation)	OCUL (Ocular Therapeutix, Inc.)	0.9025
	CMBM (Cambium Networks Corporation)	0.8257
	OMC (Omnicom Group Inc.)	0.8486
ENG (ENglobal Corporation)	UBER (Uber Technologies, Inc.)	0.8045
	SDP (ProShares UltraShort Utilities)	0.8565
	ENSV (Enservco corporation)	0.8461
	CRKN (Crown Electrokinetics Corp.)	0.8645
FB (Meta Platforms, Inc.)	FCOM (Fidelity MSCI Communication Services index ETF)	0.9127
	VOX (Vanguard Communication Services Index Fund ETF Shares)	0.9036
	OTEX (Open Text Corporation)	0.8723
GOOG (Alphabet Inc.)	EFX (Equifax Inc.)	0.9457
	RPD (Rapid7, Inc.)	0.9513
	IETC (iShares Evolved U.S. Technology ETF)	0.9487
	IYW (iShares U.S. Technology ETF)	0.9611
	ILCG (iShares Morningstar Growth ETF)	0.9444

3. The Linear Regression Equation

$$\hat{y} = \hat{b}_0 + \hat{b}_1 x$$

$$\hat{b}_0 = \bar{y} - \hat{b}_1 \bar{x} \qquad ; \bar{y} = mean\ of\ y \quad ; \bar{x}\ mean\ of\ y$$

$$\hat{b}_1 = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2} = \frac{sample\ covariance}{sample\ variance\ of\ x}$$

Code to Find $\hat{b}_0$ & $\hat{b}_1$

```
a = importdata('GOOG.csv');              % GOOG data from June 1, 2021 to Oct 31, 2021.
b = importdata('IYW.csv');               % IYW data from June 1, 2021 to Oct 31, 2021.
x = a.data(:,2);                         % the daily high price from data a
y = b.data(:,2);                         % the daily high price from data b
zx = zscore(x,1);                        % Z-score of x
zy = zscore(y,1);                        % Z-score of y
r  = mean(zx.*zy)                        % r = The Pearson's correlation coefficient between x and y using Z-score to calculate
sx = std(x,1);                           % standard deviation of x
mx = mean(x);                            % mean of x
my = mean(y);                            % mean of y
c = mean((x-mx).*(y-my));                % covariance of x and y
bhat1 = c/sx^2                           % bhat1 = covariance of x and y divided by variance of x
bhat0 = my-bhat1*mx                      % bhat0 = mean[y]- bhat1*mean[x]
```

From Code

```
bhat1 =

    0.0300

bhat0 =

   21.2306
```

$\hat{b}_0 = 21.2306$

$\hat{b}_1 = 0.0300$

Complete The Linear Regression Equation

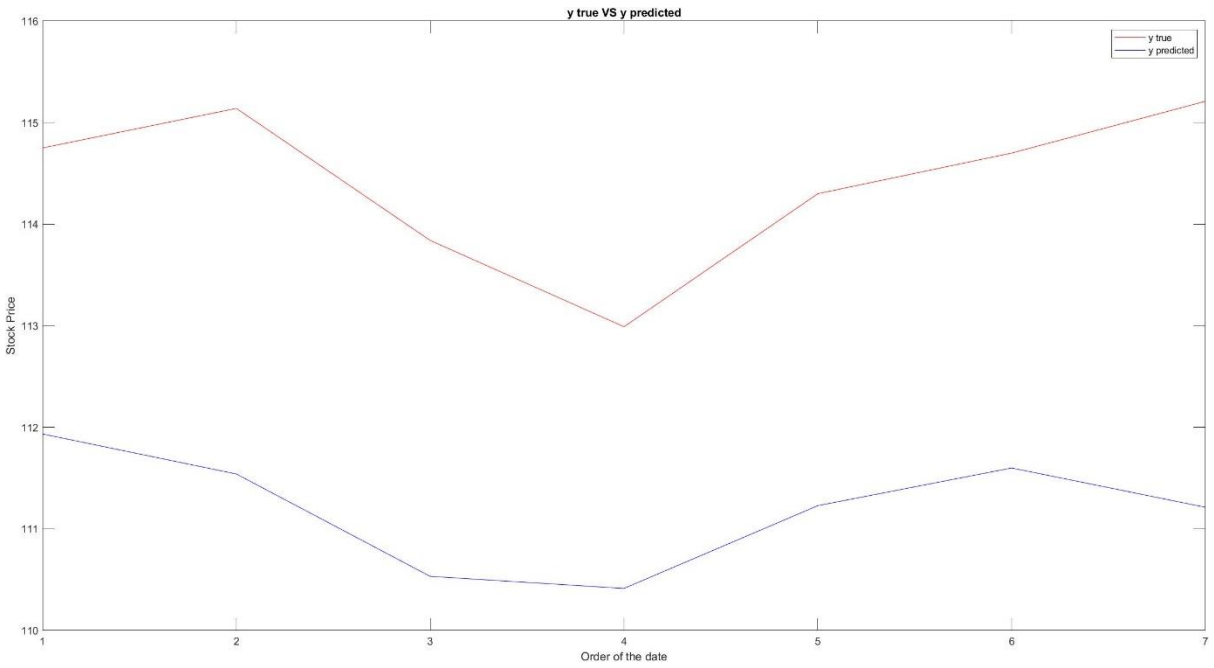
From $\hat{y} = \hat{b}_0 + \hat{b}_1 x$

The Linear Regression Equation is $\hat{y} = 21.2306 + 0.0300x$

4. The actual prices of y and predicted prices of y

Table 2: the value of actual prices of y, and predicted price of y during November 8-16,2021 (7 day)

Date	Order	Actual Prices (y_true)	Predicted Prices (y_predicted)
08/11/2021	1	114.75	111.934
09/11/2021	2	115.14	111.54
10/11/2021	3	113.84	110.532
11/11/2021	4	112.99	110.414
12/11/2021	5	114.3	111.229
15/11/2021	6	114.7	111.599
16/11/2021	7	115.21	111.212



(Graph 0f comparison between y_true and y_predicted)

5. The Values of error

5.1 Mean Squared Error

$$\text{MSE} = \frac{1}{n}\Sigma_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

Code in Matlab

```
MSE  = (sum((y_true - y_predicted).^2))/n2 % MSE = Mean square error
```

```
MSE =

   10.4989
```

5.2 Average Percent Squared Error of predicted stock price.

Use function stocker

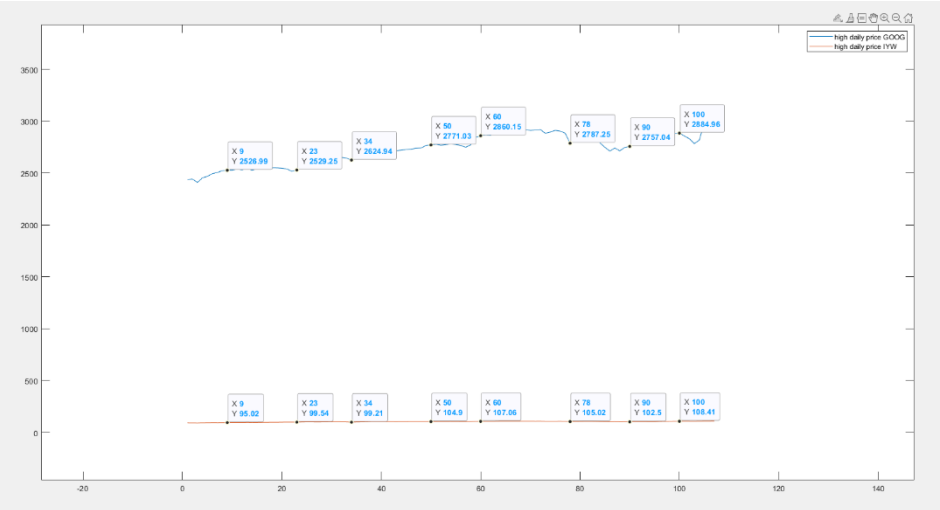
```
APSE = stockErr(y_true,y_predicted)        % stockErr compute the average percent squared error of predicted stock price.
```

```
APSE =

   8.0006e-04
```

6. Summary

Figure 1

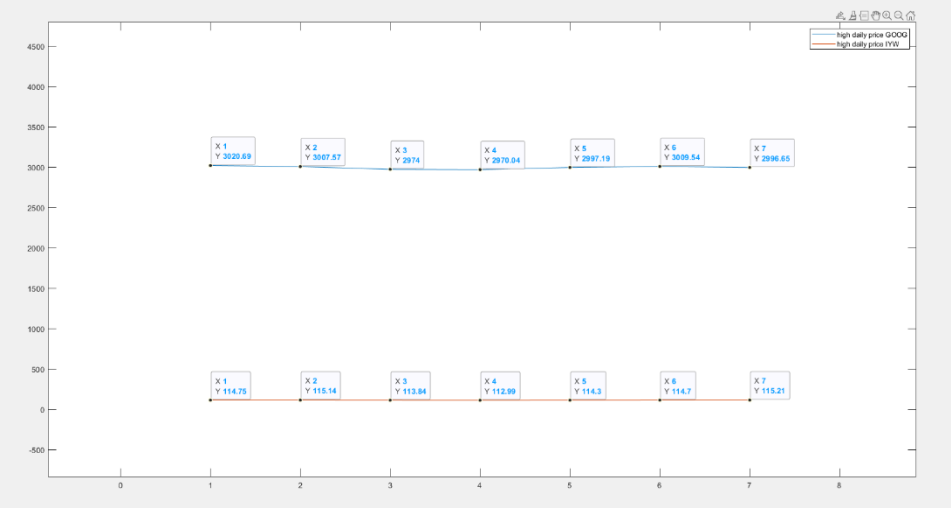



```
>> plot(x);
hold on;
plot(y);
legend('high daily price GOOG','high daily price IYW')
```

Data of 01/06/2021 - 31/10/2021

-From the data, the distances of the randomly selected intervals have similar increases and decreases. when computed, resulting in a relatively high value of Pearson's correlation coefficient, so the two variables are correlated in the same direction.

Figure 2

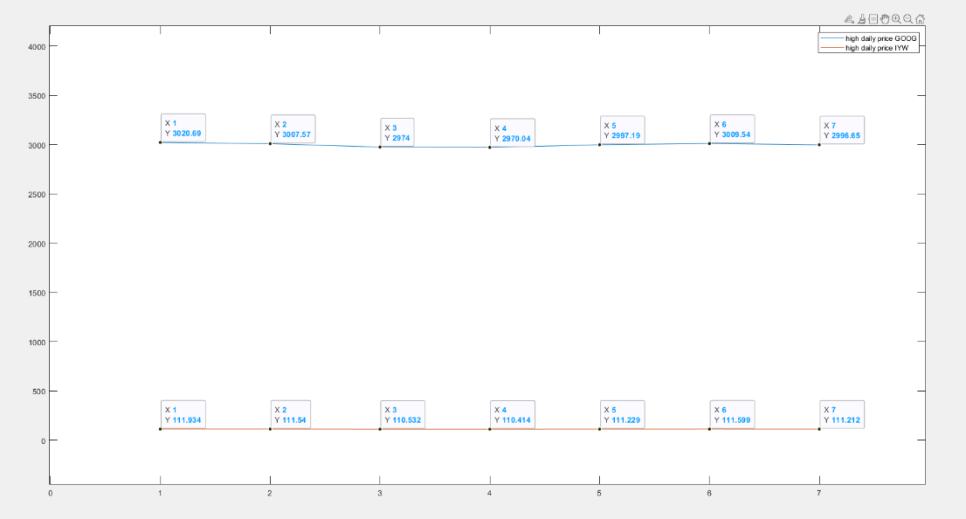



```
>> plot(xi);
hold on;
plot(y_true);
legend('high daily price GOOG','high daily price IYW')
```

Data of 08/01/2021 - 16/11/2021 (7 Days)

-From the data, the decrease and increase of values in each range are not equal.

Figure 3

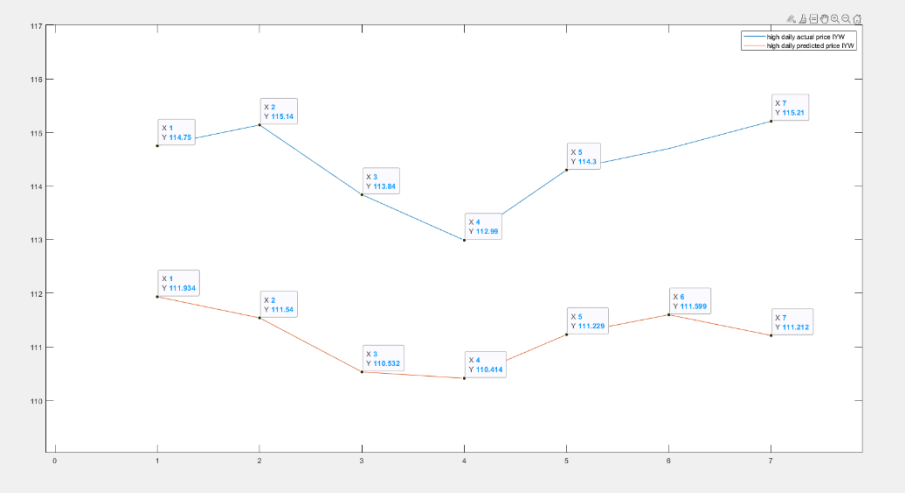



```
>> plot(xi);
hold on;
plot(y_predicted);
legend('high daily price GOOG','high daily price IYW')
>>
```

Data of 08/01/2021 - 16/11/2021 (7 Days)

-After predicting from the data and comparing it with the high daily price of GOOG, there are some intervals with different decrease and increase, possibly from randomly selecting intervals that are too far apart. It can be seen from the previous data in Figure 1 because due to the difference in the number of data Maybe the random period is not good enough.

Figure 4




```
>> plot(y_true);
hold on;
plot(y_predicted);
legend('high daily actual price IYW','high daily predicted price IYW')
```

Data of 08/01/2021 - 16/11/2021 (7 Days)

-From the 3 Figure data, the predicted value depends on x correlation, thus concluding that the higher the Pearson's correlation coefficient, the greater the Predictions are getting closer and closer to x. Or it can be predicted like x if Pearson's correlation coefficient = 1.

7.Final code

Complete code

stockErr.m | MiniProject2code.m

```
a = importdata('GOOG.csv');                % GOOG data from June 1, 2021 to Oct 31, 2021.
b = importdata('IYW.csv');                 % IYW data from June 1, 2021 to Oct 31, 2021.
y_true = importdata('IYW_08-16.csv');      % IYW from  November 8, to November 16, 2021
xi = importdata('GOOG_08-16.csv');         % GOOG data from  November 8, to November 16, 2021
x = a.data(:,2);                           % the daily high price from data a
y = b.data(:,2);                           % the daily high price from data b
y_true = y_true.data(:,2);                 % the daily high price from data y_true
xi = xi.data(:,2);                         % the daily high price from data Xi
zx = zscore(x,1);                          % Z-score of x
zy = zscore(y,1);                          % Z-score of y
r  = mean(zx.*zy)                          % r = The Pearson's correlation coefficient between x and y using Z-score to calculate
sx = std(x,1);                             % standard deviation of x
mx = mean(x);                              % mean of x
my = mean(y);                              % mean of y
c = mean((x-mx).*(y-my));                  % covariance of x and y
bhat1 = c/sx^2                             % bhat1 = covariance of x and y divided by variance of x
bhat0 = my-bhat1*mx                        % bhat0 = mean[y]- bhat1*mean[x]
                                           % regression equation is bhat0 + bhat1*xi;
y_predicted = bhat0 + bhat1*xi;            % find y_predicted by using regression equation
plot(y_true,'r');                          % plot y_true is actual prices of y
hold on;
plot(y_predicted,'b');                     % plot y_predicted is predicted prices of y from regression equation
legend('y true','y predicted')             % make legend for graph to explain
title('y true VS y predicted'); ylabel('Stock Price'); xlabel('Order of the date');
n2 = length(y_predicted);                  % amount of data y
MSE  = (sum((y_true - y_predicted).^2))/n2 % MSE = Mean square error
APSE = stockErr(y_true,y_predicted)        % stockErr compute the average percent squared error of predicted stock price.

```

Function stockErr

stockErr.m | MiniProject2code.m

```
function avgErr = stockErr(x,y)
% stockErr compute the average percent squared error of predicted stock price.
% Inputs: x is the real stock price
%         y is the predicted stock price from least square regression
% Output: avgErr is the average percent squared error of the prediction
%
% Author: Tulaya Limpiti
% Last update: Nov 5,2021

n = length(x);
n1 = length(y);
if (n ~= n1)
    fprintf('Length of data inputs are not the same!!! \n');
    return;
end

    avgErr = sum(((y-x)./x).^2)/n;
end

```