ECON M-524 FINANCIAL ECONOMETRICS FINAL PROJECT

Forecasting yen vs dollar using ARIMA model

By Sricharan Cheeti
Data Science Graduate Student
Indiana University Bloomington

INDEX

TABLE OF FIGURES:

1.RESEARCH QUESTION

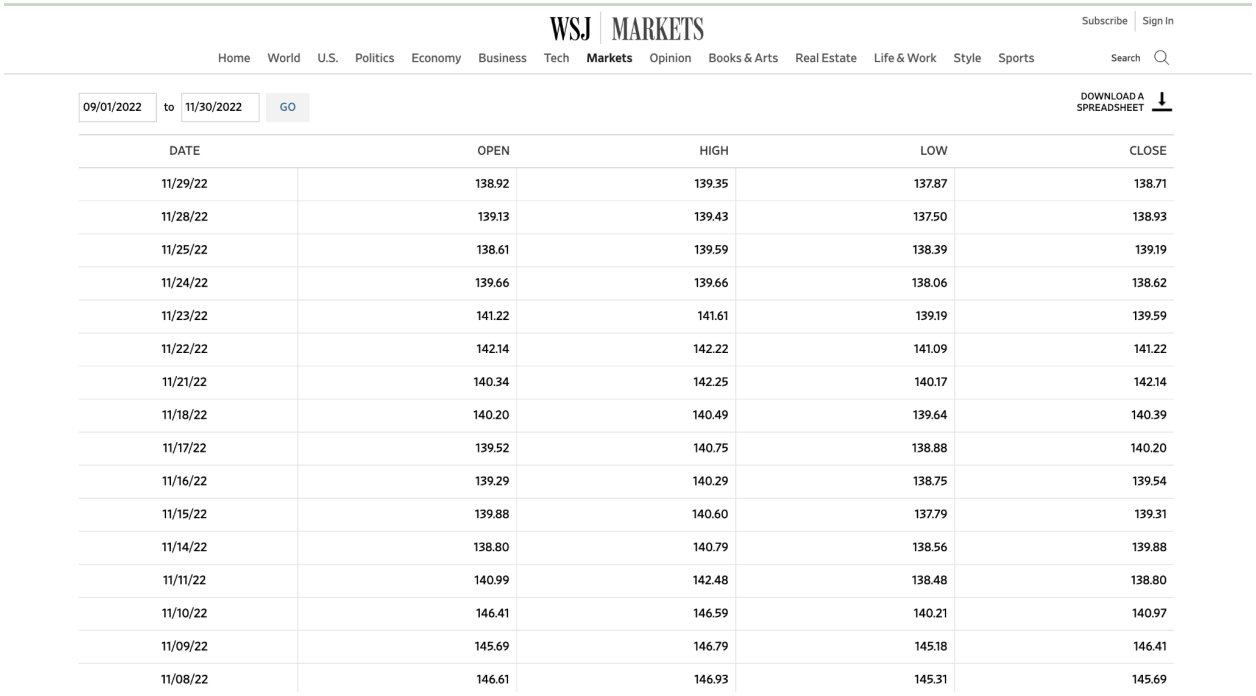
➔ Forecasting Yen vs Dollar exchange rates using Arima models.
➔ Used Multiple ARIMA models to check which combination works best.

2.INTRODUCTION

➔ The pair's exchange rate is one of the most liquid, not to mention one of the most traded, pairs in the world. That's because the yen, just like the U.S. dollar, is used as a reserve currency.
➔ The dollar/yen currency pair has traditionally had a close correlation with U.S. Treasuries. When yields on Treasury bonds, notes, and bills rise, the Yen tends to weaken relative to the dollar. This is because people can borrow Yen more cheaply to buy higher-yielding dollars.
➔ The USD/JPY pair can also be a determinant of market risk. For example, when markets are in search of risk trades, Treasury Bond yields rise as the economy grows. Yields are also a signal of risk. In the case that panic or fear hits the markets, Treasury bond prices tend to rise, causing yields to fall. In such a case the price of the U.S. dollar can weaken against the Yen.
➔ Nations with trade surplus will often see the USD/JPY pair as a favorable investment because the market traditionally views this pair as a chance to seek greater buying power and higher interest.
➔ When evaluating the relationship between the USD/JPY currency pair, the economic laws of supply and demand will ultimately serve as a strong factor in pricing but are also closely tied to bond pricing in their respective countries.

3.DATA SOURCE AND DESCRIPTION

➔ https://www.wsj.com/market-data/quotes/fx/USDJPY/historical-prices



WSJ | MARKETS

Subscribe | Sign In

Home World U.S. Politics Economy Business Tech **Markets** Opinion Books & Arts Real Estate Life & Work Style Sports Search

09/01/2022 to 11/30/2022 GO

DOWNLOAD A SPREADSHEET

DATE	OPEN	HIGH	LOW	CLOSE
11/29/22	138.92	139.35	137.87	138.71
11/28/22	139.13	139.43	137.50	138.93
11/25/22	138.61	139.59	138.39	139.19
11/24/22	139.66	139.66	138.06	138.62
11/23/22	141.22	141.61	139.19	139.59
11/22/22	142.14	142.22	141.09	141.22
11/21/22	140.34	142.25	140.17	142.14
11/18/22	140.20	140.49	139.64	140.39
11/17/22	139.52	140.75	138.88	140.20
11/16/22	139.29	140.29	138.75	139.54
11/15/22	139.88	140.60	137.79	139.31
11/14/22	138.80	140.79	138.56	139.88
11/11/22	140.99	142.48	138.48	138.80
11/10/22	146.41	146.59	140.21	140.97
11/09/22	145.69	146.79	145.18	146.41
11/08/22	146.61	146.93	145.31	145.69

➔ It contains 5 columns and 784 rows in total.
➔ Date-from 1/1/19 to 1/1/22
➔ Open-How much is the yen value opened when the market opened.
➔ High-The highest the yen value reached during the market hours on that day.
➔ Low-The least the yen value reached on that day.
➔ Close-The yen value when the market closed.

4.RESEARCH METHOD

➔ I have used the ARIMA model for forecasting.
➔ ARIMA(p,d,q)
➔ Auto Regressive Integrated Moving Average.
➔ This acronym is descriptive, capturing the key aspects of the model itself. Briefly, they are:

- **AR**: *Autoregression*. A model that uses the dependent relationship between an observation and some number of lagged observations.
- **I**: *Integrated*. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
- **MA**: *Moving Average*. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

➔ An ARIMA model is characterized by 3 terms: p, d, q where,

-> p is the order of the AR term

->q is the order of the MA term

->d is the number of differencing required to make the time series stationary.

5.STEPS:

- ➔ Imported dataset and took opening data from 2019-20 to build the ARIMA model.
- ➔ Plotted the opening dataset.
- ➔ Calculated ACF AND PACF , and plotted them.
- ➔ Fitted a basic theoretical model ARIMA(0,1,0).(The p value vas high than 0.05)
- ➔ Based on ACF and PACF try to find the p,q,d values.
- ➔ Found p-value to be 1 and tried to fit better ARIMA models.
- ➔ FItted ARIMA(1,0,0) and ARIMA(1,0,3) models.
- ➔ Plotted the predicted graphs.
- ➔ Used PMSE(Prediction Mean Square Error) and p-value to judge the models.
- ➔ ARIMA(1,0,3) proved to be the best of all.

6.OUTPUTS

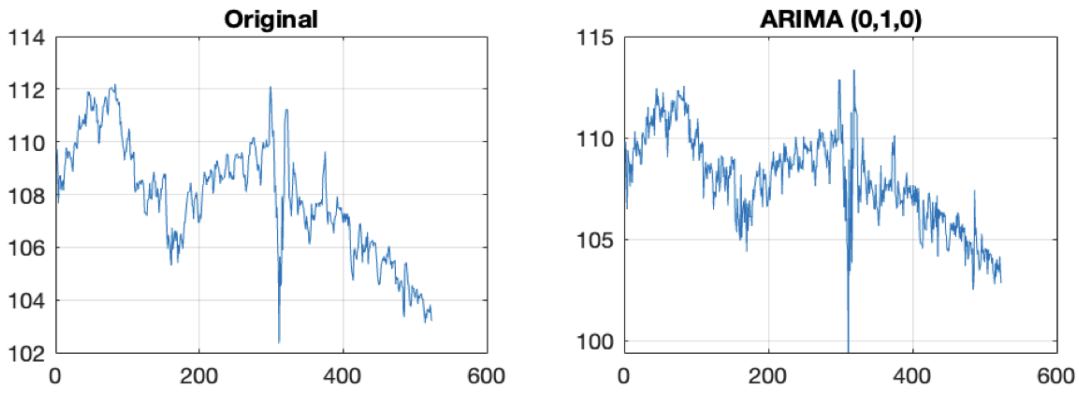


FIG 1: Original data for 2021 vs predicted data using ARIMA(0,1,0)

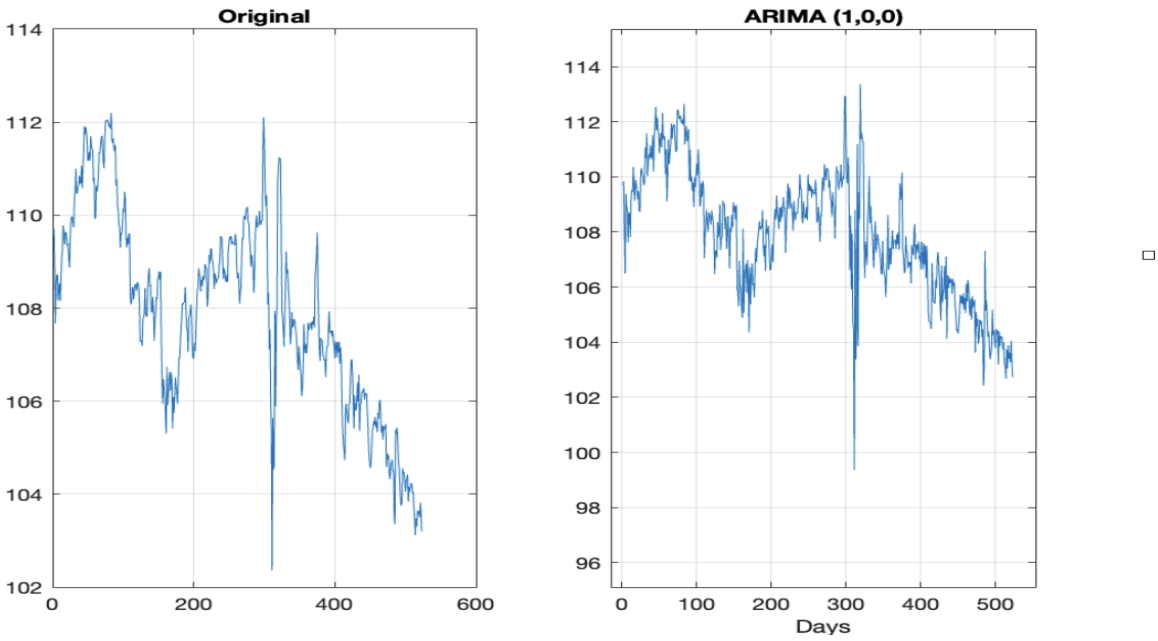


FIG 2: Original data for 2021 vs predicted data using ARIMA(1,0,0)

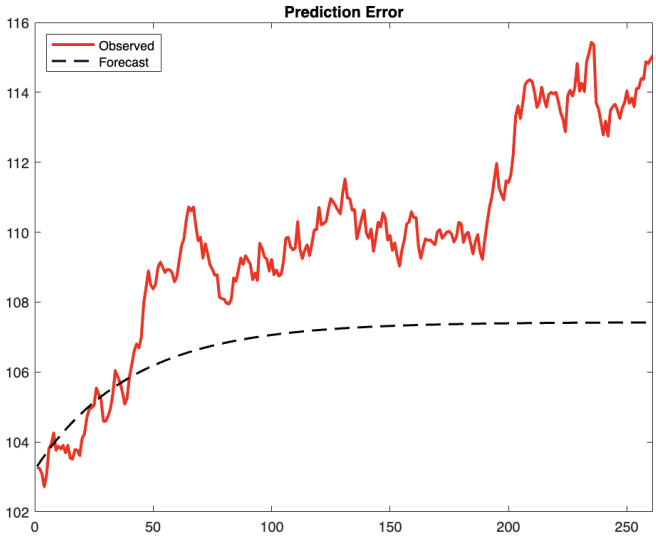


FIG 3:Original Yen exchange rate value vs Forecasted Yen value using ARIMA(1,0,0)

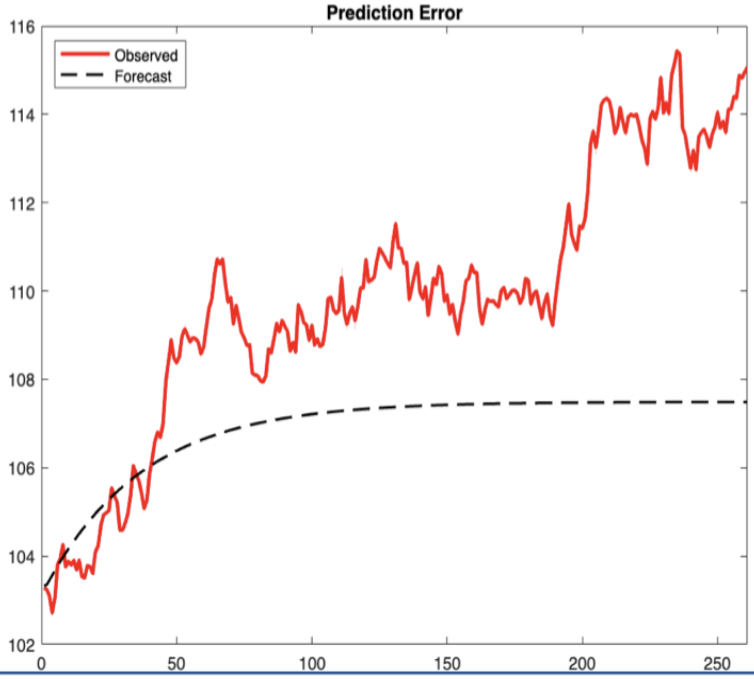


FIG 4:Original Yen value exchange rate vs Forecasted Yen value using ARIMA(1,0,3)

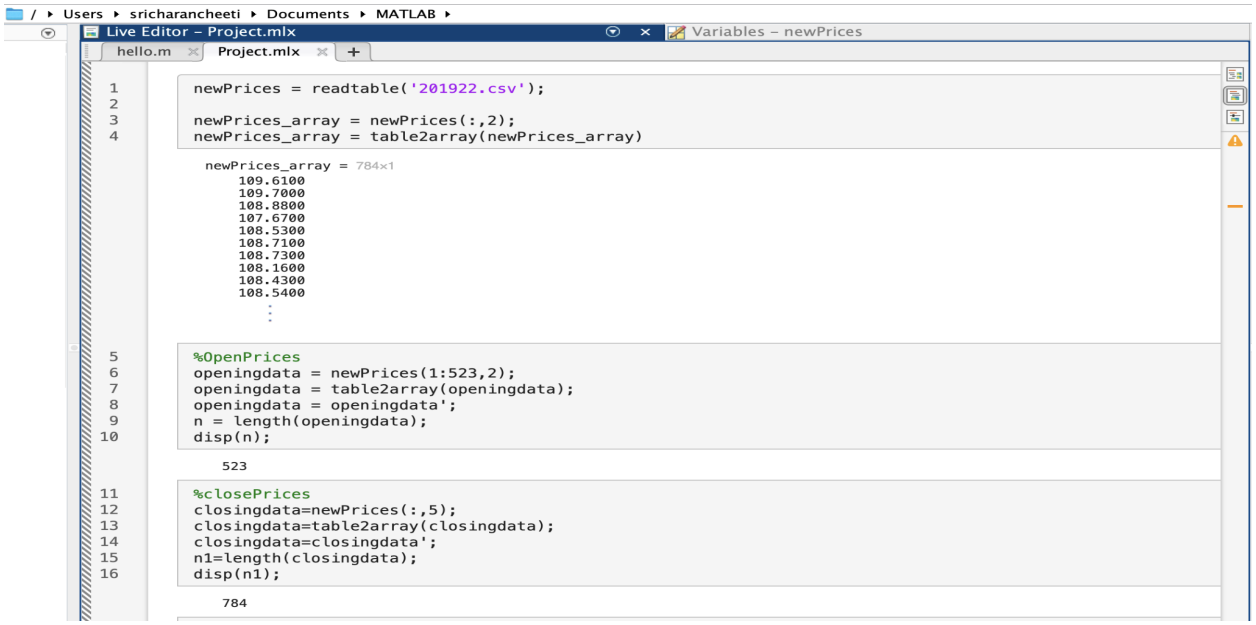
7.CONCLUSION

- ➔ The ARIMA model arima(1,0,3) has p value less than 0.005,therefore it is statistically significant.
- ➔ The Prediction Mean Square Error(PMSE) is still higher.
- ➔ We could use AIC and BIC selection criteria for finding a better fit ARIMA model.
- ➔ When it comes to investing real money, I would not base my decisions solely on the results of these models. Before using them, I would want to improve them by training the models and making them more statistically significant. If I did that, I would use them as one of the factors I consider, but would want to look at other factors as well before making any investment decisions.

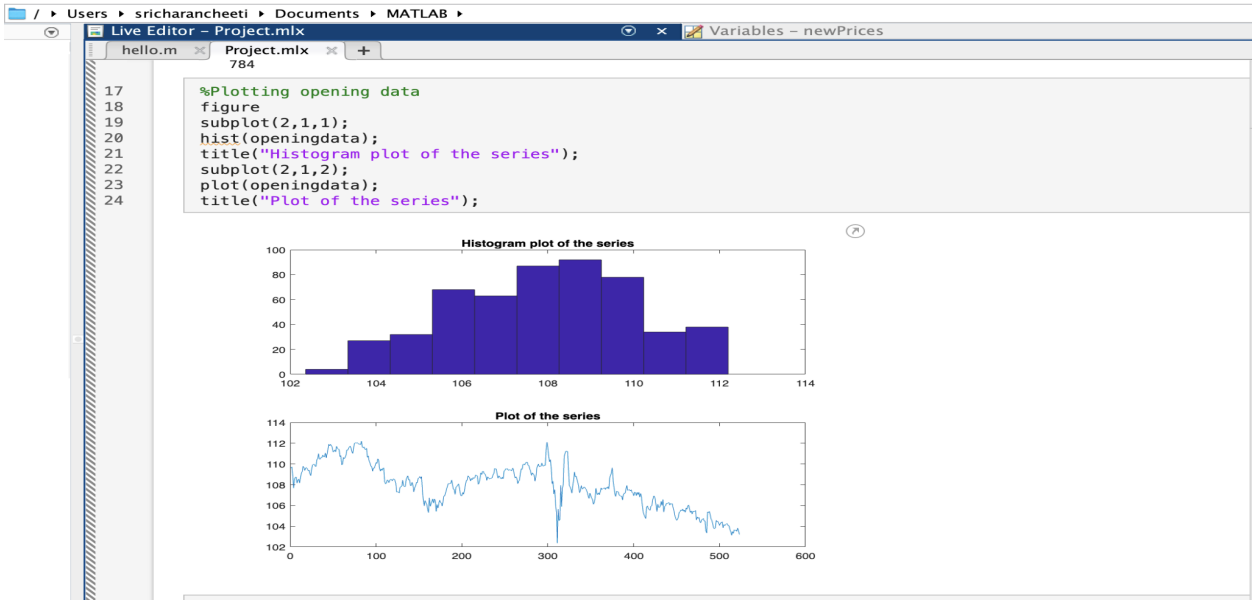
8.APPENDIX

MATLAB CODE:

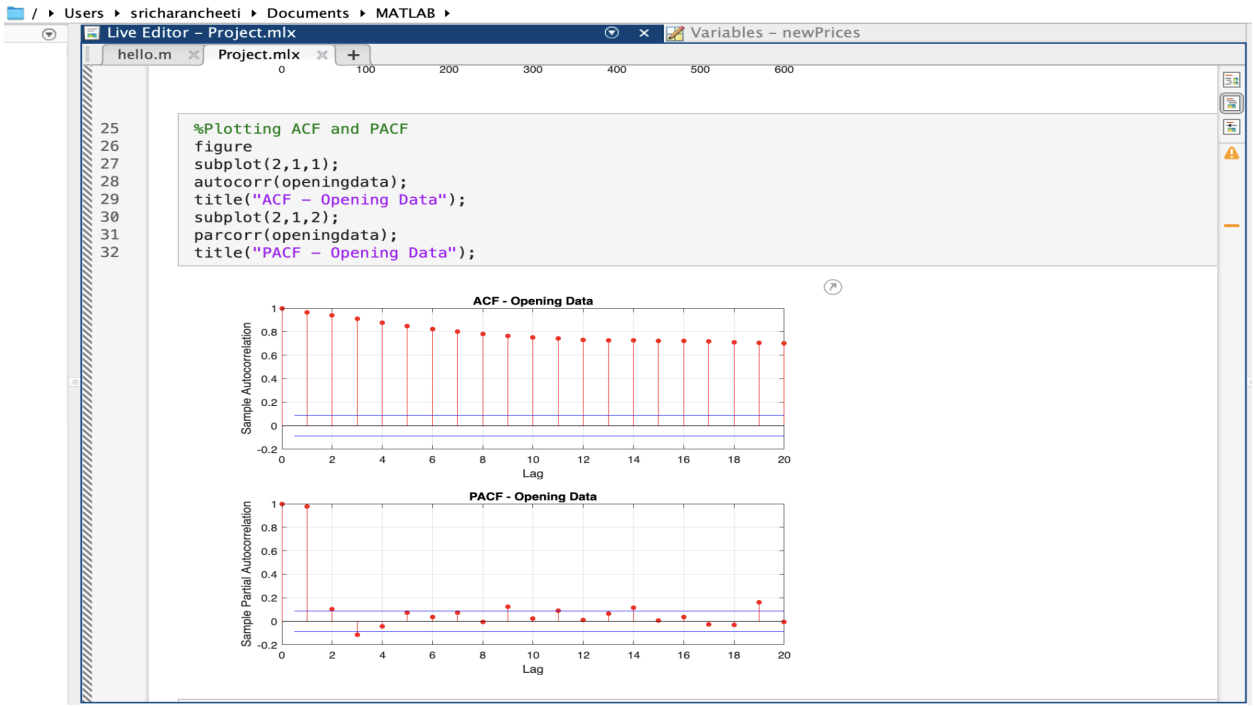
STEP 1:



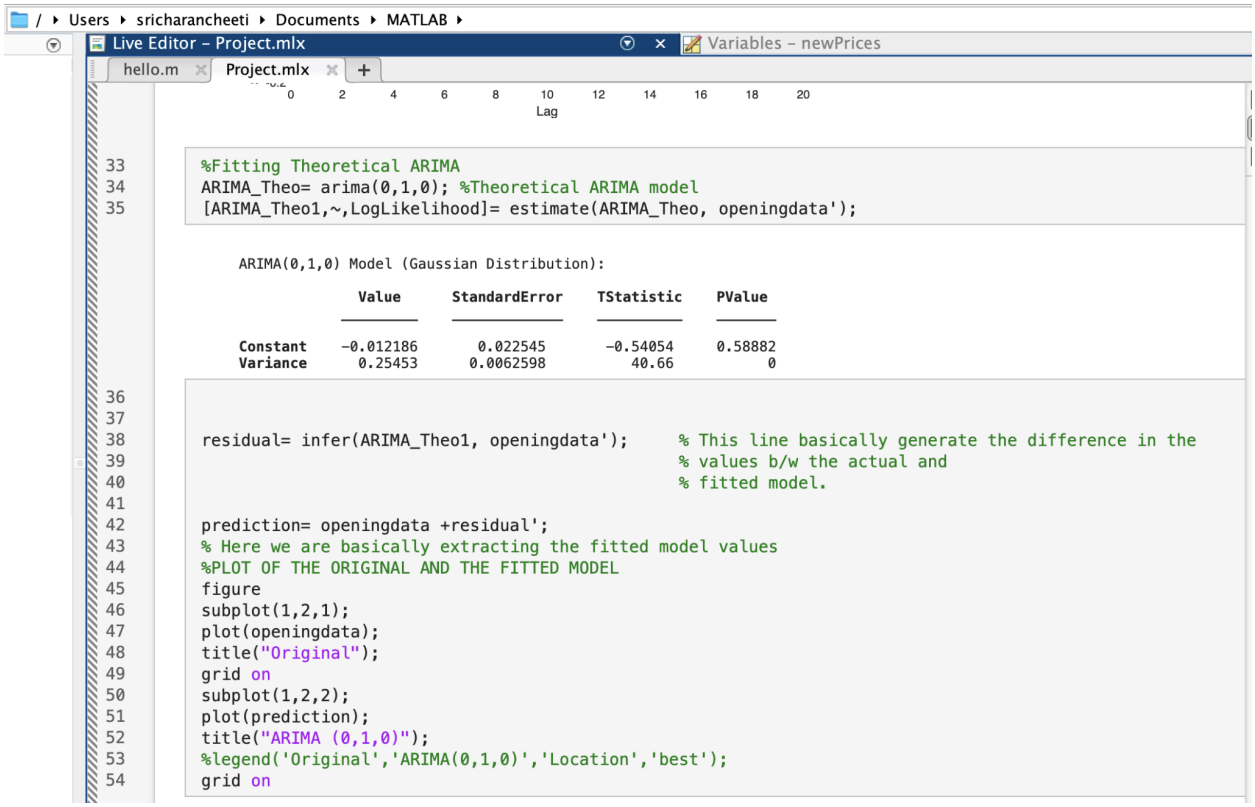
STEP 2:



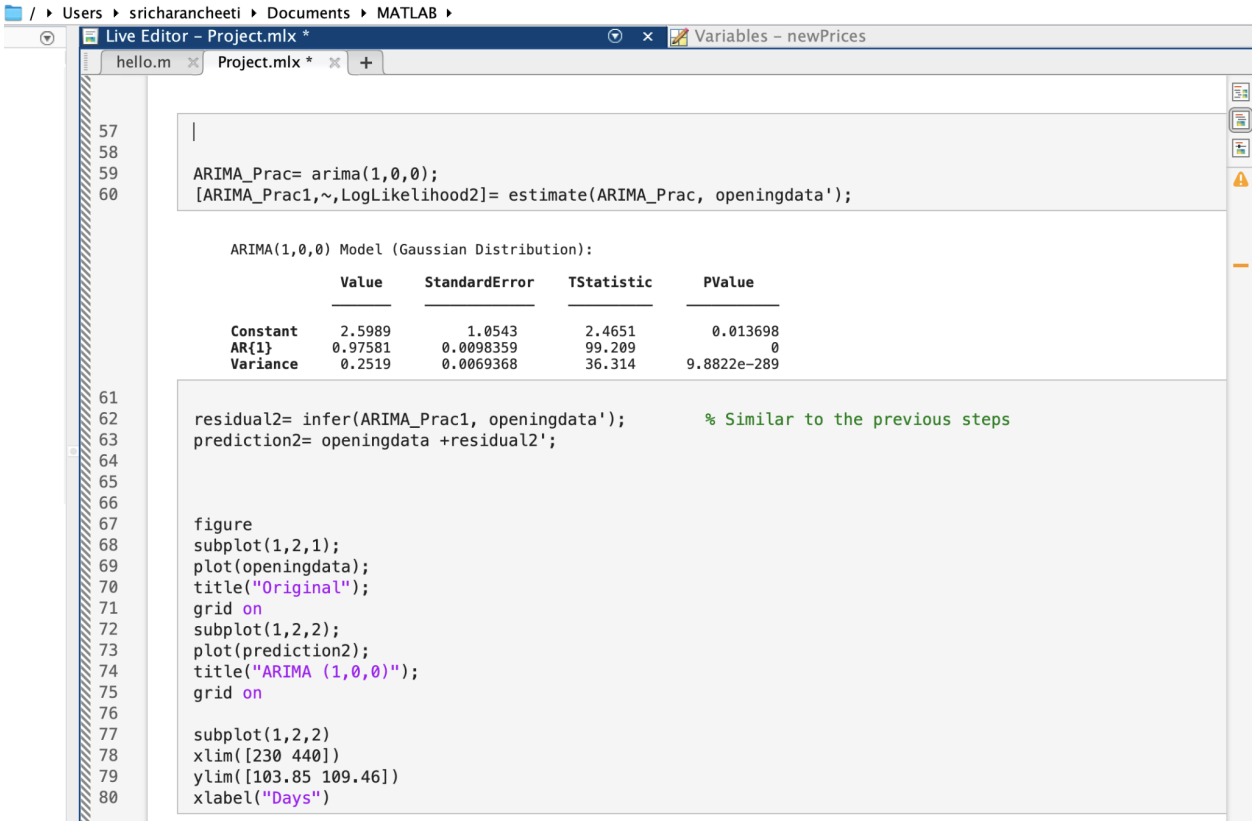
STEP 3:


/ ▸ Users ▸ sricharancheeti ▸ Documents ▸ MATLAB ▸
Live Editor – Project.mlx
Variables – newPrices
hello.m
Project.mlx
0 100 200 300 400 500 600
25 %Plotting ACF and PACF
26 figure
27 subplot(2,1,1);
28 autocorr(openingdata);
29 title("ACF – Opening Data");
30 subplot(2,1,2);
31 parcorr(openingdata);
32 title("PACF – Opening Data");
ACF - Opening Data
Sample Autocorrelation
Lag
PACF - Opening Data
Sample Partial Autocorrelation
Lag

STEP 4:


/ ▸ Users ▸ sricharancheeti ▸ Documents ▸ MATLAB ▸
Live Editor – Project.mlx
Variables – newPrices
hello.m
Project.mlx
0 2 4 6 8 10 12 14 16 18 20
Lag
33 %Fitting Theoretical ARIMA
34 ARIMA_Theo= arima(0,1,0); %Theoretical ARIMA model
35 [ARIMA_Theo1,~,LogLikelihood]= estimate(ARIMA_Theo, openingdata');
ARIMA(0,1,0) Model (Gaussian Distribution):
Value StandardError TStatistic PValue
Constant −0.012186 0.022545 −0.54054 0.58882
Variance 0.25453 0.0062598 40.66 0
36
37
38 residual= infer(ARIMA_Theo1, openingdata'); % This line basically generate the difference in the
39 % values b/w the actual and
40 % fitted model.
41
42 prediction= openingdata +residual';
43 % Here we are basically extracting the fitted model values
44 %PLOT OF THE ORIGINAL AND THE FITTED MODEL
45 figure
46 subplot(1,2,1);
47 plot(openingdata);
48 title("Original");
49 grid on
50 subplot(1,2,2);
51 plot(prediction);
52 title("ARIMA (0,1,0)");
53 %legend('Original','ARIMA(0,1,0)','Location','best');
54 grid on

STEP 5:




```
57
58
59  ARIMA_Prac= arima(1,0,0);
60  [ARIMA_Prac1,~,LogLikelihood2]= estimate(ARIMA_Prac, openingdata');
```

ARIMA(1,0,0) Model (Gaussian Distribution):

	Value	StandardError	TStatistic	PValue
Constant	2.5989	1.0543	2.4651	0.013698
AR{1}	0.97581	0.0098359	99.209	0
Variance	0.2519	0.0069368	36.314	9.8822e-289

```
61
62  residual2= infer(ARIMA_Prac1, openingdata');        % Similar to the previous steps
63  prediction2= openingdata +residual2';
64
65
66
67  figure
68  subplot(1,2,1);
69  plot(openingdata);
70  title("Original");
71  grid on
72  subplot(1,2,2);
73  plot(prediction2);
74  title("ARIMA (1,0,0)");
75  grid on
76
77  subplot(1,2,2)
78  xlim([230 440])
79  ylim([103.85 109.46])
80  xlabel("Days")
```

STEP 6:

```
Md=estimate(ARIMA_Prac1,a);
```

ARIMA(1,0,0) Model (Gaussian Distribution):

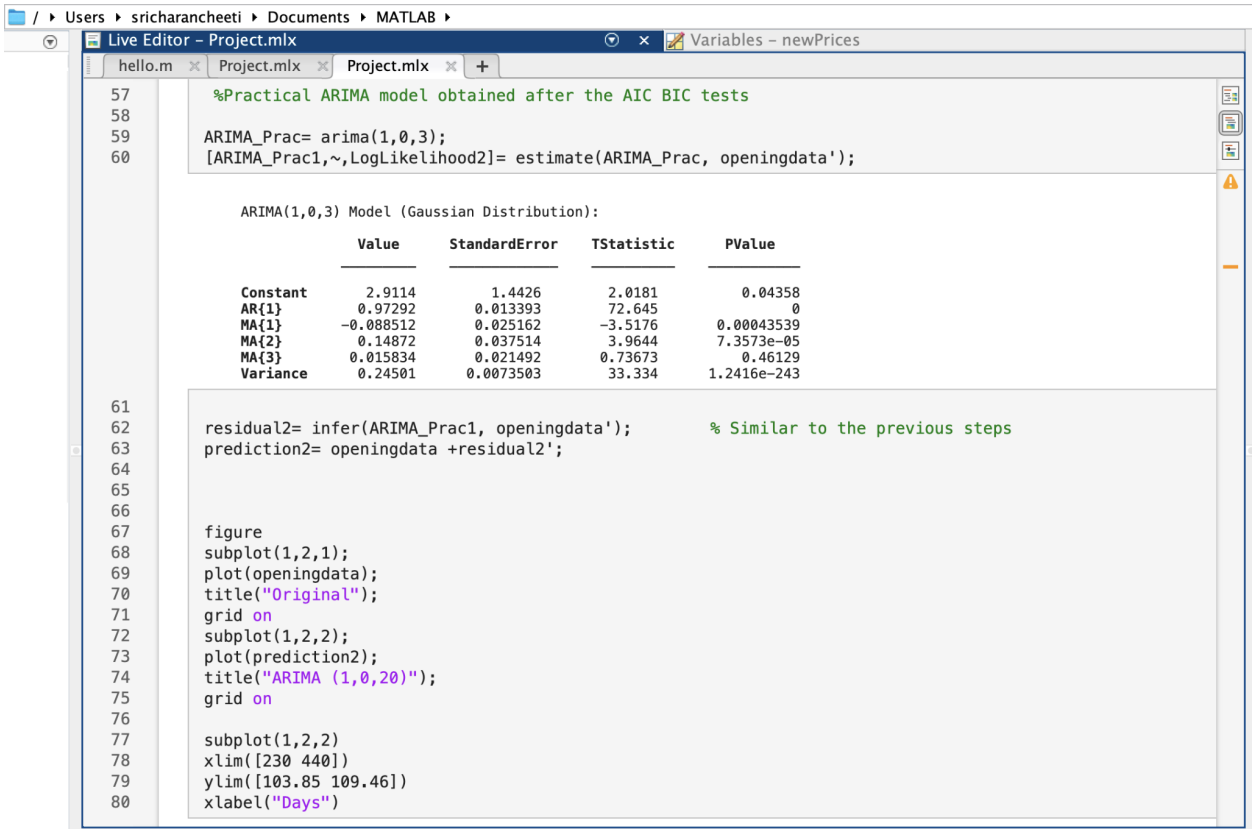
	Value	StandardError	TStatistic	PValue
Constant	2.5989	0	Inf	0
Variance	0.2519	0	Inf	0

```
f=forecast(Md,261,'Y0',a);
pmse = mean((b-f).^2)
```

```
pmse = 14.3681
```

```
figure
plot(b,'r','LineWidth',2)
hold on
plot(f,'k--','LineWidth',1.5)
xlim([0,261])
title('Prediction Error')
legend('Observed','Forecast','Location','northwest')
hold off
```

STEP 7:



STEP 8:

```
pmse = mean((b-f).^2)
```

```
pmse = 13.7493
```

```
figure
plot(b,'r','LineWidth',2)
hold on
plot(f,'k--','LineWidth',1.5)
xlim([0,261])
title('Prediction Error')
legend('Observed','Forecast','Location','northwest')
hold off
```