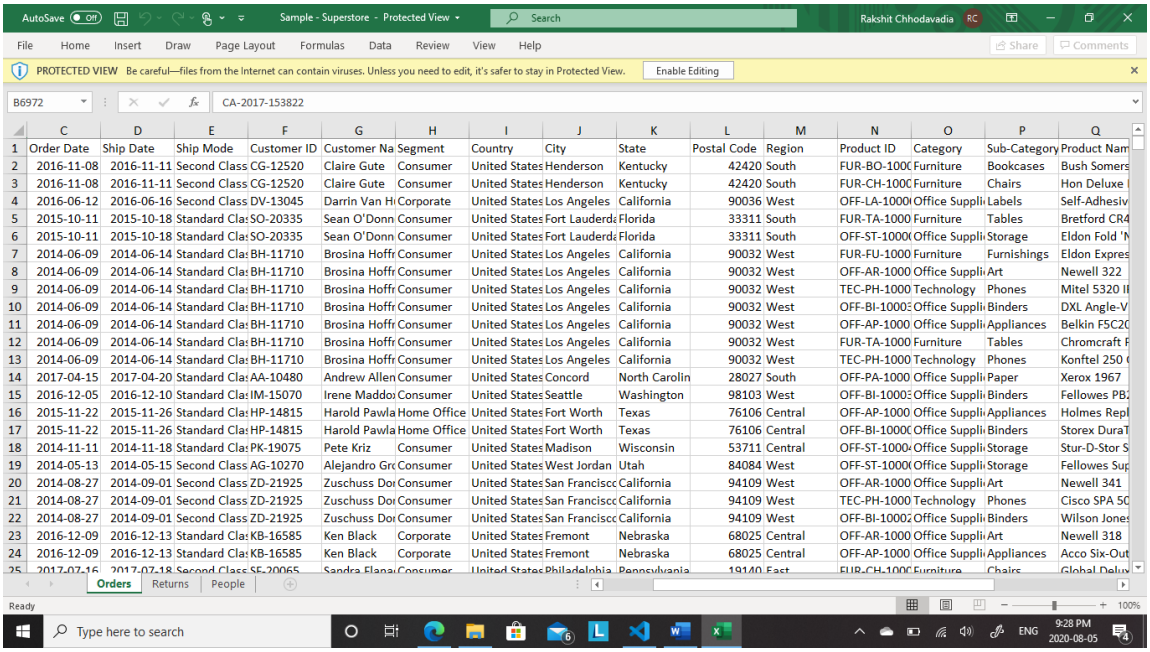
Github Link: https://github.com/helivarma/Time-Series-Analysis-for-Super-Market

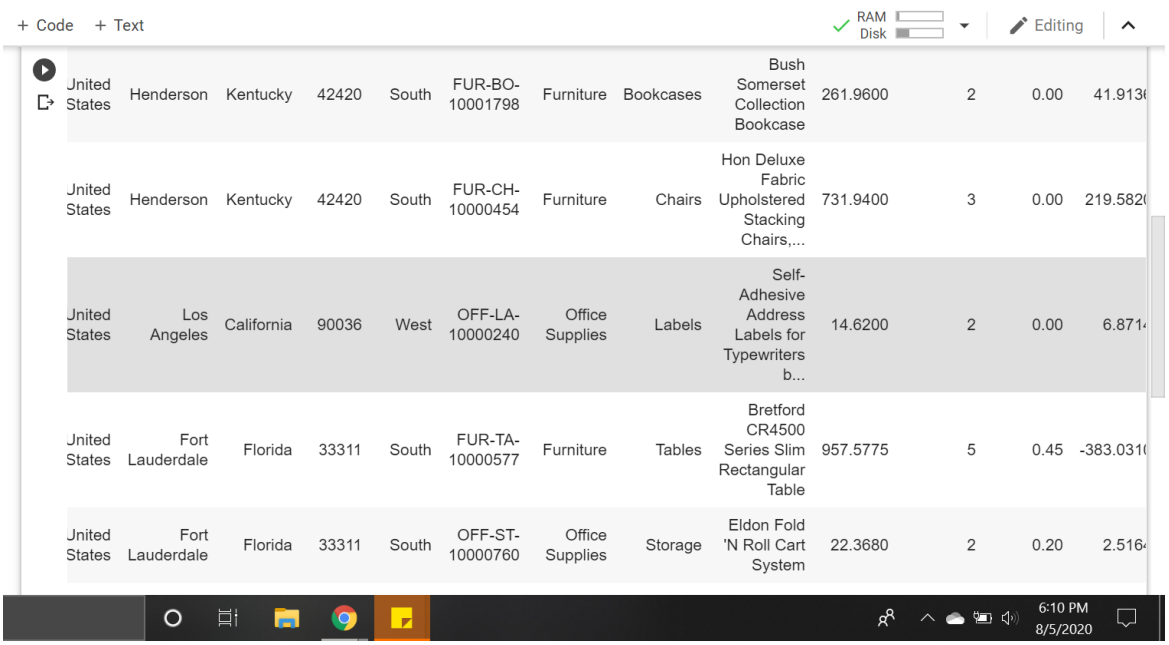
Here are fer screenshoots of the implementation completed so far.

Dataset: The dataset has the fields such as, Row ID, Order ID, Order Date, Ship Date, Ship Mode, Customer ID, Customer Name and so on.



	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Order Date	Ship Date	Ship Mode	Customer ID	Customer Na	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category	Product Nam
2	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-BO-1000	Furniture	Bookcases	Bush Somers
3	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-CH-1000	Furniture	Chairs	Hon Deluxe
4	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van H	Corporate	United States	Los Angeles	California	90036	West	OFF-LA-1000	Office Suppli	Labels	Self-Adhesiv
5	2015-10-11	2015-10-18	Standard Cla	SO-20335	Sean O'Donn	Consumer	United States	Fort Lauderda	Florida	33311	South	FUR-TA-1000	Furniture	Tables	Bretford CR4
6	2015-10-11	2015-10-18	Standard Cla	SO-20335	Sean O'Donn	Consumer	United States	Fort Lauderda	Florida	33311	South	OFF-ST-1000	Office Suppli	Storage	Eldon Fold 'N
7	2014-06-09	2014-06-14	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	FUR-FU-1000	Furniture	Furnishings	Eldon Expres
8	2014-06-09	2014-06-14	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	OFF-AR-1000	Office Suppli	Art	Newell 322
9	2014-06-09	2014-06-14	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	TEC-PH-1000	Technology	Phones	Mitel 5320 I
10	2014-06-09	2014-06-14	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	OFF-BI-1000	Office Suppli	Binders	DXL Angle-V
11	2014-06-09	2014-06-14	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	OFF-AP-1000	Office Suppli	Appliances	Belkin F5C20
12	2014-06-09	2014-06-14	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	FUR-TA-1000	Furniture	Tables	Chromcraft F
13	2014-06-09	2014-06-14	Standard Cla	BH-11710	Brosina Hoffr	Consumer	United States	Los Angeles	California	90032	West	TEC-PH-1000	Technology	Phones	Konftel 250
14	2017-04-15	2017-04-20	Standard Cla	AA-10480	Andrew Allen	Consumer	United States	Concord	North Carolin	28027	South	OFF-PA-1000	Office Suppli	Paper	Xerox 1967
15	2016-12-05	2016-12-10	Standard Cla	IM-15070	Irene Maddox	Consumer	United States	Seattle	Washington	98103	West	OFF-BI-1000	Office Suppli	Binders	Fellowes PB
16	2015-11-22	2015-11-26	Standard Cla	HP-14815	Harold Pawla	Home Office	United States	Fort Worth	Texas	76106	Central	OFF-AP-1000	Office Suppli	Appliances	Holmes Repl
17	2015-11-22	2015-11-26	Standard Cla	HP-14815	Harold Pawla	Home Office	United States	Fort Worth	Texas	76106	Central	OFF-BI-1000	Office Suppli	Binders	Storex DuraT
18	2014-11-11	2014-11-18	Standard Cla	PK-19075	Pete Kriz	Consumer	United States	Madison	Wisconsin	53711	Central	OFF-ST-1000	Office Suppli	Storage	Stur-D-Stor S
19	2014-05-13	2014-05-15	Second Class	AG-10270	Alejandro Gr	Consumer	United States	West Jordan	Utah	84084	West	OFF-ST-1000	Office Suppli	Storage	Fellowes Sup
20	2014-08-27	2014-09-01	Second Class	ZD-21925	Zuschuss Do	Consumer	United States	San Francisc	California	94109	West	OFF-AR-1000	Office Suppli	Art	Newell 341
21	2014-08-27	2014-09-01	Second Class	ZD-21925	Zuschuss Do	Consumer	United States	San Francisc	California	94109	West	TEC-PH-1000	Technology	Phones	Cisco SPA 50
22	2014-08-27	2014-09-01	Second Class	ZD-21925	Zuschuss Do	Consumer	United States	San Francisc	California	94109	West	OFF-BI-1000	Office Suppli	Binders	Wilson Jones
23	2016-12-09	2016-12-13	Standard Cla	KB-16585	Ken Black	Corporate	United States	Fremont	Nebraska	68025	Central	OFF-AR-1000	Office Suppli	Art	Newell 318
24	2016-12-09	2016-12-13	Standard Cla	KB-16585	Ken Black	Corporate	United States	Fremont	Nebraska	68025	Central	OFF-AP-1000	Office Suppli	Appliances	Acco Six-Out
25	2017-07-16	2017-07-18	Second Class	SF-20065	Sandra Flana	Consumer	United States	Philadelphia	Pennsylvania	19140	East	FUR-CH-1000	Furniture	Chairs	Global Delux

Figure 1 Original Data



	Jnited States	Henderson	Kentucky	42420	South	FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.9600	2	0.00	41.913
	Jnited States	Henderson	Kentucky	42420	South	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400	3	0.00	219.582
	Jnited States	Los Angeles	California	90036	West	OFF-LA-10000240	Office Supplies	Labels	Self-Adhesive Address Labels for Typewriters b...	14.6200	2	0.00	6.871
	Jnited States	Fort Lauderdale	Florida	33311	South	FUR-TA-10000577	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table	957.5775	5	0.45	-383.031
	Jnited States	Fort Lauderdale	Florida	33311	South	OFF-ST-10000760	Office Supplies	Storage	Eldon Fold 'N Roll Cart System	22.3680	2	0.20	2.516

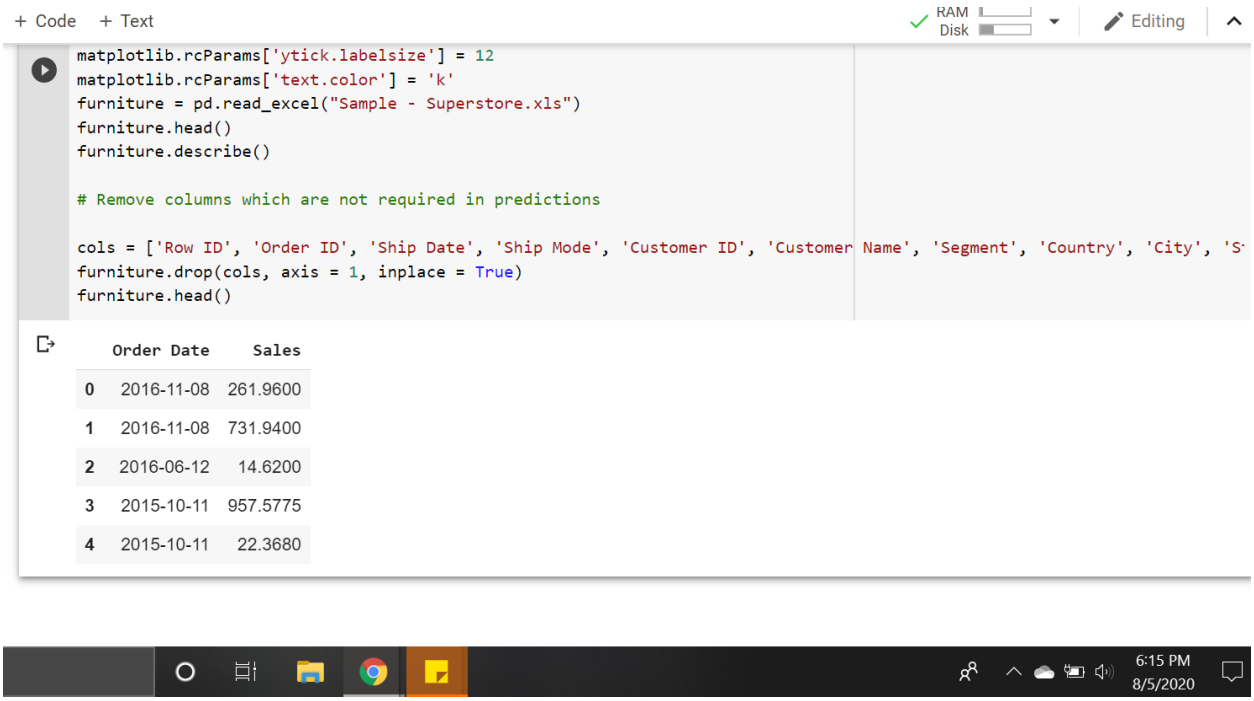
Figure 2 Data Retrieved using python code

```
matplotlib.rcParams['ytick.labelsize'] = 12
matplotlib.rcParams['text.color'] = 'k'
furniture = pd.read_excel("Sample - Superstore.xls")
furniture.head()
furniture.describe()
```

	Row ID	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	4997.500000	55190.379428	229.858001	3.789574	0.156203	28.656896
std	2885.163629	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1.000000	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	2499.250000	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	4997.500000	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	7495.750000	90008.000000	209.940000	5.000000	0.200000	29.364000
max	9994.000000	99301.000000	22638.480000	14.000000	0.800000	8399.976000

Figure 3 Data Description

Data Preprocessing: After reading dataset, with the help of data preprocessing, we removed the unwanted columns, sort them in the decreasing order of order date and check for any missing values, grouping them based on the order date. Further we searched for the oldest and the latest order, calculated the average sales of each month.



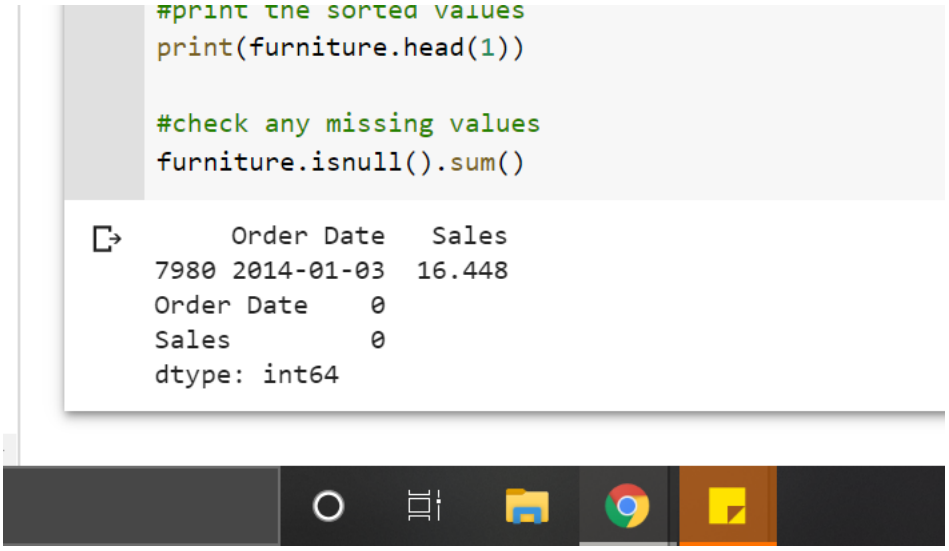
```
matplotlib.rcParams['ytick.labelsize'] = 12
matplotlib.rcParams['text.color'] = 'k'
furniture = pd.read_excel("Sample - Superstore.xls")
furniture.head()
furniture.describe()

# Remove columns which are not required in predictions

cols = ['Row ID', 'Order ID', 'Ship Date', 'Ship Mode', 'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'S
furniture.drop(cols, axis = 1, inplace = True)
furniture.head()
```

	Order Date	Sales
0	2016-11-08	261.9600
1	2016-11-08	731.9400
2	2016-06-12	14.6200
3	2015-10-11	957.5775
4	2015-10-11	22.3680

Figure 4 Data Preprocessing(Removing the unnecessary columns)



```
#print the sorted values
print(furniture.head(1))

#check any missing values
furniture.isnull().sum()
```

```
      Order Date   Sales
7980 2014-01-03  16.448
Order Date    0
Sales         0
dtype: int64
```

Figure 5 Data Preprocessing(sorting)

```
print(furniture['Order Date'].min())
print(furniture['Order Date'].max())
```

```
      Order Date   Sales
7980 2014-01-03  16.448
2014-01-03 00:00:00
2017-12-30 00:00:00
```

Figure 6 Data Processing(Finding the latest and the oldest record)

```
# grouping sales according to Order Date
furniture.groupby('Order Date')['Sales'].sum().reset_index()

# min and max values of Order Date
print(furniture['Order Date'].min())
print(furniture['Order Date'].max())

furniture = furniture.set_index('Order Date')
furniture.index
```

```
      Order Date   Sales
7980 2014-01-03  16.448
2014-01-03 00:00:00
2017-12-30 00:00:00
DatetimeIndex(['2014-01-03', '2014-01-04', '2014-01-04', '2014-01-04',
               '2014-01-05', '2014-01-06', '2014-01-06', '2014-01-06',
               '2014-01-06', '2014-01-06',
               ...
               '2017-12-29', '2017-12-29', '2017-12-29', '2017-12-30',
               '2017-12-30', '2017-12-30', '2017-12-30', '2017-12-30',
               '2017-12-30', '2017-12-30'],
              dtype='datetime64[ns]', name='Order Date', length=9994, freq=None)
```

Figure 7 Data Preprocessing (Grouping the order based on Order Date)

```
      Order Date   Sales
7980 2014-01-03  16.448
2014-01-03 00:00:00
2017-12-30 00:00:00
Order Date
2017-01-01    283.686284
2017-02-01    189.730219
2017-03-01    247.362827
2017-04-01    179.909045
2017-05-01    182.897150
2017-06-01    216.251942
2017-07-01    200.285027
2017-08-01    289.545358
2017-09-01    191.430614
2017-10-01    260.996387
2017-11-01    258.056264
2017-12-01    181.448742
Freq: MS, Name: Sales, dtype: float64
```

Figure 8 Data Preprocessing (Finding the Average Sales of each month)

Data Visualization: It allows us to decompose our time series into three distinct components

- Trend
- Seasonality
- Noise

Figure 9 data Visualization (Plotting the Average Sales)

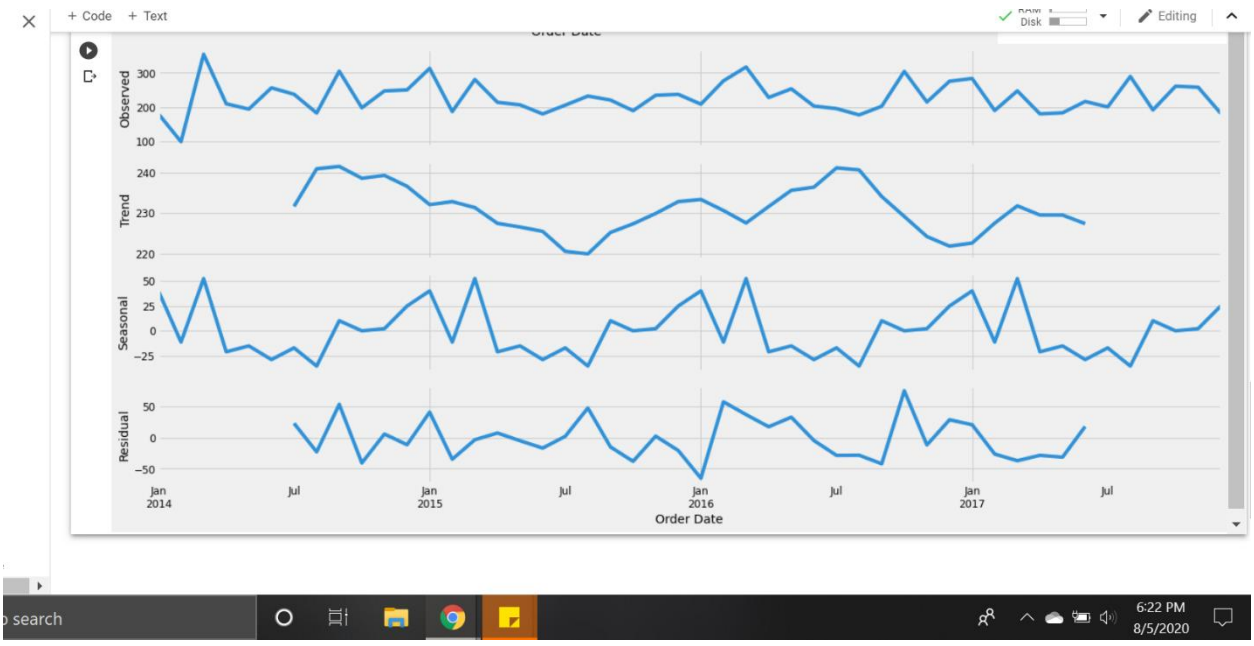


Figure 10 Decomposing the Data into Obesrved Trends and Residual

Time series forecasting with ARIMA model

About ARIMA model

- ARIMA stands for Autoregressive Integrated Moving Average
- ARIMA models are denoted with the notation ARIMA (p, d, q)
- These three parameters account for seasonality, trend, and noise in data

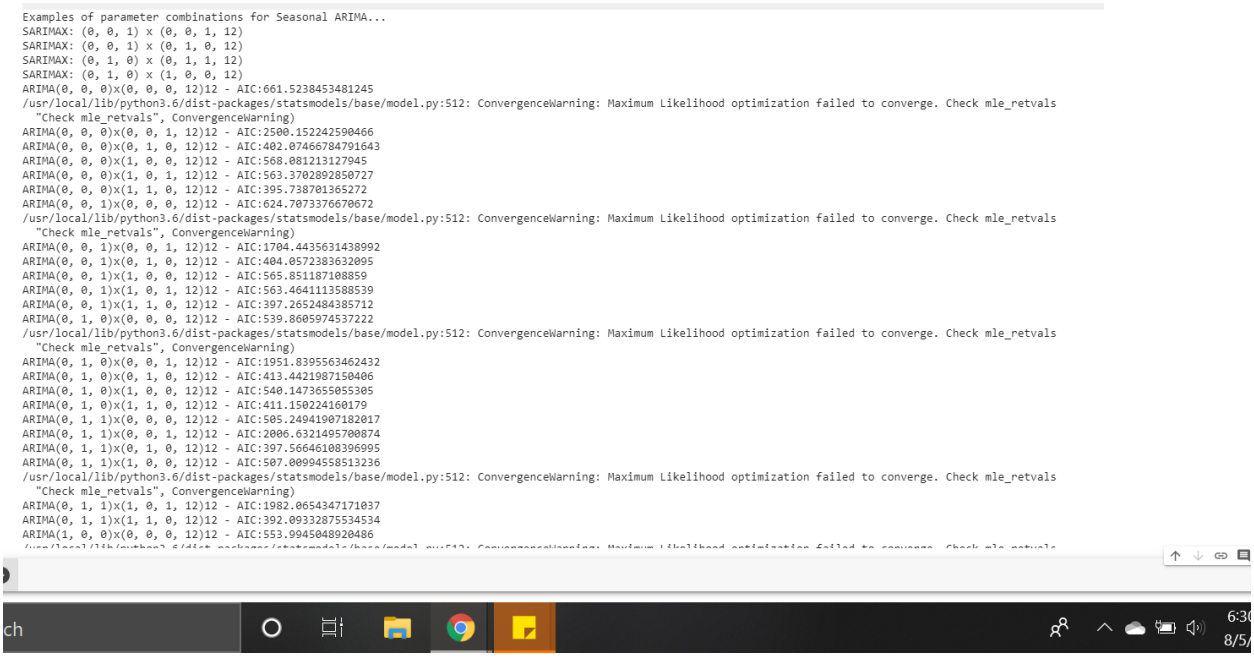
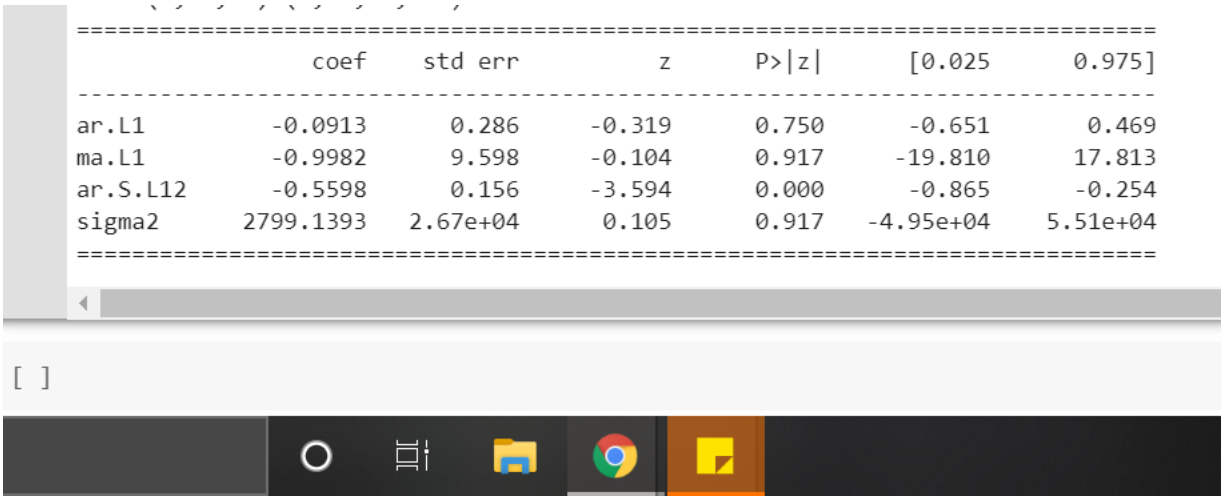


Figure 11 ARIMA Model

```
ARIMA(0, 1, 1)x(1, 1, 0, 12)12 - AIC:392.0933287553454
ARIMA(1, 0, 0)x(0, 0, 0, 12)12 - AIC:553.9945048920486
/usr/local/lib/python3.6/dist-packages/statsmodels/base/model.py:512: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
ARIMA(1, 0, 0)x(0, 0, 1, 12)12 - AIC:2428.586932370097
ARIMA(1, 0, 0)x(0, 1, 0, 12)12 - AIC:404.06203007562084
ARIMA(1, 0, 0)x(1, 0, 0, 12)12 - AIC:552.9272857949779
ARIMA(1, 0, 0)x(1, 0, 1, 12)12 - AIC:553.580386072686
ARIMA(1, 0, 0)x(1, 1, 0, 12)12 - AIC:397.2757103427853
/usr/local/lib/python3.6/dist-packages/statsmodels/base/model.py:512: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
ARIMA(1, 0, 1)x(0, 0, 0, 12)12 - AIC:520.9352938663197
/usr/local/lib/python3.6/dist-packages/statsmodels/base/model.py:512: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
ARIMA(1, 0, 1)x(0, 0, 1, 12)12 - AIC:2023.1470630212646
ARIMA(1, 0, 1)x(0, 1, 0, 12)12 - AIC:405.5034128353188
/usr/local/lib/python3.6/dist-packages/statsmodels/base/model.py:512: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
ARIMA(1, 0, 1)x(1, 0, 0, 12)12 - AIC:522.7348139302701
ARIMA(1, 0, 1)x(1, 0, 1, 12)12 - AIC:524.6939401789598
ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:396.08182424061636
ARIMA(1, 1, 0)x(0, 0, 0, 12)12 - AIC:519.8818905467256
/usr/local/lib/python3.6/dist-packages/statsmodels/base/model.py:512: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
ARIMA(1, 1, 0)x(0, 0, 1, 12)12 - AIC:1966.7083459776345
ARIMA(1, 1, 0)x(0, 1, 0, 12)12 - AIC:408.0593708055824
ARIMA(1, 1, 0)x(1, 0, 0, 12)12 - AIC:521.8609414869703
ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:2008.771572384967
ARIMA(1, 1, 0)x(1, 1, 0, 12)12 - AIC:399.8530806573002
ARIMA(1, 1, 1)x(0, 0, 0, 12)12 - AIC:505.5628866933248
/usr/local/lib/python3.6/dist-packages/statsmodels/base/model.py:512: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
ARIMA(1, 1, 1)x(0, 0, 1, 12)12 - AIC:2044.8177434355503
ARIMA(1, 1, 1)x(0, 1, 0, 12)12 - AIC:399.56190080712844
ARIMA(1, 1, 1)x(1, 0, 0, 12)12 - AIC:507.5224773702229
/usr/local/lib/python3.6/dist-packages/statsmodels/base/model.py:512: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
ARIMA(1, 1, 1)x(1, 0, 1, 12)12 - AIC:1970.7368058191455
ARIMA(1, 1, 1)x(1, 1, 0, 12)12 - AIC:393.848198764369
```

[ ]

Figure 12 ARIMA Model



	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.0913	0.286	-0.319	0.750	-0.651	0.469
ma.L1	-0.9982	9.598	-0.104	0.917	-19.810	17.813
ar.S.L12	-0.5598	0.156	-3.594	0.000	-0.865	-0.254
sigma2	2799.1393	2.67e+04	0.105	0.917	-4.95e+04	5.51e+04

[ ]

Figure 13 Fitting the ARIMA Model

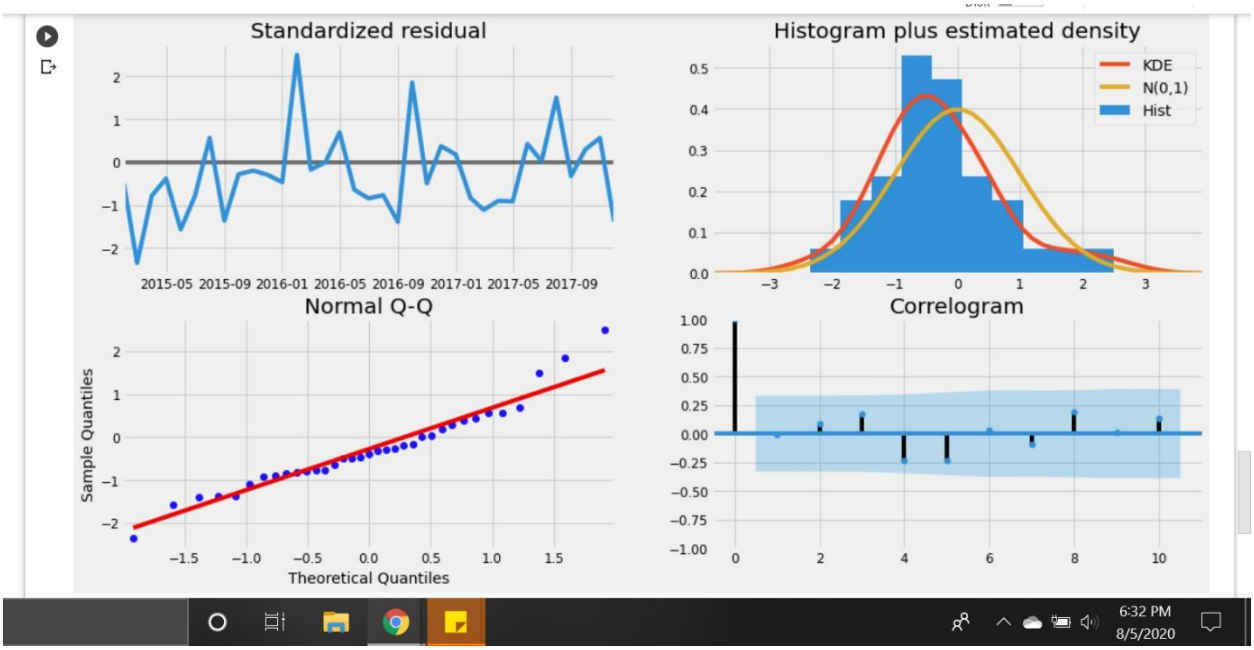


Figure 14 Model Residual

```
prediction_ci = prediction.conf_int()
prediction_ci
```

Order Date	lower Sales	upper Sales
2017-01-01	166.949829	380.855435
2017-02-01	128.374777	341.873715
2017-03-01	201.126554	414.265415
2017-04-01	122.357190	335.174994
2017-05-01	126.153430	338.683179
2017-06-01	87.013281	299.283139
2017-07-01	92.953016	304.987211
2017-08-01	102.323359	314.142881
2017-09-01	103.098060	314.721217
2017-10-01	139.372347	350.815200
2017-11-01	122.030718	333.307437
2017-12-01	150.521236	361.644383

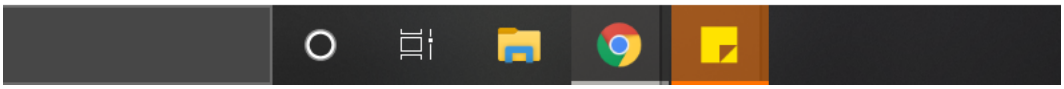


Figure 15 Validating Forecast

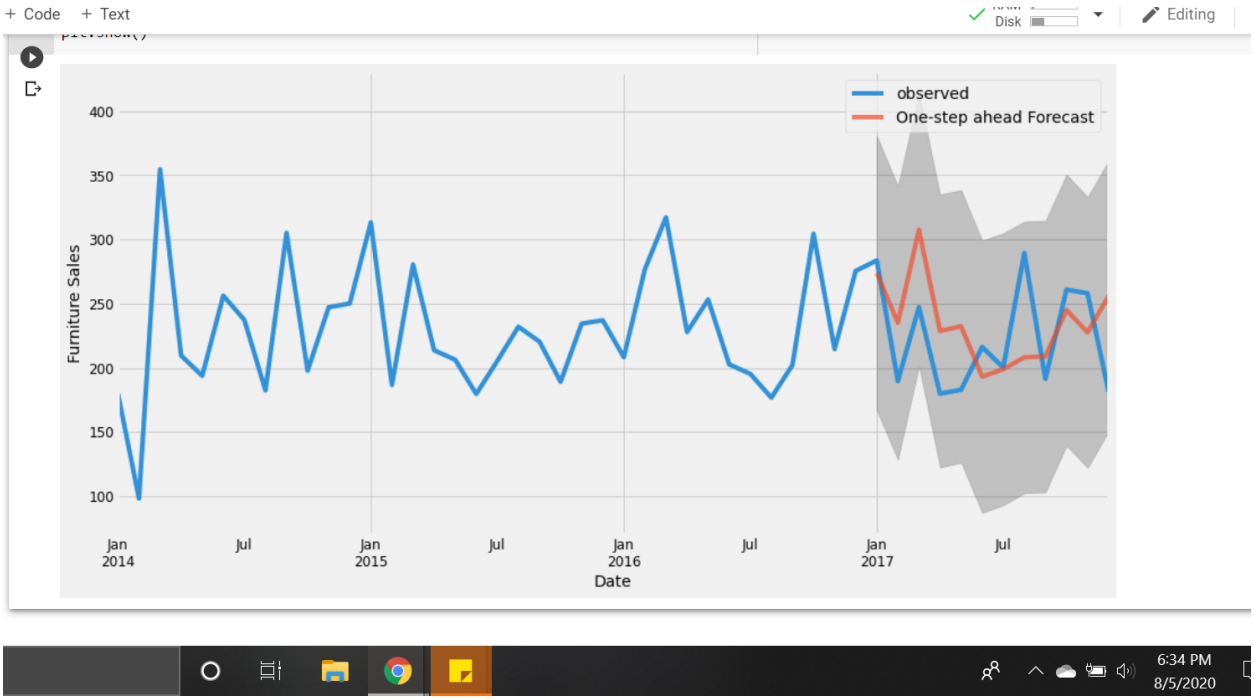


Figure 16 Visualizing Forecast

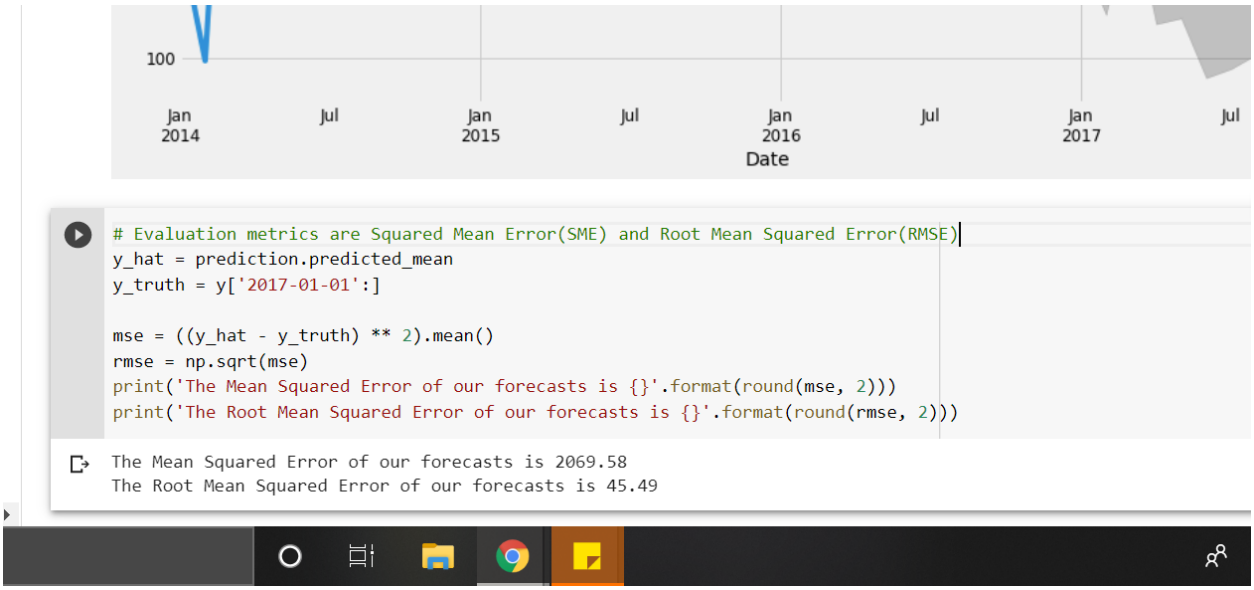


Figure 17 Error Analysis

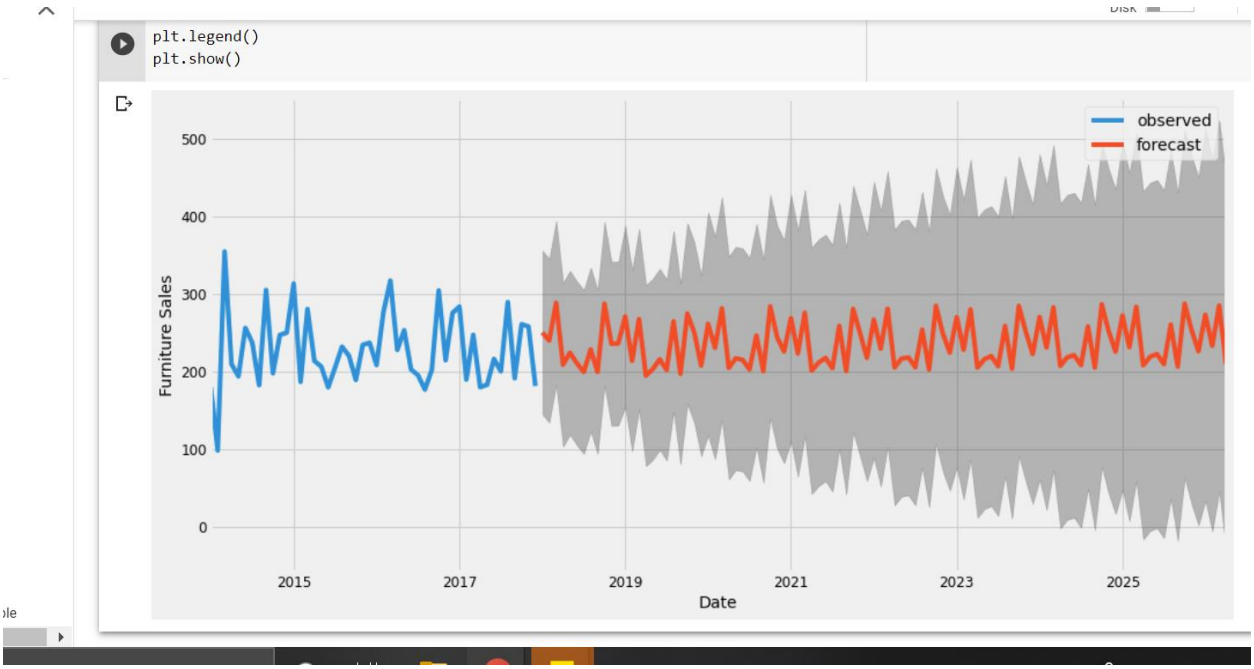


Figure 18 Producing and Visualizing forecast