

INF-Cointegration Series

July 15, 2021

1 Importing packages

```
[42]: #importing packages
import statsmodels.api as sm
from statsmodels.tsa.stattools import adfuller
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf
from sklearn import linear_model
import matplotlib.pyplot as plt
```

2 Reading Excel file saved in hard drive

```
[43]: #reading the file
df = pd.read_excel("C:\\Users\\rluck\\OneDrive\\fisher_update.xlsx")
df
```

```
[43]:
```

	DATE	P	R
0	1969-12-01	17.1	5.65
1	1970-03-01	17.3	7.15
2	1970-06-01	17.5	8.70
3	1970-09-01	17.6	6.35
4	1970-12-01	17.9	6.50
..
166	2011-06-01	178.3	4.99
167	2011-09-01	179.4	4.81
168	2011-12-01	179.4	4.51
169	2012-03-01	179.5	4.44
170	2012-06-01	180.4	3.49

[171 rows x 3 columns]

3 Calculating annual inflation from quarterly CPI

Quarterly CPI:

$$INF_{qtr} = 100 * \ln(P_t/P_{t-1})$$

Annual CPI

$$INF_{qtr} = 400 * \ln(P_t/P_{t-1})$$

```
[44]: #computing the inflation rate
df['INF'] = 400*np.log(df['P']/df['P'].shift(1))
df
```

```
[44]:
```

	DATE	P	R	INF
0	1969-12-01	17.1	5.65	NaN
1	1970-03-01	17.3	7.15	4.651215
2	1970-06-01	17.5	8.70	4.597752
3	1970-09-01	17.6	6.35	2.279208
4	1970-12-01	17.9	6.50	6.760724
..
166	2011-06-01	178.3	4.99	3.605658
167	2011-09-01	179.4	4.81	2.460170
168	2011-12-01	179.4	4.51	0.000000
169	2012-03-01	179.5	4.44	0.222903
170	2012-06-01	180.4	3.49	2.000560

[171 rows x 4 columns]

```
[45]: # Generating integrating differences series
df['DINF'] = df['INF'].diff(1).dropna()
df['DINF1'] = df['DINF'].shift(1).dropna()
df['DINF2'] = df['DINF'].shift(2).dropna()
df['DINF3'] = df['DINF'].shift(3).dropna()
df['DINF4'] = df['DINF'].shift(4).dropna()
df['DR'] = df['R'].diff(1).dropna()
df['DR1'] = df['DR'].shift(1).dropna()
df['DR2'] = df['DR'].shift(2).dropna()
df['DR3'] = df['DR'].shift(3).dropna()
df['DR4'] = df['DR'].shift(4).dropna()
df.head(60)
```

```
[45]:
```

	DATE	P	R	INF	DINF	DINF1	DINF2	\
0	1969-12-01	17.1	5.65	NaN	NaN	NaN	NaN	
1	1970-03-01	17.3	7.15	4.651215	NaN	NaN	NaN	
2	1970-06-01	17.5	8.70	4.597752	-0.053463	NaN	NaN	
3	1970-09-01	17.6	6.35	2.279208	-2.318543	-0.053463	NaN	
4	1970-12-01	17.9	6.50	6.760724	4.481516	-2.318543	-0.053463	
5	1971-03-01	18.1	8.00	4.444490	-2.316234	4.481516	-2.318543	
6	1971-06-01	18.4	8.15	6.575491	2.131000	-2.316234	4.481516	
7	1971-09-01	18.8	6.45	8.602482	2.026992	2.131000	-2.316234	
8	1971-12-01	19.2	5.90	8.421364	-0.181118	2.026992	2.131000	
9	1972-03-01	19.4	5.50	4.145115	-4.276249	-0.181118	2.026992	
10	1972-06-01	19.6	5.75	4.102600	-0.042515	-4.276249	-0.181118	

11	1972-09-01	19.9	4.50	6.076066	1.973466	-0.042515	-4.276249
12	1972-12-01	20.1	4.45	4.000033	-2.076033	1.973466	-0.042515
13	1973-03-01	20.5	5.45	7.882028	3.881995	-2.076033	1.973466
14	1973-06-01	21.2	6.40	13.430518	5.548490	3.881995	-2.076033
15	1973-09-01	21.9	9.25	12.994182	-0.436336	5.548490	3.881995
16	1973-12-01	22.7	9.25	14.351315	1.357133	-0.436336	5.548490
17	1974-03-01	23.3	10.10	10.435374	-3.915941	1.357133	-0.436336
18	1974-06-01	24.3	18.80	16.809196	6.373821	-3.915941	1.357133
19	1974-09-01	25.5	12.60	19.280841	2.471645	6.373821	-3.915941
20	1974-12-01	26.4	9.75	13.874223	-5.406618	2.471645	6.373821
21	1975-03-01	27.4	8.75	14.871601	0.997378	-5.406618	2.471645
22	1975-06-01	28.4	8.80	14.338453	-0.533149	0.997378	-5.406618
23	1975-09-01	28.6	8.10	2.807029	-11.531424	-0.533149	0.997378
24	1975-12-01	30.2	7.70	21.774083	18.967054	-11.531424	-0.533149
25	1976-03-01	31.0	8.40	10.458112	-11.315971	18.967054	-11.531424
26	1976-06-01	31.8	10.27	10.191634	-0.266478	-11.315971	18.967054
27	1976-09-01	32.6	9.31	9.938399	-0.253235	-0.266478	-11.315971
28	1976-12-01	34.5	9.44	22.658814	12.720415	-0.253235	-0.266478
29	1977-03-01	35.3	9.73	9.169456	-13.489358	12.720415	-0.253235
30	1977-06-01	36.0	10.93	8.963961	-0.205495	-13.489358	12.720415
31	1977-09-01	36.8	10.43	7.681992	-1.281969	-0.205495	-13.489358
32	1977-12-01	37.7	9.75	9.664900	1.982908	-1.281969	-0.205495
33	1978-03-01	38.2	10.04	5.270168	-4.394731	1.982908	-1.281969
34	1978-06-01	39.0	10.63	8.190152	3.020284	-4.394731	1.982908
35	1978-09-01	39.7	9.72	7.115817	-1.174636	3.020284	-4.394731
36	1978-12-01	40.6	8.76	8.966752	1.850935	-1.174636	3.020284
37	1979-03-01	41.3	9.16	6.837773	-2.128978	1.850935	-1.174636
38	1979-06-01	42.4	10.26	10.514345	3.676572	-2.128978	1.850935
39	1979-09-01	43.4	9.87	9.324432	-1.189913	3.676572	-2.128978
40	1979-12-01	44.7	10.12	11.805624	2.481193	-1.189913	3.676572
41	1980-03-01	45.7	11.47	8.849919	-2.955706	2.481193	-1.189913
42	1980-06-01	47.0	13.83	11.219722	2.369803	-2.955706	2.481193
43	1980-09-01	47.8	11.69	6.751215	-4.468506	2.369803	-2.955706
44	1980-12-01	48.8	12.45	8.281869	1.530654	-4.468506	2.369803
45	1981-03-01	50.0	14.63	9.717077	1.435208	1.530654	-4.468506
46	1981-06-01	51.1	15.58	8.704597	-1.012480	1.435208	1.530654
47	1981-09-01	52.1	15.35	7.752181	-0.952416	-1.012480	1.435208
48	1981-12-01	54.3	15.54	16.543711	8.791531	-0.952416	-1.012480
49	1982-03-01	55.3	18.89	7.299473	-9.244239	8.791531	-0.952416
50	1982-06-01	56.6	18.57	9.294431	1.994958	-9.244239	8.791531
51	1982-09-01	58.6	15.46	13.890285	4.595854	1.994958	-9.244239
52	1982-12-01	60.3	12.13	11.438963	-2.451322	4.595854	1.994958
53	1983-03-01	61.6	15.26	8.531907	-2.907056	-2.451322	4.595854
54	1983-06-01	62.9	14.24	8.353717	-0.178189	-2.907056	-2.451322
55	1983-09-01	64.0	11.06	6.934768	-1.418949	-0.178189	-2.907056
56	1983-12-01	65.5	8.89	9.266824	2.332056	-1.418949	-0.178189
57	1984-03-01	65.2	13.77	-1.836269	-11.103093	2.332056	-1.418949

58	1984-06-01	65.4	12.81	1.225116	3.061385	-11.103093	2.332056
59	1984-09-01	66.2	10.53	4.863282	3.638166	3.061385	-11.103093

	DINF3	DINF4	DR	DR1	DR2	DR3	DR4
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	1.50	NaN	NaN	NaN	NaN
2	NaN	NaN	1.55	1.50	NaN	NaN	NaN
3	NaN	NaN	-2.35	1.55	1.50	NaN	NaN
4	NaN	NaN	0.15	-2.35	1.55	1.50	NaN
5	-0.053463	NaN	1.50	0.15	-2.35	1.55	1.50
6	-2.318543	-0.053463	0.15	1.50	0.15	-2.35	1.55
7	4.481516	-2.318543	-1.70	0.15	1.50	0.15	-2.35
8	-2.316234	4.481516	-0.55	-1.70	0.15	1.50	0.15
9	2.131000	-2.316234	-0.40	-0.55	-1.70	0.15	1.50
10	2.026992	2.131000	0.25	-0.40	-0.55	-1.70	0.15
11	-0.181118	2.026992	-1.25	0.25	-0.40	-0.55	-1.70
12	-4.276249	-0.181118	-0.05	-1.25	0.25	-0.40	-0.55
13	-0.042515	-4.276249	1.00	-0.05	-1.25	0.25	-0.40
14	1.973466	-0.042515	0.95	1.00	-0.05	-1.25	0.25
15	-2.076033	1.973466	2.85	0.95	1.00	0.05	-1.25
16	3.881995	-2.076033	0.00	2.85	0.95	1.00	-0.05
17	5.548490	3.881995	0.85	0.00	2.85	0.95	1.00
18	-0.436336	5.548490	8.70	0.85	0.00	2.85	0.95
19	1.357133	-0.436336	-6.20	8.70	0.85	0.00	2.85
20	-3.915941	1.357133	-2.85	-6.20	8.70	0.85	0.00
21	6.373821	-3.915941	-1.00	-2.85	-6.20	8.70	0.85
22	2.471645	6.373821	0.05	-1.00	-2.85	-6.20	8.70
23	-5.406618	2.471645	-0.70	0.05	-1.00	-2.85	-6.20
24	0.997378	-5.406618	-0.40	-0.70	0.05	-1.00	-2.85
25	-0.533149	0.997378	0.70	-0.40	-0.70	0.05	-1.00
26	-11.531424	-0.533149	1.87	0.70	-0.40	-0.70	0.05
27	18.967054	-11.531424	-0.96	1.87	0.70	-0.40	-0.70
28	-11.315971	18.967054	0.13	-0.96	1.87	0.70	-0.40
29	-0.266478	-11.315971	0.29	0.13	-0.96	1.87	0.70
30	-0.253235	-0.266478	1.22	0.29	0.13	-0.96	1.87
31	12.720415	-0.253235	-0.52	1.22	0.29	0.13	-0.96
32	-13.489358	12.720415	-0.68	-0.52	1.22	0.29	0.13
33	-0.205495	-13.489358	0.29	-0.68	-0.52	1.22	0.29
34	-1.281969	-0.205495	0.59	0.29	-0.68	-0.52	1.22
35	1.982908	-1.281969	-0.91	0.59	0.29	-0.68	-0.52
36	-4.394731	1.982908	-0.96	-0.91	0.59	0.29	-0.68
37	3.020284	-4.394731	0.40	-0.96	-0.91	0.59	0.29
38	-1.174636	3.020284	1.10	0.40	-0.96	-0.91	0.59
39	1.850935	-1.174636	-0.39	1.10	0.40	-0.96	-0.91
40	-2.128978	1.850935	0.25	-0.39	1.10	0.40	-0.96
41	3.676572	-2.128978	1.35	0.25	-0.39	1.10	0.40
42	-1.189913	3.676572	2.36	1.35	0.25	-0.39	1.10

Assignment Project Exam Help

<https://tutors.com>

WeChat: cstutors

43	2.481193	-1.189913	-2.14	2.36	1.35	0.25	-0.39
44	-2.955706	2.481193	0.76	-2.14	2.36	1.35	0.25
45	2.369803	-2.955706	2.18	0.76	-2.14	2.36	1.35
46	-4.468506	2.369803	0.95	2.18	0.76	-2.14	2.36
47	1.530654	-4.468506	-0.23	0.95	2.18	0.76	-2.14
48	1.435208	1.530654	0.19	-0.23	0.95	2.18	0.76
49	-1.012480	1.435208	3.35	0.19	-0.23	0.95	2.18
50	-0.952416	-1.012480	-0.32	3.35	0.19	-0.23	0.95
51	8.791531	-0.952416	-3.11	-0.32	3.35	0.19	-0.23
52	-9.244239	8.791531	-3.33	-3.11	-0.32	3.35	0.19
53	1.994958	-9.244239	3.13	-3.33	-3.11	-0.32	3.35
54	4.595854	1.994958	-1.02	3.13	-3.33	-3.11	-0.32
55	-2.451322	4.595854	-3.18	-1.02	3.13	-3.33	-3.11
56	-2.907056	-2.451322	-2.17	-3.18	-1.02	3.13	-3.33
57	-0.178189	-2.907056	4.88	-2.17	-3.18	-1.02	3.13
58	-1.418949	-0.178189	-0.96	4.88	-2.17	-3.18	-1.02
59	2.332056	-1.418949	-2.28	-0.96	4.88	-2.17	-3.18

4 Selecting sample data from row 57:Qtr 1 1984 to row 170: Qtr 2012

```
[46]: #Selecting the sample from
dta =df.iloc[57:170].dropna()
dta
```

```
[46]:
```

	DATE	P	R	INF	DINF	DINF1	DINF2 \
57	1984-03-01	65.2	13.77	1.836269	-11.103093	2.332056	-1.418949
58	1984-06-01	65.4	12.81	1.225116	3.061385	-11.103093	2.332056
59	1984-09-01	66.2	10.53	4.863282	3.638166	3.061385	-11.103093
60	1984-12-01	67.2	12.34	5.997114	1.133832	3.638166	3.061385
61	1985-03-01	68.1	15.29	5.321586	-0.675528	1.133832	3.638166
..
165	2011-03-01	176.7	4.92	6.159232	4.546791	-1.166956	0.214539
166	2011-06-01	178.3	4.99	3.605658	-2.553574	4.546791	-1.166956
167	2011-09-01	179.4	4.81	2.460170	-1.145488	-2.553574	4.546791
168	2011-12-01	179.4	4.51	0.000000	-2.460170	-1.145488	-2.553574
169	2012-03-01	179.5	4.44	0.222903	0.222903	-2.460170	-1.145488

	DINF3	DINF4	DR	DR1	DR2	DR3	DR4
57	-0.178189	-2.907056	4.88	-2.17	-3.18	-1.02	3.13
58	-1.418949	-0.178189	-0.96	4.88	-2.17	-3.18	-1.02
59	2.332056	-1.418949	-2.28	-0.96	4.88	-2.17	-3.18
60	-11.103093	2.332056	1.81	-2.28	-0.96	4.88	-2.17
61	3.061385	-11.103093	2.95	1.81	-2.28	-0.96	4.88
..
165	-0.959393	1.394699	-0.11	0.21	-0.07	0.56	0.20

```

166    0.214539  -0.959393  0.07 -0.11  0.21 -0.07  0.56
167   -1.166956   0.214539 -0.18  0.07 -0.11  0.21 -0.07
168    4.546791  -1.166956 -0.30 -0.18  0.07 -0.11  0.21
169   -2.553574   4.546791 -0.07 -0.30 -0.18  0.07 -0.11

```

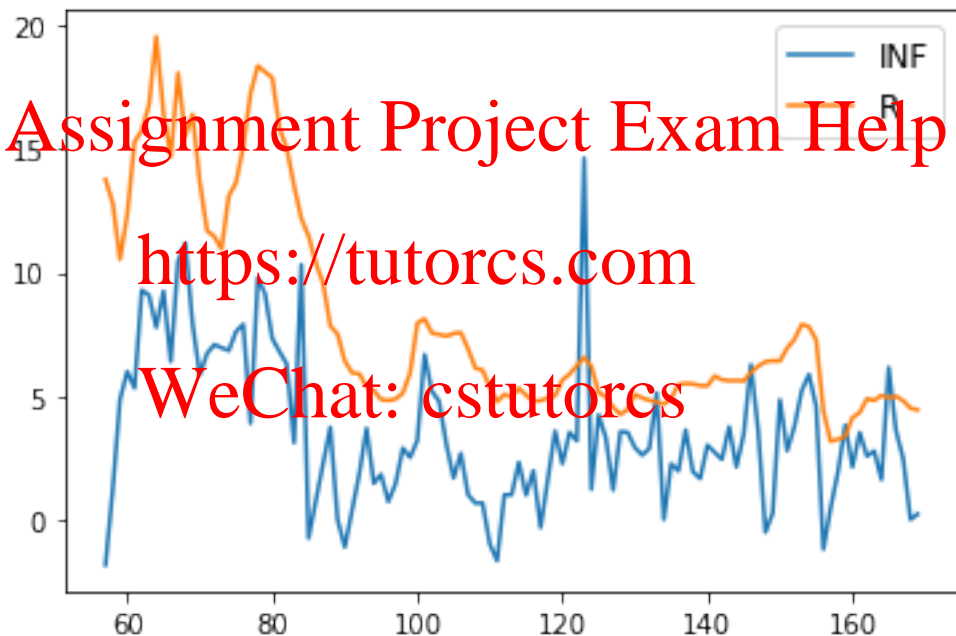
[113 rows x 14 columns]

5 Plotting the time series: Inflation

```

[47]: #plotting the series
plt.plot(dta['INF'],label='INF')
plt.plot(dta['R'],label='R')
plt.legend(loc='best', fontsize='large')
plt.show()

```



```

[48]: #Regressing Interest Rate (Y=R) against the Inflation rate (X= INF)
reg = linear_model.LinearRegression()
X =dta[['INF']].dropna()
y =dta['R'].dropna()
reg.fit(X,y)
predictions =reg.predict(X)

```

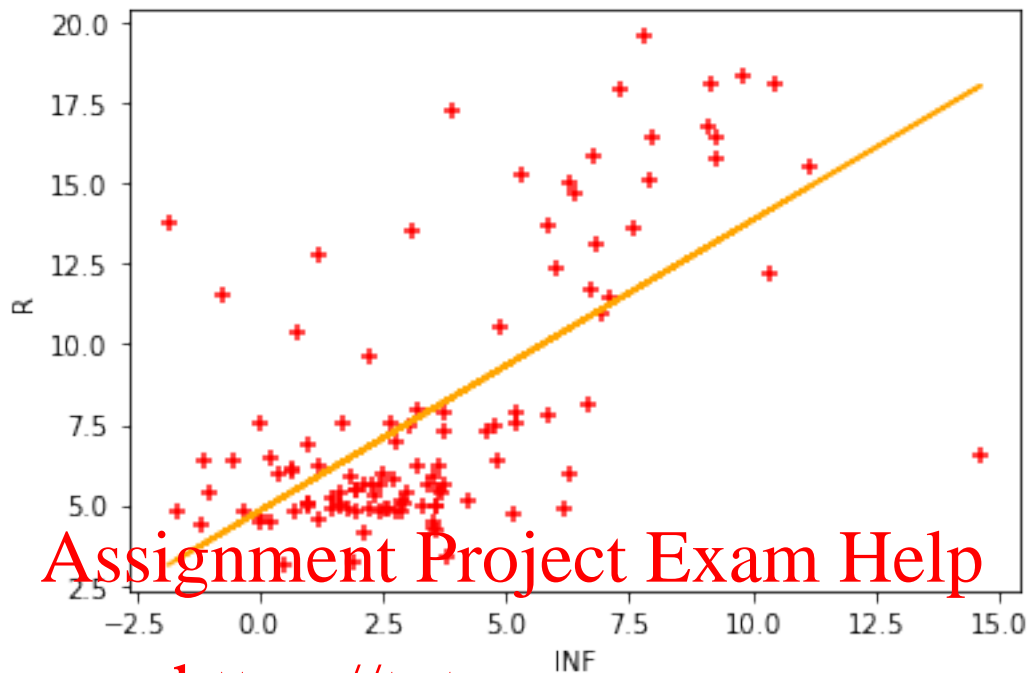
```

[49]: plt.xlabel('INF')
plt.ylabel('R')
plt.scatter(dta.INF,dta.R,color='red',marker='+')

```

```
plt.plot(dta.INF,reg.predict(dta[['INF']]), color='orange')
```

[49]: [<matplotlib.lines.Line2D at 0x1dc1046ceb0>]



Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

```
[84]: #model with intercept
X = dta.INF
y = dta.R
X= sm.add_constant(X)
model= sm.OLS(y,X).fit()
predictions = model.predict(X)
G= (model.summary())
print(G)
```

OLS Regression Results

```
=====
Dep. Variable:          R    R-squared:                0.460
Model:                  OLS    Adj. R-squared:            0.455
Method:                 Least Squares    F-statistic:            93.72
Date:                  Thu, 15 Jul 2021    Prob (F-statistic):      2.11e-16
Time:                  18:10:35    Log-Likelihood:          -286.77
No. Observations:      112    AIC:                     577.5
Df Residuals:          110    BIC:                     583.0
Df Model:               1
Covariance Type:       nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
-----	-----	-----	-----	-----	-----	-----
const	4.5105	0.467	9.659	0.000	3.585	5.436
INF	0.9610	0.099	9.681	0.000	0.764	1.158
=====	=====	=====	=====	=====	=====	=====
Omnibus:		7.745	Durbin-Watson:			0.848
Prob(Omnibus):		0.021	Jarque-Bera (JB):			12.125
Skew:		0.256	Prob(JB):			0.00233
Kurtosis:		4.529	Cond. No.			7.56
=====	=====	=====	=====	=====	=====	=====

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

6 Correlogram of Residuals: ACF and PACF

$$\epsilon_t = y_t - (\beta * X_t + \alpha)$$

```
[51]: dtr = model.resid
      dtr
```

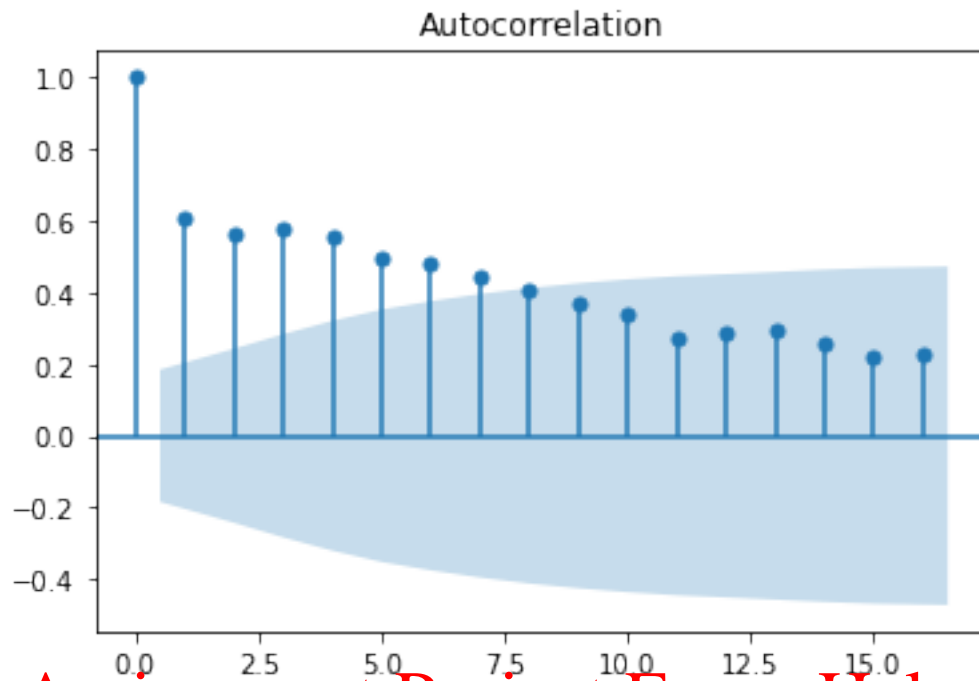
```
[51]: 57      10.617794
      58       6.890682
      59       1.322232
      60       2.107389
      61       5.667982
      ...
      165      -5.459146
      166      -3.081032
      167      -2.225653
      168      -0.301965
      169      -0.573442
      Length: 113, dtype: float64
```

```
[52]: #running ACF and PACF
      sm.graphics.tsa.plot_acf(dtr.values.squeeze(),lags=16)
      sm.graphics.tsa.plot_pacf(dtr.values.squeeze(),lags=16)
      plt.show()
```

Assignment Project Exam Help

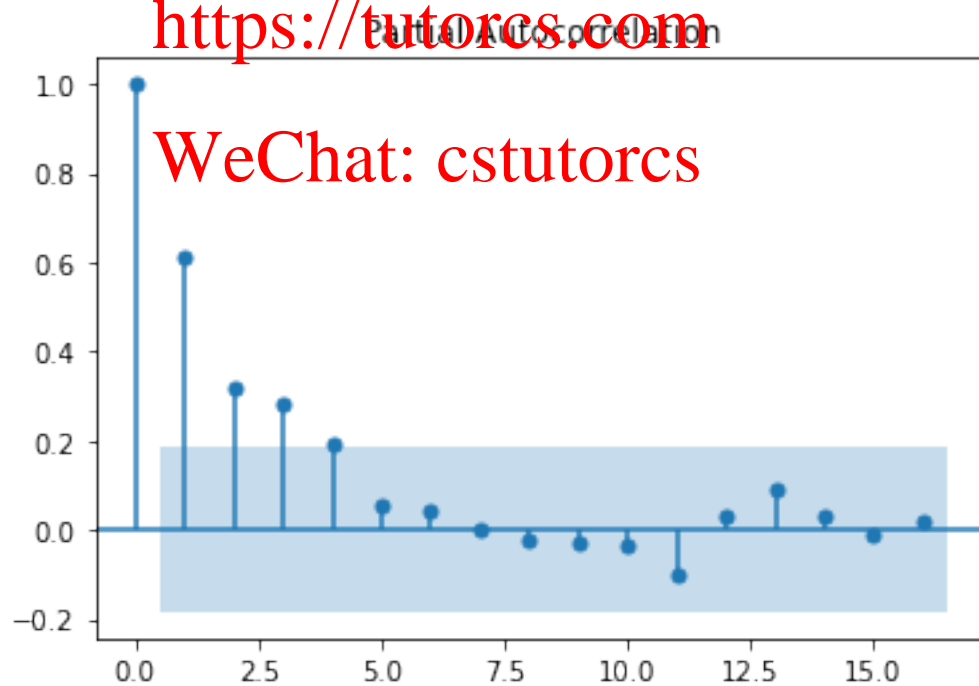
<https://tutorcs.com>

WeChat: cstutorcs



Assignment Project Exam Help

<https://tutorcs.com>



WeChat: cstutorcs

```
[53]: # Generating the Q tables
import numpy as np
r,q,p = sm.tsa.acf(dtr.values.squeeze(), qstat=True)
data = np.c_[range(1,41), r[1:], q, p]
table = pd.DataFrame(data, columns=['lag', "AC", "Q", "Prob(>Q)"])
print (table.set_index('lag'))
```

lag	AC	Q	Prob(>Q)
1.0	0.604164	42.351394	7.626197e-11
2.0	0.561012	79.197947	6.344308e-18
3.0	0.574160	118.142798	1.938169e-25
4.0	0.558060	155.271704	1.509379e-32
5.0	0.498832	185.212371	4.120822e-38
6.0	0.477251	212.874547	3.437139e-43
7.0	0.444019	237.044412	1.579656e-47
8.0	0.404992	257.343619	4.775363e-51
9.0	0.370545	274.499942	6.605085e-54
10.0	0.337784	288.895138	3.451093e-56
11.0	0.275316	298.587198	1.733371e-57
12.0	0.286480	309.146676	5.627670e-59
13.0	0.296186	320.546711	1.193303e-60
14.0	0.259174	329.363744	8.672219e-62
15.0	0.223192	335.953290	1.795163e-62
16.0	0.226696	342.854116	3.193704e-63
17.0	0.255196	351.669754	2.217145e-64
18.0	0.223284	358.189525	3.951410e-65
19.0	0.298762	370.829582	4.60293e-67
20.0	0.321913	385.309124	2.265168e-69
21.0	0.282732	396.600261	4.685822e-71
22.0	0.298276	409.305178	4.926778e-73
23.0	0.234833	417.267758	4.917945e-74
24.0	0.236777	425.453665	4.382705e-75
25.0	0.229401	433.224808	4.722116e-76
26.0	0.174816	437.789603	2.301703e-76
27.0	0.162803	441.794612	1.445240e-76
28.0	0.223076	449.402502	1.643747e-77
29.0	0.149866	452.877109	1.299917e-77
30.0	0.154104	456.595241	9.069163e-78
31.0	0.098993	458.148250	1.727635e-77
32.0	0.100837	459.779524	3.130430e-77
33.0	0.027877	459.905761	1.130134e-76
34.0	0.021439	459.981366	4.113627e-76
35.0	0.014447	460.016136	1.503308e-75
36.0	-0.023361	460.108237	5.272135e-75
37.0	-0.053373	460.595316	1.518910e-74
38.0	-0.062695	461.276367	3.950474e-74
39.0	-0.003679	461.278743	1.387193e-73

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

```
40.0 -0.076177 462.311755 2.995337e-73
```

```
C:\Users\rluck\anaconda3\lib\site-packages\statsmodels\tsa\stattools.py:657:
FutureWarning: The default number of lags is changing from 40 to min(int(10 *
np.log10(nobs)), nobs - 1) after 0.12 is released. Set the number of lags to an
integer to silence this warning.
```

```
warnings.warn(
C:\Users\rluck\anaconda3\lib\site-packages\statsmodels\tsa\stattools.py:667:
FutureWarning: fft=True will become the default after the release of the 0.12
release of statsmodels. To suppress this warning, explicitly set fft=False.
warnings.warn(
```

```
[ ]:
```

7 ADF test of stationarity and unit root

```
[54]: residuals = model.resid
      residuals
```

```
[54]: 57      10.617794
      58      6.890682
      59      1.322232
      60      2.107389
      61      5.667982
      ...
      165     -5.459146
      166     -3.081032
      167     -2.225653
      168     -0.301965
      169     -0.573442
      Length: 113, dtype: float64
```

```
[55]: dtr
```

```
[55]: 57      10.617794
      58      6.890682
      59      1.322232
      60      2.107389
      61      5.667982
      ...
      165     -5.459146
      166     -3.081032
      167     -2.225653
      168     -0.301965
      169     -0.573442
      Length: 113, dtype: float64
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

```
[56]: #ADF Tests
from arch.unitroot import ADF
ADF(residuals,trend="n",lags=1, max_lags=12, method='BIC')
```

```
[56]: <class 'arch.unitroot.unitroot.ADF'>
      """
      Augmented Dickey-Fuller Results
      =====
      Test Statistic          -3.854
      P-value                 0.000
      Lags                    1
      -----

      Trend: No Trend
      Critical Values: -2.59 (1%), -1.94 (5%), -1.61 (10%)
      Null Hypothesis: The process contains a unit root.
      Alternative Hypothesis: The process is weakly stationary.
      """
```

8 Engle Granger Cointegration Test

```
[57]: from arch.unitroot import engle_granger
engle_test = engle_granger(y,X,trend="n",lags=1)
engle_test
```

```
[57]: Engle-Granger Cointegration Test
Statistic: -3.853802630067636
P-value: 0.010421142946669465
Null: No Cointegration, Alternative: Cointegration
ADF Lag length: 1
Trend: c
Estimated Root (+1): 0.7108844276270996
Distribution Order: 2
ID: 0x1dc104bdbe0
```

9 DR Regressed against DR & DINF with lags up to 4

$$\epsilon_t = y_t - (\beta x_t + \alpha)$$

```
[58]: dta['resid'] = y - model.predict(X)
      #Residual series by lag 1
      dta['resid_1'] = dta['resid'].shift(1)
      dta = dta.dropna(subset=['resid_1'])
      dta
```

```
[58]:
```

	DATE	P	R	INF	DINF	DINF1	DINF2	\
58	1984-06-01	65.4	12.81	1.225116	3.061385	-11.103093	2.332056	
59	1984-09-01	66.2	10.53	4.863282	3.638166	3.061385	-11.103093	
60	1984-12-01	67.2	12.34	5.997114	1.133832	3.638166	3.061385	
61	1985-03-01	68.1	15.29	5.321586	-0.675528	1.133832	3.638166	
62	1985-06-01	69.7	15.75	9.289242	3.967656	-0.675528	1.133832	
..	
165	2011-03-01	176.7	4.92	6.159232	4.546791	-1.166956	0.214539	
166	2011-06-01	178.3	4.99	3.605658	-2.553574	4.546791	-1.166956	
167	2011-09-01	179.4	4.81	2.460170	-1.145488	-2.553574	4.546791	
168	2011-12-01	179.4	4.51	0.000000	-2.460170	-1.145488	-2.553574	
169	2012-03-01	179.5	4.44	0.222903	0.222903	-2.460170	-1.145488	

	DINF3	DINF4	DR	DR1	DR2	DR3	DR4	resid	resid_1
58	-1.418949	-0.178189	-0.96	4.88	-2.17	-3.18	-1.02	6.890682	10.617794
59	2.332056	-1.418949	-2.28	-0.96	4.88	-2.17	-3.18	1.322232	6.890682
60	-11.103093	2.332056	1.81	-2.28	-0.96	4.88	-2.17	2.107389	1.322232
61	3.061385	-11.103093	2.95	1.81	-2.28	-0.96	4.88	5.667982	2.107389
62	3.638166	3.061385	0.46	2.95	1.81	-2.28	-0.96	2.541714	5.667982
..
165	-0.959393	1.394699	-0.11	0.21	-0.07	0.56	0.20	-5.459146	-1.239412
166	0.214539	-0.959393	0.07	-0.11	0.21	-0.07	0.56	-3.081032	-5.459146
167	-1.166956	0.214539	-0.18	0.07	-0.11	0.21	-0.07	-2.225653	-3.081032
168	4.546791	-1.166956	0.30	-0.18	0.07	-0.11	0.21	-0.301965	-2.225653
169	-2.553574	4.546791	-0.07	-0.30	-0.18	0.07	-0.11	-0.573442	-0.301965

[112 rows x 16 columns]

Multiple Regression

```
[59]: #model with intercept
x_1 = dta[['resid_1', 'DR1', 'DR2', 'DR3', 'DR4', 'DINF1', 'DINF2', 'DINF3', 'DINF4']]
y_1 = dta['DR']
x_1 = sm.add_constant(x_1)
model_1 = sm.OLS(y_1, x_1).fit()
predictions = model_1.predict(x_1)
h = (model_1.summary())
print(h)
```

OLS Regression Results

```
=====
Dep. Variable:          DR      R-squared:                0.189
Model:                  OLS      Adj. R-squared:           0.117
Method:                 Least Squares      F-statistic:           2.633
Date:                   Thu, 15 Jul 2021    Prob (F-statistic):      0.00881
Time:                   17:24:17           Log-Likelihood:         -151.97
No. Observations:       112             AIC:                   323.9
Df Residuals:           102             BIC:                   351.1
```

Df Model: 9
Covariance Type: nonrobust

	coef	std err	t	P> t	[0.025	0.975]
const	-0.0631	0.094	-0.674	0.502	-0.249	0.123
resid_1	-0.0579	0.035	-1.631	0.106	-0.128	0.012
DR1	0.1464	0.091	1.613	0.110	-0.034	0.327
DR2	0.0222	0.091	0.243	0.808	-0.159	0.203
DR3	0.2249	0.085	2.653	0.009	0.057	0.393
DR4	-0.0686	0.082	-0.833	0.407	-0.232	0.095
DINF1	0.0264	0.047	0.558	0.578	-0.067	0.120
DINF2	0.0365	0.051	0.717	0.475	-0.064	0.137
DINF3	-0.0415	0.049	-0.843	0.401	-0.139	0.056
DINF4	-0.0570	0.040	-1.425	0.157	-0.136	0.022

Omnibus: 15.294 Durbin-Watson: 2.004
Prob(Omnibus): 0.000 Jarque-Bera (JB): 59.082
Skew: 0.087 Prob(JB): 1.48e-13
Kurtosis: 6.554 Cond. No. 5.07

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

10 DINF Regressed against DR & DINF with lags up to 4

```
[60]: #model with intercept
x_1 = dta[['resid_1', 'DR1', 'DR2', 'DR3', 'DR4', 'DINF1', 'DINF2', 'DINF3', 'DINF4']]
y_2 = dta['DINF']
x_1 = sm.add_constant(x_1)
model_2 = sm.OLS(y_2, x_1).fit()
predictions = model_2.predict(x_1)
I = (model_2.summary())
print(I)
```

OLS Regression Results

Dep. Variable:	DINF	R-squared:	0.367
Model:	OLS	Adj. R-squared:	0.311
Method:	Least Squares	F-statistic:	6.574
Date:	Thu, 15 Jul 2021	Prob (F-statistic):	2.36e-07
Time:	17:24:17	Log-Likelihood:	-252.89
No. Observations:	112	AIC:	525.8
Df Residuals:	102	BIC:	553.0
Df Model:	9		

Covariance Type: nonrobust

	coef	std err	t	P> t	[0.025	0.975]
const	-0.0106	0.231	-0.046	0.964	-0.468	0.447
resid_1	0.0622	0.087	0.712	0.478	-0.111	0.235
DR1	0.5445	0.224	2.436	0.017	0.101	0.988
DR2	0.4540	0.225	2.021	0.046	0.008	0.900
DR3	0.1029	0.209	0.493	0.623	-0.311	0.517
DR4	0.0865	0.203	0.427	0.670	-0.316	0.489
DINF1	-0.6278	0.116	-5.392	0.000	-0.859	-0.397
DINF2	-0.4934	0.125	-3.938	0.000	-0.742	-0.245
DINF3	-0.3980	0.121	-3.279	0.001	-0.639	-0.157
DINF4	-0.1518	0.098	-1.542	0.126	-0.347	0.043
Omnibus:	25.323	Durbin-Watson:	2.106			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	113.849			
Skew:	0.558	Prob(JB):	1.90e-25			
Kurtosis:	7.811	Cond. No.	5.07			

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Wald Tests

```
[82]: #Wald test on model_1: Restricting coefficient 7 to 10 (if p > 0.05,
      ↪restrictions cannot be rejected)
R = np.eye(len(model_1.params))[5:10]
model_1.wald_test(R)
```

```
[82]: <class 'statsmodels.stats.contrast.ContrastResults'>
      <F test: F=array([[1.29567193]]), p=0.2716573127484845, df_denom=102, df_num=5>
```

```
[70]: #Wald test on model_1: Restricting coefficient 5 and 6 (if p > 0.05,
      ↪Restrictions cannot be rejected)
R = np.eye(len(model_2.params))[4:6]
model_2.wald_test(R)
```

```
[70]: <class 'statsmodels.stats.contrast.ContrastResults'>
      <F test: F=array([[0.2258903]]), p=0.7982036497279359, df_denom=102, df_num=2>
```

11 DR Regressed against DR with lags up to 4

```
[63]: x_3 = dta[['resid_1', 'DR1', 'DR2', 'DR3', 'DR4']]
y_3 = dta['DR']
x_3= sm.add_constant(x_3)
model_3 = sm.OLS(y_3,x_3).fit()
predictions = model_3.predict(x_3)
J= (model_3.summary())
print(J)
```

OLS Regression Results

```
=====
Dep. Variable:          DR      R-squared:                0.144
Model:                  OLS      Adj. R-squared:           0.104
Method:                 Least Squares      F-statistic:        3.578
Date:                   Thu, 15 Jul 2021      Prob (F-statistic):    0.00498
Time:                   17:24:17      Log-Likelihood:       -154.93
No. Observations:       112      AIC:                  321.9
Df Residuals:           106      BIC:                  338.2
Df Model:                5
Covariance Type:        nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
const	-0.0653	0.094	-0.693	0.490	-0.252	0.122
resid_1	-0.0732	0.029	-2.537	0.013	-0.130	-0.016
DR1	0.1638	0.085	1.926	0.057	-0.005	0.333
DR2	-0.0138	0.081	-0.171	0.866	-0.177	0.145
DR3	0.2324	0.079	2.926	0.004	0.075	0.390
DR4	-0.0773	0.081	-0.958	0.340	-0.237	0.083

```
=====
Omnibus:                19.421      Durbin-Watson:           1.978
Prob(Omnibus):           0.000      Jarque-Bera (JB):        88.039
Skew:                    0.279      Prob(JB):                7.63e-20
Kurtosis:                7.307      Cond. No.:               3.37
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

12 DINF Regressed against DINF with lags up to 4

```
[64]: x_4 = dta[['resid_1', 'DR1', 'DR2', 'DINF1', 'DINF2', 'DINF3', 'DINF4']]
y_4 = dta['DINF']
x_4= sm.add_constant(x_4)
model = sm.OLS(y_4,x_4).fit()
```



```

predictions = model.predict(x_4)
K= (model.summary())
print(K)

```

OLS Regression Results

```

=====
Dep. Variable:          DINF      R-squared:                0.364
Model:                  OLS      Adj. R-squared:            0.322
Method:                 Least Squares      F-statistic:          8.515
Date:                  Thu, 15 Jul 2021      Prob (F-statistic):    3.26e-08
Time:                  17:24:17      Log-Likelihood:        -253.14
No. Observations:      112      AIC:                  522.3
Df Residuals:          104      BIC:                  544.0
Df Model:               7
Covariance Type:       nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	-0.0244	0.228	-0.107	0.915	-0.477	0.428
resid_1	0.0714	0.096	0.834	0.408	-0.108	0.241
DR1	0.5335	0.216	2.465	0.015	0.104	0.963
DR2	0.4309	0.220	1.960	0.053	-0.005	0.867
DINF1	-0.6069	0.111	-5.487	0.000	-0.826	-0.388
DINF2	-0.4627	0.115	-4.066	0.000	-0.691	-0.234
DINF3	-0.3719	0.114	-3.259	0.002	-0.598	-0.146
DINF4	-0.1343	0.094	-1.433	0.155	-0.320	0.052
Omnibus:	25.561	Durbin-Watson:				2.122
Prob(Omnibus):	0.000	Jarque-Bera (JB):				122.519
Skew:	0.537	Prob(JB):				2.48e-27
Kurtosis:	8.010	Cond. No.				4.64

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```

[83]: #Wald test om model_3: Restricting coefficient 4 (if p >0,05, restrictions_
      ↪ cannot be rejected )
R = np.eye(len(model_3.params))[3:4]
model_3.wald_test(R)

```

```

[83]: <class 'statsmodels.stats.contrast.ContrastResults'>
      <F test: F=array([[0.03790856]]), p=0.8459995477076325, df_denom=106, df_num=1>

```

```

[85]: #Model 5
x_5 = dta[['resid_1', 'DR1', 'DR3', 'DR4']]
y_5 = dta['DR']

```

```
x_5= sm.add_constant(x_5)
model_5 = sm.OLS(y_5,x_5).fit()
predictions = model_5.predict(x_5)
k= (model_5.summary())
print(k)
```

OLS Regression Results

```
=====
Dep. Variable:          DR      R-squared:                0.144
Model:                  OLS      Adj. R-squared:           0.112
Method:                 Least Squares      F-statistic:         4.503
Date:                  Thu, 15 Jul 2021      Prob (F-statistic):      0.00211
Time:                  18:11:56      Log-Likelihood:         -154.95
No. Observations:      112      AIC:                   319.9
Df Residuals:          107      BIC:                   333.5
Df Model:               4
Covariance Type:       nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
const	-0.0644	0.094	-0.687	0.494	-0.250	0.122
resid_1	-0.0733	0.029	-2.553	0.012	-0.130	-0.016
DR1	0.1631	0.085	1.928	0.057	-0.005	0.331
DR3	0.2313	0.079	2.963	0.004	0.075	0.388
DR4	-0.0758	0.080	-0.948	0.345	-0.234	0.083

```
=====
Omnibus:              19.887      Durbin-Watson:           1.976
Prob(Omnibus):        0.000      Jarque-Bera (JB):        90.139
Skew:                 0.307      Prob(JB):                2.67e-20
Kurtosis:              7.352      Cond. No.:               3.35
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
[86]: #Wald test om model_5: Restricting coefficient 5 (if p >0,05, restrictions_
      ↪ cannot be rejected )
R = np.eye(len(model_5.params))[4:5]
model_5.wald_test(R)
```

```
[86]: <class 'statsmodels.stats.contrast.ContrastResults'>
      <F test: F=array([[0.89881769]]), p=0.345235217373811, df_denom=107, df_num=1>
```

```
[87]: # Model 6
x_6 = dta[['resid_1', 'DR1', 'DR3']]
y_6 = dta['DR']
x_6= sm.add_constant(x_6)
```

```

model_6 = sm.OLS(y_6,x_6).fit()
predictions = model_6.predict(x_6)
k= (model_6.summary())
print(k)

```

OLS Regression Results

```

=====
Dep. Variable:          DR      R-squared:                0.137
Model:                  OLS      Adj. R-squared:           0.113
Method:                 Least Squares      F-statistic:         5.710
Date:                   Thu, 15 Jul 2021      Prob (F-statistic):      0.00115
Time:                   18:16:08      Log-Likelihood:         -155.42
No. Observations:      112      AIC:                   318.8
Df Residuals:          108      BIC:                   329.7
Df Model:               3
Covariance Type:        nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	0.0587	0.094	0.628	0.532	-0.214	0.127
resid_1	-0.0749	0.029	-2.613	0.010	-0.132	-0.018
DR1	0.1467	0.083	1.772	0.079	-0.017	0.311
DR3	0.2235	0.078	2.851	0.005	0.068	0.379
Omnibus:	16.578	Durbin-Watson:	1.925			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	66.108			
Skew:	0.188	Prob(JB):	4.41e-15			
Kurtosis:	6.745	Cond. No.	3.35			

=====

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[]: