HW2

Charles Dotson

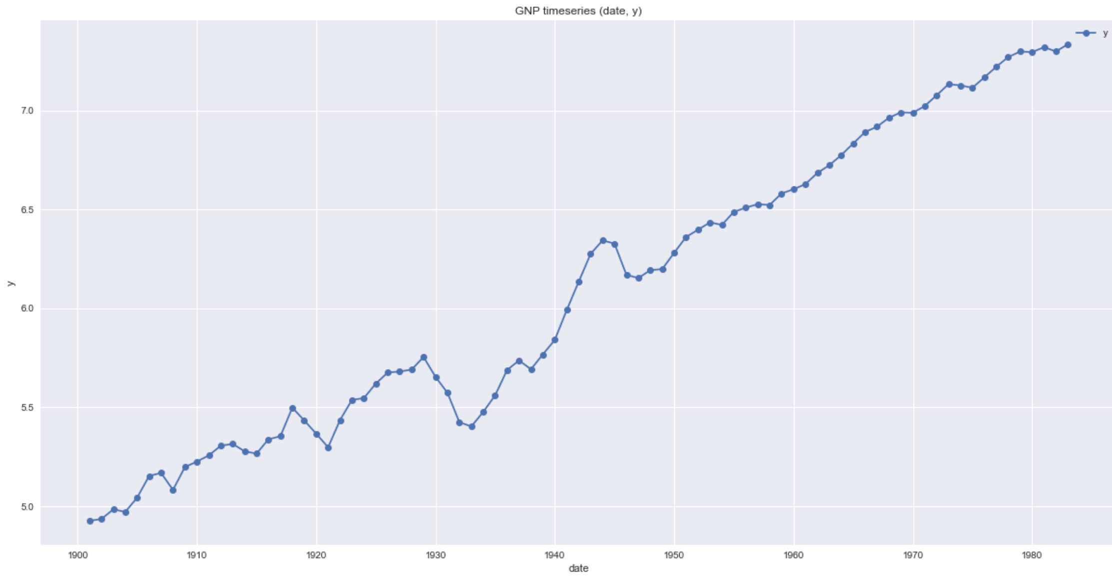
September 12, 2022

Contents

[429]:
```python
'''Importing Packages'''
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
from IPython.display import Markdown as md
import statsmodels.tsa.stattools as ts
import statsmodels.api as sm
import datetime
from loess import loess_1d
from statsmodels.graphics.tsaplots import plot_acf
from openpyxl import Workbook, load_workbook
from sklearn import linear_model
from statsmodels.tsa.ar_model import AutoReg, ar_select_order
from scipy.linalg import toeplitz
%matplotlib inline
```

1 Generate the time-series plot for "gnp"series that is to be indexed by "date" series. Visually inspect the plot and comment on whether there is a linear trend.

[430]:
```python
'''Reading in the data'''

data = pd.read_excel('GNP.xlsx', index_col='date')

'''Plotting'''
with plt.style.context('seaborn'):
    fig = plt.figure(figsize=(20,10))
    ax = plt.axes()
    plt.plot(data['y'], marker = 'o', label = 'y')
    ax.set_xlabel('date')
    ax.set_ylabel('y')
    plt.title('GNP timeseries (date, y)')
    plt.legend()
    plt.show()
```



1.1 Comments

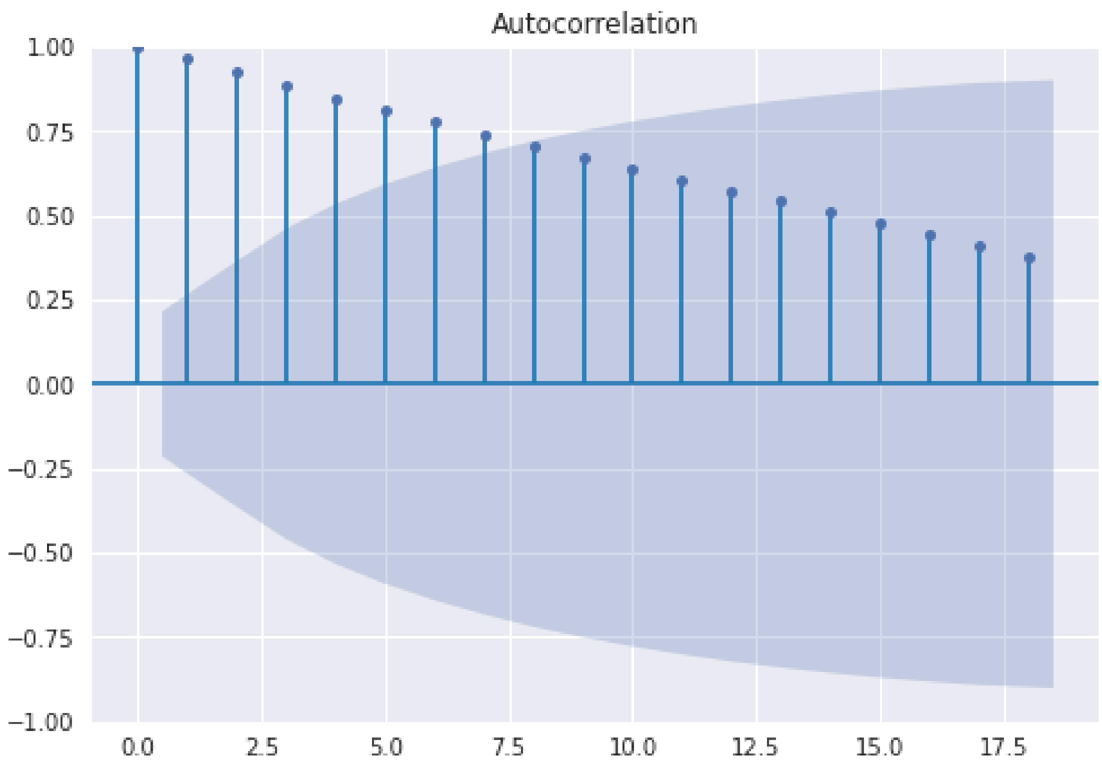
There seems to be positive linear trend.

2 Generate ACF and Variogram plots for "gnp"series and comment on whether it is stationary, why?

2.1 ACF

[431]:
```
with plt.style.context('seaborn'):
    fig = plt.figure(figsize=(20,10))
    fig = plot_acf(data['y'], lags=18)
```

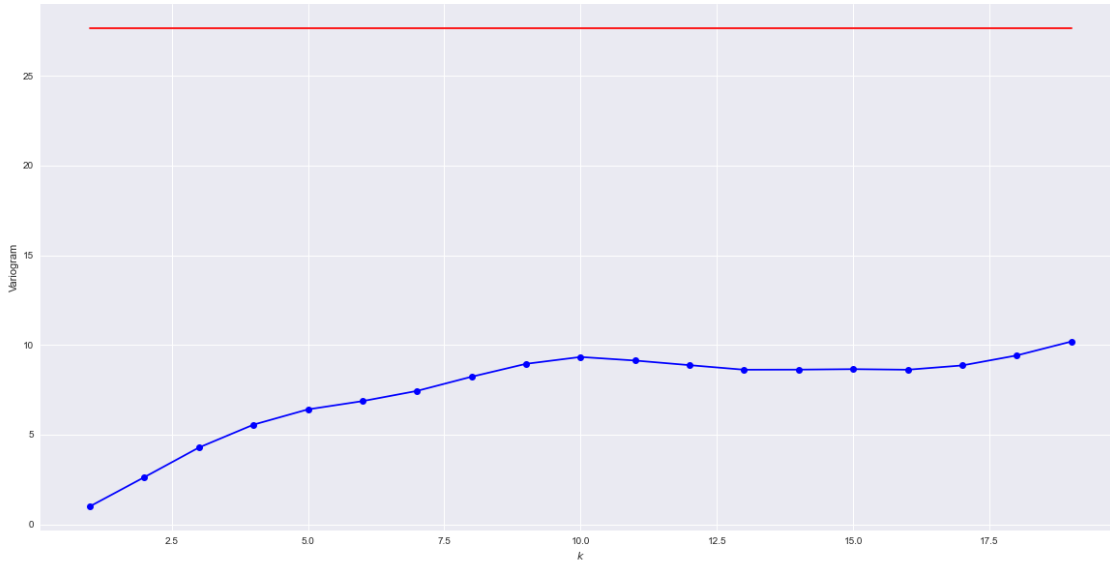
```
<Figure size 1440x720 with 0 Axes>
```



2.2 Variogram

[432]:
```
r1 = ts.acf(data['y'], nlags=int(len(data)/4))[1]
base_var  = np.var(data['y'])

variogram = pd.DataFrame(index = [i for i in range(1,int(len(data)/4))])
variogram['lagged'] = [np.var(np.array(data['y'].iloc[i:]) - np.array(data['y'].
 ↪iloc[:-i]))/np.var(np.diff(data['y'])) for i in range(1,int(len(data)/4))]
variogram ['asymp'] = [1/(1-r1) for i in range(len(variogram))]
```

[433]:
```
with plt.style.context('seaborn'):
    fig = plt.figure(figsize=(20,10))
    ax = plt.axes()
    plt.plot(variogram['lagged'], c = 'Blue', marker = 'o')
    plt.plot(variogram['asymp'], c = 'Red')
    ax.set_xlabel('$k$')
    ax.set_ylabel('Variogram')
    plt.show()
```

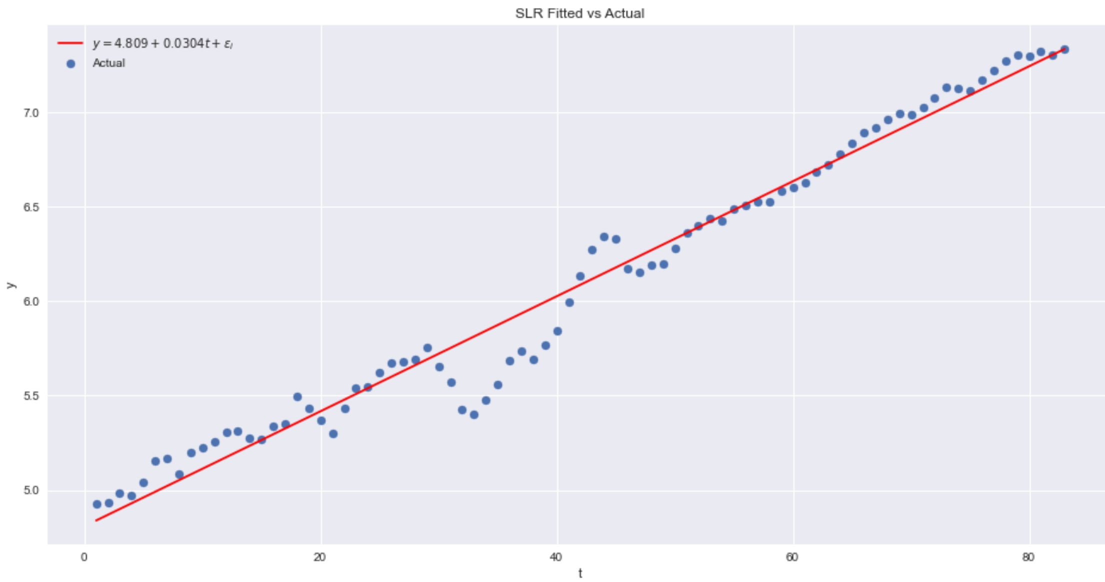


2.3 Comments

The above plots show that the current data is not stationary because the ACF plot shows that the auto-correlation is well above the signifigance levels. As for the Variogram with $\frac{1}{1-r_1}$ as our asymptote and lag $k = \frac{T}{4}$ does not pass the asymptote at all.

3 Assuming linear trend in the series, regress "gnp" on "t" with simple linear regression (SLR) model and report the following.

[434]:
```
'''Setting up data for regression'''
data_on_t = data.reset_index()
data_on_t = data_on_t.drop(['date'], axis=1)
data_on_t['t'] = [float(i) for i in data_on_t['t']]

t = pd.DataFrame(data_on_t['t'])
y = pd.DataFrame(data_on_t['y'])

'''Performing the regression'''
x = sm.add_constant(t)
OLS_fit = sm.OLS(y, x)
est = OLS_fit.fit()
summary  = est.summary() #Summary dataframe

'''Getting eveyrthing into one df'''
predict = pd.DataFrame(index = t['t'], columns=[ 'actual', 'fitted values' ,␣
↪'raw res'])
predict['fitted values'] = est.fittedvalues.values
predict['actual'] = y.values
predict['raw res'] = predict['actual'] - predict['fitted values']

with plt.style.context('seaborn'):
    fig = plt.figure(figsize=(16,8))
    ax = plt.axes()
    plt.plot(predict['fitted values'], label = '$y = 4.809 + 0.0304t +␣
↪\epsilon_i$', c='red')
    plt.scatter(predict.index.tolist(),predict['actual'], label = 'Actual')
    ax.set_xlabel('t')
    ax.set_ylabel('y')
    plt.title('SLR Fitted vs Actual')
    plt.legend()
    plt.show()
```



[435]:
```
md(summary.as_latex())
```

[435]:

Dep. Variable:	y	**R-squared:**	0.973
Model:	OLS	**Adj. R-squared:**	0.972
Method:	Least Squares	**F-statistic:**	2892.
Date:	Mon, 12 Sep 2022	**Prob (F-statistic):**	3.80e-65
Time:	20:08:10	**Log-Likelihood:**	56.863
No. Observations:	83	**AIC:**	-109.7
Df Residuals:	81	**BIC:**	-104.9
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
const	4.8090	0.027	175.834	0.000	4.755	4.863
t	0.0304	0.001	53.780	0.000	0.029	0.032

Omnibus:	28.315	**Durbin-Watson:**	0.241
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	43.763
Skew:	-1.454	**Prob(JB):**	3.14e-10
Kurtosis:	5.050	**Cond. No.**	97.6

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

3.1 The fitted SLR model.

As we can see from the summary the fitted regression model $y = \beta_0 + \beta_i x_i$ is

$$y = 4.809 + 0.0304t + \epsilon_i$$

3.2 The significance of regression effect.

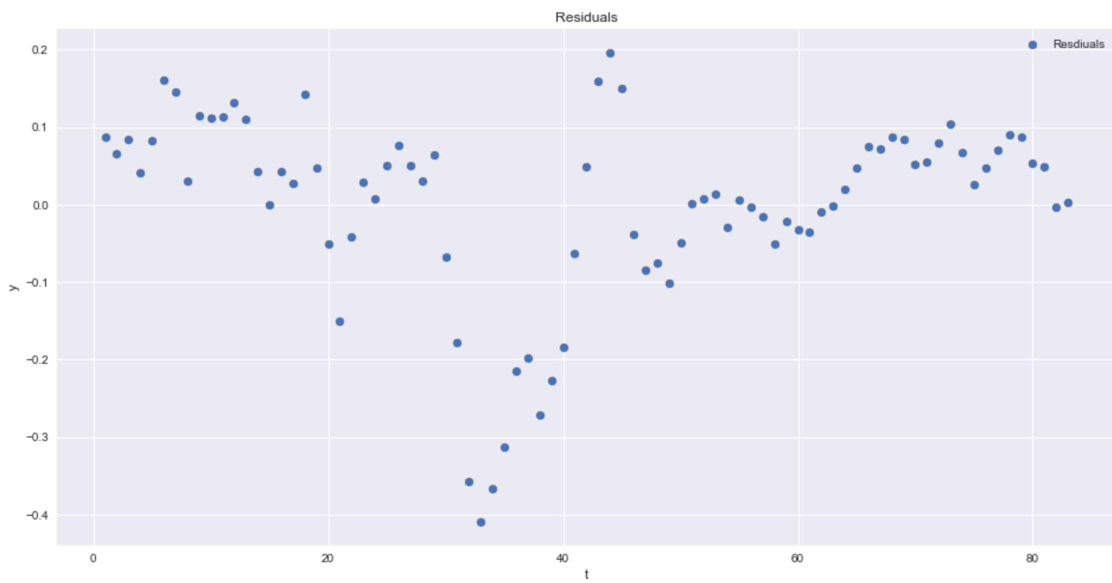
From the summary above we can see that the fitted model is significant based on the probabilities of the F-stat and p-values of the coefficients and thus for all H_0 can be rejected. This shows signifigance in the regression.

3.3 Model diagnostics.

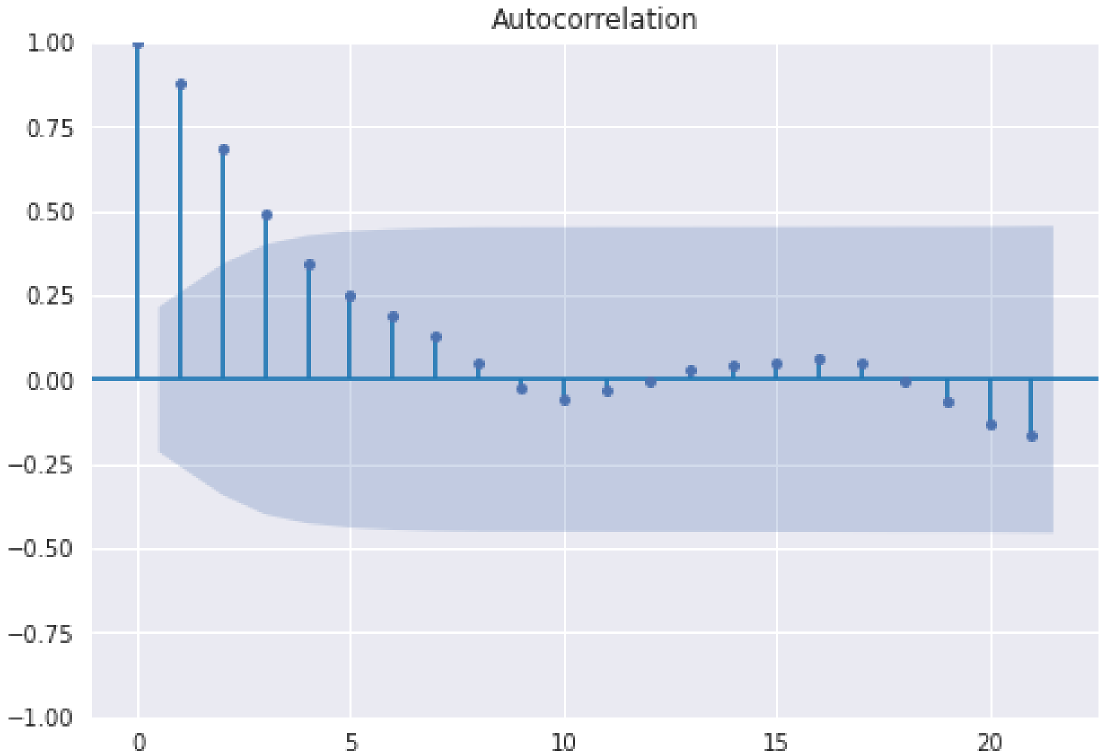
Refferring back to the summary table of the regression results. We can see that the test done for the model diagnostics include DW and JB. The DW score of 0.241 shows auto-correlation of the residuals and there is signifigance with the JB test showing normality.

4 Generate series and ACF plot for the raw residuals from the SLR fit. Comment on its plausible stationarity.

[436]:
```
with plt.style.context('seaborn'):
    fig = plt.figure(figsize=(16,8))
    ax = plt.axes()
    plt.scatter(predict.index.tolist(),est.resid, label = 'Resdiuals')
    ax.set_xlabel('t')
    ax.set_ylabel('y')
    plt.title('Residuals')
    plt.legend()
    plt.show()
```



[437]:
```
with plt.style.context('seaborn'):
    fig = plot_acf(est.resid, lags=21)
```



4.1 Comments

Based on the the ACF, we can see that there seems to be exponential deacy and thus is indicative of a stationary process.

5 Referring to "TSA04.R" for the use of gls function (nlme package is required), perform the second order Cochrane-Orcutt procedure and report the fitted autoregression model (with the estimates of ϕ_1 and ϕ_2)

We must first start presenting the formulation of the second order Autoregression model:

Let us first present the following for y_t:

$$y_t = \beta_0 + \beta_1 x_t + \epsilon_t$$

$$y_{t-1} = \beta_0 + \beta_1 x_{t-1} + \epsilon_{t-1}$$

$$y_{t-2} = \beta_0 + \beta_1 x_{t-2} + \epsilon_{t-2}$$

Introducing our parameter ϕ_i we arrive at:

$$y_t = \beta_0 + \beta_1 x_t + \epsilon_t$$

$$\phi_1 y_{t-1} = \phi_1 \beta_0 + \phi_1 \beta_1 x_{t-1} + \phi_1 \epsilon_{t-1}$$

$$\phi_2 y_{t-2} = \phi_2 \beta_0 + \phi_2 \beta_1 x_{t-2} + \phi_2 \epsilon_{t-2}$$

Since we are formulating the second order Autoregressive model we will show the following:

$$y_t - \phi_1 y_{t-1} - \phi_2 y_{t-2} = \beta_0(1 - \phi_1 - \phi_2) + \beta_1(x_t - \phi_1 x_{t-1} - \phi_2 x_{t-2}) + (\epsilon_t - \phi_1 \epsilon_{t-1} - \phi_2 \epsilon_{t-2})$$

Where:

$$y_t^{''} = (1 - \phi_1 - \phi_2) y_t$$

$$x_t^{''} = (1 - \phi_1 - \phi_2) x_t$$

$$\beta_0^{''} = (1 - \phi_1 - \phi_2) \beta_0$$

$$\epsilon_t = \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + a_t$$

Thus:

$$y_t^{''} = \beta_0^{''} + \beta_1 x_t^{''} + a_t$$

[438]:
```
import rpy2.robjects as robjects
from rpy2.robjects import FloatVector
from rpy2.robjects.packages import importr
from rpy2.robjects import DataFrame

base = importr('base')
nlme = importr('nlme')
rsummary = robjects.r['summary']

y_r = FloatVector(y.values)
t_r = FloatVector(t.values)
```

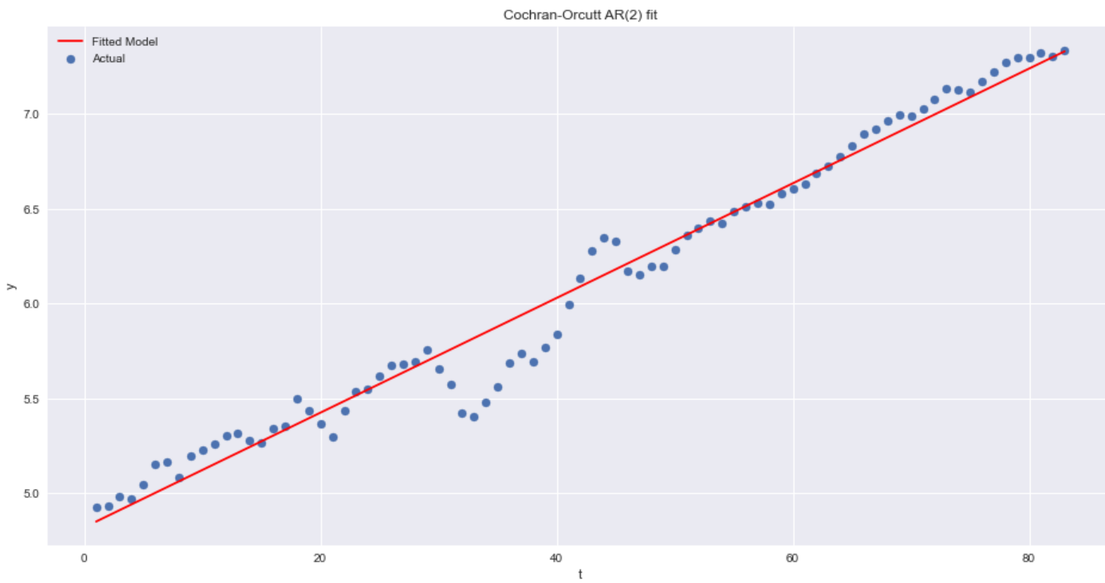
```python
fmla = robjects.Formula('y_r ~ t_r')

env = fmla.environment
env['y_r'] = y_r
env['t_r'] = t_r

fit = nlme.gls(fmla, correlation = nlme.corARMA(p=2), method="ML")
print(rsummary(fit))
```

```
Generalized least squares fit by maximum likelihood
  Model: y_r ~ t_r
  Data: NULL
        AIC       BIC   logLik
  -238.0689 -225.9747 124.0345

Correlation Structure: ARMA(2,0)
 Formula: ~1
 Parameter estimate(s):
      Phi1       Phi2
 1.2040997 -0.3748382

Coefficients:
               Value  Std.Error  t-value p-value
(Intercept) 4.820705 0.06531402 73.80812       0
t_r         0.030221 0.00132936 22.73360       0

 Correlation:
    (Intr)
t_r -0.855

Standardized residuals:
       Min         Q1        Med         Q3        Max
-3.4544505 -0.3167343  0.2016143  0.6100012  1.6131489

Residual standard error: 0.1200595
Degrees of freedom: 83 total; 81 residual
```

[439]:

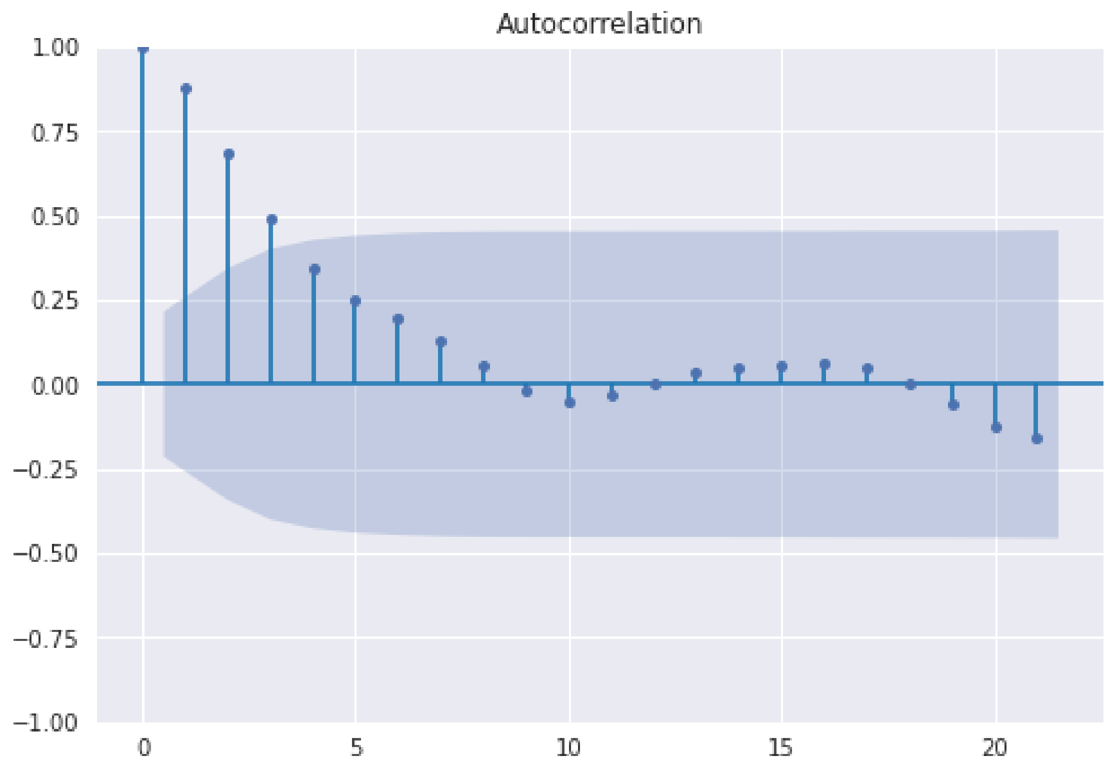
```python
co_df = pd.DataFrame(index=t['t'], columns=['fitted'])
co_df['fitted'] = [fit[12][i] for i in range(len(fit[13]))]

with plt.style.context('seaborn'):
    fig = plt.figure(figsize=(16,8))
    ax = plt.axes()
    plt.plot(co_df, label = 'Fitted Model', c = 'Red')
    plt.scatter(t,y, label = 'Actual')
```

```
    ax.set_xlabel('t')
    ax.set_ylabel('y')
    plt.title('Cochran-Orcutt AR(2) fit')
    plt.legend()
    plt.show()
```



6 Generate ACF plot for the raw residuals from the SLR fit. Comment on the significance of its autocorrelation across the lags.

[440]:
```
reside = [fit[13][i] for i in range(len(fit[13]))]
with plt.style.context('seaborn'):
    fig = plot_acf(reside, lags=21)
```



6.1 Comments

The ACF above almost completely matches the prior residuals and thus we can conclude stationarity.