Homework 4

July 15, 2025

Author:Alejandro M.Ouslan

[1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
import statsmodels.stats.api as sms
from statsmodels.tsa.arima.model import ARIMA
import statsmodels.api as sm
from statsmodels.tsa.ar_model import AutoReg
import numpy as np
from statsmodels.stats.diagnostic import acorr_ljungbox
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

[2]:

```
df = pd.read_excel("data/MONEYDEM-1.xls")
df['year'] = df['DATE'].astype(int)
df['quarter'] = ((df['DATE'] - df['year']) * 10).round().astype(int)
df['date'] = df['year'].astype(str) + 'Q' + df['quarter'].astype(str)
df['date'] = pd.PeriodIndex(df['date'], freq='Q')
df.set_index('date', inplace=True)
df
```

[2]:

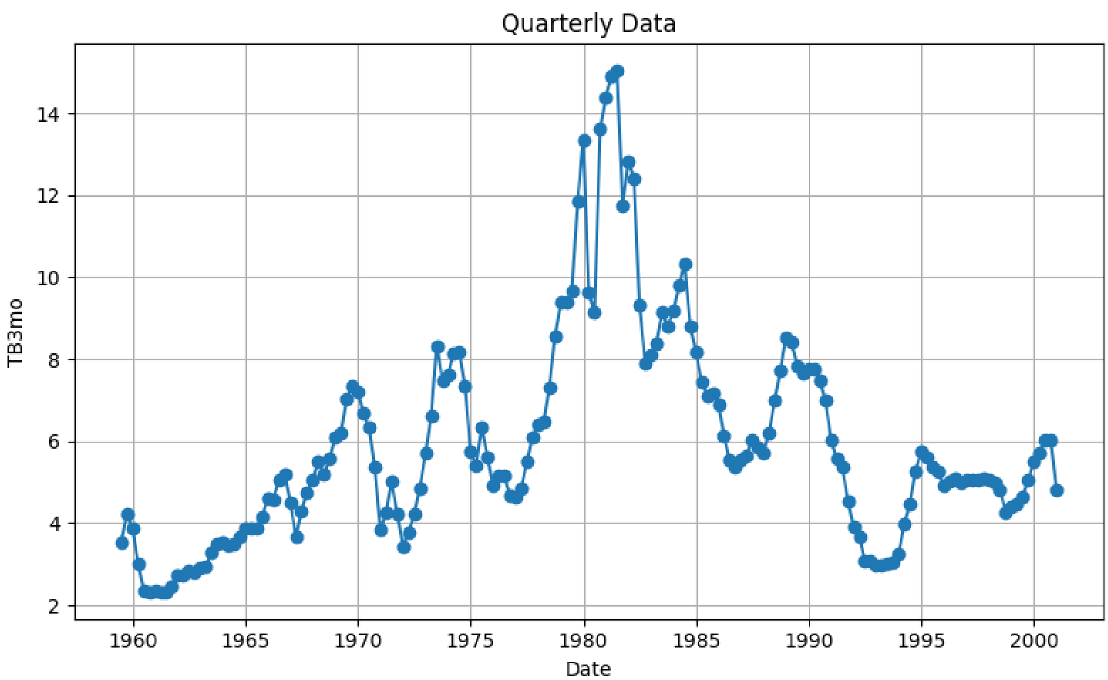
	DATE	TB3mo	TB1yr	year	quarter
date					
1959Q3	1959.3	3.540000	4.493333	1959	3
1959Q4	1959.4	4.230000	4.740000	1959	4
1960Q1	1960.1	3.873333	4.360000	1960	1
1960Q2	1960.2	2.993333	3.646667	1960	2
1960Q3	1960.3	2.360000	2.903333	1960	3
…	…	…	…	…	…
2000Q1	2000.1	5.520000	5.816667	2000	1
2000Q2	2000.2	5.713333	5.856667	2000	2
2000Q3	2000.3	6.016667	5.803333	2000	3
2000Q4	2000.4	6.016667	5.630000	2000	4
2001Q1	2001.1	4.816667	4.416667	2001	1

[167 rows x 5 columns]

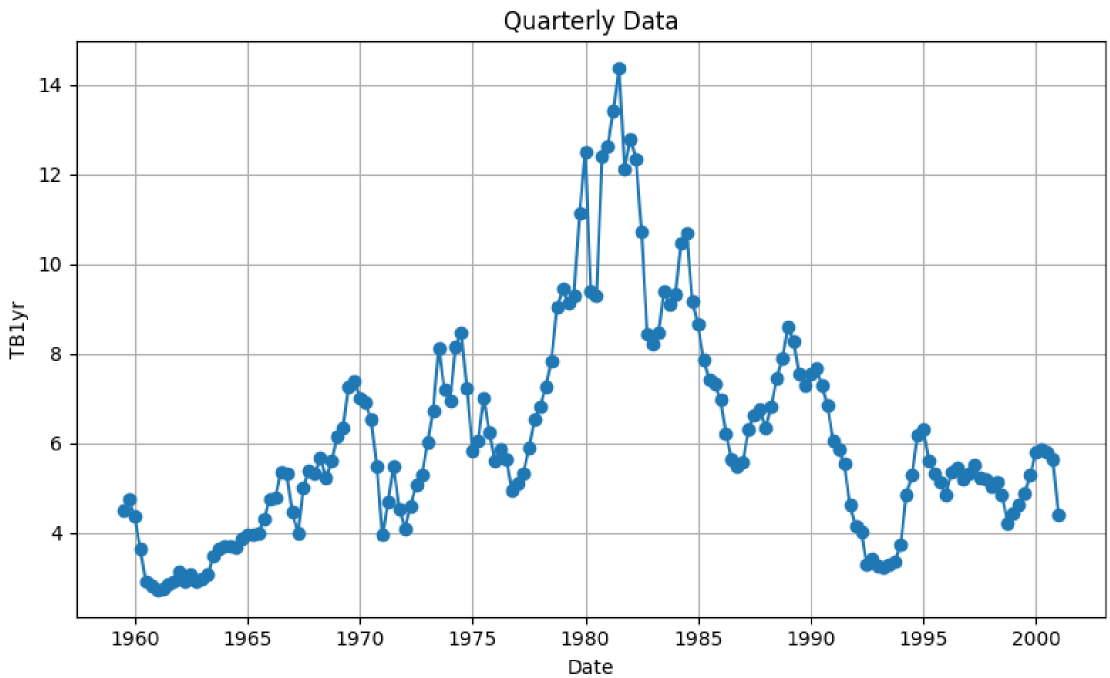
1 Problems

1.1 Problem a

[3]:
```
#1a
plt.figure(figsize=(8, 5))
plt.plot(df.index.to_timestamp(), df['TB3mo'], marker='o')
plt.title('Quarterly Data')
plt.xlabel('Date')
plt.ylabel('TB3mo')
plt.grid(True)
plt.tight_layout()
plt.show()
```



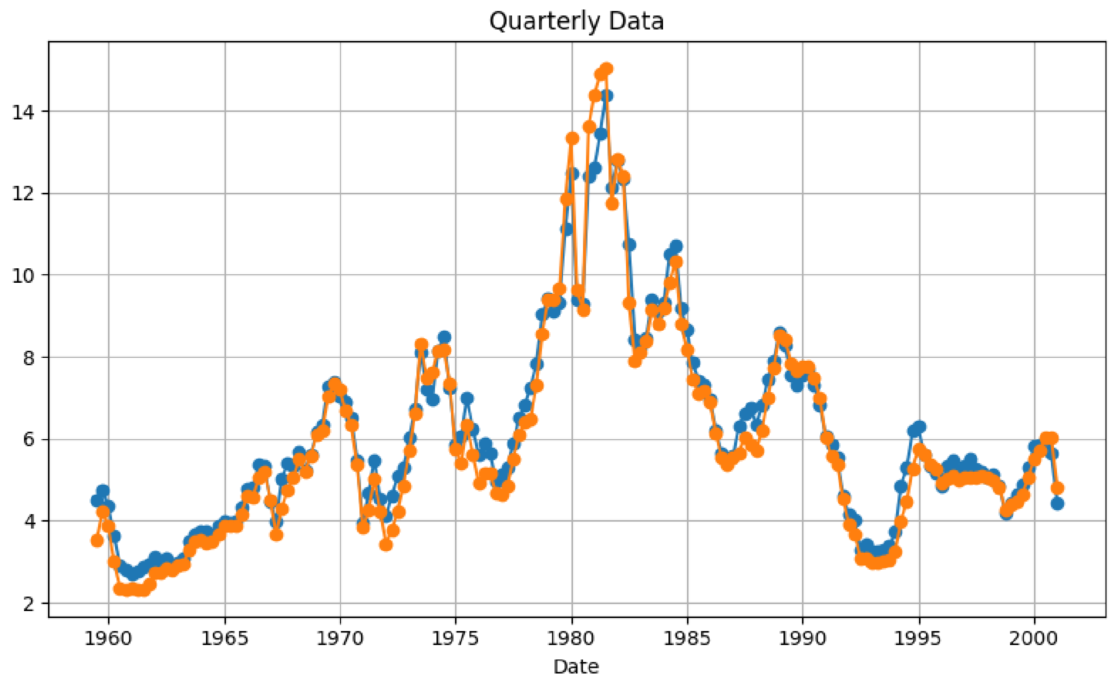
[4]:
```
plt.figure(figsize=(8, 5))
plt.plot(df.index.to_timestamp(), df['TB1yr'], marker='o')
plt.title('Quarterly Data')
plt.xlabel('Date')
plt.ylabel('TB1yr')
plt.grid(True)
plt.tight_layout()
plt.show()
```



- No, they appear to have changing mean and variance given that in 1980 had a vilont spike

1.2 Problem b

[5]:
```
plt.figure(figsize=(8, 5))
plt.plot(df.index.to_timestamp(), df['TB1yr'], marker='o')
plt.plot(df.index.to_timestamp(), df['TB3mo'], marker='o')
plt.title('Quarterly Data')
plt.xlabel('Date')
plt.grid(True)
plt.tight_layout()
plt.show()
```



- They apear to be practicaly identical the seam to follow the same trend

1.3 Problem c

[6]:
```python
model = smf.ols("TB1yr ~ TB3mo", data=df).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  TB1yr   R-squared:                       0.980
Model:                            OLS   Adj. R-squared:                  0.980
Method:                 Least Squares   F-statistic:                     7975.
Date:                Tue, 15 Jul 2025   Prob (F-statistic):          1.30e-141
Time:                        21:30:14   Log-Likelihood:                -56.689
No. Observations:                 167   AIC:                             117.4
Df Residuals:                     165   BIC:                             123.6
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.6981      0.067     10.486      0.000       0.567       0.830
TB3mo          0.9167      0.010     89.302      0.000       0.896       0.937
==============================================================================
Omnibus:                       15.536   Durbin-Watson:                   0.623
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               39.846
```

```
Skew:                           0.289   Prob(JB):                     2.23e-09
Kurtosis:                       5.322   Cond. No.                         16.7
==============================================================================

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

1.4 Problem d

- A 1 percentage point increase in the 3-month Treasury rate (short-term) is associated with a 0.9167 percentage point increase in the 1-year Treasury rate (long-term), on average, holding other factors constant.

1.5 Problem e

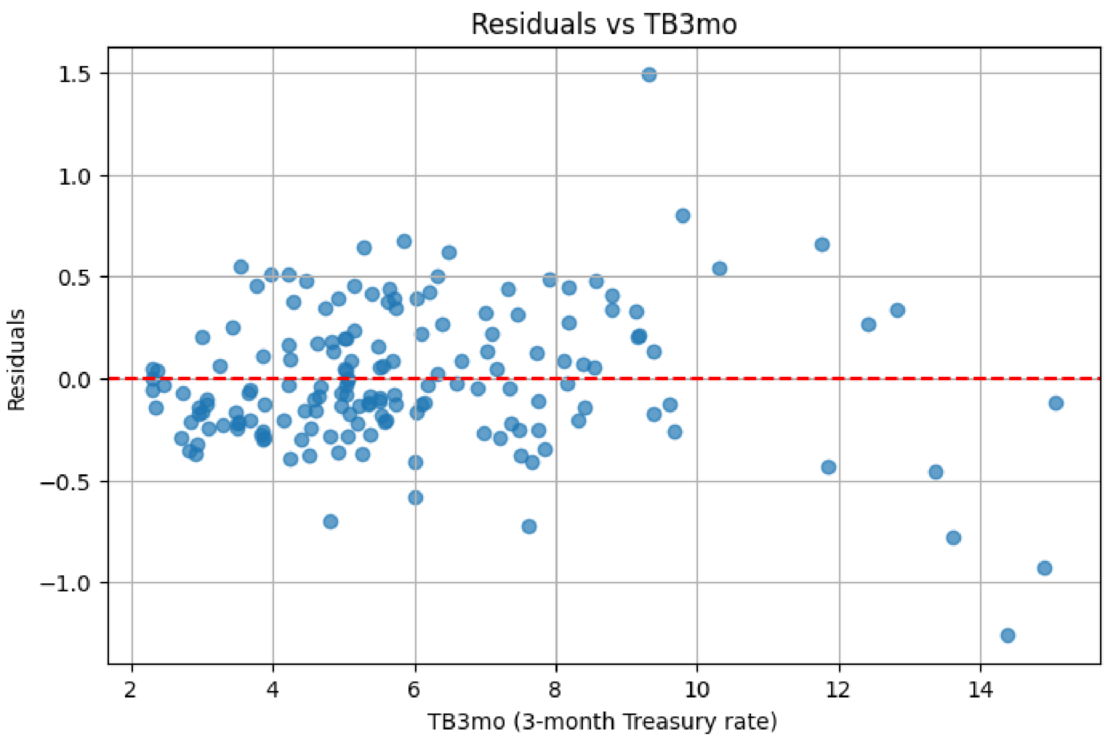
[7]:
```
t_test_result = model.t_test('TB3mo = 1')
print(t_test_result)
```

```
                             Test for Constraints
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
c0             0.9167      0.010     -8.112      0.000       0.896       0.937
==============================================================================
```

- since p-value is 0.000 there is strong but less-than-perfect pass-through from short- to long-term rates.

1.6 Problem f

[8]:
```
residuals = model.resid

plt.figure(figsize=(8, 5))
plt.scatter(df['TB3mo'], residuals, alpha=0.7)
plt.axhline(0, color='red', linestyle='--')
plt.xlabel('TB3mo (3-month Treasury rate)')
plt.ylabel('Residuals')
plt.title('Residuals vs TB3mo')
plt.grid(True)
plt.show()
```



- No there does not seam to be a pattern

1.7 Problem g

[9]:
```
residuals = model.resid
exog = model.model.exog

white_test = sms.het_white(residuals, exog)

lm_stat, lm_pvalue, f_stat, f_pvalue = white_test

print(f"White test LM statistic: {lm_stat:.4f}")
print(f"White test LM p-value: {lm_pvalue:.4f}")
print(f"White test F statistic: {f_stat:.4f}")
print(f"White test F p-value: {f_pvalue:.4f}")
```

```
White test LM statistic: 31.3738
White test LM p-value: 0.0000
White test F statistic: 18.9687
White test F p-value: 0.0000
```

- given that the p-value is 0.0000 there is evidence of heteroskedasticity

1.8 Problem h

[10]:
```
model_robust = smf.ols("TB1yr ~ TB3mo", data=df).fit(cov_type='HC0')
print(model_robust.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  TB1yr   R-squared:                       0.980
Model:                            OLS   Adj. R-squared:                  0.980
Method:                 Least Squares   F-statistic:                     3085.
Date:                Tue, 15 Jul 2025   Prob (F-statistic):          1.05e-108
Time:                        21:30:14   Log-Likelihood:                -56.689
No. Observations:                 167   AIC:                             117.4
Df Residuals:                     165   BIC:                             123.6
Df Model:                           1
Covariance Type:                  HC0
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.6981      0.088      7.940      0.000       0.526       0.870
TB3mo          0.9167      0.017     55.539      0.000       0.884       0.949
==============================================================================
Omnibus:                       15.536   Durbin-Watson:                   0.623
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               39.846
Skew:                           0.289   Prob(JB):                     2.23e-09
Kurtosis:                       5.322   Cond. No.                         16.7
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
```

- The coefecients did not change, however the std err did increase for both the intercept and the coeficient When heteroskedasticity is present, regular SEs underestimate the true variability of the coefficients, so the robust SEs tend to be larger and more reliable.

1.9 Problem j

[11]:
```
df['D'] = (df['TB3mo'] > 10.00).astype(int)
model_with_dummy = smf.ols("TB1yr ~ TB3mo + D", data=df).fit()

print(model_with_dummy.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  TB1yr   R-squared:                       0.981
Model:                            OLS   Adj. R-squared:                  0.981
Method:                 Least Squares   F-statistic:                     4176.
Date:                Tue, 15 Jul 2025   Prob (F-statistic):          2.16e-141
Time:                        21:30:14   Log-Likelihood:                -52.401
```

```
No. Observations:                 167   AIC:                             110.8
Df Residuals:                     164   BIC:                             120.2
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.5551      0.081      6.832      0.000       0.395       0.716
TB3mo          0.9452      0.014     67.730      0.000       0.918       0.973
D             -0.4456      0.152     -2.940      0.004      -0.745      -0.146
==============================================================================
Omnibus:                       15.483   Durbin-Watson:                   0.597
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               26.029
Skew:                           0.476   Prob(JB):                     2.23e-06
Kurtosis:                       4.684   Cond. No.                         40.8
==============================================================================

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

1.10 Problem k

- given that the p-value of delta is 0.004 the dummy is revelat.

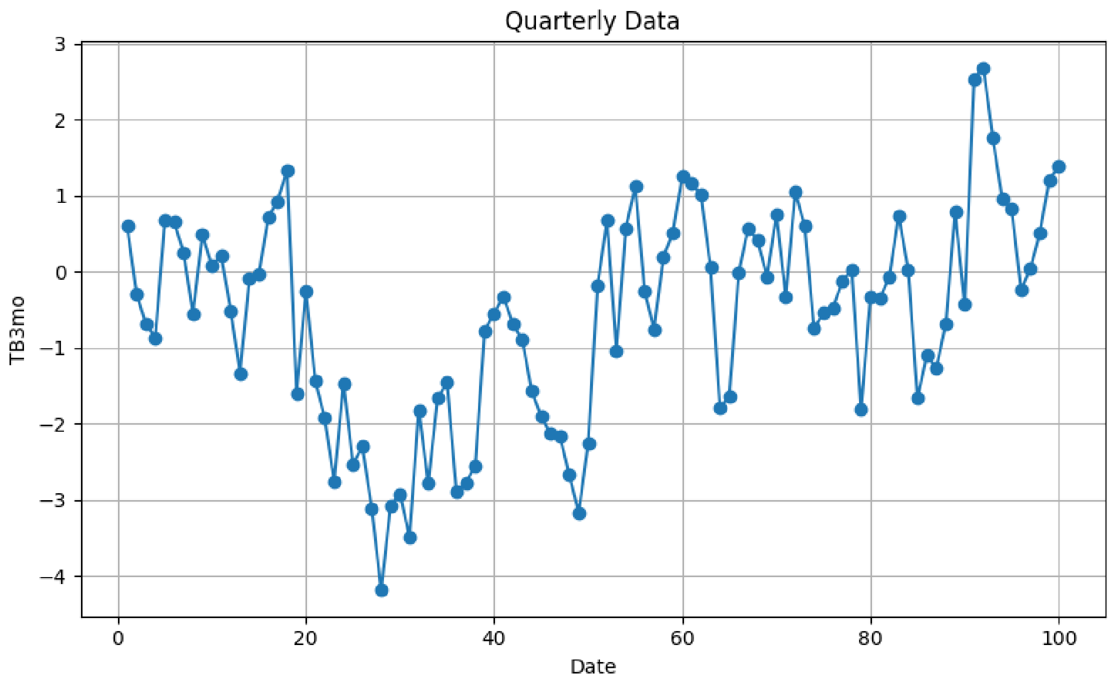
1.11 Problem l

- The coeficien increased but also did the standard error

2 Problem 2

2.1 Problem A

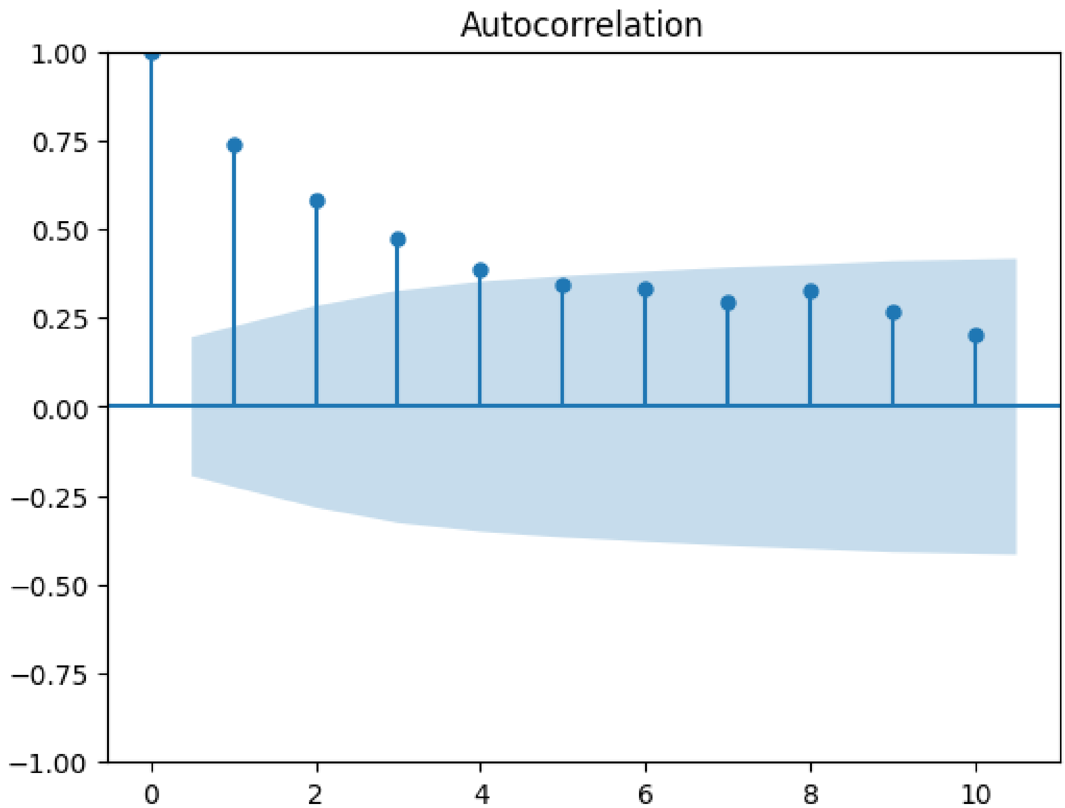
[12]:
```
df = pd.read_excel("data/SIM_2-1.xls")
plt.figure(figsize=(8, 5))
plt.plot(df["OBS"], df['Y1'], marker='o')
plt.title('Quarterly Data')
plt.xlabel('Date')
plt.ylabel('TB3mo')
plt.grid(True)
plt.tight_layout()
plt.show()
```

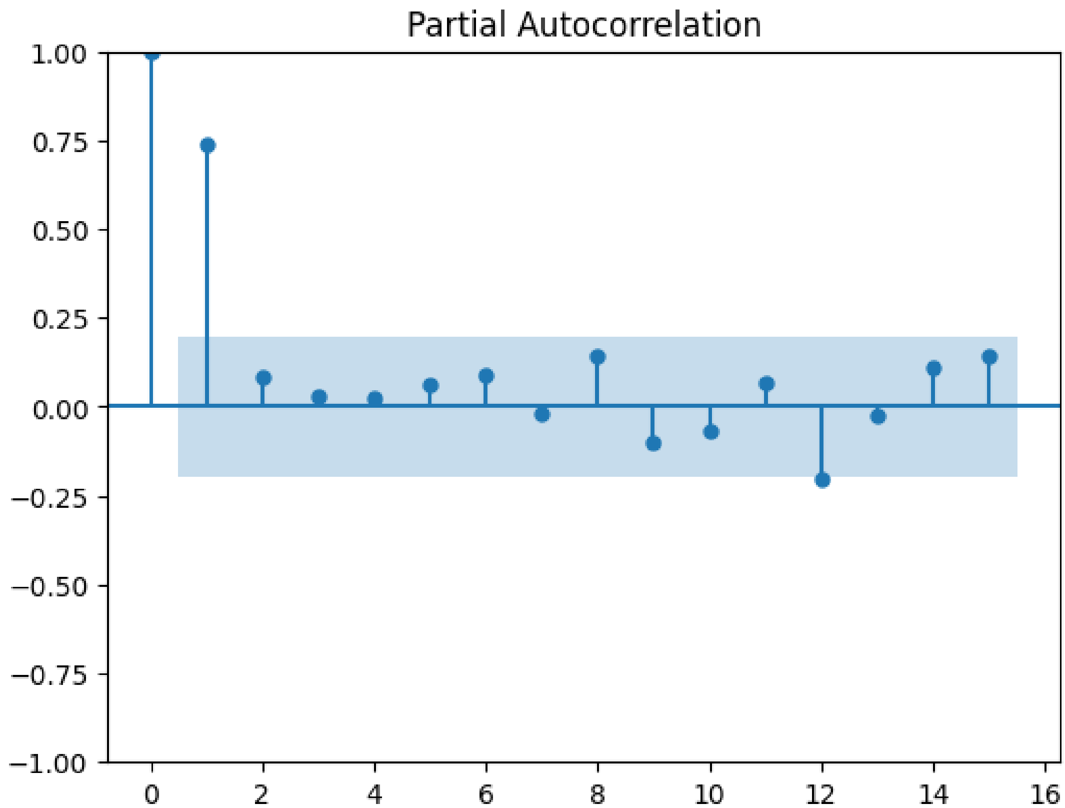


- given that the series has not broke out dwon or up and it seems to hover aroun 1 we could say that it is stationary

2.2 Problem B

[13]:
```
fig = plot_acf(df['Y1'], lags=10)
plt.show()
plot_pacf(df['Y1'], lags=15)
plt.show()
```


Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
2
4
6
8
10



- given that the series has not broke out dwon or up and it seems to hover aroun 1 we could say that it is stationary

2.3 Problem C

[14]:
```
# AR(1)
res = AutoReg(df['Y1'], lags =1).fit()
print(res.summary())
y_true = res.model.endog[res.model._hold_back:]
y_pred = res.fittedvalues


ssr = np.sum((y_true - y_pred) ** 2)

tss = np.sum((y_true - np.mean(y_true)) ** 2)

n = len(y_true)
k = res.df_model + 1

r2 = 1 - ssr / tss
r2_adj = 1 - (ssr / (n - k)) / (tss / (n - 1))
```

```
print("R^2:", round(r2, 4))
print("Adjusted R^2:", round(r2_adj, 4))
```

```
                            AutoReg Model Results
==============================================================================
Dep. Variable:                     Y1   No. Observations:                  100
Model:                     AutoReg(1)   Log Likelihood                -132.061
Method:               Conditional MLE   S.D. of innovations              0.919
Date:                Tue, 15 Jul 2025   AIC                            270.122
Time:                        21:30:14   BIC                            277.907
Sample:                             1   HQIC                           273.272
                                  100
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1372      0.100     -1.366      0.172      -0.334       0.060
Y1.L1          0.7545      0.067     11.269      0.000       0.623       0.886
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.3254           +0.0000j            1.3254            0.0000
-----------------------------------------------------------------------------
R^2: 0.5619
Adjusted R^2: 0.5528
```

[15]:
```
# AR(2)
res = AutoReg(df['Y1'], lags=2).fit()
print(res.summary())
y_true = res.model.endog[res.model._hold_back:]
y_pred = res.fittedvalues

ssr = np.sum((y_true - y_pred) ** 2)

tss = np.sum((y_true - np.mean(y_true)) ** 2)

n = len(y_true)
k = res.df_model + 1

r2 = 1 - ssr / tss
r2_adj = 1 - (ssr / (n - k)) / (tss / (n - 1))

print("R^2:", round(r2, 4))
print("Adjusted R^2:", round(r2_adj, 4))
```

```
                            AutoReg Model Results
==============================================================================
```

```
Dep. Variable:                     Y1   No. Observations:                  100
Model:                     AutoReg(2)   Log Likelihood                -130.629
Method:               Conditional MLE   S.D. of innovations              0.918
Date:                Tue, 15 Jul 2025   AIC                            269.258
Time:                        21:30:14   BIC                            279.598
Sample:                             2   HQIC                           273.440
                                  100
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1144      0.102     -1.116      0.264      -0.315       0.086
Y1.L1          0.6939      0.101      6.896      0.000       0.497       0.891
Y1.L2          0.0870      0.101      0.861      0.389      -0.111       0.285
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.2462           +0.0000j            1.2462            0.0000
AR.2           -9.2181           +0.0000j            9.2181            0.5000
-----------------------------------------------------------------------------
R^2: 0.567
Adjusted R^2: 0.5532
```

[16]:

```python
# ARMA(1,1)
arma_mod = ARIMA(df['Y1'], order=(1, 1, 0)).fit()
print(arma_mod.summary())

y_true = df['Y1'].diff().values


y_pred = arma_mod.fittedvalues

ssr = np.sum((y_true - y_pred) ** 2)
tss = np.sum((y_true - np.mean(y_true)) ** 2)

n = len(y_true)
k = arma_mod.df_model

r2 = 1 - ssr / tss
r2_adj = 1 - (ssr / (n - k)) / (tss / (n - 1))

print("R^2:", round(r2, 4))
print("Adjusted R^2:", round(r2_adj, 4))

print("AIC:", arma_mod.aic)
print("BIC:", arma_mod.bic)
```

```
                               SARIMAX Results
```

```
==============================================================================
Dep. Variable:                     Y1   No. Observations:                  100
Model:                 ARIMA(1, 1, 0)   Log Likelihood                -136.382
Date:                Tue, 15 Jul 2025   AIC                            276.764
Time:                        21:30:14   BIC                            281.954
Sample:                             0   HQIC                           278.864
                                - 100
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.1982      0.106     -1.868      0.062      -0.406       0.010
sigma2         0.9203      0.125      7.356      0.000       0.675       1.165
================================================================================
===
Ljung-Box (L1) (Q):                   0.05   Jarque-Bera (JB):
0.23
Prob(Q):                              0.82   Prob(JB):
0.89
Heteroskedasticity (H):               1.04   Skew:
-0.01
Prob(H) (two-sided):                  0.91   Kurtosis:
3.23
================================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
R^2: nan
Adjusted R^2: nan
AIC: 276.7637125558847
BIC: 281.95395225615385
```

[17]:

```python
# ARMA(1,4)
arma_mod = ARIMA(df['Y1'], order=(1, 4, 0)).fit()
print(arma_mod.summary())
y_true = df['Y1'].diff().values

y_pred = arma_mod.fittedvalues
ssr = np.sum((y_true - y_pred) ** 2)
tss = np.sum((y_true - np.mean(y_true)) ** 2)

n = len(y_true)
k = arma_mod.df_model

r2 = 1 - ssr / tss
```

```
r2_adj = 1 - (ssr / (n - k)) / (tss / (n - 1))

print("R^2:", round(r2, 4))
print("Adjusted R^2:", round(r2_adj, 4))

print("AIC:", arma_mod.aic)
print("BIC:", arma_mod.bic)
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                     Y1   No. Observations:                  100
Model:                 ARIMA(1, 4, 0)   Log Likelihood                -248.774
Date:                Tue, 15 Jul 2025   AIC                            501.548
Time:                        21:30:14   BIC                            506.677
Sample:                             0   HQIC                           503.621
                                - 100
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.7572      0.064    -11.873      0.000      -0.882      -0.632
sigma2        10.3403      1.580      6.546      0.000       7.244      13.436
================================================================================
===
Ljung-Box (L1) (Q):                  23.50   Jarque-Bera (JB):
2.57
Prob(Q):                              0.00   Prob(JB):
0.28
Heteroskedasticity (H):               1.16   Skew:
0.39
Prob(H) (two-sided):                  0.68   Kurtosis:
2.86
================================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
R^2: nan
Adjusted R^2: nan
AIC: 501.54819826952695
BIC: 506.67689465246264
```

[18]:
```
# ARMA(2,1)
arma_mod = ARIMA(df['Y1'], order=(2, 1, 0)).fit()
print(arma_mod.summary())
y_true = df['Y1'].diff().values
```

```python
y_pred = arma_mod.fittedvalues
ssr = np.sum((y_true - y_pred) ** 2)
tss = np.sum((y_true - np.mean(y_true)) ** 2)

n = len(y_true)
k = arma_mod.df_model

r2 = 1 - ssr / tss
r2_adj = 1 - (ssr / (n - k)) / (tss / (n - 1))

print("R^2:", round(r2, 4))
print("Adjusted R^2:", round(r2_adj, 4))

print("AIC:", arma_mod.aic)
print("BIC:", arma_mod.bic)
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                     Y1   No. Observations:                  100
Model:                 ARIMA(2, 1, 0)   Log Likelihood                -135.770
Date:                Tue, 15 Jul 2025   AIC                            277.540
Time:                        21:30:14   BIC                            285.325
Sample:                             0   HQIC                           280.690
                                - 100
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.2212      0.109     -2.034      0.042      -0.434      -0.008
ar.L2         -0.1107      0.105     -1.052      0.293      -0.317       0.095
sigma2         0.9087      0.122      7.439      0.000       0.669       1.148
================================================================================
===
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):
0.45
Prob(Q):                              0.92   Prob(JB):
0.80
Heteroskedasticity (H):               1.01   Skew:
0.05
Prob(H) (two-sided):                  0.97   Kurtosis:
3.31
================================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
```

```
R^2: nan
Adjusted R^2: nan
AIC: 277.5398840453032
BIC: 285.325243595707
```

2.4 Problem D

[19]:
```python
#AR(2)
res = AutoReg(df['Y1'], lags=2, trend="n").fit()
print(res.summary())
y_true = df['Y1'].diff().values

y_pred = arma_mod.fittedvalues
ssr = np.sum((y_true - y_pred) ** 2)
tss = np.sum((y_true - np.mean(y_true)) ** 2)

n = len(y_true)
k = arma_mod.df_model
r2 = 1 - ssr / tss
r2_adj = 1 - (ssr / (n - k)) / (tss / (n - 1))

print("R^2:", round(r2, 4))
print("Adjusted R^2:", round(r2_adj, 4))

print("AIC:", arma_mod.aic)
print("BIC:", arma_mod.bic)
```

```
                            AutoReg Model Results
==============================================================================
Dep. Variable:                     Y1   No. Observations:                  100
Model:                     AutoReg(2)   Log Likelihood                -131.248
Method:               Conditional MLE   S.D. of innovations              0.923
Date:                Tue, 15 Jul 2025   AIC                            268.496
Time:                        21:30:14   BIC                            276.251
Sample:                             2   HQIC                           271.633
                                  100
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Y1.L1          0.7102      0.100      7.087      0.000       0.514       0.907
Y1.L2          0.1051      0.100      1.046      0.295      -0.092       0.302
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.1963           +0.0000j            1.1963            0.0000
AR.2           -7.9532           +0.0000j            7.9532            0.5000
-----------------------------------------------------------------------------
```

```
R^2: nan
Adjusted R^2: nan
AIC: 277.5398840453032
BIC: 285.325243595707
```

[20]:
```
# ARIMA(1,1)
arma_mod = ARIMA(df['Y1'], order=(1, 1, 0), trend="n").fit()
print(arma_mod.summary())
y_true = df['Y1'].diff().values

y_pred = arma_mod.fittedvalues

ssr = np.sum((y_true - y_pred) ** 2)
tss = np.sum((y_true - np.mean(y_true)) ** 2)

n = len(y_true)
k = arma_mod.df_model

r2 = 1 - ssr / tss
r2_adj = 1 - (ssr / (n - k)) / (tss / (n - 1))

print("R^2:", round(r2, 4))
print("Adjusted R^2:", round(r2_adj, 4))

print("AIC:", arma_mod.aic)
print("BIC:", arma_mod.bic)
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                     Y1   No. Observations:                  100
Model:                 ARIMA(1, 1, 0)   Log Likelihood                -136.382
Date:                Tue, 15 Jul 2025   AIC                            276.764
Time:                        21:30:14   BIC                            281.954
Sample:                             0   HQIC                           278.864
                                - 100
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.1982      0.106     -1.868      0.062      -0.406       0.010
sigma2         0.9203      0.125      7.356      0.000       0.675       1.165
================================================================================
===
Ljung-Box (L1) (Q):                   0.05   Jarque-Bera (JB):
0.23
Prob(Q):                              0.82   Prob(JB):
0.89
Heteroskedasticity (H):               1.04   Skew:
```

```
-0.01
Prob(H) (two-sided):                  0.91   Kurtosis:
3.23
================================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
R^2: nan
Adjusted R^2: nan
AIC: 276.7637125558847
BIC: 281.95395225615385
```

2.5 Problem E

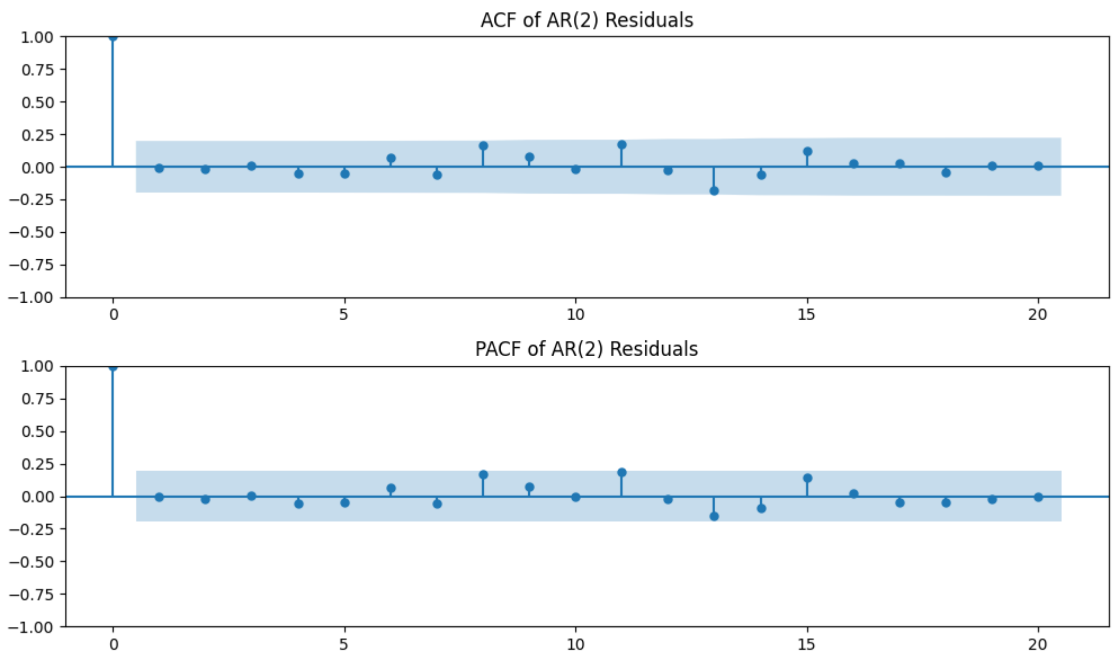
- for the part c looking at the AIC the best model is the AR(2) given that it has the smallest AIC fro the part ed the best is sitll AR(2) given that it has the smallest AIC

2.6 Problem F

- yes because looking at the AIC it shouldgest that the best model is AR(2) given that the simulated model was created with AR(1) one would expect AR(1) would be the best model

2.7 Problem G

[21]:
```
# AR(2)
model_ar2 = AutoReg(df['Y1'], lags=2, old_names=False)
res_ar2 = model_ar2.fit()

residuals = res_ar2.resid

fig, ax = plt.subplots(2, 1, figsize=(10, 6))

plot_acf(residuals, ax=ax[0], lags=20)
ax[0].set_title('ACF of AR(2) Residuals')

plot_pacf(residuals, ax=ax[1], lags=20, method='ywm')
ax[1].set_title('PACF of AR(2) Residuals')

plt.tight_layout()
plt.show()
```



- yes they look like random noice

3 Problem

3.1 Problem A

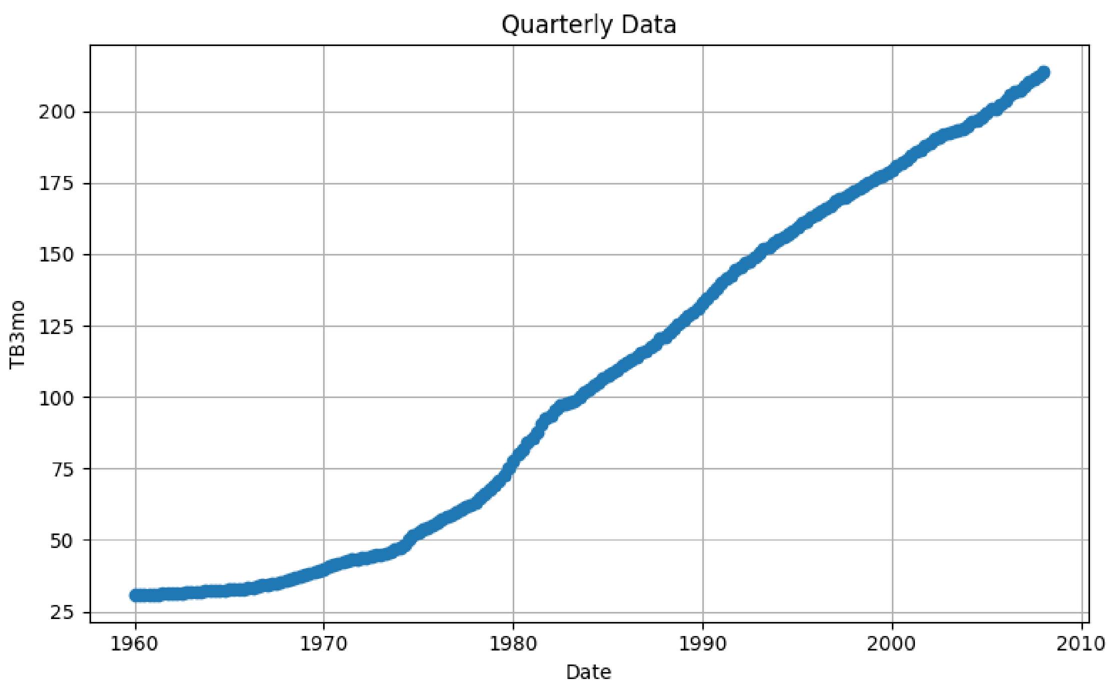
[22]:
```
df = pd.read_excel("data/QUARTERLY-1.xls")
df['date'] = pd.PeriodIndex(df['Date'], freq='Q')
df.set_index('date', inplace=True)
df = df[["CPINSA","Date"]]
df
```

[22]:

	CPINSA	Date
date		
1960Q1	30.57	1960Q1
1960Q2	30.60	1960Q2
1960Q3	30.60	1960Q3
1960Q4	30.80	1960Q4
1961Q1	30.80	1961Q1
…	…	…
2007Q1	209.01	2007Q1
2007Q2	210.37	2007Q2
2007Q3	211.16	2007Q3
2007Q4	212.37	2007Q4
2008Q1	213.96	2008Q1

```
[193 rows x 2 columns]
```

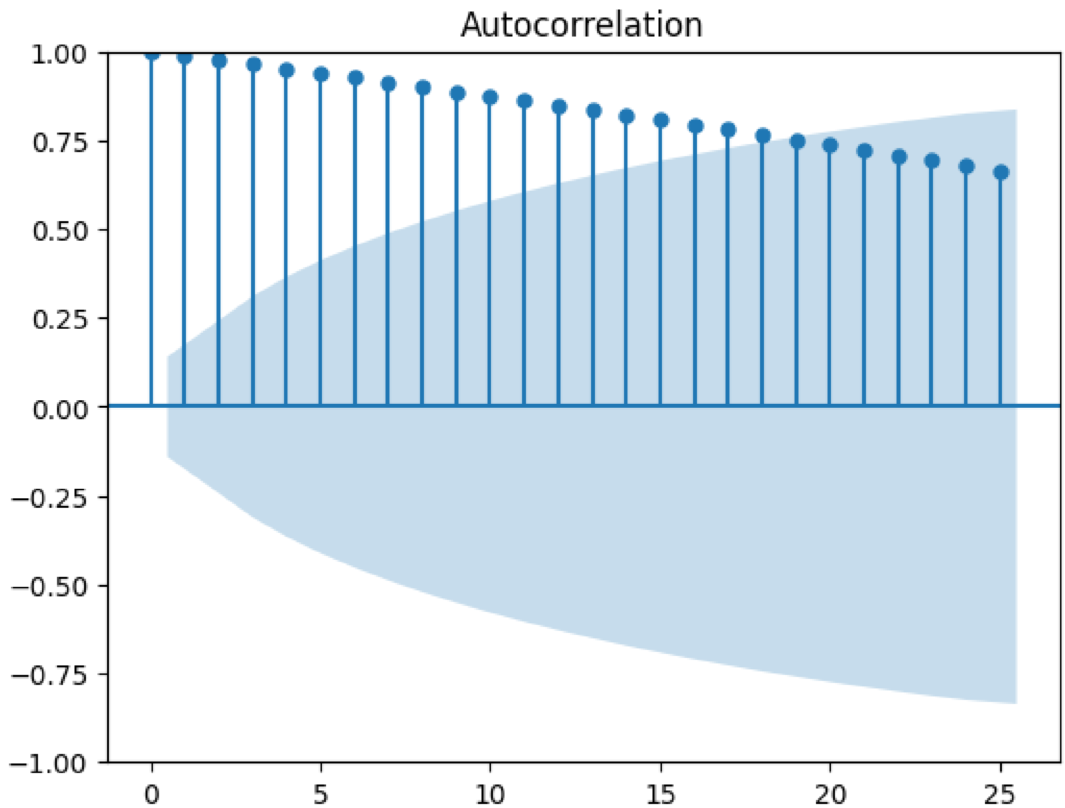
[23]:
```python
plt.figure(figsize=(8, 5))
plt.plot(df.index.to_timestamp(), df['CPINSA'], marker='o')
plt.title('Quarterly Data')
plt.xlabel('Date')
plt.ylabel('TB3mo')
plt.grid(True)
plt.tight_layout()
plt.show()
```

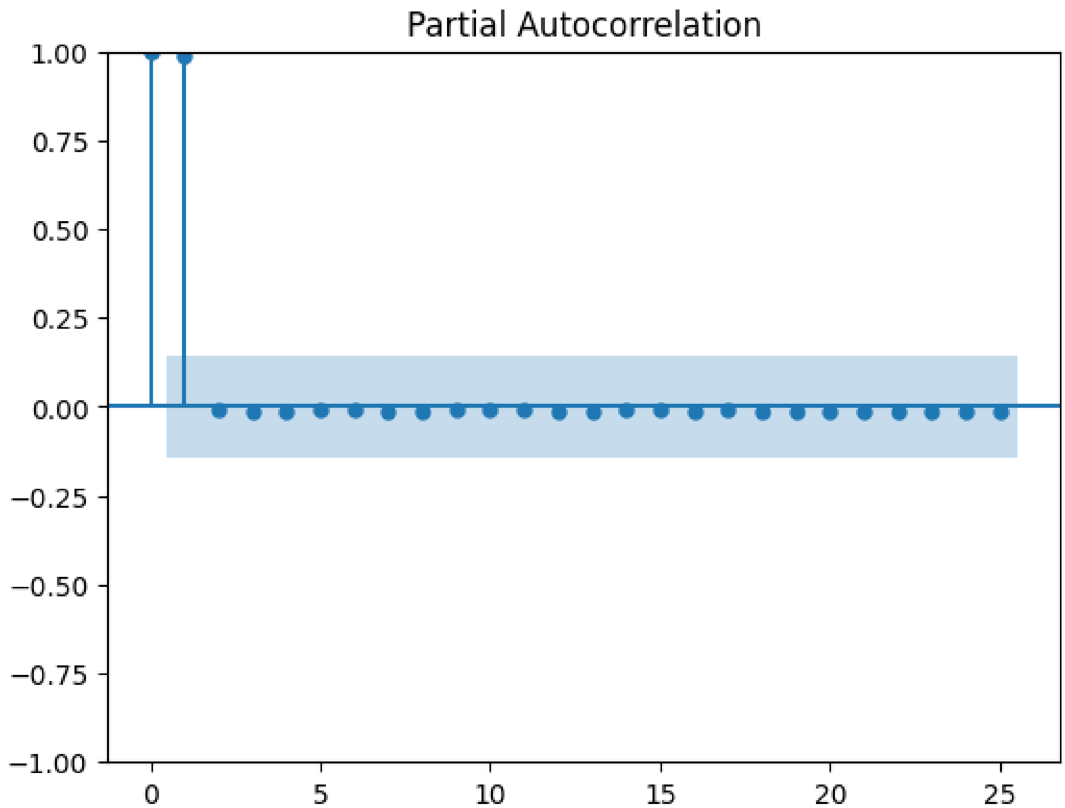


- does not look staionary

3.2 Problem B

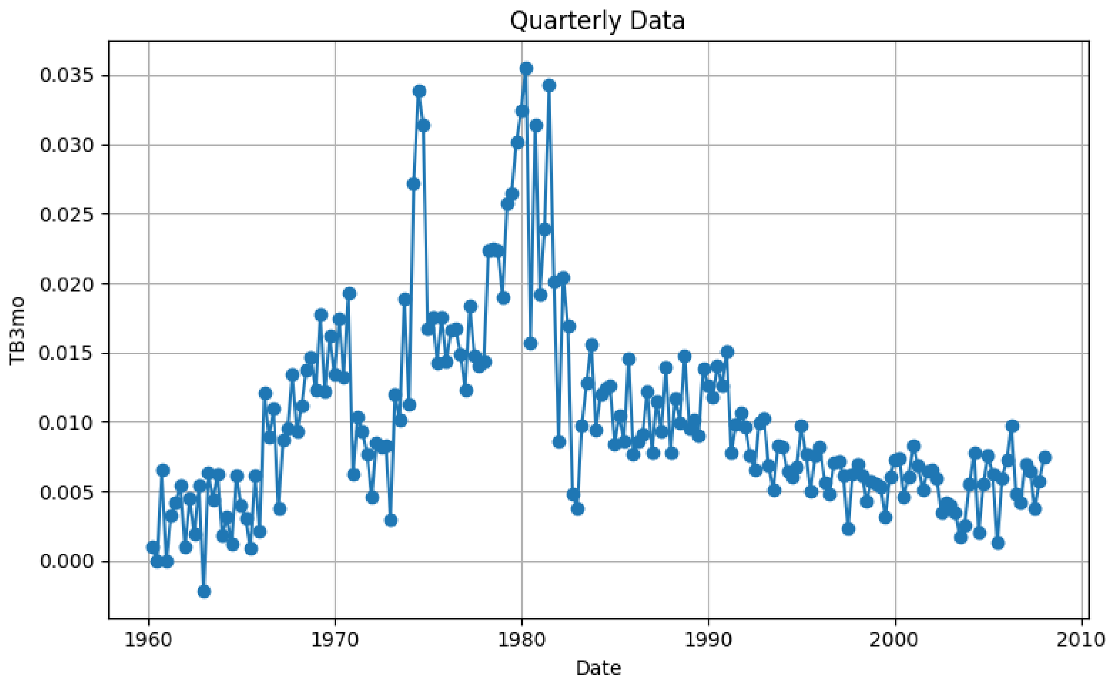
[24]:
```python
fig = plot_acf(df['CPINSA'], lags=25)
plt.show()
plot_pacf(df['CPINSA'], lags=25)
plt.show()
```


Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25



3.3 Problem C

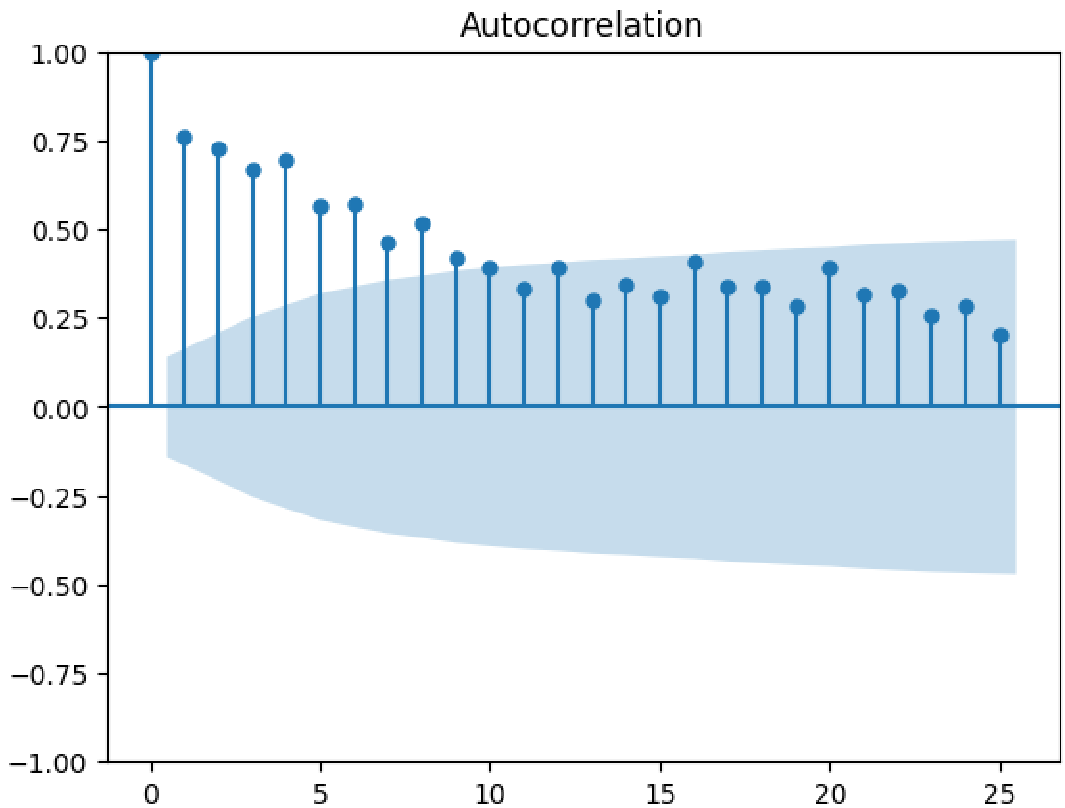
[25]:
```
df["log_CPINSA"] = np.log((df["CPINSA"] / df["CPINSA"].shift(1)))
plt.figure(figsize=(8, 5))
plt.plot(df.index.to_timestamp(), df['log_CPINSA'], marker='o')
plt.title('Quarterly Data')
plt.xlabel('Date')
plt.ylabel('TB3mo')
plt.grid(True)
plt.tight_layout()
plt.show()
```

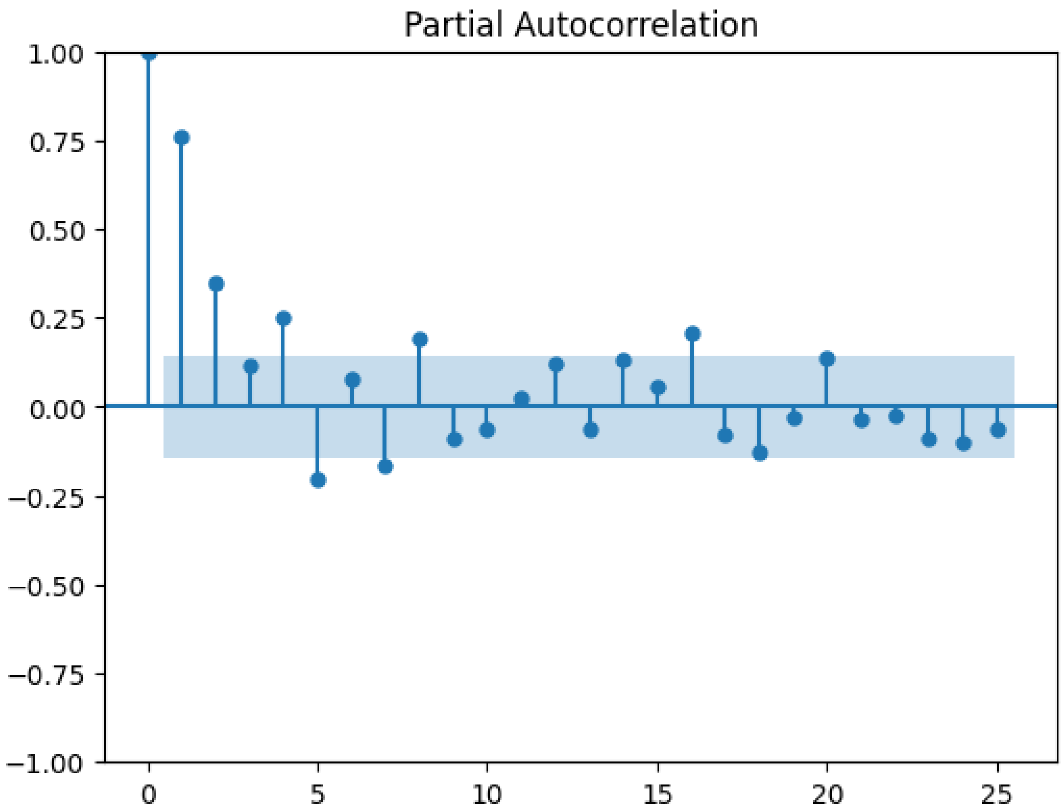


- There was a dip but it seems to be stationary given that there is no strong trend

3.4 Problem D

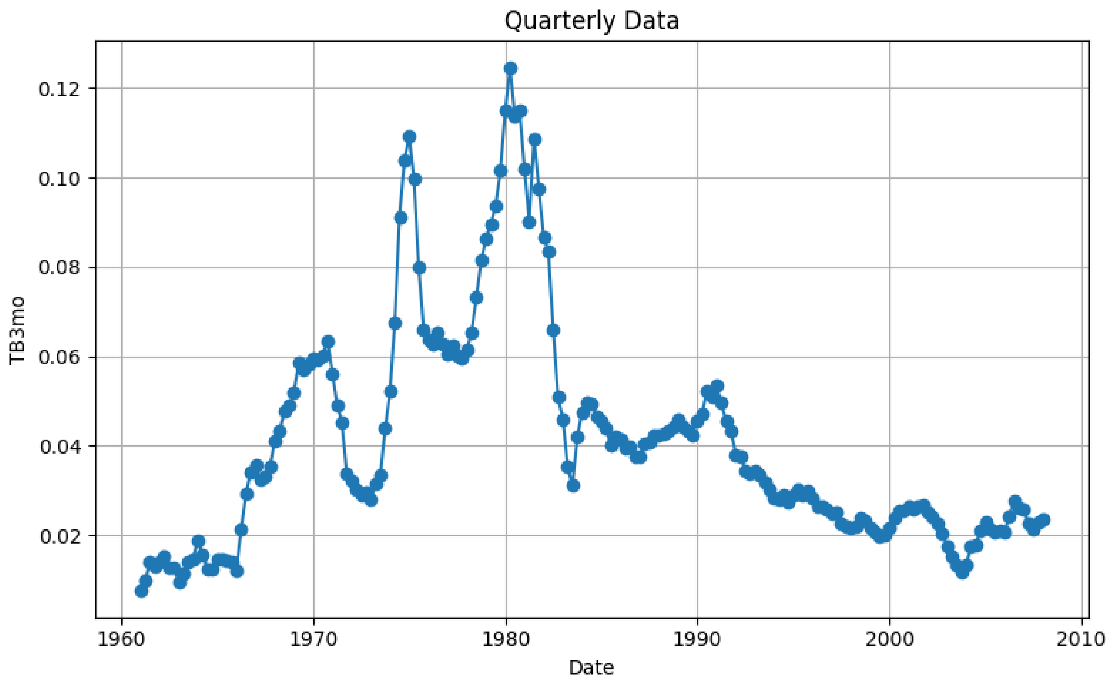
[26]:
```
fig = plot_acf(df['log_CPINSA'].dropna(), lags=25)
plt.show()
plot_pacf(df['log_CPINSA'].dropna(), lags=25)
plt.show()
```


Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25



3.5 Problem E

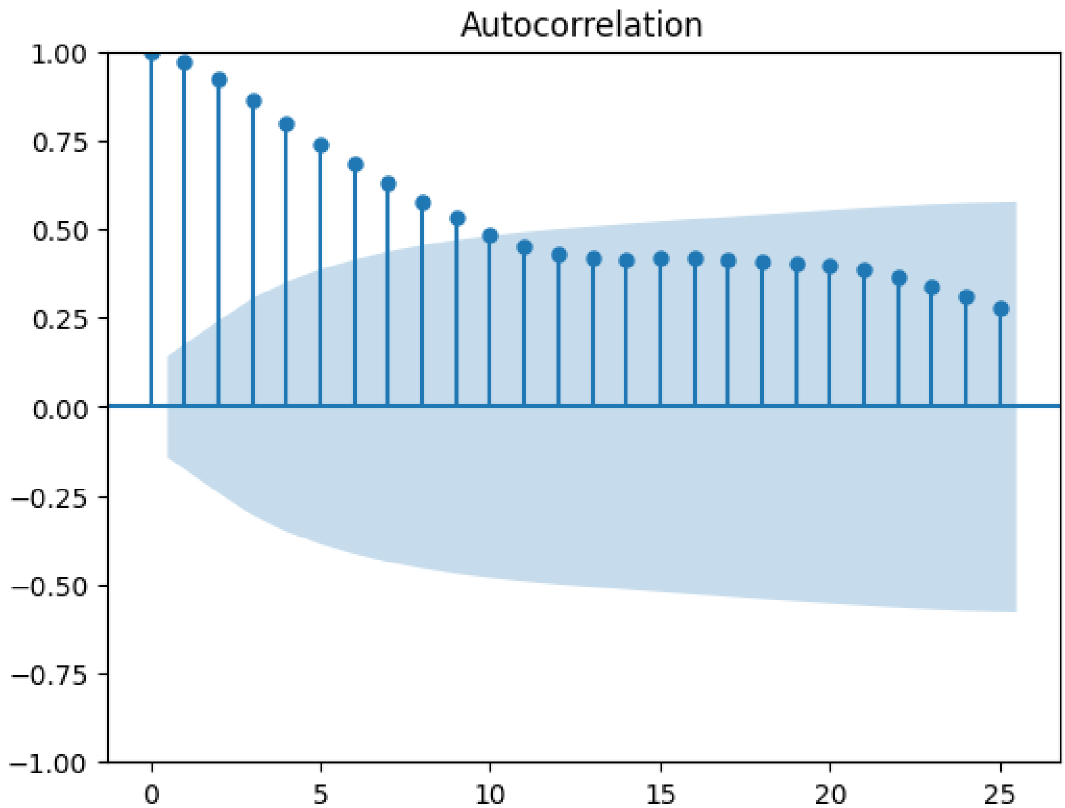
[27]:
```
df["log_CPINSA4"] = np.log((df["CPINSA"] / df["CPINSA"].shift(4)))
plt.figure(figsize=(8, 5))
plt.plot(df.index.to_timestamp(), df['log_CPINSA4'], marker='o')
plt.title('Quarterly Data')
plt.xlabel('Date')
plt.ylabel('TB3mo')
plt.grid(True)
plt.tight_layout()
plt.show()
```

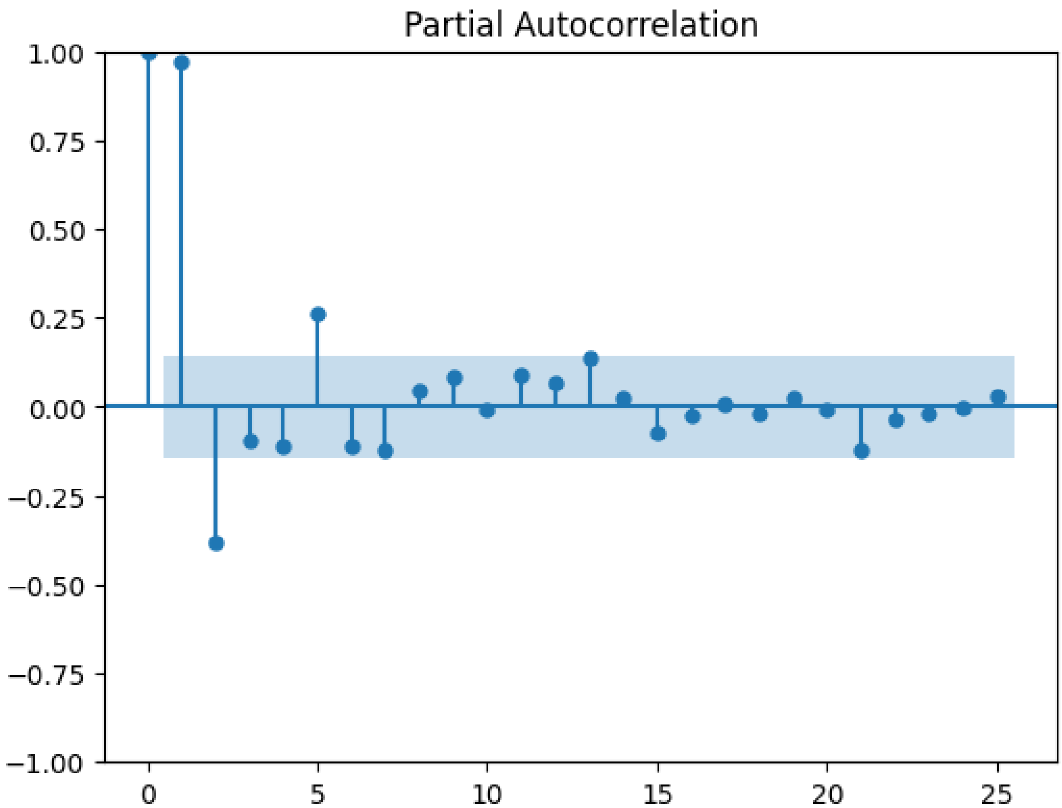


- there was a dip but it seems to be stationary given that there is no strong trend

3.6 Problem F

[28]:
```
fig = plot_acf(df['log_CPINSA4'].dropna(), lags=25)
plt.show()
fig = plot_pacf(df['log_CPINSA4'].dropna(), lags=25)
plt.show()
```


Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25



3.7 Problem G

[29]:
```
res = AutoReg(df['log_CPINSA4'].dropna(), lags =5).fit()
print(res.summary())
```

```
                            AutoReg Model Results
==============================================================================
Dep. Variable:            log_CPINSA4   No. Observations:                  189
Model:                     AutoReg(5)   Log Likelihood                 743.312
Method:               Conditional MLE   S.D. of innovations              0.004
Date:                Tue, 15 Jul 2025   AIC                          -1472.624
Time:                        21:30:15   BIC                          -1450.120
Sample:                    06-30-1962   HQIC                         -1463.503
                         - 03-31-2008
================================================================================
==
                     coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
--
const              0.0012      0.001      1.826      0.068   -8.5e-05
```

```
0.002
log_CPINSA4.L1     1.4432      0.068     21.189      0.000       1.310
1.577
log_CPINSA4.L2    -0.4339      0.117     -3.700      0.000      -0.664
-0.204
log_CPINSA4.L3     0.2177      0.120      1.811      0.070      -0.018
0.453
log_CPINSA4.L4    -0.6346      0.117     -5.425      0.000      -0.864
-0.405
log_CPINSA4.L5     0.3809      0.068      5.621      0.000       0.248
0.514
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1           -0.7595           -1.0363j            1.2848           -0.3507
AR.2           -0.7595           +1.0363j            1.2848            0.3507
AR.3            1.0440           -0.0000j            1.0440           -0.0000
AR.4            1.0704           -0.6143j            1.2342           -0.0829
AR.5            1.0704           +0.6143j            1.2342            0.0829
-----------------------------------------------------------------------------
```

[30]:

```python
y_true = res.model.endog[res.model._hold_back:]
y_pred = res.fittedvalues


ssr = np.sum((y_true - y_pred) ** 2)

tss = np.sum((y_true - np.mean(y_true)) ** 2)

n = len(y_true)
k = res.df_model + 1

r2 = 1 - ssr / tss
r2_adj = 1 - (ssr / (n - k)) / (tss / (n - 1))

print("R^2:", round(r2, 4))
print("Adjusted R^2:", round(r2_adj, 4))
```

```
R^2: 0.9708
Adjusted R^2: 0.9698
```

[31]:

```python
arma_mod = ARIMA(df['log_CPINSA4'].dropna(), order=(0, 0, 10)).fit()
print(arma_mod.summary())
```

```
/home/ouslan/Documents/Github/ECON-124/.venv/lib/python3.10/site-
packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible
starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:            log_CPINSA4   No. Observations:                  189
Model:                ARIMA(0, 0, 10)   Log Likelihood                 768.413
Date:                Tue, 15 Jul 2025   AIC                          -1512.826
Time:                        21:30:16   BIC                          -1473.925
Sample:                    03-31-1961   HQIC                         -1497.066
                         - 03-31-2008
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0395      0.005      7.731      0.000       0.029       0.050
ma.L1          1.5961      0.055     28.853      0.000       1.488       1.704
ma.L2          1.9248      0.116     16.662      0.000       1.698       2.151
ma.L3          2.2935      0.169     13.555      0.000       1.962       2.625
ma.L4          1.9670      0.200      9.857      0.000       1.576       2.358
ma.L5          1.6215      0.219      7.396      0.000       1.192       2.051
ma.L6          1.5860      0.229      6.936      0.000       1.138       2.034
ma.L7          1.2069      0.219      5.523      0.000       0.779       1.635
ma.L8          0.7927      0.171      4.638      0.000       0.458       1.128
ma.L9          0.5756      0.123      4.665      0.000       0.334       0.817
ma.L10         0.1923      0.071      2.693      0.007       0.052       0.332
sigma2      1.671e-05   1.62e-06     10.303      0.000    1.35e-05    1.99e-05
================================================================================
===
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):
156.44
Prob(Q):                              0.99   Prob(JB):
0.00
Heteroskedasticity (H):               0.16   Skew:
1.17
Prob(H) (two-sided):                  0.00   Kurtosis:
6.80
================================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
```

```
/home/ouslan/Documents/Github/ECON-124/.venv/lib/python3.10/site-
packages/statsmodels/base/model.py:607: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

[32]:
```
arma_mod = ARIMA(df['log_CPINSA4'].dropna(), order=(5, 0, 10)).fit()
print(arma_mod.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:            log_CPINSA4   No. Observations:                  189
Model:                ARIMA(5, 0, 10)   Log Likelihood                 791.966
Date:                Tue, 15 Jul 2025   AIC                          -1549.932
Time:                        21:30:16   BIC                          -1494.822
Sample:                    03-31-1961   HQIC                         -1527.605
                         - 03-31-2008
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0375      0.012      3.003      0.003       0.013       0.062
ar.L1          1.5931      0.567      2.808      0.005       0.481       2.705
ar.L2         -0.5906      0.845     -0.699      0.485      -2.247       1.066
ar.L3          0.0890      0.648      0.137      0.891      -1.180       1.358
ar.L4         -0.4775      0.718     -0.666      0.506      -1.884       0.929
ar.L5          0.3435      0.348      0.987      0.324      -0.338       1.025
ma.L1          0.0075      0.576      0.013      0.990      -1.121       1.136
ma.L2          0.0831      0.409      0.203      0.839      -0.719       0.886
ma.L3          0.1381      0.431      0.321      0.748      -0.706       0.982
ma.L4         -0.4304      0.456     -0.944      0.345      -1.324       0.463
ma.L5          0.3823      0.285      1.339      0.181      -0.177       0.942
ma.L6          0.3555      0.312      1.141      0.254      -0.255       0.966
ma.L7          0.1501      0.403      0.372      0.710      -0.641       0.941
ma.L8         -0.0303      0.186     -0.163      0.871      -0.395       0.334
ma.L9          0.0529      0.143      0.371      0.711      -0.227       0.333
ma.L10        -0.1239      0.170     -0.730      0.465      -0.457       0.209
sigma2      1.277e-05   9.72e-07     13.134      0.000    1.09e-05    1.47e-05
================================================================================
===
Ljung-Box (L1) (Q):                   0.05   Jarque-Bera (JB):
117.79
Prob(Q):                              0.82   Prob(JB):
0.00
Heteroskedasticity (H):               0.16   Skew:
0.14
Prob(H) (two-sided):                  0.00   Kurtosis:
6.86
================================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
```

[33]:
```python
arma_mod = ARIMA(df['log_CPINSA4'].dropna(), order=(6, 0, 7)).fit()
print(arma_mod.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:            log_CPINSA4   No. Observations:                  189
Model:                 ARIMA(6, 0, 7)   Log Likelihood                 791.248
Date:                Tue, 15 Jul 2025   AIC                          -1552.496
Time:                        21:30:17   BIC                          -1503.870
Sample:                    03-31-1961   HQIC                         -1532.796
                         - 03-31-2008
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0383      0.011      3.337      0.001       0.016       0.061
ar.L1          1.3653      0.955      1.429      0.153      -0.507       3.238
ar.L2         -0.3581      1.621     -0.221      0.825      -3.536       2.820
ar.L3          0.3838      1.052      0.365      0.715      -1.678       2.446
ar.L4         -0.7411      0.380     -1.949      0.051      -1.486       0.004
ar.L5          0.2318      0.414      0.560      0.576      -0.580       1.044
ar.L6          0.0712      0.318      0.224      0.823      -0.553       0.695
ma.L1          0.2100      0.970      0.216      0.829      -1.692       2.112
ma.L2          0.2335      0.854      0.273      0.785      -1.441       1.908
ma.L3         -0.0672      0.430     -0.156      0.876      -0.909       0.775
ma.L4         -0.4725      0.371     -1.275      0.202      -1.199       0.254
ma.L5          0.1967      0.654      0.301      0.764      -1.085       1.478
ma.L6          0.1879      0.584      0.322      0.747      -0.956       1.332
ma.L7          0.2809      0.280      1.004      0.315      -0.268       0.829
sigma2      1.304e-05   1.07e-06     12.198      0.000    1.09e-05    1.51e-05
================================================================================
===
Ljung-Box (L1) (Q):                   0.04   Jarque-Bera (JB):
133.37
Prob(Q):                              0.84   Prob(JB):
0.00
Heteroskedasticity (H):               0.14   Skew:
0.13
Prob(H) (two-sided):                  0.00   Kurtosis:
7.11
================================================================================
===

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-
step).
```

```
/home/ouslan/Documents/Github/ECON-124/.venv/lib/python3.10/site-
```

```
packages/statsmodels/base/model.py:607: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

3.8 Problem H

- looking at the AIC the MA(10) seems to prefrom the best

3.9 Problem I

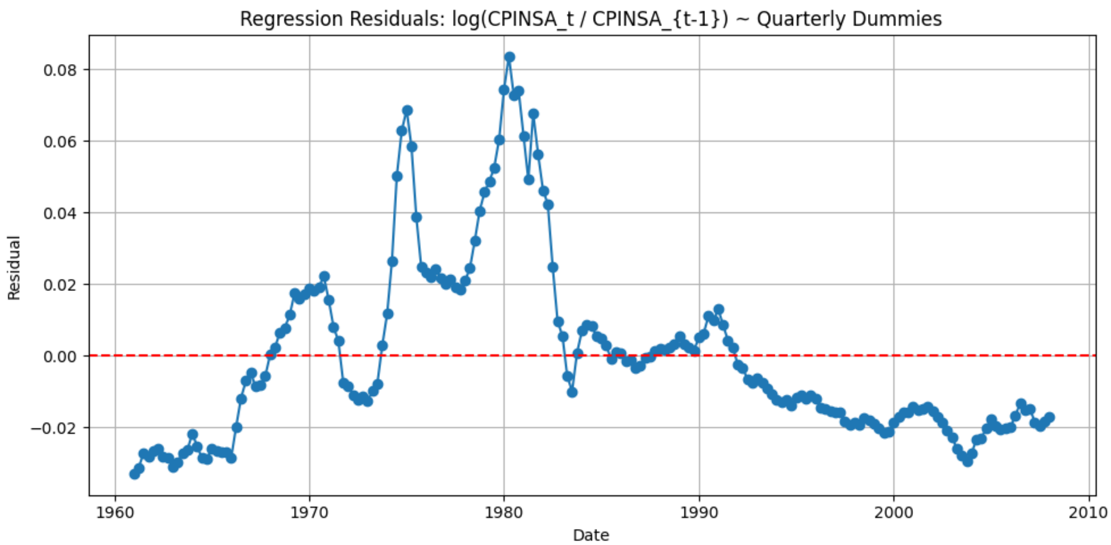
[34]:
```python
df["quarter"] = df.index.quarter
quarter_dummies = pd.get_dummies(df["quarter"], prefix="Q", drop_first=True).
 ↪astype(int)
df = pd.concat([df, quarter_dummies], axis=1)
X = df[[col for col in df.columns if col.startswith("Q_")]]
X = sm.add_constant(X)
y = df["log_CPINSA4"]

model = sm.OLS(y, X, missing='drop').fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            log_CPINSA4   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.016
Method:                 Least Squares   F-statistic:                  0.005292
Date:                Tue, 15 Jul 2025   Prob (F-statistic):              0.999
Time:                        21:30:17   Log-Likelihood:                 428.53
No. Observations:                 189   AIC:                            -849.1
Df Residuals:                     185   BIC:                            -836.1
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0405      0.004     11.086      0.000       0.033       0.048
Q_2            0.0005      0.005      0.093      0.926      -0.010       0.011
Q_3            0.0006      0.005      0.108      0.914      -0.010       0.011
Q_4            0.0005      0.005      0.105      0.917      -0.010       0.011
==============================================================================
Omnibus:                       42.124   Durbin-Watson:                   0.045
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               62.900
Skew:                           1.275   Prob(JB):                     2.19e-14
Kurtosis:                       4.218   Cond. No.                         4.76
==============================================================================

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

specified.

3.10 Problem J

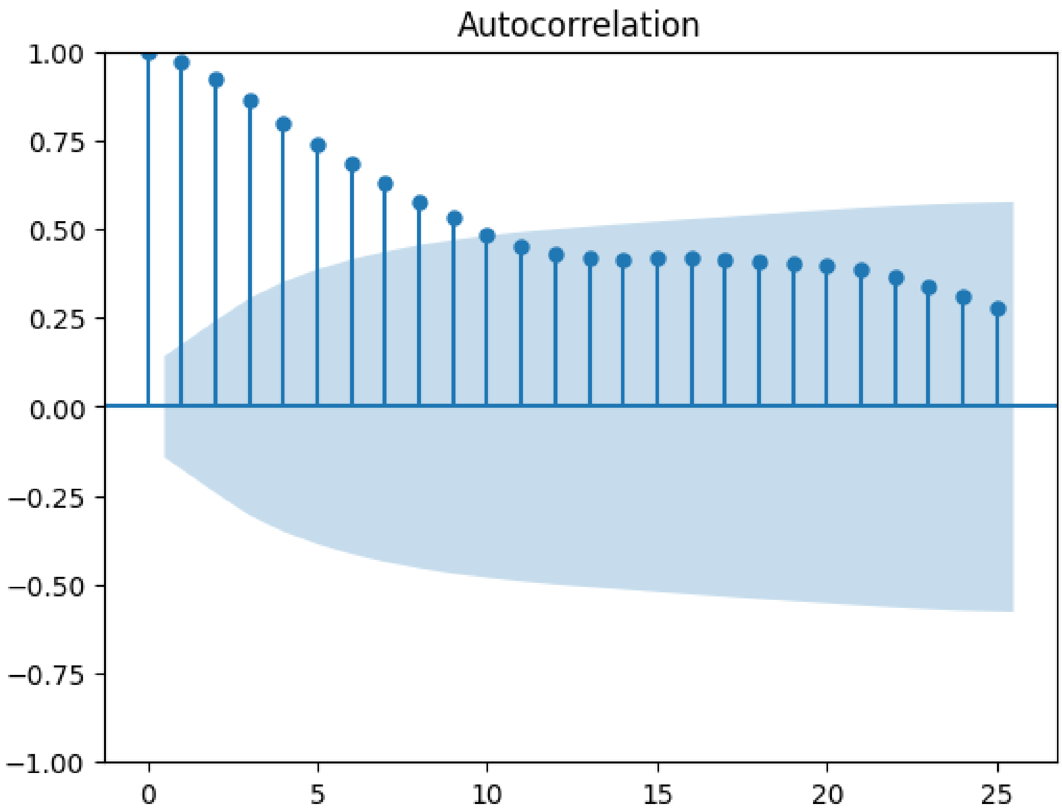
[35]:
```
df["residuals"] = model.resid
plt.figure(figsize=(10, 5))
plt.plot(df.index.to_timestamp(), df["residuals"], marker='o', linestyle='-')
plt.axhline(0, color='red', linestyle='--')
plt.title("Regression Residuals: log(CPINSA_t / CPINSA_{t-1}) ~ Quarterly␣
↪Dummies")
plt.xlabel("Date")
plt.ylabel("Residual")
plt.grid(True)
plt.tight_layout()
plt.show()
```



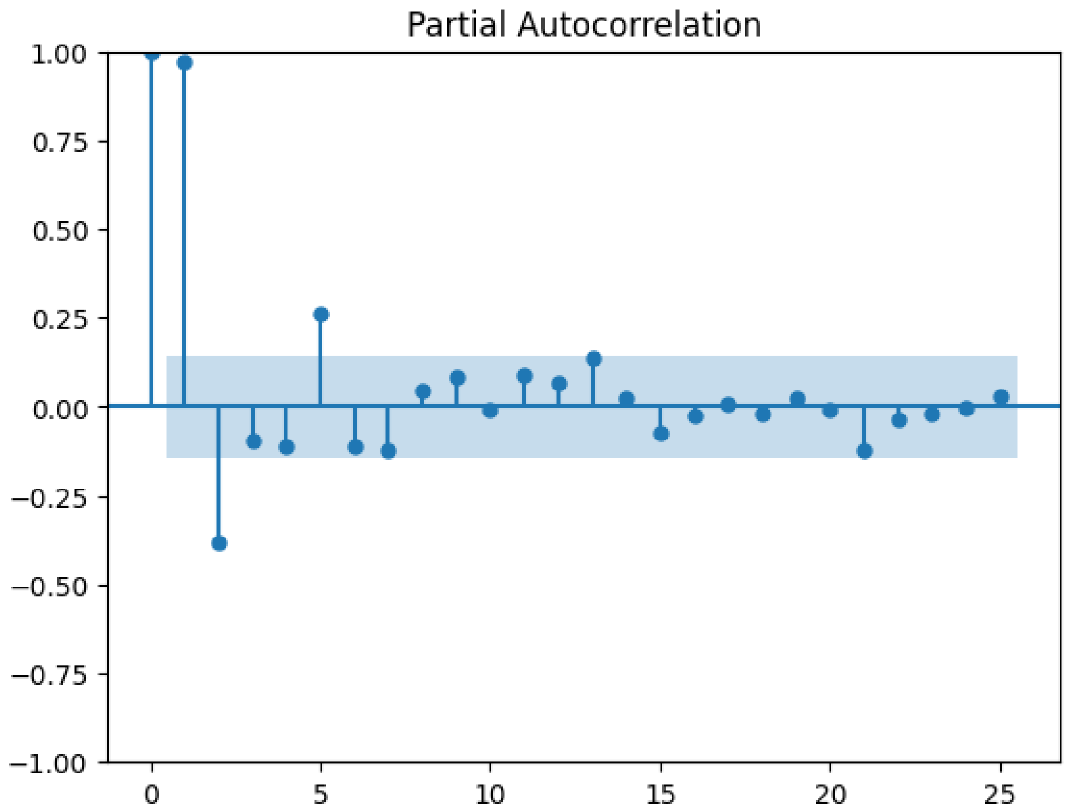
- the residuals apear to be staionary

3.11 Problem K

[36]:
```
fig = plot_acf(df['log_CPINSA4'].dropna(), lags=25)
plt.show()
fig = plot_pacf(df['log_CPINSA4'].dropna(), lags=25)
plt.show()
```



Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25



- the residuals seam to be autocorrolated by round 10 periods

4 Problem

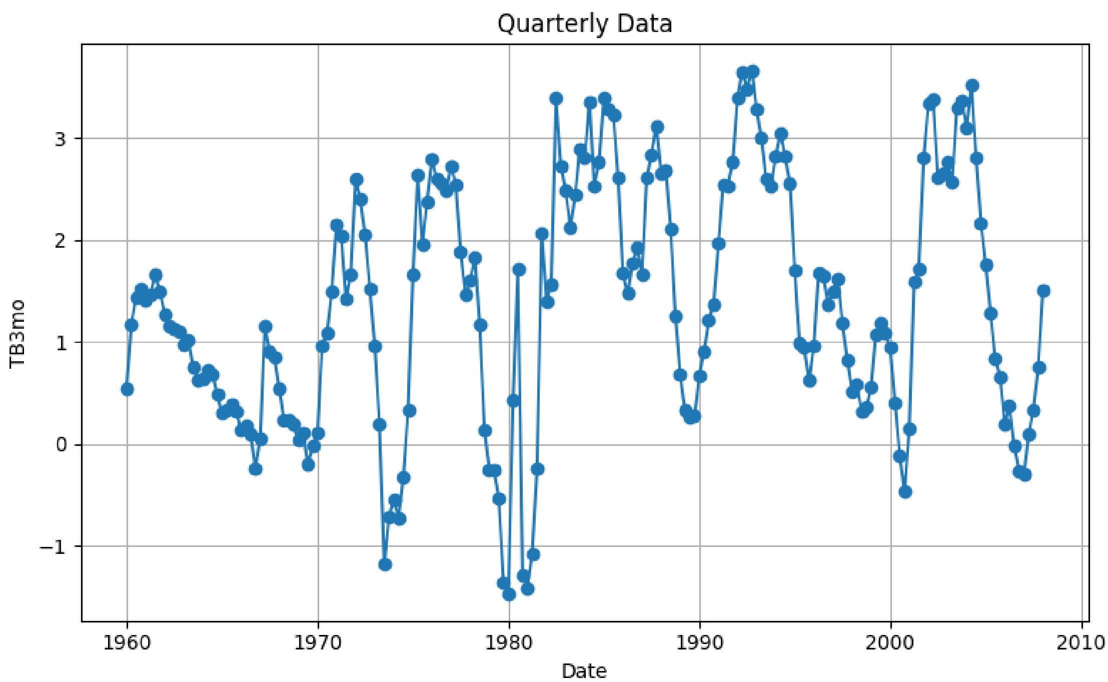
4.1 Problem A

[37]:
```
df = pd.read_excel("data/QUARTERLY-1.xls")
df['date'] = pd.PeriodIndex(df['Date'], freq='Q')
df.set_index('date', inplace=True)
# df = df[["CPINSA","Date"]]
df["s"] = df["r10"] - df["Tbill"]
df = df[["s","Date"]]
df
```

[37]:
```
              s    Date
date
1960Q1  0.54334  1960Q1
1960Q2  1.17000  1960Q2
1960Q3  1.44000  1960Q3
1960Q4  1.52667  1960Q4
1961Q1  1.41000  1961Q1
```

```
…        …      …
2007Q1 -0.30000  2007Q1
2007Q2  0.09000  2007Q2
2007Q3  0.33667  2007Q3
2007Q4  0.75000  2007Q4
2008Q1  1.51333  2008Q1

[193 rows x 2 columns]
```

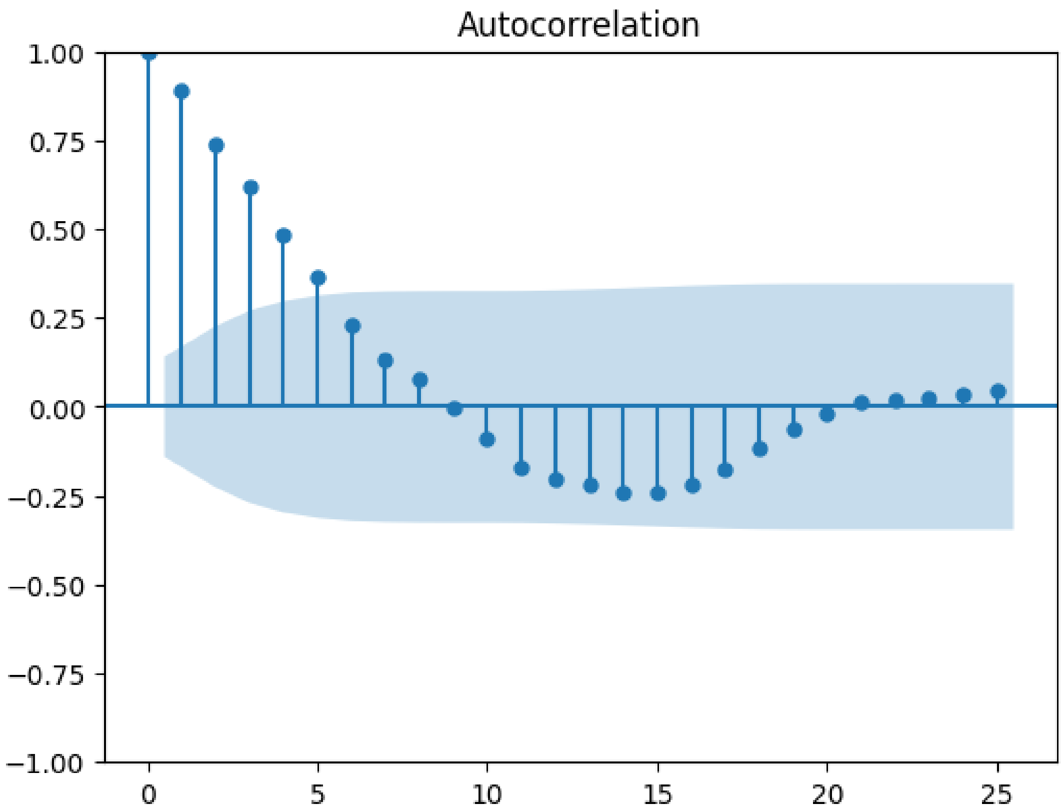
[38]:
```python
plt.figure(figsize=(8, 5))
plt.plot(df.index.to_timestamp(), df['s'], marker='o')
plt.title('Quarterly Data')
plt.xlabel('Date')
plt.ylabel('TB3mo')
plt.grid(True)
plt.tight_layout()
plt.show()
```

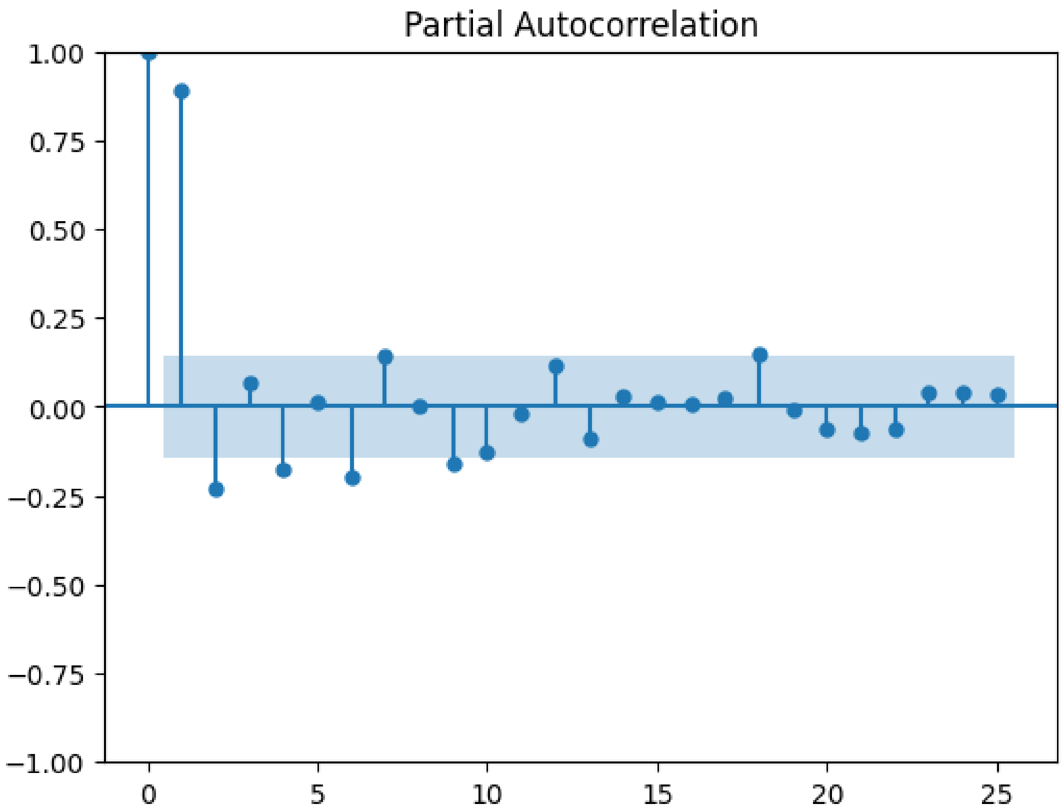


- it looks stationary

4.2 Problem B

[39]:
```
fig = plot_acf(df['s'], lags=25)
plt.show()
plot_pacf(df['s'], lags=25)
plt.show()
```



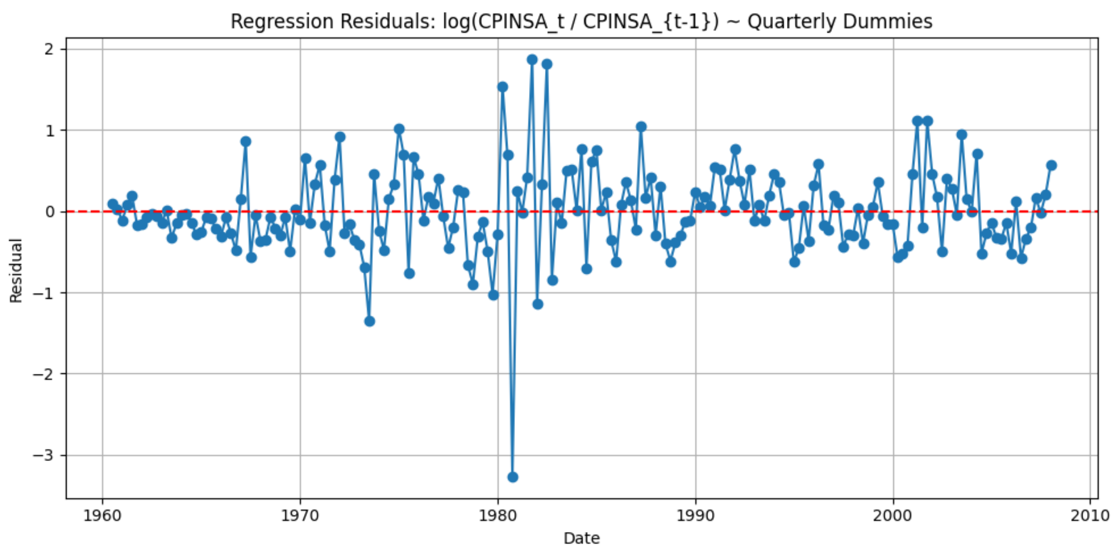


- it seams to be autocorrolated by 4 units and partial autocorrelated by 1

[40]:
```
res_ar2 = AutoReg(df['s'].dropna(), lags =2).fit()
print(res_ar2.summary())
df["residuals"] = res_ar2.resid
plt.figure(figsize=(10, 5))
plt.plot(df.index.to_timestamp(), df["residuals"], marker='o', linestyle='-')
plt.axhline(0, color='red', linestyle='--')
plt.title("Regression Residuals: log(CPINSA_t / CPINSA_{t-1}) ~ Quarterly␣
  ↪Dummies")
plt.xlabel("Date")
plt.ylabel("Residual")
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
                            AutoReg Model Results
==============================================================================
Dep. Variable:                      s   No. Observations:                  193
Model:                     AutoReg(2)   Log Likelihood                -150.856
Method:               Conditional MLE   S.D. of innovations              0.533
Date:                Tue, 15 Jul 2025   AIC                            309.713
```

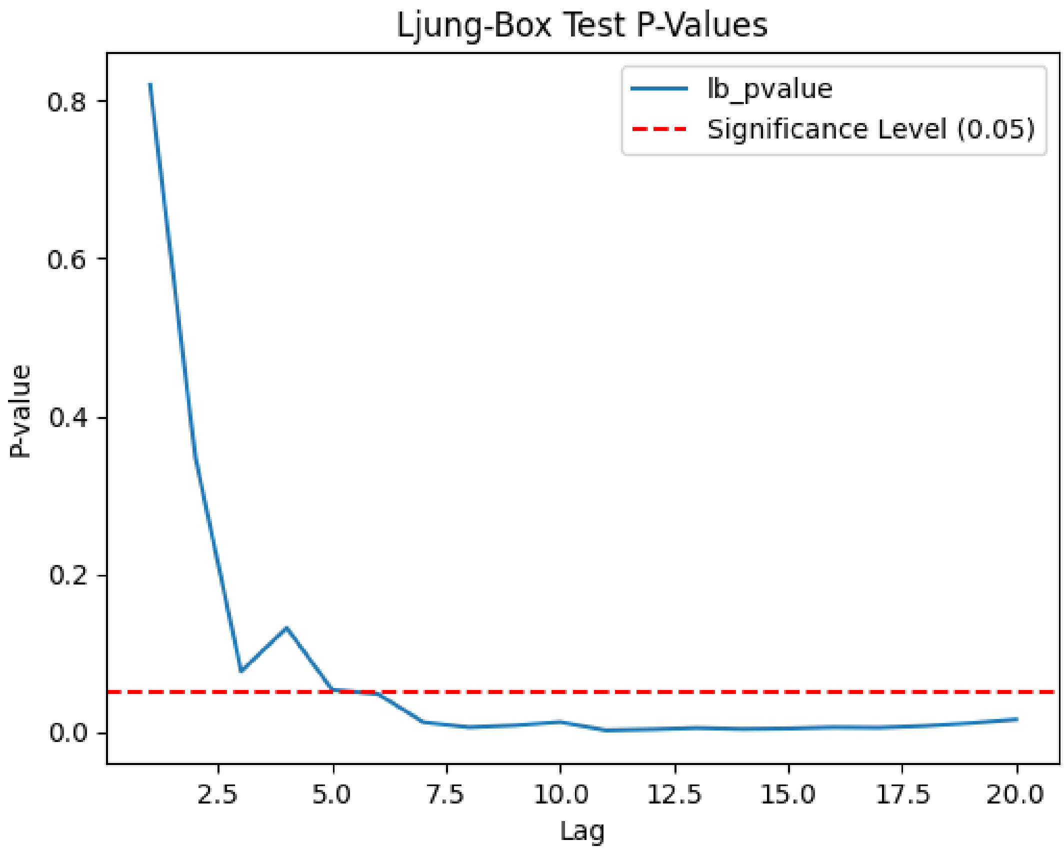
```
Time:                        21:30:17   BIC                            322.722
Sample:                    09-30-1960   HQIC                           314.982
                         - 03-31-2008
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.1892      0.059      3.185      0.001       0.073       0.306
s.L1           1.1086      0.070     15.775      0.000       0.971       1.246
s.L2          -0.2450      0.070     -3.488      0.000      -0.383      -0.107
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.2440           +0.0000j            1.2440            0.0000
AR.2            3.2815           +0.0000j            3.2815            0.0000
-----------------------------------------------------------------------------
```



[41]:
```python
residuals = res_ar2.resid

ljung_box_results = acorr_ljungbox(residuals, lags=range(1, 21), return_df=True)

print(ljung_box_results)

# Optional: Plot p-values to visualize autocorrelation
ljung_box_results['lb_pvalue'].plot(title='Ljung-Box Test P-Values')
plt.axhline(y=0.05, color='red', linestyle='--', label='Significance Level (0.
↪05)')
plt.xlabel('Lag')
plt.ylabel('P-value')
```

```
plt.legend()
plt.show()
```

```
     lb_stat  lb_pvalue
1    0.051333   0.820760
2    2.117369   0.346912
3    6.858840   0.076536
4    7.078838   0.131780
5   10.925209   0.052883
6   12.709362   0.047891
7   17.987541   0.012026
8   21.684678   0.005535
9   22.297068   0.007984
10  22.618691   0.012245
11  29.728266   0.001748
12  29.835833   0.002956
13  30.007268   0.004698
14  32.537869   0.003357
15  33.486681   0.004018
16  33.949568   0.005519
17  35.670142   0.005074
18  35.869956   0.007332
19  35.901840   0.010852
20  35.957500   0.015559
```



- there seem to be autocorroration for periods after 5

4.3 Problem E

[42]:
```python
res_ar7 = AutoReg(df['s'].dropna(), lags =7).fit()
print(res_ar7.summary())
```

```
                            AutoReg Model Results
==============================================================================
Dep. Variable:                      s   No. Observations:                  193
Model:                     AutoReg(7)   Log Likelihood                -139.381
Method:               Conditional MLE   S.D. of innovations              0.512
Date:                Tue, 15 Jul 2025   AIC                            296.763
Time:                        21:30:18   BIC                            325.794
Sample:                    12-31-1961   HQIC                           308.527
                         - 03-31-2008
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.2104      0.067      3.152      0.002       0.080       0.341
s.L1           1.1768      0.073     16.192      0.000       1.034       1.319
```

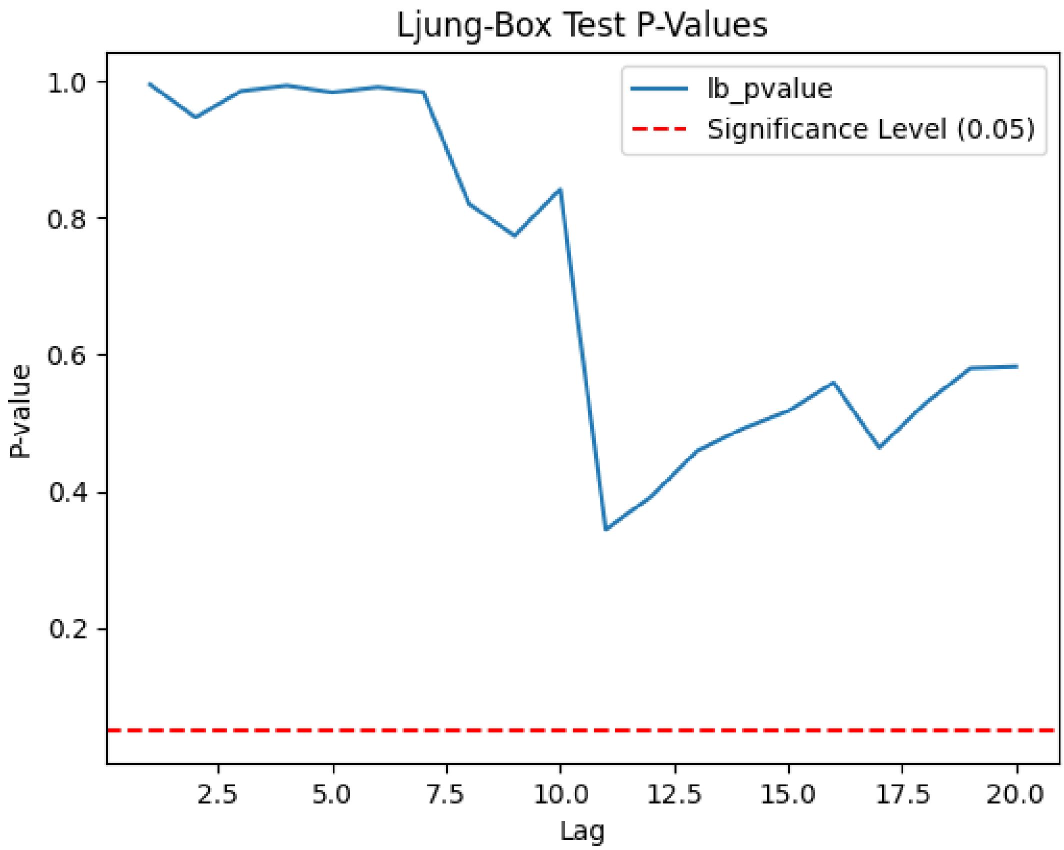
```
s.L2          -0.4658      0.109     -4.258      0.000      -0.680      -0.251
s.L3           0.3861      0.112      3.443      0.001       0.166       0.606
s.L4          -0.3386      0.113     -2.996      0.003      -0.560      -0.117
s.L5           0.3188      0.112      2.840      0.005       0.099       0.539
s.L6          -0.3791      0.109     -3.466      0.001      -0.593      -0.165
s.L7           0.1504      0.073      2.066      0.039       0.008       0.293
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1           -0.9208          -0.8352j            1.2432           -0.3828
AR.2           -0.9208          +0.8352j            1.2432            0.3828
AR.3            0.1872          -1.2949j            1.3083           -0.2271
AR.4            0.1872          +1.2949j            1.3083            0.2271
AR.5            1.2281          -0.3642j            1.2810           -0.0459
AR.6            1.2281          +0.3642j            1.2810            0.0459
AR.7            1.5317          -0.0000j            1.5317           -0.0000
-----------------------------------------------------------------------------
```

4.4 Problem F

[43]:
```python
residuals = res_ar7.resid

ljung_box_results = acorr_ljungbox(residuals, lags=range(1, 21), return_df=True)


print(ljung_box_results)

ljung_box_results['lb_pvalue'].plot(title='Ljung-Box Test P-Values')
plt.axhline(y=0.05, color='red', linestyle='--', label='Significance Level (0.
 ↪05)')
plt.xlabel('Lag')
plt.ylabel('P-value')
plt.legend()
plt.show()
```

```
      lb_stat  lb_pvalue
1    0.000031   0.995573
2    0.107644   0.947601
3    0.145260   0.985901
4    0.228593   0.993945
5    0.681262   0.983981
6    0.813668   0.991702
7    1.444265   0.984177
8    4.386175   0.820710
9    5.651757   0.774202
10   5.669816   0.842199
11  12.263550   0.344163
```

```
12  12.683548   0.392455
13  12.855834   0.459008
14  13.448330   0.491565
15  14.105729   0.517526
16  14.532605   0.559100
17  16.863113   0.463677
18  16.924938   0.528271
19  17.146294   0.579957
20  18.076740   0.582353
```



- there does not seem to be autoccoeration

4.5 Problem G

[44]:
```python
print(f"AIC AR(2): {res_ar2.aic}")
print(f"AIC AR(7): {res_ar7.aic}")

print(f"BIC AR(2): {res_ar2.bic}")
print(f"BIC AR(7): {res_ar7.bic}")
```

```
AIC AR(2): 309.7126128947031
```

```
AIC AR(7): 296.76261348236477
BIC AR(2): 322.72170660688965
BIC AR(7): 325.7943335457836
```

4.6 Problem H

[45]:
```python
df2 = df.head(-10)
y = df2['s'].dropna()

forecast_ar2 = res_ar2.predict(start=res_ar2.model._hold_back, end=len(y)-1)

error_ar2 = y[res_ar2.model._hold_back:] - forecast_ar2
error_ar2
```

[45]:
```
date
1960Q3    0.086837
1960Q4    0.027696
1961Q1   -0.118914
1961Q2    0.084988
1961Q3    0.190625
            …
2004Q3   -0.521260
2004Q4   -0.272081
2005Q1   -0.143164
2005Q2   -0.322460
2005Q3   -0.347333
Freq: Q-DEC, Length: 181, dtype: float64
```

[46]:
```python
y = df2['s'].dropna()

res_ar7 = AutoReg(df2['s'].dropna(), lags=7).fit()


forecast_ar7 = res_ar7.predict(start=res_ar7.model._hold_back, end=len(y)-1)

error_ar7 = y[res_ar7.model._hold_back:] - forecast_ar7
error_ar7
```

[46]:
```
date
1961Q4   -0.096905
1962Q1   -0.124619
1962Q2   -0.089512
1962Q3   -0.030522
1962Q4   -0.080839
            …
2004Q3   -0.449768
2004Q4   -0.268528
2005Q1   -0.218167
```

```
2005Q2   -0.097570
2005Q3   -0.433046
Freq: Q-DEC, Length: 176, dtype: float64
```

[47]:
```
mse_ar2 = (error_ar2**2).mean()
mse_ar7 = (error_ar7**2).mean()

print(f'MSE AR(2): {mse_ar2:.4f}')
print(f'MSE AR(7): {mse_ar7:.4f}')
```

```
MSE AR(2): 0.2933
MSE AR(7): 0.2697
```

- The AR(7) seems to have a smaller forecst error than AR(2)

4.7 Problem I

[48]:
```
forecast_ar2 = res_ar2.predict(start=len(y), end=len(y)+9)

# Forecast error
error_ar2 = df['s'].tail(10) - forecast_ar2
error_ar2
```

[48]:
```
date
2005Q4   -0.144416
2006Q1   -0.518433
2006Q2    0.124451
2006Q3   -0.574535
2006Q4   -0.336027
2007Q1   -0.202987
2007Q2    0.168873
2007Q3   -0.025795
2007Q4    0.209614
2008Q1    0.575152
Freq: Q-DEC, dtype: float64
```

[49]:
```
forecast_ar7 = res_ar7.predict(start=len(y), end=len(y)+9)

# Forecast error
error_ar7 = df['s'].tail(10) - forecast_ar7
forecast_ar7
```

[49]:
```
2005Q4    0.585641
2006Q1    0.592219
2006Q2    0.707072
2006Q3    0.791292
2006Q4    0.896282
2007Q1    1.037073
```

```
2007Q2    1.173433
2007Q3    1.272218
2007Q4    1.326425
2008Q1    1.368948
Freq: Q-DEC, dtype: float64
```

[50]:
```
mse_ar2 = (error_ar2**2).mean()
mse_ar7 = (error_ar7**2).mean()

print(f'MSE AR(2): {mse_ar2:.4f}')
print(f'MSE AR(7): {mse_ar7:.4f}')
```

```
MSE AR(2): 0.1193
MSE AR(7): 0.6474
```

- the AR(2) seems to preferm better than the AR(7) at forcasting the 10 steps. This is sup-poricin ggiven that looking at the entire series the AR(7) fits better the historical data