In [1]:
```
# import packages needed
from scipy import stats
from statsmodels import tsa as TSA
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.ar_model import AR
from statsmodels.tsa.ar_model import AutoReg
from statsmodels.tsa import arima_process as ARIMA_process
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.graphics.api import qqplot
import itertools
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import statsmodels.api as sm
import warnings

warnings.filterwarnings('ignore')
```

Q5

(15%) Consider the famous time series data “co2” (monthly carbon dioxide through 11 years in Alert, Canada).
(a) Fit a deterministic regression model in terms of months and time. Are the regression coefficients significant? What is the adjusted R-squared? (Note that the month variable should be treated as categorical and transformed into 11 dummy variables.)
(b) Identify, estimate the SARIMA model for the co2 level.
(c) Compare the two models above, what do you observe?

Answers

(a) According to the result produced below, the regression coefficient is not significant, with none of them surpass the critical values.
(b) Accoring our estimate, the best fir parameters are SARIMA(0, 1, 1)x(0, 1, 1, 12)12 With AIC:251.0908756777686.
(c) The SARIMA model we constructed has a AIC level of 251, with the OLS's AIC model at 774, so we could conclude that comparing to the linear regression model, our SARIMA model does have a better performance.

In [2]:
```
data = pd.read_csv("./TSA HW07.co2.csv")

month_list = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "No

def monthNumCreator(x):
    for index, value in enumerate(month_list):
        if month_list[index] == x:
            return int(index + 1)

# data preparation
data.loc[:, 'month_num'] = data.loc[:, 'month'].apply(monthNumCreator)
```

```
print(data.head(20))
```

```
    time_trend month  co2_level  month_num
0   1994.000000   Jan     363.05          1
1   1994.083333   Feb     364.18          2
2   1994.166667   Mar     364.87          3
3   1994.250000   Apr     364.47          4
4   1994.333333   May     364.32          5
5   1994.416667   Jun     362.13          6
6   1994.500000   Jul     356.72          7
7   1994.583333   Aug     350.88          8
8   1994.666667   Sep     350.69          9
9   1994.750000   Oct     356.06         10
10  1994.833333   Nov     360.09         11
11  1994.916667   Dec     363.27         12
12  1995.000000   Jan     363.49          1
13  1995.083333   Feb     364.94          2
14  1995.166667   Mar     366.72          3
15  1995.250000   Apr     366.33          4
16  1995.333333   May     365.75          5
17  1995.416667   Jun     364.32          6
18  1995.500000   Jul     358.59          7
19  1995.583333   Aug     352.06          8
```

In [3]:

```
# fit a deterministic model

X = sm.add_constant(data[["time_trend", "month_num"]])
y = data["co2_level"]

mod = sm.OLS(y, X)

result = mod.fit()

print(result.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              co2_level   R-squared:                       0.666
Model:                            OLS   Adj. R-squared:                  0.661
Method:                 Least Squares   F-statistic:                     128.5
Date:                Sun, 06 Dec 2020   Prob (F-statistic):           1.98e-31
Time:                        21:25:41   Log-Likelihood:                -384.14
No. Observations:                 132   AIC:                             774.3
Df Residuals:                     129   BIC:                             782.9
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -3288.6451    247.245    -13.301      0.000   -3777.826   -2799.464
time_trend     1.8321      0.124     14.812      0.000       1.587       2.077
month_num     -0.8476      0.114     -7.450      0.000      -1.073      -0.622
==============================================================================
Omnibus:                        7.770   Durbin-Watson:                   0.863
Prob(Omnibus):                  0.021   Jarque-Bera (JB):                7.540
Skew:                          -0.533   Prob(JB):                       0.0231
Kurtosis:                       2.516   Cond. No.                     1.26e+06
==============================================================================

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

```
[2] The condition number is large, 1.26e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [4]:

```
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]
print('Examples of parameter for SARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))
```

```
Examples of parameter for SARIMA...
SARIMAX: (0, 0, 1) x (0, 0, 1, 12)
SARIMAX: (0, 0, 1) x (0, 1, 0, 12)
SARIMAX: (0, 1, 0) x (0, 1, 1, 12)
SARIMAX: (0, 1, 0) x (1, 0, 0, 12)
```

In [5]:

```
# estimate params of SARIMA
param_list = []
param_seasonal_list = []
aic_list = []

for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(y,order=param,seasonal_order=param_seasonal
            results = mod.fit()
#             print('ARIMA{}x{}12 - AIC:{}'.format(param,param_seasonal,results.aic))

            # append to list
            param_list.append(param)
            param_seasonal_list.append(param_seasonal)
            aic_list.append(results.aic)
        except:
            continue

# print out best params
for index, value in enumerate(aic_list):
    if value == np.min(aic_list):
        print("Best Params are")
        print('ARIMA{}x{}12 - AIC:{}'.format(param_list[index],param_seasonal_list[inde
        break;
```

```
Best Params are
ARIMA(0, 1, 1)x(0, 1, 1, 12)12 - AIC:251.0908756777686
```

In [6]:

```
# fit data with SARIMA model

model=sm.tsa.statespace.SARIMAX(endog=y,order=(0,1,1),seasonal_order=(0,1,1,12),trend='
results=model.fit()

print(results.summary())
```

```
                                    SARIMAX Results
==========================================================================================
==
Dep. Variable:                          co2_level   No. Observations:                    1
32
Model:             SARIMAX(0, 1, 1)x(0, 1, 1, 12)   Log Likelihood                  -139.5
23
Date:                            Sun, 06 Dec 2020   AIC                              287.0
47
```

```
Time:                                21:25:50   BIC                             298.1
63
Sample:                                     0   HQIC                            291.5
61
                                        - 132
Covariance Type:                          opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept      0.0021      0.010      0.215      0.830      -0.017       0.022
ma.L1         -0.5794      0.093     -6.260      0.000      -0.761      -0.398
ma.S.L12      -0.8208      0.118     -6.927      0.000      -1.053      -0.589
sigma2         0.5445      0.073      7.457      0.000       0.401       0.688
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):                 2.10
Prob(Q):                              0.93   Prob(JB):                         0.35
Heteroskedasticity (H):               1.04   Skew:                            -0.15
Prob(H) (two-sided):                  0.91   Kurtosis:                         3.58
===================================================================================

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