Import packages

In [1]:
```
import pandas as pd
import statsmodels.api as sm
from statsmodels.tsa.statespace.sarimax import SARIMAX
import matplotlib.pyplot as plt
import itertools
import warnings
warnings.filterwarnings("ignore")
```

Function definition

-> S.ARIMA model gerneration

In [2]:
```
def Return_SARIMA_Model(y, p, d, q ,P, D, Q, S):
    order = [p,d,q]
    s_order = [P,D,Q,S]
    model=sm.tsa.statespace.SARIMAX(endog=y,order=order,seasonal_order=s_order)
    results=model.fit()
    return results
```

-> Add dummy columns

In [3]:
```
def Add_Month_Datecolumn(df, No):
    month_NO = []
    for i in range(1, len(df)+1):
        if  (i - No)%12 == 0:
            month_NO.append(1)
        else:
            month_NO.append(0)
    df.loc[:, str(No)] = month_NO
```

Prepare dataset (TSA HW07.co2.csv)

In [4]:
```python
# read dataset
df = pd.read_csv(r"C:\Users\TerryYang\Desktop\Github\2021-TSA-Assignment-NTUIIE\H
df
```

Out[4]:

	time_trend	month	co2_level
0	1994.000000	Jan	363.05
1	1994.083333	Feb	364.18
2	1994.166667	Mar	364.87
3	1994.250000	Apr	364.47
4	1994.333333	May	364.32
...	...	...	...
127	2004.583333	Aug	368.69
128	2004.666667	Sep	368.55
129	2004.750000	Oct	373.39
130	2004.833333	Nov	378.49
131	2004.916667	Dec	381.62

132 rows × 3 columns

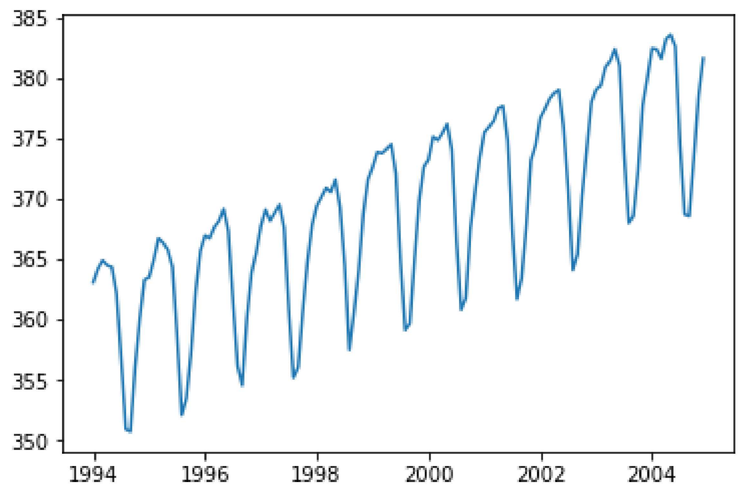
In [5]:
```python
# adding dummy columns.
for i in range(1,12):
    Add_Month_Datecolumn(df, i)
df
```

Out[5]:

	time_trend	month	co2_level	1	2	3	4	5	6	7	8	9	10	11
0	1994.000000	Jan	363.05	1	0	0	0	0	0	0	0	0	0	0
1	1994.083333	Feb	364.18	0	1	0	0	0	0	0	0	0	0	0
2	1994.166667	Mar	364.87	0	0	1	0	0	0	0	0	0	0	0
3	1994.250000	Apr	364.47	0	0	0	1	0	0	0	0	0	0	0
4	1994.333333	May	364.32	0	0	0	0	1	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
127	2004.583333	Aug	368.69	0	0	0	0	0	0	0	1	0	0	0
128	2004.666667	Sep	368.55	0	0	0	0	0	0	0	0	1	0	0
129	2004.750000	Oct	373.39	0	0	0	0	0	0	0	0	0	1	0
130	2004.833333	Nov	378.49	0	0	0	0	0	0	0	0	0	0	1
131	2004.916667	Dec	381.62	0	0	0	0	0	0	0	0	0	0	0

132 rows × 14 columns

In [6]:
```
# plot the series
plt.plot(df["time_trend"].tolist(), df["co2_level"].tolist())
plt.show()
```



Q5.

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.)

==> By the OLS regression summary no cofficients are significant. The adjusted R-squared is 0.989

In [7]:
```python
# split X and y
X = sm.add_constant(df.drop(['month', "co2_level"], axis=1))
y = df["co2_level"]

# OLS regression
reg = sm.OLS(y, X)
result = reg.fit()

# print result
print("parameters: \n", result.params)
print("\n", result.summary())
```

```
parameters:
 const        -3291.877932
time_trend       1.832098
1                1.336697
2                2.004932
3                2.300437
4                2.567763
5                2.864180
6                0.660595
7               -5.948443
8              -12.104754
9              -11.483794
10              -6.923741
11              -2.590960
dtype: float64

                            OLS Regression Results
==============================================================================
Dep. Variable:              co2_level   R-squared:                       0.990
Model:                            OLS   Adj. R-squared:                  0.989
Method:                 Least Squares   F-statistic:                     997.7
Date:                Sun, 19 Dec 2021   Prob (F-statistic):          2.93e-113
Time:                        13:11:14   Log-Likelihood:                -151.49
No. Observations:                 132   AIC:                             329.0
Df Residuals:                     119   BIC:                             366.4
Df Model:                          12
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -3291.8779     44.199    -74.478      0.000   -3379.397   -3204.359
time_trend     1.8321      0.022     82.899      0.000       1.788       1.876
1              1.3367      0.343      3.897      0.000       0.658       2.016
2              2.0049      0.343      5.847      0.000       1.326       2.684
3              2.3004      0.343      6.711      0.000       1.622       2.979
4              2.5678      0.343      7.493      0.000       1.889       3.246
5              2.8642      0.343      8.360      0.000       2.186       3.543
6              0.6606      0.343      1.928      0.056      -0.018       1.339
7             -5.9484      0.342    -17.368      0.000      -6.627      -5.270
8            -12.1048      0.342    -35.347      0.000     -12.783     -11.427
9            -11.4838      0.342    -33.537      0.000     -12.162     -10.806
10            -6.9237      0.342    -20.221      0.000      -7.602      -6.246
11            -2.5910      0.342     -7.567      0.000      -3.269      -1.913
==============================================================================
```

```
Omnibus:                        3.248   Durbin-Watson:                   0.983
Prob(Omnibus):                  0.197   Jarque-Bera (JB):                3.189
Skew:                           0.376   Prob(JB):                        0.203
Kurtosis:                       2.883   Cond. No.                     1.26e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
[2] The condition number is large, 1.26e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

(b). Identify, estimate the SARIMA model for the co2 level.

==> After testing multiple parameter settings, we found out that p, d, q = (0, 1, 1) X P, D, Q = (0, 1, 1)12 has minimum AIC

In [8]:
```
# finding AIC under different parameter settings
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
s_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]

aic_list = []
pdq_index = []
s_pdq_index = []
for i in range(0,len(pdq)):
    for j in range(0,len(s_pdq)):
        order = pdq[i]
        s_order = s_pdq[j]
        result = Return_SARIMA_Model(y, *order, *s_order)
        aic_list.append(result.aic)
        pdq_index.append(i)
        s_pdq_index.append(j)
```

In [9]:
```
# report AIC
orders = []
s_orders = []
for i in range(0, len(aic_list)):
    orders.append(pdq[pdq_index[i]])
    s_orders.append(s_pdq[s_pdq_index[i]])
data = {'p, d, q': orders,
        'P, D, Q, S': s_orders,
        'AIC': aic_list
        }

df_aic = pd.DataFrame(data)
df_aic
```

Out[9]:

	p, d, q	P, D, Q, S	AIC
0	(0, 0, 0)	(0, 0, 0, 12)	1937.143243
1	(0, 0, 0)	(0, 0, 1, 12)	1797.507621
2	(0, 0, 0)	(0, 1, 0, 12)	528.074894
3	(0, 0, 0)	(0, 1, 1, 12)	504.978544
4	(0, 0, 0)	(1, 0, 0, 12)	717.257477
...	...	...	...
59	(1, 1, 1)	(0, 1, 1, 12)	287.056872
60	(1, 1, 1)	(1, 0, 0, 12)	405.983285
61	(1, 1, 1)	(1, 0, 1, 12)	359.570864
62	(1, 1, 1)	(1, 1, 0, 12)	305.871366
63	(1, 1, 1)	(1, 1, 1, 12)	289.032710

64 rows × 3 columns

In [10]:
```
# finding minimum AIC setting
min_aic = min(aic_list)
min_index = aic_list.index(min_aic)
op_order = pdq[pdq_index[min_index]]
op_s_order = s_pdq[s_pdq_index[min_index]]

# print
print("Optimal (p, d, q) : ", op_order)
print("Optimal (P, D, Q, S) : ", op_s_order)
```

```
Optimal (p, d, q) :  (0, 1, 1)
Optimal (P, D, Q, S) :  (0, 1, 1, 12)
```

In [11]:

```
# show optimal SARIMA result
result = Return_SARIMA_Model(y, *op_order, *op_s_order)
print("least AIC: \n", result.aic)
result.summary()
```

```
least AIC:
 285.0949783730558
```

Out[11]:

SARIMAX Results

Dep. Variable:	co2_level	**No. Observations:**	132
Model:	SARIMAX(0, 1, 1)x(0, 1, 1, 12)	**Log Likelihood**	-139.547
Date:	Sun, 19 Dec 2021	**AIC**	285.095
Time:	13:12:08	**BIC**	293.432
Sample:	0	**HQIC**	288.481
	- 132		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.5791	0.093	-6.254	0.000	-0.761	-0.398
ma.S.L12	-0.8205	0.117	-7.017	0.000	-1.050	-0.591
sigma2	0.5448	0.073	7.484	0.000	0.402	0.687

Ljung-Box (L1) (Q):	0.01	**Jarque-Bera (JB):**	2.13
Prob(Q):	0.94	**Prob(JB):**	0.34
Heteroskedasticity (H):	1.04	**Skew:**	-0.15
Prob(H) (two-sided):	0.90	**Kurtosis:**	3.58

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

(c) Compare the two models above, what do you observe?

==> we conclude S.ARIMA fits this time series better, having smaller AIC than OLS regression does (285 < 329)