

Auto regression example

In [1]:

```
# Auto Regression example - suitable for data without trend and seasonal component
from statsmodels.tsa.ar_model import AutoReg

# create a linear data
data = [10,12,14,16,18,20,20,22] #linear data
```

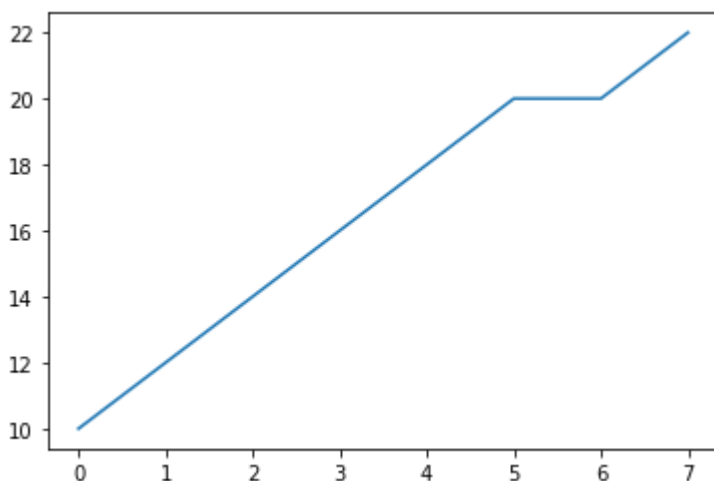
In [2]:

```
import matplotlib.pyplot as plt
plt.plot(data)
```

Bad key "text.kerning_factor" on line 4 in
 C:\Users\91920\anaconda3\lib\site-packages\matplotlib\mpl-data\stylelib_classic_test_patch.mplstyle.
 You probably need to get an updated matplotlibrc file from
<https://github.com/matplotlib/matplotlib/blob/v3.1.3/matplotlibrc.template>
 (https://github.com/matplotlib/matplotlib/blob/v3.1.3/matplotlibrc.template)
 or from the matplotlib source distribution

Out[2]:

[<matplotlib.lines.Line2D at 0x207a4765ac8>]



In [3]:

```
ARmodel = AutoReg(data, lags=1) #calling Auto regression model
ARmodel_fit = ARmodel.fit() #Fitting model
```

In [4]:

```
# making predictions
ypredicted = ARmodel_fit.predict(8,10)
print(ypredicted)
```

[23.125 24.14453125 25.06848145]

In [5]:

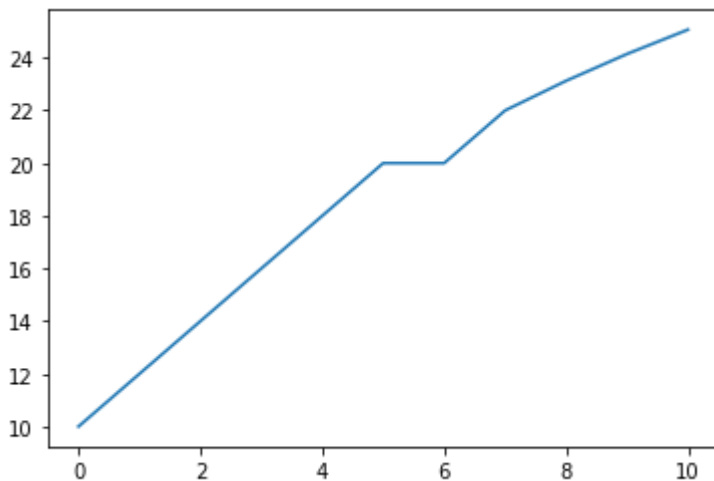
```
pred_list = ypredicted.tolist()
combinedlist = data+pred_list
```

In [6]:

```
plt.plot(combinedlist)
```

Out[6]:

[<matplotlib.lines.Line2D at 0x207a483da48>]



Moving average example

In [7]:

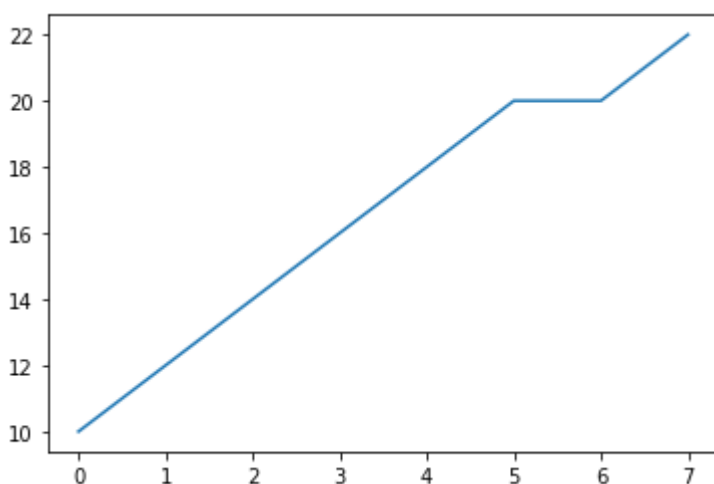
```
from statsmodels.tsa.arima_model import ARMA #suitable for data without trend and seasonal
```

In [8]:

```
import matplotlib.pyplot as plt
plt.plot(data)
```

Out[8]:

[<matplotlib.lines.Line2D at 0x207a4cde888>]



In [9]:

```
# fit model
MAmodel = ARMA(data, order=(0, 1)) #model with AR=0 and MA=1
MAmodel_fit = MAmodel.fit(dispatch=False)
```

In [10]:

```
# make prediction
ypredicted = MAmodel_fit.predict(8,10)
print(ypredicted)
```

```
[20.51275988 16.49884854 16.49884854]
```

In [11]:

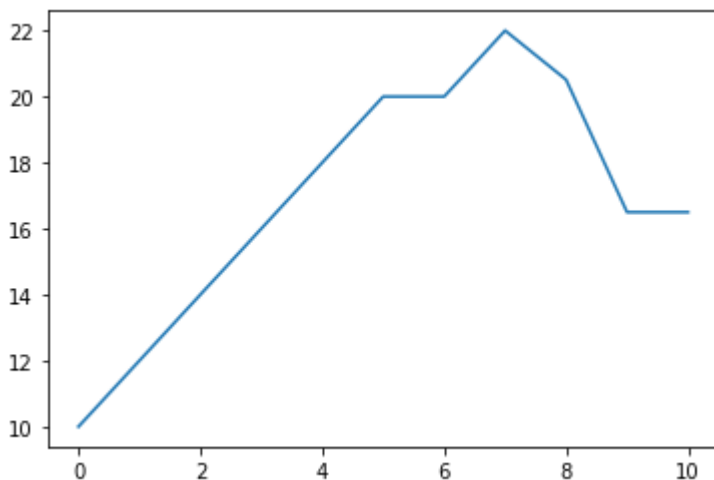
```
pred_list = ypredicted.tolist()
combinedlist = data+pred_list
```

In [12]:

```
plt.plot(combinedlist)
```

Out[12]:

```
[<matplotlib.lines.Line2D at 0x207a5d28cc8>]
```



Autoregressive Moving Average (ARMA)

In [13]:

```
# Import Data - Australian Drug Sales
import pandas as pd
DrugSalesData = pd.read_csv('C:/Users/91920/Downloads/TimeSeries.csv', parse_dates=['Date'])
```

In [14]:

DrugSalesData

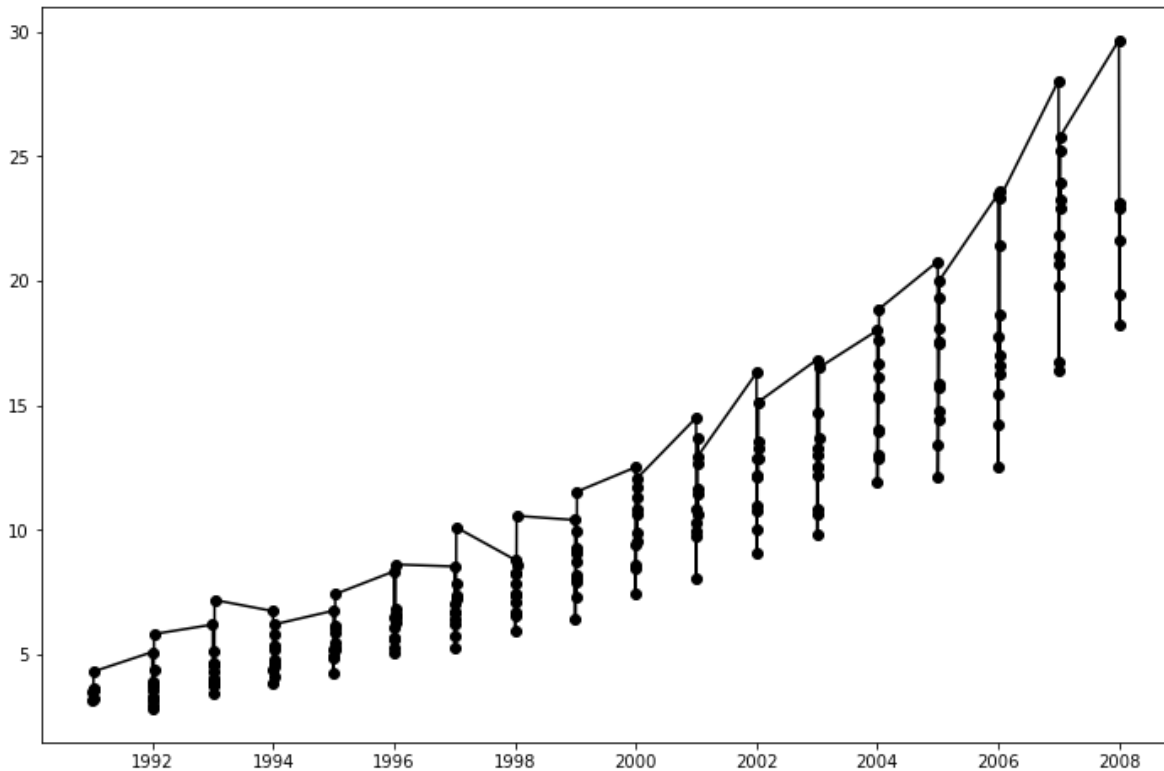
Out[14]:

	Value
Date	
1991-01-07	3.526591
1991-01-08	3.180891
1991-01-09	3.252221
1991-01-10	3.611003
1991-01-11	3.565869
...	...
2008-01-02	21.654285
2008-01-03	18.264945
2008-01-04	23.107677
2008-01-05	22.912510
2008-01-06	19.431740

204 rows × 1 columns

In [15]:

```
plt.figure(figsize=(12, 8))  
plt.plot(DrugSalesData, marker="o", color="black",label='Original data')  
plt.show()
```



In [16]:

```
from statsmodels.tsa.stattools import adfuller  
test_result=adfuller(DrugSalesData['Value'])  
test_result
```

Out[16]:

```
(3.14518568930673,  
 1.0,  
 15,  
 188,  
 {'1%': -3.465620397124192,  
  '5%': -2.8770397560752436,  
  '10%': -2.5750324547306476},  
 549.6705685364174)
```

In [17]:

```
DrugSalesData['Seasonal_Difference']=DrugSalesData['Value']-DrugSalesData['Value'].shift(1)
## Again test dickey fuller test
test_result=adfuller(DrugSalesData['Seasonal_Difference'].dropna())
test_result
```

Out[17]:

```
(-2.495172147449673,
 0.11665341686470398,
 15,
 187,
 {'1%': -3.465811691080702,
  '5%': -2.877123351472649,
  '10%': -2.5750770662586864},
 547.1880809726357)
```

In [18]:

```
DrugSalesData['Seasonal_Difference']=DrugSalesData['Value']-DrugSalesData['Value'].shift(8)
## Again test dickey fuller test
test_result=adfuller(DrugSalesData['Seasonal_Difference'].dropna())
test_result
```

Out[18]:

```
(-4.495051811502679,
 0.00020072804791958012,
 15,
 180,
 {'1%': -3.4672111510631, '5%': -2.877734766803841, '10%': -2.57540336419753
 1},
 518.1519503724882)
```

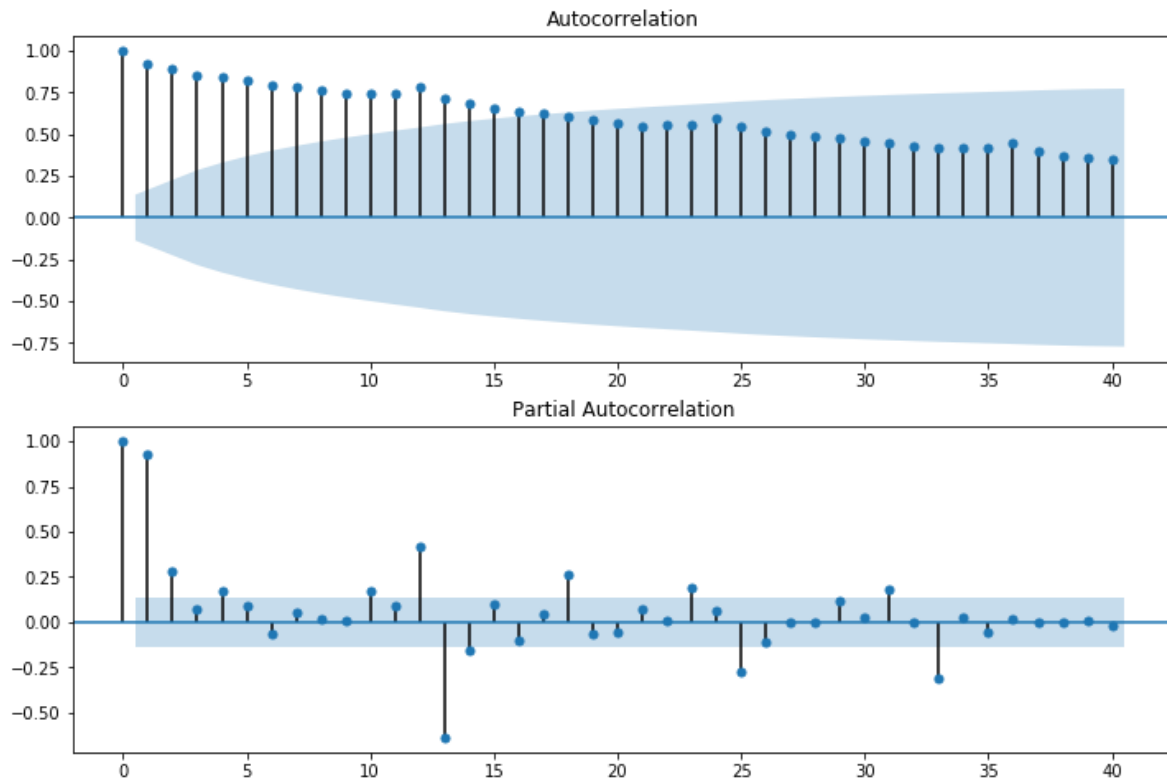
In [19]:

```
import statsmodels.api as sm
print(sm.__version__)
#from statsmodels.graphics.api import qqplot
```

0.11.0

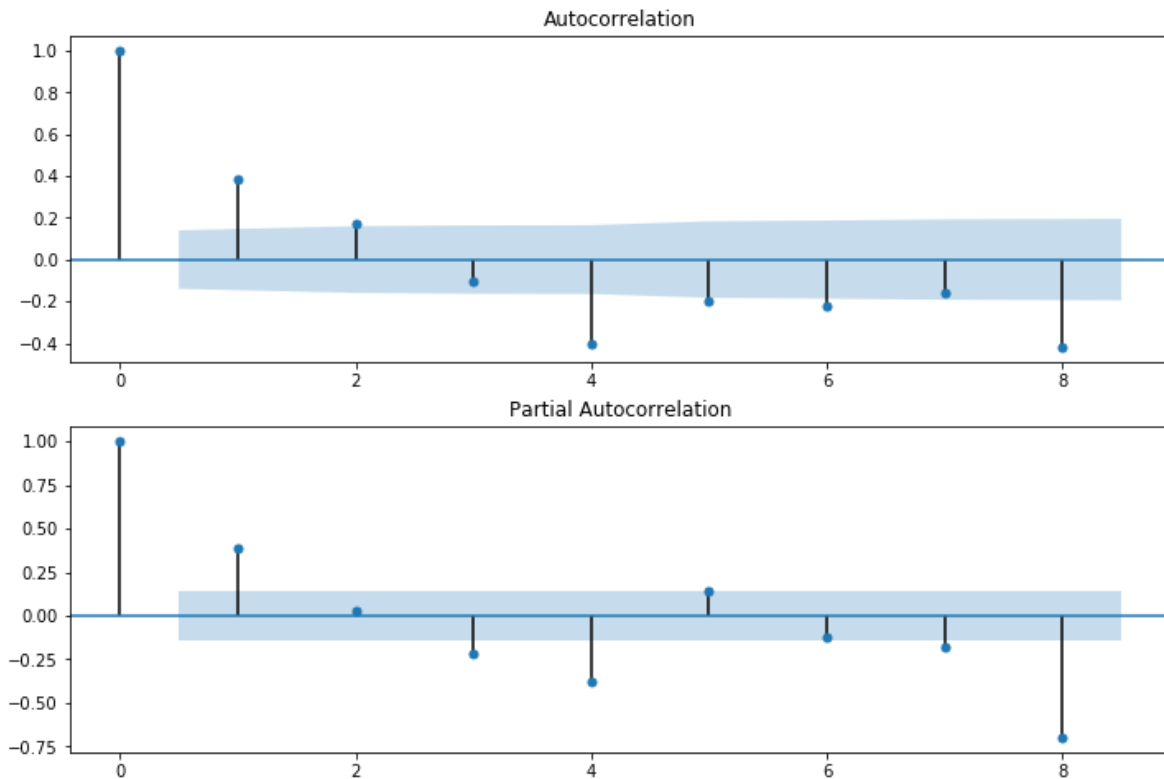
In [20]:

```
%matplotlib inline
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(DrugSalesData['Value'], lags=40, ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(DrugSalesData['Value'], lags=40, ax=ax2)
```



In [21]:

```
%matplotlib inline
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(DrugSalesData['Seasonal_Difference'].dropna(), lags=8, ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(DrugSalesData['Seasonal_Difference'].dropna(), lags=8, ax=ax2)
```



In [22]:

```
import statsmodels.api as sm
from statsmodels.tsa.arima_model import ARMA
# fit model
ARMAmodel = ARMA(DrugSalesData['Value'], order=(1, 1))
ARmodel_fit = ARMAmodel.fit(dispatch=False)
```

C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:218: ValueWarning: A date index has been provided, but it has no associated frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)

In [23]:

```
actuals = DrugSalesData['Value'][200:204]
actuals
```

Out[23]:

```
Date
2008-01-03    18.264945
2008-01-04    23.107677
2008-01-05    22.912510
2008-01-06    19.431740
Name: Value, dtype: float64
```

In [24]:

```
ypredicted = ARmodel_fit.predict(200,203) # end point included
print(ypredicted)
```

```
Date
2008-01-03    24.439976
2008-01-04    21.945462
2008-01-05    22.341809
2008-01-06    22.503963
dtype: float64
```

In [25]:

```
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(actuals, ypredicted)
print('MAE: %f' % mae)
#print(ARmodel_fit.aic)
```

MAE: 2.745043

In [26]:

```

import itertools
i = j = range(0, 4)
ij = itertools.product(i,j)
for parameters in ij:
    try:
        mod = ARMA(DrugSalesData['Value'],order=parameters)
        results = mod.fit()
        ypredicted = results.predict(200,203) # end point included
        mae = mean_absolute_error(actuals, ypredicted)
        print('ARMA{} - MAE:{}'.format(parameters, mae))
        #print('ARMA{} - AIC:{}'.format(parameters, results.aic))
    except:
        continue

```

```

ARMA(0, 0) - MAE:10.234788417843136
ARMA(0, 1) - MAE:6.273327630994247
ARMA(0, 2) - MAE:5.2735154307356025

```

```

C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)

```

```

ARMA(0, 3) - MAE:4.777577781760413
ARMA(1, 0) - MAE:2.8253588034976946

```

```

C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)

```

```

ARMA(1, 1) - MAE:2.745042526407625

```

```

C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)

```

```

ARMA(1, 2) - MAE:2.4082091629451243

```

```

C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.
py:218: ValueWarning: A date index has been provided, but it has no associ
ated frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)

```

```
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\kalmanf\kalmanf
ilter.py:220: RuntimeWarning: divide by zero encountered in true_divide
  Z_mat, R_mat, T_mat)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:68
9: RuntimeWarning: overflow encountered in exp
  newparams = ((1-np.exp(-params))/(1+np.exp(-params))).copy()
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:68
9: RuntimeWarning: invalid value encountered in true_divide
  newparams = ((1-np.exp(-params))/(1+np.exp(-params))).copy()
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:69
0: RuntimeWarning: overflow encountered in exp
  tmp = ((1-np.exp(-params))/(1+np.exp(-params))).copy()
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:69
0: RuntimeWarning: invalid value encountered in true_divide
  tmp = ((1-np.exp(-params))/(1+np.exp(-params))).copy()
```

ARMA(1, 3) - MAE:2.9438936458322402

ARMA(2, 0) - MAE:3.675098904332777

```
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\base\model.py:548: He
ssianInversionWarning: Inverting hessian failed, no bse or cov_params availa
ble
  'available', HessianInversionWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)
```

ARMA(2, 1) - MAE:2.5187571883196256

```
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\base\model.py:548: He
ssianInversionWarning: Inverting hessian failed, no bse or cov_params availa
ble
  'available', HessianInversionWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\base\model.py:568: Co
nvergenceWarning: Maximum Likelihood optimization failed to converge. Check
mle_retvals
  "Check mle_retvals", ConvergenceWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)
```

ARMA(2, 2) - MAE:2.6313071869824967

ARMA(2, 3) - MAE:2.9444555225887683

ARMA(3, 0) - MAE:3.1497453501775006

```
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.
py:218: ValueWarning: A date index has been provided, but it has no associ
ated frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.
py:218: ValueWarning: A date index has been provided, but it has no associ
ated frequency information and so will be ignored when e.g. forecasting.
```

```
' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\base\model.py:548:
HessianInversionWarning: Inverting hessian failed, no bse or cov_params av
ailable
'available', HessianInversionWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.
py:218: ValueWarning: A date index has been provided, but it has no associ
ated frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.
py:218: ValueWarning: A date index has been provided, but it has no associ
ated frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)

ARMA(3, 1) - MAE:2.727552051990841
ARMA(3, 3) - MAE:2.7942654979093646
```

```
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\base\model.py:548: He
ssianInversionWarning: Inverting hessian failed, no bse or cov_params availa
ble
'available', HessianInversionWarning)
```

In [27]:

```
ARMAmodel = ARMA(DrugSalesData['Value'], order=(1, 2))
ARmodel_fit = ARMAmodel.fit()
ypredicted = ARmodel_fit.predict(200,203) # end point included
print(ypredicted)
mae = mean_absolute_error(actuals, ypredicted)
print('MAE: %f' % mae)
print(ARmodel_fit.aic)
```

```
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)
```

```
Date
2008-01-03    22.957271
2008-01-04    21.658077
2008-01-05    23.093977
2008-01-06    22.741184
dtype: float64
MAE: 2.408209
869.2386494699966
```

We now calculate the Akaike Information Criterion (AIC), Schwarz Bayesian Information Criterion (BIC), and Hannan-Quinn Information Criterion (HQIC). Our goal is to choose a model that minimizes (AIC, BIC, HQIC).

In [28]:

```
# make prediction
ypredicted = ARmodel_fit.predict(len(DrugSalesData), len(DrugSalesData)+2)
print(ypredicted)
```

```
204    21.333698
205    21.903082
206    21.886535
dtype: float64
```

```
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:583: ValueWarning: No supported index is available. Prediction results wil
l be given with an integer index beginning at `start`.
ValueWarning)
```

In []:

In []:

Autoregressive Integrated Moving Average (ARIMA)

In [29]:

```
from statsmodels.tsa.arma_model import ARIMA
```

In [30]:

```
# fit model
ARIMAmoel = ARIMA(DrugSalesData['Value'], order=(1, 1, 1)) #notice p,d and q value here
ARIMA_model_fit = ARIMAmoel.fit(dis=False)

# make prediction
ypredicted = ARIMA_model_fit.predict(len(DrugSalesData), len(DrugSalesData)+3, typ='levels')
print(ypredicted)
```

```
203    21.584632
204    22.326892
205    22.626181
206    22.786365
dtype: float64
```

```
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:583: ValueWarning: No supported index is available. Prediction results wil
l be given with an integer index beginning at `start`.
ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:583: ValueWarning: No supported index is available. Prediction results wil
l be given with an integer index beginning at `start`.
ValueWarning)
```

In [31]:

```
mae = mean_absolute_error(actuals, ypredicted)
print('MAE: %f' % mae)
print(ARIMA_model_fit.aic)
```

```
MAE: 1.935356
843.8379425029489
```

In [32]:

```

import itertools
p= d = q = range(0, 4)
pdq = itertools.product(p,d,q)
for parameters in pdq:
    try:
        ARIMAmodel = ARIMA(DrugSalesData['Value'], order=parameters)
        results = ARIMAmodel.fit()
        ypredicted = results.predict(200,203) # end point included
        mae = mean_absolute_error(actuals, ypredicted)
        print('ARIMA{} - MAE:{}'.format(parameters, mae))
        #print('ARMA{} - AIC:{}'.format(parameters, results.aic))
    except:
        continue

```

```

C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.
py:218: ValueWarning: A date index has been provided, but it has no associ
ated frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.
py:218: ValueWarning: A date index has been provided, but it has no associ
ated frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.
py:218: ValueWarning: A date index has been provided, but it has no associ
ated frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.
py:218: ValueWarning: A date index has been provided, but it has no associ
ated frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.
py:218: ValueWarning: A date index has been provided, but it has no associ
ated frequency information and so will be ignored when e.g. forecasting.
  ' ignored when e.g. forecasting.', ValueWarning)

```

In [33]:

```
# fit model
ARIMAmoel = ARIMA(DrugSalesData['Value'], order=(1, 0, 2)) #notice p,d and q value here
ARIMA_model_fit = ARIMAmoel.fit()

# make prediction
ypredicted = ARIMA_model_fit.predict(len(DrugSalesData), len(DrugSalesData)+3, typ='levels')
print(ypredicted)
```

```
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
      ' ignored when e.g. forecasting.', ValueWarning)
```

```
204      21.333698
205      21.903082
206      21.886535
207      21.870017
dtype: float64
```

```
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:583: ValueWarning: No supported index is available. Prediction results wil
l be given with an integer index beginning at `start`.
      ValueWarning)
```

In [34]:

```
mae = mean_absolute_error(actuals, ypredicted)
print('MAE: %f' % mae)
print(ARIMA_model_fit.aic)
```

```
MAE: 1.934400
869.2386494699966
```

In []:

In [35]:

```

Sarima=sm.tsa.statespace.SARIMAX(DrugSalesData['Value'],order=(1, 0, 2),seasonal_order=(1,0
Sarima_fit = Sarima.fit()
ypredicted = Sarima_fit.predict(len(DrugSalesData), len(DrugSalesData)+3) # end point incl
mae = mean_absolute_error(actuals, ypredicted)
print('MAE: %f' % mae)

```

```

C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:218: ValueWarning: A date index has been provided, but it has no associate
d frequency information and so will be ignored when e.g. forecasting.
' ignored when e.g. forecasting.', ValueWarning)
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sarima
x.py:963: UserWarning: Non-stationary starting autoregressive parameters fou
nd. Using zeros as starting parameters.
warn('Non-stationary starting autoregressive parameters'
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sarima
x.py:994: UserWarning: Non-stationary starting seasonal autoregressive Using
zeros as starting parameters.
warn('Non-stationary starting seasonal autoregressive'
C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sarima
x.py:1006: UserWarning: Non-invertible starting seasonal moving average Usin
g zeros as starting parameters.
warn('Non-invertible starting seasonal moving average'

```

MAE: 1.719108

```

C:\Users\91920\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.p
y:583: ValueWarning: No supported index is available. Prediction results wil
l be given with an integer index beginning at `start`.
ValueWarning)

```

In [36]:

```

MAE: 2.962902 -4
MAE: 1.909766 -5
MAE: 1.719108 -8

```

File "<ipython-input-36-1ce317f3d213>", line 2

MAE: 1.909766 -5

^

IndentationError: unexpected indent

```

# Define the p, d and q parameters to take any value between 0 and 2
p = d = q = range(0, 2)

# Generate all different combinations of p, q and q triplets
pdq = list(itertools.product(p, d, q))

# Generate all different combinations of seasonal p, q and q triplets
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]

print('Examples of parameter combinations for Seasonal ARIMA...')
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]))
```

In []: