Team members:
Sirisha Lanka - PES1201700294
Malavikka R - PES1201700794
Kevin Arulraj - PES1201700659

Assignment 6

Problem Statement:
On a chosen time-series dataset, analyse the autoregressive moving average (ARMA) and from it, the autoregressive integrated moving average (ARIMA) and their respective Mean Squared Errors (MSE).

Dataset:
Sales of shampoo dataset

Code:
#1 ARMA

```
# ARMA example
from statsmodels.tsa.arima_model import ARMA
from matplotlib import pyplot
from sklearn.metrics import mean_squared_error
from random import random
# contrived dataset
data = [random() for x in range(1, 100)]
# fit model
# make prediction
size = int(len(data) * 0.66)
train, test = data[0:size], data[size:len(data)]
history = [x for x in train]
predictions = list()
for t in range(len(test)):
    model = ARMA(h)
    output = model_fit.forecast()
    yhat = output[0]
    predictions.append(yhat)
    obs = test[t]
    history.append(obs)
    print('predicted=%f, expected=%f' % (yhat, obs))
error = mean_squared_error(test, predictions)
print('Test MSE: %.3f' % error)
# plot
pyplot.plot(test)
pyplot.plot(predictions, color='red')istory, order=(5,1,0))
    model_fit = model.fit(disp=0
pyplot.show()
```

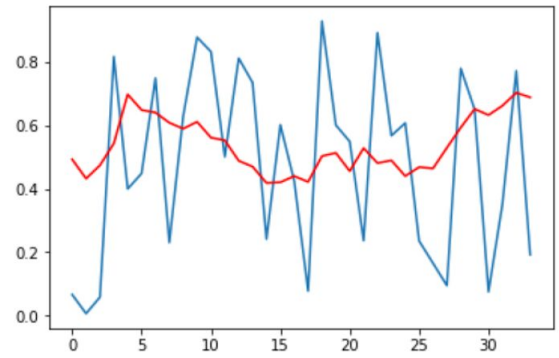
#2 ARIMA

```
from pandas import read_csv
from pandas import datetime
from pandas import DataFrame
from matplotlib import pyplot
from pandas.plotting import autocorrelation_plot
from statsmodels.tsa.arima_model import ARIMA
from sklearn.metrics import mean_squared_error
def parser(x):
    return datetime.strptime('190'+x, '%Y-%m')

series = read_csv('shampoo-sales.csv', header=0, parse_dates=[0], index_col=0, squeeze=True, date_parser=p
arser)
print(series.head())
series.plot()
pyplot.show()


autocorrelation_plot(series)
pyplot.show()


# fit model
model = ARIMA(series, order=(5,1,0))
model_fit = model.fit(disp=0)
print(model_fit.summary())
# plot residual errors
residuals = DataFrame(model_fit.resid)
residuals.plot()
pyplot.show()
residuals.plot(kind='kde')
pyplot.show()
print(residuals.describe())


X = series.values
size = int(len(X) * 0.66)
train, test = X[0:size], X[size:len(X)]
history = [x for x in train]
predictions = list()
for t in range(len(test)):
    model = ARIMA(history, order=(5,1,0))
    model_fit = model.fit(disp=0)
    output = model_fit.forecast()
    yhat = output[0]
    predictions.append(yhat)
    obs = test[t]
    history.append(obs)
    print('predicted=%f, expected=%f' % (yhat, obs))
error = mean_squared_error(test, predictions)
print('Test MSE: %.3f' % error)
# plot
pyplot.plot(test)
pyplot.plot(predictions, color='red')
pyplot.show()
```

Results:

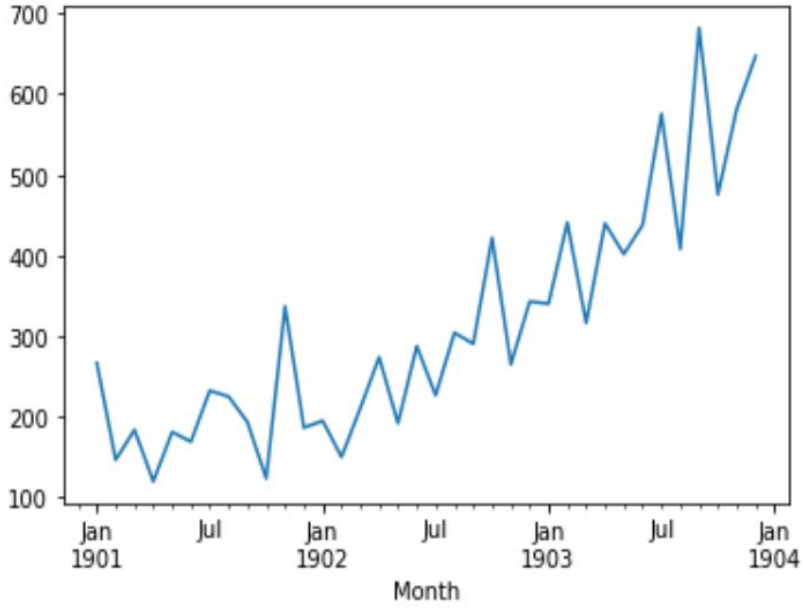
#1

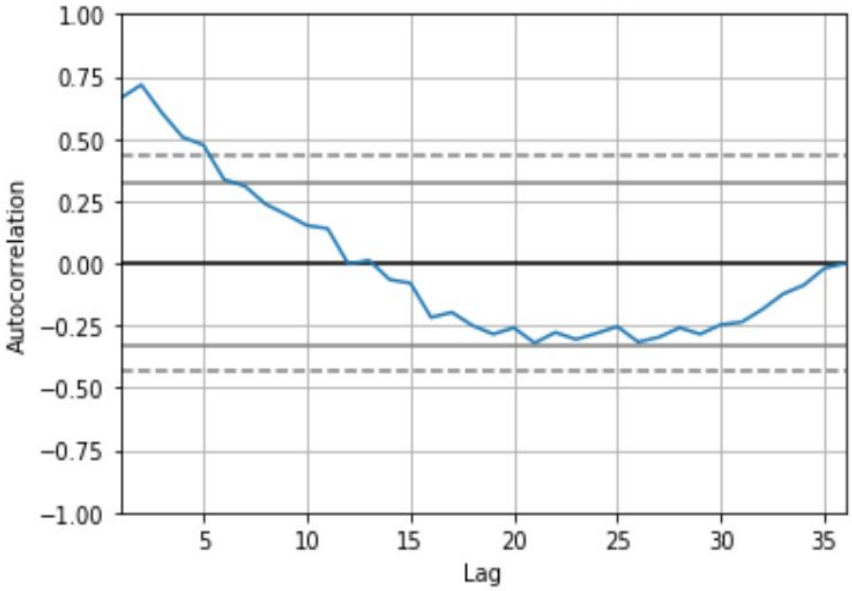
```
predicted=0.492630, expected=0.065770
predicted=0.431972, expected=0.005861
predicted=0.473877, expected=0.057566
predicted=0.542826, expected=0.817138
predicted=0.697795, expected=0.399477
predicted=0.648585, expected=0.448099
predicted=0.640436, expected=0.749134
predicted=0.608259, expected=0.229977
predicted=0.589433, expected=0.629092
predicted=0.611210, expected=0.878199
predicted=0.561222, expected=0.832480
predicted=0.552610, expected=0.500867
predicted=0.488176, expected=0.811939
predicted=0.469600, expected=0.734822
predicted=0.418156, expected=0.240812
predicted=0.420344, expected=0.601665
predicted=0.439844, expected=0.424466
predicted=0.421640, expected=0.076999
predicted=0.502956, expected=0.929210
predicted=0.513054, expected=0.600783
predicted=0.455556, expected=0.548858
predicted=0.528265, expected=0.235982
predicted=0.480825, expected=0.892300
predicted=0.489436, expected=0.567485
predicted=0.439918, expected=0.607303
predicted=0.468103, expected=0.235001
predicted=0.464081, expected=0.164903
predicted=0.529239, expected=0.094358
predicted=0.592988, expected=0.779699
predicted=0.651710, expected=0.650178
predicted=0.632460, expected=0.074733
predicted=0.661894, expected=0.350488
predicted=0.702876, expected=0.772050
predicted=0.688503, expected=0.191843
Test MSE: 0.086
```



#2

```
Month
1901-01-01    266.0
1901-02-01    145.9
1901-03-01    183.1
1901-04-01    119.3
1901-05-01    180.3
Name: Sales, dtype: float64
```





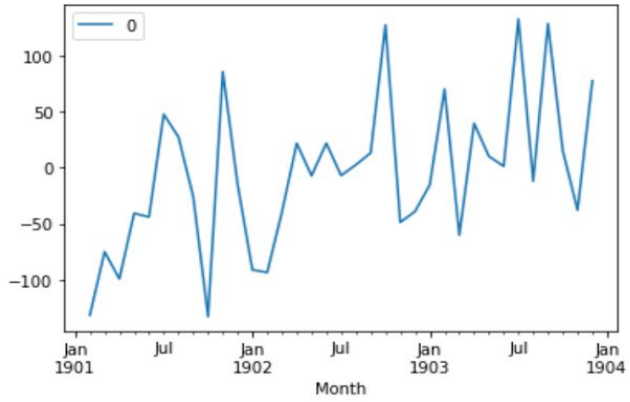
ARIMA Model Results

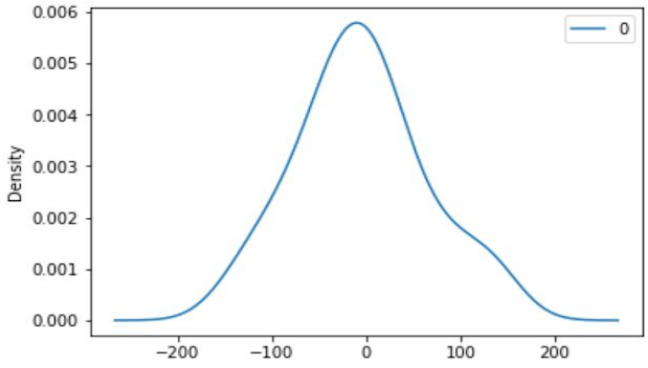
Dep. Variable:	D.Sales	No. Observations:	35
Model:	ARIMA(5, 1, 0)	Log Likelihood	-196.170
Method:	css-mle	S.D. of innovations	64.241
Date:	Fri, 18 Oct 2019	AIC	406.340
Time:	17:01:47	BIC	417.227
Sample:	02-01-1901 - 12-01-1903	HQIC	410.098

	coef	std err	z	P>|z|	[0.025	0.975]
const	12.0649	3.652	3.304	0.003	4.908	19.222
ar.L1.D.Sales	-1.1082	0.183	-6.063	0.000	-1.466	-0.750
ar.L2.D.Sales	-0.6203	0.282	-2.203	0.036	-1.172	-0.068
ar.L3.D.Sales	-0.3606	0.295	-1.222	0.231	-0.939	0.218
ar.L4.D.Sales	-0.1252	0.280	-0.447	0.658	-0.674	0.424
ar.L5.D.Sales	0.1289	0.191	0.673	0.506	-0.246	0.504

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	-1.0617	-0.5064j	1.1763	-0.4292
AR.2	-1.0617	+0.5064j	1.1763	0.4292
AR.3	0.0816	-1.3804j	1.3828	-0.2406
AR.4	0.0816	+1.3804j	1.3828	0.2406
AR.5	2.9315	-0.0000j	2.9315	-0.0000





```
count   35.000000
mean    -5.495218
std     68.132882
min   -133.296637
25%    -42.477890
50%     -7.186512
75%     24.748330
max    133.237936
predicted=349.117712, expected=342.300000
predicted=306.512952, expected=339.700000
predicted=387.376449, expected=440.400000
predicted=348.154255, expected=315.900000
predicted=386.308818, expected=439.300000
predicted=356.082087, expected=401.300000
predicted=446.379462, expected=437.400000
predicted=394.737224, expected=575.500000
predicted=434.915402, expected=407.600000
predicted=507.923547, expected=682.000000
predicted=435.482779, expected=475.300000
predicted=652.743826, expected=581.300000
predicted=546.343519, expected=646.900000
Test MSE: 6958.324
```