SAMPLE CASE - ARIMA MODELING

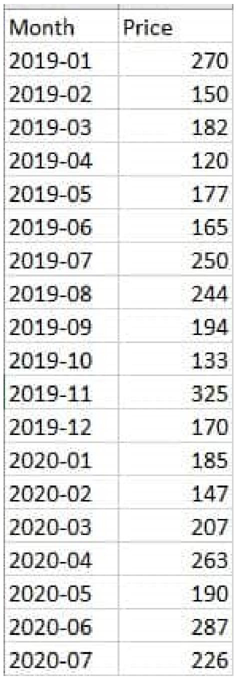
Contents

Price Forecasting - Autoregressive Integrated Moving Average Model (ARIMA)

Example shows a process of building an ARIMA model to predict the purchase price of concrete, a construction material. The model is based on historical data of price and analyzes the pattern of data. I applied ARIMA model in HIRC Inc. for material price forecasting.

- **Raw Data**

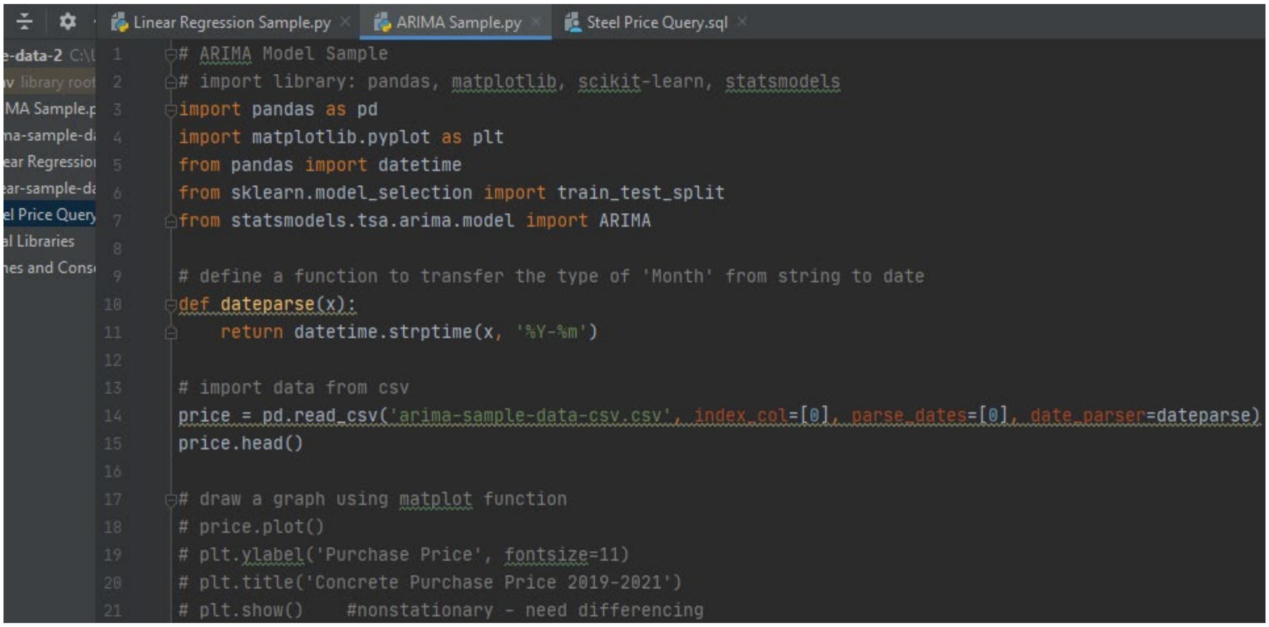
*raw data is sample data



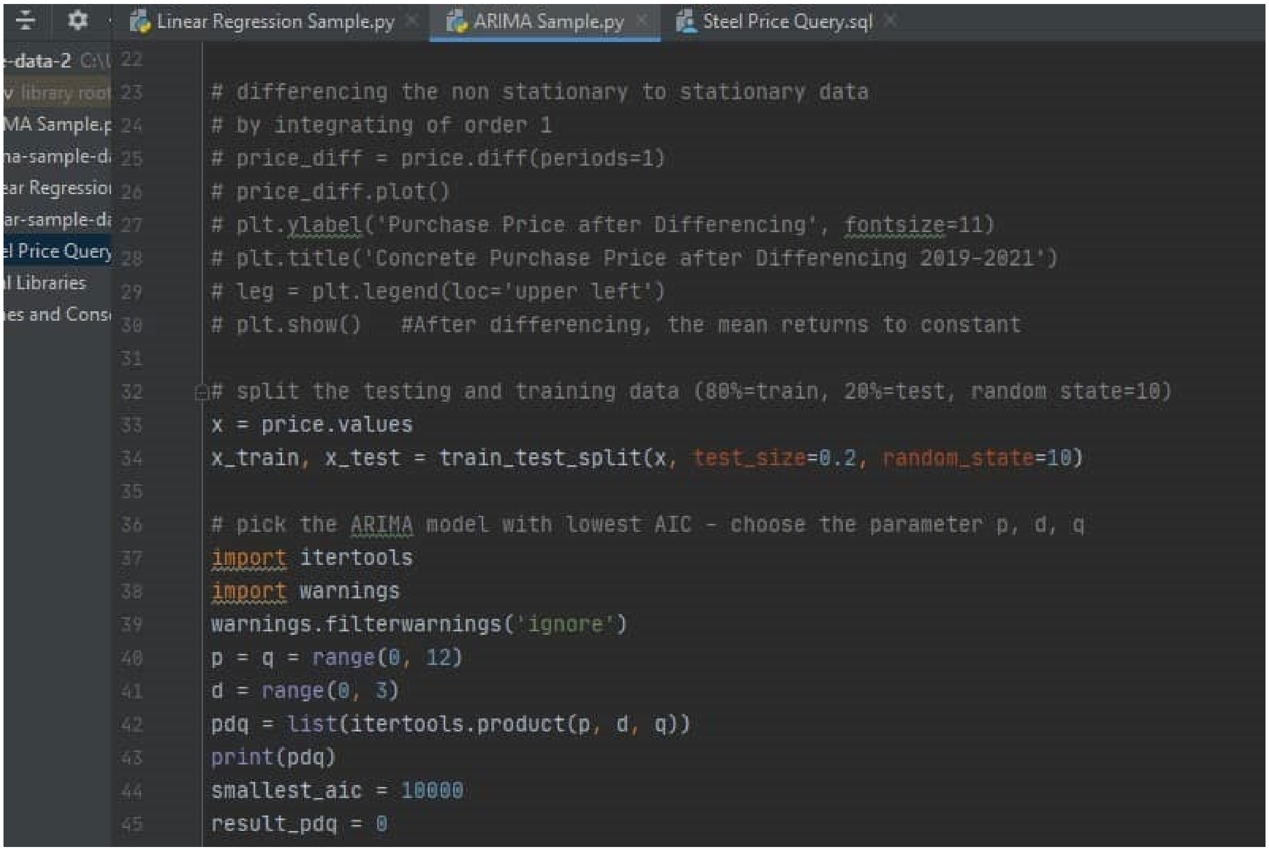
Month	Price
2019-01	270
2019-02	150
2019-03	182
2019-04	120
2019-05	177
2019-06	165
2019-07	250
2019-08	244
2019-09	194
2019-10	133
2019-11	325
2019-12	170
2020-01	185
2020-02	147
2020-03	207
2020-04	263
2020-05	190
2020-06	287
2020-07	226

- **ARIMA Modeling Code**

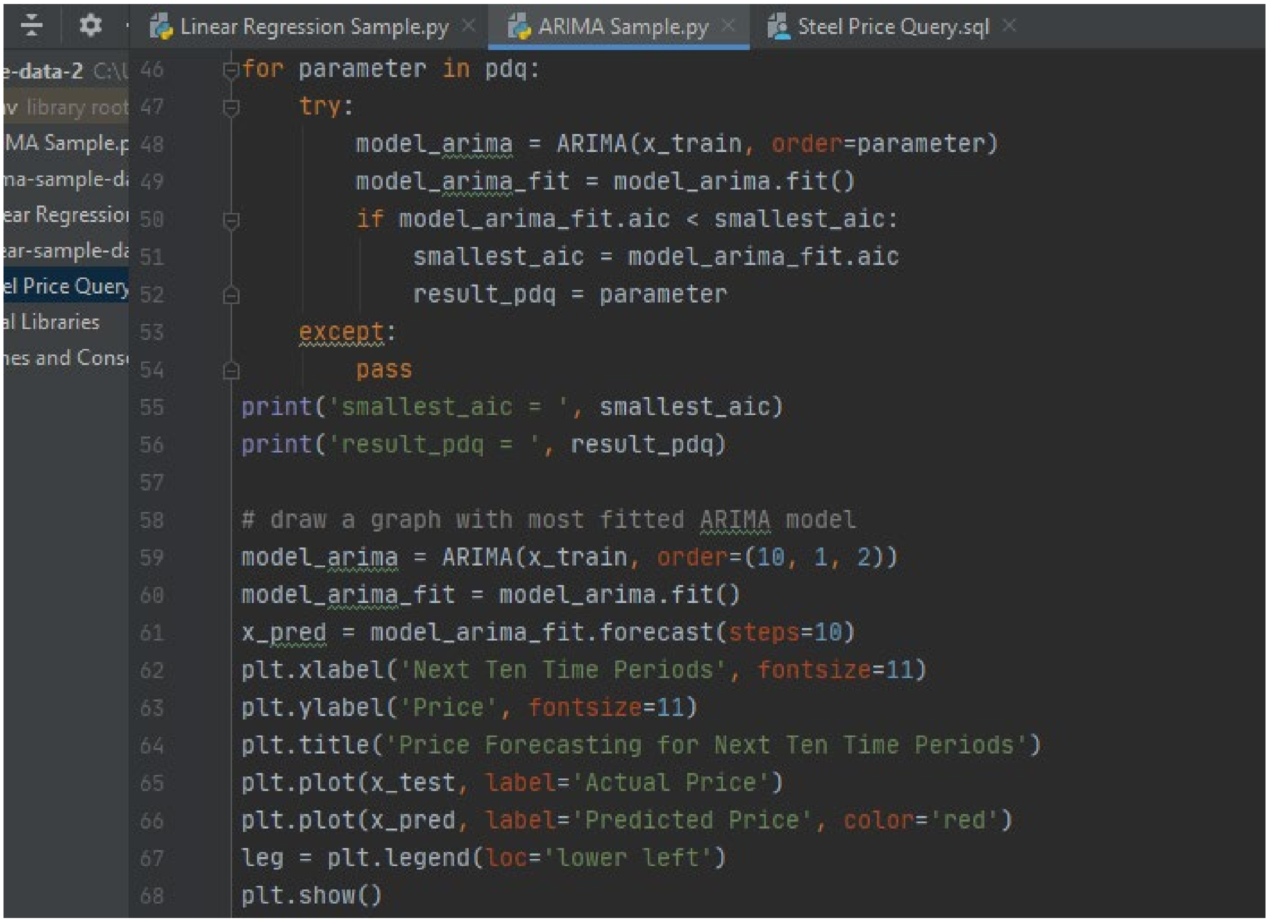
➢ Library Used: Pandas, Matplotlib, Scikit-Learn, Statsmodels



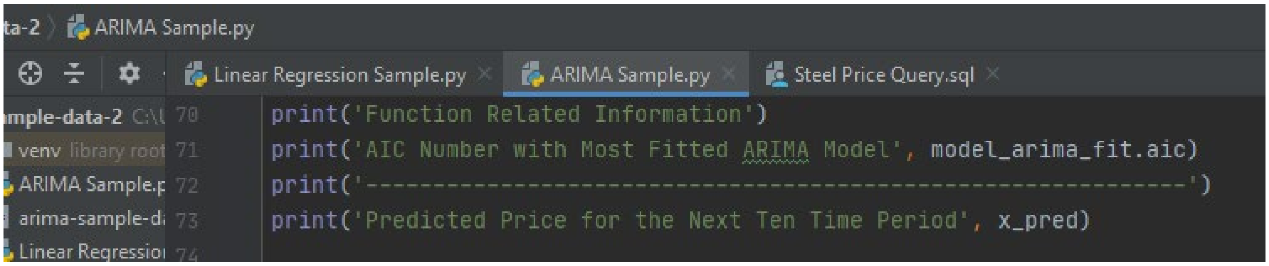
```python
# ARIMA Model Sample
# import library: pandas, matplotlib, scikit-learn, statsmodels
import pandas as pd
import matplotlib.pyplot as plt
from pandas import datetime
from sklearn.model_selection import train_test_split
from statsmodels.tsa.arima.model import ARIMA

# define a function to transfer the type of 'Month' from string to date
def dateparse(x):
    return datetime.strptime(x, '%Y-%m')

# import data from csv
price = pd.read_csv('arima-sample-data-csv.csv', index_col=[0], parse_dates=[0], date_parser=dateparse)
price.head()

# draw a graph using matplot function
# price.plot()
# plt.ylabel('Purchase Price', fontsize=11)
# plt.title('Concrete Purchase Price 2019-2021')
# plt.show()    #nonstationary - need differencing
```



```python

# differencing the non stationary to stationary data
# by integrating of order 1
# price_diff = price.diff(periods=1)
# price_diff.plot()
# plt.ylabel('Purchase Price after Differencing', fontsize=11)
# plt.title('Concrete Purchase Price after Differencing 2019-2021')
# leg = plt.legend(loc='upper left')
# plt.show()   #After differencing, the mean returns to constant

# split the testing and training data (80%=train, 20%=test, random state=10)
x = price.values
x_train, x_test = train_test_split(x, test_size=0.2, random_state=10)

# pick the ARIMA model with lowest AIC - choose the parameter p, d, q
import itertools
import warnings
warnings.filterwarnings('ignore')
p = q = range(0, 12)
d = range(0, 3)
pdq = list(itertools.product(p, d, q))
print(pdq)
smallest_aic = 10000
result_pdq = 0
```

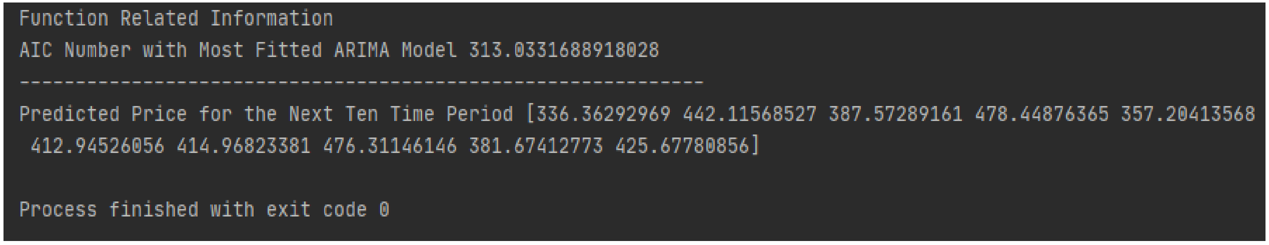


```python
for parameter in pdq:
    try:
        model_arima = ARIMA(x_train, order=parameter)
        model_arima_fit = model_arima.fit()
        if model_arima_fit.aic < smallest_aic:
            smallest_aic = model_arima_fit.aic
            result_pdq = parameter
    except:
        pass
print('smallest_aic = ', smallest_aic)
print('result_pdq = ', result_pdq)

# draw a graph with most fitted ARIMA model
model_arima = ARIMA(x_train, order=(10, 1, 2))
model_arima_fit = model_arima.fit()
x_pred = model_arima_fit.forecast(steps=10)
plt.xlabel('Next Ten Time Periods', fontsize=11)
plt.ylabel('Price', fontsize=11)
plt.title('Price Forecasting for Next Ten Time Periods')
plt.plot(x_test, label='Actual Price')
plt.plot(x_pred, label='Predicted Price', color='red')
leg = plt.legend(loc='lower left')
plt.show()
```



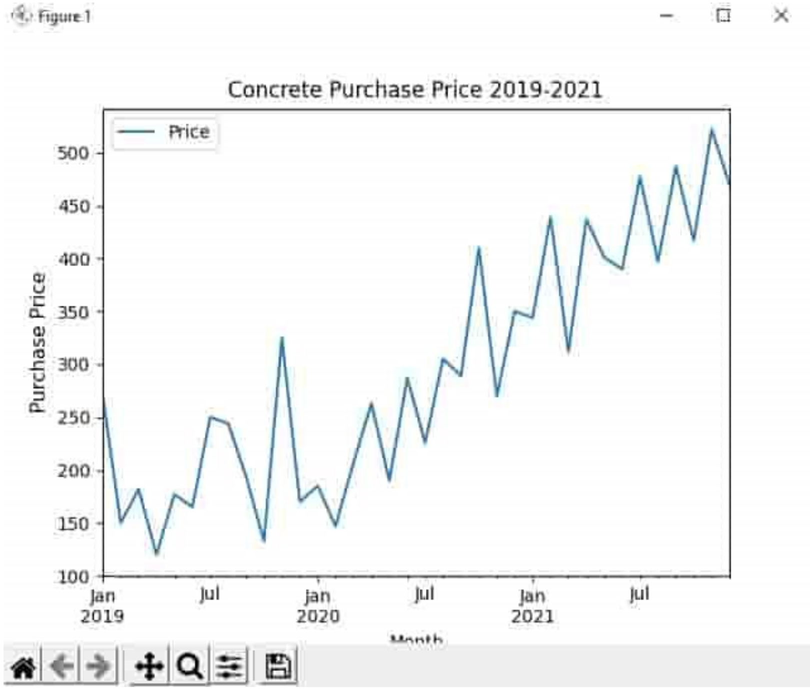
```python
print('Function Related Information')
print('AIC Number with Most Fitted ARIMA Model', model_arima_fit.aic)
print('-------------------------------------------------------------')
print('Predicted Price for the Next Ten Time Period', x_pred)
```

- **Output**

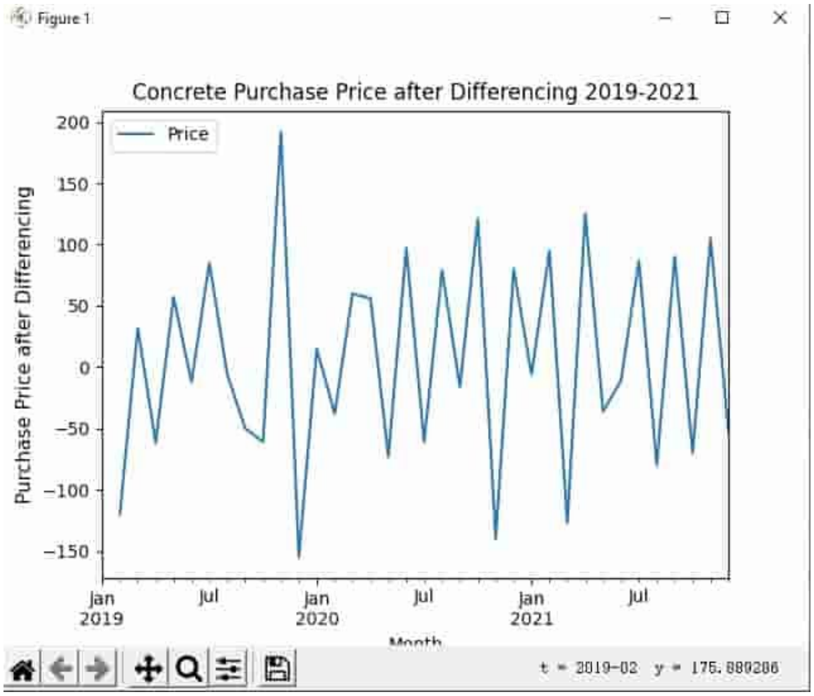


```
Function Related Information
AIC Number with Most Fitted ARIMA Model 313.0331688918028
-------------------------------------------------------------
Predicted Price for the Next Ten Time Period [336.36292969 442.11568527 387.57289161 478.44876365 357.20413568
 412.94526056 414.96823381 476.31146146 381.67412773 425.67780856]

Process finished with exit code 0
```

- Graph
  - For Original Data



  - For Data after Differencing (make data stationary)



- For Predicted Price

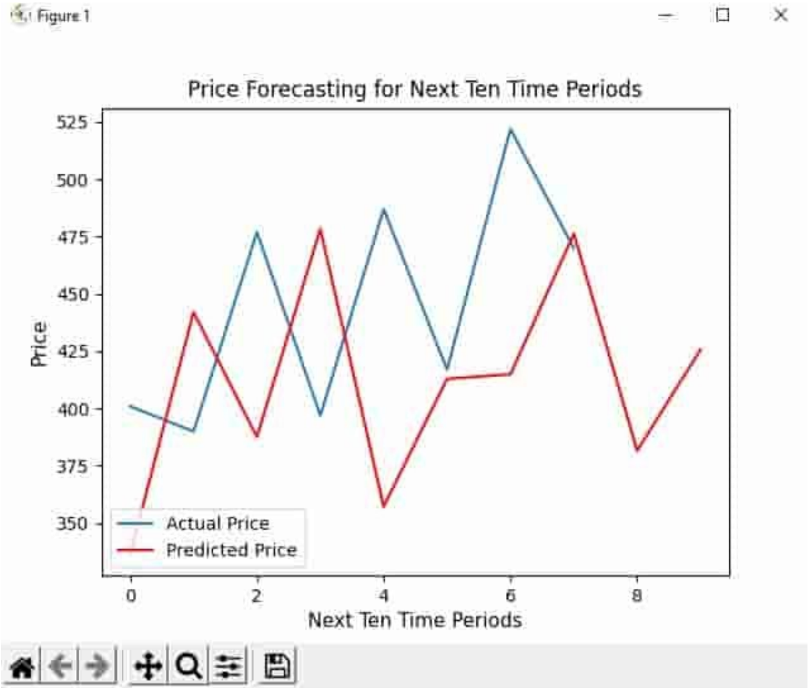


Figure 1
Price Forecasting for Next Ten Time Periods
Price
Next Ten Time Periods
Actual Price
Predicted Price