Cognixia

A Collabera LEARNING SOLUTIONS COMPANY

Time Series Analysis

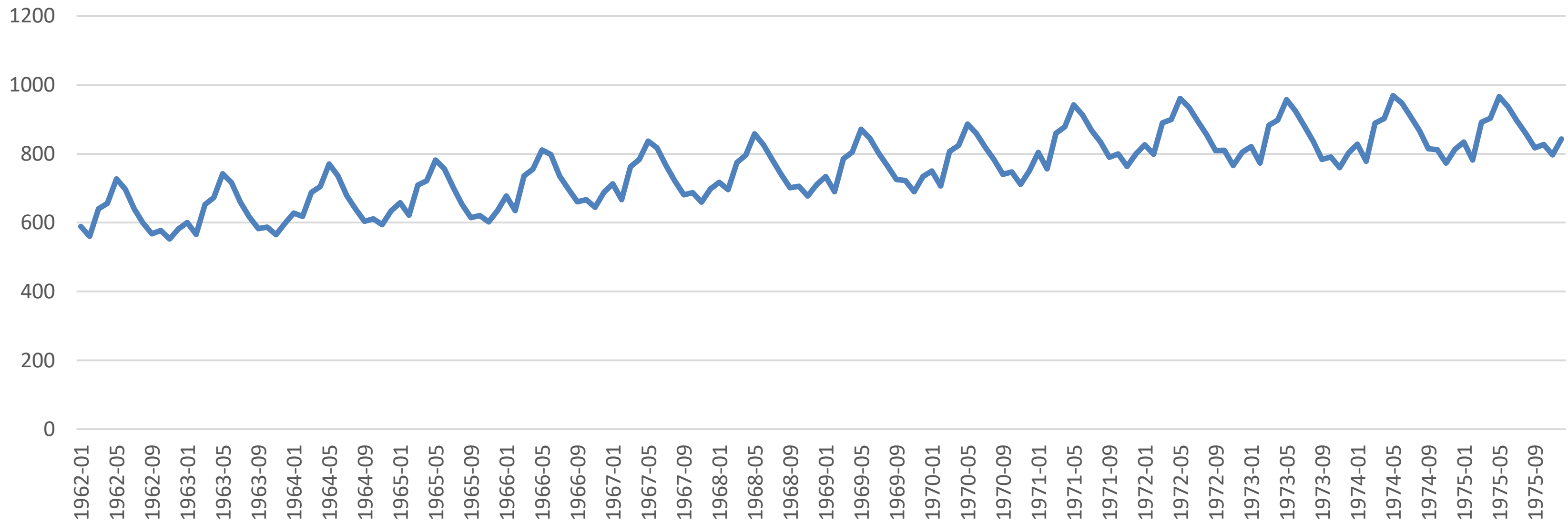
Agenda

1. Understanding Time Series Data
2. ARIMA analysis

Time Series Data

A time series can be defined as a set of data dependent on time. Time acts as an independent variable to estimate dependent variables.
Mathematically, a time series is a set of observation taken at specified times.

A time series defined by the values Y1, Y2.. of variable Y at times t1, t2.. is given by Y=F(t)



Time Series Data

Time series: Series of variables which are measured sequentially in time over some interval.

Time series analysis: To infer the impact of past data points on a series to predict the future data points.

- To describe the features of the time series pattern.
- To describe how two time series can interact.
- To predict future values of the series.
- To serve as a standard for a variable that measures the quality of product.

Importance of Time Series Analysis (TSA)

- Business Forecasting
- Understanding Past Behaviour
- Planning of future operations
- Evaluate current accomplishments

Components of Time Series Analysis

Trend

Seasonality

Irregular

Cyclical Patterns

Components of Time Series Analysis

Trend

- Gradual shift or movement to relatively higher or lower values over a long period of time
- When the time series analysis shows a general pattern, that is upward, we call it uptrend
- When the trend pattern exhibits a general pattern, that is down we call it a downtrend
- If there were no trend, we call it horizontal trend or stationary trend

Components of Time Series Analysis

- Upward or downward swings
- Repeating pattern within a fixed time period
- Usually observed within one year
- For example: If you live in a country with cold winters and hot summers, your electricity bill goes high in summers and low in winters because of the air conditioning costs

Seasonality

Components of Time Series Analysis

- Repeating up and down movements
- Usually go over more than a year of time
- Don't have a fixed period
- Much harder to predict

Cyclical Patterns

Components of Time Series Analysis

Irregular

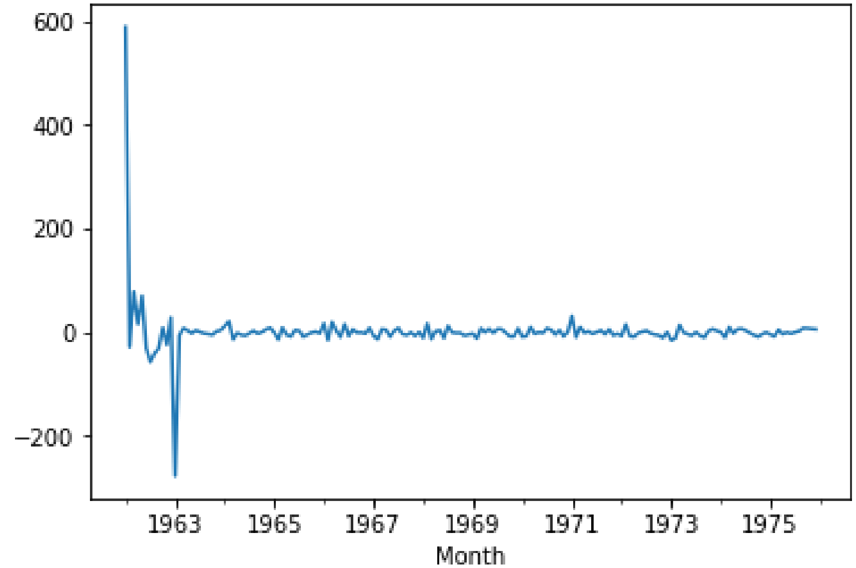
- Erratic, unsystematic, 'residual' fluctuations
- Short duration and nonrepeating
- Due to random variation or unforeseen events
- Presence of white noise

White Noise

- Describes the assumption that each element in a series is a random draw from a population

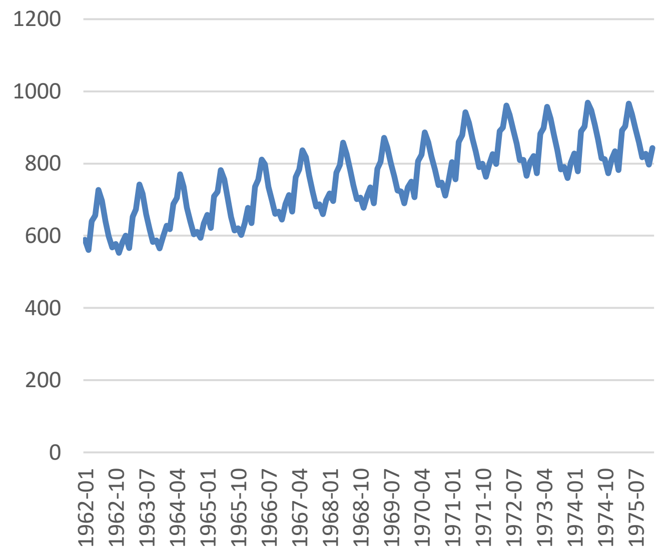
- Zero mean and constant variance

- Autoregressive (AR) and Moving Average (MA) models correct for violations of this white noise assumption



Stationary – An Example

- Here, the time series variable is not stationary as there is an increasing trend and it’s oscillating over time



Approaches to remove Non-Stationarity

Detrending

A variable can be detrended by regressing the variable on a time trend and obtaining the residuals:

yt = mu + Bt + et

Differencing

Uses the concept of differenced variable:
delta y = yt - y(t-1), for first order differences. The variable yt is integrated of order one, denoted I(1), if taking a first difference, producing a stationary process

Types of Models

There are two basic types of **time domain** models.

ARIMA

Autoregressive Integrated Moving Average

- Relate the present value of a series to past values and past prediction errors

Ordinary Regression Models

This uses time indices as x-variables

- These can be helpful for an initial description of the data and form the basis of several simple forecasting methods

ARIMA

- The Autoregressive Integrated Moving Average (ARIMA) method models the sequence as a linear function of the differenced observations and residual errors at prior time steps.

- It combines both Autoregression (AR) and Moving Average (MA) models as well as a differencing pre-processing step of the sequence to make the sequence stationary, called integration (I).

- The notation for the model involves specifying the order for the AR(p), I(d), and MA(q) models as parameters to an ARIMA function, e.g. ARIMA(p, d, q). An ARIMA model can also be used to develop AR, MA, and ARMA models.

- The method is suitable for univariate time series with trend and without seasonal components.

```
from statsmodels.tsa.arima_model import ARIMA
from random import random
# contrived dataset
data = [x + random() for x in range(1, 100)]
# fit model
model = ARIMA(data, order=(1, 1, 1))
model_fit = model.fit(disp=False)
# make prediction
yhat = model_fit.predict(len(data), len(data), typ='levels')
print(yhat)
```

ARIMA (p, d, q) denotes an ARMA model with **p autoregressive lags**, **q moving average** lags and **difference** in the order of **d**

Dickey Fuller Test for Stationarity

- You can estimate the above model for stationarity by testing the significance of the Y coefficient:
- If the null hypothesis is not rejected, Y* = 0, then yt is not stationary
- Difference the variable and repeat the test to see if the differenced variable is stationary
- If the null hypothesis is rejected, y* > 0, then yt is stationary

ACF (Auto Correlation Function)

1

ACF is the proportion of the covariance of Yt and Yt-k to the variance of a dependent variable Yt:

ACF (k) = Cov(Yt, Yt-k)/Var(Yt)

2

Gives the gross correlation between Yt and Yt-k

3

For an AR(1) model, the ACF(k) = Y^k

PACF (Partial Auto Correlation Function)

1 Simple correlation between Yt and Yt-k minus the part explained by the intervening lags

2 For an AR(1) model, the PACF is Y for the first lag

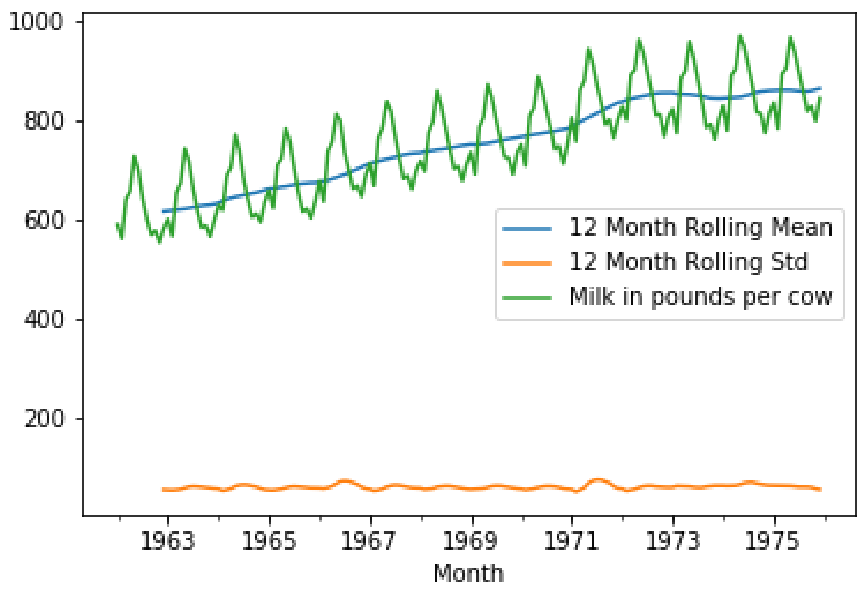
ARIMA Code and Data

Need to find the time series trend for cow milk production in pounds over the period of time

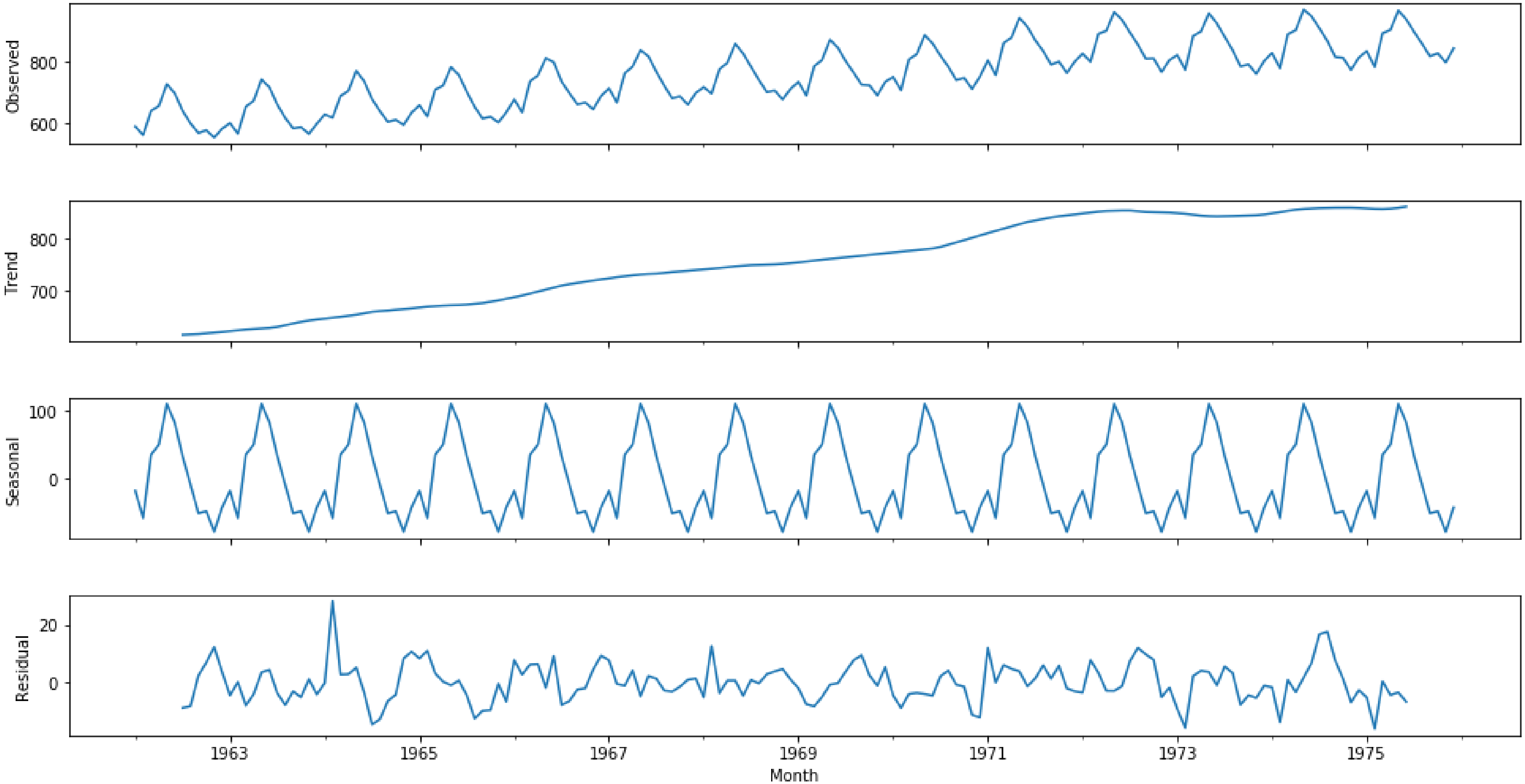
Index	Milk in pounds per cow
1962-11-01 00:00:00	553
1962-02-01 00:00:00	561
1963-11-01 00:00:00	565
1963-02-01 00:00:00	566
1962-09-01 00:00:00	568
1962-10-01 00:00:00	577
1962-12-01 00:00:00	582
1963-09-01 00:00:00	583
1963-10-01 00:00:00	587
1962-01-01 00:00:00	589
1964-11-01 00:00:00	594
1963-12-01 00:00:00	598
1962-08-01 00:00:00	599
1963-01-01 00:00:00	600
1965-11-01 00:00:00	602
1964-09-01 00:00:00	604
1964-10-01 00:00:00	611
1965-09-01 00:00:00	615

ARIMA Code and Data

```
import numpy as np
import pandas as pd
import statsmodels.api as sm

import matplotlib.pyplot as plt

import os
os.chdir('Give Path')
df = pd.read_csv('monthly-milk-production-pounds-p.csv')

timeseries = df['Milk in pounds per cow']

timeseries.rolling(12).mean().plot(label='12 Month Rolling Mean')
timeseries.rolling(12).std().plot(label='12 Month Rolling Std')
timeseries.plot()
plt.legend()


timeseries.rolling(12).mean().plot(label='12 Month Rolling Mean')
timeseries.plot()
plt.legend()
```



ETS Model


Observed
800
600
Trend
800
700
Seasonal
100
0
Residual
20
0
1963
1965
1967
1969
1971
1973
1975
Month

Dickey Fuller Test

```python
# Store in a function for later use!
def adf_check(time_series):
    """
    Pass in a time series, returns ADF report
    """
    result = adfuller(time_series)
    print('Augmented Dickey-Fuller Test:')
    labels = ['ADF Test Statistic','p-value','#Lags Used','Number of Observations Used']

    for value,label in zip(result,labels):
        print(label+' : '+str(value) )

    if result[1] <= 0.05:
        print("strong evidence against the null hypothesis, reject the null hypothesis. Data has no unit root and is stationary")
    else:
        print("weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary ")
```

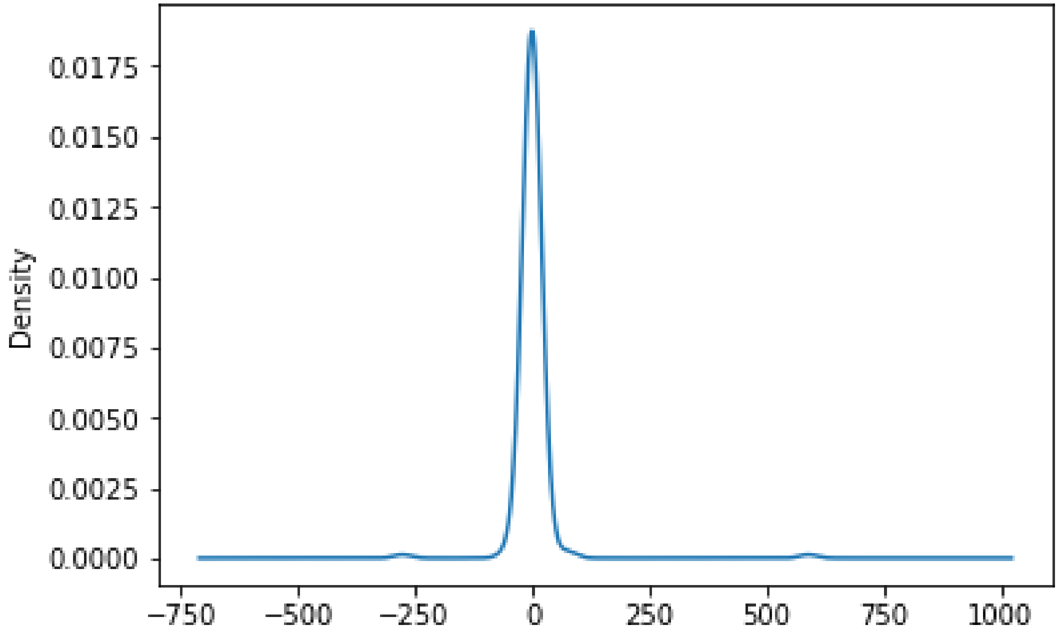
ARIMA Model

```
from statsmodels.tsa.arima_model import ARIMA
# We have seasonal data!
model = sm.tsa.statespace.SARIMAX(df['Milk in pounds per cow'],order=(0,1,0), seasonal_order=(1,1,1,12))
results = model.fit()
print(results.summary())


results.resid.plot()


results.resid.plot(kind='kde')


df['forecast'] = results.predict(start = 150, end= 168, dynamic= True)
df[['Milk in pounds per cow','forecast']].plot(figsize=(12,8))


df.tail()


from pandas.tseries.offsets import DateOffset

future_dates = [df.index[-1] + DateOffset(months=x) for x in range(0,24) ]
```

ARIMA Model

```
from statsmodels.tsa.arima_model import ARIMA
# We have seasonal data!
model = sm.tsa.statespace.SARIMAX(df['Milk in pounds per cow'],order=(0,1,0), seasonal_order=(1,1,1,12))
results = model.fit()
print(results.summary())


results.resid.plot()


results.resid.plot(kind='kde')


df['forecast'] = results.predict(start = 150, end= 168, dynamic= True)
df[['Milk in pounds per cow','forecast']].plot(figsize=(12,8))


df.tail()


from pandas.tseries.offsets import DateOffset

future_dates = [df.index[-1] + DateOffset(months=x) for x in range(0,24) ]
```

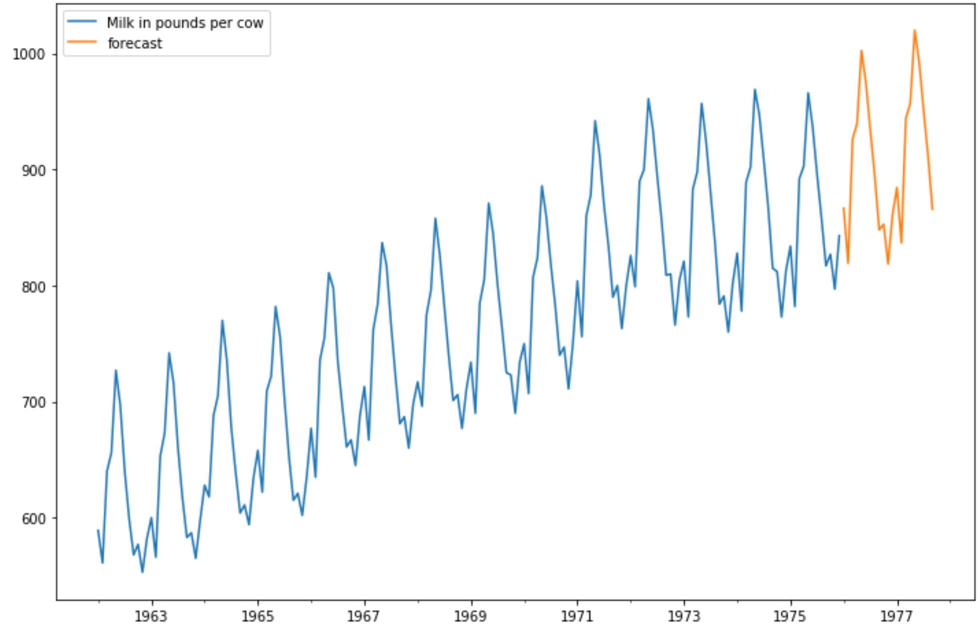
ARIMA Model



Density
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
−750
−500
−250
0
250
500
750
1000

ARIMA Model

```
future_dates

future_dates_df = pd.DataFrame(index=future_dates[1:],columns=df.columns)


future_df = pd.concat([df,future_dates_df])

future_df.head()

future_df.tail()

future_df['forecast'] = results.predict(start = 168, end = 188, dynamic= True)
future_df[['Milk in pounds per cow', 'forecast']].plot(figsize=(12, 8))
```



Cognixia
A Collabera LEARNING SOLUTIONS COMPANY

THANK YOU

www.cognixia.com