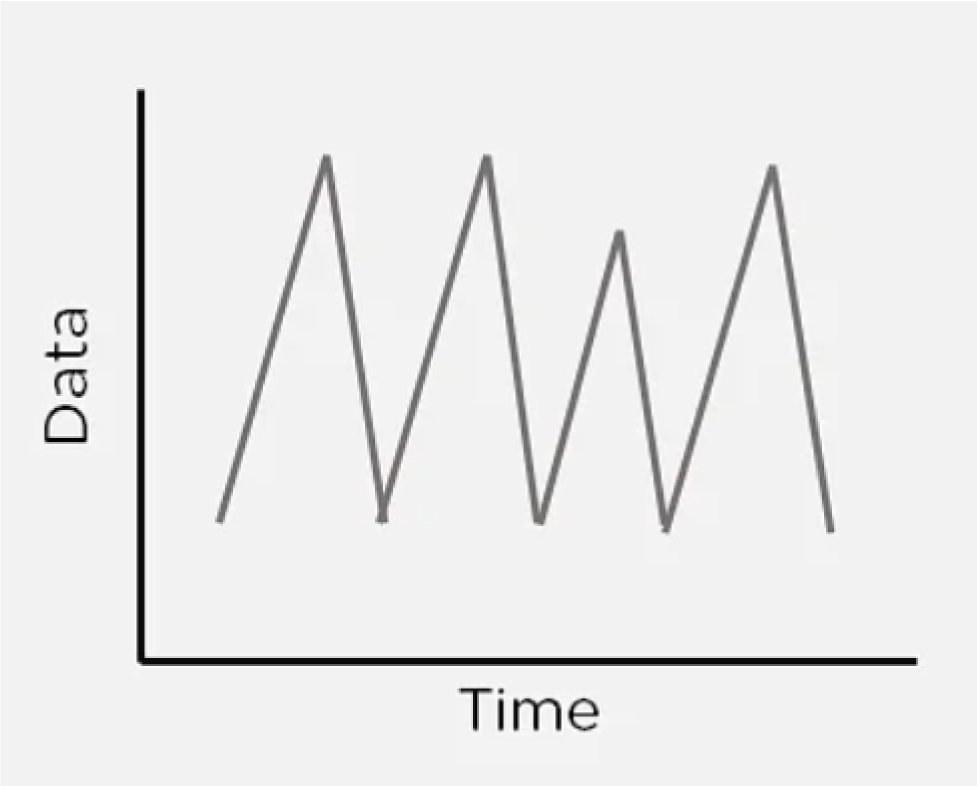
What is Time Series Analysis?

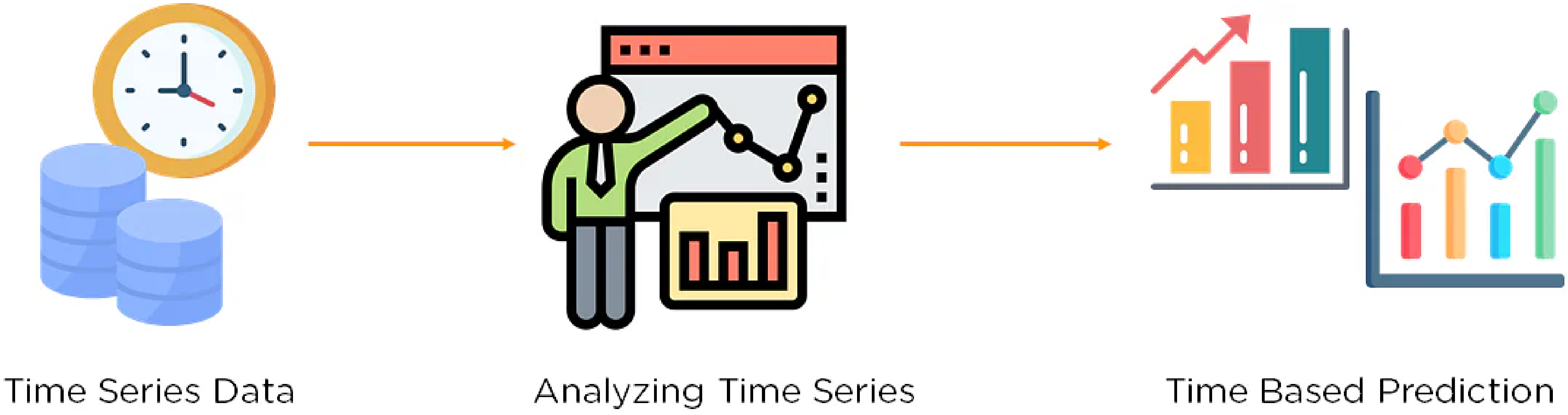
Sometimes data changes over time. This data is called time-dependent data. Given time-dependent data,we can analyze the past to predict the future.

The future prediction will also include time as a variable, and the output will vary with time. Using time-dependent data, we can find patterns that repeat over time.

A Time Series is a set of observations that are collected after regular intervals of time. If plotted, the Time series would always have one of its axes as time.



Time Series Analysis in Python considers data collected over time might have some structure; hence it analyses Time Series data to extract its valuable characteristics.



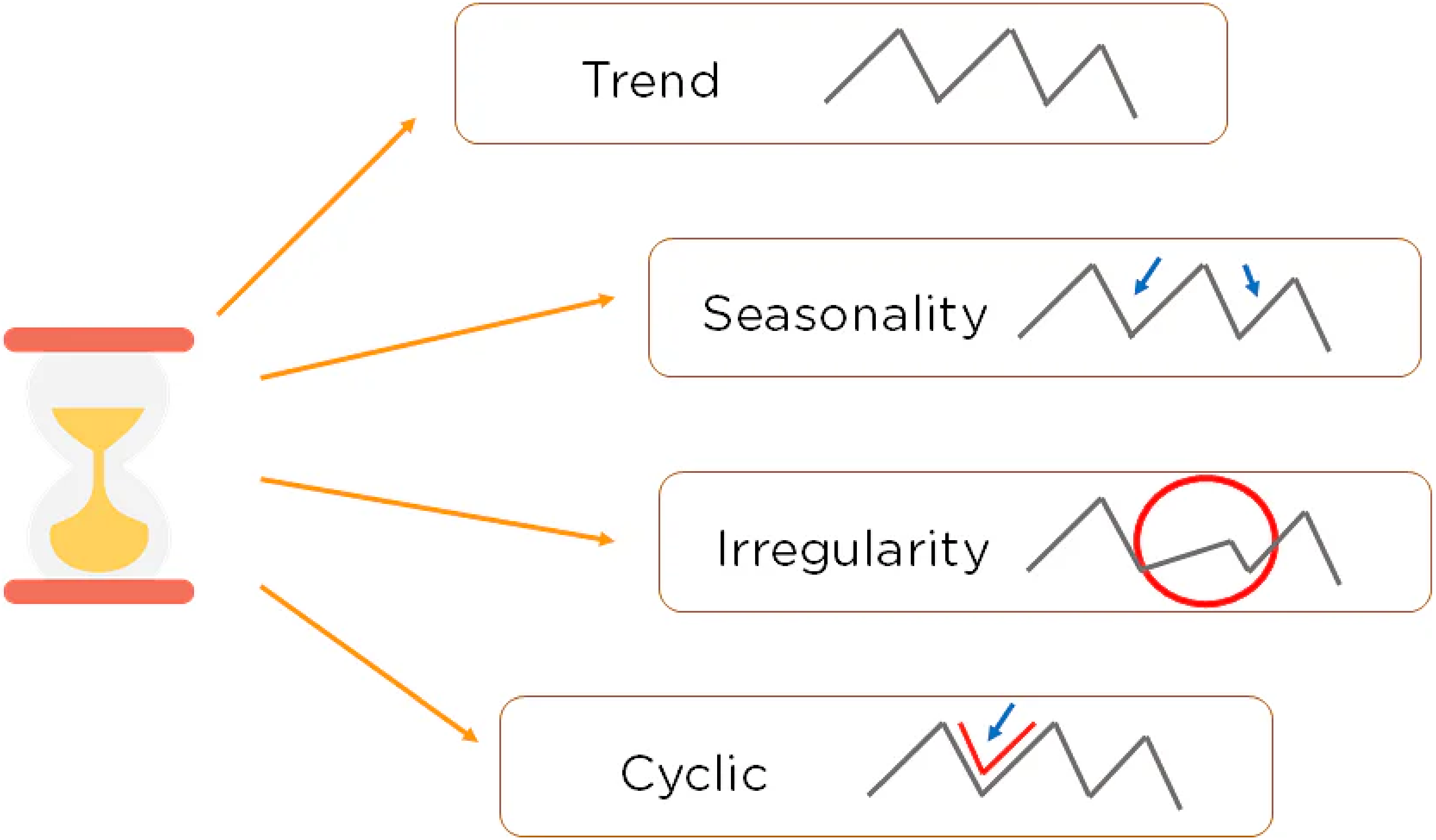
Consider the running of a bakery. Given the data of the past few months, we can predict what items we need to bake at what time.

The morning crowd would need more bread items, like bread rolls, croissants, breakfast muffins, etc.

At night, people may come in to buy cakes and pastries or other dessert items.

Using time series analysis, we can predict items popular during different times and even different seasons.

What Are the Different Components of Time Series Analysis?



Trend: The Trend shows the variation of data with time or the frequency of data. Using a Trend, you can see how your data increases or decreases over time. The data can increase, decrease, or remain stable. Over time, population, stock market fluctuations, and production in a company are all examples of trends.

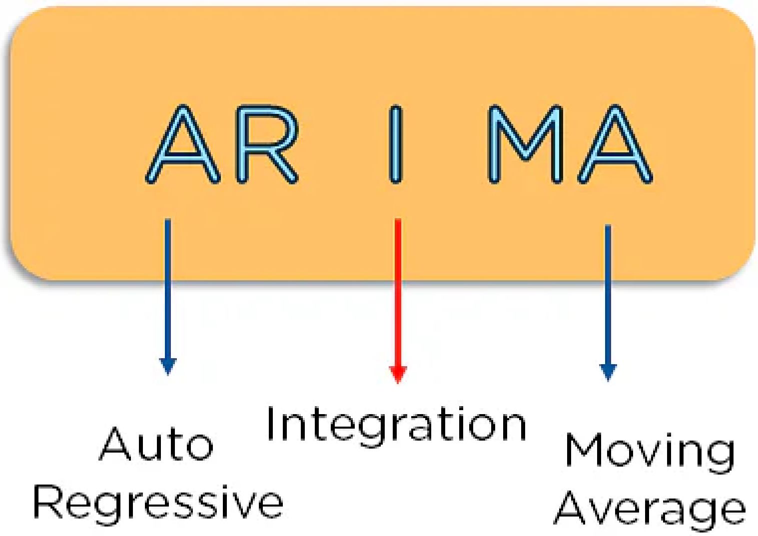
Seasonality: Seasonality is used to find the variations which occur at regular intervals of time. Examples are festivals, conventions, seasons, etc. These variations usually happen around the same time period and affect the data in specific ways which you can predict.

Irregularity: Fluctuations in the time series data do not correspond to the trend or seasonality. These variations in your time series are purely random and usually caused by unforeseeable circumstances, such as a sudden decrease in population because of a natural calamity.

Cyclic: Oscillations in time series which last for more than a year are called cyclic. They may or may not be periodic.

Stationary: A time series that has the same statistical properties over time is stationary. The properties remain the same anywhere in the series. Your data needs to be stationary to perform time-series analysis on it. A stationary series has a constant mean, variance, and

ARIMA Model ARIMA Model stands for Auto-Regressive Integrated Moving Average. It is used to predict the future values of a time series using its past values and forecast errors. The below diagram shows the components of an ARIMA model:

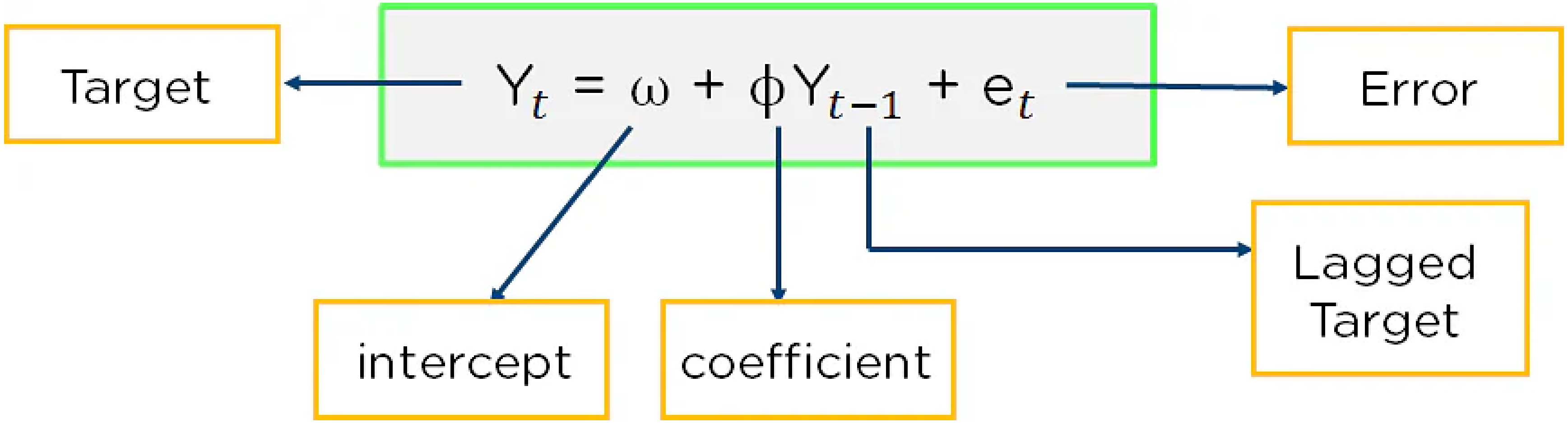


Auto Regressive Model

Auto-Regressive models predict future behavior using past behavior where there is some correlation between past and future data.

The formula below represents the autoregressive model.

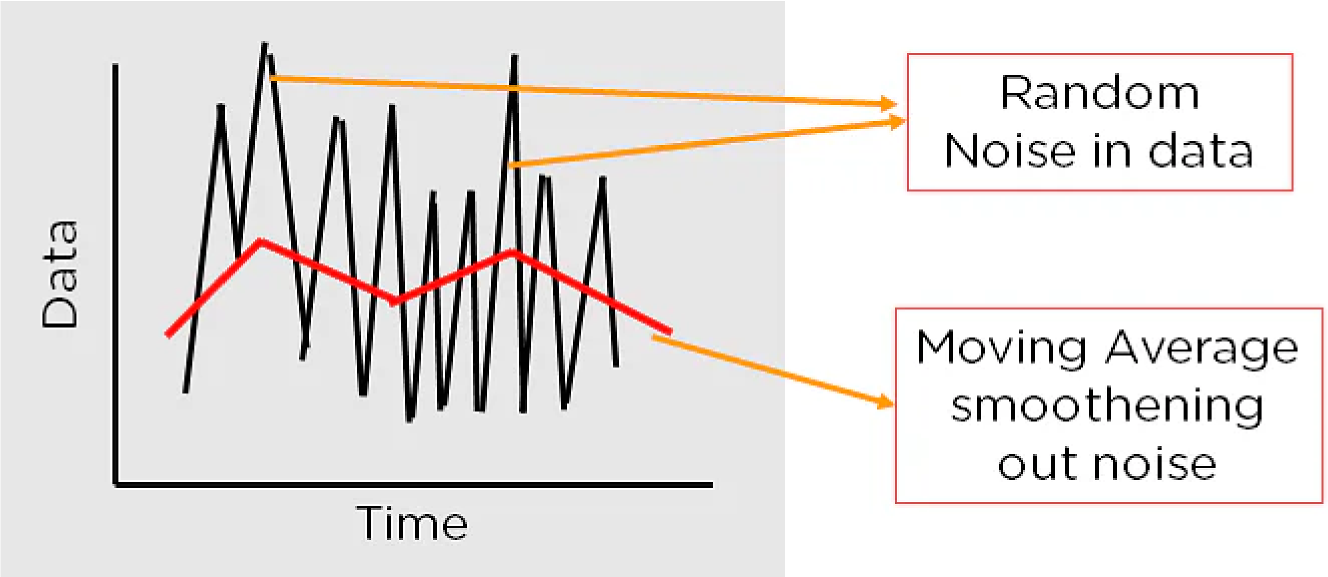
It is a modified version of the slope formula with the target value being expressed as the sum of the intercept, the product of a coefficient and the previous output, and an error correction term.



Moving Average

Moving Average is a statistical method that takes the updated average of values to help cut down on noise. It takes the average over a specific interval of time. You can get it by taking different subsets of your data and finding their respective averages.

we first consider a bunch of data points and take their average. then find the next average by removing the first value of the data and including the next value of the series.



Integration

Integration is the difference between present and previous observations. It is used to make the time series stationary.

Each of these values acts as a parameter for our ARIMA model. Instead of representing the ARIMA model by these various operators and models, you use parameters to represent them. These parameters are:

p: Previous lagged values for each time point. Derived from the Auto-Regressive Model.

q: Previous lagged values for the error term. Derived from the Moving Average.

d: Number of times data is differenced to make it stationary. It is the number of times it performs integration.

Time Series Analysis in Python

Now you will see how to perform Time Series Analysis in Python. we will use a shampoo dataset that details the monthly shampoo sales over three years. we will start by importing the necessary modules:

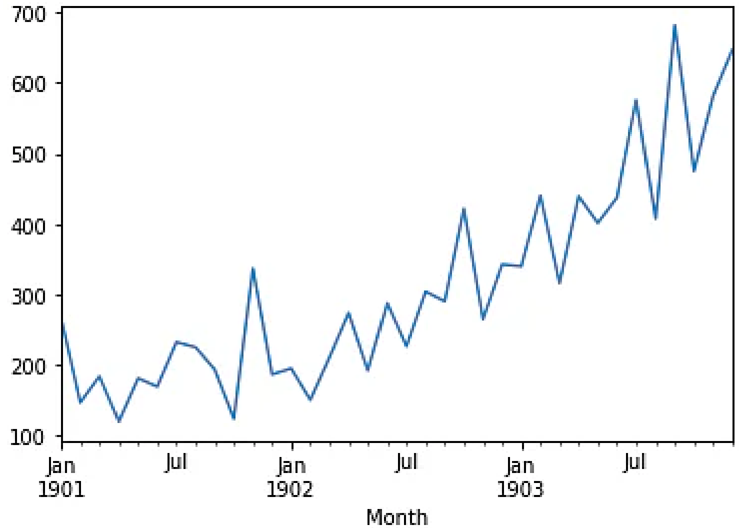
```
from pandas import read_csv
from pandas import datetime
from matplotlib import pyplot
from pandas.plotting import autocorrelation_plot
from pandas import DataFrame
from statsmodels.tsa.arima_model import ARIMA
```

Next, read in the data, then print and plot it to see how it looks.

```
def parser(x):
	return datetime.strptime('190'+x, '%Y-%m')

series = read_csv('/home/shrishti/shampoo.csv', header=0, parse_dates=[0], index_col=0, squeeze=True, date_parser=parser)
print(series.head())
series.plot()
pyplot.show()
```

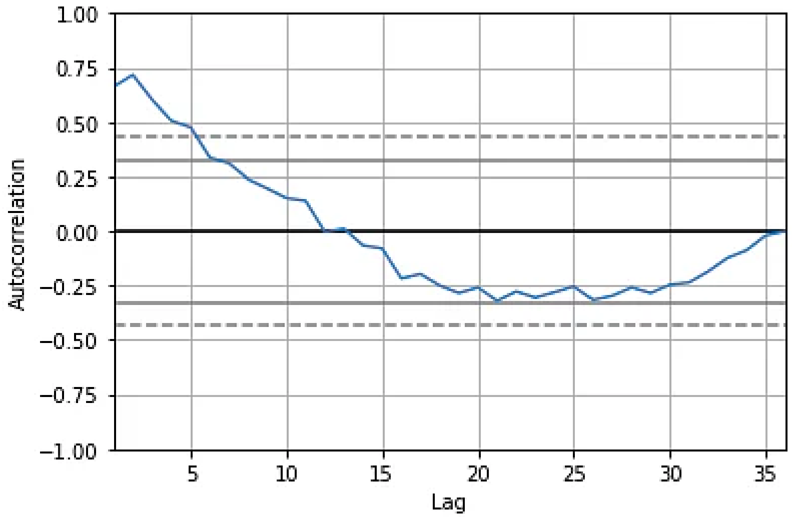
```
Month
1901-01-01    266.0
1901-02-01    145.9
1901-03-01    183.1
1901-04-01    119.3
1901-05-01    180.3
Name: Sales, dtype: float64
```



The figure below shows the values in your data and the trend in it.

Now, plot the autocorrelation in the data.

```
autocorrelation_plot(series)
pyplot.show()
#we can see that there is a positive correlation with the first 10-to-12 lags
#that is perhaps significant for the first 5 lags
```



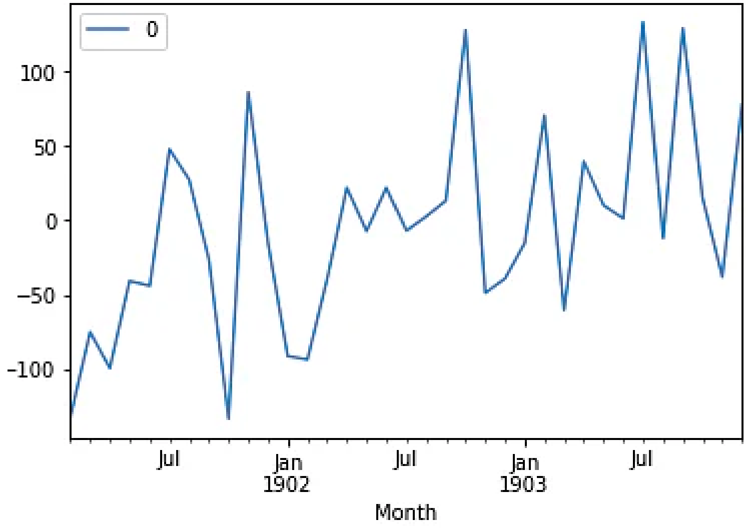
Now, fit your data to your model and find the residual error.

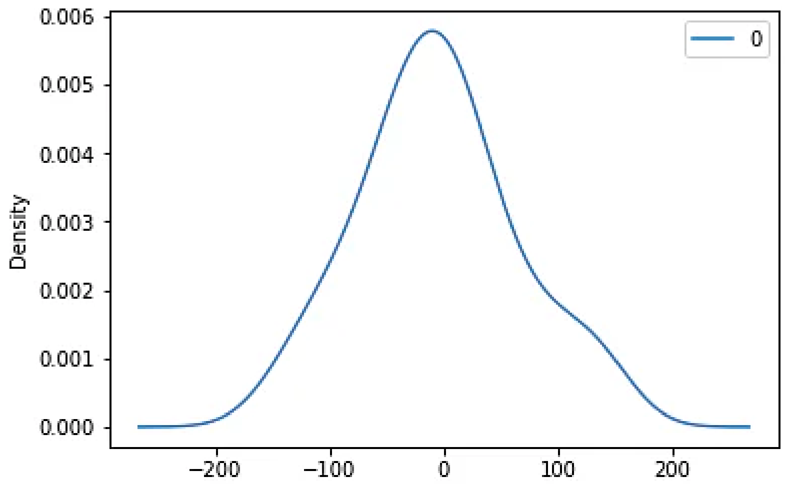
It is obtained by separating data values from the mean of the data. This helps you find out if variations in your data are huge.

```
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())
```

The diagram below shows the prediction of the ARIMA model and the trend that it has predicted. It is like the trend exhibited by your data.

```
                              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:                Thu, 12 Nov 2020   AIC                            406.340
Time:                        23:32:32   BIC                            417.227
Sample:                    02-01-1901   HQIC                           410.098
                         - 12-01-1903
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const            12.0649      3.652      3.304      0.001       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      0.028      -1.172      -0.068
ar.L3.D.Sales    -0.3606      0.295     -1.222      0.222      -0.939       0.218
ar.L4.D.Sales    -0.1252      0.280     -0.447      0.655      -0.674       0.424
ar.L5.D.Sales     0.1289      0.191      0.673      0.501      -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
-----------------------------------------------------------------------------
```


0
100
50
0
−50
−100
Jul
Jan
1902
Jul
Jan
1903
Jul
Month




```
              0
count   35.000000
mean    -5.495267
std     68.132879
min   -133.296649
25%    -42.477975
50%     -7.186677
75%     24.748283
max    133.237923
```

You also get a plot of your residual errors, as shown below.

```
we can see that the errors are Gaussian and are not centered aroun
```

we can see that the errors are Gaussian and are not centered around 0.