

[Start Here](#)[Blog](#)[Books](#)[About](#)[Contact](#)

Need help with LSTMs in Python? [Take the FREE Mini-Course.](#)

# How to Remove Trends and Seasonality with a Difference Transform in Python

by **Jason Brownlee** on July 10, 2017 in **Long Short-Term Memory Networks**



Time series datasets may contain trends and seasonality, which may need to be removed prior to modeling.

Trends can result in a varying mean over time, whereas seasonality can result in a changing variance over time, both which define a time series as being non-stationary. Stationary datasets are those that have a stable mean and variance, and are in turn much easier to model.

Differencing is a popular and widely used data transform for making time series data stationary.

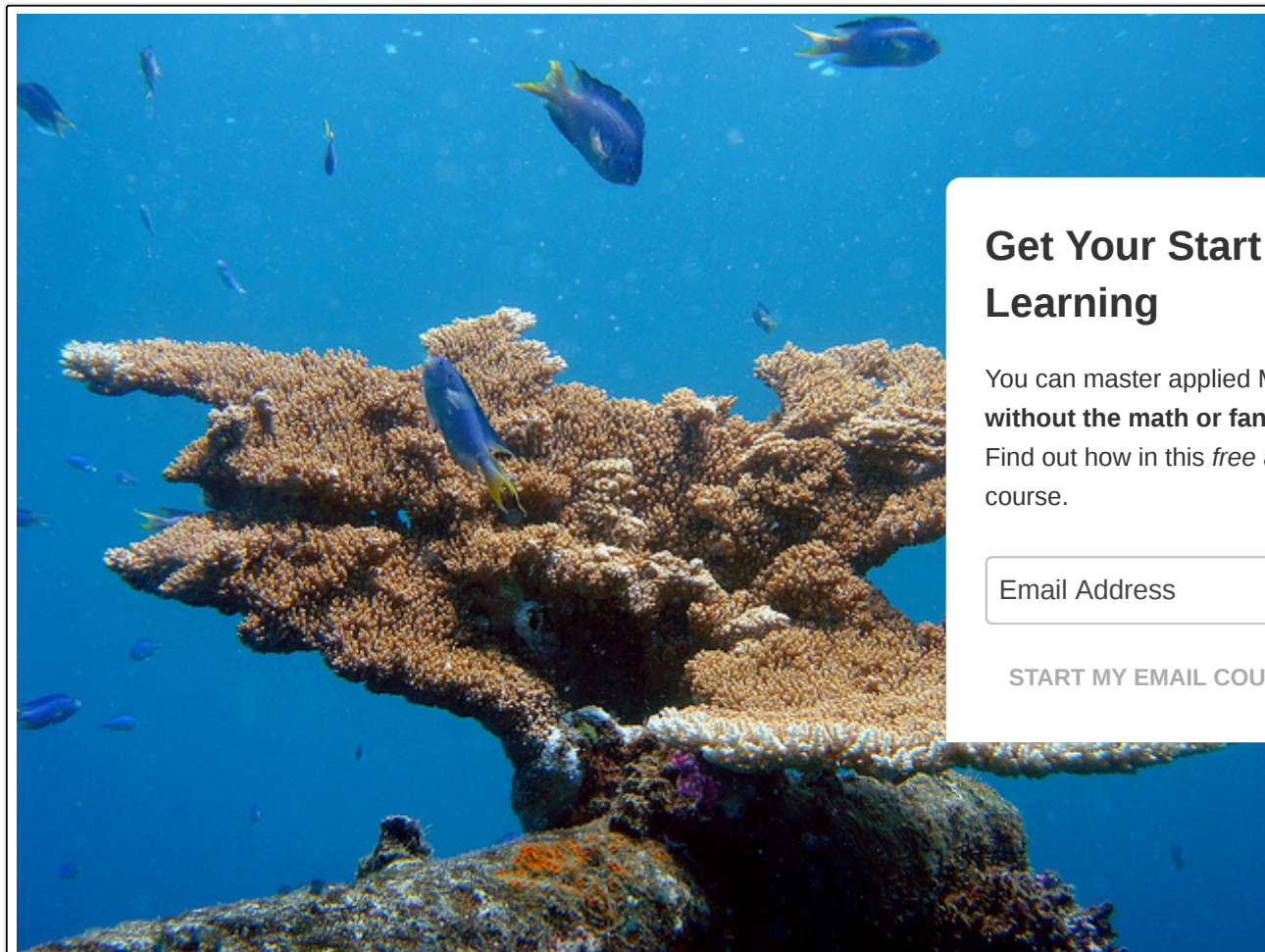
In this tutorial, you will discover how to apply the difference operation to your time series data with P

[Get Your Start in Machine Learning](#)

After completing this tutorial, you will know:

- The contrast between a stationary and non-stationary time series and how to make a series stationary with a difference transform.
- How to apply the difference transform to remove a linear trend from a series.
- How to apply the difference transform to remove a seasonal signal from a series.

Let's get started.



How to Remove Trends and Seasonality with a Difference Transform in Python

Photo by NOAA, some rights reserved.

## Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning

# Tutorial Overview

This tutorial is divided into 4 parts; they are:

1. Stationarity
2. Difference Transform
3. Differencing to Remove Trends
4. Differencing to Remove Seasonality

## Stationarity

Time series is different from more traditional classification and regression predictive modeling problems.

The temporal structure adds an order to the observations. This imposed order means that important observations need to be handled specifically.

For example, when modeling, there are assumptions that the summary statistics of observations are constant over time, with this expectation as the time series being stationary.

These assumptions can be easily violated in time series by the addition of a trend, seasonality, and cyclical effects.

## Stationary Time Series

The observations in a stationary time series are not dependent on time.

Time series are stationary if they do not have trend or seasonal effects. Summary statistics calculate the mean or the variance of the observations.

When a time series is stationary, it can be easier to model. Statistical modeling methods assume or require the time series to be stationary.

## Non-Stationary Time Series

Observations from a non-stationary time series show seasonal effects, trends, and other structures that depend on the time index.

### Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Summary statistics like the mean and variance do change over time, providing a drift in the concepts a model may try to capture.

Classical time series analysis and forecasting methods are concerned with making non-stationary time series data stationary by identifying and removing trends and removing stationary effects.

## Making Series Data Stationary

You can check if your time series is stationary by looking at a line plot of the series over time.

Sign of obvious trends, seasonality, or other systematic structures in the series are indicators of a non-stationary series.

A more accurate method would be to use a statistical test, such as the Dickey-Fuller test.

Should you make your time series stationary?

Generally, yes.

If you have clear trend and seasonality in your time series, then model these components, remove the residuals.

“ If we fit a stationary model to data, we assume our data are a realization of a stationary process. We should check whether there is any evidence of a trend or seasonal effects and, if there is, remove them.”

— Page 122, [Introductory Time Series with R](#).

Statistical time series methods and even modern machine learning methods will benefit from the clearer signal in the data.

### Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**. Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

## Need help with LSTMs for Sequence Prediction?

Take my free 7-day email course and discover 6 different LSTM architectures (with sample code).

Click to sign-up and also get a free PDF Ebook version of

[Get Your Start in Machine Learning](#)

[Start Your FREE Mini-Course Now!](#)

## Difference Transform

Differencing is a method of transforming a time series dataset.

It can be used to remove the series dependence on time, so-called temporal dependence. This includes structures like trends and seasonality.

“Differencing can help stabilize the mean of the time series by removing changes in the level of a time series and so eliminating (or reducing) trend and seasonality.

— Page 215, [Forecasting: principles and practice](#).

Differencing is performed by subtracting the previous observation from the current observation.

```
1 difference(t) = observation(t) - observation(t-1)
```

Inverting the process is required when a prediction must be converted back into the original scale.

This process can be reversed by adding the observation at the prior time step to the difference value.

```
1 inverted(t) = differenced(t) + observation(t-1)
```

In this way, a series of differences and inverted differences can be calculated.

## Lag Difference

Taking the difference between consecutive observations is called a lag-1 difference.

The lag difference can be adjusted to suit the specific temporal structure.

For time series with a seasonal component, the lag may be expected to be the period (width) of the season.

### Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree**.

Find out how in this *free* and *practical* email course.

[START MY EMAIL COURSE](#)[Get Your Start in Machine Learning](#)

## Difference Order

Some temporal structure may still exist after performing a differencing operation, such as in the case of a nonlinear trend.

As such, the process of differencing can be repeated more than once until all temporal dependence has been removed.

The number of times that differencing is performed is called the difference order.

## Calculating Differencing

We can difference the dataset manually.

This involves developing a new function that creates a differenced dataset. The function would loop through a provided series and calculate the differenced values at the specified interval or lag.

The function below named `difference()` implements this procedure.

```
1 # create a differenced series
2 def difference(dataset, interval=1):
3     diff = list()
4     for i in range(interval, len(dataset)):
5         value = dataset[i] - dataset[i - interval]
6         diff.append(value)
7     return Series(diff)
```

We can see that the function is careful to begin the differenced dataset after the specified interval to a default interval or lag value of 1 is defined. This is a sensible default.

One further improvement would be to also be able to specify the order or number of times to perform the differencing operation.

The function below named `inverse_difference()` inverts the difference operation for a single forecast. It requires that the real observation value for the previous time step also be provided.

```
1 # invert differenced forecast
2 def inverse_difference(last_ob, value):
3     return value + last_ob
```

### Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

# Differencing to Remove Trends

In this section, we will look at using the difference transform to remove a trend.

A trend makes a time series non-stationary by increasing the level. This has the effect of varying the mean time series value over time.

The example below applies the difference() function to a contrived dataset with a linearly increasing trend.

```
1 # create a differenced series
2 def difference(dataset, interval=1):
3     diff = list()
4     for i in range(interval, len(dataset)):
5         value = dataset[i] - dataset[i - interval]
6         diff.append(value)
7     return diff
8
9 # invert differenced forecast
10 def inverse_difference(last_ob, value):
11     return value + last_ob
12
13 # define a dataset with a linear trend
14 data = [i+1 for i in range(20)]
15 print(data)
16 # difference the dataset
17 diff = difference(data)
18 print(diff)
19 # invert the difference
20 inverted = [inverse_difference(data[i], diff[i]) for i in range(len(diff))]
21 print(inverted)
```

Running the example first prints the contrived sequence with a linear trend. Next, the differenced data is printed for each time step. The length of this sequence is 19 instead of 20 as the difference for the first value in the sequence is not calculated.

Finally, the difference sequence is inverted using the prior values from the original sequence as the primer for each transform.

```
1 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
2 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
3 [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```

## Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

# Differencing to Remove Seasonality

In this section, we will look at using the difference transform to remove seasonality.

Seasonal variation, or seasonality, are cycles that repeat regularly over time.

“ A repeating pattern within each year is known as seasonal variation, although the term is applied more generally to repeating patterns within any fixed period.

— Page 6, [Introductory Time Series with R](#).

There are many types of seasonality. Some obvious examples include; time of day, daily, weekly, monthly, etc. Whether there is a seasonality component in your time series problem is subjective.

The simplest approach to determining if there is an aspect of seasonality is to plot and review your data. The presence of trend lines.

The example below applies the `difference()` function to a contrived seasonal dataset. The dataset includes a linear trend and a seasonal component.

```
1 from math import sin
2 from math import radians
3 from matplotlib import pyplot
4
5 # create a differenced series
6 def difference(dataset, interval=1):
7     diff = list()
8     for i in range(interval, len(dataset)):
9         value = dataset[i] - dataset[i - interval]
10        diff.append(value)
11    return diff
12
13 # invert differenced forecast
14 def inverse_difference(last_ob, value):
15     return value + last_ob
16
17 # define a dataset with a linear trend
18 data = [sin(radians(i)) for i in range(360)] + [sin(radians(i)) for i in range(360)]
19 pyplot.plot(data)
20 pyplot.show()
```

## Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

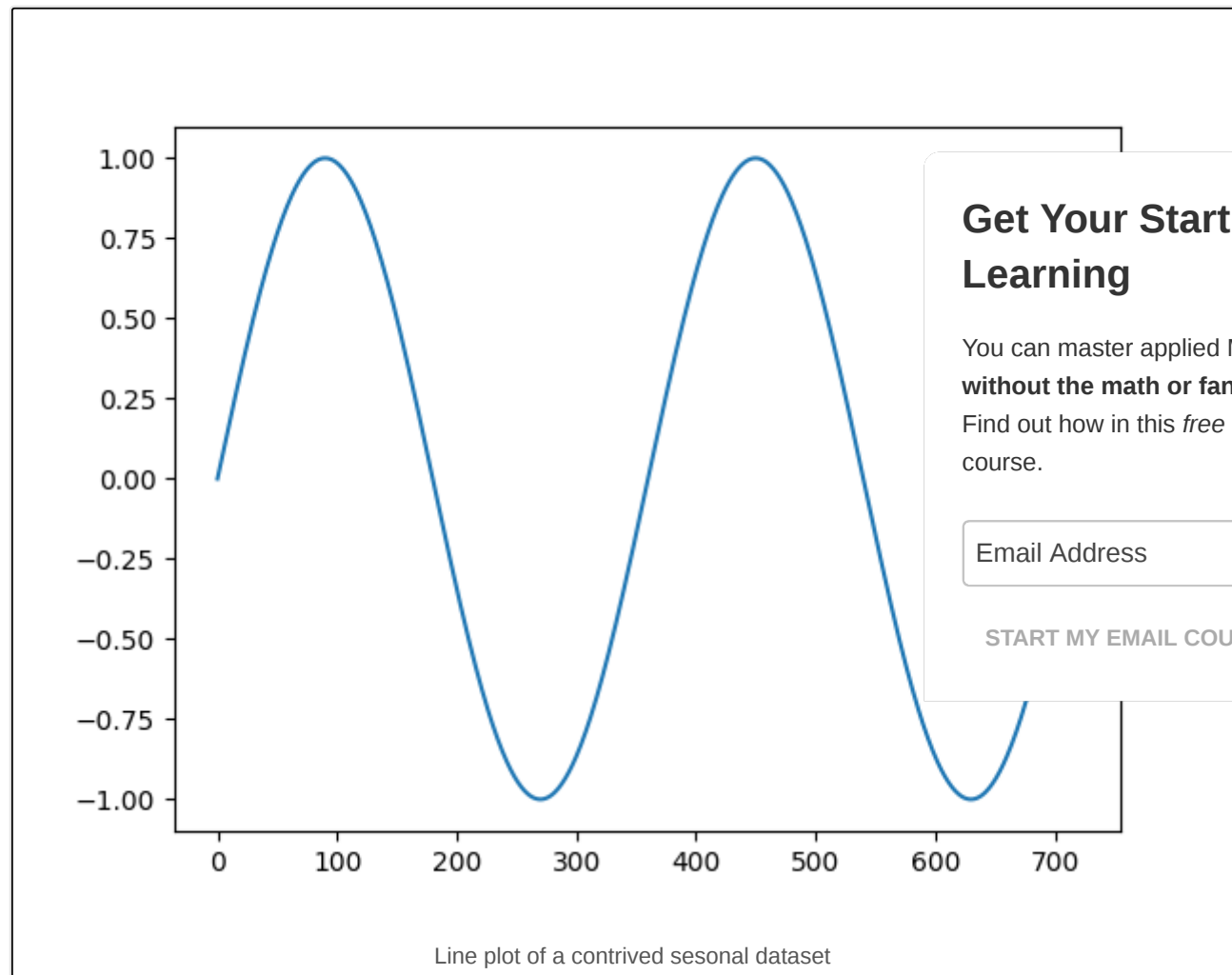
START MY EMAIL COURSE

Get Your Start in Machine Learning



```
21 # difference the dataset
22 diff = difference(data, 360)
23 pyplot.plot(diff)
24 pyplot.show()
25 # invert the difference
26 inverted = [inverse_difference(data[i], diff[i]) for i in range(len(diff))]
27 pyplot.plot(inverted)
28 pyplot.show()
```

Running the example first creates and plots the dataset of two cycles of the 360 time step series.



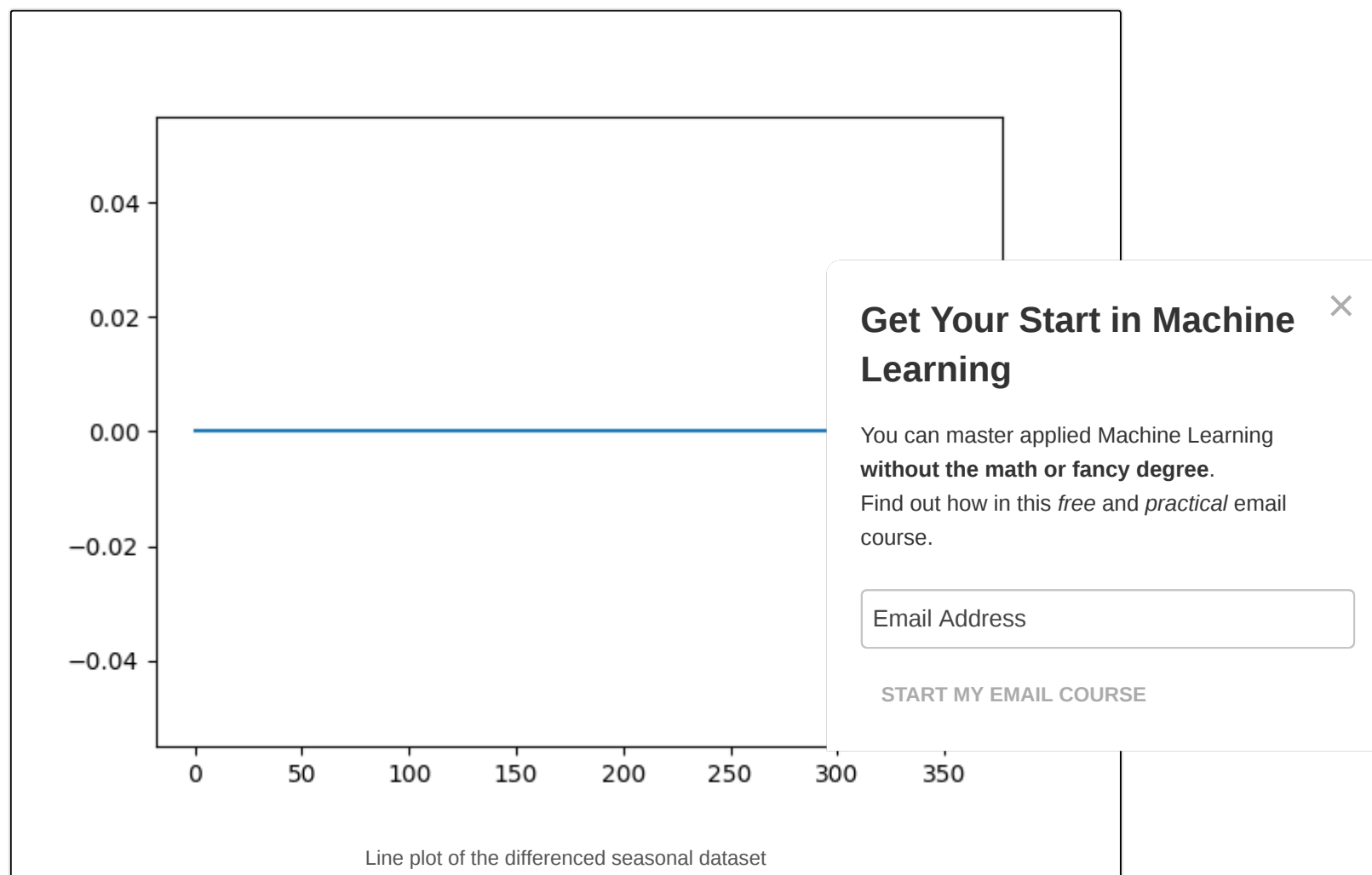
## Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

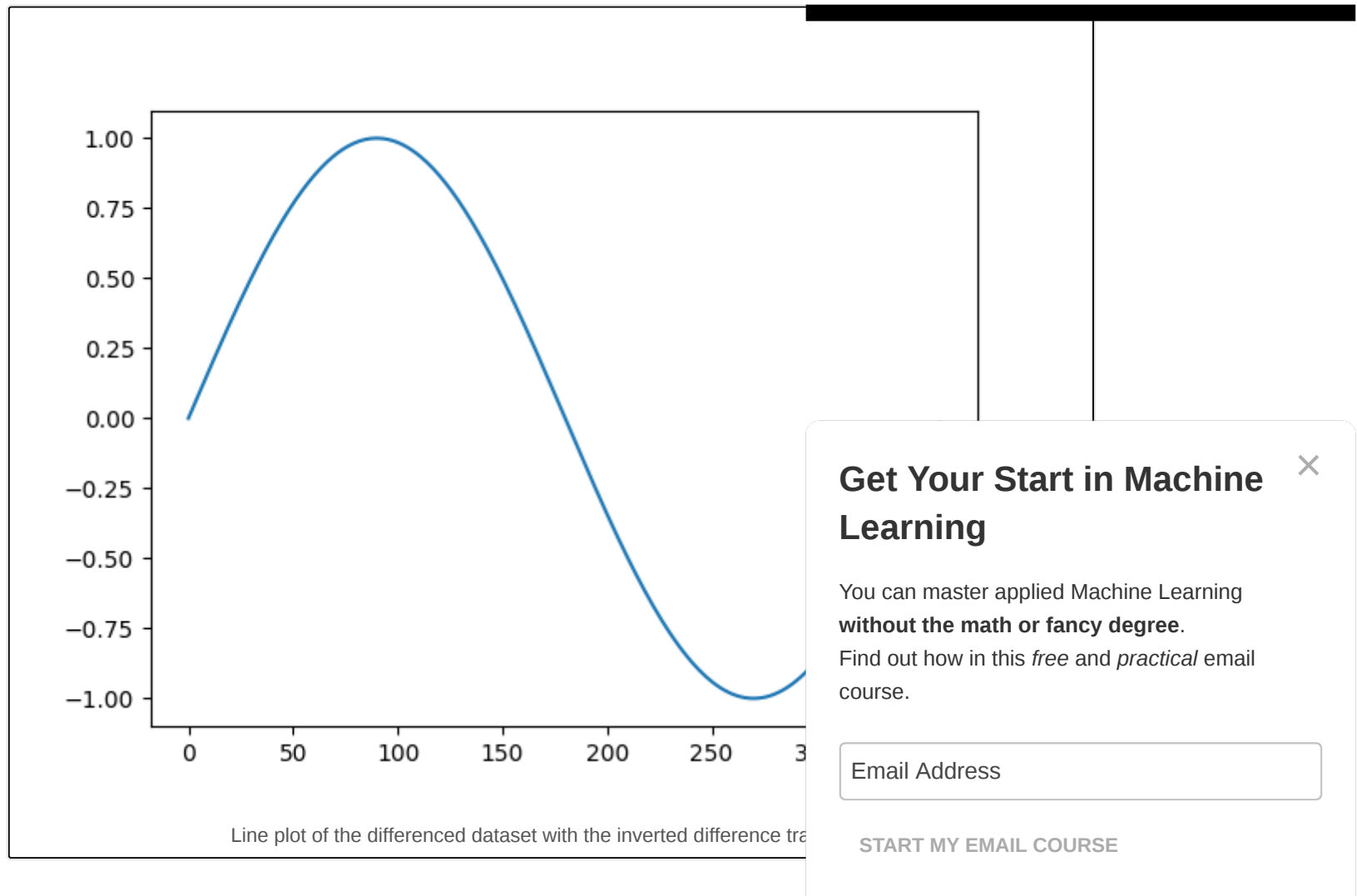
START MY EMAIL COURSE

Next, the difference transform is applied and the result is plotted. The plot shows 360 zero values with all seasonality signal removed.

In the de-trending example above, differencing was applied with a lag of 1, which means the first value was sacrificed. Here an entire cycle is used for differencing, that is 360 time steps. The result is that the entire first cycle is sacrificed in order to difference the second cycle.



Finally, the transform is inverted showing the second cycle with the seasonality restored.



## Further Reading

- [Stationary process](#) on Wikipedia
- [Seasonal Adjustment](#) on Wikipedia
- [How to Check if Time Series Data is Stationary with Python](#)
- [How to Difference a Time Series Dataset with Python](#)
- [How to Identify and Remove Seasonality from Time Series Data with Python](#)
- [Seasonal Persistence Forecasting With Python](#)

## Summary

In this tutorial, you discovered the distinction between stationary and non-stationary time series and how to use the difference transform to remove trends and seasonality with Python.

Specifically, you learned:

- The contrast between a stationary and non-stationary time series and how to make a series stationary with a difference transform.
- How to apply the difference transform to remove a linear trend from a series.
- How to apply the difference transform to remove a seasonal signal from a series.

Do you have any questions about making time series stationary?

Ask your questions in the comments and I will do my best to answer.

## Develop LSTMs for Sequence Prediction

### Develop Your Own LSTM model

...with just a few lines of python

Discover how in my new  
Long Short-Term Memory Networks

It provides **self-study tutorial**  
CNN LSTMs, Encoder-Decoder LSTMs, generative models, data

### Finally Bring LSTM Recurrent Neural Networks to Your Sequence Predictions Projects

Skip the Academics. Just Results.

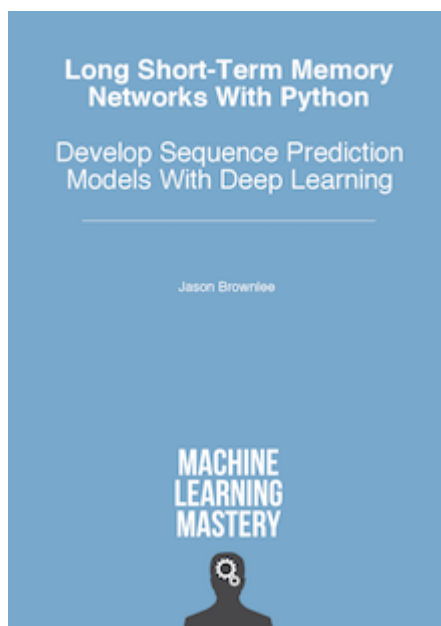
[Click to learn more.](#)

## Get Your Start in Machine Learning

You can master applied Machine Learning  
**without the math or fancy degree.**

Find out how in this *free* and *practical* email  
course.

START MY EMAIL COURSE





### About Jason Brownlee

Dr. Jason Brownlee is a husband, proud father, academic researcher, author, professional developer and a machine learning practitioner. He is dedicated to helping developers get started and get good at applied machine learning. [Learn more.](#)

[View all posts by Jason Brownlee](#) →

< How to Scale Data for Long Short-Term Memory Networks in Python

## 2 Responses to *How to Remove Trends and Seasonality with a Difference*



**James York** July 19, 2017 at 8:47 pm #

Excellent article, explains a lot. Just learned about standard deviations so this is a nice complement

I find it interesting how removing trends and seasonality resembles how some networks (such as convolutional) produce invariant outputs.

On that note, theres a tangentially related article that I thought you might enjoy – <http://news.mit.edu/2012/brain-waves-encode-rules-for-behavior-1121>

Earl Miller is doing some really fascinating work, and so are you.

**Jason Brownlee** July 20, 2017 at 6:20 am #

### Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

Get Your Start in Machine Learning



Nice!

## Leave a Reply

Name (required)

Email (will not be published) (required)

Website

### Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

## Welcome to Machine Learning Mastery

Hi, I'm Dr. Jason Brownlee.

My goal is to make practitioners like YOU awesome at applied machine learning.

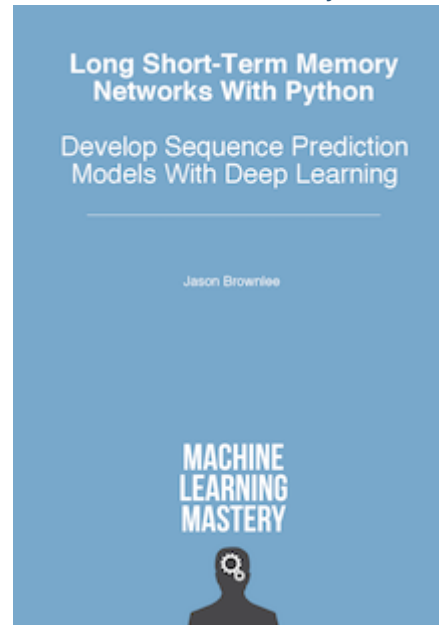
[Read More](#)[Get Your Start in Machine Learning](#)



## Deep Learning for Sequence Prediction

Cut through the math and research papers.  
Discover 4 Models, 6 Architectures, and 14 Tutorials.

Get Started With LSTMs in Python Today!



## Get Your Start in Machine Learning

You can master applied Machine Learning **without the math or fancy degree.**  
Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE

### POPULAR



Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras

JULY 21, 2016

Get Your Start in Machine Learning

**Your First Machine Learning Project in Python Step-By-Step**

JUNE 10, 2016

**Develop Your First Neural Network in Python With Keras Step-By-Step**

MAY 24, 2016

**Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras**

JULY 26, 2016

**How to Setup a Python Environment for Machine Learning and Deep Learning with Anaconda**

MARCH 13, 2017

**Time Series Forecasting with the Long Short-Term Memory Network in Python**

APRIL 7, 2017

**Multi-Class Classification Tutorial with the Keras Deep Learning Library**

JUNE 2, 2016

**Regression Tutorial with the Keras Deep Learning Library in Python**

JUNE 9, 2016

**Multivariate Time Series Forecasting with LSTMs in Keras**

AUGUST 14, 2017

**How to Implement the Backpropagation Algorithm From Scratch In Python**

NOVEMBER 7, 2016

## Get Your Start in Machine Learning



You can master applied Machine Learning **without the math or fancy degree.** Find out how in this *free* and *practical* email course.

[START MY EMAIL COURSE](#)



## Get Your Start in Machine Learning ×

You can master applied Machine Learning **without the math or fancy degree.**

Find out how in this *free* and *practical* email course.

START MY EMAIL COURSE