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

by **Jason Brownlee** on July 10, 2017 in **Deep Learning for Time Series**

Last Updated on August 5, 2019

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.

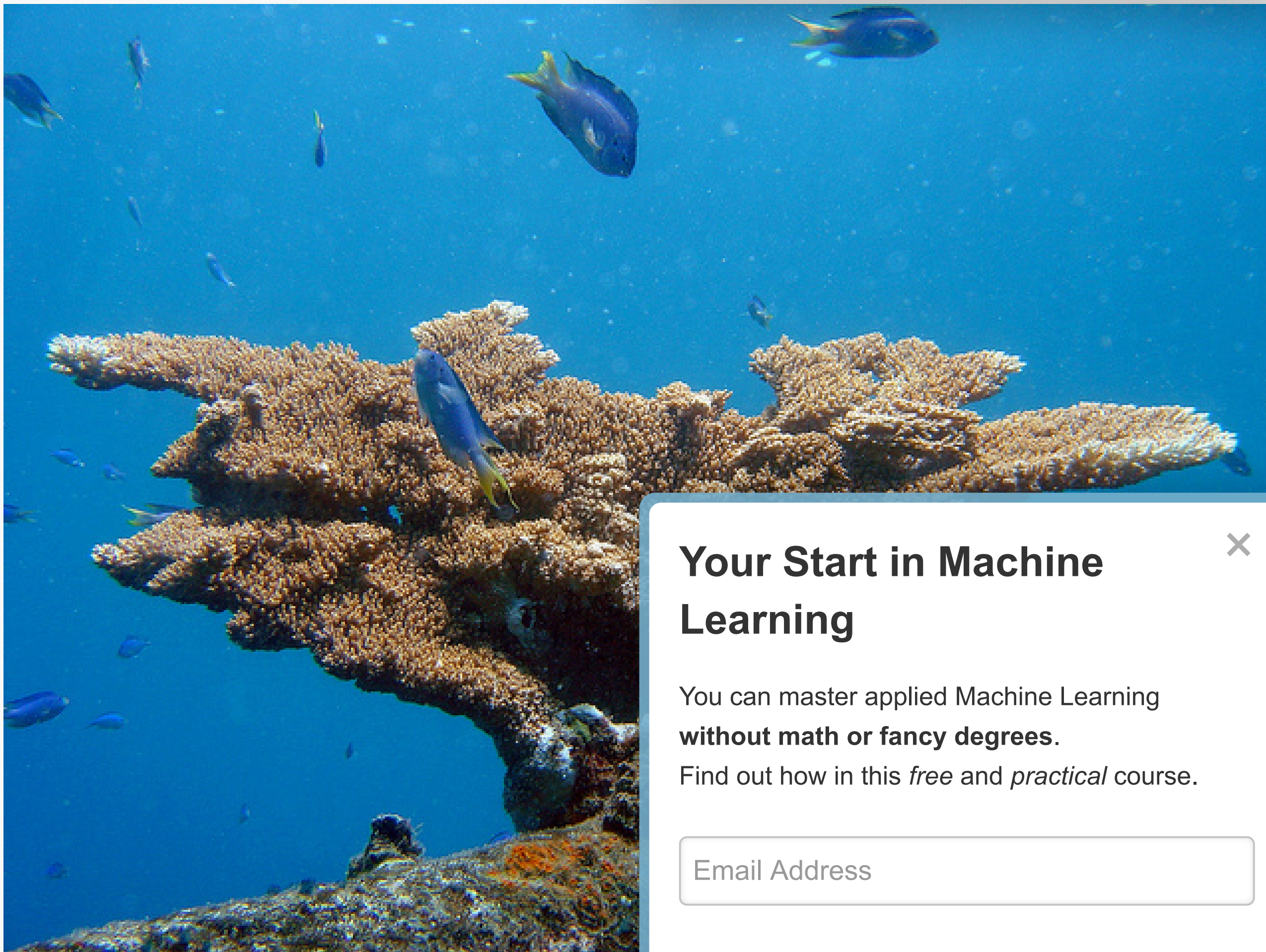
In this tutorial, you will discover how to apply the difference operation to your time series data with Python.

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.

Discover how to build models for multivariate and multi-step time series forecasting with LSTMs and more in my new book, with 25 step-by-step tutorials and full source code.

Let's get started.



How to Remove Trends and Seasonality
Photo by NOAA, so

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 assumptions about the consistency of those observations needs to be handled specifically.

For example, when modeling, there are assumptions that the summary statistics of observations are consistent. In time series terminology, we refer to 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 other time-dependent structures.

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 calculated on the time series are consistent over time, like 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 s
depend on the time index.

Summary statistics like the mean and variance do cha
model may try to capture.

Classical time series analysis and forecasting methods
series data stationary by identifying and removing tren

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 them from observations, then train models on the residuals.

> *If we fit a stationary model to data, we assume our data are a realization of a stationary process. So our first step in an analysis should be to 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.

Difference Transform

Differencing is a method of transforming a time series

It can be used to remove the series dependence on ti s
structures like trends and seasonality.

> *Differencing can help stabilize the mean of the*
> *time series, and so eliminating (or reducing) tr*

— Page 215, Forecasting: principles and practice.

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

```
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.

```
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 seasonality.

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 through a provided series and calculate the difference

The function below named difference() implements thi

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


We can see that the function is careful to begin the dif sure differenced values can, in fact, be calculated. 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.

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

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.

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

# invert differenced forecast
def inverse_difference(last_ob, value):
    return value + last_ob

# define a dataset with a linear trend
data = [i+1 for i in range(20)]
print(data)
# difference the dataset
diff = difference(data)
print(diff)
# invert the difference
inverted = [inverse_difference(data[i], diff[
print(inverted)
```

Running the example first prints the contrived sequen ... et is printed showing the increase by one unit each time ste ...) as the difference for the first value in the sequence canno

Finally, the difference sequence is inverted using the p ... ner for each transform.


```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1
```

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, annually, and so on. As such, identifying 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, perhaps at different scales and with the addition of trend lines.

The example below applies the difference() function to a contrived seasonal dataset. The dataset includes two cycles of 360 units each.

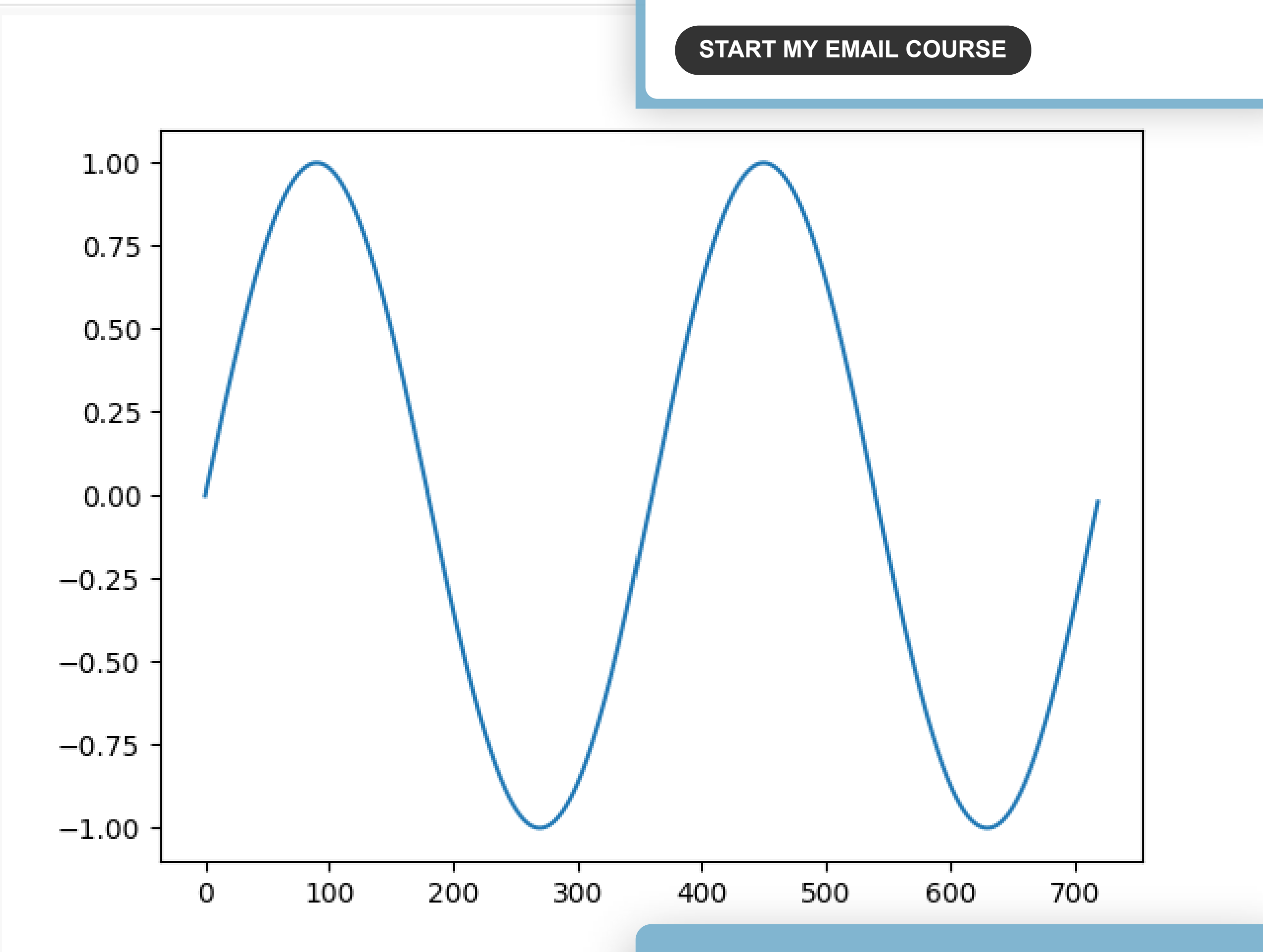
```
from math import sin
from math import radians
from matplotlib import pyplot

# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - interval]
        diff.append(value)
    return diff

# invert differenced forecast
def inverse_difference(last_ob, value):
    return value + last_ob

# define a dataset with a linear trend
data = [sin(radians(i)) for i in range(360)]
pyplot.plot(data)
pyplot.show()
# difference the dataset
diff = difference(data, 360)
pyplot.plot(diff)
pyplot.show()
# invert the difference
inverted = [inverse_difference(data[i], diff[
pyplot.plot(inverted)
pyplot.show()
```

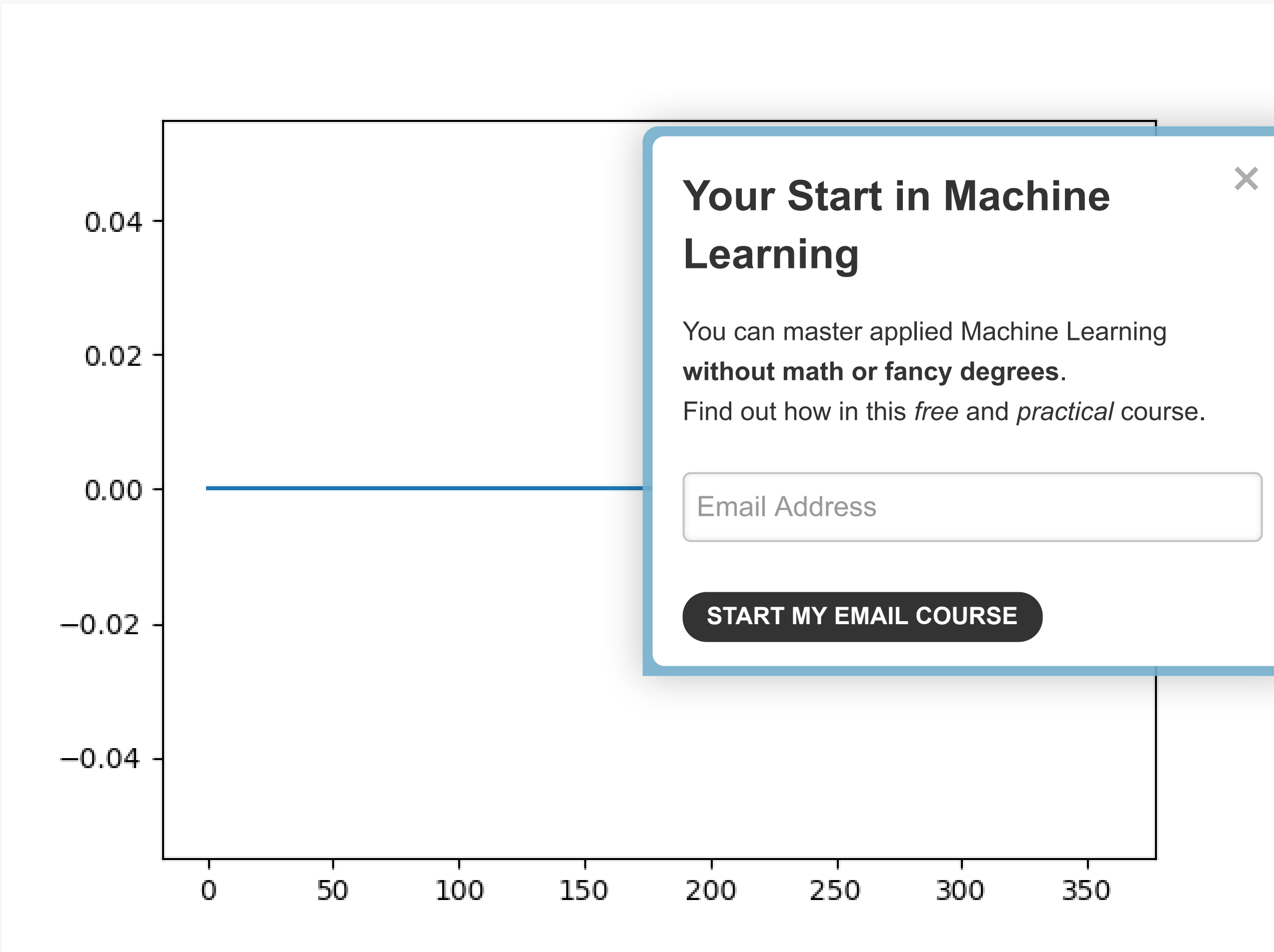

Running the example first creates and plots the datase



Line plot of a contrived sesonal dataset

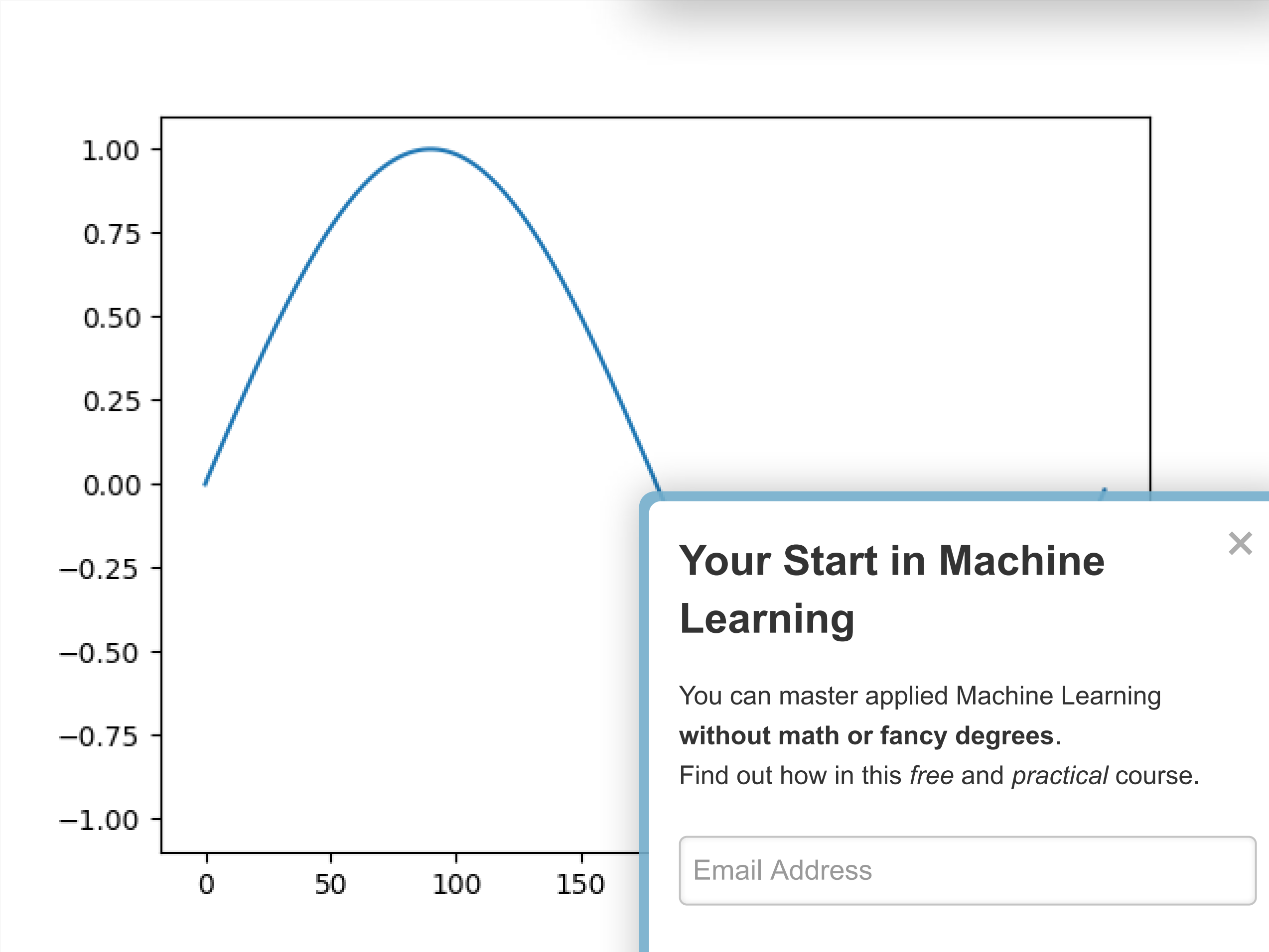
Next, the difference transform is applied and the result is plotted. The plot shows 360 zero values with all seasonality signal removed.

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.



Line plot of the differenced seasonal dataset

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



Line plot of the differenced dataset w

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.

About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.
View all posts by Jason Brownlee →

< How to Scale Data for Long Short-Term Memory Networks in Python   How to One Hot Encode Sequence Data in Python >

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

James york July 19, 2017 at 8:47 pm #   REPLY

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

I find it interesting how removing trends and seasonality resembles how some networks (such as convnet) drive towards transforming inputs into (translation) 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 # REPLY

Nice!

Nigel January 16, 2018 at 7:42 am #

Incredibly helpful – thanks for this!

Jason Brownlee January 17, 2018 at 9:54 am

Thanks, I'm glad to hear that.

GANESH July 17, 2018 at 2:35 am # REPLY

In time series if both trend and seasonality is present then first whether we remove trend and then seasonality or vice varsa to fit the model.

Jason Brownlee July 17, 2018 at 6:20 am # REPLY

Remove seasonality then trend.

Shraddha August 16, 2018 at 5:08 pm # REPLY

Why does it make a difference which one you remove first?

Jason Brownlee August 17, 2018 at 6:25 am # REPLY

Because they interact linearly.

Naira Grigoryan November 14, 2018 at 11:08 pm #

REPLY

How I can remove seasonality without knowing the lag of period?

Jason Brownlee November 15, 2018 at 5:32 am #

REPLY

Why not just find out the structure of the seasonality?

Saad December 27, 2018 at 3:07 pm #

Hi Jason!

First, my best wishes for the New Year and thank you a
I have a question concerning predicting the first order d
I have a panel data with autocorrelated dependent vari t
variables X.
After taking the first order difference, I forecast that new
independent variables, let's call these y' and X'.
y'_{i, t+1} = y_{i, t+1} – y_{i, t}
X'_{i, t} = X_{i, t} – X_{i, t-1}
and
y'_{i, t} = f(X_{i, t-1})
I run my regression model and find an R-squared of 27%.
To recover the prediction of my initial dependent variable my operation is:
\hat{y}'{i, t} = y_{i, t-12} + \sum_{s=t-11}^{t}\hat{y}'_{i, s}
(because I am not supposed to know the real ys for months t-11 through t)
My R-squared drops to -8% while the R-squared for \sum_{s=t-11}^{t}\hat{y}'_{i, s} is around 30%.
I was wondering if you can detect something wrong in that reasoning or if it is something normal.

Also, is there a way to work with non-stationary data and adjust the residual errors in the end to take into account the autocorrelation. I have some other problems in which the dependent variable can be constant for several time steps (a risk measure based on daily observations calculated over a one year time window and the time step is one month).
Thank you!

Jason Brownlee December 28, 2018 at 5:52 am #

REPLY

You can invert the differenced prediction using real or predicted obs, depending on what is available.

Abderrahim March 8, 2019 at 12:40 am #

REPLY

Hi Jason,
How to detrend using a scientifically determined trend approximation.
In my case, I have a measure called: GHI, which in simple terms: the amount of sun rays ground receives from the Sun. Used in energy fields of studies especially solar energy.

So, a real GHI measured usings sensors, and a clear_GHI, estimated following The Ineichen and Perez clear sky model.
The trend is very obvious, as you can see here: https:/ ... t least in the location I'm interested in.

First thought, I calculated
ghi / clear_ghi
But this seems very bad, since, values close to zero, a ... those values might not be so correct.

I think of
clear_ghi – ghi

Thank you so much !

Jason Brownlee March 8, 2019 at 7:52 am #

REPLY

Perhaps you mean seasonal cycle instead of trend?

You can remove the seasonal cycle by seasonal differencing.

Abderrahim March 15, 2019 at 5:23 pm #

REPLY

Many thanks Jason, your blog is amazing. I will give it a try

Jason Brownlee March 16, 2019 at 7:48 am #

REPLY

Thanks.

Hmeda March 8, 2019 at 5:36 pm #

REPLY

Hi Jason,
Thank you for your explination it was very useful.
My q is why we need to remove seasonality and trend..can you summarize the reasons in clear points ?

Jason Brownlee March 9, 2019 at 6:22 am #

REPLY

It makes the problem a lot simpler to model.

The trend and seasonality are the easy parts, so easy that we want to remove them now so we can focus on the hard parts of the problem.

Chandan Jha March 13, 2019 at 6:55 am #

Hi Jason,

Thanks for this great article. I have few doubts and I w

1) Most of the practical time series have seasonal patt
to remove both the seasonality as well as trend. Is it co

2) Models like SARIMAX(Seasonal ARIMA) have a pa
too. So does it mean that the the original time series d
difference term remove trend and seasonality paramet

3) It gets confusing here because at one point we say to remove both trend and seasonality, and then talk about SARIMAX model which can handle non-stationary data.

I hope you read my confused mind. Looking forward to hear from you.
Best Regards
Chandan

Jason Brownlee March 13, 2019 at 8:04 am #

REPLY

Yes, if the data has trend and seasonality, both should be removed before modeling with a linear algorithm.

Yes, no need to make the data stationary when using SARIMA, as you will specify how to de-trend and de-seasonalize the model as part of the config. You can make it stationary beforehand if you wish.

I hope that helps.

Chandan Jha March 14, 2019 at 1:45 am #

REPLY

Thanks for replying Sir.

So if I understand this well, we usually remove trend and seasonality by difference(with lag=1 or lag=seasonality), log transforms etc. mainly for two purposes:

– By verifying if the residuals have no pattern and is stationary
– ACF and PACF do not show high variations

And then by looking at the ACF and PACF we choose parameters which we feed to original data series when using SARIMA.

Jason Brownlee March 14, 2019 at 9:26 am #

REPLY

The SARIMA can model the trend and seasonality directly.

Chandan Jha March 15, 2019 a

I got you Sir. But SARIMA nee
posts that you use ACF and PACF cha

Jason Brownlee March 15, 20

You can also use grid search:
https://machinelearningmastery.com/how-to-grid-search-sarima-model-hyperparameters-for-time-series-forecasting-in-python/

João March 27, 2019 at 2:45 am #

REPLY

Hi Jason,

First of all thank you so much for creating these posts!

I was also reading this post of yours (https://machinelearningmastery.com/regression-tutorial-keras-deep-learning-library-python/) that uses SciKit's pipelines, and was wondering if there is an estimator for differencing the series and in this case if it would be beneficial or even a good practice to use it in the pipeline.

Kind regards,
João

Jason Brownlee March 27, 2019 at 9:05 am #

REPLY

Good point.

I've not seen one, but it would be valuable!

sanket srivastava April 18, 2019 at 3:18 pm #

REPLY

Why do we always prefer stationary data to perform time series analysis?
what is that which refrains us to analyse non- stationary data?

Jason Brownlee April 19, 2019 at 6:04 am #

A stationary time series is much simpler t

Ahmed May 4, 2019 at 10:53 am #

Jason,

Online searches show conflicting results and I would lil
necessary to remove trend and seasonality? If yes, the
ARIMA (or ARMA) can get the job done?

Jason Brownlee May 5, 2019 at 6:21 am #

REPLY

It is often a good idea and will make the dataset easier to model.

Try with/without for your data and model and compare the results.

Amit May 14, 2019 at 3:20 am #

REPLY

Hi Jason,

Can you please help me to find out -How does one invert the differencing after the residual forecast has been made to get back to a forecast including the trend and seasonality that was differenced out? After differencing the original data set 1 time and completing the prediction, how do I invert or reverse the differencing so that I can get the predicted data without differences?

Jason Brownlee May 14, 2019 at 7:51 am #

REPLY

You can invert the difference by adding the removed value and propagating this down the series.

I have an example here you can use:
https://machinelearningmastery.com/machine-learning-data-transforms-for-time-series-forecasting/

tee June 2, 2019 at 9:01 am # REPLY

Hello sir, is differencing the same as sliding window? if not, have any link to learn it ?

Jason Brownlee June 3, 2019 at 6:36 am #

No, sliding window is a way to frame a tim
trend/seasonality from time series data.

skyrim4ever July 9, 2019 at 4:00 pm #

I am not sure if my time series data is has sea
noise. So I assume the data is stationary already. Does
before training, or should I leave differencing out?

Jason Brownlee July 10, 2019 at 8:02 am # REPLY

Try modeling with an without differencing and compare model error.

Yawar Abbas September 15, 2019 at 6:17 am # REPLY

Great Article….
1. If our dataset have trend and seasonality but the model perform well on that without removing trend and seasonality. So the question is that should we remove seasonality and trend for that model or not?

2.And plz provide the link of your article of complete project of removing seasonality and trend for model. if any?

Jason Brownlee September 15, 2019 at 6:33 am # REPLY

If you get better performance by not making the series stationary, then don't make it stationary. But be sure you are making a fair comparison between the two models.

Yes, see this post:
https://machinelearningmastery.com/time-series-forecast-study-python-monthly-sales-french-champagne/

joel September 27, 2019 at 7:16 pm #

REPLY

Excellent piece.
I was wondering though can I apply the same procedure to data that has both seasonality and trend, say stock data for instance?.

Jason Brownlee September 28, 2019 at 6:14

Yes – in general. As for stocks, they are n

Imran Khan November 14, 2019 at 1:58 am #

hi
I have a time-series data of 65 years. I don't know the exact periodicity. What should be my approach to remove seasonality from the dataset

Jason Brownlee November 14, 2019 at 8:05 am #

REPLY

Plot the data a few different ways?

Test if it is stationary, and try removing cycles a few diffrent ways and see if it changes the result?

Imran Khan November 15, 2019 at 9:27 pm #

REPLY

it's a daily time scale data. It is stationary. Can you suggest some methods for removing the cycles?
Please suggest some literature also if possible.

Jason Brownlee November 16, 2019 at 7:23 am #

REPLY

Seasonal differencing.

Ric November 28, 2019 at 12:02 pm #

REPLY

Hi Jason,

Thanks for the incredibly useful articles!

I'm developing a time-series forecasting application for 15-minute data. I want to try out differencing (been applying more traditional scaling), but also want to leave out outliers (entire days that do not apply or where the data will corrupt the training), but this then introduces outliers in the differenced set!

How do you suggest I can handle outliers using differe

Many Thanks

Jason Brownlee November 28, 2019 at 1:34 p

You're welcome.

In differencing… hmmm. Probably remove them or

value.

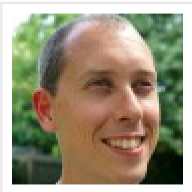
Leave a Reply

Name (required)

Email (will not be published) (required)

Website

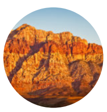
SUBMIT COMMENT



Welcome!
My name is *Jason Brownlee* PhD, and I **help developers** get results with **machine learning**.
Read more

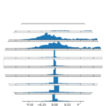
Never miss a tutorial:

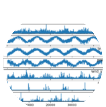
Picked for you:


How to Develop LSTM Models for Time Series Fore


How to Develop Convolutional Neural Network Mod


How to Develop Multi-Step LSTM Time Series Forecasting Models for Power Usage


How to Develop 1D Convolutional Neural Network Models for Human Activity Recognition


Multivariate Time Series Forecasting with LSTMs in Keras

Loving the Tutorials?

The Deep Learning for Time Series EBook
is where I keep the ***Really Good*** stuff.

SEE WHAT'S INSIDE