

[Open in app](#)

[Sign up](#) [Sign in](#)



Search Medium



You have **1** free member-only story left this month. [Sign up](#) for Medium and get an extra one.

♦ Member-only story

Understanding Time Series Trend

Deterministic trends vs stochastic trends, and how to deal with them



Vitor Cerqueira · [Follow](#)

Published in [Towards Data Science](#)

6 min read · Mar 14

Listen

Share



Photo by [Ali Abdul Rahman](#) on [Unsplash](#)

Detecting and dealing with the trend is a key step in the modeling of time series.

In this article, we'll:

- Describe what is the trend of a time series, and its different characteristics;
- Explore how to detect it;
- Discuss ways of dealing with trend;

Understanding Trend

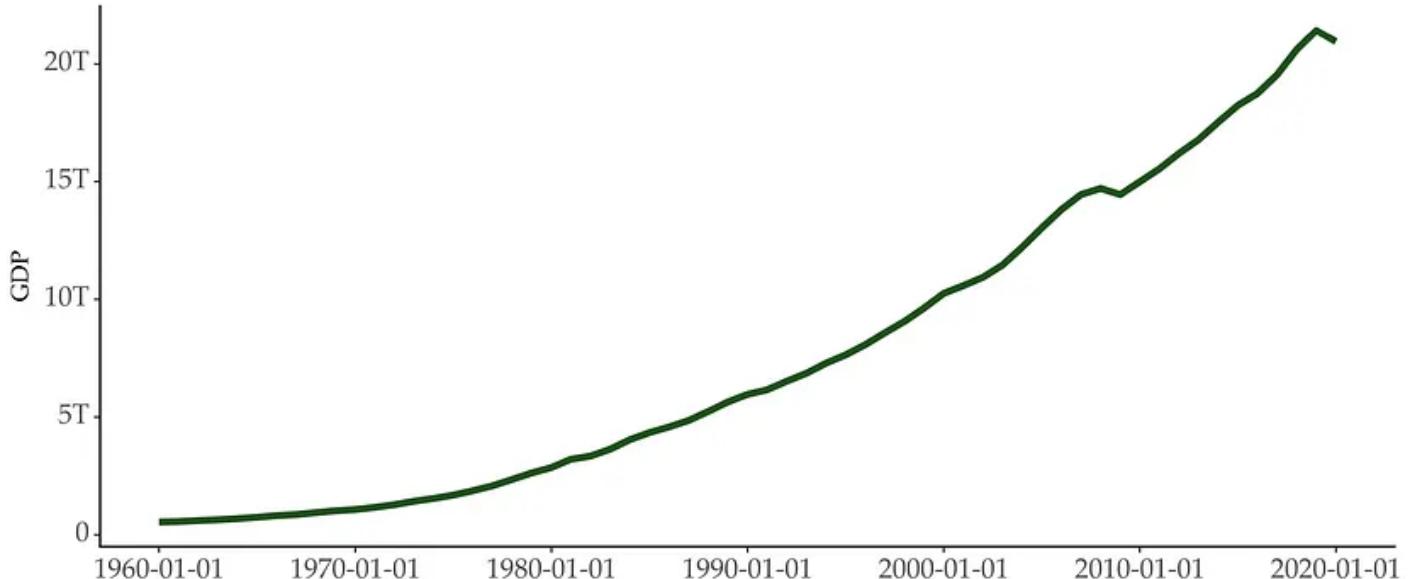
Trend as a building block of time series

At any given time, a time series can be decomposed into three parts: trend, seasonality, and the remainder.

$$\text{Time Series} = \text{Trend} + \text{Seasonality} + \text{Remainder}$$

Additive decomposition of a time series

The trend represents the long-term change in the level of a time series. This change can be either upward (increase in level) or downward (decrease in level). If the change is systematic in one direction, then the trend is monotonic.



USA GDP time series with an upward and monotonic trend. Data source in reference [1]. Image by author.

Trend as a cause of non-stationarity

A time series is stationary if its statistical properties do not change. This includes the level of the time series, which is constant under stationary conditions.

So, when a time series exhibits a trend, the stationarity assumption is not met. Modeling non-stationary time series is challenging. If untreated, statistical tests and forecasts can be misleading. This is why it's important to detect and deal with the trend before modeling time series.

A proper characterization of the trend affects modeling decisions. This, further down the line, impacts forecasting performance.

Deterministic Trends

A trend can be either deterministic or stochastic.

Deterministic trends can be modeled with a well-defined mathematical function. This means that the long-term behavior of the time series is predictable. Any deviation from the trend line is only temporary.

In most cases, deterministic trends are linear and can be written as follows:

$$\text{trend} = a + b \cdot \text{time}$$

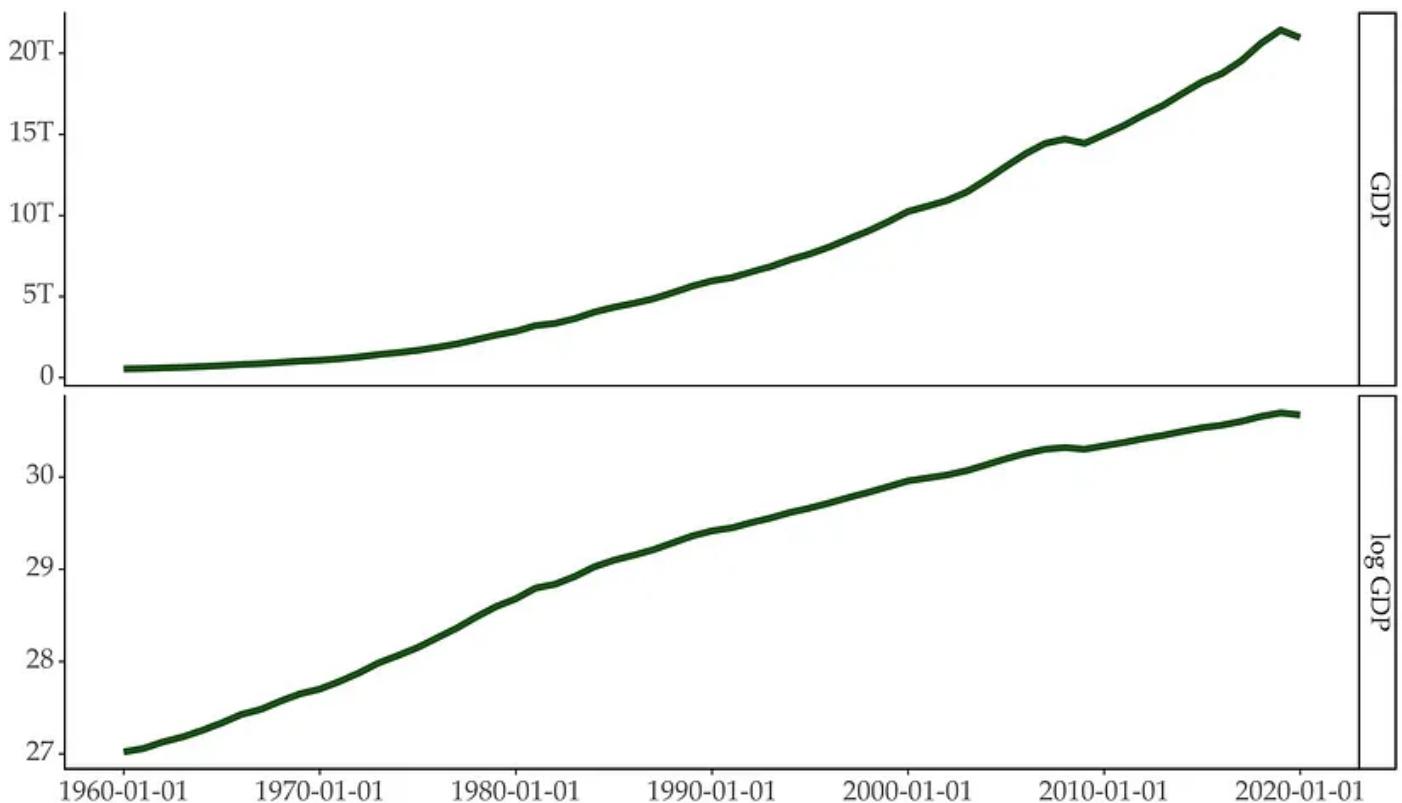
The equation for a linear trend. The coefficient b is the expected change in the trend in consecutive periods. The coefficient a is the intercept.

But, trends can also follow an exponential or polynomial form.

$$\text{trend} = e^{a+b \cdot \text{time}}$$

Exponential trend equation. This trend can be made linear by taking the log on both sides.

In the economy, there are several examples of time series that increase exponentially, such as GDP:



USA GDP time series. The original trend is exponential, but it becomes linear after the logarithm transformation. Data source in reference [1]. Image by author.

A time series with a deterministic trend is called trend-stationary. This means the series becomes stationary after removing the trend component.

Linear trends can also be modeled by including time as an explanatory variable. Here's an example of how you could do this:

```
import numpy as np
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA

# https://github.com/vcerqueira/blog/blob/main/data/gdp-countries.csv
series = pd.read_csv('data/gdp-countries.csv')[['United States']]
series.index = pd.date_range(start='12/31/1959', periods=len(series), freq='Y')

log_gdp = np.log(series)

linear_trend = np.arange(1, len(log_gdp) + 1)

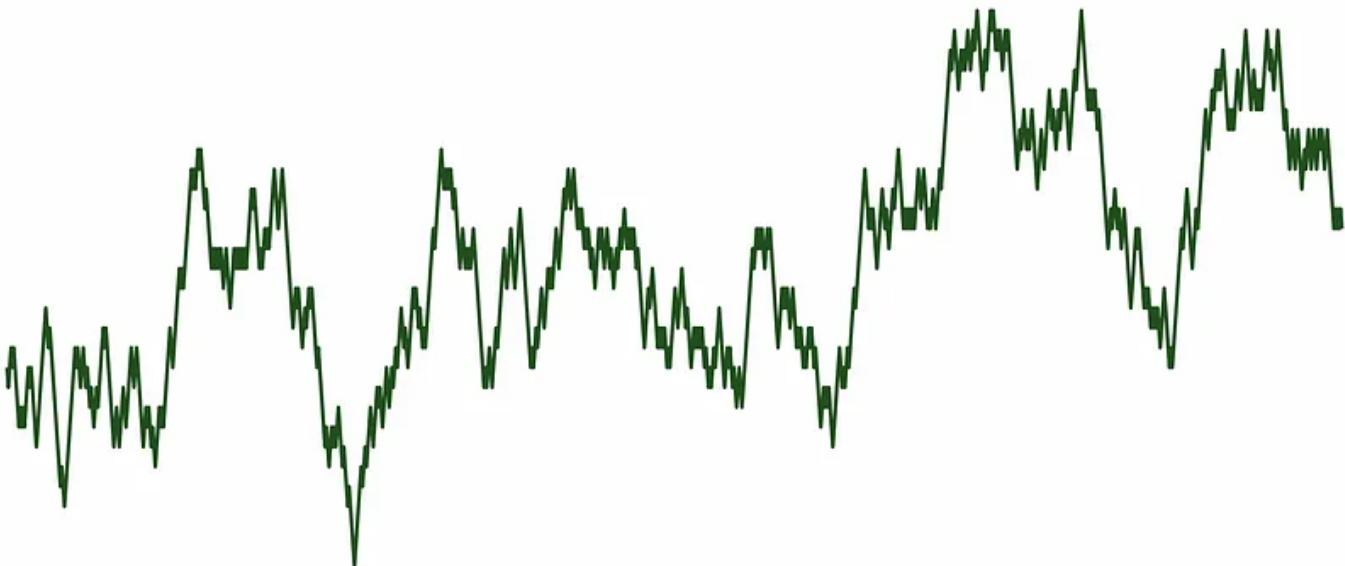
model = ARIMA(endog=log_gdp, order=(1, 0, 0), exog=linear_trend)
result = model.fit()
```

Stochastic Trends

A stochastic trend can change randomly, which makes their behavior difficult to predict.

A random walk is an example of a time series with a stochastic trend:

```
rw = np.cumsum(np.random.choice([-1, 1], size=1000))
```



A random walk time series whose trend changes suddenly and unpredictably. Image by author.

Stochastic trends are related to unit roots, integration, and differencing.

Time series with stochastic trends are referred to as difference-stationary. This means that the time series can be made stationary by differencing operations. Differencing means taking the difference between consecutive values.

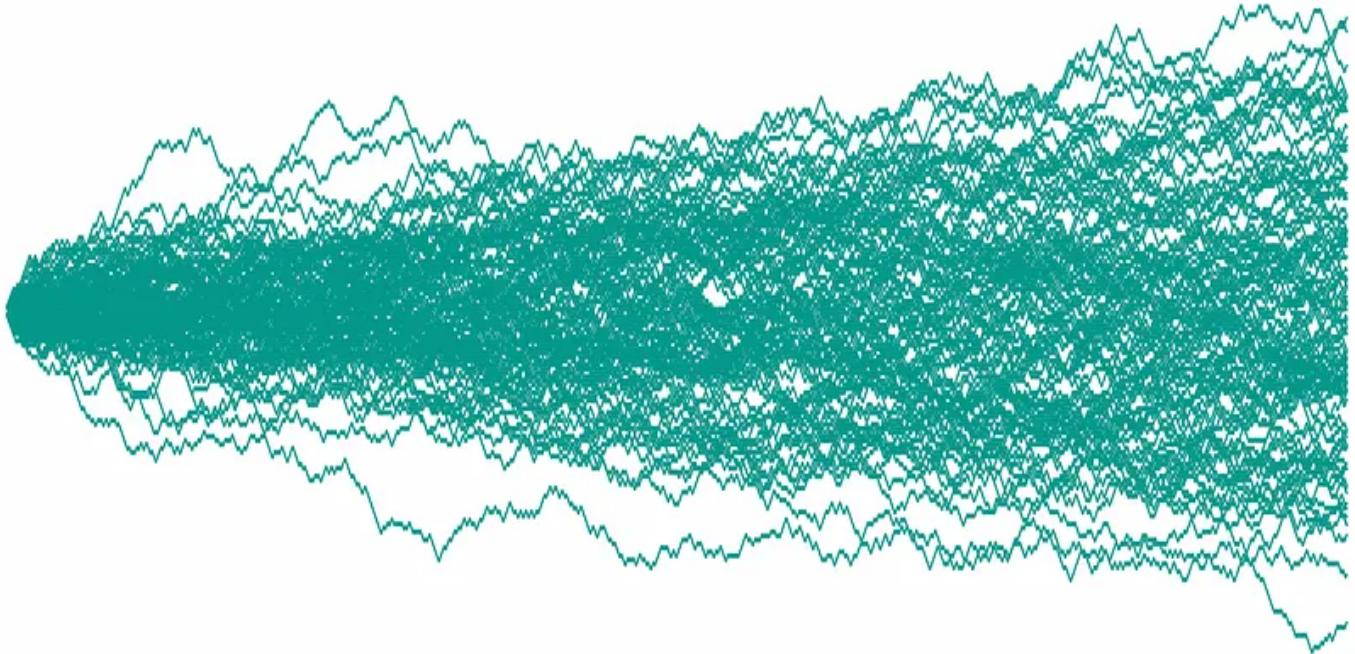
Difference-stationary time series are also called integrated. For example, ARIMA (Auto-Regressive Integrated Moving Average) models contain a specific term (I) for integrated time series. This term involves applying differencing steps until the series becomes stationary.

Finally, difference-stationary or integrated time series are characterized by unit roots. Without going into mathematical details, a unit root is a characteristic of non-stationary time series.

Forecasting Implications

Deterministic and stochastic trends have different implications for forecasting.

Deterministic trends have a constant variance throughout time. In the case of a linear trend, this implies that the slope will not change. But, real-world time series show complex dynamics with the trend changing over long periods. So, long-term forecasting with deterministic trend models can lead to poor performance. The assumption of constant variance leads to narrow forecasting intervals that underestimate uncertainty.



Many realizations of a random walk. Image by author.

Stochastic trends are assumed to change over time. As a result, the variance of a time series increases across time. This makes stochastic trends better for long-term forecasting because they provide more reasonable uncertainty estimations.

How to Detect Trend

Stochastic trends can be detected using unit root tests. For example, the augmented Dickey-Fuller test, or the KPSS test.

Augmented Dickey-Fuller (ADF) test

The ADF test checks whether an auto-regressive model contains a unit root. The hypotheses of the test are:

- Null hypothesis: There is a unit root (the time series is not stationary);
- Alternative hypothesis: There's no unit root.

This test is available in *statsmodels*:

```
from statsmodels.tsa.stattools import adfuller  
  
pvalue_adf = adfuller(x=log_gdp, regression='ct')[1]  
  
print(pvalue_adf)  
# 1.0
```

The parameter *regression='ct'* is used to include a constant term and the deterministic trend in the model. As you can check [in the documentation](#), there are four possible alternative values to this parameter:

- *c*: including a constant term (default value);
- *ct*: a constant term plus linear trend;
- *ctt*: constant term plus a linear and quadratic trend;
- *n*: no constant or trend.

Choosing which terms should be included is important. A wrong inclusion or exclusion of a term can substantially reduce the power of the test. In our case, we used the *ct* option because the log GPD series shows a linear deterministic trend behavior.

KPSS test

The KPSS test can also be used to detect stochastic trends. The test hypotheses are opposite relative to ADF:

Null hypothesis: the time series is trend-stationary;

Alternative hypothesis: There is a unit root.

```
from statsmodels.tsa.stattools import kpss  
  
pvalue_kpss = kpss(x=log_gdp, regression='ct')[1]
```

```
print(pvalue_kpss)
# 0.01
```

The KPSS rejects the null hypothesis, while ADF doesn't. So, both tests signal the presence of a unit root. Note that a time series can have a trend with both deterministic and stochastic components.

So, how can you deal with unit roots?

How to Deal with Trend

We've explored how to use time as an explanatory variable to account for a linear trend.

Another way to deal with trends is by differencing. Instead of working with the absolute values, you model how the time series changes in consecutive periods.

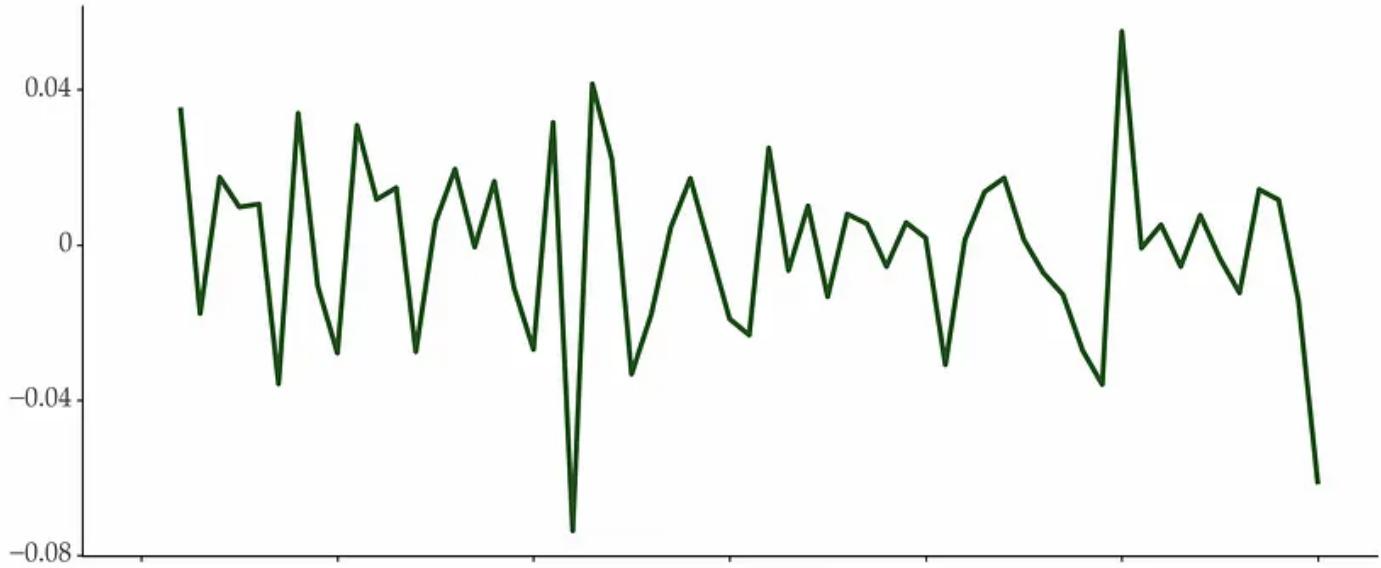
A single differencing operation is usually enough to achieve stationarity. Yet, sometimes you need to do this process many times. You can use ADF or KPSS to estimate the required number of differencing steps. The *pmdarima* library wraps this process in the function *ndiffs*:

```
from pmdarima.arima import ndiffs

# how many differencing steps are needed for stationarity?
ndiffs(log_gdp, test='adf')
# 2
```

In this case, the log GPD series needs 2 differencing steps for stationarity:

```
diff_log_gdp = log_gdp.diff().diff()
```



Second differences of the log GDP time series. Image by author.

Key Takeaways

- The trend is a key time series component;
- Time series that exhibit a trend are non-stationary because the mean level is not constant;
- The trend of a time series can be deterministic or stochastic. Deterministic trends are modeled with well-defined mathematical functions, while stochastic trends can change randomly;
- Stochastic trends can be detected using unit root tests such as ADF or KPSS;
- Differencing operations are used to remove stochastic trends from a time series.

Thank you for reading, and see you in the next story!

References

[1] [GDP Timeseries Data for various Countries](#) (License: CC0: Public Domain)

[2] [Stochastic and deterministic trends, in Forecasting: Principles and Practice](#)

[3] [A Random Walk — introduction and properties](#) by Ben Lambert



Follow

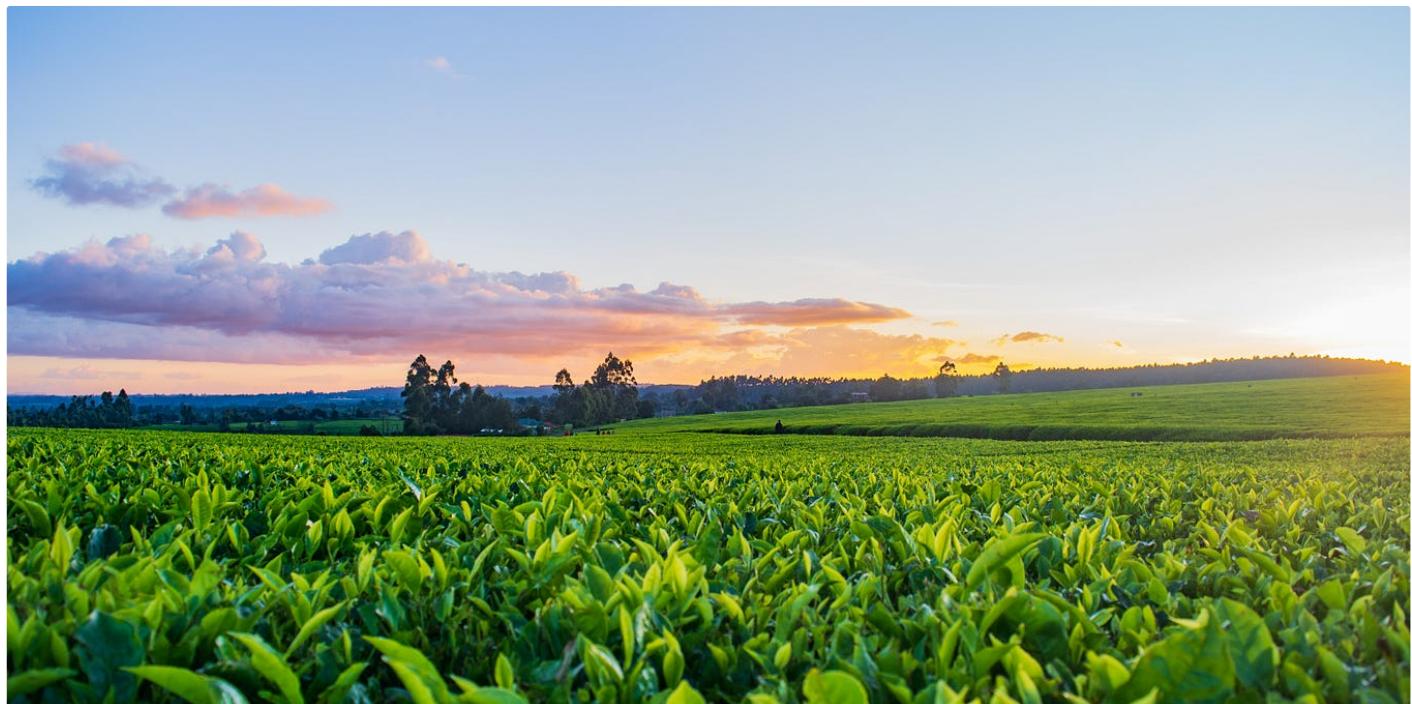


Written by Vitor Cerqueira

1.3K Followers · Writer for Towards Data Science

Ph.D and Researcher on Time Series Forecasting. Top writer in Time series

More from Vitor Cerqueira and Towards Data Science



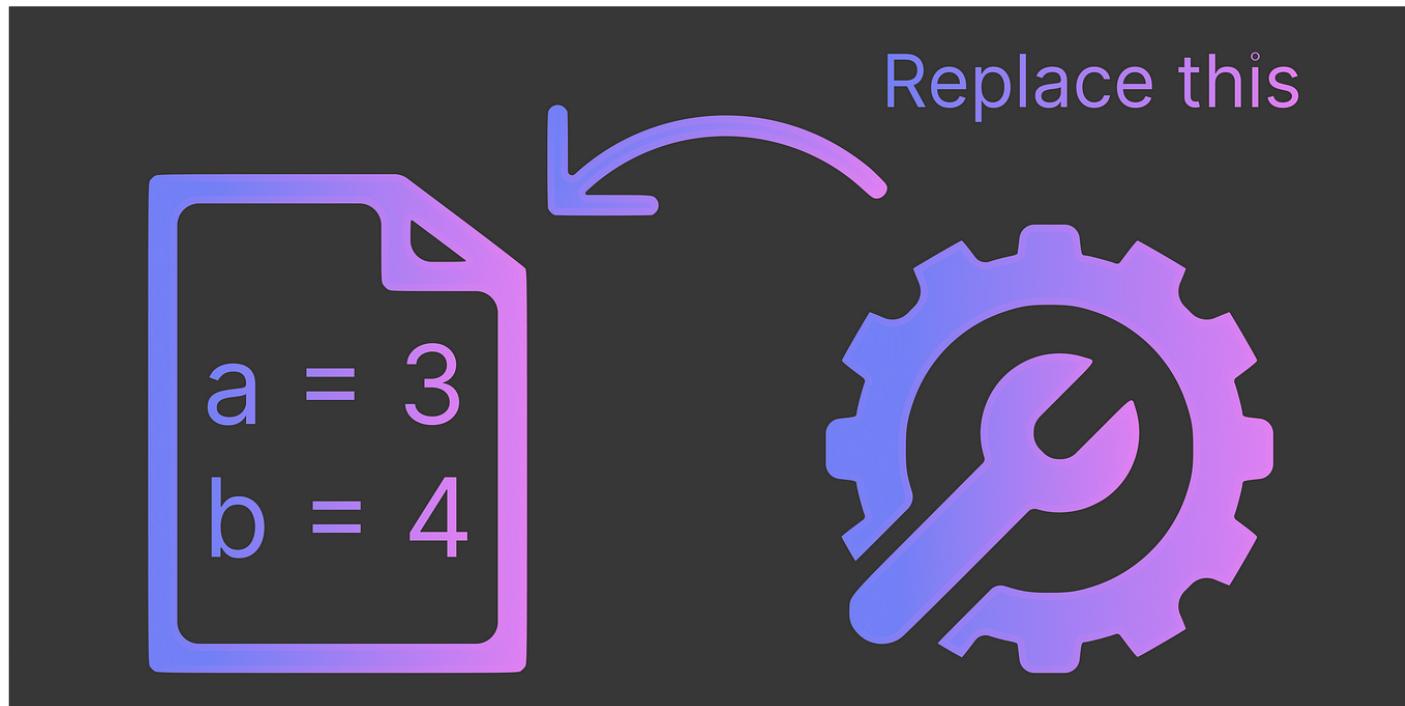
 Vitor Cerqueira in Towards Data Science

Time Series for Climate Change: Using Deep Learning for Precision Agriculture

How to use time series analysis and forecasting to tackle climate change

 · 6 min read · May 29





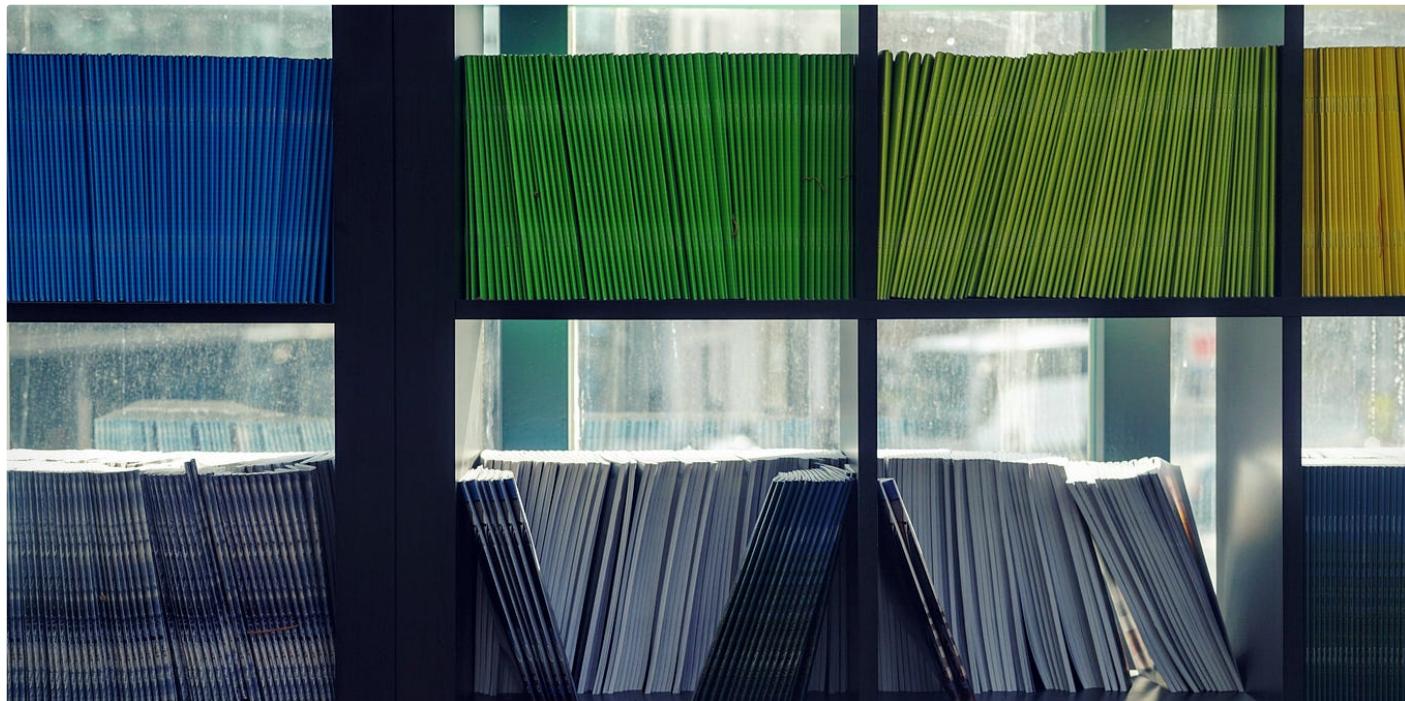
 Khuyen Tran in Towards Data Science

Stop Hard Coding in a Data Science Project—Use Config Files Instead

And How to Efficiently Interact with Config Files in Python

◆ · 6 min read · May 25

 1.8K  20



 Jacob Marks, Ph.D. in Towards Data Science

How I Turned My Company's Docs into a Searchable Database with OpenAI

And how you can do the same with your docs

15 min read · Apr 25



4.1K



48



Vitor Cerqueira in Towards Data Science

Time Series for Climate Change: Reducing Food Waste with Clustering

Using time series clustering for better demand forecasts

◆ · 6 min read · Jun 7



129



3



See all from Vitor Cerqueira

See all from Towards Data Science

Recommended from Medium

 Naina Chaturvedi in Coders Mojo

Implemented Time Series Analysis and Forecasting Projects

Repo for all the projects (vertical post)...

◆ · 85 min read · Jan 6

 139  2

↗⁺

 Marco Peixeiro  in Towards Data Science

PatchTST: A Breakthrough in Time Series Forecasting

From theory to practice, understand the PatchTST algorithm and apply it in Python alongside N-BEATS and N-HiTS

◆ · 10 min read · Jun 20

👏 368

8



Lists



Predictive Modeling w/ Python

18 stories · 39 saves

Practical Guides to Machine Learning

10 stories · 56 saves

Natural Language Processing

363 stories · 17 saves



ChatGPT prompts

17 stories · 27 saves



 Coucou Camille in CodeX

Time Series Prediction Using LSTM in Python

Implementation of Machine Learning Algorithm for Time Series Data Prediction.

 · 6 min read · Feb 10

 117

 1



 Arthur Mello in Geek Culture

Bayesian Time Series Forecasting

Theory and practice using PyBATS

◆ 6 min read · Jan 19

👏 136

💡 1

↗⁺



Marco Peixeiro In Towards Data Science

Theta Model for Time Series Forecasting

A hands-on tutorial on how to apply the Theta model for time series forecasting in Python

◆ 9 min read · Nov 2, 2022

👏 51

💡 1

↗⁺

 Ambuj Shukla

Building Interactive Web Applications with R and Shiny

A Comprehensive Guide to Creating Dynamic Data-Driven Apps

◆ · 6 min read · Feb 9

 77  ‡

[See more recommendations](#)