# Guide to Time Series Analysis with Python — 1: Analysis Techniques and Baseline Model

[[Buse Köseoğlu](https://buse-koseoglu13.medium.com/?source=post_page-----59e12fdd2eba--------------------------------)](https://buse-koseoglu13.medium.com/?source=post_page-----59e12fdd2eba--------------------------------)

[Buse Köseoğlu](https://buse-koseoglu13.medium.com/?source=post_page-----59e12fdd2eba--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2F5e1a64153870&operation=register&redirect=https%3A%2F%2Fbuse-koseoglu13.medium.com%2Fguide-to-time-series-analysis-with-python-1-analysis-techniques-and-baseline-model-59e12fdd2eba&user=Buse+K%C3%B6seo%C4%9Flu&userId=5e1a64153870&source=post_page-5e1a64153870----59e12fdd2eba---------------------post_header-----------)

6 min read

·

Jul 1

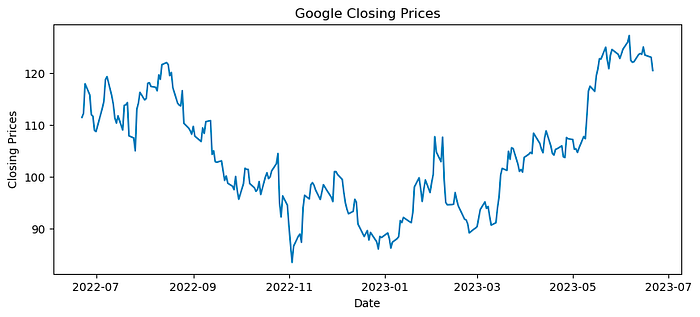
258

Time series analysis is a method used in various industries, such as stock market prediction, product price forecasting, weather forecasting, cryptocurrency prediction, etc., to forecast future values based on historical data.

Time series can be recorded in minute, hourly, daily, monthly, etc. intervals.

In this article, we will examine the characteristics of time series using the last 1-year Google stock closing data. You can find the codes on the [github repo](https://github.com/busekoseoglu/time-series-forecasting).

The graph below represents the changes in stock prices over time. By examining the graph, we can observe that there has been an upward trend in recent times.



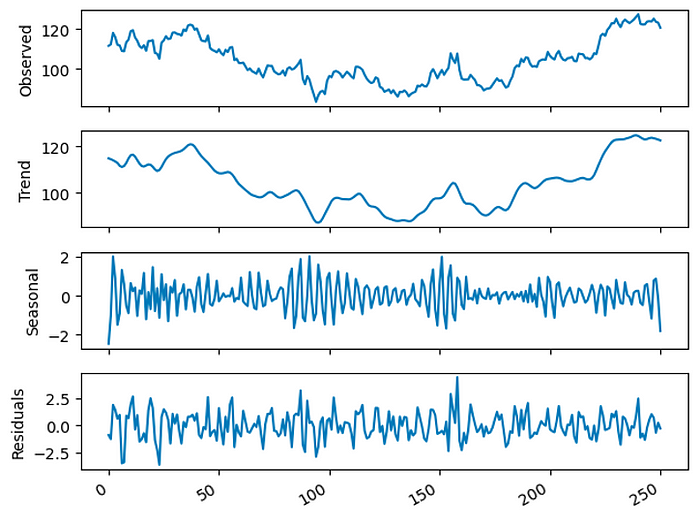
Google closing prices change over time

# Time Series Components

All time series can be divided into three components:

1. **Trend:** Slow-moving changes that occur in time series.
2. **Seasonality:** Patterns of variation that repeat at specific time intervals. These can be weekly, monthly, yearly, etc. Seasonal changes indicate deviations from the trend in specific directions.
3. **Residuals:** Unusual events that occur in the data, such as a sudden increase in heart rate for a person during exercise. These cause random errors and are also referred to as “white noise.”

The visualization of the above components is called “decomposition.”



Decompositon of data

When examining the graphs, an increasing trend can be observed in recent times. Additionally, there is seasonality in the data. If there were no seasonality in the data, the “Seasonal” graph would continue as a straight line starting from 0.

# Naive Prediction

Before trying complex models with time series data, it is important to establish a simple baseline model for comparison. This baseline model should not involve complex calculations and should not require much time. As a baseline model, we can consider:

* **Arithmetic mean:**For example, if we have daily stock data and we want to predict the next 7 days, we can use the arithmetic mean by taking the average of the previous 7 days as the prediction for the next 7 days.
* **Last observed value:** Again, considering stock data, if the date we want to predict is 20.01.2023, we would use the value observed on 19.01.2023 as the prediction for 20.01.2023.

When creating these models (predictions), we can also note their scores such as MSE (Mean Squared Error), MAPE (Mean Absolute Percentage Error), etc., in order to measure how much better our future complex models perform compared to these baseline models.

# Random Walk

Random Walk represents a journey that occurs randomly, as the name suggests. In this model, the value of the next step is determined randomly based on the value of the current step. The values obtained at each step are typically determined by random variables that are independent and have the same distribution.

Using past data to predict future values in this process is not meaningful due to the random nature of the process.

To determine whether the process is a random walk, specific analyses should be conducted. These analyses and their order are shown in the figure below. Let’s now examine the meaning of these steps.

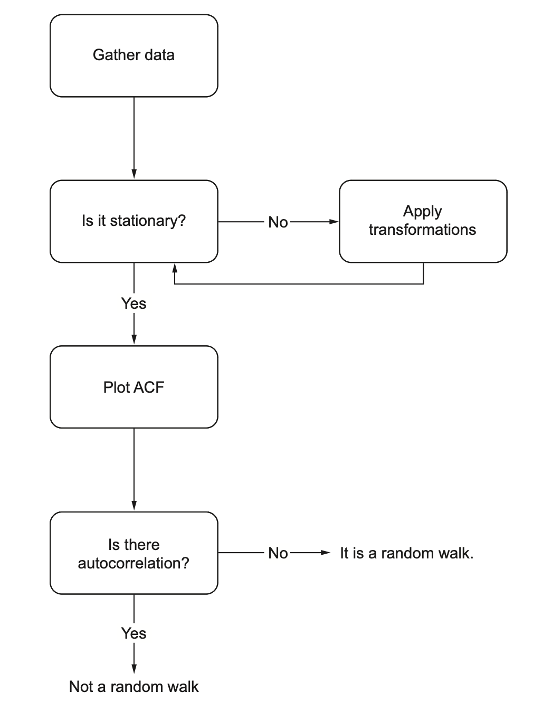
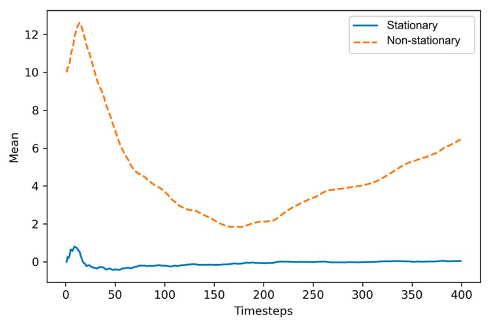
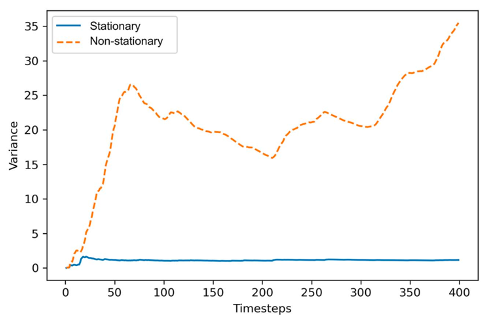


Figure from Time series forecasting in python — Marco Peixerio

**Stationary Data**

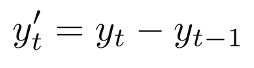
Models used for time series forecasting, such as MA, AR, ARMA, are designed to work with stationary data.

A stationary process is a process whose statistical properties do not change over time. If the mean, variance, and autocorrelation of a time series do not change over time, then the data is considered to be stationary.



When working with real-life data, the data we often obtain does not adhere to this definition of stationarity. However, there are methods available to make the data stationary. These include:

* **Differencing:**This transformation stabilizes the mean, removes trends and seasonal effects. This process is achieved by subtracting the value of the previous time step, y(t-1), from the current value, y(t). This process can be performed multiple times.



* **Taking the logarithm of the series:**Taking the logarithm helps stabilize the variance of the series.

**NOTE:**If we apply any transformation to time series, we need to reverse this process when obtaining model prediction results.

# Understanding if the Data is Stationary

To determine whether a time series data is stationary, the Dickey-Fuller (ADF) test is used. Similar to a t-test, a significance threshold is set, and the p-value is interpreted based on this threshold.

* **Null Hypothesis:** Data is non-stationary
* **Alternative Hypothesis:**Data is stationary

If:

* p-value < threshold value (e.g., 0.01, 0.05, etc.), we can reject the null hypothesis and conclude that the data is stationary.

When we apply the Dickey-Fuller test to Google stock closing data, we obtain the following results:

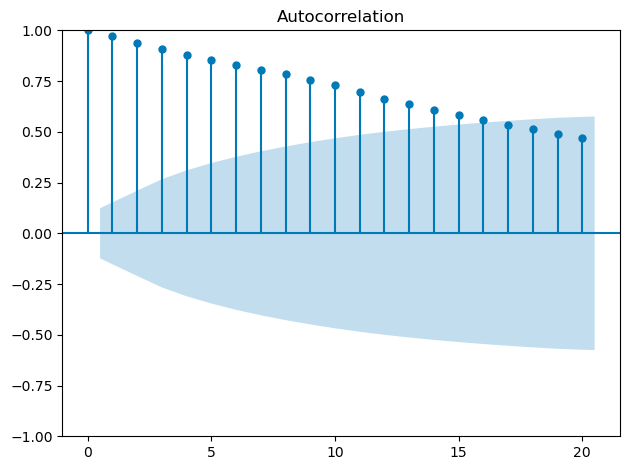
* **ADF Statistic:** -1.4904778097402294
* **p-value:**0.538216461195455

If we consider a threshold of 0.05 for the p-value, we can say that this data is non-stationary.

# ****Autocorrelation Function****

Once we have a stationary series, the next step is to determine if there is any autocorrelation. The autocorrelation function (ACF) measures the linear relationship between lagged values of a time series, specifically looking at the relationship between yt and yt-1.

In the presence of a trend, the ACF plot shows that the coefficients are high for the initial lags and decrease linearly as the lag increases. Additionally, if there is seasonality in the data, the ACF plot will also indicate it.



ACF plot for Google Closing Prices

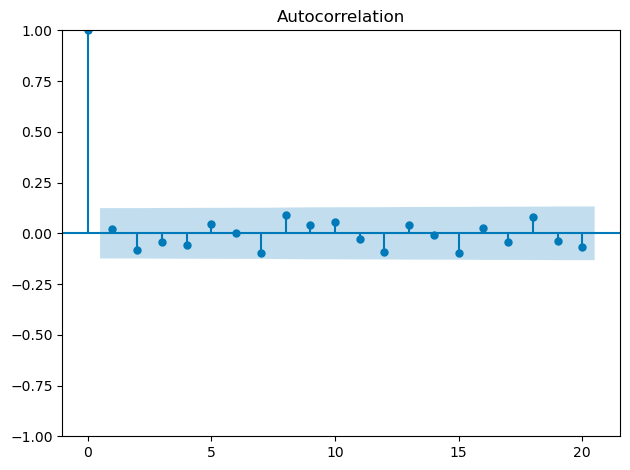
If we examine the above ACF plot, we can say that the coefficients are significant because they fall outside the shaded area, which represents the confidence interval.

# When Data Transformation is Applied: Dickey-Fuller Test Results and ACF Plot

When data transformation is applied, it can have an impact on the results of the Dickey-Fuller test and the ACF plot. By transforming the data, we aim to make it more stationary and remove any trends or seasonality.

When we obtain the above Dickey-Fuller test results and ACF plot (which is often encountered in real-life data), we need to apply transformations to make the data stationary. When we apply differencing to the data, the results change as follows:

* **ADF Statistic:** -15.35612522283515
* **p-value:** 3.664746148276414e-28



ACF plot for Google Closing Prices after Transformation

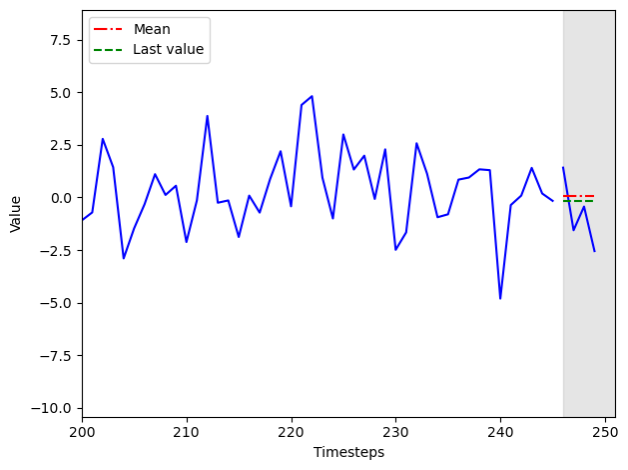
Since the p-value is less than 0.05, we can now conclude that we have obtained a stationary data. When we examine the ACF plot, we observe that the coefficients fall within the confidence interval, indicating insignificance.

After applying differencing to the data, it is important that the data we feed into the model is the data obtained after differencing.

# ****Baseline Model****

Before creating the baseline model, we converted the output of differencing into a dataframe and then split the data into train and test sets.

The baseline model will be created based on the arithmetic mean and the last observed value. If we examine the prediction graph below, we can observe that both models perform poorly. However, this is not significant. What matters is that the models we build in the future outperform these two models. We can use the MSE metric for this comparison. The MSE for the arithmetic mean model is 2.86, and for the last observed value model, it is 2.56.



Baseline model predictions

You’re welcome! Thank you for reading. In the next article, we will make predictions using Moving Average Process and Autoregressive Process. Stay tuned!

# References

* Time series forecasting in python — Marco Peixerio
* Tan,Y.-F.;Ong,L.-Y.;Leow, M.-C.; Goh, Y.-X. Exploring Time-Series Forecasting Models for Dynamic Pricing in Digital Signage Advertising. Future Internet 2021, 13, 241. <https://doi.org/10.3390/> fi13100241
* Machine learning with time series — Markus Löning
* <https://www.influxdata.com/time-series-forecasting-methods/>
* <https://builtin.com/data-science/time-series-forecasting-python>
* <https://www.machinelearningplus.com/time-series/time-series-analysis-python/>