**TIME SERIES ANALYSIS**

**What is Time Series Analysis?**

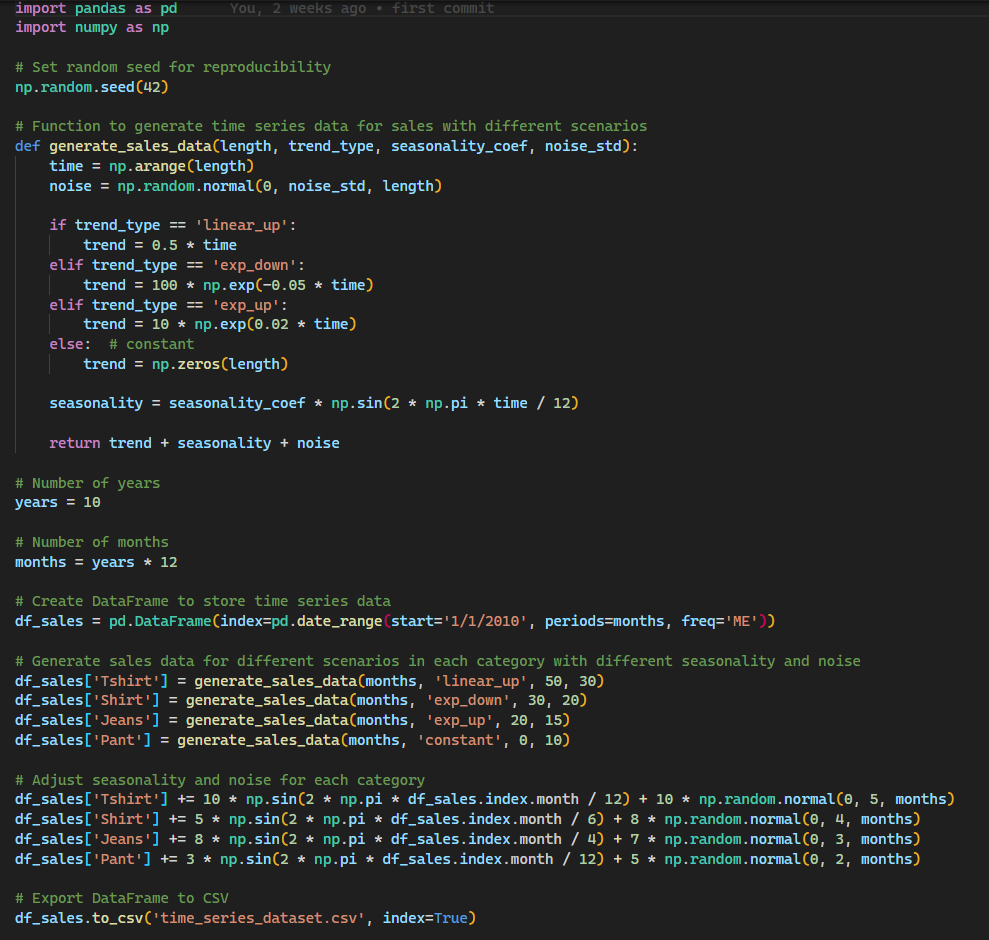
Time series analysis is a statistical method used to study and interpret patterns in data that unfold over time. If you’ve ever glanced at stock market charts, those intricate graphs depict a prime example of time series data in action.

Imagine you’re looking at a series of data points collected over time, like stock prices going up and down. Machine learning looks at this data and tries to find patterns or trends, sort of like figuring out if the stock prices tend to rise or fall during certain times of the year.Now, forecasting is like using a crystal ball, but a very smart one! It uses all the historical data it has seen to make an educated guess about what might happen in the future. It’s not saying, “This will definitely happen,” but more like, “Based on what I’ve seen before, here’s a good guess about what might happen next.” So, it’s like having a calculated estimate based on what happened in the past to give you an idea of what might happen in the future. It’s not perfect, but it’s a helpful way to make predictions.

**Key components for Time series analysis  
Time Series Data:**  
Time series data consists of observations or measurements collected over time, with each data point corresponding to a specific moment in time. Examples include stock prices, temperature readings, sales figures, and more.  
 **Components of a Time Series:**  
**a. Trend:** The long-term movement or direction in the data. It represents the underlying pattern that may be increasing, decreasing, or staying relatively constant over time.  
**b. Seasonality:**repeating patterns or cycles that occur at regular intervals, often associated with specific seasons, months, days, or times of the day.  
**c. Cyclic Patterns:** longer-term fluctuations that do not have fixed periods and may not occur at regular intervals.  
**d. Residual Component:** The data’s irregular or random fluctuations that go against trends, seasonal patterns, or cyclical patterns.  
 **Methods for Time Series Analysis:  
a. Descriptive Statistics:** This involves looking at basic numbers to understand the main and common values in our data. We calculate things like the average (mean), middle point (median), and how spread out the values are (standard deviation).  
**b. Visualization:** We use graphs and charts to see patterns and trends. For example, line charts and time series plots help us visualize how things change over time.  
**c. Smoothing Methods:** Sometimes, data can be a bit noisy. Smoothing methods help us see the main trends by reducing the impact of random fluctuations. It’s like ironing out the wrinkles in the data. We might use methods like moving averages, which show an average value over a period.  
**d. Statistical Models:** Here, we use more advanced math to understand and predict trends. Think of it like fitting a puzzle. We might use techniques like ARIMA, which looks at past values to predict future ones, or more complex models like LSTM networks, which are like super-smart tools for understanding patterns in data over time.

**Forecasting:**Time series analysis frequently includes predicting future values by examining historical patterns. Forecasting techniques span from basic extrapolation to advanced predictive modelling

**Evaluation:**The performance of time series models is assessed using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or root mean squared error (RMSE).

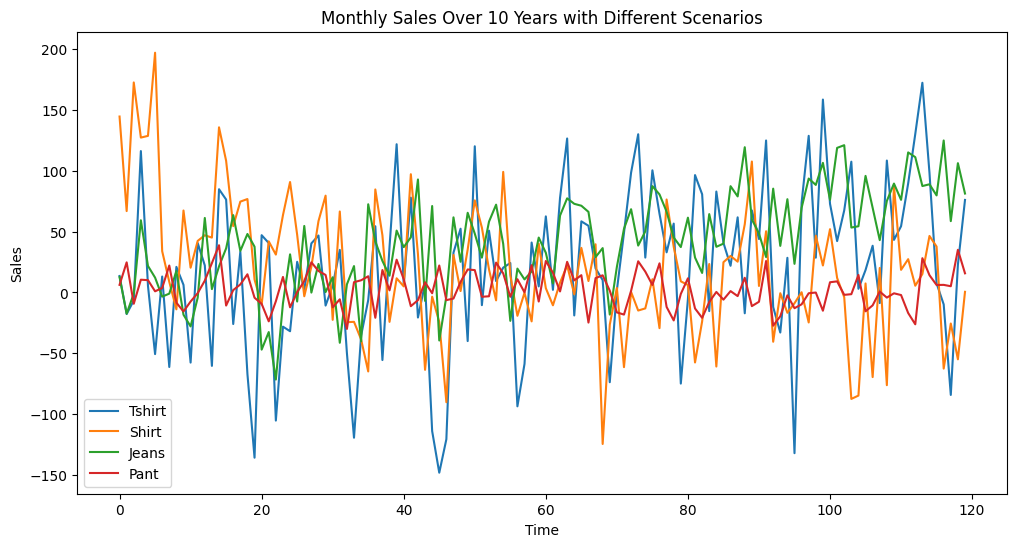
We’ll create the dataset using a Python code.

Our primary emphasis right now is not on comprehending this Python code; rather, we are directing our attention toward gaining an understanding of the different components present within the dataset.

Plotting the dataset

A screen shot of a computer code

Description automatically generated



The dataset generated using the code

This dataset appears to contain monthly changes in sales or quantities for four different types of clothing items: T-shirts, shirts, jeans, and pant. The data spans from January 2010 to December 2019. Each row in the dataset represents a specific month, and the corresponding values indicate the sales or quantities for each clothing category. As this dataset is fictional, the negative values are not our concerns

Exploring Different Components using visual representation

To visually represent trend, seasonality, cyclicity, and noise without using any library or tool, you can create simple plots using basic coding or drawing tools. Here's a conceptual guide on how you might represent each component:

1. Trend: Draw a line that captures the overall upward or downward movement of the time series over time.  
   For example, if you have a time series that generally increases, draw a diagonal line from the bottom-left to the top-right of your plot.

A graph with blue lines

Description automatically generated

1. Seasonality: Identify repeating patterns over fixed intervals (e.g., daily, weekly, monthly).  
   Draw repeating shapes or curves that represent the seasonal variations. For instance, if you have weekly seasonality, draw a repeating pattern for each week.

A graph with blue lines

Description automatically generated

1. Cyclicity: Cyclicity refers to longer-term patterns that may not have fixed intervals.  
   Represent cyclical patterns with curves or shapes that capture the longer-term ups and downs.
2. Noise: Noise is typically random fluctuations in the data.  
   Represent noise by adding small, irregular spikes or deviations around the trend, seasonality, and cyclical components.

A graph of a blue line

Description automatically generated

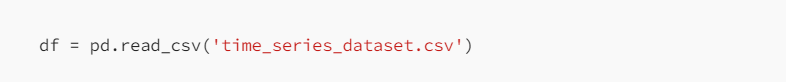
**Libraries we are going to use:**1. pandas2. matplotlib3. NumPy4. statsmodels5. scikit-learn

Import the libraries

A close-up of a computer code

Description automatically generated

Load the Dataset

  
  
The library we will use is [statsmodels.tsa.seasonal.seasonal\_decompose](https://www.statsmodels.org/dev/generated/statsmodels.tsa.seasonal.seasonal_decompose.html" \t "_blank) and additive model  
There are 2 types of model in Time Series Analysis  
1. Additive: i.e. Original Data = Trend + Seasonality + Residual  
2. Multiplicative: i.e. Original Data = Trend \* Seasonality \* Residual

**Library Application**

A close-up of a computer code

Description automatically generated

**Original Data**

A screen shot of a computer code

Description automatically generated

A blue line graph with black text

Description automatically generated

**Trend**

A white background with black text

Description automatically generated

A graph with a line

Description automatically generated

**Seasonality**

A white background with black text

Description automatically generated

A graph showing a blue line

Description automatically generated**Residual**

A white background with black text

Description automatically generated

A graph showing a number of components

Description automatically generated with medium confidence

**Mathematical or Manual Approach**

**Trend**

In order to manually calculate the trend, a technique for smoothing out short-term variations in a time series dataset must be applied. The computation of a moving average is one such technique. Here’s a step-by-step tutorial on using a basic moving average to manually calculate the trend:

**Step 1:** Define a Window Size: Select a window size that corresponds to the quantity of data points you wish to use in the moving average computation. The amount of smoothing depends on the size of the window.  
**Step 2:** Identify Data Points: Determine the range of data points, including the data point itself, that fall inside the selected window size for each data point in your time series.  
**Step 3:** Determine the Mean Determine the mean of the selected data points. For that particular data point, the smoothed value is represented by this average.  
**Step 4:** Continue with Every Data Point

****

**A graph with blue and orange lines

Description automatically generated**

**Detrend**  
Now we remove the trend from out data to make it detrended so that we can calculate seasonality

****

A graph of blue and orange lines

Description automatically generated

**Seasonality**

Step 1: Convert Index to Datetime

* Begins by converting the index of the DataFrame (seasonal\_df) to a datetime format. This ensures that the data is organized and can be easily manipulated based on time-related operations.

Step 2: Extract Month Information

* A new column named “month” is added to the DataFrame. This column is populated with the month information extracted from the datetime index. Each row in the “month” column represents the respective month of the corresponding datetime index entry.

Step 3: Calculate Seasonality

* The code proceeds to calculate the “seasonality” by grouping the DataFrame based on the “month” column.
* For each month group, it computes the mean of the “detrended” values. The “detrended” values likely represent data with trends removed.
* The calculated mean for each month is then assigned to the “seasonality” column in the DataFrame.

A close-up of a computer code

Description automatically generated

**A graph with blue and orange lines

Description automatically generated**

**Residue**

It is the part left after removing the trend and seasonality



A graph with red and blue lines

Description automatically generated

**Forecasting**

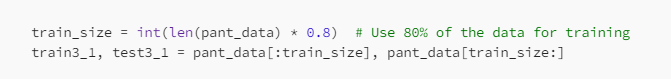
We’ll explore the real-world use of forecasting models in this section, beginning with the Exponential Smoothing technique. This method makes predictions by considering both trend and seasonal factors.

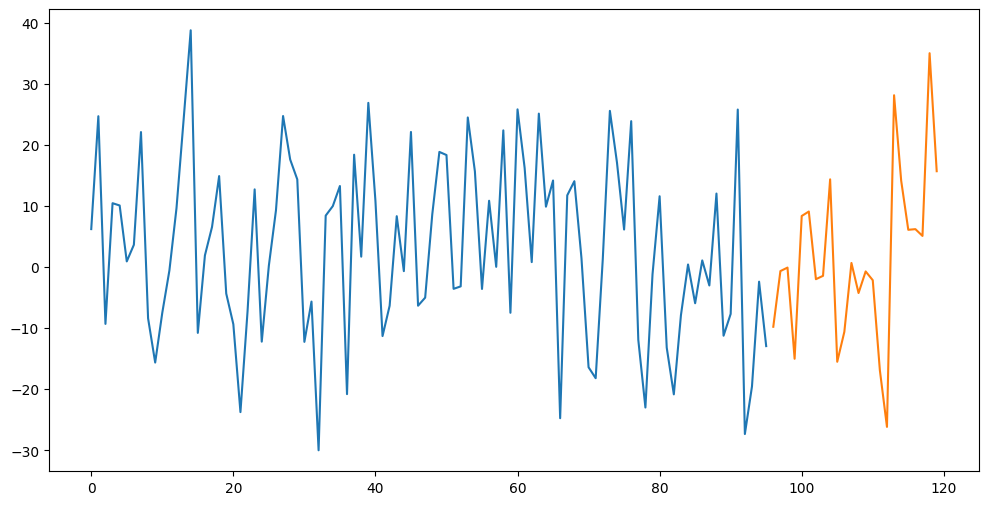
After that, we’ll investigate the Simple Moving Average model, which uses a rolling window technique to smooth the data. We’ll use similar assessment metrics to determine how effective it is.  
Finally, we will examine the Simple Average method, which is a simple technique that establishes a standard for comparison.

Metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) will be used to evaluate its accuracy.  
  
Now that we have some time series data, let’s explore the field of forecasting and assess how well these models do in making future predictions.

**Train and Test Split**

Eighty percent of the dataset is used for training, while the remaining twenty percent is used for testing.





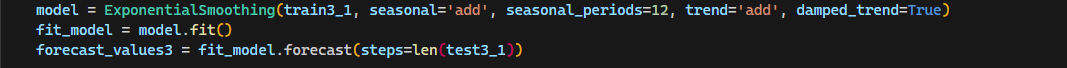
The accuracy of the model is then assessed by calculating Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

A computer error code on a white background

Description automatically generated

**Exponential Smoothing**

We use the exponential smoothing approach to predicting in this section. Parameters such as trend and seasonal components are set up in the model. We produce projections for the test set after the model has been fitted.



A graph showing a graph

Description automatically generated with medium confidence

**Accuracy**

A number on a white background

Description automatically generated

**Moving Average**

Here, we forecast using the Simple Moving Average (SMA) technique. To get the average, the dataset is subjected to a rolling window process. We divide the data, apply the moving average, and produce predictions for the test set in a manner akin to the Exponential Smoothing technique.

A computer code with green and black text

Description automatically generated

A graph of a moving average forecasting

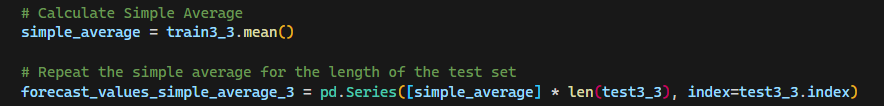
Description automatically generated  
**Accuracy**

A number on a white background

Description automatically generated

**Simple Average**

In this section, we compute the Simple Average using an uncomplicated method. The mean of the training set is calculated, and forecasts are created by repeating this mean value across the whole test set. This basic method offers a point of reference for comparing with more advanced forecasting techniques.



A graph of a graph

Description automatically generated with medium confidence

**Accuracy**

A number on a white background

Description automatically generated

**Detailed Comparison of Forecasting Model Accuracy**

In this section, we conduct a thorough examination of the accuracy of three distinct forecasting models: Simple Average, Exponential Smoothing, and Moving Average. The evaluation is based on key metrics — Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) — providing insights into the precision of each model.

**1. Simple Average:**

The Simple Average model serves as our baseline for comparison. It provides a straightforward forecast by assuming that future values will be the average of historical ones. The associated accuracy metrics act as a reference point for evaluating the performance of more sophisticated models.

**2. Exponential Smoothing:**

Exponential Smoothing, a more advanced method, considers both seasonal and trend components for forecasting. The higher MAE, MSE, and RMSE values compared to the Simple Average indicate that, in this specific scenario, the added complexity does not necessarily result in improved accuracy.

**3. Moving Average:**

The Moving Average model smoothens data using a rolling window operation, and it outperforms both the Simple Average and Exponential Smoothing in terms of accuracy. The lower MAE, MSE, and RMSE values suggest that this model captures the underlying patterns in the data more effectively.

**Comparative Analysis:**

MAE: Moving Average < Simple Average < Exponential Smoothing  
MSE: Moving Average < Simple Average < Exponential Smoothing  
RMSE: Moving Average < Simple Average < Exponential Smoothing

The Moving Average model emerges as the most accurate in this context, demonstrating its effectiveness in forecasting future trends in the given time series data. This detailed comparison allows us to make informed decisions regarding the suitability of each model for specific forecasting scenarios.