DETAILED STEPS TAKEN

**Data:**

* CHIRPS Rainfall data for African Daily.
* FAO SWARMS dataset

**FAO SWARMS dataset (MAJOR FOCUS WAS ETHIOPIA)**

* For each X/Y (Lat/Long) point with country **ID=Ethiopia**, obtain the value of **LocPresent** for each date.
* For each day, a count (i.e. sum the 1s) of the number of the grid points in the country (i.e square representing the country) where LocPresent is true was taken.
* So we get something like: **(Datetime: 1/1/2000, LocPresent: 10)** , (1/2/2000, 0), (1/3/2000, 50) etc.
* On this basis, Our time series data was formed.

**Combining the FAO SWARMS AND CHIRPS RAINFALL DATA**

* For each Lat/Long in Ethiopia, the rainfall value was added to our FAO swarms time series data.
* So we get something like : (Datetime : 1994/03/04, LocPresent : 20, Rainfall\_Value : 60(mm)).
* This formed our time series data.

**Converting the Dates to datetime**

* The Dates were of the Object data type which was parsed into datetime using the python datetime library.

**Lat/Long with No-Rainfall Value**

* Squares where there are no rainfall values were replaced with zero. We didn’t implore the nearest-neighbour interpolation because we had only two squares points (lat/long) where rainfall values were not present.

**Resampling into Monthly data**

* The datetime was resampled into a monthly number of 1° grid squares, aggregating the rainfall and locust present values by taking it's sum.
* A count of **419** months were counted for, with maximum and minimum value for the **Rainfall Value** being **1414.778174** and **0.000000** and that of the **LocPresent** being **496.000000** and **0.00000.**

**Check for Number of Months with Rainfall and No-Rainfall value**

* The Months with no-rainfall values were accounted for, and we had **383 months.**
* The months with rainfall values were also accounted for, and we had **36 months** with rainfall values which cuts across (1987 to 2021)
* The months with no-rainfall values were dropped and we continue our analysis with the months with rainfall values.

**Interactive plotting with Plotly**

* The Rainfall Value was plotted using the plotly library which shows a high rainfall value from late 2019 to date.
* The Locust Value was also plotted using the plotly library which shows a high spike from late 2019 to date.
* The Rainfall Value and Locust value were plotted together to see their relationship against time.
* Plotting these values shows some evidence of Seasonality.

**Check for Seasonality**

1. **Using the Mean and Variance**

* The stationarity and non-stationarity was checked by evaluating mean and variance in different time periods.
* Mean and Variance and other statistics of a Stationary time series remains constant, Hence, the conclusions from the analysis of stationary series is reliable.
* A stationary time series will not have trends, and seasonality,etc.

Stationary data is easier to analyze and any forecast made using non-stationary data would be erroneous and misleading.

1. **Using Augmented Dickey-Fuller test**

Statistical tests make strong assumptions about your data. They can only be used to inform the degree to which a null hypothesis can be rejected or fail to be rejected. The result must be interpreted for a given problem to be meaningful.

* Nevertheless, they can provide a quick check and confirmatory evidence that your time series is stationary or non-stationary.
* The Augmented Dickey-Fuller test is a type of statistical test called a unit root test.
* The intuition behind a unit root test is that it determines how strongly a time series is defined by a trend.
* There are a number of unit root tests and the Augmented Dickey-Fuller may be one of the more widely used. It uses an autoregressive model and optimizes an information criterion across multiple different lag values.
* The null hypothesis of the test is that the time series can be represented by a unit root, that it is not stationary (has some time-dependent structure). The alternate hypothesis (rejecting the null hypothesis) is that the time series is stationary.
* **`Null Hypothesis (H0)`**: If failed to be rejected, it suggests the time series has a unit root, meaning it is non-stationary. It has some time dependent structure.
* **`Alternate Hypothesis (H1)`**: The null hypothesis is rejected; it suggests the time series does not have a unit root, meaning it is stationary. It does not have a time-dependent structure.
* We interpret this result using the **p-value** from the test. A **p-value** below a threshold (such as 5% or 1%) suggests we reject the **null hypothesis (stationary)**, otherwise a **p-value** above the threshold suggests we fail to reject the **null hypothesis (non-stationary)**.
* **`p-value > 0.05`: Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary.**
* - `p-value <= 0.05`: Reject the null hypothesis (H0), the data does not have a unit root and is stationary.

**Check for Autocorrelation function and Partial Autocorrelation function**

* We checked for the autocorrelation function and partial autocorrelation of the Untransformed Locust data.
* A 25 month lag of **ACF** and **PACF** was used for this.
* This was implemented using some helper functions.

**Square-root transformation of the locust data**

* Square root transformation for the locust data was taken to help achieve equality of Variance and a normal distribution.
* The varying mean and Variance of the square-root transformed locust data was examined as well as the **Augmented Dickey-Fuller test.**
* A 25 month lag of the **ACF** of this Square-root transformed time locust series showed a high degree of autocorrelation.

**First-Order Differencing**

* The first-Order differencing of the Square-root transformed locust data was taken and Stationarity was checked and had no unit root and is stationary.
* The **ACF** and **PACF** were plotted.
* Calculating the first order differencing of a time series is useful for converting a non stationary time series to a stationary form.
* It is calculated as follows. The i-th data point Y\_i of a time series is replaced by Y'\_i = (Y\_i - Y\_(i-1).

**Seasonal First-Differencing**

* The Seasonal first-order difference was carried out using a sliding window of 4, which represents each quarter as a periodicity, because most months with no rainfall values were dropped earlier.
* Stationarity was also checked and evaluated to be Stationary.

**MODELLING**

**SARIMA (Seasonal Autoregressive Moving Average model)**

### **Seasonal Autoregressive Integrated Moving-Average (SARIMA)**

Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component.

It adds three new hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

Trend Elements:

There are three trend elements that require configuration. They are the same as the ARIMA model, specifically:

* p: Trend autoregression order.
* d: Trend difference order.
* q: Trend moving average order.

Seasonal Elements:

There are four seasonal elements that are not part of ARIMA that must be configured; they are:

* P: Seasonal autoregressive order.
* D: Seasonal difference order.
* Q: Seasonal moving average order.
* m: The number of time steps for a single seasonal period. For example, an S of 12 for monthly data suggests a yearly seasonal cycle.

SARIMA notation: SARIMA(p,d,q)(P,D,Q,m)

**AR(p)**

* **autoregression model** i.e. regression of the time series onto itself. The basic assumption is that the current series values depend on its previous values with some lag (or several lags). The maximum lag in the model is referred to as **p** . To determine the initial **p**, you need to look at the **PACF** plot and find the biggest significant lag after which most other lags become insignificant.

**MA(q)**

* **moving average model**. Without going into too much detail, this models the error of the time series, again with the assumption that the current error depends on the previous with some lag, which is referred to as q . The initial value can be found on the ACF plot with the same logic as before. Let's combine our first 4 letters:



What we have here is the Autoregressive–moving-average model! If the series is stationary, it can be approximated with these 4 letters. Let's continue.

**I(d)**

* **order of integration**. This is simply the number of nonseasonal differences needed to make the series stationary. In our case, it's just 1 because we used first differences. Adding this letter to the four gives us the ARIMA model which can handle non-stationary data with the help of nonseasonal differences. Great, one more letter to go!

**S(s)**

* this is responsible for **seasonality** and equals the season period length of the series With this, we have three parameters: (P,D,Q)

**P**

* **order of autoregression** for the seasonal component of the model, which can be derived from PACF. But you need to look at the number of significant lags, which are the multiples of the season period length. For example, if the period equals 24 and we see the 24-th and 48-th lags are significant in the PACF, that means the initial P should be 2.

**Q**

* similar logic using the ACF plot instead.

**D**

* **order of seasonal integration**. This can be equal to 1 or 0, depending on whether seasonal differences were applied or not.

The **rainfall value** was used as our **exogenous variable** and the **Square-root transformed locust data** was used as the **endogenous variable**

We used the **`Grid Search`** to find an optimal set of parameters that yields the best performance for our model. This strategy helped us in selecting the best parameter for our **ARIMA** Time Series Model.

The **Grid Search** output suggests that **`SARIMAX(2, 1, 0)x(2, 1, 0, 4)12**` yields the `lowest AIC` value of `**AIC:132.0736131874681**`. Therefore we should consider this to be the optimal option.

**Time series analysis** comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values.

Time series are widely used for non-stationary data, like economic, weather, stock price, and retail sales

we analysed a time series of the monthly number of 1° grid squares infested with desert

locust Schistocerca gregaria swarms throughout the geographical range of the species from

1930–1987.

The **autocorrelation function (ACF)** defines how data points in a time series are related, on average, to the preceding data points (Box, Jenkins, & Reinsel, 1994). In other words, it measures the self-similarity of the signal over different delay times.