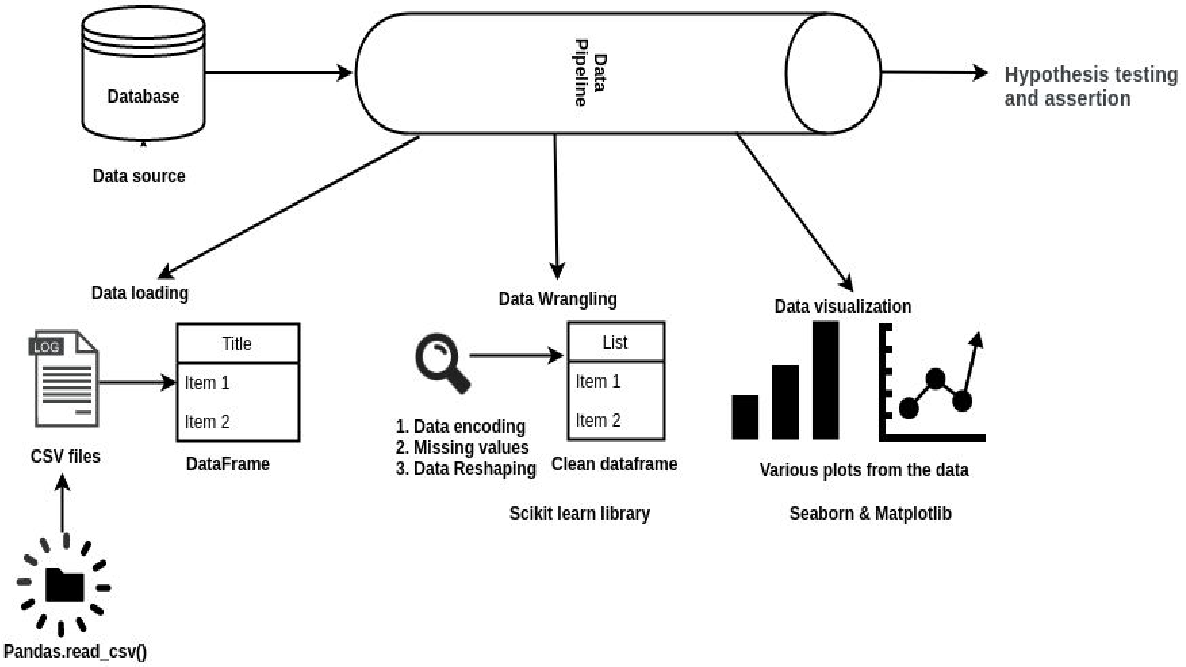
System Design Specification Document

Data sources

The data used is structured environmental data that was collected by several Federal Government agencies such as the Center for Disease Control and Prevention and the National Oceanic and Atmospheric Administration in the United States. The data is hosted on the drivendata website [1].

The Pipeline:



Data loading

The data was made available in 3 *comma-separated values* (csv) files. The first file contained all the training data that was used to learn from the acquired data. This file is what enabled us to establish relationships and also develop a better understanding of our data. The second file contained the number of dengue fever cases recorded in the time this data was collected and finally, the last file contained the testing data.

We decided to use Pandas(an open-source library for the Python programming language that is optimized for data manipulation) because of its many features and ability to work with numerous file types. The data to be analyzed was in CSV format and pandas has a `pandas.read_csv()` method that reads these types of files into a pandas DataFrame, which is a data structure in a table-like form.

Data cleaning and wrangling

Data cleaning and wrangling involves tidying up raw data to make it useable in analytical algorithms. The approach we took involved three major steps namely, encoding text data, handling missing values, and reshaping the data.

Columns with missing values in the pandas dataframe are detected with the help of the ***pandas.describe()*** method which generates descriptive statistics about a dataframe such as central tendency(mean,mode), dispersion(max,min,percentiles), and the count. It is from the count that we can identify missing values in the dataset/dataframe since and inconsistency in the count shows this.

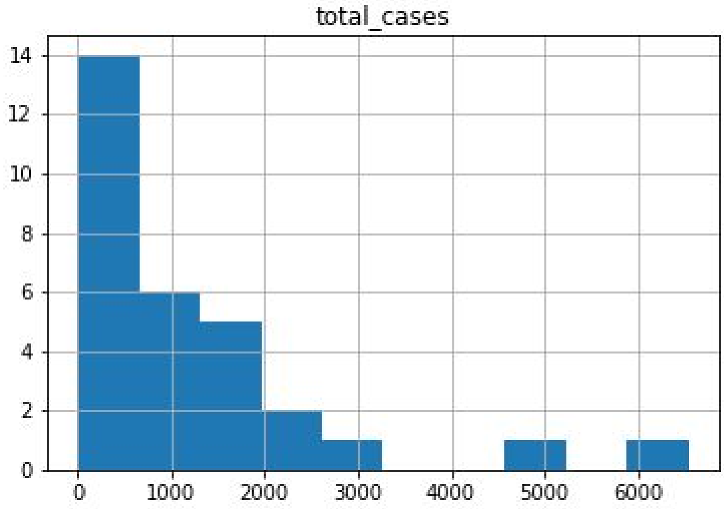
The datasets used for the analysis had a lot of missing values with `ndvi_ne` column missing up to 158 values. 20 other columns also missed values but at reduced severities. The approach used for dealing with missing values was Imputation which involves the replacement of missing data with substituted values. This process was made possible with the use of the scikit-learn python library. The method used was `sklearn.impute.SimpleImputer()` which is an imputation transformer for completing missing values. The strategy used for replacing the missing numeric values was `mean` which calculates the mean along each column and then replaces each missing value along that column with the mean.

In reshaping of the data, we carried out label encoding for the `city` column once again using another method from scikit learn. Label encoding is a process of conversion of categorical text values into numerical data. The method used was `sklearn.preprocessing.LabelEncoder()` which encodes labels with values between `0` and the number of different classes in that particular column minus one i.e.`n_classes(the number of classes) - 1.`

Exploratory data analysis

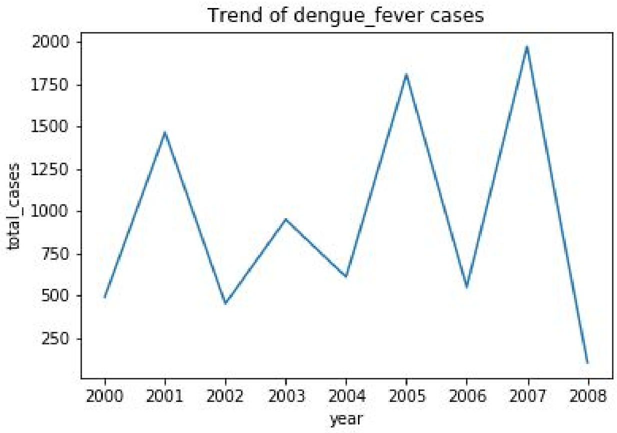
Through exploratory data analysis, attributes of the data will be established and preliminary insights into the data made with the aid of summary statistics. Hypotheses such as does an increase in temperatures lead to increased dengue fever cases?; will be tested and these assumptions into the data statistically clarified by means of determination of confidence_intervals such as the 68%,and 95% confidence intervals. These append a certain degree of certainty that an event/hypothesis is truthful. Here we also make use of the pandas and numpy libraries due to their richness in statistical and mathematical functions.

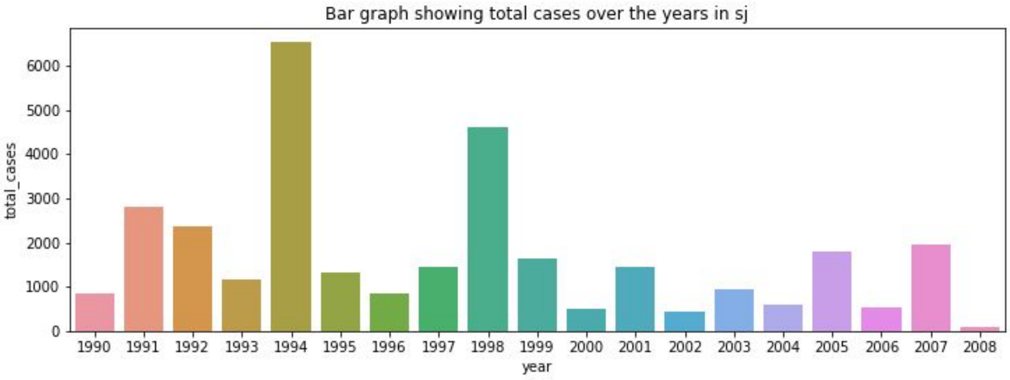
Because of the data is highly-right skewed (positive skewness), most of the data points are asymmetrical about the mean and we will test-apply square root and logarithmic transformations to reduce the skewness. The histogram of total cases against their frequency of occurrence below illustrates this skewness. There also appears to be some outliers that we have to deal with either by dropping them or keep them if they embed vital information.



Data Visualization

In data visualization, we attempt to get insights about the data by representing data in a graphical format. Several visual elements such as charts, tables, graphs, and maps can be used for this task. We put to task the matplotlib and seaborn python libraries to aid in visualizing the data given their excellent graphical tooling and coherency of operation with pandas dataframes which is the data structure we are using to store the data for analysis. We apply line graphs to clearly show trends in our data, bar graphs to visualise relationships between categorical independent values with numerical dependent values and scatter plot to visualize the correlation relationships between dependent and independent variables. Below are some of the visual insights about the spread of dengue fever.





Katwere Leo 17/U/4874/PS
Kengo Wada 17/U/5026/PS
Mayanja Benjamin Vincent 17/U/545
Mugisha Stephen 17/U/6337/PS