

# **CSC8499: Selecting different dataset comparison techniques based on automatic data set analysis**

Disha Rajpal

MSc in Advanced Computer Science,  
School of Computing Science, Newcastle University.  
D.Rajpal2@newcastle.ac.uk

**Abstract.** This dissertation has researched method for visual comparison and statistical analysis of datasets. The statistical tests are used to operate on the datasets to get more information in a driven and targeted way. The math's involved will help in forming concrete conclusions about the datasets than just guesstimating in the real world. Why it is valuable with a visual comparison of datasets is because using the visualizations in datasets will provide some high level of understanding. Also, comparison among them will provide the distribution details like skewness, kurtosis, which will be easy to use and to understand easily, which can be presented based on user's option such as histograms. Also, by applying the statistical techniques, the user can get more profound and more fine-grained insights into how the data is structured. Although specific tests have been used earlier for a longer time, to choose the right test based on dataset distributions is still a challenge in the real world. Also, the project can deal with very similar as well as dissimilar data sets used in the comparison. It would result to visually present information about how two or more datasets are different or like graphs based on user choice such as the boxplot, heatmaps and more so it is the user based tool, and here the user has passed the two different datasets based on Breast cancer datasets. That tool will also perform rigorous statistical tests for checking normality between similar and dissimilar data sets such as Anova-test, T-tests, Tuckey-HSD test and more.

**Keywords:** Data Visualizations; Statistical Analysis; User-based tool; Data Analysis.

**Declaration:** I declare that this dissertation represents my own work except where otherwise explicitly stated.

## 1 Introduction

Visualization is one of the essential components of research presentation and communication due to its ability to convert large datasets into compelling graphics, visualization techniques applied during data analysis with exploration analysis, statistical modelling validation up to results.

There is also a demand for meaningful and useful statistical analysis of available results. The increasing accessibility and quantity of data require effective ways to analyze and communicate the datasets in simple formats. There is an evidence of the fact that it is already a sizable repository of algorithms and tools to support the developer community. Further, developing new algorithms, tools, and systems have become a relatively easy task.

The process of data visualizations described from a high-level perspective consisting of two simple steps, i.e. bringing data into memory, then applying visualization algorithm. Often, statistical data consists of sample sets measured on more than two variables. For example, a dataset may consist of water samples, where each sample was tested for various contaminants such as sulphur, chlorine, ammonia, phosphorus, and other pollutants. Such data is called high dimensional or multivariate data [1]. It is easier for a researcher to detect or extract some information from graphical representation of experimental data from raw numbers.

Visualization of multivariate data is hence often used to provide a synthetic view of patterns or clusters formed by the data or to detect outliers [4]. That is why researchers, technicians or practitioners working with multidimensional datasets are very interested in data visualization software.

For many consumers (end-users) of such statistical tools; however, the major challenge lies in understanding the statistical concepts and their applications.

Methods of statistical inference depend on a complex web of assumptions about how the data were collected and analyzed and how the results presented the full set of assumptions embodied in a statistical model.

The model is a mathematical representation of data variables, and it would capture all sources of such variability accurately. Many problems arise, however, because this statistical model often incorporates unrealistic or at best-unjustified assumptions [3].

To perform the correct analysis, we compare a set of classifiers over multiple datasets. In specialized literature, many papers provide reviews on specific topics, and they also use the statistical methodology to perform comparisons. For example, in the review of ensembles of decision trees, the non-parametric test also applied to the analysis of performance [2].

So, our focus is to perform a statistical analysis of two data sets and present them visually by using the data visualization technique which tells how two or multiple data sets are different or similar. What would make the project enjoyable is the ability to deal with very similar as well as dissimilar data sets used in the comparison? Most of the techniques described work for 1-dimensional data sets, or higher-dimensional datasets provided their structure and data types are very similar. However, this project would aim at handling acute and edge cases also visually, and it is a user-interactive tool which provides user multiple options for selection of different visualizations of a graph to compare two different datasets.

## 1.1 Aims and Objective

The rapid development of computers of the last decades allowed people to store and analyze an increasing number of data. Researchers have to deal with so many of variable measurements that they obtained from the objects observed in experiments. In some situations, the structure of the objects under consideration is well understood, and a rather good model is known (e.g. a normal distribution) [4].

A data-driven search for statistical insights and models is traditionally called Exploratory Data Analysis [8]. The nature of this information can be statistical (mean, variances, and so on) or more closely related to human observation capabilities (structures, clusters or dependencies)[8]. Understanding datasets is an essential part of the scientific process. As the data becomes large and complex, visualization and data analysis techniques are required not only to address significant scale data issues but also to understand the process to produce data.

Traditional visualization approaches overlook available uncertainty information [2]. As the importance of visualizing these large, complex datasets grow, the actual task of visualizing them becomes more complicated.

Also, there are difficulties in applying pre-existing methods, and the lack of distinct visualization techniques leave uncertainty visualization and unsolved problem [2]. Also, Common statistical tools such as R (R Core Team,2013) do not scale well to such large data sets, and especially concerning memory requirements [9].

The aim is to learn and find the statistical methods for comparing the datasets.

The analytical approach that will be used while analyzing the datasets is based on datasets. The types of data sets with the relevant test are mentioned below [10].

Table 1 Statistical method for comparison of data sets		
Goal	Kind of data	
	Parametric	Nonparametric
Compare one group with a hypothetical value	One-sample t-test	Wilcoxon test
Compare two unpaired data groups	Unpaired t-test	Mann-Whitney test
Compare two paired data groups	Paired t-test	Wilcoxon test
Compare three or more unpaired data groups	One-way ANOVA	Kruskal-Wallis test
Compare three or more paired data groups	Repeated-measures ANOVA	Friedman test
Quantify association between two data groups	Pearson correlation	Spearman correlation
Predict value from another variable	Simple regression	Nonparametric regression
Predict value from several variables	Multiple regression	-

**Figure 1:** A screenshot of statistical method for comparison of datasets [26].

The aim is to visually present the information about how two or more datasets are different or similar by developing the interactive user tool.

That tool will visualize user optional representations for two different datasets also perform rigorous statistical tests for comparing similar and dissimilar data sets.

The objectives are:

1. To visually present the information of similar and dissimilar datasets which would be handling acute and edge cases by finding the distribution difference between two datasets. There has been specific research was done, which is mentioned under references.
2. Choose appropriate statistical methods through a literature review.
3. Implement statistical analysis.
4. Represent the dataset visually in the various graph's forms based on the user's choice as well as basic graphs to understand datasets difference such as histograms, replots, heatmaps and more.
5. Implement tool front-end with user -interface and visualize.
6. Implement tool backend for data analysis.

## **1.2 Structure of the Dissertation**

The structure of this dissertation report is as follows:

In section 2, background research will be discussed. The data visualisation comparison methods in two datasets in this dissertation will be explained in detail.

In section 3, Design and implementation will be shown.

In section 4, Evaluation and test cases will be discussed.

We will conclude this dissertation in section 5, by outlining the limitations and laying out the scope of this project in the future. Section6, will list out future research directions

## **2 Background Research**

### **2.1 Introduction to Data Visualization**

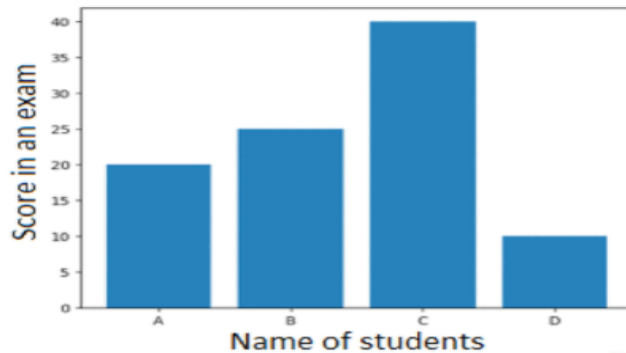
The exploration of the data includes searching the data in a graphical or statistical presentation to find relationships and patterns. Over the last decades, as visualization has been a research discipline in its own right for about 30 years [5]. Several technologies used to extract the best possible information about a dataset.

Data representation refers to store, process and transmits data. Information is not equivalent to data. Representations are useful to find insights hidden data [6]. Representations transform the hidden data into useful data before the modelling data visualisation helps in finding a suitable method of data analysis. In addition to a detailed study of graphs and charts through visualisation data, it consists of several other activities [5].

- Summarizing data
  - Grouping data
  - Exploring the relationship between various attributes
  - Identifying pattern and trends
  - Constructing regression models
  - Constructing classification models.
- To do another round of analysis, data visualisation is required.

## 2.2 The importance of Data Visualization

The visual data is easy to understand as compared to data in any other form. This dissertation will get better idea of data by using the visualization rather than just looking in columns of an excel sheet. For instance, it's easy to see a pattern emerge from numerical data given in following graph below.



**Figure 2:** A simple example of data visualization [6].

Visualizing data has many advantages, such as following [5]:

- Easy to understand complex data.
- Easy to find outliers, target audiences and predict future markets.
- Data can be explored through interactive visualizations.

### 2.3 Data Wrangling

To find the conclusions from the visualized data, to handle the data and transform into the best possible representation there only data wrangling is used. It is the discipline of augmenting, transforming, and enriching data in a way it allows it to be displayed and understood by machine learning algorithms [5]. Data Wrangling is the process in which we follow various steps to find the desired output from the data. Initially, data collected from the desired source it should be either in raw form or various other forms such as CSV, JSON or Html. The data gets imported as a data frame and cleaned. The cleaned data is transformed into corresponding graphs here, we use the visualisation techniques, from which the insights can be modelled, based on the insights we can communicate the results.

### 2.4 Data Visualisation Methods

Some of the most common data visualisation methods or techniques are [11]:

1. **Tables:** A table visualisation helps in display data from a metric set using a tabular view. A table is also known as a data grid or data table. A data table or a spreadsheet is an efficient format for comparative data analysis on categorical objects. The items that compared placed in columns and categorical objects are in rows.
2. **Charts:** Visualisation is more intuitive and meaningful, and it is essential to choose appropriate charts to visualize data.

Some commonly used charts are [12]:

- **Column Chart:** Column charts generally use vertical columns to show a numerical comparison between categories, and the total number of columns should not be too large.
- **Bar Chart:** They are similar to the column charts, but the number of bars can be relatively large, as compared with the column charts, the position of its axes interchanged.
- **Scatter Plot:** It represents the two variables in the form of points on a conventional coordinate system. It tells the relationship between two variables; the position of the points describes the value of the variable. We can infer the correlation between variables by observing the points in the chart.
- **Line Chart:** This chart visualizes the ways to demonstrate change or impact over multiple periods [11]. Lines allow to add multiple variables and compare their performance within a specific time frame [11].

3. **Maps:** The maps are divided into three types: regional map, point map and flow map [12].
4. **Graphs:** They are also one of the ways to visualize the pattern of data. Describing some most used graphs such as:
  - **Histograms:** It is the particular type of vertical bar graph that presents numeric data and its frequency distribution [13]. Here the distribution happened across the time, but data could be plotted on any temporal scale [13].
  - **Heatmaps:** A heatmap contains values representing various shades of the same color for each variable to be plotted. It generally tells the correlation among each variable with each other.

## 2.5 Tools and Libraries Visualisation

### Tools

There are several approaches for representing data visualisations. It depends on the background, for non-coding tool there is Tableau, which gives an ability to get a good view for the data[6]. MATLAB and R are also heavily used in data analytics. For implementation tool researchers prefer Jupyter , also it is the part of implementation.

Jupyter Notebook is most widely used for system interactive literate programming[28].

It was designed to make data analysis easier [28].Jupyter notebook originated from IPython[28] , it also supports other programming language such as R, JavaScript, C. It also used to interactive computational notebook environments , which allow some area of notebook to run with immediate visualisation of results and test. A notebook composed of cells which are of three types : code, markdown and raw[28].A code cell is executable cell .The tool converts notebook in other from such as html , pdf .It use kernel to execute code. After execution kernel allocates number which indicated the execution order [28].Use can execute given cell multiple times. The implementation has been done using Jupyter notebook.

However, Python is one of the most popular languages used for analytics, modelling, and other data science algorithms. This dissertation has been implemented in Python and its libraries. Here describing why here Python used and libraries.

- **Python:** The use of Python in data science has reached unprecedented levels due to the free available tools and libraries[14].The speed which requires to manipulate and visualise data, combined with the availability of libraries makes Python as the best choice[6]. It powers websites , backend services , native desktop applications, image processing systems, machine learning pipelines, data transform systems [15].Recently Python has become the programming language of choice with R being the second choice in the field of the data science community. Its extensive ecosystem consists of various libraries for every aspect of data science.

The main advantages of language are :

- It has similar syntax to native English ; it can be understood by only reading .
- It has large number of third-party modules and libraries for any application .
- It supports object-oriented programming and procedural programming paradigms depending on user needs .

### An overview of Libraries

A large number of third-party libraries that are useful for data analysis are available in Python. Python provides a package ecosystem, after doing the research mentioning the libraries which are also the part of the dissertation.

- **NumPy:** NumPy[16] builds on the successful numeric array object. It is a basic package that implements various data manipulation operations on top of array data structures[15]. It contains highly efficient implementations of data structures and commonly functions for statistical computing [15]. It also speeds up complex tasks.
- **Pandas:** Pandas [17] is a powerful and easy to use the open-source library for tabular data manipulation [15]. It has the data structures which are suitable for working with labelled data. Pandas provide the data frame to the datasets because the data structure has index objects which stores are labelling information about each tick along that axis [17].
- **Matplotlib:** Matplotlib[18] is a python implementation of MATLAB like plots. It is a library for creating static or interactive data visualisations [15]. It is generally used to draw graphs as explained in section 2.4 about the charts and graphs Matplotlib is the underlying library to provide such representation for datasets.
- **Seaborn:** Seaborn is a library for making statistical graphics in Python. It is built on top of Matplotlib and closely integrated with the Pandas data structure[18]. It supports the categorical variables to show observations or aggregate statistics. It provides the options for visualising univariate or multivariate distributions and for comparing them between subsets of data[18]. It aims to make visualisation of the central part of exploring and understanding data. It is dataset oriented provides the visualisation over the data frame and arrays containing whole datasets.
- **SciPy:** It is a collection of mathematical algorithms and convenience functions built on NumPy extension of Python [19].It has the power to interact with Python session provided by the user with a high level of commands for visualising data. Using SciPy, an interactive Python session becomes a data-processing and system prototyping systems such as MATLAB, RLAB[19].



## 2.6 Methods for comparison of datasets

Various methods might be available to compare datasets. Datasets can be compared in different forms, based on the research hence describing some of the methods below :

- **Graphs and Charts:** As discussed above in section 2.4, graphs are one of the most used technique to visualise data, we can easily compare two datasets with the help of graphs. For example, two data frames can be visualised in a histogram graph with the help of that we can easily read and understand the pattern of datasets and their variables. We can use multiple graphs or charts to compare the datasets, or the data groups within the datasets such as line chart, bar chart, histogram, scatter plot, heatmaps.
- **DataCompy Library:** It is a package to compare two Pandas data frames and provide human-readable output with their differences. It requires a user to provide a list of columns which will act as a key to join[20]. If the library detects the duplicates on the join key, then it will sort the fields and join with row number [20]. For using the DataCompy library, we have to install the package in Python script.
- **Statistical Methods:** The statistical methods included in the various types of comparison, but each method needs some criteria for selection. The main criteria for selection first are a data type, and second is the goal. Every statistical method depends on the assumptions about how data is collected and analysed and how the analysis results will be selected for presentation. As statistical methods are a significant part of the dissertation, we will discuss more deeply in another part of the background research.
  1. **Check the integrity of data:** It means reviewing the datasets in its entirety; it means it will pass the test else fail. For this, it requires two alternative algorithms [21]. First is MD5 Checksum it is an algorithm which returns hexadecimal number for the contents of a file [21]. Second is a SHA1 algorithm; it is a similar hexadecimal algorithm which converts the file contents into a string.
  2. **Using Equals parameter :** This is other way to compare two datasets , it is used by using the Pandas data frame . It generally tells us whether the datasets are matching with each other or not . This method has also been used in implementation we can understand it more clearly in implementation section .

## 2.7 Visualisation methods for data distribution

The visualisation method for data distribution tells us whether our data is normally distributed or not .It is also a most important part of the dissertation , with the help of this we can find whether our data is normally distributed or not . Some methods are:

- **Boxplot:** The box plot becomes the standard technique for representing 5-number summary which consists of the minimum and maximum range value, upper and lower quartiles and median [22]. Its collection of values helps to summarize the distribution of dataset. Also, their reduced representation summary provides the easier way to compare datasets, which represents the only important characters that requires to analyze. The box indicates the position of the upper and lower quartiles , the interior shows the inner quartile range , the area between the upper and lower quartiles consists of 50% of distribution .Lines or whiskers are extended to the

extrema of the distribution .Outliers indicates the independently marking of them in plot [22].

- **Scatter Plot:** The scatter plot is the powerful tools for data analysis. By using scatter plot, we can identify the relationship between two attributes, cluster of points and outliers [23]. However due to large datasets scatter plots have degree of overlap, which obscures the true density of data values [23]. The overlap optimized scatter plot allows variable degree of distortion and a variable degree of overlap [23]. The distortion is based on a linear direction between x and y axes to ensure the data is equally distributed in x and y dimension .For smoother visualisation it should have maximum number of overlapping points where x is in [0,1] as maximum overlap degree for visualization [23].
- **Histogram and Density plot :** A histogram is a kind of bar plot that gives a discretized display value frequency[35].The data points are split into discrete and spaced bins and number of data points are plotted in each bin[35].A related type of plot is density plot [35], which is plotted by computing an estimate of continuous probability distribution that generates the observed data [35].

## 2.8 Overview of statistics

Statistics is a known as a combination of analysis, collection, interpretation, and representation of numerical data [6]. Statistical analysis can be divided into two categories one is descriptive statistics and other is inferential statistics [25].

Descriptive statistics contains methods for data representation using visualisation, creating summary graphs. It uses figures or tables which includes frequency, percentages, measures of central tendency, measures of variation. Descriptive analysis provides information before initiating the statistical inference, subject to assumptions that is the sample is randomly drawn from the target population [25].

The sample descriptions later used to infer some information about population with some uncertainty by testing the stated research and find the confidence interval for respective parameters based on their estimations. Thus, the statistical procedures are based on assumptions rather than guessing about the population. These procedures require the sample assumptions regarding probability distribution of the random variable.

Assumptions regarding probability distributions have been made in the dissertation, and later hypothetical tests are run, as is a part of the aim of the dissertation Here some basic statistics types are:

- **Probability distribution:** It is a function that provides probability for every possible function [6]. It is used most of the time for statistical analysis. There are two types of probability distribution, namely discrete distribution, and continuous probability [6]. The discrete probability distribution shows all the valued that random variable takes. The continuous probability shows the probability for each possible value of continuous random variable [6].

- **Measures of Central Tendency:** They are defined as the averages and describe the typical or central values for a probability distribution. There are three kinds of averages [6]:
  - **Mean:** It is the computation by summing up all measurements and dividing the same by number of observations.
  - **Median:** It is the average of two middle values. The median is less prone to outliers as compared to the mean when outliers are distinct values in dataset [6].
  - **Mode:** It is defined as most frequent value, there may be more than one mode where the multiple values are frequent [6].
  - **Measures of dispersion:** It is a variability is the extent to which the probability distribution is stretched or squeezed [6]. The difference measures of dispersion as follows
    1. **Variance:** It is the squared deviation from mean [6]. It tells how far the set of numbers are spread out from their mean [6].
    2. **Standard deviation:** It is the square root variance.

## 2.9 Statistical Test for comparison of datasets

As we have discussed the basic statistics parameters, now describing the statistical methods which are part of dissertation for assumptions.

The following tests for comparison are [26]:

- **T-test:** It is a statistical test that are used in comparison of parametric data [26]. One sample T-Test finds the difference between mean of one group and population mean [26]. Paired T-Test measures difference between means of two paired groups.
- **Wilcoxon test:** It is used for comparison of non-parametric data. It compares two paired groups or one group [26]. In this one group compares mean of the group with a hypothetical value which is generally defined by researchers.
- **Mann-Whitney test:** It is generally used to compare two unpaired sample and non-parametric data [26].
- **ANOVA test:** It is a statistical method that compared parametric data [26]. It is used to compare three or more unpaired data groups [26].

## 2.10 Related work

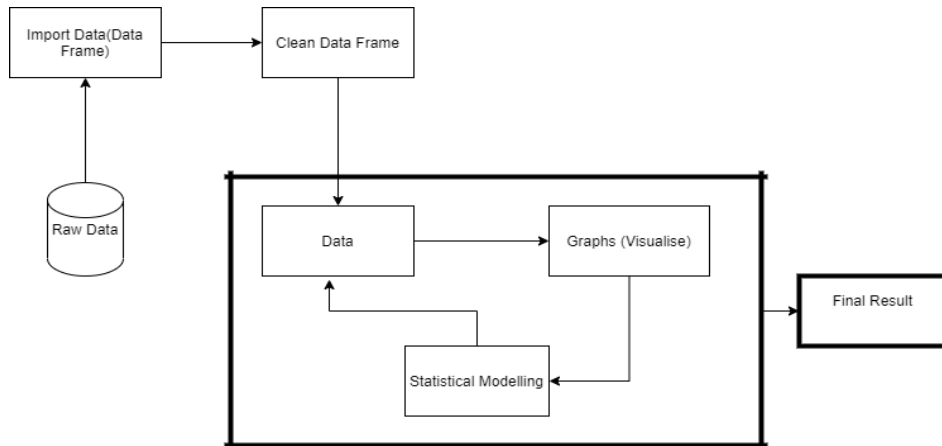
Panda power BSD licensed power system analysis tool [36].The tabular data structure used to define networks which are based on Python library pandas [36].IT also allows handling input and output parameters [36].The implementation in Python makes panda power easy to use and allows extension with third-party library[36].Similarly our tool is also based on Python and Python library pandas also dealing with input and output parameters .Furthermore these are other Python based tool such as Nengo[37]. It is a python-based tool for building large scale functional brain models

## 3 Design and Implementation

Before starting the implementation, it is most important to design the project or workflow which is how to execute the plan .Here the work flow is to collect the data first , then import into data frame , after that clean the data using data analysis techniques , then perform the visualisation after that perform statistical modelling and then final result .

In above paragraph the explanation of designing and implementation is based on agile methodology. Also, agile and data science or data visualisation are very different fields.

The agile methodology use as this project is about extracting information from raw and implementing statistical techniques .IT requires assumptions and violations which leads to uncertainty, therefore agile methodologies can be successful. Agile methodologies can cope with unpredictable realities of generating helpful analysis from raw data. This kind of projects also go down in different paths and use different techniques. Therefore, Agile can be the right fit for this kind of projects.



**Figure 3:** A basic design of project

### 3.1 Data Collection

For collecting the data ,researchers use different data collecting strategies such as surveys using structured questions or collecting data on large number of variables from large and representative samples of respondents [29].There are generally two methods which are primary data collection and other is secondary data collection. Primary data are data that are collected for the specific research problem. Other than data which are collected by researchers and made available to reuse is secondary data collection.

Data sets are also collected university- based researchers, these organizations set up chiefly for the purpose of releasing and disseminating secondary data to general research community [29].

In the dissertation, collecting two different datasets, as the model is user based the user can collect ant two datasets from and location based on user what research he wanted do.

Here primarily the data is based on breast cancer, the first data set has been collected from mldata.io [30] and the other dataset has been collected from Kaggle website [31].

Let us understand about the data, basically the data understanding is based on user but as the implementation has been done for user purpose hence finding the problem to solve from the data.

Breast cancer is the most known cancer in women. Early detection of cancer greatly increases chances for successful treatment. There are two different datasets based on breast cancer which are using for the implementation .The datasets has different columns which are responsible for breast cancer the changes in different types of cells in breast will describe whether the cancer is benign or malignant. Currently both the datasets are in CSV format.

A	B	C	D	E	F
clump_thickness	uniformity_of_cell_si	uniformity_of_cell_sha	marginal_adhesi	single_epithelial_cell_size	bare_nuclei
5	1	1	1	2	1
5	4	4	5	7	10
3	1	1	1	2	2
6	8	8	1	3	4
4	1	1	3	2	1
8	10	10	8	7	10
1	1	1	1	2	10
2	1	2	1	2	1
2	1	1	1	2	1
4	2	1	1	2	1
1	1	1	1	1	1

**Figure 4:** Screenshot of CSV file for Dataset.

### 3.2 Load Dataset

The second most important thing is loading the dataset, after the background research the most feasible library for converting the dataset into data frame is by using Pandas. As mention in section 2.4, Pandas [17] is a powerful and easy to use the open-source library for tabular data manipulation [15].It converts the any type of dataset into data frame .Also here our tool will ask user to input the dataset CSV file and user can insert dataset file from any location of its local system . Also, if the user inserts wrong file it will tell user that the file is not right. Let's see how it happens

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In this part, importing the relevant libraries which is going to require for the implementation as of now here Pandas library is imported. The output here has been shown as:

```
validate_file1 =pd.read_csv(file1)
validate_file2 =pd.read_csv(file2)
df_breast_cancer_data=validate_file1
df_breast_cancer_data_new=validate_file2
df_1=df_breast_cancer_data
df_2=df_breast_cancer_data_new

Enter the path of file 1
C:\Users\raj\Documents\breast_cancer_dataset.csv
Enter the path of file 2
C:\Users\raj\wisconsin_breast_cancer.csv
this is file C:\Users\raj\Documents\breast_cancer_dataset.csv
this is file C:\Users\raj\wisconsin_breast_cancer.csv
```

**Figure 5:** Screenshot of input CSV file in Jupyter.

Here user has entered the input files from the local system and these files are relevant to the tool. And it is validating that the files are correct.

### 3.3 Data cleaning

Data cleaning is the most important part for the data science which deals with detecting and removing errors and inconsistencies from data to improve the quality of data [32].

The problems are present in single data collection such as files or database which generates several problems such as misspellings, missing values, type of data. In this dissertation there are certain cleaning problems that resolved which are removing missing values and converting the type of dataset from float into integer. As the implementation has done in the part of data analysis, the implementation code is under next section.

### 3.4 Initial Data Analysis

Data analysis is the process that converts raw data into relevant knowledge. NumPy library is used here which helps in working on numerical dataset. As the analysis is based on user point, the tool will provide the function to the user so that it can perform analysis for any other dataset. Let's see how it happens, the analysis checks the null values, the unique value in dataset, the memory usage by the dataset, type of the dataset and later on it will display in the form of list.

```
def get_df_info(df, include_unique_values=False):
    col_name_list = list(df.columns)
    col_type_list = [type(cell) for cell in df.iloc[0, :]]
    col_null_count_list = [df[col].isnull().sum() for col
in col_name_list]
    col_unique_count_list = [df[col].nunique() for col in
col_name_list]
    col_memory_usage_list = [df[col].memory_usage() for
col in col_name_list]
    df_total_memory_usage = sum(col_memory_usage_list) /
1048576
    if include_unique_values:
        col_unique_list = [df[col].unique() for col in
col_name_list]
        df_info = pd.DataFrame({'column_name':
col_name_list, 'type': col_type_list,
                                'null_count':
col_null_count_list, 'nunique': col_unique_count_list,
                                'unique_values':
col_unique_list})
    else:
        df_info = pd.DataFrame({'column_name':
col_name_list, 'type': col_type_list,
                                'null_count':
col_null_count_list, 'nunique': col_unique_count_list})
    return df_info, df_total_memory_usage
```

After creating the function for the user so that the data analysis can be done for any other dataset.

Further passing the dataset in data frame as named as

```
df_1=df_breast_cancer_data, df_2=df_breast_cancer_data_new
```

Now assigning the dataframe to the different analysis points to display data.

```
df_breast_cancer_data_info,df_breast_cancer_data_mem =
get_df_info(df_breast_cancer_data,
include_unique_values=True)
print('{} has {} rows and {} cols, uses approx. {:.2f}
MB'.format('df_breast_cancer_data',
df_breast_cancer_data.shape[0],

df_breast_cancer_data.shape[1], df_breast_cancer_data_mem))
df_breast_cancer_data_info
```

Similarly, the values for other datasets and the data frame assigned will represent the output of the preceding code

Out[4]:

df_breast_cancer_data has 569 rows and 10 cols, uses approx. 0.04 MB					
	column_name	type	null_count	nunique	unique_values
0	clump_thickness	<class 'int'>	0	10	[5, 3, 6, 4, 8, 1, 2, 7, 10, 9]
1	uniformity_of_cell_size	<class 'int'>	0	10	[1, 4, 8, 10, 2, 3, 7, 5, 6, 9]
2	uniformity_of_cell_shape	<class 'int'>	0	10	[1, 4, 8, 10, 2, 3, 5, 6, 7, 9]
3	marginal_adhesion	<class 'int'>	0	10	[1, 5, 3, 8, 10, 4, 6, 2, 9, 7]
4	single_epithelial_cell_size	<class 'int'>	0	10	[2, 7, 3, 1, 6, 4, 5, 8, 10, 9]
5	bare_nuclei	<class 'int'>	0	11	[1, 10, 2, 4, 3, 9, 7, -100000, 5, 8, 6]
6	bland_chromatin	<class 'int'>	0	10	[3, 9, 1, 2, 4, 5, 7, 8, 6, 10]
7	normal_nucleoli	<class 'int'>	0	10	[1, 2, 7, 4, 5, 3, 10, 6, 9, 8]
8	mitosis	<class 'int'>	0	9	[1, 5, 4, 2, 3, 7, 10, 8, 6]
9	classes	<class 'int'>	0	2	[2, 4]

**Figure 6:** Screenshot of Data Frame for Dataset one



The figure 6 the data frame for the dataset one , which involves column names , the type of the dataset which is integer , the null count values , the unique values in the dataset and the number which are unique values and as observed the dataset one has 569 rows and 10 columns and the dataset one here using 0.04 MB of memory .

---

df\_breast\_cancer\_data\_new has 683 rows and 10 cols, uses approx. 0.30 MB

Out[7]:

	column_name	type	null_count	nunique	unique_values
0	thickness	<class 'int'>	0	10	[5, 3, 6, 4, 8, 1, 2, 7, 10, 9]
1	size	<class 'int'>	0	10	[1, 4, 8, 10, 2, 3, 7, 5, 6, 9]
2	shape	<class 'int'>	0	10	[1, 4, 8, 10, 2, 3, 5, 6, 7, 9]
3	adhesion	<class 'int'>	0	10	[1, 5, 3, 8, 10, 4, 6, 2, 9, 7]
4	single	<class 'int'>	0	10	[2, 7, 3, 1, 6, 4, 5, 8, 10, 9]
5	nuclei	<class 'int'>	0	10	[1, 10, 2, 4, 3, 9, 7, 5, 8, 6]
6	chromatin	<class 'int'>	0	10	[3, 9, 1, 2, 4, 5, 7, 8, 6, 10]
7	nucleoli	<class 'int'>	0	10	[1, 2, 7, 4, 5, 3, 10, 6, 9, 8]
8	mitosis	<class 'int'>	0	9	[1, 5, 4, 2, 3, 7, 10, 8, 6]
9	classes	<class 'int'>	0	2	[0, 1]

---

**Figure 7:** Screenshot of Data Frame for Dataset two.

In figure 7, there is more Exploratory Data analysis has done which is the id column has removed for getting the better clarification. However, it is the user choice how to perform the analysis with the dataset. In the second data frame as observed there are 683 rows and 10 columns, and the memory usage is approx. 0.30MB. Here, the exploratory data analysis part is shown and the sufficient understanding of the about both datasets. Now let us see if the user wants to see whether the datasets are matching or not.

### 3.5 Comparison of two similar or dissimilar datasets

The next step is to check whether the datasets are matching or not. According to the background research there are various methods for comparing the datasets. Here, giving choice to user if user wanted to check the datasets are similar or dissimilar .

```
def custom_comparator(df1, df2):
    df1 = df_breast_cancer_data
    df2 = df_breast_cancer_data_new
    matched = df1.equals(df2)
    print('Matches:', matched)
    return matched
custom_comparator(df1, df2)
```

Here, developed the function for the user so that user can compare any other datasets to check whether they are equal or not, using .equals() method will show whether the datasets are equal or not . The output for the preceding code follows as:

Matches: False

Out[10]: False

---

**Figure 8:** Screenshot of datasets equality.

The output after comparing two dissimilar datasets shows false it means they are unmatched. To get the output true for both datasets, it can be similar having same variables and values. To get result for each column whether our column values are matching or not then it will represent as , here created the function named comparator\_columns it will compare all the columns from each dataset using .eq() method to compare with dataset one with dataset two and for matching every columns.

```
def custom_comparator(df1, df2):
    df1 = df_breast_cancer_data
    df2 = df_breast_cancer_data_new
    matched_columns = df1.eq(df2)
    mt=matched_columns.all()
    return mt
```

The output of the preceding code is as follows:

```
Out[9]: adhesion           False
        bare_nuclei        False
        bland_chromatin     False
        chromatin           False
        classes             False
        clump_thickness      False
        marginal_adhesion    False
        mitosis              False
        normal_nucleoli      False
        nuclei               False
        nucleoli             False
        shape                False
        single                False
        single_epithelial_cell_size False
        size                 False
        thickness            False
        uniformity_of_cell_shape False
        uniformity_of_cell_size False
        dtype: bool
```

---

**Figure 9:** Screenshot of output for all the columns equality.

### 3.6 Visualizing the two different datasets

Now as understood that the datasets are not similar after comparing them and having the output. Now the tool will display the graphics using visualisation techniques for two datasets , first it will display the histograms it will show the frequency distribution for every columns which means how many times the values reoccur and then compare the frequency distribution with every column of one dataset with other dataset. Also, the distribution of data is symmetric if it has one peak in center and equal tails at the right and left. The mean and median of symmetric dataset are similar.

The function `def plot_histogram()` was developed, this function will plot the histogram for any two datasets .With the help of Matplotlib library and importing the library will give the interactive visualisation. Also, with matplotlib library using `%matplotlib inline` which sets the backend of matplotlib to the inline. It means with inline the frontend like Jupyter notebook display the output directly below the code one the code cell executes, and the resulting plot stored in notebook document.

```
import matplotlib.pyplot as plt
%matplotlib inline
```

Displaying the histogram distribution between two datasets, the code as follows

```
def plot_histogram(dfc_1 , dfc_2):
    dfc_1=df_breast_cancer_data.hist( color ='r',figsize =
    (12,9),bins = 10 )
    dfc_2=df_breast_cancer_data_new.hist( color ='g',alpha
    = 0.5,figsize = (12,9),bins =10 )
    plt.show()
    plt.subplots_adjust(bottom=0.50, left=0.10)
    plot_histogram(df_1,df_2)
```

The output of the preceding code as follows:

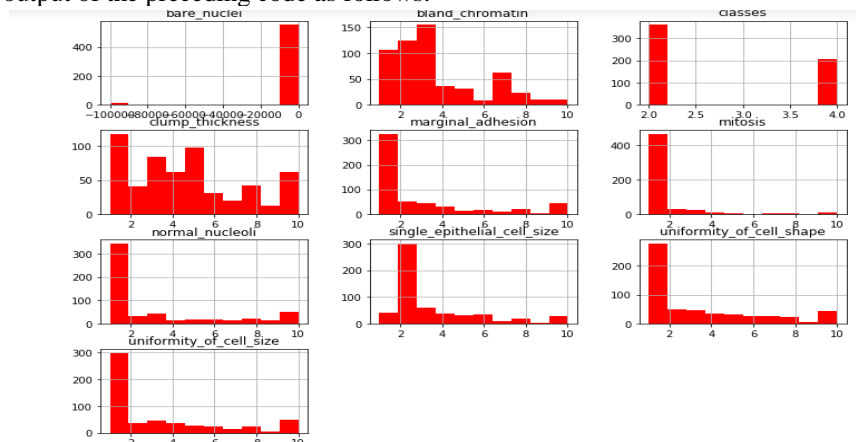
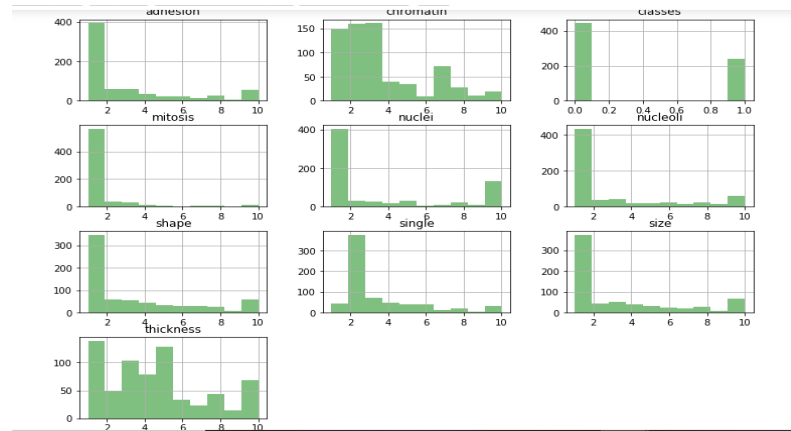


Figure 10: Screenshot of output histogram of dataset one



**Figure 11:** Screenshot of output of histogram of dataset two

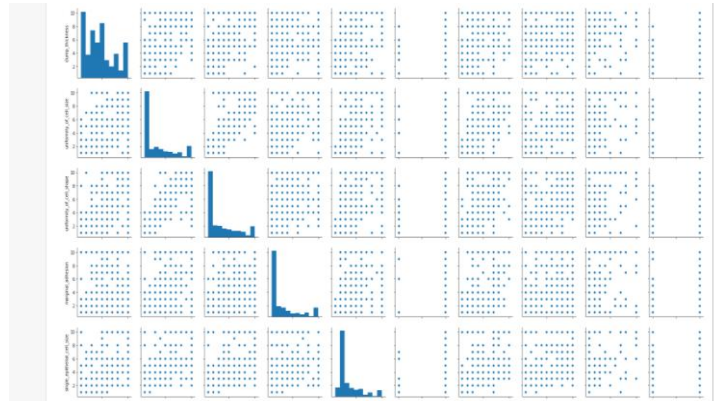
After getting the output of the histograms the user can easily compare the datasets visually and find the differences and learn the frequency distribution among each column and their range. For example, in the column name mitosis, mitosis describes the cell reproduction activity and the range is between one to ten. As observed that the cell productivity is higher in range one in both datasets. Similarly, user can compare other columns as well.

Now the second method of comparison here is Pair plot. A pair plot allows to see distribution of single variable and relationships between two variables. Here, using pair plot to see the relationship between the variables with each other in both datasets. Also using the seaborn library, seaborn library is generally use for whole dataset. Let us see the graphical representation of pair plot and the code.

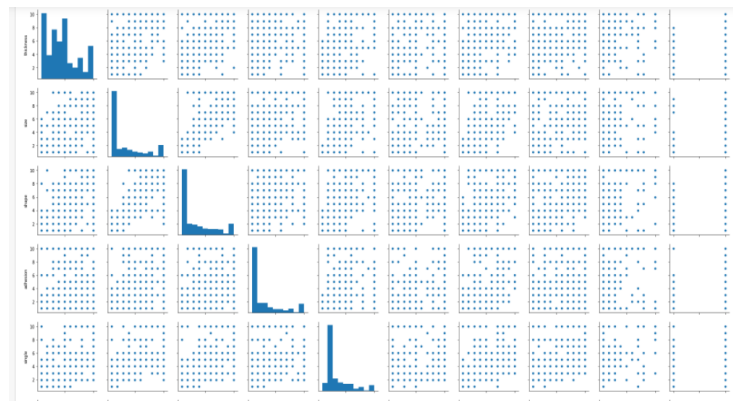
```
def plot_pairplot(df1, df2):
    df1=sns.pairplot(df_breast_cancer_data)
    df2=sns.pairplot(df_breast_cancer_data_new)
    plot_pairplot(df1, df2)
```

Here, passing the plot\_pairplot() function and then pass the parameters and assign the datasets inside seaborn library for pair plot and this will work for any other datasets . However, it is based on user whether he wanted to see the pair plot representation or not.

The output for the preceding code follows as :



**Figure 12:** Screenshot of output of Pair plot of dataset one from clump\_thickness column till single\_epithelial\_cell\_size

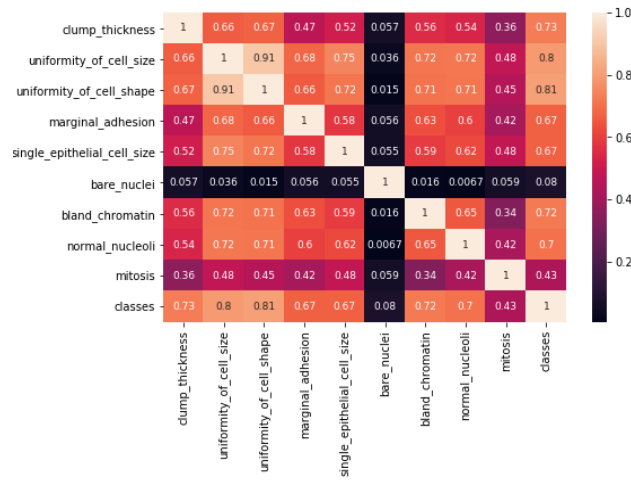


**Figure 13:** Screenshot of output of Pair plot of dataset two from thickness column till single column

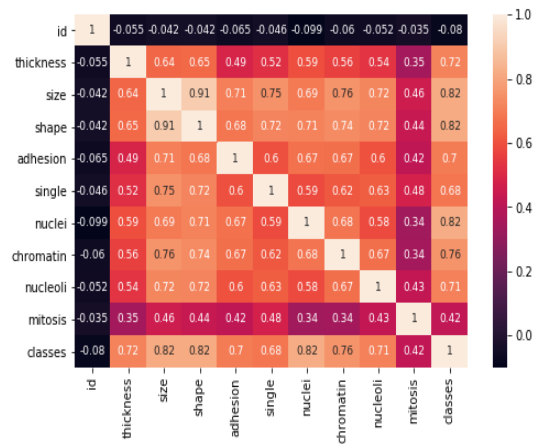
In the above graphical representations, as seen the relationship among all the variables with each other and their data representation for both datasets. As per the observation in dataset one if the uniformity of cell size increases then the uniformity of cell shape also increases, linear line is shown also the outliers among the line exist. And similar pattern in dataset second between columns name shape and size. Also, as observe in the first dataset, the values of classes column increases at malignant level and in the relationship between columns classes and mitosis there are more malignant cases with more cell productivity, also same structure occurred in second dataset between columns classes and mitosis.

Another visualisation method here for user's choice is heatmap, it is used to find the correlation among the variables within the dataset. And based on two datasets the user can compare the correlation between both datasets variables. For the heatmap we have use seaborn library.

Here passing the function `plot_heatmap()` will provide the heatmap for both datasets also it will provide the heatmap for any two datasets. Let us see the graphical representation for both the datasets.



**Figure 14:** Screenshot of output of Heatmap of dataset one



**Figure 15:** Screenshot of output of Heatmap of dataset two

In the above graphical representation of both dataset columns variables are correlated with each other in form of matrix , often in heatmap the color shade which is lighter has higher correlation than darker shade . Let us understand both heatmaps , in first dataset the class column it has higher correlation with the uniformity of shape and uniformity of size column , hence after observing them the shape and size of cell can tells us whether the breast cancer is in benign stage or malignant stage. Similarly, the second dataset and compare the column classes with shape and size, it is shows that in second dataset it gives same result, by the columns of both datasets can be compared.

Now the next implementation is regarding some more visualisation techniques. However here user will enter his choice for getting any visualisation to compare two datasets. To make more user interface here using this technique so that user can choose the option of graph and work with same. In this the options to choose with in three graphs which are boxplot, relplot, regplot are given. Let us see how it is working.

```

def boxplot(a,b):

    plt.figure()
    xx=sns.boxplot(x='classes',y= 'mitosis',data=a)#for
any two column depends on user for dataset 1
    plt.figure()
    yy=sns.boxplot(x='classes',y= 'mitosis',data=b)#for
any two column depends on user for dataset 2
plt.show()


def relplot(a,b):
    aa=sns.relplot(x = 'single_epithelial_cell_size' , y =
'marginal_adhesion',col = 'mitosis' , hue = 'classes' ,col_wrap =
3,data =a)
    bb=sns.relplot(x = 'single' , y = 'adhesion',col = 'mitosis'
, hue = 'classes' ,col_wrap = 3,data =b)
print(aa)
print(bb)


def regplot(p,q):
    plt.figure()
    pp=sns.regplot(x = 'clump_thickness' , y = 'mitosis', data =
p)
    plt.figure()
    qq=sns.regplot(x = 'thickness' , y = 'mitosis', data = q)
    plt.show()


print("Select operation.")
print("1.Boxplot")
print("2.relplot")
print("3.regplot")


while True :
    print("Enter values and 0 to exit")

    val = input()

    if val == "1":
        print("this chart")
        boxplot(df_1,df_2)

    elif val == "2":
        print("this chart")
        relplot(df_1,df_2)

    elif val == "3":
        print("this chart")
        regplot(df_1,df_2)
    else:
        break

```

In this code as observed there are three functions created which are for box plot, regplot and relplot and these functions will work for any other datasets which user wanted to compare.

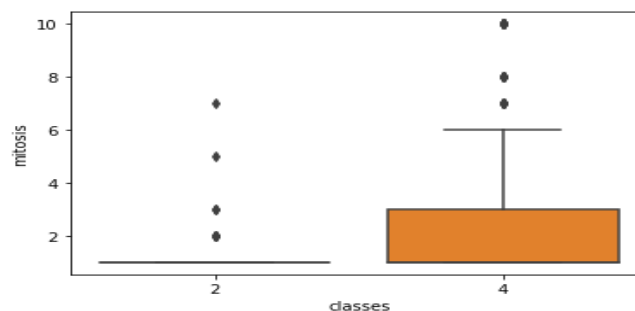
If user press key one, then user will get the graphical representation of boxplot for any columns here the given column names classes and mitosis for both datasets . Next if user press key number two then the representation of relplot between x axes and y axes is shown which are single\_epithelial\_cell\_size and marginal\_adhesion and column name is mitosis for first dataset and for second dataset the x and y axes are single and adhesion and column name is mitosis also the hue parameter determines which column in data frame should be use for color encoding and here chosen hue as a classes column .

If press number key three, the graphical representation for regplot occurred , the regplot defines the relationship between two variables in data frame .Here we have provided the variables name clump\_thickness and mitosis for first dataset and thickness and mitosis for second datasets . However, user can compare the representation between two data frames.

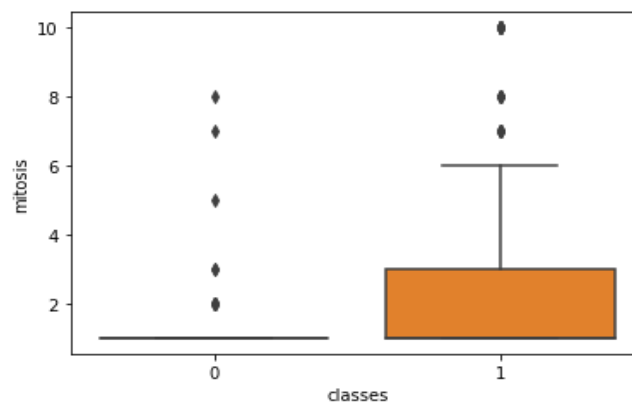
Let us see the output of the preceding code below.

If user enter the key values one the graphical representation will be:

```
Select operation.
1.Boxplot
2.relplot
3.regplot
Enter values and 0 to exit
1
this chart
```



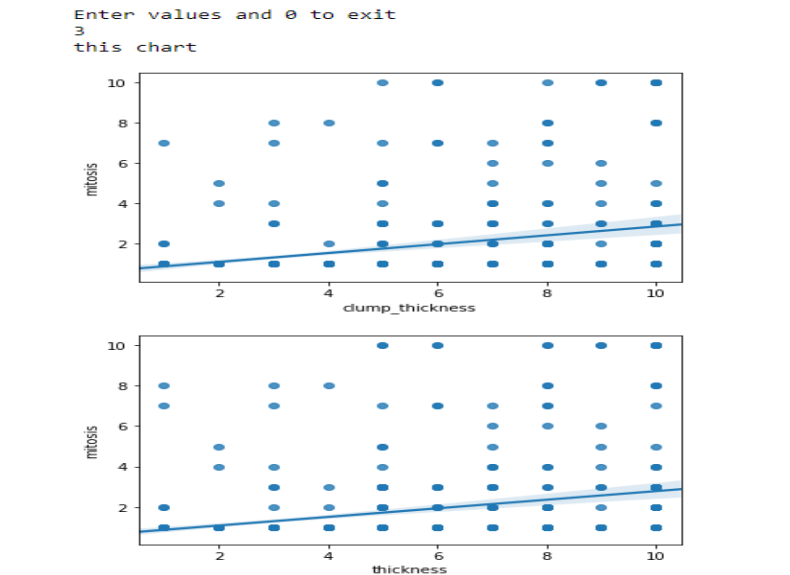
**Figure 16:** Screenshot of output of boxplot for dataset one.



**Figure 17:** Screenshot of output of boxplot for dataset two.



If the user enters the key three then graphical representation will be follows as:



**Figure 18:** Screenshot of regplot for both datasets.

Similarly, if the user press key two then the output will be relplot and if user enters key zero it will end the graphical representation. The visualisation methods end here.

### 3.7 Statistical Hypothesis assumptions testing in research

Before making the assumptions lets calculate the basic statistic values which includes mean, median, standard deviation, and other statistics. This can simply developed by developing function `describe_statistics()` for any two datasets and for calculating the statistics we are using `df_breast_cancer_data.describe()` which means name of the dataset `.describe()` method , this function is used to get basic statistics for any two datasets .Let's see how it is implemented :

```
def describe_statistics(stats1,stats2):
    stats1=df_breast_cancer_data.describe()
    stats2=df_breast_cancer_data_new.describe()
    print(stats1)
    print(stats2)
describe_statistics(df_1,df_2)
```

The output of the preceding code follows as :

	clump_thickness	uniformity_of_cell_size	uniformity_of_cell_shape
count	569.000000	569.000000	569.000000
mean	4.539543	3.184534	3.265378
std	2.896501	3.002236	2.955935
min	1.000000	1.000000	1.000000
25%	2.000000	1.000000	1.000000
50%	4.000000	1.000000	2.000000
75%	6.000000	5.000000	5.000000
max	10.000000	10.000000	10.000000

	marginal_adhesion	single_epithelial_cell_size	bare_nuclei \
count	569.000000	569.000000	569.000000
mean	2.845343	3.298770	-2632.518453
std	2.873626	2.304775	16035.653408
min	1.000000	1.000000	-100000.000000
25%	1.000000	2.000000	1.000000
50%	1.000000	2.000000	1.000000
75%	4.000000	4.000000	8.000000
max	10.000000	10.000000	10.000000

	bland_chromatin	normal_nucleoli	mitosis	classes
count	569.000000	569.000000	569.000000	569.000000
mean	3.490334	2.989455	1.637961	2.731107
std	2.324925	3.091315	1.773941	0.964018
min	1.000000	1.000000	1.000000	2.000000
25%	2.000000	1.000000	1.000000	2.000000
50%	3.000000	1.000000	1.000000	2.000000
75%	5.000000	4.000000	1.000000	4.000000
max	10.000000	10.000000	10.000000	4.000000

Figure 19: Screenshot of statistic values for datasets one .

	thickness	size	shape	adhesion	single	nuclei
count	683.000000	683.000000	683.000000	683.000000	683.000000	683.000000
mean	4.442167	3.150805	3.215227	2.830161	3.234261	3.544656
std	2.820761	3.065145	2.988581	2.864562	2.223085	3.643857
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	2.000000	1.000000	1.000000	1.000000	2.000000	1.000000
50%	4.000000	1.000000	1.000000	1.000000	2.000000	1.000000
75%	6.000000	5.000000	5.000000	4.000000	4.000000	6.000000
max	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000

	chromatin	nucleoli	mitosis	classes
count	683.000000	683.000000	683.000000	683.000000
mean	3.445095	2.869693	1.603221	0.349927
std	2.449697	3.052666	1.732674	0.477296
min	1.000000	1.000000	1.000000	0.000000
25%	2.000000	1.000000	1.000000	0.000000
50%	3.000000	1.000000	1.000000	0.000000
75%	5.000000	4.000000	1.000000	1.000000
max	10.000000	10.000000	10.000000	1.000000

Figure 20: Screenshot of statistic values for datasets two.

After getting the statistic values for all the columns of both datasets, these values can be used to understand the datasets.

**Normal distribution:** Now the next step to check the normal distribution within the datasets. The normal distribution has another essential place in statistics [33]. It is also called as gaussian distribution. Why calculating normal distribution is important? Because normality does not simply imply normality. Various analysis methods make assumptions about normality such as t tests, correlation and more .The assumptions cannot be made by observing the data about normality, but sample values will be collected and it should be compatible with the population means representation . Here using one of the methods to check the normal distribution also this will also include other methods like skewness and kurtosis for the variable. Let us see how it is happening for both datasets.

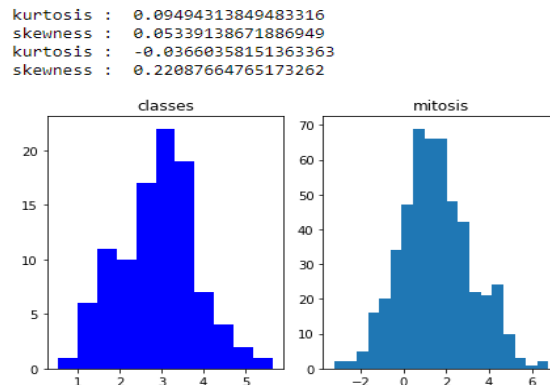
First, import the packages for calculating the skewness and kurtosis of datasets variables. We will import from the scipy.stats library .

```
from scipy.stats import skew
from scipy.stats import kurtosis
```

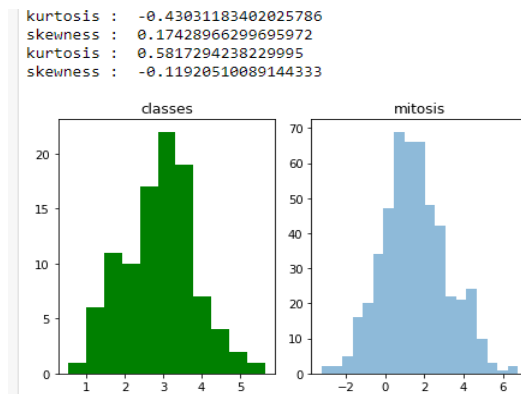
after this using `np.random.normal()` and defining the value of mean ( $\mu$ ) and standard deviation ( $\sigma$ ) and within the samples from population the normal distribution will be represented and here the normal distribution has been implementing for classes variable and mitosis variable in both datasets , if user wants to calculate other variables of datasets it can be replace by the column name and the values can be replaced by observing from basic statistic output . Also, for calculating the skewness the method as `skew(variable name)` will be used and similarly for kurtosis as `kurtosis(variable name)` can be used .It can be applicable for any other variable of datasets. However, using the same technique for another dataset.

```
val_class =np.random.normal(2.7,0.96, 100)
val_mitosis =np.random.normal(1.6,1.7,500)
f, (ax1, ax2) = plt.subplots(1, 2)
ax1.hist(val_class, bins='auto' , color = 'b')
ax1.set_title('classes')
ax2.hist(val_mitosis, bins = 'auto' )
ax2.set_title('mitosis')
plt.tight_layout()
print("kurtosis : ",kurtosis(val_class))
print("skewness : ",skew(val_class))
print("kurtosis : ",kurtosis(val_mitosis))
print("skewness : ",skew(val_mitosis))
```

**The output for preceding code as follows :**



**Figure 21:** Screenshot of output for calculating normal distribution for dataset one for columns classes and mitosis.



**Figure 22:** Screenshot of output for calculating normal distribution for dataset two for columns classes and mitosis.

After observing both the graphs of both datasets the right skewed graph could be one of the reasons for non-normality. Further now let us see the assumptions based on statistical tests.

Here, the aim is to run the parametric and non-parametric tests based on datasets distribution.

In hypotheses testing experiments, research hypothesis is tested by negotiating the null hypothesis [25]. Here our focus is to test whether the null hypotheses can be rejected on the basis of given dataset. There are several parametric and non-parametric tests available for hypotheses testing experiment. The assumptions are obtained from population having normal distribution. Each test has its own assumption but if any assumptions violated then non-parametric test should be used. However, assuming with parametric as well as non-parametric because larger number of samples are available. Now testing between two groups.

**T-test:** As, explained earlier T-test measures the means of two groups. Here comparing the mean of two datasets and their variables. The user can simply do it with other datasets for the desired variables.

Firstly, the stats package from scipy library and one-sample t-test package is imported.

```

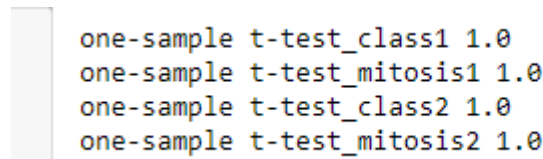
from scipy import stats
from scipy.stats import ttest_1samp

```

Now the test step is to pass function, here the function `onesample_ttest()` is developed and using with method `ttest_1samp(df_1["classes"], mean_class1)`. The one sample t test values for both datasets can be calculated and used for variable classes and mitosis for both datasets.

```
def onesample_ttest(t_statistic , p_value):
mean_class1 =df_1["classes"].mean()
mean_mitosis1=df_1["mitosis"].mean()
t_statistic , p_value = ttest_1samp(df_1["classes"] ,mean_class1)
t_statistic , p_value = ttest_1samp(df_1["mitosis"]
,mean_mitosis1)
print (" one-sample t-test_class1" , p_value)
print (" one-sample t-test_mitosis1" , p_value)
onesample_ttest(df_1 , df_2)
```

The output for preceding code as follows:



```
one-sample t-test_class1 1.0
one-sample t-test_mitosis1 1.0
one-sample t-test_class2 1.0
one-sample t-test_mitosis2 1.0
```

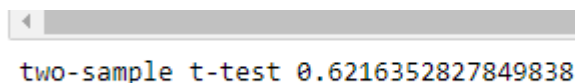
**Figure 23:** Screenshot of output for calculating one-sample t test for both datasets for columns classes and mitosis .

Here as observed that the p-value for all variables is 1.0, since the p-value is greater than 0.05 indicates weak evidence to reject the null hypothesis.

**Two sample t test:** This test is for comparing differences between two separate groups or populations [25]. This test also shows the differences in naturally occurring groups. Here ,comparing mitosis variable from dataset one and mitosis variable from dataset 2 , the two-sample t test can be used to test the significance difference [25]. Let us see the implementation.

```
def twosample_ttest(t_statistic , p_value):
t_statistic , p_value = ttest_ind(df_1["mitosis"]
,df_2["mitosis"])
print("two-sample t-test" , p_value)
twosample_ttest(df_1 ,df_2)
```

The ttest\_ind()method is used to calculate the difference .  
The output for the preceding code is :



```
two-sample t-test 0.6216352827849838
```

**Figure 24:** Screenshot of output for calculating two-sample t test for both datasets for columns mitosis .

Here as observed that the p-value  $> 0.05$ , hence it can be assuming that the value of both variables in both datasets differs also the assumptions are based on normality. As far the parametric tests have executed, now let us see the assumption using nonparametric test.

The non-parametric tests are often suitable for large datasets and here the large datasets are used. Hence nonparametric tests have more weightage than parametric test [25].

**Shapiro test:** In shapiro-wilk test if p- value is less than 0.05, then the null hypothesis would be violated. Here the variables of both datasets are used.

```
def shapiro_test(shapiro_class1, shapiro_class2):
    shapiro_class1 = scipy.stats.shapiro(df_1["classes"])
    shapiro_class2 = scipy.stats.shapiro(df_2["classes"])
    shapiro_mitosis1 = scipy.stats.shapiro(df_1["mitosis"])
    shapiro_mitosis2 = scipy.stats.shapiro(df_2["mitosis"])
    print(shapiro_class1)
    print(shapiro_class2)
    print(shapiro_mitosis1)
    print(shapiro_mitosis2)
shapiro_test(df_1, df_2)
```

using the `scipy.stats.shapiro()` method, user can generate the p- values for variables of both datasets and compare them. The output for the preceding code follows as:

```
(0.6097162961959839, 7.217476539331304e-34)
(0.6005393862724304, 3.1774509940891515e-37)
(0.4132397770881653, 2.7812817777087893e-39)
(0.39283454418182373, 4.091791515828466e-43)
```

---

**Figure 25:** Screenshot of output for calculating shapiro-wilk test for both datasets for columns mitosis and classes.

Here for every column as observed that the p-value is greater than 0.05, it means the data is normal.

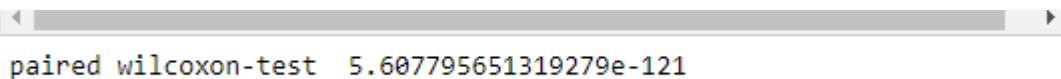
Now the next assumptions will be from Wilcoxon test.

**Wilcoxon test:** This test is alternative test of paired sample t test to test the statistical differences in mean between two random samples of datasets [25].

The paired samples are generated from either same participants or with similar characteristics [25]. Let's see the code,

```
def wilcoxon_test(z_statistic, p_value):
    z_statistic, p_value = wilcoxon(df_1["classes"] -
    df_2["classes"])
    print("paired wilcoxon-test ", p_value)
wilcoxon_test(df_1, df_2)
```

Here the Wilcoxon () method have been passed to run the test. The output is:



```
paired wilcoxon-test 5.607795651319279e-121
```

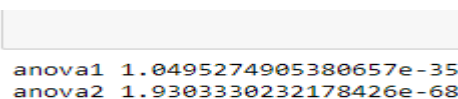
**Figure 26:** Screenshot of output for calculating Wilcoxon test for both datasets for columns classes.

Here as observed the p value is greater than 0.05, hence the assumption is violated. For an alternative way anova test can be executed.

**Anova test:** It is used to compare three or more unpaired data groups [26]. It will compare the means of the groups in which user is interested and check whether any of the mean is significantly different from each other. The implementation for comparing the mean of both datasets for the column classes and mitosis is displayed below.

```
def anova_test(F,F1):
    F,p=stats.f_oneway(df_1['classes'], df_1['mitosis'])
    F1,p_value=stats.f_oneway(df_2['classes'], df_2['mitosis'])
    print('anova1' , p)
    print('anova2' , p_value)
anova_test(df_1,df_2)
```

The output for the preceding code as follows:



```
anova1 1.0495274905380657e-35
anova2 1.9303330232178426e-68
```

**Figure 27:** Screenshot of output for calculating one-way anova test for both datasets for columns classes and mitosis.

Also, the assumptions test should be checked, for f test, import the ols model and also use sm.stats.anova method for generate anova table also the “typ” in anova\_lm method determines how the sum of the squared is calculated.

```
model = ols('classes ~ mitosis', data = df_1).fit()
aov_table = sm.stats.anova_lm(model, typ=2)
```

The output for the preceding code will be:

	sum_sq	df	F	PR(>F)
mitosis	98.975706	1.0	130.849519	2.087067e-27
Residual	428.883696	567.0	NaN	NaN
	sum_sq	df	F	PR(>F)
mitosis	28.277149	1.0	152.040239	9.682098e-32
Residual	129.631292	697.0	NaN	NaN

**Figure 28:** Screenshot of output for calculating anova table for both datasets for columns classes and mitosis

After observing the anova table, there is statistically significant difference between the groups with  $F = 130.84$  and The value in the table above seem to be  $2.08 \times 10^{-27}$  for mitosis for dataset one and for second dataset  $F = 152.040$  and The value in the table above seem to be  $9.68 \times 10^{-27}$ . To find which groups different significantly, the post-hoc test will be conducted.

**Post-Hoc Testing:** There are few approaches to conduct these tests, one of the approaches is Tukey honestly significant difference (HSD). It tests all pairwise group comparisons. Let us see the implementation.

Here importing the pairwise\_tukeyhsd library and multicomparison.

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from statsmodels.stats.multicomp import MultiComparison
def tukeyhsd_test(mc1,mc2):
    mc1 = MultiComparison(df_1['classes'], df_1['mitosis'])
    mc2 = MultiComparison(df_2['classes'], df_2['mitosis'])
    result1 = mc1.tukeyhsd()
    result2 = mc2.tukeyhsd()
    print (result1)
    print (result2)
    print(mc1.groupsunique)
    print(mc2.groupsunique)
tukeyhsd_test(df_1,df_2)
```

The output of the preceding code , due to large output, hence providing the sample .

Multiple Comparison of Means - Tukey HSD, FWER=0.05						
group1	group2	meandiff	p-adj	lower	upper	reject
1	2	1.0959	0.001	0.6066	1.5852	True
1	3	1.3615	0.001	0.8554	1.8676	True
1	4	1.5097	0.001	0.7622	2.2572	True
1	5	1.1763	0.0153	0.1259	2.2268	True
1	6	1.5097	0.0418	0.0289	2.9905	True
1	7	1.2597	0.001	0.348	2.1713	True
1	8	1.5097	0.001	0.5361	2.4832	True
1	10	1.5097	0.001	0.7622	2.2572	True
2	3	0.2656	0.9	-0.4181	0.9494	False
2	4	0.4138	0.8587	-0.4637	1.2913	False
2	5	0.0805	0.9	-1.0662	1.2271	False
2	6	0.4138	0.9	-1.1367	1.9643	False
2	7	0.1638	0.9	-0.8572	1.1848	False
2	8	0.4138	0.9	-0.6628	1.4904	False
2	10	0.4138	0.8587	-0.4637	1.2913	False
3	4	0.1481	0.9	-0.7389	1.0352	False
3	5	-0.1852	0.9	-1.3391	0.9687	False
3	6	0.1481	0.9	-1.4078	1.7041	False
3	7	-0.1019	0.9	-1.131	0.9273	False
3	8	0.1481	0.9	-0.9362	1.2325	False
3	10	0.1481	0.9	-0.7389	1.0352	False
4	5	-0.3333	0.9	-1.6116	0.945	False
4	6	0.0	0.9	-1.6503	1.6503	False
4	7	-0.25	0.9	-1.4169	0.9169	False
4	8	0.0	0.9	-1.2159	1.2159	False
4	10	0.0	0.9	-1.0437	1.0437	False
5	6	0.3333	0.9	-1.4745	2.1411	False
5	7	0.0833	0.9	-1.2974	1.4641	False

**Figure 29:** Screenshot of output for calculating Tuckey hsd test for both datasets for columns classes and mitosis

To make sense of the table, at the top the testing information is provided.

FWER is a family wise error rate, it means what is the value of  $\alpha$  and at what level it has been set .Groups 1 and groups 2 columns are the group which are compared .meandiff is the difference in mean .p-adj is the p-value .lower is lower band and upper is upper band of confidence interval at 95% level since  $\alpha=0.05$ .Reject is the decision based on p-value . Using the TukeyHSD to test for differences between groups for both datasets, there statistically significantly difference classes and mitosis mean and observe the number of the cell productivity with in two data groups.



## 4 Evaluation and test cases

For this section, the implementation has been performed with the other datasets, to check the performance and to check the user interaction with it.

First step in this we have collected the datasets, as mentioned earlier dataset can be user choice , Hence, collected the dataset based on user knowledge modelling dataset from UCI[34].Also the original data has been split into two sets to do the evaluation based on UNS categories .It means the dataset has been split into two sets in which one dataset is high performers and other is low performers. The data analysis based on these categories. After collecting the dataset in CSV file, again we perform same steps.

- Loaded the dataset and convert it into data frame and the file can be read from any location of the local system. once the dataset has been converted into data frame.
- The next step is to perform exploratory data analysis, based on user choice the tool will perform certain steps such as handling missing value, null values , the unique numbers , the type of the dataset . Let us see the output data pre-processing section :

df\_student\_data has 224 rows and 6 cols, uses approx. 0.01 MB

```
it[11]:
```

	column_name	type	null_count	nunique	unique_values
0	STG	<class 'numpy.float64'>	0	84	[0.08, 0.1, 0.09, 0.0, 0.18, 0.2, 0.12, 0.15, ...
1	SCG	<class 'numpy.float64'>	0	81	[0.08, 0.1, 0.15, 0.0, 0.18, 0.2, 0.12, 0.29, ...
2	STR	<class 'numpy.float64'>	0	83	[0.1, 0.15, 0.4, 0.43, 0.5, 0.55, 0.52, 0.7, 0...]
3	LPR	<class 'numpy.float64'>	0	81	[0.24, 0.65, 0.1, 0.29, 0.2, 0.3, 0.78, 0.15, ...
4	PEG	<class 'numpy.float64'>	0	62	[0.9, 0.3, 0.66, 0.56, 0.85, 0.81, 0.34, 0.6, ...
5	UNS	<class 'str'>	0	2	[High, Middle]

**Figure 30:** Screenshot of output for user modelling data frame one

df\_student\_data\_new has 179 rows and 6 cols, uses approx. 0.01 MB

```
it[13]:
```

	column_name	type	null_count	nunique	unique_values
0	STG	<class 'numpy.float64'>	0	69	[0.0, 0.06, 0.08, 0.15, 0.2, 0.05, 0.1, 0.12, ...
1	SCG	<class 'numpy.float64'>	0	68	[0.0, 0.06, 0.08, 0.02, 0.14, 0.07, 0.25, 0.32...]
2	STR	<class 'numpy.float64'>	0	63	[0.0, 0.05, 0.08, 0.34, 0.35, 0.51, 0.7, 0.1, ...]
3	LPR	<class 'numpy.float64'>	0	59	[0.0, 0.25, 0.98, 0.4, 0.72, 0.41, 0.01, 0.08,...]
4	PEG	<class 'numpy.float64'>	0	38	[0.0, 0.33, 0.24, 0.01, 0.25, 0.3, 0.05, 0.29,...]
5	UNS	<class 'str'>	0	2	[Very_low, Low]

**Figure 31:** Screenshot of output for user modelling data frame two

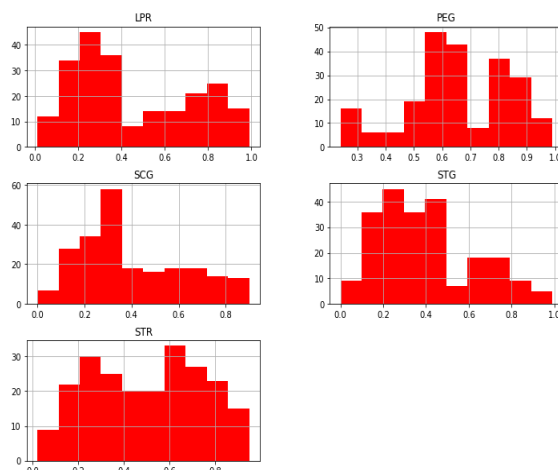
With this part the type of data frame can be understood and with the help of this various other implementations can be performed.

- After the data frame has been generated then next step is comparison of both datasets as used earlier, here using the same method for compare both datasets and checking whether they are matching or not. Using the same function for these two other datasets, the datasets are not matching with each other hence they are giving false output.
- Also, if user wanted to see which columns are not matching, it can evaluate using the same method as used earlier. And the new output is.

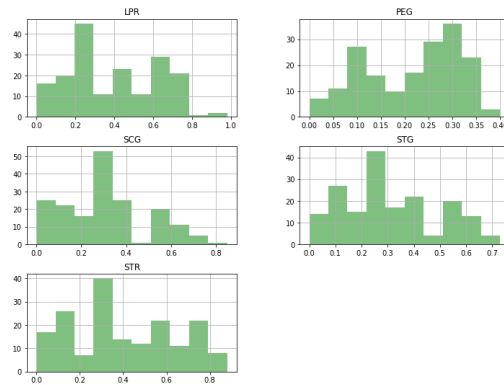
```
Out[15]: STG      False
         SCG      False
         STR      False
         LPR      False
         PEG      False
         UNS      False
         dtype: bool
```

**Figure 32:** Screenshot of output for user modelling unmatched columns.

- After the comparison section has executed, if user wanted to visualize the graphical representation, user can have a choice regarding graphs , let is see if user wanted to visualize histogram .

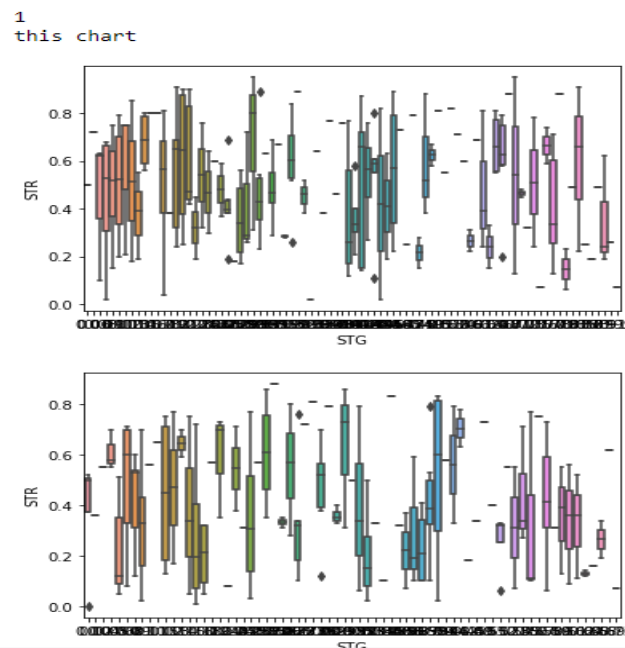


**Figure 33:** Screenshot of output for user modelling histogram for data frame one



**Figure 34:** Screenshot of output for user modelling histogram for data frame one

- Similarly, if user wanted to observe boxplot, he can simply choose the boxplot options in the user-based tool. The comparison is between STR which means the degree of study time for goal object materials and STG which means the degree of study time of user for related object with goal object. In both boxplots there is the difference between high performers and low performers. The results can be interpreted based on quartiles.



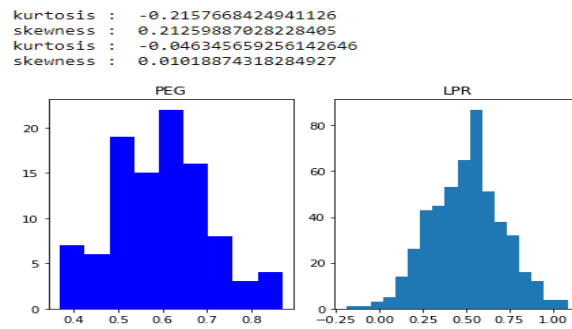
**Figure 35:** Screenshot of output for user modelling boxplot for both data frame

- Now the next comparison is to evaluate the basic statistics, basic statistics can provide the values for mean, standard deviations and more values. With the help of basic statistics the normality can be assumed.

	STG	SCG	STR	LPR	PEG
count	224.000000	224.000000	224.000000	224.000000	224.000000
mean	0.389295	0.396018	0.499621	0.457277	0.653616
std	0.227391	0.226748	0.241228	0.274310	0.180837
min	0.000000	0.000000	0.020000	0.010000	0.240000
25%	0.207500	0.245000	0.290000	0.240000	0.557500
50%	0.320000	0.320000	0.520000	0.330000	0.650000
75%	0.495000	0.592500	0.702500	0.742500	0.800000
max	0.990000	0.900000	0.950000	0.990000	0.990000
	STG	SCG	STR	LPR	PEG
count	179.000000	179.000000	179.000000	179.000000	179.000000
mean	0.307899	0.305788	0.405140	0.398888	0.209514
std	0.181763	0.189601	0.244008	0.231572	0.097571
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.160000	0.145000	0.185000	0.250000	0.115000
50%	0.280000	0.288000	0.350000	0.360000	0.240000
75%	0.405000	0.400000	0.585000	0.605000	0.300000
max	0.730000	0.850000	0.880000	0.980000	0.400000

**Figure 36:** Screenshot of output for user modelling basic statistics for both data frame

- Checking the normal distribution for both data sets.

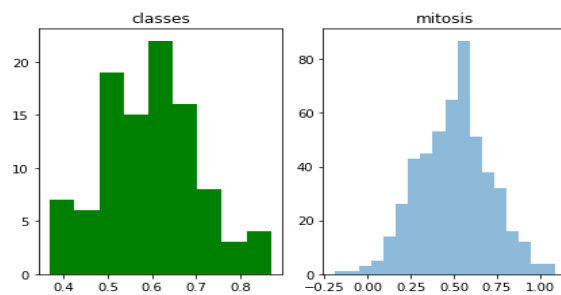


**Figure 37:** Screenshot of output for normal distribution of dataset one

```

kurtosis : -0.4613268256376508
skewness : -0.10362293372203948
kurtosis : -0.2732216297307106
skewness : -0.08035465659166312

```



**Figure 38:** Screenshot of output for normal distribution of dataset two

- After performing the normal distribution step user can run the statistical hypothesis test. This is also based on user choice , for example if it wanted to run the shapiro wilk test , he can simple choose to run the one sample ttest and conclude about the normality the user can easily choose the fuction in this two datasets the p-value =0.9 for all used columns , hence null hypothesis will not be rejected .Similary with the help of this tool user can perform other tests also .

## 5 Conclusion

In this dissertation there is a implementation of the user based tool. This tool has been developed for user to get more results regarding data which includes data analysis, data visualizations and with respect to that the user can also perform statistical assumptions based on normality of both datasets.

Starting with the tool has able to perform data collection and converting into data frame, this tool will help user to convert any dataset which are in csv file format into data frame. User can also perform exploratory data analysis dynamically. After implementing EDA user can simply perform the graphical representation according to his choice. The graphical representations can be any graph type such as histogram , heatmap, pair plot , box plot , relplot and reg plot .All the graphs will help user to have good understanding of datasets .This tool performed the comparison between two similar or dissimilar datasets and provided the output . With respect to graphical representation the tool will also perform normal distribution of both datasets and show skewness and kurtosis values of variables. The tool also helps in perform normality test which are one sample t test , shapirotest , anova test , Wilcoxon test . Also, it shows anova table and perform multicompairs test between two variables of both datasets.

This tool also performs assumptions regarding null hypothesis and in some cases the null hypothesis has rejected or its violated. This will help use to understand the statistically significance regarding normality and data distribution.

In the evaluation section the comparisons have done with other datasets to check the functionality. However in some cases this user based tool can only handle numerical datasets value not categorical for this the datasets has to be numerical .Also the data type was also different it means it can handle numerical based datatype such as integer of float .However the results are based on numerical datasets values. The data visualisation performs well with other two datasets. Also, the statistical modelling steps were similar with earlier datasets. The other two datasets were also not normally distributed. Therefore based on aim and objective the tool has successfully perform the visualisation which means it shows the graphical representations and statistically modelling which means it performed the required test.

## 6 Future Scope

The future work is to implement the GUI based tool. Also, the tool will handle categorical datasets and perform visualisation of categorical datasets. Also, the tool performs the other statistic test which are suitable from categorical values. It will also increase more data analysis performance.

## **Acknowledgement**

I would like to thank my supervisor Mrs. Sara Fernstand for her guidance from the very beginning of this dissertation project.

## References

1. The Benefits of Statistical Visualization in an Immersive Environment. Laura Arns<sup>1</sup>, Dianne Cook<sup>2</sup>, Carolina Cruz-Neira<sup>1</sup> <sup>1</sup>Iowa Centre for Emerging Manufacturing Technology Iowa State University, Ames IA 50011-1210 <sup>2</sup>Department of Statistics Iowa State University, Ames IA 50011-1210.
2. Visualizing Summary Statistics and Uncertainty K. Potter<sup>1</sup>, J. Kniss<sup>2</sup>, R. Riesenfeld<sup>3</sup>, and C.R. Johnson<sup>1</sup> <sup>1</sup>Scientific Computing and Imaging Institute, University of Utah <sup>2</sup>Department of Computer Science, University of New Mexico <sup>3</sup>School of Computing, University of Utah.
3. Are assumptions of well-known statistical techniques checked, and why (not)? Rink Hoekstra<sup>1,2\*</sup>, Henk A. L. Kiers<sup>2</sup> and Addie Johnson GION –Institute for Educational Research, University of Groningen, Groningen, The Netherlands <sup>1</sup>Department of Psychology, University of Groningen, Groningen, The Netherlands.
4. Anna Bartkowiak and Adam Szustalewicz. Some modern techniques for viewing multivariate data - a comparative look. In M. A. Kłopotek and M. Michalewicz, editors, Workshop on Intelligent Information Systems VIII, Ustronie, Poland, pages 7–11, June 14–18 1999
5. Python Data Analytics Fabio Nelli Rome, Italy ISBN-13 (pbk): 978-1-4842-3912-4 ISBN-13 (electronic): 978-1-4842-3913-1 <https://doi.org/10.1007/978-1-4842-3913-1>
6. Döbler, Mario Großmann, Tim. (2019). *Data Visualization with Python*. Packt Publishing. Retrieved from <https://app.knovel.com/hotlink/toc/id:kpDVP00001/data-visualization-with/data-visualization-with>.
7. Ensemble-Vis: A Framework for the Statistical Visualization of Ensemble Data.
8. Kristin Potter\*, Andrew Wilson†, Peer-Timo Bremer‡, Dean Williams‡, Charles Doutriaux‡, Valerio Pascucci\*, and Chris R. Johnson\* \*Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, Utah 84112 †Sandia National Laboratories, Albuquerque, New Mexico, 87185 ‡Lawrence Livermore National Laboratory, Livermore, California, 94550 Email: {kpotter,pascucci,crj}@sci.utah.edu, atwilso@sandia.gov, {bremer5,williams13,doutriaux1}@llnl.gov.
9. WiggleTools: parallel processing of large collections of genome-wide datasets for visualization and statistical analysis.
10. Statistical Methods for Comparison of Data Sets of Construction Methods and Building Evaluation Hamed Niroumanda\*, M.F.M Zainb, Maslina Jamilc a,b,cDepartment of architecture, Faculty of built environment and engineering, National University of Malaysia, Malaysia.
11. Whatagraph:<https://whatagraph.com/blog/articles/data-visualization-techniques/> Accessed 28/08/2020.
12. Towards data science ,<https://towardsdatascience.com/top-16-types-of-chart-in-data-visualization-196a76b54b62/> Accessed 28/08/2020.
13. Visme :<https://visme.co/blog/types-of-graphs/> Accessed 28/08/2020.
14. An overview and comparison of free Python libraries for data mining and big data analysis I. Stančin\* and A. Jović \* \* University of Zagreb Faculty of Electrical Engineering and Computing / Department of Electronics, Microelectronics, Computer and Intelligent Systems, Unska 3, 10 000 Zagreb, Croatia stancin.igor@gmail.com, alan.jovic@fer.hr
15. Towards data Science :<https://towardsdatascience.com/python-vs-r-for-data-science-6a83e4541000/> Accessed 28/08/2020.
16. T. E. Oliphant, A guide to NumPy, USA: Trelgol Publishing, 2006.
17. W. McKinney, pandas: a Foundational Python Library for Data Analysis and Statistics. Python High Performance Science Computer, 2011.
18. An introduction to Seaborn : <https://seaborn.pydata.org/introduction.html> / Accessed 28/08/2020.
19. Scipy.org :<https://docs.scipy.org/doc/scipy/reference/tutorial/general.html> / Accessed 28/08/2020.
20. Towards Data Science ,<https://towardsdatascience.com/how-to-quickly-compare-data-sets-76a694f6868a/> Accessed 28/08/2020

21. Medium :<https://medium.com/financeexplained/3-quick-ways-to-compare-data-in-python-65201be10b6> / Accessed 28/08/2020.
22. Methods for Presenting Statistical Information: The Box Plot Kristin Potter University of Utah School of Computing Salt Lake City, UT [kpotter@cs.utah.edu](mailto:kpotter@cs.utah.edu).
23. Generalized scatter plots Daniel A. Keima Ming C. Haob Umeshwar Dayalb Halldor Janetzkoa and Peter Baka,\* aUniversity of Konstanz, Universitaetsstr. 10, Konstanz, Germany. bHewlett Packard Research Labs, 1501 Page Mill Road, Palo Alto, CA94304, USA. \*Corresponding author.
24. A. Jain and R. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice-Hall, 1998.
25. Testing Statistical Assumptions in Research J. P. Verma Lakshmibai National Institute of Physical Education Gwalior, India Abdel-Salam G. Abdel-Salam Qatar University Doha, Qatar.
26. d Cyprus International Conference on Educational Research, (CY-ICER 2013) Statistical Methods for Comparison of Data Sets of Construction Methods and Building Evaluation Hamed Niroumanda\*, M.F.M Zainb, Maslina Jamilc.
27. Extension of the Visualization Tool MapMan to Allow Statistical Analysis of Arrays, Display of Corresponding Genes, and Comparison with Known Responses1 Bjoörn Usadel2 \*, Axel Nagel2 , Oliver Thimm3 , Henning Redestig, Oliver E. Blaesing, Natalia Palacios-Rojas4 , Joachim Selbig, Jan Hannemann, Maria Conceição Piques, Dirk Steinhäuser, Wolf-Rüdiger Scheible, Yves Gibon, Rosa Morcuende5 , Daniel Weicht, Svenja Meyer, and Mark Stitt Max Planck Institute of Molecular Plant Physiology, 14476 Golm, Germany (B.U., O.T., H.R., O.E.B., N.P.-R., J.S., J.H., M.C.P., D.S., W.-R.S., Y.G., R.M., D.
28. A Large-scale Study about Quality and Reproducibility of Jupyter Notebooks Joao Felipe Pimentel ~ \*, Leonardo Murta\*, Vanessa Braganholo\*, and Juliana Freire† \*Universidade Federal Fluminense Niteroi, Brazil ' {jpimentel,leomurta,vanessa}@ic.uff.br †New York University New York, USA [juliana.freire@nyu.edu](mailto:juliana.freire@nyu.edu)
29. Data collection, primary versus secondary JJ Hox, HR Boeijs - 2005 - [dspace.library.uu.nl](https://dspace.library.uu.nl).
30. ML.io:<https://www.mldata.io/> Accessed 28/08/2020.
31. Kaggle:<https://www.kaggle.com/datasets/> Accessed 28/08/2020.
32. Data Cleaning: Problems and Current Approaches Erhard Rahm Hong Hai Do University of Leipzig, Germany :<http://dbs.uni-leipzig.de/> / Accessed 28/08/2020.
33. The multivariate skew-normal distribution BY A. AZZALINI AND A. DALLA VALLE Department of Statistical Sciences, University of Padua, Via S. Francesco 33, 35121 Padova, Italy.
34. UCI:<https://archive.ics.uci.edu/ml/datasets/User+Knowledge+Modeling/> Accessed 28/08/2020.
35. Python for Data Analysis by Wes McKinney Released October 2012 Publisher(s):O'Reilly Media, Inc. ISBN: 9781449319793.
36. Pandapower An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems Leon Thurner , Alexander Scheidler, Florian Schafer, Jan-Hendrik Menke , Julian Dollichon, Friederike Meier, Steffen Meinecke, and Martin Braun , Senior Member, IEEE.
37. Nengo: A Python tool for building large-scale functional brain models.

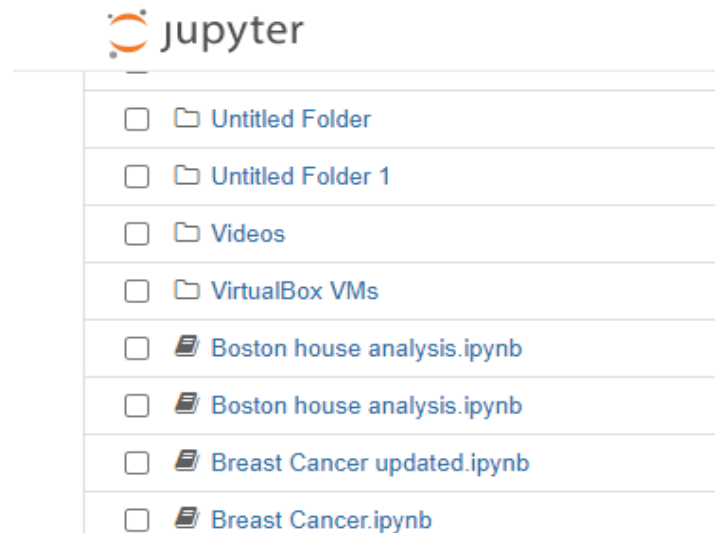


## 7 Appendix A: To run the Jupyter Notebook to Check the Data Visualization Techniques and Statistical Analysis Method to Compare Two Datasets .

**we need to have the following pre-requisites installed:**

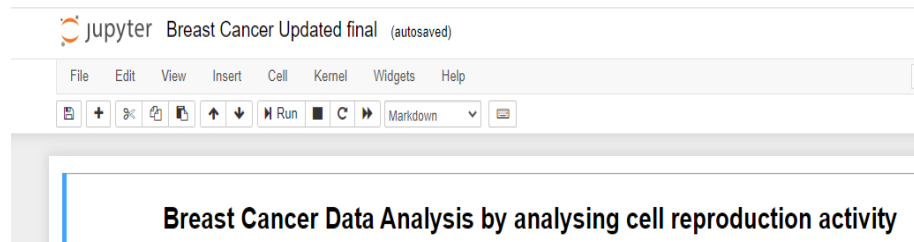
1. Text Editor. Any text editor which supports Python can be used. However, we will use Jupyter [28] for this example.
2. Pandas Library
3. Matplotlib Library
4. Seaborn Library
5. NumPy Library
6. Scipy Library
7. Datasets

Once the mentioned pre-requisites are installed on your local computer, to set up the Notebook we must first unzip the file Breast Cancer Updated final and import the extracted folder into the text editor. Using Anaconda first to launch the Jupyter notebook.




Choose the folder to where the file is downloaded, then open the file .

Once the file launches in Jupyter we can run the file.



The next step is to load the datasets, we can load the datasets from any corner of the local system.

 breast_cancer_dataset.csv	05/08/2020 17:01	Microsoft Excel C...	13 KB
---	------------------	----------------------	-------

Once the dataset has been loaded then we can simply compare two datasets and perform the test.