Name: Krutika Gajanan Rajpure

# **Topic:**

**Implementation of Data Analytics use case using Python libraries**

**Content:**

* Introduction
* Data History
* Data Summary
* Data Description
* Outlier And Missing Value Analysis
* Descriptive Statistics
* Analysis and Inference
* Correlation analysis
* Pivot Table
* Conclusion
* Bibliography
* Python Code

**Introduction**:

**Exploratory Data Analysis or Data Visualization** is a statistical approach or technique for analysing data sets in order to summarize their important and main characteristics generally by using some visual aids. The EDA approach can be used to gather knowledge about the following aspects of data:

* Main characteristics or features of the data.
* The variables and their relationships.
* Finding out the important variables that can be used in our problem.

## EDA is an iterative approach that includes:

## Generating questions about our data. Searching for the answers by using visualization, transformation, and modelling of our data. Using the lessons that we learn in order to refine our set of questions or to generate a new set of questions.

## Data Analysis in Python:

In Python language we are going to perform EDA under two broad classifications:

* **Descriptive Statistics,** which includes mean, median, mode, inter-quartile range, outlier and so on.
* **Graphical Methods,** which includes histogram, density estimation, box plots, scatter plot and so on.

**Data** **History**

**Dataset**:

Kidney Disease

**About the data:**

The data collection is a secondary method used for this assessment as it has been chosen form Kaggle.

The data was taken over a 2-month period in India with 25 features (eg, red blood cell count, white blood cell count, etc). The target is the 'classification', which is either 'ckd' or 'notckd' - ckd=chronic kidney disease. There are 400 rows. The data contains some really rich and interesting pattern among different variables including the different disease, age of the person and also count of cells and minerals which will help in better interpretation and discussion.

The data needs cleaning as it contains NaNs values and the numeric features and for categorical variable needs to cleaned and transformed. Data cleaning process will be followed by the outlier analysis and missing values. Next, a series of descriptive statistics and some basic summarization of data will be done. At times, the analysis will also be accompanied by some plots and charts for better understanding of the data.

**Data Summary**

|  |  |
| --- | --- |
| **Abstract**: This dataset can be used to predict the chronic kidney disease and it can be collected from the hospital nearly 2 months of period. |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Set Characteristics:** | Multivariate | **Number of Instances:** | 400 | **Area:** | N/A |
| **Attribute Characteristics:** | Real | **Number of Attributes:** | 25 | **Date Donated** | 2015-07-03 |
| **Associated Tasks:** | Classification | **Missing Values?** | Yes | **Number of Web Hits:** | 242396 |

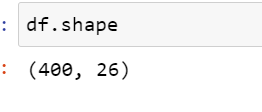
Following are the columns in the dataset:

We use 24 + class = 25 (14 numeric ,11 nominal)

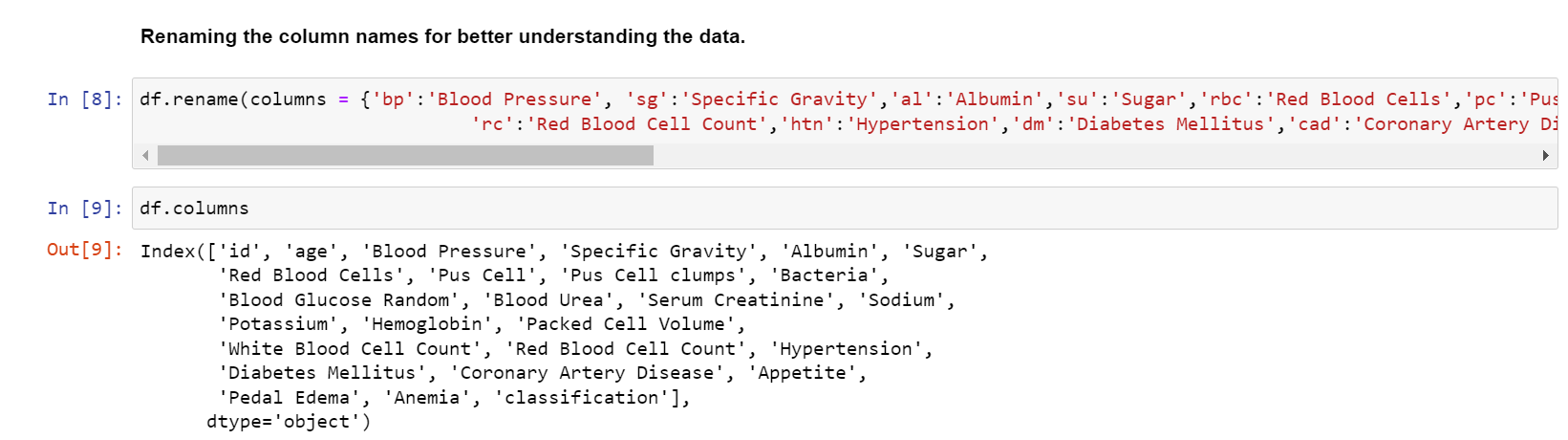
1.Age(numerical) : age in years  
2.Blood Pressure(numerical): bp in mm/Hg  
3.Specific Gravity(nominal): sg - (1.005,1.010,1.015,1.020,1.025)  
4.Albumin(nominal): al - (0,1,2,3,4,5)  
5.Sugar(nominal): su - (0,1,2,3,4,5)  
6.Red Blood Cells(nominal): rbc - (normal ,abnormal)  
7.Pus Cell (nominal): pc - (normal ,abnormal)  
8.Pus Cell clumps(nominal): pcc - (present , notpresent)  
9.Bacteria(nominal): ba - (present , notpresent)  
10.Blood Glucose Random(numerical): bgr in mgs/dl  
11.Blood Urea(numerical): bu in mgs/dl  
12.Serum Creatinine(numerical): sc in mgs/dl  
13.Sodium(numerical): sod in mEq/L  
14.Potassium(numerical): pot in mEq/L  
15.Hemoglobin(numerical): hemo in gms  
16.Packed Cell Volume(numerical)  
17.White Blood Cell Count(numerical): wc in cells/cumm  
18.Red Blood Cell Count(numerical): rc in millions/cmm  
19.Hypertension(nominal): htn - (yes , no)  
20.Diabetes Mellitus(nominal): dm - (yes , no)  
21.Coronary Artery Disease(nominal): cad - (yes, no)  
22.Appetite(nominal): appet - (good, poor)  
23.Pedal Edema(nominal): pe - (yes, no)  
24.Anemia(nominal): ane - (yes, no)  
25.Class (nominal): class - (ckd, notckd)

**Data Description**

On the looking at the data, we can observe that there are in total 26 variables and 400 observations, out of which there are 11 numeric/integer variables, and 14 categorical variables and 1 is the index.

****

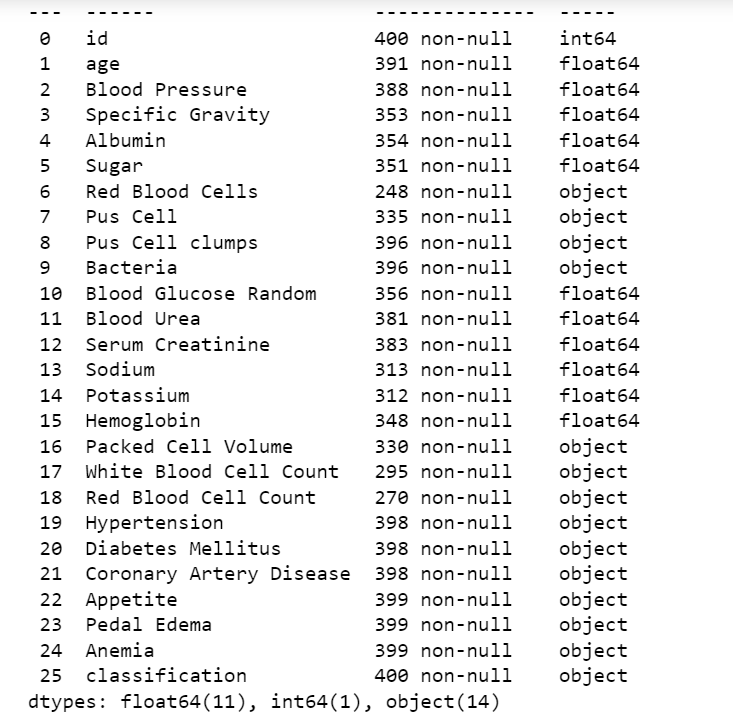
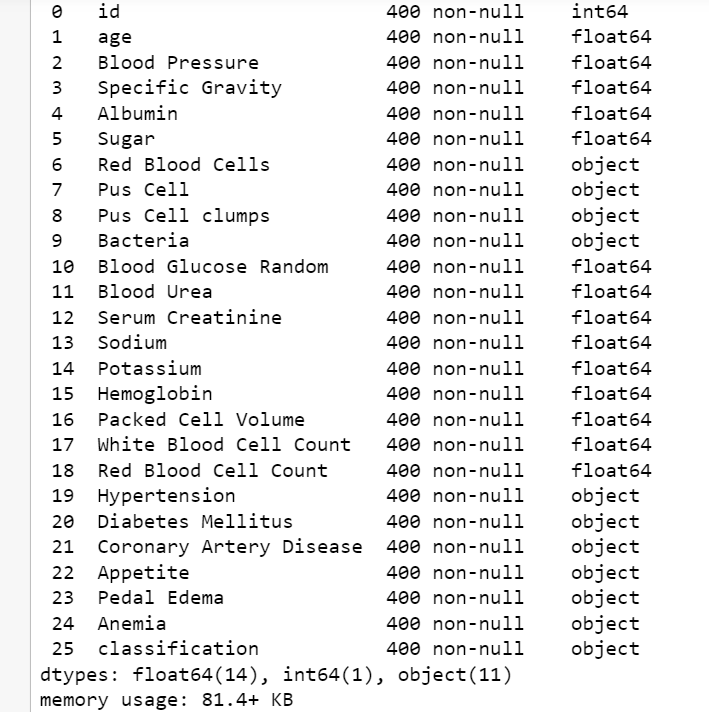
Before starting the work on the data, we have renamed the variable with a suitable and shorter name for better understanding of the variables and make our work easy and comfortable for analysing and performing data visualizing on the variables.



Out of those 14 categorical variables, we can see that there are 3 variable which are defined as numeric in the data summary given above but as seen as categorical variables. We can transform the 3 variables to numeric variables as it will help for better data visualizationand analysis. The remaining 3 variables have a lot of different level of numerical value which will them irrelevant as a factor variable.

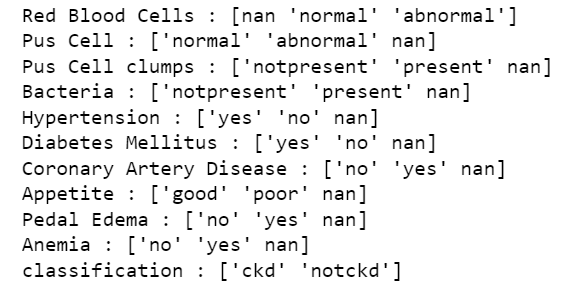
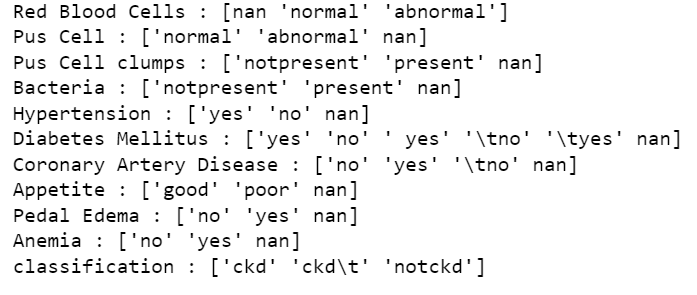
Thus, after transforming the variables to numeric, we have 14 numeric/integer variables, and 11 categorical variables and 1 is the index. Figure1: show the data type before the transformation and Figure2: shows the data type after the transformation. Giving at a quick look at the variable gives us a hint that the data has few inconsistencies, in terms of missing variables, outliers and irrelevant string which we need to clean before we move to the analysis portion.

**Fig 1: Fig 2:**

Apart from the data type issue, in categorical there were some whitespaces after and before the data which needed to be cleaned as it will cause a problem during our analyses further. Thus, we removed the whitespace in the categorical variables. We can observe it below as the unique values are displayed for before trimming the data and after cleaning the data. For e.g., the classification variable had 3 unique data before trimming the whitespaces and after trimming we can see 2 data.

**Fig 1: Fig 2:**



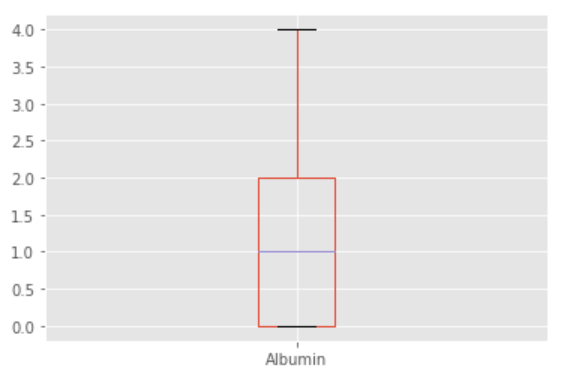
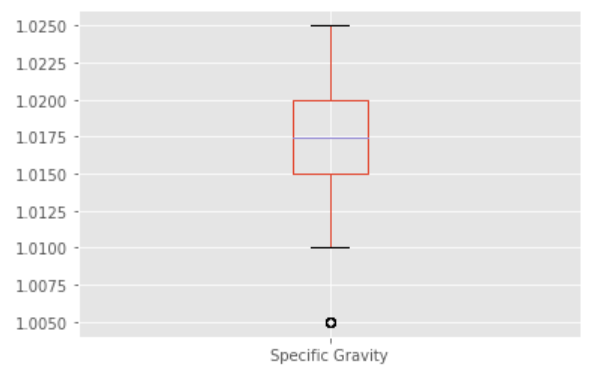
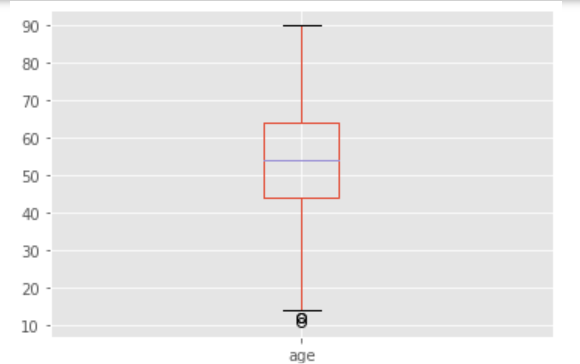
**Outlier And Missing Value Analysis**

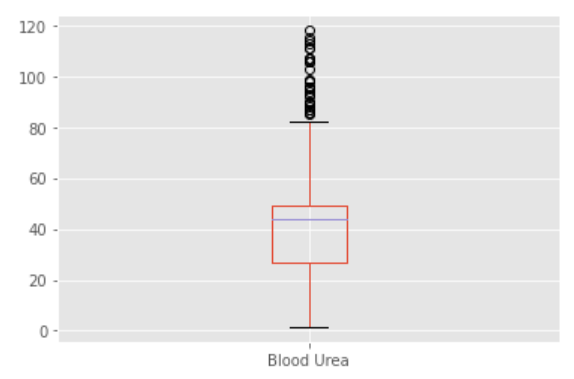
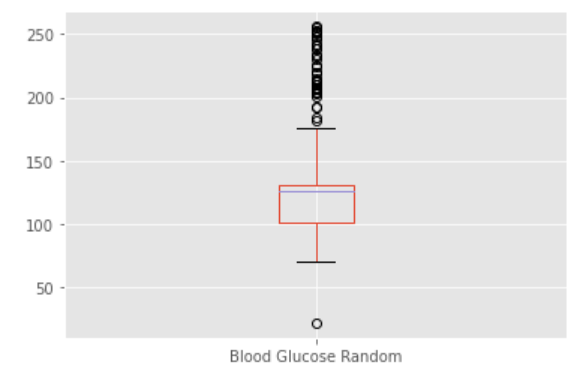
Whenever we talk about data analysis, the term outliers and missing values often come to our mind. As the name suggests, "outliers" refer to the data points that exist outside of what is to be expected and “missing values” refers to the blank data points in the data set.

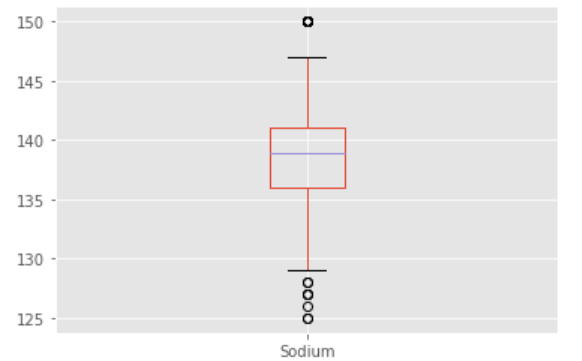
To work on outliers and missing value analysis we are going to subset the variable on the bases of data types i.e., there are two subset one is for categorical and other numerical subset. While dealing with the outliers and missing values we will treat the categorical variable differently and numerical variable differently.

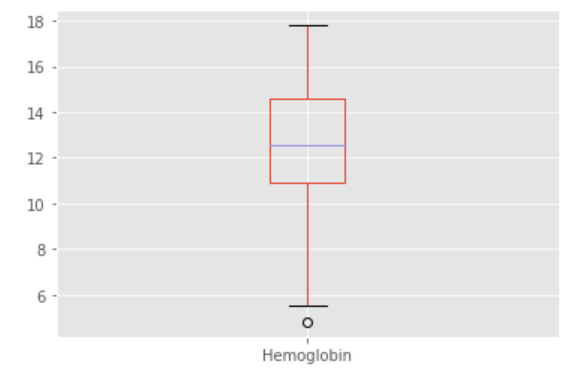
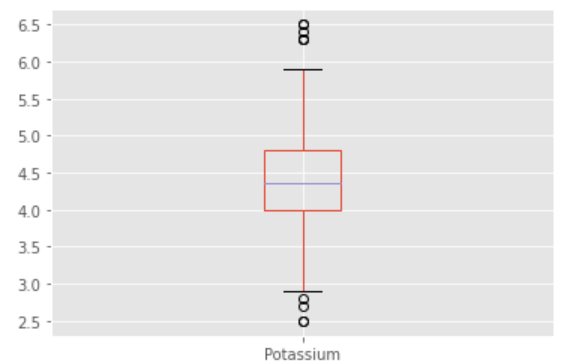
For numerical variables we are plotting boxplot. A boxplot helps us in visualizing the data in terms of quartiles. It also identifies outliers in the dataset, if any. We have plotted the boxplot for all the numerical variables, in the figure we can observe the boxplot each variable with the name. In the figure below, the length of the box is IQR, and the minimum and maximum values are represented by the whiskers. The whiskers are generally extended into 1.5\*IQR distance on either side of the box. Therefore, all data points outside these 1.5\*IQR values are flagged as outliers.

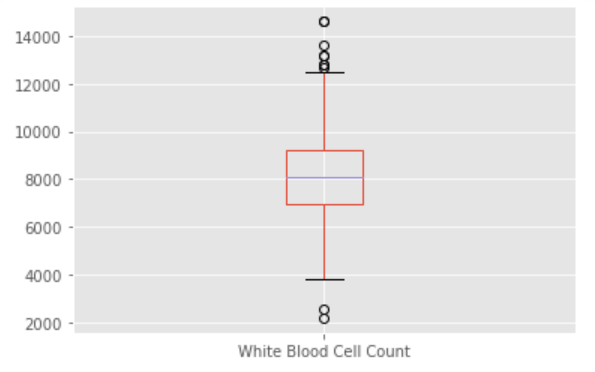
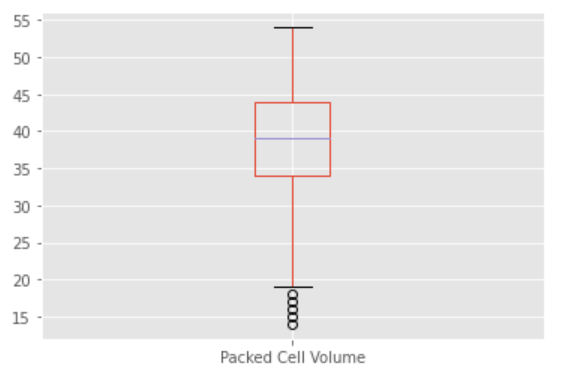
From the below box plots, it is quite clear that the data not only contains outliers but also some human error, possibly illogical numbers, for example, in blood pressure it seems like there is a person who has taken blood pressure greater that 180 which is clearly not possible as in medical, If you get a blood pressure reading of 180/110 or higher more than once, seek medical treatment right away. Hence, we need to remove any such outlier data to make the variables more logical and consistent.

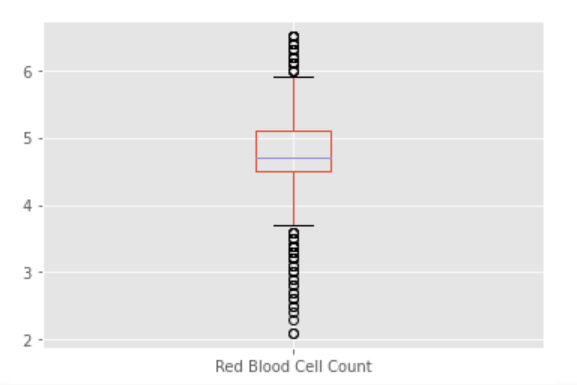










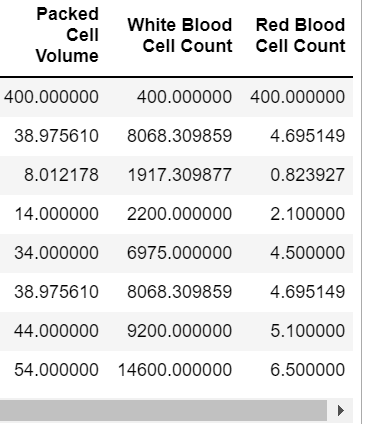
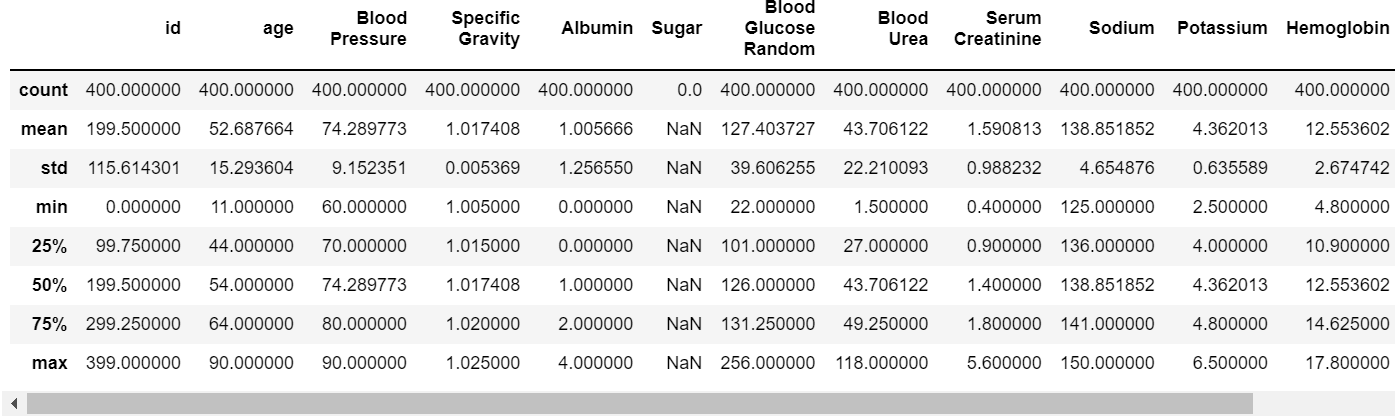


**Missing Values in Variables:**

For numeric values, we can use the mean or median of the data as the imputed value, I have selected mean of each variable to impute that in place of missing data. For categorical value, I have selected most frequently occurring value namely the mode for each column to impute that in place of missing data.

**Descriptive** **Statistics**

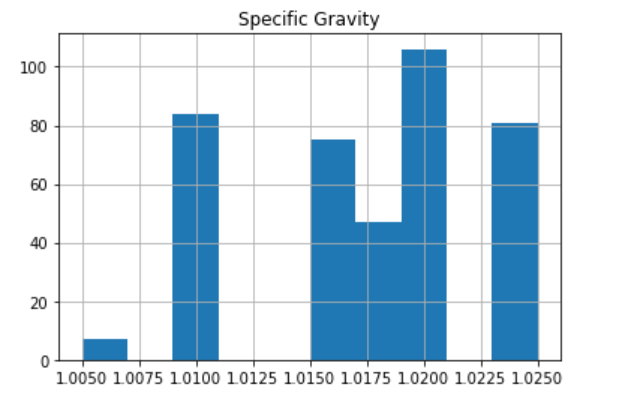
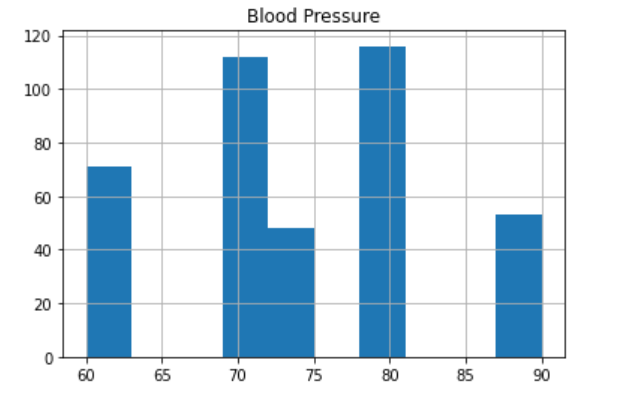
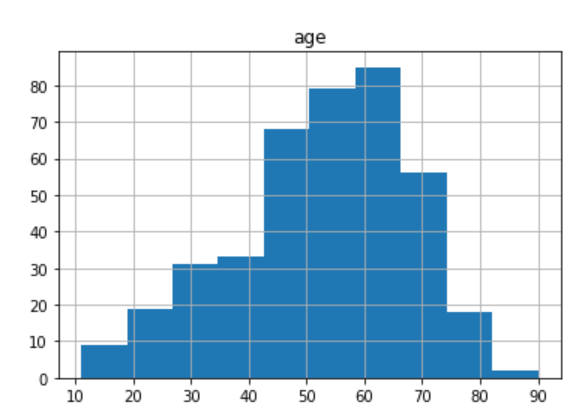
Descriptive statistics summarize and organize characteristics of a data set. A data set is a collection of responses or observations from a sample or entire population. The describe () method returns description of the data in the Data Frame.

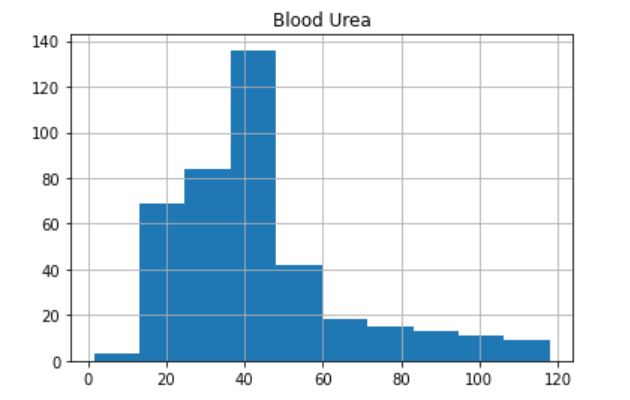
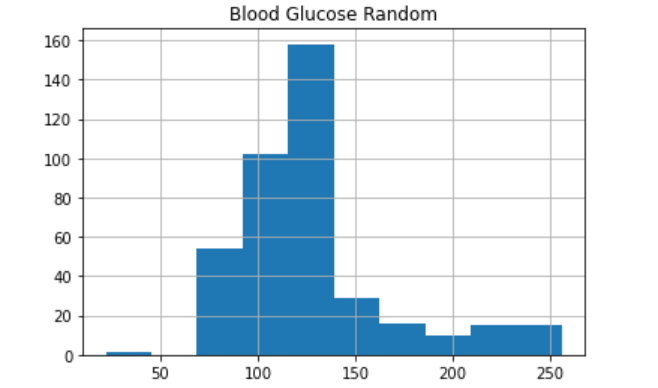
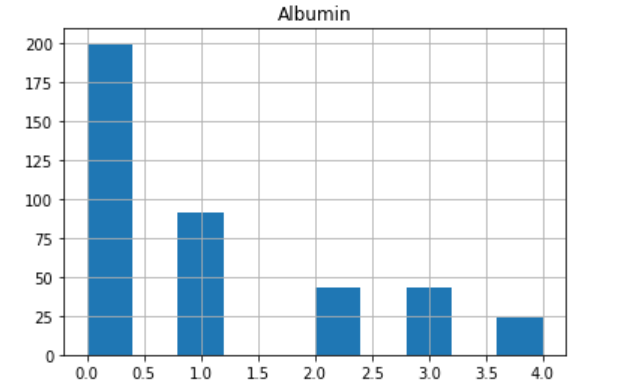


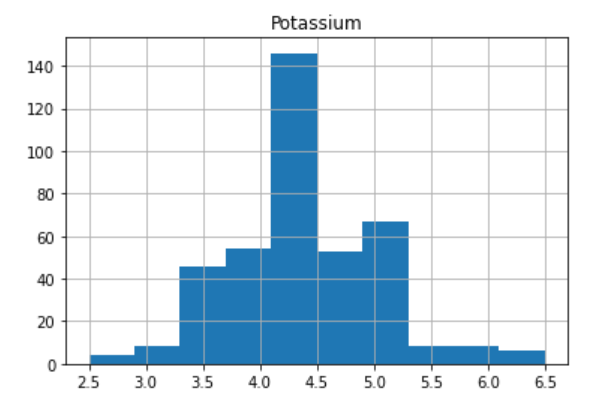
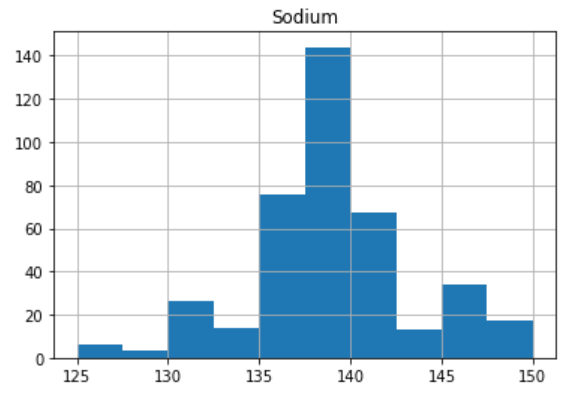
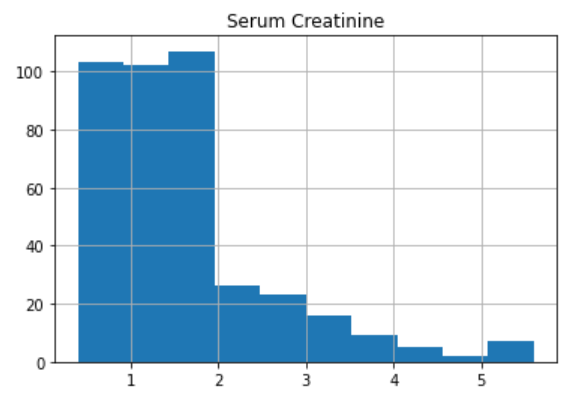
**Analysis and Inference**

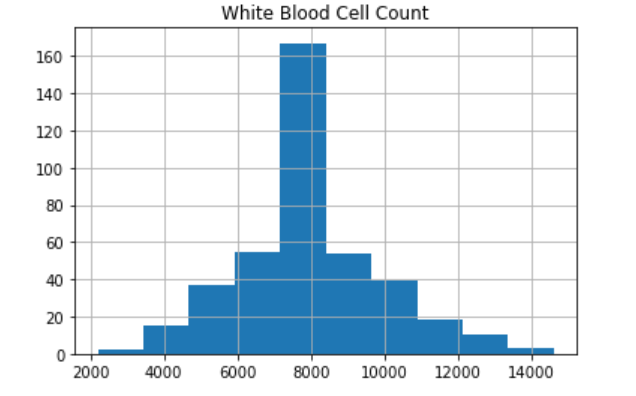
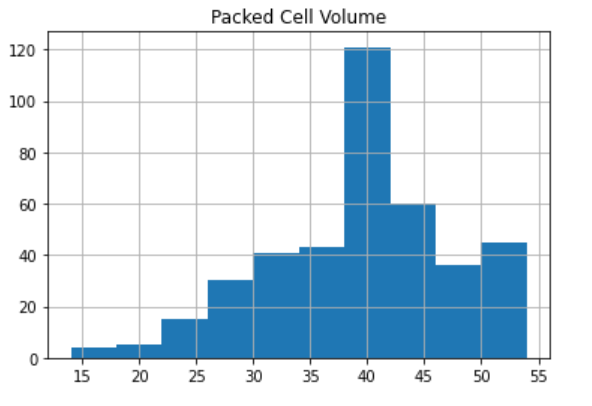
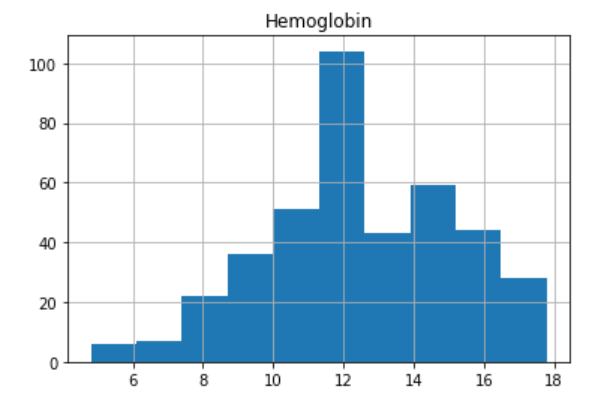
Basic graphical representation for all the numerical variable is done by histogram. Use histograms for continuous measurements and place them into ranges of values known as bins. Each bin has a bar that represents the count or percentage of observations that fall within that bin. For e.g., the highest number of frequencies for age variable lies between 40 and 60. This is the univariate analysis as we are analysis each variable individually against the bins to check the frequency.

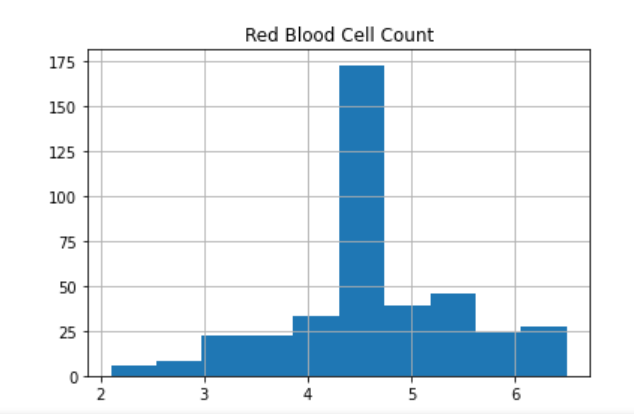
**Figure shows histogram for each numerical variable**









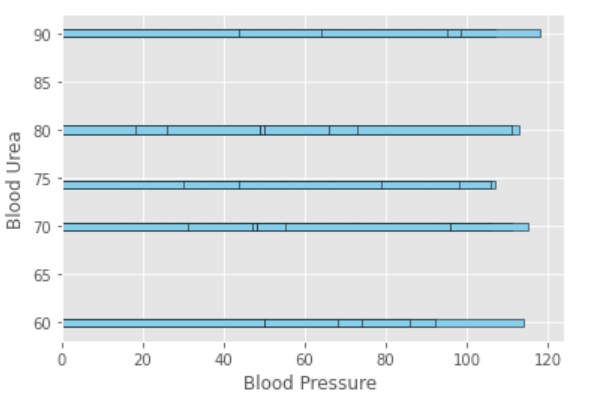
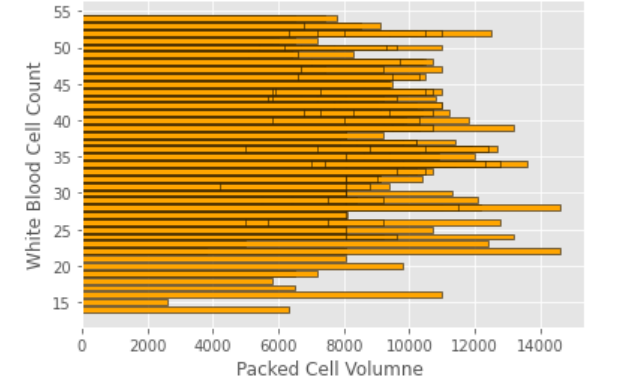


**Bar Plot**

A bar plot is a graphical representation of data in which we can highlight the category with particular shapes like a rectangle. The length and heights of the bar chart represent the data distributed in the dataset. In a bar plot, we have one axis representing a particular category of a column in the dataset and another axis representing the values or counts associated with it. Bar graph is usually used to visually represent comparison of magnitudes over different categories, or a dimension (like time).

For e.g. We have plotted variables White Blood Cell Count Vs Packed Cell Volume i.e., the length of the test measures how much of the blood consists of cells and Blood Urea Vs Blood Pressure i.e., is the pressure of circulating blood against the walls of blood vessels versus Blood Urea.

For the below bar plot, we have plotted Blood Urea vs Blood Pressure to check if the person Blood Urea is related to Blood Pressure and we can see that the high Blood Urea has a greater number of Blood Pressure.

****

**Bivariate Analysis**

The types of a bivariate analysis will depend upon the types of variables or attributes we will use for analysing. The variable could be numerical, categorical or ordinal. If the independent variable is categorical.

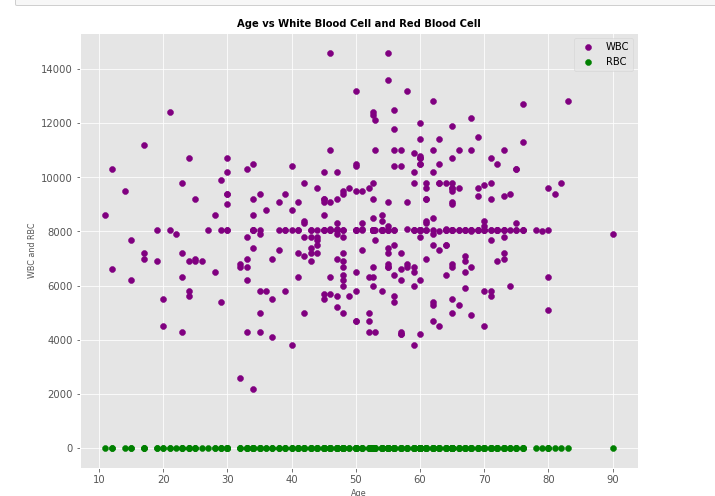
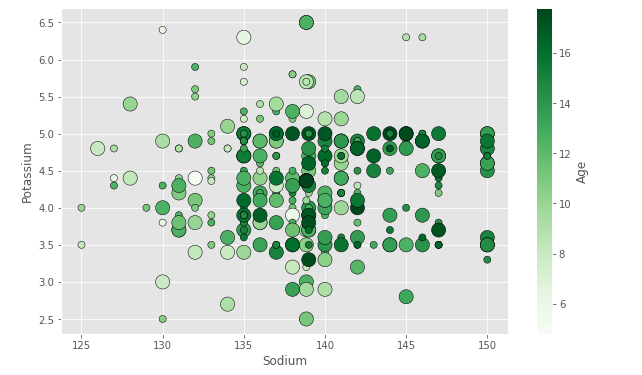
**Scatter plot**

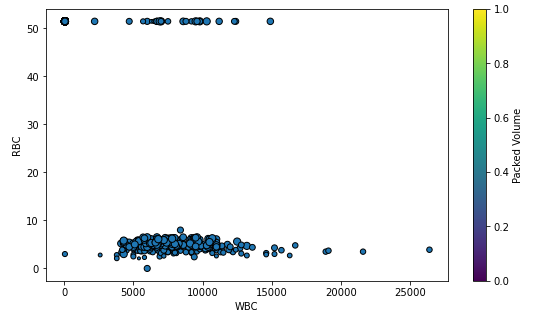
A Scatter Diagram is used in “Control Quality” process as a corrective action approach to understand causes of poor performance of dependent variable due to independent variable effect. To conclude, a scatter plot is a mechanism to understand the influence of independent variable over dependent variable. We have performed the scatter plot have on different plot such as blood glucose random vs age as we can observe in

Figure: 1, I checked the linear relationship between age and white blood cell count and red blood cell count.

For Figure:2, We have plotted sodium vs potassium by grouping the age. When the points in the graph are rising, moving from left to right, then the scatter plot shows a positive correlation.

For Figure:3, We have plotted white blood cell count vs red blood cell count and group by Red Blood Cell categorical variable if normal or abnormal. When the points are scattered all over the graph and it is difficult to conclude whether the values are increasing or decreasing, then there is no correlation between the variables.

** **

****

**Correlation analysis**

Analysis between multiple numerical variables and categorial variables by plotting it using heatmap chart. The heatmap is used to represent matrix values graphically with different colour shades for different values.

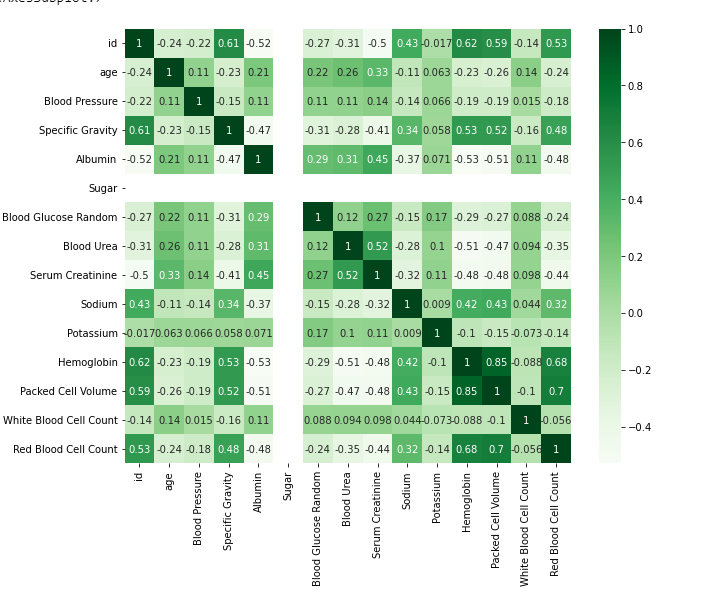
**Multivariate**

Multivariate is the investigation of data where various estimations are made on each trial unit and where the connections among multivariate estimations and their construction are significant.

Multivariate get Pearson correlation coefficient for your dataset. As we observe sugar variable is factor of it is Nan for it.



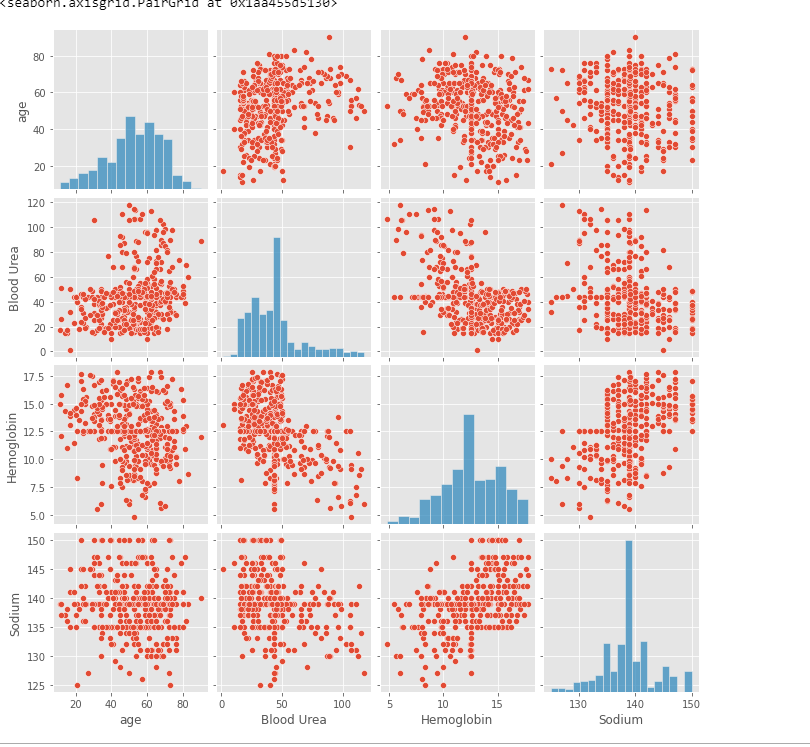
A heatmap is one of the components supported by seaborn where variation in related data is portrayed using a colour palette. This article centrally focuses on a correlation heatmap and how seaborn in combination with pandas and matplotlib can be used to generate one for a data frame.



**Pair Plot**

The pair plot function creates a grid of Axes such that each variable in data will by shared in the y-axis across a single row and in the x-axis across a single column. That creates plots as shown below. The pair plot is shown in the image below.

For the below figure we have used age, blood urea, hemoglobin and sodium variables with each other.

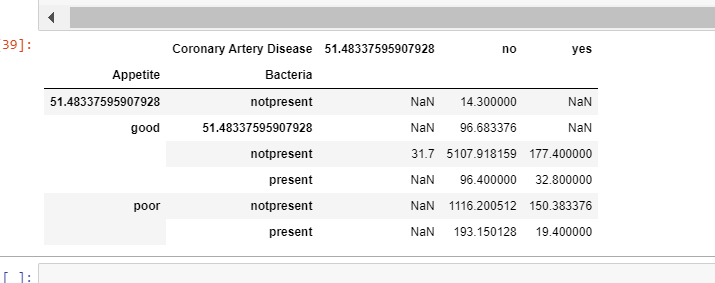


**Pivot table**

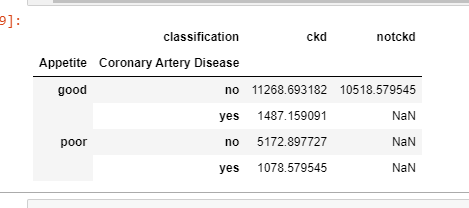
Pivot table is a statistical table that summarizes a substantial table like big datasets. It is part of data processing. This summary in pivot tables may include mean, median, sum, or other statistical terms.

We have plotted three different types of plots as given below in the figures.

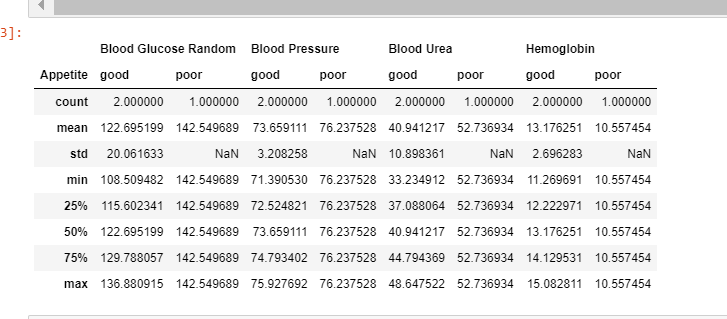
**Figure 1:**



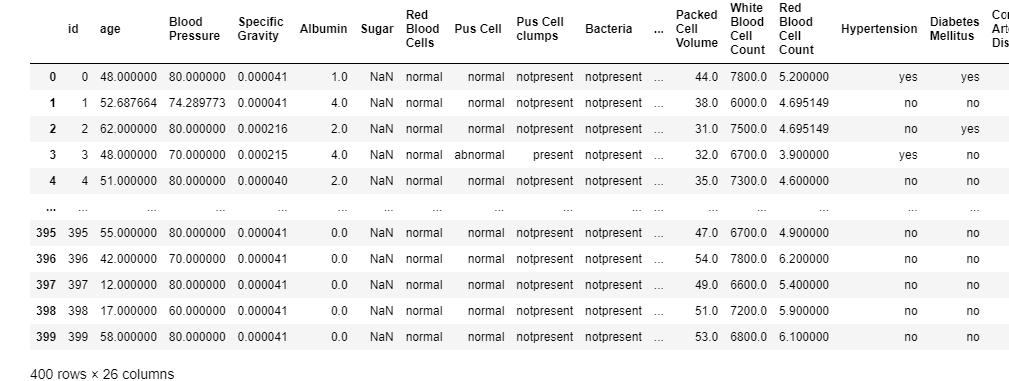
**Figure 2:**



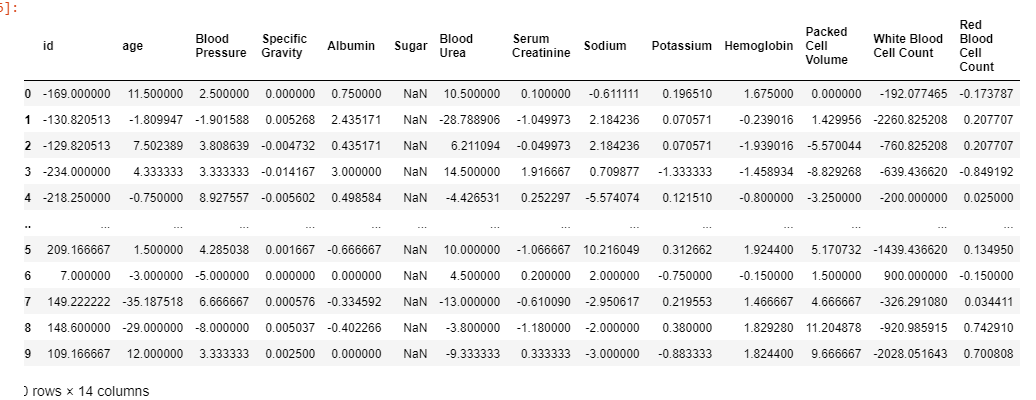
**Figure 3:**



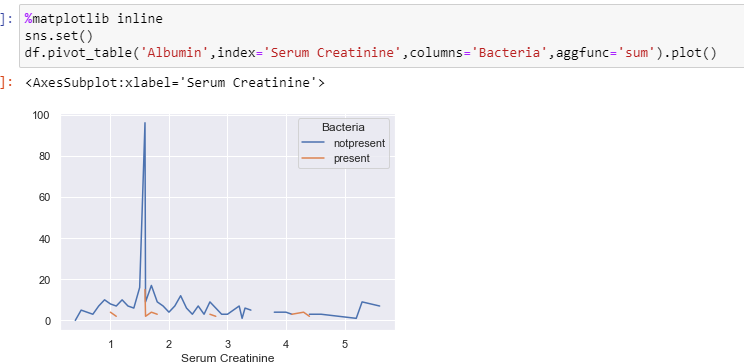
Creating a user defined function to perform normalization and then using that function for aggregations.



Transforming a variable using lambda function and then aggregating based on the transformation



We can plot pivot table in graphs format for better analysis and visualization to derive insight for it.



**Conclusion**

Exploratory Data Analysis (EDA) and Data Visualization are powerful tools and that can highlight problems to be addressed, lead to insights, and suggest patterns in the date. While doing Exploratory Data Analysis on the Data Set chosen for the Kaggle website we have find out few observations. To do any analysis on the dataset, it is necessary to remove the outliers, missing values and irrelevant data, which can cause an error while analysis, from the dataset before doing any analysis.

There are approx. 90% of columns that contains missing values in the data and outlier we have to clean the data. The numerical variables are correlated to each other. The Diabetes Mellitus is associated with the coronary artery disease.

**Bibliography**

• <https://www.kaggle.com/datasets/mansoordaku/ckdisease>

• <https://www.analyticsvidhya.com/blog/2021/05/exploratory-dataanalysis-eda-a-step-by-step-guide/>

• Geeksforgeeks. (2022, 9). Retrieved from www.geeksforgeeks.com: https://geeksforgeeks.com

**Python** **Code:**

**import** numpy **as** np

**import** matplotlib**.**pyplot **as** plt

**import** seaborn **as** sns

**%**matplotlib inline

**\*\*\***Loading the dataset**\*\*\***

df**=**pd**.**read\_csv**(**'kidney\_disease.csv'**,**encoding**=**'latin-1'**)**

df**=**pd**.**read\_csv**(**'kidney\_disease.csv'**,**encoding**=**'latin-1'**)**

df**.**head**()**

df**.**columns

df**.**shape

Renaming the column names **for** better understanding the data**.**

df**.**rename**(**columns **=** **{**'bp'**:**'Blood Pressure'**,** 'sg'**:**'Specific Gravity'**,**'al'**:**'Albumin'**,**'su'**:**'Sugar'**,**'rbc'**:**'Red Blood Cells'**,**'pc'**:**'Pus Cell'**,**'pcc'**:**'Pus Cell clumps'**,**'ba'**:**'Bacteria'**,**'bgr'**:**'Blood Glucose Random'**,**'bu'**:**'Blood Urea'**,**'sc'**:**'Serum Creatinine'**,**'sod'**:**'Sodium'**,**'pot'**:**'Potassium'**,**'hemo'**:**'Hemoglobin'**,**'pcv'**:**'Packed Cell Volume'**,**'wc'**:**'White Blood Cell Count'**,**

'rc'**:**'Red Blood Cell Count'**,**'htn'**:**'Hypertension'**,**'dm'**:**'Diabetes Mellitus'**,**'cad'**:**'Coronary Artery Disease'**,**'appet'**:**'Appetite'**,**'pe'**:**'Pedal Edema'**,**'ane'**:**'Anemia'**},** inplace **=** **True)**

df**.**columns

df**.**info**()**

Converting the 3 columns into **int** **type** form **object**

**print(**"Packed Cell Volume: "**,**df**[**'Packed Cell Volume'**].**unique**())**

**print(**"White Blood Cell Count: "**,**df**[**'White Blood Cell Count'**].**unique**())**

**print(**"Red Blood Cell Count: "**,**df**[**'Red Blood Cell Count'**].**unique**())**

df**[**'Packed Cell Volume'**]=**df**[**'Packed Cell Volume'**].str.**strip**()**

df**[**'White Blood Cell Count'**]=**df**[**'White Blood Cell Count'**].str.**strip**()**

df**[**'Red Blood Cell Count'**]=**df**[**'Red Blood Cell Count'**].str.**strip**()**

**print(**"Packed Cell Volume: "**,**df**[**'Packed Cell Volume'**].**unique**())**

**print(**"White Blood Cell Count: "**,**df**[**'White Blood Cell Count'**].**unique**())**

**print(**"Red Blood Cell Count: "**,**df**[**'Red Blood Cell Count'**].**unique**())**

#replacing ? to NA

df**[**"Packed Cell Volume"**].**replace**({**"?"**:**np**.**nan**},** inplace**=True)**

df**[**"White Blood Cell Count"**].**replace**({**"?"**:**np**.**nan**},** inplace**=True)**

df**[**"Red Blood Cell Count"**].**replace**({**"?"**:**np**.**nan**},** inplace**=True)**

#converting to numerical variable

df**[[**'Packed Cell Volume'**,** 'White Blood Cell Count'**,**'Red Blood Cell Count'**]]** **=** df**[[**'Packed Cell Volume'**,** 'White Blood Cell Count'**,**'Red Blood Cell Count'**]].**apply**(**pd**.**to\_numeric**)**

#reching if the ? is repaced by NA

**print(**"Packed Cell Volume: "**,**df**[**'Packed Cell Volume'**].**unique**())**

**print(**"White Blood Cell Count: "**,**df**[**'White Blood Cell Count'**].**unique**())**

**print(**"Red Blood Cell Count: "**,**df**[**'Red Blood Cell Count'**].**unique**())**

#rechecking data

df**.**info**()**

#Numerical subset of variables

num\_var1**=**df**.**select\_dtypes**([**'float64'**]).**columns

num\_var1

Categorical Variable

#categorical variables

cat\_var**=**df**.**select\_dtypes**([**'object'**]).**columns

**for** x **in** cat\_var**:**

**print(**x**,**":"**,**df**[**x**].**unique**())**

#cleaning the unwanted whitespaces in the data

df**[**'Diabetes Mellitus'**]=**df**[**'Diabetes Mellitus'**].str.**strip**()**

df**[**'Coronary Artery Disease'**]=**df**[**'Coronary Artery Disease'**].str.**strip**()**

df**[**'classification'**]=**df**[**'classification'**].str.**strip**()**

#rechecking if the categorical variables are

cat\_var1**=**df**.**select\_dtypes**([**'object'**]).**columns

**for** x **in** cat\_var1**:**

**print(**x**,**":"**,**df**[**x**].**unique**())**

Finding outliers **and** the missing values **in** the dataset

#to check which column has missing values

#list comprehension, features is a temporary vaiable which will iterate through df.columns if that speific feature is null

df**.**isna**().sum()**

BOX PLOT Before Outlier Treatment

For Numerical Variable

num\_var1**=**df**.**select\_dtypes**([**'float64'**]).**columns

**for** i **in** num\_var1**:**

plt**.**figure**()**

df**.**boxplot**(**i**)**

plt**.**savefig**(**'boxplot.png'**)**

Finding **and** treating outliers **in** numerical variables Filling the outliers **with** nan values

**def** remove\_outlier**(**df**):**

num\_var1**=**df**.**select\_dtypes**([**'float64'**]).**columns

**for** col **in** num\_var1**:**

#print(col)

q1 **=** df**[**col**].**quantile**(**0.25**)**

q3 **=** df**[**col**].**quantile**(**0.75**)**

iqr **=** q3**-**q1 #Interquartile range

low\_lim **=** q1**-**1.5**\***iqr

up\_lim **=** q3**+**1.5**\***iqr

#print(df[col][~((df[col] < up\_lim)&(df[col] > low\_lim))])

df**[**col**][~((**df**[**col**]** **<** up\_lim**)&(**df**[**col**]** **>** low\_lim**))]** **=** np**.**nan

**return** df

remove\_outlier**(**df**)**

Rechecking the missing values **as** we have replaced outliers **with** NAN We can observe the number of misisng values have increased **as** the oultier also have been replaced by NAN

df**.**isna**().sum()**

Replacing Categorical Variable **with** mode **and** Numerical variable **with** mean

#num\_var1=df.select\_dtypes(['float64']).columns

**for** y **in** num\_var1**:**

**print(**y**)**

mean\_value**=**df**[**y**].**mean**()**

**print(**mean\_value**)**

df**[**y**].**fillna**(**value**=**mean\_value**,** inplace**=True)**

#df[y].replace(np.nan, mean\_value)

cat\_var1**=**df**.**select\_dtypes**([**'object'**]).**columns

**for** x **in** cat\_var1**:**

**print(**x**)**

mode\_value**=**df**[**x**].**mode**()[**0**]**

**print(**mode\_value**)**

#df[x].replace(np.nan, mode\_value)

df**[**x**].**fillna**(**value**=**mode\_value**,** inplace**=True)**

#rechecking if the categorical variables are

cat\_var1**=**df**.**select\_dtypes**([**'object'**]).**columns

**for** x **in** cat\_var1**:**

**print(**x**,**":"**,**df**[**x**].**unique**())**

Rechecking **if** the missing values are present after the treating them

df**.**isna**().sum()**

BOXPLOT

After outlier **and** missing values treatment

num\_var1**=**df**.**select\_dtypes**([**'float64'**]).**columns

**for** i **in** num\_var1**:**

plt**.**figure**()**

df**.**boxplot**(**i**)**

Descriptive Statistics

Statistics **for** numerical variables only**.**

df**.**describe**()**

Graphical representataion

Univariate Analysis

Histogram **for** **all** numerical columns

num\_var1**=**df**.**select\_dtypes**([**'float64'**]).**columns

**for** i **in** num\_var1**:**

plt**.**figure**()**

df**.**hist**(**i**)**

Bar Plot

plt**.**style**.**use**(**'ggplot'**)**

plt**.**barh**(**df**[**'Packed Cell Volume'**],**df**[**'White Blood Cell Count'**],** color **=** 'Orange'**,** ec **=** 'black'**)**

plt**.**xlabel**(**'Packed Cell Volumne '**)**

plt**.**ylabel**(**'White Blood Cell Count'**)**

plt**.**show**()**

plt**.**style**.**use**(**'ggplot'**)**

plt**.**barh**(**df**[**'Blood Pressure'**],**df**[**'Blood Urea'**],** color **=** 'SkyBlue'**,** ec **=** 'black'**)**

plt**.**xlabel**(**'Blood Pressure '**)**

plt**.**ylabel**(**'Blood Urea'**)**

plt**.**show**()**

Multivariate

Multivariate **is** the investigation of data where various estimations are made on each trial unit **and** where the connections among multivariate estimations **and** their construction are significant**.**

Multivariate get Pearson correlation coefficient **for** your dataset**.**

df**.**corr**(**method**=**'pearson'**)**

As we observe sugar variable **is** factor of it **is** NaN **for** it

plt**.**figure**(**figsize **=** **(**10**,**8**))**

sns**.**heatmap**(**df**.**corr**(),**annot**=True,**cmap**=**'Greens'**)**

sns**.**pairplot**(**df**,vars=[**'age'**,**'Blood Urea'**,**'Hemoglobin'**,**'Sodium'**])**

plt**.**figure**(**figsize**=(**10**,**8**))**

plt**.**scatter**(**df**[**'age'**],**df**[**'White Blood Cell Count'**],**label**=**'WBC'**,** color **=** 'Purple'**)**

plt**.**scatter**(**df**[**'age'**],**df**[**'Red Blood Cell Count'**],**label**=**'RBC'**,** color **=** 'Green'**)**

plt**.**xlabel**(**'Age'**,**size**=**8**)**

plt**.**ylabel**(**'WBC and RBC'**,**size**=**8**)**

plt**.**title**(**'Age vs White Blood Cell and Red Blood Cell'**,** size **=** 10**,** fontweight **=** 'bold'**)**

plt**.**legend**()**

plt**.**show**()**

plt**.**figure**(**figsize **=** **(**10**,**6**))**

plt**.**scatter**(**df**[**'Sodium'**],**df**[**'Potassium'**],**edgecolor**=**'black'**,**s**=**df**[**'age'**],**sizes**=(**50**,**200**),**c**=**df**[**'Hemoglobin'**],**cmap **=** 'Greens'**)**

plt**.**xlabel**(**'Sodium'**)**

plt**.**ylabel**(**'Potassium'**)**

color\_bar **=** plt**.**colorbar**()**

color\_bar**.**set\_label**(**'Age'**)**

plt**.**show**()**

plt**.**figure**(**figsize **=** **(**9**,**5**))**

plt**.**scatter**(**df**[**'White Blood Cell Count'**],** df**[**'Red Blood Cell Count'**],**edgecolor **=** 'black'**,**s **=** df**[**'Packed Cell Volume'**])**

plt**.**xlabel**(**'White Blood Cell'**)**

plt**.**ylabel**(**'Red Blood Cell'**)**

color\_bar **=** plt**.**colorbar**()**

color\_bar**.**set\_label**(**'Packed Volume'**)**

plt**.**show**()**

table**=**pd**.**pivot\_table**(**df**,** values**=**'Hemoglobin'**,** index**=[**'Appetite'**,** 'Bacteria'**],**columns**=[**'Coronary Artery Disease'**],** aggfunc**=**np**.sum)**

table

table**=**pd**.**pivot\_table**(**df**,** values**=**'Blood Pressure'**,** index**=[**'Appetite'**,** 'Coronary Artery Disease'**],**columns**=[**'classification'**],** aggfunc**=**np**.sum)**

table

df**.**pivot\_table**(**values**=[**'Hemoglobin'**,**'Blood Urea'**,**'Blood Glucose Random'**,**'Blood Pressure'**],**index**=**'classification'**,**columns**=**'Appetite'**,**aggfunc**=**np**.**mean**).**describe**()**

Creating a user defined function to perform normalization **and** then using that function **for** aggregations

**def** norm\_by\_fun**(**df**):**

df**[**'Specific Gravity'**]/=**df**[**'Blood Pressure'**].sum()**

**return** df

df**.**groupby**(**'Anemia'**).**apply**(**norm\_by\_fun**)**

Transforming a variable using **lambda** function **and** then aggregating based on the transformation

df**.**groupby**(**'Blood Glucose Random'**).**transform**(lambda** x**:** x**-**x**.**mean**())**

**%**matplotlib inline

sns**.set()**

df**.**pivot\_table**(**'Albumin'**,**index**=**'Serum Creatinine'**,**columns**=**'Bacteria'**,**aggfunc**=**'sum'**).**plot**()**

**%**matplotlib inline

sns**.set()**

df**.**pivot\_table**(**'Albumin'**,**index**=**'Serum Creatinine'**,**columns**=**'Bacteria'**,**aggfunc**=**'sum'**).**plot**()**