Experiment No.2
Perform Exploratory data analysis of Healthcare Data.
Date of Performance: 26/07/2024
Date of Submission: 02/07/2024

Aim: Perform Exploratory data analysis of Healthcare Data.

Objective: The objective of this experiment is to perform Exploratory data analytics on healthcare data using python numpy functions

Theory:
Exploratory data analysis was promoted by John Tukey to encourage statisticians to explore data, and possibly formulate hypotheses that might cause new data collection and experiments. EDA focuses more narrowly on checking assumptions required for model fitting and hypothesis testing. It also checks while handling missing values and making transformations of variables as needed.

EDA builds a robust understanding of the data, and issues associated with either the info or process. It's a scientific approach to getting the story of the data.

TYPES OF EXPLORATORY DATA ANALYSIS:

Univariate Non-graphical

Multivariate Non-graphical

Univariate graphical

Multivariate graphical

1. Univariate Non-graphical: this is the simplest form of data analysis as during this we use just one variable to research the info. The standard goal of univariate non-graphical EDA is to know the underlying sample distribution/ data and make observations about the population. Outlier detection is additionally part of the analysis. The characteristics of population distribution include:

Central tendency: The central tendency or location of distribution has got to do with typical or middle values. The commonly useful measures of central tendency are statistics called mean, median, and sometimes mode during which the foremost common is mean. For skewed distribution or when there's concern about outliers, the median may be preferred.

Spread: Spread is an indicator of what proportion distant from the middle we are to seek out the find the info values. the quality deviation and variance are two useful measures of spread. The variance is that the mean of the square of the individual deviations and therefore the variance is the root of the variance

Skewness and kurtosis: Two more useful univariates descriptors are the skewness and kurtosis of the distribution. Skewness is that the measure of asymmetry and kurtosis may be a more subtle measure of peakedness compared to a normal distribution

2. Multivariate Non-graphical: Multivariate non-graphical EDA technique is usually wont to show the connection between two or more variables within the sort of either cross-tabulation or statistics.

For categorical data, an extension of tabulation called cross-tabulation is extremely useful. For 2 variables, cross-tabulation is preferred by

making a two-way table with column headings that match the amount of one-variable and row headings that match the amount of the opposite two variables, then filling the counts with all subjects that share an equivalent pair of levels.

For each categorical variable and one quantitative variable, we create statistics for quantitative variables separately for every level of the specific variable then compare the statistics across the amount of categorical variable.

Comparing the means is an off-the-cuff version of ANOVA and comparing medians may be a robust version of one-way ANOVA.

3. Univariate graphical: Non-graphical methods are quantitative and objective, they are not able to give the complete picture of the data; therefore, graphical methods are used more as they involve a degree of subjective analysis, also are required. Common sorts of univariate graphics are:

Histogram: The foremost basic graph is a histogram, which may be a barplot during which each bar represents the frequency (count) or proportion (count/total count) of cases for a variety of values. Histograms are one of the simplest ways to quickly learn a lot about your data, including central tendency, spread, modality, shape and outliers.

Stem-and-leaf plots: An easy substitute for a histogram may be stem-and-leaf plots. It shows all data values and therefore the shape of the distribution.

Boxplots: Another very useful univariate graphical technique is that the boxplot. Boxplots are excellent at presenting information about central tendency and show robust measures of location and spread also as providing information about symmetry and outliers, although

they will be misleading about aspects like multimodality. One among the simplest uses of boxplots is within the sort of side-by-side boxplots.

Quantile-normal plots: The ultimate univariate graphical EDA technique is that the most intricate. it's called the quantile-normal or QN plot or more generally the quantile-quantile or QQ plot. it's wont to see how well a specific sample follows a specific theoretical distribution. It allows detection of non-normality and diagnosis of skewness and kurtosis

4. Multivariate graphical: Multivariate graphical data uses graphics to display relationships between two or more sets of knowledge. The sole one used commonly may be a grouped barplot with each group representing one level of 1 of the variables and every bar within a gaggle representing the amount of the opposite variable.

Other common sorts of multivariate graphics are:

Scatterplot: For 2 quantitative variables, the essential graphical EDA technique is that the scatterplot , sohas one variable on the x-axis and one on the y-axis and therefore the point for every case in your dataset.

Run chart: It's a line graph of data plotted over time.

Heat map: It's a graphical representation of data where values are depicted by color.

Multivariate chart: It's a graphical representation of the relationships between factors and response.

Bubble chart: It's a data visualization that displays multiple circles (bubbles) in two-dimensional plot.

In a nutshell: You ought to always perform appropriate EDA before further analysis of your data. Perform whatever steps are necessary to become more conversant in your data, check for obvious mistakes, learn about variable distributions, and study about relationships between variables. EDA is not an exact science- It is very important are!

TOOLS REQUIRED FOR EXPLORATORY DATA ANALYSIS:

Some of the most common tools used to create an EDA are:

1. R: An open-source programming language and free software environment for statistical computing and graphics supported by the R foundation for statistical computing. The R language is widely used among statisticians in developing statistical observations and data analysis.

2. Python: An interpreted, object-oriented programming language with dynamic semantics. Its high level, built-in data structures, combined with dynamic binding, make it very attractive for rapid application development, also as to be used as a scripting or glue language to attach existing components together. Python and EDA are often used together to spot missing values in the data set, which is vital so you'll decide the way to handle missing values for machine learning.

Apart from these functions described above, EDA can also:

Perform k-means clustering: Perform k-means clustering: it's an unsupervised learning algorithm where the info points are assigned to clusters, also referred to as k-groups, k-means clustering is usually utilized in market segmentation, image compression, and pattern recognition

EDA is often utilized in predictive models like linear regression, where it's wont to predict outcomes.

It is also utilized in univariate, bivariate, and multivariate visualization for summary statistics, establishing relationships between each variable, and understanding how different fields within the data interact with one another.

Code: -

```
[36] import pandas as pd
     import seaborn as sns
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression
     from sklearn.preprocessing import LabelEncoder
     from sklearn.metrics import ConfusionMatrixDisplay
```

```
[37] health = pd.read_csv("/content/healthcare_dataset.csv")
     health
```

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital	Insurance Provider	Billing Amount	Room Number	Ad
0	Bobby JacksOn	30	Male	B-	Cancer	2024-01-31	Matthew Smith	Sons and Miller	Blue Cross	18856.281306	328	
1	LesLie TErRy	62	Male	A+	Obesity	2019-08-20	Samantha Davies	Kim Inc	Medicare	33643.327287	265	Em
2	DaNnY sMitH	76	Female	A-	Obesity	2022-09-22	Tiffany Mitchell	Cook PLC	Aetna	27955.096079	205	Em

```
health.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 55500 entries, 0 to 55499
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Name                55500 non-null  object
 1   Age                 55500 non-null  int64
 2   Gender              55500 non-null  object
 3   Blood Type          55500 non-null  object
 4   Medical Condition   55500 non-null  object
 5   Date of Admission   55500 non-null  object
 6   Doctor              55500 non-null  object
 7   Hospital            55500 non-null  object
 8   Insurance Provider  55500 non-null  object
 9   Billing Amount      55500 non-null  float64
 10  Room Number         55500 non-null  int64
 11  Admission Type      55500 non-null  object
 12  Discharge Date      55500 non-null  object
 13  Medication          55500 non-null  object
 14  Test Results        55500 non-null  object
dtypes: float64(1), int64(2), object(12)
memory usage: 6.4+ MB
```

```
health['Medical Condition'].value_counts()
```

Medical Condition	count
Arthritis	9308
Diabetes	9304
Hypertension	9245
Obesity	9231
Cancer	9227
Asthma	9185

dtype: int64

```
[40] health.Name
```

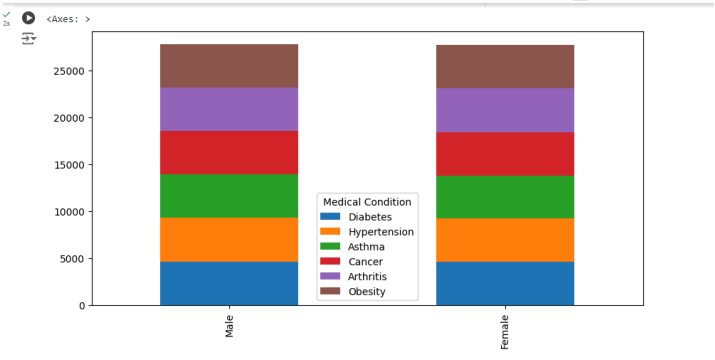
54679	wiLlIAm mcCOY
54680	braNdon tRAn
54681	MICHaEL mASon


```
health['Medical Condition'].value_counts()
```

Medical Condition	count
Arthritis	9308
Diabetes	9304
Hypertension	9245
Obesity	9231
Cancer	9227
Asthma	9185

dtype: int64


```
[42] Male=health[health['Gender']=='Male']['Medical Condition'].value_counts()
     Female=health[health['Gender']=='Female']['Medical Condition'].value_counts()
     df=pd.DataFrame([Male,Female])
     df.index=['Male','Female']
     df.plot(kind='bar',stacked=True,figsize=(10,5))
```



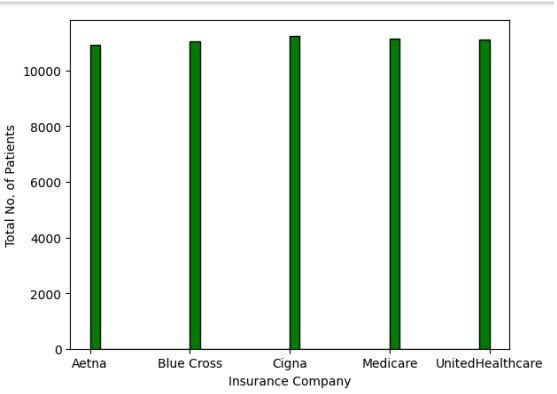
```
health['Insurance Provider'].value_counts()
```

Insurance Provider	count
Cigna	11249
Medicare	11154
UnitedHealthcare	11125
Blue Cross	11059
Aetna	10913

dtype: int64

```
[44] import matplotlib.pyplot as plt
```

```
[45] data=health['Insurance Provider'].sort_values()
     plt.hist(data,bins=40,color='green',edgecolor='black',rwidth=3)
     plt.xlabel('Insurance Company')
     plt.ylabel('Total No. of Patients')
     plt.show()
```




```
[46] health['Admission Type'].unique()
```

```
array(['Urgent', 'Emergency', 'Elective'], dtype=object)
```

```
[47] mapping={'Emergency':0,'Urgent':1,'Elective':2}

     health['Admission Type']=health['Admission Type'].map(mapping)
```

```
[48] health.head(10)
```

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital	Insurance Provider	Billing Amount	Room Number	Admissi Ty
0	Bobby JacksOn	30	Male	B-	Cancer	2024-01-31	Matthew Smith	Sons and Miller	Blue Cross	18856.281306	328	
1	LesLie TErRy	62	Male	A+	Obesity	2019-08-20	Samantha Davies	Kim Inc	Medicare	33643.327287	265	
2	DaNnY sMitH	76	Female	A-	Obesity	2022-09-22	Tiffany Mitchell	Cook PLC	Aetna	27955.096079	205	
3	andrEw waTtS	28	Female	O+	Diabetes	2020-11-18	Kevin Wells	Hernandez Rogers and Vang,	Medicare	37909.782410	450	
4	adrIENNE	43	Female	AB+	Cancer	2022-09-	Kathleen	White-				

Connected to Python 3 Google Compute Engine backend

Speakers (Realtek(R) Audio): 82%

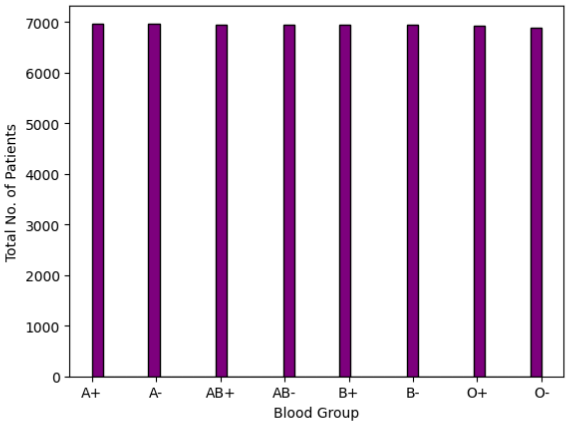
```
health['Blood Type'].value_counts()
```

Blood Type	count
A-	6969
A+	6956
AB+	6947
AB-	6945
B+	6945
B-	6944
O+	6917
O-	6877

dtype: int64

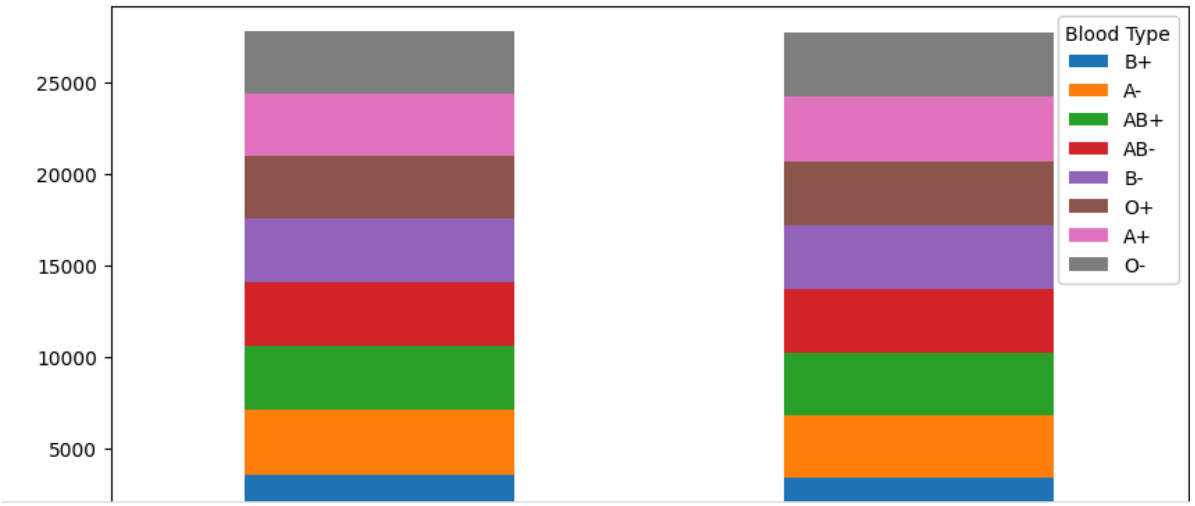
```
[50] data=health['Blood Type'].sort_values()
     plt.hist(data,bins=40,color='purple',edgecolor='black',rwidth=1.5)
     plt.xlabel('Blood Group')
     plt.ylabel('Total No. of Patients')
```

Connected to Python 3 Google Compute Engine backend




```
] Male=health[health['Gender']=='Male']['Blood Type'].value_counts()
  Female=health[health['Gender']=='Female']['Blood Type'].value_counts()
  df=pd.DataFrame([Male,Female])
  df.index=['Male','Female']
  df.plot(kind='bar',stacked=True,figsize=(10,5))
```

<Axes: >



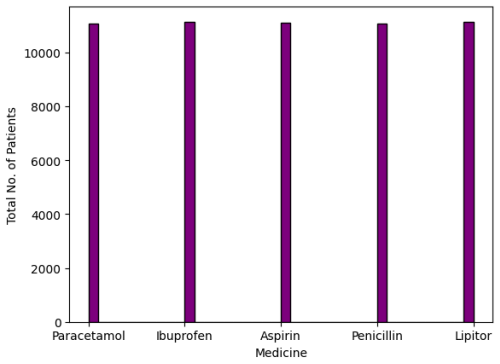
Connected to Python 3 Google Compute Engine backend

```
health.Medication.value_counts()
```

Medication	count
Lipitor	11140
Ibuprofen	11127
Aspirin	11094
Paracetamol	11071
Penicillin	11068

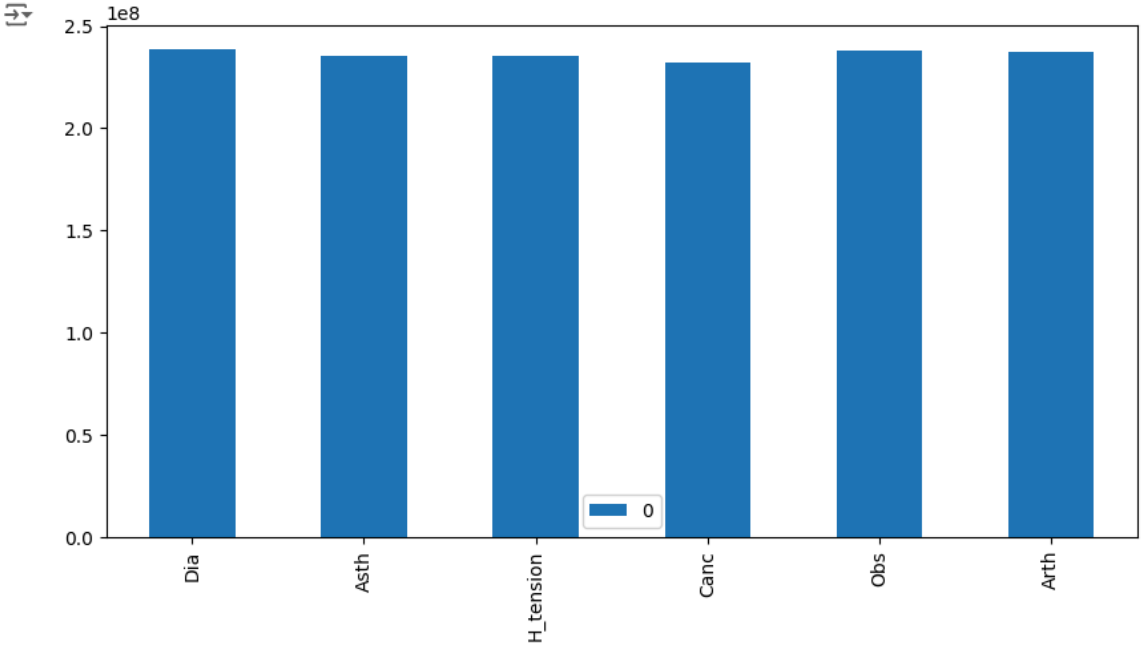
dtype: int64

```
data=health['Medication']
plt.hist(data,bins=40,color='purple',edgecolor='black',rwidth=1.5)
plt.xlabel('Medicine')
plt.ylabel('Total No. of Patients')
plt.show()
```



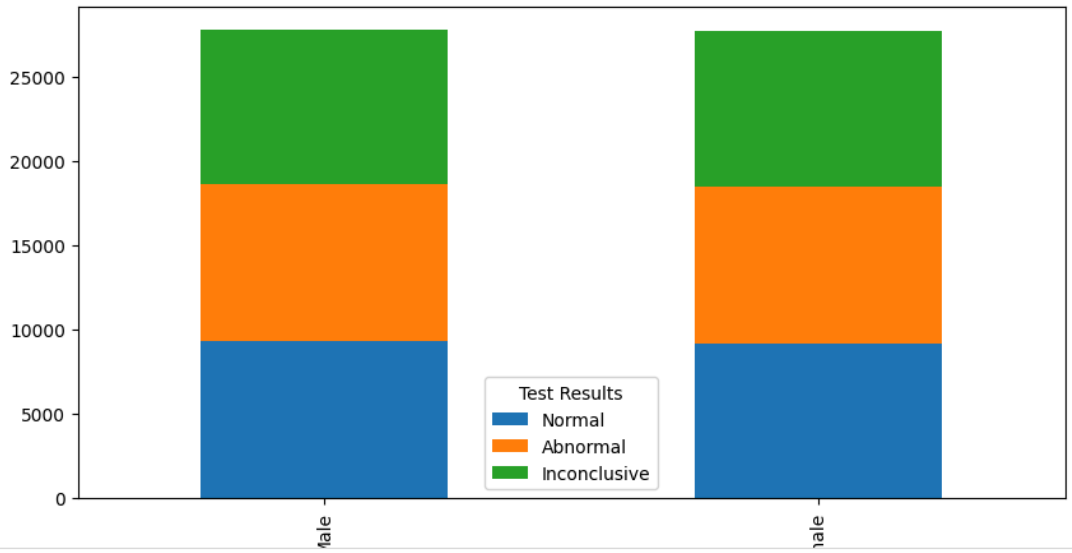

```
Dia=health[health['Medical Condition']=='Diabetes']['Billing Amount'].sum()
Asth=health[health['Medical Condition']=='Asthma']['Billing Amount'].sum()
H_tension=health[health['Medical Condition']=='Hypertension']['Billing Amount'].sum()
Canc=health[health['Medical Condition']=='Cancer']['Billing Amount'].sum()
Obs=health[health['Medical Condition']=='Obesity']['Billing Amount'].sum()
Arth=health[health['Medical Condition']=='Arthritis']['Billing Amount'].sum()

df=pd.DataFrame([Dia,Asth,H_tension,Canc,Obs,Arth])
df.index=['Dia','Asth','H_tension','Canc','Obs','Arth']
df.plot(kind='bar',figsize=(10,5))
```

[54] <Axes: >

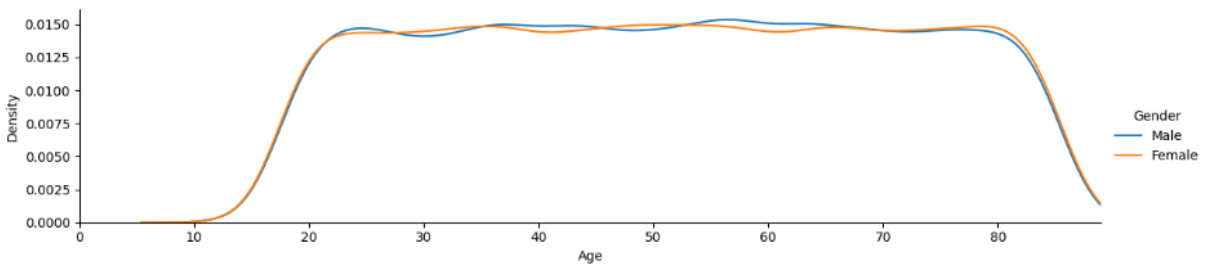


```
Male=health[health['Gender']=='Male']['Test Results'].value_counts()
Female=health[health['Gender']=='Female']['Test Results'].value_counts()
df=pd.DataFrame([Male,Female])
df.index=['Male','Female']
df.plot(kind='bar',stacked=True,figsize=(10,5))
```

```
<Axes: >
```




```
[58] facet=sns.FacetGrid(health, hue='Gender', aspect=4)
     facet.map(sns.kdeplot,'Age')
     facet.set(xlim=(0, health['Age'].max()))
     facet.add_legend()
     plt.show()
```



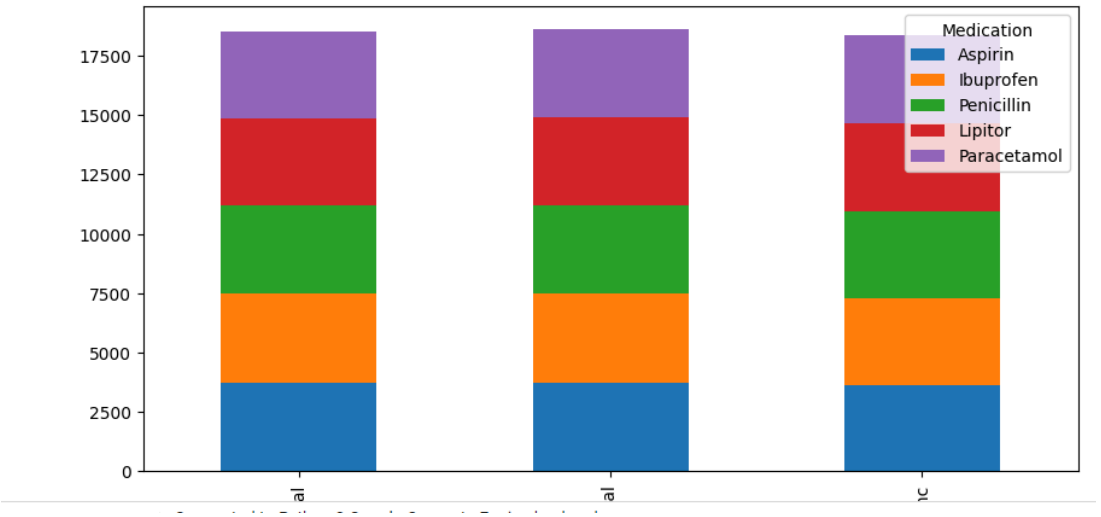

```
[59] health['Name'][2].upper()
```

```
'DANNY SMITH'
```

```
[60] health['Name']=health['Name'].str.upper()
```

```
[62] Normal=health[health['Test Results']=='Normal']['Medication'].value_counts()
     Anormal=health[health['Test Results']=='Abnormal']['Medication'].value_counts()
     Inconc=health[health['Test Results']=='Inconclusive']['Medication'].value_counts()
     ks=pd.DataFrame([Normal,Anormal,Inconc])
     ks.index=['Normal','A-normal','Inconc']
     ks.plot(kind='bar',stacked=True,figsize=(10,5))
```

```
<Axes: >
```

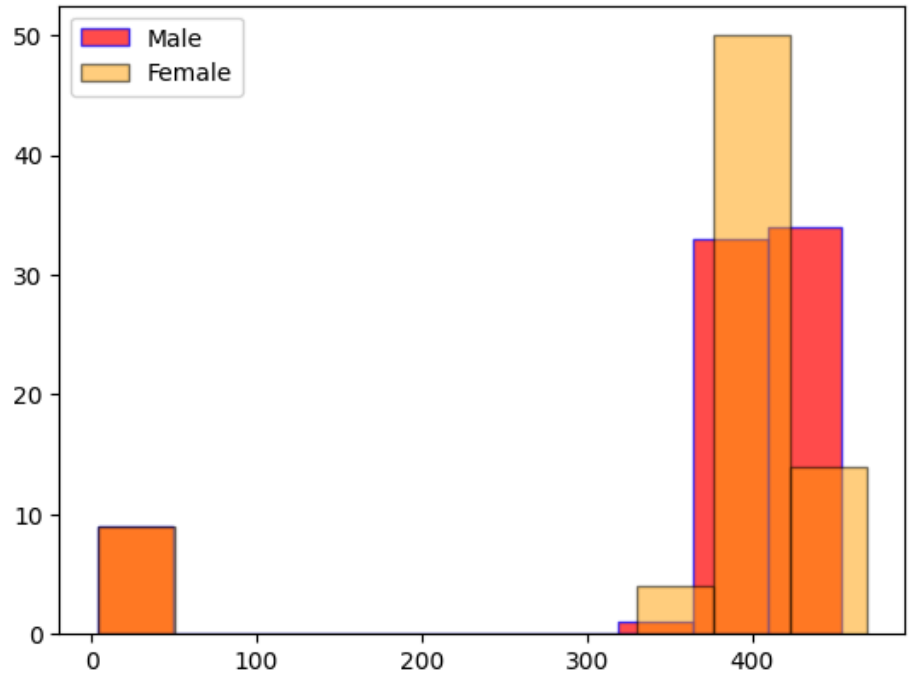



```
[63] lr1=['Age','Gender']
     health[lr1]
```

	Age	Gender
0	30	Male
1	62	Male
2	76	Female
3	28	Female
4	43	Female
...	...	...
55495	42	Female
55496	61	Female
55497	38	Female
55498	43	Male
55499	53	Female

55500 rows × 2 columns

```
g1=health[health['Gender']=='Male']['Age'].value_counts()
g2=health[health['Gender']=='Female']['Age'].value_counts()
plt.hist(g1, label='Male', alpha=.7, edgecolor='blue',color='red')
plt.hist(g2, label='Female', alpha=.5, edgecolor='black',color='orange')
plt.legend()
plt.show()
```




```
[65] map2={'Normal':0, 'Abnormal':1, 'Inconclusive':-1}

     health['Test Results']=health['Test Results'].map(map2)
```

```
[67] date=health['Date of Admission'][1]
     date
```

```
'2019-08-20'
```

```
[68] date=date[:4]
     date
```

```
'2019'
```

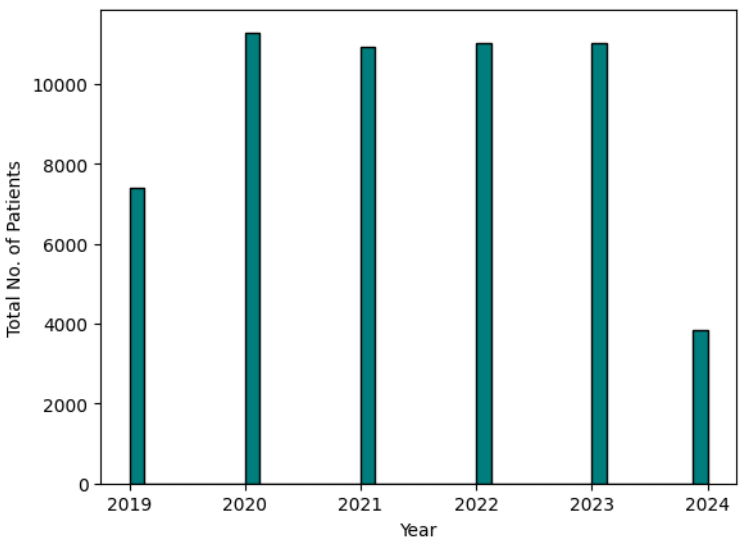
```
health['Year of Admission']=pd.DatetimeIndex(health['Date of Admission']).year
```

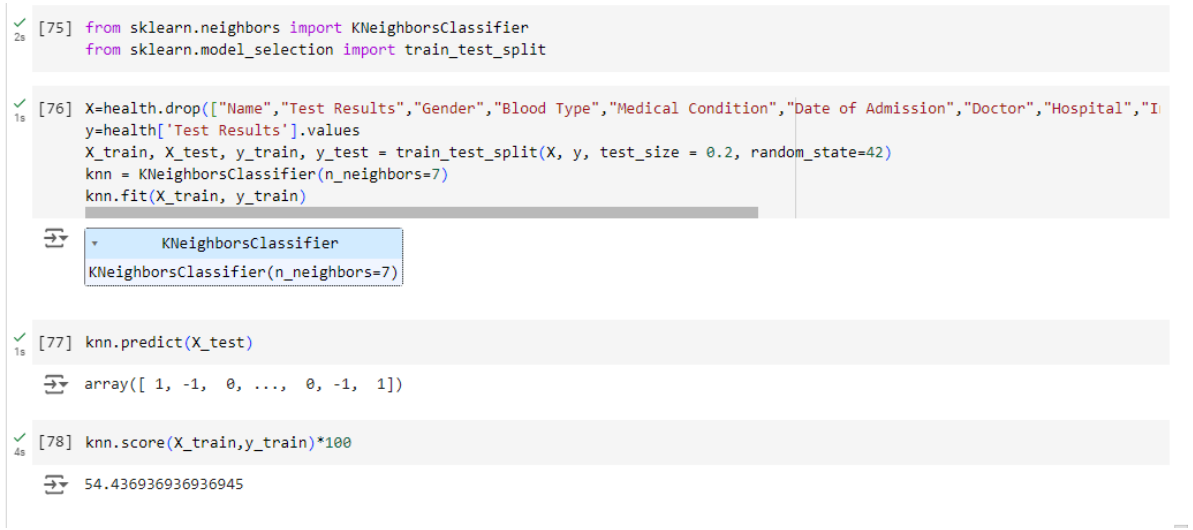
```
[72] health['Year of Admission'].value_counts()
```

Year of Admission	count
2020	11285
2023	11026
2022	11017
2021	10931
2019	7387
2024	3854

dtype: int64

```
data=health['Year of Admission'].sort_values()
plt.hist(data,bins=40,color='teal',edgecolor='black',rwidth=1.5)
plt.xlabel('Year')
plt.ylabel('Total No. of Patients')
plt.show()
```





```
[75] from sklearn.neighbors import KNeighborsClassifier
     from sklearn.model_selection import train_test_split

[76] X=health.drop(["Name","Test Results","Gender","Blood Type","Medical Condition","Date of Admission","Doctor","Hospital","I
     y=health['Test Results'].values
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state=42)
     knn = KNeighborsClassifier(n_neighbors=7)
     knn.fit(X_train, y_train)

     KNeighborsClassifier
     KNeighborsClassifier(n_neighbors=7)

[77] knn.predict(X_test)

     array([ 1, -1,  0, ...,  0, -1,  1])

[78] knn.score(X_train,y_train)*100

     54.436936936936945
```

Google Collaboratory Link: - https://colab.research.google.com/drive/1vXkDSBWIom997__WvpCpSS52IrG8U7uB?usp=sharing

Conclusion: -

Comment on the importance of EDA. After using your Healthcare related dataset, what observations did you make about the data?

Exploratory Data Analysis (EDA) is vital for understanding data before deeper analysis. It helps uncover patterns, detect anomalies, and validate assumptions, ensuring that subsequent analyses are accurate and reliable. From our healthcare dataset, key observations included variations in central tendency and spread of health metrics, revealing potential skewness and the need for further investigation. Distribution patterns showed some variables were not normally distributed, guiding appropriate statistical methods. Relationships between variables, like age and health conditions, were identified, aiding hypothesis development. Outliers and missing values were detected, highlighting areas for further scrutiny and informing data handling strategies. Overall, EDA provided essential insights, laying a solid foundation for more detailed analysis and decision-making.