Exploratory Data Analysis (EDA) – Types and Tools

-

Data analysis involves different processes of cleaning, transforming, analyzing the data, and building models to extract specific, relevant insights. These are beneficial for making important business decisions in real-time situations. Exploratory Data Analysis is important for any business. It lets data scientists analyze the data before reaching any conclusion. Also, this makes sure that the results which are out are valid and applicable to business outcomes and goals.

What is Exploratory Data Analysis (EDA)?

Exploratory Data Analysis (EDA) is one of the techniques used for extracting vital features and trends used by machine learning and deep learning models in Data Science. Thus, EDA has become an important milestone for anyone working in data science. This article covers the concept, meaning, tools, and techniques of EDA to give complete awareness to a beginner wanting to launch a career in data science. The article also enlists those fields that regularly apply EDA data analysis efficiently in promoting their business activities.

Why is EDA is Important in Data Science?

The Data Science field is now very important in the business world as it provides many opportunities to make vital business decisions by analyzing hugely gathered data. Understanding the data thoroughly needs its exploration from every aspect. The impactful features enable making meaningful and beneficial decisions; therefore, EDA occupies an invaluable place in Data science.

Steps Involved in Exploratory Data Analysis (EDA)

The key components in an EDA are the main steps undertaken to perform the EDA. These are as follows:

1. Data Collection

Nowadays, data is generated in huge volumes and various forms belonging to every sector of human life, like healthcare, sports, manufacturing, tourism, and so on. Every business knows the importance of using data beneficially by properly analyzing it. However, this depends on collecting the required data from various sources through surveys, social media, and customer reviews, to name a few. Without collecting sufficient and relevant data, further activities cannot begin.

2. Finding all Variables and Understanding Them

When the analysis process starts, the first focus is on the available data that gives a lot of information. This information contains changing values about various features or characteristics, which helps to understand and get valuable insights from them. It requires first identifying the important variables which affect the outcome and their possible impact. This step is crucial for the final result expected from any analysis.

3. Cleaning the Dataset

The next step is to clean the data set, which may contain null values and irrelevant information. These are to be removed so that data contains only those values that are relevant and important from the target point of view. This will not only reduce time but also reduces the computational power from an estimation point of view. Preprocessing takes care of all issues, such as identifying null values, outliers, anomaly detection, etc.

4. Identify Correlated Variables

Finding a correlation between variables helps to know how a particular variable is related to another. The correlation matrix method gives a clear picture of how different variables correlate, which further helps in understanding vital relationships among them.

5. Choosing the Right Statistical Methods

As will be seen in later sections, depending on the data, categorical or numerical, the size, type of variables, and the purpose of analysis, different statistical tools are employed. Statistical formulae applied for numerical outputs give fair information, but graphical visuals are more appealing and easier to interpret.

6. Visualizing and Analyzing Results

Once the analysis is over, the findings are to be observed cautiously and carefully so that proper interpretation can be made. The trends in the spread of data and correlation between variables give good insights for making suitable changes in the data parameters. The data analyst should have the requisite capability to analyze and be well-versed in all analysis techniques. The results obtained will be appropriate to data of that particular domain and are suitable for use in retail, healthcare, and agriculture.

Aspiring data science professionals must understand and practice the above EDA data science steps to master Python exploratory data analysis. Explore the **Data Science Bootcamp curriculum** to know more.

Types of Exploratory Data Analysis

There are four main types of EDA:

1. Univariate non-graphical
2. Univariate graphical
3. Multivariate nongraphical
4. Multivariate graphical

In univariate analysis, the output is a single variable and all data collected is for it. There is no cause-and-effect relationship at all. For example, data shows products produced each month for twelve months. In bivariate analysis, the outcome is dependent on two variables, e.g., the age of an employee, while the relation with it is compared with two variables, i.e., his salary earned and expenses per month.

In multivariate analysis, the outcome is more than two, e.g., type of product and quantity sold against the product price, advertising expenses, and discounts offered. The analysis of data is done on variables that can be numerical or categorical. The result of the analysis can be represented in numerical values, visualization, or graphical form. Accordingly, they could be further classified as non-graphical or graphical.

1. Univariate Non-Graphical

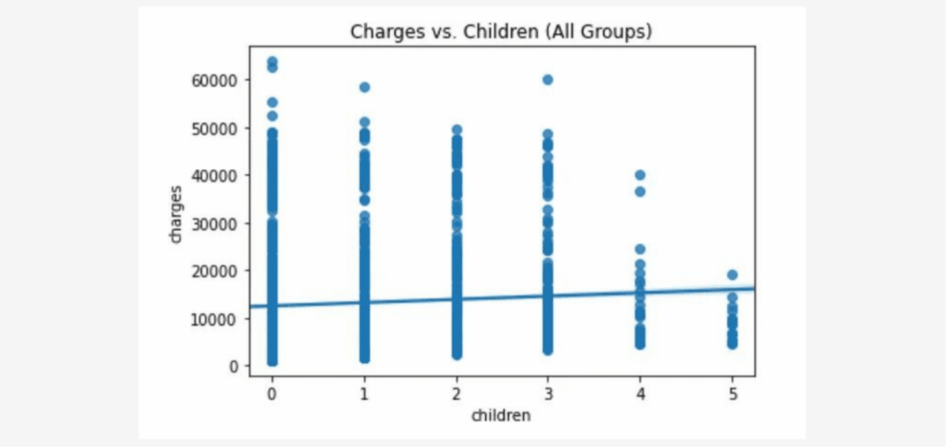
It is the simplest of all types of data analysis used in practice. As the name suggests, uni means only one variable is considered whose data (referred to as population) is compiled and studied. The main aim of univariate non-graphical EDA is to find out the details about the distribution of the population data and to know some specific parameters of statistics. The significant parameters which are estimated from a distribution point of view are as follows:

- **Central Tendency:** This term refers to values located at the data's central position or middle zone. The three generally estimated parameters of central tendency are mean, median, and mode. Mean is the average of all values in data, while the mode is the value that occurs the maximum number of times. The Median is the middle value with equal observations to its left and right.
- **Range:** The range is the difference between the maximum and minimum value in the data, thus indicating how much the data is away from the central value on the higher and lower side.
- **Variance and Standard Deviation:** Two more useful parameters are standard deviation and variance. Variance is a measure of dispersion that indicates the spread of all data points in a data set. It is the measure of dispersion mostly used and is the mean squared difference between each data point and mean, while standard deviation is the square root value of it. The larger the value of standard deviation, the farther the spread of data, while a low value indicates more values clustering near the mean.

2. Univariate Graphical

The graphs in this section are based on **Auto MPG dataset** available on the UCI repository. Some common types of univariate graphics are:

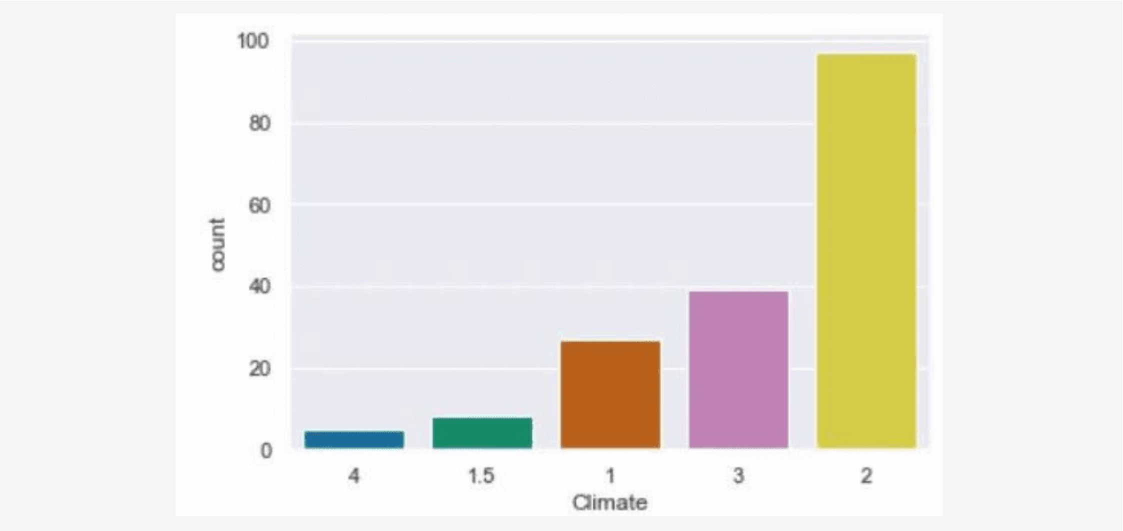
A. Stem-and-leaf Plots: This is a very simple but powerful EDA method used to display quantitative data but in a shortened format. It displays the values in the data set, keeping each observation intact but separating them as stem (the leading digits) and remaining or trailing digits as leaves. But histogram is mostly used in its place now.



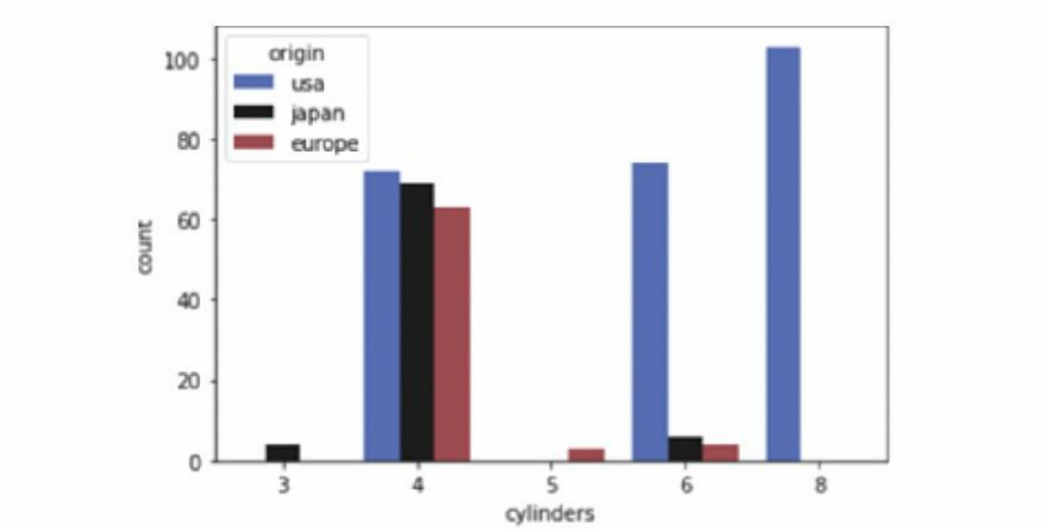
B. Histograms (Bar Charts): These plots are used to display both grouped or ungrouped data. On the x-axis, values of variables are plotted, while on the y-axis are the number of observations or frequencies. Histograms are very simple to quickly understand your data, which tell about values of data like central tendency, dispersion, outliers, etc. The simplest fundamental graph is a histogram, which is a bar plot with each bar representing the frequency, i.e., the count or proportion (the ratio of count to the total count of occurrences) for various values.

There are many types of histograms, a few of which are listed below:

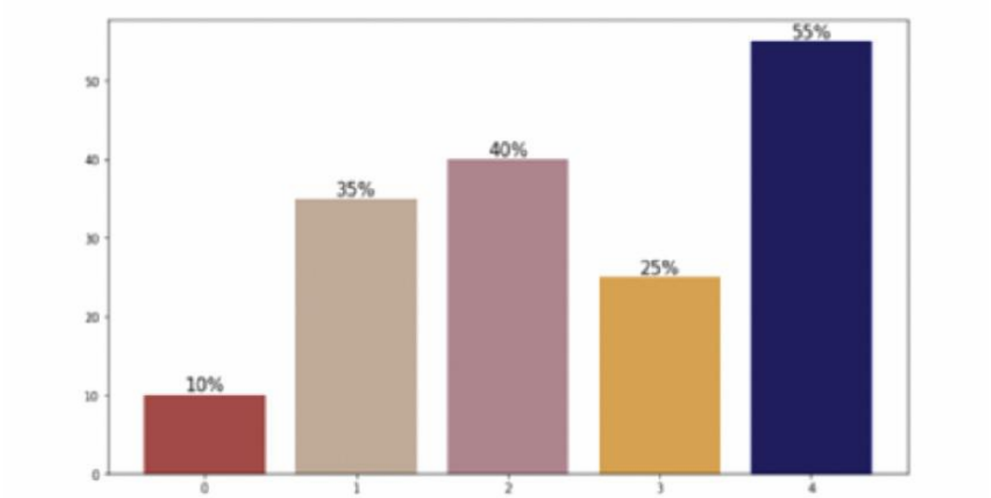
1. **Simple Bar Charts:** These are used to represent categorical variables with rectangular bars, where the different lengths correspond to the values of the variables.



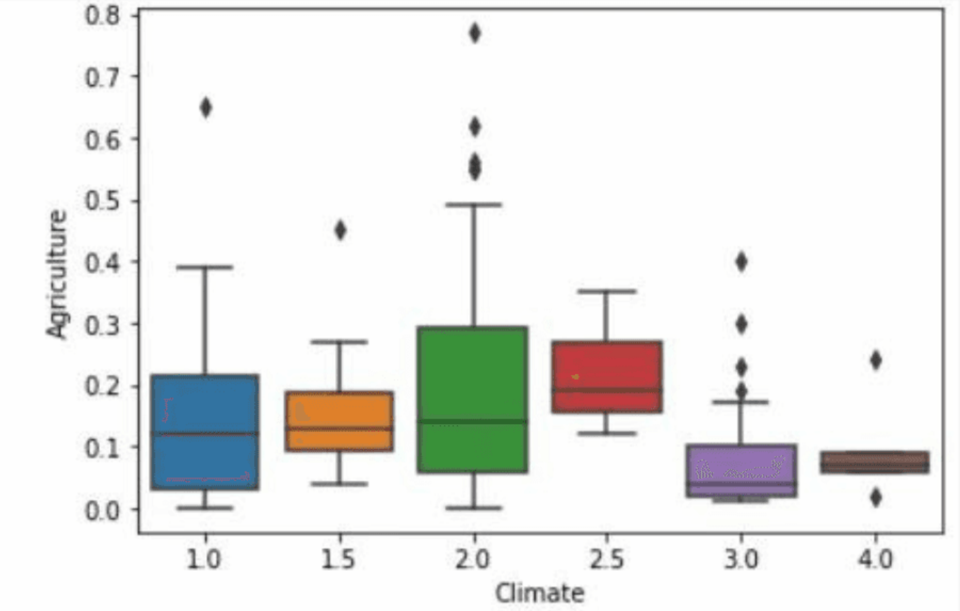
1. **Multiple or Grouped charts:** Grouped bar charts are bar charts representing multiple sets of data items for comparison where a single color is used to denote one specific series in the dataset.



2. **Percentage Bar Charts:** These are bar graphs that depict the data in the form of percentages for each observation. The following image shows a percentage bar chart with dummy values.



3. **Box Plots:** These are used to display the distribution of quantitative value in the data. If the data set consists of categorical variables, the plots can show the comparison between them. Further, if outliers are present in the data, they can be easily identified. These graphs are very useful when comparisons are to be shown in percentages, like values in the 25 %, 50 %, and 75% range (quartiles).



3. Multivariate Non-Graphical

The multivariate non-graphical exploratory data analysis technique is usually used to show the connection between two or more variables with the help of either cross-tabulation or statistics.

- For categorical data, an extension of tabulation called cross-tabulation is extremely useful. For two 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 can generate statistical information for quantitative variables separately for every level of the specific variable. We then compare the statistics across the number of categorical variables.

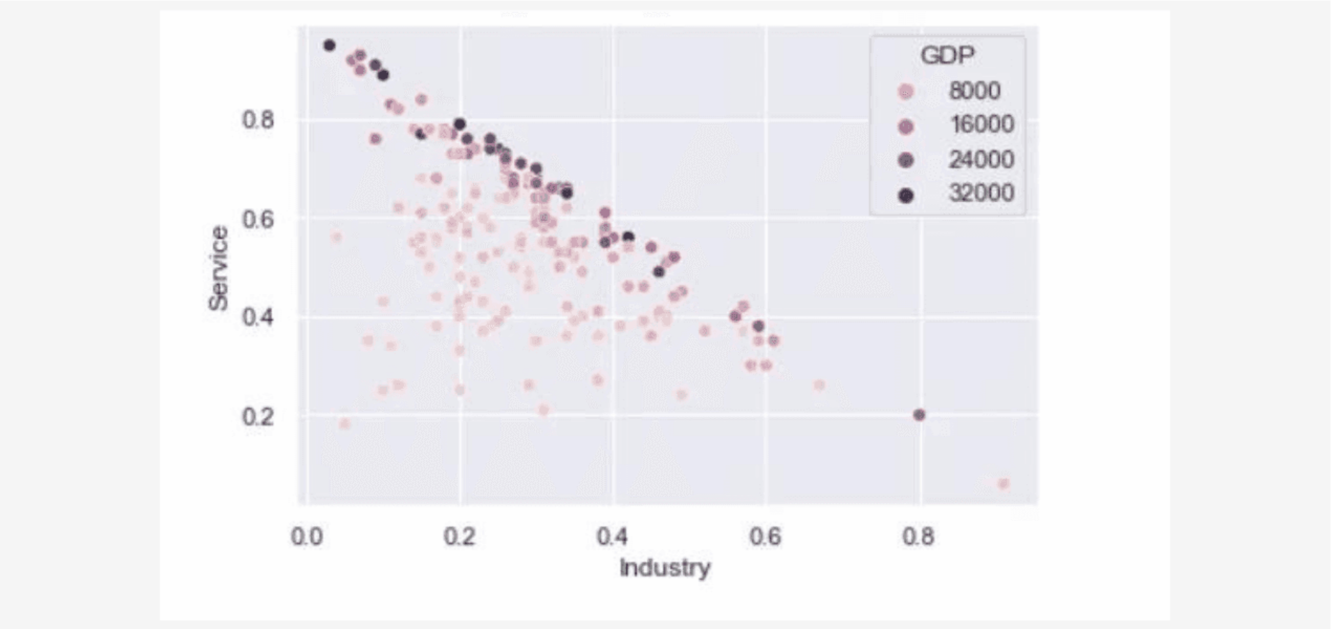
4. Multivariate Graphical

Graphics are used in multivariate graphical data to show the relationships between two or more variables. Here the outcome depends on more than two variables, while the change-causing variables can also be multiple.

Some common types of multivariate graphics include:

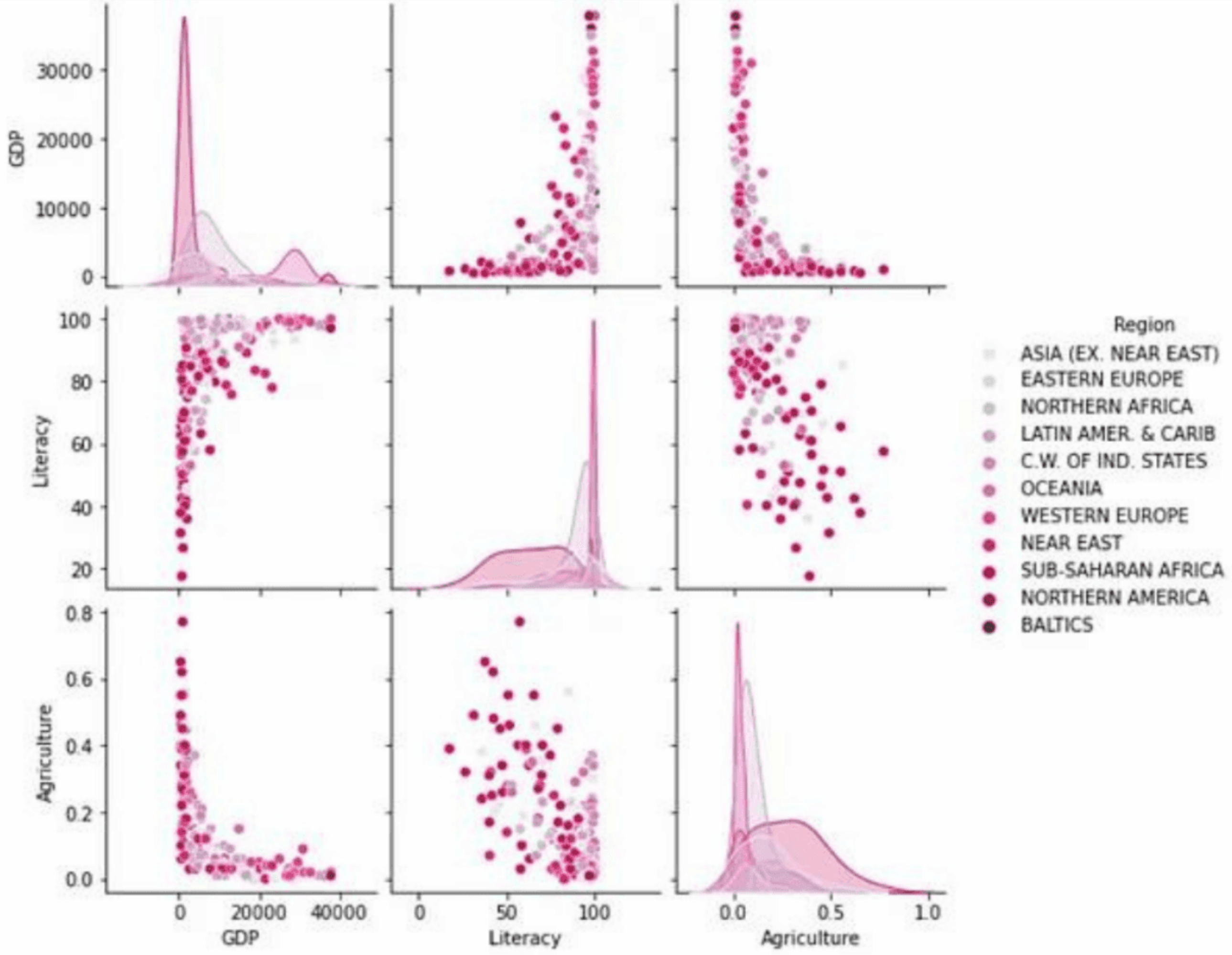
A. Scatter Plot

The essential graphical EDA technique for two quantitative variables is the scatter plot, so one variable appears on the x-axis and the other on the y-axis and, therefore, the point for every case in your dataset. This can be used for bivariate analysis.



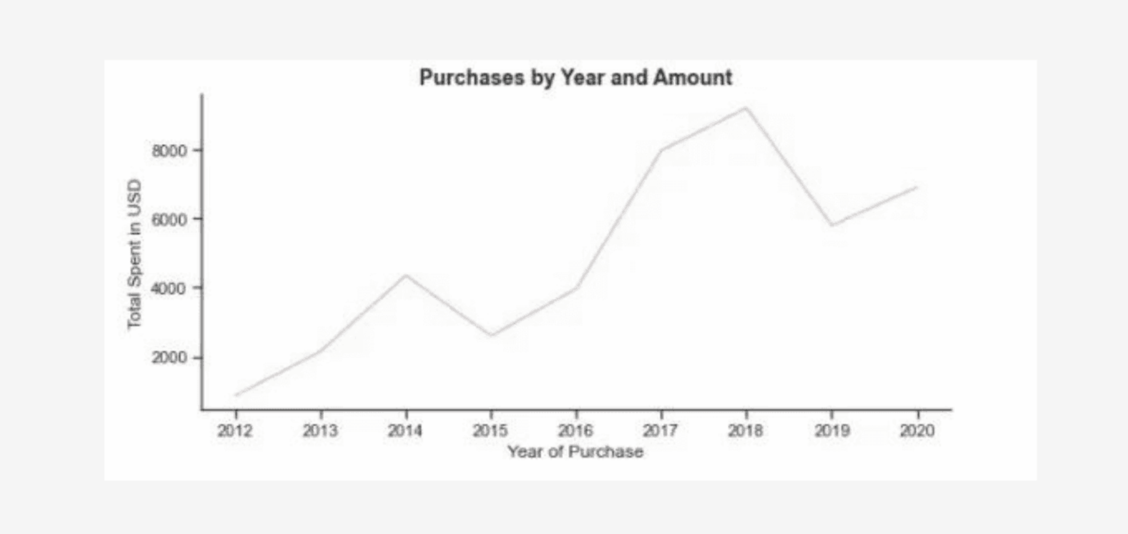
B. Multivariate Chart

A Multivariate chart is a type of control chart used to monitor two or more interrelated process variables. This is beneficial in situations such as process control, where engineers are likely to benefit from using multivariate charts. These charts allow monitoring multiple parameters together in a single chart. A notable advantage of using multivariate charts is that they help minimize the total number of control charts for organizational processes. Pair plots generated using the Seaborn library are a good example of multivariate charts as they help visualize the relationships between all numerical variables in the entire dataset at once.



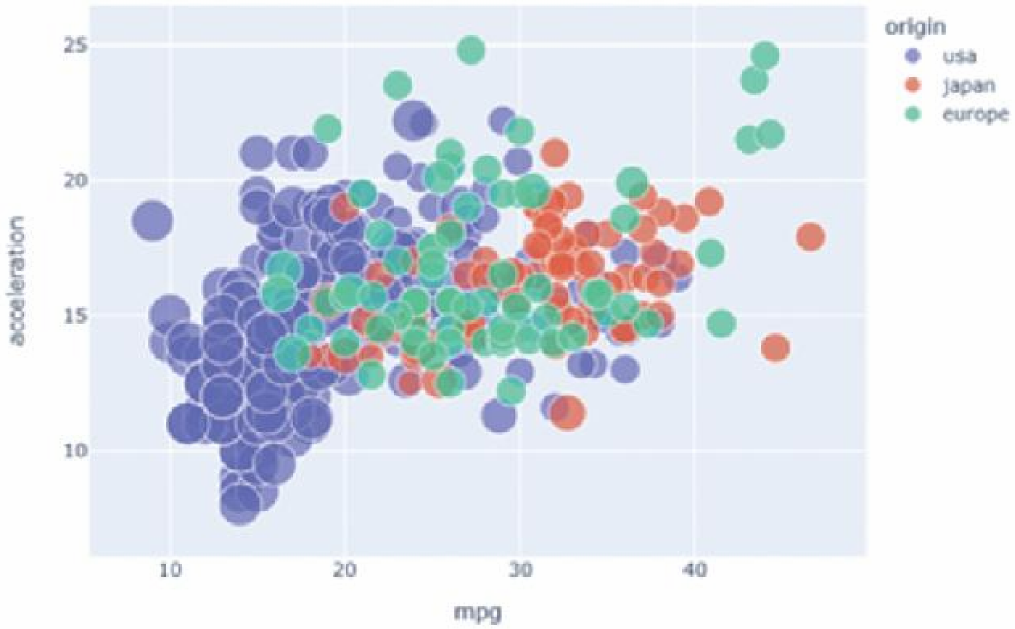
C. Run Chart

A **run chart** is a data line chart drawn over time. In other words, a run chart visually illustrates the process performance or data values in a time sequence. Rather than summary statistics, seeing data across time yields a more accurate conclusion. A trend chart or time series plot is another name for a run chart. The plot below depicts dummy values of sales over a period of time.



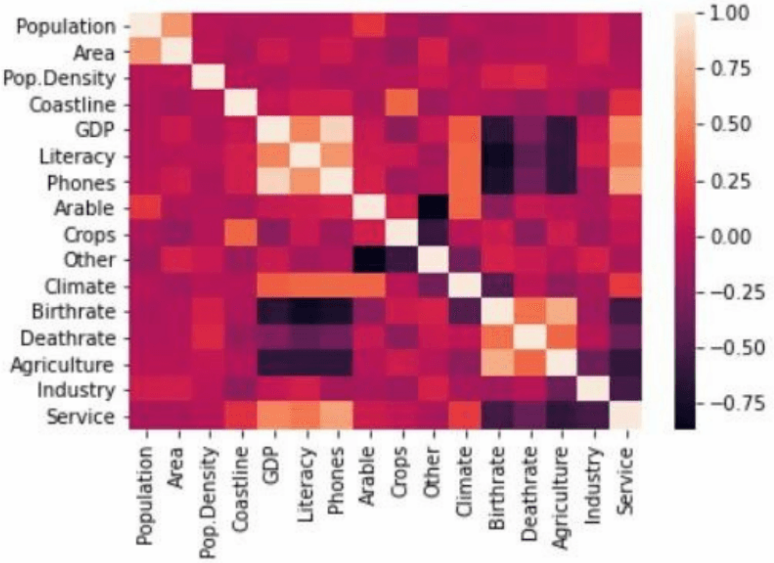
D. Bubble Chart

Bubble charts scatter plots that display multiple circles (bubbles) in a two-dimensional plot. These are used to assess the relationships between three or more numeric variables. In a bubble chart, every single dot corresponds to one data point, and the values of the variables for each point are indicated by different positions such as horizontal, vertical, dot size, and dot colors.



E. Heat Map

A heat map is a colored graphical representation of multivariate data structured as a matrix of columns and rows. The heat map transforms the correlation matrix into color coding and represents these coefficients to visualize the strength of correlation among variables. It assists in finding the best features suitable for building accurate Machine Learning models.



Apart from the above, there is also the ‘Classification or Clustering analysis’ technique used in EDA. It is an unsupervised type of machine learning used for the classification of input data into specified categories or clusters exhibiting similar characteristics in various groups. This can be further used to draw important interpretations in EDA.

Tools to Perform Exploratory Data Analysis

1. Python

Python is used for different tasks in EDA, such as finding missing values in data collection, data description, handling outliers, obtaining insights through charts, etc. The syntax for EDA libraries like:

- Matplotlib
- Pandas
- Seaborn
- NumPy
- Altair

and more in Python is fairly simple and easy to use for beginners. You can find many open-source packages in Python, such as D-Tale, AutoViz, PandasProfiling, etc., that can automate the entire exploratory data analysis process and save time.

2. R

R programming language is a regularly used option to make statistical observations and analyze data, i.e., perform detailed EDA by data scientists and statisticians. Like Python, R is also an open-source programming language suitable for statistical computing and graphics. Apart from the commonly used libraries like:

- ggplot
- Leaflet
- Lattice

there are several powerful R libraries for automated EDA, such as Data Explorer, SmartEDA, GGally, etc.

3. MATLAB

MATLAB is a well-known commercial tool among engineers since it has a very strong mathematical calculation ability. Due to this, it is possible to use MATLAB for EDA, but it requires some basic knowledge of the MATLAB programming language.

Advantages of Using EDA

Here are a few advantages of using Exploratory Data Analysis -

- **Gain Insights Into Underlying Trends and Patterns:** EDA assists data analysts in identifying crucial trends quickly through data visualizations using various graphs, such as box plots and histograms. Businesses also expect to make some unexpected discoveries in the data while performing EDA, which can help improve certain existing business strategies.
- **Improved Understanding of Variables:** Data analysts can significantly improve their comprehension of many factors related to the dataset. Using EDA, they can extract various information such as averages, means, minimum and maximum, and more such information is required for preprocessing the data appropriately.
- **Better Preprocess Data to Save Time:** EDA can assist data analysts in identifying significant mistakes, abnormalities, or missing values in the existing dataset. Handling the above entities is critical for any organization before beginning a full study as it ensures correct preprocessing of data and may help save a significant amount of time by avoiding mistakes later when applying machine learning models.
- **Make Data-driven Decisions:** The most significant advantage of employing EDA in an organization is that it helps businesses to improve their understanding of data. With EDA in machine learning, they can use the available tools to extract critical insights and make conclusions, which assist in making decisions based on the insights from the EDA.

Exploratory Data Analysis Examples

Example 1: EDA in Health Care Research

Let us perform Exploratory Data Analysis on a healthcare dataset using Python. The dataset used for this example is Stroke Prediction Dataset from Kaggle. We start by importing all necessary libraries for performing EDA.

```
import NumPy as np

import pandas as pd

import seaborn as sns
```

Copy Code

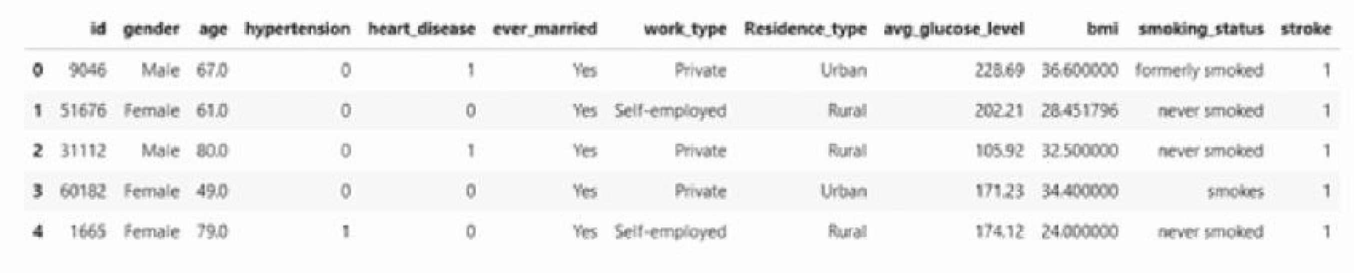
Then, we can read in the data as a pandas data frame.

```
df=pd.read_csv("../input/stroke-prediction-
dataset/healthcare-dataset-stroke-data.csv")
```

Copy Code

The dataset contains 5110 individuals' data with 12 features. It has different features like - id, gender, age, hypertension, heart disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, smoking_status, and stroke. Using the df.head() command, we can print the first five rows of the dataset.

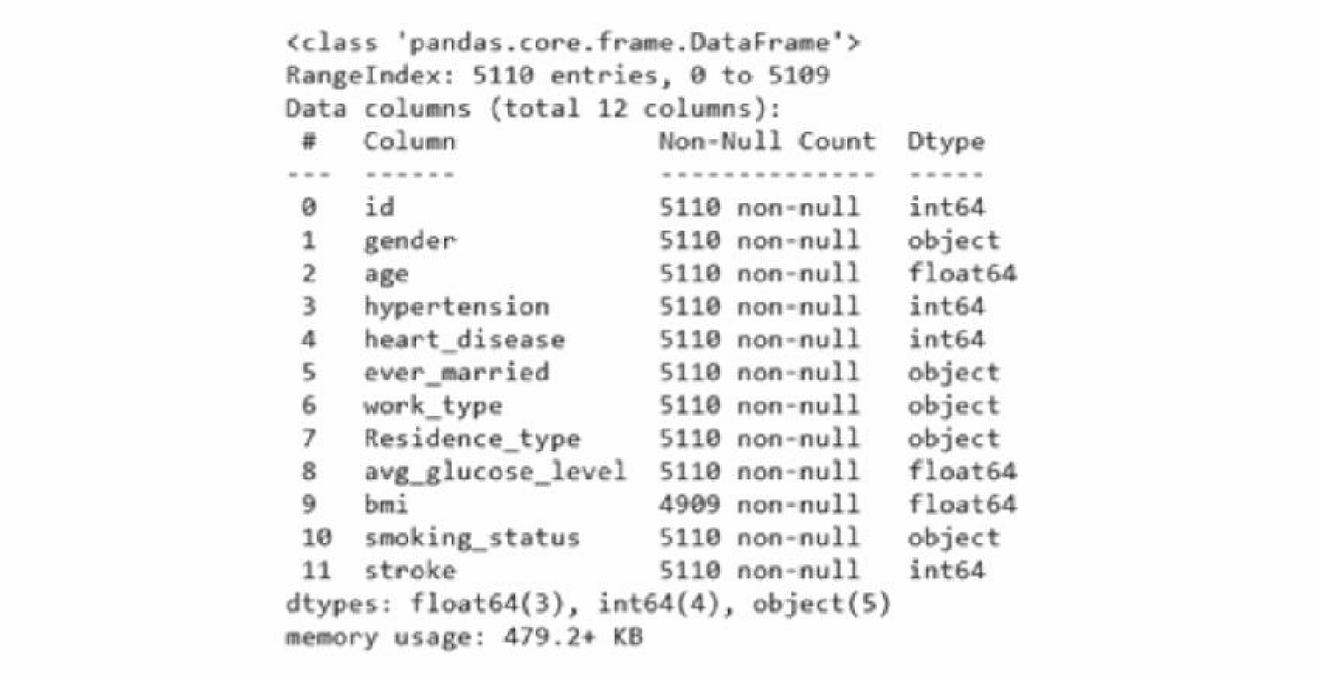
```
df.head()
```



	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.600000	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	28.451796	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.500000	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.400000	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.000000	never smoked	1

We can now conduct the EDA after importing the dataset. To get the basic information on the dataset, i.e., the number of null values, data types, and memory utilization, run the info() command:

```
df.info()
```

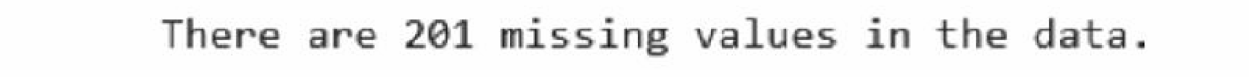


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 5110 non-null   int64  
 1   gender             5110 non-null   object 
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64  
 4   heart_disease      5110 non-null   int64  
 5   ever_married       5110 non-null   object 
 6   work_type          5110 non-null   object 
 7   Residence_type     5110 non-null   object 
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object 
 11  stroke             5110 non-null   int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

The above output shows that the attribute 'id' contains 5110 unique values, attributes (age, bmi, avg glucose level) are numerical. In contrast, attributes (gender, hypertension, heart disease, ever married, work type, Residence type, smoking status, stroke) are categorical.

First, we will find the missing values. To print the exact number of missing values, run the following command:

```
print("There are {} missing values in the data.".format(df.isna().sum().sum()))
```



```
There are 201 missing values in the data.
```

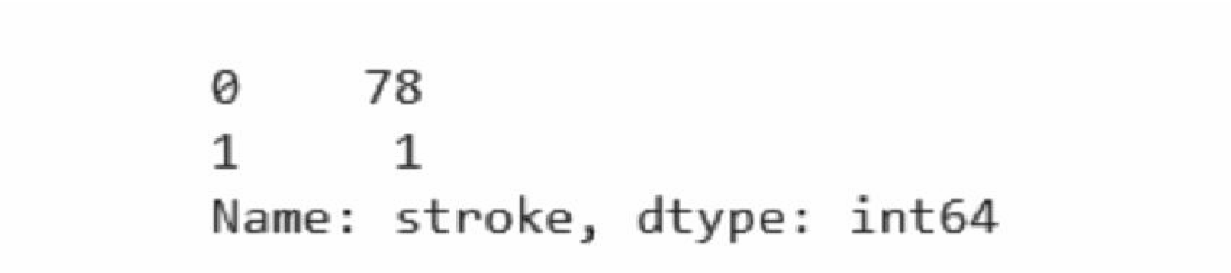
As missing values affect the outcome during analysis, we will replace the null values of bmi with the mean of the bmi column and check again to ensure that all missing values in the dataset have been correctly replaced.

```
df.bmi.replace(to_replace=np.nan,value=df.bmi.mean(), inplace=True)
```

```
print("There are {} missing values in
the data.".format(df.isna().sum().sum()))
```

It's important to search for outliers in the bmi variable and identify how many of them have outcome associated with it.

```
bmi_outliers=df[df['bmi']>50]

bmi_outliers['bmi'].shape
```



```
0    78
1     1
Name: stroke, dtype: int64
```

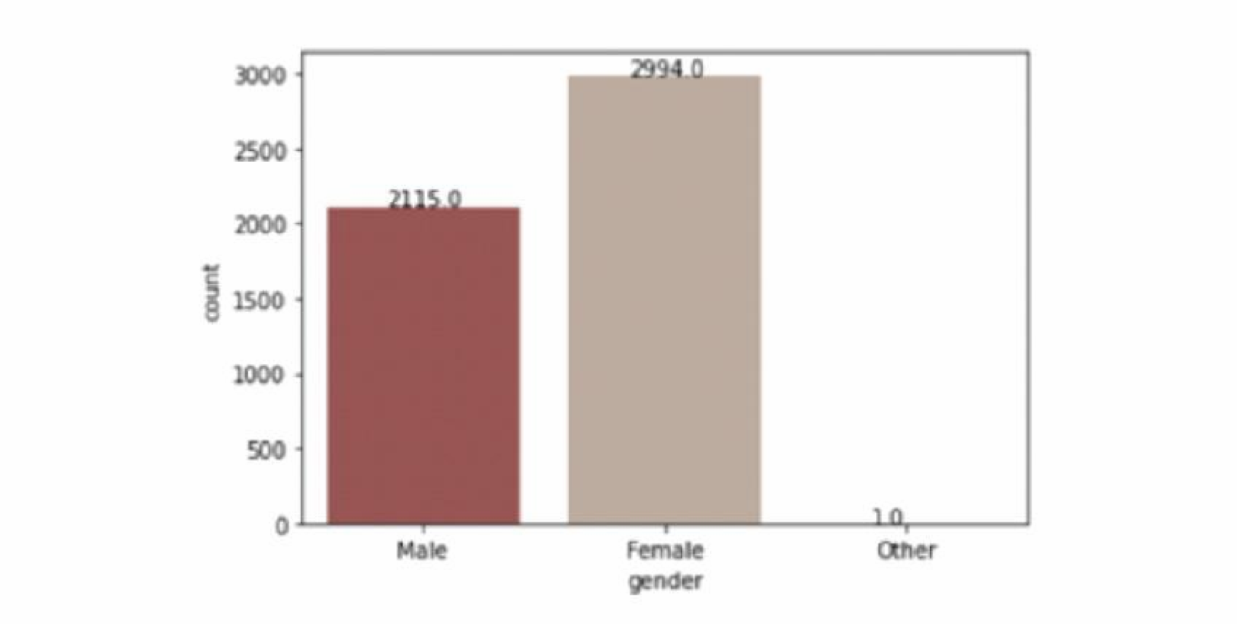
The dataset has 79 outliers in total. However, there is only one value that has the possibility of getting a stroke. Here, we will replace the bmi outliers with the mean.

```
df["bmi"] = df["bmi"].apply(lambda x: df.bmi.mean() if
x>50 else x)
```

Copy Code

We will use the matplotlib library to visualize the relationship between different variables in our dataset. The analysis is shown below with a few graphical visualizations and their interpretation.

Bar Chart (Gender Distribution)



Looking at the gender distribution in the dataset indicates that 59% of females and 41% of men have one value labeled as 'other.' To simplify the data, we can transform this single variable to male.

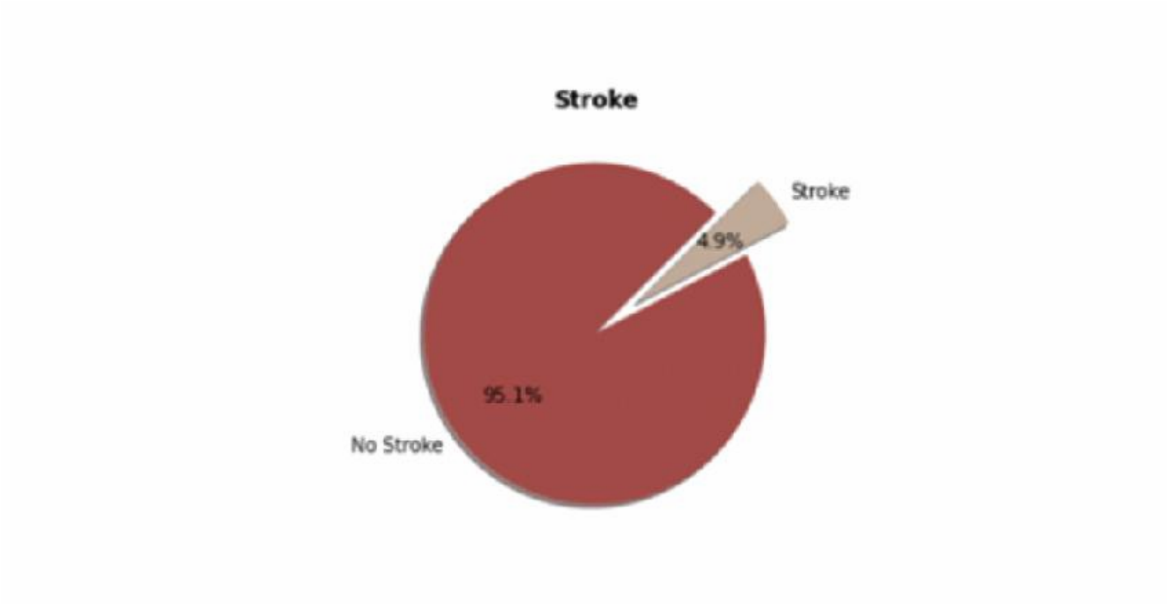
```
df['gender']=df['gender'].replace('Other','Male')
```

Copy Code

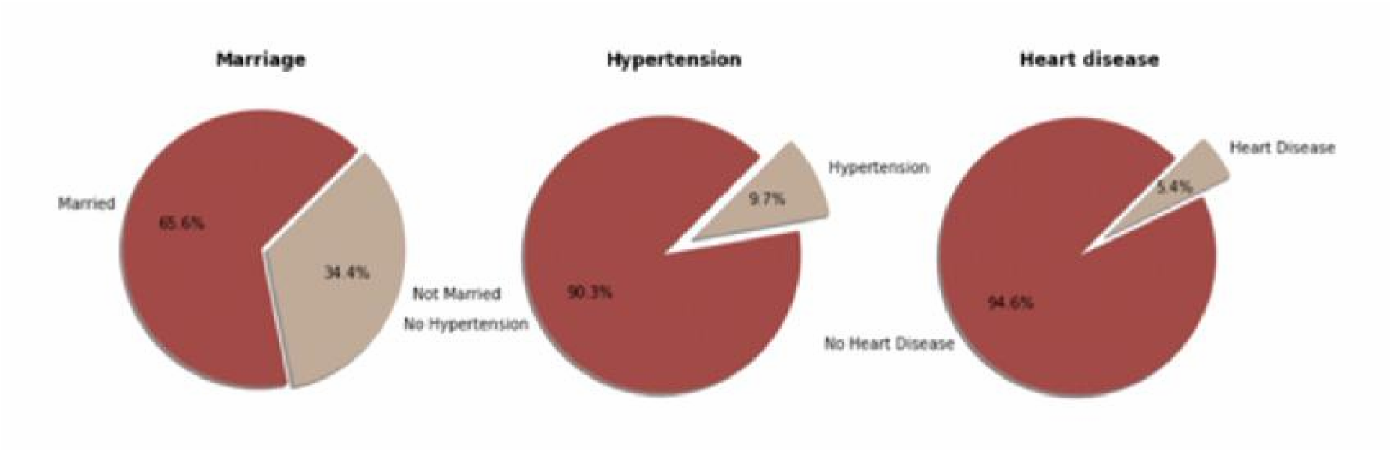
Now that the outliers and missing values have been properly set up, it's time to create some more graphs from the data to discover additional information. Let us create a pie chart for the dataset's outcome distribution ‘stroke’. We can use the following code -

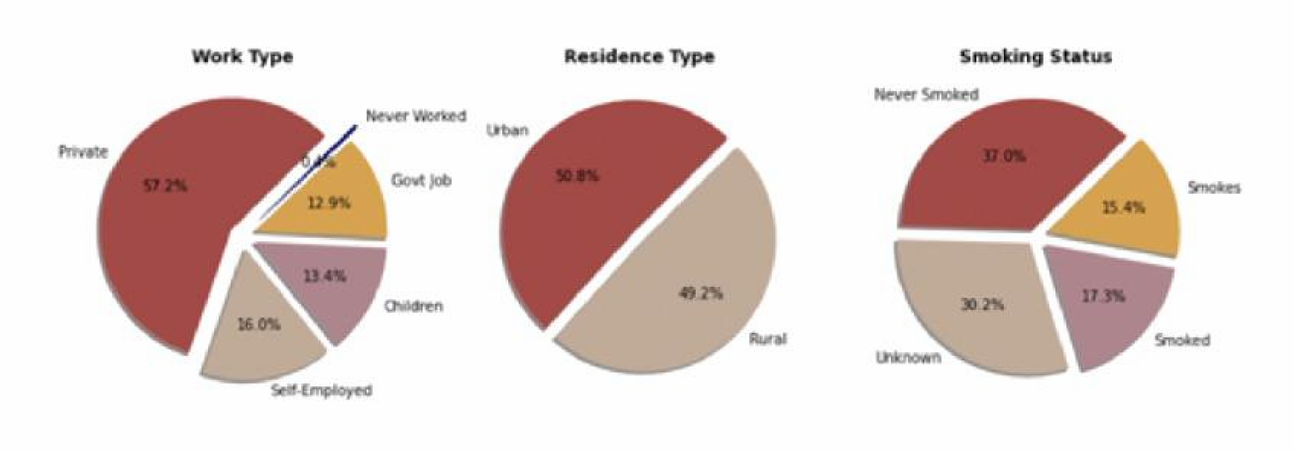
```
fig, ax = plt.subplots(1,1, figsize = (6,6))

labels = ["No Stroke", "Stroke"]

values = df['stroke'].value_counts().tolist()

ax.pie(x=values, labels=labels, autopct="%1.1f%%",

shadow=True, startangle=45,explode=[0,0]
```

```
ax.set_title("Stroke", fontdict={'fontsize':
12},fontweight ='bold')
```



Similarly, we can plot pie charts for other variables in the dataset.

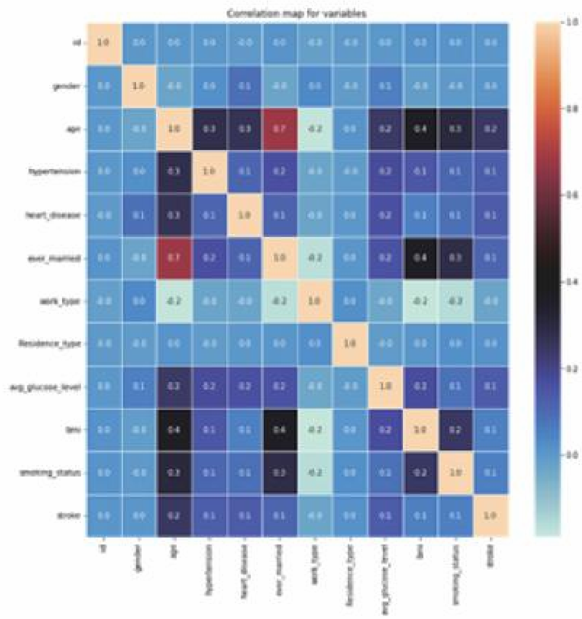




All of these charts reveal a lot of important information about the dataset, such as:

- Only 5% of people are at risk of having a stroke.
- Less than 10% of people have hypertension.
- A bit more than 5% of people have heart disease.
- The dataset has an equal distribution of residence types, with 50% of the population coming from rural regions.
- Over 65% of individuals are married, and 57% work in the private sector.
- 37% of people don't smoke at all.

Next, we will plot a correlation matrix that gives us a general understanding of the correlations between the input and the target variables.



Example 2: EDA in Retail

In the retail industry, EDA can be performed on a dataset consisting of various columns such as product categories, sales, price, discounts, region of sales, orders, etc., for understanding sales patterns, improving inventory management, predicting future demands, etc. You can follow the steps mentioned in the previous example to practice EDA for a Superstore Sales Dataset available on Kaggle.

Example 3: EDA in Electronic Medical Records

An important aspect for organizations in the healthcare domain is maintaining electronic medical records. These are digital records of the medical history of the visiting patients, such as any previous hospitalization, administered medicines, allergies or vaccinations, etc. You can explore the UCI repository **Diabetes 130-US hospitals for years 1999-2008 Data Set** for practicing EDA on similar lines as given in the previous example.

Objective of Exploratory Data Analysis (EDA)

The overall objective of exploratory data analysis is to obtain vital insights and hence usually includes the following sub-objectives:

- Identifying and removing data outliers
- Identifying trends in time and space
- Uncover patterns related to the target
- Creating hypotheses and testing them through experiments
- Identifying new sources of data

Role of EDA in Data Science

The role of data exploration analysis is based on the use of objectives achieved as above. After formatting the data, the performed analysis indicates patterns and trends that help to take the proper actions required to meet the expected goals of the business. As we expect specific tasks to be done by any executive in a particular job position, it is expected that proper EDA will fully provide answers to queries related to a particular business decision. As data science involves building models for prediction, they require optimum data features to be considered by the model. Thus, EDA ensures that the correct ingredients in patterns and trends are made available for training the model to achieve the correct outcome, like a successful recipe. Therefore, carrying out the right EDA with the correct tool based on befitting data will help achieve the expected goal.

Some key takeaways from this article are:

- EDA is subjective as it summarizes the features and characteristics of a dataset. So, depending on the project, data scientists can choose from the various plots discussed in this article to explore the data before applying machine learning algorithms.
- Since the nature of EDA depends on the data, we can say that it is an approach instead of a defined process.
- EDA presents hidden insights from data through visualizations such as graphs and plots.
- Graphical and non-graphical statistical methods can be used to perform EDA.
- Univariate analysis is simpler than multivariate analysis.
- The success of any EDA will depend on the quality and quantity of data, the choice of tools and visualization, and its proper interpretation by a data scientist.
- EDA is crucial in AI-driven businesses such as retail, e-commerce, banking and finance, agriculture, healthcare, and so on.

Frequently Asked Questions (FAQs)

1. What are the underlying principles of exploratory data analysis?

The main underlying principles of an EDA are-

- The aim should be to uncover information that should lead to showing patterns and trends.
- Missing values and outliers need to be given proper consideration
- The relationship between different variables must be established.
- A suitable technique of variate analysis should be chosen for the target to be achieved.

2. How can I master EDA for data science?

This article gives a comprehensive overview of EDA with its tools, types, and processes for a beginner. It is necessary to understand why EDA is performed and how it is beneficial to make decisions. Once you understand the basics, you can register for a data science course online, like **KnowledgeHut online Data Science certificate** to master EDA for data science. Additionally, you can keep exploring a variety of datasets on Kaggle.

3. How do you write an EDA report?

An EDA report must thoroughly explain the dataset's variables, their correlation, and any preprocessing performed on the dataset to make it suitable for applying a machine learning algorithm for further use in the organization. This report consists of multiple relevant visualizations that present a complete picture of the information in the available data and hence, can be used by senior management to make data-driven decisions.

4. Is EDA and data analysis the same?

Data analysis is a broad term involving different types of analysis like descriptive, diagnostic, predictive, and prescriptive. EDA is synonymous with descriptive analysis, where one explores the hidden relationships and patterns in the available data.