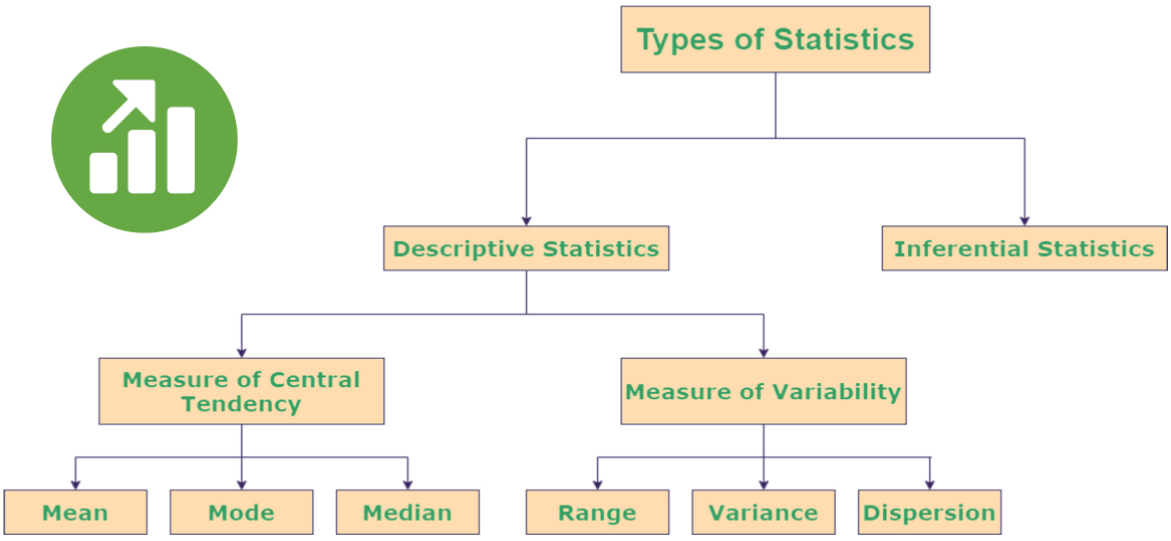
Lesson Title: **Introduction to Basic Data Statistics and Exploratory Data Analysis (EDA)**

Basic data statistics and Exploratory Data Analysis are foundational techniques that provide valuable insights into a dataset's characteristics and guide subsequent analysis steps. They are essential tools for data scientists, analysts, and researchers aiming to extract meaningful information from data.

Data analysis is a crucial step in the process of deriving insights and making informed decisions from data. Basic data statistics and Exploratory Data Analysis (EDA) are fundamental techniques that help us understand the characteristics, patterns, and trends present in a dataset. These techniques are often the first steps taken when starting to work with a new dataset.

Data Statistics

Data statistics, also known as descriptive statistics, are numerical and graphical measures used to summarize, describe, and interpret the main characteristics of a dataset. These statistics provide a concise overview of the data's distribution, central tendency, variability, and other key attributes.



Descriptive Statistics -Descriptive statistics is a concept that allows us to analyze and summarize data and organize the same in the form of numbers graph, bar plots, histogram, pie chart, etc. Descriptive statistics is simply a process to describe our existing data. It transforms the raw observations into some meaningful data that can be further interpreted and used. Concepts like standard deviation, central tendency are widely used around the world when it comes to learning descriptive statistics.

- **Mean:** The average value of a set of numbers, calculated by summing all the values and dividing by the total count.

- **Median:** The middle value in a dataset when it's arranged in ascending or descending order. It's less sensitive to outliers than the mean.
- **Mode:** The value that appears most frequently in a dataset.

Inferential Statistics – Inferential statistics on the other hand is an important concept that deals with drawing conclusions based on small samples collected from the entire population. For example, during an election poll, people will often want to predict the exit poll results so they will conduct a survey in various parts of state or country and record their opinion. Based on the information they have collected they tend to draw conclusions and make inferences to predict results for the entire population.

- **Range:** The difference between the maximum and minimum values in a dataset, providing an indication of data spread.
- **Standard Deviation:** A measure of the dispersion or spread of data points around the mean. A higher standard deviation indicates greater variability.
- **Variance:** The average of the squared differences from the mean. It provides a measure of data variability.
- **Percentiles:** These are values that divide a dataset into percentiles, indicating the value below which a given percentage of data falls. The median is the 50th percentile.

Why should you master statistics concepts?

Nowadays, almost all companies have become data-driven and are using various concepts to interpret their existing data. That’s where fundamental statistical concepts come into play & their implementations help us in describing the data that we have in hand.

To solve the ongoing problems in the company and predict a better strategy to improve the profit margin of the company we need to learn concepts that help us understand the data and categorize it according to their features. Thankfully, statistics has a set of tools that help us organize and visualize the data and provide actionable insights.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis is a process of examining or understanding the data and extracting insights or main characteristics of the data. EDA is generally classified into two methods, i.e. graphical analysis and non-graphical analysis.

Steps in EDA:

- **Examine the Data Distribution**: This involves understanding the spread and distribution of data values. It's essential to know how the data is distributed across different ranges.

Step 1 : Importing Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Step 2 : Load the DataSet

In [2]:
```
df = pd.read_excel('Desktop\loan_data.xlsx')
```

Step 3 : Observe the dataset by Head Method

In [3]:
```
df.head()
```

Out[3]:

	Sex	Age	Time_at_address	Res_status	Telephone	Occupation	Job_status	Time_employed	Time_bank	Liab_ref	Acc_ref	Home_Expn	Balance	Dec
0	M	50.750000	0.585	owner	given	unemploye	unemploye	0	0	f	given	145	0	
1	M	19.670000	10.000	rent	not_given	labourer	governmen	0	0	t	given	140	0	
2	F	52.830002	15.000	owner	given	creative_	private_s	5	14	f	given	0	2200	a
3	M	22.670000	2.540	rent	not_given	creative_	governmen	2	0	f	given	0	0	a
4	M	29.250000	13.000	owner	given	driver	governmen	0	0	f	given	228	0	

Step 4 : Check dimension of data

In [4]:
```
df.shape
```

Out[4]:
```
(429, 14)
```

step 5: Check all information about dataset

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 429 entries, 0 to 428
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Sex              429 non-null    object
 1   Age              429 non-null    float64
 2   Time_at_address  429 non-null    float64
 3   Res_status       429 non-null    object
 4   Telephone        429 non-null    object
 5   Occupation       429 non-null    object
 6   Job_status       429 non-null    object
 7   Time_employed    429 non-null    int64
 8   Time_bank        429 non-null    int64
 9   Liab_ref         429 non-null    object
 10  Acc_ref          429 non-null    object
 11  Home_Expn        429 non-null    int64
 12  Balance          429 non-null    int64
 13  Decision         429 non-null    object
dtypes: float64(2), int64(4), object(8)
memory usage: 47.0+ KB
```

In [6]:
```
df.describe()
```

Out[6]:

	Age	Time_at_address	Time_employed	Time_bank	Home_Expn	Balance
count	429.000000	429.000000	429.000000	429.000000	429.000000	429.000000
mean	31.510163	4.650758	1.871795	2.279720	176.727273	898.382284
std	11.843595	4.804037	3.254023	3.966105	142.590659	3814.565340
min	15.170000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	22.670000	1.000000	0.000000	0.000000	80.000000	0.000000
50%	28.500000	2.750000	1.000000	0.000000	160.000000	10.000000
75%	38.250000	7.000000	2.000000	3.000000	272.000000	484.000000
max	76.750000	25.209999	20.000000	23.000000	760.000000	51100.000000

- **Handling Missing Values**: Missing data is a common issue in datasets. EDA involves strategies for dealing with missing values, such as imputation or removing affected observations.

step 6 : Handling missing value

```
In [7]: df.isnull().sum()
```

```
Out[7]: Sex               0
        Age               0
        Time_at_address   0
        Res_status        0
        Telephone         0
        Occupation        0
        Job_status        0
        Time_employed     0
        Time_bank         0
        Liab_ref          0
        Acc_ref           0
        Home_Expn         0
        Balance           0
        Decision          0
        dtype: int64
```

we can see there are no missing value in our data set but if there are we can handle it by using following methods:

1)**Drop the missing values** – If the dataset is huge and missing values are very few then we can directly drop the values because it will not have much impact.

2)**Replace with mean values** – We can replace the missing values with mean values, but this is not advisable in case if the data has outliers.

3)**Replace with median values** – We can replace the missing values with median values, and it is recommended in case if the data has outliers.

4)**Replace with mode values** – We can do this in the case of a Categorical feature.

Handling Outliers: Outliers are data points that deviate significantly from the rest of the data. EDA helps in identifying and deciding how to handle these outliers.

Step 8 : Handling Outliers

```
In [9]: def handle_outliers(df, factor=1.5):
            q1 = df.quantile(0.25)
            q3 = df.quantile(0.75)
            iqr = q3 - q1
            lower_bound = q1 - factor * iqr
            upper_bound = q3 + factor * iqr
            return np.clip(df, lower_bound, upper_bound)


        # Numeric columns for outlier handling
        numeric_columns = ["Age", "Time_at_address", "Time_employed", "Time_bank", "Balance"]

        # Handle outliers using the handle_outliers function
        for column in numeric_columns:
            df[column] = handle_outliers(df[column])

        # Display the cleaned DataFrame
        print(df)
```

```
    Sex        Age  Time_at_address Res_status  Telephone Occupation  \
0     M  50.750000            0.585      owner      given  unemploye
1     M  19.670000           10.000       rent  not_given   labourer
2     F  52.830002           15.000      owner      given  creative_
3     M  22.670000            2.540       rent  not_given  creative_
4     M  29.250000           13.000      owner      given     driver
..   ..        ...              ...        ...        ...        ...
424   M  34.169998            2.750      owner      given  guard_etc
425   F  22.250000            1.250       rent  not_given  unemploye
426   M  23.330000            1.500      owner      given  creative_
427   M  21.000000            4.790       rent  not_given  productio
428   M  27.750000            1.290      owner      given   labourer
```

Here's a brief overview of each method:

Boxplots:

Boxplots, also known as box-and-whisker plots, are graphical representations of the data's distribution. They display the median, quartiles (Q1 and Q3), and potential outliers. Data points outside a certain range (usually defined by the "whiskers" of the boxplot) are considered potential outliers. Boxplots allow you to visualize the spread of the data and identify any data points that fall far from the central distribution.

Z-Score:

The z-score is a statistical measure that quantifies how many standard deviations a data point is away from the mean. A high z-score indicates that a data point is significantly distant from the mean, possibly indicating an outlier. A common threshold for identifying outliers is a z-score of 2 or 3, but this can vary based on the context.

Interquartile Range (IQR):

The IQR is a measure of the spread of data between the first quartile (Q1) and the third quartile (Q3). Data points that fall below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR are typically considered outliers. This method is less sensitive to extreme values compared to methods that use the mean and standard deviation.

- **Removing Duplicate Data**: Duplicate entries can distort analysis results. EDA identifies and removes duplicate data points to maintain data integrity.

Step 7 : Check for duplicates

```
In [8]: df.duplicated()

Out[8]: 0      False
        1      False
        2      False
        3      False
        4      False
               ...
        424    False
        425    False
        426    False
        427    False
        428    False
        Length: 429, dtype: bool
```

- **Encoding Categorical Variables**: Many machine learning algorithms require numerical data, so EDA might involve converting categorical variables into numerical format through encoding.

Step 9 : Perform Encoding

```
In [10]:
# Assuming your dataset is stored in a DataFrame called 'data'
encoded_data = pd.get_dummies(df, columns=['Sex', 'Res_status', 'Telephone', 'Occupation', 'Job_status', 'Liab_ref', 'Acc_ref',
```

- **Normalizing and Scaling**: Data normalization and scaling ensure that different features have similar scales, which is important for various algorithms that rely on distance or magnitude calculations.

Step 10 : Normalization and scalling

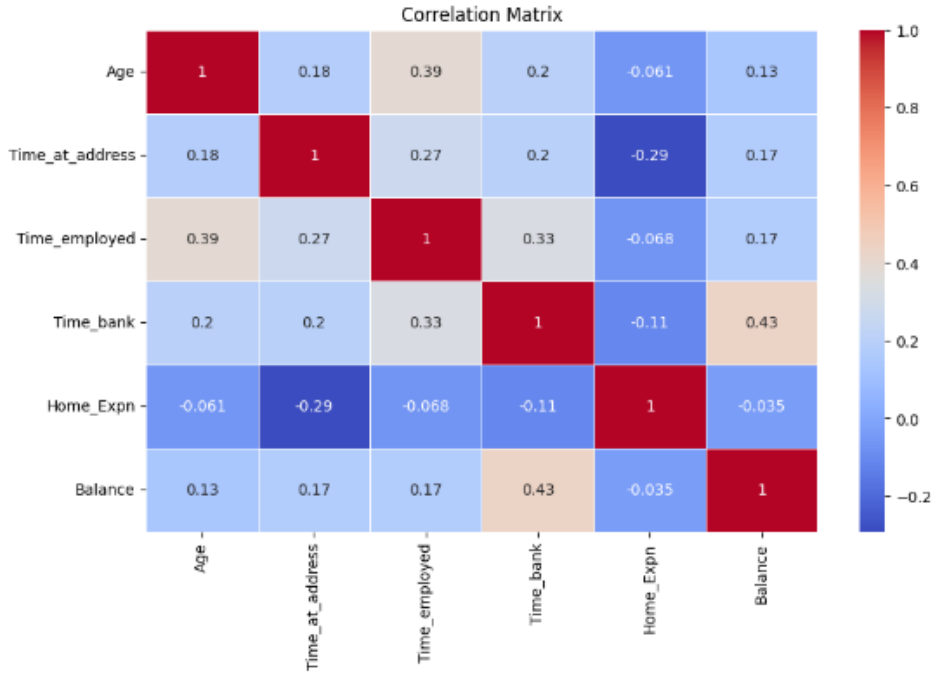
In [11]:
```
from sklearn.preprocessing import Normalizer

# Perform normalization
normalizer = Normalizer()
normalized_data = normalizer.fit_transform(encoded_data)

print(normalized_data)
```

```
[[0.33029205 0.00380731 0.         ... 0.         0.         0.00650822]
 [0.13875906 0.0705435  0.         ... 0.         0.         0.00705435]
 [0.04361493 0.01238357 0.00412786 ... 0.         0.00082557 0.        ]
 ...
 [0.04989424 0.00320794 0.00213863 ... 0.         0.00213863 0.        ]
 [0.06746975 0.01538953 0.00642569 ... 0.         0.00321285 0.        ]
 [0.19438548 0.0090363  0.         ... 0.00700488 0.         0.00700488]]
```

- **Visualization:**

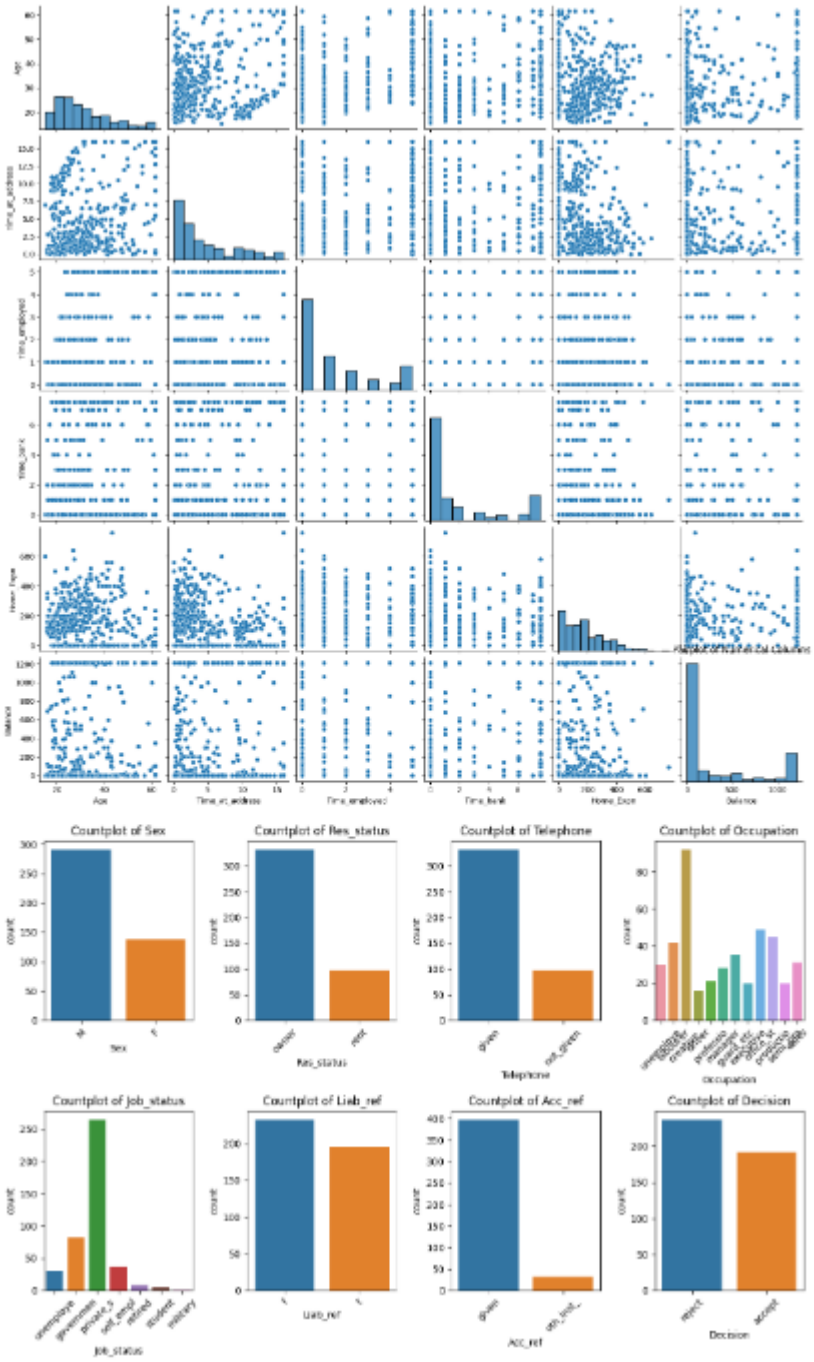
Step 11 : Correlation Matrix

In [16]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


# Calculate the correlation matrix
correlation_matrix = df.corr()

# Set up the matplotlib figure
plt.figure(figsize=(10, 6))

# Create a heatmap using seaborn
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=.5)

plt.title('Correlation Matrix')
plt.show()
```



Step 12 : Visualization plotting

[15]:
```
# Pairplot (scatter plot matrix) for numerical columns
sns.pairplot(df[['Age', 'Time_at_address', 'Time_employed', 'Time_bank', 'Home_Expn', 'Balance']])
plt.title('Pairplot of Numerical Columns')
plt.show()

# Countplot for categorical columns
categorical_columns = ['Sex', 'Res_status', 'Telephone', 'Occupation', 'Job_status', 'Liab_ref', 'Acc_ref', 'Decision']
plt.figure(figsize=(12, 8))
for i, column in enumerate(categorical_columns, 1):
    plt.subplot(2, 4, i)
    sns.countplot(data=df, x=column)
    plt.title(f'Countplot of {column}')
    plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Quiz Questions

1. **What is the purpose of descriptive statistics in data analysis?**

A. To predict future trends in the data.

B. To summarize and interpret main characteristics of the data.

C. To perform hypothesis testing.

D. To visualize data distributions.

2. Which of the following measures of central tendency is less sensitive to outliers?

A. Mean
B. Median
C. Mode
D. Standard Deviation

3. In EDA, what is the primary purpose of handling missing values?

A. To remove affected observations from the dataset.
B. To replace missing values with the mode of the variable.
C. To ensure that the dataset has a balanced distribution.
D. To maintain data integrity and prevent bias in analysis.

4. **What is the significance of the Interquartile Range (IQR) in identifying outliers?**

A. It helps determine the mean value of the dataset.
B. It provides a measure of data variability.
C. It defines a threshold for identifying potential outliers.
D. It quantifies the correlation between two variables.

5. When is Inferential Statistics typically used in data analysis?

A. To summarize and describe the main characteristics of a dataset.
B. To visualize relationships between variables using scatter plots.
C. To make predictions or draw conclusions about a population based on a sample.
D. To identify potential outliers using boxplots.