**PYTHON CODING CHALLENGE**

**Name: S.Nikhil Sai**

**Date: 15-11-2024**

**1. Printing rows of the Data:**

import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

data=pd.read\_csv(r'C:\Users\nikhi\Downloads\annual-enterprise-survey-2023-financial-year-provisional.csv')

data

A screenshot of a computer

Description automatically generated

**Explanation:** This code imports essential libraries (pandas, numpy, and matplotlib) for data analysis and visualization. It reads a CSV file located at the specified path into a DataFrame using pandas.read\_csv(). The r before the file path ensures that the string is treated as a raw string, avoiding issues with backslashes. Once loaded, the data can be explored, manipulated, or visualized using pandas and matplotlib. This setup is typically used for data preprocessing and analysis tasks.

**2**. **Printing the column names of the DataFrame:**

data.columns

**A screenshot of a computer code

Description automatically generated**

**Explanation**: The code data.columns retrieves the column names of the DataFrame data. It returns an object of type Index, which contains the labels (column names) of the DataFrame.

3.Summar of data frame:

data.info()

A screenshot of a computer

Description automatically generated

**4. Descriptive Statistical Measures of a DataFrame:**

data.describe()

A screenshot of a data

Description automatically generated

**Explanation:**The data.describe() function in pandas generates descriptive statistics of the numerical columns in a DataFrame. It provides a quick summary of the central tendency, dispersion, and shape of the distribution of the dataset.

**5**. **Missing Data Handing**

**data.dropna()**

**A screenshot of a computer

Description automatically generated**

**Explanation:** The `data.dropna()` function in pandas is used to remove rows with missing values (`NaN`). By default, it drops any row that contains at least one `NaN` value.

6. **Sorting DataFrame values**

data.sort\_values(by="Value", ascending=True).head()

A screenshot of a computer

Description automatically generated

**Explanation:** The code `data.sort\_values(by="Value", ascending=True).head()` sorts the DataFrame `data` by the `"Value"` column in ascending order. The `ascending=True` ensures the values are sorted from smallest to largest. The `.head()` function returns the first 5 rows of the sorted DataFrame. This is useful for quickly inspecting the smallest values in the `"Value"` column.

7. **Apply Function**

def combine(a):

if a >2022:

return "Latest data"

else:

return "Old data"

data['Status']= data['Year'].apply(combine)

data

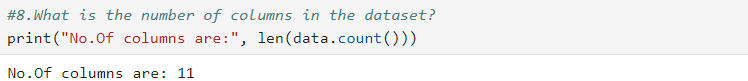
A screenshot of a computer

Description automatically generated

**Expalnation:**The code defines a function `combine(a)` that checks if the value `a` (representing the year) is greater than 2022. If it is, it returns `"Latest data"`, otherwise, it returns `"Old data"`. The `apply()` function is then used on the `Year` column of the `data` DataFrame to create a new column `Status`, which contains the result of applying the `combine` function to each year. The updated DataFrame is stored back in `data`.

8. What is the number of columns in the dataset?

print("No.Of columns are:", len(data.count()))



Explanation:The code `data.count()` counts the non-null values in each column of the DataFrame `data`. `len(data.count())` returns the number of columns that have at least one non-null value. The `print()` statement displays this count as the total number of columns with data. It shows the number of columns that contain at least one valid (non-null) entry.

9. How is the dataset indexed?

data.index

A white background with black text

Description automatically generated

**Explanation:** The code `data.index` returns the index (row labels) of the DataFrame `data`. It provides the row labels or the axis labels for the rows, which are used for indexing and accessing data. By default, it usually starts from 0 and increments by 1 for each row, unless explicitly set otherwise. It returns an object of type `Index` that represents the row labels.

10. No.Of observations

print("No of observations are:", data.shape)

A close-up of a message

Description automatically generated

The code `data.shape` returns a tuple representing the dimensions of the DataFrame `data`. The first value in the tuple represents the number of rows (observations), and the second value represents the number of columns. The `print()` statement displays this shape. To display just the number of observations, you can use `data.shape[0]`.