Python Coding Challenge

-Sarthak Niranjan Kulkarni (Maverick)

- [sarthakkul2311@gmail.com](mailto:sarthakkul2311@gmail.com) - (+91) 93256 02791

**15/11/2024 (Friday)**

1. **Printing rows of the Data**

🡪 import pandas as pd

# File path

file\_path = r"C:\Users\Sarthak Kulkarni\Desktop\Hexaware Python Training\Data\_engineering\Coding\_Challenge\Python-Coding-Challenge\annual-enterprise-survey-2023-financial-year-provisional.csv"

# Load the CSV file into a DataFrame

df = pd.read\_csv(file\_path)

# Display the first few rows of the dataset

df.head()

print(df)

A screenshot of a computer

Description automatically generated

A close-up of a computer screen

Description automatically generated

1. **Printing the column names of the DataFrame.**

🡪 print(df.columns)

A screenshot of a computer

Description automatically generated

1. **Summary of Data Frame**

🡪 print("Summary of the DataFrame structure:")

df.info()

A screenshot of a computer

Description automatically generated

1. **Descriptive Statistical Measures of a DataFrame**

🡪 # Descriptive Statistical Measures of a DataFrame

print("Descriptive statistics for numerical columns:")

print(df.describe())

# Descriptive statistics for all columns (including categorical)

print("\nDescriptive statistics for all columns:")

print(df.describe(include='all'))

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a group

Description automatically generated

1. **Missing Data Handing**

🡪 df\_filled = df.fillna(0)

print("\nDataFrame after filling missing values with 0:")

print(df\_filled)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Sorting DataFrame values.**

🡪 # 6.Sorting DataFrame values

# Ascending order

sorted\_df = df.sort\_values(by='Variable\_name', ascending=True)

print("DataFrame sorted by Variable\_name in ascending order:")

print(sorted\_df)

# Descending order

sorted\_df\_desc = df.sort\_values(by='Variable\_name', ascending=False)

print("\nDataFrame sorted by Variable\_name in descending order:")

print(sorted\_df\_desc)

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. **Merge Data Frames.**

🡪 df1 = pd.DataFrame({

'ID': [1, 2, 3],

'Name': ['Lakshita', 'Sarthak', 'Harinya']

})

df2 = pd.DataFrame({

'ID': [2, 3, 4],

'Score': [85, 90, 75]

})

merged\_df = df1.merge(df2, on='ID', how='inner')

print(merged\_df)

A screenshot of a computer

Description automatically generated

1. **Apply Function.**

🡪 def is\_year\_2023(year):

return year == 2023

df\_2023 = df[df['Year'].apply(is\_year\_2023)]

print("DataFrame with data from the year 2023:")

print(df\_2023)

A screenshot of a computer

Description automatically generated

1. **By using the lambda operator.**

🡪 # Lambda Function to get rows where 'Year' is 2023

df\_2023 = df[df['Year'].apply(lambda x: x == 2023)]

print("DataFrame with data from the year 2023:")

print(df\_2023)

A screenshot of a computer

Description automatically generated

1. **Visualizing DataFrame.**

🡪 # Visualizing DataFrame

# Simple histogram for the 'Value' column

plt.figure(figsize=(8, 5))

plt.hist(df['Value'], bins=10, color='skyblue', edgecolor='black')

plt.title('Distribution of Value')

plt.xlabel('Value')

plt.ylabel('Frequency')

plt.show()

A graph of value distribution

Description automatically generated