Project 4 : Relation Of Other Attribute With Fish Growth

Source Code :

```
#Importing necessary libraries
import pandas as pd
#Uploading files
from google.colab import files
uplodaed=files.upload()

#Reading and printing the whole dataset
df=pd.read_csv("Sample Dataset.csv")
df

#Find out all the information of the dataset
df.info()

#Find out the shape of the dataset
df.shape

#Find out the size of the dataset
df.size

#Find out the columns of the dataset
df.columns

#Find out the tail or the last 5 row of the dataset
df.tail()

#Dropping the Nan values of the dataset
df.dropna

#Dropping the Nan values of the dataset
df.dropna

#Find out the numerical columns of the dataset
numerical_cols = df.columns[df.dtypes != object]
numerical_cols

#Find out the categorical columns of the dataset
categorical_cols=df.columns[df.dtypes!=object]
categorical_cols
```

```
#Find out the datatypes of each column of the dataset
df.dtypes

#Creating a new dataset by dropping unnecessary columns of previous dataset
df1=df.drop(["created_at", "entry_id", "Population"],axis=1)
df1.head()

#Find out unique values of some columns
df1['Temperature'].unique()

#Find out unique values of some columns
df1['Turbidity DO'].unique()

#Find out unique values of some columns
df1['D0'].unique()

#Find out unique values of some columns
df1['PH'].unique()

#Find out unique values of some columns
df1['Ammonia'].unique()

#Find out unique values of some columns
df1['Nitrate'].unique()

#Find out correlation of "Length" column with other columns
correlation_matrix = df1.corr()
correlation_matrix["Length"]

#Find out correlation of "Weight" column with other columns
correlation_matrix = df1.corr()
correlation_matrix["Weight"]

#Find out the shape of new dataset
df1.shape

#Find out the size of new dataset
df1.size

#Printing the whole dataset
df1
```

```
#Find out the data types of new dataset
df1.dtypes

#Find out the numerical cols of new dataset
numerical_cols1=df1.columns[df1.dtypes!=object]
numerical_cols1

#Find out the categorical columns of new dataset
categorical_cols1=df1.columns[df1.dtypes==object]
categorical_cols1

#Importing necessary libraries
import numpy as np
#Creating another new dataset by replacing the positive and negative infinite value with nan and then drop the nan values
df2 = df1.replace([np.inf, -np.inf], np.nan, inplace=True)
df2 = df1.dropna()
#Printing the whole dataset without any nan values
df2

#Importing necessary libraries to calculate distance
from scipy.spatial import distance
#Find out the Minkowski distance of "Weight" column with "Temperature" column
mink_dist1 =
distance.minkowski(df2["Weight"].astype(float),df2["Temperature"].astype(float),
p=100)
print(mink_dist1)
#Find out the Minkowski distance of "Weight" column with "Turbidity DO" column
mink_dist2 = distance.minkowski(df2["Weight"].astype(float),df2["Turbidity
DO"].astype(float),p=100)
print(mink_dist2)
#Find out the Minkowski distance of "Weight" column with "Turbidity D0" column
mink_dist3 =
distance.minkowski(df2["Weight"].astype(float),df2["D0"].astype(float),p=100)
print(mink_dist3)
#Find out the Minkowski distance of "Weight" column with "PH" column
mink_dist4 =
distance.minkowski(df2["Weight"].astype(float),df2["PH"].astype(float),p=100)
print(mink_dist4)
#Find out the Minkowski distance of "Weight" column with "Ammonia" column
```

```
mink_dist5 = distance.minkowski(df2["Weight"].astype(float),df2["Ammonia"].astype(float),p=100)
print(mink_dist5)
#Find out the Minkowski distance of "Weight" column with "Nitrate" column
mink_dist6 = distance.minkowski(df2["Weight"].astype(float),df2["Nitrate"].astype(float),p=100)
print(mink_dist6)
#Find out the Minkowski distance of "Weight" column with "Length" column
mink_dist7 = distance.minkowski(df2["Weight"].astype(float),df2["Length"].astype(float),p=100)
print(mink_dist7)
```

Program Explanation : This Python program appears to be an analysis of a dataset using the pandas library in a Jupyter Notebook or Google Colab environment. Let's break down the code and understand each step :

1. **Importing Libraries :**

```
import pandas as pd
from google.colab import files
```

The code begins by importing the necessary libraries. **pandas** is imported with the alias **pd**, and the **files** module is imported from **google.colab** to upload files.

2. **Uploading Dataset :**

```
uploaded = files.upload()
```

The code allows the user to upload a file, presumably a CSV file named "Sample Dataset.csv."

3. **Reading Dataset :**

```
df = pd.read_csv("Sample Dataset.csv")
```

The CSV file is read into a pandas DataFrame named **df**.

4. Displaying Dataset Information :

```
df.info()
```

This command prints information about the DataFrame, including data types, non-null counts, and memory usage.

5. **Displaying Dataset Shape and Size :**

```
df.shape
df.size
```

These commands display the shape (number of rows and columns) and size (total number of elements) of the DataFrame.

6. **Displaying Columns and Tail of the Dataset :**

```
df.columns
df.tail()
```

The code displays the column names and the last five rows of the DataFrame.

7. **Handling Missing Values :**

```
df.dropna
```

This line attempts to drop missing values, but it seems to be missing parentheses and won't execute properly. It should be **df.dropna()**.

8. **Identifying Numerical and Categorical Columns :**

```
numerical_cols = df.columns[df.dtypes != object]
categorical_cols = df.columns[df.dtypes != object]
```

These lines identify and print the numerical and categorical columns in the DataFrame.

9. **Creating a New Dataset by Dropping Unnecessary Columns :**

```
df1 = df.drop(["created_at", "entry_id", "Population"], axis=1)
```

A new DataFrame **df1** is created by dropping specified columns from the original DataFrame.

10. **Displaying Unique Values of Columns :**

```
df1['Temperature'].unique()
# ... (repeated for other columns)
```

The unique values of several columns are printed.

11. **Calculating Correlation :**

```
correlation_matrix = df1.corr()
correlation_matrix["Length"]
correlation_matrix["Weight"]
```

Correlation matrices are calculated, and correlations with specific columns are printed.

12. **Creating Another New Dataset And Handling Infinite Values :**

```
df2 = df1.replace([np.inf, -np.inf], np.nan, inplace=True)
df2 = df1.dropna()
```

A new DataFrame **df2** is created by replacing infinite values with NaN and dropping rows with NaN.

13. Calculating Minkowski Distances :

```
from scipy.spatial import distance
# ... (repeated for various column pairs)
```

Minkowski distances are calculated for the "Weight" column with other specified columns.

The program concludes with various print statements displaying the calculated distances.

It's worth noting that there are a couple of issues in the code, such as missing parentheses in the **dropna** lines and a potential mistake in the replacement of infinite values in **df2**.