#Exercise 1: Load and Inspect the Data

import pandas as pd

#Load the Iris dataset from a CSV file.

df=pd.read\_csv("/content/iris - iris.csv")

#Display the first 10 rows of the dataset

df.head(10)

#Display the data types of each column

df.dtypes

#Exercise 2:Summary Statistics

#Calculate summary statistics for each feature

df.describe()

#Calculate the mean sepal length for each species.

df.groupby("Species").mean()

#Exercise 3: Data Cleaning

#Check for missing values.

df.isnull().sum() #no missing values

#Replace any missing values with the mean of the respective column (if any)

# Select only numeric columns

numeric\_cols = df.select\_dtypes(include=['float64'])

# Replace missing values with the mean of the respective column

numeric\_cols\_filled = numeric\_cols.apply(lambda x: x.fillna(x.mean()))

numeric\_cols\_filled

#Exercise 4: Data Filtering

#Filter the dataset to include only rows where the sepal length is greater than5.0

df[df["Sepal.Length"]>5.0]

#Filterthe dataset to include only rows of the species 'Setosa'.

df[df.Species=='setosa']

#Exercise 5: Data Aggregation

#Calculate the mean, median, and standard deviation of petal length for each species

df.groupby("Species")["Petal.Length"].agg(["mean","median","std"])

#Count the number of occurrences of each species

df.Species.value\_counts()

#Calculate the minimum and maximum petal width for each species

df.groupby("Species")["Petal.Width"].agg(["min","max"])

# Find the species with the highest average sepal width

df.groupby("Species")["Sepal.Width"].mean().idxmax()

#Exercise 6: Data Transformation

#Normalizethe numerical features (sepal length, sepal width, petal length, petal

#width) to a range of 0 to 1.

df\_normalized = df.copy()

columns\_to\_normalize = ['Sepal.Length', 'Sepal.Width' ,'Petal.Length' ,'Petal.Width']

for column in columns\_to\_normalize:

min\_value = df[column].min()

max\_value = df[column].max()

df\_normalized[column] = (df[column] - min\_value) / (max\_value - min\_value)

df\_normalized

#Create a new column that is the ratio of petal length to petal width

df["Ratio"]=df["Petal.Length"]/df["Petal.Width"]

df

#Exercise 7: Advanced Data Aggregation

#Calculate the 25th, 50th, and 75th percentiles of sepal length for each species

df.groupby("Species")["Sepal.Length"].quantile([0.25,0.5,0.75])

#Determine the range (max - min) of petal length for each species

range\_df=df.groupby("Species")["Petal.Length"].agg(["max","min"])

# Calculate the range (max - min) for each species

range\_df["range"] = range\_df["max"] - range\_df["min"]

range\_df

#Exercise 8:Merging and Joining

#create a new DataFrame with additional information about each species (e.g.,

#typical habitat).Merge this new DataFrame with the original Iris DataFrame."""

df1= pd.DataFrame({

'Species': ['setosa', 'versicolor', 'virginica'],

'typical\_habitat': ['Grasslands', 'Wetlands', 'Meadows']

})

df.merge(df1,on="Species")

#Exercise 9: Applying Custom Functions

#Create a custom function that categorizes flowers as 'small', 'medium', or

#'large' based on petal length.Applythis function to create a new column in the dataset

def categorize(row):

if row["Petal.Length"]<2:

return "small"

elif row["Petal.Length"]>5:

return "large"

else:

return "medium"

df["Category"]=df.apply(categorize,axis=1)

df