**Lab-05**

**Predictive Analytics**

**Submitted By**

**Name :Rhythm Kachhwaha**

**Sap Id : 500107218**

**Batch : 4**

**Roll No : R2142221314**

**Submitted To**

**Dr . Priyanka Singh**

Q1. Measure of Central Tendency: Mean, Geometric mean, Harmonic mean, Mode, Median

import pandas as pd

import numpy as np

from scipy import stats

# Load the CSV file

file\_path = r"D:\bodyPerformance.csv"

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

# Summary for all columns

summary = data.describe()

# Mean for all numeric columns

mean = data.mean(numeric\_only=True)

# Geometric Mean

geometric\_mean = data.apply(lambda x: stats.gmean(x.dropna()) if np.issubdtype(x.dtype, np.number) else np.nan)

# Harmonic Mean (only for positive values)

harmonic\_mean = data.apply(lambda x: stats.hmean(x[x > 0].dropna()) if np.issubdtype(x.dtype, np.number) and (x > 0).all() else np.nan)

# Mode

mode = data.apply(lambda x: x.mode()[0] if not x.mode().empty else np.nan)

# Median

median = data.median(numeric\_only=True)

# Combine all in one summary table

summary.loc['mean'] = mean

summary.loc['geometric\_mean'] = geometric\_mean

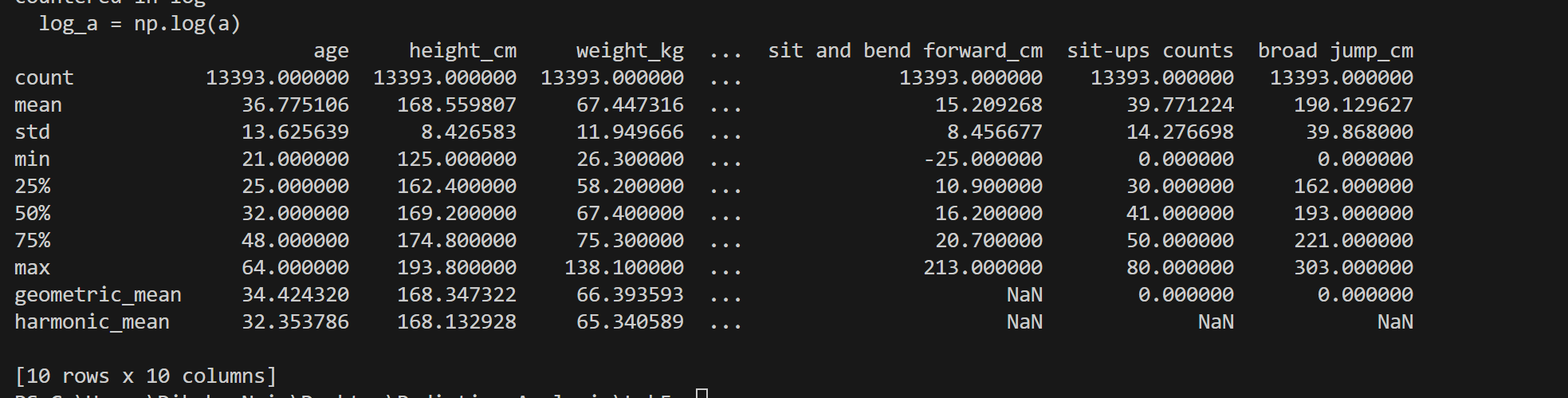
summary.loc['harmonic\_mean'] = harmonic\_mean

summary.loc['mode'] = mode

summary.loc['median'] = median

print(summary)

Output:



Q2. Measure of Dispersion: Variance, Standard deviation, Shape of Data (Symmetric, Skewness), Inter Quartile Range (IQR) / percentiles, Range, Mean Absolute Deviation(MAD)

import pandas as pd

import numpy as np

# Load the CSV file

data = pd.read\_csv('bodyPerformance.csv')

# Select only numerical columns

numerical\_data = data.select\_dtypes(include=[np.number])

# Variance

variance = numerical\_data.var()

# Standard Deviation

std\_dev = numerical\_data.std()

# Skewness (Shape of Data)

skewness = numerical\_data.skew()

# Inter Quartile Range (IQR)

iqr = numerical\_data.quantile(0.75) - numerical\_data.quantile(0.25)

# Range

data\_range = numerical\_data.max() - numerical\_data.min()

# Manually calculate Mean Absolute Deviation (MAD)

mad = numerical\_data.apply(lambda x: np.mean(np.abs(x - x.mean())), axis=0)

# Combine all in one summary table

dispersion\_summary = pd.DataFrame({

    'variance': variance,

    'std\_dev': std\_dev,

    'skewness': skewness,

    'iqr': iqr,

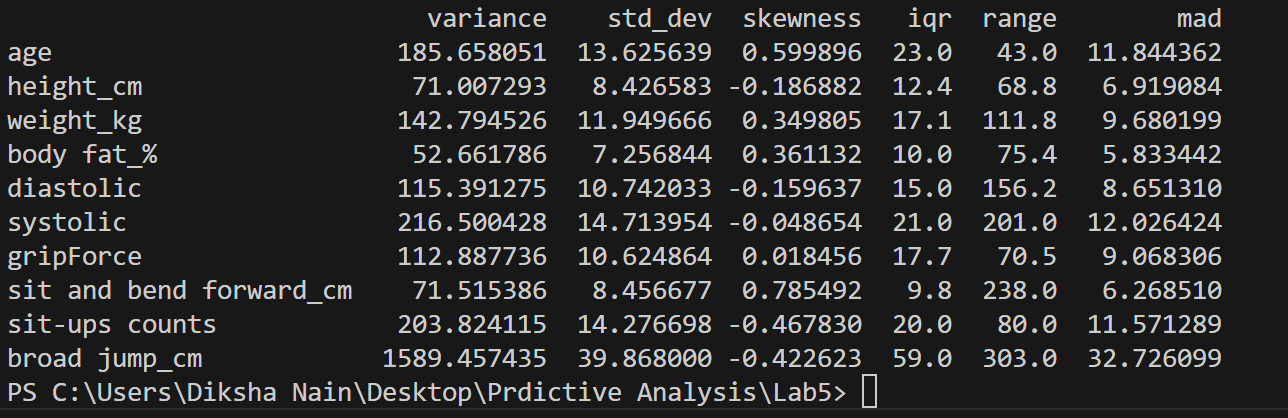
    'range': data\_range,

    'mad': mad

})

print(dispersion\_summary)

Output :



Que3.   Correlation between features.

import pandas as pd

import numpy as np

from scipy import stats

import seaborn as sns

import matplotlib.pyplot as plt

# Load the CSV file

data = pd.read\_csv('bodyPerformance.csv')

# Display the first few rows and data types to identify non-numeric columns

print(data.head())

print(data.dtypes)

# Select only numeric columns for correlation

numeric\_data = data.select\_dtypes(include=[np.number])

# Calculate the correlation matrix

correlation\_matrix = numeric\_data.corr()

# Check for NaN values

if correlation\_matrix.isnull().values.any():

    print("Warning: Correlation matrix contains NaN values.")

# Set the size of the plot

plt.figure(figsize=(10, 8))

# Create a heatmap

sns.heatmap(correlation\_matrix, annot=True, fmt=".2f", cmap='coolwarm', square=True, cbar\_kws={"shrink": .8})

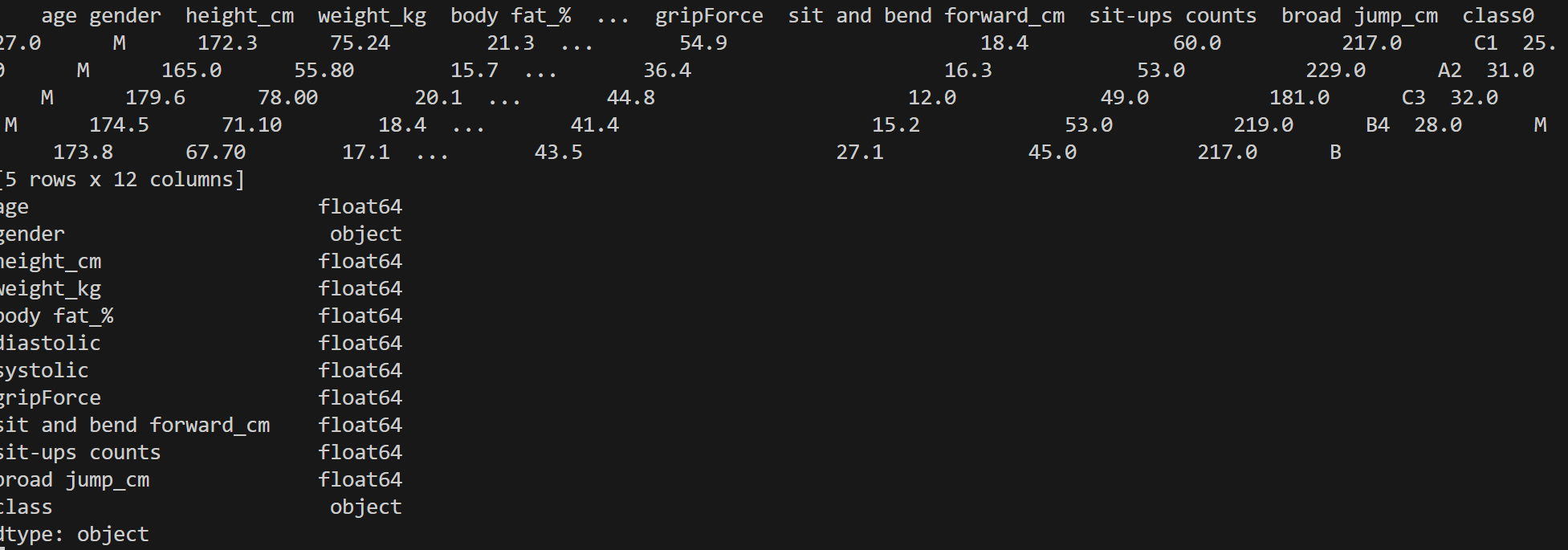
# Add title

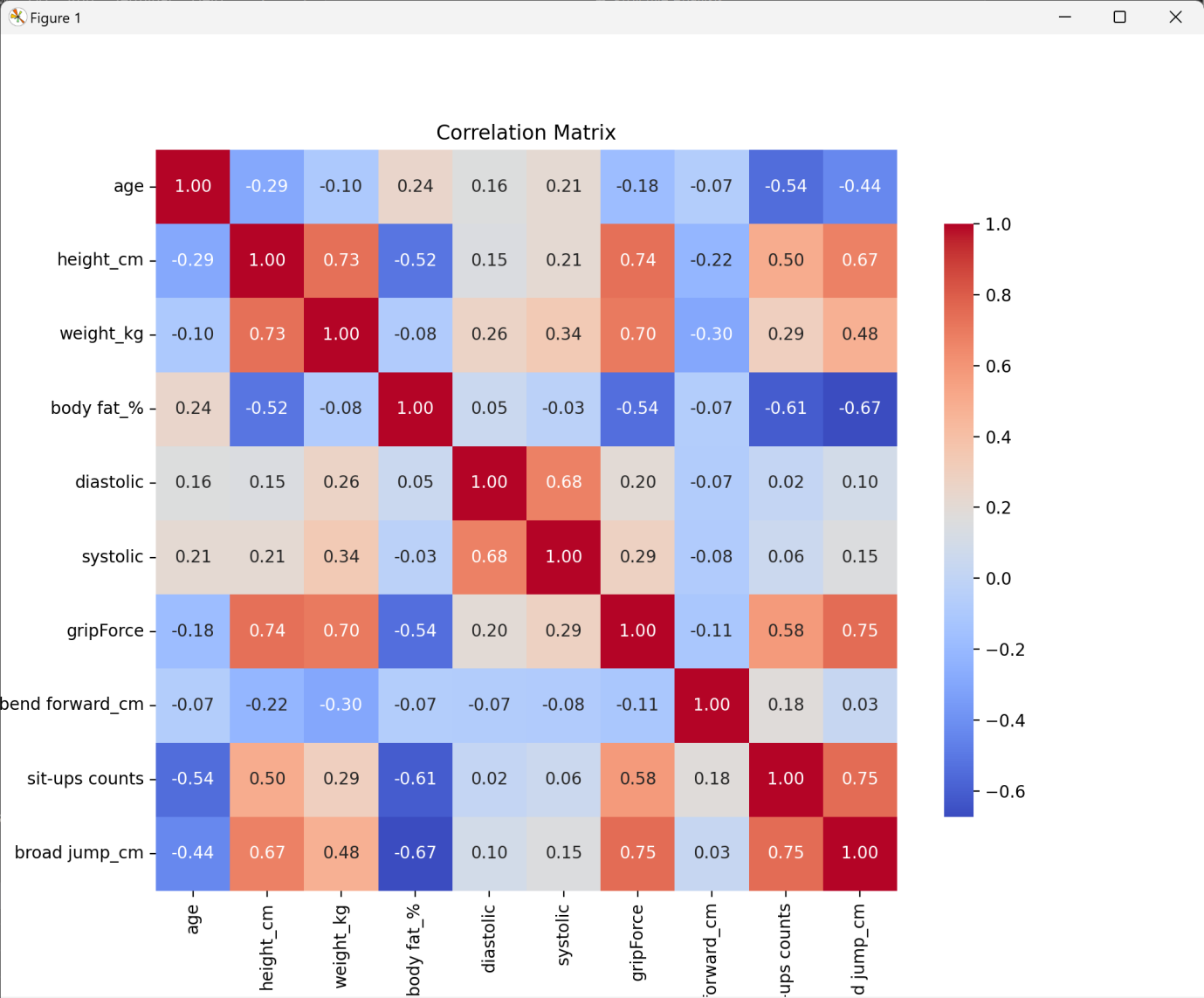
plt.title('Correlation Matrix')

# Show the plot

plt.show()

Output :



  
  
Que4. Visualizing Data Distribution: Boxplot, Histograms, Density plots, Scatterplot, Bar chart  
  
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

# Load the CSV file

data = pd.read\_csv('bodyPerformance.csv')

# Print the column names

print(data.columns)

# Boxplot

plt.figure(figsize=(12, 6))

data.boxplot(figsize=(12, 6))

plt.title('Boxplot of Numeric Columns')

plt.xticks(rotation=45)

plt.show()

# Get numeric columns

numeric\_cols = data.select\_dtypes(include=[np.number]).columns

# Histograms

data[numeric\_cols].hist(bins=20, figsize=(14, 10))

plt.suptitle('Histograms of Numeric Columns')

plt.show()

# Density Plot

plt.figure(figsize=(12, 6))

for column in numeric\_cols:

    sns.kdeplot(data[column], label=column, fill=True)

plt.title('Density Plot of Numeric Columns')

plt.legend()

plt.show()

# Scatterplot using height\_cm and weight\_kg

plt.figure(figsize=(10, 6))

sns.scatterplot(data=data, x='height\_cm', y='weight\_kg')

plt.title('Scatterplot between Height and Weight')

plt.xlabel('Height (cm)')

plt.ylabel('Weight (kg)')

plt.show()

# Bar Chart for gender

plt.figure(figsize=(10, 6))

data['gender'].value\_counts().plot(kind='bar')

plt.title('Bar Chart of Gender')

plt.xlabel('Gender')

plt.ylabel('Counts')

plt.xticks(rotation=0)  # Rotate x labels for better visibility

plt.show()

Output :

