**Predictive Analytics**

**Lab-5**

**Name: Shalu Priya**

**Sap id: 500107625**

**Batch: B5(AIML-NH)**

Descriptive Statistics of data:

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

**Code:**

# Selecting relevant numerical columns for analysis

numeric\_columns = ['age', 'height\_cm', 'weight\_kg', 'body fat\_%', 'diastolic', 'systolic', 'gripForce', 'sit and bend forward\_cm', 'sit-ups counts', 'broad jump\_cm']

# Mean

**mean\_values = df[numeric\_columns].mean()**

# Geometric Mean (ignoring negative and zero values as they are not valid for geometric mean)

**geom\_mean\_values = df[numeric\_columns].apply(lambda x: stats.gmean(x[x > 0]))**

# Harmonic Mean (ignoring negative and zero values)

**harmonic\_mean\_values = df[numeric\_columns].apply(lambda x: stats.hmean(x[x > 0]))**

# Mode

**mode\_values = df[numeric\_columns].mode().iloc[0]**

# Median

**median\_values = df[numeric\_columns].median()**

# Central Tendency Results

**central\_tendency = pd.DataFrame({**

**'Mean': mean\_values,**

**'Geometric Mean': geom\_mean\_values,**

**'Harmonic Mean': harmonic\_mean\_values,**

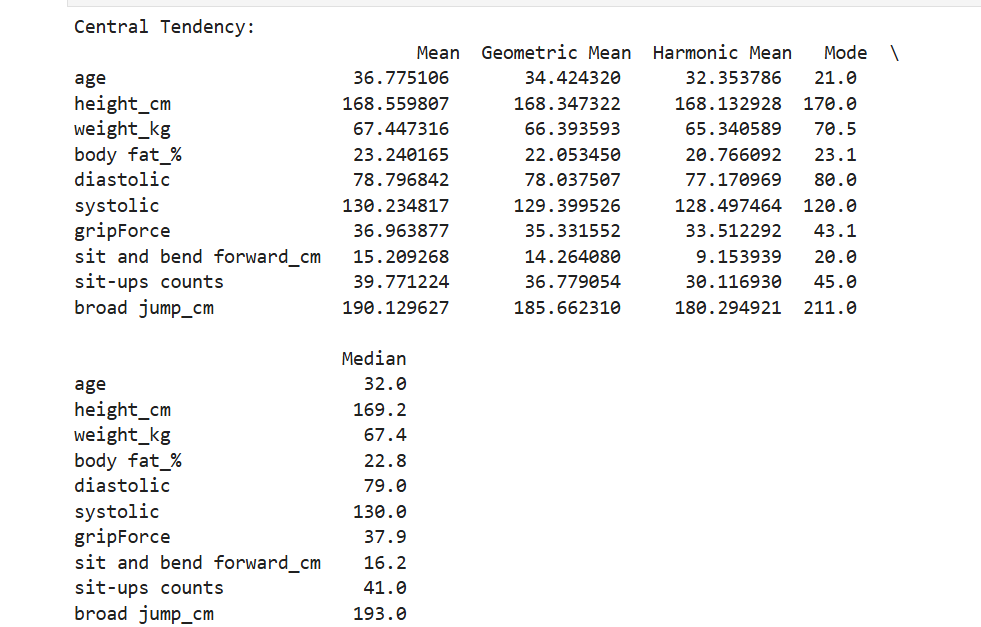
**'Mode': mode\_values,**

**'Median': median\_values**

**})**

**print("Central Tendency:\n", central\_tendency)**

**Output:**

****

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

**Code:**

# Mean Absolute Deviation (MAD)

**mad\_values = df[numeric\_columns].apply(lambda x: np.mean(np.abs(x - np.mean(x))))**

# Dispersion Results

**dispersion = pd.DataFrame({**

**'Variance': variance\_values,**

**'Standard Deviation': std\_dev\_values,**

**'Skewness': skewness\_values,**

**'IQR': iqr\_values,**

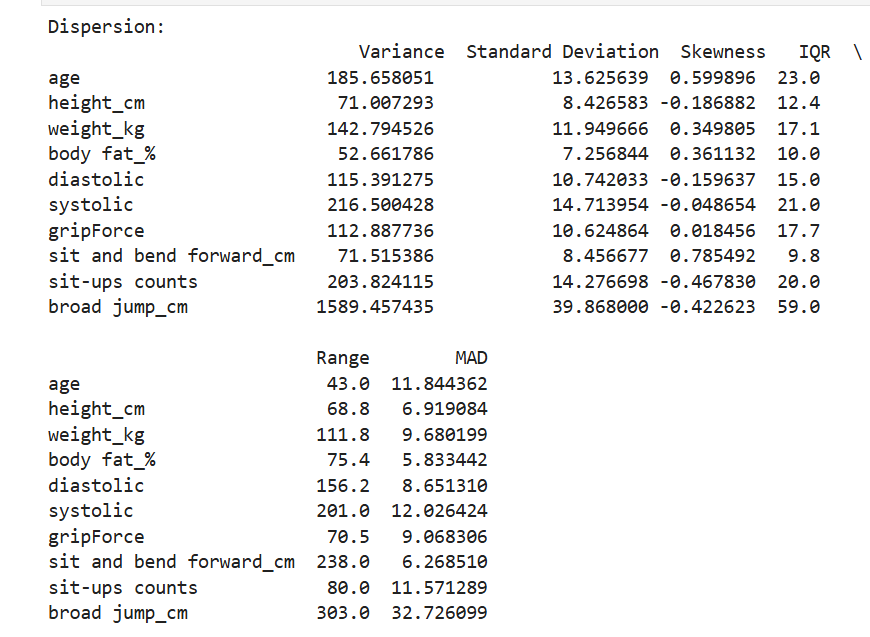
**'Range': range\_values,**

**'MAD': mad\_values**

**})**

**print("Dispersion:\n", dispersion)**

**Output:**

****

1.3 Correlation between features.

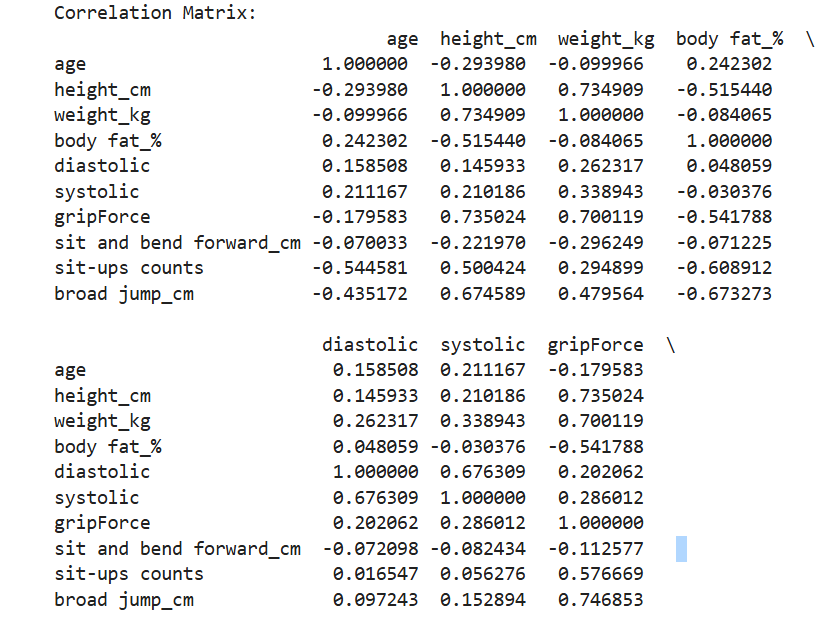
**Code:**

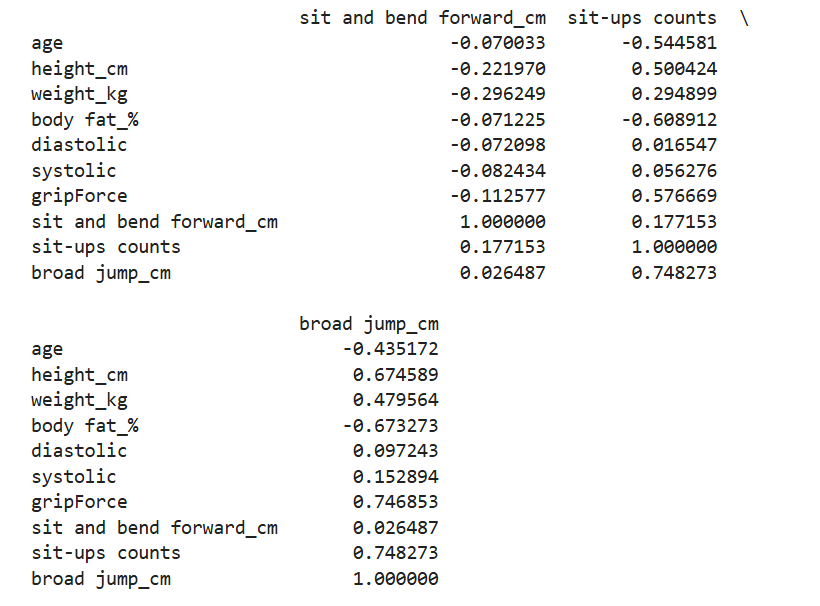
# Correlation Matrix (default is Pearson correlation)

**correlation\_matrix = df[numeric\_columns].corr()**

**print("Correlation Matrix:\n", correlation\_matrix)**

**Output:**

****

****

1.4. Visualizing Data Distribution: Boxplot, Histograms, Density plots, Scatterplot, Bar chart

**Code:**

**Boxplot:**

# Boxplot for each numerical feature

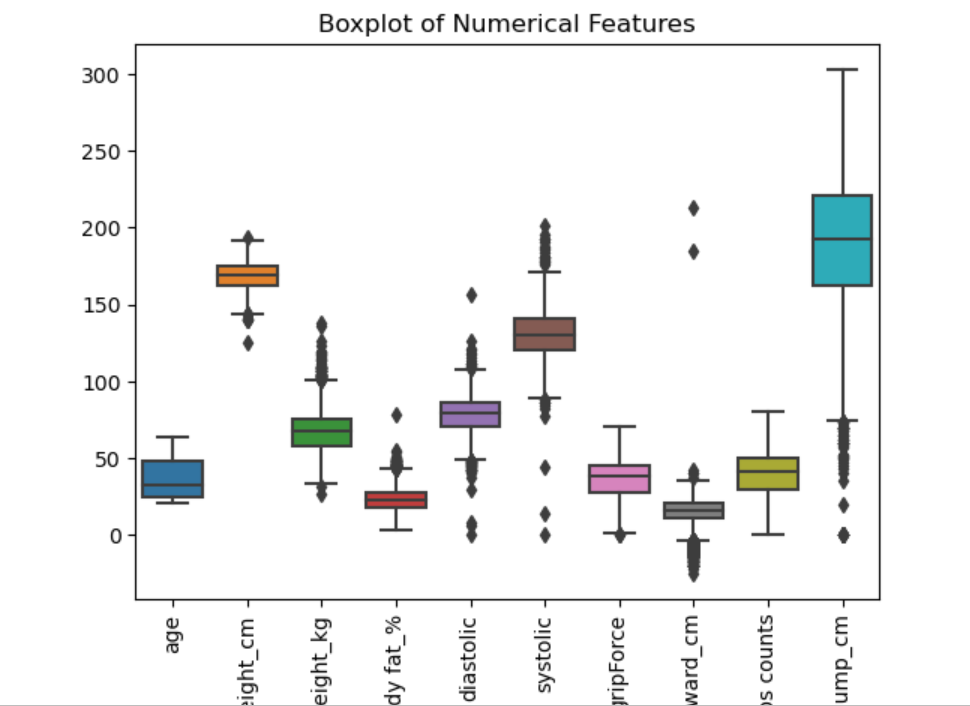
**sns.boxplot(data=df[numeric\_columns])**

**plt.title('Boxplot of Numerical Features')**

**plt.xticks(rotation=90)**

**plt.show()**

**Output:**



**Histograms**

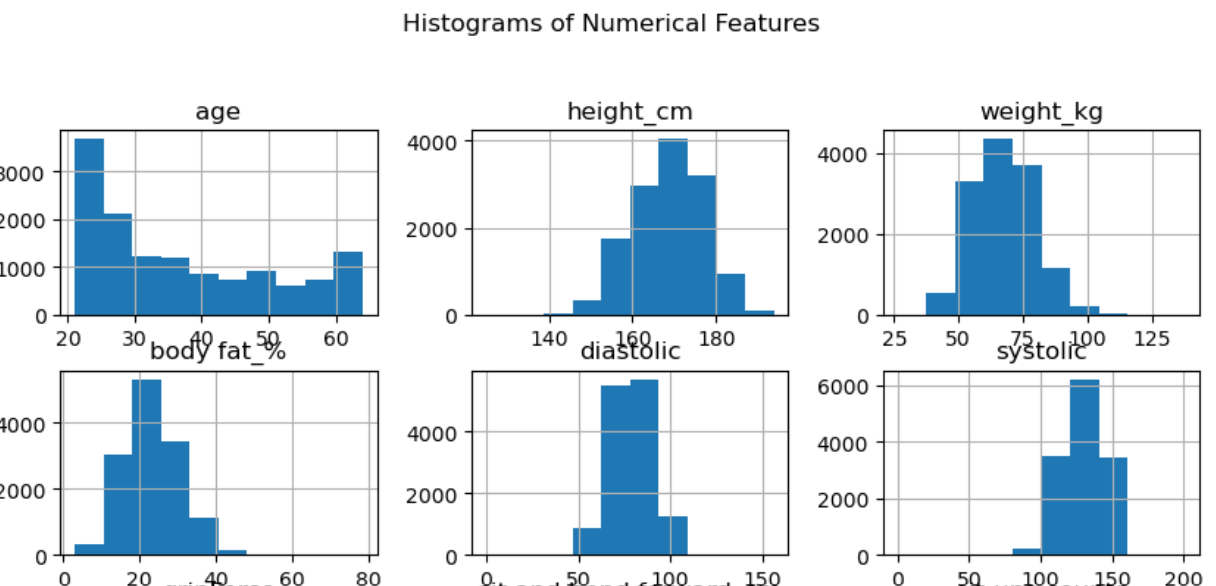
# Histogram for each feature

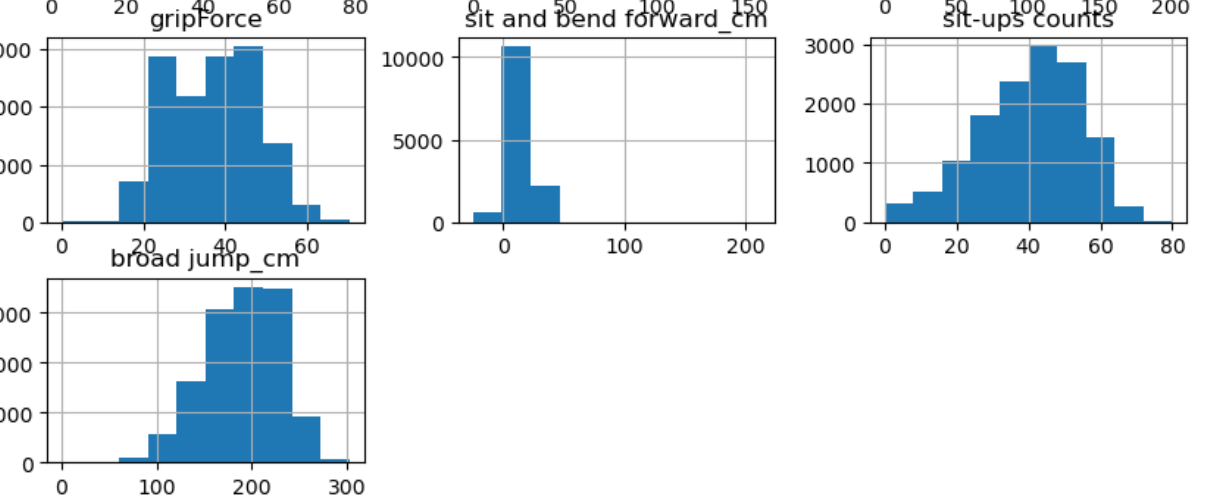
**df[numeric\_columns].hist(figsize=(10, 8), bins=10)**

**plt.suptitle('Histograms of Numerical Features')**

**plt.show()**

**Output:**

****

****

**Density Plots:**

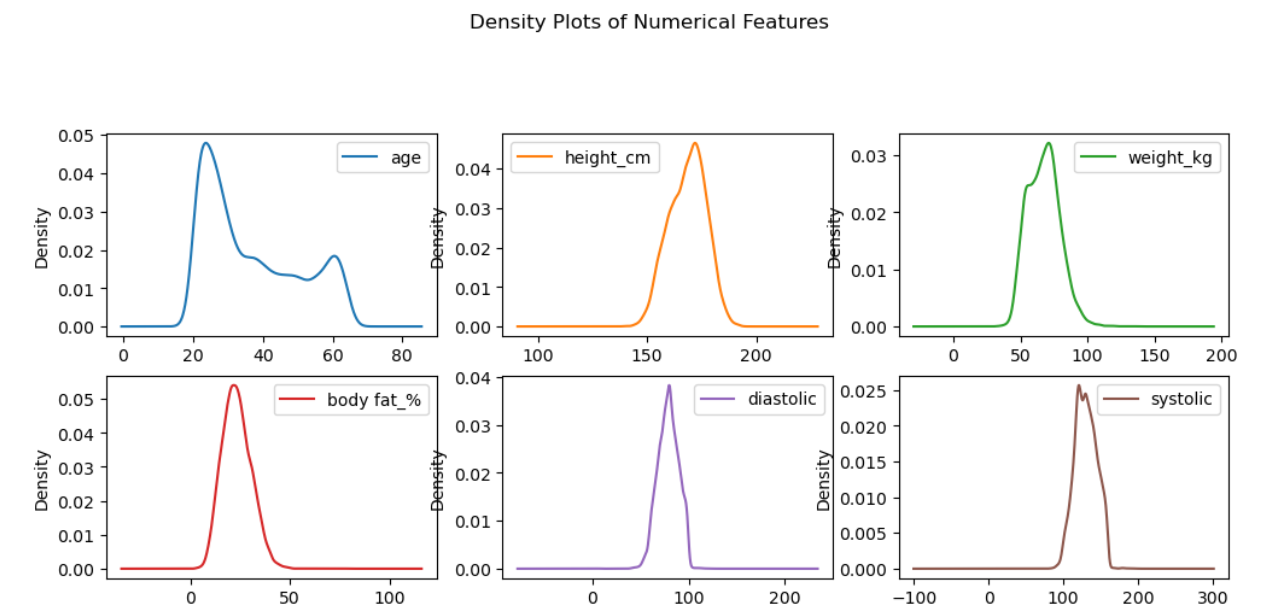
# Density Plot for each feature

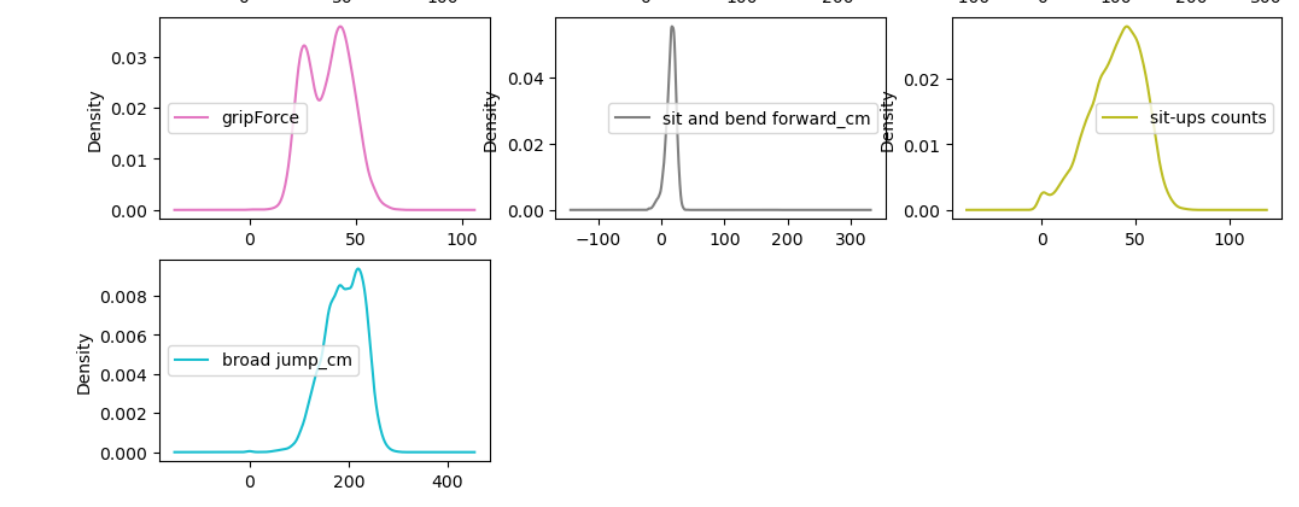
**df[numeric\_columns].plot(kind='density', subplots=True, layout=(4,3), figsize=(12,10), sharex=False)**

**plt.suptitle('Density Plots of Numerical Features')**

**plt.show()**

**Output:**

****

****

**Scatterplots:**

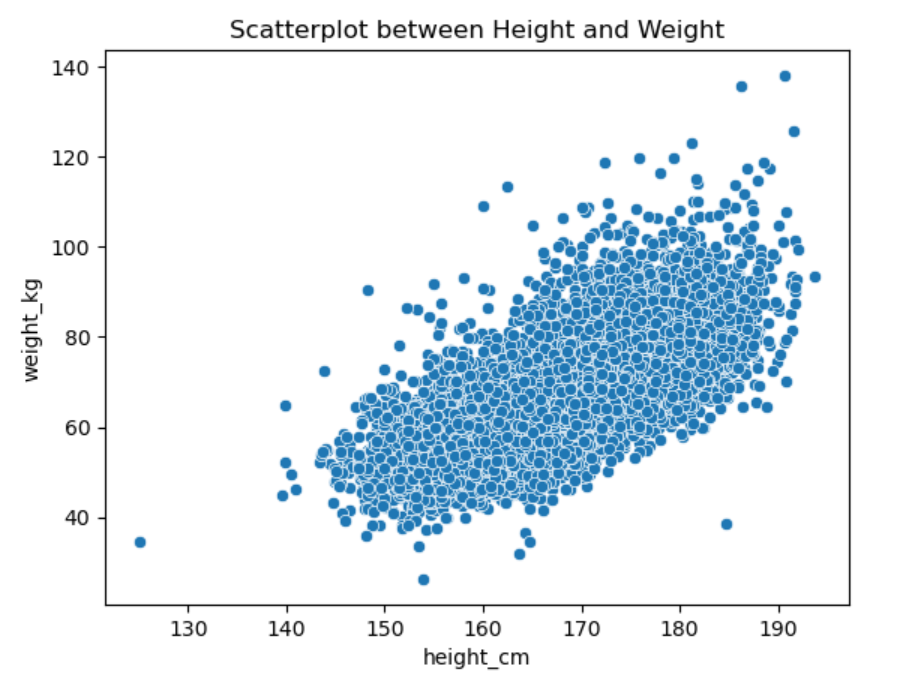
# Scatterplot between 'height\_cm' and 'weight\_kg'

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

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

**plt.show()**

**Output:**

****

**Bar Chart:**

# Bar chart of Value Counts for 'age' (or any other column)

**df['age'].value\_counts().plot(kind='bar')**

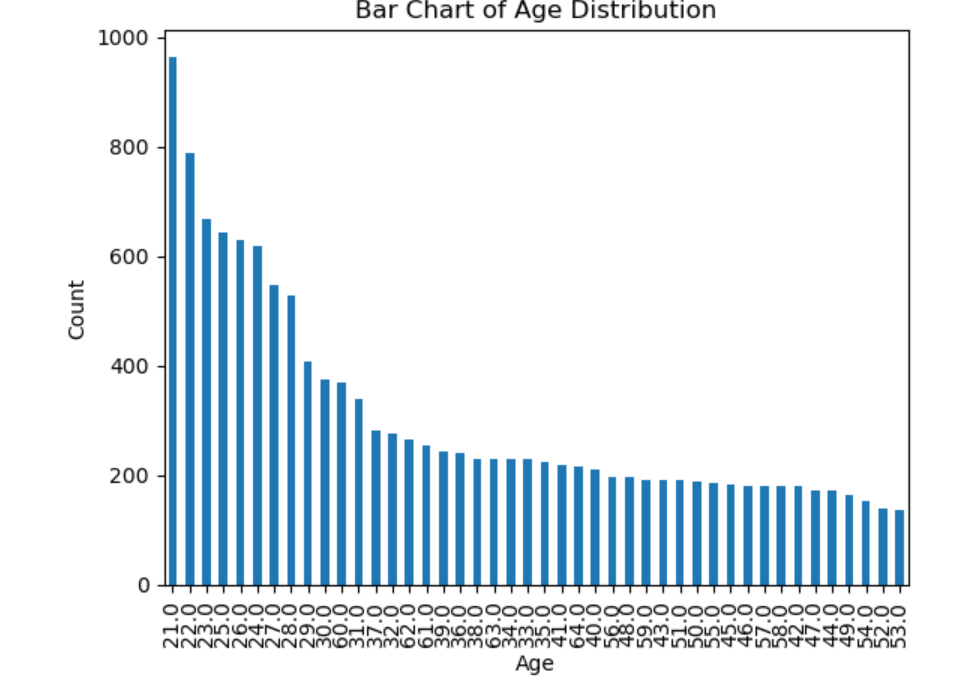
**plt.title('Bar Chart of Age Distribution')**

**plt.xlabel('Age')**

**plt.ylabel('Count')**

**plt.show()**

**Output:**

****