Student name: Islam Ibrahim
Student ID: 200209409

1.What are some of the measures you may use to explore your data as your first step of analysis?

1. **Descriptive Statistics:**
    - **Calculate basic descriptive statistics like mean, median, mode, minimum, maximum, and standard deviation for each numerical column (e.g., WEIGHT_KG, HEIGHT_METERS, WAIST_CIRCUMFERENCE(CM), HIP_CIRCUMFERENCE(CM), AGE).**
2. **Data Summary:**
    - **Use functions like describe() in Python or summary statistics in R to get a quick overview of your data, including count, mean, std deviation, min, and max.**
3. **Data Visualization:**
    - **Create visualizations such as histograms, box plots, and scatter plots to identify patterns, outliers, and the distribution of your data.**
4. **Correlation Analysis:**
    - **Calculate and visualize correlation coefficients between numerical variables. This helps you understand the relationships between different features.**
5. **Handling Missing Values:**
    - **Identify and address missing values. Decide on a strategy for imputing or handling missing data based on the nature of the missingness.**
6. **Data Cleaning:**
    - **Check for anomalies or outliers in your data. Decide whether to remove or transform them based on their impact on your analysis.**
7. **Categorical Variables Analysis:**
    - **For categorical variables like GENDER and OBESITY, calculate counts and proportions to understand the distribution of categories.**
8. **Data Quality Checks:**
    - **Check for unrealistic values or inconsistencies in your data. For example, in your data, there might be a height value of 0.74, which seems unusual.**
9. **Explore Relationships:**

- **Explore relationships between variables, especially the relationship between independent variables and the target variable (OBESITY in this case).**

10. **Data Distribution:**

    - **Examine the distribution of each variable. Are they normally distributed, skewed, or have multiple peaks?**

11. **Age Group Analysis:**

    - **Consider grouping ages into categories to better understand age-related patterns.**

Example:

```
descriptive_stats = df.describe()
print("Descriptive Statistics:\n", descriptive_stats)
```

2.Perform all the data cleaning tasks.(Handle missing data, data munging and smooth any noisy data)

1. **Handle Missing Data:**

    - Replaces 'unknown' and empty values with NaN in the first dataset.

```
df1.replace(['unknown', ''], np.nan, inplace=True)
```

2. **Data Munging:**

    - Converts 'GENDER' values to uppercase for consistency.
    - Converts 'OBESITY' values to a categorical type.

```
#Data Munging: Convert 'GENDER' to uppercase for consistency
#and 'OBESITY' to categorical type
df1['GENDER'] = df1['GENDER'].str.upper()
df1['OBESITY'] = df1['OBESITY'].astype('category')
```

3. **Smooth Noisy Data:**

    - Replaces outliers in all numerical columns with the median value.

4. **Impute Missing Values:**

    - Imputes missing values in numerical columns with the median.

5. **Display and Save the Cleaned Dataset:**

    - Displays the cleaned and munged first dataset.
    - Saves the cleaned and munged first dataset to a new CSV file.

- #Smooth Noisy Data: Replace outliers in all numerical columns with the median
- numerical_cols = ['WEIGHT_KG', 'HEIGHT_METERS', 'WAIST_CIRCUMFERENCE(CM)', 'HIP_CIRCUMFERENCE(CM)', 'AGE']
- for col in numerical_cols:
-     z_scores = np.abs(stats.zscore(df1[col]))
-     outliers = (z_scores > 3)
-     median_value = df1.loc[~outliers, col].median()
-     df1.loc[outliers, col] = median_value
- #Impute missing values using SimpleImputer for numerical columns
- imputer = SimpleImputer(strategy='median')
- df1[numerical_cols] = imputer.fit_transform(df1[numerical_cols])
-
- #Display the cleaned and munged dataset
- print("\nCleaned and Munged Dataset:\n", df1)

3. Identify and handle outliers in the dataset and explain how you handled them.

We first have to visualize Box plots for numerical columns before handling outliers like this code.

```
plt.figure(figsize=(12, 8))
sns.boxplot(data=df[['WEIGHT_KG', 'HEIGHT_METERS', 'WAIST_CIRCUMFERENCE(CM)',
'HIP_CIRCUMFERENCE(CM)', 'AGE']])
plt.title('Box Plots of Numerical Columns Before Handling Outliers')
plt.show()
```

Then

- We calculate z-scores for each numerical column. The z-score represents how many standard deviations away a data point is from the mean.
  We calculate z-scores for each numerical column. The z-score represents how many standard deviations away a data point is from the mean.
- We set a threshold of 3, and data points with z-scores greater than 3 (considered extreme) are identified as outliers.
- Outliers are then replaced with the median values of their respective columns.

```
 z_scores = np.abs(stats.zscore(df[['WEIGHT_KG', 'HEIGHT_METERS', 'WAIST_CIRCUMFERENCE(CM)',
'HIP_CIRCUMFERENCE(CM)', 'AGE']]))
```

outliers = (z_scores > 3).all(axis=1)

median_values_df1 = df1[['WEIGHT_KG', 'HEIGHT_METERS', 'WAIST_CIRCUMFERENCE(CM)', 'HIP_CIRCUMFERENCE(CM)', 'AGE']].median()

df1.loc[outliers_df1, ['WEIGHT_KG', 'HEIGHT_METERS', 'WAIST_CIRCUMFERENCE(CM)', 'HIP_CIRCUMFERENCE(CM)', 'AGE']] = median_values_df1

At the end We generate new box plots after handling outliers to observe the changes in the data distribution.

4.Investigate the correlation between WEIGHT_KG and HEIGHT_METERS and what does it imply about their relationship

correlation_coefficient = df['WEIGHT_KG'].corr(df['HEIGHT_METERS'])

# Display the correlation coefficient
print(f'Correlation Coefficient between WEIGHT_KG and HEIGHT_METERS: {correlation_coefficient:.2f}')

- If the correlation coefficient is close to 1, it implies a strong positive linear relationship. As weight increases, height tends to increase, and vice versa.
- If the correlation coefficient is close to -1, it implies a strong negative linear relationship. As weight increases, height tends to decrease, and vice versa.
- If the correlation coefficient is close to 0, it implies a weak or no linear relationship.

5. p erform data integration and create a single dataset (combine the second and first dataset.)

Here is the Merged Dataset. I converted the values from Dataset 2 to Metric units

WEIGHT_KG	HEIGHT_METERS	WAIST_CIRCUMFERENCE(CM)	HIP_CIRCUMFERENCE(CM)	AGE	GENDER	OBESITY
70	1.75	80	95	30	M	No
85	1.6	90	100	45	F	YES
60	1.8	75	85	28	M	NO
92	1.65	98	105	50	F	YES
75	1.7	85	97	35	M	NA
68	1.68	78	92	42	F	NO
78	1.72	88	unknown	33	M	NO
90	1.75	95	110	55	F	YES
72	1.78	82	98	40	M	NO
79	1.63	87	96	48	F	YES
65	1.69	76	89	31	M	NO
88	1.55	92	1001	43	F	YES
73	1.79	84	99	38	M	NO
82	1.62	89	104	46	F	YES
77	0.74	86	100	36	M	NO
70	1.73	79	94	30	M	NO
85	1.6	89	99	45	F	YES
60	1.8	76	84	28	M	NO
92	1.65	99	104	50	F	YES
75	1.7	84	97	35	M	NO

6.Using the following scatterplots, determine the relationship between the two variables in the plot.

A: Relationship between Speed and distance is a Strong and Positive Relationship where R=0.7 (70%)

b: There is no Relationship found between nControls and cCases. R=0

c: the relation between cty and “word I couldn’t reed because of the doc” is strong negative relationship r=-0