```
'''
At the start of the Lab,  in the Observation book, Write python code for the following considering  filename as “housing.csv”
 i. To load .csv file into the data frame
ii. To display information of all columns
iii. To display statistical information of all numerical
iv. To display the count of unique labels for “Ocean Proximity” column
v. To display which attributes (columns) in a dataset have missing values count greater than zero
Step-2: Show the observation book to lab batch faculty incharge.
Step-3: Do the "To Do" tasks given in the PPT
Step-4: At the end of the lab,
i. Write the answers for questions given in the PPT and show it to  lab batch faculty incharge
ii. Should upload the code in your respective GitHub account.
File name format:yourUSN_Lab-1-DataProcessing.ipynb
'''


import pandas as pd
filename = "/content/housing.csv"
df = pd.read_csv(filename)

print("Dataset Information:")
print(df.info())

print("\nStatistical Summary of Numerical Columns:")
print(df.describe())

if "ocean_proximity" in df.columns:
    print("\nUnique Value Counts for 'Ocean Proximity':")
    print(df["ocean_proximity"].value_counts())
else:
    print("\n'Ocean Proximity' column not found in the dataset.")

missing_values = df.isnull().sum()
missing_columns = missing_values[missing_values > 0]

if not missing_columns.empty:
    print("\nColumns with Missing Values:")
    print(missing_columns)
else:
    print("\nNo missing values found in the dataset.")
```

```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object 
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
None

Statistical Summary of Numerical Columns:
          longitude      latitude  housing_median_age   total_rooms  \
count  20640.000000  20640.000000        20640.000000  20640.000000   
mean    -119.569704     35.631861           28.639486   2635.763081   
std        2.003532      2.135952           12.585558   2181.615252   
min     -124.350000     32.540000            1.000000      2.000000   
25%     -121.800000     33.930000           18.000000   1447.750000   
50%     -118.490000     34.260000           29.000000   2127.000000   
75%     -118.010000     37.710000           37.000000   3148.000000   
max     -114.310000     41.950000           52.000000  39320.000000   

       total_bedrooms    population    households  median_income  \
count    20433.000000  20640.000000  20640.000000   20640.000000   
mean       537.870553   1425.476744    499.539680       3.870671   
std        421.385070   1132.462122    382.329753       1.899822   
min          1.000000      3.000000      1.000000       0.499900   
25%        296.000000    787.000000    280.000000       2.563400   
50%        435.000000   1166.000000    409.000000       3.534800   
```

```
75%        647.000000   1725.000000    605.000000       4.743250
max       6445.000000  35682.000000   6082.000000      15.000100

       median_house_value
count        20640.000000
mean        206855.816909
std         115395.615874
min          14999.000000
25%         119600.000000
50%         179700.000000
75%         264725.000000
max         500001.000000

Unique Value Counts for 'Ocean Proximity':
ocean_proximity
<1H OCEAN     9136
INLAND        6551
NEAR OCEAN    2658
NEAR BAY      2290
ISLAND           5
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder, StandardScaler, MinMaxScaler
from scipy import stats

# Load dataset
file_path = "/content/Dataset_with_Nulls.csv"
df = pd.read_csv(file_path)

# Display initial information
df.info()
print(df.head())

# Handling missing values
imputer_median = SimpleImputer(strategy="median")
imputer_mean = SimpleImputer(strategy="mean")

df["AGE"] = imputer_median.fit_transform(df[["AGE"]])
df["BMI"] = imputer_mean.fit_transform(df[["BMI"]])
df["Urea"] = imputer_mean.fit_transform(df[["Urea"]])
df["Cr"] = imputer_mean.fit_transform(df[["Cr"]])
df["HbA1c"] = imputer_mean.fit_transform(df[["HbA1c"]])
df["Chol"] = imputer_mean.fit_transform(df[["Chol"]])
df["TG"] = imputer_mean.fit_transform(df[["TG"]])
df["HDL"] = imputer_mean.fit_transform(df[["HDL"]])
df["LDL"] = imputer_mean.fit_transform(df[["LDL"]])
df["VLDL"] = imputer_mean.fit_transform(df[["VLDL"]])

# Encoding categorical data
ordinal_encoder = OrdinalEncoder()
df["Gender_Encoded"] = ordinal_encoder.fit_transform(df[["Gender"]].fillna("Unknown"))

df = pd.get_dummies(df, columns=["CLASS"], prefix="Class")


# Normalization and Standardization
normalizer = MinMaxScaler()
df[['BMI', 'Urea', 'Chol']] = normalizer.fit_transform(df[['BMI', 'Urea', 'Chol']])

scaler = StandardScaler()
df[['AGE', 'HbA1c']] = scaler.fit_transform(df[['AGE', 'HbA1c']])

# Outlier Handling using IQR
Q1 = df['TG'].quantile(0.25)
Q3 = df['TG'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df['TG'] = np.where(df['TG'] > upper_bound, upper_bound,
                    np.where(df['TG'] < lower_bound, lower_bound, df['TG']))

# Outlier Handling using Z-score
df['TG_Zscore'] = stats.zscore(df['TG'])
df['TG'] = np.where(df['TG_Zscore'].abs() > 3, np.nan, df['TG'])
df.drop(columns=["TG_Zscore"], inplace=True)
```

```
# Outlier Handling using Median Replacement
median_tg = df['TG'].median()
df['TG'] = df['TG'].fillna(median_tg)

# Final Data Preview
print(df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   ID         900 non-null    float64
 1   No_Pation  902 non-null    float64
 2   Gender     914 non-null    object 
 3   AGE        889 non-null    float64
 4   Urea       902 non-null    float64
 5   Cr         906 non-null    float64
 6   HbA1c      905 non-null    float64
 7   Chol       908 non-null    float64
 8   TG         895 non-null    float64
 9   HDL        909 non-null    float64
 10  LDL        909 non-null    float64
 11  VLDL       908 non-null    float64
 12  BMI        892 non-null    float64
 13  CLASS      917 non-null    object 
dtypes: float64(12), object(2)
memory usage: 109.5+ KB
      ID  No_Pation Gender   AGE  Urea    Cr  HbA1c  Chol   TG  HDL  LDL  \
0    NaN        NaN      F  50.0   4.7  46.0    4.9   4.2  0.9  2.4  1.4   
1  735.0    34221.0      M  26.0   4.5  62.0    4.9   3.7  1.4  1.1  2.1   
2  420.0    47975.0      F  50.0   4.7  46.0    4.9   4.2  0.9  2.4  1.4   
3  680.0    87656.0      F  50.0   4.7  46.0    4.9   4.2  NaN  2.4  1.4   
4  504.0        NaN      M  33.0   7.1  46.0    4.9   4.9  1.0  NaN  2.0   

   VLDL   BMI CLASS  
0   0.5  24.0     N  
1   0.6   NaN     N  
2   0.5  24.0     N  
3   0.5  24.0     N  
4   0.4  21.0     N  
      ID  No_Pation Gender       AGE      Urea    Cr     HbA1c      Chol  \
0    NaN        NaN      F -0.451293  0.109375  46.0 -1.393028  0.407767   
1  735.0    34221.0      M -3.378602  0.104167  62.0 -1.393028  0.359223   
2  420.0    47975.0      F -0.451293  0.109375  46.0 -1.393028  0.407767   
3  680.0    87656.0      F -0.451293  0.109375  46.0 -1.393028  0.407767   
4  504.0        NaN      M -2.524803  0.171875  46.0 -1.393028  0.475728   

         TG       HDL  LDL  VLDL       BMI  Gender_Encoded  Class_N  Class_N   \
0  0.900000  2.400000  1.4   0.5  0.173913             0.0     True     False   
1  1.400000  1.100000  2.1   0.6  0.367724             1.0     True     False   
2  0.900000  2.400000  1.4   0.5  0.173913             0.0     True     False   
3  2.337553  2.400000  1.4   0.5  0.173913             0.0     True     False   
4  1.000000  1.210451  2.0   0.4  0.069565             1.0     True     False   

   Class_P  Class_Y  Class_Y   
0    False    False     False  
1    False    False     False  
2    False    False     False  
3    False    False     False  
4    False    False     False  
```

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.