Installing the ucimlrepo library

```
pip install ucimlrepo
```

```
Requirement already satisfied: ucimlrepo in /usr/local/lib/python3.10/dist-packages (0.0.3)
```

Importing the data

```
from ucimlrepo import fetch_ucirepo

# fetching the data from dataset
bank_marketing = fetch_ucirepo(id=222)

# data (as pandas dataframes)
X = bank_marketing.data.features
y = bank_marketing.data.targets

# metadata
print(bank_marketing.metadata)

# variable information
print(bank_marketing.variables)
```

```
{'uci_id': 222, 'name': 'Bank Marketing', 'repository_url': 'https://archive.ics.uci.edu/dataset/222/bank+marketing', 'd
           name     role         type      demographic  \
0           age  Feature      Integer              Age   
1           job  Feature  Categorical       Occupation   
2       marital  Feature  Categorical   Marital Status   
3     education  Feature  Categorical  Education Level   
4       default  Feature       Binary             None   
5       balance  Feature      Integer             None   
6       housing  Feature       Binary             None   
7          loan  Feature       Binary             None   
8       contact  Feature  Categorical             None   
9   day_of_week  Feature         Date             None   
10        month  Feature         Date             None   
11     duration  Feature      Integer             None   
12     campaign  Feature      Integer             None   
13        pdays  Feature      Integer             None   
14     previous  Feature      Integer             None   
15     poutcome  Feature  Categorical             None   
16            y   Target       Binary             None   

                                          description  units missing_values
0                                                None   None             no
1   type of job (categorical: 'admin.','blue-colla...   None             no
2   marital status (categorical: 'divorced','marri...   None             no
3   (categorical: 'basic.4y','basic.6y','basic.9y'...   None             no
4                                 has credit in default?   None             no
5                                average yearly balance  euros             no
6                                     has housing loan?   None             no
7                                    has personal loan?   None             no
8   contact communication type (categorical: 'cell...   None            yes
9                              last contact day of the week   None             no
10  last contact month of year (categorical: 'jan'...   None             no
11   last contact duration, in seconds (numeric). ...   None             no
12  number of contacts performed during this campa...   None             no
13  number of days that passed by after the client...   None            yes
14  number of contacts performed before this campa...   None             no
15  outcome of the previous marketing campaign (ca...   None            yes
16          has the client subscribed a term deposit?   None             no
```

Fetching the Data

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Fetching dataset
bank_marketing = fetch_ucirepo(id=222)

# Data (as pandas dataframes)
K = bank_marketing.data.features
M = bank_marketing.data.targets

# Combine features and targets into one DataFrame
bank_data = pd.concat([X, pd.DataFrame(y, columns=["target"])], axis=1)
```

```
Descriptive Statistics for Quantitative Variables:
                age        balance   day_of_week      duration      campaign  \
count  45211.000000   45211.000000  45211.000000  45211.000000  45211.000000   
mean      40.936210    1362.272058     15.806419    258.163080      2.763841   
std       10.618762    3044.765829      8.322476    257.527812      3.098021   
min       18.000000   -8019.000000      1.000000      0.000000      1.000000   
25%       33.000000      72.000000      8.000000    103.000000      1.000000   
50%       39.000000     448.000000     16.000000    180.000000      2.000000   
75%       48.000000    1428.000000     21.000000    319.000000      3.000000   
max       95.000000  102127.000000     31.000000   4918.000000     63.000000   

              pdays      previous  target  
count  45211.000000  45211.000000     0.0  
mean      40.197828      0.580323     NaN  
std      100.128746      2.303441     NaN  
min       -1.000000      0.000000     NaN  
25%       -1.000000      0.000000     NaN  
50%       -1.000000      0.000000     NaN  
75%       -1.000000      0.000000     NaN  
max      871.000000    275.000000     NaN  

Value Counts for Categorical Variables:
blue-collar      9732
management       9458
technician       7597
admin.           5171
services         4154
retired          2264
self-employed    1579
entrepreneur     1487
unemployed       1303
housemaid        1240
student           938
Name: job, dtype: int64
secondary    23202
tertiary     13301
primary       6851
Name: education, dtype: int64
```

Descriptive statistics for a selection of quantitative and categorical variables.

```
# Combine features and targets into one DataFrame
bank_data = pd.concat([X, pd.DataFrame(y, columns=["target"])], axis=1)

# Descriptive statistics for a selection of quantitative and categorical variables
print("Descriptive Statistics for Quantitative Variables:")
print(bank_data.describe())

print("\nValue Counts for Categorical Variables:")
print(bank_data["job"].value_counts())
print(bank_data["education"].value_counts())
```

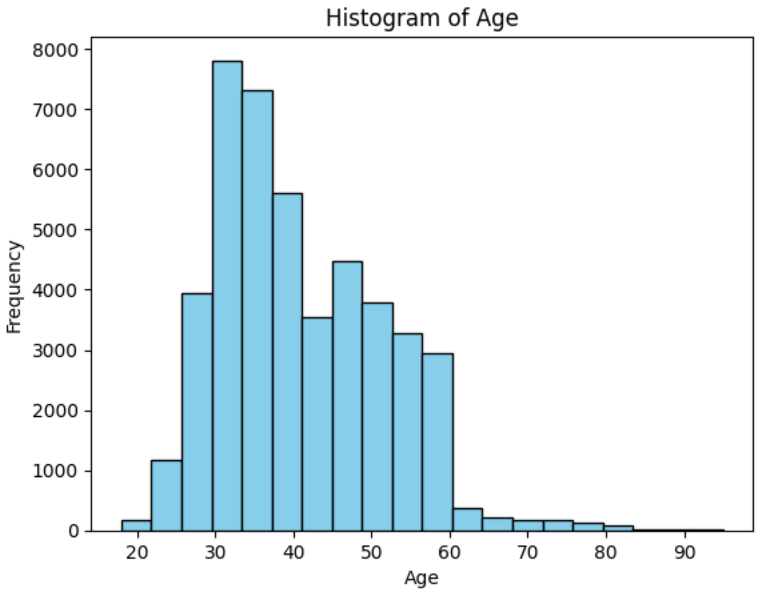
Transformation of variable

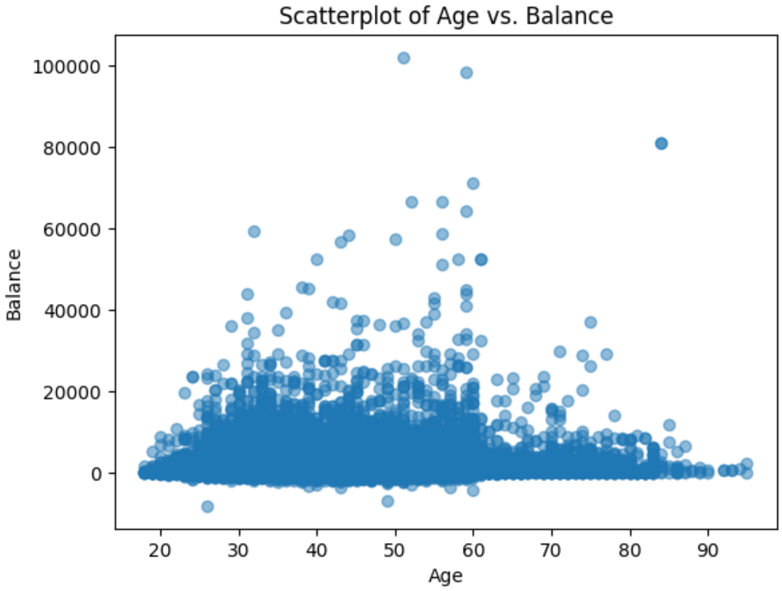
```
# Transform at least one variable (e.g., balance)
bank_data["transformed_balance"] = bank_data["balance"] ** 0.5
```

Plotting the quantitative variable and a scatterplot

```
# Plot at least one quantitative variable (e.g., age histogram)
plt.hist(bank_data["age"], bins=20, color="skyblue", edgecolor="black")
plt.title("Histogram of Age")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.show()

# Plot a scatterplot (e.g., age vs. balance)
plt.scatter(bank_data["age"], bank_data["balance"], alpha=0.5)
plt.title("Scatterplot of Age vs. Balance")
plt.xlabel("Age")
plt.ylabel("Balance")
plt.show()
```


Histogram of Age
Frequency
Age


Scatterplot of Age vs. Balance
Balance
Age