Installing the necessary libraries

```
!pip install ucimlrepo
!pip install pandas
!pip install matplotlib
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.3-py3-none-any.whl (7.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.3
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.23.5)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.47.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotli
```

Loading the data

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
bank_information = fetch_ucirepo(id=222)

# data (as pandas dataframes)
X = bank_information.data.features

# metadata
print(bank_information.metadata)

# variable information
print(bank_information.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
```

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


# Fetch dataset
bank_marketing = fetch_ucirepo(id=222)

# Data (as pandas dataframes)
X = pd.DataFrame(bank_information.data.features)

bank_data = X
```

```
bank_data.head(5)
```

	age	job	marital	education	default	balance	housing	loan	contact
0	58	management	married	tertiary	no	2143	yes	no	NaN
1	44	technician	single	secondary	no	29	yes	no	NaN
2	33	entrepreneur	married	secondary	no	2	yes	yes	NaN
3	47	blue-collar	married	NaN	no	1506	yes	no	NaN

To know the descriptive statistics for Quantitative and Categorical features present in the dataset

```
print("Variables present in the data " ,bank_data.columns)
qualitative_variables = bank_data.select_dtypes(include='object').columns.tolist()
print("Qualitative varibles present in the data ",qualitative_variables)
quantitative_variables = bank_data.select_dtypes(include=['int', 'float']).columns.tolist()
print("Quantitative variables present in the data " ,quantitative_variables)
```

```
Variables present in the data  Index(['age', 'job', 'marital', 'education', 'default', 'balance', 'housing',
       'loan', 'contact', 'day_of_week', 'month', 'duration', 'campaign',
       'pdays', 'previous', 'poutcome'],
      dtype='object')
Qualitative varibles present in the data  ['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'mont
Quantitative variables present in the data  ['age', 'balance', 'day_of_week', 'duration', 'campaign', 'pdays', 'previous
```

```
print("Descriptive statistics of Qualitative variables")
bank_data[qualitative_variables].describe()
```

Descriptive statistics of Qualitative variables

	job	marital	education	default	housing	loan	contact	month	poutc
count	44923	45211	43354	45211	45211	45211	32191	45211	8
unique	11	3	3	2	2	2	2	12	
top	blue-collar	married	secondary	no	yes	no	cellular	may	fa

```
print("Descriptive statistics of Quantitative variables")
bank_data[quantitative_variables].describe()
```

```
Descriptive statistics of Quantitative variables
```

	age	balance	day_of_week	duration	campaign	pda
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000(
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000(
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000(
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000(
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000(

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

```
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
```

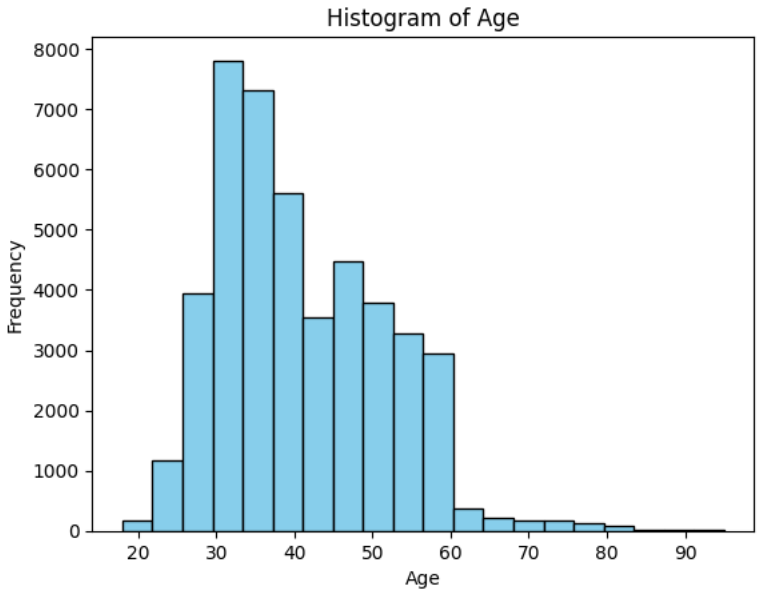
Transformation of variable

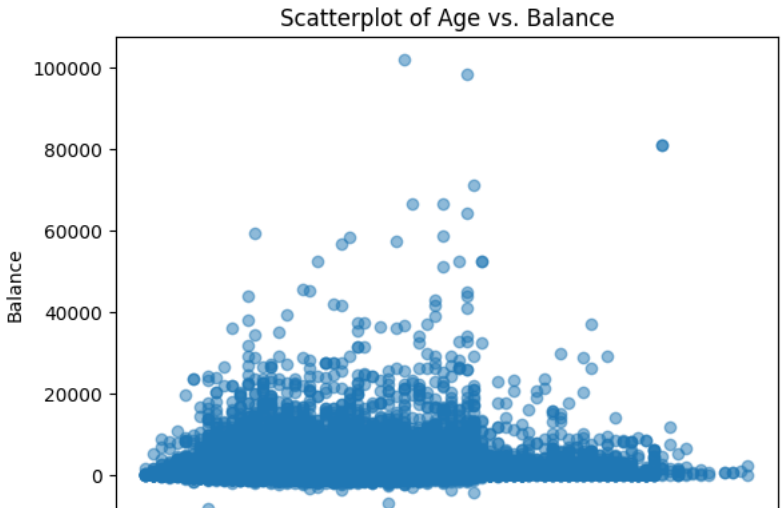
```
bank_data["transformed_balance"] = bank_data["balance"] ** 0.5
```

Plotting

```
# 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
0
1000
2000
3000
4000
5000
6000
7000
8000
20
30
40
50
60
70
80
90


Scatterplot of Age vs. Balance
Balance
0
20000
40000
60000
80000
100000