import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd

df = pd.read\_csv('banking\_data.csv')

print(df.head)

plt.hist(df['age'], color='blue', edgecolor='black')

plt.xlabel('Age')

plt.ylabel('Number of people')

plt.title('Age Distribution')

plt.show()

sns.countplot(x='job', data=df)

plt.xlabel('Jobs')

plt.ylabel('Number of people')

plt.title('Job Distribution')

plt.show()

sns.countplot(x='marital', data=df)

plt.xlabel('Marital Status')

plt.ylabel('Frequency')

plt.title('Marital Status Distribution')

plt.show()

sns.countplot(x='education', data=df)

plt.xlabel('Education')

plt.ylabel('Number of People')

plt.title('Education level among clients')

plt.show()

default\_count = df['default'].value\_counts()

proportion = default\_count['yes'] / (df['default'].count())

print(f'Proportion of clients defaulting their credit card payment: {proportion:.5f}')

#Changing yes and no to 0 and 1 for further studies

df['default'] = df['default'].replace({'yes': 1, 'no': 0})

df['default'] = df['default'].infer\_objects(copy=False)

plt.hist(df['balance'])

plt.xlabel('Average Yearly balance')

plt.ylabel('Number of People')

plt.title('Average yearly balance of client')

plt.show()

housing\_count = df['housing'].value\_counts()

print(f"Clients having Housing loan: {housing\_count['yes']}")

#Changing yes and no to 0 and 1 for further studies

df['housing'] = df['housing'].replace({'yes': 1, 'no': 0})

df['housing'] = df['housing'].infer\_objects(copy=False)

personal\_loan\_count = df['loan'].value\_counts()

print(f"Clients having Personal Loan: {personal\_loan\_count['yes']}")

#Changing yes and no to 0 and 1 for further studies

df['loan'] = df['loan'].replace({'yes': 1, 'no': 0})

df['loan'] = df['loan'].infer\_objects(copy=False)

communication\_type = df['contact'].value\_counts()

print(f"Communication Types: {communication\_type.index.tolist()}")

plt.hist(df['day'], bins=31)

plt.xlabel('Last contact day')

plt.ylabel('Number of clients')

plt.title('Last contact day Distribution')

plt.show()

sns.countplot(x='month', data = df)

plt.xlabel('last contact month')

plt.ylabel('Number of clients')

plt.title('Variation of last contact month among clients')

plt.show()

plt.hist(df['duration'], bins=60)

plt.xlabel('Duration of last contact')

plt.ylabel('Number of clients')

plt.title('Distribution of duration of last contact')

plt.show()

contacts\_performed\_count = df['campaign'].value\_counts()

sns.countplot(x='campaign', data=df)

plt.xlabel('Contacts performed')

plt.ylabel('Number of clients')

plt.title('Contacts performed during a campaign for each client')

plt.show()

filtered\_pdays = df[df['pdays'] != -1]

plt.hist(filtered\_pdays['pdays'], bins=60)

plt.xlabel('Number of days passed')

plt.ylabel('Number of clients')

plt.title('Number of days passed since the client was last contacted from a previous campaign')

plt.show()

sns.countplot(x='poutcome', data=df)

plt.xlabel('Outcome')

plt.ylabel('Number of clients')

plt.title('Outcome of previous marketing campaign')

plt.show()

sns.countplot(x='y', data=df)

plt.xlabel('Subscribed to term deposit')

plt.ylabel('Number of clients')

plt.title('Status of client subscribed to term deposit')

plt.show()

df['y'] = df['y'].replace({'no': 0, 'yes': 1})

df['y'] = df['y'].infer\_objects(copy=False)

numeric\_df = df.select\_dtypes(['int64', 'float64'])

corr\_matrix = numeric\_df.corr()

plt.figure(figsize=(10, 8))

sns.heatmap(corr\_matrix, annot=True, cmap='PuBuGn', fmt = ".2f")

plt.title('Correlation heatmap of Banking database')

plt.show()