```
                Data Analysis Project - Code Transcript
                                Banking

#Banking Dataset Analysis

#Importing appropriate libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


df = pd.read_csv('banking_data.csv')
print(df.head())
print(df.shape)
print(df.isnull().sum())

#Before analysis we should make the dataframe optimal
#Since marital and marital_status are alike, we will remove one
df.drop(columns='marital', inplace=True)

#both marital_status and education columns have three null entries. So, we
will use deletion to delete these rows.
#Since the no. of faulty rows is very less compared to the total row size,
deletion will not cause any bias
df.dropna(subset=['marital_status','education'], inplace = True)

#Q1. What is the distribution of age among the clients?
print(df['age'].value_counts())
print('Logistic of age of clients')
print(df['age'].describe())
#plot
f = plt.figure()
f.set_figwidth(100)
f.set_figheight(7)
sns.countplot(x= 'age', data= df)
plt.title('Age Distribution of Clients')
plt.show()

#Q2. How does the job type vary among the clients?
```

```
print(df['job'].value_counts())
#plot
f = plt.figure()
f.set_figwidth(100)
sns.countplot(x= 'job', data= df)
plt.title('Job Distribution of Clients')
plt.show()

#Q3. What is the marital status distribution of the clients?
print(df['marital_status'].value_counts())
#plot
sns.countplot(x= 'marital_status', data= df)
plt.title('Marital Status of Clients')
plt.show()

#Q4. What is the level of education among the clients?
print(df['education'].value_counts())
#plot
sns.countplot(x= 'education', data= df)
plt.title('Education Status of Clients')
plt.show()

#Q5. What proportion of clients have credit in default?
df['default'] = df['default'].replace({'yes': 1, 'no': 0})
print(df['default'].value_counts())
print('Percentage of clients having credit in
default',(df['default'].value_counts()[1]/df['default'].size)*100)

#Q6. What is the distribution of average yearly balance among the clients?
print(df['balance'].value_counts())
print('Logistic of average yearly balance of the clients')
print(df['balance'].describe())
#plot
f = plt.figure()
f.set_figwidth(100)
plt.hist(df['balance'], bins=100)
plt.xlabel('Average Yearly Balance in Euros')
plt.ylabel('Frequency')
plt.title('Yearly Balance Distribution')
plt.show()
```

```
#Q7. How many clients have housing loans?
print("No. of clients with housing loans",
df['housing'].value_counts()['yes'])

#Q8. How many clients have personal loans?
print("No. of clients with personal loans",
df['loan'].value_counts()['yes'])

#9. What are the communication types used for contacting clients during
the campaign?
print('Contact types used along with their count')
print(df['contact'].value_counts())

#10. What is the distribution of the last contact day of the month?
print('Last contact days count')
print(df['day'].value_counts())
print("Last contact day logistics")
print(df['day'].describe())
#plot
sns.countplot(x= 'day', data= df)
plt.title('Last Contact Day of the Month for Clients')
plt.show()

#11. How does the last contact month vary among the clients?
print(df['month'].value_counts())
#plot
sns.countplot(x= 'month', data= df)
plt.title('Last Contact Month distribution of Clients')
plt.show()

#12. What is the distribution of the duration of the last contact?
print(df['duration'].value_counts())
print('Duration of Last Contact Distribution')
print(df['duration'].describe())
#plot
f = plt.figure()
f.set_figwidth(100)
plt.hist(df['duration'], bins=100)
plt.xlabel('Duration of Last Contact')
```

```
plt.ylabel('Frequency of Clients')
plt.title('Duration of Last Contact Distribution')
plt.show()

#13. How many contacts were performed during the campaign for each client?
print(df['campaign'].value_counts())
print('Logistics of no. of contacts performed during the campaign for each client')
print(df['campaign'].describe())
#plot
f = plt.figure()
f.set_figwidth(100)
sns.countplot(x= 'campaign', data= df)
plt.title('No. of contacts performed during the campaign for clients distribution')
plt.show()

#14. What is the distribution of the number of days passed since the client was last contacted from a previous campaign?
print(df['pdays'].value_counts())
print('Logistics of no. of days passed since the client was last contacted from a previous campaign')
print(df['pdays'].describe())
#plot
f = plt.figure()
f.set_figwidth(100)
plt.hist(df['pdays'], bins=100)
plt.xlabel('Number of days passed since the client was last contacted from a previous campaign')
plt.ylabel('Frequency of Clients')
plt.title('Distribution of the number of days passed since the client was last contacted from a previous campaign')
plt.show()

#15. How many contacts were performed before the current campaign for each client?
print(df['previous'].value_counts())
print('Logistics of no. of contacts that were performed before the current campaign for each client')
print(df['previous'].describe())
```

```
#plot
f = plt.figure()
f.set_figwidth(100)
plt.hist(df['previous'], bins=100)
plt.xlabel('Number of contacts that were performed before the current
campaign for each client')
plt.ylabel('Frequency of Clients')
plt.title('Distribution of previous contact count for each client')
plt.show()

#16. What were the outcomes of the previous marketing campaigns?
print("Count of various outcomes of previous campaign")
print(df['poutcome'].value_counts())
print("Percentage of successful contacting",
(df['poutcome'].value_counts()['success']/df['poutcome'].size)*100)

#17. What is the distribution of clients who subscribed to a term deposit
vs. those who did not?
print('Count of clients who subscribed to a term deposit vs. those who did
not')
print(df['y'].value_counts())
#plot
sns.countplot(x= 'y', data= df)
plt.title('Distribution of clients who subscribed to a term deposit vs.
those who did not')
plt.show()

#18. Are there any correlations between different attributes and the
likelihood of subscribing to a term deposit?
#here we will use correlation matrix
#first we will convert all the sting columns into categorical values so
that they can be correlated
for i in df:
    df[i] = df[i].astype('category').cat.codes

#correlation matrix
corr_matrix= df.corr()
plt.figure(figsize=(10,8))
sns.heatmap(corr_matrix, annot=True, cmap= 'PuBuGn', fmt='.2f')
plt.show()
```

```
print('As we can observe from the correlation matrix, the duration of last contact is moderately related to likelihood of clients subscribing to the terms deposit, with the index being 0.41 which is the largest in all categories. This is the only possible correlation present for the subscription.')
```