PROJECT: BANKING DATA

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns


# Load the dataset

file_path = 'downloads/banking_data.csv'

data = pd.read_csv(file_path)


# Set style for the plots

sns.set(style="whitegrid")


# Question 1: Age distribution

plt.figure(figsize=(8, 6))

sns.histplot(data['age'], kde=True, bins=30, color='skyblue')

plt.title('Distribution of Age among Clients')

plt.xlabel('Age')

plt.ylabel('Frequency')

plt.show()


# Question 2: Job type distribution

plt.figure(figsize=(10, 6))

job_distribution = data['job'].value_counts()

job_distribution.plot(kind='bar', color='teal')
```

```
plt.title('Job Type Distribution')
plt.xlabel('Job Type')
plt.ylabel('Number of Clients')
plt.xticks(rotation=45)
plt.show()

# Question 3: Marital status distribution
plt.figure(figsize=(8, 6))
marital_distribution = data['marital'].value_counts()
marital_distribution.plot(kind='bar', color='coral')
plt.title('Marital Status Distribution')
plt.xlabel('Marital Status')
plt.ylabel('Number of Clients')
plt.show()

# Question 4: Education level distribution
plt.figure(figsize=(8, 6))
education_distribution = data['education'].value_counts()
education_distribution.plot(kind='bar', color='purple')
plt.title('Education Level Distribution')
plt.xlabel('Education Level')
plt.ylabel('Number of Clients')
plt.show()

# Question 5: Credit default proportion
plt.figure(figsize=(6, 6))
```

```
default_proportion = data['default'].value_counts()
default_proportion.plot(kind='pie', autopct='%1.1f%%', colors=['gold', 'lightblue'])
plt.title('Proportion of Clients with Credit in Default')
plt.ylabel('')
plt.show()
```

 # Question 6: Yearly balance distribution

```
plt.figure(figsize=(8, 6))
sns.histplot(data['balance'], kde=True, bins=30, color='green')
plt.title('Distribution of Yearly Balance')
plt.xlabel('Yearly Balance (€)')
plt.ylabel('Frequency')
plt.show()
```

 # Question 7 & 8: Housing and Personal loans

```
loan_data = data[['housing', 'loan']].melt()
plt.figure(figsize=(8, 6))
sns.countplot(data=loan_data, x='variable', hue='value', palette='Set2')
plt.title('Housing and Personal Loan Distribution')
plt.xlabel('Loan Type')
plt.ylabel('Number of Clients')
plt.legend(title='Has Loan', labels=['No', 'Yes'])
plt.show()
```

 # Question 9: Communication types

```
plt.figure(figsize=(8, 6))
```

```
communication_types = data['contact'].value_counts()
communication_types.plot(kind='bar', color='steelblue')
plt.title('Communication Types Used')
plt.xlabel('Communication Type')
plt.ylabel('Number of Clients')
plt.xticks(rotation=45)
plt.show()
```

 # Question 10: Last contact day distribution

```
plt.figure(figsize=(10, 6))
sns.countplot(data=data, x='day', color='orange')
plt.title('Last Contact Day of the Month')
plt.xlabel('Day')
plt.ylabel('Number of Clients')
plt.show()
```

 # Question 11: Last contact month distribution

```
plt.figure(figsize=(8, 6))
last_contact_month_distribution = data['month'].value_counts()
last_contact_month_distribution.plot(kind='bar', color='pink')
plt.title('Last Contact Month Distribution')
plt.xlabel('Month')
plt.ylabel('Number of Clients')
plt.xticks(rotation=45)
plt.show()
```

 # Question 12: Duration of last contact

```
plt.figure(figsize=(8, 6))
sns.histplot(data['duration'], kde=True, bins=30, color='brown')
plt.title('Distribution of Last Contact Duration')
plt.xlabel('Duration (seconds)')
plt.ylabel('Frequency')
plt.show()
```

 # Question 13: Campaign contacts

```
plt.figure(figsize=(8, 6))
campaign_contacts_distribution = data['campaign'].value_counts()
campaign_contacts_distribution.plot(kind='bar', color='darkgreen')
plt.title('Number of Contacts Performed During Campaign')
plt.xlabel('Number of Contacts')
plt.ylabel('Number of Clients')
plt.show()
```

 # Question 14: Days since last contact

```
plt.figure(figsize=(8, 6))
sns.histplot(data['pdays'], kde=False, bins=30, color='magenta')
plt.title('Distribution of Days Since Last Contact')
plt.xlabel('Days')
plt.ylabel('Frequency')
plt.show()
```

 # Question 15: Previous contacts

```
plt.figure(figsize=(8, 6))
previous_contacts_distribution = data['previous'].value_counts()
previous_contacts_distribution.plot(kind='bar', color='darkblue')
plt.title('Number of Previous Contacts Before Current Campaign')
plt.xlabel('Number of Previous Contacts')
plt.ylabel('Number of Clients')
plt.show()
```

 # Question 16: Outcomes of previous marketing campaigns

```
plt.figure(figsize=(8, 6))
previous_outcomes = data['poutcome'].value_counts()
previous_outcomes.plot(kind='bar', color='cyan')
plt.title('Outcomes of Previous Marketing Campaigns')
plt.xlabel('Outcome')
plt.ylabel('Number of Clients')
plt.show()
```

 # Question 17: Term deposit subscription

```
plt.figure(figsize=(6, 6))
term_deposit_subscription = data['y'].value_counts()
term_deposit_subscription.plot(kind='pie', autopct='%1.1f%%', colors=['red', 'lightgreen'])
plt.title('Term Deposit Subscription')
plt.ylabel('')
plt.show()
```

 # Question 18: Correlation Heatmap

```
import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd


# Filter relevant columns

correlation_data = data[['campaign', 'balance', 'y']]


# Convert the target column 'y' to numeric for correlation calculation

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


# Compute correlation matrix

correlation_matrix = correlation_data.corr()


# Visualize correlation matrix

plt.figure(figsize=(8, 6))

sns.heatmap(correlation_matrix, annot=True, fmt='.2f', cmap='coolwarm', cbar=True)

plt.title('Correlation Matrix for Campaign, Balance, and Term Deposit Subscription')

plt.show()


# Pairplot for visual analysis

sns.pairplot(correlation_data, hue='y', diag_kind='kde', palette='Set2')

plt.suptitle('Pairwise Relationships: Campaign, Balance, and Term Deposit Subscription', y=1.02)

plt.show()
```