dsbdaprac3

April 28, 2025

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
dataset = pd.read_csv('bank_data_set - bank_data_set.csv')
```

[3]:
```
dataset.columns
```

[3]:
```
Index(['age', 'job', 'marital', 'education', 'default', 'balance', 'housing',
       'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays',
       'previous', 'poutcome', 'deposit'],
      dtype='object')
```

[4]:
```
dataset.dtypes
```

[4]:
```
age           int64
job          object
marital      object
education    object
default      object
balance       int64
housing      object
loan         object
contact      object
day           int64
month        object
duration      int64
campaign      int64
pdays         int64
previous      int64
poutcome     object
deposit      object
dtype: object
```

```
[5]: dataset.head()
```

```
[5]:    age         job  marital  education default  balance housing loan  contact  \
     0   59      admin.  married  secondary      no     2343     yes   no  unknown
     1   56      admin.  married  secondary      no       45      no   no  unknown
     2   41  technician  married  secondary      no     1270     yes   no  unknown
     3   55    services  married  secondary      no     2476     yes   no  unknown
     4   54      admin.  married   tertiary      no      184      no   no  unknown

        day month  duration  campaign  pdays  previous poutcome deposit
     0    5   may      1042         1     -1         0  unknown     yes
     1    5   may      1467         1     -1         0  unknown     yes
     2    5   may      1389         1     -1         0  unknown     yes
     3    5   may       579         1     -1         0  unknown     yes
     4    5   may       673         2     -1         0  unknown     yes
```

```
[6]: dataset.shape
```

```
[6]: (11162, 17)
```

```
[7]: dataset.isnull().sum()
```

```
[7]: age          0
     job          0
     marital      0
     education    0
     default      0
     balance      0
     housing      0
     loan         0
     contact      0
     day          0
     month        0
     duration     0
     campaign     0
     pdays        0
     previous     0
     poutcome     0
     deposit      0
     dtype: int64
```

```
[8]: dataset.duplicated().sum()
```

```
[8]: 0
```

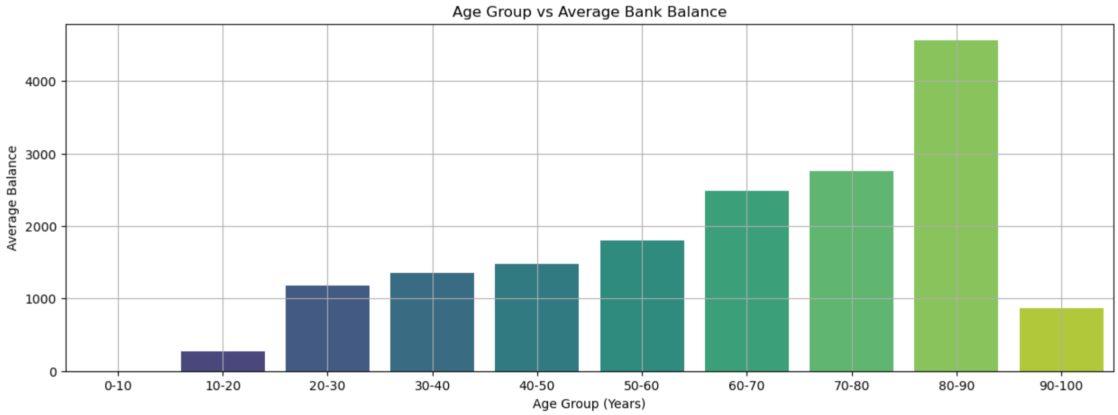
```
[9]: data = dataset.sort_values(by='age')
```

[22]:

```
bins = range(0, 101, 10)
labels = [f'{b}-{b+10}' for b in bins[:-1]]
data['age_group'] = pd.cut(data['age'], bins=bins, labels=labels, right=False)

age_group_balance = data.groupby('age_group')['balance'].mean().reset_index()
plt.figure(figsize=(15,5))
sns.barplot(x='age_group', y='balance', data=age_group_balance,␣
  ↪palette='viridis')
plt.xlabel('Age Group (Years)')
plt.ylabel('Average Balance')
plt.title('Age Group vs Average Bank Balance')
plt.grid()
plt.show()

print("Each bar represents the average balance of customers within a 10-year␣
  ↪age range.")
```

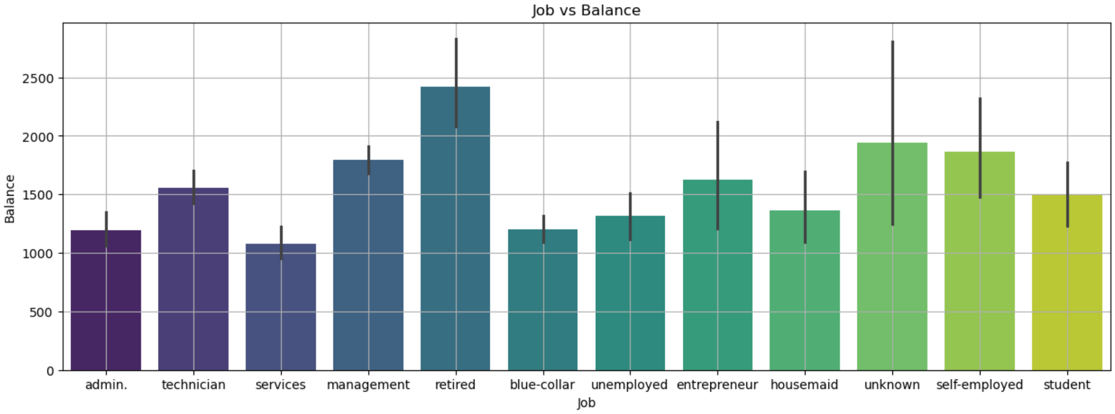


```
Each bar represents the average balance of customers within a 10-year age range.
```

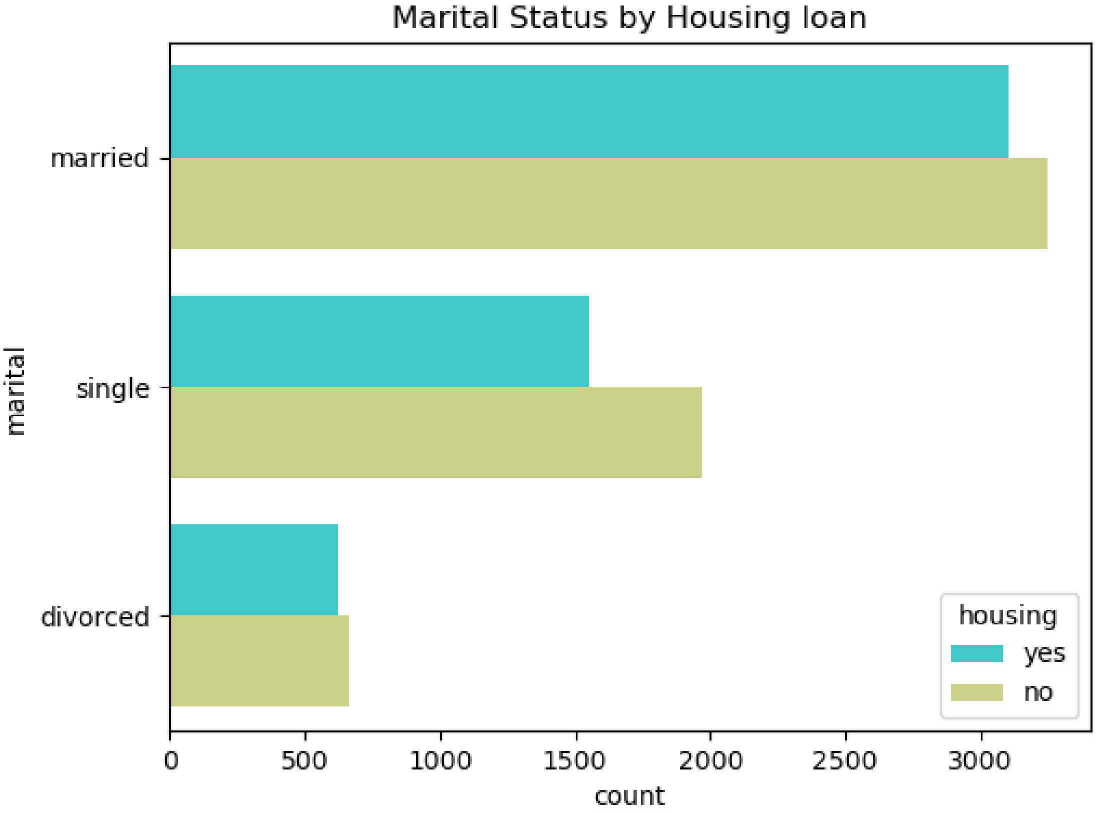
[13]:

```
plt.figure(figsize=(15,5))
sns.barplot(x=dataset['job'], y=dataset['balance'], palette='viridis')
plt.grid()
plt.xlabel('Job')
plt.ylabel('Balance')
plt.title('Job vs Balance')
print("This graph shows how various job types relate to the average bank␣
  ↪balance. It gives insight into which profession has customers with higher␣
  ↪balances.")
plt.show()
```

```
This graph shows how various job types relate to the average bank balance. It
gives insight into which profession has customers with higher balances.
```



[14]:
```
sns.countplot(y='marital', hue='housing', data=dataset, palette='rainbow')
plt.title("Marital Status by Housing loan")
plt.show()
print("This horizontal bar graph displays the marital status of customers and␣
 ↪whether they have a housing loan. It helps visualize which marital status␣
 ↪group takes more housing loans.")
```

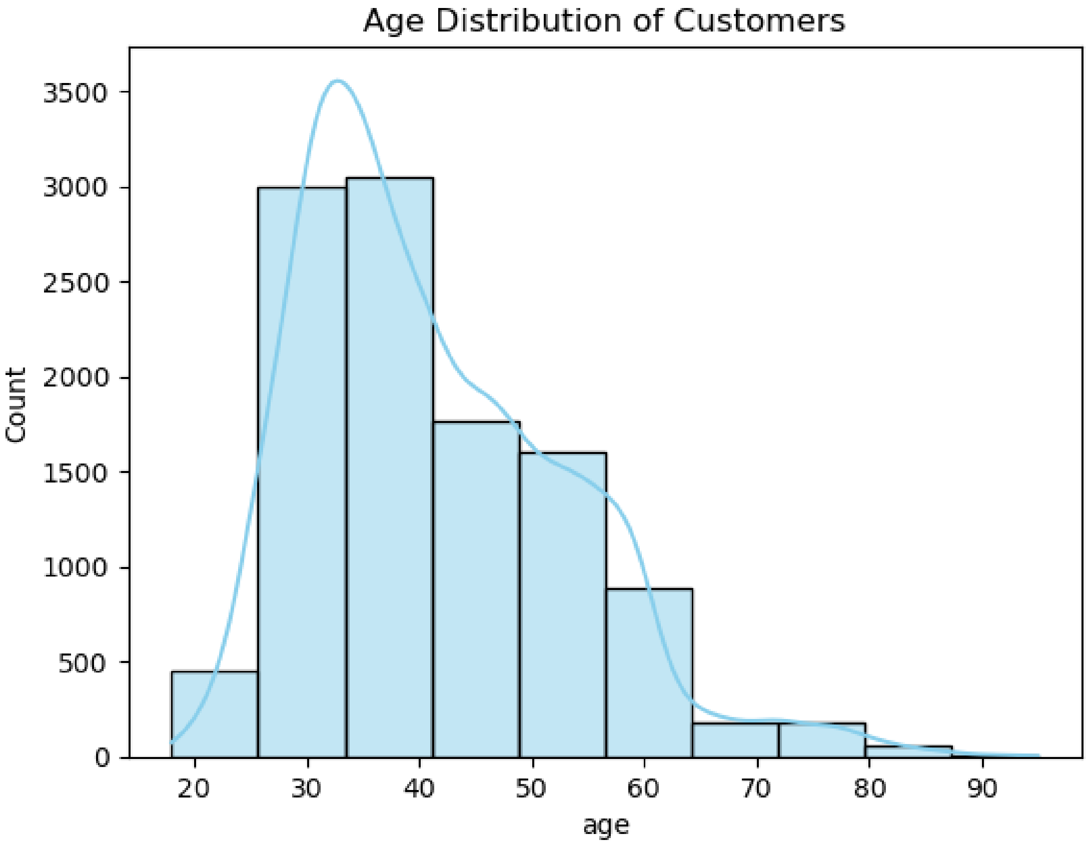


This horizontal bar graph displays the marital status of customers and whether they have a housing loan. It helps visualize which marital status group takes more housing loans.

```
[23]: print(" Visualize the age spread of customers to see which age groups dominate␣
      ↪the dataset.")
      sns.histplot(dataset['age'],kde=True,bins=10,color='skyblue')
      plt.title('Age Distribution of Customers')
```

```
 Visualize the age spread of customers to see which age groups dominate the
dataset.
```

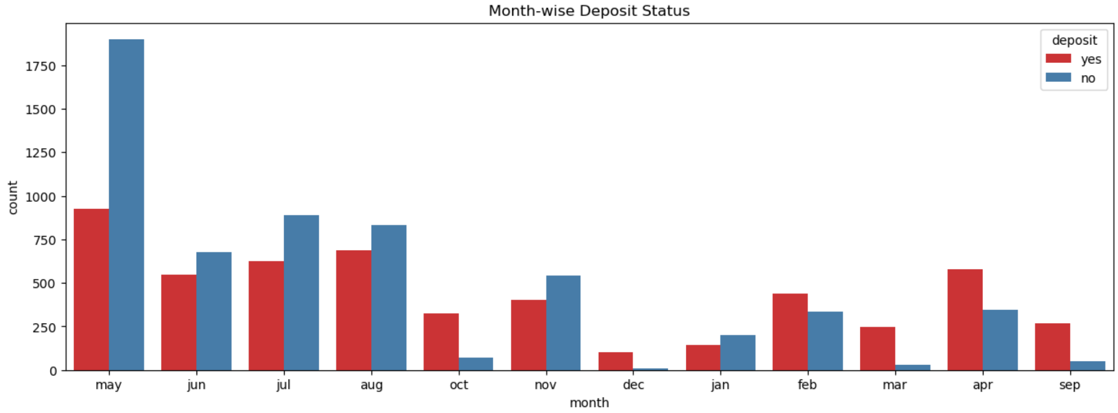
```
[23]: Text(0.5, 1.0, 'Age Distribution of Customers')
```



```
[27]: plt.figure(figsize=(15,5))
      print(" Identify which months people are most likely to deposit.")
      sns.countplot(x='month', hue='deposit', data=dataset, palette='Set1')
      plt.title('Month-wise Deposit Status')
```

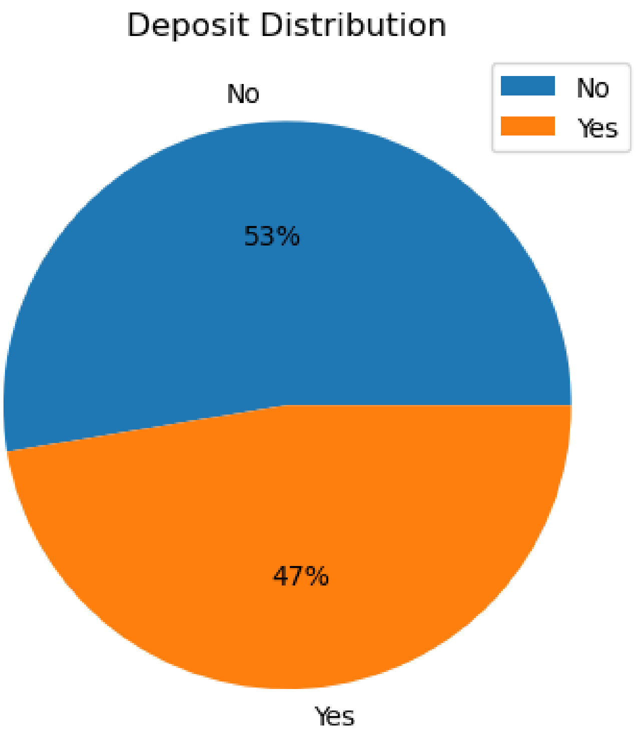
```
 Identify which months people are most likely to deposit.
```

[27]:
```
Text(0.5, 1.0, 'Month-wise Deposit Status')
```



[28]:
```python
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le.fit(dataset['deposit'])
dataset['deposit'] = le.transform(dataset['deposit'])
```

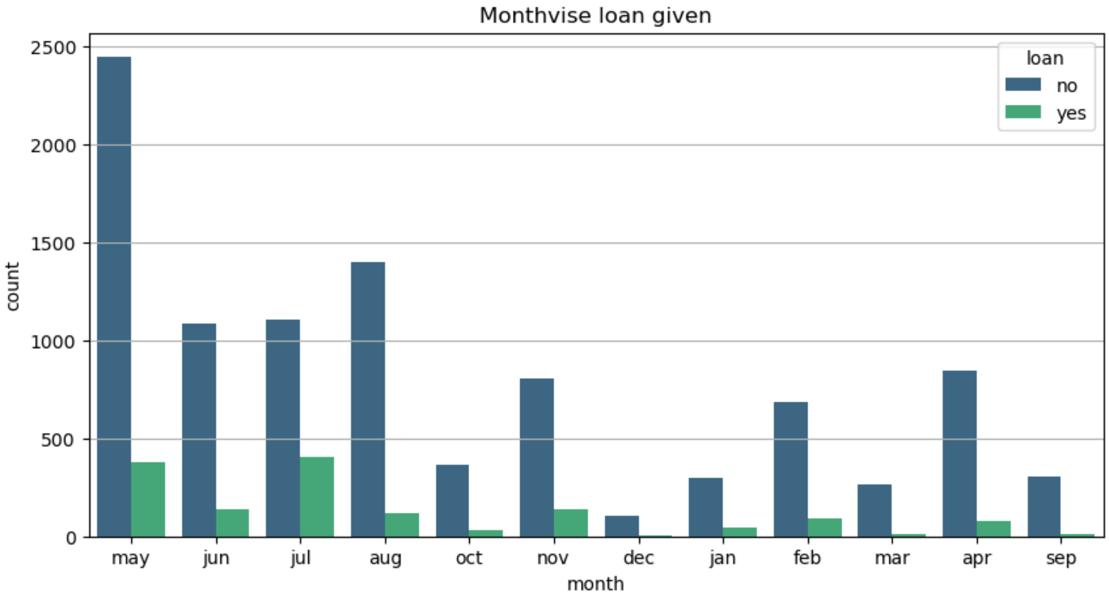
[29]:
```python
plt.pie(dataset['deposit'].value_counts(), labels=['No', 'Yes'], autopct='%1.
  ↪0f%%')
plt.title('Deposit Distribution')
plt.legend()
plt.show()
print("In this Deposit Distribution pie chat we saw:-")
print("53% have not deposited")
print("47% have deposited")
```



```
In this Deposit Distribution pie chat we saw:-
53% have not deposited
47% have deposited
```

[30]:

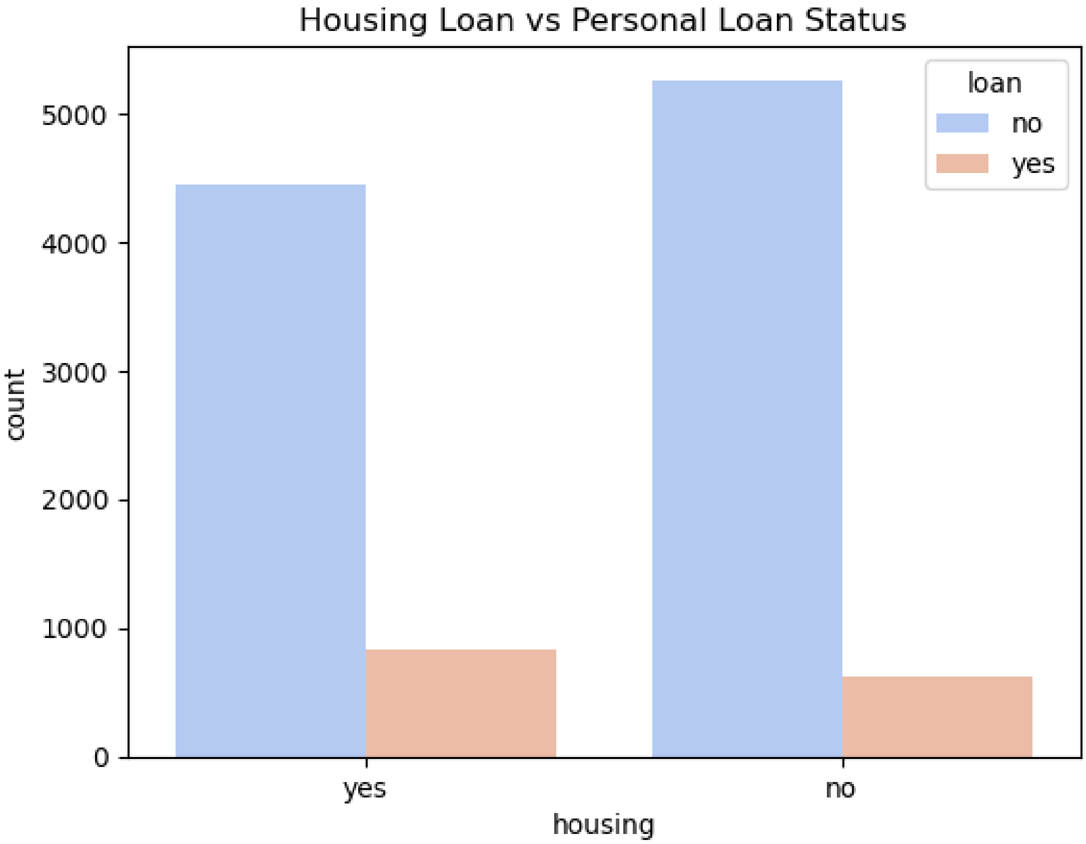
```
plt.figure(figsize=(10,5))
plt.grid()
sns.countplot(data = dataset,x='month',hue='loan',palette='viridis')
plt.title("Monthvise loan given")
plt.show()
print("This graph displays how many loans were provided each month. It helps in␣
↪understanding seasonal trends in loan distribution.")
```



This graph displays how many loans were provided each month. It helps in understanding seasonal trends in loan distribution.

```
[34]: sns.countplot(x='housing', hue='loan', data=dataset, palette='coolwarm')
plt.title('Housing Loan vs Personal Loan Status')
print("Check the relationship between customers having housing loans and␣
 ↪personal loans.")
```

Check the relationship between customers having housing loans and personal loans.


Housing Loan vs Personal Loan Status
count
5000
4000
3000
2000
1000
0
yes
no
housing
loan
no
yes

[ ]: