

In [3]: `pip install pandas`

Requirement already satisfied: pandas in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (2.2.3)
Requirement already satisfied: numpy>=1.26.0 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2.2.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.

In [4]: `pip install mysql-connector-python`

Requirement already satisfied: mysql-connector-python in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (9.2.0)
Note: you may need to restart the kernel to use updated packages.

In [5]: `pip install matplotlib`

Requirement already satisfied: matplotlib in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (3.10.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (2.2.4)
Requirement already satisfied: packaging>=20.0 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
Note: you may need to restart the kernel to use updated packages.

In [6]: `pip install seaborn`

Requirement already satisfied: seaborn in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (0.13.2)

Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from seaborn) (2.2.4)

Requirement already satisfied: pandas>=1.2 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from seaborn) (2.2.3)

Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from seaborn) (3.10.1)

Requirement already satisfied: contourpy>=1.0.1 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.1)

Requirement already satisfied: cycler>=0.10 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)

Requirement already satisfied: fonttools>=4.22.0 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.56.0)

Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)

Requirement already satisfied: packaging>=20.0 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.2)

Requirement already satisfied: pillow>=8 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (11.1.0)

Requirement already satisfied: pyparsing>=2.3.1 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.2.3)

Requirement already satisfied: python-dateutil>=2.7 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from pandas>=1.2->seaborn) (2025.2)

Requirement already satisfied: tzdata>=2022.7 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from pandas>=1.2->seaborn) (2025.2)

Requirement already satisfied: six>=1.5 in c:\users\neha\appdata\local\programs\python\python313\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)

Note: you may need to restart the kernel to use updated packages.

```
In [37]: import mysql.connector
import pandas as pd

#connect to server
cnx = mysql.connector.connect(
    host="127.0.0.1",
    port=3306,
    user="root",
    password="Bittu_N@098")
```

```
In [38]: query = """SELECT * FROM banking_case.banking"""
```

```
In [39]: df = pd.read_sql(query, cnx)
```

```
C:\Users\Neha\AppData\Local\Temp\ipykernel_12920\1600954950.py:1: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
df = pd.read_sql(query, cnx)
```

```
In [40]: cnx.close()
```

```
In [41]: df.head(5)
```

Out[41]:

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	Nationality	Occupation	Struc
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	American	Safety Technician IV	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	African	Software Consultant	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	European	Help Desk Operator	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	American	Geologist II	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	American	Assistant Professor	

5 rows × 25 columns

```
In [42]: df.describe()
```

Out[42]:

	Age	Location ID	Estimated Income	Superannuation Savings	Amount of Credit Cards	Credit Ba
count	3000.000000	3000.000000	3000.000000	3000.000000	3000.000000	3000.00
mean	51.039667	21563.323000	171305.034263	25531.599673	1.463667	3176.20
std	19.854760	12462.273017	111935.808209	16259.950770	0.676387	2497.09
min	17.000000	12.000000	15919.480000	1482.030000	1.000000	1.17
25%	34.000000	10803.500000	82906.595000	12513.775000	1.000000	1236.60
50%	51.000000	21129.500000	142313.480000	22357.355000	1.000000	2560.80
75%	69.000000	32054.500000	242290.305000	35464.740000	2.000000	4522.60
max	85.000000	43369.000000	522330.260000	75963.900000	3.000000	13991.90

```
In [43]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 25 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Client ID                            3000 non-null   object
1   Name                                  3000 non-null   object
2   Age                                   3000 non-null   int64
3   Location ID                          3000 non-null   int64
4   Joined Bank                          3000 non-null   object
5   Banking Contact                      3000 non-null   object
6   Nationality                          3000 non-null   object
7   Occupation                           3000 non-null   object
8   Fee Structure                        3000 non-null   object
9   Loyalty Classification               3000 non-null   object
10  Estimated Income                     3000 non-null   float64
11  Superannuation Savings               3000 non-null   float64
12  Amount of Credit Cards               3000 non-null   int64
13  Credit Card Balance                  3000 non-null   float64
14  Bank Loans                           3000 non-null   float64
15  Bank Deposits                        3000 non-null   float64
16  Checking Accounts                    3000 non-null   float64
17  Saving Accounts                      3000 non-null   float64
18  Foreign Currency Account             3000 non-null   float64
19  Business Lending                     3000 non-null   float64
20  Properties Owned                     3000 non-null   int64
21  Risk Weighting                       3000 non-null   int64
22  BRId                                 3000 non-null   int64
23  GenderId                             3000 non-null   int64
24  IAIId                                3000 non-null   int64
dtypes: float64(9), int64(8), object(8)
memory usage: 586.1+ KB

```

```

In [44]: import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt

```

```

In [45]: df = pd.read_csv('C:/Users/Neha/Downloads/Banking.csv')
df.head(5)

```

Out[45]:

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	Nationality	Occupation	Stru
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	American	Safety Technician IV	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	African	Software Consultant	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	European	Help Desk Operator	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	American	Geologist II	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	American	Assistant Professor	

5 rows × 25 columns



In [46]: `df.shape`

Out[46]: `(3000, 25)`

In [47]: `df.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 25 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Client ID                            3000 non-null   object
1   Name                                3000 non-null   object
2   Age                                  3000 non-null   int64
3   Location ID                          3000 non-null   int64
4   Joined Bank                          3000 non-null   object
5   Banking Contact                      3000 non-null   object
6   Nationality                          3000 non-null   object
7   Occupation                           3000 non-null   object
8   Fee Structure                        3000 non-null   object
9   Loyalty Classification               3000 non-null   object
10  Estimated Income                     3000 non-null   float64
11  Superannuation Savings               3000 non-null   float64
12  Amount of Credit Cards               3000 non-null   int64
13  Credit Card Balance                  3000 non-null   float64
14  Bank Loans                           3000 non-null   float64
15  Bank Deposits                       3000 non-null   float64
16  Checking Accounts                   3000 non-null   float64
17  Saving Accounts                     3000 non-null   float64
18  Foreign Currency Account             3000 non-null   float64
19  Business Lending                    3000 non-null   float64
20  Properties Owned                    3000 non-null   int64
21  Risk Weighting                       3000 non-null   int64
22  BRId                                3000 non-null   int64
23  GenderId                            3000 non-null   int64
24  IAIId                               3000 non-null   int64
dtypes: float64(9), int64(8), object(8)
memory usage: 586.1+ KB

```

Generating descriptive statistics for the dataframes

In [48]: `df.describe()`

Out[48]:

	Age	Location ID	Estimated Income	Superannuation Savings	Amount of Credit Cards	Credit Ba
count	3000.000000	3000.000000	3000.000000	3000.000000	3000.000000	3000.00
mean	51.039667	21563.323000	171305.034263	25531.599673	1.463667	3176.20
std	19.854760	12462.273017	111935.808209	16259.950770	0.676387	2497.09
min	17.000000	12.000000	15919.480000	1482.030000	1.000000	1.17
25%	34.000000	10803.500000	82906.595000	12513.775000	1.000000	1236.60
50%	51.000000	21129.500000	142313.480000	22357.355000	1.000000	2560.80
75%	69.000000	32054.500000	242290.305000	35464.740000	2.000000	4522.60
max	85.000000	43369.000000	522330.260000	75963.900000	3.000000	13991.90

```
In [49]: df['Estimated Income'].min()
```

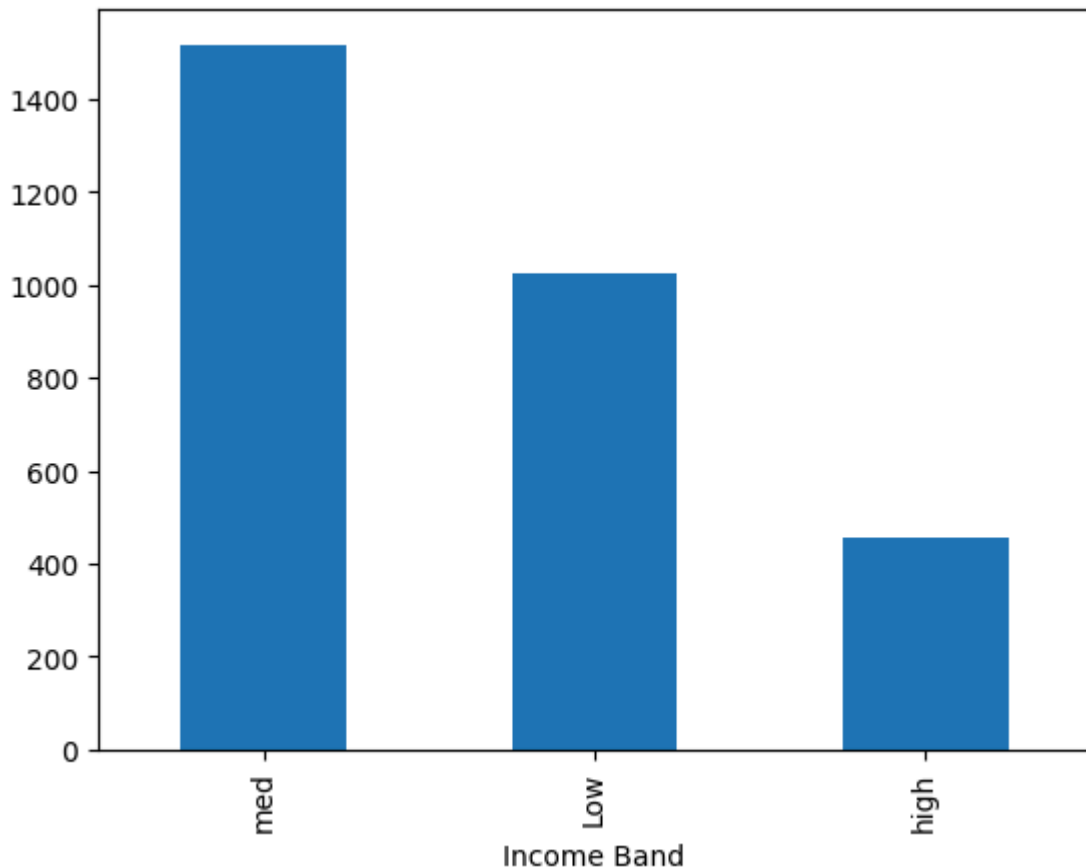
```
Out[49]: np.float64(15919.48)
```

```
In [50]: bins = [0, 100000, 300000, float('inf')]
labels = ['Low', 'med', 'high']

df['Income Band'] = pd.cut(df['Estimated Income'], bins=bins, labels=labels, right=False)
```

```
In [65]: df['Income Band'].value_counts().plot(kind='bar')
```

```
Out[65]: <Axes: xlabel='Income Band'>
```



```
In [66]: print(df.columns)
```

```
Index(['Client ID', 'Name', 'Age', 'Location ID', 'Joined Bank',
      'Banking Contact', 'Nationality', 'Occupation', 'Fee Structure',
      'Loyalty Classification', 'Estimated Income', 'Superannuation Savings',
      'Amount of Credit Cards', 'Credit Card Balance', 'Bank Loans',
      'Bank Deposits', 'Checking Accounts', 'Saving Accounts',
      'Foreign Currency Account', 'Business Lending', 'Properties Owned',
      'Risk Weighting', 'BRId', 'GenderId', 'IAId', 'Income Band'],
      dtype='object')
```

```
In [70]: categorical_cols = df[["BRId", "GenderId", "IAId", "Amount of Credit Cards", "Na

for col in categorical_cols:
    print(f"Value Counts for '{col}':")
    display(df[col].value_counts())
    df[col].value_counts().plot(kind='bar', title=f'Distribution of {col}')
    print(df.head())
```

Value Counts for 'BRId':

BRId

3 1352
1 660
2 495
4 493

Name: count, dtype: int64

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	\
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	

	Nationality	Occupation	Fee Structure	Loyalty Classification	...	\
0	American	Safety Technician IV	High	Jade	...	
1	African	Software Consultant	High	Jade	...	
2	European	Help Desk Operator	High	Gold	...	
3	American	Geologist II	Mid	Silver	...	
4	American	Assistant Professor	Mid	Platinum	...	

	Checking Accounts	Saving Accounts	Foreign Currency Account	\
0	603617.88	607332.46	12249.96	
1	229521.37	344635.16	61162.31	
2	652674.69	203054.35	79071.78	
3	1048157.49	234685.02	57513.65	
4	446644.25	128351.45	30012.14	

	Business Lending	Properties Owned	Risk Weighting	BRId	GenderId	IAId	\
0	1134475.30	1	2	1	1	1	
1	2000526.10	1	3	2	1	2	
2	548137.58	1	3	3	2	3	
3	1148402.29	0	4	4	1	4	
4	1674412.12	0	3	1	2	5	

	Income Band
0	Low
1	med
2	med
3	high
4	med

[5 rows x 26 columns]

Value Counts for 'GenderId':

GenderId

2 1512
1 1488

Name: count, dtype: int64

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	\
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	

	Nationality	Occupation	Fee Structure	Loyalty Classification	...	\
0	American	Safety Technician IV	High	Jade	...	
1	African	Software Consultant	High	Jade	...	
2	European	Help Desk Operator	High	Gold	...	
3	American	Geologist II	Mid	Silver	...	
4	American	Assistant Professor	Mid	Platinum	...	

	Checking Accounts	Saving Accounts	Foreign Currency Account	\
0	603617.88	607332.46	12249.96	
1	229521.37	344635.16	61162.31	
2	652674.69	203054.35	79071.78	
3	1048157.49	234685.02	57513.65	
4	446644.25	128351.45	30012.14	

	Business Lending	Properties Owned	Risk Weighting	BRId	GenderId	IAId	\
0	1134475.30	1	2	1	1	1	
1	2000526.10	1	3	2	1	2	
2	548137.58	1	3	3	2	3	
3	1148402.29	0	4	4	1	4	
4	1674412.12	0	3	1	2	5	

	Income Band
0	Low
1	med
2	med
3	high
4	med

[5 rows x 26 columns]
Value Counts for 'IAId':

IAId

1	177
2	177
3	177
4	177
8	177
9	176
13	176
12	176
10	176
11	176
14	176
15	176
6	89
5	89
7	89
16	88
17	88
18	88
19	88
20	88
21	88
22	88

Name: count, dtype: int64

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	\
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	

	Nationality	Occupation	Fee Structure	Loyalty Classification	...	\
0	American	Safety Technician IV	High	Jade	...	
1	African	Software Consultant	High	Jade	...	
2	European	Help Desk Operator	High	Gold	...	
3	American	Geologist II	Mid	Silver	...	
4	American	Assistant Professor	Mid	Platinum	...	

	Checking Accounts	Saving Accounts	Foreign Currency Account	\
0	603617.88	607332.46	12249.96	
1	229521.37	344635.16	61162.31	
2	652674.69	203054.35	79071.78	
3	1048157.49	234685.02	57513.65	
4	446644.25	128351.45	30012.14	

	Business Lending	Properties Owned	Risk Weighting	BRId	GenderId	IAId	\
0	1134475.30	1	2	1	1	1	
1	2000526.10	1	3	2	1	2	
2	548137.58	1	3	3	2	3	
3	1148402.29	0	4	4	1	4	
4	1674412.12	0	3	1	2	5	

	Income Band
0	Low
1	med
2	med
3	high
4	med

[5 rows x 26 columns]

Value Counts for 'Amount of Credit Cards':

Amount of Credit Cards

1	1922
2	765
3	313

Name: count, dtype: int64

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	\
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	

	Nationality	Occupation	Fee Structure	Loyalty Classification	...	\
0	American	Safety Technician IV	High	Jade	...	
1	African	Software Consultant	High	Jade	...	
2	European	Help Desk Operator	High	Gold	...	
3	American	Geologist II	Mid	Silver	...	
4	American	Assistant Professor	Mid	Platinum	...	

	Checking Accounts	Saving Accounts	Foreign Currency Account	\
0	603617.88	607332.46	12249.96	
1	229521.37	344635.16	61162.31	
2	652674.69	203054.35	79071.78	
3	1048157.49	234685.02	57513.65	
4	446644.25	128351.45	30012.14	

	Business Lending	Properties Owned	Risk Weighting	BRId	GenderId	IAId	\
0	1134475.30	1	2	1	1	1	
1	2000526.10	1	3	2	1	2	
2	548137.58	1	3	3	2	3	
3	1148402.29	0	4	4	1	4	
4	1674412.12	0	3	1	2	5	

	Income Band
0	Low
1	med
2	med
3	high
4	med

[5 rows x 26 columns]

Value Counts for 'Nationality':

Nationality

European 1309

Asian 754

American 507

Australian 254

African 176

Name: count, dtype: int64

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	\
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	

	Nationality	Occupation	Fee Structure	Loyalty Classification	...	\
0	American	Safety Technician IV	High	Jade	...	
1	African	Software Consultant	High	Jade	...	
2	European	Help Desk Operator	High	Gold	...	
3	American	Geologist II	Mid	Silver	...	
4	American	Assistant Professor	Mid	Platinum	...	

	Checking Accounts	Saving Accounts	Foreign Currency Account	\
0	603617.88	607332.46	12249.96	
1	229521.37	344635.16	61162.31	
2	652674.69	203054.35	79071.78	
3	1048157.49	234685.02	57513.65	
4	446644.25	128351.45	30012.14	

	Business Lending	Properties Owned	Risk Weighting	BRId	GenderId	IAId	\
0	1134475.30	1	2	1	1	1	
1	2000526.10	1	3	2	1	2	
2	548137.58	1	3	3	2	3	
3	1148402.29	0	4	4	1	4	
4	1674412.12	0	3	1	2	5	

	Income Band
0	Low
1	med
2	med
3	high
4	med

[5 rows x 26 columns]

Value Counts for 'Occupation':

Occupation

Associate Professor	28
Structural Analysis Engineer	28
Recruiter	25
Account Coordinator	24
Human Resources Manager	24

..

Office Assistant IV	8
Automation Specialist I	7
Computer Systems Analyst I	6
Developer III	5
Senior Sales Associate	4

Name: count, Length: 195, dtype: int64

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	\
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	

	Nationality	Occupation	Fee Structure	Loyalty Classification	...	\
0	American	Safety Technician IV	High	Jade	...	
1	African	Software Consultant	High	Jade	...	
2	European	Help Desk Operator	High	Gold	...	
3	American	Geologist II	Mid	Silver	...	
4	American	Assistant Professor	Mid	Platinum	...	

	Checking Accounts	Saving Accounts	Foreign Currency Account	\
0	603617.88	607332.46	12249.96	
1	229521.37	344635.16	61162.31	
2	652674.69	203054.35	79071.78	
3	1048157.49	234685.02	57513.65	
4	446644.25	128351.45	30012.14	

	Business Lending	Properties Owned	Risk Weighting	BRId	GenderId	IAId	\
0	1134475.30	1	2	1	1	1	
1	2000526.10	1	3	2	1	2	
2	548137.58	1	3	3	2	3	
3	1148402.29	0	4	4	1	4	
4	1674412.12	0	3	1	2	5	

	Income Band
0	Low
1	med
2	med
3	high
4	med

[5 rows x 26 columns]

Value Counts for 'Fee Structure':

Fee Structure

High 1476

Mid 962

Low 562

Name: count, dtype: int64

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	\
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	

	Nationality	Occupation	Fee Structure	Loyalty Classification	...	\
0	American	Safety Technician IV	High	Jade	...	
1	African	Software Consultant	High	Jade	...	
2	European	Help Desk Operator	High	Gold	...	
3	American	Geologist II	Mid	Silver	...	
4	American	Assistant Professor	Mid	Platinum	...	

	Checking Accounts	Saving Accounts	Foreign Currency Account	\
0	603617.88	607332.46	12249.96	
1	229521.37	344635.16	61162.31	
2	652674.69	203054.35	79071.78	
3	1048157.49	234685.02	57513.65	
4	446644.25	128351.45	30012.14	

	Business Lending	Properties Owned	Risk Weighting	BRId	GenderId	IAId	\
0	1134475.30	1	2	1	1	1	
1	2000526.10	1	3	2	1	2	
2	548137.58	1	3	3	2	3	
3	1148402.29	0	4	4	1	4	
4	1674412.12	0	3	1	2	5	

	Income Band
0	Low
1	med
2	med
3	high
4	med

[5 rows x 26 columns]

Value Counts for 'Loyalty Classification':

Loyalty Classification

Jade 1331

Silver 767

Gold 585

Platinum 317

Name: count, dtype: int64

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	\
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	

	Nationality	Occupation	Fee Structure	Loyalty Classification	...	\
0	American	Safety Technician IV	High	Jade	...	
1	African	Software Consultant	High	Jade	...	
2	European	Help Desk Operator	High	Gold	...	
3	American	Geologist II	Mid	Silver	...	
4	American	Assistant Professor	Mid	Platinum	...	

	Checking Accounts	Saving Accounts	Foreign Currency Account	\
0	603617.88	607332.46	12249.96	
1	229521.37	344635.16	61162.31	
2	652674.69	203054.35	79071.78	
3	1048157.49	234685.02	57513.65	
4	446644.25	128351.45	30012.14	

	Business Lending	Properties Owned	Risk Weighting	BRId	GenderId	IAId	\
0	1134475.30	1	2	1	1	1	
1	2000526.10	1	3	2	1	2	
2	548137.58	1	3	3	2	3	
3	1148402.29	0	4	4	1	4	
4	1674412.12	0	3	1	2	5	

	Income Band
0	Low
1	med
2	med
3	high
4	med

[5 rows x 26 columns]

Value Counts for 'Properties Owned':

Properties Owned

2 777

1 776

3 742

0 705

Name: count, dtype: int64

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	\
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	

	Nationality	Occupation	Fee Structure	Loyalty Classification	...	\
0	American	Safety Technician IV	High	Jade	...	
1	African	Software Consultant	High	Jade	...	
2	European	Help Desk Operator	High	Gold	...	
3	American	Geologist II	Mid	Silver	...	
4	American	Assistant Professor	Mid	Platinum	...	

	Checking Accounts	Saving Accounts	Foreign Currency Account	\
0	603617.88	607332.46	12249.96	
1	229521.37	344635.16	61162.31	
2	652674.69	203054.35	79071.78	
3	1048157.49	234685.02	57513.65	
4	446644.25	128351.45	30012.14	

	Business Lending	Properties Owned	Risk Weighting	BRId	GenderId	IAId	\
0	1134475.30	1	2	1	1	1	
1	2000526.10	1	3	2	1	2	
2	548137.58	1	3	3	2	3	
3	1148402.29	0	4	4	1	4	
4	1674412.12	0	3	1	2	5	

	Income Band
0	Low
1	med
2	med
3	high
4	med

[5 rows x 26 columns]

Value Counts for 'Risk Weighting':

Risk Weighting

2	1222
1	836
3	460
4	322
5	160

Name: count, dtype: int64

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	\
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	

	Nationality	Occupation	Fee Structure	Loyalty Classification	...	\
0	American	Safety Technician IV	High	Jade	...	
1	African	Software Consultant	High	Jade	...	
2	European	Help Desk Operator	High	Gold	...	
3	American	Geologist II	Mid	Silver	...	
4	American	Assistant Professor	Mid	Platinum	...	

	Checking Accounts	Saving Accounts	Foreign Currency Account	\
0	603617.88	607332.46	12249.96	
1	229521.37	344635.16	61162.31	
2	652674.69	203054.35	79071.78	
3	1048157.49	234685.02	57513.65	
4	446644.25	128351.45	30012.14	

	Business Lending	Properties Owned	Risk Weighting	BRId	GenderId	IAId	\
0	1134475.30	1	2	1	1	1	
1	2000526.10	1	3	2	1	2	
2	548137.58	1	3	3	2	3	
3	1148402.29	0	4	4	1	4	
4	1674412.12	0	3	1	2	5	

	Income Band
0	Low
1	med
2	med
3	high
4	med

[5 rows x 26 columns]

Value Counts for 'Income Band':

Income Band

med 1517

Low 1027

high 456

Name: count, dtype: int64

	Client ID	Name	Age	Location ID	Joined Bank	Banking Contact	\
0	IND81288	Raymond Mills	24	34324	06-05-2019	Anthony Torres	
1	IND65833	Julia Spencer	23	42205	10-12-2001	Jonathan Hawkins	
2	IND47499	Stephen Murray	27	7314	25-01-2010	Anthony Berry	
3	IND72498	Virginia Garza	40	34594	28-03-2019	Steve Diaz	
4	IND60181	Melissa Sanders	46	41269	20-07-2012	Shawn Long	

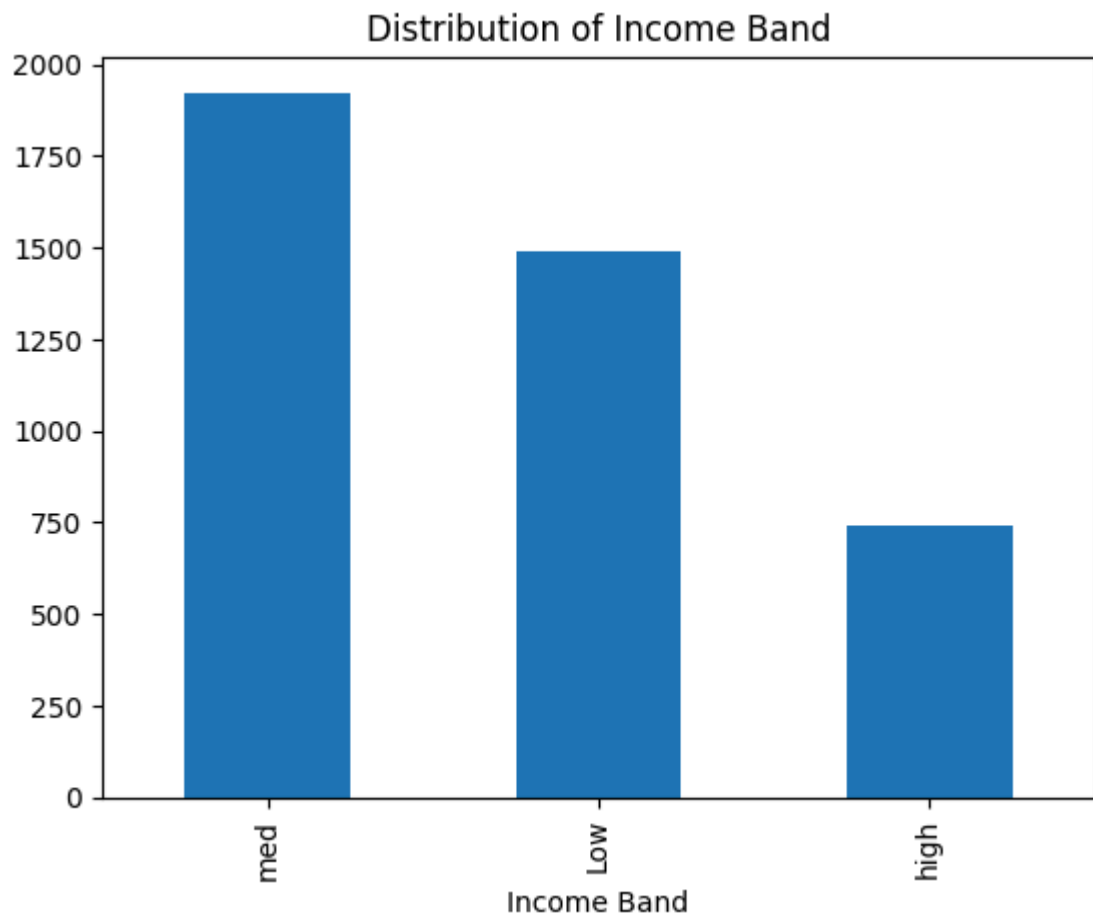
	Nationality	Occupation	Fee Structure	Loyalty Classification	...	\
0	American	Safety Technician IV	High	Jade	...	
1	African	Software Consultant	High	Jade	...	
2	European	Help Desk Operator	High	Gold	...	
3	American	Geologist II	Mid	Silver	...	
4	American	Assistant Professor	Mid	Platinum	...	

	Checking Accounts	Saving Accounts	Foreign Currency Account	\
0	603617.88	607332.46	12249.96	
1	229521.37	344635.16	61162.31	
2	652674.69	203054.35	79071.78	
3	1048157.49	234685.02	57513.65	
4	446644.25	128351.45	30012.14	

	Business Lending	Properties Owned	Risk Weighting	BRId	GenderId	IAId	\
0	1134475.30	1	2	1	1	1	
1	2000526.10	1	3	2	1	2	
2	548137.58	1	3	3	2	3	
3	1148402.29	0	4	4	1	4	
4	1674412.12	0	3	1	2	5	

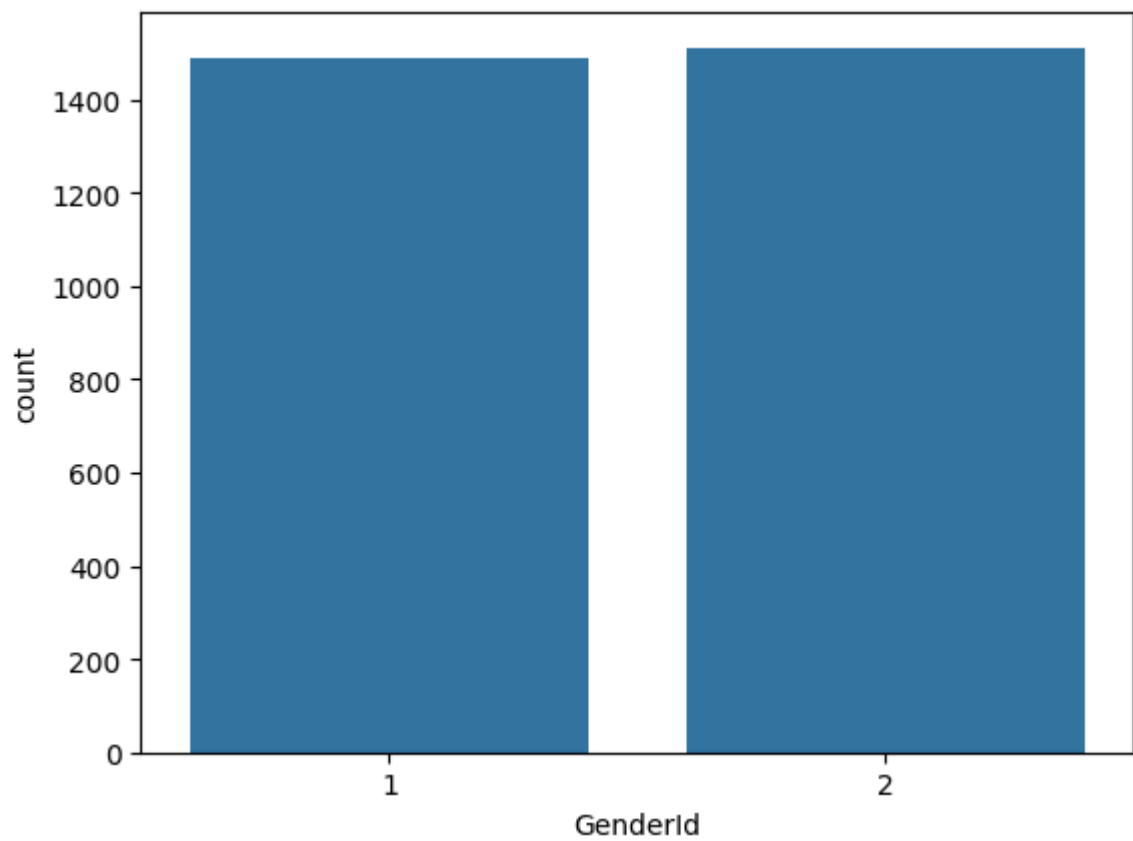
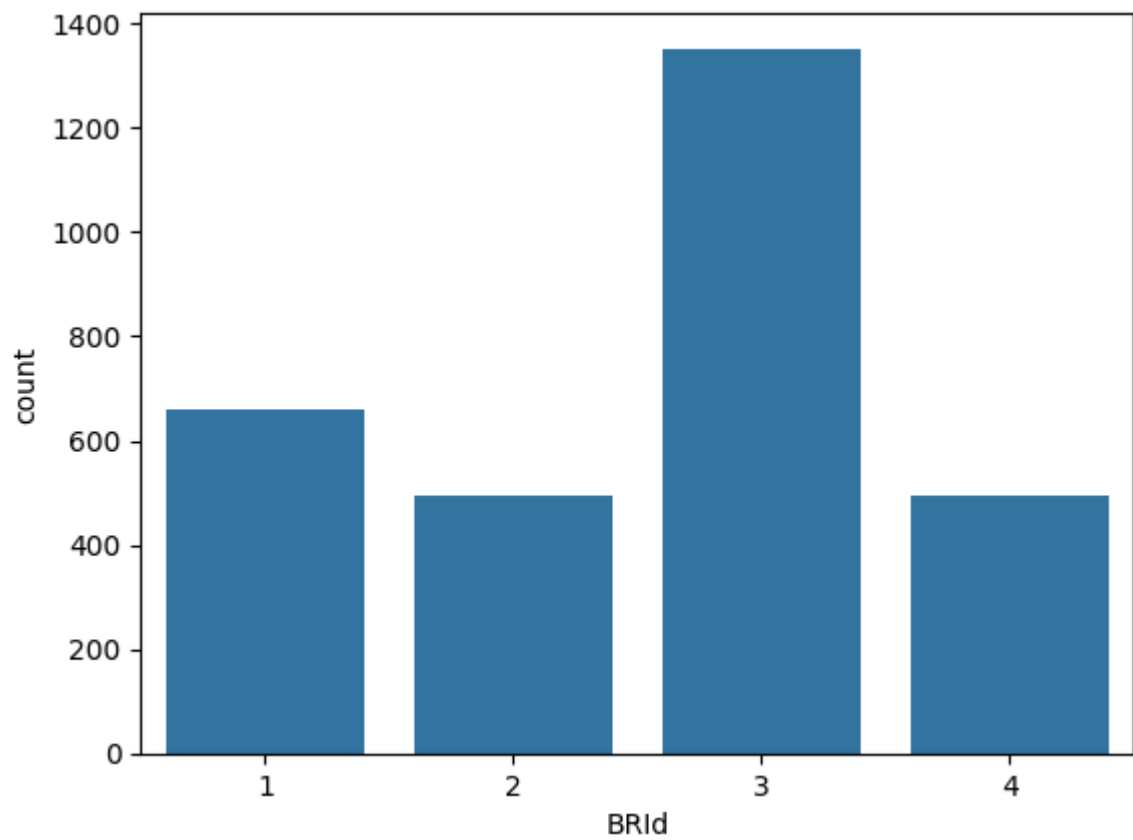
	Income Band
0	Low
1	med
2	med
3	high
4	med

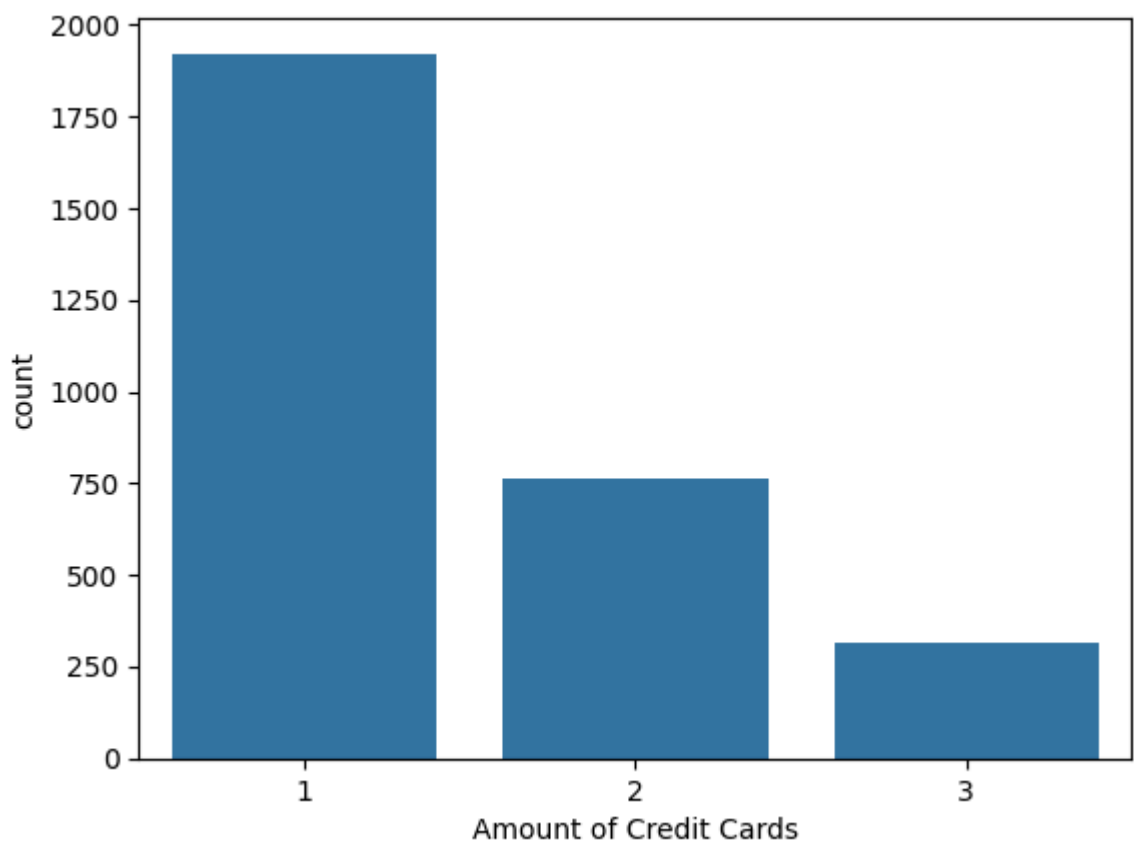
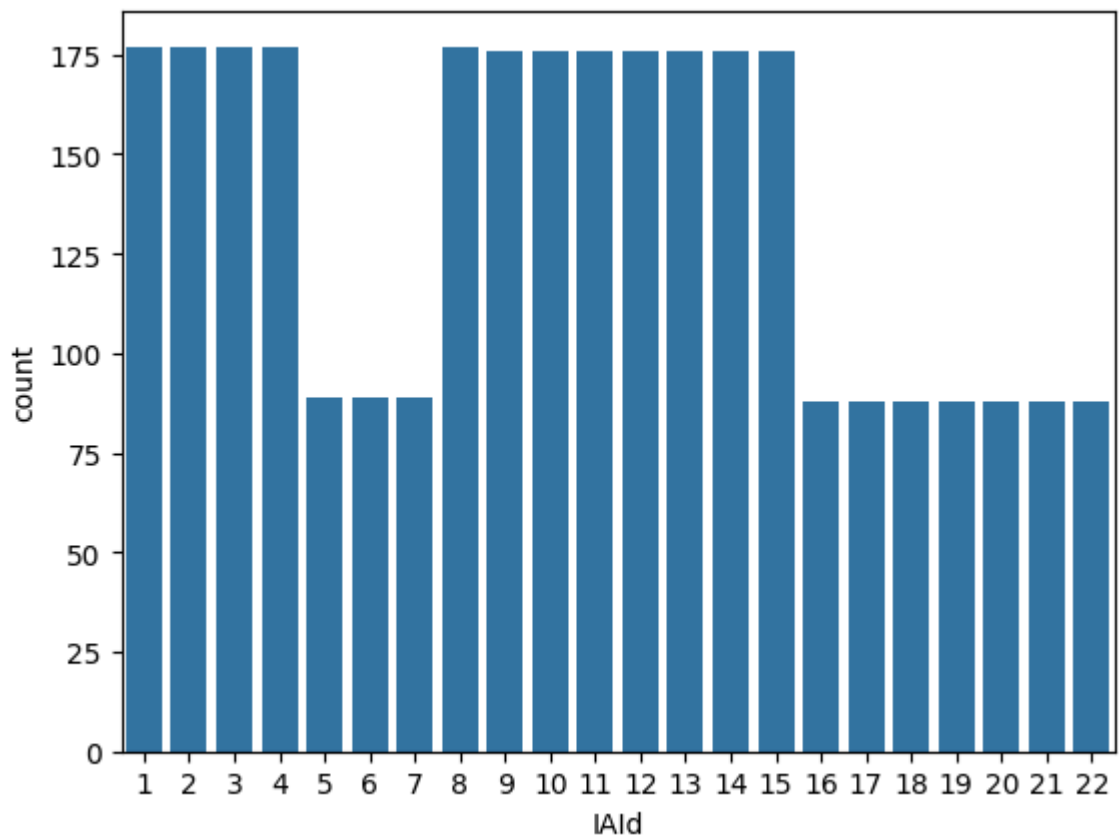
[5 rows x 26 columns]

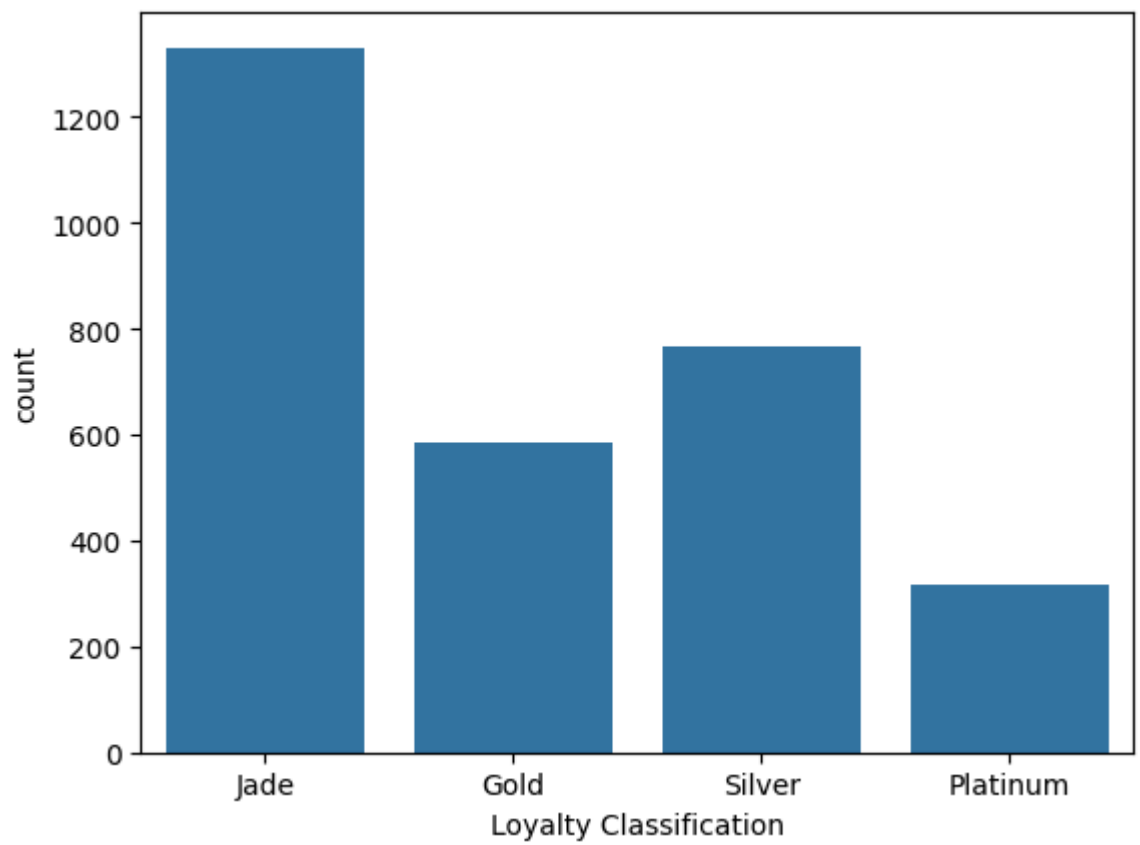
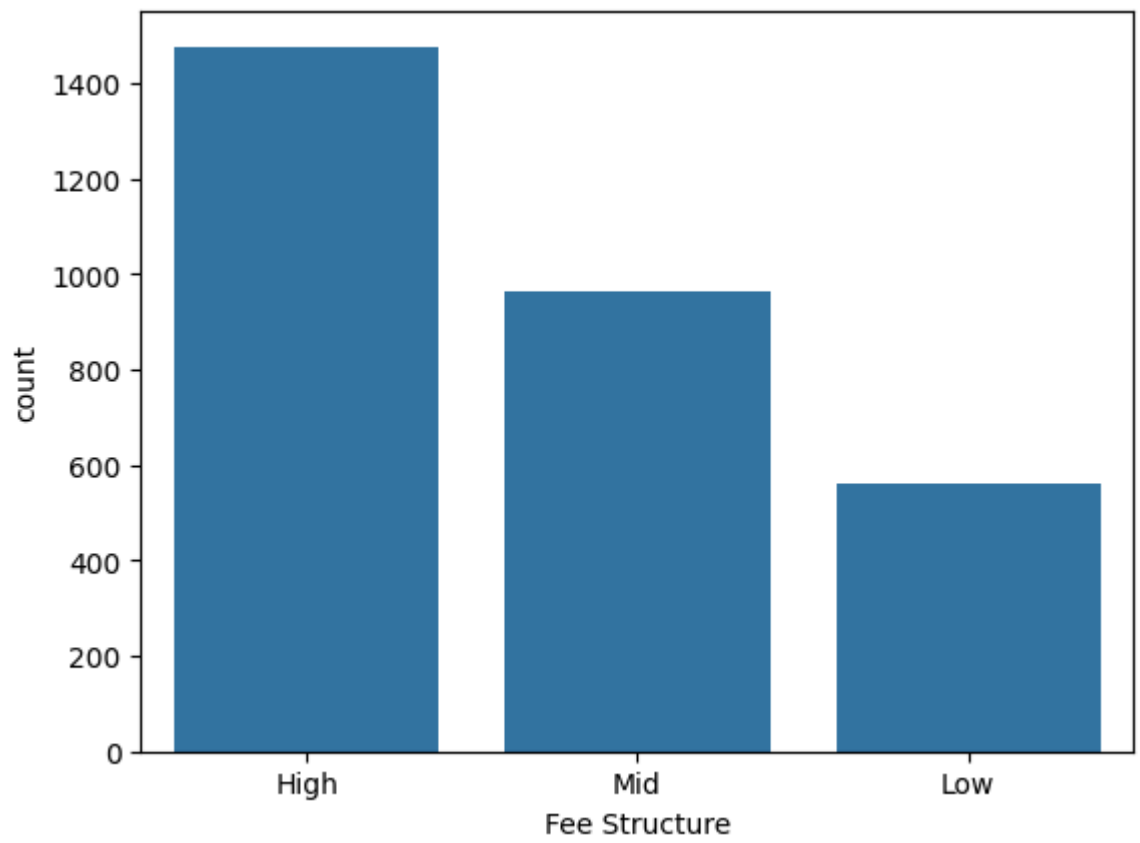


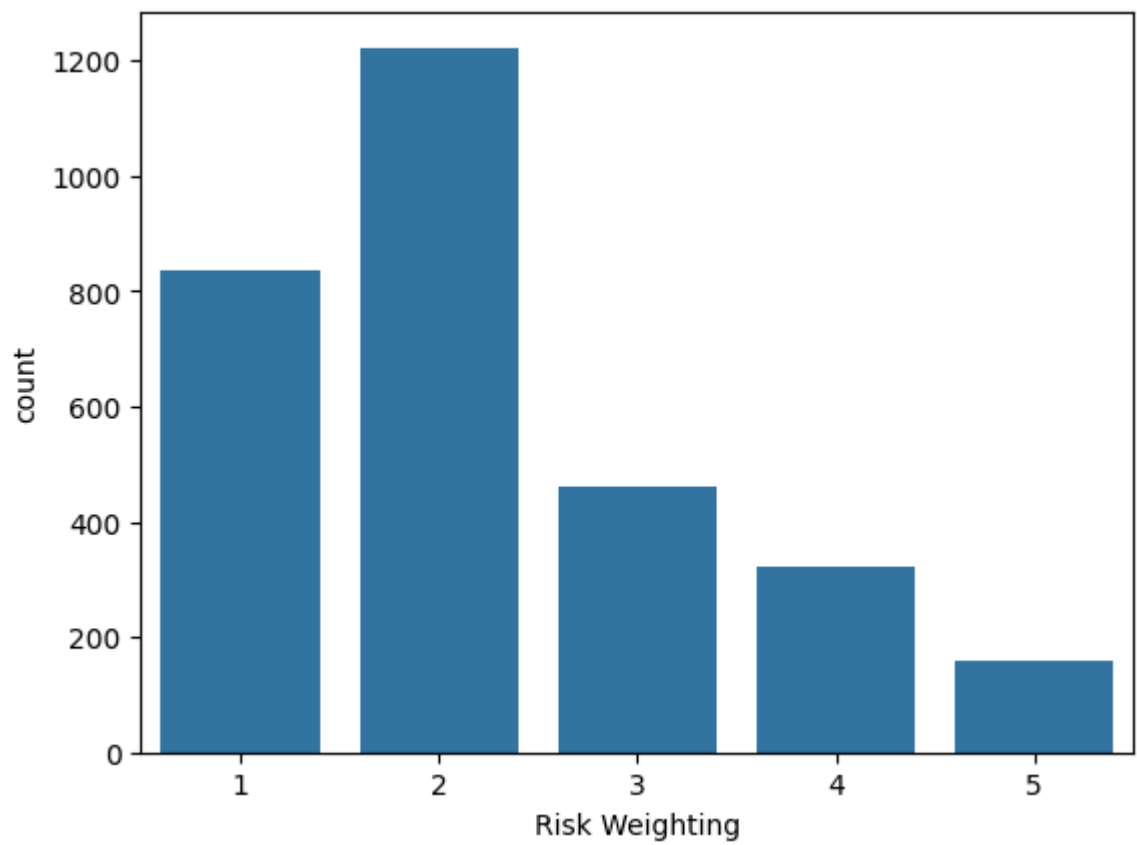
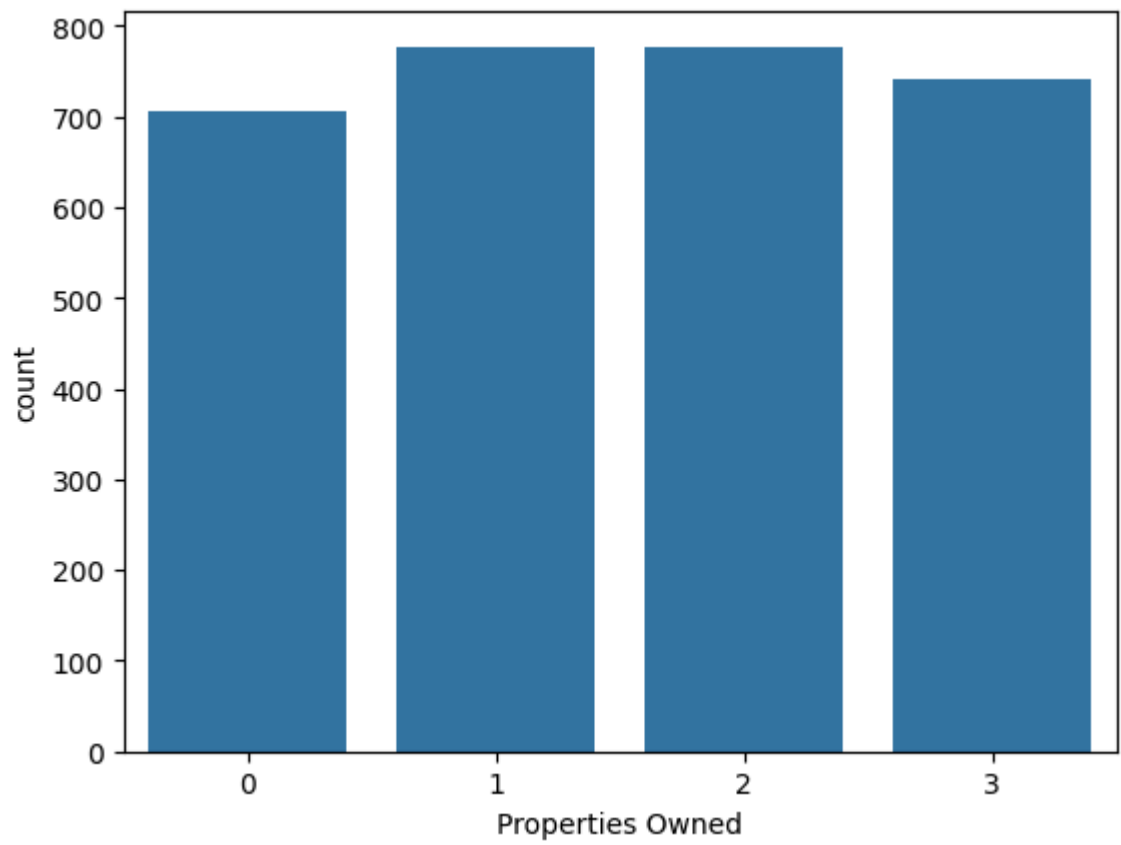
Univariate analysis

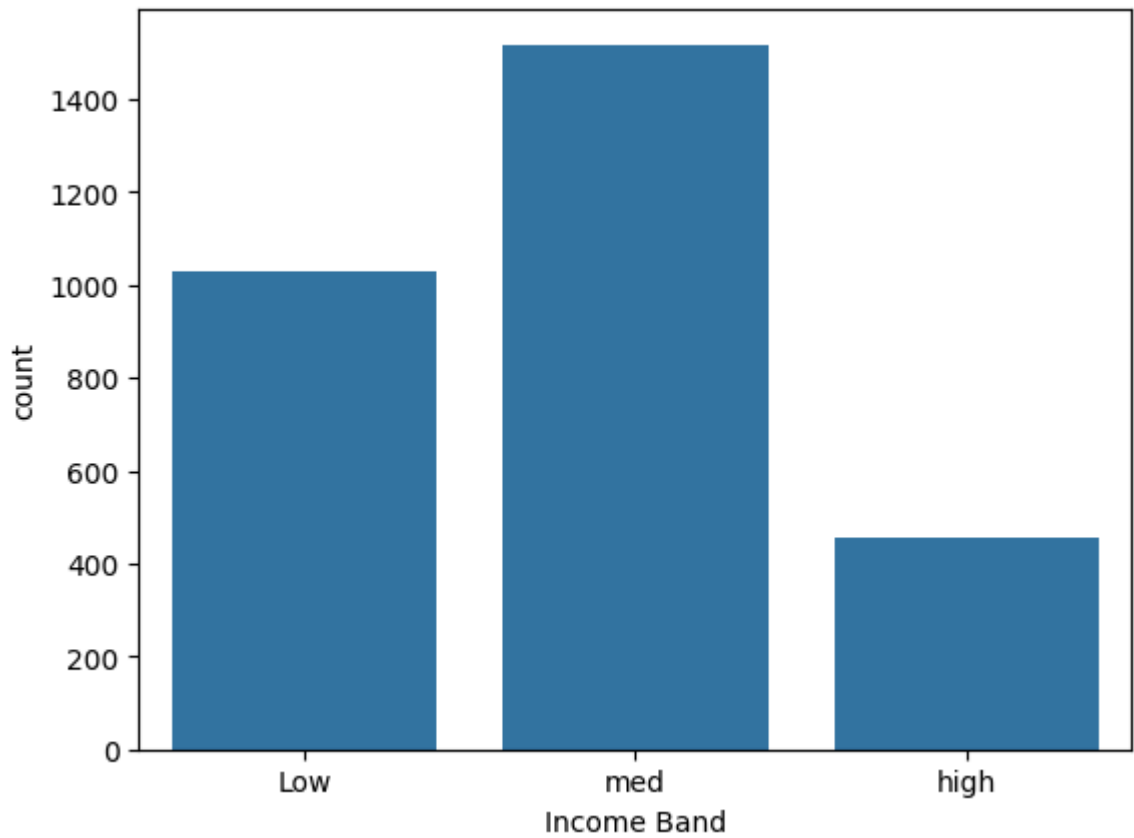
```
In [54]: for i, predictor in enumerate( df[["BRId", "GenderId", "IAId", "Amount of Credit  
plt.figure(i)  
sns.countplot(data=df, x=predictor)  
plt.figure(i)
```





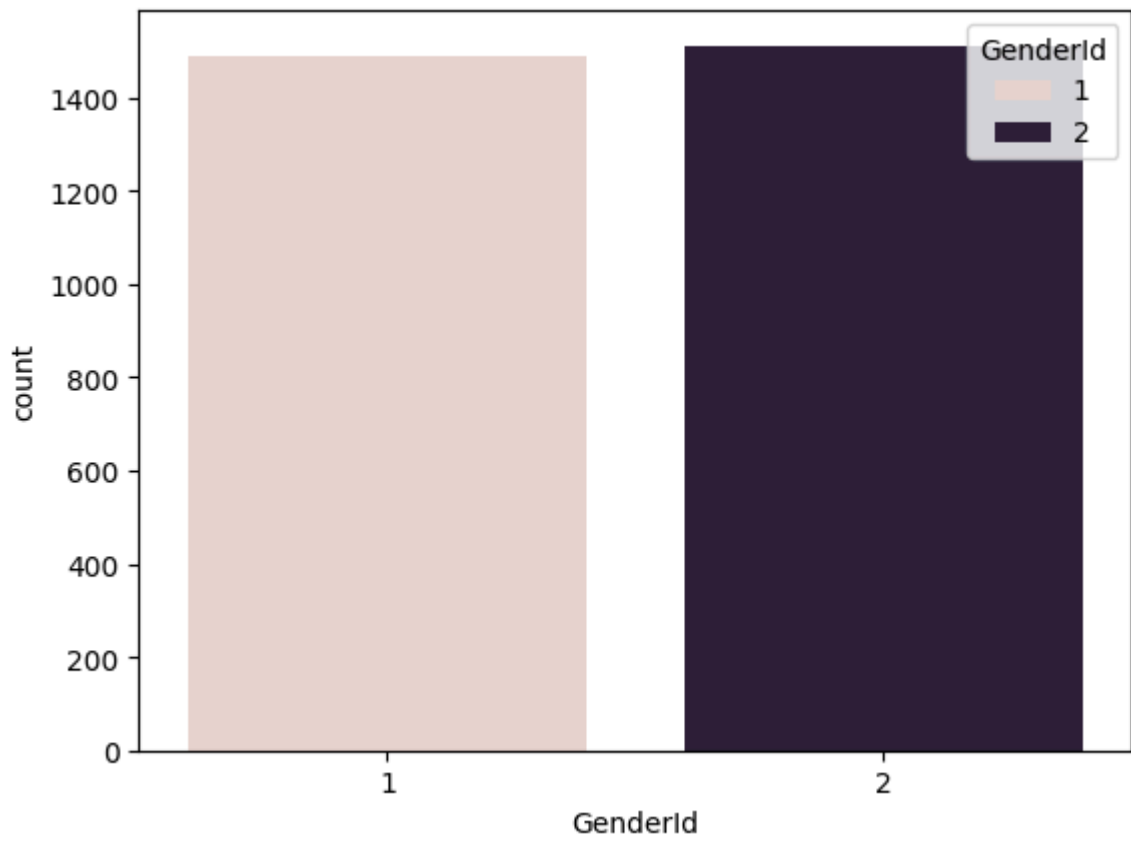
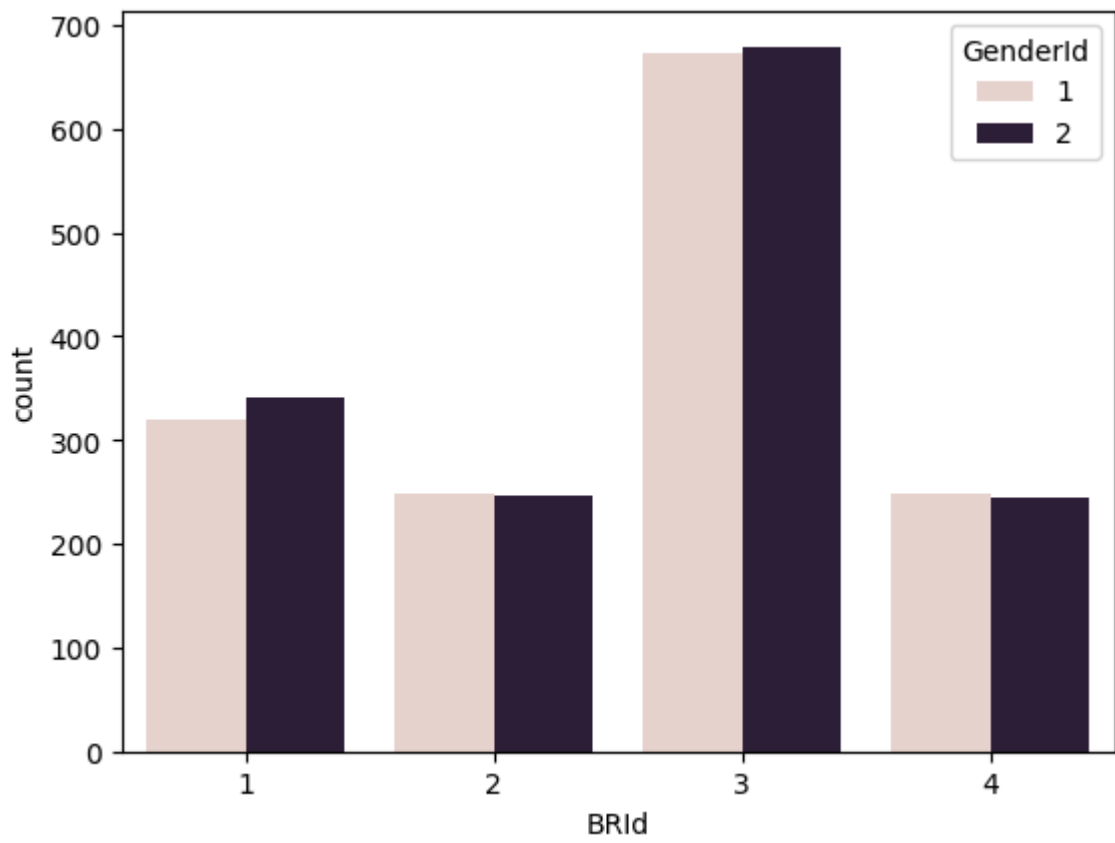


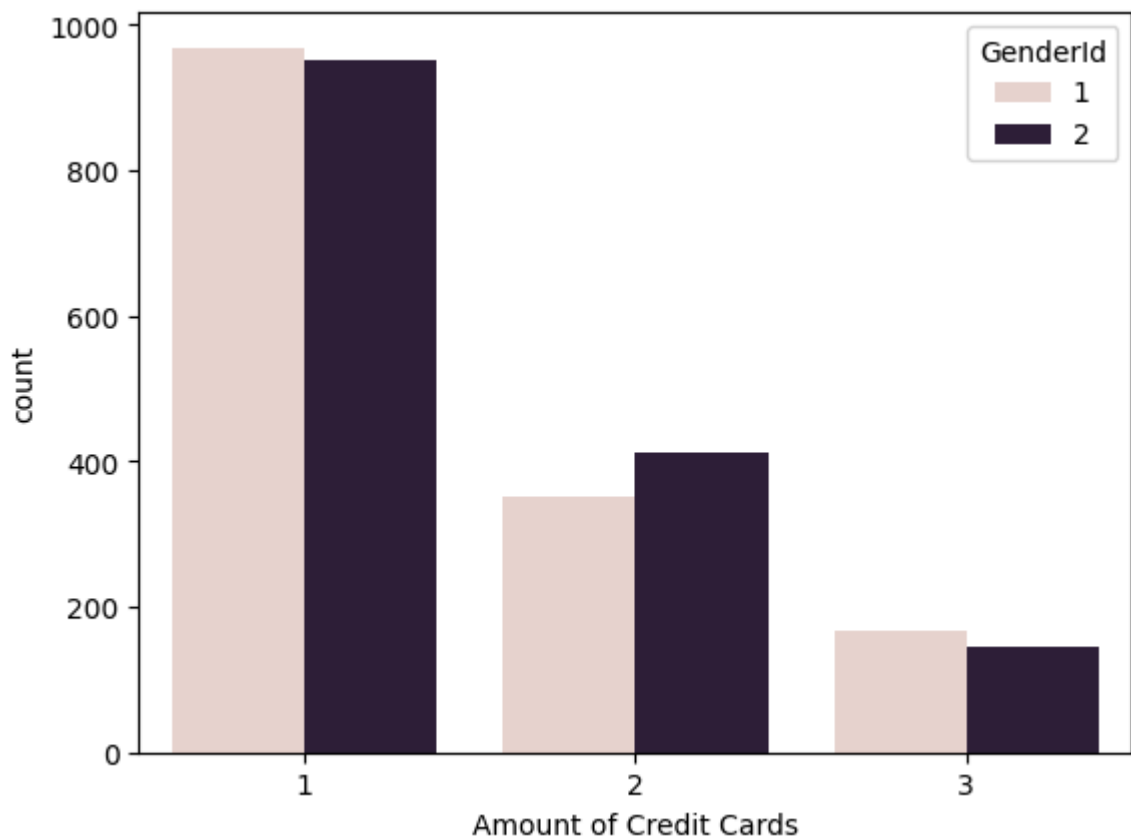
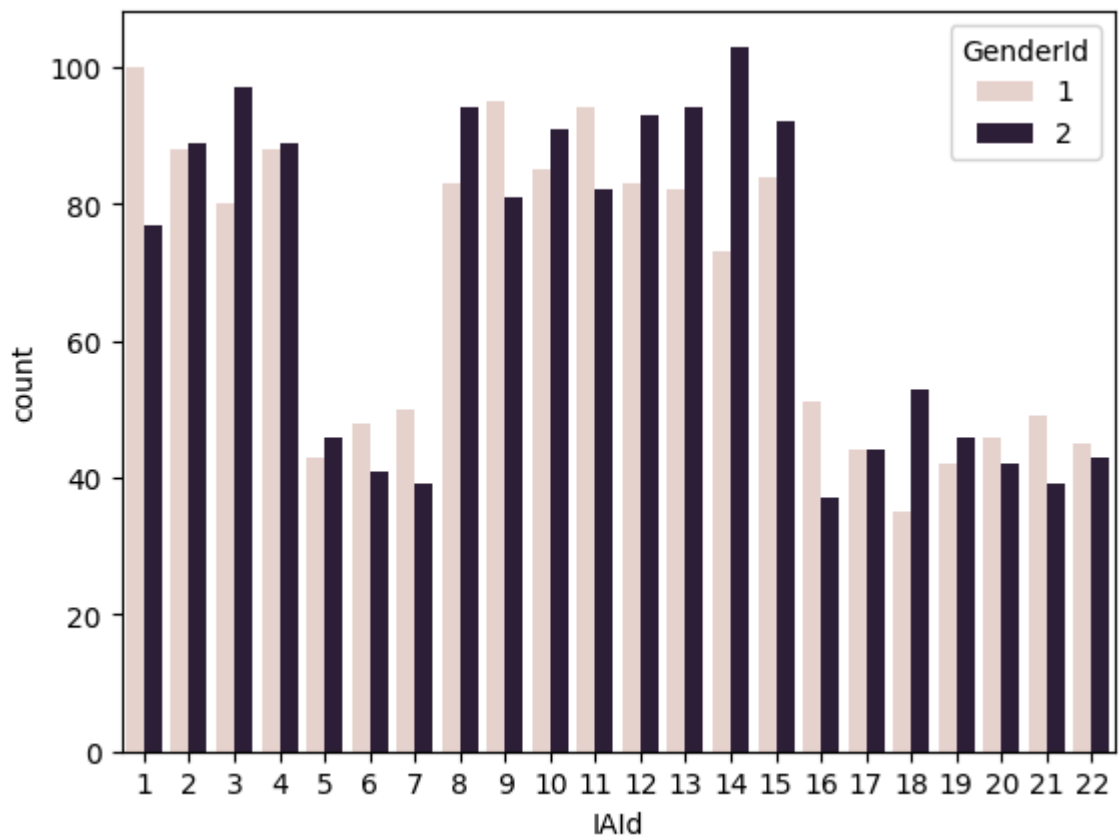


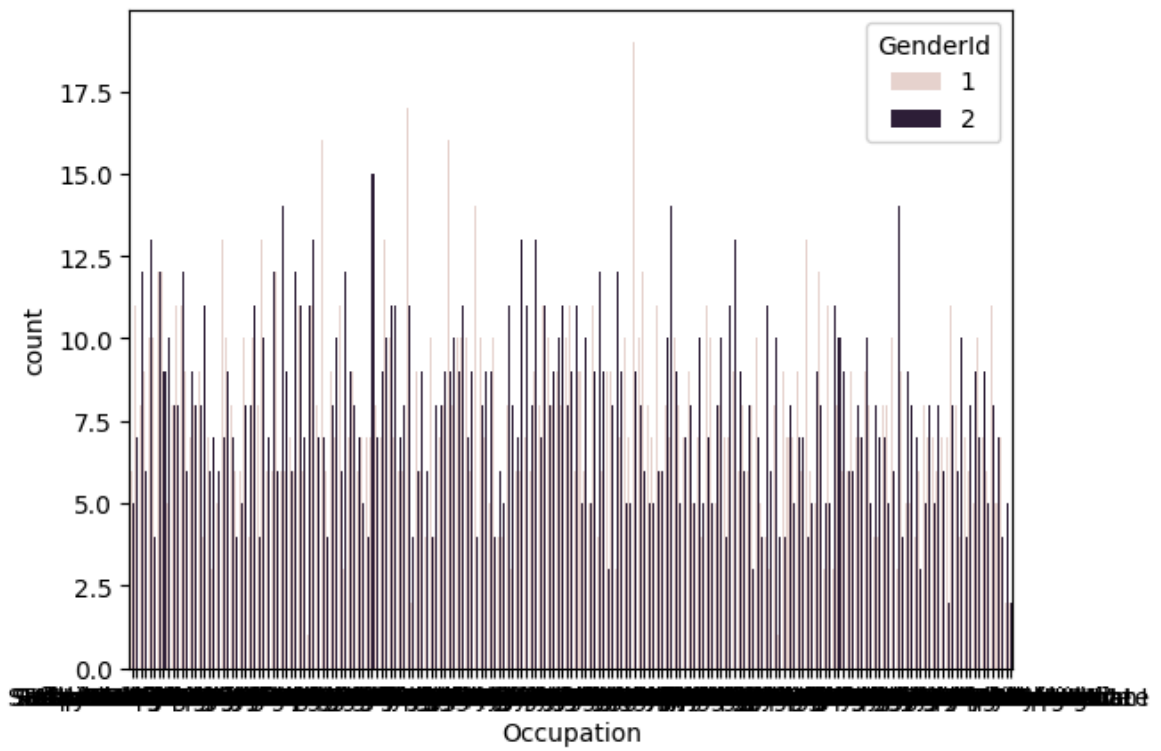
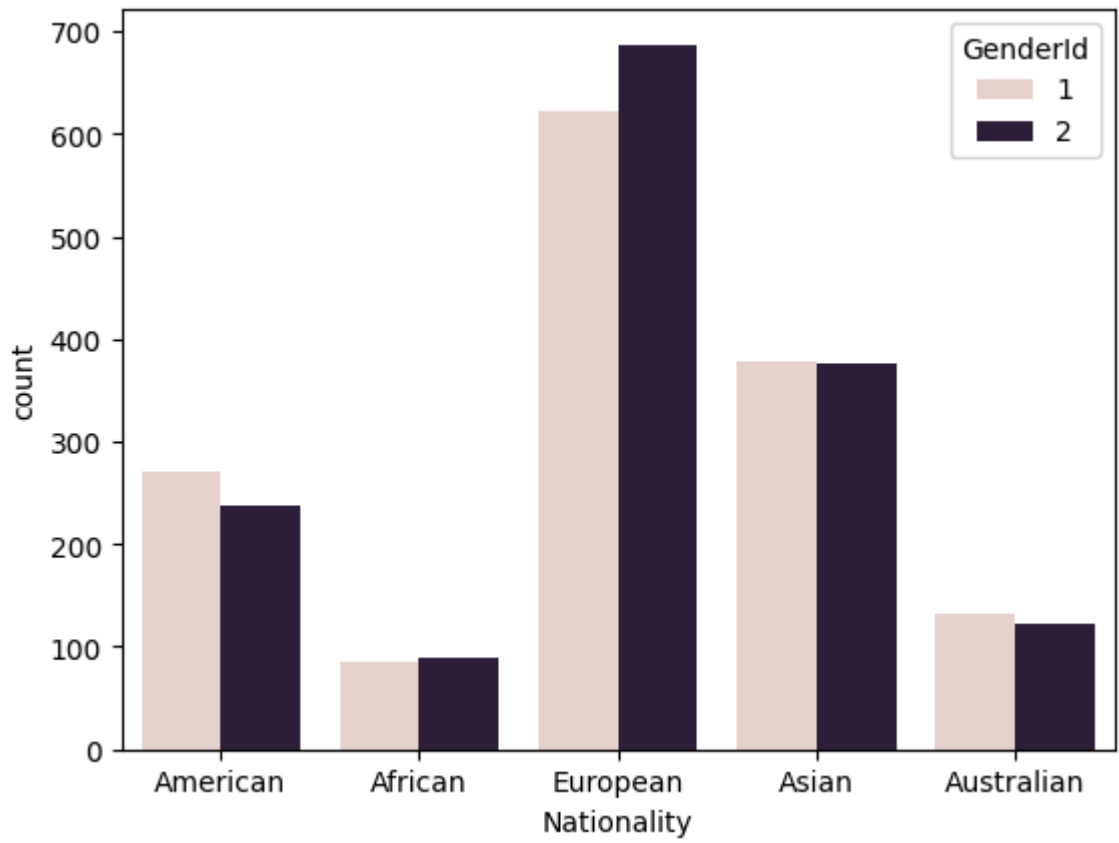


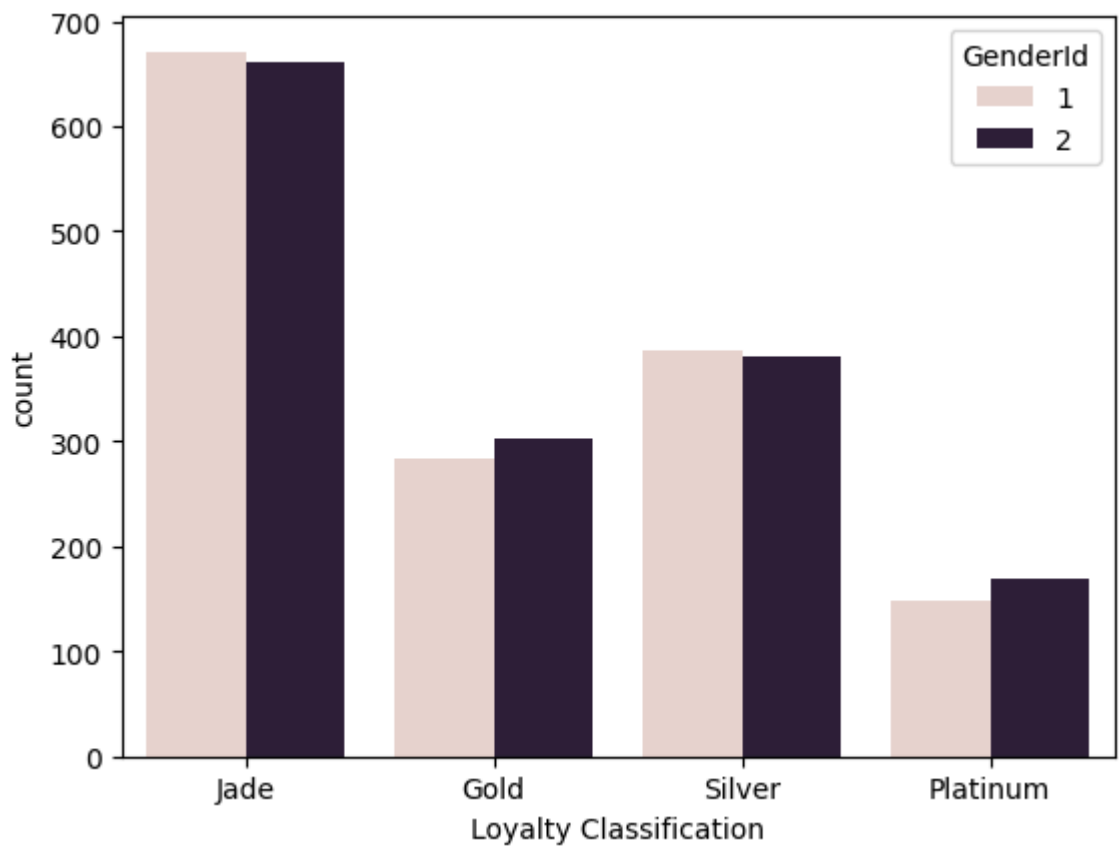
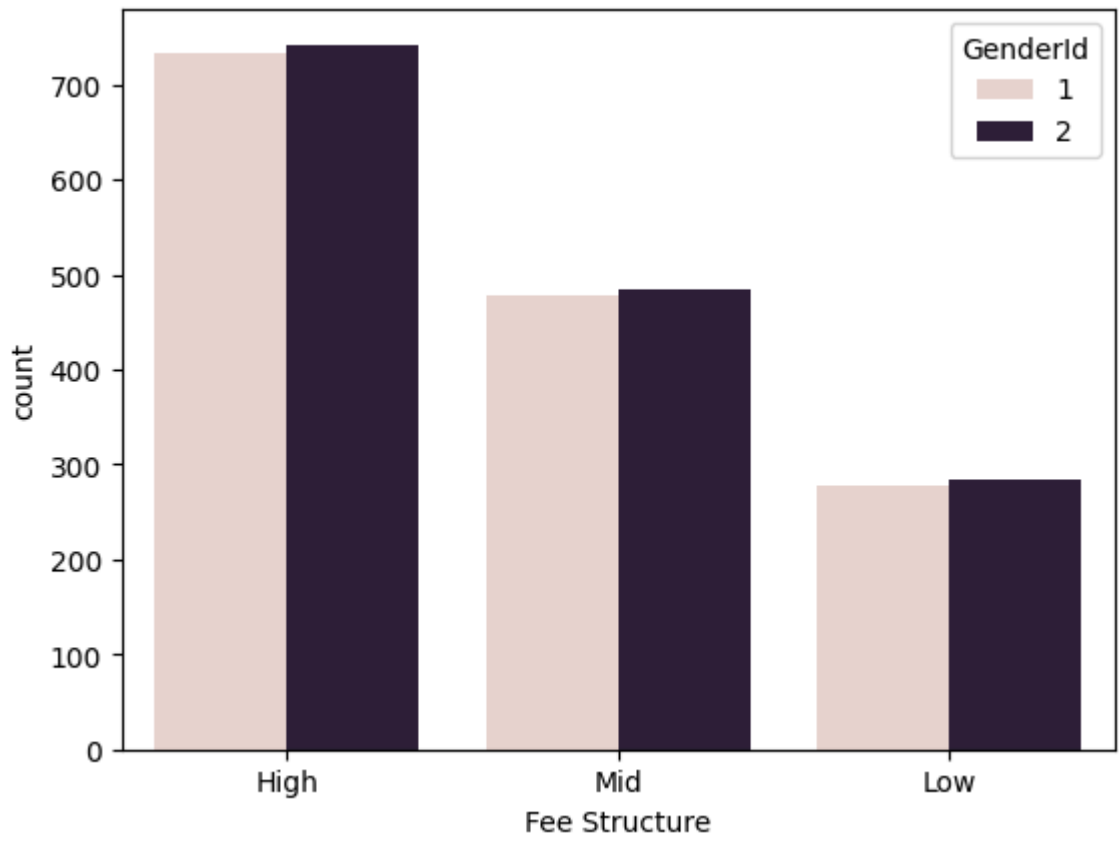
Bivariate analysis

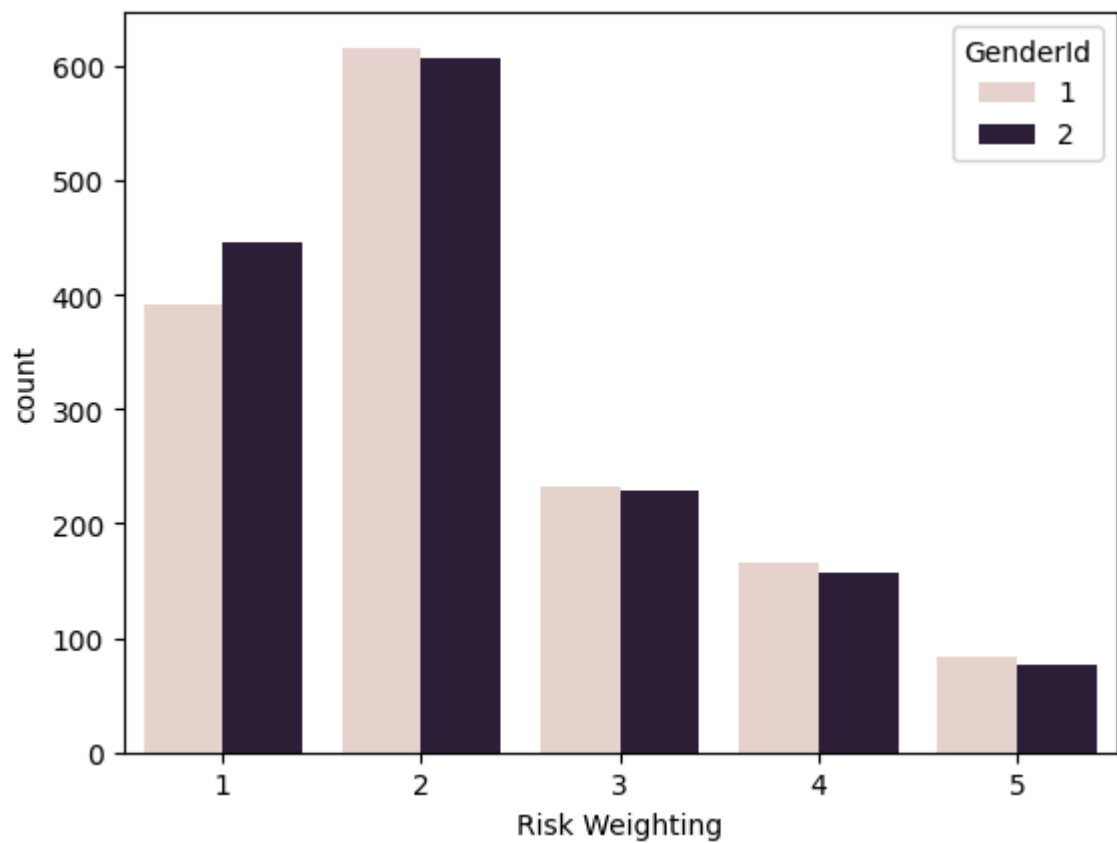
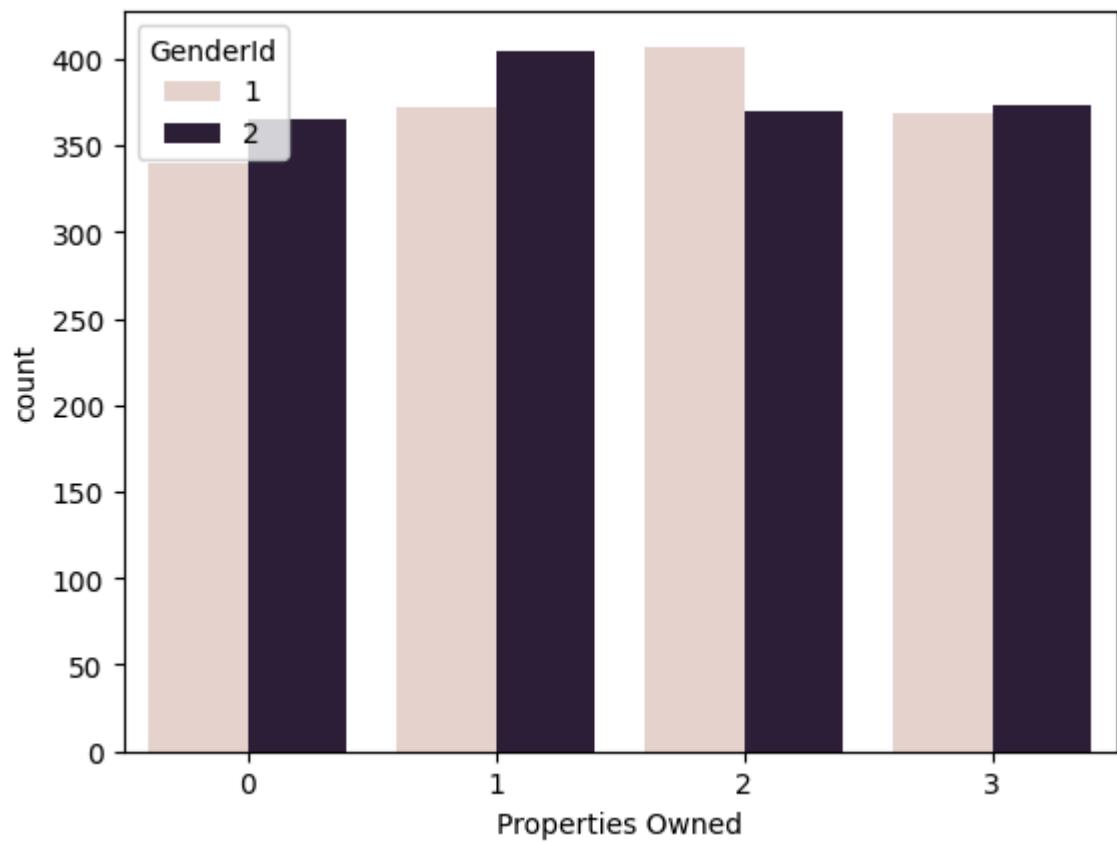
```
In [55]: for i, predictor in enumerate( df[["BRId", "GenderId", "IAId", "Amount of Credit  
plt.figure(i)  
sns.countplot(data=df, x=predictor, hue='GenderId')  
plt.figure(i)
```

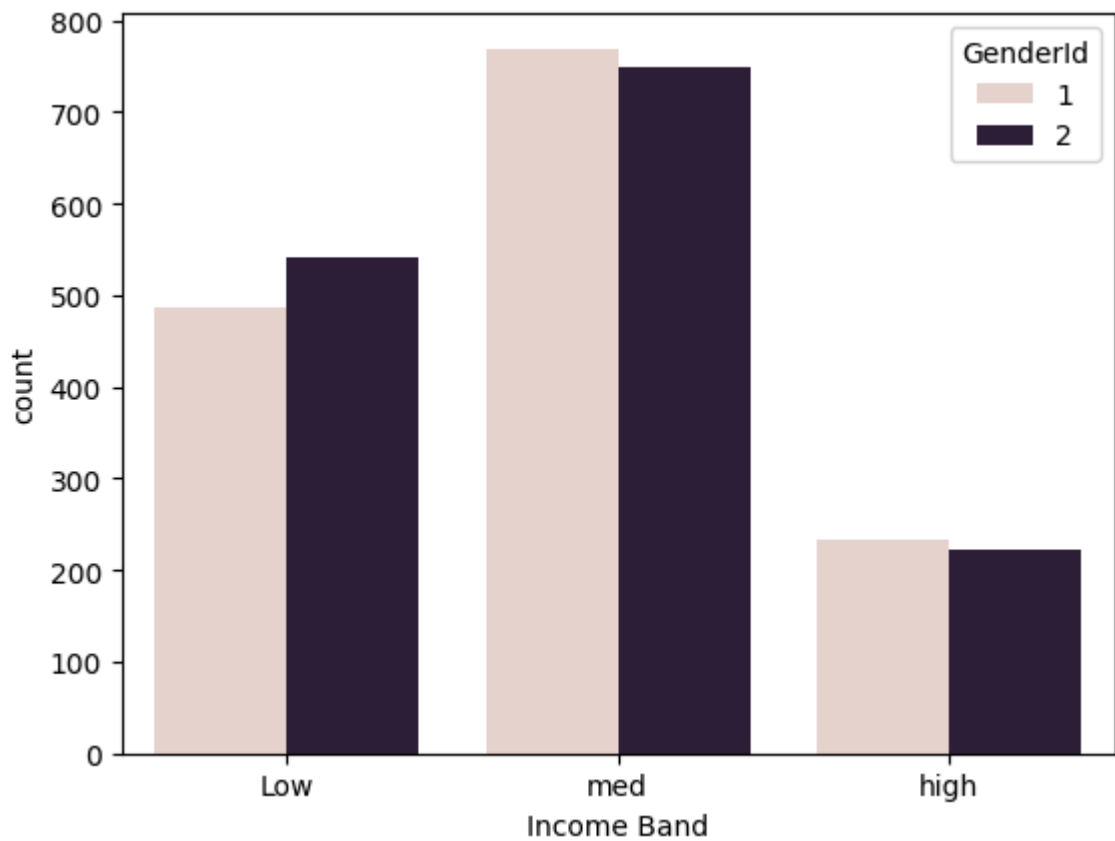




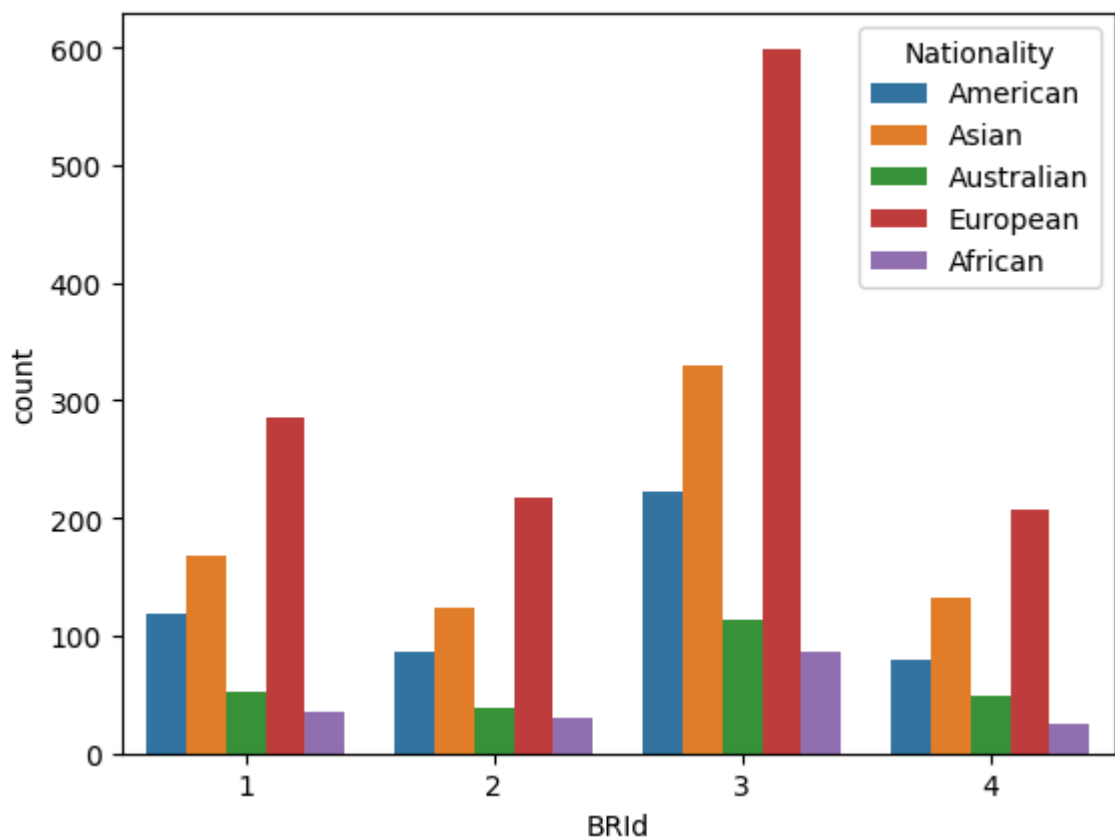


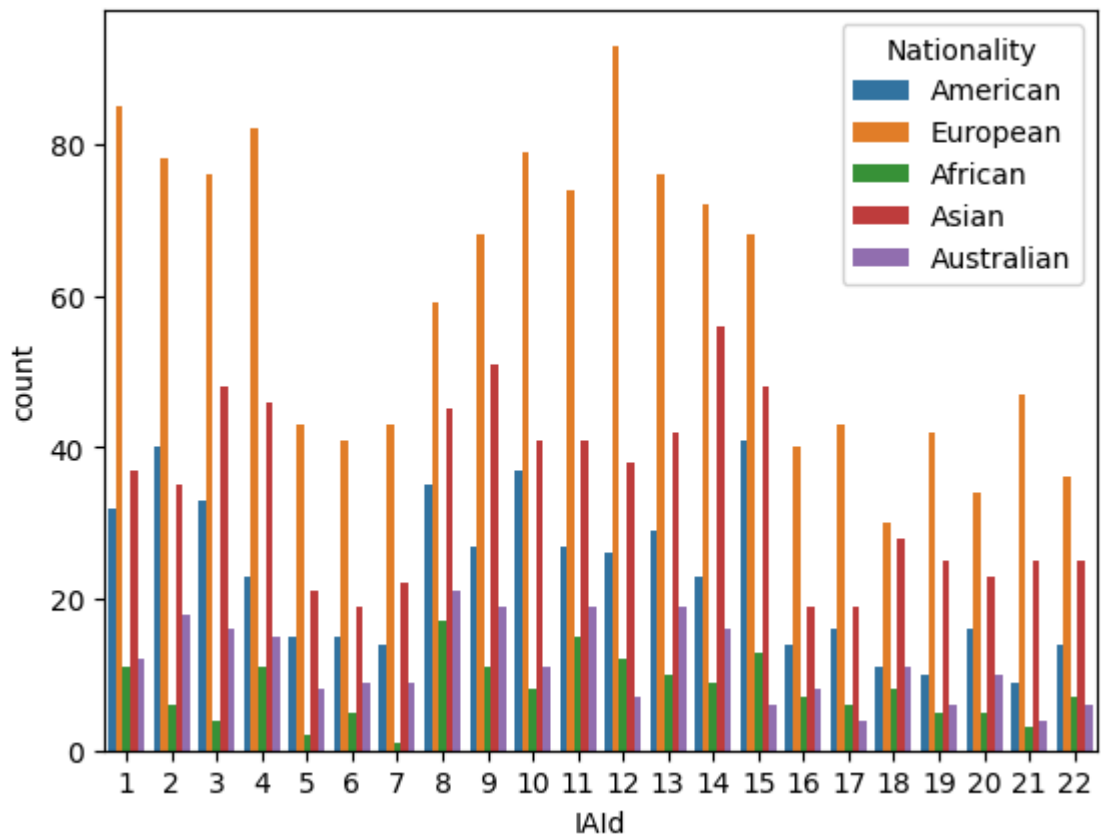
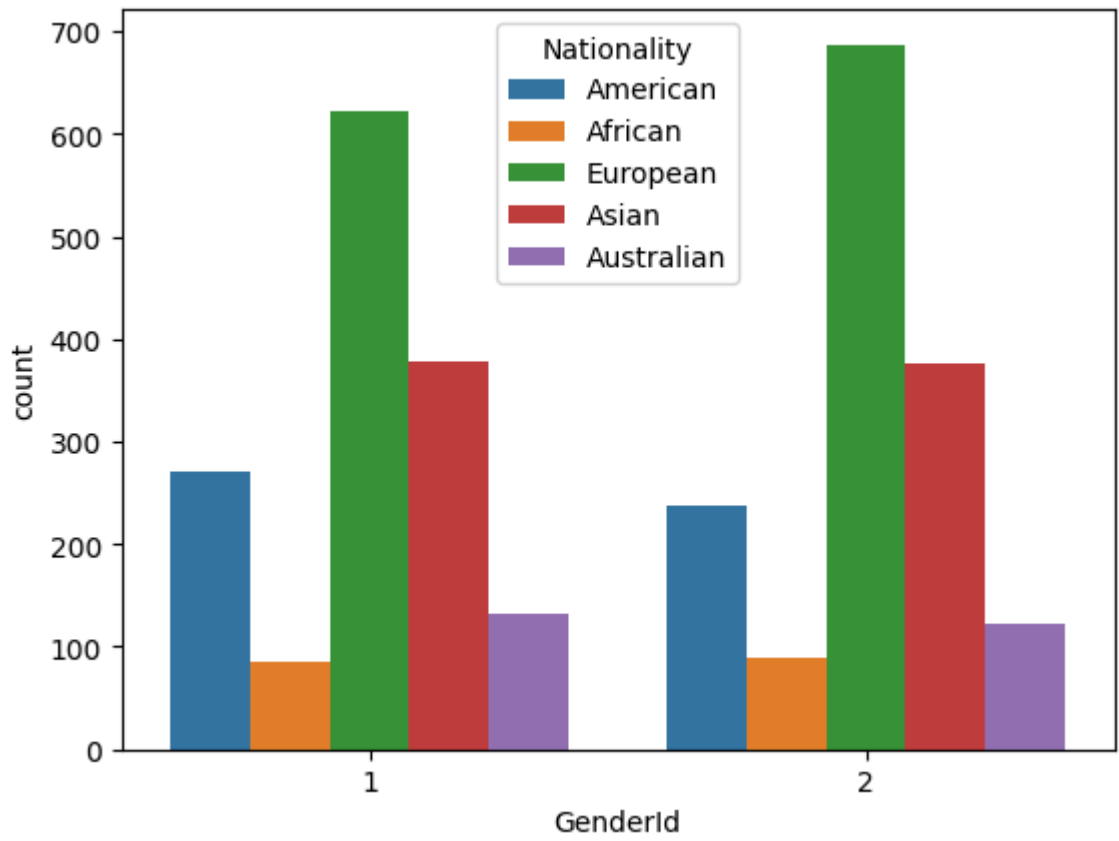


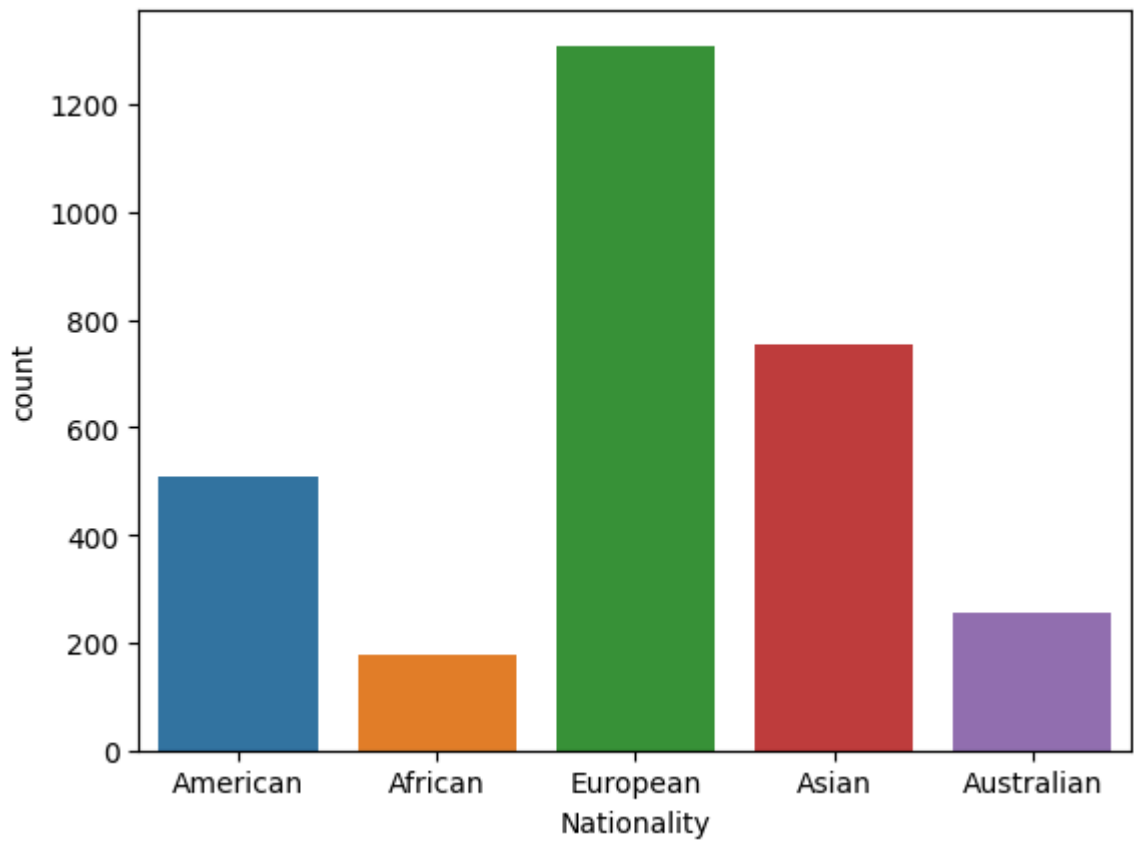
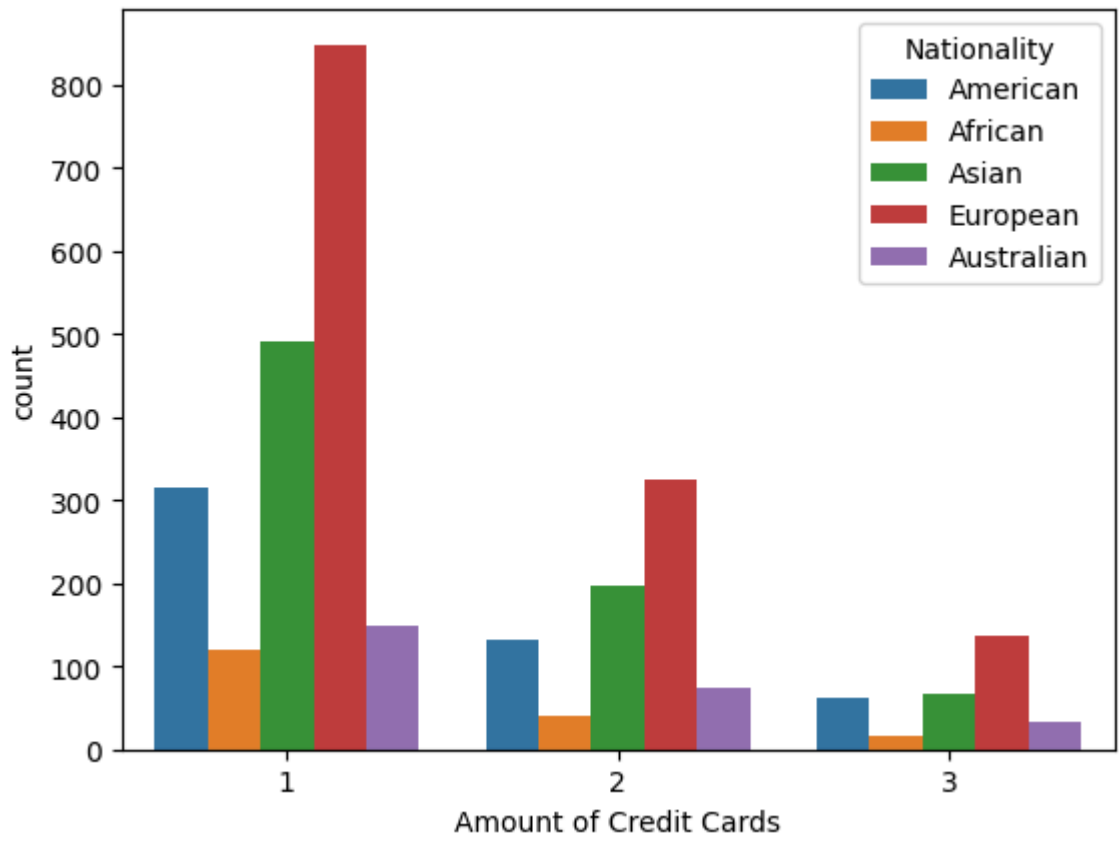


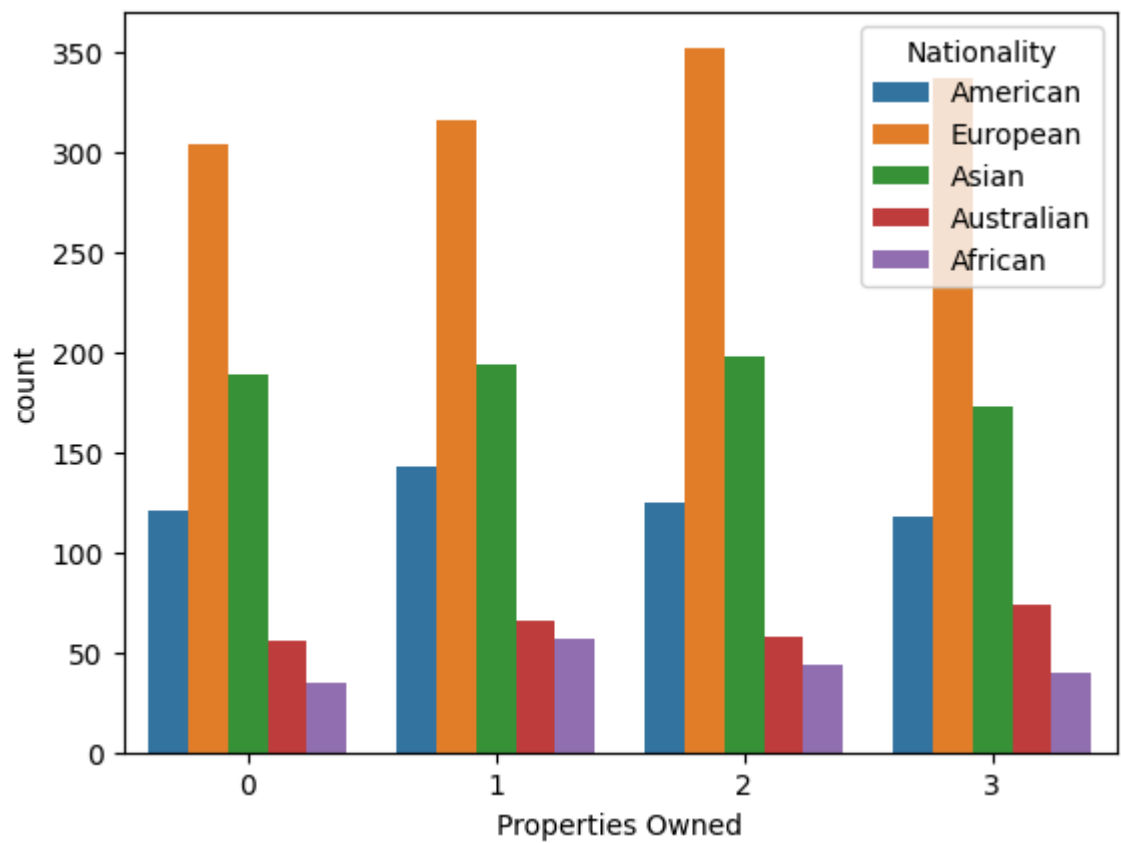
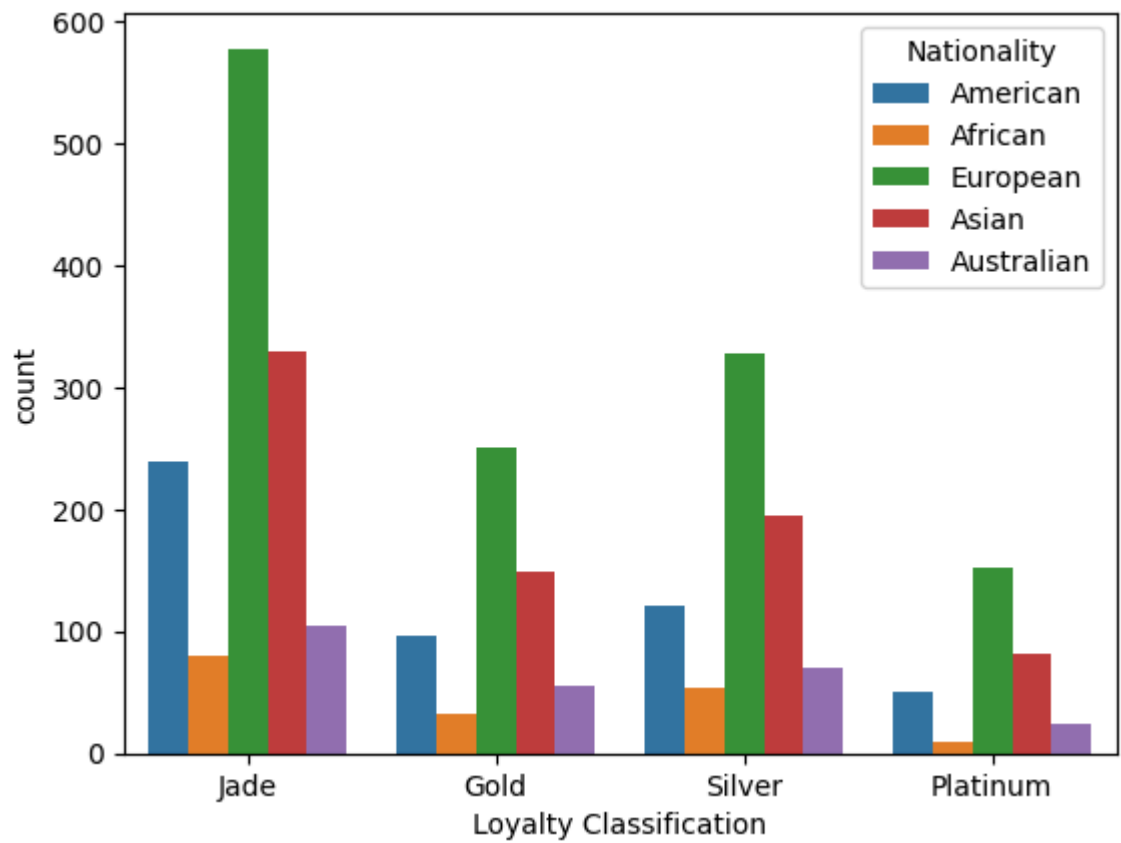


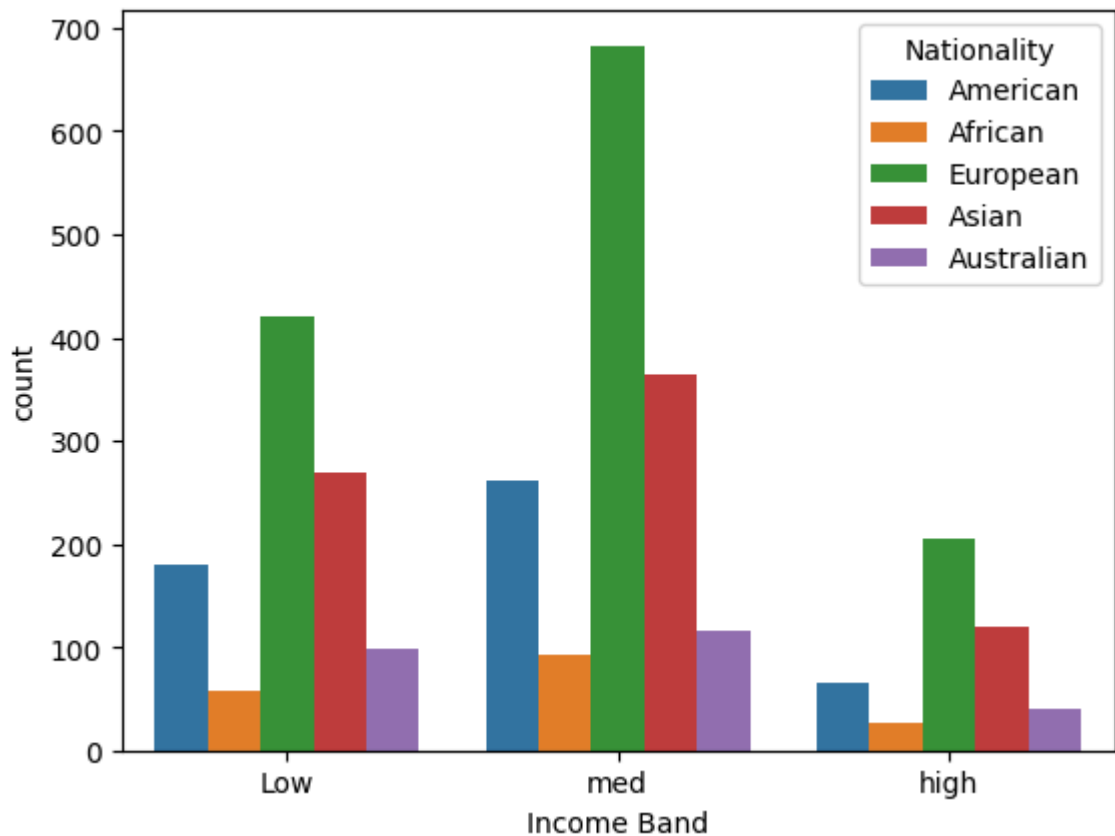
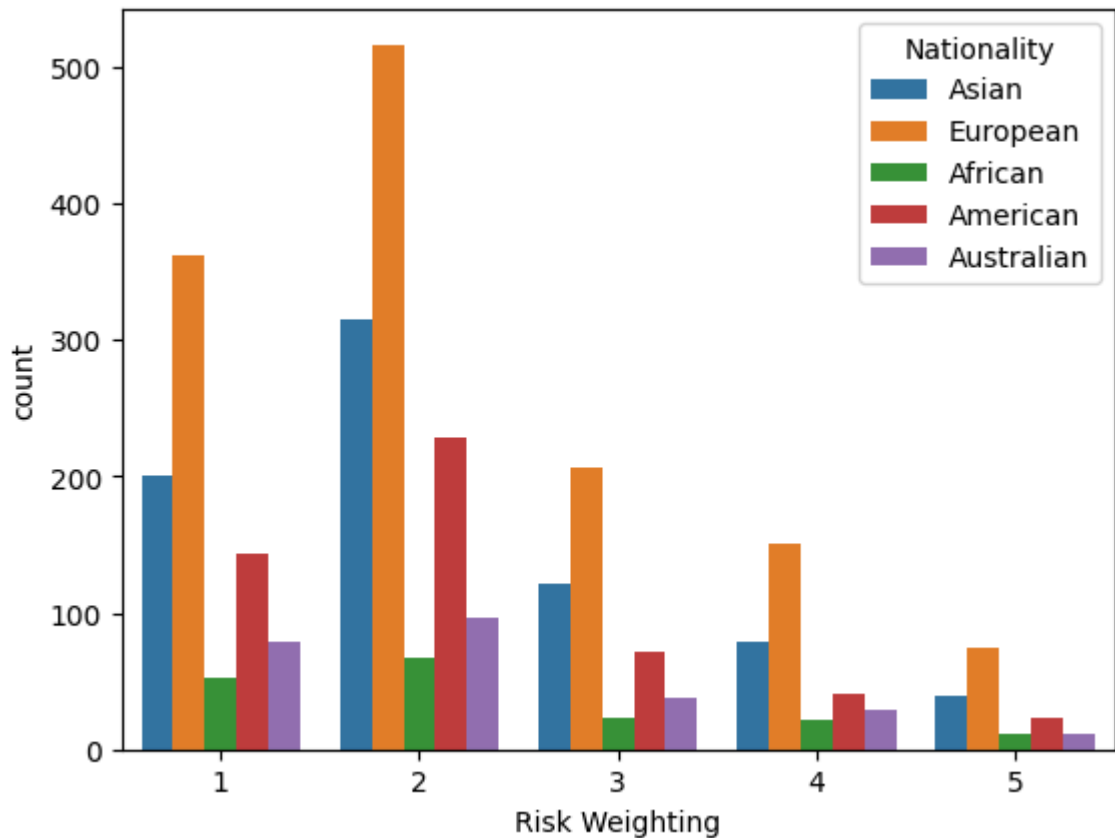
```
In [56]: for i, predictor in enumerate( df[["BRId", "GenderId", "IAId", "Amount of Credit"]):  
        plt.figure(i)  
        sns.countplot(data=df, x=predictor, hue='Nationality')  
        plt.figure(i)
```







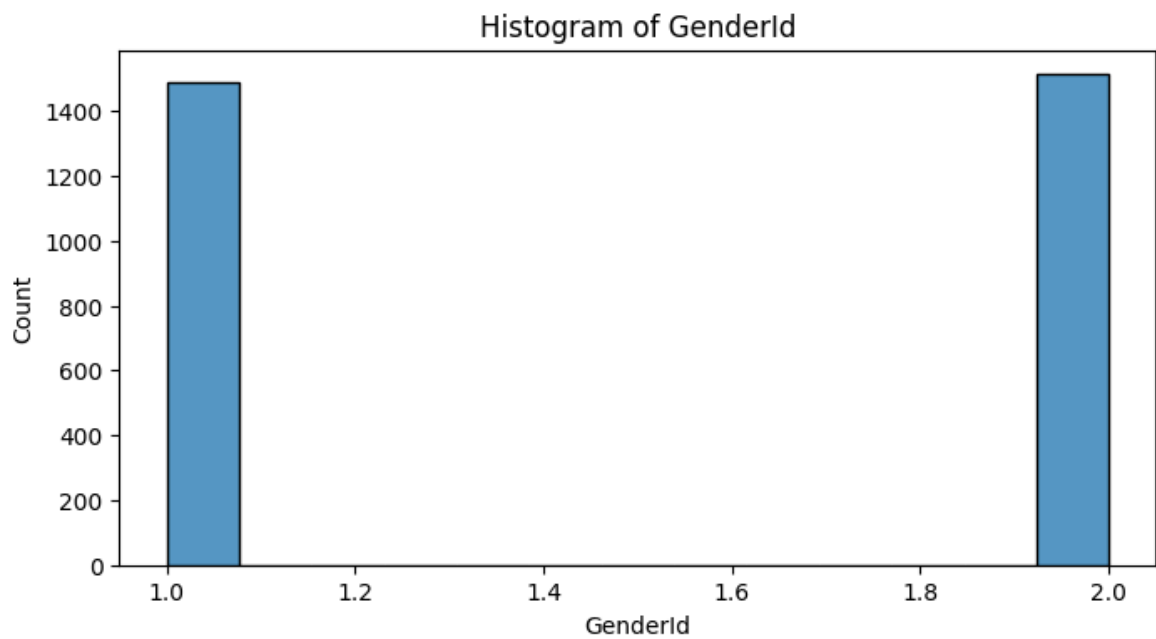
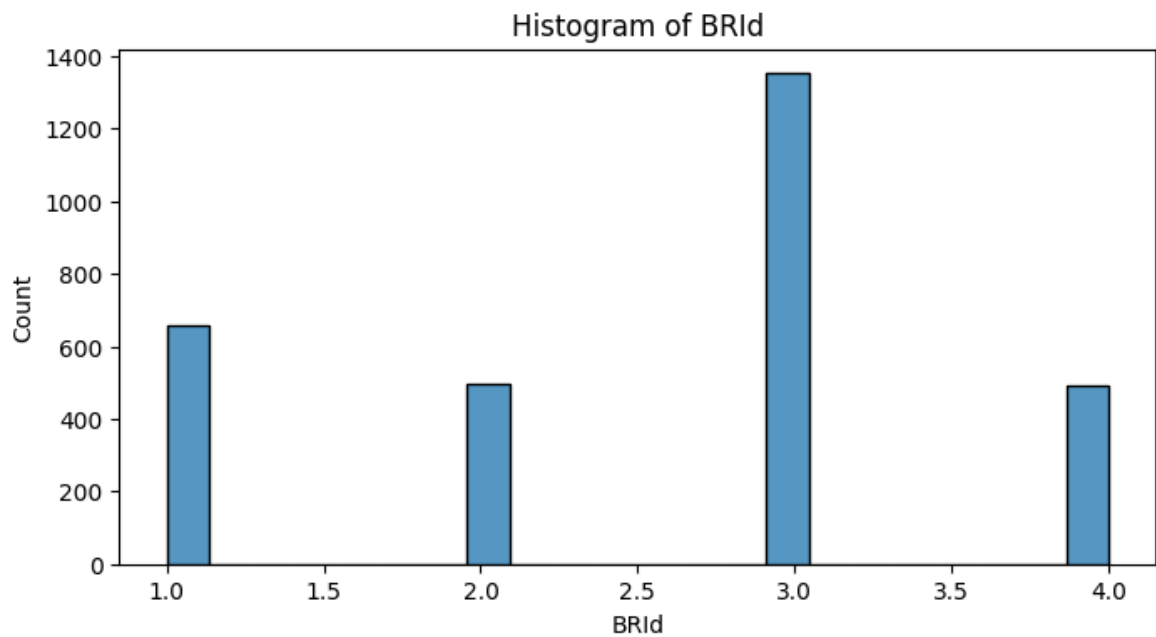


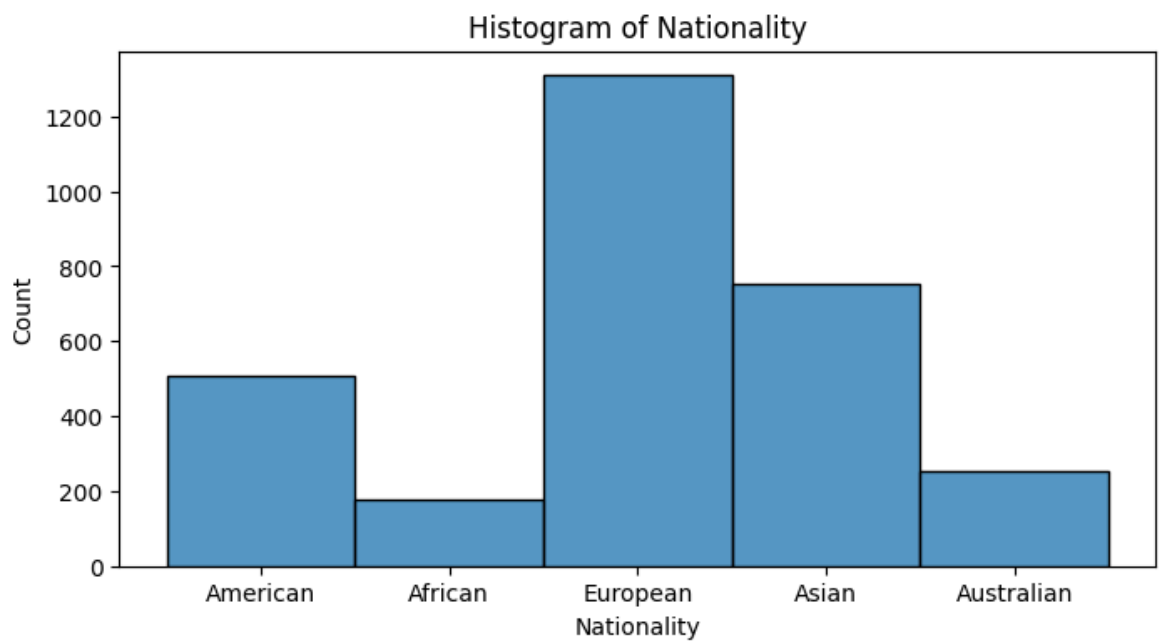
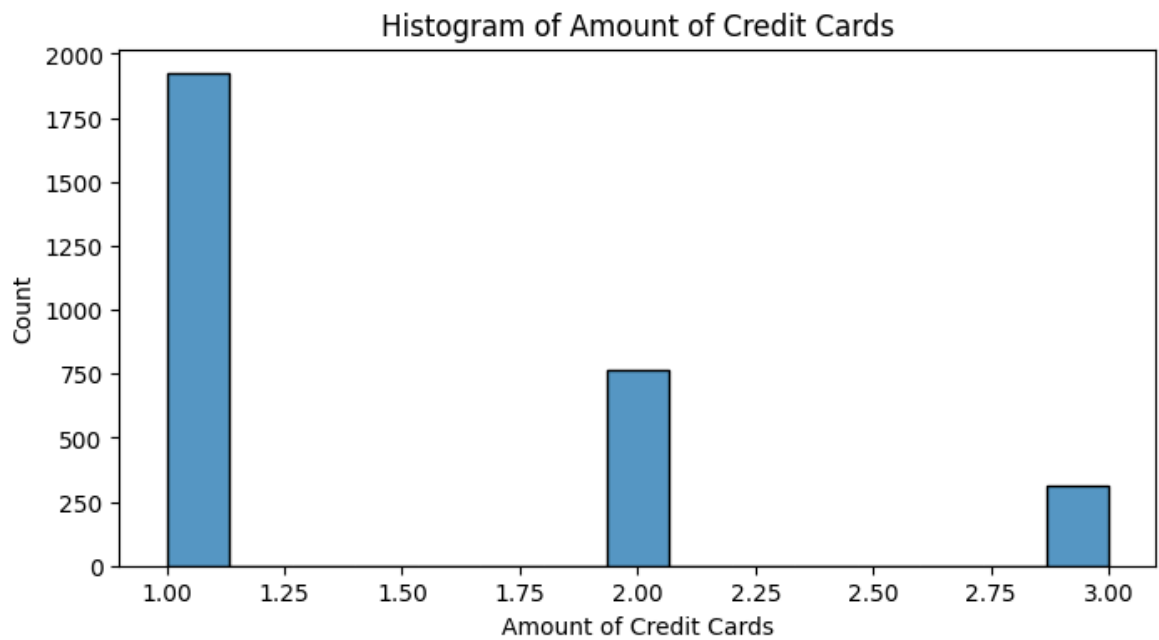
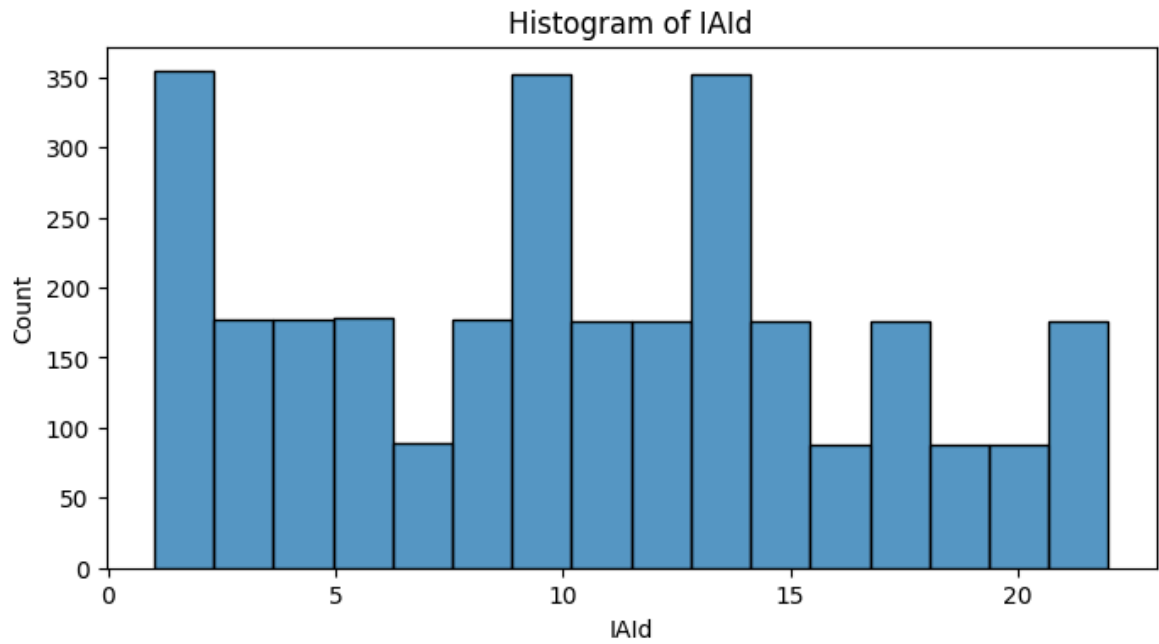


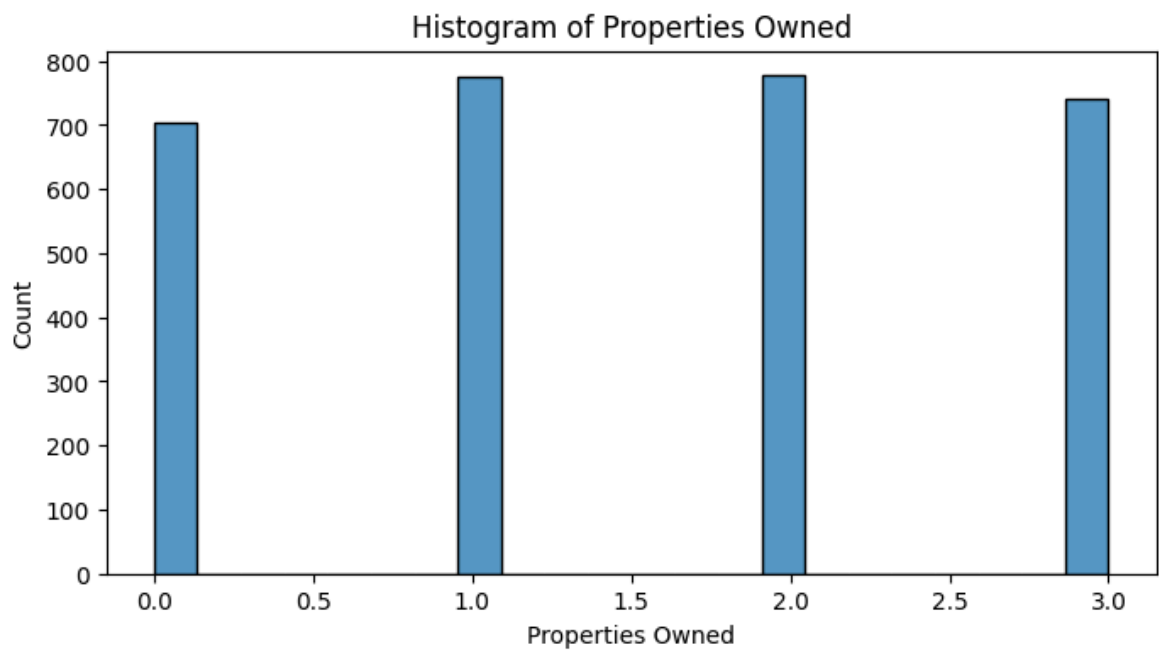
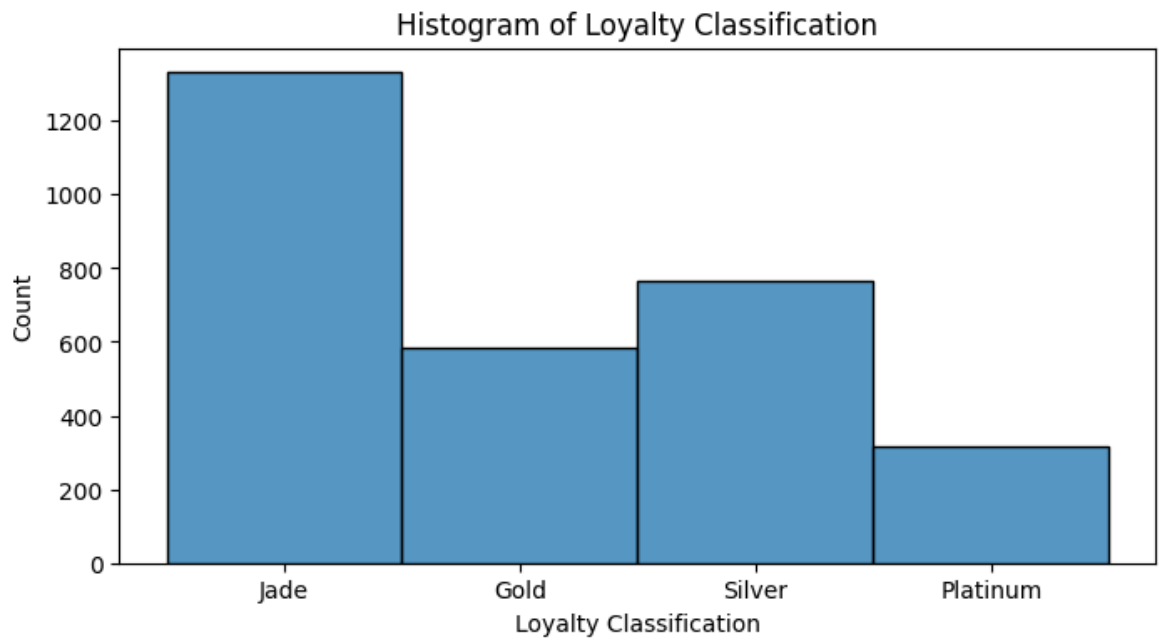
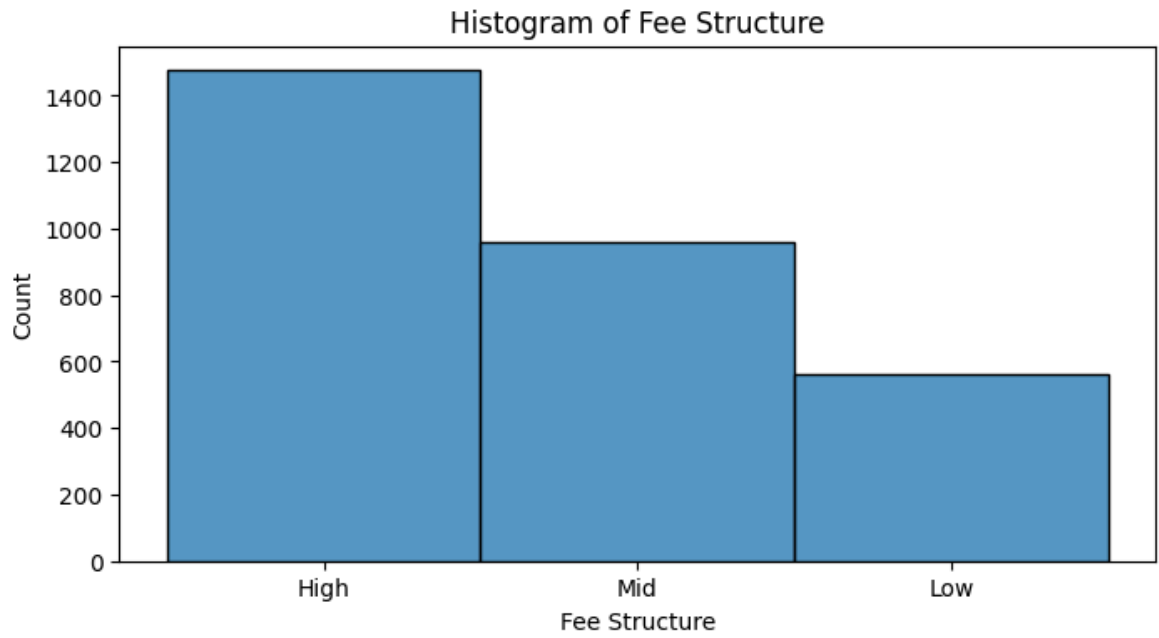
Histplot o value counts for different Occupation

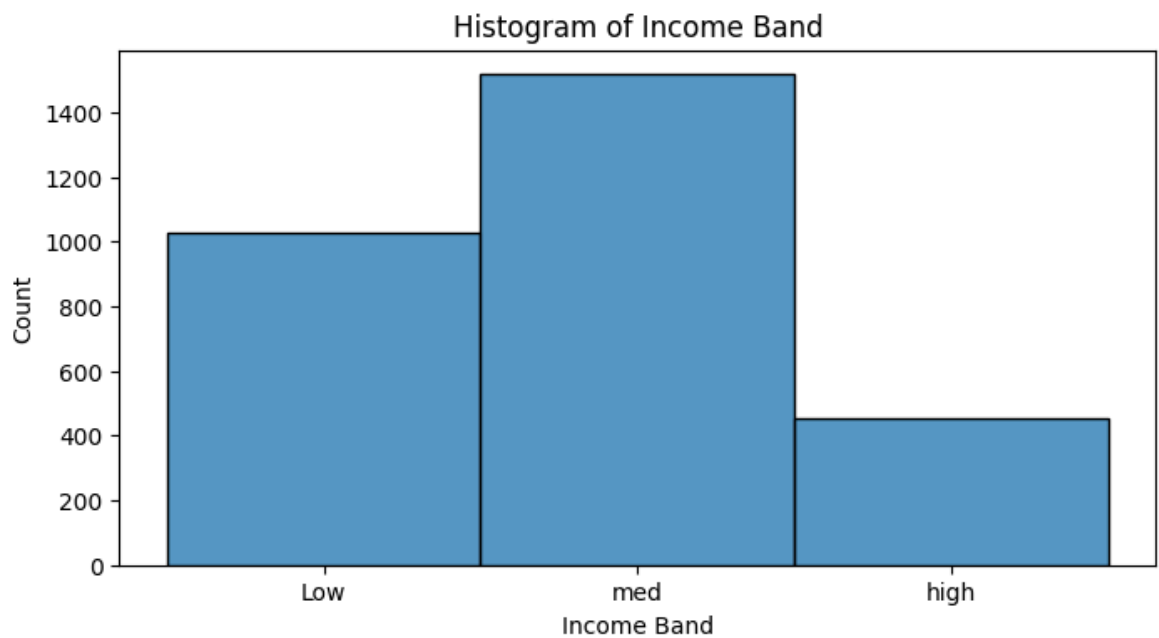
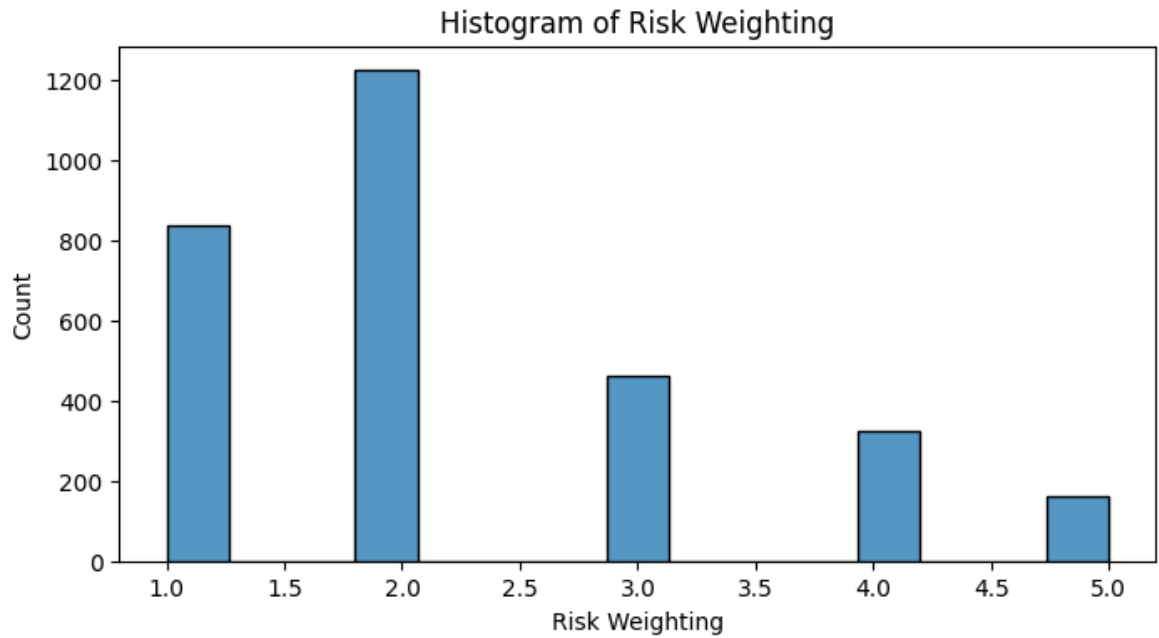
```
In [57]: for col in categorical_cols:
          if col != "Occupation":
```

```
plt.figure(figsize=(8,4))
sns.histplot(df[col])
plt.title(f'Histogram of {col}')
plt.xlabel(col)
plt.ylabel("Count")
plt.show()
```



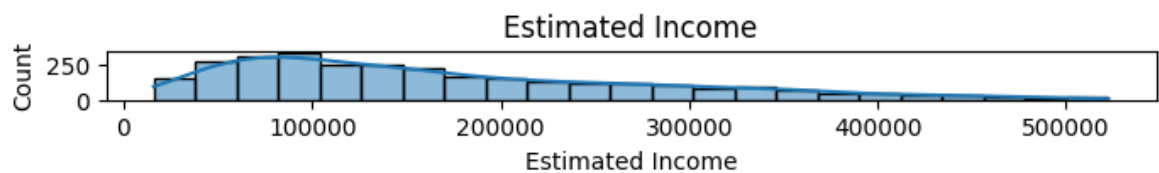


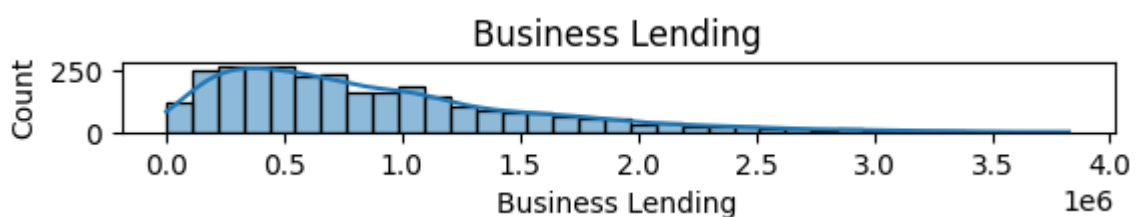
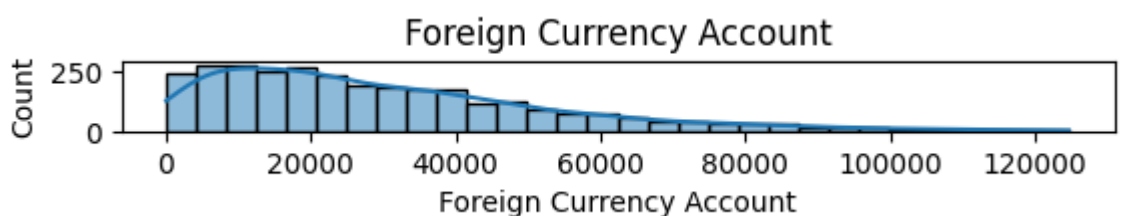
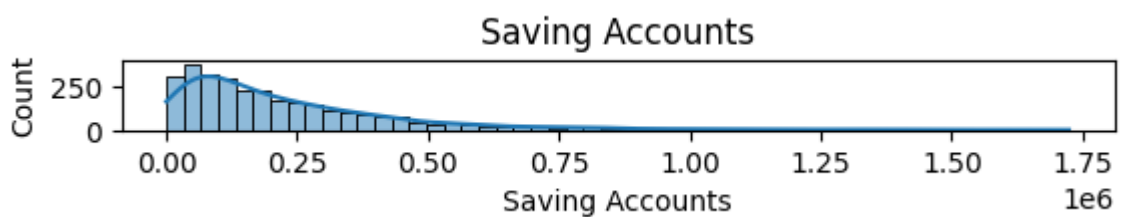
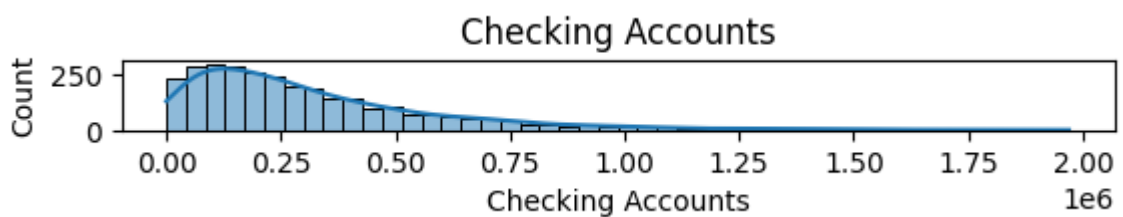
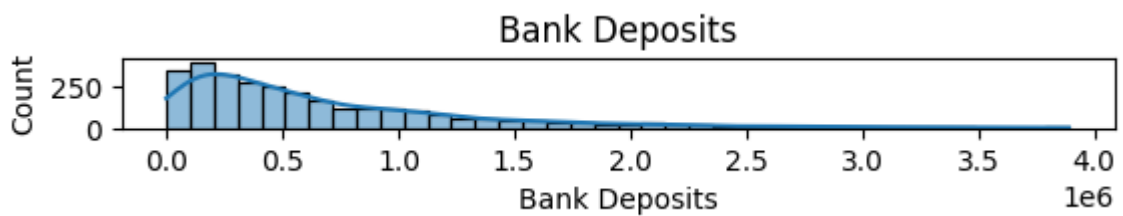
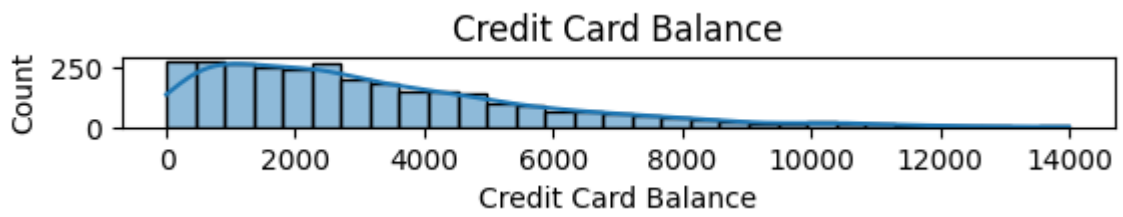
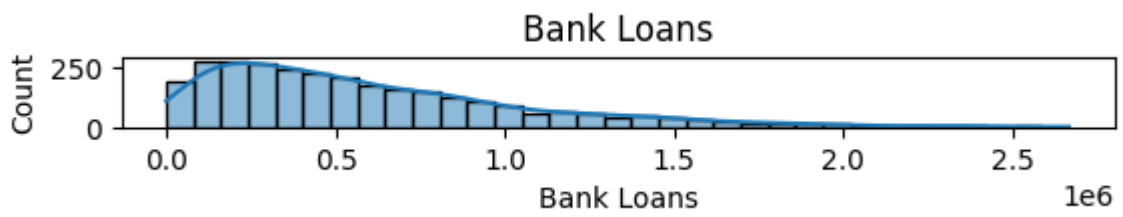
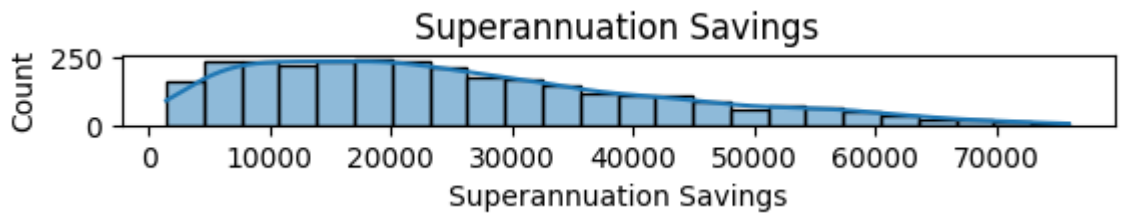




Numerical Analysis

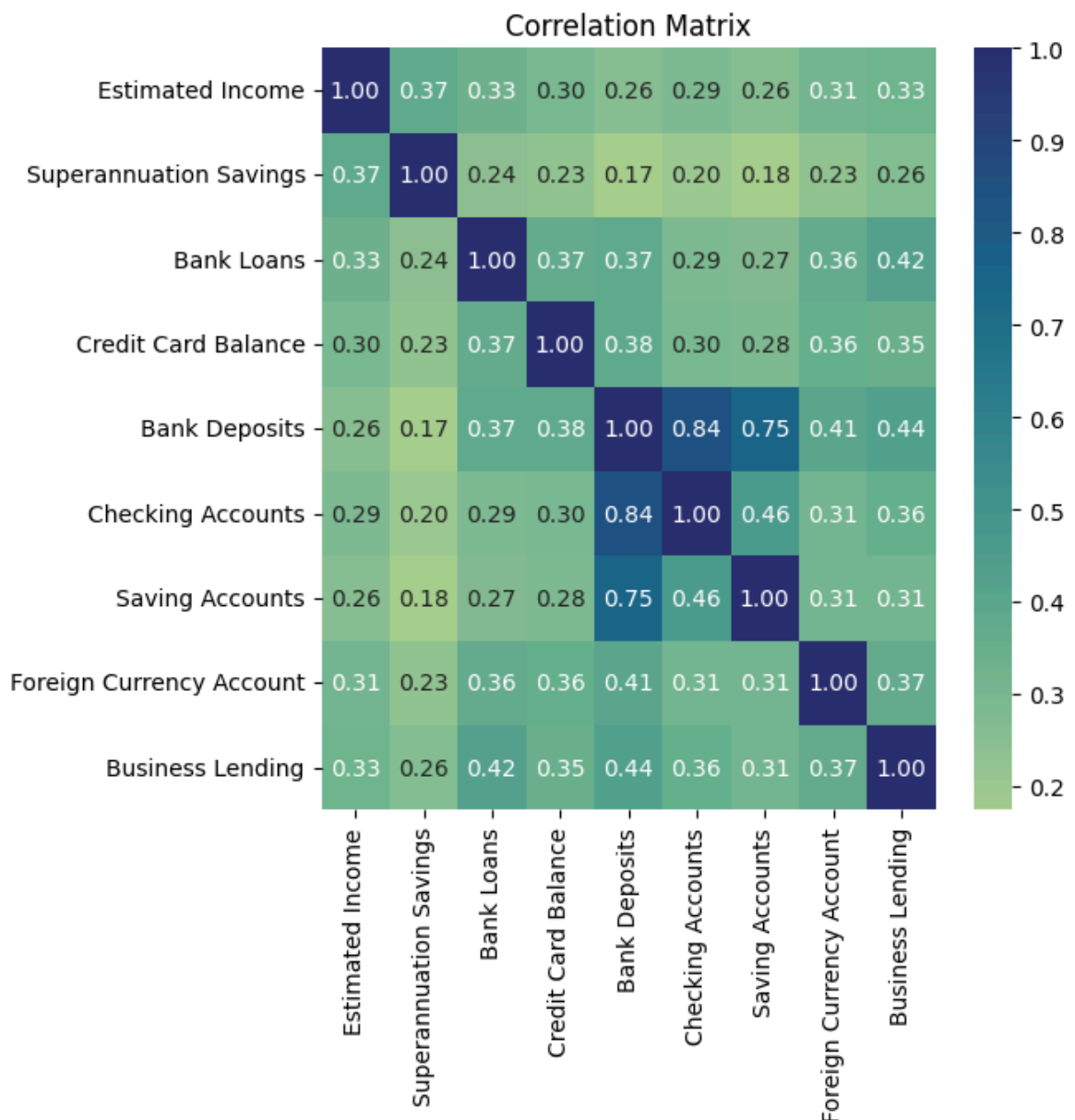
```
In [58]: numerical_cols = ['Estimated Income', 'Superannuation Savings', 'Bank Loans', 'C  
#univariate analysis and visualization  
plt.figure(figsize=(8,4))  
for i,col in enumerate(numerical_cols):  
    plt.subplot(9,1,i+1)  
    sns.histplot(df[col],kde=True)  
    plt.title(col)  
    plt.show()
```





Heatmaps

```
In [59]: numerical_cols = ['Estimated Income', 'Superannuation Savings', 'Bank Loans', 'C  
correlation_matrix = df[numerical_cols].corr()  
plt.figure(figsize=(6,6))  
sns.heatmap(correlation_matrix, annot=True, cmap='crest', fmt=".2f")  
plt.title("Correlation Matrix")  
plt.show()
```



Insights of EDA

1. The strongest poitive correlation occur among "Bank Deposits" with "Checking Accounts", and "foreign Currency Account" indicating that customers who maintain high balances in one account type often hold substancial amount/funds accross other accounts as well.

