```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

df = pd.read_csv('/content/Banking.csv')
df.head(5)
```

{"type":"dataframe","variable_name":"df"}

```python
df.shape
```

```
(3000, 25)
```

```python
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  IAId                      3000 non-null   int64
dtypes: float64(9), int64(8), object(8)
memory usage: 586.1+ KB
```

```python
# Generate descriptive statistics for the dataframe
df.describe()
```

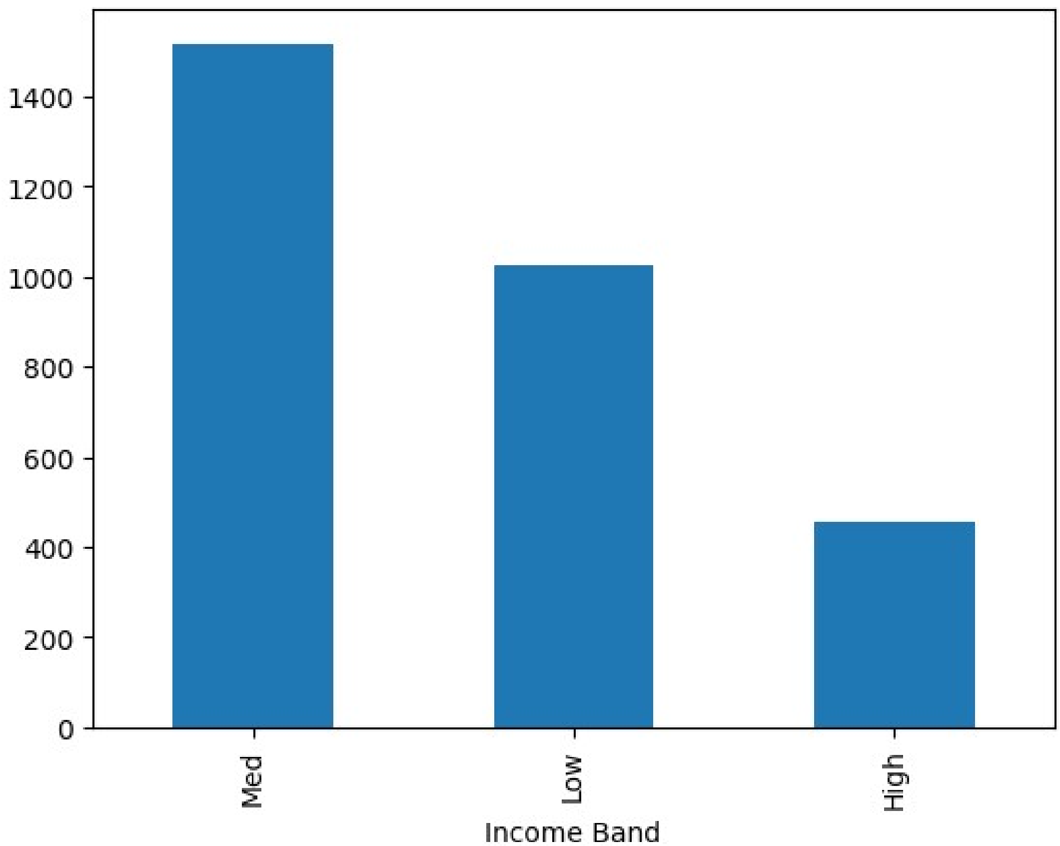
```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1044.4070732954572,\n
\"min\": 17.0,\n        \"max\": 3000.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
51.03966666666667,\n          51.0,\n          3000.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Location ID\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
14612.18148735417,\n        \"min\": 12.0,\n        \"max\": 43369.0,\
n        \"num_unique_values\": 8,\n        \"samples\": [\n
21563.323,\n          21129.5,\n          3000.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Estimated Income\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
165675.79415446558,\n        \"min\": 3000.0,\n        \"max\":
522330.26,\n        \"num_unique_values\": 8,\n        \"samples\": [\
n          171305.03426333333,\n          142313.47999999998,\n
3000.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Superannuation Savings\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 23834.521347506627,\n
\"min\": 1482.03,\n        \"max\": 75963.9,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
25531.59967333333,\n          22357.355000000003,\n          3000.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Amount of Credit Cards\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1060.1482852588065,\n        \"min\": 0.6763867645368546,\n
\"max\": 3000.0,\n        \"num_unique_values\": 6,\n
\"samples\": [\n          3000.0,\n          1.4636666666666667,\n
3.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Credit Card Balance\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 4303.156900678545,\n        \"min\":
1.17,\n        \"max\": 13991.99,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          3176.2069433333336,\n
2560.8050000000003,\n          3000.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Bank Loans\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
860222.1898574389,\n        \"min\": 0.0,\n        \"max\":
2667556.66,\n        \"num_unique_values\": 8,\n        \"samples\":
[\n          591386.1554866667,\n          479793.4,\n
3000.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Bank Deposits\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1272432.2911375419,\n        \"min\":
0.0,\n        \"max\": 3890598.08,\n        \"num_unique_values\": 8,\
n        \"samples\": [\n          671560.1939233334,\n
```

```
463316.46,\n          3000.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Checking Accounts\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 643980.7752101668,\n
\"min\": 0.0,\n        \"max\": 1969923.08,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
321092.94912666664,\n          242815.655,\n          3000.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Saving Accounts\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 569501.1225021764,\n        \"min\":
0.0,\n        \"max\": 1724118.36,\n        \"num_unique_values\": 8,\
n        \"samples\": [\n          232908.3534833333,\n
164086.555,\n          3000.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Foreign Currency Account\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 39821.13354767674,\n
\"min\": 45.0,\n        \"max\": 124704.87,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
29883.529993333334,\n          24341.190000000002,\n          3000.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Business Lending\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1231479.8807215113,\n        \"min\": 0.0,\n        \"max\":
3825961.94,\n        \"num_unique_values\": 8,\n        \"samples\":
[\n          866759.8084066667,\n          711314.6599999999,\n
3000.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Properties Owned\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1060.1241040744355,\n        \"min\":
0.0,\n        \"max\": 3000.0,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          3000.0,\n          1.5186666666666666,\n
2.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Risk Weighting\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1059.8841834225843,\n        \"min\":
1.0,\n        \"max\": 3000.0,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          3000.0,\n          2.2493333333333334,\n
3.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"BRId\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1059.8239053751968,\n        \"min\": 1.0,\n        \"max\":
3000.0,\n        \"num_unique_values\": 7,\n        \"samples\": [\n
3000.0,\n          2.5593333333333335,\n          3.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"GenderId\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
1060.1550392950937,\n        \"min\": 0.5000673512490724,\n
\"max\": 3000.0,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          1.504,\n          2.0,\n
```

0.5000673512490724\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"IAId\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1057.1739338184325,\n   \"min\": 1.0,\n   \"max\": 3000.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   10.425333333333333,\n   10.0,\n   3000.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
bins = [0, 100000, 300000, float('inf')]
labels = ['Low', 'Med', 'High']

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

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

```
<Axes: xlabel='Income Band'>
```




```
# Examine the distribution of unique cataegories in categorical
columns
categorical_cols = df[["BRId", "GenderId", "IAId", "Amount of Credit
Cards", "Nationality", "Occupation", "Fee Structure", "Loyalty
```

```
Classification", "Properties Owned", "Risk Weighting", "Income
Band"]].columns

for col in categorical_cols:
  print(f"Value Counts for '{col}':")
  display(df[col].value_counts())
```

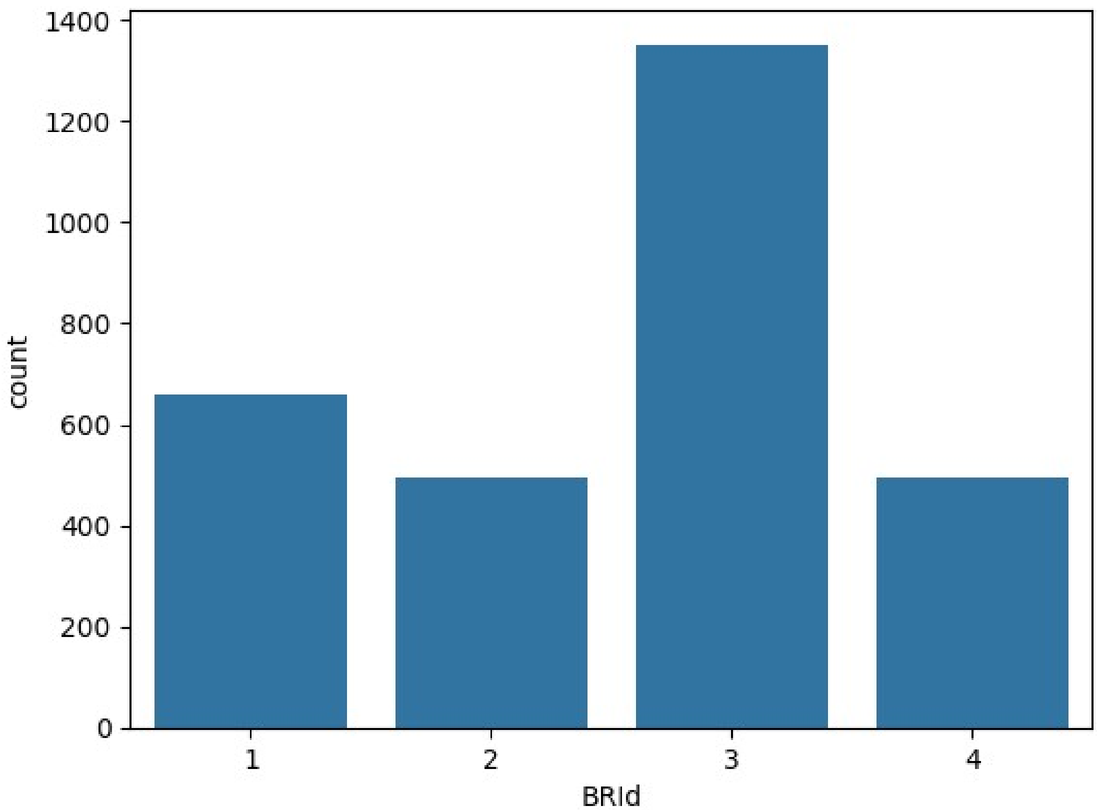
```
Value Counts for 'BRId':

BRId
3    1352
1     660
2     495
4     493
Name: count, dtype: int64

Value Counts for 'GenderId':

GenderId
2    1512
1    1488
Name: count, dtype: int64

Value Counts for 'IAId':

IAId
1     177
3     177
4     177
8     177
2     177
11    176
15    176
14    176
13    176
12    176
10    176
9     176
7      89
6      89
5      89
16     88
17     88
18     88
19     88
20     88
21     88
22     88
Name: count, dtype: int64

Value Counts for 'Amount of Credit Cards':
```

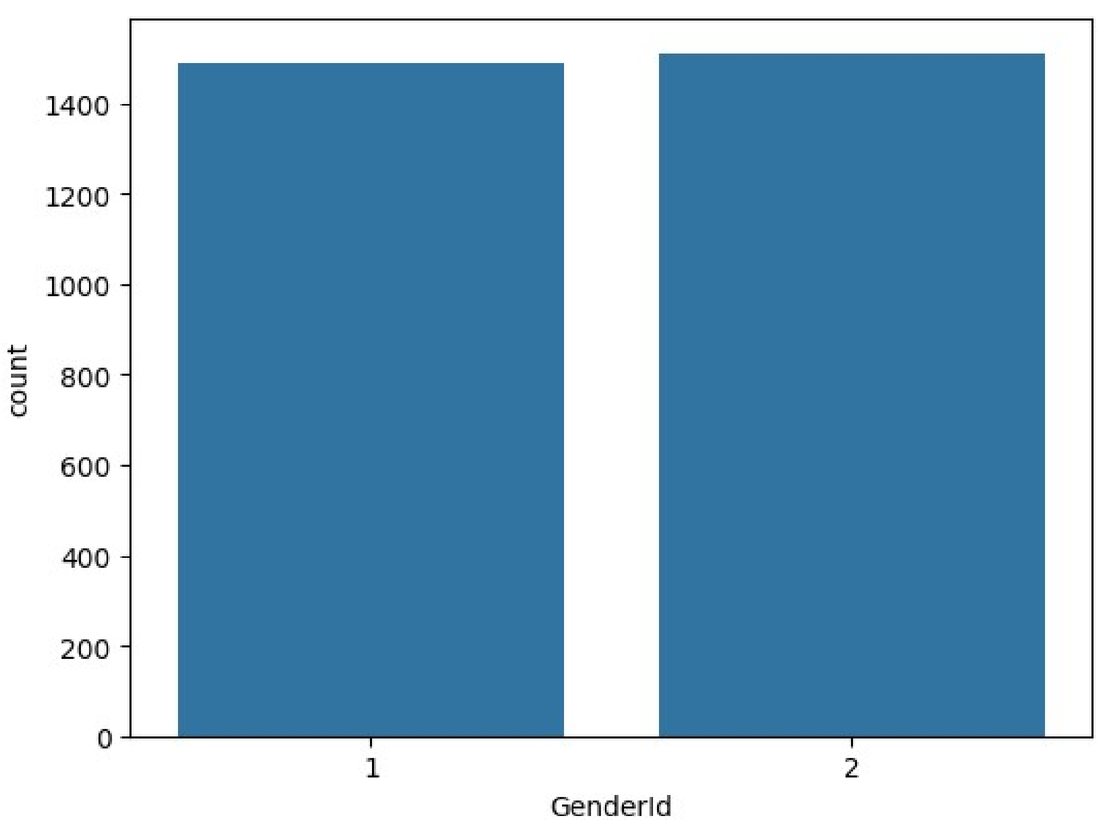
```
Amount of Credit Cards
1    1922
2     765
3     313
Name: count, dtype: int64

Value Counts for 'Nationality':

Nationality
European      1309
Asian          754
American       507
Australian     254
African        176
Name: count, dtype: int64

Value Counts for 'Occupation':

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

Value Counts for 'Fee Structure':

Fee Structure
High    1476
Mid      962
Low      562
Name: count, dtype: int64

Value Counts for 'Loyalty Classification':

Loyalty Classification
Jade        1331
Silver       767
Gold         585
Platinum     317
Name: count, dtype: int64

Value Counts for 'Properties Owned':
```

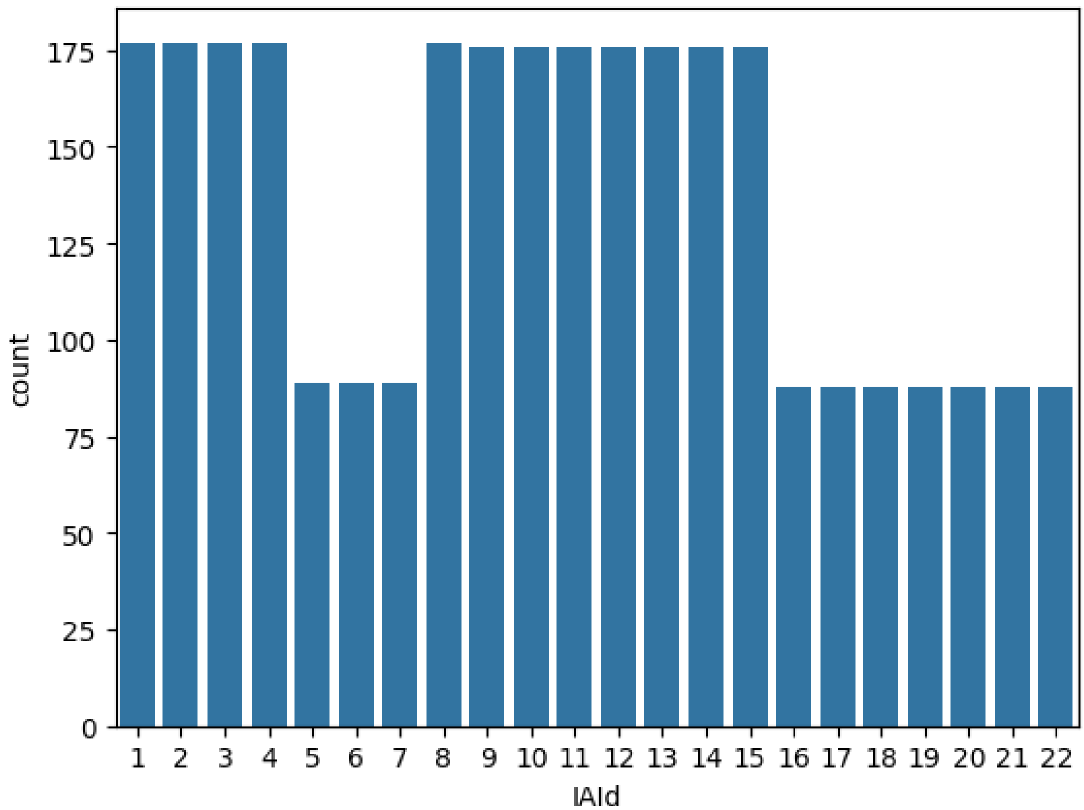
```
Properties Owned
2    777
1    776
3    742
0    705
Name: count, dtype: int64

Value Counts for 'Risk Weighting':

Risk Weighting
2    1222
1     836
3     460
4     322
5     160
Name: count, dtype: int64

Value Counts for 'Income Band':

Income Band
Med     1517
Low     1027
High     456
Name: count, dtype: int64
```

Univariate Analysis

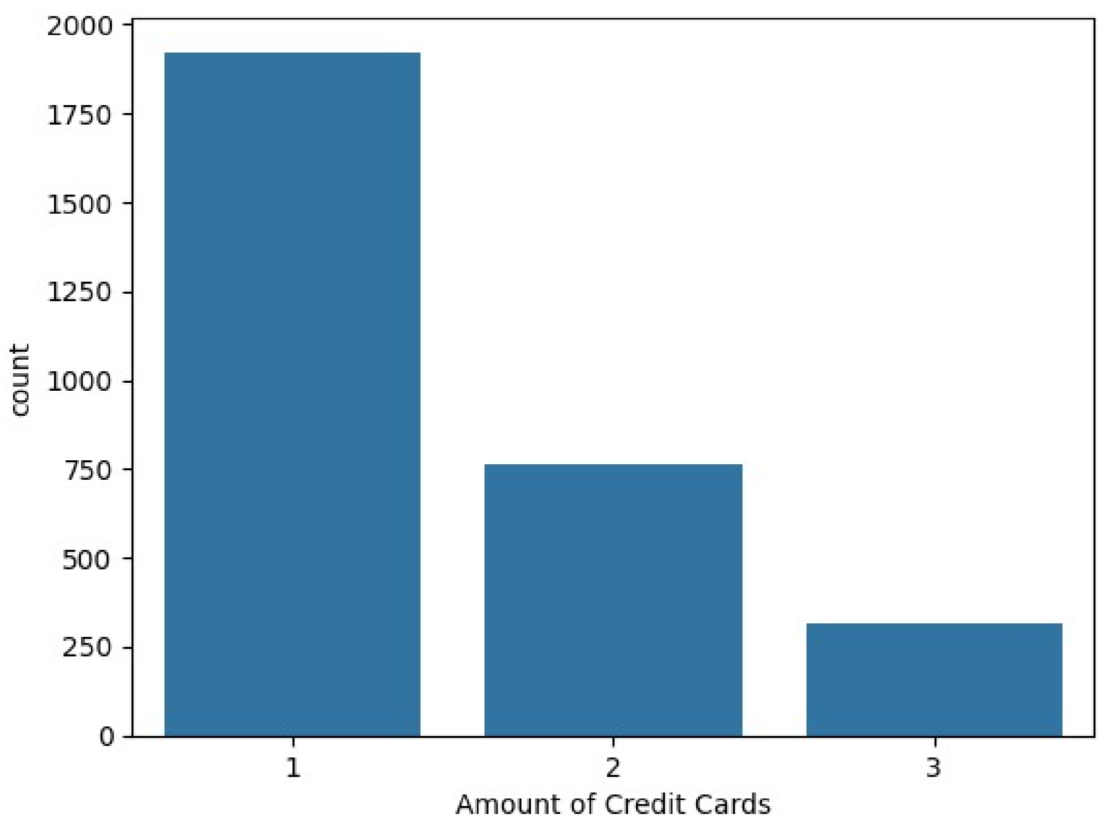
```
for i, predictor in enumerate(df[["BRId", "GenderId", "IAId", "Amount 
of Credit Cards", "Nationality", "Occupation", "Fee Structure", 
"Loyalty Classification", "Properties Owned", "Risk Weighting", 
"Income Band"]].columns):
    plt.figure(i)
    sns.countplot(data=df, x=predictor)
```


1400
1200
1000
800
600
400
200
0
count
1
2
3
4
BRId

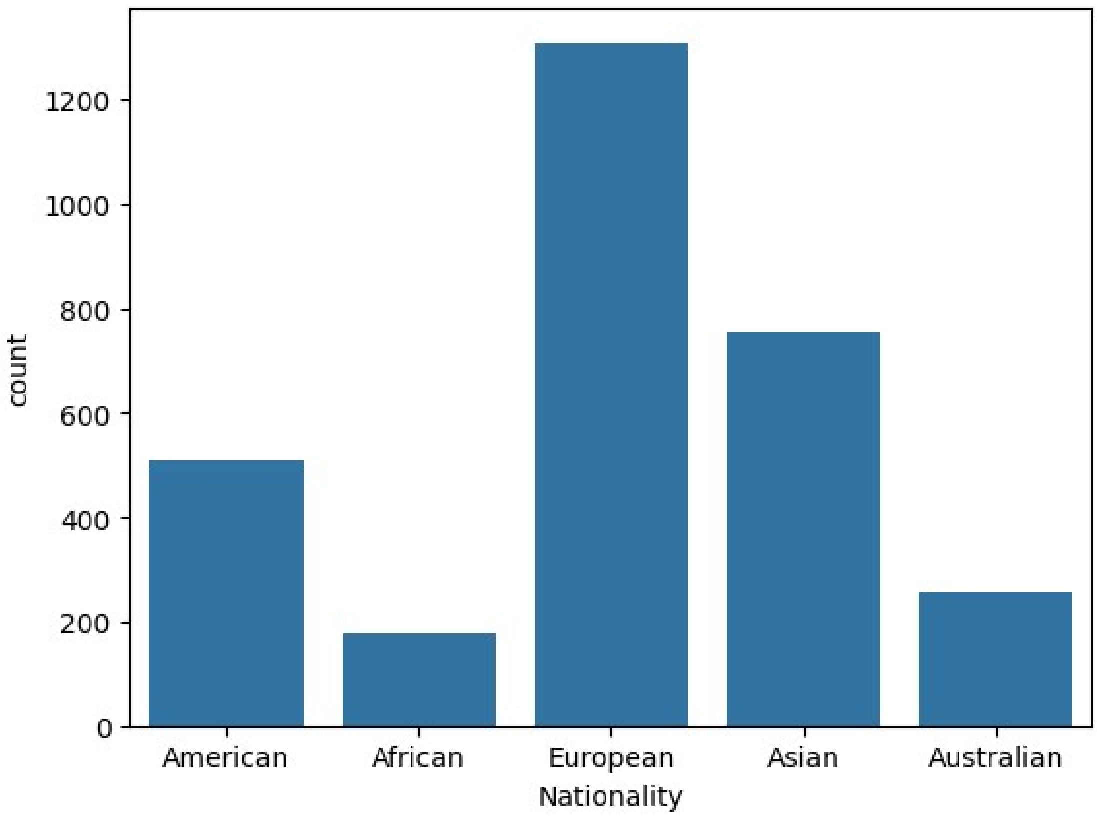

1400
1200
1000
800
600
400
200
0
count
1
2
GenderId



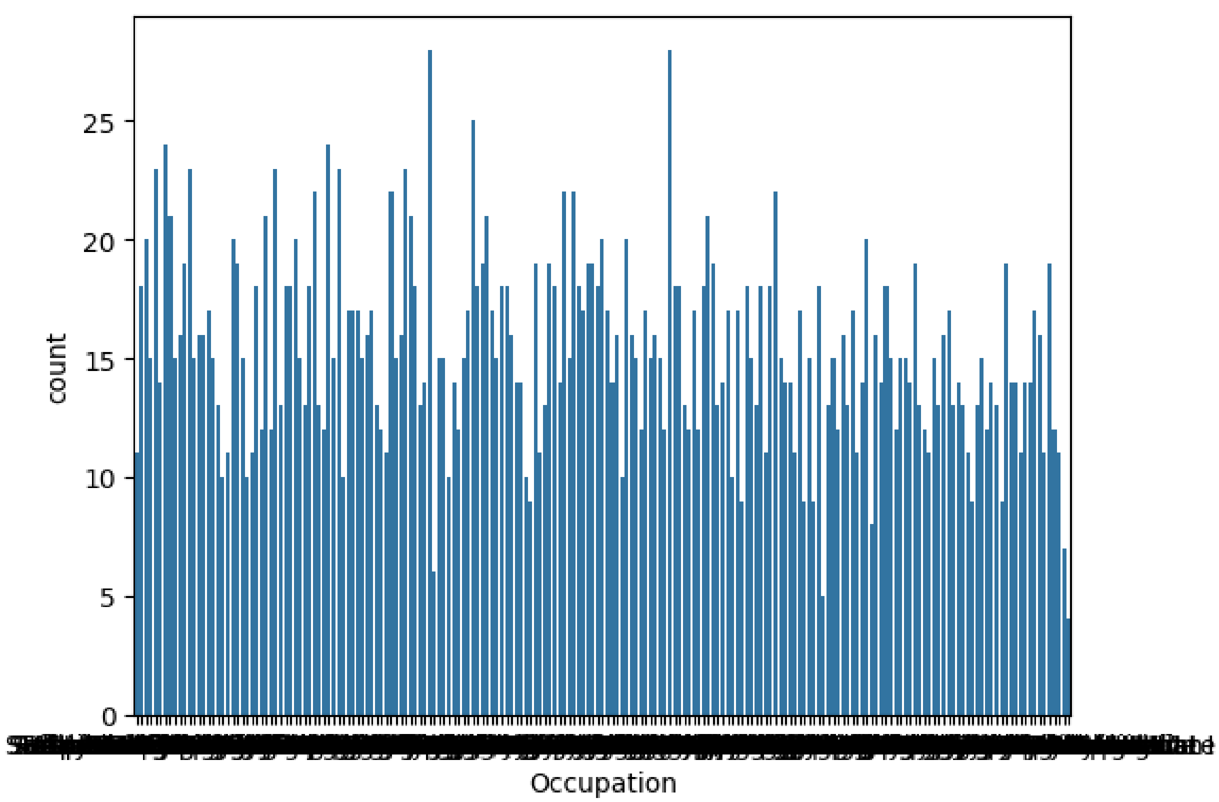
175
150
125
100
75
50
25
0
count
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
IAId



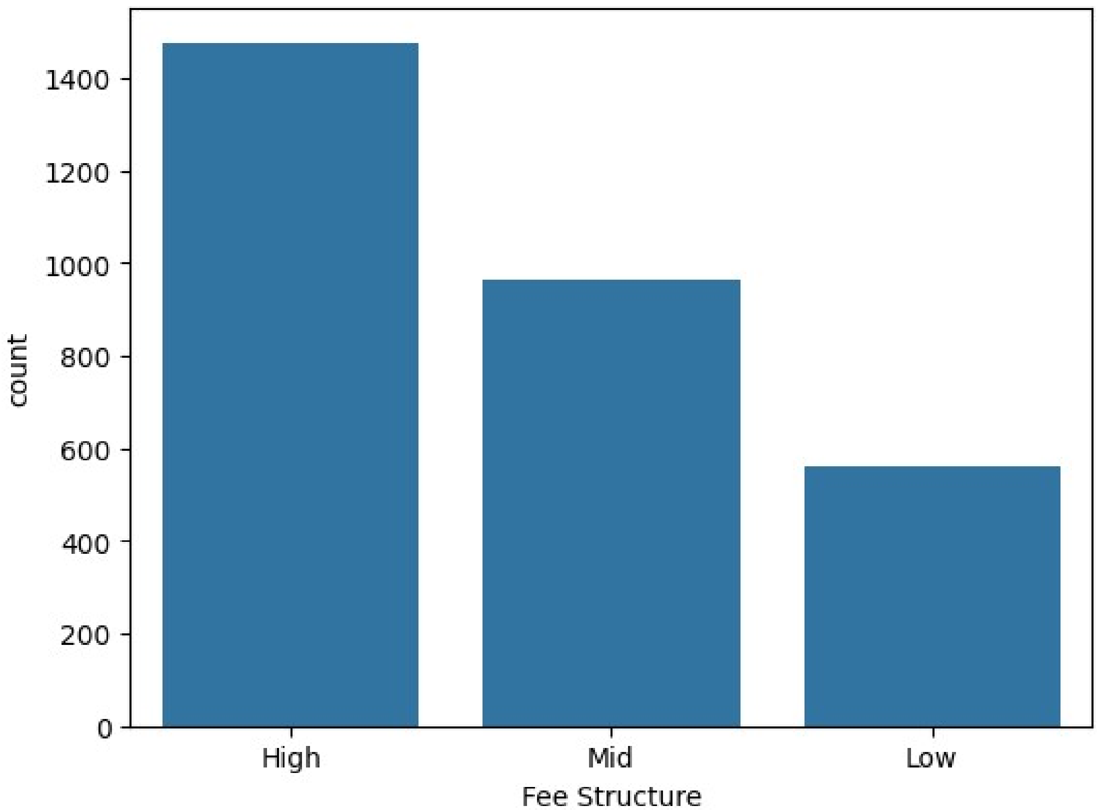
2000
1750
1500
1250
1000
750
500
250
0
count
1 2 3
Amount of Credit Cards



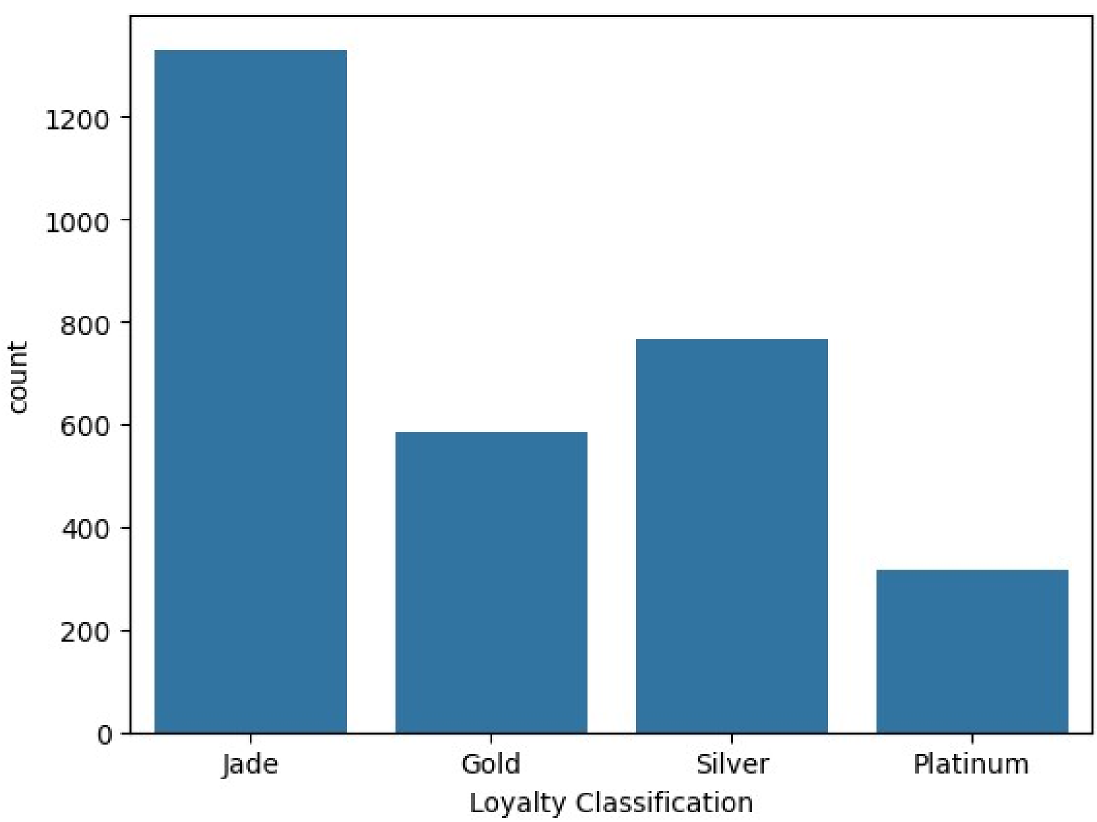
1200
1000
800
600
400
200
0
count
American
African
European
Asian
Australian
Nationality



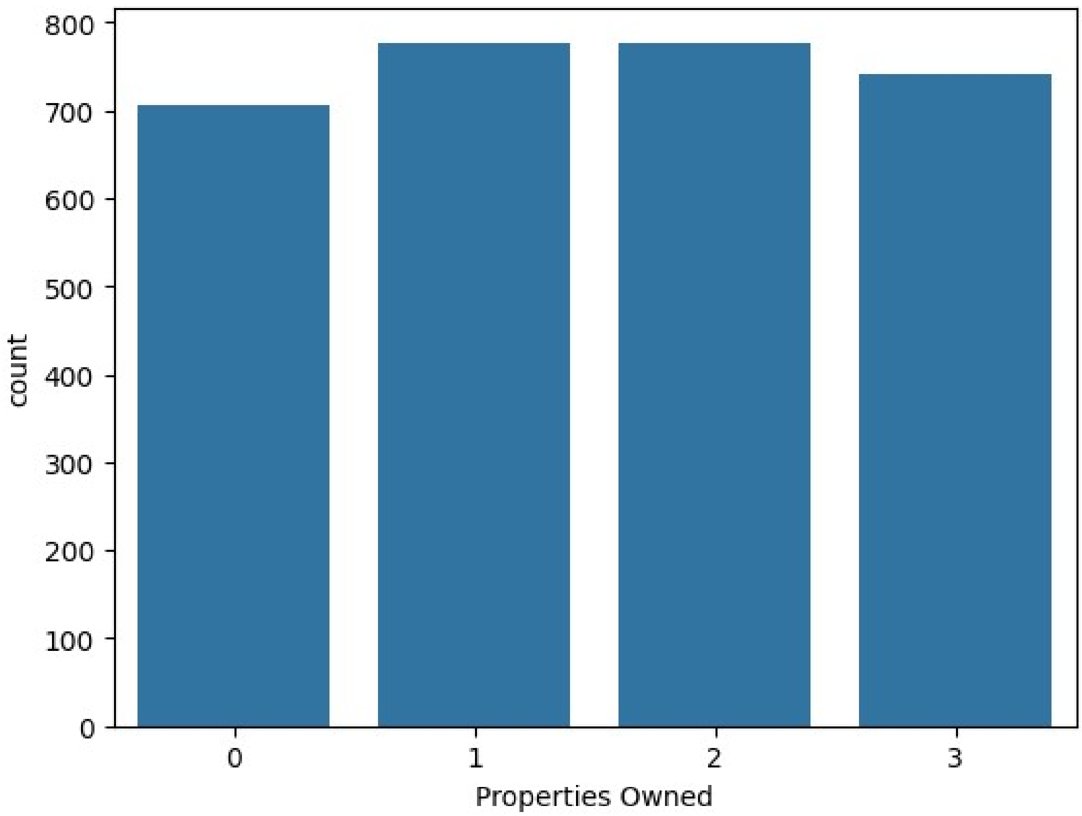
25
20
15
10
5
0
count
Occupation

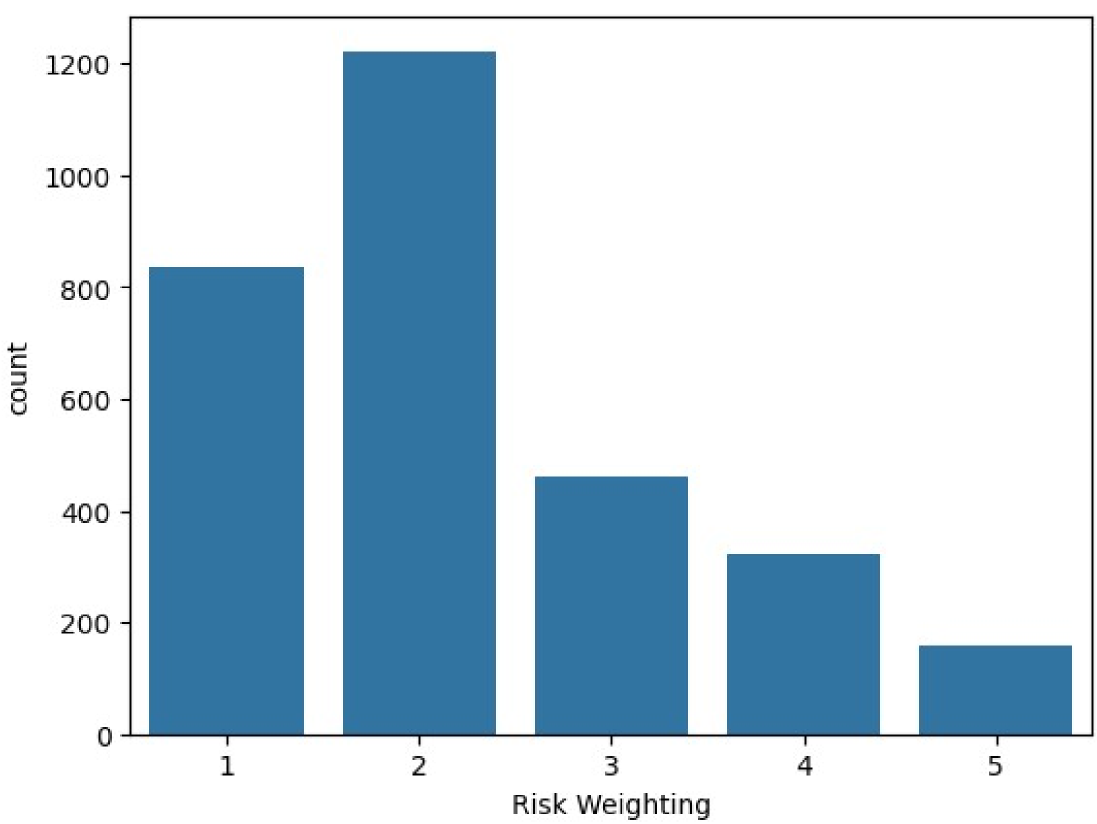


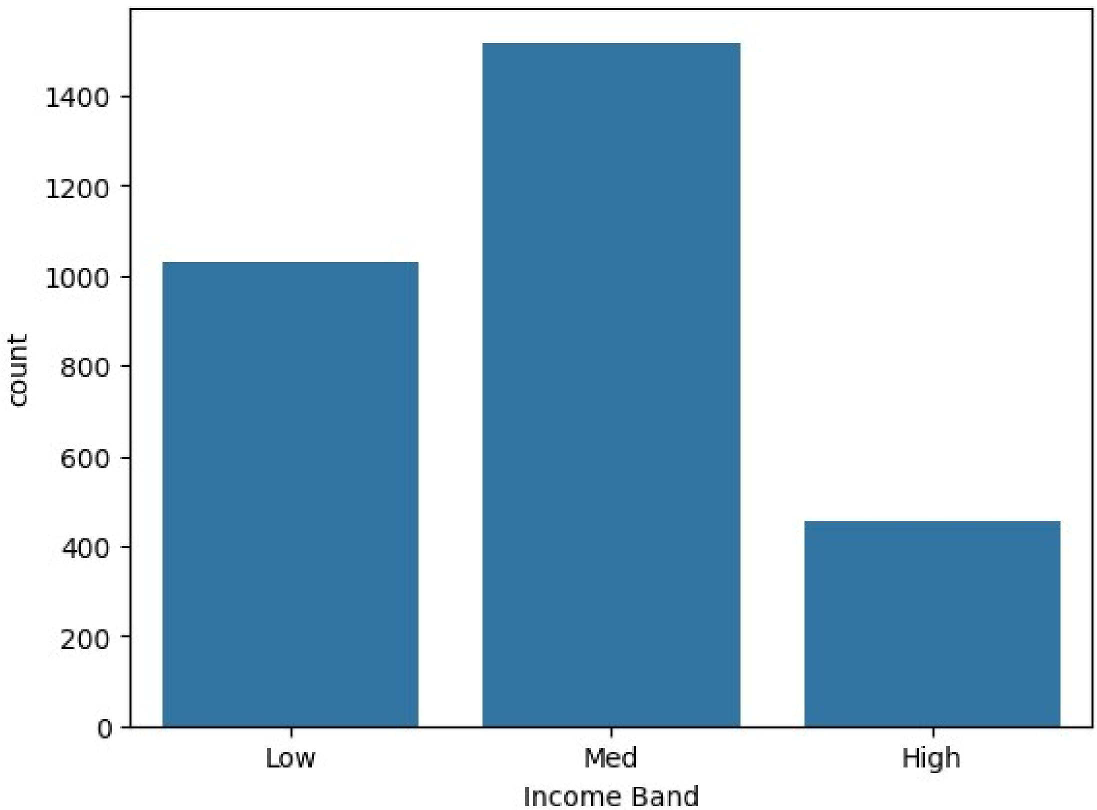
count
1400
1200
1000
800
600
400
200
0
High
Mid
Low
Fee Structure



count
1200
1000
800
600
400
200
0
Jade
Gold
Silver
Platinum
Loyalty Classification

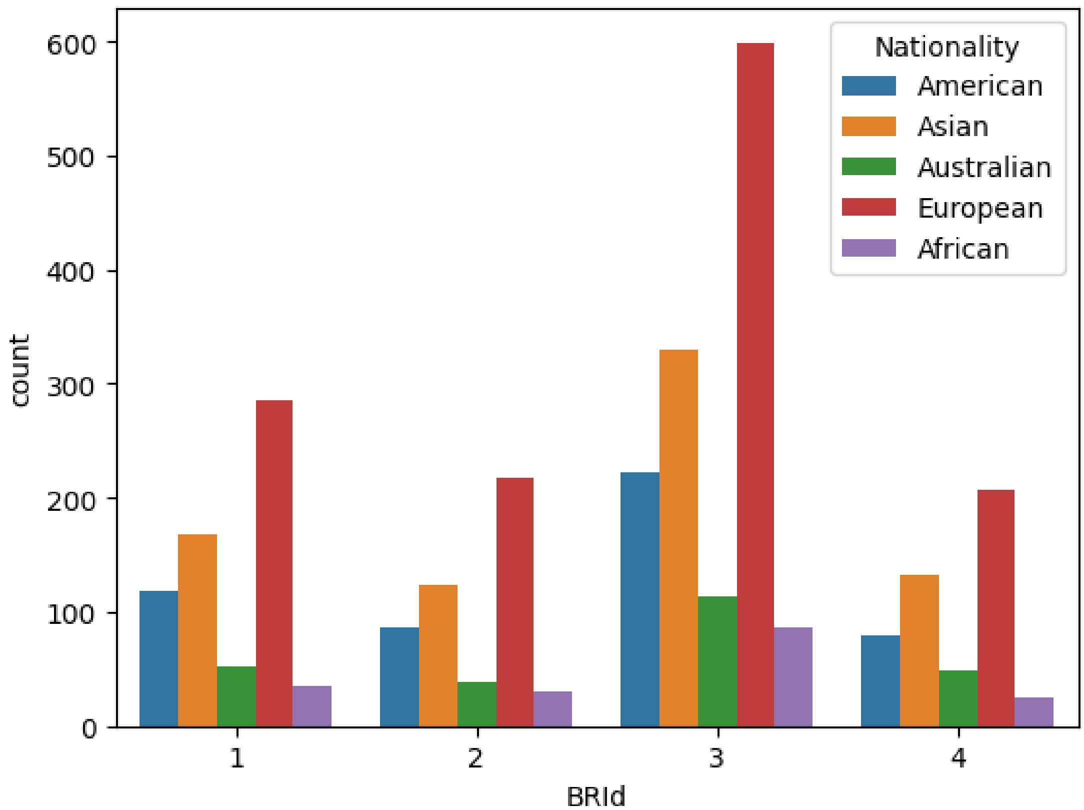

800
700
600
500
400
300
200
100
0
count
0
1
2
3
Properties Owned

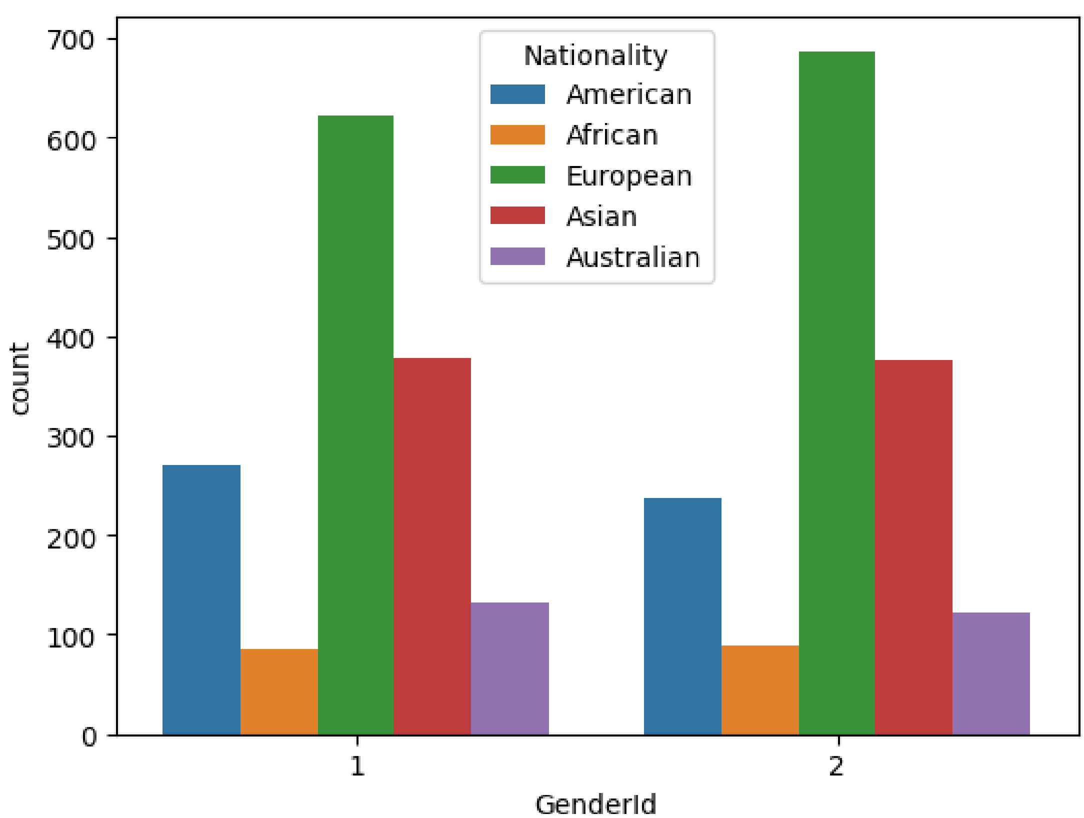

1200
1000
800
600
400
200
0
count
1
2
3
4
5
Risk Weighting

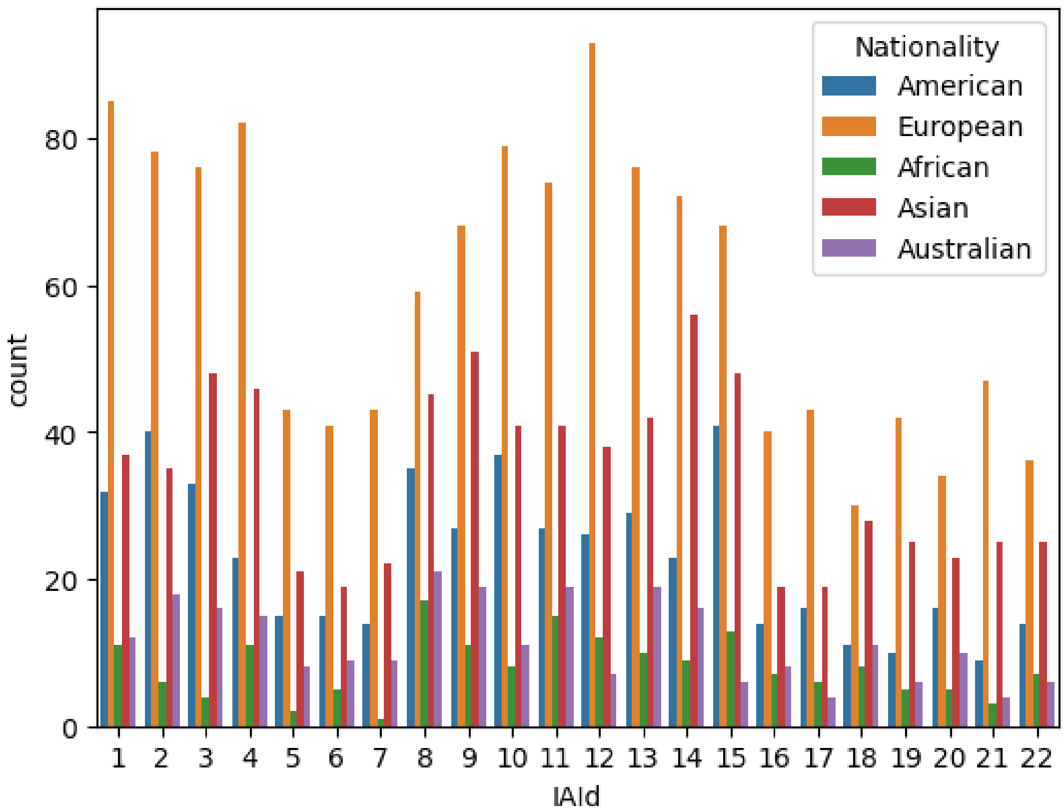


Bivariate Analysis

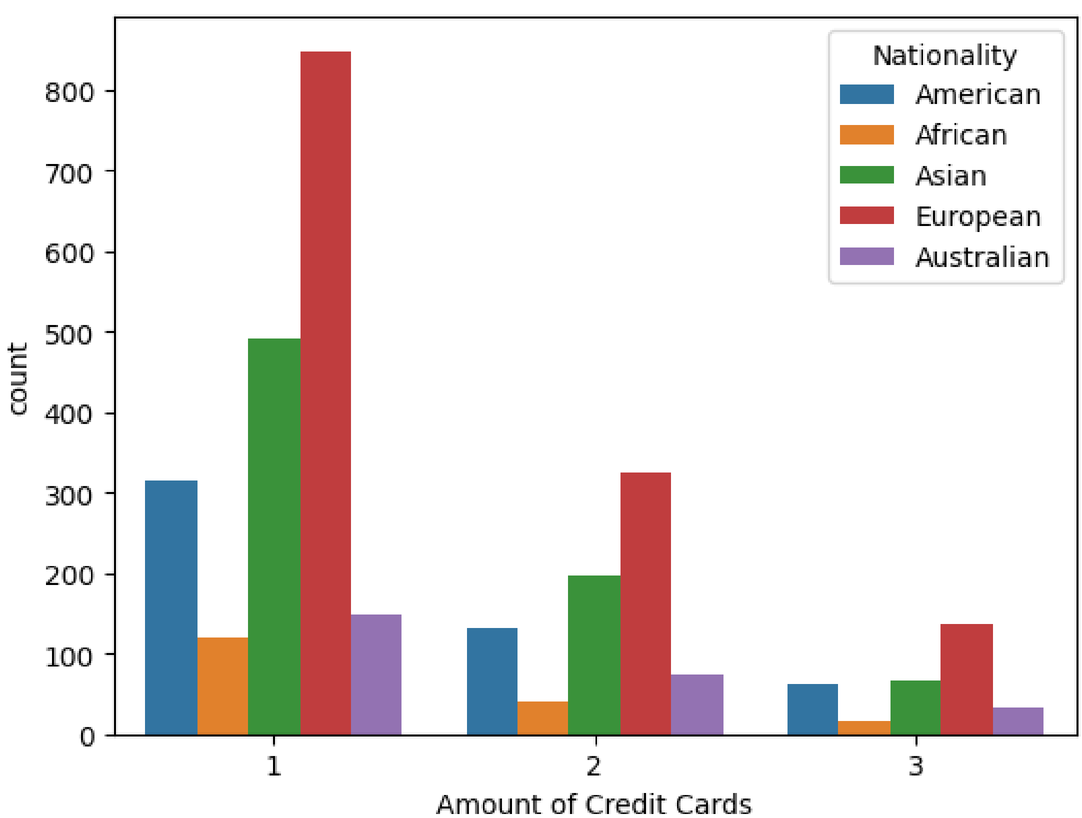
```
for i, predictor in enumerate(df[["BRId", "GenderId", "IAId", "Amount of Credit Cards", "Nationality", "Occupation", "Fee Structure", "Loyalty Classification", "Properties Owned", "Risk Weighting", "Income Band"]].columns):
    plt.figure(i)
    sns.countplot(data=df, x=predictor, hue='Nationality')
```


Nationality
American
Asian
Australian
European
African
count
600
500
400
300
200
100
0
1
2
3
4
BRId

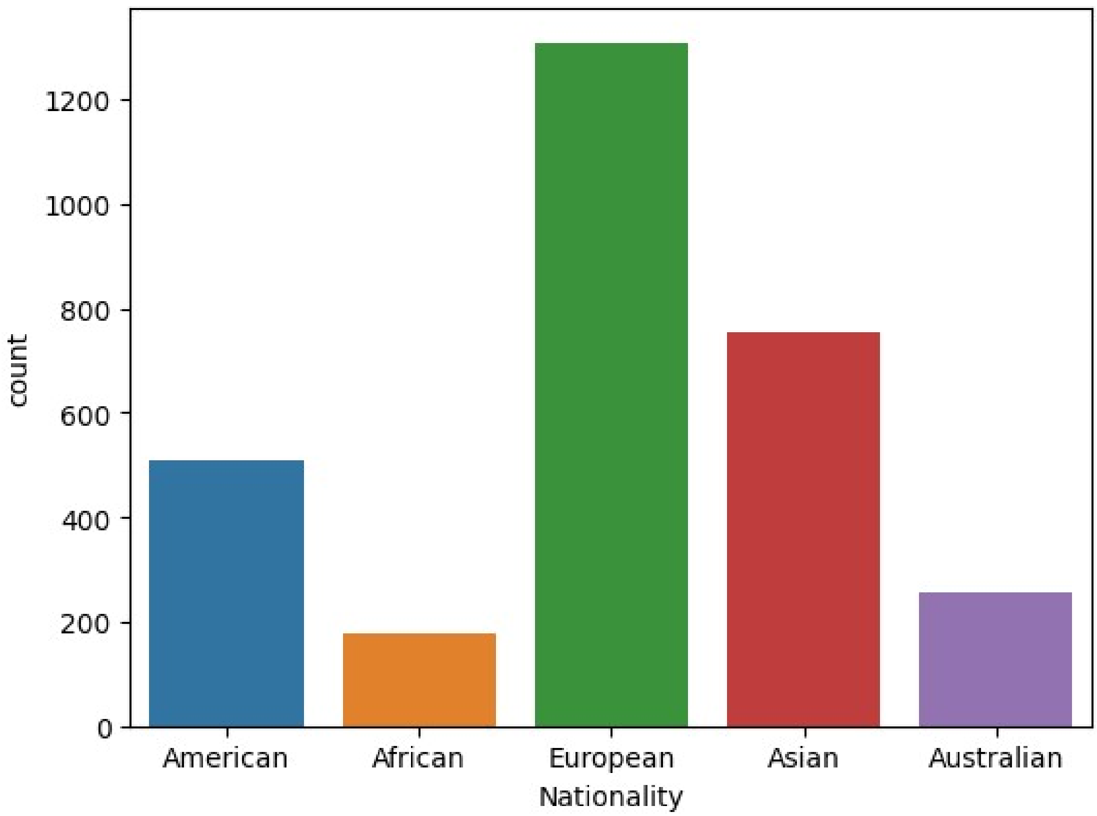

Nationality
American
African
European
Asian
Australian
count
700
600
500
400
300
200
100
0
1
2
GenderId



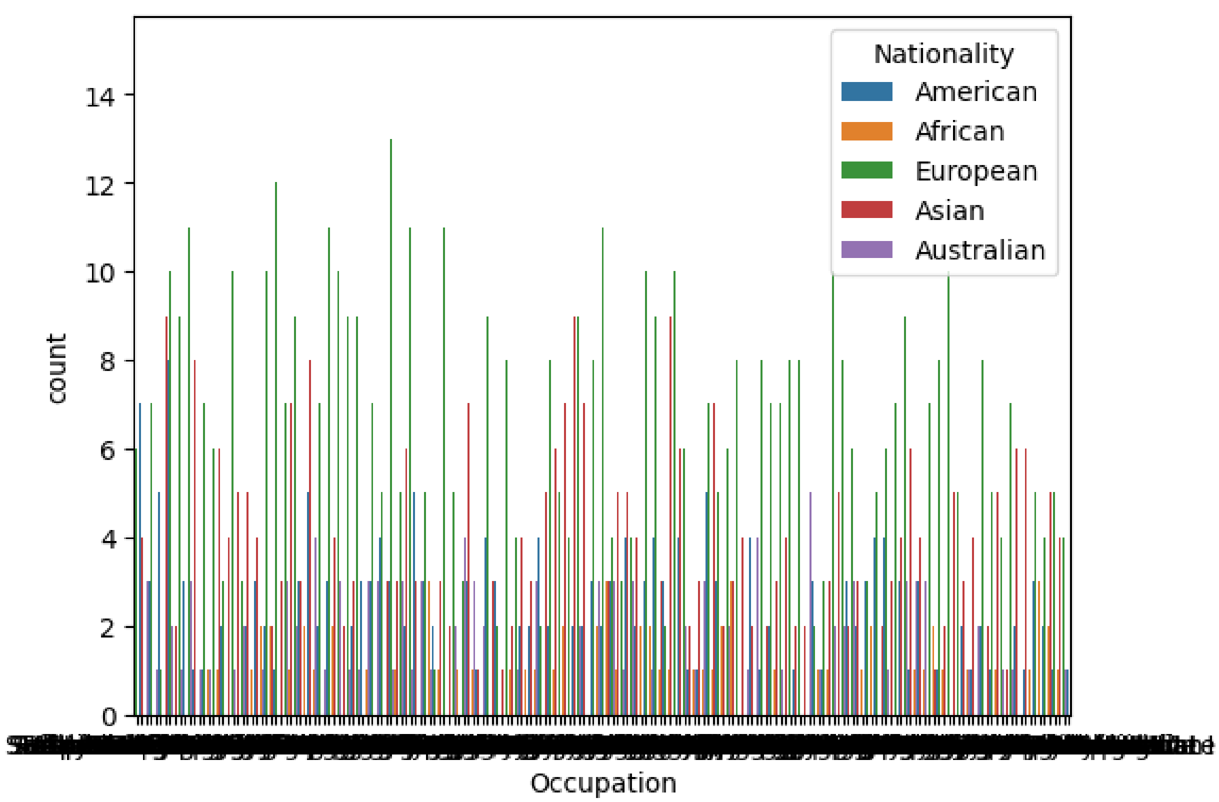
Nationality
American
European
African
Asian
Australian
count
0
20
40
60
80
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
IAId



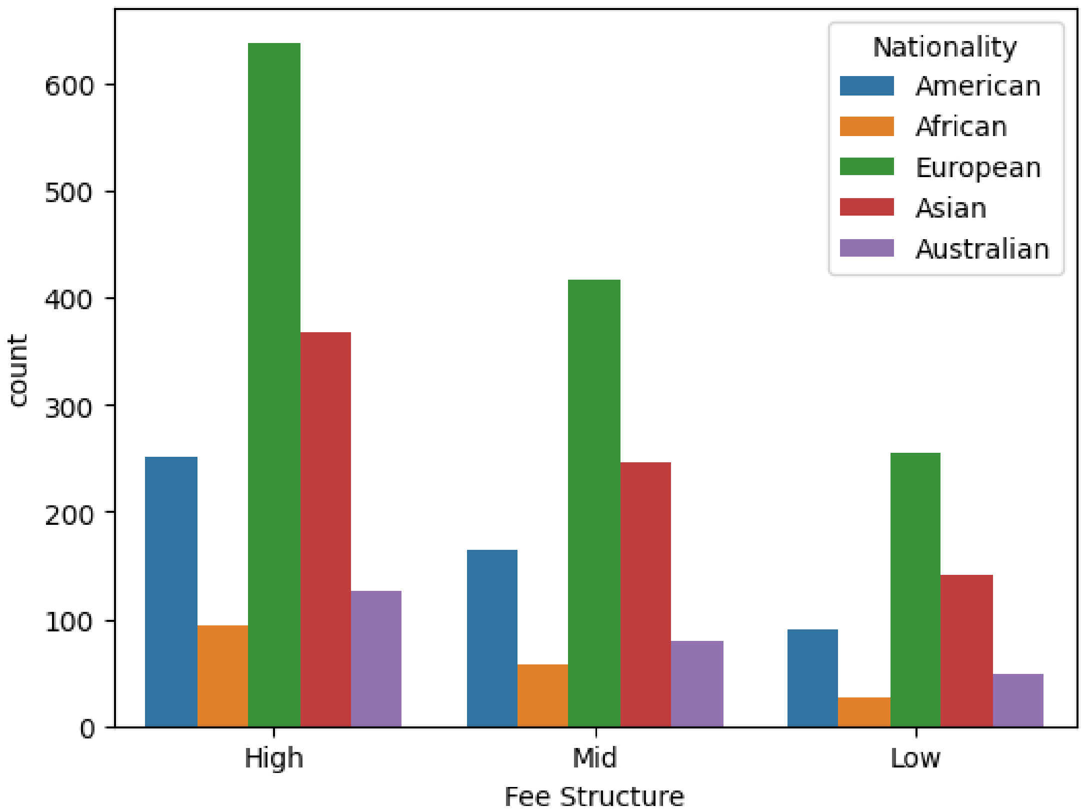
Nationality
American
African
Asian
European
Australian
count
0
100
200
300
400
500
600
700
800
1
2
3
Amount of Credit Cards

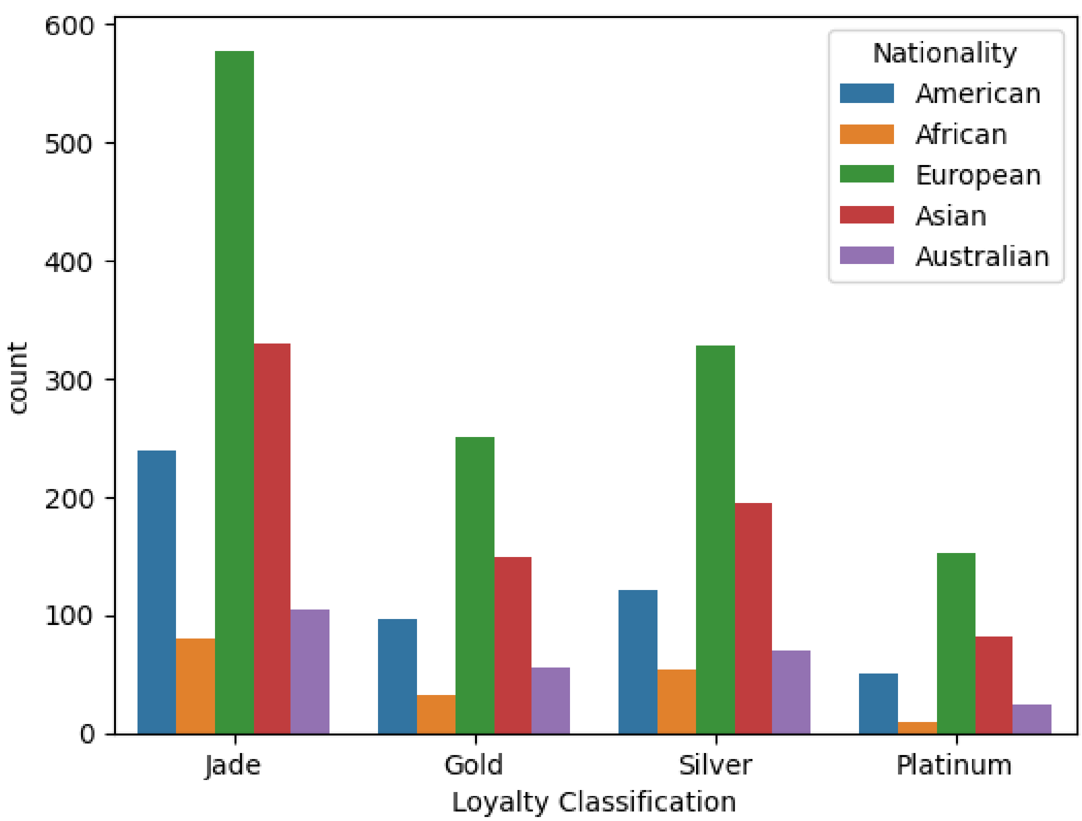


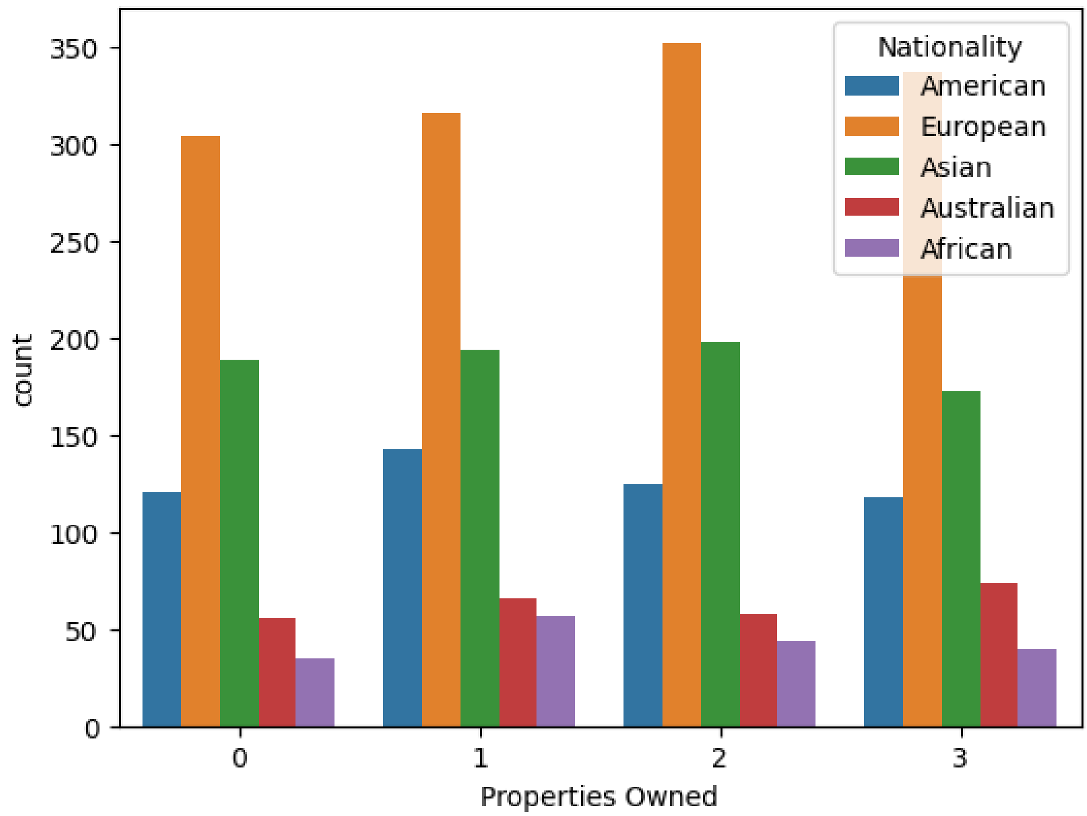
1200
1000
800
600
400
200
0
count
American
African
European
Asian
Australian
Nationality

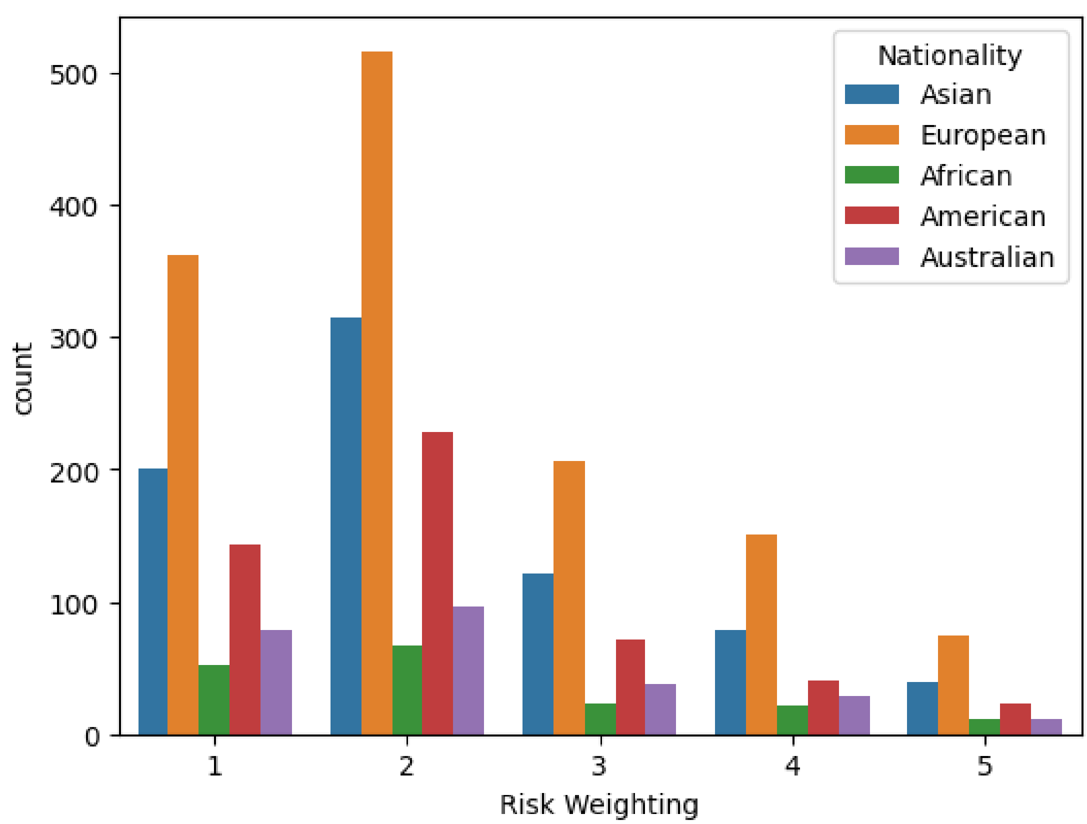


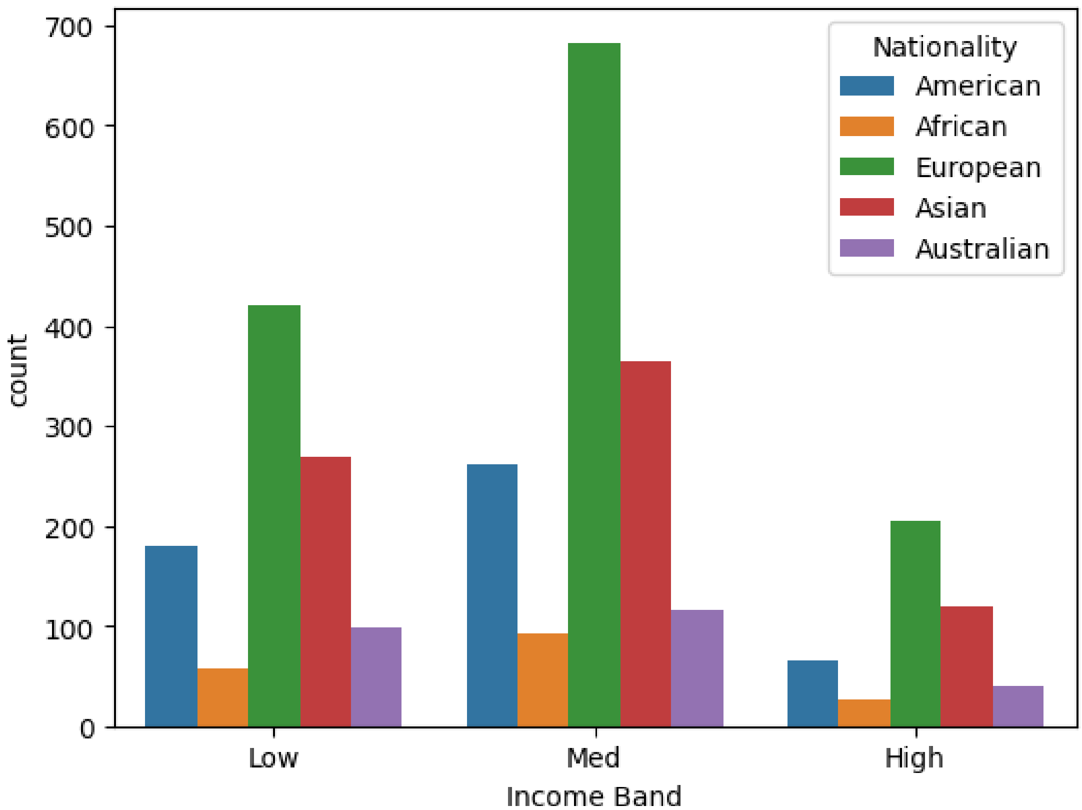
Nationality
American
African
European
Asian
Australian
14
12
10
8
6
4
2
0
count
Occupation


Nationality
American
African
European
Asian
Australian
600
500
400
300
200
100
0
count
High
Mid
Low
Fee Structure

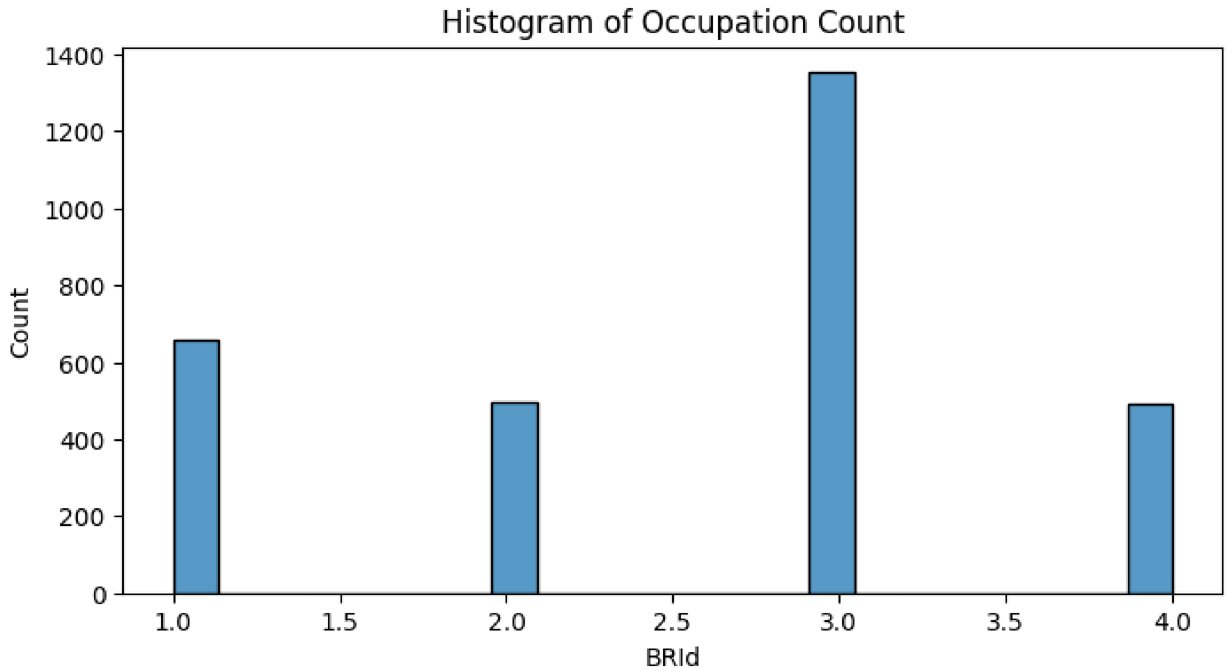

Nationality
American
African
European
Asian
Australian
600
500
400
300
200
100
0
count
Jade
Gold
Silver
Platinum
Loyalty Classification

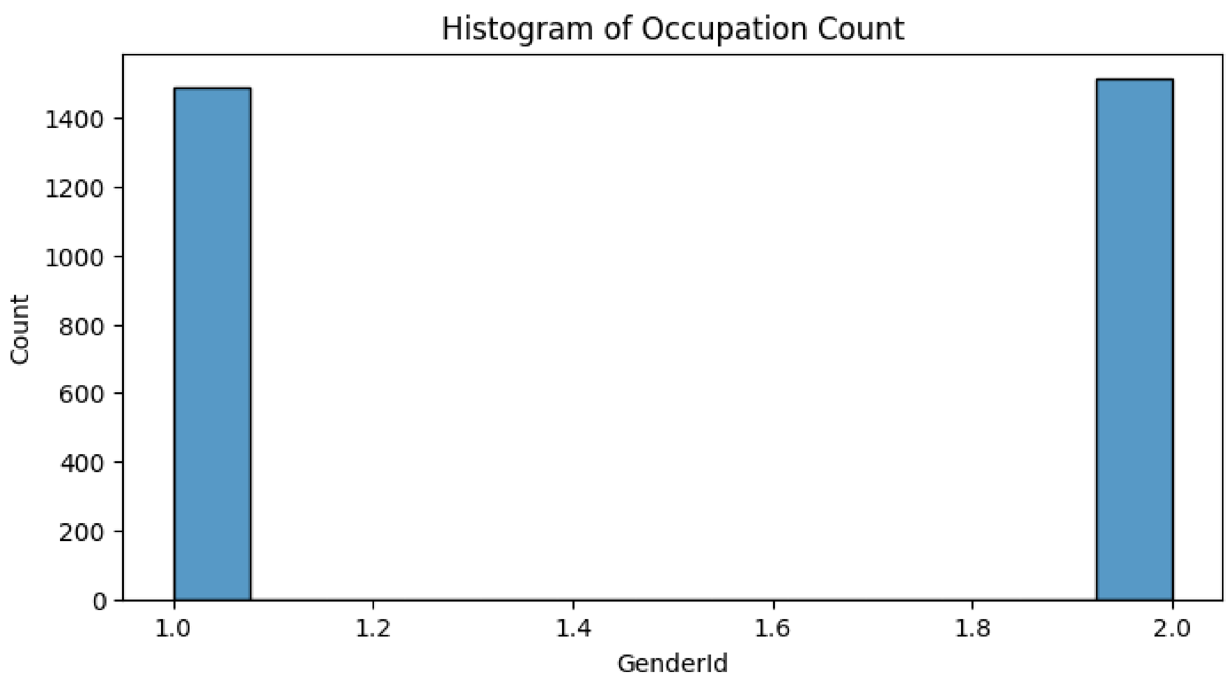

Nationality
American
European
Asian
Australian
African
count
350
300
250
200
150
100
50
0
0
1
2
3
Properties Owned

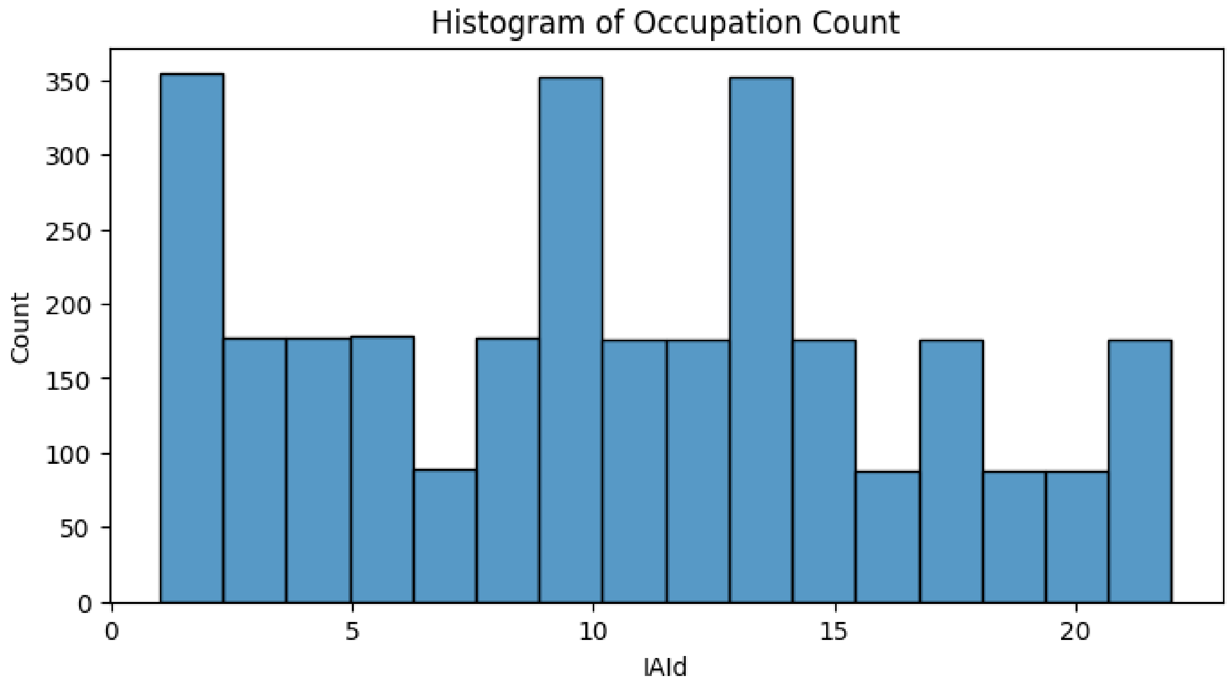

Nationality
Asian
European
African
American
Australian
count
500
400
300
200
100
0
1
2
3
4
5
Risk Weighting

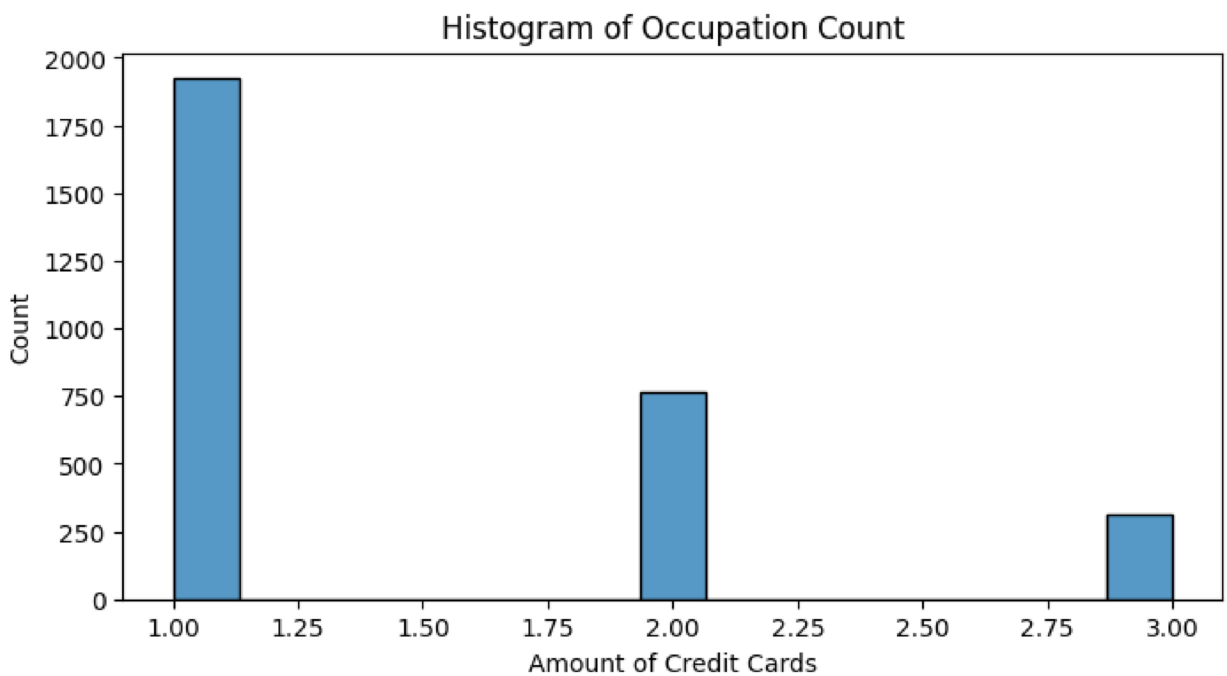


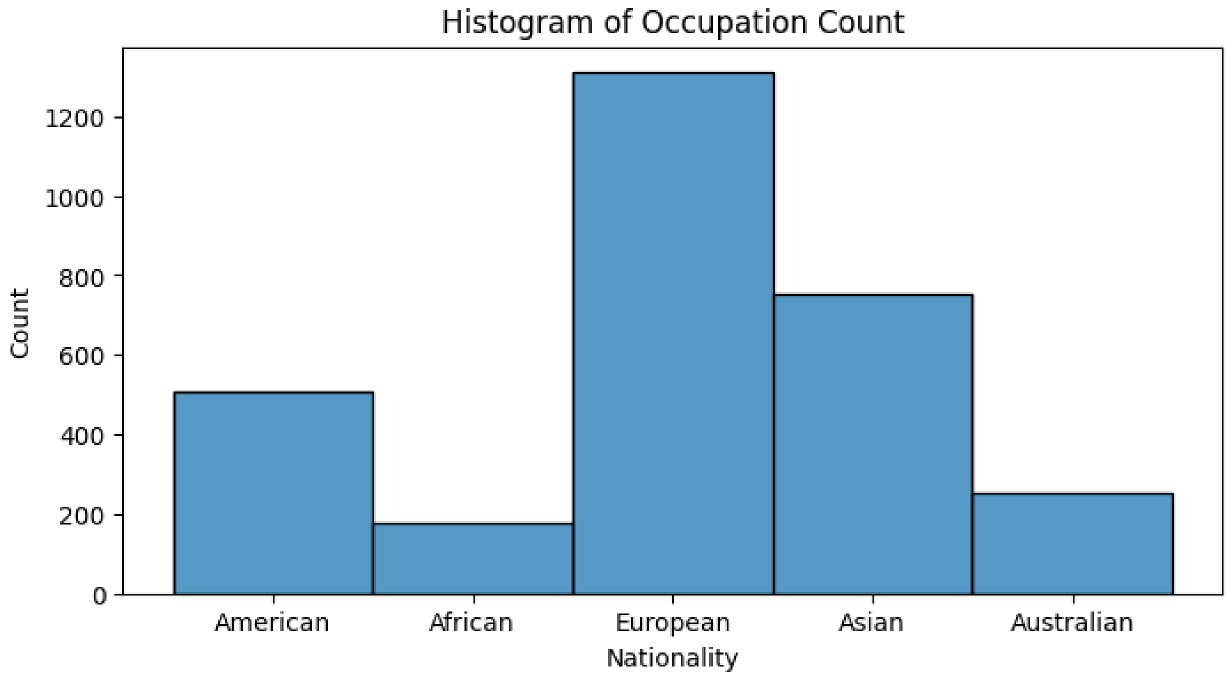

```
# HIstplot of value counts for different Occupation

for col in categorical_cols:
  if col == "Occupation":
    continue
  plt.figure(figsize=(8,4))
  sns.histplot(df[col])
  plt.title('Histogram of Occupation Count')
  plt.xlabel(col)
  plt.ylabel("Count")
  plt.show()
```


Histogram of Occupation Count
Count
BRId
0
200
400
600
800
1000
1200
1400
1.0
1.5
2.0
2.5
3.0
3.5
4.0

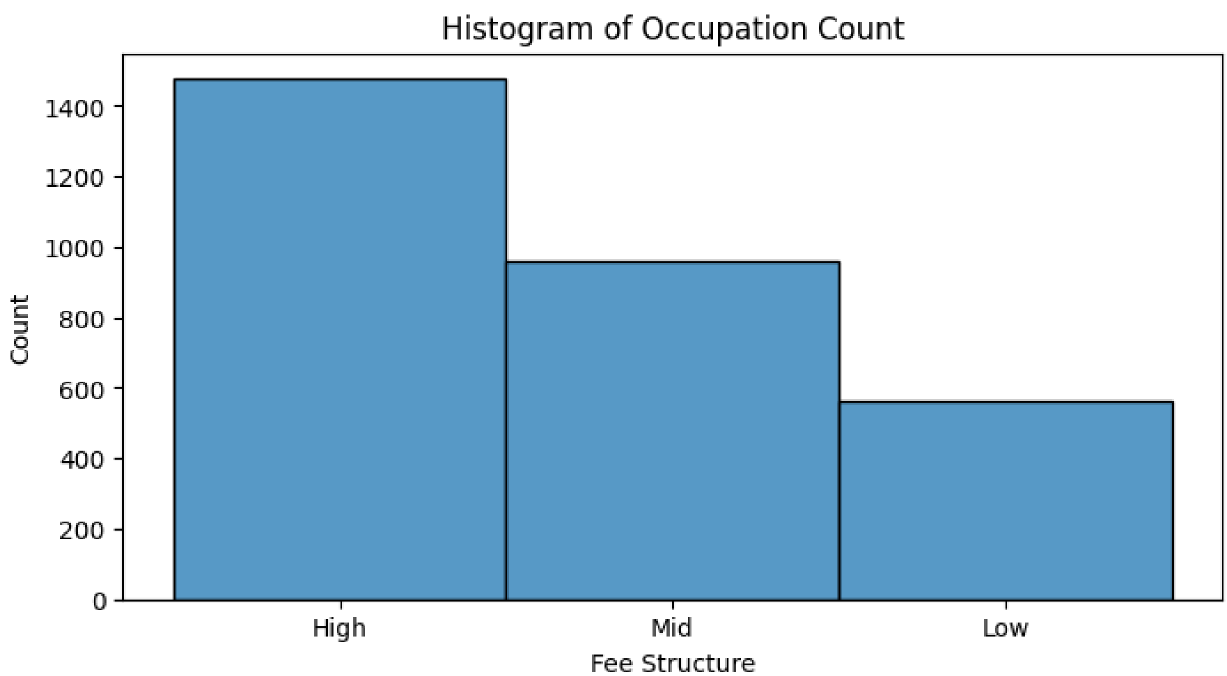

Histogram of Occupation Count
Count
GenderId
0
200
400
600
800
1000
1200
1400
1.0
1.2
1.4
1.6
1.8
2.0


Histogram of Occupation Count
Count
IAId
0
50
100
150
200
250
300
350
0
5
10
15
20

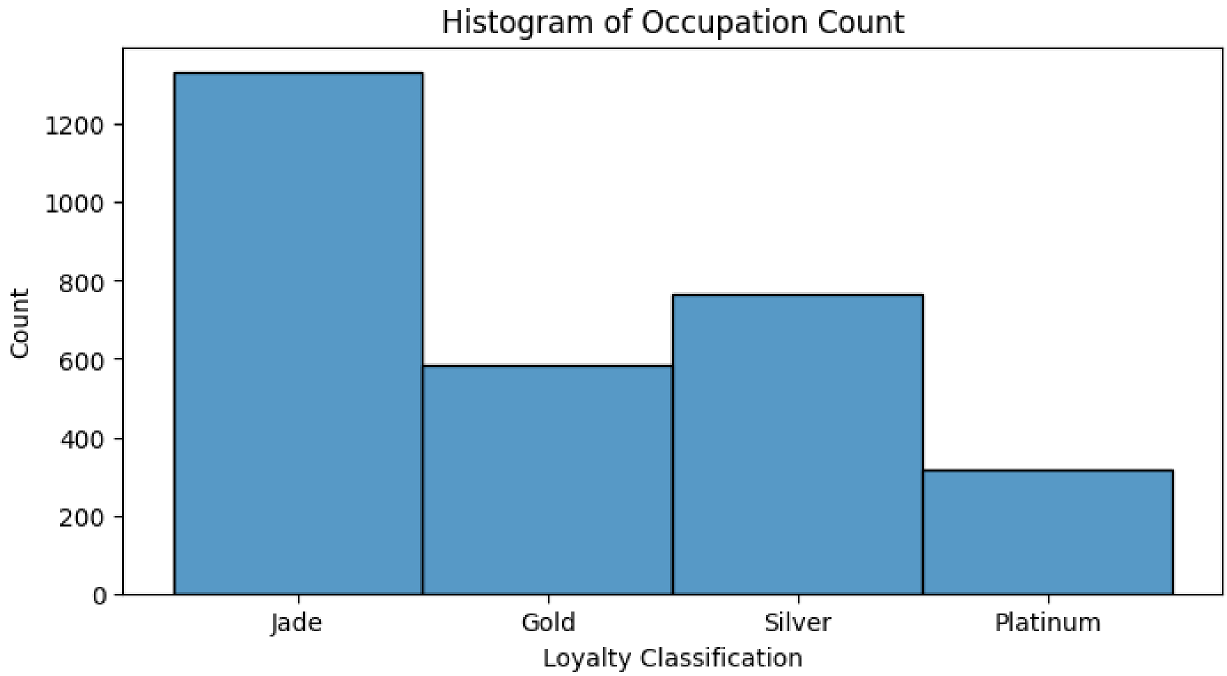

Histogram of Occupation Count
Count
Amount of Credit Cards
0
250
500
750
1000
1250
1500
1750
2000
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00

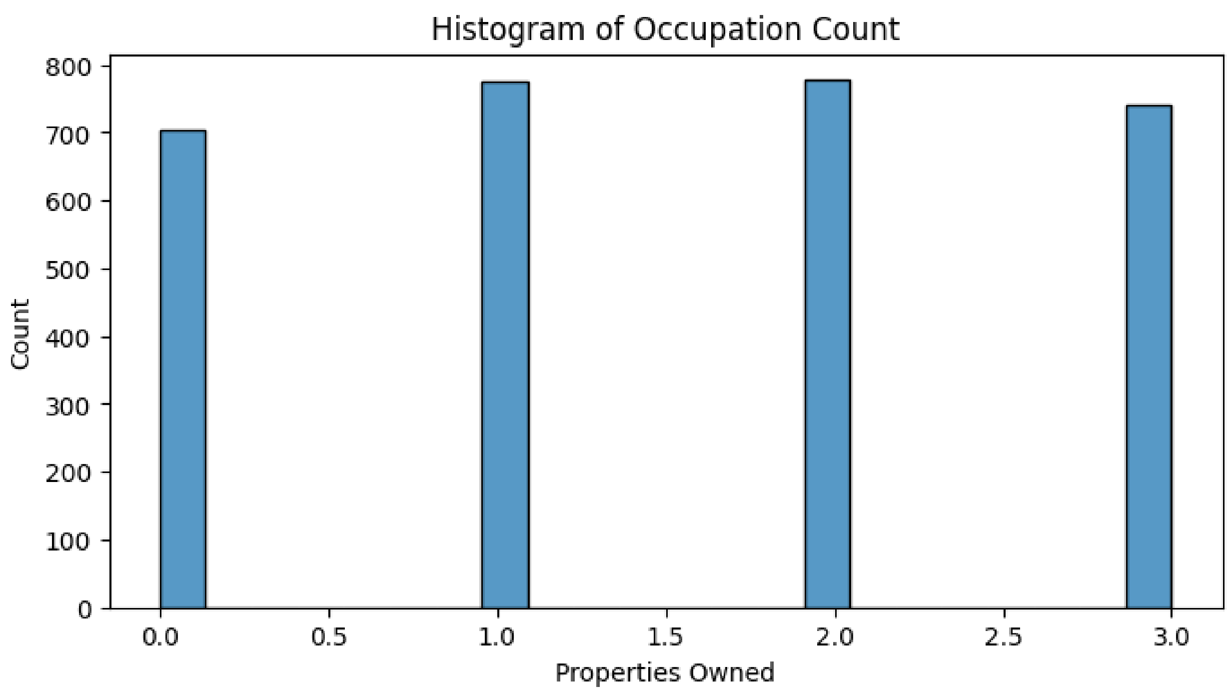


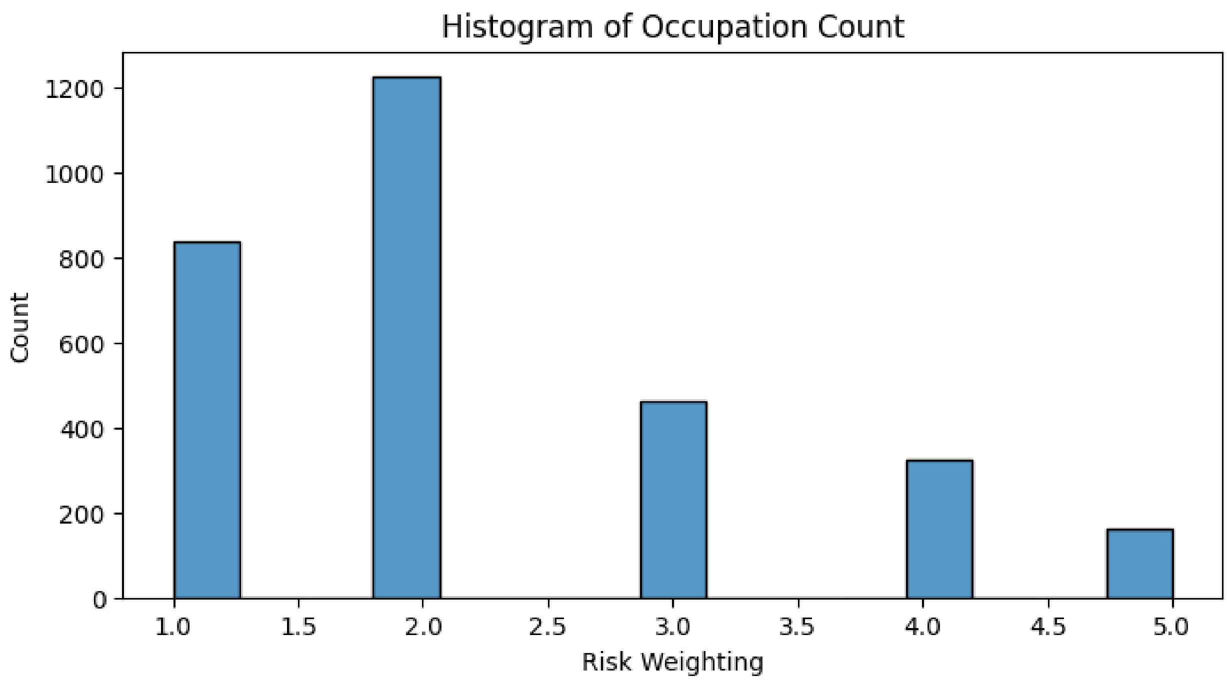
Histogram of Occupation Count
Count
0
200
400
600
800
1000
1200
American
African
European
Asian
Australian
Nationality

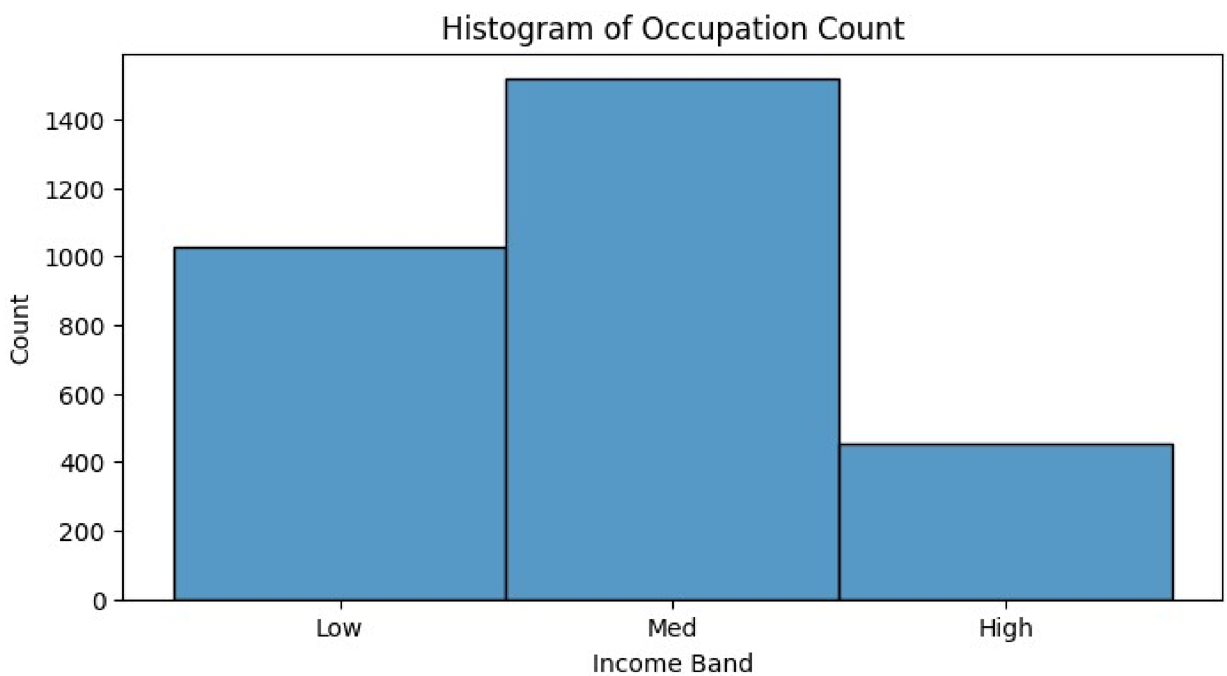


Histogram of Occupation Count
Count
0
200
400
600
800
1000
1200
1400
High
Mid
Low
Fee Structure


Histogram of Occupation Count
Count
1200
1000
800
600
400
200
0
Jade
Gold
Silver
Platinum
Loyalty Classification


Histogram of Occupation Count
Count
800
700
600
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Properties Owned

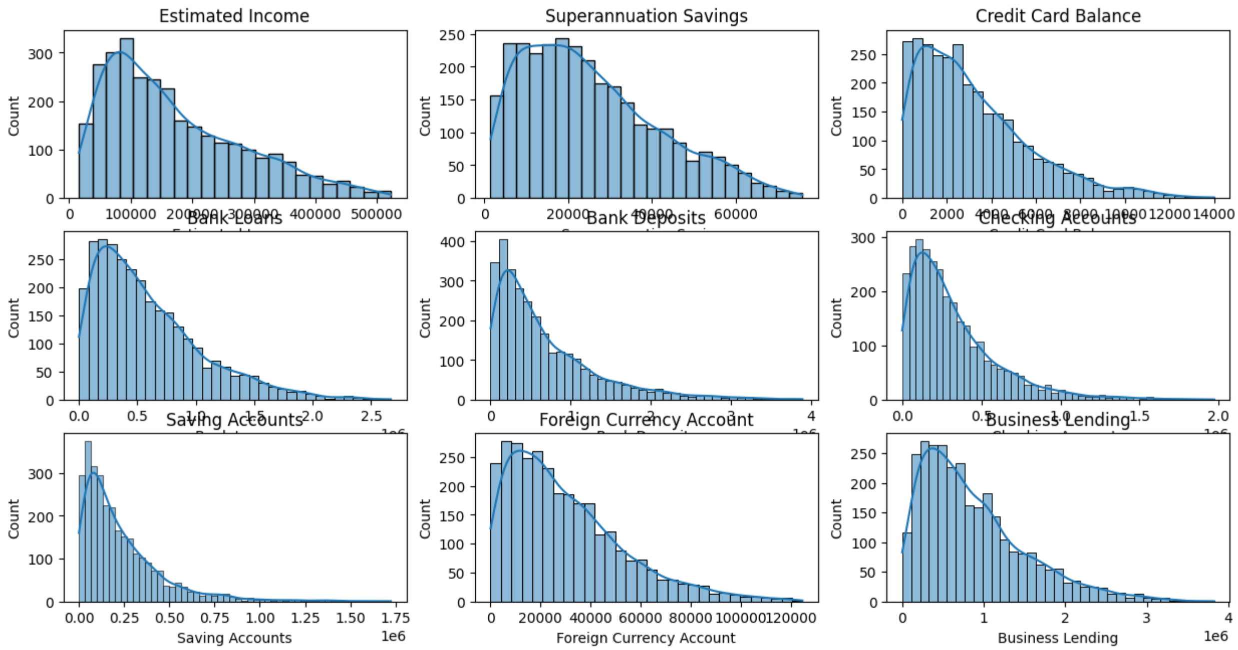




Numerical Analysis

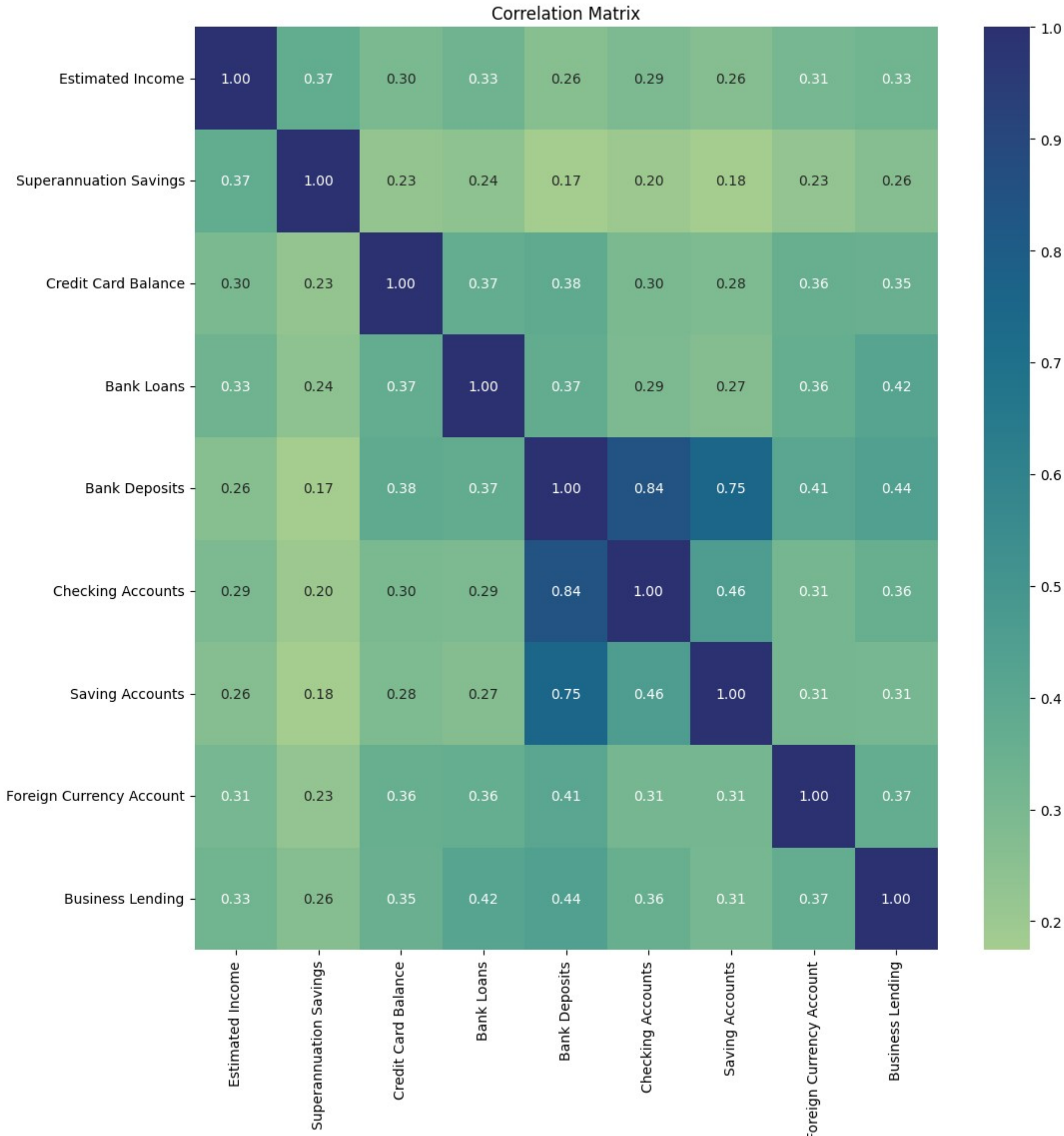
```
numerical_cols = ['Estimated Income', 'Superannuation Savings',
'Credit Card Balance', 'Bank Loans', 'Bank Deposits', 'Checking
Accounts', 'Saving Accounts', 'Foreign Currency Account', 'Business
Lending']
```

```
# Univariate analysis and visualization
plt.figure(figsize=(15,10))
for i,col in enumerate(numerical_cols):
  plt.subplot(4,3,i+1)
  sns.histplot(df[col],kde=True)
  plt.title(col)
plt.show()
```



Heatmaps

```
numerical_cols = ['Estimated Income', 'Superannuation Savings',
'Credit Card Balance', 'Bank Loans', 'Bank Deposits', 'Checking
Accounts', 'Saving Accounts', 'Foreign Currency Account', 'Business
Lending']

correlation_matrix = df[numerical_cols].corr()

plt.figure(figsize=(12,12))
sns.heatmap(correlation_matrix, annot=True, cmap='crest', fmt=".2f")
plt.title("Correlation Matrix")
plt.show()
```



Insights of EDA:

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