Banking

May 18, 2025

[13]:
```python
import pyodbc
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

[2]:
```python
conn = pyodbc.connect(
    'DRIVER={ODBC Driver 17 for SQL Server};'
    'SERVER=LAPTOP-1K5PSON5\SQLEXPRESS;'
    'DATABASE=Banking;'
    'Trusted_Connection=yes;'
)
```

[5]:
```python
df = pd.read_sql_query("SELECT * FROM Banking", conn)
df.head()
```

```
c:\users\singh\appdata\local\programs\python\python39\lib\site-
packages\pandas\io\sql.py:761: UserWarning: pandas only support SQLAlchemy
connectable(engine/connection) ordatabase string URI or sqlite3 DBAPI2
connectionother DBAPI2 objects are not tested, please consider using SQLAlchemy
  warnings.warn(
```

[5]:
```
  Client_ID             Name  Age  Location_ID Joined_Bank   Banking_Contact  \
0  IND81288    Raymond Mills   24        34324  2019-05-06    Anthony Torres
1  IND65833    Julia Spencer   23        42205  2001-12-10  Jonathan Hawkins
2  IND47499   Stephen Murray   27         7314  2010-01-25     Anthony Berry
3  IND72498   Virginia Garza   40        34594  2019-03-28        Steve Diaz
4  IND60181  Melissa Sanders   46        41269  2012-07-20        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  …

   Bank_Deposits  Checking_Accounts  Saving_Accounts  \
0   1.485829e+06       6.036179e+05    607332.437500
```

```
1   6.414828e+05       2.295214e+05    344635.156250
2   1.033402e+06       6.526747e+05    203054.343750
3   1.048158e+06       1.048158e+06    234685.015625
4   4.877825e+05       4.466442e+05    128351.453125

   Foreign_Currency_Account  Business_Lending  Properties_Owned  \
0              12249.959961      1.134475e+06                 1
1              61162.308594      2.000526e+06                 1
2              79071.781250      5.481376e+05                 1
3              57513.648438      1.148402e+06                 0
4              30012.140625      1.674412e+06                 0

   Risk_Weighting  BRId  GenderId  IAId
0               2     1         1     1
1               3     2         1     2
2               3     3         2     3
3               4     4         1     4
4               3     1         2     5

[5 rows x 25 columns]
```

[6]:
```python
print(df)
```

```
     Client_ID             Name  Age  Location_ID Joined_Bank  \
0     IND81288     Raymond Mills   24        34324  2019-05-06
1     IND65833     Julia Spencer   23        42205  2001-12-10
2     IND47499    Stephen Murray   27         7314  2010-01-25
3     IND72498    Virginia Garza   40        34594  2019-03-28
4     IND60181   Melissa Sanders   46        41269  2012-07-20
…        …               …   …          …          …
2995  IND66827         Earl Hall   82         8760  2014-10-09
2996  IND40556  Billy Williamson   44        32837  2009-02-05
2997  IND72414      Victor Black   70        36088  2009-12-29
2998  IND46652       Andrew Ford   56        24871  2006-02-13
2999  IND40216        Amy Nguyen   79        38518  2005-12-08

          Banking_Contact Nationality                     Occupation  \
0          Anthony Torres    American           Safety Technician IV
1        Jonathan Hawkins     African            Software Consultant
2           Anthony Berry    European             Help Desk Operator
3              Steve Diaz    American                   Geologist II
4              Shawn Long    American            Assistant Professor
…                     …        …                          …
2995       Joshua Bennett    American       Accounting Assistant III
2996          Dennis Ruiz    European                      Paralegal
2997          Joshua Ryan    American                Statistician IV
2998  Nicholas Cunningham    European  Human Resources Assistant III
2999           Joe Hanson    American              Biostatistician III
```

```
     Fee_Structure Loyalty_Classification  …  Bank_Deposits  \
0             High                   Jade  …   1.485829e+06
1             High                   Jade  …   6.414828e+05
2             High                   Gold  …   1.033402e+06
3              Mid                 Silver  …   1.048158e+06
4              Mid               Platinum  …   4.877825e+05
…              …                    …   …            …
2995          High                   Gold  …   1.089957e+06
2996           Mid                   Gold  …   1.368913e+05
2997           Low                   Jade  …   2.148609e+05
2998           Mid                   Jade  …   7.426302e+05
2999          High                   Jade  …   6.561766e+04

      Checking_Accounts  Saving_Accounts  Foreign_Currency_Account  \
0          6.036179e+05    607332.437500              12249.959961
1          2.295214e+05    344635.156250              61162.308594
2          6.526747e+05    203054.343750              79071.781250
3          1.048158e+06    234685.015625              57513.648438
4          4.466442e+05    128351.453125              30012.140625
…                   …              …                       …
2995       5.328679e+05    657849.625000              12947.309570
2996       5.658174e+04     93195.609375              23205.689453
2997       1.587261e+05     35539.148438              30291.810547
2998       4.046382e+05     56411.328125               6413.140137
2999       7.776908e+04     32371.380859               8992.360352

      Business_Lending  Properties_Owned  Risk_Weighting  BRId  GenderId  IAId
0         1.134475e+06                 1               2     1         1     1
1         2.000526e+06                 1               3     2         1     2
2         5.481376e+05                 1               3     3         2     3
3         1.148402e+06                 0               4     4         1     4
4         1.674412e+06                 0               3     1         2     5
…                  …              …              …   …       …   …
2995      1.238860e+06                 1               3     3         2     4
2996      2.771711e+05                 1               2     3         2     5
2997      5.029472e+05                 2               2     3         2     6
2998      1.538369e+06                 3               1     3         2     7
2999      3.294126e+05                 1               1     3         2     8

[3000 rows x 25 columns]
```

[7]:
```
df.head()
```

[7]:
```
  Client_ID            Name  Age  Location_ID Joined_Bank   Banking_Contact  \
0  IND81288   Raymond Mills   24        34324  2019-05-06    Anthony Torres
1  IND65833   Julia Spencer   23        42205  2001-12-10  Jonathan Hawkins
```

```
2  IND47499   Stephen Murray   27         7314  2010-01-25     Anthony Berry
3  IND72498   Virginia Garza   40        34594  2019-03-28        Steve Diaz
4  IND60181  Melissa Sanders   46        41269  2012-07-20        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  …

   Bank_Deposits  Checking_Accounts  Saving_Accounts  \
0   1.485829e+06       6.036179e+05    607332.437500
1   6.414828e+05       2.295214e+05    344635.156250
2   1.033402e+06       6.526747e+05    203054.343750
3   1.048158e+06       1.048158e+06    234685.015625
4   4.877825e+05       4.466442e+05    128351.453125

   Foreign_Currency_Account  Business_Lending  Properties_Owned  \
0              12249.959961      1.134475e+06                 1
1              61162.308594      2.000526e+06                 1
2              79071.781250      5.481376e+05                 1
3              57513.648438      1.148402e+06                 0
4              30012.140625      1.674412e+06                 0

   Risk_Weighting  BRId  GenderId  IAId
0               2     1         1     1
1               3     2         1     2
2               3     3         2     3
3               4     4         1     4
4               3     1         2     5

[5 rows x 25 columns]
```

[8]:
```
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
```

0.0.1 Descriptive information of the data

[9]:
```
df.describe()
```

[9]:
```
               Age   Location_ID  Estimated_Income  Superannuation_Savings  \
count  3000.000000   3000.000000       3000.000000             3000.000000
mean     51.039667  21563.323000     171305.034184            25531.599685
std      19.854760  12462.273017     111935.808180            16259.950768
min      17.000000     12.000000      15919.480469             1482.030029
25%      34.000000  10803.500000      82906.597656            12513.774902
50%      51.000000  21129.500000     142313.476562            22357.355469
75%      69.000000  32054.500000     242290.300781            35464.741211
max      85.000000  43369.000000     522330.250000            75963.898438

       Amount_of_Credit_Cards  Credit_Card_Balance    Bank_Loans  \
count             3000.000000          3000.000000  3.000000e+03
mean                 1.463667          3176.206944  5.913862e+05
std                  0.676387          2497.094709  4.575570e+05
min                  1.000000             1.170000  0.000000e+00
25%                  1.000000          1236.630005  2.396281e+05
50%                  1.000000          2560.804932  4.797934e+05
75%                  2.000000          4522.632690  8.258130e+05
max                  3.000000         13991.990234  2.667557e+06

       Bank_Deposits  Checking_Accounts  Saving_Accounts  \
count   3.000000e+03       3.000000e+03     3.000000e+03
```

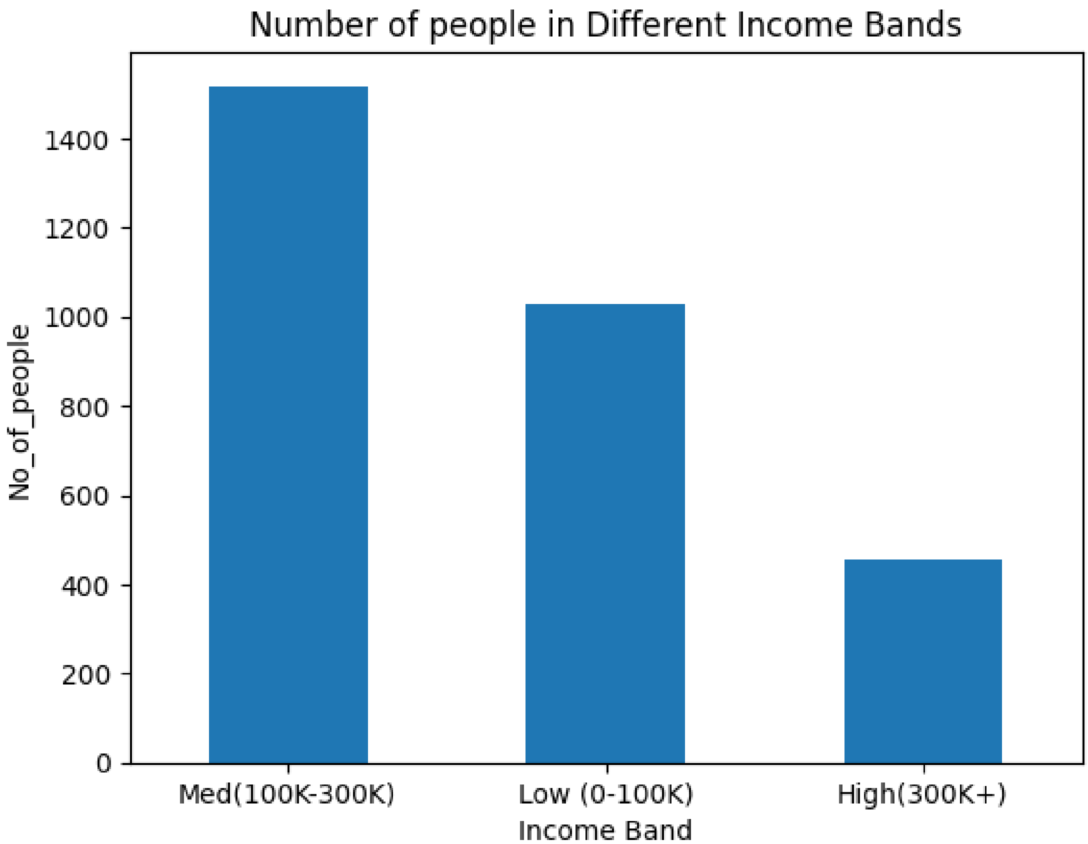
```
mean    6.715602e+05     3.210929e+05     2.329084e+05
std     6.457169e+05     2.820796e+05     2.300078e+05
min     0.000000e+00     0.000000e+00     0.000000e+00
25%     2.044004e+05     1.199475e+05     7.479440e+04
50%     4.633165e+05     2.428157e+05     1.640866e+05
75%     9.427546e+05     4.348749e+05     3.155750e+05
max     3.890598e+06     1.969923e+06     1.724118e+06

       Foreign_Currency_Account  Business_Lending  Properties_Owned  \
count               3000.000000      3.000000e+03       3000.000000
mean               29883.529998      8.667598e+05          1.518667
std                23109.924033      6.412303e+05          1.102145
min                   45.000000      0.000000e+00          0.000000
25%                11916.542236      3.748251e+05          1.000000
50%                24341.190430      7.113147e+05          2.000000
75%                41966.391602      1.185110e+06          2.000000
max               124704.867188      3.825962e+06          3.000000

       Risk_Weighting         BRId     GenderId         IAId
count     3000.000000  3000.000000  3000.000000  3000.000000
mean         2.249333     2.559333     1.504000    10.425333
std          1.131191     1.007713     0.500067     5.988242
min          1.000000     1.000000     1.000000     1.000000
25%          1.000000     2.000000     1.000000     5.000000
50%          2.000000     3.000000     2.000000    10.000000
75%          3.000000     3.000000     2.000000    15.000000
max          5.000000     4.000000     2.000000    22.000000
```

0.0.2 Adding a column to categorize estimated_income

[19]:
```
bins = [0,100000,300000, float('inf')]
labels = ['Low (0-100K)', 'Med(100K-300K)', 'High(300K+)']
df['Income_Band'] = pd.
↪cut(df['Estimated_Income'],bins=bins,labels=labels,right=False)
```

[21]:
```
df['Income_Band'].value_counts().plot(kind='bar',rot=0)
plt.title('Number of people in Different Income Bands')
plt.xlabel('Income Band')
plt.ylabel('No_of_people')
```

[21]:
```
Text(0, 0.5, 'No_of_people')
```


Number of people in Different Income Bands
No_of_people
0
200
400
600
800
1000
1200
1400
Med(100K-300K)
Low (0-100K)
High(300K+)
Income Band

0.0.3 Adding reference values for IAId and BRId

IAId	Investment Advisor
1	Victor Dean
2	Jeremy Porter
3	Ernest Knight
4	Eric Shaw
5	Kevin Kim
6	Victor Rogers
7	Eugene Cunningham
8	Joe Carroll
9	Steve Sanchez
10	Lawrence Sanchez
11	Peter Castillo
12	Victor Gutierrez
13	Daniel Carroll
14	Carl Anderson
15	Nicholas Ward
16	Fred Bryant
17	Ryan Taylor
18	Sean Vasquez
19	Nicholas Morrison
20	Jack Phillips
21	Juan Ramirez
22	Gregory Boyd

BRId	Banking Relationship
1	Retail
2	Institutional
3	Private Bank
4	Commercial

0.0.4 Examine the distribution of unique categories in categorical columns

[24]:
```
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:
```

```python
    print(f"Value counts for '{col}':")
    display(df[col].value_counts())
```

```
Value counts for 'BRId':

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

Value counts for 'GenderId':

2    1512
1    1488
Name: GenderId, dtype: int64

Value counts for '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: IAId, dtype: int64

Value counts for 'Amount_of_Credit_Cards':

1    1922
2     765
3     313
Name: Amount_of_Credit_Cards, dtype: int64

Value counts for 'Nationality':
```

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

```
Value counts for '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: Occupation, Length: 195, dtype: int64
```

```
Value counts for 'Fee_Structure':
```

```
High    1476
Mid      962
Low      562
Name: Fee_Structure, dtype: int64
```

```
Value counts for 'Loyalty_Classification':
```

```
Jade        1331
Silver       767
Gold         585
Platinum     317
Name: Loyalty_Classification, dtype: int64
```

```
Value counts for 'Properties_Owned':
```

```
2    777
1    776
3    742
0    705
Name: Properties_Owned, dtype: int64
```

```
Value counts for 'Risk_Weighting':
```

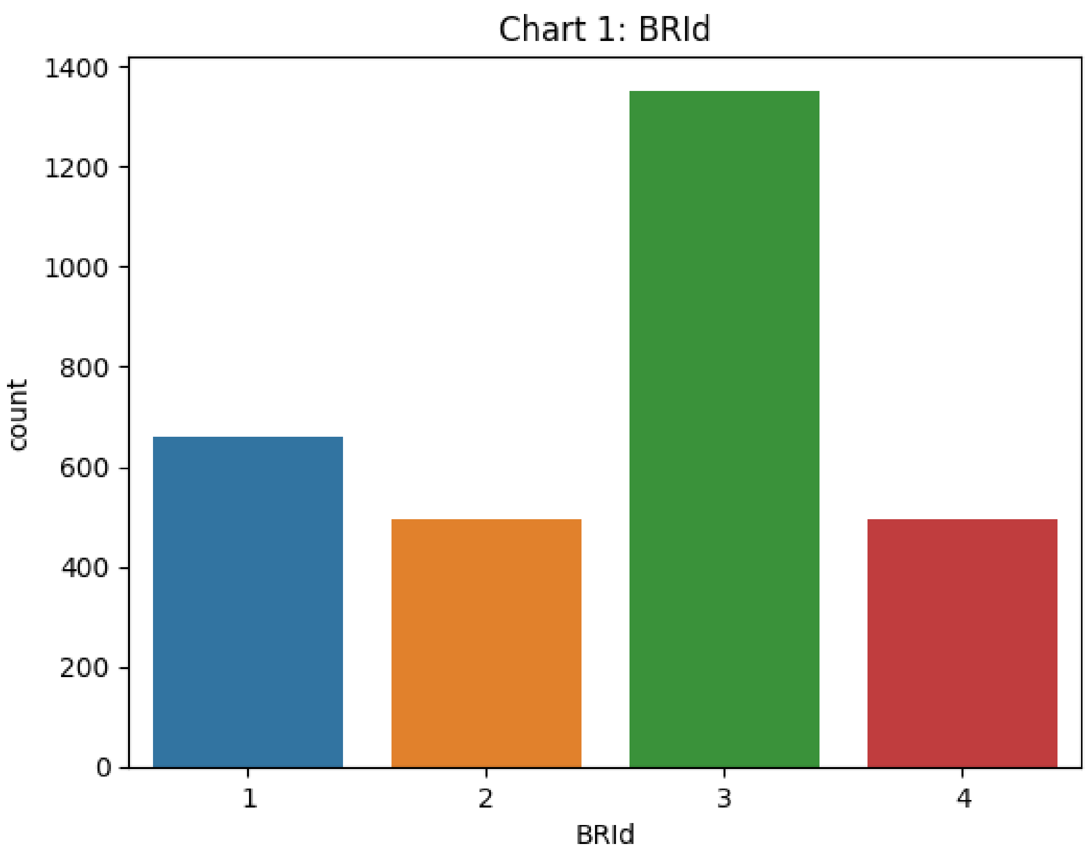
```
2    1222
1     836
3     460
4     322
5     160
Name: Risk_Weighting, dtype: int64
```

```
Value counts for 'Income_Band':

Med(100K-300K)    1517
Low (0-100K)      1027
High(300K+)        456
Name: Income_Band, dtype: int64
```

0.0.5 Univariate Analysis

[26]:

```
for i, col_name 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) #makes sure each plot starts fresh, using a separate figure
 ↪for each chart.
    sns.countplot(data=df, x=col_name) #x is the variable
    plt.title(f"Chart {i+1}: {col_name}") #Used this to name the plot according
 ↪to the index (i+1: used to skip 0)
```



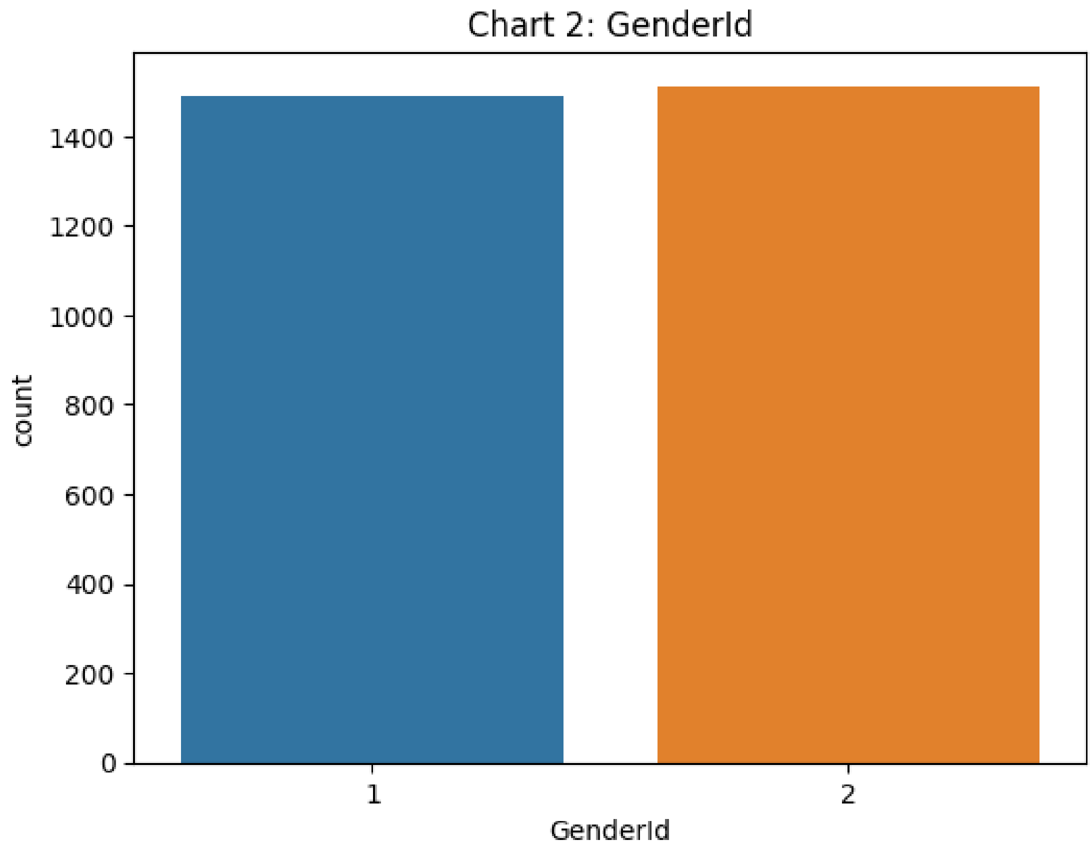


Chart 2: GenderId
count
0
200
400
600
800
1000
1200
1400
1
2
GenderId

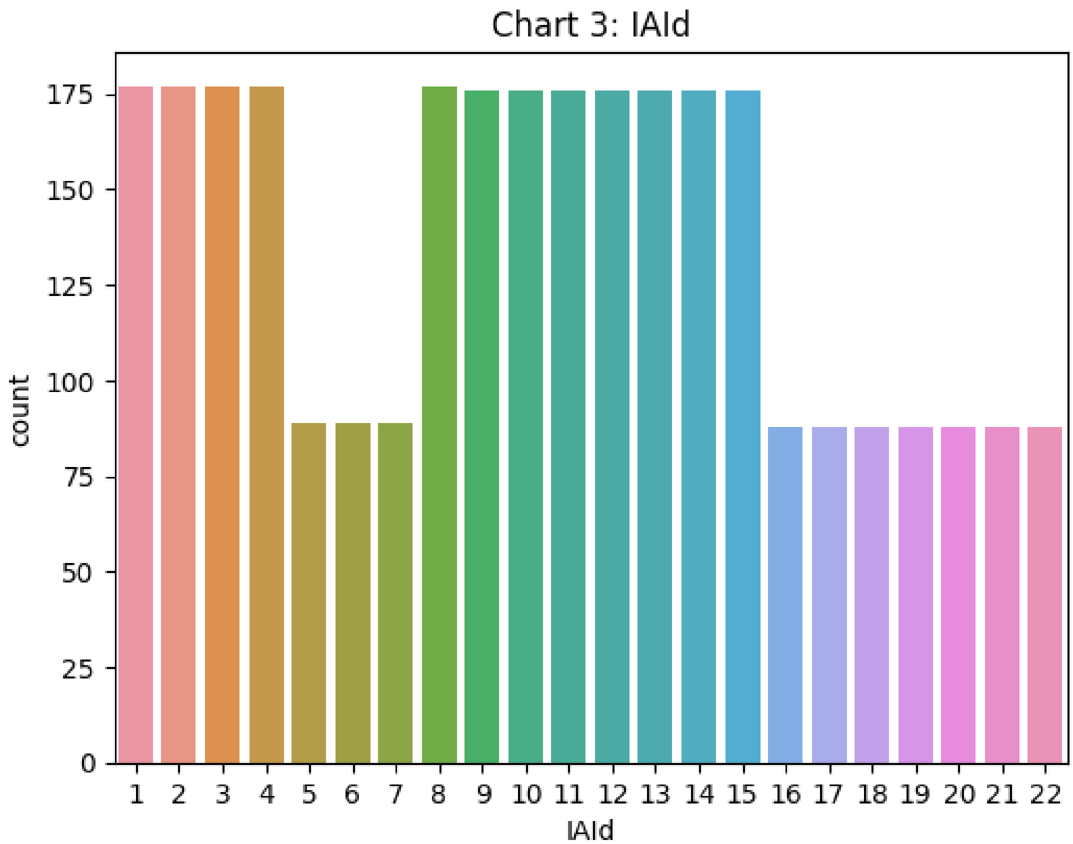


Chart 3: IAId
count
175
150
125
100
75
50
25
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
IAId

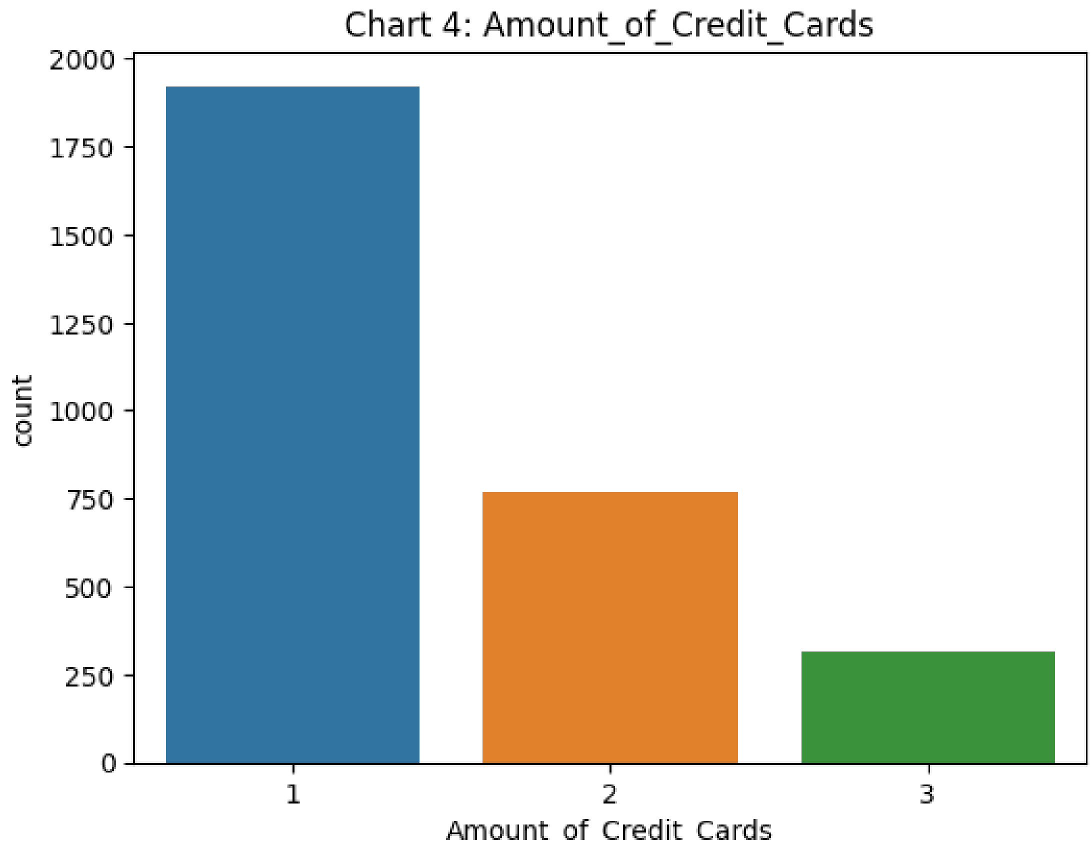


Chart 4: Amount_of_Credit_Cards
2000
1750
1500
1250
1000
750
500
250
0
count
1
2
3
Amount_of_Credit_Cards

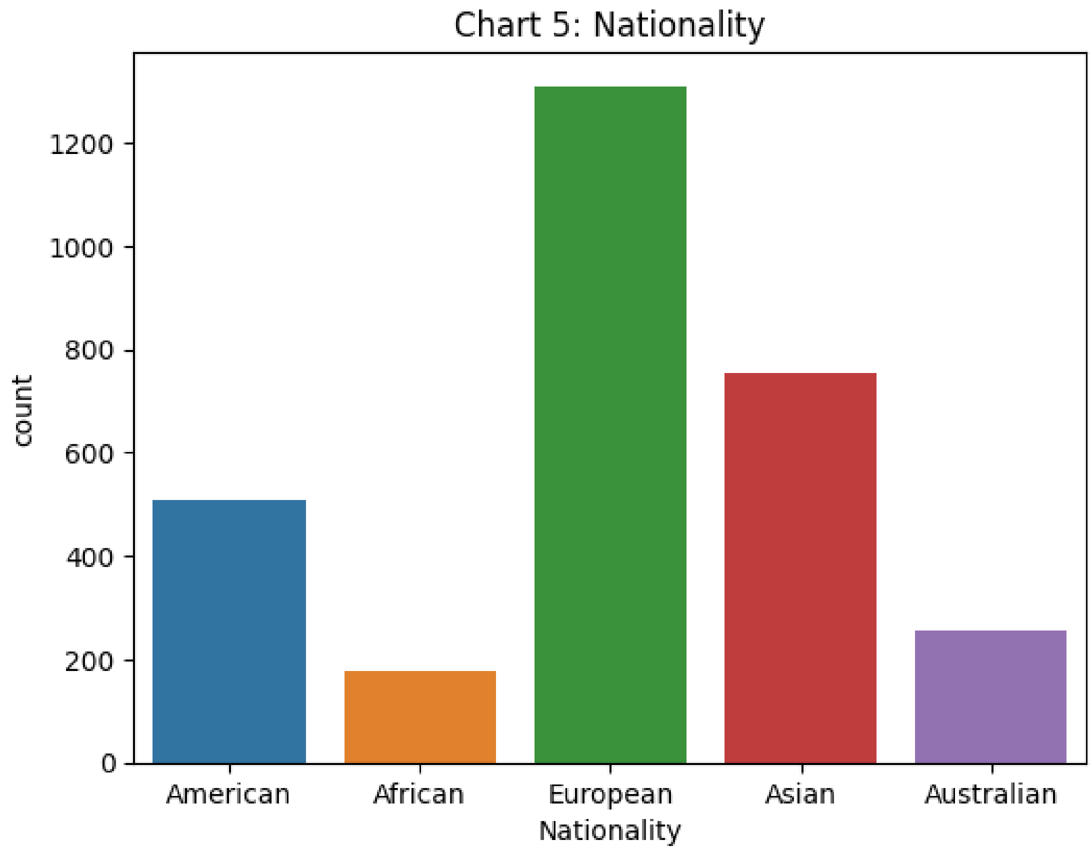


Chart 5: Nationality
count
1200
1000
800
600
400
200
0
American
African
European
Asian
Australian
Nationality

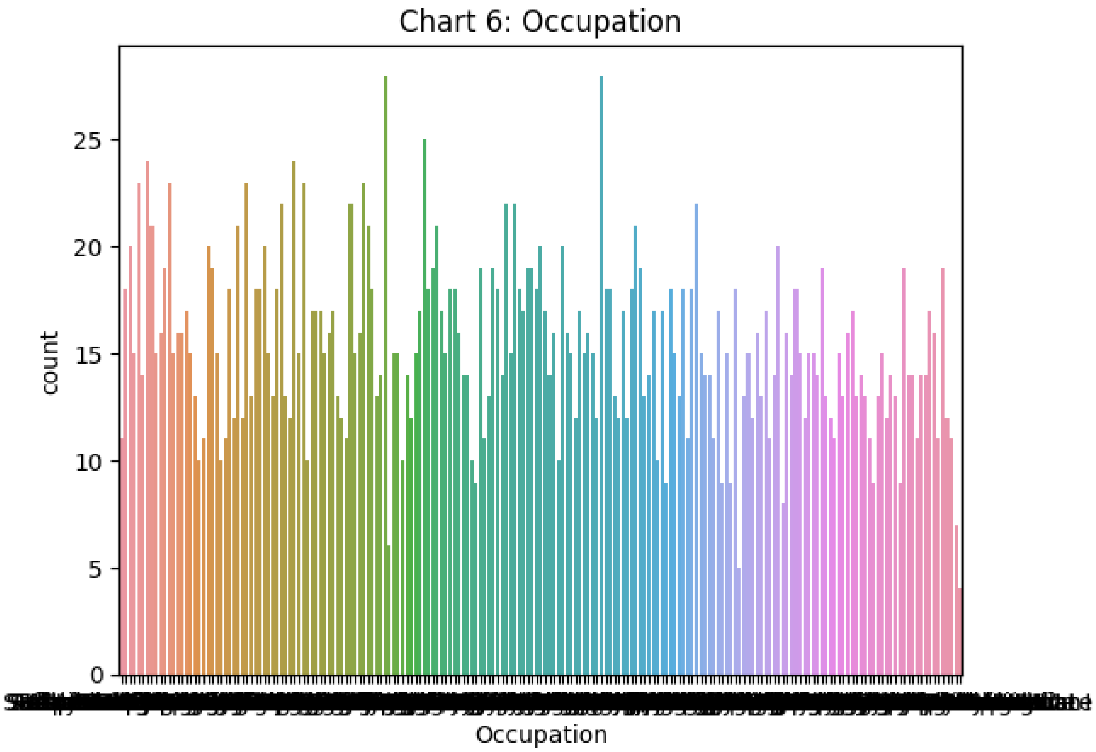


Chart 6: Occupation
count
0
5
10
15
20
25
Occupation

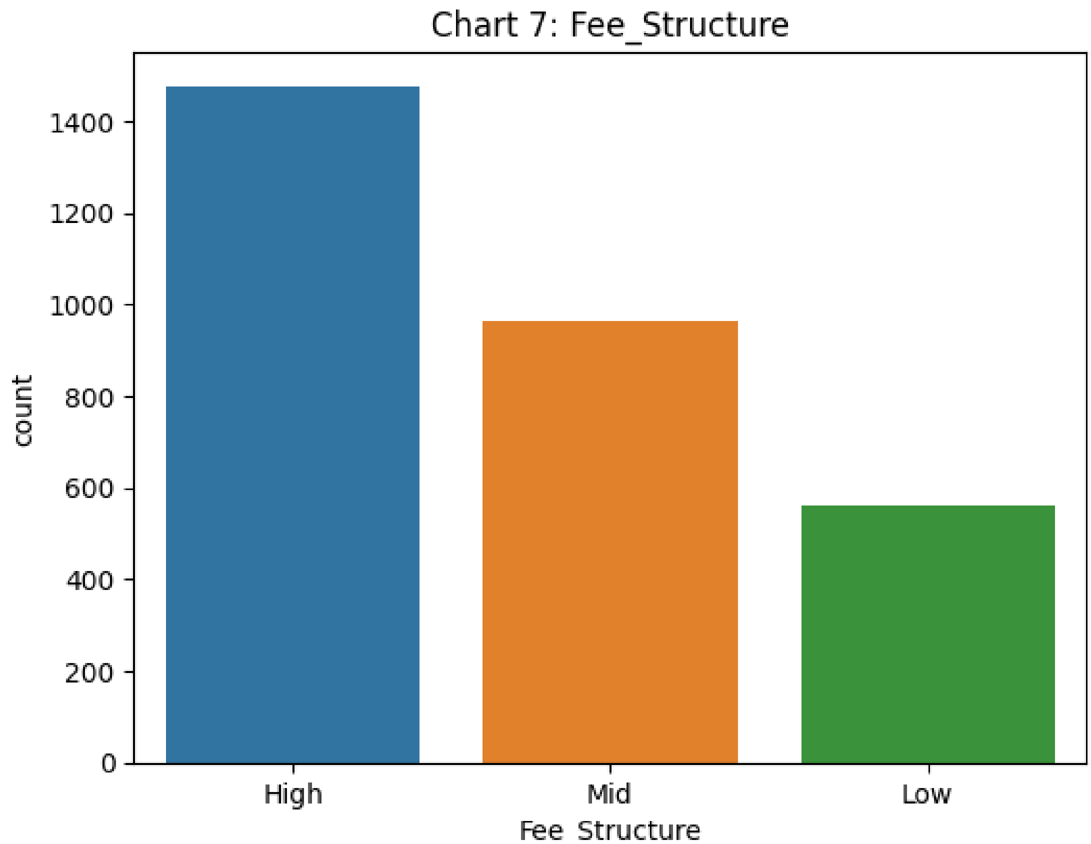


Chart 7: Fee_Structure
count
1400
1200
1000
800
600
400
200
0
High
Mid
Low
Fee_Structure

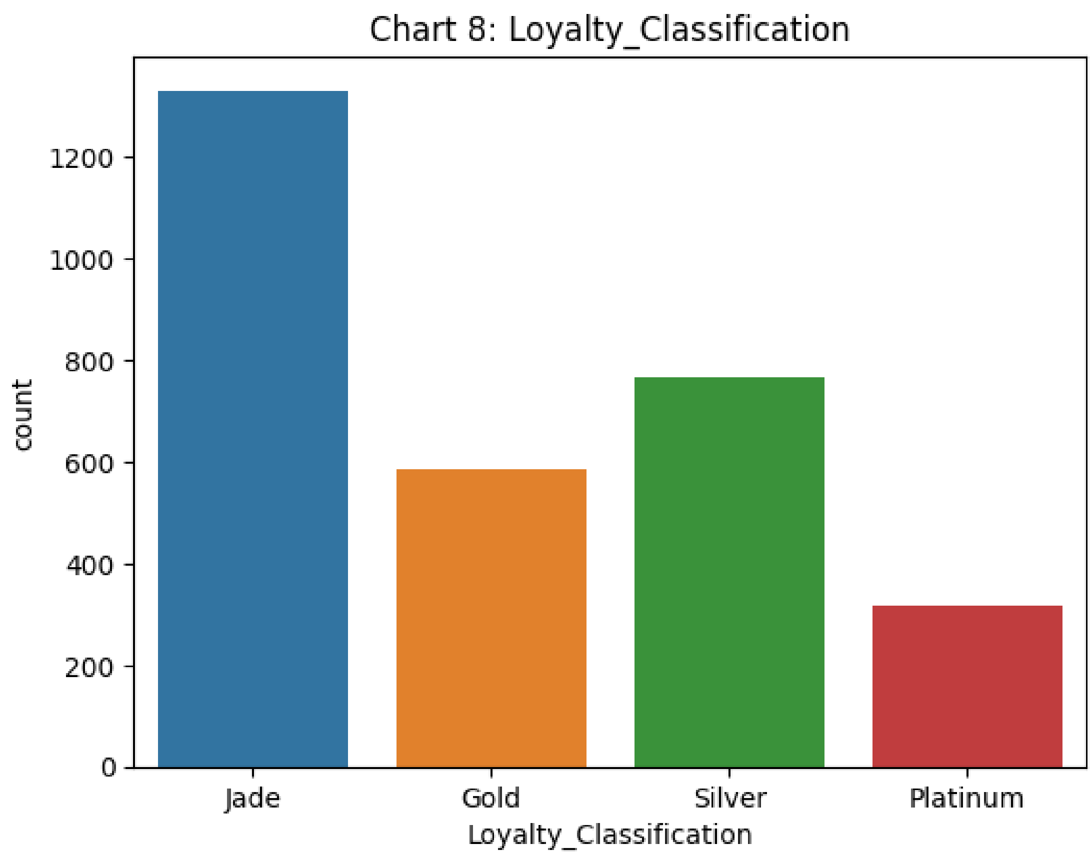


Chart 8: Loyalty_Classification
count
0
200
400
600
800
1000
1200
Jade
Gold
Silver
Platinum
Loyalty_Classification

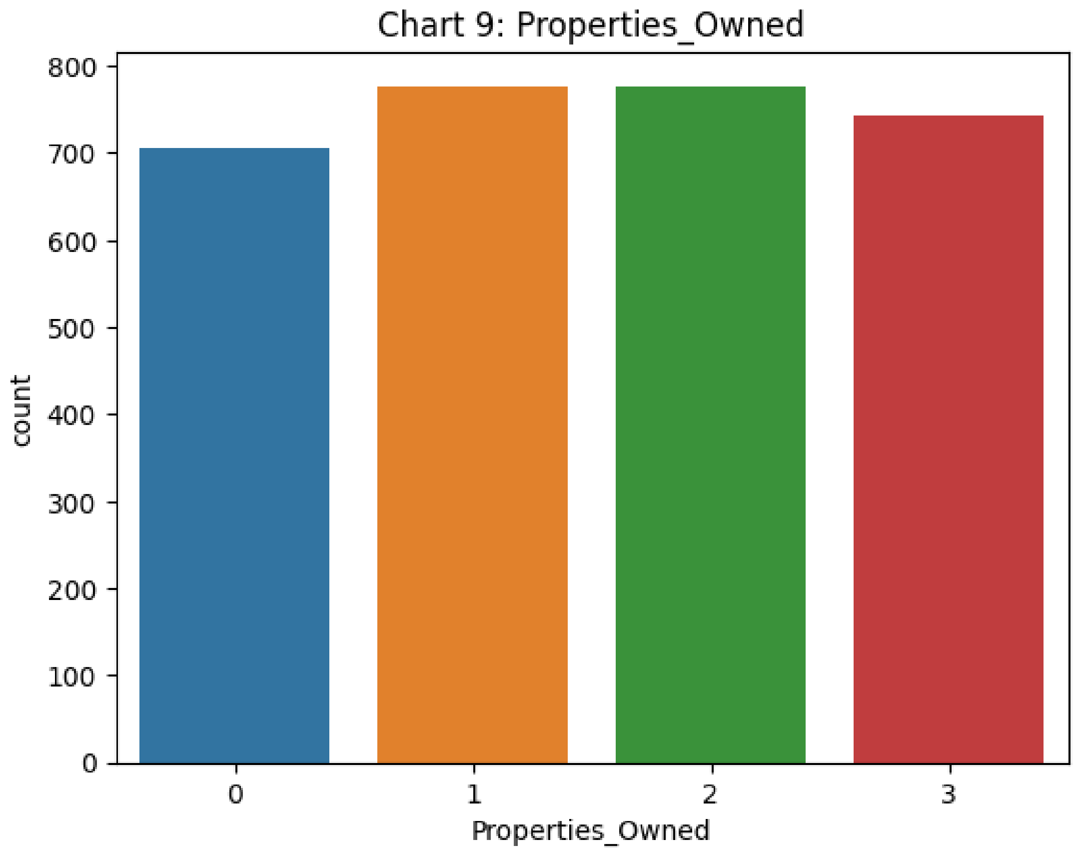


Chart 9: Properties_Owned
count
800
700
600
500
400
300
200
100
0
0
1
2
3
Properties_Owned

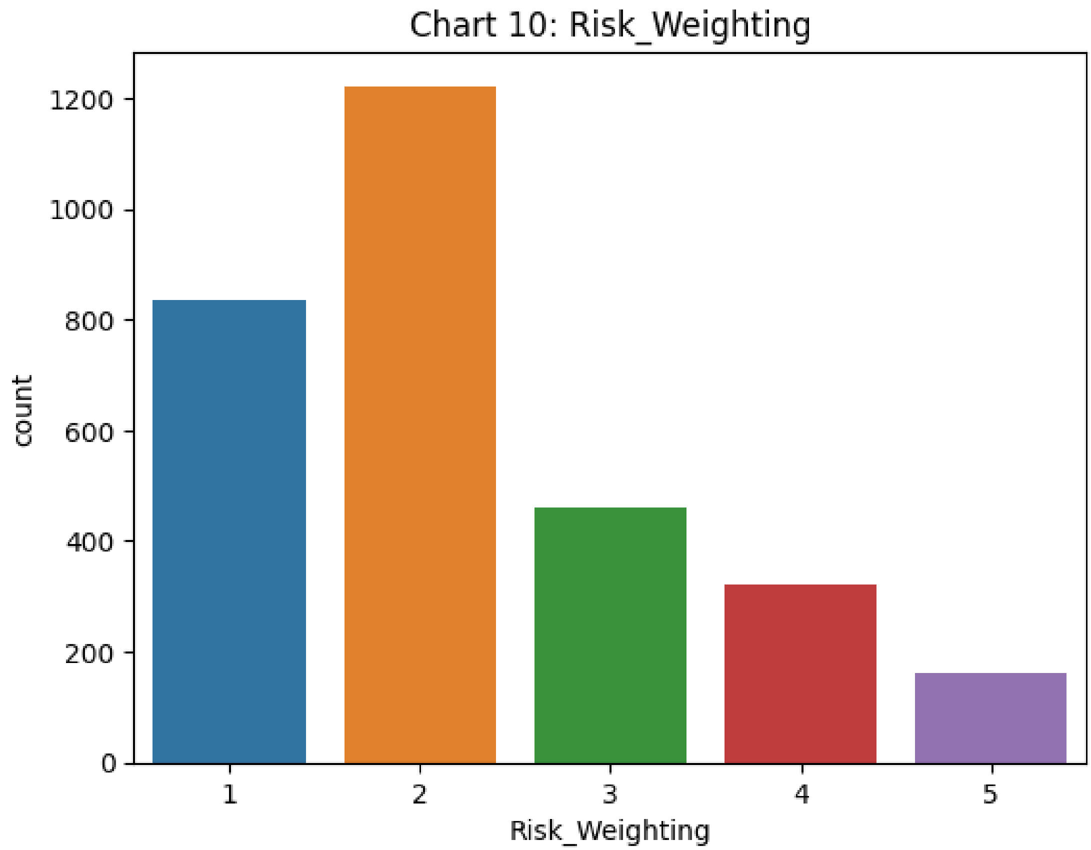
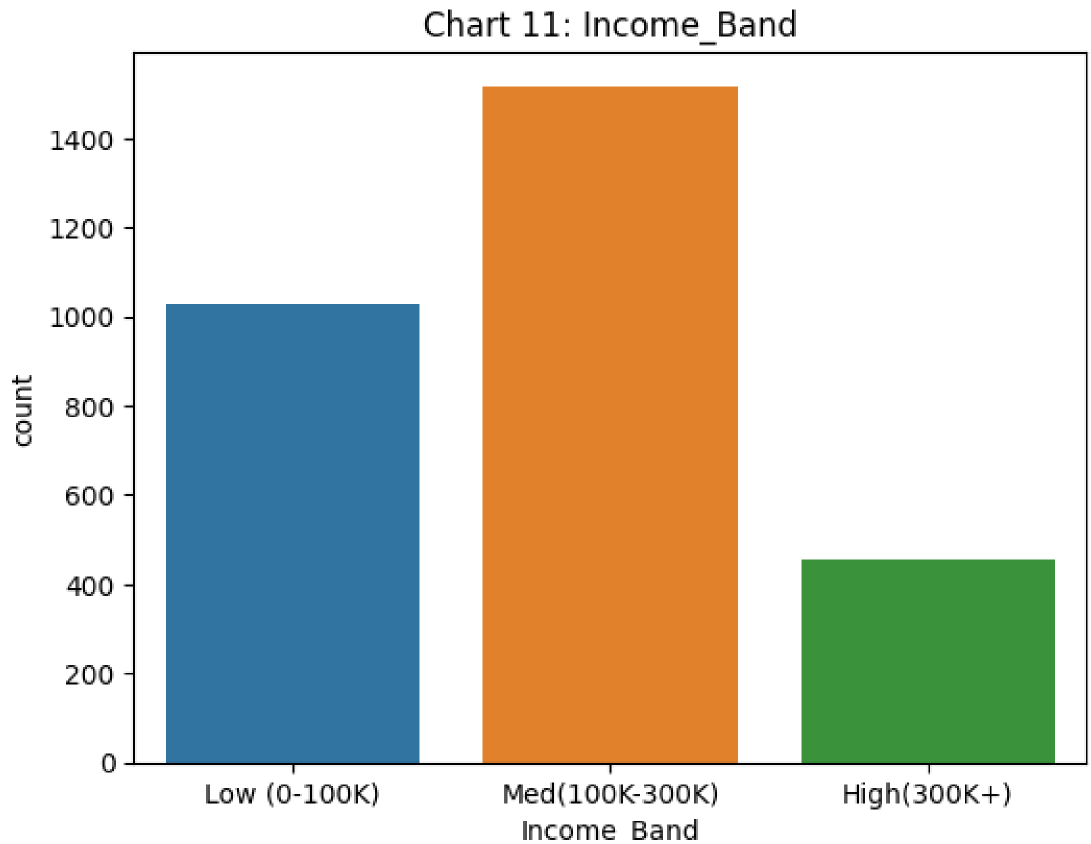


Chart 10: Risk_Weighting
count
1200
1000
800
600
400
200
0
1
2
3
4
5
Risk_Weighting



0.0.6 Bivariate Analysis using Nationality as Hue

[27]:
```
for i, col_name 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) #makes sure each plot starts fresh, using a separate figure␣
↪for each chart.
    sns.countplot(data=df, x=col_name, hue = 'Nationality') #x is the variable
    plt.title(f"Chart {i+1}: {col_name}") #Used this to name the plot according␣
↪to the index (i+1: used to skip 0)
```

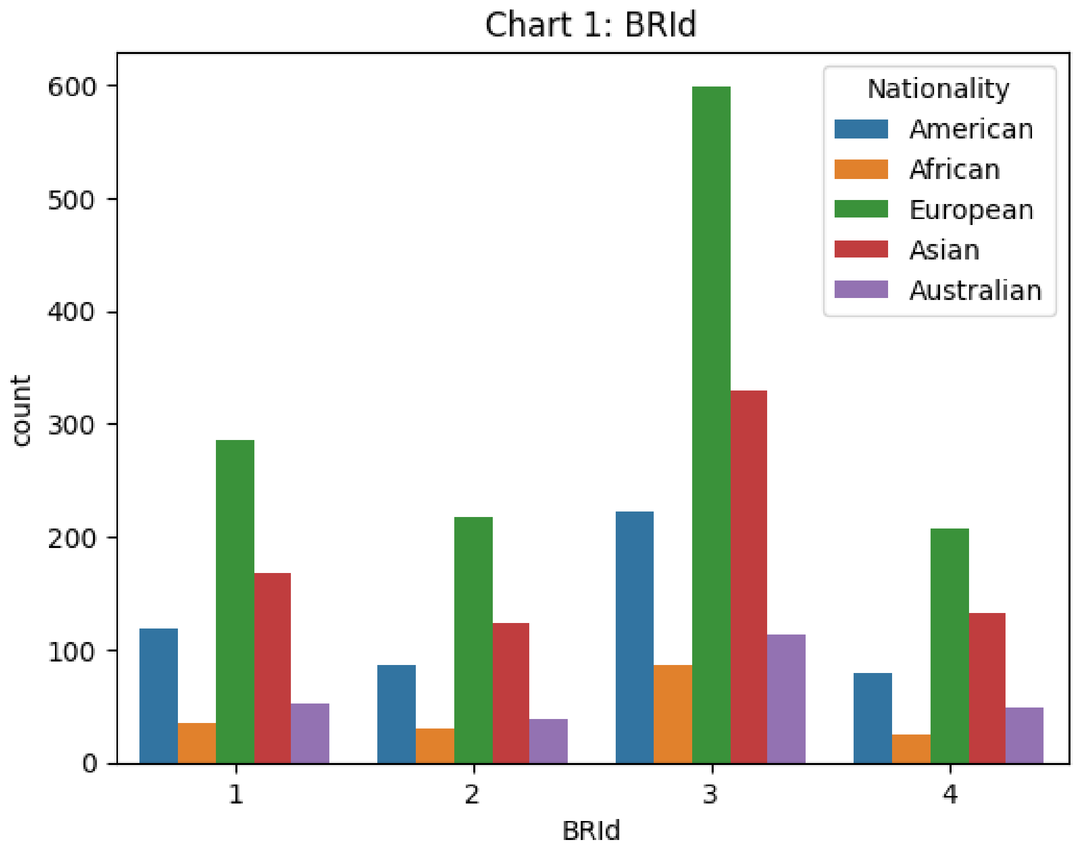


Chart 1: BRId
Nationality
American
African
European
Asian
Australian
count
0
100
200
300
400
500
600
1
2
3
4
BRId

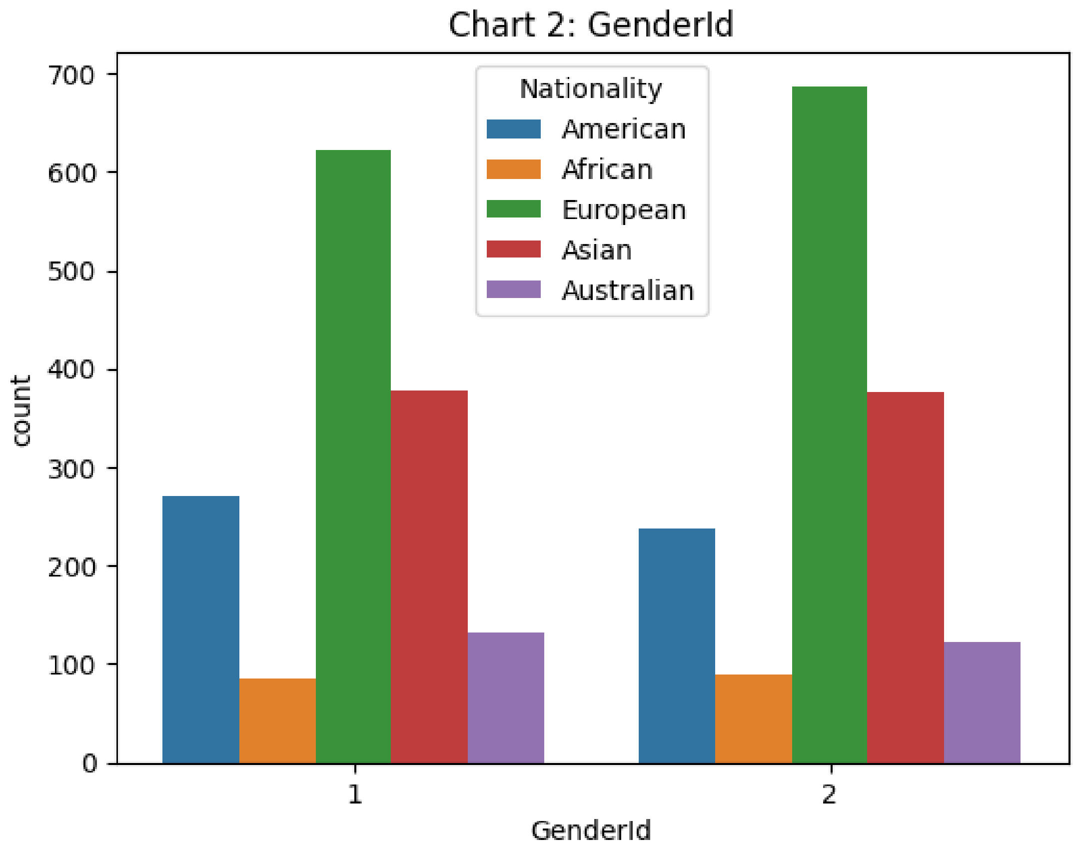


Chart 2: GenderId
Nationality
American
African
European
Asian
Australian
count
0
100
200
300
400
500
600
700
1
2
GenderId

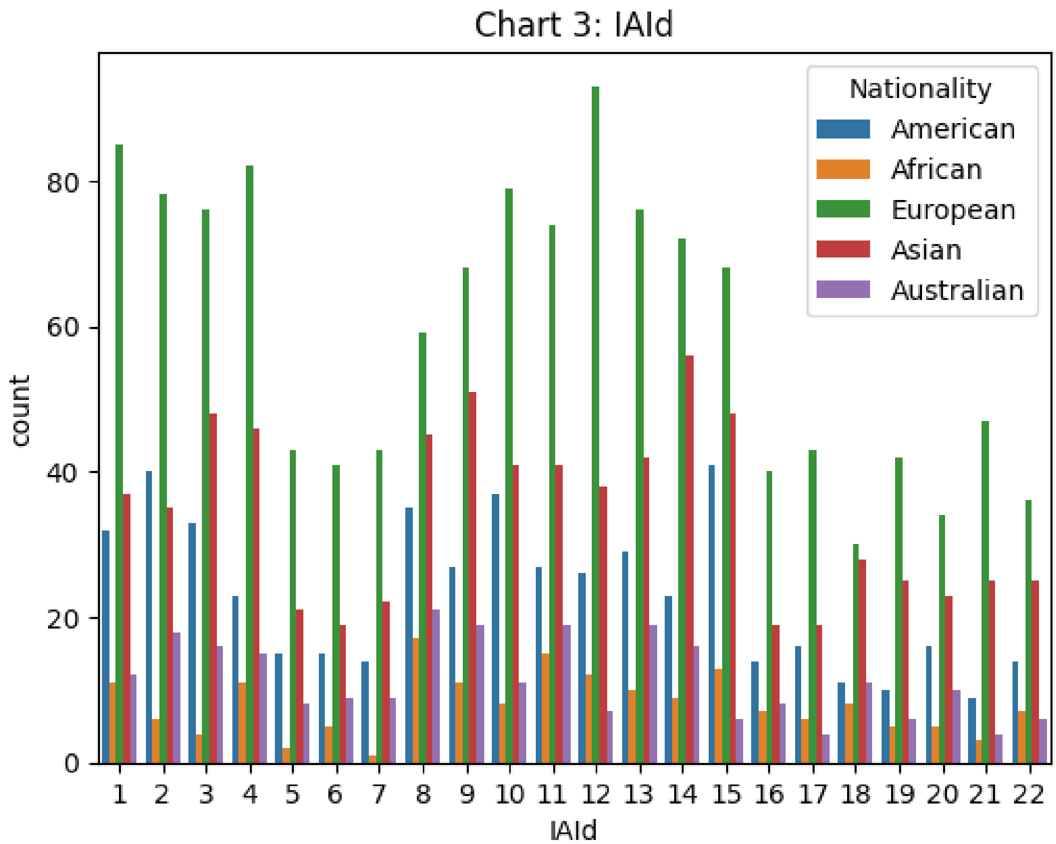


Chart 3: IAId
Nationality
American
African
European
Asian
Australian
count
0
20
40
60
80
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
IAId

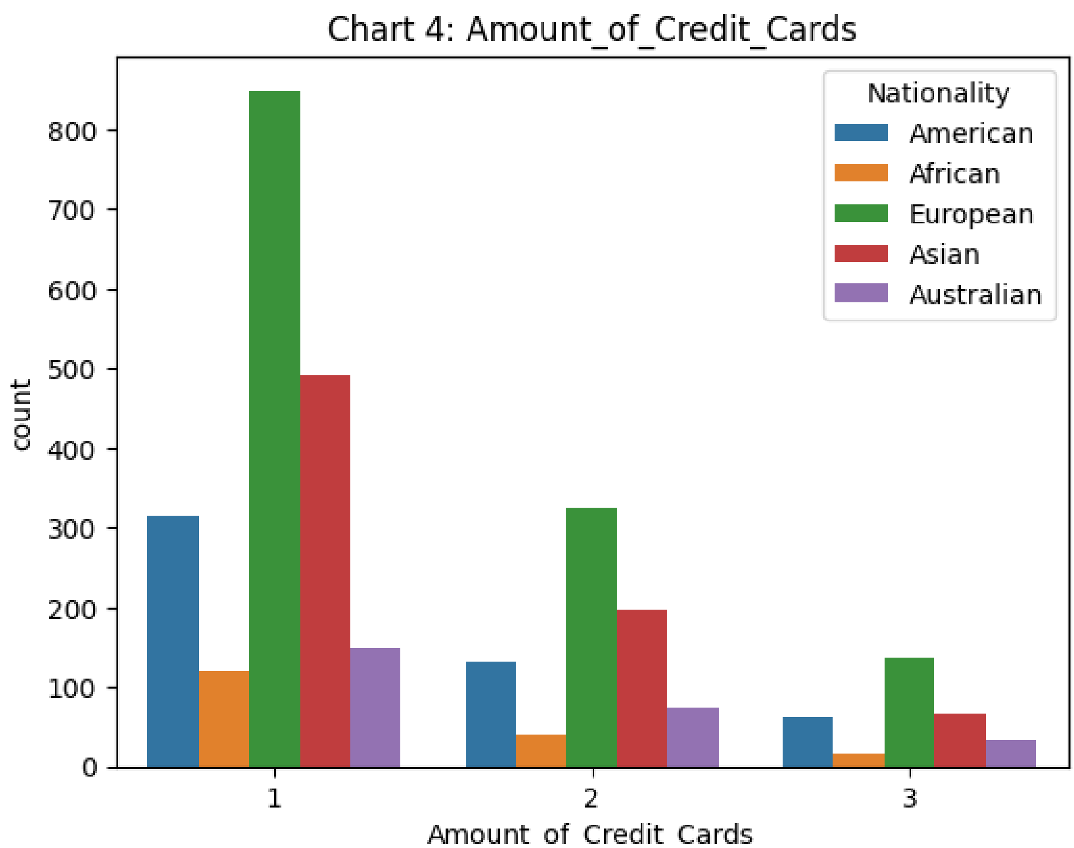


Chart 4: Amount_of_Credit_Cards
Nationality
American
African
European
Asian
Australian
count
0
100
200
300
400
500
600
700
800
1
2
3
Amount_of_Credit_Cards

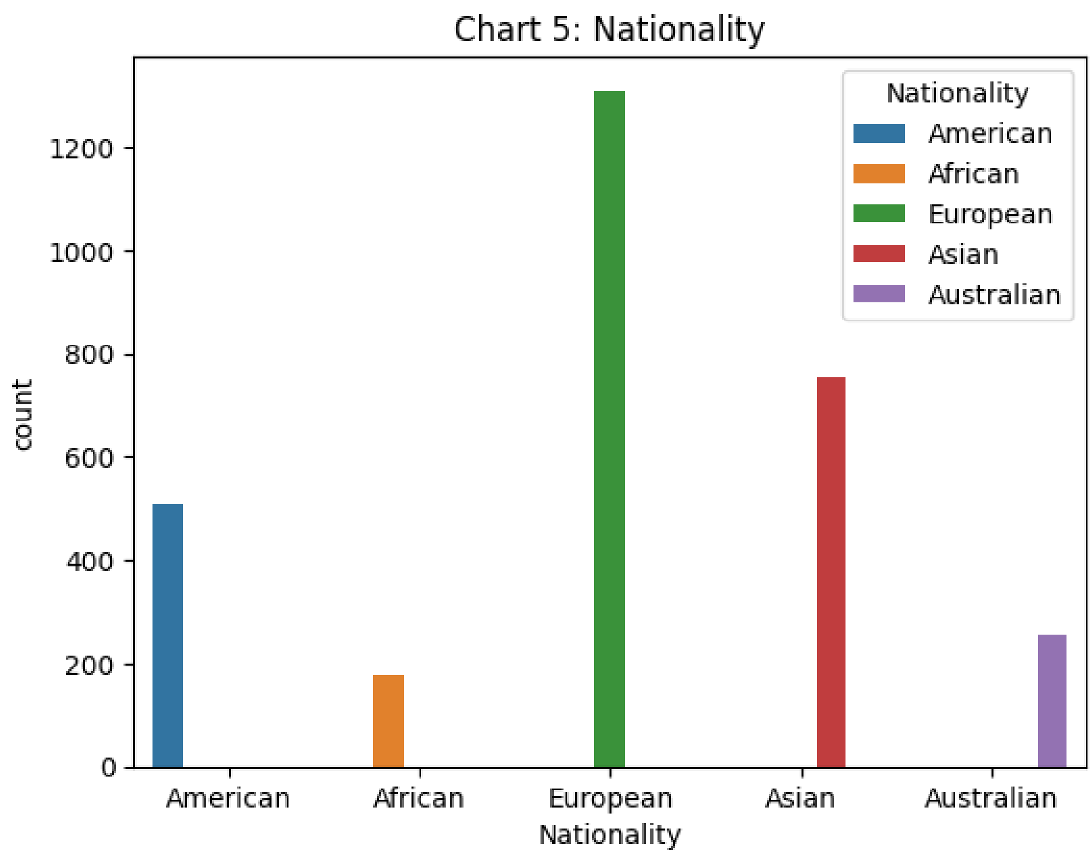


Chart 5: Nationality
Nationality
American
African
European
Asian
Australian
count
1200
1000
800
600
400
200
0
American
African
European
Asian
Australian
Nationality

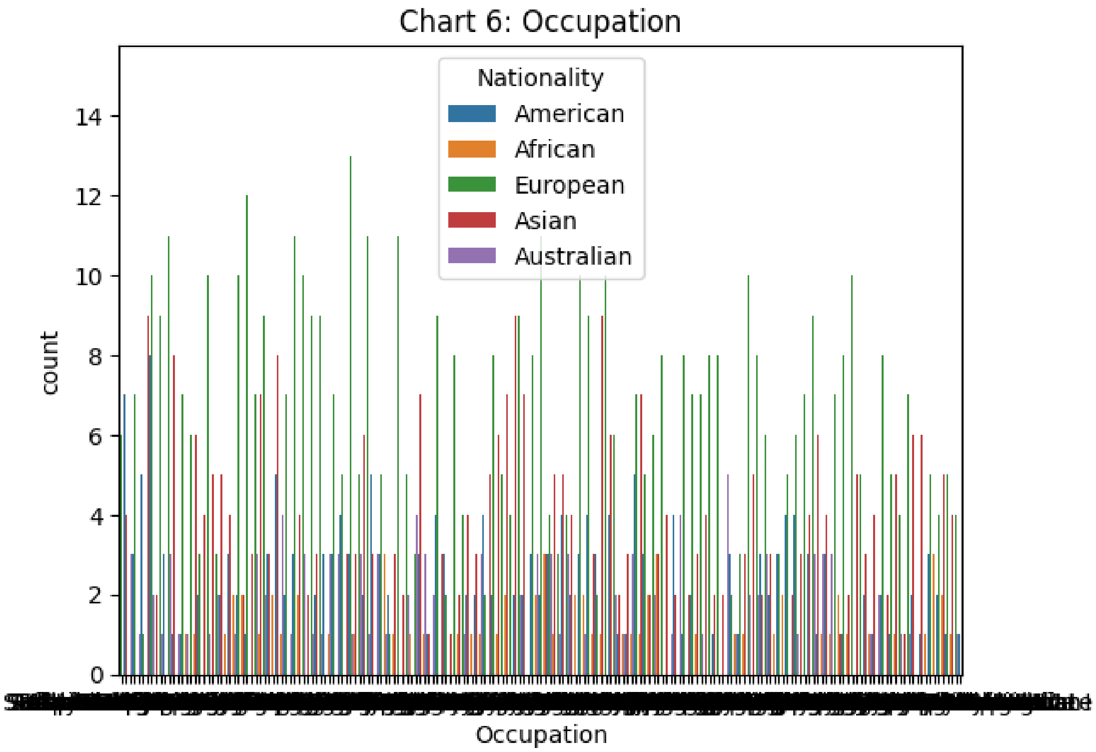


Chart 6: Occupation
Nationality
American
African
European
Asian
Australian
count
0
2
4
6
8
10
12
14
Occupation

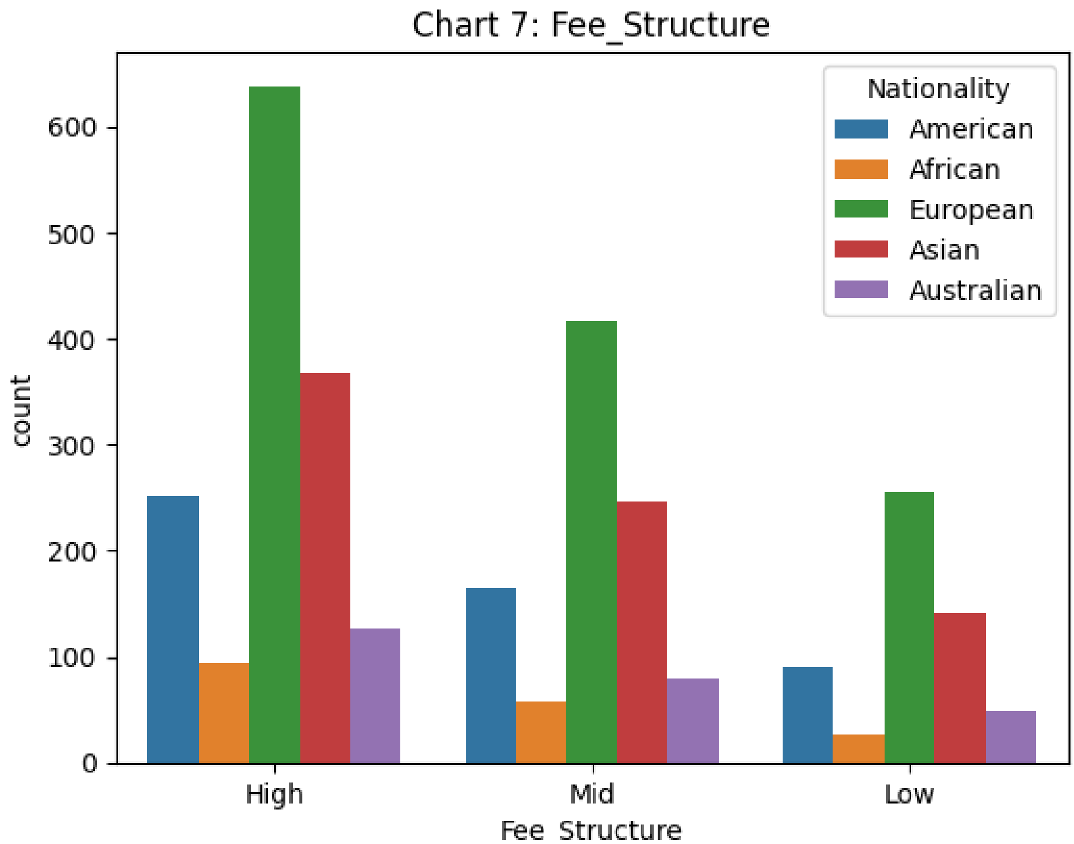


Chart 7: Fee_Structure
Nationality
American
African
European
Asian
Australian
count
600
500
400
300
200
100
0
High
Mid
Low
Fee_Structure

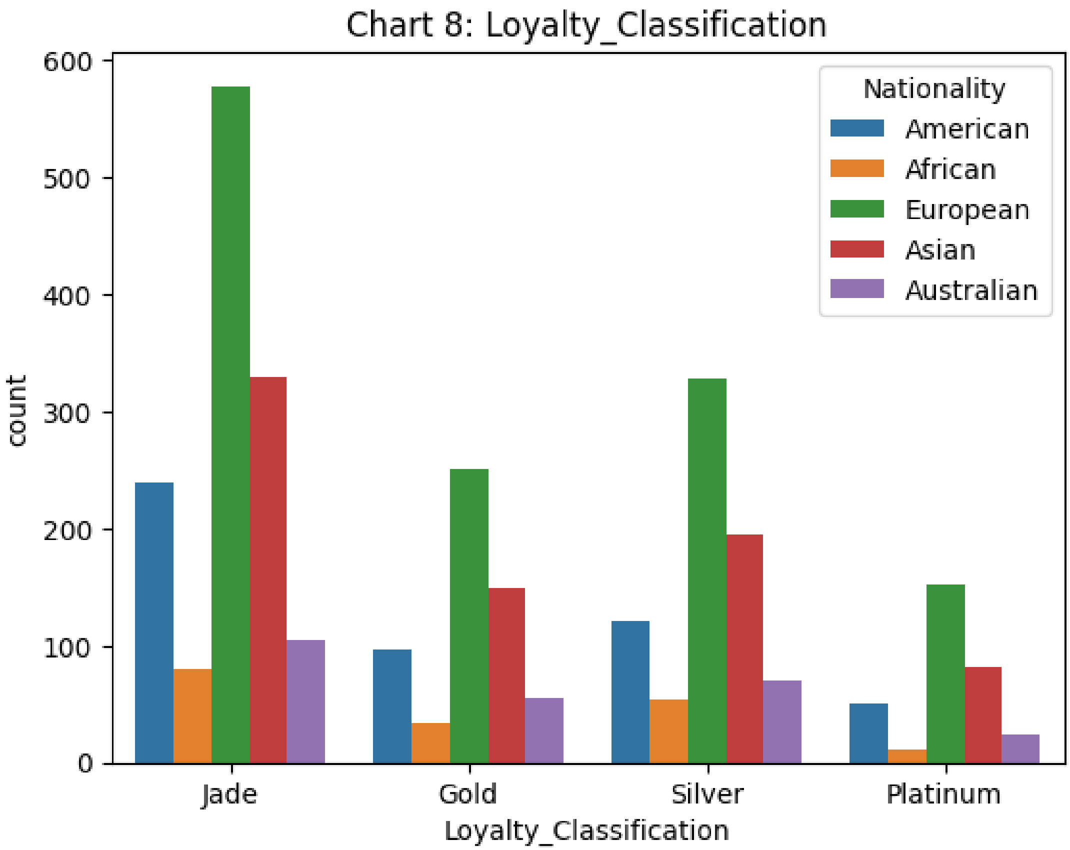


Chart 8: Loyalty_Classification
Nationality
American
African
European
Asian
Australian
count
600
500
400
300
200
100
0
Jade
Gold
Silver
Platinum
Loyalty_Classification

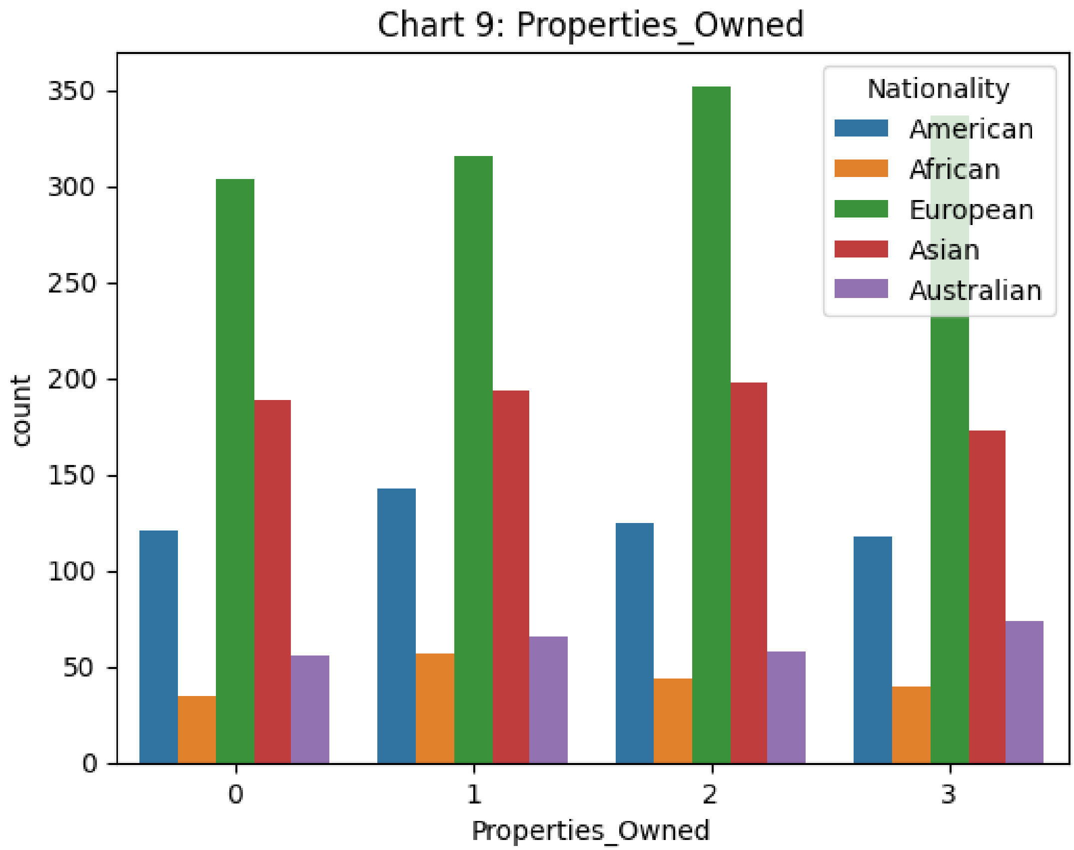


Chart 9: Properties_Owned
Nationality
American
African
European
Asian
Australian
count
0
50
100
150
200
250
300
350
0
1
2
3
Properties_Owned

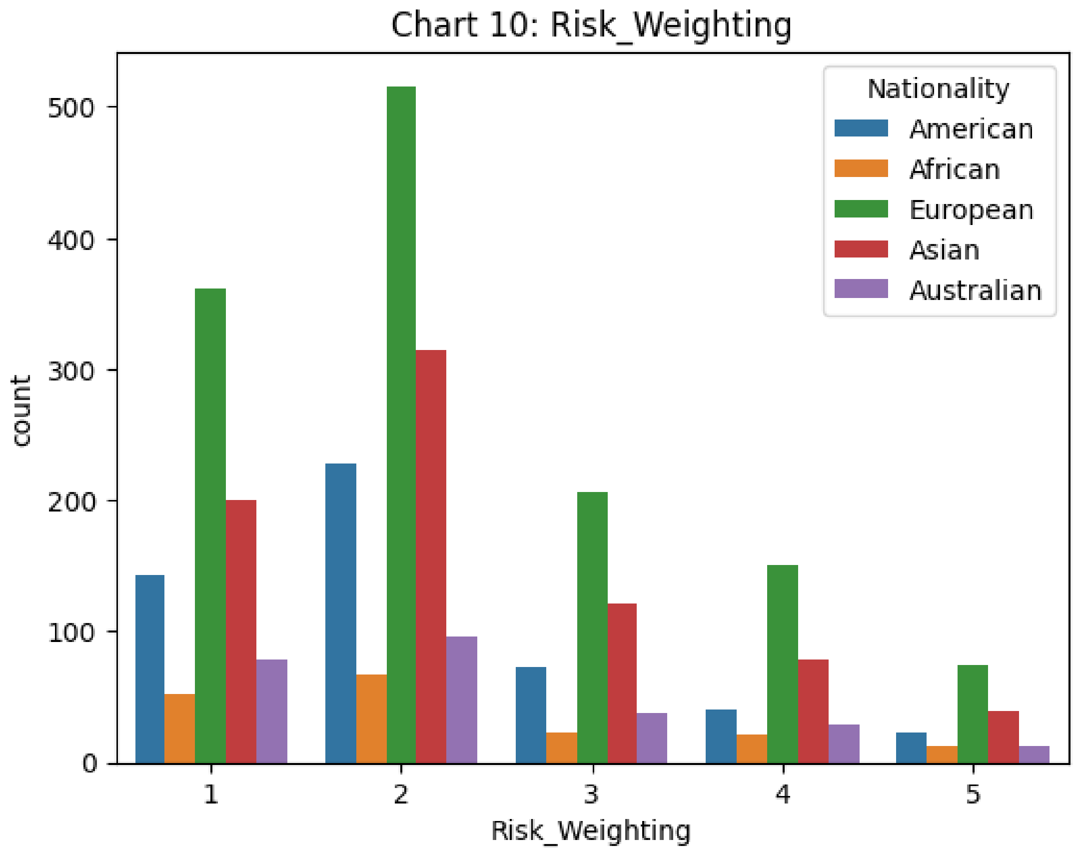
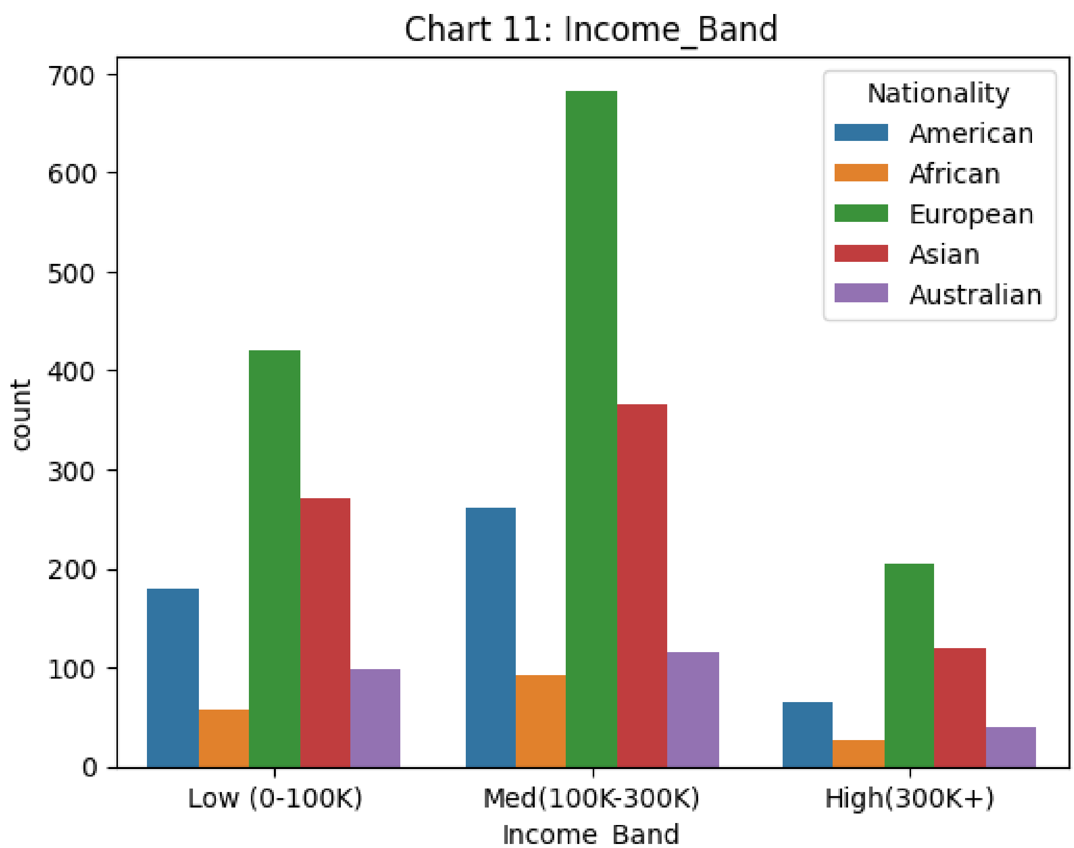
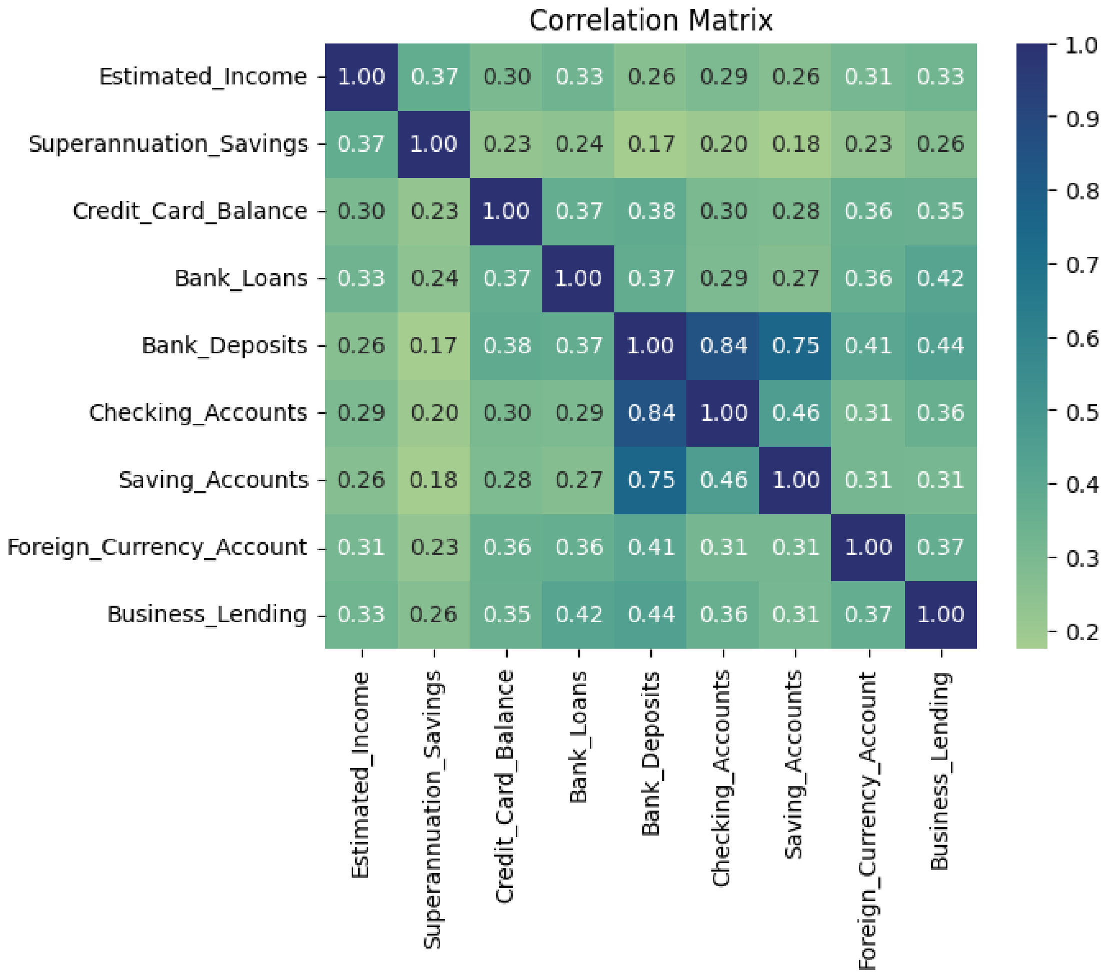


Chart 10: Risk_Weighting
Nationality
American
African
European
Asian
Australian
count
0
100
200
300
400
500
1
2
3
4
5
Risk_Weighting



0.0.7 Numerical Analysis

[31]:
```
numerical_cols = df[['Estimated_Income', 'Superannuation_Savings',
    'Credit_Card_Balance', 'Bank_Loans', 'Bank_Deposits', 'Checking_Accounts',
    'Saving_Accounts', 'Foreign_Currency_Account', 'Business_Lending']].columns
correlation_matrix = df[numerical_cols].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='crest', fmt=".2f")
plt.title("Correlation Matrix")
plt.show()
```



0.1 Insights of EDA

0.1.1 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.