

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

```
!wget https://raw.githubusercontent.com/azar-s91/dataset/master/BankChurners.csv
```

```
--2023-05-02 09:57:38--
```

```
https://raw.githubusercontent.com/azar-s91/dataset/master/BankChurners.csv
```

```
Resolving raw.githubusercontent.com (raw.githubusercontent.com)...
```

```
185.199.108.133, 185.199.109.133, 185.199.110.133, ...
```

```
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|
```

```
185.199.108.133|:443... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 1210878 (1.2M) [text/plain]
```

```
Saving to: 'BankChurners.csv'
```

```
BankChurners.csv 100%[=====>] 1.15M --.-KB/s in 0.03s
```

```
2023-05-02 09:57:38 (37.9 MB/s) - 'BankChurners.csv' saved
[1210878/1210878]
```

```
data = pd.read_csv('BankChurners.csv', sep=",")
data.head()
```

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count
0	768805383	Existing Customer	45	M	3
1	818770008	Existing Customer	49	F	5
2	713982108	Existing Customer	51	M	3
3	769911858	Existing Customer	40	F	4
4	709106358	Existing Customer	40	M	3

	Education_Level	Marital_Status	Income_Category	Card_Category
0	High School	Married	\$60K - \$80K	Blue
1	Graduate	Single	Less than \$40K	Blue
2	Graduate	Married	\$80K - \$120K	Blue
3	High School	Unknown	Less than \$40K	Blue
4	Uneducated	Married	\$60K - \$80K	Blue

	Months_on_book	...	Months_Inactive_12_mon	Contacts_Count_12_mon
\				
0	39	...	1	3
1	44	...	1	2
2	36	...	1	0
3	34	...	4	1
4	21	...	1	0

	Credit_Limit	Total_Revolving_Bal	Avg_Open_To_Buy
Total_Amt_Chng_Q4_Q1			
\			
0	12691.0	777	11914.0
1.335			
1	8256.0	864	7392.0
1.541			
2	3418.0	0	3418.0
2.594			
3	3313.0	2517	796.0
1.405			
4	4716.0	0	4716.0
2.175			

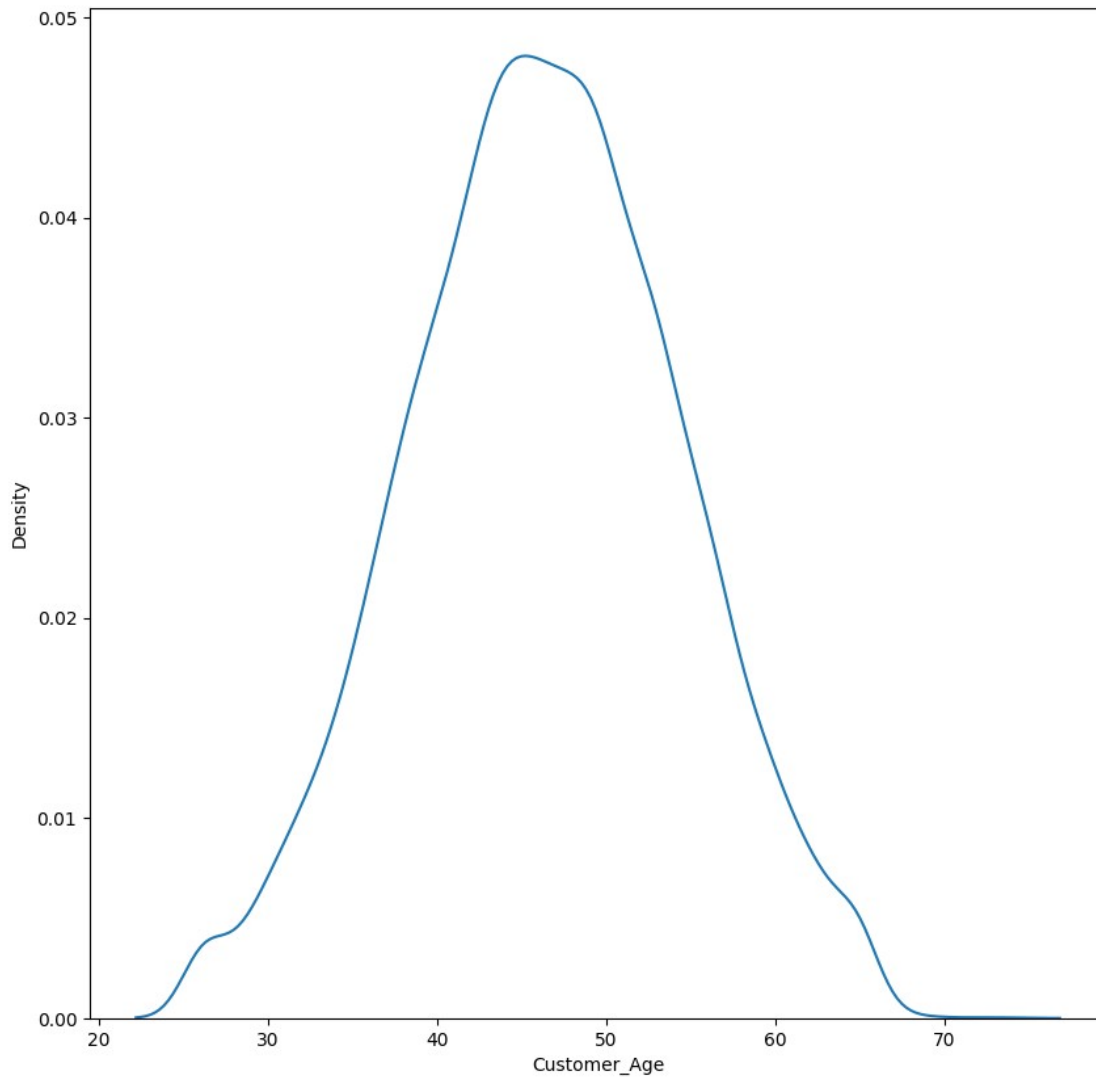
	Total_Trans_Amt	Total_Trans_Ct	Total_Ct_Chng_Q4_Q1
Avg_Utilization_Ratio			
\			
0	1144	42	1.625
0.061			
1	1291	33	3.714
0.105			
2	1887	20	2.333
0.000			
3	1171	20	2.333
0.760			
4	816	28	2.500
0.000			

[5 rows x 21 columns]

Рассмотрим распределение клиентов банка по возрасту

```
fig, ax = plt.subplots(figsize=(10,10))
sns.kdeplot(data=data, x="Customer_Age")
```

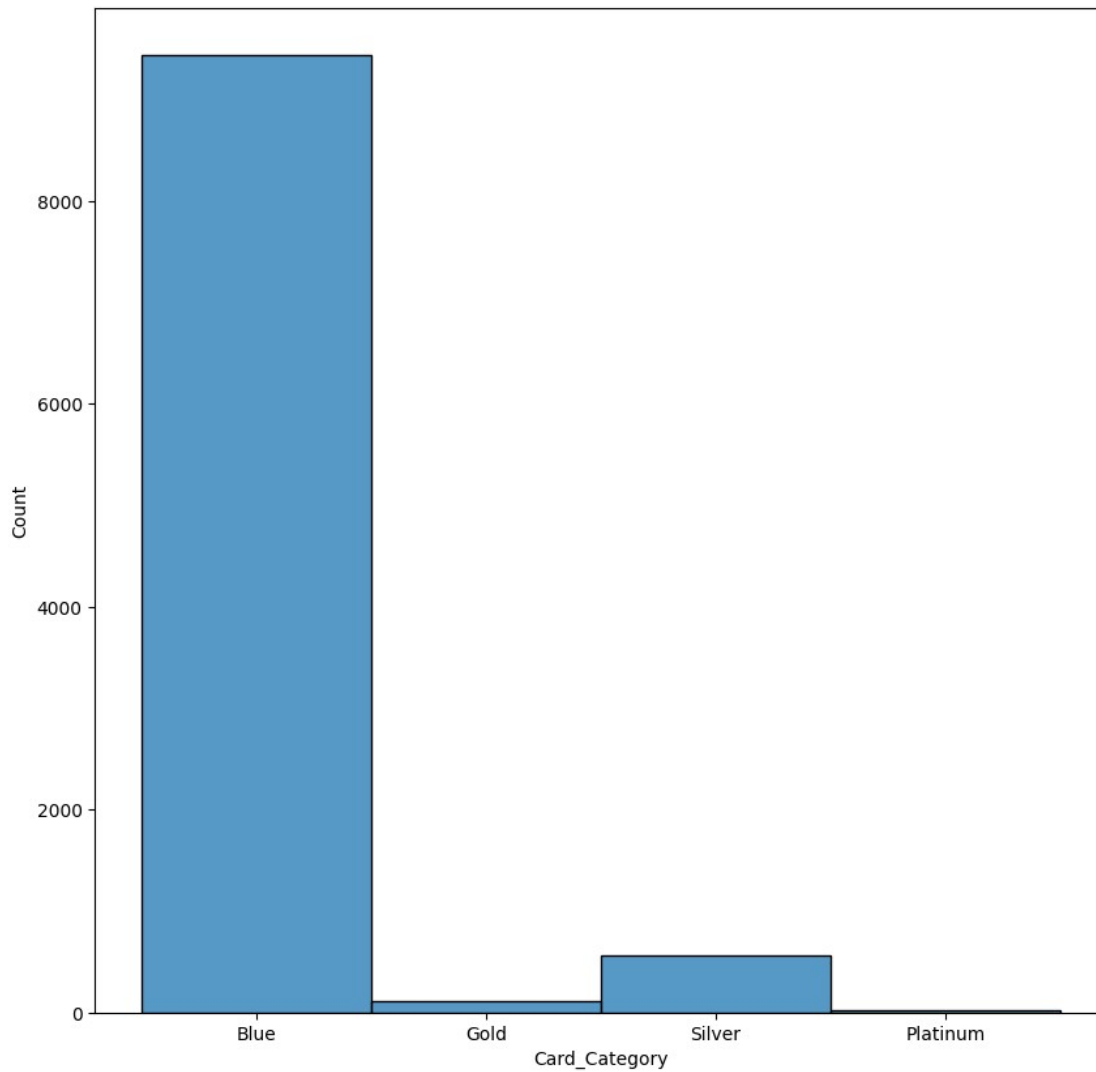
```
<Axes: xlabel='Customer_Age', ylabel='Density'>
```



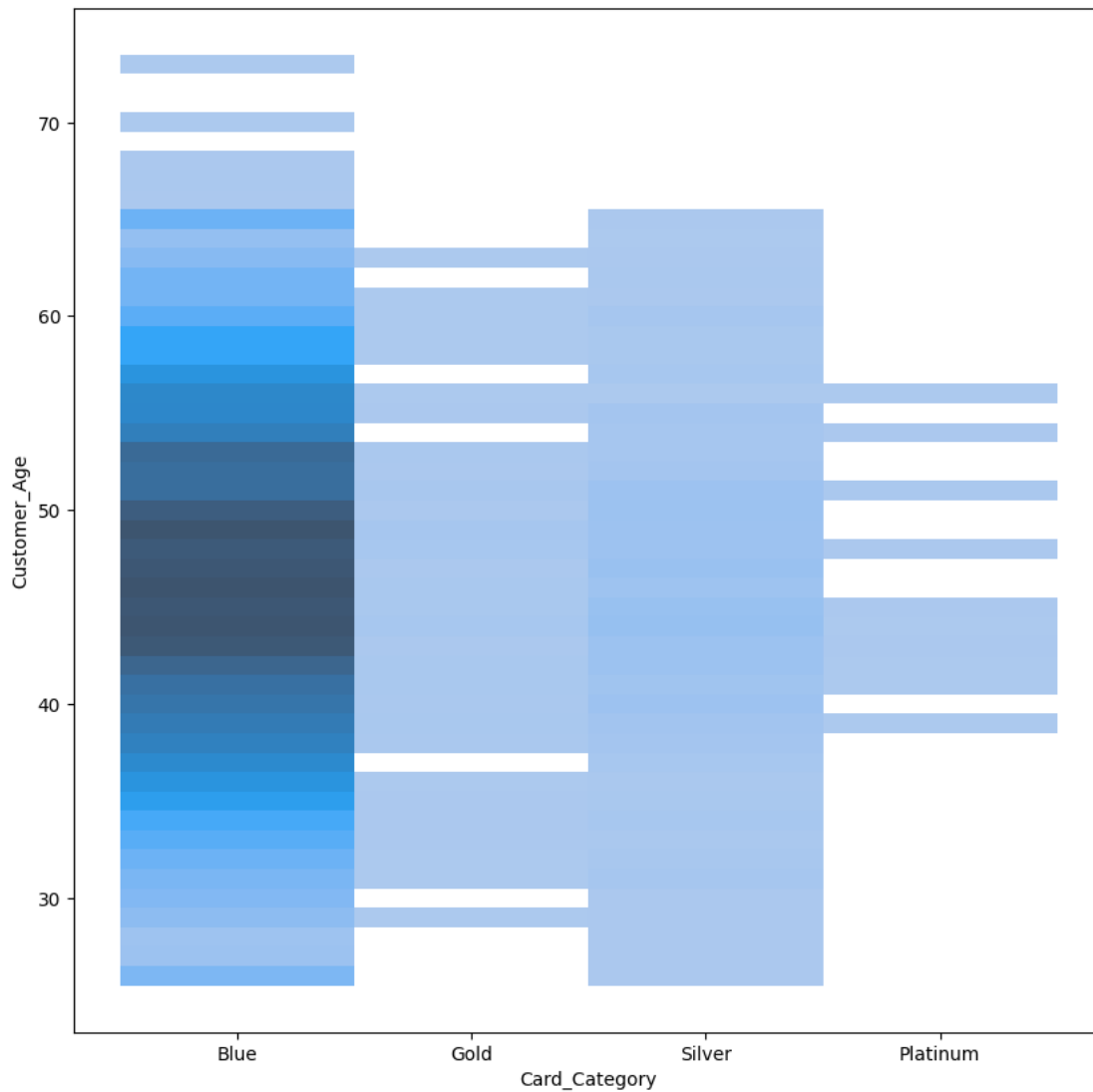
Рассмотрим число карт различного вида:

```
fig, ax = plt.subplots(figsize=(10,10))  
sns.histplot(data['Card_Category'], discrete=True)
```

```
<Axes: xlabel='Card_Category', ylabel='Count'>
```



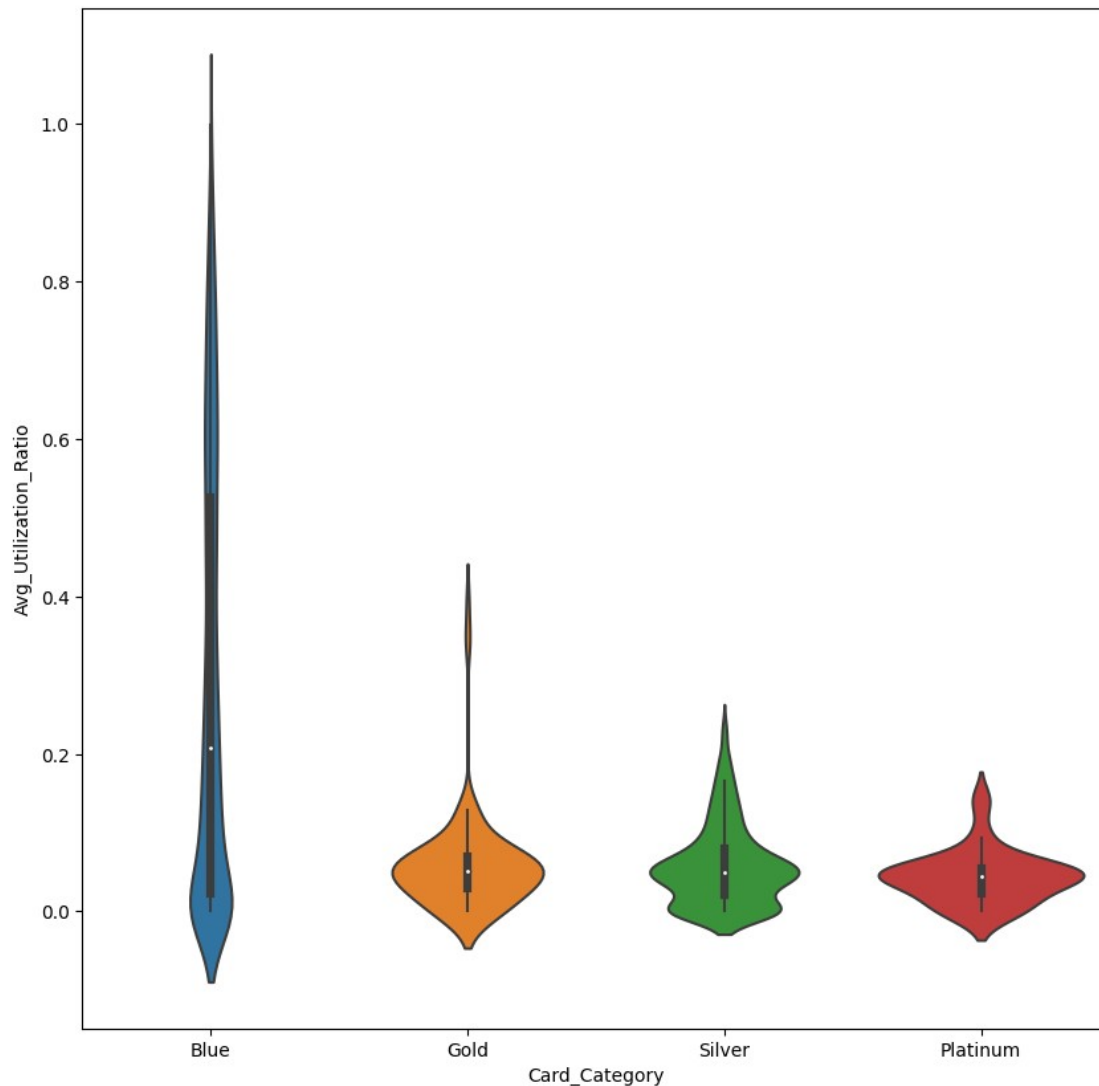
```
fig, ax = plt.subplots(figsize=(10,10))
sns.histplot(data, x="Card_Category", y="Customer_Age", discrete=True)
<Axes: xlabel='Card_Category', ylabel='Customer_Age'>
```



Рассмотрим зависимость средней используемости карты от некоторых категориальных признаков. Для этого применим график вида скрипка.

```
fig, ax = plt.subplots(figsize=(10,10))
sns.violinplot(x='Card_Category', y='Avg_Utilization_Ratio',
data=data)
```

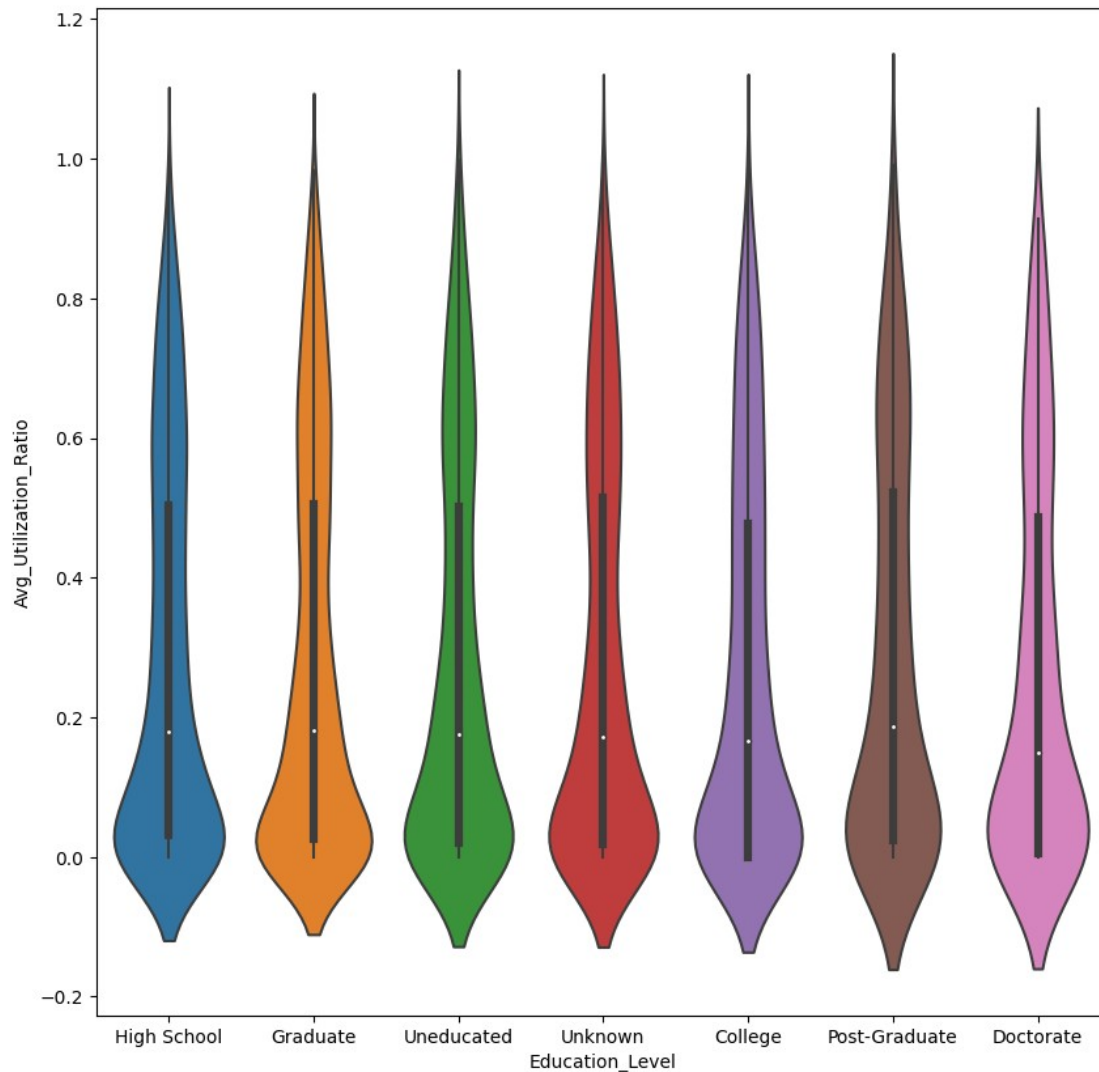
```
<Axes: xlabel='Card_Category', ylabel='Avg_Utilization_Ratio'>
```



Из данного графика можно сделать вывод о том, что наибольшая активность присутствует у карт низкого уровня.

```
fig, ax = plt.subplots(figsize=(10,10))  
sns.violinplot(x='Education_Level', y='Avg_Utilization_Ratio',  
data=data)
```

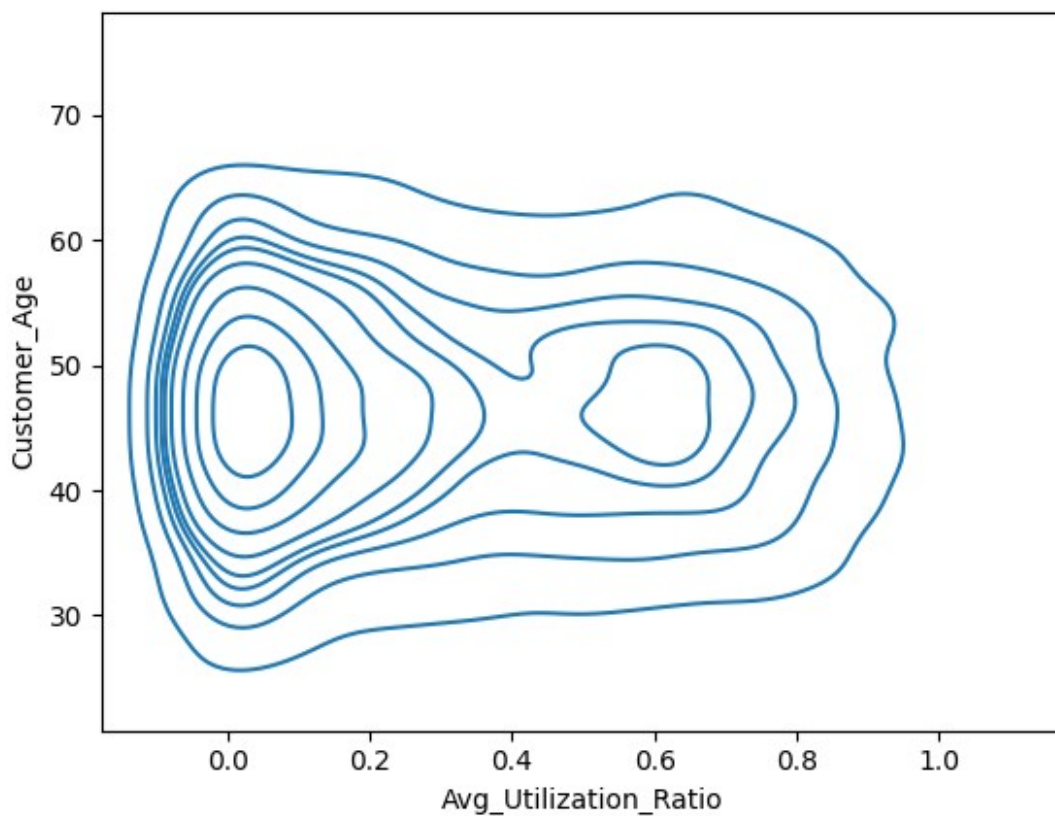
```
<Axes: xlabel='Education_Level', ylabel='Avg_Utilization_Ratio'>
```



Из данного графика видно, что зависимости между уровнем образования и активностью использования карты нет.

```
sns.kdeplot(data=data, x="Avg_Utilization_Ratio", y="Customer_Age")
```

```
<Axes: xlabel='Avg_Utilization_Ratio', ylabel='Customer_Age'>
```

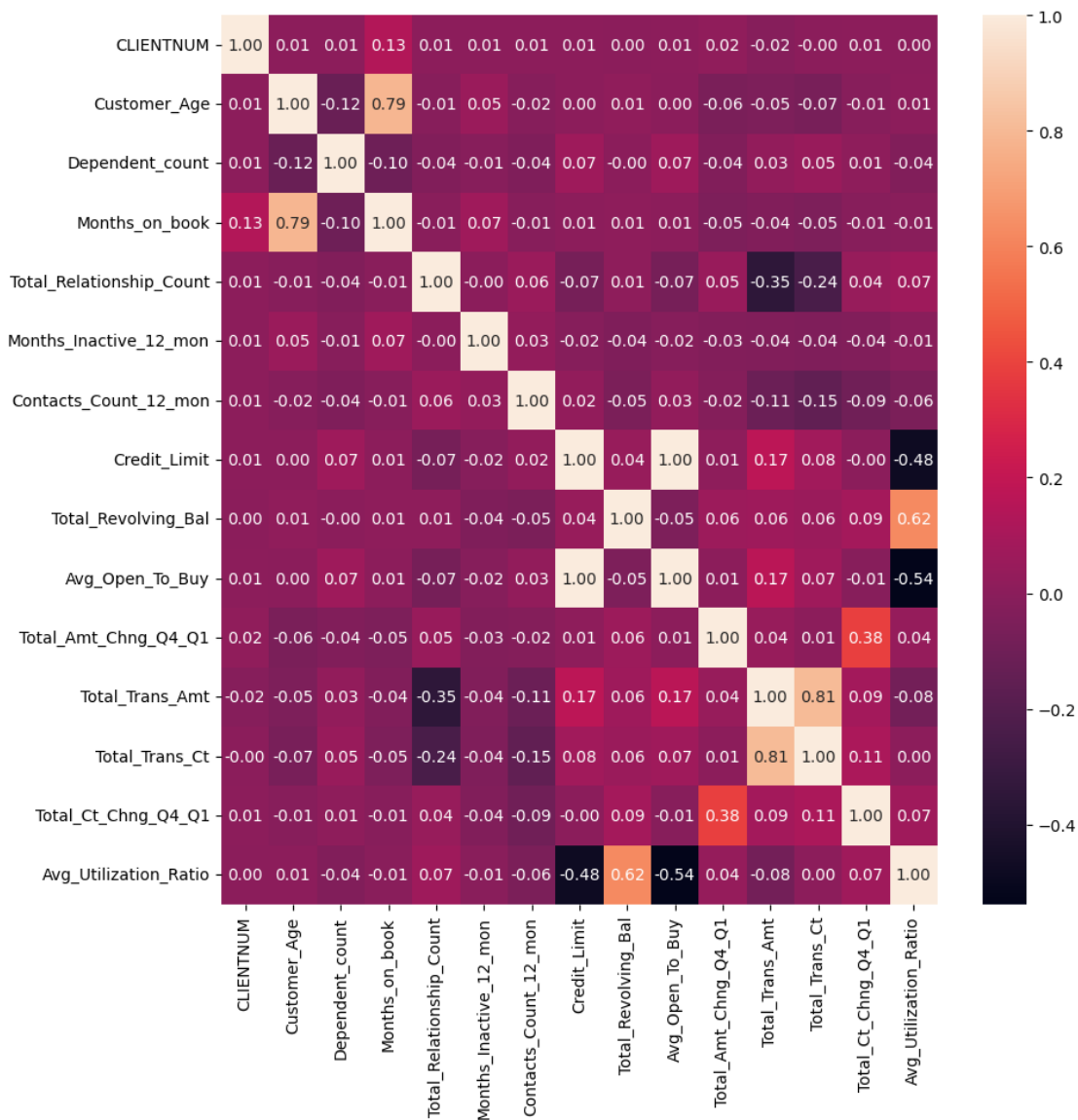


```
fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(data.corr(method='pearson'), annot=True, fmt='.2f')
fig.suptitle('Корреляционная матрица')
```

```
Text(0.5, 0.98, 'Корреляционные матрицы, построенные различными методами')
```



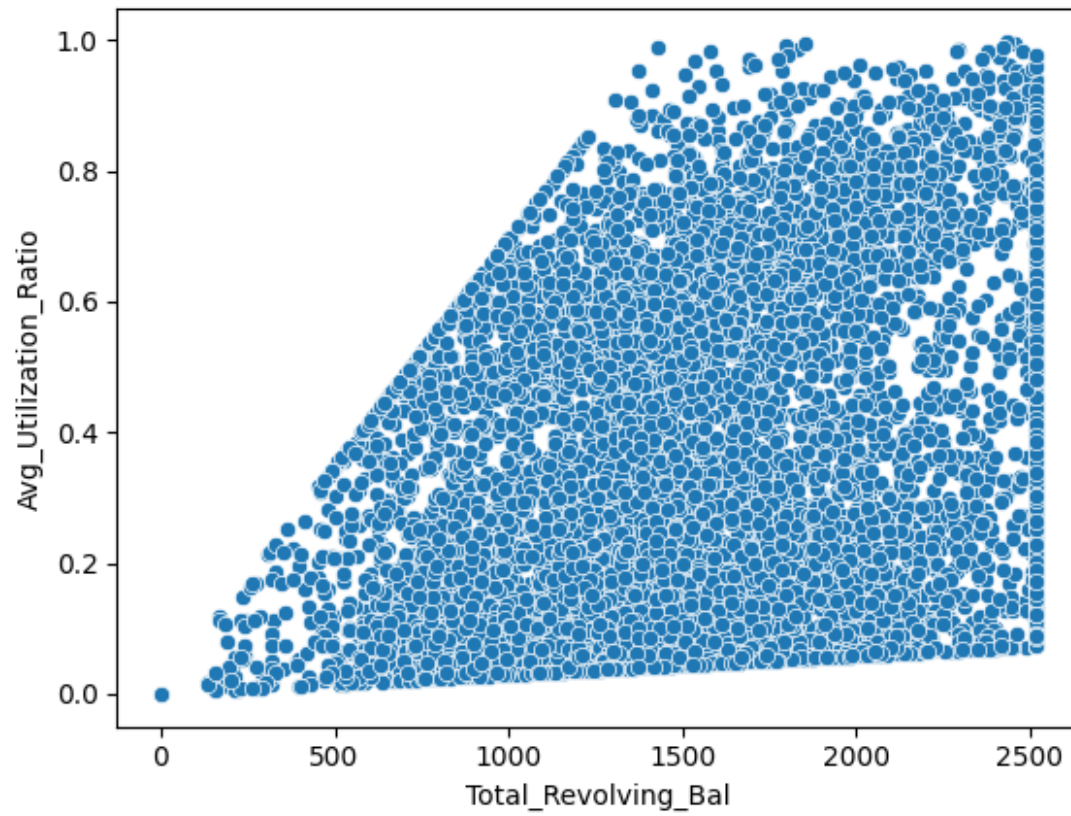
Корреляционные матрицы, построенные различными методами



Рассмотрим зависимости имеющие наибольший коэффициент корреляции:

```
sns.scatterplot(x='Total_Revolving_Bal', y='Avg_Utilization_Ratio', data=data)
```

```
<Axes: xlabel='Total_Revolving_Bal', ylabel='Avg_Utilization_Ratio'>
```



```
sns.scatterplot(x='Customer_Age', y='Months_on_book', data=data)  
<Axes: xlabel='Customer_Age', ylabel='Months_on_book'>
```

