EDA on Bank Churners

Importing Libaries

In [1]:

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

Reading the Dataset into Python

In [2]:

```
data = pd.read_csv('BankChurners.csv')
```

Data Exploration

In [3]:

```
data.head()
```

Out[3]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Sta
0	768805383	Existing Customer	45	M	3	High School	Marr
1	818770008	Existing Customer	49	F	5	Graduate	Sin
2	713982108	Existing Customer	51	M	3	Graduate	Marr
3	769911858	Existing Customer	40	F	4	High School	Unknc
4	709106358	Existing Customer	40	M	3	Uneducated	Marr

5 rows × 23 columns

In [4]:

```
data.tail()
```

Out[4]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marita
10122	772366833	Existing Customer	50	M	2	Graduate	
10123	710638233	Attrited Customer	41	M	2	Unknown	I
10124	716506083	Attrited Customer	44	F	1	High School	
10125	717406983	Attrited Customer	30	M	2	Graduate	U
10126	714337233	Attrited Customer	43	F	2	Graduate	

5 rows × 23 columns

In [5]:

```
data.ndim
```

Out[5]:

```
2
```

In [6]:

```
data.shape
```

Out[6]:

```
(10127, 23)
```

In [7]:

```
data.size
```

Out[7]:

```
232921
```

In [64]:

```
data.columns
```

Out[64]:

```
Index(['CLIENTNUM', 'Attrition_Flag', 'Customer_Age', 'Gender',
       'Dependent_count', 'Education_Level', 'Marital_Status',
       'Income_Category', 'Card_Category', 'Months_on_book',
       'Total_Relationship_Count', 'Months_Inactive_12_mon',
       'Contacts_Count_12_mon', 'Credit_Limit', 'Total_Revolving_Bal',
       'Avg_Open_To_Buy', 'Total_Amt_Chng_Q4_Q1', 'Total_Trans_Amt',
       'Total_Trans_Ct', 'Total_Ct_Chng_Q4_Q1', 'Avg_Utilization_Ratio',
       'Credit_Limit_log'],
      dtype='object')
```

In [49]:

```
data.nunique()
```

Out[49]:

```
CLIENTNUM                   10000
Attrition_Flag                  2
Customer_Age                   45
Gender                          2
Dependent_count                 6
Education_Level                 7
Marital_Status                  4
Income_Category                 6
Card_Category                   4
Months_on_book                 44
Total_Relationship_Count        6
Months_Inactive_12_mon          7
Contacts_Count_12_mon           7
Credit_Limit                 6143
Total_Revolving_Bal          1971
Avg_Open_To_Buy              6751
Total_Amt_Chng_Q4_Q1         1155
Total_Trans_Amt              5001
Total_Trans_Ct                126
Total_Ct_Chng_Q4_Q1           827
Avg_Utilization_Ratio         963
Credit_Limit_log             6143
dtype: int64
```

In [59]:

```
data['Gender']
```

Out[59]:

```
3356    1
1291    0
1402    0
8576    1
8864    0
       ..
7455    0
4091    0
6879    0
7264    1
3402    1
Name: Gender, Length: 10000, dtype: int32
```

In [9]:

```
data.dtypes
```

Out[9]:

```
CLIENTNUM
int64
Attrition_Flag
object
Customer_Age
int64
Gender
object
Dependent_count
int64
Education_Level
object
Marital_Status
object
Income_Category
object
Card_Category
object
Months_on_book
int64
Total_Relationship_Count
int64
Months_Inactive_12_mon
int64
Contacts_Count_12_mon
int64
Credit_Limit
float64
Total_Revolving_Bal
int64
Avg_Open_To_Buy
float64
Total_Amt_Chng_Q4_Q1
float64
Total_Trans_Amt
int64
Total_Trans_Ct
int64
Total_Ct_Chng_Q4_Q1
float64
Avg_Utilization_Ratio
float64
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_c
ount_Education_Level_Months_Inactive_12_mon_1    float64
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_c
ount_Education_Level_Months_Inactive_12_mon_2    float64
dtype: object
```

In [10]:

```
data.isna().sum()
```

Out[10]:

```
CLIENTNUM
0
Attrition_Flag
0
Customer_Age
0
Gender
0
Dependent_count
0
Education_Level
0
Marital_Status
0
Income_Category
0
Card_Category
0
Months_on_book
0
Total_Relationship_Count
0
Months_Inactive_12_mon
0
Contacts_Count_12_mon
0
Credit_Limit
0
Total_Revolving_Bal
0
Avg_Open_To_Buy
0
Total_Amt_Chng_Q4_Q1
0
Total_Trans_Amt
0
Total_Trans_Ct
0
Total_Ct_Chng_Q4_Q1
0
Avg_Utilization_Ratio
0
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_c
ount_Education_Level_Months_Inactive_12_mon_1    0
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_c
ount_Education_Level_Months_Inactive_12_mon_2    0
dtype: int64
```

In [11]:

```
data.duplicated()
```

Out[11]:

```
0        False
1        False
2        False
3        False
4        False
         ...  
10122    False
10123    False
10124    False
10125    False
10126    False
Length: 10127, dtype: bool
```

In [12]:

```
data.head()
```

Out[12]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Sta
0	768805383	Existing Customer	45	M	3	High School	Marr
1	818770008	Existing Customer	49	F	5	Graduate	Sin
2	713982108	Existing Customer	51	M	3	Graduate	Marr
3	769911858	Existing Customer	40	F	4	High School	Unknc
4	709106358	Existing Customer	40	M	3	Uneducated	Marr

5 rows × 23 columns

In [13]:

```
data.drop(['Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_
```

In [14]:

```
data.drop(['Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_
```

In [15]:

```
data.head()
```

Out[15]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Sta
0	768805383	Existing Customer	45	M	3	High School	Marr
1	818770008	Existing Customer	49	F	5	Graduate	Sin
2	713982108	Existing Customer	51	M	3	Graduate	Marr
3	769911858	Existing Customer	40	F	4	High School	Unknc
4	709106358	Existing Customer	40	M	3	Uneducated	Marr

5 rows × 21 columns

In [16]:

```
data.shape
```

Out[16]:

```
(10127, 21)
```

Generating Unique Dataset

In [17]:

```
data = data.sample(n = 10000, random_state = 20)
```

In [18]:

```
data.head()
```

Out[18]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_
3356	708871158	Existing Customer	60	M	1	Graduate	
1291	708971583	Existing Customer	38	F	2	Uneducated	
1402	710616183	Existing Customer	46	F	3	Uneducated	
8576	813810333	Attrited Customer	50	M	3	Post-Graduate	M
8864	720939933	Attrited Customer	46	F	4	Unknown	

5 rows × 21 columns

In [19]:

```
data.shape
```

Out[19]:

```
(10000, 21)
```

In [20]:

```
data['Gender'].value_counts()
```

Out[20]:

```
F    5289
M    4711
Name: Gender, dtype: int64
```

In [21]:

```
data['Customer_Age'].value_counts()
```

Out[21]:

```
44    497
49    489
46    485
45    483
47    473
48    467
43    467
50    445
42    419
51    392
53    382
41    373
52    372
40    356
39    329
54    302
38    300
55    275
56    257
37    257
36    220
57    216
35    181
58    155
59    154
34    140
33    126
60    126
32    106
65    100
61     93
62     93
31     90
26     78
30     69
63     64
29     55
64     43
27     32
28     29
67      4
68      2
66      2
73      1
70      1
Name: Customer_Age, dtype: int64
```

In [22]:

```
data.head()
```

Out[22]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_
3356	708871158	Existing Customer	60	M	1	Graduate	
1291	708971583	Existing Customer	38	F	2	Uneducated	
1402	710616183	Existing Customer	46	F	3	Uneducated	
8576	813810333	Attrited Customer	50	M	3	Post-Graduate	M
8864	720939933	Attrited Customer	46	F	4	Unknown	

5 rows × 21 columns

In [23]:
```
data['Customer_Age'].groupby([data['Marital_Status'],data['Education_Level']]).mean()
```

Out[23]:
```
Marital_Status  Education_Level
Divorced        College            44.247059
                Doctorate          46.777778
                Graduate           44.383929
                High School        45.165354
                Post-Graduate      45.975610
                Uneducated         45.896296
                Unknown            45.843750
Married         College            46.393873
                Doctorate          47.875000
                Graduate           46.619863
                High School        47.222937
                Post-Graduate      45.789256
                Uneducated         46.598756
                Unknown            46.635294
Single          College            46.138381
                Doctorate          47.110497
                Graduate           46.364478
                High School        45.643229
                Post-Graduate      45.216931
                Uneducated         46.436207
                Unknown            46.443902
Unknown         College            44.444444
                Doctorate          44.250000
                Graduate           45.769912
                High School        45.276316
                Post-Graduate      45.261905
                Uneducated         46.634615
                Unknown            45.584071
Name: Customer_Age, dtype: float64
```

In [24]:
```
data['Customer_Age'].max()
```

Out[24]:
```
73
```

Visualization - Using Boxplot, Histogram and Scatter Plot

In [25]:
```
data.head()
```

Out[25]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_
3356	708871158	Existing Customer	60	M	1	Graduate	
1291	708971583	Existing Customer	38	F	2	Uneducated	
1402	710616183	Existing Customer	46	F	3	Uneducated	
8576	813810333	Attrited Customer	50	M	3	Post-Graduate	M
8864	720939933	Attrited Customer	46	F	4	Unknown	

5 rows × 21 columns

In [26]:
```
data.tail()
```

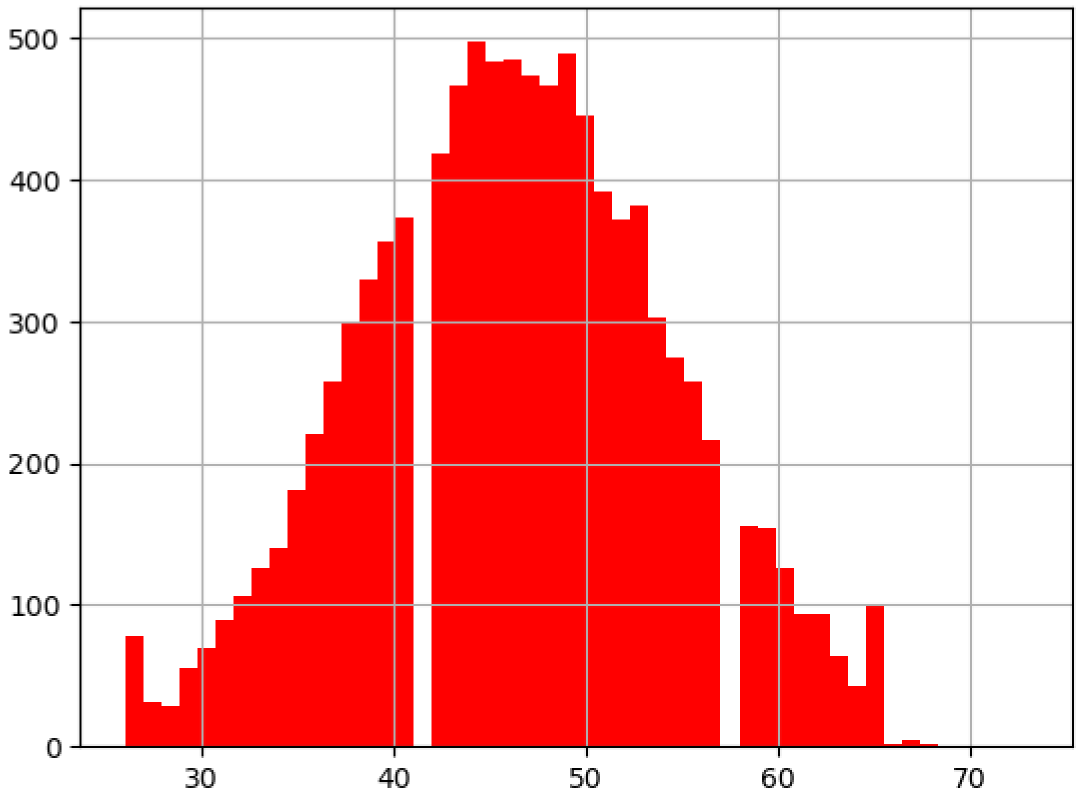
Out[26]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_
7455	770909433	Existing Customer	50	F	2	College	M
4091	710598408	Existing Customer	55	F	0	Graduate	
6879	758875908	Existing Customer	40	F	4	Unknown	
7264	708186933	Attrited Customer	33	M	1	Graduate	
3402	710809833	Existing Customer	40	M	3	Graduate	M

5 rows × 21 columns

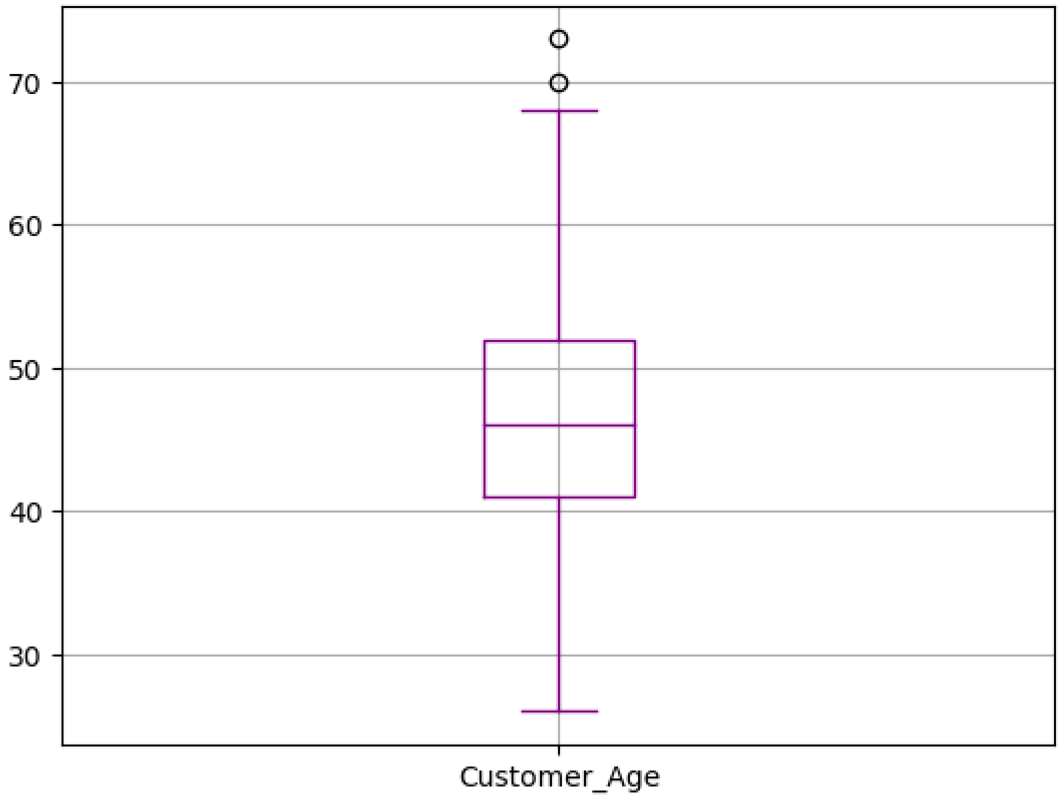
In [27]:
```
data['Customer_Age'].hist(bins=50, color = 'red')
```

Out[27]:
```
<AxesSubplot:>
```



```
In [28]: data.boxplot(column = 'Customer_Age', color = 'purple')
```

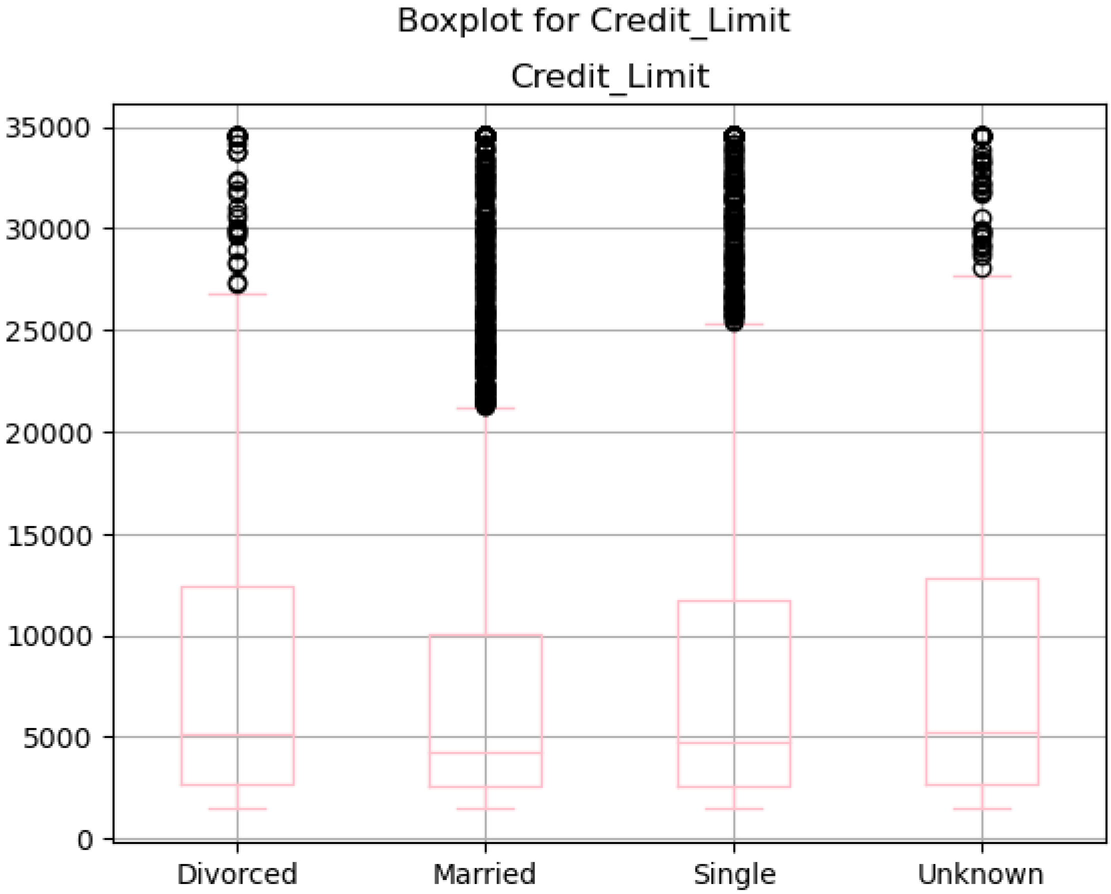
```
Out[28]: <AxesSubplot:>
```



```
In [29]: data.plot(kind = 'box', column = 'Credit_Limit', by = 'Marital_Status', grid = 'True',
```

Out[29]:

```
Credit_Limit    AxesSubplot(0.125,0.11;0.775x0.77)
dtype: object
```

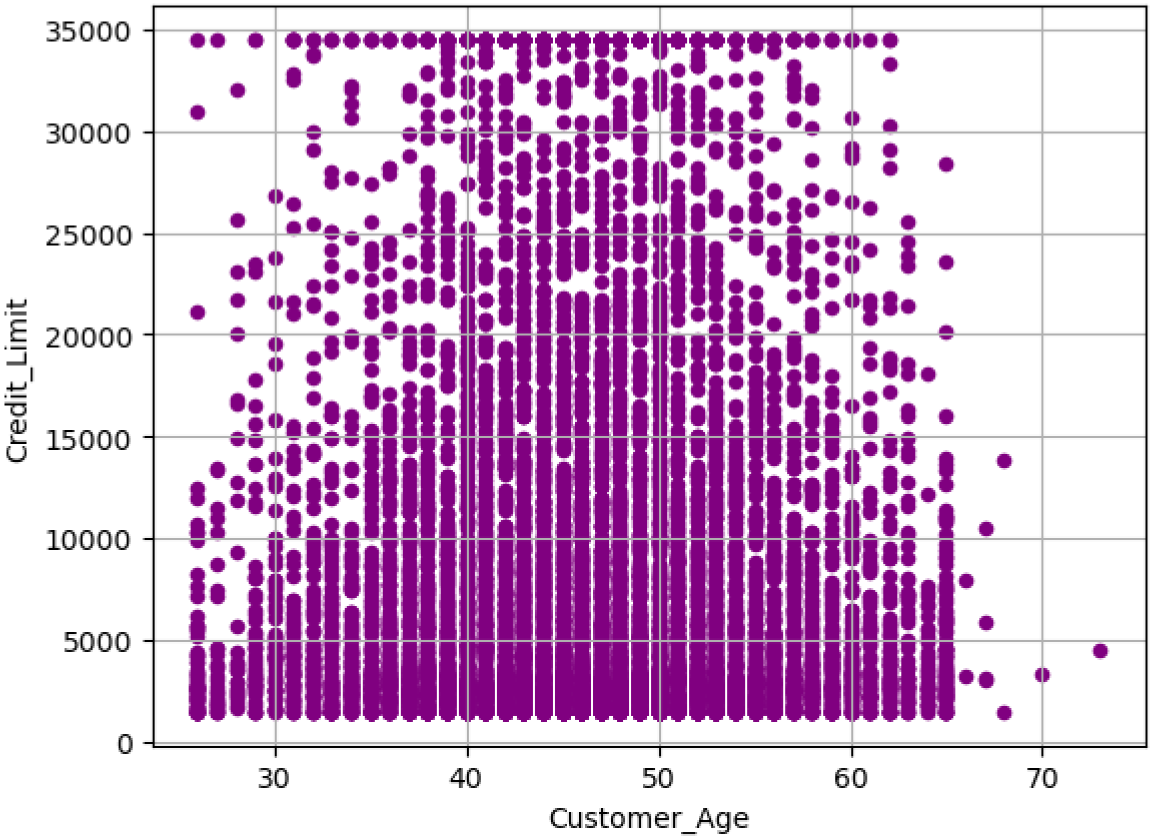


In [30]:

```
data.plot(kind = 'scatter', x = 'Customer_Age', y='Credit_Limit', color = 'purple', gr
```

Out[30]:

```
<AxesSubplot:xlabel='Customer_Age', ylabel='Credit_Limit'>
```



In [31]:
```
data.describe()
```

Out[31]:

	CLIENTNUM	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Mc
count	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	
mean	7.392206e+08	46.320400	2.347200	35.924100	3.814000	
std	3.690477e+07	8.019562	1.299239	8.000846	1.554106	
min	7.080821e+08	26.000000	0.000000	13.000000	1.000000	
25%	7.130319e+08	41.000000	1.000000	31.000000	3.000000	
50%	7.179436e+08	46.000000	2.000000	36.000000	4.000000	
75%	7.731795e+08	52.000000	3.000000	40.000000	5.000000	
max	8.283431e+08	73.000000	5.000000	56.000000	6.000000	

In [50]:
```
data.describe([.10,.20,.30])
```

Out[50]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level
count	1.000000e+04	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000
mean	7.392206e+08	0.83980	46.320400	0.471100	2.347200	3.097100
std	3.690477e+07	0.36681	8.019562	0.499189	1.299239	1.834614
min	7.080821e+08	0.00000	26.000000	0.000000	0.000000	0.000000
10%	7.101620e+08	0.00000	36.000000	0.000000	1.000000	1.000000
20%	7.121213e+08	1.00000	39.000000	0.000000	1.000000	2.000000
30%	7.139554e+08	1.00000	42.000000	0.000000	2.000000	2.000000
50%	7.179436e+08	1.00000	46.000000	0.000000	2.000000	3.000000
max	8.283431e+08	1.00000	73.000000	1.000000	5.000000	6.000000

9 rows × 22 columns

In [32]:
```
data.describe(include = 'all')
```

Out[32]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Mari
count	1.000000e+04	10000	10000.000000	10000	10000.000000	10000	
unique	NaN	2	NaN	2	NaN	7	
top	NaN	Existing Customer	NaN	F	NaN	Graduate	
freq	NaN	8398	NaN	5289	NaN	3098	
mean	7.392206e+08	NaN	46.320400	NaN	2.347200	NaN	
std	3.690477e+07	NaN	8.019562	NaN	1.299239	NaN	
min	7.080821e+08	NaN	26.000000	NaN	0.000000	NaN	
25%	7.130319e+08	NaN	41.000000	NaN	1.000000	NaN	
50%	7.179436e+08	NaN	46.000000	NaN	2.000000	NaN	
75%	7.731795e+08	NaN	52.000000	NaN	3.000000	NaN	
max	8.283431e+08	NaN	73.000000	NaN	5.000000	NaN	

11 rows × 21 columns

In [33]:
```
data.dtypes
```

Out[33]:

```
CLIENTNUM                     int64
Attrition_Flag               object
Customer_Age                  int64
Gender                       object
Dependent_count               int64
Education_Level              object
Marital_Status               object
Income_Category              object
Card_Category                object
Months_on_book                int64
Total_Relationship_Count      int64
Months_Inactive_12_mon        int64
Contacts_Count_12_mon         int64
Credit_Limit                float64
Total_Revolving_Bal           int64
Avg_Open_To_Buy             float64
Total_Amt_Chng_Q4_Q1        float64
Total_Trans_Amt               int64
Total_Trans_Ct                int64
Total_Ct_Chng_Q4_Q1         float64
Avg_Utilization_Ratio       float64
dtype: object
```

In [34]:

```
from sklearn.preprocessing import LabelEncoder
```

In [35]:

```
columns = list(data.select_dtypes(exclude=['int64']))
```

In [36]:

```
le = LabelEncoder()
for i in columns:
    data [i] = le.fit_transform (data[i])
print (columns)
```

```
['Attrition_Flag', 'Gender', 'Education_Level', 'Marital_Status', 'Income_Category',
'Card_Category', 'Credit_Limit', 'Avg_Open_To_Buy', 'Total_Amt_Chng_Q4_Q1', 'Total_Ct
_Chng_Q4_Q1', 'Avg_Utilization_Ratio']
```

In [37]:

```
data.head()
```

Out[37]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_
3356	708871158	1	60	1	1	2	
1291	708971583	1	38	0	2	5	
1402	710616183	1	46	0	3	5	
8576	813810333	0	50	1	3	4	
8864	720939933	0	46	0	4	6	

5 rows × 21 columns

In [38]:

```
data.describe()
```

Out[38]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level
count	1.000000e+04	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000
mean	7.392206e+08	0.83980	46.320400	0.471100	2.347200	3.097100
std	3.690477e+07	0.36681	8.019562	0.499189	1.299239	1.834614
min	7.080821e+08	0.00000	26.000000	0.000000	0.000000	0.000000
25%	7.130319e+08	1.00000	41.000000	0.000000	1.000000	2.000000
50%	7.179436e+08	1.00000	46.000000	0.000000	2.000000	3.000000
75%	7.731795e+08	1.00000	52.000000	1.000000	3.000000	5.000000
max	8.283431e+08	1.00000	73.000000	1.000000	5.000000	6.000000

8 rows × 21 columns

In [39]:

```
data['Gender'].groupby([data['Customer_Age'],data['Dependent_count']]).mean()
```

Out[39]:

```
Customer_Age  Dependent_count
26            0                  0.533333
              1                  0.411765
              2                  0.000000
27            0                  0.550000
              1                  0.181818
                                   ...
67            1                  0.500000
68            0                  1.000000
              1                  1.000000
70            0                  1.000000
73            0                  1.000000
Name: Gender, Length: 203, dtype: float64
```

In [40]:

```
data['Gender'].value_counts()
```

Out[40]:

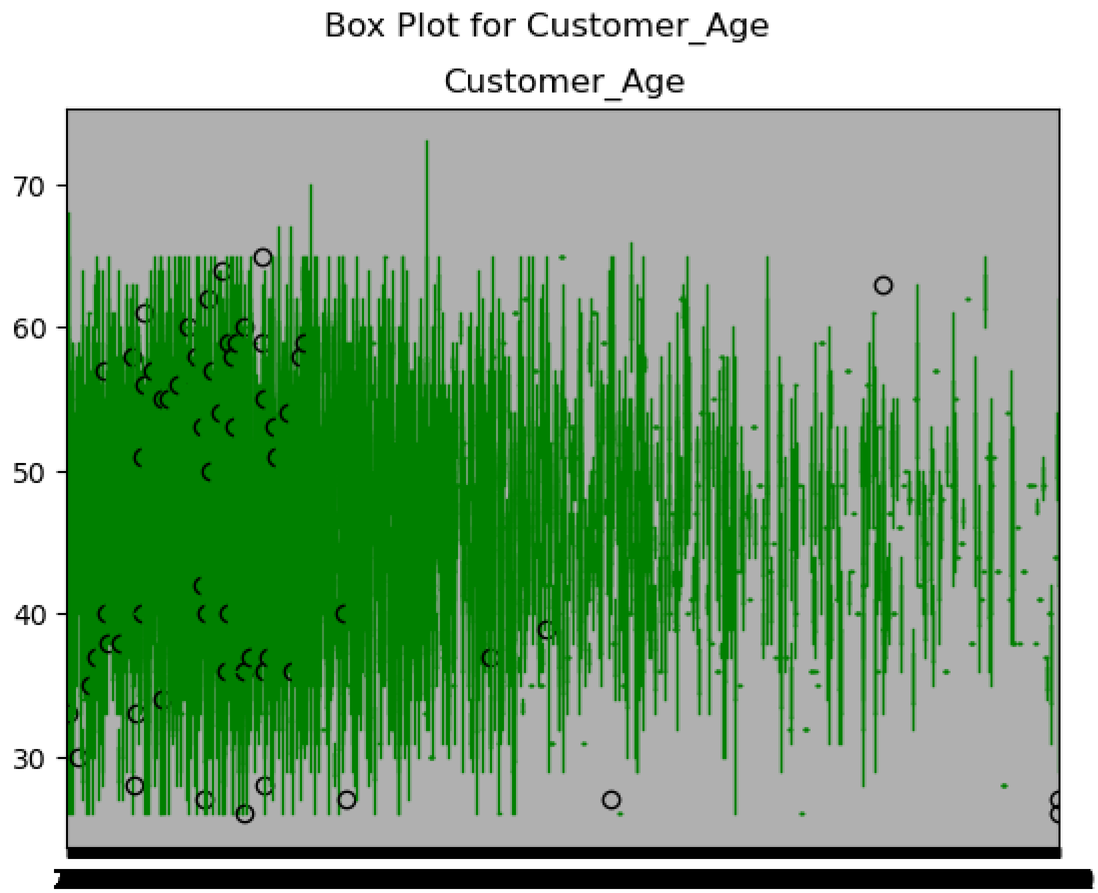
```
0    5289
1    4711
Name: Gender, dtype: int64
```

In [41]:

```
data.plot(kind = 'box', column='Customer_Age', by='Credit_Limit', grid = 'True', color
```

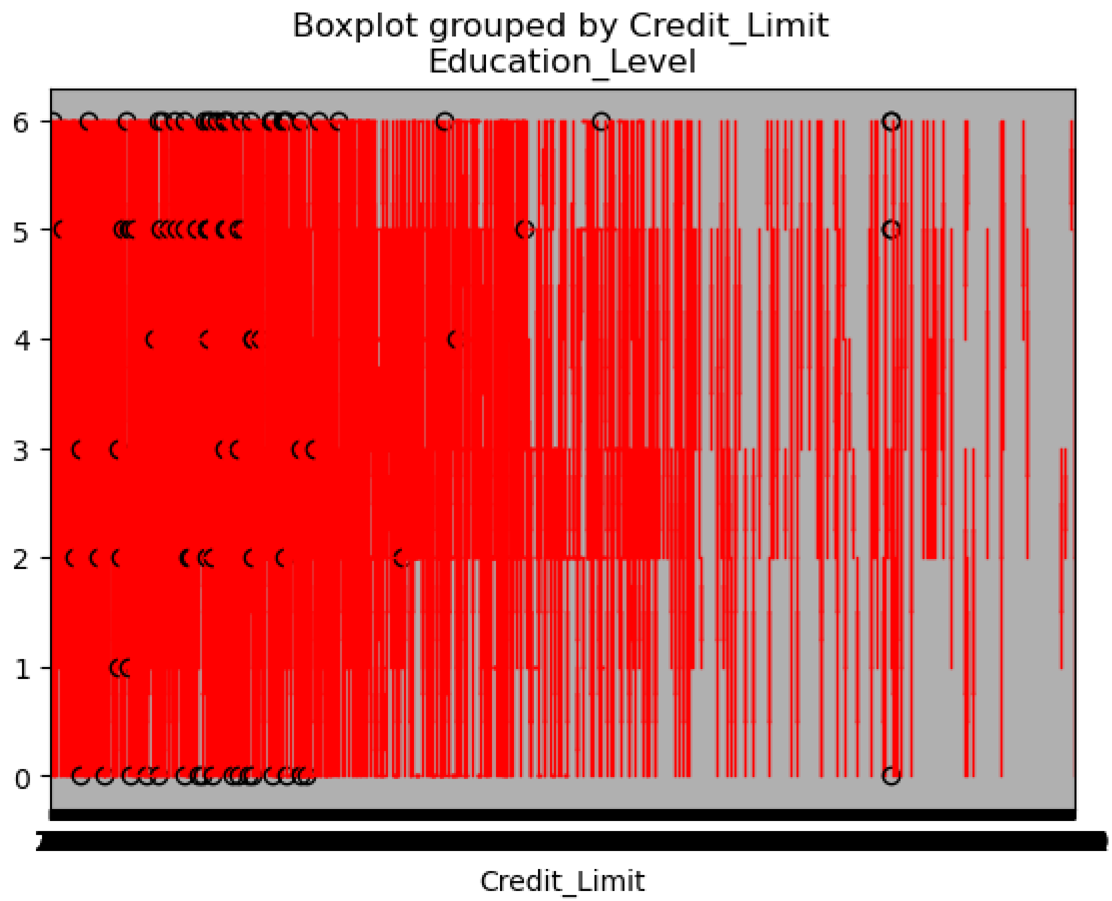
Out[41]:

```
Customer_Age    AxesSubplot(0.125,0.11;0.775x0.77)
dtype: object
```



```
In [42]: data.boxplot(column = 'Education_Level', by = 'Credit_Limit', grid = 'True', color = '
```

```
Out[42]: <AxesSubplot:title={'center':'Education_Level'}, xlabel='Credit_Limit'>
```



```
In [43]: data.head()
```

Out[43]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_
3356	708871158	1	60	1	1	2	
1291	708971583	1	38	0	2	5	
1402	710616183	1	46	0	3	5	
8576	813810333	0	50	1	3	4	
8864	720939933	0	46	0	4	6	

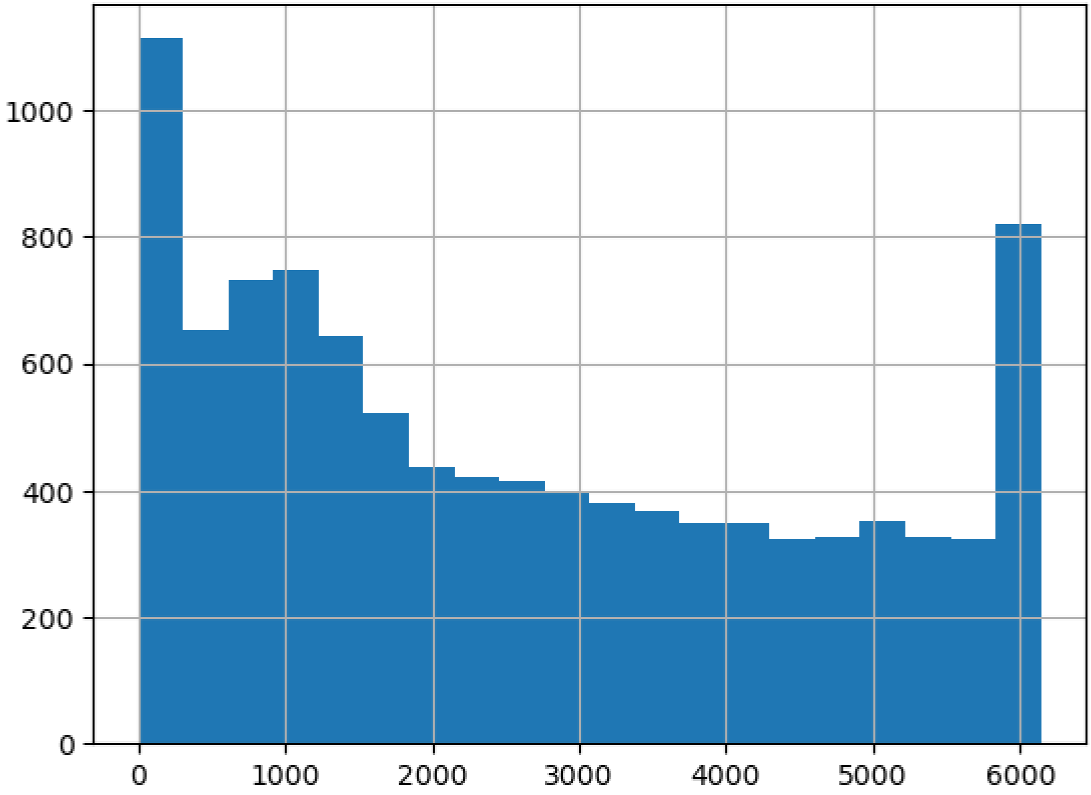
5 rows × 21 columns

```
In [44]: data['Credit_Limit_log'] = np.log(data['Credit_Limit'])
```

```
C:\Users\LenovoX260\anaconda3\lib\site-packages\pandas\core\arraylike.py:397: Runtime
Warning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

```
In [45]: data['Credit_Limit'].hist(bins = 20)
```

```
Out[45]: <AxesSubplot:>
```


1000
800
600
400
200
0
0
1000
2000
3000
4000
5000
6000