In [1]:
```
import pandas as pd
from tabulate import tabulate as tb
import numpy as np
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from IPython.display import display_html
```

In [2]:
```
df = pd.read_csv('C:/Users/sahil/Downloads/customer.csv')
df1 = pd.read_csv('C:/Users/sahil/Downloads/credit_card.csv')
```

In [3]:
```
merged = df1.merge(df,on='Client_Num',how='left')
```

In [4]:
```
merged
```

Out[4]:

	Client_Num	Card_Category	Annual_Fees	Activation_30_Days	Customer_Acq_Cost
0	708082083	Blue	200	0	87
1	708083283	Blue	445	1	108
2	708084558	Blue	140	0	106
3	708085458	Blue	250	1	150
4	708086958	Blue	320	1	106
...	...	...	...	...	...
10103	827695683	Blue	340	1	106
10104	827703258	Blue	395	1	104
10105	827712108	Blue	125	1	107
10106	827888433	Blue	410	0	96
10107	827890758	Blue	100	0	43

10108 rows × 32 columns

In [6]:
```
merged.describe()
```

Out[6]:

	Client_Num	Annual_Fees	Activation_30_Days	Customer_Acq_Cost	current_year
count	1.010800e+04	10108.000000	10108.000000	10108.000000	10108.0
mean	7.390104e+08	291.849525	0.574693	96.254056	2023.0
std	3.673623e+07	118.339384	0.494414	25.768677	0.0
min	7.080821e+08	95.000000	0.000000	40.000000	2023.0
25%	7.130267e+08	195.000000	0.000000	79.000000	2023.0
50%	7.179037e+08	295.000000	1.000000	95.000000	2023.0
75%	7.727989e+08	395.000000	1.000000	112.000000	2023.0
max	8.278908e+08	500.000000	1.000000	172.000000	2023.0

In [5]:
```python
num_col = merged.select_dtypes(include='number')
cat_col = merged.select_dtypes(exclude='number')
```

In [6]:
```python
num_col.describe()
```

Out[6]:

	Client_Num	Annual_Fees	Activation_30_Days	Customer_Acq_Cost	current_year
count	1.010800e+04	10108.000000	10108.000000	10108.000000	10108.0
mean	7.390104e+08	291.849525	0.574693	96.254056	2023.0
std	3.673623e+07	118.339384	0.494414	25.768677	0.0
min	7.080821e+08	95.000000	0.000000	40.000000	2023.0
25%	7.130267e+08	195.000000	0.000000	79.000000	2023.0
50%	7.179037e+08	295.000000	1.000000	95.000000	2023.0
75%	7.727989e+08	395.000000	1.000000	112.000000	2023.0
max	8.278908e+08	500.000000	1.000000	172.000000	2023.0

In [7]:
```python
cat_col
```

Out[7]:

	Card_Category	Week_Start_Date	Week_Num	Qtr	Use Chip	Exp Type	Gender	E
0	Blue	01-01-2023	Week-1	Q1	Chip	Travel	F	
1	Blue	01-01-2023	Week-1	Q1	Swipe	Entertainment	F	
2	Blue	01-01-2023	Week-1	Q1	Chip	Bills	F	
3	Blue	01-01-2023	Week-1	Q1	Online	Grocery	M	
4	Blue	01-01-2023	Week-1	Q1	Swipe	Fuel	M	
...	...	...	...	...	...	...	...	
10103	Blue	24-12-2023	Week-52	Q4	Swipe	Fuel	M	
10104	Blue	24-12-2023	Week-52	Q4	Swipe	Grocery	M	
10105	Blue	24-12-2023	Week-52	Q4	Swipe	Bills	M	
10106	Blue	24-12-2023	Week-52	Q4	Swipe	Bills	F	
10107	Blue	24-12-2023	Week-52	Q4	Chip	Bills	M	

10108 rows × 15 columns

In [8]:
```
merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10108 entries, 0 to 10107
Data columns (total 32 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Client_Num                10108 non-null  int64  
 1   Card_Category             10108 non-null  object 
 2   Annual_Fees               10108 non-null  int64  
 3   Activation_30_Days        10108 non-null  int64  
 4   Customer_Acq_Cost         10108 non-null  int64  
 5   Week_Start_Date           10108 non-null  object 
 6   Week_Num                  10108 non-null  object 
 7   Qtr                       10108 non-null  object 
 8   current_year              10108 non-null  int64  
 9   Credit_Limit              10108 non-null  float64
 10  Total_Revolving_Bal       10108 non-null  int64  
 11  Total_Trans_Amt           10108 non-null  int64  
 12  Total_Trans_Vol           10108 non-null  int64  
 13  Avg_Utilization_Ratio     10108 non-null  float64
 14  Use Chip                  10108 non-null  object 
 15  Exp Type                  10108 non-null  object 
 16  Interest_Earned           10108 non-null  float64
 17  Delinquent_Acc            10108 non-null  int64  
 18  Customer_Age              10108 non-null  int64  
 19  Gender                    10108 non-null  object 
 20  Dependent_Count           10108 non-null  int64  
 21  Education_Level           10108 non-null  object 
 22  Marital_Status            10108 non-null  object 
 23  state_cd                  10108 non-null  object 
 24  Zipcode                   10108 non-null  int64  
 25  Car_Owner                 10108 non-null  object 
 26  House_Owner               10108 non-null  object 
 27  Personal_loan             10108 non-null  object 
 28  contact                   10108 non-null  object 
 29  Customer_Job              10108 non-null  object 
 30  Income                    10108 non-null  int64  
 31  Cust_Satisfaction_Score   10108 non-null  int64  
dtypes: float64(3), int64(14), object(15)
memory usage: 2.5+ MB
```

In [9]:
```
merged.isna().sum()
```

```
Out[9]: Client_Num                0
        Card_Category             0
        Annual_Fees               0
        Activation_30_Days        0
        Customer_Acq_Cost         0
        Week_Start_Date           0
        Week_Num                  0
        Qtr                       0
        current_year              0
        Credit_Limit              0
        Total_Revolving_Bal       0
        Total_Trans_Amt           0
        Total_Trans_Vol           0
        Avg_Utilization_Ratio     0
        Use Chip                  0
        Exp Type                  0
        Interest_Earned           0
        Delinquent_Acc            0
        Customer_Age              0
        Gender                    0
        Dependent_Count           0
        Education_Level           0
        Marital_Status            0
        state_cd                  0
        Zipcode                   0
        Car_Owner                 0
        House_Owner               0
        Personal_loan             0
        contact                   0
        Customer_Job              0
        Income                    0
        Cust_Satisfaction_Score   0
        dtype: int64
```

```
In [10]: num_col
```

Out[10]:

	Client_Num	Annual_Fees	Activation_30_Days	Customer_Acq_Cost	current_year	Cre
0	708082083	200	0	87	2023	
1	708083283	445	1	108	2023	
2	708084558	140	0	106	2023	
3	708085458	250	1	150	2023	
4	708086958	320	1	106	2023	
...	...	...	...	...	...	
10103	827695683	340	1	106	2023	
10104	827703258	395	1	104	2023	
10105	827712108	125	1	107	2023	
10106	827888433	410	0	96	2023	
10107	827890758	100	0	43	2023	

10108 rows × 17 columns

In [13]:
```
num_col[['Activation_30_Days','Delinquent_Acc','current_year','Dependent_Count']].n
```

Out[13]:
```
Activation_30_Days    2
Delinquent_Acc        2
current_year          1
Dependent_Count       6
dtype: int64
```

In [11]:
```
num_sub = num_col.drop(columns=['Client_Num','Activation_30_Days','current_year','D
```

In [15]:
```
num_sub
```

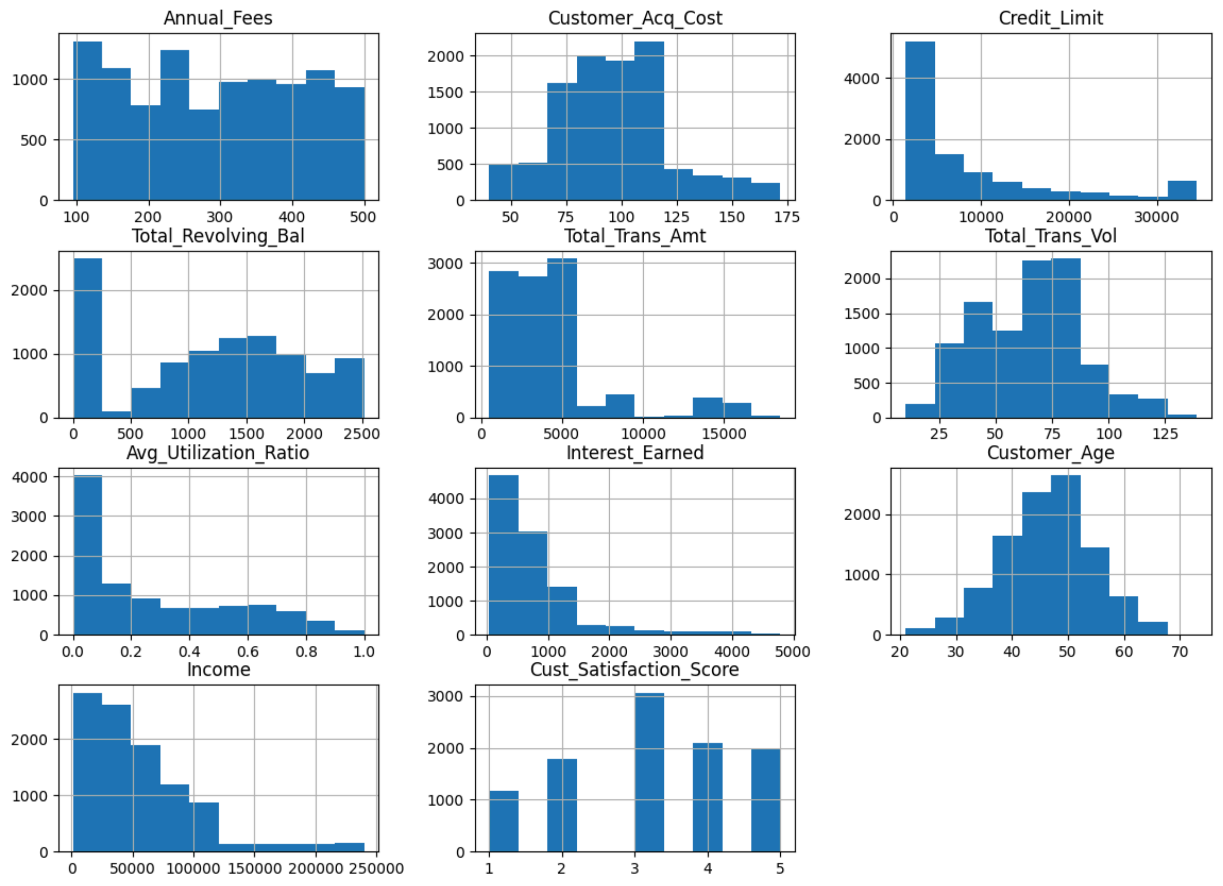
Out[15]:

	Annual_Fees	Customer_Acq_Cost	Credit_Limit	Total_Revolving_Bal	Total_Trans_Amt
0	200	87	3544.0	1661	15149
1	445	108	3421.0	2517	992
2	140	106	8258.0	1771	1447
3	250	150	1438.3	0	3940
4	320	106	3128.0	749	4369
...	...	...	...	...	...
10103	340	106	34516.0	1329	3906
10104	395	104	13426.0	0	4674
10105	125	107	2346.0	1373	4432
10106	410	96	6648.0	2242	2089
10107	100	43	2062.0	1302	3785

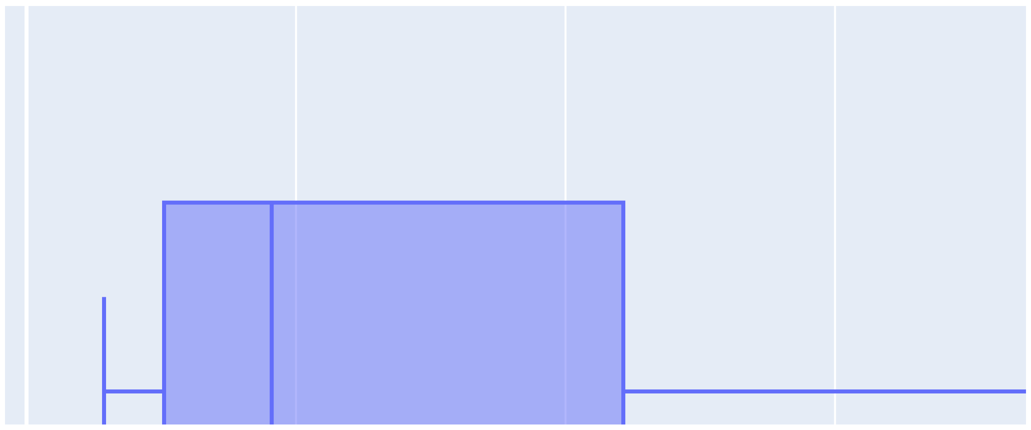
10108 rows × 11 columns

In [16]:
```
num_sub.hist(figsize=(14,10))
plt.axis('off')
```

Out[16]:
```
(0.0, 1.0, 0.0, 1.0)
```

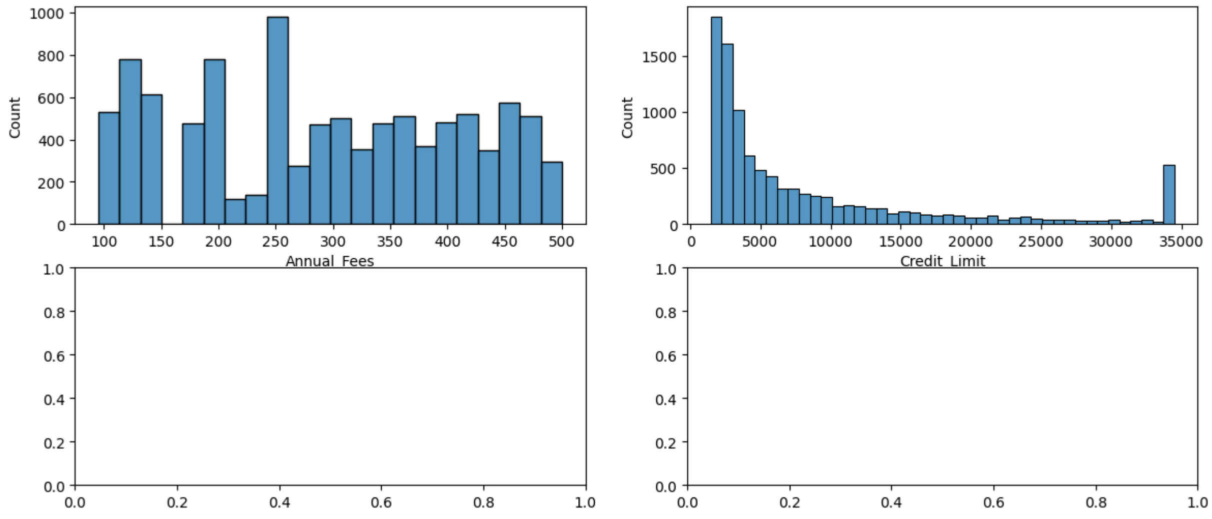


```
In [17]: px.box(num_sub,'Credit_Limit')
```



```
In [18]: fig, ax= plt.subplots(nrows= 2, ncols = 2, figsize= (14,6))
         sns.histplot(num_sub,x='Annual_Fees',ax=ax[0][0])
         sns.histplot(num_sub,x='Credit_Limit',ax=ax[0][1])
```

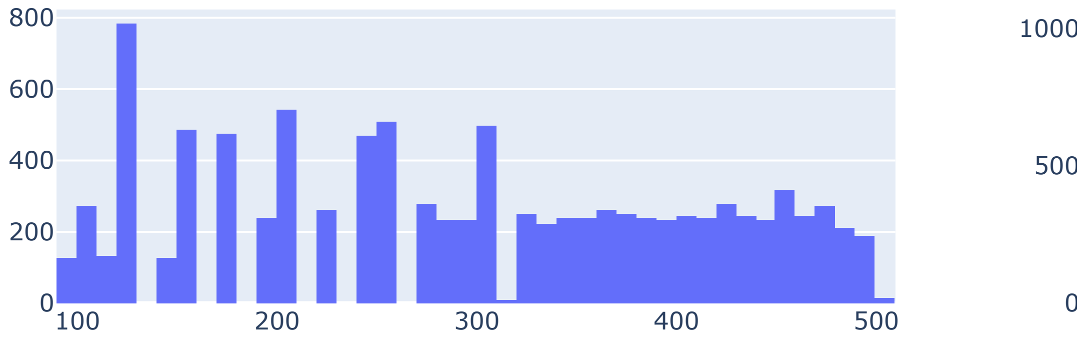
```
Out[18]: <Axes: xlabel='Credit_Limit', ylabel='Count'>
```



```
In [50]: fig = make_subplots(rows=2,cols=2)

         fig.add_trace(
             go.Histogram(x=num_sub['Annual_Fees'], name="yaxis data"),
             row=1, col=1)
         fig.add_trace(
             go.Histogram(x=num_sub['Credit_Limit'], name="yaxis data"),
             row=1, col=2)
         fig.add_trace(
             go.Histogram(x=num_sub['Income'], name="yaxis data"),
             row=2, col=1)

         fig.print_grid
         fig.show()
```



```
In [35]: num_sub[num_sub['Income']> 157000]
```

Out[35]:

	Annual_Fees	Customer_Acq_Cost	Credit_Limit	Total_Revolving_Bal	Total_Trans_Amt
0	200	87	3544.0	1661	15149
9	95	80	11898.0	2517	15798
17	355	78	11463.0	0	14511
75	200	67	29937.0	0	14863
111	470	111	3471.0	0	14381
...	...	...	...	...	...
10023	250	66	3277.0	0	14252
10037	480	140	19033.0	1555	16033
10044	345	72	16453.0	1660	14762
10085	110	106	9431.0	1785	14261
10101	95	106	4107.0	2517	16027

495 rows × 11 columns

In [37]:
```
495/10108*100
```

Out[37]:
```
4.897111199050257
```

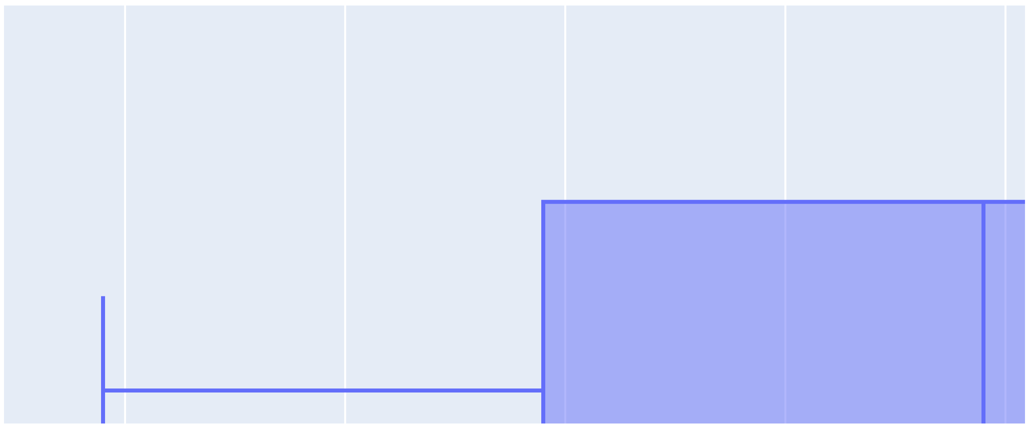
In [26]:
```
num_sub.describe()
```

Out[26]:

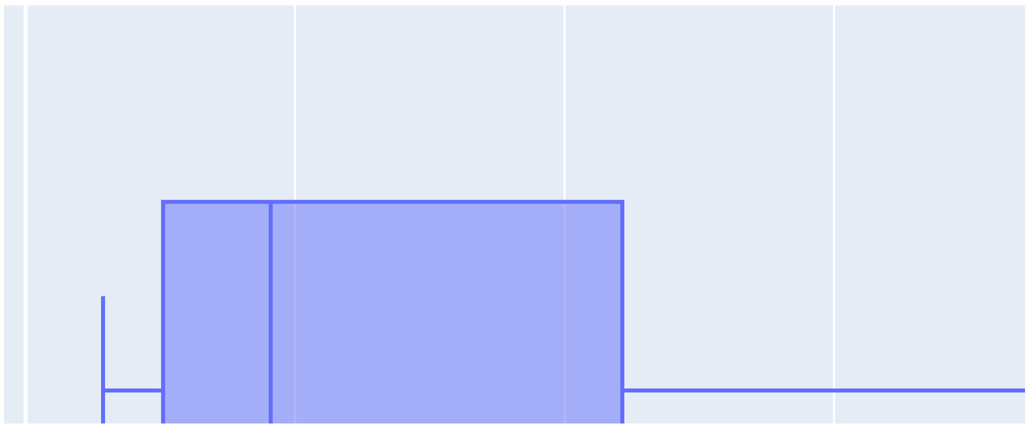
	Annual_Fees	Customer_Acq_Cost	Credit_Limit	Total_Revolving_Bal	Total_Trans_Am
count	10108.000000	10108.000000	10108.000000	10108.000000	10108.00000
mean	291.849525	96.254056	8635.642808	1162.792145	4404.63128
std	118.339384	25.768677	9093.136113	815.160709	3397.91067
min	95.000000	40.000000	1438.300000	0.000000	510.00000
25%	195.000000	79.000000	2552.750000	355.500000	2155.75000
50%	295.000000	95.000000	4549.000000	1276.500000	3899.50000
75%	395.000000	112.000000	11070.250000	1784.000000	4741.00000
max	500.000000	172.000000	34516.000000	2517.000000	18484.00000

In [27]:
```
px.box(num_sub,'Income')
```

```
In [52]: px.box(num_sub,'Annual_Fees')
```

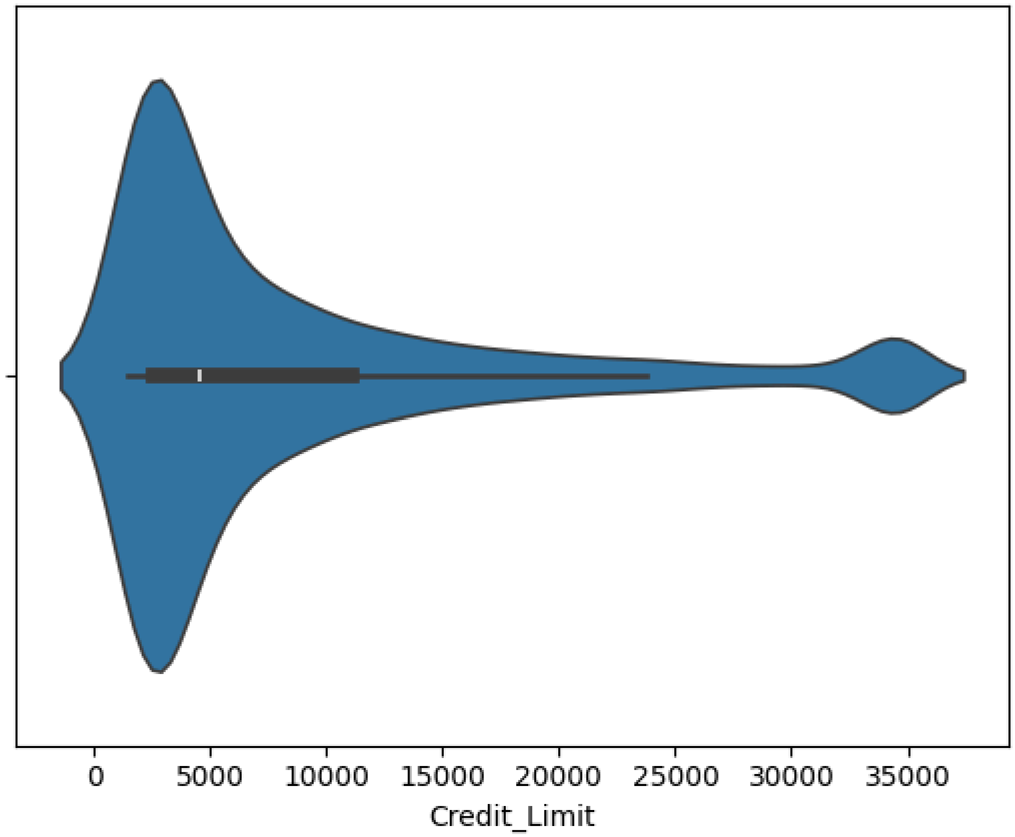


```
In [20]: import plotly
         plotly.offline.init_notebook_mode()
         px.box(num_sub,'Credit_Limit')
```



```
In [15]: sns.violinplot(data=num_sub,x='Credit_Limit')
```

```
Out[15]: <Axes: xlabel='Credit_Limit'>
```



```
In [19]: import plotly
         plotly.offline.init_notebook_mode()
```

```
In [ ]:
```