

In [1]:

```
1 import matplotlib.pyplot as plt
2 import pandas as pd
3 #import seaborn as sns
```

In [2]:

```
1 data=pd.read_csv("bank_loan.csv")
2 data
```

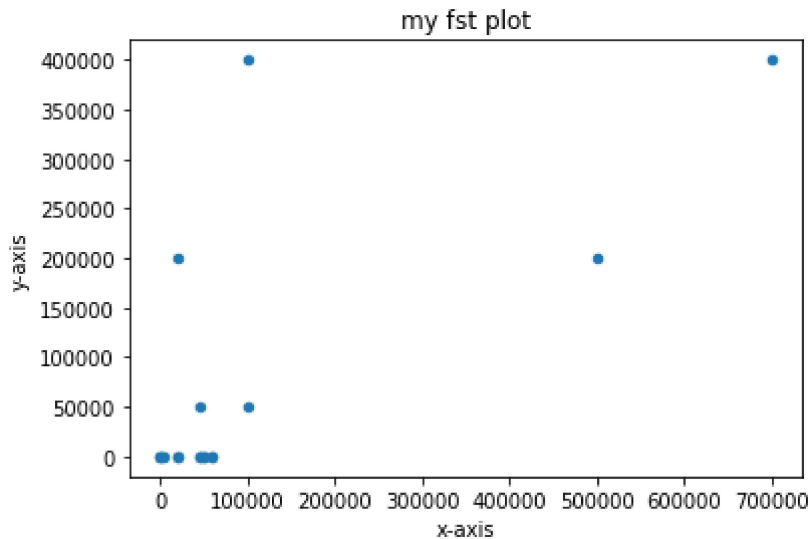
Out[2]:

	id	Balnc	Monthly_Credit	Loan_Eli	Loan_Limit	Act_type
0	1	50000	10000	NO	0	Normal
1	2	60000	10000	NO	0	Normal
2	3	44545	13000	NO	0	Normal
3	4	20000	11000	NO	0	Normal
4	5	100000	20000	YES	50000	Salary
5	6	500000	25000	YES	200000	Salary
6	7	700000	35000	YES	400000	Bussiness
7	8	5000	0	NO	0	Normal
8	9	50000	10000	NO	0	Normal
9	10	60000	10000	NO	0	Normal
10	11	44545	13000	NO	0	Normal
11	12	20000	11000	NO	0	Normal
12	13	44545	20000	YES	50000	Salary
13	14	20000	25000	YES	200000	Salary
14	15	100000	35000	YES	400000	Bussiness
15	16	1000	0	NO	0	Student_account
16	17	500	0	NO	0	Student_account
17	18	0	0	NO	0	Student_account

Scatter Plot (Matplotlib)

In [9]:

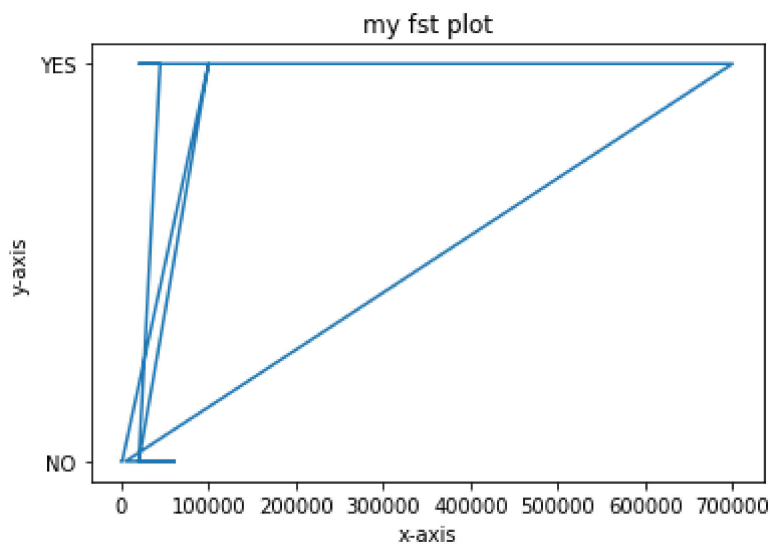
```
1 #data.plot(kind='scatter',x='Balnc',y='Loan_Eli')
2 data.plot(kind='scatter',x='Balnc',y='Loan_Limit')
3 plt.title("my fst plot")
4
5
6 plt.ylabel("y-axis")
7 plt.xlabel("x-axis")
8 plt.show()
9
```



Plot (Matplotlib)

In [10]:

```
1 plt.plot(data['Balnc'],data['Loan_Eli'])
2 plt.title("my fst plot")
3 plt.ylabel("y-axis")
4 plt.xlabel("x-axis")
5 plt.show()
```



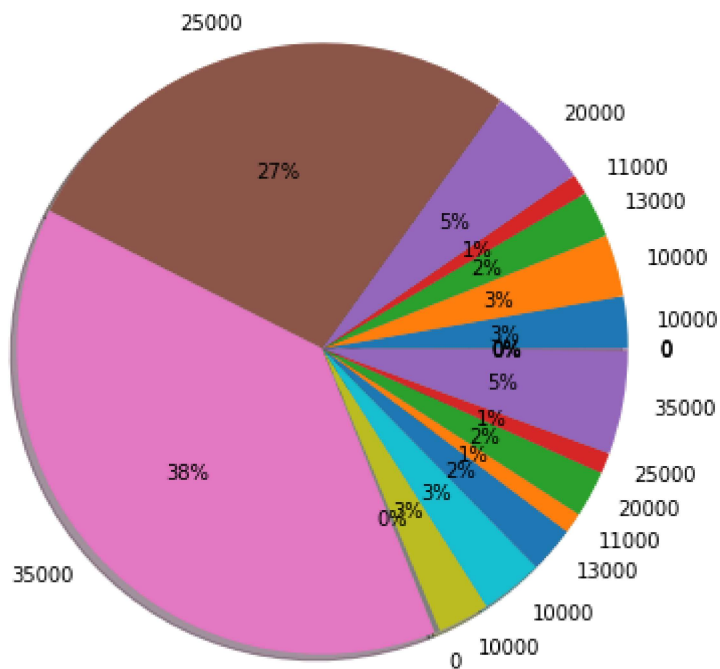
Pie Plot(Matplotlib)

In [13]:

```

1 data
2 # Creating dataset
3 balnc = data['Balnc']
4 data1 = data['Monthly_Credit']
5
6 # Creating plot
7 fig = plt.figure(figsize =(10, 7))
8 plt.pie(balnc, labels = data1,autopct='%.0f%%',shadow = True)
9
10 # show plot
11 plt.show()
12

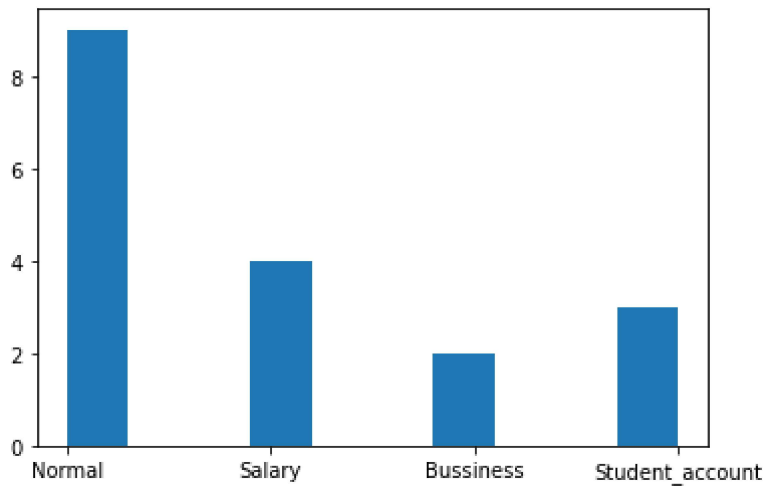
```



Histogram (Matplotlib)

In [16]:

```
1 from matplotlib import pyplot as plt
2
3 data
4 # Y-axis values
5 y = data['Act_type']
6
7 # Function to plot histogram
8 plt.hist(y)
9
10 # Function to show the plot
11 plt.show()
```



pyplot(under matplotlib)

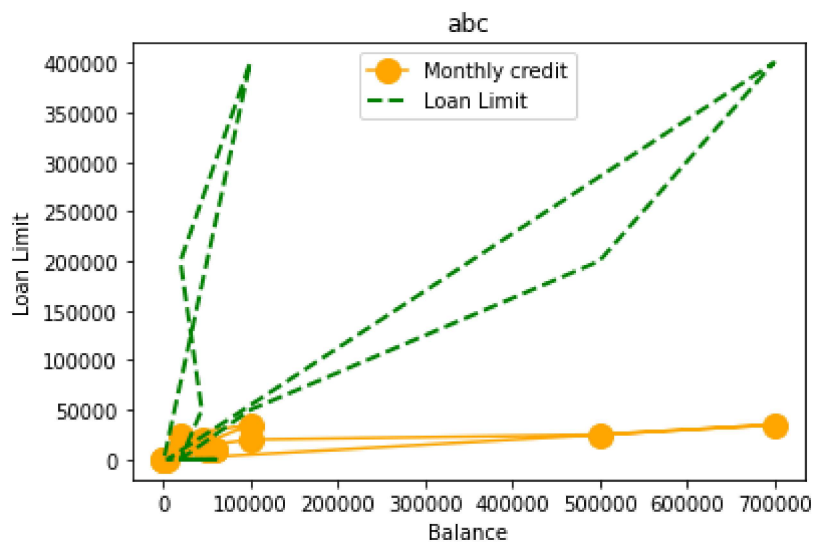
Balance wise Monthly credit and Loan Limit
'.' Point Marker 'o' Circle Marker '+' Plus Marker 's' Square Marker
'D' Diamond Marker 'H' Hexagon Marker

In [20]:

```

1 import matplotlib.pyplot as plt1
2
3 # year contains the x-axis values
4 # and e-india & e-bangladesh
5 # are the y-axis values for plotting
6
7 balnc = data['Balnc']
8 Mon = data['Monthly_Credit']
9 eli = data['Loan_Limit']
10
11
12 #plt1.plot(balnc, Mon, color='orange',label='Monthly credit')
13
14 #plt1.plot(balnc, eli, color='green',label='Loan Limit')
15 plt.plot(balnc, Mon,color='orange', marker='o', markersize=12, label='Monthly credit')
16
17 plt.plot(balnc, eli, color='g',linestyle='dashed', linewidth=2,label='Loan Limit')
18
19
20 # naming of x-axis and y-axis
21 plt1.xlabel('Balance')
22 plt1.ylabel('Loan Limit')
23
24 # naming the title of the plot
25 plt1.title('abc')
26
27 plt1.legend()
28 plt1.show()

```



Multiple Plots Create

In [93]:

```

1  # importing library
2  import matplotlib.pyplot as plt1
3
4  # Some data to display
5  x = data['Balnc']
6  y = data['Monthly_Credit']
7  z = data['Loan_Eli']
8
9  # Creating 2 subplots
10 fig, ax = plt1.subplots(2)
11
12 # Accessing each axes object to plot the data through returned array
13 ax[0].plot(x, y,color="orange") # balance & Monthly credit
14 ax[1].plot(x, z) # balanc & Loan_eli
15 ax[0].set_title("balance & Monthly credit")
16 ax[1].set_title("balanc & Loan_eli")
17

```

Out[93]:

Text(0.5, 1.0, 'balanc & Loan_eli')

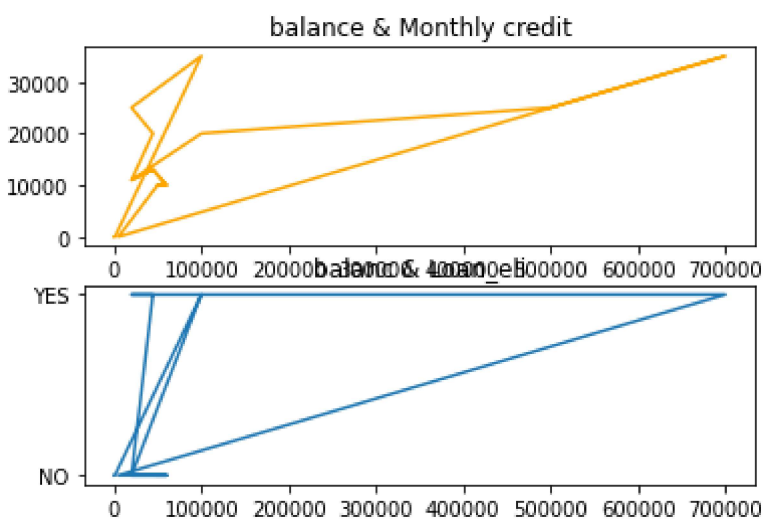


Figure Title

In [97]:

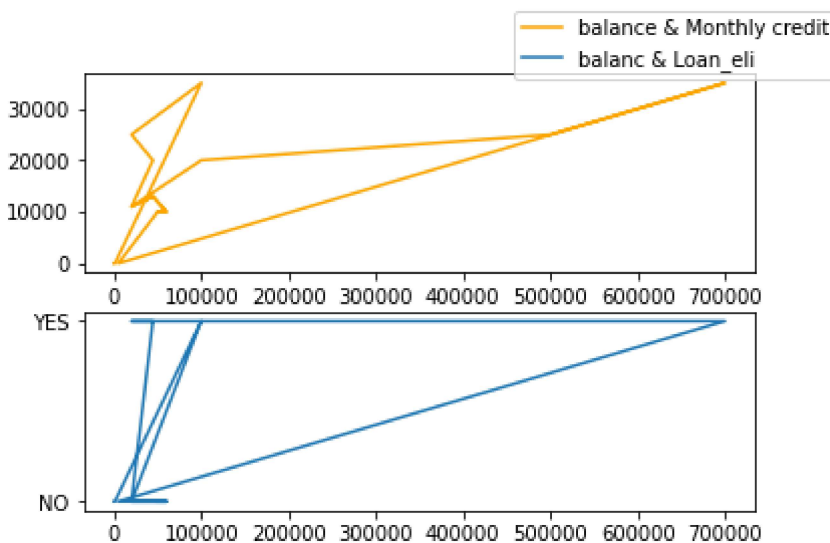
```

1 # importing library
2 import matplotlib.pyplot as plt1
3
4 # Some data to display
5 x = data['Balnc']
6 y = data['Monthly_Credit']
7 z = data['Loan_Eli']
8
9 # Creating 2 subplots
10 fig, ax = plt1.subplots(2)
11
12 # Accessing each axes object to plot the data through returned array
13 x1=ax[0].plot(x, y,color="orange") # balance & Monthly credit
14 y1=ax[1].plot(x, z) # balanc & Loan_eli
15
16 labels = ["balance & Monthly credit", "balanc & Loan_eli"]
17
18 fig.legend([x1, y1], labels=labels,loc="upper right")
19 plt.subplots_adjust(right=0.9)
20 plt.show()

```

C:\Users\EASY\AppData\Local\Temp\ipykernel_4984\2291945108.py:18: UserWarning: You have mixed positional and keyword arguments, some input may be discarded.

```
fig.legend([x1, y1], labels=labels,loc="upper right")
```



Seaborn Library

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions Distribution Plots Pie Chart & Bar Chart Scatter Plots Pair Plots Heat maps

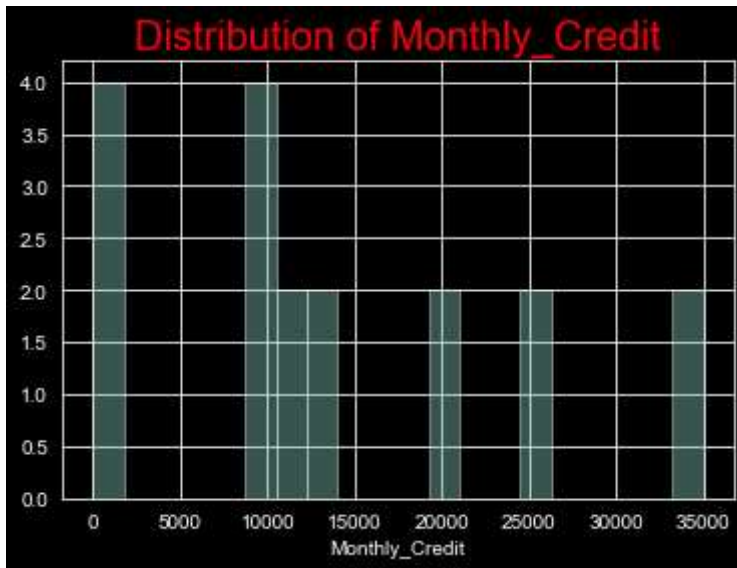
Distribution Plots(using Seaborn)

In [148]:

```
1 # Graph Create (using Seaborn )
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 plt.style.use("dark_background")
5 #sns.distplot(data.Monthly_Credit, color="g")
6 sns.distplot(data.Monthly_Credit, bins=20,kde = False)
7 plt.title("Distribution of Monthly_Credit", fontsize=20, color = 'red')
8 plt.show()
```

C:\Users\EASY\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

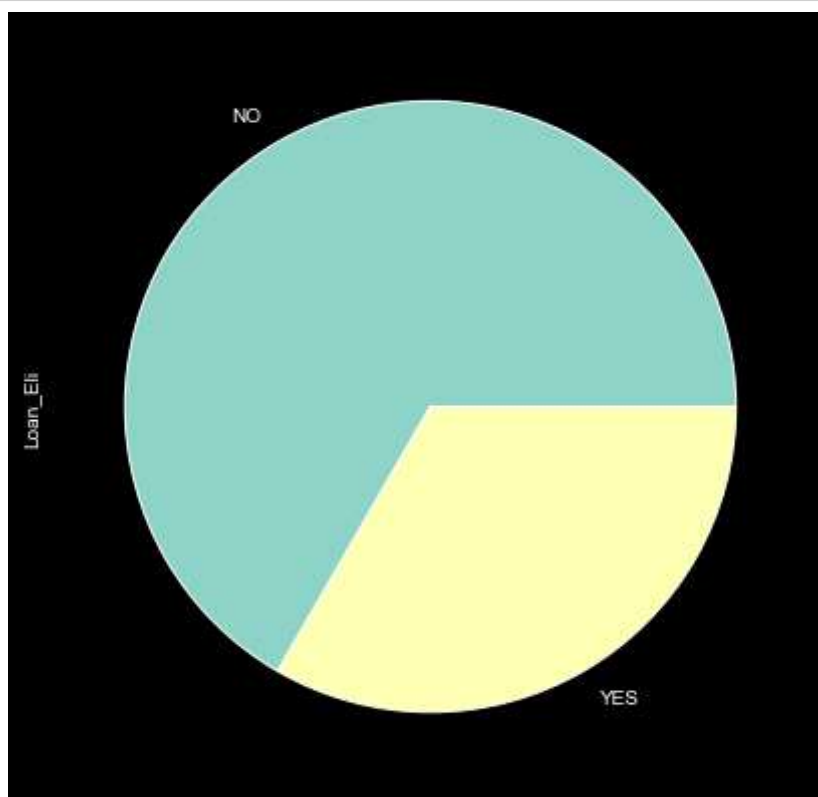
warnings.warn(msg, FutureWarning)



Pie Chart & Bar Chart

In [142]:

```
1 # how many candidate Loan Eligible
2 plt.figure(figsize=[9,7])
3 data['Loan_Eli'].value_counts().plot.pie()
4 plt.show()
5
```

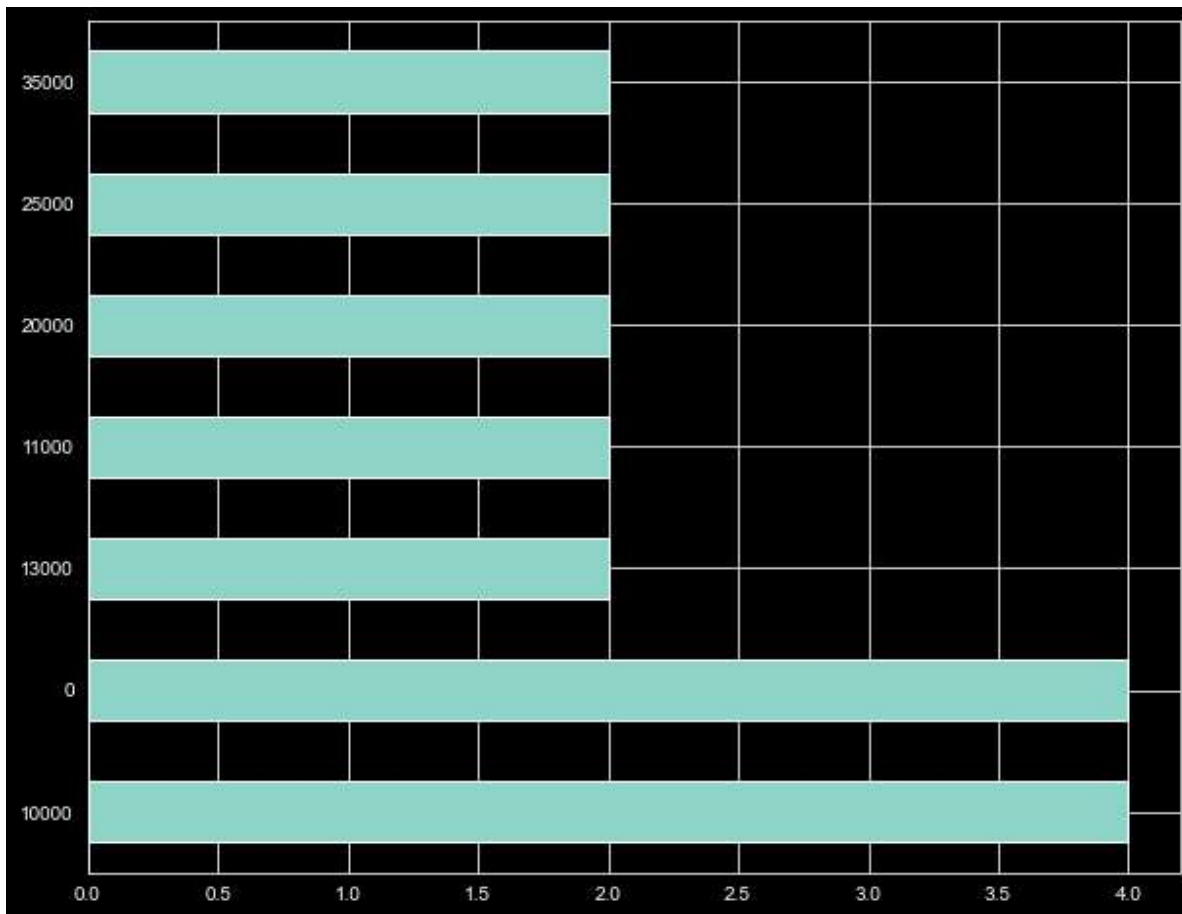


In [147]:

```

1 # bar graph(sea born)
2 plt.figure(figsize=[10,8])
3 data['Monthly_Credit'].value_counts().plot.barh()
4 plt.show()

```



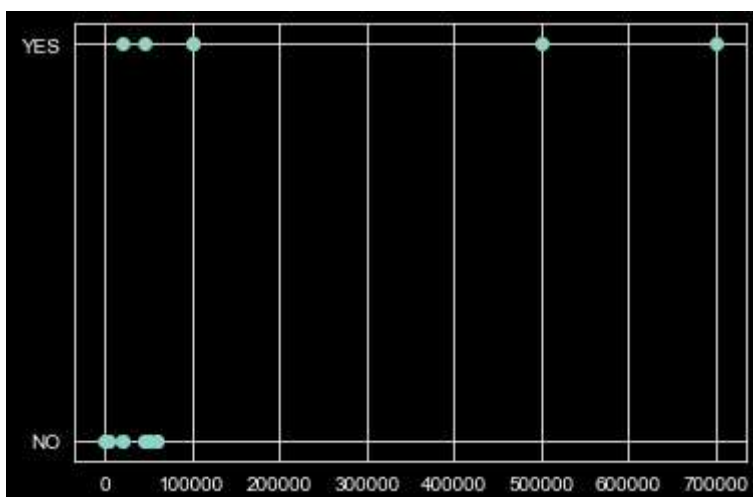
Scatter Plots(using Seaborn)

In [150]:

```

1 plt.scatter(data.Balnc, data.Loan_Eli)
2 plt.show()

```

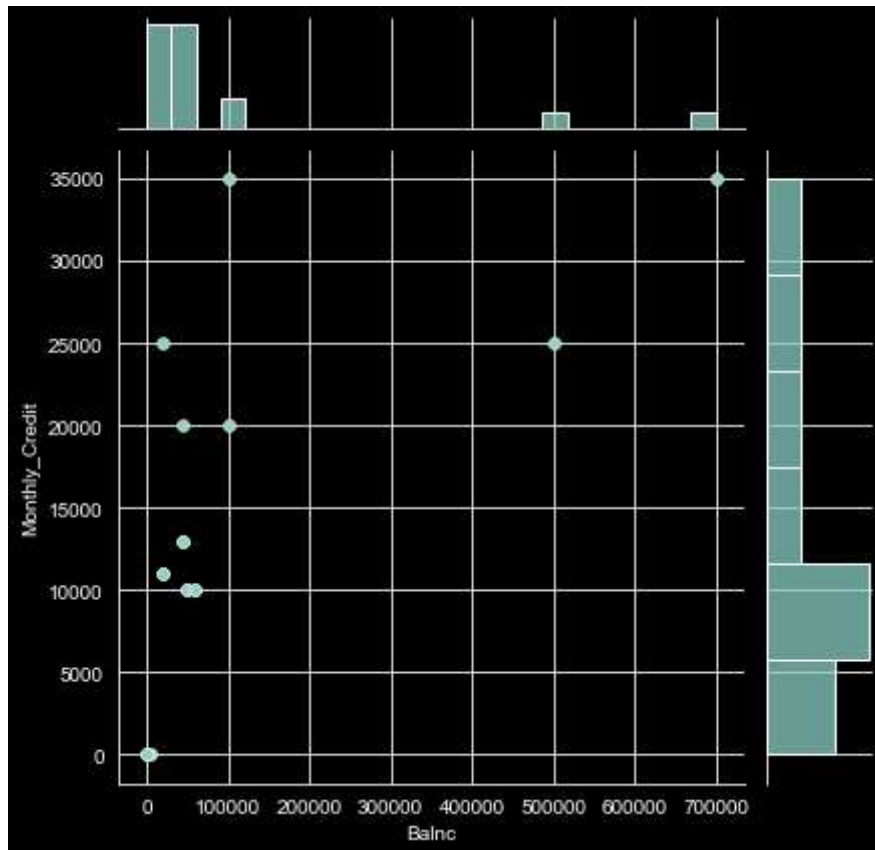


In [155]:

```
1 #scatter plot
2 sns.jointplot(data.Balnc, data.Monthly_Credit)
3 plt.show()
```

C:\Users\EASY\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

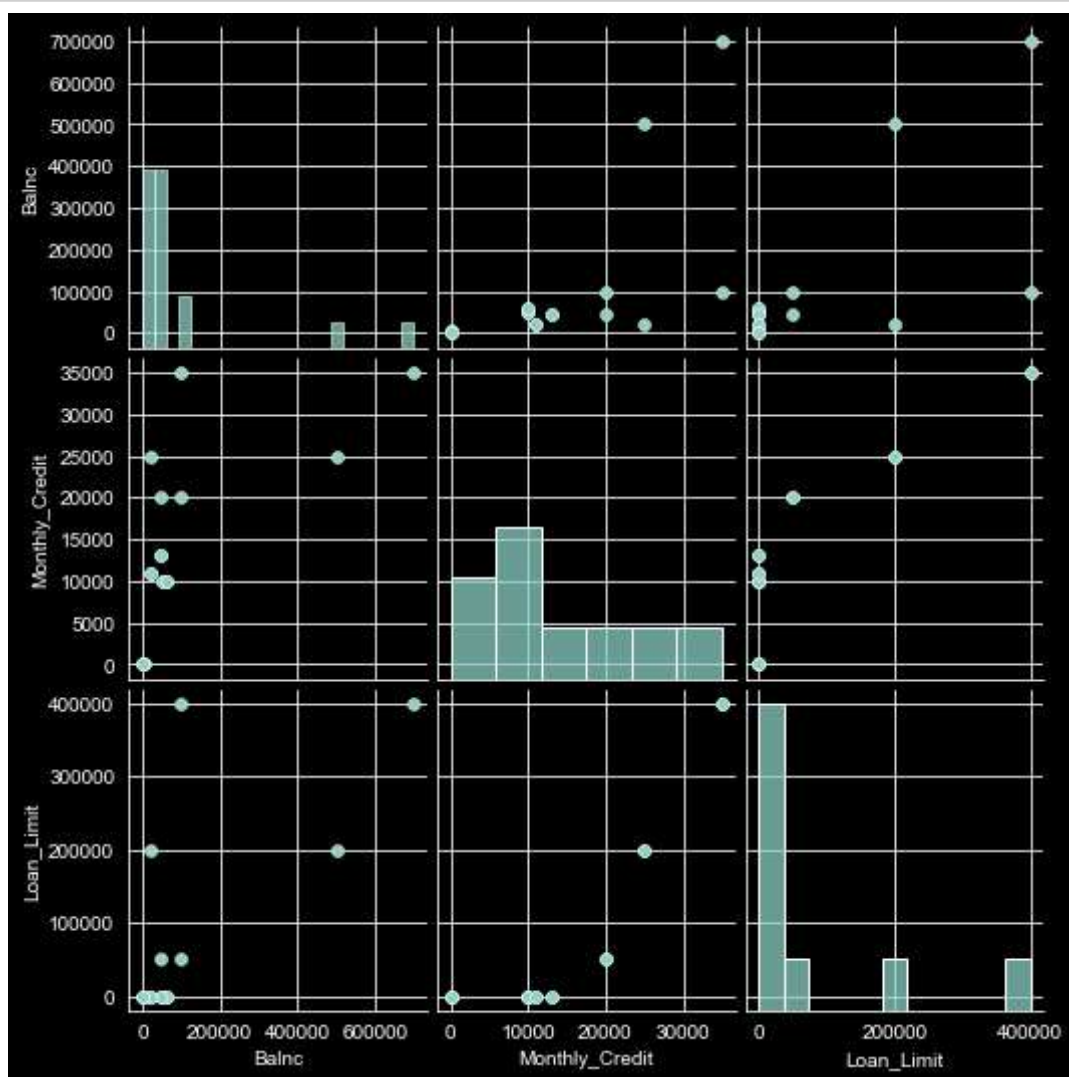
warnings.warn(



Pair Plot

In [157]:

```
1 sns.pairplot(data[['Balnc', 'Monthly_Credit', 'Loan_Eli', 'Loan_Limit']])  
2 plt.show()
```



In [178]:

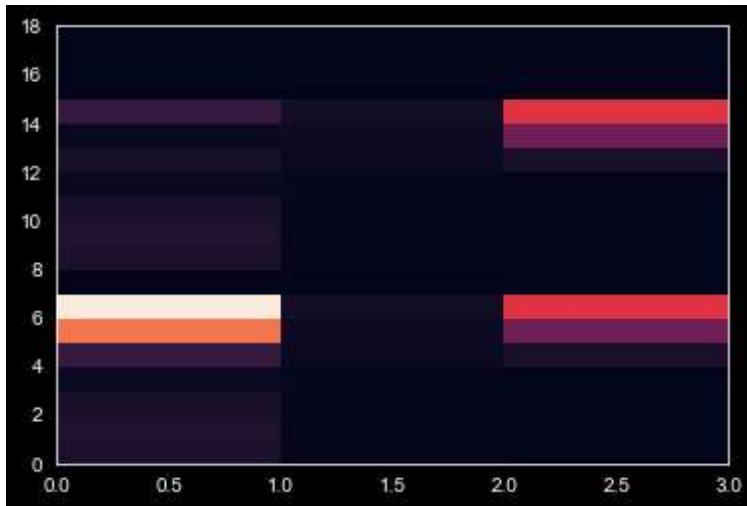
```

1  #Heat Map
2  df = pd.DataFrame(data)
3  # drop perticular column
4  df = df.drop(columns=['id', 'Loan_Eli', 'Act_type'])
5
6  #perticular row Delete
7  #i = df[((df.balnc == 10000) & (df.Monthly_Credit == 10000) & (df.Act_type == 'Normal'))
8  #df.drop(i)
9
10 plt.pcolor(df)
11 plt.show()

```

C:\Users\EASY\AppData\Local\Temp\ipykernel_4984\3329408399.py:5: MatplotlibDeprecationWarning: Auto-removal of grids by pcolor() and pcolormesh() is deprecated since 3.5 and will be removed two minor releases later; please call grid(False) first.

```
plt.pcolor(df)
```



Geographical Present Data

In [3]:

```

1  import folium
2
3  from branca.element import Figure
4  fig=Figure(width=550,height=350)
5  m1=folium.Map(width=550,height=350,location=[28.644800, 77.216721],zoom_start=11,min_zoom=11)
6  fig.add_child(m1)
7
8
9

```

Out[3]:

In [4]:

```
1 # Creating Basemap
2 #fig3=Figure(width=500,height=600)
3 m3=folium.Map(location=[21.7679, 78.8718],tiles='cartodbpositron',zoom_start=5)
4
5 data1=pd.read_csv(r"latlong.csv")
6 df=pd.DataFrame(data1)
7 df['Lat']=df['Lat'].astype('object')
8 df['Lat']=df['Lat'].astype('float')
9 df['Long']=df['Long'].astype('object')
10 df['Long']=df['Long'].astype('float')
11 for lat,long,state in zip(data1['Lat'],data1['Long'],data1['State']):
12     folium.Marker(location=[lat, long],radius=0.8,popup=('<strong>State</strong>:'+str
13 m3
```

Out[4]:

Make this Notebook Trusted to load map: File -> Trust Notebook

In []:

1