In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px

base_credit = pd.read_csv('C:\Users\Master\Desktop\Jupyter\A GREMY\credit_data.csv')
base_credit
```

Out[2]:

	clientid	income	age	loan	default
0	1	66155.925095	59.017015	8106.532131	0
1	2	34415.153966	48.117153	6564.745018	0
2	3	57317.170063	63.108049	8020.953296	0
3	4	42789.524301	45.751972	6103.642260	0
4	5	66952.688845	18.584336	8770.099235	1
...	...	...	...	...	...
1995	1996	59221.044874	48.518179	1926.729397	0
1996	1997	69014.127573	23.162104	3920.179156	0
1997	1998	44211.449292	28.017167	5522.786693	1
1998	1999	43756.056605	63.971796	1622.722598	0
1999	2000	69405.579562	56.152617	7378.833599	0

2000 rows × 5 columns

In [3]:
```
base_credit.head()
```

Out[3]:

	clientid	income	age	loan	default
0	1	66155.925095	59.017015	8106.532131	0
1	2	34415.153966	48.117153	6564.745018	0
2	3	57317.170063	63.108049	8020.953296	0
3	4	42789.524301	45.751972	6103.642260	0
4	5	66952.688845	18.584336	8770.099235	1

In [4]:
```
base_credit.tail()
```

Out[4]:

	clientid	income	age	loan	default
1995	1996	59221.044874	48.518179	1926.729397	0
1996	1997	69014.127573	23.162104	3920.179156	0
1997	1998	44211.449292	28.017167	5522.786693	1
1998	1999	43756.056605	63.971796	1622.722598	0
1999	2000	69405.579562	56.152617	7378.833599	0

In [5]:
```
base_credit.describe()
```

Out[5]:

	clientid	income	age	loan	default
count	2000.000000	2000.000000	1997.000000	2000.000000	2000.000000
mean	1000.500000	45331.600018	40.807559	4444.369385	0.141500
std	577.494589	14326.327119	13.624469	3045.419024	0.348824
min	1.000000	20014.480476	-52.423280	1.377630	0.000000
25%	500.750000	32796.459717	28.990415	1939.709847	0.000000
50%	1000.500000	45789.117313	41.317159	3974.719419	0.000000
75%	1500.250000	57796.261680	52.587040	6492.419625	0.000000
max	2000.000000	69995.685578	63.971796	13796.051239	1.000000

In [6]:

Out[6]:

In [7]:
```
base_credit[base_credit['income']>= 69995.685578]
```

Out[7]:

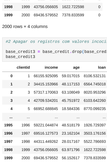
	clientid	income	age	loan	default
422	423	69995.685578	52.738673	2594.370891	0

In [8]:
```
np.unique(base_credit["default"],return_counts=True)
```

Out[8]:
```
(array([0, 1], dtype=int64), array([1717, 283], dtype=int64))
```

In [9]:
```
sns.countplot(x = base_credit['default']);
```

Out[9]:



In [10]:
```
plt.hist(x = base_credit['age']);
```

Out[10]:

In [11]:
```
plt.hist(x = base_credit['income']);
```

Out[11]:

In [12]:
```
plt.hist(x = base_credit['loan']);
```

Out[12]:

In [13]:
```
grafico = px.scatter_matrix(base_credit, dimensions=['age','income','loan'], color='default') #scatter = gráfico de dispersão != não paga dívida, 0 =paga a dí
grafico.show()
```

Out[13]:

In [14]:
```
base_credit.loc[base_credit['age']<0] #filtrar dados dentro da planilha
```

Out[14]:

	clientid	income	age	loan	default
15	16	50501.726689	-28.218361	3077.397432	0
21	22	32197.620701	-52.423280	4344.057136	0
26	27	63267.039508	-36.496976	9595.266299	0

In [15]:
```
base_credit[base_credit['age']>0] #segunda maneira de filtrar dados dentro da planilha
```

Out[15]:

	clientid	income	age	loan	default
15	16	50501.726689	-28.218361	3077.397432	0
21	22	32197.620701	-52.423280	4344.057136	0
26	27	63267.039508	-36.496976	9595.266299	0

In [16]:
```
#maneiras de tratar os valores de idade em negativo
base_credit.drop(15, inplace=True)
base_credit2 = base_credit.drop('age',axis=1)#aqui estamos deletando a coluna 'age' existirá coluna a =>0 # linha
base_credit2
```

Out[16]:

	clientid	income	loan	default
0	1	66155.925095	8106.532131	0
1	2	34415.153966	6564.745018	0
2	3	57317.170063	8020.953296	0
3	4	42789.524301	6103.642260	0
4	5	66952.688845	8770.099235	1
...	...	...	...	...
1995	1996	59221.044874	1926.729397	0
1996	1997	69014.127573	3920.179156	0
1997	1998	44211.449292	5522.786693	1
1998	1999	43756.056605	1622.722598	0
1999	2000	69405.579562	7378.833599	0

2000 rows × 4 columns

In [17]:
```
#2 Apagar os registros com valores inconsistentes
base_credit3 = base_credit.drop(base_credit[base_credit['age']<0].index)
base_credit3
```

Out[17]:

	clientid	income	age	loan	default
0	1	66155.925095	59.017015	8106.532131	0
1	2	34415.153966	48.117153	6564.745018	0
2	3	57317.170063	63.108049	8020.953296	0
3	4	42789.524301	45.751972	6103.642260	0
4	5	66952.688845	18.584336	8770.099235	1
5	6	24904.064140	57.471607	15.496998	0
7	8	44426.358613	26.890132	5722.581961	0
8	9	24590.141984	32.997548	2971.955910	1
9	10	40854.992537	55.496853	4755.825260	0
10	11	20275.977771	39.779279	1409.293071	0
11	12	64119.431072	25.679576	4781.937661	0
12	13	56436.847122	60.473936	9254.244538	0
13	14	61050.340279	35.550344	5993.264959	0
14	15	27267.995458	61.576776	4758.787661	0
16	17	43376.179284	60.848318	3277.737563	0
17	18	20542.350573	61.698571	3157.442290	0
19	20	58087.357546	26.076093	4965.510066	0
20	21	23030.784052	31.791264	1149.110917	0
22	23	32197.620701	40.920000	4344.057136	0
23	24	23529.319414	48.576675	2212.622299	0
24	25	27646.820596	51.976824	4809.921226	0
25	26	65301.904029	40.848922	5465.267986	0
26	27	47461.639123	27.031741	5361.267314	0
28	29	63267.039508	40.920000	9595.266299	0

In [18]:
```
#tratamento de dados faltantes
base_credit.isnull() #estamos verificando se ha celulas sem dados
```

Out[18]:

	clientid	income	age	loan	default
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
1995	False	False	False	False	False
1996	False	False	False	False	False
1997	False	False	False	False	False
1998	False	False	False	False	False
1999	False	False	False	False	False

2000 rows × 5 columns

In [19]:
```
base_credit.isnull().sum() # aqui estamos somando a quantidade de celuas que estão sem dados em cada coluna
```

Out[19]:
```
clientid    0
income      0
age         3
loan        0
default     0
dtype: int64
```

In [20]:
```
base_credit.loc[pd.isnull(base_credit['age'])]
```

Out[20]:

	clientid	income	age	loan	default
28	29	50417.835436	nan	2982.625938	0
30	31	48528.852796	nan	6155.784670	0
32	33	23529.319414	nan	2982.625139	0

In [21]:
```
#aqui estamos alterando a valor das celulas em branco com o valor da media da idade fillna=preencher/valor=nulo(ou seja ele vai preencher apenas em celulas que
base_credit['age'].fillna(base_credit['age'].mean),inplace=True)
```

In [22]:
```
base_credit.loc[pd.isnull(base_credit['age'])]
```

Out[22]:

	clientid	income	age	loan	default

In [23]:
```
#estamos selecionando as celulas apenas para verificar que eles forum preenchidas 'iniv' procura nas celas que estiverem dentro de parenteses
base_credit.loc[base_credit['clientid'].isin([29,31,32])]
```

Out[23]:

	clientid	income	age	loan	default
28	29	50417.835436	nan	2982.625938	0
30	31	48528.852796	nan	6155.784670	0
31	32	23529.319414	nan	2982.625139	0

In [24]:
```
# x normalmente são as variveis previsoras e y são as classe
# iloc é para pegar os valores dos atributos da tabela
x_credit = base_credit.iloc[:,1:4].values.#x primeiro!' inicia que estamos selecionando todas as linhas,'1:4' estamos selcionando income, age e loan
```

Out[24]:
```
array([[ 6.61559251e+04,  5.96170151e+01,  8.10653213e+03],
       [ 3.44151540e+04,  4.81171531e+01,  6.56474501e+03],
       [ 5.73171701e+04,  6.31080496e+01,  8.02095330e+03],
       ...,
       [ 2.31114420e+04,  2.80171600e+01,  5.52278600e+03],
       [ 4.37568566e+04,  3.39717960e+01,  1.62272300e+03],
       [ 6.94395796e+04,  5.61526179e+01,  7.37883300e+03]])
```

In [25]:
```
x_credit[:,0].min(), x_credit[:,0].max()#procurar pela pessoa com menor renda e maior renda
```

Out[25]:
```
(20014.480476000, 69995.685578329)
```

In [26]:
```
x_credit[:,1].max() , x_credit[:,1].max()
```

Out[26]:
```
(69995.685578329, 63.971796441126)
```

In [27]:
```
from sklearn.preprocessing import StandardScaler #aqui estamos padronizando os valores
scaler_credit = StandardScaler()
x_credit = scaler_credit.fit_transform(x_credit)
```

Out[28]:
```
array([[ 0.49395391,  1.38538931,  1.28283642],
       [-0.76217656,  0.94268602,  0.43642695],
       [ 0.83663275,  1.67417136,  1.17471147],
       ...,
       [-0.97122592,  -0.97440519,  0.35420661],
       [ 0.13992836,  -1.72367726,  -0.93575625],
       [ 1.02178506,  -1.14917628,  0.96381038]])
```

In [29]:
```
```