



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

import warnings
warnings.filterwarnings("ignore")

from sklearn.cluster import KMeans
```

```
In [3]: df = pd.read_csv('CreditCard_data.csv')
df
```

Out[3]:

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQU
0	C10001	40.900749	0.818182	95.40	0.00	95.40	0.000000	0.1
1	C10002	3202.467416	0.909091	0.00	0.00	0.00	6442.945483	0.0
2	C10003	2495.148862	1.000000	773.17	773.17	0.00	0.000000	1.0
3	C10004	1666.670542	0.636364	1499.00	1499.00	0.00	205.788017	0.0
4	C10005	817.714335	1.000000	16.00	16.00	0.00	0.000000	0.0
...
8945	C19186	28.493517	1.000000	291.12	0.00	291.12	0.000000	1.0
8946	C19187	19.183215	1.000000	300.00	0.00	300.00	0.000000	1.0
8947	C19188	23.398673	0.833333	144.40	0.00	144.40	0.000000	0.8
8948	C19189	13.457564	0.833333	0.00	0.00	0.00	36.558778	0.0





Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3 (ipykernel) 











Code

8950 rows × 18 columns

In [4]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8950 entries, 0 to 8949
Data columns (total 18 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   CUST_ID                               8950 non-null   object
1   BALANCE                               8950 non-null   float64
2   BALANCE_FREQUENCY                     8950 non-null   float64
3   PURCHASES                             8950 non-null   float64
4   ONEOFF_PURCHASES                      8950 non-null   float64
5   INSTALLMENTS_PURCHASES                8950 non-null   float64
6   CASH_ADVANCE                          8950 non-null   float64
7   PURCHASES_FREQUENCY                   8950 non-null   float64
8   ONEOFF_PURCHASES_FREQUENCY            8950 non-null   float64
9   PURCHASES_INSTALLMENTS_FREQUENCY      8950 non-null   float64
10  CASH_ADVANCE_FREQUENCY                 8950 non-null   float64
11  CASH_ADVANCE_TRX                       8950 non-null   int64
12  PURCHASES_TRX                         8950 non-null   int64
13  CREDIT_LIMIT                           8949 non-null   float64
14  PAYMENTS                               8950 non-null   float64
15  MINIMUM_PAYMENTS                       8637 non-null   float64
16  PRC_FULL_PAYMENT                       8950 non-null   float64
17  TENURE                                8950 non-null   int64
dtypes: float64(14), int64(3), object(1)
memory usage: 1.2+ MB
```


In [5]: df.drop('CUST_ID',axis=1,inplace = True)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3 (ipykernel) 











Code

```

16 PRC_FULL_PAYMENT      8950 non-null float64
17 TENURE                8950 non-null int64
dtypes: float64(14), int64(3), object(1)
memory usage: 1.2+ MB

```

```
In [5]: df.drop('CUST_ID',axis=1,inplace = True)
```

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

```
Out[6]:
```

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	C
count	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000	
mean	1564.474828	0.877271	1003.204834	592.437371	411.067645	978.871112	0.490351	
std	2081.531879	0.236904	2136.634782	1659.887917	904.338115	2097.163877	0.401371	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	128.281915	0.888889	39.635000	0.000000	0.000000	0.000000	0.083333	
50%	873.385231	1.000000	361.280000	38.000000	89.000000	0.000000	0.500000	
75%	2054.140036	1.000000	1110.130000	577.405000	468.637500	1113.821139	0.916667	
max	19043.138560	1.000000	49039.570000	40761.250000	22500.000000	47137.211760	1.000000	

```
In [7]: df.duplicated().sum()
```

```
Out[7]: 0
```

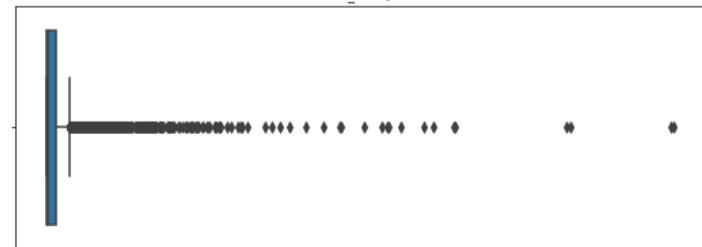
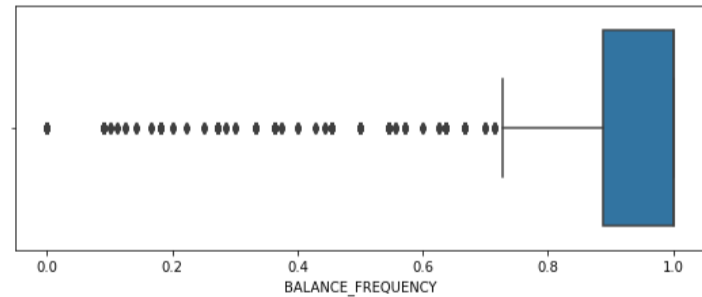
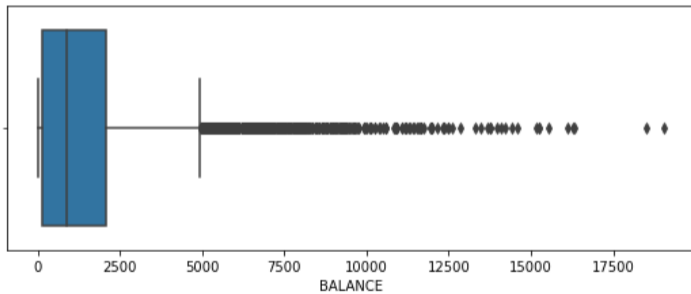
```
In [8]: plt.figure(figsize=(20,35))
for i, col in enumerate(df.columns):
    ax = plt.subplot(9, 2, i+1)
    sns.boxplot(df[col])
```

```
In [7]: df.duplicated().sum()
```

```
Out[7]: 0
```

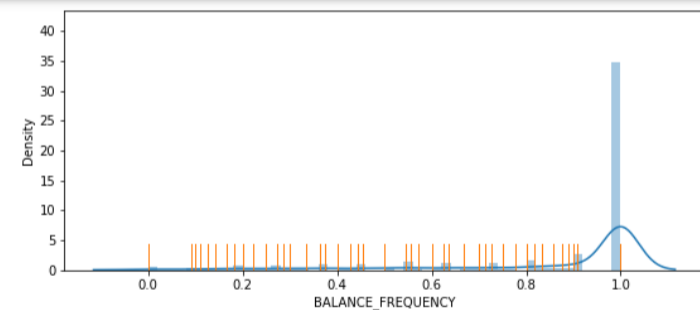
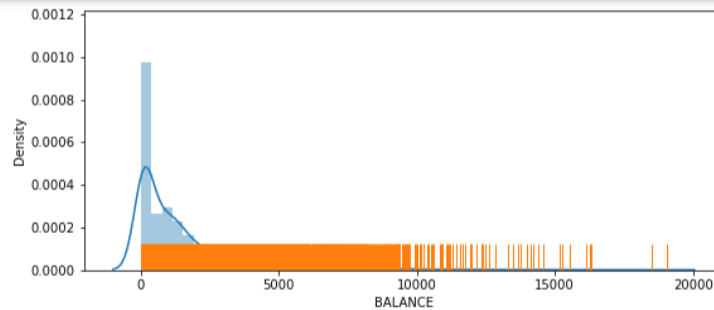
```
In [8]: plt.figure(figsize=(20,35))
for i, col in enumerate(df.columns):
    ax = plt.subplot(9, 2, i+1)
    sns.boxplot(df[col])
    plt.xlabel(col)

plt.show()
```





```
In [9]: i=1
plt.figure(figsize= (20,40))
for col in df.columns:
    plt.subplot(9,2,i)
    sns.distplot(df[col])#This function combines the matplotlib hist function with the seaborn kdeplot()
    sns.rugplot(df[col], height=.1)#This function is intended to complement other plots by showing the location of individual observations
    i=i+1
plt.show()
```




```
In [10]: df.isnull().sum()
```



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3 (ipykernel) 











Code

In [10]: df.isnull().sum()

```
Out[10]:
```

BALANCE	0
BALANCE_FREQUENCY	0
PURCHASES	0
ONEOFF_PURCHASES	0
INSTALLMENTS_PURCHASES	0
CASH_ADVANCE	0
PURCHASES_FREQUENCY	0
ONEOFF_PURCHASES_FREQUENCY	0
PURCHASES_INSTALLMENTS_FREQUENCY	0
CASH_ADVANCE_FREQUENCY	0
CASH_ADVANCE_TRX	0
PURCHASES_TRX	0
CREDIT_LIMIT	1
PAYMENTS	0
MINIMUM_PAYMENTS	313
PRC_FULL_PAYMENT	0
TENURE	0

dtype: int64

```
In [11]: def show_null(data):
sum_null = data.isnull().sum().sort_values(ascending = False)
count_percent = (data.isnull().sum() / data.isnull().count()*100).sort_values(ascending = False)
return pd.concat([sum_null, count_percent], axis=1, keys = ['sum_null', 'count_percent'])
```

In [12]: show_null(df).style.background_gradient(cmap = 'Spectral')

```
Out[12]:
```

	sum_null	count_percent
--	----------	---------------



```
In [12]: show_null(df).style.background_gradient(cmap = 'Spectral')
```

```
Out[12]:
```

	sum_null	count_percent
MINIMUM_PAYMENTS	313	3.497207
CREDIT_LIMIT	1	0.011173
BALANCE	0	0.000000
CASH_ADVANCE_FREQUENCY	0	0.000000
PRC_FULL_PAYMENT	0	0.000000
PAYMENTS	0	0.000000
PURCHASES_TRX	0	0.000000
CASH_ADVANCE_TRX	0	0.000000
PURCHASES_INSTALLMENTS_FREQUENCY	0	0.000000
BALANCE_FREQUENCY	0	0.000000
ONEOFF_PURCHASES_FREQUENCY	0	0.000000
PURCHASES_FREQUENCY	0	0.000000
CASH_ADVANCE	0	0.000000
INSTALLMENTS_PURCHASES	0	0.000000
ONEOFF_PURCHASES	0	0.000000
PURCHASES	0	0.000000
TENURE	0	0.000000


```
In [13]: # missing values aren't alot, so i drop them.  
df.dropna(inplace = True)
```



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3 (ipykernel) 











Code

TENURE 0 0.000000

```
In [13]: # missing values aren't alot, so i drop them.
df.dropna(inplace = True)
df.reset_index( inplace = True)
df.drop('index',axis=1,inplace = True)
df
```

Out[13]:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ON
0	40.900749	0.818182	95.40	0.00	95.40	0.000000	0.166667	
1	3202.467416	0.909091	0.00	0.00	0.00	6442.945483	0.000000	
2	2495.148862	1.000000	773.17	773.17	0.00	0.000000	1.000000	
3	817.714335	1.000000	16.00	16.00	0.00	0.000000	0.083333	
4	1809.828751	1.000000	1333.28	0.00	1333.28	0.000000	0.666667	
...
8631	5.871712	0.500000	20.90	20.90	0.00	0.000000	0.166667	
8632	28.493517	1.000000	291.12	0.00	291.12	0.000000	1.000000	
8633	23.398673	0.833333	144.40	0.00	144.40	0.000000	0.833333	
8634	13.457564	0.833333	0.00	0.00	0.00	36.558778	0.000000	
8635	372.708075	0.666667	1093.25	1093.25	0.00	127.040008	0.666667	

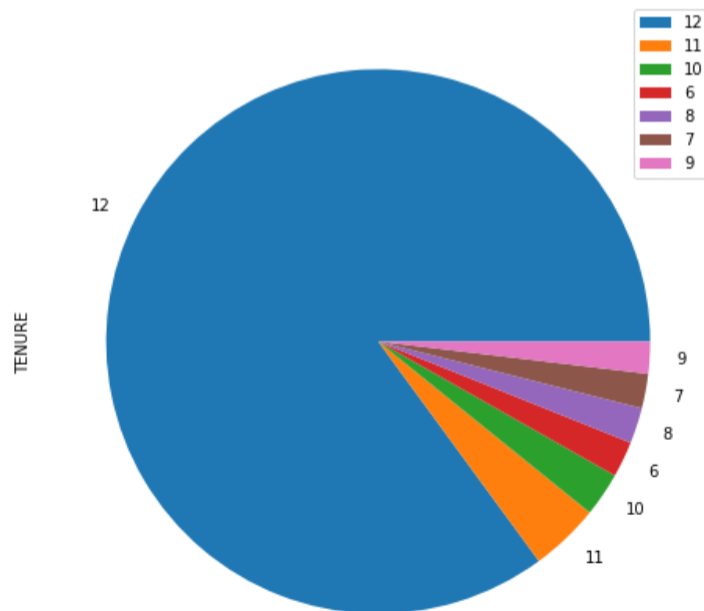
8636 rows × 17 columns

análisis con los datos limpios



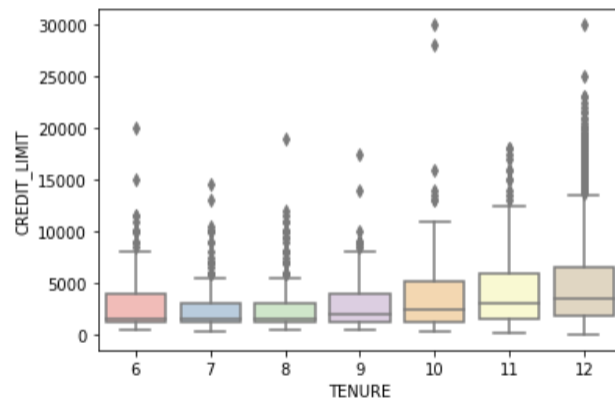
analisis con los datos limpios

```
In [14]: df_TENURE=pd.DataFrame(df['TENURE'].value_counts())
plot = df_TENURE.plot.pie(y='TENURE', figsize=(8, 8))
```



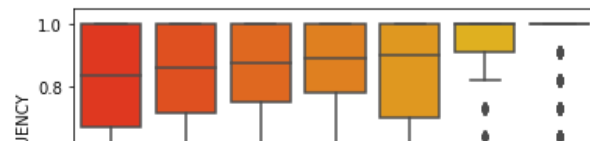
```
In [16]: sns.boxplot(x = 'TENURE', y = 'CREDIT_LIMIT', data = df,palette='Pastel1')
```

```
Out[16]: <AxesSubplot:xlabel='TENURE', ylabel='CREDIT_LIMIT'>
```



```
In [17]: sns.boxplot(x = 'TENURE', y = 'BALANCE_FREQUENCY', data = df,palette='autumn')
```

```
Out[17]: <AxesSubplot:xlabel='TENURE', ylabel='BALANCE_FREQUENCY'>
```





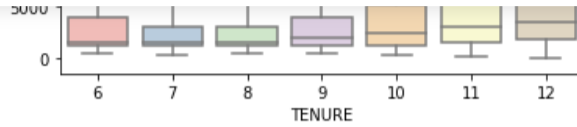
File Edit View Insert Cell Kernel Widgets Help

Trusted



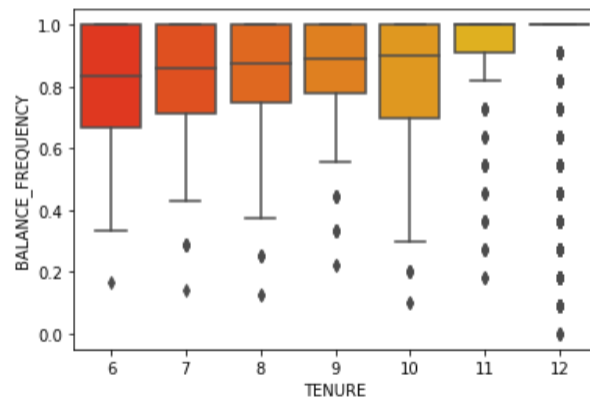
Python 3 (ipykernel)

Code



```
In [17]: sns.boxplot(x = 'TENURE', y = 'BALANCE_FREQUENCY', data = df,palette='autumn')
```

```
Out[17]: <AxesSubplot:xlabel='TENURE', ylabel='BALANCE_FREQUENCY'>
```



```
In [18]: sns.scatterplot(x='CREDIT_LIMIT', y='PRC_FULL_PAYMENT', data=df,color='purple')
```

```
Out[18]: <AxesSubplot:xlabel='CREDIT_LIMIT', ylabel='PRC_FULL_PAYMENT'>
```

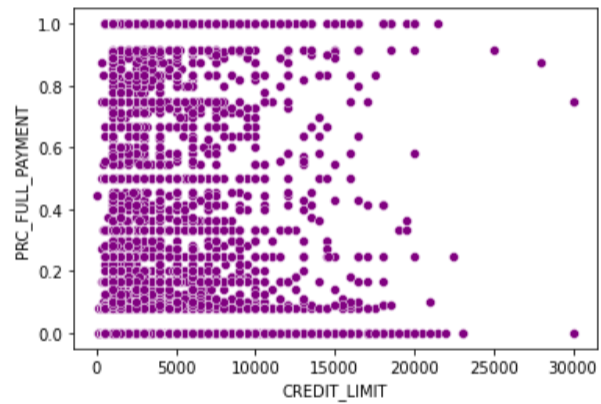




TENURE

```
In [18]: sns.scatterplot(x='CREDIT_LIMIT', y='PRC_FULL_PAYMENT', data=df,color='purple')
```

```
Out[18]: <AxesSubplot:xlabel='CREDIT_LIMIT', ylabel='PRC_FULL_PAYMENT'>
```



```
In [19]: sns.scatterplot(x='BALANCE', y='PURCHASES', data=df,color='purple')
```

```
Out[19]: <AxesSubplot:xlabel='BALANCE', ylabel='PURCHASES'>
```



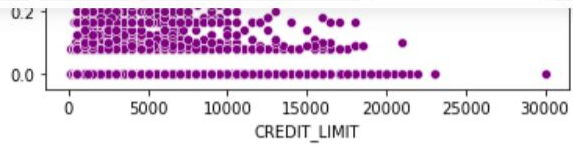


File Edit View Insert Cell Kernel Widgets Help

Trusted

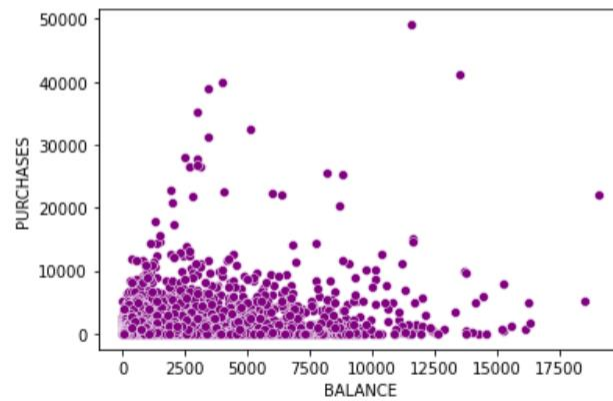


Python 3 (ipykernel)



```
In [19]: sns.scatterplot(x='BALANCE', y='PURCHASES', data=df,color='purple')
```

```
Out[19]: <AxesSubplot:xlabel='BALANCE', ylabel='PURCHASES'>
```



```
In [ ]: |
```