

Data Cleaning

Wellington Moreira Dos Santos

wsantos08@hotmail.com



O tratamento e limpeza de dados é uma etapa crucial em qualquer projeto de análise de dados. Em Python, existem diversas bibliotecas que podem ser utilizadas para realizar essa tarefa, como Pandas, Seaborn e Statistics

A limpeza de dados envolve a remoção de dados inconsistentes, incorretos, incompletos, duplicados ou irrelevantes, para garantir que os dados estejam corretos e confiáveis.

```
In [1]: import pandas as pd
import seaborn as sns
import statistics as sts
```

Explorando os dados

Carregando base de dados

```
In [2]: #importar dados
dataset = pd.read_csv("../dados/Churn.csv", sep=";")
```

Conhecendo a base de dados

Visualizando primeiros registros

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

```
Out[3]:
```

	X0	X1	X2	X3	X4	X4.1	X6	X7	X8	X9	X10	X11
0	1	619	RS	Feminino	42	2	0	1	1	1	10134888.0	1
1	2	608	SC	Feminino	41	1	8380786	1	0	1	11254258.0	0
2	3	502	RS	Feminino	42	8	1596608	3	1	0	11393157.0	1
3	4	699	RS	Feminino	39	1	0	2	0	0	9382663.0	0
4	5	850	SC	Feminino	43	2	12551082	1	1	1	790841.0	0

Dimensão do dataset

```
In [4]: #tamanho
dataset.shape
```

```
Out[4]: (999, 12)
```

Para melhor entendimento dos dados, vamos renomear as colunas conforme as regras de negócio.

```
In [5]: dataset.columns = ["Id", "Score", "Estado", "Genero", "Idade", "Patrimonio", "Saldo",
                           "Produtos", "TemCartCredito", "Ativo", "Salario", "Saiu"]
dataset.head()
```

```
Out[5]:
```

	Id	Score	Estado	Genero	Idade	Patrimonio	Saldo	Produtos	TemCartCredito	Ativo
0	1	619	RS	Feminino	42	2	0	1	1	1
1	2	608	SC	Feminino	41	1	8380786	1	0	1
2	3	502	RS	Feminino	42	8	1596608	3	1	0
3	4	699	RS	Feminino	39	1	0	2	0	0
4	5	850	SC	Feminino	43	2	12551082	1	1	1



Valores faltantes

```
In [6]: dataset.isnull().sum()
```


```
Out[6]: Id                0
Score                  0
Estado                0
Genero                 8
Idade                 0
Patrimonio            0
Saldo                 0
Produtos              0
TemCartCredito        0
Ativo                 0
Salario               7
Saiu                  0
dtype: int64
```

Valores duplicados pelo ID

```
In [7]: dataset[dataset.duplicated(['Id'],keep=False)]
```

```
Out[7]:
```

	Id	Score	Estado	Genero	Idade	Patrimonio	Saldo	Produtos	TemCartCredito	Ativo
80	81	665	RS	Feminino	34	1	9664554	2	0	(
81	81	665	RS	Feminino	34	1	9664554	2	0	(



Explorando as variáveis

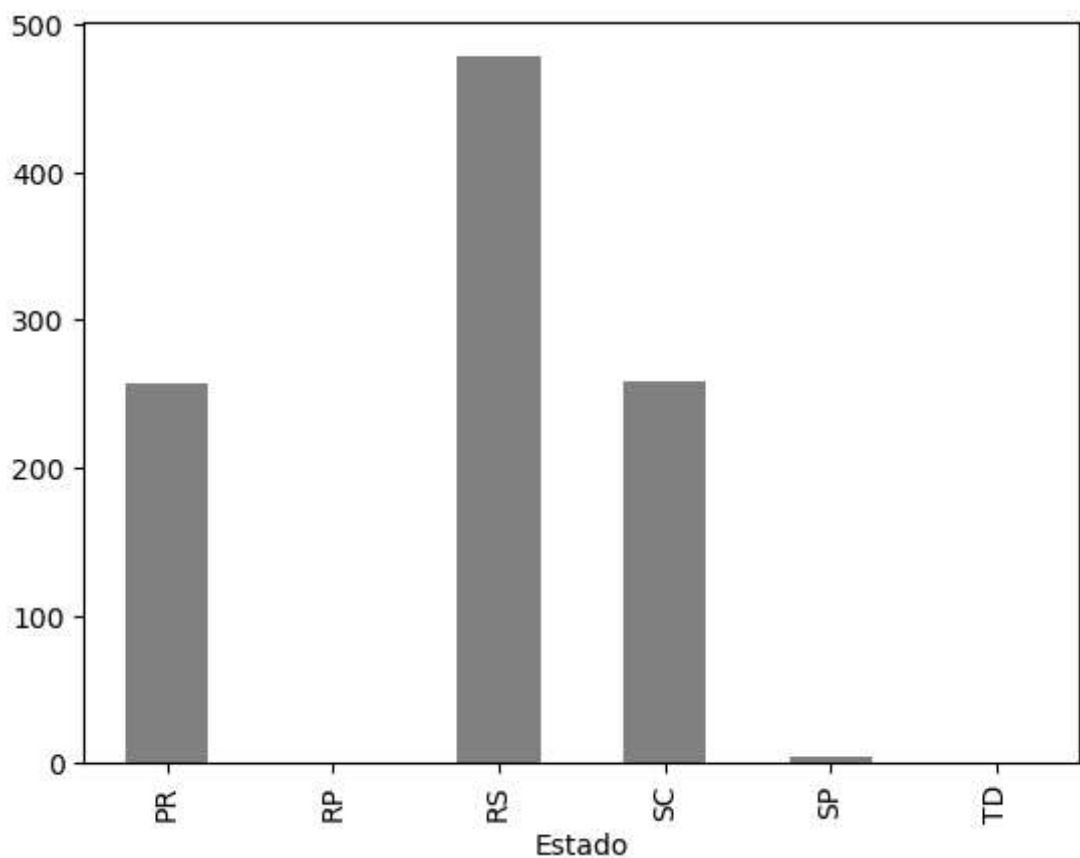
Estado

```
In [8]: agrupado = dataset.groupby(['Estado']).size()  
agrupado
```

```
Out[8]: Estado  
PR      257  
RP        1  
RS      478  
SC      258  
SP         4  
TD         1  
dtype: int64
```

```
In [9]: agrupado.plot.bar(color = 'gray')
```

```
Out[9]: <Axes: xlabel='Estado'>
```



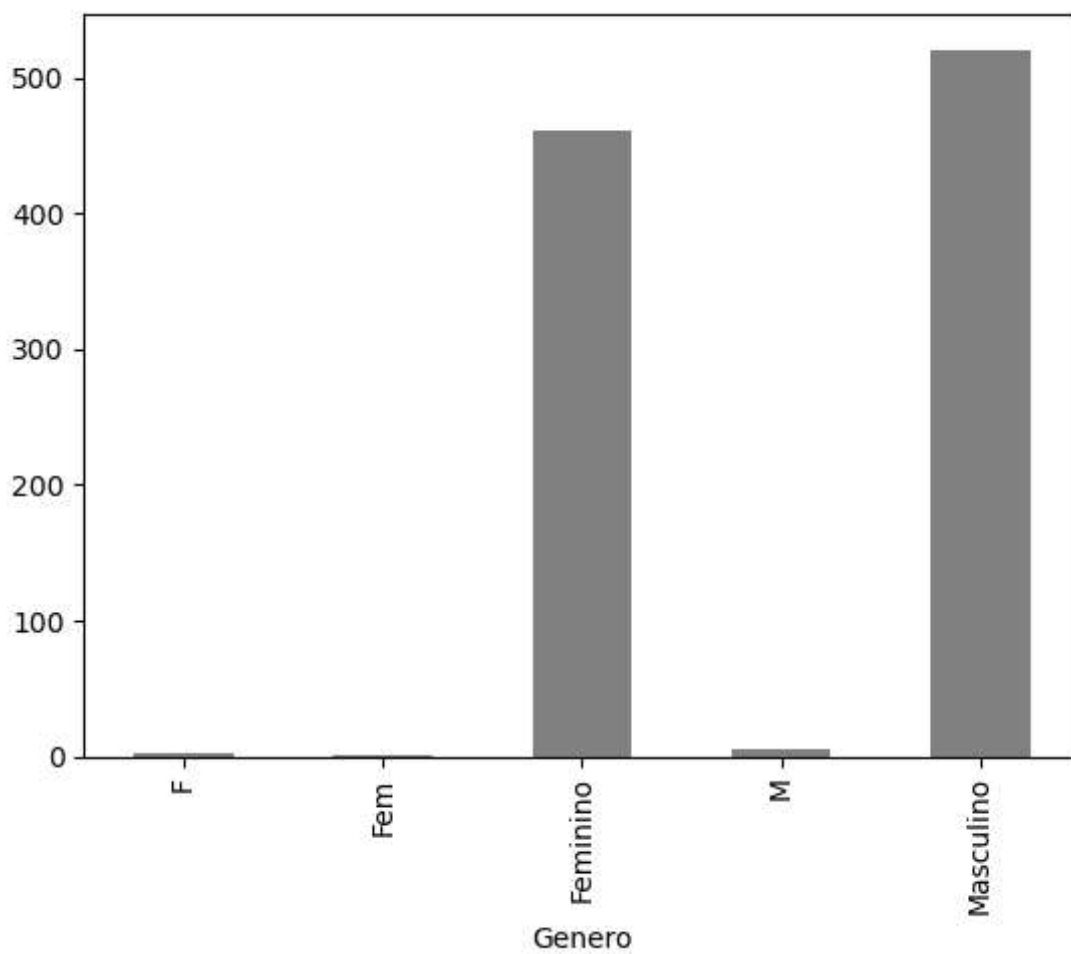
Genero

```
In [10]: agrupado = dataset.groupby(['Genero']).size()  
agrupado
```

```
Out[10]: Genero  
F          2  
Fem        1  
Feminino  461  
M          6  
Masculino  521  
dtype: int64
```

```
In [11]: agrupado.plot.bar(color = 'gray')
```

```
Out[11]: <Axes: xlabel='Genero'>
```



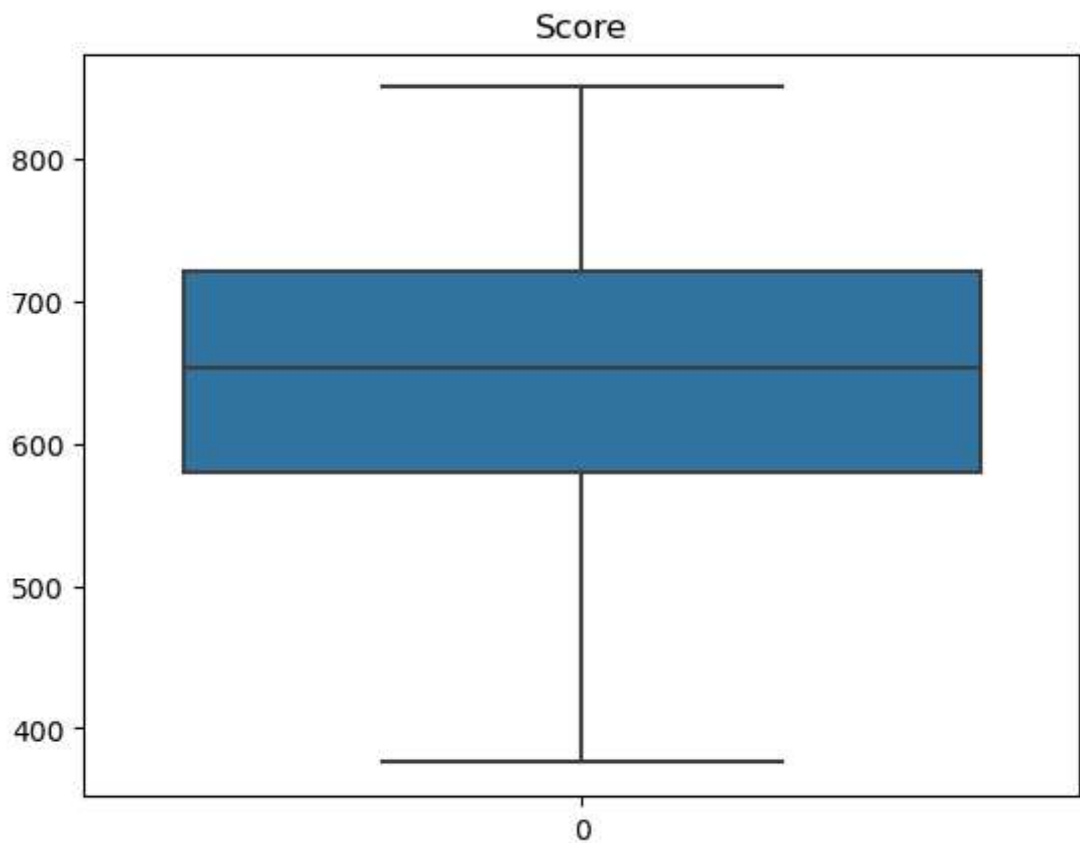
Score

```
In [13]: dataset['Score'].describe()
```

```
Out[13]: count    999.000000  
mean      648.621622  
std       98.264219  
min       376.000000  
25%      580.000000  
50%      653.000000  
75%      721.000000  
max       850.000000  
Name: Score, dtype: float64
```

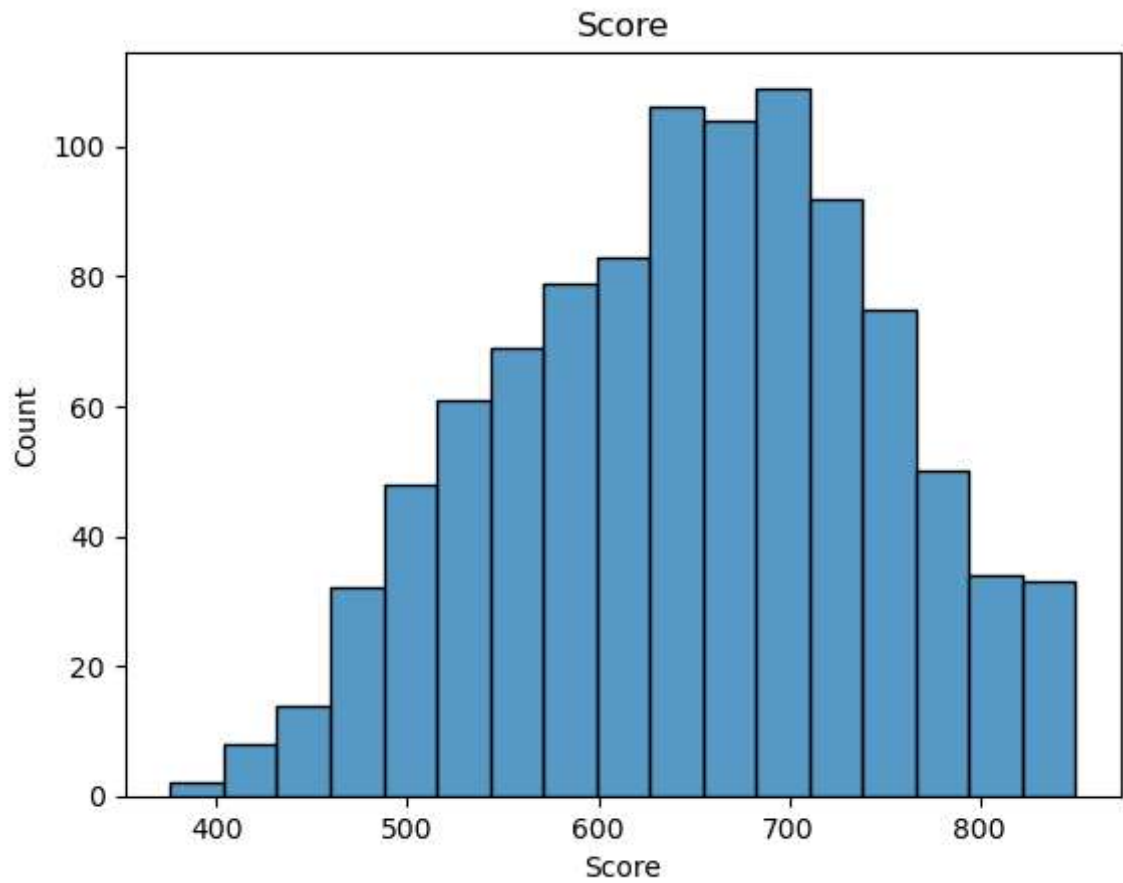
```
In [14]: srn.boxplot(dataset['Score']).set_title('Score')
```

```
Out[14]: Text(0.5, 1.0, 'Score')
```



```
In [15]: srn.histplot(dataset['Score']).set_title('Score')
```

```
Out[15]: Text(0.5, 1.0, 'Score')
```



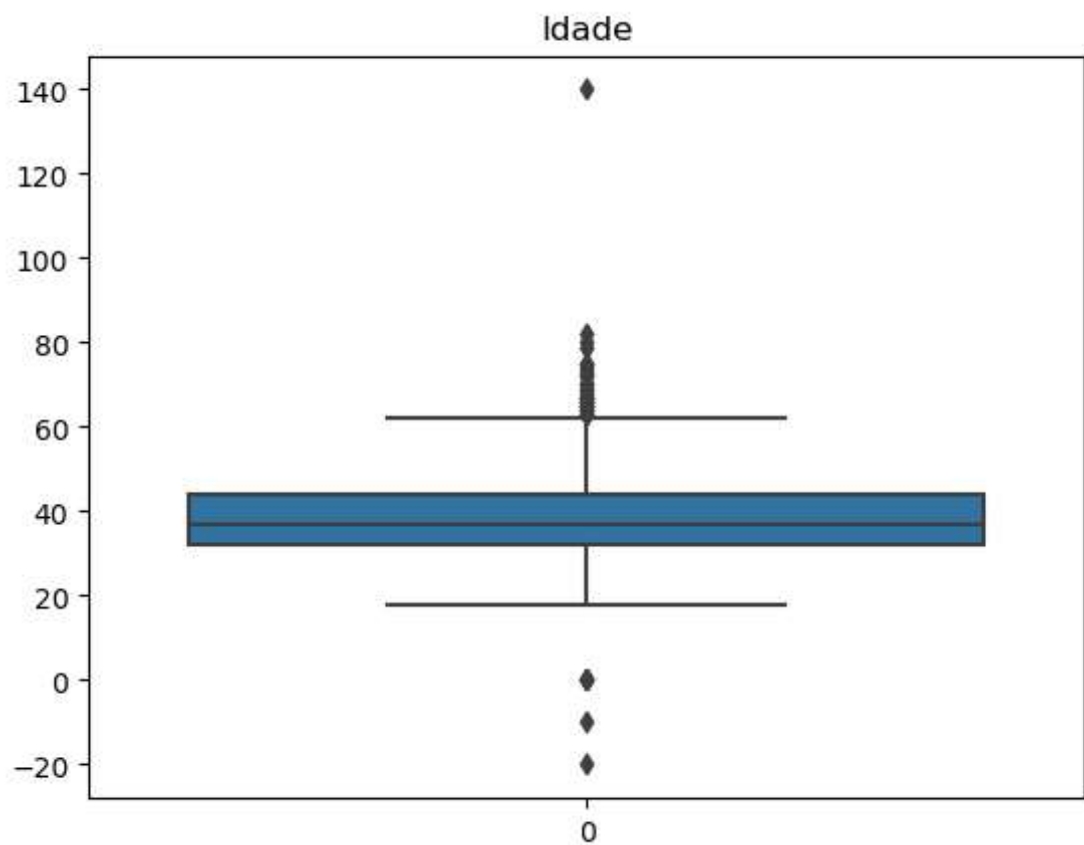
Idade

```
In [16]: dataset['Idade'].describe()
```

```
Out[16]: count    999.000000  
mean      38.902903  
std       11.401912  
min      -20.000000  
25%      32.000000  
50%      37.000000  
75%      44.000000  
max      140.000000  
Name: Idade, dtype: float64
```

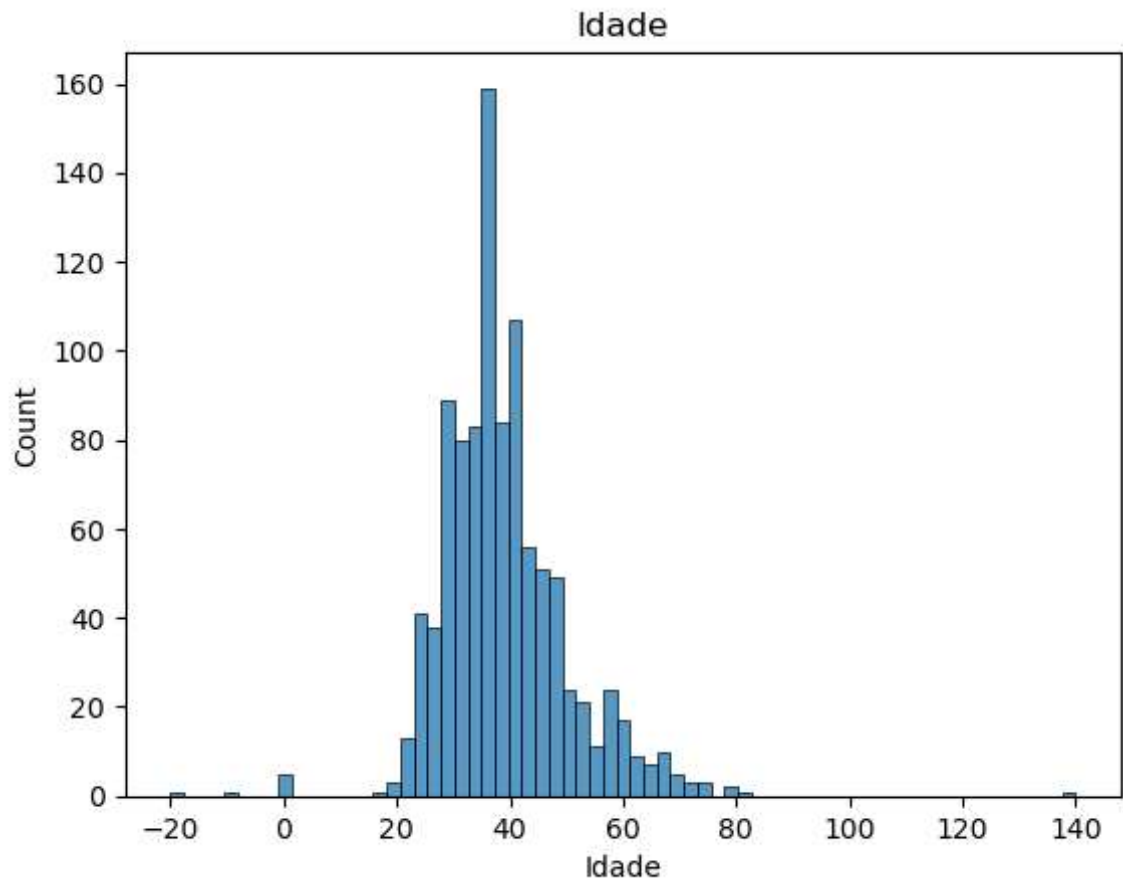
```
In [17]: srn.boxplot(dataset['Idade']).set_title('Idade')
```

```
Out[17]: Text(0.5, 1.0, 'Idade')
```




```
In [18]: srn.histplot(dataset['Idade']).set_title('Idade')
```

```
Out[18]: Text(0.5, 1.0, 'Idade')
```



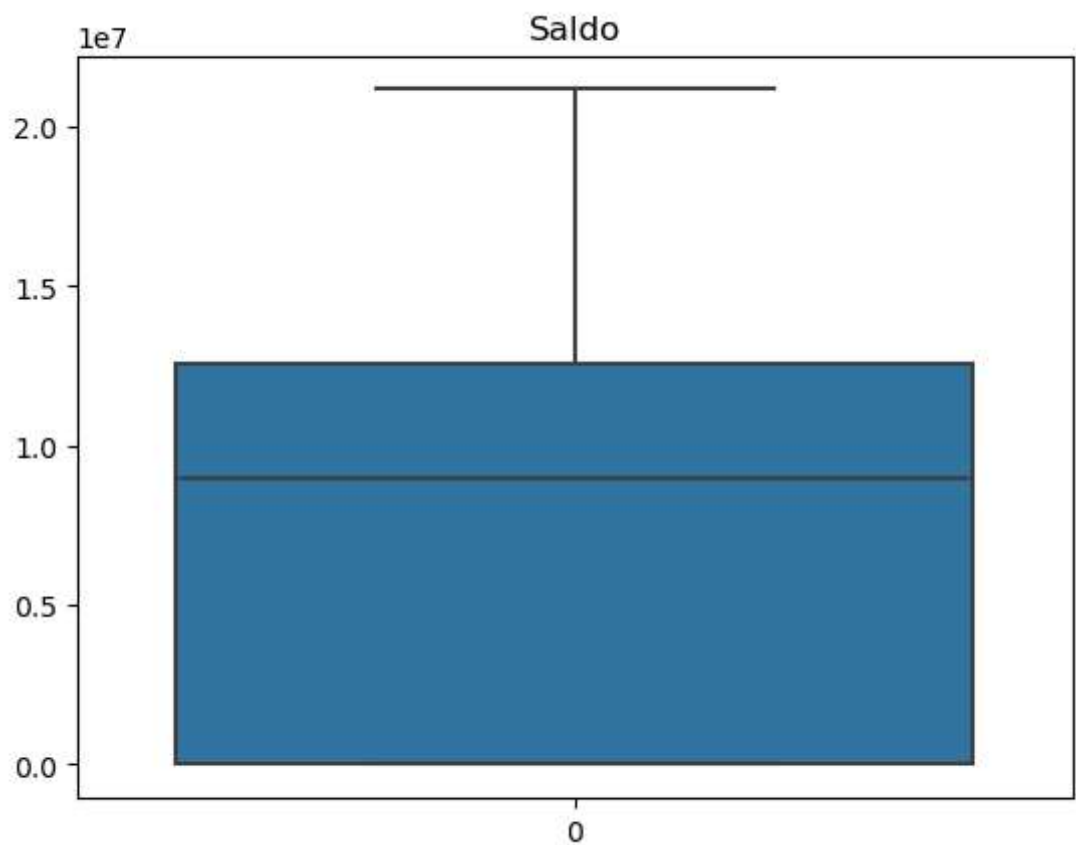
Saldo

```
In [19]: dataset['Saldo'].describe()
```

```
Out[19]: count      9.990000e+02  
mean       7.164928e+06  
std        6.311840e+06  
min        0.000000e+00  
25%        0.000000e+00  
50%        8.958835e+06  
75%       1.258684e+07  
max       2.117743e+07  
Name: Saldo, dtype: float64
```

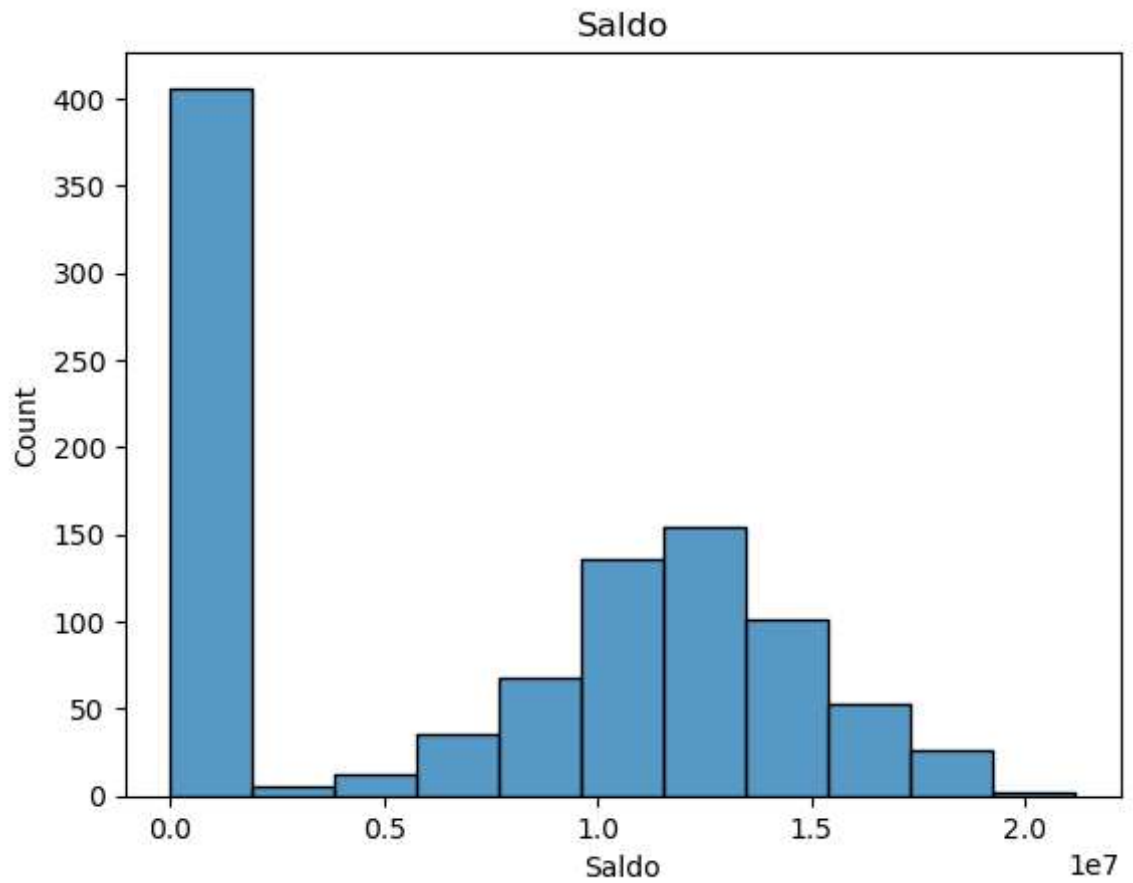
```
In [20]: srn.boxplot(dataset['Saldo']).set_title('Saldo')
```

```
Out[20]: Text(0.5, 1.0, 'Saldo')
```



```
In [21]: srn.histplot(dataset['Saldo']).set_title('Saldo')
```

```
Out[21]: Text(0.5, 1.0, 'Saldo')
```



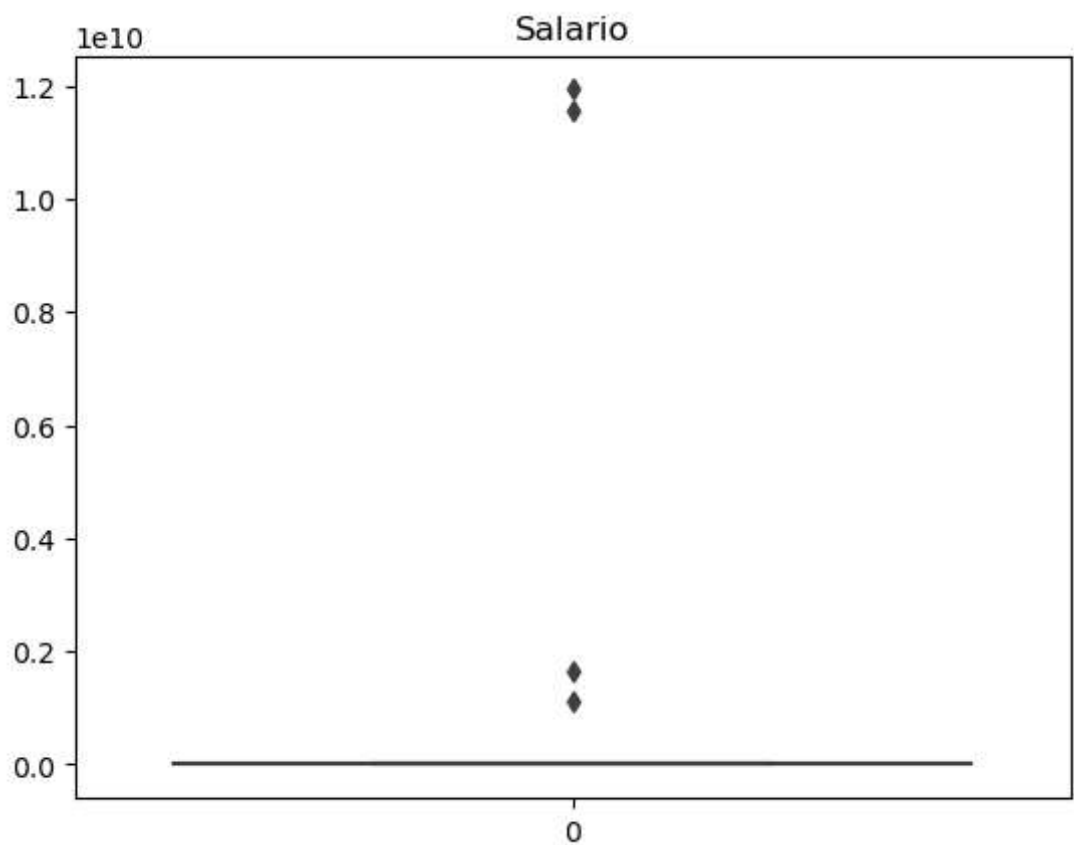
Salário

```
In [22]: dataset['Salario'].describe()
```

```
Out[22]: count      9.920000e+02  
mean       3.528762e+07  
std        5.305800e+08  
min        9.677000e+03  
25%        3.029011e+06  
50%        8.703250e+06  
75%        1.405213e+07  
max        1.193469e+10  
Name: Salario, dtype: float64
```

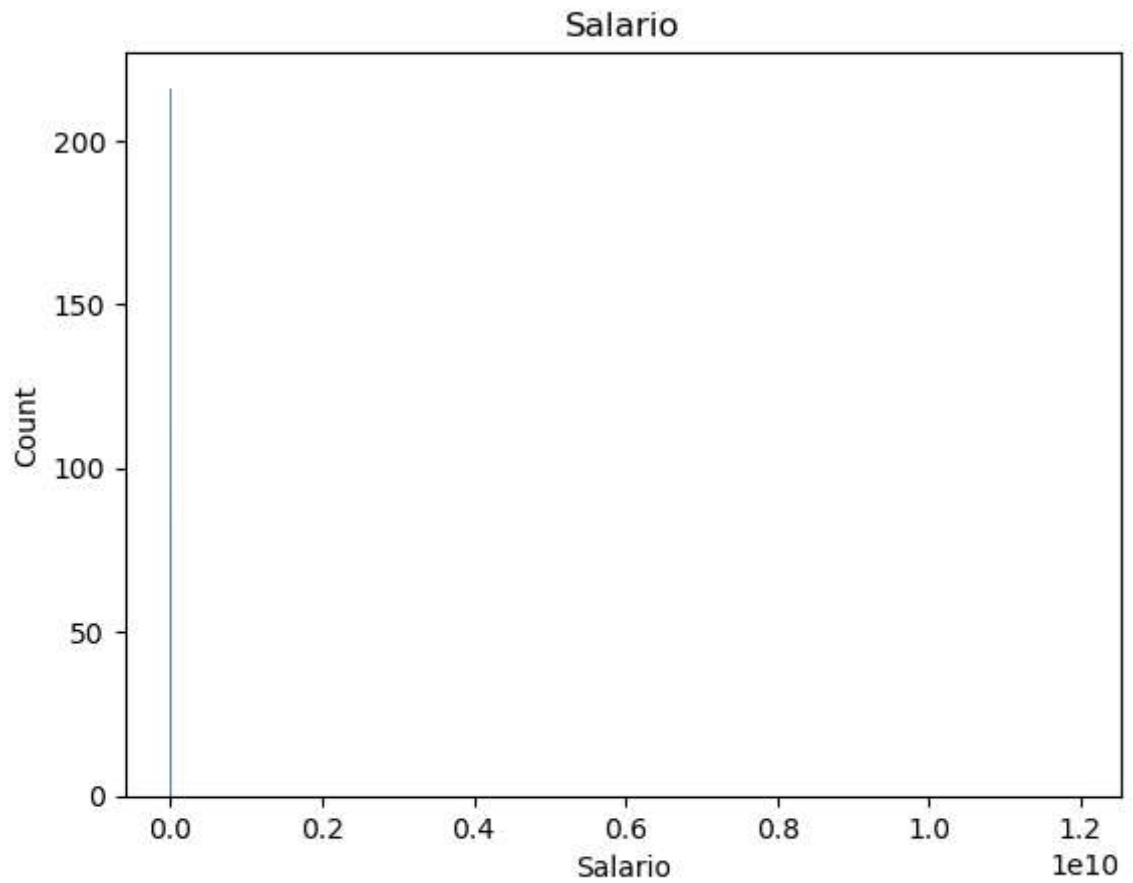
```
In [23]: srn.boxplot(dataset['Salario']).set_title('Salario')
```

```
Out[23]: Text(0.5, 1.0, 'Salario')
```



```
In [24]: srn.histplot(dataset['Salario']).set_title('Salario')
```

```
Out[24]: Text(0.5, 1.0, 'Salario')
```



Tratamento de dados

Dados duplicados

Podemos constatar que existe uma duplicata de registros com base no Id, aqui removeremos.

```
In [25]: dataset.drop_duplicates(subset="Id", keep='first', inplace=True)
```

Conferindo se ainda existe a duplicata

```
In [26]: dataset[dataset.duplicated(['Id'], keep=False)]
```

```
Out[26]:
```

Id	Score	Estado	Genero	Idade	Patrimonio	Saldo	Produtos	TemCartCredito	Ativo	Salario
----	-------	--------	--------	-------	------------	-------	----------	----------------	-------	---------



Salários

Aqui possuímos valores faltantes NAs e Outliers como vimos graficamente, utilizaremos a mediana para preencher valores faltantes e substituição de outliers.

Mediana

```
In [27]: mediana = sts.median(dataset['Salario'])
         mediana
```

```
Out[27]: 73752.0
```

Substituindo NA por mediana

```
In [33]: dataset['Salario'].fillna(mediana, inplace=True)
```

Verificando se ainda existem valores faltantes na variável

```
In [34]: dataset['Salario'].isnull().sum()
```

```
Out[34]: 0
```

Desvio padrão

```
In [54]: desv = sts.stdev(dataset['Salario'])
         desv
```


```
Out[54]: 6066172.5605729
```

Definindo critério de salário com desvio padrão

```
In [32]: dataset.loc[dataset['Salarrio'] >= 2 * desv ]
```

```
Out[32]:
```

	Id	Score	Estado	Genero	Idade	Patrimonio	Saldo	Produtos	TemCartCredito
7	8	376	PR	Feminino	29	4	11504674	4	1
116	118	668	PR	Feminino	37	6	1678644	1	1
170	172	484	RS	Feminino	29	4	13011439	1	1
230	232	673	RS	Masculino	72	1	0	2	0



Atualizando mediana e substituindo valores

```
In [35]: mediana = sts.median(dataset['Salarrio'])  
dataset.loc[dataset['Salarrio'] >= 2 * desv, 'Salarrio'] = mediana
```

Conferindo substituição de valores

```
In [36]: dataset.loc[dataset['Salarrio'] >= 2 * desv ]
```

```
Out[36]:
```

	Id	Score	Estado	Genero	Idade	Patrimonio	Saldo	Produtos	TemCartCredito	Ativo	Sal
--	----	-------	--------	--------	-------	------------	-------	----------	----------------	-------	-----



Genero

Podemos constatar que falta padronização com os dados e valores faltantes, por ser uma variável categórica utilizamos a moda para NAs e por fim, padronizamos os domínios.

```
In [37]: agrupado = dataset.groupby(['Genero']).size()  
agrupado
```

```
Out[37]: Genero  
F          2  
Fem        1  
Feminino  460  
M          6  
Masculino  521  
dtype: int64
```

Valores faltantes

```
In [38]: dataset['Genero'].isnull().sum()
```

```
Out[38]: 8
```

Substituindo NAs por Masculino (moda)

```
In [39]: dataset['Genero'].fillna('Masculino', inplace=True)
```

Verificando se ainda existem valores faltantes na variável

```
In [40]: dataset['Genero'].isnull().sum()
```

```
Out[40]: 0
```

Padronizando domínios

```
In [41]: dataset.loc[dataset['Genero'] == 'M', 'Genero'] = "Masculino"  
dataset.loc[dataset['Genero'].isin(['Fem', 'F']), 'Genero'] = "Feminino"
```

Conferindo padronização

```
In [42]: agrupado = dataset.groupby(['Genero']).size()  
agrupado
```

```
Out[42]: Genero  
Feminino    463  
Masculino   535  
dtype: int64
```


Idades

Além de haver dados faltantes também existem valores fora de domínio, conforme as regras de negócio, logo, por ser uma variável numérica utilizamos a mediana para preencher estes valores.

```
In [43]: dataset['Idade'].describe()
```

```
Out[43]: count    998.000000
mean      38.907816
std       11.406570
min       -20.000000
25%       32.000000
50%       37.000000
75%       44.000000
max       140.000000
Name: Idade, dtype: float64
```

Valores fora de domínio

```
In [45]: dataset.loc[(dataset['Idade'] < 0) | (dataset['Idade'] > 120)]
```

```
Out[45]:
```

	Id	Score	Estado	Genero	Idade	Patrimonio	Saldo	Produtos	TemCartCredito
867	869	636	RS	Feminino	-10	1	17083346	1	1
984	986	773	RS	Masculino	-20	1	12453278	2	0
990	992	655	RS	Masculino	140	5	93147	2	1

Substituindo valores fora de domínio por Mediana

```
In [46]: mediana = sts.median(dataset['Idade'])
dataset.loc[(dataset['Idade'] < 0) | (dataset['Idade'] > 120), 'Idade']
```

Conferindo se ainda existem idades fora do domínio

```
In [47]: dataset.loc[(dataset['Idade'] < 0) | (dataset['Idade'] > 120)]
```

```
Out[47]:
```

	Id	Score	Estado	Genero	Idade	Patrimonio	Saldo	Produtos	TemCartCredito	Ativo	Saldo
--	----	-------	--------	--------	-------	------------	-------	----------	----------------	-------	-------

Estado

Aqui vamos tratar valores que estão fora do domínio das regras de negócio, por serem categóricos também utilizamos a moda, aqui sendo RS para substituição de valores

```
In [48]: agrupado = dataset.groupby(['Estado']).size()  
agrupado
```

```
Out[48]: Estado  
PR      257  
RP        1  
RS      477  
SC      258  
SP        4  
TD        1  
dtype: int64
```

Substituindo pela moda 'RS'

```
In [49]: dataset.loc[dataset['Estado'].isin( ['RP','SP','TD']), 'Estado'] = "RS"  
agrupado = dataset.groupby(['Estado']).size()
```

Conferindo tratamento dos dados

```
In [50]: agrupado
```

```
Out[50]: Estado  
PR      257  
RS      483  
SC      258  
dtype: int64
```

Considerações finais

Após explorar dados podemos perceber inconsistências, são elas:

- Dados duplicados
- Valores ausentes
- Outliers
- Valores fora de domínio

Para descrever e resumir um conjunto de dados, aplicamos técnicas de estatística descritiva, e com auxílio de bibliotecas utilizamos estas técnicas neste dataset.

Dataset primeiras linhas

```
In [55]: dataset.head()
```

	Id	Score	Estado	Genero	Idade	Patrimonio	Saldo	Produtos	TemCartCredito	Ativo
0	1	619	RS	Feminino	42	2	0	1	1	1
1	2	608	SC	Feminino	41	1	8380786	1	0	1
2	3	502	RS	Feminino	42	8	1596608	3	1	0
3	4	699	RS	Feminino	39	1	0	2	0	0
4	5	850	SC	Feminino	43	2	12551082	1	1	1

Dimensões

```
In [56]: dataset.shape
```

```
Out[56]: (998, 12)
```

Descrição por variável

```
In [57]: dataset.describe()
```

	Id	Score	Idade	Patrimonio	Saldo	Produtos	TemCartCr
count	998.000000	998.000000	998.000000	998.000000	9.980000e+02	998.000000	998.000000
mean	501.337675	648.605210	38.908818	5.073146	7.162423e+06	1.526052	0.700000
std	288.500953	98.312117	10.676642	2.926320	6.314508e+06	0.574293	0.450000
min	1.000000	376.000000	0.000000	0.000000	0.000000e+00	1.000000	0.000000
25%	252.250000	580.000000	32.000000	2.000000	0.000000e+00	1.000000	0.000000
50%	501.500000	653.000000	37.000000	5.000000	8.926348e+06	1.000000	1.000000
75%	750.750000	721.000000	44.000000	8.000000	1.258767e+07	2.000000	1.000000
max	1000.000000	850.000000	82.000000	10.000000	2.117743e+07	4.000000	1.000000

Ausência de dados faltantes

```
In [61]: dataset.isnull().sum()
```

```
Out[61]: Id                0  
         Score            0  
         Estado           0  
         Genero           0  
         Idade            0  
         Patrimonio       0  
         Saldo            0  
         Produtos         0  
         TemCartCredito   0  
         Ativo            0  
         Salario          0  
         Saiu             0  
         dtype: int64
```

Ausência de duplicatas

```
In [62]: dataset[dataset.duplicated(['Id'],keep=False)]
```

```
Out[62]:
```

	Id	Score	Estado	Genero	Idade	Patrimonio	Saldo	Produtos	TemCartCredito	Ativo	Salario	Saiu
--	----	-------	--------	--------	-------	------------	-------	----------	----------------	-------	---------	------

