Análise Exploratória

Objetivo: Compreender e visualizar os dados fornecidos na pasta Dados em TXT, fornecidos na seção de microdados em https://www.ibge.gov.br/estatisticas/sociais/saude/24786-pesquisa-de-orcamentos-familiares-2.html (https://www.ibge.gov.br/estatisticas/sociais/saude/24786-pesquisa-de-orcamentos-familiares-2.html) de forma estatística descritiva. Para saber mais sobre o significado das colunas e tabelas, acessar a documentação da Análise Exploratória.

Importação e leitura das tabelas

In [1]:
```
%matplotlib inline
```

In [2]:
```
!pip install https://github.com/pandas-profiling/pandas-profiling/archive/r
```

```
Collecting https://github.com/pandas-profiling/pandas-profiling/archive/m
aster.zip (https://github.com/pandas-profiling/pandas-profiling/archive/m
aster.zip)
  Using cached https://github.com/pandas-profiling/pandas-profiling/archi
ve/master.zip (https://github.com/pandas-profiling/pandas-profiling/archi
ve/master.zip)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: scipy<1.12,>=1.4.1 in c:\users\rodrigo\ana
conda3\lib\site-packages (from ydata-profiling==0.0.dev0) (1.10.1)
Requirement already satisfied: pandas!=1.4.0,<2.1,>1.1 in c:\users\rodrig
o\anaconda3\lib\site-packages (from ydata-profiling==0.0.dev0) (1.5.3)
Requirement already satisfied: matplotlib<=3.7.3,>=3.2 in c:\users\rodrig
o\anaconda3\lib\site-packages (from ydata-profiling==0.0.dev0) (3.7.1)
Requirement already satisfied: pydantic>=2 in c:\users\rodrigo\anaconda3
\lib\site-packages (from ydata-profiling==0.0.dev0) (2.4.2)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in c:\users\rodrigo\ana
conda3\lib\site-packages (from ydata-profiling==0.0.dev0) (6.0)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in c:\users\rodrigo\an
aconda3\lib\site-packages (from ydata-profiling==0.0.dev0) (3.1.2)
Requirement already satisfied: visions[type_image_path]==0.7.5 in c:\user
s\rodrigo\anaconda3\lib\site-packages (from ydata-profiling==0.0.dev0)
(0.7.5)
Requirement already satisfied: numpy<1.26,>=1.16.0 in c:\users\rodrigo\an
aconda3\lib\site-packages (from ydata-profiling==0.0.dev0) (1.24.3)
Requirement already satisfied: htmlmin==0.1.12 in c:\users\rodrigo\anacon
da3\lib\site-packages (from ydata-profiling==0.0.dev0) (0.1.12)
Requirement already satisfied: phik<0.13,>=0.11.1 in c:\users\rodrigo\ana
conda3\lib\site-packages (from ydata-profiling==0.0.dev0) (0.12.3)
Requirement already satisfied: requests<3,>=2.24.0 in c:\users\rodrigo\an
aconda3\lib\site-packages (from ydata-profiling==0.0.dev0) (2.29.0)
Requirement already satisfied: tqdm<5,>=4.48.2 in c:\users\rodrigo\anacon
da3\lib\site-packages (from ydata-profiling==0.0.dev0) (4.65.0)
Requirement already satisfied: seaborn<0.13,>=0.10.1 in c:\users\rodrigo
\anaconda3\lib\site-packages (from ydata-profiling==0.0.dev0) (0.12.2)
Requirement already satisfied: multimethod<2,>=1.4 in c:\users\rodrigo\an
aconda3\lib\site-packages (from ydata-profiling==0.0.dev0) (1.10)
Requirement already satisfied: statsmodels<1,>=0.13.2 in c:\users\rodrigo
\anaconda3\lib\site-packages (from ydata-profiling==0.0.dev0) (0.13.5)
Requirement already satisfied: typeguard<5,>=4.1.2 in c:\users\rodrigo\an
aconda3\lib\site-packages (from ydata-profiling==0.0.dev0) (4.1.5)
Requirement already satisfied: imagehash==4.3.1 in c:\users\rodrigo\anaco
nda3\lib\site-packages (from ydata-profiling==0.0.dev0) (4.3.1)
Requirement already satisfied: wordcloud>=1.9.1 in c:\users\rodrigo\anaco
nda3\lib\site-packages (from ydata-profiling==0.0.dev0) (1.9.2)
Requirement already satisfied: dacite>=1.8 in c:\users\rodrigo\anaconda3
\lib\site-packages (from ydata-profiling==0.0.dev0) (1.8.1)
Requirement already satisfied: numba<0.59.0,>=0.56.0 in c:\users\rodrigo
\anaconda3\lib\site-packages (from ydata-profiling==0.0.dev0) (0.57.0)
Requirement already satisfied: PyWavelets in c:\users\rodrigo\anaconda3\l
ib\site-packages (from imagehash==4.3.1->ydata-profiling==0.0.dev0) (1.4.
1)
Requirement already satisfied: pillow in c:\users\rodrigo\anaconda3\lib\s
ite-packages (from imagehash==4.3.1->ydata-profiling==0.0.dev0) (9.4.0)
Requirement already satisfied: attrs>=19.3.0 in c:\users\rodrigo\anaconda
3\lib\site-packages (from visions[type_image_path]==0.7.5->ydata-profilin
g==0.0.dev0) (22.1.0)
Requirement already satisfied: networkx>=2.4 in c:\users\rodrigo\anaconda
3\lib\site-packages (from visions[type_image_path]==0.7.5->ydata-profilin
g==0.0.dev0) (2.8.4)
Requirement already satisfied: tangled-up-in-unicode>=0.0.4 in c:\users\r
```

```
odrigo\anaconda3\lib\site-packages (from visions[type_image_path]==0.7.5-
>ydata-profiling==0.0.dev0) (0.2.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\rodrigo\anacon
da3\lib\site-packages (from jinja2<3.2,>=2.11.1->ydata-profiling==0.0.dev
0) (2.1.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\rodrigo\anaco
nda3\lib\site-packages (from matplotlib<=3.7.3,>=3.2->ydata-profiling==0.
0.dev0) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\rodrigo\anaconda3
\lib\site-packages (from matplotlib<=3.7.3,>=3.2->ydata-profiling==0.0.de
v0) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\rodrigo\anac
onda3\lib\site-packages (from matplotlib<=3.7.3,>=3.2->ydata-profiling==
0.0.dev0) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\rodrigo\anac
onda3\lib\site-packages (from matplotlib<=3.7.3,>=3.2->ydata-profiling==
0.0.dev0) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\rodrigo\anacon
da3\lib\site-packages (from matplotlib<=3.7.3,>=3.2->ydata-profiling==0.
0.dev0) (23.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\rodrigo\anaco
nda3\lib\site-packages (from matplotlib<=3.7.3,>=3.2->ydata-profiling==0.
0.dev0) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\rodrigo\a
naconda3\lib\site-packages (from matplotlib<=3.7.3,>=3.2->ydata-profiling
==0.0.dev0) (2.8.2)
Requirement already satisfied: llvmlite<0.41,>=0.40.0dev0 in c:\users\rod
rigo\anaconda3\lib\site-packages (from numba<0.59.0,>=0.56.0->ydata-profi
ling==0.0.dev0) (0.40.0)
Requirement already satisfied: pytz>=2020.1 in c:\users\rodrigo\anaconda3
\lib\site-packages (from pandas!=1.4.0,<2.1,>1.1->ydata-profiling==0.0.de
v0) (2022.7)
Requirement already satisfied: joblib>=0.14.1 in c:\users\rodrigo\anacond
a3\lib\site-packages (from phik<0.13,>=0.11.1->ydata-profiling==0.0.dev0)
(1.2.0)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\rodrigo
\anaconda3\lib\site-packages (from pydantic>=2->ydata-profiling==0.0.dev
0) (0.6.0)
Requirement already satisfied: pydantic-core==2.10.1 in c:\users\rodrigo
\anaconda3\lib\site-packages (from pydantic>=2->ydata-profiling==0.0.dev
0) (2.10.1)
Requirement already satisfied: typing-extensions>=4.6.1 in c:\users\rodri
go\anaconda3\lib\site-packages (from pydantic>=2->ydata-profiling==0.0.de
v0) (4.8.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\rodri
go\anaconda3\lib\site-packages (from requests<3,>=2.24.0->ydata-profiling
==0.0.dev0) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\rodrigo\anaconda3
\lib\site-packages (from requests<3,>=2.24.0->ydata-profiling==0.0.dev0)
(3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\rodrigo
\anaconda3\lib\site-packages (from requests<3,>=2.24.0->ydata-profiling==
0.0.dev0) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\rodrigo\ana
conda3\lib\site-packages (from requests<3,>=2.24.0->ydata-profiling==0.0.
dev0) (2023.5.7)
Requirement already satisfied: patsy>=0.5.2 in c:\users\rodrigo\anaconda3
\lib\site-packages (from statsmodels<1,>=0.13.2->ydata-profiling==0.0.dev
0) (0.5.3)
Requirement already satisfied: colorama in c:\users\rodrigo\anaconda3\lib
\site-packages (from tqdm<5,>=4.48.2->ydata-profiling==0.0.dev0) (0.4.6)
```

```
Requirement already satisfied: six in c:\users\rodrigo\anaconda3\lib\site
-packages (from patsy>=0.5.2->statsmodels<1,>=0.13.2->ydata-profiling==0.
0.dev0) (1.16.0)
```

In [3]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from pandas_profiling import ProfileReport
plt.style.use('ggplot')
```

```
C:\Users\Rodrigo\AppData\Local\Temp\ipykernel_17332\2621030763.py:5: Depr
ecationWarning: `import pandas_profiling` is going to be deprecated by Ap
ril 1st. Please use `import ydata_profiling` instead.
  from pandas_profiling import ProfileReport
```

In [4]:
```
pd.set_option('display.max_columns', 200)
```

In [12]:
```
#importação do arquivo de aluguel_estimado
df_aluguel = pd.read_csv(r"..\Analise_Exploratoria\Dados_Csv\ALUGUEL_ESTIMA
df_aluguel.head()
```

Out[12]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	QUADRO	V
0	11	1103	1	110005400	1	1	0	
1	11	1103	1	110005400	4	1	0	
2	11	1103	1	110005400	5	1	0	
3	11	1103	1	110005400	6	1	0	
4	11	1103	1	110005400	7	1	0	

In [4]:
```
df_aluguel.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48935 entries, 0 to 48934
Data columns (total 19 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   UF                  48935 non-null  int64  
 1   ESTRATO_POF         48935 non-null  int64  
 2   TIPO_SITUACAO_REG   48935 non-null  int64  
 3   COD_UPA             48935 non-null  int64  
 4   NUM_DOM             48935 non-null  int64  
 5   NUM_UC              48935 non-null  int64  
 6   QUADRO              48935 non-null  int64  
 7   V9001               48935 non-null  int64  
 8   V9002               48935 non-null  int64  
 9   V8000               48935 non-null  float64
 10  V9010               48935 non-null  int64  
 11  V9011               48935 non-null  int64  
 12  DEFLATOR            48935 non-null  float64
 13  V8000_DEFLA         48935 non-null  float64
 14  COD_IMPUT_VALOR     48935 non-null  int64  
 15  FATOR_ANUALIZACAO   48935 non-null  int64  
 16  PESO                48935 non-null  float64
 17  PESO_FINAL          48935 non-null  float64
 18  RENDA_TOTAL         48935 non-null  float64
dtypes: float64(6), int64(13)
memory usage: 7.1 MB
```

In [13]:
```
#importação de arquivo despesa_coletiva
df_despesa_coletiva = pd.read_csv(r"..\Analise_Exploratoria\Dados_CSV\DESPE
df_despesa_coletiva.head()
```

Out[13]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	QUADRO	S
0	11	1103	1	110005400	1	1	6	
1	11	1103	1	110005400	1	1	6	
2	11	1103	1	110005400	1	1	6	
3	11	1103	1	110005400	1	1	6	
4	11	1103	1	110005400	1	1	6	

In [6]:
```
df_despesa_coletiva.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 478572 entries, 0 to 478571
Data columns (total 27 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   UF                    478572 non-null  int64  
 1   ESTRATO_POF           478572 non-null  int64  
 2   TIPO_SITUACAO_REG     478572 non-null  int64  
 3   COD_UPA               478572 non-null  int64  
 4   NUM_DOM               478572 non-null  int64  
 5   NUM_UC                478572 non-null  int64  
 6   QUADRO                478572 non-null  int64  
 7   SEQ                   478572 non-null  int64  
 8   V9001                 478572 non-null  int64  
 9   V9002                 478572 non-null  int64  
 10  V9005                 52182 non-null   float64
 11  V8000                 478572 non-null  float64
 12  V9010                 44470 non-null   float64
 13  V9011                 44470 non-null   float64
 14  V9012                 138423 non-null  float64
 15  V1904                 1452 non-null    float64
 16  V1905                 10277 non-null   float64
 17  DEFLATOR              470612 non-null  float64
 18  V8000_DEFLA           478572 non-null  float64
 19  V1904_DEFLA           1442 non-null    float64
 20  COD_IMPUT_VALOR       478572 non-null  int64  
 21  COD_IMPUT_QUANTIDADE  52182 non-null   float64
 22  FATOR_ANUALIZACAO     478572 non-null  int64  
 23  PESO                  478572 non-null  float64
 24  PESO_FINAL            478572 non-null  float64
 25  RENDA_TOTAL           478572 non-null  float64
 26  V9004                 258745 non-null  float64
dtypes: float64(15), int64(12)
memory usage: 98.6 MB
```

In [57]:
```
#importação de arquivo domicílio
df_domicilio = pd.read_csv(r"..\Analise_Exploratoria\Dados_CSV/DOMICILIO.cs
df_domicilio.head()
```

Out[57]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	V0201	V0202	V0203
0	11	1103	1	110005400	1	1	1	1
1	11	1103	1	110005400	2	1	1	1
2	11	1103	1	110005400	4	1	4	1
3	11	1103	1	110005400	5	1	4	1
4	11	1103	1	110005400	6	1	1	1

In [9]:
```
df_domicilio.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57920 entries, 0 to 57919
Data columns (total 38 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   UF                 57920 non-null  int64  
 1   ESTRATO_POF        57920 non-null  int64  
 2   TIPO_SITUACAO_REG  57920 non-null  int64  
 3   COD_UPA            57920 non-null  int64  
 4   NUM_DOM            57920 non-null  int64  
 5   V0201              57920 non-null  int64  
 6   V0202              57920 non-null  int64  
 7   V0203              57920 non-null  int64  
 8   V0204              57920 non-null  int64  
 9   V0205              57920 non-null  int64  
 10  V0206              57920 non-null  int64  
 11  V0207              57920 non-null  int64  
 12  V0208              44187 non-null  float64
 13  V0209              57920 non-null  int64  
 14  V02101             54854 non-null  float64
 15  V02102             54854 non-null  float64
 16  V02103             54854 non-null  float64
 17  V02104             54854 non-null  float64
 18  V02105             54854 non-null  float64
 19  V02111             57920 non-null  int64  
 20  V02112             2263 non-null   float64
 21  V02113             1910 non-null   float64
 22  V0212              57256 non-null  float64
 23  V0213              57920 non-null  int64  
 24  V02141             57920 non-null  int64  
 25  V02142             57920 non-null  int64  
 26  V0215              57333 non-null  float64
 27  V02161             57920 non-null  int64  
 28  V02162             57920 non-null  int64  
 29  V02163             57920 non-null  int64  
 30  V02164             57920 non-null  int64  
 31  V0217              57920 non-null  int64  
 32  V0219              8985 non-null   float64
 33  V0220              57920 non-null  int64  
 34  V0221              57920 non-null  int64  
 35  PESO               57920 non-null  float64
 36  PESO_FINAL         57920 non-null  float64
 37  V6199              57920 non-null  int64  
dtypes: float64(13), int64(25)
memory usage: 16.8 MB
```

In [58]:
```
#importação de arquivo morador
df_morador = pd.read_csv(r"..\Analise_Exploratoria\Dados_CSV\MORADOR.csv")
df_morador.head()
```

Out[58]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	COD_INFOR
0	11	1101	1	110000016	2	1	
1	11	1101	1	110000016	2	1	
2	11	1101	1	110000016	2	1	
3	11	1101	1	110000016	3	1	
4	11	1101	1	110000016	3	1	

In [11]:
```
df_morador.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178431 entries, 0 to 178430
Data columns (total 56 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   UF                 178431 non-null  int64  
 1   ESTRATO_POF        178431 non-null  int64  
 2   TIPO_SITUACAO_REG  178431 non-null  int64  
 3   COD_UPA            178431 non-null  int64  
 4   NUM_DOM            178431 non-null  int64  
 5   NUM_UC             178431 non-null  int64  
 6   COD_INFORMANTE     178431 non-null  int64  
 7   V0306              178431 non-null  int64  
 8   V0401              178431 non-null  int64  
 9   V04021             178431 non-null  int64  
 10  V04022             178431 non-null  int64  
 11  V04023             178431 non-null  int64  
 12  V0403              178431 non-null  int64  
 13  V0404              178431 non-null  int64  
 14  V0405              178431 non-null  int64  
 15  V0406              178431 non-null  int64  
 16  V0407              148116 non-null  float64
 17  V0408              148116 non-null  float64
 18  V0409              128278 non-null  float64
 19  V0410              128278 non-null  float64
 20  V0411              52641 non-null   float64
 21  V0412              12336 non-null   float64
 22  V0413              128278 non-null  float64
 23  V0414              178431 non-null  int64  
 24  V0415              178431 non-null  int64  
 25  V0416              16730 non-null   float64
 26  V041711            16730 non-null   float64
 27  V041712            3038 non-null    float64
 28  V041721            16730 non-null   float64
 29  V041722            14231 non-null   float64
 30  V041731            16730 non-null   float64
 31  V041732            3376 non-null    float64
 32  V041741            16730 non-null   float64
 33  V041742            748 non-null     float64
 34  V0418              49691 non-null   float64
 35  V0419              49691 non-null   float64
 36  V0420              25903 non-null   float64
 37  V0421              6452 non-null    float64
 38  V0422              41505 non-null   float64
 39  V0423              6452 non-null    float64
 40  V0424              128740 non-null  float64
 41  V0425              113671 non-null  float64
 42  V0426              34118 non-null   float64
 43  V0427              13085 non-null   float64
 44  V0428              106178 non-null  float64
 45  V0429              101188 non-null  float64
 46  V0430              107528 non-null  float64
 47  ANOS_ESTUDO        178431 non-null  int64  
 48  PESO               178431 non-null  float64
 49  PESO_FINAL         178431 non-null  float64
 50  RENDA_TOTAL        178431 non-null  float64
 51  NIVEL_INSTRUCAO    178431 non-null  int64  
 52  RENDA_DISP_PC      178369 non-null  float64
 53  RENDA_MONET_PC     178369 non-null  float64
 54  RENDA_NAO_MONET_PC 178369 non-null  float64
 55  DEDUCAO_PC         178369 non-null  float64
```

```
dtypes: float64(36), int64(20)
memory usage: 76.2 MB
```

In [59]:

```
#importação de arquivo morador qualidade de vida
df_morador_qualivida = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\MORA
df_morador_qualivida.head()
```

Out[59]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	COD_INFOR
0	11	1103	1	110005400	1	1	
1	11	1103	1	110005400	1	1	
2	11	1103	1	110005400	1	1	
3	11	1103	1	110005400	1	1	
4	11	1103	1	110005400	2	1	

In [13]:
```
df_morador_qualivida.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178369 entries, 0 to 178368
Data columns (total 72 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   UF                  178369 non-null  int64  
 1   ESTRATO_POF         178369 non-null  int64  
 2   TIPO_SITUACAO_REG   178369 non-null  int64  
 3   COD_UPA             178369 non-null  int64  
 4   NUM_DOM             178369 non-null  int64  
 5   NUM_UC              178369 non-null  int64  
 6   COD_INFORMANTE      178369 non-null  int64  
 7   CONTAGEM_PONDERADA  178369 non-null  float64
 8   FUNCAO_PERDA        178369 non-null  float64
 9   V201                178369 non-null  int64  
 10  V202                178369 non-null  int64  
 11  V204                178369 non-null  int64  
 12  V205                178369 non-null  int64  
 13  V206                178369 non-null  int64  
 14  V207                178369 non-null  int64  
 15  V208                178369 non-null  int64  
 16  V209                178369 non-null  int64  
 17  V210                178369 non-null  int64  
 18  V211                178369 non-null  int64  
 19  V212                178369 non-null  int64  
 20  V214                178369 non-null  int64  
 21  V215                178369 non-null  int64  
 22  V216                178369 non-null  int64  
 23  V217                178369 non-null  int64  
 24  V301                178369 non-null  int64  
 25  V302                178369 non-null  int64  
 26  V303                178369 non-null  int64  
 27  V304                178369 non-null  int64  
 28  V305                178369 non-null  int64  
 29  V306                178369 non-null  int64  
 30  V307                178369 non-null  int64  
 31  V308                178369 non-null  int64  
 32  V401                178369 non-null  int64  
 33  V402                178369 non-null  int64  
 34  V403                178369 non-null  int64  
 35  V501                178369 non-null  int64  
 36  V502                178369 non-null  int64  
 37  V503                178369 non-null  int64  
 38  V504                178369 non-null  int64  
 39  V505                178369 non-null  int64  
 40  V506                178369 non-null  int64  
 41  V601                178369 non-null  int64  
 42  V602                178369 non-null  int64  
 43  V603                178369 non-null  int64  
 44  V604                178369 non-null  int64  
 45  V605                178369 non-null  int64  
 46  V606                178369 non-null  int64  
 47  V607                178369 non-null  int64  
 48  V608                178369 non-null  int64  
 49  V609                178369 non-null  int64  
 50  V610                178369 non-null  int64  
 51  V611                178369 non-null  int64  
 52  V701                178369 non-null  int64  
 53  V702                178369 non-null  int64  
 54  V703                178369 non-null  int64  
 55  V704                178369 non-null  int64  
```

```
 56  V801                178369 non-null  int64
 57  V802                178369 non-null  int64
 58  V901                178369 non-null  int64
 59  V902                178369 non-null  int64
 60  GRANDE_REGIAO       178369 non-null  int64
 61  C1                  178369 non-null  int64
 62  C2                  178369 non-null  int64
 63  C3                  178369 non-null  int64
 64  C4                  178369 non-null  int64
 65  C5                  178369 non-null  int64
 66  C6                  178369 non-null  int64
 67  C7                  178369 non-null  int64
 68  RENDA_DISP_PC       178369 non-null  float64
 69  RENDA_DISP_PC_SS    178369 non-null  float64
 70  PESO                178369 non-null  float64
 71  PESO_FINAL          178369 non-null  float64
dtypes: float64(6), int64(66)
memory usage: 98.0 MB
```

In [60]:

```
#importação de arquivo servico não monetário pof2
df_serviconaomonetpof2 = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\SE
df_serviconaomonetpof2.head()
```

Out[60]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	QUADRO	S
0	11	1105	1	110000620	4	1	8	
1	11	1105	1	110000620	4	1	10	
2	11	1105	1	110000620	4	1	13	
3	11	1105	1	110000620	7	1	12	
4	11	1105	1	110000620	10	1	6	

In [39]:
```
df_serviconaomonetpof2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14711 entries, 0 to 14710
Data columns (total 24 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   UF                 14711 non-null  int64  
 1   ESTRATO_POF        14711 non-null  int64  
 2   TIPO_SITUACAO_REG  14711 non-null  int64  
 3   COD_UPA            14711 non-null  int64  
 4   NUM_DOM            14711 non-null  int64  
 5   NUM_UC             14711 non-null  int64  
 6   QUADRO             14711 non-null  int64  
 7   SEQ                14711 non-null  int64  
 8   V9001              14711 non-null  int64  
 9   V9002              14711 non-null  int64  
 10  V8000              14711 non-null  float64
 11  V9010              2512 non-null   float64
 12  V9011              2512 non-null   float64
 13  V1904              68 non-null     float64
 14  V1905              325 non-null    float64
 15  DEFLATOR           14500 non-null  float64
 16  V8000_DEFLA        14711 non-null  float64
 17  V1904_DEFLA        64 non-null     float64
 18  COD_IMPUT_VALOR    14711 non-null  int64  
 19  FATOR_ANUALIZACAO  14711 non-null  int64  
 20  PESO               14711 non-null  float64
 21  PESO_FINAL         14711 non-null  float64
 22  RENDA_TOTAL        14711 non-null  float64
 23  V9004              1460 non-null   float64
dtypes: float64(12), int64(12)
memory usage: 2.7 MB
```

In [61]:
```
#importação de arquivo cardeneta coletiva
df_cardeneta = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\caderneta_co
df_cardeneta.head()
```

Out[61]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	QUADRO	S
0	11	1103	1	110005400	1	1	67	
1	11	1103	1	110005400	1	1	67	
2	11	1103	1	110005400	1	1	67	
3	11	1103	1	110005400	1	1	67	
4	11	1103	1	110005400	1	1	67	

In [17]:
```
df_cardeneta.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 789995 entries, 0 to 789994
Data columns (total 23 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   UF                  789995 non-null  int64  
 1   ESTRATO_POF         789995 non-null  int64  
 2   TIPO_SITUACAO_REG   789995 non-null  int64  
 3   COD_UPA             789995 non-null  int64  
 4   NUM_DOM             789995 non-null  int64  
 5   NUM_UC              789995 non-null  int64  
 6   QUADRO              789995 non-null  int64  
 7   SEQ                 789995 non-null  int64  
 8   V9001               789995 non-null  int64  
 9   V9002               789995 non-null  int64  
 10  V8000               789995 non-null  float64
 11  DEFLATOR            735853 non-null  float64
 12  V8000_DEFLA         789995 non-null  float64
 13  COD_IMPUT_VALOR     789995 non-null  int64  
 14  FATOR_ANUALIZACAO   789995 non-null  int64  
 15  PESO                789995 non-null  float64
 16  PESO_FINAL          789995 non-null  float64
 17  RENDA_TOTAL         789995 non-null  float64
 18  V9005               725943 non-null  float64
 19  V9007               725943 non-null  float64
 20  V9009               279618 non-null  float64
 21  QTD_FINAL           718561 non-null  float64
 22  V9004               789995 non-null  int64  
dtypes: float64(10), int64(13)
memory usage: 138.6 MB
```

In [62]:
```
#importação de arquivo consumo alimentar
df_consumo_alimentar = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\cons
df_consumo_alimentar.head()
```

Out[62]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	COD_INFOR
0	11	1101	1	110000016	2	1	
1	11	1101	1	110000016	2	1	
2	11	1101	1	110000016	2	1	
3	11	1101	1	110000016	2	1	
4	11	1101	1	110000016	2	1	

In [19]:
```
df_consumo_alimentar.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1175390 entries, 0 to 1175389
Data columns (total 67 columns):
 #   Column                    Non-Null Count    Dtype
---  ------                    --------------    -----
 0   UF                        1175390 non-null  int64
 1   ESTRATO_POF               1175390 non-null  int64
 2   TIPO_SITUACAO_REG         1175390 non-null  int64
 3   COD_UPA                   1175390 non-null  int64
 4   NUM_DOM                   1175390 non-null  int64
 5   NUM_UC                    1175390 non-null  int64
 6   COD_INFOR.MANTE           1175390 non-null  int64
 7   QUADRO                    1175390 non-null  int64
 8   SEQ                       1175390 non-null  int64
 9   V9005                     1007126 non-null  float64
 10  V9007                     1007126 non-null  float64
 11  V9001                     1175390 non-null  int64
 12  V9015                     1175390 non-null  object
 13  V9016                     176499 non-null   float64
 14  V9017                     1175390 non-null  int64
 15  V9018                     1175390 non-null  int64
 16  V9019                     1007126 non-null  float64
 17  V9020                     1007126 non-null  float64
 18  V9021                     1007126 non-null  float64
 19  V9022                     1007126 non-null  float64
 20  V9023                     1007126 non-null  float64
 21  V9024                     1007126 non-null  float64
 22  V9025                     1007126 non-null  float64
 23  V9026                     1007126 non-null  float64
 24  V9027                     1007126 non-null  float64
 25  V9028                     1007126 non-null  float64
 26  V9029                     1007126 non-null  float64
 27  V9030                     1007126 non-null  float64
 28  COD_UNIDADE_MEDIDA_FINAL  1007126 non-null  float64
 29  COD_PREPARACAO_FINAL      1007126 non-null  float64
 30  GRAMATURA1                1007126 non-null  float64
 31  QTD                       1175390 non-null  float64
 32  COD_TBCA                  1175390 non-null  object
 33  ENERGIA_KCAL              1175390 non-null  float64
 34  ENERGIA_KJ                1175390 non-null  float64
 35  PTN                       1175390 non-null  float64
 36  CHOTOT                    1175390 non-null  float64
 37  FIBRA                     1175390 non-null  float64
 38  LIP                       1175390 non-null  float64
 39  COLEST                    1175390 non-null  float64
 40  AGSAT                     1175390 non-null  float64
 41  AGMONO                    1175390 non-null  float64
 42  AGPOLI                    1175390 non-null  float64
 43  AGTRANS                   1175390 non-null  float64
 44  CALCIO                    1175390 non-null  float64
 45  FERRO                     1175390 non-null  float64
 46  SODIO                     1175390 non-null  float64
 47  MAGNESIO                  1175390 non-null  float64
 48  FOSFORO                   1175390 non-null  float64
 49  POTASSIO                  1175390 non-null  float64
 50  COBRE                     1175390 non-null  float64
 51  ZINCO                     1175390 non-null  float64
 52  VITA_RAE                  1175390 non-null  float64
 53  TIAMINA                   1175390 non-null  float64
 54  RIBOFLAVINA               1175390 non-null  float64
 55  NIACINA                   1175390 non-null  float64
```

```
 56   PIRIDOXAMINA                1175390 non-null  float64
 57   COBALAMINA                  1175390 non-null  float64
 58   VITD                        1175390 non-null  float64
 59   VITE                        1175390 non-null  float64
 60   VITC                        1175390 non-null  float64
 61   FOLATO                      1175390 non-null  float64
 62   PESO                        1175390 non-null  float64
 63   PESO_FINAL                  1175390 non-null  float64
 64   RENDA_TOTAL                 1175390 non-null  float64
 65   DIA_SEMANA                  1175390 non-null  object
 66   DIA_ATIPICO                 1175390 non-null  int64
dtypes: float64(51), int64(13), object(3)
memory usage: 600.8+ MB
```

In [63]:

```
#importação de arquivo despesa individual
df_despesa_individual = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\des
df_despesa_individual.head()
```

Out[63]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	COD_INFOR
0	11	1103	1	110005400	1	1	
1	11	1103	1	110005400	1	1	
2	11	1103	1	110005400	1	1	
3	11	1103	1	110005400	1	1	
4	11	1103	1	110005400	1	1	

In [21]:
```
df_despesa_individual.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1836032 entries, 0 to 1836031
Data columns (total 25 columns):
 #   Column             Dtype  
---  ------             -----  
 0   UF                 int64  
 1   ESTRATO_POF        int64  
 2   TIPO_SITUACAO_REG  int64  
 3   COD_UPA            int64  
 4   NUM_DOM            int64  
 5   NUM_UC             int64  
 6   COD_INFORMANTE     int64  
 7   QUADRO             int64  
 8   SEQ                int64  
 9   V9001              int64  
 10  V9002              int64  
 11  V8000              float64
 12  V9010              float64
 13  V9011              float64
 14  V9012              float64
 15  V4104              float64
 16  V4105              float64
 17  DEFLATOR           float64
 18  V8000_DEFLA        float64
 19  COD_IMPUT_VALOR    int64  
 20  FATOR_ANUALIZACAO  int64  
 21  PESO               float64
 22  PESO_FINAL         float64
 23  RENDA_TOTAL        float64
 24  V9004              float64
dtypes: float64(12), int64(13)
memory usage: 350.2 MB
```

In [64]:
```
#importação de arquivo outros rendimento
df_outro_rendimento = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\outro
df_outro_rendimento.head()
```

Out[64]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	COD_INFOR
0	11	1103	1	110005400	4	1	
1	11	1103	1	110005400	4	1	
2	11	1103	1	110005400	5	1	
3	11	1103	1	110005400	5	1	
4	11	1103	1	110005400	5	1	

In [23]:
```
df_outro_rendimento.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 206108 entries, 0 to 206107
Data columns (total 22 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   UF                  206108 non-null  int64  
 1   ESTRATO_POF         206108 non-null  int64  
 2   TIPO_SITUACAO_REG   206108 non-null  int64  
 3   COD_UPA             206108 non-null  int64  
 4   NUM_DOM             206108 non-null  int64  
 5   NUM_UC              206108 non-null  int64  
 6   COD_INFORMANTE      206108 non-null  int64  
 7   QUADRO              206108 non-null  int64  
 8   SEQ                 206108 non-null  int64  
 9   V9001               206108 non-null  int64  
 10  V8500               206108 non-null  float64
 11  V8501               65591 non-null   float64
 12  V9010               65043 non-null   float64
 13  V9011               65043 non-null   float64
 14  DEFLATOR            193013 non-null  float64
 15  V8500_DEFLA         206108 non-null  float64
 16  V8501_DEFLA         30661 non-null   float64
 17  COD_IMPUT_VALOR     206108 non-null  int64  
 18  FATOR_ANUALIZACAO   206108 non-null  int64  
 19  PESO                206108 non-null  float64
 20  PESO_FINAL          206108 non-null  float64
 21  RENDA_TOTAL         206108 non-null  float64
dtypes: float64(10), int64(12)
memory usage: 34.6 MB
```

In [65]:
```
#importação de arquivo serviço não monetário pof4
df_serviconaomonetpof4 = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\se
df_serviconaomonetpof4.head()
```

Out[65]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	COD_INFOR
0	11	1103	1	110005400	1	1	
1	11	1103	1	110005400	2	1	
2	11	1103	1	110005400	4	1	
3	11	1103	1	110005400	4	1	
4	11	1103	1	110005400	5	1	

In [25]:
```
df_serviconaomonetpof4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126409 entries, 0 to 126408
Data columns (total 24 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   UF                  126409 non-null  int64  
 1   ESTRATO_POF         126409 non-null  int64  
 2   TIPO_SITUACAO_REG   126409 non-null  int64  
 3   COD_UPA             126409 non-null  int64  
 4   NUM_DOM             126409 non-null  int64  
 5   NUM_UC              126409 non-null  int64  
 6   COD_INFORMANTE      126409 non-null  int64  
 7   QUADRO              126409 non-null  int64  
 8   SEQ                 126409 non-null  int64  
 9   V9001               126409 non-null  int64  
 10  V9002               126409 non-null  int64  
 11  V8000               126409 non-null  float64
 12  V9010               41041 non-null   float64
 13  V9011               41041 non-null   float64
 14  V4104               6608 non-null    float64
 15  V4105               6608 non-null    float64
 16  DEFLATOR            113555 non-null  float64
 17  V8000_DEFLA         126409 non-null  float64
 18  COD_IMPUT_VALOR     126409 non-null  int64  
 19  FATOR_ANUALIZACAO   126409 non-null  int64  
 20  PESO                126409 non-null  float64
 21  PESO_FINAL          126409 non-null  float64
 22  RENDA_TOTAL         126409 non-null  float64
 23  V9004               91837 non-null   float64
dtypes: float64(11), int64(13)
memory usage: 23.1 MB
```

In [78]:
```
#importação de arquivo outros caracteristicas dieta
df_caracteristica_dieta = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\c
df_caracteristica_dieta.head()
```

Out[78]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	COD_INFOR
0	11	1103	1	110005400	2	1	
1	11	1103	1	110005400	2	1	
2	11	1103	1	110005400	5	1	
3	11	1103	1	110005400	8	1	
4	11	1103	1	110005400	9	1	

In [29]:
```
df_caracteristica_dieta.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46164 entries, 0 to 46163
Data columns (total 31 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   UF                 46164 non-null  int64  
 1   ESTRATO_POF        46164 non-null  int64  
 2   TIPO_SITUACAO_REG  46164 non-null  int64  
 3   COD_UPA            46164 non-null  int64  
 4   NUM_DOM            46164 non-null  int64  
 5   NUM_UC             46164 non-null  int64  
 6   COD_INFORMANTE     46164 non-null  int64  
 7   V7101              46164 non-null  int64  
 8   V7102              46164 non-null  int64  
 9   V71031             46164 non-null  int64  
 10  V71032             46164 non-null  int64  
 11  V71033             46164 non-null  int64  
 12  V71034             46164 non-null  int64  
 13  V71035             46164 non-null  int64  
 14  V71036             46164 non-null  int64  
 15  V71037             46164 non-null  int64  
 16  V71038             46164 non-null  int64  
 17  V7104              46164 non-null  int64  
 18  V71051             6475 non-null   float64
 19  V71052             6475 non-null   float64
 20  V71053             6475 non-null   float64
 21  V71054             6475 non-null   float64
 22  V71055             6475 non-null   float64
 23  V71056             6475 non-null   float64
 24  V71A01             24704 non-null  float64
 25  V71A02             24704 non-null  float64
 26  V72C01             46164 non-null  int64  
 27  V72C02             46164 non-null  int64  
 28  PESO               46164 non-null  float64
 29  PESO_FINAL         46164 non-null  float64
 30  RENDA_TOTAL        46164 non-null  float64
dtypes: float64(11), int64(20)
memory usage: 10.9 MB
```

In [67]:
```
#importação de arquivo condições de vida
df_condicoes_vida = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\condicc
df_condicoes_vida.head()
```

Out[67]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	COD_INFOR
0	11	1103	1	110005400	1	1	
1	11	1103	1	110005400	2	1	
2	11	1103	1	110005400	4	1	
3	11	1103	1	110005400	5	1	
4	11	1103	1	110005400	6	1	

In [33]:
```
df_condicoes_vida.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58039 entries, 0 to 58038
Data columns (total 55 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   UF                 58039 non-null  int64  
 1   ESTRATO_POF        58039 non-null  int64  
 2   TIPO_SITUACAO_REG  58039 non-null  int64  
 3   COD_UPA            58039 non-null  int64  
 4   NUM_DOM            58039 non-null  int64  
 5   NUM_UC             58039 non-null  int64  
 6   COD_INFORMANTE     58039 non-null  int64  
 7   V6101              58039 non-null  int64  
 8   V6102              58039 non-null  int64  
 9   V6103              58039 non-null  int64  
 10  V61041             58039 non-null  int64  
 11  V61042             58039 non-null  int64  
 12  V61043             58039 non-null  int64  
 13  V61044             58039 non-null  int64  
 14  V61045             58039 non-null  int64  
 15  V61046             58039 non-null  int64  
 16  V61051             58039 non-null  int64  
 17  V61052             58039 non-null  int64  
 18  V61053             58039 non-null  int64  
 19  V61054             58039 non-null  int64  
 20  V61055             58039 non-null  int64  
 21  V61056             58039 non-null  int64  
 22  V61057             58039 non-null  int64  
 23  V61058             58039 non-null  int64  
 24  V61061             58039 non-null  int64  
 25  V61062             58039 non-null  int64  
 26  V61063             58039 non-null  int64  
 27  V61064             58039 non-null  int64  
 28  V61065             58039 non-null  int64  
 29  V61066             58039 non-null  int64  
 30  V61067             58039 non-null  int64  
 31  V61068             58039 non-null  int64  
 32  V61069             58039 non-null  int64  
 33  V610610            58039 non-null  int64  
 34  V610611            58039 non-null  int64  
 35  V61071             58039 non-null  int64  
 36  V61072             58039 non-null  int64  
 37  V61073             58039 non-null  int64  
 38  V6108              58039 non-null  int64  
 39  V6109              58039 non-null  int64  
 40  V6110              58039 non-null  int64  
 41  V6111              58039 non-null  int64  
 42  V6112              23273 non-null  float64
 43  V6113              23273 non-null  float64
 44  V6114              23273 non-null  float64
 45  V6115              23273 non-null  float64
 46  V6116              13527 non-null  float64
 47  V6117              13527 non-null  float64
 48  V6118              13527 non-null  float64
 49  V6119              13527 non-null  float64
 50  V6120              13527 non-null  float64
 51  V6121              13527 non-null  float64
 52  PESO               58039 non-null  float64
 53  PESO_FINAL         58039 non-null  float64
 54  RENDA_TOTAL        58039 non-null  float64
```

```
dtypes: float64(13), int64(42)
memory usage: 24.4 MB
```

In [68]:

```
#importação de arquivo inventário
df_inventario = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\inventario.
df_inventario.head()
```

Out[68]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	QUADRO	S
0	11	1103	1	110005400	1	1	14	
1	11	1103	1	110005400	1	1	14	
2	11	1103	1	110005400	1	1	14	
3	11	1103	1	110005400	1	1	14	
4	11	1103	1	110005400	1	1	14	

In [35]:

```
df_inventario.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 870354 entries, 0 to 870353
Data columns (total 16 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   UF                 870354 non-null  int64  
 1   ESTRATO_POF        870354 non-null  int64  
 2   TIPO_SITUACAO_REG  870354 non-null  int64  
 3   COD_UPA            870354 non-null  int64  
 4   NUM_DOM            870354 non-null  int64  
 5   NUM_UC             870354 non-null  int64  
 6   QUADRO             870354 non-null  int64  
 7   SEQ                870354 non-null  int64  
 8   V9001              870354 non-null  int64  
 9   V9005              870354 non-null  int64  
 10  V9002              870354 non-null  int64  
 11  V1404              870354 non-null  int64  
 12  V9012              870354 non-null  int64  
 13  PESO               870354 non-null  float64
 14  PESO_FINAL         870354 non-null  float64
 15  RENDA_TOTAL        870354 non-null  float64
dtypes: float64(3), int64(13)
memory usage: 106.2 MB
```

In [69]:

```
#importação de arquivo restrição de produtos serviços e saúde
df_restricao = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\restricao_pr
df_restricao.head()
```

Out[69]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	COD_INFOR
0	11	1103	1	110005400	12	1	
1	11	1103	1	110005400	13	1	
2	11	1103	1	110005400	13	1	
3	11	1103	1	110005400	13	1	
4	11	1103	1	110005400	13	1	

In [37]:
```
df_restricao.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40863 entries, 0 to 40862
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   UF                 40863 non-null  int64
 1   ESTRATO_POF        40863 non-null  int64
 2   TIPO_SITUACAO_REG  40863 non-null  int64
 3   COD_UPA            40863 non-null  int64
 4   NUM_DOM            40863 non-null  int64
 5   NUM_UC             40863 non-null  int64
 6   COD_INFORMANTE     40863 non-null  int64
 7   QUADRO             40863 non-null  int64
 8   SEQ                40863 non-null  int64
 9   V9001              40863 non-null  int64
 10  V9013              40863 non-null  int64
 11  PESO               40863 non-null  float64
 12  PESO_FINAL         40863 non-null  float64
 13  RENDA_TOTAL        40863 non-null  float64
dtypes: float64(3), int64(11)
memory usage: 4.4 MB
```

In [79]:
```
#importação de arquivo rendimento trabalho
df_rendimento_trabalho = pd.read_csv(r'..\Analise_Exploratoria\Dados_CSV\re
df_rendimento_trabalho.head()
```

Out[79]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	COD_INFOR
0	11	1103	1	110005400	1	1	
1	11	1103	1	110005400	2	1	
2	11	1103	1	110005400	2	1	
3	11	1103	1	110005400	4	1	
4	11	1103	1	110005400	4	1	

Análise Geral Automática

Em cada código encontra-se as análises descritivas da tabela como:

- Tipos de dados
- Correlação entre colunas
- Número de linhas duplicadas
- Número de linhas nulas
- Insights (Alertas) Atenção - Devido a quantidade e o tamanho dos arquivos, os códigos a seguir são pesados e podem demandar tempo.

In [14]:
```
profile_aluguel = ProfileReport(df_aluguel)
```

In [15]:
```
profile_aluguel
```

Summarize dataset: 100% 172/172 [00:22<00:00, 3.90it/s, Completed]

Generate report structure: 100% 1/1 [00:07<00:00, 7.38s/it]

Render HTML: 100% 1/1 [00:05<00:00, 5.20s/it]

Overview

Dataset statistics

Number of variables	19
Number of observations	48935
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	7.1 MiB
Average record size in memory	152.0 B

Variable types

Numeric	12
Categorical	7

Alerts

`NUM_UC` has constant value ""	**Constant**
`QUADRO` has constant value ""	**Constant**
`V9001` has constant value ""	**Constant**

Out[15]:

In [16]:
```
profile_despesa_coletiva = ProfileReport(df_despesa_coletiva)
```

In [17]:
```
profile_despesa_coletiva
```

Summarize dataset: 100% 437/437 [02:48<00:00, 1.23it/s, Completed]

Generate report structure: 100% 1/1 [00:11<00:00, 11.34s/it]

Render HTML: 100% 1/1 [00:11<00:00, 11.70s/it]

Most frequent character per category

Decimal Number

Value	Count	Frequency (%)
1	388999	81.3%
2	89573	18.7%

Most occurring scripts

Value	Count	Frequency (%)
Common	478572	100.0%

Most frequent character per script

Common

Value	Count	Frequency (%)
1	388999	81.3%
2	89573	18.7%

Most occurring blocks

Value	Count	Frequency (%)
ASCII	478572	100.0%

Out[17]:

In [80]:
```
profile_despesa_individual = ProfileReport(df_despesa_individual)
```

In [81]:
```
profile_despesa_individual
```

...

In [45]:
```
profile_domicilio = ProfileReport(df_domicilio)
```

In [46]:
```
profile_domicilio
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Out[46]:

In [47]:
```
profile_morador = ProfileReport(df_morador)
```

In [48]:
```
profile_morador
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Out[48]:

In [49]:
```
profile_qualivida = ProfileReport(df_morador_qualivida)
```

In [ ]:
```
profile_qualivida
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
```

In [ ]:
```
profile_servicopof2 = ProfileReport(df_serviconaomonetpof2)
```

In [ ]:
```
profile_servicopof2
```

In [144]:
```
profile_servicopof4 = ProfileReport(df_serviconaomonetpof4)
```

In [145]:
```
profile_servicopof4
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Out[145]:

In [71]:
```
profile_cardeneta = ProfileReport(df_cardeneta)
```

In [72]:
```
profile_cardeneta
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Out[72]:

In [17]:
```
profile_consumo_alimentar = ProfileReport(df_consumo_alimentar)
```

In [77]:
```
profile_consumo_alimentar
```

Out[77]:

In [73]:
```
profile_dieta = ProfileReport(df_caracteristica_dieta)
```

In [74]:
```
profile_dieta
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Out[74]:

In [75]:
```
profile_vida = ProfileReport(df_condicoes_vida)
```

In [76]:
```
profile_vida
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Out[76]:

Pré-Processamento

Os códigos abaixo tem como objetivo realizar uma 'limpeza' nos dados da tabela, afim de facilitar a visualização e evitar problemas na construção de relação entre colunas.

In [103]:
```
#Lista de tabelas
lista_df = [df_aluguel,
df_despesa_coletiva,
df_domicilio,
df_morador,
df_morador_qualivida,
df_serviconaomonetpof2,
df_cardeneta,
df_consumo_alimentar,
df_despesa_individual,
df_outro_rendimento,
df_serviconaomonetpof4, df_rendimento_trabalho, df_restricao, df_inventario
```

In [104]:
```
#Dicionário de códigos de estado
uf_dict = {
    11: 'Rondônia',
    12: 'Acre',
    13: 'Amazonas',
    14: 'Roraima',
    15: 'Pará',
    16: 'Amapá',
    17: 'Tocantins',
    21: 'Maranhão',
    22: 'Piauí',
    23: 'Ceará',
    24: 'Rio Grande do Norte',
    25: 'Paraíba',
    26: 'Pernambuco',
    27: 'Alagoas',
    28: 'Sergipe',
    29: 'Bahia',
    31: 'Minas Gerais',
    32: 'Espírito Santo',
    33: 'Rio de Janeiro',
    35: 'São Paulo',
    41: 'Paraná',
    42: 'Santa Catarina',
    43: 'Rio Grande do Sul',
    50: 'Mato Grosso do Sul',
    51: 'Mato Grosso',
    52: 'Goiás',
    53: 'Distrito Federal'
}
```

Funções de Limpeza

In [105]:
```
#Como o NaN, das colunas, são provavelmente negação do preenchimento da inf
def nan_transform(df):
    df.fillna(-1, inplace = True)
    return df
```

In [106]:
```
# Função para fazer a substituição do código UF para o nome do estado corre
def substitute_uf_code(df):
    df['UF'] = df['UF'].map(uf_dict)
    return df
```

Limpeza

In [107]:
```
#Utilizando a função nan_transform em todas as tabelas
for df in lista_df:
    nan_transform(df)
```

In [18]:
```
#Verificando se foi realizado com sucesso
for df in lista_df:
    print('----------------')
    print('----------------')
    print(df.isna().sum())
```

```
----------------
----------------
UF                   0
ESTRATO_POF          0
TIPO_SITUACAO_REG    0
COD_UPA              0
NUM_DOM              0
NUM_UC               0
QUADRO               0
V9001                0
V9002                0
V8000                0
V9010                0
V9011                0
DEFLATOR             0
V8000_DEFLA          0
COD_IMPUT_VALOR      0
FATOR_ANUALIZACAO    0
PESO                 0
PESO_FINAL           0
```

In [19]:
```
#Verificando se existe linhas duplicadas em todas as tabelas
for df in lista_df:
    print('----------------')
    print('----------------')
    print(df.duplicated().sum())
```

```
----------------
----------------
0
----------------
----------------
0
----------------
----------------
0
----------------
----------------
0
----------------
----------------
0
----------------
----------------
0
----------------
```

In [108]:
```
#Dropando as únicas linhas encontradas duplicadas
df_outro_rendimento.drop_duplicates(inplace = True)
```

In [109]:
```
df_outro_rendimento.duplicated().sum()
```

Out[109]: 0

In [110]:
```
#Substituindo todos os códigos pelo estado correspondente
for df in lista_df:
    substitute_uf_code(df)
```

In [112]:
```
#Verificando
df_aluguel.head(3)
```

Out[112]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	QUAD
0	Rondônia	1103	1	110005400	1	1	
1	Rondônia	1103	1	110005400	4	1	
2	Rondônia	1103	1	110005400	5	1	

In [113]:
```
#Verificando
df_serviconaomonetpof2.head(3)
```

Out[113]:

	UF	ESTRATO_POF	TIPO_SITUACAO_REG	COD_UPA	NUM_DOM	NUM_UC	QUAD
0	Rondônia	1105	1	110000620	4	1	
1	Rondônia	1105	1	110000620	4	1	
2	Rondônia	1105	1	110000620	4	1	

Relação Entre Features/Gráficos

Os códigos abaixo tem como objetivo mostrar de forma gráfica as relações entre as colunas das tabelas: df_serviconaomonetpof2, df_aluguel, df_serviconaomonetpof4, df_consumo_alimentar

Funções de Gráficos

In [22]:
```
#Gráfico de dispersão
def scatter(df, coluna,titulo, coluna2, hue=True, hue_coluna=None):
    plt.title(titulo)

    if hue:
        if hue_coluna is not None:
            ax = sns.scatterplot(x=coluna, y=coluna2, hue=hue_coluna, data=
        else:
            ax = sns.scatterplot(x=coluna, y=coluna2, data=df, alpha = 0.5)
    else:
        ax = sns.scatterplot(x=coluna, y=coluna2, data=df, alpha = 0.5)
    plt.show()
```

In [142]:
```
#Gráfico de barra
def bar(df, coluna, coluna2, titulo, x_label, y_label, hue=True, hue_coluna
    plt.title(titulo)

    if hue:
        if hue_coluna is not None:
            ax = sns.barplot(data=df, x=coluna, y=coluna2, hue=hue_coluna)
        else:
            ax = sns.barplot(data=df, x=coluna, y=coluna2)
    else:
        ax = sns.barplot(data=df, x=coluna, y=coluna2)

    ax.set_xlabel(x_label)
    ax.set_ylabel(y_label)
    if rotation_angle == 90:
        ax.set_xticklabels(ax.get_xticklabels(), rotation=rotation_angle)

    plt.show()
```

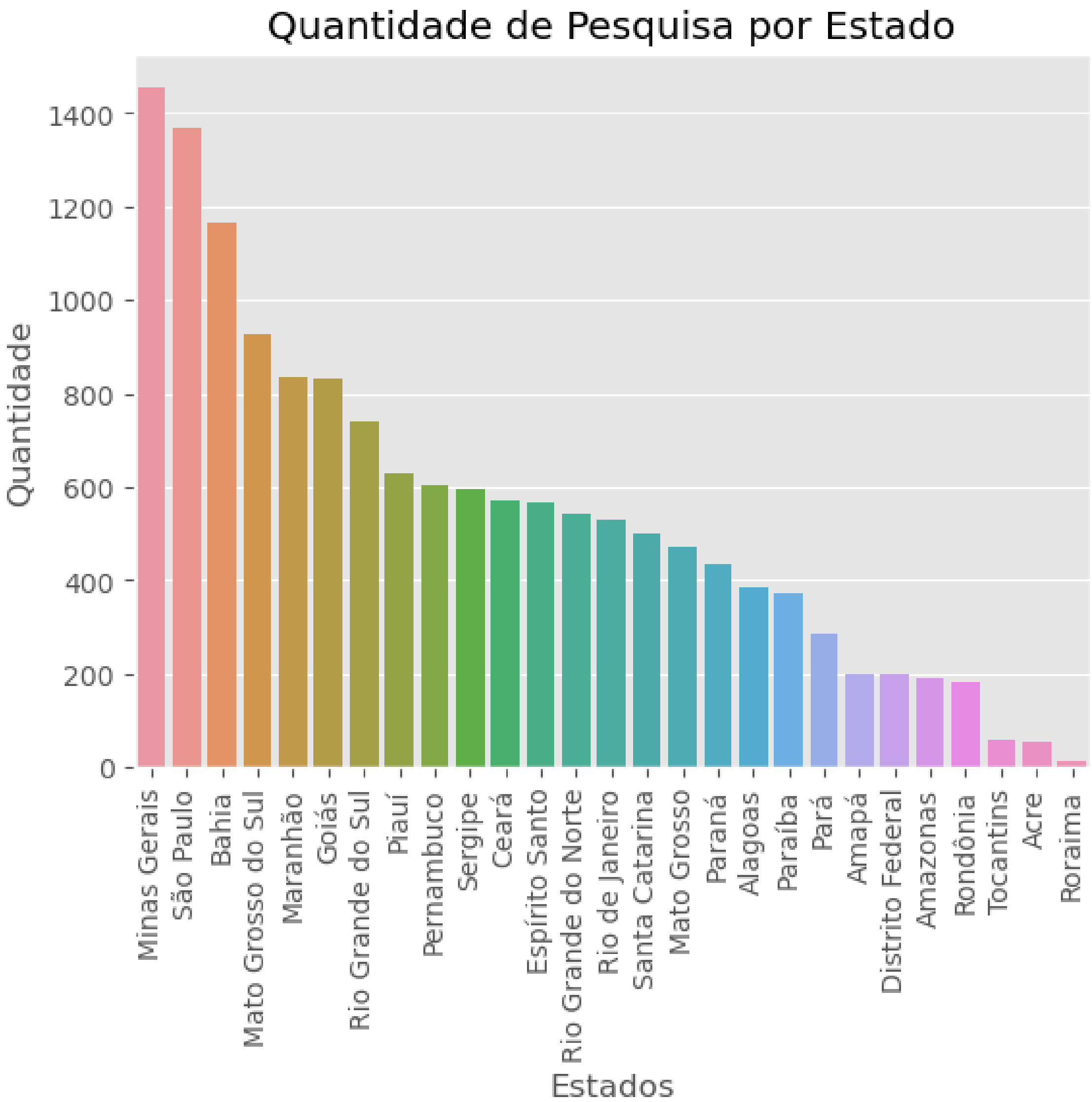
In [126]:
```
#Gráfico de Quantidade
def count(df, coluna, titulo, x_label, y_label, rotation_angle=0):
    ax = sns.countplot(data=df, x=coluna, order=df[coluna].value_counts().i

    if rotation_angle == 90:
        ax.set_xticklabels(ax.get_xticklabels(), rotation=rotation_angle)

    ax.set_xlabel(x_label)
    ax.set_ylabel(y_label)
    ax.set_title(titulo)
    plt.show()
```

Quantidade de UF/NUM_DOM/NUM_UC da Tabela POF2

In [58]:
```
len(df_serviconaomonetpof2)
```

Out[58]:
```
14711
```

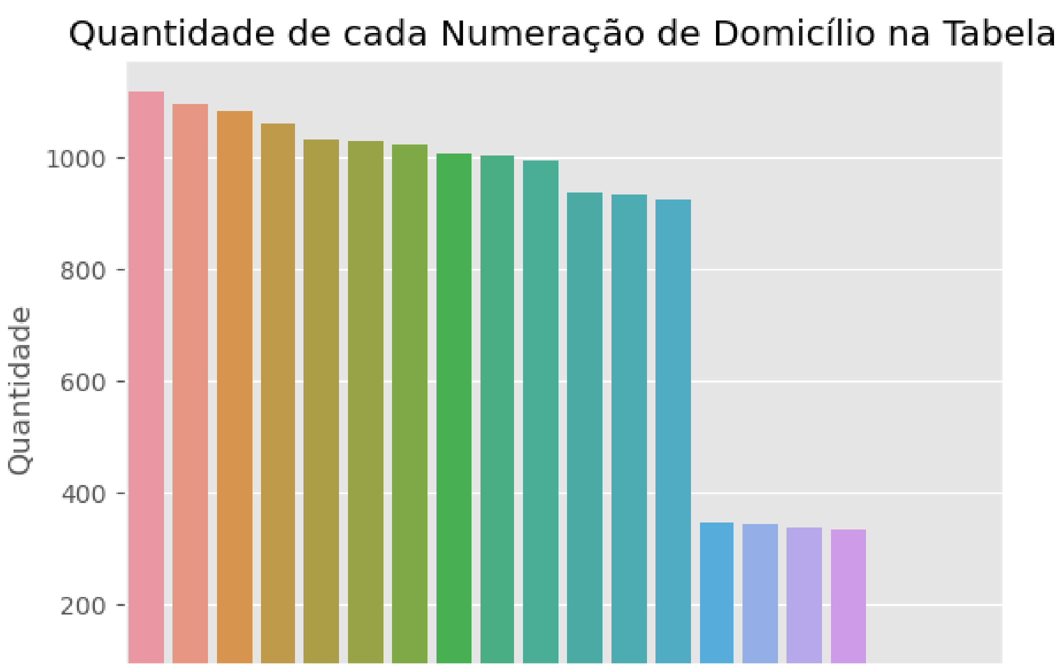
In [127]:
```
count(df_serviconaomonetpof2, 'UF', 'Quantidade de Pesquisa por Estado', 'E
```



In [124]:
```
df_serviconaomonetpof2['UF'].value_counts()
```

Out[124]:
```
Minas Gerais           1454
São Paulo              1367
Bahia                  1167
Mato Grosso do Sul      928
Maranhão                836
Goiás                   832
Rio Grande do Sul       739
Piauí                   630
Pernambuco              604
Sergipe                 595
Ceará                   570
Espírito Santo          565
Rio Grande do Norte     544
Rio de Janeiro          531
Santa Catarina          499
Mato Grosso             470
Paraná                  436
Alagoas                 385
Paraíba                 371
Pará                    288
Amapá                   199
Distrito Federal        198
Amazonas                191
Rondônia                182
Tocantins                60
Acre                     55
Roraima                  15
Name: UF, dtype: int64
```

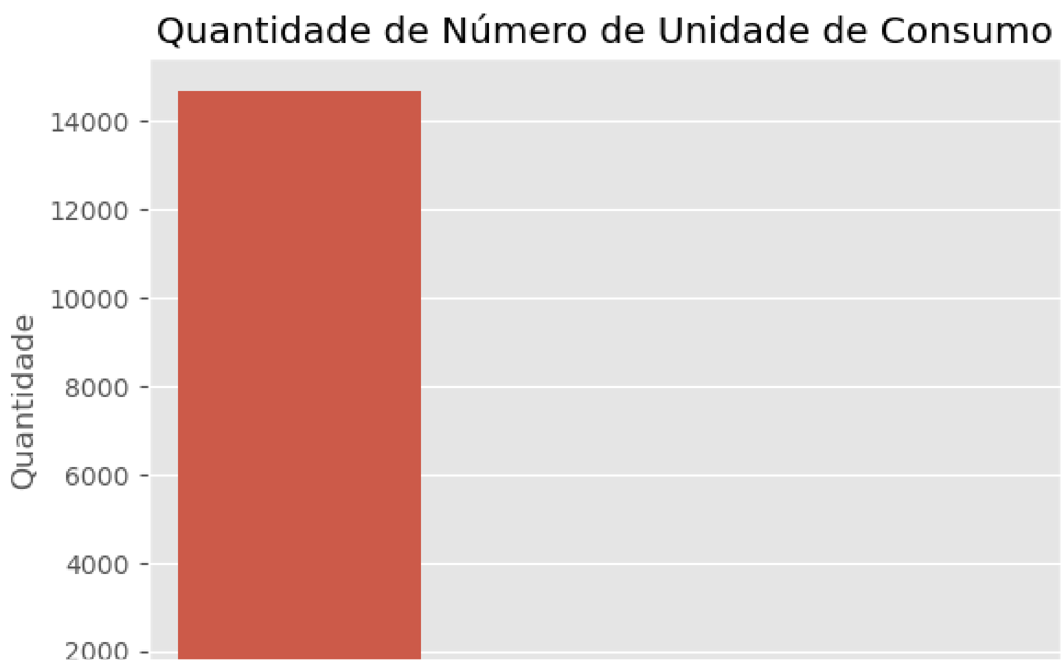
In [128]:
```
count(df_serviconaomonetpof2, 'NUM_DOM', 'Quantidade de cada Numeração de D
```



In [129]:
```
df_serviconaomonetpof2['NUM_DOM'].value_counts()
```

Out[129]:
```
13    1117
11    1094
10    1080
7     1059
12    1032
5     1026
8     1020
9     1006
1     1003
4      993
3      934
6      933
2      921
14     345
15     341
16     337
17     331
20      53
19      49
18      37
Name: NUM_DOM, dtype: int64
```

In [130]:
```
count(df_serviconaomonetpof2, 'NUM_UC', 'Quantidade de Número de Unidade de
```

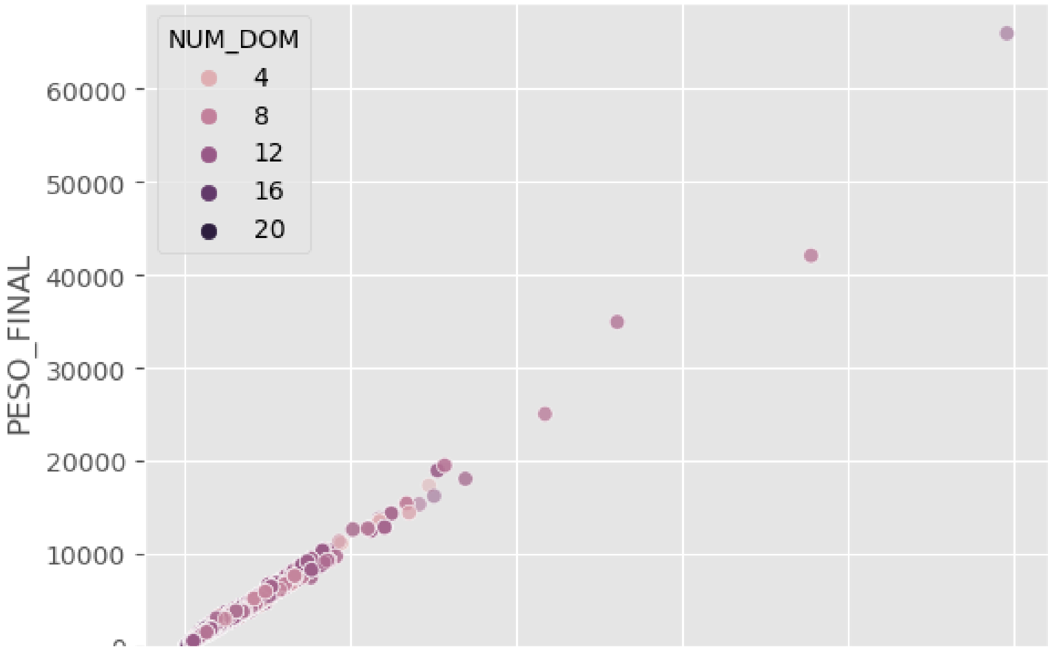


In [131]:
```
df_serviconaomonetpof2['NUM_UC'].value_counts()
```

Out[131]:
```
1    14680
2       30
3        1
Name: NUM_UC, dtype: int64
```

Relação de PESO - PESO FINAL como variam entre os domicílios da Tabela Aluguel

In [133]:
```
scatter(df_aluguel, 'PESO', 'PESO_FINAL', 'Relação entre as variáveis Peso
```

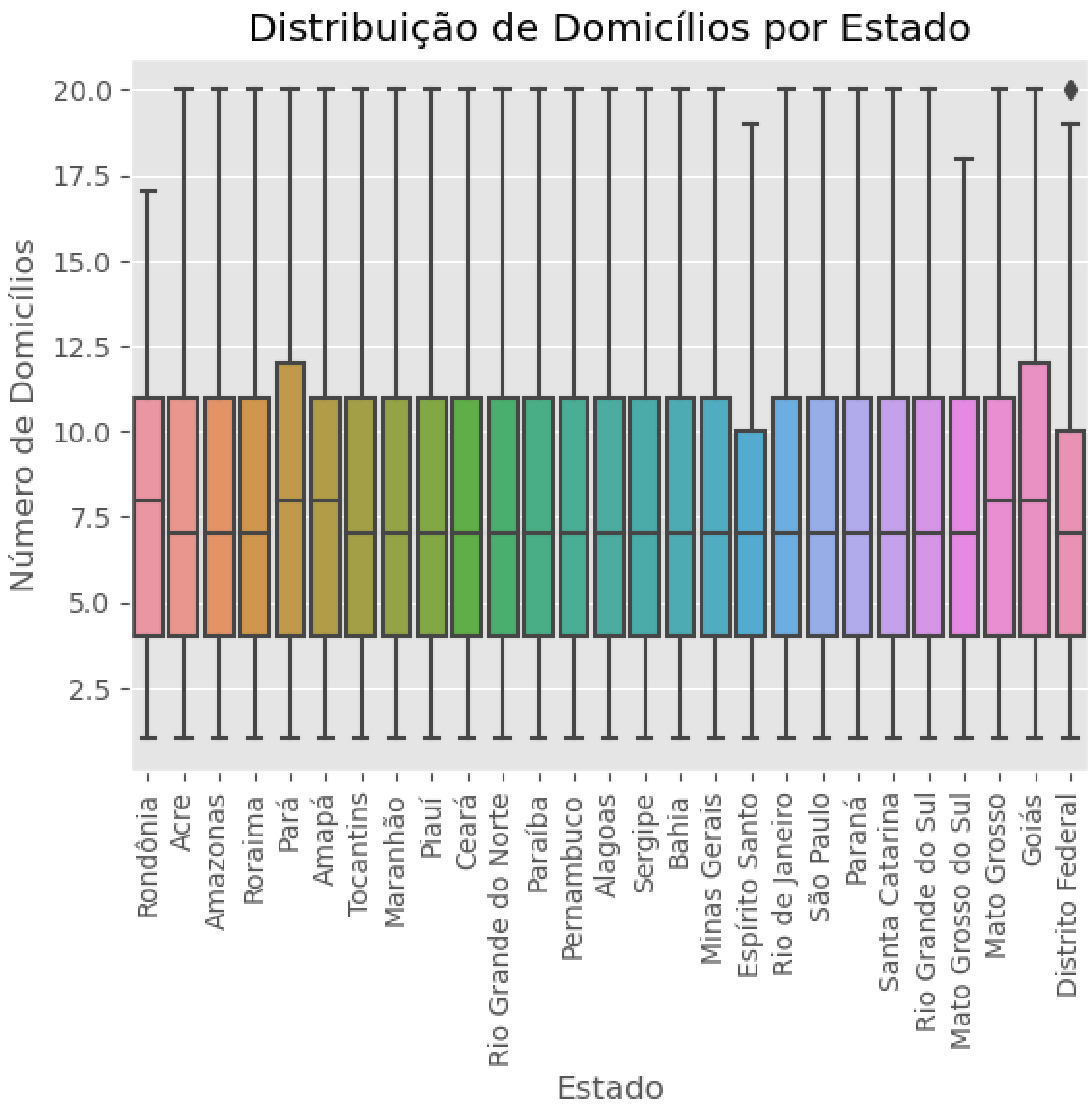


In [134]:
```
df_aluguel['NUM_DOM'].value_counts()
```

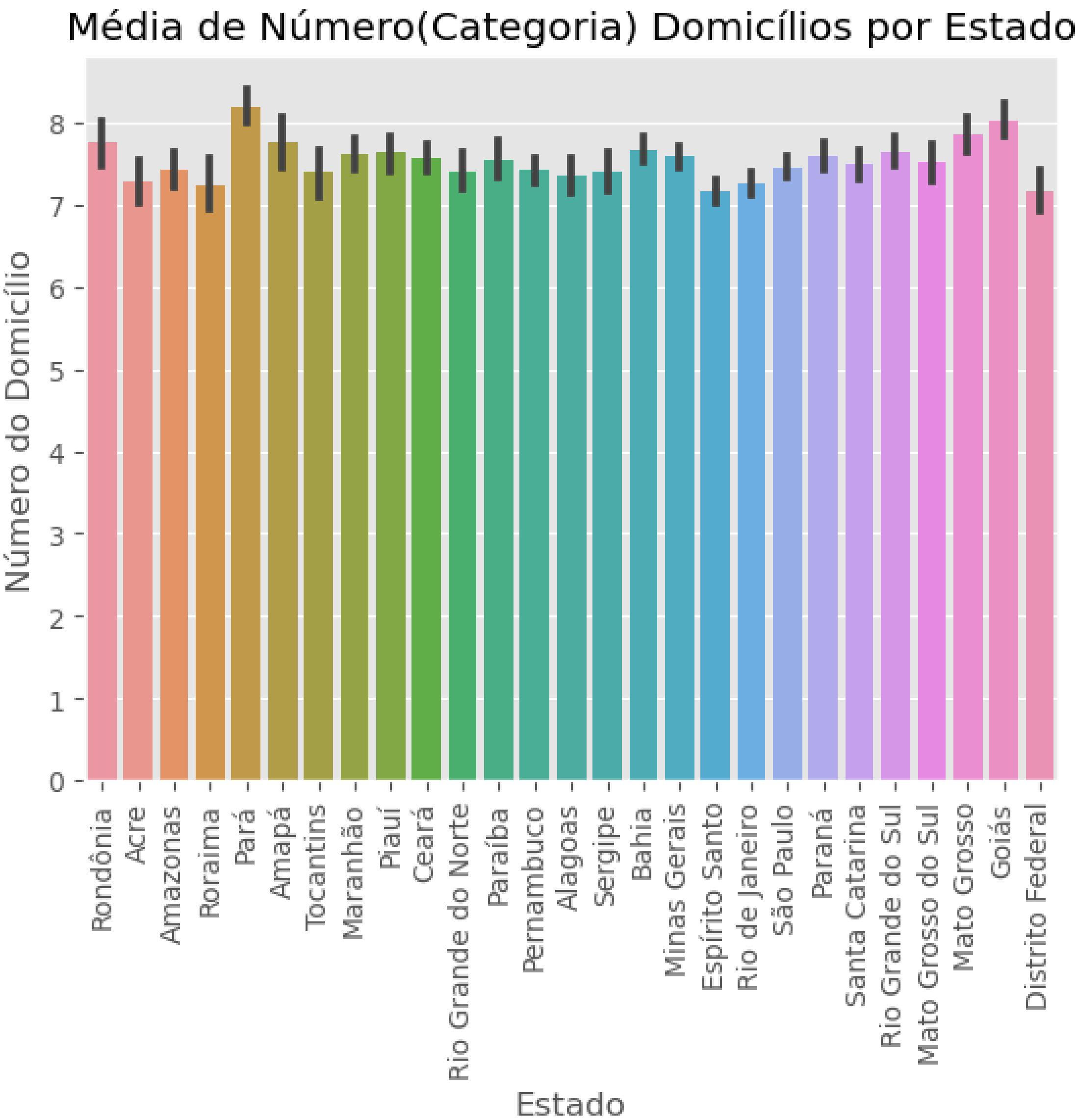
Out[134]:
```
1     3607
2     3582
7     3571
4     3564
5     3546
12    3519
9     3516
3     3505
8     3472
6     3471
10    3455
11    3446
13    3439
15     758
14     754
16     738
17     696
20     102
19      98
18      96
Name: NUM_DOM, dtype: int64
```

Distribuição de Domicílios por Estado e Relação de PESO FINAL por estado / MÉDIA / STD do PESO_FINAL por UF

In [138]:
```
sns.boxplot(x='UF', y='NUM_DOM', data=df_aluguel)
plt.xlabel('Estado')
plt.ylabel('Número de Domicílios')
plt.title('Distribuição de Domicílios por Estado')
plt.xticks(rotation=90)

plt.show()
```



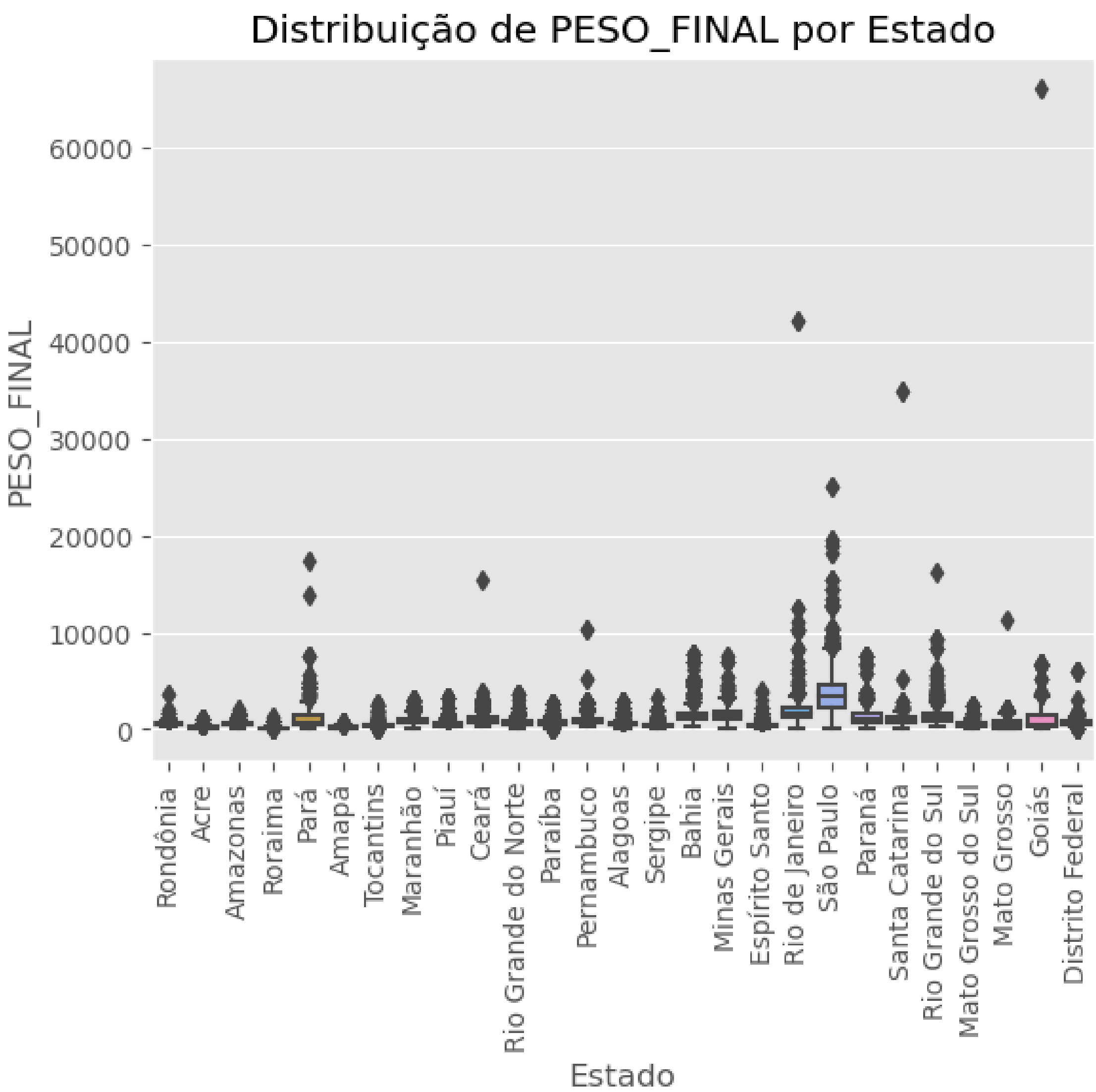
In [151]:
```
bar(df_aluguel, 'UF', 'NUM_DOM', 'Média de Número(Categoria) Domicílios por
```



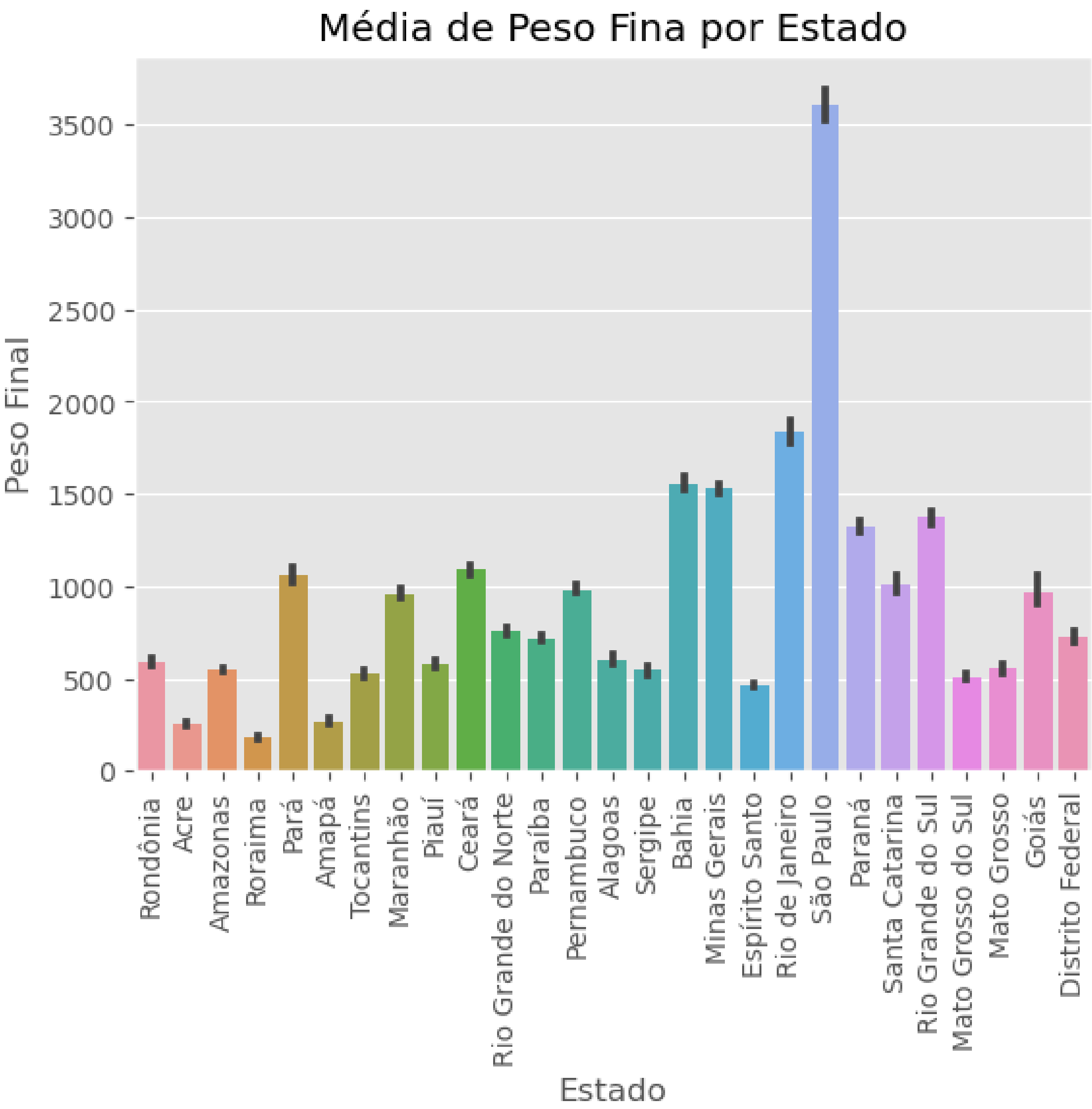
In [149]:
```
#Média
df_aluguel.groupby('UF') \
['NUM_DOM'].mean()
```

Out[149]:
```
UF
Acre                   7.283353
Alagoas                7.352732
Amapá                  7.757530
Amazonas               7.425452
Bahia                  7.667898
Ceará                  7.565851
Distrito Federal       7.160083
Espírito Santo         7.169607
Goiás                  8.015682
Maranhão               7.616569
Mato Grosso            7.848237
Mato Grosso do Sul     7.511356
Minas Gerais           7.585658
Paraná                 7.598371
Paraíba                7.553451
Pará                   8.174001
Pernambuco             7.412880
Piauí                  7.629042
Rio Grande do Norte    7.405450
Rio Grande do Sul      7.637300
Rio de Janeiro         7.252664
Rondônia               7.752475
Roraima                7.242424
Santa Catarina         7.485473
Sergipe                7.398614
São Paulo              7.450650
Tocantins              7.393524
Name: NUM_DOM, dtype: float64
```

In [148]:
```
sns.boxplot(x='UF', y='PESO_FINAL', data=df_aluguel)
plt.xlabel('Estado')
plt.ylabel('PESO_FINAL')
plt.title('Distribuição de PESO_FINAL por Estado')
plt.xticks(rotation=90)

plt.show()
```



In [150]:
```
bar(df_aluguel, 'UF', 'PESO_FINAL', 'Média de Peso Fina por Estado', 'Estad
```



In [100]:
```
#Média
df_aluguel.groupby('UF') \
['PESO_FINAL'].mean()
```

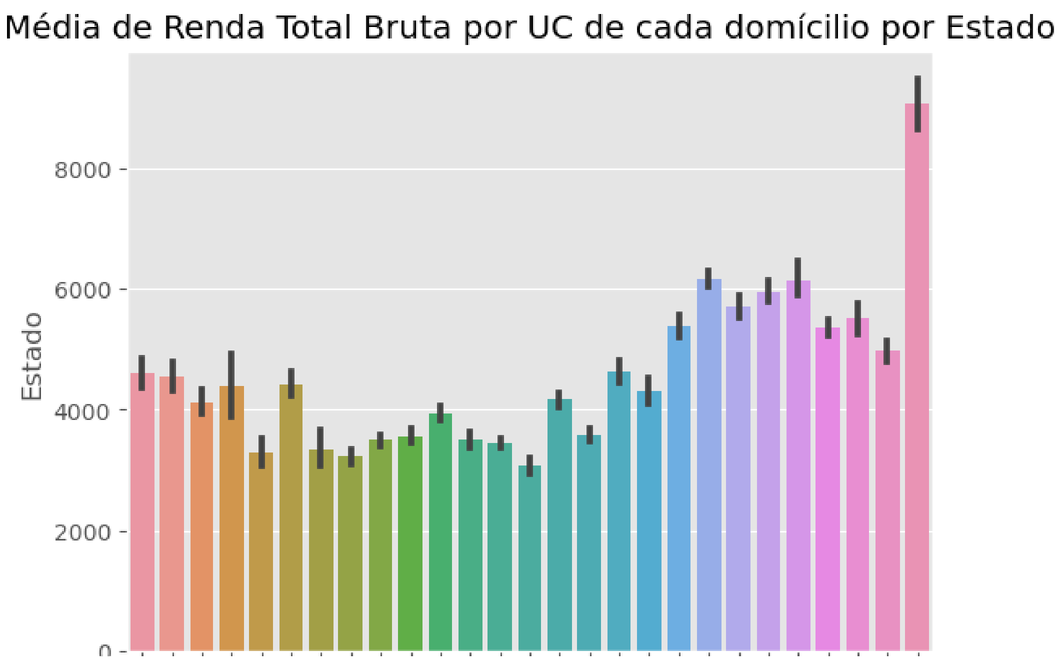
Out[100]:
```
UF
11     586.161034
12     250.016314
13     546.649754
14     181.250036
15    1060.760585
16     268.984027
17     526.550684
21     960.530468
22     578.704114
23    1088.160815
24     755.977283
25     719.895305
26     981.693586
27     604.713162
28     542.111076
29    1555.421831
31    1529.295635
32     465.324208
33    1837.019108
35    3605.279020
41    1320.580345
42    1004.815445
43    1370.491888
50     506.315969
51     554.079763
52     970.814120
53     721.831596
Name: PESO_FINAL, dtype: float64
```

In [101]:
```
#Desvio Padrão
df_aluguel.groupby('UF') \
['PESO_FINAL'].std()
```

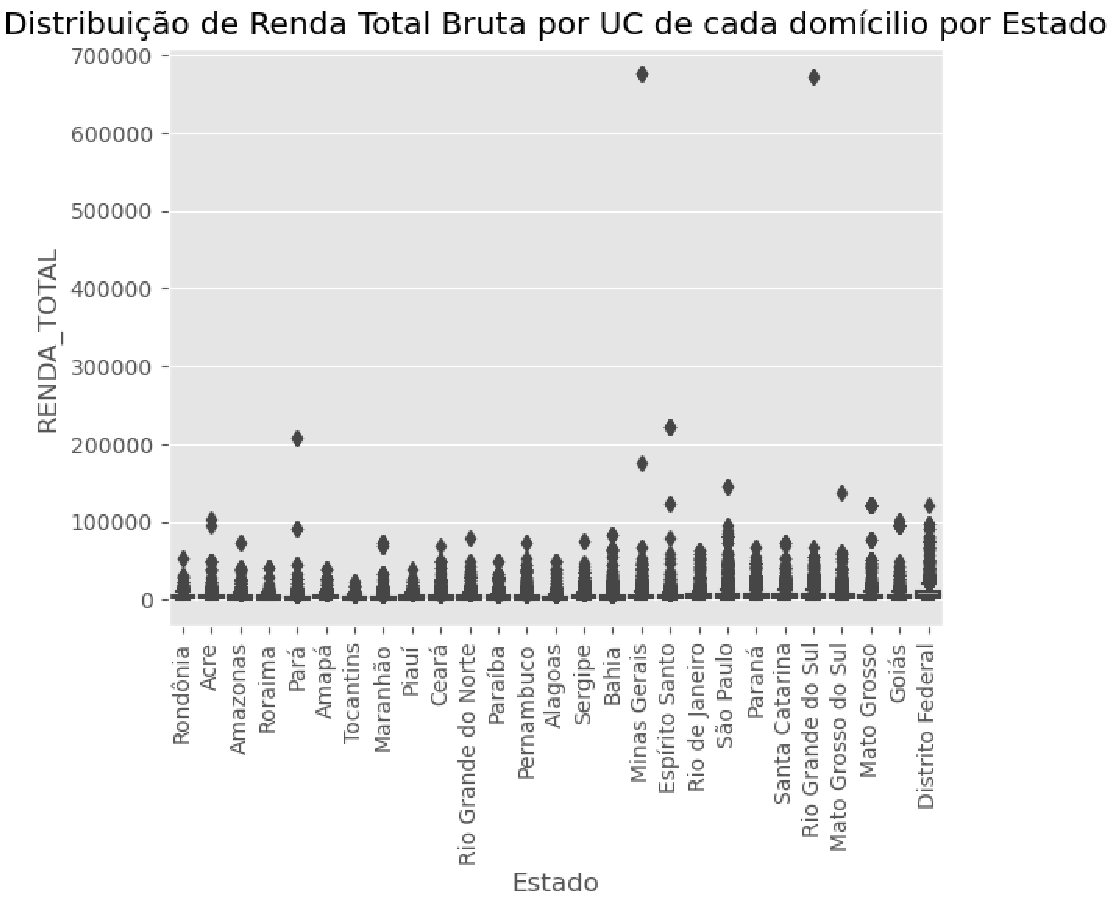
Out[101]:
```
UF
11     277.587183
12     146.911416
13     249.938078
14     162.072452
15    1054.944661
16     142.824272
17     377.991288
21     499.016657
22     443.355995
23     624.557688
24     423.601302
25     409.228804
26     537.887859
27     445.480885
28     371.318403
29     996.974743
31     794.174513
32     279.254165
33    1585.649699
35    2442.244042
41     895.013630
42    1124.035456
43     969.646222
50     304.768005
51     502.904170
52    1802.894449
53     476.571878
Name: PESO_FINAL, dtype: float64
```

Média de Renda Total por Consumo de Unidade por Domicílio por Estado POF4

In [152]:
```
bar(df_serviconaomonetpof4, df_serviconaomonetpof4['UF'], df_serviconaomone
```



In [153]:
```
sns.boxplot(x='UF', y='RENDA_TOTAL', data=df_serviconaomonetpof4)
plt.xlabel('Estado')
plt.ylabel('RENDA_TOTAL')
plt.title('Distribuição de Renda Total Bruta por UC de cada domícilio por E
plt.xticks(rotation = 90)
plt.show()
```



In [154]:
```
#Média
df_serviconaomonetpof4.groupby('UF') \
['RENDA_TOTAL'].mean()
```

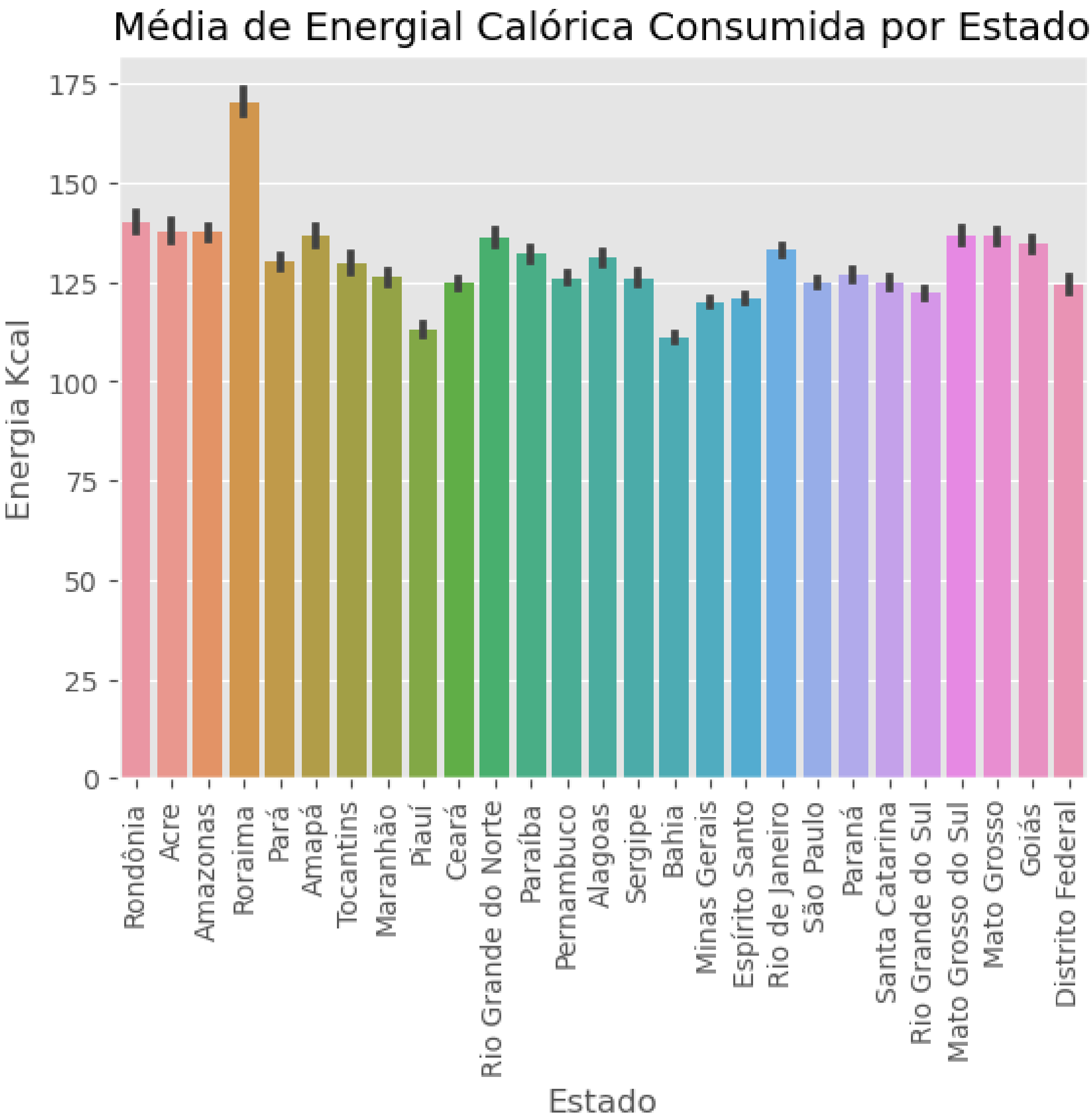
Out[154]:
```
UF
Acre                   4543.663735
Alagoas                3067.884317
Amapá                  4423.760893
Amazonas               4118.561630
Bahia                  3582.962223
Ceará                  3559.946010
Distrito Federal       9065.627968
Espírito Santo         4298.440624
Goiás                  4973.501340
Maranhão               3227.002773
Mato Grosso            5506.063696
Mato Grosso do Sul     5362.551349
Minas Gerais           4619.806945
Paraná                 5702.601244
Paraíba                3499.683579
Pará                   3296.754153
Pernambuco             3438.866010
Piauí                  3496.810610
```

In [155]:
```
#Desvio Padrão
df_serviconaomonetpof4.groupby('UF') \
['RENDA_TOTAL'].std()
```

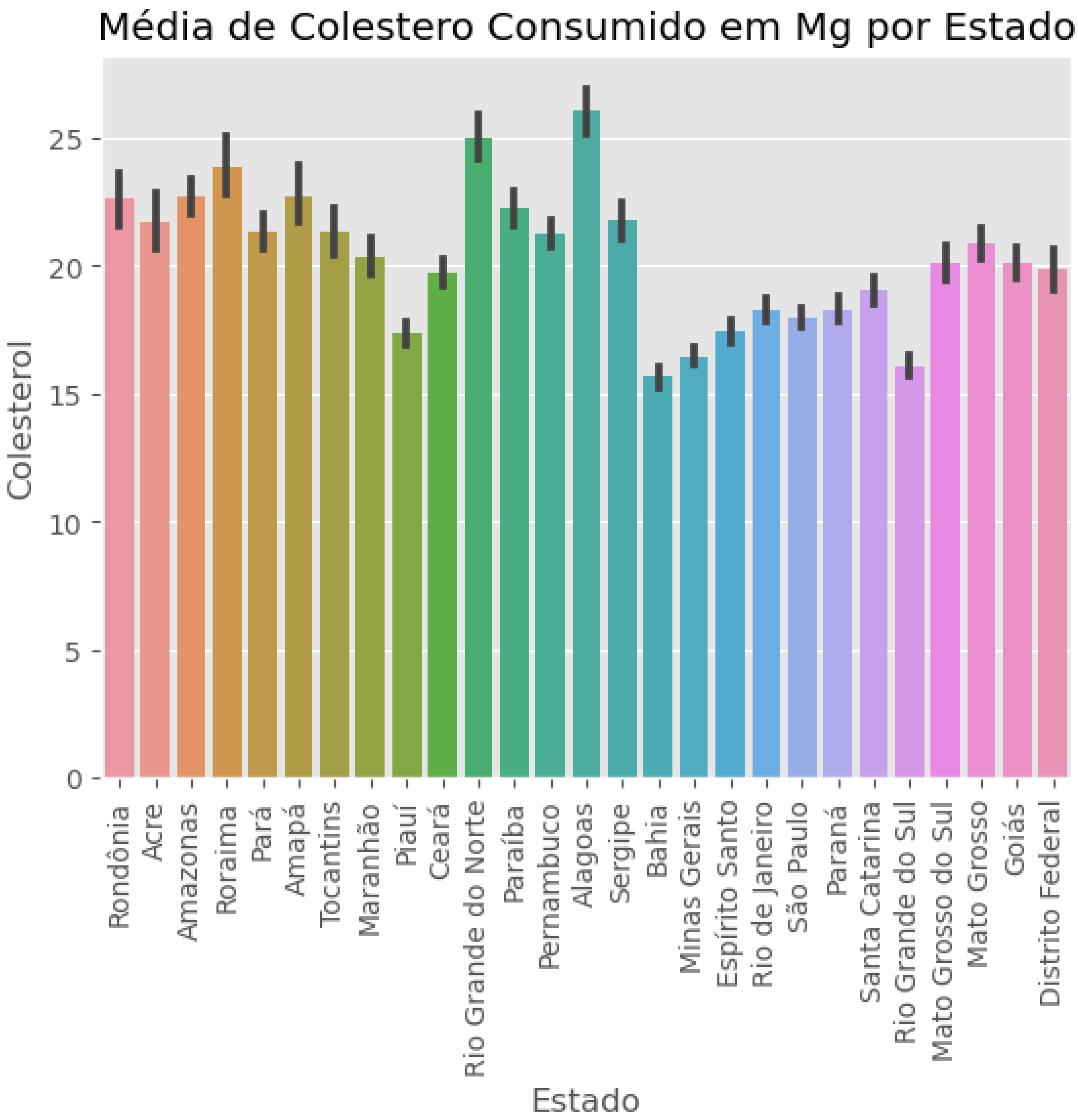
Out[155]:
```
UF
Acre                    5814.825845
Alagoas                 3923.360029
Amapá                   4486.596478
Amazonas                5007.096313
Bahia                   4713.058264
Ceará                   4345.919011
Distrito Federal       11899.264828
Espírito Santo          8288.720068
Goiás                   6665.187874
Maranhão                4021.840176
Mato Grosso             8817.167595
Mato Grosso do Sul      5302.399519
Minas Gerais           11216.508703
Paraná                  6266.379540
Paraíba                 4406.224818
Pará                    6885.932049
Pernambuco              3952.296053
Piauí                   2993.304826
```

Quantidade de Energia KCAL, COLESTEROL e Ferro por Estado na média de Dois dias em mg

In [157]:
```
bar(df_consumo_alimentar, 'UF', 'ENERGIA_KCAL', 'Média de Energial Calórica
```



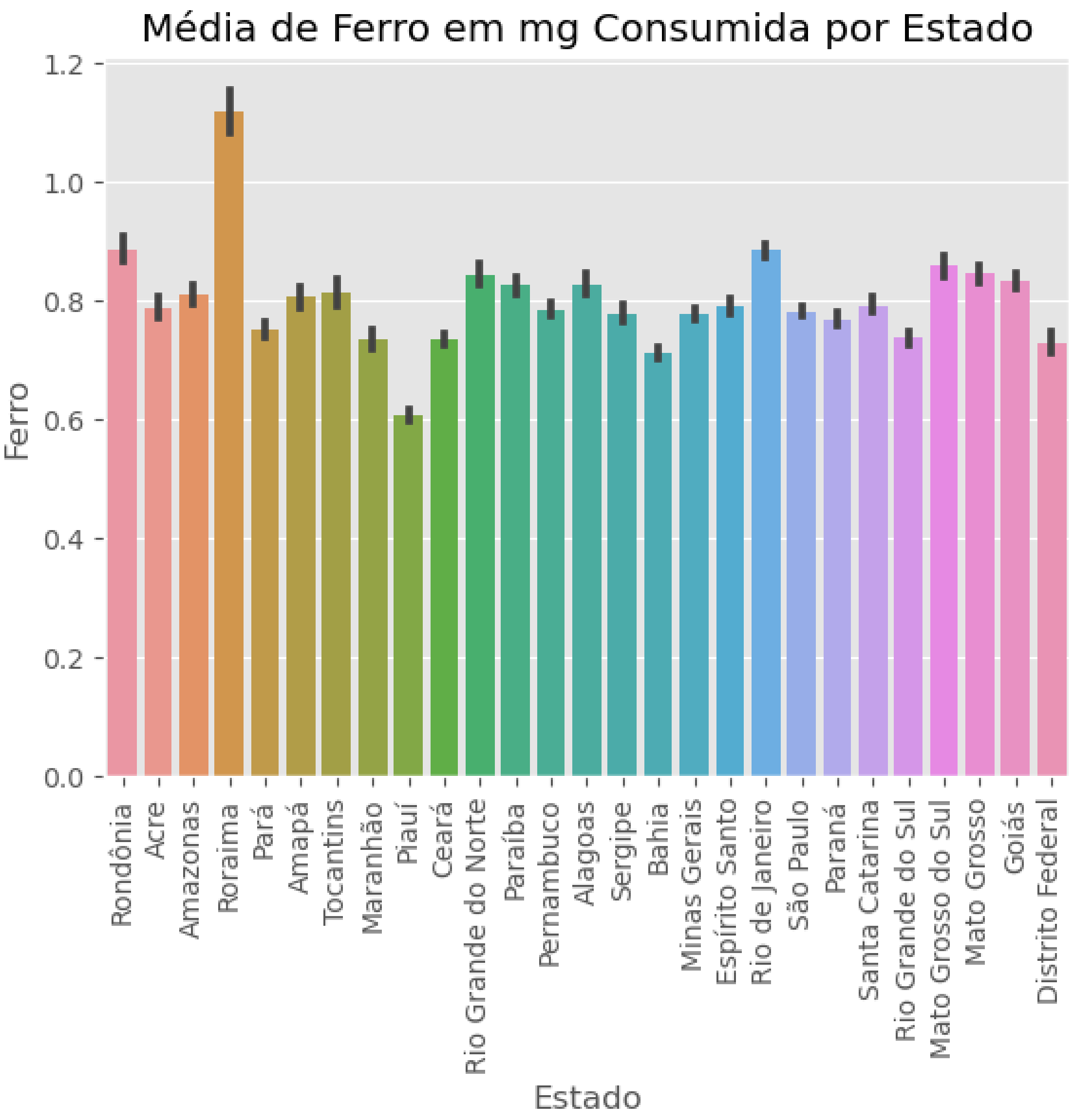
In [161]:
```
bar(df_consumo_alimentar, 'UF', 'COLEST', 'Média de Colestero Consumido em
```



In [119]:
```
#Média
df_consumo_alimentar.groupby('UF') \
['COLEST'].mean()
```

Out[119]:
```
UF
11    22.613816
12    21.738382
13    22.684715
14    23.879019
15    21.321146
16    22.724430
17    21.347946
21    20.339738
22    17.366335
23    19.741842
24    25.026682
25    22.255497
26    21.274022
27    26.059725
28    21.755748
29    15.654855
31    16.464932
32    17.443685
33    18.275220
35    17.963743
41    18.298464
42    19.022900
43    16.072541
50    20.087512
51    20.901314
52    20.102119
53    19.850851
Name: COLEST, dtype: float64
```

In [162]:
```
bar(df_consumo_alimentar, 'UF', 'FERRO', 'Média de Ferro em mg Consumida po
```



In [115]:
```
#Média
df_consumo_alimentar.groupby('UF') \
['FERRO'].mean()
```

Out[115]:
```
UF
11    0.884640
12    0.788406
13    0.809781
14    1.119062
15    0.750367
16    0.805458
17    0.812083
21    0.734996
22    0.608139
23    0.733845
24    0.843893
25    0.826198
26    0.785402
27    0.828108
28    0.778243
29    0.710449
31    0.777255
32    0.790828
33    0.884300
35    0.781986
41    0.768780
42    0.791921
43    0.736411
50    0.858456
51    0.845308
52    0.833080
53    0.729553
Name: FERRO, dtype: float64
```

Conclusões

- Minas, Bahia e São Paulo foram as regiões onde houve pesquisas. Roraima foi o estado com menos pesquisa. Os números de domicílio 13, 11, 10, 7, 12, 5, 8, 9, 1 apresentam quantidades semelhantes distribuídas na tabela POF2.
- O número total de unidades de consumo (UC) com uma quantidade desbalanceadamente maior do que as outras é 1 na tabela POF2.
- A relação entre PESO e PESO_FINAL é semelhante, mostrando estabilidade e estimativas populacionais bem aplicadas. Em relação à distribuição de domicílios, fica difícil de visualizar devido à grande quantidade, porém, a princípio, parece não ter nenhuma relação forte com o PESO.
- A relação entre PESO e Estado faz sentido e é compatível, já que São Paulo apresenta o maior Peso, sendo o estado mais populoso. Porém, é perceptível que existem muitos outliers entre os estados e um desvio padrão alto.
- O maior rendimento total bruto de consumo é o Distrito Federal. Minas Gerais apresenta o terceiro maior desvio padrão, e o Rio Grande do Sul, o segundo. Minas apresenta uma quantidade alta de outliers. O maior desvio padrão é no Distrito Federal, porém não supera a diferença de renda proporcionalmente quando comparado às outras duas regiões citadas anteriormente, não acaba sendo extremamente alta.

- No caso da pesquisa, os moradores de Roraima consumiram muito mais ferro do que as outras regiões. O menor consumo de colesterol por miligrama foi no Pará.