

```
In [ ]: import pandas as pd
import numpy as np
import glob
import matplotlib.pyplot as plt
import random
```

Os arquivos devem ser baixados da Justiça Eleitoral:

<https://dadosabertos.tse.jus.br/dataset/resultados-2022-arquivos-transmitidos-para-totalizacao>

Executando o script logjez_process.py arquivos CSV serão gerados para cada estado dois arquivos, um com votos e outro com dados das urnas.

Vamos carregar a tabela de cidades. Só será utilizada para informar o nome da cidade de acordo com o código

```
In [ ]: municipios = pd.read_csv("./municipios.csv")
municipios.sample(10)
```

```
Out[ ]:
```

	municipio_cod	municipio	estado
4695	83712	URUBICI	SC
5055	64955	IBIÃ NA	SP
4668	83194	SÃ O FRANCISCO DO SUL	SC
595	39675	WAGNER	BA
3205	74110	ALTO PIQUIRI	PR
2855	24074	FEIRA NOVA	PE
497	37958	PIRIPÃ	BA
4920	62340	NANTES	SP
2019	50890	RIACHO DOS MACHADOS	MG
4645	82732	PRESIDENTE CASTELLO BRANCO	SC

Vamos carregar os dados das urnas

```
In [ ]: path = "./csv_gerados/?.?.urnas.csv"
filenames = glob.glob(path)
dfs = []
for filename in filenames:
    dfs.append(pd.read_csv(filename, encoding = "ISO-8859-1"))

urnas = pd.concat(dfs, ignore_index=True).dropna()
```

```
In [ ]: urnas.sample(10)
```

Out[]:

	UF	municipio	zona	secao	qtdEleitoresAptos	qtdComparecimento	dataHoraAbertura	data
466966	SP	68152	171	20	299	250	20221030T080001	
398837	SP	61573	294	119	315	251	20221030T080001	
162609	MG	44458	103	192	321	262	20221030T080001	
142869	MG	50911	321	276	367	291	20221030T080001	
79418	DF	97012	5	32	308	271	20221030T080001	
362623	SE	31810	26	93	256	183	20221030T080001	
275076	RJ	60011	180	39	324	276	20221030T080001	
351897	SC	83674	33	127	279	229	20221030T080001	
311374	RN	17639	67	57	307	248	20221030T080001	
121592	MA	9113	60	65	240	155	20221030T080001	

In []:

urnas.shape[0]

Out[]:

471984

Aqui carregamos os arquivo com os votos.

In []:

path = "./csv_gerados/?.votos.csv"
filenames = glob.glob(path)
dfs = []
for filename in filenames:
 dfs.append(pd.read_csv(filename, encoding = "ISO-8859-1"))

votos = pd.concat(dfs, ignore_index=True).dropna()

In []:

votos.sample(10)

Out[]:

	UF	municipio	zona	secao	cargo	quantidadeVotos	partido	candidato
953408	PR	75639	147	197	presidente	161	22	22
1043045	PR	76678	146	233	presidente	184	22	22
1102239	RJ	60011	14	331	presidente	143	22	22
634861	MG	40673	15	3	presidente	5	nulo	nulo
1757189	SP	66010	304	169	presidente	3	branco	branco
524677	MG	41238	334	280	presidente	5	branco	branco
1445965	SP	62910	379	222	presidente	171	22	22
1222989	RO	663	25	60	presidente	5	nulo	nulo
929781	PI	10839	28	67	presidente	5	nulo	nulo
1570375	SP	71072	1	384	presidente	87	22	22

```
In [ ]: votos.shape[0]
```

```
Out[ ]: 1851420
```

Precisamos uma tabela com 2 colunas com totais de votos para os candidatos Bolsonaro e Lula

```
In [ ]: votosBolsonaro = votos.query("candidato == '22'")
votosLula = votos.query("candidato == '13'")
```

```
In [ ]: votosBolsonaro = votosBolsonaro.rename(columns={"quantidadeVotos": "Bolsonaro"})
votosLula = votosLula.rename(columns={"quantidadeVotos": "Lula"})
```

Aqui juntamos os dados com votos dos 2 candidatos e o modelo de urna

```
In [ ]: votosPresidente = votosBolsonaro[['UF', 'municipio', 'zona', 'secao', 'Bolsonaro']].join(
    votosLula[['UF', 'municipio', 'zona', 'secao', 'Lula']])
    .set_index(['UF', 'municipio', 'zona', 'secao'], how="outer",
    on=['UF', 'municipio', 'zona', 'secao']).join(
    urnas[['UF', 'municipio', 'zona', 'secao', 'modelo']])
    .set_index(['UF', 'municipio', 'zona', 'secao'],
    on=['UF', 'municipio', 'zona', 'secao'])
    )
```

```
In [ ]: votosPresidente.head(10)
```

```
Out[ ]:
```

	UF	municipio	zona	secao	Bolsonaro	Lula	modelo
1	AC	1015	2	88	84.0	53.0	UE2009
5	AC	1015	2	75	129.0	47.0	UE2009
7	AC	1015	2	69	125.0	53.0	UE2009
11	AC	1015	2	95	133.0	34.0	UE2009
14	AC	1015	2	90	144.0	47.0	UE2009
18	AC	1015	2	73	138.0	45.0	UE2009
22	AC	1015	2	74	129.0	48.0	UE2009
24	AC	1015	2	67	164.0	53.0	UE2009
28	AC	1015	2	80	124.0	98.0	UE2009
32	AC	1015	2	66	162.0	56.0	UE2009

```
In [ ]: votosPresidente.shape
```

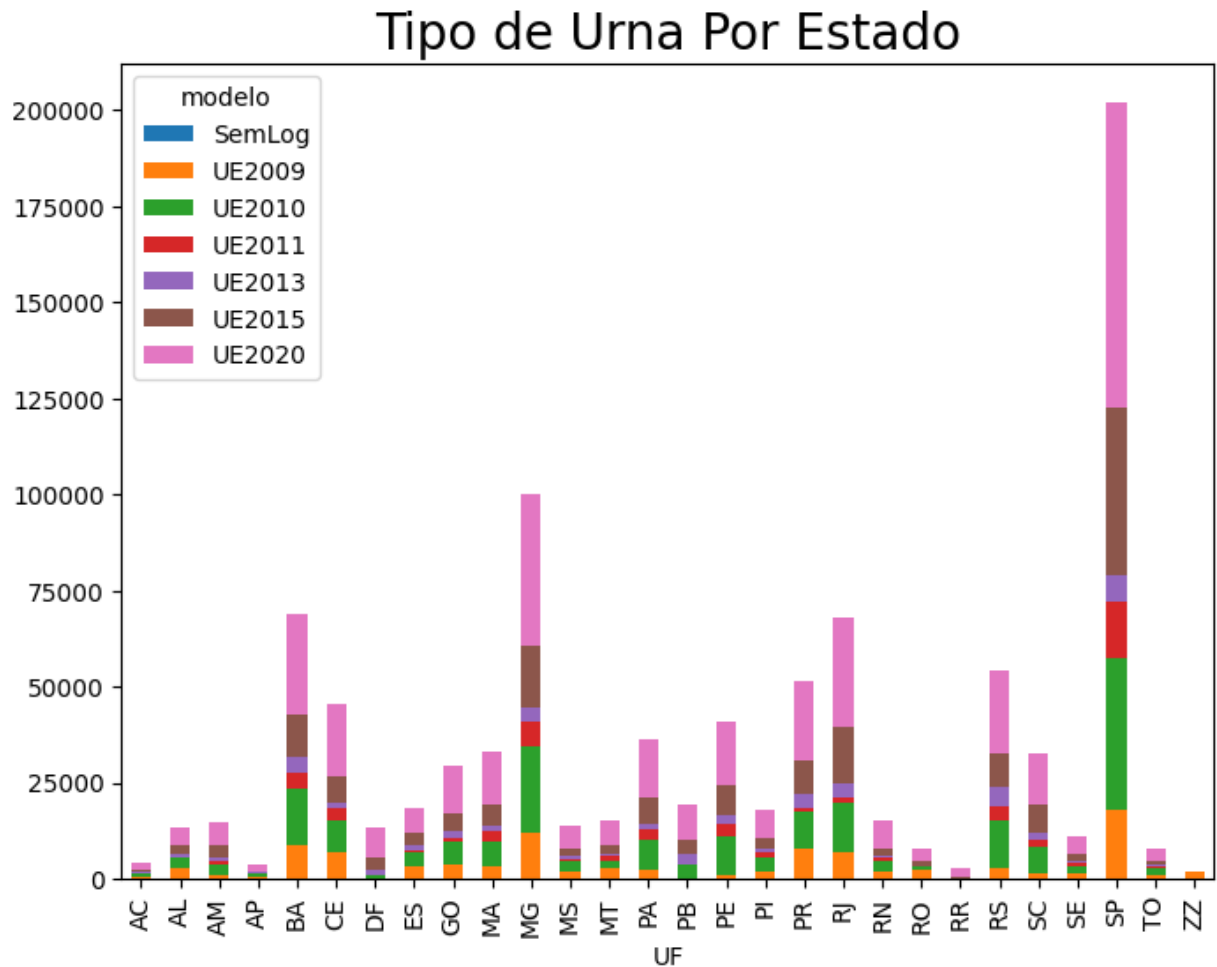
```
Out[ ]: (471525, 7)
```

O foco aqui é validar a suspeita de 1 modelo das urnas ter comportamento na totalização de votos, portanto devemos começar analisando como foram distribuídas as urnas, pelo menos por estado, para diminuir a probabilidade de o fator demográfico determinar a diferença.

```
In [ ]: urnasPorUF = pd.pivot_table(data=votosPresidente[['UF', 'modelo']], index=['UF'], columns=['modelo'])
```

```
ax = urnasPorUF.plot.bar(stacked=True, figsize=(8,6))
ax.set_title('Tipo de Urna Por Estado', fontsize=20)
```

```
Out[ ]: Text(0.5, 1.0, 'Tipo de Urna Por Estado')
```



Acima vemos a distribuição. Notem que haviam urnas que não encontramos o arquivo de log (SemLog).

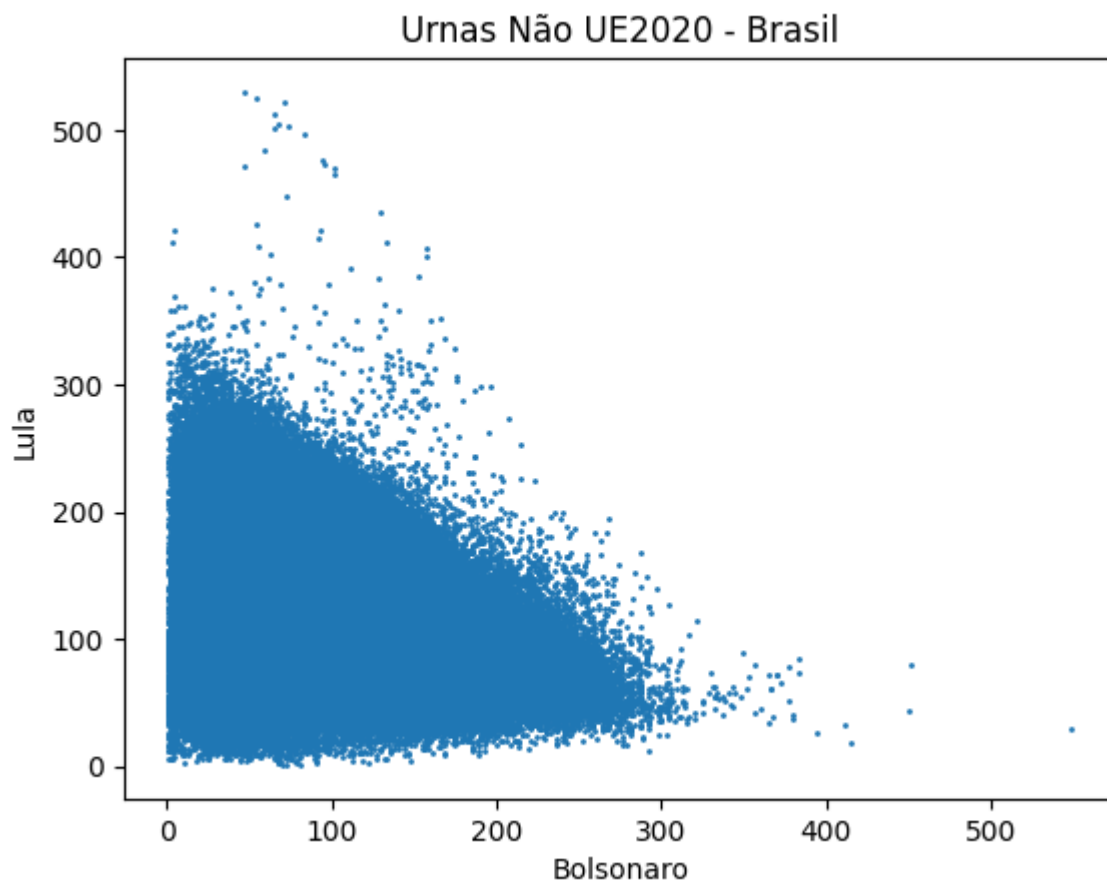
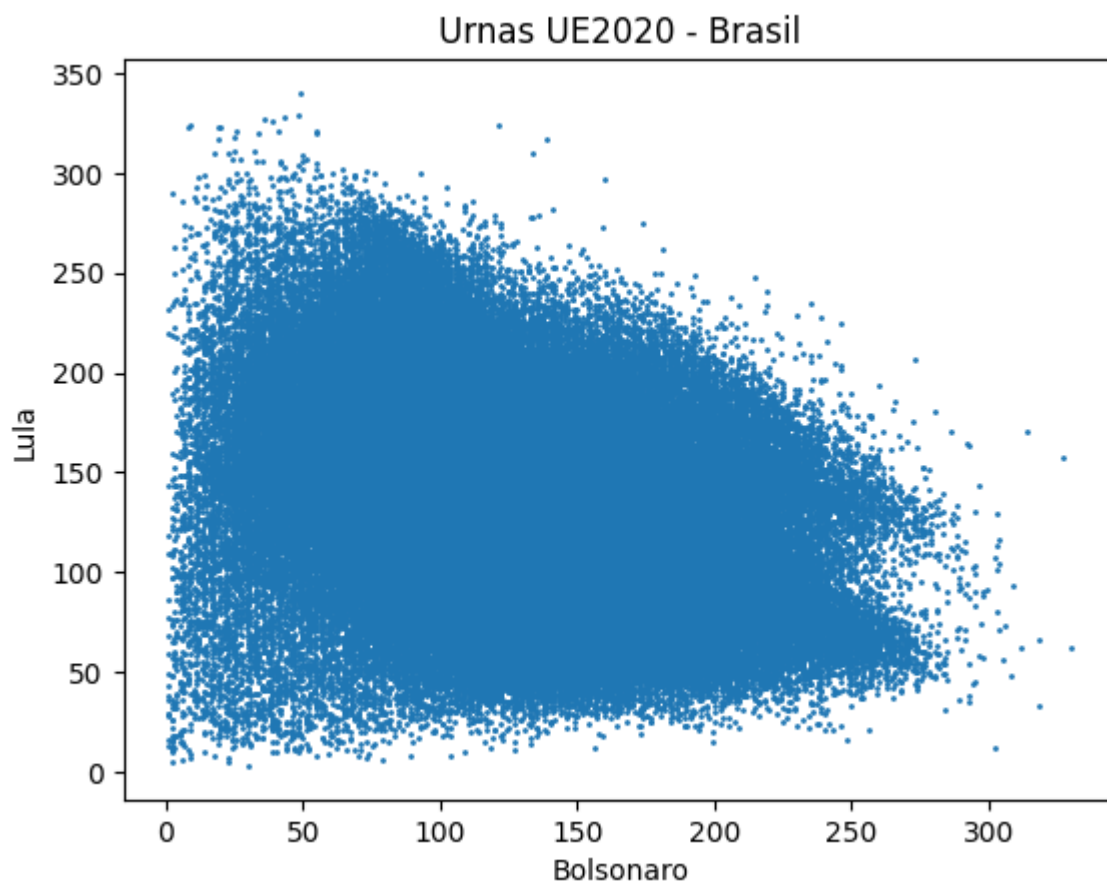
O gráfico mostra que as urnas foram distribuídas em todos os estados. Em especial o modelo mais novo que representa uma boa quantidade das urnas para cada estado (em rosa)

Os gráficos abaixo mostram a distribuição de votos apurados nas urnas UE2020 e o contraste para as não UE2020 (anteriores a 2020 que não foram auditadas no pleito de 2022)

```
In [ ]: data = votosPresidente.query("modelo == 'UE2020'")
plt.scatter(data['Bolsonaro'], data['Lula'], s=1)
plt.title("Urnas UE2020 - Brasil")
plt.xlabel("Bolsonaro")
plt.ylabel("Lula")
plt.show()

data = votosPresidente.query("modelo != 'UE2020'")
plt.scatter(data['Bolsonaro'], data['Lula'], s=1)
plt.title("Urnas Não UE2020 - Brasil")
plt.xlabel("Bolsonaro")
```

```
plt.ylabel("Lula")  
plt.show()
```



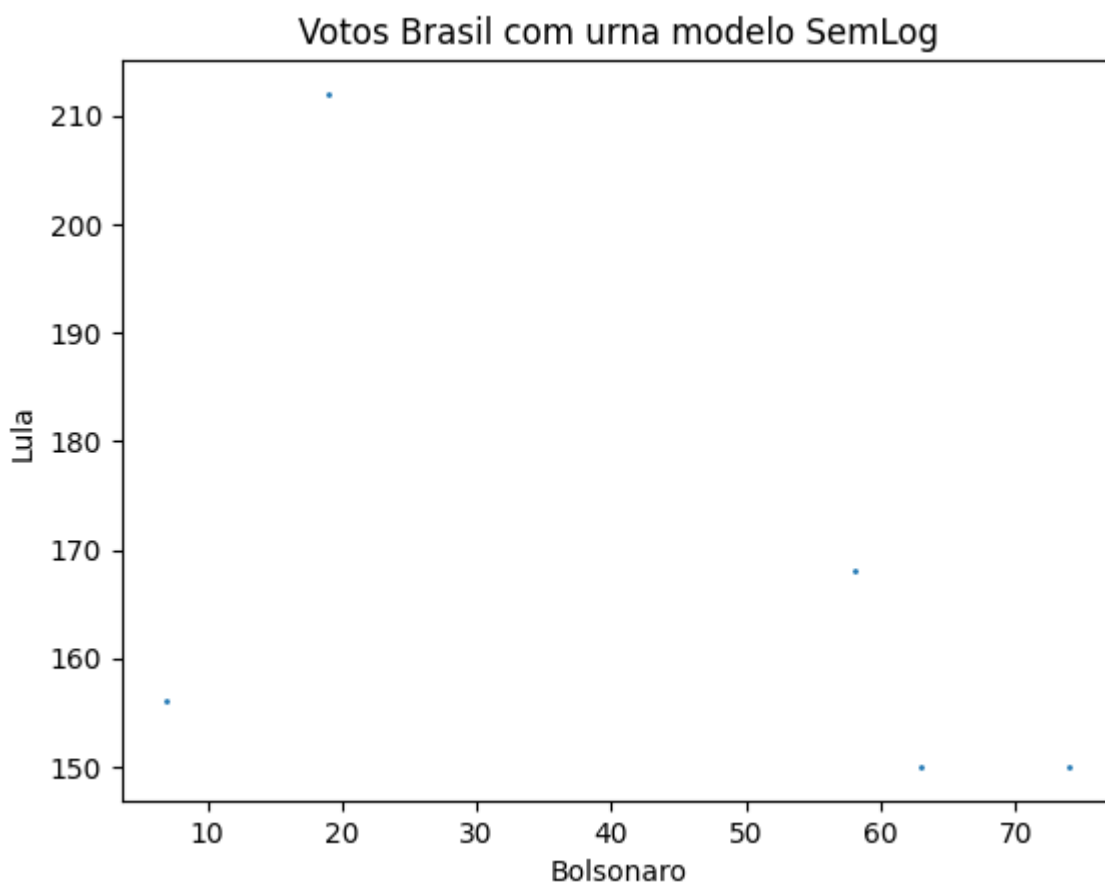
Os gráficos já mostram acima que o comportamento é muito distinto. As urnas mais antigas deram mais vantagem ao candidato Lula, aparentemente retirando votos do Bolsonaro.

No gráfico das 2020 aparece um grande numero de votos mais ao centro, enquanto as antigas parecem ter algum tipo de trava, o que gera uma linha geometrica, quase igual um triângulo escaleno.

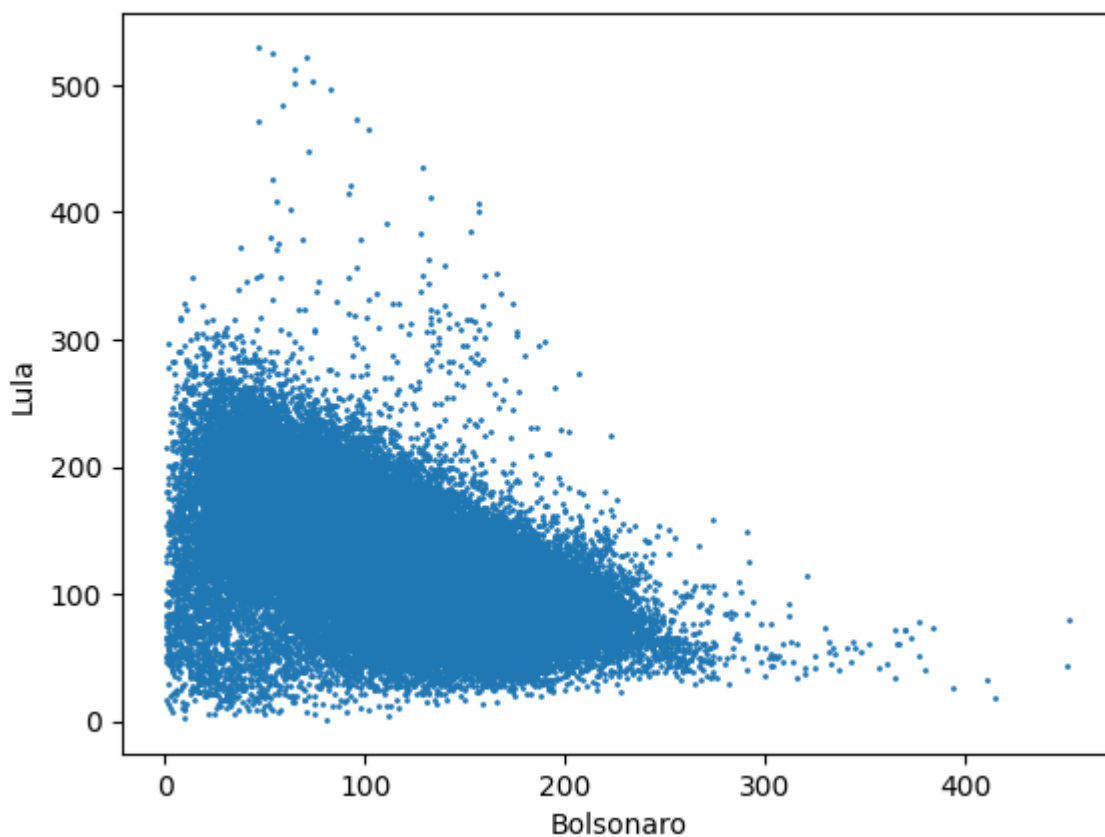
Vamos ver os mesmos gráficos para cada tipo de urna.

** note que existem 3 populações maiores de urnas, as UE2010, UE2015 e UE2020. Os demais modelos são minoria, o que pode dificultar essa análise de um ponto de vista geometrico*

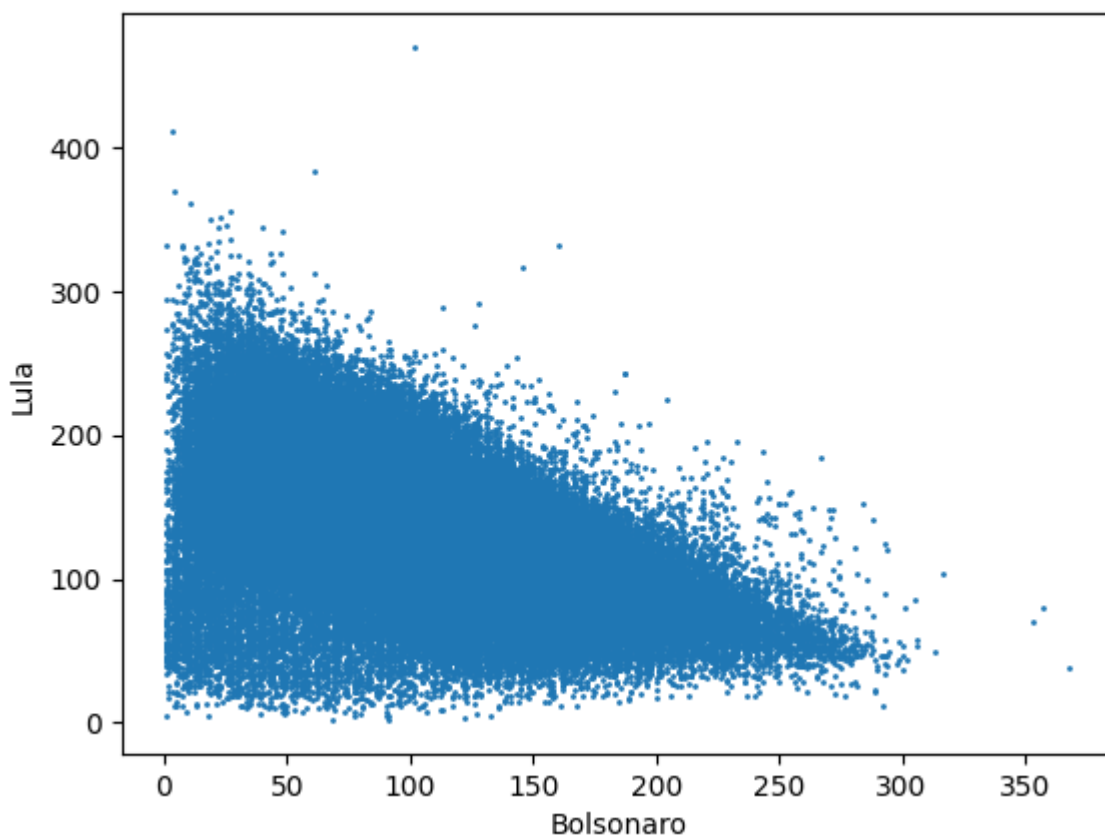
```
In [ ]: for index, row in votosPresidente.filter(['modelo'], axis=1).drop_duplicates().sort_va
        data = votosPresidente.query("modelo == '"+row['modelo']+"'")
        plt.scatter(data['Bolsonaro'], data['Lula'], s=1)
        plt.title("Votos Brasil com urna modelo "+row["modelo"])
        plt.xlabel("Bolsonaro")
        plt.ylabel("Lula")
        plt.show()
```



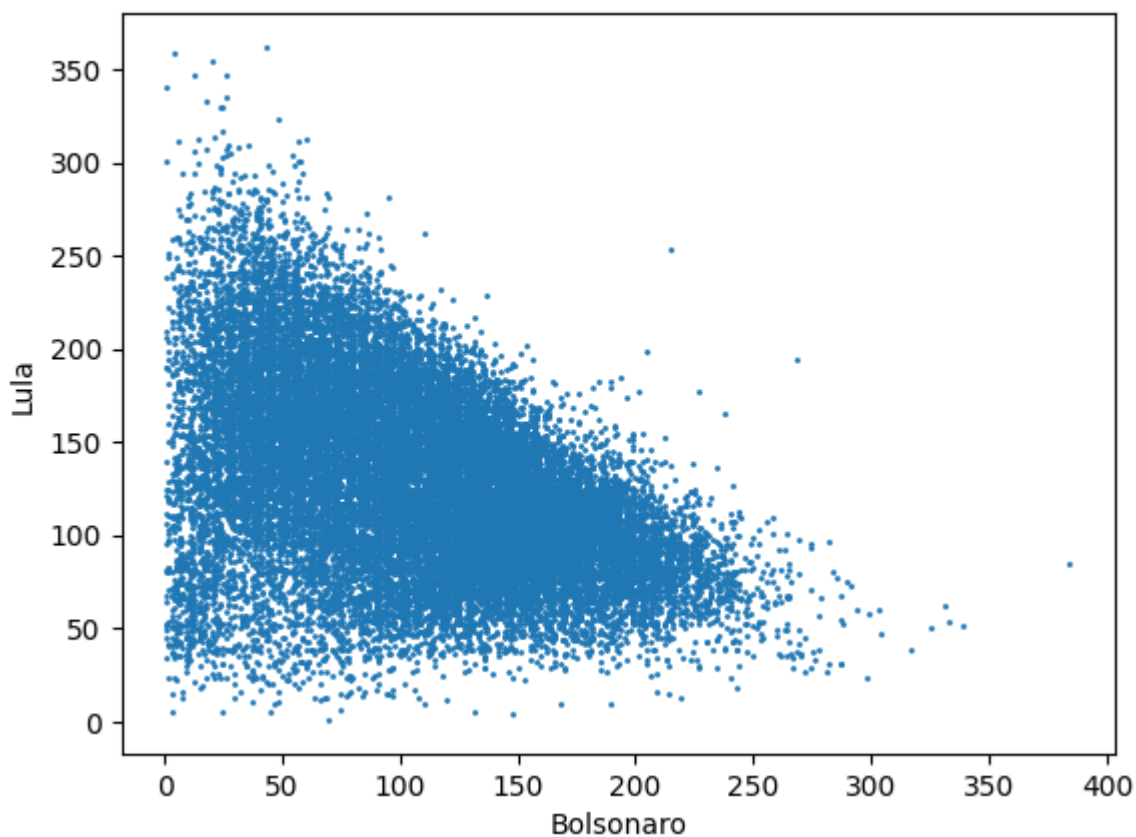
Votos Brasil com urna modelo UE2009



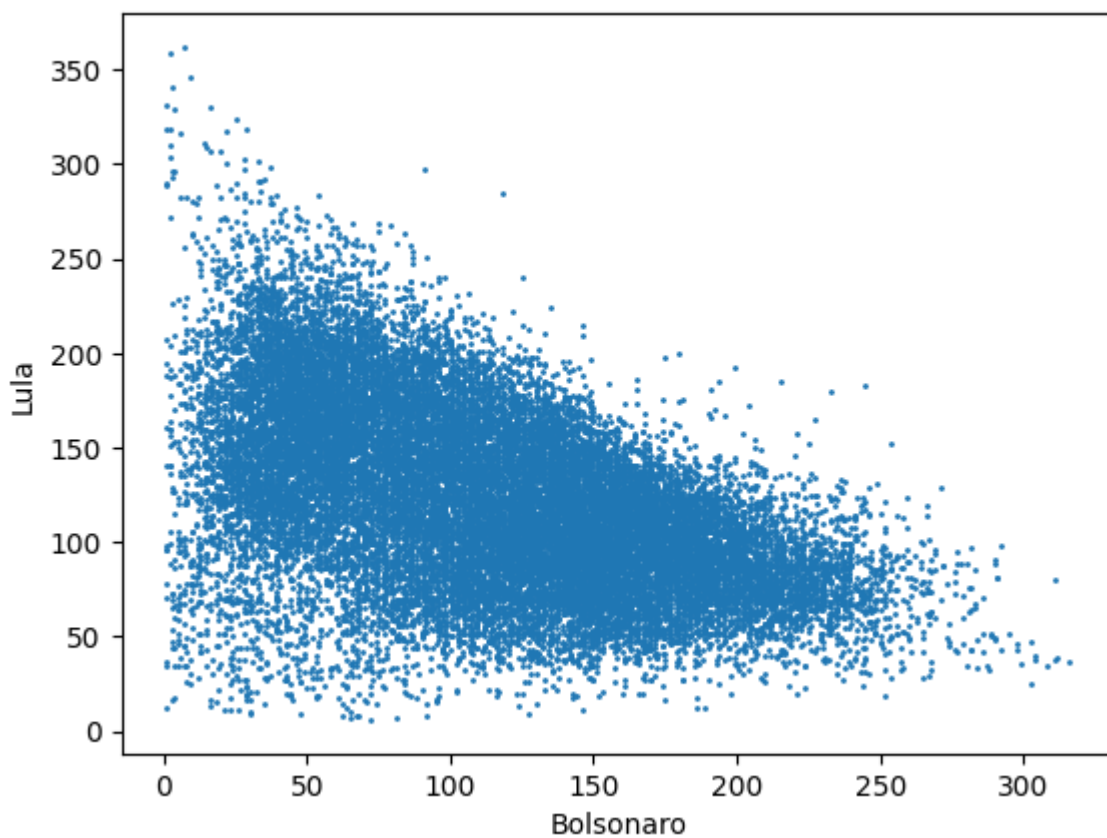
Votos Brasil com urna modelo UE2010

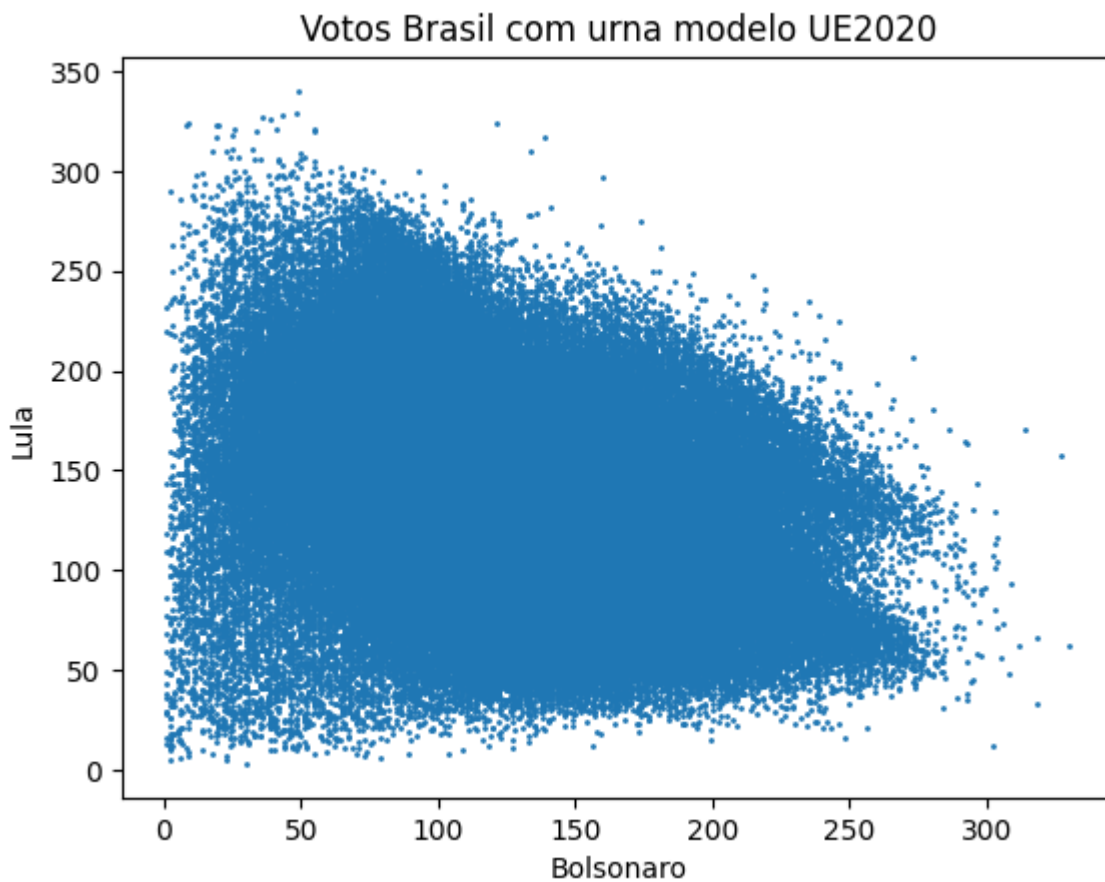
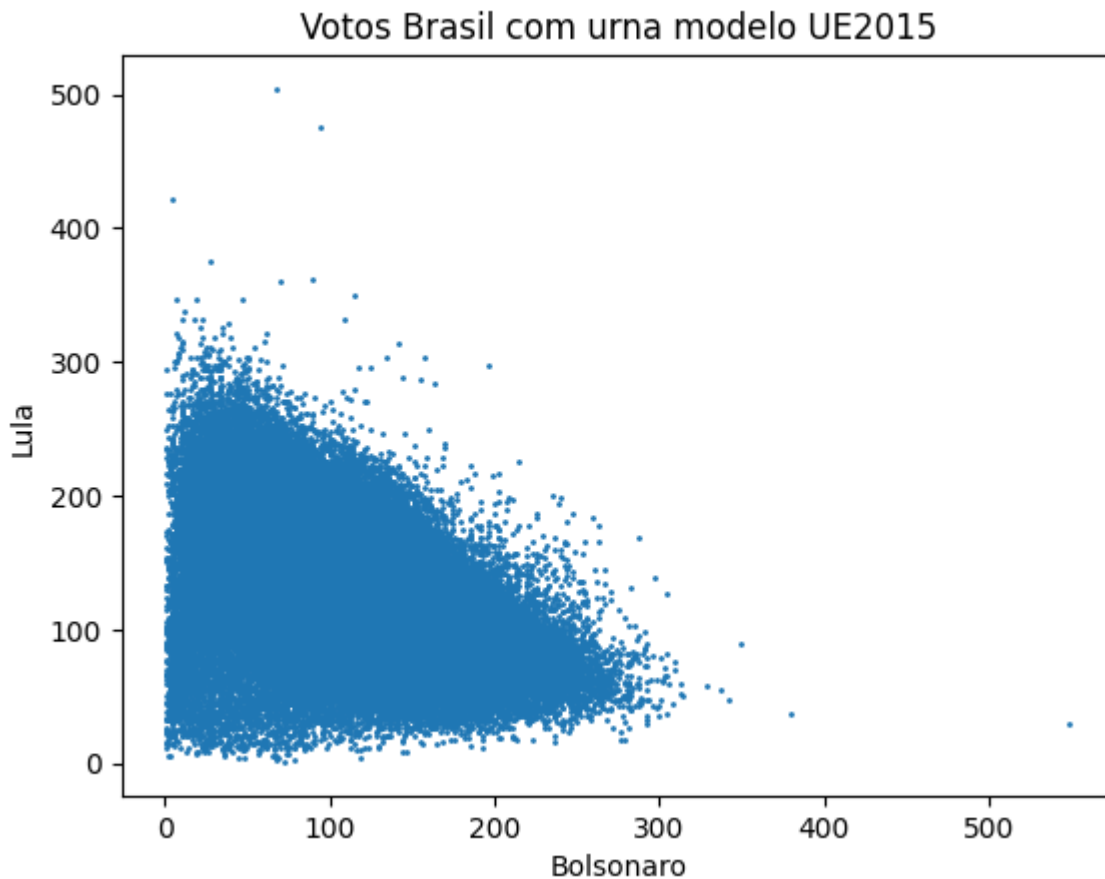


Votos Brasil com urna modelo UE2011



Votos Brasil com urna modelo UE2013





Revise os gráficos acima e note como é possível notar o tal triângulo escaleno em todos modelos, menos no modelo UE2020 onde, apesar de visualizarmos o triângulo, há um escape

(massa da votos) mais distribuida ao centro.

Isso pode indicar algumas possiveis causas

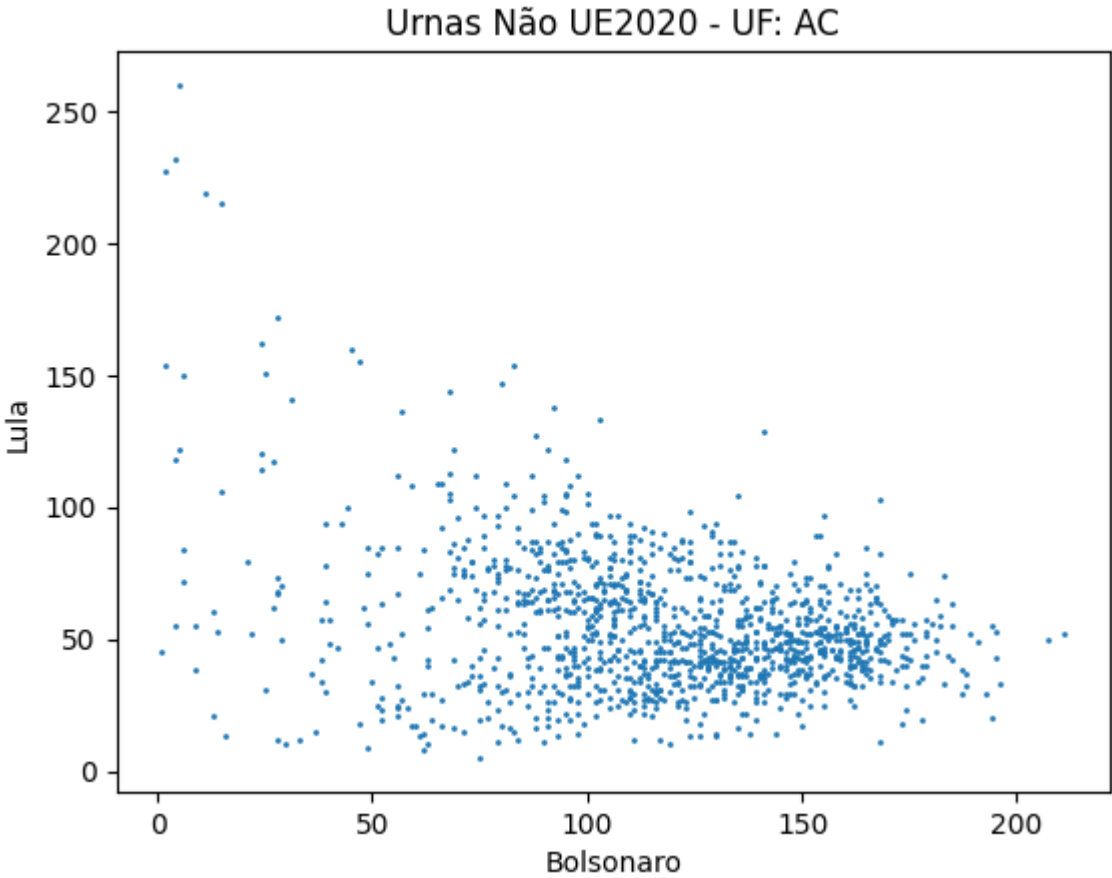
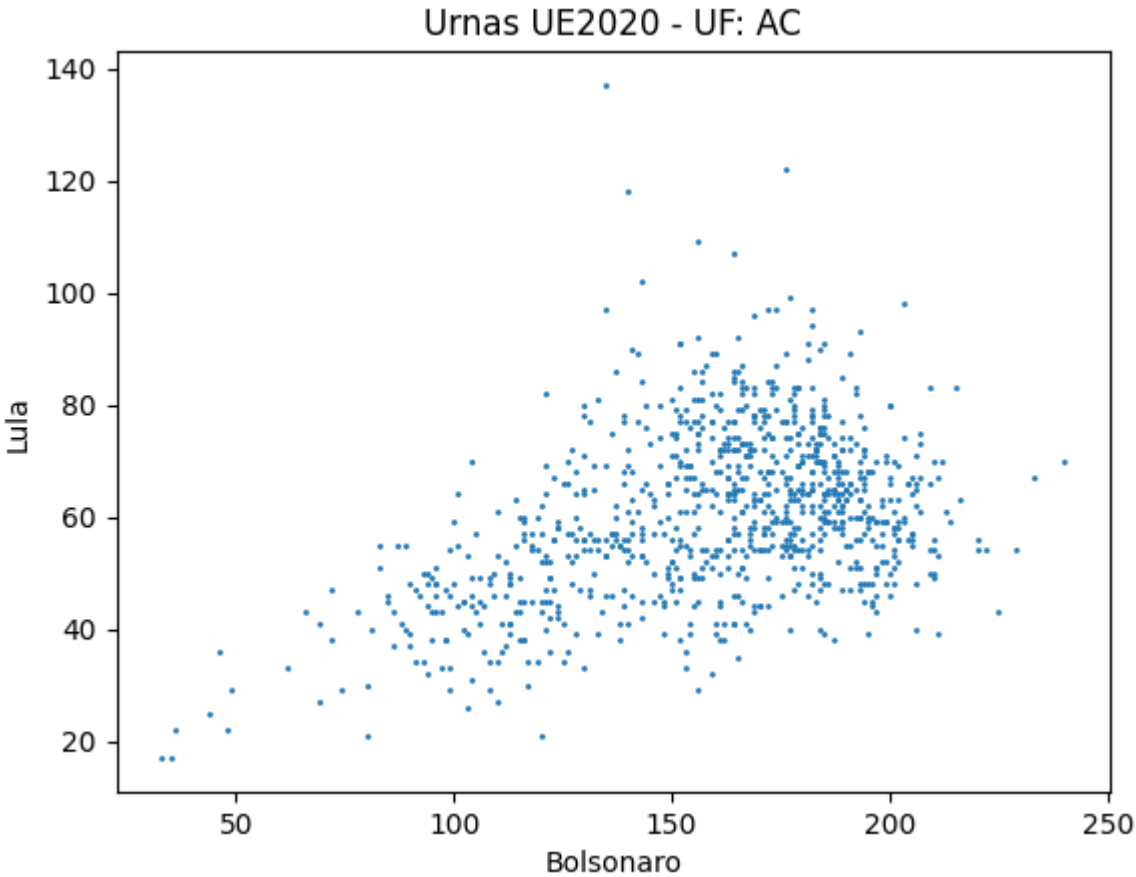
- Algum algoritmo foi criado para travar os votos, migrando de outro candidato para o candidato 13
- Seria improvável os mesmos eleitores, nos mesmos locais, terem comportamento tão diferenciado
- As urnas tem diferença no software, e o software foi adulterado

Aqui vamos explorar as urnas UE2020 e não UE2020 por cada estado

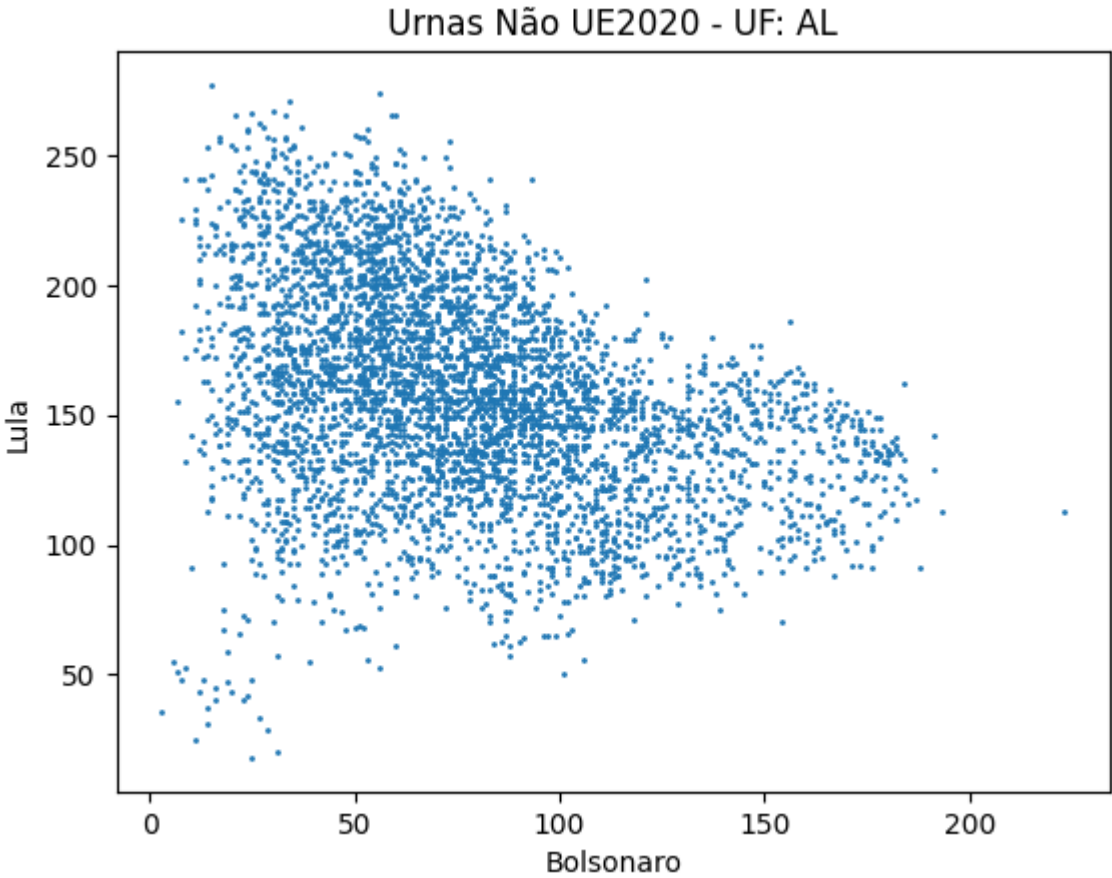
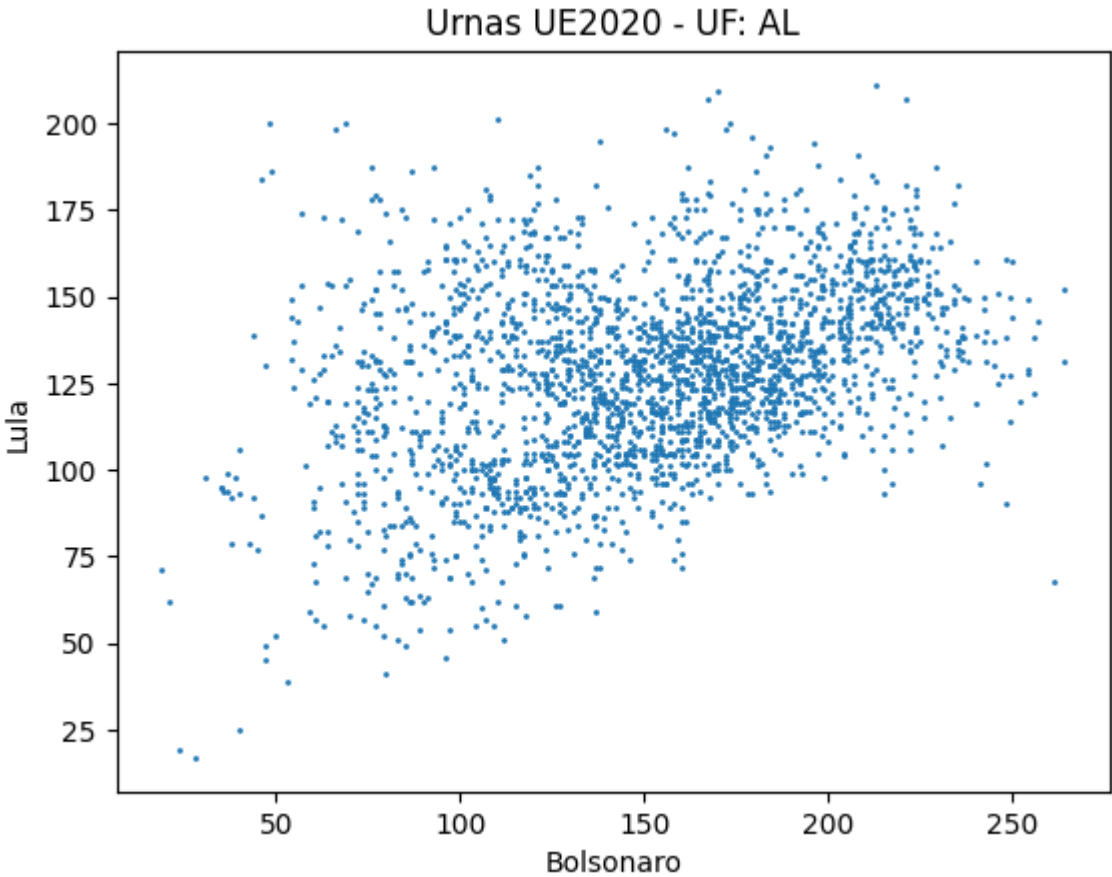
```
In [ ]: for index, row in votosPresidente.filter(['UF'], axis=1).drop_duplicates().iterrows():
        print(row['UF'])
        data = votosPresidente.query("modelo == 'UE2020' and UF == '"+row['UF']+"'")
        plt.scatter(data['Bolsonaro'], data['Lula'], s=1)
        plt.title("Urnas UE2020 - UF: "+row['UF'])
        plt.xlabel("Bolsonaro")
        plt.ylabel("Lula")
        plt.show()

        data = votosPresidente.query("modelo != 'UE2020' and UF == '"+row['UF']+"'")
        plt.scatter(data['Bolsonaro'], data['Lula'], s=1)
        plt.title("Urnas Não UE2020 - UF: "+row['UF'])
        plt.xlabel("Bolsonaro")
        plt.ylabel("Lula")
        plt.show()
```

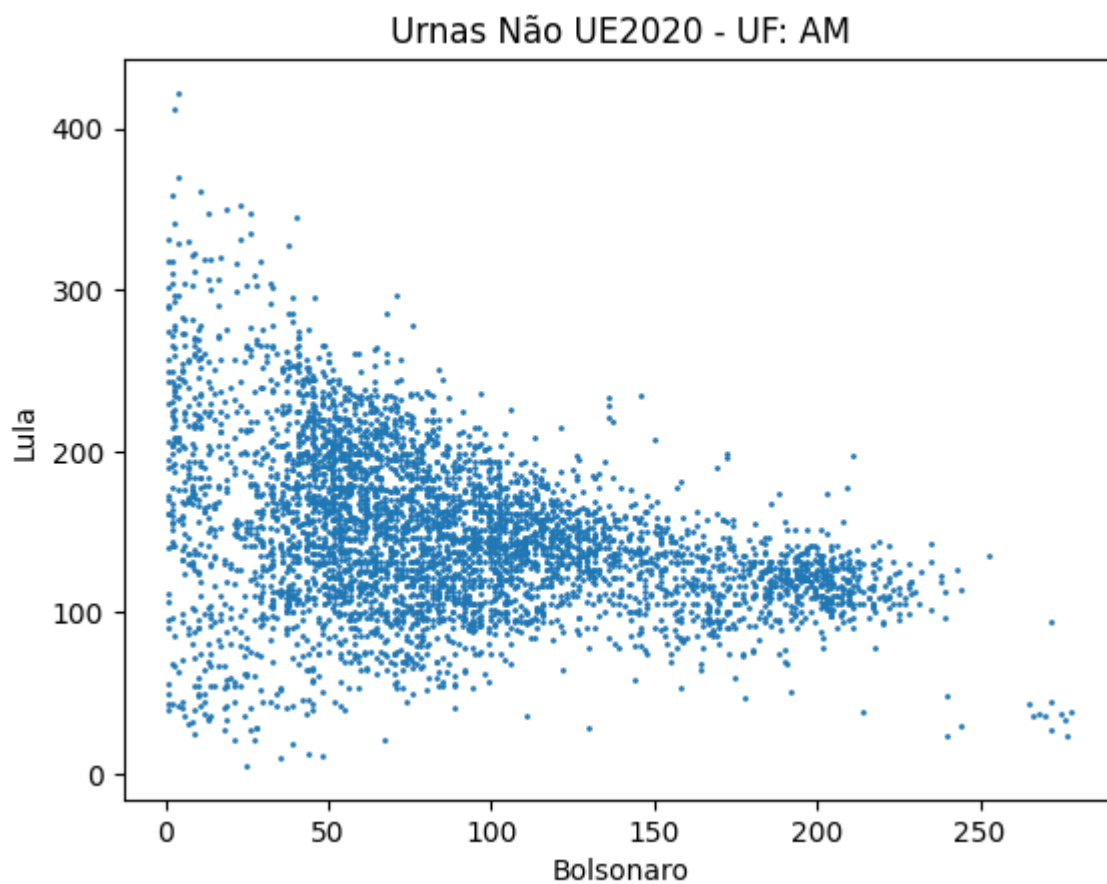
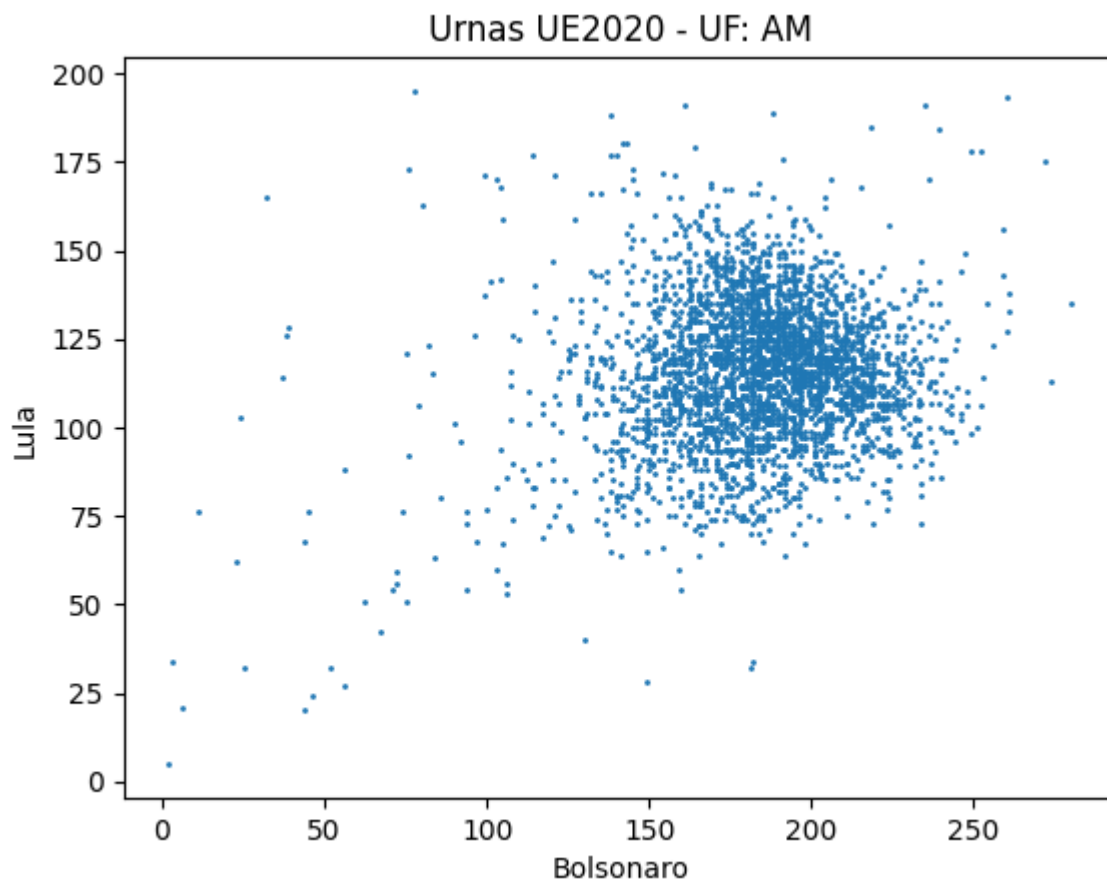
AC



AL

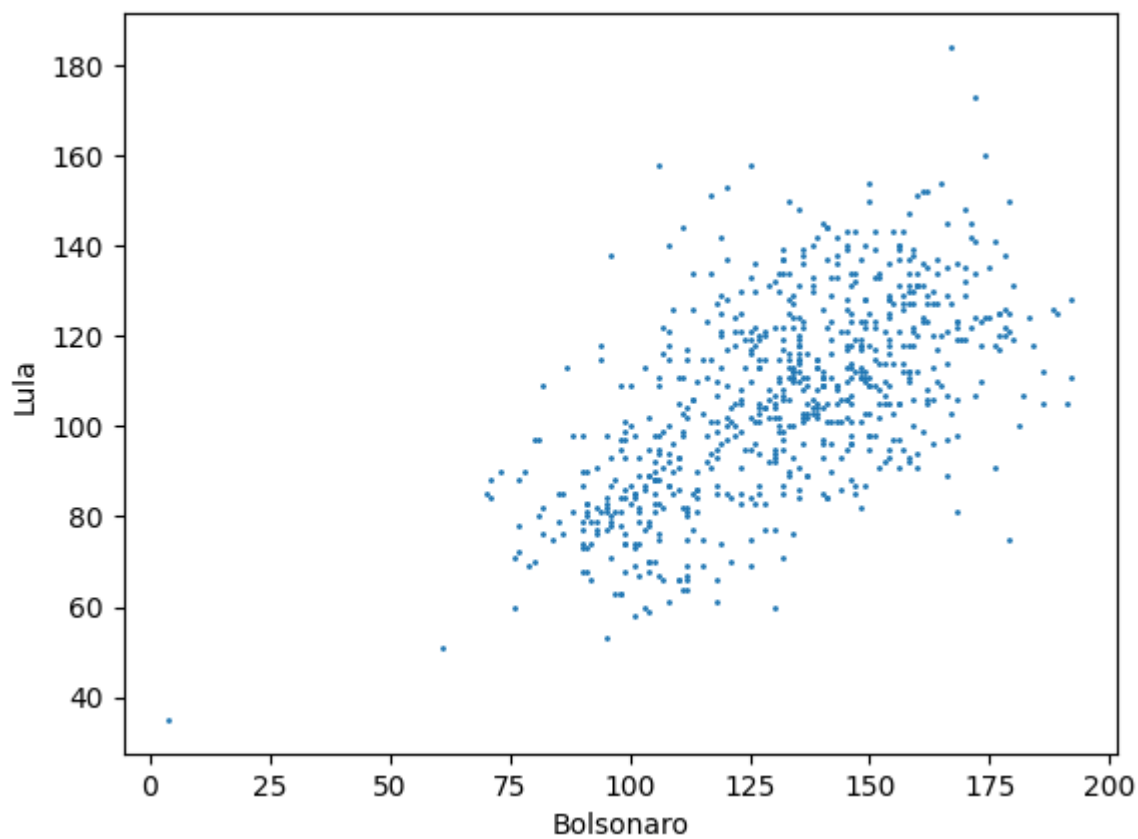


AM

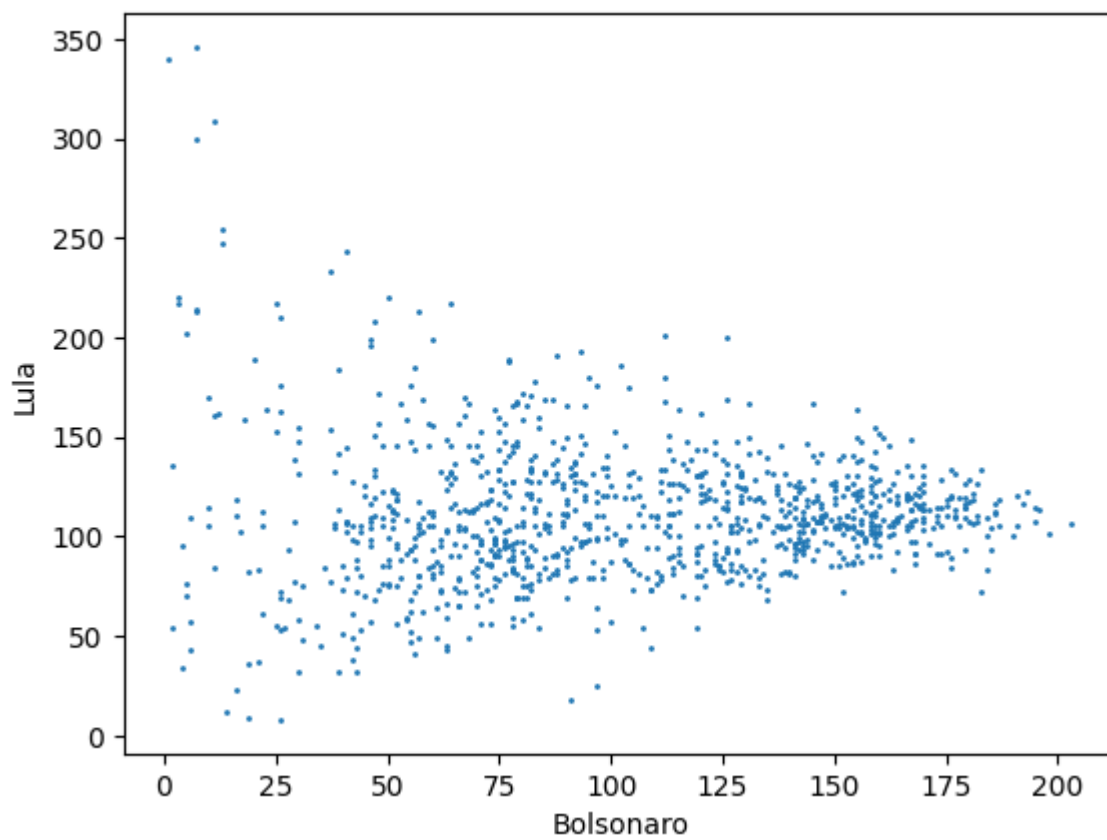


AP

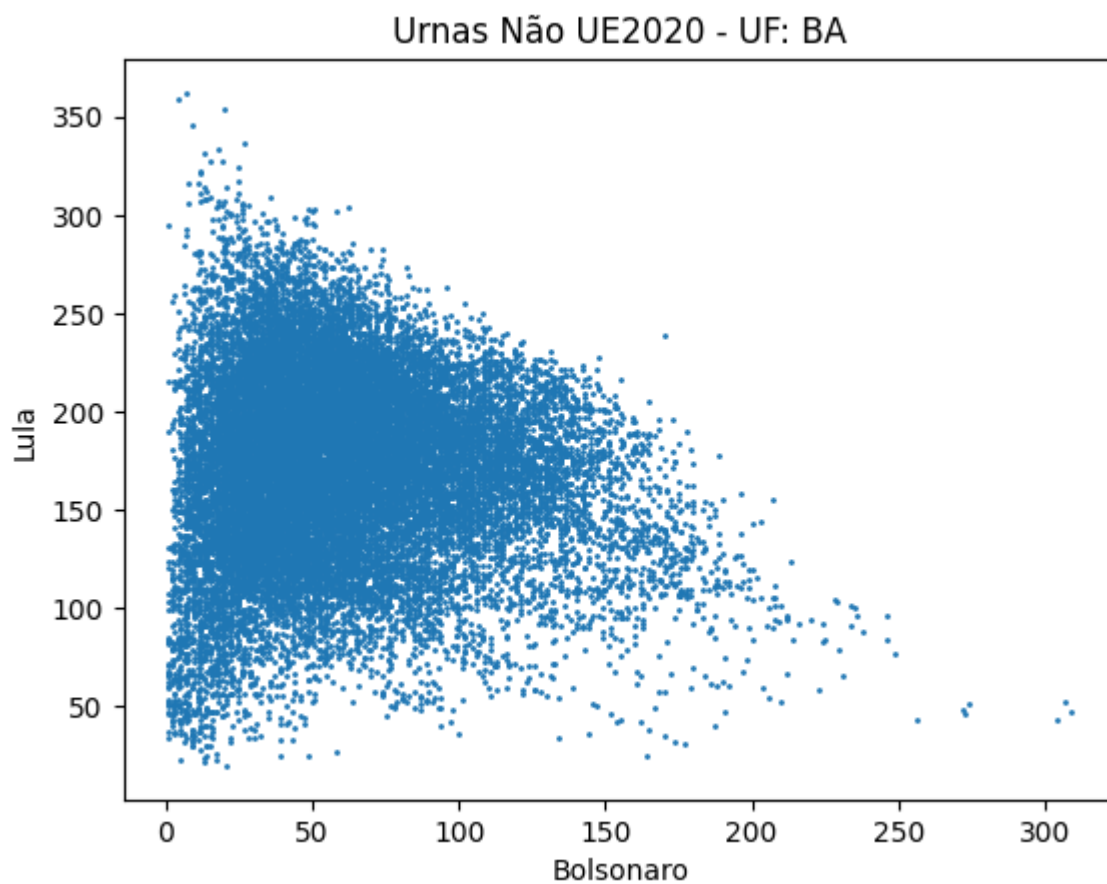
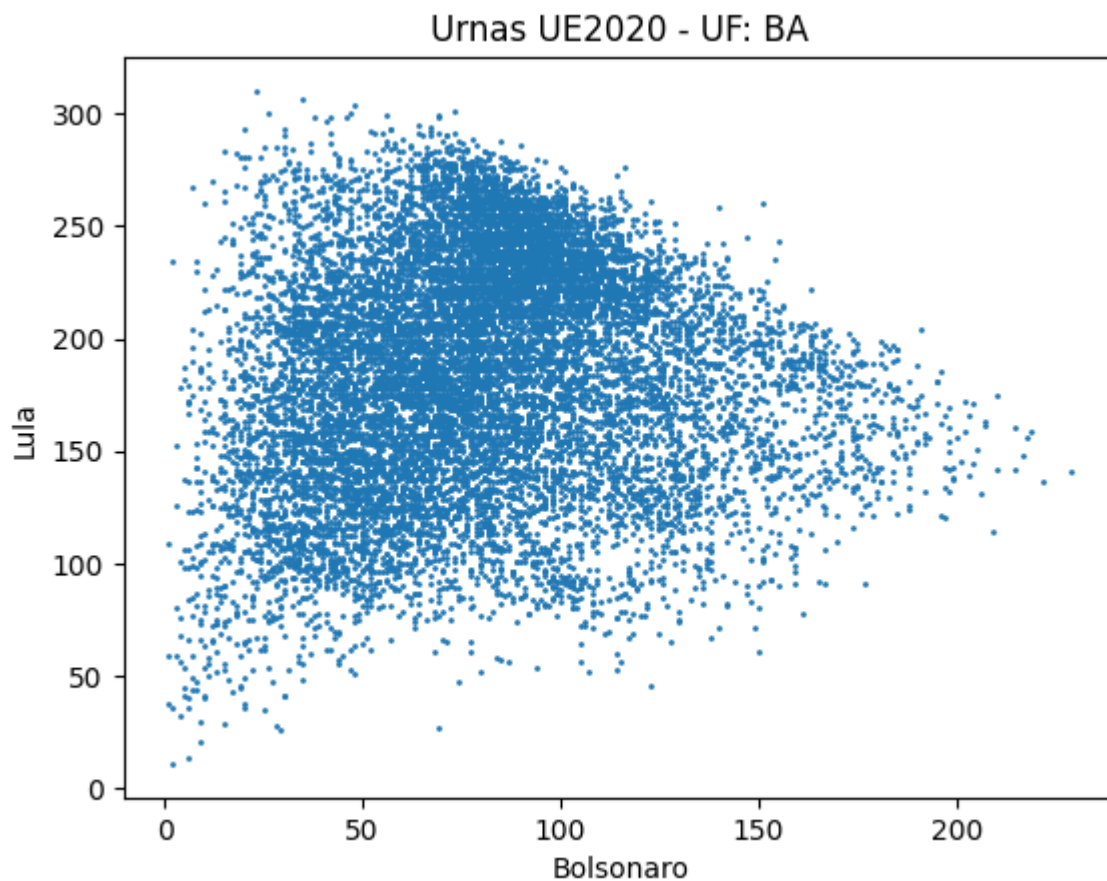
Urnas UE2020 - UF: AP



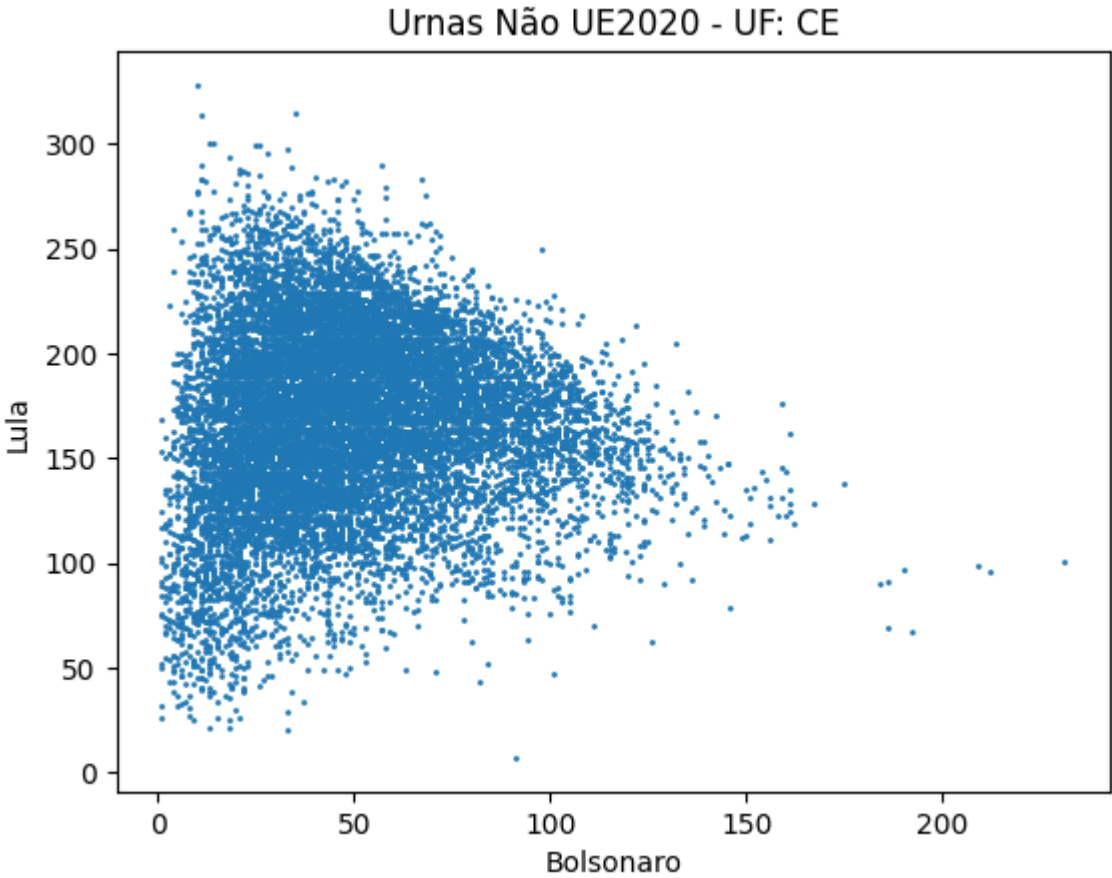
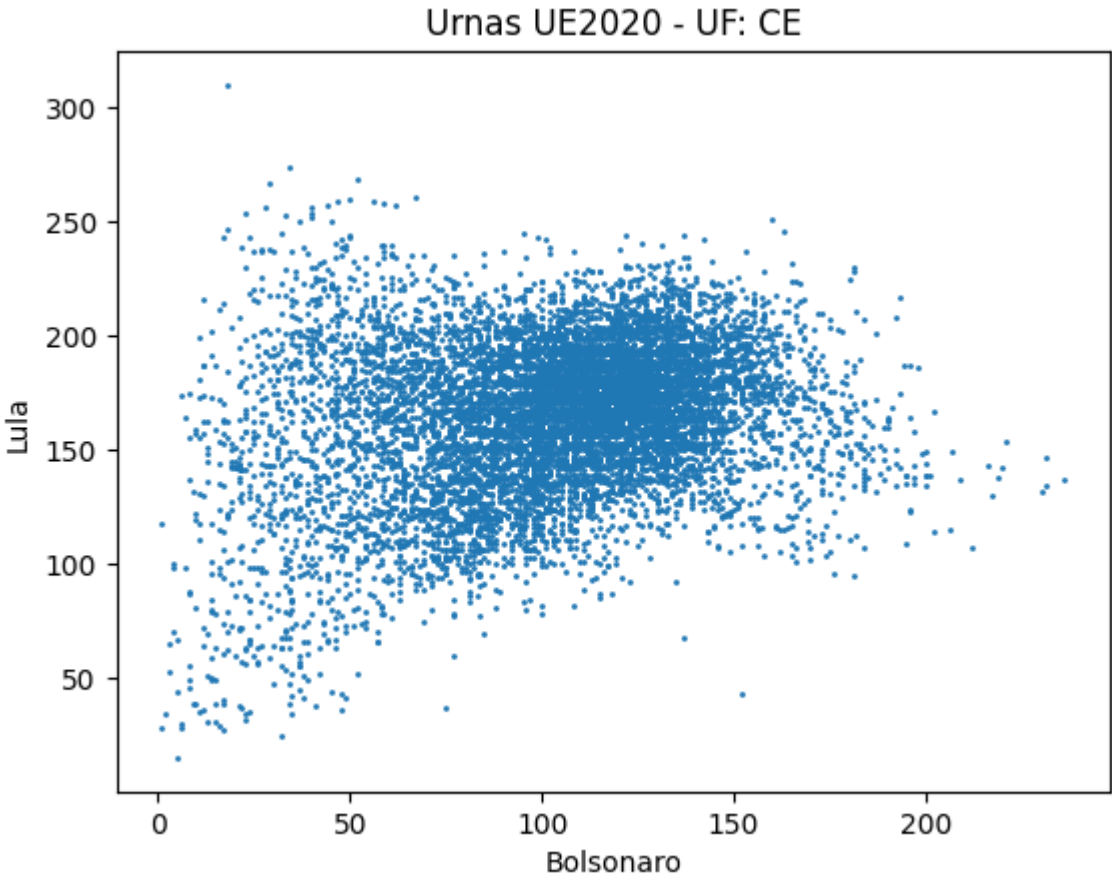
Urnas Não UE2020 - UF: AP



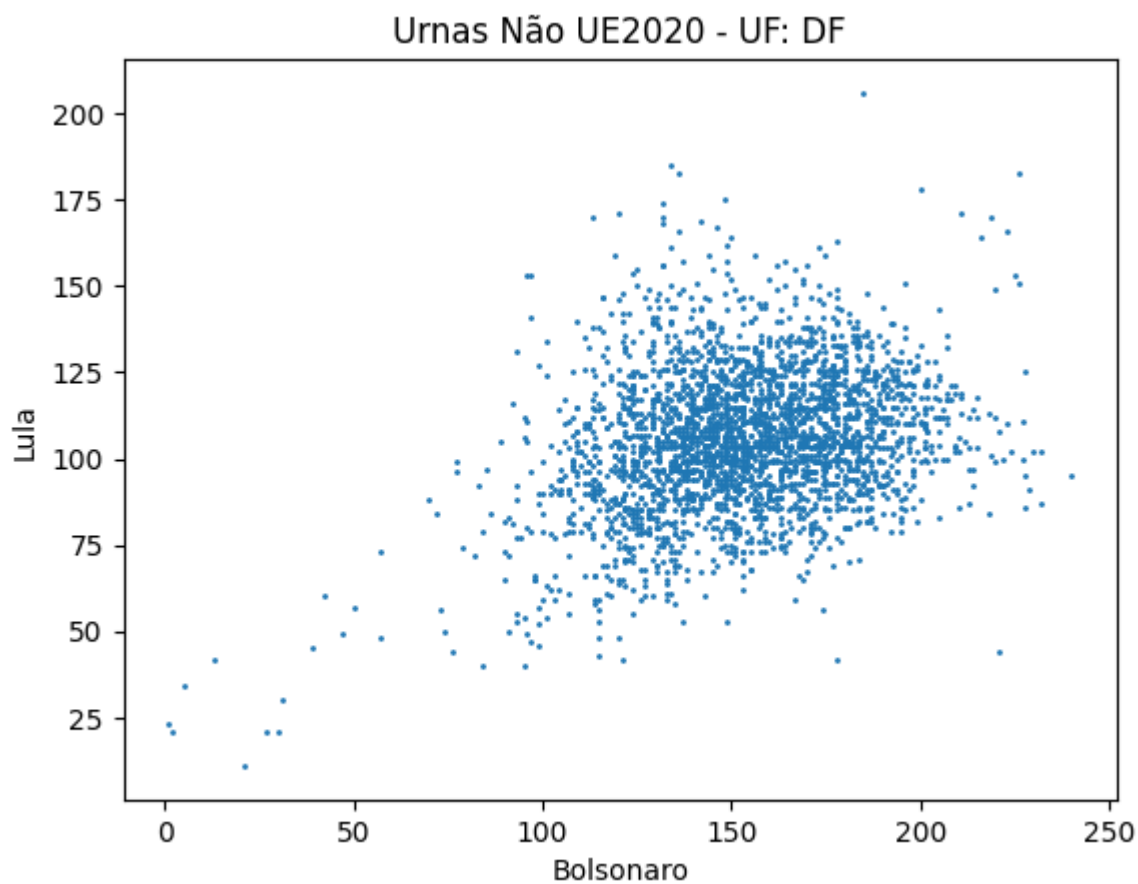
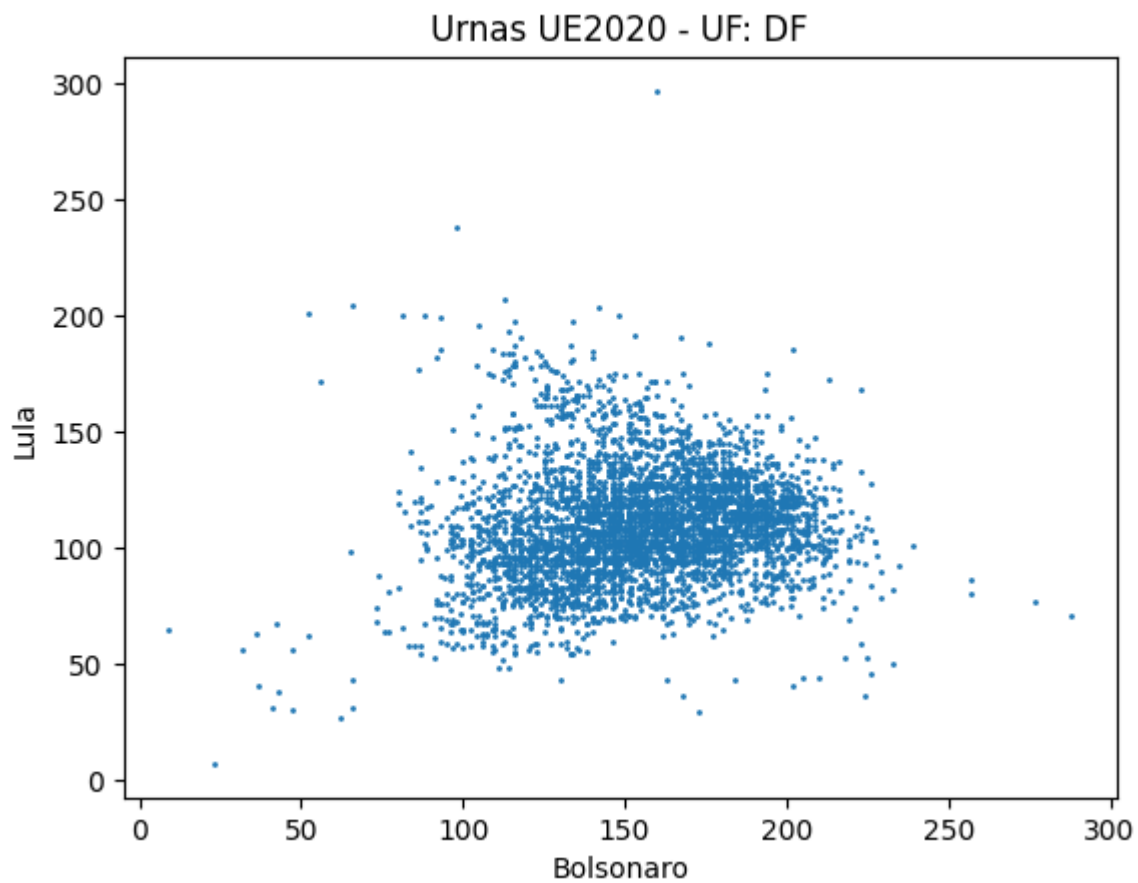
BA



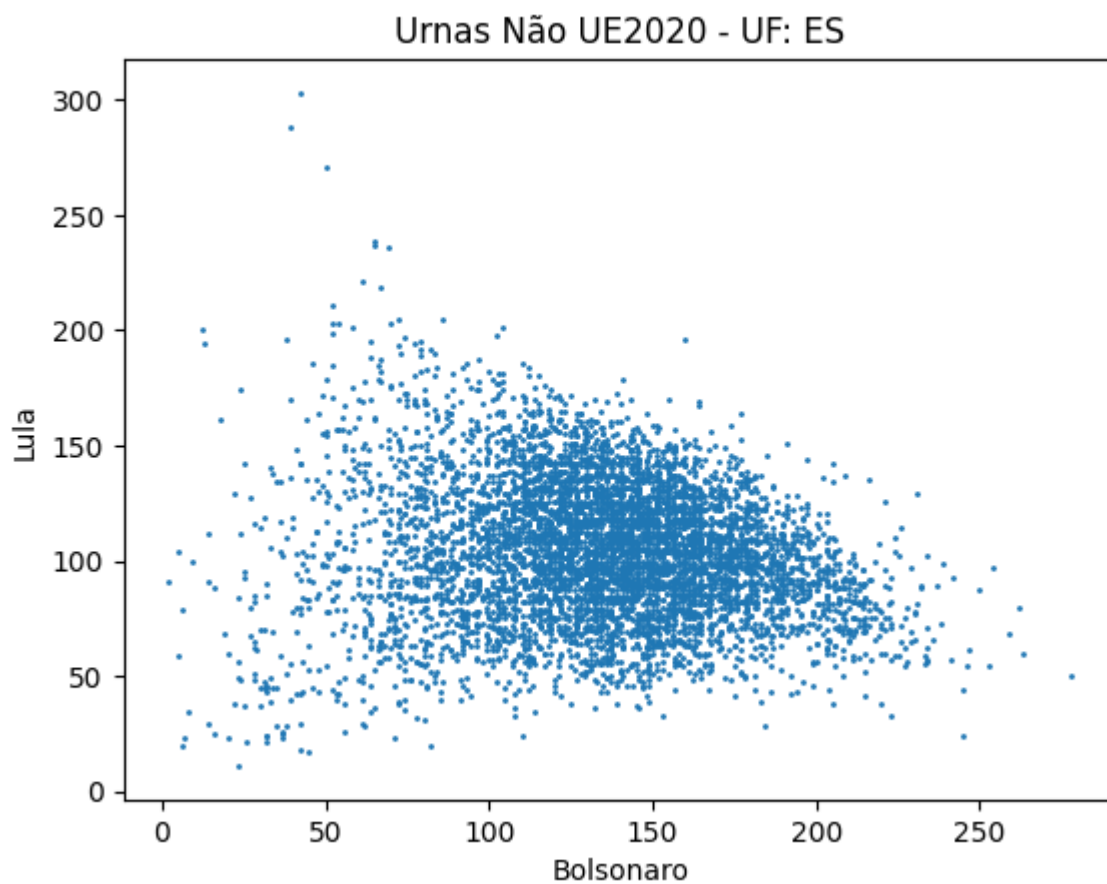
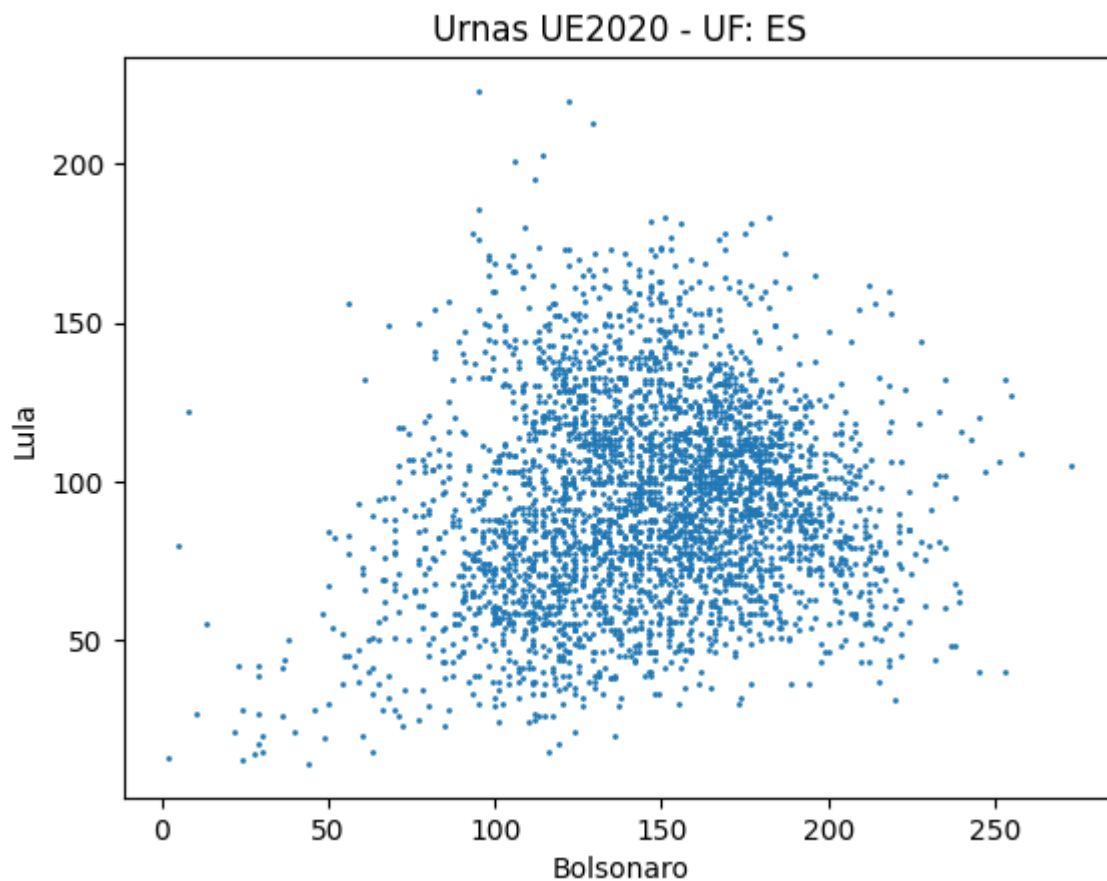
CE



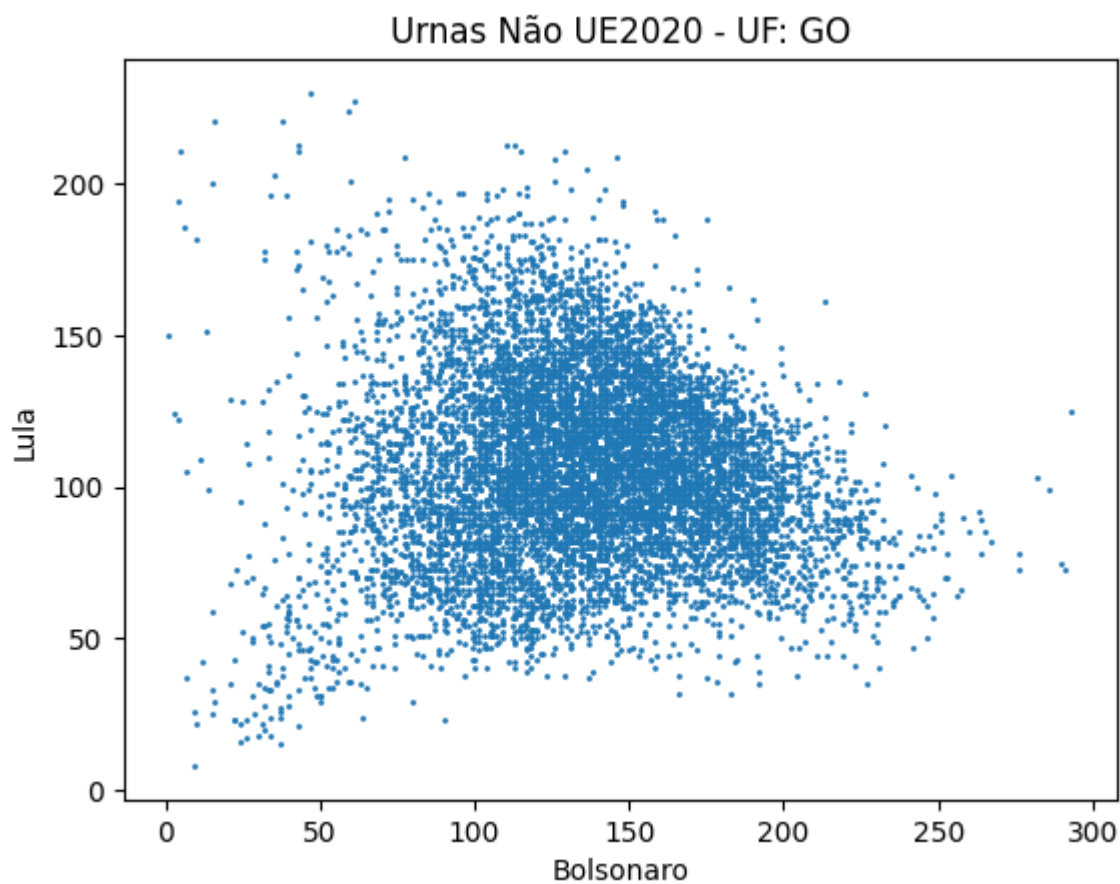
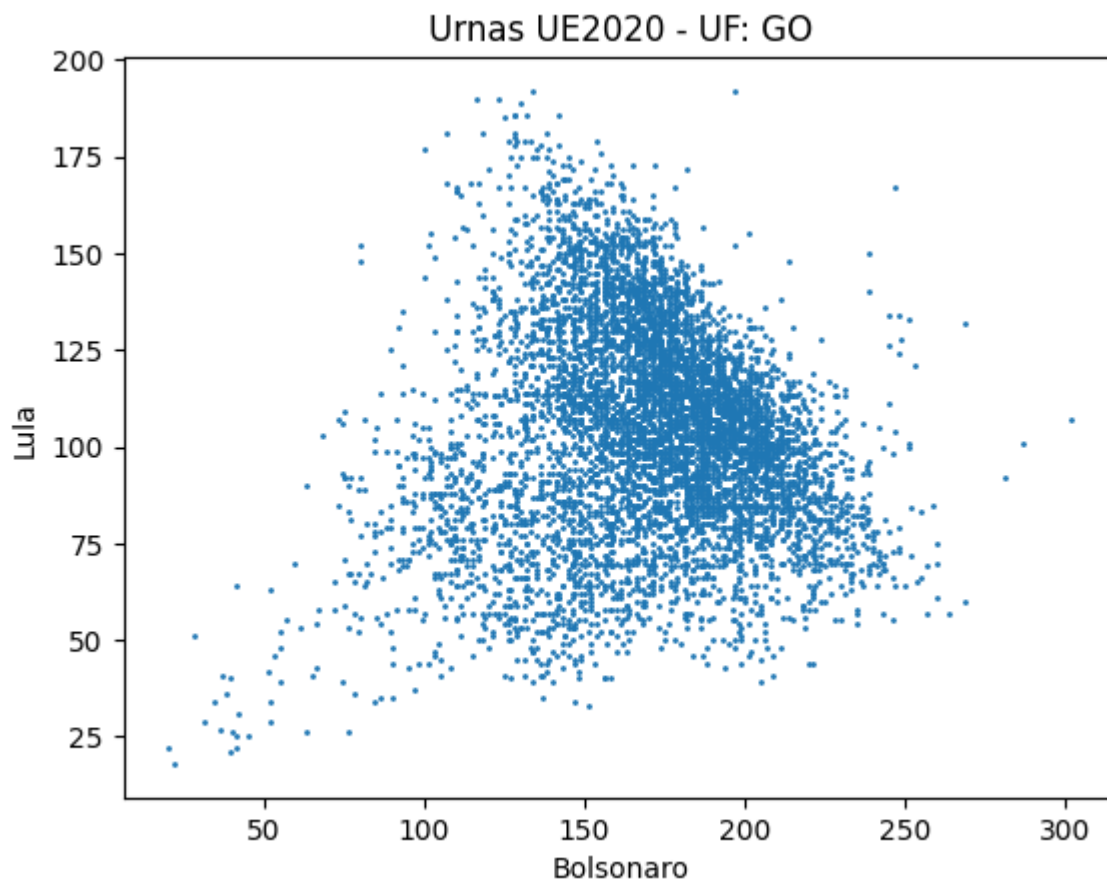
DF



ES

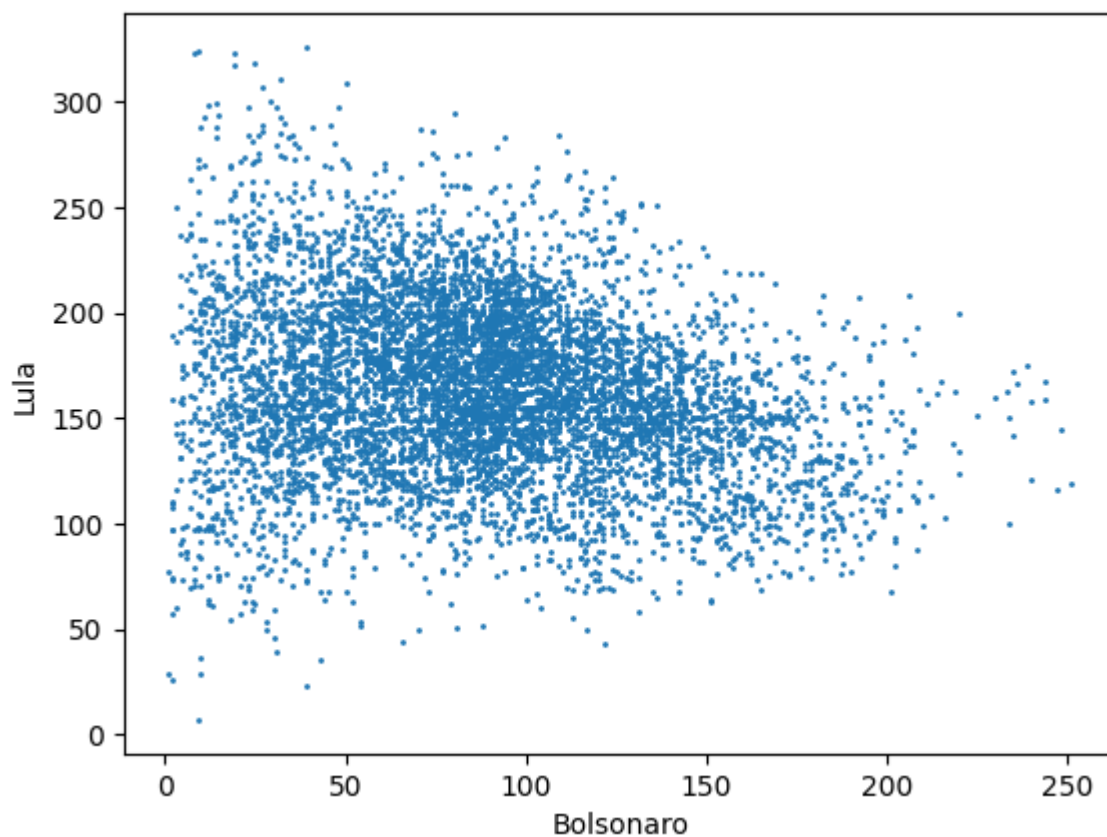


GO

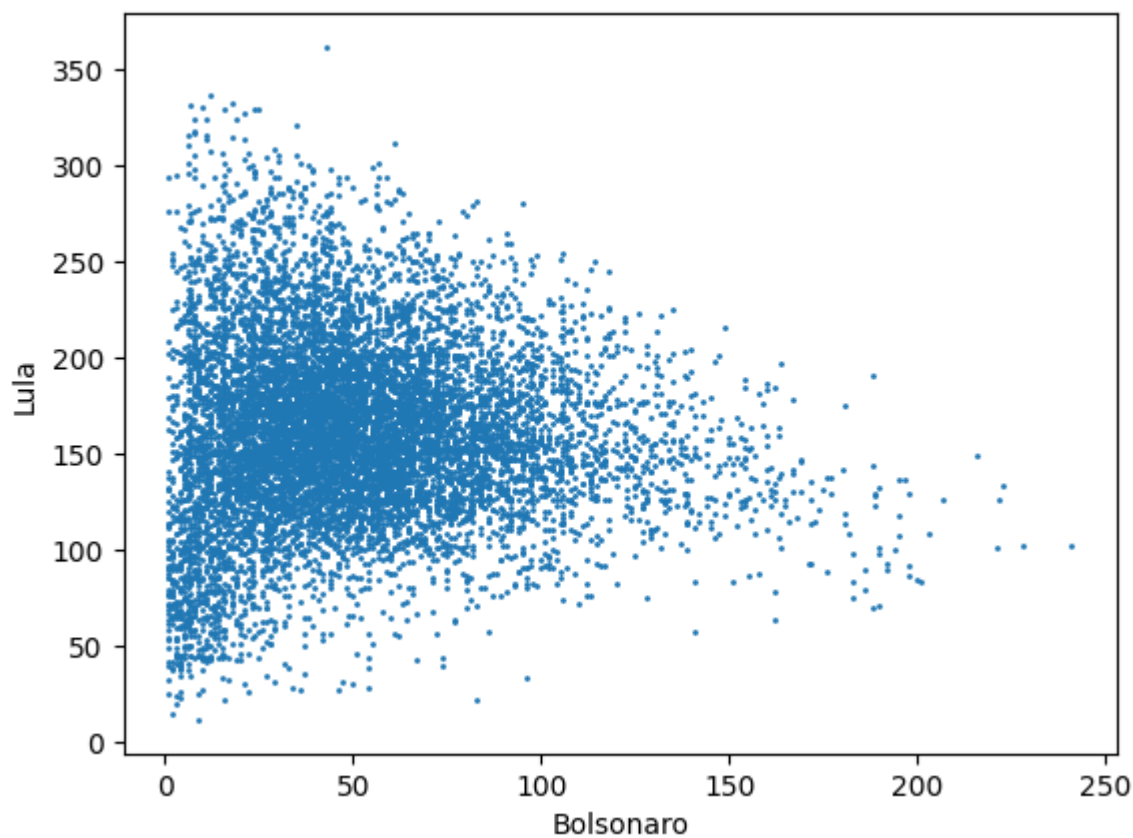


MA

Urnas UE2020 - UF: MA

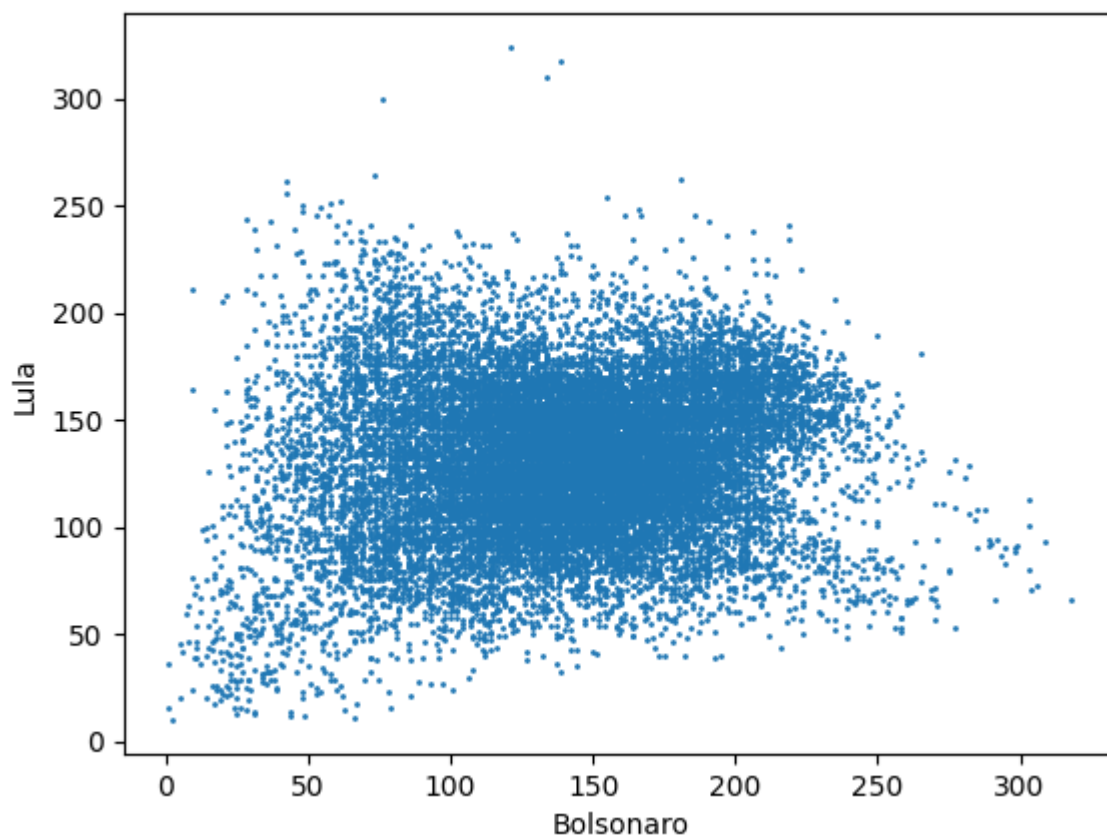


Urnas Não UE2020 - UF: MA

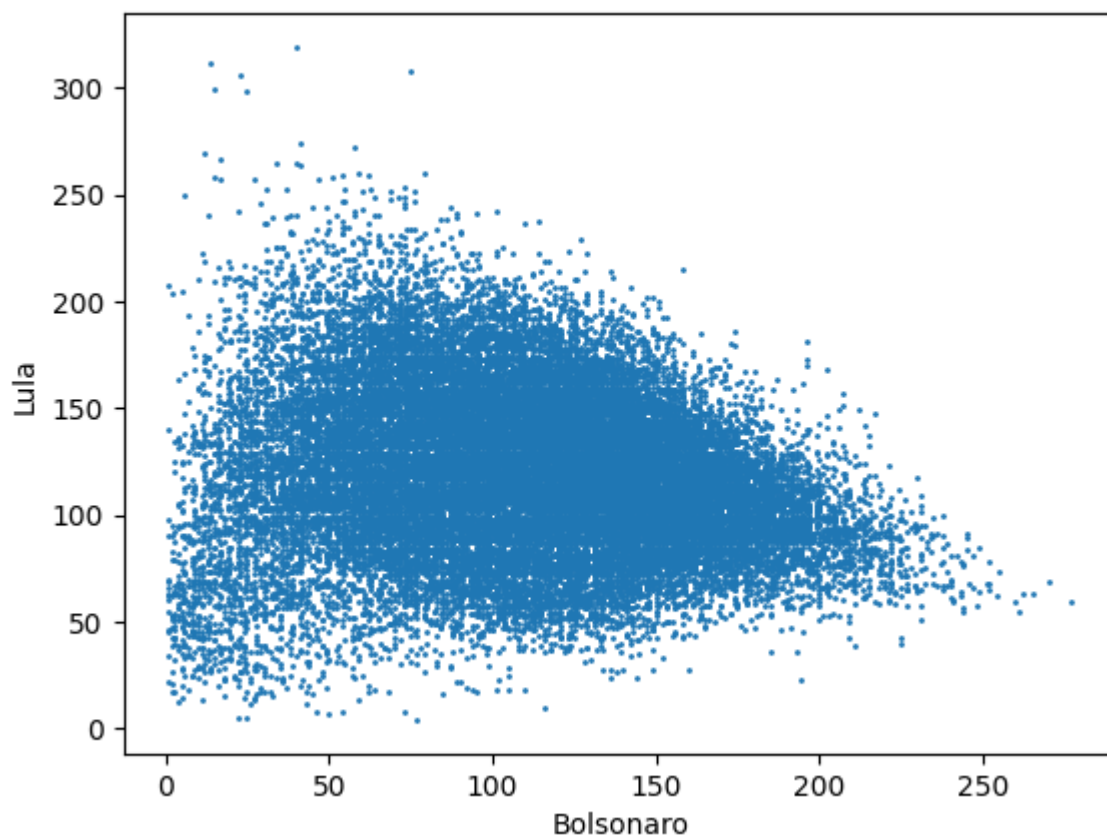


MG

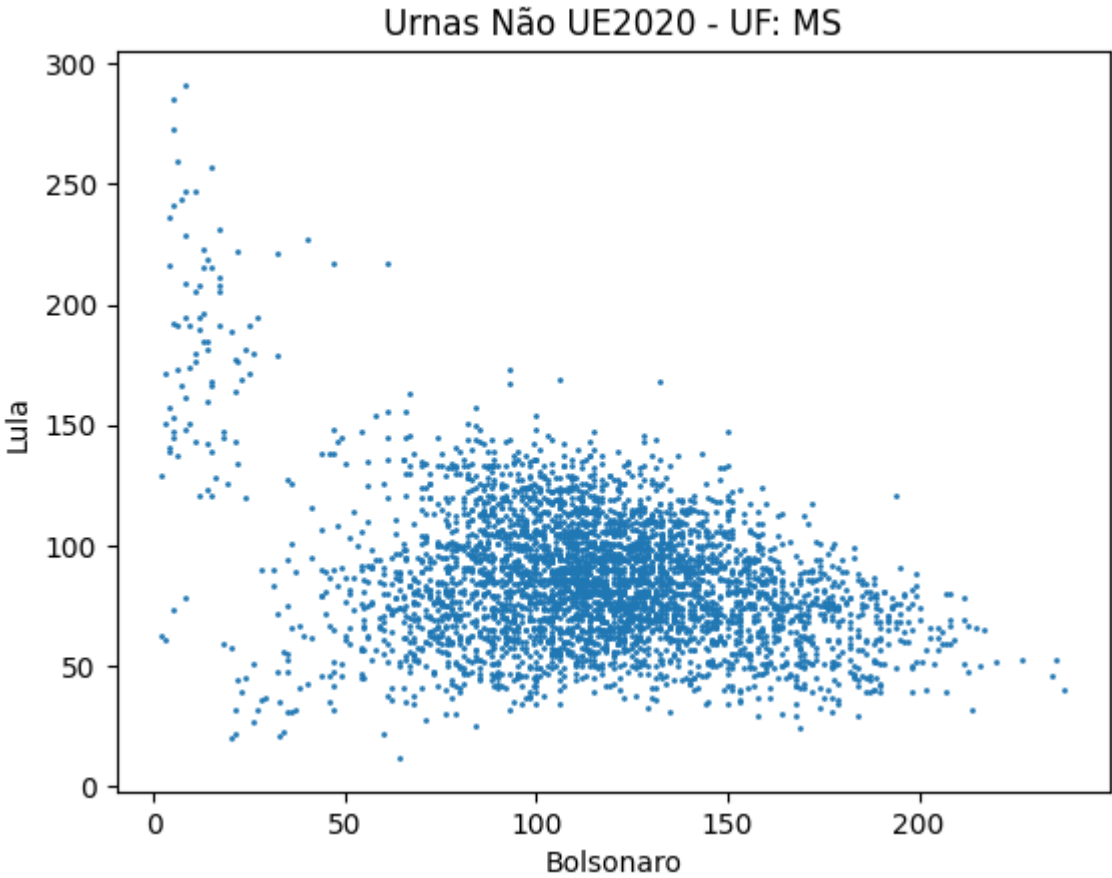
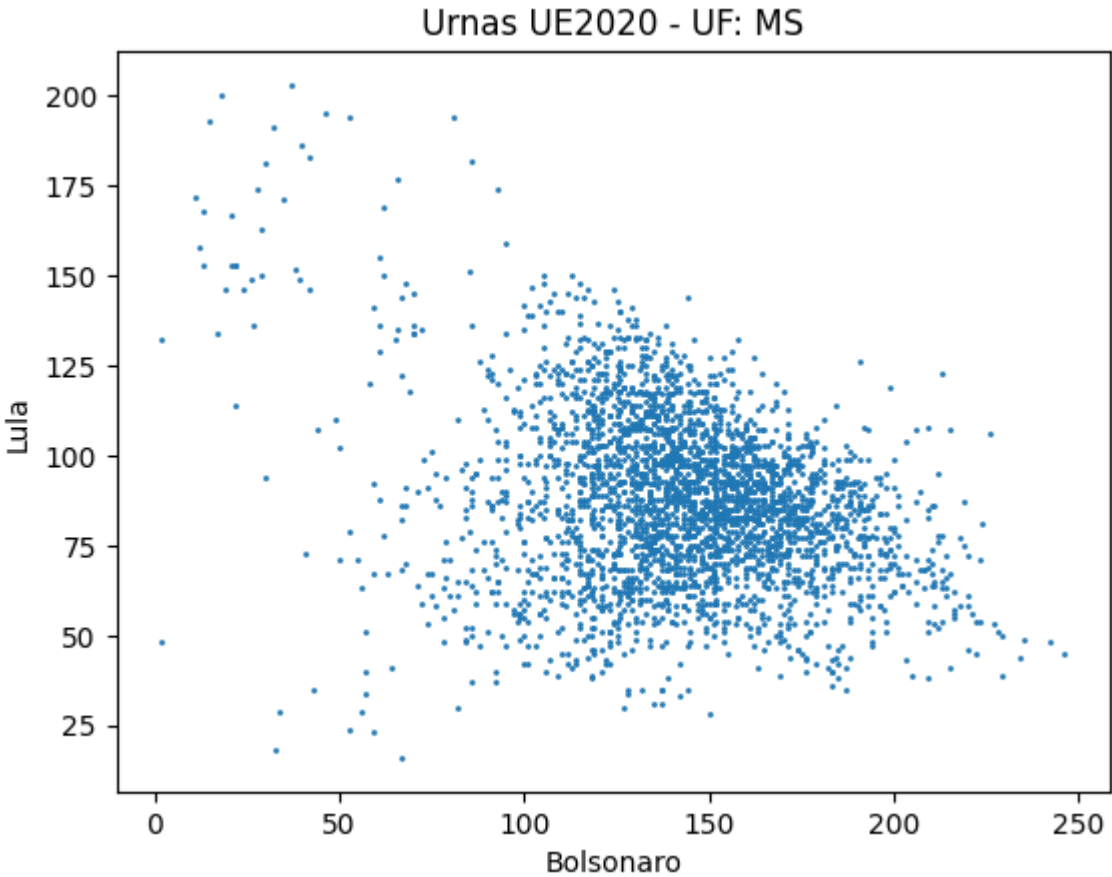
Urnas UE2020 - UF: MG



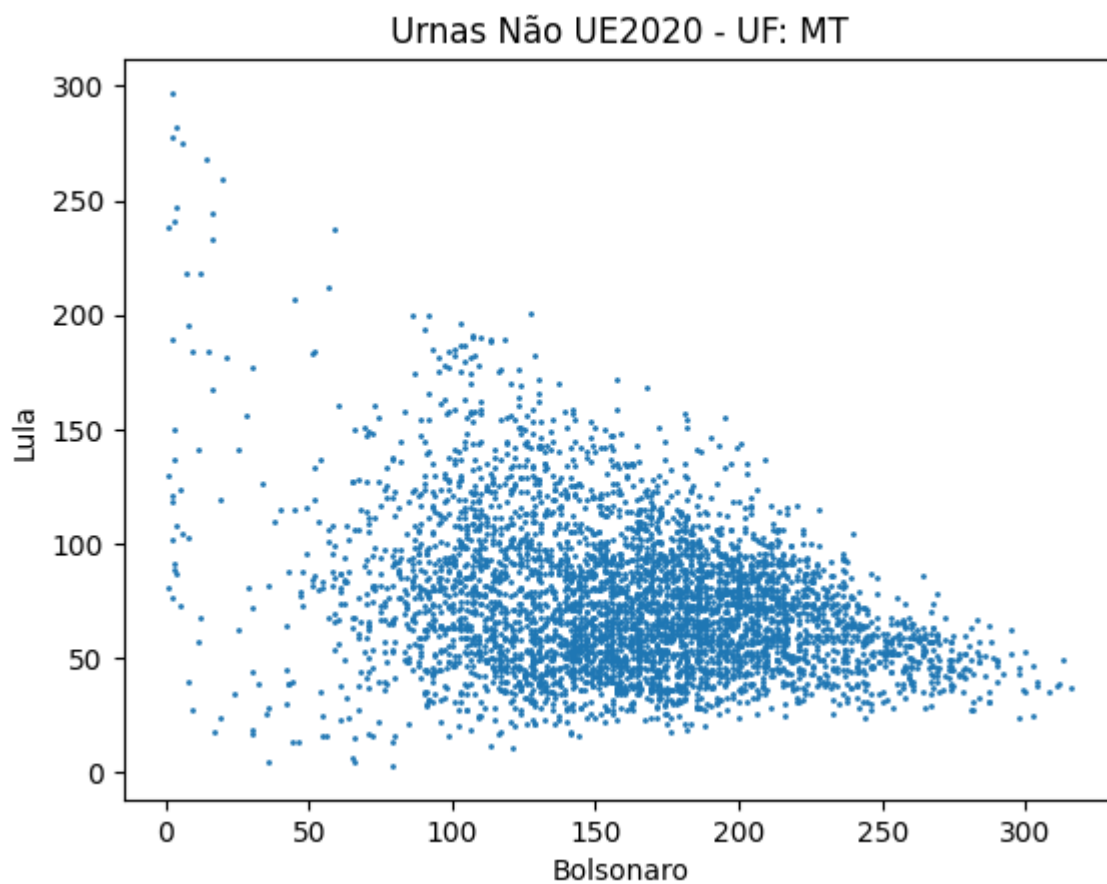
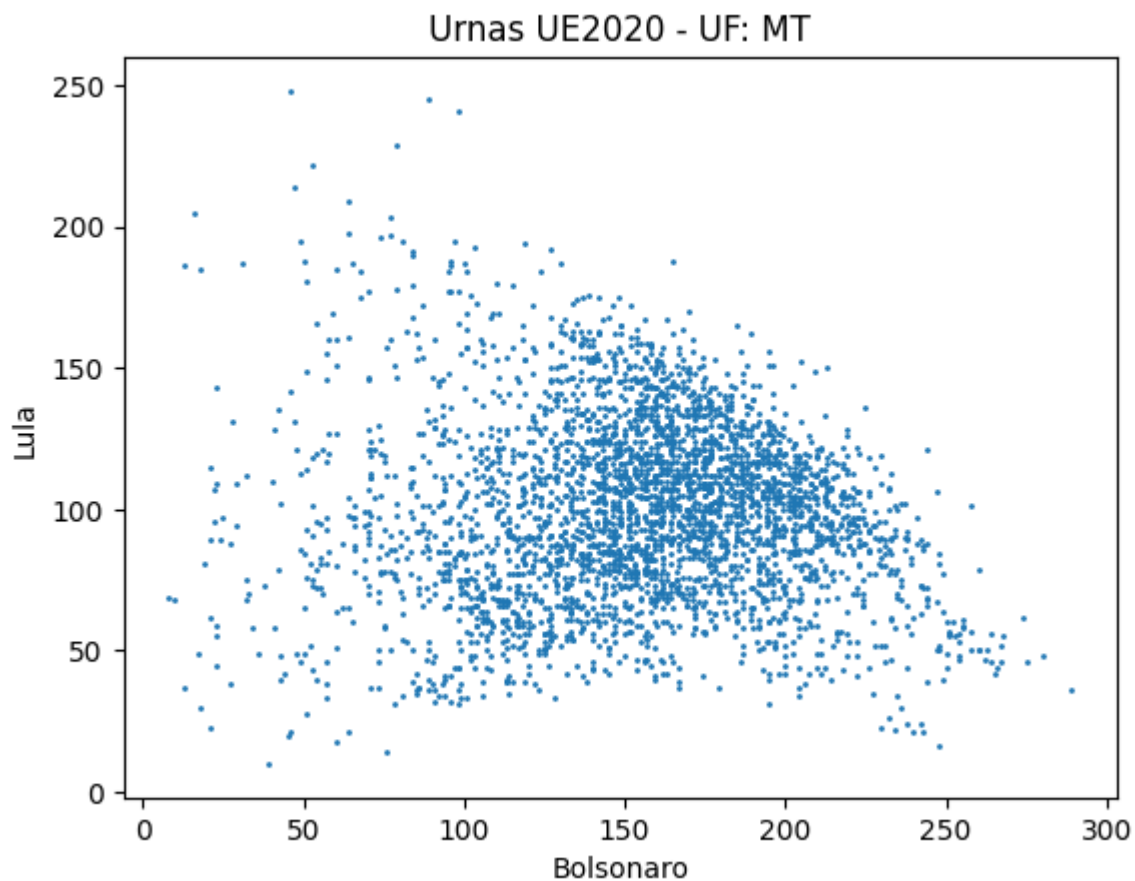
Urnas Não UE2020 - UF: MG



MS

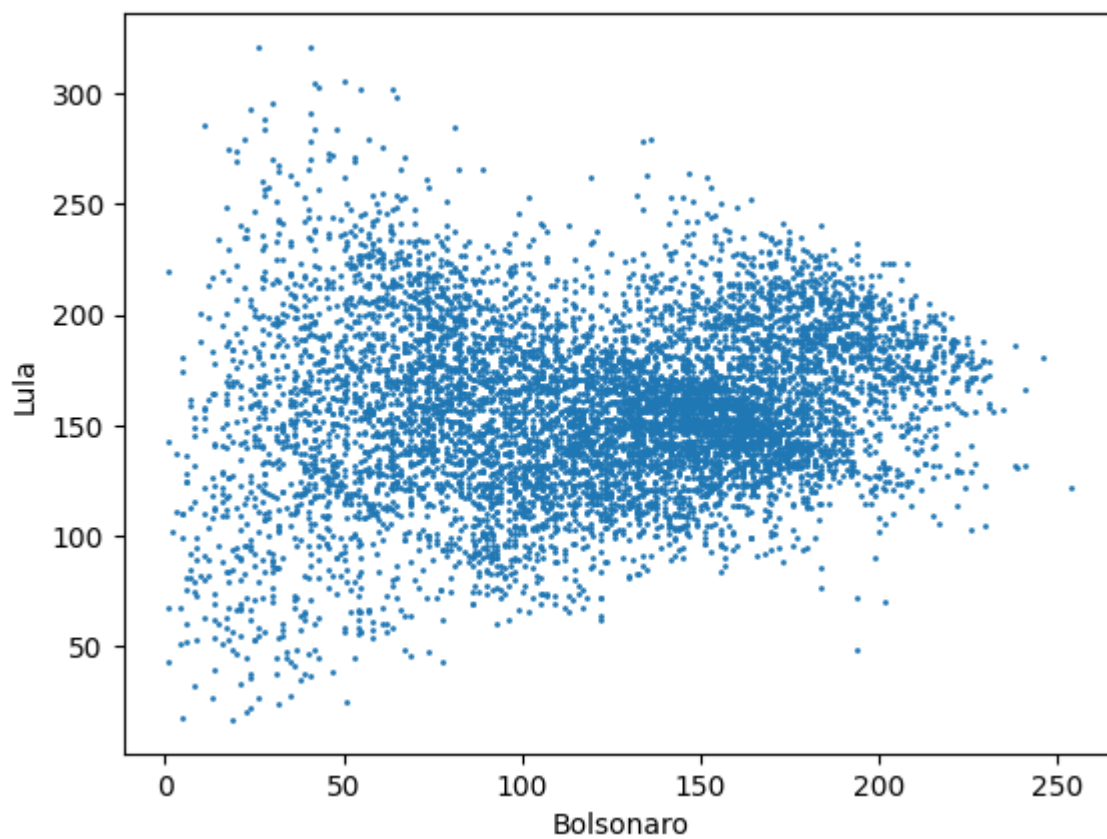


MT

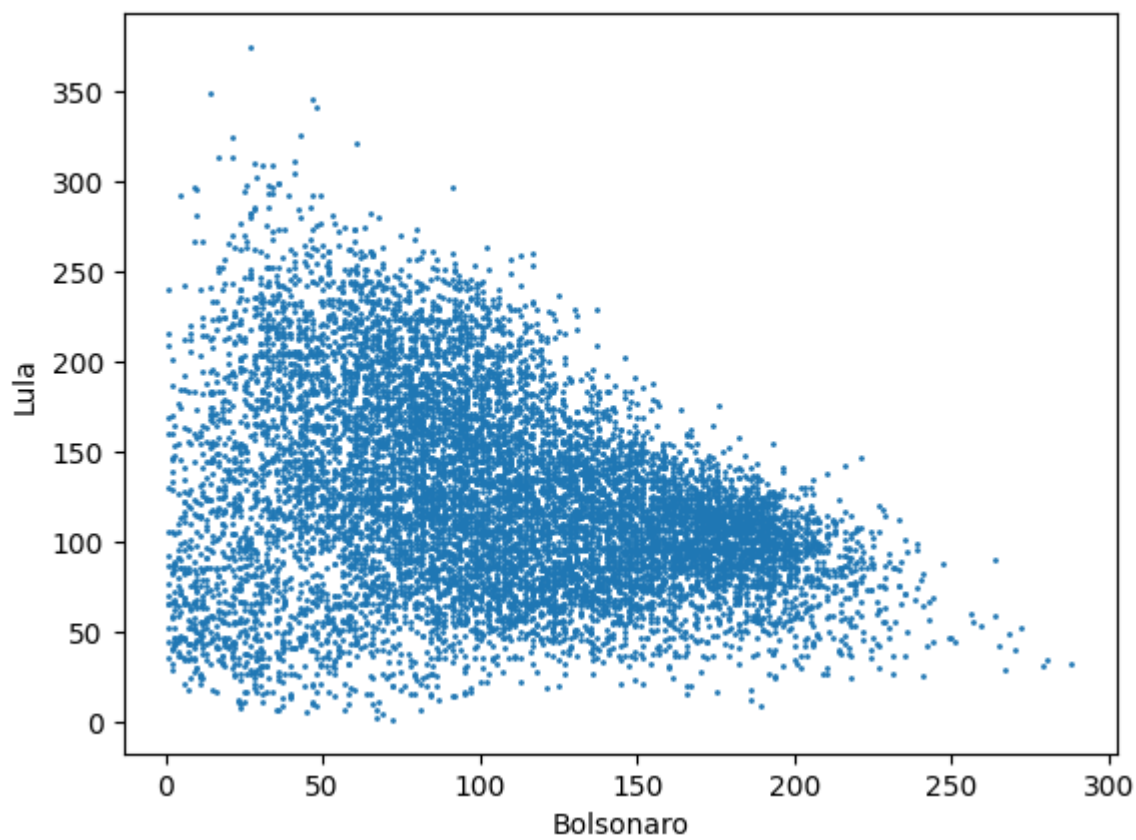


PA

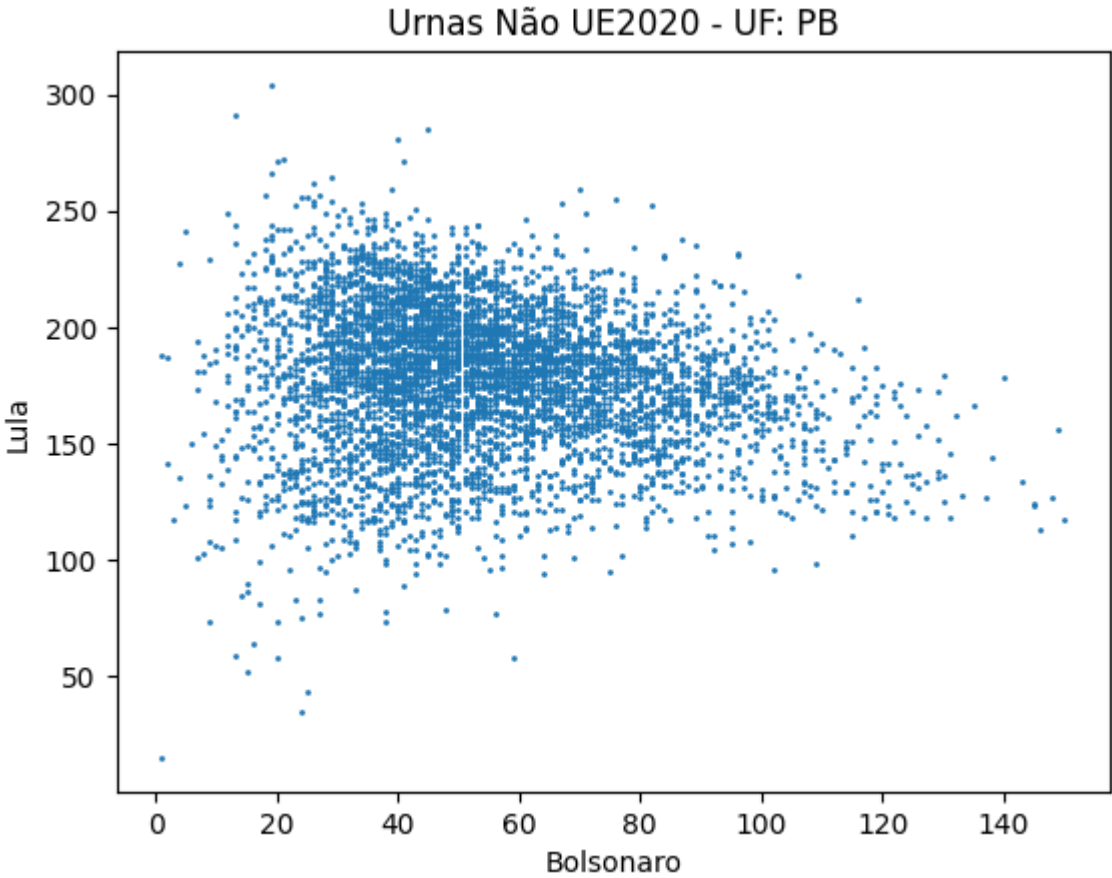
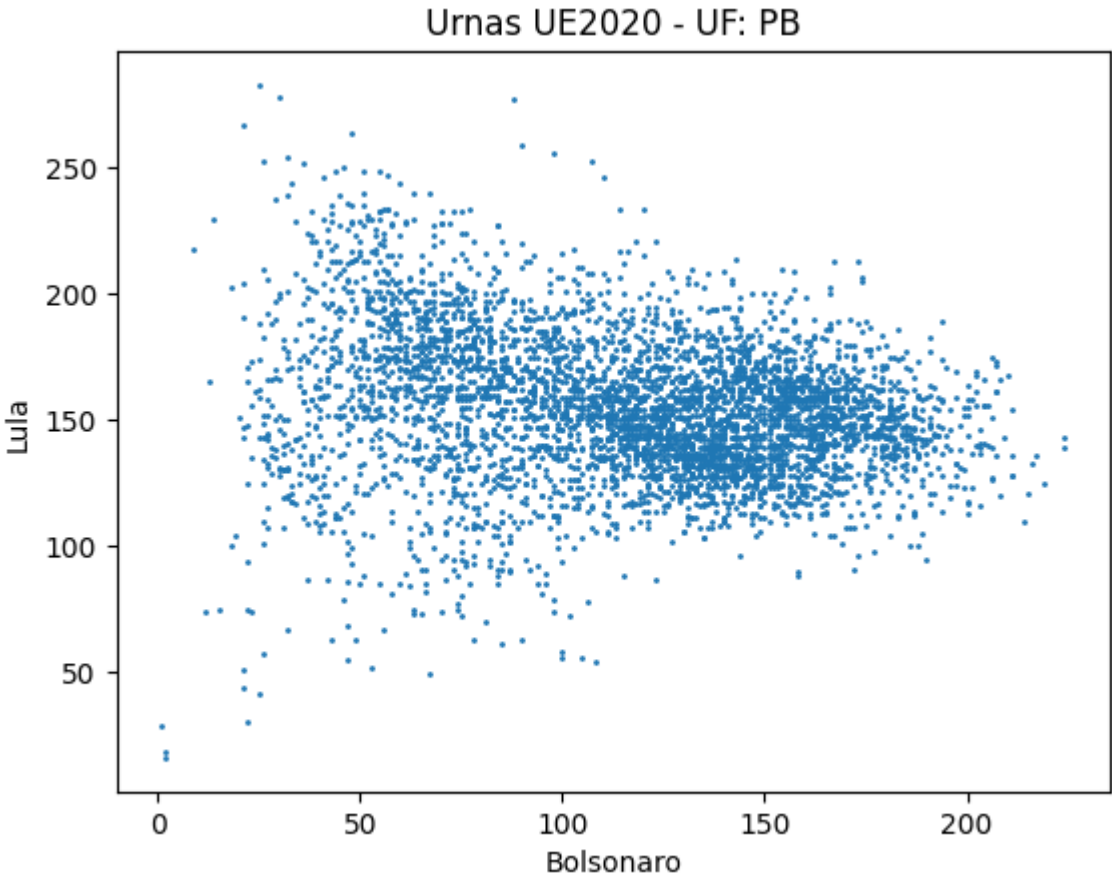
Urnas UE2020 - UF: PA



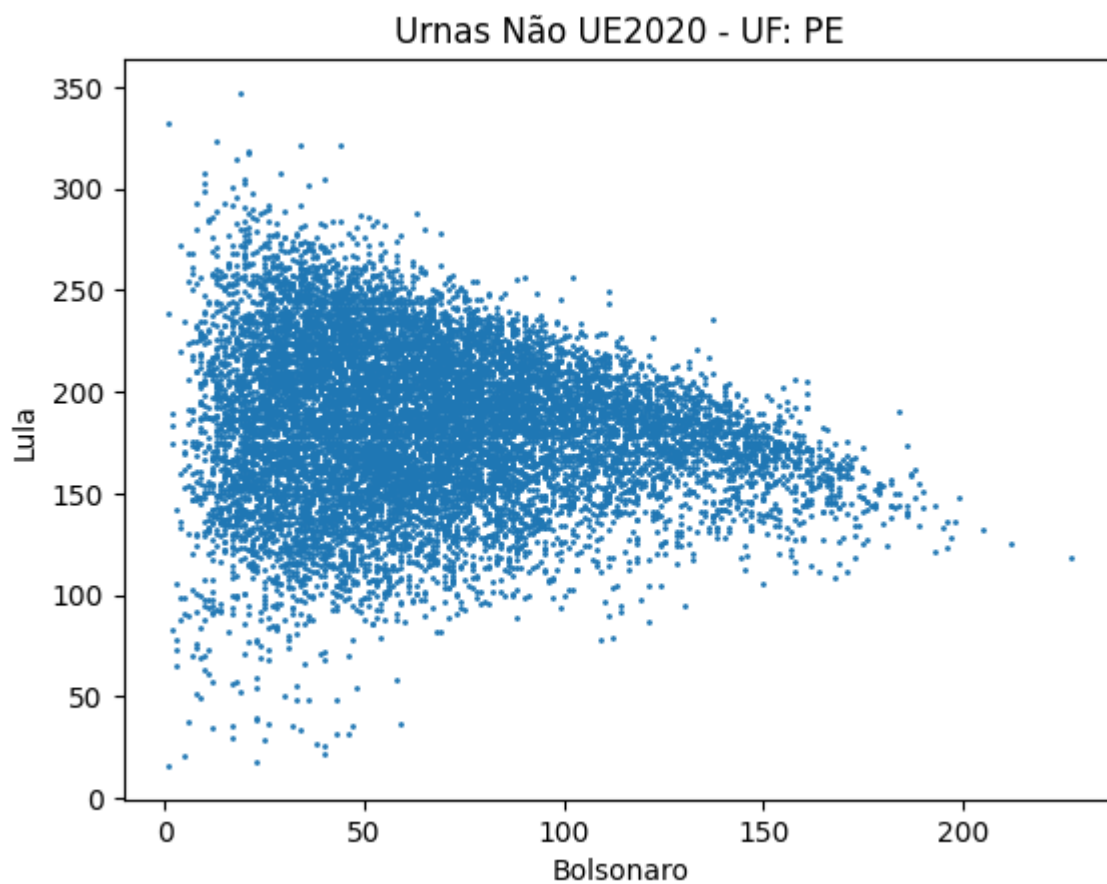
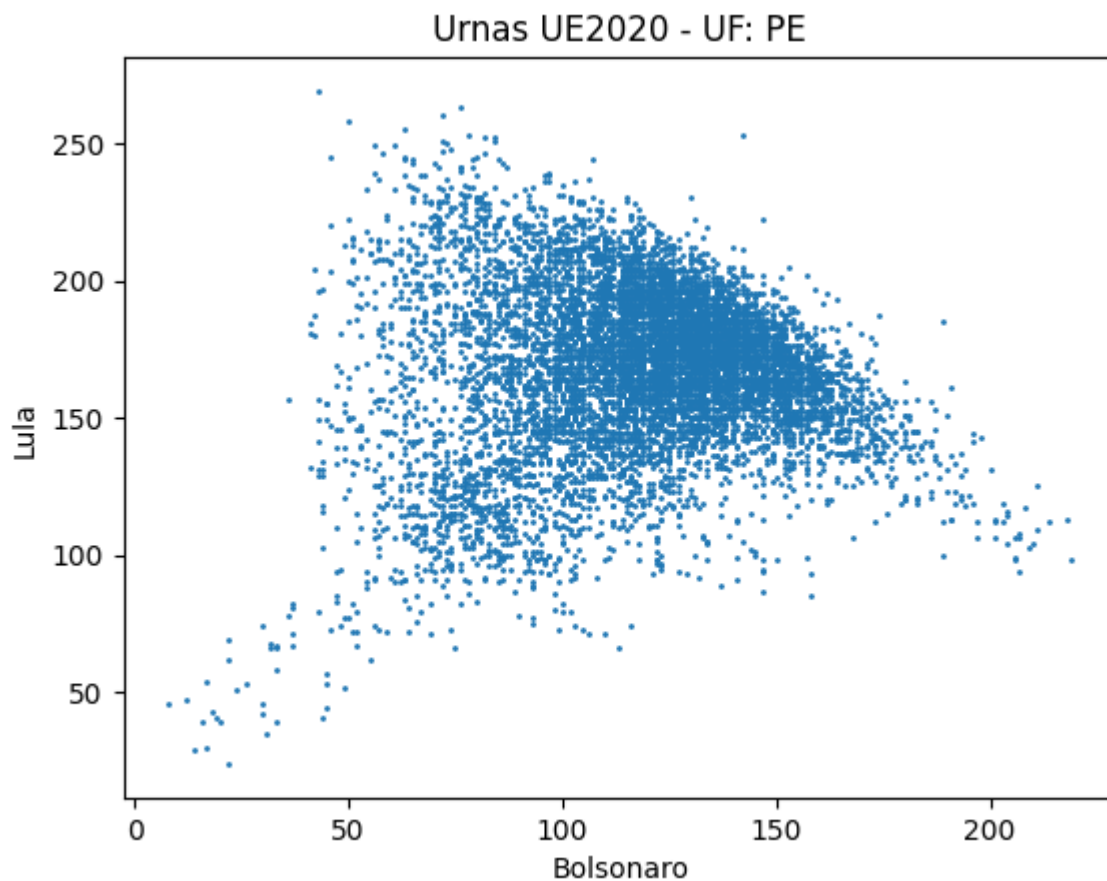
Urnas Não UE2020 - UF: PA



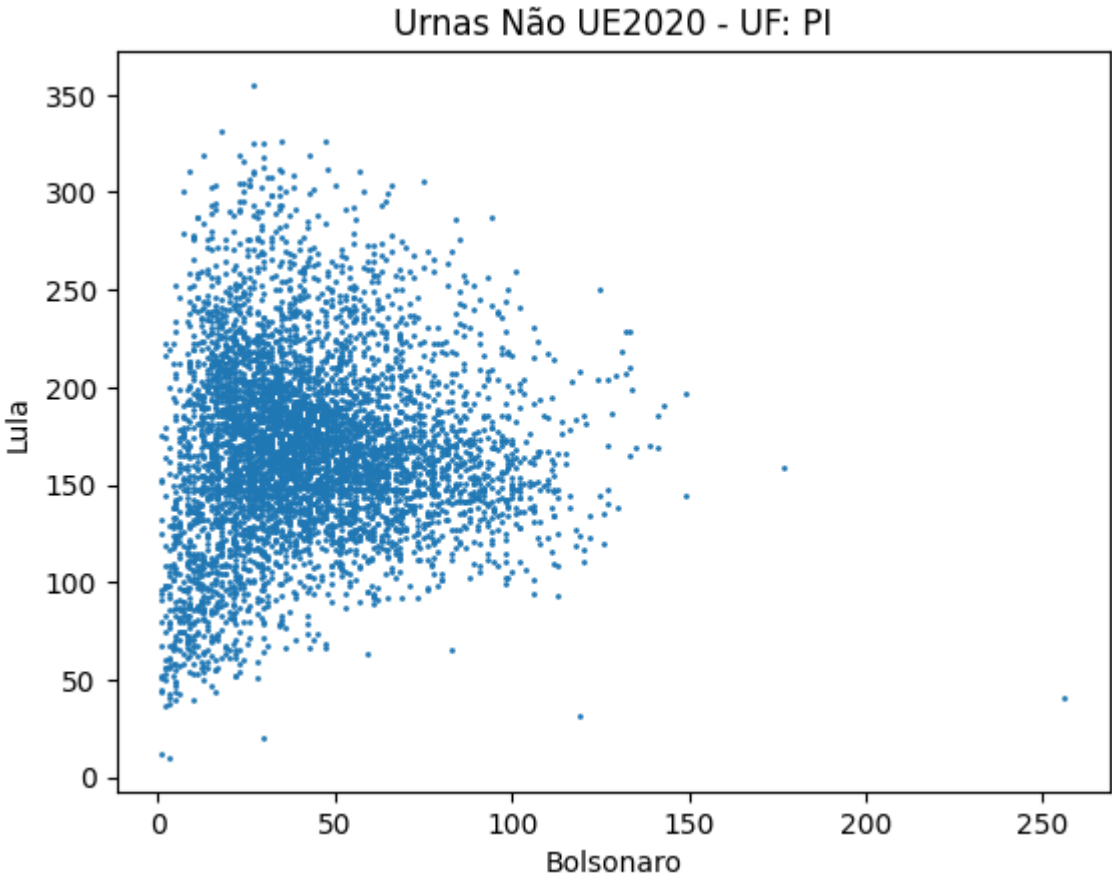
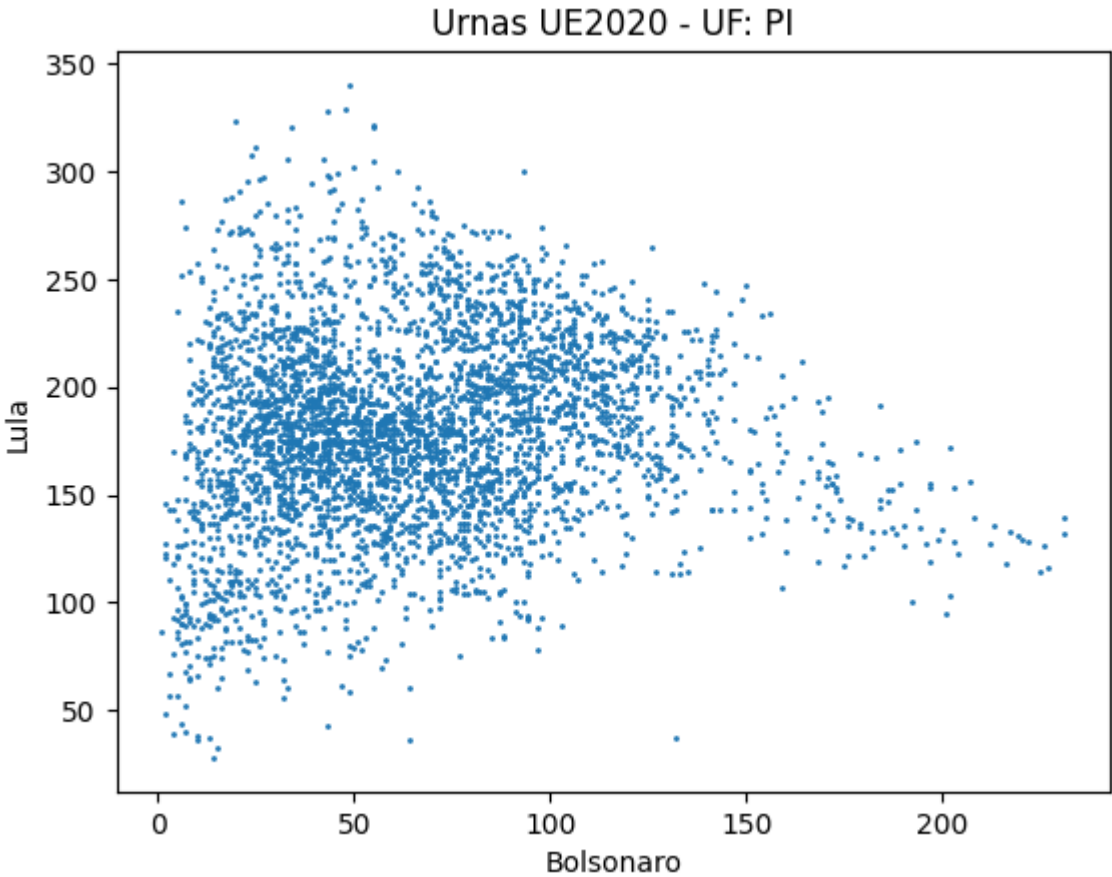
PB



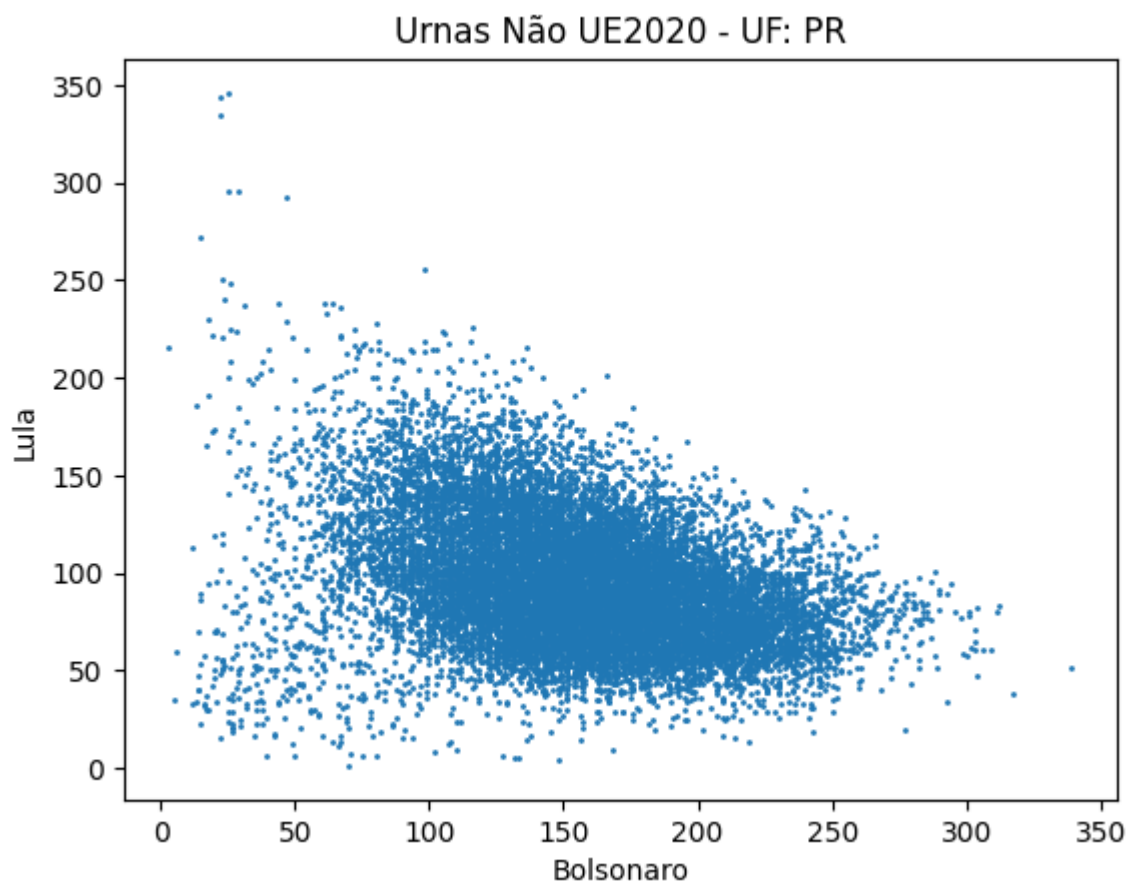
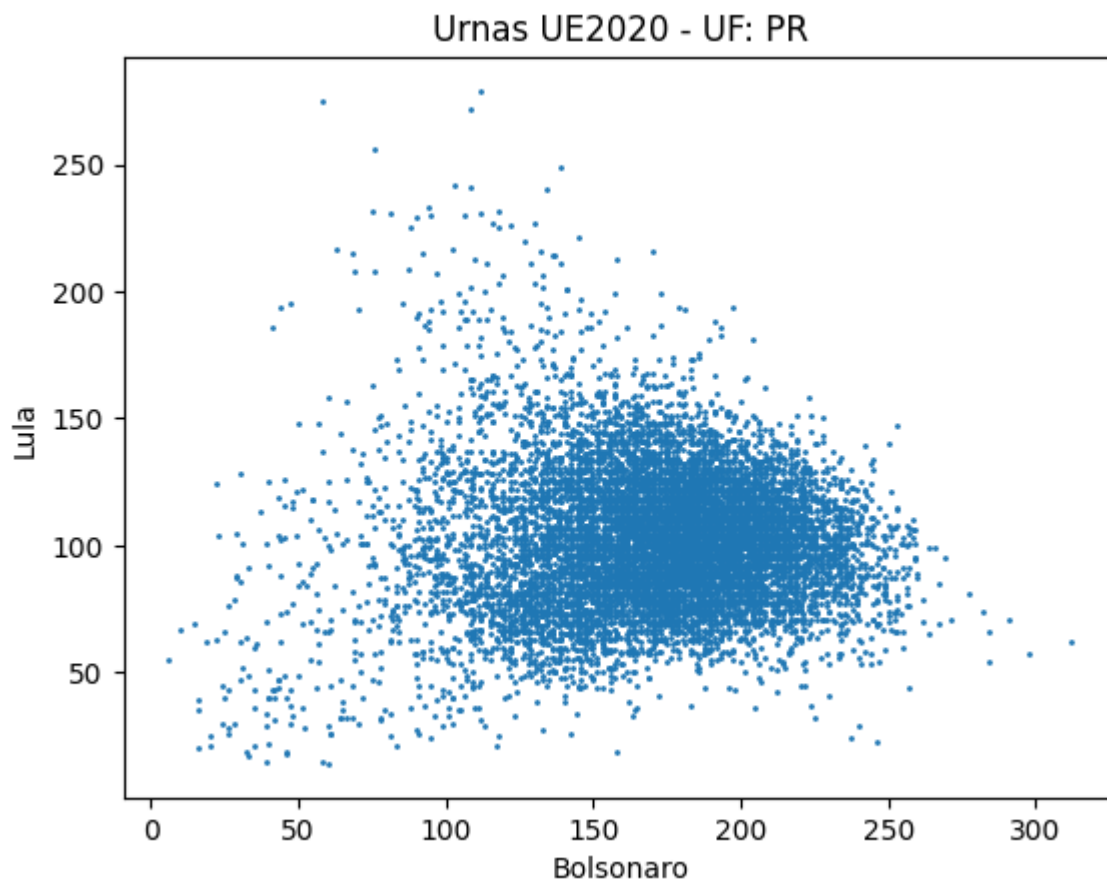
PE



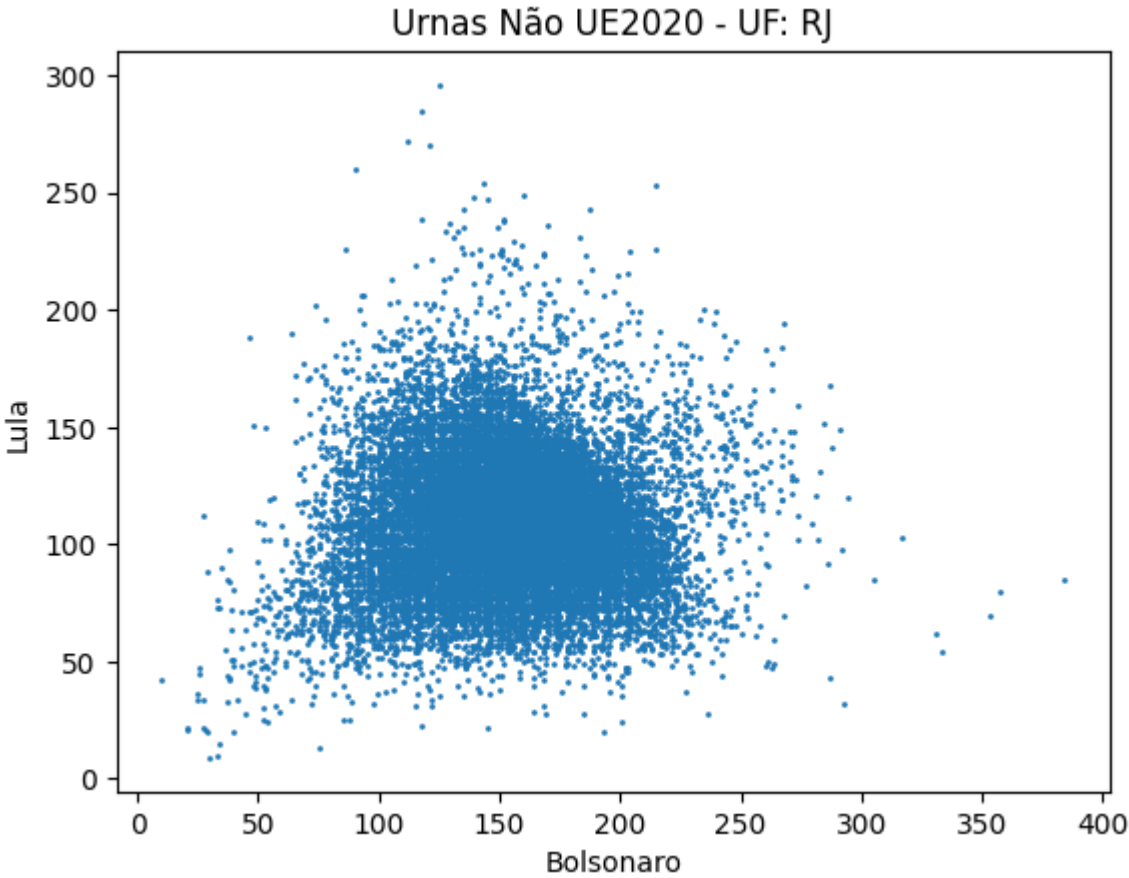
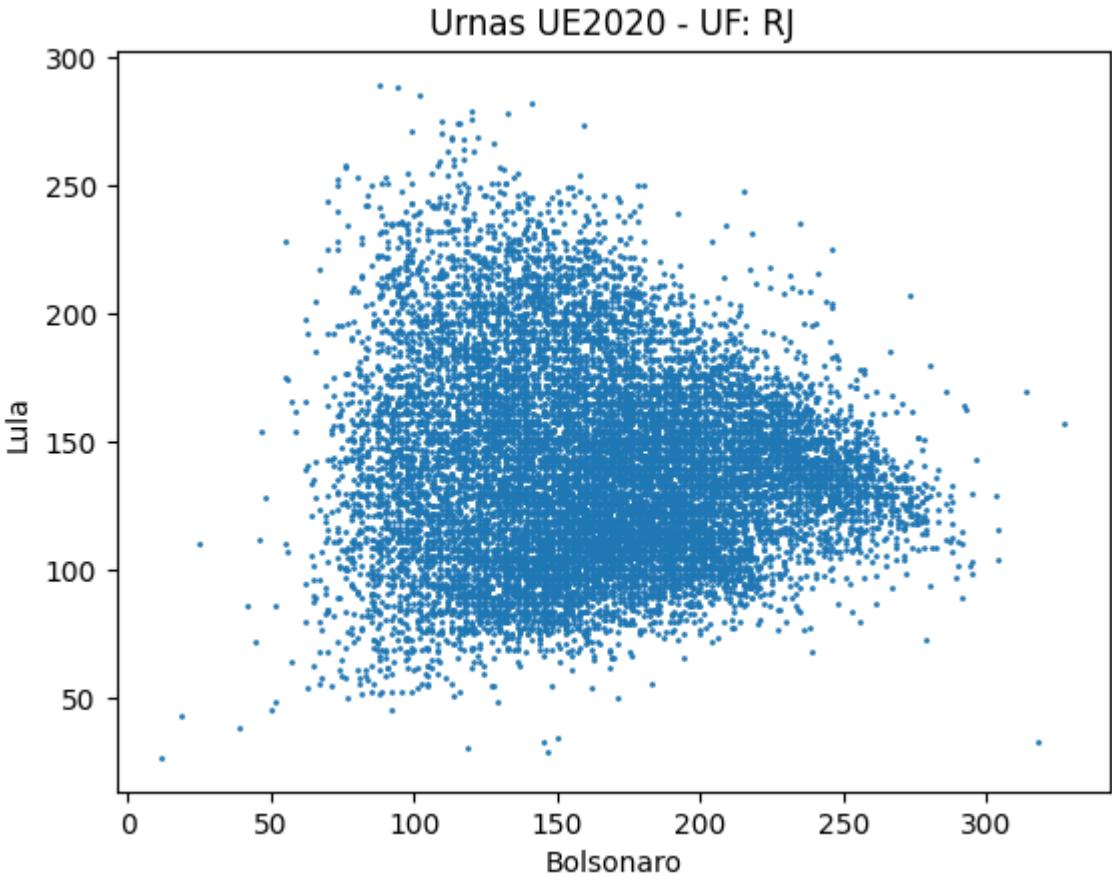
PI



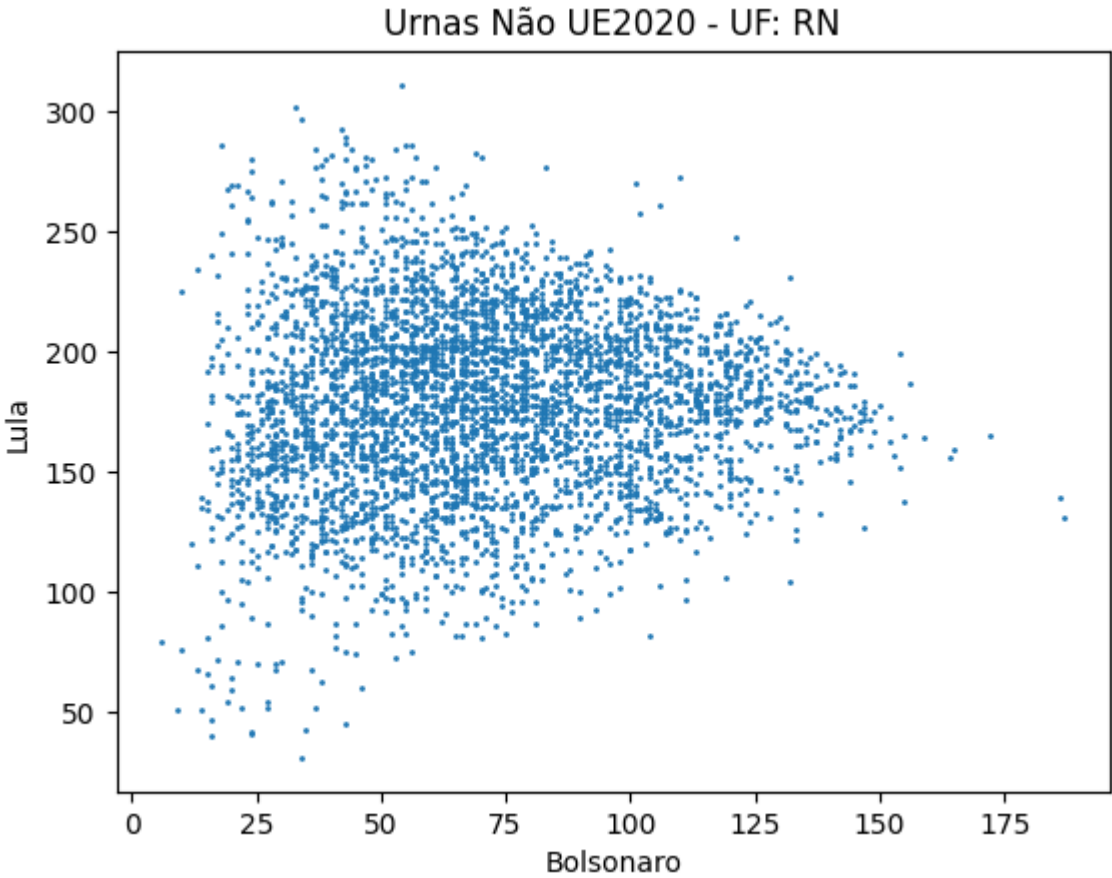
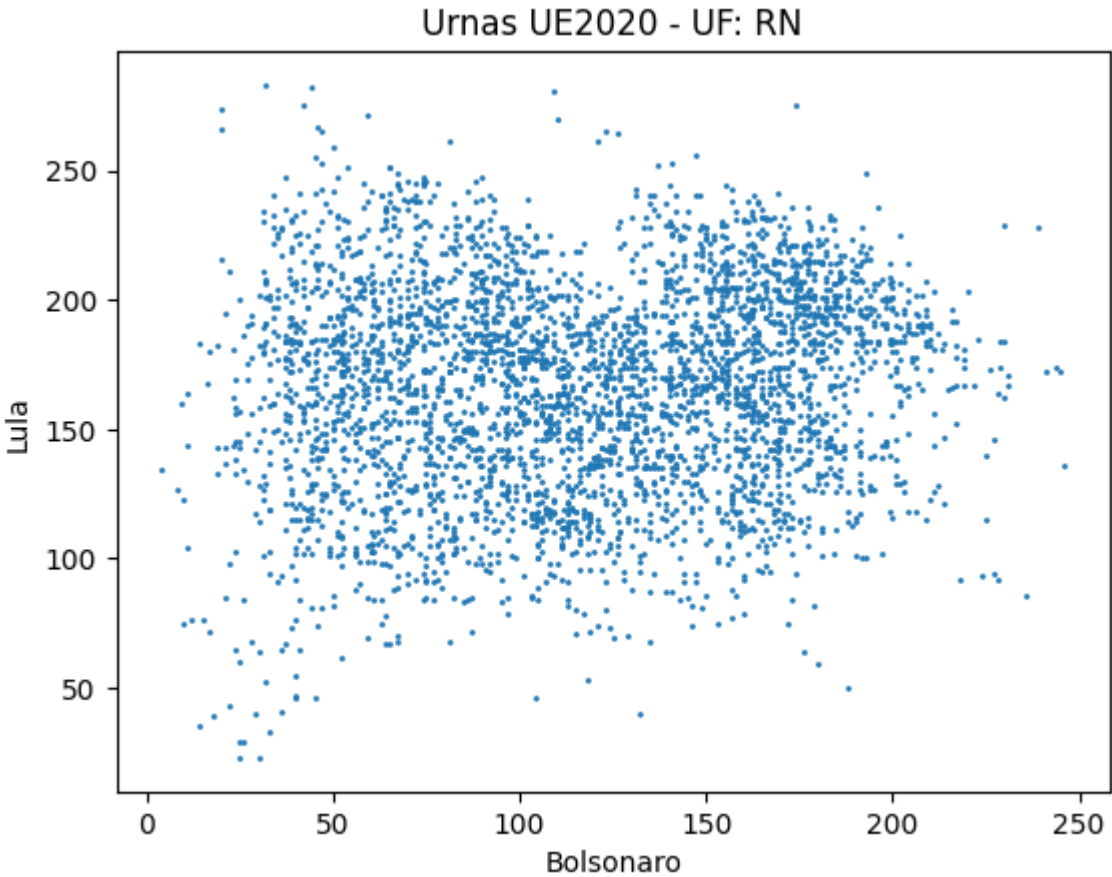
PR



RJ

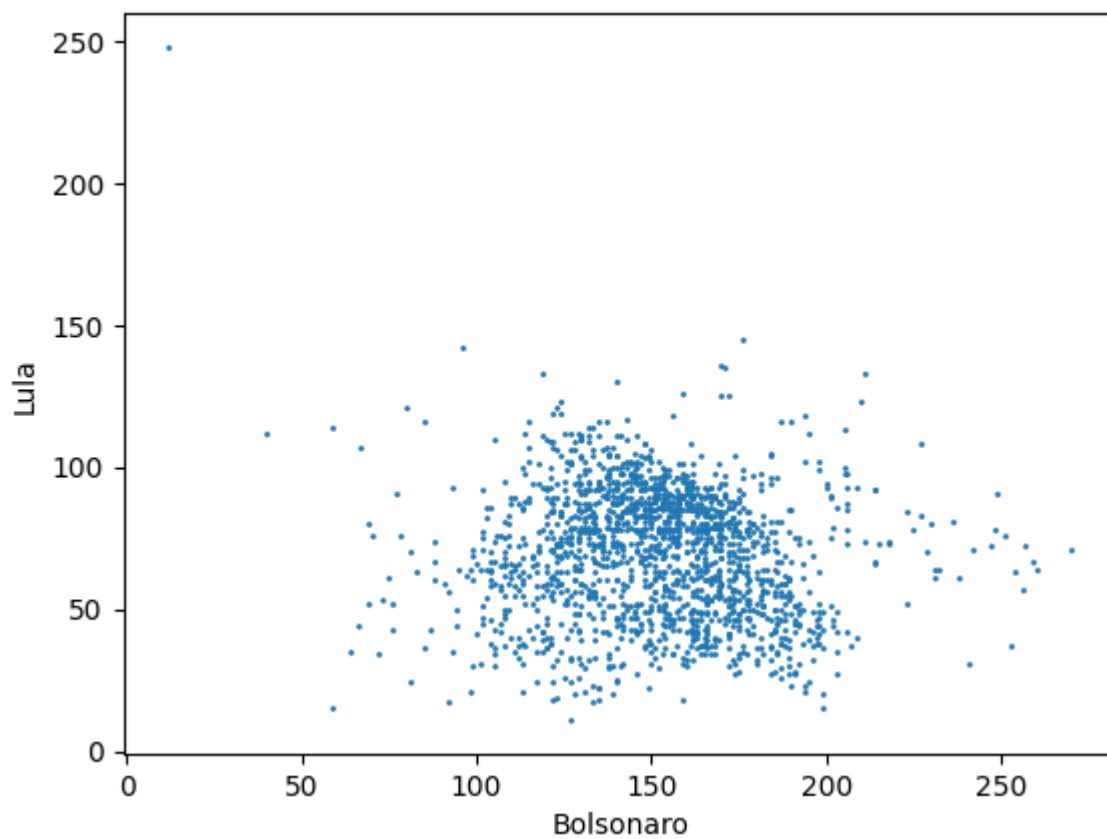


RN

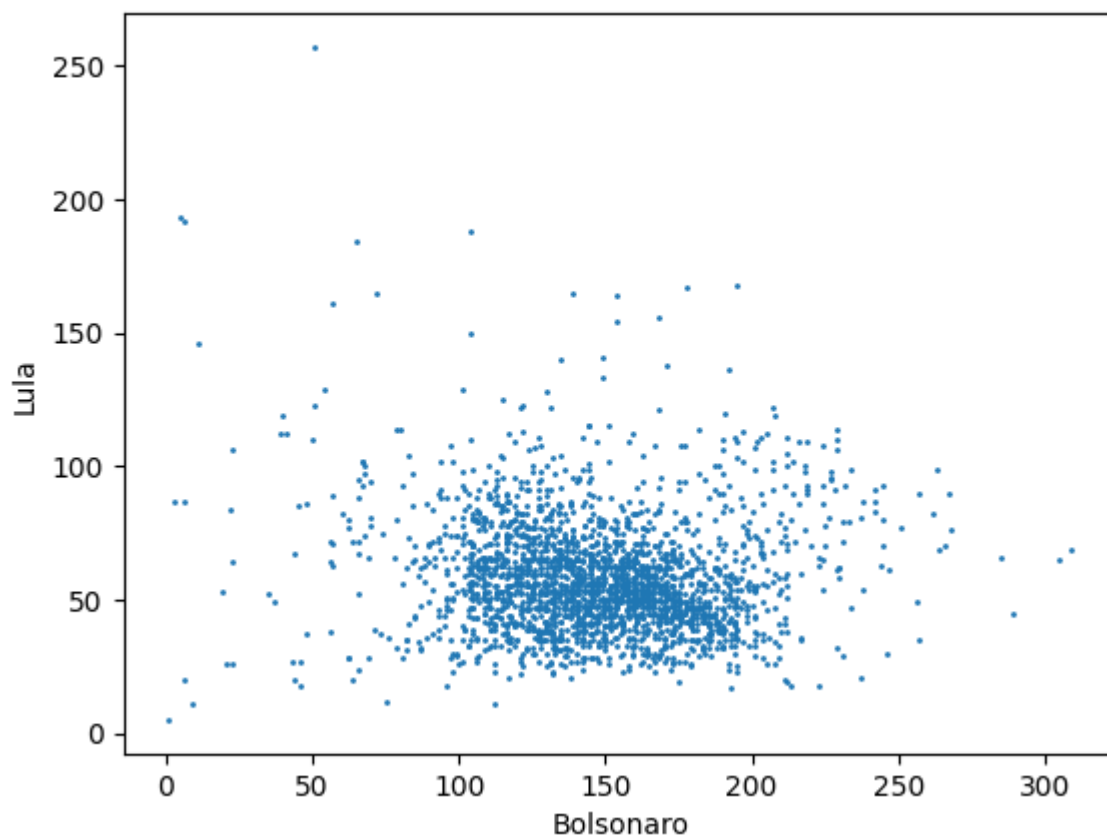


RO

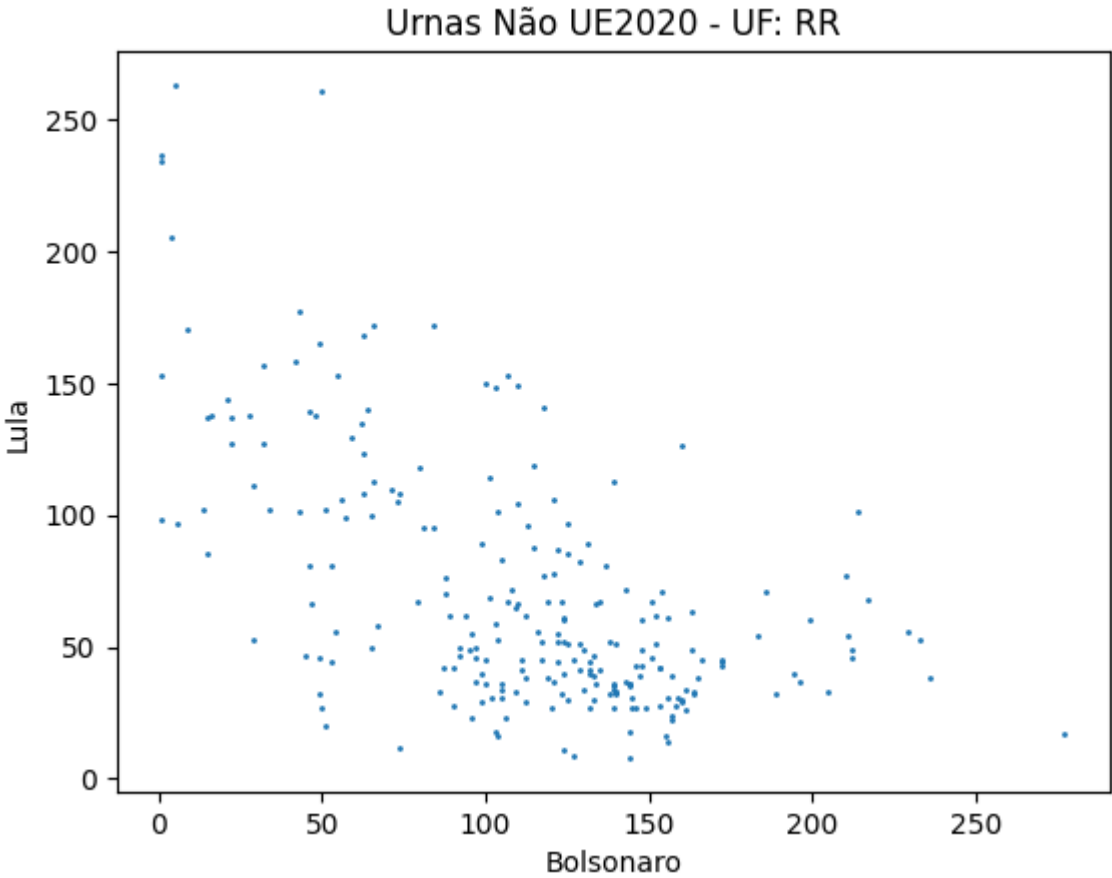
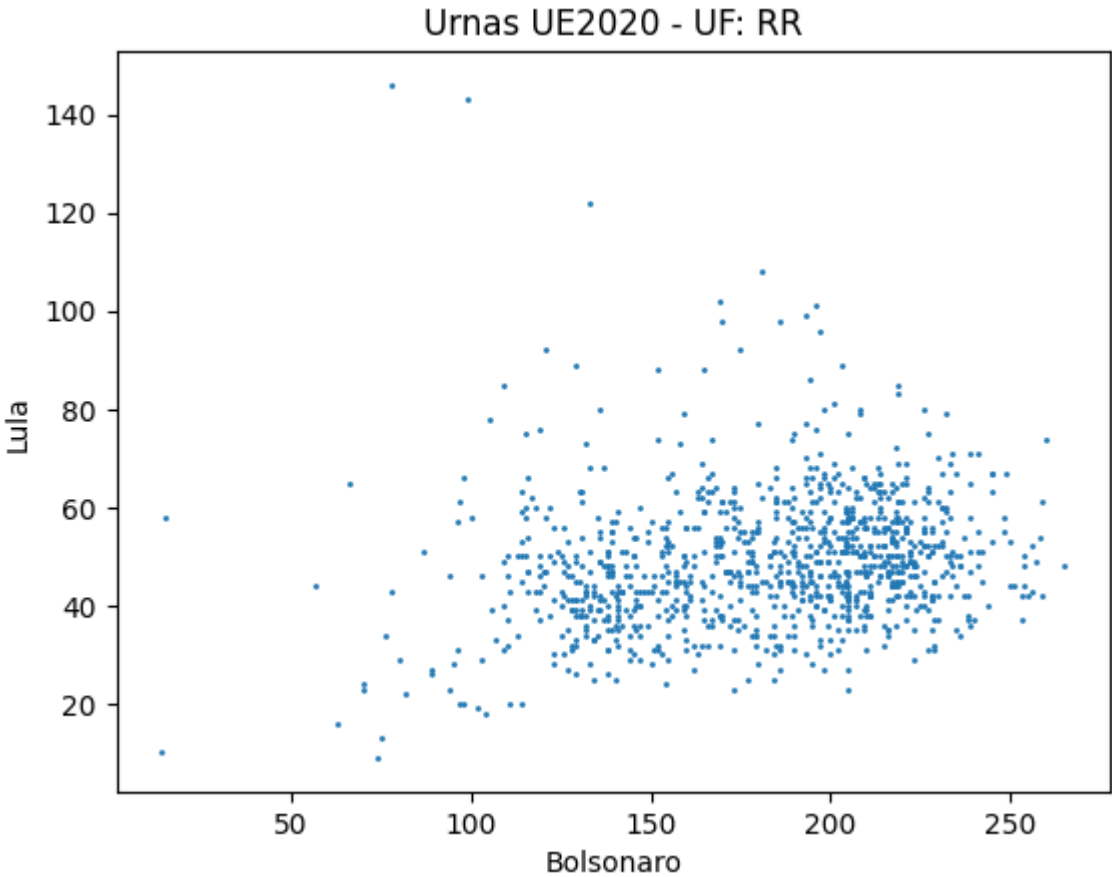
Urnas UE2020 - UF: RO



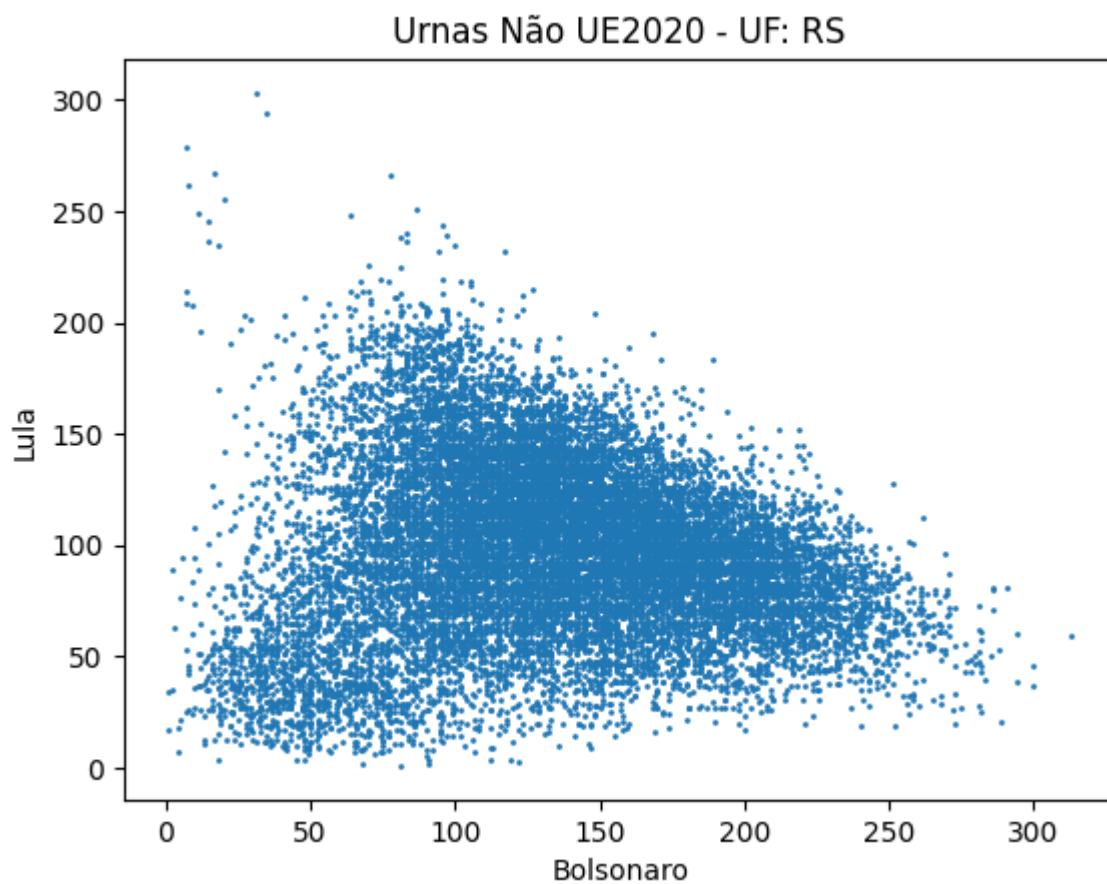
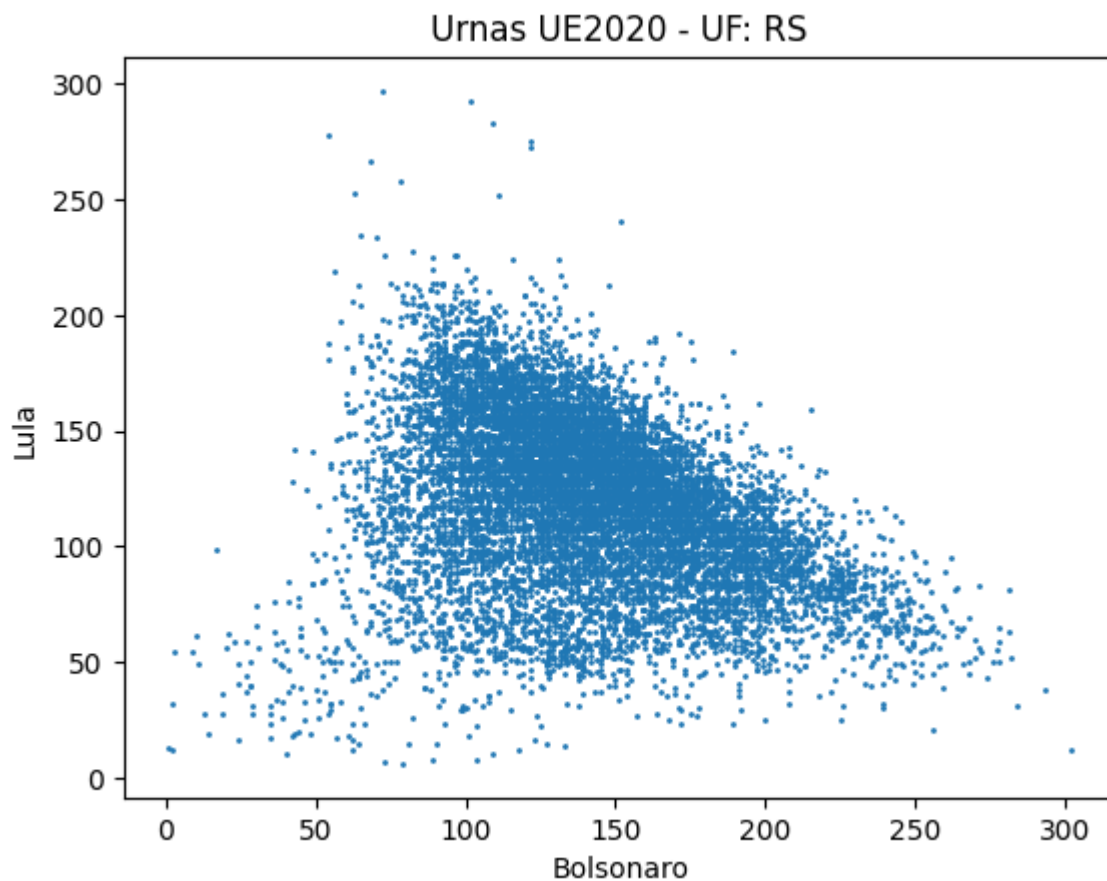
Urnas Não UE2020 - UF: RO



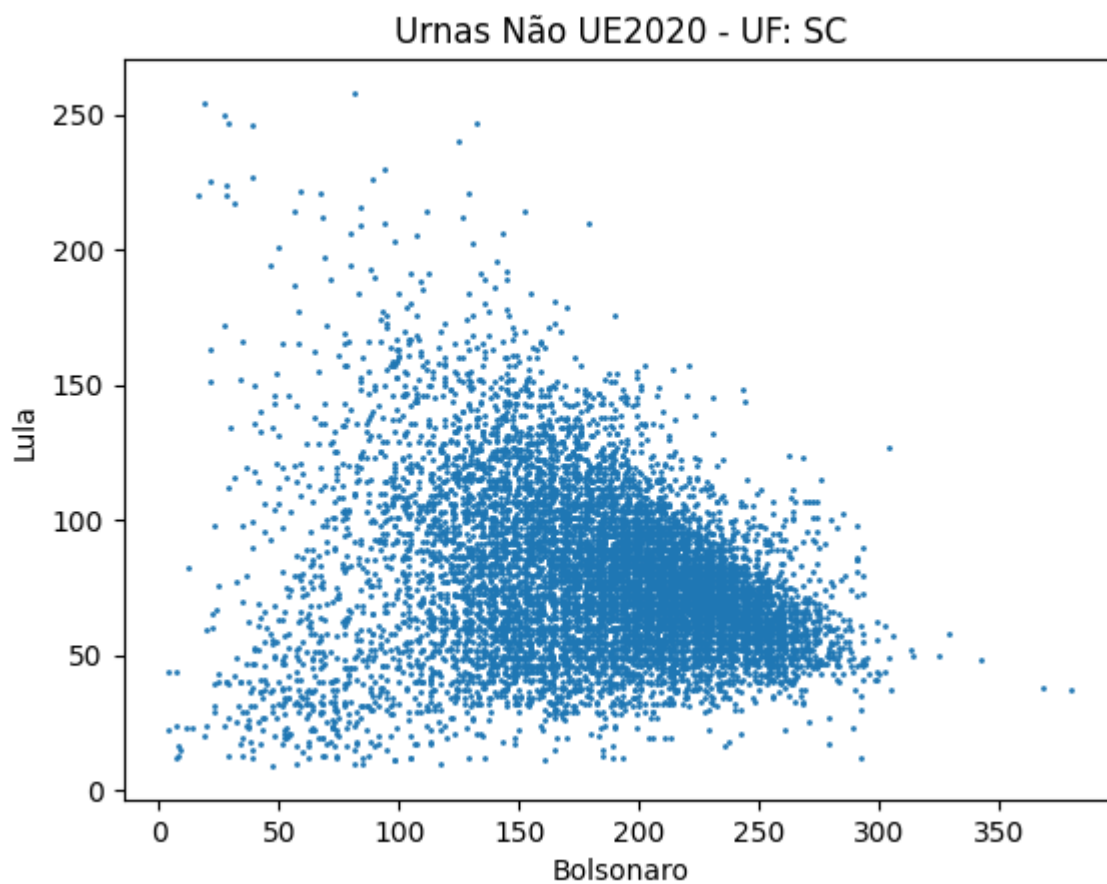
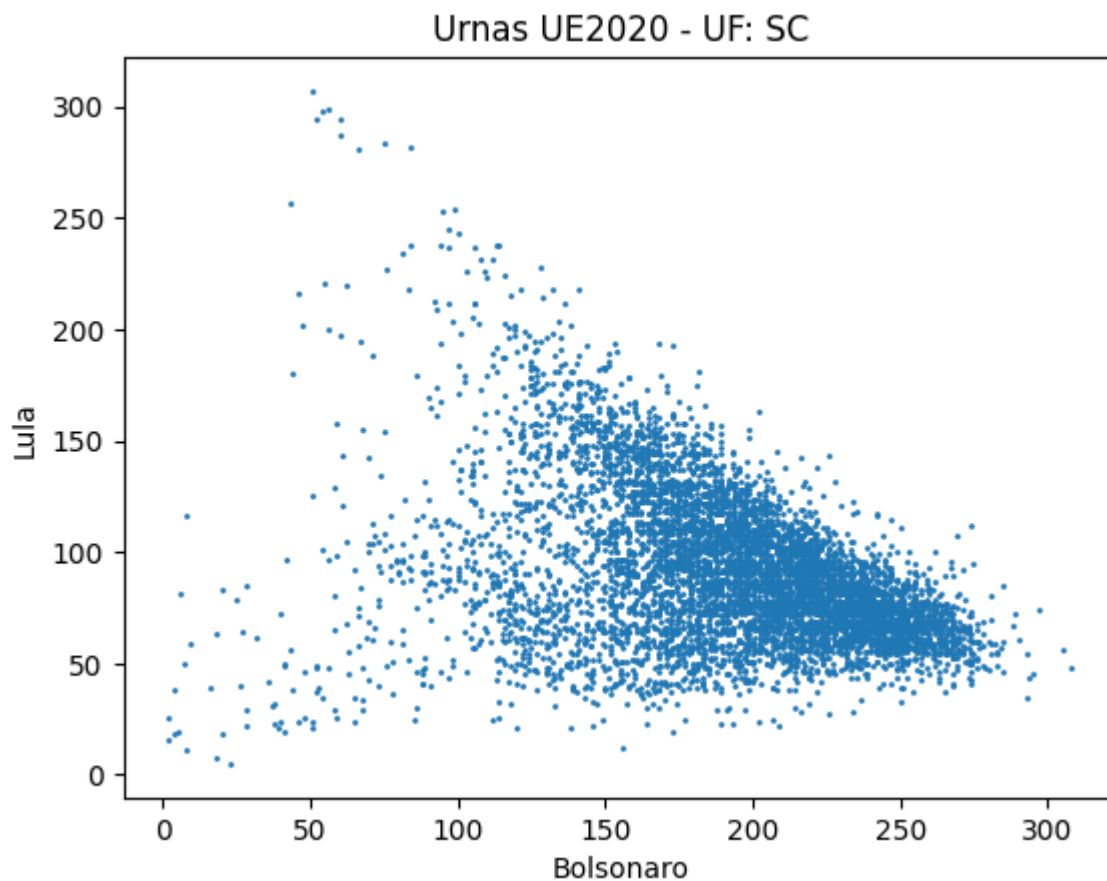
RR



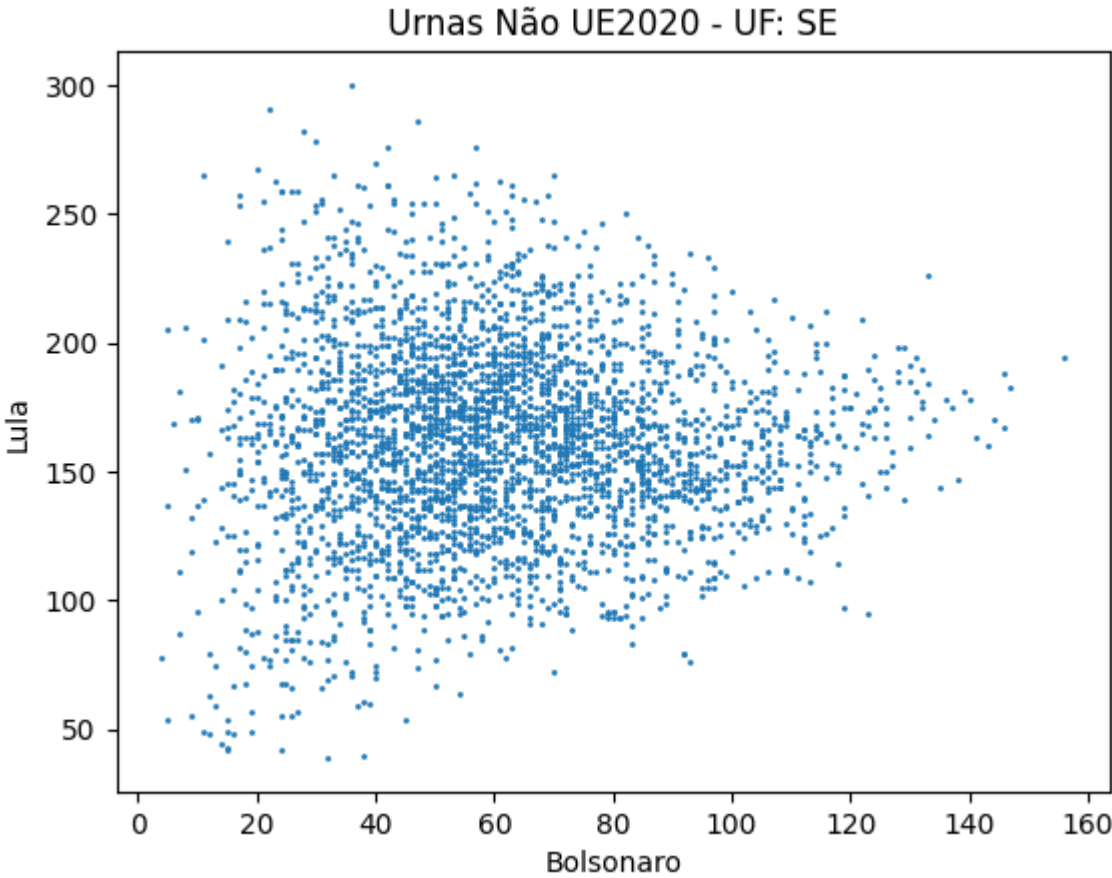
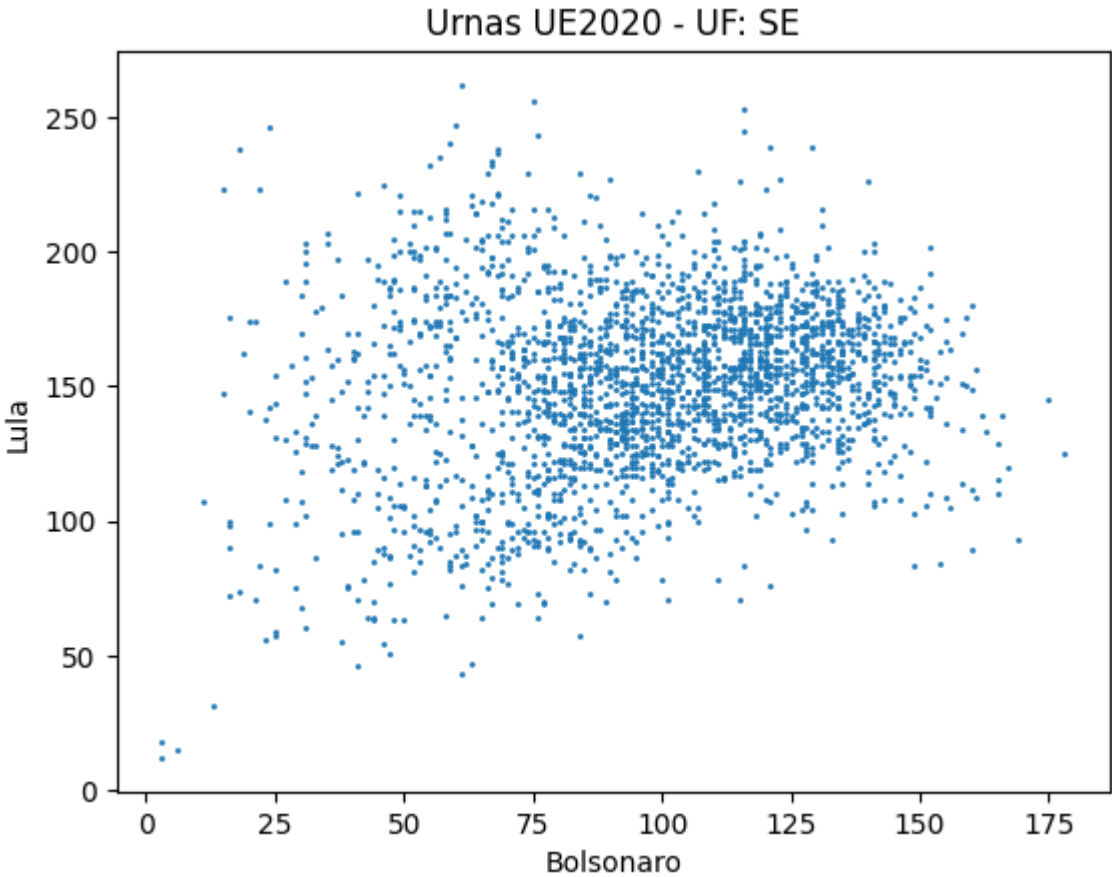
RS



SC

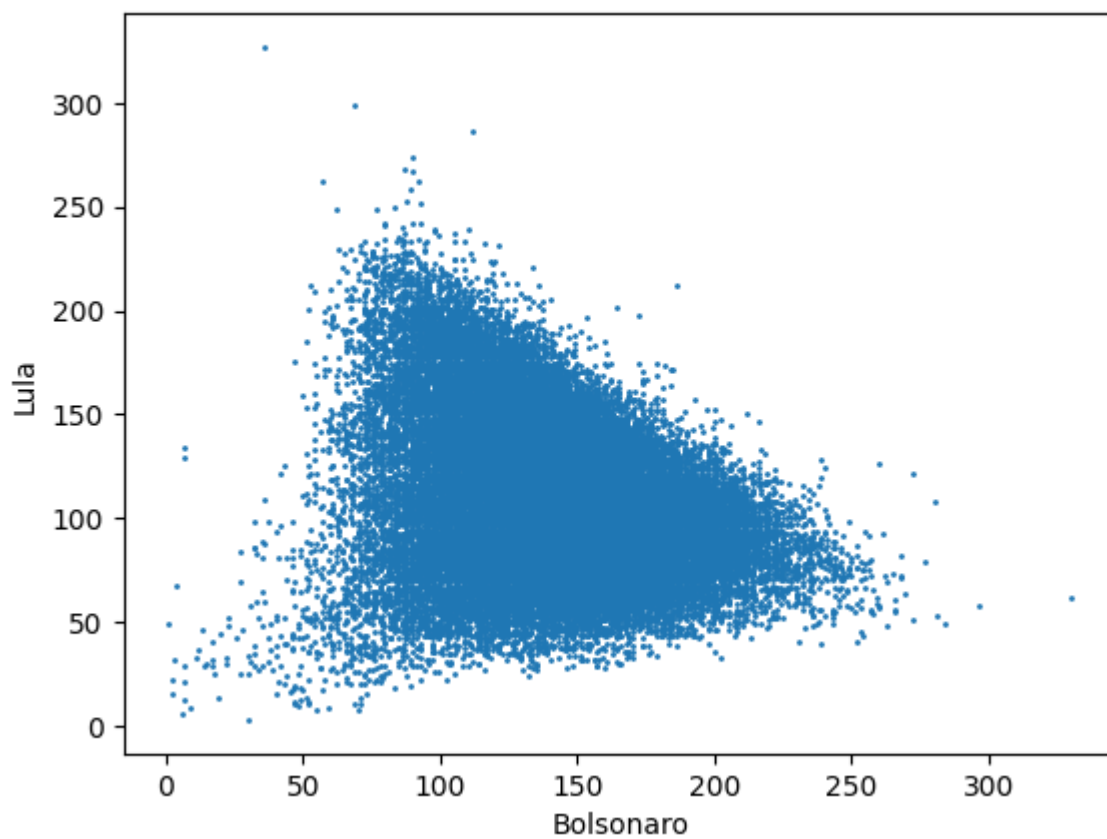


SE

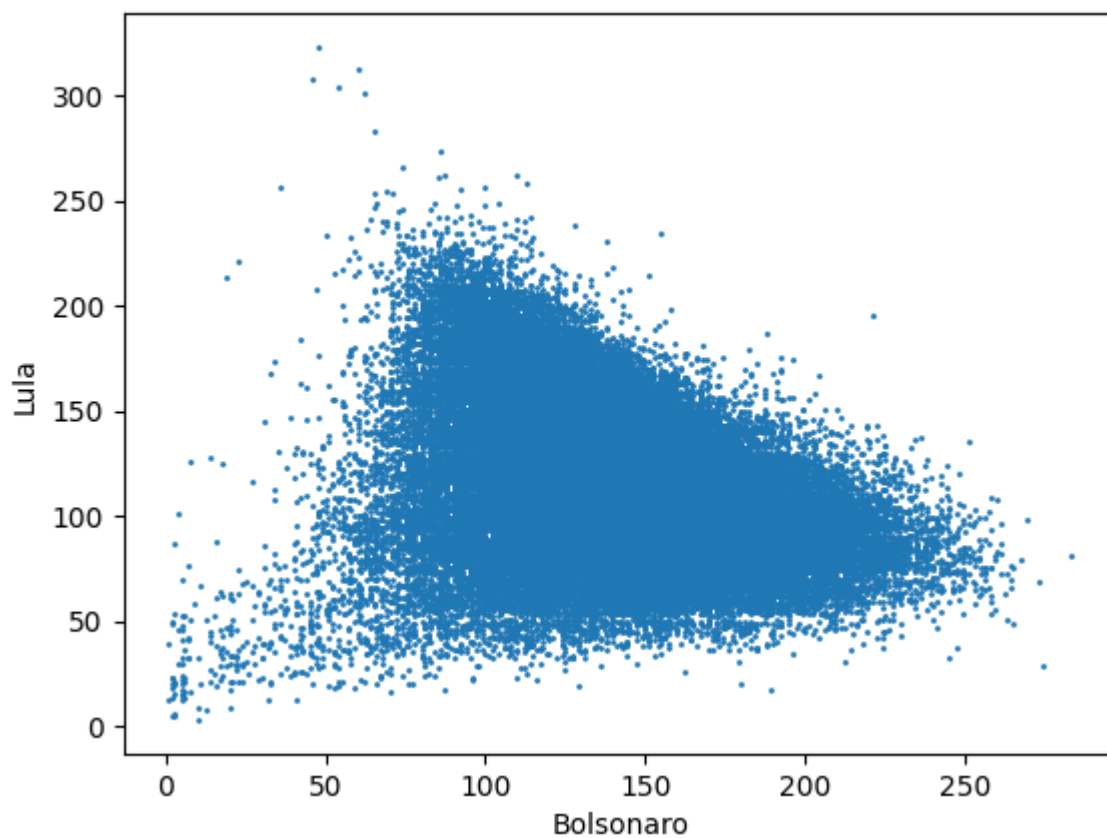


SP

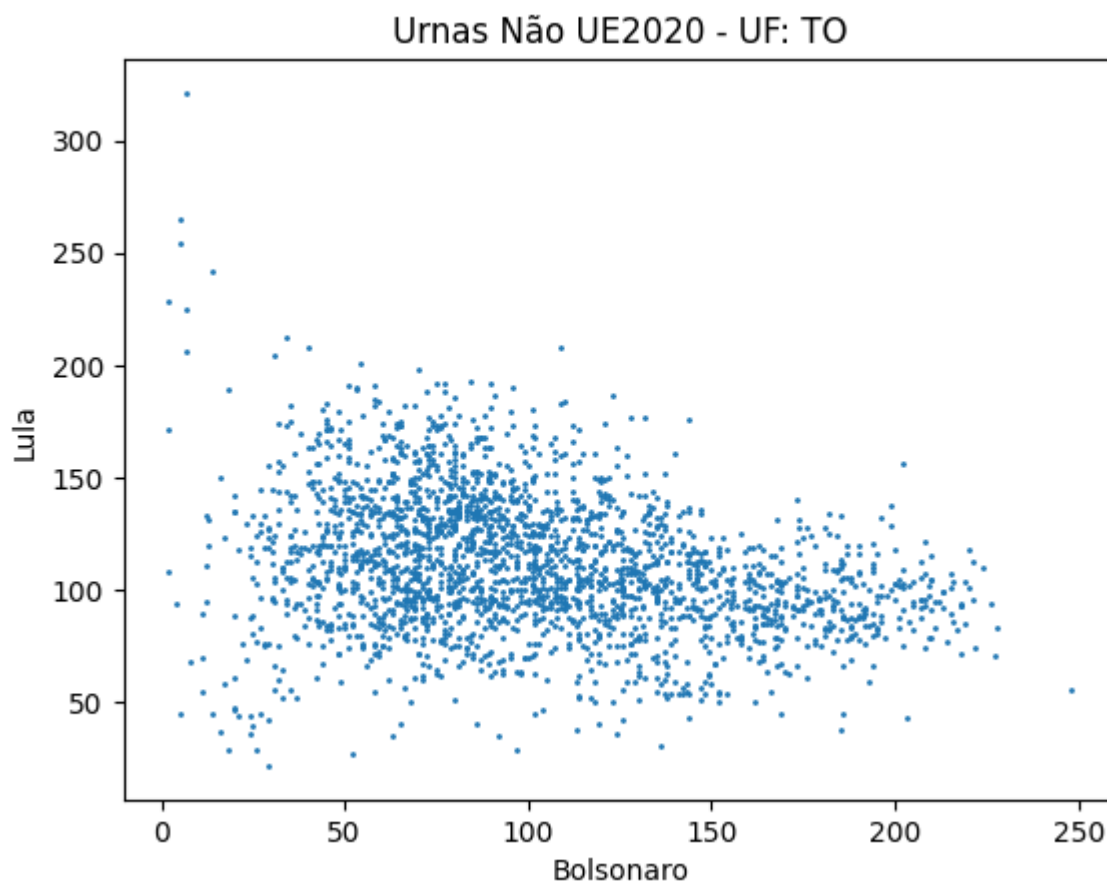
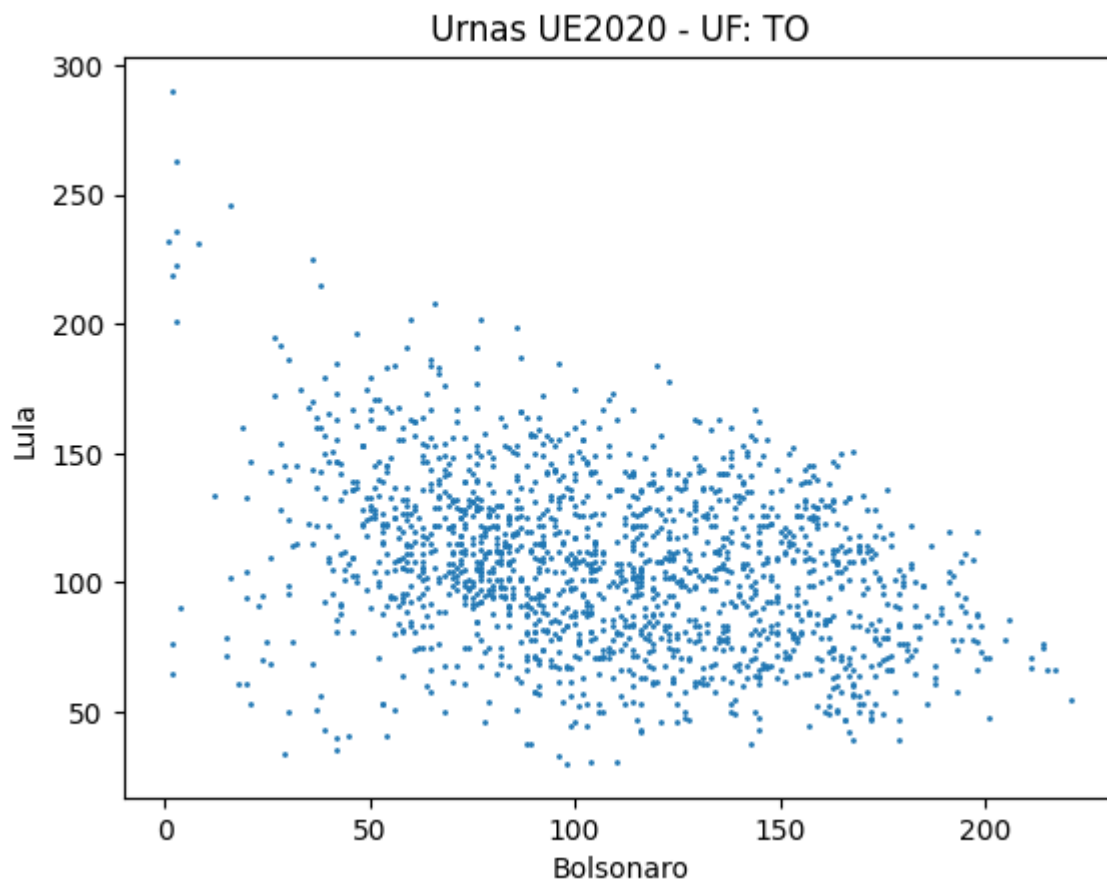
Urnas UE2020 - UF: SP



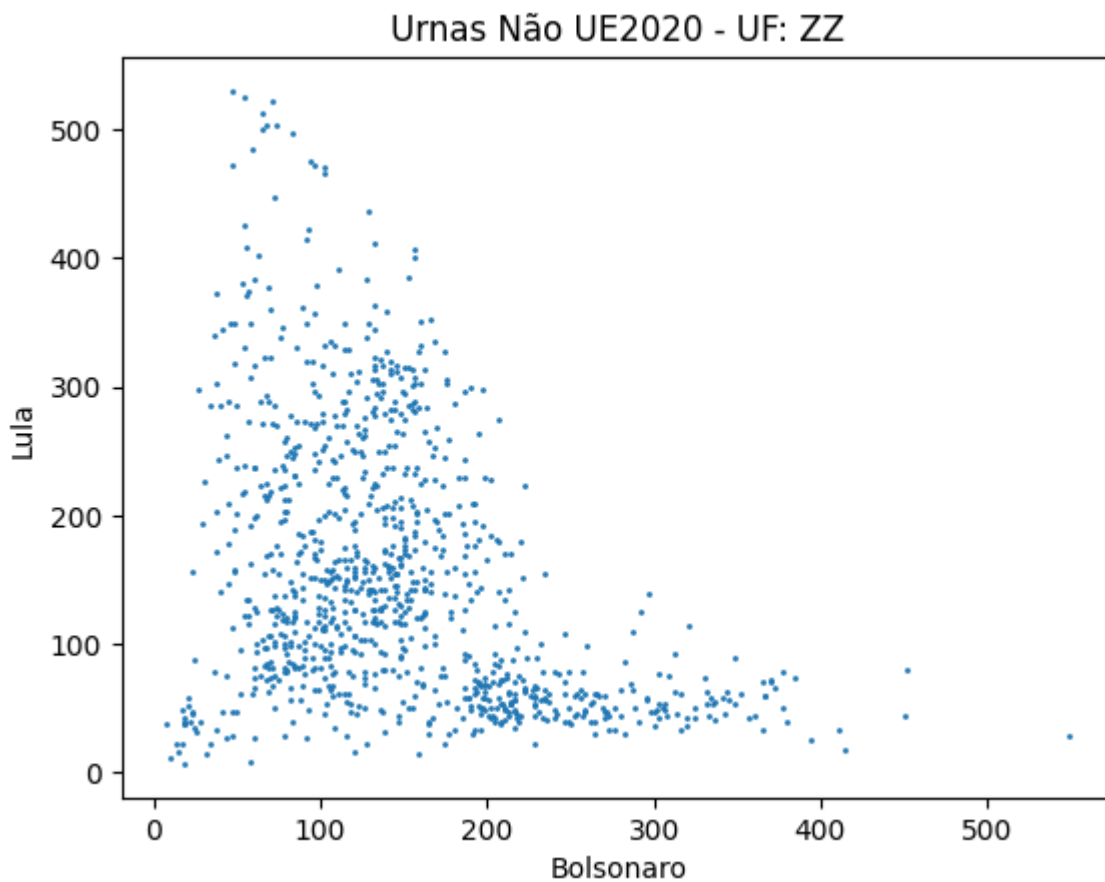
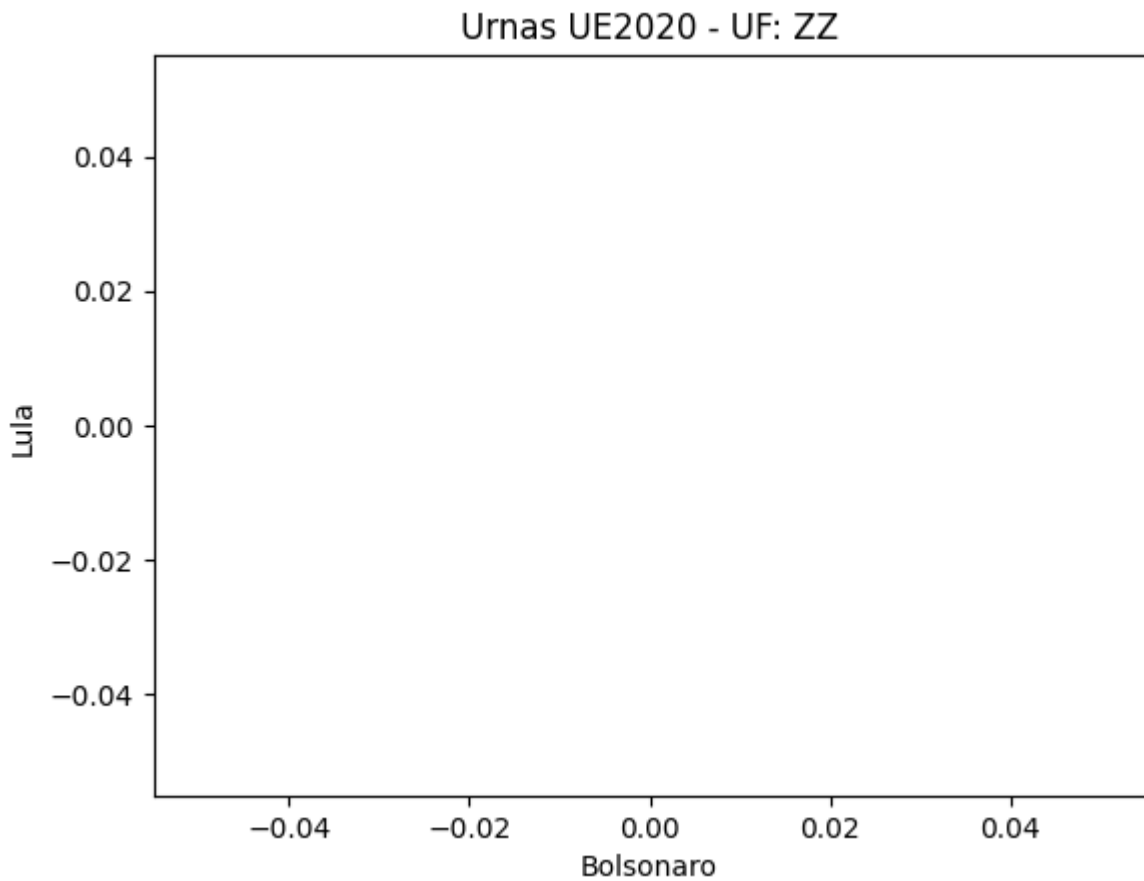
Urnas Não UE2020 - UF: SP



TO



ZZ



Você pode notar nos gráficos acima o achatamento, que fica sempre mais evidente nos estados onde o PT e Lula são mais populares. Nesses estados o achatamento do triângulo fica muito

mais evidente.

Isso demonstra que o algoritmo deve ter alguma lógica, não só aleatória, mas também inteligente para realizar a migração de votos (fraude) levando em consideração a quantidade de votos de cada candidato adversário do número 13 (número do PT)

É muito provável que esse algoritmo esteja presente desde quando as urnas começaram a operar, o que pode ter favorecido o candidato a presidente do PT desde então.

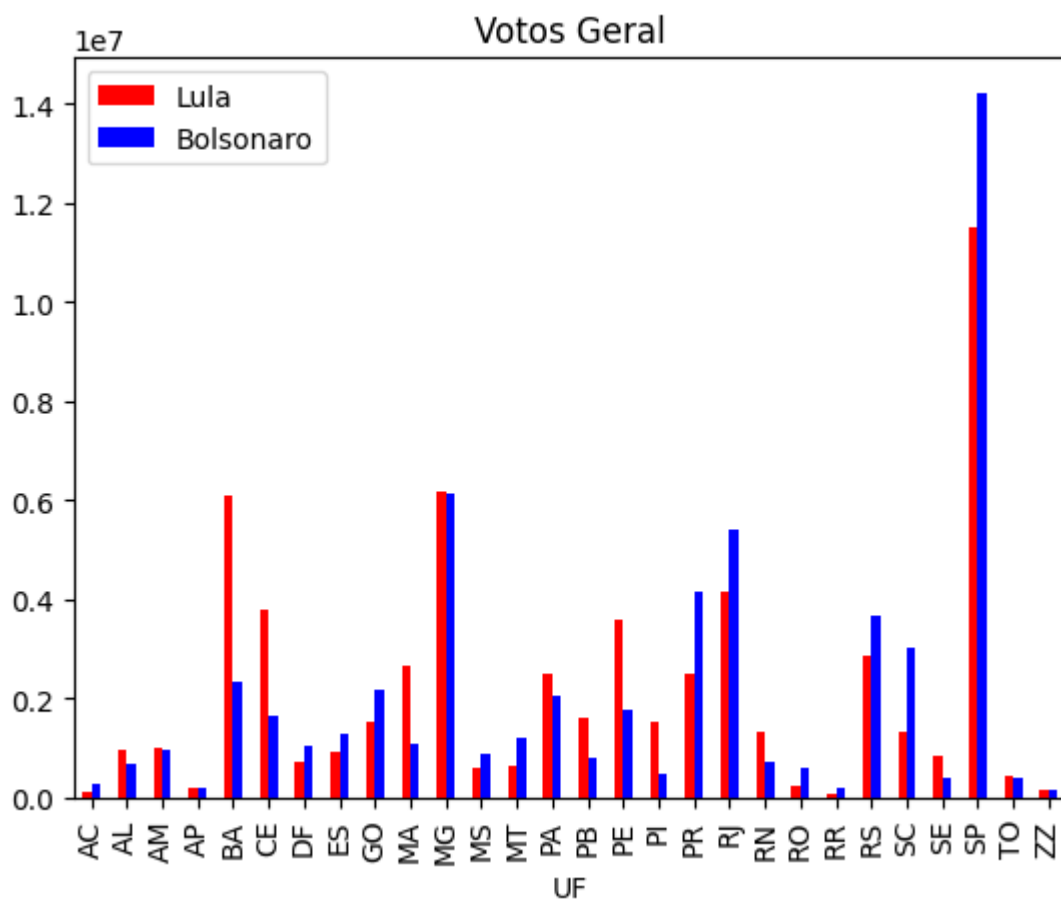
A fraude só pode ser detectada porque as UE2020 não apresentaram o mesmo comportamento, ou o algoritmo falhou nessas urnas, criando a oportunidade de comparar os dados.

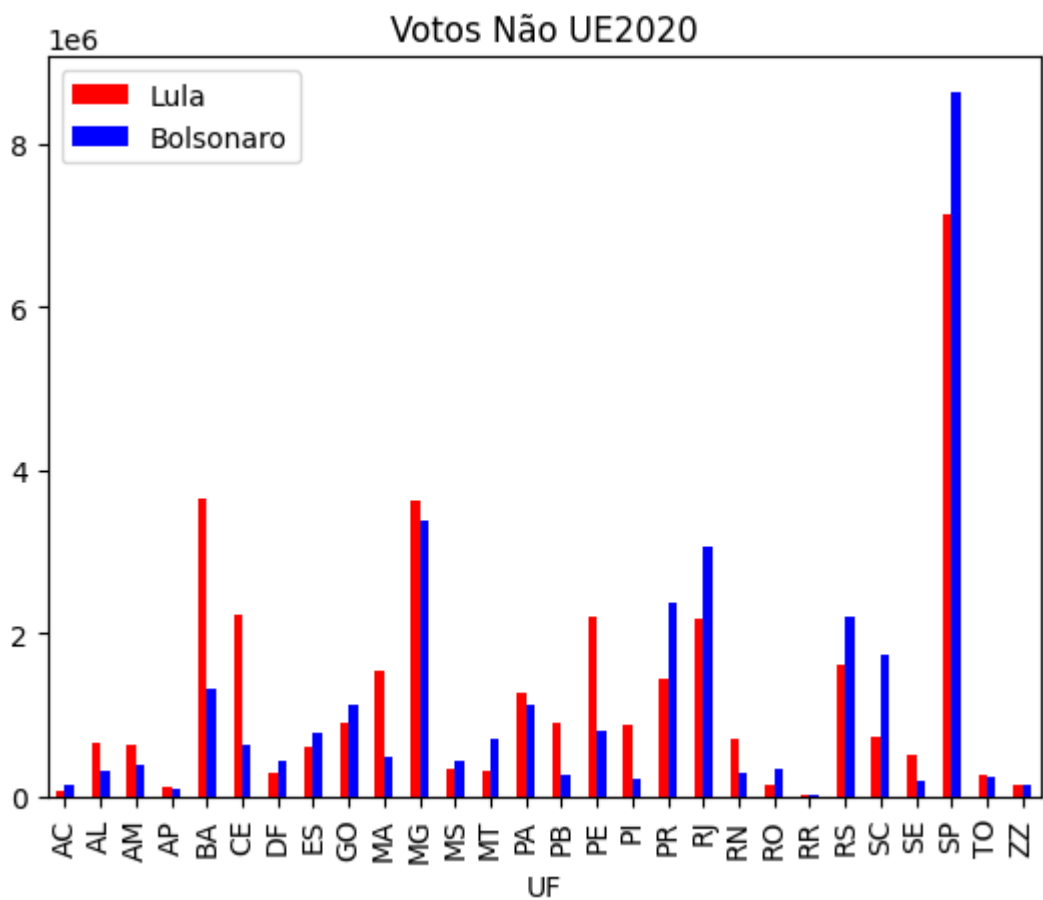
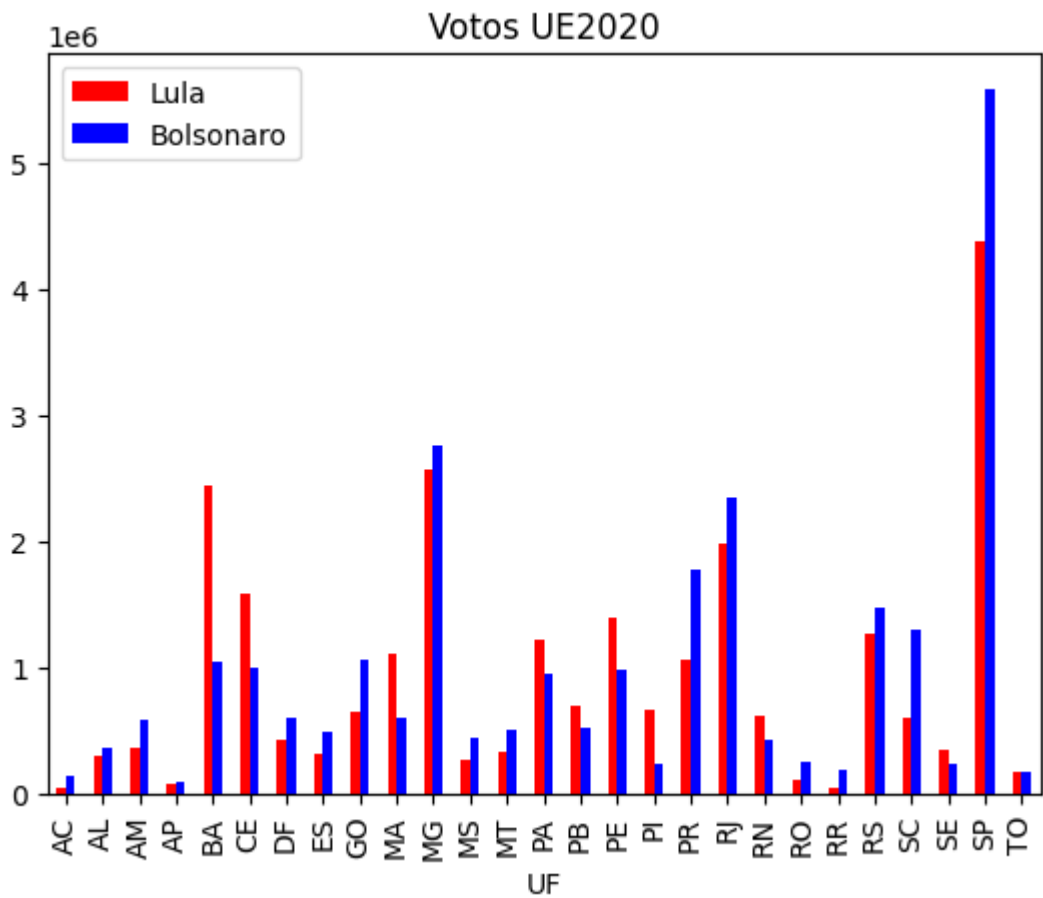
Como será que isso pode refletir nos resultados?

Vamos analisar isso visualizando os resultados por UF e por tipo de urna

```
In [ ]: votosPresidente[['UF', 'Lula', 'Bolsonaro']].groupby(['UF']).sum(numeric_only=True).plot
votosPresidente.query("modelo == 'UE2020')[['UF', 'Lula', 'Bolsonaro']].groupby(['UF'])
votosPresidente.query("modelo != 'UE2020')[['UF', 'Lula', 'Bolsonaro']].groupby(['UF'])

Out[ ]: Text(0.5, 1.0, 'Votos Não UE2020')
```





```
In [ ]: votosPresidente[['UF','modelo','Lula','Bolsonaro']].groupby(['modelo']).sum(numeric_or
```


Out[]: Text(0.5, 1.0, 'Votos por Tipo de Urna')

