

Mapa do resultado do 2º turno das eleições Presidenciais de 2018 por município ¶

Para o Jornal Ibirá News

```
In [1]: # Importando as bibliotecas
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
from descartes import PolygonPatch
import geopandas as gpd
```

```
In [2]: # Carregando o shapefile
df_mapa = gpd.read_file('mapa_eleição\BRMUE250GC_SIR.shp')
```

```
In [3]: # Verificando o dataframe do geopandas
df_mapa.head()
```

Out[3]:

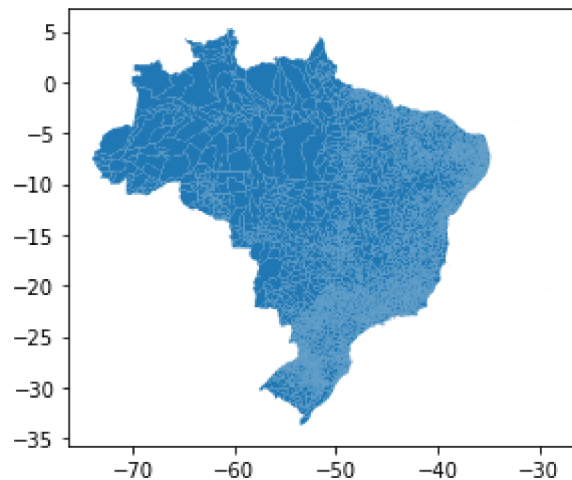
	NM_MUNICIP	CD_GEOCMU	geometry
0	CAIUÃ	3509106	POLYGON ((-51.86001 -21.58072, -51.85892 -21.5...
1	CASTILHO	3511003	POLYGON ((-51.41936 -20.68108, -51.41942 -20.6...
2	DRACENA	3514403	POLYGON ((-51.48515 -21.42765, -51.48500 -21.4...
3	ESTRELA DO NORTE	3515301	POLYGON ((-51.57097 -22.39887, -51.57099 -22.3...
4	EUCLIDES DA CUNHA PAULISTA	3515350	POLYGON ((-52.44569 -22.60828, -52.44569 -22.6...

```
In [4]: # Verificando tamanho e colunas  
df_mapa.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>  
RangeIndex: 5572 entries, 0 to 5571  
Data columns (total 3 columns):  
NM_MUNICIP      5572 non-null object  
CD_GEOCMU       5572 non-null object  
geometry        5572 non-null geometry  
dtypes: geometry(1), object(2)  
memory usage: 130.7+ KB
```

```
In [5]: # Vamos exibir o mapa (mostrar o shapefile)  
df_mapa.plot()
```

```
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x1f53d06a9c8>
```



```
In [6]: # Vamos expandir o espaço para aumentar a figura
fig, ax = plt.subplots(1, figsize=(12, 12))

# desenha o mapa
df_mapa.plot(ax=ax)

# tirar os eixos
ax.set_axis_off()

# mostrar o mapa
plt.show()
```



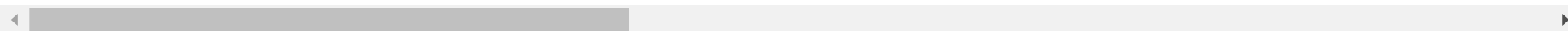
```
In [7]: # Vamos carregar o arquivo da eleição do Brasil dividido por municipio/zona eleitoral
df_votacao = pd.read_csv('mapa_eleição/votacao_candidato_munzona_2018_BR.csv', sep=';', encoding='latin1')

df_votacao.head()
```

Out[7]:

	DT_GERACAO	HH_GERACAO	ANO_ELEICAO	CD_TIPO_ELEICAO	NM_TIPO_ELEICAO	NR_TURNO	CD_ELEICAO	DS_ELEICAO	DT_ELEICAO
0	01/12/2019	18:42:56	2018	2	Eleição Ordinária	1	295	ELEIÇÃO GERAL FEDERAL 2018	07/10/2018
1	01/12/2019	18:42:56	2018	2	Eleição Ordinária	1	295	ELEIÇÃO GERAL FEDERAL 2018	07/10/2018
2	01/12/2019	18:42:56	2018	2	Eleição Ordinária	1	295	ELEIÇÃO GERAL FEDERAL 2018	07/10/2018
3	01/12/2019	18:42:56	2018	2	Eleição Ordinária	1	295	ELEIÇÃO GERAL FEDERAL 2018	07/10/2018
4	01/12/2019	18:42:56	2018	2	Eleição Ordinária	1	295	ELEIÇÃO GERAL FEDERAL 2018	07/10/2018

5 rows × 38 columns



```
In [8]: # Visualizando o dataframe transporsto  
df_votacao.head().T
```

Out[8]:

	0	1	2	3	4
DT_GERACAO	01/12/2019	01/12/2019	01/12/2019	01/12/2019	01/12/2019
HH_GERACAO	18:42:56	18:42:56	18:42:56	18:42:56	18:42:56
ANO_ELEICAO	2018	2018	2018	2018	2018
CD_TIPO_ELEICAO	2	2	2	2	2
NM_TIPO_ELEICAO	Eleição Ordinária	Eleição Ordinária	Eleição Ordinária	Eleição Ordinária	Eleição Ordinária
NR_TURNO	1	1	1	1	1
CD_ELEICAO	295	295	295	295	295
DS_ELEICAO	ELEIÇÃO GERAL FEDERAL 2018	ELEIÇÃO GERAL FEDERAL 2018	ELEIÇÃO GERAL FEDERAL 2018	ELEIÇÃO GERAL FEDERAL 2018	ELEIÇÃO GERAL FEDERAL 2018
DT_ELEICAO	07/10/2018	07/10/2018	07/10/2018	07/10/2018	07/10/2018
TP_ABRANGENCIA	F	F	F	F	F
SG_UF	MG	RS	MG	MA	PA
SG_UE	BR	BR	BR	BR	BR
NM_UE	BRASIL	BRASIL	BRASIL	BRASIL	BRASIL
CD_MUNICIPIO	53198	85537	52213	7471	4715
NM_MUNICIPIO	SENHORA DE OLIVEIRA	BUTIÁ	SÃO FRANCISCO	CAJARI	ITAITUBA
NR_ZONA	217	116	252	20	34
CD_CARGO	1	1	1	1	1
DS_CARGO	Presidente	Presidente	Presidente	Presidente	Presidente
SQ_CANDIDATO	280000601016	280000601016	280000601016	280000622281	280000607640
NR_CANDIDATO	50	50	50	15	30
NM_CANDIDATO	GUILHERME CASTRO BOULOS	GUILHERME CASTRO BOULOS	GUILHERME CASTRO BOULOS	HENRIQUE DE CAMPOS MEIRELLES	JOÃO DIONISIO FILGUEIRA BARRETO AMOEDO
NM_URNA_CANDIDATO	GUILHERME BOULOS	GUILHERME BOULOS	GUILHERME BOULOS	HENRIQUE MEIRELLES	JOÃO AMOÊDO
NM_SOCIAL_CANDIDATO	#NULO#	#NULO#	#NULO#	#NULO#	#NULO#

	0	1	2	3	4
CD_SITUACAO_CANDIDATURA	12	12	12	12	12
DS_SITUACAO_CANDIDATURA	APTO	APTO	APTO	APTO	APTO
CD_DETALHE_SITUACAO_CAND	2	2	2	2	2
DS_DETALHE_SITUACAO_CAND	DEFERIDO	DEFERIDO	DEFERIDO	DEFERIDO	DEFERIDO
TP_AGREMIACAO	Coligação	Coligação	Coligação	Coligação	Partido isolado
NR_PARTIDO	50	50	50	15	30
SG_PARTIDO	PSOL	PSOL	PSOL	MDB	NOVO
NM_PARTIDO	Partido Socialismo e Liberdade	Partido Socialismo e Liberdade	Partido Socialismo e Liberdade	Movimento Democrático Brasileiro	Partido Novo
SQ_COLIGACAO	280000050035	280000050035	280000050035	280000050592	280000050199
NM_COLIGACAO	VAMOS SEM MEDO DE MUDAR O BRASIL	VAMOS SEM MEDO DE MUDAR O BRASIL	VAMOS SEM MEDO DE MUDAR O BRASIL	ESSA É A SOLUÇÃO	PARTIDO ISOLADO
DS_COMPOSICAO_COLIGACAO	PSOL / PCB	PSOL / PCB	PSOL / PCB	MDB / PHS	NOVO
CD_SIT_TOT_TURNO	4	4	4	4	4
DS_SIT_TOT_TURNO	NÃO ELEITO	NÃO ELEITO	NÃO ELEITO	NÃO ELEITO	NÃO ELEITO
ST_VOTO_EM_TRANSITO	N	N	N	N	N
QT_VOTOS_NOMINAIS	17	52	89	151	149


```
In [9]: # Separar os dados pois precisamos apenas os dados do 2º turno
df_votacao = df_votacao[df_votacao['NR_TURNO']==2]

df_votacao.sample(3).T
```

Out[9]:

	24632	6709	61363
DT_GERACAO	01/12/2019	01/12/2019	01/12/2019
HH_GERACAO	18:42:56	18:42:56	18:42:56
ANO_ELEICAO	2018	2018	2018
CD_TIPO_ELEICAO	2	2	2
NM_TIPO_ELEICAO	Eleição Ordinária	Eleição Ordinária	Eleição Ordinária
NR_TURNO	2	2	2
CD_ELEICAO	296	296	296
DS_ELEICAO	ELEIÇÃO GERAL FEDERAL 2018	ELEIÇÃO GERAL FEDERAL 2018	ELEIÇÃO GERAL FEDERAL 2018
DT_ELEICAO	28/10/2018	28/10/2018	28/10/2018
TP_ABRANGENCIA	F	F	F
SG_UF	MG	RS	MT
SG_UE	BR	BR	BR
NM_UE	BRASIL	BRASIL	BRASIL
CD_MUNICIPIO	49611	88412	90247
NM_MUNICIPIO	PATROCÍNIO	SANTA MARIA	SÃO JOSÉ DO XINGU
NR_ZONA	211	41	28
CD_CARGO	1	1	1
DS_CARGO	Presidente	Presidente	Presidente
SQ_CANDIDATO	280000629808	280000629808	280000614517
NR_CANDIDATO	13	13	17
NM_CANDIDATO	FERNANDO HADDAD	FERNANDO HADDAD	JAIR MESSIAS BOLSONARO
NM_URNA_CANDIDATO	FERNANDO HADDAD	FERNANDO HADDAD	JAIR BOLSONARO
NM_SOCIAL_CANDIDATO	#NULO#	#NULO#	#NULO#
CD_SITUACAO_CANDIDATURA	12	12	12
DS_SITUACAO_CANDIDATURA	APTO	APTO	APTO

	24632	6709	61363
CD_DETALHE_SITUACAO_CAND	2	2	2
DS_DETALHE_SITUACAO_CAND	DEFERIDO	DEFERIDO	DEFERIDO
TP_AGREMIACAO	Coligação	Coligação	Coligação
NR_PARTIDO	13	13	17
SG_PARTIDO	PT	PT	PSL
NM_PARTIDO	Partido dos Trabalhadores	Partido dos Trabalhadores	Partido Social Liberal
SQ_COLIGACAO	280000050703	280000050703	280000050386
NM_COLIGACAO	O POVO FELIZ DE NOVO	O POVO FELIZ DE NOVO	BRASIL ACIMA DE TUDO, DEUS ACIMA DE TODOS
DS_COMPOSICAO_COLIGACAO	PT / PC do B / PROS	PT / PC do B / PROS	PSL / PRTB
CD_SIT_TOT_TURNO	4	4	1
DS_SIT_TOT_TURNO	NÃO ELEITO	NÃO ELEITO	ELEITO
ST_VOTO_EM_TRANSITO	N	N	N
QT_VOTOS_NOMINAIS	12725	29320	1314

```
In [10]: # Vamos verificar quantos municipios existem no conjunto de dados
# Vamos pegar por código de municipio
df_votacao['CD_MUNICIPIO'].nunique()
```

Out[10]: 5741

```
In [11]: # Mas o Brasil tem 5570 municipios
#Vamos verificar algumas cidades fora do Brasil
df_votacao[df_votacao['SG_UF']== 'ZZ'].sample(4).T
```

Out[11]:

	87980	16362	10469	24140
DT_GERACAO	01/12/2019	01/12/2019	01/12/2019	01/12/2019
HH_GERACAO	18:42:56	18:42:56	18:42:56	18:42:56
ANO_ELEICAO	2018	2018	2018	2018
CD_TIPO_ELEICAO	2	2	2	2
NM_TIPO_ELEICAO	Eleição Ordinária	Eleição Ordinária	Eleição Ordinária	Eleição Ordinária
NR_TURNO	2	2	2	2
CD_ELEICAO	296	296	296	296
DS_ELEICAO	ELEIÇÃO GERAL FEDERAL 2018	ELEIÇÃO GERAL FEDERAL 2018	ELEIÇÃO GERAL FEDERAL 2018	ELEIÇÃO GERAL FEDERAL 2018
DT_ELEICAO	28/10/2018	28/10/2018	28/10/2018	28/10/2018
TP_ABRANGENCIA	F	F	F	F
SG_UF	ZZ	ZZ	ZZ	ZZ
SG_UE	BR	BR	BR	BR
NM_UE	BRASIL	BRASIL	BRASIL	BRASIL
CD_MUNICIPIO	99430	38962	30155	30104
NM_MUNICIPIO	KINGSTON-JAMAICA	DAR ES SALAAM	MONTREAL	MÉXICO-MEXI
NR_ZONA	1	1	1	1
CD_CARGO	1	1	1	1
DS_CARGO	Presidente	Presidente	Presidente	Presidente
SQ_CANDIDATO	280000629808	280000614517	280000629808	280000629808
NR_CANDIDATO	13	17	13	13
NM_CANDIDATO	FERNANDO HADDAD	JAIR MESSIAS BOLSONARO	FERNANDO HADDAD	FERNANDO HADDAD
NM_URNA_CANDIDATO	FERNANDO HADDAD	JAIR BOLSONARO	FERNANDO HADDAD	FERNANDO HADDAD
NM_SOCIAL_CANDIDATO	#NULO#	#NULO#	#NULO#	#NULO#
CD_SITUACAO_CANDIDATURA	12	12	12	12
DS_SITUACAO_CANDIDATURA	APTO	APTO	APTO	APTO

	87980	16362	10469	24140
CD_DETALHE_SITUACAO_CAND	2	2	2	2
DS_DETALHE_SITUACAO_CAND	DEFERIDO	DEFERIDO	DEFERIDO	DEFERIDO
TP_AGREMIACAO	Coligação	Coligação	Coligação	Coligação
NR_PARTIDO	13	17	13	13
SG_PARTIDO	PT	PSL	PT	PT
NM_PARTIDO	Partido dos Trabalhadores	Partido Social Liberal	Partido dos Trabalhadores	Partido dos Trabalhadores
SQ_COLIGACAO	280000050703	280000050386	280000050703	280000050703
NM_COLIGACAO	O POVO FELIZ DE NOVO	BRASIL ACIMA DE TUDO, DEUS ACIMA DE TODOS	O POVO FELIZ DE NOVO	O POVO FELIZ DE NOVO
DS_COMPOSICAO_COLIGACAO	PT / PC do B / PROS	PSL / PRTB	PT / PC do B / PROS	PT / PC do B / PROS
CD_SIT_TOT_TURN0	4	1	4	4
DS_SIT_TOT_TURN0	NÃO ELEITO	ELEITO	NÃO ELEITO	NÃO ELEITO
ST_VOTO_EM_TRANSITO	N	N	N	N
QT_VOTOS_NOMINAIS	3	6	1759	305

```
In [12]: # Quantas cidades fora do Brasil estão no conjunto de dados
df_votacao[df_votacao['SG_UF']== 'ZZ']['CD_MUNICIPIO'].nunique()
```

Out[12]: 171

```
In [13]: # Vamos eliminar os municipios fora do Brasil
df_votacao = df_votacao[df_votacao['SG_UF'] != 'ZZ']
```

```
In [14]: # Verificando a quantiade de municipios
df_votacao['CD_MUNICIPIO'].nunique()
```

Out[14]: 5570

```
In [15]: # Os dados estão separados por municipio/zona
# Municipios pequenos normalmente tem uma zona eleitoral, mas municipios maiores podem ter mais de uma zona eleitoral

# Vamos verificar Belo Horizonte
df_votacao[df_votacao['NM_MUNICIPIO']=='BELO HORIZONTE']
```

Out[15]:

	DT_GERACAO	HH_GERACAO	ANO_ELEICAO	CD_TIPO_ELEICAO	NM_TIPO_ELEICAO	NR_TURNO	CD_ELEICAO	DS_ELEICAO	DT_ELEIC
232	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
9859	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
9939	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
10856	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
15161	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
16945	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
23542	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
25287	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
27036	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2

	DT_GERACAO	HH_GERACAO	ANO_ELEICAO	CD_TIPO_ELEICAO	NM_TIPO_ELEICAO	NR_TURNO	CD_ELEICAO	DS_ELEICAO	DT_ELEIC
28262	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
32950	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
33912	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
34848	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
36782	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
39361	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
39624	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
48204	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
48574	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
50396	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2

	DT_GERACAO	HH_GERACAO	ANO_ELEICAO	CD_TIPO_ELEICAO	NM_TIPO_ELEICAO	NR_TURNO	CD_ELEICAO	DS_ELEICAO	DT_ELEIC
53527	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
54202	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
55167	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
55656	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
58659	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
64864	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
71158	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
74282	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
75599	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2
75912	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2

	DT_GERACAO	HH_GERACAO	ANO_ELEICAO	CD_TIPO_ELEICAO	NM_TIPO_ELEICAO	NR_TURNO	CD_ELEICAO	DS_ELEICAO	DT_ELEICAO
82882	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2018
82999	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2018
84843	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2018
89655	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2018
91049	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2018
91354	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2018
93695	01/12/2019	18:42:56	2018	2	Eleição Ordinária	2	296	ELEIÇÃO GERAL FEDERAL 2018	28/10/2018

36 rows × 38 columns



```
In [16]: # Eliminar algumas colunas do dataframe que não serão utilizadas
df_votacao = df_votacao[['SG_UF', 'CD_MUNICIPIO', 'NM_MUNICIPIO', 'NR_CANDIDATO', 'NM_URNA_CANDIDATO', 'QT_VOTOS_NOMINAIS']]

df_votacao.head()
```

Out[16]:

	SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOTOS_NOMINAIS
54	PA	5274	SANTA CRUZ DO ARARI	17	JAIR BOLSONARO	821
55	MG	53350	SERRA DO SALITRE	17	JAIR BOLSONARO	3559
58	MG	40290	ALÉM PARAÍBA	17	JAIR BOLSONARO	9512
59	MG	44458	DIVINÓPOLIS	17	JAIR BOLSONARO	30438
60	BA	36099	ITAJU DO COLÔNIA	17	JAIR BOLSONARO	1398

```
In [20]: # Precisamos agregar os dados por municipio e candidato
df_votacao.groupby(['SG_UF', 'CD_MUNICIPIO', 'NM_MUNICIPIO', 'NR_CANDIDATO', 'NM_URNA_CANDIDATO']).sum()
```

Out[20]:

					QT_VOTOS_NOMINAIS
SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	
AC	1007	BUJARI	13	FERNANDO HADDAD	1274
			17	JAIR BOLSONARO	4676
	1015	CAPIXABA	13	FERNANDO HADDAD	1086
			17	JAIR BOLSONARO	3895
	1023	PORTO ACRE	13	FERNANDO HADDAD	2059

TO	96954	ALIANÇA DO TOCANTINS	17	JAIR BOLSONARO	1215
	96970	APARECIDA DO RIO NEGRO	13	FERNANDO HADDAD	1291
			17	JAIR BOLSONARO	1171
			13	FERNANDO HADDAD	1545
	96997	BARROLÂNDIA	17	JAIR BOLSONARO	1501

11140 rows × 6 columns

```
In [21]: # Fazer o groupby, reindexar o indice, vai salvar isso em um dataframe
df_votacao.groupby(['SG_UF', 'CD_MUNICIPIO', 'NM_MUNICIPIO', 'NR_CANDIDATO', 'NM_URNA_CANDIDATO']).sum().reset_index()
df_votacao.sample(4)
```

Out[21]:

SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOTOS_NOMINAIS	
9405	SC	83852	XAVANTINA	17	JAIR BOLSONARO	1666
6847	PR	76830	MARECHAL CÂNDIDO RONDON	17	JAIR BOLSONARO	23281
2292	MA	7188	BELA VISTA DO MARANHÃO	13	FERNANDO HADDAD	5266
10193	SP	65773	ITOBI	17	JAIR BOLSONARO	2783

```
In [22]: # Vamos verificar novamente a cidade de Belo Horizonte
df_votacao[df_votacao['NM_MUNICIPIO'] == "BELO HORIZONTE"]
```

Out[22]:

	SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOTOS_NOMINAIS
2936	MG	41238	BELO HORIZONTE	13	FERNANDO HADDAD	472887
2937	MG	41238	BELO HORIZONTE	17	JAIR BOLSONARO	901513

```
In [24]: # Ordenar o dataframe pela quantidade de votos e depois deixar apenas o primeiro registro de cada municipio
df_votacao.sort_values(by='QT_VOTOS_NOMINAIS', ascending=False).head(10)
```

Out[24]:

	SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOTOS_NOMINAIS
10723	SP	71072	SÃO PAULO	17	JAIR BOLSONARO	3694834
10722	SP	71072	SÃO PAULO	13	FERNANDO HADDAD	2424125
7369	RJ	60011	RIO DE JANEIRO	17	JAIR BOLSONARO	2179896
7368	RJ	60011	RIO DE JANEIRO	13	FERNANDO HADDAD	1105393
1607	DF	97012	BRASÍLIA	17	JAIR BOLSONARO	1080411
1054	BA	38490	SALVADOR	13	FERNANDO HADDAD	985532
2937	MG	41238	BELO HORIZONTE	17	JAIR BOLSONARO	901513
6659	PR	75353	CURITIBA	17	JAIR BOLSONARO	786377
1376	CE	13897	FORTALEZA	13	FERNANDO HADDAD	739265
309	AM	2550	MANAUS	17	JAIR BOLSONARO	686999

```
In [25]: # Vamos ordenar
df_votacao.sort_values(by='QT_VOTOS_NOMINAIS', ascending=False, inplace=True)

# dropar as linhas duplicadas, mantendo a primeira
df_votacao.drop_duplicates(subset='CD_MUNICIPIO', keep='first', inplace=True)

df_votacao.head(10)
```

Out[25]:

	SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOTOS_NOMINAIS
10723	SP	71072	SÃO PAULO	17	JAIR BOLSONARO	3694834
7369	RJ	60011	RIO DE JANEIRO	17	JAIR BOLSONARO	2179896
1607	DF	97012	BRASÍLIA	17	JAIR BOLSONARO	1080411
1054	BA	38490	SALVADOR	13	FERNANDO HADDAD	985532
2937	MG	41238	BELO HORIZONTE	17	JAIR BOLSONARO	901513
6659	PR	75353	CURITIBA	17	JAIR BOLSONARO	786377
1376	CE	13897	FORTALEZA	13	FERNANDO HADDAD	739265
309	AM	2550	MANAUS	17	JAIR BOLSONARO	686999
2013	GO	93734	GOIÂNIA	17	JAIR BOLSONARO	528525
5834	PE	25313	RECIFE	13	FERNANDO HADDAD	482673

```
In [27]: # Vamos verificar novamente a cidade de Belo Horizonte
df_votacao[df_votacao['NM_MUNICIPIO'] == "BELO HORIZONTE"]
```

Out[27]:

	SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOTOS_NOMINAIS
2937	MG	41238	BELO HORIZONTE	17	JAIR BOLSONARO	901513

```
In [29]: # Agora vamos criar uma coluna representando o resultado da eleição naquele município, sendo que o resultado é True
# se o candidato Bolsonaro venceu naquele município e False em caso contrário
df_votacao['RESULTADO'] = df_votacao['NR_CANDIDATO'].apply(lambda x: x == 17)
```

```
In [30]: # Verificando o resultado
df_votacao.head(10)
```

Out[30]:

	SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOTOS_NOMINAIS	RESULTADO
10723	SP	71072	SÃO PAULO	17	JAIR BOLSONARO	3694834	True
7369	RJ	60011	RIO DE JANEIRO	17	JAIR BOLSONARO	2179896	True
1607	DF	97012	BRASÍLIA	17	JAIR BOLSONARO	1080411	True
1054	BA	38490	SALVADOR	13	FERNANDO HADDAD	985532	False
2937	MG	41238	BELO HORIZONTE	17	JAIR BOLSONARO	901513	True
6659	PR	75353	CURITIBA	17	JAIR BOLSONARO	786377	True
1376	CE	13897	FORTALEZA	13	FERNANDO HADDAD	739265	False
309	AM	2550	MANAUS	17	JAIR BOLSONARO	686999	True
2013	GO	93734	GOIÂNIA	17	JAIR BOLSONARO	528525	True
5834	PE	25313	RECIFE	13	FERNANDO HADDAD	482673	False

```
In [31]: # Mostrar alguns valores do dataframe do geopandas
df_mapa.sample(3)
```

Out[31]:

	NM_MUNICIP	CD_GEOCMU	geometry
3819	ANTÃ NIO MARTINS	2400901	POLYGON ((-37.88950 -6.16687, -37.88724 -6.168...
5285	SÃ O TOMÃ	4126108	POLYGON ((-52.42197 -23.46970, -52.42041 -23.4...
3853	FLORÃ NIA	2403806	POLYGON ((-36.71989 -6.09526, -36.72124 -6.096...

```
In [33]: # Tentando encontrar o municipio pelo código
df_votacao[df_votacao['CD_MUNICIPIO'] == 2400901]
```

Out[33]:

SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOTOS_NOMINAIS	RESULTADO
-------	--------------	--------------	--------------	-------------------	-------------------	-----------


```
In [34]: # Por que não podemos utilizar o nome do municipio para juntar os nomes
df_votacao[df_votacao['NM_MUNICIPIO'] == 'OURO BRANCO']
```

Out[34]:

	SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOTOS_NOMINAIS	RESULTADO
3867	MG	49174	OURO BRANCO	17	JAIR BOLSONARO	12198	True
174	AL	28215	OURO BRANCO	13	FERNANDO HADDAD	5156	False
7570	RN	17698	OURO BRANCO	13	FERNANDO HADDAD	2463	False

```
In [37]: # Recuperar a tabela de correspondência entre IBGE e TSE direto do github do estado
link = 'https://raw.githubusercontent.com/estadao/como-votou-sua-vizinhanca/master/data/votos/correspondencia-tse-ibge.csv'

df_equivalencia = pd.read_csv(link)

df_equivalencia.head()
```

Out[37]:

	chave	NOME	UF	GEOCOD_IBGE	COD_TSE	AJUSTE
0	ac acrelandia	ACRELÂNDIA	AC	1200013	1120	NaN
1	ac assis brasil	ASSIS BRASIL	AC	1200054	1570	NaN
2	ac brasileira	BRASILÉIA	AC	1200104	1058	NaN
3	ac bujari	BUJARI	AC	1200138	1007	NaN
4	ac capixaba	CAPIXABA	AC	1200179	1015	NaN

```
In [39]: # Vamos prepara os dataframes
# Vamos reindexar o dataframe de votação pelo código do TSE
df_vot_novo = df_votacao.copy()

df_vot_novo.set_index('CD_MUNICIPIO', drop=False, inplace=True)

df_vot_novo.head()
```

Out[39]:

	SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOTOS_NOMINAIS	RESULTADO
CD_MUNICIPIO							
71072	SP	71072	SÃO PAULO	17	JAIR BOLSONARO	3694834	True
60011	RJ	60011	RIO DE JANEIRO	17	JAIR BOLSONARO	2179896	True
97012	DF	97012	BRASÍLIA	17	JAIR BOLSONARO	1080411	True
38490	BA	38490	SALVADOR	13	FERNANDO HADDAD	985532	False
41238	MG	41238	BELO HORIZONTE	17	JAIR BOLSONARO	901513	True

```
In [40]: # Vamos também reindexar o dataframe de equivalencia pelo código do TSE
df_equi_novo = df_equivalencia.copy()

df_equi_novo.set_index('COD_TSE', drop = False, inplace=True)

df_equi_novo.head()
```

Out[40]:

	chave	NOME	UF	GEOCOD_IBGE	COD_TSE	AJUSTE
COD_TSE						
1120	ac acrelandia	ACRELÂNDIA	AC	1200013	1120	NaN
1570	ac assis brasil	ASSIS BRASIL	AC	1200054	1570	NaN
1058	ac brasileira	BRASILÉIA	AC	1200104	1058	NaN
1007	ac bujari	BUJARI	AC	1200138	1007	NaN
1015	ac capixaba	CAPIXABA	AC	1200179	1015	NaN

```
In [41]: # Verificando o tamanho dos dataframes
df_vot_novo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5570 entries, 71072 to 90239
Data columns (total 7 columns):
SG_UF                5570 non-null object
CD_MUNICIPIO         5570 non-null int64
NM_MUNICIPIO         5570 non-null object
NR_CANDIDATO         5570 non-null int64
NM_URNA_CANDIDATO    5570 non-null object
QT_VOTOS_NOMINAIS    5570 non-null int64
RESULTADO            5570 non-null bool
dtypes: bool(1), int64(3), object(3)
memory usage: 310.0+ KB
```

```
In [43]: df_equi_novo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5570 entries, 1120 to 96431
Data columns (total 6 columns):
chave                5570 non-null object
NOME                 5570 non-null object
UF                  5570 non-null object
GEOCOD_IBGE         5570 non-null int64
COD_TSE             5570 non-null int64
AJUSTE              29 non-null object
dtypes: int64(2), object(4)
memory usage: 304.6+ KB
```

In [44]: *# Criando novo dataframe com base no merge de votação e equivalencia*

```
df_vot_equi = df_vot_novo.join(df_equi_novo)

df_vot_equi.head()
```

Out[44]:

	SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOTOS_NOMINAIS	RESULTADO	chave
CD_MUNICIPIO								
71072	SP	71072	SÃO PAULO	17	JAIR BOLSONARO	3694834	True	sp sao paulo
60011	RJ	60011	RIO DE JANEIRO	17	JAIR BOLSONARO	2179896	True	rj rio de janeiro
97012	DF	97012	BRASÍLIA	17	JAIR BOLSONARO	1080411	True	df brasilia
38490	BA	38490	SALVADOR	13	FERNANDO HADDAD	985532	False	ba salvador
41238	MG	41238	BELO HORIZONTE	17	JAIR BOLSONARO	901513	True	mg belo horizonte

In [45]: *# Vamos juntar o dataframe do geopandas com o dataframe de votação equivalencia*

In [46]: df_vot_equi.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5570 entries, 71072 to 90239
Data columns (total 13 columns):
SG_UF                5570 non-null object
CD_MUNICIPIO         5570 non-null int64
NM_MUNICIPIO         5570 non-null object
NR_CANDIDATO         5570 non-null int64
NM_URNA_CANDIDATO    5570 non-null object
QT_VOTOS_NOMINAIS    5570 non-null int64
RESULTADO            5570 non-null bool
chave                5570 non-null object
NOME                 5570 non-null object
UF                   5570 non-null object
GEOCOD_IBGE          5570 non-null int64
COD_TSE              5570 non-null int64
AJUSTE               29 non-null object
dtypes: bool(1), int64(5), object(7)
memory usage: 731.1+ KB
```

In [47]: df_mapa.info()

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 3 columns):
NM_MUNICIP           5572 non-null object
CD_GEOCMU            5572 non-null object
geometry             5572 non-null geometry
dtypes: geometry(1), object(2)
memory usage: 130.7+ KB
```

```
In [49]: # transformar a coluna de código do ibge do dataframe de votação/equivalencia
df_vot_equi['GEOCOD_IBGE'] = df_vot_equi['GEOCOD_IBGE'].astype(str)

df_vot_equi.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5570 entries, 71072 to 90239
Data columns (total 13 columns):
SG_UF                5570 non-null object
CD_MUNICIPIO         5570 non-null int64
NM_MUNICIPIO         5570 non-null object
NR_CANDIDATO         5570 non-null int64
NM_URNA_CANDIDATO    5570 non-null object
QT_VOTOS_NOMINAIS    5570 non-null int64
RESULTADO            5570 non-null bool
chave                5570 non-null object
NOME                 5570 non-null object
UF                   5570 non-null object
GEOCOD_IBGE          5570 non-null object
COD_TSE              5570 non-null int64
AJUSTE               29 non-null object
dtypes: bool(1), int64(4), object(8)
memory usage: 731.1+ KB
```

```
In [51]: # Vamos unificar os indices

# Mudando o indice do mapa
df_mapa.set_index('CD_GEOCMU', drop=False, inplace=True)

df_mapa.head()
```

```
Out[51]:
```

	NM_MUNICIP	CD_GEOCMU	geometry
CD_GEOCMU			
3509106	CAIUÃ	3509106	POLYGON ((-51.86001 -21.58072, -51.85892 -21.5...
3511003	CASTILHO	3511003	POLYGON ((-51.41936 -20.68108, -51.41942 -20.6...
3514403	DRACENA	3514403	POLYGON ((-51.48515 -21.42765, -51.48500 -21.4...
3515301	ESTRELA DO NORTE	3515301	POLYGON ((-51.57097 -22.39887, -51.57099 -22.3...
3515350	EUCLIDES DA CUNHA PAULISTA	3515350	POLYGON ((-52.44569 -22.60828, -52.44569 -22.6...

```
In [52]: # Mudandno o indice do dataframe votação/equivalencia
df_vot_equi.set_index('GEOCOD_IBGE', drop = False, inplace= True)
```

```
In [58]: # Vamos juntar os dataframes
df_mapa_novo = df_mapa.join(df_vot_equi)

df_mapa_novo.head().T
```

Out[58]:

	CD_GEOCMU	3509106	3511003	3514403	3515301	3515350
	NM_MUNICIP	CAIUÃ	CASTILHO	DRACENA	ESTRELA DO NORTE	EUCLIDES DA CUNHA PAULISTA
	CD_GEOCMU	3509106	3511003	3514403	3515301	3515350
	POLYGON	POLYGON	POLYGON	POLYGON	POLYGON	POLYGON
geometry	((-51.86001052999998 -21.5807199056459...	((-51.41936432700003 -20.6810765586240...	((-51.48515406800003 -21.4276462056422...	((-51.57097467399996 -22.3988718766653...	((-52.44569233499996 -22.6082794076701...	
	SG_UF	SP	SP	SP	SP	SP
	CD_MUNICIPIO	62839	63215	63894	64076	61000
	NM_MUNICIPIO	CAIUÃ	CASTILHO	DRACENA	ESTRELA DO NORTE	EUCLIDES DA CUNHA PAULISTA
	NR_CANDIDATO	13	17	17	17	13
	NM_URNA_CANDIDATO	FERNANDO HADDAD	JAIR BOLSONARO	JAIR BOLSONARO	JAIR BOLSONARO	FERNANDO HADDAD
	QT_VOTOS_NOMINAIS	1320	5005	19141	875	2860
	RESULTADO	False	True	True	True	False
	chave	sp caiua	sp castilho	sp dracena	sp estrela do norte	sp euclides da cunha paulista
	NOME	CAIUÃ	CASTILHO	DRACENA	ESTRELA DO NORTE	EUCLIDES DA CUNHA PAULISTA
	UF	SP	SP	SP	SP	SP
	GEOCOD_IBGE	3509106	3511003	3514403	3515301	3515350
	COD_TSE	62839	63215	63894	64076	61000
	AJUSTE	NaN	NaN	NaN	NaN	NaN


```
In [59]: # Verificando o tamanho do mapa novo
df_mapa_novo.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
Index: 5572 entries, 3509106 to 1101807
Data columns (total 16 columns):
NM_MUNICIP          5572 non-null object
CD_GEOCMU           5572 non-null object
geometry            5572 non-null geometry
SG_UF               5570 non-null object
CD_MUNICIPIO        5570 non-null float64
NM_MUNICIPIO        5570 non-null object
NR_CANDIDATO        5570 non-null float64
NM_URNA_CANDIDATO   5570 non-null object
QT_VOTOS_NOMINAIS   5570 non-null float64
RESULTADO           5570 non-null object
chave               5570 non-null object
NOME                5570 non-null object
UF                  5570 non-null object
GEOCOD_IBGE         5570 non-null object
COD_TSE             5570 non-null float64
AJUSTE              29 non-null object
dtypes: float64(4), geometry(1), object(11)
memory usage: 900.0+ KB
```

```
In [60]: # Vamos descobrir os dois municipios a mais no mapa
df_mapa_novo[df_mapa_novo['NR_CANDIDATO'].isnull()]
```

Out[60]:

	NM_MUNICIP	CD_GEOCMU	geometry	SG_UF	CD_MUNICIPIO	NM_MUNICIPIO	NR_CANDIDATO	NM_URNA_CANDIDATO	QT_VOT
CD_GEOCMU									
4300001	LAGOA MIRIM	4300001	POLYGON ((-52.62241 -32.14662, -52.62802 -32.1...	NaN	NaN	NaN	NaN	NaN	NaN
4300002	LAGOA DOS PATOS	4300002	POLYGON ((-51.29028 -30.00222, -51.28939 -30.0...	NaN	NaN	NaN	NaN	NaN	NaN

```
In [61]: # Matar a Coluna de ajuste
df_mapa_novo.drop(columns=['AJUSTE'], inplace=True )
```

```
In [62]: # Vamos eliminar informação das duas lagos do RS
df_mapa_novo.dropna(inplace=True)
```

```
In [73]: # Vamos desenha o mapa

# Criar a referência para o código de cor (False/0 = red, True/1 = green)
from matplotlib.colors import ListedColormap

cmap = ListedColormap(['red', 'green'])

# Desenhando o mapa
fig, ax = plt.subplots(1, figsize=(12, 12))

ax = df_mapa_novo.plot(column='RESULTADO', cmap=cmap, legend=True, linewidth=0.1, ax=ax,
                        edgecolor='grey')

# Escondendo o eixos
ax.set_axis_off()

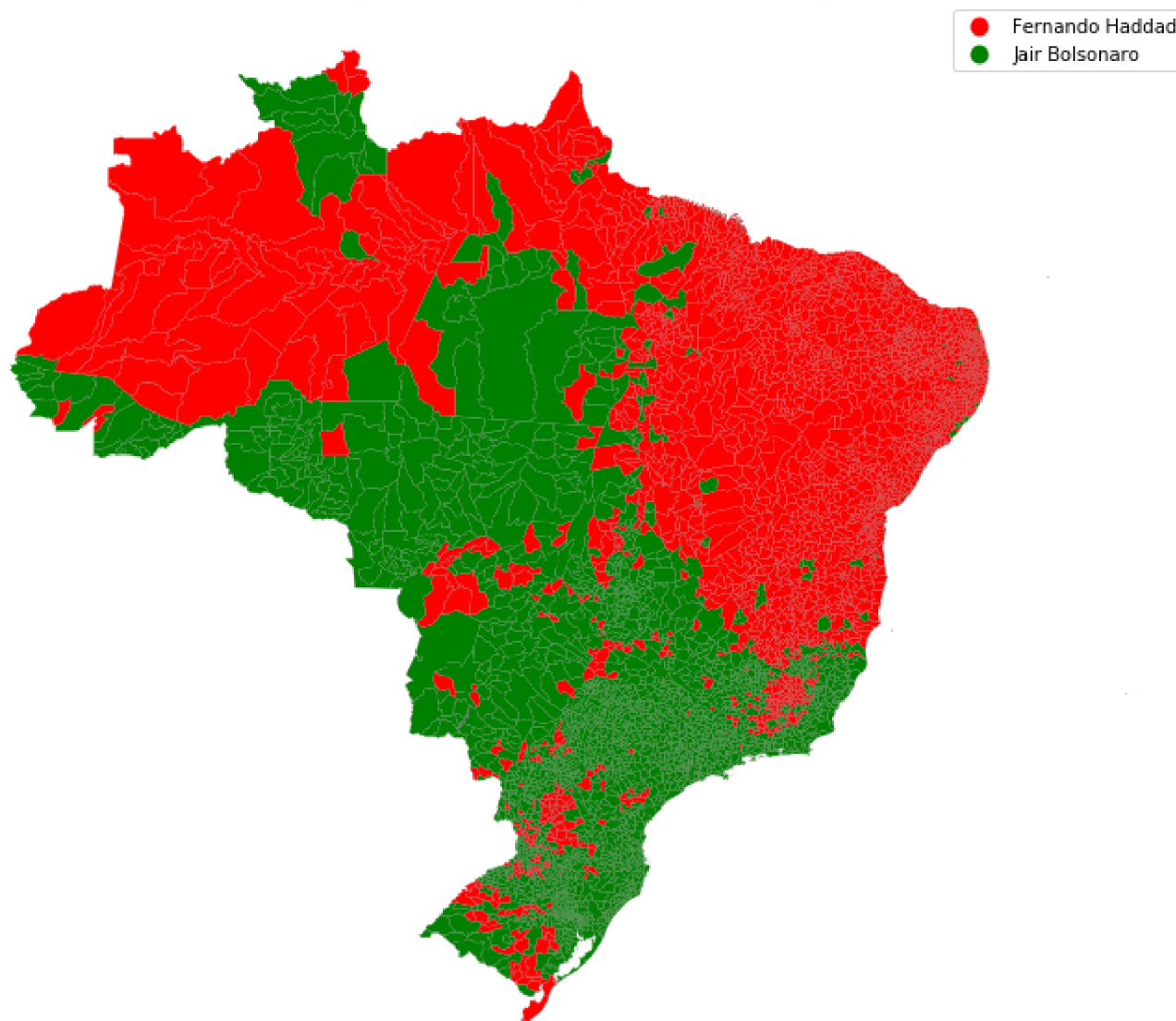
# Colocando um título
ax.set_title('Resultado Eleição 2018 por Município / 2º Turno', fontdict={'fontsize': '25', 'fontweight': '3'})

# Criando uma nota do rodapé
ax.annotate('Fonte: IBGE, 2015; TSE, 2019; Jornal o Estado de São Paulo, 2019',
            xy=(0.1, 0.08), xycoords='figure fraction', horizontalalignment='left',
            verticalalignment='top', fontsize=10, color='#555555')

# Alterando a Legenda
leg = ax.get_legend()
leg.get_texts()[0].set_text('Fernando Haddad')
leg.get_texts()[1].set_text('Jair Bolsonaro')

plt.show()
```

Resultado Eleição 2018 por Município / 2º Turno



Fonte: IBGE, 2015; TSE, 2019; Jornal o Estado de São Paulo, 2019

In []: