

Projeto final Visualização da Informação - Análise Exploratória

O objetivo desse projeto é explorar dados eleitorais do sudeste do país, mais especificamente das eleições presidenciais de 2018. O ano escolhido a princípio havia sido 2022, pois é um tema bem recente e de interesse, mas ao utilizar os dados do TSE (através da Base dos Dados, como farei ao longo do trabalho) tive problemas com dados faltantes e muito inconsistentes. Portanto, o projeto está sendo desenvolvido para 2018, mas tenho forte intenção de revisitá-lo mais adiante para explorar possibilidades com o ano de 2022.

Queremos, ao final da análise, ter gráficos prontos para utilização em uma peça final de visualização, que terá o formato de um infográfico, então as análises aqui presentes serão posteriormente "limpas", isto é, removerei legendas, eixos e tudo que será composto posteriormente na peça final.

Iniciarei realizando as importações necessárias:

```
# importações
import pandas as pd
import altair as alt
import json
import geopandas as gpd
import basedosdados as bd
from shapely import wkt
import matplotlib.pyplot as plt
import seaborn as sns
```

Carregaremos nossos dados através do uso da Base dos Dados, conforme mencionado:

```
# carregar dados eleitorais (tse)
detalhes_votacao_municipio =
bd.read_table(dataset_id='br_tse_eleicoes',
              table_id='detalhes_votacao_municipio',
              billing_project_id="brazilian-elections-2022")
```

```
Downloading: 100%|██████████| 320628/320628 [01:42<00:00,
3128.73rows/s]
```

```
# carregar dados geográficos (ipea)
municipio_geometry = bd.read_table(dataset_id='br_geobr_mapas',
                                   table_id='municipio',
                                   billing_project_id="brazilian-elections-2022")
```

```
Downloading: 100%|██████████| 5567/5567 [01:18<00:00, 70.77rows/s]
```

Iniciamos o processo de tratamento dos dados, buscando apenas aqueles que nos interessam e convertendo os dados espaciais para o formato adequado para plotagem:

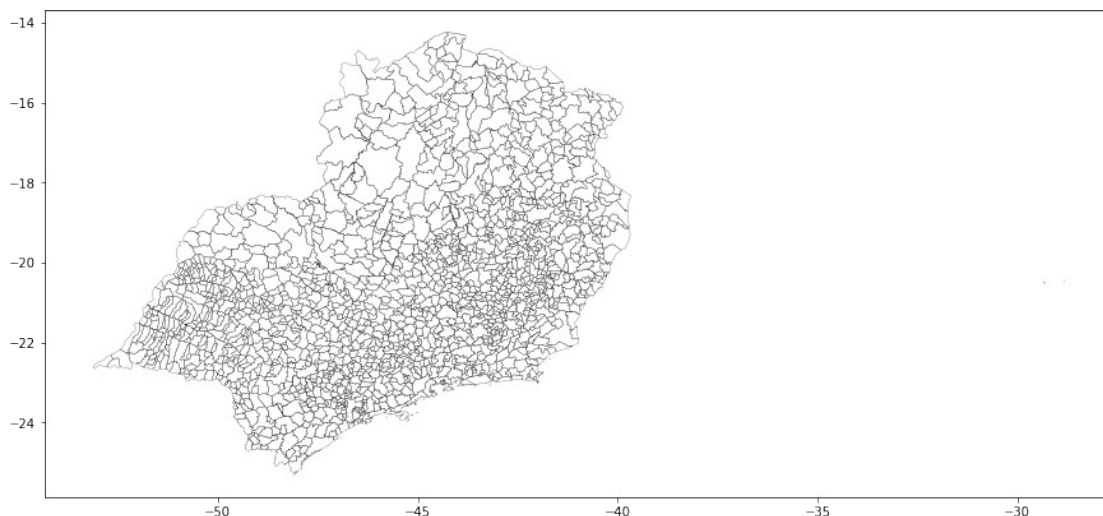
```
# pegar linhas desejadas (ano 2018, cargo presidente, uf valida)
election_2018 = detalhes_votacao_municipio.query("cargo ==
```

```
'presidente' & ano == 2018 & sigla_uf != 'ZZ'),)
election_2018['proporcao_abstencao'] = 100 -
election_2018['proporcao_comparecimento']

# filtra o sudeste, junta dados eleitorais e geográficos
municipio_geometry_sudeste = municipio_geometry.query("sigla_uf in
['RJ', 'SP', 'ES', 'MG']")
municipio_geometry_sudeste = pd.merge(municipio_geometry_sudeste,
election_2018, on='id_municipio', how = 'inner')
municipio_geometry_sudeste['geometria'] =
municipio_geometry_sudeste['geometria'].apply(wkt.loads)
municipio_geometry_sudeste_gpd =
gpd.GeoDataFrame(municipio_geometry_sudeste, geometry='geometria')
municipio_geometry_sudeste_gpd['proporcao_brancos_nulos'] =
municipio_geometry_sudeste_gpd['proporcao_votos_brancos'] +
municipio_geometry_sudeste_gpd['proporcao_votos_nulos']
```

A princípio os dados já podem ser utilizados para desenvolver gráficos espaciais, como desejava, então realizamos um teste plotando o sudeste sem informações:

```
# testando geometrias
ax = municipio_geometry_sudeste_gpd.boundary.plot(figsize = (15, 15),
edgecolor = 'black', linewidth = 0.1)
```



Deixamos preparados, também, filtros por turno de eleições, que serão usados mais adiante:

```
# filtros turno
flt1 = ((municipio_geometry_sudeste_gpd['turno'] == 1))
flt2 = ((municipio_geometry_sudeste_gpd['turno'] == 2))
```

Análise brancos e nulos

A princípio realizaremos análises quanto à presença de votos brancos e nulos no sudeste. Note

```
# proporcao brancos e nulos 1º turno
```

```
fig, ax = plt.subplots(1, figsize=(20, 20))
ax2 = municipio_geometry_sudeste_gpd.boundary.plot(figsize = (20,20),
edgecolor = 'k', linewidth = 0.1, ax = ax)
ax = municipio_geometry_sudeste_gpd[flt1].plot(column =
'proporcao_brancos_nulos',
categorical = False,
legend = True,
figsize = (20,20),
ax = ax,
legend_kwds={'shrink': 0.5},
cmap = 'Purples_r',
vmin = 0,
vmax = 25
)

ax.set_axis_off()
# set title
ax.set_title('Proporção de votos brancos e nulos no 1º turno',
fontdict={'fontsize': '15', 'fontweight' : '3'})

plt.show()
```



```
# proporcao brancos e nulos 2º turno
```

```
fig, ax = plt.subplots(1, figsize=(20, 20))
```

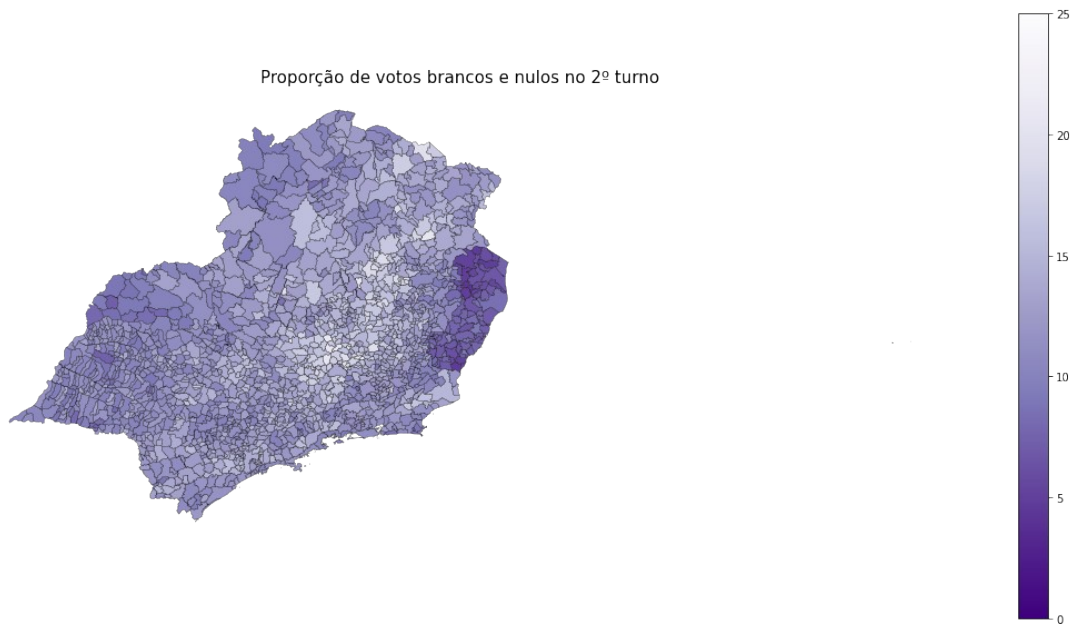
```

ax2 = municipio_geometry_sudeste_gpd.boundary.plot(figsize = (20,20),
edgecolor = 'k', linewidth = 0.1, ax = ax)
ax = municipio_geometry_sudeste_gpd[flt2].plot(column =
'proporcao_brancos_nulos',
categorical = False,
legend = True,
figsize = (20,20),
ax = ax,
legend_kwds={'shrink': 0.5},
cmap = 'Purples_r',
vmin = 0,
vmax = 25
)

ax.set_axis_off()

# set title
ax.set_title('Proporção de votos brancos e nulos no 2º turno',
fontdict={'fontsize': '15', 'fontweight' : '3'})
plt.show()

```



Análise a respeito de abstenções

Realizei também esta plotagem (com relação a abstenções), mas posteriormente, ao pensar melhor em como organizar o infográfico, decidi manter os anteriores na peça e representar essa informação como gráficos de barra, dispostos mais adiante. Mantive aqui os gráficos espaciais produzidos:

```

# proporcao abstencao 1º turno

```

```

fig, ax = plt.subplots(1, figsize=(20, 20))

```

```

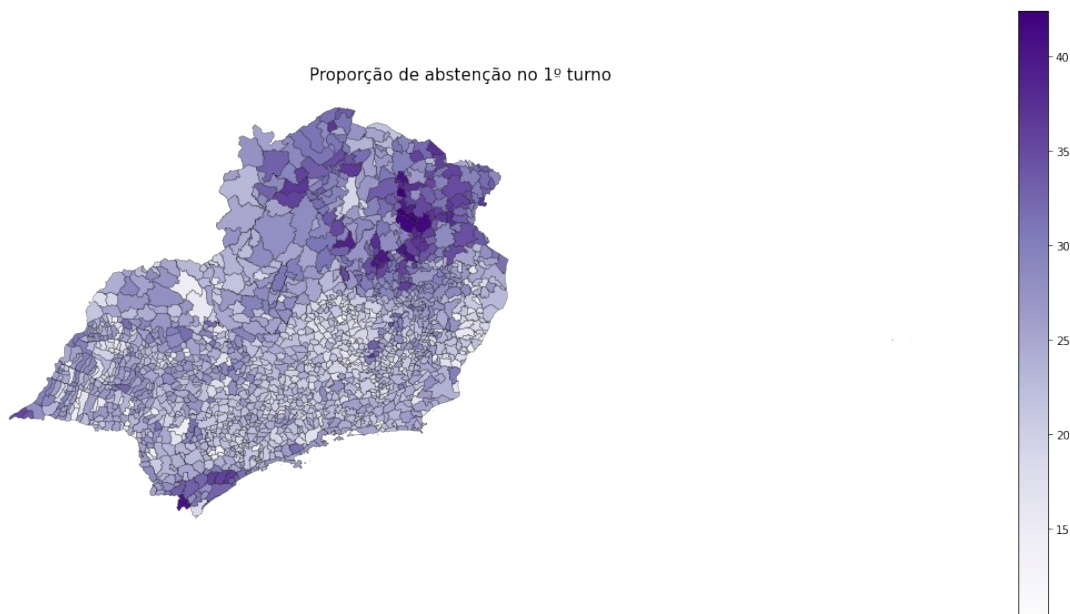
ax2 = municipio_geometry_sudeste_gpd.boundary.plot(figsize = (20,20),
edgecolor = 'k', linewidth = 0.1, ax = ax)
ax = municipio_geometry_sudeste_gpd[flt1].plot(column =
'proporcao_abstencao',
categorical = False,
legend = True,
figsize = (20,20),
ax = ax,
legend_kwds={'shrink': 0.5},
cmap = 'Purples'
)

ax.set_axis_off()

# set title
ax.set_title('Proporção de abstenção no 1º turno',
fontdict={'fontsize': '15', 'fontweight' : '3'})

plt.show()

```



proporcao abstencao 2º turno

```

fig, ax = plt.subplots(1, figsize=(20, 20))
ax2 = municipio_geometry_sudeste_gpd.boundary.plot(figsize = (20,20),
edgecolor = 'k', linewidth = 0.1, ax = ax)
ax = municipio_geometry_sudeste_gpd[flt2].plot(column =
'proporcao_abstencao',
categorical = False,
legend = True,
figsize = (20,20),
ax = ax,
legend_kwds={'shrink': 0.5},

```

```

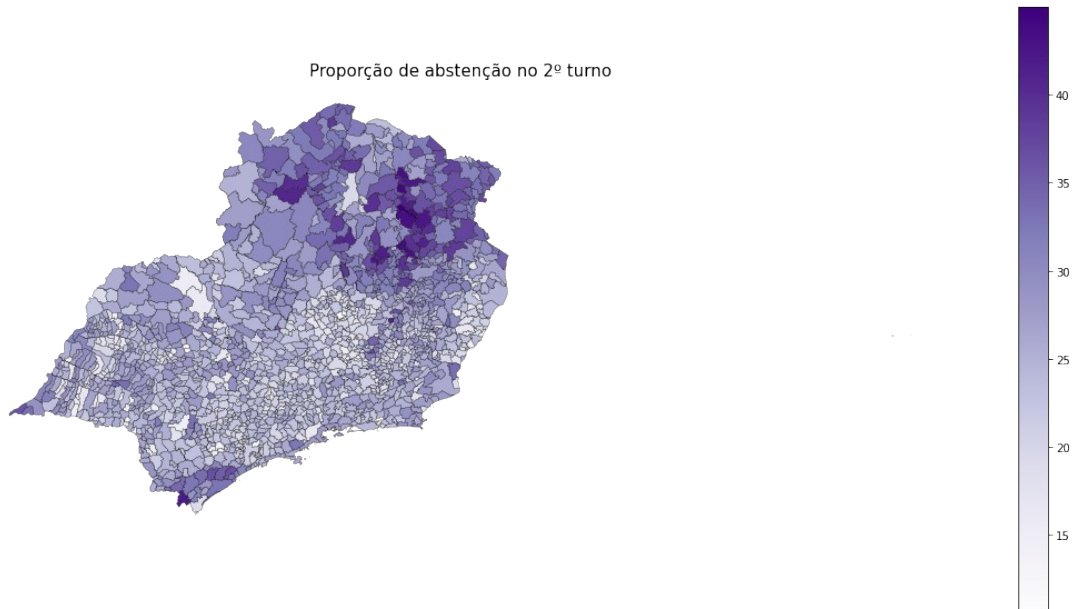
cmap = 'Purples'
)

ax.set_axis_off()

# set title
ax.set_title('Proporção de abstenção no 2º turno',
fontdict={'fontsize': '15', 'fontweight' : '3'})

plt.show()

```



A princípio havia um plano de utilizar 2 gráficos de barra, um para cada turno, alinhados ao gráfico de mapa correspondente. No entanto, a diferença sutil entre os turnos não ficou suficientemente clara ao "empilhar" os gráficos, então preferi utilizar um único de barras duplas. Mantive, também, os gráficos individuais aqui:

```

# proporcao abstencoes 1º turno

```

```

election_2018_1 = election_2018.query("sigla_uf in ['RJ', 'SP', 'ES',
'MG'] & turno == 1")

```

```

# do the same with purple color

```

```

fig, ax = plt.subplots(1, figsize=(7, 5))
ax.set_ylim(0, 27)
ax.set_facecolor('#E6E6E6')
sns.barplot(x = 'sigla_uf', y = 'proporcao_abstencao', data =
election_2018_1, ax = ax, color = '#4A1587')
sns.despine(left=True, bottom=True)

```

```

# set title

```

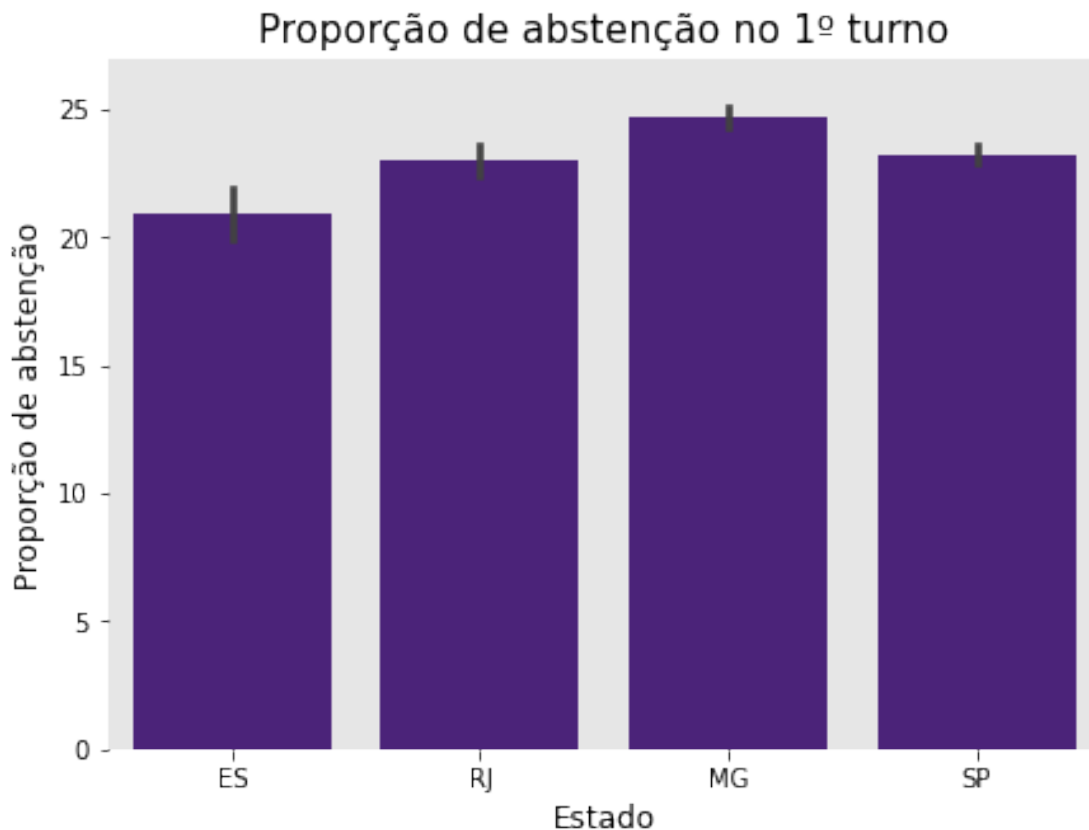
```

ax.set_title('Proporção de abstenção no 1º turno',
fontdict={'fontsize': '15', 'fontweight' : '3'})

```

```
# set labels
ax.set_xlabel('Estado', fontdict={'fontsize': '12', 'fontweight' :
'3'})
ax.set_ylabel('Proporção de abstenção', fontdict={'fontsize': '12',
'fontweight' : '3'})

plt.show()
```



```
# proporcao abstencoes 2º turno

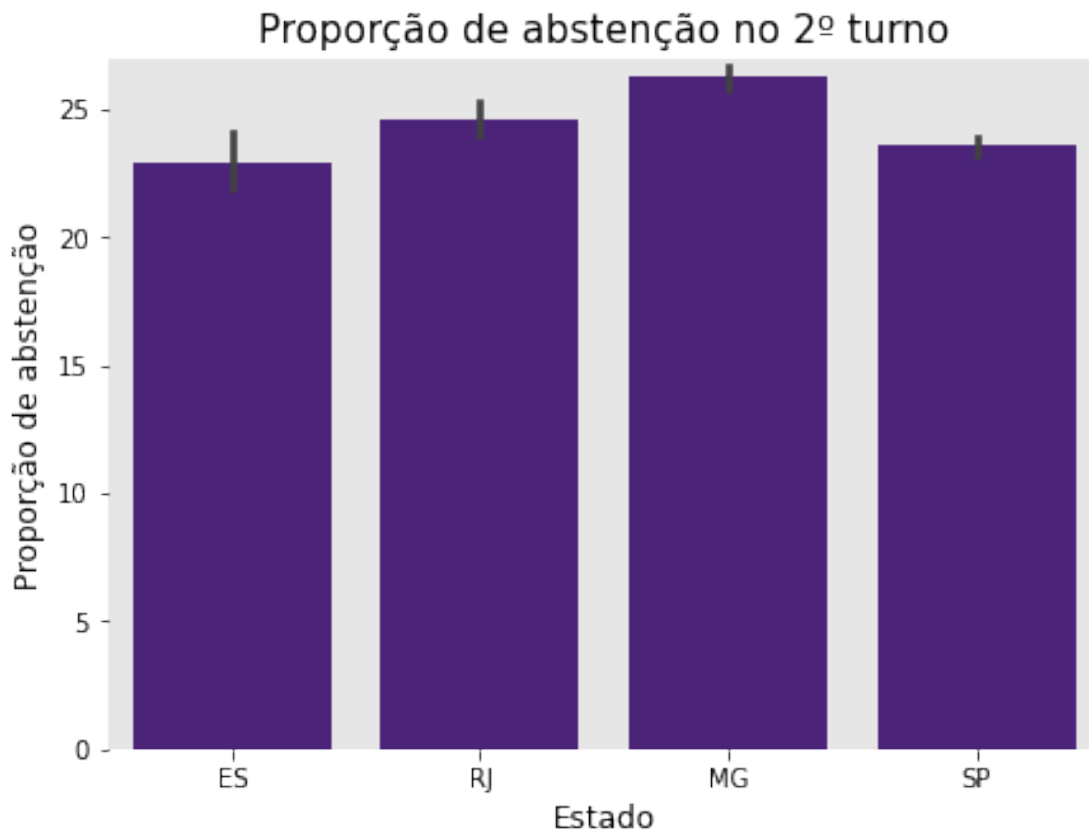
election_2018_2 = election_2018.query("sigla_uf in ['RJ', 'SP', 'ES',
'MG'] & turno == 2")

fig, ax = plt.subplots(1, figsize=(7, 5))
ax.set_ylim(0, 27)
ax.set_facecolor('#E6E6E6')
sns.barplot(x = 'sigla_uf', y = 'proporcao_abstencao', data =
election_2018_2, ax = ax, color = '#4A1587')
sns.despine(left=True, bottom=True)

# set title
ax.set_title('Proporção de abstenção no 2º turno',
fontdict={'fontsize': '15', 'fontweight' : '3'})
```

```
# set labels
ax.set_xlabel('Estado', fontdict={'fontsize': '12', 'fontweight' :
'3'})
ax.set_ylabel('Proporção de abstenção', fontdict={'fontsize': '12',
'fontweight' : '3'})

plt.show()
```



Por fim, desenvolvi o gráfico que preferi utilizar no infográfico, que está a seguir. Observe que, posteriormente, a maior parte dos metadados foi removido e inserido posteriormente na peça final de forma a favorecer a organização da peça.

```
# make horizontal barplot of abstencao by state of 1st and 2nd turn
```

```
election_2018_3 = election_2018.query("sigla_uf in ['RJ', 'SP', 'ES',
'MG']")
```

```
fig, ax = plt.subplots(1, figsize=(5, 7))
ax.set_ylim(0, 27)
ax.set_facecolor('#E6E6E6')
sns.barplot(x = 'proporcao_abstencao', y = 'sigla_uf', data =
election_2018_3, ax = ax, hue = 'turno', palette = ['#4A1587',
'#B39DDB'], ci=None, order = ['ES', 'RJ', 'SP', 'MG'])
```

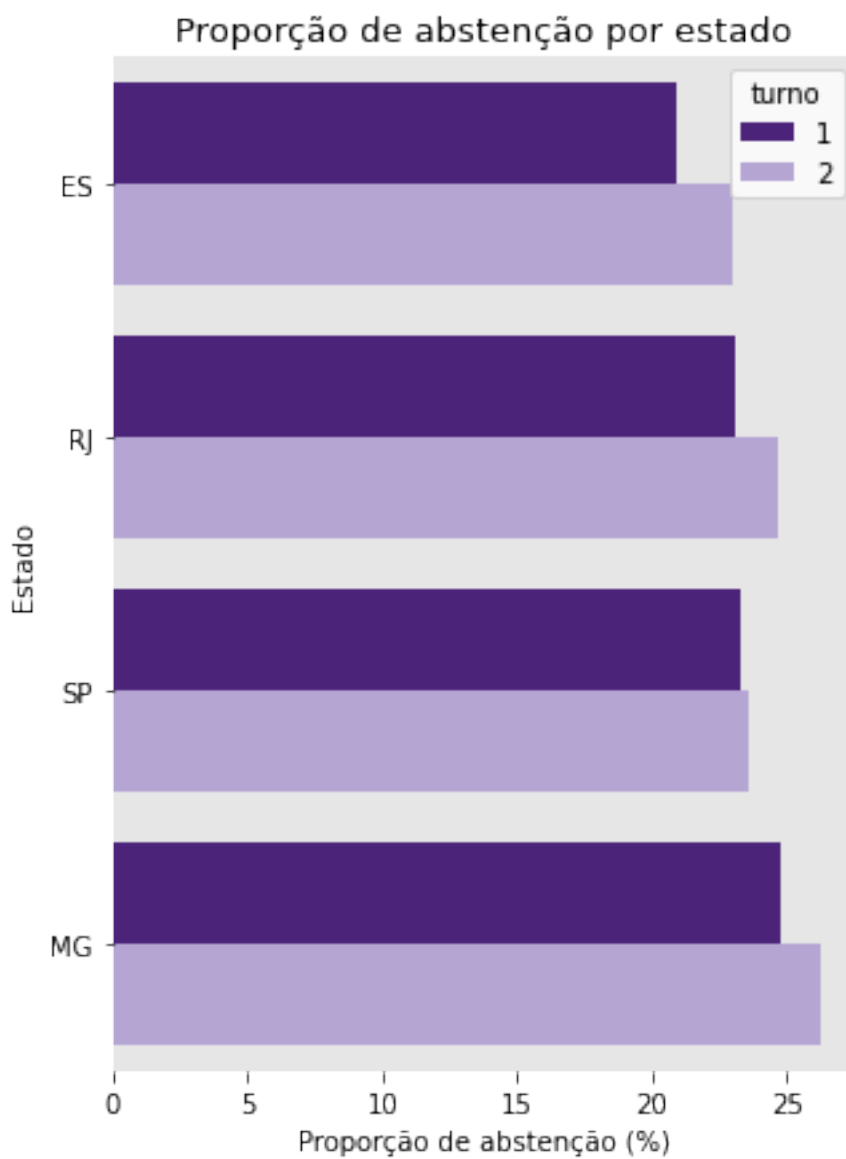


```
sns.despine(left=True, bottom=True)

# set title
ax.set_title('Proporção de abstenção por estado',
fontdict={'fontsize': '13', 'fontweight' : '3'})

# set axis labels
ax.set_xlabel('Proporção de abstenção (%)')
ax.set_ylabel('Estado')

plt.show()
```



Análise Espírito Santo resultado

Na peça final há um destaque a esse estado devido ao resultado observado no gráfico de brancos e nulos no segundo turno. Por isso, realizei algumas análises de resultados nesse estado, e foi necessário receber outra tabela da Base dos Dados:

```
# obtencao de resultados por municipio no 1º turno
```

```
query = """
SELECT *
FROM basedosdados.br_tse_eleicoes.resultados_candidato_municipio
WHERE ano in (2018) AND sigla_uf in ('ES') AND cargo in ('presidente')
AND turno in (1)
"""
```

```
resultados_candidato_municipio_main = bd.read_sql(query,
                                                    billing_project_id="brazilian-
elections-2022")
```

```
Downloading: 100%|██████████| 1014/1014 [00:00<00:00, 2304.18rows/s]
```

```
resultados_candidato_municipio = resultados_candidato_municipio_main
```

Também para diagramação da peça final, decidi fazer um gráfico setorial (havia muitas barras e linhas verticais, pelo próprio gráfico de barras, escala de cores do gráfico espacial, fonte de texto...). O "movimento" me parecia necessário para completar a peça e eu estava decidida a fazer 2 ou 3 setores apenas:

```
# do pie chart of votes by candidate at 1st turn
```

```
# make all numero_candidato not in ['13', '17'] to 'outros'
resultados_candidato_municipio.loc[~resultados_candidato_municipio['nu
mero_candidato'].isin(['13', '17']), 'numero_candidato'] = '14'
```

```
# group by numero_candidate and sum votos
```

```
resultados_candidato_municipio =
resultados_candidato_municipio.groupby(['numero_candidato']).sum().res
et_index()
resultados_candidato_municipio =
resultados_candidato_municipio.sort_values(by=['numero_candidato'],
ascending=False)
```

```
fig, ax = plt.subplots(1, figsize=(10, 5))
ax.pie(resultados_candidato_municipio['votos'], startangle=90, colors
= ['#4E49AF', '#B5A5D3', '#D13D55'])
ax.axis('equal')
```

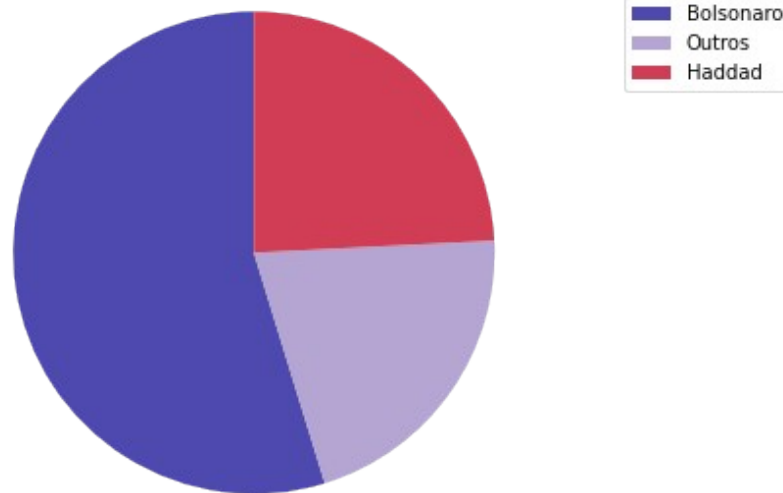
```
# set title
```

```
ax.set_title('Distribuição votos válidos no 1º turno',
fontdict={'fontsize': '15', 'fontweight' : '3'})
```

```
# show legend
plt.legend(labels = ['Bolsonaro', 'Outros', 'Haddad'], loc = 'upper
right')

plt.show()
```

Distribuição votos válidos no 1º turno



```
# dados de votos por municipio no 2º turno
```

```
query = """
SELECT *
FROM basedosdados.br_tse_eleicoes.resultados_candidato_municipio
WHERE ano in (2018) AND sigla_uf in ('ES') AND cargo in ('presidente')
AND turno in (2)
"""
```

```
resultados_candidato_municipio_main2 = bd.read_sql(query,
                                                    billing_project_id="brazilian-
elections-2022")
```

```
Downloading: 100%|██████████| 156/156 [00:00<00:00, 529.18rows/s]
```

```
resultados_candidato_municipio = resultados_candidato_municipio_main2
```

```
# do pie chart of votes by candidate at 1st turn
```

```
# make all numero_candidato not in ['13', '17'] to 'outros'
resultados_candidato_municipio.loc[~resultados_candidato_municipio['nu
mero_candidato'].isin(['13', '17']), 'numero_candidato'] = '14'
```

```
# group by numero_candidate and sum votos
resultados_candidato_municipio =
```

```

resultados_candidato_municipio.groupby(['numero_candidato']).sum().reset_index()
resultados_candidato_municipio =
resultados_candidato_municipio.sort_values(by=['votos'],
ascending=False)

fig, ax = plt.subplots(1, figsize=(10, 5))
ax.pie(resultados_candidato_municipio['votos'], startangle=90,
colors=['#4E49AF', '#D13D55'])
ax.axis('equal')

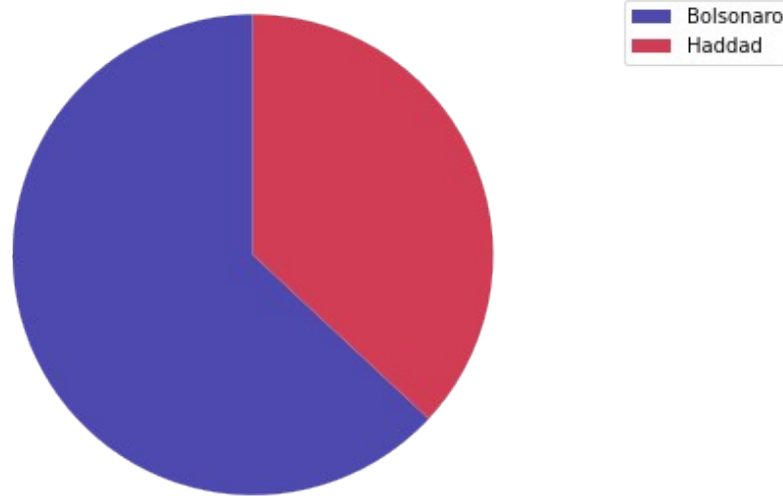
# set title
ax.set_title('Distribuição votos válidos no 2º turno',
fontdict={'fontsize': '15', 'fontweight' : '3'})

# show legend
plt.legend(labels = ['Bolsonaro', 'Haddad'], loc = 'upper right')

plt.show()

```

Distribuição votos válidos no 2º turno



Por fim, realizei a plotagem do mapa do Espírito Santo para composição de detalhes na peça.

```

municipio_geometry_es = municipio_geometry.query("sigla_uf in ['ES']")
municipio_geometry_es = pd.merge(municipio_geometry_es,
resultados_candidato_municipio_main, on='id_municipio', how = 'inner')
municipio_geometry_es['geometria'] =
municipio_geometry_es['geometria'].apply(wkt.loads)
municipio_geometry_es_gpd = gpd.GeoDataFrame(municipio_geometry_es,
geometry='geometria')
ax = municipio_geometry_es_gpd.boundary.plot(figsize = (15, 20),
edgecolor = 'black', linewidth = 0.1)

```

```
ax.set_axis_off()
```

