

✓ Semana 9 - Dashboards

```
1 # Comando de abertura para Google Drive
2 from google.colab import drive
3 drive.mount('/content/drive')
```

↗ Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
1 # Instalando pacotes necessários para execução da visualização
2 !pip install geopandas
```

↗ Requirement already satisfied: geopandas in /usr/local/lib/python3.10/dist-packages (0.13.2)
 Requirement already satisfied: fiona>=1.8.19 in /usr/local/lib/python3.10/dist-packages (from geopandas) (1.9.6)
 Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from geopandas) (24.0)
 Requirement already satisfied: pandas>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from geopandas) (2.0.3)
 Requirement already satisfied: pyproj>=3.0.1 in /usr/local/lib/python3.10/dist-packages (from geopandas) (3.6.1)
 Requirement already satisfied: shapely>=1.7.1 in /usr/local/lib/python3.10/dist-packages (from geopandas) (2.0.4)
 Requirement already satisfied: attr>=19.2.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (23.2.0)
 Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (2024.2.2)
 Requirement already satisfied: click~8.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (8.1.7)
 Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (1.1.1)
 Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (0.7.2)
 Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (1.16.0)
 Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.0->geopandas) (2023.4)
 Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.0->geopandas) (2023.4)
 Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.0->geopandas) (2024.1)
 Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.0->geopandas) (1.25.2)

```
1 # Abertura de bibliotecas
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import geopandas as gpd
```

```
1 # Base de dados dos municípios do Brasil
2 df_mapa = gpd.read_file('/content/drive/MyDrive/BR_Municipios_2022/BR_Municipios_2022.shp')
```

```
1 # Renomeação
2 df_mapa = df_mapa.rename({'CD_MUN': 'CODIGO'}, axis = 1)
3 df_mapa.info()
```

↗ <class 'geopandas.geodataframe.GeoDataFrame'>
 RangeIndex: 5572 entries, 0 to 5571
 Data columns (total 5 columns):

#	Column	Non-Null Count	Dtype
0	CODIGO	5572 non-null	object
1	NM_MUN	5572 non-null	object
2	SIGLA_UF	5572 non-null	object
3	AREA_KM2	5572 non-null	float64
4	geometry	5572 non-null	geometry

 dtypes: float64(1), geometry(1), object(3)
 memory usage: 217.8+ KB

```
1 # Lista os pacotes pré-instalados no Colab
2 !pip list
```

↗ [Mostrar saída oculta](#)

```
1 # Base de dados dos municípios do Brasil
2 df_dados = pd.read_excel('/content/drive/MyDrive/CHAVES/tabela9542.xlsx')
```

```
1 # Renomeando a Coluna Cód. do conjunto de dados df_dados
2 df_dados = df_dados.rename({'Cód.': 'CODIGO'}, axis = 1)
```

```
1 df_dados.info()
```

↗ <class 'pandas.core.frame.DataFrame'>
 RangeIndex: 5570 entries, 0 to 5569
 Data columns (total 6 columns):

#	Column	Non-Null Count	Dtype
0	Nível	5570 non-null	object
1	CODIGO	5570 non-null	int64
2	Município	5570 non-null	object
3	Cor ou raça	5570 non-null	object
4	Idade	5570 non-null	object

```
5 Não_alfabetizadas_2022 5570 non-null int64
dtypes: int64(2), object(4)
memory usage: 261.2+ KB
```

```
1 # Alterando o tipo de dado
2 df_mapa['CODIGO'] = df_mapa['CODIGO'].astype(str)
3 df_dados['CODIGO'] = df_dados['CODIGO'].astype(str)

1 # União dos dataframe mapa e dados
2 uniao_oficial = df_mapa.merge(df_dados, on = "CODIGO", how = "left")
```

```
1 # Visualização
2 uniao_oficial
```

↗

	CODIGO	NM_MUN	SIGLA_UF	AREA_KM2	geometry	Nível	Município	Cor ou raça	Idade	Não_alfabetizadas_2022
0	1100015	Alta Floresta D'Oeste	RO	7067.127	POLYGON ((-62.00806 -12.13379, -62.00784 -12.2...	MU	Alta Floresta D'Oeste (RO)	Total	Total	1419.0
1	1100023	Ariquemes	RO	4426.571	POLYGON ((-63.17933 -10.13924, -63.17746 -10.1...	MU	Ariquemes (RO)	Total	Total	4480.0
2	1100031	Cabixi	RO	1314.352	POLYGON ((-60.52408 -13.32137, -60.37162 -13.3...	MU	Cabixi (RO)	Total	Total	433.0
3	1100049	Cacoal	RO	3793.000	POLYGON ((-61.35502 -11.50452, -61.35524 -11.5...	MU	Cacoal (RO)	Total	Total	4363.0

◀ ▶

```
1 # Valores nulos
2 uniao_oficial.isnull().sum()
```

↗

CODIGO	0
NM_MUN	0
SIGLA_UF	0
AREA_KM2	0
geometry	0
Nível	2
Município	2
Cor ou raça	2
Idade	2
Não_alfabetizadas_2022	2
dtype: int64	

```
1 # Filtrando os valores vazios
2 uniao_oficial[uniao_oficial['Não_alfabetizadas_2022'].isna()]
```

↗

	CODIGO	NM_MUN	SIGLA_UF	AREA_KM2	geometry	Nível	Município	Cor ou raça	Idade	Não_alfabetizadas_2022
4606	4300001	Lagoa Mirim	RS	2884.340	POLYGON ((-52.62752 -32.15022, -52.62816 -32.1...	NaN	NaN	NaN	NaN	NaN

◀ ▶

```
1 # Eliminando os valores vazios
2 uniao_oficial = uniao_oficial.dropna()
```


```
1 # Visualizando os tipos de dados
2 uniao_oficial.info()
```

↗

<class 'geopandas.geodataframe.GeoDataFrame'>				
Index: 5570 entries, 0 to 5571				
Data columns (total 10 columns):				
#	Column	Non-Null Count	Dtype	
0	CODIGO	5570 non-null	object	
1	NM_MUN	5570 non-null	object	
2	SIGLA_UF	5570 non-null	object	
3	AREA_KM2	5570 non-null	float64	
4	geometry	5570 non-null	geometry	
5	Nível	5570 non-null	object	
6	Município	5570 non-null	object	
7	Cor ou raça	5570 non-null	object	
8	Idade	5570 non-null	object	


```
9 Não_alfabetizadas_2022 5570 non-null float64
dtypes: float64(2), geometry(1), object(7)
memory usage: 478.7+ KB
```

```
1 uniao_oficial.head()
```



	CODIGO	NM_MUN	SIGLA_UF	AREA_KM2	geometry	Nível	Município	Cor ou raça	Idade	Não_alfabetizadas_2022
0	1100015	Alta Floresta D'Oeste	RO	7067.127	POLYGON ((-62.00806 -12.13379, -62.00784 -12.2...	MU	Alta Floresta D'Oeste (RO)	Total	Total	1419.0
1	1100023	Ariquemes	RO	4426.571	POLYGON ((-63.17933 -10.13924, -63.17746 -10.1...	MU	Ariquemes (RO)	Total	Total	4480.0


```
1 # Transformando as informações
2 uniao_oficial['Não_alfabetizadas_2022'] = uniao_oficial['Não_alfabetizadas_2022'].astype(int)
```



```
/usr/local/lib/python3.10/dist-packages/geopandas/geodataframe.py:1538: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```


See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-super\(\).__setitem__\(key, value\)](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-super().__setitem__(key, value))

```
1 uniao_oficial.info()
```



```
<class 'geopandas.geodataframe.GeoDataFrame'>
Index: 5570 entries, 0 to 5571
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   CODIGO                5570 non-null   object
1   NM_MUN                5570 non-null   object
2   SIGLA_UF              5570 non-null   object
3   AREA_KM2              5570 non-null   float64
4   geometry              5570 non-null   geometry
5   Nível                 5570 non-null   object
6   Município             5570 non-null   object
7   Cor ou raça           5570 non-null   object
8   Idade                 5570 non-null   object
9   Não_alfabetizadas_2022 5570 non-null   int64
dtypes: float64(1), geometry(1), int64(1), object(7)
memory usage: 478.7+ KB
```

```
1 # Calculos
2 uniao_oficial.describe()
```




	AREA_KM2	Não_alfabetizadas_2022
count	5570.000000	5570.000000
mean	1525.553304	2047.361041
std	5605.417197	5275.921510
min	3.565000	20.000000
25%	204.755250	397.000000
50%	417.692500	939.500000
75%	1026.610000	2248.000000
max	159533.306000	245357.000000

```
1 # Municipio com maior valor de analfabetos
2 uniao_oficial[uniao_oficial['Não_alfabetizadas_2022'] == 245357]
```



	CODIGO	NM_MUN	SIGLA_UF	AREA_KM2	geometry	Nível	Município	Cor ou raça	Idade	Não_alfabetizadas_2022
--	--------	--------	----------	----------	----------	-------	-----------	----------------	-------	------------------------

```
1 # Municipio com menor valor de analfabetos
2 uniao_oficial[uniao_oficial['Não_alfabetizadas_2022'] == 20]
```

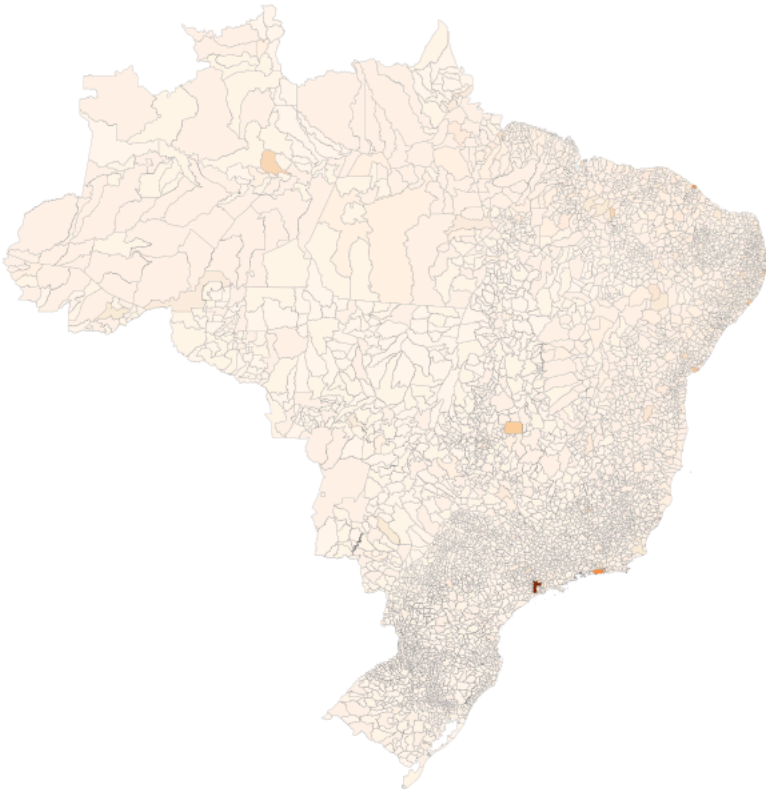


	CODIGO	NM_MUN	SIGLA_UF	AREA_KM2	geometry	Nível	Município	Cor ou raça	Idade	Não_alfabetizadas_2022
--	--------	--------	----------	----------	----------	-------	-----------	----------------	-------	------------------------

```
1 # Expandir o tamanho do mapa
2 fig, ax = plt.subplots(1, figsize = (12,12))
3
4 # Desenhar o mapa
5 ax = uniao_oficial.plot(column='Não_alfabetizadas_2022',
6                          cmap='Oranges',
7                          linewidth=.1,
8                          ax=ax,
9                          edgecolor='gray',
10                         legend=True,
11                         legend_kwds={'label': "Pessoas não-alfabetizadas", 'orientation': "horizontal"})
12
13 # Tirar os eixos do mapa
14 ax.set_axis_off()
15
16 #Colocando título
17 ax.set_title('Analfabetismo no Brasil - 2022', fontdict={'fontsize':'25',
18                                                         'fontweight': '3'})
19
20 # Nota de rodapé
21 ax.annotate('Fonte: IBGE - Censo Demográfico (Tabela 9542)',
22             xy=(0.1, 0.05),
23             xycoords='figure fraction',
24             horizontalalignment='left',
25             verticalalignment='top',
26             fontsize=10,
27             color='#555555')
28
29 # Mostrar o mapa
30 plt.show()
```



Analfabetismo no Brasil - 2022



```
1 # Filtrar apenas as linhas onde 'SIGLA_UF' é 'SP'
2 uniao_oficial_SP = uniao_oficial[uniao_oficial['SIGLA_UF'] == 'SP']
```

```
1 # Visualizando o Estado de SP
2 uniao_oficial_SP
```



	CODIGO	NM_MUN	SIGLA_UF	AREA_KM2	geometry	Nível	Município	Cor ou raça	Idade	Não_alfabetizadas_2022
3267	3500105	Adamantina	SP	411.987	POLYGON ((-51.09557 -21.57029, -51.09617 -21.5...	MU	Adamantina (SP)	Total	Total	1032
3268	3500204	Adolfo	SP	211.055	POLYGON ((-49.61249 -21.26110, -49.61249 -21.2...	MU	Adolfo (SP)	Total	Total	242
3269	3500303	Aguai	SP	474.554	POLYGON ((-47.01254 -22.00527, -47.01219 -22.0...	MU	Aguai (SP)	Total	Total	1287
3270	3500402	Águas da Prata	SP	142.673	POLYGON ((-46.71875 -21.95837, -46.71878 -21.9...	MU	Águas da Prata (SP)	Total	Total	196
3271	3500501	Águas de Lindóia	SP	60.126	POLYGON ((-46.61761 -22.51172, -46.61864 -22.5...	MU	Águas de Lindóia (SP)	Total	Total	544

```
1 # Expandir o tamanho do mapa
2 fig, ax = plt.subplots(1, figsize = (12,12))
3
4 # Desenhar o mapa
5 ax = uniao_oficial_SP.plot(column='Não_alfabetizadas_2022',
6                             cmap='Oranges',
7                             linewidth=.1,
8                             ax=ax,
9                             edgecolor='gray',
10                             legend=True,
11                             legend_kwds={'label': "Pessoas não-alfabetizadas", 'orientation': "horizontal"})
12
13 # Tirar os eixos do mapa
14 ax.set_axis_off()
15
16 #Colocando título
17 ax.set_title('Analfabetismo no SP - 2022', fontdict={'fontsize':'25',
18                                                         'fontweight': '3'})
19
20 # Nota de rodapé
21 ax.annotate('Fonte: IBGE - Censo Demográfico (Tabela 9542)',
22             xy=(0.1, 0.05),
23             xycoords='figure fraction',
24             horizontalalignment='left',
25             verticalalignment='top',
26             fontsize=10,
27             color='#555555')
28
29 # Mostrar o mapa
30 plt.show()
```

Analfabetismo no SP - 2022

```
1 # Cálculos numéricos
2 uniao_oficial_SP.describe()
```

	AREA_KM2	Não_alfabetizadas_2022
count	645.000000	645.000000
mean	384.836405	1757.241860
std	319.930434	9951.338419
min	3.612000	30.000000
25%	158.025000	268.000000
50%	281.033000	562.000000
75%	511.621000	1341.000000
max	1978.795000	245357.000000

```
1 # Município com maior valor de analfabetos
2 uniao_oficial_SP[uniao_oficial_SP['Não_alfabetizadas_2022'] == 245357]
```

CODIGO	NM_MUN	SIGLA_UF	AREA_KM2	geometry	Nível	Município	Cor ou raça	Idade	Não_alfabetizadas_2022
--------	--------	----------	----------	----------	-------	-----------	-------------	-------	------------------------

```
1 # Município com menor valor de analfabetos
2 uniao_oficial_SP[uniao_oficial_SP['Não_alfabetizadas_2022'] == 30]
```

CODIGO	NM_MUN	SIGLA_UF	AREA_KM2	geometry	Nível	Município	Cor ou raça	Idade	Não_alfabetizadas_2022
--------	--------	----------	----------	----------	-------	-----------	-------------	-------	------------------------

```
1 !pip install plotly
```

```
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (5.15.0)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly) (8.3.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from plotly) (24.0)
```

```
1 import pandas as pd
2 import geopandas as gpd
3 import plotly.express as px
```

```
1 # Crie um GeoDataFrame
2 gdf = gpd.GeoDataFrame(uniao_oficial_SP, geometry='geometry')
3
4 # Crie o mapa interativo usando Plotly
5 fig = px.choropleth(gdf,
6                     geojson=gdf.geometry,
7                     locations=gdf.index,
8                     color="Não_alfabetizadas_2022",
9                     color_continuous_scale= "mint",
10                    hover_name="NM_MUN",
11                    hover_data={"AREA_KM2": True, "Não_alfabetizadas_2022": True},
12                    labels={"Não_alfabetizadas_2022": "Não Alfabetizadas (2022)"},
13                    )
14
15 # Atualize o layout do mapa
16 fig.update_geos(fitbounds="locations", visible=False)
17 fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
18
19 # Mostre o mapa
```