

```

1 # import das bibliotecas
2
3 import pandas as pd
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import seaborn as sns
7 import plotly.express as px
8 import plotly.figure_factory as ff

```

Criar os DataFrame

```

1 from google.colab import drive
2 drive.mount('/content/drive')

```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.r



```

1 tb5 = pd.read_csv('tb_vendors_treatment.csv')
2 tb5.head(3)

```

	id_fornecedor	nome_fornecedor	data_registro	qtde_produtos	preco_me
0	5f550609cba01d00210ed015	Meus Bolos	2020-09-06 00:42:34	0	
1	5f584189fc58110021464422	Moça Doceria	2020-09-09 01:56:25	0	
2	5f5a8be1ccc6c40021a8185f	Cami's Candy	2020-09-10 00:49:18	0	

```

1 tb1 = pd.read_csv('tb_addresses_treatment.csv')
2 tb1.tail(2)

```

	id_endereco	bairro	cidade	estado	uf	pais
81031	616058c21770da002259306f	Setor Morada do Sol (Taquaralto)	Palmas	Tocantins	TO	Brasil
81032	60e4bec7a3d5dd0021ab88d3	Plano Diretor Sul	Palmas	Tocantins	TO	Brasil

```

1 tb13 = pd.read_csv('tb_products_treatment.csv')
2 tb13.head(2)

```

	id_produto	id_fornecedor	id_tipo	
0	5f550676cba01d00210ed018	5f550609cba01d00210ed015	5f518a26a4a5fbed86099757	Bol cen t n-1

▼ Rodrigo

Esse arquivo não é o original, fiz um a parte para não correr o risco de perder o principal, observem as relações a seguir:

OBS: Isso_aqui_eh_Flamengo

```
1 ## Usei o 'tb_products_treatment.csv'
2 # Filtrei para obter apenas os produtos aprovados
3 # Filtrei para obter apenas os produtos que podem ser entregáveis
4
5 tb13 = tb13[tb13['aprovado'] == True]
6 tb13 = tb13[tb13['entregavel'] == True]
7 tb13.shape

(1995, 11)
```

```
1 ## Usei o 'tb_vendors_treatment.csv'
2 # Filtrei para obter apenas as vendedores aprovados
3
4 tb5 = tb5[tb5['e_aprovado'] == True]
5 tb5.head(2)
```

	id_fornecedor	nome_fornecedor	data_registro	qtde_produtos	preco_me
2	5f5a8be1ccc6c40021a8185f	Cami's Candy	2020-09-10 00:49:18	0	
3	5f5bc310b8742200213bda1d	Re doces	2020-09-11 01:07:10	0	

```
1 ## Concatenei vendedores e produtos
2 df_vendors_products = pd.merge(tb5, tb13)
3 df_vendors_products.head(2)
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/reshape/merge.py:1127: UserWarning
```

```
You are merging on int and float columns where the float values are not equal to thei
```

```
1 ## Usei o 'tb_addresses_treatment.csv'
2 # Filtrei para obter dados da cidade que pré-determinamos
3 tb1 = tb1[tb1['cidade'] == 'São Paulo']
4 tb1.head(2)
```

	id_endereco	bairro	cidade	estado	uf	pais	tipo	
96	61421da6c6638d0022d9565d	Indianópolis	São Paulo	São Paulo	SP	Brasil	HOME	4
99	6165c6fcc1fbab0022f398e3	Vila Nair	São Paulo	São Paulo	SP	Brasil	HISTORY	4

```
1 ## Concatenei vendedores e produtos com endereços
2 # assim temos em um dataframe somente produtos e vendedores aprovados na cidade de São
3 # com os id_endereco de vendedores e endereços ordenados?
4 df_vendors_products_addresses = pd.merge(df_vendors_products, tb1)
5 df_vendors_products_addresses.head(2)
```

	id_fornecedor	nome_fornecedor	data_registro	qtde_produtos	preco_med
0	5f5a8be1ccc6c40021a8185f	Cami's Candy	2020-09-10 00:49:18	0	
1	5f5a8be1ccc6c40021a8185f	Cami's Candy	2020-09-10 00:49:18	0	

```
1 # Adelaide disse que faz sentido com a quantidade de empresas ativos segundo as Lives
2 len(df_vendors_products_addresses['nome_empresa'].unique())
```

```
105
```

```
1 df_vendors_products_addresses['delivery_raio_km'].unique()
```

```
array([15, 38, 20, 12,  6, 30, 50,  5, 40, 25, 10,  1,  7, 16, 19,  8, 13,
        3, 22,  2, 18,  4])
```

```
1 df_vendors_products_addresses['avaliacao_media'].unique()
```

```
array([0])
```

```
1 df_vendors_products_addresses.shape
```

```
(1680, 28)
```

```
1 # df_saopaulo = pd.read_csv('dataset-limpo.csv', sep=',')
2 # df_saopaulo.head()
```

```
1 df_kaggle = pd.read_csv('dataset-limpo.csv')
2 df_kaggle.head(2)
```

	id	bairro	created_at	descricao	endereco	id.1	ip_address_origin	lati
0	49787	Butantã	2015-06-16 23:34:01.000	5 bandidos, armas, entraram em casa e vasculha...	Avenida Trona Constanzo, 177, Butantã	49787	NaN	-23.58
1	10679	Itaquera	2013-09-11 18:26:31.819	Meu caso foi para o Pequenas Causas, ganhamos ...	Avenida José Pinheiro Borges, 450-876 - Itaque...	10679	187.11.1.51	-23.54

```
1 df_kaggle_sp = pd.DataFrame(df_kaggle, columns=['bairro', 'latitude', 'longitude'])

1 df_kaggle_sp.shape

(12899, 3)

1 df_kaggle_sp.drop_duplicates(subset=['bairro'], inplace=True)

1 teste = df_kaggle_sp['bairro'].unique().tolist()

1 df_vendors_products_addresses_filtrado = pd.DataFrame(df_vendors_products_addresses, co
2                                                         'e_aprovado', 'nome',
3                                                         'bairro', 'cidade', 'e
4                                                         'uf', 'tipo'])

1 df_nestle = pd.merge(df_vendors_products_addresses_filtrado, df_kaggle_sp, how='left',

1 fig = px.density_mapbox(df_nestle, lat='latitude', lon='longitude', radius=5, zoom=10,
2                             mapbox_style="stamen-terrain", hover_name= 'nome_fornecedor', z
3 fig.update_layout(
4     title = 'Regiões Vem de Bolo',
5 )
6 fig.show()
```

Regiões Vem de Bolo



```
1 df_cognatis = pd.read_csv('tb_solds_cognatis_treatment.csv')
```

```
1 df_cognatis_sp = df_cognatis[df_cognatis['MUNICIPIO'] == 'SAO PAULO']
```

Dropar colunas menos importantes para análise

```
1 df_cognatis_sp.head(1)
```

	SETOR_ATIV	UF	MUNICIPIO	CEP	TRABALHADORES	S_RENDA_MEDIA	S_FLI
4871	AGROPECUARIA	SP	SAO PAULO	49400000.0	8889.0	1787.380005	

```
1 df_cognatis_sp_filter = pd.DataFrame(df_cognatis_sp, columns=['CEP', 'TRABALHADORES', '
2                                     'S_PCT_CLASSE_A', 'S_PCT_
3                                     'S_PCT_JOVENS', 'S_PCT_AD
```

```
1 df_cognatis_sp_filter.head(5)
```

	CEP	TRABALHADORES	S_POP_TOTAL	S_PCT_CLASSE_A	S_PCT_CLASSE_B	S_PCT_
	4871	49400000.0	8889.0	158260.0	0.04	17.910000
	4872	55290001.0	17338.0	105274.0	4.26	45.820000
	4873	40000001.0	16464.0	144552.0	1.88	38.810001

```

1 plt.figure(figsize=(12,6))
2 sns.heatmap(df_cognatis_sp_filter.corr(), annot=True, cmap='Wistia',)
3 plt.xticks(rotation=70)
4 plt.show()

```



```
1 df_vendors_products_addresses['cidade'].value_counts()
```

```

São Paulo    1680
Name: cidade, dtype: int64

```

```

1 df_vendedores = pd.DataFrame(df_vendors_products_addresses, columns=[
2     'nome_fornecedor', 'data_registro',
3     'hora_abertura', 'hora_fechamento',
4     'delivery_raio_km', 'tempo_preparacao_minutos', 'bairro',
5     'cidade', 'tipo'])

```

```
1 df_vendedores['tipo'].value_counts(True) *100
```

```

OFFICE    99.22619
HISTORY    0.77381
Name: tipo, dtype: float64

```

```
1 df_office = df_vendedores[df_vendedores['tipo'] == 'OFFICE']
2 df_office.head(2)
```

	nome_fornecedor	data_registro	hora_abertura	hora_fechamento	delivery_raio_km
0	Cami's Candy	2020-09-10 00:49:18	08:00	19:00	15
1	Cami's Candy	2020-09-10 00:49:18	08:00	19:00	15

```
1 df_office['delivery_raio_km'].value_counts(True).head() *100
```

```
50    15.416917
15    13.257349
10    13.197361
25     9.958008
20     7.798440
Name: delivery_raio_km, dtype: float64
```

```
1 df_office['hora_abertura'].value_counts().head()
```

```
09:00    514
10:00    382
08:00    281
12:00    110
11:00    103
Name: hora_abertura, dtype: int64
```

```
1 df_office['hora_fechamento'].value_counts().head()
```

```
20:00    411
18:00    369
19:00    207
21:00    153
18:30    143
Name: hora_fechamento, dtype: int64
```

```
1 df_nestle.head()
```

	nome_fornecedor	nome_empresa	e_aprovado	nome	delivery_raio_km	bairro	c:
0	Cami's Candy	Csmws Csydy	True	Brigadeiro prestígio	15	Brooklin	
1	Cami's Candy	Csmws Csydy	True	Brigadeiro tradicional	15	Brooklin	

```
1 df_nestle['bairro'].value_counts().head(10)
```

```
Bela Vista          109
Vila Olímpia        101
Vila Andrade         78
Mirandópolis        69
Campo Belo          64
Alto de Pinheiros   60
Jardim Sao Paulo(Zona Leste) 60
Santana              55
Pinheiros           53
Itaim Bibi           50
Name: bairro, dtype: int64
```

```
1 df_top10_aprov = pd.DataFrame(df_nestle['bairro'].value_counts().head(10))
2 df_top10_aprov.reset_index(inplace=True)
3 df_top10_aprov.rename(columns={'index': 'Bairros', 'bairro': 'Qtde'}, inplace=True)
```

```
1 df_top10_aprov
```

	Bairros	Qtde
0	Bela Vista	109
1	Vila Olímpia	101
2	Vila Andrade	78
3	Mirandópolis	69
4	Campo Belo	64
5	Alto de Pinheiros	60
6	Jardim Sao Paulo(Zona Leste)	60
7	Santana	55
8	Pinheiros	53
9	Itaim Bibi	50

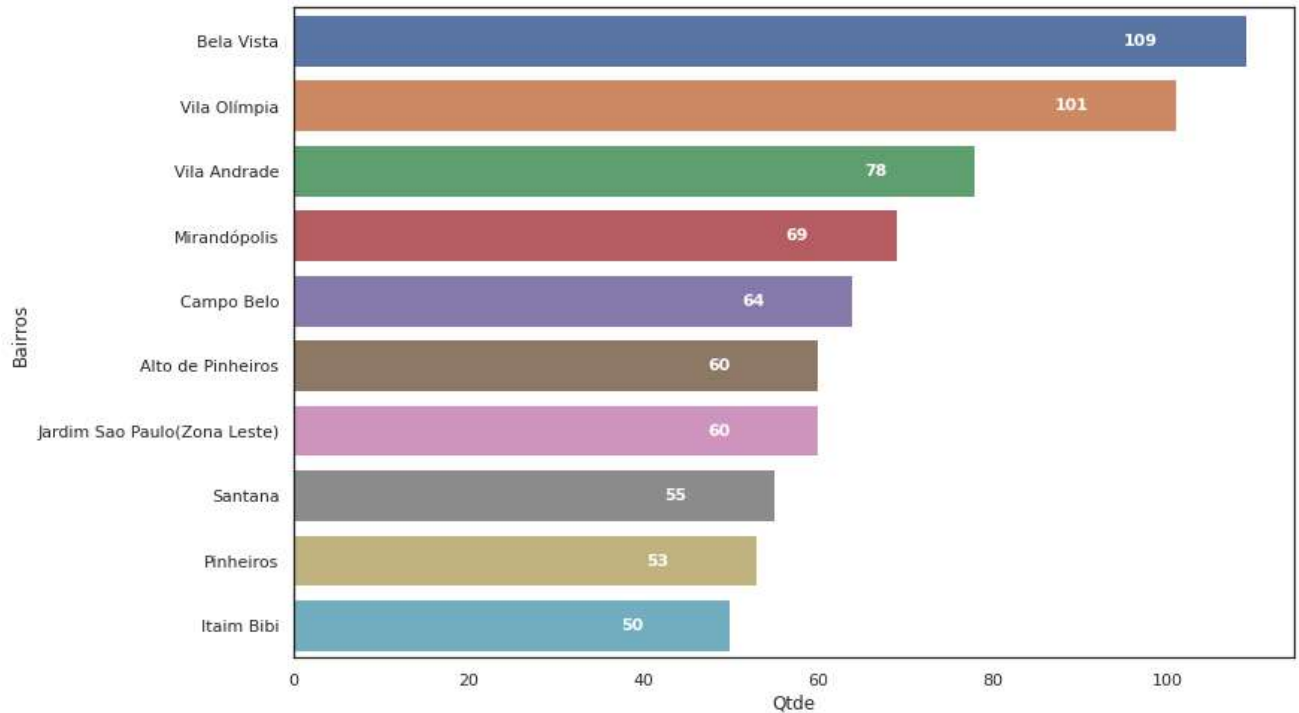
```
1 #sns.set(rc={'figure.figsize':(15,8)})
2 plt.figure(figsize=(12,8))
3 ax=sns.barplot(y=df_top10_aprov['Bairros'],x=df_top10_aprov['Qtde']);
4
5
6 #Colocando os rótulos de dados no gráfico
7 for i in range(len(df_top10_aprov)): #Varre-se todo o intervalo de iogos vendidos no
```



```

8  ..ax.text(df_top10_aprov['Qtde'][i]--10,.i,.
9  .....df_top10_aprov['Qtde'][i],.fontsize=11,.
10  .....color='white',verticalalignment='center',
11  .....horizontalalignment='right',weight='bold')

```



```

1 df_top10_bairros = pd.DataFrame(tb1['bairro'].value_counts().head(10))
2 df_top10_bairros.reset_index(inplace=True)
3 df_top10_bairros.rename(columns={'index': 'Bairros', 'bairro': 'Qtde'}, inplace=True)

```

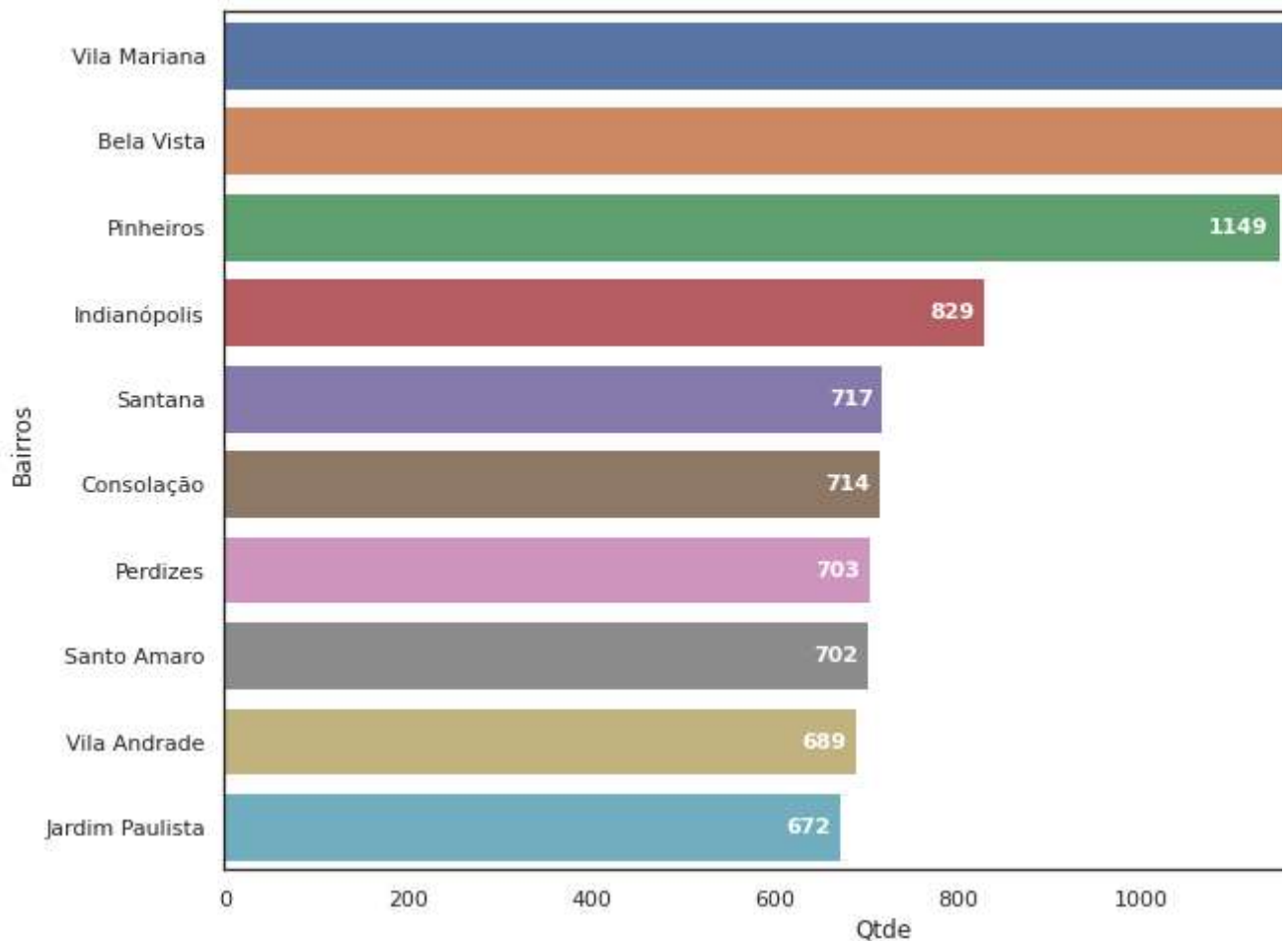
```
1 df_top10_bairros
```

	Bairros	Qtde
0	Vila Mariana	1369

```

1 plt.figure(figsize=(12,8))
2 ax1 = sns.barplot(y = df_top10_bairros['Bairros'], x = df_top10_bairros['Qtde']);
3 for i in range(len(df_top10_bairros)): # Varre-se todo o intervalo de jogos vendidos no
4     ax1.text(df_top10_bairros['Qtde'][i] - 10, i,
5             df_top10_bairros['Qtde'][i], fontsize=11,
6             color='white',verticalalignment='center',
7             horizontalalignment='right',weight='bold')

```



1

