



Impactos da
pandemia na
economia.

Análise dos seguintes índices:

➡ População

➡ Total de casos de covid

➡ Total de óbitos

➡ GDP per capita (PIB)

Questionamentos

O poder de compra dos
!7 brasileiros sobre a Cesta Básica
foi alterado?

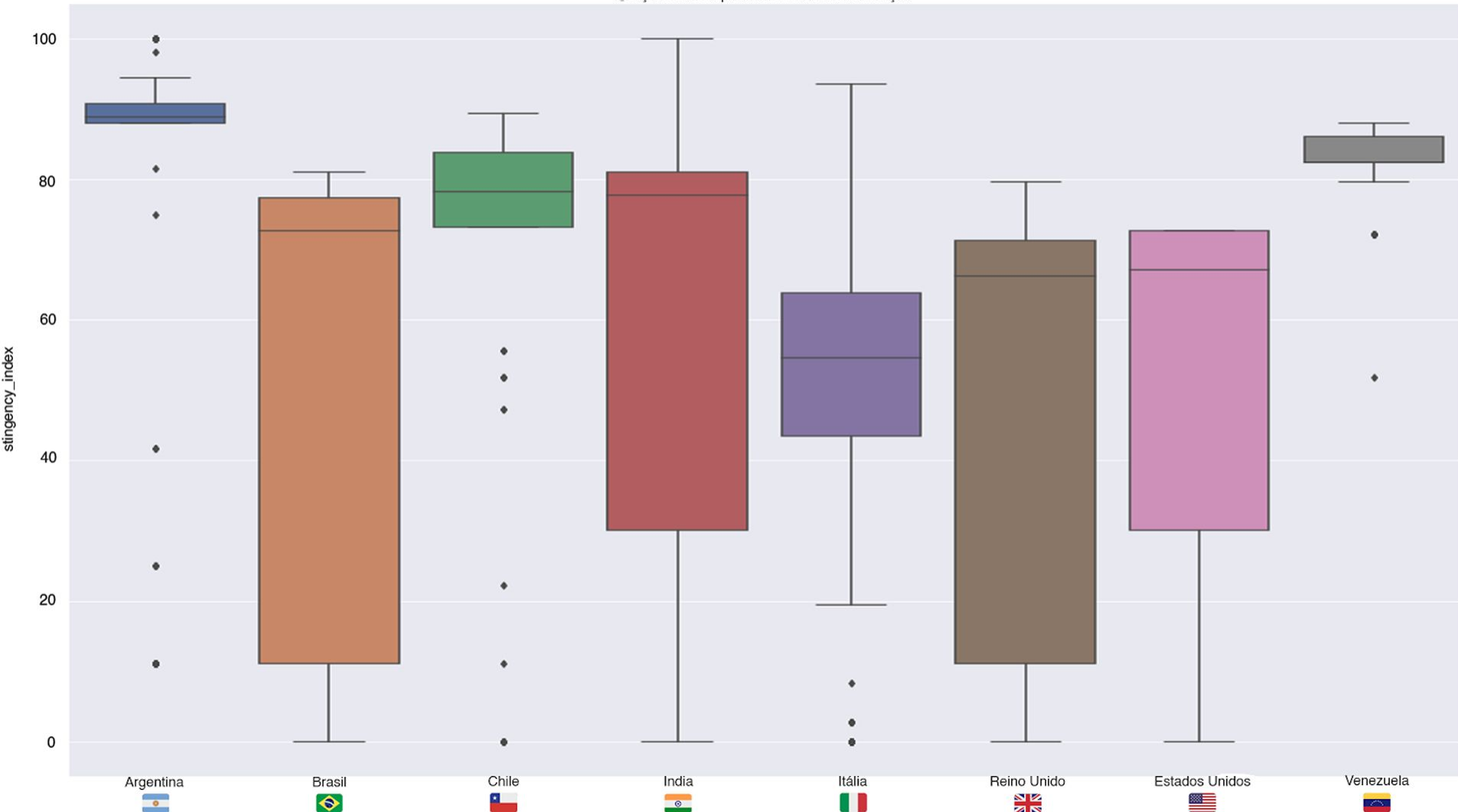
Como a pandemia afetou o Índice
!7 de confiança do consumidor e do
empresariado?

Como a pandemia afetou os
!7 diferentes setores da
economia?

Quais setores da economia
!7 tiveram impacto positivo ou
negativo durante a pandemia?

Impacto do covid-19 na economia

Relação entre os países e o índice de contenção



Código

6.O que representa o stringency_index em relação a cada país ?

```
In [70]: sns.set_style()
plt.figure(figsize=(39, 11))

aux1 = df_covid_raw2[(df_covid_raw2["location"] != "0") & (df_covid_raw2["stringency_index"] > 0)]
plt.subplot(1, 3, 1)
sns.boxplot(x="location", y="stringency_index", data=aux1)
plt.ylabel('stringency_index',fontsize = 20)
plt.title("Relação entre países e indicador de contenção",fontsize = 20)

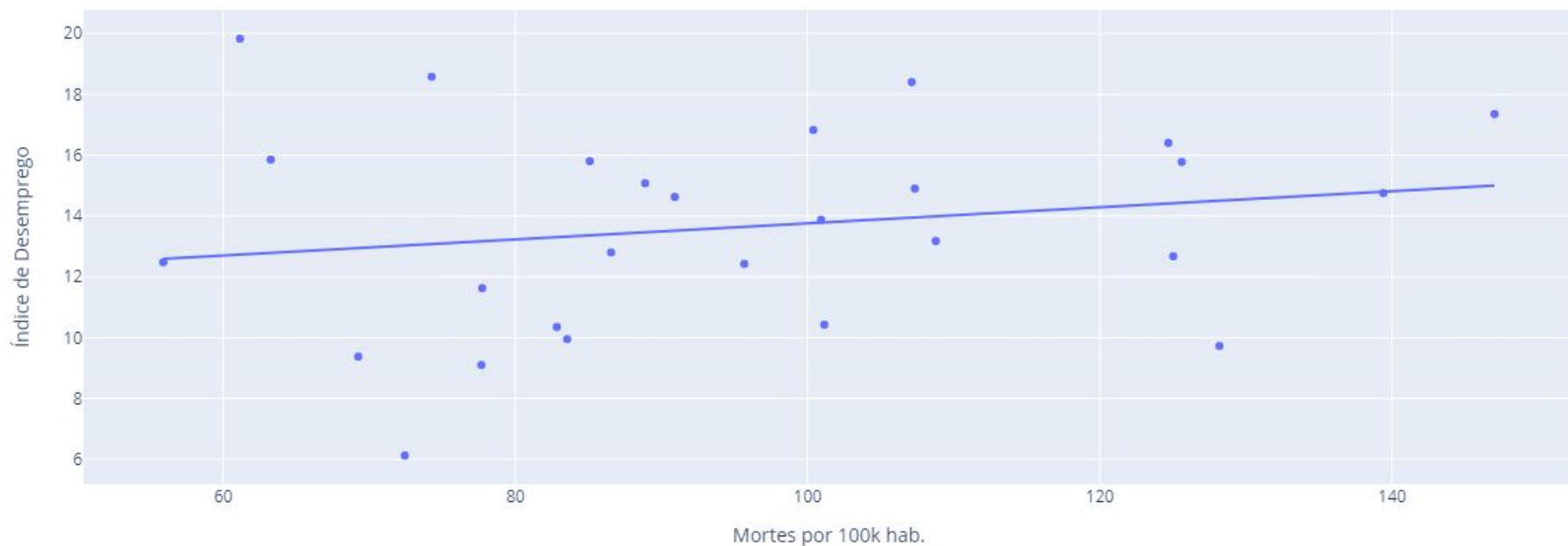
plt.xlabel('location',fontsize = 20)
```

Out[70]: Text(0.5, 0, 'location')

Mapa com panorama interativo



Mortes de COVID-19 por 100K hab. x Índice de Desemprego em 2020. Cada ponto é uma UF.



```
des_tot = pd.read_csv("Taxa de desocupação total.tsv", sep="\t")
des_tot.head()
```



		Brasil, Grande Região e Unidade da Federação	1º trimestre 2012	2º trimestre 2012	3º trimestre 2012	4º trimestre 2012	1º trimestre 2013	2º trimestre 2013	3º trimestre 2013	4º trimestre 2013	1º trimestre 2014	2º trimestre 2014	3º trimestre 2014	4º trimestre 2014	1º trimestre 2015
0	1	Brasil	7.9	7.5	7.1	6.9	8.0	7.4	6.9	6.2	7.2	6.8	6.8	6.5	7.9
1	1	Norte	8.9	8.2	7.8	7.3	8.6	8.3	7.6	6.5	7.7	7.2	6.9	6.8	8.7
2	2	Nordeste	9.7	9.6	9.3	9.3	10.9	10.0	8.9	7.9	9.3	8.8	8.6	8.2	9.5
3	3	Sudeste	7.9	7.4	6.9	6.6	7.6	7.2	7.0	6.2	7.0	6.9	6.9	6.6	8.0
4	4	Sul	5.1	4.8	4.3	4.0	4.8	4.3	4.1	3.8	4.4	4.1	4.2	3.8	5.1

```
[12] des_tot = des_tot.set_index("Brasil, Grande Região e Unidade da Federação")
des_tot = des_tot.transpose().reset_index()
des_tot = des_tot.drop(0)
des_tot = des_tot.rename_axis(None, axis=1)
des_tot = des_tot.rename(columns={"index": "Período"})
des_tot.head()
```

	Período	Brasil	Norte	Nordeste	Sudeste	Sul	Centro- Oeste	RO	AC	AM	RR	PA	AP	TO	MA	PI	CE	RN	PB	PE	AL	SE	BA	MG	ES	RJ	SP	PR
1	1º trimestre 2012	7.9	8.9	9.7	7.9	5.1	7.0	8.0	9.0	11.0	8.5	7.9	12.5	8.4	7.9	7.6	7.2	11.5	9.9	9.6	11.3	10.3	11.5	7.8	7.6	8.5	7.8	5.6
2	2º trimestre 2012	7.5	8.2	9.6	7.4	4.8	6.2	6.2	8.9	9.1	5.8	7.7	14.6	7.7	9.1	7.0	8.1	11.3	9.3	8.2	11.7	10.8	11.2	7.1	7.3	7.4	7.5	5.3
3	3º trimestre 2012	7.1	7.8	9.3	6.9	4.3	5.7	5.9	7.6	9.5	7.5	7.1	14.0	7.2	7.7	6.1	8.0	11.3	8.5	9.3	11.4	10.4	10.7	6.4	6.9	7.4	6.9	4.6
4	4º trimestre 2012	6.9	7.3	9.3	6.6	4.0	5.7	5.3	8.1	8.4	8.4	6.8	11.8	7.6	7.5	6.9	7.5	11.5	9.0	9.2	11.0	9.5	10.8	6.2	6.7	6.8	6.8	4.3
5	1º trimestre 2013	8.0	8.6	10.9	7.6	4.8	6.8	6.1	10.8	10.2	8.8	7.8	11.4	9.3	9.3	8.3	8.8	12.1	9.4	10.6	12.1	11.4	13.2	7.4	7.8	7.2	7.7	4.9

```
[13] des_2020 = des_tot.iloc[-4:, 7:].mean()
des_2020
```

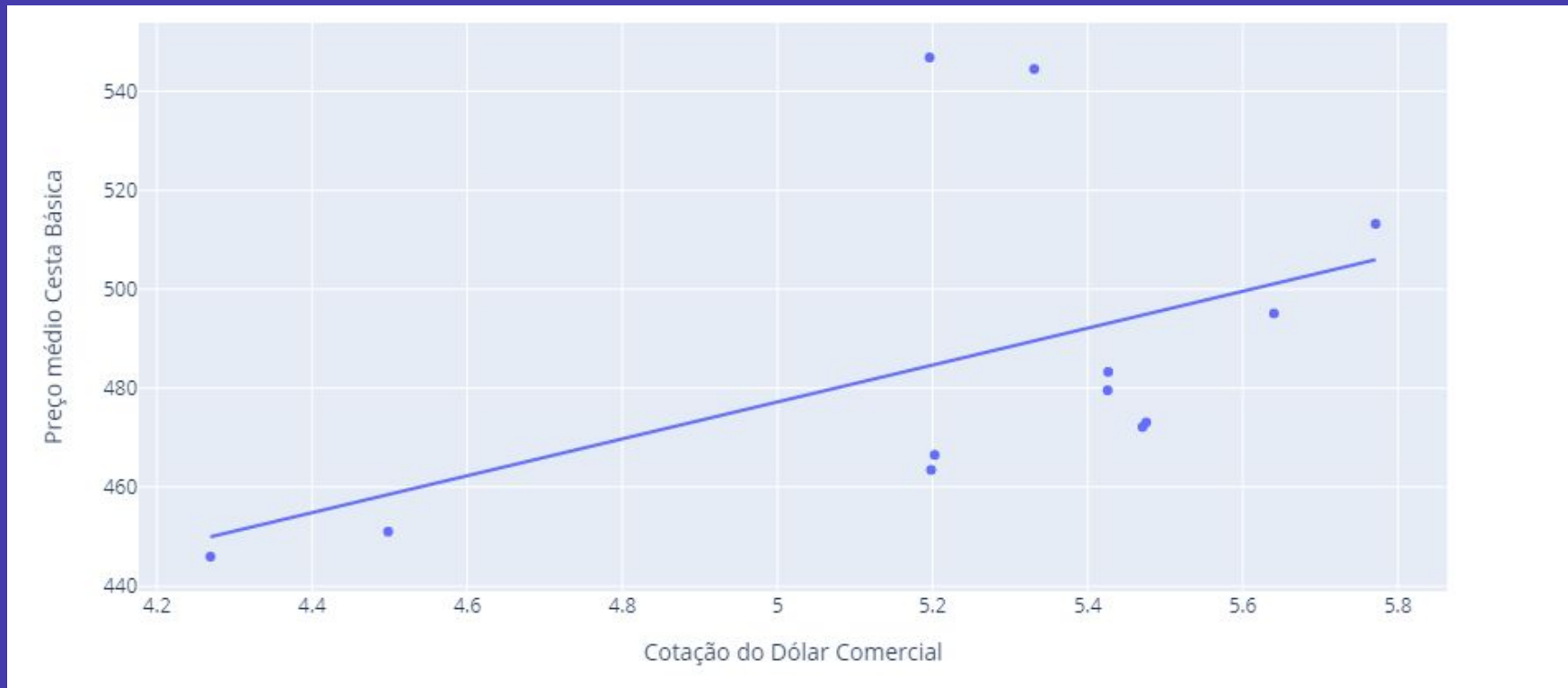


```
brasilio_util = brasilio.loc[:, ["state", "confirmed", "deaths", "estimated_population", "confirmed_per_100k_inhabitants"]]  
brasilio_util = brasilio_util.rename({"confirmed_per_100k_inhabitants": "Casos por 100k hab."}, axis=1)  
brasilio_util["Mortes por 100k hab."] = (brasilio_util.deaths*100000)/brasilio_util.estimated_population  
merged = brasilio_util.set_index("state").join(des_2020.rename("Índice de Desemprego")).reset_index()  
merged
```



	state	confirmed	deaths	estimated_population	Casos por 100k hab.	Mortes por 100k hab.	Índice de Desemprego
0	AC	41620	795	894470	4653.03476	88.879448	15.075
1	AL	104818	2489	3351543	3127.45503	74.264302	18.575
2	AM	201013	5285	4207714	4777.24959	125.602643	15.775
3	AP	68201	925	861773	7914.03305	107.336851	14.900
4	BA	493400	9129	14930634	3304.61520	61.142749	19.825
5	CE	335992	9993	9187103	3657.21381	108.772047	13.175
6	DF	251701	4259	3055149	8238.58345	139.404003	14.750
7	ES	248251	5081	4064052	6108.46023	125.023007	12.675
8	GO	309110	6805	7113540	4345.37516	95.662638	12.425
9	MA	200938	4500	7114598	2824.30574	63.250236	15.850
10	MG	542909	11902	21292666	2549.74647	55.897181	12.475
11	MS	134750	2347	2809394	4796.40805	83.541148	9.950
12	MT	180451	4520	3526220	5117.40617	128.182586	9.725
13	PA	262222	7122	8222715	3202.22277	86.222212	12.250

Correlação entre a cotação média do dólar comercial e o preço médio da Cesta Básica em 2020. Cada ponto é um mês.



Cotação do dólar disponível em <http://yahii.com.br/dolar.html>. Preço Cesta Básica disponível em <https://www.dieese.org.br/cesta/>

```
In [2]: > url = "http://www.yahii.com.br/dolar.html"
page = requests.get(url)
soup = BeautifulSoup(page.content, "html.parser")
```

```
In [3]: > table = soup.find("table", attrs={"cellspacing":0, "bordercolordark": "#ffffff", "cellpadding": "2", "width":"850"}).tbody
linhas = table.find_all("tr")[:-1]
```

```
In [4]: > linhas[-1].text
```

```
Out[4]: '\n2021\n5,4753\n5,5296\n5,6967\n5,4030\n5,2316\n\n\xa0\n\xa0\n\xa0\n\xa0\n\xa0\n\xa0\n\xa0\n'
```

```
In [5]: > import re
def formatador_linhas(row):
    row = re.sub("\n", "\t", row)
    row = re.sub(",", ".", row)
    row = re.sub("\xa0", "", row)
    row = re.sub("\t", "", row, 1)
    return row
```

```
In [6]: > all_text = ""
for e in linhas[:-1]:
    all_text += formatador_linhas(e.text) + "\n"
all_text += formatador_linhas(linhas[-1].text)
```

```
In [7]: > arquivo = open("dolar_por_mes.tsv", "w")
arquivo.write(all_text)
arquivo.close()
```

```
In [8]: > df = pd.read_csv("dolar_por_mes.tsv", sep="\t").iloc[-5:, :].loc[2020, :]
df = df.astype(float)
df.ANO = df.ANO.astype(int)
```

```
In [9]: > df_2020 = df.loc[df.ANO == 2020]
df_2020
```

```
Out[9]:
```

	ANO	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
50	2020	4.2689	4.4981	5.1981	5.4264	5.4257	5.4754	5.2027	5.4707	5.6401	5.7712	5.3311	5.1961

```
In [21]: cesta_basica = pd.read_csv("cesta_basica.csv").dropna(axis=1).rename({"Unnamed: 0": "Período"}, axis=1)
cesta_basica.head()
#17 capitais
```

Out[21]:

	Período	Brasília	Campo Grande	Goiânia	Belo Horizonte	Rio de Janeiro	São Paulo	Vitória	Curitiba	Florianópolis	Porto Alegre	Belém	Aracaju	Fortaleza	João Pessoa	Natal	Recife
0	01-2020	483.17	458.00	455.08	456.35	507.13	517.51	492.20	452.32	489.13	502.98	415.56	368.69	433.39	388.02	389.26	395.12
1	02-2020	481.78	445.40	448.62	459.10	505.55	519.76	480.03	447.91	493.15	492.83	432.95	371.22	462.99	403.98	405.88	420.15
2	03-2020	484.15	474.53	456.18	461.28	533.65	518.50	504.78	465.47	517.13	497.88	418.80	390.20	475.11	414.05	426.00	433.12
3	04-2020	471.17	495.69	493.21	484.55	544.34	556.25	537.89	511.23	523.80	527.01	434.19	401.37	482.46	437.93	428.28	461.12
4	05-2020	441.09	455.35	505.99	484.02	558.81	556.36	536.73	531.27	524.07	518.63	453.36	400.15	455.18	440.25	429.57	451.12

```
In [22]: #Media da cesta básica nas capitais
cesta_basica = cesta_basica.set_index("Período").mean(axis=1).reset_index().rename({0:"Preço médio Cesta Básica"}, axis=1)
cesta_basica.head()
```

Out[22]:

	Período	Preço médio Cesta Básica
0	01-2020	445.953529
1	02-2020	450.994706
2	03-2020	463.473529
3	04-2020	483.279412
4	05-2020	479.565294

```
In [23]: df_2020_mod = df_2020.T.drop("ANO").rename({50:"Cotação do Dólar Comercial"}, axis=1).reset_index(drop=True)
df_2020_mod.head()
```

Out[23]:

	Cotação do Dólar Comercial
0	4.2689
1	4.4981
2	5.1981
3	5.4264
4	5.4257

```
In [24]: temp = cesta_basica.join(df_2020_mod)
```

Comportamento da confiança sobre as indústrias brasileiras

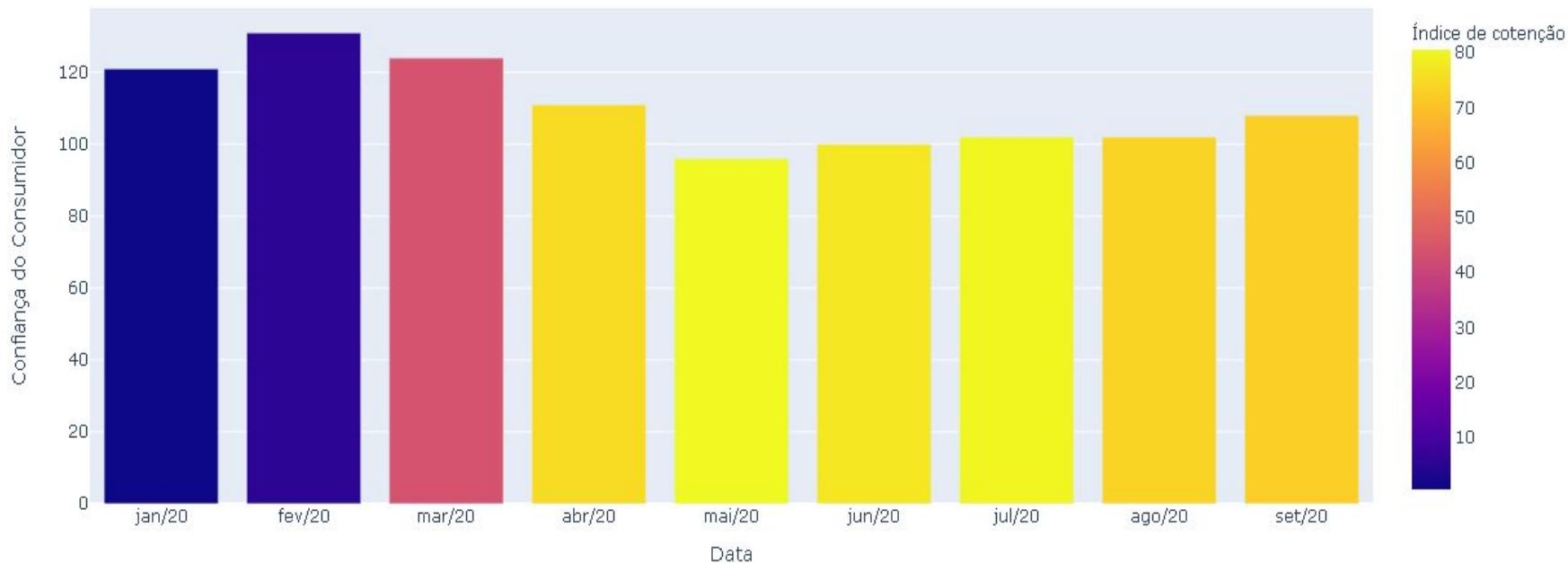
- Início de barreiras de contenção da pandemia
- Impactos da pandemia na construção civil, comércio, e indústria
- Índice de confiança do empresariado e do consumidor

fontes: IBGE, CNI



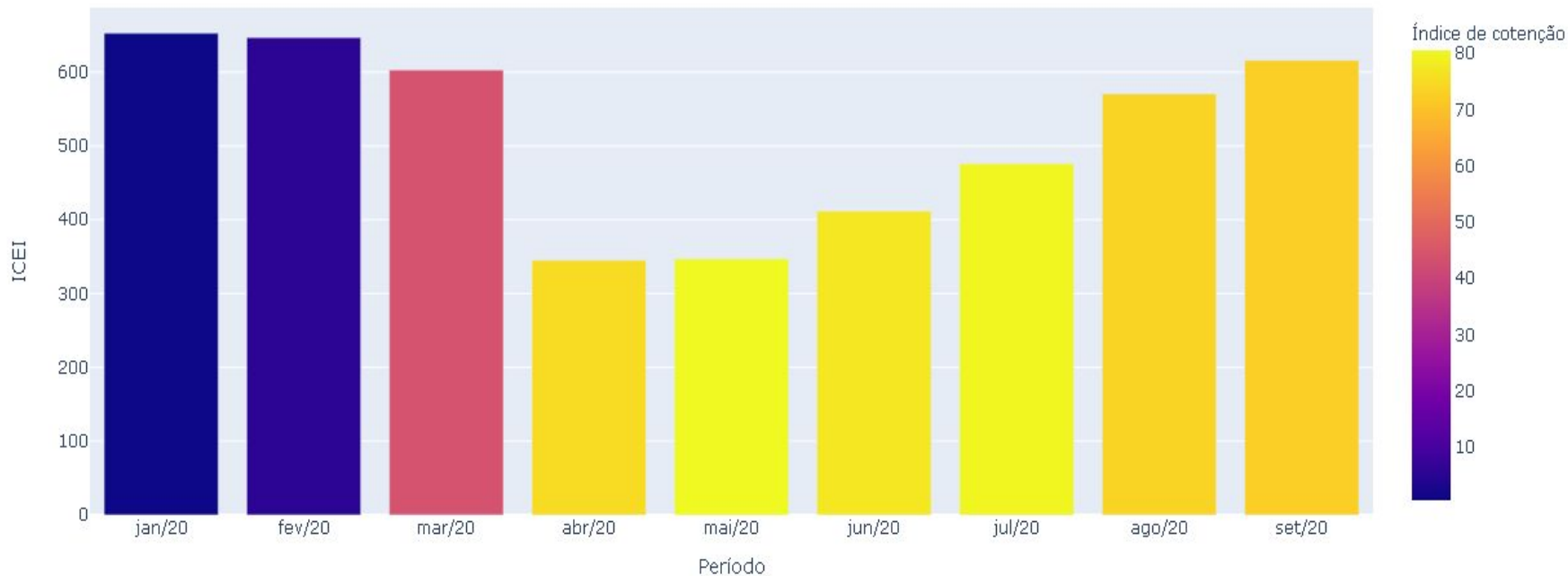
Confiança do consumidor e índice de contenção

Relação entre índice de confiança do consumidor e o índice de contenção



Expectativa do desempenho industrial e o índice de contenção

Relação entre índice de confiança do empresariado sobre o desempenho industrial e o índice de contenção



```
1 raw = '/content/raw_data.csv'
2 df4 = pd.read_csv(raw)
3 df4
```

```
[98] 1 df4 = df4.loc[:, ["location", "date", "stringency_index"]]
     2 df4
```

```
[99] 1 df5 = df4.loc[(df4.location=="Brazil")].dropna()
     2 df5
```

```
[100] 1 def desconsiderar_dia(row):
     2     row.date = row.date[:-3]
     3     return row
     4
     5 df5 = df5.apply(desconsiderar_dia, axis='columns')
     6 df5
```

```
[101] 1 df5 = df5.groupby(["date"]).stringency_index.mean().reset_index()
     2 df5
```

```
1 df_junto = df.join(df5).drop("date", axis=1)
2 df_junto = df_junto.rename({"stringency_index": "Índice de cotenção"}, axis=1)
3 df_junto
```

	Data	Confiança do Consumidor	Índice de cotenção
0	jan/20	121	0.538065
1	fev/20	131	5.560000
2	mar/20	124	43.996452
3	abr/20	111	75.001667
4	mai/20	96	80.541290
5	jun/20	100	77.310000
6	jul/20	102	79.974516
7	ago/20	102	73.525484
8	set/20	108	72.690000

```
1 fig_7 = px.bar(df_junto, x = 'Data', y = 'Confiança do Consumidor', color = 'Índice de cotenção', title = 'Relação entre
2 fig_7.show()
```

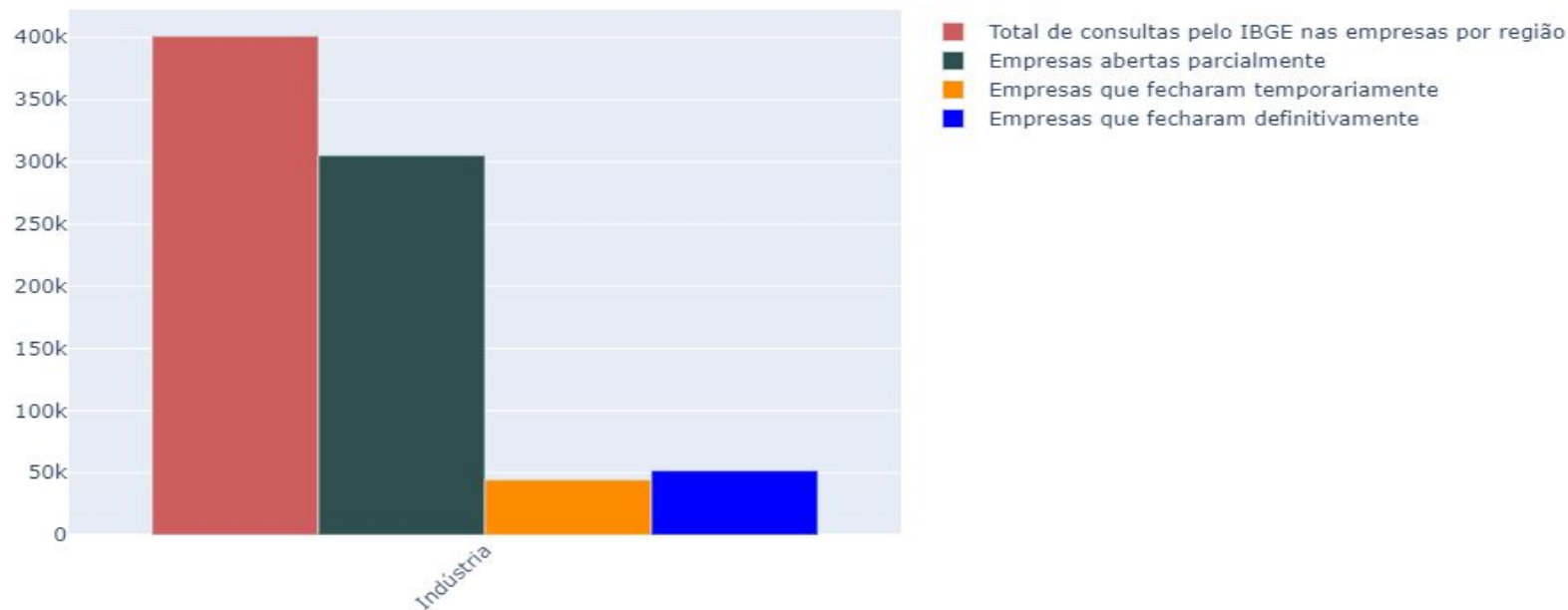

Situação das indústrias durante a pandemia

- Redução nas vendas e na produção
- Dificuldade para a adquirirem insumos e matérias-primas
- Fechamento temporários e permanentes das indústrias
- Fechamento de empresas multinacionais (Ford,Audi,Sony)

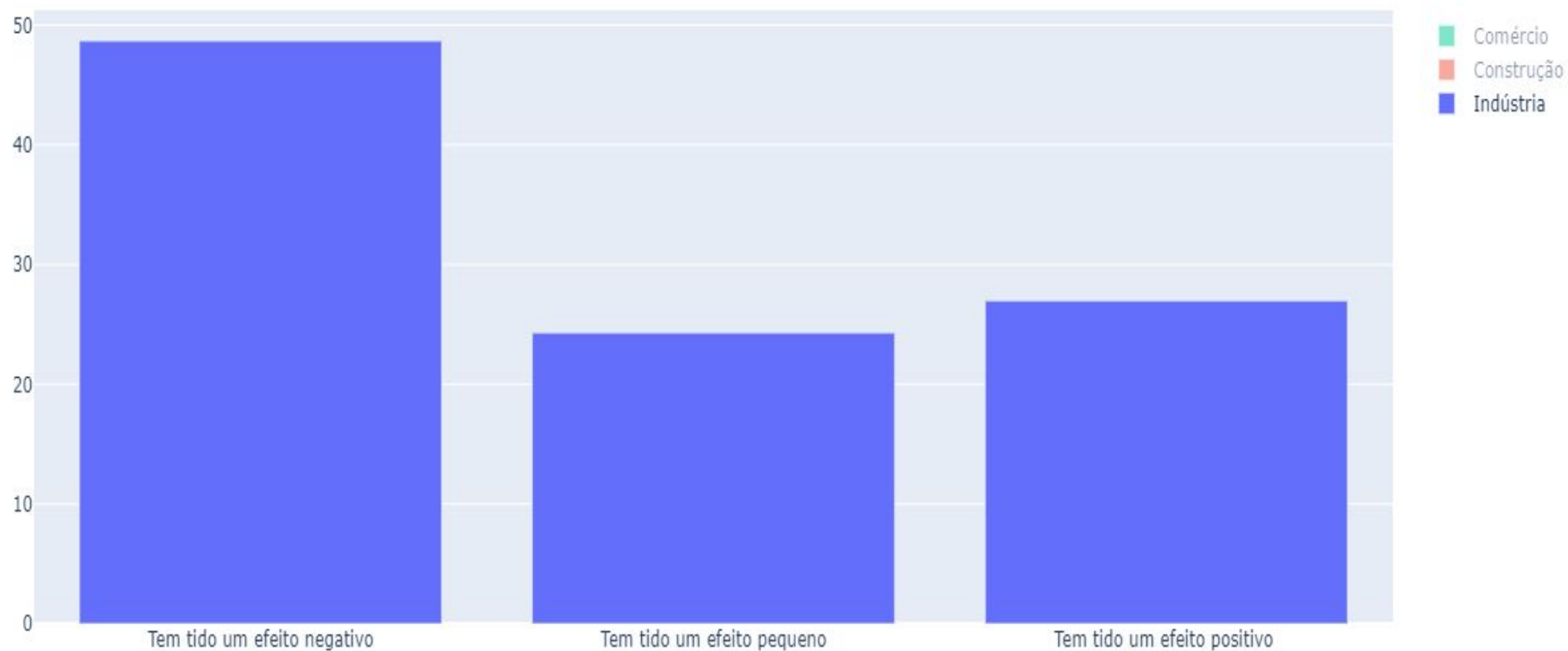


Funcionamento das indústrias

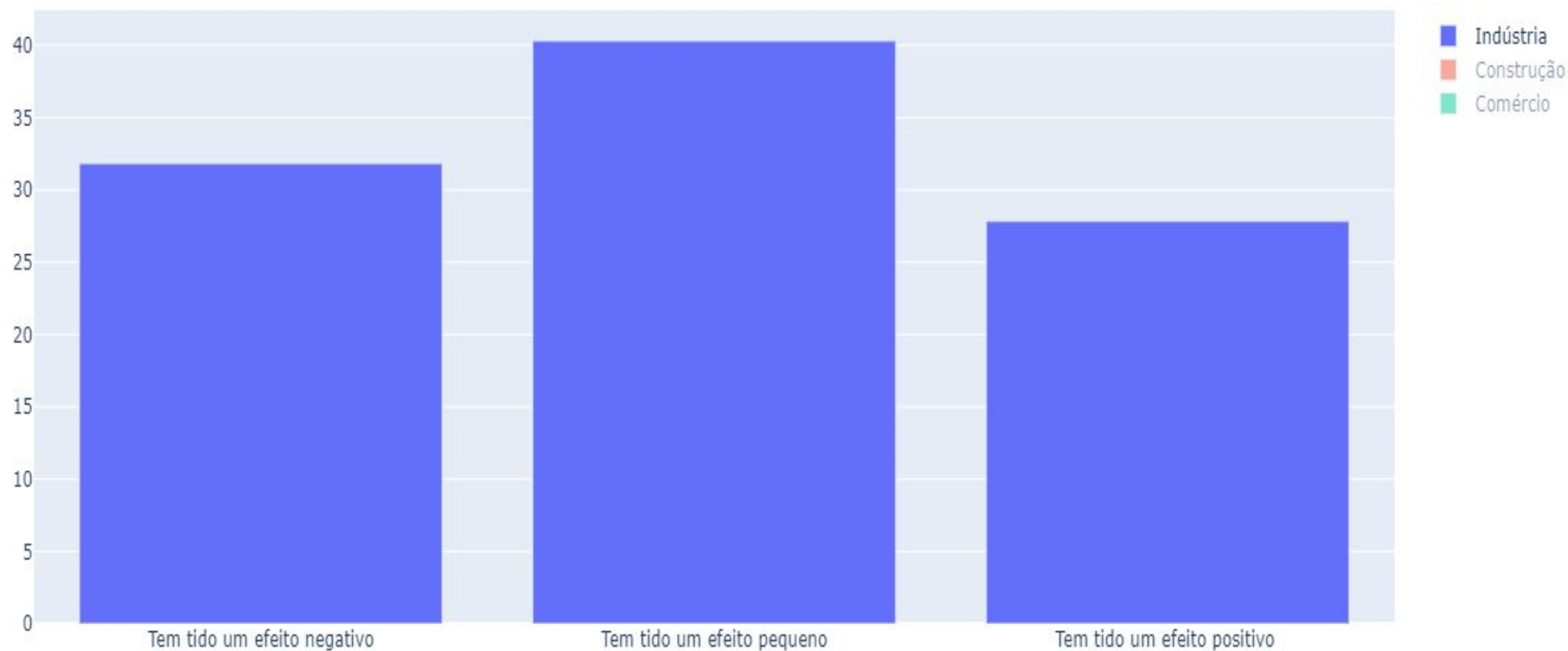
Situação da indústria na pandemia



Pesquisa efeito positivo ou negativo na indústria na primeira quinzena de julho de 2020



Pesquisa efeito positivo ou negativo na indústria na primeira quinzena de setembro de 2020



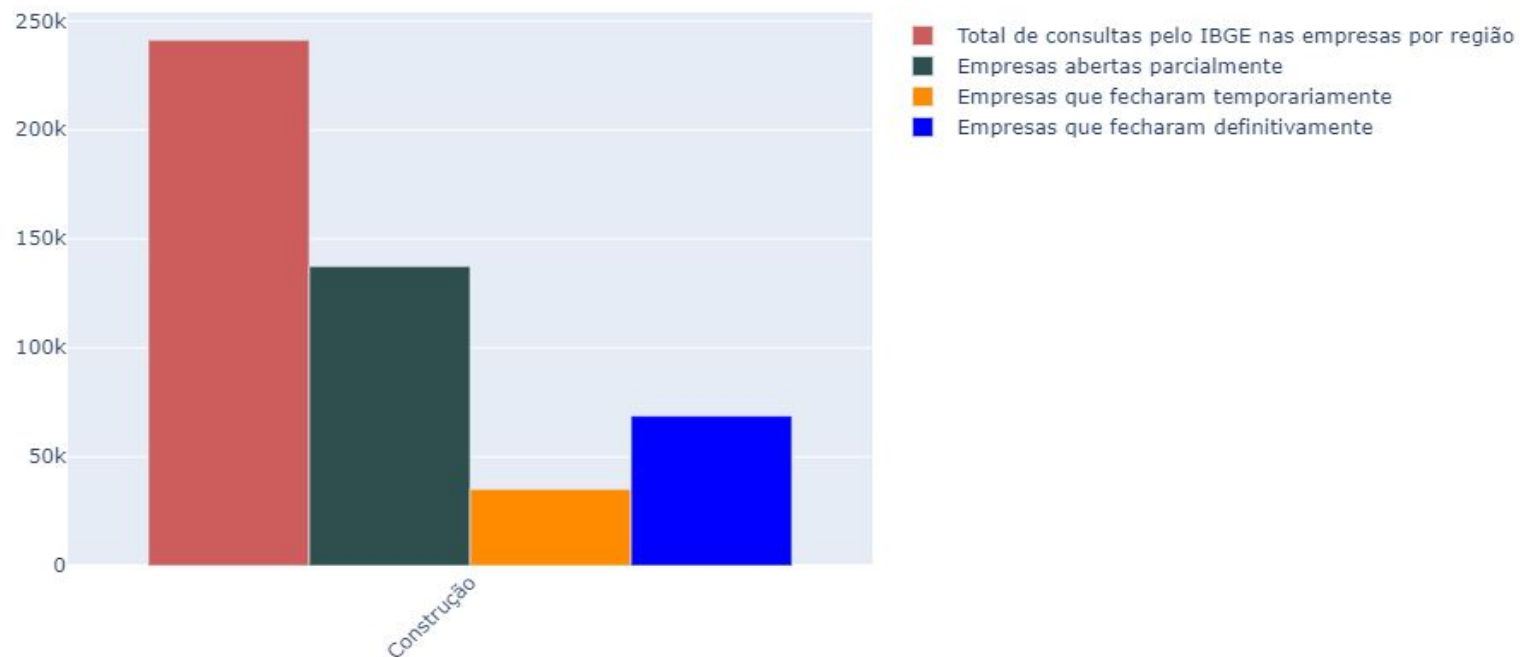
Construção civil e seu crescimento

- Setor da economia menos impactado
- Aquecimento do setor durante a pandemia
- Preocupação com a desaceleração do setor pós pandemia
- Crescimento das empresas

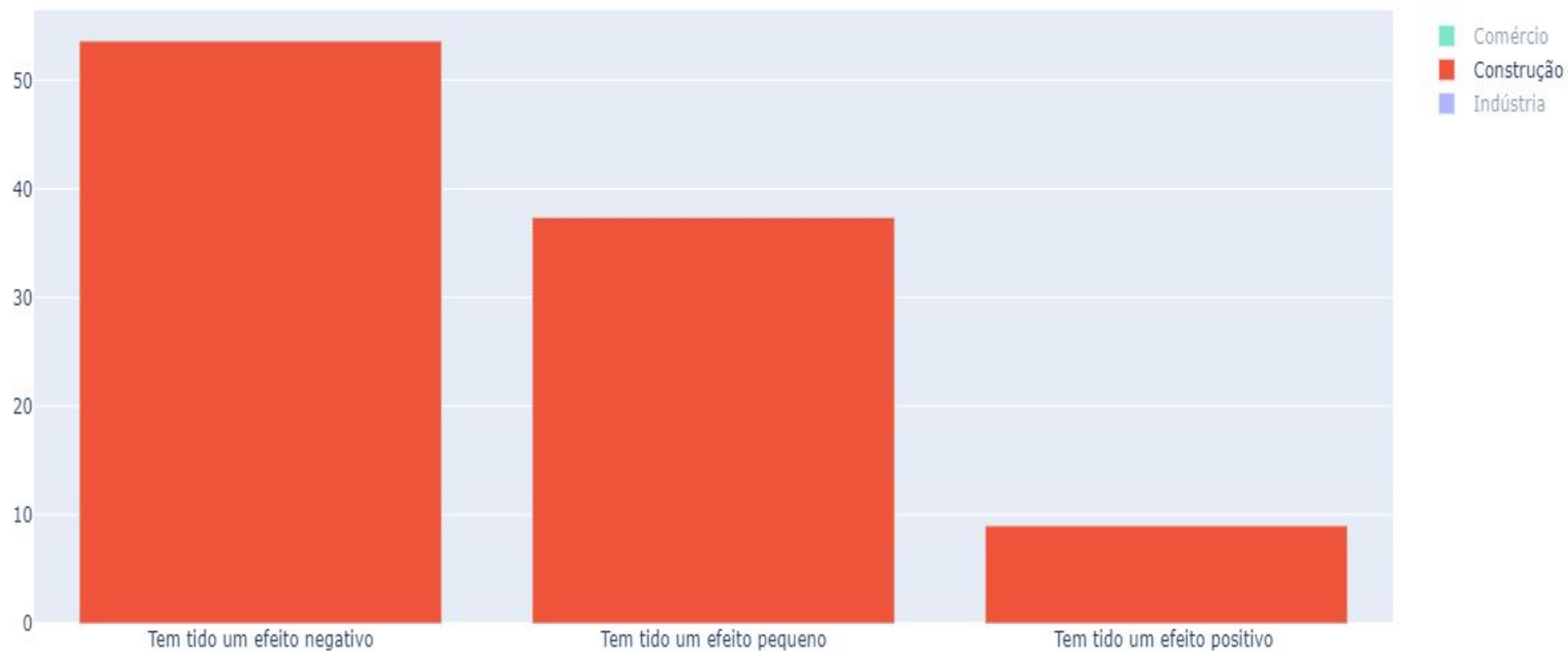


Funcionamento da construção civil

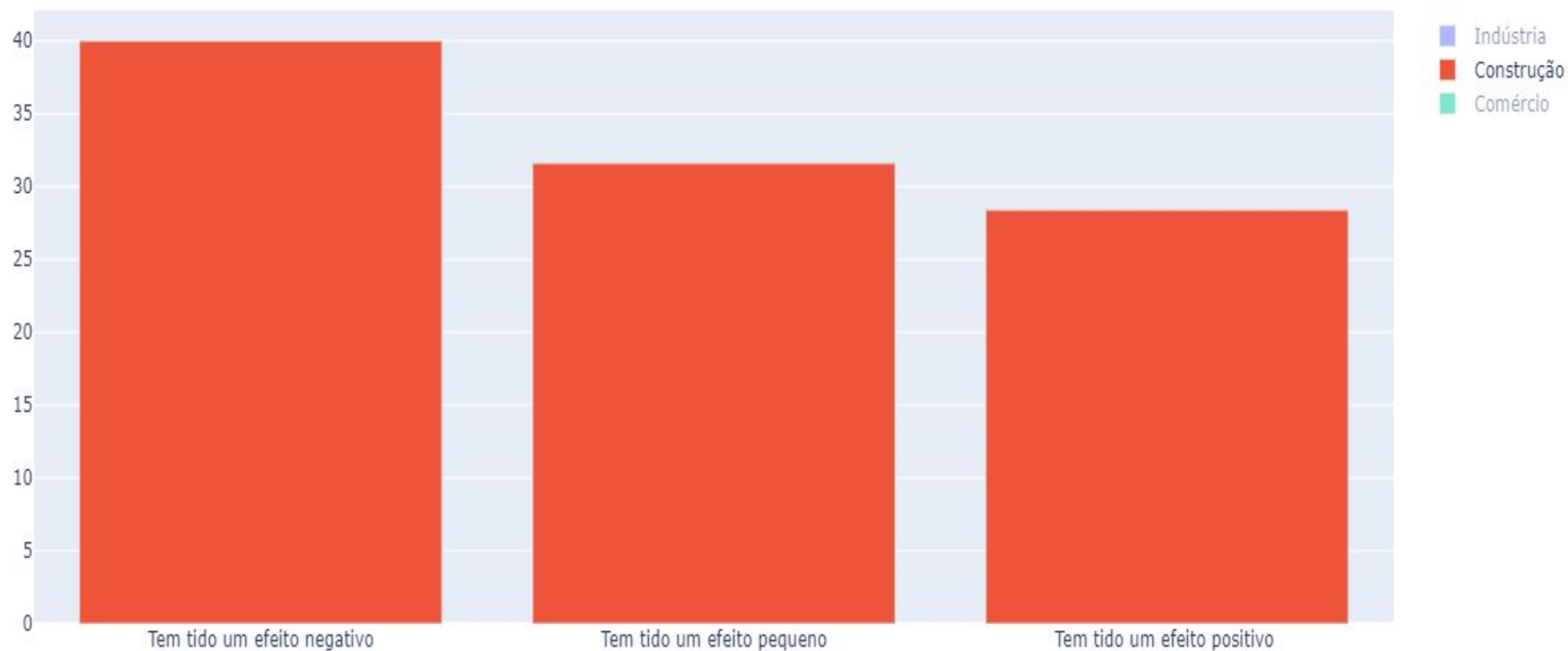
Situação da Construção na pandemia



Pesquisa efeito positivo ou negativo na construção na primeira quinzena de julho de 2020



Pesquisa efeito positivo ou negativo na construção na primeira quinzena de setembro de 2020



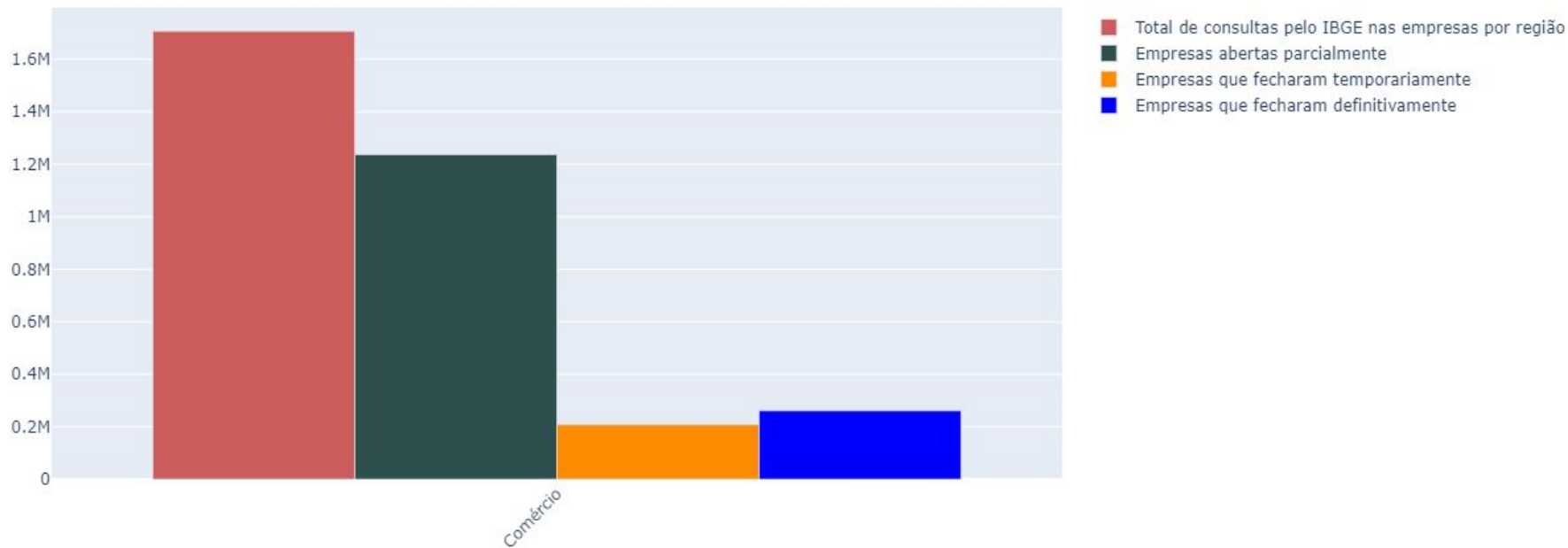
Comércio e os impactos sofridos

- Setor que mais sofreu e está sofrendo durante a pandemia
- Apresentou prejuízo de 29,5% na primeira quinzena de julho e de 44,5 na primeira quinzena de agosto de 2020
- Fechamento de 75 mil lojas desde o início da pandemia
- Cerca de 2 milhões de pessoas perderam seus empregos por conta do fechamento do comércio

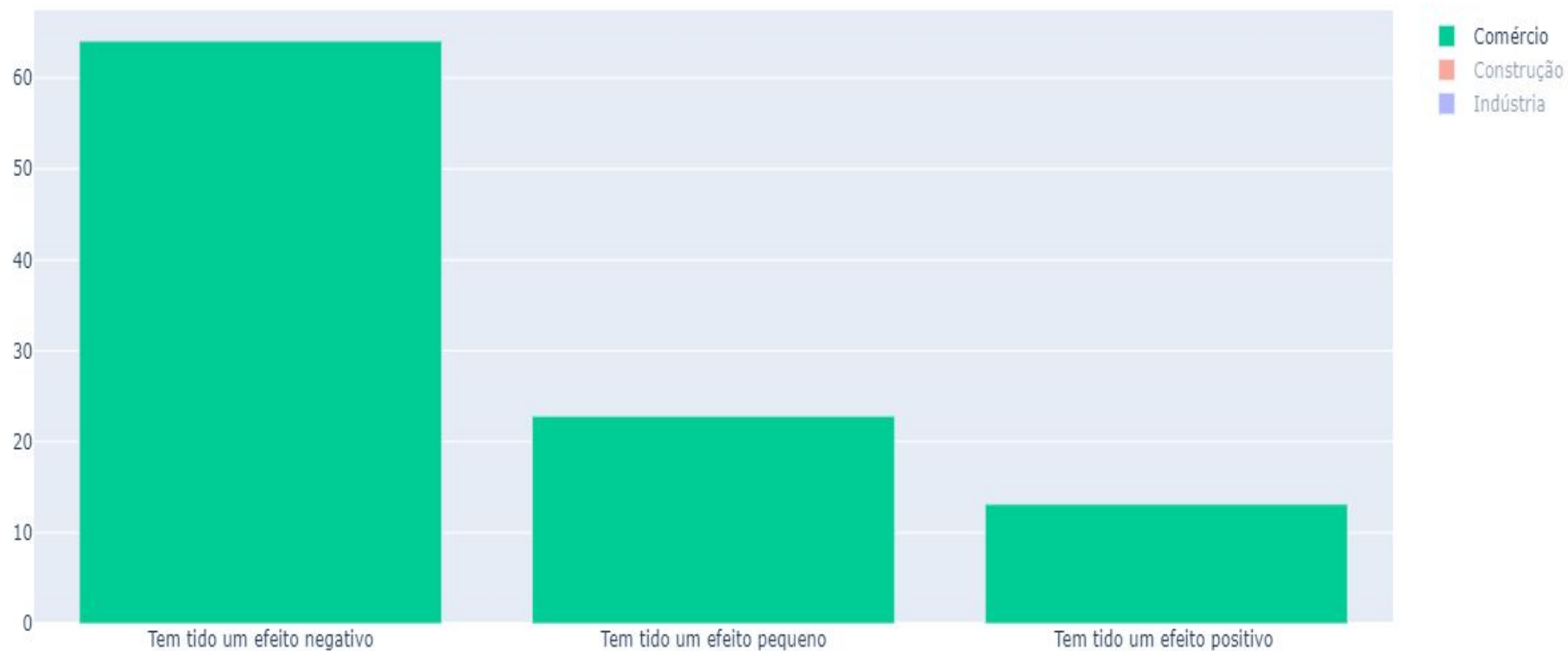


Comércios fechados temporariamente e permanentemente

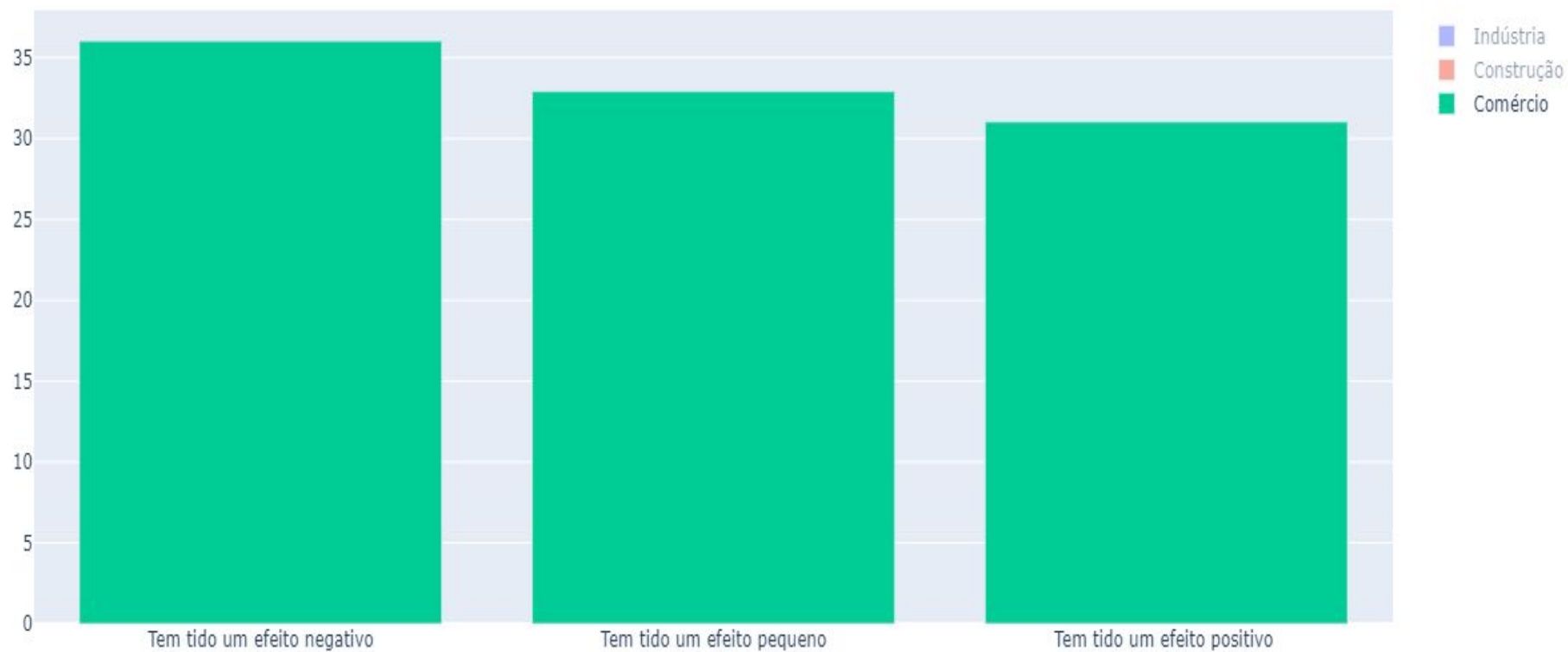
Situação do Comércio na pandemia



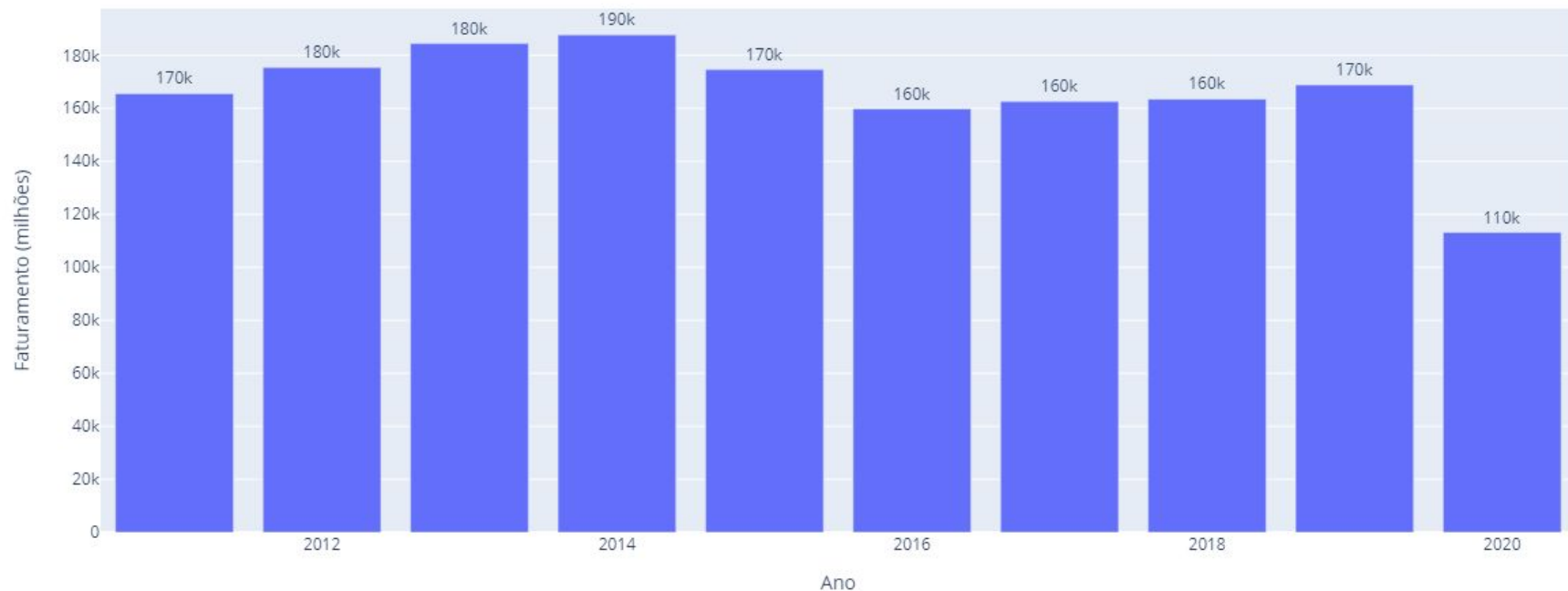
Pesquisa efeito positivo ou negativo no comércio na primeira quinzena de julho de 2020



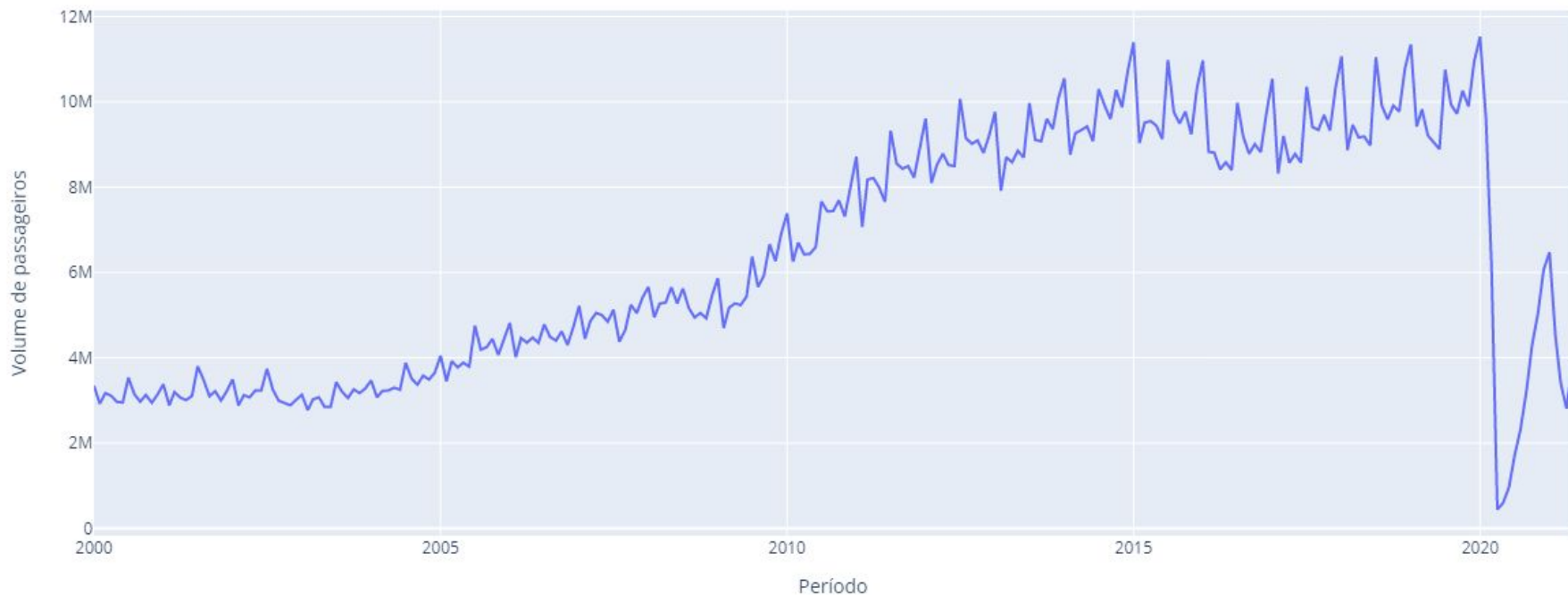
Pesquisa efeito positivo ou negativo no comércio na primeira quinzena de setembro de 2020



O faturamento do setor de Turismo

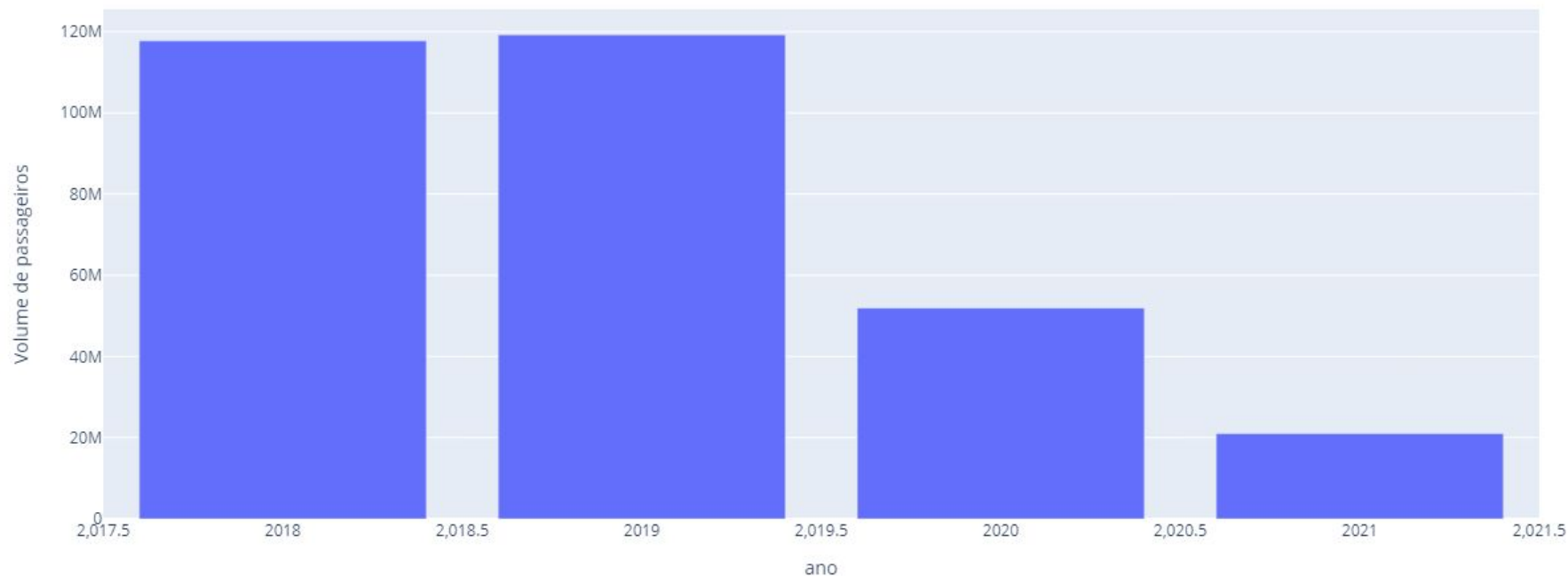


O fluxo de passageiros na aviação civil



Fonte: ANAC

O fluxo de passageiros na aviação civil



Fonte: ANAC

```
[89] aviac = pd.read_csv("Demanda_e_Oferta05.csv", sep=";")
      aviac.head()
```

	ano	mes	empresa_nome	empresa_sigla	empresa_pais	natureza	passageiros_pg	carga_correio_kg	rpk	ask
0	2000	1	AMERICAN AIRLINES, INC.	AAL	ESTADOS UNIDOS DA AMÉRICA	INTERNACIONAL	65102.0	2803329.0	419939104.0	667155136.0
1	2000	1	ABATÉ LINHAS AÉREAS S.A.	ABJ	BRASIL	DOMÉSTICA	428.0	1557.0	233327.0	725085.0
2	2000	1	AIR CANADA	ACA	CANADÁ	INTERNACIONAL	163.0	838.0	1335622.0	1679770.0
3	2000	1	AEROFLOT-AEROLÍNEAS INTERNACIONAIS DA RÚSSIA	AFL	RÚSSIA	INTERNACIONAL	1634.0	6497.0	15006158.0	22362600.0
4	2000	1	SOCIÉTÉ AIR FRANCE	AFR	FRANÇA	INTERNACIONAL	18048.0	1470274.0	168573888.0	217613136.0

```
[90] aviac_tot = aviac.groupby(["ano", "mes"]).passageiros_pg.sum().reset_index()
      aviac_tot.head()
```

```

▶ aviac_tot.ano = aviac_tot.ano.astype(int)
  aviac_tot["Período"] = aviac_tot["ano"].astype(str) + "-" + aviac_tot["mes"].astype(str)
  aviac_tot.passageiros_pg = aviac_tot.passageiros_pg.astype(int)
  aviac_tot
```

```
[93] aviac_tot_ano = aviac_tot.groupby(["ano"]).passageiros_pg.sum().reset_index()
      aviac_tot_ano
```