

Obtenção de insights dos dados do Portal da Transparência através EDA

William Sanches



Contextualização



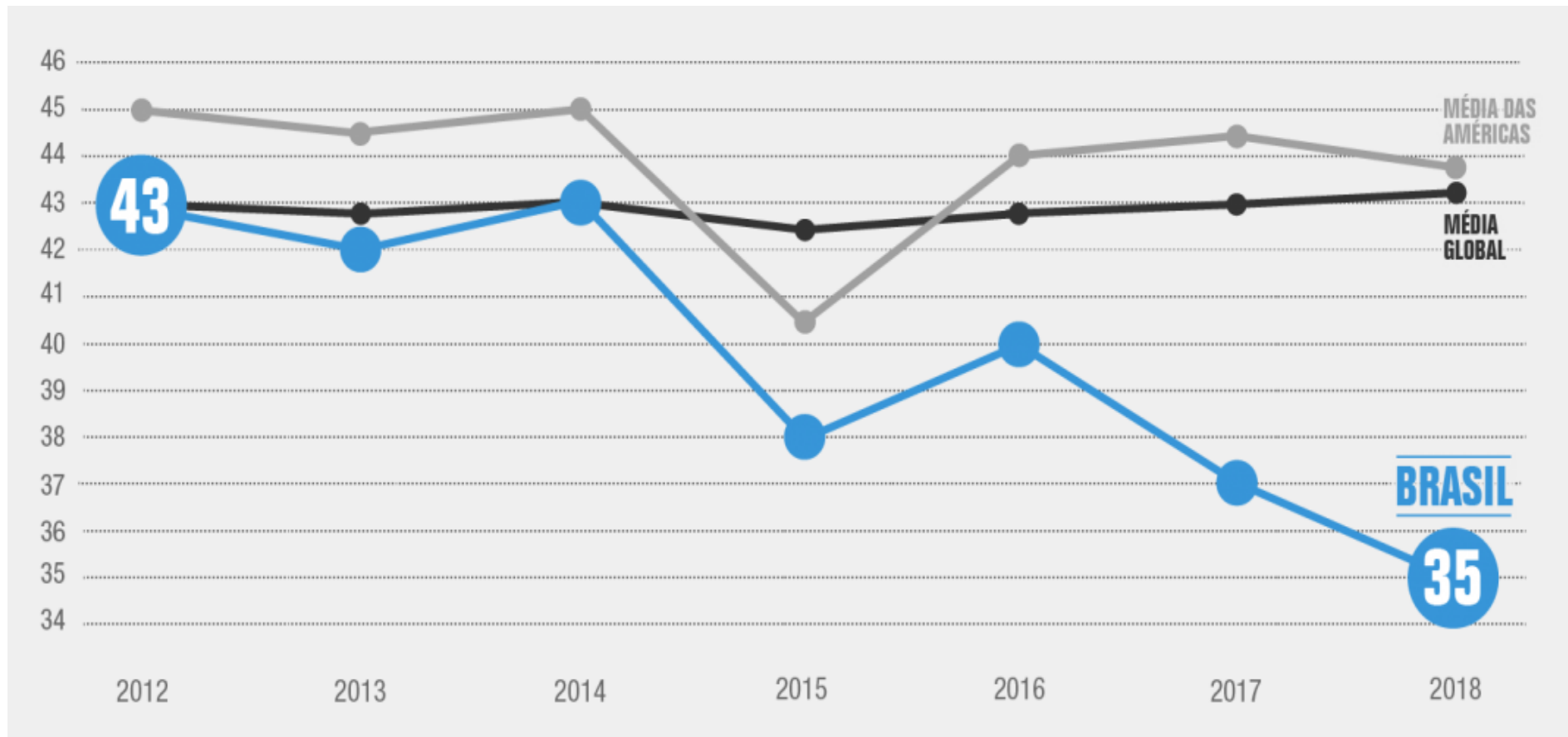
Contextualização

- Brasil vem caindo no IPC desde 2012



<https://transparenciainternacional.org.br/ipc/>

Contextualização



<https://transparenciainternacional.org.br/ipc/>

Definição do problema



Definição do problema

- Corrupção afeta os investimentos no país
- CGU => Portal da Transparência
- Os dados analisados são do governo federal
- Os dados possuem abrangência nacional



Objetivo



Objetivo

- Avaliar eficácia da aplicação das técnicas de EDA na obtenção dos insights



Coleta dos dados



Coleta dos dados

- API do Portal da Transparência



←	→	↻	🏠	🔒 Not secure	transparencia.gov.br/swagger-ui.html#/Cadastro32Nacional32de32Empresas32Inid244neas32e32Suspensas3240CEIS41	☆	👤
Benefício de Prestação Continuada (BPC)					Mostrar/Esconder	Listar operações	Expandir operações
Bolsa Família					Mostrar/Esconder	Listar operações	Expandir operações
Cadastro de Expulsões da Administração Federal (CEAF)					Mostrar/Esconder	Listar operações	Expandir operações
Cadastro Nacional de Empresas Inidôneas e Suspensas (CEIS)					Mostrar/Esconder	Listar operações	Expandir operações
GET	/api-de-dados/ceis				Consulta os registros do CEIS por CNPJ ou CPF Sancionado/Órgão Sancionador/Período		
GET	/api-de-dados/ceis/{id}				Consulta um registro do CEIS pelo id		
Cadastro Nacional de Empresas Punidas (CNEP)					Mostrar/Esconder	Listar operações	Expandir operações
Contratos do Poder Executivo Federal					Mostrar/Esconder	Listar operações	Expandir operações
Convênios do Poder Executivo Federal					Mostrar/Esconder	Listar operações	Expandir operações
Despesas Públicas					Mostrar/Esconder	Listar operações	Expandir operações
Emendas parlamentares					Mostrar/Esconder	Listar operações	Expandir operações

Coleta dos dados

- Python 3.8.2
- Biblioteca requests para as APIs
- MongoDB para persistência dos dados
- Biblioteca pymongo para operações com o MongoDB



Processamento e tratamento dos dados



Processamento e tratamento dos dados

- Utilizada biblioteca `flatten_json` retornando um `pandas.DataFrame` “achatado”

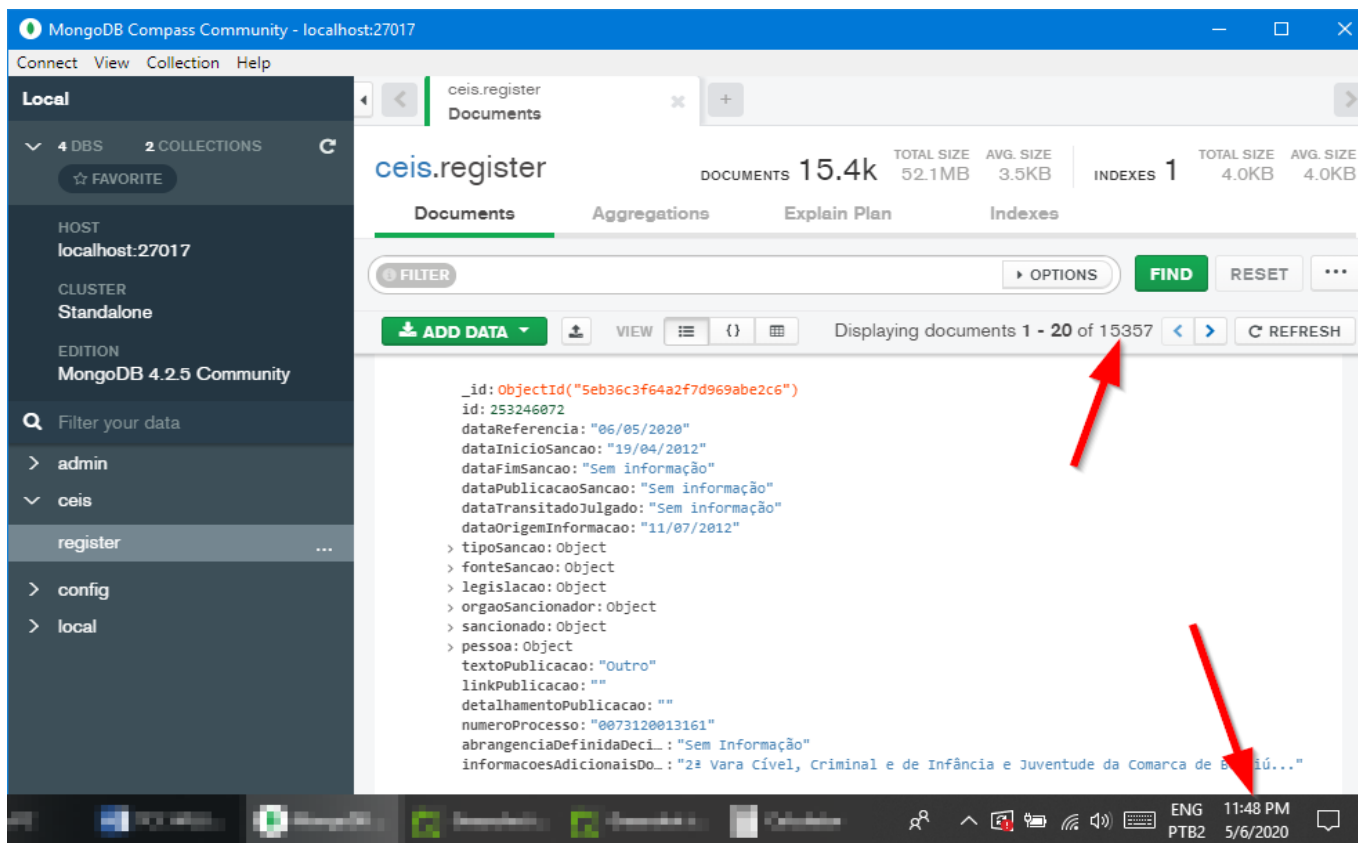
```
eda.py > ...  
1  import pandas as pd  
2  import utils as utl  
3  
4  data = utl.getListFromMongoCol('ceis', 'register')  
5  df = utl.flattenListAsDF(data)  
6
```

```
21  
22  def flattenListAsDF(data):  
23      data_row_flattened = []  
24      for data_row in data:  
25          data_row_flattened.append(flatten(data_row))  
26      return pd.DataFrame(data_row_flattened)  
27
```



Processamento e tratamento dos dados

- df.shape (15357, 61)



Processamento e tratamento dos dados

- `df.shape => (15357, 61)`

← → ↻ ↗ ⓘ Not secure | portaltransparencia.gov.br/sancoes

Ir para o conteúdo 1 Ir para o menu 2 Ir para a busca 3 Ir para o rodapé 4

A+ A- ACESSIBILIDADE ALTO CONTRASTE MAPA DO SITE

Portal da Transparência

CONTROLADORIA-GERAL DA UNIÃO

Busque por órgão, cidade, CNPJ, servidor...

Sobre o Portal | Painéis | Consultas Detalhadas | Controle social | Rede de Transparência | Receba Notificações | Aprenda mais

VOCÊ ESTÁ AQUI: INÍCIO > PAINEL DE SANÇÕES

Sanções

TOTAL DE SANÇÕES VIGENTES	QUANTIDADE DE SANCIONADOS (PESSOAS FÍSICAS OU JURÍDICAS)
38.414	31.246

Dados referentes aos cadastros de sanções aplicadas a pessoas físicas, jurídicas e servidores públicos federais

Consulte a origem dos dados para saber a última atualização das informações específicas.

Visão geral das sanções vigentes

CADASTRO DE SANÇÕES	QUANTIDADE DE SANÇÕES VIGENTES	QUANTIDADE DE SANCIONADOS (PESSOAS FÍSICAS OU JURÍDICAS)
CEIS - Cadastro de Empresas Inidôneas e Suspensas	15.357	12.832
CNEP - Cadastro Nacional de Empresas Punidas	206	146
CEPIM - Cadastro de Entidades Privadas sem Fins Lucrativos Impedidas	4.750	2.558
CEAF - Cadastro de Expulsões da Administração Federal	18.090	15.699
Acordos de Leniência	11	11
TOTAL	38.414	31.246

ENG 11:50 PM
PTB2 5/6/2020

Processamento e tratamento dos dados

- `df.pessoa_tipoCodigo.value_counts()`

```
CPF      8473  
CNPJ     6874  
      10  
Name: pessoa_tipoCodigo, dtype: int64
```

- Identificados 10 nulos

```
# identificados valores vazios => substituindo por valores nao numericos  
df['pessoa_tipoCodigo'].replace([''], np.nan, inplace=True)  
print(df.pessoa_tipoCodigo.value_counts())
```


```
CPF      8473  
CNPJ     6874  
Name: pessoa_tipoCodigo, dtype: int64
```



Processamento e tratamento dos dados

- df.pessoa_municipio_uf_sigla.value_counts()

```
SP    4513
PR    1159
RS    1114
BA    1097
MG     927
RJ     844
SC     595
GO     518
MA     489
DF     422
RN     367
PB     347
ES     337
PE     337
MT     334
CE     324
RO     284
PA     232
AM     207
MS     185
TO     132
PI     123
SE     117
AL      86
AP      81
AC      72
-1      67
RR      47
Name: pessoa_municipio_uf_sigla, dtype: int64
```



```
# identificado valor invalido => substituindo por valor nao numerico
df['pessoa_municipio_uf_sigla'].replace(['-1'], np.nan, inplace=True)
print(df.pessoa_municipio_uf_sigla.value_counts())
```


```
SP    4513
PR    1159
RS    1114
BA    1097
MG     927
RJ     844
SC     595
GO     518
MA     489
DF     422
RN     367
PB     347
PE     337
ES     337
MT     334
CE     324
RO     284
PA     232
AM     207
MS     185
TO     132
PI     123
SE     117
AL      86
AP      81
AC      72
RR      47
Name: pessoa_municipio_uf_sigla, dtype: int64
```



Processamento e tratamento dos dados

- df.orgaoSancionador_siglaUf.value_counts()

```
SP    4001
DF    1241
RS    1088
BA    1059
MG     959
PR     930
RJ     818
       796
SC     563
MA     458
RN     343
PB     333
GO     313
RO     309
PE     287
ES     278
CE     256
PA     208
AM     173
MT     161
MS     159
TO     132
PI     121
SE     117
AL      88
AC      69
AP      47
RR      46
ba       2
rj       2
Name: orgaoSancionador_siglaUf, dtype: int64
```



```
# identificados estados com siglas minusculas => padronizando todas para maiusculas
df['orgaoSancionador_siglaUf'] = df['orgaoSancionador_siglaUf'].str.upper()
# identificados estados vazios => substituindo para valores nao numericos
df['orgaoSancionador_siglaUf'].replace([''], np.nan, inplace=True)
print(df.orgaoSancionador_siglaUf.value_counts())
```

```
SP    4001
DF    1241
RS    1088
BA    1061
MG     959
PR     930
RJ     820
SC     563
MA     458
RN     343
PB     333
GO     313
RO     309
PE     287
ES     278
CE     256
PA     208
AM     173
MT     161
MS     159
TO     132
PI     121
SE     117
AL      88
AC      69
AP      47
RR      46
Name: orgaoSancionador_siglaUf, dtype: int64
```

Processamento e tratamento dos dados

- Tranformação das séries dataInicioSancao e dataFimSancao (string => dateTime)

```
# convertendo series dataInicioSancao de string para dateTime
dataInicioSancao = pd.to_datetime(df['dataInicioSancao'])
df['dataInicioSancao'] = dataInicioSancao

# convertendo serie dataFimSancao de string para dateTime utilizando "coerce"
dataFimSancao = pd.to_datetime(df['dataFimSancao'], errors='coerce')
df['dataFimSancao'] = dataFimSancao
```



Análise e exploração dos dados



Análise e exploração dos dados

- Série “pessoa_tipoCodigo”

```
#inicio do eda  
  
# distribuicao PF/PJ  
print(df.pessoa_tipoCodigo.value_counts(normalize=True)*100)
```

```
CPF      55.209487  
CNPJ     44.790513  
Name: pessoa_tipoCodigo, dtype: float64
```



Análise e exploração dos dados

- Série “pessoa_tipoPessoa”

```
# distribuicao categoria pessoa
print(df.pessoa_tipoPessoa.value_counts(normalize=True)*100)
```

Pessoa Física	55.193072
Entidades Empresariais Privadas	43.439474
Entidades Sem Fins Lucrativos	1.243732
Sem Informação	0.071629
Administração Pública Municipal	0.032558
Administração Pública Estadual ou do Distrito Federal	0.013023
Administração Pública Federal	0.006512

Name: pessoa_tipoPessoa, dtype: float64

```
# pivot para verificar a relacao entre pessoa_tipoCodigo e pessoa_tipoPessoa
pivot = pd.pivot_table(df,index=["pessoa_tipoCodigo","pessoa_tipoPessoa"], values=["id"],aggfunc={"id":len})
result = pivot.sort_values('id', ascending=False)
print(result)
```

pessoa_tipoCodigo	pessoa_tipoPessoa	id
CPF	Pessoa Física	8473
CNPJ	Entidades Empresariais Privadas	6671
	Entidades Sem Fins Lucrativos	191
	Administração Pública Municipal	5
	Pessoa Física	3
	Administração Pública Estadual ou do Distrito F...	2
	Administração Pública Federal	1
	Sem Informação	1

Análise e exploração dos dados

- Série “pessoa_municipio_uf_sigla”

```
# distribuicao sancoes por estado  
print(df.pessoa_municipio_uf_sigla.value_counts(normalize=True)*100)
```

```
SP    29.516024  
PR     7.580118  
RS     7.285808  
BA     7.174624  
MG     6.062786  
RJ     5.519948  
SC     3.891432  
GO     3.387835  
MA     3.198169  
DF     2.759974  
RN     2.400262  
PB     2.269457  
PE     2.204055  
ES     2.204055  
MT     2.184434  
CE     2.119032  
RO     1.857423  
PA     1.517332  
AM     1.353826  
MS     1.209941  
TO     0.863309  
PI     0.804447  
SE     0.765206  
AL     0.562459  
AP     0.529758  
AC     0.470896  
RR     0.307390  
Name: pessoa_municipio_uf_sigla, dtype: float64
```



Análise e exploração dos dados

- Série “pessoa_cnae_secao”

```
# distribuicao sancoes por cnae
print(df.pessoa_cnae_secao.value_counts(normalize=True)*100)
```

Sem informação	65.956893
COMÉRCIO; REPARAÇÃO DE VEÍCULOS AUTOMOTORES E MOTOCICLETAS	14.651299
ATIVIDADES ADMINISTRATIVAS E SERVIÇOS COMPLEMENTARES	6.557270
ATIVIDADES PROFISSIONAIS, CIENTÍFICAS E TÉCNICAS	3.451195
TRANSPORTE, ARMAZENAGEM E CORREIO	3.047470
INDÚSTRIAS DE TRANSFORMAÇÃO	2.617699
INFORMAÇÃO E COMUNICAÇÃO	1.002800
ALOJAMENTO E ALIMENTAÇÃO	0.800938
SAÚDE HUMANA E SERVIÇOS SOCIAIS	0.560005
EDUCAÇÃO	0.280003
ARTES, CULTURA, ESPORTE E RECREAÇÃO	0.260468
ATIVIDADES IMOBILIÁRIAS	0.214886
ATIVIDADES FINANCEIRAS, DE SEGUROS E SERVIÇOS RELACIONADOS	0.182327
INDÚSTRIAS EXTRATIVAS	0.149769
OUTRAS ATIVIDADES DE SERVIÇOS	0.104187
ADMINISTRAÇÃO PÚBLICA, DEFESA E SEGURIDADE SOCIAL	0.078140
AGRICULTURA, PECUÁRIA, PRODUÇÃO FLORESTAL, PESCA E AQUICULTURA	0.065117
SERVIÇOS DOMÉSTICOS	0.019535

Name: pessoa_cnae_secao, dtype: float64

	pessoa_cnae_secao	id
CPF	Sem informação	8473
CNPJ	COMÉRCIO; REPARAÇÃO DE VEÍCULOS AUTOMOTORES E M...	2250
	Sem informação	1646
	ATIVIDADES ADMINISTRATIVAS E SERVIÇOS COMPLEMEN...	1007
	ATIVIDADES PROFISSIONAIS, CIENTÍFICAS E TÉCNICAS	530
	TRANSPORTE, ARMAZENAGEM E CORREIO	468
	INDÚSTRIAS DE TRANSFORMAÇÃO	402
	INFORMAÇÃO E COMUNICAÇÃO	154
	ALOJAMENTO E ALIMENTAÇÃO	123
	SAÚDE HUMANA E SERVIÇOS SOCIAIS	86
	EDUCAÇÃO	43
	ARTES, CULTURA, ESPORTE E RECREAÇÃO	40
	ATIVIDADES IMOBILIÁRIAS	33
	ATIVIDADES FINANCEIRAS, DE SEGUROS E SERVIÇOS R...	28
	INDÚSTRIAS EXTRATIVAS	23
	OUTRAS ATIVIDADES DE SERVIÇOS	16
	ADMINISTRAÇÃO PÚBLICA, DEFESA E SEGURIDADE SOCIAL	12
	AGRICULTURA, PECUÁRIA, PRODUÇÃO FLORESTAL, PESCA...	10
	SERVIÇOS DOMÉSTICOS	3

Análise e exploração dos dados

- Série “orgaoSancionador_siglaUf”

```
# distribuicao sancoes por estado do orgao sancionador  
print(df.orgaoSancionador_siglaUf.value_counts(normalize=True)*100)
```

```
SP    27.477508  
DF     8.522766  
RS     7.472014  
BA     7.286587  
MG     6.586086  
PR     6.386924  
RJ     5.631481  
SC     3.866493  
MA     3.145388  
RN     2.355607  
PB     2.286931  
GO     2.149578  
RO     2.122107  
PE     1.971018  
ES     1.909210  
CE     1.758121  
PA     1.428473  
AM     1.188105  
MT     1.105693  
MS     1.091958  
TO     0.906531  
PI     0.830987  
SE     0.803516  
AL     0.604354  
AC     0.473869  
AP     0.322780  
RR     0.315912  
Name: orgaoSancionador_siglaUf, dtype: float64
```



Análise e exploração dos dados

- Série “orgaoSancionador_poder”

```
# distribuicao sancoes pela esfera de poder do orgao sancionador  
print(df.orgaoSancionador_poder.value_counts(normalize=True)*100)
```

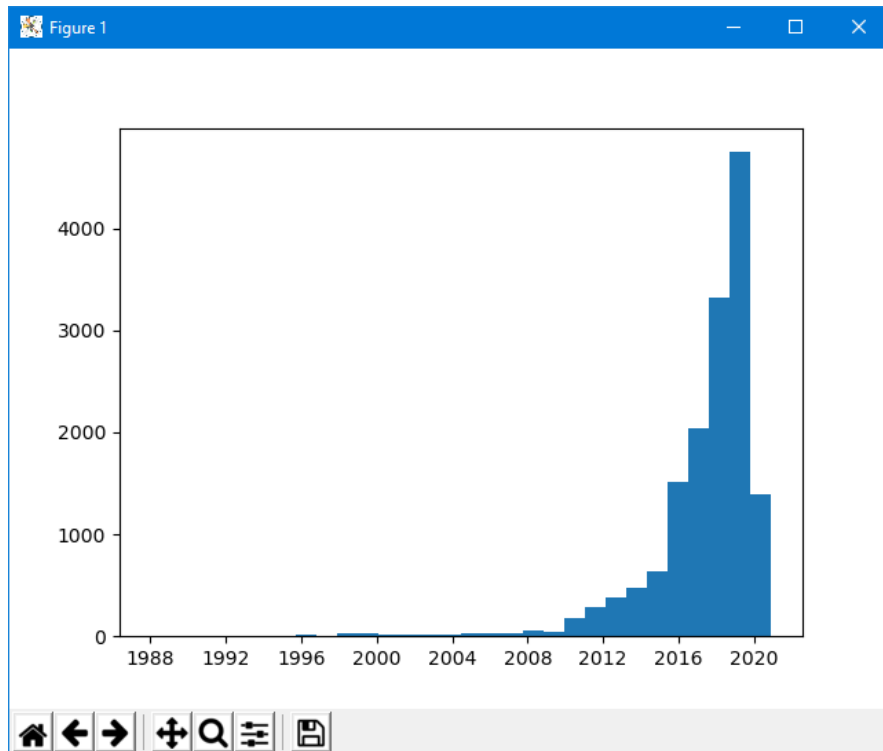
```
Judiciário          60.793124  
Executivo           35.951032  
Legislativo         2.513512  
Tribunal de Contas  0.462330  
Ministério Público  0.182327  
Entidade Paraestatal 0.097675  
Name: orgaoSancionador_poder, dtype: float64
```



Análise e exploração dos dados

- Série “dataInicioSancao”

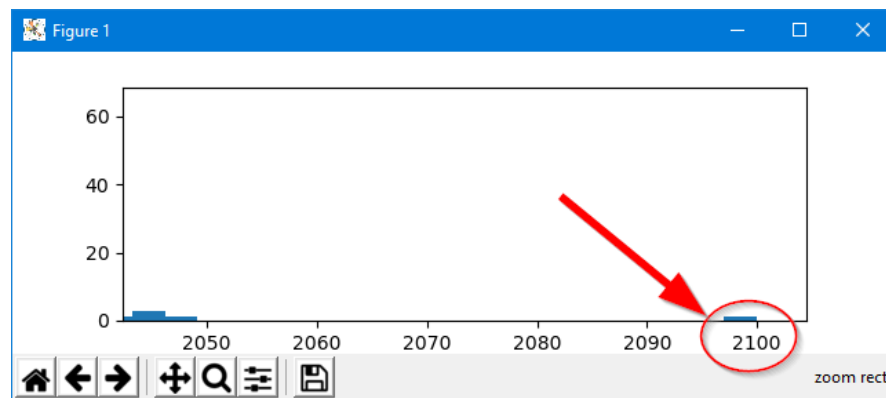
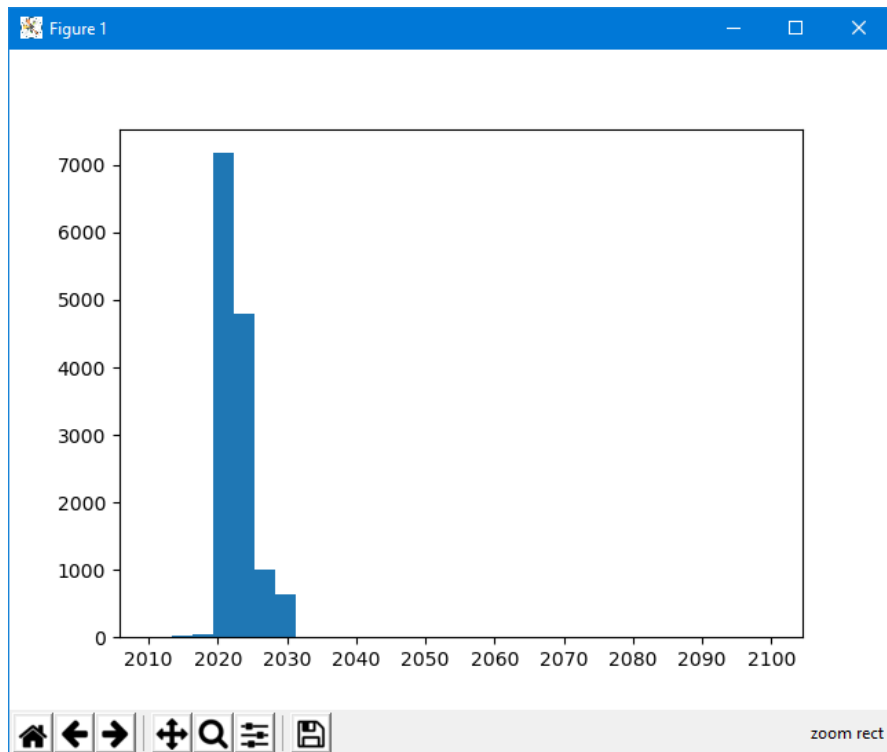
```
# verificando a distribuicao de sancoes por "dataInicioSancao"  
plt.hist(df['dataInicioSancao'].dropna(), bins=30)  
plt.show()
```



Análise e exploração dos dados

- Série “dataFimSancao”

```
# verificando a distribuicao de sancoes por "dataFimSancao"  
plt.hist(df['dataFimSancao'].dropna(), bins=30)  
plt.show()
```



Análise e exploração dos dados

- Série “dataFimSancao”

```
# no grafico anterior foram identificados valores extremos na serie dataFimSancao  
# filtrando apenas datas posteriores a 2030 na serie dataFimSancao para analise  
print(dataFimSancao[df['dataFimSancao'] > '31/12/2030'].value_counts().sort_index())
```

```
2032-06-09    1  
2032-07-02    2  
2032-09-18    1  
2032-09-23    1  
2039-10-07   14  
2040-09-24    1  
2044-01-30    1  
2044-09-24    2  
2048-12-05    1  
2099-12-31    1  
Name: dataFimSancao, dtype: int64
```

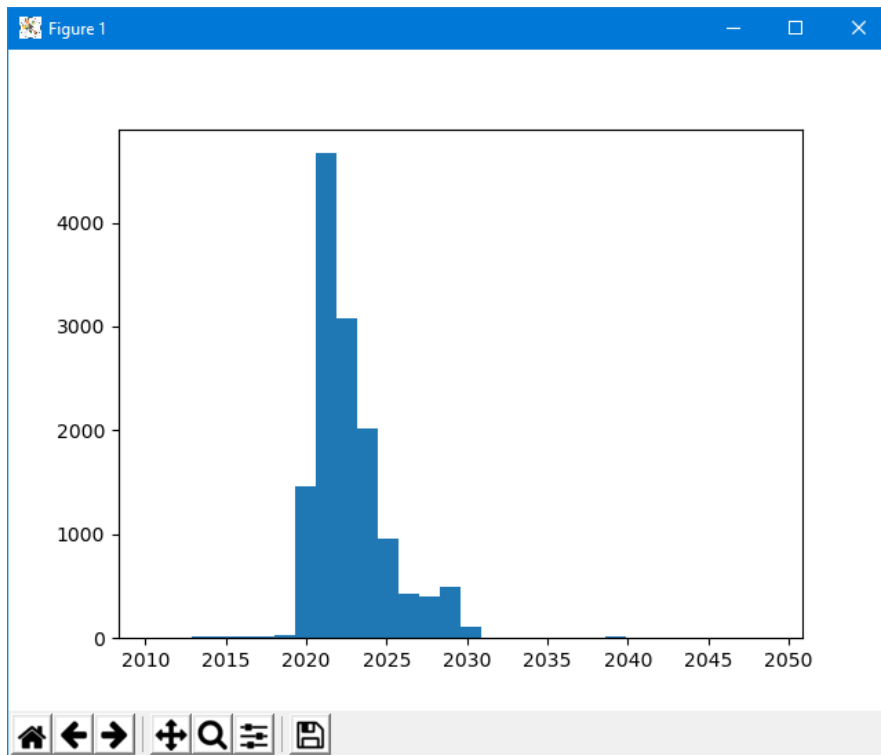
```
# atribuindo valor NaT para data outlier  
df.loc[df['dataFimSancao'] == '31/12/2099', 'dataFimSancao'] = pd.Timedelta('nat')  
# atribuindo a correcao a variavel dataFimSancao  
dataFimSancao = df['dataFimSancao']  
# vizualizando datas acima de 2030 para verificar a exclusao da data outlier  
print(dataFimSancao[df['dataFimSancao'] > '31/12/2030'].value_counts().sort_index())
```

```
2032-06-09    1  
2032-07-02    2  
2032-09-18    1  
2032-09-23    1  
2039-10-07   14  
2040-09-24    1  
2044-01-30    1  
2044-09-24    2  
2048-12-05    1  
Name: dataFimSancao, dtype: int64
```



Análise e exploração dos dados

- Série “dataFimSancao”



Análise e exploração dos dados

- Prazos das sanções

```
# obtendo os prazos das sancões para analises
prazoSancao = dataFimSancao - dataInicioSancao
# describe para analisar os dados ref prazo das sancões
print(prazoSancao.describe(include='all'))
```

```
count          13724
mean    1786 days 06:05:14.777032
std      1076 days 02:47:41.264456
min      -156 days +00:00:00
25%       1096 days 00:00:00
50%       1826 days 00:00:00
75%       1827 days 00:00:00
max       11894 days 00:00:00
dtype: object
```



Análise e exploração dos dados

■ Prazos das sanções

```
# no describe acima foi identificado que o min e negativo, o que nao faz sentido
# essas datas serao removidas do dataframe
# filtrando os prazos negativos (onde dataFimSancao < dataInicioSancao)
dataFimSancao_negative_boolean = dataFimSancao < dataInicioSancao
# substituindo essas datas por NaT no dataframe
df.at[dataFimSancao_negative_boolean, 'dataFimSancao'] = pd.Timedelta('nat')
# atribuindo os valores atualizados a serie dataFimSancao
dataFimSancao = df['dataFimSancao']
# recalculando o prazo
prazoSancao = dataFimSancao - dataInicioSancao
# visualizando o describe para validar a correcao
print(prazoSancao.describe(include='all'))
```

```
count      13681
mean    1791 days 23:40:06.403040
std      1072 days 22:16:35.928344
min        0 days 00:00:00
25%       1096 days 00:00:00
50%       1826 days 00:00:00
75%       1827 days 00:00:00
max       11894 days 00:00:00
dtype: object
```

```
# obtendo o valor medio em meses
prazoSancao_mean = prazoSancao.describe().loc['mean']
prazoSancao_meses = prazoSancao_mean / np.timedelta64(1, 'M')
print(round(prazoSancao_meses))
```

59

Coeficiente de Variação
~60%



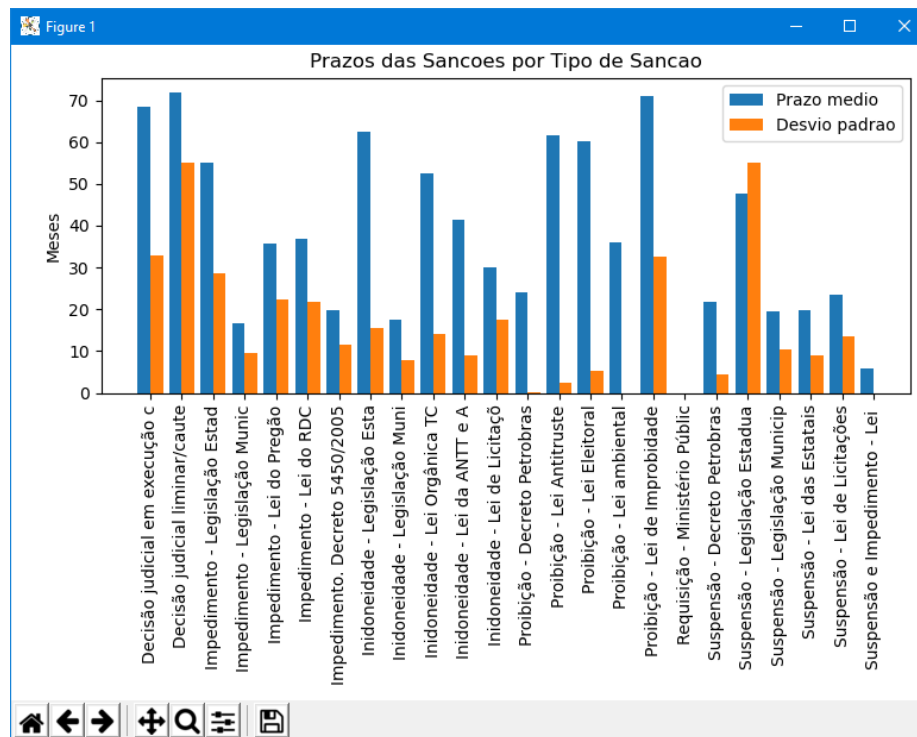
Análise e exploração dos dados

- Prazo das sanções por tipo de sanção

```
# analisando prazos das sancões por tipo de sanção
df_tipoSancao = df['tipoSancao_descricaoResumida'].unique()
df_tipoSancao = list(df_tipoSancao)
df_tipoSancao = np.sort(df_tipoSancao)

labels = []
prazo_mean = []
prazo_std = []
for tipoSancao in df_tipoSancao:
    df_tipoSancao_it = df[df['tipoSancao_descricaoResumida'] == tipoSancao]
    prazo_tipoSancao = (df_tipoSancao_it['dataFimSancao'] - df_tipoSancao_it['dataInicioSancao'])
    labels.append(tipoSancao[:30])
    prazo_mean.append(prazo_tipoSancao.describe().loc['mean'] / np.timedelta64(1, 'M'))
    prazo_std.append(prazo_tipoSancao.describe().loc['std'] / np.timedelta64(1, 'M'))

utl.groupedBarWithLabels(prazo_mean, prazo_std, labels,
    'Prazo medio', 'Desvio padrao', 'Meses', 'Prazos das Sancões por Tipo de Sanção')
```



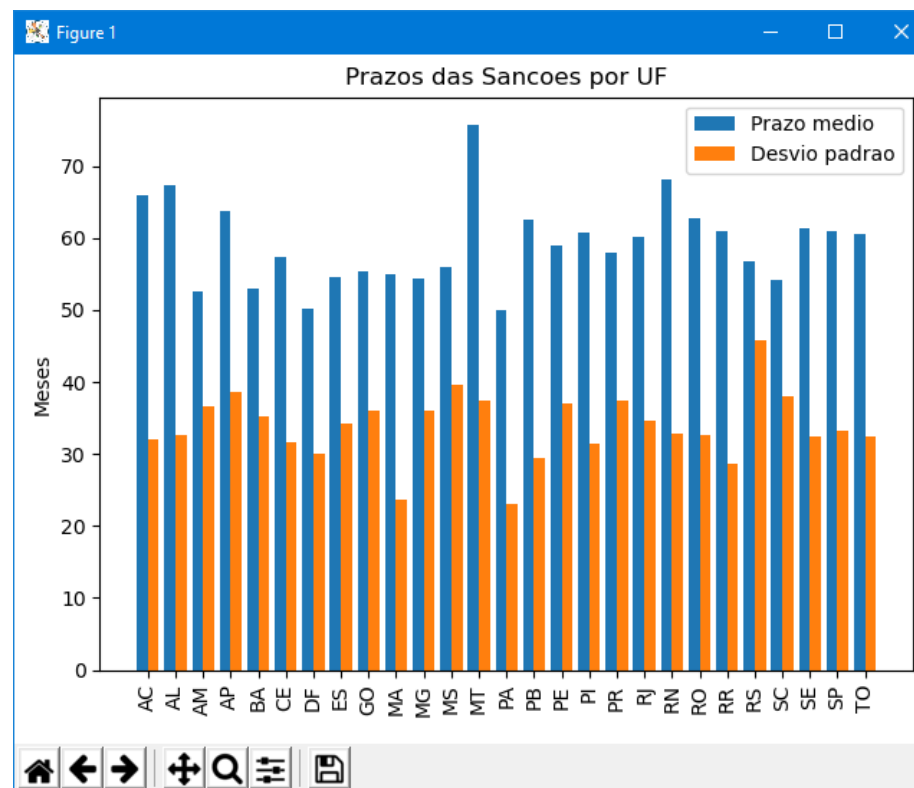
Análise e exploração dos dados

- Prazo das sanções por estado

```
# analisando prazos das sancões por estado
df_uf = df['pessoa_municipio_uf_sigla'].unique()
df_uf = list(df_uf)
df_uf.remove(np.nan)
df_uf = np.sort(df_uf)

labels = []
prazo_mean = []
prazo_std = []
prazoSancao_uf = []
for uf in df_uf:
    df_uf_it = df[df['pessoa_municipio_uf_sigla'] == uf]
    prazo_uf = (df_uf_it["dataFimSancao"] - df_uf_it["dataInicioSancao"])
    prazoSancao_uf.append(prazo_uf.describe())
    labels.append(uf)
    prazo_mean.append(prazo_uf.describe().loc['mean'] / np.timedelta64(1, 'M'))
    prazo_std.append(prazo_uf.describe().loc['std'] / np.timedelta64(1, 'M'))

utl.groupedBarWithLabels(prazo_mean, prazo_std, labels,
    ['Prazo medio', 'Desvio padrao', 'Meses', 'Prazos das Sancões por UF'])
```



Análise e exploração dos dados

- Análise de reincidência

```
# efetuando análises nas entidades sancionadas
# verificando o índice de reincidência de sanções por entidades
pessoaCodigo = df['pessoa_codigoFormatado']
pessoaCodigo_value_counts = pessoaCodigo.value_counts()
print(pessoaCodigo_value_counts[pessoaCodigo_value_counts > 2].sum() / pessoaCodigo_value_counts.sum())
```

Percentual de reincidência

0.16357361463827572

~16%

```
# filtrando o dataframe para obter apenas pessoas com quantidade de sanções > 2
pessoaReincidente = pessoaCodigo_value_counts[pessoaCodigo_value_counts > 2]
df_pessoaReincidente = df[df.pessoa_codigoFormatado.isin(pessoaReincidente.index)]
print(df_pessoaReincidente.describe(include='all'))
```

```
# verificando os tipos de pessoa (PF ou PJ?)
print(df_pessoaReincidente.pessoa_tipoCodigo.value_counts(normalize=True)*100)
```

```
CPF      54.956035
CNPJ     45.043965
Name: pessoa_tipoCodigo, dtype: float64
```



Apresentação dos resultados



Apresentação dos resultados

CEIS - Cadastro de Empresas Inidôneas e Suspensas

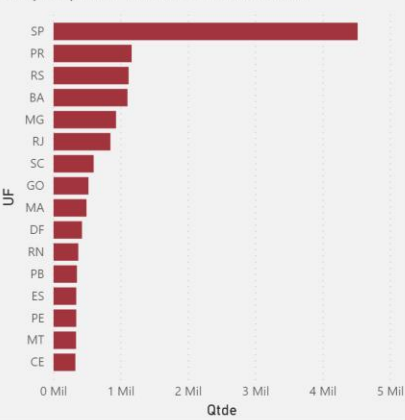
15.357

SANÇÕES VIGENTES EM MAIO/2020

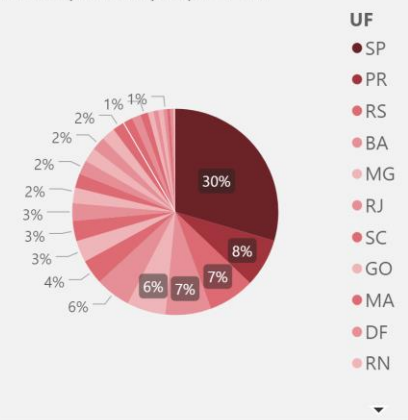
Mapa das sanções no Brasil



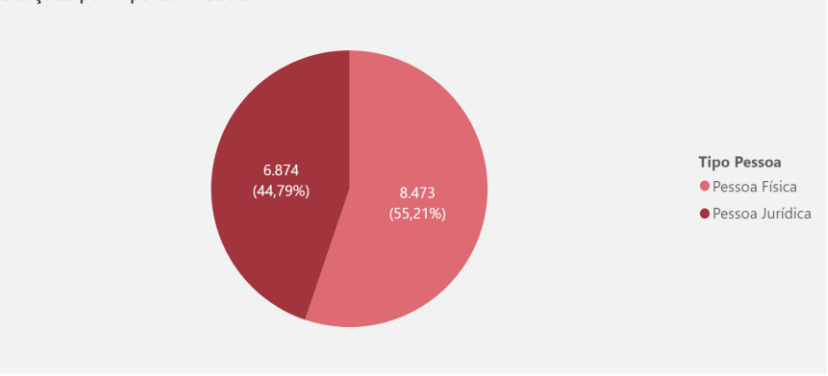
Sanções por UF (somente acima de 300)



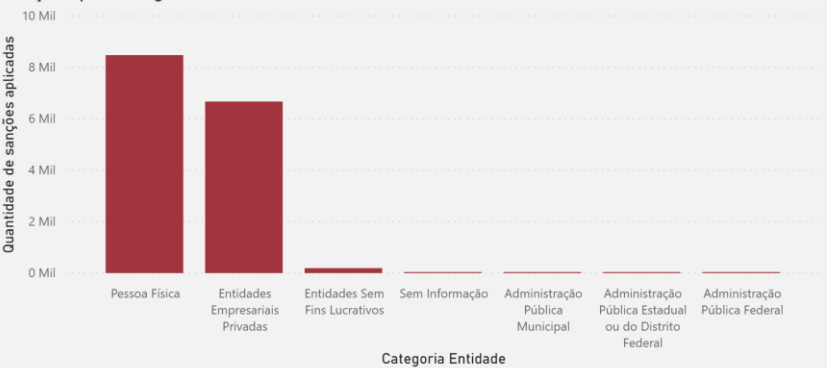
Distribuição de sanções por UF (%)



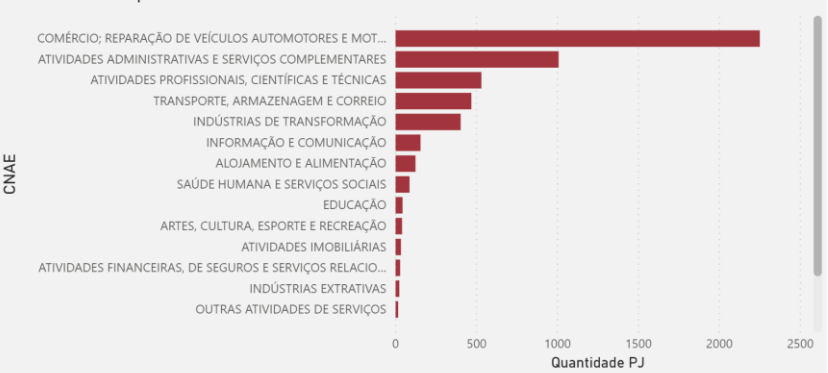
Sanções por Tipo de Pessoa



Sanções por Categoria Entidade

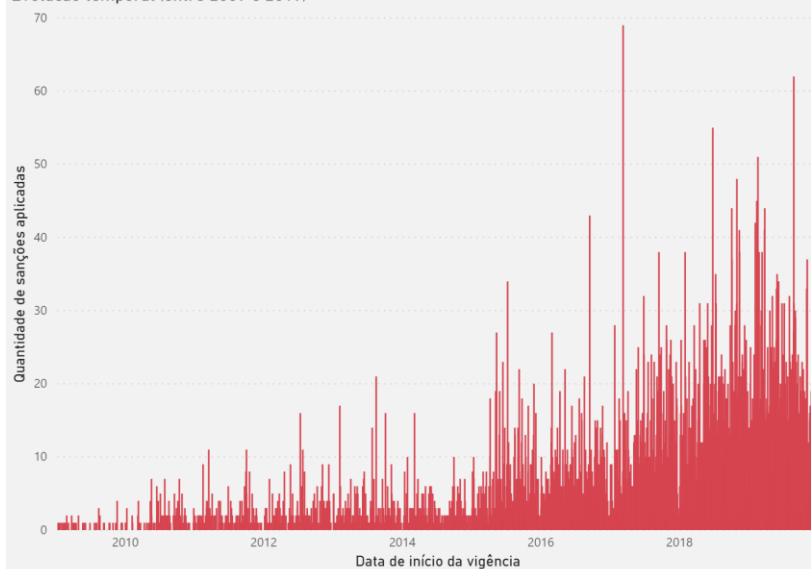


Quantidade de PJ por CNAE

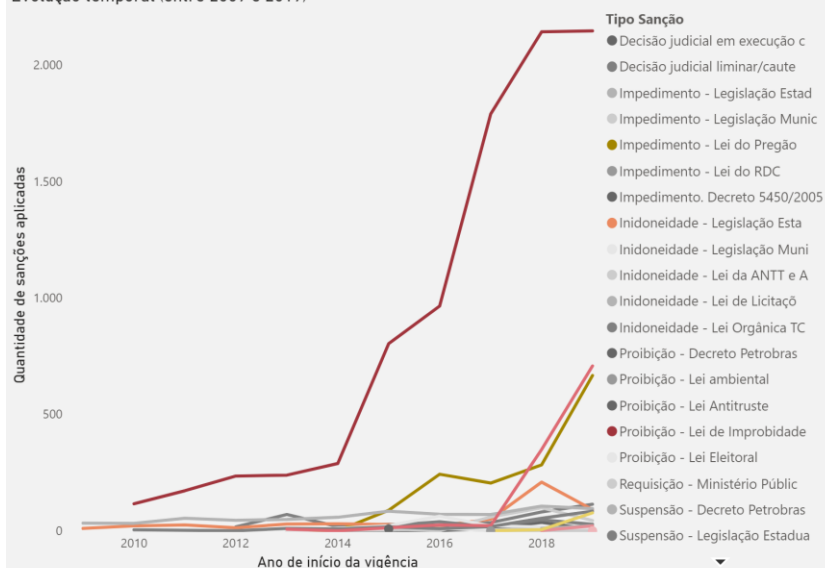


Apresentação dos resultados

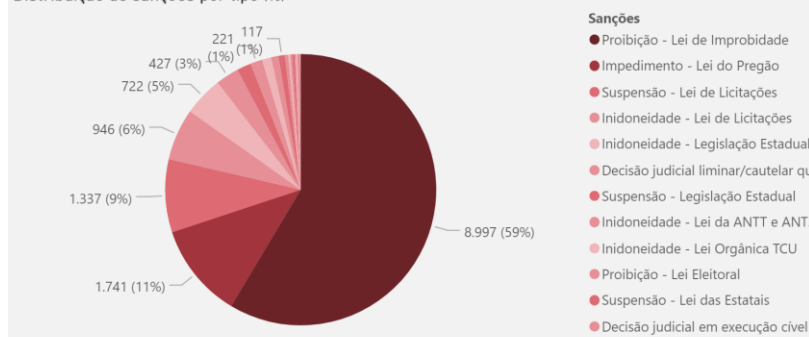
Evolução temporal (entre 2009 e 2019)



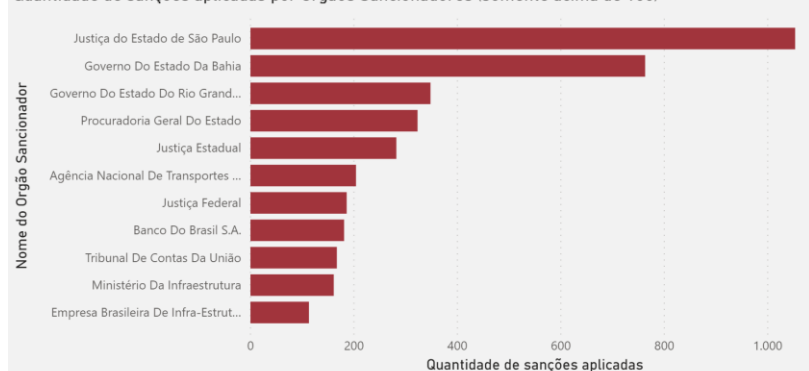
Evolução temporal (entre 2009 e 2019)



Distribuição de sanções por tipo (%)



Quantidade de sanções aplicadas por Órgãos Sancionadores (somente acima de 100)



Quantidade de sanções aplicadas por tipo de poder

