**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA**

**Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data**

**William Sanches Lima**

**obtenção de insights dos dados do portal da transparência através de Análise exploratória e representação visual dos dados**

Belo Horizonte

2020

**William Sanches Lima**

**OBTENÇÃO DE INSIGHTS DOS DADOS DO PORTAL DA TRANSPARÊNCIA ATRAVÉS DE ANÁLISE EXPLORATÓRIA E REPRESENTAÇÃO VISUAL DOS DADOS**

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Ciência de Dados e Big Data como requisito parcial à obtenção do título de especialista.

Belo Horizonte

2020 **SUMÁRIO**

[1. Introdução 4](#_Toc445198572)

[1.1. Contextualização 4](#_Toc445198573)

[1.1. O problema proposto 4](#_Toc445198574)

[2. Coleta de Dados 4](#_Toc445198576)

[3. Processamento/Tratamento de Dados 5](#_Toc445198582)

[4. Análise e Exploração dos Dados 5](#_Toc445198585)

[5. Criação de Modelos de Machine Learning 5](#_Toc445198586)

[6. Apresentação dos Resultados 5](#_Toc445198587)

[7. Links 6](#_Toc445198588)

[REFERÊNCIAS 7](#_Toc445198589)

# 1. Introdução

## 1.1. Contextualização

A Controladoria-Geral da União (CGU) acredita que a transparência é o melhor antídoto contra a corrupção, uma vez que ela é um importante mecanismo indutor para que os gestores públicos ajam com responsabilidade. Uma gestão pública transparente permite à sociedade, com informações, colaborar no controle das ações de seus governantes, com intuito de checar se os recursos públicos estão sendo usados como deveriam. Nesse contexto a CGU criou o programa Brasil Transparente para auxiliar Estados e Municípios na implementação das medidas de governo transparente previstas na Lei de Acesso à Informação (LAI). O objetivo é juntar esforços no incremento da transparência pública e na adoção de medidas de governo aberto. A cobrança por mais transparência tem crescido no Brasil em ritmo acelerado nos últimos anos. A promoção da transparência e do acesso à informação é considerada medida indispensável para o fortalecimento da democracia e para a melhoria da gestão pública. Nesse sentido foi lançado pelo Ministério da Transparência e Controladoria-Geral da União em 2004, o Portal da Transparência do Governo Federal é um site de acesso livre, no qual o cidadão pode encontrar informações sobre como o dinheiro público é utilizado, além de se informar sobre assuntos relacionados à gestão pública do Brasil. Com base nesses dados foi possível utilizar técnicas de Análise Exploratória de Dados (Exploratory Data Analysis – EDA) e Visualização de Dados para obtenção de “insights” (também conhecido como “data insight”).

## 1.2. O problema proposto

O problema proposto foi analisar os dados disponíveis no Portal da Transparência da Controladoria Geral da União para obtenção de “insights”, aplicando técnicas de Análise Exploratória de Dados e Visualização de Dados, com o objetivo de comprovar a eficácia dessas técnicas no apoio ao processo de seleção e contratação de fornecedores no âmbito da administração pública.

(Why?) Por que esse problema é importante? Desde 2014 o Brasil vem caindo no ranking do Índice de Percepção da Corrupção (IPC), elaborado pela organização Transparência Internacional que avalia a percepção da corrupção no setor público em 180 países. Em 2016, o Brasil ficou em 79º; em 2017 o país estava na 96ª colocação; em 2019 ocupou o 105º e agora em 2020 está em 106º lugar. Dessa forma entende-se que o problema em questão é de extrema relevância pois o “custo” da corrupção tem impacto direto na diminuição dos investimentos na saúde, na educação, em infraestrutura, segurança, habitação, entre outros direitos essenciais à vida.

(Who?) De quem são os dados analisados? De um governo? Um ministério ou secretaria? Dados de clientes? Os dados são do governo, do portal da transparência, do Ministério da Transparência e Controladoria-Geral da União, fruto do programa Brasil Transparente para auxiliar Estados e Municípios na implementação das medidas de governo transparente previstas na Lei de Acesso à Informação (LAI).

(What?): Quais os objetivos com essa análise? O que iremos analisar? O objetivo é recuperar os dados do Portal da Transparência para realização de Análise Exploratória dos Dados e Visualização dos Dados afim de obter “insights”, para comprovar a eficácia da aplicação dessas técnicas no apoio ao processo de seleção e contratação de fornecedores no âmbito da administração pública e também para direcionar os órgãos sancionadores e legisladores no entendimento do cenário afim de possibilitar melhorias nas normas e legislações que tratam desse assunto. Os dados analisados serão do Cadastro de Empresas Inidôneas e Suspensas (CEIS), do Portal da Transparência da Controladoria Geral da União.

(Where?): Trata dos aspectos geográficos e logísticos de sua análise. Os dados tem abrangência nacional, mas possuem detalhes relacionados aos estados e munícipios, possibilitando a qualificação dos dados com essas informações.

# 2. Coleta de Dados

Os dados foram obtidos do portal da transparência, do Ministério da Transparência e Controladoria-Geral da União. Os dados divulgados no Portal são provenientes de diversas fontes de informação, entre as quais estão os grandes sistemas estruturadores do Governo Federal – como o Sistema Integrado de Administração Financeira do Governo Federal (Siafi) e o Sistema Integrado de Administração de Recursos Humanos (Siape) –, as bases de benefícios sociais, as faturas de Cartão de Pagamentos do Governo Federal, as bases de imóveis funcionais, entre diversas outras.

Os órgãos responsáveis por cada fonte de informação encaminham seus dados para a CGU, que recebe, reúne e disponibiliza as informações na ferramenta. A periodicidade de envio dos dados depende do assunto tratado, assim como a periodicidade de atualização das informações no Portal.

Uma vez carregadas no Portal, as informações são disponibilizadas para conhecimento do cidadão de diversas formas, como: painéis, consultas detalhadas, gráficos, dados abertos.

O acesso ao Portal não requer usuário nem senhas, sendo permitido a qualquer cidadão navegar pelas páginas de forma livre, bem como visualizar e utilizar os dados disponíveis da forma que melhor lhe convier.

O Portal da Transparência está atento aos princípios de Governo Eletrônico e sabe que os dados devem ser disponibilizados de formas diferentes a fim de atender aos diversos perfis de usuários. Para isso, além de consultas online e com visualizações que buscam transmitir, de forma simples, como o governo usa os recursos públicos, formas de acesso aos dados para desenvolvedores e engajados com a tecnologia da informação também estão disponíveis.

O acesso para desenvolvedores e engajados ocorre através de uma Interface de Programa de Aplicativos (do inglês, “Application Programming Interface”), ou simplesmente “API”. Com ela, é possível ter um serviço de consulta direta aos dados do Portal da Transparência sem precisar navegar pelo site ou utilizar robôs para a obtenção das informações de forma automática. Os dados disponíveis são os mesmos apresentados em tela, com a flexibilidade característica das APIs.

No Portal da Transparência, o serviço de consulta via API foi implementado em REST (Representational State Transfer). Entendemos que o REST é mais simples de ser utilizado, além de poder ser testado aqui mesmo no Portal. Além do mais, o REST costuma ser a escolha preferida dos desenvolvedores.

As consultas via API têm restrição quanto ao número de requisições por minuto, de modo a não sobrecarregar o Portal da Transparência e impactar no tempo de resposta das consultas realizadas por meio dos navegadores. No período de 6:00 às 23:59, o Portal aceita 90 requisições por minuto. Já no período das 00:00 às 5:59, são aceitas 300 requisições por minuto.

O acesso via API fornece toda uma flexibilidade para consultas pontuais. Para acesso ao conjunto completo de dados, no entanto, sugerimos utilizar as planilhas de dados abertos. O Portal da Transparência agrega uma enorme quantidade de dados e baixar as planilhas para uso local certamente fornecerá melhores resultados para grandes volumes de dados. O download de planilhas pode ser feito em seção específica.

A documentação do uso das APIs pode ser acessada em <http://www.transparencia.gov.br/swagger-ui.html>. Estão disponíveis as seguintes consultas: Bolsa Família, Programa de Erradicação do Trabalho Infantil (Peti), Garantia-Safra, Seguro Defeso, Cadastro de Expulsões da Administração Federal (CEAF), Cadastro Nacional de Empresas Inidôneas e Suspensas (CEIS), Cadastro Nacional de Empresas Punidas (CNEP), Contratos do Poder Executivo Federal, Convênios do Poder Executivo Federal, Despesas Públicas, Entidades Privadas sem Fins Lucrativos Impedidas (CEPIM), Licitações do Poder Executivo Federal, Servidores do Poder Executivo Federal e Viagens a Serviço.

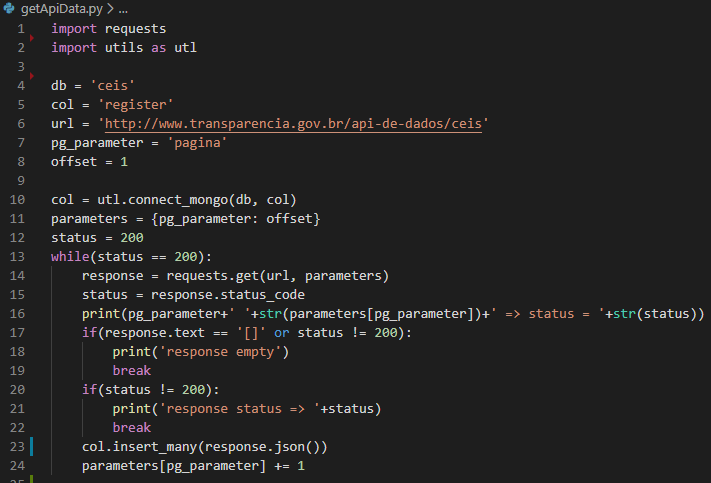
Para o desenvolvimento desse trabalho foram utilizadas as API’s do Cadastro Nacional de Empresas Inidôneas e Suspensa (CEIS), que apresenta um dataset maior e com mais atributos para as análises.

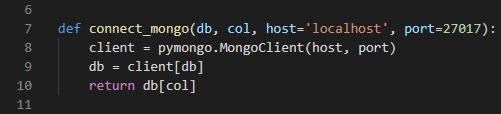
Os dados foram obtidos através da API de Consulta dos registros do CNEP por CNPJ ou CPF Sancionado/Órgão Sancionador/Período. Abaixo a documentação da API:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GET | /api-de-dados/ceis | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| Notas de Implementação | | |  |  |  |  |  |  |
| Filtros mínimos: Página (padrão = 1); | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| Classe de resposta (Status 200) | | | |  |  |  |  |  |
| OK |  |  |  |  |  |  |  |  |
| Example Value: | |  |  |  |  |  |  |  |
| [  {  "abrangenciaDefinidaDecisaoJudicial": "string",  "dataFimSancao": "string",  "dataInicioSancao": "string",  "dataOrigemInformacao": "string",  "dataPublicacaoSancao": "string",  "dataReferencia": "string",  "dataTransitadoJulgado": "string",  "detalhamentoPublicacao": "string",  "fonteSancao": {  "enderecoContato": "string",  "nomeExibicao": "string",  "telefoneContato": "string"  },  "id": 0,  "informacoesAdicionaisDoOrgaoSancionador": "string",  "legislacao": {  "descricaoFundamentacaoLegal": "string",  "fundamentacaoLegal": "string"  },  "linkPublicacao": "string",  "numeroProcesso": "string",  "orgaoSancionador": {  "nome": "string",  "poder": "string",  "siglaUf": "string"  },  "pessoa": {  "cnae": {  "classe": "string",  "codigoClasse": "string",  "codigoDivisao": "string",  "codigoGrupo": "string",  "codigoSecao": "string",  "codigoSubclasse": "string",  "divisao": "string",  "grupo": "string",  "secao": "string",  "subclasse": "string"  },  "codigoFormatado": "string",  "complementoEndereco": "string",  "dataAbertura": "string",  "descricaoLogradouro": "string",  "enderecoEletronico": "string",  "localidadePessoa": "string",  "municipio": {  "codigoIBGE": "string",  "nomeIBGE": "string",  "pais": "string",  "uf": {  "nome": "string",  "sigla": "string"  }  },  "naturezaJuridica": {  "codigo": "string",  "codigoTipo": "string",  "descricao": "string",  "descricaoTipo": "string"  },  "nome": "string",  "nomeBairro": "string",  "nomeFantasiaReceita": "string",  "numeroCEP": "string",  "numeroEndereco": "string",  "numeroInscricaoSocial": "string",  "numeroTelefone": "string",  "razaoSocialReceita": "string",  "tipoCodigo": "string",  "tipoPessoa": "string"  },  "sancionado": {  "codigoFormatado": "string",  "nome": "string"  },  "textoPublicacao": "string",  "tipoSancao": {  "descricaoPortal": "string",  "descricaoResumida": "string"  }  } ] | | | | | | | | |
|
|
|  |  |  |  |  |  |  |  |  |
| Parâmetros | |  |  |  |  |  |  |  |
| Parâmetro | | Descrição | | | | Tipo de parâmetro | | Tipo de dados |
|  | |  | | | |  | |  |
| cnpjSancionado | | CNPJ ou CPF do Sancionado | | | | query | | string |
| nomeSancionado | | Nome, nome fantasia ou razão social do Sancionado | | | | query | | string |
| orgaoSancionador | | Órgão Sancionador | | | | query | | string |
| dataInicialSancao | | Data Inicial da Sanção (DD/MM/AAAA) | | | | query | | string |
| dataFinalSancao | | Data Final da Sanção (DD/MM/AAAA) | | | | query | | string |
| **pagina (required)** | | **Página consultada** | | | | query | | integer |
| Mensagens de resposta | | |  |  |  |  |  |  |
| Código de status HTTP | | Razão | |  |  |  |  |  |
|  | |  | |  |  |  |  |  |
| 401 | | Unauthorized | |  |  |  |  |  |
| 403 | | Forbidden | |  |  |  |  |  |
| 404 | | Not Found | |  |  |  |  |  |

Tabela 1 - Documentação API CEIS

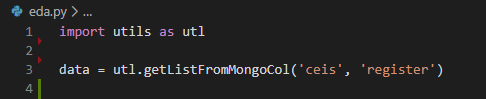
Para obtenção dos dados foi utilizada função “get” da biblioteca “requests” (<https://requests.readthedocs.io/en/master/>) - “requests.get(url, parameters)” - passando a URL da API no primeiro parâmetro e o número da página no segundo parâmetro. Para que não fosse necessário calcular o número total de páginas que seriam retornadas pela API foi implementado um laço que executa a função get e incrementa o número da página enquanto o status HTTP de retorno for igual 200 (<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status/200>). A cada iteração os dados eram persistidos no Banco MongoDB e, para tanto, foi utilizada a biblioteca pymongo. Através da classe MongoClient, da biblioteca pymongo, obtem-se o objeto de conexão com o MongoDB local passando como parâmetro a coleção que será manipulada. Para essas regras foi implementada a classe “getApiData.py”. Essa classe importa, além da biblioteca “requests”, já mencionada anteriormente, a classe utils.py, importada com pseudônimo “utl”, onde está implementada a função de conexão com o MongoDB, chamada “connect\_mongo”.

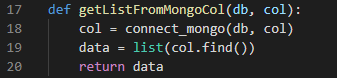




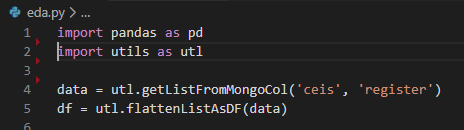
# 3. Processamento/Tratamento de Dados

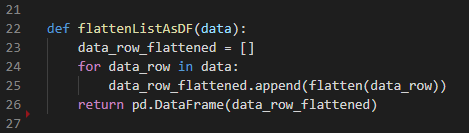
Após a persistência de todo conteúdo obtido através da API do Portal da Transparência no MongoDB, recuperou-se o dataset para o início da fase de validação e tratamento dos dados. Para recuperar o dataset foi criada a função “getListFromMongoCol”, implementada na classe utils.py.



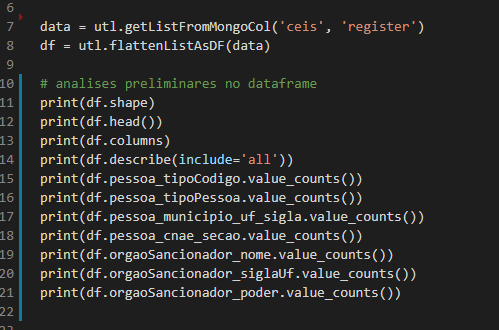


O primeiro ponto tratado é referente à estrutura dos dados. Conforme pode ser observado na Tabela 1, os dados possuem uma estrutura aninhada por categorias (ex: pessoa.municipio.uf.sigla). Como os dados foram persistidos “in natura” o MongoDB preservou essa estrutura. Na recuperação do dataset e posterior amazenamento em um “pandas DataFrame”, notou-se a dificuldade em obtenção de subsets e consultas agrupadas, porque os dados aninhados eram do tipo dict (dicionário), o que impossibilitava utilização de funções do “pandas DataFrame” e, para tanto, seriam necessárias diversas implementações em estruturas de repetição para efetuar as análises. Para contornar esse problema o dataset foi “achatado” e todo conteúdo convertido em “pandas DataFrame”. Para isso foi utilizada a biblioteca “flatten\_json” (<https://pypi.org/project/flatten-json/>). Foi criada uma função para executar essa instrução, chamada “flattenListAsDF” (implementada na classe utils.py).





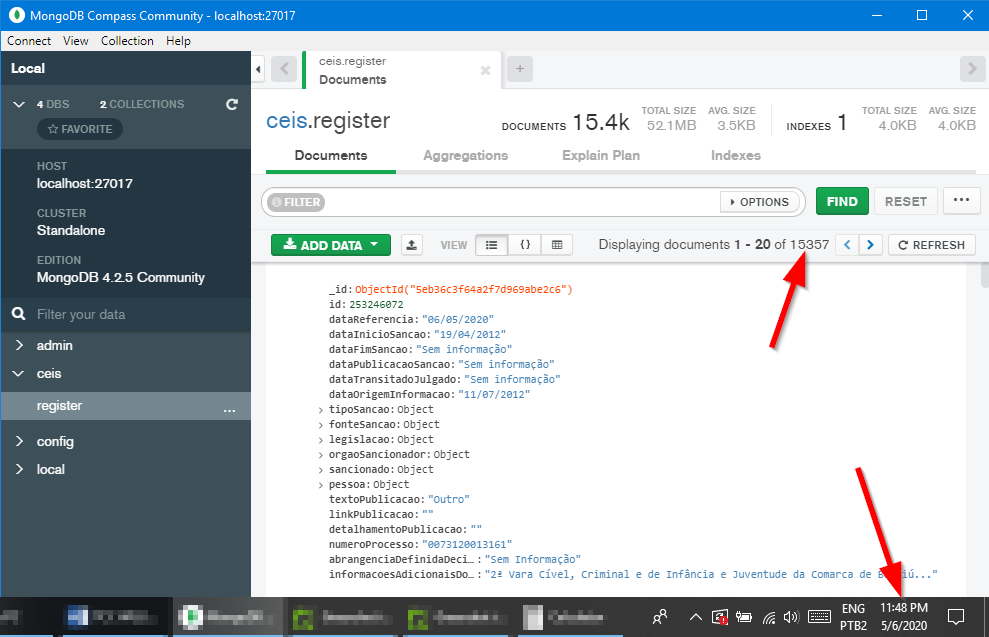
Após o armazenamento dos dados foram executadas análises preliminares no dataframe.

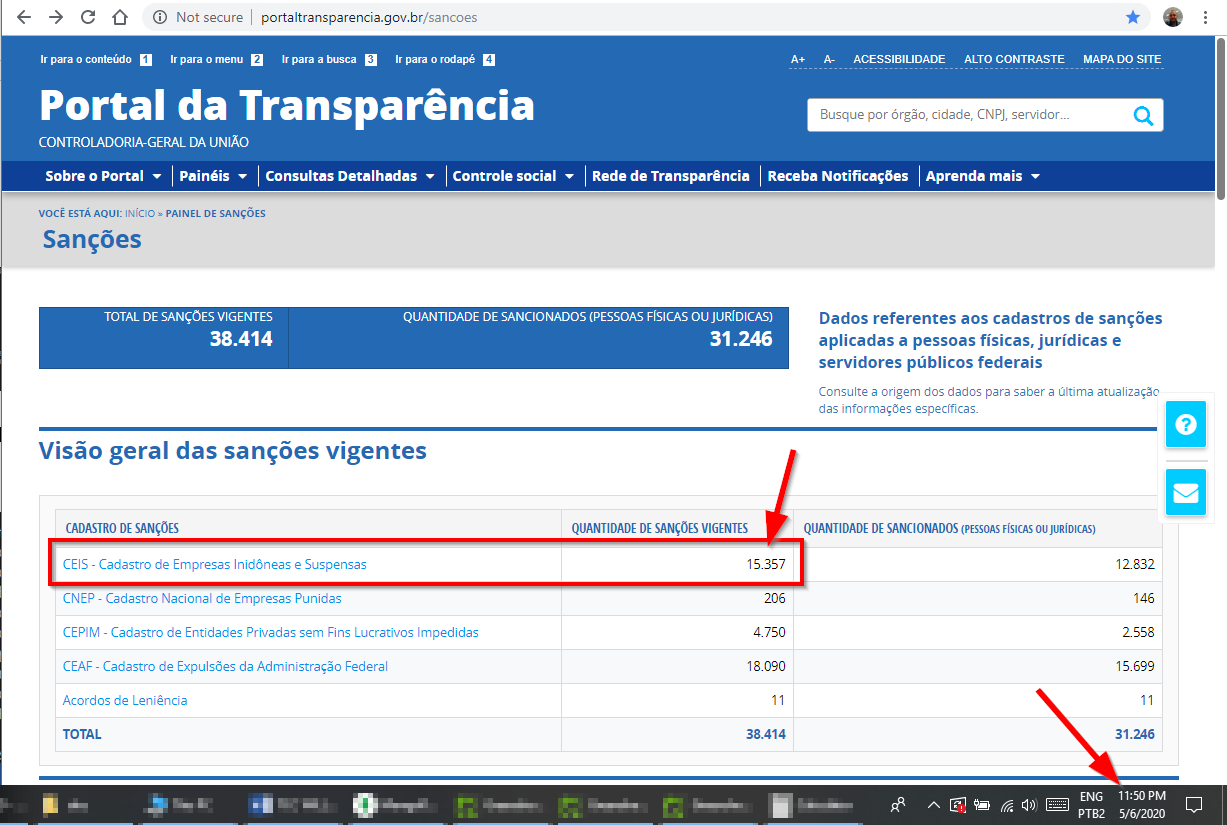


* Propriedade “df.shape”

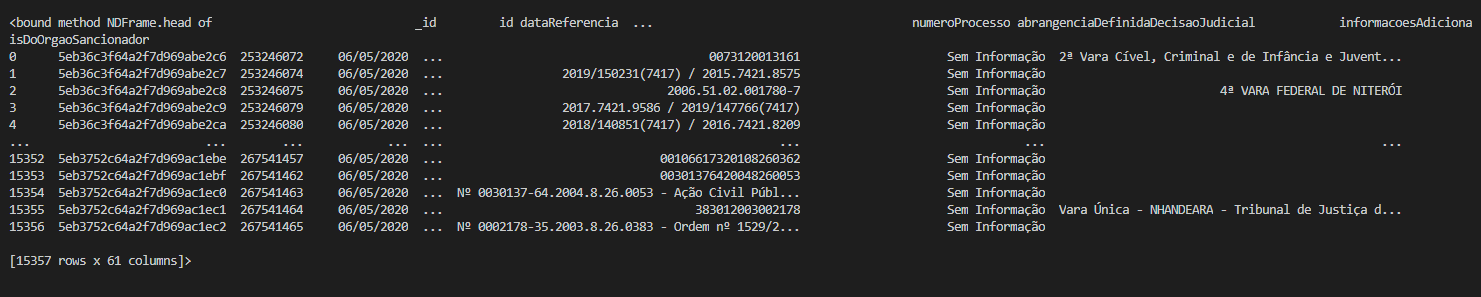


Informa que o frame tem 15.357 linhas e 61 colunas. A quantidade de linhas corrobora com o quantidade total de linhas informada pelo pelo MongoDB e também com o quantidade total de sanções informadas no Portal da Transparência na data em que as análises foram publicadas (06/05/2020). Isso demonstra que os dados analisados possuem um certo grau de confiabilidade em relação à sua origem.



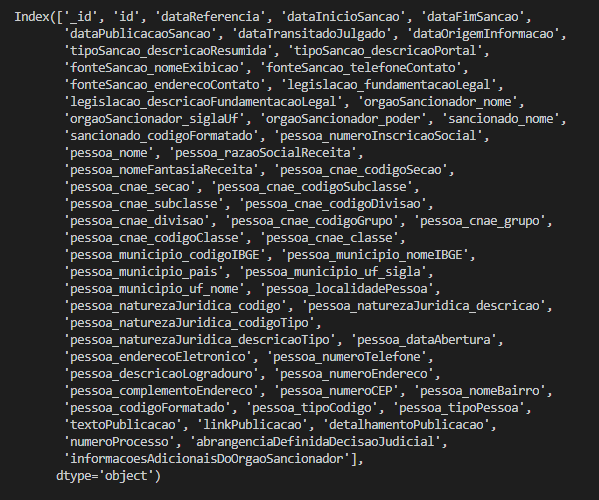


* Função “df.head()”



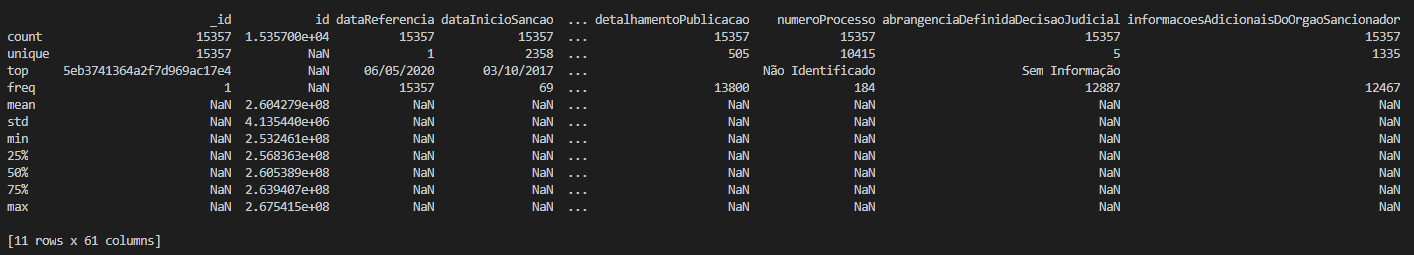
Exibe os 5 primeiros e 5 últimos itens do dataframe.

* Propriedade “df.columns”



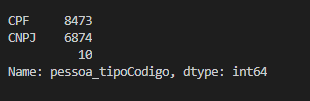
Exibe os nomes das colunas do dataframe.

* Função “df.describe(include=’all’)”

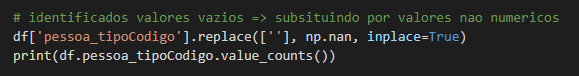


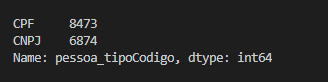
Exibe alguns dados estatísticos sobre o dataframe.

* Função “df.pessoa\_tipoCodigo.value\_counts()”

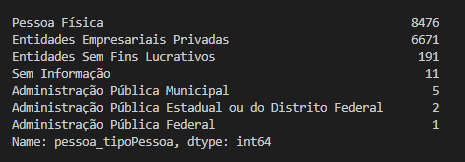


Contagem de valores da série “pessoa\_tipoPessoa”. Nota-se uma quantidade irrelevante de “missing value” nessa série (cerca de 0,06% do total). De qualquer forma esses dados serão tratados como valores não numéricos para que não sejam considerados em eventuais análises (<https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html>).



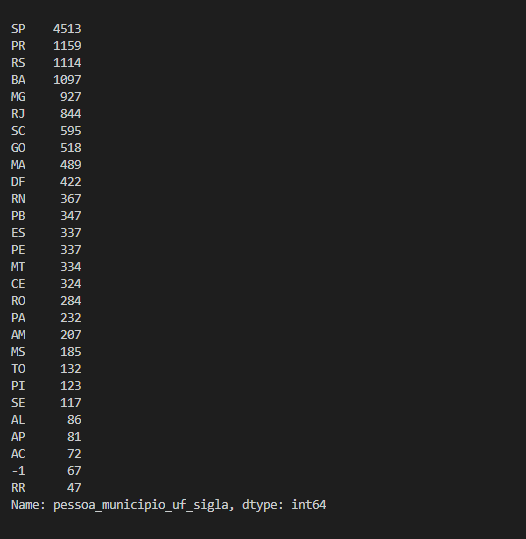


* Função “df.pessoa\_tipoPessoa.value\_counts()”

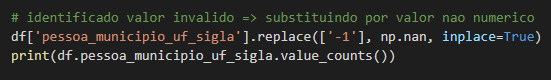


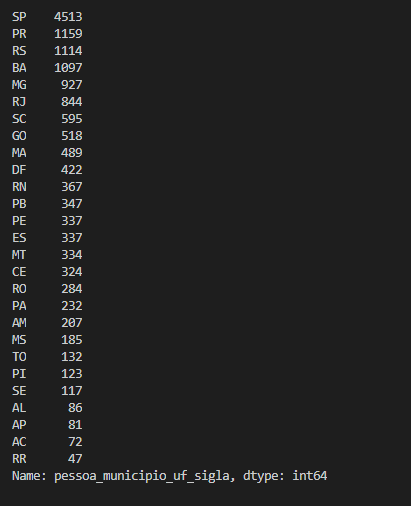
Contagem de valores da série “pessoa\_tipoPessoa”.

* Função “df.pessoa\_municipio\_uf\_sigla.value\_counts()”

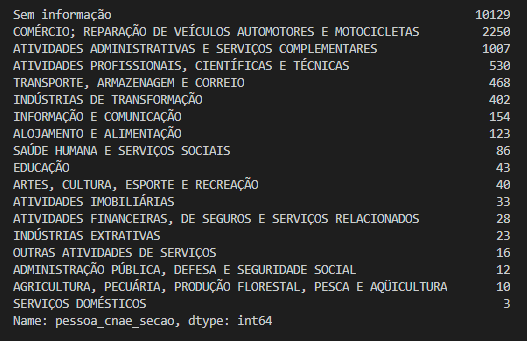


Contagem de valores da série “pessoa\_municipio\_uf\_sigla”. Aqui é possível observar que existem 67 registros onde esse dado está incorreto. Esses dados serão tratados como valores não numéricos para que não sejam considerados em eventuais análises.



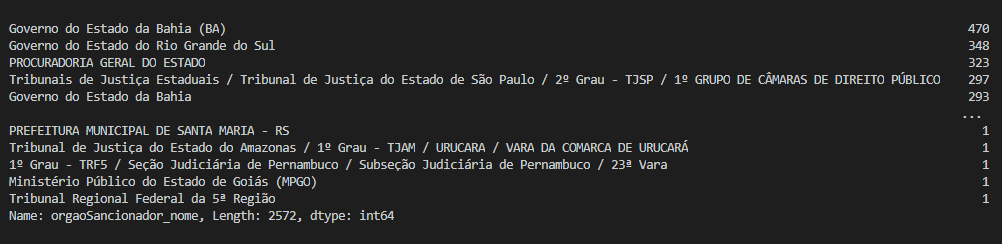


* Função “df.pessoa\_cnae\_secao.value\_counts()”



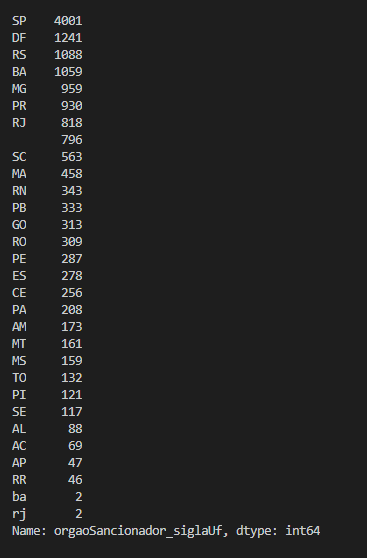
Contagem dos valores da série “pessoa\_cnae\_secao”.

* Função “df.orgaoSancionador\_nome.value\_counts()”

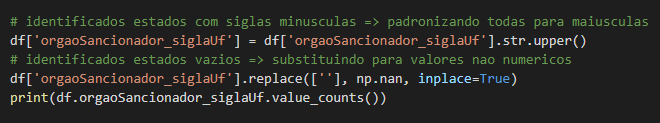


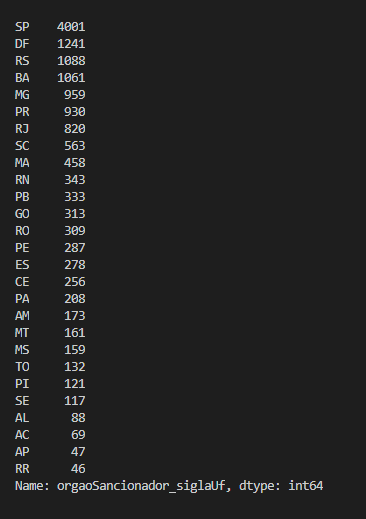
Contagem dos valores da série “orgaoSancionador\_nome”. Aqui é possível observar que os nomes dos órgãos sancionadores não possuem uma padronização, ou seja, caso essa série venha a ser considerada no roteiro de EDA, será necessário aplicar alguma técnica de classificação do texto por similaridade (ex: “cosine similarity” ou “levenshtein distance”).

* Função “df.orgaoSancionador\_siglaUf.value\_counts()”

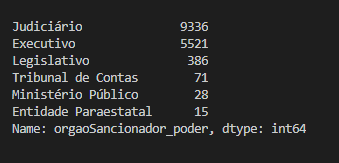


Contagem dos valores da série “orgaoSancionador\_siglaUf”. Aqui identificou-se dois problemas nos dados: 1) algumas siglas estão em letra minúscula (“lowercase”), o que pode interferir nas análises comparativas e agrupamentos; 2) existe um volume considerável de “missing value” nessa série, onde 728 registros, cerca 5% do total, não possuem esse dado. O primeiro problema será tratado transformando todas as siglas minúsculas em maiúsculas (“uppercase”). O segundo problema será tratado inferindo-se valores não numéricos para que esses registros não sejam considerados em eventuais análises.



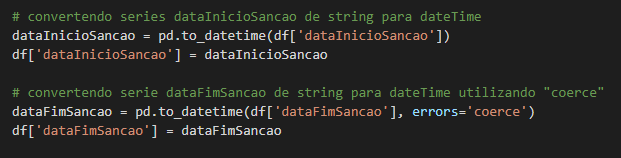


* Função “df.orgaoSancionador\_poder.value\_counts()”



Contagem dos valores da série “orgaoSancionador\_poder”.

Posteriormente às análises preliminares também foi identificada a necessidade de padronização das séries “dataInicioSancao” e “dataFimSancao”, de string para dateTime, pois serão alvo do roteiro de análise (<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.to_datetime.html>).

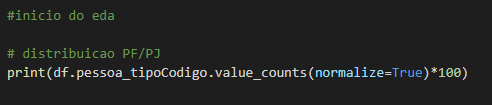


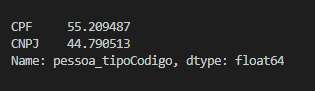
# 4. Análise e Exploração dos Dados

Nessa seção você deve mostrar como foi realizada a análise e exploração dos seus. Mostre as hipóteses levantadas durante essa etapa e os padrões e *insights* identificados.

Com o intuito de obter insights dos dados algumas séries (“pandas.Series”) foram eleitas para serem alvo da EDA (Exploratory Data Analysis). As séries “dataInicioSancao” e “dataFimSancao”, primordialmente, trazem informações da distribuição das sanções ao longo do tempo e seus respectivos de prazos de duração. A série “pessoa\_tipoCodigo” informa a quantidade total de sanções entre pessoas físicas e jurídicas. A série “pessoa\_tipoPessoa” pode informar qual a categorização das entidades que mais incorrem nas sanções. A série “pessoa\_municipio\_uf\_sigla” pode informar em quais estados estão os maiores números de entidades sancionadas. A série “pessoa\_cnae\_secao” pode informar quais os tipos de atividade mais comuns das pessoas jurídicas sancionadas. A série “orgaoSancionador\_nome” pode informar quais os órgãos aplicam mais sanções. A série “orgaoSancionador\_siglaUf” pode informar em quais estados os órgãos aplicam mais sanções. A série “orgaoSancionador\_poder” pode informar quais são os poderes que mais aplicam sanções. Outras séries podem ser incluídas no roteiro das análises conforme os insights obtidos.

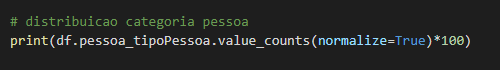
* Série “pessoa\_tipoCodigo”

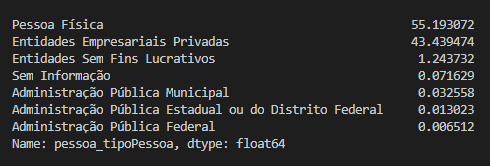




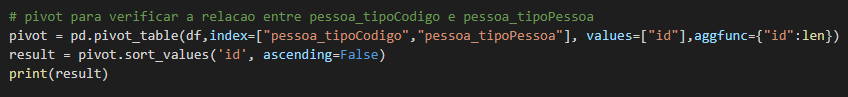
Distribuição de sanções em função dos valores da série “pessoa\_tipoPessoa”. Nota-se que a distribuição é balanceada entre pessoas físicas e jurídicas.

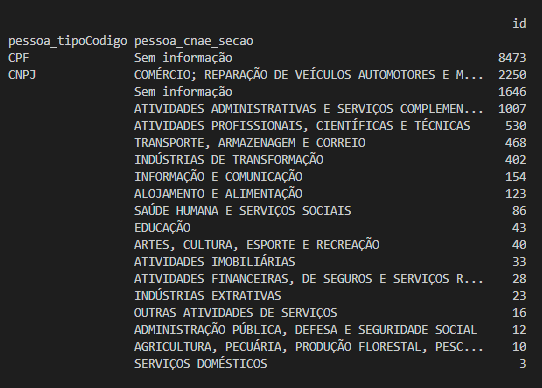
* Série “pessoa\_tipoPessoa”





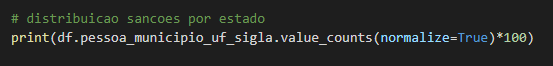
Aqui é possível verificar que o universo de análise é preenchido, quase que em sua totalidade, por Pessoas Físicas e Entidades Empresariais Privadas. Também observa-se que a quantidade de pessoas físicas é praticamente igual ao valor obtido na série anterior, o que leva a crer que, praticamente, todo valor “CPF”, da série “pessoa\_tipoCodigo”, deve estar relacionado ao valor “Pessoa Física”, da série “pessoa\_tipoPessoa”. Para confirmar essa hipótese essas séries foram pivotadas, utilizando a função “pivot\_table”, do Pandas:

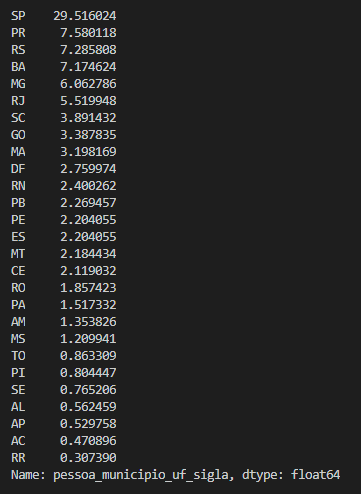




Com a pivotação dos dados ficou claro que todo CPF está, de fato, relacionado à Pessoa Física, e que as Entidades Empresarias Privadas representam, praticamente, a totalidade dos códigos CNPJ. A diferença entre a porcentagem de CPF e Pessoas Físicas (55.209487 - 55.193072 = 0.016415) pode ser devido à regras de arredondamento e/ou aproximação.

* Série “pessoa\_municipio\_uf\_sigla”

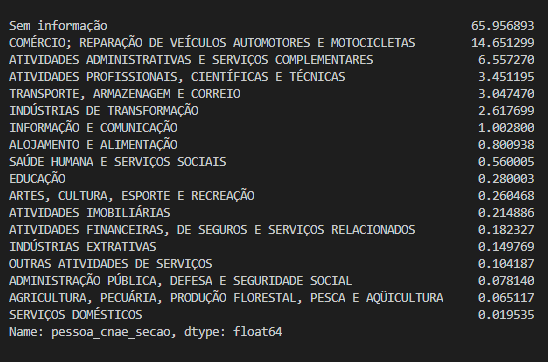




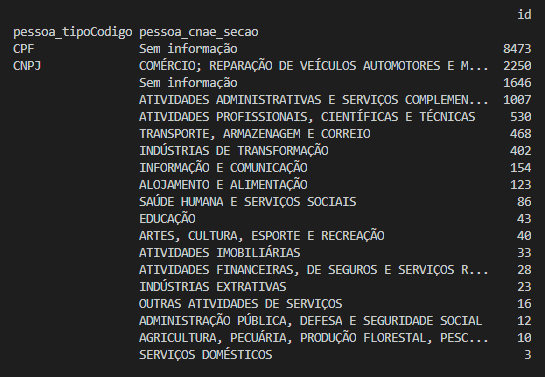
Aqui é possível observar que, somente o estado de São Paulo, concentra cerca de 29% do total de sanções aplicadas. Também observa-se que o “top 5” concentra cerca de 57% do total de sanções, o que é um valor muito expressivo.

* Série “pessoa\_cnae\_secao”



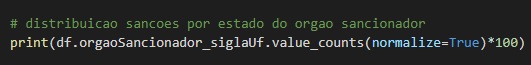


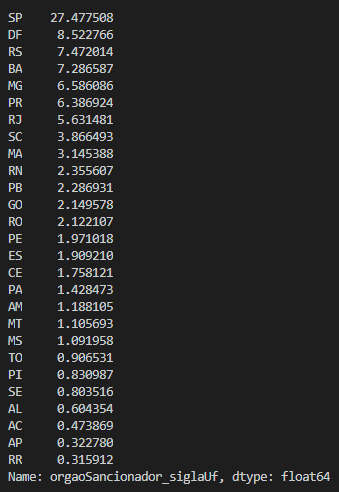
Aqui é possível verificar que, boa parte dos registros não possuem essa informação (quase 66%). Provavelmente isso pode estar relacionado ao fato de que cerca de 55% dos registros são de pessoas físicas, que, realmente, não possuem esse atributo. Mesmo assim, além dos 55% de pessoas físicas, cerca de 11% dos registros também não possuem esse atributo. Aqui, mais uma vez, foi utilizada a função “pivot\_table”, do pandas, para verificar a relação entre tipo de pessoa (física ou jurídica) e o cadastro nacional de atividades econômicas (CNAE) e entender onde estão localizados os registros de CNAE com valor “Sem informação”.



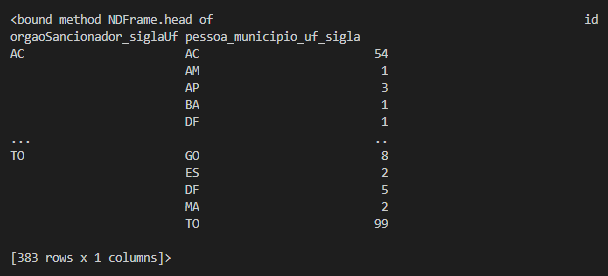
Ratificando o que já havia sido observado anteriormente, é possível verificar que, no escopo de pessoas jurídicas (“pessoa\_tipoCodigo”), também existe um alto volume de entidades sem CNAE (cerca de 24%), o que pode diminuir a assertividade quando os dados são analisados por esse prisma.

* Série “orgaoSancionador\_siglaUf”





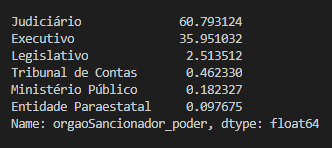
Como já vinha sendo visto nas análises anteriores aqui também o estado de São Paulo lidera a quantidade órgãos sancionadores (órgão responsáveis pelas aplicações das sanções), o que, aparentemente, demonstra a existência de correlação entre o domicílio da entidade sancionada e o domicílio do órgão sancionador. Para confirmar essa possibilidade mais uma vez foi utilizada a pivotação afim de verificar a relação entre estado do órgão sancionador e estado da entidade sancionada.



Aqui foi utilizada a função “df.head()” para que fosse analisada, visualmente, apenas uma amostragem do dataframe. De fato observa-se que órgãos sancionadores aplicam, predominantemente, sanções à entidades do mesmo estado.

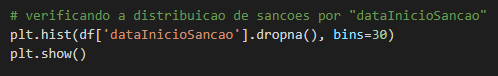
* Série “orgaoSancionador\_poder”

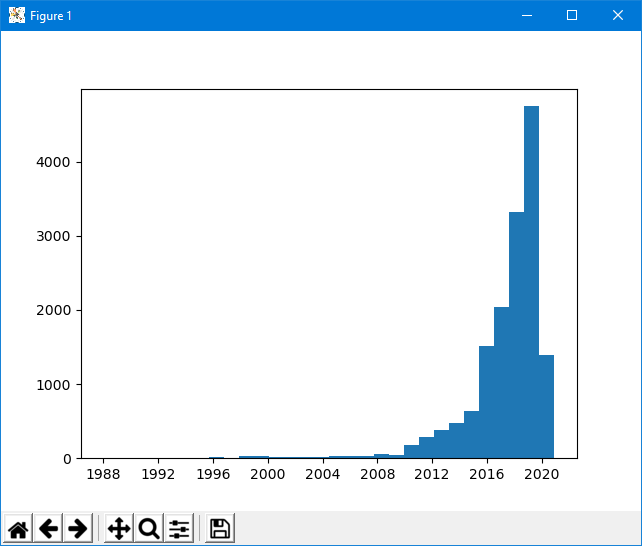




Aqui é possível notar que os poderes judiciários e executivos, juntamente, são predominantes na aplicação das sanções, representando quase 97% do total de sanções aplicadas.

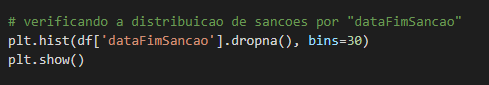
* Série “dataInicioSancao”

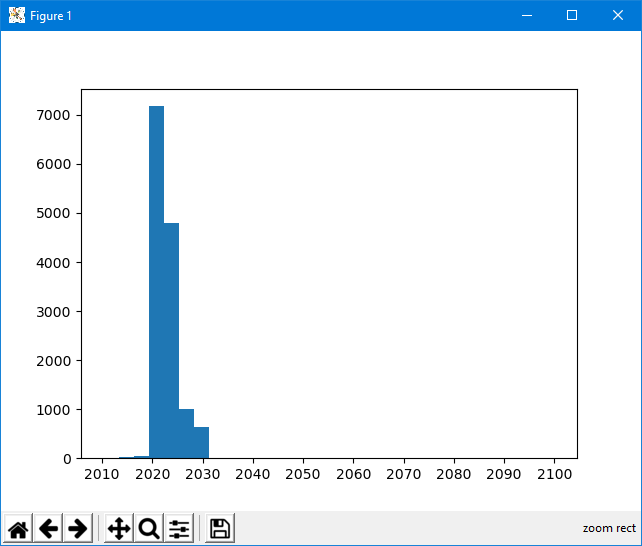




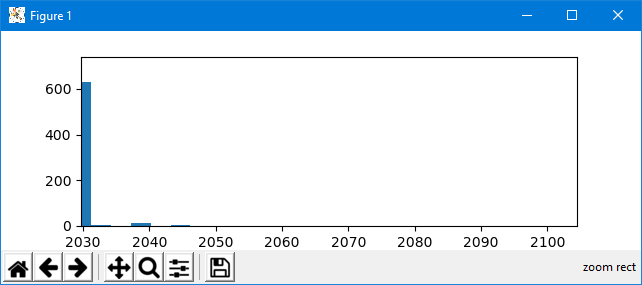
Ná análise da distribuição da série “dataInicioSancao” foi utilizado um histograma (<https://matplotlib.org/3.2.1/api/_as_gen/matplotlib.pyplot.hist.html>). Aqui pode-se notar que, de 1988 até, aproximadamente, 2009, a quantidade de sanções aplicadas é inexpressiva. A partir de 2010 inicia-se uma crescente vertiginosa.

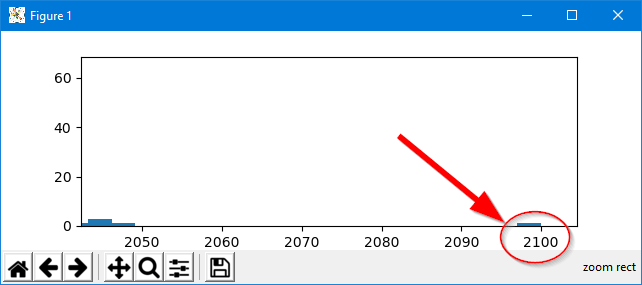
* Série “dataFimSancao”



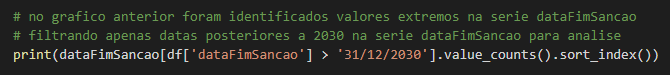


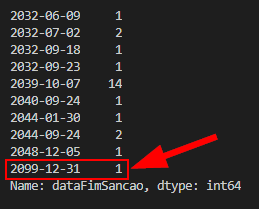
Aqui pode-se observar que os prazos das sanções concentram-se, predominantemente, entre 2020 e 2030. Também nota-se que o eixo x do gráfico foi “plotado” até 2100, o que indica que existem valores nesses intervalos.



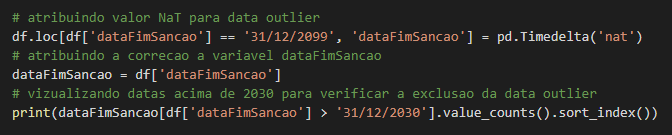


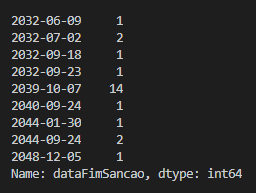
Com o aumento do “zoom” no histograma foi possível verificar que, de fato, existe algum valor (ou alguns valores) com características de outlier.



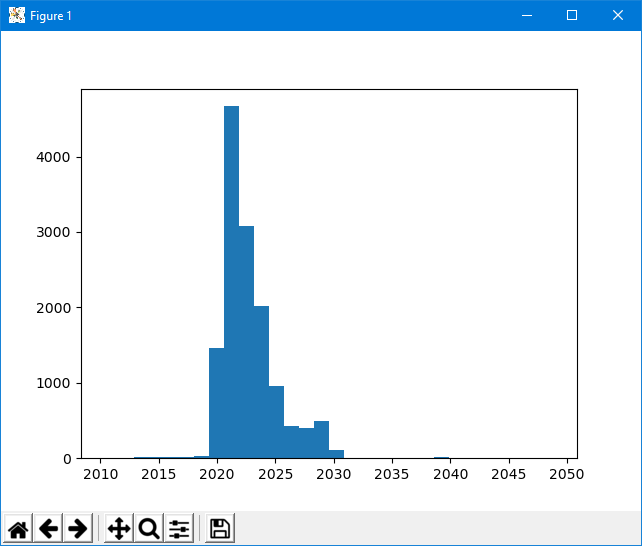


Aqui foi possível identificar o valor outlier. Embora seja apenas um registro foi atribuído um valor “not a time” (NaT) para que o mesmo não interfira nas análises posteriores.



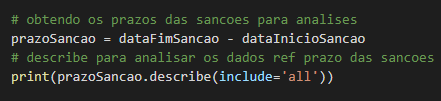


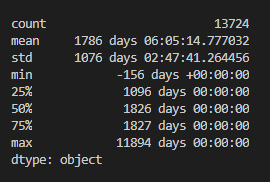
Após essa correção o histograma foi gerado novamente.



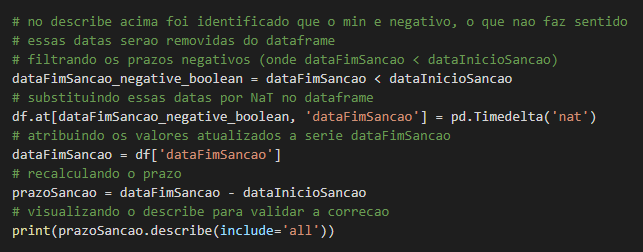
É possível verificar que, após a correção do valor “outlier”, a escala do eixo x diminuiu.

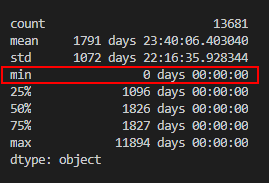
As análises prosseguiram concentrada nas datas, de onde poderiam ser obtidos mais “insights”. Uma das análises propostas é sobre os prazos das sanções.



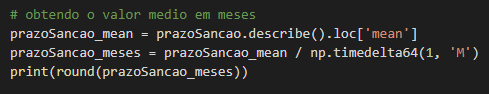


Através da função “describe” da série “prazoSancao” – resultado em dias da subtração entre as séries dataFimSancao e dataInicioSancao – observou-se que o mínimo (min) apresentou valor negativo, o que não faz sentido, pois não existe prazo negativo. Considerando que essa informação é inconsistente foi necessário corrigi-la no dataframe.



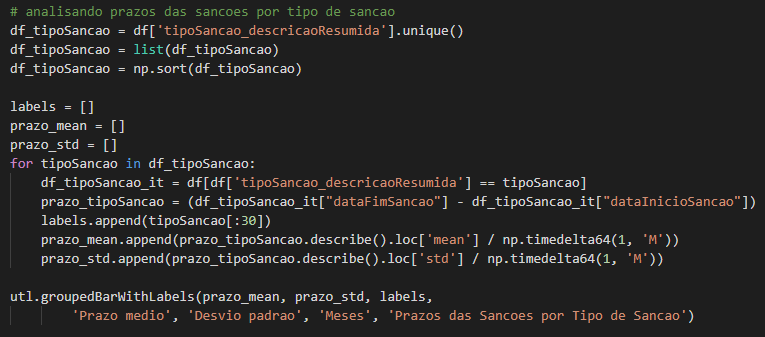


Dando continuidade na exploração após a correção obteve-se o prazo médio das sanções em meses.

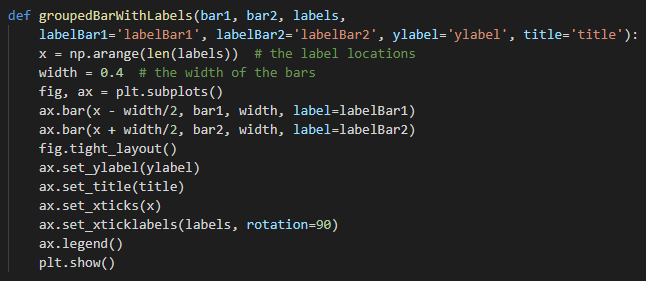


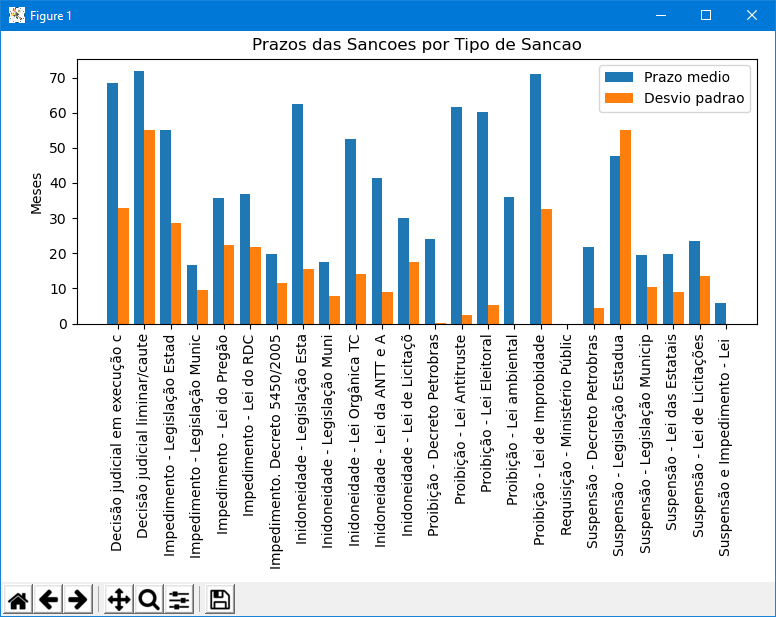


O valor obtido é de 59 meses, ou seja, o prazo médio de uma sanção é de, praticamente, 5 anos, em média. Porém, na função “describe” anterior, foi possível observar que o valor do desvio padrão é muito alto, pois o coeficiente de variação é próximo a 60%. Considerando isso é interessante explorar mais esses dados afim de obter algum “insight”. A primeira análise foi sobre as estatísticas dos prazos em função das sanções. Para isso utilizou-se um gráfico de barras agrupados. Para essa implementação, primeiramente, obteve-se os valores únicos das descrições resumidas de cada sanção e armazenou-os em um array. Esse array foi iterado obtendo-se as respectivas descrições resumidas, prazo médio e desvio padrão de cada sanção.

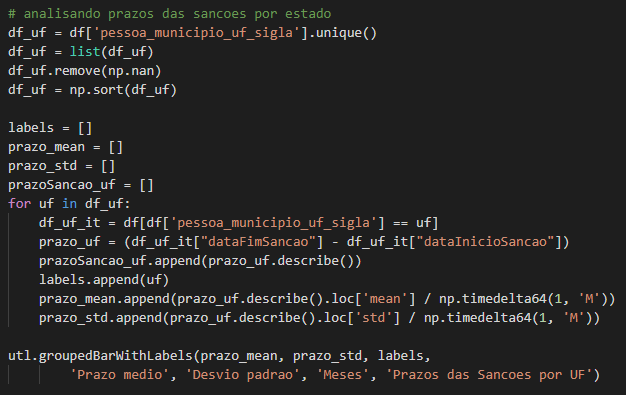


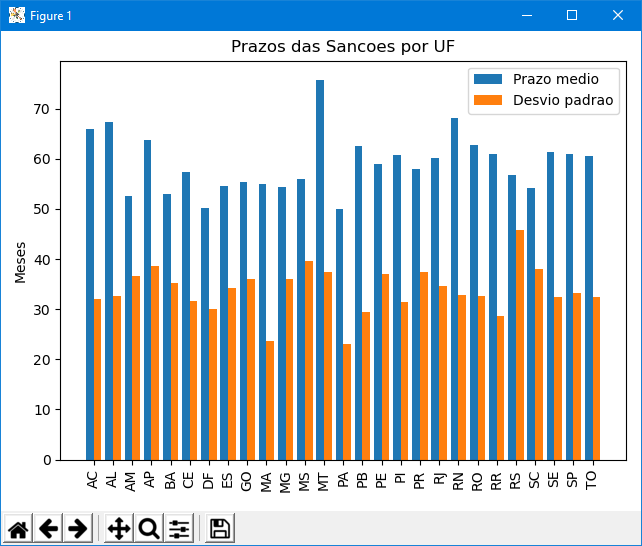
A função de impressão do gráfico (“groupedBarWithLabels”) foi implementada na classe “utils.py”.



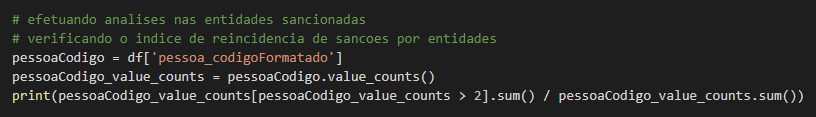


A segunda análise foi sobre as estatísticas dos prazos das sanções em função dos estados das entidades sancionadas. Para isso também utilizou-se um gráfico de barras agrupados. Assim como anteriormente, obteve-se os valores únicos dos estados e armazenando-os em um array. Esse array foi iterado para se obter os estados e respectivos prazo médio e desvio padrão de cada sanção por estado.



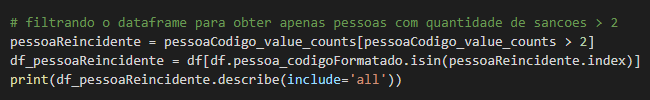


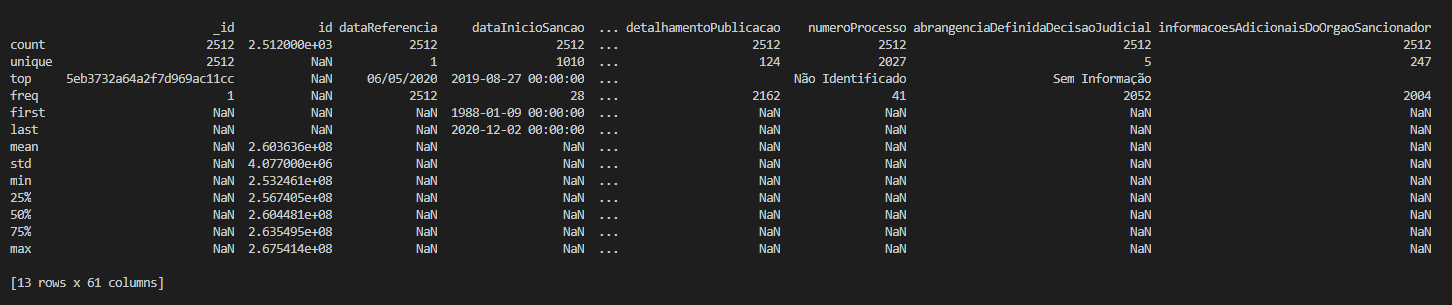
Após essas análises foram realizadas análises exploratórias pela perspectiva das entidades sancionadas. A primeira análise verificou o índice de reincidência de das entidades, ou seja, se uma mesma entidade já sofreu mais de uma sanção.





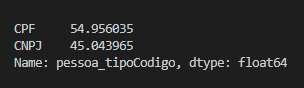
Pode-se verificar que o índice de reincidência e de cerca de 16,3 %, uma taxa razoável que vale a pena ser analisada. O dataframe foi filtrado para obter apenas pessoas reincidentes, para focar a análise nesse escopo.





Verificou-se a distribuição de reincidentes entre pessoa física e jurídica

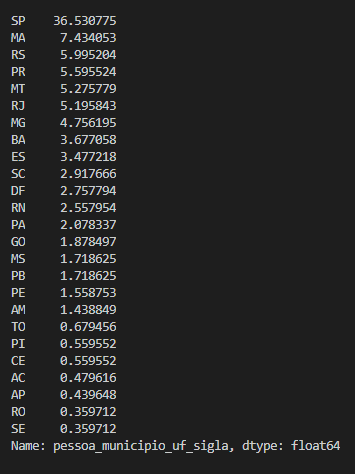




Nota-se que a distribuição de reincidentes é muito parecida com a distribuição geral entre pessoas físicas e jurídicas.

Também foi analisado qual é o estado de origem das entidades reincidentes.

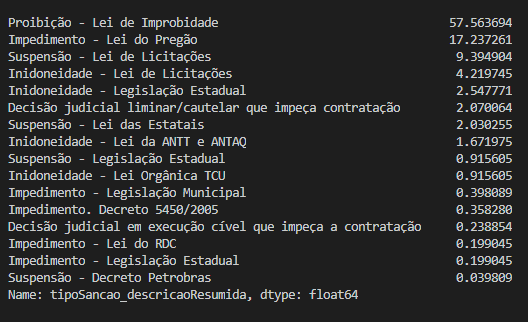




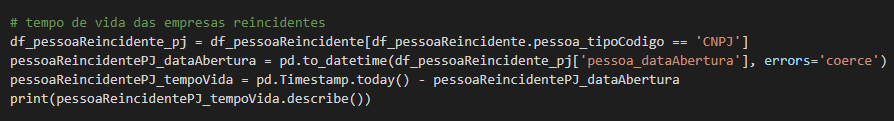
Aqui é possível verificar que o estado de São Paulo extrapolou, que lidera o ranking de sanções, com cerca de 29% do total de sanções aplicadas, supera esse índice no quesito reincidência, sendo responsável por um pouco mais de 36% do quantidade total de reincidências

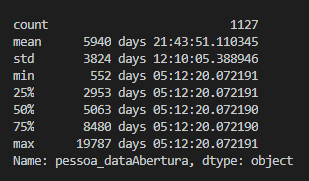
Outro dado verificado é a distribuição de sanções em função das entidades reincidentes.



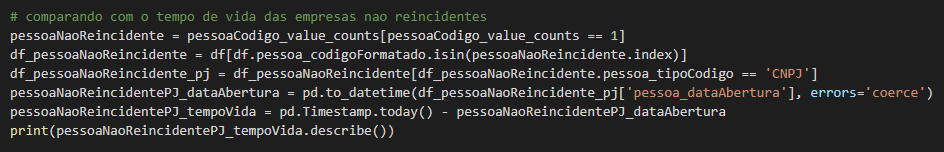


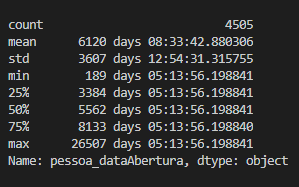
Outro dado importante que foi analisado é o tempo de vida das empresas reincidentes. Obviamente que, nesse caso, esse atributo é exclusivo de pessoas jurídicas.





Pode-se observar que as empresas reincidente possuem, em média, cerca de 16 anos de vida mas, como em outros dados que já foram analisados, aqui também o coeficiente de variação é alto (cerca de 64%), o que faz com o que a média não seja um parâmetro de comparação assertivo. De qualquer forma é interessante que essas informações sejam comparadas com as estatísticas das empresas não reincidentes, para que seja verificado se o padrão é o mesmo ou diferente.





Verificou-se que a média é, praticamente, a mesma, sendo que as empresas não reincidentes possuem, em média, cerca de 6 meses a mais de vida que as empresas reincidentes. Também foi possível verificar que o coeficiente baixou um pouco, para cerca de 59%, mas, mesmo assim, é um coeficiente alto.

# 5. Apresentação dos Resultados

Nessa seção você deve apresentar os resultados obtidos. Apresente gráficos, dahsboards, conte a sua história de forma bastante criativa. Aqui você pode utilizar os modelos de Canvas propostos por Dourard (clique [aqui](https://www.louisdorard.com/machine-learning-canvas)) ou por Vasandani (clique [aqui](https://towardsdatascience.com/a-data-science-workflow-canvas-to-kickstart-your-projects-db62556be4d0)).



# 6. Links

Aqui você deve disponibilizar os links para o vídeo com sua apresentação de 5 minutos e para o repositório contendo os dados utilizados no projeto, scripts criados, etc.

# REFERÊNCIAS

Um projeto de Ciência de Dados não requer revisão bibliográfica. Portanto, a inclusão das referências não é obrigatória. No entanto, caso você deseje incluir referências relacionadas às tecnologias ou às metodologias usadas em seu trabalho, relacione-as de acordo com o modelo a seguir.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.