Continuando a coleta de dados, inserimos o arquivo https://dados.gov.br/dataset/cadastro-nacional-de-reclamacoes-fundamentadas-procons-sindec1 (https://dados.gov.br/dataset/cadastro-nacional-de-reclamacoes-fundamentadas-procons-sindec1), acessado em 25/07/2021. Os arquivos do endereço https://dadosabertos.pgfn.gov.br/2021_trimestre_02/Dados_abertos_Nao_Previdenciario.zip (https://dadosabertos.pgfn.gov.br/2021_trimestre_02/Dados_abertos_Nao_Previdenciario.zip), acessado em 25/07/2021, foram concatenados no notebook 01, resultando no arquivo "concatenadoordenado.csv", de 4GB. Arquivo igualmente "upado" para este notebook.

In [1]:

```
import pandas as pd
from collections import Counter
```

# Inserindo o dataset do Sindec e o preparando para o merge com o "concatenadoordenado" oriundo do notebook 01

In [2]:

```
#Carregando o dataset do Sindec (Procon)
df_procon = pd.read_csv(r'C:\Users\73594253368\Desktop\Curso\Datasets\Procon\CRF2019Dac
```

```
b'Skipping line 117: expected 23 fields, saw 25\nSkipping line 1205: expecte
d 23 fields, saw 25\nSkipping line 1285: expected 23 fields, saw 24\nSkippin
g line 3596: expected 23 fields, saw 25\nSkipping line 4134: expected 23 fie
lds, saw 24\nSkipping line 5483: expected 23 fields, saw 25\nSkipping line 8
028: expected 23 fields, saw 24\nSkipping line 8307: expected 23 fields, saw
24\nSkipping line 8824: expected 23 fields, saw 25\nSkipping line 9516: expe
cted 23 fields, saw 24\nSkipping line 9696: expected 23 fields, saw 24\nSkip
ping line 9698: expected 23 fields, saw 24\nSkipping line 10545: expected 23
fields, saw 25\nSkipping line 11232: expected 23 fields, saw 25\nSkipping li
ne 11503: expected 23 fields, saw 25\nSkipping line 11548: expected 23 field
s, saw 24\nSkipping line 11554: expected 23 fields, saw 25\nSkipping line 11
703: expected 23 fields, saw 24\nSkipping line 12473: expected 23 fields, sa
w 24\nSkipping line 13879: expected 23 fields, saw 24\nSkipping line 16696:
expected 23 fields, saw 24\nSkipping line 17302: expected 23 fields, saw 24
\n'
```

In [3]:

```
df_procon.shape
```

Out[3]:

```
(17555, 23)
```

In [4]:

```
1 df_procon.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17555 entries, 0 to 17554
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   AnoCalendario          17555 non-null  int64  
 1   DataArquivamento       17552 non-null  object 
 2   DataAbertura           17552 non-null  object 
 3   CodigoRegiao           17555 non-null  int64  
 4   Regiao                 17555 non-null  object 
 5   UF                     17555 non-null  object 
 6   strRazaoSocial         17555 non-null  object 
 7   strNomeFantasia        14034 non-null  object 
 8   Tipo                   17555 non-null  int64  
 9   NumeroCNPJ             16028 non-null  float64
 10  RadicalCNPJ            15974 non-null  float64
 11  RazaoSocialRFB         14572 non-null  object 
 12  NomeFantasiaRFB        7651 non-null   object 
 13  CNAEPrincipal          14572 non-null  float64
 14  DescCNAEPrincipal      14502 non-null  object 
 15  Atendida               17555 non-null  object 
 16  CodigoAssunto          17541 non-null  float64
 17  DescricaoAssunto       17541 non-null  object 
 18  CodigoProblema         45 non-null     float64
 19  DescricaoProblema      45 non-null     object 
 20  SexoConsumidor         17546 non-null  object 
 21  FaixaEtariaConsumidor  17555 non-null  object 
 22  CEPConsumidor          13921 non-null  float64
dtypes: float64(6), int64(3), object(14)
memory usage: 3.1+ MB
```

In [5]:

```
df_procon.head()
```

Out[5]:

| | AnoCalendario | DataArquivamento | DataAbertura | CodigoRegiao | Regiao | UF | strR |
|---|---|---|---|---|---|---|---|
| **0** | 2019 | 2019-10-04 11:12:54.000 | 2019-09-02 09:27:15.000 | 1 | Norte | RO | |
| **1** | 2019 | 2019-01-08 10:56:05.000 | 2018-12-04 15:19:18.000 | 1 | Norte | RO | BANCO DO |
| **2** | 2019 | 2019-08-15 15:14:14.000 | 2019-07-16 17:00:46.000 | 1 | Norte | RO | CENTRAIS DE ROI |
| **3** | 2019 | 2019-01-04 11:31:47.000 | 2018-04-19 10:09:02.000 | 1 | Norte | RO | NOVA PRO PROFISSION |
| **4** | 2019 | 2019-01-04 10:26:36.000 | 2018-08-30 09:46:37.000 | 1 | Norte | RO | OI MO EI |

5 rows × 23 columns

In [6]:

```
#Preparando o campo CNPJ para o merge: retirar nulos, sinais e transformar para inteiro
df_procon = df_procon.dropna(subset=['NumeroCNPJ']).dropna(axis=1, how = 'all')
```

In [8]:

```
df_procon.shape
```

Out[8]:

```
(16028, 23)
```

In [ ]:

```

```

In [ ]:

```
df_procon.info()
```

In [ ]:

```
df_procon['NumeroCNPJ'] = df_procon['NumeroCNPJ'].replace('.', '')
df_procon['NumeroCNPJ'] = df_procon['NumeroCNPJ'].replace('/', '')
df_procon['NumeroCNPJ'] = df_procon['NumeroCNPJ'].replace('-', '')
```

In [ ]:

```
df_procon['NumeroCNPJ']
```

In [ ]:

```
df_procon['NumeroCNPJ'] = pd.to_numeric(df_procon['NumeroCNPJ'],downcast='integer')
```

In [ ]:

```
df_procon
```

In [ ]:

```
# Colocando index para permitir visualização, na hora do merge com o dataset da PFN, de
# empresas em DAU
# df_procon_2 é um ProconIndexado
df_procon_2 = df_procon
```

In [ ]:

```
df_procon_2 = df_procon_2.rename_axis('index1').reset_index()
```

In [ ]:

```
df_procon_2
```

In [ ]:

```
df_procon_2.shape
```

In [ ]:

```
df_procon_2.head()
```

# As 27 tabelas de devedores da PFN foram concatenadas no dataset concatenadoordenado.csv. A seguir o tratamento para o merge com a base do Sindec

In [ ]:

```
#Carregando o dataset da PFN
df_pfn = pd.read_csv(r'C:\Users\73594253368\Desktop\Curso\Datasets\Procon\concatenadoor
```

In [ ]:

```
df_pfn.info()
```

In [ ]:

```
df_pfn.rename(columns = {'TIPO_PESSOA': 'Tipo_Pessoa'}, inplace=True)
```

Restringimos o data frame a apenas duas colunas, pois nosso objetivo com os dados da PFN é, a partir dos dados das demandas no Sindec, sabermos se a empresa constante nas demandas é listada, também, como devedora da Fazenda Nacional.

In [ ]:

```
df_pfn_colunas = df_pfn[['CPF_CNPJ','Tipo_Pessoa']]
```

In [ ]:

```
df_pfn_colunas.head()
```

In [ ]:

```
#Removendo os nulos
df_pfn_colunas = df_pfn_colunas.dropna(subset=['CPF_CNPJ']).dropna(axis=1, how = 'all')
df_pfn_colunas = df_pfn_colunas.dropna(subset=['Tipo_Pessoa']).dropna(axis=1, how = 'al
```

In [ ]:

```
#Retirando as pessoas físicas
df_pfn_colunas = df_pfn_colunas[df_pfn_colunas.Tipo_Pessoa.str.contains('Pessoa jurídic
df_pfn_colunas.rename(columns = {'CPF_CNPJ': 'CNPJ'}, inplace=True)
```

In [ ]:

```
#Limpando os CNPJs
df_pfn_colunas['CNPJ'] = df_pfn_colunas['CNPJ'].str.replace('.', '')
df_pfn_colunas['CNPJ'] = df_pfn_colunas['CNPJ'].str.replace('/', '')
df_pfn_colunas['CNPJ'] = df_pfn_colunas['CNPJ'].str.replace('-', '')
df_pfn_colunas['CNPJ'] = pd.to_numeric(df_pfn_colunas['CNPJ'],downcast='integer')
```

In [ ]:

```
df_pfn_colunas.info()
```

In [ ]:

```
df_pfn_colunas['Tipo_Pessoa'].value_counts()
```

In [ ]:

```
df_pfn_colunas.info()
```

In [ ]:

```
df_pfn_colunas.shape
```

In [ ]:

```

```

In [ ]:

```

```

# Merge Procon Indexado com PFN duas colunas

In [ ]:

```

df_merged = pd.merge(df_procon_2, df_pfn_colunas, how='left',
                     left_on=['NumeroCNPJ'],
                     right_on =['CNPJ'])
```

In [ ]:

```
df_merged.shape
```

In [ ]:

```

```

In [ ]:

```
df_merged.info()
```

In [ ]:

```

```

In [ ]:

```
#Retirando os registros duplicados
df_merged = df_merged.drop_duplicates()
```

In [ ]:

```

```

In [ ]:

```
df_merged.info()
```

In [ ]:

```

```

In [ ]:

```
# Exportanto o dataframe criado após os merges pra a etapa seguinte, a de tratamento do
# df_merged.to_csv(r'C:\Users\73594253368\Desktop\Curso\Datasets\Procon\df_merged.csv')
```

In [ ]:

```

```