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]:

```
1 df_procon.head()
```

Out[5]:

	AnoCalendario	DataArquivamento	DataAbertura	CodigoRegiao	Regiao	UF	strF
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 DC
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 PRC 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]:

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

In [7]:

```
1 df_procon.shape
```

Out[7]:

```
(16028, 23)
```

In [8]:

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16028 entries, 0 to 17554
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   AnoCalendario          16028 non-null  int64
 1   DataArquivamento       16028 non-null  object
 2   DataAbertura           16028 non-null  object
 3   CodigoRegiao           16028 non-null  int64
 4   Regiao                 16028 non-null  object
 5   UF                     16028 non-null  object
 6   strRazaoSocial         16028 non-null  object
 7   strNomeFantasia        13123 non-null  object
 8   Tipo                   16028 non-null  int64
 9   NumeroCNPJ             16028 non-null  float64
 10  RadicalCNPJ            15974 non-null  float64
 11  RazaoSocialRFB         14311 non-null  object
 12  NomeFantasiaRFB        7470 non-null   object
 13  CNAEPrincipal          14311 non-null  float64
 14  DescCNAEPrincipal      14241 non-null  object
 15  Atendida               16028 non-null  object
 16  CodigoAssunto          16014 non-null  float64
 17  DescricaoAssunto       16014 non-null  object
 18  CodigoProblema         43 non-null     float64
 19  DescricaoProblema      43 non-null     object
 20  SexoConsumidor         16022 non-null  object
 21  FaixaEtariaConsumidor  16028 non-null  object
 22  CEPConsumidor          13003 non-null  float64
dtypes: float64(6), int64(3), object(14)
memory usage: 2.9+ MB
```

In [9]:

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

In [10]:

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

Out[10]:

```
0        4.043254e+13
1        1.910000e+02
2        5.914650e+12
3        1.311034e+13
4        5.423963e+12
            ...
17550    3.325432e+13
17551    3.130170e+12
17552    3.603050e+11
17553    3.603050e+11
17554    6.074695e+13
Name: NumeroCNPJ, Length: 16028, dtype: float64
```

In [11]:

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

In [12]:

```
df_procon
```

Out[12]:

	AnoCalendario	DataArquivamento	DataAbertura	CodigoRegiao	Regiao	UF	st
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 D
2	2019	2019-08-15 15:14:14.000	2019-07-16 17:00:46.000	1	Norte	RO	CENTRAI DE R
3	2019	2019-01-04 11:31:47.000	2018-04-19 10:09:02.000	1	Norte	RO	NOVA PF PROFISSIC
4	2019	2019-01-04 10:26:36.000	2018-08-30 09:46:37.000	1	Norte	RO	OI M
...	...	...	...	...	...	...	
17550	2019	2019-04-03 09:41:41.000	2019-03-07 08:19:35.000	1	Norte	RO	BANCO LO BANC
17551	2019	2019-04-04 14:31:20.000	2019-03-08 10:44:47.000	1	Norte	RO	E ADMINIS CA
17552	2019	2019-03-27 14:16:44.000	2019-02-20 16:10:15.000	1	Norte	RO	CAIXA
17553	2019	2019-03-13 12:15:18.000	2019-02-15 15:35:18.000	1	Norte	RO	CAIXA
17554	2019	2019-04-23 11:29:35.000	2019-02-20 09:15:26.000	1	Norte	RO	BANCO BF

16028 rows × 23 columns

In [13]:

```
# 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 [14]:

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

In [15]:

```
df_procon_2
```

Out[15]:

	index1	AnoCalendario	DataArquivamento	DataAbertura	CodigoRegiao	Regiao	UF	
0	0	2019	2019-10-04 11:12:54.000	2019-09-02 09:27:15.000	1	Norte	RO	
1	1	2019	2019-01-08 10:56:05.000	2018-12-04 15:19:18.000	1	Norte	RO	
2	2	2019	2019-08-15 15:14:14.000	2019-07-16 17:00:46.000	1	Norte	RO	(
3	3	2019	2019-01-04 11:31:47.000	2018-04-19 10:09:02.000	1	Norte	RO	PF
4	4	2019	2019-01-04 10:26:36.000	2018-08-30 09:46:37.000	1	Norte	RO	
...	...	...	...	...	...	...	...	
16023	17550	2019	2019-04-03 09:41:41.000	2019-03-07 08:19:35.000	1	Norte	RO	BA
16024	17551	2019	2019-04-04 14:31:20.000	2019-03-08 10:44:47.000	1	Norte	RO	
16025	17552	2019	2019-03-27 14:16:44.000	2019-02-20 16:10:15.000	1	Norte	RO	
16026	17553	2019	2019-03-13 12:15:18.000	2019-02-15 15:35:18.000	1	Norte	RO	
16027	17554	2019	2019-04-23 11:29:35.000	2019-02-20 09:15:26.000	1	Norte	RO	B

16028 rows × 24 columns

In [16]:

```
df_procon_2.shape
```

Out[16]:

```
(16028, 24)
```

In [17]:

```
1 df_procon_2.head()
```

Out[17]:

	index1	AnoCalendario	DataArquivamento	DataAbertura	CodigoRegiao	Regiao	UF	strRazaoSoc
0	0	2019	2019-10-04 11:12:54.000	2019-09-02 09:27:15.000	1	Norte	RO	CLARO S
1	1	2019	2019-01-08 10:56:05.000	2018-12-04 15:19:18.000	1	Norte	RO	BANCO DO BRASIL
2	2	2019	2019-08-15 15:14:14.000	2019-07-16 17:00:46.000	1	Norte	RO	CENTRAIS ELETRIC DE RONDONIA S
3	3	2019	2019-01-04 11:31:47.000	2018-04-19 10:09:02.000	1	Norte	RO	NOVA PROFISSION CURS PROFISSIONALIZANT TF

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 [18]:

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

In [19]:

```
1 df_pfn.info()
```

```
RangeIndex: 11258612 entries, 0 to 11258611
Data columns (total 14 columns):
 #   Column                   Dtype
---  ------                   -----
 0   Unnamed: 0               int64
 1   CPF_CNPJ                 object
 2   TIPO_PESSOA              object
 3   TIPO_DEVEDOR             object
 4   NOME_DEVEDOR             object
 5   UF_UNIDADE_RESPONSAVEL   object
 6   UNIDADE_RESPONSAVEL      object
 7   NUMERO_INSCRICAO         int64
 8   TIPO_SITUACAO_INSCRICAO  object
 9   SITUACAO_INSCRICAO       object
 10  RECEITA_PRINCIPAL        object
 11  DATA_INSCRICAO           object
 12  INDICADOR_AJUIZADO       object
 13  VALOR_CONSOLIDADO        float64

dtypes: float64(1), int64(2), object(11)
memory usage: 1.2+ GB
```

In [20]:

```
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 [21]:

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

In [22]:

```
df_pfn_colunas.head()
```

Out[22]:

	CPF_CNPJ	Tipo_Pessoa
0	XXX631.543XX	Pessoa física
1	22.138.364/0001-75	Pessoa jurídica
2	XXX754.642XX	Pessoa física
3	XXX236.462XX	Pessoa física
4	34.710.061/0001-64	Pessoa jurídica

In [23]:

```
#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 [24]:

```
#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 [25]:

```
#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 [26]:

```
df_pfn_colunas.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6654391 entries, 1 to 11238011
Data columns (total 2 columns):
 #   Column       Dtype 
---  ------       ----- 
 0   CNPJ         int64 
 1   Tipo_Pessoa  object
dtypes: int64(1), object(1)
memory usage: 152.3+ MB
```

In [27]:

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

Out[27]:

```
Pessoa jurídica    6654391
Name: Tipo_Pessoa, dtype: int64
```

In [28]:

```
df_pfn_colunas.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6654391 entries, 1 to 11238011
Data columns (total 2 columns):
 #   Column       Dtype 
---  ------       ----- 
 0   CNPJ         int64 
 1   Tipo_Pessoa  object
dtypes: int64(1), object(1)
memory usage: 152.3+ MB
```

In [29]:

```
df_pfn_colunas.shape
```

Out[29]:

```
(6654391, 2)
```

Merge Procon Indexado com PFN duas colunas

In [30]:

```

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

In [31]:

```
1 df_merged.shape
```

Out[31]:

```
(121662, 26)
```

In [32]:

```
1 df_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 121662 entries, 0 to 121661
Data columns (total 26 columns):
 #   Column                 Non-Null Count   Dtype  
---  ------                 --------------   -----  
 0   index1                 121662 non-null  int64  
 1   AnoCalendario          121662 non-null  int64  
 2   DataArquivamento       121662 non-null  object 
 3   DataAbertura           121662 non-null  object 
 4   CodigoRegiao           121662 non-null  int64  
 5   Regiao                 121662 non-null  object 
 6   UF                     121662 non-null  object 
 7   strRazaoSocial         121662 non-null  object 
 8   strNomeFantasia        106268 non-null  object 
 9   Tipo                   121662 non-null  int64  
 10  NumeroCNPJ             121662 non-null  int64  
 11  RadicalCNPJ            121608 non-null  float64
 12  RazaoSocialRFB         119303 non-null  object 
 13  NomeFantasiaRFB        75294 non-null   object 
 14  CNAEPrincipal          119303 non-null  float64
 15  DescCNAEPrincipal      119207 non-null  object 
 16  Atendida               121662 non-null  object 
 17  CodigoAssunto          121542 non-null  float64
 18  DescricaoAssunto       121542 non-null  object 
 19  CodigoProblema         302 non-null     float64
 20  DescricaoProblema      302 non-null     object 
 21  SexoConsumidor         121574 non-null  object 
 22  FaixaEtariaConsumidor  121662 non-null  object 
 23  CEPConsumidor          95910 non-null   float64
 24  CNPJ                   111288 non-null  float64
 25  Tipo_Pessoa            111288 non-null  object 
dtypes: float64(6), int64(5), object(15)
memory usage: 25.1+ MB
```

In [33]:

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

In [34]:

```
1 df_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16028 entries, 0 to 121660
Data columns (total 26 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   index1                 16028 non-null  int64
 1   AnoCalendario          16028 non-null  int64
 2   DataArquivamento       16028 non-null  object
 3   DataAbertura           16028 non-null  object
 4   CodigoRegiao           16028 non-null  int64
 5   Regiao                 16028 non-null  object
 6   UF                     16028 non-null  object
 7   strRazaoSocial         16028 non-null  object
 8   strNomeFantasia        13123 non-null  object
 9   Tipo                   16028 non-null  int64
 10  NumeroCNPJ             16028 non-null  int64
 11  RadicalCNPJ            15974 non-null  float64
 12  RazaoSocialRFB         14311 non-null  object
 13  NomeFantasiaRFB        7470 non-null   object
 14  CNAEPrincipal          14311 non-null  float64
 15  DescCNAEPrincipal      14241 non-null  object
 16  Atendida               16028 non-null  object
 17  CodigoAssunto          16014 non-null  float64
 18  DescricaoAssunto       16014 non-null  object
 19  CodigoProblema         43 non-null     float64
 20  DescricaoProblema      43 non-null     object
 21  SexoConsumidor         16022 non-null  object
 22  FaixaEtariaConsumidor  16028 non-null  object
 23  CEPConsumidor          13003 non-null  float64
 24  CNPJ                   5654 non-null   float64
 25  Tipo_Pessoa            5654 non-null   object
dtypes: float64(6), int64(5), object(15)
memory usage: 3.3+ MB
```

In [36]:

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

In [ ]:

```
1
```