Trabalho Final - Ciência de Dados - Enap / 2021

Tratamento das bases

Aluno: William Lapa Santos Filho

Apresentação do Dataset:

Trata-se de base dados PÚBLICA coletada do portal Grandes Números DIRPF elaborado pelo CETAD (Centro de Estudos Tributários e Aduaneiros) vinculado à Receita Federal do Brasil. Foram extraídas as informações das DIRPFs entregues por município de jurisdição dos contribuintes dos anos-calendários de 2015 a 2018.

Fonte: https://receita.economia.gov.br/dados/receitadata/estudos-e-tributarios-e-aduaneiros/estudos-e-estatisticas/11-08-2014-grandes-numeros-dirpf/grandes-numeros-dirpf-capa (https://receita.economia.gov.br/dados/receitadata/estudos-e-tributarios-e-aduaneiros/estudos-e-estatisticas/11-08-2014-grandes-numeros-dirpf/grandes-numeros-dirpf-capa)

Também utilizaremos a base de população, PIB e PIB per Capita por município obtida no site do IBGE https://sidra.ibge.gov.br/ (https://sidra.ibge.gov.br/)

Motivação Pessoal:

Tratam-se de informações relevantes para análise da evolução de rendimentos e bens declarados para a administração tributária federal, cujo escopo faz parte da minha atividade profissional.

Importando as principais bibliotecas

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Este notebook fará apenas a importação, tratamento e exportação dos arquivos para CSV com o objetivo de utilizar no notebook "William_Trabalho_Final" que segue em anexo.

Dataset - Importando a base de dados e realizando tratamento iniciais

In [2]:

```
# Importando a bse de dados da RFB
# fonte: https://receita.economia.gov.br/dados/receitadata/estudos-e-tributarios-e-aduaneiros/estudos-e-estatisticas/11-08-2014-grandes-numeros-dirpf
# concatenando os arquivos de 2015 a 2018.
# As colunas dos arquivos originais estão milhões e por isso os valores serão convertidos.

# As bases originais que serão importadas abaixo estão na pasta /DadosOriginais/

dataset2015=pd.read_excel('Rendimentos2015.xlsx', skiprows=10, decimal=',')
dataset2015.set_axis(['Tipo de Formulário da Declaração', 'Município do Declarante', 'Qtde Declarantes', 'Rendim. Tribut.', 'Rendim. Tribut. Exclus.',
                              'Rendim. Isentos', 'Contrib. Previd.', 'Dependentes', 'Instrução', 'Médicas', 'Livro Caixa', 'Pensão Alimen.', 'Desc. Padrão', 'Base de Cálculo (RTL)', 'Imposto Devido', 'Imposto Pago', 'Imposto a Pagar', 'Imposto a Restituir', 'Bens e Direitos', 'Dívidas e Ônus', 'Doações e Heranças']
                             , axis=1, inplace=True)
dataset2015 = dataset2015.groupby(['Município do Declarante'], as_index=False).sum()
dataset2015['ano'] = 2015


dataset2016=pd.read_excel('Rendimentos2016.xlsx', skiprows=10, decimal=',')
dataset2016.set_axis(['Tipo de Formulário da Declaração', 'Município do Declarante', 'Qtde Declarantes', 'Rendim. Tribut.', 'Rendim. Tribut. Exclus.',
                              'Rendim. Isentos', 'Contrib. Previd.', 'Dependentes', 'Instrução', 'Médicas', 'Livro Caixa', 'Pensão Alimen.', 'Desc. Padrão', 'Base de Cálculo (RTL)', 'Imposto Devido', 'Imposto Pago', 'Imposto a Pagar', 'Imposto a Restituir', 'Bens e Direitos', 'Dívidas e Ônus', 'Doações e Heranças']
                             , axis=1, inplace=True)
dataset2016 = dataset2016.groupby(['Município do Declarante'], as_index=False).sum()
dataset2016['ano'] = 2016

# no dataset 2017 foram encontrados valores 'x' como NAN e por isso usamos a conversao no argumento do read_excel
dataset2017=pd.read_excel('Rendimentos2017.xlsx', skiprows=10, decimal=',', na_values='x')
dataset2017.set_axis(['Tipo de Formulário da Declaração', 'Município do Declarante', 'Qtde Declarantes', 'Rendim. Tribut.', 'Rendim. Tribut. Exclus.',
                              'Rendim. Isentos', 'Contrib. Previd.', 'Dependentes', 'Instrução', 'Médicas', 'Livro Caixa', 'Pensão Alimen.', 'Desc. Padrão', 'Base de Cálculo (RTL)', 'Imposto Devido', 'Imposto Pago', 'Imposto a Pagar', 'Imposto a Restituir', 'Bens e Direitos', 'Dívidas e Ônus', 'Doações e Heranças']
                             , axis=1, inplace=True)
dataset2017 = dataset2017.groupby(['Município do Declarante'], as_index=False).sum()
dataset2017['ano'] = 2017

dataset2018=pd.read_excel('Rendimentos2018.xlsx', skiprows=10, decimal=',')
dataset2018.set_axis(['Tipo de Formulário da Declaração', 'Município do Declarante', 'Qtde Declarantes', 'Rendim. Tribut.', 'Rendim. Tribut. Exclus.',
                              'Rendim. Isentos', 'Contrib. Previd.', 'Dependentes', 'Instrução', 'Médicas', 'Livro Caixa', 'Pensão Alimen.', 'Desc. Padrão', 'Base de Cálculo (RTL)', 'Imposto Devido', 'Imposto Pago', 'Imposto a Pagar', 'Imposto a Restituir', 'Bens e Direitos', 'Dívidas e Ônus', 'Doações e Heranças']
                             , axis=1, inplace=True)
dataset2018 = dataset2018.groupby(['Município do Declarante'], as_index=False).sum()
dataset2018['ano'] = 2018

df = pd.concat([dataset2015,dataset2016,dataset2017, dataset2018])
```

```
#Extraindo UF e nome do municipio
df['Municipio'] = df['Município do Declarante'].apply(lambda x: x.split('-')[0].strip
())
df['UF'] = df['Município do Declarante'].apply(lambda x: x[-2:].strip())

# Normalizando nome dos municipios
df['Municipio'] = df['Municipio'].str.normalize('NFKD').str.encode('ascii', errors='ign
ore').str.decode('utf-8')
df['Municipio'] = df['Municipio'].apply(lambda x: x.upper())

del df['Município do Declarante']
df['ano'] = df['ano'].astype('int64')
df['Qtde Declarantes'] = df['Qtde Declarantes'].astype('int64')

#Criando chave comum para cruzar com as bases do Pib, Pib per Capita e População
df['chave'] = df['ano'].astype(str)+df['Municipio']+df['UF']
df['Total Rend'] = df['Rendim. Tribut.'] + df['Rendim. Tribut. Exclus.'] + df['Rendim.
 Isentos']
```

In [3]:

```
instrucao = {'Total Rend por declarante': 'Total Rend','Instrução por declarante': 'Ins
trução','Médicas por declarante': 'Médicas','Bens e Direitos por declarante': 'Bens e D
ireitos' }

# O objetivo da conversão abaixo é obter os valores de rendimentos, despesas e bens e d
ireitos por declarante.
# Essa conversão será relevante para comparar os dados com o Pib per capita.

for nome, coluna in instrucao.items():
    df[nome] = df[coluna]/df['Qtde Declarantes']*1000000

# Conversão da base original que está em múltiplo de R$ milhões
df['Total Rend'] = df['Total Rend']*1000000
df['Bens e Direitos'] = df['Bens e Direitos']*1000000
df['Instrução'] = df['Instrução']*1000000
df['Médicas'] = df['Médicas']*1000000
```

In [4]:

```
# Exportando para CSV apenas as colunas que serão utilizadas na análise
df.to_csv('dirpf2015_2018.csv', index=False,columns=['Qtde Declarantes', 'Instrução',
       'Médicas','Bens e Direitos','ano', 'Municipio', 'UF', 'chave', 'Total Rend', 'To
tal Rend por declarante',
       'Instrução por declarante', 'Médicas por declarante',
       'Bens e Direitos por declarante'])
```

In [23]:

```
# Importando base de população do site IBGE: https://sidra.ibge.gov.br/

#UF - População:
df_uf=pd.read_excel('pop.xlsx', sheet_name='UF')
print(df_uf.head())

#Municiípios - População
df_mun=pd.read_excel('pop.xlsx', sheet_name='Mun')
df_mun['UF'] = df_mun['Municipio'].apply(lambda x: x[-4:].strip('(').strip(')'))
df_mun['Municipio'] = df_mun['Municipio'].apply(lambda x: x[:-4].strip())


# Normalização dos nomes dos municípios para utilização no merge
df_mun['Municipio'] = df_mun['Municipio'].str.normalize('NFKD').str.encode('ascii', err
ors='ignore').str.decode('utf-8')
df_mun['Municipio'] = df_mun['Municipio'].apply(lambda x: x.upper())

df_mun['chave'] = df_mun['Municipio']+df_mun['UF']

# Exportando para CSV apenas as colunas que serão utilizadas na análise
df_mun.to_csv('pop_mun.csv', index=False)
df_uf.to_csv('pop_uf.csv', index=False)
```

```
   UF  Pop UF 2015  Pop UF 2016  Pop UF 2017  Pop UF 2018
0  RO    1768204.0    1787279.0    1805788.0    1757589.0
1  AC     803513.0     816687.0     829619.0     869265.0
2  AM    3938336.0    4001667.0    4063614.0    4080611.0
3  RR     505665.0     514229.0     522636.0     576568.0
4  PA    8175113.0    8272724.0    8366628.0    8513497.0
```

In [25]:

```
df_mun.head()
```

Out[25]:

	Municipio	Pop Mun 2015	Pop Mun 2016	Pop Mun 2017	Pop Mun 2018	UF	chave
0	ALTA FLORESTA D'OESTE	25578	25506	25437	23167	RO	ALTA FLORESTA D'OESTERO
1	ARIQUEMES	104401	105896	107345	106168	RO	ARIQUEMESRO
2	CABIXI	6355	6289	6224	5438	RO	CABIXIRO
3	CACOAL	87226	87877	88507	84813	RO	CACOALRO
4	CEREJEIRAS	17986	17959	17934	16444	RO	CEREJEIRASRO

In [6]:

```
pib = pd.read_excel('pib_municipios.xlsx')
pib.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22280 entries, 0 to 22279
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Ano                  22280 non-null  int64  
 1   UF                   22280 non-null  object 
 2   Código do Município  22280 non-null  int64  
 3   Municipio            22280 non-null  object 
 4   PIB x 1000           22280 non-null  float64
 5   PIB per capita x 1   22280 non-null  float64
dtypes: float64(2), int64(2), object(2)
memory usage: 1.0+ MB
```

In [7]:

```
# Importando base de população do site IBGE: https://sidra.ibge.gov.br/

#PIB per caspita por município:

pib_munic = pib[['Ano','UF', 'Código do Município', 'Municipio', 'PIB x 1000',
       'PIB per capita x 1 ']]

# Normalização dos nomes dos municípios para utilização no merge
pib_munic['Municipio'] = pib_munic['Municipio'].str.normalize('NFKD').str.encode('asci
i', errors='ignore').str.decode('utf-8')
pib_munic['Municipio'] = pib_munic['Municipio'].apply(lambda x: x.upper())

pib_munic['chave'] = pib_munic['Ano'].astype(str)+pib_munic['Municipio']+pib_munic['UF'
]

# Exportando para CSV apenas as colunas que serão utilizadas na análise
pib_munic.to_csv('pib_munic.csv', index=False)
```