SENEX

Projeto Senex

Reuniu-se o alto escalão administrativo da Senex Enterprise Inc. para avaliar as ferramentas apresentadas pelos desenvolvedores do projeto Senex Health ®.

Ferramentas escolhidas

Para atingir o objetivo de análise exploratória dos dados foram escolhidos as ferramentas PySUS, Pandas, Seaborn, Numpy e Matplotlib.

PySUS: o PySUS é um pacote que coleta dados abertos de saúde dos usuários do Sistema Único de Saúde o qual dá cobertura de diagnóstico e tratamento para seus clientes. Como os dados do SUS são gerenciados pelo governo brasileiro, estas informações já estão alinhadas com o LGPD podendo serem usadas sem restrição ética.

Pandas: Pacote responsável por realizar a montagem de dataframes e fazer e acessar estatísticas descritivas iniciais a fim de facilitar o processo de preparação dos dados.

Seaborn e Matplotlib: Pacote de visualização das distribuições estatísticas dos dados como por exemplo em histogramas e boxplots.

Numpy: Biblioteca para operar dados organizados em matrizes multi-dimensionais além de diversas ferramentas matemáticas.

Processo de aquisição e exploratória

Dev Ítalo: Construiu parte do código que requisitou os dados junto ao DataSUS e montou o banco de dados para iniciar a exploratória.

```
import sys
IN_COLAB = 'google.colab' in sys.modules

if IN_COLAB:
    !pip install pySUS==0.5.8
    !pip install geopandas
    !pip install dython
```

Figura 1 - Instalação do PySUS

```
from pysus.online_data import SIM
from pysus.preprocessing.decoders import decodifica_idade_SIM, translate_variables_SIM
import pandas as pd
import numpy as np
```

Figura 2 – Instalação dos módulos para requisitar dados de morte do DataSUS

```python
def dwdados(ano, estado, conv_cat=True, create_datetime=True, delete_old_datetime=True):
    df = SIM.download(estado, ano)

    df.IDADE = pd.to_numeric(df.IDADE)

    if (create_datetime==True):
        df['DATETIME'] = pd.to_datetime(df['DTOBITO']+' '+df['HORAOBITO'], format="%d%m%Y %H%M", errors='coerce')

    if (delete_old_datetime==True):
        df=df.drop(columns=['DTOBITO', 'HORAOBITO'])


    return df
```

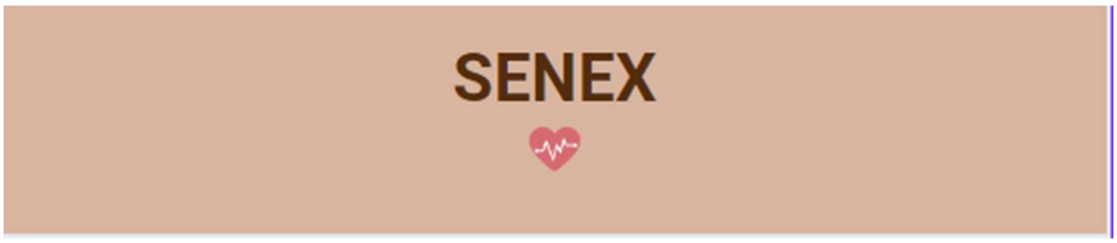
Figura 3 – Função que faz as requisições ao banco de dados

```python
estados =['SC'] #colocar os estados
anos = [2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020] #colocar os anos
banco={}
for y in anos:
 for uf in estados:
  banco[uf, y] = dwdados(y, uf, create_datetime=False, delete_old_datetime=False)

dados = pd.concat({ k: pd.DataFrame.from_dict(v)
for k, v in banco.items() }, axis=0).reset_index()

dados.to_csv('dados_2011_2020.csv')
```

Figura 4 – Requisição ao banco de dados SIM e criação do dataframe no arquivo "dados"



Dev Juana: Realizou a primeira exploratória e limpeza de dados com a ferramenta Pandas e realizou uma feature engineering transformando as colunas de data de nascimento e óbito na coluna idade.

```
df = pd.read_csv('/content/gdrive/MyDrive/SENEX/dados_2011_2020.csv')
df.head()
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:3
  exec(code_obj, self.user_global_ns, self.user_ns)
```

	Unnamed: 0	ESTADO	ANO	level_2	CONTADOR	ORIGEM	TIPOBITO	DTOBITO
0	0	SC	2010	0	1	1	2	3102010
1	1	SC	2010	1	2	1	2	15022010
2	2	SC	2010	2	3	1	2	8112010
3	3	SC	2010	3	4	1	2	4052010
4	4	SC	2010	4	5	1	2	17052010

5 rows × 100 columns

Figura 6 – Banco de dados original com 100 atributos e milhares de instâncias

```
obito = df['DTOBITO'].astype(str).str.slice()
df['DTOBITO'] = obito.str[-4:].apply(pd.to_numeric)

nasc = df['DTNASC'].astype(str).str.replace('.','0')
nasc.dropna()
df['DTNASC'] = pd.to_numeric(nasc.str[-6:-2], errors='coerce')

df['IDADE'] = df['DTOBITO'] - df['DTNASC']
```

Figura 7 – Feature engineering realizada para obter a coluna "IDADE"

```
df_senex = df[['ESTADO', 'ANO', 'IDADE', 'SEXO', 'RACACOR', 'ESTCIV', 'CAUSABAS']]
```

```
df_senex
```

	ESTADO	ANO	IDADE	SEXO	RACACOR	ESTCIV	CAUSABAS
0	SC	2010	91.0	1	1.0	2.0	C61
1	SC	2010	80.0	1	1.0	2.0	I500
2	SC	2010	79.0	2	1.0	3.0	I500
3	SC	2010	63.0	1	1.0	2.0	C710
4	SC	2010	68.0	2	1.0	2.0	C349
...	...	...	...	...	...	...	...
427268	SC	2020	79.0	2	1.0	1.0	C349
427269	SC	2020	41.0	1	1.0	2.0	I219
427270	SC	2020	79.0	1	1.0	2.0	B342
427271	SC	2020	82.0	1	1.0	2.0	Y831
427272	SC	2020	81.0	2	1.0	9.0	G10

427273 rows × 7 columns

Figura 8 – Construção do dataframe após a primeira limpeza

SENEX

Dev Josafat: Realizou uma segunda limpeza selecionando as idades do público-alvo e apresentou uma visualização dos dados.

```
sns.set(style='whitegrid', palette="deep", font_scale=1.1, rc={"figure.figsize": [8, 5]})
sns.distplot(
    df_senex['IDADE'], norm_hist=False, kde=False, bins=10, hist_kws={"alpha": 1}
).set(xlabel='IDADE', ylabel='Count');
```

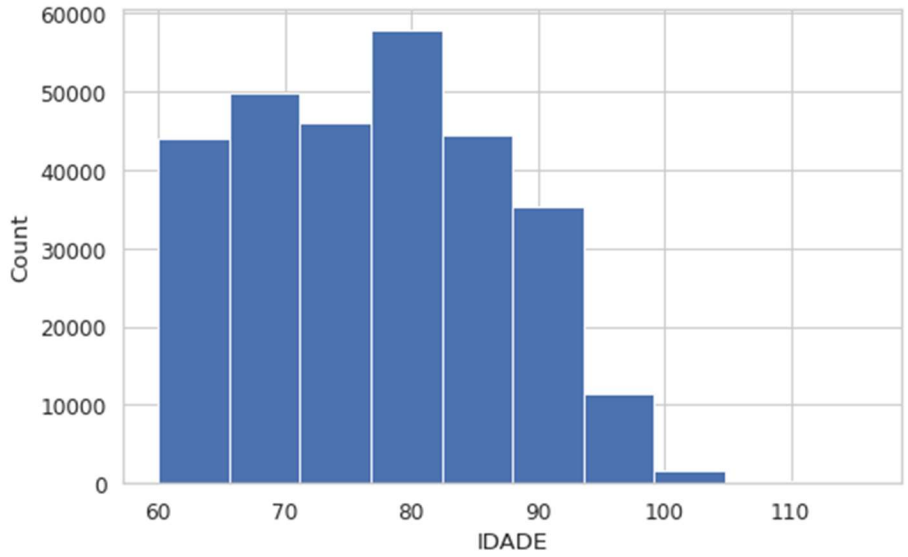


Figura 9 – Distribuição das idades (pessoas maiores de 60 anos)

```
df_senex[numerical].hist(bins=15, figsize=(15, 6), layout=(2, 4));
```

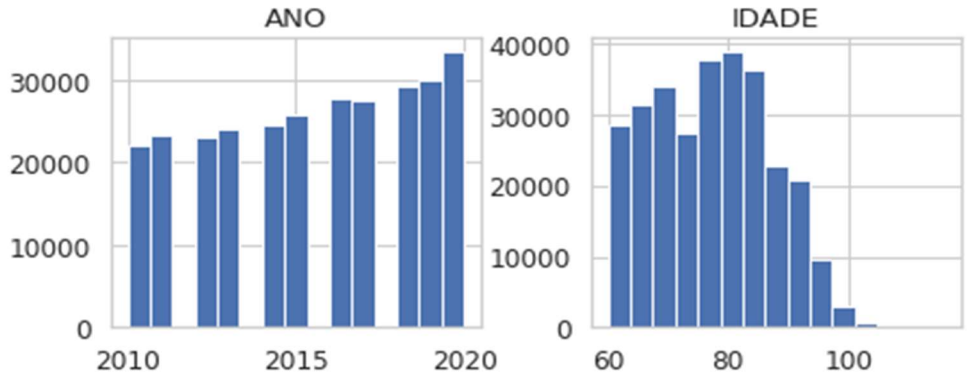


Figura 10 – histograma dos dados por ano e por idade

```
fig, ax = plt.subplots(1, 5, figsize=(25, 10))
for var, subplot in zip(categorical, ax.flatten()):
    sns.boxplot(x=var, y='IDADE', data=df_senex, ax=subplot)
```

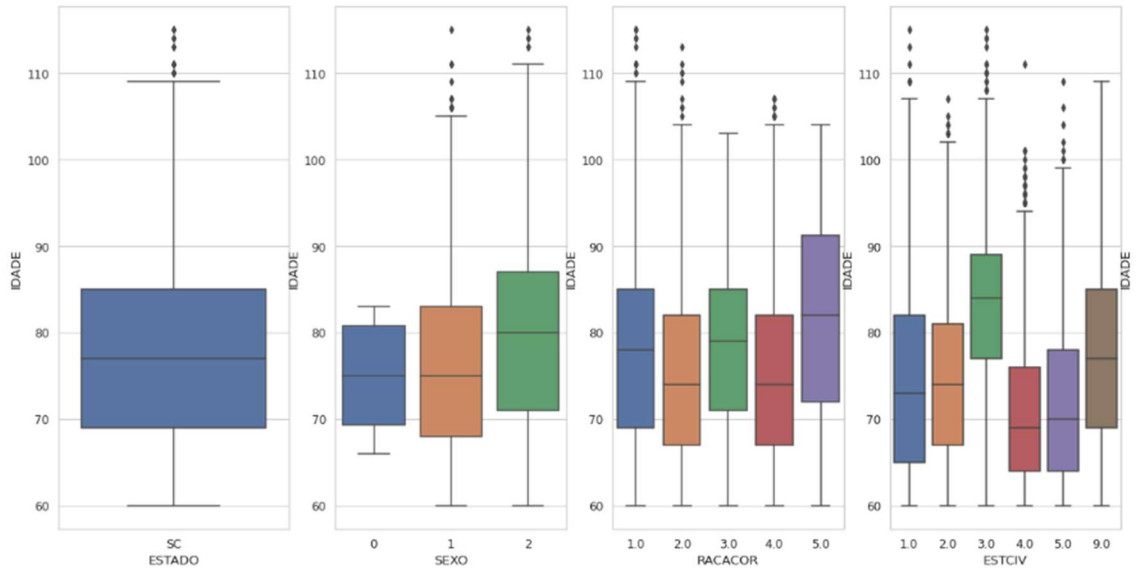


Figura 11 – Distribuição dos atributos em relação às idades.

```
fig, ax = plt.subplots(figsize=(30, 10))
sorted_nb = df_senex.groupby(['CAUSABAS'])['CAUSABAS'].count().nlargest(50)
sns.countplot(ax=ax,data=df_senex,x=df_senex['CAUSABAS'],hue="SEXO", order=list(sorted_nb.index[0:50]))
```

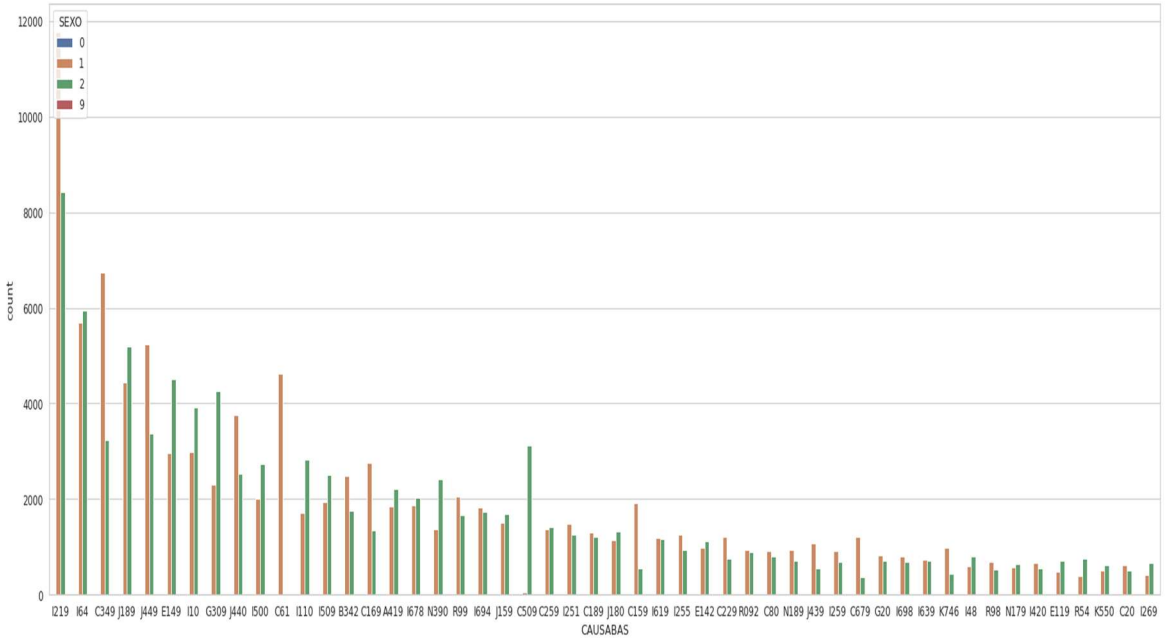


Figura 12 – Frequências de CID por sexo

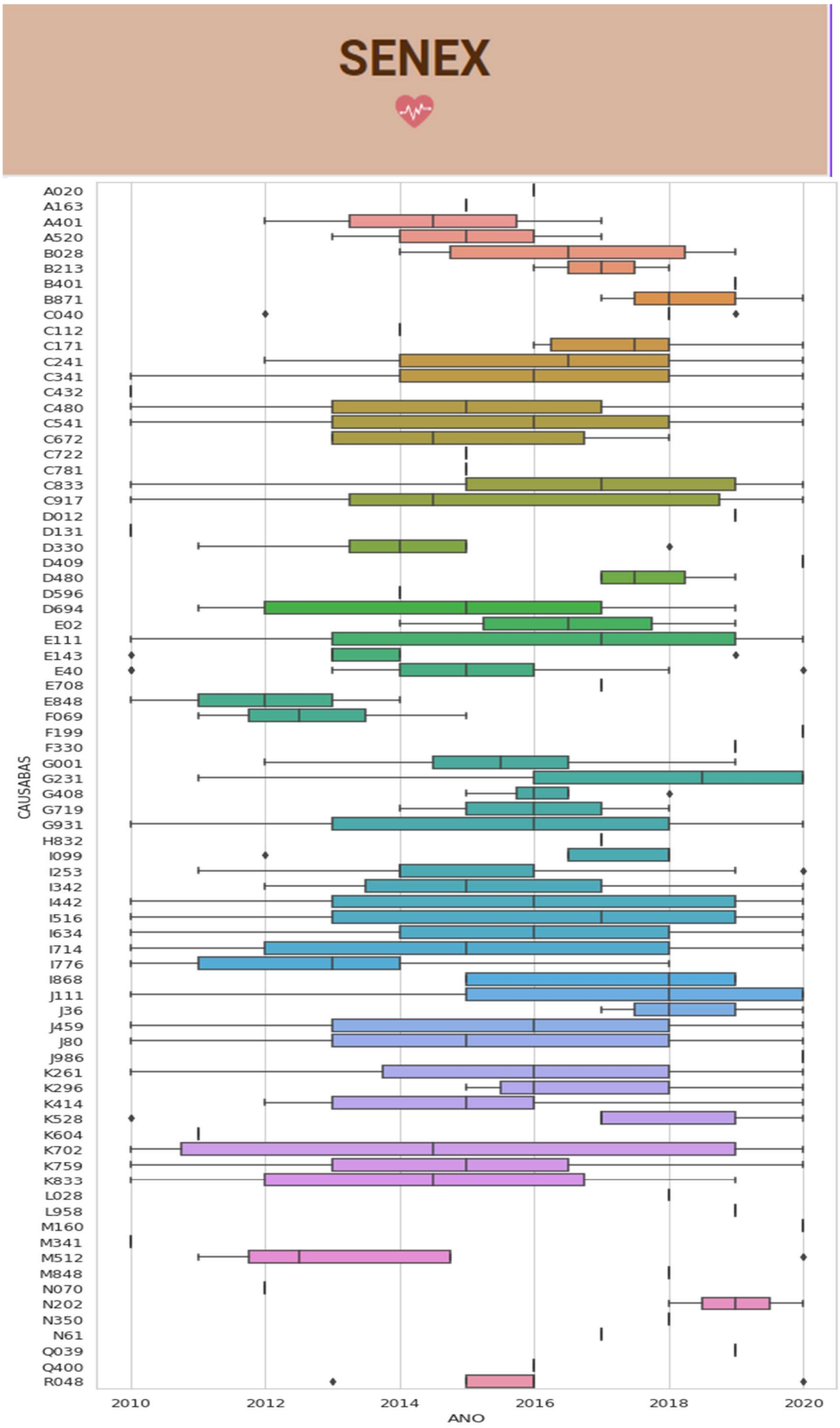


Figura 13 – Frequências de CID por ano

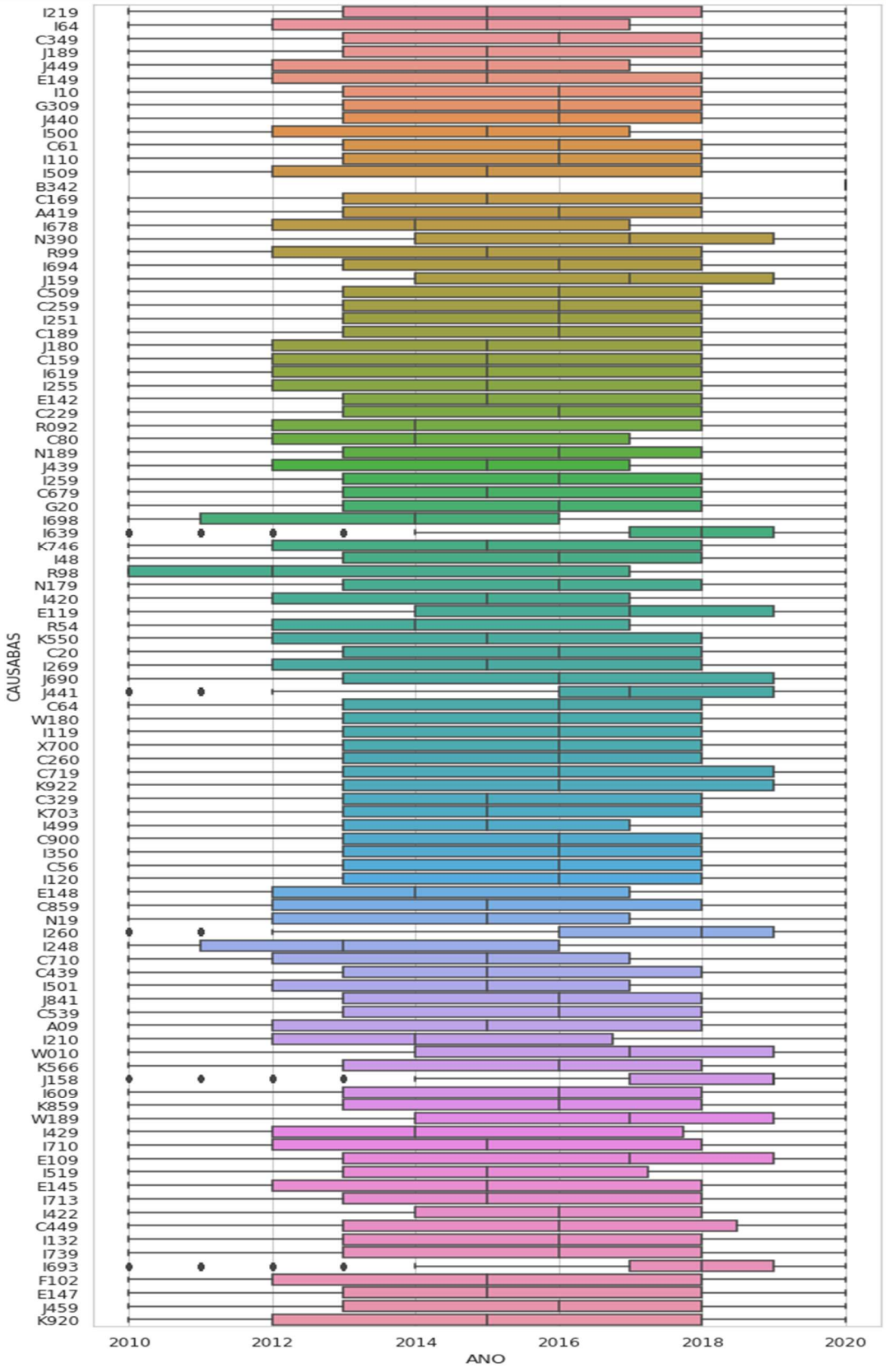


Figura 14 – CID por ano (nlargest = 100)

SENEX

```
fig, ax = plt.subplots(figsize=(25, 10))
sorted_nb = df_senex.groupby(['CAUSABAS'])['CAUSABAS'].count().nlargest(50)
sns.countplot(ax=ax,data=df_senex,x=df_senex['CAUSABAS'],hue="RACACOR", order=list(sorted_nb.index[0:40]))
```

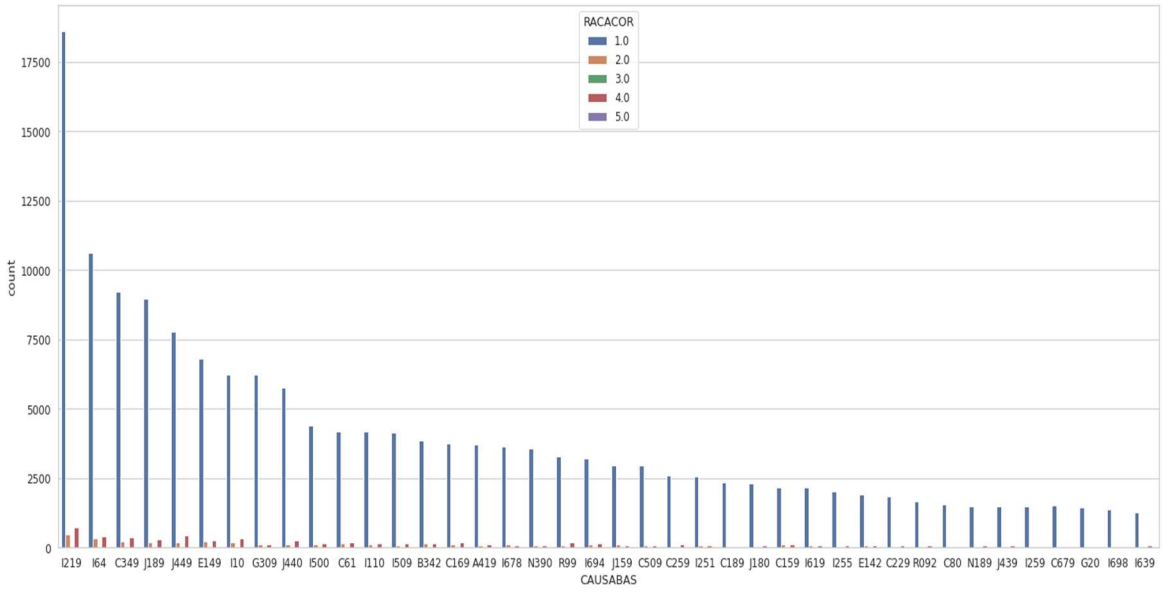


Figura 15 – Frequência de CID por raça/cor

SENEX

Insights

Com essa exploratória inicial nota-se alguma conformidades dos dados com o esperado para região, quando observamos os agrupamentos por raça/cor vemos uma predominância da cor branca em comparação as outras cores, o que era esperado já que de acordo com o censo de 2010 em SC há 83% de brancos.

Uma análise inicial dos CIDs mostraram que a doença com maior frequência em SC, independente do sexo, para a população idosa é o infarto, o que está de acordo com informações no contexto nacional. O CID que só apresentou casos em homens foi o câncer de próstata e em mulheres mama, significando que os dados utilizados são condizentes com a realidade. Por se tratar de uma quantidade muito grande de CIDs que podem estar relacionados, ainda há uma necessidade de um melhor agrupamentos dos CIDs por categorias de doenças correlacionadas, para isso será utilizados agrupamentos já utilizados na medicina.

Para complementar os dados, acredita-se que adicionar informações de aspectos sociais como renda econômica, região da moradia pode trazer informações cruciais para o treinamento já que existem doenças que estão mais relacionadas a qualidade de vida.