Senac

Fecomércio Sesc

Data Science – Princípios e Técnicas

Setembro

2024

Data Science

Onde me encontrar:

https://www.linkedin.com/in/marco-mialaret-junior/

e

https://github.com/MatmJr

Data Wrangling II

Data Science

Lembrando que processo de preparar dados para análise é chamado de "data wrangling" e geralmente ocupa a maior parte do tempo de um analista em um projeto de dados. Alguns dos problemas comuns incluem:

Data Science

- Dados ausentes ou valores incorretos/problemas no conjunto de dados;
- Dados formatados de maneira inadequada, impedindo o trabalho adequado do analista;
- Dados distribuídos em vários arquivos ou tabelas;
- Dados no formato incorreto para análise e visualização.

E o Pysus

Data Science

Alguns grupos tiveram problemas com os dados do pysus, na aula de hoje vamos tentar padronizar uma forma de acessar os dados e criar maneiras de começar um estudo de análise de dados.

Data Science

Instalando o pysus:

Abra um notebook no colab. Na primeira célula, execute:

```
!pip install pysus
```

Data Science

A base de dados SIH:

```
from pysus.ftp.databases.sih import SIH
import pandas as pd

sih = SIH().load()
sih.metadata['description']
```

Data Science

A finalidade do AIH (Sistema SIHSUS) é a de transcrever todos os atendimentos que provenientes de internações hospitalares que foram financiadas pelo SUS, e após o processamento, gerarem relatórios para os gestores que lhes possibilitem fazer os pagamentos dos estabelecimentos de saúde. Além disso, o nível Federal recebe mensalmente uma base de dados de todas as internações autorizadas (aprovadas ou não para pagamento) para que possam ser repassados às Secretarias de Saúde os valores de Produção de Média e Alta complexidade além dos valores de CNRAC, FAEC e de Hospitais Universitários – em suas variadas formas de contrato de gestão.

Data Science

sih.groups

```
{'RD': 'AIH Reduzida',
 'RJ': 'AIH Rejeitada',
 'ER': 'AIH Rejeitada com erro',
 'SP': 'Serviços Profissionais',
 'CH': 'Cadastro Hospitalar',
 'CM': ''}
```

Data Science

```
files = sih.get_files("RD", uf="PE", year=2024, month=[1])
data = sih.download(files)
#usar sih.download(files)[0] quando estiver usando vários meses
df = data.to_dataframe()
```

Data Science

O comando df.columns, gera o resultado:

```
Index(['UF_ZI', 'ANO_CMPT', 'MES_CMPT', 'ESPEC', 'CGC_HOSP', 'N_AIH', 'IDENT',
       'CEP', 'MUNIC_RES', 'NASC',
       ...
       'DIAGSEC9', 'TPDISEC1', 'TPDISEC2', 'TPDISEC3', 'TPDISEC4', 'TPDISEC5',
       'TPDISEC6', 'TPDISEC7', 'TPDISEC8', 'TPDISEC9'],
      dtype='object', length=113)
```

Muitos atributos foram omitidos.

Data Science

O comando df.columns, gera o resultado:

```
Index(['UF_ZI', 'ANO_CMPT', 'MES_CMPT', 'ESPEC', 'CGC_HOSP', 'N_AIH', 'IDENT',
       'CEP', 'MUNIC_RES', 'NASC',
       ...
       'DIAGSEC9', 'TPDISEC1', 'TPDISEC2', 'TPDISEC3', 'TPDISEC4', 'TPDISEC5',
       'TPDISEC6', 'TPDISEC7', 'TPDISEC8', 'TPDISEC9'],
      dtype='object', length=113)
```

Muitos atributos foram omitidos.

Data Science

Uma saída é usar : df.columns.tolist()

```
['UF_ZI',
 'ANO_CMPT',
 'MES_CMPT',
 'ESPEC',
 'CGC_HOSP',
 'N_AIH',
 'IDENT',
 'CEP',
 'MUNIC_RES',
 'NASC',
 'SEXO',
 'UTI_MES_IN',
 'UTI_MES_AN',
 'UTI_MES_AL',
 'UTI_MES_TO',
 'MARCA_UTI',
 'UTI_INT_IN',
 'UTI_INT_AN',
 'UTI_INT_AL',
 'UTI_INT_TO',
 'DIAR_ACOM',
 'QT_DIARIAS',
```

Data Science

Assim podemos explorar os atributos que existem no banco de dados e começar a elaborar perguntas para a nossa análise. Exemplo:

Qual a distribuição das idades dos pacientes?

Data Science

df['IDADE']

	IDADE
0	9
1	5
2	13
3	3
4	1
...	...
50168	35
50169	42
50170	36
50171	32
50172	62

50173 rows × 1 columns

dtype: string

```
df['IDADE'][0]
```

```
' 9'
```

dtype: string

Data Science

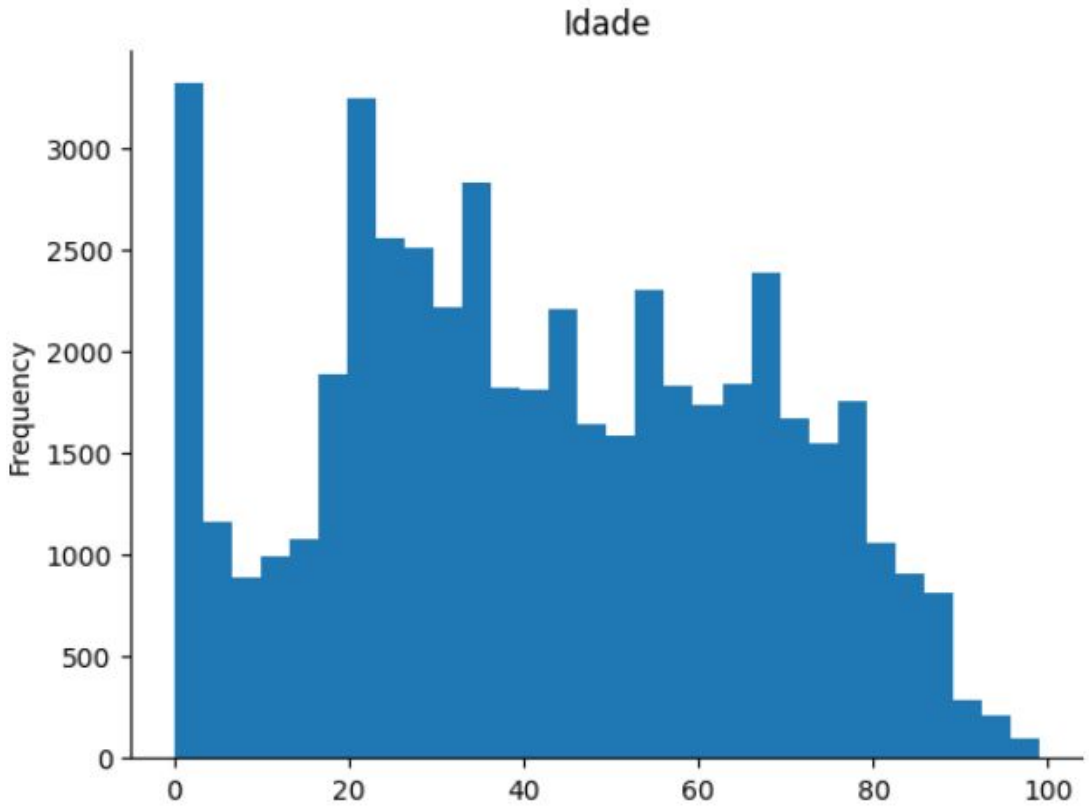
```
df['IDADE'] = df['IDADE'].str.strip()
df['IDADE'] = pd.to_numeric(df['IDADE'], errors='coerce').fillna(0).astype(float)
```

Data Science

```
from matplotlib import pyplot as plt

df['IDADE'].plot(kind='hist', bins=30, title='Idade')

plt.gca().spines[['top', 'right',]].set_visible(False)
```



Data Science

Vamos dar uma olhada nas datas de nascimento df['NASC']

	NASC
0	20140629
1	20180505
2	20101029
3	20200305
4	20220530
...	...
50168	19880122
50169	19811024
50170	19870204
50171	19910611
50172	19611122

50173 rows × 1 columns

dtype: string

dtype: string

Data Science

df['NASC'] = pd.to_datetime(df['NASC'], format='%Y%m%d', errors='coerce')

```
df['NASC']
```

	NASC
0	2014-06-29
1	2018-05-05
2	2010-10-29
3	2020-03-05
4	2022-05-30

Data Science

Depois podemos começar a fazer novas perguntas:

- Qual a raça dos paciente?
- Qual a idade das pessoas que faleceram?
- Qual o municípios dos pacientes?
- ...

Data Science

Podemos fazer o mesmo com as outras bases:

```
from pysus.ftp.databases.sia import SIA
import pandas as pd

sia = SIA().load()
sia.metadata['description']
```

Data Science

O Sistema de Informação Ambulatorial (SIA) foi instituído pela Portaria GM/MS n.º 896 de 29 de junho de 1990. Originalmente, o SIA foi concebido a partir do projeto SICAPS (Sistema de Informação e Controle Ambulatorial da Previdência Social), em que os conceitos, os objetivos e as diretrizes criados para o desenvolvimento do SICAPS foram extremamente importantes e amplamente utilizados para o desenvolvimento do SIA, tais como: (i) o acompanhamento das programações físicas e orçamentárias; (ii) o acompanhamento das ações de saúde produzidas; (iii) a agilização do pagamento e controle orçamentário e financeiro; e (iv) a formação de banco de dados para contribuir com a construção do SUS.

Data Science

sia.groups

```
{'AB': 'APAC de Cirurgia Bariátrica',
 'ABO': 'APAC de Acompanhamento Pós Cirurgia Bariátrica',
 'ACF': 'APAC de Confecção de Fístula',
 'AD': 'APAC de Laudos Diversos',
 'AM': 'APAC de Medicamentos',
 'AMP': 'APAC de Acompanhamento Multiprofissional',
 'AN': 'APAC de Nefrologia',
 'AQ': 'APAC de Quimioterapia',
 'AR': 'APAC de Radioterapia',
 'ATD': 'APAC de Tratamento Dialítico',
 'BI': 'Boletim de Produção Ambulatorial individualizado',
 'IMPBO': '',
 'PA': 'Produção Ambulatorial',
 'PAM': '',
 'PAR': '',
 'PAS': '',
 'PS': 'RAAS Psicossocial',
 'SAD': 'RAAS de Atenção Domiciliar'}
```

Data Science

```
files2 = sia.get_files("AM", uf="PE", year=2023, month=[11])
data2 = sia.download(files2)
#usar sih.download(files)[0] quando estiver usando vários meses
df2 = data.to_dataframe()
```

Data Science

```
df2['AP_NUIDADE']
```

	AP_NUIDADE
0	48
1	79
2	62
3	46
4	43
...	...
64055	68
64056	72
64057	09
64058	09
64059	06

64060 rows × 1 columns

dtype: string

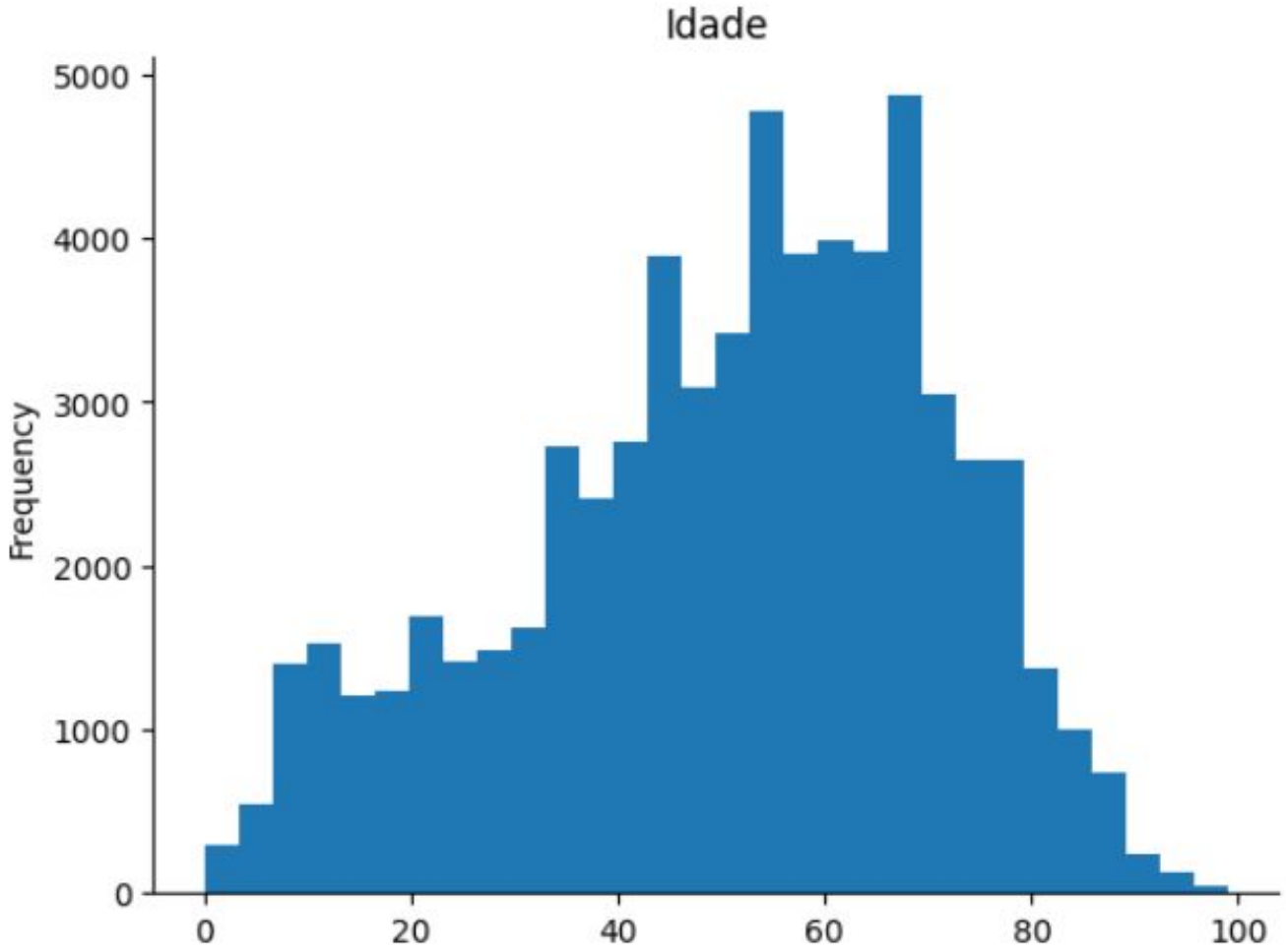
Data Science

```
df2['AP_NUIDADE'] = df2['AP_NUIDADE'].str.strip()
df2['AP_NUIDADE'] = pd.to_numeric(df2['AP_NUIDADE'], errors='coerce').fillna(0).astype(float)

from matplotlib import pyplot as plt

df2['AP_NUIDADE'].plot(kind='hist', bins=30, title='Idade')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

Data Science



Data Science

A nova estratégia de criar novas perguntas pode ser usada

- Qual a raça dos paciente?
- Qual a idade das pessoas que faleceram?
- Qual o municípios dos pacientes?
- ...

Dúvidas?

Marco Mialaret, MSc

Telefone:
81 98160 7018

E-mail:
marcomialaret@gmail.com