Módulo 5 Tarefa 1

Base de nascidos vivos do DataSUS

O DataSUS disponibiliza diversos arquivos de dados com relação a seus segurados, conforme a lei da transparência de informações públicas.

Essas informações podem ser obtidas pela internet aqui. Como o processo de obtenção desses arquivos foge um pouco do nosso escopo, deixamos o arquivo `SINASC_RO_2019.csv` já como vai ser encontrado no DataSUS. O dicionário de dados está no arquivo `estrutura_sinasc_para_CD.pdf` (o nome do arquivo tal qual no portal do DataSUS).

Nosso objetivo

Queremos deixar uma base organizada para podermos estudar a relação entre partos com risco para o bebê e algumas condições como tempo de parto, consultas de pré-natal etc.

Preparação da base

1. Carregue a base 'SINASC_RO_2019.csv'. Conte o número de registros e o número de registros não duplicados da base. Dica: você aprendeu um método que remove duplicados, encadeie este método com um outro método que conta o número de linhas. **Há linhas duplicadas?**

2. Conte o número de valores *missing* por variável.

3. Ok, no item anterior você deve ter achado pouco prático ler a informação de tantas variáveis, muitas delas nem devem ser interesantes. Então crie uma seleção dessa base somente com as colunas que interessam. São elas:

   ```
   ['LOCNASC', 'IDADEMAE', 'ESTCIVMAE', 'ESCMAE', 'QTDFILVIVO',
    'GESTACAO', 'GRAVIDEZ', 'CONSULTAS', 'APGAR5']
   ```
   Refaça a contagem de valores *missings*.

4. Apgar é uma *nota* que o pediatra dá ao bebê quando nasce de acordo com algumas características associadas principalmente à respiração. Apgar 1 e Apgar 5 são as notas 1 e 5 minutos do nascimento. Apgar5 será a nossa variável de interesse principal. Então remova todos os registros com Apgar5 não preenchido. Para esta seleção, conte novamente o número de linhas e o número de *missings*.

5. observe que as variáveis `['ESTCIVMAE', 'CONSULTAS']` possuem o código `9`, que significa *ignorado*. Vamos assumir que o não preenchido é o mesmo que o código `9`.

6. Substitua os valores faltantes da quantitativa (`QTDFILVIVO`) por zero.
7. Das restantes, decida que valore te parece mais adequado (um 'não preenchido' ou um valor 'mais provável' como no item anterior) e preencha. Justifique. Lembre-se de que tratamento de dados é trabalho do cientista, e que estamos tomando decisões a todo o momento - não há necessariamente certo e errado aqui.
8. O Apgar possui uma classificação indicando se o bebê passou por asfixia:

- Entre 8 e 10 está em uma faixa 'normal'.

- Entre 6 e 7, significa que o recém-nascido passou por 'asfixia leve'.
- Entre 4 e 5 significa 'asfixia moderada'.
- Entre 0 e 3 significa 'asfixia severa'.

Crie uma categorização dessa variável com essa codificação e calcule as frequências dessa categorização.

1. Renomeie as variáveis para que fiquem no *snake case*, ou seja, em letras minúsculas, com um *underscore* entre as palávras. Dica: repare que se você não quiser criar um *dataframe* novo, você vai precisar usar a opção `inplace = True`.

In [1]:
```
import pandas as pd
import requests

# 1) Conte o número de registros e o número de registros não duplicados da base.
sinasc = pd.read_csv('SINASC_RO_2019.csv')
print(sinasc.shape)
sinasc.drop_duplicates().shape
# Não há duplicados
```

```
(27028, 69)
```

Out[1]:
```
(27028, 69)
```

In [2]:
```
# 2) Conte o número de valores missing por variável.
print(f"{sinasc.shape[1]} Colunas\n")
for i in sinasc.columns:
    print(f"{i}: {sinasc[i].isna().sum()}")
```

```
69 Colunas

ORIGEM: 0
CODESTAB: 115
CODMUNNASC: 0
LOCNASC: 0
IDADEMAE: 0
ESTCIVMAE: 317
ESCMAE: 312
CODOCUPMAE: 2907
QTDFILVIVO: 1573
QTDFILMORT: 2098
CODMUNRES: 0
GESTACAO: 1232
GRAVIDEZ: 79
PARTO: 49
CONSULTAS: 0
DTNASC: 0
HORANASC: 21
SEXO: 4
APGAR1: 96
APGAR5: 103
RACACOR: 647
PESO: 0
IDANOMAL: 591
DTCADASTRO: 0
CODANOMAL: 26814
NUMEROLOTE: 0
VERSAOSIST: 0
DTRECEBIM: 0
DIFDATA: 0
DTRECORIGA: 27028
NATURALMAE: 298
CODMUNNATU: 298
```

```
CODUFNATU: 298
ESCMAE2010: 249
SERIESCMAE: 12710
DTNASCMAE: 40
RACACORMAE: 661
QTDGESTANT: 1212
QTDPARTNOR: 1879
QTDPARTCES: 1791
IDADEPAI: 19421
DTULTMENST: 10072
SEMAGESTAC: 1238
TPMETESTIM: 1238
CONSPRENAT: 930
MESPRENAT: 2867
TPAPRESENT: 265
STTRABPART: 947
STCESPARTO: 747
TPNASCASSI: 61
TPFUNCRESP: 67
TPDOCRESP: 14
DTDECLARAC: 52
ESCMAEAGR1: 249
STDNEPIDEM: 0
STDNNOVA: 0
CODPAISRES: 0
TPROBSON: 0
PARIDADE: 0
KOTELCHUCK: 0
CONTADOR: 0
munResStatus: 0
munResTipo: 0
munResNome: 0
munResUf: 0
munResLat: 1
munResLon: 1
munResAlt: 1
munResArea: 1
```

In [3]:
```
# 3) Então crie uma seleção dessa base somente com as colunas que interessam.
Colunas_int = ['LOCNASC', 'IDADEMAE', 'ESTCIVMAE', 'ESCMAE', 'QTDFILVIVO', 'GESTACAO', '
for i in Colunas_int:
    print(f"{i}: {sinasc[i].isna().sum()}")
```

```
LOCNASC: 0
IDADEMAE: 0
ESTCIVMAE: 317
ESCMAE: 312
QTDFILVIVO: 1573
GESTACAO: 1232
GRAVIDEZ: 79
CONSULTAS: 0
APGAR5: 103
```

In [4]:
```
# 4) Então remova todos os registros com Apgar5 não preenchido.
sinasc_APGAR5_dropped = sinasc[sinasc['APGAR5'].isna()]
```

In [5]:
```
# 4) Então remova todos os registros com Apgar5 não preenchido.
sinasc.dropna(subset=['APGAR5'], inplace=True)
print(f"Após a remoção, o número de linhas é: {sinasc.shape[0]}\n")
print(f"Foram removidas: {sinasc_APGAR5_dropped[Colunas_int].shape[0]} linhas")
```

```
Após a remoção, o número de linhas é: 26925

Foram removidas: 103 linhas
```

In [6]:
```
# 5) observe que as variáveis ['ESTCIVMAE', 'CONSULTAS']. Vamos assumir que o não preenc
```

```
sinasc[['ESTCIVMAE', 'CONSULTAS']].isna().sum()
```

Out[6]:
```
ESTCIVMAE    315
CONSULTAS      0
dtype: int64
```

In [7]:
```
# 5) observe que as variáveis ['ESTCIVMAE', 'CONSULTAS']. Vamos assumir que o não preenc
sinasc['ESTCIVMAE'].fillna(value=9, inplace=True)
sinasc[['ESTCIVMAE', 'CONSULTAS']].isna().sum()
```

Out[7]:
```
ESTCIVMAE    0
CONSULTAS    0
dtype: int64
```

In [8]:
```
# 6) Substitua os valores faltantes da quantitativa (QTDFILVIVO) por zero.
sinasc['QTDFILVIVO'].isna().sum()
```

Out[8]:
```
1566
```

In [9]:
```
# 6) Substitua os valores faltantes da quantitativa (QTDFILVIVO) por zero.
sinasc['QTDFILVIVO'].fillna(value=0, inplace=True)
sinasc['QTDFILVIVO'].isna().sum()
```

Out[9]:
```
0
```

In [10]:
```
# 7) decida que valor te parece mais adequado (um 'não preenchido' ou um valor 'mais pro
# Os campos vazios de ['ESCMAE', 'GESTACAO', 'GRAVIDEZ'] foram preenchidos com o número
sinasc['ESCMAE'].fillna(value=9, inplace=True)
sinasc['GESTACAO'].fillna(value=9, inplace=True)
sinasc['GRAVIDEZ'].fillna(value=9, inplace=True)

for i in Colunas_int:
    print(f"{i}: {sinasc[i].isna().sum()}")
```

```
LOCNASC: 0
IDADEMAE: 0
ESTCIVMAE: 0
ESCMAE: 0
QTDFILVIVO: 0
GESTACAO: 0
GRAVIDEZ: 0
CONSULTAS: 0
APGAR5: 0
```

In [11]:
```
# 8) Entre 8 e 10 está em uma faixa 'normal'.
#    Entre 6 e 7, significa que o recém-nascido passou por 'asfixia leve'.
#    Entre 4 e 5 significa 'asfixia moderada'.
#    Entre 0 e 3 significa 'asfixia severa'.

sinasc.loc[sinasc['APGAR1'] >= 8, ['APGAR1_cat']] = 'normal'
sinasc.loc[(sinasc['APGAR1'] < 8) & (sinasc['APGAR1'] >= 6), ['APGAR1_cat']] = 'asfixia
sinasc.loc[(sinasc['APGAR1'] < 6) & (sinasc['APGAR1'] >= 4), ['APGAR1_cat']] = 'asfixia
sinasc.loc[sinasc['APGAR1'] <= 3, ['APGAR1_cat']] = 'asfixia severa'

sinasc.loc[sinasc['APGAR5'] >= 8, ['APGAR5_cat']] = 'normal'
sinasc.loc[(sinasc['APGAR5'] < 8) & (sinasc['APGAR5'] >= 6), ['APGAR5_cat']] = 'asfixia
sinasc.loc[(sinasc['APGAR5'] < 6) & (sinasc['APGAR5'] >= 4), ['APGAR5_cat']] = 'asfixia
sinasc.loc[sinasc['APGAR5'] <= 3, ['APGAR5_cat']] = 'asfixia severa'

print(f"1° minuto:\n{sinasc['APGAR1_cat'].value_counts()}\n")
print(f"5° minuto:\n{sinasc['APGAR5_cat'].value_counts()}")
```

```
1° minuto:
normal             23793
asfixia leve        2522
asfixia moderada     376
```

```
asfixia severa        230
Name: APGAR1_cat, dtype: int64

5° minuto:
normal              26463
asfixia leve          320
asfixia severa         74
asfixia moderada       68
Name: APGAR5_cat, dtype: int64
```

In [12]:
```
# 9) Renomeie as variáveis para que fiquem no snake case, ou seja, em letras minúsculas
sinasc.columns = sinasc.columns.str.lower()
sinasc.columns
```

Out[12]:
```
Index(['origem', 'codestab', 'codmunnasc', 'locnasc', 'idademae', 'estcivmae',
       'escmae', 'codocupmae', 'qtdfilvivo', 'qtdfilmort', 'codmunres',
       'gestacao', 'gravidez', 'parto', 'consultas', 'dtnasc', 'horanasc',
       'sexo', 'apgar1', 'apgar5', 'racacor', 'peso', 'idanomal', 'dtcadastro',
       'codanomal', 'numerolote', 'versaosist', 'dtrecebim', 'difdata',
       'dtrecoriga', 'naturalmae', 'codmunnatu', 'codufnatu', 'escmae2010',
       'seriescmae', 'dtnascmae', 'racacormae', 'qtdgestant', 'qtdpartnor',
       'qtdpartces', 'idadepai', 'dtultmenst', 'semagestac', 'tpmetestim',
       'consprenat', 'mesprenat', 'tpapresent', 'sttrabpart', 'stcesparto',
       'tpnascassi', 'tpfuncresp', 'tpdocresp', 'dtdeclarac', 'escmaeagr1',
       'stdnepidem', 'stdnnova', 'codpaisres', 'tprobson', 'paridade',
       'kotelchuck', 'contador', 'munresstatus', 'munrestipo', 'munresnome',
       'munresuf', 'munreslat', 'munreslon', 'munresalt', 'munresarea',
       'apgar1_cat', 'apgar5_cat'],
      dtype='object')
```