

Terceira_Entrega

October 10, 2021

1 Trabalho Prático: Introdução à Ciência dos Dados

1.0.1 Entrega 3 - Análise exploratória e extração de conhecimento

Integrantes: Aryel Penido - 3500 Claudio Barbosa - 3492 Isabela Ramos - 3474

Tema: Análise da população em situação de rua em BH

Dados: <https://dados.pbh.gov.br/dataset/populacao-de-rua>

1.1 Perguntas a serem respondidas:

1. Qual a distribuição geral de idade dos moradores de rua em Belo Horizonte?
2. Qual a relação entre o contato com parentes e o tempo que estão em situação de rua?
3. Qual a distribuição de idade de homens moradores de rua em Belo Horizonte?
4. Qual a distribuição de idade de mulheres moradoras de rua em Belo Horizonte?
5. Qual o percentual, entre os moradores de rua, cadastrados em um Cras?
6. Análise temporal da cobertura do CRAS desde o início da pesquisa?(09/2020 a 07/2021)**
7. Quais regiões os critérios para que uma implantação de unidade do CRAS seja necessária?
8. Qual o percentual de moradores de rua em cada regional de Belo Horizonte?
9. Qual o percentual de moradores de rua que possuem contato com parentes fora das ruas?
10. É possível vincular áreas com mais pessoas em situação de rua com uma maior criminalidade?
11. Houve crescimento da população em situação de rua durante o período da pandemia?
12. Quais os impactos da pandemia na remuneração dessa população.
13. Qual a cobertura do Bolsa Família nas pessoas em situação de rua?
14. Qual a relação das associações entre idade, escolaridade, valor de remuneração do trabalho principal, o sexo e idade da população em situação de rua?**
15. Quais as características que podem definir desta população em um processo de classificação?
16. Aproximadamente quantas moradias seriam necessárias para abrigar essa população?* (apresentação)
17. Quais políticas públicas devem ser aprimoradas para atender essa população?* (apresentação)
18. Qual a média de tempo em que os moradores vivem na rua?
19. Quais são a raça e o sexo predominante de cada regional presente no banco de dados?
20. Qual é o grau de instrução da população de rua?

*Alguns dos tópicos serão abordados na apresentação.

**Perguntas que ainda demandam pesquisa e análise.

2 Importantando as bibliotecas:

```
[247]: import pandas as pd
from datetime import datetime, date
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

2.1 Coleta dos dados

```
[248]: #URLs dos arquivos (direto da base de dados)
url_9_20 = 'https://ckan.pbh.gov.br/dataset/
↳1f868d62-d7c3-4121-8ffa-9ed66d71d3e2/resource/
↳aa8477c0-81fc-442c-b1bf-25475674adda/download/
↳4-data_set_poprua_cadunico_09-2020.csv'
url_10_20 = 'https://ckan.pbh.gov.br/dataset/
↳1f868d62-d7c3-4121-8ffa-9ed66d71d3e2/resource/
↳d498812a-0d81-46de-93d1-9c7cd84b85b8/download/
↳data_set_poprua_cadunico-10-2020.csv'
url_11_20 = 'https://ckan.pbh.gov.br/dataset/
↳1f868d62-d7c3-4121-8ffa-9ed66d71d3e2/resource/
↳952c9ff9-a7e6-4800-9ee7-fe7231d92a2c/download/
↳data_set_poprua_cadunico-11-2020.csv'
url_12_20 = 'https://ckan.pbh.gov.br/dataset/
↳1f868d62-d7c3-4121-8ffa-9ed66d71d3e2/resource/
↳153bf11e-5416-433b-bca4-790baa1e8f44/download/data_set_poprua_cadunico.csv'
url_1_21 = 'https://ckan.pbh.gov.br/dataset/
↳1f868d62-d7c3-4121-8ffa-9ed66d71d3e2/resource/
↳076df2cf-7f98-462d-a544-a10bc675974b/download/data_set_poprua_cadunico.csv'
url_2_21 = 'https://ckan.pbh.gov.br/dataset/
↳1f868d62-d7c3-4121-8ffa-9ed66d71d3e2/resource/
↳dd2a60bb-3600-4248-815e-69863417f91f/download/data_set_poprua_cadunico.csv'
url_3_21 = 'https://ckan.pbh.gov.br/dataset/
↳1f868d62-d7c3-4121-8ffa-9ed66d71d3e2/resource/
↳319191cc-20a5-4e7b-8b13-59cd433902dc/download/data_set_poprua_cadunico.csv'
url_4_21 = 'https://ckan.pbh.gov.br/dataset/
↳1f868d62-d7c3-4121-8ffa-9ed66d71d3e2/resource/
↳360b17f7-5262-4a4e-ad3d-6097b756871b/download/data_set_poprua_cadunico.csv'
url_5_21 = 'https://ckan.pbh.gov.br/dataset/
↳1f868d62-d7c3-4121-8ffa-9ed66d71d3e2/resource/
↳9096f15c-d400-481f-a078-5238c83ca0e7/download/data_set_poprua_cadunico.csv'
url_6_21 = 'https://ckan.pbh.gov.br/dataset/
↳1f868d62-d7c3-4121-8ffa-9ed66d71d3e2/resource/
↳44baba43-85ba-4838-8c6e-6adcb469e03b/download/
↳data_set_poprua_cadunico_06-2021.csv'
```

```

url_7_21 = 'https://ckan.pbh.gov.br/dataset/
↳1f868d62-d7c3-4121-8ffa-9ed66d71d3e2/resource/
↳f904a3a6-1f6e-42b6-b326-7ab0bfe8eca6/download/
↳data_set_poprua_cadunico_07-2021.csv'

#Leitura individual de cada um
data9_20 = pd.read_csv(url_9_20,sep = ";")
data10_20 = pd.read_csv(url_10_20,sep = ";")
data11_20 = pd.read_csv(url_11_20,sep = ";")
data12_20 = pd.read_csv(url_12_20,sep = ";")
data1_21 = pd.read_csv(url_1_21,sep = ";")
data2_21 = pd.read_csv(url_2_21,sep = ";")
data3_21 = pd.read_csv(url_3_21,sep = ";")
data4_21 = pd.read_csv(url_4_21,sep = ";")
data5_21 = pd.read_csv(url_5_21,sep = ";")
data6_21 = pd.read_csv(url_6_21,sep = ";")
data7_21 = pd.read_csv(url_7_21,sep = ";")

```

3 Agrupando todos os dados em um único Dataframe

```

[249]: df=pd.concat([data9_20,data10_20, data11_20, data12_20, data1_21, data2_21,
↳data3_21, data4_21, data5_21, data6_21, data7_21],ignore_index=True)

```

4 Agrupando os dados por ano

```

[250]: df2020=pd.concat([data9_20,data10_20, data11_20, data12_20],ignore_index=True)
df2021=pd.concat([data1_21, data2_21, data3_21, data4_21, data5_21, data6_21,
↳data7_21],ignore_index=True)

```

5 Tratamento dos dados

Verificar campos vazios/nulos:

```

[251]: df.isnull().sum()

```

```

[251]: TEMPO_VIVE_NA_RUA          0
CONTATO_PARENTE_FORA_RUAS       0
DATA_NASCIMENTO                 0
IDADE                           8282
SEXO                             0
BOLSA_FAMILIA                   0
POP_RUA                          0
GRAU_INSTRUCAO                  0
COR_RACA                         0

```

```

Faixa da renda familiar per capita      0
VAL_REMUNERACAO_MES_PASSADO            495
CRAS                                    0
REGIONAL                               0
FAIXA_DESATUALICACAO_CADASTRAL         0
MES_ANO_REFERENCIA                     0
dtype: int64

```

- É possível observar que apesar de todos os campos de data de nascimento estarem devidamente preenchidos, alguns dos campos de idade não estão.

Verificar os tipos dos dados:

```
[252]: df.dtypes
```

```

[252]: TEMPO_VIVE_NA_RUA      object
CONTATO_PARENTE_FORA_RUAS    object
DATA_NASCIMENTO              object
IDADE                        float64
SEXO                         object
BOLSA_FAMILIA                object
POP_RUA                      object
GRAU_INSTRUCAO               object
COR_RACA                     object
Faixa da renda familiar per capita  object
VAL_REMUNERACAO_MES_PASSADO    float64
CRAS                         object
REGIONAL                     object
FAIXA_DESATUALICACAO_CADASTRAL  object
MES_ANO_REFERENCIA            object
dtype: object

```

- Alguns detalhes importantes: A data_nascimento não está no formato date e os campos idade e val_remuneracao_mes_passado estão no formato float.
- Um dicionário com os tipos de cada atributo é fornecido juntamente ao banco de dados, contudo ele mostrou-se não confiável.

5.1 Utilizando a data de nascimento para cálculo da idade:

- Convertendo MES_ANO_REFERENCIA de object para datetime

```
[253]: df['MES_ANO_REFERENCIA'] = pd.to_datetime(df['MES_ANO_REFERENCIA'])
df.dtypes
```

```

[253]: TEMPO_VIVE_NA_RUA      object
CONTATO_PARENTE_FORA_RUAS    object
DATA_NASCIMENTO              object
IDADE                        float64

```

```

SEXO                                object
BOLSA_FAMILIA                      object
POP_RUA                            object
GRAU_INSTRUCAO                    object
COR_RACA                           object
Faixa da renda familiar per capita  object
VAL_REMUNERACAO_MES_PASSADO        float64
CRAS                               object
REGIONAL                           object
FAIXA_DESATUALICACAO_CADASTRAL     object
MES_ANO_REFERENCIA                datetime64[ns]
dtype: object

```

- Escolhemos calcular as idades das linhas que cujo valor é nulo já que todos tem o atributo DATA_NASCIMENTO preenchido

```

[254]: ##função para calcular idade
def age(born):
    born =datetime.strptime(born, "%d/%m/%Y").date()
    today = date.today()
    return today.year - born.year - ((today.month,
                                      today.day) < (born.month,
                                                      born.day))

```

```

[255]: df['IDADE'] = df['DATA_NASCIMENTO'].apply(age)
df.isnull().sum()

```

```

[255]: TEMPO_VIVE_NA_RUA          0
CONTATO_PARENTE_FORA_RUAS       0
DATA_NASCIMENTO                 0
IDADE                           0
SEXO                            0
BOLSA_FAMILIA                   0
POP_RUA                         0
GRAU_INSTRUCAO                  0
COR_RACA                        0
Faixa da renda familiar per capita 0
VAL_REMUNERACAO_MES_PASSADO      495
CRAS                            0
REGIONAL                        0
FAIXA_DESATUALICACAO_CADASTRAL    0
MES_ANO_REFERENCIA              0
dtype: int64

```

- as linhas com valores nulos de VAL_REMUNERACAO_MES_PASSADO serão retiradas visto que preencher com -1 ou algum texto marcados poderia alterar médias futuras e não temos informações suficientes para afirmar que todos os valores nulos dessa coluna podem ser substituídos por 0

```
[256]: df = df.drop(columns = ['VAL_REMUNERACAO_MES_PASSADO'])
```

6 Análise exploratória dos dados

Cabeçalho do Dataframe:

```
[257]: df.head()
```

```
[257]:
```

	TEMPO_VIVE_NA_RUA	...	MES_ANO_REFERENCIA
0	Entre um e dois anos	...	2020-01-09
1	Ate seis meses	...	2020-01-09
2	Entre cinco e dez anos	...	2020-01-09
3	Ate seis meses	...	2020-01-09
4	Entre cinco e dez anos	...	2020-01-09

[5 rows x 14 columns]

Tipos de dados:

```
[258]: df.dtypes
```

```
[258]:
```

TEMPO_VIVE_NA_RUA	object
CONTATO_PARENTE_FORA_RUAS	object
DATA_NASCIMENTO	object
IDADE	int64
SEXO	object
BOLSA_FAMILIA	object
POP_RUA	object
GRAU_INSTRUCAO	object
COR_RACA	object
Faixa da renda familiar per capita	object
CRAS	object
REGIONAL	object
FAIXA_DESATUALICACAO_CADASTRAL	object
MES_ANO_REFERENCIA	datetime64[ns]
dtype:	object

*O atributo “Faixa de renda familiar per capita” apresentou comportamento inesperado durante os trabalhos e o grupo optou por não utilizá-lo nas análises.

```
[259]: df.drop(['Faixa da renda familiar per capita'], axis=1)
```

```
[259]:
```

	TEMPO_VIVE_NA_RUA	...	MES_ANO_REFERENCIA
0	Entre um e dois anos	...	2020-01-09
1	Ate seis meses	...	2020-01-09
2	Entre cinco e dez anos	...	2020-01-09
3	Ate seis meses	...	2020-01-09

```

4      Entre cinco e dez anos ...      2020-01-09
...
95178      Ate seis meses ...      2021-01-07
95179      Entre um e dois anos ...      2021-01-07
95180      Ate seis meses ...      2021-01-07
95181      Entre cinco e dez anos ...      2021-01-07
95182      Entre dois e cinco anos ...      2021-01-07

```

```
[95183 rows x 13 columns]
```

#IDADE ### Qual a distribuição geral de idade dos moradores de rua em Belo Horizonte?

Observou-se que a maioria dos moradores tem entre 34 e 51 anos.

```
[260]: df['IDADE'].describe()
```

```

[260]: count      95183.000000
      mean         42.753811
      std         11.805424
      min          1.000000
      25%          34.000000
      50%          42.000000
      75%          51.000000
      max          84.000000
      Name: IDADE, dtype: float64

```

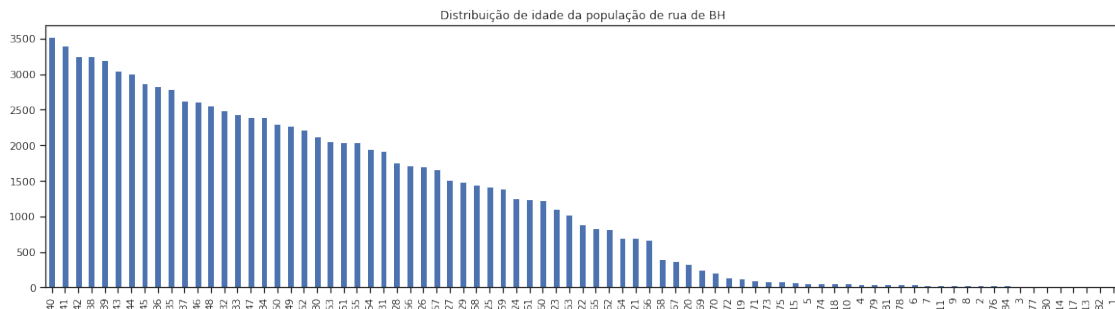
Gráfico por incidência:

Observa-se que a idade de 40 anos é a que possui mais representantes.

```

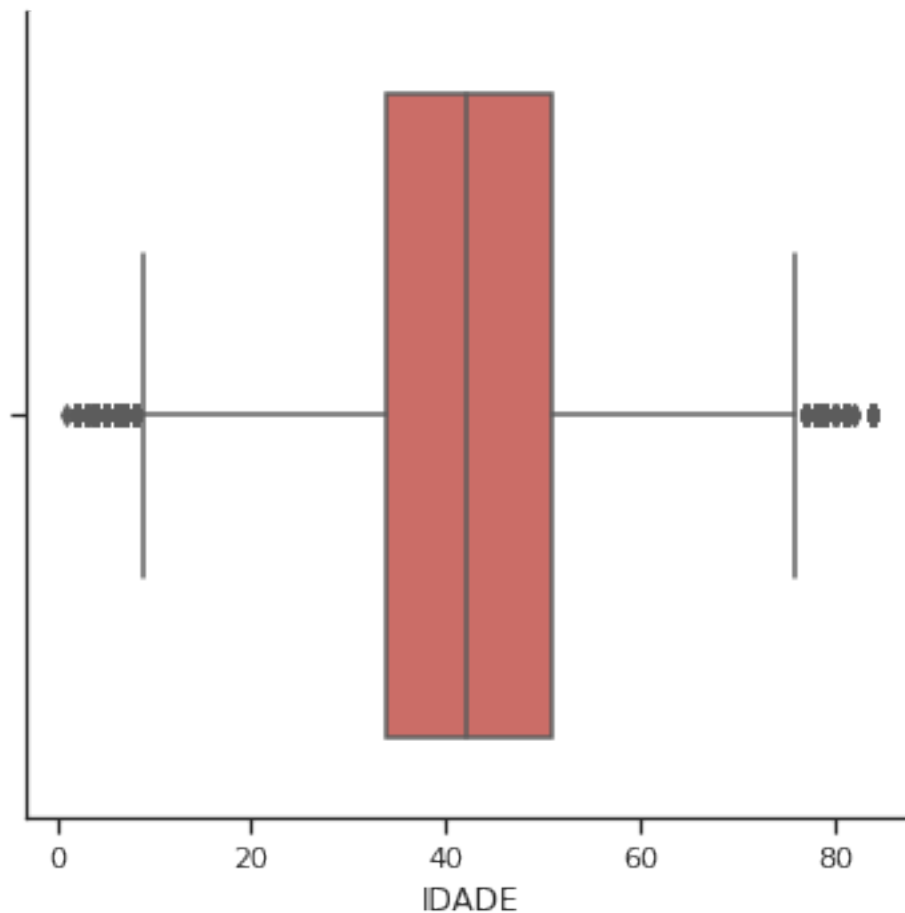
[261]: fig = plt.figure(figsize=(20,5))
      fig = df['IDADE'].value_counts().plot.bar()
      fig.set_title('Distribuição de idade da população de rua de BH')
      plt.show()

```



```
[262]: sns.catplot(x="IDADE", kind="box", palette="hls", data=df)
```

```
[262]: <seaborn.axisgrid.FacetGrid at 0x7f4c38e8dd90>
```



#SEXO

6.0.1 Qual é a distribuição do sexo da população de rua de BH?

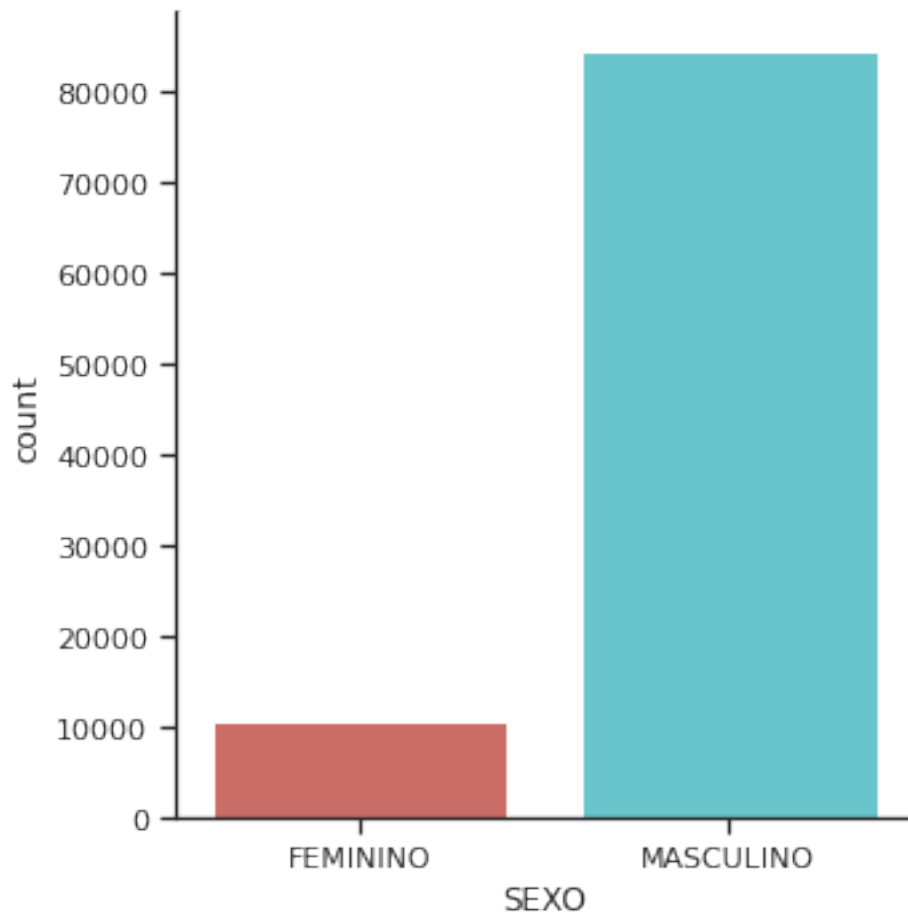
Quase 90% (89%, precisamente) da população de rua é do sexo masculino.

```
[263]: df['SEXO'].value_counts()
```

```
[263]: MASCULINO      84545
      FEMININO       10638
      Name: SEXO, dtype: int64
```

```
[264]: sns.catplot(x="SEXO", kind="count", palette="hls", data=df)
```

```
[264]: <seaborn.axisgrid.FacetGrid at 0x7f4c38e5dc90>
```

6.0.2 Qual a relação entre o contato com parentes e o tempo que estão em situação de rua?

- 46% das pessoas nunca veem seus parentes.
- 22% das pessoas quase nunca veem seus parentes.
- 14% das pessoas veem seus parentes todo mês.
- 9% das pessoas veem seus parentes toda semana.
- 5% das pessoas veem seus parentes todo ano.
- 4% das pessoas veem seus parentes todo dia.

```
[265]: df['CONTATO_PARENTE_FORA_RUAS'].value_counts()
```

```
[265]: Nunca          43304
       Quase nunca    21189
       Todo mes       13726
       Toda semana     8221
       Todo ano        5162
       Todo dia        3581
```

Name: CONTATO_PARENTE_FORA_RUAS, dtype: int64

- 42% das mulheres moradoras de rua nunca veem seus parentes
- 22% das mulheres moradoras de rua quase nunca veem seus parentes
- 16% das mulheres moradoras de rua veem seus parentes todo mês
- 11% das mulheres moradoras de rua veem seus parentes toda semana
- 5% das mulheres moradoras de rua veem seus parentes todo ano.
- 4% das mulheres moradoras de rua veem seus parentes todo dia.

```
[266]: fem = df[df['SEXO'] == 'FEMININO']  
fem['CONTATO_PARENTE_FORA_RUAS'].value_counts()
```

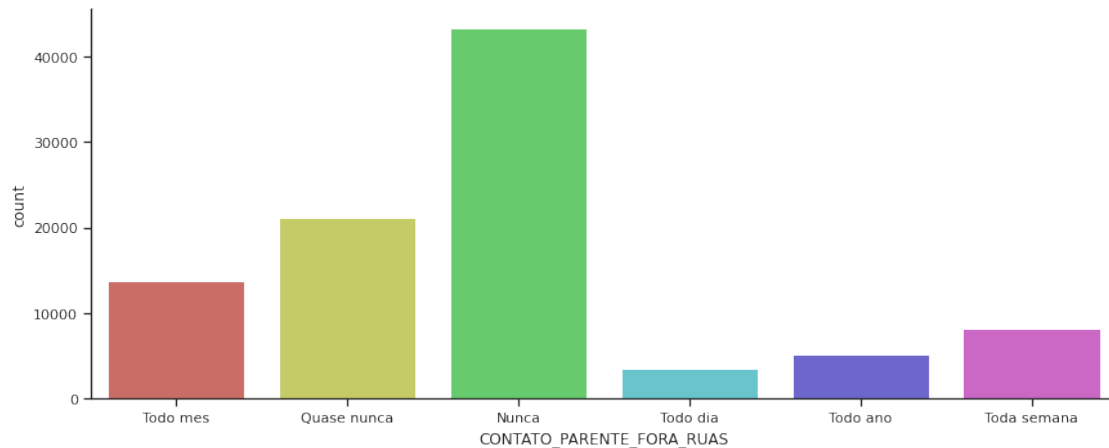
```
[266]: Nunca          4508  
Quase nunca      2352  
Todo mes         1654  
Toda semana      1110  
Todo ano         507  
Todo dia         507  
Name: CONTATO_PARENTE_FORA_RUAS, dtype: int64
```

- 46% dos homens moradores de rua nunca veem seus parentes.
- 22% dos homens moradores de rua quase nunca veem seus parentes.
- 14% dos homens moradores de rua veem seus parentes todo mês.
- 8% dos homens moradores de rua veem seus parentes toda semana.
- 6% dos homens moradores de rua veem seus parentes todo ano.
- 4% dos homens moradores de rua veem seus parentes todo dia.

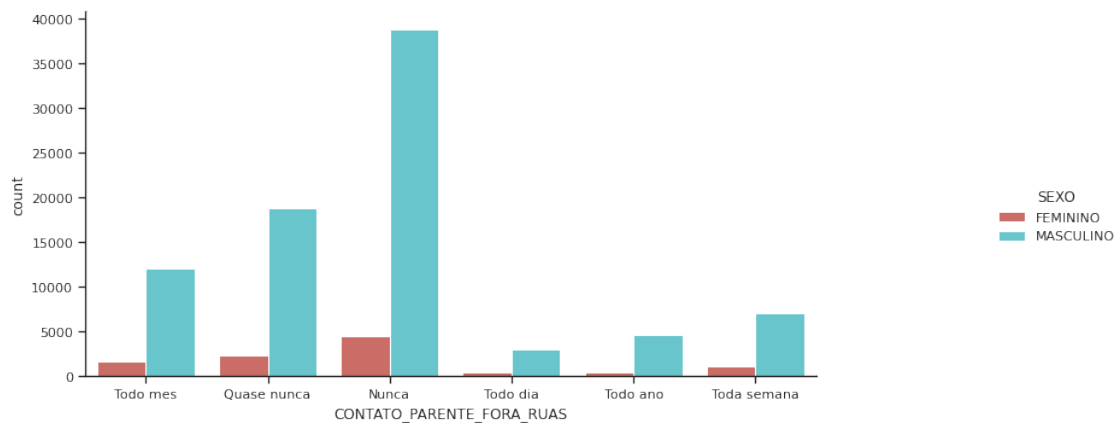
```
[267]: masc = df[df['SEXO'] == 'MASCULINO']  
masc['CONTATO_PARENTE_FORA_RUAS'].value_counts()
```

```
[267]: Nunca          38796  
Quase nunca      18837  
Todo mes         12072  
Toda semana       7111  
Todo ano         4655  
Todo dia         3074  
Name: CONTATO_PARENTE_FORA_RUAS, dtype: int64
```

```
[268]: g = sns.catplot(x="CONTATO_PARENTE_FORA_RUAS", kind="count", palette="hls",  
    ↪data=df)  
g.fig.set_size_inches(15,5)
```



```
[269]: g = sns.catplot(x="CONTATO_PARENTE_FORA_RUAS", kind="count", hue="SEXO",
    ↪ palette="hls", data=df)
g.fig.set_size_inches(15,5)
```



6.0.3 Qual a distribuição de idade de homens moradores de rua em Belo Horizonte?

A maioria dos homens tem entre 35 e 51 anos.

```
[270]: masc['IDADE'].value_counts()
```

```
[270]: 40    3063
      41    2965
      42    2918
      39    2885
      38    2850
      ...
      11     11
```

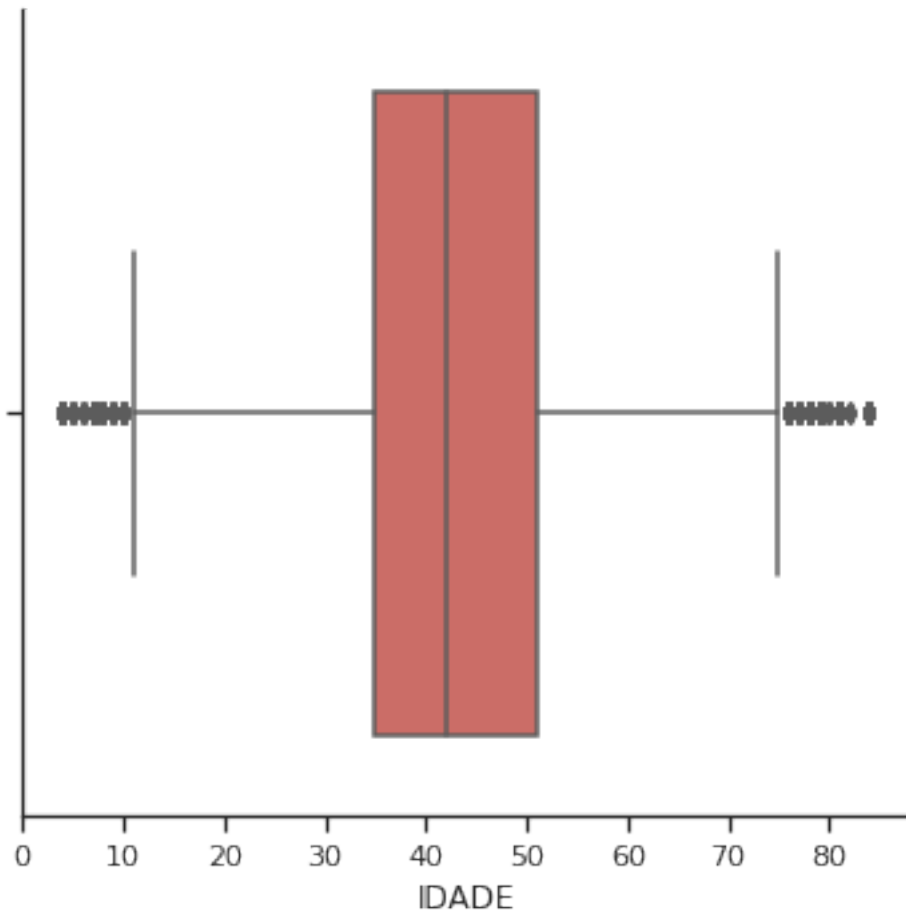
```
80      11
6       11
13      4
82      3
Name: IDADE, Length: 76, dtype: int64
```

```
[271]: masc['IDADE'].describe()
```

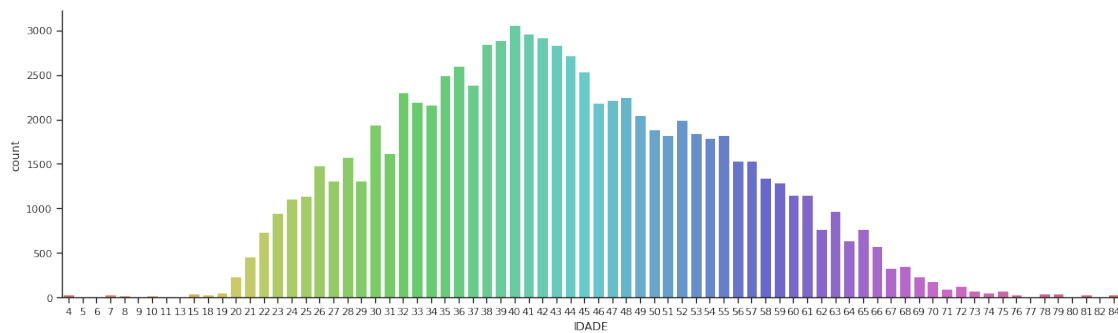
```
[271]: count      84545.000000
      mean        43.068449
      std         11.643860
      min         4.000000
      25%         35.000000
      50%         42.000000
      75%         51.000000
      max         84.000000
      Name: IDADE, dtype: float64
```

```
[272]: sns.catplot(x="IDADE", kind="box", palette="hls", data=masc)
```

```
[272]: <seaborn.axisgrid.FacetGrid at 0x7f4c3a121750>
```



```
[273]: g = sns.catplot(x="IDADE", kind="count", palette="hls", data=masc)
g.fig.set_size_inches(20,5)
```



6.0.4 Qual a distribuição de idade de mulheres moradoras de rua em Belo Horizonte?

Observou-se que a maioria das mulheres tem entre 33 e 49 anos de idade.

```
[274]: fem['IDADE'].value_counts()
```

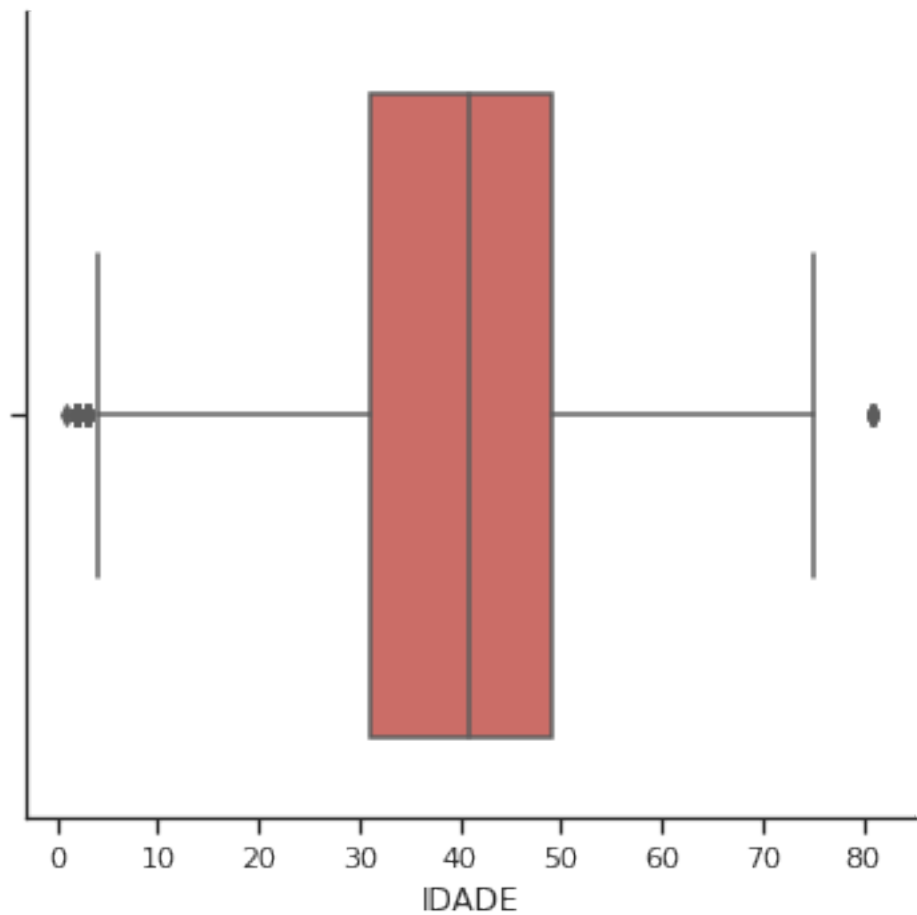
```
[274]: 40    455
      41    443
      46    428
      50    416
      38    403
      ...
      81     11
      8      11
      69      8
      7      8
      1      1
      Name: IDADE, Length: 71, dtype: int64
```

```
[275]: fem['IDADE'].describe()
```

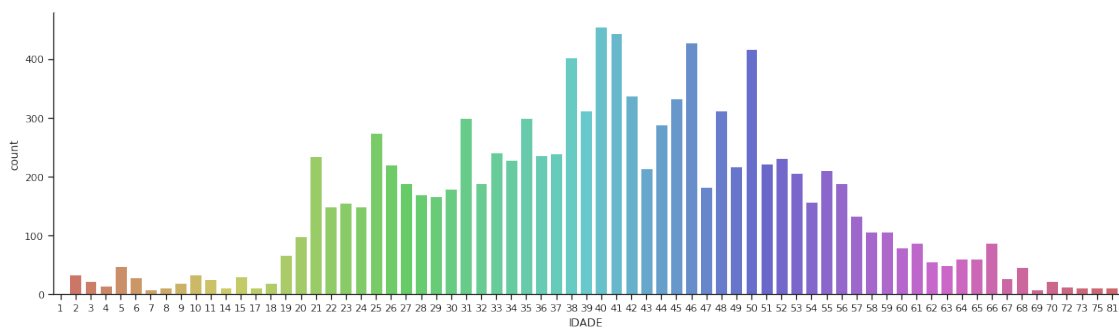
```
[275]: count    10638.000000
      mean      40.253243
      std       12.745695
      min       1.000000
      25%       31.000000
      50%       41.000000
      75%       49.000000
      max       81.000000
      Name: IDADE, dtype: float64
```

```
[276]: sns.catplot(x="IDADE", kind="box", palette="hls", data=fem)
```

```
[276]: <seaborn.axisgrid.FacetGrid at 0x7f4c429de690>
```



```
[277]: g = sns.catplot(x="IDADE", kind="count", palette="hls", data=fem)
g.fig.set_size_inches(20,5)
```



#BOLSA FAMILIA

Separando por mês:

```
[278]: #Leitura individual de cada um
df920 = pd.DataFrame(data9_20[u'BOLSA_FAMILIA'].value_counts())
df1020 = pd.DataFrame(data10_20[u'BOLSA_FAMILIA'].value_counts())
df1120 = pd.DataFrame(data11_20[u'BOLSA_FAMILIA'].value_counts())
df1220 = pd.DataFrame(data12_20[u'BOLSA_FAMILIA'].value_counts())
df121 = pd.DataFrame(data1_21[u'BOLSA_FAMILIA'].value_counts())
df221 = pd.DataFrame(data2_21[u'BOLSA_FAMILIA'].value_counts())
df321 = pd.DataFrame(data3_21[u'BOLSA_FAMILIA'].value_counts())
df421 = pd.DataFrame(data4_21[u'BOLSA_FAMILIA'].value_counts())
df521 = pd.DataFrame(data5_21[u'BOLSA_FAMILIA'].value_counts())
df621 = pd.DataFrame(data6_21[u'BOLSA_FAMILIA'].value_counts())
df721 = pd.DataFrame(data7_21[u'BOLSA_FAMILIA'].value_counts())
```

```
[279]: df920.rename(columns={'BOLSA_FAMILIA': '09/20'}, inplace = True)
df1020.rename(columns={'BOLSA_FAMILIA': '10/20'}, inplace = True)
df1120.rename(columns={'BOLSA_FAMILIA': '11/20'}, inplace = True)
df1220.rename(columns={'BOLSA_FAMILIA': '12/20'}, inplace = True)
df121.rename(columns={'BOLSA_FAMILIA': '1/21'}, inplace = True)
df221.rename(columns={'BOLSA_FAMILIA': '2/21'}, inplace = True)
df321.rename(columns={'BOLSA_FAMILIA': '3/21'}, inplace = True)
df421.rename(columns={'BOLSA_FAMILIA': '4/21'}, inplace = True)
df521.rename(columns={'BOLSA_FAMILIA': '5/21'}, inplace = True)
df621.rename(columns={'BOLSA_FAMILIA': '6/21'}, inplace = True)
df721.rename(columns={'BOLSA_FAMILIA': '7/21'}, inplace = True)
```

```
[280]: dfBolsa = df920.join([df1020,df1120,df1220,
                           df121,df221,df321,df421,df521,df621,df721])

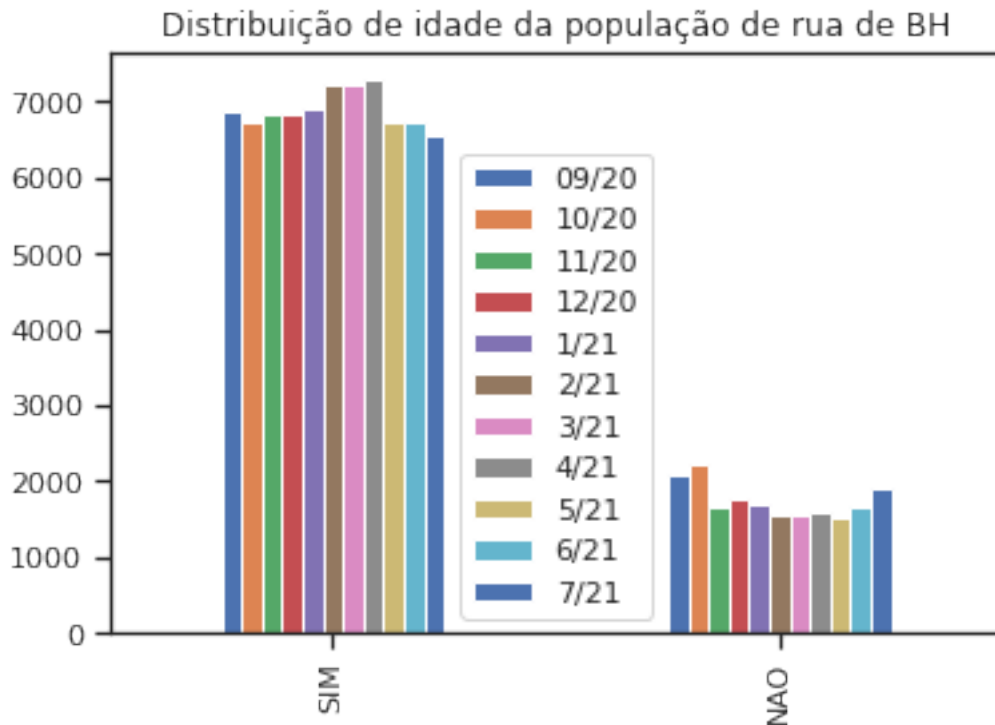
dfBolsa
```

```
[280]:
```

	09/20	10/20	11/20	12/20	1/21	2/21	3/21	4/21	5/21	6/21	7/21
SIM	6880	6744	6826	6823	6913	7217	7217	7293	6746	6728	6559
NAO	2096	2222	1676	1754	1706	1540	1540	1608	1536	1646	1913

```
[281]: fig = plt.figure(figsize=(100,100))
fig = dfBolsa.plot.bar()
fig.set_title('Distribuição de idade da população de rua de BH')
plt.show()
```

<Figure size 7200x7200 with 0 Axes>



APONTAMENTOS: * Observa-se que o benefício é essencial para estes moradores, com mais que 70% de cobertura. *Para a confirmação de recursos enviados pelo governo para manutenção deste programa, foi consultado o dataframe “beneficios.csv”, presente no repositório deste trabalho.

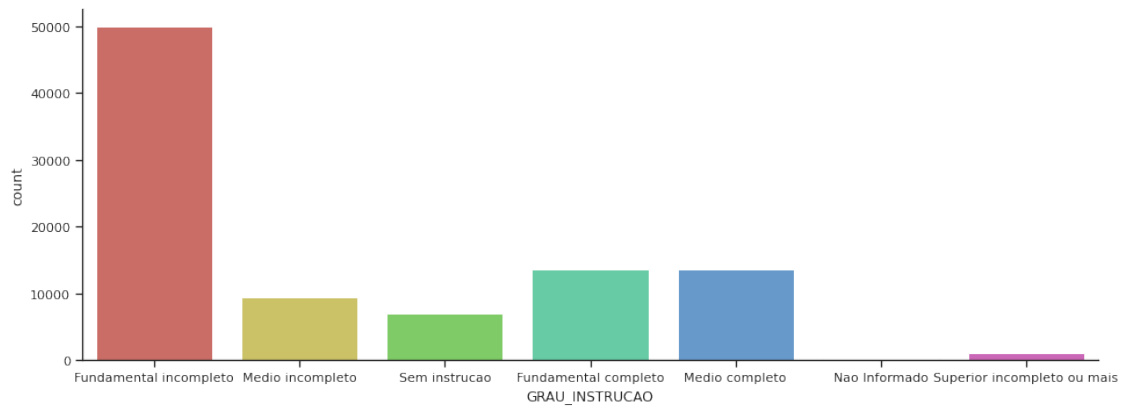
```
[282]: #Comandos para acesso via upload de arquivo:
#df_beneficios = pd.read_csv('/beneficio.csv', index_col=False, sep = ";")
#df_beneficiosBolsa
```

#GRAU DE INSTRUÇÃO

```
[283]: df['GRAU_INSTRUCAO'].value_counts()
```

```
[283]: Fundamental incompleto      49992
Fundamental completo             13679
Medio completo                   13631
Medio incompleto                 9430
Sem instrucao                    7063
Superior incompleto ou mais      1116
Nao Informado                    272
Name: GRAU_INSTRUCAO, dtype: int64
```

```
[285]: g = sns.catplot(x="GRAU_INSTRUCAO", kind="count", palette="hls", data=df)
g.fig.set_size_inches(20,5)
```



O Fundamental incompleto é predominante, indicando a necessidade de intervenção também a nível educacional.

#Tempo em que os moradores vivem na rua?

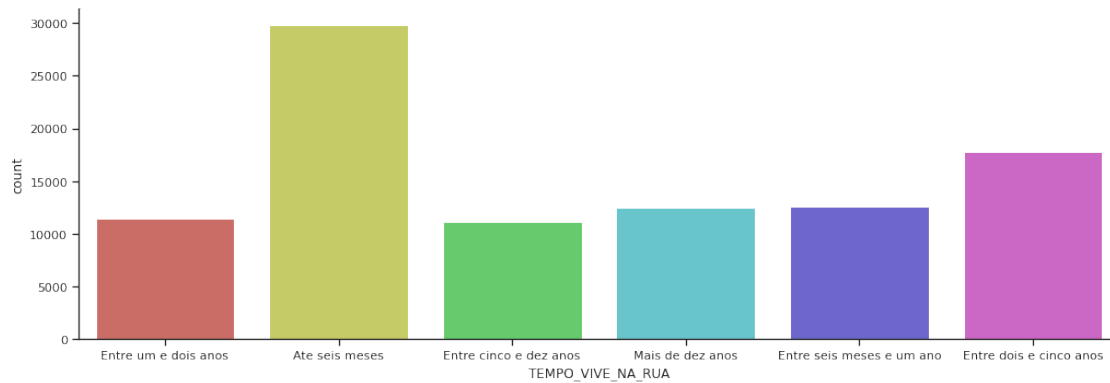
```
[286]: df.columns
```

```
[286]: Index(['TEMPO_VIVE_NA_RUA', 'CONTATO_PARENTE_FORA_RUAS', 'DATA_NASCIMENTO',
        'IDADE', 'SEXO', 'BOLSA_FAMILIA', 'POP_RUA', 'GRAU_INSTRUCAO',
        'COR_RACA', 'Faixa da renda familiar per capita', 'CRAS', 'REGIONAL',
        'FAIXA_DESATUALICACAO_CADASTRAL', 'MES_ANO_REFERENCIA'],
        dtype='object')
```

```
[288]: df['TEMPO_VIVE_NA_RUA'].value_counts()
```

```
[288]: Ate seis meses                29833
Entre dois e cinco anos           17798
Entre seis meses e um ano         12560
Mais de dez anos                  12448
Entre um e dois anos              11407
Entre cinco e dez anos            11137
Name: TEMPO_VIVE_NA_RUA, dtype: int64
```

```
[289]: g = sns.catplot(x="TEMPO_VIVE_NA_RUA", kind="count", palette="hls", data=df)
g.fig.set_size_inches(20,5)
```

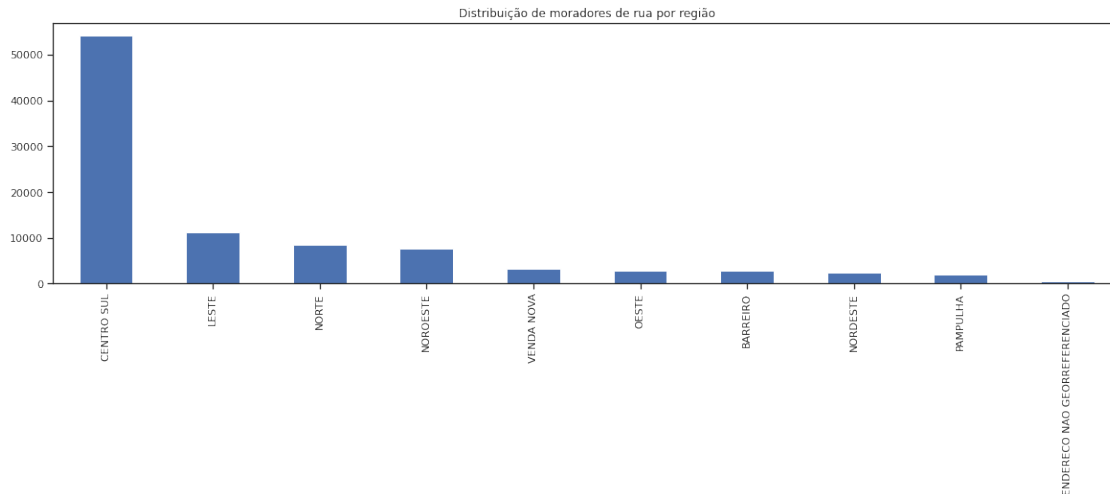


#Qual o percentual de moradores de rua em cada regional de Belo Horizonte?

```
[290]: df['REGIONAL'].value_counts()
```

```
[290]: CENTRO SUL          54242
LESTE                     11277
NORTE                     8387
NOROESTE                  7721
VENDA NOVA                3185
OESTE                     2833
BARREIRO                  2809
NORDESTE                  2317
PAMPULHA                  1917
ENDERECO NAO GEORREFERENCIADO  495
Name: REGIONAL, dtype: int64
```

```
[291]: fig = plt.figure(figsize=(20,5))
fig = df['REGIONAL'].value_counts().plot.bar()
fig.set_title('Distribuição de moradores de rua por região')
plt.show()
```



#Quais são a raça e o sexo predominante de cada regional presente no banco de dados?

Com base nos dados obtidos nas regionais o sexo predominante foi o **masculino** e a raça predominante foi **parda**, como verificado abaixo:

```
[292]: group_regional = df.groupby('REGIONAL')
group_centroSul = group_regional.get_group('CENTRO SUL')
group_leste = group_regional.get_group('LESTE')
group_norte = group_regional.get_group('NORTE')
group_noroeste = group_regional.get_group('NOROESTE')
group_vendaNova = group_regional.get_group('VENDA NOVA')
group_oeste = group_regional.get_group('OESTE')
group_barreiro = group_regional.get_group('BARREIRO')
group_nordeste = group_regional.get_group('NORDESTE')
group_pampulha = group_regional.get_group('PAMPULHA')
```

```
[293]: group_centroSul['SEXO'].value_counts()
```

```
[293]: MASculINO      49301
FEMININO      4941
Name: SEXO, dtype: int64
```

```
[294]: group_centroSul['COR_RACA'].value_counts()
```

```
[294]: Parda      32237
Preta      12573
Branca      9037
Amarela      257
Indigena      71
Nao Informado    67
Name: COR_RACA, dtype: int64
```

```
[295]: group_leste['SEXO'].value_counts()
```

```
[295]: MASCULINO    10202  
      FEMININO     1075  
      Name: SEXO, dtype: int64
```

```
[296]: group_leste['COR_RACA'].value_counts()
```

```
[296]: Parda          6854  
      Preta         2661  
      Branca        1680  
      Amarela         52  
      Indigena        19  
      Nao Informado   11  
      Name: COR_RACA, dtype: int64
```

```
[297]: group_norte['SEXO'].value_counts()
```

```
[297]: MASCULINO     6918  
      FEMININO    1469  
      Name: SEXO, dtype: int64
```

```
[298]: group_norte['COR_RACA'].value_counts()
```

```
[298]: Parda          5103  
      Preta         2007  
      Branca        1245  
      Amarela         21  
      Nao Informado   11  
      Name: COR_RACA, dtype: int64
```

```
[299]: group_noroeste['SEXO'].value_counts()
```

```
[299]: MASCULINO     6640  
      FEMININO    1081  
      Name: SEXO, dtype: int64
```

```
[300]: group_noroeste['COR_RACA'].value_counts()
```

```
[300]: Parda          4776  
      Preta         1811  
      Branca        1099  
      Amarela         30  
      Indigena         5  
      Name: COR_RACA, dtype: int64
```

```
[301]: group_vendaNova['SEXO'].value_counts()
```

```
[301]: MASCULINO    2726  
      FEMININO     459  
      Name: SEX0, dtype: int64
```

```
[302]: group_vendaNova['COR_RACA'].value_counts()
```

```
[302]: Parda          1975  
      Preta          716  
      Branca         464  
      Amarela         26  
      Nao Informado    4  
      Name: COR_RACA, dtype: int64
```

```
[303]: group_oeste['SEX0'].value_counts()
```

```
[303]: MASCULINO    2448  
      FEMININO    385  
      Name: SEX0, dtype: int64
```

```
[304]: group_oeste['COR_RACA'].value_counts()
```

```
[304]: Parda          1865  
      Preta          489  
      Branca         471  
      Amarela         8  
      Name: COR_RACA, dtype: int64
```

```
[305]: group_barreiro['SEX0'].value_counts()
```

```
[305]: MASCULINO    2454  
      FEMININO    355  
      Name: SEX0, dtype: int64
```

```
[306]: group_barreiro['COR_RACA'].value_counts()
```

```
[306]: Parda          1844  
      Preta          618  
      Branca         314  
      Amarela         22  
      Nao Informado    11  
      Name: COR_RACA, dtype: int64
```

```
[307]: group_nordeste['SEX0'].value_counts()
```

```
[307]: MASCULINO    1904  
      FEMININO    413  
      Name: SEX0, dtype: int64
```

```
[308]: group_nordeste['COR_RACA'].value_counts()
```

```
[308]: Parda          1343
      Preta          621
      Branca         331
      Nao Informado   11
      Amarela         7
      Indigena        4
      Name: COR_RACA, dtype: int64
```

```
[309]: group_pampulha['SEXO'].value_counts()
```

```
[309]: MASCULINO    1567
      FEMININO     350
      Name: SEXO, dtype: int64
```

```
[310]: group_pampulha['COR_RACA'].value_counts()
```

```
[310]: Parda      1223
      Preta      417
      Branca     277
      Name: COR_RACA, dtype: int64
```

#Qual o percentual de moradores de rua cadastrados em um Cras?

```
[311]: df['CRAS'].value_counts()
```

```
[311]: ENDERECO FORA AREA CRAS          58521
      ENDERECO FORA DE AREA CRAS    22179
      CRAS PROVIDENCIA              5966
      CRAS VILA SENHOR DOS PASSOS   3313
      CRAS VILA BIQUINHAS           1895
      ENDERECO NAO GEORREFERENCIADO  495
      CRAS GRANJA DE FREITAS        281
      CRAS MANTIQUEIRA              242
      CRAS PETROPOLIS               167
      CRAS VILA SAO JOSE             128
      CRAS VILA MARIA               121
      CRAS ZILAH SPOSITO            117
      CRAS VILA SANTA RITA DE CASSIA 115
      CRAS NOVO AARAO REIS  BRASILINA MARIA DE OLIVEIRA 110
      CRAS VILA MARCOLA             108
      CRAS PEDREIRA PRADO LOPES     100
      CRAS TAQUARIL                 99
      CRAS MORRO DAS PEDRAS  GRACA SABOIA                98
      CRAS ARTHUR DE SA  UNIAO                          91
      CRAS VILA NOSSA SENHORA DE FATIMA 89
```

CRAS SANTA ROSA	80
CRAS INDEPENDENCIA	77
CRAS VILA COQUEIRAL	73
CRAS VILA CEMIG	71
CRAS ALTO VERA CRUZ	68
CRAS JARDIM FELICIDADE	67
CRAS NOVO OURO PRETO	63
CRAS LAGOA	60
CRAS CONJUNTO PAULO VI	56
CRAS VILA ANTENA	53
CRAS HAVAI VENTOSA	52
CRAS VISTA ALEGRE	43
CRAS CALIFORNIA	38
CRAS MARIANO DE ABREU	37
CRAS CONFISCO	37
CRAS VILA SUMARE	35
CRAS APOLONIA	25
CRAS VILA SANTA ROSA	9
CRAS VILA FATIMA	4

Name: CRAS, dtype: int64

```
[312]: total = df['CRAS'].value_counts().sum()
fora_CRAS = 58521 + 22179
fora_CRAS
dentro_CRAS = total - fora_CRAS
percentual = (dentro_CRAS*100)/total
print("O percentual de moradores de rua que estão cadastrados em um CRAS é: ",
      percentual,"%")
```

O percentual de moradores de rua que estão cadastrados em um CRAS é:
15.215952428479875 %

#A cobertura do CRAS aumentou ou diminuiu desde o início da pesquisa?(09/2020 a 07/2021)

```
[313]: df['MES_ANO_REFERENCIA'].value_counts()
```

```
[313]: 2021-01-02    17514
2020-01-09      8976
2020-01-10      8966
2021-01-04      8901
2021-01-01      8619
2020-01-12      8577
2020-01-11      8502
2021-01-07      8472
2021-01-06      8374
2021-01-05      8282
```

Name: MES_ANO_REFERENCIA, dtype: int64

Aqui foram retirados endereços que não nos ajudam nessa análise, como os valores marcados com ‘ENDERECO FORA AREA CRAS’, ‘ENDERECO FORA DE AREA CRAS’, ‘ENDERECO NAO GEORREFERENCIADO’

```
[314]: df2 = df
df2 = df2.drop(df2[(df2['CRAS'] == 'ENDERECO FORA AREA CRAS') | (df2['CRAS'] == 'ENDERECO FORA DE AREA CRAS ') | (df2['CRAS'] == 'ENDERECO NAO GEORREFERENCIADO')].index)
df2
```

```
[314]:
```

	TEMPO_VIVE_NA_RUA	...	MES_ANO_REFERENCIA
0	Entre um e dois anos	...	2020-01-09
1	Ate seis meses	...	2020-01-09
2	Entre cinco e dez anos	...	2020-01-09
3	Ate seis meses	...	2020-01-09
4	Entre cinco e dez anos	...	2020-01-09
...
95073	Entre dois e cinco anos	...	2021-01-07
95078	Entre dois e cinco anos	...	2021-01-07
95080	Entre um e dois anos	...	2021-01-07
95101	Entre dois e cinco anos	...	2021-01-07
95140	Ate seis meses	...	2021-01-07

[36167 rows x 14 columns]

```
[315]: df2['CRAS'].value_counts()
```

```
[315]:
```

ENDERECO FORA DE AREA CRAS	22179
CRAS PROVIDENCIA	5966
CRAS VILA SENHOR DOS PASSOS	3313
CRAS VILA BIQUINHAS	1895
CRAS GRANJA DE FREITAS	281
CRAS MANTIQUEIRA	242
CRAS PETROPOLIS	167
CRAS VILA SAO JOSE	128
CRAS VILA MARIA	121
CRAS ZILAH SPOSITO	117
CRAS VILA SANTA RITA DE CASSIA	115
CRAS NOVO AARAO REIS BRASILINA MARIA DE OLIVEIRA	110
CRAS VILA MARCOLA	108
CRAS PEDREIRA PRADO LOPES	100
CRAS TAQUARIL	99
CRAS MORRO DAS PEDRAS GRACA SABOIA	98
CRAS ARTHUR DE SA UNIAO	91
CRAS VILA NOSSA SENHORA DE FATIMA	89
CRAS SANTA ROSA	80
CRAS INDEPENDENCIA	77
CRAS VILA COQUEIRAL	73

CRAS VILA CEMIG	71
CRAS ALTO VERA CRUZ	68
CRAS JARDIM FELICIDADE	67
CRAS NOVO OURO PRETO	63
CRAS LAGOA	60
CRAS CONJUNTO PAULO VI	56
CRAS VILA ANTENA	53
CRAS HAVAI VENTOSA	52
CRAS VISTA ALEGRE	43
CRAS CALIFORNIA	38
CRAS MARIANO DE ABREU	37
CRAS CONFISCO	37
CRAS VILA SUMARE	35
CRAS APOLONIA	25
CRAS VILA SANTA ROSA	9
CRAS VILA FATIMA	4

Name: CRAS, dtype: int64

```
[316]: #g = sns.catplot(x="CRAS", kind="count", hue="MES_ANO_REFERENCIA",
→palette="hls", data=df2)
#g.fig.set_size_inches(15,5)
```

***Este levantamento passará por ajustes até a próxima entrega e apresentação, visando uma representação visual adequada. Algumas das perguntas iniciais serão tratadas na apresentação pois acreditamos que o modelo preditivo ajudará a respondê-las.**