

Data Engineering (Big Data & Analytics)

Dabla Arévalo Ferreira
Gabrielle Brito Cadurim
Larissa Alves da Silva
Mateus Soares da Silva
Vinicius Miranda Lopes Schulz

Utilizamos o ambiente do Vagrant para a criação de alguns gráficos e o CodiLab para outros.

Testamos todos os gráficos pelo Vagrant, foi escolha de cada a utilização de qual gráfico e código usar.

Subimos o passo a passo de cada código no github, do qual tem as informações detalhadas de como fizemos.

GitHub : <https://github.com/vinicio-schulz/data-engineering-trabalho-final.git>

1. Total de acidentes com vítima por bairro em acidentes com embriaguez;

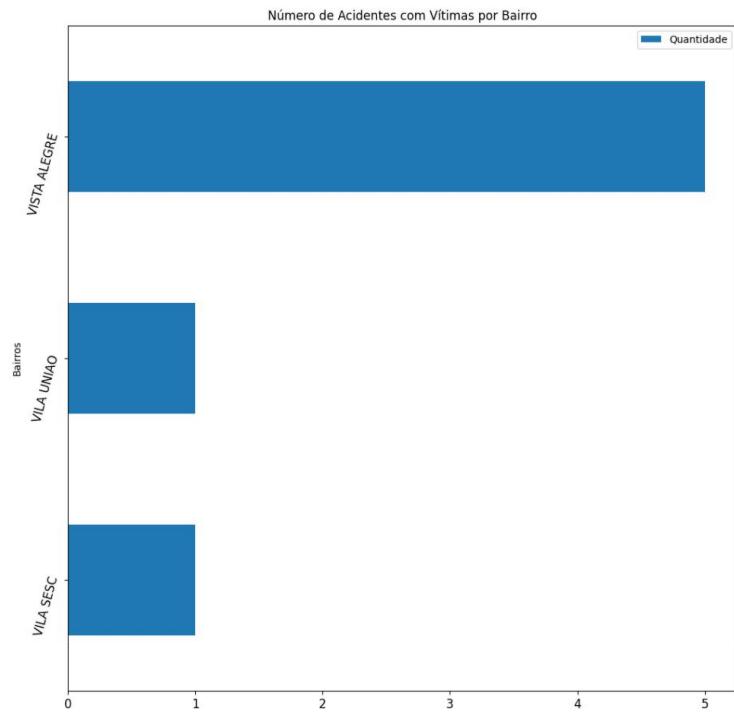
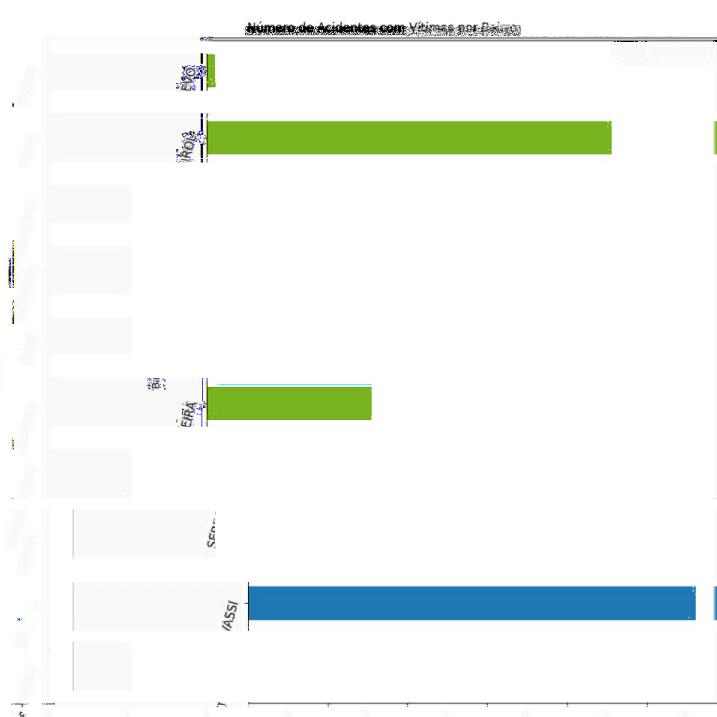
Usando o ambiente Vagrant para realizar a análise de dados:

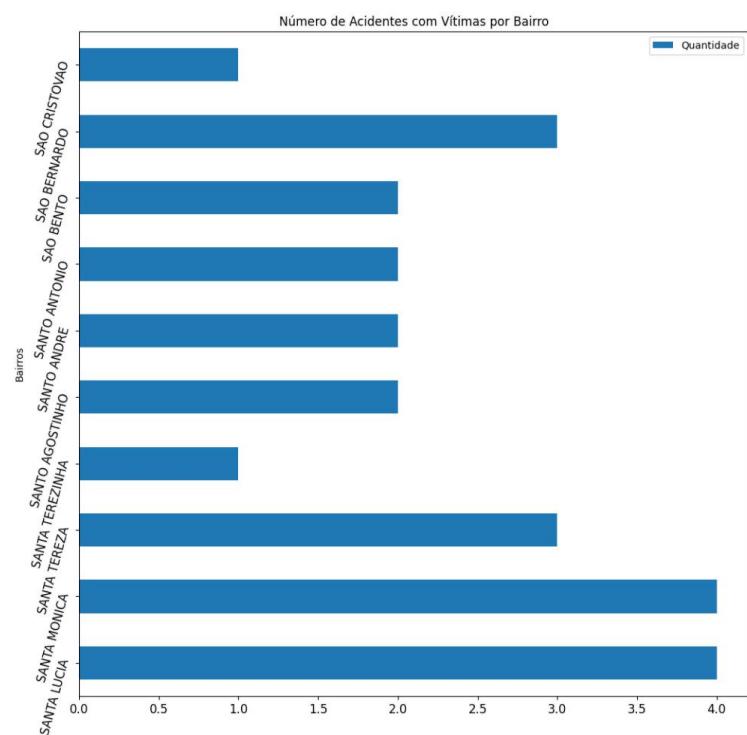
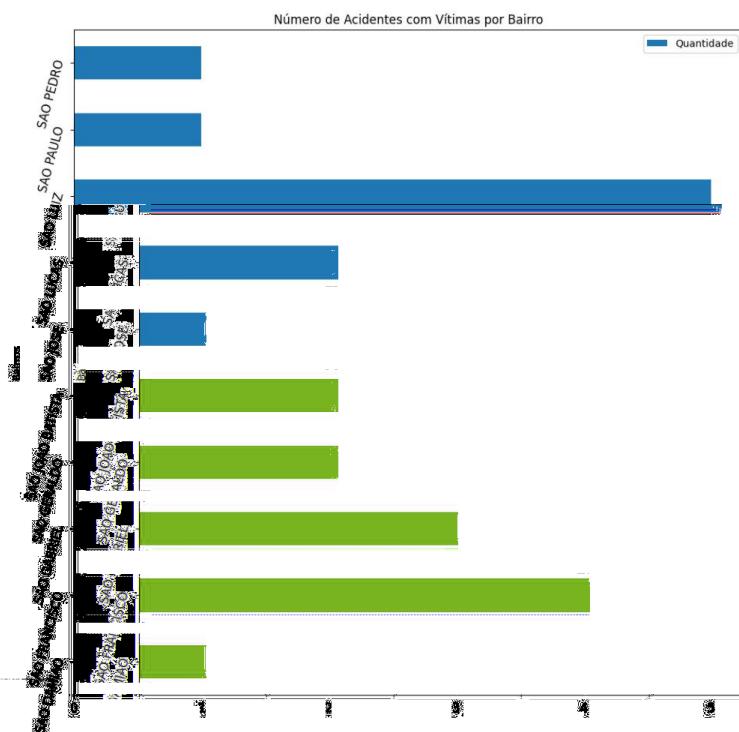
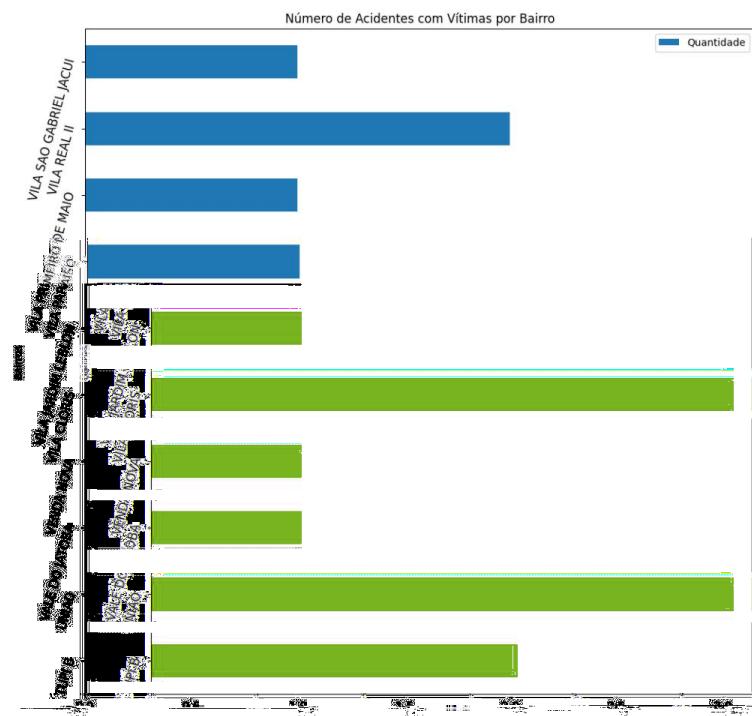
```
from pyspark.sql import SparkSession
from pyspark.sql import Row
import matplotlib.pyplot as plt
spark = SparkSession.builder.appName("Total de acidentes com vitima por bairro em acidentes com
embriaguez").enableHiveSupport().getOrCreate()
spark.sql("use gpdb")
df = spark.sql("SELECT trim(log.`nome_bairro`) as nome_bairro, COUNT(log.`nome_bairro`) as Quantidade
FROM si_log log JOIN si_bol bol ON bol.`NUMERO_BOLETIM` = log.`Nº_boletim` WHERE EXISTS (SELECT *
FROM si_env env where env.`Embreagues` = 'SIM' AND bol.`NUMERO_BOLETIM` = env.`num_boletim`) AND
bol.`DESC_TIPO_ACIDENTE` NOT LIKE '%SEM VITIMA%' GROUP BY log.`Nº_bairro`, log.`nome_bairro` 
ORDER BY log.`nome_bairro`")

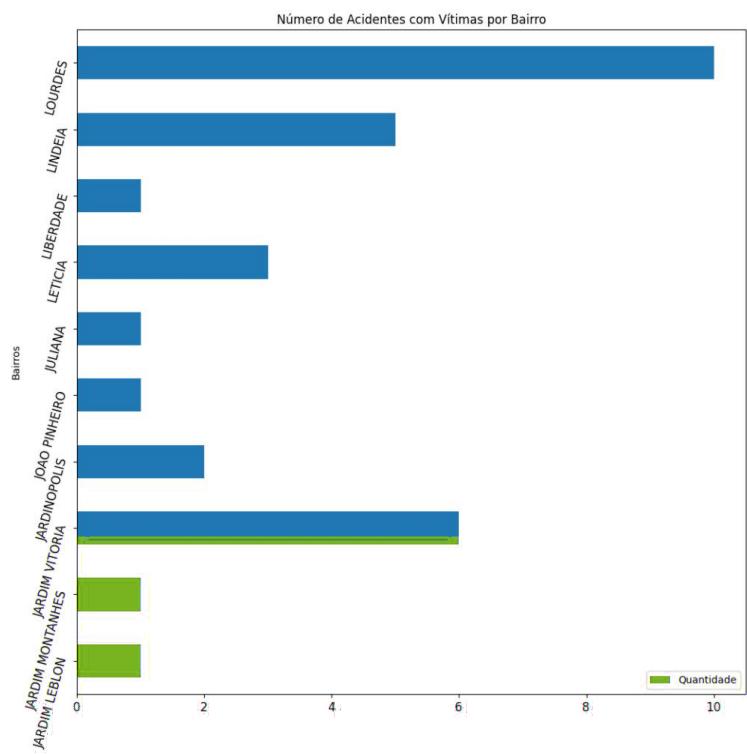
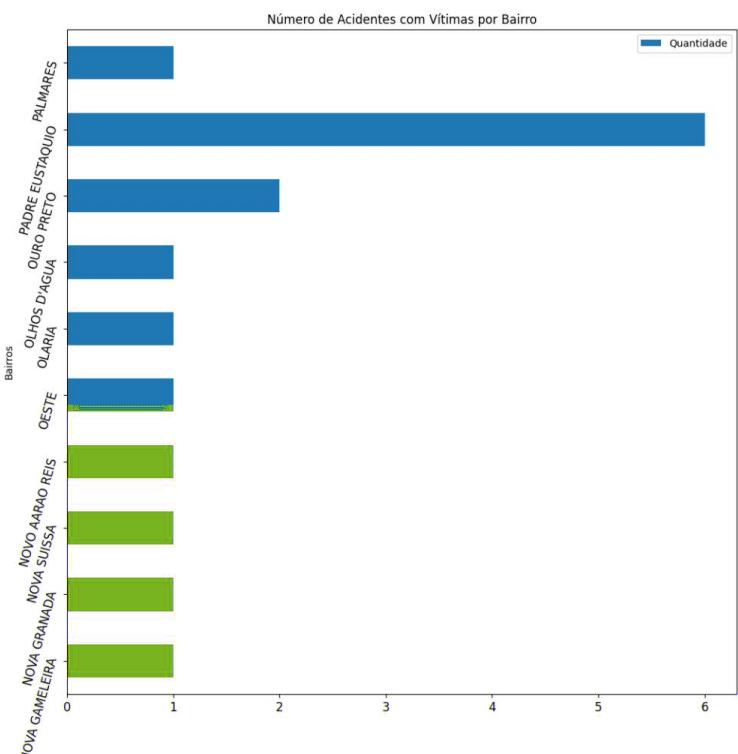
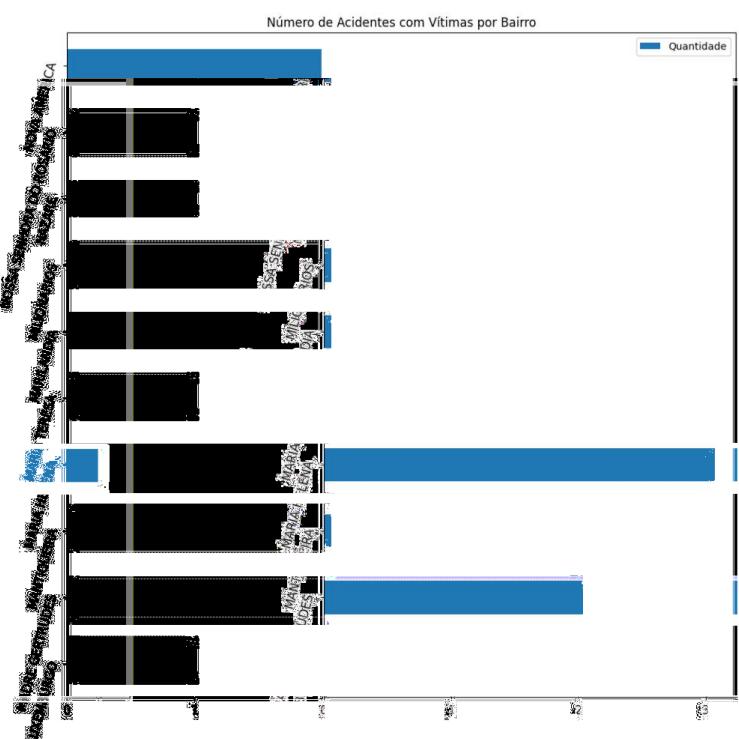
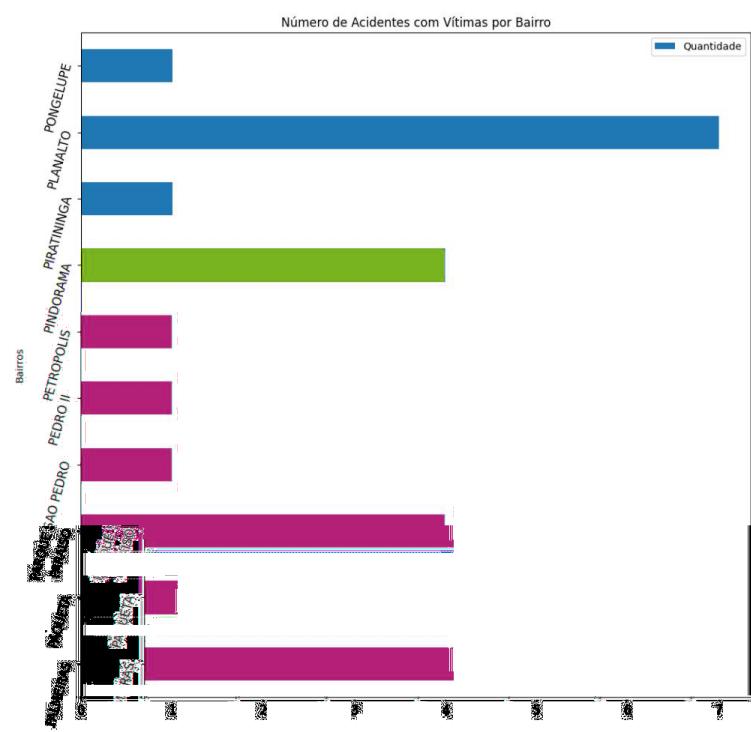
dfpandas=df.toPandas()

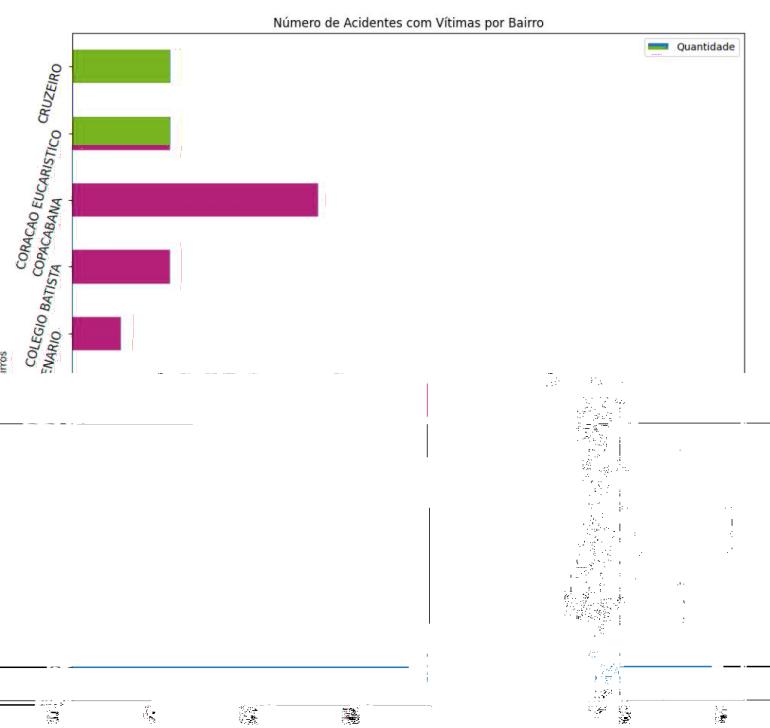
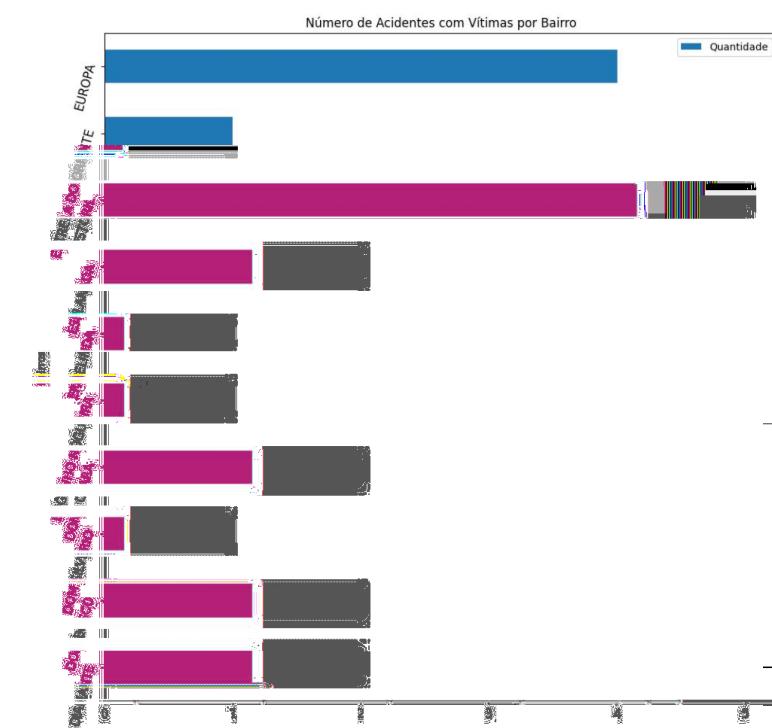
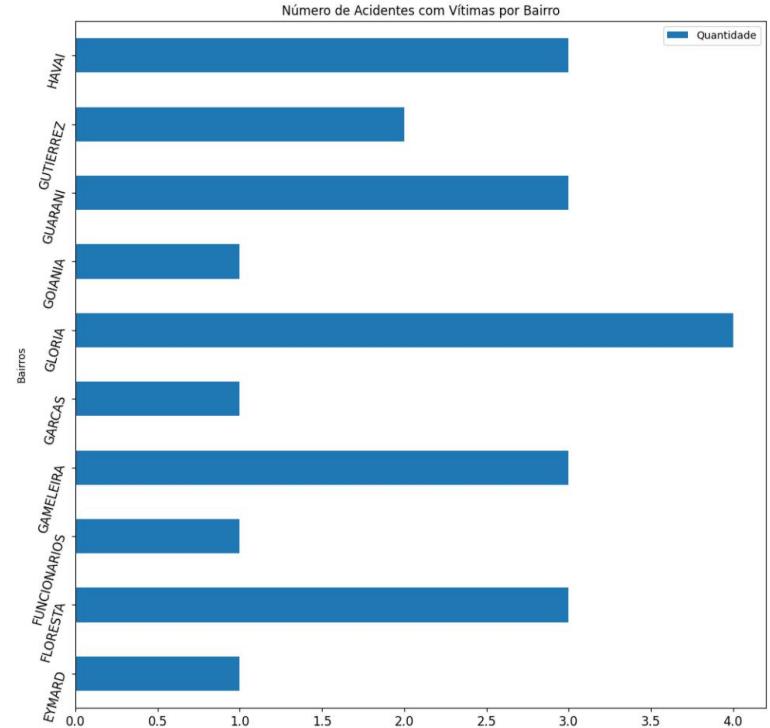
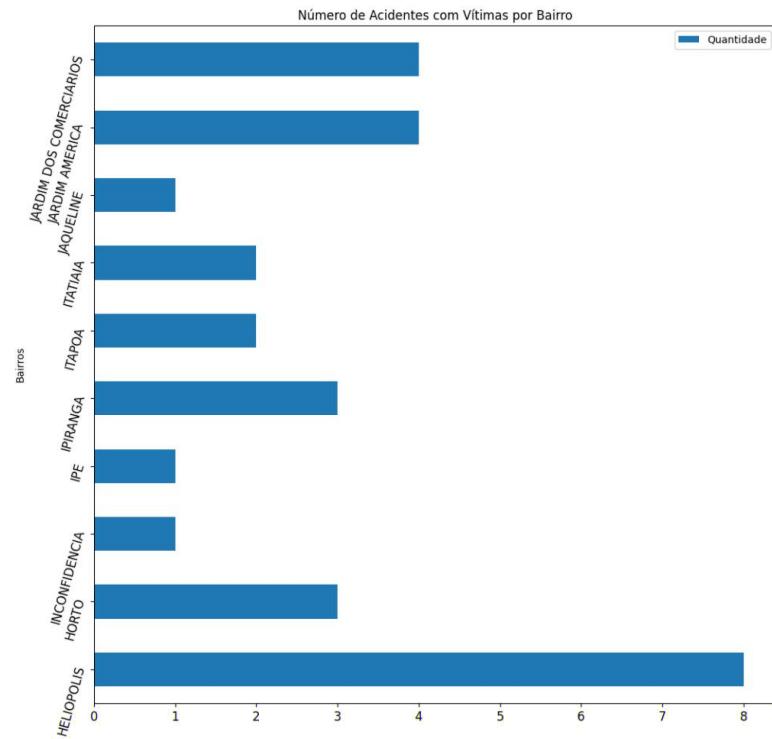
size = 10
df_dict = {n: dfpandas.iloc[n:n+size, :] for n in range(0, len(dfpandas), size)}

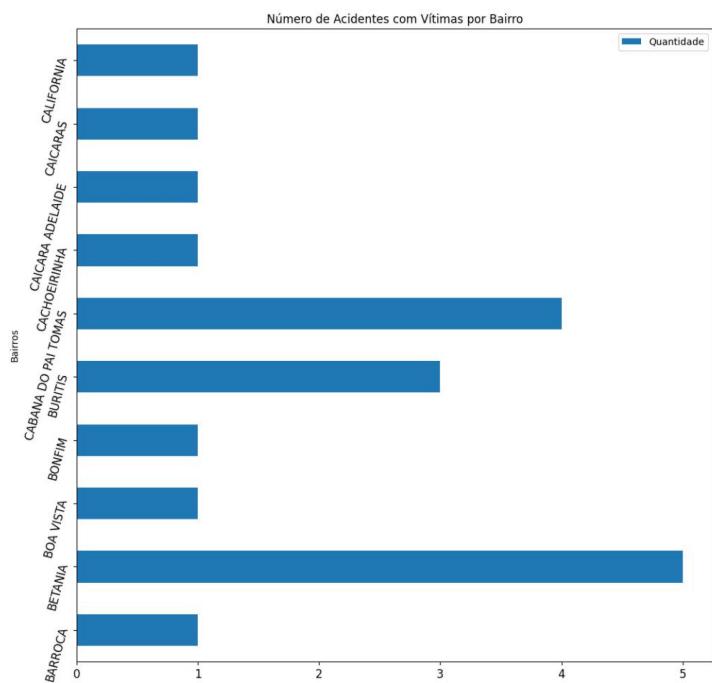
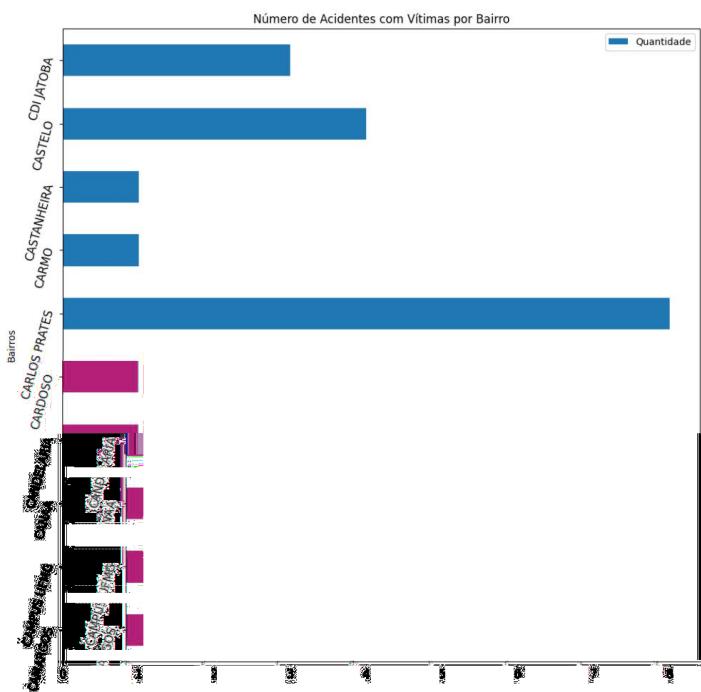
for key in df_dict:
    plt.clf()
    plt.close()
    df_dict[key].plot.bart(y='Quantidade', x='nome_bairro', rot=75, figsize=(12, 12), fontsize=12, title='Número de
Acidentes com Vítimas por Bairro', xlabel='Bairros')
    plt.savefig('output/output'+str(key)+'.png')
```









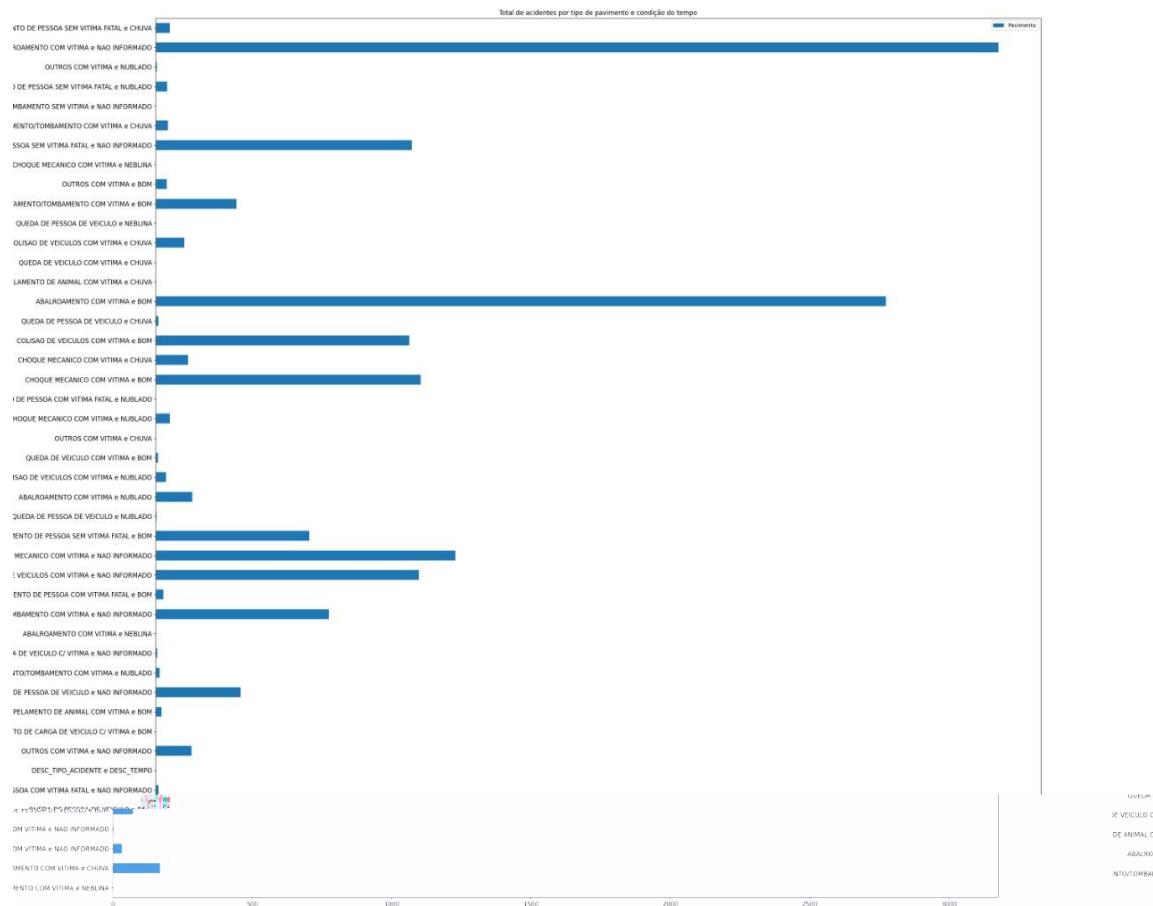


2. Total de acidentes por tipo de pavimento e condição do tempo;

Usando o ambiente Vagrant para realizar a análise de dados:

```
from pyspark.sql import SparkSession
from pyspark.sql import Row
import matplotlib.pyplot as plt
spark = SparkSession.builder.appName("Total de pessoas acidentadas por tipo de veiculo e tipo de pavimentação").enableHiveSupport().getOrCreate()
spark.sql("use gpdb")
df = spark.sql("SELECT CONCAT_WS(' e ', trim(env.especie_veiculo), trim(bol.PAVIMENTO)) as description,
count(*) as amount FROM si_log log JOIN si_bol bol ON bol.`NUMERO_BOLETIM` = log.`Nº_boletim` JOIN
si_env env ON env.`num_boletim` = log.`Nº_boletim` GROUP BY env.especie_veiculo, bol.PAVIMENTO")
dfpandas=df.toPandas()

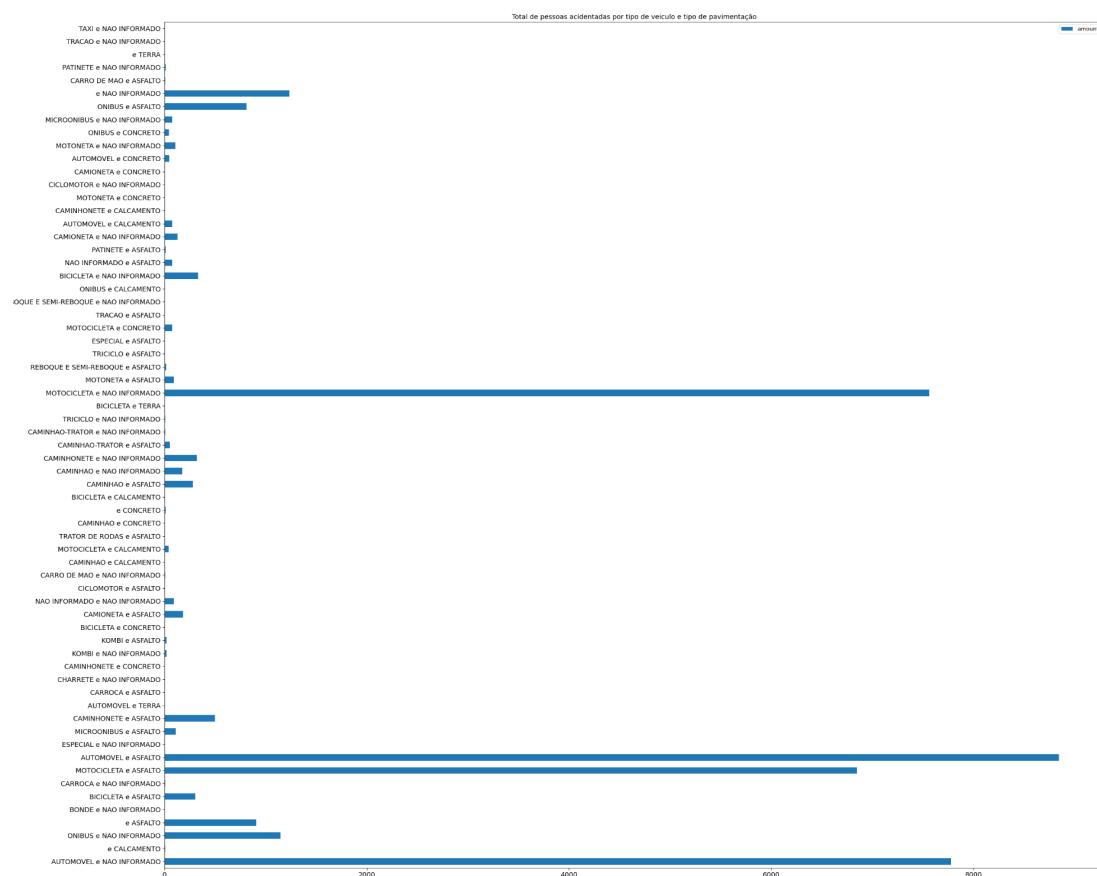
plt.clf()
plt.close()
dfpandas.plot.barh(x="description", figsize=(30, 30), fontsize=12, title='Total de pessoas acidentadas por tipo de veiculo e tipo de pavimentação')
plt.savefig('output/output.png')
```



3. Total de pessoas acidentadas por tipo de veículo e tipo de pavimentação;

Usando o ambiente Vagrant para realizar a análise de dados:

```
from pyspark.sql import SparkSession
from pyspark.sql import Row
import matplotlib.pyplot as plt
spark = SparkSession.builder.appName("Total de acidentes por tipo de pavimento e condição do tempo;").enableHiveSupport().getOrCreate()
spark.sql("use gpdb")
df = spark.sql(" SELECT CONCAT_WS(' e \\", trim(bol.DESC_TIPO_ACIDENTE), trim(bol.DESC_TEMPO)) as description, COUNT(bol.`PAVIMENTO`) as Pavimento FROM `si_bol` bol GROUP BY bol.`DESC_TIPO_ACIDENTE`, bol.`DESC_TEMPO`")
dfpandas=df.toPandas()
plt.clf()
plt.close()
dfpandas.plot.barh(x="description", figsize=(30, 30), fontsize=12, title='Total de acidentes por tipo de pavimento e condição do tempo')
plt.savefig('output/output.png')
```



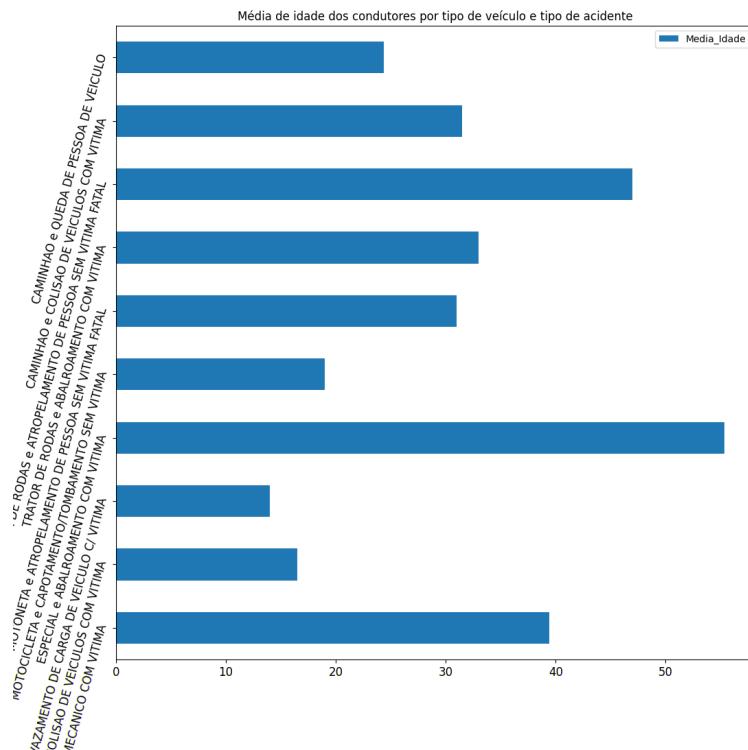
4. Média de idade dos condutores por tipo de veículo e tipo de acidente;

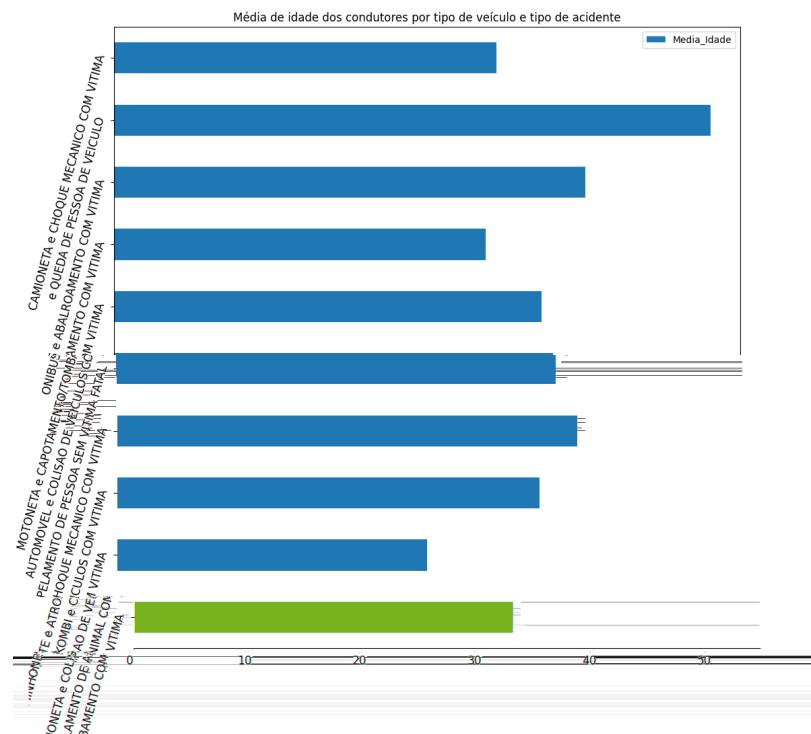
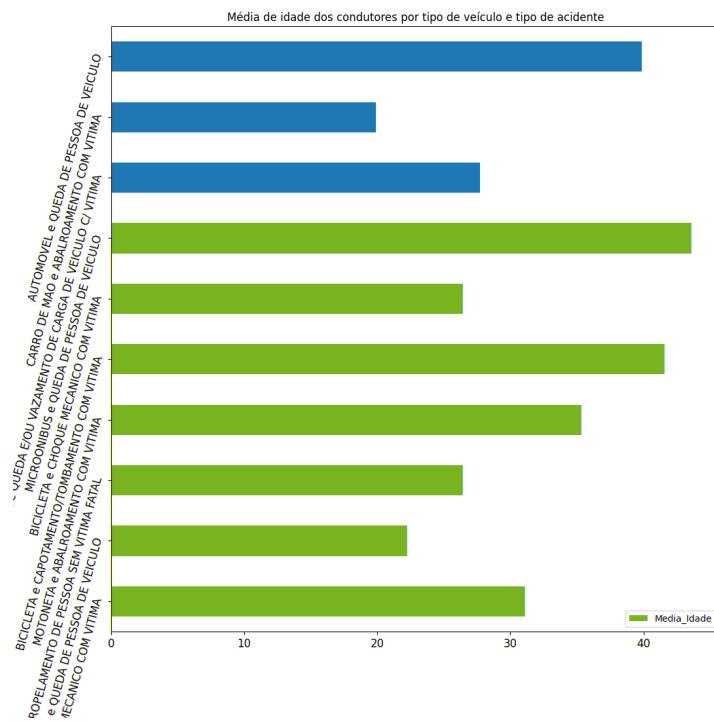
Usando o ambiente Vagrant para realizar a análise de dados:

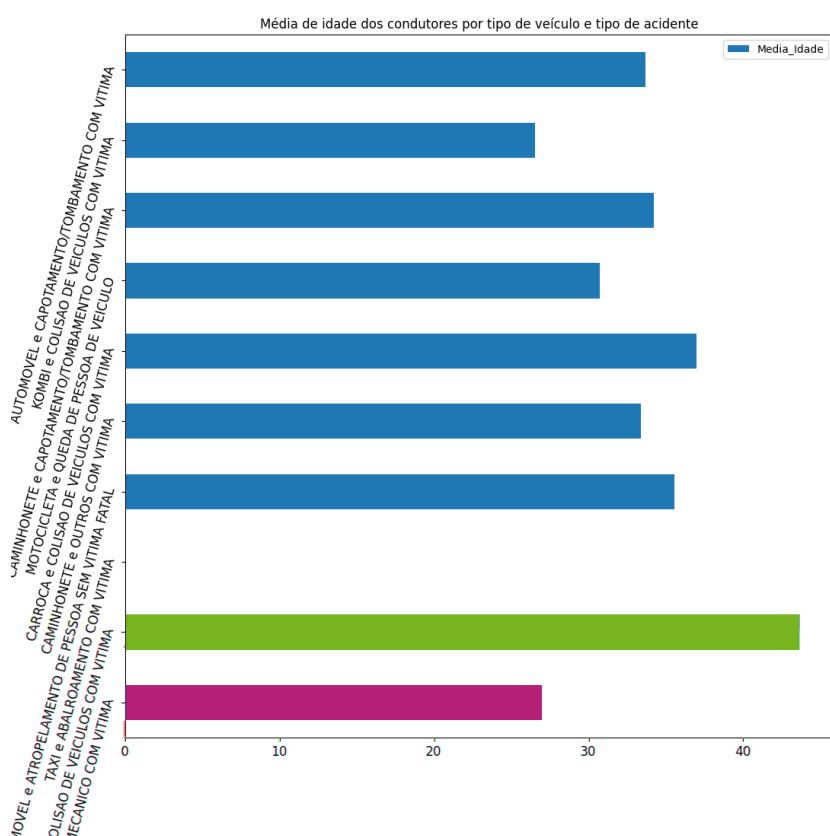
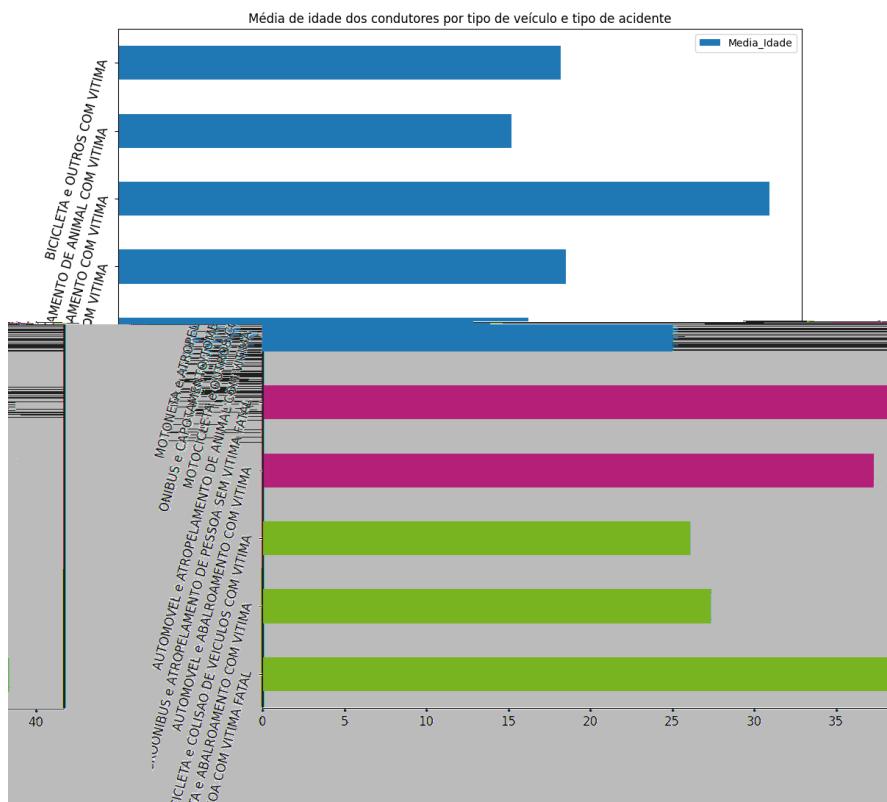
```
from pyspark.sql import SparkSession
from pyspark.sql import Row
import matplotlib.pyplot as plt
spark = SparkSession.builder.appName("Média de idade dos condutores por tipo de veículo e tipo de acidente").enableHiveSupport().getOrCreate()
spark.sql("use gpdb")
df = spark.sql("SELECT CONCAT_WS(' e \\", trim(env.especie_veiculo), trim(bol.DESC_TIPO_ACIDENTE)) as
Tipo_Accidente, AVG(env.`Idade`)AS Media_Idade FROM si_log log JOIN si_bol bol ON
bol.`NUMERO_BOLETIM` = log.`Nº_boletim` JOIN si_env env ON env.`num_boletim` = log.`Nº_boletim` GROUP
BY env.especie_veiculo, bol.DESC_TIPO_ACIDENTE")
dfpandas=df.toPandas()

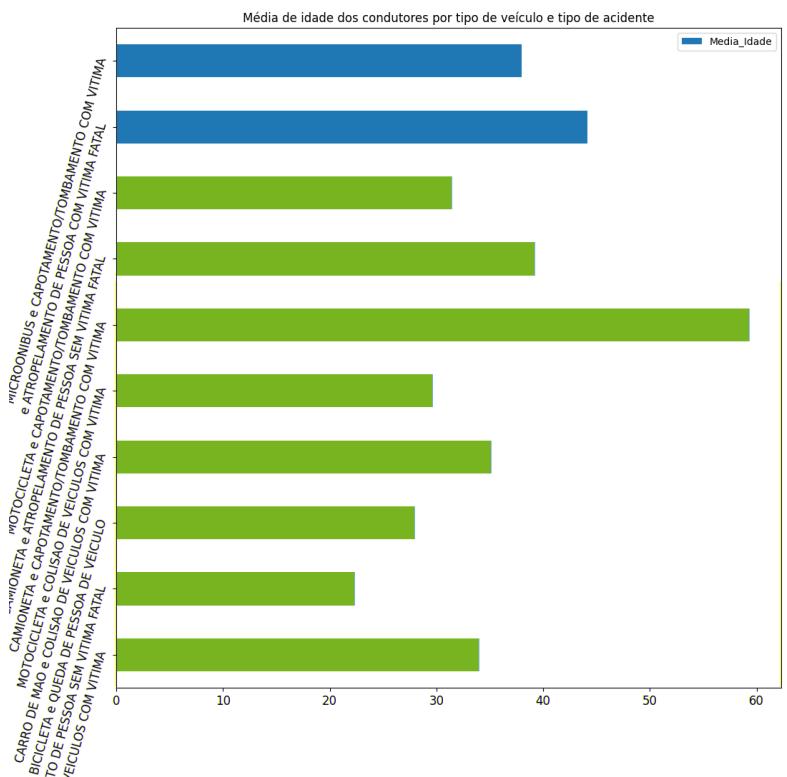
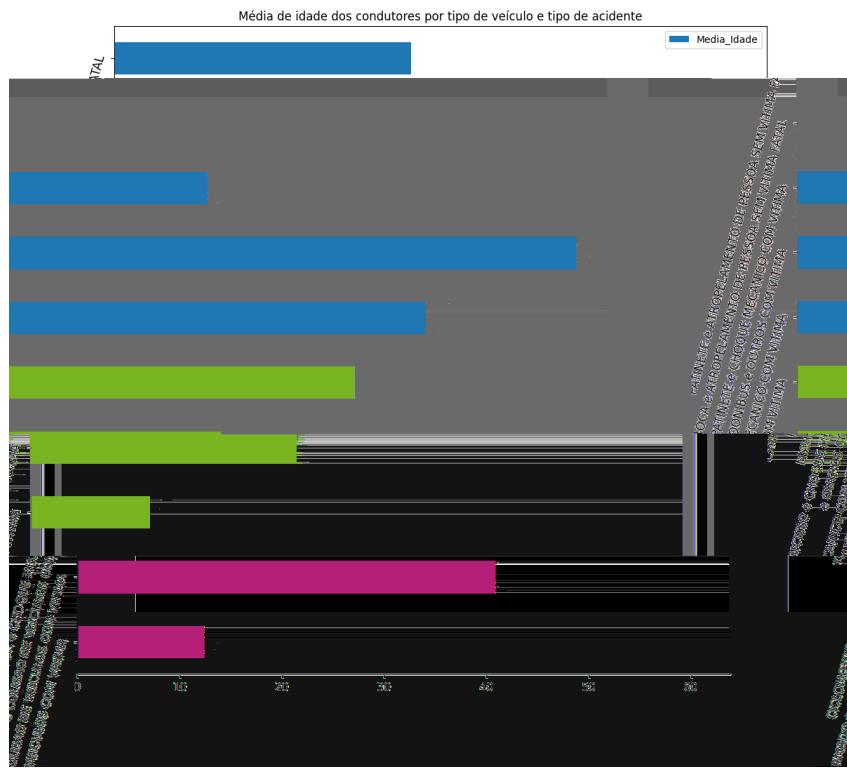
size = 10
df_dict = {n: dfpandas.iloc[n:n+size, :] for n in range(0, len(dfpandas), size)}

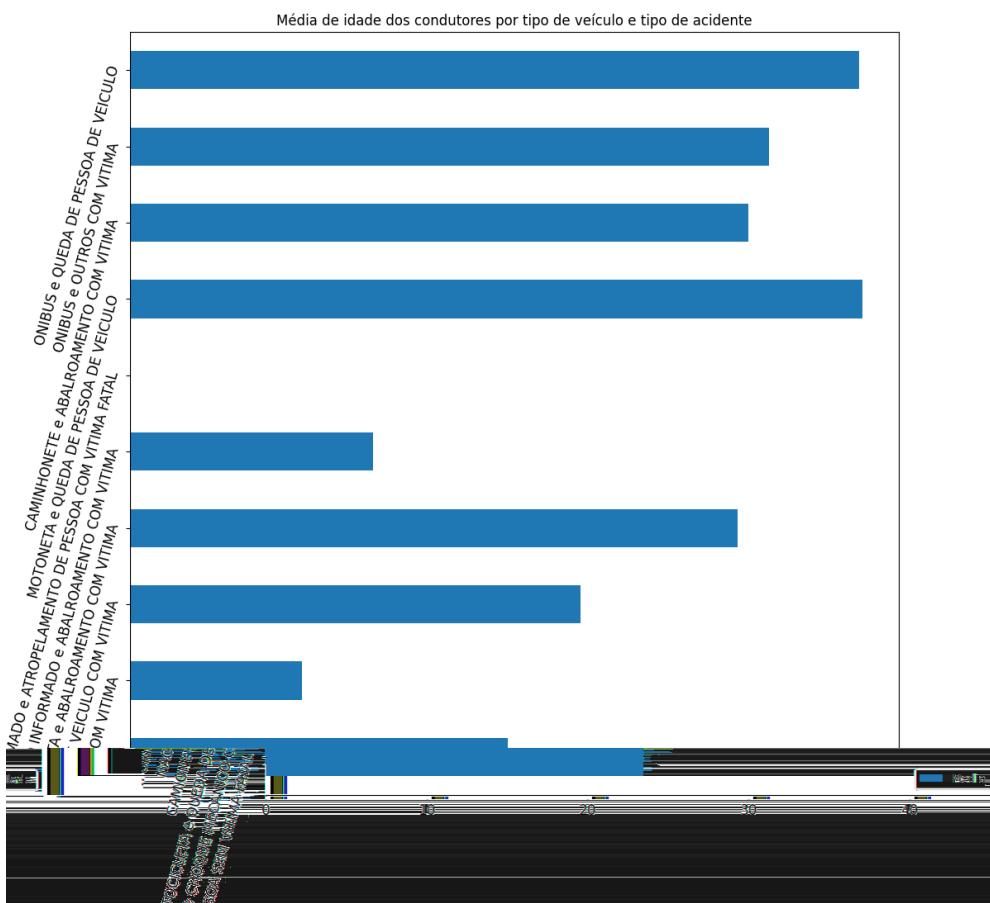
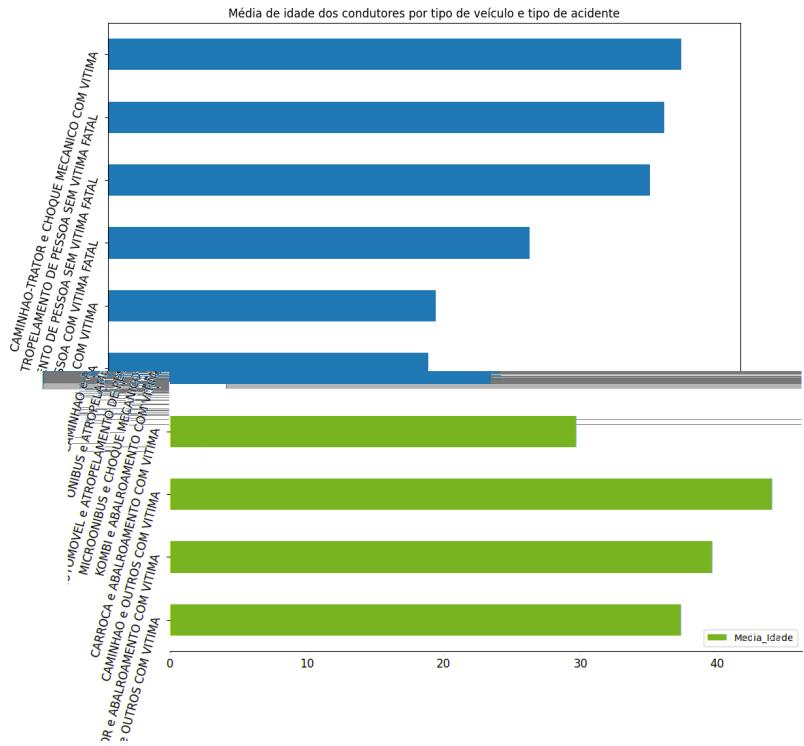
for dabla in df_dict:
    plt.clf()
    plt.close()
    df_dict[dabla].plot.barh(y='Media_Idade', x='Tipo_Accidente', rot=75, figsize=(12, 12), fontsize=12, title='Média
de idade dos condutores por tipo de veículo e tipo de acidente', xlabel='Média')
    plt.savefig('output/'+str(dabla)+'.png')
```

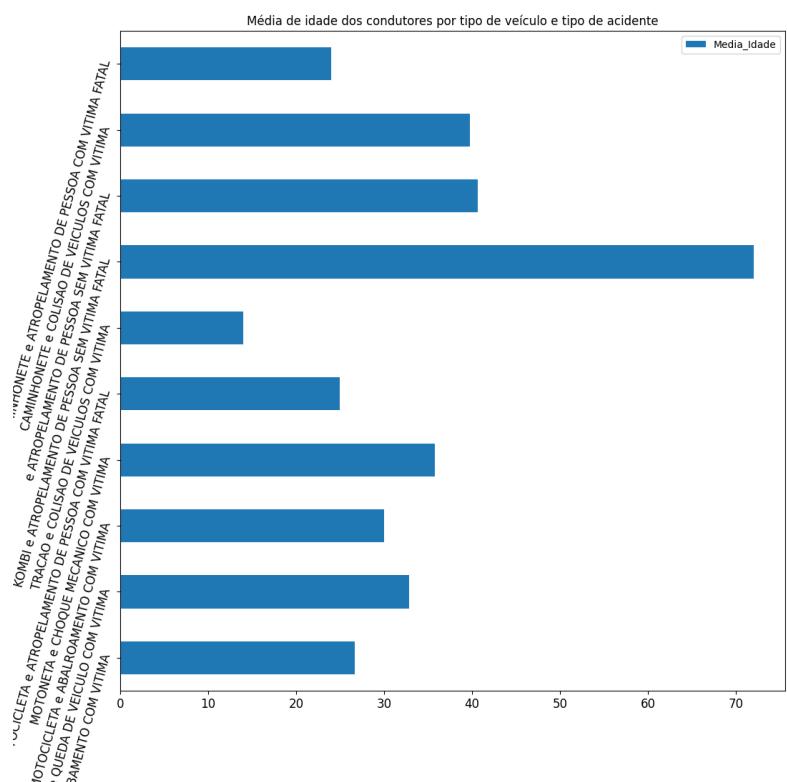
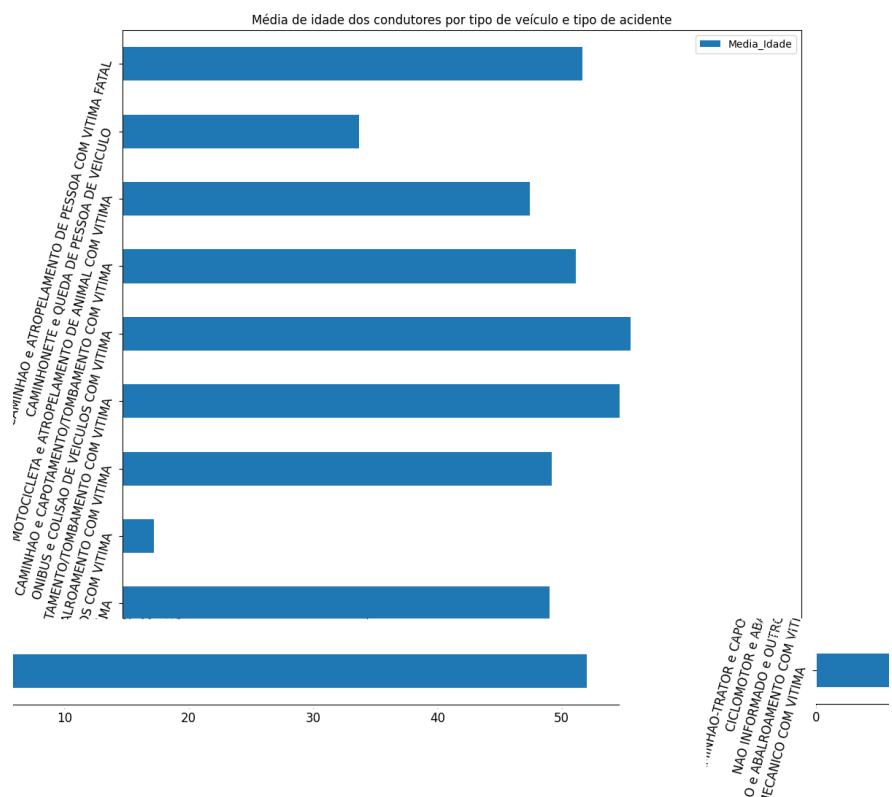


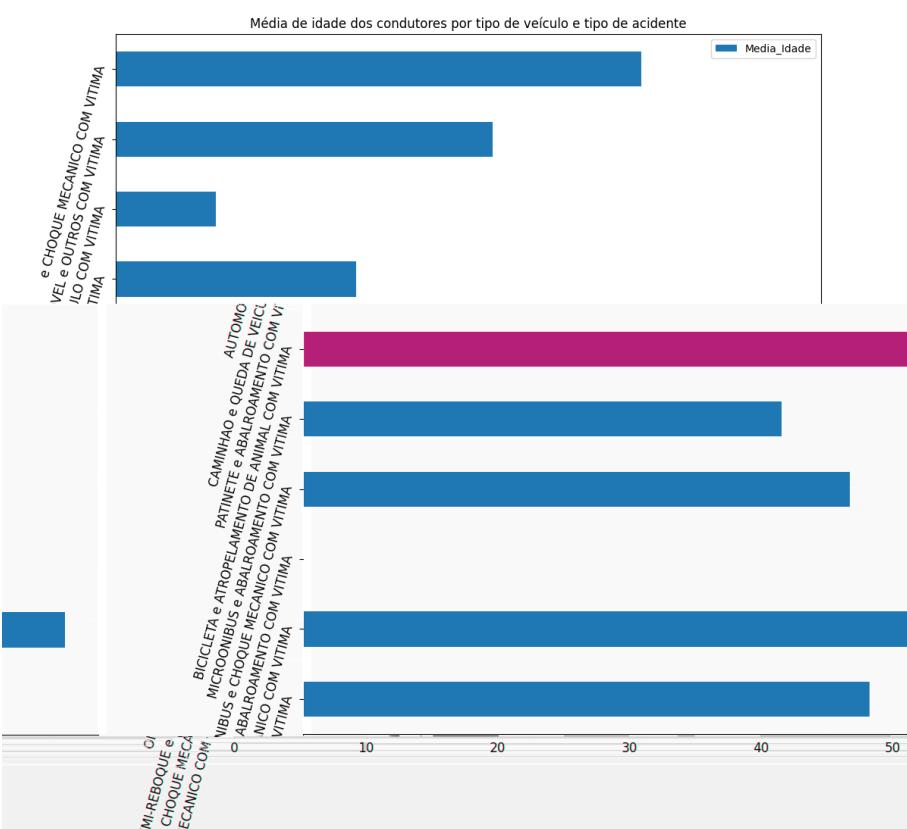
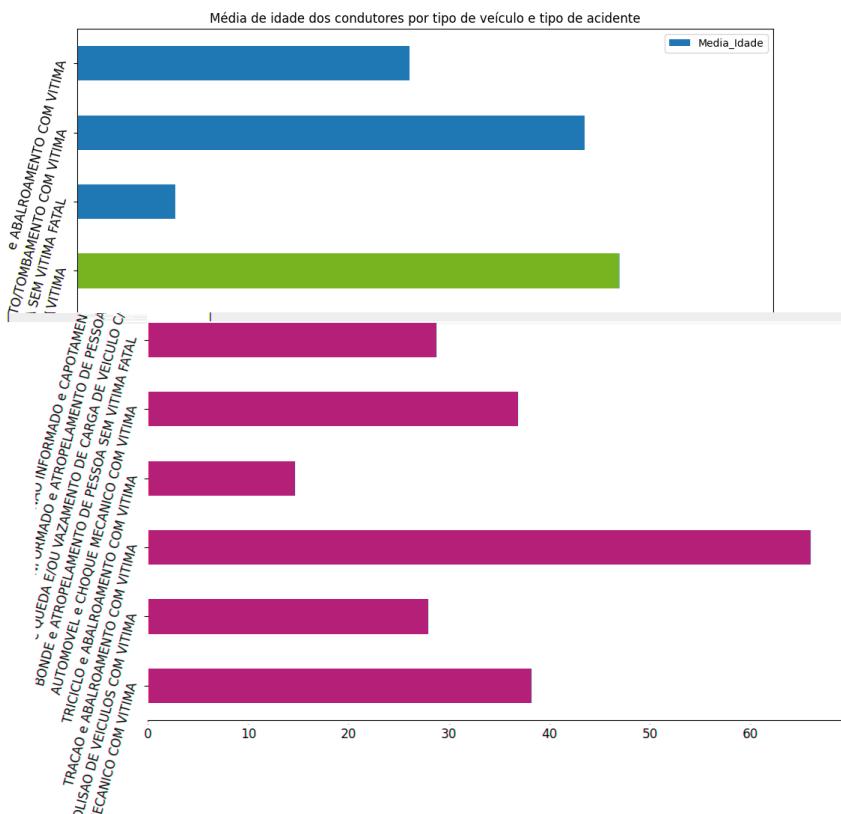












5. Média de idade dos condutores por indicativo de embriaguez;

Usando o ambiente Pandas para realizar a análise de dados:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

#criando o DF
df = pd.read_csv('si_env-2019.csv', encoding='latin1', error_bad_lines=False, sep=';').rename(columns={"Idade": "Idade"}).rename(columns={"Embreagues": "Embreagues"})
#criando um novo df somente com os embreagados positivo
embreag = df.Embreagues.eq("SIM")
emb_df = (df[embreag])
#criando um novo dicionario com a idade e quantidade
newdf = emb_df[emb_df.Idade >
0]["Idade"].value_counts().rename_axis("Idade").reset_index(name='Quantidade').sort_values(by=['Quantidade'])
#obtendo a media de idade
media_idade = newdf['Idade'].median()
#configuracoes da figura de grafico
plt.figure(figsize=(16,9))
plt.xlabel("Idade", fontsize = 18)
plt.ylabel("Quantidade", fontsize = 18)
plt.yticks(np.arange(0, 20, 2), fontsize=15)
plt.xticks(np.arange(0, 100, 5), fontsize=15)
#texto que aparece na parte direita superior do grafico
plt.text(55, 16, "Media de idade dos embriagados: %s"%media_idade, fontsize=15, bbox={'facecolor': 'none',
'alpha': 1, 'pad': 10})
plt.bar(newdf.Idade , newdf.Quantidade)
```

