Primeiros passos: 1 - Imports das bibliotecas 2 - Input do diretório do arquivo a ser analisado 3 - Extrações das informações das colunas para criar a tabela.

In [179…

```
import pandas as pd
import numpy as np
```

In [180…

```
!pip install psycopg2
```

```
Requirement already satisfied: psycopg2 in c:\projetos\dataglowup\lib\site-packages (2.9.9)
```

In [181…

```
import psycopg2
from sqlalchemy import create_engine
```

In [182…

```
caminho_csv = r'C:\projetos\DataGlowUp\Datasets'
```

In [183…

```
df_extract = pd.read_csv(caminho_csv + '\\nyc_collisions.csv', sep=',', thousands='.', decimal=',')
```

In [184…

```
# algumas conversões que foi identificado logo no incio.
df_extract['Date'] = pd.to_datetime(df_extract['Date'])
df_extract['Time'] = pd.to_datetime(df_extract['Time'], format='%H:%M:%S', errors='coerce')
df_extract['Time'] = df_extract['Time'].dt.time
```

In [185…

```
df_extract.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 238421 entries, 0 to 238420
Data columns (total 18 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   Collision ID         238421 non-null  int64
 1   Date                 238421 non-null  datetime64[ns]
 2   Time                 238421 non-null  object
 3   Borough              231224 non-null  object
 4   Street Name          238058 non-null  object
 5   Cross Street         111291 non-null  object
 6   Latitude             216098 non-null  float64
 7   Longitude            216098 non-null  float64
 8   Contributing Factor  237134 non-null  object
 9   Vehicle Type         238421 non-null  object
 10  Persons Injured      238420 non-null  float64
 11  Persons Killed       238421 non-null  int64
 12  Pedestrians Injured  238421 non-null  int64
 13  Pedestrians Killed   238421 non-null  int64
 14  Cyclists Injured     238421 non-null  int64
 15  Cyclists Killed      238421 non-null  int64
 16  Motorists Injured    238421 non-null  int64
 17  Motorists Killed     238421 non-null  int64
dtypes: datetime64[ns](1), float64(3), int64(8), object(6)
memory usage: 32.7+ MB
```

- Criar a conexão com o banco de dados;
- Mapear os dados das colunas;
-
- Criação da tabela baseada nos tipos de dados que estão nas colunas

In [186…

```
# Configurações com a base de dados
db_config = {
    'dbname': 'dataglowup',
    'user': 'postgres',
    'password': 'ohdelta',
    'host': 'localhost',
    'port': '5432'
}

dsn = f"dbname={db_config['dbname']} user={db_config['user']} password={db_config['password']} host={db_config['host']} port={db_config['port']}"
```

In [187…

```
conn = psycopg2.connect(dsn)
```

In [188…

```
#renomear as colunas do dataframe para criação da tabela
df_extract = df_extract.rename(columns={
    'Collision ID': 'id_collision',
    'Date': 'dt_collision',
    'Time': 'time_collision',
    'Borough': 'borough',
    'Street Name': 'nm_street',
    'Cross Street': 'cr_street',
    'Latitude': 'latitude',
    'Longitude': 'longitude',
    'Contributing Factor': 'contr_factor',
    'Vehicle Type': 'vehicle_type',
    'Persons Injured': 'pers_injured',
    'Persons Killed': 'pers_killed',
    'Pedestrians Injured': 'ped_injured',
    'Pedestrians Killed': 'ped_killed',
    'Cyclists Injured': 'cyclists_injured',
    'Cyclists Killed': 'cyclists_killed',
    'Motorists Injured': 'motorists_injured',
    'Motorists Killed': 'motorists_killed'
})
```

In [189…

```
#Mapeamento das colunas
mapeamento_coluna ={
    'id_collision': 'BIGINT',
    'dt_collision': 'TIMESTAMP',
    'time_collision': 'TIMESTAMP',
    'borough': 'TEXT',
    'nm_street': 'TEXT',
    'cr_street': 'TEXT',
    'latitude': 'FLOAT',
    'longitude': 'FLOAT',
    'contr_factor': 'TEXT',
    'vehicle_type': 'TEXT',
    'pers_injured': 'BIGINT',
    'pers_killed': 'BIGINT',
    'ped_injured': 'BIGINT',
    'ped_killed': 'BIGINT',
    'cyclists_injured': 'BIGINT',
    'cyclists_killed': 'BIGINT',
    'motorists_injured': 'BIGINT',
    'motorists_killed': 'BIGINT'
}
```

In [191…

```
nome_tabela = 'nyc_collisions'
```

In [192…

```
#criando a partir do dataset uma base de dados (somente os nomes das colunas, sem inserir dados
df_extract.head(0).to_sql(nome_tabela, engine, if_exists='replace', index=False)
```

Out[192…

```
0
```

In [193…

```
# Confirmar que a tabela foi criada
consulta = f"SELECT * FROM {nome_tabela} LIMIT 5;"
df_resultado = pd.read_sql_query(consulta, engine)
print(df_resultado)
```

```
Empty DataFrame
Columns: [id_collision, dt_collision, time_collision, borough, nm_street, cr_street, latitude, longitude, contr_factor, vehicle_type, pers_injured, pers_killed, ped_injured, ped_killed, cyclists_injured, c
yclists_killed, motorists_injured, motorists_killed]
Index: []
```

TRATAMENTO DOS DADOS

- Conhecendo a base
- Mapeando as informações
- Indentificando os outliers
- Tratando os dados
- inserindo as no banco de dados criado.

In [194…

```
df_extract.shape
```

Out[194…

```
(238421, 18)
```

In [195…

```
## Verificando os dados e como eles estão no dataset
display(df_extract)
```

	id_collision	dt_collision	time_collision	borough	nm_street	cr_street	latitude	longitude	contr_factor	vehicle_type	pers_injured	pers_killed	ped_injured	ped_killed	cyclists_injured	cyclists_killed
0	4491746	2021-01-01	20:00:00	Bronx	Bruckner Expressway	NaN	4083398.0	-7382635.0	Pavement Slippery	Passenger Vehicle	0.0	0	0	0	0	0
1	4441905	2021-01-01	05:28:00	Brooklyn	Lafayette Avenue	NaN	406873.0	-73973656.0	Unspecified	Passenger Vehicle	0.0	0	0	0	0	0
2	4382769	2021-01-01	06:00:00	Staten Island	West Shore Expressway	NaN	NaN	NaN	Fell Asleep	Passenger Vehicle	0.0	0	0	0	0	0
3	4380949	2021-01-01	19:30:00	Bronx	Sedgwick Avenue	Vancortlandt Avenue West	408827.0	-7389273.0	NaN	Not Reported	0.0	0	0	0	0	0
4	4380940	2021-01-01	07:40:00	Brooklyn	Cortelyou Road	Mc Donald Avenue	4063791.0	-7397864.0	Unspecified	Passenger Vehicle	0.0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
238416	4619581	2023-04-09	04:21:00	Brooklyn	Meeker Avenue	NaN	40715443.0	-7395185.0	Failure to Yield Right-of-Way	Not Reported	1.0	0	1	0	0	0
238417	4619685	2023-04-09	08:30:00	Queens	Elbertson Street	Elmhurst Avenue	40746864.0	-7387717.0	Backing Unsafely	Passenger Vehicle	1.0	0	1	0	0	0
238418	4619519	2023-04-09	21:19:00	Brooklyn	Cortelyou Road	East 17 Street	40642017.0	-7396266.0	Unspecified	Passenger Vehicle	1.0	0	1	0	0	0
238419	4619921	2023-04-09	11:00:00	Manhattan	West 50 Street	NaN	4076379.0	-73989655.0	Driver Inattention/Distraction	Transport	0.0	0	0	0	0	0
238420	4619618	2023-04-09	19:10:00	Bronx	Watson Avenue	Manor Avenue	408264.0	-7387581.0	Driver Inattention/Distraction	Passenger Vehicle	0.0	0	0	0	0	0

238421 rows × 18 columns

In [196…

```
#identificando os valores nulos na base
#nessa etapa eu ainda não decidi se os valores nulos devem ou não serem excluídos
df_extract.isnull().sum()
```

Out[196…

```
id_collision              0
dt_collision              0
time_collision            0
borough                7197
nm_street               363
cr_street            127130
latitude              22323
longitude             22323
contr_factor           1287
vehicle_type              0
pers_injured              1
pers_killed               0
ped_injured               0
ped_killed                0
cyclists_injured          0
cyclists_killed           0
motorists_injured         0
motorists_killed          0
dtype: int64
```

In [197…

```
## como a base tem dia, eu vou ver o período que irei analisar

dt_inicio = pd.to_datetime(df_extract['dt_collision']).dt.date.min()
print(dt_inicio)
```

```
2021-01-01
```

In [198…

```
dt_inicio = pd.to_datetime(df_extract['dt_collision']).dt.date.max()
print(dt_inicio)
```

```
2023-04-09
```

In [199…

```
colunas_plotar = df_extract.columns.drop(['pers_injured', 'pers_killed', 'ped_injured', 'ped_killed', 'cyclists_injured', 'cyclists_killed', 'motorists_injured', 'motorists_killed'])
```

In [200…

```
# Lista das colunas desejadas
colunas_desejadas = ['pers_injured', 'pers_killed', 'ped_injured', 'ped_killed', 'cyclists_injured', 'cyclists_killed', 'motorists_injured', 'motorists_killed']

# Criar um novo DataFrame com base nas colunas desejadas
df_resumo = df_extract[colunas_desejadas].copy()
```

In [201…

```
df_resumo.describe() ## Descrições dos dados somente das variaveis.
```

Out[201…

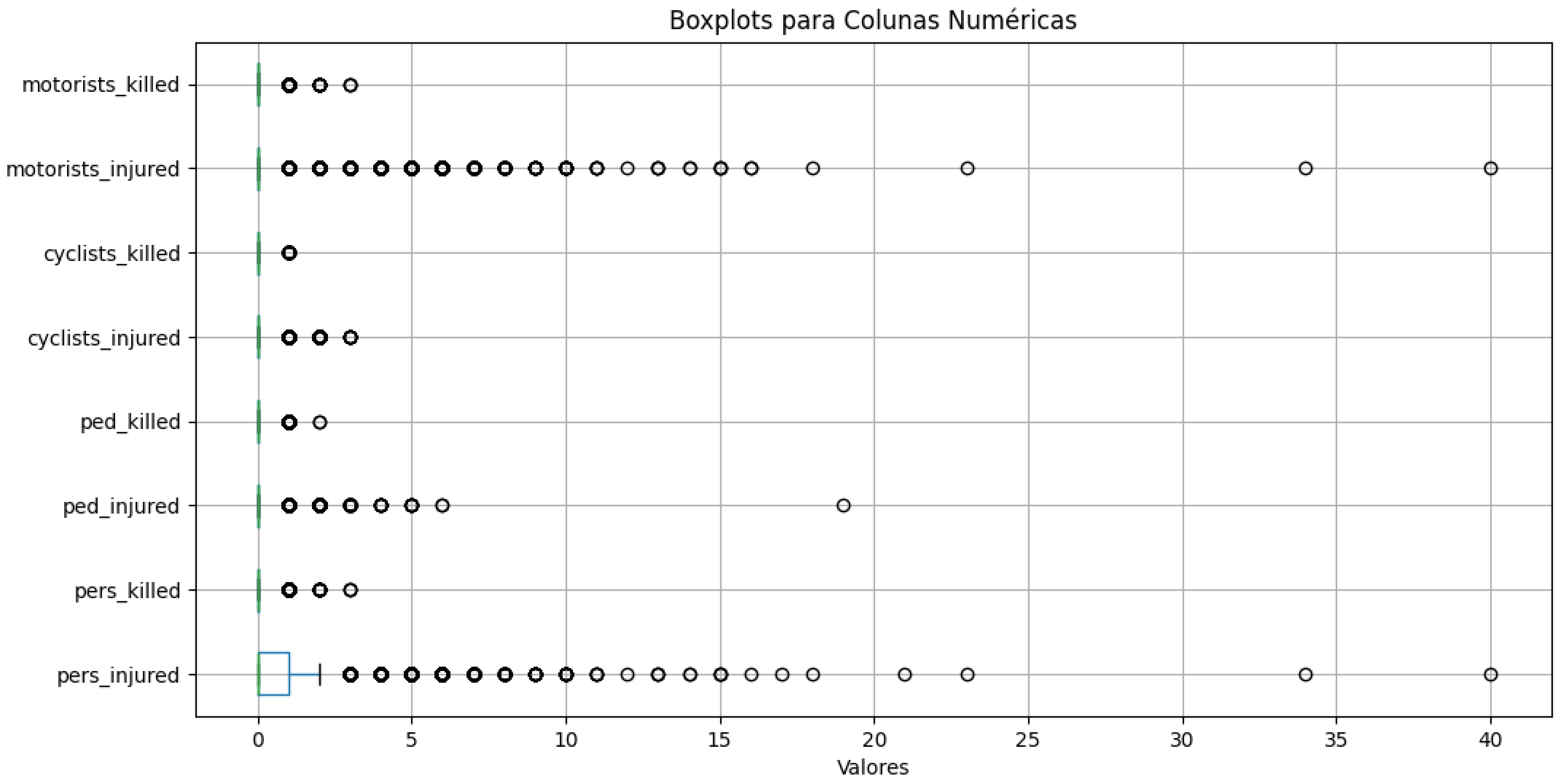
	pers_injured	pers_killed	ped_injured	ped_killed	cyclists_injured	cyclists_killed	motorists_injured	motorists_killed
count	238420.000000	238421.000000	238421.000000	238421.000000	238421.000000	238421.000000	238421.00000	238421.000000
mean	0.487484	0.002663	0.079175	0.001200	0.045852	0.000197	0.34153	0.001124
std	0.806650	0.053535	0.285696	0.034855	0.212723	0.014039	0.77766	0.036041
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000
75%	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000
max	40.000000	3.000000	19.000000	2.000000	3.000000	1.000000	40.00000	3.000000

In [202…

```
import matplotlib.pyplot as plt

# Selecionar as colunas desejadas
colunas_plotar = ['pers_injured', 'pers_killed', 'ped_injured', 'ped_killed', 'cyclists_injured', 'cyclists_killed', 'motorists_injured', 'motorists_killed']

# Criar boxplot para colunas numéricas
df_extract[colunas_plotar].boxplot(vert=False, figsize=(12, 6))
plt.title('Boxplots para Colunas Numéricas')
plt.xlabel('Valores')
plt.show()
```



In [204…
```
df_erro = df_extract[df_extract['ped_injured']==0]
```

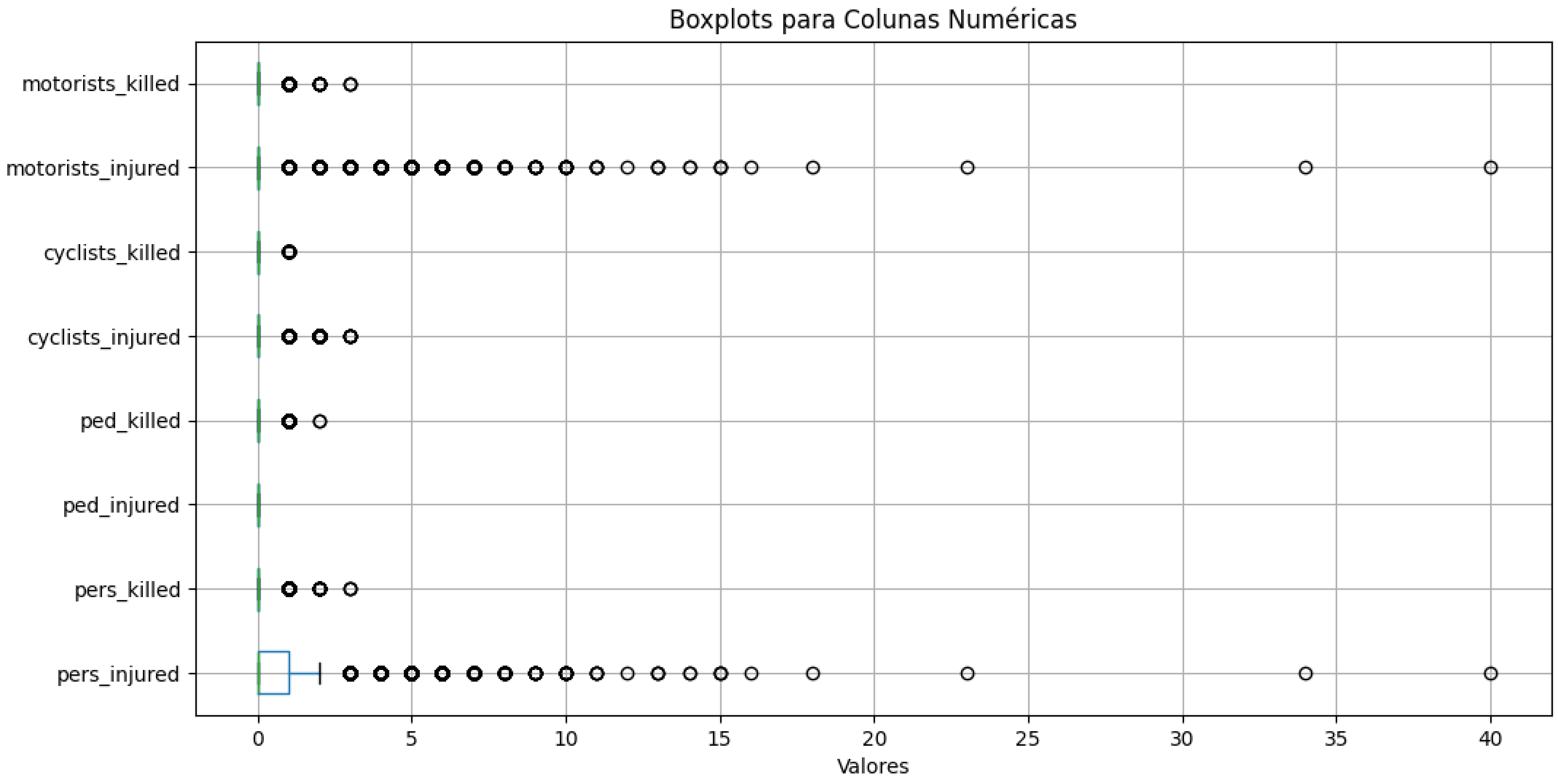
In [205…
```
df_erro.describe()
```

Out[205…

	id_collision	dt_collision	latitude	longitude	pers_injured	pers_killed	ped_injured	ped_killed	cyclists_injured	cyclists_killed	motorists_injured	motorists_killed
count	2.203020e+05	220302	1.991580e+05	1.991580e+05	220301.000000	220302.000000	220302.0	220302.000000	220302.000000	220302.000000	220302.000000	220302.000000
mean	4.499604e+06	2022-02-06 05:30:32.992891392	3.402654e+07	-3.683620e+07	0.439263	0.002764	0.0	0.001194	0.049182	0.000213	0.367682	0.001203
min	4.073803e+06	2021-01-01 00:00:00	4.070000e+02	-7.425003e+08	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000
25%	4.439845e+06	2021-07-19 00:00:00	4.074646e+06	-7.373444e+07	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000
50%	4.499292e+06	2022-01-30 00:00:00	4.065717e+07	-7.394218e+06	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000
75%	4.558890e+06	2022-08-23 00:00:00	4.074539e+07	-7.387058e+06	1.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000
max	4.619988e+06	2023-04-09 00:00:00	4.091020e+08	-7.400000e+01	40.000000	3.000000	0.0	2.000000	3.000000	1.000000	40.000000	3.000000
std	6.883890e+04	NaN	5.573236e+07	1.036603e+08	0.812997	0.054623	0.0	0.034793	0.219913	0.014605	0.800521	0.037311

In [206…
```
# Selecionar as colunas desejadas
colunas_plotar = ['pers_injured', 'pers_killed', 'ped_injured', 'ped_killed', 'cyclists_injured', 'cyclists_killed', 'motorists_injured', 'motorists_killed']

# Criar boxplot para colunas numéricas
df_erro[colunas_plotar].boxplot(vert=False, figsize=(12, 6))
plt.title('Boxplots para Colunas Numéricas')
plt.xlabel('Valores')
plt.show()
```



In [207…
```
df_extract[df_extract['motorists_injured']>20]
```

Out[207…

	id_collision	dt_collision	time_collision	borough	nm_street	cr_street	latitude	longitude	contr_factor	vehicle_type	pers_injured	pers_killed	ped_injured	ped_killed	cyclists_injured	cyclists_killed	moto
168087	4552186	2022-07-21	05:56:00	Bronx	Hutchinson River Parkway Ramp	NaN	NaN	NaN	Unsafe Speed	Bus	40.0	0	0	0	0	0	
233053	4616707	2023-03-17	21:22:00	Queens	220 Street	Jamaica Avenue	4071893.0	-7373439.0	Driver Inattention/Distraction	Passenger Vehicle	23.0	0	0	0	0	0	
237980	4619206	2023-04-07	22:24:00	Queens	Bell Boulevard	45 Road	4075958.0	-7376894.0	Traffic Control Disregarded	Bus	34.0	0	0	0	0	0	

In [213…
```
df_extract['pers_injured'].fillna(0, inplace=True)
df_extract['pers_injured'] = df_extract['ped_injured'].astype('int64')
```

In [214…
```
df_extract.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 238421 entries, 0 to 238420
Data columns (total 18 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   id_collision       238421 non-null  int64
 1   dt_collision       238421 non-null  datetime64[ns]
 2   time_collision     238421 non-null  object
 3   borough            231224 non-null  object
 4   nm_street          238058 non-null  object
 5   cr_street          111291 non-null  object
 6   latitude           216098 non-null  float64
 7   longitude          216098 non-null  float64
 8   contr_factor       237134 non-null  object
 9   vehicle_type       238421 non-null  object
 10  pers_injured       238421 non-null  int64
 11  pers_killed        238421 non-null  int64
 12  ped_injured        238421 non-null  int64
 13  ped_killed         238421 non-null  int64
 14  cyclists_injured   238421 non-null  int64
 15  cyclists_killed    238421 non-null  int64
 16  motorists_injured  238421 non-null  int64
 17  motorists_killed   238421 non-null  int64
dtypes: datetime64[ns](1), float64(2), int64(9), object(6)
memory usage: 32.7+ MB
```

In [215…

```
df_extract[df_extract['borough'].isnull()]
```

Out[215…

	id_collision	dt_collision	time_collision	borough	nm_street	cr_street	latitude	longitude	contr_factor	vehicle_type	pers_injured	pers_killed	ped_injured	ped_killed	cyclists_injured	cyclists_killed	motor
35	4383824	2021-01-01	01:28:00	NaN	Major Deegan Expressway Ramp	NaN	NaN	NaN	Passing or Lane Usage Improper	Passenger Vehicle	0	0	0	0	0	0	
116	4381008	2021-01-01	16:50:00	NaN	Brooklyn Bridge	NaN	NaN	NaN	Driver Inattention/Distraction	Passenger Vehicle	0	0	0	0	0	0	
162	4380792	2021-01-01	05:12:00	NaN	Triborough Bridge	NaN	NaN	NaN	Alcohol Involvement	Passenger Vehicle	0	0	0	0	0	0	
188	4380898	2021-01-01	09:25:00	NaN	Belt Parkway	NaN	NaN	NaN	Unsafe Speed	Passenger Vehicle	0	0	0	0	0	0	
256	4382133	2021-01-01	22:01:00	NaN	Bronx Whitestone Bridge	NaN	NaN	NaN	Driver Inexperience	Passenger Vehicle	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
238307	4619198	2023-04-09	02:50:00	NaN	Brooklyn Bridge	NaN	NaN	NaN	Reaction to Uninvolved Vehicle	Passenger Vehicle	0	0	0	0	0	0	
238330	4619806	2023-04-09	14:25:00	NaN	Bronx Whitestone Bridge	NaN	NaN	NaN	Following Too Closely	Passenger Vehicle	0	0	0	0	0	0	
238331	4619290	2023-04-09	05:45:00	NaN	Van Wyck Service Road	Jamaica Avenue	NaN	NaN	Turning Improperly	Passenger Vehicle	0	0	0	0	0	0	
238342	4619389	2023-04-09	10:28:00	NaN	31 Street	Astoria Boulevard	NaN	NaN	Turning Improperly	Passenger Vehicle	0	0	0	0	0	0	
238347	4619326	2023-04-09	11:30:00	NaN	Hillside Avenue	Cross Island Parkway	NaN	NaN	Unspecified	Passenger Vehicle	0	0	0	0	0	0	

7197 rows × 18 columns

In [216…

```
filtro = (
    ~df_extract['borough'].isna() &
    (
        (df_extract['pers_injured'] != 0) |
        (df_extract['pers_killed'] != 0) |
        (df_extract['ped_injured'] != 0) |
        (df_extract['ped_killed'] != 0) |
        (df_extract['cyclists_injured'] != 0) |
        (df_extract['cyclists_killed'] != 0) |
        (df_extract['motorists_injured'] != 0) |
        (df_extract['motorists_killed'] != 0)
    )
)

df_transformado = df_extract[filtro]
```

In [217…

```
df_transformado.head()
```

Out[217…

	id_collision	dt_collision	time_collision	borough	nm_street	cr_street	latitude	longitude	contr_factor	vehicle_type	pers_injured	pers_killed	ped_injured	ped_killed	cyclists_injured	cyclists_killed	motorists_injure
14	4381374	2021-01-01	17:25:00	Manhattan	Central Park South	NaN	40766277.0	-7397769.0	Pavement Slippery	Passenger Vehicle	0	0	0	0	0	0	
16	4380801	2021-01-01	05:10:00	Staten Island	Bement Avenue	NaN	40628098.0	-7411104.0	Unspecified	Passenger Vehicle	0	0	0	0	0	0	
21	4380882	2021-01-01	07:50:00	Bronx	Major Deegan Expressway	NaN	40811638.0	-739316.0	Alcohol Involvement	Passenger Vehicle	0	0	0	0	0	0	
22	4380843	2021-01-01	20:05:00	Manhattan	West 25 Street	NaN	40748974.0	-7400324.0	Accelerator Defective	Passenger Vehicle	0	0	0	0	0	0	
23	4382872	2021-01-01	02:45:00	Manhattan	Broadway	W 116 Street	NaN	NaN	Following Too Closely	Taxi	0	0	0	0	0	0	

In [218…

```
df_transformado.isnull().sum()
```

Out[218…

```
id_collision             0
dt_collision             0
time_collision           0
borough                  0
nm_street               94
cr_street            35772
latitude              5536
longitude             5536
contr_factor          1016
vehicle_type             0
pers_injured             0
pers_killed              0
ped_injured              0
ped_killed               0
cyclists_injured         0
cyclists_killed          0
motorists_injured        0
motorists_killed         0
dtype: int64
```

In [219…

```
conn = psycopg2.connect(dsn)
dsn = f"dbname={db_config['dbname']} user={db_config['user']} password={db_config['password']} host={db_config['host']} port={db_config['port']}"
```

In [220…

```
nome_tabela = 'nyc_collisions'
```

In [221…

```
df_transformado.to_sql(nome_tabela, engine, if_exists='replace', index=False)
```

Out[221…

```
321
```

In [223…

```
conn.close()
```

In [ ]:

```
```