

# PROJETO 3 ANÁLISE DE RISCO NO TRANSPORTE PÚBLICO

ELABORADO POR DANILO TEMERLOGLOU DE ABREU

CONCLUÍDO EM 05/12/2023

```
In [1]: #Cada problema foi resolvido de duas formas: query SQL e gráfico
```

```
In [2]: # Versão da Linguagem Python
from platform import python_version
print('Versão da Linguagem Python Usada Neste Jupyter Notebook:', python_version())
```

Versão da Linguagem Python Usada Neste Jupyter Notebook: 3.9.13

```
In [3]: #pip install sqldf -U
```

```
In [4]: #pip install numpy -U
```

```
In [5]: # Imports
import numpy as np
import pandas as pd
import sqldf
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
In [6]: # Versões dos pacotes usados neste jupyter notebook
%reload_ext watermark
%watermark -a "Danilo Temerloglou de Abreu" --iversions
```

Author: Danilo Temerloglou de Abreu

matplotlib: 3.7.1  
seaborn : 0.12.2  
numpy : 1.23.5  
pandas : 1.3.4  
sqldf : 0.4.2

```
In [7]: # Parâmetros de configuração dos gráficos
from matplotlib import rcParams

rcParams['figure.figsize'] = 12, 4
rcParams['lines.linewidth'] = 3
rcParams['xtick.labelsize'] = 'x-large'
rcParams['ytick.labelsize'] = 'x-large'
```

```
In [8]: # Carrega os dados
df = pd.read_excel('TFL_Bus_Safety.xlsx')
```

In [9]: df.dtypes

Out[9]: Year int64  
Date Of Incident datetime64[ns]  
Route object  
Operator object  
Group Name object  
Bus Garage object  
Borough object  
Injury Result Description object  
Incident Event Type object  
Victim Category object  
Victims Sex object  
Victims Age object  
dtype: object

In [10]: df.sort\_index(inplace = True)  
df.sample(5)

Out[10]:

	Year	Date Of Incident	Route	Operator	Group Name	Bus Garage	Borough	Injury Result Description	Incident Event Type	
18412	2018	2018-01-01	18	London United	London United	Park Royal	Brent	Injuries treated on scene	Personal Injury	P
11033	2016	2016-02-01	105	Metroline	Metroline	Garage Not Available	Ealing	Injuries treated on scene	Onboard Injuries	P
14116	2017	2017-05-01	185	London Central	Go-Ahead	Camberwell	Lambeth	Injuries treated on scene	Slip Trip Fall	P
22466	2018	2018-08-01	259	Arriva London North	Arriva London	Tottenham	Haringey	Injuries treated on scene	Slip Trip Fall	P
20327	2018	2018-04-01	309	C T Plus	CT Plus	Hackney Ash Grove	Tower Hamlets	Injuries treated on scene	Slip Trip Fall	P

In [11]: # Tipo do objeto  
type(df)

Out[11]: pandas.core.frame.DataFrame

In [12]: # Atributo shape  
df.shape

Out[12]: (23158, 12)

In [13]: # Método info()  
df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23158 entries, 0 to 23157
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Year                                23158 non-null  int64
1   Date Of Incident                    23158 non-null  datetime64[ns]
2   Route                              23158 non-null  object
3   Operator                           23158 non-null  object
4   Group Name                         23158 non-null  object
5   Bus Garage                         23158 non-null  object
6   Borough                            23158 non-null  object
7   Injury Result Description           23158 non-null  object
8   Incident Event Type                 23158 non-null  object
9   Victim Category                     23158 non-null  object
10  Victims Sex                         23158 non-null  object
11  Victims Age                         23158 non-null  object
dtypes: datetime64[ns](1), int64(1), object(10)
memory usage: 2.1+ MB
```

```
In [14]: # Método describe()
df.describe()
```

Out[14]:

	Year
count	23158.000000
mean	2016.449607
std	1.074073
min	2015.000000
25%	2016.000000
50%	2016.000000
75%	2017.000000
max	2018.000000

```
In [15]: # Método value_counts()
df.value_counts()
```

Out[15]:

Year	Date Of Incident	Route	Operator	Group Name	Bus Garage
Borough	Injury Result	Description			Incident Event
Type	Victim Category	Victims	Sex	Victims Age	
2016	2016-11-01	52	Metroline	Metroline	Willesden
Westminster	Injuries treated on scene				Collision Incident
ent	Passenger	Unknown	Unknown	13	
2015	2015-06-01	275	East London	Stagecoach	Garage Not Avail
lable	Redbridge	Injuries treated on scene			Co
llision Incident	Passenger	Female	Adult	10	
2016	2016-11-01	211	Abellio London	Abellio	Battersea (Abel
lio London)	Kensington & Chelsea	Injuries treated on scene			Co
llision Incident	Passenger	Female	Adult	7	
2015	2015-08-01	W19	Blue Triangle Buses	Go-Ahead	Garage Not Avail
lable	Newham	Reported Minor Injury - Treated			at Hospital Co
llision Incident	Passenger	Unknown	Unknown	6	
	2015-06-01	43	Metroline	Metroline	Garage Not Avail
lable	Islington	Injuries treated on scene			Co
llision Incident	Passenger	Male	Unknown	6	
..					
2016	2016-06-01	180	London Central	Go-Ahead	Garage Not Avail
lable	Greenwich	Injuries treated on scene			On
board Injuries	Passenger	Female	Adult	1	
		177	Selkent	Stagecoach	Garage Not Avail
lable	Greenwich	Injuries treated on scene			On
board Injuries	Passenger	Unknown	Child	1	
		176	Arriva London South	Arriva London	Garage Not Avail
lable	Lewisham	Injuries treated on scene			On
board Injuries	Passenger	Female	Youth	1	
Collision Incident	Passenger	Female	Child	1	
2018	2018-09-01	W8	Metroline	Metroline	Potters Bar
Enfield	Reported Minor Injury - Treated				at Hospital
Passenger	Female	Youth	1		Slip Trip Fall

Length: 21464, dtype: int64

In [16]:

```
# Atributo size (total de registros considerando cada coluna)
df.size
```

Out[16]:

277896

In [17]:

```
#renomeando as colunas. Elas não podem ter espaço nem acentuação
df.rename(columns={'Year': 'Ano'}, inplace = True)
df.rename(columns={'Date Of Incident': 'Data_Incidente'}, inplace = True)
df.rename(columns={'Route': 'Rota'}, inplace = True)
df.rename(columns={'Operator': 'Operador'}, inplace = True)
df.rename(columns={'Group Name': 'Nome_Grupo'}, inplace = True)
df.rename(columns={'Bus Garage': 'Garagem_Onibus'}, inplace = True)
df.rename(columns={'Borough': 'Bairro'}, inplace = True)
df.rename(columns={'Injury Result Description': 'Descricao_result_ferida'}, inplace = True)
df.rename(columns={'Incident Event Type': 'Tipo_Evento_Incidente'}, inplace = True)
df.rename(columns={'Victim Category': 'Categoria_Vitima'}, inplace = True)
df.rename(columns={'Victims Sex': 'Sexo_Vitima'}, inplace = True)
df.rename(columns={'Victims Age': 'Idade_Vitima'}, inplace = True)
df.head(5)
```

Out[17]:

	Ano	Data_Incidente	Rota	Operador	Nome_Grupo	Garagem_Onibus	Bairro	Descricao_r
0	2015	2015-01-01	1	London General	Go-Ahead	Garage Not Available	Southwark	Injuri
1	2015	2015-01-01	4	Metroline	Metroline	Garage Not Available	Islington	Injuri
2	2015	2015-01-01	5	East London	Stagecoach	Garage Not Available	Havering	Taken Reported S
3	2015	2015-01-01	5	East London	Stagecoach	Garage Not Available	None London Borough	Taken Reported S
4	2015	2015-01-01	6	Metroline	Metroline	Garage Not Available	Westminster	Reported M Treate

In [18]:

```
df['Data_Incidente'] = pd.to_datetime(df['Data_Incidente'])
df['Mes'] = df['Data_Incidente'].dt.month
df.head(5)
```

Out[18]:

	Ano	Data_Incidente	Rota	Operador	Nome_Grupo	Garagem_Onibus	Bairro	Descricao_r
0	2015	2015-01-01	1	London General	Go-Ahead	Garage Not Available	Southwark	Injuri
1	2015	2015-01-01	4	Metroline	Metroline	Garage Not Available	Islington	Injuri
2	2015	2015-01-01	5	East London	Stagecoach	Garage Not Available	Havering	Taken Reported S
3	2015	2015-01-01	5	East London	Stagecoach	Garage Not Available	None London Borough	Taken Reported S
4	2015	2015-01-01	6	Metroline	Metroline	Garage Not Available	Westminster	Reported M Treate

PERGUNTA 1

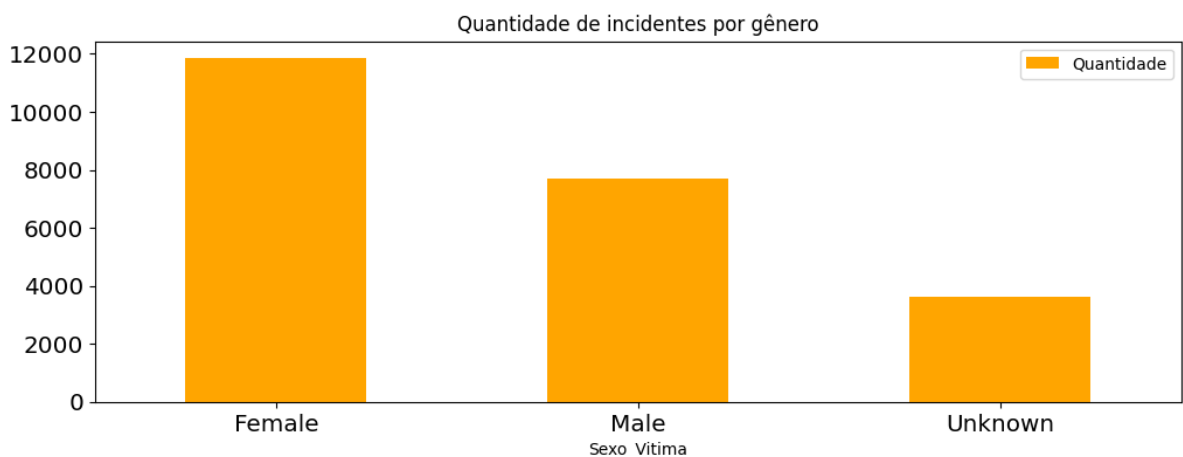
In [19]:

```
# Pergunta 1- Qual a quantidade de incidentes por gênero?
# Resposta 1 VIA QUERY
query = """
SELECT Sexo_Vitima, count(*) AS Quantidade
FROM df
GROUP BY Sexo_Vitima
;
"""
df1=sqldf.run(query)
df1
```

```
Out[19]:
```

	Sexo_Vitima	Quantidade
0	Female	11847
1	Male	7709
2	Unknown	3602

```
In [20]: # Pergunta 1- Qual a quantidade de incidentes por gênero?
# Resposta 1 VIA GRÁFICO
df1.plot(x='Sexo_Vitima',kind='bar', color = 'Orange')
plt.title('Quantidade de incidentes por gênero')
plt.xticks(rotation=0)
plt.show()
#plt.title(r'$\sigma_i=15$', fontsize=20)
#plt.bar(query[Sexo_Vitima], queryheight =3)
#plt.bar(df['Tipo_Evento_Incidente'], df['Sexo_Vitima'])
#plt.bar(, color = Red)
#plt.legend()
#plt.show()
```



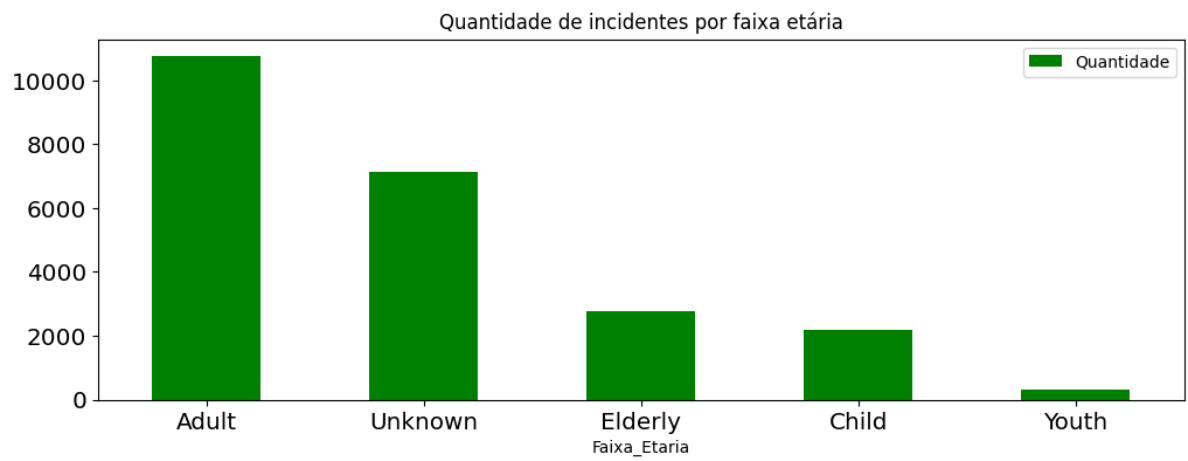
```
In [21]: # Pergunta 2- Qual faixa etária esteve mais envolvida nos incidentes?
# Resposta 2 VIA QUERY
# A faixa etária mais envolvida nos incidentes é a faixa adulta
query = """
SELECT Idade_Vitima AS Faixa_Etaria, count(*) AS Quantidade
FROM df
GROUP BY Idade_Vitima
ORDER BY count(*) DESC;
"""
df2 = sqldf.run(query)
df2
```

```
Out[21]:
```

	Faixa_Etaria	Quantidade
0	Adult	10754
1	Unknown	7135
2	Elderly	2769
3	Child	2181
4	Youth	319

## PERGUNTA 2

```
In [22]: # Pergunta 2- Qual faixa etária esteve mais envolvida nos incidentes?
# Resposta 2 VIA GRÁFICO
#OK!
# A faixa etária mais envolvida nos incidentes é a faixa adulta
df2.plot(x='Faixa_Etaria',kind='bar', color = 'Green')
plt.title('Quantidade de incidentes por faixa etária')
plt.xticks(rotation=0)
plt.show()
```



# PERGUNTA 3

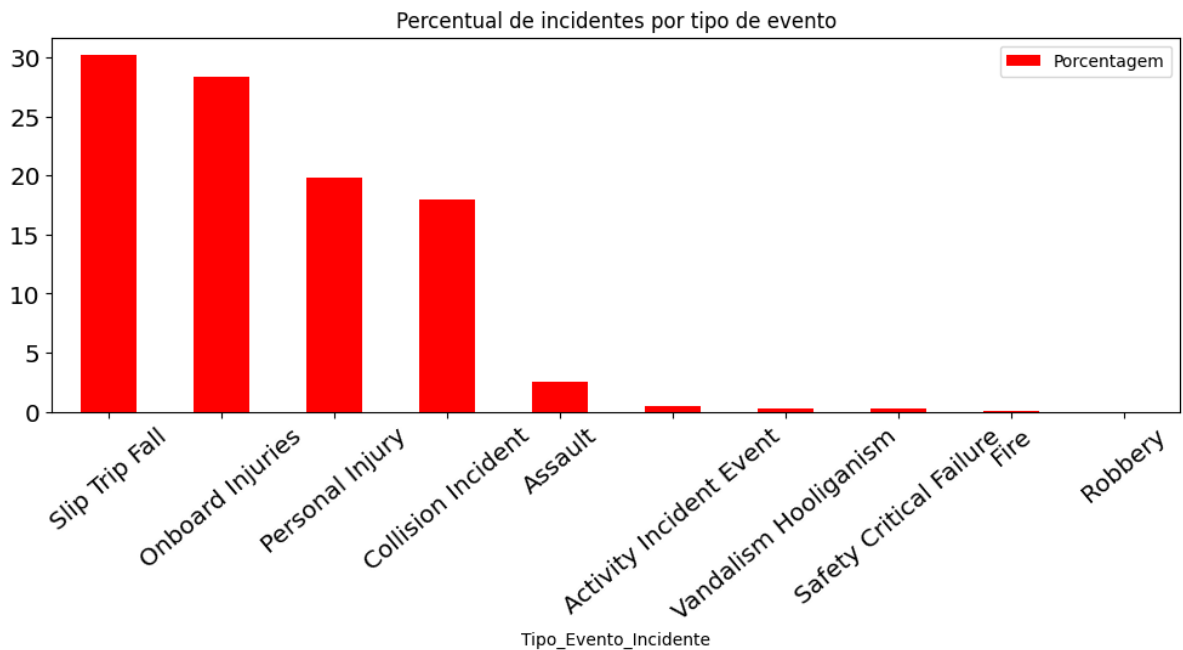
```
In [23]: # Pergunta 3- Qual o percentual de incidentes por tipo de evento (Incident Event Ty
# Resposta 3 VIA QUERY
query = """
SELECT Tipo_Evento_Incidente, count(*) AS Quantidade, cast( 100. * count(*) / sum(c
FROM df
GROUP BY Tipo_Evento_Incidente
ORDER BY count(*) DESC
;
"""

df3 = sqldf.run(query)
df3.rename(columns={'cast( 100. * count(*) / sum(count(*) over () as decimal(10,2)
df3['Porcentagem'] = round (df3['Porcentagem'],2)
df3
```

Out[23]:

	Tipo_Evento_Incidente	Quantidade	Porcentagem
0	Slip Trip Fall	6981	30.15
1	Onboard Injuries	6563	28.34
2	Personal Injury	4596	19.85
3	Collision Incident	4166	17.99
4	Assault	590	2.55
5	Activity Incident Event	114	0.49
6	Vandalism Hooliganism	73	0.32
7	Safety Critical Failure	66	0.28
8	Fire	6	0.03
9	Robbery	3	0.01

```
In [24]: # Pergunta 3- Qual o percentual de incidentes por tipo de evento (Incident Event Ty
# Resposta 3 VIA GRÁFICO
#OK!
df3.plot(x='Tipo_Evento_Incidente', y='Porcentagem', kind='bar', color='Red')
plt.title('Percentual de incidentes por tipo de evento')
plt.xticks(rotation=40)
plt.show()
```



## PERGUNTA 4

```
In [25]: # Pergunta 4- Como foi a evolução de incidentes por mês ao longo do tempo?
# Resposta 4 VIA QUERY
query = """
SELECT Mes, count(*) AS Quantidade
FROM df
GROUP BY Mes
;
"""
df4 = sqldf.run(query)
df4
```

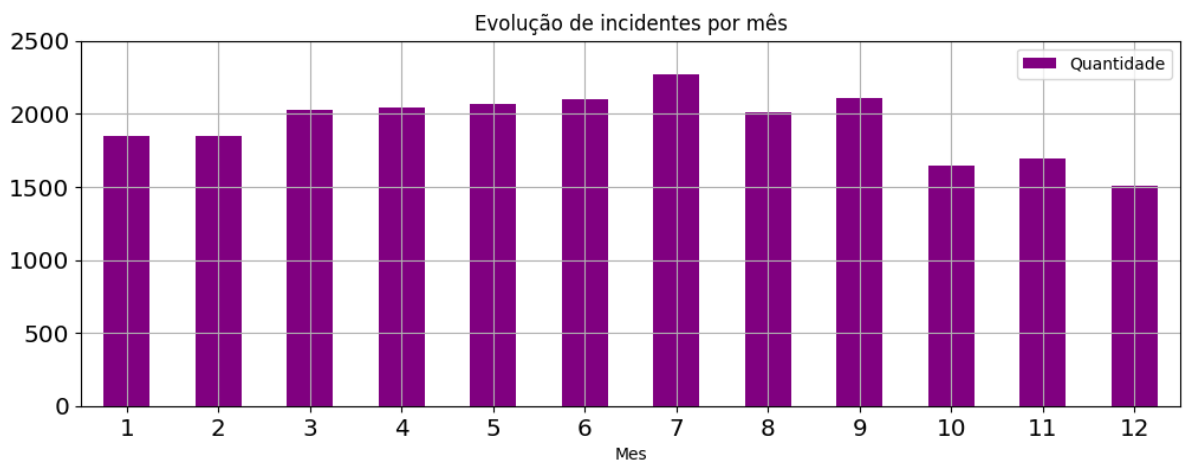


Out[25]:

	Mes	Quantidade
0	1	1847
1	2	1849
2	3	2023
3	4	2044
4	5	2065
5	6	2097
6	7	2268
7	8	2013
8	9	2109
9	10	1642
10	11	1691
11	12	1510

In [26]:

```
# Pergunta 4- Como foi a evolução de incidentes por mês ao longo do tempo?
# Resposta 4 VIA GRÁFICO
df4.plot(x='Mes',y='Quantidade', kind='bar', color = 'Purple')
plt.yticks(np.arange(0, 3000, step=500))
plt.xticks(rotation=0)
plt.title('Evolução de incidentes por mês')
plt.grid()
plt.show()
```



## PERGUNTA 5

In [27]:

```
# Pergunta 5- Quando o incidente foi "Collision Incident" em qual mês houve o maior
#envolvendo pessoas do sexo feminino?
# Resposta 5 VIA QUERY
#Quando o incidente foi "Collision Incident", no mês de Setembro ocorreu maior número
#envolvendo pessoas do sexo feminino
query = """
SELECT Mes, count(*) AS Quantidade
FROM df
WHERE Tipo_Evento_Incidente = 'Collision Incident' AND Sexo_Vitima = 'Female'
GROUP BY Mes
ORDER BY count(*) DESC
```

```
;
"""
df5 = sqldf.run(query)
df5
```

Out[27]:

	Mes	Quantidade
0	9	158
1	6	150
2	7	147
3	4	147
4	11	140
5	1	137
6	3	129
7	5	123
8	8	122
9	2	116
10	10	113
11	12	76

```
In [28]: # Pergunta 5- Quando o incidente foi “Collision Incident” em qual mês houve o maior
#envolvendo pessoas do sexo feminino?
# Resposta 5 VIA GRÁFICO
#Quando o incidente foi “Collision Incident”, no mês de Setembro ocorreu maior número
#envolvendo pessoas do sexo feminino
df5.plot(x='Mes', y='Quantidade',kind='bar', color = '#157143')
plt.title('Quantidade de incidentes por mês pessoas do sexo feminino para incidente
plt.xticks(rotation=0)
plt.grid()
plt.show()
```



## PERGUNTA 6

```
In [29]: # Pergunta 6- Qual foi a média de incidentes por mês envolvendo crianças (Child)?
# Resposta 6 VIA QUERY
query = """
SELECT Mes, count(*) as Media
```

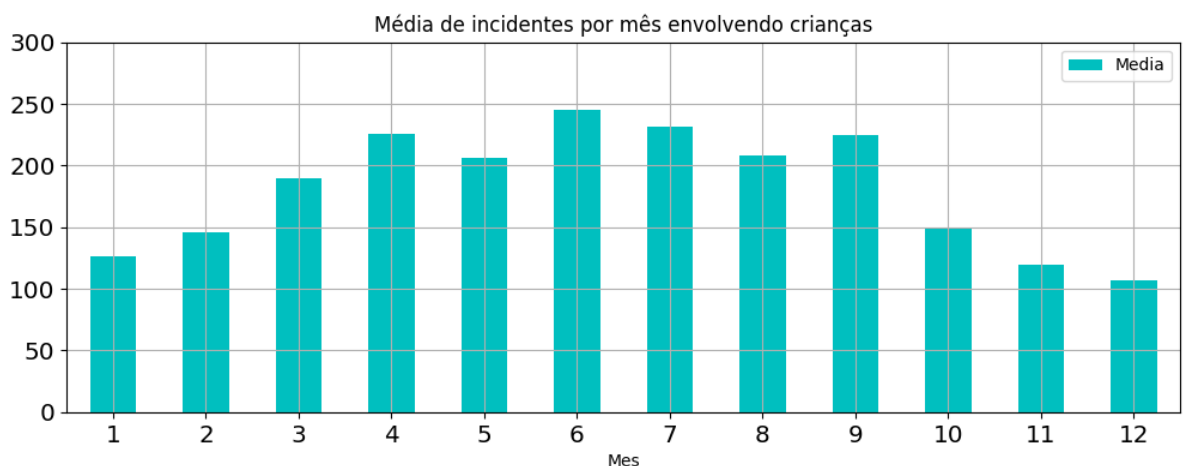
```
FROM df
WHERE Idade_Vitima = 'Child'
GROUP BY Mes
;
"""
df6 = sqldf.run(query)
df6
```

Out[29]:

	Mes	Media
0	1	126
1	2	146
2	3	190
3	4	226
4	5	206
5	6	245
6	7	232
7	8	208
8	9	225
9	10	150
10	11	120
11	12	107

In [30]:

```
# Pergunta 6- Qual foi a média de incidentes por mês envolvendo crianças (Child)?
# Resposta 6 VIA GRAFICO
df6.plot(x='Mes', y='Media', kind='bar', color='c')
plt.title('Média de incidentes por mês envolvendo crianças')
plt.yticks(np.arange(0, 350, step=50))
plt.xticks(rotation=0)
plt.grid()
plt.show()
```



In [31]:

```
# Pergunta 7- Considerando a descrição de incidente como "Injuries treated on scene"
# qual o total de incidentes de pessoas do sexo masculino e sexo feminino?
# Resposta 7 VIA QUERY
# Total de incidents com sexo feminino 8816 para incidente "Lesões tratadas no local"
# Total de incidents com sexo masculino 7709 para incidente "Lesões tratadas no local"
query = """
SELECT Sexo_Vitima, count(*) AS Quantidade
```

```

FROM df
WHERE Descricao_result_ferida = 'Injuries treated on scene' AND Sexo_Vitima = 'Female'
GROUP BY Sexo_Vitima
ORDER BY count(*) DESC
;
"""
df7 = sqldf.run(query)
df7

```

```

Out[31]:
  Sexo_Vitima  Quantidade
0      Female         8816
1      Male          7709

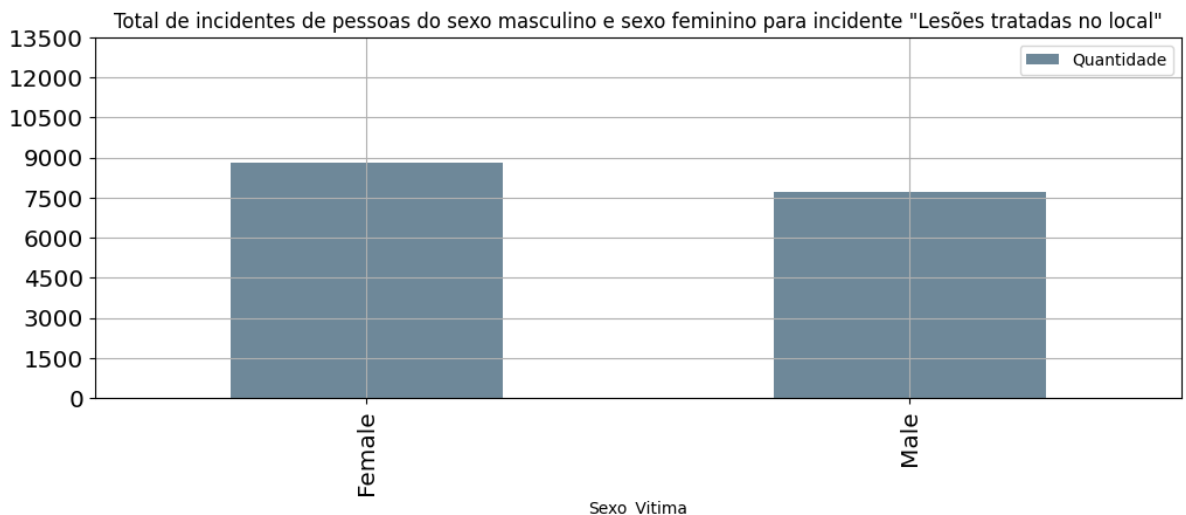
```

## PERGUNTA 7

```

In [32]: # Pergunta 7- Considerando a descrição de incidente como "Injuries treated on scene"
# qual o total de incidentes de pessoas do sexo masculino e sexo feminino?
# Resposta 7 VIA GRAFICO
# Total de incidents com sexo feminino 8816 para incidente "Lesões tratadas no local"
# Total de incidents com sexo masculino 7709 para incidente "Lesões tratadas no local"
df7.plot(x='Sexo_Vitima', kind='bar', color = '#6e8899')
plt.title('Total de incidentes de pessoas do sexo masculino e sexo feminino para incidente "Lesões tratadas no local"')
plt.yticks(np.arange(0, 15000, step=1500))
plt.grid()
plt.show()

```



## PERGUNTA 8

```

In [33]: # Pergunta 8- No ano de 2017 em qual mês houve mais incidentes com idosos (Elderly)?
# Resposta 8 VIA QUERY
# No ano de 2017, o mês de Julho teve mais incidentes com idosos
query = """
SELECT Mes, count(*) AS Quantidade
FROM df
WHERE Ano = '2017' AND Idade_Vitima = 'Elderly'
GROUP BY Mes
ORDER BY count(*) DESC
;
"""

```

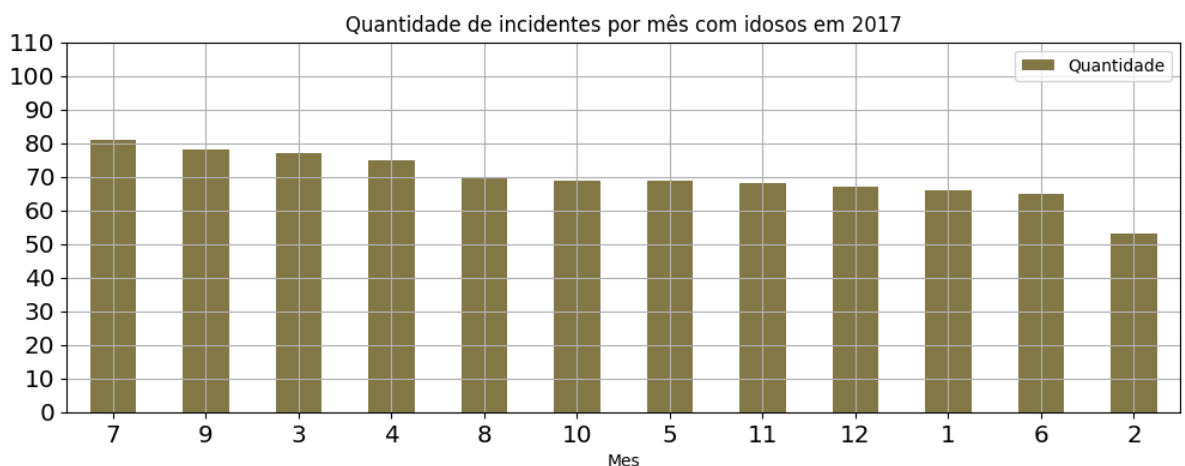
```
df8 = sqldf.run(query)
df8
```

Out[33]:

	Mes	Quantidade
0	7	81
1	9	78
2	3	77
3	4	75
4	8	70
5	10	69
6	5	69
7	11	68
8	12	67
9	1	66
10	6	65
11	2	53

In [34]:

```
# Pergunta 8- No ano de 2017 em qual mês houve mais incidentes com idosos (Elderly)
# Resposta 8 VIA GRAFICO
#No ano de 2017, o mês de Julho teve mais incidentes com idosos
df8.plot(x='Mes', y='Quantidade', kind='bar', color='#827845')
plt.title('Quantidade de incidentes por mês com idosos em 2017')
plt.yticks(np.arange(0, 120, step=10))
plt.xticks(rotation=0)
plt.grid()
plt.show()
```



## PERGUNTA 9

In [35]:

```
# Pergunta 9- Considerando o Operador qual a distribuição de incidentes ao longo do ano
# Resposta 9 VIA QUERY
query = """
SELECT Ano, Operador, count(*) AS Quantidade
FROM df
GROUP BY Operador, Ano
#ORDER BY count(*) DESC
```

```
;
"""
df9 = sqldf.run(query)
df9.head()
```

Out[35]:

	Ano	Operador	Quantidade
0	2015	Abellio London	117
1	2016	Abellio London	233
2	2017	Abellio London	329
3	2018	Abellio London	284
4	2015	Abellio West	27

In [36]:

```
#lista com nomes dos operadores
#nomes_operadores[nomes_operadores == 'Sullivan Buses']
nomes_operadores=pd.unique(df9['Operador'])
nomes_operadores
```

Out[36]:

```
array(['Abellio London', 'Abellio West', 'Arriva Kent Thameside',
      'Arriva London North', 'Arriva London South', 'Arriva The Shires',
      'Blue Triangle Buses', 'C T Plus', 'Docklands Buses',
      'East London', 'H R Richmond', 'London Central', 'London General',
      'London Sovereign', 'London United', 'Metrobus', 'Metroline',
      'Metroline West', 'Quality Line', 'Selkent',
      'Sullivan Bus & Coach', 'Sullivan Buses', 'TGM GROUP',
      'Tower Transit', 'Uno Buses'], dtype=object)
```

In [37]:

```
pd.set_option('display.max_rows', None)
df9[['Ano', 'Operador']]
```

Out[37]:

	Ano	Operador
0	2015	Abellio London
1	2016	Abellio London
2	2017	Abellio London
3	2018	Abellio London
4	2015	Abellio West
5	2016	Abellio West
6	2017	Abellio West
7	2018	Abellio West
8	2015	Arriva Kent Thameside
9	2016	Arriva Kent Thameside
10	2015	Arriva London North
11	2016	Arriva London North
12	2017	Arriva London North
13	2018	Arriva London North
14	2015	Arriva London South
15	2016	Arriva London South
16	2017	Arriva London South
17	2018	Arriva London South
18	2015	Arriva The Shires
19	2016	Arriva The Shires
20	2017	Arriva The Shires
21	2015	Blue Triangle Buses
22	2016	Blue Triangle Buses
23	2017	Blue Triangle Buses
24	2018	Blue Triangle Buses
25	2015	C T Plus
26	2016	C T Plus
27	2017	C T Plus
28	2018	C T Plus
29	2015	Docklands Buses
30	2016	Docklands Buses
31	2017	Docklands Buses
32	2018	Docklands Buses
33	2015	East London
34	2016	East London
35	2017	East London

	Ano	Operador
36	2018	East London
37	2015	H R Richmond
38	2016	H R Richmond
39	2017	H R Richmond
40	2015	London Central
41	2016	London Central
42	2017	London Central
43	2018	London Central
44	2015	London General
45	2016	London General
46	2017	London General
47	2018	London General
48	2015	London Sovereign
49	2016	London Sovereign
50	2017	London Sovereign
51	2018	London Sovereign
52	2015	London United
53	2016	London United
54	2017	London United
55	2018	London United
56	2015	Metrobus
57	2016	Metrobus
58	2017	Metrobus
59	2018	Metrobus
60	2015	Metroline
61	2016	Metroline
62	2017	Metroline
63	2018	Metroline
64	2015	Metroline West
65	2016	Metroline West
66	2017	Metroline West
67	2018	Metroline West
68	2017	Quality Line
69	2018	Quality Line
70	2015	Selkent
71	2016	Selkent



	Ano	Operador
72	2017	Selkent
73	2018	Selkent
74	2018	Sullivan Bus & Coach
75	2016	Sullivan Buses
76	2016	TGM GROUP
77	2017	TGM GROUP
78	2015	Tower Transit
79	2016	Tower Transit
80	2017	Tower Transit
81	2018	Tower Transit
82	2017	Uno Buses

```
In [38]: #ocorre problema porquene todos os operadores tem dados dos 4 anos
#a=0
#b=0
#for i in range (len(nomes_operadores) -1 ):
#    df9.iloc[a:a+4].plot(x='Ano',y='Quantidade', kind='bar', label = nomes_operadores[i])
#    a+=4
#    b+=1
#    plt.xticks(rotation=0)
#    plt.show()
```

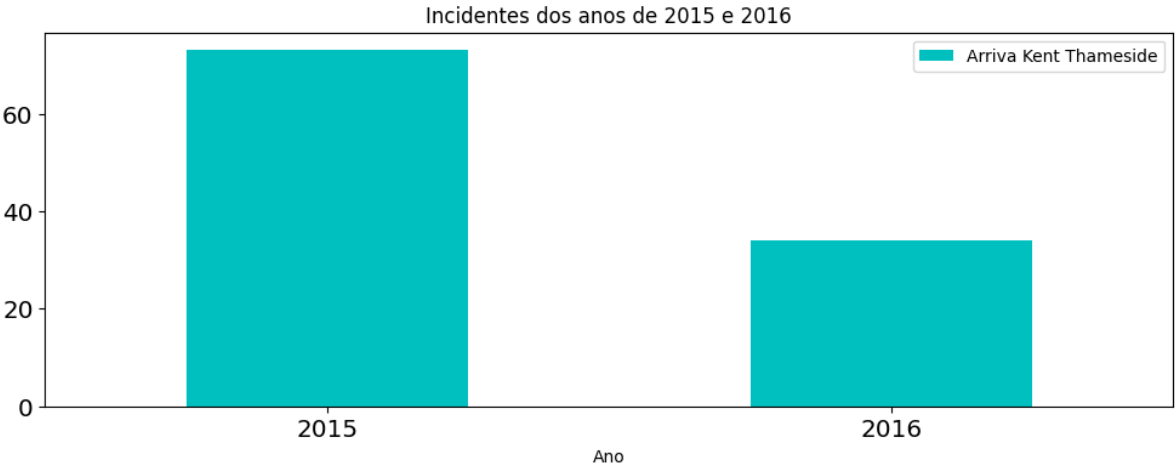
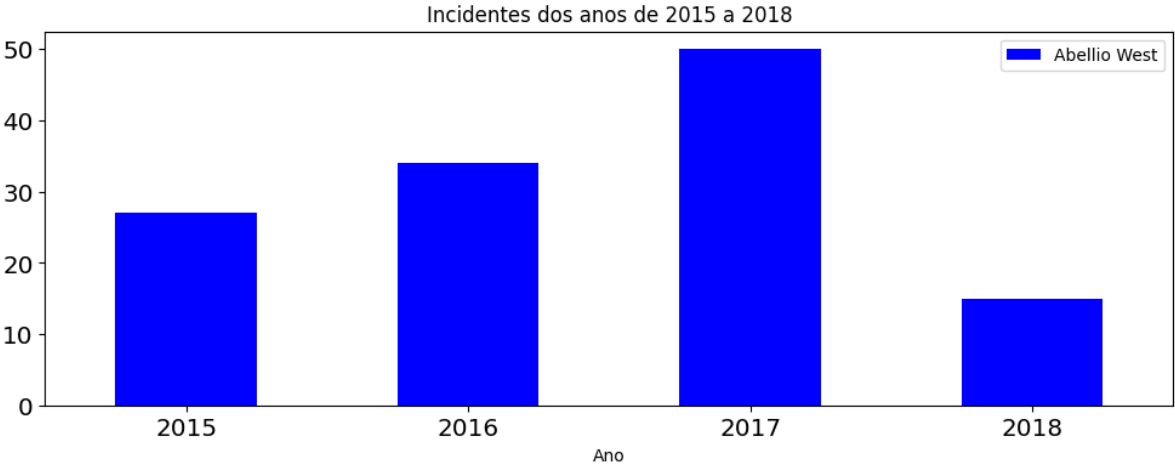
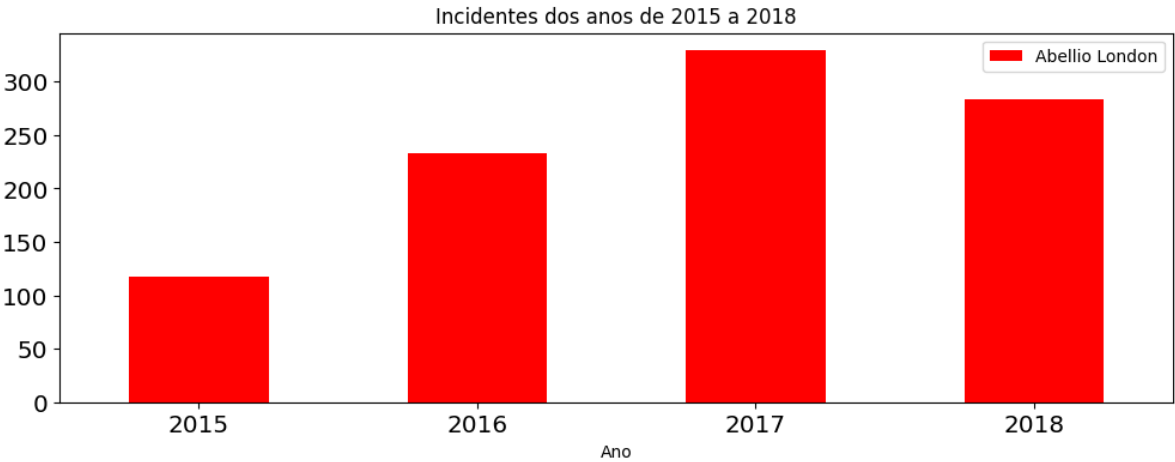
```
In [39]: # Pergunta 9- Considerando o Operador qual a distribuição de incidentes ao Longo do tempo?
# Resposta 9 VIA GRAFICO
df9.iloc[0:4].plot(x='Ano',y='Quantidade', kind='bar', label = 'Abellio London', color='red')
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[4:8].plot(x='Ano',y='Quantidade', kind='bar', label = 'Abellio West', color='blue')
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[8:10].plot(x='Ano',y='Quantidade', kind='bar', label = 'Arriva Kent Thames Valley', color='green')
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 e 2016')
plt.show()
df9.iloc[10:14].plot(x='Ano',y='Quantidade', kind='bar', label = 'Arriva London North', color='orange')
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[14:18].plot(x='Ano',y='Quantidade', kind='bar', label = 'Arriva London South', color='purple')
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[18:21].plot(x='Ano',y='Quantidade', kind='bar', label = 'Arriva The Shires', color='brown')
plt.xticks(rotation=0)
plt.show()
df9.iloc[21:25].plot(x='Ano',y='Quantidade', kind='bar', label = 'Blue Triangle Buses', color='pink')
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[25:29].plot(x='Ano',y='Quantidade', kind='bar', label = 'C T Plus', color='gray')
```

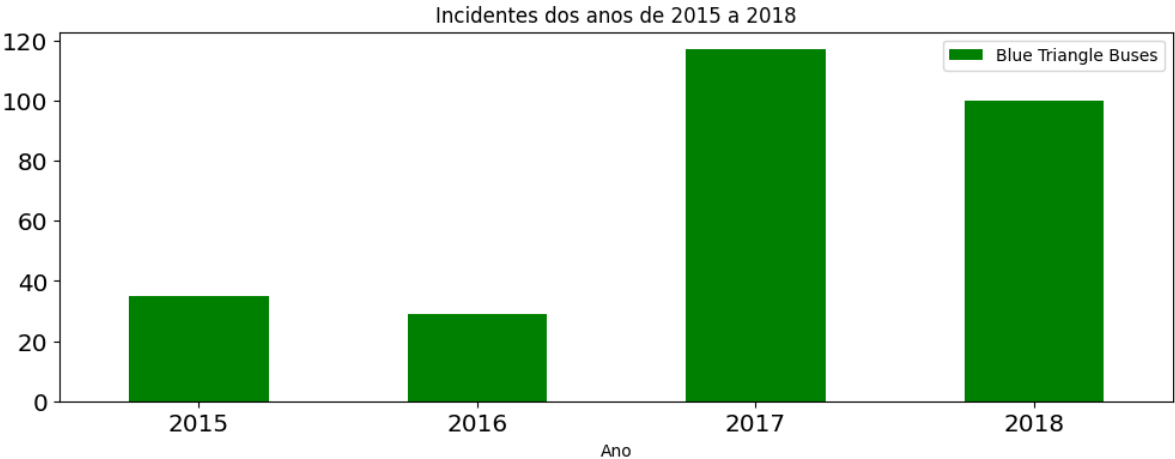
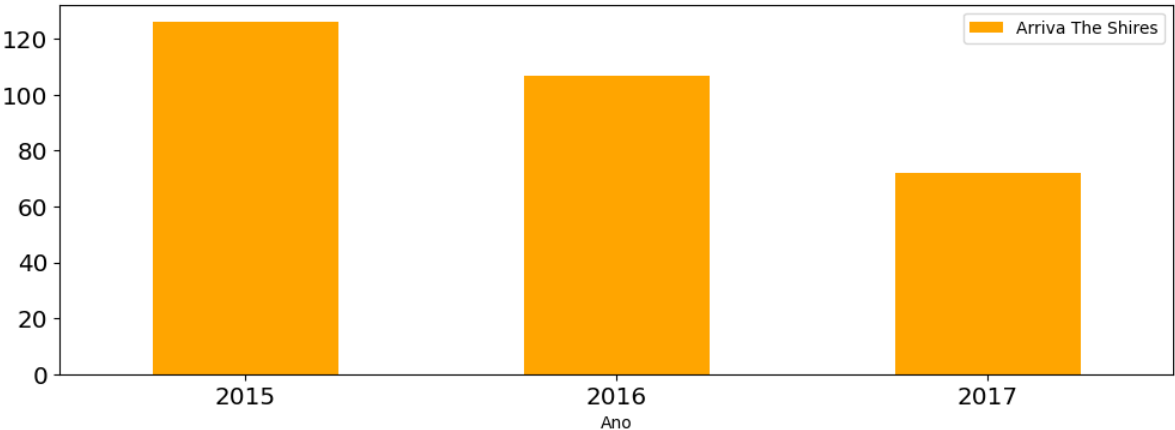
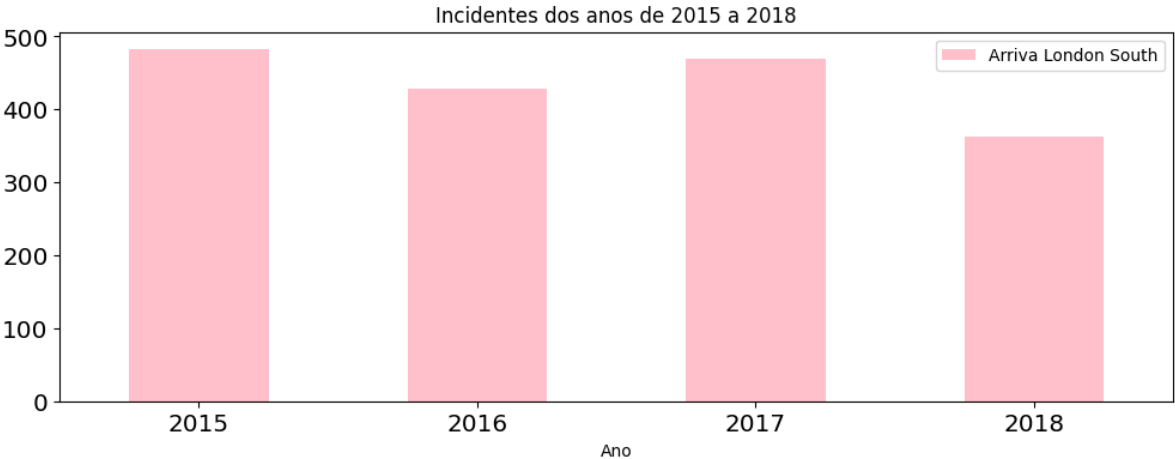
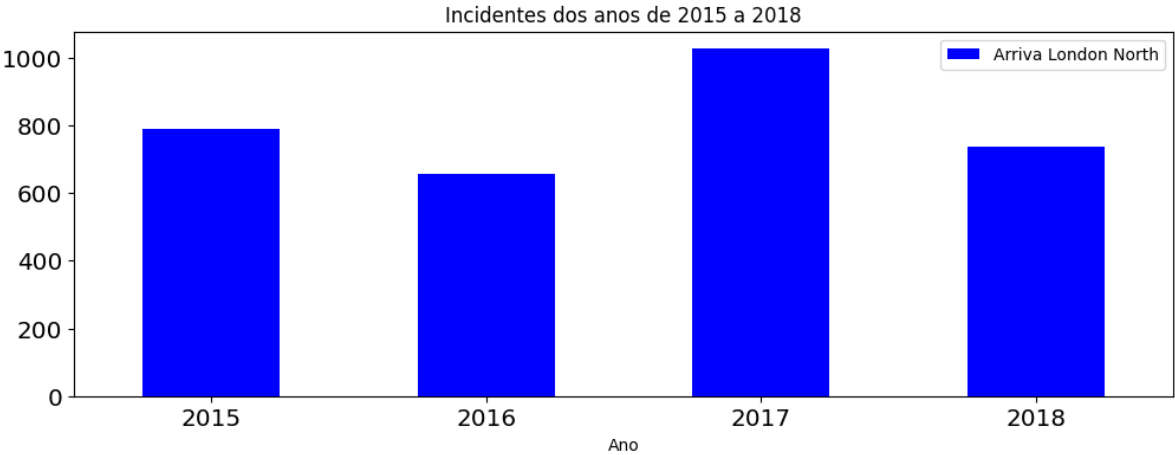
```

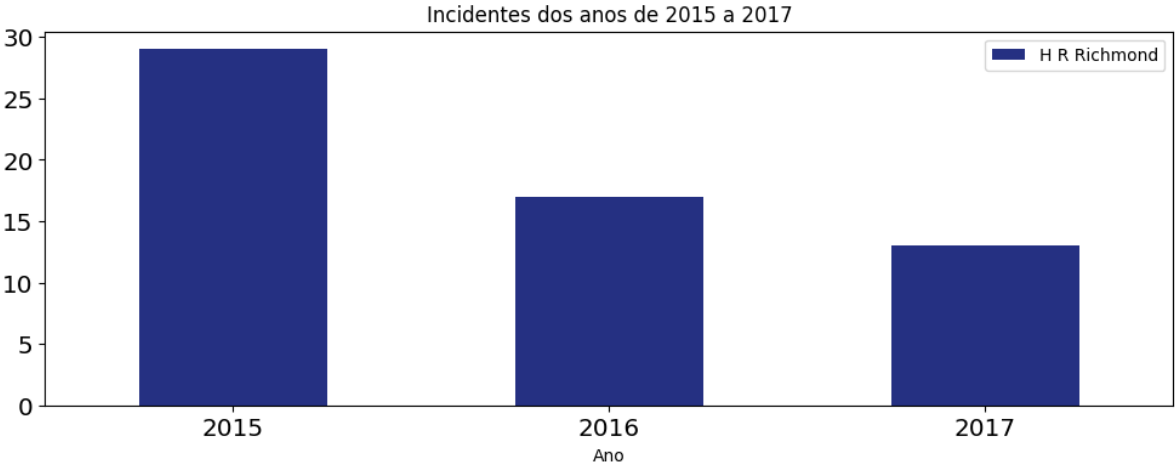
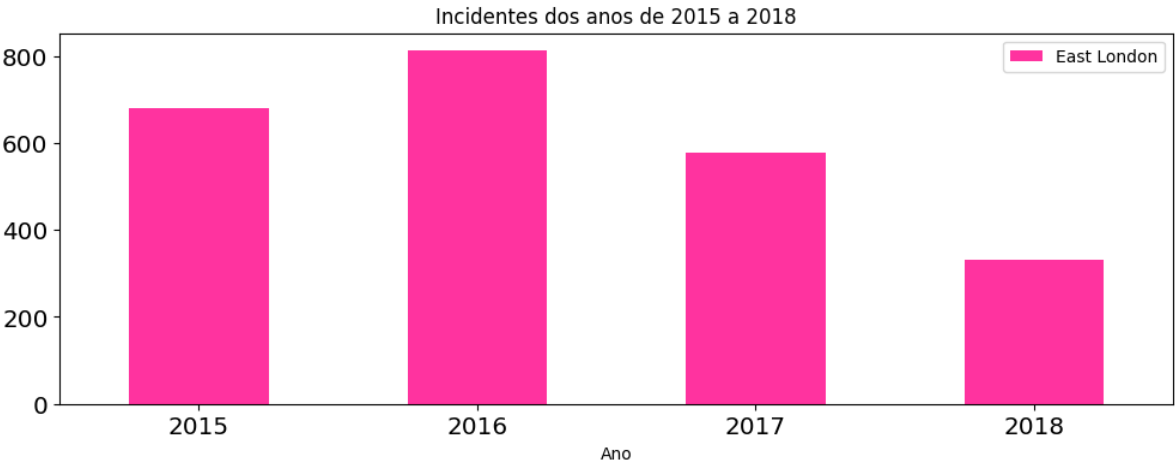
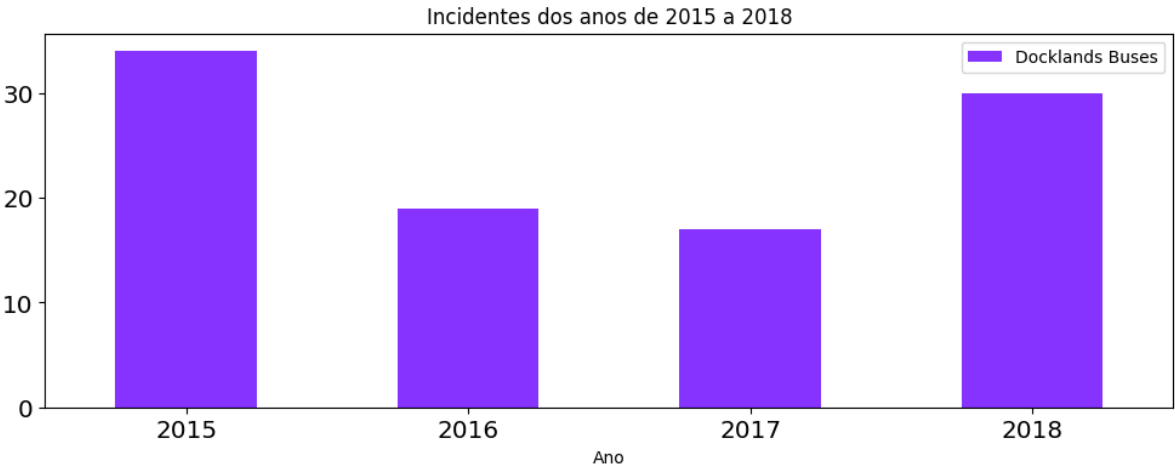
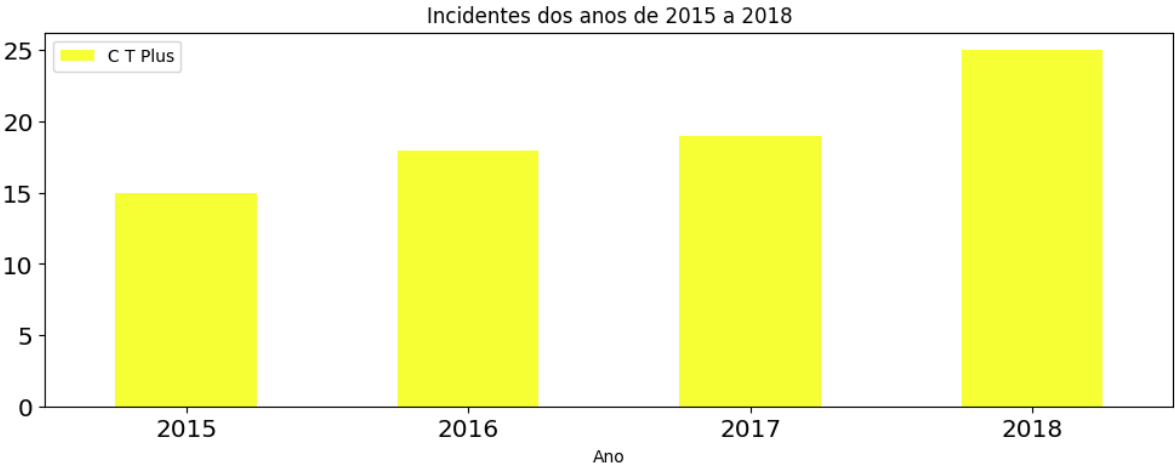
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[29:33].plot(x='Ano',y='Quantidade', kind='bar', label = 'Docklands Buses',
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[33:37].plot(x='Ano',y='Quantidade', kind='bar', label = 'East London', col
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[37:40].plot(x='Ano',y='Quantidade', kind='bar', label = 'H R Richmond', co
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2017')
plt.show()
df9.iloc[40:44].plot(x='Ano',y='Quantidade', kind='bar', label = 'London Central',
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[44:48].plot(x='Ano',y='Quantidade', kind='bar', label = 'London General',
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[48:52].plot(x='Ano',y='Quantidade', kind='bar', label = 'London Sovereign'
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[52:56].plot(x='Ano',y='Quantidade', kind='bar', label = 'London United', c
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[56:60].plot(x='Ano',y='Quantidade', kind='bar', label = 'Metrobus', color
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[60:64].plot(x='Ano',y='Quantidade', kind='bar', label = 'Metroline', color
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[64:68].plot(x='Ano',y='Quantidade', kind='bar', label = 'Metroline West',
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[68:70].plot(x='Ano',y='Quantidade', kind='bar', label = 'Quality Line', co
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2017 e 2018')
plt.show()
df9.iloc[70:74].plot(x='Ano',y='Quantidade', kind='bar', label = 'Selkent', color =
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[74:75].plot(x='Ano',y='Quantidade', kind='bar', label = 'Sullivan Bus & Co
plt.xticks(rotation=0)
plt.title('Incidentes do ano de 2018')
plt.show()
df9.iloc[75:76].plot(x='Ano',y='Quantidade', kind='bar', label = 'Sullivan Buses',
plt.xticks(rotation=0)
plt.title('Incidentes do ano de 2016')
plt.show()
df9.iloc[76:78].plot(x='Ano',y='Quantidade', kind='bar', label = 'TGM GROUP', color
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2016 e 2017')
plt.show()
df9.iloc[78:82].plot(x='Ano',y='Quantidade', kind='bar', label = 'Tower Transit', c

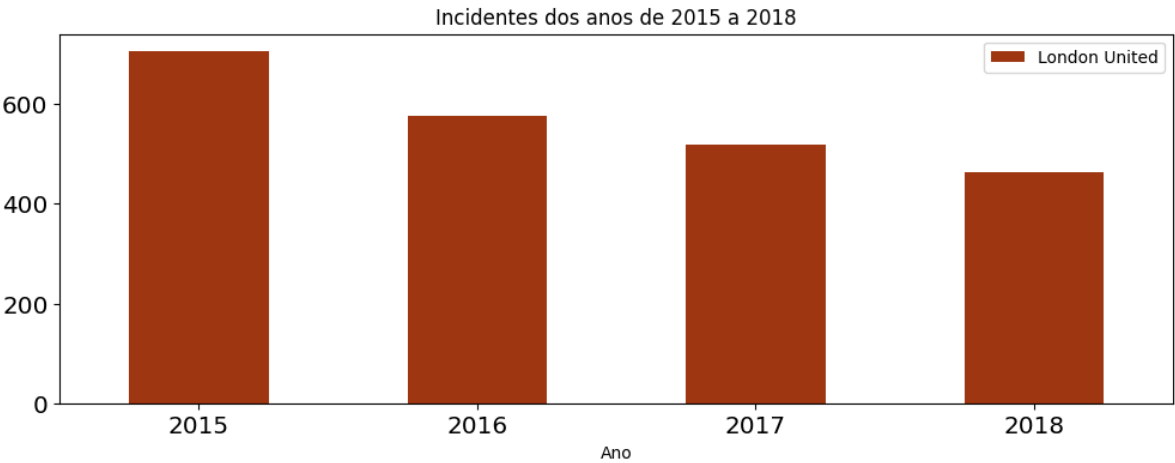
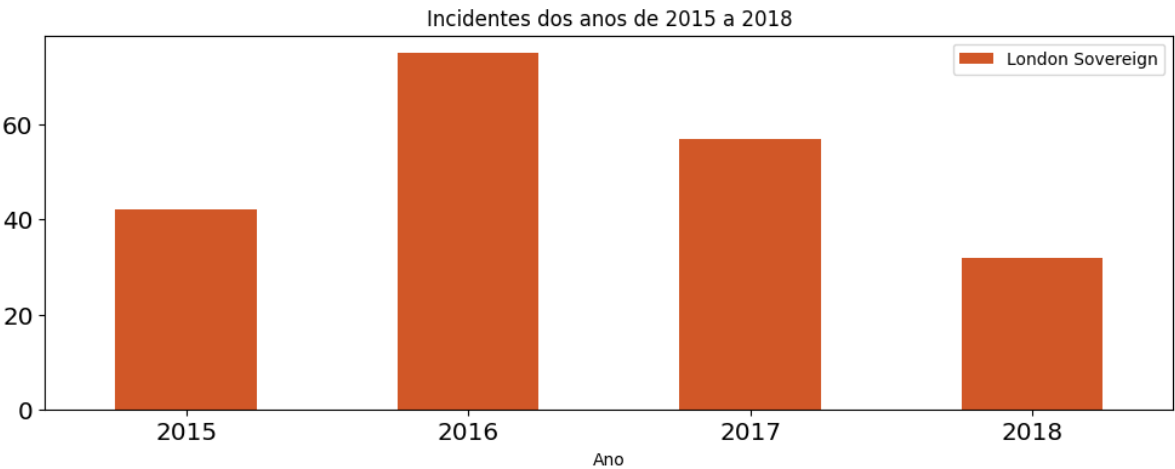
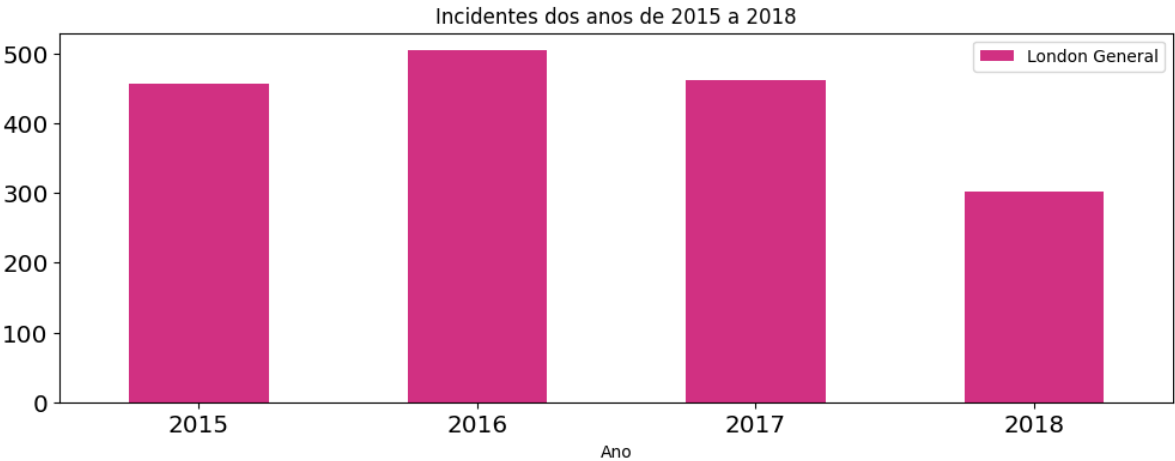
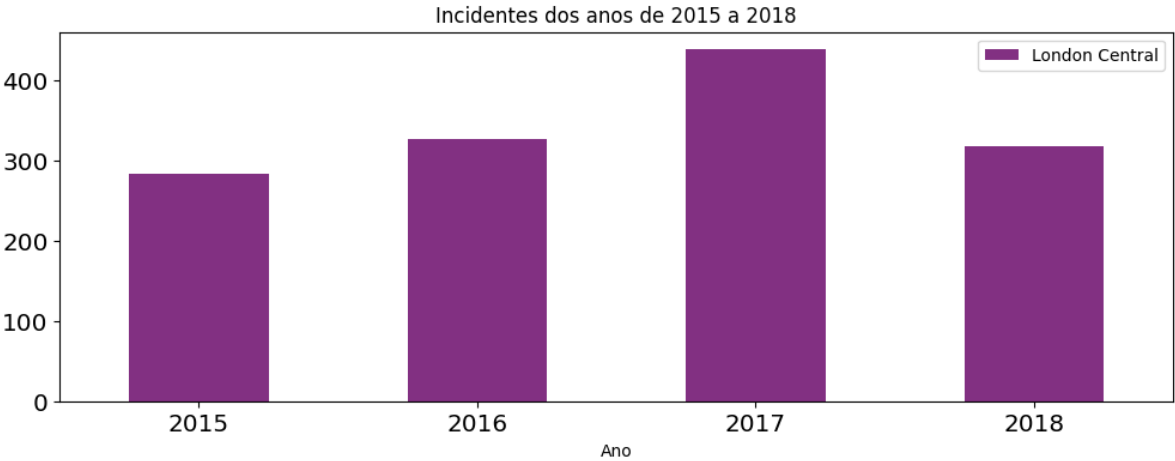
```

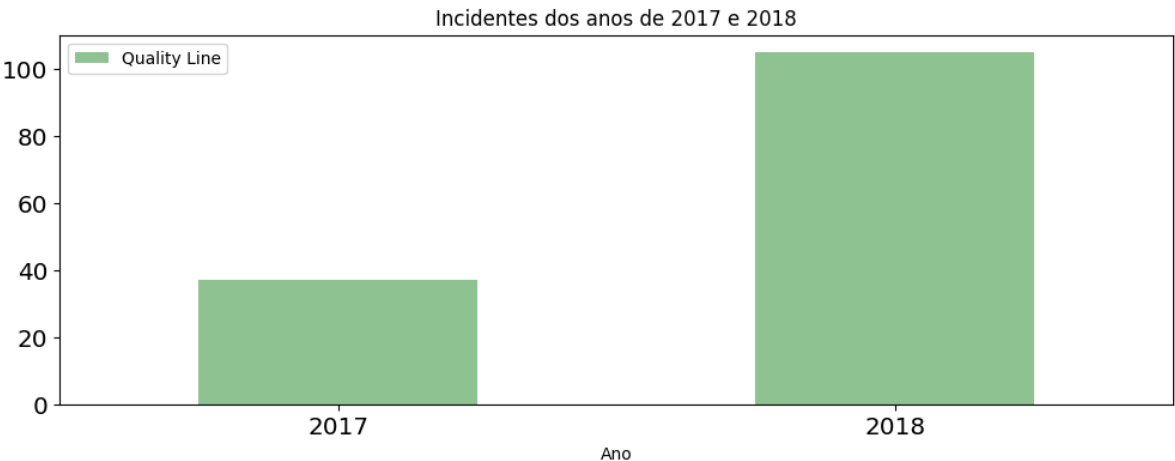
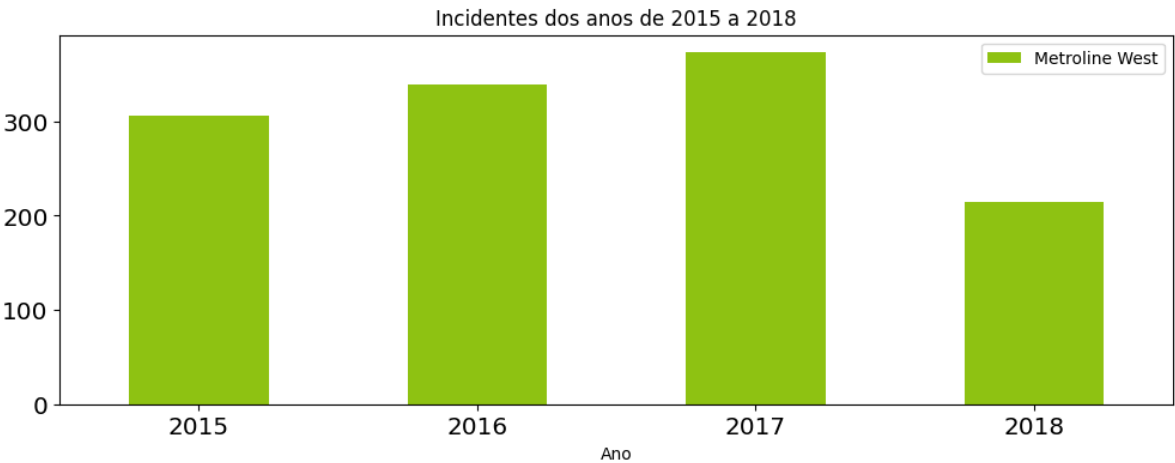
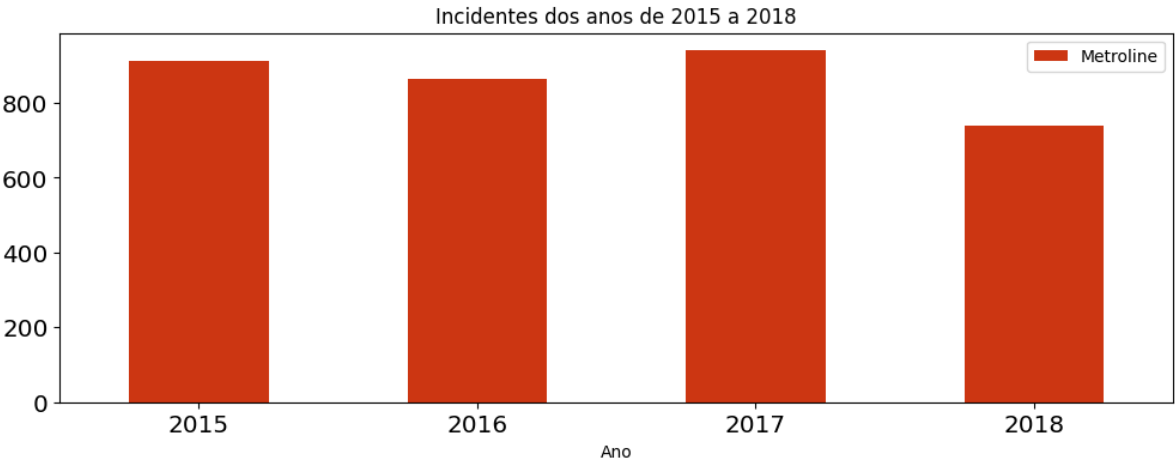
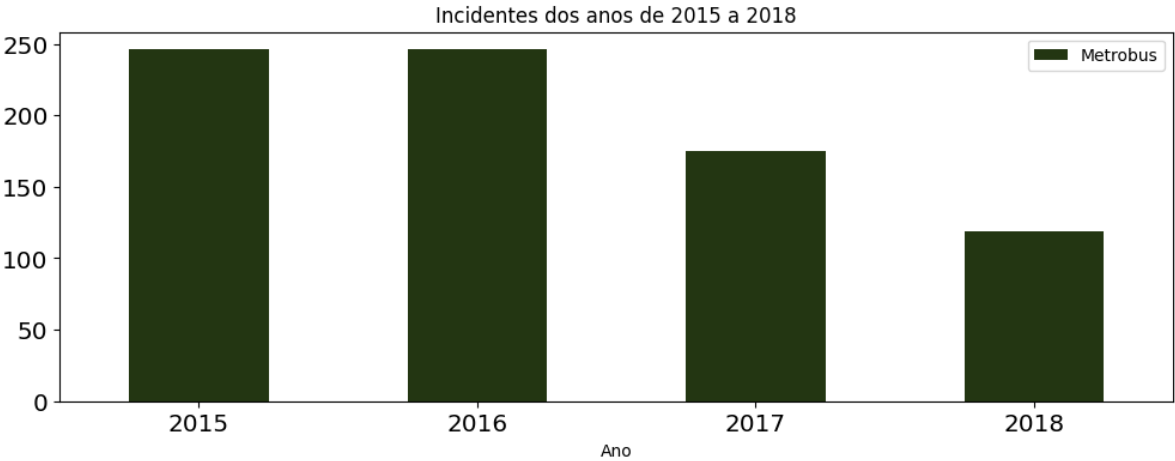
```
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 a 2018')
plt.show()
df9.iloc[82:83].plot(x='Ano',y='Quantidade', kind='bar', label = 'Uno Buses', color=
plt.xticks(rotation=0)
plt.title('Incidentes do ano de 2017')
plt.show()
```

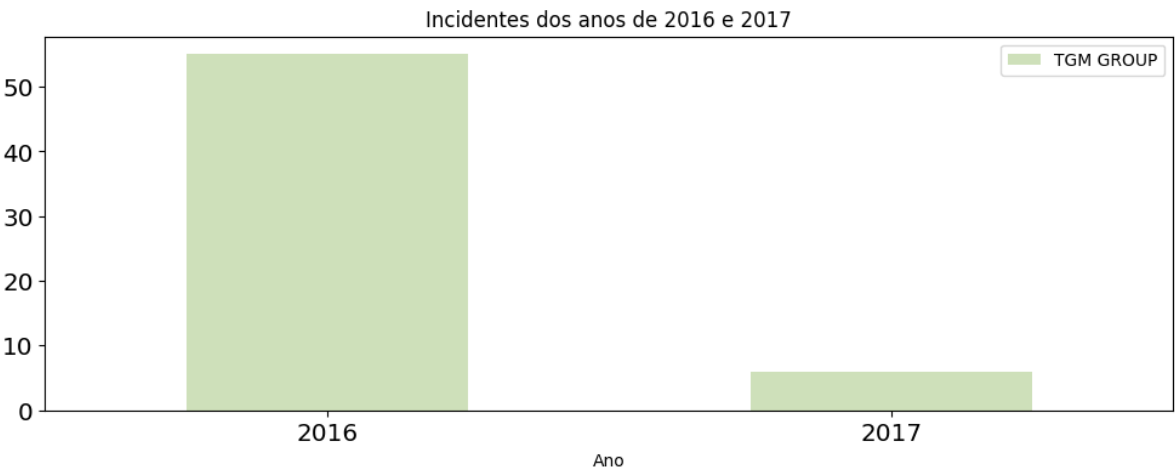
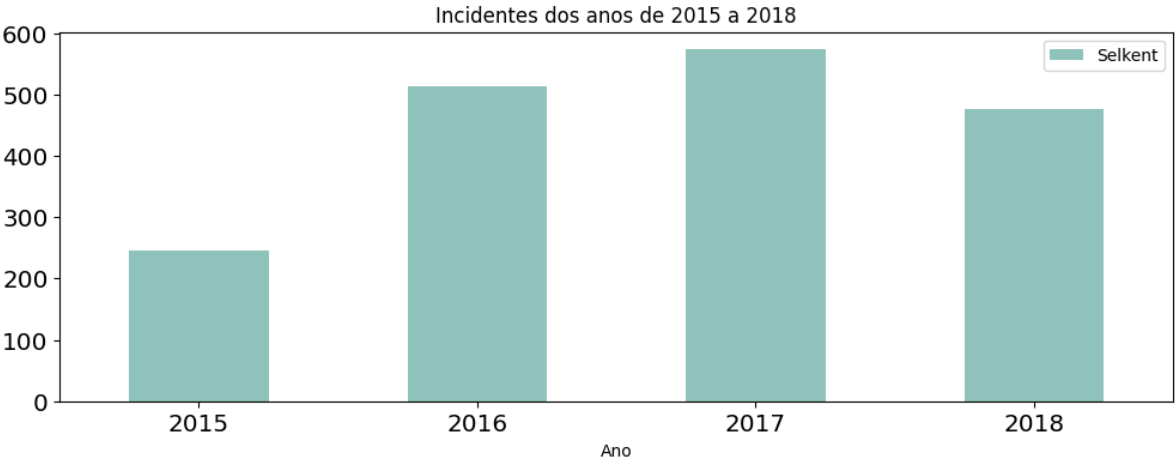




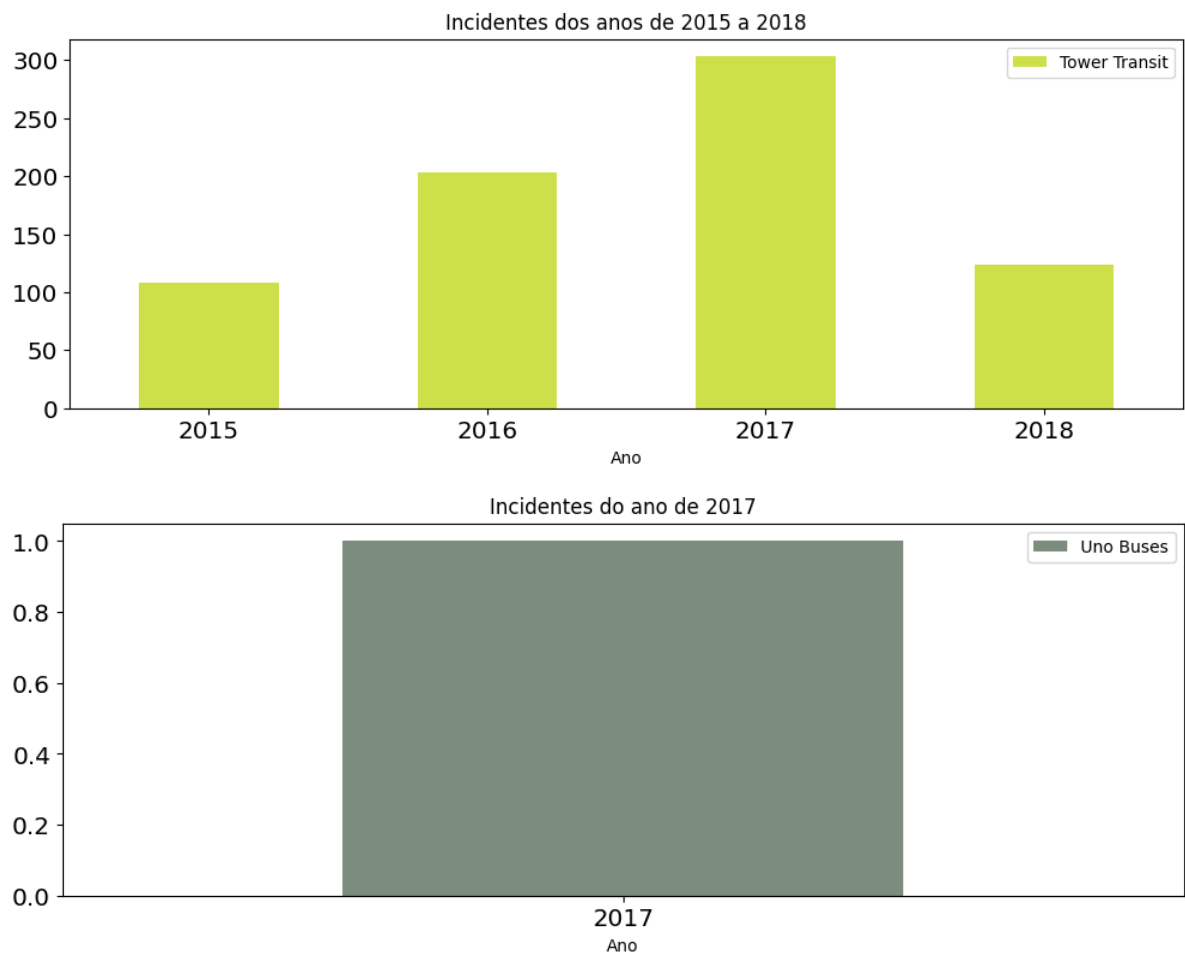












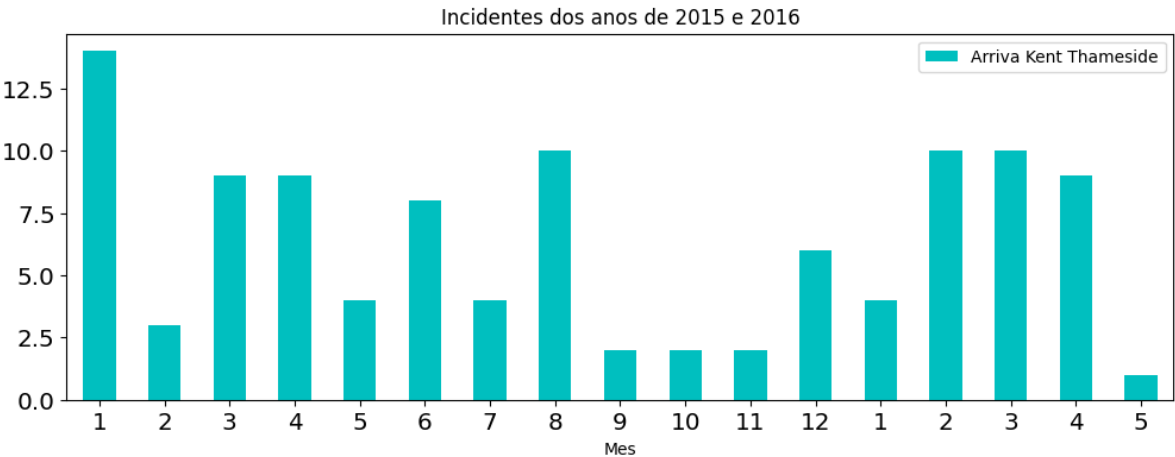
```
In [40]: #para operadores com somente 1 ou 2 anos vou apresentar a evolução em meses também
query = """
SELECT Mes,Ano, Operador, count(*) AS Quantidade
FROM df
GROUP BY Operador, Ano, Mes
#ORDER BY count(*) DESC
;
"""
df9mes = sqldf.run(query)
df9mes.head()
```

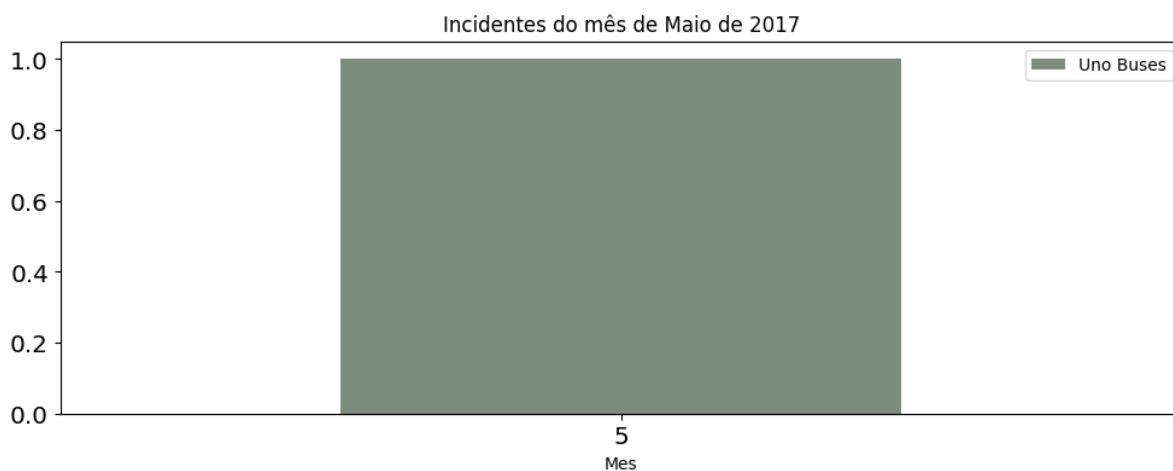
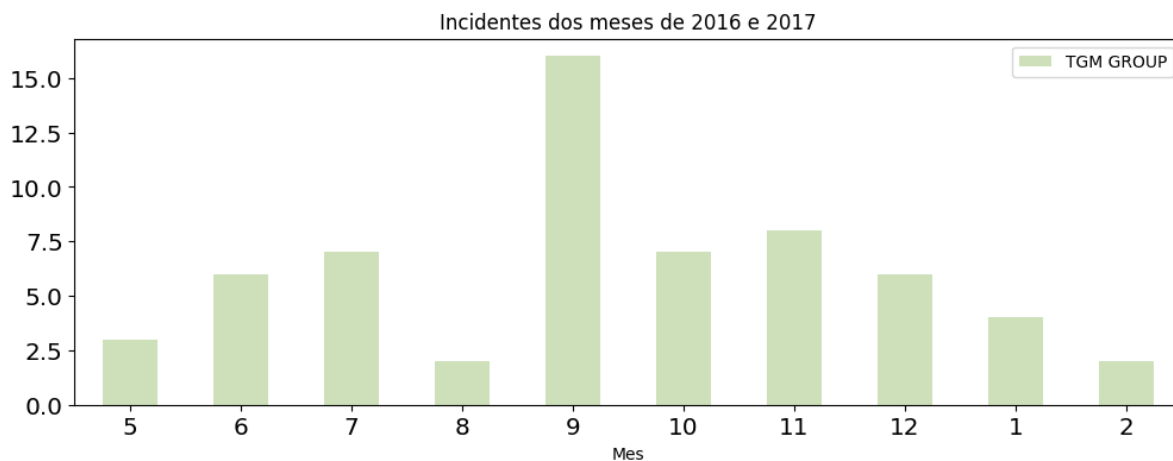
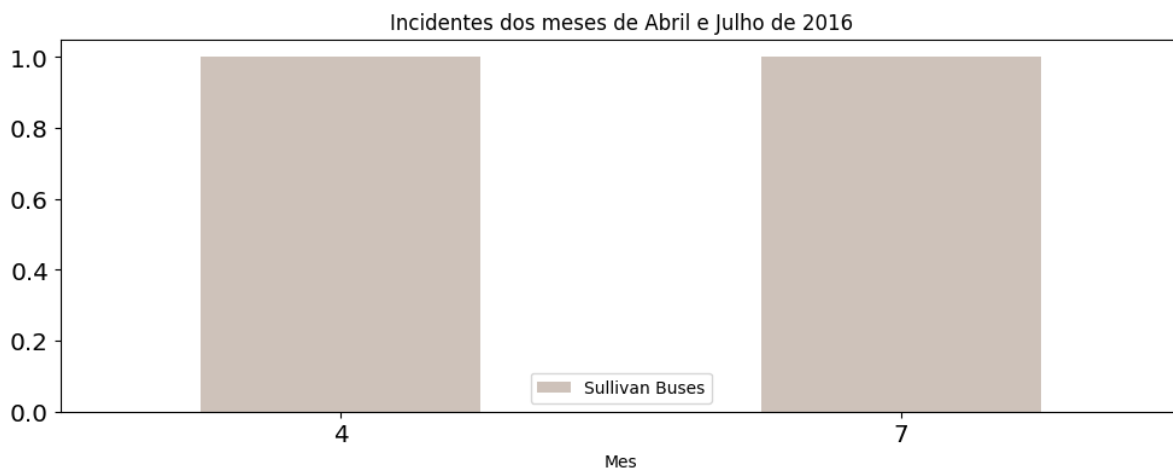
Out[40]:

	Mes	Ano	Operador	Quantidade
0	1	2015	Abellio London	11
1	2	2015	Abellio London	6
2	3	2015	Abellio London	7
3	4	2015	Abellio London	9
4	5	2015	Abellio London	11

```
In [41]: df9mes.iloc[85:102].plot(x='Mes',y='Quantidade', kind='bar', label = 'Arriva Kent 1')
plt.xticks(rotation=0)
plt.title('Incidentes dos anos de 2015 e 2016')
plt.show()
df9mes.iloc[719:734].plot(x='Mes',y='Quantidade', kind='bar', label = 'Quality Line')
plt.xticks(rotation=0)
plt.title('Incidentes dos meses dos anos de 2017 e 2018')
plt.show()
df9mes.iloc[777:778].plot(x='Mes',y='Quantidade', kind='bar', label = 'Sullivan Bus')
plt.xticks(rotation=0)
```

```
plt.title('Incidentes do mês de Janeiro de 2018')
plt.show()
df9mes.iloc[778:780].plot(x='Mes',y='Quantidade', kind='bar', label = 'Sullivan Bus')
plt.xticks(rotation=0)
plt.title('Incidentes dos meses de Abril e Julho de 2016')
plt.show()
df9mes.iloc[780:790].plot(x='Mes',y='Quantidade', kind='bar', label = 'TGM GROUP',
plt.xticks(rotation=0)
plt.title('Incidentes dos meses de 2016 e 2017')
plt.show()
df9mes.iloc[835:836].plot(x='Mes',y='Quantidade', kind='bar', label = 'Uno Buses',
plt.xticks(rotation=0)
plt.title('Incidentes do mês de Maio de 2017')
plt.show()
```





## PERGUNTA 10

```
In [42]: # Pergunta 10- Qual o tipo de incidente mais comum com ciclistas?
# Resposta 10 VIA QUERY
# A colisão é o incidente mais comum entre os ciclistas
query = """
SELECT Tipo_Evento_Incidente, count(*) AS Quantidade
FROM df
WHERE Categoria_Vitima = 'Cyclist'
GROUP BY Tipo_Evento_Incidente
ORDER BY count(*) DESC
;
"""
df10 = sqldf.run(query)
df10
```

Out[42]:

	Tipo_Evento_Incidente	Quantidade
0	Collision Incident	256
1	Personal Injury	8
2	Slip Trip Fall	7
3	Onboard Injuries	4

```
In [43]: # Pergunta 10- Qual o tipo de incidente mais comum com ciclistas?
# Resposta 10 VIA GRAFICO
# A colisão é o incidente mais comum entre os ciclistas
df10.plot(x='Tipo_Evento_Incidente',y='Quantidade', kind='bar', color = '#d6bc75')
plt.title('Quantidade por tipo de incidentes envolvendo ciclistas')
plt.yticks(np.arange(0, 350, step=50))
plt.xticks(rotation=0)
plt.show()
```

