ProjetoIIEstatística

March 25, 2022

Aluna: Maria Eduarda Pereira de Souza Melo

Questão 1A: Após a importação dos dados notamos que 14 variáveis estão disponiveis, e temos um total de 91741 de registros catalogados.

[1]:
```
import pandas as pd
import numpy as np
police = pd.read_csv('/home/eduarda/Downloads/police.csv')
weather = pd.read_csv('/home/eduarda/Downloads/weather.csv')
police.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91741 entries, 0 to 91740
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   state               91741 non-null  object
 1   stop_date           91741 non-null  object
 2   stop_time           91741 non-null  object
 3   county_name         0 non-null      float64
 4   driver_gender       86536 non-null  object
 5   driver_race         86539 non-null  object
 6   violation_raw       86539 non-null  object
 7   violation           86539 non-null  object
 8   search_conducted    91741 non-null  bool
 9   search_type         3307 non-null   object
 10  stop_outcome        86539 non-null  object
 11  is_arrested         86539 non-null  object
 12  stop_duration       86539 non-null  object
 13  drugs_related_stop  91741 non-null  bool
 14  district            91741 non-null  object
dtypes: bool(2), float64(1), object(12)
memory usage: 9.3+ MB
```

Questão 1B: Contagem dos Valores ausentes.

[2]:
```
print(police.isnull().sum())
```

```
state                     0
stop_date                 0
```

```
stop_time                0
county_name          91741
driver_gender         5205
driver_race           5202
violation_raw         5202
violation             5202
search_conducted         0
search_type          88434
stop_outcome          5202
is_arrested           5202
stop_duration         5202
drugs_related_stop       0
district                 0
dtype: int64
```

Questão 1C: remoção das variáveis county_name e state

[3]:
```
police =police.drop('county_name', axis=1)
police =police.drop('state', axis=1)
police.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91741 entries, 0 to 91740
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   stop_date           91741 non-null  object
 1   stop_time           91741 non-null  object
 2   driver_gender       86536 non-null  object
 3   driver_race         86539 non-null  object
 4   violation_raw       86539 non-null  object
 5   violation           86539 non-null  object
 6   search_conducted    91741 non-null  bool
 7   search_type         3307 non-null   object
 8   stop_outcome        86539 non-null  object
 9   is_arrested         86539 non-null  object
 10  stop_duration       86539 non-null  object
 11  drugs_related_stop  91741 non-null  bool
 12  district            91741 non-null  object
dtypes: bool(2), object(11)
memory usage: 7.9+ MB
```

Questão 1D: Remover as linhas em que driver_gender é nulo. Conforme mostrado na questão 1b, a variável driver_gender possuia um total de 5205 valores nulos.

[4]:
```
police = police[police['driver_gender'].notna()]
```

Questão 1E: De acordo com a descrição dos dados Search_contucted é uma variável qualitativa. is_arrested é uma variável qualitativa. District é uma variável qualitativa. Todos descrevem uma categoria.

[5]:
```
police.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 86536 entries, 0 to 91740
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   stop_date           86536 non-null  object
 1   stop_time           86536 non-null  object
 2   driver_gender       86536 non-null  object
 3   driver_race         86536 non-null  object
 4   violation_raw       86536 non-null  object
 5   violation           86536 non-null  object
 6   search_conducted    86536 non-null  bool
 7   search_type         3307 non-null   object
 8   stop_outcome        86536 non-null  object
 9   is_arrested         86536 non-null  object
 10  stop_duration       86536 non-null  object
 11  drugs_related_stop  86536 non-null  bool
 12  district            86536 non-null  object
dtypes: bool(2), object(11)
memory usage: 8.1+ MB
```

Questão 1F: Junção das colunas stop_date e stop_time.

[6]:
```
police["stop_datetime"] = police['stop_date'] + " " + police['stop_time']
police['stop_datetime'] = pd.to_datetime(police['stop_datetime'], format="%Y/%m/
 ↪%d %H:%M")
```

Questão 1G: Transformar a coluna stop_datetime no indice do dataframe.

[7]:
```
index = pd.DatetimeIndex(police['stop_datetime'].values)
police2 = police.set_index(index)
police2
```

[7]:

	stop_date	stop_time	driver_gender	driver_race	\
2005-01-04 12:55:00	2005-01-04	12:55	M	White	
2005-01-23 23:15:00	2005-01-23	23:15	M	White	
2005-02-17 04:15:00	2005-02-17	04:15	M	White	
2005-02-20 17:15:00	2005-02-20	17:15	M	White	
2005-02-24 01:20:00	2005-02-24	01:20	F	White	
…	…	…	…	…	
2015-12-31 21:21:00	2015-12-31	21:21	F	Black	
2015-12-31 21:59:00	2015-12-31	21:59	F	White	
2015-12-31 22:04:00	2015-12-31	22:04	M	White	

```
2015-12-31 22:09:00  2015-12-31     22:09             F    Hispanic
2015-12-31 22:47:00  2015-12-31     22:47             M       White

                                    violation_raw            violation  \
2005-01-04 12:55:00  Equipment/Inspection Violation            Equipment
2005-01-23 23:15:00                        Speeding             Speeding
2005-02-17 04:15:00                        Speeding             Speeding
2005-02-20 17:15:00                Call for Service                Other
2005-02-24 01:20:00                        Speeding             Speeding
…                                             …                  …
2015-12-31 21:21:00         Other Traffic Violation     Moving violation
2015-12-31 21:59:00                        Speeding             Speeding
2015-12-31 22:04:00         Other Traffic Violation     Moving violation
2015-12-31 22:09:00  Equipment/Inspection Violation            Equipment
2015-12-31 22:47:00          Registration Violation  Registration/plates

                     search_conducted search_type   stop_outcome is_arrested  \
2005-01-04 12:55:00             False         NaN       Citation       False
2005-01-23 23:15:00             False         NaN       Citation       False
2005-02-17 04:15:00             False         NaN       Citation       False
2005-02-20 17:15:00             False         NaN  Arrest Driver        True
2005-02-24 01:20:00             False         NaN       Citation       False
…                                   …           …             …         …
2015-12-31 21:21:00             False         NaN       Citation       False
2015-12-31 21:59:00             False         NaN       Citation       False
2015-12-31 22:04:00             False         NaN       Citation       False
2015-12-31 22:09:00             False         NaN        Warning       False
2015-12-31 22:47:00             False         NaN       Citation       False

                    stop_duration  drugs_related_stop district  \
2005-01-04 12:55:00      0-15 Min               False  Zone X4
2005-01-23 23:15:00      0-15 Min               False  Zone K3
2005-02-17 04:15:00      0-15 Min               False  Zone X4
2005-02-20 17:15:00     16-30 Min               False  Zone X1
2005-02-24 01:20:00      0-15 Min               False  Zone X3
…                           …                  …      …
2015-12-31 21:21:00      0-15 Min               False  Zone K2
2015-12-31 21:59:00      0-15 Min               False  Zone K3
2015-12-31 22:04:00      0-15 Min               False  Zone X3
2015-12-31 22:09:00      0-15 Min               False  Zone K3
2015-12-31 22:47:00      0-15 Min               False  Zone X4

                          stop_datetime
2005-01-04 12:55:00 2005-01-04 12:55:00
2005-01-23 23:15:00 2005-01-23 23:15:00
2005-02-17 04:15:00 2005-02-17 04:15:00
2005-02-20 17:15:00 2005-02-20 17:15:00
```

```
2005-02-24 01:20:00 2005-02-24 01:20:00
…                                   …
2015-12-31 21:21:00 2015-12-31 21:21:00
2015-12-31 21:59:00 2015-12-31 21:59:00
2015-12-31 22:04:00 2015-12-31 22:04:00
2015-12-31 22:09:00 2015-12-31 22:09:00
2015-12-31 22:47:00 2015-12-31 22:47:00

[86536 rows x 14 columns]
```

Questão2A: Percebemos que a infração mais comum é Speeding(Excesso de velocidade) e a menos notificada é Seat belt(Sinto de segurança).

[8]:
```
police2['violation'].value_counts()
```

[8]:
```
Speeding               48423
Moving violation       16224
Equipment              10921
Other                   4409
Registration/plates     3703
Seat belt               2856
Name: violation, dtype: int64
```

Questão 2B: Frequência absoluta e relativa da distribuição conjunta das variaveis driver_gender e violation

Frequência absoluta

[9]:
```
tc1 = pd.crosstab(police2['violation'],police2['driver_gender'])
tc1.loc['Total',:]= tc1.sum(axis=0)
tc1.loc[:,'Total'] = tc1.sum(axis=1)
tc1
```

[9]:

driver_gender	F	M	Total
violation			
Equipment	2501.0	8420.0	10921.0
Moving violation	3286.0	12938.0	16224.0
Other	707.0	3702.0	4409.0
Registration/plates	1056.0	2647.0	3703.0
Seat belt	578.0	2278.0	2856.0
Speeding	15646.0	32777.0	48423.0
Total	23774.0	62762.0	86536.0

Frequência relativa

[10]:
```
tc2 = pd.crosstab(police2['violation'],police2['driver_gender'],␣
↪normalize=True)*100
tc2.loc['Total',:]= tc2.sum(axis=0)
```

```
tc2.loc[:,'Total'] = tc2.sum(axis=1)
tc2.round(2)
```

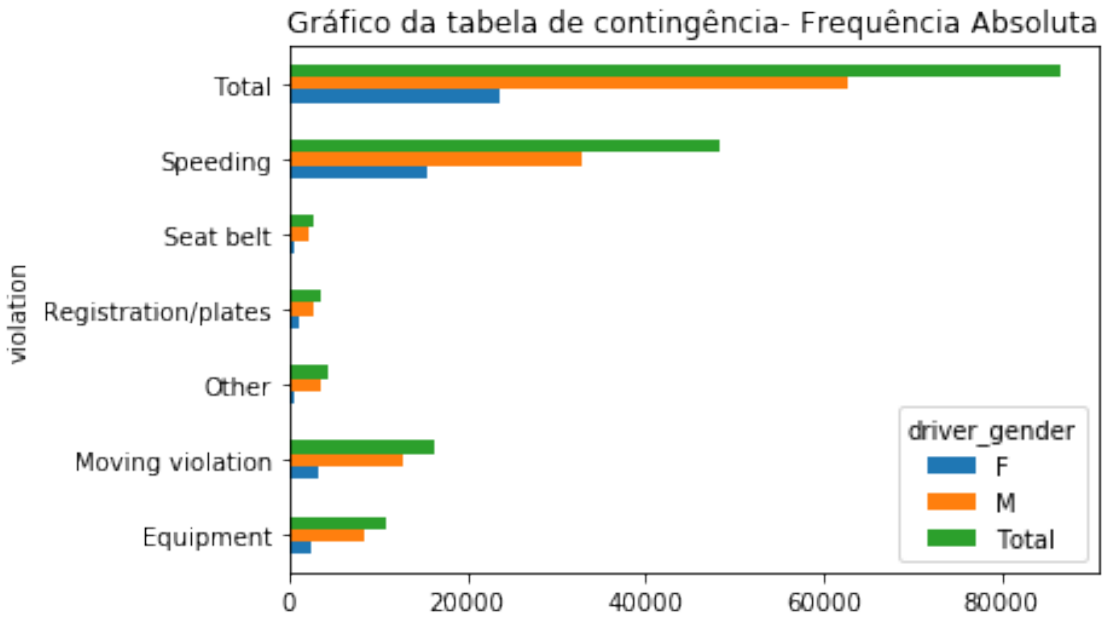
[10]:

driver_gender	F	M	Total
violation			
Equipment	2.89	9.73	12.62
Moving violation	3.80	14.95	18.75
Other	0.82	4.28	5.09
Registration/plates	1.22	3.06	4.28
Seat belt	0.67	2.63	3.30
Speeding	18.08	37.88	55.96
Total	27.47	72.53	100.00

Questão 2C: Gráfico de barras ilustrando a tabela de contigência da frequência absoluta

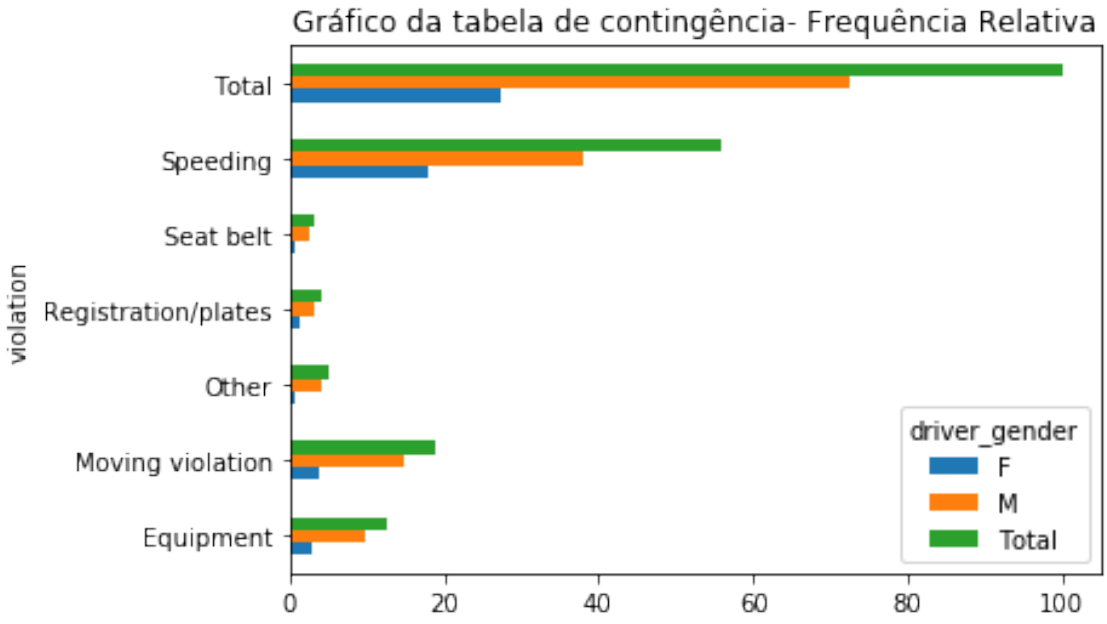
[11]:
```
%matplotlib inline
tc1.plot.barh(title='Gráfico da tabela de contingência- Frequência Absoluta')
```

[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7f95483ac850>



[12]:
```
%matplotlib inline
tc2.plot.barh(title='Gráfico da tabela de contingência- Frequência Relativa')
```

[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f954631b8e0>



Questão 2D: Notamos que os homens recebem bem mais multas que as mulheres.

[13]:
```
tc3 = pd.crosstab(police2['stop_outcome'],police2['driver_gender'],␣
↪normalize=True)*100
tc3.round(2)
```

[13]:
```
driver_gender          F      M
stop_outcome
Arrest Driver       0.64   2.52
Arrest Passenger    0.13   0.27
Citation           24.56  64.53
N/D                 0.19   0.51
No Action           0.23   0.49
Warning             1.72   4.22
```

Questão 2E:

[14]:
```
tc3 = pd.crosstab(police2['search_conducted'], police2['driver_gender'],␣
↪normalize=True)*100
tc3.round(2)
```

[14]:
```
driver_gender          F      M
search_conducted
False              26.95  69.23
True                0.53   3.29
```

Porcentagem de todas as paradas

[15]:
```
X = police2['search_conducted'].value_counts(normalize=True)*100
X.round(2)
```

[15]:
```
False    96.18
True      3.82
Name: search_conducted, dtype: float64
```

Questão 2F:

[16]:
```
police2.groupby(['driver_gender']).sum()
```

[16]:

driver_gender	search_conducted	drugs_related_stop
F	456	107
M	2851	755

Questão 2G: Relação entre o tipo de infração com os gêneros e o revistamento dos carros.

[17]:
```
tc4= pd.crosstab(police2['violation'],␣
 ↪[police2['search_conducted'],police2['driver_gender']], normalize=True)*100
tc4.round(2)
```

[17]:

search_conducted	False		True	
driver_gender	F	M	F	M
violation				
Equipment	2.77	9.03	0.12	0.70
Moving violation	3.65	14.03	0.15	0.92
Other	0.78	4.08	0.03	0.20
Registration/plates	1.15	2.73	0.07	0.33
Seat belt	0.66	2.54	0.01	0.09
Speeding	17.93	36.82	0.15	1.06

Notamos que a taxa de revistamento dos carros onde os homens estão presentes é bem superior ao das mulheres. Dessa forma, concluimos que homens e mulheres não são revistados com a mesma taxa para cada infração.

Questão 3A: Taxa de prisão -> Parada que leva a prisão.

[18]:
```
taxa_em_todas_paradas = police2.groupby(['is_arrested']).mean()*100
taxa_em_todas_paradas.round(2)
```

[18]:

is_arrested	search_conducted	drugs_related_stop
False	2.6	0.75
True	37.0	7.57

Questão 3B: Taxa de prisão por hora usando o atributo de hora do índice. Lembrando que 0h é meia noite e 12h é meio dia.

[19]:
```
hourly_arrest_rate =police2.groupby(by = [police2.index.map(lambda x: (x.
 ↪hour))]).mean()*100
hourly_arrest_rate.round(2)
```

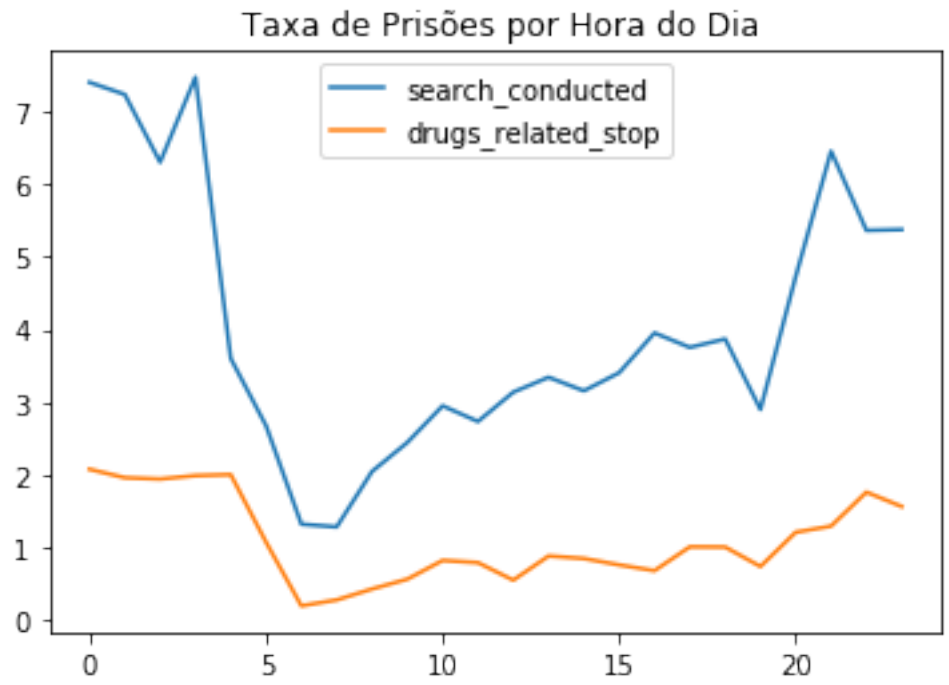
[19]:

	search_conducted	drugs_related_stop
0	7.40	2.07
1	7.24	1.96
2	6.30	1.94
3	7.47	1.99
4	3.60	2.00
5	2.67	1.07
6	1.32	0.19
7	1.28	0.27
8	2.04	0.42
9	2.44	0.56
10	2.95	0.82
11	2.73	0.79
12	3.14	0.55
13	3.34	0.88
14	3.15	0.84
15	3.40	0.76
16	3.95	0.68
17	3.75	1.01
18	3.87	1.00
19	2.90	0.73
20	4.73	1.21
21	6.45	1.29
22	5.36	1.76
23	5.38	1.56

Questão 3C: Gráfico da variável hourly_arrest_rate

[20]:
```
hourly_arrest_rate.plot(title=' Taxa de Prisões por Hora do Dia')
```

[20]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9546266820>
```



Questão 3D e 3E:

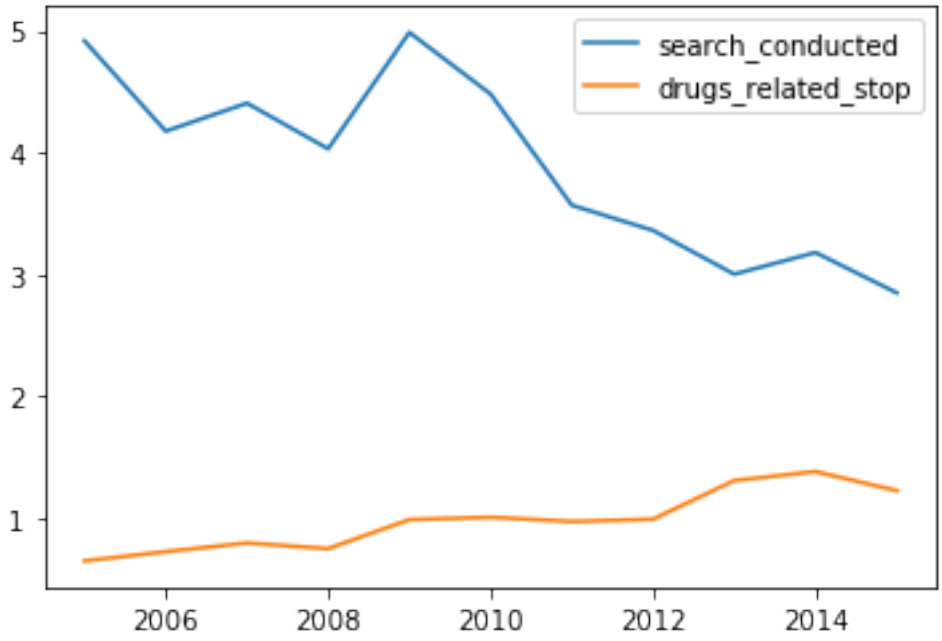
[21]:
```
Annual = police2.groupby(by = [police2.index.map(lambda x: (x.year))]).
↪mean()*100
Annual.round(2)
Annual_search_rate= Annual['search_conducted']
Annual_drug_rate= Annual['drugs_related_stop']
Annual
```

[21]:

	search_conducted	drugs_related_stop
2005	4.916701	0.650142
2006	4.175780	0.725790
2007	4.405579	0.796989
2008	4.030989	0.750514
2009	4.986072	0.988858
2010	4.480538	1.008121
2011	3.568185	0.973141
2012	3.361587	0.992102
2013	3.002248	1.309351
2014	3.180090	1.382648
2015	2.849709	1.226614

[22]:
```
Annual.plot()
```

[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95461cb400>
```



Ao analizarmos o gráfico percebemos que as interrupções relacionadas à drogas estão se tornando mais comuns com o tempo. Observamos também que a hipotese apresentada na alternativa E é falsa, pois à medida que as interrupções diminuem as paradas relacionadas a drogas aumentam.

Questão 3F:

[23]:
```
tc5 = pd.crosstab(police2['district'], police2['violation'])
tc5
```

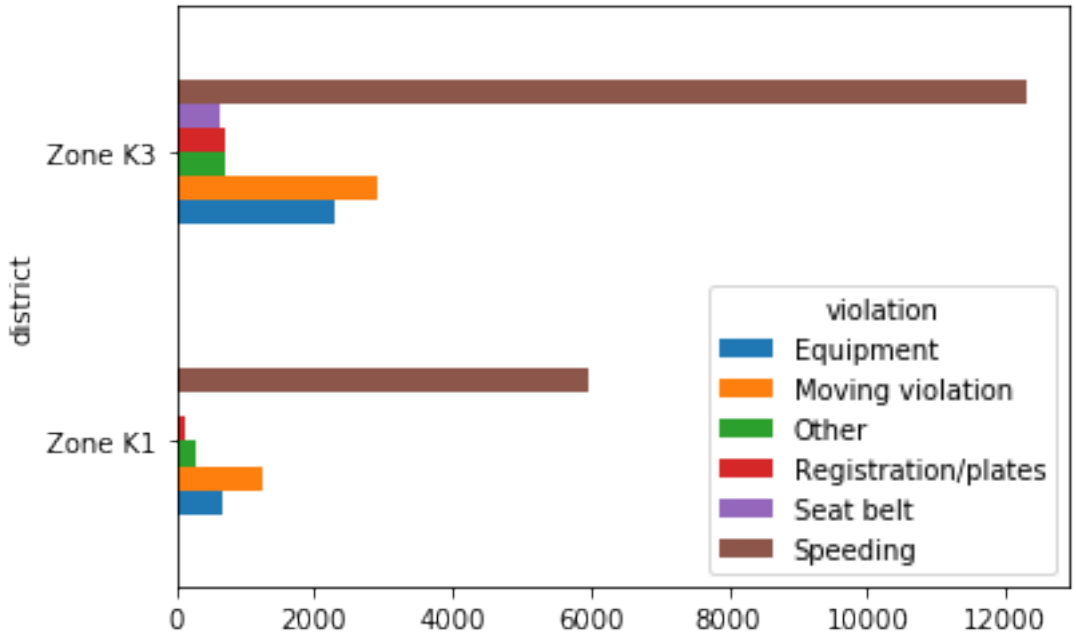
[23]:
```
violation  Equipment  Moving violation  Other  Registration/plates  Seat belt  \
district
Zone K1          672              1254    290                  120          0
Zone K2         2061              2962    942                  768        481
Zone K3         2302              2898    705                  695        638
Zone X1          296               671    143                   38         74
Zone X3         2049              3086    769                  671        820
Zone X4         3541              5353   1560                 1411        843

violation  Speeding
district
Zone K1        5960
Zone K2       10448
Zone K3       12322
Zone X1        1119
Zone X3        8779
Zone X4        9795
```

[24]:
```
filtro = tc5.query("district== 'Zone K1'or district =='Zone K3'")
filtro
```

[24]:
```
violation  Equipment  Moving violation  Other  Registration/plates  Seat belt  \
district
Zone K1          672              1254    290                  120          0
Zone K3         2302              2898    705                  695        638

violation  Speeding
district
Zone K1        5960
Zone K3       12322
```

[25]:
```
%matplotlib inline
filtro.plot.barh()
```

[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95461a39a0>
```



Percebemos que a quantidade de ocorrências da Zona 1 são bem menores que as acorrências da Zona 2.

Questão 4A:

[26]:
```
temperaturas = weather[['TMIN', 'TAVG', 'TMAX']]
temperaturas
```

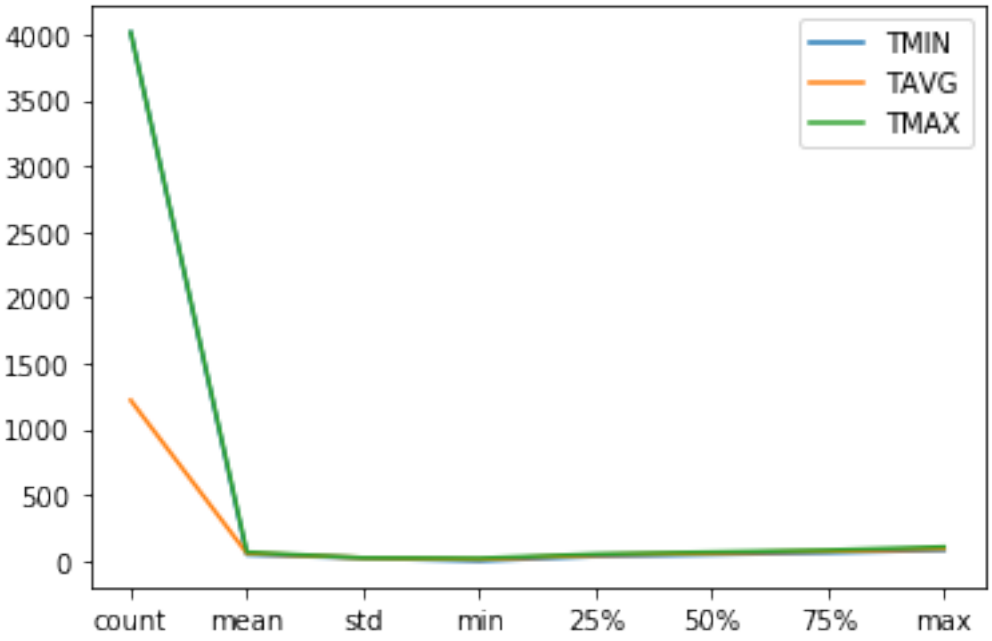
[26]:
```
      TMIN  TAVG  TMAX
0       35  44.0    53
1       28  36.0    44
2       44  49.0    53
3       39  42.0    45
4       28  36.0    43
…     …   …   …
4012    44  51.0    61
4013    30  40.0    44
4014    28  33.0    40
4015    27  30.0    35
4016    35  39.0    50

[4017 rows x 3 columns]
```

[27]:
```
t =temperaturas.describe()
t.round(2)
```

[27]:
```
           TMIN     TAVG     TMAX
count  4017.00  1217.00  4017.00
mean     43.48    52.49    61.27
std      17.02    17.83    18.20
min      -5.00     6.00    15.00
25%      30.00    39.00    47.00
50%      44.00    54.00    62.00
75%      58.00    68.00    77.00
max      77.00    86.00   102.00
```

[28]:
```
%matplotlib inline
t.plot()
```

[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9546129fa0>
```



Questão 4B:

[29]:
```
tdiff = weather['TMAX'] - weather['TMIN']
tdiff
```

[29]:
```
0       18
1       16
2        9
3        6
4       15
        ..
4012    17
4013    14
4014    12
4015     8
4016    15
Length: 4017, dtype: int64
```

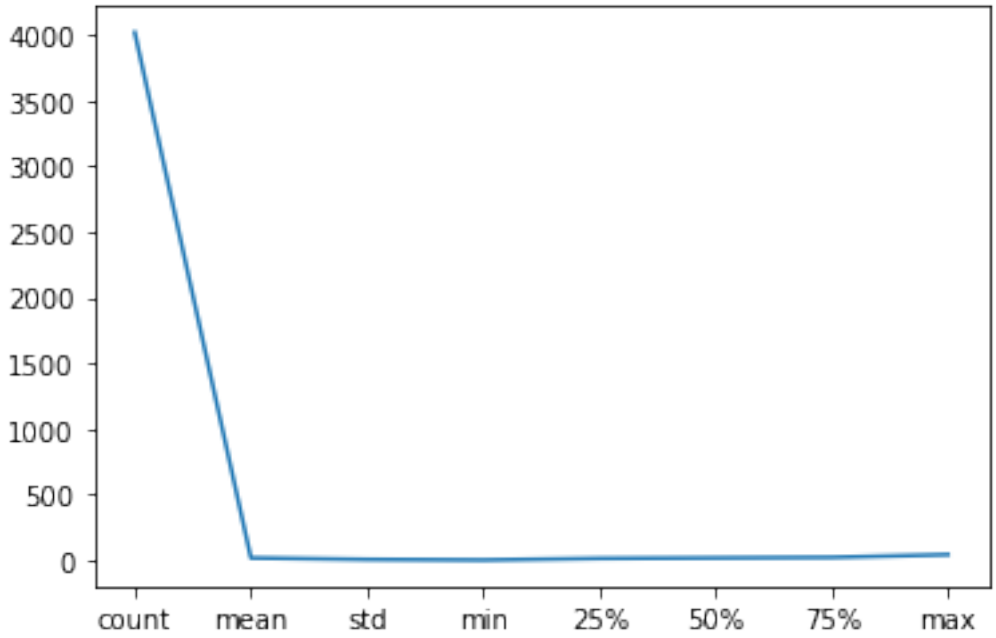
[30]:
```
descrição = tdiff.describe()
descrição
```

[30]:
```
count    4017.000000
mean       17.784167
std         6.350720
min         2.000000
25%        14.000000
```

```
50%         18.000000
75%         22.000000
max         43.000000
dtype: float64
```

[31]:
```
descrição.plot()
```

[31]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95460443d0>
```



Questão 4C:

[32]:
```
police3 = police2.rename_axis('reset_index)').reset_index()
police3
```

[32]:
```
             reset_index)   stop_date stop_time driver_gender driver_race  \
0     2005-01-04 12:55:00  2005-01-04     12:55             M       White
1     2005-01-23 23:15:00  2005-01-23     23:15             M       White
2     2005-02-17 04:15:00  2005-02-17     04:15             M       White
3     2005-02-20 17:15:00  2005-02-20     17:15             M       White
4     2005-02-24 01:20:00  2005-02-24     01:20             F       White
…                     …          …         …              …          …
86531 2015-12-31 21:21:00  2015-12-31     21:21             F       Black
86532 2015-12-31 21:59:00  2015-12-31     21:59             F       White
86533 2015-12-31 22:04:00  2015-12-31     22:04             M       White
86534 2015-12-31 22:09:00  2015-12-31     22:09             F    Hispanic
86535 2015-12-31 22:47:00  2015-12-31     22:47             M       White
```

```
                        violation_raw            violation  search_conducted  \
0      Equipment/Inspection Violation            Equipment             False
1                            Speeding             Speeding             False
2                            Speeding             Speeding             False
3                   Call for Service                Other             False
4                            Speeding             Speeding             False
…                                  …                   …               …
86531         Other Traffic Violation     Moving violation             False
86532                        Speeding             Speeding             False
86533         Other Traffic Violation     Moving violation             False
86534  Equipment/Inspection Violation            Equipment             False
86535          Registration Violation  Registration/plates             False

      search_type   stop_outcome is_arrested stop_duration  \
0             NaN       Citation       False      0-15 Min
1             NaN       Citation       False      0-15 Min
2             NaN       Citation       False      0-15 Min
3             NaN  Arrest Driver        True     16-30 Min
4             NaN       Citation       False      0-15 Min
…           …             …         …           …
86531         NaN       Citation       False      0-15 Min
86532         NaN       Citation       False      0-15 Min
86533         NaN       Citation       False      0-15 Min
86534         NaN        Warning       False      0-15 Min
86535         NaN       Citation       False      0-15 Min

       drugs_related_stop district       stop_datetime
0                   False  Zone X4 2005-01-04 12:55:00
1                   False  Zone K3 2005-01-23 23:15:00
2                   False  Zone X4 2005-02-17 04:15:00
3                   False  Zone X1 2005-02-20 17:15:00
4                   False  Zone X3 2005-02-24 01:20:00
…                   …      …                   …
86531               False  Zone K2 2015-12-31 21:21:00
86532               False  Zone K3 2015-12-31 21:59:00
86533               False  Zone X3 2015-12-31 22:04:00
86534               False  Zone K3 2015-12-31 22:09:00
86535               False  Zone X4 2015-12-31 22:47:00

[86536 rows x 15 columns]
```

[33]:
```
weather1 = weather[['DATE']]
po = police3['stop_date']
novodf = pd.concat([weather1, po],axis=1)
index = pd.DatetimeIndex(police['stop_datetime'].values)
novodf = novodf.set_index(index)
novodf
```

[33]:
```
                            DATE   stop_date
2005-01-04 12:55:00  2005-01-01  2005-01-04
2005-01-23 23:15:00  2005-01-02  2005-01-23
2005-02-17 04:15:00  2005-01-03  2005-02-17
2005-02-20 17:15:00  2005-01-04  2005-02-20
2005-02-24 01:20:00  2005-01-05  2005-02-24
…                           …         …
2015-12-31 21:21:00         NaN  2015-12-31
2015-12-31 21:59:00         NaN  2015-12-31
2015-12-31 22:04:00         NaN  2015-12-31
2015-12-31 22:09:00         NaN  2015-12-31
2015-12-31 22:47:00         NaN  2015-12-31

[86536 rows x 2 columns]
```

Questão 4E:

Faça a contagem dos valores nulos

[34]:
```
novodf.isnull().sum()
```

[34]:
```
DATE         82519
stop_date        0
dtype: int64
```

Faça a distruição de frequências do dataframe.

[35]:
```
novodf.value_counts()
```

[35]:
```
DATE        stop_date
2005-01-01  2005-01-04    1
2012-05-07  2006-01-08    1
2012-04-24  2006-01-08    1
2012-04-25  2006-01-08    1
2012-04-26  2006-01-08    1
                         ..
2008-09-06  2005-11-16    1
2008-09-07  2005-11-16    1
2008-09-08  2005-11-16    1
2008-09-09  2005-11-16    1
2015-12-31  2006-02-22    1
Length: 4017, dtype: int64
```