PROJETO1

January 6, 2022

ESTATÍSTICA APLICADA

ENGENHARIA DE COMPUTAÇÃO

PROJETO 1

ALUNO: Maria Eduarda Pereira de Souza Melo

QUESTÃO 1: No código abaixo utilizamos o read_csv para abrir os arquivos anteriormente baixados no site disponibilizado.

[1]:
```
import pandas as pd
import numpy as np
df2018 = pd.read_csv('/home/eduarda/Downloads/dados2018.csv',
                 encoding='latin1',
                     sep=';')
df2019 = pd.read_csv('/home/eduarda/Downloads/datatran2019.csv',
                     encoding='latin1',
                      sep=';')
df2020 = pd.read_csv('/home/eduarda/Downloads/datatran2020.csv',
                     encoding='latin1',
                     sep=';')
```

QUESTÃO2A: Para visualizarmos a quantidade de acidentes que ocorreram em cada ano, utilizamos a função len(), já que cada linha do dataframe corresponde a 1 acidente.

[2]:
```
print(len(df2018),'acidentes de trânsito ocorreram em 2018')
```

```
69295 acidentes de trânsito ocorreram em 2018
```

[3]:
```
print(len(df2019),'acidentes de trânsito ocorreram em 2019')
```

```
67446 acidentes de trânsito ocorreram em 2019
```

[4]:
```
print(len(df2020),'acidentes de trânsito ocorreram em 2020')
```

```
63548 acidentes de trânsito ocorreram em 2020
```

QUESTAO2B:

[5]:
```
print("O conjunto de dados de 2018 possui", len(df2018.columns),'variáveis')
```

```
O conjunto de dados de 2018 possui 30 variáveis
```

[6]:
```
print("O conjunto de dados de 2019 possui", len(df2019.columns),'variáveis')
```

```
O conjunto de dados de 2019 possui 30 variáveis
```

[7]:
```
print("O conjunto de dados de 2020 possui", len(df2020.columns),'variáveis')
```

```
O conjunto de dados de 2020 possui 30 variáveis
```

QUESTÃO2C: Primeiramente, juntamos todos os dataframes em um só, para realizar tal ação utilizamos o *concat()*. Na sequência, criamos um novo DataFrame e passamos como parâmetro a quantidade de ocorrências de acidentes por municipio, através do *value_counts()*. Após esses passos, filtramos as 5 primeiras linhas do dataframe, já que o mesmo já se encontrava ordenado. Dessa forma, visualizamos as 5 cidades brasileiras com maior ocorrência de acidentes.

[8]:
```
juncao = pd.concat([df2018,df2019,df2020])
cidades = pd.DataFrame(juncao["municipio"].value_counts())
cidades[0:5]
```

[8]:

	municipio
CURITIBA	3063
BRASILIA	2997
SAO JOSE	2322
GUARULHOS	2111
PALHOCA	1887

QUESTÃO2D: Para calcular a quantidade de acidentes com feridos graves na paraíba, realizamos um filtro, no qual o mesmo seleciona as ocorrências onde a quantidade de feridos graves foi maior que zero e que aconteceram no estado na paraíba.

[9]:
```
gravespb = df2019[(df2019["feridos_graves"]>0) & (df2019["uf"]=='PB')]
print('A quantidade de acidentes com feridos graves na Paraíba em 2019 foi␣
 ↪de',len(gravespb))
```

```
A quantidade de acidentes com feridos graves na Paraíba em 2019 foi de 414
```

QUESTÃO3A: Agrupamos, inicialmente, a quantidade de ocorrências de acidentes por estado, como mostrado em *juncao["uf"].value_counts()*, e passamos essas ocorrências como parâmetro para a criação do DataFrame estados. Em seguida, gravamos esse dataframe em um arquivo *.csv* denominado de Ranking_acidentes.csv. O *index=True* é utilizado para que os índices do Dataframe sejam mantidos, já que cada índice corresponde as siglas de um estado.

[10]:
```
estados = pd.DataFrame(juncao["uf"].value_counts())
estados.to_csv('/home/eduarda/Downloads/Ranking_acidentes.csv',␣
 ↪encoding='utf-8', index=True, sep=',')
estados
```

[10]:

	uf
MG	26160
SC	24145
PR	22850
RJ	13417
RS	13216
SP	12936
BA	10482
GO	10046
ES	8018
PE	8011
MT	6963
CE	4962
RO	4648
PB	4553
MS	4505
RN	4190
PI	3920
MA	3501
PA	3040
DF	2997
AL	1948
SE	1746
TO	1704
AC	789
RR	723
AP	456
AM	363

QUESTÃO3B: Para visualizarmos o ranking de acidentes por dia da semana, agrupamos a quantidade de ocorrências de acidentes em cada dia, mostrado em *juncao["dia_semana"].value_counts()*. E passamos tal agrupamento como parametro para a criação do dataframe. Em seguida, gravamos esse dataframe em um arquivo .csv denominado de Ranking_semana.csv. O *index=True* é utilizado para que os índices do Dataframe sejam mantidos, já que cada índice corresponde ao nome de um dia da semana.

[11]:
```
dataframe = pd.DataFrame(juncao["dia_semana"].value_counts())
dataframe.to_csv('/home/eduarda/Downloads/Ranking_semana.csv',␣
↪encoding='utf-8', index=True, sep=',')
dataframe
```

[11]:

	dia_semana
sábado	33005
domingo	33001
sexta-feira	30777
segunda-feira	27075
quinta-feira	26229

```
quarta-feira        25353
terça-feira         24849
```

QUESTÃO4A: Para visualizarmos a causa mais frequente de acidentes registrados nos últimos três anos, agrupamos a quantidade de ocorrências de cada causa, mostrado em *juncao["causa_acidente"].value_counts()*. Como o dataframe já está organizado em ordem decrescente, a maior ocorrência estará na primeira linha, por isso a utilização do *maisfrequente[0:1]*.

[12]:
```
maisfrequente = pd.DataFrame(juncao["causa_acidente"].value_counts())
maisfrequente[0:1]
```

[12]:

	causa_acidente
Falta de Atenção à Condução	73231

Para visualizarmos as causas de acidente mais raras de ocorrer, realizamos a mesma lógica de agrupamento do item acima, apenas especificamos que as causas mais raras de acontecer são aquelas que tiveram apenas uma ocorrência.

[13]:
```
menosfrequente=pd.DataFrame(juncao["causa_acidente"].value_counts())
menosfrequente =menosfrequente[menosfrequente["causa_acidente"]==1]
menosfrequente
```

[13]:

	causa_acidente
Acumulo de areia ou detritos sobre o pavimento	1
Pista esburacada	1
Estacionar ou parar em local proibido	1
Fumaça	1
Pedestre cruzava a pista fora da faixa	1
Acostamento em desnível	1
Condutor usando celular	1
Frear bruscamente	1
Ausência de sinalização	1
Acumulo de óleo sobre o pavimento	1
Ingestão de álcool ou de substâncias psicoativa…	1
Curva acentuada	1

QUESTÃO4B: Para calcularmos a proporção de pessoas ilesas e de feridos graves por mês em cada um dos anos, começamos verificando se a variável *data_inversa* está no formato *datetime64*. Após isso, agrupamos e somamos os dados referentes a cada mês, passando o *data_inversa* como chave e espeficando que o agrupamento será por mês em *freq='M'*. Em seguida, criamos duas listas, *di* para o cálculo de proporção de ilesos, e *dg* para o cálculo de proporção de feridos graves. O *FOR* é utilizado para iterar sobre cada linha do nosso dataframe, adicionando em di a divisão entre o número de ilesos pelo total pessoas, e adicionando em dg a divisão entre o número de feridos pelo total de pessoas. E para concluir, adicionamos duas novas colunas em nosso dataframe, uma denominada *Proporção_ilesos* para adicionar os valores de di, e a outra denominada *Proporção_feridosgraves* para adicionar os valores de dg. Tais passos também são aplicados para o cálculo dos anos de 2019 e 2020.

proporção = numero de ilesos ou feridos graves/ número total de pessoas envolvidas no acidente

[14]:
```
df2018['data_inversa'] = df2018['data_inversa'].astype('datetime64')
novodf = df2018.groupby([pd.Grouper(key= "data_inversa", freq='M')]).sum().
 ↪reset_index()
di =[]
dg =[]
for indice,linha in novodf.iterrows():
    di.append(linha["ilesos"]/linha["pessoas"])
    dg.append(linha["feridos_graves"]/linha["pessoas"])
novodf["Proporção_ilesos"] = di
novodf["Proporção_feridosgraves"] = dg
novodf
```

[14]:

	data_inversa	id	br	pessoas	mortos	feridos_leves	\
0	2018-01-31	7.276468e+08	1456066.0	16665	456	5360	
1	2018-02-28	6.946953e+08	1351785.0	14047	394	4627	
2	2018-03-31	8.001268e+08	1447465.0	15110	426	5006	
3	2018-04-30	7.105128e+08	1190396.0	13230	439	4914	
4	2018-05-31	6.523576e+08	1017864.0	11489	389	4296	
5	2018-06-30	7.870912e+08	1212209.0	12870	444	4685	
6	2018-07-31	8.024132e+08	1168015.0	13618	490	4928	
7	2018-08-31	8.386776e+08	1167542.0	12776	454	4718	
8	2018-09-30	8.731549e+08	1177046.0	13027	492	4812	
9	2018-10-31	9.042249e+08	1171908.0	12890	376	4802	
10	2018-11-30	9.438812e+08	1164979.0	13197	426	4834	
11	2018-12-31	1.140497e+09	1370113.0	15883	485	5959	

	feridos_graves	ilesos	ignorados	feridos	veiculos	Proporção_ilesos	\
0	1510	8554	785	6870	11225	0.513291	
1	1295	6999	732	5922	9950	0.498256	
2	1539	7276	863	6545	10887	0.481535	
3	1492	5691	694	6406	9240	0.430159	
4	1315	4851	638	5611	7976	0.422230	
5	1498	5471	772	6183	9254	0.425097	
6	1502	5958	740	6430	9161	0.437509	
7	1359	5456	789	6077	9120	0.427051	
8	1504	5480	739	6316	9044	0.420665	
9	1452	5545	715	6254	8930	0.430178	
10	1523	5719	695	6357	9126	0.433356	
11	1717	6876	846	7676	10560	0.432916	

	Proporção_feridosgraves
0	0.090609
1	0.092191
2	0.101853
3	0.112774
4	0.114457
5	0.116395

```
6                   0.110295
7                   0.106371
8                   0.115453
9                   0.112645
10                  0.115405
11                  0.108103
```

[15]:
```
df2019['data_inversa'] =  df2019['data_inversa'].astype('datetime64')
novodf = df2019.groupby([pd.Grouper(key= "data_inversa", freq='M')]).sum().
 ↪reset_index()
di =[]
dg =[]
for indice,linha in novodf.iterrows():
    di.append(linha["ilesos"]/linha["pessoas"])
    dg.append(linha["feridos_graves"]/linha["pessoas"])
novodf["Proporção_ilesos"] = di
novodf["Proporção_feridosgraves"] = dg
novodf
```

[15]:

	data_inversa	id	br	pessoas	mortos	feridos_leves
0	2019-01-31	9.540553e+08	1080718.0	13111	406	5015
1	2019-02-28	9.533792e+08	1064183.0	11799	394	4542
2	2019-03-31	1.106939e+09	1195550.0	13569	406	4979
3	2019-04-30	1.109989e+09	1156791.0	12815	392	4951
4	2019-05-31	1.150753e+09	1186990.0	12795	471	4716
5	2019-06-30	1.208621e+09	1205969.0	13411	458	4829
6	2019-07-31	1.291849e+09	1232678.0	14331	487	5081
7	2019-08-31	1.309215e+09	1264000.0	13421	455	5051
8	2019-09-30	1.386461e+09	1256302.0	13682	449	5145
9	2019-10-31	1.430256e+09	1278207.0	13931	435	5220
10	2019-11-30	1.450134e+09	1255037.0	13823	479	5123
11	2019-12-31	1.600099e+09	1327054.0	15585	501	5848

	feridos_graves	ilesos	ignorados	feridos	veiculos	Proporção_ilesos
0	1485	5562	643	6500	8452	0.424224
1	1260	5004	599	5802	8116	0.424104
2	1555	5878	751	6534	9327	0.433193
3	1358	5327	787	6309	8964	0.415685
4	1545	5312	751	6261	9154	0.415162
5	1631	5683	810	6460	9326	0.423757
6	1672	6260	831	6753	9785	0.436815
7	1566	5544	805	6617	9512	0.413084
8	1514	5761	813	6659	9677	0.421064
9	1669	5836	771	6889	9709	0.418922
10	1535	5886	800	6658	9649	0.425812
11	1783	6581	872	7631	10380	0.422265

```
    Proporção_feridosgraves
0                  0.113264
1                  0.106789
2                  0.114599
3                  0.105970
4                  0.120750
5                  0.121617
6                  0.116670
7                  0.116683
8                  0.110656
9                  0.119805
10                 0.111047
11                 0.114405
```

[16]:
```
df2020['data_inversa'] =  df2020['data_inversa'].astype('datetime64')
novodf = df2020.groupby([pd.Grouper(key= "data_inversa", freq='M')]).sum().
 ↪reset_index()
di =[]
dg =[]
for indice,linha in novodf.iterrows():
    di.append(linha["ilesos"]/linha["pessoas"])
    dg.append(linha["feridos_graves"]/linha["pessoas"])
novodf["Proporção_ilesos"] = di
novodf["Proporção_feridosgraves"] = dg
novodf
```

[16]:

	data_inversa	id	br	pessoas	mortos	feridos_leves	\
0	2020-01-31	1.450788e+09	1169028.0	14070	410	5528	
1	2020-02-29	1.475943e+09	1161561.0	13183	386	5003	
2	2020-03-31	1.322712e+09	1026868.0	10775	391	4090	
3	2020-04-30	1.090075e+09	844320.0	8385	354	3299	
4	2020-05-31	1.329357e+09	999775.0	10074	387	3712	
5	2020-06-30	1.377019e+09	995995.0	10454	359	3660	
6	2020-07-31	1.518552e+09	1079402.0	11628	455	4147	
7	2020-08-31	1.706336e+09	1186047.0	12881	455	4550	
8	2020-09-30	1.755905e+09	1203360.0	13187	499	4719	
9	2020-10-31	1.930380e+09	1305478.0	14501	487	5323	
10	2020-11-30	1.850969e+09	1205130.0	13490	526	4865	
11	2020-12-31	2.038252e+09	1289448.0	15050	582	5480	

	feridos_graves	ilesos	ignorados	feridos	veiculos	Proporção_ilesos	\
0	1512	5847	773	7040	9035	0.415565	
1	1468	5537	789	6471	8952	0.420011	
2	1213	4321	760	5303	7701	0.401021	
3	989	3155	588	4288	6005	0.376267	
4	1168	4043	764	4880	7412	0.401330	
5	1212	4425	798	4872	7718	0.423283	

6	1263	4934	829	5410	8403	0.424321
7	1535	5454	887	6085	9313	0.423414
8	1548	5486	935	6267	9444	0.416016
9	1736	5960	995	7059	10081	0.411006
10	1624	5491	984	6489	9597	0.407042
11	1836	6205	947	7316	10186	0.412292

	Proporção_feridosgraves
0	0.107463
1	0.111356
2	0.112575
3	0.117949
4	0.115942
5	0.115936
6	0.108617
7	0.119168
8	0.117388
9	0.119716
10	0.120385
11	0.121993

QUESTÃO4C: Após verificarmos a quantidade de acidentes que ocorreu em cada ano, percebemos que em 2020 houve uma queda na quantidade do mesmo. No entanto, creio que seria necessário a análise de outros fatores para concluirmos que a pandemia foi a causadora dessa redução.

[17]:
```
obj = {'Nº Acidentes': [69295, 67446, 63548]}
Conclusao = pd.DataFrame(obj, index=[2018,2019,2020])
Conclusao
```

[17]:

	Nº Acidentes
2018	69295
2019	67446
2020	63548