Lista 3 MAE0399

Prof. Fábio Prates Machado

Gustavo Gomes, NUSP 12557438

Configurando o Ambiente

Primeiro vamos importar a biblioteca Pandas do Python e abrir o arquivo referente ao mês de julho de 2018 do Citibike.

```
    import pandas as pd
    data = pd.read_csv('/content/drive/MyDrive/201807-citibike-
        tripdata.csv.zip')
    data.head()
```

Questão 1

Aqui, calcula-se os quantis 1%,25%,50%,75%,99% da variável tripduration e armazena os valores na tabela 1. Logo após, a partir do drop excluimos os valores de tripduration que estão fora do intervalo entre os quartis 1% e 99%.

```
    tabela = pd.DataFrame()
valores_quantis = []

for i in [0.01,0.25,0.5,0.75,0.99]:
  valores_quantis.append(data['tripduration'].quantile(q = i))

tabela.insert(0,'Quantil',['1%','25%','50%','75%','99%'])
tabela.insert(1,'Valor',valores_quantis)
#exclui os valores fora do intervalo entre o quantil 1% e o quantil
    99%
data.drop(data.loc[(data['tripduration'] > valores_quantis[4]) | (
    data['tripduration'] < valores_quantis[0]) ].index, inplace=
    True)
#tabelas com os quantis
tabela
```

	Quantil	Valor
0	1%	103.0
1	25%	381.0
2	50%	656.0
3	75%	1161.0
4	99%	4157.0

Questão 2

Aqui vamos calcular quantos ids diferentes de bike temos no dataset a partir do comando unique e depois calculamos o tamanho da lists que este comando retornará a partir do len. Executando, obtemos 10.667.

```
#todas as bicicletas usadas pelo menos uma vez
len(data['bikeid'].unique())
```

Questão 3

Aqui, criamos uma nova variável 'day in month' que marcará o dia no mes que a bike foi usada. Depois contabilizamos a quantidade de bikes cujos dias de utilização distintos foram 31 (total de dias do mês de julho), obtendo 292.

```
#variavel que marca o dia no mes do uso
data['day_in_month'] = data['starttime'].str[5:10]

bikes_mais_usadas = []

for i in data['bikeid'].unique():
  if len(data[data['bikeid'] == i]['day_in_month'].unique()) == 31:
    bikes_mais_usadas.append(i)
#total de bicicletas usadas todos os dias no mes
len(bikes_mais_usadas)
```

Questão 4

Aqui, obviamente nossa primeira suspeita é que o menor número de dias de utilização seja 1. O código abaixo percorre os ids das bikes usadas e imprime a quantidade delas cujos dias distintos de utilização contabilizam apenas 1. Obtemos, 100 bikes com 1 dia de utilização.

```
#todas as bicicletas usadas pelo menos uma vez
usadas = data['bikeid'].unique()
uma_aparicao = []
for i in range(len(usadas)):
  #imprime os id que apareceram apenas 1 vez
   if len(data[data['bikeid'] == usadas[i]]['day_in_month'].unique()
       ) == 1:
     uma_aparicao.append(i)
#total de bicicletas usadas apenas uma vez no mes
len(uma_aparicao)
```

Questão 5

Aqui nós agrupamos as bikes por id e somamos as tridurations correspondentes e ordenamos em ordem decrescente. Logo depois, pegamos o top 10 da tabela, que corresponde as 10 bikes mais usadas. (tabela 2)

```
#agrupa as bikes por id e soma os trips durations
top_10_mais_usadas = data[['tripduration','bikeid']].groupby(by = [
    'bikeid'], as_index = False).sum()
#ordena conforme o tempo total decrescente
top_10_mais_usadas.sort_values(by = ['tripduration'],ascending =
    False).head(10)
```

Questão 6

Aqui eu reordenei a tabela em ordem crescente, pois assim os 10 primeiros dela irão corresponder as 10 bikes menos utlizadas. (tabela 3)

```
#ordena conforme o tempo total crescente
top_10_mais_usadas.sort_values(by = ['tripduration'],ascending =
    True).head(10)
```

Questão 7

Aqui considerei que se uma bike foi utilizada pelo menos uma vez em um dia, então ela estava pronta para uso todo aquele dia, sem estar em manutenção.

Bom, primeiramente agrupamos as bikes por id e somamos os trip durations, após contabilizamos os segundos disponiveis a partir da quantidade de dias que uma determinada bike foi usada pelo menos uma vez.

Depois fazemos a razão entre o tempo total de viagem e o tempo total disponivel e armazanamos a porcentagem na tabela 4.

```
#agrupa as bikes por id e soma os trips durations
bikes_total_trip_duration = data[['tripduration','bikeid']].groupby
    (by = ['bikeid'], as_index = False).sum()
percentual_utilizacao = []
k = 0
for i in bikes_total_trip_duration['bikeid'].unique():
  total_disponivel = 24*60*60*len(data[data['bikeid'] == i]['
      day_in_month'].unique())
  percentual_utilizacao.append(100* bikes_total_trip_duration['
      tripduration'][k]/total_disponivel)
  k+=1

utilizacao = pd.DataFrame()
utilizacao.insert(0,'Percentual de Utilizacao',
    percentual_utilizacao)
utilizacao.sort_values(by = ['Percentual de Utilizacao'], ascending
     = False)
```

Fazendo algumas medições, concluimos que o percentual de utilização das bikes gira em torno de 7%.

```
utilizacao['Percentual de Utilizacao'].mean()
#output: 7.323968855980135
utilizacao['Percentual de Utilizacao'].median()
#output: 7.201668263090677
utilizacao['Percentual de Utilizacao'].std()
#output: 2.0224021644615453
```

	Quantil	Valor
0	1%	103.0
1	25%	381.0
2	50%	656.0
3	75%	1161.0
4	99%	4157.0

Figure 1: Tabela 1

bikeid	tripduration
30476	356012
31577	338084
30168	329555
32772	327246
30036	327185
31397	326900
32739	325070
33525	324623
32863	324544
31121	322499

Figure 2: Tabela 2

bikeid	tripduration
21437	117
17667	134
30968	150
19691	183
20619	192
18362	197
19451	213
32255	214
31463	272
19882	302

Figure 3: Tabela 3

	Percentual de Utilização
9415	19.020370
2936	16.810764
2155	16.504630
8787	16.406829
7988	15.923322
...	...
3587	0.222222
3104	0.211806
8662	0.173611
1882	0.155093
4093	0.135417

Figure 4: Tabela 4