09

Mãos na massa: Mesclando dados

Chegou a hora de você executar o que foi visto na aula! Para isso, execute os passos listados abaixo.

1) Você agora utilizará uma rotina Python para mesclar a base de dados **clima** com o CSV de UFOs, o arquivo **ufo.csv**, baixado na primeira aula (caso você ainda não o tenha baixado, faça o download aqui (https://s3.amazonaws.com/caelum-online-public/787-data-science-coleta-de-dados/01/ufo.csv)). Neste arquivo há colunas que representam a **cidade**, **cor**, **forma**, **estado** e **hora**, que inclui o **mês**, **dia**, **ano**, **hora** e **minuto**, sendo que o ano varia de **1930** a **2000** e algumas observações não possuem todas as variáveis.

2) Na base de clima, sequencialmente há dados somente desde **1997**, então os dados que poderão ser juntados se referem ao período de intersecção da base e do CSV, dos anos de **1997**, **1998**, **1999** e **2000**.

3) Baixe o código responsável pela mescla dos dados aqui (https://s3.amazonaws.com/caelum-online-public/787-data-science-coleta-de-dados/04/Incorpora_Kaggle.zip) e a biblioteca de utilidades aqui (https://s3.amazonaws.com/caelum-online-public/787-data-science-coleta-de-dados/04/Util.zip).

4) Em **Incorpora_Kaggle.py**, lembre-se de modificar a variável `df_ufo` com o diretório onde o arquivo **ufo.csv** está localizado. No seguinte código, você também deve dizer onde o arquivo **caixa_de_areia.csv**, no qual serão salvas as linhas não incorporadas, será salvo:

```
if caixa_de_areia:
    print ("----------------------------Descarregando a Caixa de Areia!")
    f = open('D://Datasets//caixa_de_areia.csv', 'wt')
    try:
        writer = csv.writer(f)
        for i in caixa_de_areia:
            writer.writerow(i)
        print("----------------------------Gerado arquivo caixa_de_areia.csv!")
    finally:

        f.close()
else:
    print ("----------------------------Caixa de areia vazia!")
```

Com isso feito, você pode executar a rotina.

5) Caso a rotina seja executada com sucesso, a coleção **clima_consolidado** será criada. Você já pode fazer consultas, através do **Robo 3T**, por exemplo, para contar os documentos da coleção:

```
db.clima_consolidado.count()
```

6) Logo após, veja se há algum documento sem histórico:

```
db.clima_consolidado.find (
    { history : { $exists : 0 } },
    { posicao : 1, estado : 1 }
).count()
```

7) Veja também o primeiro documento da coleção:

```
db.clima_consolidado.findOne()
```

8) Agora, visualize a frequência por ano:

```
db.clima_consolidado.aggregate ( [
    { $group : {_id : "$ano", quantos : {$sum : 1} } },
    { $sort: {quantos: -1} }
] )
```

9) E a frequência por estado:

```
db.clima_consolidado.aggregate ( [
    { $group : {_id : "$estado", quantos : {$sum : 1} } },
    { $sort: {quantos: -1} }
] )
```

10) Por fim, veja a frequência por estado e cidade:

```
db.clima_consolidado.aggregate ( [
    { $group : {_id : {estado: "$estado", cidade: "$cidade"}, quantos : {$sum : 1} } },
    { $sort: {quantos: -1} }
] )
```