08

Mãos na massa: Preparando os dados para o armazenamento

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

Será utilizado o **RStudio** como interface gráfica, então todos os comandos aqui mostrados deverão ser executados no seu console.

1) Abra o **RStudio** e carregue o script **Prepara_e_Armazena.R**, que você pode baixar aqui (https://s3.amazonaws.com/caelum-online-public/731-pipeline-big-data/03/arquivos/Prepara_e_Armazena.R). Caso a acentuação do conteúdo do script esteja errada, abra-o novamente, acessando o menu ***File -> Reopen with Enconding...***, escolhendo **UTF-8** em seguida.

2) Especifique o diretório de trabalho, executando o comando `setwd` do script, por exemplo:

```
setwd("C:\\Users\\eduar\\OneDrive\\aBig Data\\aAlura\\R")
```

3) E carregue os dados previamente coletados na aula anterior, atribuindo o *data frame* à variável `df_OVNI` :

```
df_OVNI <- read.csv("OVNIS.csv",stringsAsFactors = FALSE)
```

Removendo valores vazios e não disponíveis

4) Verifique se há valores não disponíveis e valores em branco para as **cidades**:

```
any(is.na(df_OVNI$State))
any(df_OVNI$State=="")
```

Como os dois comandos retornaram `TRUE` , remova as cidades vazias e não disponíveis do *data frame*:

```
df_OVNI <- df_OVNI[!(df_OVNI$City == "" | is.na(df_OVNI$City)), ]
```

5) Faça o mesmo para os **estados**:

```
any(is.na(df_OVNI$State))
any(df_OVNI$State=="")
```

Como apenas o segundo comando retorna `TRUE` , remova os estados vazios do *data frame*:

```
df_OVNI <- df_OVNI[!(df_OVNI$State == "" ), ]
```

6) E para os **tipos de OVNI**:

```
any(df_OVNI$Shape=="")

df_OVNI <- df_OVNI[!(df_OVNI$Shape == "" ), ]
```

Removendo estados fora dos Estados Unidos

7) Veja quantos estados diferentes estão no *data frame*:

```
unique(df_OVNI$State)
```

8) Crie um novo *data frame*, com somente estados válidos dos EUA, a partir do CSV **states.csv**, que você pode baixar aqui (https://s3.amazonaws.com/caelum-online-public/731-pipeline-big-data/03/arquivos/states.csv):

```
df_estados_validos <- read.csv("states.csv",stringsAsFactors = FALSE)
```

9) Faça um filtro no *data frame* `df_OVNI` , mantendo somente os estados que estão no *data frame* que você criou no passo anterior:

```
df_OVNI <- df_OVNI[(df_OVNI$State %in% df_estados_validos$Abbreviation),]
```

10) Conte novamente os estados e veja que desta vez há somente os 51 estados dos EUA no *data frame*:

```
unique(df_OVNI$State)
```

Removendo variáveis irrelevantes

11) Remova as variáveis irrelevantes, `Posted` , `Duration` , `Summary` e `X` , atribuindo `NULL` a elas:

```
df_OVNI$Posted <- NULL
df_OVNI$Duration <- NULL
df_OVNI$Summary <- NULL
df_OVNI$X <- NULL
```

Removendo tipos de OVNIs incomuns

12) Conte os tipos de OVNIs contidos no *data frame*:

```
unique(df_OVNI$Shape)
```

13) Executando o comando acima, você consegue ver que há 33 tipos diferentes de OVNIs. Crie então um novo *data frame* os tipos e a quantidade de ocorrências, carregando o *package* **sqldf** e executando a seguinte *query*:

```
require (sqldf)
OVNI_EUA_por_Tipo = sqldf("select Shape, count(*) Views
                from df_OVNI group by Shape order by 2 desc")
```

14) Nesse *data frame*, remova todos os tipos de OVNIs com menos de 1000 ocorrências:

```
OVNI_EUA_por_Tipo = sqldf("select Shape, count(*) Views
                from df_OVNI group by Shape
                having count(*) > 1000")
```

15) Agora, faça um filtro no *data frame* `df_OVNI` , mantendo somente os tipos de OVNIs que estão no *data frame* que você criou no passo anterior:

```
df_OVNI <- df_OVNI[(df_OVNI$Shape %in% OVNI_EUA_por_Tipo$Shape),]
```

16) Aproveite e remova também o tipo **Unkown**:

```
df_OVNI <- df_OVNI[!(df_OVNI$Shape == "Unknown" ), ]
```

Acrescentando novas variáveis

17) Como os dados muitas vezes estão ordenados de uma forma que não são muito úteis para fazer pesquisas, como a variável `Date...Time` . Então, separe-a em uma variável de **data** e outra de **hora**. Não se esqueça de deletar a variável `Date...Time` em seguida:

```
d <- strsplit(df_OVNI$Date...Time, ' ')
e <- do.call(rbind.data.frame, d)
colnames(e) <- c("Sight_Date", "Sight_Time")
e <- data.frame(lapply(e, as.character), stringsAsFactors=FALSE)
df_OVNI <- cbind(df_OVNI, e)
df_OVNI$Date...Time <- NULL
```

18) Acrescente também a variável para o **dia da semana** da ocorrência:

```
df_OVNI$Sight_Weekday <- weekdays(as.Date(df_OVNI$Sight_Date, '%m/%d/%y'))
```

19) Por fim, separe também em **mês** e **dia**, removendo o ano em seguida:

```
e <- do.call(rbind.data.frame, strsplit(df_OVNI$Sight_Date, '/'))
e <- data.frame(lapply(e, as.integer))
colnames(e) <- c("Sight_Month", "Sight_Day", "Sight_Year")
df_OVNI <- cbind(df_OVNI, e)
df_OVNI$Sight_Year <- NULL
```