07

Mãos na massa: Análise exploratória de dados

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

1) Caso o MongoDB não esteja no ar, faça, colocando os devidos diretórios:

```
start /b mongod --dbpath D:\MongoDB\data\db --logpath D:\MongoDB\log.txt --oplogSize 50 --smallfiles
```

2) Abra o **RStudio** e no seu próprio console, instale e importe o **mongolite**:

```
> install.packages("mongolite")
> library(mongolite)
```

3) Após, faça a conexão, criando um objeto e passando o nome da coleção, seguida da URL da base de dados:

```
> m <- mongo("clima_consolidado", url="mongodb://localhost:27017/dbclima")
```

3) Em seguida, carregue todos os dados da coleção **clima_consolidado** em um *data frame*:

```
> df_clima_consolidado <- m$find ('{}')
```

4) Veja quantas observações e quantas variáveis há no *data frame*:

```
> dim(df_clima_consolidado)
```

5) Observe também a sua estrutura:

```
> str(df_clima_consolidado)
```

6) Veja as primeiras linhas do *data frame*:

```
> head(df_clima_consolidado)
```

7) Com o `view` você tem uma visualização melhor:

```
> view(df_clima_consolidado)
```

8) Para ver as últimas medidas de temperatura em graus Celsius, faça:

```
> tail(df_clima_consolidado$history$observations$tempm)
```

9) Para saber os tipos de dados das variáveis do *data frame*, utilize o comando `sapply`:

```
> sapply(df_clima_consolidado, class)
```

10) Veja quantas observações há para cada estado, através do `table`:

```
> table(df_clima_consolidado$estado)
```

11) Para fazer uma estrutura plana, instale e importe o **jsonlite**:

```
> install.packages("jsonlite")
> library(jsonlite)
```

12) E utilize a função `flatten`, visualizando a sua estrutura em seguida:

```
> df_clima_consolidado = flatten(df_clima_consolidado)
> str(df_clima_consolidado)
```

13) Como agora não há mais um *data frame* dentro de outro, você pode visualizar diretamente a variável, por exemplo:

```
> table(df_clima_consolidado$history.observations.pressurem)
```

Consultas mais elaboradas com DPLYR

14) Baixe aqui (https://s3.amazonaws.com/caelum-online-public/799-data-science-preparacao-e-exploracao-dos-dados/02/Arquivo_Apoio_Aula_02_dplyr.R) o arquivo **Arquivo_Apoio_Aula_02_dplyr.R** e abra-o no **RStudio**.

15) Seguindo passo-a-passo o programa, instale e importe o pacote **dplyr**:

```
install.packages("dplyr")
library(dplyr)
```

16) Crie o *data frame* **temperaturas**, com as colunas **cidade**, **history.observations.tempm** e **history.observations.tempi** do *data frame* **df_clima_consolidado**:

```
temperaturas <- select(df_clima_consolidado,
                       cidade, history.observations.tempm,
                       history.observations.tempi)
```

17) Você pode fazer o `head` do *data frame* ou só de algumas colunas:

```
head(temperaturas)
head(select(df_clima_consolidado, estado:hora))
head(select(df_clima_consolidado, starts_with("history")))
```

18) Em seguida, filtre por linha, selecionando somente a de posição **10000**:

```
filter(df_clima_consolidado, posicao == 10000)
```

19) Você também pode ver as primeiras observações que tenham como critério o estado **CA** e a cidade **Sacramento**:

```
head(filter(df_clima_consolidado,
            estado == "CA",
            cidade == "Sacramento"))
```

20) E as primeiras observações que tenham como critério o estado **FL** ou **TX**:

```
head(filter(df_clima_consolidado, estado %in% c("FL", "TX")))
```

21) O operador `%>%` serve para encadear operações. Por exemplo, no *data frame* **df_clima_consolidado**, você pode selecionar somente a linha de posição **10000**, selecionando em seguida as suas colunas **estado** até **hora**:

```
df_clima_consolidado  %>%
  filter(posicao == 10000) %>%
  select(estado:hora)
```

22) Outro exemplo de encadeamento de operações é na **ordenação**. Você pode filtrar por observação, selecionar algumas colunas, ordenar por alguma delas, através do `arrange`, e exibir somente as primeiras, através do `head`:

```
df_clima_consolidado  %>%
    filter(estado %in% c("FL", "TX")) %>%
    select(estado:hora, posicao) %>%
    arrange(cidade) %>%
    head
```

23) Agora, a **agregação**, com `summarise`. Por exemplo, para selecionar a médias das temperaturas, você utiliza a função `mean`, mas esse é um campo numérico, então a temperatura deve ser convertida, através da função `as.numeric`:

```
df_clima_consolidado %>%
    summarise(media_temperaturas =
                mean(as.numeric(history.observations.tempm)))
```

24) Por fim, mais funções de agregação. Para descobrir a maior temperatura, usa-se a função `max`, para descobrir a menor temperatura, usa-se a função `min`, e para descobrir a quantidade, usa-se a função `n`:

```
df_clima_consolidado %>%
  summarise(temp_media = mean(as.numeric(history.observations.tempm)),
  temp_max = max(as.numeric(history.observations.tempm)),
  temp_min = min(as.numeric(history.observations.tempm)),
  quantas = n())
```