

desafio 3 - me315

Sophia Santiago

Formato JSON

Descrição:

- JSON (JavaScript Object Notation) é um formato de texto leve utilizado para representar e transmitir dados estruturados.
- Ele organiza a informação em pares de **chave e valor**, semelhantes a um dicionário.
- Sua estrutura permite a inclusão de listas e objetos aninhados, sendo muito comum em aplicações web e APIs.

Formato Parquet

Descrição:

- Parquet é um formato de arquivo binário colunar desenvolvido para manipulação eficiente de grandes volumes de dados.
- Ao armazenar os dados por colunas, permite compressão mais eficaz e leitura mais rápida — o que é especialmente útil em ambientes de Big Data.
- É amplamente utilizado em plataformas como Apache Spark, Hadoop e sistemas analíticos modernos.

Base de Dados Utilizada

- Para este exemplo, foi utilizado o dataset **cars**, que já vem integrado no R.
- Esse conjunto de dados contém informações sobre **velocidade de veículos (speed)** e **distância de frenagem (dist)**.
- Embora o dataset não tenha vindo de um arquivo externo, ele foi exportado nos formatos **JSON** e **Parquet** para simular um processo real de transformação e armazenamento de dados.

Código em R – Exportação e Importação

Instalação de pacote

```
#install.packages("arrow") #Leitura do parquet
#install.packages("rio") #pacote utilizado na ultima aula
#rio::install_formats()
#install.packages("tinytex")
#tinytex::install_tinytex()
#tinytex::tlmgr_install("koma-script")
```

```
library(tinytex)
library(arrow) # Para salvar e ler Parquet
```

Anexando pacote: 'arrow'

O seguinte objeto é mascarado por 'package:utils':

```
timestamp
```

```
library(rio)
library(readr)
```

Leitura do banco de dados utilizado

```
data("cars") #data cars
```

Transformação do banco em seus devidos formatos (csv, json, parquet)

```
# Exportar para CSV
export(cars, "cars.csv")

# Exportar para JSON (rio usa jsonlite por baixo)
export(cars, "cars.json")

# Exportar para Parquet (arrow)
write_parquet(cars, "cars.parquet")
```

```
#Leitura dos dados em seus devidos formatos
```

```
# Ler CSV
```

```
cars_csv <- import("cars.csv")
```

```
# Ler JSON
```

```
cars_json <- import("cars.json")
```

```
# Ler Parquet
```

```
cars_parquet <- read_parquet("cars.parquet")
```

```
# Comparar as estruturas
```

```
str(cars)
```

```
'data.frame':  50 obs. of  2 variables:
 $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
 $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
```

```
str(cars_csv)
```

```
'data.frame':  50 obs. of  2 variables:
 $ speed: int  4 4 7 7 8 9 10 10 10 11 ...
 $ dist : int  2 10 4 22 16 10 18 26 34 17 ...
```

```
str(cars_json)
```

```
'data.frame':  50 obs. of  2 variables:
 $ speed: int  4 4 7 7 8 9 10 10 10 11 ...
 $ dist : int  2 10 4 22 16 10 18 26 34 17 ...
```

```
str(cars_parquet)
```

```
tibble [50 x 2] (S3: tbl_df/tbl/data.frame)
 $ speed: num [1:50] 4 4 7 7 8 9 10 10 10 11 ...
 $ dist : num [1:50] 2 10 4 22 16 10 18 26 34 17 ...
```