

Desafio - 03

Iuri Santos Oliveira - 194610

1. Dados de Táxi de Nova York

O conjunto de dados obtido pelo link https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2023-01.parquet representam, aproximadamente, 3 milhões de corridas de táxis em Nova York durante o período de janeiro de 2023. As variáveis presentes neste conjunto de dados são: Identificadores (ID da empresa de táxi), geolocalização, métricas financeiras (como valor de gorjeta, pedágio) e formas de pagamento. O código para a leitura dos dados foi a seguinte:

```
library(arrow) # Carrega o pacote arrow

url <- "https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2023-01.parquet"
dados_voos <- read_parquet(url) # Lê o arquivo Parquet diretamente da URL

dados_voos_df <- as.data.frame(dados_voos) # Converte para Data Frame
```

Para a leitura do banco de dados, utilizou-se o pacote arrow, uma ferramenta para trabalhar com grandes conjuntos de dados de forma eficiente, podendo ler arquivos no formato parquet. Basicamente, o código lê o arquivo diretamente da URL sem download prévio, depois detecta o formato, identifica a estrutura das colunas, carrega-os e os converte em Data Frame.

2. Dados de Criptomoedas da CoinGecko

No conjunto de dados proveniente do link https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=100&page=1&sparkline=false lidamos com a CoinGecko, uma interface de programação de aplicativo (API) na qual fornece informações sobre criptomoedas e ativos digitais (NFTs) para desenvolvedores, traders e projetos. As variáveis presentes neste conjunto de dados são: métricas de mercado (preço atual da moeda, volume de negociações) e os diferentes tipos de variações (variação percentual do próprio preço da moeda).

```
library(jsonlite) # Carrega o pacote jsonlite

url <- "https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=10&page=1&sparkline=false"
dados_cripto <- fromJSON(url) # Faz requisição HTTP e converte JSON para lista

dados_cripto_df <- as.data.frame(dados_cripto) # Converte a lista para Data Frame
```

Para esse conjunto de dados, o código carrega o pacote especializado em JSON, faz uma requisição HTTP GET para a URL, recebe a resposta em formato JSON e converte a lista aninhada em Data Frame.