



Data Science Academy

www.datascienceacademy.com.br

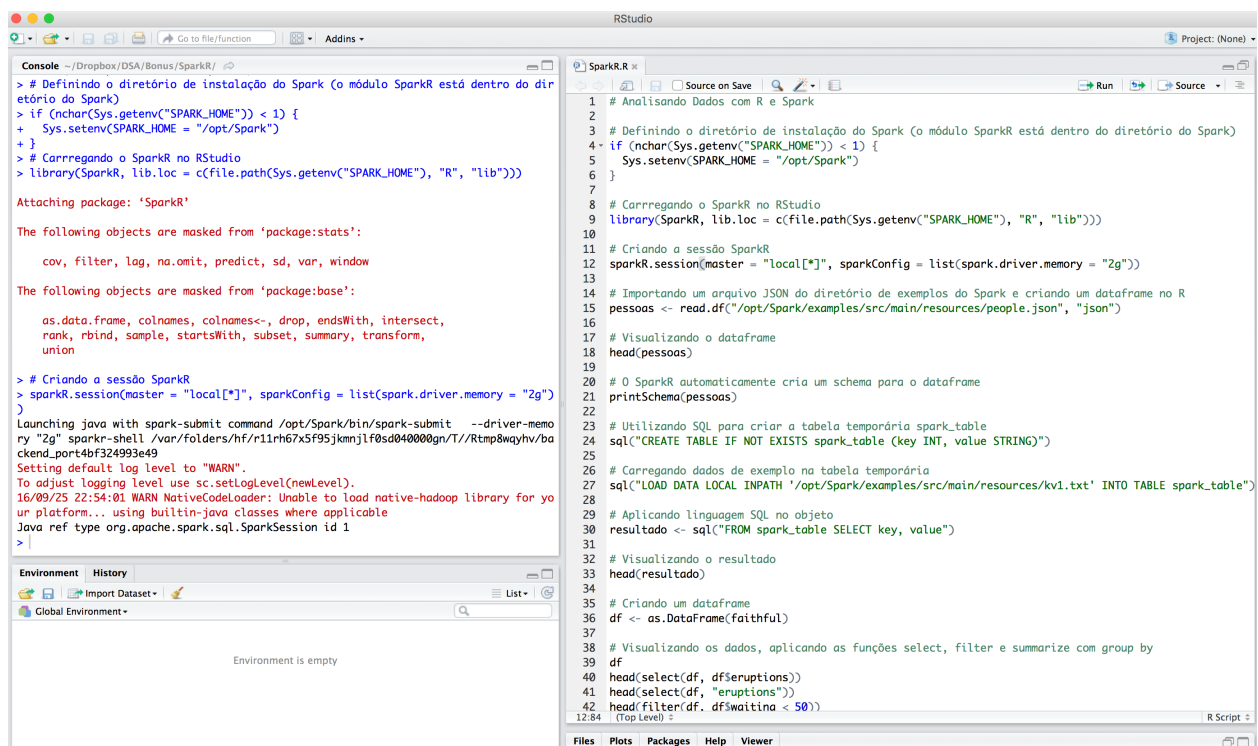
Big Data Real-Time Analytics com Python e Spark

Bonus
R e Spark com SparkR



A implementação da linguagem R com Spark é recente e muitas operações do Spark ainda não podem ser realizadas em linguagem R. Mas o suporte a linguagem vem crescendo e muito em breve será possível executar todo seu processo de análise em R, com o Framework Spark e sua capacidade de processamento paralelo e distribuído. No arquivo anexo, você encontra um script R, comentado linha a linha. Abra o arquivo no RStudio e execute cada um dos comandos. Experimente outras opções e tente reproduzir em R, o que você aprendeu ao longo do curso com a linguagem Python.

Obs: configure no script o diretório de instalação do Spark na sua máquina.



```
# Definindo o diretório de instalação do Spark (o módulo SparkR está dentro do diretório do Spark)
> if (nchar(Sys.getenv("SPARK_HOME")) < 1) {
+   Sys.setenv(SPARK_HOME = "/opt/Spark")
+ }
> # Carregando o SparkR no RStudio
> library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib"))

Attaching package: 'SparkR'

The following objects are masked from 'package:stats':
  cov, filter, log, na.omit, predict, sd, var, window

The following objects are masked from 'package:base':
  as.data.frame, colnames, colnames<-, drop, endsWith, intersect,
  rank, rbind, sample, startsWith, subset, summary, transform,
  union

> # Criando a sessão SparkR
> sparkR.session(master = "local[*]", sparkConfig = list(spark.driver.memory = "2g"))
Launching java with spark-submit command /opt/Spark/bin/spark-submit --driver-memory "2g" sparkr-shell /var/folders/hf/r1rh67x5f95jkmjlf0sd040000gn/T//Rtmp8wayhv/ba
ckend_port4bf324993e49
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel).
16/09/25 22:54:01 WARN NativeCodeLoader: Unable to load native-hadoop library for yo
ur platform... using builtin-java classes where applicable
Java ref type org.apache.spark.sql.Session id 1
>

1 # Analisando Dados com R e Spark
2
3 # Definindo o diretório de instalação do Spark (o módulo SparkR está dentro do diretório do Spark)
4 if (nchar(Sys.getenv("SPARK_HOME")) < 1) {
5   Sys.setenv(SPARK_HOME = "/opt/Spark")
6 }
7
8 # Carregando o SparkR no RStudio
9 library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib")))
10
11 # Criando a sessão SparkR
12 sparkR.session(master = "local[*]", sparkConfig = list(spark.driver.memory = "2g"))
13
14 # Importando um arquivo JSON do diretório de exemplos do Spark e criando um dataframe no R
15 pessoas <- read.df("/opt/Spark/examples/src/main/resources/people.json", "json")
16
17 # Visualizando o dataframe
18 head(pessoas)
19
20 # O SparkR automaticamente cria um schema para o dataframe
21 printSchema(pessoas)
22
23 # Utilizando SQL para criar a tabela temporária spark_table
24 sql("CREATE TABLE IF NOT EXISTS spark_table (key INT, value STRING)")
25
26 # Carregando dados de exemplo na tabela temporária
27 sql("LOAD DATA LOCAL INPATH '/opt/Spark/examples/src/main/resources/kv1.txt' INTO TABLE spark_table")
28
29 # Aplicando linguagem SQL no objeto
30 resultado <- sql("FROM spark_table SELECT key, value")
31
32 # Visualizando o resultado
33 head(resultado)
34
35 # Criando um dataframe
36 df <- as.DataFrame(faithful)
37
38 # Visualizando os dados, aplicando as funções select, filter e summarize com group by
39 df
40 head(select(df, df$eruptions))
41 head(select(df, "eruptions"))
42 head(filter(df, df$waittime < 50))
12:84 (Top Level) > R Script >
```