

# **RICARDO SOUSA**

## **BUSINESS CASE 6**

### **DOCUMENTAÇÃO**

#### **RESUMO:**

Este script escrito em linguagem R faz o trabalho de ler um arquivo, processar os dados, e retornar o arquivo transformado.

*Dica: É necessário executar cada comando em sequencia para o script funcionar corretamente, e para executar cada comando segure a tecla 'Ctrl' e aperte 'Enter' em seguida com o 'Ctrl' ainda pressionado.*

#### **Como utilizar:**

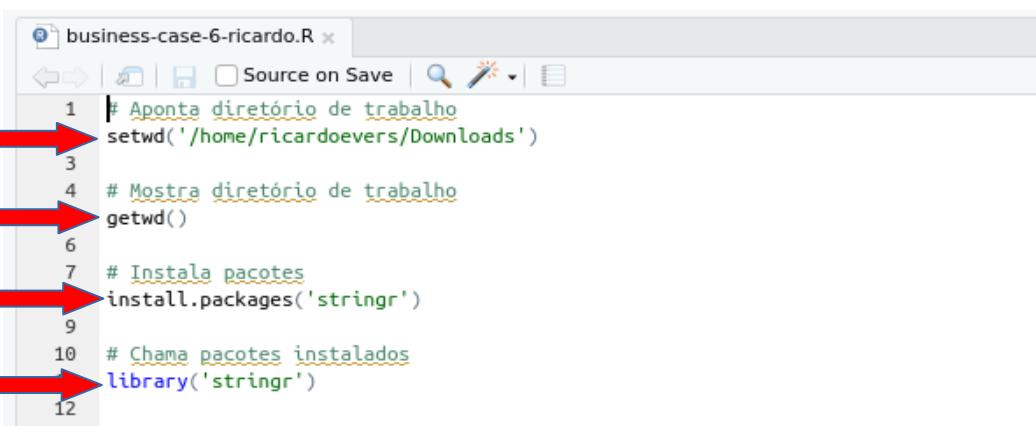
--Primeira parte--

1 - Será necessário apontar o diretório de trabalho dentro da função ‘setwd()’.

2 - Checar se o diretório escolhido foi realmente selecionado com a função ‘getwd()’.

3 - Instalar os pacotes necessário com a função ‘install.packages()’.

4 - Chamar e acessar os pacotes instalados com a função ‘library()’.



The screenshot shows an R script editor window titled "business-case-6-ricardo.R". The code is as follows:

```
# Aponta diretório de trabalho
setwd('/home/ricardoevers/Downloads')

# Mostra diretório de trabalho
getwd()

# Instala pacotes
install.packages('stringr')

# Chama pacotes instalados
library('stringr')
```

Four red arrows point to the following lines of code:

- Line 1: # Aponta diretório de trabalho
- Line 4: # Mostra diretório de trabalho
- Line 7: # Instala pacotes
- Line 10: # Chama pacotes instalados

## --Segunda parte--

1 - Informar o arquivo desejado na variável ‘df’ e colocar o separador de coluna desejado no ‘sep=’.

```
14
15 # Lê arquivo csv
16 df <- read.csv("Worksheet.csv", sep=";")
17
```



## --Terceira parte--

Esta parte não é necessário fazer muita coisa, apenas altere o que precisar, como selecionar as colunas que deseja utilizar em cada parte da função, e para selecionar uma coluna específica do Data Frame será necessário utilizar a sintaxe ‘dataframe\$coluna’. Especificações do que cada pedaço de código faz foram inseridas diretamente dentro do arquivo do script.

```
18 # Função para processar DataFrame e retorna-lo transformado
19 table_reading_and_processing <- function(df){
20
21 # Remove coluna Id
22 df$id <- NULL
23
24 # Transforma valores das colunas IsApproved e NotMonetize em binario
25 transform_to_binary <- c("N" = 0, "Y" = 1)
26 df$NotMonetize <- transform_to_binary[df$NotMonetize]
27 df$IsApproved <- transform_to_binary[df$IsApproved]
28 df$IsApproved <- replace(df$IsApproved, is.na(df$IsApproved), 0)
29 df$NotMonetize <- replace(df$NotMonetize, is.na(df$NotMonetize), 0)
30
31 # Conta e transforma os dias da coluna InformedDate
32 initial_date <- '01/01/2001'
33 initial_date <- as.Date(initial_date, format='%d/%m/%Y')
34 final_date <- as.Date(df$InformedDate, format='%d/%m/%Y')
35 day_count <- final_date - initial_date
36 df$InformedDate <- day_count
37
38 #Retira artigos e caracteres especiais
39 df$Description <- str_replace_all(df$Description, " o ", " ")
40 df$Description <- str_replace_all(df$Description, " os ", " ")
41 df$Description <- str_replace_all(df$Description, " a ", " ")
42 df$Description <- str_replace_all(df$Description, " as ", " ")
43 df$Description <- str_replace_all(df$Description, " um ", " ")
44 df$Description <- str_replace_all(df$Description, " uns ", " ")
45 df$Description <- str_replace_all(df$Description, " uma ", " ")
46 df$Description <- str_replace_all(df$Description, " unas ", " ")
47 df$Description <- str_replace_all(df$Description, "[[:punct:]]", ' ')
48
49 #Transforma horas em fração do dia
50 df$Worked.hours <- as.POSIXlt(df$Worked.hours, format='XH: XM')
51 df$Worked.hours <- with(df, (Worked.hours$min*60 + Worked.hours$hour*3600) / (24*3600))
52 df$StartTime <- as.POSIXlt(df$StartTime, format='XH: XM')
53 df$StartTime = with(df, (StartTime$min*60 + StartTime$hour*3600) / (24*3600))
54 df$EndTime <- as.POSIXlt(df$EndTime, format='XH: XM')
55 df$EndTime = with(df, (EndTime$min*60 + EndTime$hour*3600) / (24*3600))
56 df$Worked.hours <- replace(df$Worked.hours, is.na(df$Worked.hours), 0)
57
58 # Cria coluna com diferença de tempo entre StartTime e EndTime
59 df$TotalTime <- df$EndTime - df$StartTime
60
61
62 # Retorna DataFrame transformado
63 return(df)
64 }
```

## --Quarta parte – (Final)

1 - Chamar a função ‘table\_reading\_and\_processing(nome\_da\_variavel\_da\_leitura)’ com o parâmetro sendo a variável atribuída à leitura do arquivo logo no início.

```
1   table_reading_and_processing(df)  
64  
66  
67
```

---