Universidade de Brasília

DEPARTAMENTO DE ESTATÍSTICA

07 setembro 2022

Lista 3

Prof. Guilherme Rodrigues

Computação em Estatística para dados e cálculos massivos

Tópicos especiais em Estatística 1

Aluno: Bruno Gondim Toledo | Matrícula: 15/0167636

```
knitr::opts_chunk$set(echo = TRUE)

if (!require("pacman")) install.packages("pacman")

p_load(pacman,installr,tidyverse,sparklyr,doParallel,readr,foreach,arrow,read.dbc,vroom,fs,data.tabl

threads=round(as.numeric(detectCores()*0.8))

memoria <- round(as.numeric(get_ram()) * 0.6 / 1000000000)
memoria <- as.character(memoria)
memoria <- paste(memoria,'g',sep="")
```

Questão 1: Criando o cluster spark.

a) Crie uma pasta (chamada datasus) em seu computador e faça o download dos arquivos referentes ao Sistema de informação de Nascidos Vivos (SINASC), os quais estão disponíveis em https://datasus.saude.gov.br/transferencia-de-arquivos/.

Atenção: Considere apenas os Nascidos Vivos no Brasil (sigla DN) entre 1994 e 2020, incluindo os dados estaduais e excluindo os arquivos referentes ao Brasil (sigla BR). Use wi-fi para fazer os downloads!

Dica: O endereço ftp://ftp.datasus.gov.br/dissemin/publicos/SINASC/1996_/Dados/DNRES/permite a imediata identificação dos endereços e arquivos a serem baixados.

```
pasta <- "datasus"

if (file.exists(pasta)) {

  print("A pasta já existe")
  rm(pasta)

} else {

  dir.create(pasta)
  rm(pasta)

}
```

```
## [1] "A pasta já existe"
```

```
### download.file("ftp://ftp.datasus.gov.br/dissemin/publicos/SINASC/1996_/Dados/DNRES/", "./datasus

referencia <- read_table("datasus/referencia.txt",col_names = FALSE)

nome <- referencia$X4
rm(referencia)

link <- paste("ftp://ftp.datasus.gov.br/dissemin/publicos/SINASC/1996_/Dados/DNRES/" , nome, sep="")
destino <- paste("./datasus/",nome,sep="")

### foreach(i = 1:686) %dopar% {download.file(link[i], destino[i],mode="wb")}
```

b) Usando a função `p_load`(do pacote `pacman`), carregue os pacotes `arrow` e `read.dbc` e converta os arquivos baixados no item *a)* para formato o *.parquet*. Em seguida, converta para *.csv* apenas os arquivos referentes aos estados GO, MS e ES. Considerando apenas os referidos estados, compare o tamanho ocupado pelos arquivos nos formatos *.parquet* e *.csv* (use a função `file.size`).

```
###banco <- list()
###for (i in 1:686){
###  banco[[i]] <- read.dbc(destino[i])
###}
###rm(i)

nome2 <- str_sub(nome,start=1,end=8)
nomep <- paste(nome2,".parquet",sep="")
nomep <- paste("./datasus/",nomep,sep="")

### for (i in 1:686){
###   write_parquet(as.data.frame(banco[i]),nomep[i])
### }
### rm(i)

nomet <- as.tibble(nome)
nometGO <- nomet %>%
  filter(str_detect(value, "GO"))
nometMS <- nomet %>%
  filter(str_detect(value, "MS"))
nometES <- nomet %>%
  filter(str_detect(value, "ES"))

nomet <- rbind(nometES,nometGO,nometMS)

selecao <- as.vector(nomet)
selecao <- selecao$value

selecaod <- paste("./datasus/",selecao,sep="")

### banco <- list()
### for (i in 1:75){
###   banco[[i]] <- read.dbc(selecaod[i])
### }

selecao2 <- str_sub(selecao,start=1,end=8)
selecao2 <- paste(selecao2,".csv",sep="")
selecao2p <- paste("./datasus/",selecao2,sep="")

###for (i in 1:75){
###  vroom_write(as.data.frame(banco[i]),selecao2p[i])
###}

selecao3 <- str_sub(selecao,start=1,end=8)
selecao3 <- paste(selecao3,".parquet",sep="")
selecao3 <- paste("./datasus/",selecao3,sep="")

arquivosCSV <- selecao2p
arquivosPARQUET <- selecao3
```

```
fscsv1 <- file.size(arquivosCSV)
fsparquet1 <- file.size(arquivosPARQUET)

fscsv2 <- file_size(arquivosCSV)
fsparquet2 <- file_size(arquivosPARQUET)

diferenca <- file_size(arquivosCSV) - file_size(arquivosPARQUET)

diferencasomada <- sum(diferenca)

diferenca <- as.data.frame(diferenca)
```

Diferença no tamanho ocupado pelos arquivos nos formatos *.parquet* e *.csv*

```
kable(diferenca)
```

	diferenca
./datasus/DNES1996.csv	3.4M
./datasus/DNES1997.csv	3.68M
./datasus/DNES1998.csv	3.53M
./datasus/DNES1999.csv	3.39M
./datasus/DNES2000.csv	3.23M
./datasus/DNES2001.csv	3.99M
./datasus/DNES2002.csv	4.31M
./datasus/DNES2003.csv	4.19M
./datasus/DNES2004.csv	4.17M
./datasus/DNES2005.csv	4.11M
./datasus/DNES2006.csv	5.1M
./datasus/DNES2007.csv	5.08M
./datasus/DNES2008.csv	5.12M
./datasus/DNES2009.csv	5.07M
./datasus/DNES2010.csv	9.39M
./datasus/DNES2011.csv	9.65M
./datasus/DNES2012.csv	9.25M
./datasus/DNES2013.csv	10.39M
./datasus/DNES2014.csv	10.67M
./datasus/DNES2015.csv	10.77M
./datasus/DNES2016.csv	10.16M
./datasus/DNES2017.csv	10.5M
./datasus/DNES2018.csv	10.72M
./datasus/DNES2019.csv	10.4M
./datasus/DNES2020.csv	10.16M
./datasus/DNGO1996.csv	6.06M
./datasus/DNGO1997.csv	6.17M
./datasus/DNGO1998.csv	6.03M
./datasus/DNGO1999.csv	5.26M
./datasus/DNGO2000.csv	5.24M
./datasus/DNGO2001.csv	6.46M
./datasus/DNGO2002.csv	7.34M
./datasus/DNGO2003.csv	7.22M
./datasus/DNGO2004.csv	7.18M
./datasus/DNGO2005.csv	7.2M
./datasus/DNGO2006.csv	8.57M
./datasus/DNGO2007.csv	8.38M
./datasus/DNGO2008.csv	8.57M
./datasus/DNGO2009.csv	8.57M

	diferenca
./datasus/DNGO2010.csv	15.81M
./datasus/DNGO2011.csv	16.31M
./datasus/DNGO2012.csv	16.44M
./datasus/DNGO2013.csv	18.29M
./datasus/DNGO2014.csv	18.58M
./datasus/DNGO2015.csv	19.1M
./datasus/DNGO2016.csv	17.98M
./datasus/DNGO2017.csv	18.01M
./datasus/DNGO2018.csv	18.32M
./datasus/DNGO2019.csv	17.87M
./datasus/DNGO2020.csv	17.16M
./datasus/DNMS1996.csv	2.67M
./datasus/DNMS1997.csv	2.65M
./datasus/DNMS1998.csv	2.53M
./datasus/DNMS1999.csv	2.56M
./datasus/DNMS2000.csv	2.24M
./datasus/DNMS2001.csv	2.8M
./datasus/DNMS2002.csv	3.12M
./datasus/DNMS2003.csv	3.07M
./datasus/DNMS2004.csv	3.24M
./datasus/DNMS2005.csv	3.23M
./datasus/DNMS2006.csv	3.92M
./datasus/DNMS2007.csv	3.85M
./datasus/DNMS2008.csv	4.08M
./datasus/DNMS2009.csv	3.98M
./datasus/DNMS2010.csv	7.27M
./datasus/DNMS2011.csv	7.63M
./datasus/DNMS2012.csv	7.4M
./datasus/DNMS2013.csv	8.13M
./datasus/DNMS2014.csv	8.29M
./datasus/DNMS2015.csv	8.34M
./datasus/DNMS2016.csv	8M
./datasus/DNMS2017.csv	8.35M
./datasus/DNMS2018.csv	8.32M
./datasus/DNMS2019.csv	8.19M
./datasus/DNMS2020.csv	7.7M

Somando as diferenças, no final os arquivos em *.csv* são 6.2299546×10^8 (**594 Megabytes**) maiores que os arquivos em *.parquet*

c) Crie uma conexão *Spark*, carregue para ele os dados em formato *.parquet* e *.csv* e compare os respectivos tempos computacionais. Se desejar, importe apenas as colunas necessárias para realizar a Questão2.

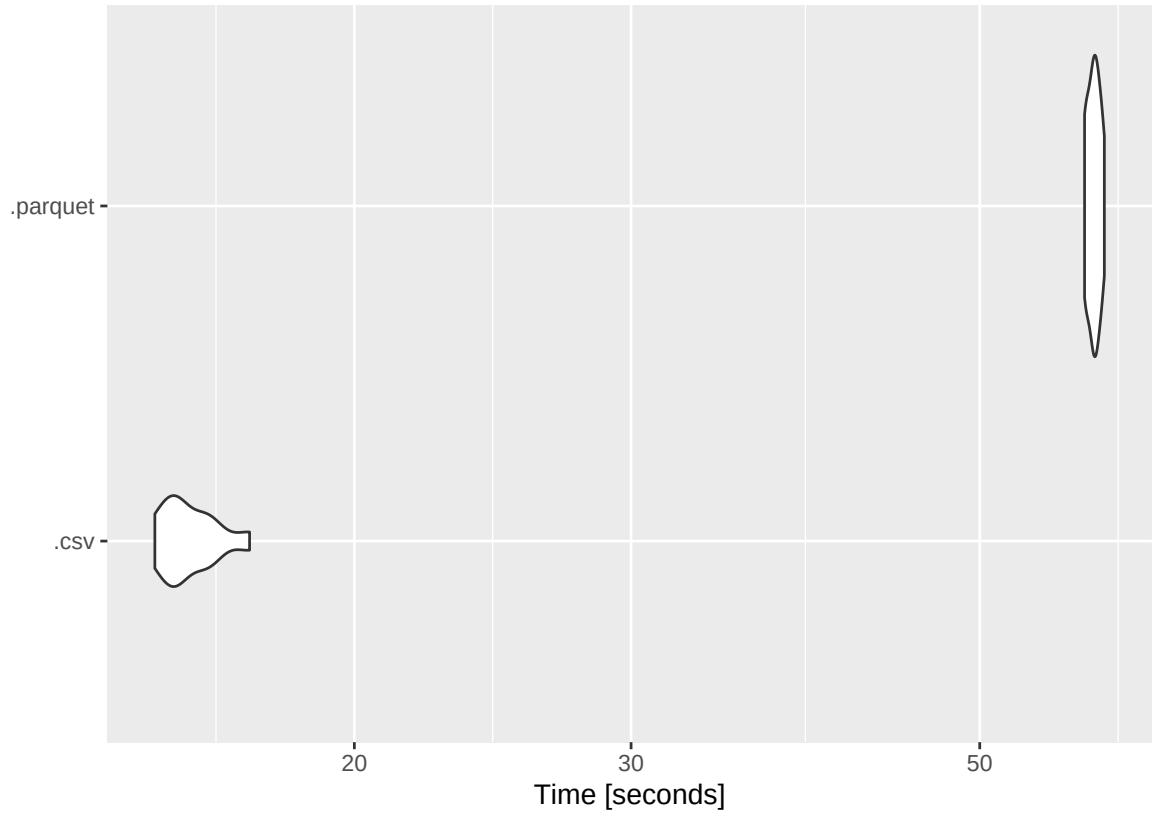
OBS: Lembre-se de que quando indicamos uma pasta na conexão, as colunas escolhidas para a análise precisam existir em todos os arquivos.

```
###df <- arquivosCSV %>%
###  map(fread,
###      nThread=threads,
###      fill=TRUE) %>%
###  rbindlist(fill=TRUE)
###df <- df[,1:21]

###vroom_write(df,"./datasus/dfcsv")

###dfparquet <- arquivosPARQUET %>%
###  map(read_parquet) %>%
###  rbindlist(fill=TRUE)
###dfparquet <- dfparquet[,1:21]

###write_parquet(dfparquet,"./datasus/dfparquet")

config <- spark_config()
config$spark.executor.cores <- threads
config$spark.executor.memory <- memoria
sc <- spark_connect(master = "local", config = config)

dfparquet = spark_read_parquet(sc=sc,
                               name = "dfparquet",
                               path = "./datasus/dfparquet",
                               header = TRUE,
                               delimiter = "\\t",
                               charset = "latin1",
                               infer_schema = T,
                               overwrite = T)

###dfcsv = spark_read_csv(sc=sc,
###                       name = "dfcsv",
###                       path = "./datasus/dfcsv",
###                       header = TRUE,
###                       delimiter = "\\t",
###                       charset = "latin1",
###                       infer_schema = T,
###                       overwrite = T)



### mbmcsv <- microbenchmark(
###   {dfcsv = spark_read_csv(sc=sc,
###                           name = "dfcsv",
###                           path = "./datasus/dfcsv",
###                           header = TRUE,
###                           delimiter = "\\t",
###                           charset = "latin1",
###                           infer_schema = T,
###                           overwrite = T)
###   }, times = 10)
```

```
###
### mbmparquet <- microbenchmark(
###   {dfparquet = spark_read_parquet(sc=sc,
###                         name = "dfparquet",
###                         path = "./datasus/dfparquet",
###                         header = TRUE,
###                         delimiter = "\\t",
###                         charset = "latin1",
###                         infer_schema = T,
###                         overwrite = T)
###   },times = 10)

### mbmcsv$expr <- NA
### mbmcsv$expr <- ".csv"

### mbmparquet$expr <- NA
### mbmparquet$expr <- ".parquet"

### mbm <- rbind(mbmcsv,mbmparquet)

### saveRDS(mbm,file="mbmsparkload.rds")
```

Comparando tempos computacionais:

```
mbm <- readRDS("mbmsparkload.rds")

autoplot(mbm)
```



Percebe-se que o arquivo em *.csv* foi carregado bem mais rápido que o arquivo em *.parquet* para o *spark*, resultado esse inesperado visto que o arquivo *.parquet* é menor e mais otimizado. Porém, após carregado,

todas as manipulações e comandos rodam mais rapidamente no arquivo *.parquet* em relação ao arquivo *.csv*. Portanto, utilizarei somente o arquivo *.parquet* para os exercícios abaixo.

Questão 2: Preparando e modelando os dados.

Atenção: Elabore seus comandos dando preferência as funcionalidades do pacote `sparklyr`.

a) Faça uma breve análise exploratória dos dados (tabelas e gráficos) com base somente nas colunas existente nos arquivos de 1996. O dicionário das variaveis encontra-se no mesmo site do item a), na parte de documentação. Corrija eventuais erros encontrados; por exemplo, na variavel sexo são apresentados rótulos distintos para um mesmo significado.

```
dim <- sdf_dim(dfparquet)

glimpse <- glimpse(dfparquet)
```

```
## Rows: ??
## Columns: 21
## Database: spark_connection
## $ contador   <chr> "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "1~
## $ LOCNASC    <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",~
## $ CODMUNNASC <chr> "2922003", "2922003", "3205200", "3200102", "3204559", "320~
## $ IDADEMAE   <chr> "19", "18", "99", "30", "22", "27", "36", "25", "21", "25",~
## $ ESTCIVMAE  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ ESCMAE     <chr> "8", "7", "9", "6", "6", "6", "6", "7", "6", "6", "6", "6",~
## $ CODOCUPMAE <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ QTDFILVIVO <chr> NA, NA, "03", "02", "00", "0", "01", "0", "0", "01", "02", ~
## $ QTDFILMORT <chr> NA, NA, NA, "00", "00", NA, "00", NA, NA, "00", NA, NA, "00~
## $ CODMUNRES  <chr> "3204054", "3204054", "3205101", "3205101", "3205101", "320~
## $ GESTACAO   <chr> "5", "5", "5", "5", "5", "5", "5", "5", "5", "5", "5", "5",~
## $ GRAVIDEZ   <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",~
## $ PARTO      <chr> "2", "2", "2", "2", "1", "2", "2", "2", "2", "1", "2", "1",~
## $ CONSULTAS  <chr> "4", "4", "9", "4", "8", "4", "4", "4", "4", "1", "1", "1",~
## $ DTNASC     <chr> "16121996", "16121996", "28041996", "05121996", "11121996",~
## $ SEXO       <chr> "1", "1", "2", "2", "1", "2", "1", "2", "2", "2", "2", "2",~
## $ APGAR1     <chr> "09", "09", "08", "10", "05", "09", "09", "09", "08", "09",~
## $ APGAR5     <chr> "09", "09", "09", "10", "07", "10", "09", "10", "09", "10",~
## $ RACACOR    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ PESO       <chr> "3250", "3250", "3300", "2690", "2400", "3770", "3130", "37~
## $ CODANOMAL  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

```
na <- dfparquet %>%
  summarise_all(~sum(as.integer(is.na(.)))) %>%
  collect()

summary <- sdf_describe(dfparquet) %>%
  collect ()

###dfparquet %>%
###  group_by(SEXO) %>%
###  tally()
```

```r
dfparquet <- dfparquet %>%
  mutate(SEXO = case_when(SEXO == 'I' ~ 0,
                          SEXO == 0 ~ 0,
                          SEXO == 9 ~ 0,
                          SEXO == 'F' ~ 2,
                          SEXO == 2 ~ 2,
                          SEXO == 1 ~ 1,
                          SEXO == 'M' ~ 1))
dfparquet <- dfparquet %>%
  mutate(SEXO = ifelse(SEXO == 0,NA,SEXO))

###dfparquet %>%
###  group_by(SEXO) %>%
###  tally()

###dfparquet %>%
###  group_by(LOCNASC) %>%
###  tally()

dfparquet <- dfparquet %>%
  mutate(LOCNASC = ifelse(LOCNASC == 9 ,NA,LOCNASC))

###dfparquet %>%
###  group_by(ESTCIVMAE) %>%
###  tally()

###dfparquet %>%
###  group_by(ESCMAE) %>%
###  tally() %>%
###  print(n = Inf)

dfparquet <- dfparquet %>%
  mutate(ESCMAE = ifelse(ESCMAE == 9 | ESCMAE == 8 | ESCMAE == 7 | ESCMAE == 6 | ESCMAE == 0,NA,ESCM

###dfparquet %>%
###  group_by(GESTACAO) %>%
###  tally()

###dfparquet %>%
###  group_by(GRAVIDEZ) %>%
###  tally()

###dfparquet %>%
###  group_by(PARTO) %>%
###  tally()

###dfparquet %>%
###  group_by(CONSULTAS) %>%
###  tally()

###dfparquet %>%
###  group_by(APGAR1) %>%
###  tally() %>%
###  print(n = Inf)

dfparquet <- dfparquet %>%
  mutate(APGAR1 = case_when(APGAR1 == '07' ~ '07',
```

```
                            APGAR1 == '01' ~ '01',
                            APGAR1 == '.' ~ NA,
                            APGAR1 == '08' ~ '08',
                            APGAR1 == '09' ~ '09',
                            APGAR1 == '8' ~ '08',
                            APGAR1 == '10' ~ '10',
                            APGAR1 == '0' ~ '00',
                            APGAR1 == '9' ~ '09',
                            APGAR1 == '05' ~ '05',
                            APGAR1 == '06' ~ '06',
                            APGAR1 == '<e8>' ~ NA,
                            APGAR1 == NA ~ NA,
                            APGAR1 == '03' ~ '03',
                            APGAR1 == '7' ~ '07',
                            APGAR1 == '04' ~ '04',
                            APGAR1 == '00' ~ '00',
                            APGAR1 == '--' ~ NA,
                            APGAR1 == '..' ~ NA,
                            APGAR1 == '99' ~ NA,
                            APGAR1 == '02' ~ '02'
                            ))

###dfparquet %>%
###  group_by(APGAR1) %>%
###  tally() %>%
###  print(n = Inf)

###dfparquet %>%
###  group_by(APGAR5) %>%
###  tally() %>%
###  print(n = Inf)

dfparquet <- dfparquet %>%
  mutate(APGAR5 = case_when(APGAR5 == '07' ~ '07',
                            APGAR5 == '01' ~ '01',
                            APGAR5 == '.' ~ NA,
                            APGAR5 == '08' ~ '08',
                            APGAR5 == '\u0017' ~ NA,
                            APGAR5 == '09' ~ '09',
                            APGAR5 == '8' ~ '08',
                            APGAR5 == '10' ~ '10',
                            APGAR5 == '0' ~ '00',
                            APGAR5 == '9' ~ '09',
                            APGAR5 == '4' ~ '04',
                            APGAR5 == '05' ~ '05',
                            APGAR5 == '06' ~ '06',
                            APGAR5 == '5' ~ '05',
                            APGAR5 == NA ~ NA,
                            APGAR5 == '03' ~ '03',
                            APGAR5 == '04' ~ '04',
                            APGAR5 == '00' ~ '00',
                            APGAR5 == '--' ~ NA,
                            APGAR5 == '0-' ~ '00',
                            APGAR5 == '..' ~ NA,
                            APGAR5 == '99' ~ NA,
                            APGAR5 == '02' ~ '02',
                            APGAR5 == '39' ~ NA
  ))
```

```r
###dfparquet %>%
###  group_by(RACACOR) %>%
###  tally() %>%
###  print(n = Inf)

dfparquet <- dfparquet %>%
  mutate(RACACOR = case_when(RACACOR == 3 ~ 3,
                             RACACOR == 9 ~ NA,
                             RACACOR == 2 ~ 2,
                             RACACOR == 4 ~ 4,
                             RACACOR == 1 ~ 1,
                             RACACOR == 5 ~ 5,
                             RACACOR == NA ~ NA

  ))

###dfparquet %>%
###  group_by(IDADEMAE) %>%
###  tally() %>%
###  print(n = Inf)

dfparquet <- dfparquet %>%
  mutate(IDADEMAE = ifelse(IDADEMAE == '00' | IDADEMAE == '99', NA, IDADEMAE))
```

```r
sexoraca <- dfparquet %>%
  group_by(RACACOR,SEXO) %>%
  filter(RACACOR %in% 1:5 & SEXO %in% 1:2) %>%
  tally() %>%
  collect()

tipolocal <- dfparquet %>%
  group_by(LOCNASC,PARTO) %>%
  filter(LOCNASC %in% 1:3 & PARTO %in% 1:2) %>%
  tally() %>%
  collect ()

diasemana <-dfparquet %>%
  select(DTNASC) %>%
  collect()

diasemana <- as.vector(diasemana$DTNASC)
diasemana <- lubridate::wday(lubridate::dmy(diasemana),  label = TRUE)
diasemana <- factor(diasemana)

diasemana1 <- diasemana

histograma <- dfparquet %>%
  select(PESO) %>%
  filter(PESO <= '7.500') %>%
  dbplot_histogram(PESO, bins = 15)

estcivil <- dfparquet %>%
  group_by(ESTCIVMAE) %>%
  filter(ESTCIVMAE %in% 1:4) %>%
  tally() %>%
  collect()
```

```r
correlacao1 <- dfparquet %>%
  select(IDADEMAE,PESO) %>%
  filter(PESO <= '7.500') %>%
  mutate(IDADEMAE = as.integer(IDADEMAE)) %>%
  mutate(PESO = as.numeric(PESO)) %>%
  collect()

correlacao1 <- correlate(correlacao1)
correlacao1 <- as.numeric(correlacao1$IDADEMAE[2])

diasemanatibble <- as_tibble(diasemana1)
diasemanatibble <- diasemanatibble %>%
  mutate(value = ifelse(value == '"s\u00e1b"', 'sáb', value)) %>%
  group_by(value) %>%
  tally ()
diasemanatibble <- diasemanatibble %>%
  mutate(value = case_when(value == 1 ~ "dom",
                           value == 2 ~ "seg",
                           value == 3 ~ "ter",
                           value == 4 ~ "qua",
                           value == 5 ~ "qui",
                           value == 6 ~ "sex",
                           value == 7 ~ "sab"

  ))

dia_freq <- diasemanatibble$n
dia_freq <- dia_freq[1:7]

#coeficiente de variação dos dias da semana:
vardiasem <- sd(dia_freq)/mean(dia_freq)*100

tipolocal <- tipolocal %>%
  mutate(LOCNASC = case_when(LOCNASC == 1 ~ "Hospital",
                           LOCNASC == 2 ~ "Outro Estab Saúde",
                           LOCNASC == 3 ~ "Domicílio")) %>%
  mutate(PARTO = case_when(PARTO == 1 ~ "Vaginal",
                           PARTO == 2 ~ "Cesáreo"))


colnames(tipolocal) <- c("Local de Nascimento","Tipo de parto","Contagem")

estcivil <- estcivil %>%
  mutate(ESTCIVMAE = case_when(ESTCIVMAE == 1 ~ "Solteira",
                           ESTCIVMAE == 2 ~ "Casada",
                           ESTCIVMAE == 3 ~ "Viúva",
                           ESTCIVMAE == 4 ~ "Separado judicialmente/Divorciado"))

colnames(estcivil) <- c("Estado civil da mãe","Contagem")

peso <- dfparquet %>%
  select(PESO) %>%
  filter(PESO <= '7.500')%>%
  collect()

peso$PESO <- as.numeric(peso$PESO)
peso <- peso$PESO
```

```r
#coeficiente de variação:
cvpeso <- sd(peso,na.rm=TRUE)/mean(peso,na.rm=TRUE)*100

#Coeficiente de assimetria de pearson:
AS<-3*(mean(peso,na.rm=TRUE) - median(peso,na.rm=TRUE))/sd(peso,na.rm=TRUE)

#coef momento de assimetria
cma <- skewness(peso, na.rm = TRUE, method = "moment", l.moment.method = "unbiased")

#Coeficiente momento de curtose
cmc <- kurtosis(peso, na.rm = TRUE, method = "moment", l.moment.method = "unbiased")

medidaspeso <- c(min(peso,na.rm=TRUE),as.numeric(quantile(peso,na.rm=TRUE)[2]),
                 median(peso,na.rm=TRUE),mean(peso,na.rm=TRUE),
                 as.numeric(quantile(peso,na.rm=TRUE)[4]),max(peso,na.rm=TRUE),
                 IQR(peso,na.rm=TRUE),sd(peso,na.rm=TRUE),var(peso,na.rm=TRUE),
                 cvpeso,AS,cma,cmc)

labels <- c("Mínimo","1º Quartil","Mediana","Média","3º Quartil","Máximo",
            "Distância Interquartílica","Desvio padrão","Variância",
            "Coeficiente de Variação","Coeficiente de assimetria de pearson",
            "Coeficiente momento de assimetria","Coeficiente momento de curtose")

labels <- as_tibble(labels)

labels$est <- NA

labels$est <- medidaspeso

colnames(labels) <- c("Estatísticas","Valores")

summary <- summary %>%
  select(summary,IDADEMAE,APGAR1,APGAR5,PESO)

sexoraca <- sexoraca %>%
  mutate(RACACOR = case_when(RACACOR == 1 ~ "Branca",
                             RACACOR == 2 ~ "Preta",
                             RACACOR == 3 ~ "Amarela",
                             RACACOR == 4 ~ "Parda",
                             RACACOR == 5 ~ "Indígena"
                             )) %>%
  mutate(SEXO = case_when(SEXO == 1 ~ "Masculino",
                          SEXO == 2 ~ "Feminino"))

colnames(sexoraca) <- c("Raça/Cor","Sexo","Contagem")
```

1. Introdução Nesse ambiente contemporâneo onde o volume de dados crescente e a necessidade de tomada de decisões acertadas cada vez em um tempo menor, a estatística vem ganhando cada vez mais importância. Esse cenário se reflete tanto no setor público quanto no setor privado. Falando especificamente para o setor público, segundo **Ignácio** *(2010)* os gestores públicos estão se tornando cada vez mais dependentes de dados estatísticos para obter informações essenciais que auxiliem suas análises sobre a conjuntura econômica e social. As informações estatísticas devem ser concisas, específicas e eficazes, fornecendo, assim, subsídios imprescindíveis para a tomada de decisão. Neste sentido, a estatística fornece ferramentas importantes para que os governos possam definir melhor suas metas, avaliar sua performance, identificar seus pontos fortes e fracos e atuar na melhoria contínua das políticas públicas.

Dentro desse cenário de crescimento do protagonismo da estatística no auxílio na tomada de decisão e a busca de ferramentas que possibilitem aumentar a eficiência das políticas públicas é que foi criado o **SINASC** (Sistema de informações sobre os nascidos vivos). O **SINASC** é um sistema criado pelo **DATASUS** com o objetivo de reunir informações epidemiológicas referentes aos nascimentos informados em todo território nacional. Nele constam dados socioeconômicas dos pais, dados sobre no nascimento da criança, local de nascimento e etc.

No Brasil, segundo dados do **Unicef** *(2017)* o elevado número de cesarianas no país coloca o Brasil em segundo lugar no mundo em percentual deste tipo de parto. Enquanto a Organização Mundial da Saúde *(OMS)* estabelece em até 15% a proporção recomendada de partos por cesariana, no Brasil esse percentual é de 57%. As cesarianas representam 40% dos partos realizados na rede pública de saúde. Já na rede particular, chegam a 84% dos partos. Todavia, sendo que 35% dos bebês analisados nasceram entre as 37^a e 38^a semanas de gestação, uma fase que tecnicamente não é considerada prematura. Porém, as últimas semanas de gestação permitem maior ganho de peso, maturidade cerebral e pulmonar e para a mãe o trabalho de parto propicia a liberação de hormônios que irão prepará-la para a amamentação. Tais dados reforçam a necessidade de analisarmos as circunstâncias dos nascimentos Brasil.

Diante disso, qual é o perfil dos nascimentos ocorridos nos estados de **Goiás**, **Mato Grosso do Sul** e **Espírito Santo**? Será que o tipo de parto varia conforme o dia da semana? Será que há relação entre a idade da mãe e o peso do recém-nascido? Para isso, farei uma análise exploratória de dados das informações constantes no **SINASC** dos anos de 1994 à 2020 dentro dos seguintes aspectos:

1. **Verificar se o número de partos varia entre os dias da semana;**
2. **Averiguar o percentual de mães solteiras;**
3. **Examinar a variável peso dos recém-nascidos;**
4. **Verificar a relação entre o peso do recém-nascido e idade da mãe**

2. Objetivos

2.1 Objetivo geral:

1. **Verificar o perfil dos nascimentos ocorridos nos estados de Goiás, Mato Grosso do Sul e Espírito Santo nos anos de 1994 à 2020.**

2.2 Objetivos específicos

1. **Verificar se o número de partos varia entre os dias da semana;**
2. **Verificar o estado civil das mães;**
3. **Examinar a variável peso dos recém-nascidos;**
4. **Verificar a relação entre o peso do recém-nascido e idade da mãe**

3. Métodos O banco de dados utilizado foi o presente no site do **SINASC**, referente os dados constantes na declaração de nascidos vivos nos estados de **Goiás**, **Mato Grosso do Sul** e **Espírito Santo** de 1994 à 2020. A declaração de nascido vivo é gerada quando a criança nasce viva. Os casos de nascimento com o bebê morto não entram na análise. O arquivo foi importado no seguinte link: (https://datasus.saude.gov.br/transferencia-de-arquivos/). No banco de dados havia dados de todas Unidades da Federação.

O banco de dados contava com 4.760107×10^6 observações de indivíduos nascidos vivos nos anos e estados selecionados. Desses, 4699600 constavam como nascidos em hospitais.

4. Análise dos Resultados

4.1 Análise dos nascimentos quanto aos dias da semana O coeficiente de variação nos dados (13.5210637%) evidencia razoável homogeneidade quanto ao dia do nascimento.

Indicador:

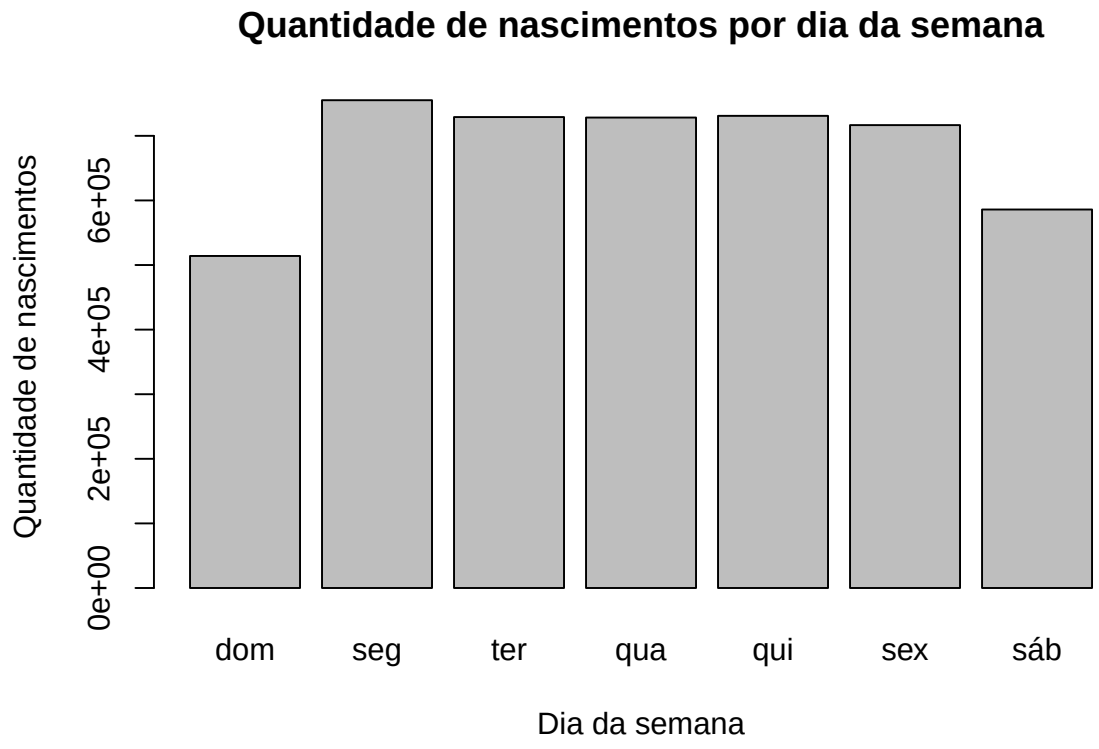
Desvio-Padrão: 9.1942036×10^4

Média: 6.7999114×10^5

Coef.var: 13.5210637

Gráfico:

```
plot(diasemana1,
     xlab='Dia da semana',
     ylab='Quantidade de nascimentos',
     main='Quantidade de nascimentos por dia da semana',
     sub='')
```



Analisando o tipo de parto relacionado ao local de nascimento nós temos:

```
kable(tipolocal)
```

Local de Nascimento	Tipo de parto	Contagem
Hospital	Vaginal	2070725
Domicílio	Vaginal	14611
Outro Estab Saúde	Vaginal	16356
Hospital	Cesáreo	2628875
Outro Estab Saúde	Cesáreo	16094

Segundo **Ratner e Moura** *(2016)*, há uma preponderância dos dias de semana em relação ao fim de semana para a realização das cesarianas. Esse padrão foi consistente nos três anos estudados, 2000, 2005 e 2010, havendo uma pequena tendência de queda ao longo desses anos na proporção de nascimentos cirúrgicos nos finais de semana.

4.2 Perfil das gestantes quanto a seu estado civil Este se trata de um ponto importante da presente análise. Segundo um estudo de **Piccini et. al** *(2007)* considerando todos os níveis de análise realizados, foram verificadas diferenças expressivas entre os grupos das mães solteiras e casadas tanto nas seqüências responsivas, em particular com relação ao choro, como nas não-responsivas, quanto ao choro e as vocalizações, apontando que as mães solteiras foram menos responsivas do que as mães casadas. Além disso, verificou-se que embora os comportamentos maternos e infantis durante a interação, embora os resultados tenham mostrado semelhanças entre as mães e os bebês nas diversas categorias examinadas, o total de comportamentos maternos revelou um maior envolvimento das mães casadas com seus bebês. Tal situação pode demandar uma atenção maior do Estado para esse grupo de mães solteiras na educação de seus filhos. Dentro da pesquisa verificou-se uma grande concentração nas situações de união consensual/casada e solteira. O alto número de mães solteiras pode chamar a atenção quanto a necessidade de políticas especiais para esse grupo.

Dados da variável "Estado civil da mãe"

```
kable(estcivil)
```

Estado civil da mãe	Contagem
Viúva	9925
Casada	1607219
Separado judicialmente/Divorciado	47378
Solteira	1779522

A respeito do estado civil das mães, verificou-se que 46.6666221% eram casadas/união consensual, 51.6695489% eram solteiras, 1.3756503% eram separadas/divorciadas e 0.2881787% eram viúvas. O resultado converge razoavelmente para os resultados da pesquisa de **SCHILLER** (2015), onde identificou-se que 58,33% das mães eram casadas/união consensual, 40,46% de mulheres solteiras, apenas 1,03% de separadas/divorciadas; e, somente 0,19% de viúvas.

4.3 Análise da variável "peso do recém-nascido" Efetuei uma análise da variável peso. Os dados apresentados demonstram uma baixa dispersão em torno da média, conforme evidenciado na tabela abaixo:

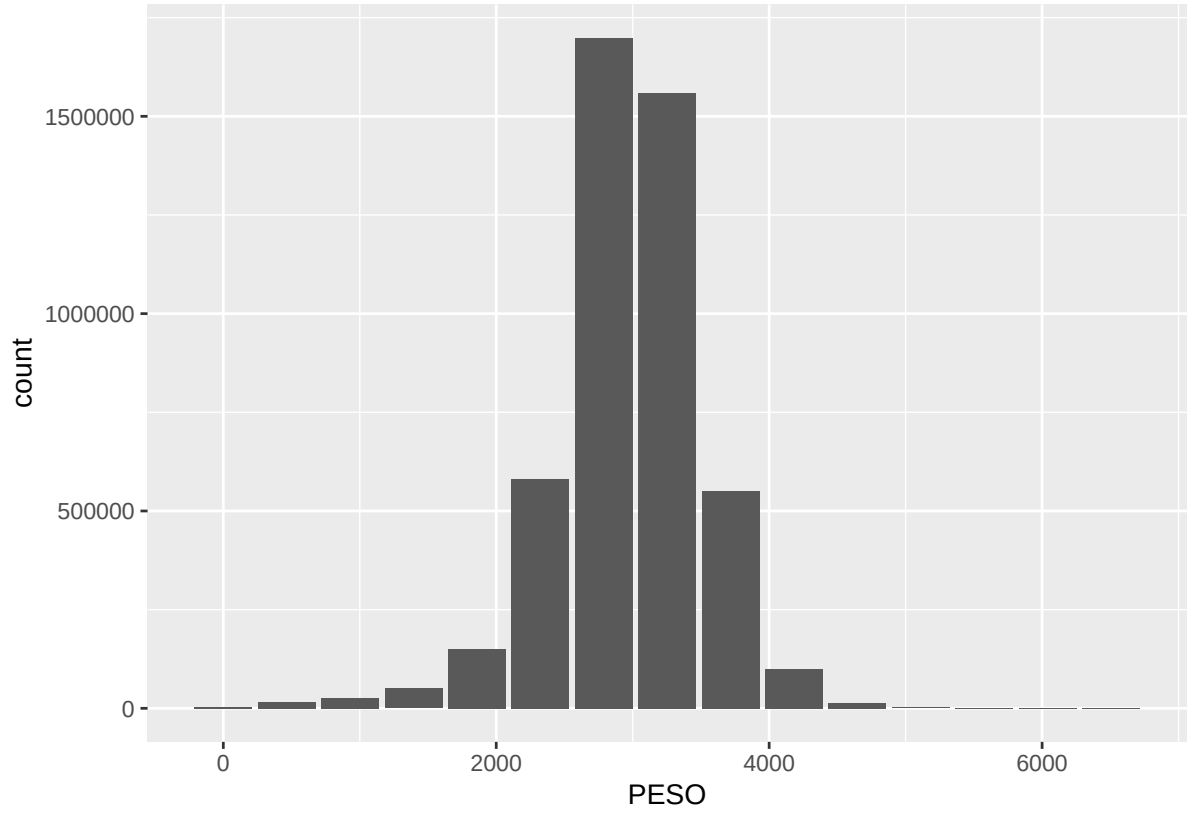
```
kable(labels)
```

Estatísticas	Valores
Mínimo	0.000000e+00
1º Quartil	2.910000e+03
Mediana	3.220000e+03
Média	3.197450e+03
3º Quartil	3.530000e+03
Máximo	6.970000e+03
Distância Interquartílica	6.200000e+02
Desvio padrão	5.406110e+02
Variância	2.922603e+05
Coeficiente de Variação	1.690757e+01
Coeficiente de assimetria de pearson	-1.251359e-01
Coeficiente momento de assimetria	-7.521480e-01
Coeficiente momento de curtose	2.769255e+00

Com base nas informações apontadas, a distribuição dos dados tanto da amostra quanto da população da variável "peso do recém-nascido" é uma distribuição leptocúrtica alongada, com uma leve assimetria à direita, o que demonstra que a média é menor que a mediana. Neste caso há preponderância de valores superiores à média.

Histograma da variável "peso do recém-nascido"

```
histograma
```



Avalio que estas situações deveriam ser analisadas de forma mais aprofundada. Por exemplo, a declaração de nascido vivo é preenchida se o recém-nascido nasce com vida, mas não analisa o que ocorre depois, há situações em que prematuros nascem com um peso muito baixo. Existem vários *outliers* nos dados tanto à esquerda quanto a direita.

4.4 Análise da relação entre o peso do recém-nascido e a idade da mãe O coeficiente de assimetria de Pearson entre as variáveis peso do recém-nascido e a idade da mãe apresentou um resultado de 0.044708, o que demonstra uma leve assimetria à direita ou positiva ($0 < x < 0.15$). Porém, pela insignificância do valor, pode-se concluir que não há assimetria.

Analisando a relação entre idade materna e peso do recém-nascido, verificou-se por meio do coeficiente de assimetria de *Pearson*, que a relação é fraca. Todavia, este resultado não converge para o estudo de **Ximenes Aragão e Oliveira** *(2004)*. Segundo as autoras, a associação entre a idade materna e o peso ao nascer demonstra a ocorrência de baixo peso e peso insuficiente diminuindo à medida do aumento da idade materna, revertendo essa tendência quando a idade atinge 35 anos.

Há três hipóteses para explicar o baixo peso ao nascer em filhos de mães adolescentes: desvantagem social, imaturidade biológica e estilo de vida inadequado durante a gravidez. Com relação aos fatores associados ao baixo peso ao nascer entre os filhos de parturientes acima de 40 anos, como prováveis a hipertensão arterial, ruptura prévia de placenta, cardiopatias e infecção urinária.

O coeficiente de correlação apresentou o resultado de 0.044708, o que significa uma baixa (quase nula) relação positiva entre a idade da mãe e o peso do recém-nascido. Este resultado também discorda dos resultados obtidos nos estudos de **Ximenes Aragão e Oliveira** *(2004)* mencionados anteriormente.

```
kable(sexoraca)
```

4.5 Tabela de quantidade de nascidos por sexo e raça:

Raça/Cor	Sexo	Contagem
Amarela	Feminino	10627
Parda	Feminino	991332
Branca	Masculino	874978
Preta	Masculino	58775
Branca	Feminino	856746
Parda	Masculino	1063789
Amarela	Masculino	11343
Preta	Feminino	53874
Indígena	Masculino	25224
Indígena	Feminino	23810

5. Discussão A presente pesquisa teve como objetivo averiguar perfil dos nascimentos ocorridos nos estados de **Goiás**, **Mato Grosso do Sul** e **Espírito Santo** de 1994 à 2020. Para isso, efetuei uma análise exploratória de dados. Com isso, foi possível examinar os dados dos nascimentos nos dias da semana, para verificar se há algum dia da semana preferido para os nascimentos, e o estado civil das mães, para ter um retrato do estado civil das mães. Também verifiquei tipos de parto (normal e cesárea) e a sua relação com o local de nascimento.

No que tange os objetivos da presente análise, identificou-se:

1. **Baixa correlação entre os nascimentos e os dias das semanas;**
2. **A preponderância das mães casadas/união estável e solteiras;**
3. **Peso médio dos recém-nascidos de 3197g, com baixa dispersão em torno da média;**
4. **baixa relação entre peso do recém-nascido e a idade da mãe.**

Para uma análise mais aprofundada é necessário que se estude mais variáveis que influenciam nas decisões de nascimentos dos recém-nascidos e no pós nascimento também. Atualmente há disponibilidade de bancos de dados robustos que possibilitam várias análises que ajudam a mostrar o retrato da saúde pública.

Referências Bibliográficas:

1. **Ignácio, Sérgio Aparecido. Importância da Estatística para o processo de conhecimento e tomada de decisão. Revista Paranaense de Desenvolvimento. Curitiba, n.118, p.175-192, jan./jun. 2010**
2. **UNICEF: Unicef chama atenção para a importância do trabalho de parto espontâneo. Disponível em:** $https://www.unicef.org/brazil/pt/media_35978.html$**.**
3. **Rattner, Daphne e Moura; Erly, Catarina de. Nascimentos no Brasil: associação do tipo de parto com variáveis temporais e sócio demográficas. Rev. Bras. Saúde Matern. Infant., Recife, 16 (1): 39-47 jan. / Mar., 2016**
4. **Haidar, Fátima Hussein; Oliveira, Urânia Fernandes; Nascimento, Luiz Fernando Costa. Escolaridade materna: correlação com os indicadores obstétricos. Cad. Saúde Pública, Rio de Janeiro, 17(4):1025-1029, jul-ago, 2001**
5. **Ximenes Aragão, Fernanda Maria; Oliveira, Mylza Carvalho Rosado de. A influência da idade materna sobre as condições perinatais. Revista Brasileira em Promoção da Saúde. vol. 17, núm. 2, 2004, pp. 56-60 Universidade de Fortaleza Fortaleza-Ceará, Brasil**

b) Utilizando as funções do `sparklyr`, preencha os dados faltantes na idade da mãe com base na mediana.Se necessário, faça imputação de dados também nas demais váriaveis.

```
###dfparquet %>%
###  group_by(IDADEMAE) %>%
###  tally() %>%
###  print(n = Inf)

###dfparquet %>%
###  mutate(IDADEMAE = as.integer(IDADEMAE)) %>%
###  sdf_quantile(
###    column = "IDADEMAE",
###    probabilities = c(0.5),
###    relative.error = 0.01)

# O valor é 25.

dfparquet <- dfparquet %>%
  mutate(IDADEMAE = as.integer(IDADEMAE)) %>%
  mutate(IDADEMAE = if_else(is.na(IDADEMAE), 25 , IDADEMAE))

###dfparquet %>%
###  mutate(PESO = as.integer(PESO)) %>%
###  sdf_quantile(
###    column = "PESO",
###    probabilities = c(0.5),
###    relative.error = 0.01)

# O valor é 3210

dfparquet <- dfparquet %>%
  mutate(PESO = as.integer(PESO)) %>%
  mutate(PESO = if_else(is.na(PESO), 3210 , PESO))
```

c) Novamente, utilizando as funções do `sparklyr`, normalize (retire a média e divida pelo desvio padrão) as variáveis quantitativas do banco.

```
scale_values <- dfparquet %>%
  summarize(
    mean_PESO = mean(PESO),
    mean_IDADEMAE = mean(IDADEMAE),
    sd_PESO = sd(PESO),
    sd_IDADEMAE = sd(IDADEMAE)
  ) %>%
  collect()

dfparquet <- dfparquet %>%
  mutate(scaled_PESO = (PESO - local(scale_values$mean_PESO)) /
           !!scale_values$sd_PESO,
         scaled_IDADEMAE = (IDADEMAE - !!scale_values$mean_IDADEMAE) /
           !!scale_values$sd_IDADEMAE)
```

d) Crie variáveis *dummy* (*one-hot-encoding*) que conjuntamente indiquem o dia da semana do nascimento. Em seguida,binarize o número de consultas pré-natais de modo que "0" represente "até 5 consultas" e "1" indique "6 ou mais consultas". (Ultilize as funções ft_)

```
diasemana <- as.data.frame(diasemana)

diasemana <- tibble::rowid_to_column(diasemana, "id")

dfparquet <- sdf_with_sequential_id(dfparquet, id = "id", from = 1L)

diasemana <- copy_to(sc, as.data.frame(diasemana), name = deparse(substitute(diasemana)),
        memory = TRUE, repartition = 0, overwrite = TRUE)

dfparquet <- left_join(dfparquet,diasemana)

###dfparquet %>%
###  group_by(diasemana) %>%
###  tally() %>%
###  print(n = Inf)

###sdf_dim(dfparquet)

dfparquet <- dfparquet %>%
  mutate(CONSULTAS = ifelse(CONSULTAS == '8' | CONSULTAS == '9', NA, CONSULTAS))

###dfparquet %>%
###  group_by(CONSULTAS) %>%
###  tally()

dfparquet <- dfparquet %>%
  mutate(CONSULTAS = as.numeric(CONSULTAS)) %>%
  ft_binarizer(
    input_col = "CONSULTAS",
    output_col = "CONSULTAS_bin",
    threshold = 2

  )

###dfparquet %>%
###  group_by(CONSULTAS_bin) %>%
###  tally()
```

e) Particione os dados aleatoriamente em bases de treinamento e teste. Ajuste, sobre a base de treinamento, um modelo de regressão logistica em que a variável resposta (y), indica se o parto foi ou não cesáreo. Analise o desempenho preditivo do modelo com base na matriz de confusão obtida no conjunto de teste.

OBS A *chunk* dos cálculos dessa questão, apesar de funcionar no ambiente `R`, impede o documento de *knitar* neste arquivo *.Rmd*. Portanto, incluirei aqui somente os resultados, e os códigos (resumidos) utilizados no formato de comentário

```
#partition <- dfparquet %>%
#  sdf_random_split(training = 0.85, test = 0.15, seed = 150167636)

#data_training <- sdf_register(partition$train, "df_train")
#data_test <- sdf_register(partition$test, "df_test")

#tbl_cache(sc, "df_train")
#tbl_cache(sc, "df_test")

#formula <- ('PARTO ~ LOCNASC + IDADEMAE + ESCMAE + SEXO +
#            APGAR1 + APGAR5 + RACACOR + CONSULTAS_bin + PESO')

#lr2_model <- ml_logistic_regression(x= data_training,formula=formula)
#validation_summary <- ml_evaluate(lr2_model, data_test)
#pred <- ml_predict(lr2_model, data_test)

#rf_model <- data_training %>%
#  ml_random_forest(formula, type = "classification")
#pred4 <- ml_predict(rf_model, data_test)

#lr3_model <- ml_generalized_linear_regression(x= data_training,formula=formula)
#validation_summary3 <- ml_evaluate(lr3_model, data_test)
#pred3 <- ml_predict(lr3_model, data_test)
#pred3 <- pred3 %>%
#  mutate(PARTO = as.numeric(PARTO))
```

Resultados:

```
# ml_multiclass_classification_evaluator(pred)

# [1] 0.6451208

# ml_multiclass_classification_evaluator(pred4)

# [1] 0.6289712

# ml_regression_evaluator(pred3,label_col="PARTO")

# [1] 1.297908
```

A íntegra dos arquivos utilizados neste trabalho encontra-se em meu github