

```
#####PROTOCOLO DE CRÍTICA DE DADOS
```

```
##ETAPAS INICIAIS
```

```
##1- Instalar os pacotes utilizados para organização do banco
```

```
install.packages ("readxl")  
install.packages ("dplyr")  
install.packages ("data.table")  
install.packages ("tidyr")  
install.packages ("svDialogs")
```

```
##2- Ativar os pacotes na biblioteca do R
```

```
library (readxl)  
library(dplyr)  
library(data.table)  
library(tidyr)  
library (svDialogs)
```

```
##3-Identificar o diretório de trabalho
```

```
setwd(dlg_dir(default = getwd())$res)  
z<- getwd()
```

```
##4-Ler o banco disponibilizado pelo Ministério da Saúde
```

```
banco <- rstudioapi::selectFile(caption = "Select xlsx File",  
                                filter = "xlsx Files (*.xlsx)",  
                                existing = TRUE)
```

```
Amostra <- read_xlsx(banco)
```

```
##5-Transformar separador de vírgula para ponto
```

```
Amostra$`Fluoreto (mg/L)` <-  
gsub(',', '.', Amostra$`Fluoreto (mg/L)`)
```

```
##6-Certificar a variável Fluoreto está como numérica
```

```
class(Amostra$`Fluoreto (mg/L)`)  
Amostra$`Fluoreto (mg/L)`<- as.numeric (Amostra$`Fluoreto (mg/L)`)  
##checar
```

```
##7-Renomear coluna 'Código IBGE' e 'Fluoreto mg/L' para evitar  
erros no decorrer do script
```

```
colnames(Amostra)[3]<-'C?digo_IBGE'  
colnames(Amostra)[16]<-'Fluoreto'
```

```
##8-Exclusão de 5 regiões administrativas DF
```

```

i <- 530002
while (i < 530032) {

  Amostra <- subset(Amostra, !C?digo_IBGE == i)

  i = i + 1

}

Amostra <- subset(Amostra, !Código_IBGE==530000)

###ETAPAS DA LIMPEZA DO BANCO

##1-Identificar e deletar municípios com menos de 4 meses de
registro

#1.1-Classificar a variável 'Data de Coleta' como 'date'
Amostra$`Data da coleta`<- as.Date (Amostra$`Data da coleta`)
class(Amostra$`Data da coleta`)

#1.2-Criar uma variável com o mês da coleta
Amostra$M?sR <- month(Amostra$`Data da coleta`)

#1.3- Usar a função duplicate para identificar amostras
realizadas no mesmo mês
Amostra$months <-!duplicated(Amostra[,c("M?sR","C?digo_IBGE")])

#1.4- Criar um dataframe com a informação de quantos meses foram
analisados em cada município
temp1 <- Amostra %>%
  group_by(C?digo_IBGE) %>%
  summarise(months = sum(months))

#1.5- Classificar como TRUE os municípios com < 4 meses de
analise
temp1$menosque4meses <- temp1$months < 4

#1.6- Unir a variável MoreThan3months no banco original
Amostra <- merge(Amostra,temp1,by='C?digo_IBGE')

#1.7- Eliminar os municípios identificados como TRUE para < 4
meses
Amostra <- Amostra[!(Amostra$menosque4meses == TRUE),]
ungroup(Amostra)

##2- Deletar laudos zerados
Amostra2 <- subset(Amostra, Fluoreto!=0)

```

##3- Remover outliers

#3.1-Fórmula de identificação de Outlier

```
remove_outliers <- function(x, na.rm = TRUE, ...) {  
  qnt <- quantile(x, probs=c(.25, .75), na.rm = na.rm, ...)  
  H <- 1.5 * IQR(x, na.rm = na.rm)  
  y <- x  
  y[x < (qnt[1] - H)] <- NA  
  y[x > (qnt[2] + H)] <- NA  
  y  
}
```

#3.2- Se certificar que o objeto está como numérico

```
Amostra2$Fluoreto <- as.numeric (Amostra2$Fluoreto)  
class(Amostra2$Fluoreto)
```

#3.3-Aplicar a fórmula de remoção de outliers

```
Amostra3 <- Amostra2 %>%  
  group_by(C?digo_IBGE) %>%  
  mutate(Fluoreto = remove_outliers(Fluoreto)) %>%  
  drop_na(Fluoreto)
```

##4-Exportar os arquivos resultantes de cada etapa de filtro

```
write.table(Amostra, "./Amostra.csv", sep=';', dec=',',  
  row.names=FALSE)  
write.table(Amostra2, "./Amostra2.csv", sep=';', dec=',',  
  row.names=FALSE)  
write.table(Amostra3, "./Amostra3.csv", sep=';', dec=',',  
  row.names=FALSE)
```

###ETAPAS DE CONSTRUÇÃO DOS INDICADORES

#1- Calcular a média e desvio padrão do fluoreto por município

```
mean <- aggregate(Fluoreto ~ C?digo_IBGE, Amostra3, FUN = mean,  
  na.rm = TRUE)  
sd <- aggregate(Fluoreto ~ C?digo_IBGE, Amostra3, FUN = sd,  
  na.rm = TRUE)
```

#1.1- Renomear as colunas com a média (mean) e desvio-padrão (sd)

```
colnames(mean) <- c ("C?digo_IBGE", "mean")  
colnames(sd) <- c ("C?digo_IBGE", "sd")
```

#1.2- Unir informações da média e desvio-padrão em um único dataframe

```
final <- merge(mean, sd, by="C?digo_IBGE")
```

```

#1.3- Criar colunas com valores arredondados para 3 casas
decimais
final$finalmean <- round (final$mean, digits=3)
final$finalsd <- round (final$sd, digits=3)

##2- Categorizar os registros de concentração de fluoreto
Amostra3$fFluoreto[0<=Amostra3$Fluoreto&Amostra3$Fluoreto<0.445]
<-'0,001-0,444'
Amostra3$fFluoreto[0.445<=Amostra3$Fluoreto&Amostra3$Fluoreto<0.
945]<-'0,445-0,944'
Amostra3$fFluoreto[0.945<=Amostra3$Fluoreto&Amostra3$Fluoreto<1.
445]<-'0,945-1,444'
Amostra3$fFluoreto[1.445<=Amostra3$Fluoreto&Amostra3$Fluoreto<12
]<-'Maior que 1,444'

#2.1- Nomear as categorias da variável como "Baixo, Ideal, Alto,
Muito Alto"
Amostra3$fFluoreto <- factor(Amostra3$fFluoreto, levels =
c("0,001-0,444", "0,445-0,944", "0,945-1,444", "Maior que
1,444"), labels = c("Baixo", "Ideal", "Alto", "Muito Alto"))

##3- Calcular a proporção e número de registros em cada
categoria
proporcao <- Amostra3 %>%
  count(C?digo_IBGE, fFluoreto) %>%
  group_by(C?digo_IBGE) %>%
  mutate(prop = prop.table(n))

proporcao_final <- with(proporcao, tapply(prop,
list(C?digo_IBGE, fFluoreto), sum))
n_final <- with(proporcao, tapply(n, list(C?digo_IBGE,
fFluoreto), sum))

#3.1- Unir as duas informações em um único dataframe

#3.1.1- Proporção
final2 <- (merge.r.base2 <- merge(final, proporcao_final, by.x =
"C?digo_IBGE", by.y = "row.names"))

#3.1.2- Número de amostras
final3 <- merge(final2, n_final, by.x = "C?digo_IBGE", by.y =
"row.names")

###4- Organizar os dados por município

#4.1- Inserir informações sobre a UF na planilha final 3

```

```

#4.1.2- Criar um banco de dados somente com Código e UF das amostras
UF <- data.frame(Amostra3$C?digo_IBGE, Amostra3$UF)

#4.1.3- Agrupar os municípios
UF<- UF %>%
  group_by(Amostra3.C?digo_IBGE, Amostra3.UF) %>%
  count()

#4.1.4- Renomear a coluna "Código_IBGE" do database
UF <- rename(UF, C?digo_IBGE = Amostra3.C?digo_IBGE)

#4.1.5- Renomear a coluna "UF" do database
UF <- rename(UF, UF = Amostra3.UF)

#4.1.6- Renomear a coluna "N_amostra" do database
UF <- rename(UF, N_amostra = n)

#4.1.7- Organizar o banco de dados com: UF + Código IBGE, média e desvio padrão da concentraç?o de fluoreto, N e proporção de amostras em cada categoria.
final4 <- merge (final3, UF, by ="C?digo_IBGE")

##4.1.8- Categorizar as UF por macrorregiões
final4$Regi?o <- factor(final4$UF, levels = c("AC", "AM", "RO", "RR", "AP", "PA", "TO", "MA", "PI", "CE", "RN", "PE", "PB", "SE", "AL", "BA", "MT", "MS", "GO", "DF", "SP", "MG", "ES", "RJ", "PR", "SC", "RS"), labels = c("NORTE", "NORTE", "NORTE", "NORTE", "NORTE", "NORTE", "NORTE", "NORTE", "NORDESTE", "NORDESTE", "NORDESTE", "NORDESTE", "NORDESTE", "NORDESTE", "NORDESTE", "NORDESTE", "CENTRO-OESTE", "CENTRO-OESTE", "CENTRO-OESTE", "CENTRO-OESTE", "SUDESTE", "SUDESTE", "SUDESTE", "SUDESTE", "SUDESTE", "SUL", "SUL", "SUL", "SUL"))

##5- Organização das variáveis no banco final
final5 <- final4 %>%
  select(C?digo_IBGE, UF, Regi?o, finalmean, finalsd, Baixo.x, Ideal.x, Alto.x, `Muito Alto.x`, Baixo.y, Ideal.y, Alto.y, `Muito Alto.y`, N_amostra )

##6-Exportação do banco final5 como arquivo .csv
write.table(final5, "./Final5.2.csv", sep=';', dec=',', row.names=FALSE)
if (ok_cancel_box("Mostrar onde o banco foi salvo?"))
cat(dlg_message(z)$res) else cat("stop it!\n")

```

