

Exploração dos Dados

Bibliotecas

```
library(tidyverse)
library(data.table)
library(Hmisc)
library(knitr)
library(kableExtra)
library(ggcorrplot)
library(ggdendro)
library(dendextend)
library(biotools)
library(ggfortify)
library(factoextra)
library(yaml)
library(corrplot)
select = dplyr::select

options(OutDec= ",")
```

Carregando os dados

```
constants = yaml.load_file("constants.yaml")

categorical_columns = constants$categorical_columns

daily_columns = constants$daily_columns

numerical_columns = constants$numerical_columns

solutos_columns = constants$solutos_columns

all_columns = c(numerical_columns, categorical_columns)

original_columns = constants$original_columns

derived_columns = setdiff(daily_columns, original_columns)

df = readRDS('./data/dados_processados.rds') %>%
  mutate_at(categorical_columns, list(~factor(.)))

df_names = readxl::read_excel('./data/Nomes das variaveis.xlsx') %>%
  mutate(variavel = tolower(variavel),
         nome = coalesce(nome, variavel)) %>%
  select(nome, variavel)
```

Funções

```
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    rho = (cormat)[ut]
  )
}

highlyCorrelated = function(df, threshold = 0.8){
  df_cor <- df %>%
    as.matrix() %>%
    rcorr(type = "pearson")

  df_flatten = flattenCorrMatrix(df_cor$r, df_cor$p) %>%
    mutate(abs_rho = abs(rho))

  return(df_flatten %>% filter(abs_rho > threshold))
}

highlyCorrelatedByGroup = function(df, group_col, threshold=0.8){
  group_levels = levels(df[[group_col]])
  concat_df = tibble()

  for (level in group_levels){
    corr_df = df %>%
      filter(!sym(col) == level) %>%
      select(-all_of(group_col)) %>%
      highlyCorrelated(threshold = threshold) %>%
      mutate(group = col,
             group_level = level,
             size = dim(df %>% filter(!sym(col) == level))[1])

    concat_df = bind_rows(concat_df, corr_df)
  }
  return(concat_df)
}

niceFormatting = function(df, caption=""){
  df %>%
    kbl(booktabs = T, longtable = T, caption = caption, digits = 2) %>%
    kable_styling(latex_options = c("striped", "HOLD_position", "repeat_header"))
}

rename_matrix = function(df){
  rownames(df) = tibble(variavel = cbind(rownames(df))) %>%
    inner_join(df_names, by='variavel') %>%
    .$nome

  colnames(df) = tibble(variavel = cbind(colnames(df))) %>%
    inner_join(df_names, by='variavel') %>%
```

```
    .$nome  
    return(df)  
}
```

Correlação

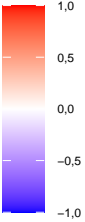
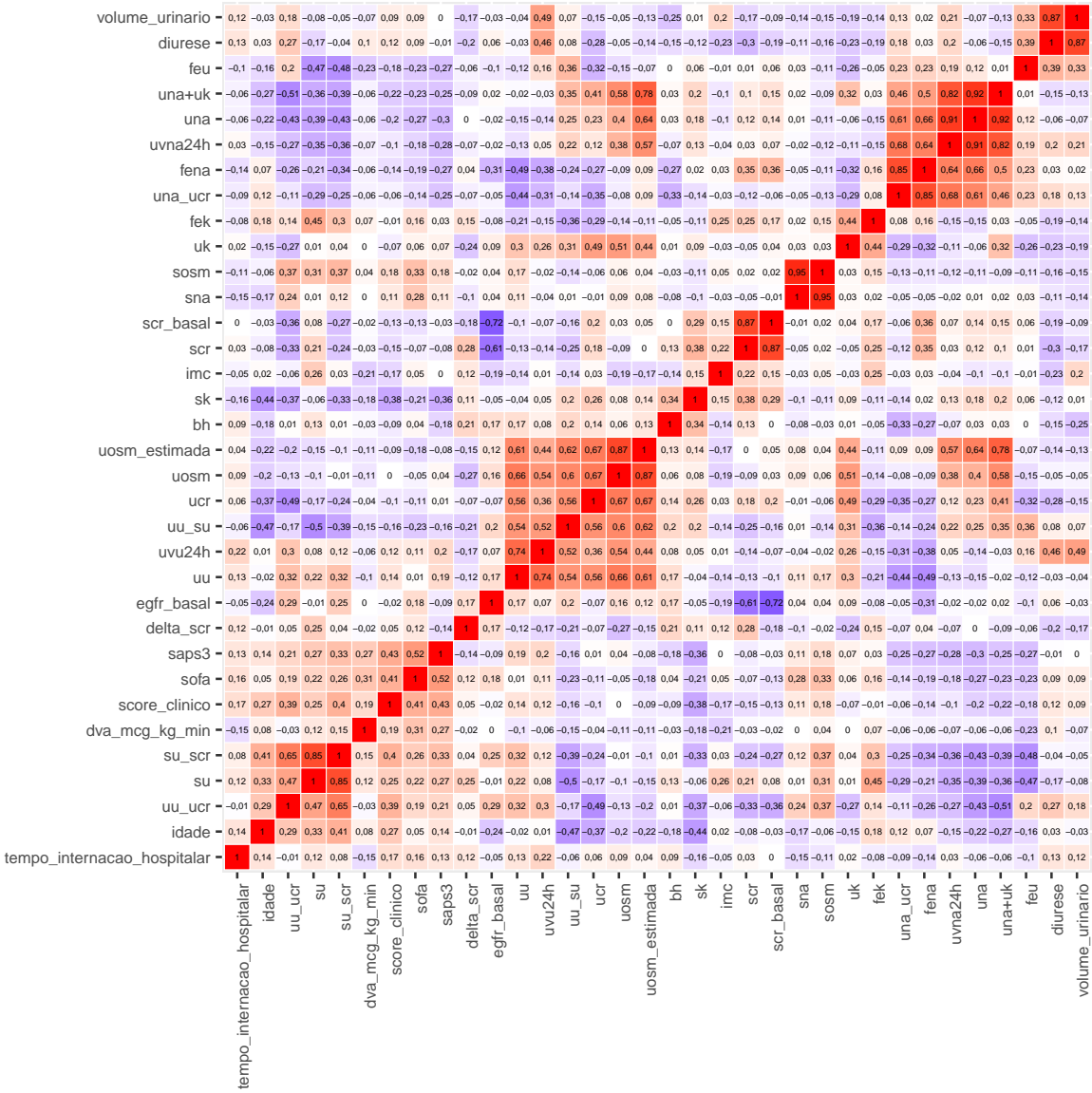
```

corr = df %>%
  select(all_of(numerical_columns)) %>%
  drop_na %>%
  cor %>%
  as.matrix

ggcorrplot(corr, hc.order = TRUE, outline.col = "white", type = 'full',
            tl.cex = 7, tl.srt = 90, lab = TRUE, lab_size = 1.5,
            ggtheme = theme(legend.position="right",
                           legend.title=element_blank(),
                           legend.text=element_text(size=5))) +
  labs(subtitle = "Todos pacientes")

```

Todos pacientes



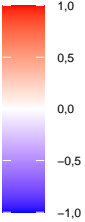
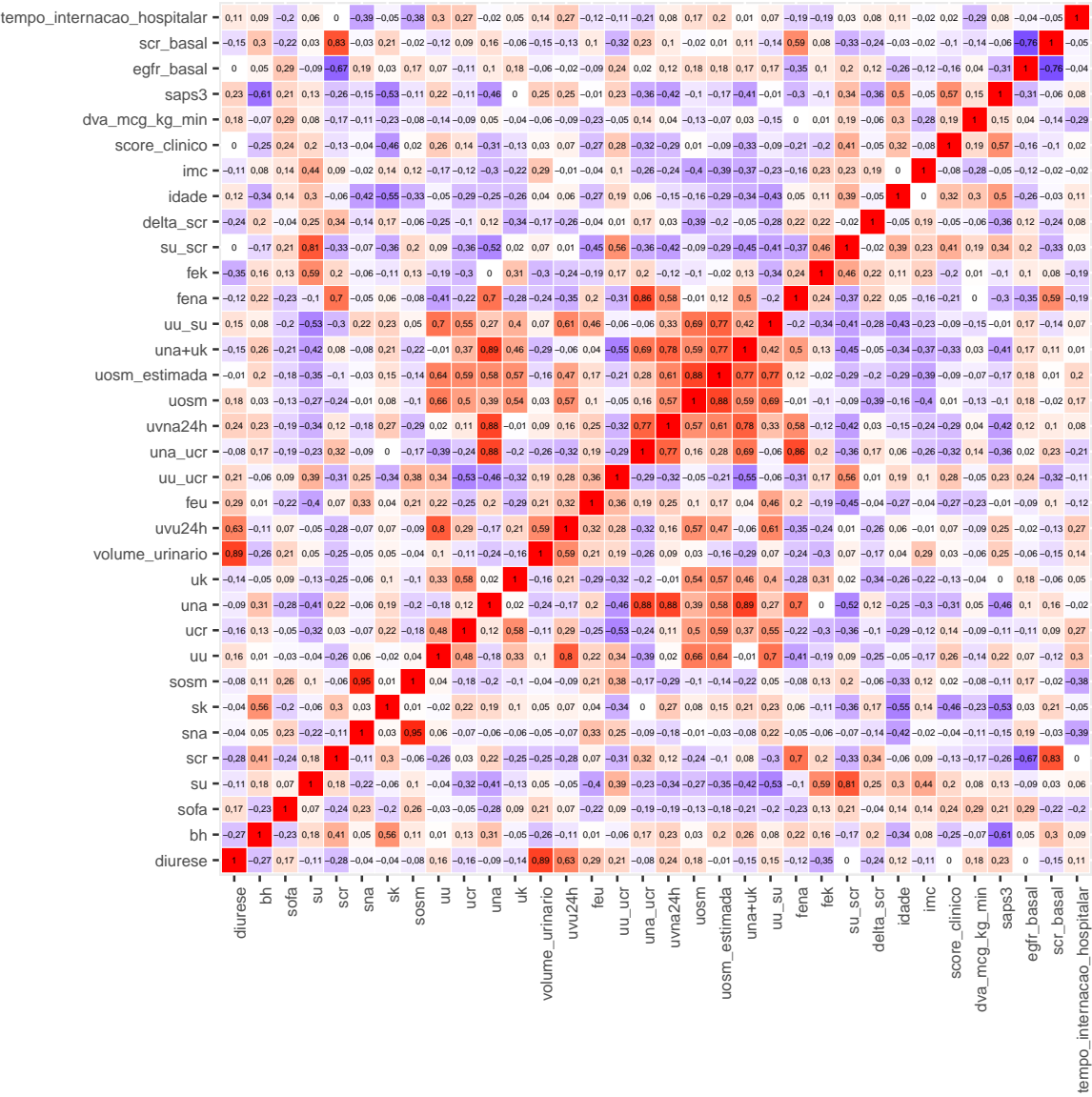
```

corr = df %>%
  filter(ira == 1) %>%
  select(all_of(numerical_columns)) %>%
  drop_na %>%
  cor %>%
  as.matrix

ggcorrplot(corr, hc.order = F, outline.col = "white", type = 'full',
            tl.cex = 7, tl.srt = 90, lab = TRUE, lab_size = 1.5,
            ggtheme = theme(legend.position="right",
                           legend.title=element_blank(),
                           legend.text=element_text(size=5))) +
  labs(subtitle = "Pacientes que desenvolveram IRA")

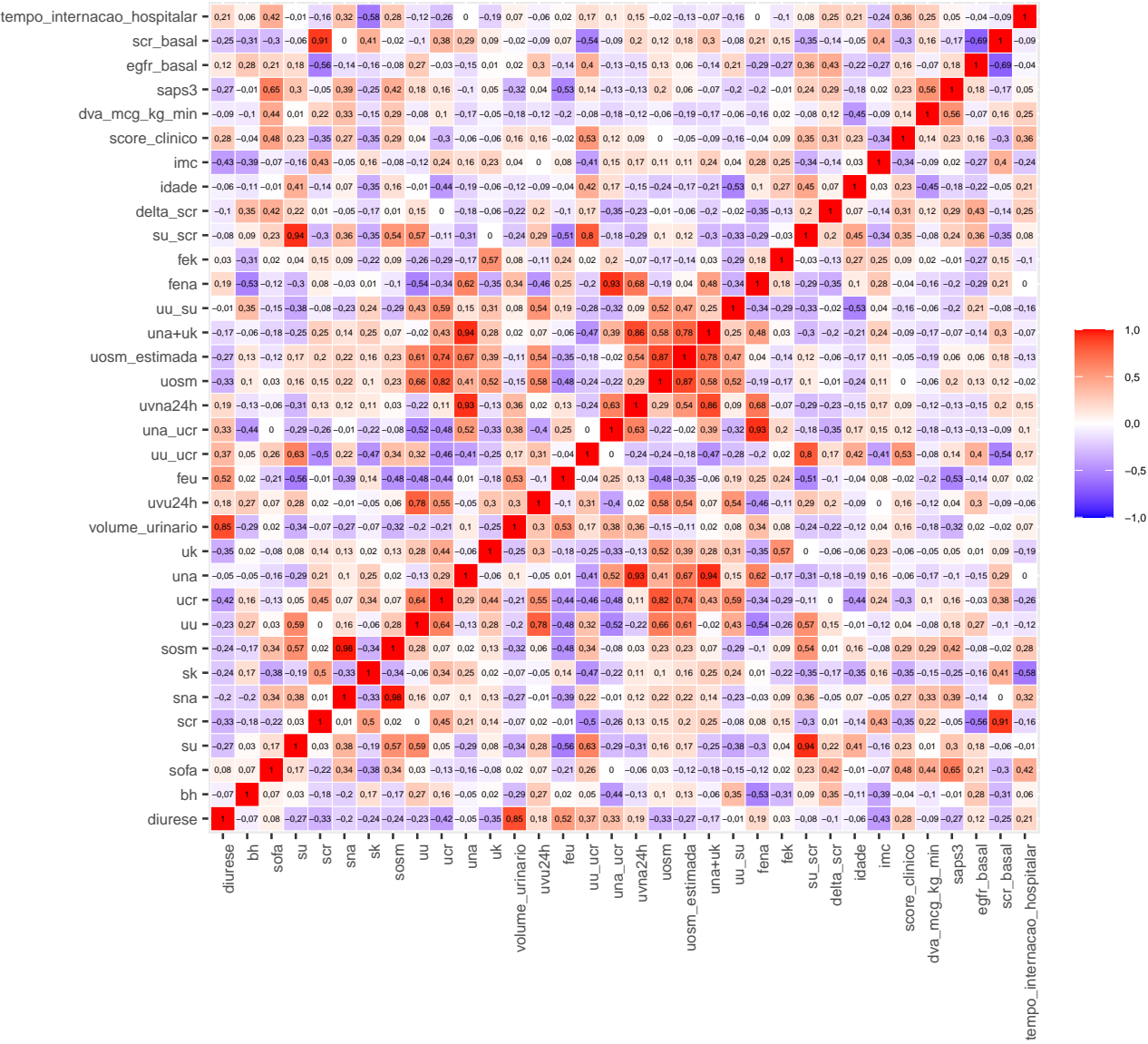
```

Pacientes que desenvolveram IRA



```
corr = df %>%  
  filter(ira == 0) %>%  
  select(all_of(numerical_columns)) %>%  
  drop_na %>%  
  cor %>%  
  as.matrix  
  
ggcorrplot(corr, hc.order = F, outline.col = "white", type = 'full',  
            tl.cex = 7, tl.srt = 90, lab = TRUE, lab_size = 1.5,  
            ggtheme = theme(legend.position="right",  
                             legend.title=element_blank(),  
                             legend.text=element_text(size=5))) +  
  labs(subtitle = "Pacientes que não desenvolveram IRA")
```


Pacientes que não desenvolveram IRA

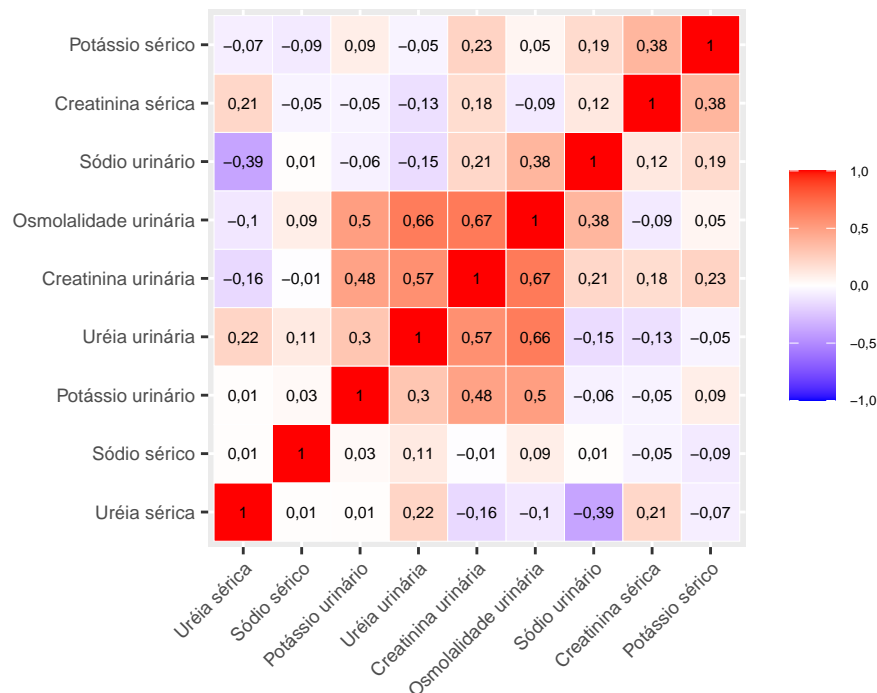


Variáveis Originais

```
corr = df %>%
  select(all_of(original_columns)) %>%
  drop_na %>%
  cor %>%
  rename_matrix

ggcorrplot(corr, hc.order = TRUE, outline.col = "white",
  tl.cex = 7, lab = TRUE, lab_size = 2,
  title = 'Matriz de correlação (Variáveis originais)',
  ggtheme = theme(legend.position="right",
    legend.title=element_blank(),
    legend.text=element_text(size=5)))
```

Matriz de correlação (Variáveis originais)

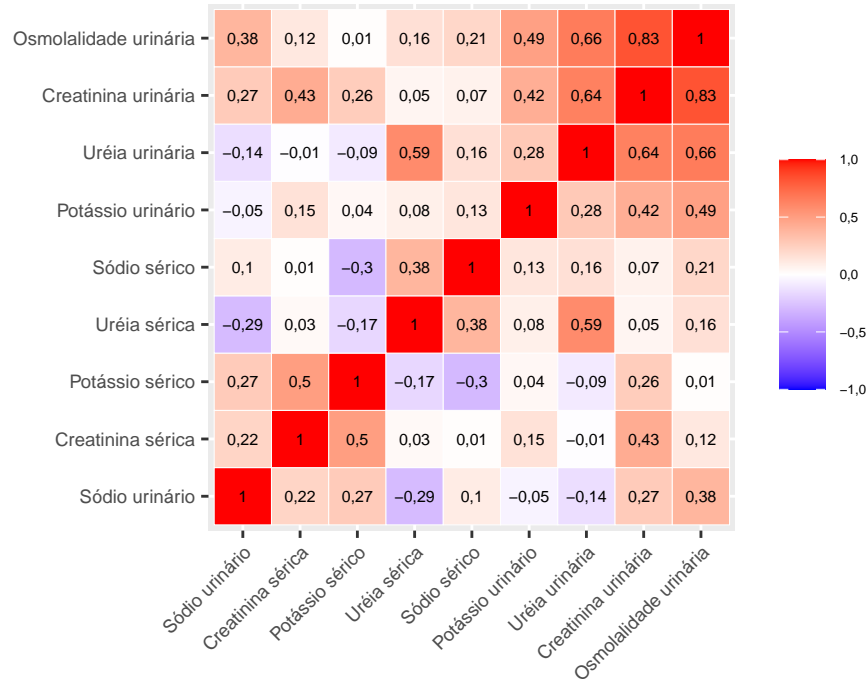


```
corr = df %>%
  filter(ira == 0) %>%
  select(all_of(original_columns)) %>%
  drop_na %>%
  cor %>%
  as.matrix %>%
  rename_matrix

ggcorrplot(corr, hc.order = TRUE, outline.col = "white",
  tl.cex = 7, lab = TRUE, lab_size = 2,
  title = 'Matriz de correlação para os pacientes que não desenvolveram IRA \n (Variáveis originais)',
  ggtheme = theme(plot.title = element_text(size=8),
    legend.position="right",
    legend.title=element_blank(),
    legend.text=element_text(size=5)))
```

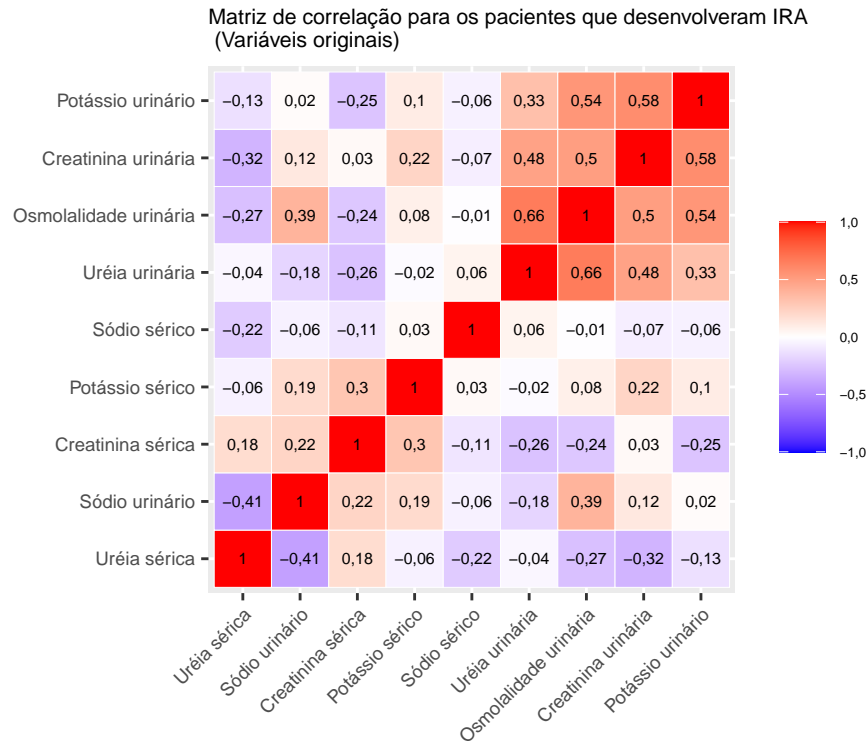
```
legend.text=element_text(size=5))
```

Matriz de correlação para os pacientes que não desenvolveram IRA
(Variáveis originais)



```
corr = df %>%
  filter(ira == 1) %>%
  select(all_of(original_columns)) %>%
  drop_na %>%
  cor %>%
  as.matrix %>%
  rename_matrix

ggcorrplot(corr, hc.order = TRUE, outline.col = "white",
  tl.cex = 7, lab = TRUE, lab_size = 2,
  title = 'Matriz de correlação para os pacientes que desenvolveram IRA \n (Variáveis originais)',
  ggtheme = theme(plot.title = element_text(size=8),
    legend.position="right",
    legend.title=element_blank(),
    legend.text=element_text(size=5)))
```

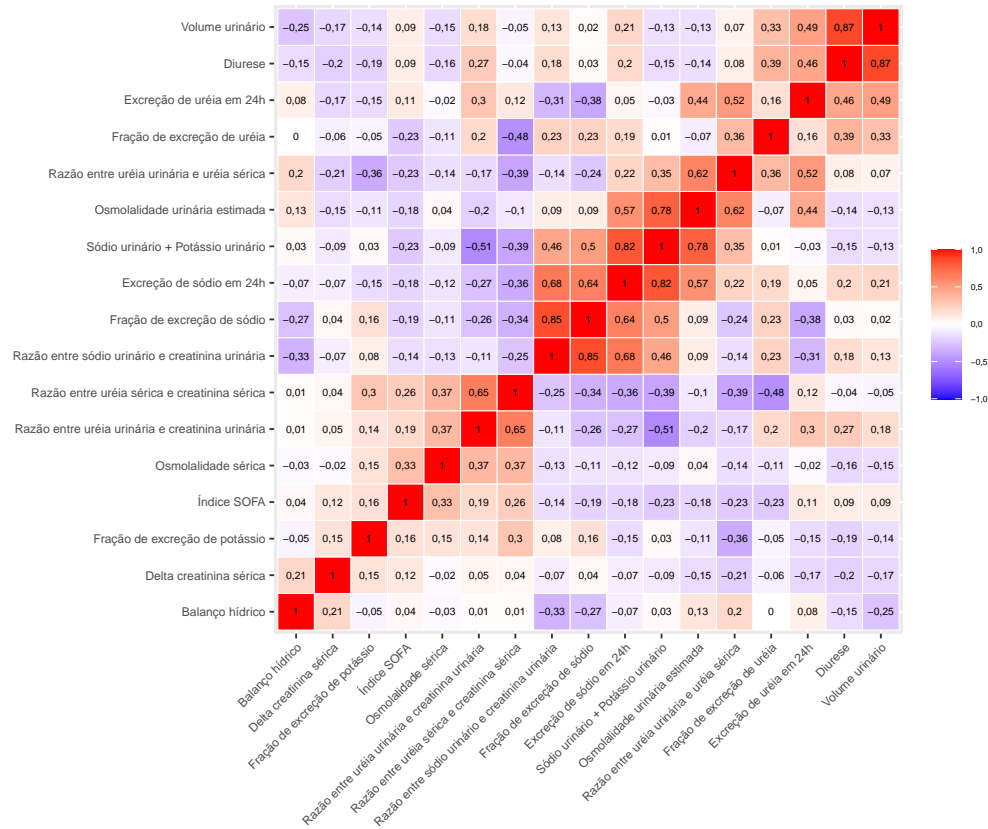


Variáveis derivadas

```
corr = df %>%
  select(all_of(derived_columns)) %>%
  drop_na %>%
  cor %>%
  as.matrix %>%
  rename_matrix

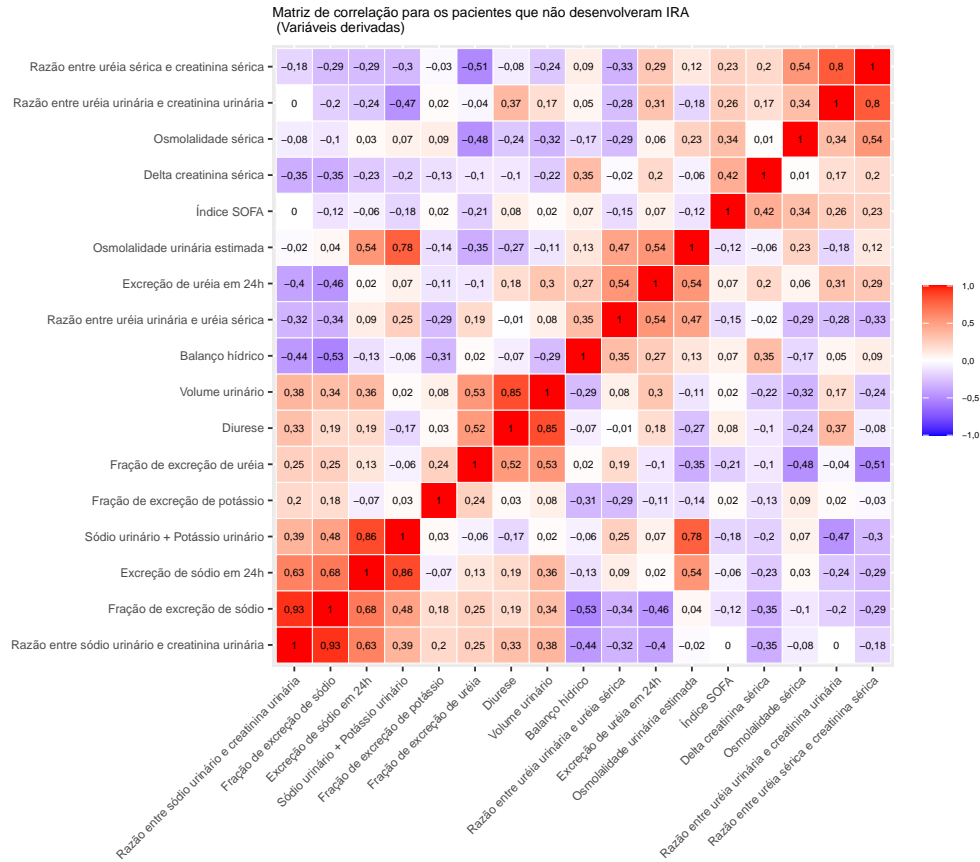
ggcorrplot(corr, hc.order = TRUE, outline.col = "white",
  tl.cex = 7, lab = TRUE, lab_size = 2,
  title = 'Matriz de correlação (Variáveis derivadas)',
  ggtheme = theme(legend.position="right",
    legend.title=element_blank(),
    legend.text=element_text(size=5)))
```

Matriz de correlação (Variáveis derivadas)



```
corr = df %>%
  filter(ira == 0) %>%
  select(all_of(derived_columns)) %>%
  drop_na %>%
  cor %>%
  as.matrix %>%
  rename_matrix

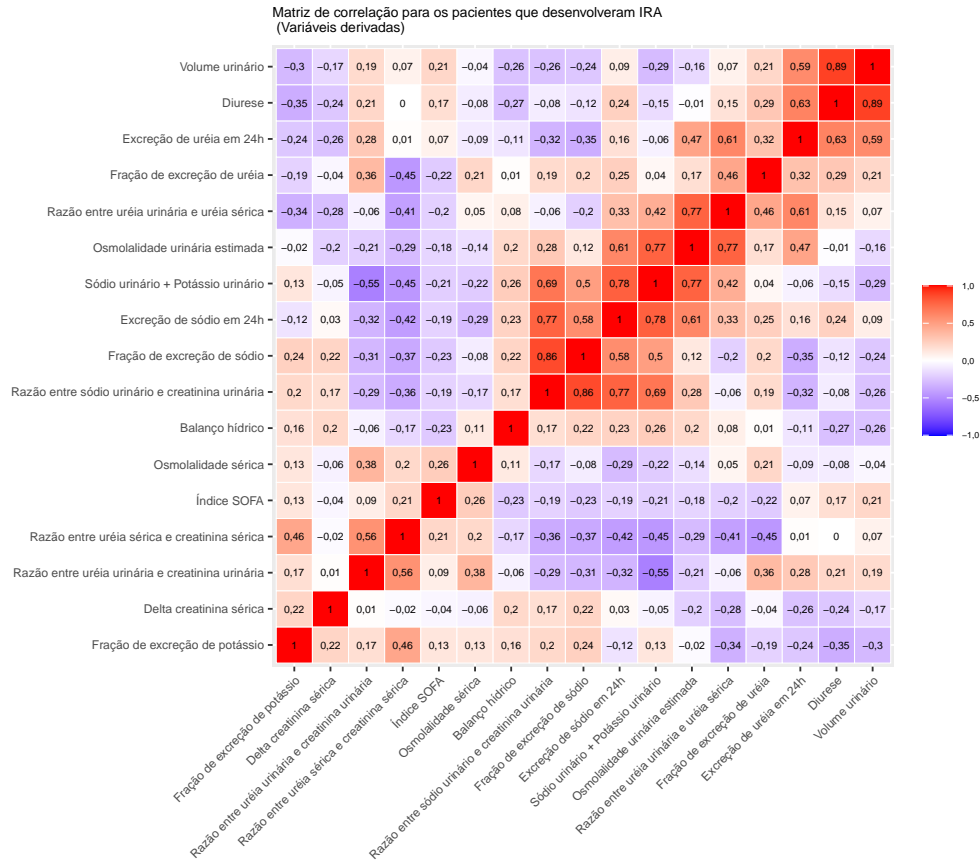
ggcorrplot(corr, hc.order = TRUE, outline.col = "white",
  tl.cex = 7, lab = TRUE, lab_size = 2,
  title = 'Matriz de correlação para os pacientes que não desenvolveram IRA \n (Variáveis derivadas)',
  ggtheme = theme(plot.title = element_text(size=8),
    legend.position="right",
    legend.title=element_blank(),
    legend.text=element_text(size=5)))
```



```
corr = df %>%
  filter(ira == 1) %>%
  select(all_of(derived_columns)) %>%
  drop_na %>%
  cor %>%
  as.matrix %>%
  rename_matrix

p.mat = df %>%
  filter(ira == 1) %>%
  select(all_of(derived_columns)) %>%
  drop_na %>%
  cor_pmat %>%
  as.matrix

ggcorrplot(corr, hc.order = TRUE, outline.col = "white",
  tl.cex = 7, lab = TRUE, lab_size = 2,
  title = 'Matriz de correlação para os pacientes que desenvolveram IRA \n (Variáveis derivadas)',
  ggtheme = theme(plot.title = element_text(size=8),
    legend.position="right",
    legend.title=element_blank(),
    legend.text=element_text(size=5)))
```



Dendrograma

```
d0 = df %>%
  filter(ira == 0) %>%
  select(all_of(original_columns)) %>%
  drop_na %>%
  cor %>%
  abs %>%
  rename_matrix %>%
  dist %>%
  hclust(method = 'complete') %>%
  as.dendrogram

d1 = df %>%
  filter(ira == 1) %>%
  select(all_of(original_columns)) %>%
  drop_na %>%
  cor %>%
  abs %>%
  rename_matrix %>%
  dist %>%
  hclust(method = 'ward.D2') %>%
  as.dendrogram
```

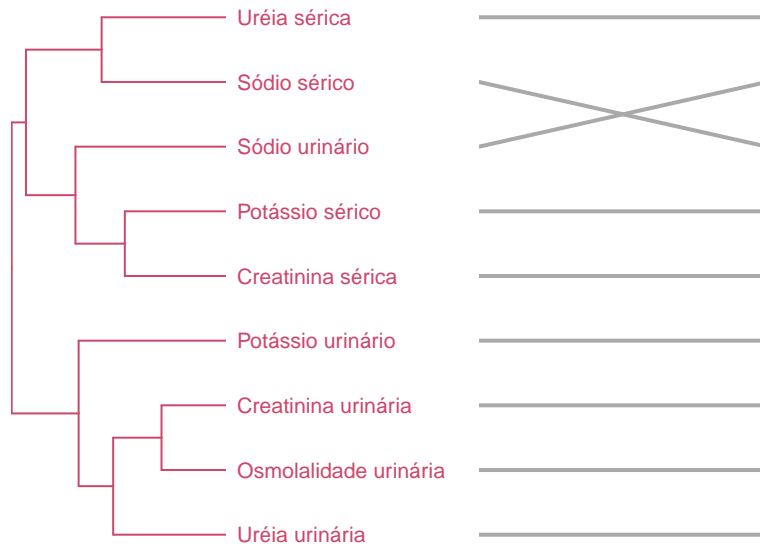
```

d1 <- dendlist(
  d0 %>%
    set("labels_col",
        value = c("skyblue", "orange", "grey"), k=3) %>%
    set("branches_lty", 1) %>%
    set("branches_k_color",
        value = c("skyblue", "orange", "grey"), k = 3),
  d1 %>%
    set("labels_col",
        value = c("skyblue", "orange", "grey"), k=3) %>%
    set("branches_lty", 1) %>%
    set("branches_k_color",
        value = c("skyblue", "orange", "grey"), k = 3)
)

d1 %>%
  untangle(method = "step2") %>%
  tanglegram(common_subtrees_color_lines = FALSE,
    highlight_distinct_edges = TRUE,
    highlight_branches_lwd = FALSE,
    margin_inner = 10,
    main_left = 'Pacientes sem IRA',
    main_right = 'Pacientes com IRA',
    lwd = 2,
    k_branches = 1,
    k_labels = 1,
    cex_main = 1.2,
    columns_width = c(5, 3, 5),
    margin_outer = 0.5)

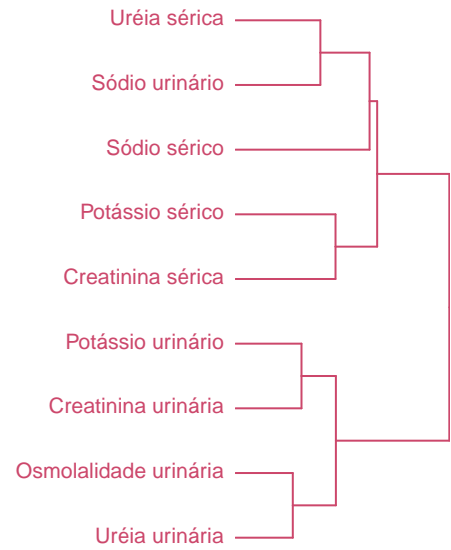
```


Pacientes sem IRA



1,5 1,0 0,5 0,0

Pacientes com IRA



0,0 1,0 2,0

Testes de hipótese

```
df_wilcox = tibble()

for (variable in numerical_columns){
  x = filter(df, ira == 0)[[variable]]
  y = filter(df, ira == 1)[[variable]]

  test = wilcox.test(x, y, alternative = "two.sided", exact = FALSE)

  df_wilcox = bind_rows(df_wilcox,
                        list("variavel" = variable, "Estatística" = test$statistic, "p-valor" = test$p.value))
}

df_wilcox = df_wilcox %>%
  arrange('p-valor') %>%
  mutate('Estatística' = round('Estatística', 3)) %>%
  inner_join(df_names, by='variavel') %>%
  rename('Variável' = nome)

significant_numerical_columns = df_wilcox %>%
  filter('p-valor' <= 0.25) %>%
  select(variavel) %>%
  pull

df_wilcox %>%
  select('Variável', 'Estatística', 'p-valor') %>%
  mutate('p-valor' = case_when('p-valor' == 1 ~ sprintf('> 0%s999', getOption("OutDec")),
                              'p-valor' < 0.001 ~ sprintf('< 0%s001', getOption("OutDec")),
                              TRUE ~ as.character(round('p-valor', 3)))) %>%
  niceFormatting(caption = "Teste Mann-Whitney")
```

Table 1: Teste Mann-Whitney

Variável	Estatística	p-valor
Índice SOFA	140,5	< 0,001
Uréia sérica	176,5	0,004
Creatinina sérica	211,0	0,025
SAPS3	212,5	0,026
Razão entre sódio urinário e creatinina urinária	448,0	0,036
Fração de excreção de potássio	230,0	0,058
Razão entre uréia urinária e uréia sérica	436,0	0,06
Sódio urinário	433,0	0,068
Creatinina sérica basal	244,0	0,101
Osmolalidade sérica	246,0	0,109
Excreção de sódio em 24h	421,0	0,109
Fração de excreção de sódio	392,0	0,188
Delta creatinina sérica	264,0	0,203
Razão entre uréia sérica e creatinina sérica	264,0	0,204
Tempo de internação hospitalar	265,5	0,213
Escore Clínico	272,5	0,219
Dose de noradrenalina	273,0	0,224

Table 1: Teste Mann-Whitney (*continued*)

Variável	Estatística	p-valor
Potássio urinário	270,0	0,246
Fração de excreção de uréia	394,0	0,269
Sódio sérico	276,5	0,297
Balanço hídrico	282,0	0,347
Sódio urinário + Potássio urinário	383,0	0,367
Diurese	377,0	0,428
Osmolalidade urinária estimada	372,0	0,484
Volume urinário	360,0	0,631
IMC	313,5	0,719
Creatinina urinária	351,0	0,754
Osmolalidade urinária	349,0	0,782
Razão entre uréia urinária e creatinina urinária	320,0	0,811
Excreção de uréia em 24h	328,0	0,927
Potássio sérico	329,5	0,948
Ritmo de filtração glomerular	337,0	0,956
Idade	330,5	0,963
Uréia urinária	332,0	0,985

```

df_chisq = tibble()
selected_categorical_columns = categorical_columns[!categorical_columns %in%
                                                    c('d_ira', 'kdigo')]

for (variable in selected_categorical_columns){
  if (length(unique(df[[variable]])) > 1){
    test = chisq.test(df$ira, df[[variable]],
                     simulate.p.value = TRUE)

    df_chisq = bind_rows(df_chisq,
                        list("variavel" = variable,
                            "Estatística" = test$statistic,
                            "p-valor" = test$p.value))
  }
}

df_chisq %>%
  arrange('p-valor') %>%
  mutate('p-valor' = case_when('p-valor' == 1 ~ sprintf('> 0%s999', getOption("OutDec")),
                              'p-valor' < 0.001 ~ sprintf('< 0%s001', getOption("OutDec")),
                              TRUE ~ as.character(round('p-valor', 3))),
        'Estatística' = round('Estatística', 3)) %>%
  inner_join(df_names, by='variavel') %>%
  rename('Variável' = nome) %>%
  select('Variável', 'Estatística', 'p-valor') %>%
  niceFormatting(caption = "Teste Chi-quadrado")

```

Table 2: Teste Chi-quadrado

Variável	Estatística	p-valor
Uso de ventilação mecânica	3,02	0,134
Asma ou DPOC	1,65	0,492
Câncer ativo	0,65	0,624
Cor	0,92	0,691
Diabetes mellitus	0,17	0,72
Hipertensão arterial sistêmica	0,12	0,794
Fator de risco	0,56	0,828
Insuficiência cardíaca congestiva	0,06	> 0,999
Acidente vascular cerebral	0,09	> 0,999
Doença vascular periférica	0,15	> 0,999
Uso de diuréticos	0,15	> 0,999
Uso de vasopressina	0,03	> 0,999
Uso de iECA ou espironolactona	0,02	> 0,999
Pós operatório	0,01	> 0,999

Removendo variáveis altamente correlacionadas

```
length(significant_numerical_columns)
```

```
## [1] 18
```

```

drop = c('uvna24h', 'una_uvr', 'fena', 'su_scr', 'saps3')

selected_numerical_columns = setdiff(significant_numerical_columns, drop)

length(selected_numerical_columns)

## [1] 13

dput(selected_numerical_columns)

## c("sofa", "su", "scr", "fek", "uu_su", "una", "scr_basal", "sosl",
## "delta_scr", "tempo_internacao_hospitalar", "score_clinico",
## "dva_mcg_kg_min", "uk")

```

Salvando variáveis selecionadas

```

write_yaml(list('selected_numerical_columns' = selected_numerical_columns,
               'selected_categorical_columns' = c('causa_ira', 'ventilacao_mecanica')),
           'selected_columns.yaml')

```