

# Correlations - death\_30days

Eduardo Yuki Yada

## Imports

```
library(tidyverse)
library(yaml)
library(kableExtra)
library(ggcorrplot)
```

## Loading data

```
load('dataset/processed_data.RData')
load('dataset/processed_dictionary.RData')

columns_list <- yaml.load_file("./auxiliar/columns_list.yaml")

outcome_column <- params$outcome_column
threshold <- params$threshold
print(threshold)

## [1] 0.1

df[columns_list$outcome_columns] <- lapply(df[columns_list$outcome_columns], as.character)
df[columns_list$outcome_columns] <- lapply(df[columns_list$outcome_columns], as.integer)
```

## Functions

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

## Correlation

```
na_eligible_columns <- df %>%
  summarise(across(everything(), ~ mean(is.na(.)))) %>%
  select_if(function(.) last(.) < 0.8) %>%
  names

unique_eligible_columns <- df %>%
  summarise(across(everything(), ~ length(unique(.)))) %>%
  select_if(function(.) last(.) > 1) %>%
  names

pre_columns = df_names %>%
  filter(momento.aquisicao == 'Admissão t0') %>%
  .$variable.name
```

```

weird_columns <- c('dieta_parenteral', 'dieta_enteral')

eligible_columns <- intersect(na_eligible_columns,
                             unique_eligible_columns) %>%
  intersect(pre_columns)

eligible_columns <- setdiff(eligible_columns, weird_columns)

corr <- df %>%
  select(all_of(intersect(columns_list$numerical_columns,
                          eligible_columns))) %>%

  drop_na %>%
  cor %>%
  as.matrix

corr_table <- corr %>%
  as.data.frame %>%
  tibble::rownames_to_column(var = 'row') %>%
  tidyr::pivot_longer(-row, names_to = 'column', values_to = 'correlation') %>%
  filter(row < column)

rename_column <- function(df, column_name){
  variable.name <- 'variable.name'
  df <- df %>%
    left_join(df_names %>% select(variable.name, abbrev.field.label),
              by = setNames(variable.name, column_name)) %>%
    select(-all_of(column_name)) %>%
    rename(!sym(column_name) := abbrev.field.label) %>%
    relocate(!sym(column_name))
}

corr_table %>%
  filter(correlation > 0.9) %>%
  rename_column('row') %>%
  rename_column('column') %>%
  select(row, column, correlation) %>%
  niceFormatting(caption = "Pearson Correlation", font_size = 9)

```

Table 1: Pearson Correlation

row	column	correlation
Idade no momento do primeiro procedimento	Idade no Procedimento 1	1.00
Núm. de hospitalizações pré-procedimento	Número da Admissão T0	0.98
Ano da admissão T0	Ano do procedimento 1	1.00
Antibióticos	Quantidade de antimicrobianos	1.00
Quantidade de procedimentos invasivos	Suporte cardiocirculatório	0.97
ECG	Quantidade de exames por métodos gráficos	1.00
Quantidade de exames de análises clínicas	Exames laboratoriais	1.00
Quantidade de exames de análises clínicas	Quantidade de exames diagnóstico por imagem	0.93
Biopsias	Quantidade de exames histopatológicos	0.93
Quantidade de exames diagnóstico por imagem	Exames laboratoriais	0.93
Quantidade de exames diagnóstico por imagem	Radiografias	0.98
Quantidade de classes medicamentosas de ação cardiovascular	Quantidade de classes medicamentosas utilizadas	0.91

## Hypothesis Tests

```

df_wilcox <- tibble()

for (variable in intersect(columns_list$numerical_columns,

```

```

eligible_columns)){
  if (mean(is.na(df[[variable]])) > 0.95) next

  x <- filter(df, !!sym(outcome_column) == 0)[[variable]]
  y <- filter(df, !!sym(outcome_column) == 1)[[variable]]

  test = tryCatch(wilcox.test(x, y, alternative = "two.sided", exact = FALSE),
    error=function(cond) {
      message("Can't calculate Wilcox test for variable ", variable)
      message(cond)
      return(list(statistic = NaN, p.value = NaN))
    })

  df_wilcox = bind_rows(df_wilcox,
    list("Variable" = variable,
      "Statistic" = test$statistic,
      "p-value" = test$p.value))
}

significant_num_cols <- df_wilcox %>%
  filter(`p-value` <= threshold) %>%
  select(Variable) %>%
  pull

df_wilcox <- df_wilcox %>%
  arrange(`p-value`) %>%
  mutate(`Statistic` = round(`Statistic`, 3)) %>%
  rename_column('Variable')

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

```

Table 2: Mann-Whitney Test

Variable	Statistic	p-value
Culturas	321249.5	< 0.001
Equipe Multiprofissional	265327.5	< 0.001
Ultrassom	345367.5	< 0.001
Ecocardiograma	290072.0	< 0.001
Angiografia	415705.0	< 0.001
Insulina	336262.0	< 0.001
Citologias	410289.0	< 0.001
UTI durante a admissão T0	427027.0	< 0.001
Quantidade de exames diagnóstico por imagem	280369.5	< 0.001
Ventilação não invasiva	415840.5	< 0.001
Tomografia	355076.0	< 0.001
Radiografias	295628.5	< 0.001
Quantidade de classes medicamentosas utilizadas	178689.5	< 0.001
Holter	362151.0	< 0.001
Psicofármacos	288508.5	< 0.001
Quantidade de exames por métodos gráficos	301962.5	< 0.001
ECG	303459.5	< 0.001
Núm. de hospitalizações pré-procedimento	432442.0	< 0.001
Exames laboratoriais	308627.0	< 0.001
Quantidade de exames de análises clínicas	308660.5	< 0.001

Table 2: Mann-Whitney Test (*continued*)

Variable	Statistic	p-value
Diuretico	298420.0	< 0.001
Cateter venoso central	398355.5	< 0.001
Número da Admissão T0	451594.0	< 0.001
Número de comorbidades	421317.5	< 0.001
DVA	325513.5	< 0.001
Quantidade de classes medicamentosas de ação cardiovascular	151258.5	< 0.001
Quantidade de exames histopatológicos	412144.5	0.001
Transfusão de hemoderivados	412116.0	0.001
Idade no momento do primeiro procedimento	435655.5	0.001
Idade no Procedimento 1	435655.5	0.001
Interconsulta médica	375417.5	0.002
Intervenção cardiovascular em laboratório de hemodinâmica	416859.0	0.002
Vasodilator	329469.0	0.003
Quantidade de procedimentos invasivos	370579.0	0.004
Antiplaquetario EV	386947.5	0.005
Quantidade de medicamentos de ação cardiovascular	318341.0	0.005
Bomba de infusão contínua	370640.0	0.006
Antifúngicos	379443.5	0.007
Antagonista da Aldosterona	346584.0	0.012
Antiarrítmicos	351217.5	0.014
Ressonancia magnetica	396585.0	0.014
Ano do procedimento 1	465138.5	0.016
Ano da admissão T0	465636.0	0.017
Cintilografia	404265.5	0.017
Antihipertensivo	376315.0	0.034
Quantidade de antimicrobianos	339526.5	0.038
Antibióticos	341516.5	0.045
Flebografia	415644.0	0.073
Angio TC	413111.5	0.081
Betabloqueador	371126.5	0.086
Antiviral	392356.5	0.105
Anticoagulantes orais	380577.5	0.132
Cateterismo	404956.5	0.148
Diárias no serviço de Emergência na admissão T0	224900.5	0.15
PET-CT	423343.5	0.205
Cirurgia Cardiovascular	416183.5	0.225
Número de procedimentos na admissão T0	548716.0	0.263
Outros procedimentos cirúrgicos	414386.5	0.281
Estatinas	372090.0	0.307
Eletrofisiologia	418278.0	0.341
Teste de esforço	432135.0	0.416
Cavografia	424431.0	0.438
Intervenção coronária percutânea	431475.0	0.455
Insuficiência cardíaca	382235.0	0.487
IECA/BRA	414977.5	0.513
Anticonvulsivante	391842.5	0.556
Biopsias	429858.0	0.58
Espirometria / Ergoespirometria	429759.0	0.589
Marca-passo temporário	385810.0	0.594
Hipoglicemiante	402267.5	0.61
Drenagem de tórax e punção pericárdica ou pleural	429495.0	0.617
Cardioversão/ Desfibrilação	389728.0	0.633
Angio RM	428967.0	0.682

Table 2: Mann-Whitney Test (continued)

Variable	Statistic	p-value
Tilt Test	428901.0	0.691
Suporte cardiocirculatório	428868.0	0.696
Exames endoscópicos	425842.0	0.703
Cirurgia Toracica	428736.0	0.716
Aortografia	428670.0	0.727
Transplante cardíaco	428571.0	0.744
Díalise durante a admissão T0	557989.0	0.747
Angioplastia	428439.0	0.769
Polissonografia	428406.0	0.775
Instalação de CEC	426346.0	0.786
Traqueostomia	428307.0	0.797
Digoxina	399852.0	0.803
Trombolítico	397345.0	0.837
Antiretroviral	397345.0	0.837
Arteriografia	428109.0	0.851
Bloqueador do canal de calcio	397738.0	0.919
Antiplaquetario VO	397085.0	NaN
Hormonio tireoidiano	397085.0	NaN
Broncodilator	397085.0	NaN
Stent	427878.0	NaN

```
df_chisq <- tibble()

for (variable in intersect(columns_list$categories,
                           eligible_columns)){
  if (length(unique(df[[variable]])) > 1){
    test <- tryCatch(chisq.test(df[[outcome_column]],
                                df[[variable]] %>% replace_na('NA'), # counting NA as cat
                                simulate.p.value = TRUE),
                    error = function (cond) {
                      message("Can't calculate Chi Squared test for variable ", variable)
                      message(cond)
                      return(list(statistic = NaN, p.value = NaN))
                    })

    df_chisq <- bind_rows(df_chisq,
                        list("Variable" = variable,
                            "Statistic" = test$statistic,
                            "p-value" = test$p.value))
  }
}

significant_cat_cols <- df_chisq %>%
  filter(`p-value` <= threshold) %>%
  select(Variable) %>%
  pull

df_chisq <- df_chisq %>%
  arrange(`p-value`) %>%
  mutate(`Statistic` = round(`Statistic`, 3)) %>%
  rename_column('Variable')

df_chisq %>%
  mutate(`p-value` = case_when(`p-value` == 1 ~ sprintf('> 0%s999', getOption("OutDec")),
                                `p-value` < 0.001 ~ sprintf('< 0%s001', getOption("OutDec")),
```

```
TRUE ~ as.character(round(`p-value`, 3))) %>%
niceFormatting(caption = "Chi-squared test")
```

Table 3: Chi-squared test

Variable	Statistic	p-value
Escolaridade	26.26	< 0.001
Insuficiência renal crônica	31.75	< 0.001
Admissão em até 180 dias antes da T0	18.85	< 0.001
Diabetes mellitus	12.31	0.001
Tipo de Procedimento 1	9.34	0.001
Hipertensão arterial	8.87	0.002
Tipo de Dispositivo ao final do procedimento 1	12.86	0.002
Doença cardíaca	14.49	0.003
Tipo de Dispositivo ao final do procedimento 1	15.60	0.003
Insuficiência cardíaca	9.77	0.004
Valvopatias/ Prótese valvares	8.87	0.007
Infarto do miocárdio prévio / Doença arterial coronariana	7.19	0.013
Tipo de Reoperação 1	10.72	0.019
Tipo de Procedimento 1	10.72	0.021
Doença cardíaca	19.77	0.035
Fibrilação / flutter atrial	4.33	0.038
Classe funcional de IC	20.39	0.049
Sexo	3.23	0.065
Hemodiálise	9.83	0.079
Neoplasia em tratamento ou tratada recentemente	4.42	0.09
Estado de residência	49.98	0.151
Acidente Vascular Cerebral/ Acidente isquêmico transitório prévios	0.26	0.746
Parada cardíaca prévia/ Taquicardia ventricular instável	0.04	0.86
Raça	0.98	0.97
Transplante cardíaco prévio	0.05	> 0.999
Endocardite prévia	0.27	> 0.999
Doença pulmonar obstrutiva crônica	0.00	> 0.999

```
dir.create(file.path("./auxiliar/significant_columns/"), showWarnings = FALSE)

saveRDS(significant_cat_cols,
        file = sprintf("./auxiliar/significant_columns/categorical_%s.rds", outcome_column))

saveRDS(significant_num_cols,
        file = sprintf("./auxiliar/significant_columns/numerical_%s.rds", outcome_column))

## [1] 78
## [1] 20
## [1] 144
## [1] 50
```