

Correlations - readmission_1year

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
```

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_parentral', '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
EKG	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
Número da Admissão T0	10904164	< 0.001
Quantidade de classes medicamentosas utilizadas	4385363	< 0.001
Quantidade de medicamentos de ação cardiovascular	6141405	< 0.001
Quantidade de exames diagnóstico por imagem	7101047	< 0.001
Quantidade de classes medicamentosas de ação cardiovascular	3731433	< 0.001
DVA	6872166	< 0.001
Antiarrítmicos	7083153	< 0.001
Quantidade de exames por métodos gráficos	7246848	< 0.001
Núm. de hospitalizações pré-procedimento	11412867	< 0.001
Equipe Multiprofissional	7512409	< 0.001
UTI durante a admissão T0	11765170	< 0.001
ECG	7322103	< 0.001
Diurético	6665950	< 0.001
Exames laboratoriais	7394740	< 0.001
Quantidade de exames de análises clínicas	7395616	< 0.001
Antagonista da Aldosterona	7085124	< 0.001
Radiografias	7434065	< 0.001
Insuficiência cardíaca	7007152	< 0.001
Ultrassom	8584189	< 0.001
Ecocardiograma	7949768	< 0.001
Holter	8676786	< 0.001
Quantidade de procedimentos invasivos	8440705	< 0.001
Biopsias	9359373	< 0.001
Culturas	8606973	< 0.001

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

Variable	Statistic	p-value
Quantidade de exames histopatológicos	9292689	< 0.001
Número de comorbidades	11866387	< 0.001
Transplante cardíaco	9444051	< 0.001
Cateterismo	8759359	< 0.001
Psicofármacos	7246017	< 0.001
Tomografia	8805922	< 0.001
Ressonancia magnetica	8934547	< 0.001
Antifúngicos	8187988	< 0.001
Anticoagulantes orais	7987603	< 0.001
Vasodilator	7470164	< 0.001
Cateter venoso central	9194171	< 0.001
Quantidade de antimicrobianos	7274314	< 0.001
Antibióticos	7292261	< 0.001
Exames endoscópicos	9318312	< 0.001
Cintilografia	9126638	< 0.001
Antiviral	8363226	< 0.001
Estatinas	7493319	< 0.001
Digoxina	8066716	< 0.001
Bloqueador do canal de calcio	8243220	< 0.001
Eletrofisiologia	9183094	< 0.001
Bomba de infusão contínua	8133843	< 0.001
IECA/BRA	7572438	< 0.001
Diárias no serviço de Emergência na admissão T0	5101688	< 0.001
Betabloqueador	8053232	< 0.001
Suporte cardiocirculatório	9477646	< 0.001
Outros procedimentos cirúrgicos	9193207	< 0.001
Citologias	9449908	< 0.001
Instalação de CEC	9390590	< 0.001
Transfusão de hemoderivados	9419579	< 0.001
Intervenção coronária percutânea	9433229	< 0.001
Antiplaquetario EV	8399106	< 0.001
Diálise durante a admissão T0	13772958	< 0.001
Idade no momento do primeiro procedimento	14719971	< 0.001
Idade no Procedimento 1	14719971	< 0.001
Insulina	8242196	< 0.001
Espirometria / Ergoespirometria	9480879	< 0.001
Angio TC	9382792	< 0.001
Angio RM	9499750	< 0.001
Anticonvulsivante	8319368	< 0.001
Cardioversão/ Desfibrilação	8311078	< 0.001
Arteriografia	9530009	< 0.001
Angioplastia	9518881	< 0.001
Intervenção cardiovascular em laboratório de hemodinâmica	9497367	0.002
Interconsulta médica	9352080	0.013
PET-CT	9509441	0.016
Flebografia	9476403	0.022
Ano da admissão T0	14196849	0.031
Tilt Test	9524146	0.032
Cirurgia Toracica	9526438	0.033
Ano do procedimento 1	14234661	0.037
Antiretroviral	8470643	0.038
Teste de esforço	9502297	0.053
Marca-passo temporário	8321002	0.063

Table 2: Mann-Whitney Test (continued)

Variable	Statistic	p-value
Aortografia	9531281	0.077
Ventilação não invasiva	9568916	0.117
Número de procedimentos na admissão T0	13777933	0.12
Polissonografia	9537582	0.149
Traqueostomia	9541571	0.274
Trombolítico	8476784	0.308
Antihipertensivo	8441704	0.364
Drenagem de tórax e punção pericárdica ou pleural	9539240	0.474
Hipoglicemiante	8464050	0.691
Cirurgia Cardiovascular	9565657	0.734
Cavografia	9543735	0.759
Angiografia	9552516	0.834
Antiplaquetário VO	8482721	NaN
Hormônio tireoidiano	8482721	NaN
Broncodilatador	8482721	NaN
Stent	9550184	NaN

```
df_chisq <- tibble()

for (variable in intersect(columns_list$categorical_columns,
                           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
Sexo	23.60	< 0.001
Escolaridade	50.27	< 0.001
Doença cardíaca	100.64	< 0.001
Doença cardíaca	48.81	< 0.001
Classe funcional de IC	68.67	< 0.001
Infarto do miocárdio prévio / Doença arterial coronariana	32.48	< 0.001
Insuficiência cardíaca	205.82	< 0.001
Parada cardíaca prévia/ Taquicardia ventricular instável	18.41	< 0.001
Transplante cardíaco prévio	22.40	< 0.001
Valvopatias/ Prótese valvares	20.60	< 0.001
Diabetes melittus	10.81	< 0.001
Tipo de Procedimento 1	134.69	< 0.001
Tipo de Reoperação 1	160.91	< 0.001
Tipo de Procedimento 1	160.91	< 0.001
Tipo de Dispositivo ao final do procedimento 1	298.23	< 0.001
Tipo de Dispositivo ao final do procedimento 1	130.10	< 0.001
Admissão em até 180 dias antes da T0	261.65	< 0.001
Estado de residência	61.04	< 0.001
Fibrilação / flutter atrial	13.87	< 0.001
Insuficiência renal crônica	13.46	< 0.001
Raça	21.68	0.004
Hemodiálise	9.91	0.006
Endocardite prévia	4.39	0.04
Doença pulmonar obstrutiva crônica	3.32	0.082
Acidente Vascular Cerebral/ Acidente isquêmico transitório prévios	1.29	0.269
Neoplasia em tratamento ou tratada recentemente	1.03	0.324
Hipertensão arterial	0.32	0.586

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

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

```
## [1] 78
```

```
## [1] 24
```

```
## [1] 144
```

```
## [1] 78
```