

Correlations - readmission_180d

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

## Warning in cor(.): the standard deviation is zero

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
Quantidade de classes medicamentosas utilizadas	3361538	< 0.001
Número da Admissão T0	8553267	< 0.001
Quantidade de medicamentos de ação cardiovascular	4686948	< 0.001
Quantidade de exames diagnóstico por imagem	5415608	< 0.001
Antiarrítmicos	5394317	< 0.001
DVA	5269875	< 0.001
Quantidade de classes medicamentosas de ação cardiovascular	2885731	< 0.001
UTI durante a admissão T0	8959897	< 0.001
Quantidade de exames por métodos gráficos	5606223	< 0.001
Radiografias	5625305	< 0.001
Equipe Multiprofissional	5806147	< 0.001
ECG	5653599	< 0.001
Exames laboratoriais	5682400	< 0.001
Quantidade de exames de análises clínicas	5682959	< 0.001
Ultrassom	6589651	< 0.001
Antagonista da Aldosterona	5469613	< 0.001
Diuretico	5169253	< 0.001
Insuficiência cardíaca	5419983	< 0.001
Núm. de hospitalizações pré-procedimento	9002547	< 0.001
Ecocardiograma	6115587	< 0.001
Transplante cardíaco	7291228	< 0.001
Quantidade de exames histopatológicos	7159374	< 0.001
Biopsias	7224310	< 0.001

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

Variable	Statistic	p-value
Culturas	6550458	< 0.001
Quantidade de procedimentos invasivos	6478937	< 0.001
Número de comorbidades	9013798	< 0.001
Cateterismo	6746283	< 0.001
Antifúngicos	6312115	< 0.001
Antiviral	6454746	< 0.001
Tomografia	6783884	< 0.001
Anticoagulantes orais	6149381	< 0.001
Psicofármacos	5588764	< 0.001
Vasodilator	5697385	< 0.001
Holter	6805548	< 0.001
Exames endoscópicos	7193206	< 0.001
Cateter venoso central	7099224	< 0.001
Ressonancia magnetica	6919055	< 0.001
Quantidade de antimicrobianos	5575384	< 0.001
Antibióticos	5582354	< 0.001
Cintilografia	7039536	< 0.001
Estatinas	5804127	< 0.001
Bloqueador do canal de calcio	6374799	< 0.001
Digoxina	6235145	< 0.001
Betabloqueador	6144053	< 0.001
Citologias	7295480	< 0.001
IECA/BRA	5835916	< 0.001
Eletrofisiologia	7137533	< 0.001
Diárias no serviço de Emergência na admissão T0	3973019	< 0.001
Antiplaquetario EV	6486239	< 0.001
Outros procedimentos cirúrgicos	7094556	< 0.001
Suporte cardiocirculatório	7335938	< 0.001
Díalise durante a admissão T0	10567523	< 0.001
Bomba de infusão contínua	6323954	< 0.001
Instalação de CEC	7268646	< 0.001
Insulina	6320000	< 0.001
Transfusão de hemoderivados	7292104	< 0.001
Anticonvulsivante	6410692	< 0.001
Intervenção coronária percutânea	7307233	< 0.001
Idade no momento do primeiro procedimento	11352291	< 0.001
Idade no Procedimento 1	11352291	< 0.001
Angio RM	7357406	< 0.001
Angio TC	7285546	0.002
Cardioversão/ Desfibrilação	6431563	0.002
Interconsulta médica	7182340	0.003
Espirometria / Ergoespirometria	7352944	0.003
PET-CT	7354805	0.006
Antiretroviral	6556922	0.008
Tilt Test	7369188	0.015
Flebografia	7326198	0.015
Teste de esforço	7345952	0.023
Intervenção cardiovascular em laboratório de hemodinâmica	7362028	0.026
Angioplastia	7379942	0.052
Ventilação não invasiva	7414168	0.073
Número de procedimentos na admissão T0	10571323	0.08
Arteriografia	7386690	0.089
Marca-passo temporário	6434240	0.092

Table 2: Mann-Whitney Test (continued)

Variable	Statistic	p-value
Cirurgia Toracica	7379056	0.097
Trombolitico	6563063	0.141
Polissonografia	7385825	0.216
Ano da admissão T0	10783655	0.242
Aortografia	7384330	0.243
Ano do procedimento 1	10815364	0.265
Drenagem de tórax e punção pericárdica ou pleural	7406542	0.404
Antihipertensivo	6539813	0.438
Cavografia	7382056	0.473
Cirurgia Cardiovascular	7413221	0.655
Hipoglicemiante	6552765	0.666
Angiografia	7392110	0.743
Traqueostomia	7396977	0.812
Antiplaquetario VO	6570608	NaN
Hormonio tireoidiano	6570608	NaN
Broncodilator	6570608	NaN
Stent	7395328	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
Escolaridade	39.67	< 0.001
Doença cardíaca	65.58	< 0.001
Doença cardíaca	35.93	< 0.001
Classe funcional de IC	49.78	< 0.001
Infarto do miocárdio prévio / Doença arterial coronariana	29.71	< 0.001
Insuficiência cardíaca	148.15	< 0.001
Valvopatias/ Prótese valvares	15.80	< 0.001
Diabetes mellitus	13.59	< 0.001
Tipo de Procedimento 1	126.21	< 0.001
Tipo de Reoperação 1	145.07	< 0.001
Tipo de Procedimento 1	145.07	< 0.001
Tipo de Dispositivo ao final do procedimento 1	223.71	< 0.001
Tipo de Dispositivo ao final do procedimento 1	102.97	< 0.001
Admissão em até 180 dias antes da T0	180.10	< 0.001
Parada cardíaca prévia/ Taquicardia ventricular instável	19.14	< 0.001
Fibrilação / flutter atrial	15.59	0.001
Transplante cardíaco prévio	23.09	0.001
Sexo	10.54	0.002
Hemodiálise	10.90	0.005
Estado de residência	52.28	0.011
Insuficiência renal crônica	6.68	0.012
Endocardite prévia	4.91	0.034
Doença pulmonar obstrutiva crônica	3.46	0.075
Neoplasia em tratamento ou tratada recentemente	1.95	0.195
Raça	6.55	0.364
Acidente Vascular Cerebral/ Acidente isquêmico transitório prévios	0.84	0.399
Hipertensão arterial	0.26	0.643

```

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] 23
## [1] 144
## [1] 77

```