Correlations

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_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
```

```
## 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
Exames laboratoriais	Radiografias	0.90
Quantidade de exames de análises clínicas	Exames laboratoriais	1.00
Quantidade de exames de análises clínicas	Radiografias	0.90
Quantidade de exames de análises clínicas	Quantidade de exames diagnóstico por imagem	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	1564734	< 0.001
Quantidade de exames diagnóstico por imagem	2487195	< 0.001
Número da Admissão T0	3861792	< 0.001
Radiografias	2566380	< 0.001
Quantidade de exames por métodos gráficos	2569499	< 0.001
ECG	2573006	< 0.001
Quantidade de medicamentos de ação cardiovascular	2275320	< 0.001
Quantidade de classes medicamentosas de ação cardiovascular	1347955	< 0.001
UTI durante a admissão T0	3982081	< 0.001
Ecocardiograma	2713427	< 0.001
Equipe Multiprofissional	2688548	< 0.001
Antiarritmicos	2550803	< 0.001
Quantidade de exames de análises clínicas	2668665	< 0.001
Exames laboratoriais	2668809	< 0.001
DVA	2517982	< 0.001
Diuretico	2433199	< 0.001
Vasodilator	2502238	< 0.001
Culturas	2956507	< 0.001
Núm. de hospitalizações pré-procedimento	4042770	< 0.001
Anticoagulantes orais	2757078	< 0.001
Antagonista da Aldosterona	2594074	< 0.001
Ultrassom	3045992	< 0.001
Psicofármacos	2500656	< 0.001

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

Variable	Statistic	p-value
Quantidade de procedimentos invasivos	2963280	< 0.001
Quantidade de antimicrobianos	2529161	< 0.001
Antibióticos	2533901	< 0.001
Antifúngicos	2841474	< 0.001
Biopsias	3262107	< 0.001
Suporte cardiocirculatório	3275079	< 0.001
Cateterismo	3078357	< 0.001
Ressonancia magnetica	3128877	< 0.001
Betabloqueador	2741472	< 0.001
Insuficiência cardíaca	2657679	< 0.001
Cintilografia	3176540	< 0.001
Digoxina	2802468	< 0.001
Número de comorbidades	4134390	< 0.001
Cateter venoso central	3209625	< 0.001
Tomografia	3109248	< 0.001
Quantidade de exames histopatológicos	3249213	< 0.001
Bloqueador do canal de calcio	2876961	< 0.001
Bomba de infusão contínua	2814483	< 0.001
Holter	3132524	< 0.001
Diárias no serviço de Emergência na admissão T0	1761464	< 0.001
Exames endoscópicos	3250420	< 0.001
Antiviral	2923988	< 0.001
Estatinas	2715080	< 0.001
Transplante cardíaco	3297204	< 0.001
Eletrofisiologia	3224269	< 0.001
Angio RM	3293722	0.001
IECA/BRA	2727646	0.002
Anticonvulsivante	2879949	0.003
Diálise durante a admissão T0	4621489	0.004
Instalação de CEC	3275335	0.01
Citologias	3287293	0.01
Outros procedimentos cirúrgicos	3224604	0.01
Insulina	2870029	0.013
Angio TC	3260807	0.018
Marca-passo temporário	2865841	0.03
Intervenção coronária percutânea	3288416	0.032
PET-CT	3295907	0.037
Flebografia	3278211	0.04
Antiplaquetario EV	2935137	0.05
Transfusão de hemoderivados	3288930	0.058
Cardioversão/ Desfibrilação	2880534	0.092
Intervenção cardiovascular em laboratório de hemodinâmica	3301035	0.094
Tilt Test	3306179	0.099
Angioplastia	3309422	0.117
Teste de esforço	3298521	0.181
Ventilação não invasiva	3328148	0.206
Arteriografia	3313440	0.212
Antihipertensivo	2925032	0.266
Antiretroviral	2951161	0.268
Angiografia	3311505	0.323
Hipoglicemiante	2981659	0.341
Idade no momento do primeiro procedimento	4746091	0.386
Idade no Procedimento 1	4746091	0.386

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

Variable	Statistic	p-value
Polissonografia	3322678	0.391
Drenagem de tórax e punção pericárdica ou pleural	3325956	0.408
Trombolitico	2957904	0.502
Cirurgia Toracica	3313321	0.528
Cirurgia Cardiovascular	3334523	0.549
Cavografia	3310894	0.58
Ano da admissão T0	4689596	0.583
Traqueostomia	3315269	0.586
Número de procedimentos na admissão T0	4637873	0.613
Ano do procedimento 1	4699863	0.65
Stent	3318249	0.84
Espirometria / Ergoespirometria	3319581	0.87
Interconsulta médica	3316142	0.969
Aortografia	3318134	0.982
Antiplaquetario VO	2955198	NaN
Hormonio tireoidiano	2955198	NaN
Broncodiltador	2955198	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
Insuficiência cardíaca	38.70	< 0.001
Tipo de Procedimento 1	38.90	< 0.001
Tipo de Reoperação 1	51.06	< 0.001
Tipo de Procedimento 1	51.06	< 0.001
Tipo de Dispositivo ao final do procedimento 1	49.06	< 0.001
Tipo de Dispositivo ao final do procedimento 1	19.03	< 0.001
Admissão em até 180 dias antes da T0	49.31	< 0.001
Desfecho principal da admissão T0	10.25	0.003
Escolaridade	18.94	0.005
Doença cardíaca	11.67	0.01
Infarto do miocárdio prévio / Doença arterial coronariana	6.95	0.011
Endocardite prévia	4.66	0.035
Doença cardíaca	18.60	0.038
Hemodiálise	5.93	0.042
Classe funcional de IC	11.73	0.052
Transplante cardíaco prévio	4.85	0.079
Valvopatias/ Prótese valvares	2.49	0.118
Acidente Vascular Cerebral/ Acidente isquêmico transitório prévios	2.73	0.124
Fibrilação / flutter atrial	2.30	0.125
Insuficiência renal crônica	1.89	0.168
Parada cardíaca prévia/ Taquicardia ventricular instável	1.89	0.176
Neoplasia em tratamento ou tratada recentemente	1.73	0.208
Estado de residência	33.62	0.239
Diabetes melittus	1.25	0.281
Hipertensão arterial	0.58	0.465
Sexo	0.48	0.5
Raça	1.48	0.923
Doença pulmonar obstrutiva crônica	0.01	> 0.999
Óbito intraoperatório 1	0.27	> 0.999

```
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] 16
```

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

```
## [1] 66
```