Correlations - readmission_30d

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
Quantidade de classes medicamentosas utilizadas	1506044	< 0.001
Quantidade de exames diagnóstico por imagem	2388006	< 0.001
Número da Admissão T0	3783923	< 0.001
Radiografias	2469724	< 0.001
UTI durante a admissão T0	3860551	< 0.001
Quantidade de exames por métodos gráficos	2479542	< 0.001
ECG	2483026	< 0.001
Quantidade de medicamentos de ação cardiovascular	2191750	< 0.001
Ecocardiograma	2620736	< 0.001
Quantidade de classes medicamentosas de ação cardiovascular	1298232	< 0.001
Equipe Multiprofissional	2595063	< 0.001
DVA	2427006	< 0.001
Quantidade de exames de análises clínicas	2574255	< 0.001
Exames laboratoriais	2574410	< 0.001
Antiarritmicos	2484440	< 0.001
Culturas	2865748	< 0.001
Diuretico	2349605	< 0.001
Vasodilator	2435689	< 0.001
Antifúngicos	2769829	< 0.001
Ultrassom	2966386	< 0.001

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

Variable	Statistic	p-value
Núm. de hospitalizações pré-procedimento	3963198	< 0.001
Psicofármacos	2428243	< 0.001
Quantidade de procedimentos invasivos	2888024	< 0.001
Antagonista da Aldosterona	2534509	< 0.001
Anticoagulantes orais	2703904	< 0.001
Suporte cardiocirculatório	3214771	< 0.001
Quantidade de antimicrobianos	2446306	< 0.001
Antibióticos	2451300	< 0.001
Biopsias	3202942	< 0.001
Cateterismo	3014749	< 0.001
Quantidade de exames histopatológicos	3183951	< 0.001
Tomografia	3029978	< 0.001
Ressonancia magnetica	3073788	< 0.001
Insuficiência cardíaca	2598630	< 0.001
Cintilografia	3117169	< 0.001
Betabloqueador	2688898	< 0.001
Exames endoscópicos	3183561	< 0.001
Cateter venoso central	3147508	< 0.001
Digoxina	2749082	< 0.001
Número de comorbidades	4046318	< 0.001
Bomba de infusão contínua	2758370	< 0.001
Diálise durante a admissão T0	4539192	< 0.001
Bloqueador do canal de calcio	2822742	< 0.001
Holter	3071220	< 0.001
Diárias no serviço de Emergência na admissão T0	1725439	< 0.001
Antiviral	2868180	< 0.001
Citologias	3222300	< 0.001
Estatinas	2662153	< 0.001
Anticonvulsivante	2815530	< 0.001
Transplante cardíaco	3238522	< 0.001
Eletrofisiologia	3166334	< 0.001
Angio RM	3235154	0.001
Outros procedimentos cirúrgicos	3152152	0.002
IECA/BRA	2679101	0.003
Insulina	2804958	0.003
Instalação de CEC	3213756	0.004
Angio TC	3199243	0.01
Transfusão de hemoderivados	3224107	0.011
PET-CT	3235629	0.017
Intervenção coronária percutânea	3228734	0.022
Marca-passo temporário	2812741	0.028
Antiplaquetario EV	2879717	0.031
Flebografia	3218950	0.033
Cardioversão/ Desfibrilação	2826893	0.073
Intervenção cardiovascular em laboratório de hemodinâmica	3242466	0.094
Angioplastia	3250513	0.102
Tilt Test	3247644	0.106
Arteriografia	3254420	0.164
Teste de esforço	3240552	0.198
Antiretroviral	2896414	0.221
Antihipertensivo	2868965	0.231
Ventilação não invasiva	3266930	0.263
Hipoglicemiante	2928694	0.306

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

Variable	Statistic	p-value
Cirurgia Toracica	3252851	0.335
Angiografia	3252858	0.336
Idade no momento do primeiro procedimento	4668106	0.379
Idade no Procedimento 1	4668106	0.379
Polissonografia	3263284	0.414
Número de procedimentos na admissão T0	4555361	0.452
Drenagem de tórax e punção pericárdica ou pleural	3265370	0.483
Cavografia	3250762	0.496
Traqueostomia	3255987	0.497
Trombolitico	2902554	0.563
Ano da admissão T0	4608534	0.599
Ano do procedimento 1	4618554	0.668
Cirurgia Cardiovascular	3270469	0.669
Espirometria / Ergoespirometria	3260934	0.85
Interconsulta médica	3250514	0.854
Aortografia	3258853	0.967
Antiplaquetario VO	2900586	NaN
Hormonio tireoidiano	2900586	NaN
Broncodiltador	2900586	NaN
Stent	3259116	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	42.49	< 0.001
Tipo de Procedimento 1	40.39	< 0.001
Tipo de Reoperação 1	53.16	< 0.001
Tipo de Procedimento 1	53.16	< 0.001
Tipo de Dispositivo ao final do procedimento 1	50.34	< 0.001
Tipo de Dispositivo ao final do procedimento 1	20.75	< 0.001
Admissão em até 180 dias antes da T0	52.95	< 0.001
Escolaridade	19.24	0.003
Doença cardíaca	12.31	0.005
Infarto do miocárdio prévio / Doença arterial coronariana	7.34	0.013
Doença cardíaca	18.65	0.03
Hemodiálise	7.40	0.039
Classe funcional de IC	12.30	0.039
Endocardite prévia	4.98	0.044
Transplante cardíaco prévio	5.38	0.072
Valvopatias/ Prótese valvares	2.59	0.107
Acidente Vascular Cerebral/ Acidente isquêmico transitório prévios	2.69	0.128
Insuficiência renal crônica	2.42	0.128
Fibrilação / flutter atrial	2.22	0.159
Parada cardíaca prévia/ Taquicardia ventricular instável	1.83	0.185
Diabetes melittus	1.73	0.196
Neoplasia em tratamento ou tratada recentemente	1.74	0.208
Estado de residência	33.30	0.249
Hipertensão arterial	0.51	0.492
Sexo	0.45	0.517
Raça	1.84	0.887
Doença pulmonar obstrutiva crônica	0.00	> 0.999

```
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] 15
```

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

```
## [1] 65
```