Aufgabe1

Max Melchior Lang

8/4/2021

Aufgabe 1

a)

First a `get_mode` function gets defined, which will get wrapped in the `impute` function.

```
### get_mode function
###
### This function computes the mode that is defined as the  the value that appears most
### often in a set of data values.
### Arguments:
### x: A numeric vector. The vector for which the mode should be
### computed.
### na.rm: A logical of length 1. Should NA's be removed when the mode is
### computed?
### Returns: A numeric of length one: the computed mode

get_mode <- function(x, na.rm= FALSE) {
  if(na.rm == TRUE){
    x <- na.omit(x)
  }

   unique_values <- unique(x)
   unique_values[which.max(tabulate(match(x, unique_values)))]
}
```

```
### impute function
###
### This function imputes missing values in a dataset. For metric variables the
### median will replace the missing values, while for categorical variables the mode
### will replace the missing values. This function wraps the previously defined
### 'get_mode' function.
###
### Arguments:
### data: A data frame The dataset for which the missing values should get
### imputed.
###
### Returns: A data frame: The imputed dataset.

impute <- function(data){
  # Input checks
  if(!is.data.frame(data)){
    stop("The Arguments data has to be a data.frame object")
```

```
  }

  # Iteration over columns and rows
  for (column in colnames(data)){
    # Two cases for imputation
      # Metric variable
      if(is.numeric(data[[column]]) && !(all(na.omit(data[[column]]) %in% 0:1))){
        na_index <- which(is.na(data[[column]]))
        col_median <- median(data[[column]], na.rm = TRUE)
        data[[column]] <- replace(data[[column]], na_index, col_median)
      }
      # Categorical variable
      if(any(is.factor(data[[column]]),is.character(data[[column]]),
           is.logical(data[[column]]), all(na.omit(data[[column]]) %in% 0:1))){
      na_index <- which(is.na(data[[column]]))
      col_mode <- get_mode(data[[column]], na.rm = TRUE)
      data[[column]] <- replace(data[[column]], na_index, col_mode)
    }
  }
  return(data)
}
```

b)

```
### pairwise_correlation function
###
### This function computes the pairwise correlation (Pearson) for all metric
### variables in the data set and crosstables for all combinations
### of categorical variables.
### Pairs of metric and categorical variables are ignored.
### The function returns a list with two elements:
###     1) Correlation Matrix of metric variables in the dataset
###     2) A list of crosstables of all categorical variables in the dataset
###
### Arguments:
### data: A data frame The dataset for which the missing values should get
### imputed.
### Returns: A list of two elements : The list contains 1) a correlation matrix
###                                   all metric variables and 2) a list of
###                                   crosstables of all categorical variables
###                                   in the dataset.

pairwise_correlation <- function(data){
  # Input checks
  if(!is.data.frame(data)){
    stop("The Argument data has to be a data.frame object")
  }

  # Declaration of metric and categorical variables
  metric_cols <- vector()
  catego_cols <- vector()

  for(column in 1:ncol(data)){
```

```
  # Metric Variables
    if(is.numeric(data[[column]])&& !all(na.omit(data[[column]]) %in% 0:1)){
      metric_cols <- c(metric_cols,colnames(data[column]))
    }
  # Categorical Variables
    if(any(is.factor(data[[column]]),
         is.character(data[[column]]),
         is.logical(data[[column]]),
         all(na.omit(data[[column]]) %in% 0:1))){
    catego_cols <- c(catego_cols,names(data[column]))
    }
  }

  # Subset of metric and
  metric_data <- data[metric_cols]
  catego_data <- data[catego_cols]

  # Output element metric variables
    # Correlation calculation for metric variables
  cor_matrix <- as.matrix(cor(data[metric_cols],
                              use= "complete.obs", method ="pearson"))

  # Output element categorical variables
   # Creating Matrix with all combinations for crosstable
  catego_combn <- combn(colnames(catego_data), 2)
   # Creating list with crosstables for each combination,
  data_list <- list()
  for(i in 1:ncol(catego_combn)){
  data_list[[i]] <- prop.table(table(data[[catego_combn[1,i]]],
                                     data[[catego_combn[2,i]]]), margin = 2)
  names(data_list)[i] <- paste(catego_combn[1,i], "vs.", catego_combn[2,i])
  }

  # Output (list) of function
  output <- list("correlation_metric_variables"= cor_matrix,
                 "crosstables_catego_variables"= data_list)
  return(output)
}
```

c)

```
data("patient", package = "pammtools")
str(patient)
```

```
## 'data.frame':    2000 obs. of  12 variables:
##  $ Year         : Factor w/ 4 levels "2007","2008",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ CombinedicuID: Factor w/ 456 levels "21","24","25",..: 355 355 355 191 112 112 270 270 270 270 ..
##  $ CombinedID   : int  1110 1111 1116 1316 1410 1414 1507 1509 1518 1520 ...
##  $ Survdays     : num  30.1 30.1 9.8 30.1 30.1 30.1 30.1 30.1 9 30.1 ...
##  $ PatientDied  : num  0 0 1 0 0 0 0 0 1 0 ...
##  $ survhosp     : num  30.1 30.1 9.8 30.1 30.1 5.4 30.1 6.4 8 30.1 ...
##  $ Gender       : Factor w/ 2 levels "Female","Male": 2 1 1 2 2 2 1 2 2 2 ...
##  $ Age          : int  68 57 68 47 69 47 36 58 36 57 ...
```

```
##  $ AdmCatID     : Factor w/ 3 levels "Medical","Surgical Elective",..: 2 1 2 1 2 1 1 1 3 1 ...
##  $ ApacheIIScore: int  20 22 25 16 20 21 18 26 14 16 ...
##  $ BMI          : num  31.6 24.3 18 33.7 38.8 ...
##  $ DiagID2      : Factor w/ 9 levels "Gastrointestinal",..: 2 8 2 6 8 8 9 4 1 1 ...
```

```
# Imputation

patient <- impute(patient)
# Checking for NA values
na_matrix <- matrix(data = NA, nrow = ncol(patient), ncol= 1)
for (i in 1:ncol(patient)){
  na_matrix[i,1] <- sum(is.na(patient[i]))
}
na_matrix
```

```
##       [,1]
##  [1,]    0
##  [2,]    0
##  [3,]    0
##  [4,]    0
##  [5,]    0
##  [6,]    0
##  [7,]    0
##  [8,]    0
##  [9,]    0
## [10,]    0
## [11,]    0
## [12,]    0
```

```
# Pairwise correlation
patient_no_id <- patient[!(colnames(patient) %in% c("CombinedicuID", "CombinedID"))]
pairwise_correlation(patient_no_id)
```

```
## $correlation_metric_variables
##                  Survdays      survhosp         Age ApacheIIScore           BMI
## Survdays       1.00000000  0.4864554699 -0.14846658   -0.21911575  0.0304426441
## survhosp       0.48645547  1.0000000000 -0.09531638   -0.04456446 -0.0002456162
## Age           -0.14846658 -0.0953163830  1.00000000    0.24559797 -0.0245339739
## ApacheIIScore -0.21911575 -0.0445644628  0.24559797    1.00000000  0.0238129354
## BMI            0.03044264 -0.0002456162 -0.02453397    0.02381294  1.0000000000
##
## $crosstables_catego_variables
## $crosstables_catego_variables$`Year vs. PatientDied`
##
##                0         1
##   2007 0.2246835 0.2666667
##   2008 0.2215190 0.2380952
##   2009 0.2500000 0.2142857
##   2011 0.3037975 0.2809524
##
## $crosstables_catego_variables$`Year vs. Gender`
##
##           Female      Male
##   2007 0.2358247 0.2320261
##   2008 0.2177835 0.2295752
##   2009 0.2487113 0.2385621
```

```
##   2011 0.2976804 0.2998366
##
## $crosstables_catego_variables$`Year vs. AdmCatID`
##
##         Medical Surgical Elective Surgical Emergency
##   2007 0.2155887         0.2442244          0.2708758
##   2008 0.2247098         0.2244224          0.2260692
##   2009 0.2537313         0.2145215          0.2321792
##   2011 0.3059701         0.3168317          0.2708758
##
## $crosstables_catego_variables$`Year vs. DiagID2`
##
##        Gastrointestinal Cardio-Vascular     Other Metabolic Neurologic
##   2007        0.2384342       0.2535613 0.1370968 0.3333333  0.2201493
##   2008        0.2419929       0.2364672 0.3306452 0.1666667  0.2164179
##   2009        0.2028470       0.1965812 0.2580645 0.2916667  0.2313433
##   2011        0.3167260       0.3133903 0.2741935 0.2083333  0.3320896
##
##        Orthopedic/Trauma     Renal Respiratory    Sepsis
##   2007         0.2564103 0.1304348   0.2474849 0.1896552
##   2008         0.2051282 0.3913043   0.2072435 0.1839080
##   2009         0.2777778 0.1739130   0.2575453 0.3103448
##   2011         0.2606838 0.3043478   0.2877264 0.3160920
##
## $crosstables_catego_variables$`PatientDied vs. Gender`
##
##        Female      Male
##   0 0.7938144 0.7875817
##   1 0.2061856 0.2124183
##
## $crosstables_catego_variables$`PatientDied vs. AdmCatID`
##
##       Medical Surgical Elective Surgical Emergency
##   0 0.7470978         0.8679868          0.8472505
##   1 0.2529022         0.1320132          0.1527495
##
## $crosstables_catego_variables$`PatientDied vs. DiagID2`
##
##     Gastrointestinal Cardio-Vascular     Other Metabolic Neurologic
##   0        0.8256228       0.7635328 0.7903226 0.8750000  0.7873134
##   1        0.1743772       0.2364672 0.2096774 0.1250000  0.2126866
##
##     Orthopedic/Trauma     Renal Respiratory    Sepsis
##   0         0.8888889 0.7391304   0.7625755 0.7183908
##   1         0.1111111 0.2608696   0.2374245 0.2816092
##
## $crosstables_catego_variables$`Gender vs. AdmCatID`
##
##            Medical Surgical Elective Surgical Emergency
##   Female 0.3963516         0.4224422          0.3462322
##   Male   0.6036484         0.5775578          0.6537678
##
## $crosstables_catego_variables$`Gender vs. DiagID2`
##
```

```
##          Gastrointestinal Cardio-Vascular     Other Metabolic Neurologic
##   Female        0.3736655       0.3646724 0.4112903 0.4166667  0.4216418
##   Male          0.6263345       0.6353276 0.5887097 0.5833333  0.5783582
##
##          Orthopedic/Trauma     Renal Respiratory    Sepsis
##   Female         0.2435897 0.4347826   0.4426559 0.4137931
##   Male           0.7564103 0.5652174   0.5573441 0.5862069
##
## $crosstables_catego_variables$`AdmCatID vs. DiagID2`
##
##                      Gastrointestinal Cardio-Vascular      Other  Metabolic
##   Medical                 0.22419929      0.48148148 0.45967742 1.00000000
##   Surgical Elective       0.25266904      0.36467236 0.25000000 0.00000000
##   Surgical Emergency      0.52313167      0.15384615 0.29032258 0.00000000
##
##                      Neurologic Orthopedic/Trauma      Renal Respiratory
##   Medical            0.56716418        0.31196581 0.52173913  0.92152918
##   Surgical Elective  0.11194030        0.02564103 0.34782609  0.05835010
##   Surgical Emergency 0.32089552        0.66239316 0.13043478  0.02012072
##
##                          Sepsis
##   Medical            1.00000000
##   Surgical Elective  0.00000000
##   Surgical Emergency 0.00000000
```

d)

The correlation between the age of the patients (`age`) and the time until they were released from the hospital (`survhosp`) almost no correlation. The Pearson correlation coefficient is with `-0.0953` slightly negative.

Taking a look at the crosstable of the diagnose (`DiagID2`) and the categorical variable if or if no the patient survived (`PatientDied`). One can see that (relatively) always more people survived than died for each diagnosis. The highest probability to survive is for an `Orthopedic` and `Metabolic` emergencies, the lowest probability to survive is for a `Sepsis` diagnose.

```
# Age and survhosp
pairwise_correlation(patient_no_id)[[1]][3,2, drop= FALSE]
```

```
##        survhosp
## Age -0.09531638
```

```
# DiagID2 and PatientDied
pairwise_correlation(patient_no_id)[[2]][7]
```

```
## $`PatientDied vs. DiagID2`
##
##     Gastrointestinal Cardio-Vascular     Other Metabolic Neurologic
##   0        0.8256228       0.7635328 0.7903226 0.8750000  0.7873134
##   1        0.1743772       0.2364672 0.2096774 0.1250000  0.2126866
##
##     Orthopedic/Trauma     Renal Respiratory    Sepsis
##   0         0.8888889 0.7391304   0.7625755 0.7183908
##   1         0.1111111 0.2608696   0.2374245 0.2816092
```

Session Info

```
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS  10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] de_DE.UTF-8/de_DE.UTF-8/de_DE.UTF-8/C/de_DE.UTF-8/de_DE.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7     purrr_0.3.4
##  [5] readr_1.4.0     tidyr_1.1.3     tibble_3.1.1    ggplot2_3.3.3
##  [9] tidyverse_1.3.1 pammtools_0.5.7
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.6          lubridate_1.7.10    mvtnorm_1.1-1
##  [4] lattice_0.20-44     assertthat_0.2.1    digest_0.6.27
##  [7] foreach_1.5.1       utf8_1.2.1          R6_2.5.0
## [10] cellranger_1.1.0    backports_1.2.1     reprex_2.0.0
## [13] evaluate_0.14       httr_1.4.2          pillar_1.6.0
## [16] rlang_0.4.11        lazyeval_0.2.2      readxl_1.3.1
## [19] rstudioapi_0.13     Matrix_1.3-3        checkmate_2.0.0
## [22] rmarkdown_2.8       splines_4.0.2       munsell_0.5.0
## [25] broom_0.7.6         compiler_4.0.2      numDeriv_2016.8-1.1
## [28] modelr_0.1.8        xfun_0.22           pkgconfig_2.0.3
## [31] mgcv_1.8-35         htmltools_0.5.1.1   tidyselect_1.1.1
## [34] prodlim_2019.11.13  codetools_0.2-18    fansi_0.4.2
## [37] withr_2.4.2         crayon_1.4.1        dbplyr_2.1.1
## [40] timereg_2.0.0       grid_4.0.2          nlme_3.1-152
## [43] jsonlite_1.7.2      gtable_0.3.0        lifecycle_1.0.0
## [46] DBI_1.1.1           magrittr_2.0.1      scales_1.1.1
## [49] cli_2.5.0           stringi_1.6.1       fs_1.5.0
## [52] xml2_1.3.2          ellipsis_0.3.2      generics_0.1.0
## [55] vctrs_0.3.8         Formula_1.2-4       lava_1.6.9
## [58] iterators_1.0.13    tools_4.0.2         glue_1.4.2
## [61] hms_1.0.0           pec_2020.11.17      survival_3.2-11
## [64] yaml_2.2.1          colorspace_2.0-1    rvest_1.0.0
## [67] knitr_1.33          haven_2.4.1
```