

07/18/2023

Matt/Abi code-snippet to guide one using 'cat_trauma' function

```
suppressWarnings({
  invisible({
    rm(list = ls())

    if (!require(icdpicr, quietly = TRUE, warn.conflicts = FALSE)) install.packages('icdpicr')
    if (!require(dplyr, quietly = TRUE, warn.conflicts = FALSE)) install.packages('dplyr')
    if (!require(readr, quietly = TRUE, warn.conflicts = FALSE)) install.packages('readr')
    if (!require(tidyr, quietly = TRUE, warn.conflicts = FALSE)) install.packages('tidyr')

    library(icdpicr, quietly = TRUE, warn.conflicts = FALSE)
    library(dplyr, quietly = TRUE, warn.conflicts = FALSE)
    library(readr, quietly = TRUE, warn.conflicts = FALSE)
    library(tidyr, quietly = TRUE, warn.conflicts = FALSE)
  })
})
```

720.1 install the relevant packages

```
ls("package:icdpicr")
```

720.2 get some documentation

```
## [1] "cat_trauma" "injury"
```

```
print(cat_trauma)
```

720.3 study the syntax

```
## function (df, dx_pre, icd10, i10_iss_method, calc_method = 1,
##   verbose = FALSE)
## {
##   if (!is.data.frame(df))
##     stop("First argument must be a dataframe")
##   if (NROW(df) == 0)
##     stop("Data contains no observations. It must contain at least one row.")
##   if (!is.character(dx_pre))
##     stop("Second argument must be a character string")
##   if (make.names(dx_pre) != dx_pre)
##     stop("Second argument must be a valid variable name in R")
##   if (!(calc_method %in% c(1, 2)))
##     stop("calc_method must be either 1 or 2")
##   if (!(icd10 %in% c(TRUE, FALSE, "cm", "base")))
##     stop("icd10 must be TRUE, FALSE, 'cm', or 'base'")
##   if (icd10 == FALSE)
##     i10_iss_method <- ""
```

```

## if (i10_iss_method == "roc_max")
##   stop("The roc_max option has been deprecated. Please use roc_max_NIS, roc_max_TQIP, roc_max_
## if ((icd10 != FALSE) && !(i10_iss_method %in% c("roc_max_NIS",
##   "roc_max_TQIP", "roc_max_NIS_only", "roc_max_TQIP_only",
##   "gem_max", "gem_min")))
##   stop("i10_iss_method must be roc_max_NIS, roc_max_TQIP, roc_max_NIS_only, roc_max_TQIP_only
## regex_dx <- paste0("^", dx_pre, "([0-9]+)$")
## dx_colnames <- grep(regex_dx, names(df), value = TRUE)
## dx_nums <- as.numeric(sub(regex_dx, "\\1", dx_colnames))
## num_dx <- length(dx_nums)
## if (num_dx == 0)
##   stop("No variables with prefix found in data")
## df <- data.frame(df)
## if (isTRUE(icd10))
##   icd10 <- "cm"
## if (icd10 %in% c("base", "cm")) {
##   etab <- rbind(etab_s1, i10_ecode)
##   ntab <- switch(i10_iss_method, roc_max_NIS = rbind(ntab_s1,
##     .select_i10_data("NIS", icd10)), roc_max_TQIP = rbind(ntab_s1,
##     .select_i10_data("TQIP", icd10)), roc_max_NIS_only = rbind(ntab_s1,
##     .select_i10_data("NIS_only", icd10)), roc_max_TQIP_only = rbind(ntab_s1,
##     .select_i10_data("TQIP_only", icd10)), gem_max = rbind(ntab_s1,
##     i10_map_max), gem_min = rbind(ntab_s1, i10_map_min))
## }
## else {
##   ntab <- ntab_s1
##   etab <- etab_s1
## }
## for (i in dx_nums) {
##   dx_name <- paste0(dx_pre, i)
##   df_ss <- df[, dx_name, drop = FALSE]
##   df_ss$n <- 1:NROW(df_ss)
##   df_ss[, dx_name] <- sub("\\.", "", df_ss[, dx_name])
##   if (icd10 == TRUE & i10_iss_method == "roc_max") {
##     i9_valid <- c("8", "9", "E")
##     i10_valid <- c("S", "T", "U", "V", "W", "X", "Y")
##     df_ss[, dx_name] <- ifelse(substr(df_ss[, dx_name],
##       1, 1) %in% c(i9_valid, i10_valid), df_ss[, dx_name],
##       NA)
##     process_i10 <- function(s) {
##       stopifnot(is.character(s) | is.na(s))
##       ret_val <- NA
##       s <- sub("\\.", "", s)
##       if (!substr(s, 1, 1) %in% c("S", "T", "U", "V",
##         "W", "X", "Y")) {
##         ret_val <- s
##       }
##       else if (nchar(s) < 7 & !grepl("X", substr(s,
##         2, nchar(s)))) {
##         ret_val <- s
##       }
##       else if (nchar(s) != 7) {
##         ret_val <- ""
##       }
##     }
##   }

```

```

##         else if (substr(s, 7, 7) != "A") {
##             ret_val <- ""
##         }
##         else if (substr(s, 5, 5) == "X") {
##             ret_val <- substr(s, 1, 4)
##         }
##         else if (substr(s, 6, 6) == "X") {
##             ret_val <- substr(s, 1, 5)
##         }
##         else {
##             ret_val <- substr(s, 1, 6)
##         }
##         return(ret_val)
##     }
##     df_ss[, dx_name] <- sapply(df_ss[, dx_name], process_i10)
## }
## temp <- merge(df_ss, ntab, by.x = dx_name, by.y = "dx",
##             all.x = TRUE, all.y = FALSE, sort = FALSE)
## temp <- temp[order(temp$n), ]
## temp <- temp[, c("severity", "issbr")]
## if (calc_method == 2) {
##     temp[which(temp$severity == 6), "severity"] <- 5
## }
## names(temp) <- paste0(c("sev_", "issbr_"), i)
## df <- .insert_columns(df, dx_name, temp)
## }
## body_regions <- unique(i10_map_max$issbr)
## issbr_names <- gsub("/", "_", body_regions)
## for (i in body_regions) {
##     temp <- df[, grepl("sev_", names(df)), drop = FALSE] *
##         (1 * (df[, grepl("issbr_", names(df))] == i))
##     df[, paste0("mxaisbr_", gsub("/", "", i))] <- apply(temp,
##         1, function(row) {
##             row <- ifelse(row == 0, NA, row)
##             if (all(is.na(row))) {
##                 maxaisbr <- 0
##             }
##             else if (all(row == 9, na.rm = TRUE)) {
##                 maxaisbr <- 9
##             }
##             else {
##                 maxaisbr <- max(c(0, row[row != 9]), na.rm = TRUE)
##             }
##             return(maxaisbr)
##         })
## }
## c9to0 <- function(x) ifelse(x == 9, 0, x)
## df$maxais <- apply(df, 1, function(row) {
##     row <- row[grepl("mxaisbr", names(row))]
##     if (all(is.na(row))) {
##         maxais <- as.numeric(NA)
##     }
##     else if (max(c9to0(row), na.rm = TRUE) == 0) {
##         maxais <- max(row, na.rm = TRUE)
##     }
## }

```

```

##     }
##     else {
##         maxais <- max(c9to0(row), na.rm = TRUE)
##     }
##     return(maxais)
## })
## df$maxais <- as.numeric(df$maxais)
## df$riss <- apply(df, 1, function(row) {
##     temp <- row[grepl("^mxaisbr", names(row))]
##     temp <- as.numeric(c9to0(temp))
##     sum(temp[order(-temp)[1:3]]^2)
## })
## df[df$maxais == 6, "riss"] <- 75
## df[df$maxais == 9, "riss"] <- NA
## df$niss <- apply(df, 1, function(row) {
##     temp <- row[grepl("^sev_", names(row))]
##     temp <- as.numeric(temp)
##     temp <- ifelse(is.na(temp) | temp == 9, 0, temp)
##     sum(temp[order(-temp)[1:3]]^2)
## })
## df[df$maxais == 6, "niss"] <- 75
## df[df$maxais == 9, "niss"] <- NA
## ecode_colnames <- paste0("ecode_", 1:4)
## df[, ecode_colnames] <- NA
## ecode_regex <- paste0("^", etab$dx, collapse = "|")
## df[, ecode_colnames] <- t(apply(df, 1, function(row) {
##     row <- sub("\\.", "", row)
##     row_ecodes <- stringr::str_extract(as.character(unlist(row)),
##         ecode_regex)
##     row_ecodes <- na.omit(row_ecodes)
##     row_ecodes[1:4]
## })))
## for (i in 1:4) {
##     col_name <- paste("ecode_", i, sep = "")
##     df_ss <- df[, col_name, drop = FALSE]
##     df_ss$n <- 1:NROW(df_ss)
##     df_ss[, col_name] <- sub("\\.", "", df_ss[, col_name])
##     temp <- merge(df_ss, etab, by.x = col_name, by.y = "dx",
##         all.x = TRUE, all.y = FALSE, sort = FALSE)
##     temp <- temp[order(temp$n), ]
##     temp <- temp[, c("mechmaj", "mechmin", "intent")]
##     names(temp) <- paste(c("mechmaj", "mechmin", "intent"),
##         i, sep = "")
##     df <- .insert_columns(df, col_name, temp)
## }
## if (stringr::str_detect(i10_iss_method, "NIS|TQIP") && icd10 %in%
##     c("cm", "base")) {
##     if (verbose)
##         print("Calculating mortality prediction")
##     coef_df <- .select_i10_coef(prefix = stringr::str_extract(i10_iss_method,
##         "NIS|TQIP"), icd10)
##     stopifnot(max(coef_df$intercept, na.rm = TRUE) == min(coef_df$intercept,
##         na.rm = TRUE))
##     intercept <- max(coef_df$intercept, na.rm = TRUE)

```

```
##      coef_df <- coef_df[!is.na(coef_df$effect), ]
##      effect_hash <- coef_df$effect
##      names(effect_hash) <- coef_df$dx
##      calc_mortality_prediction <- function(dx) {
##          x <- sum(effect_hash[sub("\\.", "", dx)], na.rm = TRUE) +
##              intercept
##          1/(1 + exp(-x))
##      }
##      mat <- as.matrix(df[, grepl(paste0("^", dx_pre), names(df))])
##      df$Pmort <- apply(mat, 1, calc_mortality_prediction)
##  }
##  rownames(df) <- 1:nrow(df)
##  df
## }
```

<bytecode: 0x7fe2ac9beb70>
<environment: namespace:icdpicr>

```
print(injury)
```

720.4 what is the value of dx_pre?

```
## # A tibble: 100,477 x 11
##   dx1      dx2      dx3      dx4      dx5 dx6 dx7 dx8 dx9 dx10 died
##   <chr>   <chr>   <chr>   <chr>   <chr> <chr> <chr> <chr> <chr> <chr> <int>
## 1 S72.342A <NA>      <NA>      <NA>      <NA> <NA> <NA> <NA> <NA> <NA>      0
## 2 S05.22XA <NA>      <NA>      <NA>      <NA> <NA> <NA> <NA> <NA> <NA>      0
## 3 S00.01XA S00.03XA S00.11XA S00.12XA S00.~ S00.~ S00.~ S01.~ S02.~ S80.~      0
## 4 S21.119A <NA>      <NA>      <NA>      <NA> <NA> <NA> <NA> <NA> <NA>      0
## 5 S82.191A <NA>      <NA>      <NA>      <NA> <NA> <NA> <NA> <NA> <NA>      0
## 6 S22.42XA <NA>      <NA>      <NA>      <NA> <NA> <NA> <NA> <NA> <NA>      0
## 7 S92.052A S92.065A S92.325A S92.335A S92.~ S93.~ <NA> <NA> <NA> <NA>      0
## 8 S02.112A S06.5X0A S12.090A S12.100A S12.~ S20.~ S20.~ S22.~ S22.~ S22.~      0
## 9 S00.03XA S22.058A S22.068A S22.078A S22.~ S30.~ S42.~ S62.~ S81.~ S82.~      0
## 10 S61.411A S62.624B S62.626B S66.300A S66.~ S66.~ S66.~ <NA> <NA> <NA>      0
## # i 100,467 more rows
```

```
# use subset of injury given its size
```

```
inj = injury[1:100,1:3]
```

```
df_score = cat_trauma(df=inj,dx_pre="dx",icd10=TRUE,i10_iss_method="roc_max_NIS",calc_method = 1,verbos
```

```
# visualize the output
```

```
df_score[1:9,1:9]; df_score[1:9,10:14]; df_score[1:9,15:18] ; df_score[1:9,34:35]
```

720.5 as such the syntax is:

```
##      dx1 sev_1      issbr_1      dx2 sev_2      issbr_2      dx3 sev_3
## 1 S72.342A      1 Extremities <NA>      NA      <NA>      <NA>      NA
## 2 S05.22XA      1      Face <NA>      NA      <NA>      <NA>      NA
## 3 S00.01XA      1 Head/Neck S00.03XA      2 Head/Neck S00.11XA      2
## 4 S21.119A      3      Chest <NA>      NA      <NA>      <NA>      NA
## 5 S82.191A      3 Extremities <NA>      NA      <NA>      <NA>      NA
## 6 S22.42XA      3      Chest <NA>      NA      <NA>      <NA>      NA
```

```

## 7 S92.052A      1 Extremities S92.065A      1 Extremities S92.325A      1
## 8 S02.112A      1          Face S06.5X0A      4    Head/Neck S12.090A      4
## 9 S00.03XA      2    Head/Neck S22.058A      1          Chest S22.068A      3
##      issbr_3
## 1      <NA>
## 2      <NA>
## 3      Face
## 4      <NA>
## 5      <NA>
## 6      <NA>
## 7 Extremities
## 8    Head/Neck
## 9      Chest

##      mxaisbr_General mxaisbr_HeadNeck mxaisbr_Face mxaisbr_Extremities
## 1              0              0              0              1
## 2              0              0              1              0
## 3              0              2              2              0
## 4              0              0              0              0
## 5              0              0              0              3
## 6              0              0              0              0
## 7              0              0              0              1
## 8              0              4              1              0
## 9              0              2              0              0

##      mxaisbr_Chest
## 1              0
## 2              0
## 3              0
## 4              3
## 5              0
## 6              3
## 7              0
## 8              0
## 9              3

##      mxaisbr_Abdomen maxais riss niss
## 1              0          1    1    1
## 2              0          1    1    1
## 3              0          2    8    9
## 4              0          3    9    9
## 5              0          3    9    9
## 6              0          3    9    9
## 7              0          1    1    3
## 8              0          4   17   33
## 9              0          3   13   14

##      intent4      Pmort
## 1      <NA> 0.013857916
## 2      <NA> 0.014412795
## 3      <NA> 0.017673701
## 4      <NA> 0.026078454
## 5      <NA> 0.027839150
## 6      <NA> 0.024364215
## 7      <NA> 0.005516156
## 8      <NA> 0.037222518

```

9 <NA> 0.021037152