900_funcionExtraccionDummies.R

Daniel

2021-08-31

Función de separación de dummies

Esta función se puede utilizar para la separación y conteo de respuestas en preguntas de múltiples atributos.

@param vector vector de caracteres, con un delimitador (por defecto es '\,') @param descartar booleano, descartar valores NAN @param delimitador carácter, delimitador para elementos de campo

@return @export

@examples vect_ejemplo <- c('Manzana, Pera, Oso, Perro', 'Manzana, Jaguar, Perro', 'Naranja, Orca, Pera', 'Leon, Cerdo', 'Manzana, Perro')

separarDummies(vect_ejemplo)

```
separarDummies <- function(vector, descartar = F, delimitador = '\\,') {
  # Separar los factores únicos
  dimFactor <- lapply(vector, function(x) str_split(x, delimitador)) %>%
    unlist() %>%
    sapply(., function(x){str_trim(x)}) %>%
    unique()

  # print(dimFactor)
  if (descartar) {
    dimFactor <- discard(dimFactor, is.na)
  }

  # Lista vacía
  ls_factorEscogencia <- list()

  # Llenar las listas vacías con verdadero o falso si tiene la palabra
  for (i in 1:length(dimFactor)) {

    ls_factorEscogencia[[i]] <- vector %>%
      sapply(., function(x){str_detect(x, dimFactor[i])})
  }

  # Convertir lista a dataframe
  dataframeElementos <- as_tibble(do.call(cbind, ls_factorEscogencia))
  colnames(dataframeElementos) <- dimFactor


  return(dataframeElementos)
}
```

```r
graficoVariables <- function(var, fill_color = 'green', alpha_fill = 0.5, contour_color = 'green4') {
  dimVar <- dim(var)

  lista <- apply(var, 2, function(x) {
    sum(x)
  })

  df <- tibble(nombre = names(lista),
               Frec = lista,
               Frec_rel = lista)

  g <- df %>%
    ggplot(aes(x = Frec, y = fct_reorder(nombre, Frec))) +
    geom_bar(stat = 'identity', fill = alpha(fill_color, alpha_fill), color = contour_color) +
    theme(axis.title.y = element_blank()) +
    xlab('Frecuencia')

  return(list(df = df, g = g))
}


graficoBarrasAnidado <- function(df, xvar, xlab, yvar = 'perc', ylab = 'Porcentaje (%)', title = '', op
  xvar_quo <- rlang::ensym(xvar)
  yvar_quo <- rlang::ensym(yvar)

  df1 <- df %>% filter(!!yvar_quo > 0.1)

  ggplot(df, aes(x = !!xvar_quo,
                 color = name,
                 y = !!yvar_quo * 100,
                 fill = name)) +
    geom_bar(stat = 'identity') +
    geom_text(aes(label = scales::percent(!!yvar_quo, 1),
                  y = !!yvar_quo * 100), color = 'white',
              fill = 'white', position = position_stack(vjust = 0.5), size = 3) +
    scale_fill_viridis_d(option = option) +
    scale_color_viridis_d(option = option) +
    xlab(xlab) + ylab(ylab) +
    labs(title = title) +
    theme(legend.title = element_blank())
}
```