

coverage - 99.47%

Files R/eq5d.R

```

1 #####  

2 #' Function to check the EQ-5D-3L scores  

3 #' @param dimen a must input,response for EQ-5D-3L mobility or the 5 digit  

4 #' response, or the vector of responses, e.g. 11111, c(1, 1, 1, 1, 1) or 1  

5 #' @param dimen2 response for EQ-5D-3L self care, or NA if the responses  

6 #' are given as dimensions  

7 #' @param dimen3 response for EQ-5D-3L usual activities,or NA if the  

8 #' responses are given as dimensions  

9 #' @param dimen4 response for EQ-5D-3L pain/discomfort, or NA if the  

10 #' responses are given as dimensions  

11 #' @param dimen5 response for EQ-5D-3L anxiety/depression, or NA if  

12 #' the responses are given as dimensions  

13 #' @examples  

14 #' check_scores_3L(c(1, 2, 3, 3, 3))  

15 #' check_scores_3L(1, 2, 3, 3, 3)  

16 #' check_scores_3L(1, 2, 3, 2, 3)  

17 #' check_scores_3L(12323)  

18 #' @export  

19 check_scores_3L <- function(dimen, dimen2 = NA, dimen3 = NA, dimen4 = NA,  

20 dimen5 = NA) {  

21   7605745x responses <- c(dimen, dimen2, dimen3, dimen4, dimen5)  

22   # first value should be not be a NA, do not contain NA  

23   7605745x if (sum(is.na(dimen)) > 0) {  

24     6x       this_score <- NA  

25     6x       return(NA)  

26   } else {  

27     7605739x       if (length(dimen) != 5 && length(dimen) != 1) {  

28       6x         stop("Invalid EQ-5D-3L responses-check the responses to each question")  

29     } else {  

30       7605733x         if (length(dimen) == 5) { # first value a vector  

31         21x           this_score <- paste(dimen, collapse = "")  

32         21x           responses <- dimen  

33       } else {  

34         7605712x           if (length(dimen) == 1) {  

35           7605712x             this_score <- paste(responses[!is.na(responses)], collapse = "")  

36           # first value 5 digit number or actual response for mobility  

37           7605712x             responses <- convert_number_to_digits(this_score)  

38         }  

39       }  

40     }  

41   }  

42 }  

43 7605725x if (!all(responses %in% 1:3)) {  

44   15x     stop("Responses not valid for EQ-5D-3L scores")  

45 } else {  

46   7605710x     this_score <- as.numeric(this_score)  

47   7605710x     if (this_score < 11111) {  

48     10x       return(NA)  

49   } else {  

50     7605700x       return(responses)  

51   }  

52 }  

53 #####  

54 #' Function to check the EQ-5D-5L scores  

55 #' @param dimen a must input,response for EQ-5D-3L mobility or the 5 digit  

56 #' response, or the vector of responses, e.g. 11111, c(1,1,1,1,1) or 1  

57 #' @param dimen2 response for EQ-5D-5L self care, or NA if the responses are  

58 #' given as dimensions  

59 #' @param dimen3 response for EQ-5D-5L usual activities,or NA if the responses  

60 #' are given as dimensions  

61 #' @param dimen4 response for EQ-5D-5L pain/discomfort, or NA if the responses  

62 #' are given as dimensions  

63 #' @param dimen5 response for EQ-5D-5L anxiety/depression, or NA if the  

64 #' responses are given as dimensions  

65 #' @examples  

66 #' check_scores_5L(c(1, 2, 3, 5, 3))  

67 #' check_scores_5L(1, 2, 3, 4, 3)  

68 #' check_scores_5L(12323)  

69 #' @export  

70 check_scores_5L <- function(dimen, dimen2 = NA, dimen3 = NA, dimen4 = NA,  

71 dimen5 = NA) {  

72   65681x   responses <- c(dimen, dimen2, dimen3, dimen4, dimen5)  

73   65681x   if (sum(is.na(dimen)) > 0) {  

74     7x       # first value should be not be a NA, do not contain NA  

75     7x       this_score <- NA  

76     7x       return(NA)  

77   } else {  

78     65674x       if (length(dimen) != 5 && length(dimen) != 1) {  

79       3x         stop("Invalid EQ-5D-5L responses-check the responses to each question")  

80     } else {  

81       65671x         if (length(dimen) == 5) { # first value a vector  

82         17x           this_score <- paste(dimen, collapse = "")  

83         17x           responses <- dimen  

84       } else {  

85         65654x           if (length(dimen) == 1) {  

86           65654x             this_score <- paste(responses[!is.na(responses)], collapse = "")  

87           # first value 5 digit number or actual response for mobility  

88           65654x             responses <- convert_number_to_digits(this_score)  

89         }  

90       }  

91     }  

92   }  

93 }
```

```

91      }
92    }
93  }
94 65665X if (!all(responses %in% 1:5)) {
95    4X   stop("Responses not valid for EQ-5D-5L scores")
96  } else {
97    65661X   this_score <- as.numeric(this_score)
98    65661X   if (this_score < 11111) {
99      2X     return(NA)
100    }else{
101      65659X       return(responses)
102    }
103  }
104 #####
105 #' Function to value EQ-5D-5L scores for various countries
106 #' @param country a country name from the List Canada,China,England,
107 #' Germany,HongKong,Indonesia,Ireland,Japan,Korea,Malaysia,Netherlands,
108 #' Poland,Spain,Taiwan,Thailand, and Uruguay
109 #' @param dimen a must input,response for EQ-5D-5L mobility or the 5 digit
110 #' response, or the vector of responses, e.g. 11111, c(1,1,1,1,1) or 1
111 #' @param dimen2 response for EQ-5D-5L self care, or NA if the responses are
112 #' given as dimen
113 #' @param dimen3 response for EQ-5D-5L usual activities,or NA if the responses
114 #' are given as dimen
115 #' @param dimen4 response for EQ-5D-5L pain/discomfort, or NA if the responses
116 #' are given as dimen
117 #' @param dimen5 response for EQ-5D-5L anxiety/depression, or NA if the
118 #' responses are given as dimen
119 #' @return index values if success, negative values if failure
120 #' @examples
121 #' value_5L_Ind("England", 23434)
122 #' value_5L_Ind("China", 2, 3, 4, 3, 4)
123 #' value_5L_Ind("Poland", c(1, 2, 3, 4, 3))
124 #' @export
125 value_5L_Ind <- function(country, dimen, dimen2 = NA, dimen3 = NA, dimen4 = NA,
126                           dimen5 = NA) {
127
128  65674X   country_list <- c(
129    "Canada", "China", "England", "Ethiopia", "France", "Germany", "Hong_Kong",
130    "Indonesia", "Ireland", "Japan", "Korea", "Malaysia", "Netherlands",
131    "Poland", "Portugal", "Spain", "Taiwan", "Thailand", "Uruguay", "USA",
132    "Vietnam"
133  )
134 65674X   country <- replace_space_underscore(country)
135 65674X   if (country %in% country_list) {
136    65671X     scores <- check_scores_5L(dimen, dimen2, dimen3, dimen4, dimen5)
137    9X       if (sum(is.na(scores)) > 0) return(NA)
138    65653X     if (sum(scores) > 0) {
139      65653X       eq5d_valueset <- EQ5D5L_tariffs.df
140      65653X       names(scores) <- c("MO", "SC", "UA", "PD", "AD")
141      65653X       rows <- paste0(names(scores), scores)
142      65653X       rownum1 <- which(row.names(eq5d_valueset) == rows[1])
143      65653X       rownum2 <- which(row.names(eq5d_valueset) == rows[2])
144      65653X       rownum3 <- which(row.names(eq5d_valueset) == rows[3])
145      65653X       rownum4 <- which(row.names(eq5d_valueset) == rows[4])
146      65653X       rownum5 <- which(row.names(eq5d_valueset) == rows[5])
147      65653X       rownumfh <- which(row.names(eq5d_valueset) == "fullHealth")
148      65653X       rownuminter <- which(row.names(eq5d_valueset) == "intercept")
149      65653X       rownum4 <- which(row.names(eq5d_valueset) == "N4")
150      65653X       rownum45 <- which(row.names(eq5d_valueset) == "Num45sq")
151      65653X       inter_value <- NA
152      65653X       if (any(scores > 1) && !is.na(eq5d_valueset[rownuminter, country])) {
153        12498X         inter_value <- eq5d_valueset[rownuminter, country]
154      }
155      65653X       n4value <- NA
156      65653X       if (any(scores >= 4) && !is.na(eq5d_valueset[rownum4, country])) {
157        2883X          n4value <- eq5d_valueset[rownum4, country]
158      }
159      65653X       n45 <- which(scores %in% c(4, 5))
160      65653X       n45value <- NA
161      65653X       if (length(n45) >= 1 & !is.na(eq5d_valueset[rownum45, country])) {
162        2883X          n45value <- (length(n45) - 1)^2 * eq5d_valueset[rownum45, country]
163      }
164      65653X       n45all <- 0
165      65653X       if (length(n45) >= 1) {
166        60530X         for (i in seq_len(length(n45))) {
167          131273X           names45row <- paste0(names(scores)[n45[i]], "45")
168          131273X           rownum45r <- which(row.names(eq5d_valueset) == names45row)
169          131273X           if (!is.na(eq5d_valueset[rownum45r, country])) {
170            6252X              n45rvalue <- eq5d_valueset[rownum45r, country]
171            6252X              n45sall <- n45sall + n45rvalue
172            } else {
173              125021X                n45rvalue <- 0
174              125021X                n45sall <- n45sall + n45rvalue
175            }
176          }
177        }
178      65653X       dim_response <- c(
179        eq5d_valueset[rownum1, country], eq5d_valueset[rownum2, country],
180        eq5d_valueset[rownum3, country],
181        eq5d_valueset[rownum4, country], eq5d_valueset[rownum5, country]
182      )
183      65653X       sum_response <- sum(dim_response, na.rm = TRUE)
184      65653X       values <- c(
185        eq5d_valueset[rownumfh, country], inter_value, sum_response,
186        n4value, n45value, n45sall
187      )
188      65653X       values_state <- sum(values, na.rm = TRUE)

```

```

189 |     return(values_state)
190 |   }
191 | } else {
192 |   3x   stop("No tariffs found for the country you specified for EQ-5D-5L. Please try later")
193 |
194 | }
195 |
196 #####'
197 #' Function to value EQ-5D-5L scores for any country and group by gender and age
198 #' @param eq5dresponse_data the data containing eq5d responses
199 #' @param mo column name for EQ-5D-5L mobility
200 #' @param sc column name for response for EQ-5D-5L self care
201 #' @param ua column name for response for EQ-5D-5L usual activities
202 #' @param pd column name for response for EQ-5D-5L pain/discomfort
203 #' @param ad column name for response for EQ-5D-5L anxiety/depression
204 #' @param country country of interest, by default is England
205 #' @param groupby male or female -grouping by gender, default NULL
206 #' @param agelimit vector of ages to show upper and lower limits, default NULL
207 #' @return index value if success, negative values for failure
208 #' @examples
209 #' data <- data.frame(
210 #'   age = c(10, 20), sex = c("M", "F"),
211 #'   mo = c(1, 2), sc = c(1, 2), ua = c(3, 4), pd = c(3, 4), ad = c(3, 4)
212 #' )
213 #' value_5L(data, "mo", "sc", "ua", "pd", "ad", country = "England",
214 #'           groupby = NULL, agelimit = NULL) {
215 |
216 |   11x   country <- replace_space_underscore(country)
217 |   10x   eq5d_colnames <- c(mo, sc, ua, pd, ad)
218 |   10x   ans_eq5d_colnames <- sapply(eq5d_colnames, check_column_exist, eq5dresponse_data)
219 |   10x   if (all(ans_eq5d_colnames == 0)) { # if the eq5d column names match
220 |     9x     working_data <- subset_gender_age_to_group(eq5dresponse_data, groupby,
221 |                                                 agelimit)
222 |     9x     scores <- c()
223 |     9x     if (nrow(working_data) < 1) {
224 |       1x         stop("no entries with the given criteria - Please check the contents
225 |                           or the criteria")
226 |     } else {
227 |       8x         for (j in 1:nrow(working_data)) {
228 |         11x           res1 <- working_data[j, mo]
229 |         11x           res2 <- working_data[j, sc]
230 |         11x           res3 <- working_data[j, ua]
231 |         11x           res4 <- working_data[j, pd]
232 |         11x           res5 <- working_data[j, ad]
233 |         11x           this_score <- value_5L_Ind(country, c(res1, res2, res3, res4, res5))
234 |         10x           scores <- c(scores, this_score)
235 |       }
236 |     }
237 |   }
238 |   7x   new_data <- cbind(working_data, scores)
239 |   7x   colnames(new_data) <- c(colnames(working_data), "EQ-5D-5L scores")
240 |   7x   scores_noNA <- scores[is.na(scores)]
241 |   7x   if (length(scores_noNA) >= 1) {
242 |     5x       stats <- descriptive_stat_data_column(scores_noNA, "EQ-5D-5L")
243 |     5x       freq_table <- get_frequency_table(scores_noNA)
244 |     5x       first <- is.null(groupby) || toupper(groupby) == "NA" || is.na(groupby)
245 |     5x       second <- is.null(agelimit) || sum(toupper(agelimit)) == "NA" != 0 ||
246 |     5x       sum(is.na(agelimit)) != 0
247 |     5x       if (first & second) {
248 |       1x         title <- paste("Histogram of EQ-5D-5L index values", sep = "")
249 |     } else {
250 |       4x         if (first & !second) {
251 |         2x           title <- paste("Histogram of EQ-5D-5L index values",
252 |                           "with ages between ", agelimit[1], " and ", agelimit[2],
253 |                           sep = "")
254 |         }
255 |       }
256 |     2x     } else {
257 |       1x       if (!first & second) {
258 |         1x           title <- paste("Histogram of EQ-5D-5L index values for ",
259 |                           groupby,
260 |                           sep = ""
261 |         }
262 |       } else {
263 |         1x           title <- paste("Histogram of EQ-5D-5L index values for ",
264 |                           groupby, " with ages between ", agelimit[1], " and ", agelimit[2],
265 |                           sep = ""
266 |         }
267 |     }
268 |   }
269 |   5x   oldpar <- graphics::par(no.readonly = TRUE)
270 |   graphics::par(mar = c(4, 4, 2, 2))
271 |
272 |   5x   hist_plot <- graphics::hist(scores_noNA, main = title)
273 |   results <- list("stats" = stats, "freq_table" = freq_table,
274 |                  "histogram" = hist_plot, "modified_data" = new_data)
275 |   5x   return(results)
276 |   !
277 |   on.exit(graphics::par(oldpar))
278 |   } else {
279 |     2x       print("No relevant rows with non NA scores")
280 |   }
281 |   }
282 |   1x   } else {
283 |     stop("EQ-5D column names do not match")
284 |   }
285 |
286 #####'
287 #' Function to value EQ-5D-3L scores for various countries

```

```

287  #' @param country a country name from the list Belgium,Brazil,Canada,Chile,
288  #' Denmark,Europe,Finland,France,Germany,Italy,Japan,Korea,Netherlands,
289  #' NewZealand,Poland,Portugal,Slovenia,Spain,Taiwan,Thailand,UK,USA, and Zimbabwe
290  #' @param method method name either TTO or VAS
291  #' @param dimen a must input,response for EQ-5D-5L mobility or the 5 digit
292  #' response, or the vector of responses, e.g. 11111, c(1,1,1,1,1) or 1
293  #' @param dimen2 response for EQ-5D-3L self care, or NA if the responses are
294  #' given as dimen
295  #' @param dimen3 response for EQ-5D-3L usual activities,or NA if the responses
296  #' are given as dimen
297  #' @param dimen4 response for EQ-5D-3L pain/discomfort, or NA if the responses
298  #' are given as dimen
299  #' @param dimen5 response for EQ-5D-3L anxiety/depression, or NA if the
300  #' responses are given as dimen
301  #' @return index value based if success, negative values for failure
302  #' @examples
303  #' value_3L_Ind("UK", "TTO", 23131)
304  #' value_3L_Ind("Spain", "TTO", 2, 3, 1, 3, 1)
305  #' value_3L_Ind("Denmark", "VAS", c(1, 2, 3, 1, 3))
306  #' @export
307  value_3L_Ind <- function(country, method, dimen, dimen2 = NA, dimen3 = NA,
308                           dimen4 = NA, dimen5 = NA) {
309  7605743X   countrylist <- c(
310  7605743X     "Argentina", "Australia", "Belgium", "Brazil", "Canada", "Chile", "China",
311  7605743X     "Denmark", "Europe", "Finland", "France", "Germany", "Iran", "Italy", "Japan",
312  7605743X     "Korea", "Malaysia", "Netherlands", "New_Zealand", "Poland", "Portugal",
313  7605743X     "Singapore", "Slovenia", "Spain", "Sri_Lanka", "Sweden",
314  7605743X     "Taiwan", "Thailand", "Trinidad_and_Tobago", "UK", "USA", "Zimbabwe"
315  )
316
317  7605743X   VAS_countrylist <- c(
318  7605743X     "Argentina", "Belgium", "Denmark", "Europe", "Finland", "Germany", "Malaysia",
319  7605743X     "New_Zealand", "Slovenia", "Spain", "UK"
320  )
321  7605743X   TTO_countrylist <- c(
322  7605743X     "Argentina", "Australia", "Brazil", "Canada", "Chile", "China", "Denmark",
323  7605743X     "France", "Germany", "Iran", "Italy", "Japan", "Korea", "Netherlands", "Poland",
324  7605743X     "Portugal", "Singapore", "Spain", "Sri_Lanka", "Sweden",
325  7605743X     "Taiwan", "Thailand", "Trinidad_and_Tobago", "UK", "USA", "Zimbabwe"
326  )
327
328  7605743X   australia.impalausibleordering.scores <- c(33132, 12133, 13133, 22133, 23133, 32133, 33133, 12233, 13233, 22233, 23233, 32233, 33233, 33323, 33323, 13332, 13333,
329  7605743X   country %> replace_space_underscore(country)
330  7605743X   if (country %in% countrylist) {
331  7605735X     scores <- check_scores_3L(dimen, dimen2, dimen3, dimen4, dimen5)
332  16X       if (sum(is.na(scores)) > 0) return(NA)
333  7605696X     if (sum(scores) > 0) {
334  7605696X       if (method == "TTO" & country %in% TTO_countrylist) {
335  7603006X         eq5d_valueset <- EQ5D3L_tariffs_TTO.df
336  }
337  2690X       } else {
338  2689X         if (method == "VAS" & country %in% VAS_countrylist) {
339  1X           eq5d_valueset <- EQ5D3L_tariffs_VAS.df
340  1X           stop("No tariff found")
341  }
342  }
343  7605695X     score_num <- as.numeric(paste(scores, collapse = ""))
344  7605695X     if (country == "Australia" & sum(score_num %in% australia.impalausibleordering.scores) > 0) {
345  23X       values_state <- .correctImplausibleOrdering(scores)
346  } else {
347  7605672X       names(scores) <- c("MO", "SC", "UA", "PD", "AD")
348  7605672X       rows <- paste0(names(scores), scores)
349  7605672X       col <- check_column_exist(country, eq5d_valueset)
350  7605672X       if (col == 0) {
351  7605672X         min2or3 <- which(scores %in% c(2, 3))
352  7605672X         if (length(min2or3) == 5) {
353  1001559X           all_equals2or3 <- 1
354  1X           } else {
355  6604113X             all_equals2or3 <- c()
356  1X           }
357  7605672X         which3 <- which(scores %in% c(3))
358  7605672X         which2 <- which(scores %in% c(2))
359  7605672X         rownums <- c()
360  7605672X         dim_response <- NA
361  7605672X         min3_value <- NA
362  7605672X         all_equals2or3_value <- NA
363  7605672X         min2or3_value <- NA
364  7605672X         c3sq_value <- NA
365  7605672X         d1_value <- NA
366  7605672X         i2_value <- NA
367  7605672X         i2_sq_value <- NA
368  7605672X         i3_value <- NA
369  7605672X         i3_sq_value <- NA
370  7605672X         onlyisand2s_value <- NA
371  7605672X         onlyisand3s_value <- NA
372  7605672X         atleast2andatleast3_value <- NA
373  7605672X         nos2withatleast3_value <- NA
374  7605672X         nos2sq_value <- NA
375  7605672X         nos3sq_value <- NA
376  7605672X         mo3sc3_value <- NA
377  7605672X         mo3ua3_value <- NA
378  7605672X         mo3pd3_value <- NA
379  7605672X         mo3ad3_value <- NA
380  7605672X         sc3ua3_value <- NA
381  7605672X         sc3pd3_value <- NA
382  7605672X         sc3ad3_value <- NA
383  7605672X         ua3pd3_value <- NA
384  7605672X         ua3ad3_value <- NA

```

```

385 | 7605672x pd3ad3_value <- NA
386 | 7605672x mo2ua2_value <- NA
387 | 7605672x sc3ua2_value <- NA
388 | 7605672x rownumfh <- which(row.names(eq5d_valueset) == "FullHealth")
389 | 7605672x rownum_min2or3 <- which(row.names(eq5d_valueset) == "Constant")
390 | 7605672x rownum_min3 <- which(row.names(eq5d_valueset) == "N3")
391 | 7605672x rownum_only1sand2s <- which(row.names(eq5d_valueset) == "Only1sand2s")
392 | 7605672x rownum_only1sand3s <- which(row.names(eq5d_valueset) == "Only1sand3s")
393 | 7605672x rownum_atleast2andatleast3 <- which(row.names(eq5d_valueset) == "Atleast2andatleast3")
394 | 7605672x rownum_nos2withatleast3 <- which(row.names(eq5d_valueset) == "Nos2withatleast3")
395 | 7605672x rownum_nos2sq <- which(row.names(eq5d_valueset) == "Nos2sq")
396 | 7605672x rownum_nos3sq <- which(row.names(eq5d_valueset) == "Nos3sq")
397 | 7605672x if (method == "TTO") {
398 | 7602983x rownum_all_equals2or3 <- which(row.names(eq5d_valueset) == "X5")
399 | 7602983x rownum_C3sq <- which(row.names(eq5d_valueset) == "C3sq")
400 | 7602983x rownum_D1 <- which(row.names(eq5d_valueset) == "D1")
401 | 7602983x rownum_I2 <- which(row.names(eq5d_valueset) == "I2")
402 | 7602983x rownum_I2_sq <- which(row.names(eq5d_valueset) == "I2_sq")
403 | 7602983x rownum_I3 <- which(row.names(eq5d_valueset) == "I3")
404 | 7602983x rownum_I3_sq <- which(row.names(eq5d_valueset) == "I3_sq")
405 | 7602983x rownum_M03C3 <- which(row.names(eq5d_valueset) == "M03C3")
406 | 7602983x rownum_M03UA3 <- which(row.names(eq5d_valueset) == "M03UA3")
407 | 7602983x rownum_M03PD3 <- which(row.names(eq5d_valueset) == "M03PD3")
408 | 7602983x rownum_M03AD3 <- which(row.names(eq5d_valueset) == "M03AD3")
409 | 7602983x rownum_SC3UA3 <- which(row.names(eq5d_valueset) == "SC3UA3")
410 | 7602983x rownum_SC3PD3 <- which(row.names(eq5d_valueset) == "SC3PD3")
411 | 7602983x rownum_SC3AD3 <- which(row.names(eq5d_valueset) == "SC3AD3")
412 | 7602983x rownum_UA3PD3 <- which(row.names(eq5d_valueset) == "UA3PD3")
413 | 7602983x rownum_UA3AD3 <- which(row.names(eq5d_valueset) == "UA3AD3")
414 | 7602983x rownum_PD3AD3 <- which(row.names(eq5d_valueset) == "PD3AD3")
415 | 7602983x rownum_M02UA2 <- which(row.names(eq5d_valueset) == "M02UA2")
416 | 7602983x rownum_SC3UA2 <- which(row.names(eq5d_valueset) == "SC3UA2")
417 } else {
418 | 2689x rownum_all_equals2or3 <- NA
419 | 2689x rownum_C3sq <- NA
420 | 2689x rownum_D1 <- NA
421 | 2689x rownum_I2 <- NA
422 | 2689x rownum_I2_sq <- NA
423 | 2689x rownum_I3 <- NA
424 | 2689x rownum_I3_sq <- NA
425 | 2689x rownum_M03C3 <- NA
426 | 2689x rownum_M03UA3 <- NA
427 | 2689x rownum_M03PD3 <- NA
428 | 2689x rownum_M03AD3 <- NA
429 | 2689x rownum_SC3UA3 <- NA
430 | 2689x rownum_SC3PD3 <- NA
431 | 2689x rownum_SC3AD3 <- NA
432 | 2689x rownum_UA3PD3 <- NA
433 | 2689x rownum_UA3AD3 <- NA
434 | 2689x rownum_PD3AD3 <- NA
435 | 2689x rownum_M02UA2 <- NA
436 | 2689x rownum_SC3UA2 <- NA
437 }
438 | 7605672x if (length(min2or3) > 0) {
439 | 7574371x for (i in seq_len(length(min2or3))) {
440 | 25352145x rownames <- row.names(eq5d_valueset)
441 | 25352145x ro <- which(rownams == rows[min2or3[i]])
442 | 25352145x rownums <- cbind(rownums, ro)
443 | 7574371x }
444 | 7574371x dim_response <- eq5d_valueset[rownums, country]
445 | 7605672x }
446 | 7605672x if (any(scores >= 3) && !is.na(eq5d_valueset[rownum_min3, country])) {
447 | 3962375x min3_value <- eq5d_valueset[rownum_min3, country]
448 | 7605672x }
449 | 7605672x if (length(which3) >= 1 & sum(is.na(rownum_C3sq)) == 0)) {
450 | 6601751x if (!is.na(eq5d_valueset[rownum_C3sq, country])) {
451 | 211x c3sq_value <- (length(which3))^2 * eq5d_valueset[rownum_C3sq, country]
452 | 7605672x }
453 | 7605672x if (length(all_equals2or3) >= 1 & sum(is.na(rownum_all_equals2or3) == 0)) {
454 | 1001206x if (!is.na(eq5d_valueset[rownum_all_equals2or3, country])) {
455 | 32x all_equals2or3_value <- eq5d_valueset[rownum_all_equals2or3, country]
456 | 7605672x }
457 | 7605672x if (sum(scores) > 5 & length(min2or3) >= 1 & sum(is.na(rownum_min2or3) == 0)) {
458 | 7574371x if (!is.na(eq5d_valueset[rownum_min2or3, country])) {
459 | 6606177x min2or3_value <- eq5d_valueset[rownum_min2or3, country]
460 | 7605672x }
461 | 7605672x if (sum(scores) > 5 & length(which2) >= 1 & sum(is.na(rownum_D1) == 0)) {
462 | 7571694x if (!is.na(eq5d_valueset[rownum_D1, country])) {
463 | 756976x d1_value <- (length(min2or3) - 1) * eq5d_valueset[rownum_D1, country]
464 | 7605672x }
465 | 7605672x if (sum(scores) > 5 & length(which2) >= 1 & sum(is.na(rownum_I2) == 0)) {
466 | 6601766x if (!is.na(eq5d_valueset[rownum_I2, country])) {
467 | 211x i2_value <- (length(which2) - 1) * eq5d_valueset[rownum_I2, country]
468 | 7605672x }
469 | 7605672x if (sum(scores) > 5 & length(which2) >= 1 & sum(is.na(rownum_I2_sq) == 0)) {
470 | 6601766x if (!is.na(eq5d_valueset[rownum_I2_sq, country])) {
471 | 659586x i2_sq_value <- (length(which2) - 1)^2 * eq5d_valueset[rownum_I2_sq, country]
472 | 7605672x }
473 | 7605672x if (sum(scores) > 5 & length(which3) >= 1 & sum(is.na(rownum_I3) == 0)) {
474 | 6601766x if (!is.na(eq5d_valueset[rownum_I3, country])) {
475 | 659797x i3_value <- (length(which3) - 1) * eq5d_valueset[rownum_I3, country]
476 | 7605672x }
477 | 7605672x

```

```

483 }
484 7605672X
485 6601751X
486 659797X
487 }
488 }
489 7605672X
490 970298X
491 62X
492 }
493 }
494 ## !all(scores==3) & need ?
495 7605672X
496 970270X
497 62X
498 }
499 }
500 7605672X
501 5633803X
502 360X
503 }
504 }
505 7605672X
506 5633803X
507 360X
508 }
509 }
510 7605672X
511 6604101X
512 422X
513 }
514 }
515 7605672X
516 6604073X
517 422X
518 }
519 }
520 7605672X
521 844770X
522 47X
523 }
524 }
525 7605672X
526 844773X
527 50X
528 }
529 }
530 7605672X
531 844768X
532 45X
533 }
534 }
535 7605672X
536 844770X
537 47X
538 }
539 }
540 7605672X
541 844770X
542 20X
543 }
544 }
545 7605672X
546 844763X
547 13X
548 }
549 }
550 7605672X
551 844767X
552 17X
553 }
554 }
555 7605672X
556 844773X
557 47X
558 }
559 }
560 7605672X
561 844772X
562 22X
563 }
564 }
565 7605672X
566 844761X
567 11X
568 }
569 }
570 7605672X
571 844775X
572 27X
573 }
574 }
575 7605672X
576 844772X
577 27X
578 }
579 }
580 7605672X
if (country == "Germany" && method == "VAS") {

```

```

581 | 243x prod.response <- prod(dim_response, na.rm = TRUE)
582 | 243x values <- c(
583 | 243x   eq5d_valueset$rownumfh, country], prod.response, min2or3_value, min3_value, all_equals2or3_value, c3sq_value, d1_value, i2_value,
584 | 243x   i2_sq_value, i3_value, i3_sq_value, only1sand2s_value, only1sand3s_value, atleast2andatleast3_value, nos2withatleast3_value,
585 | 243x   nos2Sq_value, nos3Sq_value
586 | )
587 | 243x values_state <- prod(values, na.rm = TRUE)
588 | } else {
589 | 7605429x sum_response <- sum(dim_response, na.rm = TRUE)
590 | 7605429x values <- c(
591 | 7605429x   eq5d_valueset$rownumfh, country], sum_response, min2or3_value, min3_value, all_equals2or3_value, c3sq_value, d1_value, i2_value,
592 | 7605429x   i2_sq_value, i3_value, i3_sq_value, only1sand2s_value, only1sand3s_value, atleast2andatleast3_value, nos2withatleast3_value,
593 | 7605429x   nos2Sq_value, nos3Sq_value, mo3sc3_value, mo3ua3_value, mo3pd3_value, mo3ad3_value, sc3ua3_value, sc3pd3_value, sc3ad3_value,
594 | 7605429x   ua3pd3_value, ua3ad3_value, pd3ad3_value, mo2ua2_value, sc3ua2_value
595 | )
596 | 7605429x values_state <- sum(values, na.rm = TRUE)
597 | }
598 | } else {
599 | !
600 |   stop("No country tariffs on valueset")
601 | }
602 | 7605695x return(values_state)
603 | }
604 | } else {
605 | 8x stop("No country tariffs found for the country you specified for EQ-5D-3L. Please try later")
606 | }
607 | }
608 ##### Function to value EQ-5D-3L columns to index values for any country and group
609 #' Function to value EQ-5D-3L columns to index values for any country and group
610 #' by gender and age
611 #' @param eq5dresponse_data the data containing eq5d responses
612 #' @param mo column name for EQ-5D-3L mobility
613 #' @param sc column name for response for EQ-5D-3L self care
614 #' @param ua column name for response for EQ-5D-3L usual activities
615 #' @param pd column name for response for EQ-5D-3L pain/discomfort
616 #' @param ad column name for response for EQ-5D-3L anxiety/depression
617 #' @param country country of interest, by default is UK, if groupby has to
618 #' specify the country should be specified
619 #' @param method Either "TTO" or "VAS"
620 #' @param groupby male or female -grouping by gender, default NULL
621 #' @param agelimit vector of ages to show upper and lower limits
622 #' @return the descriptive statistics of index values, frequency table and
623 #' the modified data where the last column will be the index values
624 #' @data-data.frame(age=c(10,20),sex=c("M","F"),mo=c(1,2),sc=c(1,2),ua=c(3,4),
625 #' pd=c(3,1),ad=c(3,1))
626 #' @value_3L(data, "mo", "sc", "ua", "pd", "ad", "UK", "TTO",NULL,c(10,70))
627 #' @export
628 #' @description Main function to value EQ-5D-3L descriptive system to 5L
629 #' index values.
630 value_3L <- function(eq5dresponse_data, mo, sc, ua, pd, ad, country, method,
631   groupby, agelimit) {
632   11x country <- replace_space_underscore(country)
633   10x eq5d_colnames <- c(mo, sc, ua, pd, ad)
634   10x ans_eq5d_colnames <- sapply(eq5d_colnames, check_column_exist,
635     eq5dresponse_data)
636   10x if (all(ans_eq5d_colnames == 0)) { # if the eq5d column names match
637     working_data <- subset_gender_age_to_group(eq5dresponse_data,
638       groupby, agelimit)
639     9x if (nrow(working_data) < 1) {
640       1x stop("no entries with the given criteria - Please check
641         the contents or the criteria")
642     1x else {
643       8x scores <- c()
644       8x for (j in 1:nrow(working_data)) {
645         10x res1 <- working_data[j, mo]
646         10x res2 <- working_data[j, sc]
647         10x res3 <- working_data[j, ua]
648         10x res4 <- working_data[j, pd]
649         10x res5 <- working_data[j, ad]
650         10x this_score <- value_3L_Ind(country, method, res1, res2,
651           res3, res4, res5)
652         9x scores <- c(scores, this_score)
653       }
654     7x new_data <- cbind(working_data, scores)
655     7x colnames(new_data) <- c(colnames(working_data), "EQ-5D-3L scores")
656     7x scores_noNA <- scores[!is.na(scores)]
657     7x if (length(scores_noNA) >= 1) {
658       4x stats <- descriptive_stat_data_column(scores_noNA, "EQ-5D-3L")
659       4x freq_table <- get_frequency_table(scores_noNA)
660       4x first <- is.null(groupby) || toupper(groupby) == "NA" ||
661         is.na(groupby)
662       4x second <- is.null(agelimit) || sum(toupper(agelimit) == "NA") != 0 ||
663       4x sum(is.na(agelimit)) != 0
664       4x if (first & second) {
665         1x title <- paste("Histogram of EQ-5D-3L index values", sep = "")
666       } else {
667         3x if (first & !second) {
668           1x title <- paste("Histogram of EQ-5D-3L index values",
669             " with ages between ", agelimit[1], " and ", agelimit[2],
670             1x sep = ""
671           )
672         } else {
673           2x if (!first & second) {
674             1x title <- paste("Histogram of EQ-5D-3L index values for ",
675               groupby,
676               1x sep = ""
677             )
678           } else {

```

```

679 | 1x      title <- paste("Histogram of EQ-5D-3L index values for ",
680 | 1x          groupby, " with ages between ", agelimit[1], " and ", agelimit[2],
681 | 1x          sep = ""
682 |       )
683 |     }
684 |   }
685 | 4x   hist_plot <- graphics::hist(scores_noNA, main = title)
686 | 4x   results <- list("stats" = stats, "frequency_table" = freq_table,
687 | 4x           "histogram" = hist_plot, "modified_data" = new_data)
688 | 4x   return(results)
689 | } else {
690 | 3x   print("No relevant rows with non NA scores")
691 | }
692 |
693 | }
694 | } else {# if the eq 5d column names do not match
695 | 1x   stop("EQ-5D column names do not match")
696 | }
697 |
698 ##### Function to map EQ-5D-5L descriptive system to 3L index value
699 #' Function to map EQ-5D-5L descriptive system to 3L index value
700 #' @param country default is "UK"
701 #' @param method CW cross walk
702 #' @param dimen response for EQ-5D-5L mobility or the 5 digit response, or
703 #' the vector of responses, e.g. 11111, c(1,1,1,1,1) or 1
704 #' @param dimen2 response for EQ-5D-5L self care, or NA if the responses
705 #' given as dimen
706 #' @param dimen3 response for EQ-5D-5L usual activities,or NA if the responses
707 #' are given as dimen
708 #' @param dimen4 response for EQ-5D-5L pain/discomfort, or NA if the responses
709 #' are given as dimen
710 #' @param dimen5 response for EQ-5D-5L anxiety/depression, or NA if the
711 #' responses are given as dimen
712 #' @return index value of EQ-5D-3L, -1 if any error
713 #' @examples
714 #' map_5Lto3L_Ind("UK", "CW", 11125)
715 #' map_5Lto3L_Ind("UK", "CW", c(1, 1, 1, 2, 5))
716 #' map_5Lto3L_Ind("UK", "CW", 1, 1, 1, 2, 5)
717 #' @export
718 #' @description Function to map EQ-5D-5L descriptive system to 3L index value
719 #'(ref:Van Hout et al 2012 and code inspired from
720 #'https://github.com/brechtdv/eq5d-mapping)
721 map_5Lto3L_Ind <- function(country = "UK", method = "CW", dimen, dimen2 = NA,
722                           dimen3 = NA, dimen4 = NA, dimen5 = NA) {
723 31275x   country_list <- c("Denmark", "France", "Germany", "Japan", "Netherlands",
724 31275x     "Spain", "Thailand", "UK", "USA", "Zimbabwe")
725 31275x   country <- replace_space_underscore(country)
726 31275x   if (country %in% country_list) {
727 31273x     responses <- c(dimen, dimen2, dimen3, dimen4, dimen5)
728 31273x     if (sum(is.na(dimen)) > 0) {
729 2x       # first value should be not be a NA, do not contain NA
730 2x       this_score_5L <- NA
731 2x       values_state <- NA
732 2x       return(values_state)
733   } else {
734     # check first value should be a vector containing responses or a
735     # digit number
736 31271x     if (length(dimen) != 5 && length(dimen) != 1) {
737 1x       stop("Expecting the full response as5 digit number or just
738 1x         the response for mobility")
739   } else {# first value a vector or a 5 digit number
740 31270x     if (length(dimen) == 5) {# first value a vector
741 12x       if (any(dimen < 1) || any(dimen > 5)) {
742 4x         stop("Invalid EQ-5D-5L responses-check the responses to each question")
743   } else {
744     8x       this_score_5L <- as.numeric(paste(dimen, collapse = ""))
745   } else {# first value 5 digit number or actual response for mobility
746 31258x     if (length(dimen) == 1) {
747 31258x       if (dimen >= 11111 & dimen <= 55555) {# valid 5 digit number
748 31251x         this_score_5L <- dimen
749   } else {
750     7x       if (dimen <= 5 && dimen > 0) {# valid response to mobility
751 6x         four_res <- c(dimen2, dimen3, dimen4, dimen5)
752 6x         if (sum(is.na(four_res)) == 0) {
753 5x           if (all(responses <= 5) && all(responses > 0)) {
754 2x             this_score_5L <- paste(responses, collapse = "")
755           # all valid and generate the score
756         } else {# error values
757 3x           stop("Invalid EQ-5D-5L responses-check the responses
758 3x             to each question")
759         }
760       } else {
761         # missing values
762 1x         this_score_5L <- NA
763 1x         values_state <- NA
764 1x         return(values_state)
765       }
766     } else {
767 1x       stop("Invalid EQ-5D-5L response to mobility")
768     }
769   }
770 }
771 }
772 }
773 }
774
775 ## create a vector of all possible 3L index values (Length == 3^5)
776 31261x index_3L <- numeric(243)

```

```

777 |      ## create a dataframe of all possible 3L scores
778 | 31261x  scores_3L <- expand.grid(
779 | 31261x    AD = seq(3),
780 | 31261x    PD = seq(3),
781 | 31261x    UA = seq(3),
782 | 31261x    SC = seq(3),
783 | 31261x    MO = seq(3)
784 | 31261x )
785 |
786 |      ## calculate the index value for each score
787 |      ## using function EQSD_be based on Cleemput et al, 2010
788 | 31261x for (i in seq(243)) {
789 | 7596423x   index_3L[i] <-
790 | 7596423x     value_3L_Ind(
791 | 7596423x       country, "TTO", scores_3L[i, "MO"],
792 | 7596423x       scores_3L[i, "SC"],
793 | 7596423x       scores_3L[i, "UA"],
794 | 7596423x       scores_3L[i, "PD"],
795 | 7596423x       scores_3L[i, "AD"]
796 | 31261x   )
797 |
798 |      ## create a dataframe of all possible 5L scores
799 | 31261x  scores_5L <- expand.grid(
800 | 31261x    AD = seq(5),
801 | 31261x    PD = seq(5),
802 | 31261x    UA = seq(5),
803 | 31261x    SC = seq(5),
804 | 31261x    MO = seq(5)
805 | 31261x )
806 |
807 |      ## 5L to 3L CROSSWALK
808 |      ## Load 'probability matrix' from 'EQ-SD-5L_Crosswalk_Value_Sets'
809 |      ## this is saved as dataframe 'm'
810 | 31261x if (toupper(method) == "CW") {
811 | 31259x   prob.matrix <- Probability_matrix_crosswalk.df
812 | 31259x   m <- prob.matrix
813 | 31259x   rows_m <- nrow(m)
814 | 31259x   cols_m <- ncol(m)
815 | 31259x   if (rows_m != 3125 || cols_m != 243) {
816 | 31259x     ! stop("Error in number of cols or rows of probability matrix")
817 | 31259x   }
818 | 31259x   ## multiply each row of 't(m)' with 'index_3L'
819 | 31259x   m_prod <- t(t(m)) * index_3L
820 | 31259x   ## obtain sum per row
821 | 31259x   ## crosswalking index value for each 5L score
822 | 31259x   m_sums <- rowSums(m_prod)
823 | 31259x   ## reorder columns and convert to matrix
824 | 31259x   scores_5L <- with(scores_5L, cbind(MO, SC, UA, PD, AD))
825 | 31259x   ## create 5L score Labels
826 | 31259x   scores_5L_chr <- apply(scores_5L, 1, paste, collapse = "")
827 | 31259x   this_score <- which(scores_5L_chr == paste(this_score_5L,
828 | 31259x           collapse = ""))
829 | 31259x   if (country == "Zimbabwe" & this_score_5L == "11111") {
830 | 31259x     return(0.9)
831 | 31259x   } else {
832 | 31258x     return(m_sums[this_score])
833 | 31258x   }
834 | 31258x   } else {
835 | 2x     stop("The specified method is not implemented")
836 | 2x   }
837 | 2x   } else {
838 | 2x     stop("Crosswalk for the country specified is not implemented")
839 | 2x   }
840 | 2x }
841 ##### Function to map EQ-SD-5L scores to EQ-SD-3L index values as per the
842 #' specific country and group by gender and age
843 #' @param eq5dresponse_data the data containing eq5d5L responses
844 #' @param mobility column name for EQ-SD-5L mobility
845 #' @param self_care column name for response for EQ-5D-5L self care
846 #' @param usual_activities column name for response for EQ-5D-5L usual
847 #' activities
848 #' @param pain_discomfort column name for response for EQ-5D-5L pain/discomfort
849 #' @param anxiety column name for response for EQ-5D-5L anxiety/depression
850 #' @param country country of interest, by default is UK, if groupby has to
851 #' specify the country should be specified
852 #' @param method CW cross walk
853 #' @param groupby male or female -grouping by gender, default NULL
854 #' @param agelimit vector of ages to show upper and lower limits
855 #' @return index value if success, negative values for failure
856 #' @examples
857 #' @examples
858 #' map_5Lto3L(data.frame(
859 #'   mo = c(1), sc = c(4), ua = c(4), pd = c(3),
860 #'   ad = c(3)
861 #' ), "mo", "sc", "ua", "pd", "ad")
862 #' @export
863 #' @description Function to map EQ-SD-5L scores to EQ-SD-3L index values
864 map_5Lto3L <- function(eq5dresponse_data, mobility, self_care, usual_activities,
865   pain_discomfort, anxiety, country = "UK", method = "CW",
866   groupby = NULL, agelimit = NULL) {
867   10x   country <- replace_space_underscore(country)
868   9x   eq5d_colnames <- c(mobility, self_care, usual_activities, pain_discomfort,
869   9x     anxiety)
870   9x   ans_eq5d_colnames <- sapply(eq5d_colnames, check_column_exist,
871   9x     eq5dresponse_data)
872   9x   if (all(ans_eq5d_colnames == 0)) { # if the eq5d column names match
873   8x     working_data <- subset_gender_age_to_group(eq5dresponse_data, groupby,
874   8x       agelimit)

```

```

875 | 8X scores <- c()
876 | 8X if (nrow(working_data) < 1) {
877 | 1X   stop("no entries with the given criteria - Please check the contents
878 |       or the criteria")
879 |
880 | 7X } else {
881 | 9X   for (j in 1:nrow(working_data)) {
882 | 9X     res1 <- working_data[j, mobility]
883 | 9X     res2 <- working_data[j, self_care]
884 | 9X     res3 <- working_data[j, usual_activities]
885 | 9X     res4 <- working_data[j, pain_discomfort]
886 | 9X     res5 <- working_data[j, anxiety]
887 | 9X     this_score <- map_5Lto3L_Ind(country, method, c(res1, res2, res3,
888 |           res4, res5))
889 | 7X   scores <- c(scores, this_score)
890 |
891 | 5X new_data <- cbind(working_data, scores)
892 | 5X colnames(new_data) <- c(colnames(working_data), "Mapped EQ-5D-3L scores")
893 | 5X scores_noNA <- scores[!is.na(scores)]
894 | 5X if (length(scores_noNA) >= 1) {
895 | 4X   stats <- descriptive_stat_data_column(scores_noNA, "EQ-5D-3L")
896 | 4X   freq_table <- get_frequency_table(scores_noNA)
897 | 4X   first <- is.null(groupby) || toupper(groupby) == "NA" || is.na(groupby)
898 | 4X   second <- is.null(agelimit) || sum(toupper(agelimit)) == "NA" != 0 ||
899 | 4X     sum(is.na(agelimit)) != 0
900 | 4X   if (first & second) {
901 | 1X     title <- paste("Histogram of EQ-5D-3L index values", sep = "")
902 | 1X   } else {
903 | 3X     if (first & !second) {
904 | 1X       title <- paste("Histogram of EQ-5D-3L index values",
905 |           " with ages between ", agelimit[1], " and ", agelimit[2],
906 |           sep = ""
907 |         )
908 |     } else {
909 | 2X       if (!first & second) {
910 | 1X         title <- paste("Histogram of EQ-5D-3L index values for ",
911 |             groupby,
912 |             sep = ""
913 |           )
914 |     } else {
915 | 1X       title <- paste("Histogram of EQ-5D-3L index values for ",
916 |           groupby, " with ages between ", agelimit[1], " and ",
917 |           agelimit[2], sep = ""
918 |         )
919 |     }
920 |   }
921 | }
922 | 4X hist_plot <- graphics::hist(scores, main = title)
923 | 4X results <- list("stats" = stats, "freq_table" = freq_table,
924 | 4X   "histogram" = hist_plot, "modified_data" = new_data)
925 | 4X return(results)
926 | } else {
927 | 1X   print("No relevant rows with non NA scores")
928 | }
929 |
930 | } else {# if the eq 5d column names do not match
931 | 1X   stop("EQ-5D column names do not match")
932 | }
933 |
934 ##########
935 #' Function to correct implausible ordering in Australian valueset for EQ-5D-3L
936 #' @param scores , EQ-5D-3L scores as a number
937 #' @return the value that read from the stored dataframe
938 #' @examples
939 #' .correctImplausibleOrdering(11121)
940 #' @export
941 #' @description Correcting the implausible ordering
942 .correctImplausibleOrdering <- function(scores) {
943   value <- 0
944   score_num <- as.numeric(paste(scores, collapse = ""))
945   australia_impalusibleordering_scores <- c(
946     33132, 12133, 13133, 22133, 23133, 32133, 33133, 12233, 13233,
947     22233, 23233, 32233, 33233,
948     33232, 33323, 13332, 13333, 23332, 23333, 32333, 33332, 33333
949   )
950   australia_impalusibleordering_values <- c(
951     -0.045, 0.154, 0.154, 0.086, 0.086, -0.083, -0.083, 0.101, 0.101, 0.033,
952     0.033, -0.136, -0.136, -0.098, -0.199, 0.020, 0.020, -0.048, -0.048, -0.206,
953     -0.217, -0.217
954   )
955   if (sum(score_num %in% australia_impalusibleordering_scores) > 0) {
956     index <- which(score_num == australia_impalusibleordering_scores)
957     value <- australia_impalusibleordering_values[index]
958   }
959   return(value)
960 }
##########

```