routing

Max Lang

10/27/2022

`find_route`

Function will find a route between several cities passed in a data frame.

Inputs

- A data frame with a "from" and a "to" column (can be named differently)

Output

A named vector of a possible connection of the cities

Code

```
find_route <- function(df) {
    assertDataFrame(df, types = "character", any.missing = FALSE,
                    ncols = 2, min.rows = 1)
    checks <- checkCustom(df)

    cities <- unique(c(df[, 1], df[, 2]))
    edge.mat <- matrix(0, nrow = length(cities), ncol = length(cities))
    colnames(edge.mat) <- cities
    rownames(edge.mat) <- cities

    for (city in cities) {
        cons <- df[df[, 1] == city | df[, 2] == city, , drop = FALSE]
        for (con in seq_len(nrow(cons))) {
            edge.mat[cons[con, 1], cons[con, 2]] <- 1
            edge.mat[cons[con, 2], cons[con, 1]] <- 1
        }
    }
    diag(edge.mat) <- 0
    # construct output string
    tab <- table(c(df[, 1], df[, 2]))
    cities.visited <- names(which.min(tab))[[1]]
    i <- 1
    not.all.visited <- TRUE
    while (not.all.visited) {
        possible.next.stop <- which(edge.mat[cities.visited[i], ] == 1)
        edge.mat[cities.visited[i], possible.next.stop[[1]]] <- 0
        edge.mat[possible.next.stop[[1]], cities.visited[[i]]] <- 0
        edge.mat[, cities.visited[i]] <- 0
```

```
        cities.visited <- c(cities.visited, colnames(edge.mat)[possible.next.stop[[1]]])
        if (length(cities.visited) == length(cities)) {
            not.all.visited <- FALSE
        }
        i <- i + 1
    }
    paste0(cities.visited, collapse = " -> ")
}

checkCustom <- function(df) {
    tab <- as.data.frame(table(c(df$from, df$to)))$Freq
    tab2 <- as.data.frame(table(tab))
    check <- c(FALSE, FALSE, FALSE)
    # hier einmal abchecken, ob der df genau zwei Einträge hat mit 1 und die restlichen mit 2
    if (all(tab %in% c(1, 2))) check[[1]] <- TRUE
    if (identical(as.numeric(tab2[1, 1]), 1) && identical(as.numeric(tab2[1, 2]), 2)) check[[2]] <- TRU
    if (nrow(df) > 1) {
        if (identical(as.numeric(tab2[2, 1]), 2)) check[[3]] <- TRUE
    } else check[[3]] <- TRUE

    message <- c("Check1: The dataframe must contain two cities exactly once and the other cities exactl
                 "Check2: The dataframe must contain two cities exactly once",
                 "Check3: The dataframe must contain all other cities exactly twice")
    if (all(check)) return(TRUE) else stop(message[!check])
}
```

Worked Example

```
df <- data.frame(from = c("London", "Amsterdam", "Birmingham", "Hamburg", "Munich"),
           to = c("Birmingham", "Hamburg", "Amsterdam", "Munich", "Paris"))
find_route(df)
```

```
## [1] "London -> Birmingham -> Amsterdam -> Hamburg -> Munich -> Paris"
```