basic-matrix-df-ops

@MaxMLang (Github)

chess_black_sum

This functions sums all numbers that would be on black fields on a chessboard, here represented as a matrix.

Inputs

- matrix: A matrix representing a chessboard

Output

- A numeric vector containing the number of black fields

Code

```
chess_black_sum <- function(matrix) {
  sum(matrix[(row(matrix) + col(matrix)) %% 2 == 1])
}
```

Worked example

```
matrix <- matrix(data= 1:9, nrow = 3)
matrix
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
chess_black_sum(matrix)
```

```
## [1] 20
```

which_mat_min

This function finds the indexed positon of the minimum value in the matrix.

Inputs

- matrix: A matrix

Output

- A numeric vector containing the minimum indices i,j

Code

```
which_mat_min <- function(matrix) {
  # your code
  wm <- which.min(matrix)
  as.numeric(c(row(matrix)[wm], col(matrix)[wm]))
}
```

Worked example

```
matrix <- matrix(sample(100, size = 25), ncol = 5)
matrix
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   60   84   66   23   13
## [2,]   77   27   41   35   80
## [3,]    3   61   79   81   30
## [4,]   85   37   51   45   82
## [5,]   69   97   43    1    7
```

```
which_mat_min(matrix)
```

```
## [1] 5 4
```

diag_df

This function returns the diagonal of a dataframe.

Inputs

- `df`: a data.frame with arbitrary rows and columns

Output

- unnamed `list` of the diagonal elements of `df`

Code

```
diag_df <- function(df) {
  # your code
  assertDataFrame(df)
  lapply(seq_len(min(nrow(df), ncol(df))), function(i) df[i, i])
}
```

Worked example

```
df <- data.frame(v1= sample(50, 10), v2= sample(50, 10), v3= sample(50, 10))
df
```

```
##    v1 v2 v3
## 1  14 31 38
## 2  26 35 23
## 3  43 29 31
## 4  19 48 19
## 5  29  4 22
```

```
## 6  23  1 37
## 7  31 13 44
## 8  17 39  4
## 9   9  5 35
## 10  4 43 39
```

```
diag_df(df)
```

```
## [[1]]
## [1] 14
##
## [[2]]
## [1] 35
##
## [[3]]
## [1] 31
```