Package ‘eyeRead’

October 13, 2022

Title Prepare/Analyse Eye Tracking Data for Reading

Version 0.0.4

Description Functions to prepare and analyse eye tracking data of reading exercises. The functions allow some basic data preparations and code fixations as first and second pass. First passes can be further devided into forward and reading. The package further allows for aggregating fixation times per AOI or per AOI and per type of pass (first forward, first rereading, second). These methods are based on Hyönä, Lorch, and Rinck (2003) <doi:10.1016/B978-044451020-4/50018-9>
and Hyönä, and Lorch (2004) <doi:10.1016/j.learninstruc.2004.01.001>. It is also possible to convert between metric length and visual degrees.

Depends R (>= 3.6)

URL `https://github.com/SanVerhavert/eyeRead`

BugReports `https://github.com/SanVerhavert/eyeRead/issues`

License GPL-3

Encoding UTF-8

LazyData true

Suggests testthat, dplyr

RoxygenNote 7.1.1

Imports tibble (>= 2.1.3), data.table (>= 1.12.8), tidyr (>= 1.0.0)

NeedsCompilation no

Author San Verhavert [aut, cre],
Tine van Daal [aut],
Leen Catrysse [aut]

Maintainer San Verhavert <`verhavertsan@gmail.com`>

Repository CRAN

Date/Publication 2020-08-07 13:00:02 UTC

R topics documented:

AOItransition *calculate transitions for eye-tracking data*

Description

function to calculate transitions between AOI's for eye-tracking data

Usage

```
AOItransitions(AOI)
```

Arguments

`AOI` character vector with one entry per fixation, containing AOI's upon which was fixated

Details

Indicates each time a participant's fixation moves from one AOI to r AOI whereby absence of transition is coded as 0 and transitions are coded as X-Y whereby

X = ID of AOI left

Y = ID of AOI entered

Author(s)

Tine van Daal [aut], <tine.vandaal@uantwerpen.be> San Verhavert [ctb], <san.verhavert@uantwerpen.be>

Examples

```
# The function accepts a single AOI column
  data(SimData )

  AOItransitions( SimData$AOI )
```

`codePasses` *Codes the Fixations as First Pass and Second Pass*

Description

The fixations are coded as first pass and second pass with or without rereading as discussed in Hyönä, Lorch, and Rinck (2003) and Hyönä, and Lorch (2004).

Usage

```
codePasses(
  data,
  AOI,
  rereading = FALSE,
  fpx = NULL,
  fpy = NULL,
  origin = c("topLeft", "bottomLeft", "center", "topRight", "bottomRight"),
  fix_size = 42,
  fix_min = 3
)
```

Arguments

`data` A data frame containing fixation information of an eye tracing experiment. Each row indicates a fixation.

`AOI` The name or number of the column in `data` containing the name of the area of interest (AOI) that was fixated. If `data` contains one column per AOI, also a vector is accepted (see Details).

`rereading` Logical. Indicating if the first pass fixations should be split according to forward and rereading (`TRUE`) or not (`FALSE` [Default])

`fpx` The name or number of the column containing the x coordinate of the fixation point. Required if `Rereading` is `TRUE`.

`fpy` The name or number of the column containing the y coordinate of the fixation point. Required if `Rereading` is `TRUE`.

`origin` Character string specifying where the origin of the fixation coordinates `fpx` and `fpy` is located. The following values are possible: "topLeft" (default), "bottomLeft", "center", "topRight", "bottomRight".

`fix_size` The size or acuity of the saccade. (default = 42; see Details)

`fix_min` [optional] minimal number of fixations for first pass. (default = 3; see Details)

Details

This function takes a data frame containing information of an eye tracking reading exercise Each row indicates a fixation and the columns at least indicate the AOI that was fixated or if a specific

AOI was fixated or not. Optionally it can indicate the x and y coordinates of the fixation point. The fixations in this dataset are then coded according to their pass status: first pass and second pass.

If `AOI` is a single value it indicates the name or the number of the column with the name of the respective area of interest (AOI) that was fixated. In this case, the AOI column consists of names or numbers identifying the AOI's. If `AOI` is a vector it indicates the name or number of the AOI columns. In this case the AOI columns indicate if the respective fixation was in the corresponding AOI (1) or not (0). NOTE: if the names of the AOI columns passed to the function are just numbers, make sure to pass these as character for the function to work properly. The function does not check for this case.

First pass fixations are further divided into forward and rereading fixations if `rereading` is set to `TRUE`. In this case the names of the columns containing the x and y coordinates of the fixation point should be supplied by `fpx` and `fpy` respectively. The unit of these coordinates does not matter as long as it is the same for both and for the value of `fix_size`.

It is important to set the minimal distance between fixations (or the visual acuity) via `fix_size`. This value is used to determine if two fixations are on the same line and/or on the same position in the line. When this value is to small it is possible that some first-pass fixations are falsely categorized as rereading fixations. Specifically `fix_size` determines what the minimal distance between fixations should be in order for fixations to be considered in a different position on the line or on a different line. The default value is specified in pixels(px). The value is the number of pixels equivalent to 2 visual degrees, taken a screen of 1020px and about 54cm in width and a viewing distance of 60cm. (see `px2deg` for conversions). The value of 2 visual degrees is the average visual angle of the fovea (Llewellyn-Thomas, 1968; Haber & Hershenson, 1973). It is recommended to play around with the `fix_size` value for every participant.

By default this function considers the first three (3) fixations in any AOI as first pass fixations. And it does this regardless of whether the fixations are consecutive or interrupted by fixations in a different AOI. The minimal number of fixations considered as first pass can be changed through `fix_min`.

Value

The function returns a character vector of the same length as the number of rows in data. Depending on the respective settings it contains the following values with their respective meanings.

- `rereading` is `FALSE`:
  - `FP_\#` First Pass
  - `SP_\#` Second Pass
- `rereading` is `TRUE`
  - `FPF_\#` First Pass Forward
  - `FPR_\#` First Pass Rereading
  - `SP_\#` Second Pass

Where \# stands for the name of the respecitve AOI.

References

Haber, R. N., & Hershenson, M. (1973) *The psychology of visual perception.* New York: Holt, Rinehart, and Winston.

Llewellyn-Thomas, E. (1968) Movements of the eye. *Scientific American, 219*(2), 88-95.

Hyönä, J., Lorch, R. F., & Rinck, M. (2003). Eye movement measures to study global text processing. In J. Hyönä, R. Radach, & H. Deubel (Eds.), *The mind's eye: cognitive and applied aspects of eye movement research* (pp. 313-334). Amsterdam: Elsevier Science.

Hyönä, J., & Lorch, R. F. (2004). Effects of topic headings on text processing: evidence from adult readers' eye fixation patterns. *Learning and Instruction, 14*, 131-152. doi:10.1016/j.learninstruc.2004.01.001

Examples

```
data( "SimData" )

### codePasses calculates first and second  passes
## if a single AOI column is provided
# by name
codePasses( data = SimData, AOI = "AOI" )

# by column number
codePasses( data = SimData, AOI = 5 )

## and if multiple AOI columns are provided
# by name
resultA <- codePasses( data = SimData,
                       AOI = c( "AOI1", "AOI2", "AOI3" ) )
resultA

# by number
codePasses( data = SimData, AOI = 2:4 )

## \code{fix_min} influences how many fixations are needed in an AOI
## independent of any fixations in between.
resultB <- codePasses( data = SimData,
                       AOI = c( "AOI1", "AOI2", "AOI3" ), fix_min = 1 )

data.frame( fix_min3 = resultA, fix_min1 = resultB )

rm( resultA, resultB )

### it also calculates forward and backward first passes if the x and y
### coordinates of the fixations are provided and \code{rereading} is \code{TRUE}
resultA <- codePasses( data = SimData, AOI = "AOI",
                       rereading = TRUE, fpx = "xcoord", fpy = "ycoord",
                       fix_size = 20 )
resultA

# and allows for different coordinate origins
resultB <- codePasses( data = SimData, AOI = "AOI", rereading = TRUE,
                       fpx = "xcoord", fpy = "ycoord", origin = "bottomLeft",
                       fix_size = 20 )
 data.frame( topLeft = resultA, bottomLeft = resultB )

## mind that fix_size can influence the results
resultB <- codePasses( data = SimData, AOI = "AOI",
                       rereading = TRUE, fpx = "xcoord", fpy = "ycoord",
                       fix_size = 10 )

data.frame( fix_size20 = resultA, fix_size10 = resultB )
```

compileAOI *compile AOI columns in one column*

Description

Compiles the information on AOI's in separate variables to one variable

Usage

```
compileAOI(data, AOI, labels = NULL)
```

Arguments

`data` A data frame containing fixation information of an eye tracking experiment. Each row indicates a fixation.

`AOI` A vector containing the name or number of the columns in `data` indicating if the respective AOI was fixated (1) or not (0).

`labels` [optional] A vector containing the names of the AOI in the same order as the column names or -numbers provided to `AOI`

Details

This function can be used to convert a wide format eye tracking data frame to a long format eye tracking data frame. It takes a data frame with multiple binary variables that indicate whether an AOI is fixated on (=1) or not (=0) and returns a vector with the AOI's that were fixated on.

NOTE: if the names of the AOI columns passed to the function are just numbers, make sure to pass these as character for the function to work properly. The function does not check for this case.

Value

A factor that contains the AOI fixated on for each row in the data frame whereby absence of a fixation on AOI is coded as 0.

Author(s)

Tine van Daal [aut], <tine.vandaal@uantwerpen.be>

San Verhavert [ctb], <san.verhavert@uantwerpen.be>

Examples

```
data( SimData )

# compileAOI accepts AOI's as names
compileAOI( data = SimData, AOI = c( "AOI1", "AOI2", "AOI3" ) )

# and as column numbers
compileAOI( data = SimData, AOI = 2:4 )

# and it returns 0 if some fixations are oitside the provided AOI's
compileAOI( data = SimData, AOI = c( "AOI1", "AOI2" ) )

# it is also possible to suply different lables for the AOI's
compileAOI( data = SimData, AOI = c( "AOI1", "AOI2", "AOI3" ),
            labels = c( "1", "2", "3" ) )
```

convert *convert between units in eye-tracking*

Description

The functions convert between visual degrees and centimeters, inches or pixels.

Usage

```
size2deg(x, dist)

px2deg(x, dist, res, screenW)

deg2size(x, dist)

deg2px(x, dist, res, screenW)
```

Arguments

x	The value to convert. This can be a single number or a numerical vector.
dist	The distance to the screen. See Details for the measurement unit.
res	A vector indicating the screen resolution in the horizontal direction (in pixels).
screenW	The width of the screen in the horizontal direction. See Details for the measurement unit.

Details

When converting from size to degrees (size2deg), the measurement unit of the distance to the screen (dist) should be consistent with the measurement unit of the size you are converting from (x. Namely if the size is in centimeters, then the distance should be to, and if the size is in inches, then the distance should be in inches.

When converting from pixels to degrees (px2deg), the measurement unit of the distance to the screen (dist) should be consistent with the measurement unit of the screen width (screen width). Namely if the size is in centimeters, then the screen width should be to, and if the size is in inches, then the screen width should be in inches.

Value

Returns a value or vector of values indicating the corresponding visual degrees.

Functions

- size2deg: converts from centimeters to visual degrees
- px2deg: converts from pixels to visual degrees
- deg2size: converts from visual degrees to centimeters
- deg2px: converts from visual degrees to pixels

Examples

```
### these functions convert between units
  ## from size to degrees
  # for single values
  size2deg( x = 2, dist = 30 ) # 3.818304866

  # and multiple values
  size2deg( x = c( 2, 0.5, 7, 2, 20, 0.5 ),
            dist = c( 30, 30, 60, 15, 30, 15 ) )
   # 3.818304866, 0.954907555, 6.676941008, 7.628149669, 36.86989765,
   # 1.909682508

## from pixels to degrees
  # for single values
  px2deg( x = 2, dist = 30, res = 1024, screenW = 32 ) # 0.119366164

  # and multiple values
  px2deg( x = c( 2, 8, 100 ), dist = 30, res = 1024, screenW = 32 )
    # 0.119366164, 0.477462066, 5.96292244

## from degrees to size
  # for single values
  deg2size( x = 2, dist = 30 ) # 1.047303896

  # and multiple values
  deg2size( x = c( 2, 8, 100 ), dist = 30 )
    # 1.047303896, 4.195608717, 71.50521556
```

```
## from degrees to pixels
  # for single values
  deg2px( x = 0.119366164, dist = 30, res = 1024, screenW = 32 ) # 2

  # and multiple values
   deg2px( x = c( 0.119366164, 0.477462066, 5.96292244 ), dist = 30,
           res = 1024, screenW = 32 )
     # 2, 8, 100
```

eyeRead *eyeRead: A Package to Prepare/Analyse Eye Tracking Data for Reading*

Description

This package contains some functions to prepare and analyse eye tracking data of reading exercises. Essentially it identifies first pass and second pass fixations and their respective total durations.

eyeRead functions

- compileAOI Compiles the information on AOI's in separate variables to one variable
- AOItransitions Calculates transitions between AOI's
- codePasses Codes fixations as first pass and second pass with or without rereading
- convert Converts between visual degrees and centimeters, inches or pixels
- fixDur Calculates the fixation durations for the passes or the AOI's

fixDur *Fixation Duration*

Description

Calculates the fixation durations for the passes (Hyönä, Lorch, and Rinck, 2003; Hyönä, and Lorch, 2004) or the AOI's

Usage

```
fixDur(data, fixTime, passes, AOI = NULL)
```

Arguments

`data` A data frame containing fixation information of an eye tracing experiment and the coded passes. Each row indicates a fixation.

`fixTime` The name or number of the column containing the time per fixation.

`passes` The name or number of the column containing the coded passes.

`AOI` The name or number of the column in `data` containing the name of the area of interest (AOI) that was fixated.

Details

This function is a wrapper for aggregate

The function will only return the fixation duration of the values in the passes column. The passes column is the vector returned by the link{codePasses} function. It is also possible to provide the column name of the column containing the AOI's if you require the fixation durations for the AOI's only. The column of which the name or number is passed to passes,will be converted to a factor if it is not yet the case.

You can provide the names of the AOI's to AOI_label and indicate if rereading passes were coded in rereading. This ensures that the output contains all types of passes for each AOI, even if they did not occur. In that case the value in the output will be 0.

Value

A data frame with the following columns If the AOI column is provided to passes: $AOI: containing the AOI names $duration: containing the aggregated durations

If the passes column is provided to passes and there is no rereading: $AOI: containing the AOI names $FirstPass: containing the aggregated first pass durations $SecondPass: containing the aggregated second pass durations

If the passes column is provided to passes and there is rereading: $AOI: containing the AOI names $FirstPassForward: containing the aggregated first pass forward durations $FirstPassRereading: containing the aggregated first pass rereading durations $SecondPass: containing the aggregated second pass durations

The result will be in the same unit as the duration input.

If the data contains fixations that were outside the AOI, the first line of the results will contain the total fixation duration outside the AOI's.

References

Hyönä, J., Lorch, R. F., & Rinck, M. (2003). Eye movement measures to study global text processing. In J. Hyönä, R. Radach, & H. Deubel (Eds.), *The mind's eye: cognitive and applied aspects of eye movement research* (pp. 313-334). Amsterdam: Elsevier Science.

Hyönä, J., & Lorch, R. F. (2004). Effects of topic headings on text processing: evidence from adult readers' eye fixation patterns. *Learning and Instruction, 14*, 131-152. doi:10.1016/j.learninstruc.2004.01.001

See Also

aggregate, by, tapply

Examples

```
data( SimData )

### This function compiles fixation durations
## for first and second passes
# when the column name is given
fixDur( data = SimData, fixTime = "fixTime",
        passes = "passes" )
```

```
# and when column number is given
fixDur( data = SimData, fixTime = 8, passes = 9 )

## for forward and rereading passes
fixDur( data = SimData, fixTime = ”fixTime”,
        passes = ”passesReread” )

## and for AOI's
fixDur( data = SimData, fixTime = ”fixTime”,
        passes = ”AOI” )
```

`SimData` *A simulated dataset with eye tracking data*

Description

This data set contains simulated eye tracking data for demonstration purposes. It has the correct data structure for use with the package eyeRead. Mind that it also contains expected results from functions.

Usage

```
SimData
```

Format

A data set with 37 rows and 10 variables

fixationIndex a unique numerical index for each fixation

AOI1 1 if AOI1 was fixated at that time and 0 otherwise

AOI2 1 if AOI2 was fixated at that time and 0 otherwise

AOI3 1 if AOI3 was fixated at that time and 0 otherwise

AOI the name of the AOI that was fixated at that time; expected outcome of codePasses

xcoord the x coordinate of the fixation position; origin top left

ycoord the y coordinate of the fixation position; origin top left

fixTime the time duration that the AOI was fixated

passes the result of codePasses with `fix_min = 3`

passesReread the result of codePasses with `fix_min = 3` and `rereading = TRUE`

Index