Package 'mnreadR'

October 11, 2017

Type Package

Title MNREAD Parameters Estimation and Curve Plotting

Version 1.2.0

Description Allows to analyze the reading data obtained with the MNREAD Acuity Chart, a continuous-text reading acuity chart for normal and low vision. Provides the necessary functions to plot the MNREAD curve and estimate automatically the four MNREAD parameters: Maximum Reading Speed, Critical Print Size, Reading Acuity and Reading Accessibility Index. Reference: Calabrese et al. 2016 <doi:10.1001/jamaophthalmol.2015.6097>.

Depends R (>= 2.10), dplyr, ggplot2

Imports stats

URL http://legge.psych.umn.edu/mnread-acuity-charts

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Aurélie Calabrèse [aut, cre],
J. Steve Mansfield [aut],
Gordon E. Legge [aut]

Maintainer Aurélie Calabrèse <acalabre@umn.edu>

Repository CRAN

Date/Publication 2017-10-11 03:27:24 UTC

R topics documented:

`accIndex` *Reading ACCessibility Index (ACC) calculation*

Description

This function calculates the Reading Accessibility Index, while applying suited rules for missing data.

Usage

```
accIndex(data, print_size, reading_time, errors, ... = NULL)
```

Arguments

`data`	The name of your dataframe
`print_size`	The variable that contains print size values for each sentence (print size uncorrected for viewing distance)
`reading_time`	The variable that contains the reading time for each sentence
`errors`	The variable that contains the number of errors for each sentence
`...`	Optional grouping arguments

Value

The function returns a new dataframe with a variable called "ACC" that contains the Reading Accessibility Index estimate.

Notes

The Reading ACCessibility Index (ACC) is a new measure representing an individual's access to text over the range of print sizes found in everyday life. Its calculation does not rely on curve fitting and gives a direct comparison with the performance of normally sighted individuals. The ACC calculation uses the print size values non corrected for non-standard viewing distance.

For more details on the Reading Accessibility Index, see http://archopht.jamanetwork.com/article.aspx?articleid=2487490

Warning

To ensure that missing data are handled properly and that ACC calculation is correct, data need to be entered along certain rules:

- For the smallest print size that is presented but not read, right before the test is stopped: **reading_time = NA, errors = 10**
- For all the small sentences that are not presented because the test was stopped before them: **reading_time = NA, errors = NA**
- If a sentence is presented, and read, but the time was not recorded by the experimenter: **reading_time = NA, errors = actual number of errors** (cf. s5-regular in low vision data sample)
- If a large sentence was skipped to save time but would have been read well: **reading_time = NA, errors = NA** (cf. s1-regular in normal vision data sample)
- If a large sentence was skipped to save time because the subject cannot read large print: **reading_time = NA, errors = 10** (cf. s7 in low vision data sample)

See Also

mnreadParam for all MNREAD parameters estimation

curveParam_RT for MRS and CPS estimation using values of reading time (instead of reading speed)

curveParam_RS for MRS and CPS estimation using values of reading speed (instead of reading time)

readingAcuity for Reading Acuity calculation

Examples

```
# inspect the structure of the dataframe
head(data_low_vision, 10)

#------

# restrict dataset to one MNREAD test only (subject s1, regular polarity)
data_s1 <- data_low_vision %>%
   filter (subject == "s1", polarity == "regular")

# run the reading accessibility index calculation
data_low_vision_ACC <- accIndex(data_s1, ps, rt, err)

# inspect the newly created dataframe
data_low_vision_ACC

#------

# run the reading accessibility index calculation
# on the whole dataset grouped by subject and polarity
data_low_vision_ACC <- accIndex(data_low_vision, ps, rt, err,
                                subject, polarity)
```

```
# inspect the structure of the newly created dataframe
head(data_low_vision_ACC, 10)
```

`curveParam_RS` *Maximum Reading Speed (MRS) and Critical Print Size (CPS) estimation using reading speed values.*

Description

This function estimates simultaneously:

- Maximum Reading Speed (MRS)
- Critical Print Size (CPS)

while performing print size correction for non-standard testing viewing distance.

Usage

```
curveParam_RS(data, print_size, viewing_distance, reading_speed, ... = NULL)
```

Arguments

`data`	The name of your dataframe
`print_size`	The variable that contains print size values for each sentence (print size uncorrected for viewing distance)
`viewing_distance`	The variable that contains the viewing distance value used for testing
`reading_speed`	The variable that contains the reading speed for each sentence
`...`	Optional grouping arguments

Value

The function returns a new dataframe with two variables:

- "CPS" -> contains the Critical Print Size estimate (in logMAR)
- "MRS" -> contains the Maximum Reading Speed estimate (in words/min)

Notes

This function uses the original algorithm described in Legge (2007) to estimate Maximum Reading Speed (MRS) and Critical Print Size (CPS). This algorithm searches for a reading speed plateau in the data. A plateau is defined as a range of print sizes that supports reading speed at a significantly faster rate than the print sizes smaller or larger than the plateau range. Concretely, the plateau is determined as print sizes which reading speed is at least 1.96 SD faster than the other print sizes. The Maximum Reading Speed is estimated as the mean reading speed for print sizes included in the plateau. The Critical Print Size is defined as the smallest print size on the plateau.

For more details on the original algorithm, see Chapter 5 of this book:\ Legge, G.E. (2007). Psychophysics of Reading in Normal and Low Vision. Mahwah, NJ & London: Lawrence Erlbaum Associates. ISBN 0-8058-4328-0 https://books.google.fr/books/about/Psychophysics_of_Reading_in_Normal_and_L.html?id=BGTHS8zANiUC&redir_esc=y

To ensure proper estimation of the MRS and CPS, individual MNREAD curves should be plotted and inspected visually.

Warning

To run the function properly, one needs to make sure that the variables are of the class:

- **print_size** -> numeric
- **viewing_distance** -> integer
- **reading_speed** -> numeric

In cases where only 3 or less sentences were read during a test, the function won't be able to estimate the MRS and CPS and will return NA values instead. The ACC should be used to estimate the MNREAD score in such cases where there are not enough data points to fit the MNREAD curve.

See Also

curveParam_RT for MRS and CPS estimation using values of reading time (instead of reading speed)

mnreadParam for all MNREAD parameters estimation

readingAcuity for Reading Acuity calculation

accIndex for Reading Accessibility Index calculation

Examples

```
# inspect the structure of the dataframe
head(data_low_vision, 10)

#------

# restrict dataset to one MNREAD test only (subject s1, regular polarity)
data_s1 <- data_low_vision %>%
   filter (subject == "s1", polarity == "regular")

# run the parameters estimation
data_low_vision_MRS_CPS <- curveParam_RS(data_s1, ps, vd, rt, err)

# inspect the newly created dataframe
data_low_vision_MRS_CPS

#------

# run the parameters estimation on the whole dataset grouped by subject and polarity
data_low_vision_MRS_CPS <- curveParam_RS(data_low_vision, ps, vd, rt, err,
                                         subject, polarity)
```

```
# inspect the structure of the newly created dataframe
head(data_low_vision_MRS_CPS, 10)
```

`curveParam_RT` *Maximum Reading Speed (MRS) and Critical Print Size (CPS) estimation using raw data of reading time and number of errors.*

Description

This function estimates simultaneously:

- Maximum Reading Speed (MRS)
- Critical Print Size (CPS)

while performing print size correction for non-standard testing viewing distance.

Usage

```
curveParam_RT(data, print_size, viewing_distance, reading_time, errors,
  ... = NULL)
```

Arguments

`data`	The name of your dataframe
`print_size`	The variable that contains print size values for each sentence (print size uncorrected for viewing distance)
`viewing_distance`	The variable that contains the viewing distance value used for testing
`reading_time`	The variable that contains the reading time for each sentence
`errors`	The variable that contains the number of errors for each sentence
`...`	Optional grouping arguments

Value

The function returns a new dataframe with two variables:

- "CPS" -> contains the Critical Print Size estimate (in logMAR)
- "MRS" -> contains the Maximum Reading Speed estimate (in words/min)

Notes

This function uses the original algorithm described in Legge (2007) to estimate Maximum Reading Speed (MRS) and Critical Print Size (CPS). This algorithm searches for a reading speed plateau in the data. A plateau is defined as a range of print sizes that supports reading speed at a significantly faster rate than the print sizes smaller or larger than the plateau range. Concretely, the plateau is determined as print sizes which reading speed is at least 1.96 SD faster than the other print sizes. The Maximum Reading Speed is estimated as the mean reading speed for print sizes included in the plateau. The Critical Print Size is defined as the smallest print size on the plateau.

For more details on the original algorithm, see Chapter 5 of this book:\ Legge, G.E. (2007). Psychophysics of Reading in Normal and Low Vision. Mahwah, NJ & London: Lawrence Erlbaum Associates. ISBN 0-8058-4328-0 https://books.google.fr/books/about/Psychophysics_of_Reading_in_Normal_and_L.html?id=BGTHS8zANiUC&redir_esc=y

To ensure proper estimation of the MRS and CPS, individual MNREAD curves should be plotted and inspected visually.

Warning

For the function to run properly, one needs to make sure that the variables are of the class:

- **print_size** -> numeric
- **viewing_distance** -> integer
- **reading_time** -> numeric
- **errors** -> integer

In cases where only 3 or less sentences were read during a test, the function won't be able to estimate the MRS and CPS and will return NA values instead. The ACC should be used to estimate the MNREAD score in such cases where there are not enough data points to fit the MNREAD curve.

See Also

curveParam_RS for MRS and CPS estimation using values of reading speed (instead of reading time)

mnreadParam for all MNREAD parameters estimation

readingAcuity for Reading Acuity calculation

accIndex for Reading Accessibility Index calculation

Examples

```
# inspect the structure of the dataframe
head(data_low_vision, 10)

#------

# restrict dataset to one MNREAD test only (subject s1, regular polarity)
data_s1 <- data_low_vision %>%
   filter (subject == "s1", polarity == "regular")

# run the parameters estimation
```

```
data_low_vision_MRS_CPS <- curveParam_RT(data_s1, ps, vd, rt, err)

# inspect the newly created dataframe
data_low_vision_MRS_CPS

#------

# run the parameters estimation on the whole dataset grouped by subject and polarity
data_low_vision_MRS_CPS <- curveParam_RT(data_low_vision, ps, vd, rt, err,
                                         subject, polarity)

# inspect the structure of the newly created dataframe
head(data_low_vision_MRS_CPS, 10)
```

`data_low_vision` *MNREAD data collected in subjects with low vision.*

Description

A dataset containing raw MNREAD data for 12 subjects with low vision. Each subject was tested twice:

- once on the regular polarity of the test (black print on white background)
- once on the reverse polarity of the test (white print on black background)

Usage

```
data_low_vision
```

Format

A data frame with 437 rows and 6 variables, where each line stores data for one sentence:

subject subject ID code

polarity test polarity used (regular or reverse)

vd viewing distance in cm

ps print size in logMAR, as written on the chart (print size uncorrected for viewing distance)

rt reading time in seconds

err number of errors ...

Source

Data collected at the Minnesota Laboratory for Low-Vision Research (UMN)

data_normal_vision *MNREAD data collected in subjects with normal vision.*

Description

A dataset containing raw MNREAD data for 18 young adults with normal vision. Each subject was tested twice:

- once on the regular polarity of the test (black print on white background)
- once on the reverse polarity of the test (white print on black background)

Usage

```
data_normal_vision
```

Format

A data frame with 684 rows and 6 variables, where each line stores data for one sentence:

subject subject ID code

polarity test polarity used (regular or reverse)

vd viewing distance in cm

ps print size in logMAR, as written on the chart (print size uncorrected for viewing distance)

rt reading time in seconds

err number of errors ...

Source

Data collected at the Minnesota Laboratory for Low-Vision Research (UMN)

logMARcorrect *Print size correction for non-standard viewing distance*

Description

The logMAR scale allows simple conversion of print size between different viewing distances. When the MNREAD test is not run at the standard distance (ie. 40 cm - 16 inches), the angular print size (in logMAR) must be adjusted to compensate for the change in viewing distance. This function allows to correct the print size accordingly to the viewing distance used for testing.

Usage

```
logMARcorrect(data, print_size, viewing_distance)
```

Arguments

data	The name of your dataframe
print_size	The variable that contains print size values (print size uncorrected for viewing distance)
viewing_distance	The variable that contains the viewing distance value used for testing

Value

The function returns the original dataframe with an added variable called "correct_ps" that contains corrected print size values (in logMAR).

Examples

```
# inspect the strucutre of the dataframe
head(data_low_vision, 10)

# run the correction
data_low_vision_new <- logMARcorrect(data_low_vision, ps, vd)

# inspect the structure of the newly created dataframe
head(data_low_vision_new, 10)
```

---

mnreadCurve *MNREAD curve plotting*

Description

This function plots individual MNREAD curves, while showing the estimated MNREAD parameters:

- Maximum Reading Speed (MRS)
- Critical Print Size (CPS)
- Reading Acuity (RA)

Usage

```
mnreadCurve(data, print_size, viewing_distance, reading_time, errors,
  ... = NULL)
```

Arguments

`data`	The name of your dataframe
`print_size`	The variable that contains print size values for each sentence (print size uncorrected for viewing distance)
`viewing_distance`	The variable that contains the viewing distance value used for testing
`reading_time`	The variable that contains the reading time for each sentence
`errors`	The variable that contains the number of errors for each sentence
`...`	Optional grouping arguments

Value

The function returns a plot of reading speed (in words/min) as a function of print size (in logMAR). Reading Acuity is marked as a triangle, Maximum Reading Speed and Critical Print Size are shown with dashed lines. When using two grouping arguments, a colored diamond is added for clarification. Highlighted data points represent the range of print sizes included in the Reading Accessibility Index calculation.

Notes

This function can take no more that two grouping arguments. The first grouping argument is used to draw sub-plots (using facet_wrap from ggplot2). The second grouping argument is color-coded.

This function performs print size correction for non-standard testing viewing distance before plotting the curve.

This function uses the original algorithm described in Legge (2007) to estimate Maximum Reading Speed (MRS) and Critical Print Size (CPS). For more details on the parameters estimation, see `curveParam_RT`.

Warning

For the function to run properly, one needs to make sure that the variables are of the class:

- **print_size** -> numeric
- **viewing_distance** -> integer
- **reading_time** -> numeric
- **errors** -> integer

In cases where only 3 or less sentences were read during a test, MRS and CPS cannot be estimated and won't be displayed on the plot. In such cases, the Reading Accessibility Index (ACC) can be used to estimate the MNREAD score instead (cf. `accIndex`).

See Also

`curveParam_RT` for MRS and CPS estimation using values of reading time (instead of reading speed)

`readingAcuity` for Reading Acuity calculation

Examples

```
# inspect the structure of the dataframe
head(data_low_vision, 10)

#------

# restrict dataset to one MNREAD test only (subject s1, regular polarity)
data_s1_reg <- data_low_vision %>%
   filter (subject == "s1", polarity == "regular")

# plot the MNREAD curve
## Not run:  mnreadCurve(data_s1_reg, ps, vd, rt, err)

#------

# restrict dataset to one subject (s1) and plot the MNREAD curves using ONE GROUPING ARGUMENT
# (ie. polarity)
data_s1 <- data_low_vision %>%
   filter (subject == "s1")

# plot the MNREAD curve using ONE GROUPING ARGUMENT (ie. polarity)
 ## Not run:  mnreadCurve(data_s1, ps, vd, rt, err, polarity)

#------

# restrict dataset to two subject (s1 & s2) and plot the MNREAD curves using TWO GROUPING ARGUMENTS
# (ie. subject and polarity)
data_s2 <- data_low_vision %>%
   filter (subject == "s1" | subject == "s2")

 ## Not run:  mnreadCurve(data_s2, ps, vd, rt, err, subject, polarity)

#------

# Once created, the MNREAD curve can be customized as needed using ggplot2,
# for ex., by adding the number of errors for each sentence on top of the curve

# plot the MNREAD curve
my.plot <- mnreadCurve(data_s1, ps, vd, rt, err, polarity)

# displays my.plot
print(my.plot)

# calculates reading speed and perform print size correction
data_s1_new <- as.data.frame(
data_s1 %>%
    filter (err != "NA" & rt > 0) %>%
    mutate (errors10 = replace (err, err > 10, 10) ) %>%
    mutate (rs = 60 * (10 - errors10) / rt ) %>%
    mutate (correct_ps = ps + round(log10(40/(vd)), 2)) )

# add the number of errors for each sentence
```

```
my.new.plot <- my.plot + geom_text(aes(x = correct_ps, y = rs + 5, label = errors10),
                                   alpha = 0.5,
                                   data = data_s1_new %>% filter (errors10 != 0) )

# displays my.new.plot
print(my.new.plot)
```

`mnreadParam` *MNREAD parameters' estimation*

Description

This function calculates simultaneously all four MNREAD parameters:

- Maximum Reading Speed (MRS)
- Critical Print Size (CPS)
- Reading Acuity (RA)
- Reading ACCessibility Index (ACC)

while performing print size correction for non-standard testing viewing distance.

Usage

```
mnreadParam(data, print_size, viewing_distance, reading_time, errors,
  ... = NULL)
```

Arguments

`data`	The name of your dataframe
`print_size`	The variable that contains print size values for each sentence (print size uncorrected for viewing distance)
`viewing_distance`	The variable that contains the viewing distance value used for testing
`reading_time`	The variable that contains the reading time for each sentence
`errors`	The variable that contains the number of errors for each sentence
`...`	Optional grouping arguments

Value

The function returns a new dataframe with four variables:

- "RA" -> contains the Reading Acuity estimate (in logMAR)
- "CPS" -> contains the Critical Print Size estimate (in logMAR)
- "MRS" -> contains the Maximum Reading Speed estimate (in words/min)
- "ACC" -> contains the Reading Accessibility Index estimate

Notes

This function uses the original algorithm described in Legge (2007) to estimate Maximum Reading Speed (MRS) and Critical Print Size (CPS). This algorithm searches for a reading speed plateau in the data. A plateau is defined as a range of print sizes that supports reading speed at a significantly faster rate than the print sizes smaller or larger than the plateau range. Concretely, the plateau is determined as print sizes which reading speed is at least 1.96 SD faster than the other print sizes. The Maximum Reading Speed is estimated as the mean reading speed for print sizes included in the plateau. The Critical Print Size is defined as the smallest print size on the plateau.

For more details on the parameters estimation, see http://legge.psych.umn.edu/mnread-acuity-charts

For more details on the original algorithm, see Chapter 5 of this book:\ Legge, G.E. (2007). Psychophysics of Reading in Normal and Low Vision. Mahwah, NJ & London: Lawrence Erlbaum Associates. ISBN 0-8058-4328-0 https://books.google.fr/books/about/Psychophysics_of_Reading_in_Normal_and_L.html?id=BGTHS8zANiUC&redir_esc=y

To ensure proper estimation of the MRS and CPS, individual MNREAD curves should be plotted and inspected visually.

Warning

For the function to run properly, one needs to make sure that the variables are of the class:

- **print_size** -> numeric
- **viewing_distance** -> integer
- **reading_time** -> numeric
- **errors** -> integer

In cases where only 3 or less sentences were read during a test, the function won't be able to estimate the MRS and CPS and will return NA values instead. The ACC should be used to estimate the MNREAD score in such cases where there are not enough data points to fit the MNREAD curve.

To ensure proper ACC calculation, the data needs to be entered along certain rules:

- For the smallest print size that is presented but not read, right before the test is stopped: **reading_time = NA, errors = 10**
- For all the small sentences that are not presented because the test was stopped before them: **reading_time = NA, errors = NA**
- If a sentence is presented, and read, but the time was not recorded by the experimenter: **reading_time = NA, errors = actual number of errors** (cf. s5-regular in low vision data sample)
- If a large sentence was skipped to save time but would have been read well: **reading_time = NA, errors = NA** (cf. s1-regular in normal vision data sample)
- If a large sentence was skipped to save time because the subject cannot read large print: **reading_time = NA, errors = 10** (cf. s7 in low vision data sample)

See Also

curveParam_RT for MRS and CPS estimation using values of reading time (instead of reading speed)

curveParam_RS for MRS and CPS estimation using values of reading speed (instead of reading time)

readingAcuity for Reading Acuity calculation

accIndex for Reading Accessibility Index calculation

Examples

```
# inspect the structure of the dataframe
head(data_low_vision, 10)

#------

# restrict dataset to one MNREAD test only (subject s1, regular polarity)
data_s1 <- data_low_vision %>%
   filter (subject == "s1", polarity == "regular")

# run the parameters estimation
data_low_vision_param <- mnreadParam(data_s1, ps, vd, rt, err)

# inspect the newly created dataframe
data_low_vision_param

#------

# run the parameters estimation on the whole dataset grouped by subject and polarity
data_low_vision_param <- mnreadParam(data_low_vision, ps, vd, rt, err,
                                          subject, polarity)

# inspect the structure of the newly created dataframe
head(data_low_vision_param, 10)
```

---

mnreadR *mnreadR: An R package for analyzing MNREAD data*

Description

mnreadR provides simple functions to estimate the four MNREAD parameters:

- **Maximum Reading Speed** (MRS) -> can be estimated alone with curveParam_RT and curveParam_RS or simultaneously with the other MNREAD parameters with mnreadParam
- **Critical Print Size** (CPS) -> can be estimated alone with curveParam_RT and curveParam_RS or simultaneously with the other MNREAD parameters with mnreadParam
- **Reading Acuity** (RA) -> can be estimated alone with readingAcuity or simultaneously with the other MNREAD parameters with mnreadParam
- **Reading ACCessibility Index** (ACC) -> can be estimated alone with accIndex or simultaneously with the other MNREAD parameters with mnreadParam

`readingAcuity` *Reading Acuity (RA) calculation*

Description

Reading Acuity (RA) is defined as the smallest print size at which one can read without making significant errors. This function measures Reading Acuity to the nearest 0.1 logMAR, while performing print size correction for non-standard testing viewing distance.

Usage

```
readingAcuity(data, print_size, viewing_distance, reading_time, errors,
  ... = NULL)
```

Arguments

`data`	The name of your dataframe
`print_size`	The variable that contains print size values for each sentence (print size uncorrected for viewing distance)
`viewing_distance`	The variable that contains the viewing distance value used for testing
`reading_time`	The variable that contains the reading time for each sentence
`errors`	The variable that contains the number of errors for each sentence
`...`	Optional grouping arguments

Value

The function returns a new dataframe with a variable called "RA" that contains the Reading Acuity estimate (in logMAR).

See Also

`mnreadParam` for all MNREAD parameters estimation

`curveParam_RT` for MRS and CPS estimation using values of reading time (instead of reading speed)

`curveParam_RS` for MRS and CPS estimation using values of reading speed (instead of reading time)

`accIndex` for Reading Accessibility Index calculation

Examples

```
# inspect the structure of the dataframe
head(data_low_vision, 10)

#------

# restrict dataset to one MNREAD test only (subject s1, regular polarity)
data_s1 <- data_low_vision %>% filter (subject == "s1" & polarity == "regular")

# run the reading acuity calculation
data_low_vision_RA <- readingAcuity(data_s1, ps, vd, rt, err)

# inspect the newly created dataframe
data_low_vision_RA

#------

# run the reading acuity calculation on the whole dataset grouped by subject and polarity
data_low_vision_RA <- readingAcuity(data_low_vision, ps, vd, rt, err,
                                    subject, polarity)

# inspect the structure of the newly created dataframe
head(data_low_vision_RA, 10)
```

---

readingSpeed *Reading speed calculation corrected for the number of errors*

Description

This function calculates reading speed (in words per minute) for each sentence tested. This calculation takes into account the number of misread words and gives a more precise reading speed measurement than `readingSpeed_nonCorrected`.

Usage

```
readingSpeed(data, reading_time, errors)
```

Arguments

`data`	The name of your dataframe
`reading_time`	The variable that contains the reading time for each sentence
`errors`	The variable that contains the number of errors for each sentence

Value

The function returns the original dataframe with an added variable called "reading_speed" that contains reading speed (in words/min) for each sentence tested.

Notes

For general purposes, this method of reading speed calculation should be used preferentially over the less precise readingSpeed_nonCorrected.

See Also

readingSpeed_nonCorrected for reading speed non corrected for errors

Examples

```
# inspect the strucutre of the dataframe
head(data_low_vision, 10)

# run the reading speed calculation
data_low_vision_new <- readingSpeed(data_low_vision, rt, err)

# inspect the structure of the newly created dataframe
head(data_low_vision_new, 10)
```

---

readingSpeed_nonCorrected

Reading speed calculation not corrected for the number of errors

---

Description

This function calculates reading speed (in words per minute) using reading time (in seconds) only. This calculation provides a simplified value of reading speed, that does not take into account the number of misread words.

Usage

```
readingSpeed_nonCorrected(data, reading_time)
```

Arguments

data	The name of your dataframe
reading_time	The variable that contains the reading time for each sentence

Value

The function returns the original dataframe with an added variable called "reading_speed_nonCorrected" that contains reading speed (in words/min) for each sentence tested.

Notes

This function gives a less precise reading speed measurement than readingSpeed. Unless you know what you are doing, consider using readingSpeed instead of this function.

See Also

readingSpeed for reading speed corrected for errors

Examples

```
# inspect the strucutre of the dataframe
head(data_low_vision, 10)

# run the reading speed calculation
data_low_vision_new <- readingSpeed_nonCorrected(data_low_vision, rt)

# inspect the structure of the newly created dataframe
head(data_low_vision_new, 10)
```

Index