Package ‘JumpeR’

March 7, 2021

Title Importing and Working with Track and Field Data
Version 0.1.4
Description Primarily used to convert human readable track and field results into dataframes for further analysis. Results can come from central repositories like <https://www.flashresults.com/> or <http://www.deltatiming.com/>, or from individual team sites, like those for colleges. Also contains functions useful for working with track and field data.
License MIT + file LICENSE
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Imports magrittr, dplyr, purrr, pdftools, rvest, stringr, SwimmeR, xml2
Suggests testthat
NeedsCompilation no
Author Greg Pilgrim [aut, cre] (<https://orcid.org/0000-0001-7831-442X>), George Perry [ctb], Caitlin Baldwin [ctb]
Maintainer Greg Pilgrim <gpilgrim2670@gmail.com>
Repository CRAN
Date/Publication 2021-03-07 15:10:08 UTC

R topics documented:

add_row_numbers ................................................................. 2
attempts_remove ................................................................. 3
attempts_split ................................................................. 3
attempts_split_cols .......................................................... 4
attempts_split_long ......................................................... 5
attempts_split_long_helper ............................................. 5
collect Relay_athletes .................................................... 6
add_row_numbers

Description
Takes the output of read_results and adds row numbers to it

Usage
add_row_numbers(text)

Arguments
text output from read_results

Value
returns a dataframe with event names and row numbers to eventually be recombined with swimming results inside swim_parse

Author(s)
Greg Pilgrim <gpilgrim2670@gmail.com>
attempts_remove

See Also

add_row_numbers is a helper function inside swim_parse

attempts_remove Collects flight attempts within tf_parse

Description

Takes the output of read_results and, inside of tf_parse, extracts vertical jump attempts and associated row numbers

Usage

attempts_remove(df)

Arguments

df dataframe with jump attempt columns containing (X, O, PASS etc) and other columns

Value

returns a dataframe with the attempt columns removed

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

See Also

attempts_remove runs inside flash_parse

attempts_split Creates new columns of split attempts strings

Description

Given a dataframe with columns "Flight_1_Attempts" it will output three columns, for each of the attempts in flight 1 (Flight_1_Attempt_1, Flight_1_Attempt_2 etc.)

Usage

attempts_split(data_to_split)

Arguments

data_to_split output from read_results followed by add_row_numbers
attempts_split_cols

Value

returns a dataframe with Flight_X_Attempts columns split into individual attempts inside tf_parse

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

See Also

attempts_split is a helper function inside tf_parse

attempts_split_cols  Creates new columns for splitting attempts strings

Description

Given a dataframe with columns "Flight_1_Attempts" it will produce three columns, for each of the attempts in flight 1 (Flight_1_Attempt_1, Flight_1_Attempt_2 etc.)

Usage

attempts_split_cols(i, data, new_cols, old_cols)

Arguments

i  iterative value
data  output from tf_parse
new_cols  a list of new column names to make
old_cols  a list of old columns to split

Value

returns a dataframe with Flight_X_Attempts columns split into individual attempts inside tf_parse

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

See Also

attempts_split_cols is a helper function inside attempts_split
attempts_split_long

Description
Given a dataframe with columns "Flight_1_Attempts" it will three new rows, one for each of the attempts in flight 1

Usage
attempts_split_long(data_to_split)

Arguments
data_to_split output from read_results followed by add_row_numbers

Value
returns a dataframe with Flight_X_Attempts columns split into individual attempts as rows

Author(s)
Greg Pilgrim <gpilgrim2670@gmail.com>

Examples
df <- tf_parse(
  read_results("https://www.flashresults.com/2018_Meets/Outdoor/05-05_A10/015-1.pdf"),
  flights = TRUE,
  flight_attempts = TRUE,
  split_attempts = TRUE)
df %>%
  attempts_split_long()

attempts_split_long_helper

Description
Given a dataframe with columns "Flight_1_Attempts" it will produce three rows, for each of the attempts in flight 1
collect_relay_athletes

**Usage**

attempts_split_long_helper(i, data, old_cols)

**Arguments**

- `i`: output from `read_results` followed by `add_row_numbers`
- `data`: output from `tf_parse`
- `old_cols`: a list of old columns to split

**Value**

returns a dataframe with Flight_\text{X}_\text{Attempts} columns split into individual rows

**Author(s)**

Greg Pilgrim <gpilgrim2670@gmail.com>

**See Also**

- `attempts_split_long_helper` is a helper function inside `attempts_split_long`

---

**collect_relay_athletes**

*Collects relay athletes as a data frame within `tf_parse`*

**Description**

Collects relay athletes as a data frame within `tf_parse`

**Usage**

`collect_relay_athletes(x)`

**Arguments**

- `x`: output from `read_results` followed by `add_row_numbers`

**Value**

returns a dataframe of relay athletes and the associated performance row number

**Author(s)**

Greg Pilgrim <gpilgrim2670@gmail.com>

**See Also**

- `collect_relay_athletes_data` runs inside of `tf_parse`
event_parse

Pulls out event labels from text

Description
Locates event labels in text of results output from read_results and their associated row numbers. The resulting dataframe is joined back into results to include event names

Usage
event_parse(text)

Arguments
text output from read_results followed by add_row_numbers

Value
returns a dataframe with event names and row numbers to eventually be recombined with track and field results inside tf_parse

Author(s)
Greg Pilgrim <gpilgrim2670@gmail.com>

See Also
event_parse is a helper function inside tf_parse

fill_down

Fills NA values with previous non-NA value

Description
This is a base approximation of tidyr::fill()

Usage
fill_down(x)

Arguments
x a list having some number of non-NA values

Value
a list where NA values have been replaced with the closest previous non-NA value
Author(s)
Greg Pilgrim <gpilgrim2670@gmail.com>

See Also
fill_down is a helper function inside lines_sort

---

fill_left  

Shifts non-NA values to left in dataframe

Description
Moves non-NA data left into NA spaces, then removes all columns that contain only NA values

Usage
fill_left(df)

Arguments

df  
a dataframe having some NA values

Value

a dataframe where all values have been pushed left, replacing NAs, and all columns containing only NAs have been removed

Author(s)
Greg Pilgrim <gpilgrim2670@gmail.com>

See Also
fill_left is a helper function inside lines_sort
flash_flights_parse  

Collects attempts within tf_parse

---

**Description**

Takes the output of read_results and, inside of tf_parse, extracts jump/throw attempts and associated row numbers

**Usage**

flash_flights_parse(text)

**Arguments**

text  
output of read_results with row numbers appended by add_row_numbers

**Value**

returns a dataframe with split times and row numbers

**Author(s)**

Greg Pilgrim <gpilgrim2670@gmail.com>

**See Also**

flights_parse_flash runs inside flash_parse on the output of read_results with row numbers from add_row_numbers

---

flash_flight_attempts_parse  

Collects results of high jump & pole vault flight attempts within tf_parse

---

**Description**

Takes the output of read_results and, inside of tf_parse, extracts vertical jump flight attempts (XXO etc) and associated row numbers

**Usage**

flash_flight_attempts_parse(text)

**Arguments**

text  
output of read_results with row numbers appended by add_row_numbers
Value
returns a dataframe with split times and row numbers

Author(s)
Greg Pilgrim <gpilgrim2670@gmail.com>

See Also
flash_flight_attempts_parse runs inside flash_parse on the output of read_results with row numbers from add_row_numbers

Description
Outputs list of strings to be processed by tf_parse

Usage
flash_parse(
  flash_file,
  flash_flights = flights,
  flash_flight_attempts = flight_attempts,
  flash_split_attempts = split_attempts
)

Arguments
flash_file a .pdf or .html file (could be a url) where containing track and field results. Must be formatted in a "normal" fashion - see vignette
flash_flights should tf_parse try to include flights for jumping/throwing events? Defaults to FALSE
flash_flight_attempts should tf_parse try to include outcomes for flights for vertical jumping events? Defaults to FALSE
flash_split_attempts should flight_attempts columns be split into individual attempts

Value
a dataframe of track and field results

Author(s)
Greg Pilgrim <gpilgrim2670@gmail.com>
flights_parse

See Also

tf_parse is meant to be preceded by read_results

Description

Collects flights within tf_parse

Usage

flights_parse(text)

Arguments

text output of read_results with row numbers appended by add_row_numbers

Value

returns a dataframe with split times and row numbers

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

See Also

flights_parse runs inside tf_parse on the output of read_results with row numbers from add_row_numbers

flight_attempts_parse

Description

Collects results of high jump & pole vault attempts within tf_parse

Usage

flight_attempts_parse(text)

Arguments

text output of read_results with row numbers appended by add_row_numbers
Value

returns a dataframe with split times and row numbers

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

See Also

flight_attempts_parse runs inside tf_parse on the output of read_results with row numbers from add_row_numbers

Description

Used in testing links to external data, specifically inside of internal package tests. Attempts to connect to link for the length of duration (in s). If it fails it returns TRUE

Usage

is_link_broken(link_to_test, duration = 1)

Arguments

link_to_test a link
duration the lowest row number

Value

FALSE if the link works, TRUE if it fails

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>
lines_sort

**Sorts and collects lines by performance and row number**

**Description**

Collects all lines, (for example containing splits or relay swimmers) associated with a particular performance into a dataframe with the appropriate row number for that performance.

**Usage**

```r
lines_sort(x, min_row = minimum_row)
```

**Arguments**

- **x**
  - a list of character strings including performances, with row numbers added by `add_row_numbers`
- **min_row**
  - the lowest row number

**Value**

a dataframe with `Row_Numb` as the first column. Other columns are performance elements, like splits or relay swimmers, both in order of occurrence left to right.

**Author(s)**

Greg Pilgrim <gpilgrim2670@gmail.com>

**See Also**

- `lines_sort` is a helper function inside `splits_parse` and `swim_parse_ISL`

---

list_transform

**Transform list of lists into dataframe**

**Description**

Converts list of lists, with all sub-lists having the same number of elements into a dataframe where each sub-list is a row and each element a column.

**Usage**

```r
list_transform(x)
```

**Arguments**

- **x**
  - a list of lists, with all sub-lists having the same length
math_format

Value

a dataframe where each sub-list is a row and each element of that sub-list is a column

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

See Also

list_transform is a helper function used inside of tf_parse and event_parse

math_format                Formatting mm:ss.th times as seconds

Description

Takes a character string (or list) representing time in track format (e.g. 1:35.37) and converts it to a numeric value (95.37) or a list of values representing seconds.

Usage

math_format(x)

Arguments

x            A character vector of time(s) in track format (e.g. 1:35.93, as minutes:seconds.tenths hundreths) to be converted to seconds (95.93)

Value

returns the value of the string x which represents a time in track format (mm:ss.th) and converts it to seconds

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

Examples

math_format("1:35.93")
math_format("16:45.19")
math_format("25.43")
math_format(c("1:35.93", "16:45.19", NA, "25.43"))
**math_format_helper**  
*Helper function for formatting mm:ss.th times as seconds*

**Description**

Helper function for formatting mm:ss.th times as seconds

**Usage**

```r
desc_arg = math_format_helper(x)
```

**Arguments**

- `x`: A character vector of time(s) in track format (e.g. `1:35.93`) to be converted to seconds (95.93)

**Value**

A numeric value representing a time or distance. Units are not included

**Author(s)**

Greg Pilgrim <gpilgrim2670@gmail.com>

---

**metric_conversion**  
*Formatting feet-inches lengths as meters*

**Description**

Takes a character string (or list) representing a length in feet-inches format (e.g. "12-07.45") and converts it to a distance in meters ("3.85m")

**Usage**

```r
desc_arg = metric_conversion(x)
```

**Arguments**

- `x`: A character vector of distance(s) in feet-inches format (e.g. "12-07.45"), to be converted to meters ("3.85m")

**Value**

returns the value of the string x which represents a distance in meters, as a character, with unit "m" included
Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

Examples

distances <- c("1.23m", "5'-02.34", "43.45", "6.89", NA)
meter_conversion(distances)
math_format(meter_conversion(distances))
meter_conversion("5.45m")

metric_conversion_helper

Converts distances in feet-inches to meters

Description

Converts distances in feet-inches to meters

Usage

metric_conversion_helper(x)

Arguments

x A character vector of distance(s) to be converted from feet-inches to meters

Value

a numeric value representing a number of meters. Units are not included

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

read_results

Reads track and field results into a list of strings in preparation for parsing with tf_parse

Description

Outputs list of strings to be processed by tf_parse

Usage

read_results(file, node = "pre")
remove_unneeded_flights

Arguments

file
  a .pdf or .html file (could be a url) where containing swimming track and field results. pdfs with multiple columns will not work.

node
  a CSS node where html results are stored. Required for html results. Default is "pre", which nearly always works.

Value

returns a list of strings containing the information from file. Should then be parsed with tf_parse

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

See Also

read_results is meant to be followed by tf_parse

Examples

read_results("https://www.flashresults.com/2018_Meets/Outdoor/05-05_A10/015-1.pdf")

remove_unneeded_flights

Removes unneeded flights columns within tf_parse

Description

Inside of tf_parse & tf_parse, removes flight columns that do not have an associated flight_attempts column

Usage

remove_unneeded_flights(x)

Arguments

x
dataframe with columns called both "Flight_X" and "Flight_X_Results" where X is a number

Value

returns a dataframe where Flight_X columns that do not have a corresponding Flight_X_Results have been removed
standard_conversion

Description
Takes a character string (or list) representing a length in meters format (e.g. "3.85m") and converts it to a distance in feet-inches ("12-07.45")

Usage
standard_conversion(x)

Arguments
x A character vector of distance(s) in meters format ("3.85m") , to be converted to meters ("12-07.45")

Value
returns the value of the string x which represents a distance in feet-inches

Author(s)
Greg Pilgrim <gpilgrim2670@gmail.com>

standard_conversion_helper

Description
Converts distances in meters to feet-inches

Usage
standard_conversion_helper(x)
tf_parse

Arguments

x  A character vector of distance(s) to be converted from meters to feet-inches

Value

a character vector in feet-inches

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

Description

Outputs list of strings to be processed by tf_parse

Usage

tf_parse(file,
    avoid = avoid_default,
    typo = typo_default,
    replacement = replacement_default,
    relay_athletes = FALSE,
    flights = FALSE,
    flight_attempts = FALSE,
    split_attempts = FALSE)

Arguments

file  a .pdf or .html file (could be a url) where containing track and field results. Must be formatted in a "normal" fashion - see vignette

avoid  a list of strings. Rows in file containing these strings will not be included. For example "Record:," often used to label records, could be passed to avoid. The default is avoid_default, which contains many strings similar to "Record:". Users can supply their own lists to avoid.

typo  a list of strings that are typos in the original results. tf_parse is particularly sensitive to accidental double spaces, so "Central High School", with two spaces between "Central" and "High" is a problem, which can be fixed. Pass "Central High School" to typo.

replacement  a list of fixes for the strings in typo. Here one could pass "Central High School" (one space between "Central" and "High") to fix the issue described in typo
tf_parse

relay_athletes  should tf_parse try to include the names of relay athletes for relay events? Names will be listed in new columns "Relay-Athlete_1", "Relay_Athlete_2" etc. Defaults to FALSE.

flights  should tf_parse try to include flights for jumping/throwing events? Please note this will add a significant number of columns to the resulting dataframe. Defaults to FALSE.

flight_attempts  should tf_parse try to include flights results (i.e. "PASS", "X", "O") for high jump and pole value events? Please note this will add a significant number of columns to the resulting dataframe. Defaults to FALSE

split_attempts  should tf_parse split attempts from each flight into separate columns? For example "XXO" would result in three columns, one for "X", another for the second "X" and third for "O". There will be a lot of columns. Defaults to FALSE

Value

a dataframe of track and field results

Author(s)

Greg Pilgrim <gpilgrim2670@gmail.com>

See Also

tf_parse is meant to be preceded by read_results

Examples

tf_parse(  
read_results("https://www.flashresults.com/2018_Meets/Outdoor/05-05_A10/015-1.pdf"),  
flights = TRUE,  
flight_attempts = TRUE,  
split_attempts = TRUE)

Index

add_row_numbers, 2, 9–12
attempts_remove, 3
attempts_split, 3
attempts_split_cols, 4
attempts_split_long, 5
attempts_split_long_helper, 5

collect_relay_athletes, 6

event_parse, 7

fill_down, 7
fill_left, 8
flash_flight_attempts_parse, 9
flash_flights_parse, 9
flash_parse, 3, 9, 10, 10, 18
flight_attempts_parse, 11
flights_parse, 11

is_link_broken, 12

lines_sort, 13
list_transform, 13

math_format, 14
math_format_helper, 15
metric_conversion, 15
metric_conversion_helper, 16

read_results, 9–12, 16, 20
remove_unneeded_flights, 17

standard_conversion, 18
standard_conversion_helper, 18
swim_parse, 3

tf_parse, 4, 7, 11, 12, 17, 18, 19