Package 'retrosheet'

July 7, 2023

Type Package

Title Import Professional Baseball Data from 'Retrosheet'

Version 1.1.5

Date 2023-07-05

Maintainer Colin Douglas <colin@douglas.science>

Description A collection of tools to import and structure the (currently) single-season event, game-log, roster, and schedule data available from <https://www.retrosheet.org>. In particular, the event (a.k.a. play-by-play) files can be especially difficult to parse. This package does the parsing on those files, returning the requested data in the most practical R structure to use for sabermetric or other analyses.

URL https://github.com/colindouglas/retrosheet

Depends R (>= 2.10)

License GPL (>= 2)

Imports xml2 (>= 1.2.2), stringi (>= 0.4-1), httr (>= 1.4.1), stringr (>= 1.4.0), rvest (>= 0.3.5)

Note NOTICE regarding the transfer of data from Retrosheet: The information used here was obtained free of charge from and is copyrighted by Retrosheet. Interested parties may contact Retrosheet at ``www.retrosheet.org''.

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0), rmarkdown (>= 2.0.0)

NeedsCompilation no

Author Colin Douglas [aut, cre, cph], Richard Scriven [aut, cph]

Repository CRAN

Date/Publication 2023-07-07 07:40:02 UTC

R topics documented:

`getFileNames` *Files currently available for download*

Description

A convenience function, returning the base file names of the available downloads for the `year` and `type` arguments in `getRetrosheet`.

Usage

```
getFileNames()
```

Value

A named list of available single-season Retrosheet event and game-log zip files, and schedule text files. These file names are not intended to be passed to `getRetrosheet`, but is simply a fast way to determine if the desired data is available.

Examples

```
getFileNames()
```

`getParkIDs` *A data frame of ballpark IDs*

Description

This function returns a two-column data frame of ballpark IDs along with current stadium name

Usage

```
getParkIDs()
```

Examples

```
getParkIDs()
```

`getPartialGamelog` *Partial parser for game-log files*

Description

Instead of returning the entire file, this function allows the user to choose the columns and date for game-log data.

Usage

```
getPartialGamelog(year, glFields, date = NULL)

gamelogFields
```

Arguments

`year`	A single four-digit year.
`glFields`	character. The desired game-log columns. This should be a subset of `gamelogFields`, and **not** the entire vector.
`date`	One of either NULL (the default), or a single four-digit character string identifying the date 'mmdd'

Format

An object of class `character` of length 161.

Value

- `getPartialGamelog` - A data table with dimensions `length(date)` x `length(glFields)` if `date` is not NULL, otherwise the row dimension is the number of games for the given year.
- `gamelogFields` - A character vector of possible values to choose from for the `glFlields` argument in `getPartialGamelog`.

Examples

```
## Get Homerun and RBI info for the 2012 season, with park ID

f <- grep("HR|RBI|Park", gamelogFields, value = TRUE)
getPartialGamelog(2012, glFields = f)

## Get Homerun and RBI info for August 25, 2012 - with park ID
```

```
getPartialGamelog(glFields=f, date = "20120825")
```

`getRetrosheet` *Import single-season retrosheet data as a structured R object*

Description

This function downloads and parses data from https://www.retrosheet.org for the game-log, event, (play-by-play), roster, and schedule files.

Usage

```
getRetrosheet(
  type,
  year,
  team,
  schedSplit = NULL,
  stringsAsFactors = FALSE,
  cache = NA
)
```

Arguments

`type`	character. This argument can take on either of "game" for game-logs, "play" for play-by-play (a.k.a. event) data, "roster" for team rosters, or "schedule" for the game schedule for the given year.
`year`	integer. A valid four-digit year.
`team`	character. Only to be used if `type = "play"`. A single valid team ID for the given year. For available team IDs for the given year call `getTeamIDs(year)`. The available teams are in the "TeamID" column.
`schedSplit`	One of "Date", "HmTeam", or "TimeOfDay" to return a list split by the given value, or NULL (the default) for no splitting.
`stringsAsFactors`	logical. The `stringsAsFactors` argument as used in `data.frame`. Currently applicable to types "game" and "schedule".
`cache`	character. Path to local cache of retrosheet data. If file doesn't exist, files will be saved locally for future use. Defaults to "NA" so as not to save local data without explicit permission

Value

The following return values are possible for the given `type`

- `game` - a data frame of gamelog data for the given year
- `play` - a list, each element of which is a single game's play-by-play data for the given team and year. Each list element is also a list, containing the play-by-play data split into individual matrices.
- `roster` - a named list, each element containing the roster for the named team for the given year, as a data frame.
- `schedule` - a data frame containing the game schedule for the given year

Examples

```
## get the full 1995 season schedule
getRetrosheet("schedule", 1995)

## get the same schedule, split by time of day
getRetrosheet("schedule", 1995, schedSplit = "TimeOfDay")

## get the roster data for the 1995 season, listed by team
getRetrosheet("roster", 1995)

## get the full gamelog data for the 2012 season
getRetrosheet("game", 2012)

## get the play-by-play data for the San Francisco Giants' 2012 season
getRetrosheet("play", 2012, "SFN")
```

`getTeamIDs` *Retrieve team IDs for event files*

Description

This function retrieves the team ID needed for the `team` argument of `getRetrosheet("play", year, team)`.

Usage

```
getTeamIDs(year)
```

Arguments

`year`	A single valid four-digit numeric year.

Details

All currently available years can be retrieved with `type.convert(substr(getFileNames()$event, 1L, 4L))`

Value

If the file exists, a named vector of IDs for the given year. Otherwise `NA`.

Examples

```
getTeamIDs(2010)
```

---

get_retrosheet *Import single-season retrosheet data as data frames*

Description

This function is a wrapper for getRetrosheet(). It downloads and parses data from https://www.retrosheet.org for the game-log, event, (play-by-play), roster, and schedule files. While getRetrosheet() returns a list of matrices, this function returns an equivalent list of dataframes. It takes the same arguments, and can act as a drop-in replacement.

Usage

```
get_retrosheet(...)
```

Arguments

...	Arguments passed to 'getRetrosheet()'. 'stringsAsFactors' argument is always FALSE, and will warn if passed as TRUE

Value

The following return values are possible for the given `type`

- `game` - a data frame of gamelog data for the given year
- `play` - a list, each element of which is a single game's play-by-play data for the given team and year. Each list element is also a list, containing the play-by-play data split into individual matrices.
- `roster` - a named list, each element containing the roster for the named team for the given year, as a data frame.
- `schedule` - a data frame containing the game schedule for the given year

Examples

```
## get the full 1995 season schedule
get_retrosheet("schedule", 1995)

## get the same schedule, split by time of day
get_retrosheet("schedule", 1995, schedSplit = "TimeOfDay")

## get the roster data for the 1995 season, listed by team
get_retrosheet("roster", 1995)

## get the full gamelog data for the 2012 season
get_retrosheet("game", 2012)

## get the play-by-play data for the San Francisco Giants' 2012 season
get_retrosheet("play", 2012, "SFN")
```

Index