Package 'BANEScarparkinglite'

January 23, 2018

Type Package

Title Working with Car Parking Data for Bath and North East Somerset

Version 0.1.1

Description Contains functions for importing and working with the BANES car parking records and other related datasets. For the full version of the package, including all datasets, see the repo at <https://github.com/owenjonesuob/BANEScarparking>. The original dataset of parking records can be found at <https://data.bathhacked.org/Government-and-Society/BANES-Historic-Car-Park-Occupancy/x29s-cczc>.

License MIT + file LICENSE

URL https://github.com/owenjonesuob/BANEScarparkinglite

BugReports https://github.com/owenjonesuob/BANEScarparkinglite

Imports dplyr, httr, lubridate, RCurl, readr, rvest, utils, XML, xml2, zoo

Suggests testthat

RoxygenNote 6.0.1

NeedsCompilation no

Author Owen Jones [aut, cre], Ryan Kenning [aut, ctb]

Maintainer Owen Jones <olj23@bath.ac.uk>

Repository CRAN

Date/Publication 2018-01-23 09:56:26 UTC

R topics documented:

`BANEScarparkinglite-package` *Functions for obtaining and working with car parking data from Bath and North East Somerset*

Description

Contains functions for importing and working with the BANES car parking records and other related datasets. For the full version of the package, including all datasets, see the repo at https://github.com/owenjonesuob/BANESca

Author(s)

Owen Jones (<olj23@bath.ac.uk>)

`get_all_crude` *Download all raw records from Bath: Hacked datastore*

Description

Reads the entire CSV file found on the Bath: Hacked datastore (http://bit.ly/2i3Y1uF). The data frame created can subsequently be updated using `refuel_crude`.
Warning: *The file is very large! This may take a while to run, depending on your internet connection.*

Usage

```
get_all_crude()
```

Value

The full dataset of car parking records

See Also

- `refuel_crude` for updating raw records
- `refuel` for updating data already processed with `refine`

Examples

```
raw_data <- get_all_crude()

str(raw_data)
```

`get_daily_weather` *Scrape daily weather records for Bath*

Description

This function scrapes the Wunderground website to get daily weather summaries for Bath over a given date range.

Usage

```
get_daily_weather(from, to)
```

Arguments

`from`	A date or date-time object, or YYYY-MM-DD string: the first day from which to get a weather summary.
`to`	A date or date-time object, or YYYY-MM-DD string: the last day from which to get a weather summary.

Value

A data frame of daily weather summaries for each day in the specified range.

Examples

```
# Return weather summary for 01 January 2015
weather <- get_daily_weather("2015-01-01", "2015-01-01")
## Not run:
# Return daily weather summaries from 01 Oct 2014 to 17 Jul 2016
weather <- get_daily_weather("2014-10-01", "2016-07-17")


# Return daily event counts for all days in date range of parking records

library(lubridate)

raw_data <- get_all_crude()
df <- refine(raw_data)

weather <- rbind(get_daily_weather(min(df$LastUpdate), max(df$LastUpdate))

## End(Not run)
```

get_events *Scrape the number of advertised events in Bath for each day*

Description

Web scraping function to retrieve the number of events advertised at http://www.bath.co.uk/events for each day in a specified range of months.

Note: Have a look at this package's GitHub repo - in particular, here - to see the code for this function, along with comments which explain the process followed. See get_rugby *for a similar function with more detailed commentary!*

Usage

```
get_events(from, to)
```

Arguments

from	A date or date-time object, or YYYY-MM-DD string: the first day from which to get an event count.
to	A date or date-time object, or YYYY-MM-DD string: the last day from which to get an event count.

Value

A data frame of daily event counts for each day in the specified range.

See Also

get_events_detail

Examples

```
# Return event count for 01 January 2015
events <- get_events("2015-01-01", "2015-01-01")

# Return daily event counts from 01 Oct 2014 to 17 Jul 2015
events <- get_events("2014-10-01", "2015-07-17")

# Return daily event counts for all months in date range of parking records
raw_data <- get_all_crude()
df <- refine(raw_data)

events <- get_events(min(df$LastUpdate), max(df$LastUpdate))
```

`get_events_detail` *Scrape the details of advertised events in Bath for each day*

Description

Web scraping function to retrieve the detail for events advertised at http://www.bath.co.uk/events for each day in a specified range of dates.

Usage

```
get_events_detail(from, to)
```

Arguments

`from`	A date or date-time object, or string, of the first date for which to find events.
`to`	A date or date-time object, or string, of the last date for which to find events.

Value

A data frame of daily event details for each day in the specified range of months.

Author(s)

Ryan Kenning (@rkenning)

See Also

`get_events`

Examples

```
# Return event details for 01 January 2015
events <- get_events_detail("2015-01-01", "2015-01-01")
## Not run:
# Return daily event details from 01 Oct 2014 to 08 Oct 2014
events <- get_events_detail("2014-10-01", "2014-10-08")

## End(Not run)
```

get_range_crude *Download records from a specified range from the Bath: Hacked datastore*

Description

Retrieve raw records uploaded to the datastore within a specified date range and/or from a subset of car parks.

Usage

```
get_range_crude(from = NULL, to = NULL, abbrs = NULL)
```

Arguments

from	Datetime object (or "YYYY-MM-DD HH:MM:SS" string) for the earliest record to retrieve.
to	Datetime object (or "YYYY-MM-DD HH:MM:SS" string) for the latest record to retrieve.
abbrs	Abbreviations of names of car parks from which to retrieve records: **as** Avon Street CP **cs** Charlotte Street CP **l** Lansdown P+R **n** Newbridge P+R **od** Odd Down P+R **p** Podium CP **sg** SouthGate General CP **sr** Southgate Rail CP **t** test car park

Value

Car parking records from the specified date range.

See Also

get_all_crude

Examples

```
# Records for 1st June 2016
raw_data <- get_range_crude("2016-06-01 00:00:00", "2016-06-01 23:59:59")

# All records from Podium CP since 14:30 on 1st January 2017
raw_data <- get_range_crude(from = "2017-01-01 14:30:00", abbrs = "p")
```

```
# All records from P+Rs before 2015
raw_data <- get_range_crude(to = "2014-12-31 23:59:59", abbrs = c("l", "n", "od"))
```

`get_rugby` *Scrape Bath Rugby match dates, kick-off times and results*

Description

Web scraping function to gather dates and times of Bath Rugby matches, and whether or not Bath won, from the Bath Rugby website.

Note: This function's code is heavily commented so that if you want to, you can use it as a tutorial/guide for writing similar functions of your own! You can view the commented code on the GitHub repo here.

Usage

```
get_rugby(x)
```

Arguments

x	A vector of years of seasons to retrieve records from.

Value

A data frame of kick-off date-times and match outcomes.

Examples

```
# Return matches from 2016/17 season
rugby <- get_rugby(2017)

# Return matches from 2014/15, 2015/16 seasons
seasons <- c(2015, 2016)
rugby <- get_rugby(seasons)
```

`refine` *Clean up raw records*

Description

Uses functions from the `dplyr` package to clean up raw records obtained from the Bath: Hacked datastore. The process is as follows:

- Select columns containing useful information only
- Remove any records with NA entries
- Remove records for "test car park"
- Convert Name and Status to factors
- Remove records with negative occupancies
- Calculate Proportion column (Occupancy/Capacity)
- Remove records with Proportion greater than `max_prop`
- Remove duplicate records (see `first_upload`)

Usage

```
refine(x, max_prop = 1.1, first_upload = FALSE)
```

Arguments

`x`	A data frame containing records to be cleaned up (e.g. the data frame obtained by calling `get_all_crude`).
`max_prop`	The point at which records are discarded due to overly-full Occupancy values (default is 1.1, or 110% full, to allow for circulating cars).
`first_upload`	If `TRUE`, ensures that when duplicate records are removed, the copy which is kept is the first one uploaded after the record was taken. This takes much longer to run, due to sorting.

Value

A data frame of clean records, with 7 columns:

Name The name of the car park where the record was taken.

LastUpdate The time the record was taken (POSIXct date-time object).

DateUploaded The time the record was uploaded to the Bath: Hacked database (POSIXct date-time object).

Occupancy The total number of cars in the car park.

Capacity The number of parking spaces in the car park.

Status Description of the change in occupancy since the previous record from that car park.

Proportion Calculated as (Occupancy/Capacity).

Examples

```
raw_data <- get_all_crude()
some_records <- raw_data[1:1000, ]

dim(some_records)
## 1000   16

df <- refine(raw_data)
dim(df)
## 813   7
```

`refine.deduplicate` *Remove duplicate records (internal method)*

Description

(Internal method)

Usage

```
refine.deduplicate(x, first_upload)
```

Arguments

`x`	Data frame containing records.
`first_upload`	If TRUE, ensures record with oldest DateUploaded is kept.

`refuel` *Add new processed records from Bath: Hacked datastore*

Description

Update data frame of records already processed with `refine` with any records that have since been added to the Bath: Hacked datastore.

Usage

```
refuel(x, max_prop = 1.1, first_upload = FALSE)
```

Arguments

`x`	The result of calling `refine` on the data frame of records from the Bath: Hacked datastore (see `get_all_crude`).
`max_prop, first_upload`	See `refine`.

Value

The data frame updated with any more recent records.

See Also

- get_all_crude for obtaining data frame of raw records
- refuel_crude for updating data frame of raw records

Examples

```
raw_data <- get_all_crude()
df <- refine(get_all_crude)

# Some time later, add most recent records
df <- refuel(df)
```

---

refuel_crude *Add new raw records from Bath: Hacked datastore*

Description

Update data frame of raw records with any records that have since been added to the Bath: Hacked datastore.

Usage

```
refuel_crude(x)
```

Arguments

x	The data frame of raw records from the Bath: Hacked datastore (see get_all_crude).

Value

The data frame updated with any more recent records.

See Also

- get_all_crude for obtaining data frame of raw records
- refuel for updating data already processed with refine

Examples

```
raw_data <- get_all_crude()

# Some time later...
raw_data <- refuel_crude(raw_data)
```

Index