Package ‘valleybikeData’

October 12, 2020

Title ValleyBike.org Data Package

Version 0.0.1

Description All currently-available ValleyBike.org point data for 2018-2020, as well as some aggregated datasets.

Encoding UTF-8

LazyData true

Imports data.table,
dplyr,
fasttime,
fuzzyjoin,
janitor,
magrittr,
readr,
R.utils,
parallel,
stringr,
tibble

RoxygenNote 7.1.0

License MIT + file LICENSE

Depends R (>= 2.10)

R topics documented:

byMonth *By-month trajectory data*

Description

The by-month data sets (for 2018-2020) contain monthly trajectory data (latitude, longitude) collected during every trip, at 5-second intervals. These datasets are quite large (a few million entries), so they might lag your R session.

Usage

```
june2018

july2018

august2018

september2018

october2018

november2018

april2019

may2019

june2019

july2019

august2019

september2019

october2019

november2019

june2020

july2020

august2020
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 36773 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2054773 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 5802790 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 7180077 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 7331158 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2598266 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 4681751 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 3379888 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 4875254 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 4369828 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 4006793 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 3360060 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2223486 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 879494 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 435629 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2641636 rows and 6 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2825952 rows and 6 columns.

Variables

- route_id <chr>, the trip's unique route id (primary key)
- user_id <chr>, the rider's unique user id
- bike <chr>, unique bike id
- time <dttm>, the time at which the location was recorded (down to seconds)
- longitude <dbl>, the longitude of the bike at that point in time
- latitude <dbl>, the latitude of the bike at that point in time

---

`download_data` *download_data*

Description

Download raw data files

Usage

```
download_data(path, overwrite = FALSE)
```

Arguments

`path`	The path where to download the data files. Presumably, this will be inst/extdata.
`overwrite`	Whether to overwrite the existing files at the destination path. Defaults to FALSE.

Details

Download all available .csv.gz raw trajectory data files for the years 2018-2020 into a specified directory. Intended usage is for updating the files in inst/extdata to mirror those online.

Examples

```
## Not run:
download_data(path = "inst/extdata", overwrite = TRUE)

## End(Not run)
```

`import_day` *import_day*

Description

Import trajectory data for one day.

Usage

```
import_day(day, return = c("clean", "anomalous", "all"), future_cutoff = 24)
```

Arguments

`day`	The day for which to import the data (as a string of the form "YYYY-MM-DD").
`return`	The type of data to return (one of "clean", "anomalous", "all). Defaults to "clean".
`future_cutoff`	The next-day cutoff (in hours) past which observations are categorized as "anomalous", since rides may last past midnight. Defaults to 24.0 hours.

Details

Import trajectory data for a specific day. The user can choose to import the raw data, the clean data (i.e. the raw data minus any anomalous observations), or the anomalous data.

Value

A tibble of available trajectory data for that specific day.

Examples

```
data_22_may_2019 <- import_day("2019-05-22", return = "clean")
```

`import_full` *import_full*

Description

Import full trajectory data (raw)

Usage

```
import_full()
```

Details

Import all available trajectory data for the years 2018-2020, in raw format.

Value

A 60,910,226 x 6 tibble of all available trajectory data.

Examples

```
## Not run:
full_data <- import_full()

## End(Not run)
```

`import_month` *import_month*

Description

Import trajectory data for one month.

Usage

```
import_month(month, ...)
```

Arguments

`month`	The month for which to import the data (as a string of the form "YYYY-MM").
`...`	Further parameters to pass to ‘import_day()‘ (e.g. ‘return‘ or ‘future_cutoff‘).

Details

Import trajectory data for a specific month. The user can choose to import the raw data, the clean data (i.e. the raw data minus any anomalous observations), or the anomalous data.

Value

A tibble of available trajectory data for that specific month.

stations *ValleyBike stations (as of 2020)*

Description

This data set is contains information on the 54 ValleyBike stations.

Usage

```
stations
```

Format

A 54 x 8 data frame

Variables

- serial_num <int>, the station's serial number (primary key)
- name <chr>, the station's name
- address <chr> the station's address
- city <chr>, the city in which the station is
- latitude <dbl>, the station's latitude
- longitude <dbl>, the station's longitude
- docks <int>, the number of bike docks at the station
- display <chr>, display name for the station (usually name + city)

trips *ValleyBike trips over 2018-2020*

Description

This data set is an aggregated one-row-per-trip version of the original point-in-time ValleyBike data for the years 2018, 2019, and 2020. Some raw by-day .csv files were corrupted, so trips from those days are not documented. Many trips also show up with either a very low duration (e.g. 0-3 seconds) or an impossibly high one (e.g. 900 hours). They have been left in the data set to give people the opportunity of exploring them further.

Usage

```
trips
```

Format

A 118,839 x 12 data frame

Variables

- route_id <chr>, the trip's unique route id (primary key)
- user_id <chr>, the rider's unique user id
- bike <chr>, unique bike id
- start_time <dttm>, the trip's starting date-time (EDT)
- end_time <dttm>, the trip's ending date-time (EDT)
- start_station <chr>, the trip's starting station
- end_station <chr>, the trip's ending station
- start_latitude <dbl>, the trip's starting latitude
- start_longitude <dbl>, the trip's starting longitude
- end_latitude <dbl>, the trip's ending latitude
- end_longitude <dbl>, the trip's ending longitude
- duration <int>, the trip's duration (in seconds)

`users` *ValleyBike user statistics over 2018-2019*

Description

This data set is contains anonymous statistics for ValleyBike users in 2018, 2019, and 2020.

Usage

```
users
```

Format

A 12,553 x 10 data frame

Variables

- user_id <chr>, the user's unique id (primary key)
- num_trips <int>, the total number of trips taken by the user
- first_trip <dttm> the date-time of the user's first recorded trip
- last_trip <dttm> the date-time of the user's last recorded trip
- mean_trip_duration <dbl>, the user's mean trip duration
- median_trip_duration <dbl>, the user's median trip duration
- most_freq_start_station <chr>, the station at which the user most frequently starts a trip
- num_starting_there <int>, the number of trips starting at the user's most frequent start station
- most_freq_end_station <chr>, the station at which the user most frequently ends a trip
- num_ending_there <int>, the number of trips ending at the user's most frequent end station

Index