Package 'trread'

February 4, 2019

Type Package

Title Read, Validate, and Analyze Files in the General Transit Feed Specification

Version 0.2.7

Description Read General Transit Feed Specification (GTFS) zipfiles into a list of R dataframes. Perform validation of the data structure against the specification. Analyze the headways and frequencies at routes and stops. Please see the GTFS documentation here for more detail: <http://gtfs.org/>.

License GPL

LazyData TRUE

Depends R (>= 3.2.5)

Imports dplyr, zip, tibble, readr, data.table, httr, htmltools, magrittr, stringr, assertthat, scales, here, rlang, lubridate, hms, tidyr, tools

Suggests testthat, knitr, rmarkdown

RoxygenNote 6.1.0

URL https://github.com/r-transit/trread

BugReports https://github.com/r-transit/trread

VignetteBuilder knitr

NeedsCompilation no

Author Tom Buckley [aut, cre],
Danton Noriega-Goodwin [aut],
Flavio Poletti [aut],
Angela Li [aut],
Mark Padgham [aut],
Elaine McVey [ctb],
Charles Hans Thompson [ctb],
Michael Sumner [ctb],
Patrick Hausmann [ctb],
Bob Rudis [ctb],
Kearey Smith [ctb],

Dave Vautin [ctb],
Kyle Walker [ctb]

Maintainer Tom Buckley <tom@tbuckl.com>

Repository CRAN

Date/Publication 2019-02-04 08:30:03 UTC

R topics documented:

feedlist_df 2
filter_stops 3
get_date_service_table 3
get_feedlist 4
get_route_frequency 5
get_stop_frequency 5
gtfs_obj 6
import_gtfs 7
read_gtfs 8
route_type_names_df 9
set_api_key 9

Index **10**

`feedlist_df` *Dataframe of source GTFS data from Transitfeeds*

Description

A dataset containing a list of URLs for GTFS feeds

Usage

```
feedlist_df
```

Format

A data frame with 911 rows and 10 variables:

id the id of the feed on transitfeeds.com

t title of the feed

loc_id location id

loc_pid location placeid of the feed on transitfeeds.com

loc_t the title of the location

loc_n the shortname fo the location

loc_lat the location latitude

loc_lng the location longitude

url_d GTFS feed url

url_i the metadata url for the feed

Source

http://www.transitfeeds.com/

`filter_stops` *Get a set of stops for a given set of service ids and route ids*

Description

Get a set of stops for a given set of service ids and route ids

Usage

```
filter_stops(gtfs_obj, service_ids, route_ids)
```

Arguments

`gtfs_obj`	as read by read_gtfs()
`service_ids`	the service for which to get stops
`route_ids`	the route_ids for which to get stops

Value

stops for a given service

Examples

```
local_gtfs_path <- system.file("extdata", "google_transit_nyc_subway.zip", package = "trread")
nyc <- read_gtfs(local_gtfs_path,local=TRUE)
select_service_id <- filter(nyc$calendar_df,monday==1) %>% pull(service_id)
select_route_id <- sample_n(nyc$routes_df,1) %>% pull(route_id)
filtered_stops_df <- filter_stops(nyc,select_service_id,select_route_id)
```

`get_date_service_table`
Returns all possible date/service_id combinations as a data frame

---

Description

Use it to summarise service. For example, get a count of the number of services for a date. See example.

Usage

```
get_date_service_table(gtfs_obj)
```

Arguments

gtfs_obj	a gtfs_object as read by read_gtfs

Value

a date_service data frame

Examples

```
library(dplyr)
local_gtfs_path <- system.file("extdata", "google_transit_nyc_subway.zip", package = "trread")
nyc <- read_gtfs(local_gtfs_path, local=TRUE)
nyc_services_by_date <- nyc %>% get_date_service_table()
# count the number of services running on each date
nyc_services_by_date %>% group_by(date) %>% count()
```

---

get_feedlist *Get list of all available feeds from transitfeeds API*

Description

Get list of all available feeds from transitfeeds API

Usage

```
get_feedlist()
```

Value

a data frame with the gtfs feeds on transitfeeds.

See Also

feedlist_df

Examples

```
feedlist_df <- get_feedlist()
```

get_route_frequency *Get Route Frequency*

Description

should take:

Usage

```
get_route_frequency(gtfs_obj, start_hour = 6, end_hour = 22,
  quiet = FALSE, service_id = "", dow = c(1, 1, 1, 1, 1, 0, 0))
```

Arguments

gtfs_obj	a list of gtfs dataframes as read by the trread package.
start_hour	(optional) an integer, default 6 (6 am)
end_hour	(optional) an integer, default 22 (10 pm)
quiet	default FALSE. whether to echo process messages
service_id	(optional) a string from the calendar_df dataframe identifying a particular service schedule.
dow	(optional) an integeger vector with days of week. monday=1. default: c(1,1,1,1,1,0,0)

Value

a gtfs_obj with a dataframe of routes with variables for headway/frequency for a route within a given time frame

Examples

```
data(gtfs_obj)
gtfs_obj <- get_route_frequency(gtfs_obj)
x <- order(gtfs_obj$routes_frequency_df$median_headways)
head(gtfs_obj$routes_frequency_df[x,])
```

get_stop_frequency *Get Stop Frequency*

Description

Get Stop Frequency

Usage

```
get_stop_frequency(gtfs_obj, start_hour = 6, end_hour = 22,
  service_id = "", dow = c(1, 1, 1, 1, 1, 0, 0), by_route = TRUE,
  wide = FALSE)
```

Arguments

gtfs_obj	a list of gtfs dataframes as read by read_gtfs().
start_hour	(optional) an integer indicating the start hour (default 7)
end_hour	(optional) an integer indicating the end hour (default 20)
service_id	(optional) a string from the calendar_df dataframe identifying a particular service schedule.
dow	(optional) integer vector indicating which days of week to calculate for. default is weekday, e.g. c(1,1,1,1,1,0,0)
by_route	default TRUE, if FALSE then calculate headway for any line coming through the stop in the same direction on the same schedule.
wide	(optional) if true, then return a wide rather than tidy data frame

Value

a gtfs_obj with a dataframe of stops with a "Trips" variable representing the count trips taken through each stop for a route within a given time frame

Examples

```
data(gtfs_obj)
gtfs_obj <- get_stop_frequency(gtfs_obj)
x <- order(gtfs_obj$stops_frequency_df$headway)
head(gtfs_obj$stops_frequency_df[x,])
```

---

gtfs_obj *Example GTFS data*

Description

Data obtained from http://data.trilliumtransit.com/gtfs/duke-nc-us/duke-nc-us.zip.

Usage

```
gtfs_obj
```

Format

An object of class gtfs of length 22.

See Also

read_gtfs

`import_gtfs` *This function is deprecated. Please use read_gtfs*

Description

This function reads GTFS text files from a local or remote zip file. It also validates the files against the GTFS specification by file, requirement status, and column name The data are returned as a list of dataframes and a validation object, which contains details on whether all required files were found, and which required and optional columns are present.

Usage

```
import_gtfs(path, local = FALSE, quiet = FALSE)
```

Arguments

`path`	Character. url link to zip file OR path to local zip file. if to local path, then option `local` must be set to TRUE.
`local`	Boolean. If the paths are searching locally or not. Default is FALSE (that is, urls).
`quiet`	Boolean. Whether to see file download progress and files extract. FALSE by default.

Value

Dataframes of GTFS data.

Examples

```
library(dplyr)
u1 <- "https://github.com/r-transit/trread/raw/master/inst/extdata/sample-feed-fixed.zip"
sample_gtfs <- import_gtfs(u1)
attach(sample_gtfs)
#list routes by the number of stops they have
routes_df %>% inner_join(trips_df, by="route_id") %>%
  inner_join(stop_times_df) %>%
    inner_join(stops_df, by="stop_id") %>%
      group_by(route_long_name) %>%
        summarise(stop_count=n_distinct(stop_id)) %>%
          arrange(desc(stop_count))
```

`read_gtfs` *Get and validate dataframes of General Transit Feed Specification (GTFS) data.*

Description

This function reads GTFS text files from a local or remote zip file. It also validates the files against the GTFS specification by file, requirement status, and column name The data are returned as a list of dataframes and a validation object, which contains details on whether all required files were found, and which required and optional columns are present.

Usage

```
read_gtfs(path, local = FALSE, quiet = TRUE, frequency = FALSE)
```

Arguments

`path`	Character. url link to zip file OR path to local zip file. if to local path, then option `local` must be set to TRUE.
`local`	Boolean. If the paths are searching locally or not. Default is FALSE (that is, urls).
`quiet`	Boolean. Whether to see file download progress and files extract. FALSE by default.
`frequency`	Boolean. Whether to add frequency/headway calculations to the gtfs object

Value

A GTFS object. That is, a list of dataframes of GTFS data.

Examples

```
library(dplyr)
u1 <- "https://github.com/r-transit/trread/raw/master/inst/extdata/sample-feed-fixed.zip"
sample_gtfs <- read_gtfs(u1)
attach(sample_gtfs)
#list routes by the number of stops they have
routes_df %>% inner_join(trips_df, by="route_id") %>%
  inner_join(stop_times_df) %>%
    inner_join(stops_df, by="stop_id") %>%
      group_by(route_long_name) %>%
        summarise(stop_count=n_distinct(stop_id)) %>%
          arrange(desc(stop_count))
```

route_type_names_df *Dataframe of route type id's and the names of the types (e.g. "Cable Car")*

Description

Dataframe of route type id's and the names of the types (e.g. "Cable Car")

Usage

```
route_type_names_df
```

Format

A data frame with 122 rows and 2 variables:

id the id of route type

name name of the gtfs route type

Source

https://gist.github.com/derhuerst/b0243339e22c310bee2386388151e11e

set_api_key *Set API key for recall*

Description

Set API key for recall

Usage

```
set_api_key()
```

Index

∗Topic **datasets**
feedlist_df, 2
gtfs_obj, 6
route_type_names_df, 9

feedlist_df, 2
filter_stops, 3

get_date_service_table, 3
get_feedlist, 4
get_route_frequency, 5
get_stop_frequency, 5
gtfs_obj, 6

import_gtfs, 7

read_gtfs, 8
route_type_names_df, 9

set_api_key, 9