

Package ‘gtfstools’

February 23, 2021

Type Package

Title General Transit Feed Specification (GTFS) Editing and Analysing Tools

Version 0.1.0

Description Utility functions to read, manipulate, analyse and write transit feeds in the General Transit Feed Specification (GTFS) data format.

License MIT + file LICENSE

URL <https://ipeagit.github.io/gtfstools/>,
<https://github.com/ipeaGIT/gtfstools>

BugReports <https://github.com/ipeaGIT/gtfstools/issues>

Depends R (>= 2.10)

Imports checkmate, data.table, methods, sf, sfheaders, units, utils,
zip

Suggests covr, knitr, lwgeom, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Daniel Herszenhut [aut, cre] (<<https://orcid.org/0000-0001-8066-1105>>),
Ipea - Institute for Applied Economic Research [cph, fnd]

Maintainer Daniel Herszenhut <dhersz@gmail.com>

Repository CRAN

Date/Publication 2021-02-23 09:00:02 UTC

R topics documented:

get_trip_duration	2
get_trip_geometry	3
get_trip_segment_duration	4
get_trip_speed	5
gtfstools	6
merge_gtfs	7
read_gtfs	8
set_trip_speed	9
validate_gtfs	10
write_gtfs	12

Index	14
--------------	-----------

get_trip_duration	<i>Get trip duration</i>
-------------------	--------------------------

Description

Returns the duration of each specified trip_id.

Usage

```
get_trip_duration(gtfs, trip_id = NULL, unit = "min")
```

Arguments

gtfs	A GTFS object as created by read_gtfs .
trip_id	A string vector including the trip_ids to have their duration calculated. If NULL (the default) the function calculates the duration of every trip_id in the GTFS.
unit	A string representing the time unit in which the duration is desired. One of "s" (seconds), "min" (minutes, the default), "h" (hours) or "d" (days).

Value

A data.table containing the duration of each specified trip.

Details

The duration of a trip is defined as the time difference between its last arrival time and its first departure time, as specified in the stop_times file.

Examples

```
data_path <- system.file("extdata/spo_gtfs.zip", package = "gtfstools")

gtfs <- read_gtfs(data_path)

trip_duration <- get_trip_duration(gtfs)
head(trip_duration)

trip_ids <- c("CPTM L07-0", "2002-10-0")
trip_duration <- get_trip_duration(gtfs, trip_id = trip_ids)
trip_duration

trip_duration <- get_trip_duration(gtfs, trip_id = trip_ids, unit = "h")
trip_duration
```

get_trip_geometry	<i>Get trip geometry</i>
-------------------	--------------------------

Description

Returns the geometry of each specified `trip_id`, based either on the `shapes` or the `stop_times` file (or both).

Usage

```
get_trip_geometry(  
  gtfs,  
  trip_id = NULL,  
  file = c("shapes", "stop_times"),  
  crs = 4326  
)
```

Arguments

<code>gtfs</code>	A GTFS object as created by read_gtfs .
<code>trip_id</code>	A string vector including the <code>trip_ids</code> to have their geometries generated. If <code>NULL</code> (the default), the function generates geometries for every <code>trip_id</code> in the GTFS.
<code>file</code>	The file from which geometries should be generated. By default uses both <code>shapes</code> and <code>stop_times</code> .
<code>crs</code>	The CRS of the resulting object. Defaults to 4326 (WGS 84).

Value

A `LINestring sf`.

Details

The geometry generation works differently for the two files. In the case of shapes, the shape as described in the text file is converted to an sf object. For stop_times the geometry is the result of linking subsequent stops along a straight line (stops' coordinates are retrieved from the stops file). Thus, the resolution of the geometry when generated with shapes tends to be much higher than when created with stop_times.

Examples

```
data_path <- system.file("extdata/spo_gtfs.zip", package = "gtfstools")

gtfs <- read_gtfs(data_path)

trip_geometry <- get_trip_geometry(gtfs)
head(trip_geometry)

trip_ids <- c("CPTM L07-0", "2002-10-0")
trip_geometry <- get_trip_geometry(gtfs, trip_id = trip_ids)
trip_geometry
plot(trip_geometry["origin_file"])
```

```
get_trip_segment_duration
      Get trip segments' duration
```

Description

Returns the duration of segments between stops of each specified trip_id.

Usage

```
get_trip_segment_duration(gtfs, trip_id = NULL, unit = "min")
```

Arguments

gtfs	A GTFS object as created by read_gtfs .
trip_id	A string vector including the trip_ids to have their segments' duration calculated. If NULL (the default) the function calculates the segments' duration of every trip_id in the GTFS.
unit	A string representing the time unit in which the duration is desired. One of "s" (seconds), "min" (minutes, the default), "h" (hours) or "d" (days).

Value

A data.table containing the segments' duration of each specified trip.

Details

A trip segment is defined as the path between two subsequent stops in the same trip. The duration of a segment is defined as the time difference between its arrival time and its departure time, as specified in the stop_times file.

Examples

```
data_path <- system.file("extdata/spo_gtfs.zip", package = "gtfstools")

gtfs <- read_gtfs(data_path)

trip_segment_dur <- get_trip_segment_duration(gtfs)
head(trip_segment_dur)

trip_segment_dur <- get_trip_segment_duration(gtfs, trip_id = "CPTM L07-0")
trip_segment_dur

trip_segment_dur <- get_trip_segment_duration(gtfs, "CPTM L07-0", unit = "s")
trip_segment_dur
```

get_trip_speed	<i>Get trip speed</i>
----------------	-----------------------

Description

Returns the speed of each specified trip_id, based on the geometry created from either on the shapes or the stop_times file (or both).

Usage

```
get_trip_speed(gtfs, trip_id = NULL, file = "shapes", unit = "km/h")
```

Arguments

gtfs	A GTFS object as created by read_gtfs .
trip_id	A string vector including the trip_ids to have their speeds calculated. If NULL (the default), the function calculates the speed of every trip_id in the GTFS.
file	The file from which geometries should be generated, either shapes and stop_times (geometries are used to calculate the length of a trip). Defaults to shapes.
unit	A string representing the unit in which the speeds are desired. Either "km/h" (the default) or "m/s".

Value

A data.table containing the duration of each specified trip and the file from which geometries were generated.

Details

Please check [get_trip_geometry](#) documentation to understand how geometry generation differs depending on the chosen file.

See Also

[get_trip_geometry](#)

Examples

```
data_path <- system.file("extdata/spo_gtfs.zip", package = "gtfstools")

gtfs <- read_gtfs(data_path)

trip_speed <- get_trip_speed(gtfs)
head(trip_speed)

trip_ids <- c("CPTM L07-0", "2002-10-0")
trip_speed <- get_trip_speed(gtfs, trip_ids)
trip_speed

trip_speed <- get_trip_speed(
  gtfs,
  trip_ids,
  file = c("shapes", "stop_times")
)
trip_speed

trip_speed <- get_trip_speed(gtfs, trip_ids, unit = "m/s")
trip_speed
```

gtfstools

*gtfstools: General Transit Feed Specification (GTFS) Editing and
Analysing Tools*

Description

Utility functions to read, manipulate, analyse and write transit feeds in the General Transit Feed Specification (GTFS) data format.

Basic usage

Please check the introductory vignette for basic usage:

- Run `vignette("gtfstools")`;
- Alternatively, check it on [{gtfstools} website](#).

Author(s)

Maintainer: Daniel Herszenhut <dhersz@gmail.com> ([ORCID](#))

Other contributors:

- Ipea - Institute for Applied Economic Research [copyright holder, funder]

See Also

Useful links:

- <https://ipeagit.github.io/gtfstools/>
- <https://github.com/ipeaGIT/gtfstools>
- Report bugs at <https://github.com/ipeaGIT/gtfstools/issues>

merge_gtfs

Merge GTFS files

Description

Combines many GTFS file into a single one and validates the resulting object.

Usage

```
merge_gtfs(..., files = NULL, quiet = TRUE, warnings = TRUE)
```

Arguments

...	GTFS objects, as created by read_gtfs , to be merged. Each argument can either be a GTFS or a list of GTFS objects.
files	A character vector listing the GTFS text files (i.e. the ones represented by <code>data.tables</code>) to be merged. If <code>NULL</code> (the default) all files are merged.
quiet	Whether to hide log messages (defaults to <code>TRUE</code>).
warnings	Whether to display warning messages (defaults to <code>TRUE</code>).

Value

Returns a GTFS object, with an updated `validation_result` attribute, in which each `data.table` is the combination (by row) of `data.tables` with the same name from the GTFS objects passed in

....

Details

Please note that this function does not disambiguate `ids` that may be repeated within different GTFS objects. Please let us know if you'd like to see this feature implemented.

See Also[validate_gtfs](#)**Examples**

```
spo_path <- system.file("extdata/spo_gtfs.zip", package = "gtfstools")
ggl_path <- system.file("extdata/ggl_gtfs.zip", package = "gtfstools")

spo_gtfs <- read_gtfs(spo_path)
names(spo_gtfs)

ggl_gtfs <- read_gtfs(ggl_path)
names(ggl_gtfs)

merged_gtfs <- merge_gtfs(spo_gtfs, ggl_gtfs)
names(merged_gtfs)

# use a list() to programatically merge many GTFS objects
merged_gtfs <- merge_gtfs(list(spo_gtfs, ggl_gtfs))
```

`read_gtfs`*Read and validate GTFS files*

Description

Reads GTFS text files from either a local .zip file or an URL and validates them against GTFS specifications.

Usage

```
read_gtfs(path, files = NULL, quiet = TRUE, warnings = TRUE)
```

Arguments

<code>path</code>	The path to a GTFS .zip file.
<code>files</code>	A character vector containing the text files to be read from the GTFS (without the .txt extension). If NULL (the default) all existing files are read.
<code>quiet</code>	Whether to hide log messages and progress bars (defaults to TRUE).
<code>warnings</code>	Whether to display warning messages (defaults to TRUE).

Value

A GTFS object: a list of `data.tables` in which each index represents a GTFS text file. In case of parsing failures (e.g. files with more/less columns than specified in the file header), the function throws an error detailing where such failures occurred.

Details

The column types of each `data.table` in the final GTFS object conform as closely as possible to the [Google's Static GTFS Reference](#). Exceptions are date-related columns (such as `calendar.txt`'s `start_date` and `end_date`, for example), which are converted to `Date` objects, instead of being kept as integers, allowing for easier data manipulation. These columns are converted back to integers when writing the GTFS object to a `.zip` file using [write_gtfs](#).

See Also

[validate_gtfs](#)

Examples

```
data_path <- system.file("extdata/spo_gtfs.zip", package = "gtfstools")

gtfs <- read_gtfs(data_path)
names(gtfs)

gtfs <- read_gtfs(data_path, files = c("trips", "stop_times"))
names(gtfs)
```

set_trip_speed	<i>Set trip average speed</i>
----------------	-------------------------------

Description

Sets the average speed of each specified `trip_id` by changing the `arrival_time` and `departure_time` columns in `stop_times`.

Usage

```
set_trip_speed(gtfs, trip_id, speed, unit = "km/h", by_reference = FALSE)
```

Arguments

<code>gtfs</code>	A GTFS object as created by read_gtfs .
<code>trip_id</code>	A string vector including the <code>trip_ids</code> to have their average speed set.
<code>speed</code>	A numeric representing the speed to be set. Its length must either equal 1, in which case the value is recycled for all <code>trip_ids</code> , or equal <code>trip_id</code> 's length.
<code>unit</code>	A string representing the unit in which the speed is given. One of "km/h" (the default) or "m/s".
<code>by_reference</code>	Whether to update <code>stop_times</code> ' <code>data.table</code> by reference. Defaults to <code>FALSE</code> .

Value

If `by_reference` is set to `FALSE`, returns a GTFS object with the time columns of its `stop_times` adjusted. Else, returns a GTFS object invisibly (note that in this case the original GTFS object is altered).

Details

The average speed is calculated as the difference between the arrival time at the last stop minus the departure time at the first stop, over the trip's length (as calculated via `get_trip_geometry`, based on the shapes file). The arrival and departure times at all other stops (i.e. not the first neither the last) are set as `"`, which is written as NA with `write_gtfs`. Some transport routing software, such as [OpenTripPlanner](#), support specifying stop times like so. In such cases, they estimate arrival/departure times at the others stops based on the average speed as well. We plan to add that feature to this function in the future.

Examples

```
data_path <- system.file("extdata/spo_gtfs.zip", package = "gtfstools")

gtfs <- read_gtfs(data_path)

gtfs_new_speed <- set_trip_speed(gtfs, trip_id = "CPTM L07-0", 50)
gtfs_new_speed$stop_times[trip_id == "CPTM L07-0"]

# original gtfs remains unchanged
gtfs$stop_times[trip_id == "CPTM L07-0"]

# now do it by reference
set_trip_speed(gtfs, trip_id = "CPTM L07-0", 50, by_reference = TRUE)
gtfs$stop_times[trip_id == "CPTM L07-0"]
```

 validate_gtfs

Validate GTFS file

Description

Validates the GTFS object against GTFS specifications and raises warnings if required files/fields are not found.

Usage

```
validate_gtfs(gtfs, files = NULL, quiet = TRUE, warnings = TRUE)
```

Arguments

gtfs	A GTFS object as created by read_gtfs .
files	A character vector containing the text files to be validated against the GTFS specification (without the .txt extension). If NULL (the default) the provided GTFS is validated against all possible GTFS text files.
quiet	Whether to hide log messages (defaults to TRUE).
warnings	Whether to display warning messages (defaults to TRUE).

Value

A GTFS object with a `validation_result` attribute. This attribute is a `data.table` containing the validation summary of all possible fields from the specified files.

Details

GTFS object's files and fields are validated against the GTFS specifications as documented in [Google's Static GTFS Reference](#):

- GTFS feeds are considered valid if they include all required files and fields. If a required file/field is missing the function (optionally) raises a warning.
- Optional files/fields are listed in the reference above but are not required, thus no warning is raised if they are missing.
- Extra files/fields are those who are not listed in the reference above (either because they refer to a specific GTFS extension or due to any other reason).

Note that some files (`calendar.txt`, `calendar_dates.txt` and `feed_info.txt`) are conditionally required. This means that:

- `calendar.txt` is initially set as a required file. If it's not present, however, it becomes optional and `calendar_dates.txt` (originally set as optional) becomes required.
- `feed_info.txt` is initially set as an optional file. If `translations.txt` is present, however, it becomes required.

Examples

```
data_path <- system.file("extdata/spo_gtfs.zip", package = "gtfstools")

gtfs <- read_gtfs(data_path)
attr(gtfs, "validation_result")

# should not raise a warning, because 'shapes' is not a required file
gtfs$shapes <- NULL
validation_result <- validate_gtfs(gtfs)

# should raise a warning, because 'stop_times' is a required file
gtfs$stop_times <- NULL
validation_result <- validate_gtfs(gtfs)
```

`write_gtfs`*Write GTFS files*

Description

Writes in-memory GTFS objects as GTFS .zip files. Conditionally includes optional and extra .txt files (check [validate_gtfs](#) documentation to check what are optional/extra files).

Usage

```
write_gtfs(  
  gtfs,  
  path,  
  optional = TRUE,  
  extra = TRUE,  
  overwrite = TRUE,  
  quiet = TRUE,  
  warnings = TRUE  
)
```

Arguments

<code>gtfs</code>	A GTFS object as created by read_gtfs .
<code>path</code>	The path to the .zip file in which the feed should be written to.
<code>optional</code>	Whether to write optional .txt. Defaults to TRUE.
<code>extra</code>	Whether to write extra .txt. Defaults to TRUE.
<code>overwrite</code>	Whether to overwrite existing .zip file. Defaults to TRUE.
<code>quiet</code>	Whether to hide log messages and progress bars (defaults to TRUE).
<code>warnings</code>	Whether to display warning messages (defaults to TRUE).

Value

Invisibly returns the provided GTFS object with an updated `validation_result` attribute.

See Also

[validate_gtfs](#)

Examples

```
data_path <- system.file("extdata/spo_gtfs.zip", package = "gtfstools")  
gtfs <- read_gtfs(data_path)  
  
tmp_dir <- file.path(tempdir(), "tmpdir")  
dir.create(tmp_dir)  
list.files(tmp_dir)
```

```
tmp_file <- tempfile(pattern = "gtfs", tmpdir = tmp_dir, fileext = ".zip")
write_gtfs(gtfs, tmp_file)
list.files(tmp_dir)

gtfs_all_files <- read_gtfs(tmp_file)
names(gtfs_all_files)

write_gtfs(gtfs_all_files, tmp_file, optional = FALSE)
gtfs_no_opt <- read_gtfs(tmp_file)
names(gtfs_no_opt)
```

Index

`_PACKAGE` (gtfstools), 6

`get_trip_duration`, 2

`get_trip_geometry`, 3, 6, 10

`get_trip_segment_duration`, 4

`get_trip_speed`, 5

gtfstools, 6

gtfstools-package (gtfstools), 6

`merge_gtfs`, 7

`read_gtfs`, 2–5, 7, 8, 9, 11, 12

`set_trip_speed`, 9

`validate_gtfs`, 8, 9, 10, 12

`write_gtfs`, 9, 10, 12