Package ‘r5r’

April 16, 2021

Type Package

Title Rapid Realistic Routing with 'R5'

Version 0.4.0

Description Rapid realistic routing on multimodal transport networks (walk, bike, public transport and car) using 'R5', the Rapid Realistic Routing on Real-world and Reimagined networks <https://github.com/conveyal/r5>. The package allows users to generate detailed routing analysis or calculate travel time matrices using seamless parallel computing on top of the R5 Java machine.

Date 2021-04-15

URL https://github.com/ipeaGIT/r5r

BugReports https://github.com/ipeaGIT/r5r/issues

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.6)

Suggests akima, covr, dplyr, ggplot2 (>= 3.3.1), knitr, mapview, markdown, testthat

Imports checkmate, curl, data.table, httr, methods, jdx (>= 0.1.4), rJava (>= 0.9-10), sf (>= 0.9-3), sfheaders, utils

SystemRequirements Java JDK (>= 11.0)

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Marcus Saraiva [aut] (<https://orcid.org/0000-0001-6218-2338>), Rafael H. M. Pereira [aut, cre] (<https://orcid.org/0000-0003-2125-7465>), Daniel Herszenhut [aut] (<https://orcid.org/0000-0001-8066-1105>), Carlos Kaue Vieira Braga [aut] (<https://orcid.org/0000-0002-6104-7297>), Matthew Wigginton Conway [aut] (<https://orcid.org/0000-0002-1210-2982>), Ipea - Institute for Applied Economic Research [cph, fnd]
assert_points_input

**Description**

Assert class of origin and destination inputs and the type of its columns.

**Usage**

```r
assert_points_input(df, name)
```

**Arguments**

- `df` Any object.
- `name` Object name.

**Value**

A data.frame with columns `id`, `lon` and `lat`.
**check_connection**

Check internet connection with Ipea server

**Description**
Checks if there is internet connection to Ipea server to download geobr data.

**Usage**

```r
check_connection(
  file_url = "https://www.ipea.gov.br/geobr/metadata/metadata_gpkg.csv"
)
```

**Arguments**

- `file_url` A string with the file_url address of an geobr dataset

**Value**
No return value, called for side effects.

**See Also**

Other support functions: `check_connection()`, `download_metadata()`, `posix_to_string()`, `select_mode()`, `set_max_lts()`, `set_max_rides()`, `set_max_street_time()`, `set_n_threads()`, `set_speed()`, `set_suboptimal_minutes()`, `set_verbose()`, `stop_r5()`, `street_network_to_sf()`

---

**detailed_itineraries**  
Calculate detailed itineraries between origin destination pairs

**Description**
Fast computation of (multiple) detailed itineraries between one or many origin destination pairs.
detailed_itineraries (r5r_core, origins, destinations, mode = "WALK", mode_egress = "WALK", departure_datetime = Sys.time(), max_walk_dist = Inf, max_trip_duration = 120L, walk_speed = 3.6, bike_speed = 12, max_rides = 3, max_lts = 2, shortest_path = TRUE, n_threads = Inf, verbose = TRUE, drop_geometry = FALSE)

Arguments

r5r_core rJava object to connect with R5 routing engine
origins, destinations either a spatial sf POINT object or a data.frame containing the columns 'id', 'lon', 'lat'
mode string. Transport modes allowed for the trips. Defaults to "WALK". See details for other options.
mode_egress string. Transport mode used after egress from public transport. It can be either 'WALK', 'BICYCLE', or 'CAR'. Defaults to "WALK". Ignored when public transport is not used.
departure_datetime POSIXct object. If working with public transport networks, please check calendar.txt within the GTFS file for valid dates.
max_walk_dist numeric. Maximum walking distance (in meters) for the whole trip. Defaults to no restrictions on walking, as long as max_trip_duration is respected.
max_trip_duration numeric. Maximum trip duration in minutes. Defaults to 120 minutes (2 hours).
walk_speed numeric. Average walk speed in km/h. Defaults to 3.6 km/h.
bike_speed numeric. Average cycling speed in km/h. Defaults to 12 km/h.
max_rides numeric. The max number of public transport rides allowed in the same trip. Defaults to 3.
max_lts numeric (between 1 and 4). The maximum level of traffic stress that cyclists will tolerate. A value of 1 means cyclists will only travel through the quietest streets, while a value of 4 indicates cyclists can travel through any road. Defaults to 2. See details for more information.
detailed_itineraries

shortest_path logical. Whether the function should only return the fastest route alternative (the default) or multiple alternatives.

n_threads numeric. The number of threads to use in parallel computing. Defaults to use all available threads (Inf).

verbose logical. TRUE to show detailed output messages (the default) or FALSE to show only eventual ERROR messages.

drop_geometry logical. Indicates whether R5 should drop segment’s geometry column. It can be helpful for saving memory.

Value

A LINESTRING sf with detailed information about the itineraries between specified origins and destinations. Distances are in meters and travel times are in minutes.

Transpor modes:

R5 allows for multiple combinations of transport modes. The options include:

**Transit modes:**

TRAM, SUBWAY, RAIL, BUS, FERRY, CABLE_CAR, GONDOLA, FUNICULAR. The option 'TRANSIT’ automatically considers all public transport modes available.

**Non transit modes:**

WALK, BICYCLE, CAR, BICYCLE_RENT, CAR_PARK

max_lts, Maximum Level of Traffic Stress:

When cycling is enabled in R5, setting max_lts will allow cycling only on streets with a given level of danger/stress. Setting max_lts to 1, for example, will allow cycling only on separated bicycle infrastructure or low-traffic streets; routing will revert to walking when traversing any links with LTS exceeding 1. Setting max_lts to 3 will allow cycling on links with LTS 1, 2, or 3.

The default methodology for assigning LTS values to network edges is based on commonly tagged attributes of OSM ways. See more info about LTS at [https://docs.conveyal.com/learn-more/traffic-stress](https://docs.conveyal.com/learn-more/traffic-stress). In summary:

- **LTS 1:** Tolerable for children. This includes low-speed, low-volume streets, as well as those with separated bicycle facilities (such as parking-protected lanes or cycle tracks).

- **LTS 2:** Tolerable for the mainstream adult population. This includes streets where cyclists have dedicated lanes and only have to interact with traffic at formal crossing.

- **LTS 3:** Tolerable for “enthused and confident” cyclists. This includes streets which may involve close proximity to moderate- or high-speed vehicular traffic.

- **LTS 4:** Tolerable for only “strong and fearless” cyclists. This includes streets where cyclists are required to mix with moderate- to high-speed vehicular traffic.
Routing algorithm:

The detailed_itineraries function uses an R5-specific extension to the McRAPTOR routing algorithm to find paths that are optimal or less than optimal, with some heuristics around multiple access modes, riding the same patterns, etc. The specific extension to McRAPTOR to do suboptimal path routing are not documented yet, but a detailed description of base McRAPTOR can be found in Delling et al (2015).


See Also

Other routing: travel_time_matrix()
download_r5

Value

A data.frame with url address of r5r Jar files

See Also

Other support functions: assert_points_input(), check_connection(), posix_to_string(), select_mode(), set_max_lts(), set_max_rides(), set_max_street_time(), set_n_threads(), set_speed(), set_suboptimal_minutes(), set_verbose(), stop_r5(), street_network_to_sf()

---

download_r5  Download R5 Jar file

Description

Download a compiled JAR file of R5 and saves it locally. The JAR file is saved within the package directory. The package uses a compilation of R5 tailored for the purposes of r5r that keeps R5’s essential features. Source code available at https://github.com/ipeaGIT/r5r.

Usage

download_r5(
  version = "6.0.1",
  quiet = FALSE,
  force_update = FALSE,
  temp_dir = FALSE
)

Arguments

version  character string with the version of R5 to be downloaded. Defaults to latest version '1.0'.
quiet  logical, passed to download.file. Defaults to FALSE
force_update  logical, Replaces the jar file stored locally with a new one. Defaults to FALSE.
temp_dir  logical, whether the R5 Jar file should be saved in temporary directory. Defaults to FALSE

Value

A jar file is saved locally in the r5r package directory

See Also

Other setup: setup_r5()
Examples
if (interactive()) {
library(r5r)

download_r5(version = "6.0.1", temp_dir = TRUE)
}

posix_to_string
Generate date and departure time strings from POSIXct

Description
Generate date and departure time strings from POSIXct

Usage
posix_to_string(datetime)

Arguments
datetime
An object of POSIXct class.

Value
A list with the date and time of the trip departure as characters
A list with 'date' and 'departure_time' names.

See Also
Other support functions: assert_points_input(), check_connection(), download_metadata(),
select_mode(), set_max_lts(), set_max_rides(), set_max_street_time(), set_n_threads(),
set_speed(), set_suboptimal_minutes(), set_verbose(), stop_r5(), street_network_to_sf()

select_mode
Select transport mode

Description
Select transport mode

Usage
select_mode(mode, mode_egress)
setup_r5

Arguments

mode character string passed from routing functions.
mode_egress character string passed from routing functions.

Value

A list with the transport modes used in the routing.

See Also

Other support functions: assert_points_input(), check_connection(), download_metadata(), posix_to_string(), set_max_lts(), set_max_rides(), set_max_street_time(), set_n_threads(), set_speed(), set_suboptimal_minutes(), set_verbose(), stop_r5(), street_network_to_sf()

setup_r5 Create transport network used for routing in R5

Description

Combine data inputs in a directory to build a multimodal transport network used for routing in R5. The directory must contain at least one street network file (in .pbf format). One or more public transport data sets (in GTFS.zip format) are optional. If there is more than one GTFS file in the directory, both files will be merged. If there is already a ‘network.dat’ file in the directory the function will simply read it and load it to memory.

Usage

setup_r5(data_path, version = "6.2.0", verbose = TRUE, temp_dir = FALSE)

Arguments

data_path character string, the directory where data inputs are stored and where the built network.dat will be saved.
version character string, the version of R5 to be used. Defaults to latest version ‘6.2.0’.
verbose logical, TRUE to show detailed output messages (Default) or FALSE to show only eventual ERROR and WARNING messages.
temp_dir logical, whether the R5 Jar file should be saved in temporary directory. Defaults to FALSE

Value

An rJava object to connect with R5 routing engine

See Also

Other setup: download_r5()
Examples

```r
if (interactive()) {
  library(r5r)

  # directory with street network and gtfs files
  path <- system.file("extdata/poa", package = "r5r")

  r5r_core <- setup_r5(data_path = path, temp_dir = TRUE)
}
```

---

**set_max_lts**  
*Set max Level of Transit Stress (LTS)*

---

**Description**

Set max Level of Transit Stress (LTS)

**Usage**

```r
set_max_lts(r5r_core, max_lts)
```

**Arguments**

- `r5r_core`: rJava object to connect with R5 routing engine
- `max_lts`: numeric (between 1 and 4). The maximum level of traffic stress that cyclists will tolerate. A value of 1 means cyclists will only travel through the quietest streets, while a value of 4 indicates cyclists can travel through any road.

**Value**

No return value, called for side effects.

**See Also**

Other support functions: `assert_points_input()`, `check_connection()`, `download_metadata()`, `posix_to_string()`, `select_mode()`, `set_max_rides()`, `set_max_street_time()`, `set_n_threads()`, `set_speed()`, `set_suboptimal_minutes()`, `set_verbose()`, `stop_r5()`, `street_network_to_sf()`
**set_max_rides**

| set_max_rides | Set max number of transfers |

**Description**

Set maxTransfers parameter in R5.

**Usage**

```
set_max_rides(r5r_core, max_rides)
```

**Arguments**

- `r5r_core` rJava object to connect with R5 routing engine
- `max_rides` numeric. The max number of public transport rides allowed in the same trip. Passed from routing function.

**Value**

No return value, called for side effects.

**See Also**

Other support functions: assert_points_input(), check_connection(), download_metadata(), posix_to_string(), select_mode(), set_max_lts(), set_max_street_time(), set_n_threads(), set_speed(), set_suboptimal_minutes(), setVerbose(), stop_r5(), street_network_to_sf()

**set_max_street_time**

| set_max_street_time | Set max street time |

**Description**

Set max street time

**Usage**

```
set_max_street_time(max_walk_dist, walk_speed, max_trip_duration)
```

**Arguments**

- `max_walk_dist` numeric. Maximum walking distance (in meters) for the whole trip. Passed from routing functions.
- `walk_speed` numeric. Average walk speed in Km/h. Defaults to 3.6 Km/h. Passed from routing functions.
- `max_trip_duration` numeric. Maximum trip duration in seconds. Defaults to 120 minutes (2 hours). Passed from routing functions.
set_n_threads

Value

An integer representing the maximum number of minutes walking.

See Also

Other support functions: `assert_points_input()`, `check_connection()`, `download_metadata()`, `posix_to_string()`, `select_mode()`, `set_max_lts()`, `set_max_rides()`, `set_n_threads()`, `set_speed()`, `set_suboptimal_minutes()`, `set_verbose()`, `stop_r5()`, `street_network_to_sf()`

---

<table>
<thead>
<tr>
<th>set_n_threads</th>
<th>Set number of threads</th>
</tr>
</thead>
</table>

Description

Sets number of threads to be used by the r5r.jar.

Usage

`set_n_threads(r5r_core, n_threads)`

Arguments

- `r5r_core`: a rJava object to connect with R5 routing engine
- `n_threads`: Any object.

Value

No return value, called for side effects.

See Also

Other support functions: `assert_points_input()`, `check_connection()`, `download_metadata()`, `posix_to_string()`, `select_mode()`, `set_max_lts()`, `set_max_rides()`, `set_max_street_time()`, `set_speed()`, `set_suboptimal_minutes()`, `set_verbose()`, `stop_r5()`, `street_network_to_sf()`
set_speed

Set walk and bike speed

Description
This function receives the walk and bike 'speed' inputs in Km/h from routing functions above and converts them to meters per second, which is then used to set these speed profiles in r5 JAR.

Usage
set_speed(r5r_core, speed, mode)

Arguments
- r5r_core: a rJava object to connect with R5 routing engine
- speed: A numeric representing the speed in km/h.
- mode: Either "bike" or "walk".

Value
No return value, called for side effects.

See Also
Other support functions: assert_points_input(), check_connection(), download_metadata(), posix_to_string(), select_mode(), set_max_lts(), set_max_rides(), set_max_street_time(), set_n_threads(), set_suboptimal_minutes(), set_verbose(), stop_r5(), street_network_to_sf()
set_verbose

**Arguments**

- **r5r_core**: rJava object to connect with R5 routing engine
- **suboptimal_minutes**: numeric. The number of suboptimal minutes in a public transport point-to-point query. From R5’s documentation: This parameter compensates for the fact that GTFS does not contain information about schedule deviation (lateness). The min-max travel time range for some trains is zero, since the trips are reported to always have the same timings in the schedule. Such an option does not overlap (temporally) its alternatives, and is too easily eliminated by an alternative that is only marginally better. We want to effectively push the max travel time of alternatives out a bit to account for the fact that they don’t always run on schedule.

**Value**

No return value, called for side effects.

**See Also**

Other support functions: assert_points_input(), check_connection(), download_metadata(), posix_to_string(), select_mode(), set_max_lts(), set_max_rides(), set_max_street_time(), set_n_threads(), set_speed(), set_verbose(), stop_r5(), street_network_to_sf()
stop_r5

Description

Stops running r5r cores.

Usage

stop_r5(...)

Arguments

... r5r_core objects currently running. By default, if no cores are supplied all running cores are stopped.

Value

No return value, called for side effects.

See Also

Other support functions: assert_points_input(), check_connection(), download_metadata(), posix_to_string(), select_mode(), set_max_lts(), set_max_rides(), set_max_street_time(), set_n_threads(), set_speed(), set_suboptimal_minutes(), set_verbose(), street_network_to_sf()

Examples

if (interactive()) {
  library(r5r)

  path <- system.file("extdata/poa", package = "r5r")

  r5r_core <- setup_r5(data_path = path)

  stop_r5(r5r_core)
}
street_network_to_sf

Extract OpenStreetMap network in sf format from a network.dat built with setup_r5

Description
Extract OpenStreetMap network in sf format from a network.dat built with setup_r5

Usage
street_network_to_sf(r5r_core)

Arguments
r5r_core 
a rJava object, the output from 'r5r::setup_r5()'

Value
A list with two components of a street network in sf format: vertices (POINT) and edges (LINESTRING).

See Also
Other support functions: assert_points_input(), check_connection(), download_metadata(), POSIX_to_string(), select_mode(), set_max_lts(), set_max_rides(), set_max_street_time(), set_n_threads(), set_speed(), set_suboptimal_minutes(), set_verbose(), stop_r5()

Examples
if (interactive()) {
  library(r5r)

  # build transport network
  path <- system.file("extdata/poa", package = "r5r")
  r5r_core <- setup_r5(data_path = path)

  # extract street network from r5r_core
  street_net <- street_network_to_sf(r5r_core)

  stop_r5(r5r_core)
}
travel_time_matrix

Calculate travel time matrix between origin destination pairs

Description

Fast computation of travel time estimates between one or multiple origin destination pairs.

Usage

```r
travel_time_matrix(
  r5r_core,
  origins, destinations,
  mode = "WALK",
  mode_egress = "WALK",
  departure_datetime = Sys.time(),
  time_window = 1L,
  percentiles = 50L,
  max_walk_dist = Inf,
  max_trip_duration = 120L,
  walk_speed = 3.6,
  bike_speed = 12,
  max_rides = 3,
  max_lts = 2,
  n_threads = Inf,
  verbose = TRUE
)
```

Arguments

- **r5r_core**: a rJava object to connect with R5 routing engine
- **origins, destinations**: a spatial sf POINT object, or a data.frame containing the columns 'id', 'lon', 'lat'
- **mode**: string. Transport modes allowed for the trips. Defaults to "WALK". See details for other options.
- **mode_egress**: string. Transport mode used after egress from public transport. It can be either 'WALK', 'BICYCLE', or 'CAR'. Defaults to "WALK".
- **departure_datetime**: POSIXct object. If working with public transport networks, please check calendar.txt within the GTFS file for valid dates.
- **time_window**: numeric. Time window in minutes for which r5r will calculate multiple travel time matrices departing each minute. By default, the number of simulations is 5 times the size of 'time_window' set by the user. Defaults window size to `1`, the function only considers 5 departure times. This parameter is only used with frequency-based GTFS files. See details for further information.

- **percentiles**: numeric. Percentiles to compute from the distribution of travel times.
- **max_walk_dist**: numeric. Maximum distance to walk. If not specified, it will use the maximum distance allowed by R5.
- **max_trip_duration**: numeric. Maximum allowed trip duration. It is expressed in minutes.
- **walk_speed**: numeric. Speed in km/h for walking.
- **bike_speed**: numeric. Speed in km/h for biking.
- **max_rides**: numeric. Maximum number of rides allowed in one trip.
- **max_lts**: numeric. Maximum number of layover times allowed in one trip.
- **n_threads**: numeric. Number of threads to use. Use Inf to use all available threads.
- **verbose**: logical. Show progress messages and warnings.
percentiles numeric vector. Defaults to '50', returning the median travel time for a given time_window. If a numeric vector is passed, for example c(25, 50, 75), the function will return additional columns with the travel times within percentiles of trips. For example, if the 25 percentile of trips between A and B is 15 minutes, this means that 25% of all trips taken between A and B within the set time window are shorter than 15 minutes. Only the first 5 cut points of the percentiles are considered. For more details, see R5 documentation at 'https://docs.conveyal.com/analysis/methodology#accounting-for-variability'.

max_walk_dist numeric. Maximum walking distance (in meters) for the whole trip. Defaults to no restrictions on walking, as long as max_trip_duration is respected.

max_trip_duration numeric. Maximum trip duration in minutes. Defaults to 120 minutes (2 hours).

walk_speed numeric. Average walk speed in km/h. Defaults to 3.6 km/h.

bike_speed numeric. Average cycling speed in km/h. Defaults to 12 km/h.

max_rides numeric. The max number of public transport rides allowed in the same trip. Defaults to 3.

max_lts numeric (between 1 and 4). The maximum level of traffic stress that cyclists will tolerate. A value of 1 means cyclists will only travel through the quietest streets, while a value of 4 indicates cyclists can travel through any road. Defaults to 2. See details for more information.

n_threads numeric. The number of threads to use in parallel computing. Defaults to use all available threads (Inf).

verbose logical. TRUE to show detailed output messages (the default) or FALSE to show only eventual ERROR messages.

Value

A data.table with travel time estimates (in minutes) between origin destination pairs by a given transport mode. Note that origins/destinations that were beyond the maximum travel time, and/or origins that were far from the street network are not returned in the data.table.

Transport modes:

R5 allows for multiple combinations of transport modes. The options include:

**Transit modes:**
TRAM, SUBWAY, RAIL, BUS, FERRY, CABLE_CAR, GONDOLA, Funicular. The option 'TRANSIT' automatically considers all public transport modes available.

**Non transit modes:**
WALK, BICYCLE, CAR, BICYCLE_RENT, CAR_PARK

**max_lts, Maximum Level of Traffic Stress:**

When cycling is enabled in R5, setting max_lts will allow cycling only on streets with a given level of danger/stress. Setting max_lts to 1, for example, will allow cycling only on separated bicycle
infrastructure or low-traffic streets; routing will revert to walking when traversing any links with LTS exceeding 1. Setting max_lts to 3 will allow cycling on links with LTS 1, 2, or 3.

The default methodology for assigning LTS values to network edges is based on commonly tagged attributes of OSM ways. See more info about LTS at [https://docs.conveyal.com/learn-more/traffic-stress](https://docs.conveyal.com/learn-more/traffic-stress). In summary:

- **LTS 1**: Tolerable for children. This includes low-speed, low-volume streets, as well as those with separated bicycle facilities (such as parking-protected lanes or cycle tracks).
- **LTS 2**: Tolerable for the mainstream adult population. This includes streets where cyclists have dedicated lanes and only have to interact with traffic at formal crossing.
- **LTS 3**: Tolerable for “enthused and confident” cyclists. This includes streets which may involve close proximity to moderate- or high-speed vehicular traffic.
- **LTS 4**: Tolerable for only “strong and fearless” cyclists. This includes streets where cyclists are required to mix with moderate- to high-speed vehicular traffic.

**Routing algorithm:**

The travel_time_matrix function uses an R5-specific extension to the RAPTOR routing algorithm (see Conway et al., 2017). This RAPTOR extension uses a systematic sample of one departure per minute over the time window set by the user in the ‘time_window’ parameter. A detailed description of base RAPTOR can be found in Delling et al (2015).


**See Also**

Other routing: `detailed_itineraries()`

**Examples**

```r
if (interactive()) {
  library(r5r)

  # build transport network
  data_path <- system.file("extdata/spo", package = "r5r")
  r5r_core <- setup_r5(data_path = data_path)

  # load origin/destination points
  points <- read.csv(file.path(data_path, "spo_hexgrid.csv"))[1:5,]
  departure_datetime <- as.POSIXct("13-05-2019 14:00:00", format = "%d-%m-%Y %H:%M:%S")

  # estimate travel time matrix
  ttm <- travel_time_matrix(r5r_core, 
    origins = points, 
    destinations = points, 
    max_lts = 3, 
    time_window = c(departure_datetime, "01:00:00"))
}
```
mode = c("WALK", "TRANSIT"),
departure_datetime = departure_datetime,
max_walk_dist = Inf,
max_trip_duration = 120L)

stop_r5(r5r_core)

}
Index

* routing
  detailed_itineraries, 3
  travel_time_matrix, 17

* setup
  download_r5, 7
  setup_r5, 9

* support functions
  assert_points_input, 2
  check_connection, 3
  download_metadata, 6
  posix_to_string, 8
  select_mode, 8
  set_max_lts, 10
  set_max_rides, 11
  set_max_street_time, 11
  set_n_threads, 12
  set_speed, 13
  set_suboptimal_minutes, 13
  set_verbose, 14
  stop_r5, 15
  street_network_to_sf, 16

assert_points_input, 2, 3, 7–16
check_connection, 3, 3, 7–16
detailed_itineraries, 3, 19
download_metadata, 3, 6, 8–16
download_r5, 7, 9
posix_to_string, 3, 7, 8, 9–16
select_mode, 3, 7, 8, 8, 10–16
set_max_lts, 3, 7–9, 10, 11–16
set_max_rides, 3, 7–10, 11, 12–16
set_max_street_time, 3, 7–11, 11, 12–16
set_n_threads, 3, 7–12, 12, 13–16
set_speed, 3, 7–12, 13, 14–16
set_suboptimal_minutes, 3, 7–13, 13, 14–16
set_verbose, 3, 7–14, 14, 15, 16
setup_r5, 7, 9
stop_r5, 3, 7–14, 15, 16
street_network_to_sf, 3, 7–15, 16
travel_time_matrix, 6, 17