

# Package: rrstools (via r-universe)

June 8, 2026

**Title** Analyzing RoboCupRescue Simulation Data

**Version** 1.1.0

**Description** Tools for analyzing data from RoboCupRescue Simulation (RRS) <<https://rescuesim.robocup.org>>, a disaster rescue simulation platform. Supports reading virtual city map and disaster scenario files into analyzable data structures and provides functions for their visualization.

**License** MIT + file LICENSE

**URL** <https://nononoexe.github.io/rrstools/>

**Depends** R (>= 3.5)

**Imports** graphics, jsonlite, sf, stats, utils, xml2

**Suggests** spelling, testthat (>= 3.3.0), withr

**LinkingTo** cpp11

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 8.0.0

**Language** en-US

**NeedsCompilation** yes

**Author** Keisuke Ando [aut, cre] (ORCID:  
<<https://orcid.org/0009-0002-4085-2067>>)

**Maintainer** Keisuke Ando <nononoexe@ymail.ne.jp>

**Repository** <https://cran.r-universe.dev>

**Date/Publication** 2026-06-08 19:30:14 UTC

**RemoteUrl** <https://github.com/cran/rrstools>

**RemoteRef** HEAD

**RemoteSha** e7b3a745a76af3edd75f705d2a435bb1724d2498

## Contents

download_scenarios . . . . .	2
get_elements . . . . .	3
get_hub_url . . . . .	4
get_scenarios . . . . .	4
morph_road_network . . . . .	5
plot.rrs_map . . . . .	6
plot.rrs_scenario . . . . .	8
print.rrs_map . . . . .	8
print.rrs_scenario . . . . .	9
read_rrs_map . . . . .	10
read_rrs_scenario . . . . .	10
set_hub_url . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

download_scenarios	<i>Download RoboCupRescue Simulation scenarios</i>
--------------------	--

---

### Description

download\_scenarios downloads scenario archives for the RoboCupRescue Simulation from the RCRS Scenario Hub. It allows filtering by scenario name.

### Usage

```
download_scenarios(names, download_dir, extract = FALSE)
```

### Arguments

names	A character vector of scenario names to download (e.g., "kobe").
download_dir	A string specifying the directory where files will be downloaded. If the directory does not exist, it will be created.
extract	A logical value. If TRUE, downloaded archives will be extracted into the download_dir, and the original archive file will be removed upon successful extraction. Supported formats: .zip (extracted using <code>utils::unzip()</code> ), .7z (requires 7-zip command-line tool (7z) to be installed and available in the system's PATH). Defaults to FALSE.

### Value

Invisibly returns a character vector of the local paths corresponding to the results of the process. If extraction is successful, the path to the newly created data directory is returned; otherwise, the path to the downloaded archive file is returned. Returns an empty character vector if no files were downloaded.

**Examples**

```
# Download the kobe scenario
download_scenarios(names = "kobe", download_dir = tempdir())

# Download and extract the kobe scenario
download_scenarios(names = "kobe", download_dir = tempdir(), extract = TRUE)
```

---

`get_elements`*Get elements from a RoboCupRescue Simulation map*

---

**Description**

These functions provide methods to extract specific geographical elements from an `rrs_map` object. You can retrieve all elements of a certain type or filter them by their IDs.

**Usage**

```
get_nodes(map, ids = character())
get_edges(map, ids = character())
get_roads(map, ids = character())
get_buildings(map, ids = character())
get_entrances(map, ids = character())
get_passage(map, ids = character())
```

**Arguments**

<code>map</code>	An object of class <code>rrs_map</code> .
<code>ids</code>	A character vector of element IDs to filter by. If empty (default), all elements of the specified type are returned.

**Value**

An sf data frame containing the requested map elements.

**Examples**

```
gml <- system.file("extdata", "map-test.gml", package = "rrstools")
map <- read_rrs_map(gml)

# Get all nodes
get_nodes(map)
```

```
# Get a specific node by ID
get_nodes(map, c("0", "2"))
```

---

get_hub_url	<i>Get the URL of the RCRS Scenario Hub</i>
-------------	---

---

### Description

get\_hub\_url returns the current URL of the RCRS Scenario Hub.

### Usage

```
get_hub_url()
```

### Value

A string of the current URL of the RCRS Scenario Hub.

### Examples

```
get_hub_url()
```

---

get_scenarios	<i>Get available scenarios from the RCRS Scenario Hub</i>
---------------	---

---

### Description

get\_scenarios retrieves the list of available scenarios from the RCRS Scenario Hub and returns it as a data frame. The result is cached within the session to avoid repeated downloads. Use refresh = TRUE to force re-fetching from the Hub.

### Usage

```
get_scenarios(refresh = FALSE)
```

### Arguments

refresh      A logical value. If TRUE, the cached data is discarded and the scenario list is re-fetched from the Hub. Defaults to FALSE.

**Value**

A data frame with the following columns:

**name** Scenario name

**download\_url** URL to download the map scenario archive

**map** Map name

**city** Name of the city that map is modeled after

**country** Name of the country where the city is located

**lat** Latitude of the map location. If the exact location is unknown, the center of the city is used.

**lon** Longitude of the map location. If the exact location is unknown, the center of the city is used.

**thumbnail\_url** URL of the thumbnail image of the map

**Examples**

```
# Get available scenarios
get_scenarios()

# Force re-fetch from the Hub
get_scenarios(refresh = TRUE)
```

---

morph_road_network	<i>Morph raw map data into a road network</i>
--------------------	---

---

**Description**

morph\_road\_network() converts map data from the RoboCupRescue Simulation into a road network.

**Usage**

```
morph_road_network(map, mode = c("all", "roads_only", "roads_no_entrances"))
```

**Arguments**

**map** rrs\_map data, the GML map data loaded by the read\_rrs\_map().

**mode** A character string specifying the mode of the road network:

- "all": All areas are included in the network (default).
- "roads\_only": Only road areas are included.
- "roads\_no\_entrances": Only road areas are included, excluding entrances.

When mode is set to "roads\_only" or "roads\_no\_entrances", passages (connections between buildings) are excluded from the network to prevent the formation of isolated nodes.

## Details

This road network consists of nodes and edges. Nodes are defined as the centroids of all areas and the center points the boundaries between areas. Edges connect nodes between an agent can travel.

## Value

A list containing the network components:

- nodes: A sf object of nodes with point geometries.
- edges: A sf object of edges with linestring geometries.

## Examples

```
gml <- system.file("extdata", "map-test.gml", package = "rrstools")
map <- read_rrs_map(gml)
net <- morph_road_network(map)
net
```

---

plot.rrs\_map

*Plot a RoboCupRescue Simulation map*

---

## Description

plot visualizes a rrs\_map object, displaying the geographical features such as buildings and roads. If a rrs\_scenario object is provided, it also overlays the initial locations of rescue platoons, civilians, and special buildings from the scenario. It leverages the sf package's plotting capabilities to render the map components.

## Usage

```
## S3 method for class 'rrs_map'
plot(
  x,
  scenario = NULL,
  building_colour = "#f0e7d8",
  building_border = "#121212",
  road_colour = "#dbdbdb",
  road_border = "#121212",
  background_colour = NA,
  ...
)
```

**Arguments**

x	An object class rrs_map.
scenario	An object of class rrs_scenario. If provided, entities from the scenario will be plotted on the map. Defaults to NULL.
building_colour	The colour of the buildings.
building_border	The border colour of the buildings.
road_colour	The colour of the roads.
road_border	The border colour of the roads.
background_colour	The background colour of the plot.
...	Additional arguments passed to plot().

**Details**

plot first plots the basic map layout, including buildings and roads. Then, if a scenario is given, it plots the following entities:

- **Refuges:** Plotted in green (#00ff00), with the capacity displayed.
- **Fire Stations:** Plotted in yellow (ffff00).
- **Ambulance Centers:** Plotted in white (ffffff).
- **Police Offices:** Plotted in blue (#0000ff).
- **Civilians:** Plotted as green circles (#00ff00), with the number indicating the count at that location.
- **Fire Brigades:** Plotted as red circles (ff0000), with the number indicating the count at that location.
- **Ambulance Teams:** Plotted as white circles (ffffff), with the number indicating the count at that location.
- **Police Forces:** Plotted as blue circles (#0000ff), with the number indicating the count at that location.

The positions of human entities (agents and civilians) are slightly offset from the center of their location to avoid overlap.

**Value**

Invisibly returns NULL (called for side effects).

**Examples**

```
gml <- system.file("extdata", "map-test.gml", package = "rrstools")
map <- read_rrs_map(gml)

# Plot only the map
plot(map)
```

```
# Plot the map with a scenario
xml <- system.file("extdata", "scenario-test.xml", package = "rrstools")
scenario <- read_rrs_scenario(xml)
plot(map, scenario)
```

---

plot.rrs\_scenario      *Plot a RRS scenario object (Not supported directly)*

---

### Description

A `rrs_scenario` object cannot be plotted by itself because it lacks the necessary spatial context provided by a map. This function exists to intercept calls to `plot(scenario)` and provide a helpful error message.

### Usage

```
# Collect usage:
# plot(map, scenario)
```

### Arguments

<code>x</code>	An object class <code>rrs_scenario</code> .
<code>...</code>	Not used.

### Details

To visualize a scenario, you must plot it together with a `rrs_map` object.

### Value

Does not return; always throws an error.

---

print.rrs\_map      *Print a RoboCupRescue Simulation map object*

---

### Description

`print` prints a concise summary of a `rrs_map` object. It displays the object type, the number of nodes, edges, and faces it contains, its overall bounding box.

### Usage

```
## S3 method for class 'rrs_map'
print(x, ...)
```

### Arguments

x	An object class <code>rrs_map</code> .
...	Not used.

### Value

Invisibly returns `x` (called for side effects).

### Examples

```
gml <- system.file("extdata", "map-test.gml", package = "rrstools")
map <- read_rrs_map(gml)
map
```

---

`print.rrs_scenario`     *Print a RRS scenario object*

---

### Description

`print` prints a concise summary of a `rrs_scenario` object. It displays the object type, the total number of entities, and a table with the counts of entities for each type.

### Usage

```
## S3 method for class 'rrs_scenario'
print(x, ...)
```

### Arguments

x	An object class <code>rrs_scenario</code> .
...	Not used.

### Value

Invisibly returns `x` (called for side effects).

### Examples

```
xml <- system.file("extdata", "scenario-test.xml", package = "rrstools")
scenario <- read_rrs_scenario(xml)
scenario
```

---

read_rrs_map	<i>Read RoboCupRescue Simulation map data from GML file</i>
--------------	---

---

**Description**

This function reads and processes map data for RoboCupRescue Simulation from a GML file. It extracts nodes, edges, buildings and roads.

**Usage**

```
read_rrs_map(gml, scale_data = FALSE)
```

**Arguments**

gml	Path to the GML file.
scale_data	Logical. If TRUE, coordinates are scaled up by a factor of 1000 to match the simulation environment, and adjusted such that the minimum x and y values are 0. (Default: TRUE)

**Value**

A list of sf objects: nodes, edges, buildings, and roads.

**Examples**

```
gml <- system.file("extdata", "map-test.gml", package = "rrstools")
map <- read_rrs_map(gml)
map
```

---

read_rrs_scenario	<i>Read RoboCupRescue Simulation scenario data from XML file</i>
-------------------	--

---

**Description**

read\_rrs\_scenario() reads and processes scenario data for RoboCupRescue Simulation from a XML file.

**Usage**

```
read_rrs_scenario(xml)
```

**Arguments**

xml	Path to the XML file containing the scenario data.
-----	--

**Value**

A `rrs_scenario` object. This object is a `data.frame`, and contains the following columns:

- `type`: Character. The type of the entity (e.g., "refuge", "civilian").
- `location`: Character. The entity ID of the location where the entity is placed.
- `attributes`: List. A list containing other attributes of the entity.

**Examples**

```
xml <- system.file("extdata", "scenario-test.xml", package = "rrstools")
scenario <- read_rrs_scenario(xml)
scenario
```

---

`set_hub_url`*Set the URL of the RCRS Scenario Hub*

---

**Description**

`set_hub_url` sets the URL of the RCRS Scenario Hub for the current session. The cached scenario list is cleared when the URL is changed.

**Usage**

```
set_hub_url(url)
```

**Arguments**

`url` A string specifying the URL of the RCRS Scenario Hub.

**Value**

Invisibly returns the previous URL.

**Examples**

```
set_hub_url("https://example.com/scenarios.json")
```

# Index

[download\\_scenarios](#), 2

[get\\_buildings \(get\\_elements\)](#), 3

[get\\_edges \(get\\_elements\)](#), 3

[get\\_elements](#), 3

[get\\_entrances \(get\\_elements\)](#), 3

[get\\_hub\\_url](#), 4

[get\\_nodes \(get\\_elements\)](#), 3

[get\\_passage \(get\\_elements\)](#), 3

[get\\_roads \(get\\_elements\)](#), 3

[get\\_scenarios](#), 4

[morph\\_road\\_network](#), 5

[plot.rrs\\_map](#), 6

[plot.rrs\\_scenario](#), 8

[print.rrs\\_map](#), 8

[print.rrs\\_scenario](#), 9

[read\\_rrs\\_map](#), 10

[read\\_rrs\\_scenario](#), 10

[set\\_hub\\_url](#), 11

[utils::unzip\(\)](#), 2