

Package ‘escapeR’

September 27, 2026

Type Package

Title Escape Room Adventures for Learning R in Ecological Statistics

Version 0.1.0

Description A lightweight classroom game where students learn R by solving ecological-statistics puzzles inside a virtual escape room. The package remembers each player's progress, offers hints, and uses tasks inspired by introductory R teaching material, numerical ecology, ecological modelling, and distance sampling.

License GPL (>= 3)

URL <https://github.com/TiagoAMarques/escapeR>

BugReports <https://github.com/TiagoAMarques/escapeR/issues>

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.0.0)

Imports tools, utils

Suggests knitr, rmarkdown, testthat (>= 3.1.7), withr

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Tiago Marques [aut, cre],
OpenAI Codex [ctb]

Maintainer Tiago Marques <tiago.marques@st-andrews.ac.uk>

Repository CRAN

Date/Publication 2026-09-27 16:20:02 UTC

Contents

escapeR-package	2
build_escape	2

delete_progress	3
escape	4
escapeR_file	5
play	5
Index	7

escapeR-package	<i>Escape Room Adventures for Learning R in Ecological Statistics</i>
-----------------	---

Description

A lightweight classroom game where students learn R by solving ecological-statistics puzzles inside a virtual escape room.

Details

Call `escape()` in an interactive R session, solve each room with ordinary R commands, and use `submit()` to unlock the next task.

To learn how to play, see `vignette("getting-started-with-escapeR", package = "escapeR")`.
To create themed rooms or contributor room packs, see `vignette("creating-themed-escape-rooms", package = "escapeR")`.

build_escape	<i>Build and Register escapeR Rooms</i>
--------------	---

Description

Helpers for composing custom escape-room sequences and adding contributed rooms or room packs.

Usage

```
build_escape(rooms = .builtin_room_ids())

list_escapes()

new_room(id, module, title, learning_goal, introduction, challenge,
  hints, correct_result = NULL, checker = NULL, success, failure = NULL)

new_room_pack(id, title, description, rooms, escapes)

register_room_pack(pack, replace = FALSE)

register_rooms(..., replace = FALSE)
```

Arguments

rooms	Character vector of room IDs for <code>build_escape()</code> , or a list of room objects for <code>new_room_pack()</code> . Use <code>list_rooms()</code> to see bundled and registered rooms.
id	Unique room ID: a word of at most 8 characters.
module, title, learning_goal, introduction, challenge	Character strings describing the room.
description	Character string describing a room pack.
hints	Character vector of hints shown sequentially by <code>hint()</code> .
correct_result	Optional expected answer. Numeric and character results get helpful default checking.
checker	Optional function that takes a submitted answer and returns TRUE for success.
success	Message shown when the room is solved.
failure	Optional message shown when the submitted answer is wrong.
escapes	Named list where each element is a character vector of room IDs.
pack	Room pack created by <code>new_room_pack()</code> .
...	Room objects created by <code>new_room()</code> , or a list of room objects.
replace	Logical. If TRUE, replace a previously registered room with the same ID.

Value

`build_escape()` returns an escape sequence object for `escape()`. `list_escapes()` returns a data frame of registered sequences. `new_room()` returns a modular room object. `new_room_pack()` returns a shareable room pack. Registration helpers return IDs invisibly.

Examples

```
build_escape(c("console", "vector"))$room_ids
list_escapes()
room <- new_room(
  id = "meanroom", module = "Arithmetic", title = "A field sample",
  learning_goal = "Calculate a mean", introduction = "Three samples await.",
  challenge = "Submit the mean of 1, 2 and 3.", hints = "Use mean().",
  correct_result = 2, success = "The door opens."
)
room$id
```

delete_progress

Delete a Saved Player Profile

Description

Removes a saved profile and closes the active game if it uses that profile.

Usage

```
delete_progress(player = .state$player)
```

Arguments

player Single non-empty player name. Defaults to the active player.

Details

Progress is stored in the per-user directory returned by `tools::R_user_dir()`. Set `options(escapeR.progress_dir = path)` to select another directory. Delete profiles when no longer needed. Player filenames are case-insensitive and punctuation is replaced by underscores.

Value

Invisibly returns whether a saved file was removed.

Examples

```
local({
  path <- tempfile("escapeR-demo-")
  old <- options(escapeR.progress_dir = path)
  on.exit(options(old))
  on.exit(unlink(path, recursive = TRUE), add = TRUE)
  escape(player = "demo", reset = TRUE)
  submit(70)
  delete_progress("demo")
})
```

escape	<i>Start or Resume an escapeR Game</i>
--------	--

Description

Start a new game or resume saved progress for a returning player.

Usage

```
escape(player = NULL, reset = FALSE, escape = NULL)
```

Arguments

player	Character scalar identifying the player. If omitted interactively, escapeR asks for a name.
reset	Logical. If TRUE, restart the named player's game.
escape	Escape sequence created by <code>build_escape()</code> , or a character vector of room IDs. Defaults to the bundled room sequence.

Value

Invisibly returns the current progress list.

escapeR_file	<i>Helper Data Functions</i>
--------------	------------------------------

Description

Small helper functions used by the escape-room puzzles.

Usage

```
escapeR_file(filename)
```

```
survey_counts()
```

Arguments

filename	Name of a bundled file in inst/extdata.
----------	---

Value

escapeR_file() returns a file path, and survey_counts() returns a data frame.

Examples

```
head(survey_counts())  
read.csv(escapeR_file("survey_counts.csv"))
```

play	<i>Play the escapeR Game</i>
------	------------------------------

Description

Functions used by students to move through the virtual escape room.

Usage

```
play()

submit(answer)

hint()

status()

reset_game(player = NULL)

list_rooms()
```

Arguments

answer	The answer to check for the current room.
player	Character scalar. Defaults to the active player.

Value

`play()`, `hint()`, `status()`, and `reset_game()` return useful game objects invisibly. `submit()` invisibly returns whether the answer was correct. `list_rooms()` returns a data frame with room IDs, modules, titles, and learning goals.

Examples

```
head(list_rooms())
```

Index

`build_escape`, [2](#)

`delete_progress`, [3](#)

`escape`, [4](#)

`escapeR (escapeR-package)`, [2](#)

`escapeR-package`, [2](#)

`escapeR_file`, [5](#)

`hint (play)`, [5](#)

`list_escapes (build_escape)`, [2](#)

`list_rooms (play)`, [5](#)

`new_room (build_escape)`, [2](#)

`new_room_pack (build_escape)`, [2](#)

`play`, [5](#)

`register_room_pack (build_escape)`, [2](#)

`register_rooms (build_escape)`, [2](#)

`reset_game (play)`, [5](#)

`status (play)`, [5](#)

`submit (play)`, [5](#)

`survey_counts (escapeR_file)`, [5](#)