

# Package ‘TroublemakerR’

May 7, 2026

**Title** Generates Spatial Problems in R for 'AMPL'

**Version** 0.0.1

**Description** Provides methods for generating .dat files for use with the 'AMPL' software using spatial data, particularly rasters. It includes support for various spatial data formats and different problem types. By automating the process of generating 'AMPL' datasets, this package can help streamline optimization workflows and make it easier to solve complex optimization problems. The methods implemented in this package are described in detail in a publication by Fourer et al. (<[doi:10.1287/mnsc.36.5.519](https://doi.org/10.1287/mnsc.36.5.519)>).

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Depends** R (>= 4.1.0)

**LazyData** true

**Imports** purrr, terra

**URL** <https://github.com/Sustainscapes/TroublemakerR>

**BugReports** <https://github.com/Sustainscapes/TroublemakerR/issues>

**NeedsCompilation** no

**Author** Derek Corcoran [aut, cre]

**Maintainer** Derek Corcoran <[derek.corcoran.barrios@gmail.com](mailto:derek.corcoran.barrios@gmail.com)>

**Repository** CRAN

**Date/Publication** 2023-04-03 16:00:02 UTC

## Contents

create_budget . . . . .	2
Current . . . . .	3
CurrentLanduse . . . . .	3
define_cells . . . . .	4
landuse_names . . . . .	5

Species . . . . .	5
Species_Landuse . . . . .	6
species_names . . . . .	6
species_suitability . . . . .	7
species_suitability_landuse . . . . .	8
troublemaker . . . . .	9
write_ampl_lines . . . . .	11
write_cell_param . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

create_budget	<i>Create budget</i>
---------------	----------------------

---

## Description

This function generates or appends the budget and transition cost to a .dat file for ampl. The file will be written to the location specified by the name argument. If the file already exists, it will be overwritten. The file format is plain text, with each line terminated by a newline character.

## Usage

```
create_budget(
  budget,
  Rastercurrentlanduse,
  landuses,
  name = "Problem",
  verbose = FALSE
)
```

## Arguments

budget	maximum cost for the problem
Rastercurrentlanduse	raster object of current landuses
landuses	character vector with all landuses
name	The name of the output file
verbose	Logical whether messages will be written while the function is generating calculations, defaults to FALSE

## Value

A .dat file. This function is used for the side-effect of writing values to a file.

## Author(s)

Derek Corcoran

**Examples**

```

data(CurrentLanduse)
CurrentLU <- terra::unwrap(CurrentLanduse)

TroublemakerR::create_budget(budget = 2,
Rastercurrentlanduse = CurrentLU,
landuses = c("Agriculture", "Forest", "Urban"),
name = "Problem",
verbose = TRUE)

# delete the file so the test on cran can pass this

file.remove("Problem.dat")

```

---

Current	<i>A PackedSpatRaster of 4 species with its projected distribution for current conditions</i>
---------	---

---

**Description**

A PackedSpatRaster of 4 species with its projected distribution for current conditions

**Usage**

Current

**Format****A PackedSpatRaster with 4 layer::**

**Spp1** Predicted presence absence for species 1 in current conditions

**Spp2** Predicted presence absence for species 1 in current conditions

**Spp3** Predicted presence absence for species 1 in current conditions

**Spp4** Predicted presence absence for species 1 in current conditions

---

CurrentLanduse	<i>A PackedSpatRaster of the current landuse</i>
----------------	--

---

**Description**

A PackedSpatRaster of the current landuse

**Usage**

CurrentLanduse

**Format**

**A PackedSpatRaster with 1 layer::**

**Landuse** current landuse

---

define_cells	<i>Define Cells</i>
--------------	---------------------

---

**Description**

This function takes a Raster object and identifies non NA cells and writes them to a .dat file. The file will be written to the location specified by the name argument. If the file already exists, it will be overwritten. The file format is plain text, with each line terminated by a newline character.

**Usage**

```
define_cells(Rasterdomain, name = "Problem")
```

**Arguments**

Rasterdomain	A Raster object with any value in the cells that are part of the problem and NA values where the problem is not to be solved
name	The name of the output file

**Value**

.dat file. This function is used for the side-effect of writing values to a file.

**Author(s)**

Derek Corcoran

**Examples**

```
data(Species)
library(terra)
Test <- Species[[1]] |>
terra::unwrap()

# Generate the "Problem.dat" file

define_cells(Test[[1]])

file.remove("Problem.dat")
```

---

landuse_names	<i>Landuse names</i>
---------------	----------------------

---

**Description**

This function takes a vector of landuse names and writes them to a .dat file. The file will be written to the location specified by the name argument. If the file already exists, it will be overwritten. The file format is plain text, with each line terminated by a newline character.

**Usage**

```
landuse_names(landuses = NULL, name = "Problem")
```

**Arguments**

landuses	a vector with the names of the landuses
name	The name of the output file

**Value**

.dat file. This function is used for the side-effect of writing values to a file.

**Author(s)**

Derek Corcoran

**Examples**

```
landuse_names(landuses = c("Agriculture", "Forest", "Urban"))  
  
# delete the file so the test on cran can pass this  
  
file.remove("Problem.dat")
```

---

Species	<i>A list of 4 species with its projected distribution for 4 landuses</i>
---------	---

---

**Description**

A list of 4 species with its projected distribution for 4 landuses

**Usage**

```
Species
```

**Format**

**A list of 4 Spatrasters with 4 layers each::**

**Species 1** Predicted presence absence for species 1 in current, forest, agriculture, and urban landuse

**Species 2** Predicted presence absence for species 1 in current, forest, agriculture, and urban landuse

**Species 3** Predicted presence absence for species 1 in current, forest, agriculture, and urban landuse

**Species 4** Predicted presence absence for species 1 in current, forest, agriculture, and urban landuse

---

Species_Landuse	<i>A list of 4 species with its projected distribution for 4 landuses</i>
-----------------	---

---

**Description**

A list of 4 species with its projected distribution for 4 landuses

**Usage**

Species\_Landuse

**Format**

**A list of 4 Spatrasters with 4 layers each::**

**Species 1** Predicted presence absence for species 1 in forest, agriculture, and urban landuse

**Species 2** Predicted presence absence for species 1 in forest, agriculture, and urban landuse

**Species 3** Predicted presence absence for species 1 in forest, agriculture, and urban landuse

**Species 4** Predicted presence absence for species 1 in forest, agriculture, and urban landuse

---

species_names	<i>Species names</i>
---------------	----------------------

---

**Description**

This function takes a vector of species names and writes them to a .dat file. The file will be written to the location specified by the name argument. If the file already exists, it will be overwritten. The file format is plain text, with each line terminated by a newline character.

**Usage**

```
species_names(species_names = NULL, name = "Problem")
```

**Arguments**

species\_names a vector with the names of species  
name The name of the output file

**Value**

.dat file. This function is used for the side-effect of writing values to a file.

**Author(s)**

Derek Corcoran

**Examples**

```
species_names(species_names = c("Spp1", "Spp2"))  
file.remove("Problem.dat")
```

---

species\_suitability *Calculate species suitability*

---

**Description**

Calculate species suitability from a given raster and species names and writes them to a .dat file. The file will be written to the location specified by the name argument. If the file already exists, it will be overwritten. The file format is plain text, with each line terminated by a newline character.

**Usage**

```
species_suitability(  
  Rastercurrent,  
  species_names,  
  name = "Problem",  
  verbose = FALSE  
)
```

**Arguments**

Rastercurrent raster object of current suitability  
species\_names character vector of species names  
name The name of the output file  
verbose Logical whether messages will be written while the function is generating calculations, defaults to FALSE

**Value**

.dat file. This function is used for the side-effect of writing values to a file.

## Examples

```
library(terra)
data(Current)
Current <- terra::unwrap(Current)
species_suitability(Rastercurrent = Current, species_names = c("Spp1", "Spp2", "Spp3", "Spp4"))

file.remove("Problem.dat")
```

---

species\_suitability\_landuse

*Calculate species suitability for each landuse*

---

## Description

Calculate species suitability from a given raster, species names and landuse and writes them to a .dat file. The file will be written to the location specified by the name argument. If the file already exists, it will be overwritten. The file format is plain text, with each line terminated by a newline character.

## Usage

```
species_suitability_landuse(  
  Rasterspecieslanduse,  
  species_names,  
  landuses,  
  name = "Problem",  
  verbose = FALSE  
)
```

## Arguments

Rasterspecieslanduse	a list of species suitability for each landuse
species_names	character vector of species names
landuses	character vector with all landuses
name	The name of the output file
verbose	Logical whether messages will be written while the function is generating calculations, defaults to FALSE

## Value

.dat file. This function is used for the side-effect of writing values to a file.

**Examples**

```
library(terra)
data("Species_Landuse")
Species_Landuse <- Species_Landuse |> purrr::map(terra::unwrap)
species_suitability_landuse(Rasterspecieslanduse = Species_Landuse,
species_names = c("Spp1", "Spp2", "Spp3", "Spp4"),
landuses = c("Agriculture", "Forest", "Urban"), name = "Test")
file.remove("Test.dat")
```

troublemaker

*Troublemaker***Description**

This function is a metafunction with several functions inside of it it takes several spatial objects and generates a .dat file with a spatial dataset for AMPL

**Usage**

```
troublemaker(
  Rasterdomain = NULL,
  Rastercurrent = NULL,
  species_names = NULL,
  Rasterspecieslanduse = NULL,
  landuses = NULL,
  budget = NULL,
  Rastercurrentlanduse = NULL,
  name = "Problem",
  verbose = FALSE
)
```

**Arguments**

Rasterdomain	A Raster object with any value in the cells that are part of the problem and NA values where the problem is not to be solved
Rastercurrent	raster object of current suitability
species_names	a vector with the names of species
Rasterspecieslanduse	a list of species suitability for each landuse
landuses	character vector with all landuses
budget	maximum cost for the problem
Rastercurrentlanduse	raster object of current landuses
name	The name of the output file
verbose	Logical whether messages will be written while the function is generating calculations, defaults to FALSE

**Value**

A .dat file with the spatial problem formatted for AMPL. This function is used for the side-effect of writing values to a file.

**Author(s)**

Derek Corcoran

**Examples**

```
# Example 1 with current suitabilities
data(Species)
data(Current)
library(terra)
Test <- Species[[1]] |>
terra::unwrap()

Current <- terra::unwrap(Current)

# Generate the "Problem.dat" file

TroublemakerR::troublemaker(Rasterdomain =Test[[1]],
Rastercurrent = Current,
species_names = c("Spp1", "Spp2", "Spp3", "Spp4"),
name = "Problem")

# delete the file so the test on cran can pass this

file.remove("Problem.dat")

# Example 2 with landuse suitabilities

data(Species)
data("Species_Landuse")

library(terra)
Test <- Species[[1]] |>
terra::unwrap()

Species_Landuse <- Species_Landuse |> purrr::map(terra::unwrap)

# Generate the "Problem2.dat" file

TroublemakerR::troublemaker(Rasterdomain =Test[[1]],
Rasterspecieslanduse = Species_Landuse,
species_names = c("Spp1", "Spp2", "Spp3", "Spp4"),
landuses = c("Agriculture", "Forest", "Urban"),
name = "Problem2")

# delete the file so the test on cran can pass this

file.remove("Problem2.dat")
```

```
# Example 3 with budget and transition cost

data("CurrentLanduse")
CurrentLU <- terra::unwrap(CurrentLanduse)
TroublemakerR::troublemaker(Rasterdomain =Test[[1]],
Rasterspecieslanduse = Species_Landuse,
species_names = c("Spp1", "Spp2", "Spp3", "Spp4"),
landuses = c("Agriculture", "Forest", "Urban"),
Rastercurrentlanduse = CurrentLU,
budget = 2,
name = "Problem3",
verbose = FALSE)

file.remove("Problem3.dat")
```

---

write_ampl_lines	<i>Writes an AMPL line</i>
------------------	----------------------------

---

### Description

This function takes a character and writes them to a .dat file. The file will be written to the location specified by the name argument. If the file already exists, it will be overwritten. The file format is plain text, with each line terminated by a newline character.

### Usage

```
write_ampl_lines(line, name = "Problem")
```

### Arguments

line	line to be written to .dat file
name	The name of the output file

### Value

.dat file. This function is used for the side-effect of writing values to a file.

### Examples

```
write_ampl_lines("param s:= 1")

file.remove("Problem.dat")
```

---

write_cell_param	<i>Write cell parameters</i>
------------------	------------------------------

---

### Description

This function takes a Raster object, uses its values as a parameter and writes them to a .dat file. The file will be written to the location specified by the name argument. If the file already exists, it will be overwritten. The file format is plain text, with each line terminated by a newline character.

### Usage

```
write_cell_param(
  Rasterparam,
  parameter,
  default = NULL,
  name = "Problem",
  verbose = FALSE
)
```

### Arguments

Rasterparam	A Raster object with the values for the parameter
parameter	The name of the parameter to use
default	The value of the default value for the parameter if there is one, otherwise keep it as NULL
name	The name of the output file
verbose	Logical whether messages will be written while the function is generating calculations, defaults to FALSE

### Value

.dat file. This function is used for the side-effect of writing values to a file.

### Examples

```
library(terra)

A <- TroublemakerR::Current |> terra::unwrap()
A <- A[[1]]

write_cell_param(Rasterparam = A, parameter = "Suitability", name = "Problem")

write_cell_param(Rasterparam = A, parameter = "Carbon", default = 1,
  name = "Problem")

write_cell_param(Rasterparam = A, parameter = "Cost", default = 0,
  name = "Problem")
```

```
file.remove("Problem.dat")
```

# Index

## \* datasets

Current, [3](#)

CurrentLanduse, [3](#)

Species, [5](#)

Species\_Landuse, [6](#)

create\_budget, [2](#)

Current, [3](#)

CurrentLanduse, [3](#)

define\_cells, [4](#)

landuse\_names, [5](#)

Species, [5](#)

Species\_Landuse, [6](#)

species\_names, [6](#)

species\_suitability, [7](#)

species\_suitability\_landuse, [8](#)

troublemaker, [9](#)

write\_ampl\_lines, [11](#)

write\_cell\_param, [12](#)