

# Package ‘simodels’

July 23, 2024

**Title** Flexible Framework for Developing Spatial Interaction Models

**Version** 0.1.0

**Description** Develop spatial interaction models (SIMs). SIMs predict the amount of interaction, for example number of trips per day, between geographic entities representing trip origins and destinations. Contains functions for creating origin-destination datasets from geographic input datasets and calculating movement between origin-destination pairs with constrained, production-constrained, and attraction-constrained models (Wilson 1979) <doi:10.1068/a030001>.

**License** AGPL (>= 3)

**URL** <https://github.com/robinlovelace/simodels>,  
<https://robinlovelace.github.io/simodels/>

**BugReports** <https://github.com/robinlovelace/simodels/issues>

**Depends** R (>= 2.10)

**Imports** dplyr, geodist, od (>= 0.4.0), rlang, sf

**Suggests** ggplot2, knitr, minpack.lm, rmarkdown, tmap

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.1

**NeedsCompilation** no

**Author** Robin Lovelace [aut, cre] (<<https://orcid.org/0000-0001-5679-6536>>),  
Jakub Nowosad [aut] (<<https://orcid.org/0000-0002-1057-3721>>)

**Maintainer** Robin Lovelace <rob00x@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-07-23 14:40:02 UTC

## Contents

si_calculate	2
si_od_census	3
si_predict	4
si_pubs	5
si_to_od	5
si_zones	6
<b>Index</b>	<b>7</b>

---

si_calculate	<i>Calculate flow using a pre-existing function</i>
--------------	---

---

### Description

Executes a spatial interaction model based on an OD data frame and user-specified function

### Usage

```
si_calculate(
  od,
  fun,
  constraint_production,
  constraint_attraction,
  constraint_total,
  output_col = "interaction",
  ...
)
```

### Arguments

od	A data frame representing origin-destination data, e.g. as created by <a href="#">si_to_od()</a>
fun	A function that calculates the interaction (e.g. the number of trips) between each OD pair
constraint_production	Character representing column in od. This argument, when set, ensures that the outputs are 'production constrained': the total 'interaction' (e.g. n. trips) for all OD pairs is set such that the total for each zone of origin cannot go above this value.
constraint_attraction	Character representing column in od. This argument, when set, ensures that the outputs are 'attraction constrained': the total 'interaction' (e.g. n. trips) for all OD pairs is set such that the sum of trips to destination is equal to the mean value per destination.
constraint_total	Single number representing the total interaction. This argument, when set, ensures that the sum of the interaction calculated will equal the value given.

output_col	Character string containing the name of the new output column. "interaction" by default.
...	Arguments passed to fun

**Value**

An sf data frame

**Examples**

```

od = si_to_od(si_zones, si_zones, max_dist = 4000)
fun_dd = function(d = "distance_euclidean", beta = 0.3) exp(-beta * d / 1000)
fun_dd(d = (1:5)*1000)
od_dd = si_calculate(od, fun = fun_dd, d = distance_euclidean)
plot(od$distance_euclidean, od_dd$interaction)
fun = function(0, n, d, beta) 0 * n * exp(-beta * d / 1000)
od_output = si_calculate(od, fun = fun, beta = 0.3, 0 = origin_all,
  n = destination_all, d = distance_euclidean)
head(od_output)
plot(od$distance_euclidean, od_output$interaction)
od_pconst = si_calculate(od, fun = fun, beta = 0.3, 0 = origin_all,
  n = destination_all, d = distance_euclidean, constraint_production = origin_all)
plot(od_pconst$distance_euclidean, od_pconst$interaction)
plot(od_pconst["interaction"], logz = TRUE)
od_dd = si_calculate(od, fun = fun_dd, d = distance_euclidean, output_col = "res")
head(od_dd$res)
od_dd = si_calculate(od, fun = fun_dd, d = distance_euclidean, constraint_total = 10)
sum(od_dd$interaction)

```

---

si\_od\_census

*Example OD dataset*

---

**Description**

Example OD dataset from the 2011 UK Census

**Note**

Regenerate the data with scripts in the data-raw directory.

**Examples**

```
head(si_od_census)
```

---

 si\_predict
 

---



---

*Predict spatial interaction based on pre-trained model*


---

**Description**

Predict spatial interaction based on pre-trained model

**Usage**

```
si_predict(
  od,
  model,
  constraint_production,
  constraint_attraction,
  constraint_total,
  output_col = "interaction",
  ...
)
```

**Arguments**

od	A data frame representing origin-destination data, e.g. as created by <code>si_to_od()</code>
model	A model object, e.g. from <code>lm()</code> or <code>glm()</code>
constraint_production	Character representing column in od. This argument, when set, ensures that the outputs are 'production constrained': the total 'interaction' (e.g. n. trips) for all OD pairs is set such that the total for each zone of origin cannot go above this value.
constraint_attraction	Character representing column in od. This argument, when set, ensures that the outputs are 'attraction constrained': the total 'interaction' (e.g. n. trips) for all OD pairs is set such that the sum of trips to destination is equal to the mean value per destination.
constraint_total	Single number representing the total interaction. This argument, when set, ensures that the sum of the interaction calculated will equal the value given.
output_col	Character string containing the name of the new output column. "interaction" by default.
...	Arguments passed to fun

**Value**

An sf data frame

**See Also**

`si_calculate`

**Examples**

```
od = si_to_od(si_zones, si_zones, max_dist = 4000)
m = lm(od$origin_all ~ od$origin_bicycle)
od_updated = si_predict(od, m)
```

si\_pubs

*Example destinations dataset: pubs in Leeds***Description**

Example dataset from Leeds, UK

**Note**

Regenerate the data with scripts in the data-raw directory.

**Examples**

```
head(si_pubs)
```

si\_to\_od

*Prepare OD data frame***Description**

Prepares an OD data frame that next could be used to estimate movement between origins and destinations with a spatial interaction model.

**Usage**

```
si_to_od(origins, destinations, max_dist = Inf, intrazonal = TRUE)
```

**Arguments**

origins	sf object representing origin locations/zones
destinations	sf object representing destination locations/zones
max_dist	Euclidean distance in meters (numeric). Only OD pairs that are this distance apart or less will be returned and therefore included in the SIM.
intrazonal	Include intrazonal OD pairs? Intrazonal OD pairs represent movement from one place in a zone to another place in the same zone. TRUE by default.

## Details

In most origin-destination datasets the spatial entities that constitute origins (typically administrative zones) also represent destinations. In this 'unipartite' case origins and destinations should be passed the same object, an *sf* data frame representing administrative zones.

'Bipartite' datasets, by contrast, represent "spatial interaction systems where origins cannot act as destinations and vice versa" (Hasova et al. 2022).

a different *sf* object can be passed to the `destinations` argument.

## Value

An *sf* data frame

## Examples

```
library(sf)
origins = si_centroids[c(1, 2, 99), ]
destinations = origins
plot(origins$geometry)
odsf = si_to_od(origins, destinations, max_dist = 1200)
plot(odsf)
# note: result contains intrazonal flows represented by linestrings
# with a length of 0, e.g.:
sf::st_coordinates(odsf$geometry[1])
# With different destinations compared with origins
library(sf)
origins = si_centroids[c(2, 99), c(1, 6, 7)]
destinations = si_centroids[1, c(1, 6, 8)]
odsf = si_to_od(origins, destinations)
nrow(odsf) # no intrazonal flows
plot(odsf)
```

---

si\_zones

*Example zones and centroids*

---

## Description

`si_zones` and `si_centroids` represent administrative zones between which flows are to be estimated.

## Note

The schema data can be (re-)generated using code in `data-raw`

## Examples

```
si_zones
plot(si_zones$geometry)
plot(si_centroids$geometry, add = TRUE)
```

# Index

## \* datasets

si\_od\_census, 3

si\_pubs, 5

si\_zones, 6

glm(), 4

lm(), 4

si\_calculate, 2

si\_centroids (si\_zones), 6

si\_od\_census, 3

si\_predict, 4

si\_pubs, 5

si\_to\_od, 5

si\_to\_od(), 2, 4

si\_zones, 6