

# Package ‘sdc.redistribute’

June 18, 2026

**Title** Redistribute Values Between Geographic Areas

**Version** 0.1.0

**Description** Estimate attribute values for one set of polygons from values measured on a different, misaligned set. Provides area-weighted areal interpolation and a dasymetric method that distributes values across a point layer (such as parcel centroids). Count (extensive) measures are total-preserving; rate (intensive) measures use area-weighted means.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Language** en-US

**Depends** R (>= 4.1)

**Imports** sf

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, areal, spelling, covr

**Config/testthat/edition** 3

**RoxygenNote** 7.3.2

**URL** <https://dads2busy.github.io/sdc.redistribute/>,  
<https://github.com/dads2busy/sdc.redistribute>

**BugReports** <https://github.com/dads2busy/sdc.redistribute/issues>

**VignetteBuilder** knitr

**LazyData** true

**NeedsCompilation** no

**Author** Aaron Schroeder [aut, cre] (ORCID:  
<https://orcid.org/0000-0003-4372-2241>)

**Maintainer** Aaron Schroeder <ads7fg@virginia.edu>

**Repository** CRAN

**Date/Publication** 2026-06-18 13:30:02 UTC

## Contents

redistribute_direct . . . . .	2
redistribute_parcel . . . . .	3
sdc_example . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

redistribute_direct	<i>Area-weighted redistribution between polygon layers</i>
---------------------	--

---

### Description

Area-weighted redistribution between polygon layers

### Usage

```
redistribute_direct(
  source,
  target,
  extensive = NULL,
  intensive = NULL,
  preserve_totals = TRUE,
  suffix = NULL
)
```

### Arguments

source	An sf polygon layer carrying the values to redistribute.
target	An sf polygon layer to estimate values for.
extensive	Character vector of count column names in source to redistribute as totals (area-share weighted, optionally rescaled to preserve the source total).
intensive	Character vector of rate/density column names in source to redistribute as area-weighted means.
preserve_totals	Logical; if TRUE (default) extensive results are rescaled so each target column sums to the source total.
suffix	Optional string appended to each new column name.

### Details

Extensive measures (counts) are redistributed by each intersection's share of the source polygon area and, when `preserve_totals = TRUE`, rescaled so the target totals match the source totals. Intensive measures (rates/densities) are area-weighted means: the sum of each source value times the intersection's share of the target polygon area (the standard areal-weighting intensive estimator). This equals a true area-weighted mean when the target is fully covered by the source and treats any uncovered part of a target as contributing zero. NA source values are omitted from the weighted

sums. Targets that no source polygon covers receive 0 for extensive measures and NA for intensive measures. If target already has a column named like a redistributed measure, it is overwritten; pass `suffix` to keep both.

### Value

The target layer (an sf object) with one new column per redistributed measure.

### Examples

```
src <- sf::st_sf(pop = 100, geometry = sf::st_sfc(
  sf::st_polygon(list(rbind(c(0,0), c(2,0), c(2,2), c(0,2), c(0,0))))),
  crs = 3857))
tgt <- sf::st_sf(id = c("A", "B"), geometry = sf::st_sfc(
  sf::st_polygon(list(rbind(c(0,0), c(1,0), c(1,2), c(0,2), c(0,0))))),
  sf::st_polygon(list(rbind(c(1,0), c(2,0), c(2,2), c(1,2), c(1,0))))),
  crs = 3857))
redistribute_direct(src, tgt, extensive = "pop")
```

---

redistribute\_parcel *Dasymetric redistribution via a point layer*

---

### Description

Distributes each source value across the points (e.g. parcel centroids) that fall inside it, then reaggregates the point-level values to target polygons. With `weights = NULL` the value is split evenly across points; otherwise it is split in proportion to a points column (the extension point for household-size or unit-count weighting).

### Usage

```
redistribute_parcel(
  source,
  target,
  points,
  extensive = NULL,
  weights = NULL,
  suffix = NULL
)
```

### Arguments

<code>source</code>	An sf polygon layer carrying the values to redistribute.
<code>target</code>	An sf polygon layer to estimate values for.
<code>points</code>	An sf point layer (e.g. parcel centroids).
<code>extensive</code>	Character vector of count column names in source.
<code>weights</code>	Optional name of a numeric column in points to weight by.
<code>suffix</code>	Optional string appended to each new column name.

**Details**

Each source value is split across the points inside that source polygon in proportion to weights (equally when weights = NULL), then summed within each target polygon. A source polygon that contains no points contributes nothing to any target (its value cannot be placed). If the total weight of a source's points is zero, that source likewise contributes nothing. If target already has a column named like a redistributed measure, it is overwritten; pass `suffix` to keep both.

**Value**

The target layer (an `sf` object) with one new column per measure.

**Examples**

```
src <- sf::st_sf(pop = 100, geometry = sf::st_sfc(
  sf::st_polygon(list(rbind(c(0,0), c(2,0), c(2,2), c(0,2), c(0,0)))),
  crs = 3857))
tgt <- sf::st_sf(id = c("A", "B"), geometry = sf::st_sfc(
  sf::st_polygon(list(rbind(c(0,0), c(1,0), c(1,2), c(0,2), c(0,0)))),
  sf::st_polygon(list(rbind(c(1,0), c(2,0), c(2,2), c(1,2), c(1,0)))),
  crs = 3857))
pts <- sf::st_sf(geometry = sf::st_sfc(
  sf::st_point(c(0.5, 1)), sf::st_point(c(1.5, 1)), crs = 3857))
redistribute_parcel(s, tgt, pts, extensive = "pop")
```

---

sdc\_example

*Synthetic example geographies*


---

**Description**

A small, self-contained set of `sf` layers used in examples and vignettes.

**Usage**

```
sdc_example
```

**Format**

A named list with three `sf` elements:

**source** Two source polygons (tract, pop).

**target** Three target polygons (nbhd).

**parcels** Parcel centroid points (units).

# Index

## \* datasets

sdc\_example, 4

redistribute\_direct, 2

redistribute\_parcel, 3

sdc\_example, 4