

# Package ‘remap’

July 23, 2025

**Type** Package

**Title** Regional Spatial Modeling with Continuous Borders

**Version** 0.3.2

**Description** Automatically creates separate regression models for different spatial regions. The prediction surface is smoothed using a regional border smoothing method. If regional models are continuous, the resulting prediction surface is continuous across the spatial dimensions, even at region borders. Methodology is described in Wagstaff and Bean (2023) <[doi:10.32614/RJ-2023-004](https://doi.org/10.32614/RJ-2023-004)>.

**License** GPL-3

**URL** <https://github.com/jadonwagstaff/remap>

**BugReports** <https://github.com/jadonwagstaff/remap/issues>

**Encoding** UTF-8

**LazyData** true

**Imports** graphics (>= 4.1.0), methods (>= 4.1.0), parallel (>= 4.1.0),  
sf (>= 1.0.0), stats (>= 4.1.0), units (>= 0.6.7), utils (>= 4.1.0)

**RoxygenNote** 7.3.2

**Suggests** dplyr (>= 1.0.2), ggplot2 (>= 3.3.2), knitr (>= 1.30), maps (>= 3.3.0), mgcv (>= 1.8.33), rmarkdown (>= 2.5)

**VignetteBuilder** knitr

**Depends** R (>= 4.1.0)

**NeedsCompilation** no

**Author** Jadon Wagstaff [aut, cre],  
Brennan Bean [aut]

**Maintainer** Jadon Wagstaff <[jadonw@gmail.com](mailto:jadonw@gmail.com)>

**Repository** CRAN

**Date/Publication** 2025-01-09 23:00:02 UTC

## Contents

plot.remap . . . . .	2
predict.remap . . . . .	2
print.remap . . . . .	4
redist . . . . .	4
remap . . . . .	5
summary.remap . . . . .	7
utsnow . . . . .	8
utws . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

plot.remap	<i>Plot method for remap object.</i>
------------	--------------------------------------

---

### Description

Plots the regions used for modeling.

### Usage

```
## S3 method for class 'remap'
plot(x, ...)
```

### Arguments

x	S3 object output from remap.
...	Arguments to pass to regions plot.

### Value

A list that plots a map of the regions used for modeling.

---

predict.remap	<i>Make predictions given a set of data and smooths predictions at region borders. If an observation is outside of all regions and smoothing distances, the closest region will be used to predict.</i>
---------------	---

---

### Description

Make predictions given a set of data and smooths predictions at region borders. If an observation is outside of all regions and smoothing distances, the closest region will be used to predict.

**Usage**

```
## S3 method for class 'remap'  
predict(  
  object,  
  data,  
  smooth,  
  distances,  
  cores = 1,  
  progress = FALSE,  
  se = FALSE,  
  ...  
)
```

**Arguments**

object	S3 object output from remap.
data	An sf dataframe with point geometry.
smooth	The distance in km within a region where a smooth transition to the next region starts. If smooth = 0, no smoothing occurs between regions unless an observation falls on the border of two or more polygons. (Can be a named vector with different values for each unique object\$region_id in 'object\$region'.)
distances	An optional matrix of distances between 'data' and 'object\$regions' generated by redist() function (calculated internally if not provided).
cores	Number of cores for parallel computing. 'cores' above default of 1 will require more memory.
progress	If TRUE, a text progress bar is printed to the console. (Progress bar only appears if 'cores' = 1.)
se	If TRUE, predicted values are assumed to be standard errors and an upper bound of combined model standard errors are calculated at each prediction location. Should stay FALSE unless predicted values from remap are standard error values.
...	Arguments to pass to individual model prediction functions.

**Value**

Predictions in the form of a numeric vector. If se is TRUE, upper bound for combined standard errors in the form of a numeric vector.

**See Also**

[remap](#) building a regional model.

---

print.remap	<i>Print method for remap object.</i>
-------------	---------------------------------------

---

**Description**

Print method for remap object.

**Usage**

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

**Arguments**

x	S3 object output from remap.
...	Extra arguments.

**Value**

No return value, a description of the remap object is printed in the console.

---

redist	<i>Get distances between data and regions.</i>
--------	--

---

**Description**

Finds distances in km between data provided as sf dataframe with point geometry and regions provided as sf dataframe with polygon or multipolygon geometry.

**Usage**

```
redist(data, regions, region_id, max_dist, cores = 1, progress = FALSE)
```

**Arguments**

data	An sf data frame with point geometry.
regions	An sf dataframe with polygon or multipolygon geometry.
region_id	Optional name of column in 'regions' that contains the id that each region belongs to (no quotes). If null, it will be assumed that each row is its own region.
max_dist	a maximum distance that is needed for future calculations. (Set equal to maximum 'smooth' when predicting on new observations.)
cores	Number of cores for parallel computing. 'cores' above default of 1 will require more memory. (Progress bar only appears if 'cores' = 1.)
progress	If true, a text progress bar is printed to the console. Progress set to FALSE will find distances quicker if max_dist is not specified.

**Value**

A matrix where each row corresponds one-to-one with each row in provided 'data'. Matrix columns are either named with regions from 'region\_id' column of 'regions' or the row numbers of 'regions' if 'region\_id' is missing. Values are in kilometers.

**See Also**

[remap](#) - uses `redist` for regional models.

**Examples**

```
library(remap)
data(utsnow)
data(utws)

# Build a matrix of distances between objects of utsnow and utws
# We will not set max_dist, so all distances will be found
dists <- redist(
  data = utsnow,
  regions = utws,
  region_id = HUC2
)

head(dists)
```

---

remap

*Build separate models for mapping multiple regions.*

---

**Description**

Separate models are built for each given region and combined into one S3 object that can be used to predict on new data using generic function `predict()`.

**Usage**

```
remap(
  data,
  regions,
  region_id,
  model_function,
  buffer,
  min_n = 1,
  distances,
  cores = 1,
  progress = FALSE,
  ...
)
```

**Arguments**

<code>data</code>	An sf data frame with point geometry.
<code>regions</code>	An sf dataframe with polygon or multipolygon geometry.
<code>region_id</code>	Optional name of column in 'regions' that contains the id that each region belongs to (no quotes). If null, it will be assumed that each row of 'regions' is its own region.
<code>model_function</code>	A function that can take a subset of 'data' and output a model that can be used to predict new values when passed to generic function <code>predict()</code> .
<code>buffer</code>	The length of the buffer zone around each region in km where observations are included in the data used to build models for each region. (Can be a named vector with different values for each unique 'region_id' in 'region'.)
<code>min_n</code>	The minimum number of observations to use when building a model. If there are not enough observations in the region and buffer, then the closest <code>min_n</code> observations are used. <code>min_n</code> must be at least 1.
<code>distances</code>	An optional matrix of distances between 'data' and 'regions' generated by <code>redist()</code> function (calculated internally if not provided). Note that unless you know that you have <code>min_n</code> within a certain distance, no <code>max_dist</code> parameter should be used in <code>redist()</code> .
<code>cores</code>	Number of cores for parallel computing. 'cores' above default of 1 will require more memory.
<code>progress</code>	If true, a text progress bar is printed to the console. (Progress bar only appears if 'cores' = 1.)
<code>...</code>	Extra arguments to pass to 'model_function' function.

**Details**

If a model fails for a region, a warning is given but the modeling process will continue.

A description of the methodology can be found in Wagstaff and Bean (2023) "remap: Regionalized Models with Spatially Smooth Predictions" <doi:10.32614/RJ-2023-004>.

**Value**

A *remap* S3 object containing:

***models*** A list of models containing a model output by 'model\_function' for each region.

***regions*** 'regions' object passed to the function (used for prediction). The first column is 'region\_id' or the row number of 'regions' if 'region\_id' is missing. The second column is the region geometry.

***call*** Shows the parameters that were passed to the function.

**See Also**

[predict.remap](#) - used for predicting on new data. [redist](#) - used for pre-computing distances.

**Examples**

```

library(remap)
data(utsnow)
data(utws)

# We will keep these examples simple by only modeling non-zero values of
# snow water equivalent (WESD)

utsnz <- utsnow[utsnow$WESD > 0, ]

# Build a remap model with lm that has formula WESD ~ ELEVATION
# The buffer to collect data around each region is 30km
# The minimum number of observations per region is 10
remap_model <- remap(
  data = utsnz,
  regions = utws,
  region_id = HUC2,
  model_function = lm,
  formula = log(WESD) ~ ELEVATION,
  buffer = 20,
  min_n = 10,
  progress = TRUE
)

# Resubstitution predictions
remap_preds <- exp(predict(remap_model, utsnz, smooth = 10))
head(remap_preds)

```

---

summary.remap

*Summary method for remap object.*


---

**Description**

Summary method for remap object.

**Usage**

```

## S3 method for class 'remap'
summary(object, ...)

```

**Arguments**

```

object      S3 object output from remap.
...         Extra arguments to pass to regional models.

```

**Value**

No return value, a brief summary of the remap object is printed in the console. This includes the class(es) of the regional models, the CRS of the regions, and the bounding box of the regions.

---

utsnow

*Snowpack at weather stations in Utah on April 1st, 2011.*

---

### Description

Water equivalent of snow density (WESD) in mm of water at various location within and surrounding the state of Utah. WESD are measured at weather stations within the Daily Global Historical Climatology Network. April first measurements are used to estimate snowpack for the state of Utah.

### Usage

utsnow

### Format

An sf points object with 394 rows and 8 variables:

**ID** Weather station identification code.

**STATION\_NAME** Weather station name.

**LATITUDE** Latitude of weather station.

**LONGITUDE** Longitude of weather station.

**ELEVATION** Elevation of weather station.

**HUC2** Largest watershed region containing this weather station (see [utws](#) data).

**WESD** Water equivalent of snow density in mm of water.

**geometry** sfc points in geographic coordinates.

### Source

<https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/>

### Examples

```
# If you run into issues with loading this dataset try running:
utsnow <- sf::st_read(system.file("extdata/utsnow.shp", package="remap"))
names(utsnow) <- c(
  "ID", "STATION_NAME", "LATITUDE", "LONGITUDE", "ELEVATION", "HUC2",
  "WESD", "geometry"
)
```



---

utws

*Watershed polygons within the state of Utah.*

---

### Description

Watersheds are defined by the United States Geological Survey. Only the largest defines watersheds are used.

### Usage

utws

### Format

An sf object with 4 rows and 2 variables:

**HUC2** Largest watershed ID's defined by the USGS.

**geometry** sfc multipolygon object in geographic coordinates.

### Source

<https://www.usgs.gov/national-hydrography/watershed-boundary-dataset>

### Examples

```
# If you run into issues with loading this dataset try running:
```

```
utws <- sf::st_read(system.file("extdata/utws.shp", package="remap"))
```

# Index

## \* datasets

utsnow, 8

utws, 9

plot.remap, 2

predict.remap, 2, 6

print.remap, 4

redist, 4, 6

remap, 3, 5, 5

summary.remap, 7

utsnow, 8

utws, 8, 9