

# Package: prepr (via r-universe)

October 10, 2026

**Type** Package

**Title** Automatic Repair of Spatial Polygons

**Version** 0.3.1

**Description** Automatically repair broken spatial polygons using constrained triangulation. The computational methodology is derived from Ledoux et al. (2014)  
<[doi:10.1016/j.cageo.2014.01.009](https://doi.org/10.1016/j.cageo.2014.01.009)>.

**Imports** Rcpp (>= 1.0.12), sf (>= 1.0.8)

**Suggests** testthat (>= 2.0.1), knitr (>= 1.2.0), roxygen2 (>= 6.1.1),  
rmarkdown (>= 1.10),

**Depends** R (>= 4.2.0)

**LinkingTo** Rcpp (>= 1.0.12), BH (>= 1.71.0.0), sf (>= 1.0.8), RcppCGAL  
(>= 5.6.2), RcppEigen (>= 0.3.4.0.0)

**SystemRequirements** gmp (>= 4.2.3), mpfr (>= 3.0.0) gdal (>= 2.0.1)

**License** GPL-3

**Encoding** UTF-8

**URL** <https://github.com/prioritizr/prepr>

**BugReports** <https://github.com/prioritizr/prepr/issues>

**VignetteBuilder** knitr

**RoxygenNote** 7.3.1

**Collate** 'RcppExports.R' 'package.R' 'utils.R' 'st\_prepar.R'

**Roxygen** list(markdown = TRUE)

**Config/testthat/edition** 3

**Config/pak/sysreqs** libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libgmp3-dev libmpfr-  
dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

**Repository** <https://prioritizr.r-universe.dev>

**Date/Publication** 2024-12-13 19:02:32 UTC

**RemoteUrl** <https://github.com/prioritizr/prepr>

**RemoteRef** debug-autoconf

**RemoteSha** dbfcd69dec76388d843d4328b407963d68197cd4

## Contents

|                         |          |
|-------------------------|----------|
| prepr-package . . . . . | 2        |
| st_prepair . . . . .    | 3        |
| <b>Index</b>            | <b>5</b> |

---

|               |                                                              |
|---------------|--------------------------------------------------------------|
| prepr-package | <i>A package for automatically repairing broken polygons</i> |
|---------------|--------------------------------------------------------------|

---

## Description

This package contains functions to repair single polygons according to the international standards ISO 19107 using a constrained triangulation approach. It is based on the prepair C++ tool.

## Author(s)

**Maintainer:** Jeffrey O Hanson <jeffrey.hanson@uqconnect.edu.au> ([ORCID](#))

Authors:

- Ahmadou Dicko <mail@ahmadoudicko.com>

Other contributors:

- Edzer Pebesma <edzer.pebesma@uni-muenster.de> ([ORCID](#)) [contributor]
- Ken Arroyo Ohori <g.a.k.arroyoohori@tudelft.nl> (<https://github.com/tudelft3d/prepair>) [contributor, copyright holder]
- Hugo Ledoux <h.ledoux@tudelft.nl> (<https://github.com/tudelft3d/prepair>) [contributor, copyright holder]
- Martijn Meijers <b.m.meijers@tudelft.nl> (<https://github.com/tudelft3d/prepair>) [contributor, copyright holder]

## See Also

Useful links:

- <https://github.com/prioritizr/prepr>
- Report bugs at <https://github.com/prioritizr/prepr/issues>

---

st\_prepairAutomatic repair of polygon geometries

---

## Description

Repair polygon geometries according to the international standards ISO 19107 using a constrained triangulation approach (van Oosterom et al. 2005; Ledoux et al. 2014)

## Usage

```
st_prepair(x)
```

## Arguments

`x` `sf::st_sf()`, `sf::st_sfc()` or `sfg` object (containing POLYGON or MULTIPOLYGON geometries).

## Details

The function supports two algorithms:

- `oddeven`: an extension of the odd-even algorithm to handle polygons containing inner rings and degeneracies;
- `setdiff`: one where we follow a point set difference rule for the rings (outer - inner).

## Value

A `sf::st_sf()`, `sf::st_sfc()` or `sfg` object (same as the argument to `x`).

## References

Ledoux H, Arroyo Ohori K, and Meijers M (2014) A triangulation-based approach to automatically repair GIS polygons. *Computers & Geosciences* 66:121–131.

van Oosterom P, Quak W, and Tijssen T (2005) *About Invalid, Valid and Clean Polygons* In: Developments in Spatial Data Handling. Springer, Berlin, Heidelberg

## See Also

See `sf::st_make_valid()` for another approach to repair polygon geometries.

## Examples

```
## Not run:  
# create an object containing a broken polygon geometry  
x <- sf::st_as_sfc("POLYGON((0 0, 0 10, 10 0, 10 10, 0 0))")  
  
# check if this polygon is indeed broken  
sf::st_is_valid(x)
```

```
# repair the polygon
y <- st_repair(x)

# check that the repaired polygon has been fixed
print(st_is_valid(y))

## End(Not run)
```

# Index

prepr (prepr-package), [2](#)  
prepr-package, [2](#)

sf::st\_make\_valid(), [3](#)  
sf::st\_sf(), [3](#)  
sf::st\_sfc(), [3](#)  
sfg, [3](#)  
st\_prepair, [3](#)