

# Package ‘sfcentral’

March 12, 2024

**Title** Spatial Centrality and Dispersion Statistics

**Version** 0.1.0

**Date** 2024-03-11

**Maintainer** Gabriel V. Gaona <gabo@gavg712.com>

**Description** Computing centographic statistics (central points, standard distance, standard deviation ellipse, standard deviation box) for observations taken at point locations in 2D or 3D. The 'sfcentral' library was inspired in 'aspace' package but conceived to be used in a spatial 'tidyverse' context.

**URL** <https://gavg712.gitlab.io/sfcentral/>,  
<https://gitlab.com/gavg712/sfcentral>

**BugReports** <https://gitlab.com/gavg712/sfcentral/-/issues>

**Language** en-US

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Imports** geodist (>= 0.0.7), Hmisc (>= 4.6.0), lwgeom (>= 0.2.0),  
scales (>= 1.2.0), sf (>= 1.0.8), stats

**Suggests** ggplot2 (>= 3.3.6), testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Gabriel V. Gaona [aut, cre] (<<https://orcid.org/0000-0003-4666-2248>>)

**Repository** CRAN

**Date/Publication** 2024-03-12 19:50:02 UTC

## R topics documented:

st_central_point . . . . .	2
st_sd_box . . . . .	4
st_sd_distance . . . . .	5
st_sd_ellipse . . . . .	6

---

st_central_point	<i>Spatial centrality</i>
------------------	---------------------------

---

### Description

Functions to find spatial measures of gravity centers.

### Usage

```
st_central_point(.x, .y, ...)
```

```
## S3 method for class 'sfg'
```

```
st_central_point(
  .x,
  .y = NULL,
  weights = NULL,
  method = c("mean", "median", "geometric", "feature", "min.dist"),
  ...
)
```

```
## S3 method for class 'sf'
```

```
st_central_point(
  .x,
  .y = NULL,
  weights = NULL,
  method = c("mean", "median", "geometric", "feature", "min.dist"),
  ...
)
```

```
## S3 method for class 'sfc'
```

```
st_central_point(
  .x,
  .y = NULL,
  weights = NULL,
  method = c("mean", "median", "geometric", "feature", "min.dist"),
  ...
)
```

### Arguments

.x, .y	sf points 2D or 3D
...	arguments to be passed to or from other methods
weights	Numeric. Used in for weighed Mean Center. Has to be same length as number of points.
method	Character. Type of center point to calculate

`dist` Atomic numeric, Default 100. Starting distance value for center moving during iterations.

## Details

Spatial centers are spatial measures of the gravity center.

methods options are: "mean" is the mean center (equivalent to centroid of the points) calculated by the arithmetic mean of each axis; "geometric", is the corresponding geometric mean of each axis; "median", is the median center, a pair of `c(median(x), median(y))` coordinates; "feature", is a minimization of the sum of distances from *ith* point to every point; "min.dist", is iterative looking for the closest point in `bbox` of `.x` that minimizes the sum of distances from *ith* point to every point in the dataset

## Value

"Simple Features" of length 1.

## Note

Inspired on `aspace::*( )` from Ron Buliung & Randy Bui (2012)

## Author(s)

Gabriel Gaona

## Examples

```
requireNamespace("ggplot2", quietly = TRUE)
library(sf, quietly = TRUE)
library(ggplot2)
bbx <- matrix(c(697047,9553483,
               696158,9560476,
               700964,9561425,
               701745,9555358),
             byrow = TRUE,
             ncol = 2)
bbx <- st_multipoint(bbx)
bbx <- st_cast(bbx,"POLYGON")
bbx <- st_sfc(bbx, crs = 31992)
set.seed(1234)
points <- st_sf(geometry = st_sample(bbx, 100))
mean_center <- st_central_point(points, method = "mean")
median_center <- st_central_point(points, method = "median")
geom_center <- st_central_point(points, method = "geometric")
central_feature <- st_central_point(points, method = "feature")
min_dist_center <- st_central_point(points, method = "min.dist")
ggplot() +
  geom_sf(data = points, color = "steelblue", size = 0.5) +
  geom_sf(data = mean_center, color = "blue", size = 3) +
  geom_sf(data = median_center, color = "red") +
  geom_sf(data = geom_center, color = "grey80") +
```

```
geom_sf(data = central_feature, color = "orange") +
geom_sf(data = min_dist_center, color = "green")
```

---

st\_sd\_box

*Standard deviation box calculator in 2D or 3D*


---

## Description

Calculate the spatial deviation box from a points sf dataset. #' @author Gabriel Gaona

## Usage

```
st_sd_box(.x, centre = NULL, weights = NULL, ...)

## S3 method for class 'sfg'
st_sd_box(.x, centre = NULL, weights = NULL, ...)

## S3 method for class 'sf'
st_sd_box(.x, centre = NULL, weights = NULL, ...)

## S3 method for class 'sfc'
st_sd_box(.x, centre = NULL, weights = NULL, ...)
```

## Arguments

.x	sf points 2D or 3D
centre	Numeric. Coordinates 2D or 3D of central point. Default NULL, performs a calculation of mean_centre() from point localities
weights	Numeric. Same length of number of .x.
...	ignored

## Value

Depends on input, "coords" returns a data.frame of 2 or 3 columns and 4 or 8 point coordinates. "param" returns a data.frame with centre coordinates, standard deviation in each axis, space(area for 2D, volume for 3D) and number of dimensions in coordinates.

## Examples

```
requireNamespace("ggplot2", quietly = TRUE)
library(sf, quietly = TRUE)
library(ggplot2)
bbx <- matrix(c(697047,9553483,
               696158,9560476,
               700964,9561425,
               701745,9555358),
              byrow = TRUE,
              ncol = 2)
```

```

bbx <- st_multipoint(bbx)
bbx <- st_cast(bbx,"POLYGON")
bbx <- st_sfc(bbx, crs = 31992)
set.seed(1234)
points <- st_sf(geometry = st_sample(bbx, 100))
SD_BOX <- st_sd_box(points)
ggplot() +
  geom_sf(data = SD_BOX, fill = NA, color = "darkolivegreen") +
  geom_sf(data = points, color = "steelblue", size = 0.5)

```

---

st\_sd\_distance

*Standard deviation distance calculator*


---

## Description

Calculate the spatial deviation distance from a points sf dataset.

## Usage

```
st_sd_distance(.x, centre = NULL, weights = NULL, ...)
```

```
## S3 method for class 'sfg'
```

```
st_sd_distance(.x, centre = NULL, weights = NULL, ...)
```

```
## S3 method for class 'sf'
```

```
st_sd_distance(.x, centre = NULL, weights = NULL, ...)
```

```
## S3 method for class 'sfc'
```

```
st_sd_distance(.x, centre = NULL, weights = NULL, ...)
```

## Arguments

<code>.x</code>	<code>sf</code> points 2D or 3D
<code>centre</code>	One central point of class <code>sf</code> , <code>sfc</code> , <code>numeric</code> (length 2), <code>matrix</code> (2 col, 1 row), <code>data.frame</code> (2 col, 1 row), or <code>list</code> (length 2). Default NULL, means a calculation of the <code>st_central_point()</code> from <code>.x</code> localities.
<code>weights</code>	Numeric. Same length as number of points in <code>.x</code> .
<code>...</code>	other parameters for <code>sf::st_distance()</code>

## Value

A `sf` "POLYGON" with attributes:

- radius (standard deviation distance)
- area surrounding,
- perimeter,
- center coordinates,
- weighted indicator if weights were used or not in the calculator.

**Author(s)**

Gabriel Gaona

**Examples**

```

requireNamespace("ggplot2", quietly = TRUE)
library(sf, quietly = TRUE)
library(ggplot2)
bbx <- matrix(c(697047,9553483,
               696158,9560476,
               700964,9561425,
               701745,9555358),
             byrow = TRUE,
             ncol = 2)
bbx <- st_multipoint(bbx)
bbx <- st_cast(bbx,"POLYGON")
bbx <- st_sfc(bbx, crs = 31992)
set.seed(1234)
points <- st_sf(geometry = st_sample(bbx, 100))
SDD <- st_sd_distance(points)
ggplot() +
  geom_sf(data = SDD, fill = NA, color = "darkolivegreen") +
  geom_sf(data = points, color = "steelblue", size = 0.5)

```

---

st\_sd\_ellipse

*Standard deviation ellipse calculator*


---

**Description**

Calculate the spatial deviation ellipse from a points sf dataset.

**Usage**

```

st_sd_ellipse(.x, centre = NULL, weights = NULL, ...)

## S3 method for class 'sfg'
st_sd_ellipse(.x, centre = NULL, weights = NULL, ...)

## S3 method for class 'sf'
st_sd_ellipse(.x, centre = NULL, weights = NULL, ...)

## S3 method for class 'sfc'
st_sd_ellipse(.x, centre = NULL, weights = NULL, ...)

```

**Arguments**

.x                    sf points 2D or 3D

centre	Numeric. Coordinates 2D of central point. Default NULL, performs a calculation of mean_centre() from point localities
weights	Numeric. Same length of number of points.
...	ignored

**Value**

simple features as "POLYGON" with attributes: centre coordinates, values for mayor and minor axis radius (sigma.x and sigma.y), rotation (theta and theta\_corrected) and geometry properties (eccentricity, area and perimeter)

**Author(s)**

Gabriel Gaona

**Examples**

```
requireNamespace("ggplot2", quietly = TRUE)
library(sf, quietly = TRUE)
library(ggplot2)
bbx <- matrix(c(697047,9553483,
               696158,9560476,
               700964,9561425,
               701745,9555358),
             byrow = TRUE,
             ncol = 2)
bbx <- st_multipoint(bbx)
bbx <- st_cast(bbx,"POLYGON")
bbx <- st_sfc(bbx, crs = 31992)
set.seed(1234)
points <- st_sf(geometry = st_sample(bbx, 100))
SDE <- st_sd_ellipse(points)
ggplot() +
  geom_sf(data = SDE, fill = NA, color = "darkolivegreen") +
  geom_sf(data = points, color = "steelblue", size = 0.5)
```

# Index

sf, 5  
sf::st\_distance(), 5  
st\_central\_point, 2  
st\_central\_point(), 5  
st\_sd\_box, 4  
st\_sd\_distance, 5  
st\_sd\_ellipse, 6