

# Package ‘markstat’

December 15, 2025

**Type** Package

**Version** 0.1.5

**Title** Mark Correlation Functions for Spatial Point Patterns

**Maintainer** Mehdi Moradi <m2.moradi@yahoo.com>

**Imports** spatstat.univar, spatstat.geom, spatstat.random,  
spatstat.explore, spatstat.linnet, spatstat.utils, GET, stats,  
dplyr, ggplot2, tidyverse, patchwork

**Description** Provides a range of functions for computing both global and local mark correlation functions for spatial point patterns in either Euclidean spaces or on linear networks, with points carrying either real-valued or function-valued marks. For a review of mark correlation functions, see Eckardt and Moradi (2024) <[doi:10.1007/s13253-024-00605-1](https://doi.org/10.1007/s13253-024-00605-1)>.

**License** GPL (>= 2)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Mehdi Moradi [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-3905-4498>>),  
Matthias Eckardt [aut]

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2025-12-15 20:10:12 UTC

## Contents

|                  |   |
|------------------|---|
| markstat-package | 2 |
| as.data.frame.mc | 2 |
| fmcorr           | 3 |
| fmcorrihom       | 5 |
| lfmcorr          | 7 |
| lfmcorrihom      | 8 |

|             |    |
|-------------|----|
| lmc         | 10 |
| lmc         | 12 |
| lmc         | 14 |
| lmc         | 15 |
| lmc         | 17 |
| lmc         | 20 |
| lmc         | 22 |
| lmc         | 24 |
| Pfynwald    | 27 |
| plot.mc     | 28 |
| plot.testmc | 31 |
| print.mc    | 34 |
| testmc      | 35 |

**Index****41**markstat-package      *The markstat Package.***Description**

markstat: mark correlation functions for spatial point patterns

**Details**

The markstat package provides a range of functions for computing both global and local mark correlation functions for spatial point patterns in either Euclidean spaces or on linear networks, with points carrying either real-valued or function-valued marks.

**Author(s)**

**Maintainer:** Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> ([ORCID](#))

Authors:

- Matthias Eckardt

as.data.frame.mc      *Methods for mc objects.***Description**

Methods for mc objects.

**Usage**

```
## S3 method for class 'mc'
as.data.frame(x, ...)
```

**Arguments**

- x An object of class mc.
- ... Arguments passed to [as.data.frame](#).

**Value**

A data frame.

**Author(s)**

Mehdi Moradi <m2.moradi@yahoo.com>

**fmcorr**

*Mark correlation functions for homogeneous point patterns with function-valued marks.*

**Description**

Mark correlation functions for homogeneous point patterns with function-valued marks.

**Usage**

```
fmcorr(X,
       ftype = c("variogram", "stoyan", "rcorr", "shimatani", "beisbart",
                 "isham", "stoyancov", "schlather"),
       r = NULL,
       method = c("density", "loess"),
       normalise = TRUE,
       f = NULL,
       tol = 0.01,
       ...)
```

**Arguments**

- X An object of class ppp or lpp.
- ftype Type of the test function  $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather".
- r Optional. The values of the argument  $r$  at which the mark correlation function should be evaluated.
- method Type of smoothing, either density or loess. See details.
- normalise If `normalise=FALSE`, only the numerator of the expression for the mark correlation function will be computed.
- f Optional. Test function  $t_f$  used in the definition of the mark correlation function. If `ftype` is given,  $t_f$  should be `NULL`.
- tol Tolerance used in the calculation of the conditional mean of the marks. This is used only if `ftype` is `schlather`.
- ... Arguments passed to [unnormdensity](#) or [loess](#).

## Details

The object  $X$  should be an object of class `ppp` or `lpp`, with a function-valued mark. The mark should be a `data.frame` whose columns give the mark values per time point in an increasing order. The marks per each time point should be numeric.

See detailed information in the help pages of the functions `mcorr.ppp` and `mcorr.lpp`. As here,  $X$  has a function-valued mark, depending on the class of  $X$  being `ppp` or `lpp`, the function `mcorr.ppp` or `mcorr.lpp` will be applied to each time point of the function-valued mark, and at the end an overall mark correlation function will be given. See details in Eckardt et. al (2024).

## Value

a `data.frame` which gives the estimated mark correlation function and the distance vector  $r$  at which the mark correlation function is estimated. The outputs of the mark correlation functions for each time point are stored as an attribute, which can be extracted as `attr(., "ests")`.

## Author(s)

Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> and Matthias Eckardt

## References

Eckardt, M., Mateu, J., & Moradi, M. (2024). Function-Valued Marked Spatial Point Processes on Linear Networks: Application to Urban Cycling Profiles. *Stat*, 13(4), e70013.

## See Also

`mcorr.ppp`, `mcorr.lpp`, `fmc`  
`corrinhom`.

## Examples

```
library(spatstat.random)
library(spatstat.geom)
X <- rpoispp(100)
marks(X) <- data.frame(
  t1 = runif(npoints(X), 1, 10),
  t2 = runif(npoints(X), 1, 10),
  t3 = runif(npoints(X), 1, 10),
  t4 = runif(npoints(X), 1, 10),
  t5 = runif(npoints(X), 1, 10))
fmc(X, ftype = "stoyan", method = "density")
```

---

fmcorrinhom*Mark correlation functions for inhomogeneous point patterns with function-valued marks.*

---

## Description

Mark correlation functions for inhomogeneous point patterns with function-valued marks.

## Usage

```
fmcorrinhom(X,
  ftype = c("variogram", "stoyan", "rcorr", "shimatani", "beisbart",
  "isham", "stoyancov", "schlather"),
  r = NULL,
  lambda = NULL,
  method_lambda = c("kernel", "Voronoi"),
  bw = NULL,
  f = NULL,
  method = c("density", "loess"),
  correction = c("Ripley", "translate", "none"),
  normalise = TRUE,
  tol = 0.01,
  ...)
```

## Arguments

|               |  |
|---------------|--|
| X             | An object of class ppp or lpp.   |
| ftype         | Type of the test function $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather". |
| r             | Optional. The values of the argument r at which the mark correlation function should be evaluated.   |
| lambda        | Estimated intensity at data points. If not given, it will be estimated internally. See details.  |
| method_lambda | The method to be used for estimating intensity at data points, if lambda = NULL.   |
| bw            | Bandwidth method to be used for estimating intensity at data points if lambda = NULL and method_lambda = "kernel".                                       |
| f             | Optional. Test function $t_f$ used in the definition of the mark correlation function. If ftype is given, $t_f$ should be NULL.                          |
| method        | Type of smoothing, either density or loess.  |
| correction    | Type of edge correction to be applied, either of "Ripley", "translate", "none". This is used only if X is of class ppp.                                  |
| normalise     | If normalise=FALSE, only the numerator of the expression for the mark correlation function will be computed.   |
| tol           | Tolerance used in the calculation of the conditional mean of marks. This is used only if ftype is schlather.   |
| ...           | Arguments passed to <a href="#">unnormdensity</a> or <a href="#">loess</a> .   |

## Details

The object  $X$  should be an object of class `ppp` or `lpp`, with a function-valued mark. The mark should be a `data.frame` whose columns give the mark values per time point in an increasing order. The marks per each time point should be numeric.

See detailed information in the help pages of the functions `mcorrhinhom.ppp` and `mcorrhinhom.lpp`. As here,  $X$  has a function-valued mark, depending on the class of  $X$  being `ppp` or `lpp`, the function `mcorrhinhom.ppp` or `mcorrhinhom.lpp` will be applied to each time point of the function-valued mark, and at the end an overall mark correlation function will be given. See details in Eckardt et. al (2024).

## Value

a `data.frame` which gives the estimated mark correlation function and the distance vector  $r$  at which the mark correlation function is estimated. The outputs of the mark correlation functions for each time point are stored as an attribute, which can be extracted as `attr(., "ests")`.

## Author(s)

Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> and Matthias Eckardt

## References

- Moradi, M., & Eckardt, M. (2025). Inhomogeneous mark correlation functions for general marked point processes. arXiv e-prints, arXiv-2505.
- Eckardt, M., Mateu, J., & Moradi, M. (2024). Function-Valued Marked Spatial Point Processes on Linear Networks: Application to Urban Cycling Profiles. Stat, 13(4), e70013.

## See Also

`fmcorr`, `mcorrhinhom.ppp`, `mcorrhinhom.lpp`.

## Examples

```
library(spatstat.random)
library(spatstat.geom)
library(spatstat.explore)
X <- rpoispp(100)
marks(X) <- data.frame(
  t1 = runif(npoin
```

---

**lfc***Local mark correlation functions for homogeneous point patterns with function-valued marks.*

---

## Description

Local mark correlation functions for homogeneous point patterns with function-valued marks.

## Usage

```
lfc(X,
  ftype = c("variogram", "stoyan", "rcorr", "shimatani", "beisbart",
  "isham", "stoyancov", "schlather"),
  r = NULL,
  method = c("density", "loess"),
  normalise = TRUE,
  f = NULL,
  tol = 0.01,
  ...)
```

## Arguments

|           |  |
|-----------|--|
| X         | An object of class <code>ppp</code> or <code>lpp</code> .  |
| ftype     | Type of the test function $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather".   |
| r         | Optional. The values of the argument $r$ at which the mark correlation function should be evaluated.   |
| method    | Type of smoothing, either density or loess. See details.   |
| normalise | If <code>normalise=FALSE</code> , only the numerator of the expression for the mark correlation function will be computed.                                 |
| f         | Optional. Test function $t_f$ used in the definition of the mark correlation function. If <code>ftype</code> is given, $t_f$ should be <code>NULL</code> . |
| tol       | Tolerance used in the calculation of the conditional mean of the marks. This is used only if <code>ftype</code> is <code>schlather</code> .                |
| ...       | Arguments passed to <code>unnormdensity</code> or <code>loess</code> .   |

## Details

This function computes local mark correlation functions for a homogeneous point pattern with a function-valued mark. See the details of test functions used in `fmc`. Technical details are given in Eckardt and Moradi (2025).

## Value

a data.frame which gives the estimated overall local mark correlation function and the distance vector  $r$  at which the local mark correlation function is estimated. The outputs of the local mark correlation functions for each time point are stored as an attribute, which can be extracted as `attr(., "ests.time")`. The outputs of the local mark correlation functions for each data point are stored as an attribute, which can be extracted as `attr(., "ests.points")`.

## Author(s)

Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> and Matthias Eckardt

## References

Eckardt, M., Mateu, J., & Moradi, M. (2024). Function-Valued Marked Spatial Point Processes on Linear Networks: Application to Urban Cycling Profiles. *Stat*, 13(4), e70013.

Eckardt, M., & Moradi, M. (2025). Local indicators of mark association for marked spatial point processes.

## See Also

`mcorr.ppp`, `mcorr.lpp`, `fmcorr`.

## Examples

```
library(spatstat.random)
library(spatstat.geom)
X <- rpoispp(100)
marks(X) <- data.frame(
  t1 = runif(npoints(X), 1, 10),
  t2 = runif(npoints(X), 1, 10),
  t3 = runif(npoints(X), 1, 10),
  t4 = runif(npoints(X), 1, 10),
  t5 = runif(npoints(X), 1, 10))
lfmcorr(X, ftype = "stoyan", method = "density")
```

## Description

Local mark correlation functions for inhomogeneous point patterns with function-valued marks.

## Usage

```
lfmcorrinhom(X,
  ftype = c("variogram", "stoyan", "rcorr", "shimatani", "beisbart",
    "isham", "stoyancov", "schlather"),
  r = NULL,
  lambda = NULL,
  method_lambda = c("kernel", "Voronoi"),
  bw = NULL,
  f = NULL,
  method = c("density", "loess"),
  correction = c("Ripley", "translate", "none"),
  normalise = TRUE,
  tol = 0.01,
  ...)
```

## Arguments

|               |  |
|---------------|--|
| X             | An object of class ppp or lpp.   |
| ftype         | Type of the test function $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather". |
| r             | Optional. The values of the argument r at which the mark correlation function should be evaluated.   |
| lambda        | Estimated intensity at data points. If not given, it will be estimated internally. See details.  |
| method_lambda | The method to be used for estimating intensity at data points, if lambda = NULL.   |
| bw            | Bandwidth method to be used for estimating intensity at data points if lambda = NULL and method_lambda = "kernel".                                       |
| f             | Optional. Test function $t_f$ used in the definition of the mark correlation function. If ftype is given, $t_f$ should be NULL.                          |
| method        | Type of smoothing, either density or loess.  |
| correction    | Type of edge correction to be applied, either of "Ripley", "translate", "none". This is used only if X is of class ppp.                                  |
| normalise     | If normalise=FALSE, only the numerator of the expression for the mark correlation function will be computed.   |
| tol           | Tolerance used in the calculation of the conditional mean of marks. This is used only if ftype is schlather.   |
| ...           | Arguments passed to <a href="#">unnormdensity</a> or <a href="#">loess</a> .   |

## Details

This function computes local mark correlation functions for an inhomogeneous point pattern with a function-valued mark. See the details of test functions used in [fmcorrinhom](#). Technical details are given in Eckardt and Moradi (2025).

**Value**

a data.frame which gives the estimated overall local mark correlation function and the distance vector  $r$  at which the local mark correlation function is estimated. The outputs of the local mark correlation functions for each time point are stored as an attribute, which can be extracted as `attr(., "ests.time")`. The outputs of the local mark correlation functions for each data point are stored as an attribute, which can be extracted as `attr(., "ests.points")`.

**Author(s)**

Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> and Matthias Eckardt

**References**

- Eckardt, M., Mateu, J., & Moradi, M. (2024). Function-Valued Marked Spatial Point Processes on Linear Networks: Application to Urban Cycling Profiles. *Stat*, 13(4), e70013.
- Eckardt, M., & Moradi, M. (2025). Local indicators of mark association for marked spatial point processes.
- Moradi, M., & Eckardt, M. (2025). Inhomogeneous mark correlation functions for general marked point processes. arXiv e-prints, arXiv-2505.

**See Also**

[fmcorr](#), [mcorrinhom.ppp](#), [mcorrinhom.lpp](#).

**Examples**

```
library(spatstat.random)
library(spatstat.geom)
library(spatstat.explore)
X <- rpoispp(100)
marks(X) <- data.frame(
  t1 = runif(npoints(X), 1, 10),
  t2 = runif(npoints(X), 1, 10),
  t3 = runif(npoints(X), 1, 10),
  t4 = runif(npoints(X), 1, 10),
  t5 = runif(npoints(X), 1, 10))
lfmcorrinhom(X, ftype = "stoyan", method = "density", method_lambda = "kernel", bw = bw.scott)
```

**Description**

Local mark correlation functions for homogeneous point patterns on linear networks.

## Usage

```
## S3 method for class 'lpp'
lmc当地(X,
ftype = c("variogram", "stoyan", "rcorr", "shimatani", "beisbart",
"isham", "stoyancov", "schlather"),
r = NULL,
method = c("density", "loess"),
normalise = TRUE,
f = NULL,
tol = 0.01,
...)
```

## Arguments

|           |  |
|-----------|--|
| X         | An object of class lpp.  |
| ftype     | Type of the test function $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather". |
| r         | Optional. The values of the argument $r$ at which the mark correlation function should be evaluated.   |
| method    | Type of smoothing, either density or loess. See details.   |
| normalise | If normalise=FALSE, only the numerator of the expression for the mark correlation function will be computed.   |
| f         | Optional. Test function $t_f$ used in the definition of the mark correlation function. If ftype is given, $t_f$ should be NULL.                          |
| tol       | Tolerance used in the calculation of the conditional mean of the marks. This is used only if ftype is schlather.   |
| ...       | Arguments passed to <a href="#">unnormdensity</a> or <a href="#">loess</a> .   |

## Details

This function computes local mark correlation functions for a homogeneous point pattern on a linear network. See the details of test functions used in [mcorr.lpp](#). Technical details are given in Eckardt and Moradi (2025).

## Value

A data.frame where the first column contains the values of the argument  $r$  at which the mark correlation function is evaluated, and the remaining columns contain the estimated values of the mark correlation function for each data point; column names correspond to the IDs of the data points. If there are multiple numeric marks, the result will instead be a list, with each element corresponding to one of the marks.

## Author(s)

Mehdi Moradi <m2.moradi@yahoo.com> and Matthias Eckardt

## References

Eckardt, M., & Moradi, M. (2025). Local indicators of mark association for marked spatial point processes.

## See Also

[mcorr.lpp](#).

## Examples

```
library(spatstat.geom)
library(spatstat.linnet)
X <- rpoislpp(10, simiplenet)
marks(X) <- runif(npoints(X), 1 ,10)
lmcorr.lpp(X, ftype = "stoyan", method = "density")
```

**lmcorr.ppp**

*Local mark correlation functions for homogeneous point patterns on Euclidean spaces.*

## Description

Local mark correlation functions for homogeneous point patterns on Euclidean spaces.

## Usage

```
## S3 method for class 'ppp'
lmcorr(X,
       ftype = c("variogram", "stoyan", "rcorr", "shimatani", "beisbart",
                 "isham", "stoyancov", "schlather"),
       r = NULL,
       method = c("density", "loess"),
       normalise = TRUE,
       f = NULL,
       tol = 0.01,
       ...)
```

## Arguments

|        |  |
|--------|--|
| X      | An object of class ppp.  |
| ftype  | Type of the test function $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather". |
| r      | Optional. The values of the argument $r$ at which the mark correlation function should be evaluated.   |
| method | Type of smoothing, either density or loess. See details.   |

|           |   |
|-----------|---|
| normalise | If <code>normalise=FALSE</code> , only the numerator of the expression for the mark correlation function will be computed.                                  |
| f         | Optional. Test function $t_f$ used in the definition of the mark correlation function. If <code>f.type</code> is given, $t_f$ should be <code>NULL</code> . |
| tol       | Tolerance used in the calculation of the conditional mean of the marks. This is used only if <code>f.type</code> is <code>schlather</code> .                |
| ...       | Arguments passed to <code>unnormdensity</code> or <code>loess</code> .  |

## Details

This function computes local mark correlation functions for a stationary point pattern in  $\mathbb{R}^2$ . See the details of test functions used in `mcorr.hpp`. Technical details are given in Eckardt and Moradi (2025).

## Value

A `data.frame` where the first column contains the values of the argument  $r$  at which the mark correlation function is evaluated, and the remaining columns contain the estimated values of the mark correlation function for each data point; column names correspond to the IDs of the data points. If there are multiple numeric marks, the result will instead be a list, with each element corresponding to one of the marks.

## Author(s)

Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> and Matthias Eckardt

## References

Eckardt, M., & Moradi, M. (2025). Local indicators of mark association for marked spatial point processes.

## See Also

[mcorr.hpp](#).

## Examples

```
library(spatstat.geom)
library(spatstat.random)
X <- rpoispp(100)
marks(X) <- runif(npoints(X),10,11)
lmcorr.hpp(X, ftype = "stoyan", method = "density")
```

---

|                        |  |
|------------------------|--|
| <b>lmcorrinhom.lpp</b> | <i>Local mark correlation functions for inhomogeneous point patterns on linear networks.</i> |
|------------------------|--|

---

## Description

Local mark correlation functions for inhomogeneous point patterns on linear networks.

## Usage

```
## S3 method for class 'lpp'
lmcorrinhom(X,
ftype = c("variogram", "stoyan", "rcorr", "shimatani",
"beisbart", "isham", "stoyancov", "schlather"),
r = NULL,
lambda = NULL,
method_lambda = c("kernel", "Voronoi"),
bw = bw.scott.iso,
f = NULL,
method = c("density", "loess"),
normalise = TRUE,
tol = 0.01,
...)
```

## Arguments

|               |  |
|---------------|--|
| X             | An object of class lpp.  |
| ftype         | Type of the test function $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather". |
| r             | Optional. The values of the argument r at which the mark correlation function should be evaluated.   |
| lambda        | Estimated intensity at data points. If not given, it will be estimated internally. See details.  |
| method_lambda | The method to be used for estimating intensity at data points, if lambda = NULL.   |
| bw            | Bandwidth method to be used for estimating intensity at data points if lambda = NULL and method_lambda = "kernel".                                       |
| f             | Optional. Test function $t_f$ used in the definition of the mark correlation function. If ftype is given, $t_f$ should be NULL.                          |
| method        | Type of smoothing, either density or loess.  |
| normalise     | If normalise=FALSE, only the numerator of the expression for the mark correlation function will be computed.   |
| tol           | Tolerance used in the calculation of the conditional mean of marks. This is used only if ftype is schlather.   |
| ...           | Arguments passed to <a href="#">unnormdensity</a> or <a href="#">loess</a> .   |

## Details

This function computes local mark correlation functions for an inhomogeneous point pattern on a linear network. See the details of test functions used in [mcorrinhom.lpp](#). Technical details are given in Eckardt and Moradi (2025) and Moradi and Eckardt (2025).

## Value

A `data.frame` where the first column contains the values of the argument  $r$  at which the mark correlation function is evaluated, and the remaining columns contain the estimated values of the mark correlation function for each data point; column names correspond to the IDs of the data points. If there are multiple numeric marks, the result will instead be a list, with each element corresponding to one of the marks.

## Author(s)

Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> and Matthias Eckardt

## References

- Moradi, M., & Eckardt, M. (2025). Inhomogeneous mark correlation functions for general marked point processes. arXiv e-prints, arXiv-2505.
- Eckardt, M., & Moradi, M. (2025). Local indicators of mark association for marked spatial point processes.

## See Also

[mcorrinhom.lpp](#).

## Examples

```
library(spatstat.linnet)
library(spatstat.geom)
library(spatstat.explore)
X <- rpoislpp(10, simplenet)
marks(X) <- runif(npoints(X), 1, 10)
lmcorrinhom.lpp(X, ftype = "stoyan", method = "density",
method_lambda = "kernel", bw = bw.scott.iso)
```

## Description

Local mark correlation functions for inhomogeneous point patterns on Euclidean spaces.

## Usage

```
## S3 method for class 'ppp'
lmcorrinhom(X,
ftype = c("variogram", "stoyan", "rcorr", "shimatani",
"beisbart", "isham", "stoyancov", "schlather"),
r = NULL,
lambda = NULL,
method_lambda = c("kernel", "Voronoi"),
bw = bw.scott,
f = NULL,
method = c("density", "loess"),
correction = c("Ripley", "translate", "none"),
normalise = TRUE,
tol = 0.01,
...)
```

## Arguments

|               |  |
|---------------|--|
| X             | An object of class ppp.  |
| ftype         | Type of the test function $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather". |
| r             | Optional. The values of the argument r at which the mark correlation function should be evaluated.   |
| lambda        | Estimated intensity at data points. If not given, it will be estimated internally. See details.  |
| method_lambda | The method to be used for estimating intensity at data points, if lambda = NULL.   |
| bw            | Bandwidth method to be used for estimating intensity at data points if lambda = NULL and method_lambda = "kernel".                                       |
| f             | Optional. Test function $t_f$ used in the definition of the mark correlation function. If ftype is given, $t_f$ should be NULL.                          |
| method        | Type of smoothing, either density or loess.  |
| correction    | Type of edge correction to be applied, either of "Ripley", "translate", "none".  |
| normalise     | If normalise=FALSE, only the numerator of the expression for the mark correlation function will be computed.   |
| tol           | Tolerance used in the calculation of the conditional mean of marks. This is used only if ftype is schlather.   |
| ...           | Arguments passed to <a href="#">unnormdensity</a> or <a href="#">loess</a> .   |

## Details

This function computes local mark correlation functions for an inhomogeneous point pattern in  $\mathbb{R}^2$ . See the details of test functions used in [mcorrinhom.ppp](#). Technical details are given in Eckardt and Moradi (2025) and Moradi and Eckardt (2025).

**Value**

A `data.frame` where the first column contains the values of the argument  $r$  at which the mark correlation function is evaluated, and the remaining columns contain the estimated values of the mark correlation function for each data point; column names correspond to the IDs of the data points. If there are multiple numeric marks, the result will instead be a list, with each element corresponding to one of the marks.

**Author(s)**

Mehdi Moradi <m2.moradi@yahoo.com> and Matthias Eckardt

**References**

- Moradi, M., & Eckardt, M. (2025). Inhomogeneous mark correlation functions for general marked point processes. arXiv e-prints, arXiv-2505.
- Eckardt, M., & Moradi, M. (2025). Local indicators of mark association for marked spatial point processes.

**See Also**

[mcorrinhom.ppp](#).

**Examples**

```
library(spatstat.geom)
library(spatstat.random)
library(spatstat.explore)
X <- rpoispp(function(x,y) {100 * exp(-3*x)}, 100)
marks(X) <- runif(npoints(X), 1, 10)
mcorrinhom.ppp(X, ftype = "stoyan",
               method = "density", correction = "translate",
               method_lambda = "kernel", bw = bw.scott)
```

**mcorr.lpp**

*Mark correlation functions for homogeneous point patterns on linear networks.*

**Description**

Mark correlation functions for homogeneous point patterns on linear networks.

**Usage**

```
## S3 method for class 'lpp'
mcorr(X,
      ftype = c("variogram", "stoyan", "rcorr", "shimatani", "beisbart",
              "isham", "stoyancov", "schlather"),
      r = NULL,
```

```
method = c("density", "loess"),
normalise = TRUE,
f = NULL,
tol = 0.01,
...)
```

## Arguments

|           |  |
|-----------|--|
| X         | An object of class lpp.  |
| ftype     | Type of the test function $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather".   |
| r         | Optional. The values of the argument $r$ at which the mark correlation function should be evaluated.   |
| method    | Type of smoothing, either density or loess. See details.   |
| normalise | If <code>normalise=FALSE</code> , only the numerator of the expression for the mark correlation function will be computed.                                 |
| f         | Optional. Test function $t_f$ used in the definition of the mark correlation function. If <code>ftype</code> is given, $t_f$ should be <code>NULL</code> . |
| tol       | Tolerance used in the calculation of the conditional mean of the marks. This is used only if <code>ftype</code> is <code>schlather</code> .                |
| ...       | Arguments passed to <code>unnormdensity</code> or <code>loess</code> .   |

## Details

For a homogeneous point process  $X$  on a linear network, the  $t_f$ -correlation function  $\kappa_{t_f}(r)$  is given as

$$\kappa_{t_f}(r) = \frac{\mathbb{E}[t_f(m_x, m_y) | x, y \in X]}{c_{t_f}}, \quad d(x, y) = r,$$

where  $m_x, m_y$  are the marks of  $x, y \in X$ ,  $c_{t_f}$  is a normalizing factor, and  $d(x, y) = r$  is the shortest-path distance. Therefore, each mark correlation function is defined by a specific test function  $t_f(m_x, m_y)$  and its associated normalising factor  $c_{t_f}$ . Let  $\mu_m$  and  $\sigma_m^2$  be the mean and variance of marks, then, the list below gives different test functions  $t_f$  and their normalised factors  $c_{t_f}$ , following distinct available `ftype`.

**variogram:**  $t_f(m_x, m_y) = \frac{1}{2}(m_x - m_y)^2$ ,  $c_{t_f} = \sigma_m^2$ .

**stoyan:**  $t_f(m_x, m_y) = m_x m_y$ ,  $c_{t_f} = \mu_m^2$ .

**rcorr:**  $t_f(m_x, m_y) = m_x$ ,  $c_{t_f} = \mu_m$ .

**shimatani:**  $t_f(m_x, m_y) = (m_x - \mu_m)(m_y - \mu_m)$ ,  $c_{t_f} = \sigma_m^2$ .

**beisbart:**  $t_f(m_x, m_y) = m_x + m_y$ ,  $c_{t_f} = 2\mu_m$ .

**isham:**  $t_f(m_x, m_y) = m_x m_y - \mu_m^2$ ,  $c_{t_f} = \sigma_m^2$ .

**stoyancov:**  $t_f(m_x, m_y) = m_x m_y - \mu_m^2$ ,  $c_{t_f} = 1$ .

**schlather:**  $t_f(m_x, m_y) = (m_x - \mu_m(r))(m_y - \mu_m(r))$ ,  $c_{t_f} = \sigma_m^2$ .

For `ftype="schlather"`,  $\mu_m(r)$  denotes the mean of the marks of all pairs of points whose pairwise distance lies within a tolerance `tol` of  $r$ . We refer to Eckardt and Moradi (2024) for details of these mark correlation functions.

Regarding the smoothing functions, if `method="density"`, the functions `unnormdensity` will be called, and if `method="loess"`, the function `loess` will be called.

If your `f` type is not one of the defaults, then you need to give your test function  $t_f(m_1, m_2)$  using the argument `f`. In this case, `normalise` should be set as `FALSE`, as only the unnormalised version will be calculated. Depending on the form of the test function  $t_f(m_1, m_2)$ , one can manually compute the normalisation factor.

If the point pattern  $X$  has multiple real-valued marks, the function estimates the mark correlation function for each mark individually. In such case, marks are given as a `data.frame` whose columns represents different marks. The functions checks which columns are numeric, and for those the mark correlation function will be computed.

## Value

a `data.frame` which gives the estimated mark correlation function and the distance vector  $r$  at which the mark correlation function is estimated. If the point pattern  $X$  has multiple real-valued marks, the estimated mark correlation function will be given for each mark. Name of columns will be the name of marks.

## Author(s)

Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> and Matthias Eckardt

## References

Eckardt, M., & Moradi, M. (2024). Marked spatial point processes: current state and extensions to point processes on linear networks. *Journal of Agricultural, Biological and Environmental Statistics*, 29(2), 346-378.

## See Also

[mcorr.ppp](#).

## Examples

```
library(spatstat.linnet)
library(spatstat.geom)
X <- rpoislpp(10, simplenet)
marks(X) <- runif(npoints(X), 1, 10)
mcorr.lpp(X, ftype = "stoyan", method = "density")
```

---

**mcorr.hpp***Mark correlation functions for homogeneous point patterns on Euclidean spaces.*

---

## Description

Mark correlation functions for homogeneous point patterns on Euclidean spaces.

## Usage

```
## S3 method for class 'ppp'
mcorr(X,
ftype = c("variogram", "stoyan", "rcorr", "shimatani", "beisbart",
"isham", "stoyancov", "schlather"),
r = NULL,
method = c("density", "loess"),
normalise = TRUE,
f = NULL,
tol = 0.01,
...)
```

## Arguments

|           |  |
|-----------|--|
| X         | An object of class ppp.  |
| ftype     | Type of the test function $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather". |
| r         | Optional. The values of the argument $r$ at which the mark correlation function should be evaluated.   |
| method    | Type of smoothing, either density or loess. See details.   |
| normalise | If normalise=FALSE, only the numerator of the expression for the mark correlation function will be computed.   |
| f         | Optional. Test function $t_f$ used in the definition of the mark correlation function. If ftype is given, $t_f$ should be NULL.                          |
| tol       | Tolerance used in the calculation of the conditional mean of the marks. This is used only if ftype is schlather.   |
| ...       | Arguments passed to <a href="#">unnormdensity</a> or <a href="#">loess</a> .   |

## Details

For a homogeneous point process  $X$  in  $R^2$ , the  $t_f$ -correlation function  $\kappa_{t_f}(r)$  is given as

$$\kappa_{t_f}(r) = \frac{\mathbb{E}[t_f(m_x, m_y) | x, y \in X]}{c_{t_f}}, \quad d(x, y) = r,$$

where  $m_x, m_y$  are the marks of  $x, y \in X$ ,  $c_{t_f}$  is a normalizing factor, and  $d(x, y) = r$  is the Euclidean distance. Therefore, each mark correlation function is defined by a specific test function

$t_f(m_x, m_y)$  and its associated normalising factor  $c_{t_f}$ . Let  $\mu_m$  and  $\sigma_m^2$  be the mean and variance of marks, then, the list below gives different test functions  $t_f$  and their normalised factors  $c_{t_f}$ , following distinct available ftype.

**variogram:**  $t_f(m_x, m_y) = \frac{1}{2}(m_x - m_y)^2$ ,  $c_{t_f} = \sigma_m^2$ .

**stoyan:**  $t_f(m_x, m_y) = m_x m_y$ ,  $c_{t_f} = \mu_m^2$ .

**rcorr:**  $t_f(m_x, m_y) = m_x$ ,  $c_{t_f} = \mu_m$ .

**shimatani:**  $t_f(m_x, m_y) = (m_x - \mu_m)(m_y - \mu_m)$ ,  $c_{t_f} = \sigma_m^2$ .

**beisbart:**  $t_f(m_x, m_y) = m_x + m_y$ ,  $c_{t_f} = 2\mu_m$ .

**isham:**  $t_f(m_x, m_y) = m_x m_y - \mu_m^2$ ,  $c_{t_f} = \sigma_m^2$ .

**stoyancov:**  $t_f(m_x, m_y) = m_x m_y - \mu_m^2$ ,  $c_{t_f} = 1$ .

**schlather:**  $t_f(m_x, m_y) = (m_x - \mu_m(r))(m_y - \mu_m(r))$ ,  $c_{t_f} = \sigma_m^2$ .

For ftype="schlather",  $\mu_m(r)$  denotes the mean of the marks of all pairs of points whose pairwise distance lies within a tolerance tol of  $r$ . We refer to Eckardt and Moradi (2024) for details of these mark correlation functions.

Regarding the smoothing functions, if method="density", the functions `unnormdensity` will be called, and if method="loess", the function `loess` will be called.

If your ftype is not one of the defaults, then you need to give your test function  $t_f(m_1, m_2)$  using the argument f. In this case, normalise should be set as FALSE, as only the unnormalised version will be calculated. Depending on the form of the test function  $t_f(m_1, m_2)$ , one can manually compute the normalisation factor.

If the point pattern  $X$  has multiple real-valued marks, the function estimates the mark correlation function for each mark individually. In such case, marks are given as a data.frame whose columns represents different marks. The functions checks which columns are numeric, and for those the mark correlation function will be computed.

## Value

a data.frame which gives the estimated mark correlation function and the distance vector  $r$  at which the mark correlation function is estimated. If the point pattern  $X$  has multiple real-valued marks, the estimated mark correlation function will be given for each mark. Name of columns will be the name of marks.

## Author(s)

Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> and Matthias Eckardt

## References

Eckardt, M., & Moradi, M. (2024). Marked spatial point processes: current state and extensions to point processes on linear networks. *Journal of Agricultural, Biological and Environmental Statistics*, 29(2), 346–378.

Stoyan, D. and Stoyan, H. (1994) Fractals, random shapes and point fields: methods of geometrical statistics. John Wiley and Sons.

**See Also**

[markcorr](#), [mcorr.lpp](#).

**Examples**

```
library(spatstat.geom)
library(spatstat.random)
X <- rpoispp(100)
marks(X) <- runif(npoints(X),1,10)
mcorr.ppp(X, ftype = "stoyan", method = "density")
```

**mcorrinhom.lpp**

*Mark correlation functions for inhomogeneous point patterns on linear networks.*

**Description**

Mark correlation functions for inhomogeneous point patterns on linear networks.

**Usage**

```
## S3 method for class 'lpp'
mcorrinhom(X,
  ftype = c("variogram", "stoyan", "rcorr", "shimatani",
  "beisbart", "isham", "stoyancov", "schlather"),
  r = NULL,
  lambda = NULL,
  method_lambda = c("kernel", "Voronoi"),
  bw = bw.scott.iso,
  f = NULL,
  method = c("density", "loess"),
  normalise = TRUE,
  tol = 0.01,
  ...)
```

**Arguments**

- X An object of class lpp.
- ftype Type of the test function  $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather".
- r Optional. The values of the argument r at which the mark correlation function should be evaluated.
- lambda Estimated intensity at data points. If not given, it will be estimated internally. See details.
- method\_lambda The method to be used for estimating intensity at data points, if lambda = NULL.

|                        |   |
|------------------------|---|
| <code>bw</code>        | Bandwidth method to be used for estimating intensity at data points if <code>lambda = NULL</code> and <code>method_lambda = "kernel"</code> .               |
| <code>f</code>         | Optional. Test function $t_f$ used in the definition of the mark correlation function. If <code>f</code> type is given, $t_f$ should be <code>NULL</code> . |
| <code>method</code>    | Type of smoothing, either <code>density</code> or <code>loess</code> .  |
| <code>normalise</code> | If <code>normalise=FALSE</code> , only the numerator of the expression for the mark correlation function will be computed.                                  |
| <code>tol</code>       | Tolerance used in the calculation of the conditional mean of marks. This is used only if <code>f</code> type is <code>schlather</code> .                    |
| <code>...</code>       | Arguments passed to <code>unnormdensity</code> or <code>loess</code> .  |

## Details

Most of the details are given in `mcorrinhom.ppp`. However, here distances are measured via the shortest-path distances, and intensities are estimated differently. If `method_lambda = "kernel"`, then the function calls `densityQuick.lpp` in combination with the bandwidth chosen via `bw.scott.iso`. If `method_lambda = "Voronoi"`, the functions calls `densityVoronoi.lpp` with arguments `f=0.2`, `nrep = 400` which are recommended by Moradi et al. (2019).

In the case of point processes on linear networks, adding correction will slow down the calculations. Given that, it is argued that the type of correction does not have significant effect on the final output, we have here ignored correction terms. We refer to Eckardt and Moradi (2024) and Moradi and Eckardt (2025) for further details.

If the point pattern  $X$  has multiple real-valued marks, the function estimates the mark correlation function for each mark individually. In such case, marks are given as a `data.frame` whose columns represents different marks. The functions checks which columns are numeric, and for those the mark correlation function will be computed.

## Value

a `data.frame` which gives the estimated mark correlation function and the distance vector  $r$  at which the mark correlation function is estimated. If the point pattern  $X$  has multiple real-valued marks, the estimated mark correlation function will be given for each mark. Name of columns will be the name of marks.

## Author(s)

Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> and Matthias Eckardt

## References

- Moradi, M., Cronie, O., Rubak, E., Lachieze-Rey, R., Mateu, J., & Baddeley, A. (2019). Resampling-smoothing of Voronoi intensity estimators. *Statistics and computing*, 29(5), 995-1010.
- Moradi, M., & Eckardt, M. (2025). Inhomogeneous mark correlation functions for general marked point processes. arXiv e-prints, arXiv-2505.

## See Also

`mcorr.lpp`, `mcorrinhom.ppp`.

## Examples

```
library(spatstat.linnet)
library(spatstat.geom)
library(spatstat.explore)
X <- rpoislpp(10, simlenet)
marks(X) <- runif(npoints(X), 1, 10)
mcorrinhom.lpp(X, ftype = "stoyan", method = "density",
               method_lambda = "kernel", bw = bw.scott.iso)
```

mcorrinhom.hpp

*Mark correlation functions for inhomogeneous point patterns on Euclidean spaces.*

## Description

Mark correlation functions for inhomogeneous point patterns on Euclidean spaces.

## Usage

```
## S3 method for class 'ppp'
mcorrinhom(X,
ftype = c("variogram", "stoyan", "rcorr", "shimatani",
          "beisbart", "isham", "stoyancov", "schlather"),
r = NULL,
lambda = NULL,
method_lambda = c("kernel", "Voronoi"),
bw = bw.scott,
f = NULL,
method = c("density", "loess"),
correction = c("Ripley", "translate", "none"),
normalise = TRUE,
tol = 0.01,
...)
```

## Arguments

|               |  |
|---------------|--|
| X             | An object of class ppp.  |
| ftype         | Type of the test function $t_f$ . Currently any selection of "variogram", "stoyan", "rcorr", "shimatani", "beisbart", "isham", "stoyancov", "schlather". |
| r             | Optional. The values of the argument r at which the mark correlation function should be evaluated.   |
| lambda        | Estimated intensity at data points. If not given, it will be estimated internally. See details.  |
| method_lambda | The method to be used for estimating intensity at data points, if lambda = NULL.   |
| bw            | Bandwidth method to be used for estimating intensity at data points if lambda = NULL and method_lambda = "kernel".                                       |

|                         |   |
|-------------------------|---|
| <code>f</code>          | Optional. Test function $t_f$ used in the definition of the mark correlation function. If <code>f</code> type is given, $t_f$ should be <code>NULL</code> . |
| <code>method</code>     | Type of smoothing, either <code>density</code> or <code>loess</code> .  |
| <code>correction</code> | Type of edge correction to be applied, either of "Ripley", "translate", "none".   |
| <code>normalise</code>  | If <code>normalise=FALSE</code> , only the numerator of the expression for the mark correlation function will be computed.                                  |
| <code>tol</code>        | Tolerance used in the calculation of the conditional mean of marks. This is used only if <code>f</code> type is <code>schlather</code> .                    |
| ...                     | Arguments passed to <code>unnormdensity</code> or <code>loess</code> .  |

## Details

For an inhomogeneous point process  $X$  in  $R^2$ , the  $t_f$ -correlation function  $\kappa_{t_f}^{inhom}(r)$  is given as

$$\kappa_{t_f}^{inhom}(r) = \frac{\mathbb{E}_w [t_f(m_x, m_y) \mid x, y \in X]}{c_{t_f}}, \quad d(x, y) = r,$$

where  $m_x, m_y$  are the marks of  $x, y \in X$ ,  $\mathbb{E}_w$  is a conditional expectation with respect to a weighted Palm distribution,  $c_{t_f}$  is a normalising factor, and  $d(x, y) = r$  is the Euclidean distance. Therefore, each mark correlation function is defined by a specific test function  $t_f(m_x, m_y)$  and its associated normalising factor  $c_{t_f}$ . Let  $\mu_m$  and  $\sigma_m^2$  be the mean and variance of marks, then, the list below gives different test functions  $t_f$  and their normalised factors  $c_{t_f}$ , following distinct available `f` type.

**variogram:**  $t_f(m_x, m_y) = \frac{1}{2}(m_x - m_y)^2$ ,  $c_{t_f} = \sigma_m^2$ .

**stoyan:**  $t_f(m_x, m_y) = m_x m_y$ ,  $c_{t_f} = \mu_m^2$ .

**rcorr:**  $t_f(m_x, m_y) = m_x$ ,  $c_{t_f} = \mu_m$ .

**shimatani:**  $t_f(m_x, m_y) = (m_x - \mu_m)(m_y - \mu_m)$ ,  $c_{t_f} = \sigma_m^2$ .

**beisbart:**  $t_f(m_x, m_y) = m_x + m_y$ ,  $c_{t_f} = 2\mu_m$ .

**isham:**  $t_f(m_x, m_y) = m_x m_y - \mu_m^2$ ,  $c_{t_f} = \sigma_m^2$ .

**stoyancov:**  $t_f(m_x, m_y) = m_x m_y - \mu_m^2$ ,  $c_{t_f} = 1$ .

**schlather:**  $t_f(m_x, m_y) = (m_x - \mu_m(r))(m_y - \mu_m(r))$ ,  $c_{t_f} = \sigma_m^2$ .

For `f` type="schlather",  $\mu_m(r)$  denotes the mean of the marks of all pairs of points whose pairwise distance lies within a tolerance `tol` of  $r$ . We refer to Eckardt and Moradi (2024) for details of these mark correlation functions.

Regarding the smoothing functions, if `method="density"`, the functions `unnormdensity` will be called, and if `method="loess"`, the function `loess` will be called.

If your `f` type is not one of the defaults, then you need to give your test function  $t_f(m_1, m_2)$  using the argument `f`. In this case, `normalise` should be set as `FALSE`, as only the unnormalised version will be calculated. Depending on the form of the test function  $t_f(m_1, m_2)$ , one can manually compute the normalisation factor.

If `lambda = NULL`, the function internally estimates the intensity function using the given method via `method_lambda`. If `method_lambda = "kernel"`, the function calls `density.hpp`, with the bandwidth chosen by the given method `bw`, and argument `diggle=TRUE`. If `method_lambda = "Voronoi"`,

then the functions calls `densityVoronoi.ppp` with arguments `f=0.2`, `nrep = 400` which are recommended by Moradi et al. (2019).

Regarding the smoothing functions, if `method="density"`, the functions `unnormdensity` will be called, and if `method="loess"`, the function `loess` will be called.

Type of edge correction is chosen among "Ripley", "translate", "none". See details in `edge.Trans` and `edge.Ripley`.

If the point pattern  $X$  has multiple real-valued marks, the function estimates the mark correlation function for each mark individually. In such case, marks are given as a `data.frame` whose columns represents different marks. The functions checks which columns are numeric, and for those the mark correlation function will be computed.

### Value

a `data.frame` which gives the estimated mark correlation function and the distance vector  $r$  at which the mark correlation function is estimated. If the point pattern  $X$  has multiple real-valued marks, the estimated mark correlation function will be given for each mark. Name of columns will be the name of marks.

### Author(s)

Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> and Matthias Eckardt

### References

- Moradi, M., Cronie, O., Rubak, E., Lachieze-Rey, R., Mateu, J., & Baddeley, A. (2019). Resampling smoothing of Voronoi intensity estimators. *Statistics and computing*, 29(5), 995-1010.
- Moradi, M., & Eckardt, M. (2025). Inhomogeneous mark correlation functions for general marked point processes. arXiv e-prints, arXiv-2505.

### See Also

`mcorr.ppp`, `mcorrinhom.lpp`.

### Examples

```
library(spatstat.geom)
library(spatstat.random)
library(spatstat.explore)
X <- rpoispp(function(x,y) {100 * exp(-3*x)}, 100)
marks(X) <- runif(npoints(X), 1, 10)
mcorrinhom.ppp(X, ftype = "stoyan",
               method = "density", correction = "translate",
               method_lambda = "kernel", bw = bw.scott)
```

---

Pfynwald

*Scots pine trees in Pfynwald, Switzerland.*

---

## Description

Location and height of trees from the Pfynwald project, available via an Open Database License (<https://opendata.swiss>), which originates from a long-term irrigation experiment begun in 2003 within Switzerland's Pfyn-Finges national park.

## Usage

```
data(Pfynwald)
```

## Format

An object of class ppp, giving Cartesian xy-coordinate of trees in meters, and their height as mark.

## Author(s)

Mehdi Moradi <[m2.moradi@yahoo.com](mailto:m2.moradi@yahoo.com)> and Matthias Eckardt

## Source

<https://envidat.ch/#/metadata/pfynwald>

## References

Schaub, M., Haeni, M., Hug, C., Gessler, A., Rigling, A. (2016). Tree measurements 2002-2016 from the long-term irrigation experiment Pfynwald, Switzerland. EnviDat. <https://www.doi.org/10.16904/11>.

Moradi, M., & Eckardt, M. (2025). Inhomogeneous mark correlation functions for general marked point processes. arXiv e-prints, arXiv-2505.

## Examples

```
library(spatstat.geom)
data(Pfynwald)
plot(Pfynwald)
```

## Description

Plots objects of class ‘mc’ (mark correlation functions). Supports global and local types, multiple marks, and fully customizable appearance via ‘layers’ or ‘...’.

## Usage

```
## S3 method for class 'mc'
plot(
  x,
  which_marks = NULL,
  which_points = NULL,
  layers = list(),
  xlab = "Distance r",
  ylab = "Mark correlation",
  ...
)
```

## Arguments

|              |   |
|--------------|---|
| x            | An object of class ‘mc’.  |
| which_marks  | For global or local multi-mark objects: character or numeric vector specifying which marks to plot. Default ‘NULL’ (all marks). |
| which_points | For local MC objects: numeric or character vector specifying which points to overlay. Default ‘NULL’ (all points).              |
| layers       | A list of additional ggplot2 layers (themes, labs, scales, etc.) to add to the plot.  |
| xlab         | Label for x-axis (default "Distance r").  |
| ylab         | Label for y-axis (default "Mark correlation").  |
| ...          | Additional arguments passed to ‘geom_line()’.   |

## Value

Invisibly returns the ggplot object.

## Additional graphical customization

The object returned by `plot.testmc()` is a ggplot object. This means that all standard graphical components from the **ggplot2** package (such as layers, scales, and themes) can be added to the plot using the `+` operator. This allows users to further customize labels, colors, axes, and the overall appearance of the plot.

**Author(s)**

Mehdi Moradi <m2.moradi@yahoo.com>

**Examples**

```

library(spatstat.geom)
library(spatstat.random)
library(spatstat.explore)
library(spatstat.linnet)
library(ggplot2)

# --- Example 1: Single mark ----

X1 <- rpoispp(100)
marks(X1) <- cbind(m1 = runif(npoints(X1), 1, 10))

mc1 <- mcorr.ppp(X1, ftype = "stoyan", method = "density")

plot(mc1) +
  labs(
    title = "Mark correlation for a single mark",
    x = expression(r),
    y = "Correlation"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 11),
    legend.text = element_text(size = 10)
  )

# --- Example 2: Two marks ----

X2 <- rpoispp(100)
marks(X2) <- cbind(
  m1 = runif(npoints(X2), 1, 10),
  m2 = runif(npoints(X2), 1, 10)
)

mc2 <- mcorr.ppp(X2, ftype = "stoyan", method = "density")

# Plot both marks
plot(mc2) +
  labs(
    title = "Mark correlation for two marks",
    x = expression(r),
    y = "Correlation"
  )

```

```

theme_bw() +
theme(
  plot.title = element_text(size = 14, face = "bold"),
  axis.title = element_text(size = 12),
  axis.text = element_text(size = 10),
  legend.title = element_text(size = 11),
  legend.text = element_text(size = 10),
  legend.position = "top"
)

# Plot only mark m1
plot(mc2, which_marks = "m1") +
  labs(
    title = "Mark correlation for mark m1 only",
    x = expression(r),
    y = "Correlation"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10),
    legend.position = "none"
  )

# --- Example 3: Local mark correlation -----
lc1 <- lmcorr.ppp(X2, ftype = "stoyan", method = "density")

# Two selected points, mark m1
plot(lc1, which_points = 1:2, which_marks = "m1") +
  labs(
    title = "Local mark correlation (points 1-2, mark m1)",
    x = expression(r),
    y = "Local correlation"
  ) +
  theme_classic() +
  theme(
    plot.title = element_text(size = 13, face = "bold"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 11),
    legend.text = element_text(size = 10),
    legend.position = "right"
  )

# Single selected point
plot(lc1, which_points = 1) +
  labs(
    title = "Local mark correlation (single selected point)",
    x = expression(r),
    y = "Local correlation"
  )

```

```
) +
  theme_light() +
  theme(
    plot.title = element_text(size = 13, face = "bold"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10),
    legend.position = "bottom"
  )
```

---

**plot.testmc***Plot function.*

---

**Description**

Plots objects of class ‘testmc’. Supports global and local types, multiple marks, and fully customizable appearance via ‘layers’ or ‘...’.

**Usage**

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

**Arguments**

- x An object of class ‘testmc’ or ‘global\_envelope’.
- ... Ignored.

**Value**

Invisibly returns the ggplot object.

**Additional graphical customization**

The object returned by `plot.testmc()` is a `ggplot` object. This means that all standard graphical components from the `ggplot2` package (such as layers, scales, and themes) can be added to the plot using the `+` operator. This allows users to further customize labels, colors, axes, and the overall appearance of the plot.

**Author(s)**

Mehdi Moradi <m2.moradi@yahoo.com>

## Examples

```

library(spatstat.geom)
library(spatstat.random)
library(spatstat.explore)
library(spatstat.linnet)
library(ggplot2)

#####
## 1. Planar point patterns (ppp)
#####

# Global mark correlation (real-valued marks)
X_ppp <- rpoispp(200)
marks(X_ppp) <- data.frame(m1 = runif(npoints(X_ppp), 1, 10))

tc_ppp1 <- testmc(X_ppp,
                    fun = mcorr.ppp,
                    fun_args = list(ftype = "stoyan", method = "density"))

plot(tc_ppp1) +
  labs(
    title = "Global mark correlation (ppp)",
    x = expression(r),
    y = "Observed / Envelope"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 11),
    legend.text = element_text(size = 10)
  )

# Local mark correlation (real-valued marks)
X_ppp2 <- rpoispp(200)
marks(X_ppp2) <- data.frame(m1 = runif(npoints(X_ppp2), 1, 10))

tc_ppp2 <- testmc(X_ppp2,
                    fun = lmcorr.ppp,
                    fun_args = list(ftype = "stoyan", method = "density"))

plot(tc_ppp2[[1]]) +
  labs(
    title = "Local mark correlation (ppp)",
    x = expression(r),
    y = "Local correlation"
  ) +
  theme_classic() +
  theme(
    plot.title = element_text(size = 13, face = "bold"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)
  )

```

```

axis.text = element_text(size = 10),
legend.title = element_text(size = 11),
legend.text = element_text(size = 10),
legend.position = "right"
)

plot(tc_ppp2) +
  labs(
    title = "Effective range",
    x = expression(r),
    y = "Id of significant points"
  ) +
  theme_classic() +
  theme(
    plot.title = element_text(size = 23, face = "bold"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 20),
    legend.title = element_text(size = 21),
    legend.text = element_text(size = 30),
    legend.position = "right"
  )

#####
## 2. Linear network point patterns (lpp)
#####

X_lpp <- rpoislpp(40, simplenet)
marks(X_lpp) <- data.frame(m1 = runif(npoints(X_lpp), 1, 10))

tc_lpp1 <- testmc(X_lpp,
                    fun = mcorr.lpp,
                    fun_args = list(ftype = "stoyan", method = "density"))

plot(tc_lpp1) +
  labs(
    title = "Global mark correlation (lpp)",
    x = expression(r),
    y = "Observed / Envelope"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 11),
    legend.text = element_text(size = 10),
    legend.position = "top"
  )

# Local functional mark correlation (function-valued marks)
marks(X_lpp) <- data.frame(
  t1 = runif(npoints(X_lpp), 1, 10),
  t2 = runif(npoints(X_lpp), 1, 10)
)

```

```
)
tc_lpp2 <- testmc(X_lpp,
                    fun = lfmcorr,
                    fun_args = list(ftype = "stoyan", method = "density"))

plot(tc_lpp2[[1]]) +
  labs(
    title = "Local functional mark correlation (lpp)",
    x = expression(r),
    y = "Local correlation"
  ) +
  theme_light() +
  theme(
    plot.title = element_text(size = 13, face = "bold"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10),
    legend.title = element_text(size = 11),
    legend.text = element_text(size = 10),
    legend.position = "bottom"
  )
}
```

---

**print.mc***Print function.***Description**

Print function.

**Usage**

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

**Arguments**

|     |                        |
|-----|------------------------|
| x   | An object of class mc. |
| ... | Ignored.               |

**Value**

No return value. Used for printing.

**Author(s)**

Mehdi Moradi <m2.moradi@yahoo.com>

---

**testmc***Global envelope tests for random labelling.*

---

## Description

Generic function to perform global envelope test for random labelling using mark correlation functions.

## Usage

```
testmc(X,  
       fun,  
       nsim = 100,  
       fun_args = list(),  
       get_args = list(),  
       rlabel_args = list()  
       )
```

## Arguments

|             |  |
|-------------|--|
| X           | An object of class ppp or lpp.                                 |
| fun         | Function that computes the desired mark correlation function.  |
| nsim        | Number of permutations.  |
| fun_args    | Arguments passed to the chosen mark correlation function.      |
| get_args    | Arguments passed to the <a href="#">global_envelope_test</a> . |
| rlabel_args | Arguments passed to the <a href="#">rlabel</a> .               |

## Details

Generic function to perform global envelope test for random labelling using mark correlation functions.

If the point pattern has several real-valued marks, then the random labelling test will be performed for each mark individually, and the output will be a list of global envelop tests one per each mark.

If the mark correlation function is any of the LIMA family, a random labelling test will be performed for each individual point.

## Value

Either an object of class [global\\_envelope](#) or a list of such objects if the point pattern has multiple marks or the employed mark correlation function is of type LIMA.

## Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

## References

- Eckardt, M., & Moradi, M. (2024). Marked spatial point processes: current state and extensions to point processes on linear networks. *Journal of Agricultural, Biological and Environmental Statistics*, 29(2), 346–378.
- Moradi, M., & Eckardt, M. (2025). Inhomogeneous mark correlation functions for general marked point processes.
- Eckardt, M., & Moradi, M. (2025). Local indicators of mark association for marked spatial point processes.

## See Also

[global\\_envelope\\_test](#), [mcorr.ppp](#), [mcorr.lpp](#), [mcorrinhom.ppp](#), [mcorrinhom.lpp](#), [fmcorr](#), [lmcorr.ppp](#), [lmcorr.lpp](#).

## Examples

```
library(spatstat.geom)
library(spatstat.random)
library(spatstat.explore)
library(spatstat.lienet)

#####
## 1. Planar point patterns (ppp)
#####

# -----
# GLOBAL MARK CORRELATION (real-valued marks)
# -----
X <- rpoispp(200)
marks(X) <- data.frame(m1 = runif(npoints(X), 1, 10))

ts1 <- testmc(X,
               fun = mcorr.ppp,
               fun_args = list(ftype = "stoyan", method = "density"))
plot(ts1)
# Global mark correlation (real-valued marks)

ts2 <- testmc(X,
               fun = mcorrinhom.ppp,
               fun_args = list(ftype = "stoyan",
                              method = "density",
                              method_lambda = "kernel"))
plot(ts2)
# Inhomogeneous global mark correlation (real-valued marks)

# -----
# GLOBAL FUNCTIONAL MARK CORRELATION (function-valued marks)
# -----
marks(X) <- data.frame(
  t1 = runif(npoints(X), 1, 10),
```

```
t2 = runif(npoints(X), 1, 10),
t3 = runif(npoints(X), 1, 10),
t4 = runif(npoints(X), 1, 10),
t5 = runif(npoints(X), 1, 10)
)

ts3 <- testmc(X,
                fun = fmcorr,
                fun_args = list(ftype = "stoyan", method = "density"))
plot(ts3)
# Global functional mark correlation (function-valued marks)

ts4 <- testmc(X,
                fun = fmcorrinhom,
                fun_args = list(ftype = "stoyan",
                               method = "density",
                               method_lambda = "kernel",
                               bw = bw.scott))
plot(ts4)
# Inhomogeneous functional mark correlation (function-valued marks)

# -----
# LOCAL MARK CORRELATION (real-valued marks)
# -----
X <- rpoispp(200)
marks(X) <- data.frame(m1 = runif(npoints(X), 1, 10))

ts5 <- testmc(X,
                fun = lmcorr.ppp,
                fun_args = list(ftype = "stoyan", method = "density"))
plot(ts5[[1]])
# Local mark correlation (real-valued marks)

ts6 <- testmc(X,
                fun = lmcorrinhom.ppp,
                fun_args = list(ftype = "stoyan",
                               method = "density",
                               method_lambda = "kernel"))
plot(ts6[[1]])
# Inhomogeneous local mark correlation (real-valued marks)

# -----
# LOCAL FUNCTIONAL MARK CORRELATION (function-valued marks)
# -----
marks(X) <- data.frame(
  t1 = runif(npoints(X), 1, 10),
  t2 = runif(npoints(X), 1, 10),
  t3 = runif(npoints(X), 1, 10),
  t4 = runif(npoints(X), 1, 10),
  t5 = runif(npoints(X), 1, 10)
)
```

```

ts7 <- testmc(X,
                fun = lfmcorr,
                fun_args = list(ftype = "stoyan", method = "density"))
plot(ts7[[1]])
# Local functional mark correlation (function-valued marks)

ts8 <- testmc(X,
                fun = lfmcorrihom,
                fun_args = list(ftype = "stoyan",
                               method = "density",
                               method_lambda = "kernel",
                               bw = bw.scott))
plot(ts8[[1]])
# Inhomogeneous local functional mark correlation (function-valued marks)

#####
## 2. Linear network point patterns (lpp)
#####

X1 <- rpoislpp(40, simplenet)

# -----
# GLOBAL MARK CORRELATION (real-valued marks)
# -----
marks(X1) <- data.frame(m1 = runif(npoints(X1), 1, 10))

ts9 <- testmc(X1,
                fun = mcorr.lpp,
                fun_args = list(ftype = "stoyan", method = "density"))
plot(ts9)
# Global mark correlation on linear networks (real-valued marks)

ts10 <- testmc(X1,
                 fun = mcorrihom.lpp,
                 fun_args = list(ftype = "stoyan",
                                method = "density",
                                method_lambda = "kernel"))
plot(ts10)
# Inhomogeneous global mark correlation on linear networks (real-valued marks)

# -----
# GLOBAL FUNCTIONAL MARK CORRELATION (function-valued marks)
# -----
marks(X1) <- data.frame(
  t1 = runif(npoints(X1), 1, 10),
  t2 = runif(npoints(X1), 1, 10),
  t3 = runif(npoints(X1), 1, 10),
  t4 = runif(npoints(X1), 1, 10),
  t5 = runif(npoints(X1), 1, 10)
)

```

```
ts11 <- testmc(Xl,
                 fun = fmcorr,
                 fun_args = list(ftype = "stoyan", method = "density"))
plot(ts11)
# Global functional mark correlation on linear networks (function-valued marks)

ts12 <- testmc(Xl,
                 fun = fmcorrihom,
                 fun_args = list(ftype = "stoyan",
                                method = "density",
                                method_lambda = "kernel",
                                bw = bw.scott.iso))
plot(ts12)
# Inhomogeneous global functional mark correlation on linear networks
# (function-valued marks)

# -----
# LOCAL MARK CORRELATION (real-valued marks)
# -----
marks(Xl) <- data.frame(m1 = runif(npoints(Xl), 1, 10))

ts13 <- testmc(Xl,
                 fun = lmcorr.lpp,
                 fun_args = list(ftype = "stoyan", method = "density"))
plot(ts13[[1]])
# Local mark correlation on linear networks (real-valued marks)

ts14 <- testmc(Xl,
                 fun = lmcorrihom.lpp,
                 fun_args = list(ftype = "stoyan",
                                method = "density",
                                method_lambda = "kernel"))
plot(ts14[[1]])
# Inhomogeneous local mark correlation on linear networks (real-valued marks)

# -----
# LOCAL FUNCTIONAL MARK CORRELATION (function-valued marks)
# -----
marks(Xl) <- data.frame(
  t1 = runif(npoints(Xl), 1, 10),
  t2 = runif(npoints(Xl), 1, 10),
  t3 = runif(npoints(Xl), 1, 10),
  t4 = runif(npoints(Xl), 1, 10),
  t5 = runif(npoints(Xl), 1, 10)
)
ts15 <- testmc(Xl,
                 fun = lfmcorr,
                 fun_args = list(ftype = "stoyan", method = "density"))
plot(ts15[[1]])
```

```
# Local functional mark correlation on linear networks (function-valued marks)

ts16 <- testmc(Xl,
                 fun = lfmcorrinhom,
                 fun_args = list(ftype = "stoyan",
                                method = "density",
                                method_lambda = "kernel",
                                bw = bw.scott.iso))
plot(ts16[[1]])
# Inhomogeneous local functional mark correlation on linear networks
# (function-valued marks)
```

# Index

\* datasets  
Pfynwald, 27

as.data.frame, 3  
as.data.frame.mc, 2

bw.scott.iso, 23

density.ppp, 25  
densityQuick.lpp, 23  
densityVoronoi.lpp, 23  
densityVoronoi.ppp, 26

edge.Ripley, 26  
edge.Trans, 26

fmccorr, 3, 6–8, 10, 36  
fmcorrinhom, 4, 5, 9

global\_envelope\_test, 35, 36

lfmcorr, 7  
lfmcorrinhom, 8  
lmcorr.lpp, 10, 36  
lmcorr.ppp, 12, 36  
lmcorrinhom.lpp, 14  
lmcorrinhom.ppp, 15  
loess, 3, 5, 7, 9, 11, 13, 14, 16, 18–21, 23, 25, 26  
lpp, 4, 6

markcorr, 22  
markstat (markstat-package), 2  
markstat-package, 2

mcorr.lpp, 4, 8, 11, 12, 17, 22, 23, 36  
mcorr.ppp, 4, 8, 13, 19, 20, 26, 36  
mcorrinhom.lpp, 6, 10, 15, 22, 26, 36  
mcorrinhom.ppp, 6, 10, 16, 17, 23, 24, 36

Pfynwald, 27  
plot.mc, 28