Summary of recent updates to `spatstat`

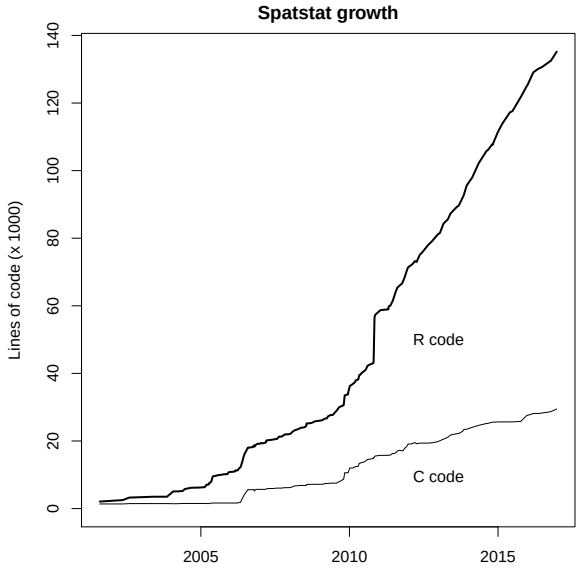
Adrian Baddeley, Rolf Turner and Ege Rubak

For `spatstat` version `1.48-0`

This is a summary of changes that have been made to the `spatstat` package since the publication of the accompanying book [2]. The book, published in December 2015, covers everything in `spatstat` up to version `1.42-0`, released in May 2015.

The current version of `spatstat` is `1.48-0`. It contains 389 new functions and 2 new datasets introduced after May 2015. This document summarises the most important changes.

This document also lists all *important* bugs detected *since 2010*.



Contents

1 Precis of all changes

Here is the text from the 'overview' sections of the News and Release Notes for each update.

- Sufficient Dimension Reduction for point processes.
- Alternating Gibbs Sampler for point process simulation.
- New class of spatially sampled functions.
- ROC and AUC extended to other types of point patterns and models.
- More support for linear networks.
- More support for infinite straight lines.
- `spatstat` now depends on the packages `nlme` and `rpart`.
- Important bug fix in `linearK`, `linearpcf`
- Changed internal format of `linnet` and `lpp` objects.
- Faster computation in linear networks.
- Bias correction techniques.
- Bounding circle of a spatial object.
- Option to plot marked points as arrows.
- Kernel smoothing accelerated.
- Workaround for bug in some graphics drivers affecting image orientation.
- Non-Gaussian smoothing kernels.
- Improvements to inhomogeneous multitype K and L functions.
- Variance approximation for pair correlation function.
- Leverage and influence for multitype point process models.
- Functions for extracting components of vector-valued objects.
- Recursive-partition point process models.
- Minkowski sum, morphological dilation and erosion with any shape.
- Minkowski sum also applicable to point patterns and line segment patterns.
- Important bug fix in Smooth.ppp
- Important bug fix in spatial CDF tests.
- More bug fixes for replicated patterns.
- Simulate a model fitted to replicated point patterns.
- Inhomogeneous multitype F and G functions.

- Summary functions recognise `correction="all"`
- Leverage and influence code handles bigger datasets.
- More support for pixel images.
- Improved progress reports.
- New dataset `redwood3`
- Fixed namespace problems arising when spatstat is not loaded.
- Important bug fix in leverage/influence diagnostics for Gibbs models.
- Surgery with linear networks.
- Tessellations on a linear network.
- Laslett's Transform.
- Colour maps for point patterns with continuous marks are easier to define.
- Pair correlation function estimates can be pooled.
- Stipulate a particular version of a package.
- More support for replicated point patterns.
- More support for tessellations.
- More support for multidimensional point patterns and point processes.
- More options for one-sided envelopes.
- More support for model comparison.
- Convexifying operation.
- Subdivide a linear network.
- Penttinen process can be simulated (by Metropolis-Hastings or CFTP).
- Calculate the predicted variance of number of points.
- Accelerated algorithms for linear networks.
- Quadrat counting accelerated, in some cases.
- Simulation algorithms have been accelerated; simulation outcomes are *not* identical to those obtained from previous versions of `spatstat`.
- Determinantal point process models.
- Random-effects and mixed-effects models for replicated patterns.
- Dao-Genton test, and corresponding simulation envelopes.
- Simulated annealing and simulated tempering.

- spatstat colour tools now handle transparent colours.
- Improvements to `[` and `subset` methods
- Extensions to kernel smoothing on a linear network.
- Support for one-dimensional smoothing kernels.
- Mark correlation function may include weights.
- Cross-correlation version of the mark correlation function.
- Penttinen pairwise interaction model.
- Improvements to simulation of Neyman-Scott processes.
- Improvements to fitting of Neyman-Scott models.
- Extended functionality for pixel images.
- Fitted intensity on linear network
- Triangulation of windows.
- Corrected an edge correction.

2 New datasets

The following datasets have been added to the package.

- `austates`: The states and large mainland territories of Australia represented as polygonal regions forming a tessellation.
- `redwood3`: a more accurate version of the `redwood` data.

3 New classes

- `ssf`: Class of spatially sampled functions.

4 New Functions

Following is a list of all the functions that have been added.

- `sdr`, `dimhat`: Sufficient Dimension Reduction for point processes.
- `simulate.rhohat`: Simulate a Poisson point process with the intensity estimated by `rhohat`.
- `rlpp`: Random points on a linear network with a specified probability density.
- `cut.lpp`: Method for `cut` for point patterns on a linear network.
- `has.close`: Faster way to check whether a point has a close neighbour.
- `psib`: Sibling probability (index of clustering strength in a cluster process).

- `rags`, `ragsAreaInter`, `ragsMultiHard`: Alternating Gibbs Sampler for point processes.
- `bugfixes`: List all bug fixes in recent versions of a package.
- `ssf`: Create a spatially sampled function
- `print.ssf`, `plot.ssf`, `contour.ssf`, `image.ssf`: Display a spatially sampled function
- `as.im.ssf`, `as.ppp.ssf`, `marks.ssf`, `marks<-.ssf`, `unmark.ssf`, `[.ssf`, `with.ssf`: Manipulate data in a spatially sampled function
- `Smooth.ssf`: Smooth a spatially sampled function
- `integral.ssf`: Approximate integral of spatially sampled function
- `roc.kppm`, `roc.lppm`, `roc.lpp`: Methods for `roc` for fitted models of class `"kppm"` and `"lppm"` and point patterns of class `"lpp"`
- `auc.kppm`, `auc.lppm`, `auc.lpp`: Methods for `auc` for fitted models of class `"kppm"` and `"lppm"` and point patterns of class `"lpp"`
- `timeTaken`: Extract the timing data from a `"timed"` object or objects.
- `rotate.infline`, `shift.infline`, `reflect.infline`, `flipxy.infline`: Geometrical transformations for infinite straight lines.
- `whichhalfplane`: Determine which side of an infinite line a point lies on.
- `matrixpower`, `matrixsqrt`, `matrixinvsqrt`: Raise a matrix to any power.
- `points.lpp`: Method for `points` for point patterns on a linear network.
- `pairs.linim`: Pairs plot for images on a linear network.
- `closetriples`: Find close triples of points.
- `anyNA.im`: Method for `anyNA` for pixel images.
- `bc`: Bias correction (Newton-Raphson) for fitted model parameters.
- `rex`: Richardson extrapolation for numerical integrals and statistical model parameter estimates.
- `boundingcircle`, `boundingcentre`: Find the smallest circle enclosing a window or point pattern.
- `[.linim` : Subset operator for pixel images on a linear network.
- `mean.linim`, `median.linim`, `quantile.linim`: The mean, median, or quantiles of pixel values in a pixel image on a linear network.
- `weighted.median`, `weighted.quantile`: Median or quantile of numerical data with associated weights.
- `"[.linim"`: Subset operator for pixel images on a linear network.
- `mean.linim`, `median.linim`, `quantile.linim`: The mean, median, or quantiles of pixel values in a pixel image on a linear network.
- `boundingcircle`, `boundingcentre`: Smallest circle enclosing a spatial object.

- `split.msr`: Decompose a measure into parts.
- `unstack.msr`: Decompose a vector-valued measure into its component measures.
- `unstack.ppp`, `unstack.psp`, `unstack.lpp`: Given a spatial pattern with several columns of marks, separate the columns and return a list of spatial patterns, each having only one column of marks.
- `kernel.squint`: Integral of squared kernel, for the kernels used in density estimation.
- `as.im.data.frame`: Build a pixel image from a data frame of coordinates and pixel values.
- `covering`: Cover a window using discs of a given radius.
- `dilationAny`, `erosionAny`, `%(-)%` : Morphological dilation and erosion by any shape.
- `FmultiInhom`, `GmultiInhom` Inhomogeneous multitype/marked versions of the summary functions `Fest`, `Gest`.
- `kernel.moment` Moment or incomplete moment of smoothing kernel.
- `MinkowskiSum`, `%(+)%`: Minkowski sum of two windows: `A %(+)% B`, or `MinkowskiSum(A,B)`
- `nobjects`: New generic function for counting the number of 'things' in a dataset. There are methods for `ppp`, `ppx`, `psp`, `tess`.
- `parameters.interact`, `parameters.fii`: Extract parameters from interpoint interactions. (These existing functions are now documented.)
- `ppmInfluence`: Calculate `leverage.ppm`, `influence.ppm` and `dfbetas.ppm` efficiently.
- `rppm`, `plot.rppm`, `predict.rppm`, `prune.rppm`: Recursive-partition point process models.
- `simulate.mppm` Simulate a point process model fitted to replicated point patterns.
- `update.interact`: Update the parameters of an interpoint interaction. [This existing function is now documented.]
- `where.max`, `where.min` Find the spatial location(s) where a pixel image achieves its maximum or minimum value.
- `compileK`, `compilepcf`: make a K function or pair correlation function given the pairwise distances and their weights. [These existing internal functions are now documented.]
- `laslett`: Laslett's Transform.
- `lintess`: Tessellation on a linear network.
- `divide.linnet`: Divide a linear network into pieces demarcated by a point pattern.
- `insertVertices`: Insert new vertices in a linear network.
- `thinNetwork`: Remove vertices and/or segments from a linear network etc.
- `connected.linnet`: Find connected components of a linear network.
- `nvertices`, `nvertices.linnet`, `nvertices.owin`: Count the number of vertices in a linear network or vertices of the boundary of a window.

- `as.data.frame.linim`, `as.data.frame.linfun`: Extract a data frame of spatial locations and function values from an object of class `linim` or `linfun`.
- `as.linfun`, `as.linfun.linim`, `as.linfun.lintess`: Convert other kinds of data to a `linfun` object.
- `requireversion`: Require a particular version of a package (for use in stand-alone R scripts).
- `as.function.tess`: Convert a tessellation to a `function(x,y)`. The function value indicates which tile of the tessellation contains the point (x, y).
- `tileindex`: Determine which tile of a tessellation contains a given point (x, y).
- `persp.leverage.ppm`: Method for persp plots for objects of class `leverage.ppm`
- `AIC.mppm`, `extractAIC.mppm`: AIC for point process models fitted to replicated point patterns.
- `nobs.mppm`, `terms.mppm`, `getCall.mppm`: Methods for point process models fitted to replicated point patterns.
- `rPenttinen`: Simulate the Penttinen process using perfect simulation.
- `varcount`: Given a point process model, compute the predicted variance of the number of points falling in a window.
- `inside.boxx`: Test whether multidimensional points lie inside a specified multidimensional box.
- `lixellate`: Divide each segment of a linear network into smaller segments.
- `nsegments.linnet`, `nsegments.lpp`: Count the number of line segments in a linear network.
- `grow.boxx`: Expand a multidimensional box.
- `deviance.ppm`, `deviance.lppm`: Deviance for a fitted point process model.
- `pseudoR2`: Pseudo-R-squared for a fitted point process model.
- `tiles.empty` Checks whether each tile of a tessellation is empty or nonempty.
- `summary.linim`: Summary for a pixel image on a linear network.
- Determinantal Point Process models:
  - `dppm`: Fit a determinantal point process model.
  - `fitted.dppm`, `predict.dppm`, `intensity.dppm`: prediction for a fitted determinantal point process model.
  - `Kmodel.dppm`, `pcfmodel.dppm`: Second moments of a determinantal point process model.
  - `rdpp`, `simulate.dppm`: Simulation of a determinantal point process model.
  - `logLik.dppm`, `AIC.dppm`, `extractAIC.dppm`, `nobs.dppm`: Likelihood and AIC for a fitted determinantal point process model.
  - `print.dppm`, `reach.dppm`, `valid.dppm`: Basic information about a `dpp` model.
  - `coef.dppm`, `formula.dppm`, `print.dppm`, `terms.dppm`, `labels.dppm`, `model.frame.dppm`, `model.matrix.dppm`, `model.images.dppm`, `is.stationary.dppm`, `reach.dppm`, `unitname.dppm`, `unitname<-.dppm`, `Window.dppm`: Various methods for `dppm` objects.

  - `parameters.dppm`: Extract meaningful list of model parameters.
  - `objsurf.dppm`: Objective function surface of a `dppm` object.
  - `residuals.dppm`: Residual measure for a `dppm` object.

- Determinantal Point Process model families:
  - `dppBessel`, `dppCauchy`, `dppGauss`, `dppMatern`, `dppPowerExp`: Determinantal Point Process family functions.
  - `detpointprocfamilyfun`: Create a family function.
  - `update.detpointprocfamily`: Set parameter values in a determinantal point process model family.
  - `simulate.dppm`: Simulation.
  - `is.stationary.detpointprocfamily`, `intensity.detpointprocfamily`, `Kmodel.detpointprocfamily`, `pcfmodel.detpointprocfamily`: Moments.
  - `dim.detpointprocfamily`, `dppapproxkernel`, `dppapproxpcf`, `dppeigen`, `dppkernel`, `dppparbounds`, `dppspecdenrange`, `dppspecden`: Helper functions.
- `dg.envelope`: Simulation envelopes corresponding to Dao-Genton test.
- `dg.progress`: Progress plot (envelope representation) for the Dao-Genton test.
- `dg.sigtrace`: significance trace for the Dao-Genton test.
- `markcrosscorr`: Mark cross-correlation function for point patterns with several columns of marks.
- `rtemper`: Simulated annealing or simulated tempering.
- `rgb2hsva`: Convert RGB to HSV data, like `rgb2hsv`, but preserving transparency.
- `superimpose.ppplist`, `superimpose.splitppp`: New methods for 'superimpose' for lists of point patterns.
- `dkernel`, `pkernel`, `qkernel`, `rkernel`: Probability density, cumulative probability, quantiles and random generation from distributions used in basic one-dimensional kernel smoothing.
- `kernel.factor`: Auxiliary calculations for one-dimensional kernel smoothing.
- `spatdim`: Spatial dimension of any object in the `spatstat` package.
- `as.boxx`: Convert data to a multi-dimensional box.
- `intensity.ppx`: Method for `intensity` for multi-dimensional space-time point patterns.
- `fourierbasis`: Evaluate Fourier basis functions in any number of dimensions.
- `valid`: New generic function, with methods `valid.ppm`, `valid.lppm`, `valid.dppm`.
- `emend`, `emend.ppm`, `emend.lppm`: New generic function with methods for `ppm` and `lppm`. `emend.ppm` is equivalent to `project.ppm`.
- `Penttinen`: New pairwise interaction model.
- `quantile.density`: Calculates quantiles from kernel density estimates.

- `CDF.density`: Calculates cumulative distribution function from kernel density estimates.
- `triangulate.owin`: decompose a spatial window into triangles.
- `fitted.lppm`: fitted intensity values for a point process on a linear network.
- `parameters`: Extract all parameters from a fitted model.

5 Alphabetical list of changes

Here is a list of all changes made to existing functions, listed alphabetically.

- `affine.owin`: Allows transformation matrix to be singular, if the window is polygonal.
- `anova.mppm`: Now handles Gibbs models, and performs the adjusted composite likelihood ratio test. New argument `fine`.
- `as.function.tess`: New argument `values` specifies the function values.
- `as.im.function`: New argument `strict`.
- `as.layered`: Default method now handles a (vanilla) list of spatial objects.
- `as.linnet.psp`: If the line segment pattern has marks, then the resulting linear network also carries these marks in the `$lines` component.
- `as.owin.data.frame`: New argument `step`
- `as.polygonal`: Can now repair errors in polygon data, if `repair=TRUE`.
- `bw.ppl`: New arguments `weights` and `sigma`.
- `cdf.test`: Calculations are more robust against numerical rounding effects.
- `cdf.test.mppm`:
  - Now handles Gibbs models.
  - Now recognises `covariate="x"` or `"y"`.
- `clarkevans`: The argument `correction="all"` is now recognised: it selects all the available options. [This is also the default.]
- `clickpoly`: The polygon is now drawn progressively as the user clicks new vertices.
- `closepairs.ppp`, `closepairs.pp3`:
  - New arguments `distinct` and `neat` allow more options.
  - Argument `ordered` has been replaced by `twice` (but `ordered` is still accepted, with a warning).
  - Performance improved (computation time and memory requirements reduced.) This should improve the performance of many functions in `spatstat`.
- `clusterset`: Improved behaviour.
- `clusterfit`: New argument `algorithm` specifies the choice of optimisation algorithm.

- `collapse.fv`: This is now treated as a method for the `nlme` generic `collapse`. Its syntax has been adjusted slightly.
- `connected.im`: Now handles a logical-valued image properly. Arguments `...` now determine pixel resolution.
- `connected.owin`: Arguments `...` now determine pixel resolution.
- `contour.im`: New argument `col` specifies the colour of the contour lines. If `col` is a colour map, then the contours are drawn in different colours.
- `crossing.psp`: New argument `details` gives more information about the intersections between the segments.
- `cut.ppp`: Argument `z` can be `"x"` or `"y"` indicating one of the spatial coordinates.
- `dclf.test, mad.test, dclf.progress, mad.progress, dclf.sigtrace, mad.sigtrace, dg.progress, dg.sigtrace`:
  - New argument `clamp` determines the test statistic for one-sided tests.
  - New argument `rmin` determines the left endpoint of the test interval.
  - New argument `leaveout` specifies how to calculate discrepancy between observed and simulated function values.
  - New argument `scale` allows summary function values to be rescaled before the comparison is performed.
  - New argument `interpolate` supports interpolation of p-value.
  - New argument `interpolate` supports interpolation of critical value of test.
- `default.rmhcontrol, default.rmhexpand`: New argument `w`.
- `density.lpp`:
  - New argument `kernel` specifies the smoothing kernel. Any of the standard one-dimensional smoothing kernels can be used.
  - Now supports both the 'equal-split continuous' and 'equal-split discontinuous' smoothers. New argument `continuous` determines the choice of smoother.
  - New argument `weights`.
- `density.ppp`:
  - A non-Gaussian kernel can now be specified using the argument `kernel`.
  - Argument `weights` can now be a pixel image.
  - Accelerated by about 30% when `at="pixels"`.
  - Accelerated by about 15% in the case where `at="points"` and `kernel="gaussian"`.
  - Accelerated in the cases where weights are given or `diggle=TRUE`.
- `density.psp`:
  - New argument `method`.
  - Accelerated by 1 to 2 orders of magnitude.

- `dfbetas.ppm`: For Gibbs models, memory usage has been dramatically reduced, so the code can handle larger datasets and finer quadrature schemes.
- `diagnose.ppm`: Infinite values of `rbord` are now ignored and treated as zero. This ensures that `diagnose.ppm` has a sensible default when the fitted model has infinite reach.
- `diagnose.ppm, plot.diagppm`: New arguments `col.neg, col.smooth` control the colour maps.
- `dilation.ppp`: Improved geometrical accuracy. Now accepts arguments to control resolution of polygonal approximation.
- `discs`:
  - Now accepts a single numeric value for `radii`.
  - New argument `npoly`.
  - Accelerated in some cases.
- `distfun`: When the user calls a distance function that was created by `distfun`, the user may now give a `ppp` or `lpp` object for the argument `x`, instead of giving two coordinate vectors `x` and `y`.
- `edge.Trans`: New argument `gW` for efficiency.
- `envelope`:
  - New argument `clamp` gives greater control over one-sided envelopes.
  - New argument `funargs`
  - New argument `scale` allows global envelopes to have width proportional to a specified function of r, rather than constant width.
  - New argument `funYargs` contains arguments to the summary function when applied to the data pattern only.
- `envelope.lpp`, `envelope.lppm`: New arguments `fix.n` and `fix.marks` allow envelopes to be computed using simulations conditional on the observed number of points.
- `Fest`: Additional checks for errors in input data.
- `funxy`: When the user calls a function that was created by `funxy`, the user may now give a `ppp` or `lpp` object for the argument `x`, instead of giving two coordinate vectors `x` and `y`.
- `Geyer`: The saturation parameter `sat` can now be less than 1.
- `grow.rectangle`: New argument `fraction`.
- `Hest`:
  - Argument `X` can now be a pixel image with logical values.
  - New argument `W`. [Based on code by Kassel Hingee.]
  - Additional checks for errors in input data.
- `influence.ppm`: For Gibbs models, memory usage has been dramatically reduced, so the code can handle larger datasets and finer quadrature schemes.

- `intensity.ppm`: Intensity approximation is now implemented for area-interaction model, and Geyer saturation model.
- `Kcross.inhom`, `Kdot.inhom`, `Kmulti.inhom`: These functions now allow intensity values to be given by a fitted point process model. New arguments `update`, `leaveoneout`, `lambdaX`.
- `Kest` Accelerated computation (for translation and rigid corrections) when window is an irregular shape.
- `Kest.fft`: Now has `...` arguments allowing control of spatial resolution.
- `kppm`:
  - Fitting a model with `clusters="LGCP"` no longer requires the package `RandomFields` to be loaded explicitly.
  - New argument `algorithm` specifies the choice of optimisation algorithm.
  - Left hand side of formula can now involve entries in the list `data`.
  - A warning about infinite values of the summary function no longer occurs when the default settings are used. Also affects `mincontrast`, `cauchy.estpcf`, `lgcp.estpcf`, `matclust.estpcf`, `thomas.estpcf`, `vargamma.estpcf`.
- `Lcross.inhom`, `Ldot.inhom`: These functions now allow intensity values to be given by a fitted point process model. New arguments `update`, `leaveoneout`, `lambdaX`.
- `lengths.psp`: New argument `squared`.
- `leverage.ppm`: For Gibbs models, memory usage has been dramatically reduced, so the code can handle larger datasets and finer quadrature schemes.
- `linearK`, `linearpcf` and relatives: substantially accelerated.
- `linearKinhom`: new argument `normpower`.
- `linearpcf`: new argument `normpower`.
- `linnet`:
  - The internal format of a `linnet` (linear network) object has been changed. Existing datasets of class `linnet` are still supported. However, computation will be faster if they are converted to the new format. To convert a linnet object `L` to the new format, use `L <- as.linnet(L)`.
  - If the argument `edges` is given, then this argument now determines the ordering of the sequence of line segments. For example, the `i`-th row of `edges` specifies the `i`-th line segment in `as.psp(L)`.
  - New argument `warn`.
- `lintess`: Argument `df` can be missing or `NULL`, resulting in a tesellation with only one tile.
- `logLik.ppm`: New argument `absolute`.
- `logLik.mppm`: new argument `warn`.
- `lpp`:

  - The internal format of an `lpp` object has been changed. Existing datasets of class `lpp` are still supported. However, computation will be faster if they are converted to the new format. To convert an `lpp` object `X` to the new format, use `X <- as.lpp(X)`.
  - `X` can be missing or `NULL`, resulting in an empty point pattern.

- `lpp`, `as.lpp`: These functions now handle the case where coordinates `seg` and `tp` are given but `x` and `y` are missing.

- `lppm`:
  - New argument `random` controls placement of dummy points.
  - Computation accelerated.

- `markcorr`: New argument `weights` allows computation of the weighted version of the mark correlation function.

- `mppm`:
  - Now handles models with a random effect component. (This is covered in [2, Chap. 16].)
  - New argument `random` is a formula specifying the random effect. (This is covered in [2, Chap. 16].)
  - Performs more checks for consistency of the input data.
  - New arguments `gcontrol` and `reltol.pql` control the fitting algorithm.

- `nndist.lpp, nnwhich.lpp, nncross.lpp, distfun.lpp`: New argument `k` allows computation of k-th nearest point. Computation accelerated.

  `nnfun.lpp`: New argument `k`.

- `padimage`: New argument `W` allows an image to be padded out to fill any window.

- `pcf.ppp`:
  - New argument `ratio` allows several estimates of pcf to be pooled.
  - Now calculates an analytic approximation to the variance of the estimate of the pair correlation function (when `var.approx=TRUE`).
  - Now returns the smoothing bandwidth used, as an attribute of the result.

- `pcfinhom`: Default behaviour is changed when `lambda` is a fitted model. The default is now to re-fit the model to the data before computing pcf. New arguments `update` and `leaveoneout` control this.

- `pixellate.ppp`: Accelerated in the case where weights are given.

- `plot.anylist`:
  - If a list entry `x[[i]]` belongs to class `"anylist"`, it will be expanded so that each entry `x[[i]][[j]]` will be plotted as a separate panel.
  - New arguments `panel.begin.args`, `panel.end.args`

- `plot.im`:
  - Now handles complex-valued images.

  - New argument `workaround` to avoid a bug in some MacOS device drivers that causes the image to be displayed in the wrong spatial orientation.

- `plot.influence.ppm`: New argument `multiplot`.
- `plot.kppm`:
  - New arguments `pause` and `xname`.
  - The argument `what="all"` is now recognised: it selects all the available options. [This is also the default.]
- `plot.leverage.ppm`: New argument `multiplot`.
- `plot.linfun`: Now passes arguments to the function being plotted.
- `plot.lintess`: Improved plot method, with more options.
- `plot.lpp`:
  - New argument `show.network`.
  - For a point pattern with continuous marks ("real numbers") the colour arguments `cols`, `fg`, `bg` can now be vectors of colour values, and will be used to determine the default colour map for the marks.
- `plot.mppm`: New argument `se`.
- `plot.msr`: Now handles multitype measures. New argument `multiplot`.
- `plot.pp3`: New arguments `box.front`, `box.back` control plotting of the box.
- `plot.ppp`:
  - The default colour for the points is now a transparent grey, if this is supported by the plot device.
  - For a point pattern with continuous marks ("real numbers") the colour arguments `cols`, `fg`, `bg` can now be vectors of colour values, and will be used to determine the default colour map for the marks.
  - Now recognises graphics parameters for text, such as `family` and `srt`
  - When `clipwin` is given, any parts of the boundary of the window of `x` that lie inside `clipwin` will also be plotted.
- `plot.profilepl` ,`plot.quadratcount`, `plot.quadrattest`, `plot.tess`: Now recognise graphics parameters for text, such as `family` and `srt`
- `plot.solist`: New arguments `panel.begin.args`, `panel.end.args`
- `pool.fv`: The default plot of the pooled function no longer includes the variance curves.
- `ppm.ppp, ppm.quad` New argument `emend`, equivalent to `project`.
- `predict.kppm, residuals.kppm` Now issues a warning when the calculation ignores the cluster/Cox component and treats the model as if it were Poisson. (This currently happens in predict.kppm when se=TRUE or interval != "none", and in residuals.kppm when type != "raw").

- predict.mppm: The argument type="all" is now recognised: it selects all the available options. [This is also the default.]
- print.quad: More information is printed.
- progressreport
  - Behaviour improved.
  - New arguments state, tick, showtime.
  - New option: style="tk"
- quadratcount.ppp: Computation accelerated in some cases.
- quadrat.test.ppm: Computation accelerated in some cases.
- qqplot.ppm Argument expr can now be a list of point patterns, or an envelope object containing a list of point patterns.
- rcellnumber: New argument mu.
- rgbim, hsvim: New argument A controls the alpha (transparency) channel.
- rgb2hex, col2hex, paletteindex, is.colour, samecolour, complementarycolour, is.grey, to.grey These colour tools now handle transparent colours.
- rgb2hex: New argument maxColorValue

  rhohat.lpp: New argument random controls placement of dummy points.
- rLGCP: This function no longer requires the package RandomFields to be loaded explicitly.
- rMaternI, rMaternII: These functions can now generate random patterns in three dimensions and higher dimensions, when the argument win is of class box3 or boxx.
- rmh.ppm, rmhmodel.ppm, simulate.ppm: A model fitted using the Penttinen interaction can now be simulated.
- rmh.default, rmhmodel.default:
  - These functions now recognise cif='penttinen' for the Penttinen interaction.
  - New arguments nsim, saveinfo.
- rose.default New argument weights.
- rose New arguments start and clockwise specify the convention for measuring and plotting angles.
- rotmean: New argument padzero. Default behaviour has changed.
- rpoispp Accelerated, when lambda is a pixel image.
- rpoisline: Also returns information about the original infinite random lines.
- rStrauss, rHardcore, rStraussHard, rDiggleGratton, rDGS, rPenttinen: New argument drop.

- `rthin` Accelerated, when `P` is a single number.
- `rThomas, rMatClust, rCauchy, rVarGamma`:
  - When the model is approximately Poisson, it is simulated using rpoispp. This avoids computations which would require huge amounts of memory. New argument `poisthresh` controls this behaviour.
  - New argument `saveparents`.
- Simulation: Several basic simulation algorithms have been accelerated. Consequently, simulation outcomes are not identical to those obtained with previous versions of `spatstat`, even when the same random seed is used. To ensure compatibility with previous versions of spatstat, revert to the slower code by setting `spatstat.options(fastthin=FALSE, fastpois=FALSE)`.
- `simulate.ppm` New argument `w` controls the window of the simulated patterns. New argument `verbose`.
- `Smooth.ppp`:
  - A non-Gaussian kernel can now be specified using the argument `kernel`.
  - Argument `weights` can now be a pixel image.
  - Accelerated by about 30% in the case where `at="pixels"`.
  - Accelerated by about 15% in the case where `at="points"` and `kernel="gaussian"`.
- `spatstat.options` New options `fastthin` and `fastpois` enable fast simulation algorithms. Set these options to `FALSE` to reproduce results obtained with previous versions of `spatstat`.
- `split.ppp` The splitting variable `f` can now be a logical vector.
- `step`: now works for models of class `"mppm"`.
- `subset.ppp`, `subset.lpp`, `subset.pp3`, `subset.ppx`: The argument `subset` can now be any argument acceptable to the `"["` method.
- summary functions The argument `correction="all"` is now recognised: it selects all the available options.

  This applies to `Fest`, `F3est`, `Gest`, `Gcross`, `Gdot`, `Gmulti`, `G3est`, `Gfox`, `Gcom`, `Gres`, `Hest`, `Jest`, `Jmulti`, `Jcross`, `Jdot`, `Jfox`, `Kest`, `Kinhom`, `Kmulti`, `Kcross`, `Kdot`, `Kcom`, `Kres`, `Kmulti.inhom`, `Kcross.inhom`, `Kdot.inhom`, `Kscaled`, `Ksector`, `Kmark`, `K3est`, `Lscaled`, `markcorr`, `markcrosscorr`, `nnorient`, `pairorient`, `pcfinhom`, `pcfcross.inhom`, `pcfcross`, `pcf`, `Tstat`.
- `summary.ppm`: New argument `fine` selects the algorithm for variance estimation.
- `summary.owin`, `summary.im`: The fraction of frame area that is occupied by the window/image is now reported.
- `sumouter`: New argument `y` allows computation of asymmetric outer products.
- `symbolmap`:
  - Now accepts a vector of colour values for the arguments `col`, `cols`, `fg`, `bg` if the argument `range` is given.

  - New option: `shape="arrows"`.

- `tess`: Argument `window` is ignored when xgrid, ygrid are given.
- `texturemap`: Argument `textures` can be missing or NULL.
- `textureplot`: Argument `x` can now be something acceptable to `as.im`.
- `to.grey` New argument `transparent`.
- `union.owin`: Improved behaviour when there are more than 2 windows.
- `update`: now works for models of class `"mppm"`.
- `update.kppm`: Now handles additional arguments in any order, with or without names. Changed arguments. Improved behaviour.
- `valid.ppm` This is now a method for the generic function `valid`.
- `vcov.mppm`: Now handles models with Gibbs interactions.
- `vcov.ppm`: Performance slightly improved, for Gibbs models.
- `[<-.im` Accepts an array for `value`.
- `[.im` The subset index `i` can now be a linear network. Then the result of `x[i, drop=FALSE]` is a pixel image of class `linim`.
- `[.linnet`, `[.lpp`: New argument `snip` determines what to do with segments of the network that cross the boundary of the window. Default behaviour has changed.
- `[.ppx`: The subset index `i` may now be a spatial domain of class `boxx` or `box3`.
- `[.ppp` New argument `clip` determines whether the window is clipped.
- `[.ppp` The previously-unused argument `drop` now determines whether to remove unused levels of a factor.
- `[.pp3`, `[.lpp`, `[.ppx`, `subset.ppp`, `subset.pp3`, `subset.lpp`, `subset.ppx`: These methods now have an argument `drop` which determines whether to remove unused levels of a factor.

6 Serious Bugs Fixed

Hundreds of bugs have been detected and fixed in `spatstat`. Bugs that may have affected the user are listed in the package `NEWS` file. To read all these bug reports, type

```
> news(grepl("^BUG", Category), package="spatstat")
```

which currently produces a list of 597 bugs, of which 129 were detected after publication of the book [2].

Following is a list of the **most serious bugs** only, in order of potential impact.

- `nncross.ppp`: Results were completely incorrect if $k > 1$.
  (Bug introduced in `spatstat 1.31-2`, april 2013; fixed in `spatstat 1.35-0`, december 2013)

- `nncross.pp3`: Results were completely incorrect in some cases.
  (Bug introduced in `spatstat 1.32-0`, august 2013; fixed in `spatstat 1.34-0`, october 2013)

- `cdf.test.ppm`: Calculation of p-values was incorrect for Gibbs models: $1 - p$ was computed instead of p.
  (Bug introduced in `spatstat 1.40-0`, december 2014; fixed in `spatstat 1.45-2`, may 2016)

- `Smooth.ppp`: Results of `Smooth(X, at="points", leaveoneout=FALSE)` were completely incorrect.
  (Bug introduced in `spatstat 1.20-5`, august 2010; fixed in `spatstat 1.46-0`, july 2016)

- `rmh`:

  - Simulation was completely incorrect in the case of a multitype point process with an interaction that does not depend on the marks, such as `ppm(betacells, ~marks, Strauss(60))` due to a coding error in the `C` interface.
    (Bug introduced in `spatstat 1.22-3`, march 2010; fixed in `spatstat 1.22-3`, june 2011)
  - Simulation of the Area-Interaction model was completely incorrect.
    (Bug introduced in `spatstat 1.23-6`, october 2011; fixed in `spatstat 1.31-0`, january 2013)
  - Simulation of the Geyer saturation process was completely incorrect.
    (Bug introduced in `spatstat 1.31-0`, january 2013; fixed in `spatstat 1.31-1`, march 2013)
  - Simulation of the Strauss-Hard Core process was partially incorrect, giving point patterns with a slightly lower intensity.
    (Bug introduced in `spatstat 1.31-0`, january 2013; fixed in `spatstat 1.37-0`, may 2014)
  - The result of simulating a model with a hard core did not necessarily respect the hard core constraint, and simulation of a model with strong inhibition did not necessarily converge. This only happened if the first order trend was large, the starting state (`n.start` or `x.start`) was not given, and the number of iterations `nrep` was not very large. It occurred because of a poor choice for the default starting state. (Bug was present since about 2010. Fixed in `spatstat 1.40-0`, december 2014)
  - Simulation was incorrect in the case of an inhomogeneous multitype model with `fixall=TRUE` (i.e. with a fixed number of points of each type) if the model was segregated (i.e. if different types of points had different first order trend). The effect of the error was that all types of points had the same first order trend. (Bug was present since about 2010. Fixed in `spatstat 1.43-0`, september 2015)
  - Simulation of the Geyer saturation process was incorrectly initialised, so that the results of a short run (i.e. small value of `nrep`) were incorrect, while long runs were correct.
    (Bug introduced in `spatstat 1.17-0`, october 2009; fixed in `spatstat 1.31-1`, march 2013)

- `rVarGamma`: Simulations were incorrect; they were generated using the wrong value of the parameter `nu.ker`.
  (Bug introduced in `spatstat 1.25-0`, december 2011; fixed in `spatstat 1.35-0`, december 2013)

- `rCauchy`: Simulations were incorrect; they were generated using the wrong value of the parameter `omega`.
  (Bug introduced in `spatstat 1.25-0`, december 2011; fixed in `spatstat 1.25-2`, january 2012)

- `lppm`: For multitype patterns, the fitted model was completely incorrect due to an error in constructing the quadrature scheme.
  (Bug introduced in `spatstat 1.23-0`, july 2011; fixed in `spatstat 1.30-0`, december 2012)

- `[.lpp`: The local coordinate `seg` was completely incorrect, when `i` was a window.
  (Bug introduced in `spatstat 1.31-2`, april 2013; fixed in `spatstat 1.45-0`, march 2016)

- `leverage.ppm`, `influence.ppm`, `dfbetas.ppm`: Results were incorrect for non-Poisson processes.
  (Bug introduced in `spatstat 1.25-0`, december 2011; fixed in `spatstat 1.45-0`, march 2016)

- `envelope.ppm`: If the model was an inhomogeneous Poisson process, the resulting envelope object was incorrect (the simulations were correct, but the envelopes were calculated assuming the model was CSR).
  (Bug introduced in `spatstat 1.23-5`, september 2011; fixed in `spatstat 1.23-6`, october 2011)

- `linearK`, `linearpcf`, `linearKinhom`, `linearpcfinhom` and multitype versions: These functions were sometimes greatly underestimated when the network had segments shorter than 10 coordinate units.
  (Bug introduced in `spatstat 1.44-0`, december 2015; fixed in `spatstat 1.46-2`, july 2016)

- `nncross`, `distfun`, `AreaInter`: Results of `nncross` were possibly incorrect when `X` and `Y` did not have the same window. This bug affected values of `distfun` and may also have affected ppm objects with interaction `AreaInter`.
  (Bug introduced in `spatstat 1.9-4`, june 2006; fixed in `spatstat 1.25-2`, january 2012)

- `update.kppm`: If the call to `update` did not include a formula argument or a point pattern argument, then all arguments were ignored. Example: `update(fit, improve.type="quasi")` was identical to `fit`.
  (Bug introduced in `spatstat 1.42-2`, june 2015; fixed in `spatstat 1.45-0`, march 2016)

- `markcorrint`: Results were completely incorrect.
  (Bug introduced in `spatstat 1.39-0`, october 2014; fixed in `spatstat 1.40-0`, december 2014)

- `density.ppp`: Values of `density(X, at="points")` and `Smooth(X, at="points")` were sometimes incorrect, due to omission of the contribution from the data point with the smallest x coordinate.
  (Bug introduced in `spatstat 1.26-0`, april 2012; fixed in `spatstat 1.46-1`, july 2016)

- `update.ppm`: If the argument `Q` was given, the results were usually incorrect, or an error was generated.
  (Bug introduced in `spatstat 1.38-0`, august 2014; fixed in `spatstat 1.38-1`, august 2014)

- `subfits`: The interaction coefficients of the submodels were incorrect for Gibbs models with a multitype interaction (`MultiStrauss`, etc).
  (Bug introduced in `spatstat 1.35-0`, december 2013; fixed in `spatstat 1.45-2`, may 2016)

- `F3est`: Estimates of $F(r)$ for the largest value of r were wildly incorrect. (Bug was present since about 2010. Fixed in `spatstat 1.48-0`, december 2016)

- `kppm`, `matclust.estpcf`, `pcfmodel`: The pair correlation function of the Mátern Cluster Process was evaluated incorrectly at distances close to 0. This could have affected the fitted parameters in `matclust.estpcf()` or `kppm(clusters="MatClust")`.
  (Bug introduced in `spatstat 1.20-2`, august 2010; fixed in `spatstat 1.33-0`, september 2013)

- `ppm`: Results were incorrect for the Geyer saturation model with a non-integer value of the saturation parameter `sat`.
  (Bug introduced in `spatstat 1.20-0`, july 2010; fixed in `spatstat 1.31-2`, april 2013)

- `clip.infline`: Results were incorrect unless the midpoint of the window was the coordinate origin.
  (Bug introduced in `spatstat 1.15-1`, april 2009; fixed in `spatstat 1.48-0`, december 2016)

- `intensity.ppm`: Result was incorrect for Gibbs models if the model was exactly equivalent to a Poisson process (i.e. if all interaction coefficients were exactly zero).
  (Bug introduced in `spatstat 1.28-1`, june 2012; fixed in `spatstat 1.47-0`, october 2016)

- `funxy`: Did not correctly handle one-line functions. The resulting objects evaluated the wrong function in some cases.
  (Bug introduced in `spatstat 1.45-0`, march 2016; fixed in `spatstat 1.46-0`, july 2016)

- `selfcrossing.psp`: y coordinate values were incorrect.
  (Bug introduced in `spatstat 1.23-2`, august 2011; fixed in `spatstat 1.25-3`, february 2012)

- `Geyer`: For point process models with the `Geyer` interaction, `vcov.ppm` and `suffstat` sometimes gave incorrect answers.
  (Bug introduced in `spatstat 1.27-0`, may 2012; fixed in `spatstat 1.30-0`, december 2012)

- `vcov.ppm`, `suffstat`: These functions sometimes gave incorrect values for marked point process models.
  (Bug introduced in `spatstat 1.27-0`, may 2012; fixed in `spatstat 1.29-0`, october 2012)

- `diagnose.ppm`: When applied to a model obtained from `subfits()`, in the default case (`oldstyle=FALSE`) the variance calculations were incorrect. Consequently the dotted lines representing significance bands were incorrect. An error or warning about negative variances occurred sometimes. However, calculations with `oldstyle=TRUE` were correct. The default has now been changed to `oldstyle=TRUE` for such models.
  (Bug introduced in `spatstat 1.35-0`, december 2013; fixed in `spatstat 1.45-0`, march 2016)

- `Smooth.ppp`: Results for `at="points"` were garbled, for some values of `sigma`, if `X` had more than one column of marks.
  (Bug introduced in `spatstat 1.38-0`, october 2014; fixed in `spatstat 1.46-0`, july 2016)

- `linearK`, `linearKinhom`: If any data points were located exactly at a vertex of the linear network, the weights for Ang's correction were incorrect, due to numerical error. This sometimes produced infinite or NA values of the linear K function.
  (Bug introduced in `spatstat 1.23-0`, july 2011; fixed in `spatstat 1.27-0`, may 2012)

- `Kinhom`, `Linhom`: the results were not renormalised (even if `renormalise=TRUE`) in some cases.
  (Bug introduced in `spatstat 1.21-0`, december 2010; fixed in `spatstat 1.37-0`, may 2014)

- `Kinhom`, `Linhom`: Ignored argument `reciplambda2` in some cases.
  (Bug introduced in `spatstat 1.39-0`, october 2014; fixed in `spatstat 1.40-0`, december 2014)

- `Kinhom`, `Linhom`: Calculations were incorrect if `lambda` was a fitted point process model.
  (Bug introduced in `spatstat 1.38-0`, august 2014; fixed in `spatstat 1.38-1`, august 2014)

- `integral.linim`, `integral.linfun`:

  – results were inaccurate because of a bias in the distribution of sample points.
    (Bug introduced in `spatstat 1.41-0`, february 2015; fixed in `spatstat 1.47-0`, october 2016)

  - results were inaccurate if many of the segment lengths were shorter than the width of a pixel.
    (Bug introduced in `spatstat 1.41-0`, february 2015; fixed in `spatstat 1.48-0`, december 2016)

- `predict.ppm`: Calculation of the conditional intensity omitted the edge correction if `correction='translate'` or `correction='periodic'`.
  (Bug introduced in `spatstat 1.17-0`, october 2009; fixed in `spatstat 1.31-3`, may 2013)

- `varblock`: Calculations were incorrect if more than one column of edge corrections was computed.
  (Bug introduced in `spatstat 1.21-1`, november 2010; fixed in `spatstat 1.39-0`, october 2014)

- `scan.test` Results were sometimes incorrect due to numerical instability (a 'Gibbs phenomenon').
  (Bug introduced in `spatstat 1.24-1`, october 2011; fixed in `spatstat 1.26-1`, april 2012)

- `relrisk`: When `at="pixels"`, a small fraction of pixel values were sometimes wildly inaccurate, due to numerical errors. This affected the range of values in the result, and therefore the appearance of plots. (Bug fixed in `spatstat 1.40-0`, december 2014)

- `predict.slrm`: Results of `predict(object, newdata)` were incorrect if the spatial domain of `newdata` was larger than the original domain.
  (Bug introduced in `spatstat 1.21-0`, november 2010; fixed in `spatstat 1.25-3`, february 2012)

- `Lest`: The variance approximations (Lotwick-Silverman and Ripley) obtained with `var.approx=TRUE` were incorrect for `Lest` (although they were correct for `Kest`) due to a coding error.
  (Bug introduced in `spatstat 1.24-1`, october 2011; fixed in `spatstat 1.24-2`, november 2011)

- `bw.diggle`: Bandwidth was too large by a factor of 2.
  (Bug introduced in `spatstat 1.23-4`, september 2011; fixed in `spatstat 1.23-5`, september 2011)

- pair correlation functions (`pcf.ppp`, `pcfdot`, `pcfcross` etc:) The result had a negative bias at the maximum r value, because contributions to the pcf estimate from interpoint distances greater than `max(r)` were mistakenly omitted. (Bugs fixed in `spatstat 1.35-0`, december 2013)

- `Kest`, `Lest`: Gave incorrect values in very large datasets, due to numerical overflow. 'Very large' typically means about 1 million points in a random pattern, or 100,000 points in a tightly clustered pattern. [Overflow cannot occur unless there are at least 46,341 points.]

- `bw.relrisk`: Implementation of `method="weightedleastsquares"` was incorrect and was equivalent to `method="leastsquares"`.
  (Bug introduced in `spatstat 1.21-0`, november 2010; fixed in `spatstat 1.23-4`, september 2011)

- `triangulate.owin`: Results were incorrect in some special cases.
  (Bug introduced in `spatstat 1.42-2`, june 2015; fixed in `spatstat 1.44-0`, december 2015)

- `crosspairs`: If `X` and `Y` were identical point patterns, the result was not necessarily symmetric (on some machines) due to numerical artifacts.
  (Bug introduced in `spatstat 1.35-0`, december 2013; fixed in `spatstat 1.44-0`, december 2015)

- `bdist.tiles`: Values were incorrect in some cases due to numerical error. (Bug fixed in `spatstat 1.29-0`, october 2012)

- `Kest.fft`: Result was incorrectly normalised.
  (Bug introduced in `spatstat 1.21-2`, january 2011; fixed in `spatstat 1.44-0`, december 2015)

- `crossdist.ppp`: Ignored argument `squared` if `periodic=FALSE`. (Bug fixed in `spatstat 1.38-0`, july 2014)
- polygon geometry: The point-in-polygon test gave the wrong answer in some boundary cases. (Bug fixed in `spatstat 1.23-2`, august 2011)
- `MultiStraussHard`: If a fitted model with `MultiStraussHard` interaction was invalid, `project.ppm` sometimes yielded a model that was still invalid. (Bug fixed in `spatstat 1.42-0`, may 2015)
- `pool.envelope`: Did not always respect the value of `use.theory`.
  (Bug introduced in `spatstat 1.23-5`, september 2011; fixed in `spatstat 1.43-0`, september 2015)
- `nncross.lpp`, `nnwhich.lpp`, `distfun.lpp`: Sometimes caused a segmentation fault.
  (Bug introduced in `spatstat 1.44-0`, december 2015; fixed in `spatstat 1.44-1`, december 2015)
- `anova.ppm`: If a single `object` was given, and it was a Gibbs model, then `adjust` was effectively set to `FALSE`.
  (Bug introduced in `spatstat 1.39-0`, october 2014; fixed in `spatstat 1.44-1`, december 2015)

References

[1] A. Baddeley. Analysing spatial point patterns in R. Technical report, CSIRO, 2010. Version 4. URL `https://research.csiro.au/software/r-workshop-notes/`

[2] A. Baddeley, E. Rubak, and R. Turner. *Spatial Point Patterns: Methodology and Applications with R*. Chapman & Hall/CRC Press, 2015.