SimLGCP

Package Installation

```r
library(mgcv)
```

```
Warning: package 'mgcv' was built under R version 4.3.3

Loading required package: nlme

This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```r
library(spatstat)
```

```
Loading required package: spatstat.data

Warning: package 'spatstat.data' was built under R version 4.3.3

Loading required package: spatstat.univar

spatstat.univar 3.1-3

Loading required package: spatstat.geom

spatstat.geom 3.4-1

Loading required package: spatstat.random
```

```
spatstat.random 3.4-1

Loading required package: spatstat.explore

spatstat.explore 3.4-3

Loading required package: spatstat.model

Loading required package: rpart

spatstat.model 3.3-6

Loading required package: spatstat.linnet

spatstat.linnet 3.2-6


spatstat 3.3-3
For an introduction to spatstat, type 'beginner'
```

Simulating LGCP Data Using SpatStat

Define the spatial window

Using unit square for this example

```r
win <- owin(xrange = c(0,1), yrange = c(0,1))
```

Define Covariate Functions

```r
cov_x <- as.im(function(x,y) x, W = win)
cov_y <- as.im(function(x,y) y, W = win)
cov_z <- as.im(function(x,y) x+y+5, W = win)
cov_a <- as.im(function(x,y) 2*x, W = win)
```

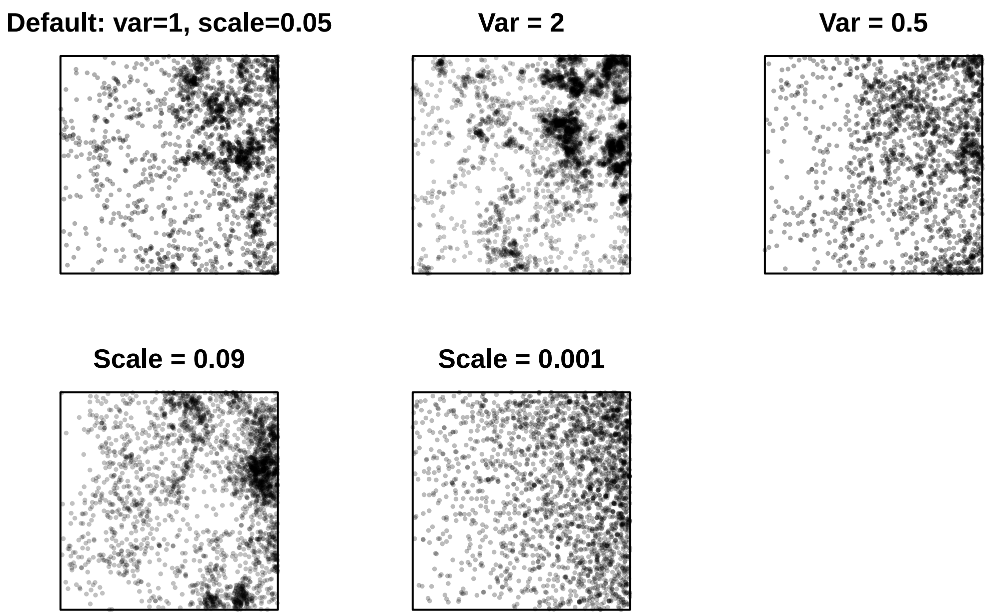
Define Mean Function Based on Covariates

```r
mean <- function(x,y)
  {2*x + x + y + 5}

mean.im <- as.im(mean, W = win)
```

Simulate LGCP using rLGCP()

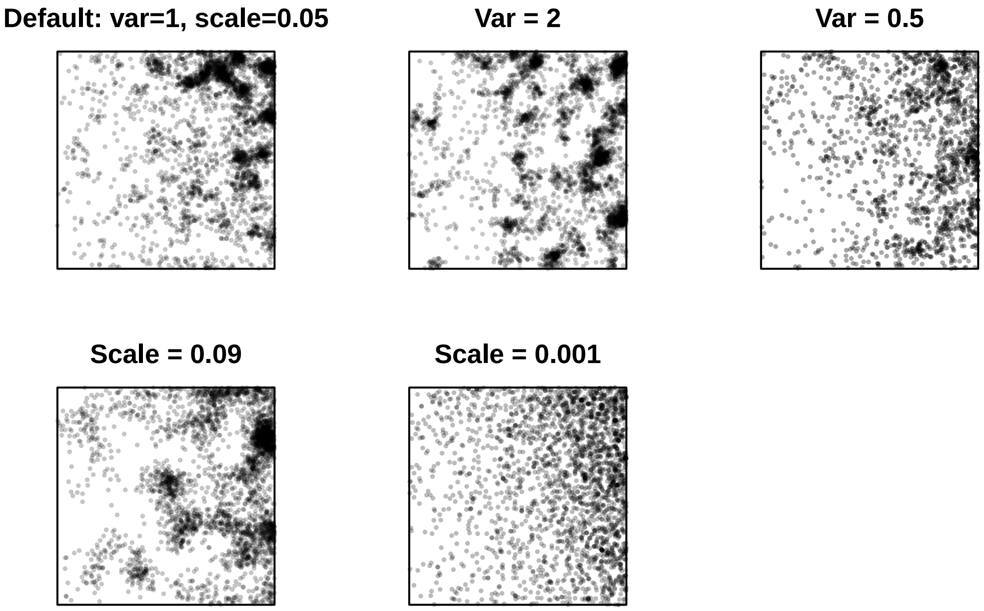
Exponential

```r
set.seed(111)
ppp_exp <- rLGCP("exponential", mu = mean.im, var = 1, scale = 0.05, win = win)

#Change var
ppp_exp_var1 <- rLGCP("exponential", mu = mean.im, var = 2, scale = 0.05, win = win)

ppp_exp_var2 <- rLGCP("exponential", mu = mean.im, var = 0.5, scale = 0.05, win = win)

# Change scale
ppp_exp_scale1 <- rLGCP("exponential", mu = mean.im, var = 1, scale = 0.09, win = win)
# if scale goes above 1, becomes similar to small scale

ppp_exp_scale2 <- rLGCP("exponential", mu = mean.im, var = 1, scale = 0.001, win = win)

par(mfrow = c(2, 3), mar = c(2, 2, 2, 1))

plot(ppp_exp, main = "Default: var=1, scale=0.05", pch = 16, cex = 0.5)
plot(ppp_exp_var1, main = "Var = 2", pch = 16, cex = 0.5)
plot(ppp_exp_var2, main = "Var = 0.5", pch = 16, cex = 0.5)
plot(ppp_exp_scale1, main = "Scale = 0.09", pch = 16, cex = 0.5)
plot(ppp_exp_scale2, main = "Scale = 0.001", pch = 16, cex = 0.5)
```



Gauss

```r
set.seed(112)
ppp_gauss <- rLGCP("gauss", mu = mean.im, var = 1, scale = 0.05, win = win)

#Change var
ppp_gauss_var1 <- rLGCP("gauss", mu = mean.im, var = 2, scale = 0.05, win = win)

ppp_gauss_var2 <- rLGCP("gauss", mu = mean.im, var = 0.5, scale = 0.05, win = win)

# Change scale
ppp_gauss_scale1 <- rLGCP("gauss", mu = mean.im, var = 1, scale = 0.09, win = win)
# if scale goes above 1, becomes similar to small scale

ppp_gauss_scale2 <- rLGCP("gauss", mu = mean.im, var = 1, scale = 0.001, win = win)

par(mfrow = c(2, 3), mar = c(2, 2, 2, 1))
plot(ppp_gauss, main = "Default: var=1, scale=0.05", pch = 16, cex = 0.5)
plot(ppp_gauss_var1, main = "Var = 2", pch = 16, cex = 0.5)
plot(ppp_gauss_var2, main = "Var = 0.5", pch = 16, cex = 0.5)
plot(ppp_gauss_scale1, main = "Scale = 0.09", pch = 16, cex = 0.5)
plot(ppp_gauss_scale2, main = "Scale = 0.001", pch = 16, cex = 0.5)
```



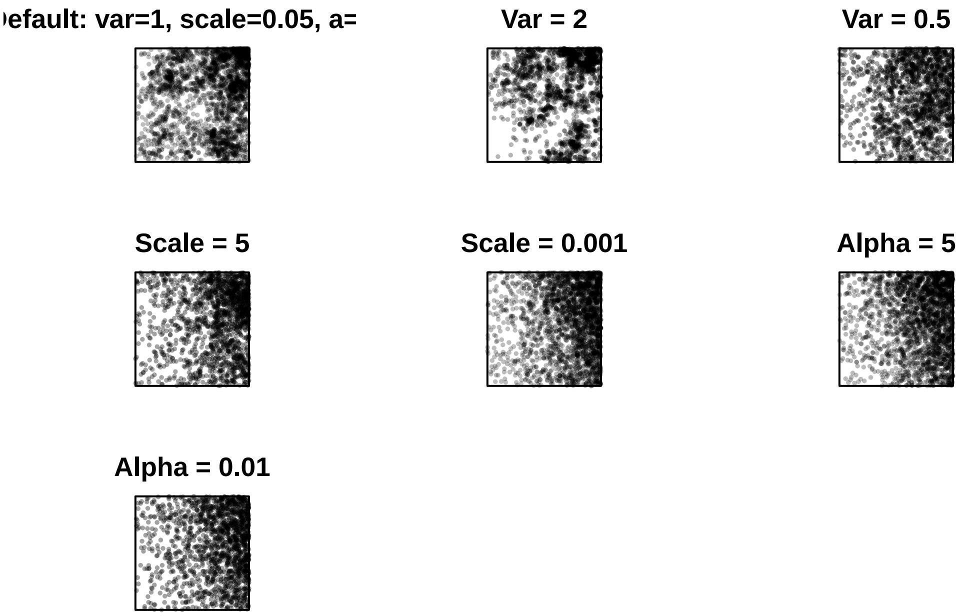
Stable

```
set.seed(113)

ppp_stable <- rLGCP("stable", mu = mean.im, var = 1, scale = 0.05, alpha = 0.5, win = win)

#Change var
ppp_stable_var1 <- rLGCP("stable", mu = mean.im, var = 2, scale = 0.05,alpha = 0.5, win = wi

ppp_stable_var2 <- rLGCP("stable", mu = mean.im, var = 0.5, scale = 0.05, alpha = 0.5,win = 

# Change scale
ppp_stable_scale1 <- rLGCP("stable", mu = mean.im, var = 1, scale = 5, alpha = 0.5,win = win
```

```
Warning: 250 out of 65536 terms (0.381%) in FFT calculation of matrix square
root were negative, and were set to zero. Range: [-9.91, 44800]
```

```
ppp_stable_scale2 <- rLGCP("stable", mu = mean.im, var = 1, scale = 0.001, alpha = 0.5,win = 

# Change alpha
ppp_stable_alpha1 <- rLGCP("stable", mu = mean.im, var = 1, scale = 0.001, alpha = 5,win = wi

ppp_stable_alpha2 <- rLGCP("stable", mu = mean.im, var = 1, scale = 0.001, alpha = 0.01,win =
```

```
par(mfrow = c(3, 3), mar = c(2, 2, 2, 1))

plot(ppp_stable, main = "Default: var=1, scale=0.05,  =0.5", pch = 16, cex = 0.5)
plot(ppp_stable_var1, main = "Var = 2", pch = 16, cex = 0.5)
plot(ppp_stable_var2, main = "Var = 0.5", pch = 16, cex = 0.5)
plot(ppp_stable_scale1, main = "Scale = 5", pch = 16, cex = 0.5)
plot(ppp_stable_scale2, main = "Scale = 0.001", pch = 16, cex = 0.5)
plot(ppp_stable_alpha1, main = "Alpha = 5", pch = 16, cex = 0.5)
plot(ppp_stable_alpha2, main = "Alpha = 0.01", pch = 16, cex = 0.5)
```



Generalised Cauchy

```
set.seed(114)

ppp_gencauchy <- rLGCP("gencauchy", mu = mean.im, var = 1, scale = 0.05, alpha = 1, beta = 1

#Change var
ppp_gencauchy_var1 <- rLGCP("gencauchy", mu = mean.im, var = 2, scale = 0.05, alpha = 1, beta

ppp_gencauchy_var2 <- rLGCP("gencauchy", mu = mean.im, var = 0.5, scale = 0.05, alpha = 1, be

#Change scale
ppp_gencauchy_scale1 <- rLGCP("gencauchy", mu = mean.im, var = 1, scale = 5, alpha = 1, beta
```

```
Warning: 978 out of 65536 terms (1.49%) in FFT calculation of matrix square
root were negative, and were set to zero. Range: [-114, 57000]
```

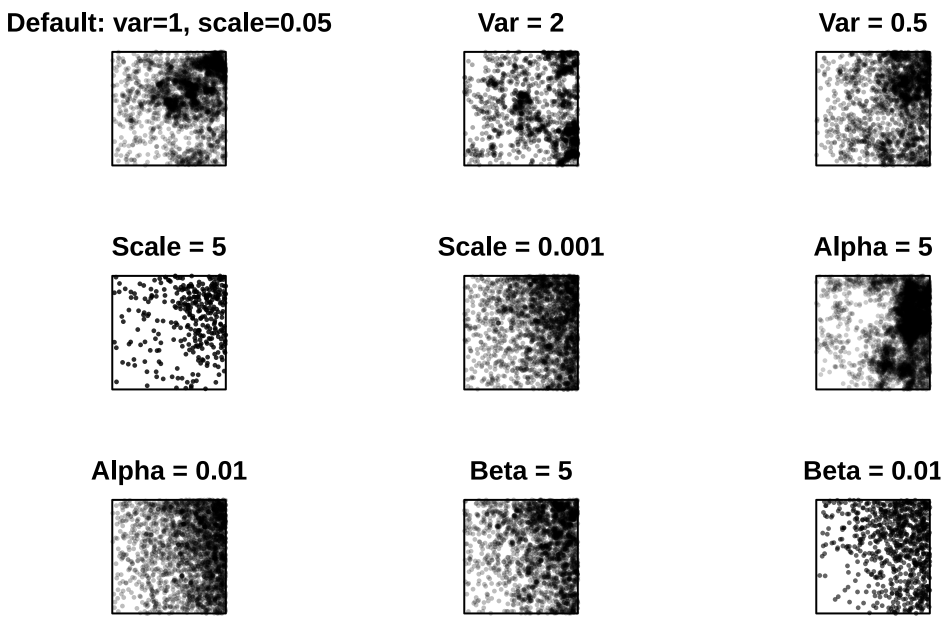
```
ppp_gencauchy_scale2 <- rLGCP("gencauchy", mu = mean.im, var = 1, scale = 0.001, alpha = 1, b

#Change alpha
ppp_gencauchy_alpha1 <- rLGCP("gencauchy", mu = mean.im, var = 1, scale = 0.05, alpha = 5, be
```

```
Warning: 28350 out of 65536 terms (43.3%) in FFT calculation of matrix square
root were negative, and were set to zero. Range: [-2.99, 5630]
```

```
ppp_gencauchy_alpha2 <- rLGCP("gencauchy", mu = mean.im, var = 1, scale = 0.05, alpha = 0.01

#Change beta
ppp_gencauchy_beta1 <- rLGCP("gencauchy", mu = mean.im, var = 1, scale = 0.05, alpha = 1, bet

ppp_gencauchy_beta2 <- rLGCP("gencauchy", mu = mean.im, var = 1, scale = 0.05, alpha = 1, bet
```

```
Warning: 370 out of 65536 terms (0.565%) in FFT calculation of matrix square
root were negative, and were set to zero. Range: [-0.0654, 63800]
```

```
par(mfrow = c(3, 3), mar = c(2, 2, 2, 1))

plot(ppp_gencauchy, main = "Default: var=1, scale=0.05", pch = 16, cex = 0.5)
plot(ppp_gencauchy_var1, main = "Var = 2", pch = 16, cex = 0.5)
plot(ppp_gencauchy_var2, main = "Var = 0.5", pch = 16, cex = 0.5)
plot(ppp_gencauchy_scale1, main = "Scale = 5", pch = 16, cex = 0.5)
plot(ppp_gencauchy_scale2, main = "Scale = 0.001", pch = 16, cex = 0.5)
plot(ppp_gencauchy_alpha1, main = "Alpha = 5", pch = 16, cex = 0.5)
plot(ppp_gencauchy_alpha2, main = "Alpha = 0.01", pch = 16, cex = 0.5)
plot(ppp_gencauchy_beta1, main = "Beta = 5", pch = 16, cex = 0.5)
plot(ppp_gencauchy_beta2, main = "Beta = 0.01", pch = 16, cex = 0.5)
```

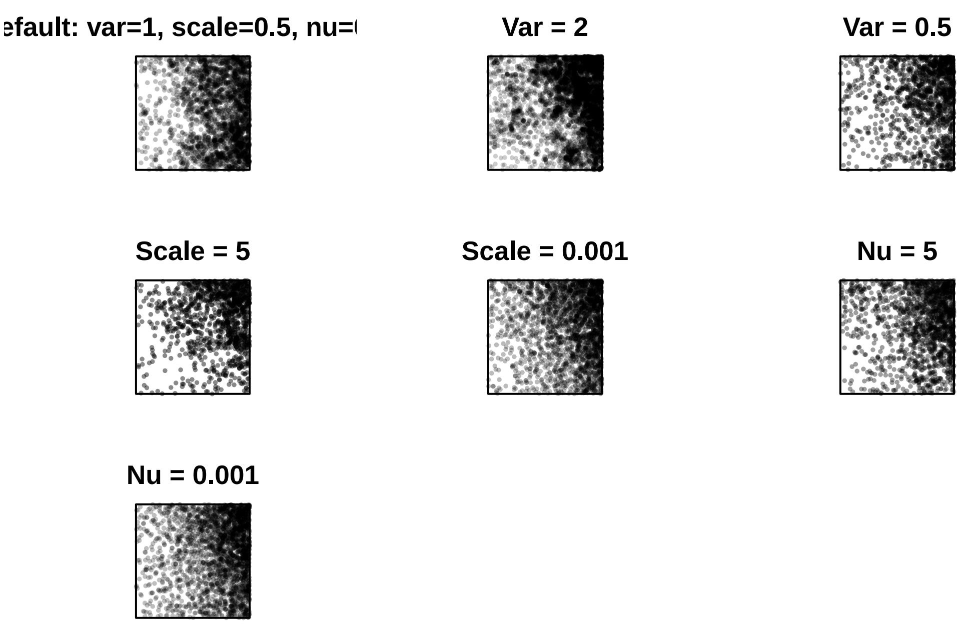


Matern

```
set.seed(115)

ppp_matern <- rLGCP("matern", mu = mean.im, var = 1, scale = 0.5, nu = 0.05, win = win)

#Change var
ppp_matern_var1 <- rLGCP("matern", mu = mean.im, var = 2, scale = 0.5, nu = 0.05, win = win)

ppp_matern_var2 <- rLGCP("matern", mu = mean.im, var = 0.5, scale = 0.5, nu = 0.05, win = wi

#Change scale
ppp_matern_scale1 <- rLGCP("matern", mu = mean.im, var = 1, scale = 5, nu = 0.05, win = win)

ppp_matern_scale2 <- rLGCP("matern", mu = mean.im, var = 1, scale = 0.001, nu = 0.05, win =

#Change nu
ppp_matern_nu1 <- rLGCP("matern", mu = mean.im, var = 1, scale = 0.5, nu = 5, win = win)
```

```
Warning: 32746 out of 65536 terms (50%) in FFT calculation of matrix square
root were negative, and were set to zero. Range: [-114, 22300]
```

```
ppp_matern_nu2 <- rLGCP("matern", mu = mean.im, var = 1, scale = 0.5, nu = 0.001, win = win)

par(mfrow = c(3, 3), mar = c(2, 2, 2, 1))


plot(ppp_matern, main = "Default: var=1, scale=0.5, nu=0.05", pch = 16, cex = 0.5)
plot(ppp_matern_var1, main = "Var = 2", pch = 16, cex = 0.5)
plot(ppp_matern_var2, main = "Var = 0.5", pch = 16, cex = 0.5)
plot(ppp_matern_scale1, main = "Scale = 5", pch = 16, cex = 0.5)
plot(ppp_matern_scale2, main = "Scale = 0.001", pch = 16, cex = 0.5)
plot(ppp_matern_nu1, main = "Nu = 5", pch = 16, cex = 0.5)
plot(ppp_matern_nu2, main = "Nu = 0.001", pch = 16, cex = 0.5)
```



Simulate Covariate Values Using Defined Functions for Observed Points

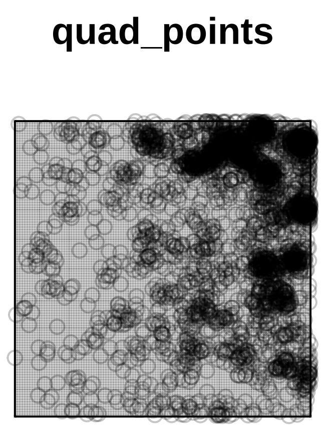
Using Gaussian model for rest of fitting (but can use any)

```
lgcp_df_gauss <- as.data.frame(ppp_gauss)

lgcp_df_gauss$cov_x <- cov_x[lgcp_df_gauss]
lgcp_df_gauss$cov_y <- cov_y[lgcp_df_gauss]
lgcp_df_gauss$cov_z <- cov_z[lgcp_df_gauss]
lgcp_df_gauss$cov_a <- cov_a[lgcp_df_gauss]
```

```
# Adding pt and wt for the "observed points"
lgcp_df_gauss$pt <- 1
lgcp_df_gauss$wt <- 1e-6
```

Create Grid in Window for Quadrature Points

```
# mask <- as.mask(win, eps = 0.01)

# dummy_points <- gridcentres(mask)

quad_points <- quadscheme(ppp_gauss, method = "grid")

plot(quad_points)
```



Create Quadrature Data Frame

```
quad_df_gauss <- data.frame(
  x = quad_points$dummy$x,
  y = quad_points$dummy$y,
  pt = 0,
  wt = 6.94e-05 # came from areas vector in quad_points list
)
```

Generate Quadrature Covariates

```
quad_df_gauss$cov_x <- cov_x[quad_df_gauss]
quad_df_gauss$cov_y <- cov_y[quad_df_gauss]
quad_df_gauss$cov_z <- cov_z[quad_df_gauss]
quad_df_gauss$cov_a <- cov_a[quad_df_gauss]
```

Use rbind() to Merge Observed and Quadrature Points Together

```
data_gauss <- rbind(lgcp_df_gauss, quad_df_gauss)
```