Norwegian spruces dataset

```
library(ppjsdm)
#> Registered S3 method overwritten by 'spatstat':
#>   method     from
#>   print.boxx cli
library(spatstat)
#> Loading required package: spatstat.data
#> Loading required package: nlme
#> Loading required package: rpart
#>
#> spatstat 1.63-0       (nickname: 'Space camouflage')
#> For an introduction to spatstat, type 'beginner'
remove(list = ls())

set.seed(1)
```

This vignette explains how to use the `ppjsdm` package with the `spruces` dataset from `spatstat`.

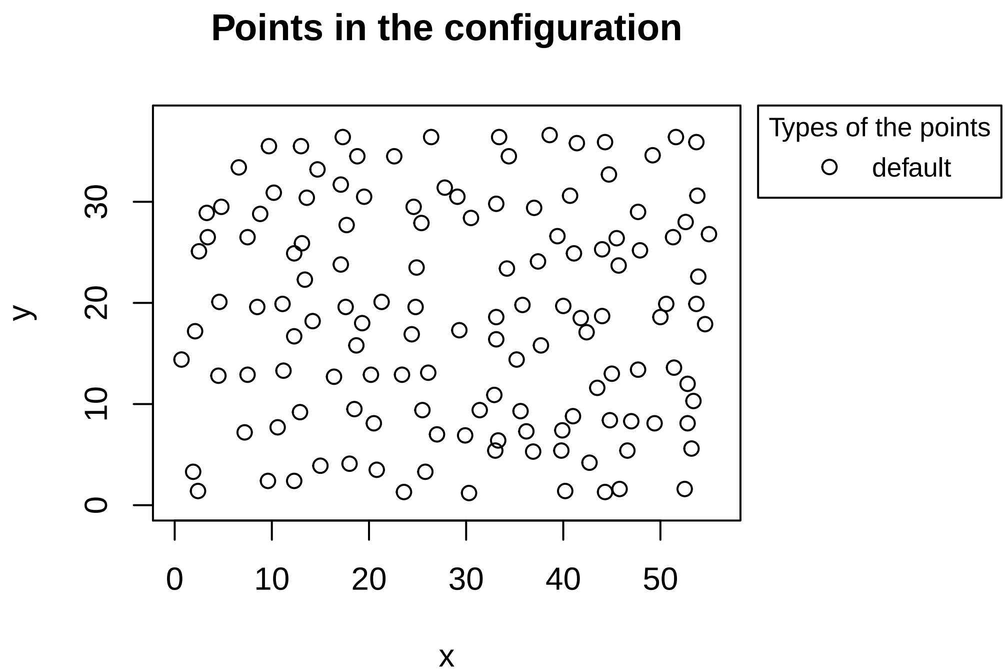
Taking marks into account

If marks are provided, the interaction radii are proportional to the marks. We begin by that setting.

```
configuration <- Configuration(spruces$x, spruces$y, marks = spruces$marks)
window <- Rectangle_window(c(0, 56), c(0, 38))
```

The point configuration is plotted below.

```
print(configuration)
#> An S3 object representing a configuration.
#>
#> Number of points: 134.
par(mar = c(5, 4, 4, 13) + 0.1)
plot(configuration, window = window)
```

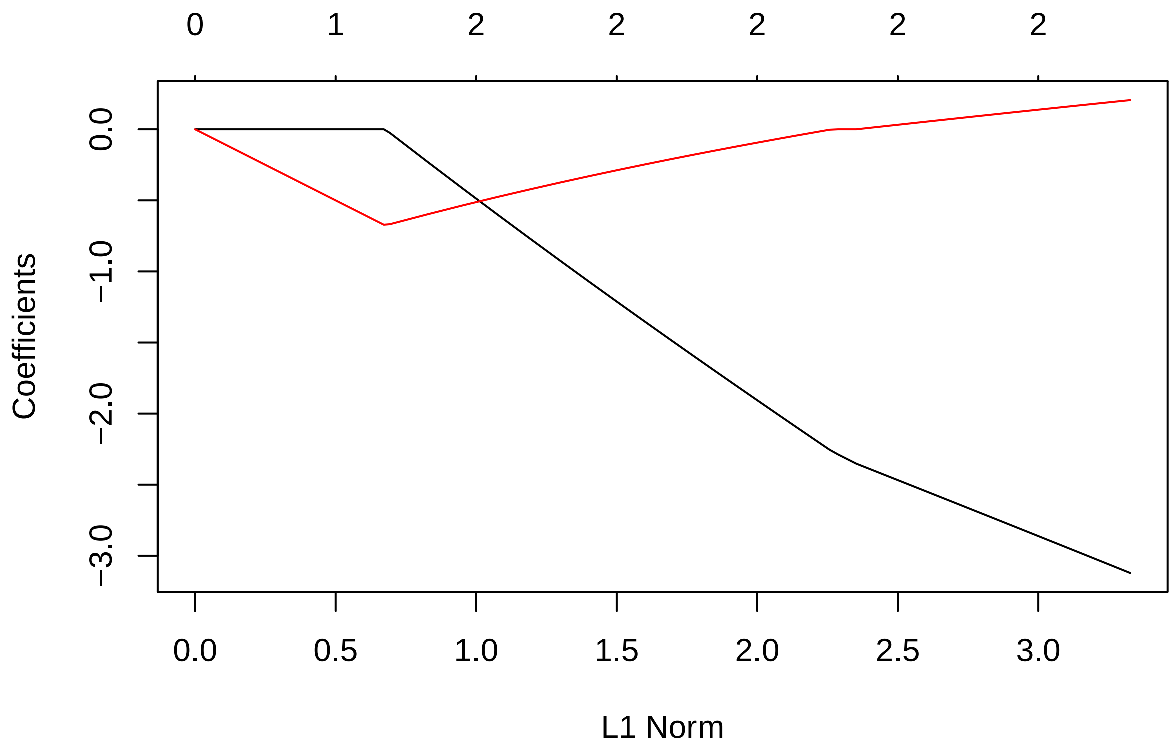


We provide a series of ranges for the interaction radii, and let the fitting function calibrate the model.

```
short_range <- c(0, 20)
medium_range <- c(0, 20)
long_range <- c(0, 20)
model <- "square_exponential"
medium_range_model <- "square_exponential"
```

We can now call the fitting function.

```
fit <- ppjsdm::gibbsm(configuration,
                      window = window,
                      model = model,
                      medium_range_model = medium_range_model,
                      short_range = short_range,
                      medium_range = medium_range,
                      long_range = long_range,
                      use_glmnet = TRUE,
                      use_aic = TRUE,
                      saturation = 2)
#> (Intercept) log_lambda1    alpha_1_1    gamma_1_1
#>    0.000000   -1.378804    -3.121569     0.205435
plot(fit$complete)
```

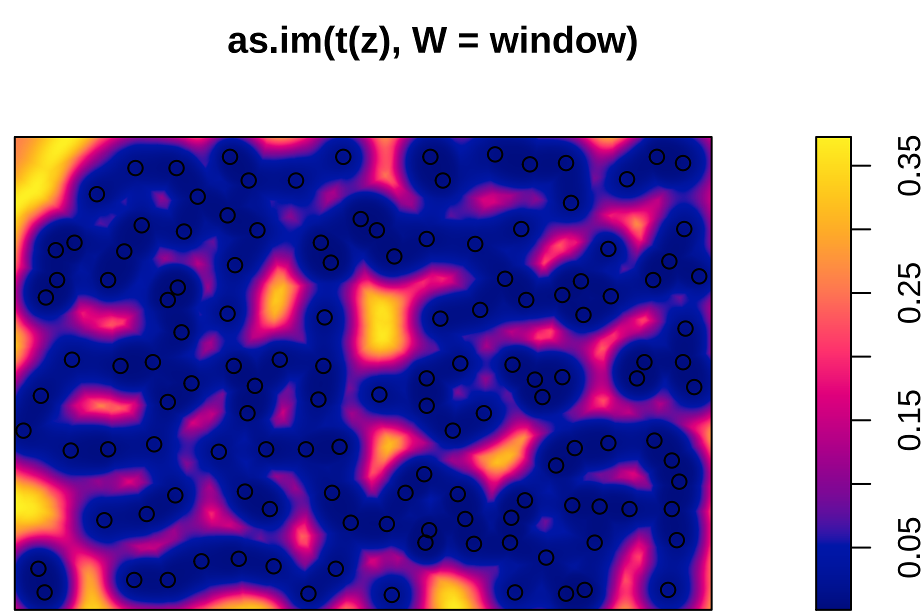



```
print(fit$coefficients)
#> (Intercept) log_lambda1   alpha_1_1   gamma_1_1
#>    0.000000   -1.378804   -3.121569    0.205435
print(fit$best_short)
#>          [,1]
#> [1,] 7.198859
print(fit$best_medium)
#>          [,1]
#> [1,] 16.98333
print(fit$best_long)
#>          [,1]
#> [1,] 23.77533
print(fit$aic)
#> [1] -372.2846
print(fit$bic)
#> [1] -363.288
```

We may then plot the corresponding Papangelou conditional intensity.

```
parameters <- get_parameters_from_fit(fit)
lambda <- parameters$lambda
alpha <- parameters$alpha
gamma <- parameters$gamma
plot_papangelou(window = window,
                configuration = configuration,
                type = 1,
                mark = mean(get_marks(configuration)),
```

```
                  model = model,
                  medium_range_model = medium_range_model,
                  alpha = alpha,
                  lambda = lambda,
                  beta = matrix(0, 1, 0),
                  gamma = gamma,
                  covariates = list(),
                  short_range = fit$best_short,
                  medium_range = fit$best_medium,
                  long_range = fit$best_long,
                  saturation = 2)
```

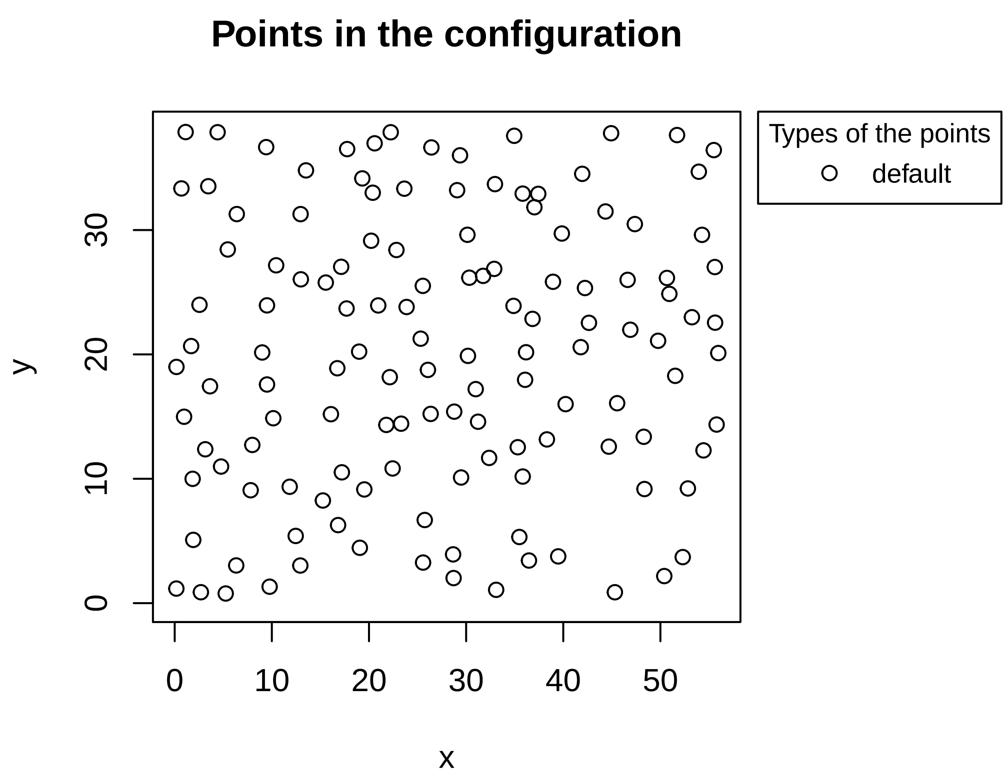


It is also possible to draw from the model.

```
parameters <- get_parameters_from_fit(fit)
lambda <- parameters$lambda
alpha <- parameters$alpha
gamma <- parameters$gamma
draw <- ppjsdm::rgibbs(window = window,
                       alpha = alpha,
                       lambda = lambda,
                       gamma = gamma,
                       model = model,
                       medium_range_model = medium_range_model,
                       short_range = fit$best_short,
                       medium_range = fit$best_medium,
                       long_range = fit$best_long,
```

```
                        types = levels(types(configuration)),
                        mark_range = c(min(get_marks(configuration)), max(get_marks(configuration))),
                        steps = 10000000)
print(draw)
#> An S3 object representing a configuration.
#>
#> Number of points: 130.

par(mar = c(5, 4, 4, 13) + 0.1)
plot(draw, window = window)
```



Interaction radii not proportional to marks

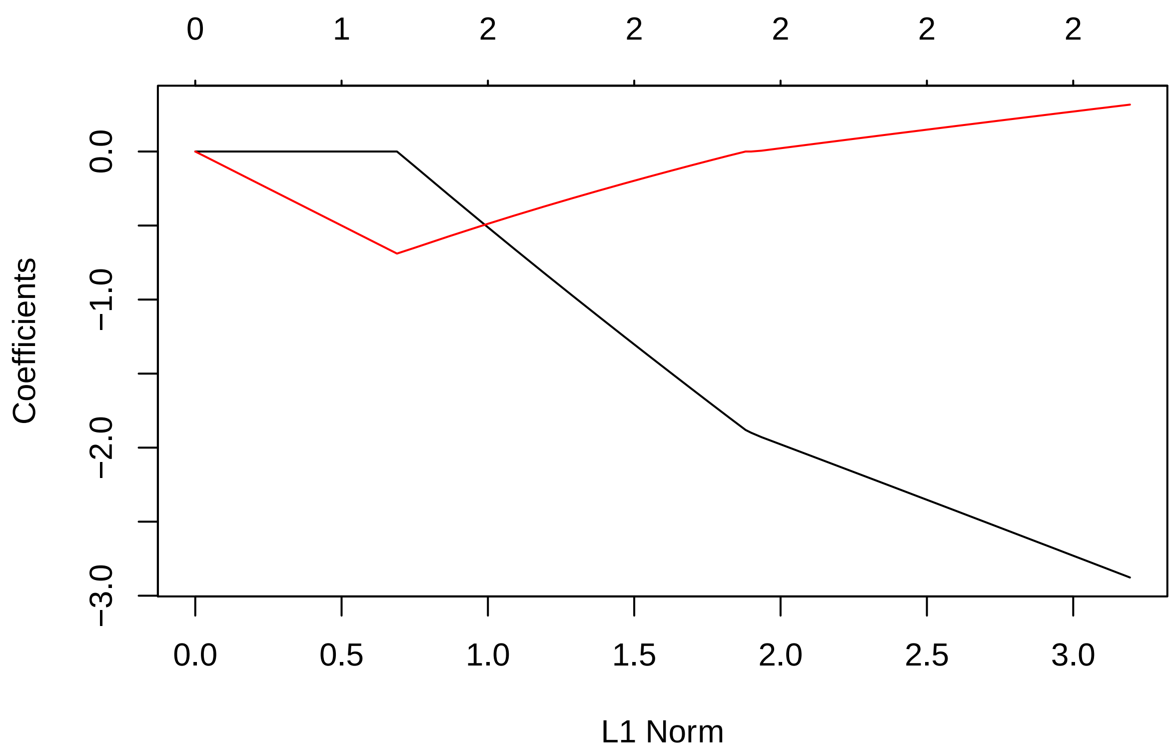
In this section, we do not account for marks.

```
configuration <- Configuration(spruces$x, spruces$y)
```

We call the fitting function on this unmarked point process.

```
fit <- ppjsdm::gibbsm(configuration,
                       window = window,
                       model = model,
                       medium_range_model = medium_range_model,
                       short_range = short_range,
                       medium_range = medium_range,
```

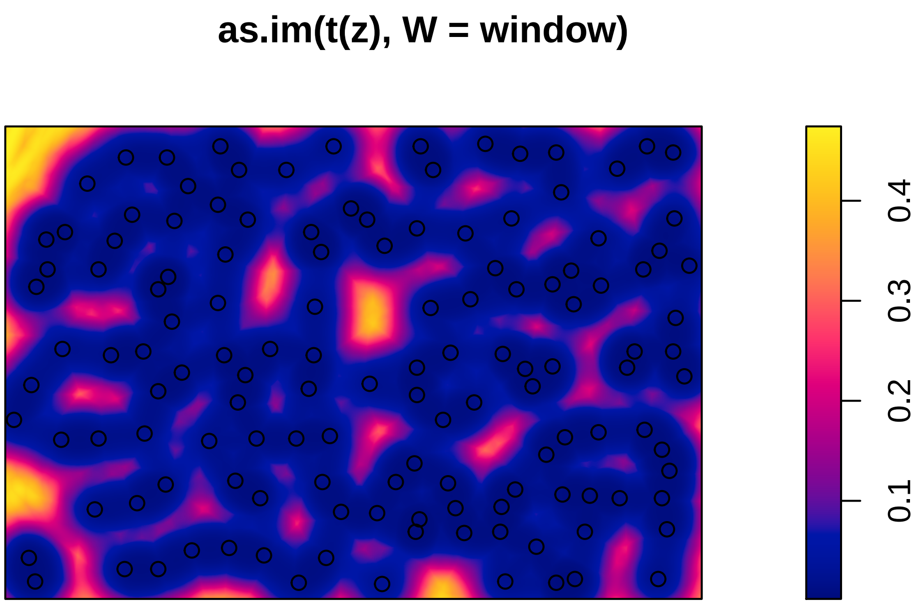
```r
                        long_range = long_range,
                        use_glmnet = TRUE,
                        use_aic = TRUE,
                        saturation = 2)
#> (Intercept) log_lambda1   alpha_1_1   gamma_1_1
#>   0.0000000  -1.3788037  -2.8771354   0.3167827
plot(fit$complete)
```




```r
print(fit$coefficients)
#> (Intercept) log_lambda1   alpha_1_1   gamma_1_1
#>   0.0000000  -1.3788037  -2.8771354   0.3167827
print(fit$best_short)
#>          [,1]
#> [1,] 2.048799
print(fit$best_medium)
#>          [,1]
#> [1,] 13.92174
print(fit$best_long)
#>         [,1]
#> [1,] 15.5096
print(fit$aic)
#> [1] -354.9212
print(fit$bic)
#> [1] -345.9246
```

We may then plot the corresponding Papangelou conditional intensity.

```r
parameters <- get_parameters_from_fit(fit)
lambda <- parameters$lambda
alpha <- parameters$alpha
gamma <- parameters$gamma
plot_papangelou(window = window,
                configuration = configuration,
                type = 1,
                mark = mean(get_marks(configuration)),
                model = model,
                medium_range_model = medium_range_model,
                alpha = alpha,
                lambda = lambda,
                beta = matrix(0, 1, 0),
                gamma = gamma,
                covariates = list(),
                short_range = fit$best_short,
                medium_range = fit$best_medium,
                long_range = fit$best_long,
                saturation = 2)
```

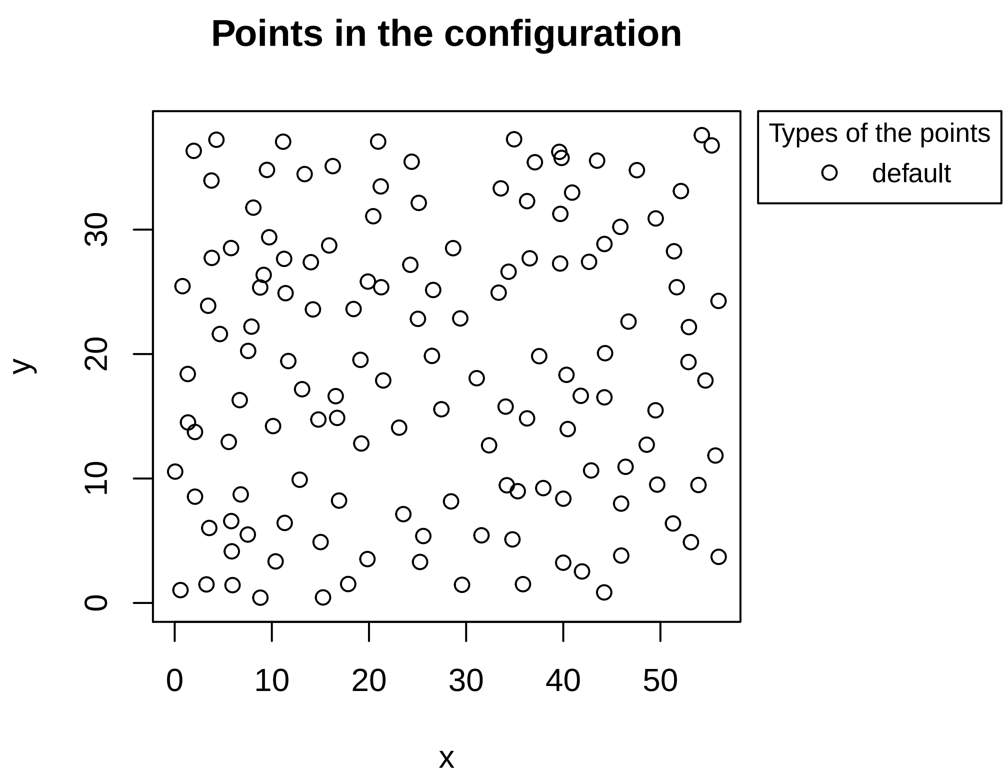


And as previously, we draw from the model.

```r
parameters <- get_parameters_from_fit(fit)
lambda <- parameters$lambda
alpha <- parameters$alpha
gamma <- parameters$gamma
```

```r
draw <- ppjsdm::rgibbs(window = window,
                       alpha = alpha,
                       lambda = lambda,
                       gamma = gamma,
                       model = model,
                       medium_range_model = medium_range_model,
                       short_range = fit$best_short,
                       medium_range = fit$best_medium,
                       long_range = fit$best_long,
                       types = levels(types(configuration)),
                       mark_range = c(min(get_marks(configuration)), max(get_marks(configuration))),
                       steps = 10000000)
print(draw)
#> An S3 object representing a configuration.
#>
#> Number of points: 137.

par(mar = c(5, 4, 4, 13) + 0.1)
plot(draw, window = window)
```

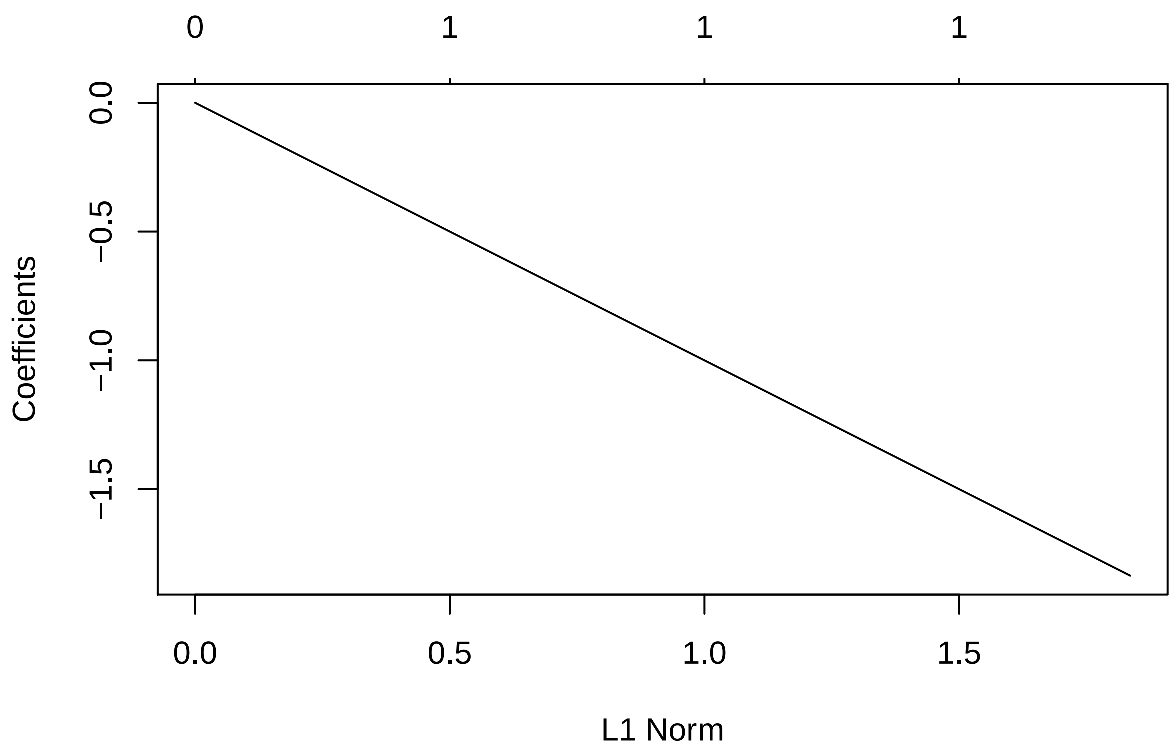


Comparison to classical Hard-core model.

```r
configuration <- Configuration(spruces$x, spruces$y)
model <- "Geyer"
medium_range_model <- "Geyer"
```

We call the fitting function on this unmarked point process.

```r
fit <- ppjsdm::gibbsm(configuration,
                      window = window,
                      model = model,
                      medium_range_model = medium_range_model,
                      short_range = short_range,
                      medium_range = medium_range,
                      long_range = long_range,
                      use_glmnet = TRUE,
                      use_aic = TRUE,
                      saturation = 1)
#> (Intercept) log_lambda1   alpha_1_1   gamma_1_1
#>    0.000000   -1.378804   -1.835898    0.000000
plot(fit$complete)
#> Warning in plotCoef(x$beta, lambda = x$lambda, df = x$df, dev = x$dev.ratio, : 1
#> or less nonzero coefficients; glmnet plot is not meaningful
```

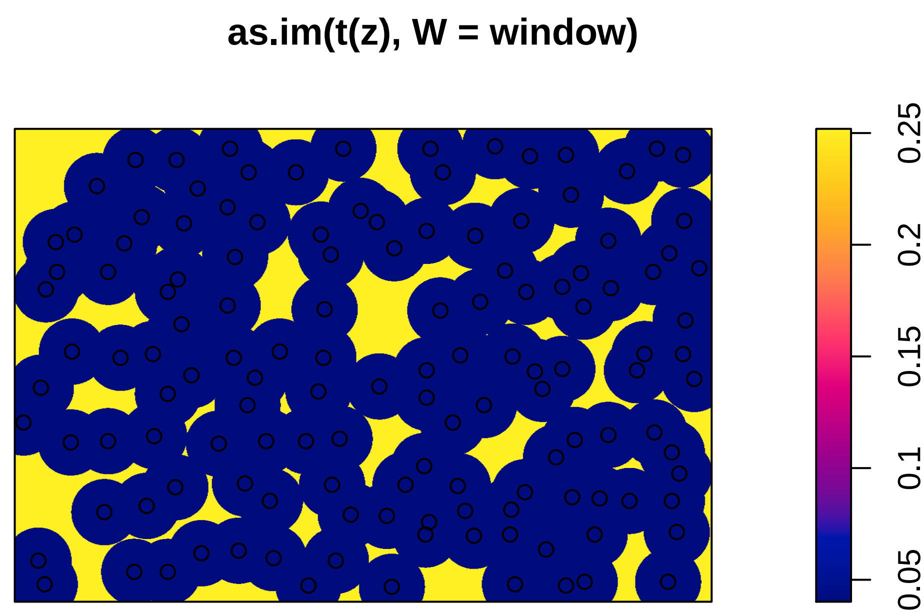



```r
print(fit$coefficients)
#> (Intercept) log_lambda1   alpha_1_1   gamma_1_1
#>    0.000000   -1.378804   -1.835898    0.000000
print(fit$best_short)
#>         [,1]
```

```r
#> [1,] 2.66237
print(fit$best_medium)
#>          [,1]
#> [1,] 12.44643
print(fit$best_long)
#>          [,1]
#> [1,] 22.93299
print(fit$aic)
#> [1] -309.05
print(fit$bic)
#> [1] -304.5487
```

We may then plot the corresponding Papangelou conditional intensity.

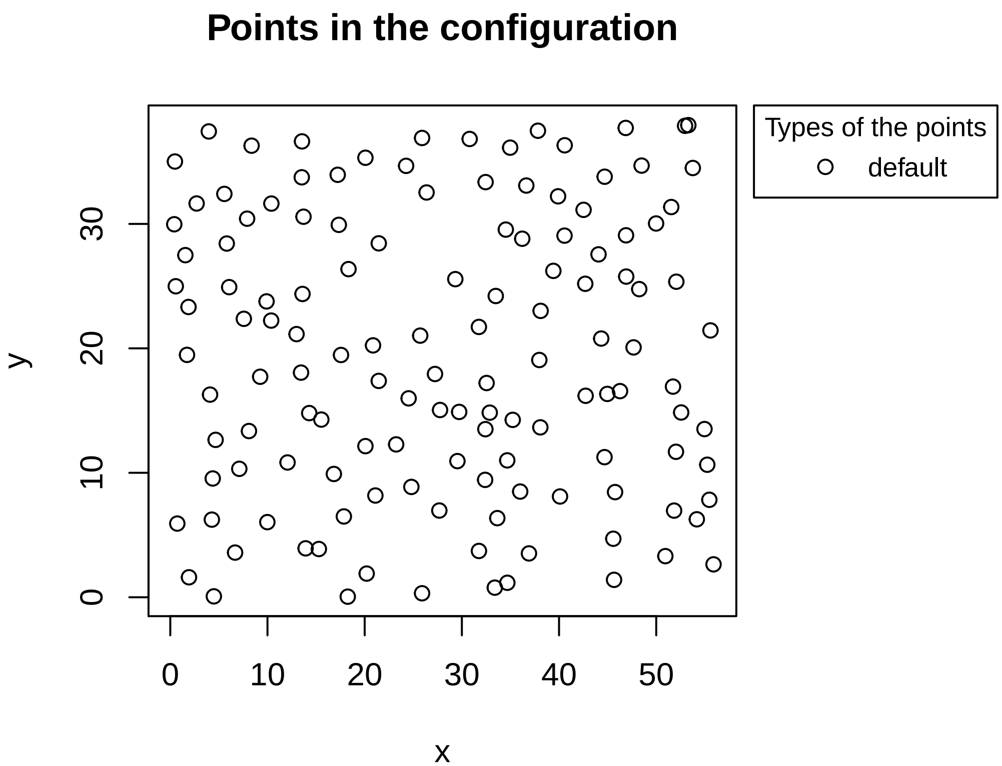
```r
parameters <- get_parameters_from_fit(fit)
lambda <- parameters$lambda
alpha <- parameters$alpha
gamma <- parameters$gamma
plot_papangelou(window = window,
                configuration = configuration,
                type = 1,
                mark = mean(get_marks(configuration)),
                model = model,
                medium_range_model = medium_range_model,
                alpha = alpha,
                lambda = lambda,
                beta = matrix(0, 1, 0),
                gamma = gamma,
                covariates = list(),
                short_range = fit$best_short,
                medium_range = fit$best_medium,
                long_range = fit$best_long,
                saturation = 1)
```



And as previously, we draw from the model.

```
parameters <- get_parameters_from_fit(fit)
lambda <- parameters$lambda
alpha <- parameters$alpha
gamma <- parameters$gamma
draw <- ppjsdm::rgibbs(window = window,
                       alpha = alpha,
                       lambda = lambda,
                       gamma = gamma,
                       model = model,
                       medium_range_model = medium_range_model,
                       short_range = fit$best_short,
                       medium_range = fit$best_medium,
                       long_range = fit$best_long,
                       types = levels(types(configuration)),
                       mark_range = c(min(get_marks(configuration)), max(get_marks(configuration))),
                       steps = 10000000)
print(draw)
#> An S3 object representing a configuration.
#>
#> Number of points: 132.

par(mar = c(5, 4, 4, 13) + 0.1)
plot(draw, window = window)
```


Points in the configuration
Types of the points
default
y
x
0
10
20
30
40
50
0
10
20
30