

A vignette for the INA R package: Smaller functions that create system components for the INAscene function

K. A. Garrett

Plant Pathology Department, Food Systems Institute, Emerging Pathogens Institute
University of Florida, Gainesville, USA

The INA package (with test version package named 'INApreliminary') implements impact network analysis (INA), an analysis of the effects of management technologies on regional outcomes such as the establishment of an invasive species or pathogen, or the establishment of a new crop variety or an endangered species, through linked socioeconomic networks and biophysical networks.

This vignette goes through the mechanics of the smaller functions called by one of the main INA functions, INAscene, to illustrate how these components work together. Some of these functions are always called by INAscene, while others are optional.

Note that package devtools needs to be installed to use the command to access the INApreliminary package from GitHub.

```
# if package devtools is not already installed, it will need to be installed  
# install.packages("devtools")  
  
library(devtools)
```

```
## Loading required package: usethis
```

```
devtools::install_github("GarrettLab/INApreliminary")
```

```
## Skipping install of 'INApreliminary' from a github remote, the SHA1 (8cff55fa) has not changed since  
##   Use `force = TRUE` to force installation
```

```
library(INApreliminary)
```

This vignette is designed to show the components of an impact network model. Once an impact network model is constructed, it can be used in impact network analysis to evaluate likely system outcomes for selected scenarios.

A simple form of impact network analysis addresses the impact of a single management technology, which modifies the probability of establishment of an invasive at a node in the biophysical network describing invasive species spread.

Note that many of the first functions described below are used to set up the impact network model. Some parts of this model might come from observations, such as the effect size of a management technology or a map of where land nodes in the biophysical network are located. When such data are not available, the INAscene function can generate examples of these model components for analysis, including a range of potential values of parameters in uncertainty quantification.

The following code illustrates functions in the INA package that are called by the function INAscene. These functions can be used separately, also, and are illustrated separately here to show how they work. The section numbering here corresponds to section numbers in the INA User Guide, available through a link at www.garrettlab.com/ina.

1. Geographic locations of biophysical nodes

An observed set of geographic coordinates for the biophysical nodes can be entered in INAscene, or the geographic locations can be generated for a scenario analysis in INAscene when INAscene calls the function `genlocs`.

Function `genlocs`

The coordinates for a specified area can be generated using the function `genlocs`. This function currently only generates random locations, but will be expanded to offer more options. `xrange4` gives the x-axis range and `yrange4` gives the y-axis range. `numnodes4` gives the number of locations and `randgeo4=TRUE` indicates that the locations will be generated randomly within the defined range.

```
geolocs99 <-  
  genlocs(  
    xrange4 = c(0, 50),  
    yrange4 = c(0, 50),  
    numnodes4 = 100,  
    randgeo4 = TRUE  
  )  
geolocs99
```

```
##           xvec           yvec  
## [1,] 49.8667574  8.8362584  
## [2,]  9.1774883  2.9685178  
## [3,] 34.8366907  9.6925393  
## [4,]  9.7825837 42.9502065  
## [5,] 37.9635900 42.9553311  
## [6,] 10.6308453 24.4859553  
## [7,] 34.7663996 38.8963194  
## [8,] 38.9067763 11.7870411  
## [9,] 22.7848964 32.4402879  
## [10,] 34.0024086 28.9604759  
## [11,] 33.9428577  2.1088167  
## [12,]  4.2530106 31.9202774  
## [13,]  9.9880283 24.4541918  
## [14,] 33.2372078 12.4509631  
## [15,]  0.2270362 32.7237777  
## [16,] 36.7184065 17.9855681  
## [17,]  4.0794064 26.2903948  
## [18,]  2.2778867  0.1778813  
## [19,] 49.8705667  1.9060372  
## [20,] 49.2712327 26.5535558  
## [21,]  4.0034803 40.6578176  
## [22,] 14.0779191 33.0838221  
## [23,] 13.7563885 47.8594562  
## [24,] 37.3996748 27.5195400
```

```

## [25,] 47.8869500 46.0468078
## [26,] 28.1968887 47.0883035
## [27,] 2.3753049 27.4054351
## [28,] 14.1282170 35.1618449
## [29,] 13.4434338 0.7031436
## [30,] 24.4743746 39.4663026
## [31,] 28.6258550 29.2549301
## [32,] 17.6874102 29.0289054
## [33,] 1.8317413 31.9182560
## [34,] 28.6873797 35.9221203
## [35,] 28.8117969 46.1976393
## [36,] 33.5327559 10.6067609
## [37,] 45.7381475 4.9209733
## [38,] 41.2413419 40.7916430
## [39,] 27.4115993 18.4272386
## [40,] 10.8251619 27.4033018
## [41,] 13.5612026 44.5750046
## [42,] 25.1819175 6.8405860
## [43,] 18.6752126 24.7342168
## [44,] 27.1255104 45.5289219
## [45,] 31.2218036 37.5818661
## [46,] 24.3015040 23.8165513
## [47,] 43.6801949 45.5502733
## [48,] 1.0809916 5.4905680
## [49,] 32.0840260 29.3562898
## [50,] 45.2855914 13.5763776
## [51,] 4.6471012 32.1867639
## [52,] 44.9264632 48.2390544
## [53,] 1.3461026 39.4414771
## [54,] 9.5404056 30.7785827
## [55,] 12.9778344 36.5666274
## [56,] 35.4547496 30.1722149
## [57,] 36.4377216 11.7288965
## [58,] 37.5788526 34.8622920
## [59,] 43.0741785 31.9356521
## [60,] 5.2342276 14.3836098
## [61,] 3.4420447 31.6690105
## [62,] 11.5545158 10.2828346
## [63,] 26.9585393 28.3795811
## [64,] 31.6097567 30.9249292
## [65,] 5.9987885 49.0363084
## [66,] 7.1419795 16.8666924
## [67,] 4.8706043 1.3388338
## [68,] 49.1100956 13.6509810
## [69,] 9.9823780 34.9871285
## [70,] 13.2609660 35.5365623
## [71,] 5.3283389 4.6577445
## [72,] 26.9164501 47.6255069
## [73,] 9.6456452 48.3419466
## [74,] 27.9552580 21.5656093
## [75,] 22.5097681 47.4125309
## [76,] 37.9298326 48.2798763
## [77,] 45.4774168 22.7472766
## [78,] 21.8614873 30.8947930

```

```
## [79,] 42.0699263 24.2912778
## [80,] 40.0669161 26.5270584
## [81,]  0.8971600 11.9454157
## [82,] 10.7173005 38.1670787
## [83,] 13.3396122 10.4270687
## [84,] 41.9146794 15.0652545
## [85,] 12.0763645 23.0611410
## [86,]  8.0155734 24.9015559
## [87,]  4.2749157 10.9273827
## [88,]  3.8578284  6.4328557
## [89,] 24.0461914 31.6737311
## [90,] 26.3083185 18.8485992
## [91,] 21.6871063 41.7899212
## [92,]  4.6181826 18.6010330
## [93,] 23.3484929 17.7550405
## [94,]  1.3570625 31.8707777
## [95,] 38.1818303 26.9350982
## [96,] 26.5576946 18.4874151
## [97,] 12.8379844  7.5940924
## [98,] 10.7859882 11.7240261
## [99,] 18.0784206 36.4546424
## [100,] 21.4872679 41.6513121
```

2.1. Initial locations for information about management in the socioeconomic network & 2.2. Initial locations for the bioentity in the biophysical network

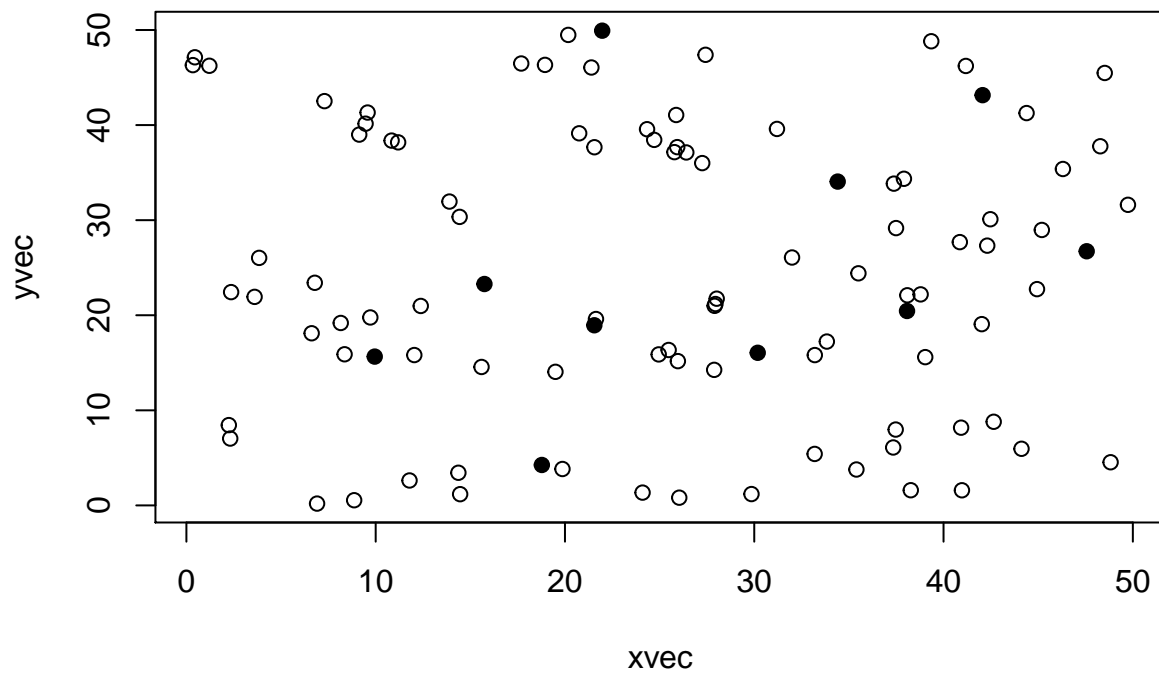
Observed initial locations for management technology information or for the bioentity (species, cultivar, subpopulation with specific traits, etc.) can be read into INAscene. Alternatively, the initial locations can be generated in INAscene when it calls the function `initvals`.

Function `initvals`

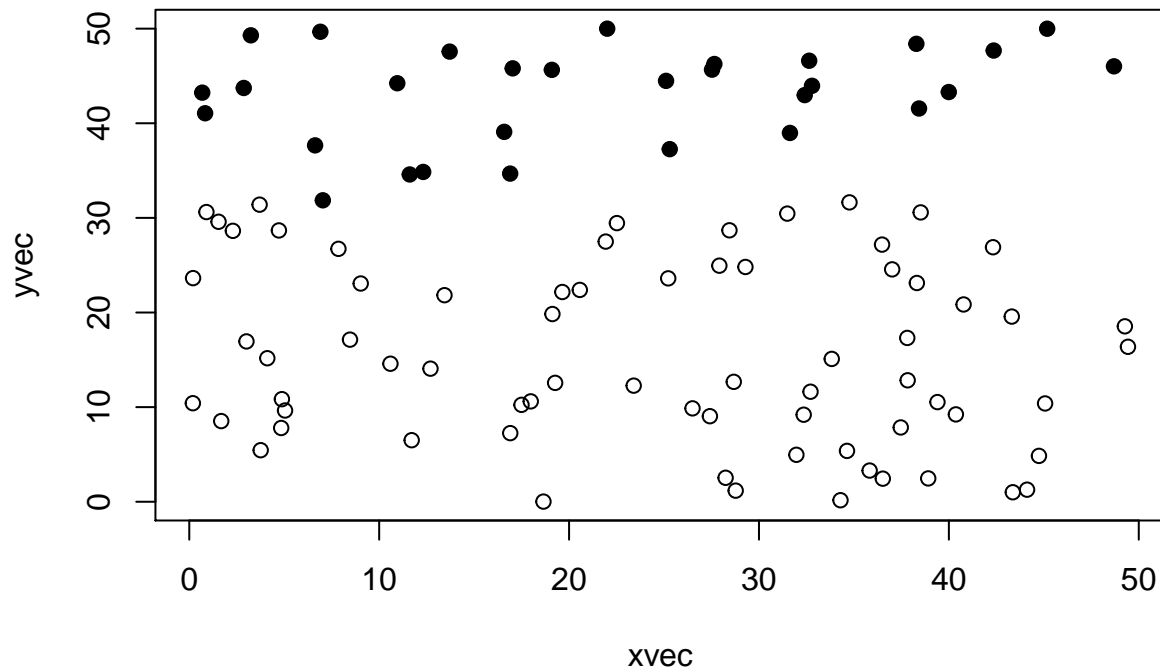
The function `initvals` can determine where information or a bioentity is initially located based on an indicated scenario. Try multiple runs of each example to see how the results may vary. Presence is indicated by a filled symbol, where some simulations may result in no presence by chance.

Note that in the first three examples, the initial node locations are generated using function `genlocs`, while in the last two examples the initial node locations are supplied directly.

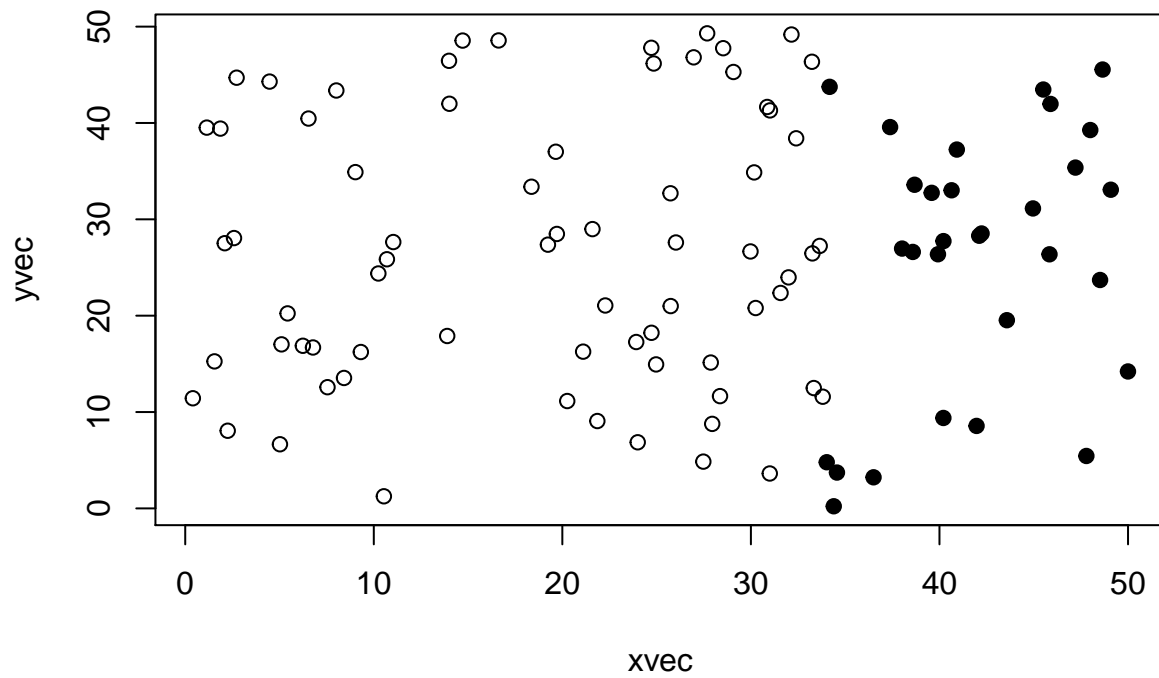
```
x2 <-
  initvals(
    j2 <-
      genlocs(
        xrange4 = c(0, 50),
        yrange4 = c(0, 50),
        numnodes4 = 100,
        randgeo4 = TRUE
      ),
    dist4 = 'random',
    init.p4 = 0.1,
    norp4 = 'prop',
    plotmp = T
  )
```



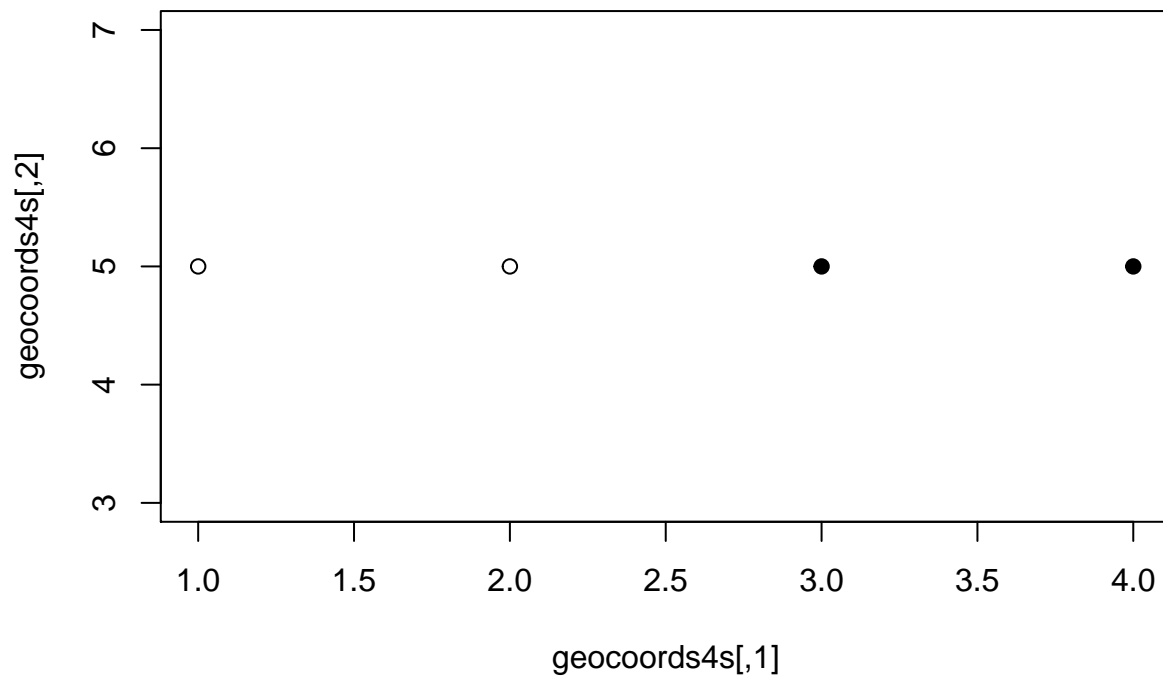
```
x3 <-
  initvals(
    j2 <-
      genlocs(
        xrange4 = c(0, 50),
        yrange4 = c(0, 50),
        numnodes4 = 100,
        randgeo4 = TRUE
      ),
    dist4 = 'upedge',
    init.p4 = 0.3,
    norp4 = 'prop',
    plotmp = T
  )
```



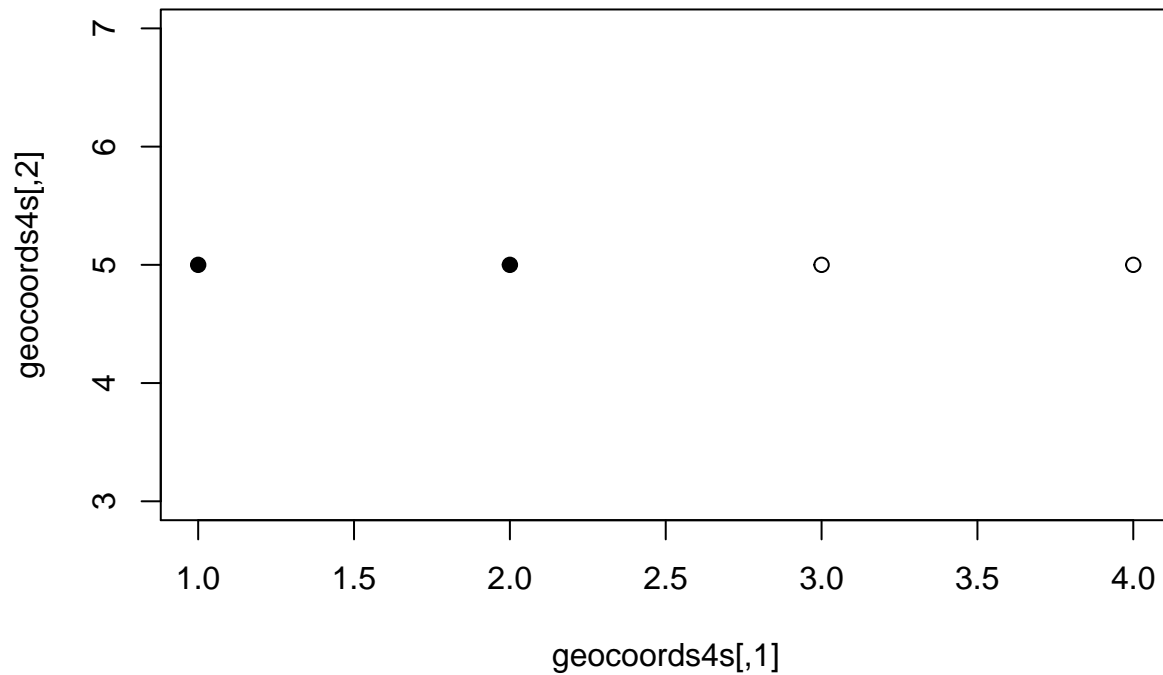
```
x4 <-
  initvals(
    j2 <-
      genlocs(
        xrange4 = c(0, 50),
        yrange4 = c(0, 50),
        numnodes4 = 100,
        randgeo4 = TRUE
      ),
    dist4 = 'rightedge',
    init.p4 = 0.3,
    norp4 = 'prop',
    plotmp = T
  )
```



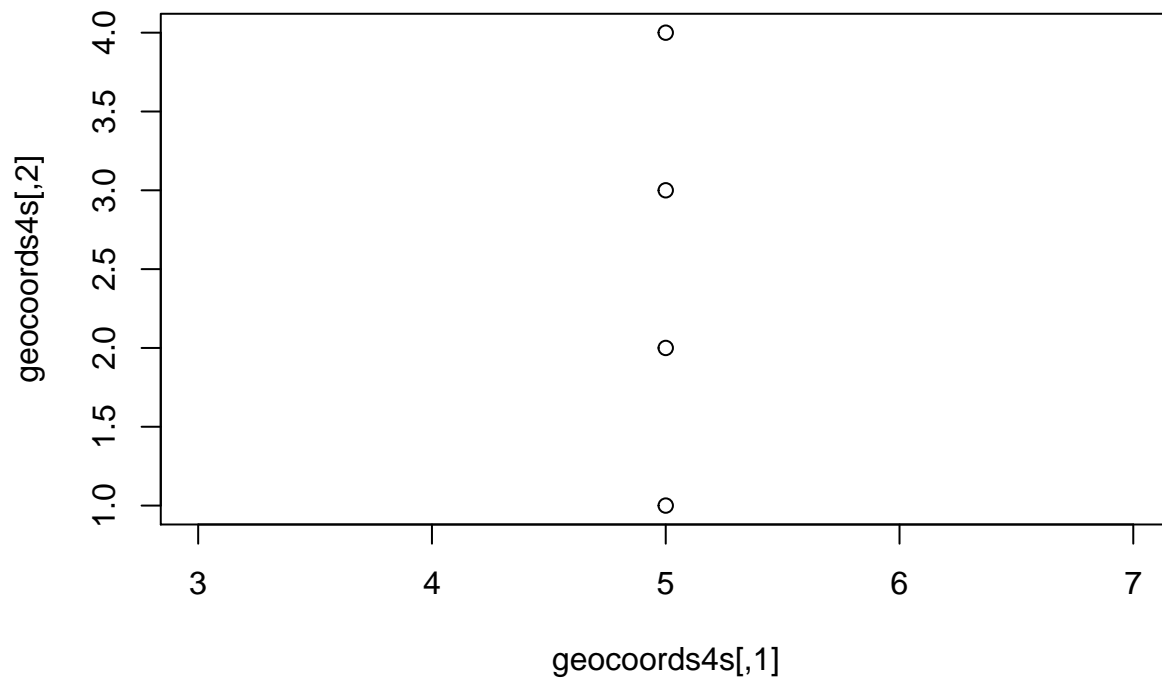
```
x5 <-
  initvals(
    geocoords4s = matrix(c(1, 5, 2, 5, 3, 5, 4, 5), byrow = T, ncol = 2),
    dist4 = 'rightedge',
    norp4 = 'num',
    init.n4 = 2,
    plotmp = T
  )
```



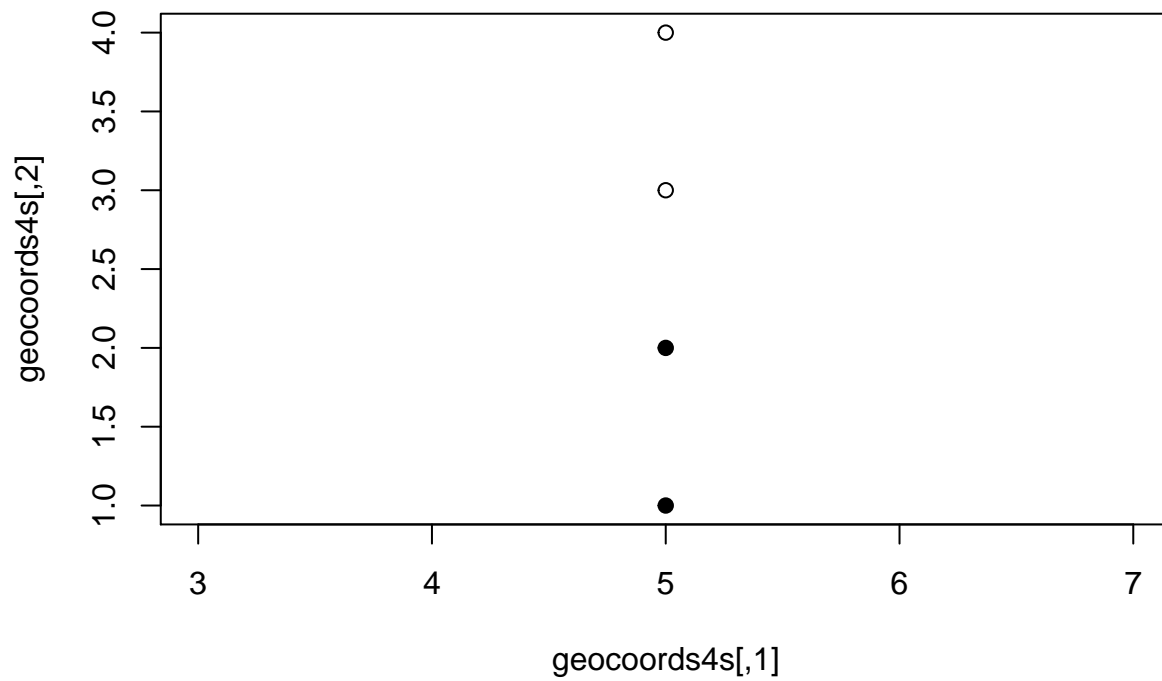
```
x5b <-
  initvals(
    geocoords4s = matrix(c(1, 5, 2, 5, 3, 5, 4, 5), byrow = T, ncol = 2),
    dist4 = 'upedge',
    norp4 = 'num',
    init.n4 = 2,
    plotmp = T
  )
```



```
x6 <-
  initvals(
    geocoords4s = matrix(c(5, 1, 5, 2, 5, 3, 5, 4), byrow = T, ncol = 2),
    dist4 = 'upedgeedge',
    norp4 = 'num',
    init.n4 = 2,
    plotmp = T
  )
```

```
x6b <-
  initvals(
    geocoords4s = matrix(c(5, 1, 5, 2, 5, 3, 5, 4), byrow = T, ncol = 2),
    dist4 = 'rightedge',
    norp4 = 'num',
    init.n4 = 2,
    plotmp = T
  )
```



3.1. Adjacency matrix and spread of communication about management in socioeconomic network & 3.2. Adjacency matrix and spread of bioentity dispersal in biophysical network

An adjacency matrix describing the socioeconomic network and the biophysical network can be read directly into function INAscene, or it can be generated in INAscene by having INAscene call the function genmovnet.

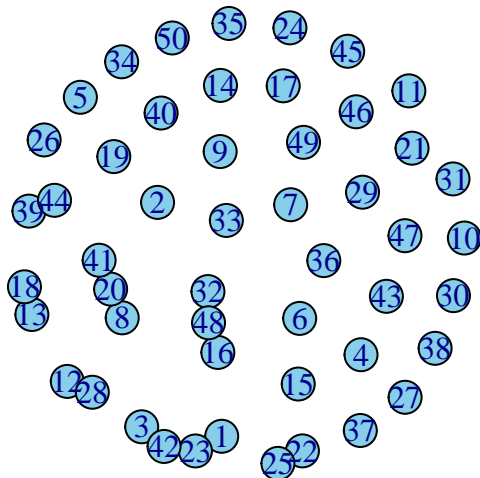
Then the function spreadstep simulates spread of information or of a bioentity across one time step.

Function genmovnet

The function genmovnet creates an adjacency matrix for movement. (It currently is set up to generate a random symmetric matrix, or for an inverse powerlaw function of distance. A negative exponential function of distance will also be added.) genmovnet uses the igraph package if the plotting option is selected.

Here's an example using genlocs to generate node locations, where the genmovnet function generates a random adjacency matrix (and provides a figure if iplot=T). In the first example there is a lower probability of existence of a link (amrandp4=0.01) than in the second example (amrandp4=0.02).

```
x1 <-  
  genmovnet(  
    j <-  
    genlocs(  
      xrange4 = c(0, 50),  
      yrange4 = c(0, 50),  
      numnodes4 = 50,  
      randgeo4 = TRUE  
    ),  
    amdist4 = 'random',  
    amrandp4 = 0.01,  
    iplot = T  
  )
```



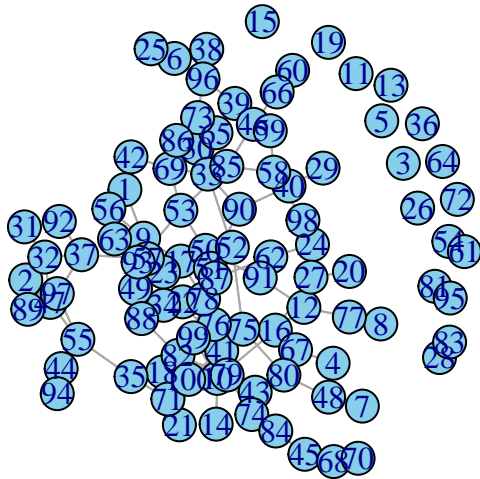
```
x2 <-  
  genmovnet(  
    j <-  
    genlocs(  

```

```

xrange4 = c(0, 50),
yrange4 = c(0, 50),
numnodes4 = 100,
randgeo4 = TRUE
),
amdist4 = 'random',
amrandp4 = 0.02,
ipplot = T
)

```



Illustrating how the inverse inverse power law function works, for $a \cdot d^{-b}$

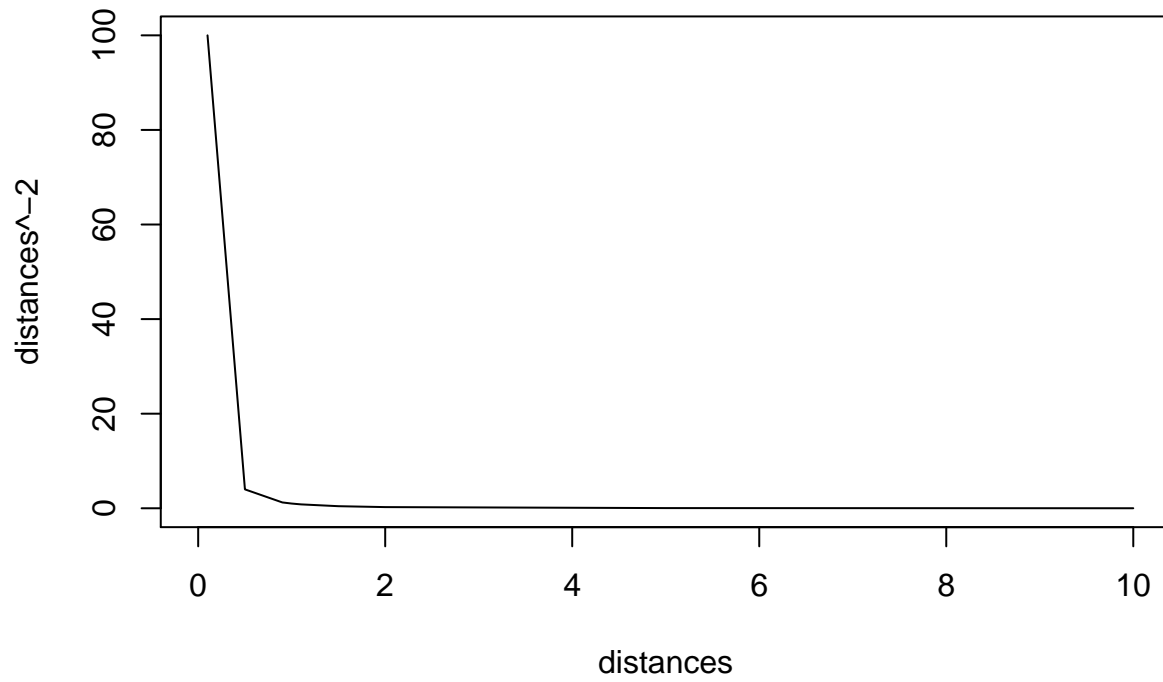
The inverse power law function is often used in models of the likelihood of movement as a function of distance. Here's a reminder of the effects of changing the b parameter in an inverse power law function:

```

distances <- c(0,0.1,0.5,0.9,1,1.1,1.5,2,5,10)
plot(distances,distances^-2,main='b=2', type='l')

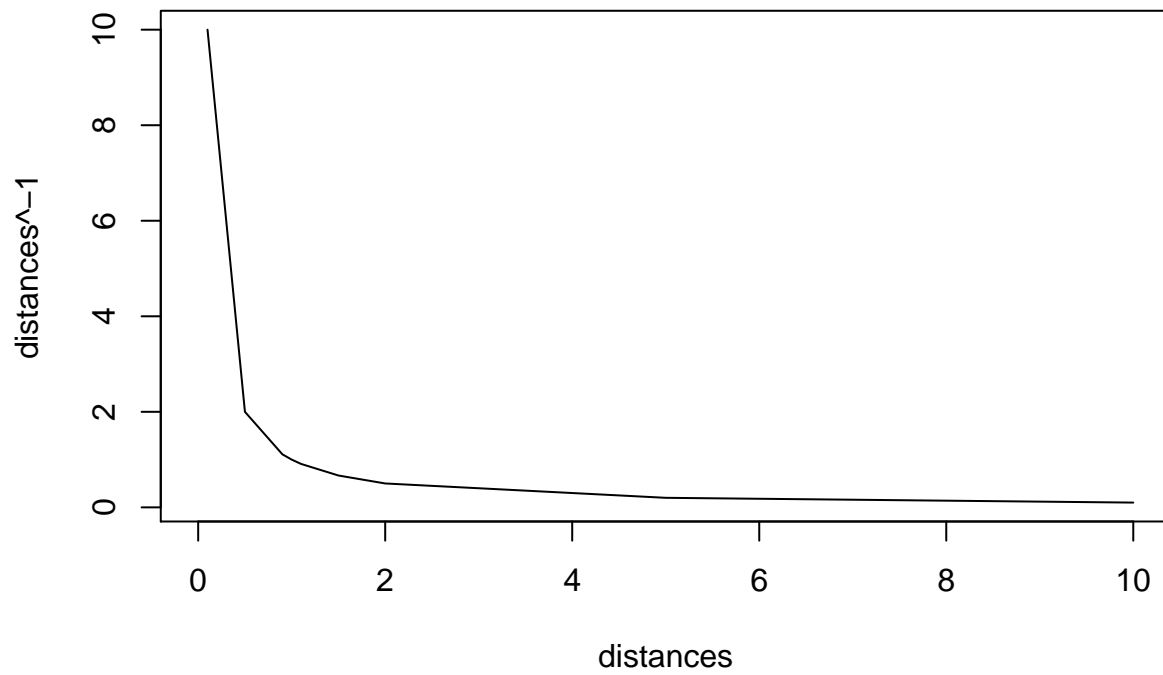
```

b=2



```
plot(distances,distances-1,main='b=1', type='l')
```

b=1



```
plot(distances,distances-0.5,main='b=0.05', type='l')
```

b=0.05



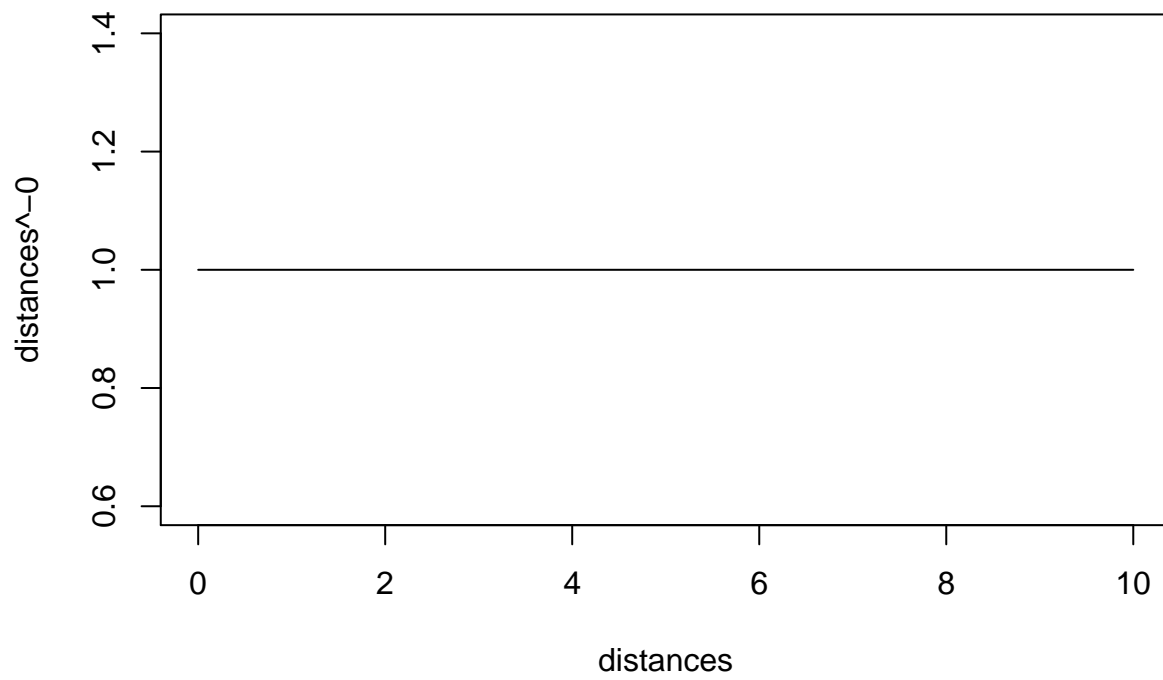
```
plot(distances,distances-0.1,main='b=0.01', type='l')
```

b=0.01



```
plot(distances,distances-0,main='b=0', type='l') # no effect of distance
```

b=0



```
plot(distances,distances^1,main='b=-1', type='l') # not usually relevant
```

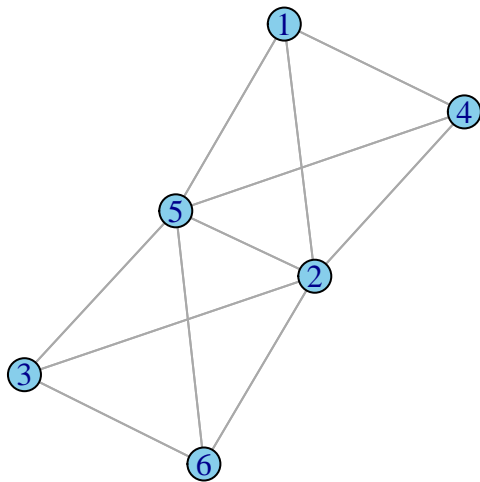
b=-1



Examples of networks created using the inverse power law function

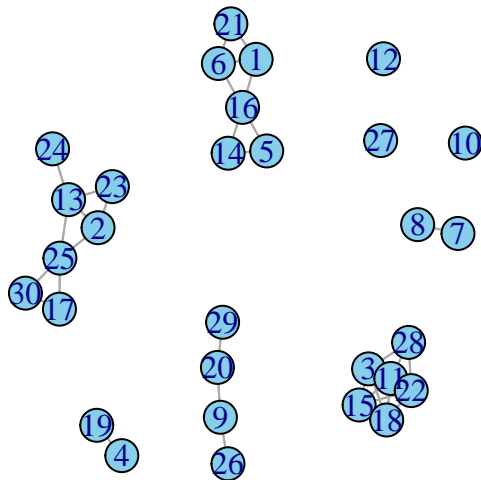
```
x7 <-
  genmovnet(
    geocoords4n = matrix(
      c(1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3),
      ncol = 2,
      byrow = T
    ),
    amdistr4 = 'powerlaw',
    ampla4 = 1,
    amplb4 = 2,
    iplot = T
  )
```

Links with $p > 0.3$



```
x8 <-
  genmovnet(
    j <-
      genlocs(
        numnodes4 = 30,
        xrange4 = c(0, 10),
        yrange4 = c(0, 10)
      ),
    amdistr4 = 'powerlaw',
    ampla4 = 1,
    amplb4 = 2,
    iplot = T
  )
```

Links with $p > 0.3$



Function spreadstep

The function `spreadstep`, called by the function `INAscene`, generates a vector of status for each node after one spread step, multiplying the initial state vector by the adjacency matrix. Examples of status include

Socioeconomic network:

Informed about management or not

Using management or not

Biophysical network:

Bioentity established or not

The function assumes that the input adjacency matrix is composed of 0s and 1s, and that rows represent source nodes and columns represent sink nodes.

Because bioentity establishment and technology adoption are handled separately in `INAscene`, this function maintains the information or bioentity at a location even if the diagonal of the adjacency matrix provided is not composed of 1s.

```
tmat <- matrix(byrow=T, ncol=6,
               c(0,1,0,0,0,0, 0,0,1,0,0,0, 0,0,0,1,0,0, 0,0,0,0,1,0,
                 0,0,0,0,0,1, 1,0,0,0,0,0))
tmat
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0    1    0    0    0    0
## [2,]    0    0    1    0    0    0
## [3,]    0    0    0    1    0    0
## [4,]    0    0    0    0    1    0
## [5,]    0    0    0    0    0    1
## [6,]    1    0    0    0    0    0
```



```

startvec <- c(1,1,1,0,0,0)
endvec <- spreadstep(amat=tmat, vect1=startvec)
cbind(startvec,endvec)

```

```

##      startvec endvec
## [1,]        1      1
## [2,]        1      1
## [3,]        1      1
## [4,]        0      1
## [5,]        0      0
## [6,]        0      0

```

```

startvec <- c(0,0,0,1,1,0)
endvec <- spreadstep(amat=tmat, vect1=startvec)
cbind(startvec,endvec)

```

```

##      startvec endvec
## [1,]        0      0
## [2,]        0      0
## [3,]        0      0
## [4,]        1      1
## [5,]        1      1
## [6,]        0      1

```

Considering an example with 100 nodes:

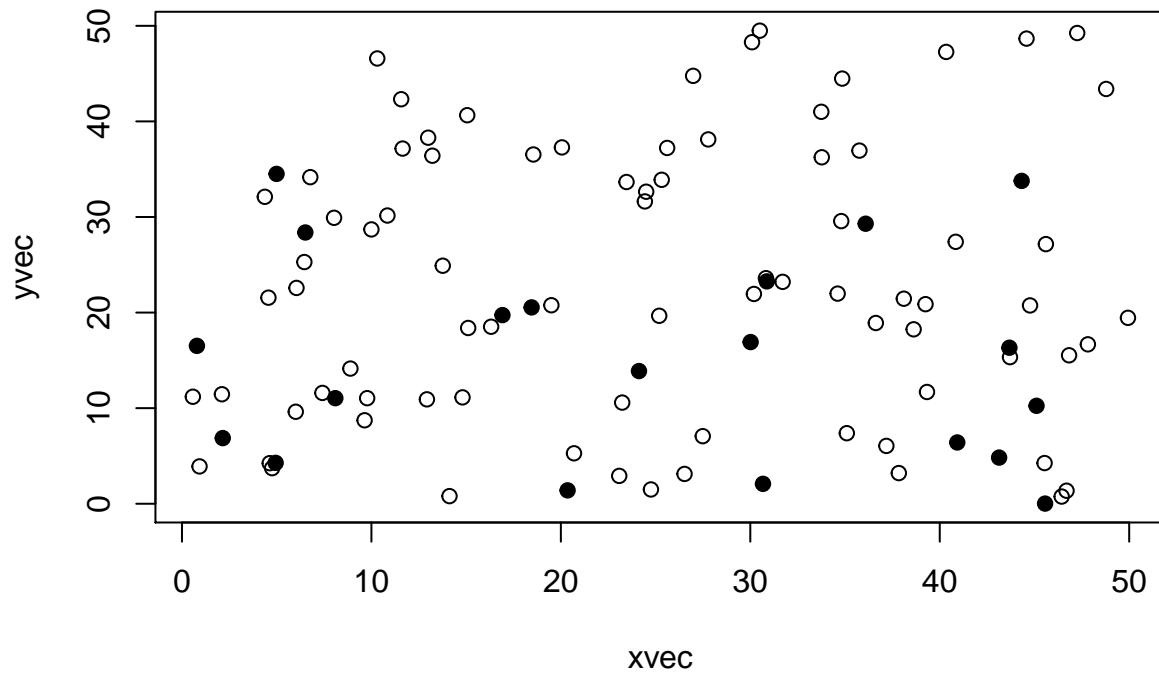
First the starting locations for information are illustrated, then the starting locations for a bioentity, then the corresponding adjacency matrices describing movement of the info and bioentity.

```

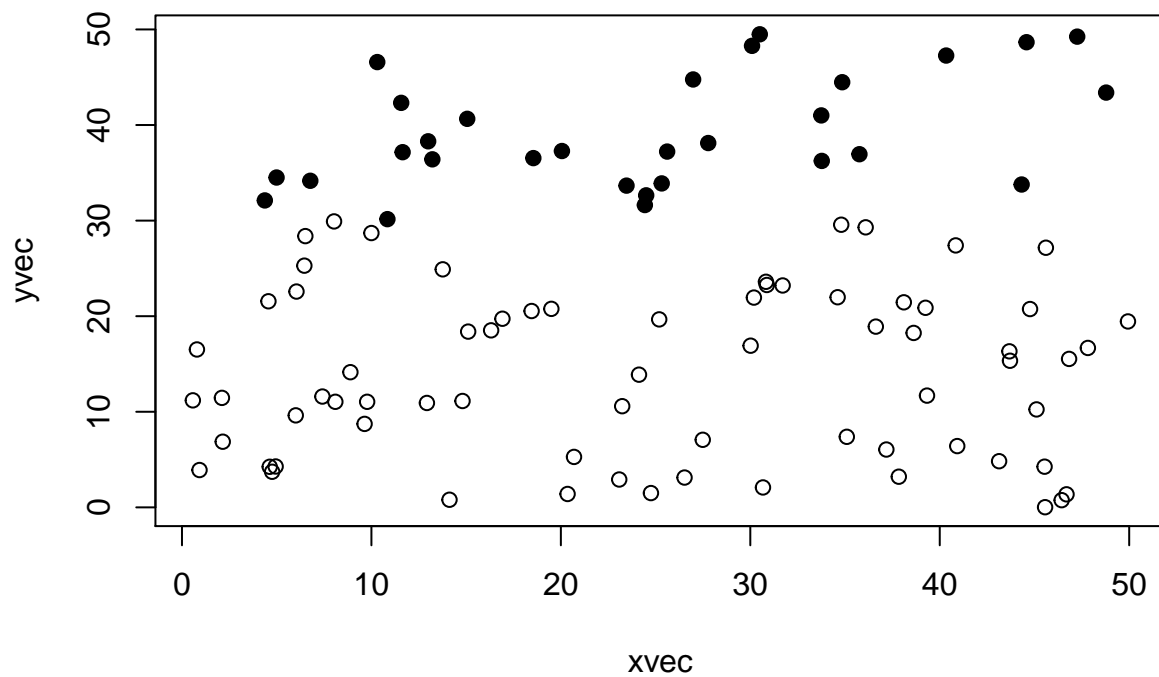
geolocs100 <-
  genlocs(
    xrange4 = c(0, 50),
    yrange4 = c(0, 50),
    numnodes4 = 100,
    randgeo4 = TRUE
  )

init100.info <-
  initvals(
    geocoords4s = geolocs100,
    dist4 = 'random',
    init.p4 = 0.2,
    norp4 = 'prop',
    plotmp = T
  )

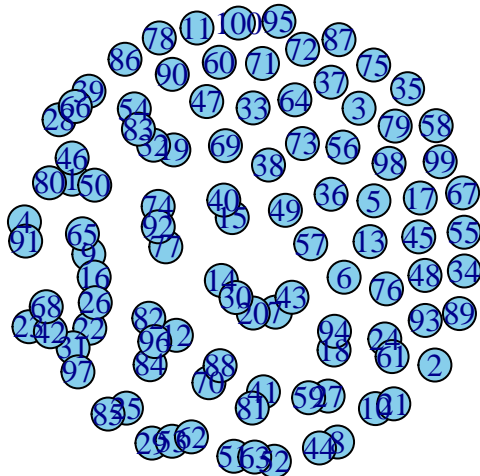
```



```
init100.bio <-
initvals(
geocoords4s = geolocs100,
dist4 = 'upedge',
init.p4 = 0.3,
norp4 = 'prop',
plotmp = T
)
```

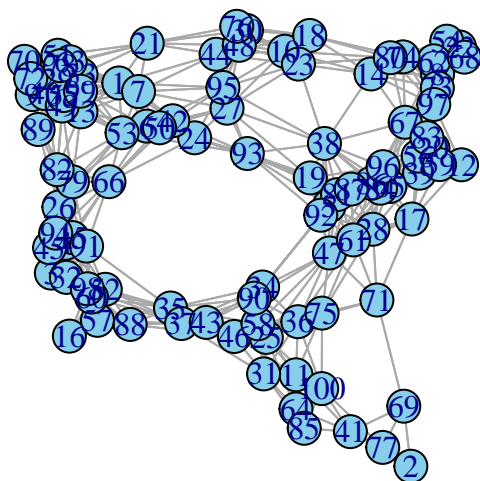


```
adjmat100.info <-
  genmovnet(
    geocoords4n = geolocs100,
    amdists4 = 'random',
    amrandp4 = 0.01,
    iplot = T
  )
```



```
adjmat100.bio <-
  genmovnet(
    geocoords4n = geolocs100,
    amdists4 = 'powerlaw',
    ampla4 = 1,
    amplb4 = 0.5,
    iplot = T
  )
```

Links with $p > 0.3$



4.1. Adoption of management

The probability of management technology adoption at each location in the socioeconomic network can be read into the INAscene function, or it can be generated by having INAscene call the function `makedec`

Function `makedec`

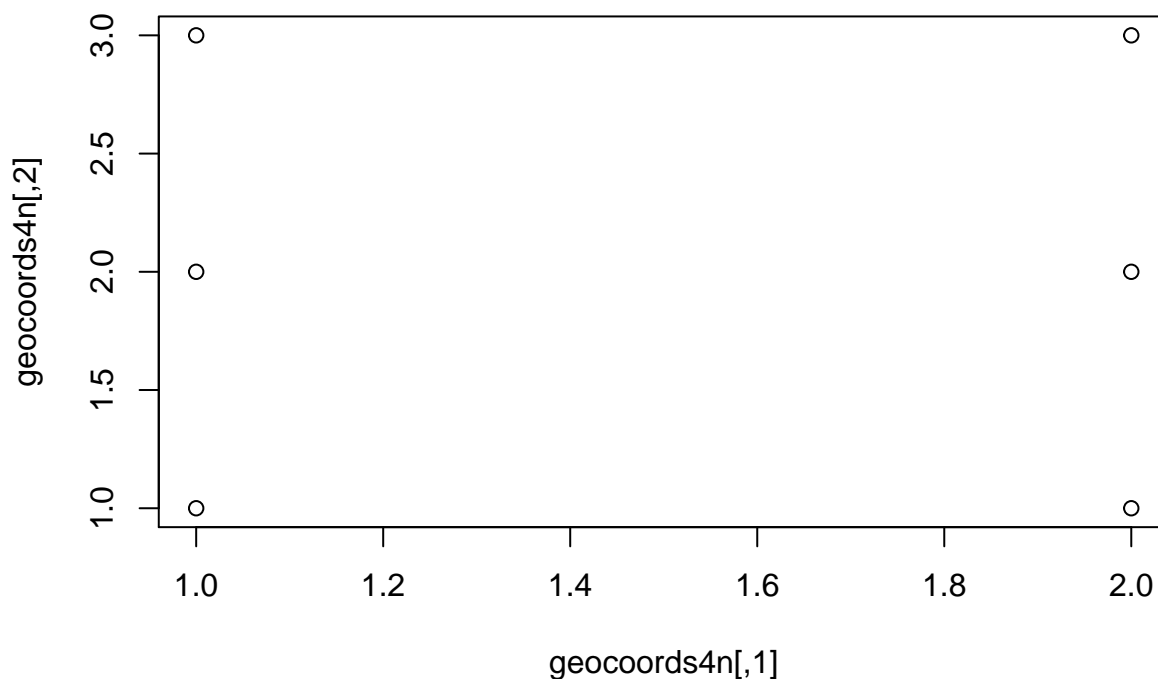
The function `makedec` generates decisions about management at each location. It takes as input the status of communication at each location, in terms of whether the manager of that location is informed about the management option.

It also reads in or generates the probability of management adoption at each location (the probability of adoption given communication has informed the manager at that location). Then it combines the two components to evaluate whether management will be implemented at each location.

Here is an example for 6 decision makers. The probability of technology adoption has mean `probadoptmean4` and standard deviation `probadoptsd4`, and the probability at each location is generated based on this in a truncated normal distribution.

```
x5 <-  
  makedec(  
    comvec = c(1, 1, 1, 0, 0, 0),  
    geocoords4n = matrix(  
      c(1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3),  
      byrow = T,  
      ncol = 2  
    ),  
    readprobadoptvec4 = F,  
    probadoptmean4 = 0.1,  
    probadoptsd4 = 0.1,  
    plotmp = T  
  )
```

Decision to adopt, shaded = yes



For the example of 100 nodes:

```
dec100.t2 <-
  makedec(
    comvec = init100.info,
    geocoords4n = geolocs100,
    probadoptmean4 = 0.8,
    probadoptsd4 = 0.1,
    readprobadoptvec4 = F
  )
dec100.t2
```

```
## [1] FALSE FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [37] FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [49] FALSE TRUE FALSE FALSE FALSE TRUE TRUE TRUE FALSE FALSE FALSE FALSE
## [61] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE
## [73] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [85] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE
## [97] TRUE FALSE FALSE FALSE
```

4.2. Establishment of bioentity and the effect of management

Similarly, the probability of establishment at a location can be read in to INAscene as a vector, or generated in INAscene calling the function `estab`.

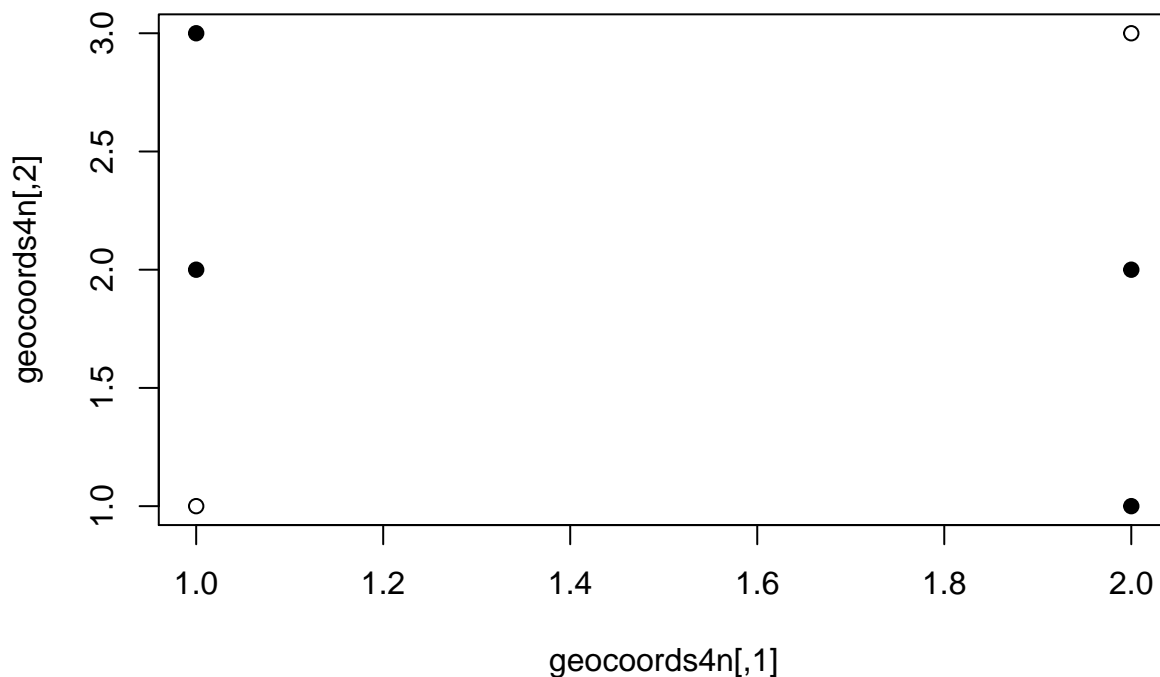
Function estab

The function `estab` determines whether a bioentity becomes (or remains) established at a location, given that it arrived there. It reads in vectors indicating for each location (a) whether the species has arrived there, and (b) whether management is being applied there. If management is being applied, the probability of establishment is changed. The probability is generally reduced for invasive species management technologies (`maneffdir4='decrease_estab'`), generally increased for management of endangered species or new crop varieties (`maneffdir4='increase_estab'`).

Examples of management that reduces the probability of establishment

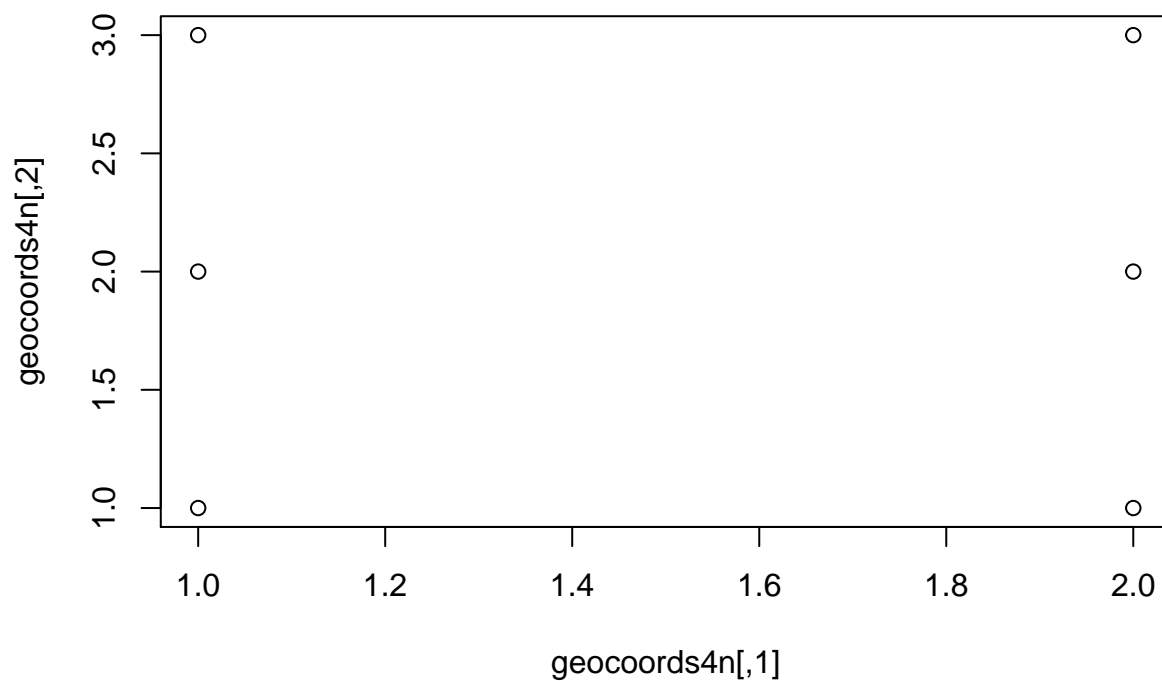
```
x6decrease <-  
  estab(  
    decvec = c(1, 1, 1, 0, 0, 0),  
    dispvec = c(1, 1, 1, 1, 1, 0),  
    geocoords4n = matrix(  
      c(1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3),  
      byrow = T,  
      ncol = 2  
    ),  
    readprobestabvec4 = F,  
    probestabmean4 = 0.5,  
    probestabsd4 = 0.1,  
    maneffdir4 = 'decrease_estab',  
    maneffmean4n = 0.1,  
    maneffsd4n = 0.1,  
    plotmp = T  
  )
```

Establishment, shaded = yes



```
x7decrease <-
  estab(
    decvec = c(1, 1, 1, 0, 0, 0),
    dispvec = c(1, 1, 1, 1, 1, 0),
    geocoords4n = matrix(
      c(1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3),
      byrow = T,
      ncol = 2
    ),
    readprobestabvec4 = F,
    probestabmean4 = 0.5,
    probestabsd4 = 0.1,
    maneffdir4 = 'decrease_estab',
    maneffmean4n = 0.9,
    maneffsd4n = 0.1,
    plotmp = T
  )
```

Establishment, shaded = yes



```
x8decrease <-
  estab(
    decvec = c(1, 1, 1, 0, 0, 0),
    dispvec = c(1, 1, 1, 1, 1, 0),
    geocoords4n = matrix(
      c(1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3),
      byrow = T,
      ncol = 2
    ),
    readprobestabvec4 = T,
    probestabvec4 = c(0.9, 0.1, 0.1, 0.1, 0.1, 0.9),
```

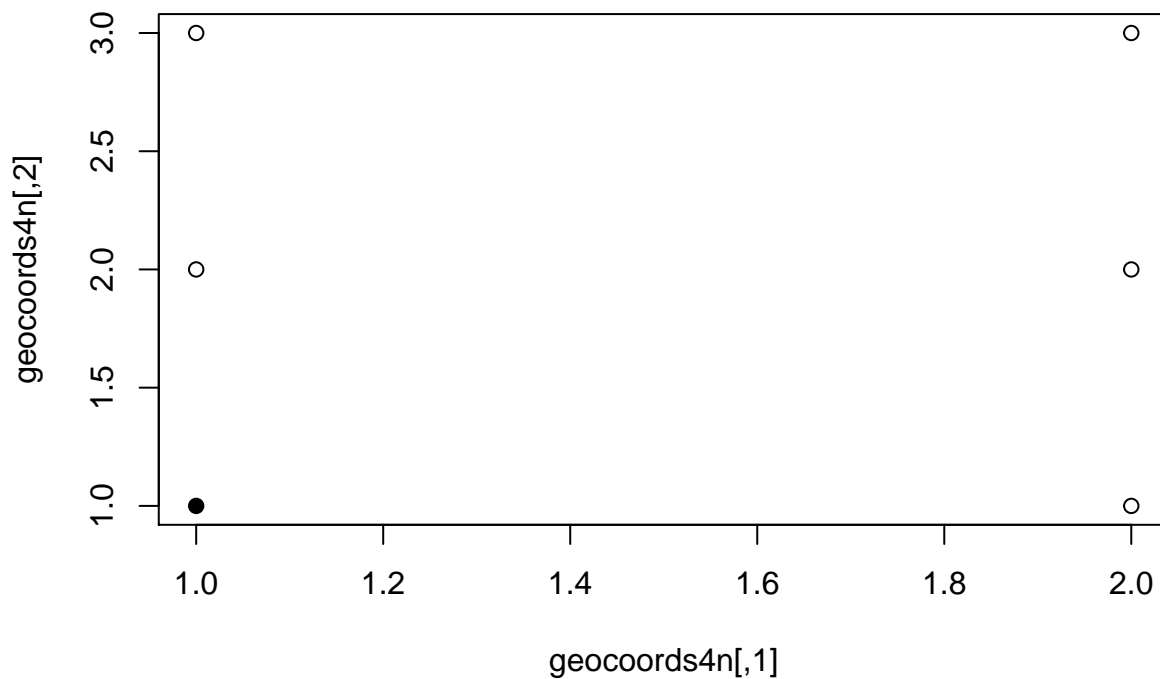
```

maneffdir4 = 'decrease_estab',
maneffmean4n = 0.9,
maneffsd4n = 0.1,
plotmp = T
)

x9decrease <-
  estab(
    decvec = c(1, 1, 1, 0, 0, 0),
    dispvec = c(1, 1, 1, 1, 1, 0),
    geocoords4n = matrix(
      c(1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3),
      byrow = T,
      ncol = 2
    ),
    readprobestabvec4 = T,
    probestabvec4 = c(0.9, 0.1, 0.1, 0.1, 0.1, 0.9),
    maneffdir4 = 'decrease_estab',
    maneffmean4n = 0.1,
    maneffsd4n = 0.1,
    plotmp = T
  )

```

Establishment, shaded = yes



*# Now some examples of management to increase the probability of establishment
(in practice, reduce the probability of NOT establishing)*

```

x6increase <-
  estab(
    decvec = c(1, 1, 1, 0, 0, 0),

```

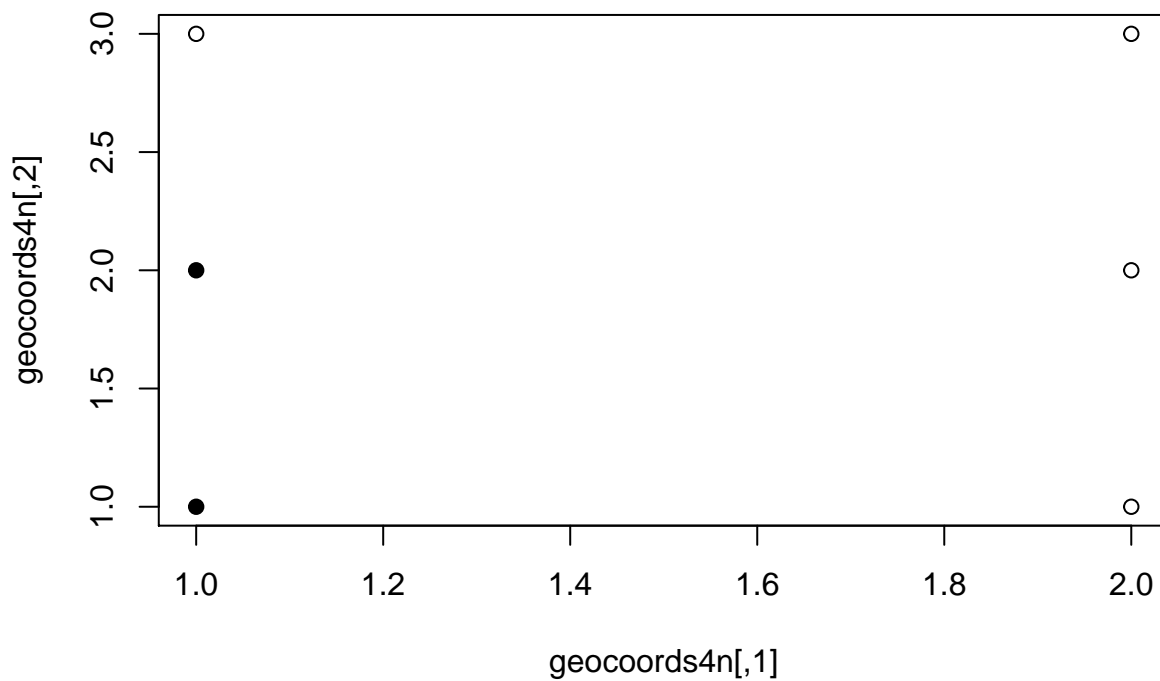


```

dispvec = c(1, 1, 1, 1, 1, 0),
geocoords4n = matrix(
c(1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3),
byrow = T,
ncol = 2
),
readprobestabvec4 = F,
probestabmean4 = 0.5,
probestabsd4 = 0.1,
maneffdir4 = 'increase_estab',
maneffmean4n = 0.1,
maneffsd4n = 0.1,
plotmp = T
)

```

Establishment, shaded = yes



```

x7increase <-
  estab(
    decvec = c(1, 1, 1, 0, 0, 0),
    dispvec = c(1, 1, 1, 1, 1, 0),
    geocoords4n = matrix(
      c(1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3),
      byrow = T,
      ncol = 2
    ),
    readprobestabvec4 = F,
    probestabmean4 = 0.5,
    probestabsd4 = 0.1,
    maneffdir4 = 'increase_estab',
    maneffmean4n = 0.9,
  )

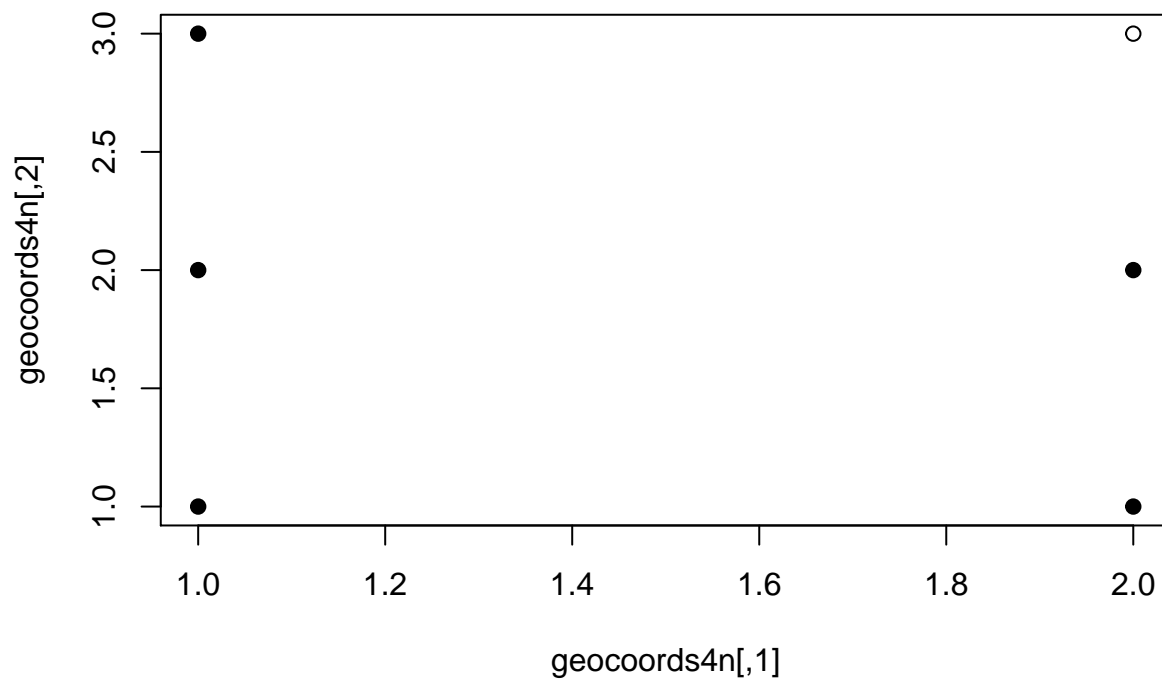
```

```

maneffsd4n = 0.1,
plotmp = T
)

```

Establishment, shaded = yes

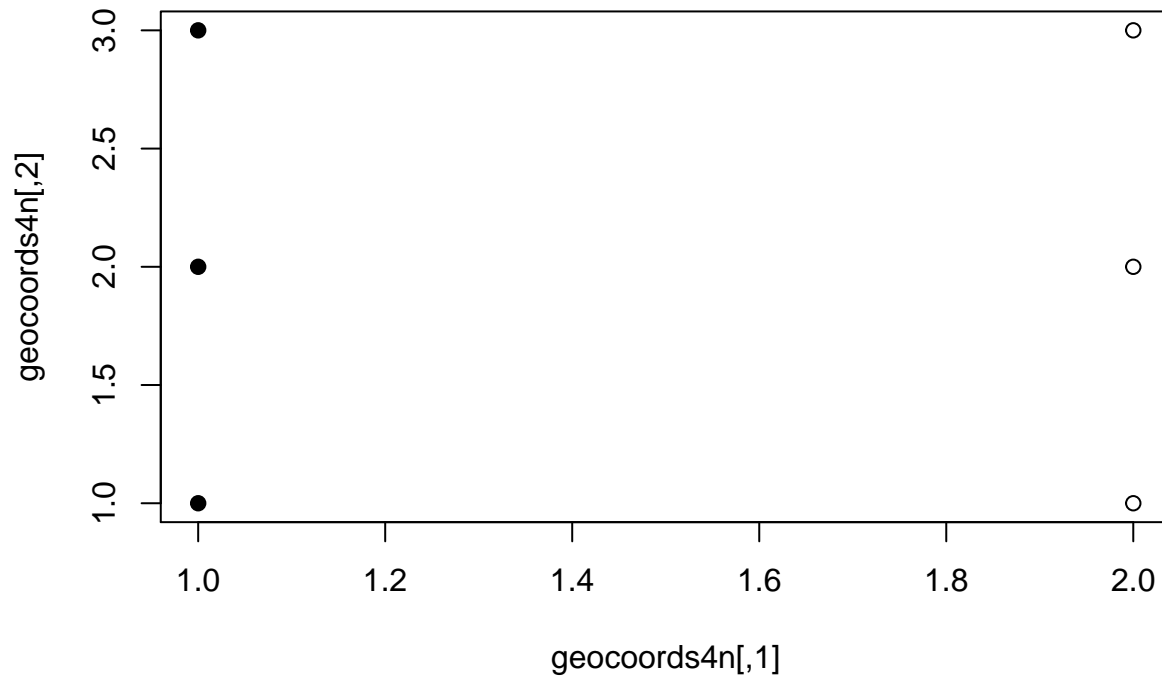


```

x8increase <-
  estab(
    decvec = c(1, 1, 1, 0, 0, 0),
    dispvec = c(1, 1, 1, 1, 1, 0),
    geocoords4n = matrix(
      c(1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3),
      byrow = T,
      ncol = 2
    ),
    readprobestabvec4 = T,
    probestabvec4 = c(0.9, 0.1, 0.1, 0.1, 0.1, 0.9),
    maneffdir4 = 'increase_estab',
    maneffmean4n = 0.9,
    maneffsd4n = 0.1,
    plotmp = T
  )

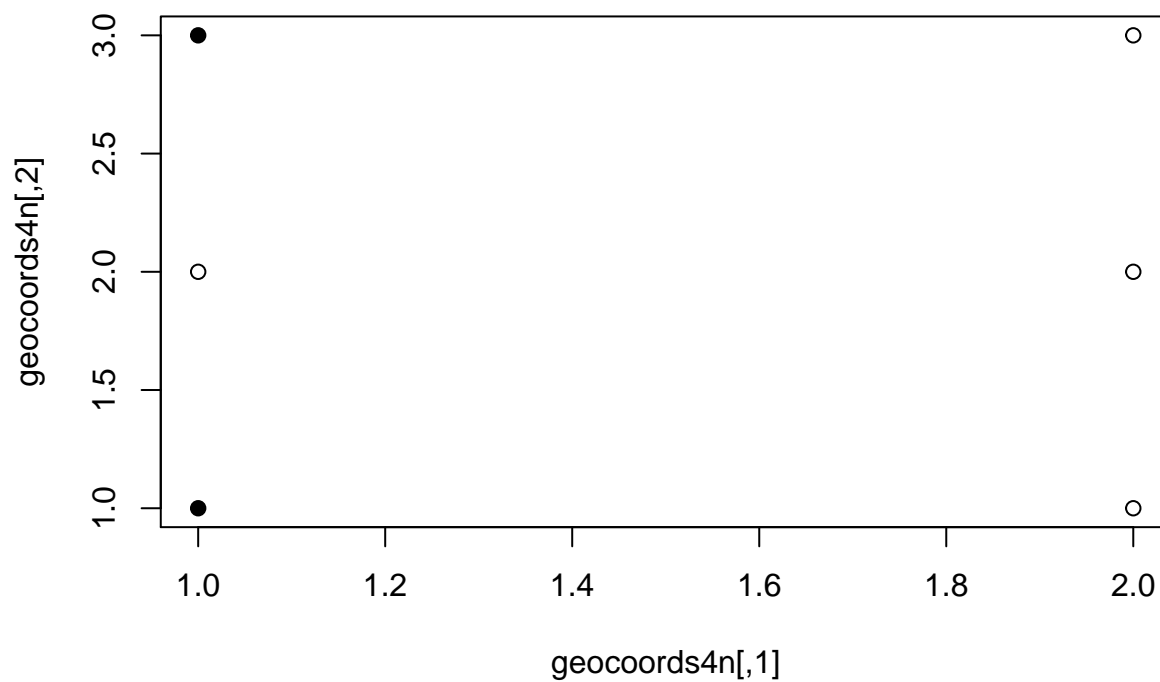
```

Establishment, shaded = yes



```
x9increase <-  
  estab(  
    decvec = c(1, 1, 1, 0, 0, 0),  
    dispvec = c(1, 1, 1, 1, 1, 0),  
    geocoords4n = matrix(  
      c(1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3),  
      byrow = T,  
      ncol = 2  
    ),  
    readprobestabvec4 = T,  
    probestabvec4 = c(0.9, 0.1, 0.1, 0.1, 0.1, 0.9),  
    maneffdir4 = 'increase_estab',  
    maneffmean4n = 0.1,  
    maneffsd4n = 0.1,  
    plotmp = T  
  )
```

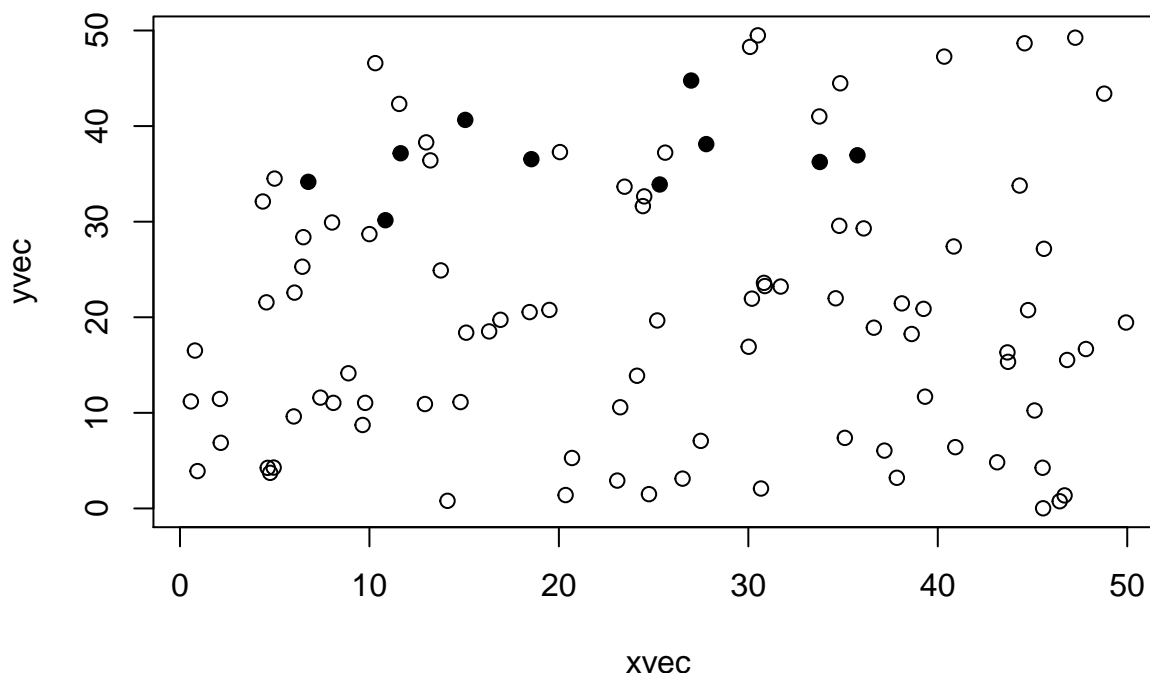
Establishment, shaded = yes



The example with 100 nodes, with management to reduce establishment:

```
estab100.t2 <-  
  estab(  
    decvec = dec100.t2,  
    dispvec = init100.bio,  
    geocoords4n = geolocs100,  
    readprobestabvec4 = F,  
    probestabmean4 = 0.5,  
    probestabsd4 = 0.1,  
    maneffdir4 = 'decrease_estab',  
    maneffmean4n = 0.5,  
    maneffsd4n = 0.1,  
    plotmp = T  
  )
```

Establishment, shaded = yes



```
estab100.t2
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE TRUE TRUE
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE
## [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [73] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [85] FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE
## [97] FALSE TRUE FALSE FALSE
```

5. Science of science component

Function estinfo

The function `estinfo` provides an estimate of the management effect (as a proportional change in the management effect, `obschange`), and indicates whether the estimated management effect size is above a threshold that would trigger communication about the management through a communication network (`com.yes`). This threshold might be determined by economic considerations - for example, whether the management benefit is great enough to convince potential adopters that the technology is worth its cost.

The trigger for information communication could come from a scientific experiment, determining whether scientists communicate the experimental results. In more detailed models, communication about whether management is successful could come from a farm node, determining whether farmers communicate about the management being successful.

The mean effect size of management can be modified using the parameter `maneffmean4s`, standard deviation of the mean effect size with `maneffsd4s`, the threshold for communicating with `maneffthresh4`, and the sampling effort (which increases the precision of the estimate of the management effect size) with `sampeffort4`.

```
estinfo(
  maneffmean4s = 0.5,
  maneffsd4s = 0.5,
  maneffthresh4 = 0.5,
  sampeffort4 = 1
)
```

```
## $obschange
## [1] 0.692828
##
## $com.yes
## [1] TRUE
```

Evaluating multiple time steps

The function INAscene calls on the function ntsteps2 to evaluate changes in the system across multiple time steps

Function ntsteps2

The function ntsteps2 generates outcomes for the linked communication and dispersal networks for the designated number of time steps.

It uses many of the smaller functions described above.

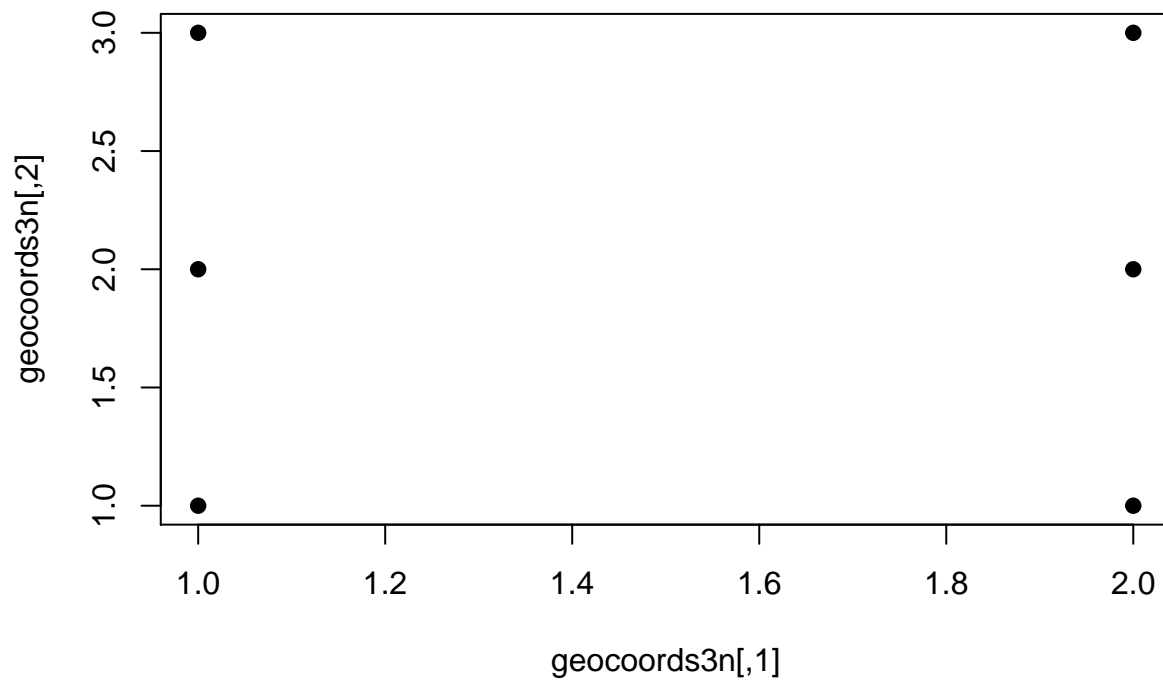
Note that if plotting is called for (plotmpn=T), at each time step maps are illustrated for

- 1 whether information is available at each node
- 2 whether management is adopted at each node
- 3 whether the species has dispersed to each node
- 4 whether the species is established at each node

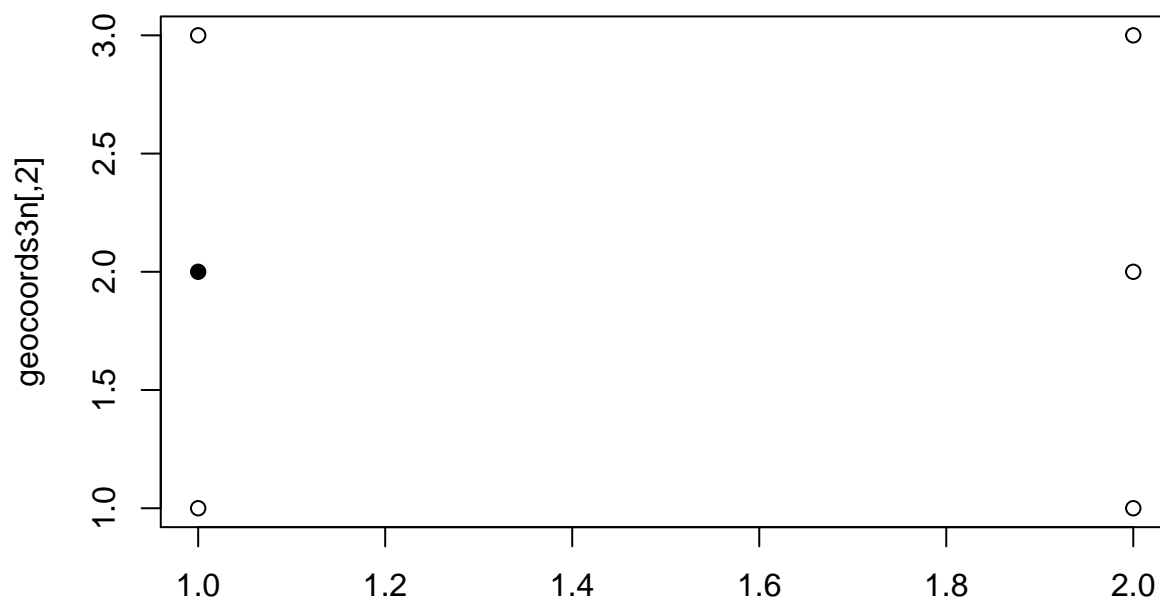
```
x13 <-
  ntsteps2(
    nsteps = 5,
    infon = T,
    geocoords3n = matrix(
      c(1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3),
      byrow = T,
      ncol = 2
    ),
    vect1cn = c(1, 1, 1, 0, 0, 0),
    vect1dn = c(0, 0, 0, 1, 1, 1),
    readseam3 = F,
    seamdist3 = 'powerlaw',
    seampla3 = 1,
    seamplb3 = 1,
    readbpam3 = F,
    bpamdist3 = 'powerlaw',
    bpampla3 = 1,
    bpamplb3 = 1,
    probadoptmean3 = 0.3,
```

```
probadoptsd3 = 0.1,  
probestabmean3 = 0.2,  
probestabsd3 = 0.1,  
maneffdir3 = 'decrease_estab',  
maneffmean3n = 0.5,  
maneffsd3n = 0.1,  
readprobestabvec3 = F,  
readprobadoptvec3 = F,  
plotmpn = T  
)
```

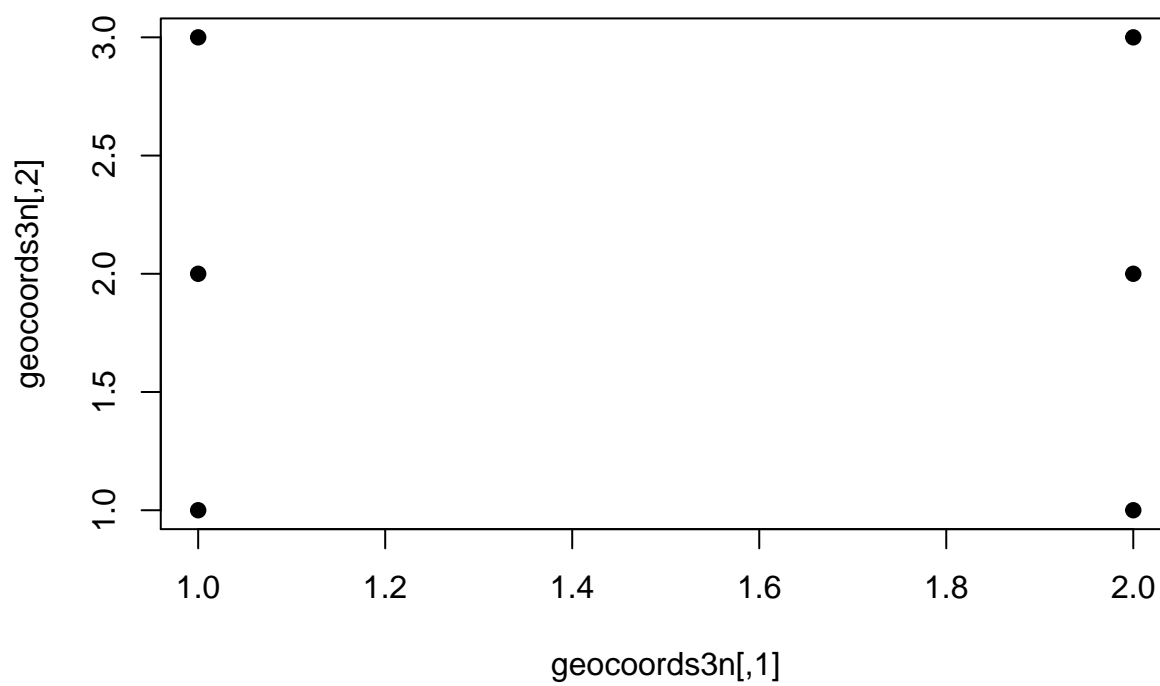
Time 1 , 1 Information, shaded = yes



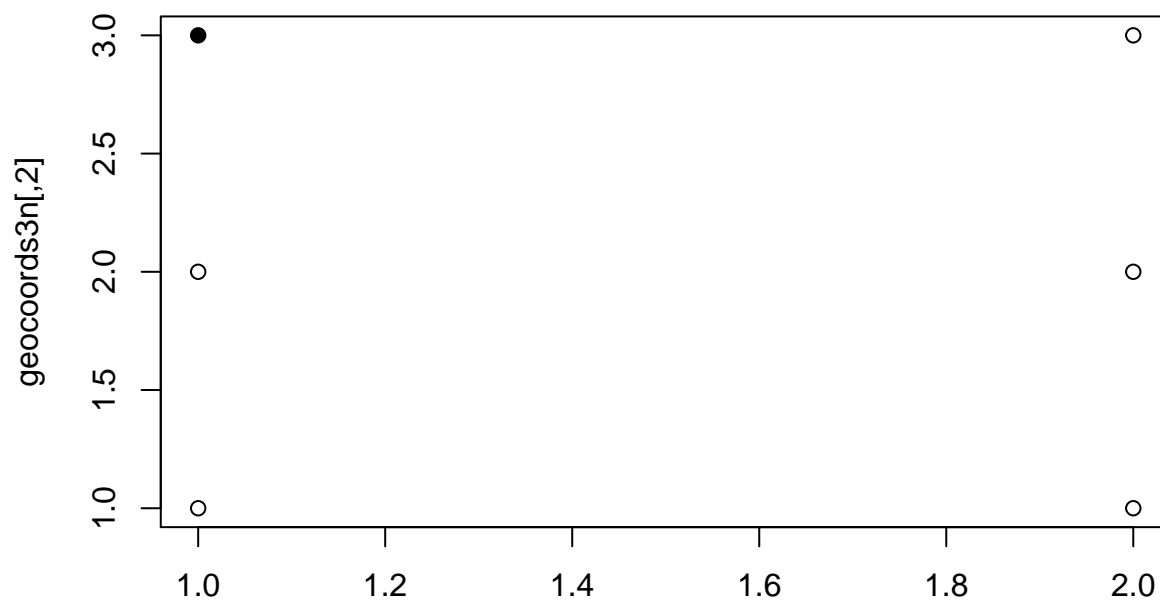
Time 1 , 2 Adoption, shaded = yes



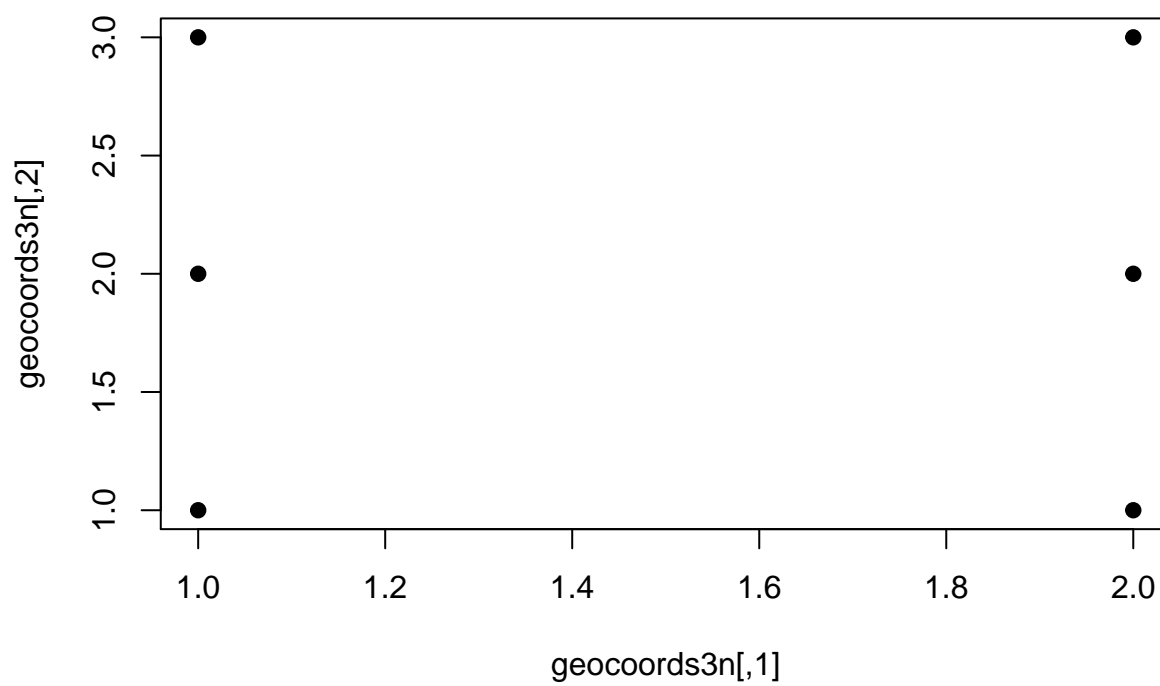
Time 1 , 3 Sp dispersal, shaded = yes



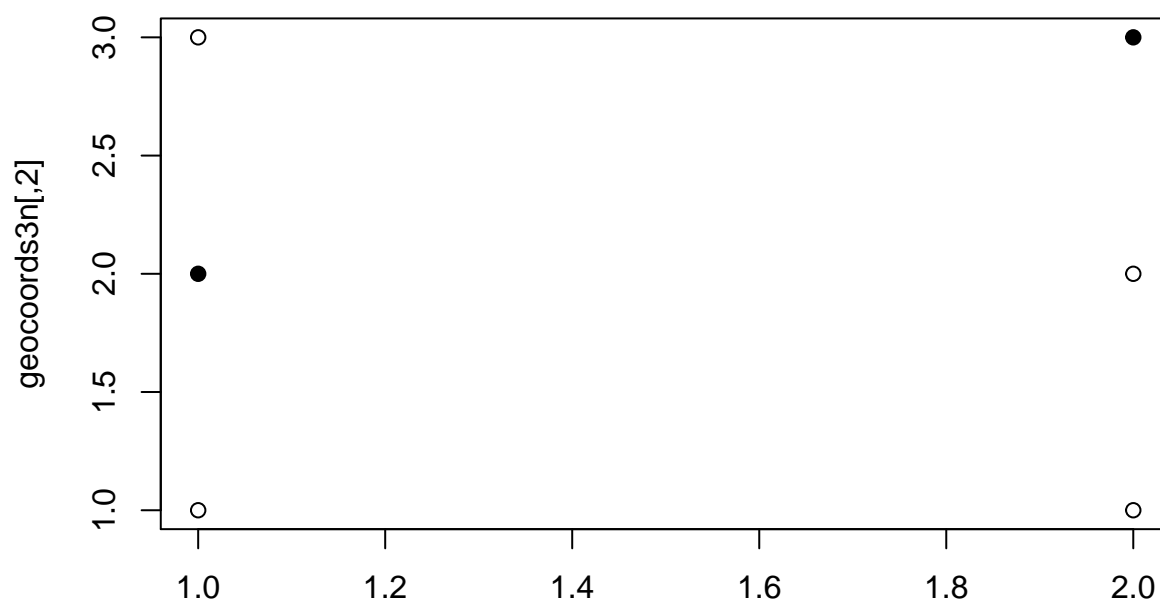
Time 1 , 4 Sp establishment, shaded = yes



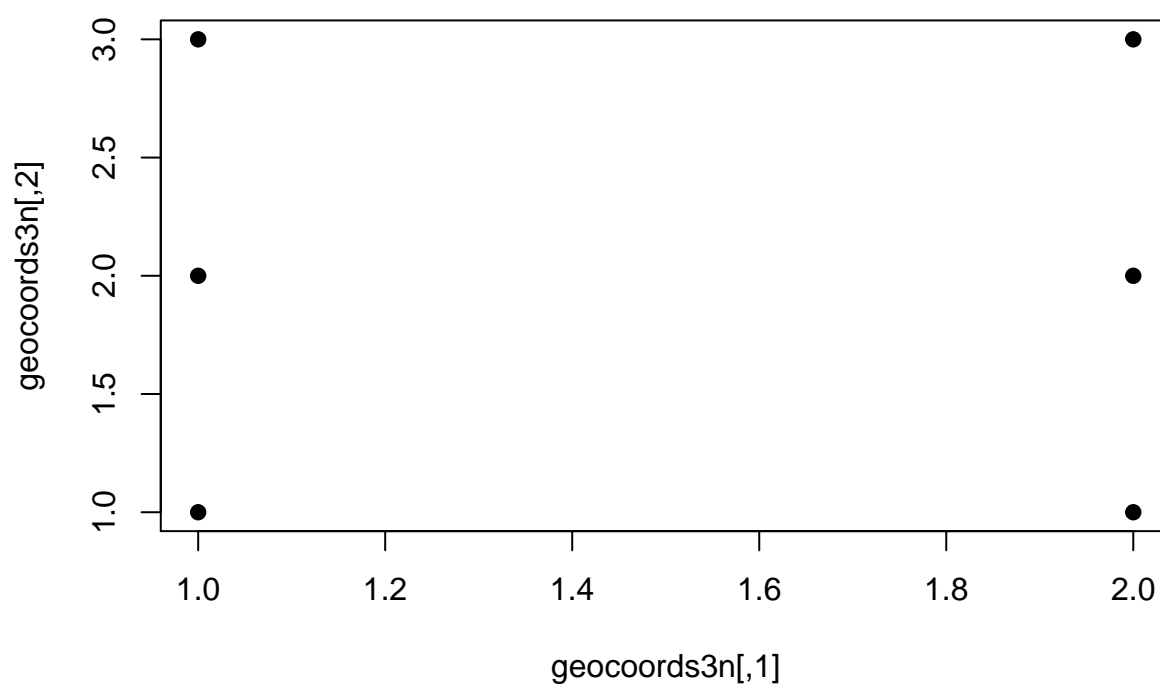
Time 2 , 1 Information, shaded = yes



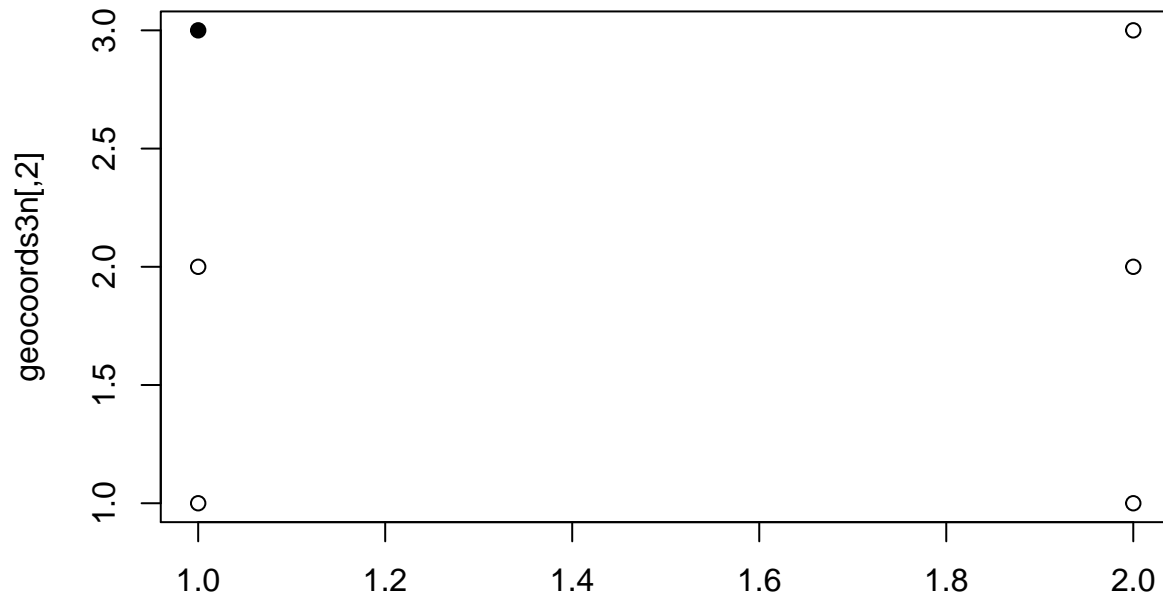
Time 2 , 2 Adoption, shaded = yes



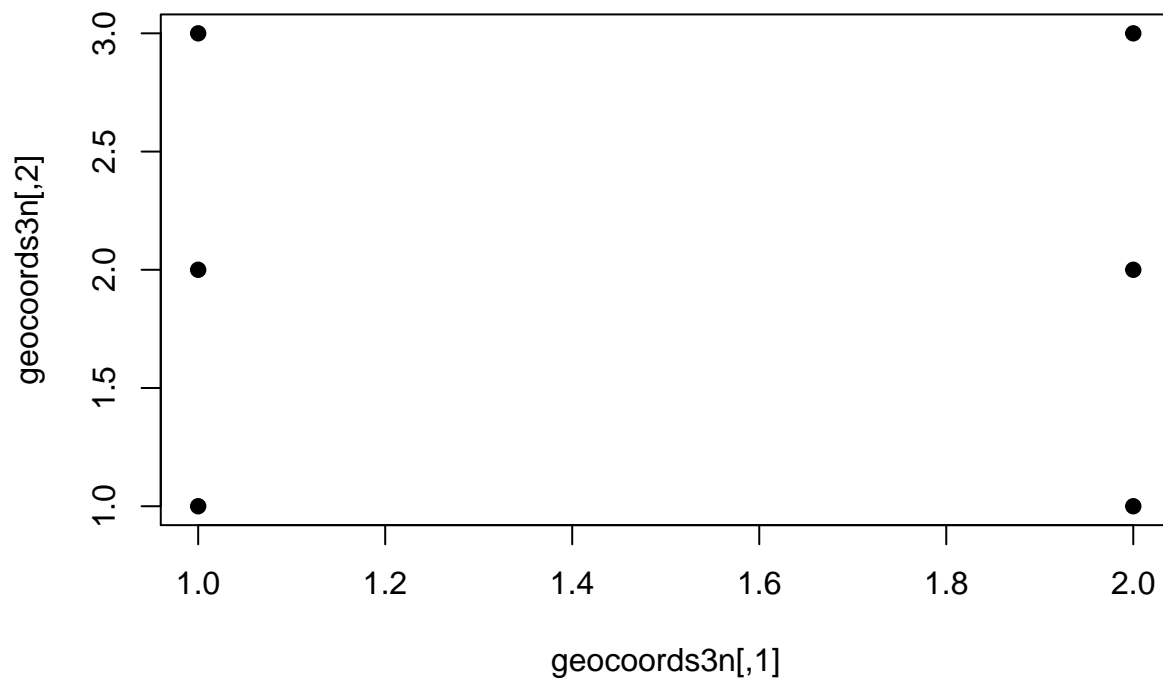
Time 2 , 3 Sp dispersal, shaded = yes



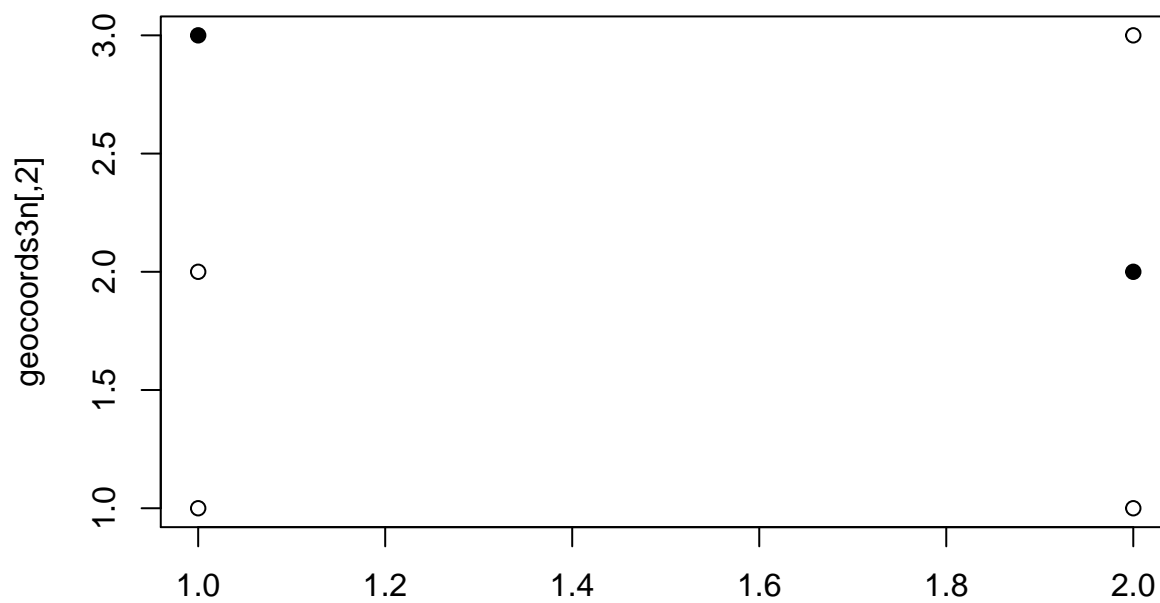
Time 2 , 4 Sp establishment, shaded = yes



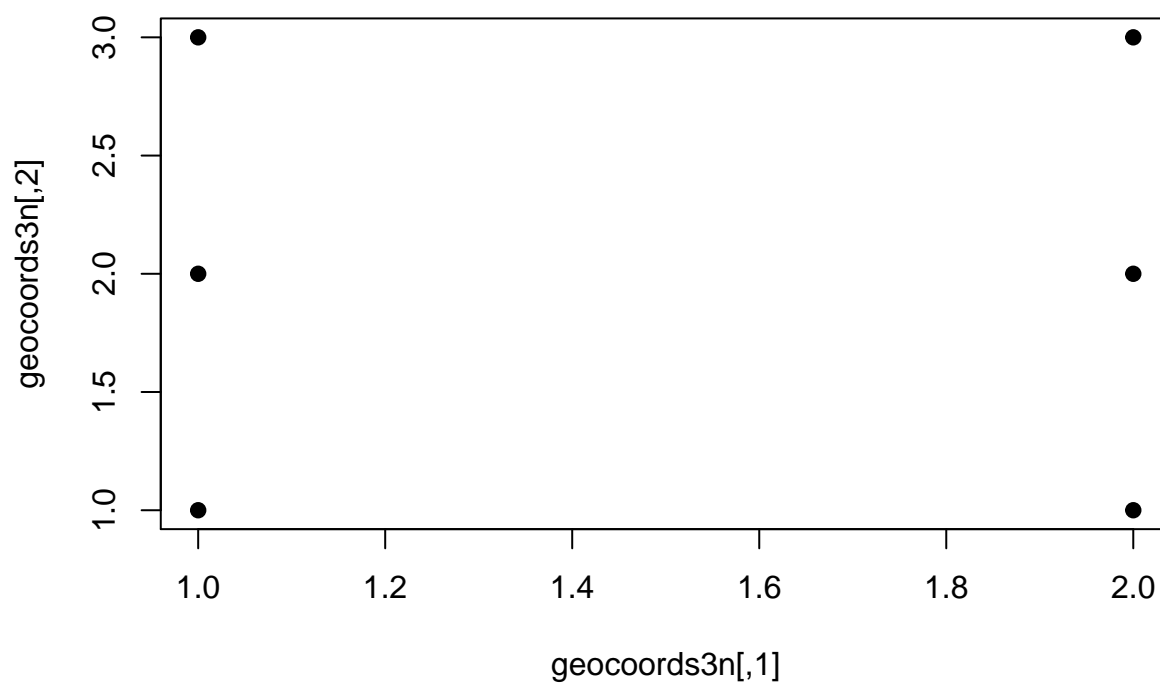
Time 3 , 1 Information, shaded = yes



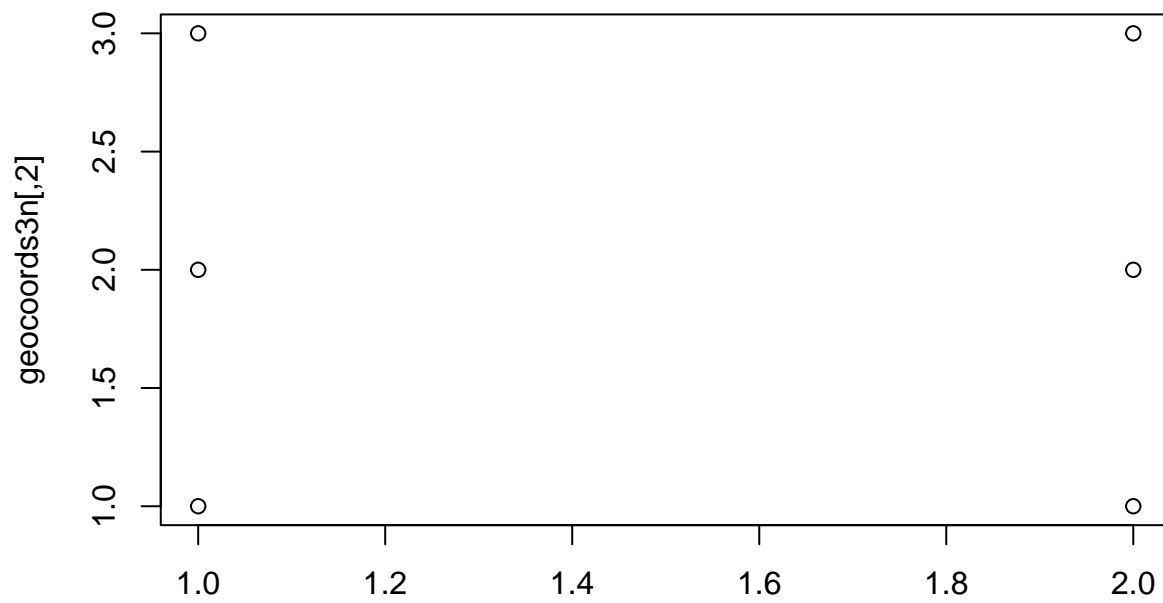
Time 3 , 2 Adoption, shaded = yes



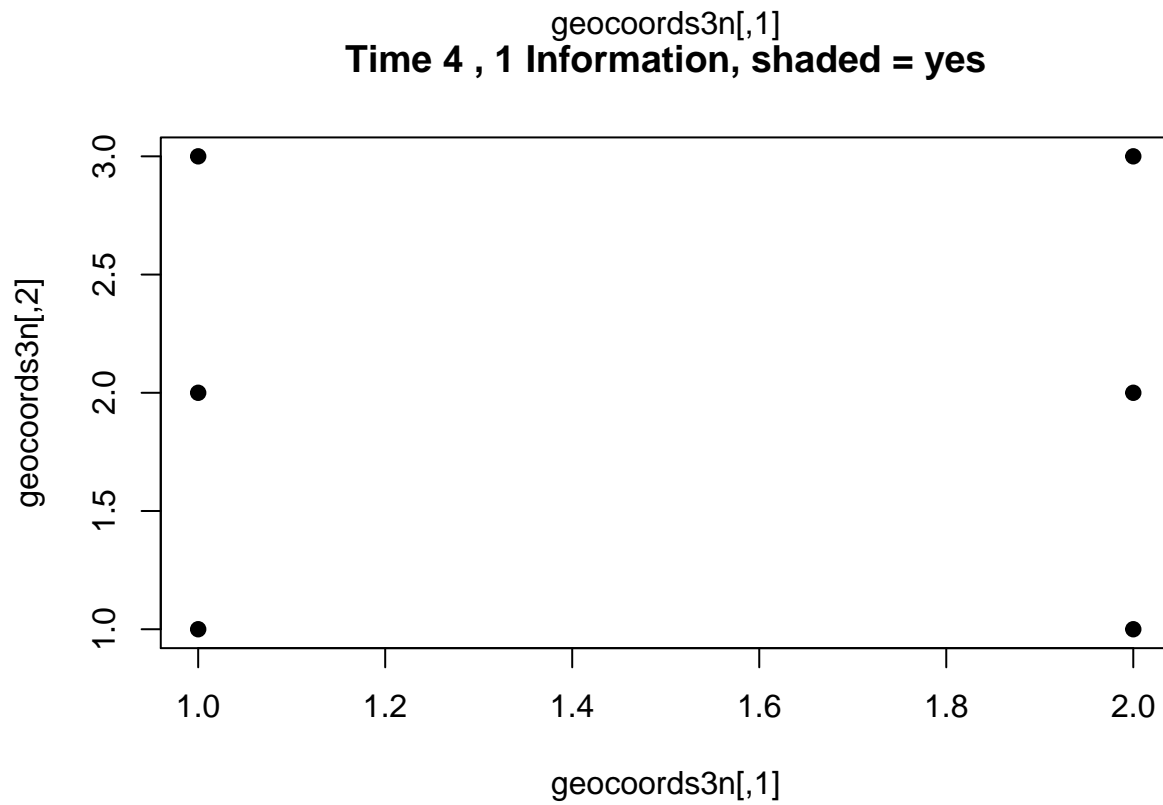
Time 3 , 3 Sp dispersal, shaded = yes



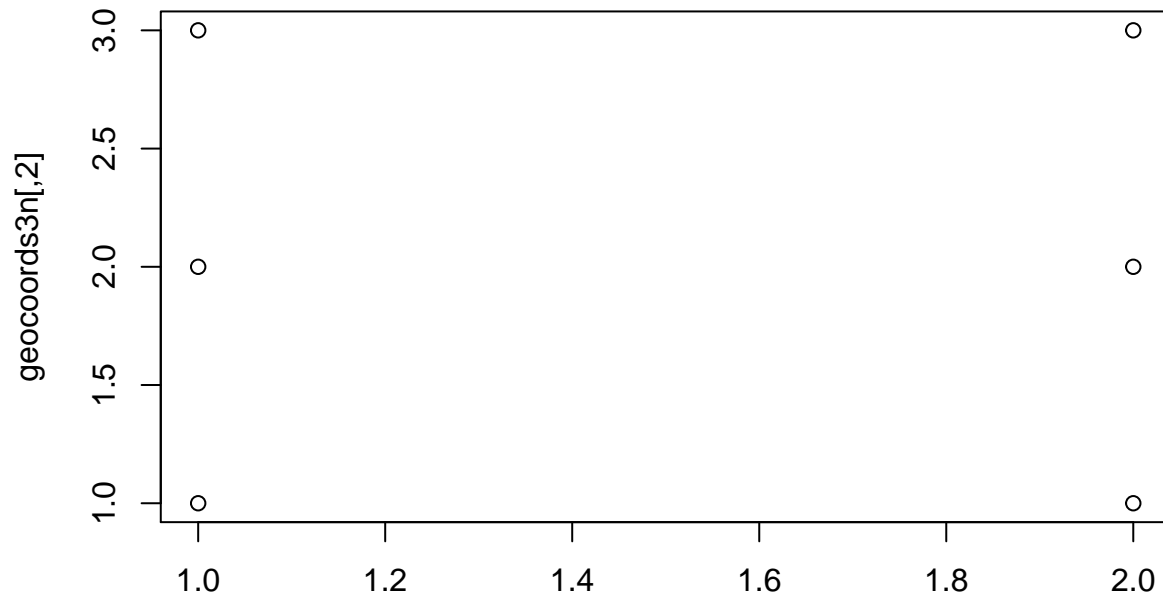
Time 3 , 4 Sp establishment, shaded = yes



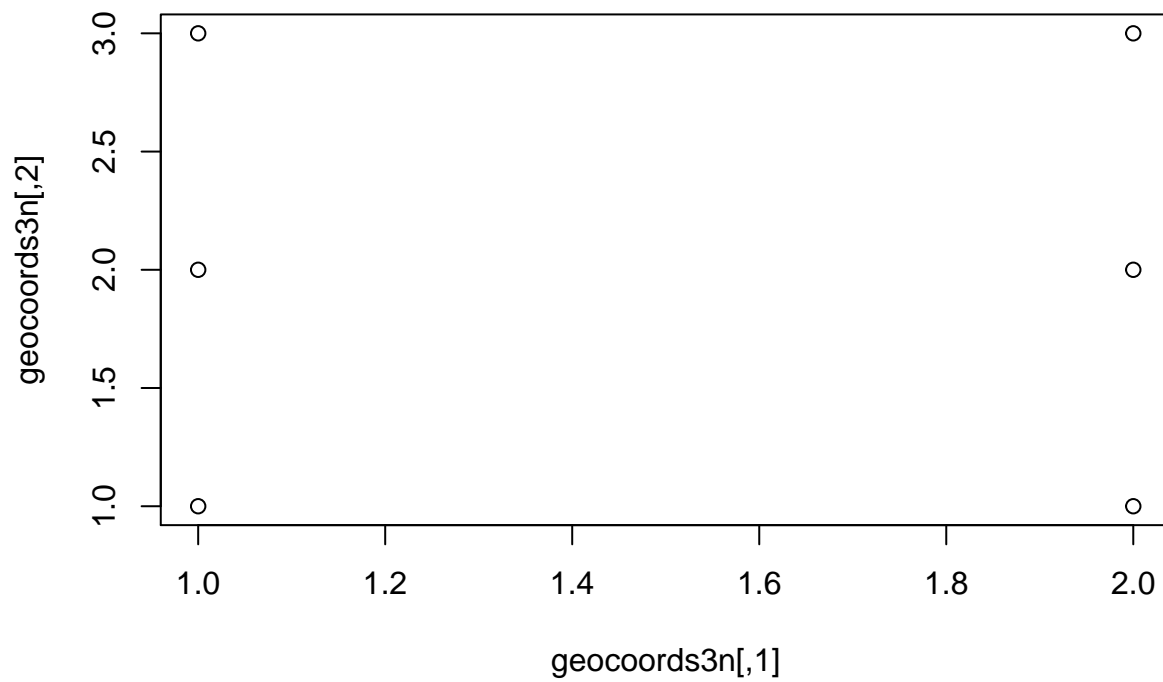
Time 4 , 1 Information, shaded = yes



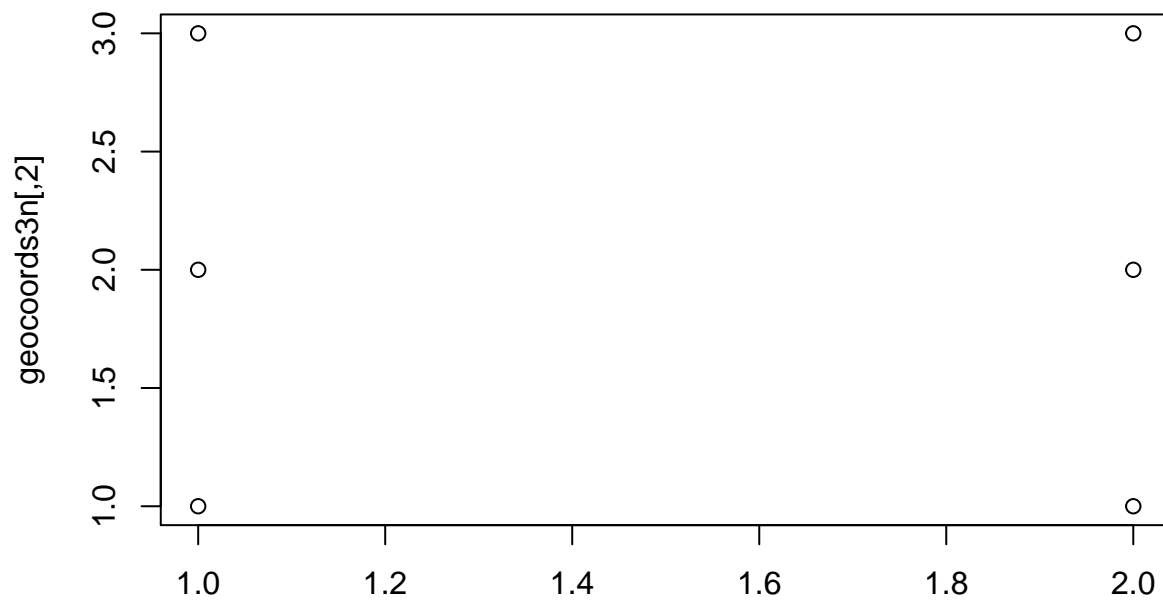
Time 4 , 2 Adoption, shaded = yes



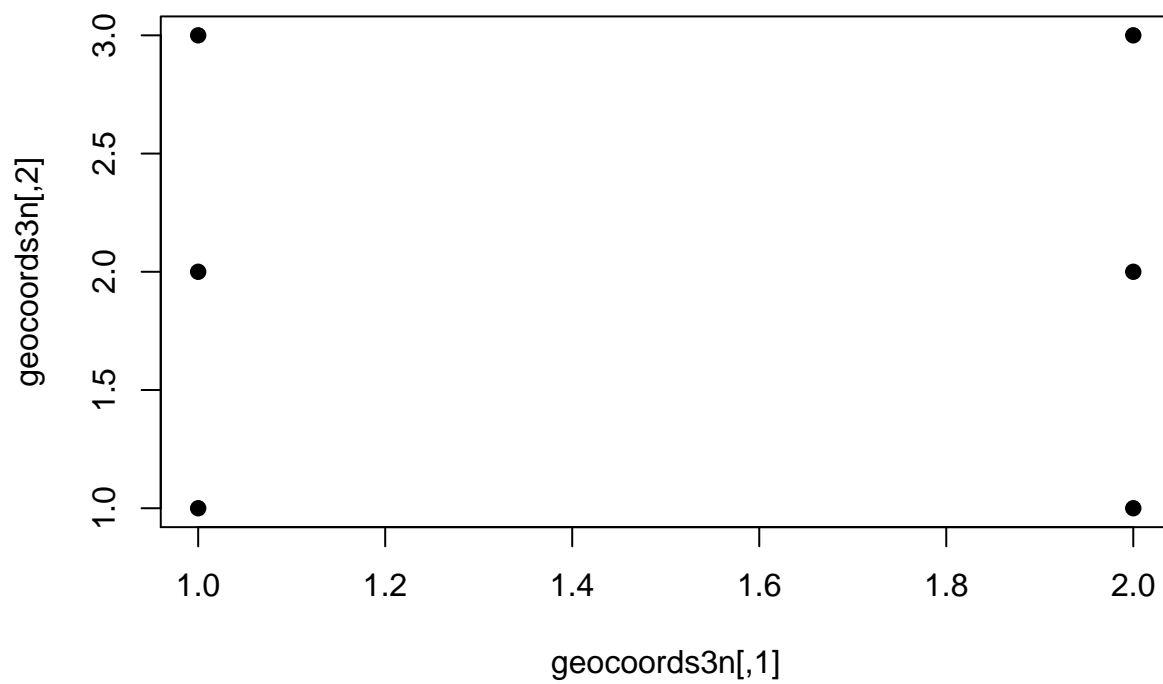
Time 4 , 3 Sp dispersal, shaded = yes



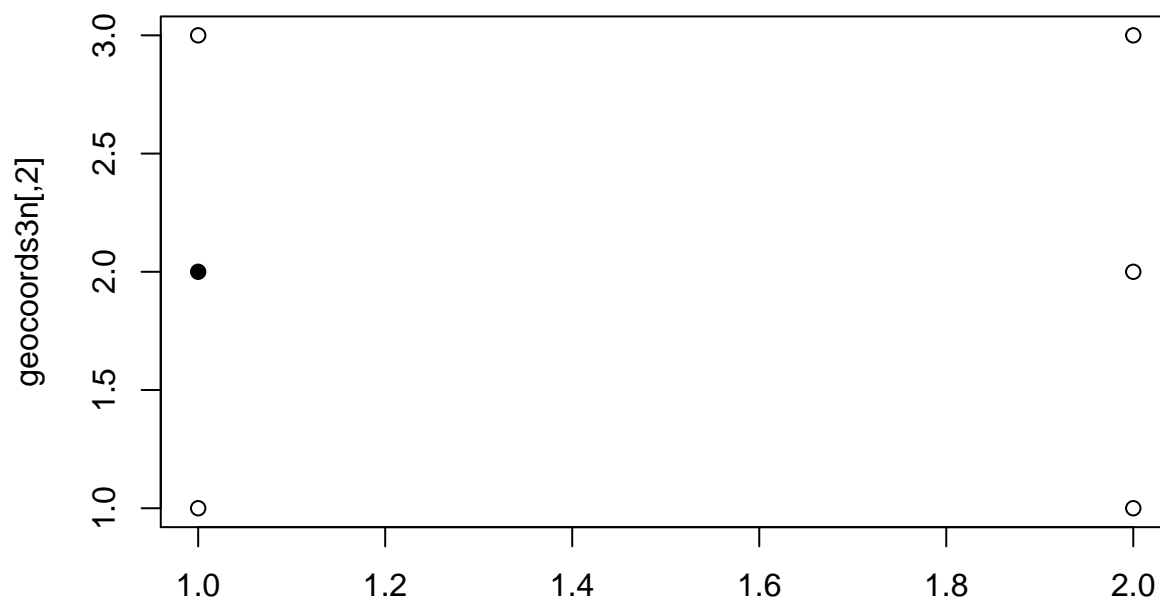
Time 4 , 4 Sp establishment, shaded = yes



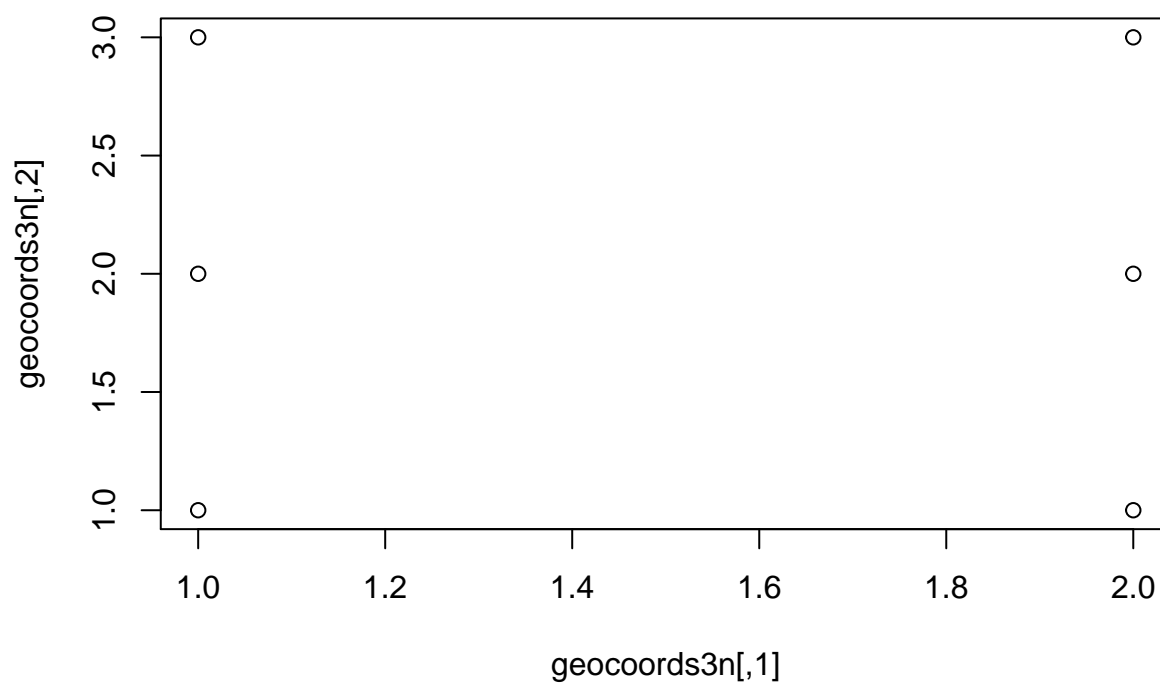
Time 5 , 1 Information, shaded = yes



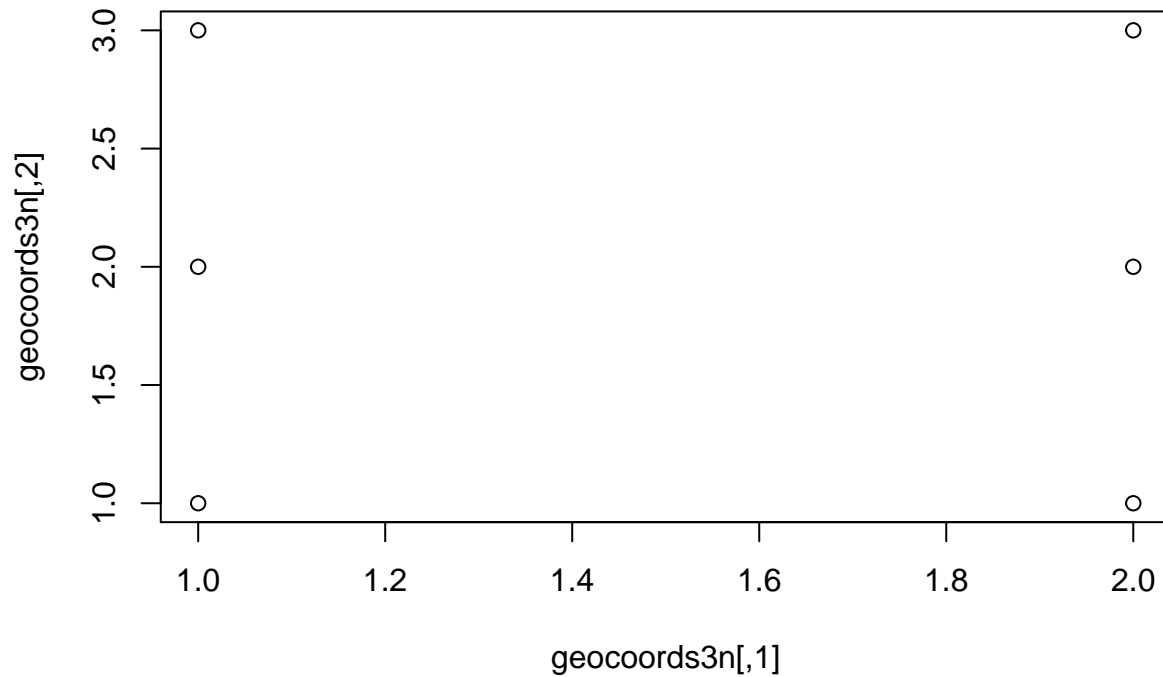
Time 5 , 2 Adoption, shaded = yes



Time 5 , 3 Sp dispersal, shaded = yes



Time 5 , 4 Sp establishment, shaded = yes



```
# This is the socioeconomic adjacency matrix (seam), giving the probability of
# communication occurring
x13$seam3
```

```
##           1           2           3           4           5           6
## 1 1.0000000 1.0000000 0.5000000 1.0000000 0.7071068 0.4472136
## 2 1.0000000 1.0000000 1.0000000 0.7071068 1.0000000 0.7071068
## 3 0.5000000 1.0000000 1.0000000 0.4472136 0.7071068 1.0000000
## 4 1.0000000 0.7071068 0.4472136 1.0000000 1.0000000 0.5000000
## 5 0.7071068 1.0000000 0.7071068 1.0000000 1.0000000 1.0000000
## 6 0.4472136 0.7071068 1.0000000 0.5000000 1.0000000 1.0000000
```

```
# This is the biophysical adjacency matrix (bpam), giving the probability of
# dispersal occurring
x13$bpam3
```

```
##           1           2           3           4           5           6
## 1 1.0000000 1.0000000 0.5000000 1.0000000 0.7071068 0.4472136
## 2 1.0000000 1.0000000 1.0000000 0.7071068 1.0000000 0.7071068
## 3 0.5000000 1.0000000 1.0000000 0.4472136 0.7071068 1.0000000
## 4 1.0000000 0.7071068 0.4472136 1.0000000 1.0000000 0.5000000
## 5 0.7071068 1.0000000 0.7071068 1.0000000 1.0000000 1.0000000
## 6 0.4472136 0.7071068 1.0000000 0.5000000 1.0000000 1.0000000
```

```
# This is the probability of adoption for each socioeconomic node
x13$probadoptvec3
```

```
## [1] 0.4325116 0.2211490 0.3760079 0.1664110 0.2062991 0.3858999
```

```
# This is the probability of establishment for each biophysical node (before
# management is taken into account - confirm!)
x13$probestabvec3
```

```
## [1] 0.0003171551 0.3137911646 0.4074646310 0.1276502582 0.2095335289
## [6] 0.1950357592
```

```
# This is the times series of status vectors for communication about the
# management technology
x13$vect1cL
```

```
## [[1]]
## [1] 1 1 1 0 0 0
##
## [[2]]
## [1] 1 1 1 1 1 1
##
## [[3]]
## [1] 1 1 1 1 1 1
##
## [[4]]
## [1] 1 1 1 1 1 1
##
## [[5]]
## [1] 1 1 1 1 1 1
##
## [[6]]
## [1] 1 1 1 1 1 1
```

```
x13$vect1dL # This is the times series of status vectors for species dispersal
```

```
## [[1]]
## [1] 0 0 0 1 1 1
##
## [[2]]
## [1] 1 1 1 1 1 1
##
## [[3]]
## [1] 1 1 1 1 1 1
##
## [[4]]
## [1] 1 1 1 1 1 1
##
## [[5]]
## [1] 0 0 0 0 0 0
##
## [[6]]
## [1] 0 0 0 0 0 0
```

```
x13$decvecL # This is the times series of status vectors for decisions about adoption
```

```
## [[1]]
## [1] FALSE TRUE FALSE FALSE FALSE FALSE
##
## [[2]]
## [1] FALSE TRUE FALSE FALSE FALSE TRUE
##
## [[3]]
## [1] FALSE FALSE TRUE FALSE TRUE FALSE
##
## [[4]]
## [1] FALSE FALSE FALSE FALSE FALSE FALSE
##
## [[5]]
## [1] FALSE TRUE FALSE FALSE FALSE FALSE
```

```
x13$estabvecL # This is the times series of status vectors for establishment
```

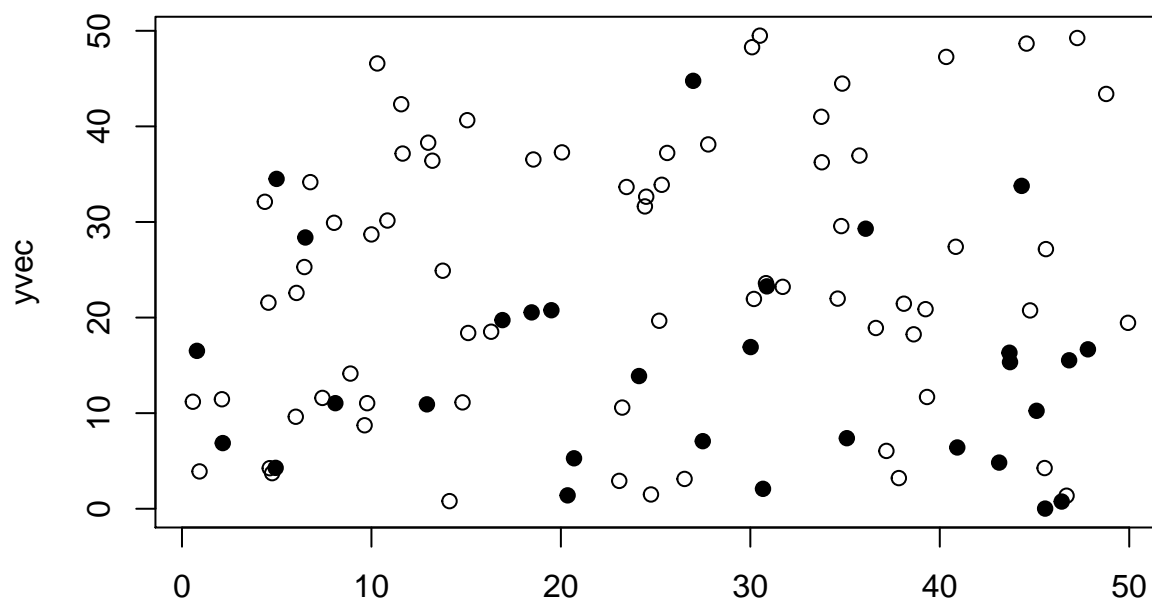
```
## [[1]]
## [1] FALSE FALSE TRUE FALSE FALSE FALSE
##
## [[2]]
## [1] FALSE FALSE TRUE FALSE FALSE FALSE
##
## [[3]]
## [1] FALSE FALSE FALSE FALSE FALSE FALSE
##
## [[4]]
## [1] FALSE FALSE FALSE FALSE FALSE FALSE
##
## [[5]]
## [1] FALSE FALSE FALSE FALSE FALSE FALSE
```

For the example with 100 nodes:

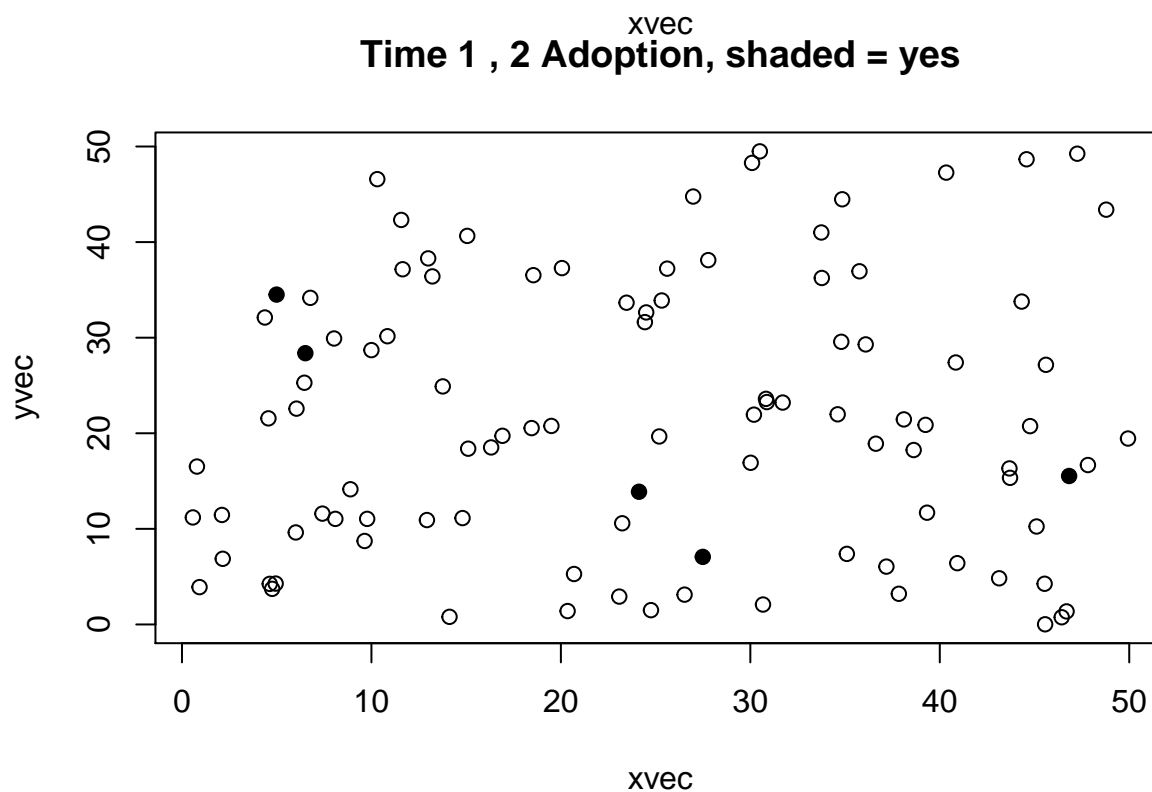
```
tstep100 <-
  ntsteps2(
    nsteps = 5,
    infon = T,
    geocoords3n = geolocs100,
    vect1cn = init100.info,
    vect1dn = init100.bio,
    readseam3 = T,
    seam3 = adjmat100.info,
    readbpam3 = T,
    bpam3 = adjmat100.bio,
    probadoptmean3 = 0.3,
    probadoptsd3 = 0.1,
    probestabmean3 = 0.2,
    probestabsd3 = 0.1,
    maneffdir3 = 'decrease_estab',
    maneffmean3n = 0.5,
    maneffsd3n = 0.1,
    readprobestabvec3 = F,
```

```
readprobadoptvec3 = F,  
plotmpn = T  
)
```

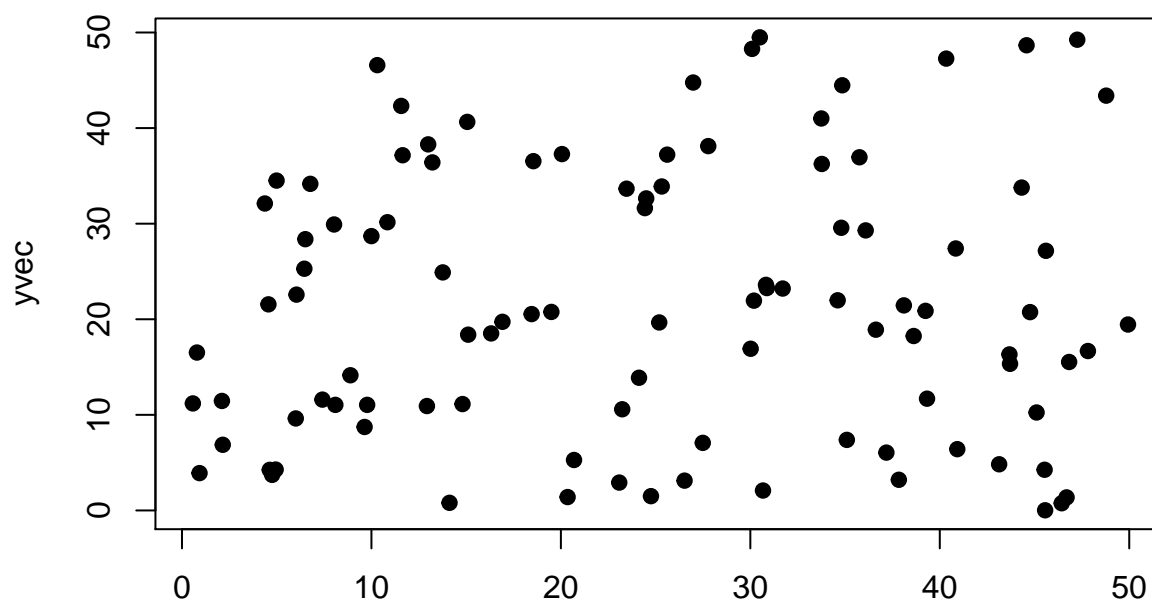
Time 1 , 1 Information, shaded = yes



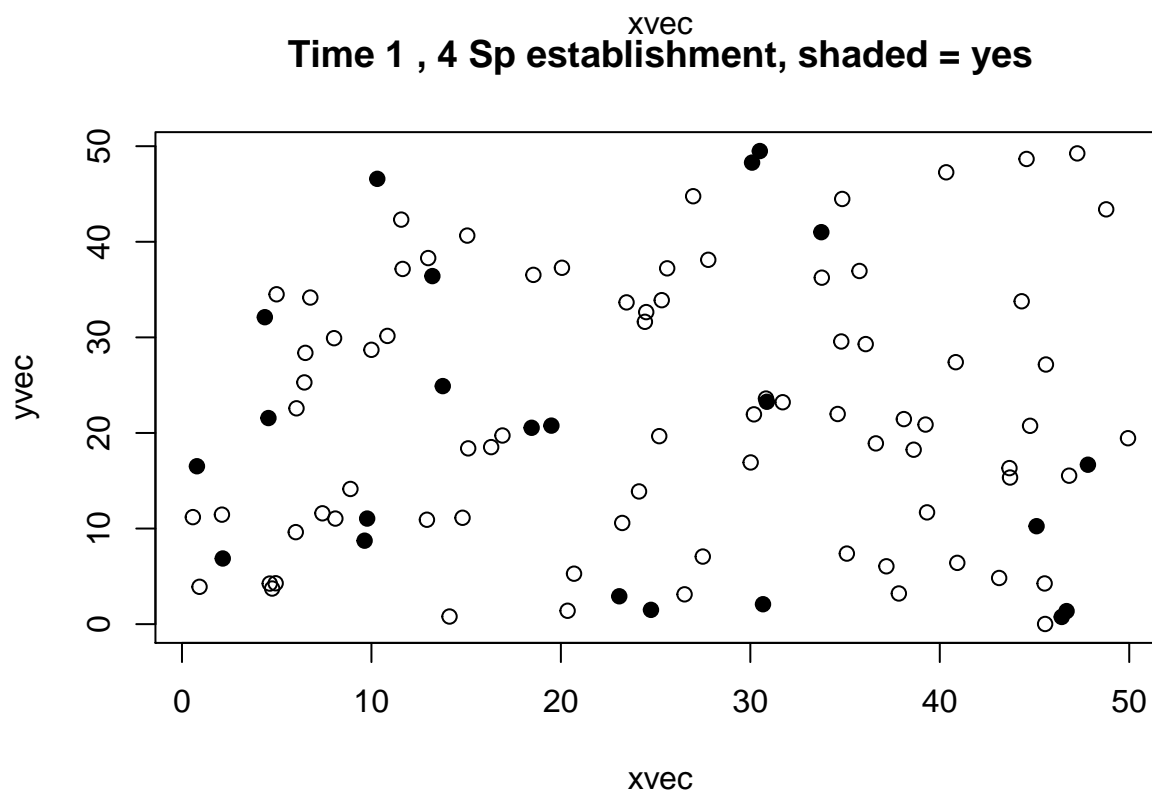
Time 1 , 2 Adoption, shaded = yes



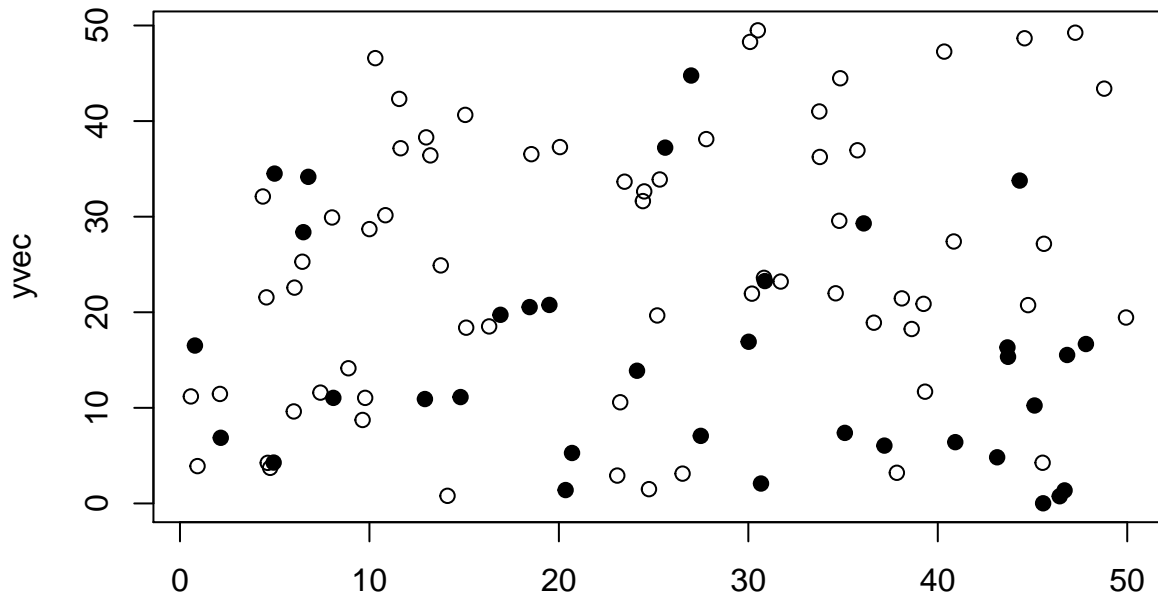
Time 1 , 3 Sp dispersal, shaded = yes



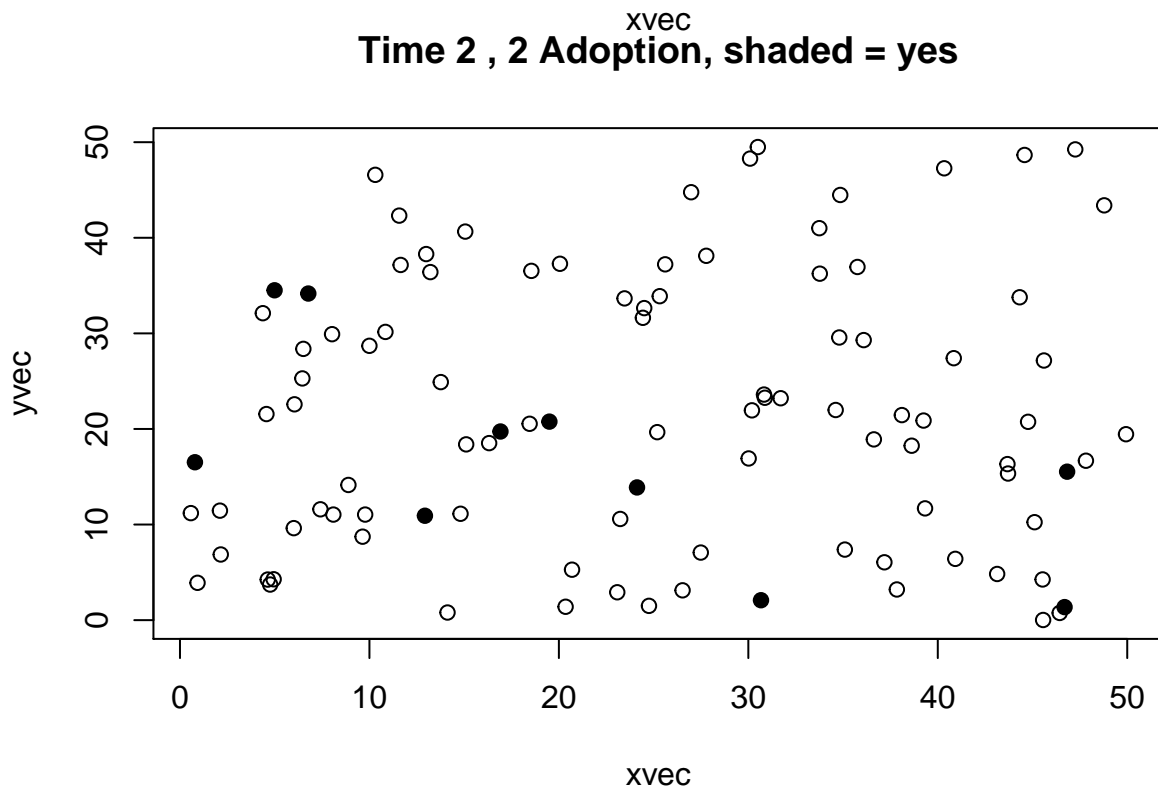
Time 1 , 4 Sp establishment, shaded = yes



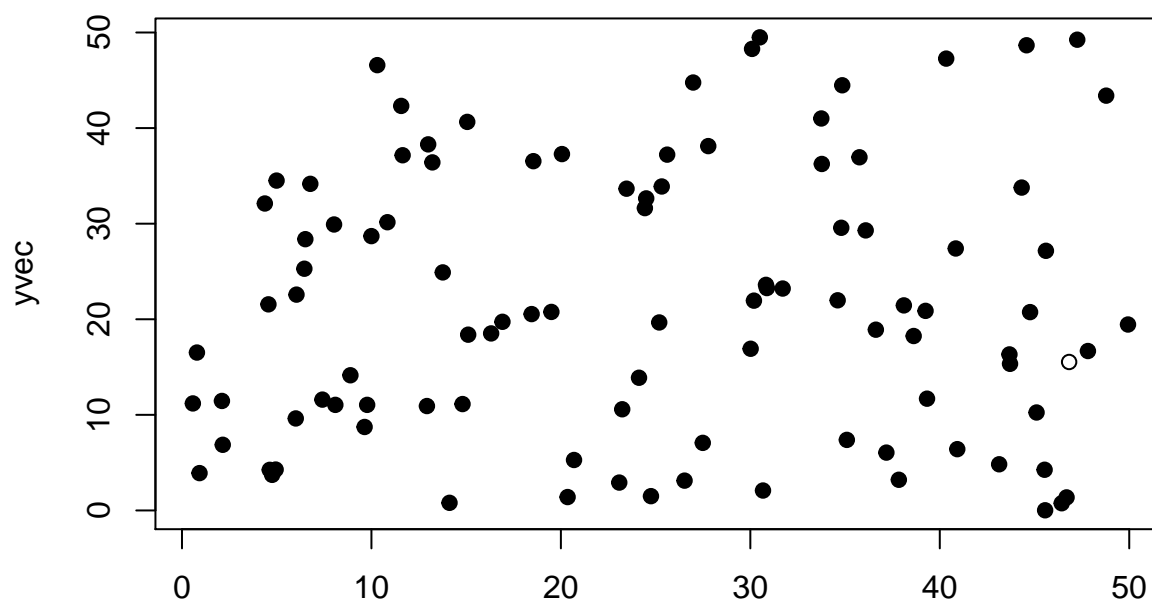
Time 2 , 1 Information, shaded = yes



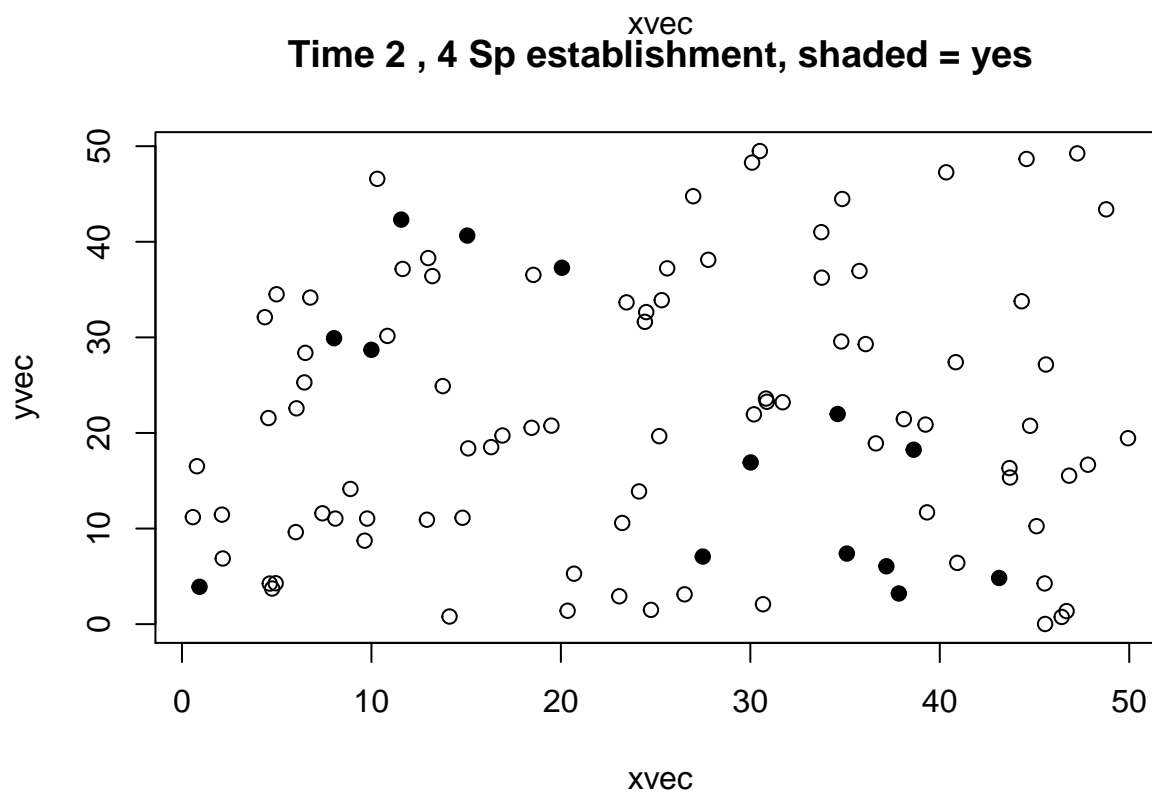
Time 2 , 2 Adoption, shaded = yes



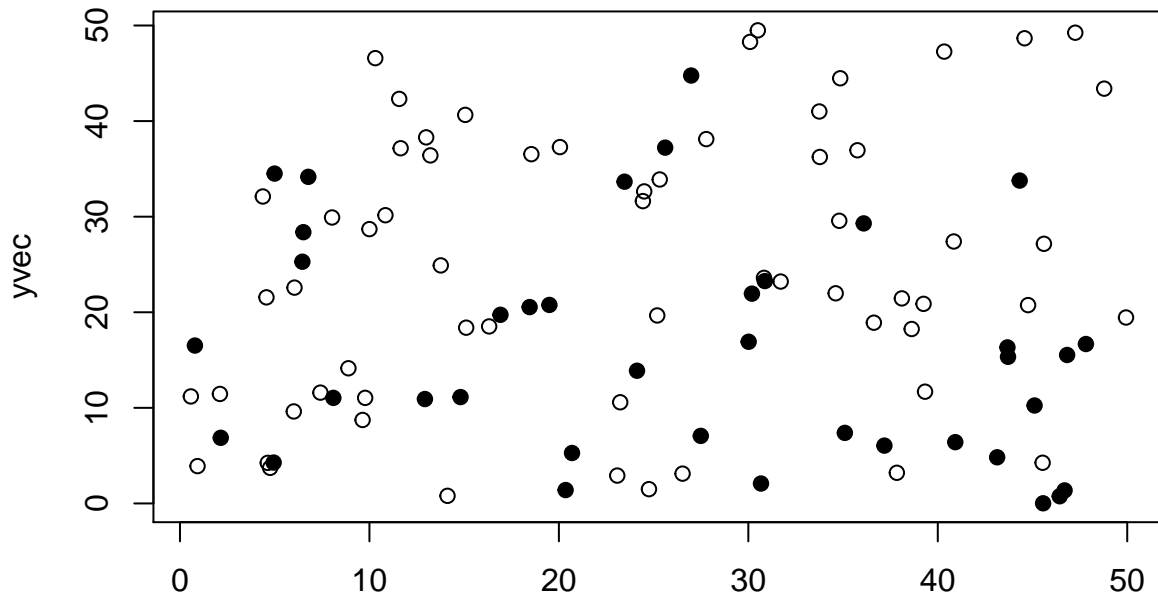
Time 2 , 3 Sp dispersal, shaded = yes



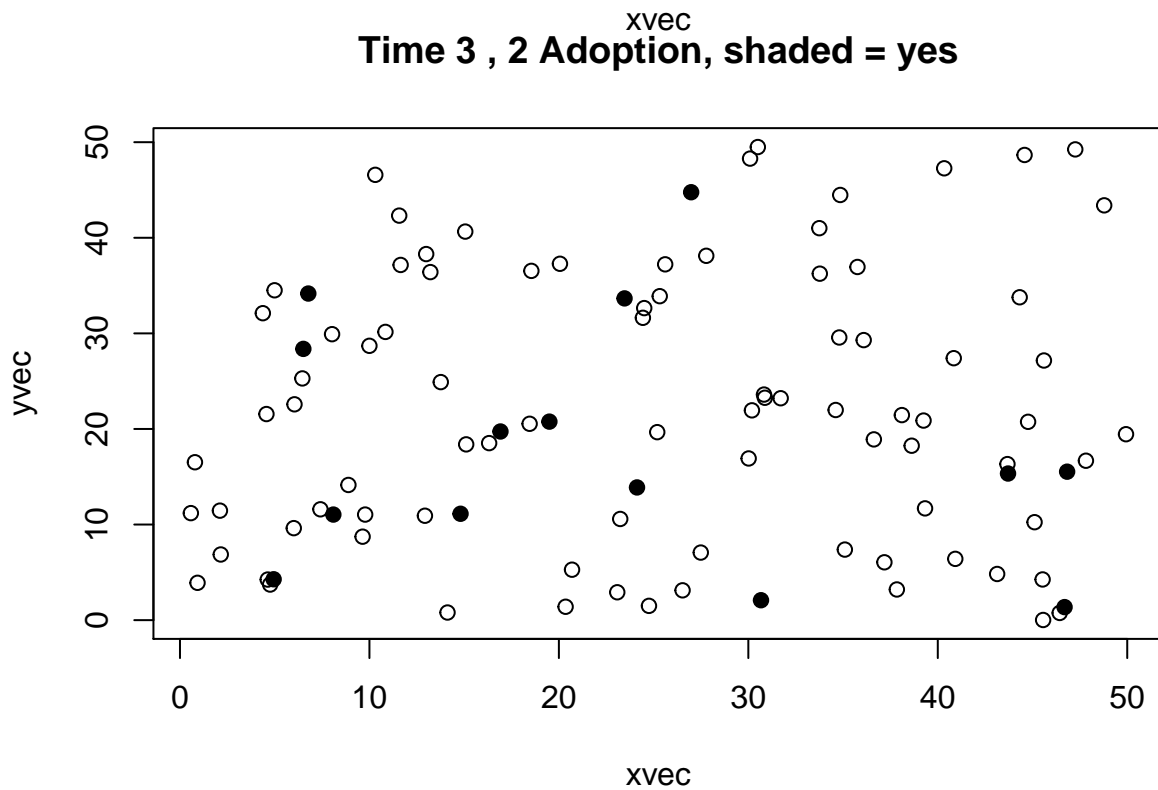
Time 2 , 4 Sp establishment, shaded = yes



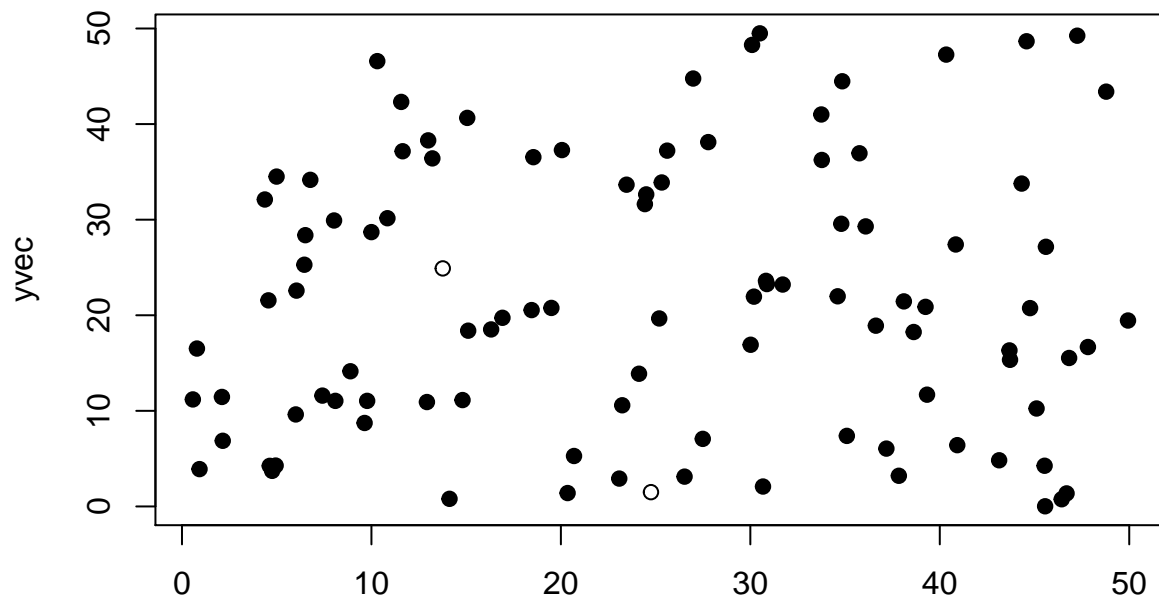
Time 3 , 1 Information, shaded = yes



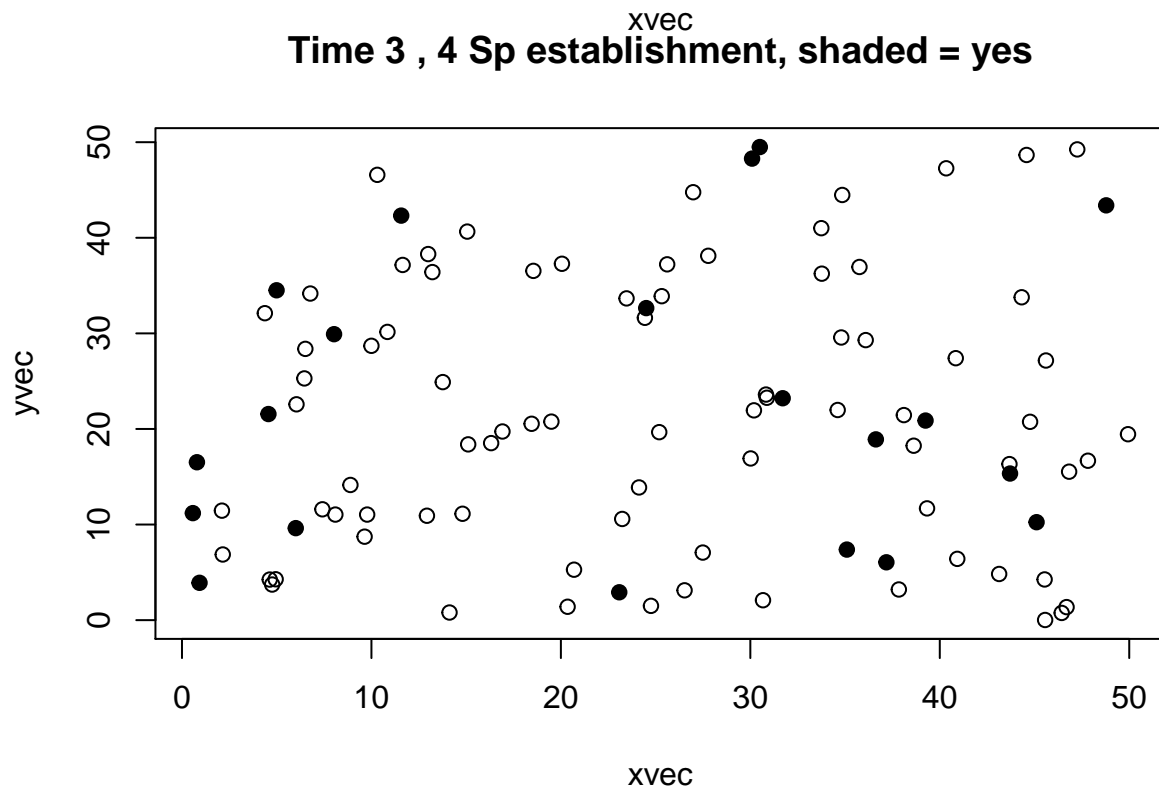
Time 3 , 2 Adoption, shaded = yes



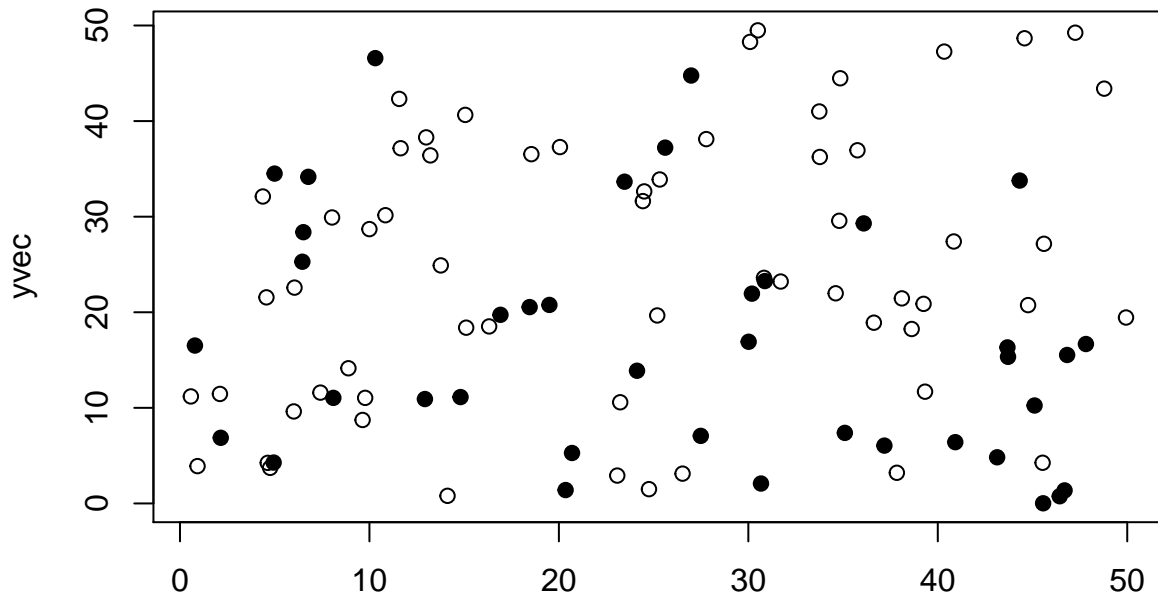
Time 3 , 3 Sp dispersal, shaded = yes



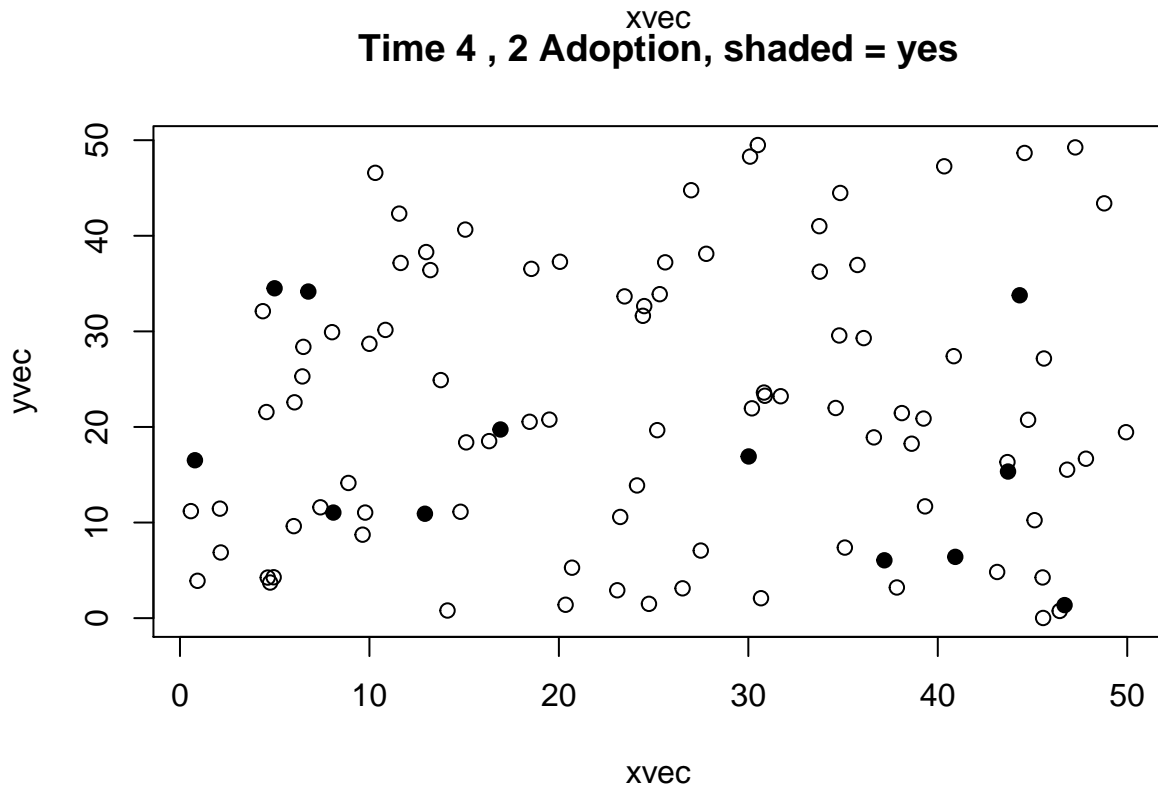
Time 3 , 4 Sp establishment, shaded = yes



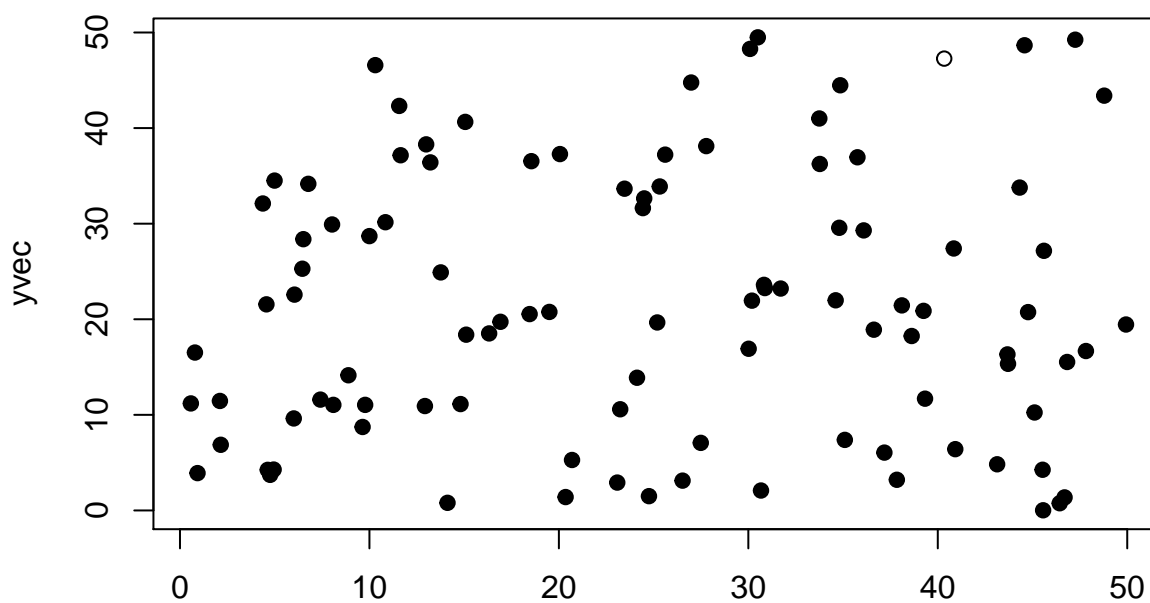
Time 4 , 1 Information, shaded = yes



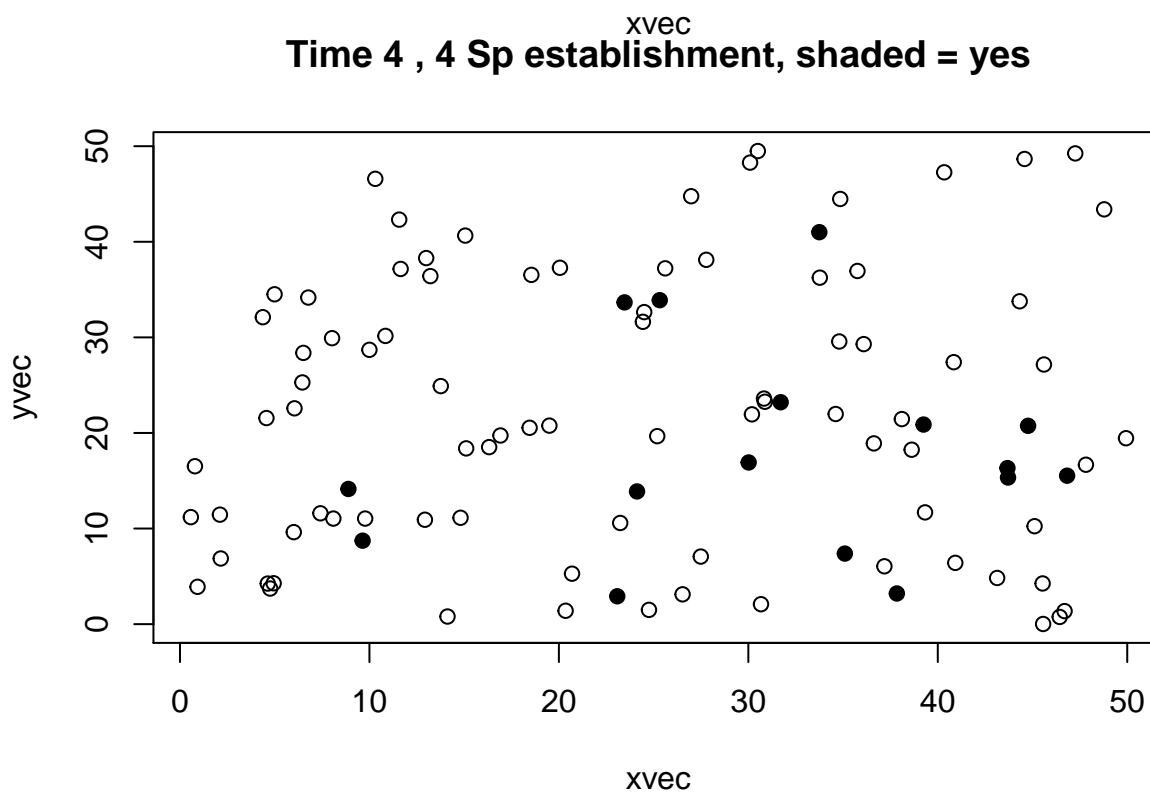
Time 4 , 2 Adoption, shaded = yes



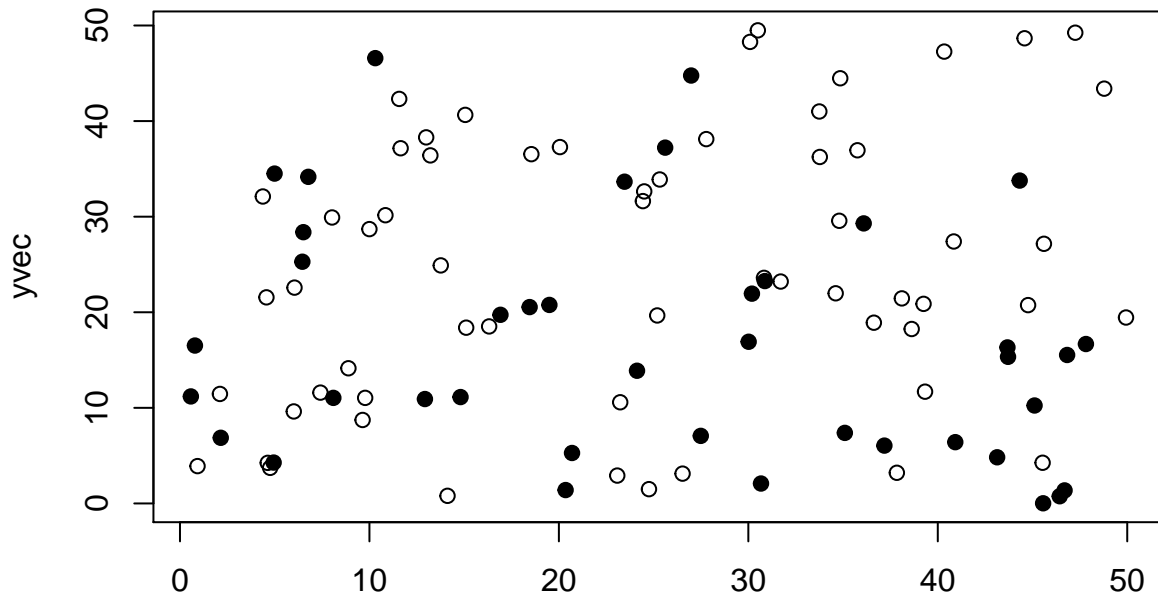
Time 4 , 3 Sp dispersal, shaded = yes



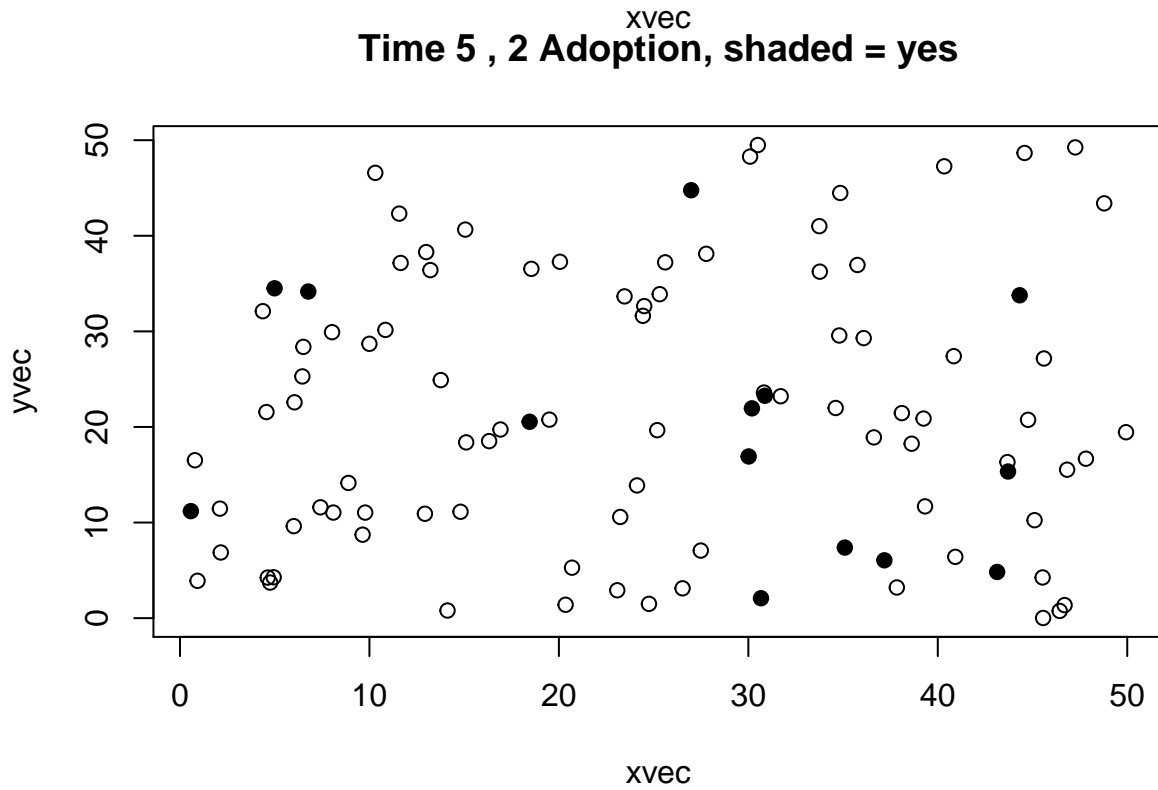
Time 4 , 4 Sp establishment, shaded = yes



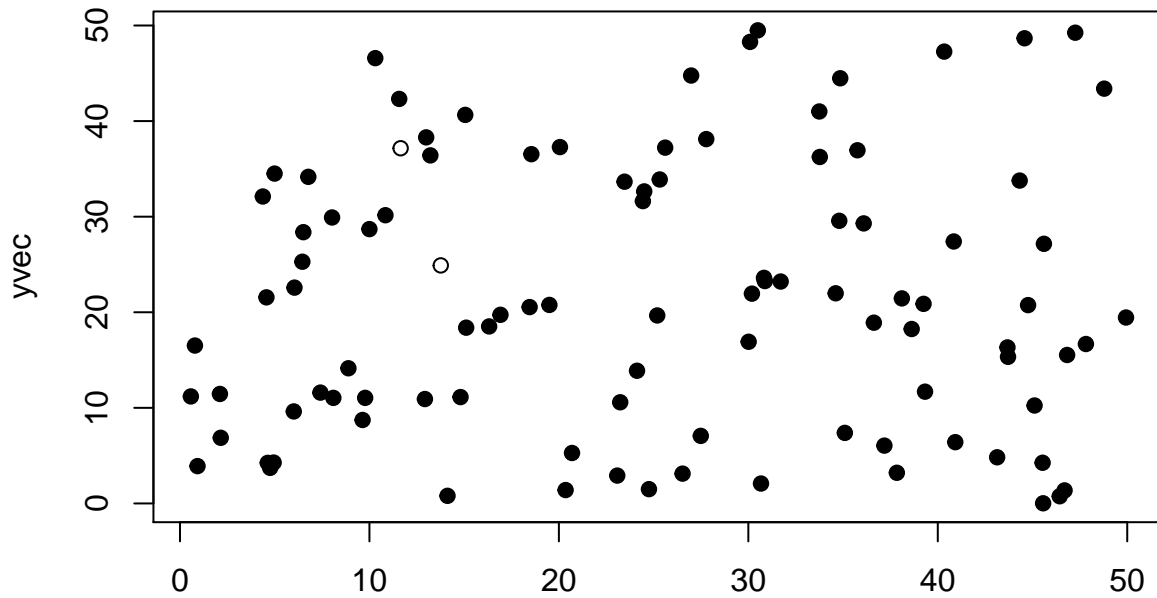
Time 5 , 1 Information, shaded = yes



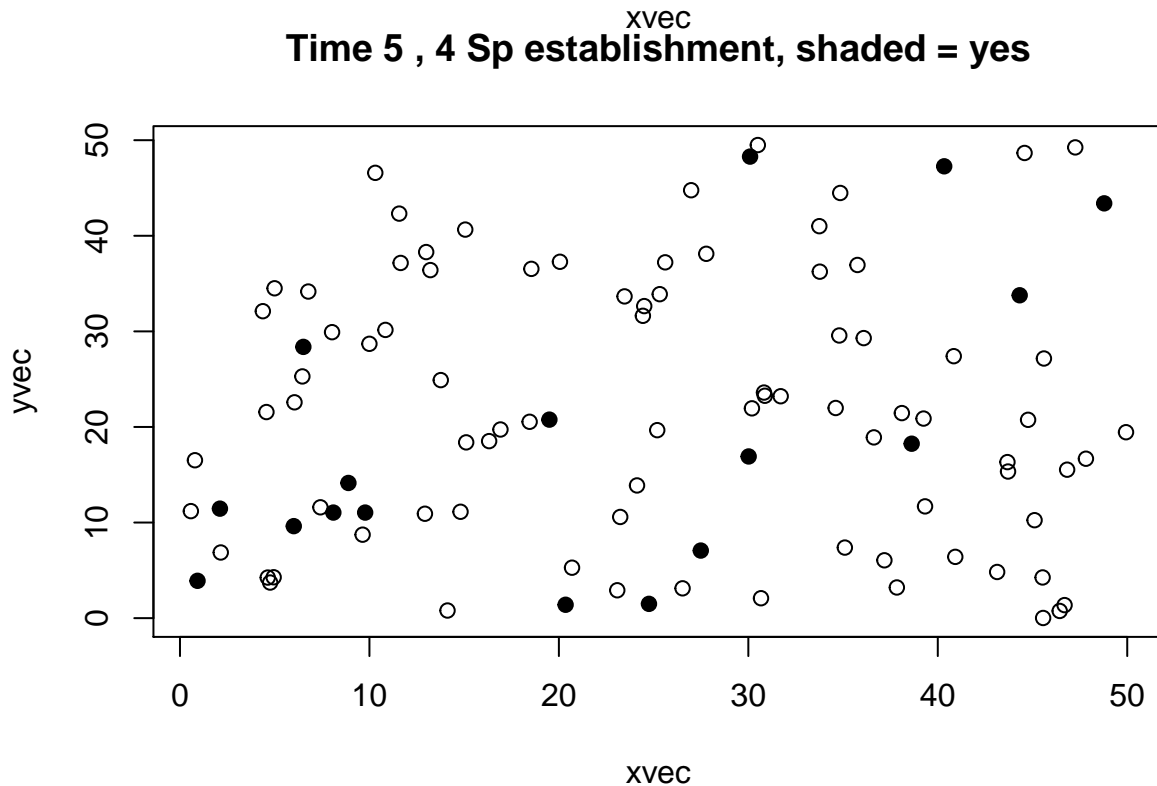
Time 5 , 2 Adoption, shaded = yes



Time 5 , 3 Sp dispersal, shaded = yes



Time 5 , 4 Sp establishment, shaded = yes



Evaluating multiple realizations for the same set of parameters

The INAscene function calls the function multisame2 to evaluate multiple realizations for the same set of parameter values. Because there are several stochastic components in the scenario analyses, the distribution of outcomes for the same parameter values will often be interesting.

Function multisame2

Shown here, for the four statuses being evaluated, are the 5th percentile, the mean, and the 95th percentile across the indicated number of realizations (nreals) and the indicated number of time steps (nsteps).

```
x3 <-
  multisame2(
    nreals2 = 2,
    nimesteps = 3,
    usethreshman2 = F,
    readgeocoords2 = T,
    geocoords2 = matrix(runif(n = 100) * 100, byrow = T, ncol = 2),
    maneffdir2 = 'decrease_estab',
    maneffmean2 = 0.5,
    maneffsd2 = 0.1,
    maneffthresh2 = 0.5,
    sampeffort2 = 1,
    xrange2 = NA,
    yrange2 = NA,
    numnodes2 = NA,
    randgeo2 = NA,
    readinitinfo2 = F,
    initinfo.dist2 = 'random',
    initinfo.n2 = 5,
    initinfo.norp2 = 'num',
    initinfo.p2 = 0.05,
    readinitbio2 = F,
    initbio.dist2 = 'upedge',
    initbio.n2 = 5,
    initbio.norp2 = 'num',
    initbio.p2 = 0.05,
    readseam2 = F,
    seam2 = NA,
    seamdist2 = 'powerlaw',
    seampla2 = 1,
    seamlb2 = 1,
    seamrandp2 = NA,
    readbpam2 = F,
    bpam2 = NA,
    bpamdist2 = 'powerlaw',
    bpampla2 = 1,
    bpamlb2 = 1,
    bpamrandp2 = NA,
    probadoptmean2 = 0.5,
    probadoptsd2 = c(0, 1),
    probadoptvec2 = NA,
    readproadoptvec2 = F,
    probestabmean2 = 0.5,
    probestabsd2 = 0.1,
    readprobestabvec2 = F,
    probestabvec2 = NA,
    doplot2 = F
  )
```

```
c(x3$com5, x3$meancom, x3$com95) # communication status
```

```
##      5%      95%  
## 0.783 0.810 0.837
```

```
c(x3$dec5, x3$meandec, x3$dec95) # decision status
```

```
##      5%      95%  
## 0.42 0.42 0.42
```

```
c(x3$disp5, x3$meandisp, x3$disp95) # sp dispersal status
```

```
##      5%      95%  
## 0.442 0.460 0.478
```

```
c(x3$estab5, x3$meanestab, x3$estab95) # sp establishment status
```

```
##      5%      95%  
## 0.142 0.160 0.178
```

Function INAscene

The function INAscene allows evaluation of multiple parameter combinations, as described in the INA user guide available through a link at www.garrettlab.com/ina. It is the function that you are likely to use most for scenario analyses, once you are used to the structure of the functions that it uses, described above.

Use of INAscene is illustrated further in additional vignettes:

Garrett, K. A. 2020. A vignette for the INA R package: Evaluating potential adaptation plans for regional management under global change using the INAscene function.

Garrett, K. A. 2020. A vignette for the INA R package: Evaluating the likelihood of project success in scenario analyses for regional management using the INAscene function.

The following plots give one illustration of INAscene, looking at the effects of changing the mean probability of adoption of a technology for one scenario.

```
sens.probadoptmean <-  
  INAscene(  
    nreals = 15,  
    nimesteps = 3,  
    doplot = F,  
    readgeocoords = F,  
    geocoords = NA,  
    numnodes = 50,  
    xrange = c(0, 10),  
    yrange = c(0, 20),  
    randgeo = T,  
    readinitinfo = F,  
    initinfo = NA,  
    initinfo.norp = 'num',  
    initinfo.n = 5,
```

```

initinfo.p = NA,
initinfo.dist = 'random',
readinitbio = F,
initbio = NA,
initbio.norp = 'num',
initbio.n = 5,
initbio.p = NA,
initbio.dist = 'random',
readseam = F,
seam = NA,
seamdist = 'powerlaw',
seamrandp = NA,
seampla = 1,
seamplb = 0.5,
readbpam = F,
bpam = NA,
bpamdist = 'powerlaw',
bpamrandp = NA,
bpampla = 1,
bpamplb = 0.5,
readprobadoptvec = F,
probadoptvec = NA,
probadoptmean = seq(0, 1, 0.1),
probadoptsd = 0.1,
readprobestabvec = F,
probestabvec = NA,
probestabmean = 0.5,
probestabsd = 0.1,
manefffdir = 'decrease_estab',
manefffmean = 0.9,
manefffsd = 0.2,
usethreshman = T,
manefffthresh = 0.3,
sampeffort = 2
)

```

```

## [1] "ending parameter combination 1"
## [1] "ending parameter combination 2"
## [1] "ending parameter combination 3"
## [1] "ending parameter combination 4"
## [1] "ending parameter combination 5"
## [1] "ending parameter combination 6"
## [1] "ending parameter combination 7"
## [1] "ending parameter combination 8"
## [1] "ending parameter combination 9"
## [1] "ending parameter combination 10"
## [1] "ending parameter combination 11"

```

```

jt <- sens.probadoptmean$multout
plot(
  jt$probadoptmean,
  jt$mestab,
  xlab = 'Mean probability of adopting technology if informed',

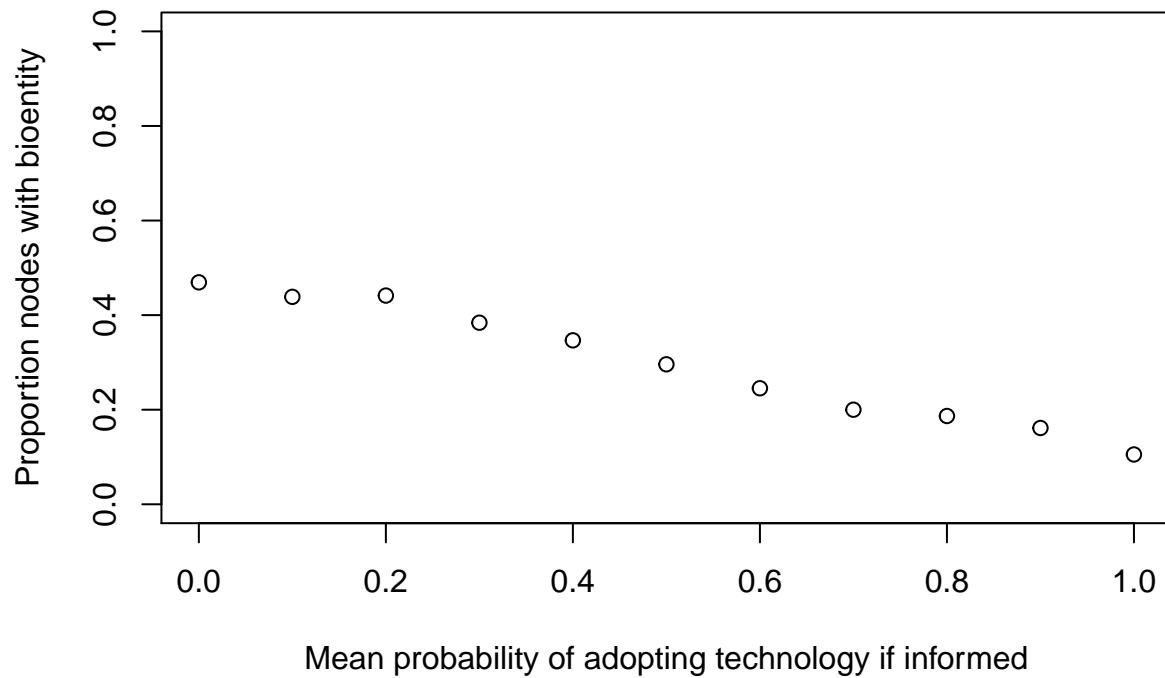
```



```

ylab = 'Proportion nodes with bioentity',
xlim = c(0, 1),
ylim = c(0, 1)
)

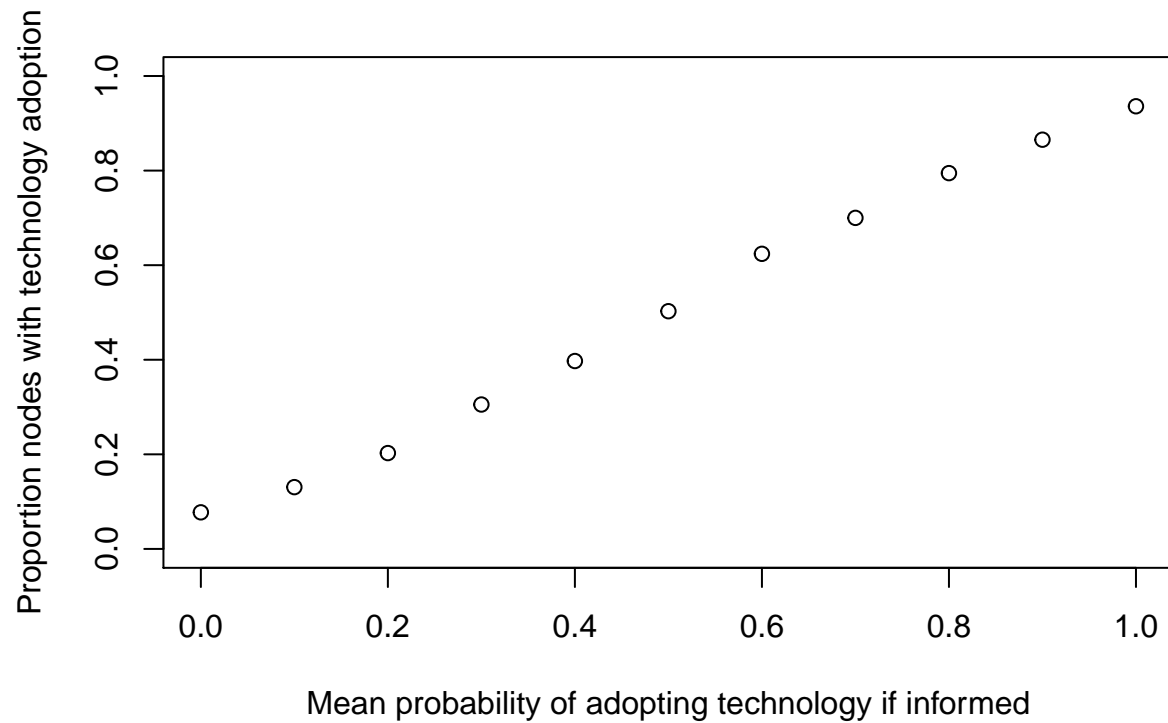
```



```

plot(
  jt$probadoptmean,
  jt$mdec,
  xlab = 'Mean probability of adopting technology if informed',
  ylab = 'Proportion nodes with technology adoption',
  xlim = c(0, 1),
  ylim = c(0, 1)
)

```



References

Csardi, G. and T. Nepusz. 2006. The igraph software package for complex network research. *InterJournal, Complex Systems*: 1695. <http://igraph.org>

Wickham, H., J. Hester, and W. Chang. devtools: Tools to Make Developing R Packages Easier. 2020. R package version 2.3.0. <https://CRAN.R-project.org/package=devtools>