

# Appendix 3: SpatialDemography Example

Alexander C. Keyel [skeyel@gmail.com](mailto:skeyel@gmail.com)

2015-11-18

## Citation

Keyel, A.C., Gerstenlauer, J.L.K. and Wiegand, K. YEAR. SpatialDemography: a new, spatially explicit, stage-structured, metacommunity model for studying plant demography. *Ecography* 000: 000-000

## Introduction

This example is designed to illustrate the basic concepts of the SpatialDemography model. This example demonstrates the importance of a model being both spatially explicit and of modeling the dynamics of multiple species simultaneously.

In this example, we consider a landscape that is a mosaic of grassland and forest habitat. We will consider three hypothetical grassland species. Species 1 is r-selected, and reproduces strongly due to a high SLA, but is short-lived and susceptible to an herbicide. Species 2 has lower resource accumulation (k-selected, low SLA) and therefore has lower reproductive rates. However, it invests more strongly in defense and is therefore longer lived and it is resistant to the herbicide. Species 3 shows intermediate reproductive potential (intermediate SLA), and is also susceptible to the herbicide. Unlike the other two species, it can survive in forest habitat. Species 1 and 2 are initially widespread in the landscape, while Species 3 is initially found only in the forested corner of the landscape (Cell 4).

We will examine the metacommunity dynamics of these three species given the initial conditions and an herbicide treatment during the 7th and 8th time steps.

## Set up paths and labels

In order to run SpatialDemography, we need to set up several variables to ensure that the code runs properly. Mainly these inputs have to do with paths and file locations.

```
# Optionally set the working directory. The working directory should default to
# the vignettes directory. If it is changed, some of the inputs from the input
# files will need to be copied into the changed directory.
setwd("C:/docs/beplants/Scripts/spatialdemography/vignettes")

# Load the spatialdemography package
library(spatialdemography)

## Loading required package: Matrix

# Set up a RunName.
# Not actually an input, but we'll use it to build some of the required inputs
```

```

run.name = "spdem_ex1"

## Set up required inputs

# scn = indicator variable to denote scenario and to extract appropriate row
# from initial.conditions.file and settings.file.
# If only one scenario is run, this should be 1, otherwise it should be updated
# inside a loop with one SpatialDemography command for each scenario.
scn = 1

# s.lbl A label used in the construction of the Species file and new Landscape files.
# Here we will leave it blank, but it may be relevant when running multiple scenarios.
s.lbl = ""

# file.ending A text string that matches the end of the species file
# (either to be generated or that already exists).
file.ending = sprintf("_%s", run.name)

# DispPath The path of the dispersal tables (if existing)
# or the path where dispersal tables should be created.
# Note that dispersal tables may take a long time to generate, so it is often
# desirable to share the same dispersal tables across multiple models.
DispPath = "dt/"

# run.path The main path for the specific overall model run.
run.path = sprintf("%s/", run.name)

# The path for model outputs.
opath = sprintf("%soutputs/", run.path)

# Create the output path
dir.create(opath, showWarnings = F, recursive = T)

# A file to be created containing the model results.
# If there is an existing file, the results will be appended to the existing file.
ResultsFile = sprintf("%sResults%s.csv", opath, file.ending)

# Code to (optionally) delete previous versions of the ResultsFile and TimeFile.
do.del = 1
if (do.del == 1){
  unlink(ResultsFile) #Delete any previous ResultsFile
}

```

## Set paths to input files

---

In this section, we'll discuss the requirements of each of the required input files separately. Note that none of the R code that reads in the files is necessary to run the SpatialDemography model - each file is read in by the SpatialDemography function.

### Initial conditions file

---

The initial conditions file specifies the initial conditions for the model. For example, how big of a landscape, how many time steps, and how many species. We will look at the file structure in greater detail below.

```
# Set the path where your inputs are stored
input.path = sprintf("%sinputs/",run.path)

# initial.conditions.file A file to specify the initial model conditions.
initial.conditions.file = sprintf("%sInitial_conditions%s.csv"
                                   ,input.path,file.ending)

# Here we will read in the file, to look at its contents
ic = read.csv(initial.conditions.file)
```

We will discuss the components of the initial conditions file in several groups:

```
knitr::kable(head(ic[1:5]))
```

Scenario	ModelName	Extent	Edge.Type	MaxTime
1	Example1	2	TORUS	12

**Scenario** indicates the scenario number

**ModelName** is a name for the scenario

**Extent** is the length of one side of a square landscape. The number of cells in the landscape is equal to the extent squared.

**Edge.Type** determines whether dispersers crossing the landscape boundary are removed from the simulation ("ABSORBING") or wrap around to the other side of the landscape ("TORUS")

**MaxTime** specifies the number of model iterations

```
knitr::kable(head(ic[6:9]))
```

K_g	multi.species.K	competitiontype	microsites
1e+05	1	2	2e+06

**K\_g** indicates the adult carrying capacity in grams

**multi.species.K** is an indicator. When 0, each species has its own carrying capacity. When 1, species share a single carrying capacity

**competitiontype** is an indicator. When 0, there is no microsite competition. When 1, each cell has a set number of microsites available to each species. When 2, all species are limited by a single total number of microsites

**microsites** This specifies the number of microsites. If competitiontype == 0, then this can be NA.

```
knitr::kable(head(ic[10:13]))
```

invasion	num.invaders	cells.to.invade	invader.repro.proportion
0	NA	NA	NA

**invasion** This setting indicates how frequently invasion will occur (every X time steps). Zero indicates that invasion will not occur.

**num.invaders** Specifies the number of species to invade the landscape every time there is an invasion event.

These species are drawn from the regional species pool, and may include species already present in the landscape.

**cells.to.invade** The number of cells in the landscape each invading species will invade.

**invader.repro.proportion** The strength at which an invader invades, expressed as a proportion of its maximum reproductive potential (e.g., 0.5 would have half the maximum reproduction, while 5 would have 5 times the normal reproduction's worth of propagules invading the landscape)

As you can see, invasion is turned off for this example.

```
knitr::kable(head(ic[14:20]))
```

num.adults	Tot.sp.rich	Land.sp.rich	Loc.sp.rich	Tot.rtd	Land.rtd	Loc.rtd
100	3	NA	NA	NA	NA	NA

**num.adults** Specifies the number of starting adults in each occupied cell. This can be overridden when reading in species' locations from a file **Tot.sp.rich** The total number of species, including species not in the landscape but that are part of the species pool (relevant for invasion)

The remaining options are set to NA here, as they relate to species generation and assignment by SpatialDemography (not relevant to the present example)

**Land.sp.rich** The number of species to assign to the landscape.

**Loc.sp.rich** The number of species to assign to each patch in the landscape.

**Tot.rtd** When assigning species with respect to response trait diversity, the total response trait diversity to have in the regional species pool.

**Land.rtd** When assigning species with respect to response trait diversity, the response trait diversity to have in the landscape.

**Loc.rtd** When assigning species with respect to response trait diversity, the response trait diversity to have in each cell of the landscape.

## Settings file

```
# settings.file A file to specify overall model settings.
settings.file = sprintf("%sSettings%s.csv",input.path,file.ending)

set = read.csv(settings.file)
knitr::kable(head(set[1:5]))
```

Scenario	GenerateVisualDebuggerData	GenerateNewSpecies	GenerateNewLandscape	Ind.lbl
1	1	0	0	scn1

**Scenario** This should correspond to the scenario number in the initial conditions file.

**GenerateVisualDebuggerData** This will generate a large amount of data related to the scenario run. Useful for understanding transient dynamics and checking the model run for errors.

**GenerateNewSpecies** An indicator for whether new species should be generated. When 1, SpatialDemography will generate new species.

**GenerateNewLandscape** An indicator for whether a new landscape should be generated (1) or read in either from file or as an R object (0).

**Ind.lbl** A label for the landscape ending (in this case scn1). This allows a user to use landscapes from previous scenario runs in later scenarios.

```
knitr::kable(head(set[6:7]))
```

loc.extraction	timestep.extraction
1	1

**loc.extraction** If a species location file is input, this setting guides the extraction of species locations, with one of three options. 1 indicates that species locations and abundances should be extracted from file. 2 indicates that only species' locations should be extracted from the file, and that abundances will be assigned by spatialdemography. 3 indicates that adult abundances and locations will be extracted from the file, but seeds and juveniles will be constructed according to the code given in the initial.n field (not relevant to this example, and not shown here)

**timestep.extraction** Optional. It indicates which time step should be used for loc.extraction (loc.extraction is designed based on the SpeciesData.csv output, consequently model outputs can be used as model inputs for later models)

```
knitr::kable(head(set[8:12]))
```

N.sim.dispersal.table	ignore.rtd	RunSimulation	RunMatrixDiagnostics	Write.Timefile
1e+06	NA	1	1	0

**N.sim.dispersal.table** The number of simulations to use when generating dispersal tables. While we recommend a large number here (e.g., 1,000,000), that may take more than one hour to run on a standard computer. Note that dispersal tables once generated can be reused.

**ignore.rtd** An indicator for whether response trait diversity should be taken into account when generating species. In this example, it has no function, as the species file already exists.

**RunSimulation** An indicator for whether or not one wishes to run the simulation. This should almost always be 1, as the simulation is the main point of the package.

**RunMatrixDiagnostics** An indicator for whether or not to create diagnostic information about the matrices used in the model. 1 Indicates that diagnostics should be run.

**Write.Timefile** A setting indicating whether model run time should be output to file. 0 indicates no, 1 indicates total scenario run time only, and 2 indicates total run time and details should be output to file.

## Environmental Layers Description File

```
# env.file A file to describe environmental layers and how they change.
env.file = sprintf("%sEnvironmental_layers%s.csv",input.path,file.ending)

ef = read.csv(env.file)
knitr::kable(head(ef[1:2]))
```

env.lbl	landscape.identifiers
landcover	L
herbicide	H

In this example, there are two environmental layers: landcover, grassland (1) vs. forest (2) and herbicide not applied (0) vs. applied (1).

**env.lbl** This is a label or descriptor for the environmental layer.

**landscape.identifier** A single letter abbreviation for the environmental layer

```
knitr::kable(head(ef[3:8]))
```

env.type	cover.levels	param1	param2	cond.lyr	cond.vals
cover	0.75;0.25	1	NA	NA	NA
NA	NA	NA	NA	NA	NA

Ignore these for this example. They pertain to landscapes created by SpatialDemography, and here we are working with an existing landscape. More information is available by typing ?env.type and ?env.file into the R console.

```
knitr::kable(head(ef[9:11]))
```

env.change.freq	env.change.type	env.change.mag
0	swap	0
1	from.file	0

**env.change.freq** The frequency of environmental change. For example, if it is 3, a given environment will change every 3 time steps. If it is 0, an environmental layer will not change. **env.change.type** The nature of the environmental change. Type ?env.change.type for a listing of the options. Here, we are reading in the environmental changes from file, so this option is set to from.file. **env.change.mag** The magnitude of the environmental change. The entries here will depend on the env.change.type, and more details can be found by typing ?env.change.type or ?env.change.mag.

## Landscape Files

Landscape files can either be generated by SpatialDemography, or can be read in from existing files. In this example, we are working with a preexisting landscape, so will look at how to read in landscape files. While these are in .csv format, SpatialDemography can also read in raster files, although at the time of this writing, support for this is rudimentary. These features may be addressed in a future vignette. In this example, we have two environmental layers: landcover and herbicides. Each file has a change step column, and then a column for every cell. Cells are numbered by row, consequently, this needs to be accounted for when displaying the landscapes in R.

```
lpath = "spdem_ex1/landscape/"
lc.file = sprintf("%slandcover_scn1.csv", lpath)
h.file = sprintf("%sherbicide_scn1.csv", lpath)

lc = read.csv(lc.file)
knitr::kable(lc)
```

ChangeStep	Cell1	Cell2	Cell3	Cell4
1	1	1	1	2

```

# Above is the basic file structure

# starts at 2 to avoid the change.step
lc2 = matrix(lc[1,2:length(lc)], nrow = 2, byrow = TRUE)
lc2

##      [,1] [,2]
## [1,] 1    1
## [2,] 1    2

# Above is the Layer in Landscape format for the first change step
# (landcover in this example only has one change step)

# We will now examine the herbicide layer:
# Please note that the change.step in the landscape files is equal to the
# timestep + 1, as the change takes effect in the NEXT model run
# Forest cells (2) are not treated with herbicides.

# Herbicides
# The herbicides layer has many change steps specified because it does not
# change predictably at a regular interval (i.e. herbicides are applied twice,
# and then the layer reverts back to no application.)
herb = read.csv(h.file)
knitr::kable(herb)

```

ChangeStep	Cell1	Cell2	Cell3	Cell4
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	1	1	1	0
8	1	1	1	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0

## Species File

```

# spfile A file specifying species traits.

```

```
# This file should be in the folder Species (it is not in the inputs folder
# because sometimes a species file is an output of SpatialDemography)
spfile = sprintf("%sSpecies/Species%s.csv",run.path, file.ending)

# sp.instr.file A file to describe how generation of species' base vital rates
# is to occur.
# Optional file only needed if SpatialDemography is generating species so we'll
# set it to NA (it can also be absent entirely)
sp.instr.file = NA

# sp.resp.instr.file A file to describe how generation of species' response
# traits is to occur
# Optional file only needed if SpatialDemography is generating species,
# so we'll set it to NA (it can also be absent entirely)
sp.resp.instr.rile = NA

## Examine the species file
spf = read.csv(spfile)
# Display first part of species file
knitr::kable(head(spf[,1:10]))
```

sp	p01	p02	p11	p12	p22	p23	p30	p32	p33
1	0.2	0	0	0.1	0	0.1	2500	0	0.10
2	0.2	0	0	0.1	0	0.1	100	0	0.95
3	0.2	0	0	0.1	0	0.1	2000	0	0.50

**sp** gives the identity of each species. Species must be numbered consecutively starting at 1.

**p01 .. p33** These correspond to the base vital rates for each species. The first number is the stage of origin, the second is the destination stage. E.g., p12 is the transition from stage 1 to stage 2. See the below LifeCycle graph for an illustration.

```
# Display middle part of species file (adult biomass and dispersal traits)
knitr::kable(head(spf[,11:14]))
```

biomass.adult	dispersalfunction	disppar1	disppar2
10	1	0.14	1
10	1	0.14	1
10	1	0.14	1

**biomass.adult** The biomass of an adult, in g. This is used with respect to the carrying capacity.

**dispersalfunction** An option to set the dispersal function. Currently the default is 1, which gives log-normal dispersal. **disppar1** The first parameter for the dispersal function. For log-normal dispersal, this is mean dispersal distance, in cells. **disppar2** The second dispersal parameter. For log-normal dispersal, this is the variance of dispersal distance, also in cells.

```
# Examine the rest of the species file
knitr::kable(head(spf[,15:ncol(spf)]))
```



L.p23	H.p12	H.p23	H.p33
1;101;1;0	0;101;1;0.01	0;101;1;0.01	0;101;1;0
1;101;1;0	0;101;1;1	0;101;1;1	0;101;1;1
1;101;1;0.8	0;101;1;0.01	0;101;1;0.01	0;101;1;0

The remaining values are response traits. They have a format of I.VR, where I = Landscape Identifier and VR = the vital rate that will respond. While there are several options for response traits (see `?response.functions` for details), the format used here has four components: The first number specifies the landscape value to respond to. E.g., 1 indicates that a 1 is the target value. The second number defines how the response type. See `?response.functions` for more options.

101 codes for an “a”/“not a” sort of coding. The species’ vital rate will be modified by the third value if it matches the environmental condition, and by the fourth value if it does not.

For example, L.p23 means that the transition from stage 2 to stage 3 responds to landcover. For species 1 and 2, when the value is 1, they have no reduction in the transition rate, but if the value is not 1 (i.e. 2), then no individuals successfully transition to stage 3 (adults). For Species 3, its transition rate is cut in half in forest habitat.

In this simple example, all species show a 50% reduction in reproduction in mowed habitat. Species 1 juveniles have 0.1 their normal transition rate in grazed habitat, while the other species have 0.9 of their normal transition rate. Finally, Species 3 does not transition seeds to juveniles, nor from juvenile to adult in the presence of herbicides and adult survival is 0 (the other species are unaffected)

## Species locations file

```
locations.file = sprintf("%slocations%.csv", input.path, file.ending)
locs = read.csv(locations.file)
knitr::kable(locs)
```

LifeStage	Species	TimeStep	Cell1	Cell2	Cell3	Cell4
Seeds	1	1	1e+06	1e+06	1e+06	0
Juveniles	1	1	1e+05	1e+05	1e+05	0
Adults	1	1	5e+03	5e+03	5e+03	0
Seeds	2	1	5e+05	5e+05	5e+05	0
Juveniles	2	1	5e+04	5e+04	5e+04	0
Adults	2	1	5e+03	5e+03	5e+03	0
Seeds	3	1	0e+00	0e+00	0e+00	50000
Juveniles	3	1	0e+00	0e+00	0e+00	5000
Adults	3	1	0e+00	0e+00	0e+00	500

**LifeStage** There should be one row for each life stage for each species for each desired time step. Options are: Seeds (stage 1), Juveniles (stage 2), and Adults (stage 3).

**Species** This should give the species number.

**TimeStep** This should indicate the timestep to which the locations correspond to. Options in settings allow

initializing the model at timesteps other than 1.

**CellX** Each cell in the landscape should have a column (in order) listing the abundances for the lifestage, species, and timestep.

## Run the model

The model is run by calling the `SpatialDemography` function with all of the required inputs.

```
# set.seed function makes the stochastic results deterministic for
# repeatability purposes. However, it is not actually relevant for this
# example, as there are no stochastic processes included.
set.seed(567891234)

# Run SpatialDemography. This is the main command to run the model,
# and uses the inputs defined above.
SpatialDemography(scn,s.lbl,file.ending,DispPath,run.path,opath,
  ResultsFile,initial.conditions.file,settings.file,env.file,
  spfile,sp.instr.file,sp.resp.instr.file,lpath,locations.file)

## Loading required package: multirich

##   Scenario Scenario.Number   Scale Timepoint LastTime Sp.Rich   Beta.Div
## 1 Example1                1 landscape   Initial      12      3  1.7142857
## 2 Example1                1 landscape    Final      12      3  1.2000000
## 3 Example1                1 landscape   Change      12      0 -0.5142857
##      Biomass FTD.UTC RTD.UTC
## 1 305000.00      3      3
## 2 279315.09      3      3
## 3 -25684.91      0      0
```

## Examine model outputs

Now we will examine the various outputs generated by `spatialdemography`.

```
# Results file
out.results = read.csv(ResultsFile)
knitr::kable(out.results)
```

Scenario	Scenario.Number	Scale	Timepoint	LastTime	Sp.Rich	Beta.Div	Biomass	FTD.UTC	RTD.UTC
Example1	1	landscape	Initial	12	3	1.7142857	305000.00	3	3
Example1	1	landscape	Final	12	3	1.2000000	279315.09	3	3
Example1	1	landscape	Change	12	0	-0.5142857	-25684.91	0	0

The above results file summarizes the model's initial state, final state, and change in state.

**Scale** Label for what portion of the landscape is considered. In this case, the entire landscape was examined. It is possible to consider a subset of the landscape to be summarized (e.g., to exclude large patches of unsuitable

habitat from the calculations)

**LastTime** is an indicator that the model ran to completion, and should match the MaxTime option in the initial.conditions.file

**Sp.Rich** indicates species richness

**Beta.Div** indicates multiplicative beta diversity

**Biomass** indicates the total biomass present in the landscape

**FTD.MVR** indicates the functional trait richness based on Unique Trait Combinations. Both vital rates and response traits are included here.

**RTD.MVR** indicates the response trait richness. Only dispersal traits and response traits are considered here.

```
# Next, we will consider the scenario-specific files.
```

```
epath = "spdem_ex1/Example1/"
```

```
# SpeciesData.csv will contain species abundances of each Lifestage
```

```
# (and all lifestages together) for every cell in the landscape.
```

```
sp.dat = read.csv(sprintf("%sSpeciesData.csv", epath))
```

```
knitr::kable(head(sp.dat))
```

LifeStage	Species	TimeStep	Cell1	Cell2	Cell3	Cell4
Seeds	1	1	384195.332	383576.781	383592.986	203705.0
Juveniles	1	1	100000.000	100000.000	100000.000	0.0
Adults	1	1	5185.185	5185.185	5185.185	0.0
ALL	1	1	489380.517	488761.966	488778.171	203705.0
Seeds	2	1	15367.813	15343.071	15343.719	8148.2
Juveniles	2	1	50000.000	50000.000	50000.000	0.0

```
# These data are summarized in the following files (in the Example1 folder):
```

```
# Cells_occupied.csv indicates how many cells are occupied by each species at  
# each timestep.
```

```
# SpeciesStats_Adults_v2.csv indicates how many adults of each species there is  
# in the landscape for each timestep, and also the log of Lambda (growth rate).
```

```
# SpeciesStats_ALL_v2.csv is the same as SpeciesStats_Adults_v2.csv, except for  
# all life stages pooled.
```

```
# SpeciesStats_Juvs.csv indicates how many juveniles there are of each species  
# in the landscape, but does not include loglambda values
```

```
# SpeciesStats_Seeds.csv as with juveniles, but for seeds
```

```
# One other file in this folder bears note: change_count_lookup.csv provides an  
# index for which change steps correspond to which timesteps (because a change  
# in environmental conditions may be less frequent than a change in timesteps).
```

```
# Finally, we will consider the diagnostic files.
```

```
dpath = sprintf("%sDiagnostics/", epath)
```

```
# There are four files in this folder:
```

```
# AMatrices.csv contains the entire matrix A for each species for each timestep.
```

```
# This can be useful for checking that vital rates are as intended and that the
# matrix was properly constructed.
```

```
# TransitionMatrices.csv gives the transition matrices within a cell for each
# cell in the landscape, and can be useful for examining local dynamics
# exclusive of immigration and emigration.
```

```
# AMatricesSummaries.csv summarizes the matrix content for each cell
```

```
# Broken into two sections to improve readability
```

```
mat.sum = read.csv(sprintf("%sAMatricesSummaries.csv",dpath))
```

```
knitr::kable(head(mat.sum[,1:5]))
```

ChangeStep	Species	Cell	Dominant.EigenValue	Per.Capita.Seeds.Exported
1	1	1	1.696044	40.74100
1	1	2	1.696044	40.74100
1	1	3	1.696044	40.74100
1	1	4	0.100000	40.74100
1	2	1	1.101390	1.62964
1	2	2	1.101390	1.62964

```
knitr::kable(head(mat.sum[,6:ncol(mat.sum)]))
```

Per.Capita.Seeds.Imported	Prop.seeds.exported
40.74100	0.0815059
40.74100	0.0815059
40.74100	0.0815059
40.74100	0.0815059
1.62964	0.0815059
1.62964	0.0815059

```
# MatrixDiagnostics.csv summarizes the matrices across all cells for a species
# for each change step.
```

```
# Broken into 3 parts to improve readability
```

```
mat.dia = read.csv(sprintf("%sMatrixDiagnostics.csv",dpath))
```

```
knitr::kable(head(mat.dia[,1:7]))
```

ChangeStep	Species	A.Eigen	Is.Irreducible	Is.Ergodic	Max.Local.Eigen	Median.Local.Eigen
1	1	1.728706	0	0	1.696044	1.696044
1	2	1.108456	0	0	1.101390	1.101390
1	3	1.762370	1	1	1.728845	1.728845
2	1	1.728706	0	0	1.696044	1.696044

ChangeStep	Species	A.Eigen	Is.Irreducible	Is.Ergodic	Max.Local.Eigen	Median.Local.Eigen
2	2	1.108456	0	0	1.101390	1.101390
2	3	1.762370	1	1	1.728845	1.728845

```
knitr::kable(head(mat.dia[,8:11]))
```

Min.Local.Eigen	Max.Seed.Export	Median.Seed.Export	Min.Seed.Export
0.100000	40.74100	40.74100	40.74100
0.950000	1.62964	1.62964	1.62964
1.619839	32.59280	32.59280	32.59280
0.100000	40.74100	40.74100	40.74100
0.950000	1.62964	1.62964	1.62964
1.619839	32.59280	32.59280	32.59280

```
knitr::kable(head(mat.dia[,11:ncol(mat.dia)]))
```

Min.Seed.Export	Max.Seed.Import	Median.Seed.Import	Min.Seed.Import	Prop.Dispersing
40.74100	40.74100	40.74100	40.74100	0.0815059
1.62964	1.62964	1.62964	1.62964	0.0815059
32.59280	32.59280	32.59280	32.59280	0.0815059
40.74100	40.74100	40.74100	40.74100	0.0815059
1.62964	1.62964	1.62964	1.62964	0.0815059
32.59280	32.59280	32.59280	32.59280	0.0815059

## Model Interpretation

In this section, we will take a closer look at some of the model outputs, and put them in the context of the ecology of our species.

```
# We'll start by looking at the SpeciesData.csv a bit more closely.
# Specifically, we'll look at the adult numbers for each species (rounded for
# ease of interpretation)
sp.dat.ad = sp.dat[sp.dat$LifeStage == "Adults", ]
sp.dat.ad$LifeStage = NULL
#sp.dat.ad = sp.dat.ad[order(sp.dat.ad$Species), ]
sp.dat.ad1 = sp.dat.ad[sp.dat.ad$Species == 1, ]
sp.dat.ad2 = sp.dat.ad[sp.dat.ad$Species == 2, ]
sp.dat.ad3 = sp.dat.ad[sp.dat.ad$Species == 3, ]

knitr::kable(round(sp.dat.ad1,0)) #Examine Species 1 Adult data
```

	Species	TimeStep	Cell1	Cell2	Cell3	Cell4
3	1	1	5185	5185	5185	0
15	1	2	5235	5235	5235	0
27	1	3	4365	4359	4359	0
39	1	4	4288	4280	4280	0
51	1	5	4282	4273	4273	0
63	1	6	4243	4213	4213	0
75	1	7	38	38	38	0
87	1	8	0	0	0	0
99	1	9	38	37	37	0
111	1	10	191	185	186	0
123	1	11	21	20	20	0
135	1	12	188	181	181	0

Looking at the data for Species 1, there was a slight initial decline, because the model was initialized slightly above carrying capacity. Species 1 declined dramatically due to the herbicide application, and recovered slowly.

```
knitr::kable(round(sp.dat.ad2,0)) #Examine Species 2 Adult data
```

	Species	TimeStep	Cell1	Cell2	Cell3	Cell4
7	2	1	4815	4815	4815	0
19	2	2	4765	4765	4765	0
31	2	3	4680	4680	4680	0
43	2	4	4589	4589	4589	0
55	2	5	4500	4499	4499	0
67	2	6	4439	4437	4437	0
79	2	7	4380	4377	4377	0
91	2	8	4321	4316	4316	0
103	2	9	4263	4256	4256	0
115	2	10	4913	4888	4889	0
127	2	11	5519	5477	5478	0
139	2	12	6084	6026	6027	0

When the populations of Species 1 & 3 were reduced by the herbicide, Species 2 increased due to the increased availability of habitat. Due to its slower growth rate, its per-capita increase in the landscape was not as great as that of Species 3. However, it showed the greatest absolute numerical increase due to its greater overall population size.

```
knitr::kable(round(sp.dat.ad3,0)) #Examine Species 3 Adult data
```

	Species	TimeStep	Cell1	Cell2	Cell3	Cell4
11	3	1	0	0	0	650
23	3	2	0	0	0	725
35	3	3	4	11	11	1832
47	3	4	8	19	19	2584
59	3	5	10	24	24	3041
71	3	6	27	64	63	4030
83	3	7	0	1	1	4700
95	3	8	0	0	0	5103
107	3	9	1	1	1	5407
119	3	10	265	643	633	5887
131	3	11	419	1015	1000	6132
143	3	12	515	1248	1229	6252

In the beginning of the simulation, Species 3 had only been able to weakly invade the grassland habitat, despite the grassland habitat being more suitable. This was due to competition from Species 1 and Species 2 because the system was at carrying capacity. Once the herbicides were applied, there was an empty niche that Species 3 was able to fill. Species 3 responded more rapidly than Species 1 because of the source population in the forest.

```
# Graphically examine the overall patterns by species
# Read in SpeciesStats_Adults_v2.csv
ssa = "spdem_ex1/Example1/SpeciesStats_Adults_v2.csv"
my.df = read.csv(ssa)

x.limit = 15
y.limit = 22500

# Plot Species 1
plot(my.df$RunTime, my.df$Sp1,
     xlab = "Time Step", ylab = "Number of Adults in Landscape",
     xlim = c(0,x.limit), ylim = c(0, y.limit), pch = 15, col = 4)

par(new = T) #Tell R to add next plot to existing plot

# Plot Species 2
plot(my.df$RunTime, my.df$Sp2, xlim = c(0,x.limit), ylim = c(0, y.limit),
     pch = 16, col = 2, xlab = "", ylab = "", xaxt = 'n', yaxt = 'n')

par(new = T) #Tell R to add next plot to existing plot

# Plot Species 3
plot(my.df$RunTime, my.df$Sp3, xlim = c(0,x.limit), ylim = c(0, y.limit),
```

```

pch = 17, col = 3, xlab = "", ylab = "", xaxt = 'n', yaxt = 'n')

#Legend(0,y.limit, c('Species 1','Species 2','Species 3'), pch = c(15,16,17),
#col = c(4,2,3), cex = 1.5)

## Examine Species 3's performance by cell
x.limit = 15
y.limit = log(7000,10)

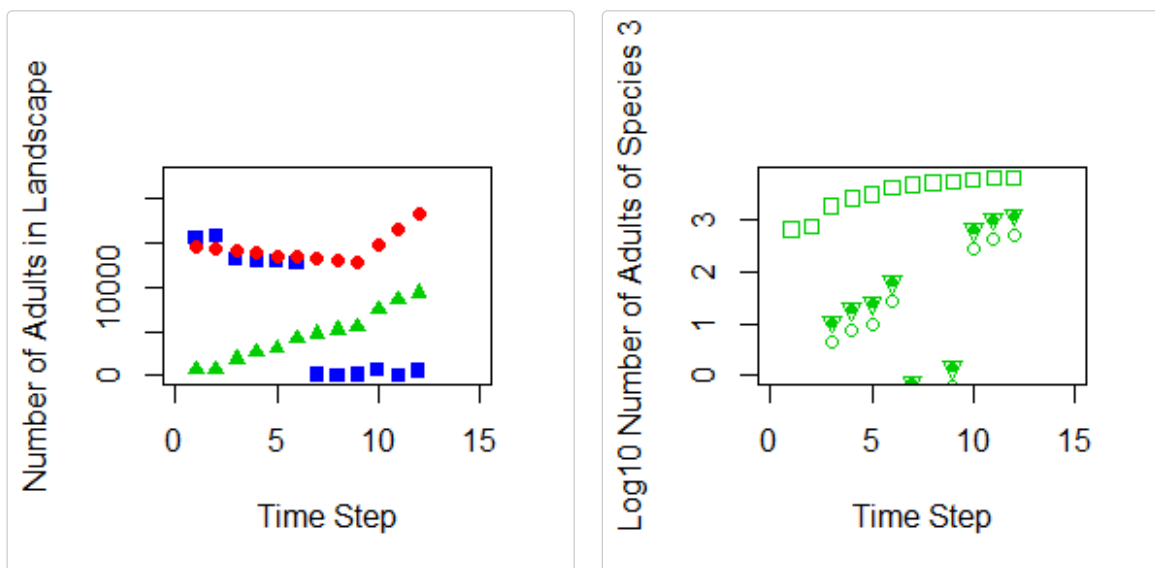
# One point type for each cell
pch.vec = c(1, 6,20,0)
# Same color for each cell to make it clear it is Species 3
col.vec = c(3,3,3,3)

# Plot Cell 1
plot(sp.dat.ad3$TimeStep,log(sp.dat.ad3$Cell1,10),
     xlim = c(0,x.limit), ylim = c(0,y.limit),
     pch = pch.vec[1], col = col.vec[1],
     xlab = "Time Step", ylab = "Log10 Number of Adults of Species 3" )

# Plot Cells 2 - 4
for (i in 2:4){
  par(new = T) #Tell R to add next plot to existing plot
  this.cell = sprintf("Cell%s", i)
  plot(sp.dat.ad3$TimeStep, log(sp.dat.ad3[[this.cell]],10),
       xlim = c(0,x.limit), ylim = c(0, y.limit),
       pch = pch.vec[i], col = col.vec[i], xlab = "", ylab = "",
       xaxt = 'n', yaxt = 'n')
}

#Legend(11.7,1.2,c('Cell 1','Cell 2','Cell 3','Cell 4'),
#pch = pch.vec, col = col.vec, cex = 1.5 )

```



The first plot shows the patterns described above. (Species 1 = blue squares, Species 2 = red circles, and



Species 3 = green triangles). The second plot shows the importance of space. The abundance in each cell of the landscape is plotted for Species 3. Cell 4 (open squares) serves as a reference - this is the population unaffected by herbicides. Following the herbicide treatment, the other three Cells all increased, but Cells 1 and 3 (inverted triangles and filled circles) increased faster because they received more propagules due to their greater proximity to Cell 4. Cell 2 (open circles) increased the slowest.