M. Calvescens Invasion Model

David Lewis

Jan 25, 2021

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

#Dataset Import

```
library(readr)
Miconia_distance_data <- read_csv("Lewis_R0_data.csv")
  > Rows: 29807 Columns: 1
  > -- Column specification --------------------------------------------------------
  > Delimiter: ","
  > dbl (1): Dist
  >
  > i Use `spec()` to retrieve the full column specification for this data.
  > i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

##Dispersal Kernel {.tabset}

Dispersal distance, the Euclidian distance between 'start' and 'end' points of a dispersal event, is recognized as a fundamental characteristic of the dispersal process, defined here as the movement of seeds from their natal site (adult tree) to a new place of potential establishment (Natan et al. 2015). The statistical distribution of dispersal distances in a population is termed the 'dispersal kernel'. It is a probability density function (pdf) describing the distribution of the post-dispersal locations relatively to the source point (Natan et al. 2015, Mollison 1977).

###PDF

The probability density function for this model is based on the data collected from miconia management actions between 1991 and 2016 by the National Park Service and Maui Invasive Species Council. The distances in the dataset range from 0 meters to 1608 meters.

```
  > Rows: 29807 Columns: 1
  > -- Column specification --------------------------------------------------------
  > Delimiter: ","
  > dbl (1): Dist
  >
  > i Use `spec()` to retrieve the full column specification for this data.
  > i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

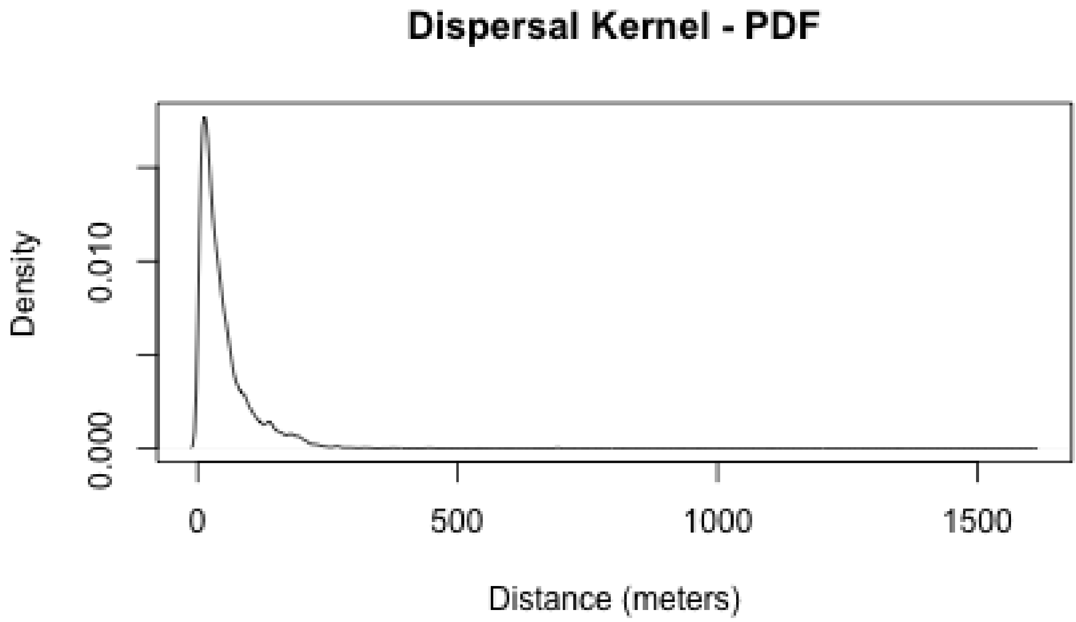


Figure 1: Figure 1. The probabikity density function that shows the likilhood of how far a seed with disperse (in meters) from its origin. Ranges from 0 to 1608 meters

###Histogram

#Models

#Population Model

I built out a simplified stage matrix model to project population dynamics of miconia over a 20 year period in 1 year intervals in an illustrative uninvaded environment. The purpose of this model is generate a population of adult trees that occur over the 20 year period from one initial adult tree. The 3 by 3 matrix consists of only 5 basic life history traits as parameters:

1. Germination rate and seedling survival
2. Adult fecundity
3. Juvenile survivorship
4. Juveniles that progress to adulthood
5. Adult surivorship

It is important to note that this simplified model assumes that density dependency, carrying capacity, and by extension Allee effects have little siginificance in simplified individual population models with short time periods. Therefore they have been omitted from the current model. However, this model is simple enough to be adapted to include these factors.

###Model Parameters

```
# Establish model parameters

G <- 0.000055                # Germination rate and seedling survival
Fa <- 36855                  # Fecundity of adults
Pj1 <- 0.85                  # Juvenile survivorship
Pj2 <- 0.15                  # Juvenile survivorship to adults
Ma <- 0.90                   # Adult survivorship
```

###Life Stage Matrix Model

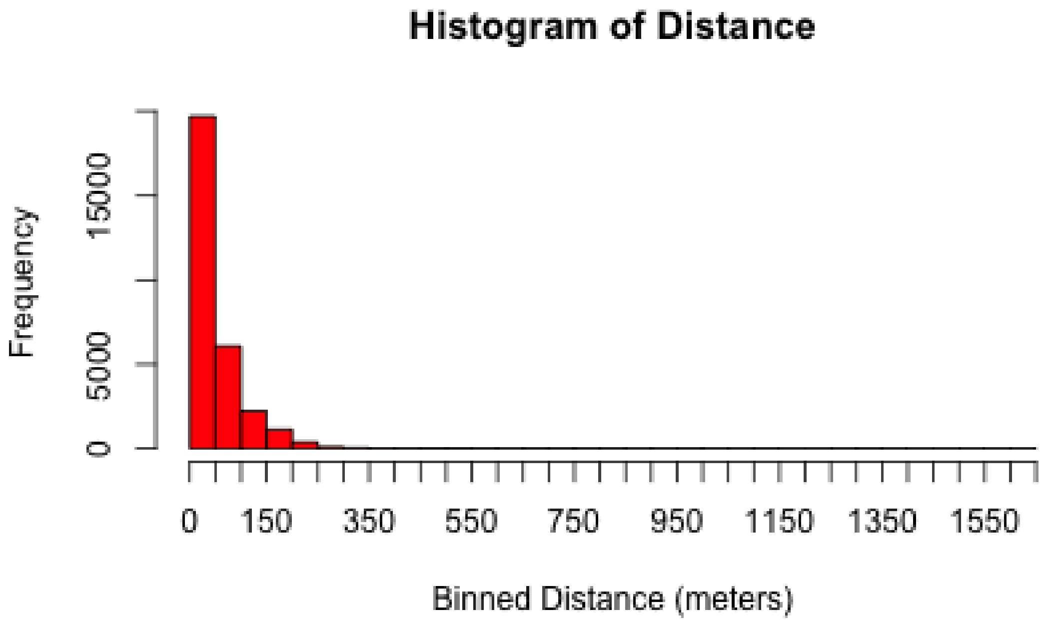


Figure 2: Figure 2. Histogram showing frequency at which distances seeds may disperse to between 0 and 1608 meters.

```
# Built simple 3 x 3 life stage matrix

projection_matrix <- matrix(
  c(0,      0,           Fa,
    G,      Pj1,         0,
    0,      Pj2,         Ma )
  , nrow=3, byrow=TRUE
)

projection_matrix
  >          [,1] [,2]    [,3]
  > [1,] 0.0e+00 0.00 36855.0
  > [2,] 5.5e-05 0.85     0.0
  > [3,] 0.0e+00 0.15     0.9
```

The package 'popbio' was used to compute the population growth rate. Miconia has a growth rate of 1.3 ~ 1.45 in tropical climates (Hester, Leary).

###Growth Rate

The package 'popbio' was used to compute the population growth rate. Miconia has a growth rate of 1.3 ~ 1.45 in tropical climates (Hester, Leary).

```
# 'popbio' package to compute lambda (population growth rate).
# lambda < 1, population decreasing
# lambda = 1, population stable
# lamdba > 1, population increasing

lambda(projection_matrix)
  > [1] 1.35
```

###Set Initial Abundance

Table 1: Table 1. The population of each life stage by year.

	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Year 7	Year 8	Year 9	Year 10	Year 11	Year 12
Seeds	0	36855	33170	29853	38073	53876	74234	100271	135066	182229	246064	332273	448647
Juveniles	0	0	2	4	5	6	8	11	15	20	27	37	49
Adults	1	1	1	1	1	2	3	4	5	7	9	12	16

Initial abundance is set at 1 adult tree.

```
# Initial abundance for each life stage (Seeds, juveniles, adults).

Abundance_year0 <- (matrix(c(0, 0, 1), ncol=1))
Abundance_year0
  >      [,1]
  > [1,]    0
  > [2,]    0
  > [3,]    1
```

##Individual Population Demographics Projected Over 20 Year Period

```
# Built loop to project out to 20 year time horizon

nYears <- 20                                          # set the number of years to project
TimeMat <- projection_matrix                          # define the proj matrix as time matrix
InitAbund <- Abundance_year0                          # define the initial abundance ay yr 0

allYears <- matrix(0,nrow=nrow(TimeMat),ncol=nYears+1)  # build a storage array for all abundances
rownames(allYears) <- c("Seeds", "Juveniles", "Adults") # establish row names
colnames(allYears) <- paste("Year", 0:20)             # establish column names
allYears[,1] <- InitAbund                             # set the year 0 abundance
for(t in 2:(nYears+1)){                               # loop through all years
  allYears[,t] <-  TimeMat %*% allYears[,t-1]
}

library(kableExtra)
  >
  > Attaching package: 'kableExtra'
  > The following object is masked from 'package:dplyr':
  >
  >     group_rows
kable(allYears, digits = 0,  caption = "Table 1. The population of each life stage by year.") %>%
  kable_styling("striped", full_width = F) %>%
  row_spec(0, angle = -90) %>%
  scroll_box(width = "750px", height = "225px")
```

#Spread & Management by Generation {.tabset}

The second piece of the puzzle is the spatial component. Now that we know what the population demographics of an individual adult tree look like over a 20 year period, we can find out where the population occurs relative to the maternal source. Below is a model that functions in four steps at each time step:

1. Draw a number of random variates N from the dispersal kernel equivalent to the adult population in that time step

2. Draw a uniform random number 0 between 0 and 2pi, which is used to choose a direction for dispersal.
3. Calculate the relative location of the dispersed individuals using trig functions: x = cos(0) d y = sin(0) d
4. Store the new location coordinates (x, y) in a list that is sorted by time step.

The spatial-temporal model is applied at each year, with offspring maturing into adults at 5 years.This creates a new population (Generation 1) with new adult trees that occur starting in Year 5. As a result, Gen 3 offspring that mature into adult show up in the model between years 15 to 20 (see timeline).

##Timeline

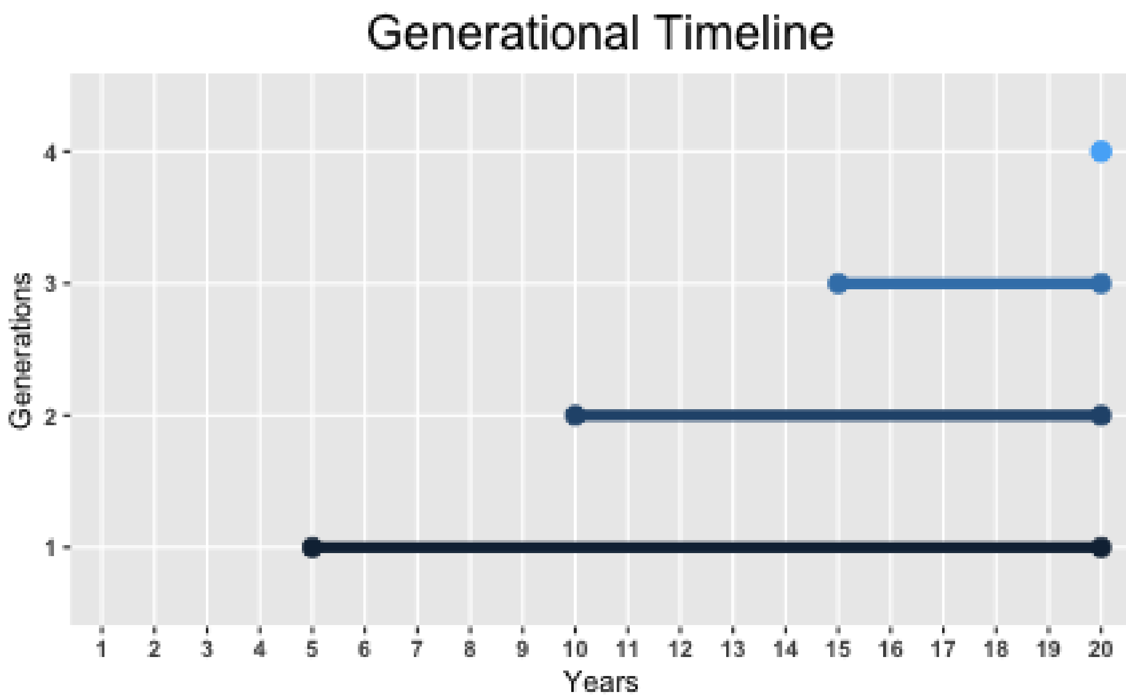


Figure 3: Figure 5. Timeline of each generation of new adult trees that will occur over the 20 year time horizon.

##Generation 1 Generation 1 are the offspring of the intial offspring in the population model. The spatial-temporal model is applied at each year, with offspring maturing into adults after 4 years. As a result, Gen 1 adult trees show up in the model between years 4 to 20 (see timeline).

```
set.seed(29807)                                                  # Sets random number generation to static f
allYears <- unname(allYears)                                     # Turns the population into a table with no
Adults <- round(allYears[3,])                                    # Adult tree population years 1:20

NewPop <- list()                                                 # Creates an empty list for coordinate stor
Spread <- function(t, yr, gen) {
  NewPop <- list()
  Time <- t
  year <- yr
  Gen <- gen
  for (i in Time:6) {                                            # for loop that iterates between over 20
    N <- sample(Distance, size = Adults[i], replace = FALSE)    # draw number of random variate from dist
    theta <- runif(N,0,2*pi)                                     # distance is assigned polar coordinates
    NewPop[[i]] <- data.frame(x=cos(theta)*N, y=sin(theta)*N) # assign variate a coordinate x, y and st
    NewPop[[i]]$Year <- i + year
```

```
    NewPop[[i]]$Generation <- Gen
    rbind(NewPop[[i]])
  }
#Bind yearly populations into one data frame
NewPop <- do.call(rbind, NewPop)
return(NewPop)
}


Gen1 <- list(Spread(21, -1, 1))
Gen1 <- do.call(rbind, Gen1)
row_sub = apply(Gen1, 1, function(row) all(row !=0 ))
Gen1 <- Gen1[row_sub,]
```

Management

Asset Protection

```
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen1
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen1 <- gen

#create function apply management within protected area
Exclude <- function(gen, AP, yr){
  Exclusion <- filter(gen, radius > AP, Year %in% yr)
  return(Exclusion)
}

#storage for management results
Asset.Protec <- Exclude(Gen1, AP, Years)
```

Containment Strategy

```
C1 <- 80     #containment boundary radius set
AP <- 550
Years <- c(1:20)    #years where management will occur

#create function apply management
Contain <- function(gen, C1, AP, yr){
  Containment <- filter(gen, radius > C1 & radius < AP, Year %in% yr)
  return(Containment)
}

#storage for management results
Containment <- Contain(Gen1, C1, AP, Years)
```

```r
Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Total.Remaining <- nrow(Gen1)
```

##Generation 2 {.tabset}

Generation 2 are the new adult trees that are the offspring of Generation 1 trees. The spatial-temporal model is applied at each year, with offspring maturing into adults after 4 years. As a result, Gen 2 offspring that mature into adults show up in the model between years 8 to 20 (see timeline).

###Year 5

```r
# Generation 2

# Year 5 New Population Growth

#function to count number of new populations to be simulated
nreps <- function(gen, yr){
  total <- nrow(filter(gen, Year == yr))
  return(total)
}

Gen2.Yr5 <- replicate(nreps(Gen1, 5), list(Spread(16, 4, 2)))
Gen2.Yr5 <- do.call(rbind, Gen2.Yr5)
row_sub = apply(Gen2.Yr5, 1, function(row) all(row !=0 ))
Gen2.Yr5 <- Gen2.Yr5[row_sub,]
orig.shft <- filter(Gen1, Year == 5)
Gen2.Yr5 <- Gen2.Yr5 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```r
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen2.Yr5
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen2.Yr5 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen2.Yr5, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```r
C1 <- 80      #containment boundary set
AP <- 550
```

```
Years <- c(1:20)    #years where management will occur

#storage for management results
Containment2 <- Contain(Gen2.Yr5, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen2 <- Gen2.Yr5
Total.Pop <- rbind(Gen1, Gen2)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 6

```
# Generation 2

# Year 6 New Population Growth

Gen2.Yr6 <- replicate(nreps(Gen1, 6), list(Spread(15, 5, 2)))
Gen2.Yr6 <- do.call(rbind, Gen2.Yr6)
row_sub = apply(Gen2.Yr6, 1, function(row) all(row !=0 ))
Gen2.Yr6 <- Gen2.Yr6[row_sub,]
orig.shft <- filter(Gen1, Year == 6)
Gen2.Yr6 <- Gen2.Yr6 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen2.Yr6
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen2.Yr6 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen2.Yr6, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```
C1 <- 80      #containment boundary set
AP <- 550
Years <- c(1:20)    #years where management will occur
```

```
#storage for management results
Containment2 <- Contain(Gen2.Yr6, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen2 <- rbind(Gen2, Gen2.Yr6)
Total.Pop <- rbind(Total.Pop, Gen2.Yr6)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 7

```
# Generation 2
# Year 7 New Population Growth
# Time steps at 13

Gen2.Yr7 <- replicate(nreps(Gen1, 7), list(Spread(14, 6, 2)))
Gen2.Yr7 <- do.call(rbind, Gen2.Yr7)
row_sub = apply(Gen2.Yr7, 1, function(row) all(row !=0 ))
Gen2.Yr7 <- Gen2.Yr7[row_sub,]
orig.shft <- filter(Gen1, Year == 7)
Gen2.Yr7 <- Gen2.Yr7 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen2.Yr7
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen2.Yr7 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen2.Yr7, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```
C1 <- 80      #containment boundary set
AP <- 550
Years <- c(1:20)     #years where management will occur

#storage for management results
```

```
Containment2 <- Contain(Gen2.Yr7, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen2 <- rbind(Gen2, Gen2.Yr7)
Total.Pop <- rbind(Total.Pop, Gen2.Yr7)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 8

```
# Generation 2
# Year 8 New Population Growth

Gen2.Yr8 <- replicate(nreps(Gen1, 8), list(Spread(13, 7, 2)))
Gen2.Yr8 <- do.call(rbind, Gen2.Yr8)
row_sub = apply(Gen2.Yr8, 1, function(row) all(row !=0 ))
Gen2.Yr8 <- Gen2.Yr8[row_sub,]
orig.shft <- filter(Gen1, Year == 8)
Gen2.Yr8 <- Gen2.Yr8 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen2.Yr8
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen2.Yr8 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen2.Yr8, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```
C1 <- 80     #containment boundary set
AP <- 550
Years <- c(1:20)    #years where management will occur

#storage for management results
Containment2 <- Contain(Gen2.Yr8, C1, AP, Years)
```

```r
#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen2 <- rbind(Gen2, Gen2.Yr8)
Total.Pop <- rbind(Total.Pop, Gen2.Yr8)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 9

```r
# Generation 2
# Year 9 New Population Growth

Gen2.Yr9 <- replicate(nreps(Gen1, 9), list(Spread(12, 8, 2)))
Gen2.Yr9 <- do.call(rbind, Gen2.Yr9)
row_sub = apply(Gen2.Yr9, 1, function(row) all(row !=0 ))
Gen2.Yr9 <- Gen2.Yr9[row_sub,]
orig.shft <- filter(Gen1, Year == 9)
Gen2.Yr9 <- Gen2.Yr9 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```r
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen2.Yr9
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen2.Yr9 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen2.Yr9, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```r
C1 <- 80     #containment boundary set
AP <- 550
Years <- c(1:20)    #years where management will occur

#storage for management results
Containment2 <- Contain(Gen2.Yr9, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)
```

```r
Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen2 <- rbind(Gen2, Gen2.Yr9)
Total.Pop <- rbind(Total.Pop, Gen2.Yr9)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 10

```r
# Generation 2
# Year 10 New Population Growth

Gen2.Yr10 <- replicate(nreps(Gen1, 10), list(Spread(11, 9, 2)))
Gen2.Yr10 <- do.call(rbind, Gen2.Yr10)
row_sub = apply(Gen2.Yr10, 1, function(row) all(row !=0 ))
Gen2.Yr10 <- Gen2.Yr10[row_sub,]
orig.shft <- filter(Gen1, Year == 10)
Gen2.Yr10 <- Gen2.Yr10 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```r
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen2.Yr10
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen2.Yr10 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen2.Yr10, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```r
C1 <- 80      #containment boundary set
AP <- 550
Years <- c(1:20)     #years where management will occur

#storage for management results
Containment2 <- Contain(Gen2.Yr10, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)
```

```r
Gen2 <- rbind(Gen2, Gen2.Yr10)
Total.Pop <- rbind(Total.Pop, Gen2.Yr10)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 11

```r
# Generation 2
# Year 11 New Population Growth

Gen2.Yr11 <- replicate(nreps(Gen1, 11), list(Spread(10, 10, 2)))
Gen2.Yr11 <- do.call(rbind, Gen2.Yr11)
row_sub = apply(Gen2.Yr11, 1, function(row) all(row !=0 ))
Gen2.Yr11 <- Gen2.Yr11[row_sub,]
orig.shft <- filter(Gen1, Year == 11)
Gen2.Yr11 <- Gen2.Yr11 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```r
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen2.Yr11
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen2.Yr11 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen2.Yr11, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```r
C1 <- 80     #containment boundary set
AP <- 550
Years <- c(1:20)     #years where management will occur

#storage for management results
Containment2 <- Contain(Gen2.Yr11, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen2 <- rbind(Gen2, Gen2.Yr11)
Total.Pop <- rbind(Total.Pop, Gen2.Yr11)
```

```
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 12

```
# Generation 2
# Year 12 New Population Growth
# Time steps at 9

Gen2.Yr12 <- replicate(nreps(Gen1, 12), list(Spread(9, 11, 2)))
Gen2.Yr12 <- do.call(rbind, Gen2.Yr12)
row_sub = apply(Gen2.Yr12, 1, function(row) all(row !=0 ))
Gen2.Yr12 <- Gen2.Yr12[row_sub,]
orig.shft <- filter(Gen1, Year == 12)
Gen2.Yr12 <- Gen2.Yr12 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen2.Yr12
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen2.Yr12 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen2.Yr12, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```
C1 <- 80      #containment boundary set
AP <- 550
Years <- c(1:20)     #years where management will occur

#storage for management results
Containment2 <- Contain(Gen2.Yr12, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen2 <- rbind(Gen2, Gen2.Yr12)
Total.Pop <- rbind(Total.Pop, Gen2.Yr12)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 13

```
# Generation 2
# Year 13 New Population Growth
# Time steps at 8

Gen2.Yr13 <- replicate(nreps(Gen1, 13), list(Spread(8, 12, 2)))
Gen2.Yr13 <- do.call(rbind, Gen2.Yr13)
row_sub = apply(Gen2.Yr13, 1, function(row) all(row !=0 ))
Gen2.Yr13 <- Gen2.Yr13[row_sub,]
orig.shft <- filter(Gen1, Year == 13)
Gen2.Yr13 <- Gen2.Yr13 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen2.Yr13
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen2.Yr13 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen2.Yr13, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```
C1 <- 80     #containment boundary set
AP <- 550
Years <- c(1:20)     #years where management will occur

#storage for management results
Containment2 <- Contain(Gen2.Yr13, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen2 <- rbind(Gen2, Gen2.Yr13)
Total.Pop <- rbind(Total.Pop, Gen2.Yr13)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 14

```
# Generation 2
# Year 14 New Population Growth
# Time steps at 7

Gen2.Yr14 <- replicate(nreps(Gen1, 14), list(Spread(7, 13, 2)))
Gen2.Yr14 <- do.call(rbind, Gen2.Yr14)
row_sub = apply(Gen2.Yr14, 1, function(row) all(row !=0 ))
Gen2.Yr14 <- Gen2.Yr14[row_sub,]
orig.shft <- filter(Gen1, Year == 14)
Gen2.Yr14 <- Gen2.Yr14 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen2.Yr14
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen2.Yr14 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen2.Yr14, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```
C1 <- 80     #containment boundary set
AP <- 550
Years <- c(1:20)    #years where management will occur

#storage for management results
Containment2 <- Contain(Gen2.Yr14, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen2 <- rbind(Gen2, Gen2.Yr14)
Total.Pop <- rbind(Total.Pop, Gen2.Yr14)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 15

```r
# Generation 2
# Year 15 New Population Growth
# Time steps at 6

Gen2.Yr15 <- replicate(nreps(Gen1, 15), list(Spread(6, 14, 2)))
Gen2.Yr15 <- do.call(rbind, Gen2.Yr15)
row_sub = apply(Gen2.Yr15, 1, function(row) all(row !=0 ))
Gen2.Yr15 <- Gen2.Yr15[row_sub,]
orig.shft <- filter(Gen1, Year == 15)
Gen2.Yr15 <- Gen2.Yr15 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```r
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen2.Yr15
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen2.Yr15 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen2.Yr15, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```r
C1 <- 80     #containment boundary set
AP <- 550
Years <- c(1:20)    #years where management will occur

#storage for management results
Containment2 <- Contain(Gen2.Yr15, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen2 <- rbind(Gen2, Gen2.Yr15)
Total.Pop <- rbind(Total.Pop, Gen2.Yr15)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

##Generation 3 {.tabset}

Generation 3 are the new adult trees that are the offspring of Generation 2 trees. The spatial-temporal model is applied at each year, with offspring maturing into adults after 4 years. As a result, Gen 3 offspring that mature into adults show up in the model between years 12 to 20 (see timeline).

###Year 10

```
# Generation 3
# Year 10 New Population Growth

Gen3.Yr10 <- replicate(nreps(Gen2.Yr5, 10), list(Spread(11, 9, 3)))
Gen3.Yr10 <- do.call(rbind, Gen3.Yr10)
row_sub = apply(Gen3.Yr10, 1, function(row) all(row !=0 ))
Gen3.Yr10 <- Gen3.Yr10[row_sub,]
orig.shft <- filter(Gen2.Yr5, Year == 10)
Gen3.Yr10 <- Gen3.Yr10 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen3.Yr10
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen3.Yr10 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen3.Yr10, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```
C1 <- 80     #containment boundary set
AP <- 550
Years <- c(1:20)    #years where management will occur

#storage for management results
Containment2 <- Contain(Gen3.Yr10, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen3 <- Gen3.Yr10
Total.Pop <- rbind(Total.Pop, Gen3.Yr10)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 11

```
# Generation 3
# Year 11 New Population Growth
# Time steps at 10

Gen3.Yr11 <- replicate(nreps(Gen2.Yr6, 11), list(Spread(10, 10, 3)))
Gen3.Yr11 <- do.call(rbind, Gen3.Yr11)
row_sub = apply(Gen3.Yr11, 1, function(row) all(row !=0 ))
Gen3.Yr11 <- Gen3.Yr11[row_sub,]
orig.shft <- filter(Gen2.Yr6, Year == 11)
Gen3.Yr11 <- Gen3.Yr11 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen3.Yr11
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen3.Yr11 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen3.Yr11, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```
C1 <- 80      #containment boundary set
AP <- 550
Years <- c(1:20)     #years where management will occur

#storage for management results
Containment2 <- Contain(Gen3.Yr11, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen3 <- rbind(Gen3, Gen3.Yr11)
Total.Pop <- rbind(Total.Pop, Gen3.Yr11)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 12

```r
# Generation 3
# Year 12 New Population Growth

Gen3.Yr12 <- replicate(nreps(Gen2.Yr7, 12), list(Spread(9, 11, 3)))
Gen3.Yr12 <- do.call(rbind, Gen3.Yr12)
row_sub = apply(Gen3.Yr12, 1, function(row) all(row !=0 ))
Gen3.Yr12 <- Gen3.Yr12[row_sub,]
orig.shft <- filter(Gen2.Yr7, Year == 12)
Gen3.Yr12 <- Gen3.Yr12 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```r
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen3.Yr12
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen3.Yr12 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen3.Yr12, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```r
C1 <- 80     #containment boundary set
AP <- 550
Years <- c(1:20)    #years where management will occur

#storage for management results
Containment2 <- Contain(Gen3.Yr12, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen3 <- rbind(Gen3, Gen3.Yr12)
Total.Pop <- rbind(Total.Pop, Gen3.Yr12)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 13

```r
# Generation 3
# Year 13 New Population Growth
```

```r
# Time steps at 8

Gen3.Yr13 <- replicate(nreps(Gen2.Yr8, 13), list(Spread(8, 12, 3)))
Gen3.Yr13 <- do.call(rbind, Gen3.Yr13)
row_sub = apply(Gen3.Yr13, 1, function(row) all(row !=0 ))
Gen3.Yr13 <- Gen3.Yr13[row_sub,]
orig.shft <- filter(Gen2.Yr8, Year == 13)
Gen3.Yr13 <- Gen3.Yr13 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```r
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen3.Yr13
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen3.Yr13 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen3.Yr13, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```r
C1 <- 80      #containment boundary set
AP <- 550
Years <- c(1:20)     #years where management will occur

#storage for management results
Containment2 <- Contain(Gen3.Yr13, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen3 <- rbind(Gen3, Gen3.Yr13)
Total.Pop <- rbind(Total.Pop, Gen3.Yr13)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 14

```r
# Generation 3
# Year 14 New Population Growth
```

```r
Gen3.Yr14 <- replicate(nreps(Gen2.Yr9, 14), list(Spread(7, 13, 3)))
Gen3.Yr14 <- do.call(rbind, Gen3.Yr14)
row_sub = apply(Gen3.Yr14, 1, function(row) all(row !=0 ))
Gen3.Yr14 <- Gen3.Yr14[row_sub,]
orig.shft <- filter(Gen2.Yr9, Year == 14)
Gen3.Yr14 <- Gen3.Yr14 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```r
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen3.Yr14
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen3.Yr14 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen3.Yr14, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```r
C1 <- 80      #containment boundary set
AP <- 550
Years <- c(1:20)    #years where management will occur

#storage for management results
Containment2 <- Contain(Gen3.Yr14, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen3 <- rbind(Gen3, Gen3.Yr14)
Total.Pop <- rbind(Total.Pop, Gen3.Yr14)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

###Year 15

```r
# Generation 3
# Year 15 New Population Growth

Gen3.Yr15 <- replicate(nreps(Gen2.Yr10, 15), list(Spread(6, 14, 3)))
Gen3.Yr15 <- do.call(rbind, Gen3.Yr15)
```

```r
row_sub = apply(Gen3.Yr15, 1, function(row) all(row !=0 ))
Gen3.Yr15 <- Gen3.Yr15[row_sub,]
orig.shft <- filter(Gen2.Yr10, Year == 15)
Gen3.Yr15 <- Gen3.Yr15 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

```r
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen3.Yr15
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen3.Yr15 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen3.Yr15, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

```r
C1 <- 80      #containment boundary set
AP <- 550
Years <- c(1:20)     #years where management will occur

#storage for management results
Containment2 <- Contain(Gen3.Yr15, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen3 <- rbind(Gen3, Gen3.Yr15)
Total.Pop <- rbind(Total.Pop, Gen3.Yr15)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

##Generation 4 {.tabset} Generation 4 are the new adult trees that are the offspring of Generation 3 trees. The spatial-temporal model is applied at each year, with offspring maturing into adults after 4 years. As a result, Gen 4 offspring that mature into adults show up in the model between years 16 to 20 (see timeline).

###Year 15

```r
# Generation 4
# Year 15 New Population Growth

Gen4.Yr15 <- replicate(nreps(Gen3.Yr10, 15), list(Spread(6, 14, 4)))
```

```r
Gen4.Yr15 <- do.call(rbind, Gen4.Yr15)
row_sub = apply(Gen4.Yr15, 1, function(row) all(row !=0 ))
Gen4.Yr15 <- Gen4.Yr15[row_sub,]
orig.shft <- filter(Gen3.Yr10, Year == 15)
Gen4.Yr15 <- Gen4.Yr15 %>%
  mutate(x = x + orig.shft[,1], y = y + orig.shft[,2])
```

####Management

Exlusion/Asset Protection Strategy

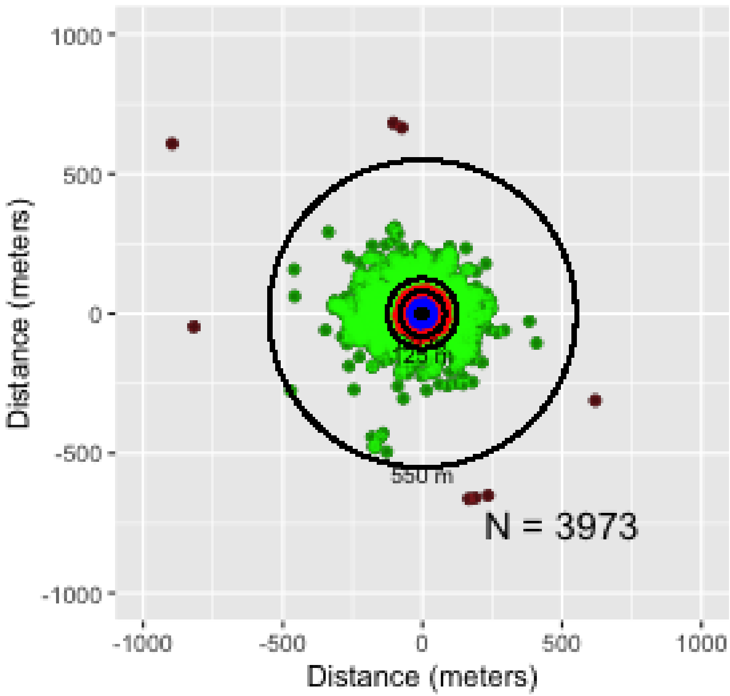
```r
# years where management will occur
Years <- c(1:20)

# radius from invasion point
AP <- 550

## add radius distance to polar coordinates
gen <- Gen4.Yr15
row.sums <- apply(gen[1:2]^2, 1, sum)
row.sums <- sqrt(row.sums)
gen <- rbind(cbind(gen, radius = row.sums))
Gen4.Yr15 <- gen

#storage for management results
Asset.Protec2 <- Exclude(Gen4.Yr15, AP, Years)

# Update count of total trees managed
Asset.Protec <- rbind(Asset.Protec, Asset.Protec2)
```

Containment Strategy

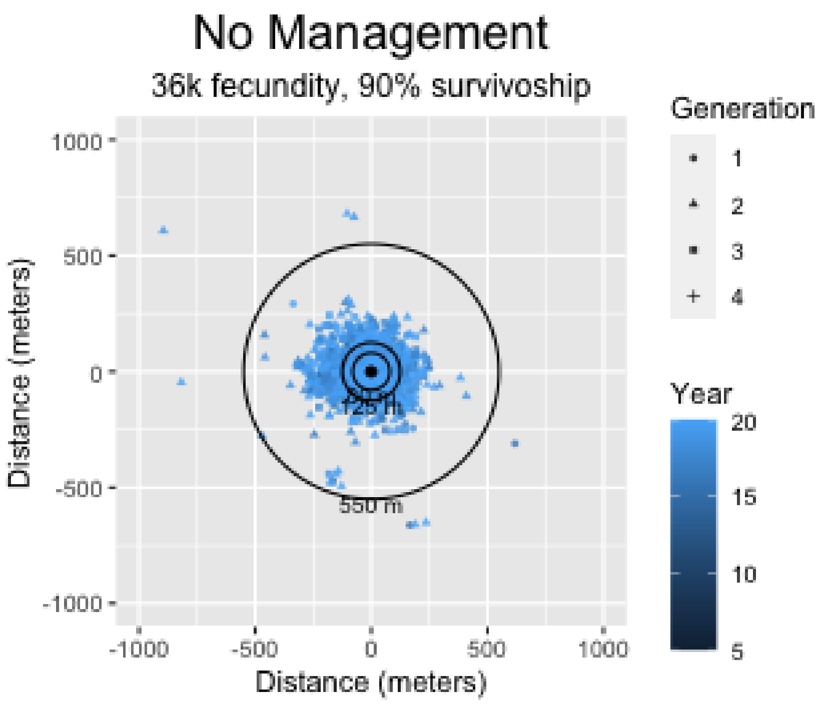
```r
C1 <- 80     #containment boundary set
AP <- 550
Years <- c(1:20)    #years where management will occur

#storage for management results
Containment2 <- Contain(Gen4.Yr15, C1, AP, Years)

#update total tree managed
Containment <- rbind(Containment, Containment2)

Total.Managed <- nrow(Asset.Protec) + nrow(Containment)

Gen4 <- Gen4.Yr15
Total.Pop <- rbind(Total.Pop, Gen4.Yr15)
Total.Remaining <- nrow(Total.Pop) - Total.Managed
```

```r
library(gridExtra)
  >
  > Attaching package: 'gridExtra'
  > The following object is masked from 'package:dplyr':
  >
  >     combine
library(grid)
library(ggforce)
```

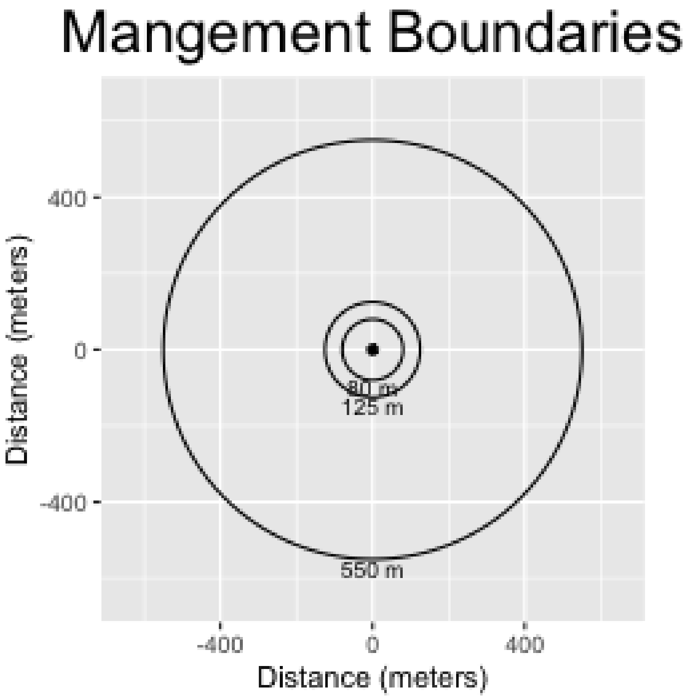
```
center <- Total.Pop %>%
                   filter(radius < 80)
plot80 <- Total.Pop %>%
  filter(radius >= 80 & radius < 125)
plot125 <- Total.Pop %>%
  filter(radius >= 125 & radius < 550)
plot550 <- Total.Pop %>%
  filter(radius >= 550)

p1 <- ggplot(Total.Pop) +
  theme(aspect.ratio = 1) +
  geom_point(aes(x = x, y = y))
p1 <- p1 + geom_point(data = center, aes(x = x, y = y),
             color = "blue",
             alpha = 1/2)
p1 <- p1 + geom_point(data = plot80, aes(x = x, y = y),
             color = "red",
             alpha = 1/2)
p1 <- p1 + geom_point(data = plot125, aes(x = x, y = y),
             color = "green",
             alpha = 1/2)
p1 <- p1 + geom_point(data = plot550, aes(x = x, y = y),
             color = "brown",
             alpha = 1/2)
p1 <- p1 + geom_point(aes(x = 0, y = 0))
p1 <- p1 + geom_circle(aes(x0 = 0, y0 = 0, r = 550))
p1 <- p1 + geom_circle(aes(x0 = 0, y0 = 0, r = 80))
p1 <- p1 + geom_circle(aes(x0 = 0, y0 = 0, r = 125)) +
labs(x = "Distance (meters)",
     y = "Distance (meters)") +
  theme(plot.title = element_text(hjust = 0.5, size = 22)) +
  ylim(-1000,1000)+
  xlim(-1000,1000)
p1 <- p1 + annotate("text", x = 0, y = -96, label = "80 m", size = 3)
p1 <- p1 + annotate("text", x = 0, y = -146, label = "125 m", size = 3)
p1 <- p1 + annotate("text", x = 0, y = -575, label = "550 m", size = 3)
p1 <- p1 + annotate("text", x = 500, y = -750, label = "N = 3973", size = 5)
p1
```




```
library(gridExtra)
library(grid)
library(ggforce)

p1 <- ggplot() +
  theme(aspect.ratio = 1) +
  geom_point(Total.Pop, mapping = aes(x = x, y = y,
                                      color = Year,
                                      shape = factor(Generation)),
             size = 1,
             alpha = 0.8)
p1 <- p1 + geom_point(aes(x = 0, y = 0))
p1 <- p1 + geom_circle(aes(x0 = 0, y0 = 0, r = 550))
p1 <- p1 + geom_circle(aes(x0 = 0, y0 = 0, r = 80))
p1 <- p1 + geom_circle(aes(x0 = 0, y0 = 0, r = 125)) +
labs(title = "No Management",
     subtitle = "36k fecundity, 90% survivoship",
     shape = "Generation",
     x = "Distance (meters)",
     y = "Distance (meters)") +
  theme(plot.title = element_text(hjust = 0.5, size = 18),
        plot.subtitle = element_text(hjust = 0.5, size = 12)) +
  ylim(-1000,1000)+
  xlim(-1000,1000)
p1 <- p1 + annotate("text", x = 0, y = -96, label = "80 m", size = 3)
p1 <- p1 + annotate("text", x = 0, y = -146, label = "125 m", size = 3)
p1 <- p1 + annotate("text", x = 0, y = -575, label = "550 m", size = 3)
p1
```




```
p2 <- ggplot() +
  theme(aspect.ratio = 1)
p2 <- p2 + geom_point(aes(x = 0, y = 0))
p2 <- p2 + geom_circle(aes(x0 = 0, y0 = 0, r = 550))
p2 <- p2 + geom_circle(aes(x0 = 0, y0 = 0, r = 80))
p2 <- p2 + geom_circle(aes(x0 = 0, y0 = 0, r = 125)) +
labs(title = "Mangement Boundaries",
     x = "Distance (meters)",
     y = "Distance (meters)") +
  theme(plot.title = element_text(hjust = 0.5, size = 22)) +
  ylim(-650,650)+
  xlim(-650,650)
p2 <- p2 + annotate("text", x = 0, y = -96, label = "80 m", size = 3)
p2 <- p2 + annotate("text", x = 0, y = -146, label = "125 m", size = 3)
p2 <- p2 + annotate("text", x = 0, y = -575, label = "550 m", size = 3)
p2
```



##Outputs

```
library(readxl)

#Functions for outputs and evaluation
AP.list <- list()
for(i in 1:20){
  AP.list[[i]] <- nrow(filter(Asset.Protec, Year == i))
  rbind(AP.list[[i]])
}
AP.list <- do.call(rbind, AP.list)

Contain.list <- list()
for(i in 1:20){
  Contain.list[[i]] <- nrow(filter(Containment, Year == i))
  rbind(Contain.list[[i]])
}
Contain.list <- do.call(rbind, Contain.list)

Pop.list <- list()
for(i in 1:20){
  Pop.list[[i]] <- nrow(filter(Total.Pop, Year == i))
  rbind(Pop.list[[i]])
}
Pop.list <- do.call(rbind, Pop.list)
#Containment/Exclusion
#write.xlsx(Contain.list, file = "NoMng_36KFecundity_90AdultSurv_C1.xlsx")
#Asset Protection
#write.xlsx(AP.list, file = "NoMng_36KFecundity_90AdultSurv_C1AP.xlsx")
#Unmanaged Population
#write.xlsx(Pop.list, file = "NoMng_36KFecundity_90AdultSurv_byYear.xlsx")
#OverPop
```

```r
#write.xlsx(Total.Pop, file = "NoMng_36KFecundity_90AdultSurv.xlsx")
```