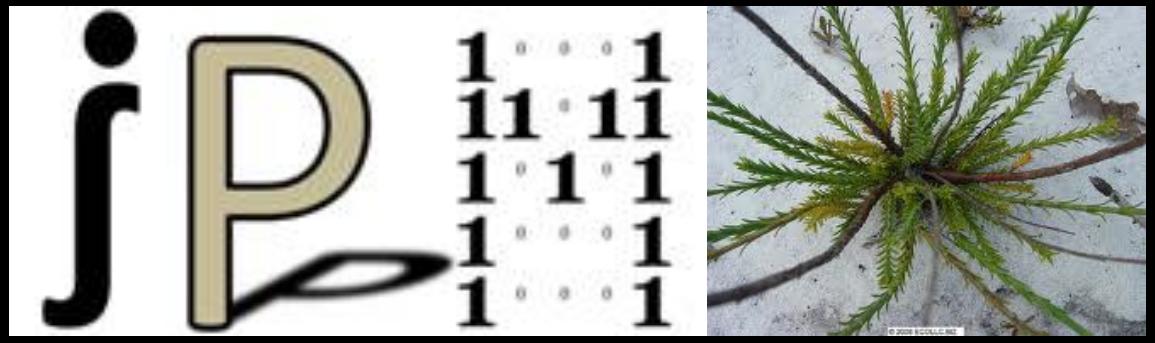
ESA 2015 - Integral Project Model

IPMing a complex life cycle: *Hypericum cumulicola*



Rob Salguero-Gómez, Jessica Metcalf, Sean McMahon, Eelke Jongejans & Cory Merow

MAX PLANCK INSTITUTE FOR DEMOGRAPHIC RESEARCH

UNIVERSITY OF OXFORD

Smithsonian Institution

Radboud University Nijmegen

University of Connecticut

IPMing a complex life cycle - *Hypericum*

Hypericum cumulicola (Small) P. Adams (Hypericaceae)

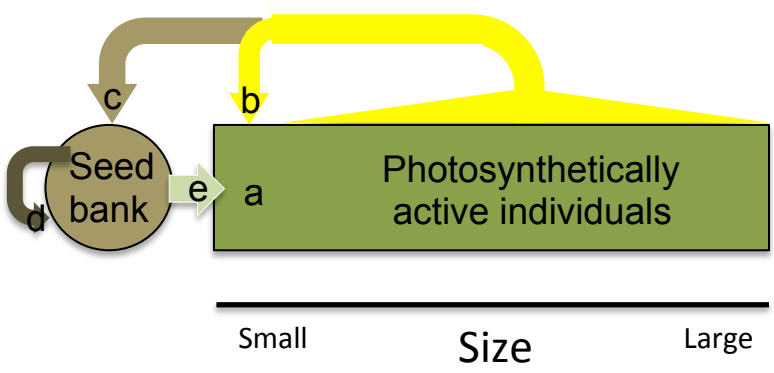
A Fire-Explicit Population Viability Analysis of *Hypericum cumulicola* in Florida Rosemary Scrub

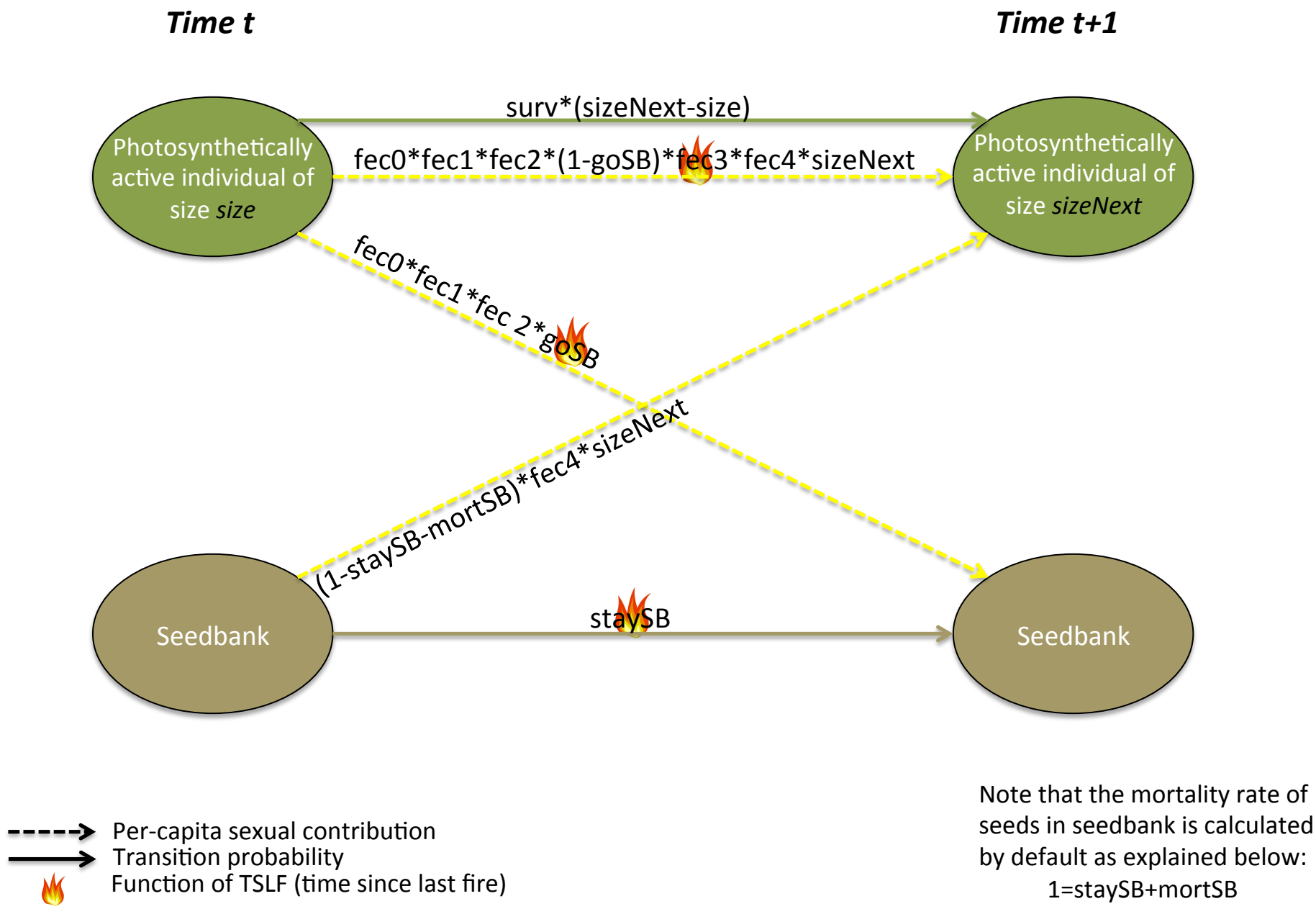
PEDRO FRANCISCO QUINTANA-ASCENCIO,*† ERIC S. MENGES,*
AND CARL W. WEEKLEY*

A (kinda) *complex life cycle*

IPMing a complex life cycle - *Hypericum*

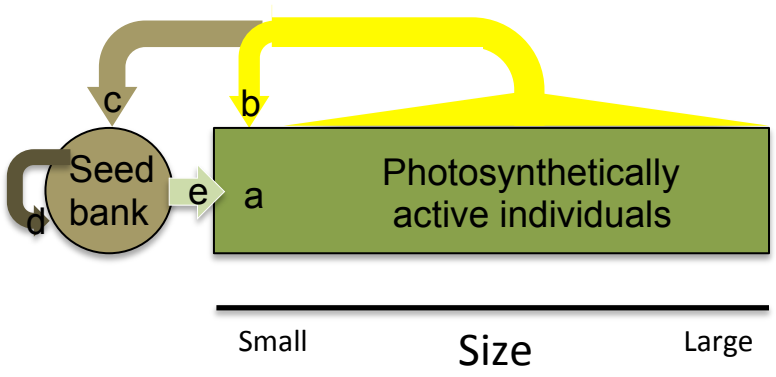
Hypericum cumulicola

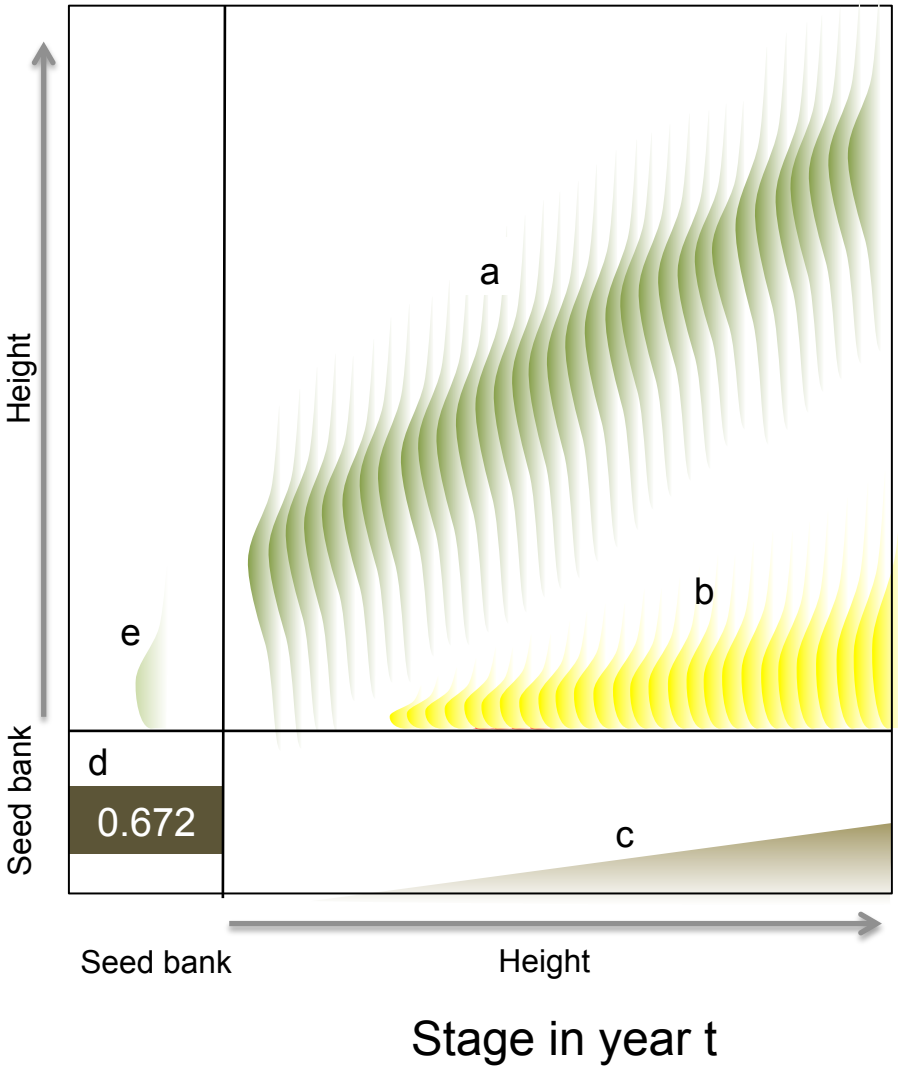



Time t
Time t+1
surv*(sizeNext-size)
Photosynthetically active individual of size size
fec0*fec1*fec2*(1-goSB)*fec3*fec4*sizeNext
Photosynthetically active individual of size sizeNext
fec0*fec1*fec 2*goSB
(1-staySB-mortSB)*fec4*sizeNext
Seedbank
staySB
Seedbank
Per-capita sexual contribution
Transition probability
Function of TSLF (time since last fire)
Note that the mortality rate of seeds in seedbank is calculated by default as explained below:
1=staySB+mortSB

IPMing a complex life cycle - *Hypericum*

Hypericum cumulicola





IPMing a complex life cycle - *Hypericum*

Explaining modular organization

Getting some important stuff in

Module I: building IPM
Survival
Growth
Fecundity
dto
Pmatrix
Diagnostics
Fmatrix
IPM

Module II: Basic demographic output
Population growth rate
RV
SSD
Perturbations

Getting some important stuff in

```
rm(list=ls(all=TRUE))

#Load IPMpack (Make sure you are working with version 2.1!)
library(IPMpack)

#Information on the subset of information made available in IPMpack 2.1 can be accessed through the help manual
data(dataIPMpackHypericum)

#Take a look at the help file of this data in IPMpack to understand how it is structured
help(dataIPMpackHypericum)
d1 <- dataIPMpackHypericum

#Due to the sampling design described in the help manual, here we consider only individuals for which we are ce
d1 <- subset(d1,is.na(d1$size)==FALSE | d1$ontogenyNext==1)

#Side experiments carried out by Quintana-Ascencio and Menges estimated the following vital rates for the studi
  #Number of seeds produced per fruit:
    fec2 <- 13.78
  #Probability of seedling establishment
    fec3 <- 0.001336
  #Probability of seedling survival half a year after germinating, corresponding to the next annual census
    fec4 <- 0.14
  #Probability of a seed going into the seed bank
    goSB <- 0.08234528
  #Probability of a seed staying in the seed bank
    staySB <- 0.671

  #The following part does a simple re-organization of the data, getting rid of non-critical information for th
    d1 <- d1[,c("surv","size","sizeNext","fec0","fec1")]

  #The following lines of code state the continuous (max height of individual plant) part of the IPM. Note that
    d1$stageNext <- d1$stage <- "continuous"
    d1$stage[is.na(d1$size)] <- NA

  #If an individual did not survive, it is labelled as dead to t+1.
    d1$stageNext[d1$surv==0] <- "dead"
```

Getting some important stuff in

```
#The following lines of command compile a dataframe that contains the population dynamics of the seedba
  d1$number <- 1
  sb1 <- data.frame(stage=c("seedbank","seedbank","continuous"),
                    stageNext=c("seedbank","continuous","seedbank"),
                    surv=1,size=NA,sizeNext=NA,fec0=NA,fec1=NA,number=c(staySB,(1-staySB)*fec3*fec4,1))
  sb1

#The following lines simply put together the parts of the dataset for continuous and discrete parts of
  d1 <- rbind(d1,sb1)
  d1$stage <- as.factor(d1$stage)
  d1$stageNext <- as.factor(d1$stageNext)
#Note that the variables stage and stageNext must be factors to work with IPMpack (common mistake to f
```

```
>       sb1
       stage  stageNext surv size sizeNext fec0 fec1       number
1   seedbank   seedbank    1   NA       NA   NA   NA 6.710000e-01
2   seedbank continuous    1   NA       NA   NA   NA 6.153616e-05
3 continuous   seedbank    1   NA       NA   NA   NA 1.000000e+00
```

Getting some important stuff in

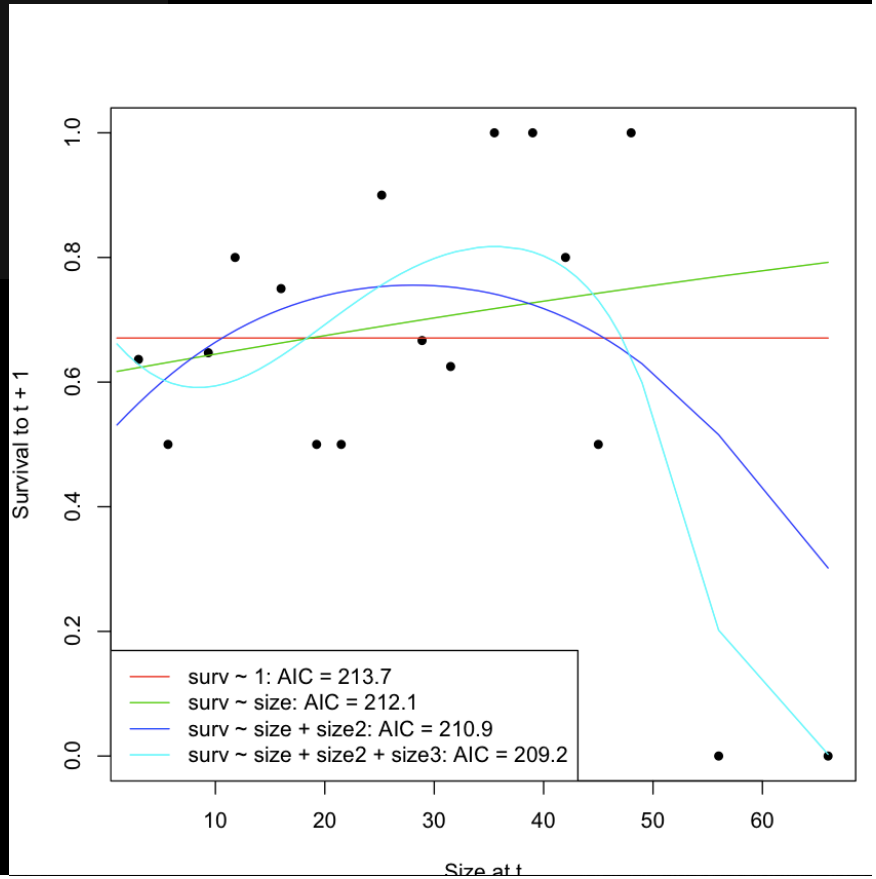
```
#The following lines simply put together the pa
  d1 <- rbind(d1,sb1)
  d1$stage <- as.factor(d1$stage)
  d1$stageNext <- as.factor(d1$stageNext)
#Note that the variables stage and stageNext mu
```

```
> head(d1)
  surv size sizeNext fec0 fec1      stage  stageNext number
1    1    1        3    0   NA continuous continuous      1
2    1    1        3    0   NA continuous continuous      1
3    1    2        1    0   NA continuous continuous      1
4    1    2        2    0   NA continuous continuous      1
5    1    2        3    0   NA continuous continuous      1
6    1    2        3    0   NA continuous continuous      1
```

```
    surv size sizeNext fec0 fec1      stage  stageNext number
183   NA   NA        6   NA   NA       <NA> continuous      1
184   NA   NA        6   NA   NA       <NA> continuous      1
185   NA   NA        9   NA   NA       <NA> continuous      1
186    1   NA       NA   NA   NA   seedbank   seedbank      1
187    1   NA       NA   NA   NA   seedbank continuous      1
188    1   NA       NA   NA   NA continuous   seedbank      1
```

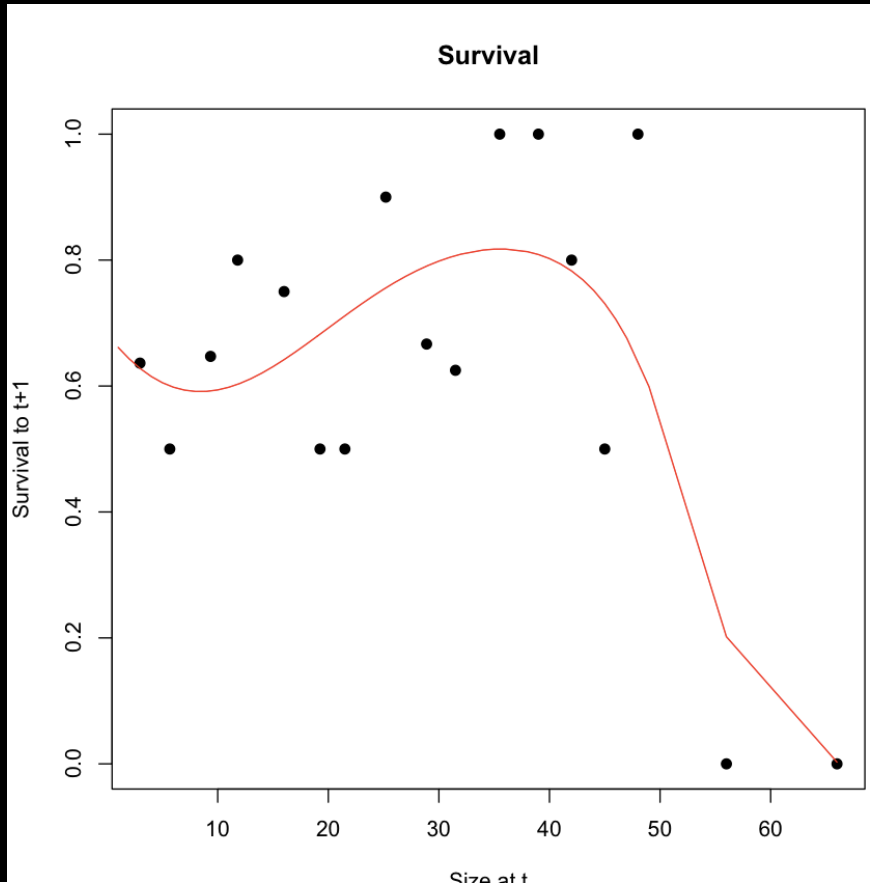
Object construction and model comparison

```
#The following plot will compare a model for survival that is indepe
    soComparison <- survModelComp(d1,
                    expVars = c(surv~1,
                                surv~size,
                                surv~size + size2,
                                surv~size + size2 + size3),
                                testType = "AIC",
                                makePlot = T)

#Exercise 1. Choose from the numbered lines of code below the approp
#Choice 1:
  so <- makeSurvObj(d1, Formula = surv~1)
#Choice 2:
  so <- makeSurvObj(d1, Formula = surv~size)
#Choice 3:
  so <- makeSurvObj(d1, Formula = surv~size +size2)
#Choice 4:
  so <- makeSurvObj(d1, Formula = surv~size +size2+size3)
```



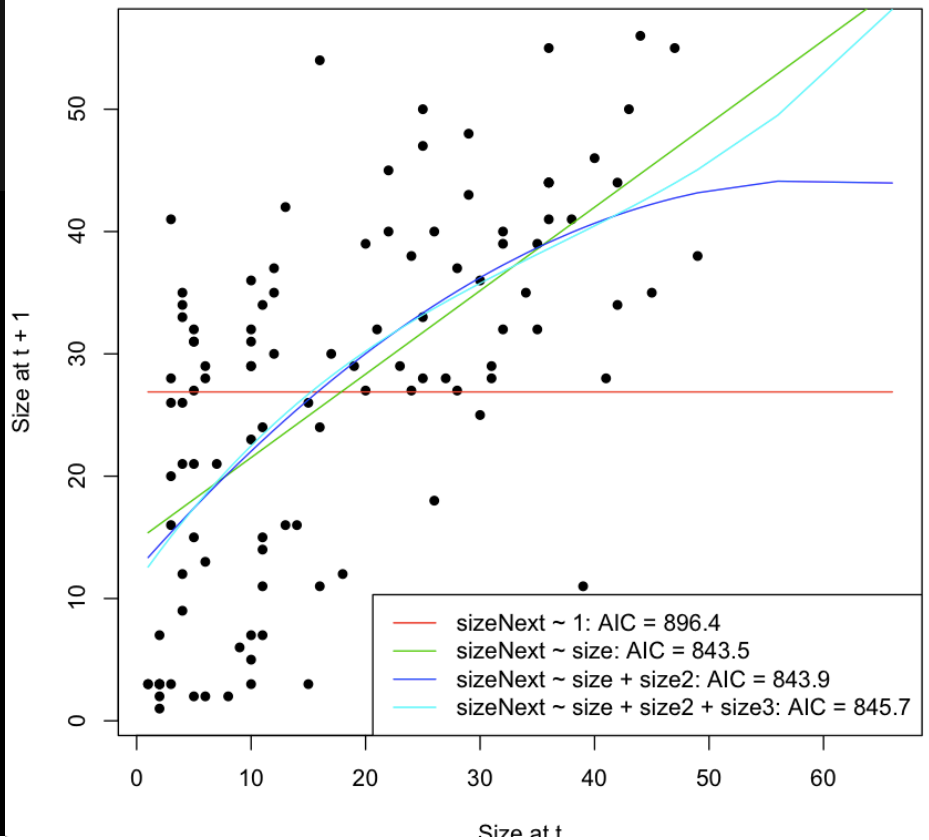
Object construction and model comparison

```
#Answer: The polynomial
    picSurv(d1, so)
```



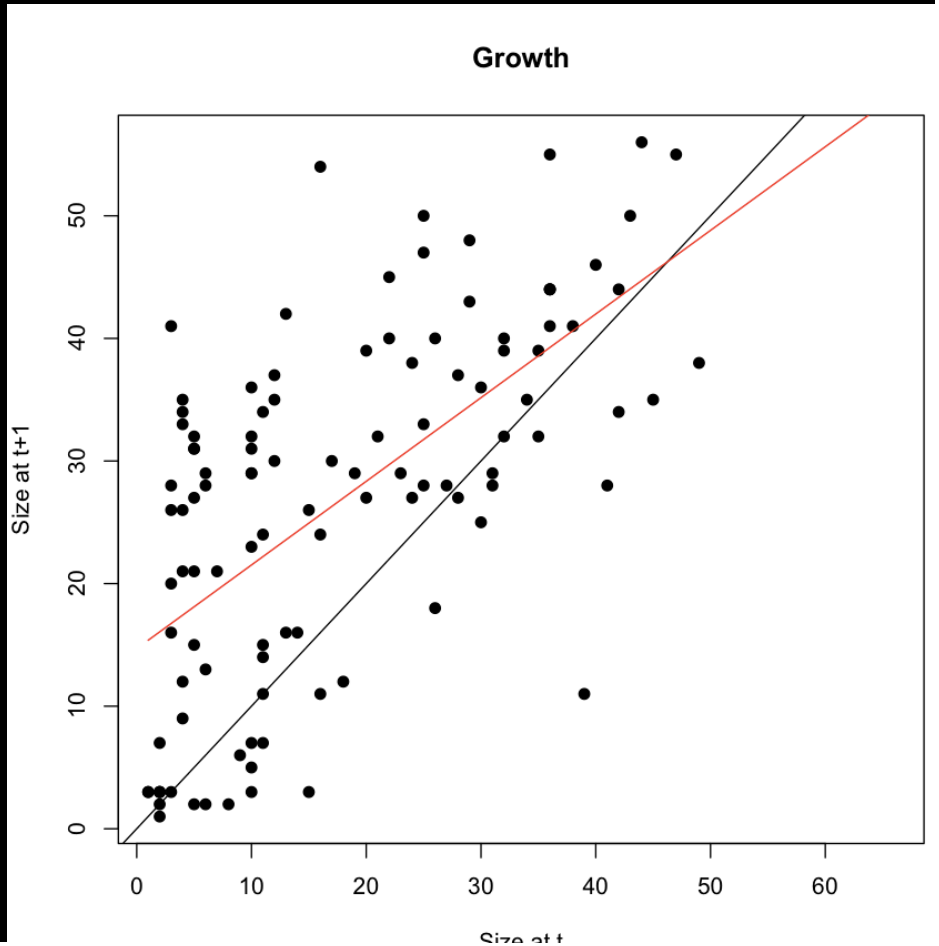
Object construction and model comparison

```
#The following plot will compare growth models from t to t+1, just as you did before for sur
    goComparison <- growthModelComp(d1,
                                    expVars = c(sizeNext~1,
                                                sizeNext~size,
                                                sizeNext~size + size2,
                                                sizeNext~size + size2 + size3),
                                    testType = "AIC",
                                    makePlot = T,
                                    legendPos = "bottomright")

#Exercise 2. Choose from the numbered lines of code below the appropriate model for growth
#Choice 1:
  go <- makeGrowthObj(d1, Formula = sizeNext~1)
#Choice 2:
  go <- makeGrowthObj(d1, Formula = sizeNext~size)
#Choice 3:
  go <- makeGrowthObj(d1, Formula = sizeNext~size +size2)
#Choice 4:
  go <- makeGrowthObj(d1, Formula = sizeNext~size +size2+size3)
```



Object construction and model comparison

```
#The following will save
    picGrow(d1, go)
    #Note that the black
```

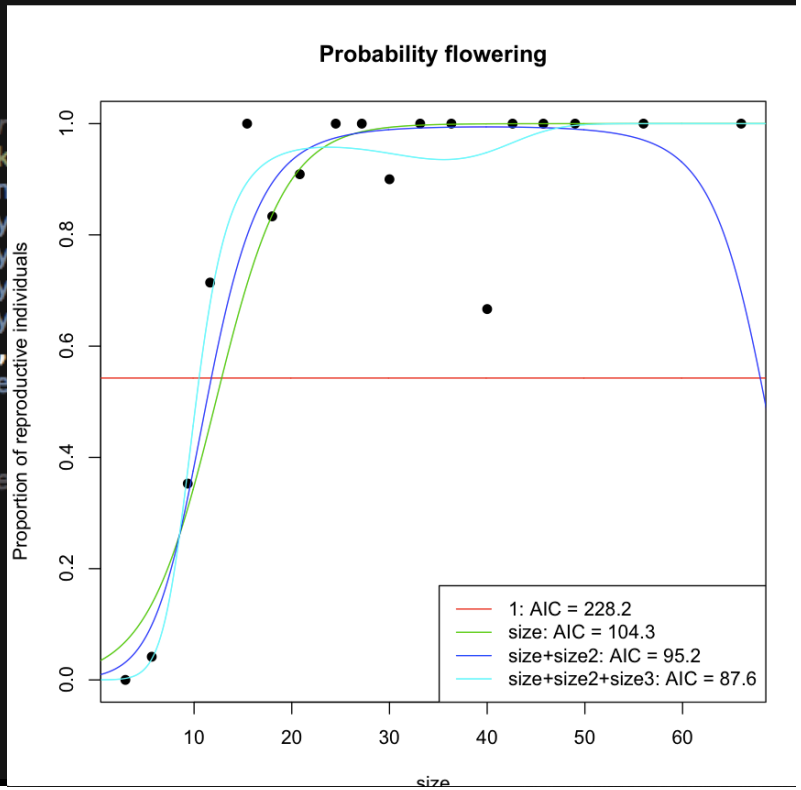


Object construction and model comparison

```
#The following are some of the options for fec0 that will be compared visually and quantitatively below
  fec01 <- makeFecObj(d1, Formula = fec0~1, Family = "binomial")
  fec02 <- makeFecObj(d1, Formula = fec0~size, Family = "binomial")
  fec03 <- makeFecObj(d1, Formula = fec0~size+size2, Family = "binomial")
  fec04 <- makeFecObj(d1, Formula = fec0~size+size2+size3, Family = "binomial")

#The following are a few lines re-organize the data for plotting below
  #Reorganizes individuals by size in time t
    fs <- order(d1$size)
  #Values of reproduction organized by size of individuals
    fsFec0 <- (d1$fec0)[fs]
    fsSize <- (d1$size)[fs]
  #Means at cut-points to be plotted for size and fec0
    pfz <- tapply(fsSize, as.numeric(cut(fsSize, 21)), mean, na.rm = TRUE)
    ps0 <- tapply(fsFec0, as.numeric(cut(fsSize, 21)), mean, na.rm = TRUE)

#Make dummy size axis
  x <- seq(from = 0, to = 100, length = 1001)
  x0 <- data.frame(size = x, size2 = x^2, size3 =x^3)

#This is the actual plot produced to compare the models for fec0 (the probability of being reproductive at time t)
  plot(as.numeric(pfz), as.numeric(ps0), pch = 19, cex = 1, col = "black", ylim = c(0, 1),
    xlab = "size", ylab = "Proportion of reproductive individuals", main = "Probability flowering")
    y0 <- predict(fec01@fitFec[[1]], newdata = x0); y0 <- exp(y0)/(exp(y0)+1); lines(x, y0, col=2)
    y0 <- predict(fec02@fitFec[[1]], newdata = x0); y0 <- exp(y0)/(exp(y0)+1); lines(x, y0, col=3)
    y0 <- predict(fec03@fitFec[[1]], newdata = x0); y0 <- exp(y0)/(exp(y0)+1); lines(x, y0, col=4)
    y0 <- predict(fec04@fitFec[[1]], newdata = x0); y0 <- exp(y0)/(exp(y0)+1); lines(x, y0, col=5)
    legend("bottomright", legend = sprintf("%s: %s = %.1f",c("1","size","size+size2","size+size2+size3"),
      c("AIC"), c(AIC(fec01@fitFec[[1]]), AIC(fec02@fitFec[[1]]), AIC(fec03@fitFec[[1]]), AIC(fec04@fitFec[[1]]))),
      col = c(2:5), lty = 1, xjust = 1, bg = "white")

#Exercise 3: Choose (i.e. execute only that line) from the lines of code below the appropriate model for the probability of
#Choice 1:
  fec0ChosenModel <- fec0~1
#Choice 2:
  fec0ChosenModel <- fec0~size
#Choice 3:
  fec0ChosenModel <- fec0~size+size2
#Choice 4:
  fec0ChosenModel <- fec0~size+size2+size3
```

Object construction and model comparison

```r
#The following are some of the options for fec0 that will be compared visually and quantitatively below
  fec01 <- makeFecObj(d1, Formula = fec0~1, Family = "binomial")
  fec02 <- makeFecObj(d1, Formula = fec0~size, Family = "binomial")
  fec03 <- makeFecObj(d1, Formula = fec0~size+size2, Family = "binomial")
  fec04 <- makeFecObj(d1, Formula = fec0~size+size2+size3, Family = "binomial")

#The following are a few lines re-organize the data for plotting below
  #Reorganizes individuals by size in time t
    fs <- order(d1$size)
  #Values of reproduction organized by size of individuals
    fsFec0 <- (d1$fec0)[fs]
    fsSize <- (d1$size)[fs]
  #Means at cut-points to be plotted for size and fec0
    pfz <- tapply(fsSize, as.numeric(cut(fsSize, 21)), mean, na.rm = TRUE)
    ps0 <- tapply(fsFec0, as.numeric(cut(fsSize, 21)), mean, na.rm = TRUE)

#Make dummy size axis
  x <- seq(from = 0, to = 100, length = 1001)
  x0 <- data.frame(size = x, size2 = x^2, size3 =x^3)

#This is the actual plot produced to compare the models for fec0 (the pr
  plot(as.numeric(pfz), as.numeric(ps0), pch = 19, cex = 1, col = "black
    xlab = "size", ylab = "Proportion of reproductive individuals", main
    y0 <- predict(fec01@fitFec[[1]], newdata = x0); y0 <- exp(y0)/(exp(y
    y0 <- predict(fec02@fitFec[[1]], newdata = x0); y0 <- exp(y0)/(exp(y
    y0 <- predict(fec03@fitFec[[1]], newdata = x0); y0 <- exp(y0)/(exp(y
    y0 <- predict(fec04@fitFec[[1]], newdata = x0); y0 <- exp(y0)/(exp(y
    legend("bottomright", legend = sprintf("%s: %s = %.1f",c("1","size",
      c("AIC"), c(AIC(fec01@fitFec[[1]]), AIC(fec02@fitFec[[1]]), AIC(fe
      col = c(2:5), lty = 1, xjust = 1, bg = "white")

#Exercise 3: Choose (i.e. execute only that line) from the lines of code
#Choice 1:
  fec0ChosenModel <- fec0~1
#Choice 2:
  fec0ChosenModel <- fec0~size
#Choice 3:
  fec0ChosenModel <- fec0~size+size2
#Choice 4:
  fec0ChosenModel <- fec0~size+size2+size3
```

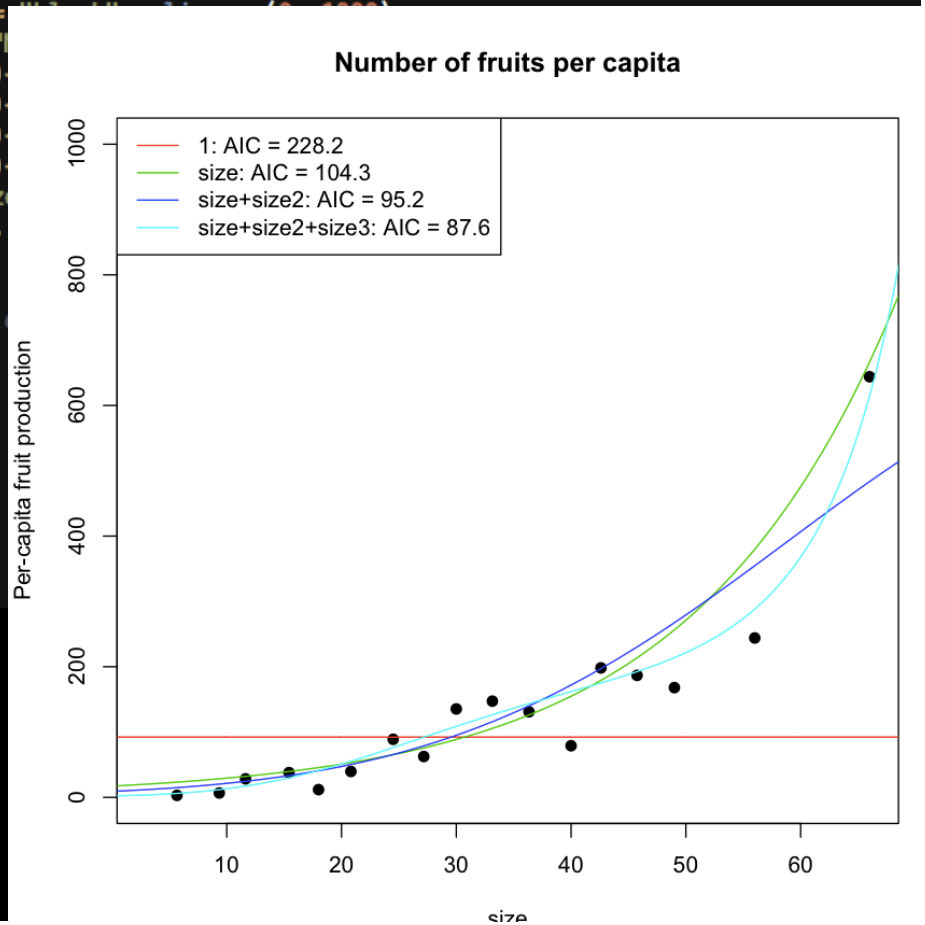


Object construction and model comparison

```
#Number of fruits produced per capita (fec1)
  #Values of fruit production organized by size of individuals
  fsFec1 <- (d1$fec1)[fs]
  #Means at cut-points to be plotted for size and fec0
  ps1 <- tapply(fsFec1, as.numeric(cut(fsSize, 21)), mean, na.rm = TRUE)

  #The following are some of the options for fec1 that will be compared visually and quantitatively below
    fec11 <- makeFecObj(d1, Formula = fec1~1, Family = "poisson")
    fec12 <- makeFecObj(d1, Formula = fec1~size, Family = "poisson")
    fec13 <- makeFecObj(d1, Formula = fec1~size+size2, Family = "poisson")
    fec14 <- makeFecObj(d1, Formula = fec1~size+size2+size3, Family = "poisson")

  #This is the actual plot produced to compare the models for fec1
    plot(as.numeric(pfz), as.numeric(ps1), pch = 19, cex = 1, col = "black", ylim = c(0, 1000),
      xlab = "size", ylab = "Per-capita fruit production", main = "Number of fruits per capita")
      y1 <- predict(fec11@fitFec[[1]], newdata = x0); y1 <- exp(y1)+1; lines(x, y1, col=2)
      y1 <- predict(fec12@fitFec[[1]], newdata = x0); y1 <- exp(y1)+1; lines(x, y1, col=3)
      y1 <- predict(fec13@fitFec[[1]], newdata = x0); y1 <- exp(y1)+1; lines(x, y1, col=4)
      y1 <- predict(fec14@fitFec[[1]], newdata = x0); y1 <- exp(y1)+1; lines(x, y1, col=5)
      legend("topleft", legend = sprintf("%s: %s = %.1f",c("1","size","size+size2","size+size2+size3"),
        c("AIC"), c(AIC(fec01@fitFec[[1]]), AIC(fec02@fitFec[[1]]), AIC(fec03@fitFec[[1]]), AIC(fec04@fitFec[[1]]))),
        col = c(2:5), lty = 1, xjust = 1, bg = "white")

  #Exercise 4: Choose (i.e. execute only that line) from the lines of code below the appropriate model for fruit production, fec1
  #Choice 1:
    fec1ChosenModel <- fec1~1
  #Choice 2:
    fec1ChosenModel <- fec1~size
  #Choice 3:
    fec1ChosenModel <- fec1~size+size2
  #Choice 4:
    fec1ChosenModel <- fec1~size+size2+size3
```

Object construction and model comparison

```
#Number of fruits produced per capita (fec1)
  #Values of fruit production organized by size of individuals
  fsFec1 <- (d1$fec1)[fs]
  #Means at cut-points to be plotted for size and fec0
  ps1 <- tapply(fsFec1, as.numeric(cut(fsSize, 21)), mean, na.rm = TRUE)

  #The following are some of the options for fec1 that will be compared visually and quantitatively below
    fec11 <- makeFecObj(d1, Formula = fec1~1, Family = "poisson")
    fec12 <- makeFecObj(d1, Formula = fec1~size, Family = "poisson")
    fec13 <- makeFecObj(d1, Formula = fec1~size+size2, Family = "poisson")
    fec14 <- makeFecObj(d1, Formula = fec1~size+size2+size3, Family = "poisson")

  #This is the actual plot produced to compare the models for fec1
    plot(as.numeric(pfz), as.numeric(ps1), pch = 19, cex = 1, col =
      xlab = "size", ylab = "Per-capita fruit production", main = "
      y1 <- predict(fec11@fitFec[[1]], newdata = x0); y1 <- exp(y1)
      y1 <- predict(fec12@fitFec[[1]], newdata = x0); y1 <- exp(y1)
      y1 <- predict(fec13@fitFec[[1]], newdata = x0); y1 <- exp(y1)
      y1 <- predict(fec14@fitFec[[1]], newdata = x0); y1 <- exp(y1)
      legend("topleft", legend = sprintf("%s: %s = %.1f",c("1","siz
        c("AIC"), c(AIC(fec01@fitFec[[1]]), AIC(fec02@fitFec[[1]]),
        col = c(2:5), lty = 1, xjust = 1, bg = "white")

  #Exercise 4: Choose (i.e. execute only that line) from the lines
  #Choice 1:
    fec1ChosenModel <- fec1~1
  #Choice 2:
    fec1ChosenModel <- fec1~size
  #Choice 3:
    fec1ChosenModel <- fec1~size+size2
  #Choice 4:
    fec1ChosenModel <- fec1~size+size2+size3
```

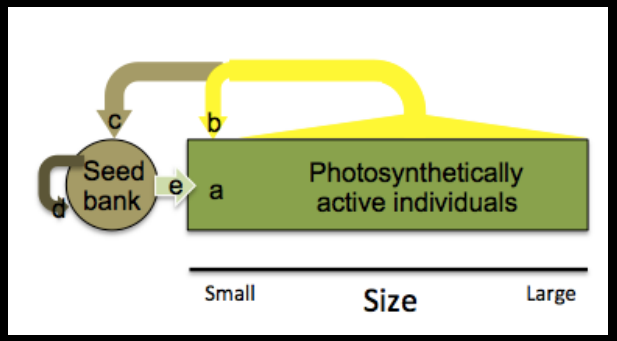


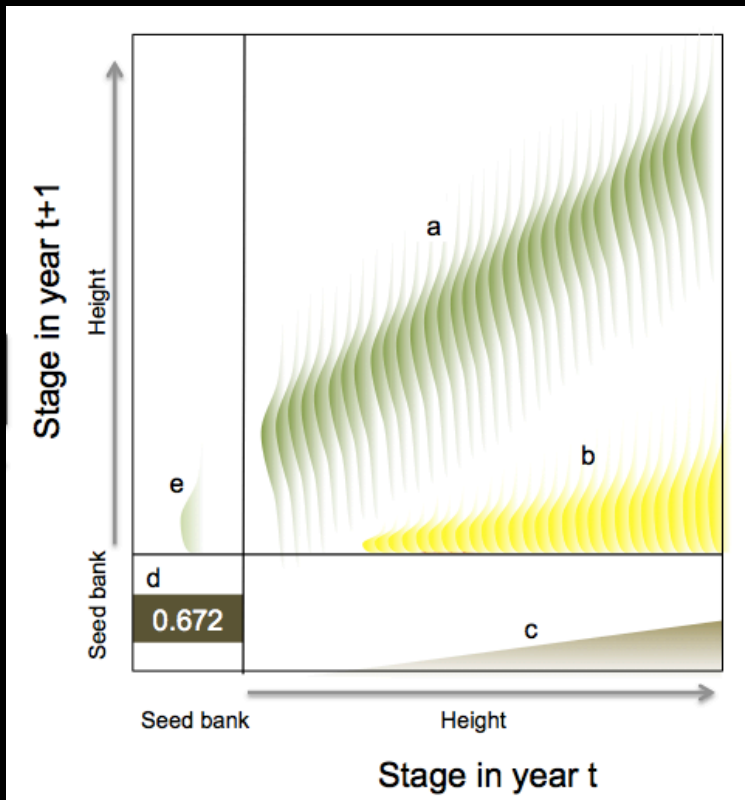
Object construction and model comparison

```
#Fecundity is almost always the most complicated part of an IPM, and things can get even more complicated when
  fo <- makeFecObj(d1, Formula = c(fec0ChosenModel, fec1ChosenModel),
                   Family = c("binomial", "poisson"),
                   Transform = c("none", "none"),
                   meanOffspringSize = mean(d1[is.na(d1$size)==TRUE & is.na(d1$sizeNext)==FALSE, "sizeNext"]),
                   sdOffspringSize = sd(d1[is.na(d1$size)==TRUE & is.na(d1$sizeNext)==FALSE, "sizeNext"]),
                   fecConstants = data.frame(fec2=fec2,fec3=fec3,fec4=fec4),
                   offspringSplitter = data.frame(seedbank = goSB,continuous=(1-goSB)),
                   vitalRatesPerOffspringType = data.frame(seedbank = c(1, 1, 1, 0, 0),
                                                           continuous = rep(1,5),
                                                           row.names = c("fec0", "fec1", "fec2", "fec3", "fec4")))
```

Object construction and model comparison

```
#Fecundity is almost always the most complicated part of an IPM, and things can get even more complicated when
  fo <- makeFecObj(d1, Formula = c(fec0ChosenModel, fec1ChosenModel),
                   Family = c("binomial", "poisson"),
                   Transform = c("none", "none"),
                   meanOffspringSize = mean(d1[is.na(d1$size)==TRUE & is.na(d1$sizeNext)==FALSE, "sizeNext"]),
                   sdOffspringSize = sd(d1[is.na(d1$size)==TRUE & is.na(d1$sizeNext)==FALSE, "sizeNext"]),
                   fecConstants = data.frame(fec2=fec2,fec3=fec3,fec4=fec4),
                   offspringSplitter = data.frame(seedbank = goSB,continuous=(1-goSB)),
                   vitalRatesPerOffspringType = data.frame(seedbank = c(1, 1, 1, 0, 0),
                                                           continuous = rep(1,5),
                                                           row.names = c("fec0", "fec1", "fec2", "fec3", "fec4")))
```



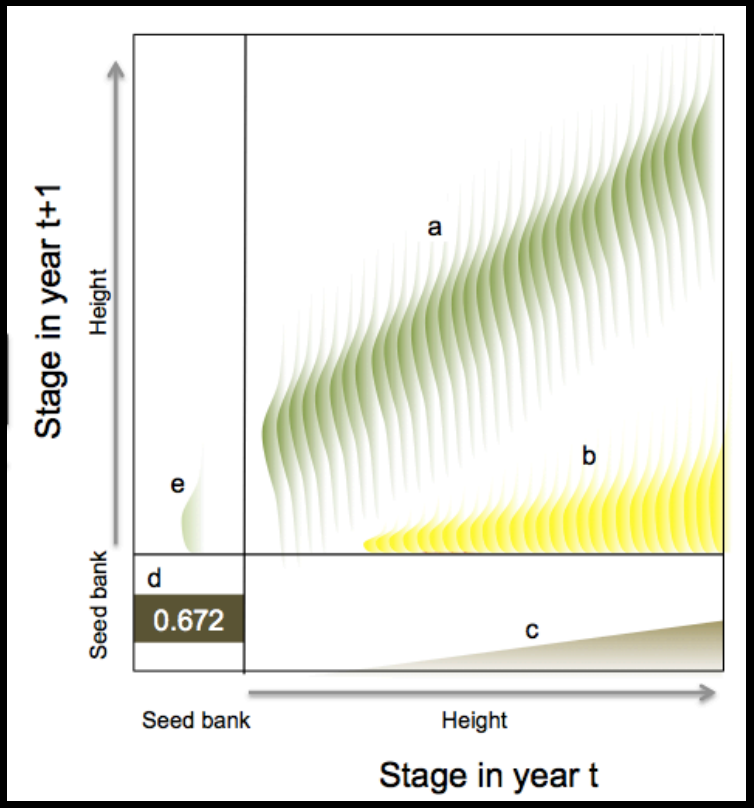


Object construction and model comparison

```
#The following part creates a matrix that will accommodate the probabilit
  dto <- makeDiscreteTrans(d1)
  dummy <- as.matrix(fo@offspringRel$coefficients[1])
  dimnames(dummy) <- list(1, "seedbank")
  dto@meanToCont <- as.matrix(dummy, dimnames = c(1, "seedbank"))
  dummy <- as.matrix(fo@sdOffspringSize)
  dimnames(dummy) <- list(1,"seedbank")
  dto@sdToCont <- as.matrix(dummy, dimnames = c(1, "seedbank"))
```

Object construction and model comparison

```
#The following part creates a matrix that will accommodate the probabilit
  dto <- makeDiscreteTrans(d1)
  dummy <- as.matrix(fo@offspringRel$coefficients[1])
  dimnames(dummy) <- list(1, "seedbank")
  dto@meanToCont <- as.matrix(dummy, di
  dummy <- as.matrix(fo@sdOffspringSize)
  dimnames(dummy) <- list(1,"seedbank")
  dto@sdToCont <- as.matrix(dummy, dimna
```



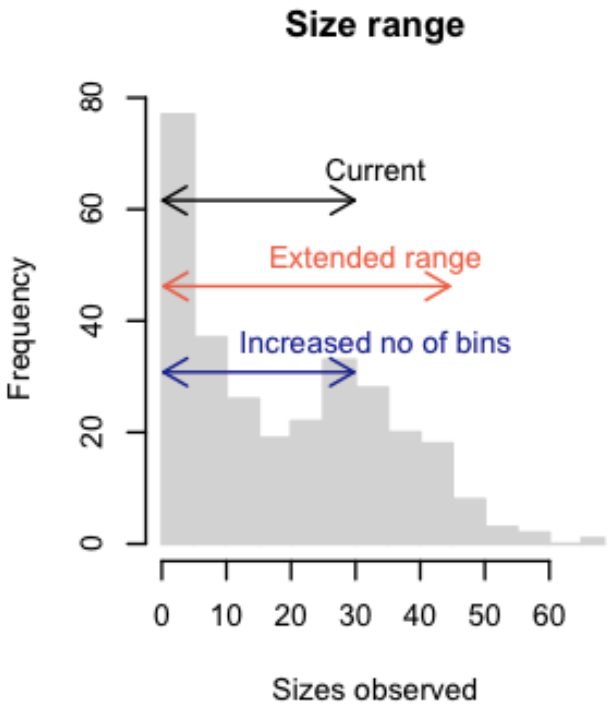

```
> dto
An object of class "discreteTrans"
Slot "discreteTrans":
           seedbank  continuous
seedbank        0.5 0.006060606
continuous      0.5 0.660606061
dead            0.0 0.333333333

Slot "meanToCont":
  seedbank
1 2.428571

Slot "sdToCont":
  seedbank
1  2.31455

Slot "moveToDiscrete":

Call:  glm(formula = paste("contToDiscrete~",
continuousToDiscreteExplanatoryVariables,
    sep = ""), family = binomial, data = subData)

Coefficients:
(Intercept)         size
 -2.657e+01    4.916e-16

Degrees of Freedom: 108 Total (i.e. Null);  107 Residual
  (1 observation deleted due to missingness)
Null Deviance:       0
Residual Deviance: 6.324e-10  AIC: 4
```

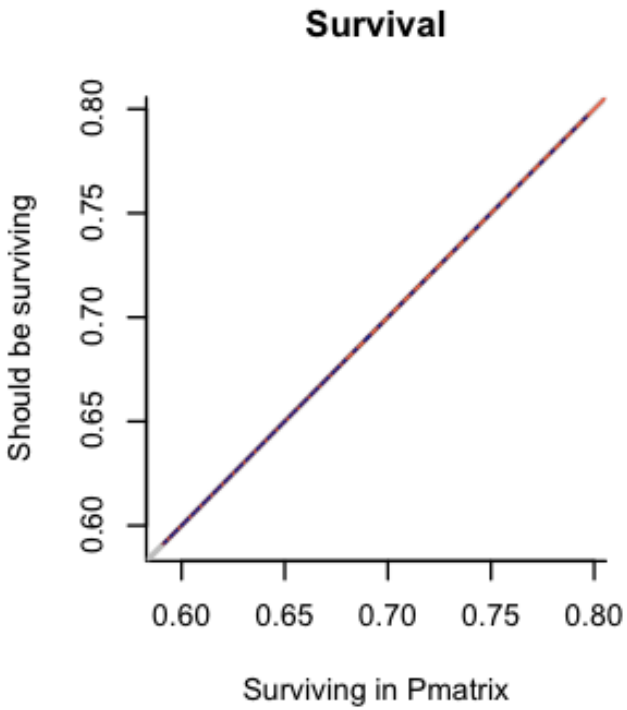
Object construction and model comparison

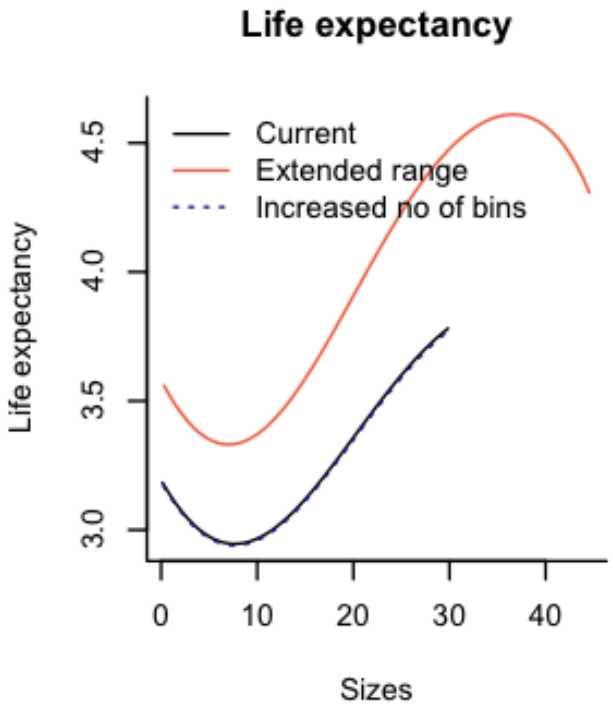
```
#Creating a P matrix reflecting only the continuous part of the mod
  PmatrixContinuousOnly <- makeIPMPmatrix(growObj = go,
                                          survObj = so,
                                          minSize = 0,
                                          maxSize = 30,
                                          nBigMatrix = 100,
                                          correction = "constant")
```

Testing for accidental eviction

```
#...by means of looking at the diagnostics of the P matrix
  diagnosticsPmatrix(PmatrixContinuousOnly,
                     growObj = go,
                     survObj = so,
                     dff = d1,
                     correction = "constant")
```

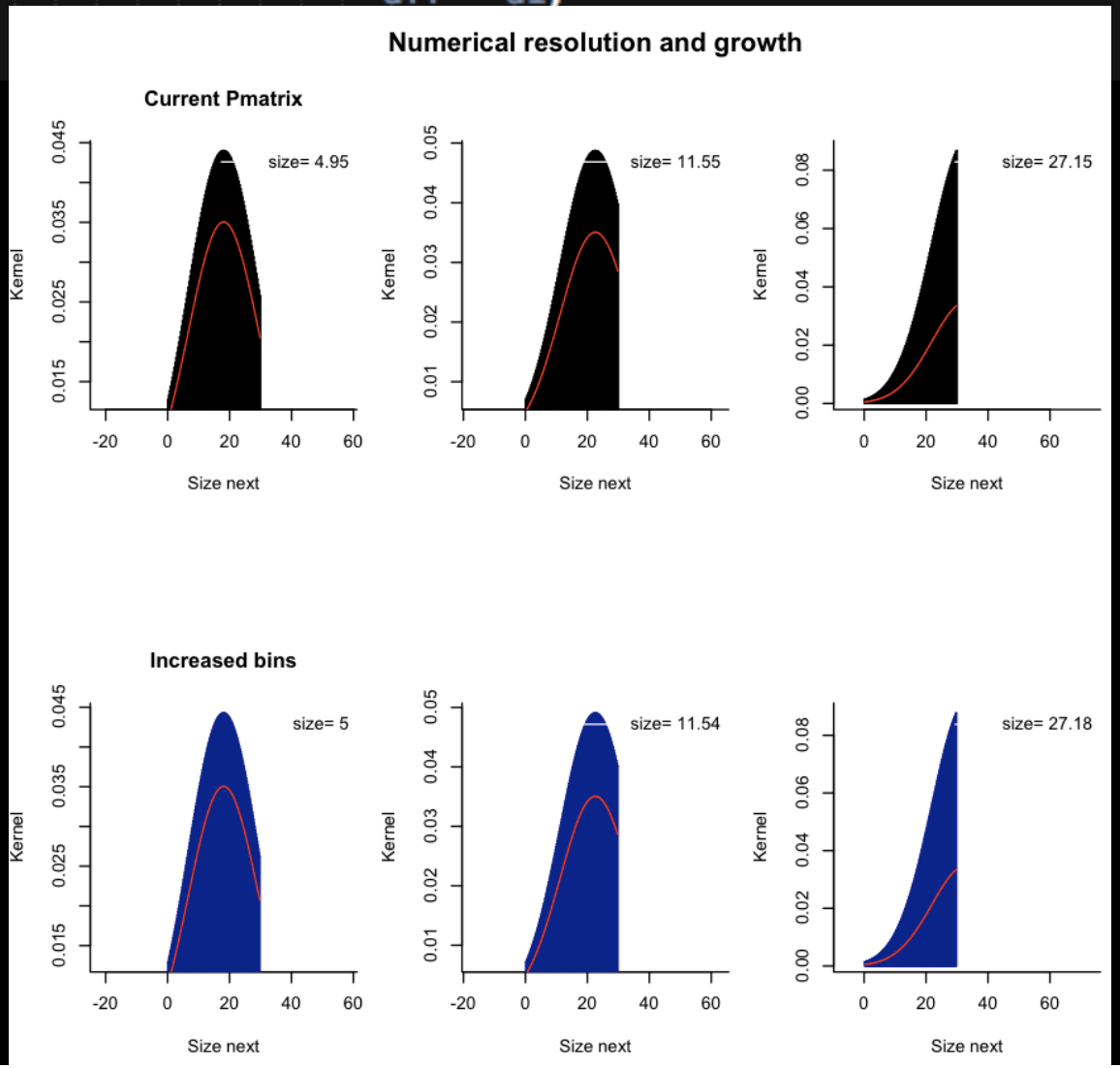






Testing for accidental eviction

```
#...by means of looking at the diagnostics of the P matrix
  diagnosticsPmatrix(PmatrixContinuousOnly,
                     growObj = go,
                     survObj = so,
                     dff = d1,
```



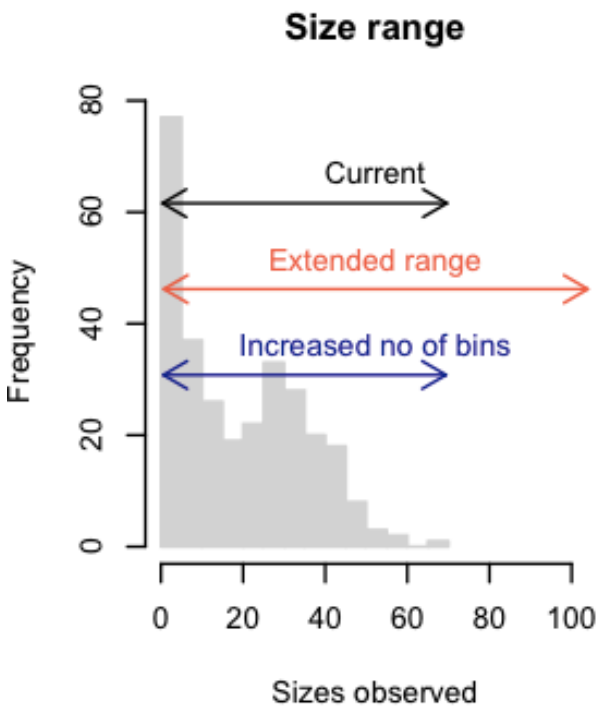
Testing for accidental eviction

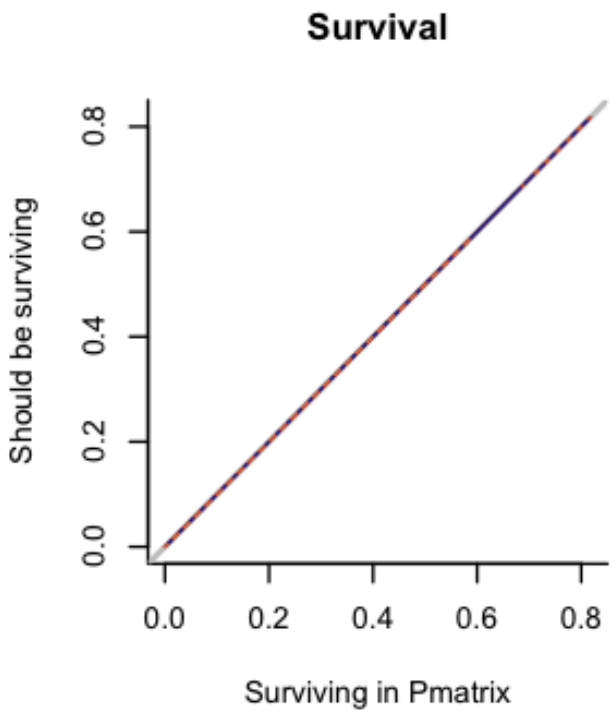
```
Pmatrix <- makeIPMPmatrix(growObj = go,
                          survObj = so,
                          discreteTrans = dto,
                          minSize = 0,
                          maxSize = 70,
                          nBigMatrix = 100,
                          correction = "constant")

PmatrixContinuousOnly <- makeIPMPmatrix(growObj = go,
                                        survObj = so,
                                        minSize = 0,
                                        maxSize = 70,
                                        nBigMatrix = 100,
                                        correction = "constant")
```

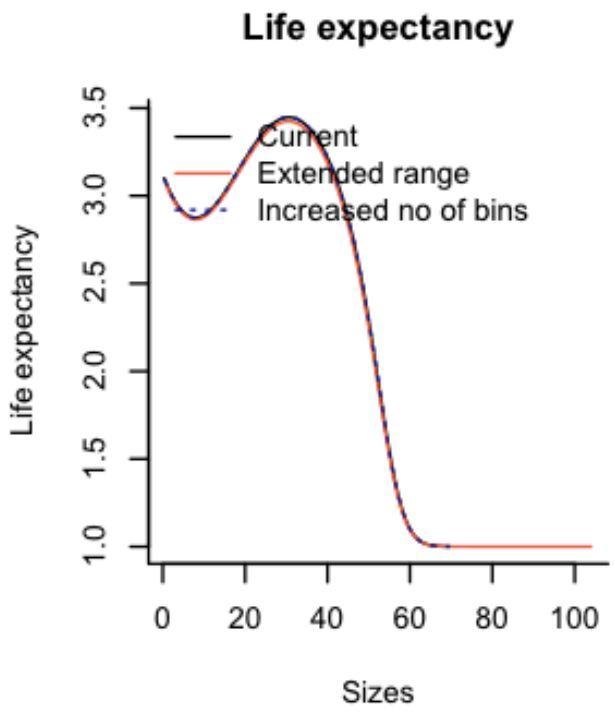
```
#Now the diagnostics should look a lot better:
  diagnosticsPmatrix(PmatrixContinuousOnly,
                     growObj = go,
                     survObj = so,
                     dff = d1,
                     correction = "constant")
```

Testing for accidental eviction

```
#Now the diagnostics should look a lot better:
  diagnosticsPmatrix(PmatrixContinuousOnly,
                     growObj = go,
                     survObj = so,
                     dff = d1,
                     correction = "constant")
```

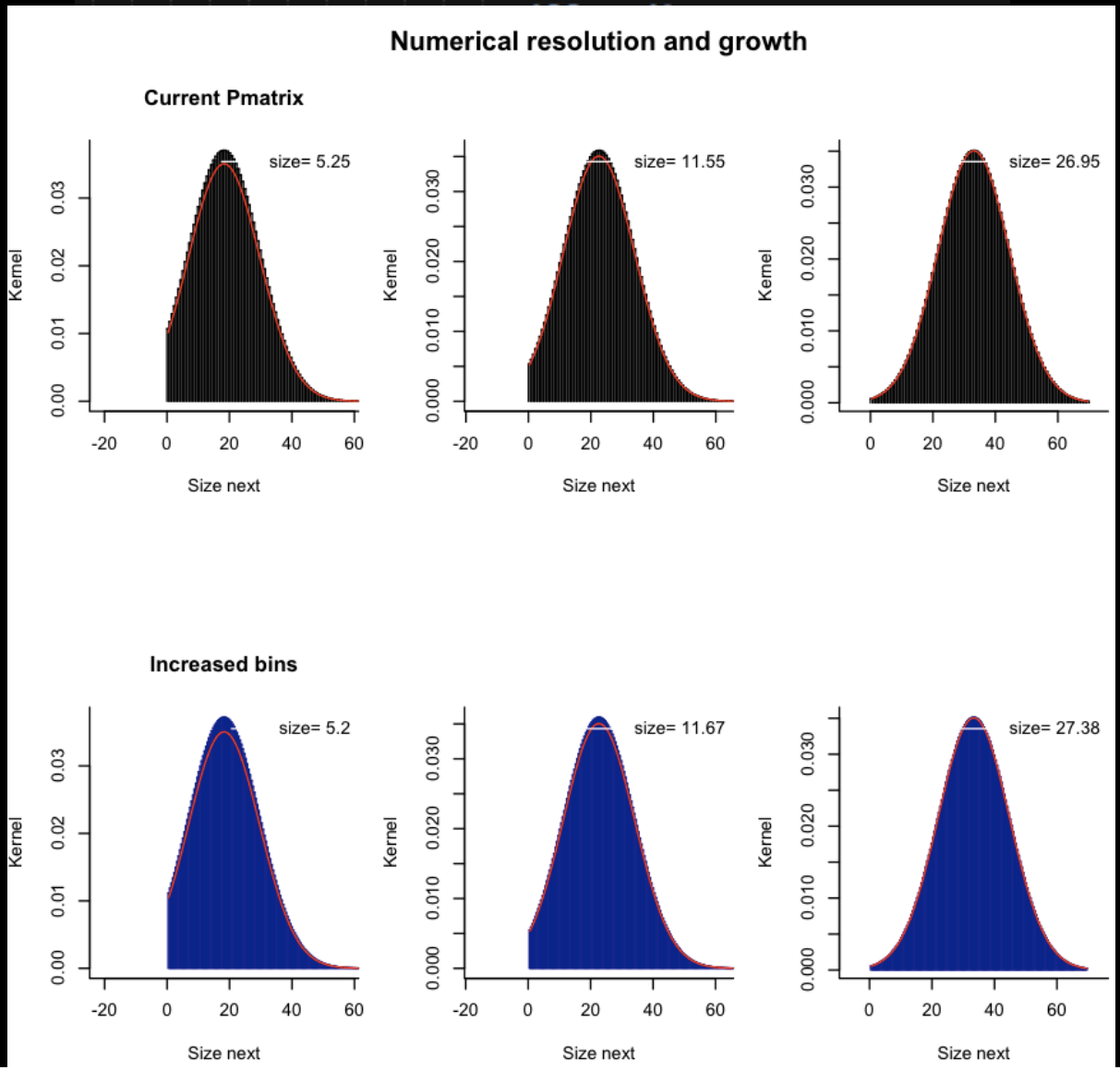






Testing for accidental eviction

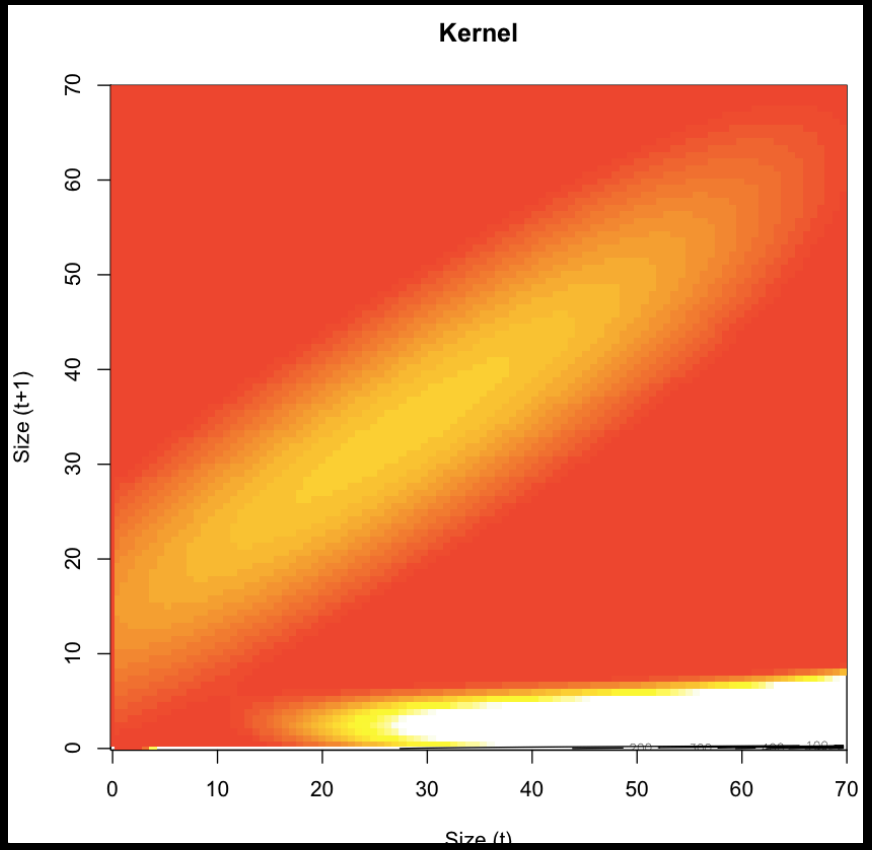
```
#Now the diagnostics should look a lot better:
diagnosticsPmatrix(PmatrixContinuousOnly,
                   growObj = go,
                   survObj = so,
```

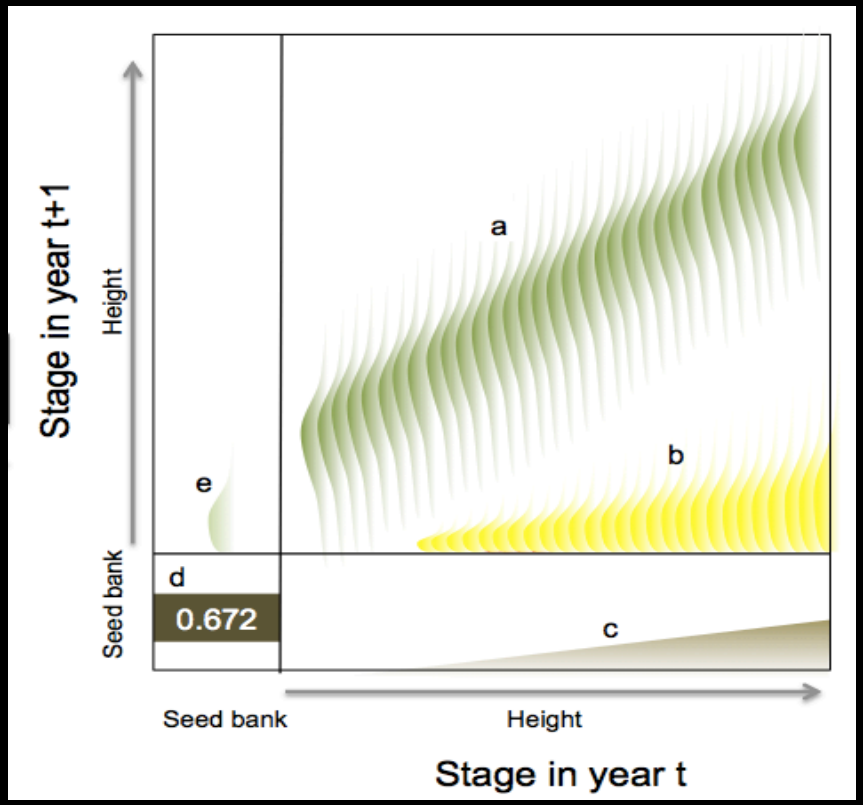


Putting it all together

```
#Forming the IPM as a result of adding the P and F matrices
  IPM <- Pmatrix + Fmatrix
```

```
#Visualizing the IPM kernel via the function IPMpack built-in function "contourPlot", which creates
  image(c(0, Pmatrix@meshpoints),c(0, Pmatrix@meshpoints),t(IPM),
        xlab="Size (t)",ylab="Size (t+1)",col=heat.colors(30), main="Kernel",zlim=c(0,0.03))
  contour(c(0, Pmatrix@meshpoints),c(0, Pmatrix@meshpoints),t(IPM), add = TRUE, drawlabels = TRUE)
```



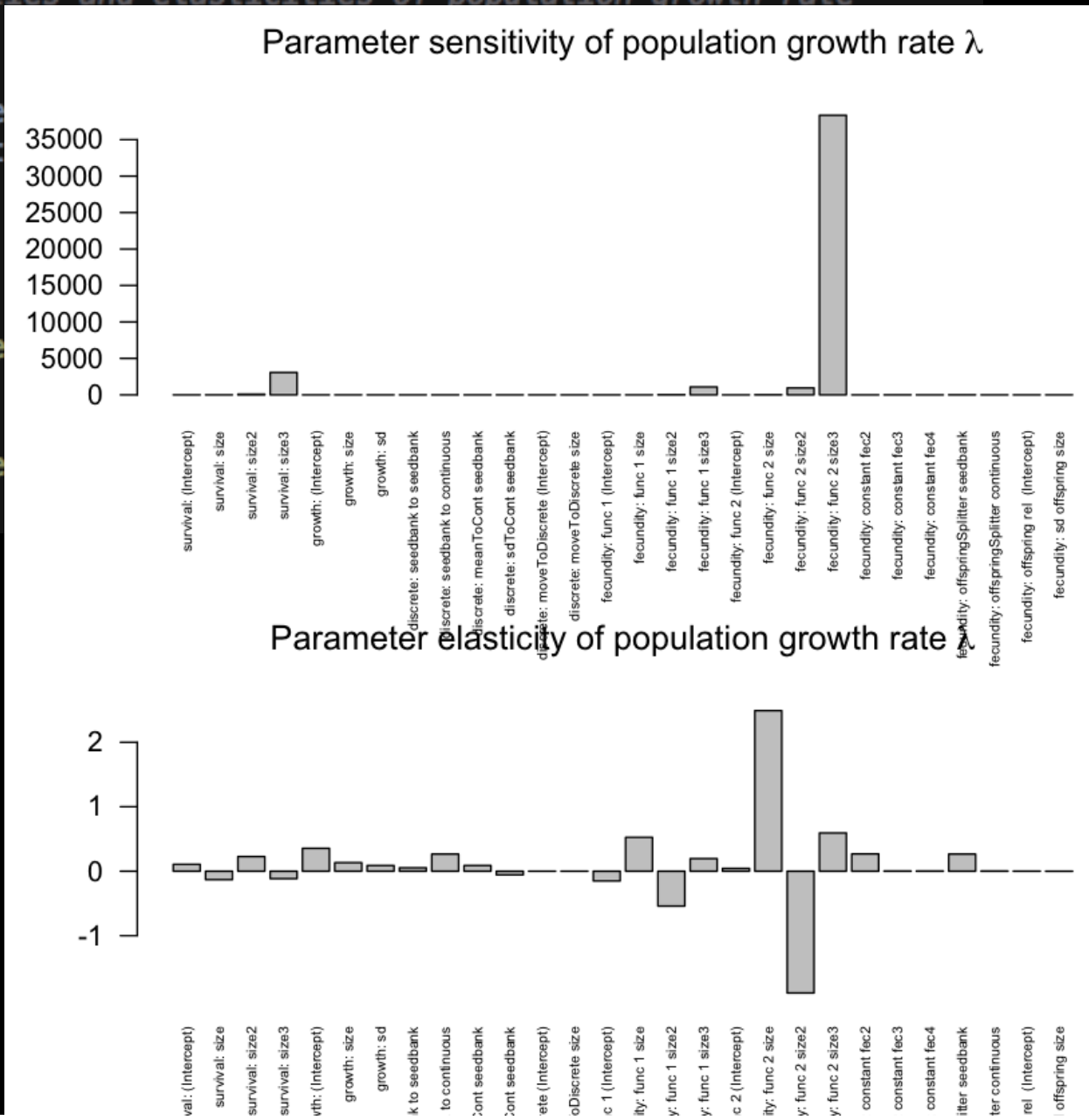


Perturbations

```
#Parameter level sensitivities and elasticities of population growth rate

res1 <- sensParams(growObj = go, survObj = so, fecObj = fo,
  nBigMatrix = 100, minSize = 0, maxSize = 70,
  discreteTrans = dto, delta = 1e-04,
  response="lambda")

par(mfrow = c(2, 1), bty = "l", pty = "m")
barplot(res1$sens,
main = expression("Parameter sensitivity of population growth rate "* lambda),
las = 2, cex.names = 0.5)
barplot(res1$elas,
main = expression("Parameter elasticity of population growth rate "* lambda),
las = 2, cex.names = 0.5)
```

Perturbations

```
#Parameter level sensitivities and elasticities of population growth rate

res1 <- sensParams(growObj
  nBigMatrix = 100, minSize
  discreteTrans = dto, delt
  response="lambda")

par(mfrow = c(2, 1), bty =
barplot(res1$sens,
main = expression("Paramete
las = 2, cex.names = 0.5)
barplot(res1$elas,
main = expression("Paramete
las = 2, cex.names = 0.5)
```

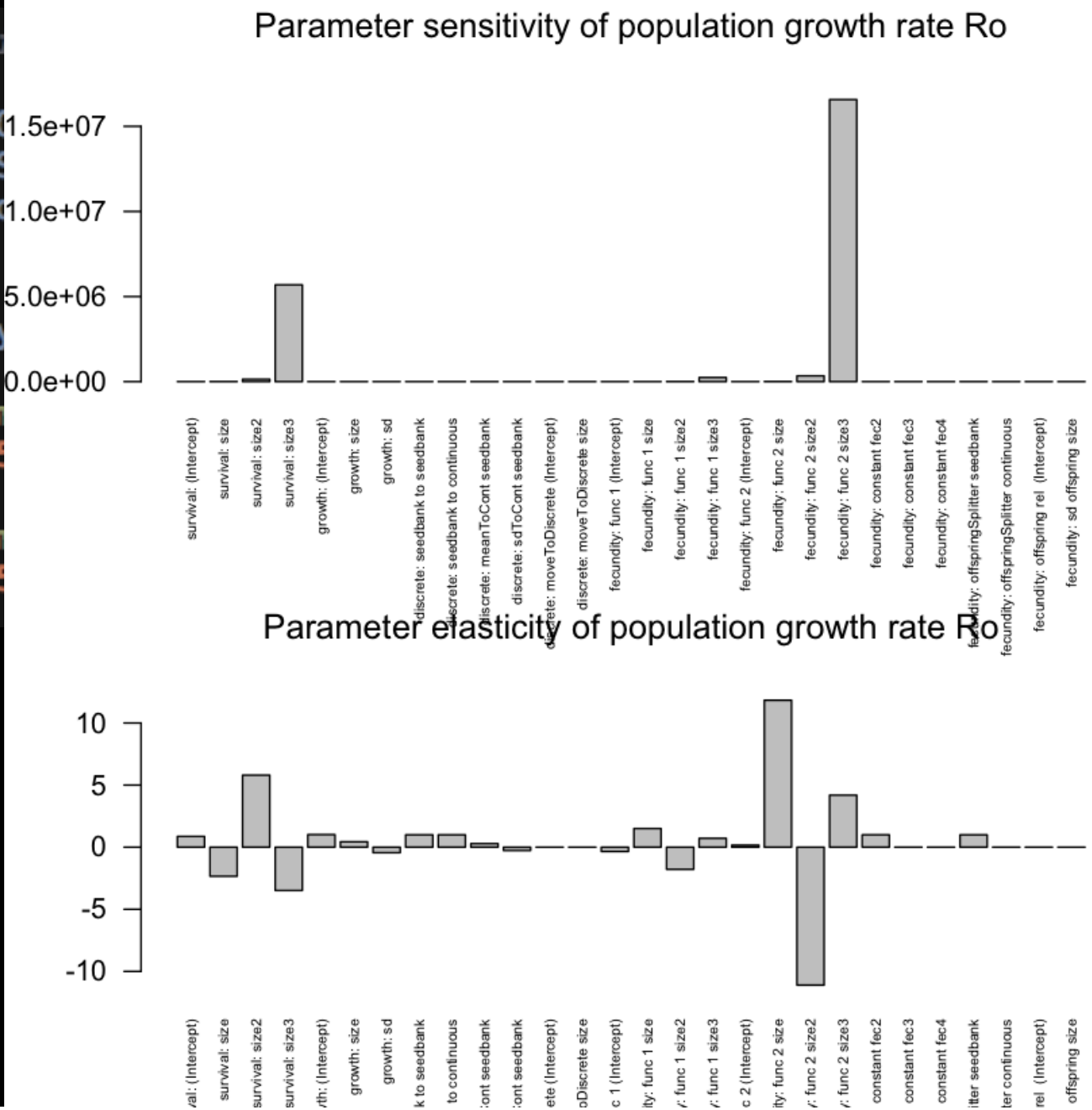


Perturbations

```
#Parameter level sensitivities and elasticities of net reproductive output

res2 <- sensParams(growObj = go, survObj = so, fecObj = fo,
  nBigMatrix = 100, minSize = 0, maxSize = 70,
  discreteTrans = dto, delta = 1e-04,
  response="R0")

par(mfrow = c(2, 1), bty = "l", pty = "m")
barplot(res2$sens,
main = expression("Parameter sensitivity of population growth rate Ro"),
las = 2, cex.names = 0.5)
barplot(res2$elas,
main = expression("Parameter elasticity of population growth rate Ro"),
las = 2, cex.names = 0.5)
```

Perturbations

```
#Parameter level sensiti

res2 <- sensParams(grow
  nBigMatrix = 100, minS
  discreteTrans = dto, c
  response="R0")

par(mfrow = c(2, 1), bty
barplot(res2$sens,
main = expression("Paran
las = 2, cex.names = 0.5
barplot(res2$elas,
main = expression("Paran
las = 2, cex.names = 0.5
```



Acknowledgements

Pedro Quintana-Ascencio
University of Central Florida

Eric Menges
Archbold Biological Station

MAX PLANCK INSTITUTE
FOR DEMOGRAPHIC
RESEARCH