IPMpack:
an R package for Integral Projection Models

C. Jessica E. Metcalf
Sean. M. McMahon
Rob Salguero-Gomez
Eelke Jongejans

Workflow for IPMpack

Import data

Data enter as a ***data frame*** with set column names

```
> library(IPMpack)
Loading required package: Matrix
Loading required package: lattice
Loading required package: MASS
Loading required package: nlme
Loading required package: mvtnorm
> dff<-generateData()
> head(dff)
      size sizeNext surv covariate covariateNext       fec      stage  stageNext
1 4.709132 4.477829    0         1             0 0.0000000 continuous continuous
2 7.381911 7.788379    1         0             0 0.1152413 continuous continuous
3 1.572960 1.933876    0         1             0 0.0000000 continuous continuous
4 4.302018 4.866706    1         0             1 0.0000000 continuous continuous
5 5.142006 4.675990    0         1             0 0.0000000 continuous continuous
6 5.116661 5.905220    0         0             1 8.5576014 continuous continuous
>
```

minimum requirement for a Tmatrix

```
> library(IPMpack)
Loading required package: Matrix
Loading required package: lattice
Loading required package: MASS
Loading required package: nlme
Loading required package: mvtnorm
> dff<-generateData()
> head(dff)
      size sizeNext surv covariate covariateNext       fec      stage  stageNext
1 4.709132 4.477829    0         1              0 0.0000000 continuous continuous
2 7.381911 7.788379    1         0              0 0.1152413 continuous continuous
3 1.572960 1.933876    0         1              0 0.0000000 continuous continuous
4 4.302018 4.866706    1         0              1 0.0000000 continuous continuous
5 5.142006 4.675990    0         1              0 0.0000000 continuous continuous
6 5.116661 5.905220    0         0              1 8.5576014 continuous continuous
>
```

minimum requirement for a Tmatrix

only necessary if you have a mixture of discrete and continuous stages (defaults to all continuous)

```
> library(IPMpack)
Loading required package: Matrix
Loading required package: lattice
Loading required package: MASS
Loading required package: nlme
Loading required package: mvtnorm
> dff<-generateData()
> head(dff)
      size sizeNext surv covariate covariateNext       fec      stage  stageNext
1 4.709132 4.477829    0         1             0 0.0000000 continuous continuous
2 7.381911 7.788379    1         0             0 0.1152413 continuous continuous
3 1.572960 1.933876    0         1             0 0.0000000 continuous continuous
4 4.302018 4.866706    1         0             1 0.0000000 continuous continuous
5 5.142006 4.675990    0         1             0 0.0000000 continuous continuous
6 5.116661 5.905220    0         0             1 8.5576014 continuous continuous
>
```

minimum requirement for a Tmatrix

only necessary if you have a mixture of discrete and continuous stages (defaults to all continuous)

```
> dff<-generateDataDiscrete()
> head(dff)
      size sizeNext surv      fec      stage  stageNext number
1       NA       NA    1 3.763959    dormant    dormant      1
2 5.975158 8.118134    1 0.000000 continuous continuous      1
3 9.093308       NA    0 4.603087 continuous       dead      1
4 7.410467 5.777279    1 0.000000 continuous continuous      1
5 5.627206       NA    0 0.000000 continuous       dead      1
6 5.133702 3.731515    1 0.000000 continuous continuous      1
>
```

only necessary for models with transition between discrete stages

```
> library(IPMpack)
Loading required package: Matrix
Loading required package: lattice
Loading required package: MASS
Loading required package: nlme
Loading required package: mvtnorm
> dff<-generateData()
> head(dff)
      size sizeNext surv covariate covariateNext       fec      stage  stageNext
1 4.709132 4.477829    0         1             0 0.0000000 continuous continuous
2 7.381911 7.788379    1         0             0 0.1152413 continuous continuous
3 1.572960 1.933876    0         1             0 0.0000000 continuous continuous
4 4.302018 4.866706    1         0             1 0.0000000 continuous continuous
5 5.142006 4.675990    0         1             0 0.0000000 continuous continuous
6 5.116661 5.905220    0         0             1 8.5576014 continuous continuous
>
```

minimum requirement for a Tmatrix

only necessary if you have a mixture of discrete and continuous stages (defaults to all continuous)

```
> dff<-generateDataDiscrete()
> head(dff)
      size sizeNext surv      fec      stage  stageNext number
1       NA       NA    1 3.763959    dormant    dormant      1
2 5.975158 8.118134    1 0.000000 continuous continuous      1
3 9.093308       NA    0 4.603087 continuous       dead      1
4 7.410467 5.777279    1 0.000000 continuous continuous      1
5 5.627206       NA    0 0.000000 continuous       dead      1
6 5.133702 3.731515    1 0.000000 continuous continuous      1
>
```

only necessary for models with transition between discrete stages

```
> head(dff)
      size sizeNext surv covariate covariateNext      fec      stage  stageNext offspringNext
1 5.299204 5.864362    1         0             0 0.000000 continuous continuous         <NA>
2 5.510951 5.167714    0         1             0 6.345835 continuous continuous       sexual
3 4.279321 2.626327    0         1             0 3.379862 continuous continuous         <NA>
4 5.026918 7.419591    1         1             0 0.000000 continuous continuous       sexual
5 5.103107 6.078352    1         1             0 0.000000 continuous continuous         <NA>
6 5.422322 5.951682    1         1             0 8.198216 continuous continuous         <NA>
```

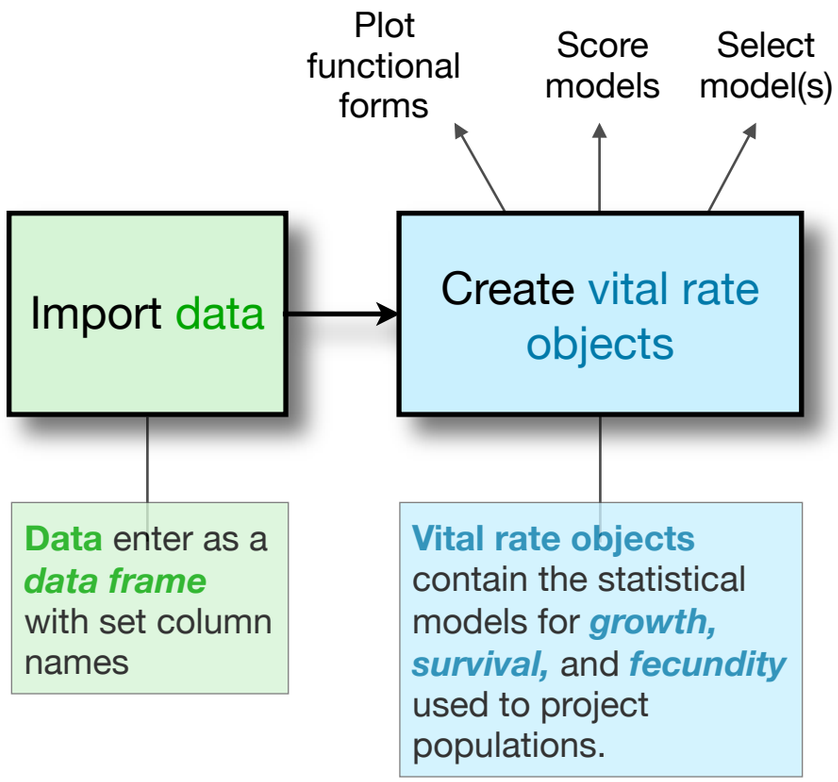
only necessary for models with dependence of offspring size on adult size

Workflow for IPMpack

Import data

Data enter as a ***data frame*** with set column names

Workflow for IPMpack



A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula ‘size+size2’, etc.

A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula ‘size+size2’, etc.

handling of variance

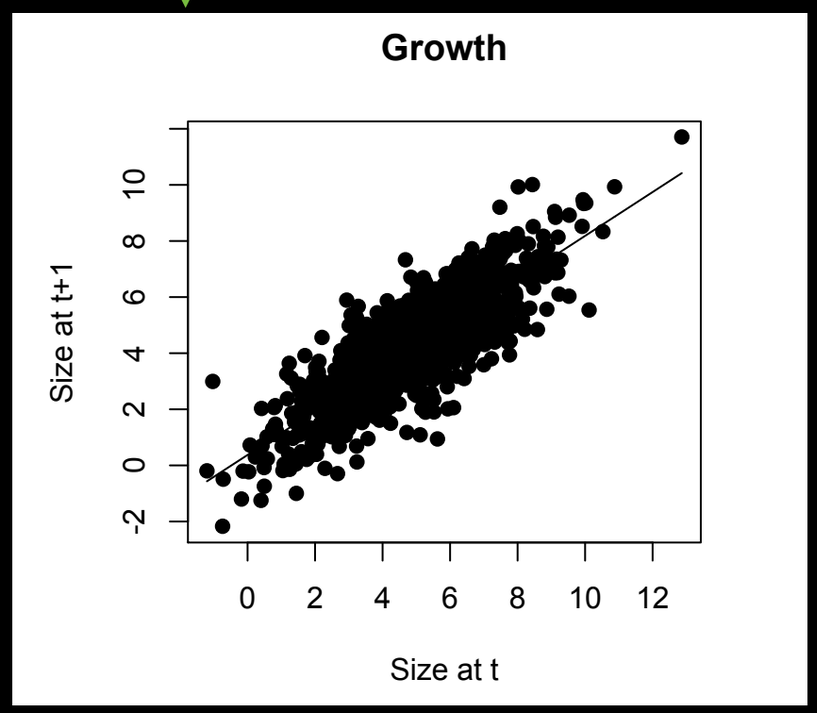
A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula 'size+size2', etc.

handling of variance



A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula ‘size+size2’, etc.

handling of variance

A survival object

```
>
> sv1 <- makeSurvObj(dataf=dff  Formula=surv~size)
> picSurv(dff,sv1)
>
```

data

regression formula ‘size+size2’, etc.

A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula ‘size+size2’, etc.

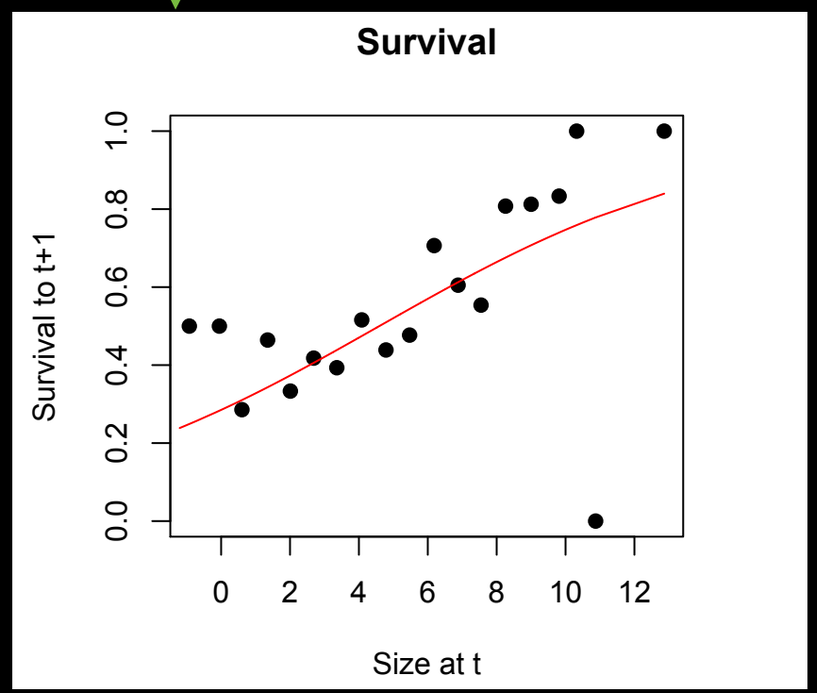
handling of variance

A survival object

```
>
> sv1 <- makeSurvObj(dataf=dff  Formula=surv~size)
> picSurv(dff,sv1)
>
```

data

regression formula ‘size+size2’, etc.



A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula 'size+size2', etc.

handling of variance

A survival object

```
>
> sv1 <- makeSurvObj(dataf=dff  Formula=surv~size)
> picSurv(dff,sv1)
>
```

data

regression formula 'size+size2', etc.

```
>
> fv1 <- makeFecObj(dataf=dff, fecConstants=data.frame(prob.est=0.5),
+       Formula=list(fec~size),Family="gaussian",Transform="log",
+       meanOffspringSize=NA,sdOffspringSize=NA,
+       offspringSplitter=data.frame(continuous=1),
+       offspringTypeRates=data.frame(NA),
+       fecByDiscrete=data.frame(NA),
+       offspringSizeExplanatoryVariables="1")
>
```

A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula 'size+size2', etc.

handling of variance

A survival object

```
>
> sv1 <- makeSurvObj(dataf=dff  Formula=surv~size)
> picSurv(dff,sv1)
>
```

data

regression formula 'size+size2', etc.

```
>
> fv1 <- makeFecObj(dataf=dff, fecConstants=data.frame(prob.est=0.5),
+       Formula=list(fec~size),Family="gaussian",Transform="log",
+       meanOffspringSize=NA,sdOffspringSize=NA,
+       offspringSplitter=data.frame(continuous=1),
+       offspringTypeRates=data.frame(NA),
+       fecByDiscrete=data.frame(NA),
+       offspringSizeExplanatoryVariables="1")
>
```

A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula 'size+size2', etc.

handling of variance

A survival object

```
>
> sv1 <- makeSurvObj(dataf=dff  Formula=surv~size)
> picSurv(dff,sv1)
>
```

data

regression formula 'size+size2', etc.

constant multipliers (seed establishment probability)

```
>
> fv1 <- makeFecObj(dataf=dff, fecConstants=data.frame(prob.est=0.5),
+       Formula=list(fec~size),Family="gaussian",Transform="log",
+       meanOffspringSize=NA,sdOffspringSize=NA,
+       offspringSplitter=data.frame(continuous=1),
+       offspringTypeRates=data.frame(NA),
+       fecByDiscrete=data.frame(NA),
+       offspringSizeExplanatoryVariables="1")
>
```

A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula 'size+size2', etc.

handling of variance

A survival object

```
>
> sv1 <- makeSurvObj(dataf=dff  Formula=surv~size)
> picSurv(dff,sv1)
>
```

data

regression formula 'size+size2', etc.

constant multipliers (seed establishment probability)

```
>
> fv1 <- makeFecObj(dataf=dff, fecConstants=data.frame(prob.est=0.5),
+       Formula=list(fec~size),Family="gaussian",Transform="log",
+       meanOffspringSize=NA,sdOffspringSize=NA,
+       offspringSplitter=data.frame(continuous=1),
+       offspringTypeRates=data.frame(NA),
+       fecByDiscrete=data.frame(NA),
+       offspringSizeExplanatoryVariables="1")
>
```

formula, type of distribution (can have several!)

A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula 'size+size2', etc.

handling of variance

A survival object

```
>
> sv1 <- makeSurvObj(dataf=dff  Formula=surv~size)
> picSurv(dff,sv1)
>
```

data

regression formula 'size+size2', etc.

constant multipliers (seed establishment probability)

```
>
> fv1 <- makeFecObj(dataf=dff, fecConstants=data.frame(prob.est=0.5),
+       Formula=list(fec~size),Family="gaussian",Transform="log",
+       meanOffspringSize=NA,sdOffspringSize=NA,
+       offspringSplitter=data.frame(continuous=1),
+       offspringTypeRates=data.frame(NA),
+       fecByDiscrete=data.frame(NA),
+       offspringSizeExplanatoryVariables="1")
>
```

Transform applied to fertility measure (can be several)

formula, type of distribution (can have several!)

A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula 'size+size2', etc.

handling of variance

A survival object

```
>
> sv1 <- makeSurvObj(dataf=dff  Formula=surv~size)
> picSurv(dff,sv1)
>
```

data

regression formula 'size+size2', etc.

constant multipliers (seed establishment probability)

```
>
> fv1 <- makeFecObj(dataf=dff, fecConstants=data.frame(prob.est=0.5),
+       Formula=list(fec~size),Family="gaussian",Transform="log",
+       meanOffspringSize=NA,sdOffspringSize=NA,
+       offspringSplitter=data.frame(continuous=1),
+       offspringTypeRates=data.frame(NA),
+       fecByDiscrete=data.frame(NA),
+       offspringSizeExplanatoryVariables="1")
>
```

Transform applied to fertility measure (can be several)

Offspring sizes (NAs means the function will the formula in offspringSizeExplanatoryVariables

formula, type of distribution (can have several!)

A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula 'size+size2', etc.

handling of variance

A survival object

```
>
> sv1 <- makeSurvObj(dataf=dff  Formula=surv~size)
> picSurv(dff,sv1)
>
```

data

regression formula 'size+size2', etc.

constant multipliers (seed establishment probability)

```
>
> fv1 <- makeFecObj(dataf=dff, fecConstants=data.frame(prob.est=0.5),
+       Formula=list(fec~size),Family="gaussian",Transform="log",
+       meanOffspringSize=NA,sdOffspringSize=NA,
+       offspringSplitter=data.frame(continuous=1),
+       offspringTypeRates=data.frame(NA),
+       fecByDiscrete=data.frame(NA),
+       offspringSizeExplanatoryVariables="1")
>
```

Transform applied to fertility measure (can be several)

Offspring sizes (NAs means the function will the formula in offspringSizeExplanatoryVariables

formula, type of distribution (can have several!)

Fraction fecundity for each offspring class

A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula 'size+size2', etc.

handling of variance

A survival object

```
>
> sv1 <- makeSurvObj(dataf=dff  Formula=surv~size)
> picSurv(dff,sv1)
>
```

data

regression formula 'size+size2', etc.

constant multipliers (seed establishment probability)

```
>
> fv1 <- makeFecObj(dataf=dff, fecConstants=data.frame(prob.est=0.5),
+       Formula=list(fec~size),Family="gaussian",Transform="log",
+       meanOffspringSize=NA,sdOffspringSize=NA,
+       offspringSplitter=data.frame(continuous=1),
+       offspringTypeRates=data.frame(NA),
+       fecByDiscrete=data.frame(NA),
+       offspringSizeExplanatoryVariables="1")
>
```

Transform applied to fertility measure (can be several)

Offspring sizes (NAs means the function will the formula in offspringSizeExplanatoryVariables

formula, type of distribution (can have several!)

Link fertility relationships and discrete offspring classes

Fraction fecundity for each offspring class

A growth object

```
>
> gr1 <- makeGrowthObj(dataf=dff, Formula=sizeNext~size,regType="constantVar")
> picGrow(dff,gr1)
>
```

data

regression formula 'size+size2', etc.

handling of variance

A survival object

```
>
> sv1 <- makeSurvObj(dataf=dff  Formula=surv~size)
> picSurv(dff,sv1)
>
```

data

regression formula 'size+size2', etc.

constant multipliers (seed establishment probability)

```
>
> fv1 <- makeFecObj(dataf=dff, fecConstants=data.frame(prob.est=0.5),
+       Formula=list(fec~size),Family="gaussian",Transform="log",
+       meanOffspringSize=NA,sdOffspringSize=NA,
+       offspringSplitter=data.frame(continuous=1),
+       offspringTypeRates=data.frame(NA),
+       fecByDiscrete=data.frame(NA),
+       offspringSizeExplanatoryVariables="1")
>
```

Transform applied to fertility measure (can be several)

Offspring sizes (NAs means the function will the formula in offspringSizeExplanatoryVariables

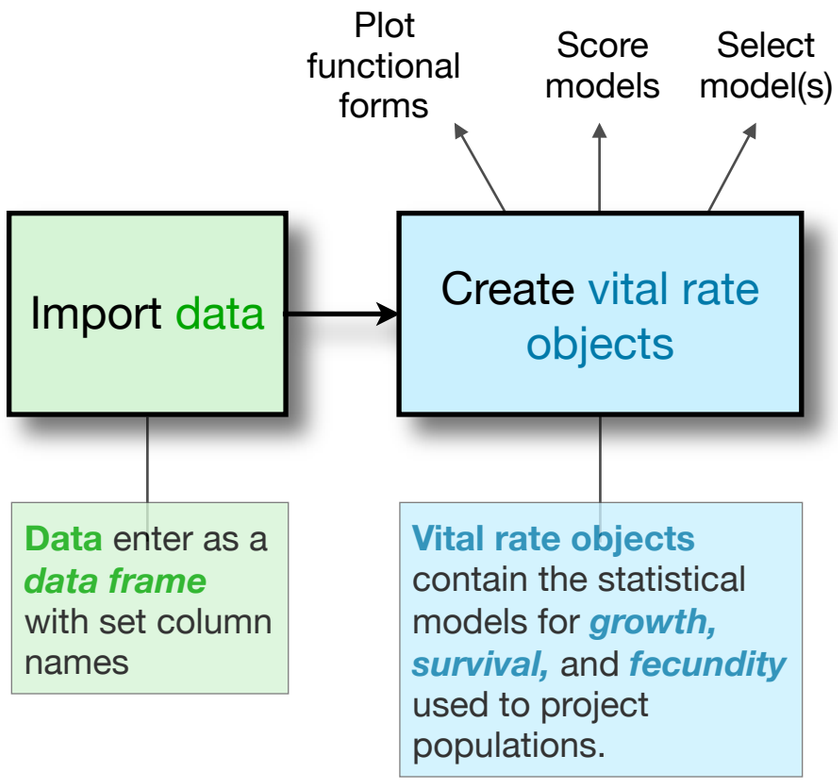
formula, type of distribution (can have several!)

Fec by discrete

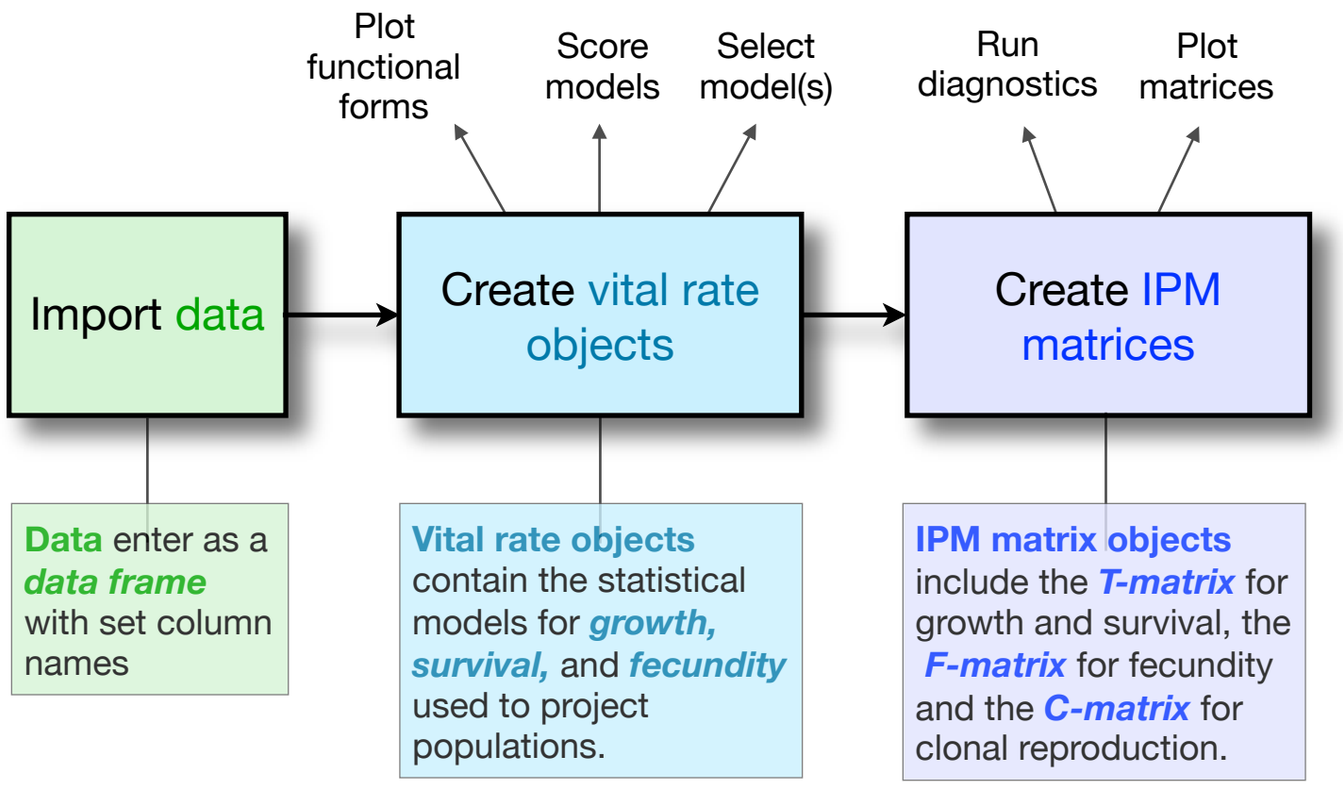
Link fertility relationships and discrete offspring classes

Fraction fecundity for each offspring class

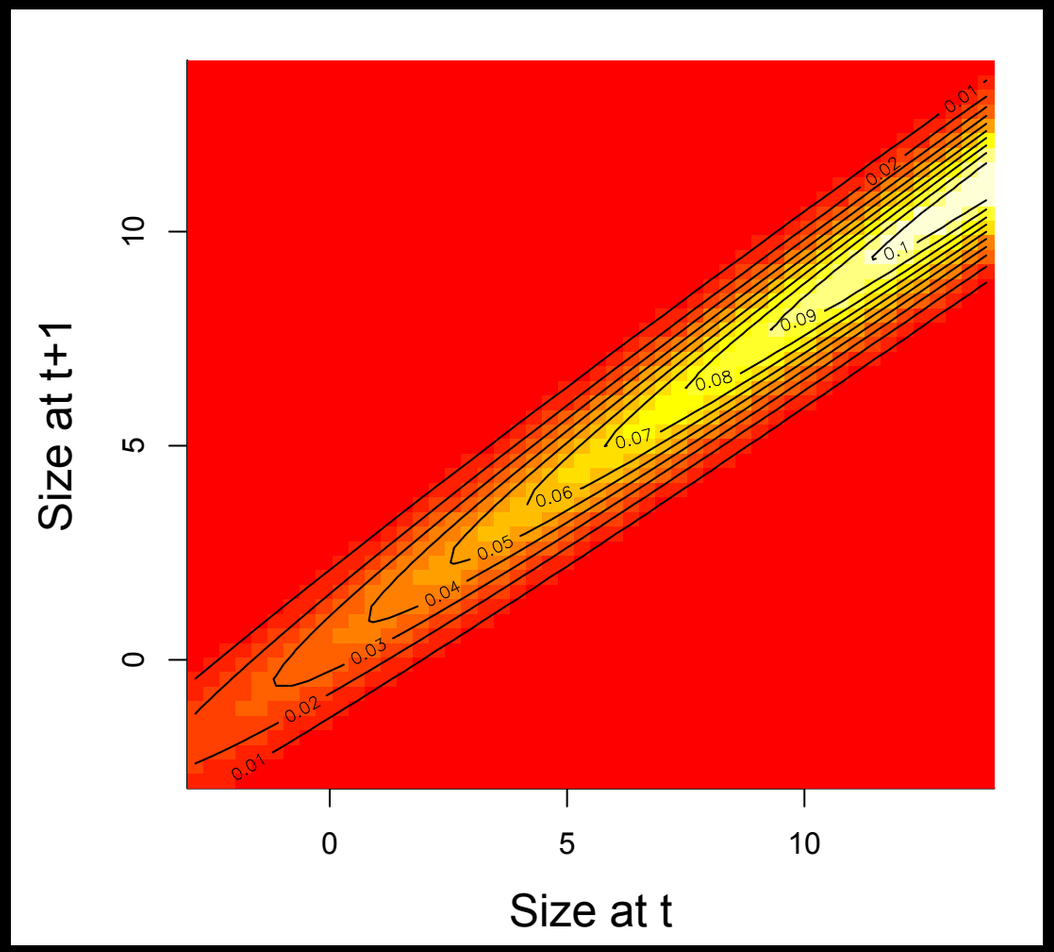
Workflow for IPMpack



Workflow for IPMpack

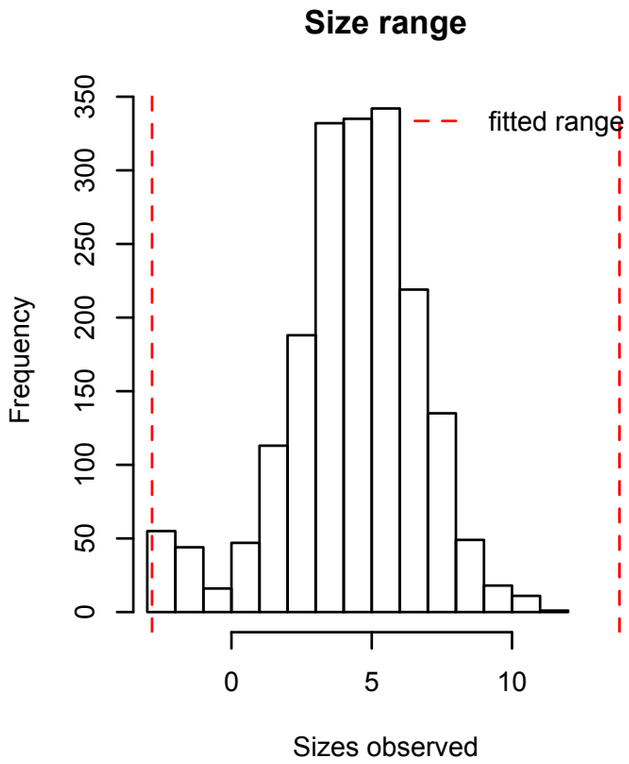


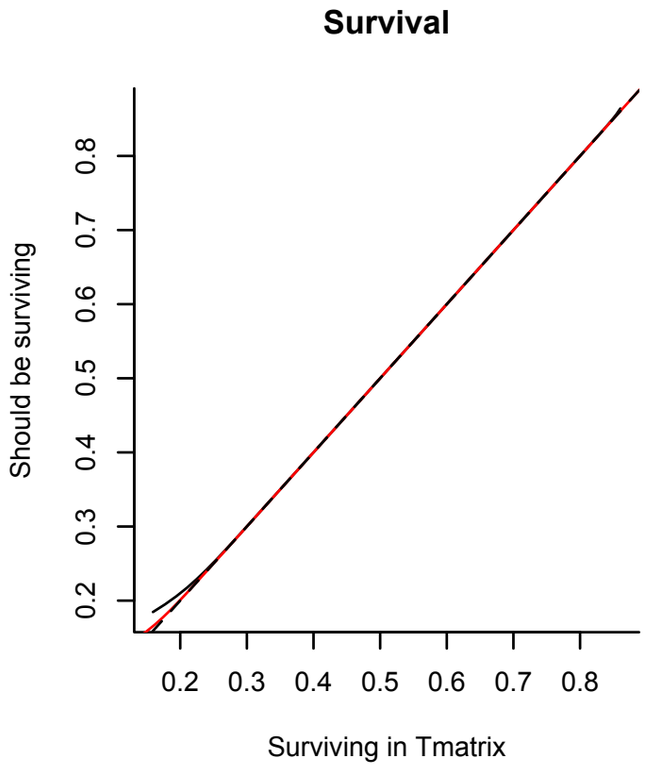
```
>
> Tmat<-createIPMTmatrix(nBigMatrix=50, minSize=-3,maxSize=14,growObj=gr1,survObj=sv1)
>
> image(Tmat@meshpoints,Tmat@meshpoints,t(Tmat),xlab="Size at t", ylab="Size at t+1",cex.lab=1.5)
> contour(Tmat@meshpoints,Tmat@meshpoints,t(Tmat),add=TRUE)
>
```

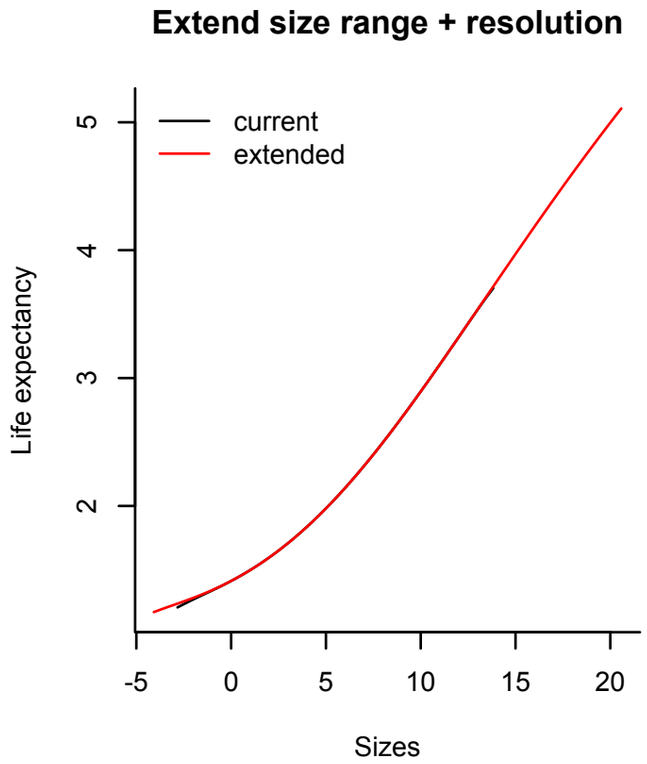


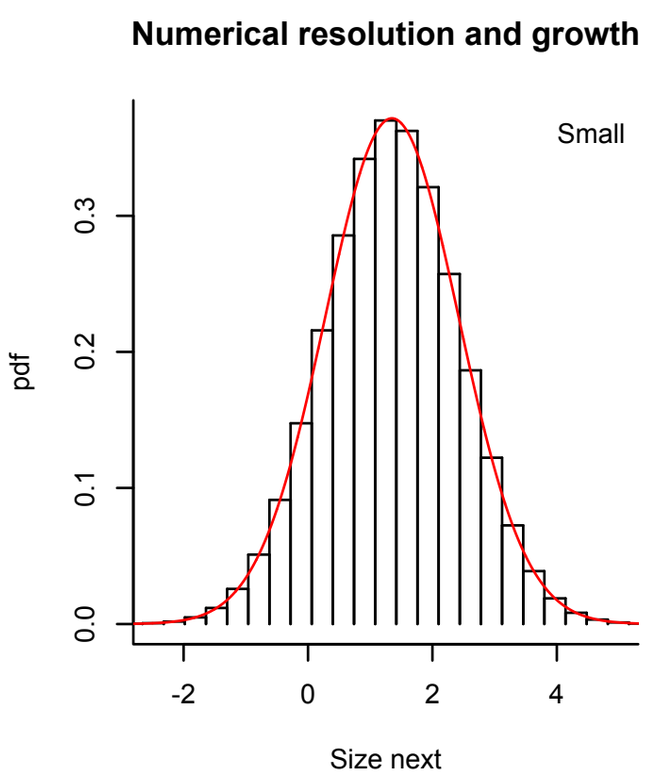

```
>
> diagnosticsTmatrix(Tmatrix=Tmat,growObj=gr1,survObj=sv1,dff=dff)
[1] "Range of Tmatrix is "
[1] 1.304382e-48 1.089820e-01
>
```

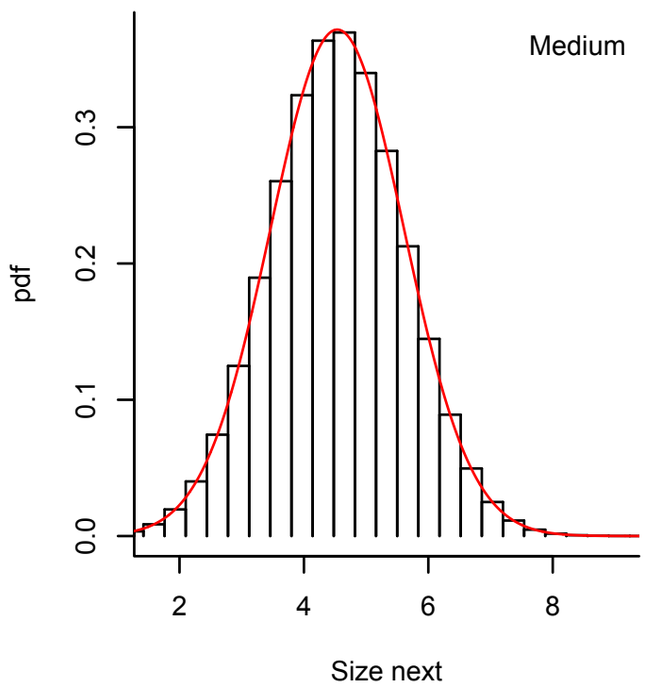
Diagnostics Tmatrix

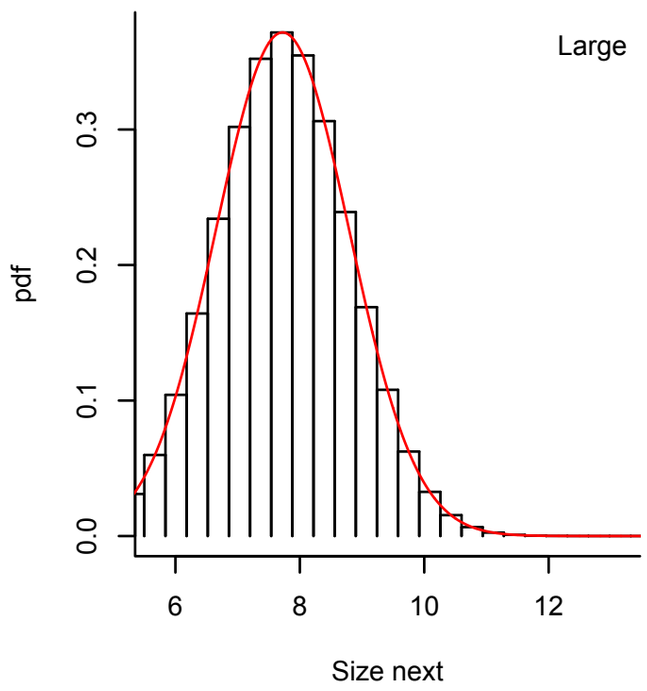




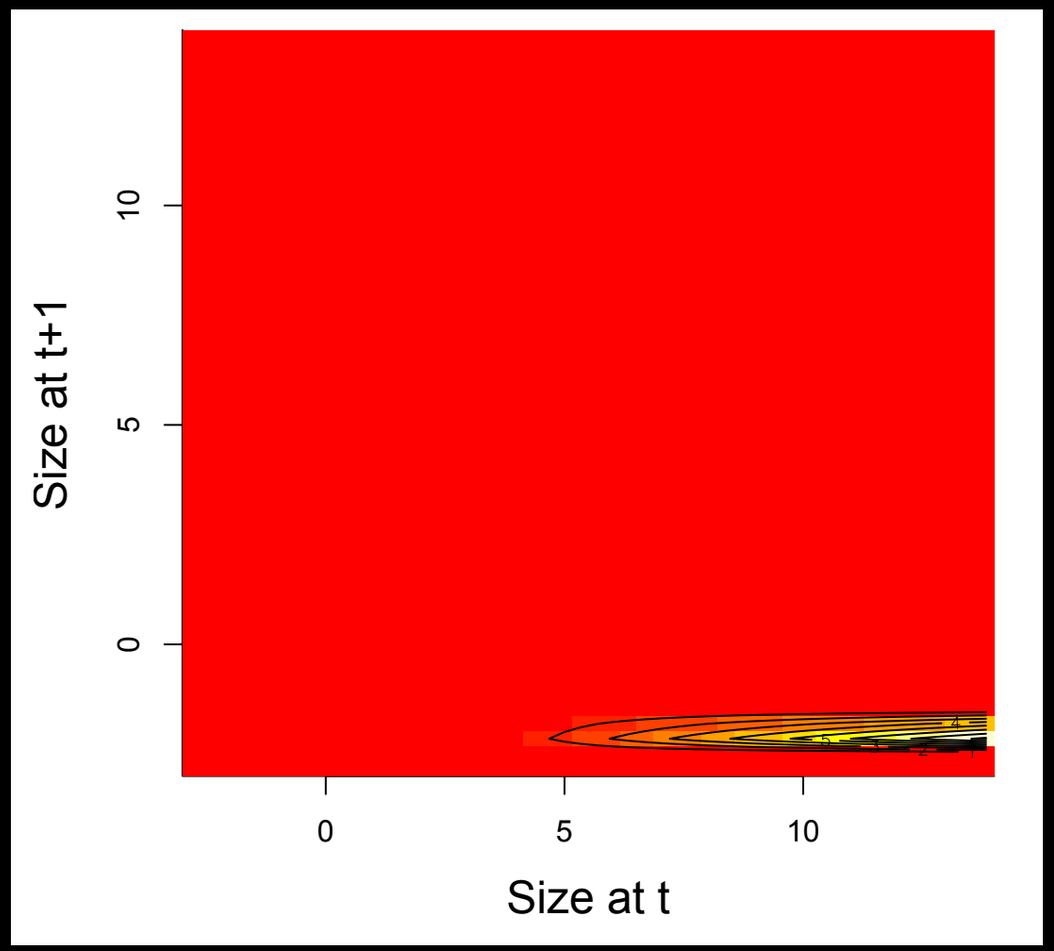






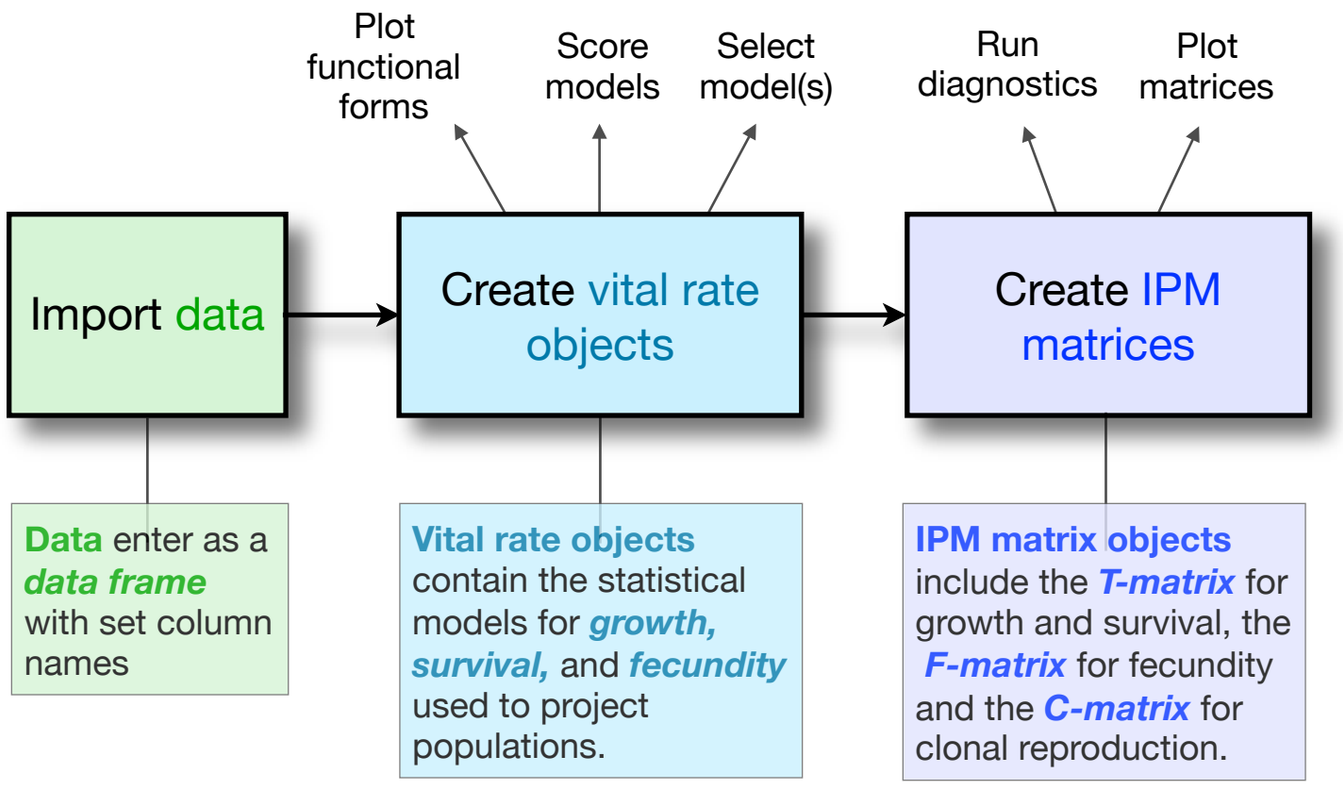


```
>
> Fmat <- createIPMFmatrix(nBigMatrix=50,minSize=-3,maxSize=14,fecObj=fv1)
[1] "Warning: fertility values < 0 exist in matrix, consider transforms. Negative values set to zero"
>
> image(Fmat@meshpoints,Fmat@meshpoints,(t(Fmat)),xlab="Size at t", ylab="Size at t+1",cex.lab=1.5)
> contour(Fmat@meshpoints,Fmat@meshpoints,t(Fmat),add=TRUE)
>
```

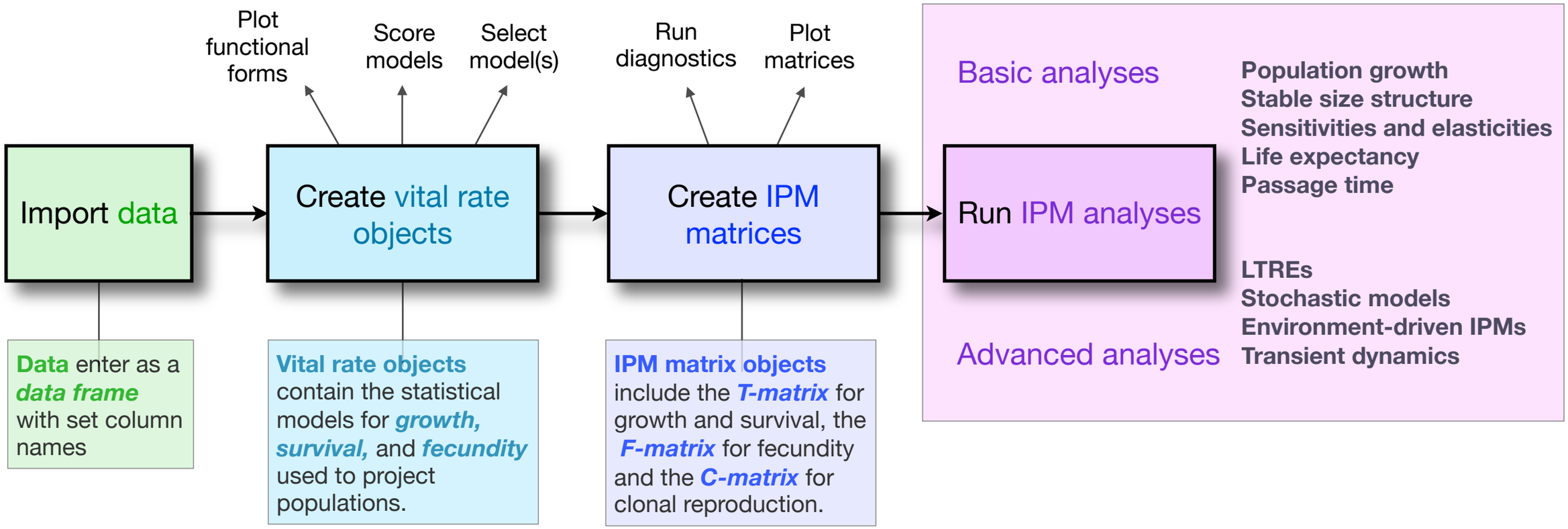



```
>
> IPM <- Tmat + Fmat
>
```

Workflow for IPMpack

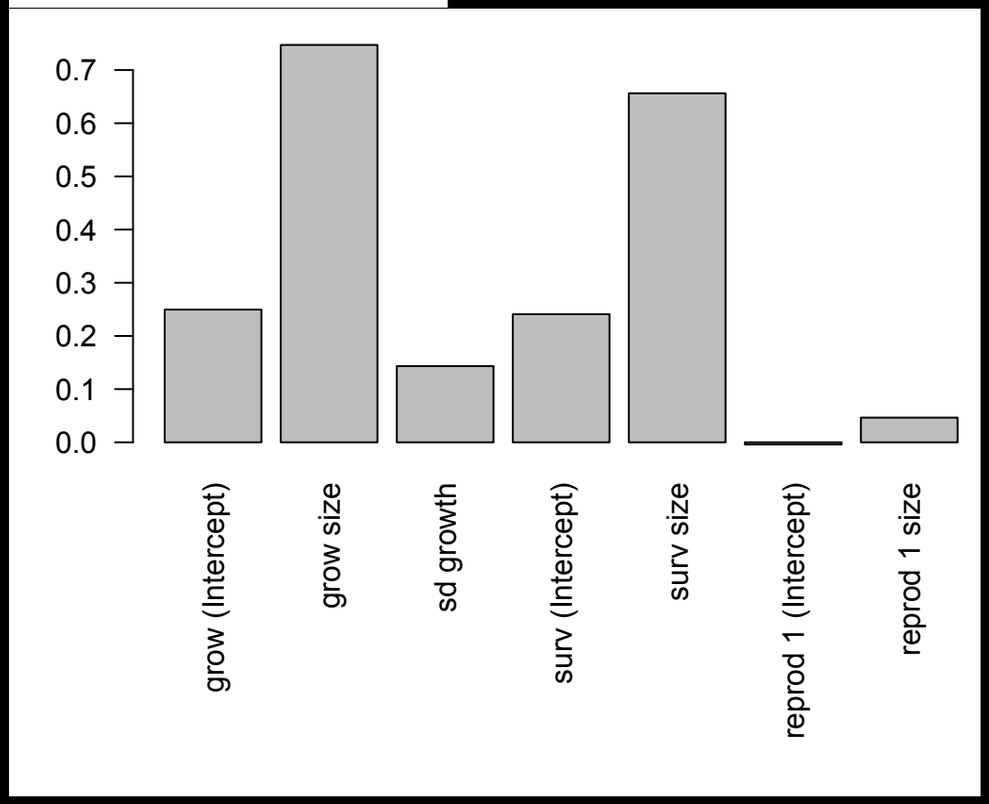


Workflow for IPMpack



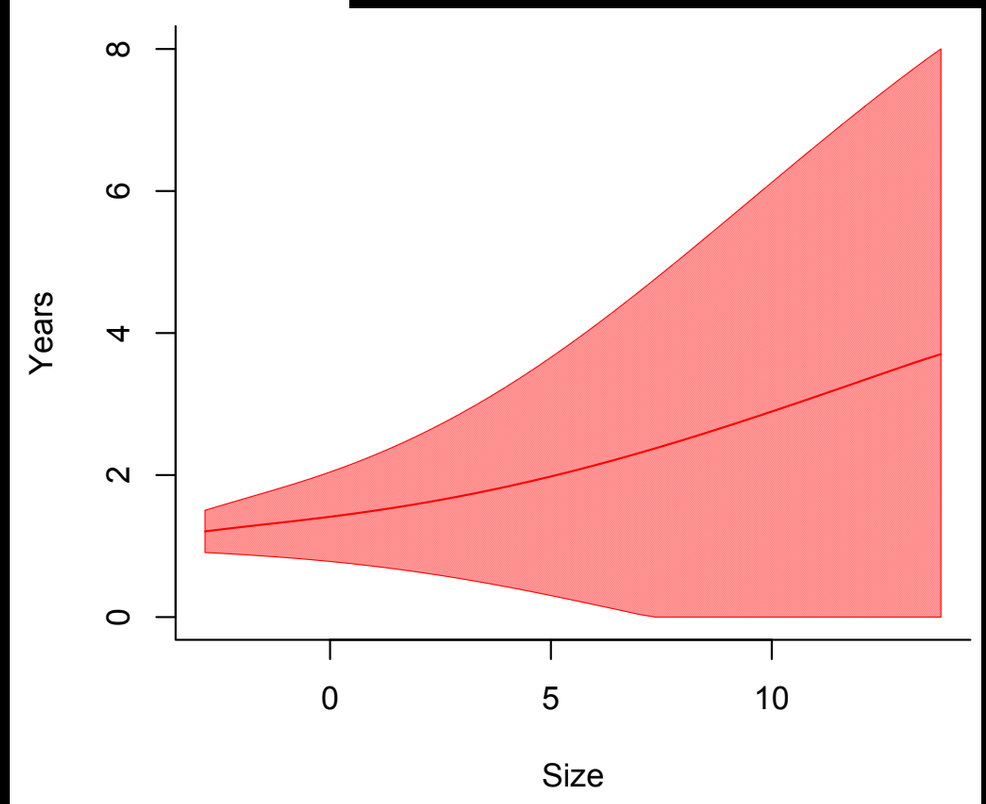
Functions that work only with an IPMpack constructed IPM

Parameter sensitivity



Functions that will work with any matrix

Life expectancy



... and many more....

We want IPMpack to be

- an easy package interface into using IPMs

- NOT a black box. We are trying to design functions so that without knowing all of the code that goes into making the models:
    - assumptions are transparent
    - decisions about what models to use are facilitated
    - implications of different functional forms, etc, are clear

- a handy toolkit of functions even for those who are very comfortable and prefer to make their own IPMs.

Basta





Thank-you!

MAX PLANCK INSTITUTE FOR DEMOGRAPHIC RESEARCH

Smithsonian

University of Connecticut

EXCELLENCE IN SCIENCE

Radboud University Nijmegen

UNIVERSITY OF OXFORD

Imposing your preferred statistical fit

```
> gr1 <- makeGrowthObj(dff,explanatoryVariables="size")
> gr1
An object of class "growthObj"
Slot "fit":

Call:
lm(formula = Formula, data = dataf)

Coefficients:
(Intercept)         size
     0.3444       0.7957


Slot "sd":
[1] 1.097115
```

build a template

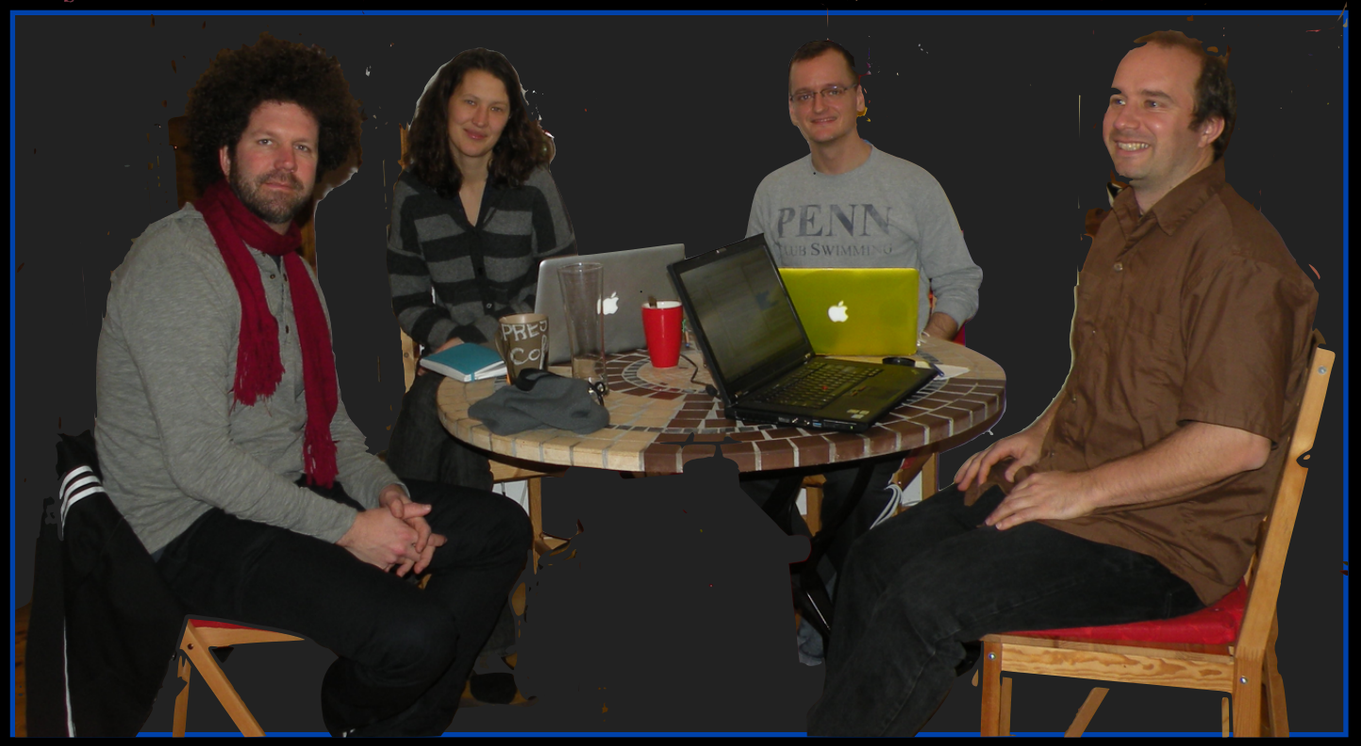
```
> slotNames(gr1)
[1] "fit" "sd"
> gr1@fit$coefficients
(Intercept)        size
  0.3444347   0.7957427
> gr1@sd
[1] 1.097115
>
> gr1@fit$coefficients <- c(1,2)
> gr1@sd <- 3
> gr1
An object of class "growthObj"
Slot "fit":

Call:
lm(formula = Formula, data = dataf)

Coefficients:
[1]  1  2


Slot "sd":
[1] 3
```

over-write the slots with your desired parameters


PENN