Introduction to demographic analysis of structured population models with variable or stochastic development using R package varDev

Perry de Valpine

January 12, 2009

Contents

This document provides a brief outline of some examples. It does not repeat material in the help files for functions and classes. This package is based on the paper “Stochastic development in biologically structured population models” by Perry de Valpine, accepted to *Ecology* for publication in 2009. This package was used for the cactus example but not for the mediterranean fruit fly example.

The earliest version of this package will be included in *Ecological Archives* as an appendix to the paper. You can check with the author or CRAN to see if there have been any updates.

1 Defining a model

A model requires definitions for:

1. Marginal distributions of stage durations. For constant probability of maturation, as in a Lefkovitch matrix, these are geometric distributions. However, the models are defined with ages starting at 1 instead of 0, but R’s geometric distribution starts at 0. Therefore there is a “geomp1” distribution defined for “geometric plus 1”, starting at 1. See “help(geomp1)” for more information. For distributions of continuous (non-integer) positive values, age and time are discretized by rounding up to the nearest integer.

2. Marginal distributions of age-within-stage of death distributions. For constant mortality these are geometric (plus-one) distributions.

3. Optionally: a correlation matrix for a multivariate Gaussian copula defining correlated stage durations.

4. Optionally: a fecundity or VD.fec object. If omitted, it must be supplied to appropriate functions later.

See “help(VD.model)” for more information.

For an example I will use the use the stage transition and survival rates for a cactus species from Site A of Schmalzel et al. (1995) as a starting point; this has been used by Fox and Gurevitch (2000) and Haridas and Tuljapurkar (2007) as an example of stage-structured demography. The Lefkovitch matrix, with a time-step of one-year, is:

$$\mathbf{A} = \begin{bmatrix} s_{11} & 0 & F_3 \\ s_{12} & s_{22} & 0 \\ 0 & s_{23} & s_{33} \end{bmatrix} = \begin{bmatrix} 0.67 & 0 & 0.56 \\ 0.02 & 0.84 & 0 \\ 0 & 0.14 & 0.97 \end{bmatrix} \tag{1}$$

This has a growth rate of $\lambda = 0.9998 = \exp(-0.00024) = \exp(r)$ and stable stage distribution of (0.58, 0.07, 0.34).

The code to define this model as a literal interpretation of the matrix, i.e. with constant probabilities of maturation across each stage at each time step and independent stage durations, is:

```
> VDM <- VD.model(3, marginal.durations = list(VD.dist("geomp1",
+     list(prob = 2/69)), VD.dist("geomp1", list(prob = 14/98)),
+     VD.dist("geomp1", list(prob = 0.03))), marginal.death.times = list(VD.dist("geomp1",
+     list(prob = 0.31)), VD.dist("geomp1", list(prob = 0.02)),
+     VD.dist("infinite")))
```

```
Warning: no fecundity specified.  You can simulate development and survival, but you will ne
```

Note that the final survival probability is set to be 1 (infinite age-within-stage of death) so that stage "maturation" is equivalent to death.

2 Running a sequential Monte Carlo simulation

This is done by calling "VD.run" with a VD.model object.

```
> VDS <- VD.run(VDM)
```

Now the VDS object can be used for various Monte Carlo integral approximations of demographic interest.

3 Development tables

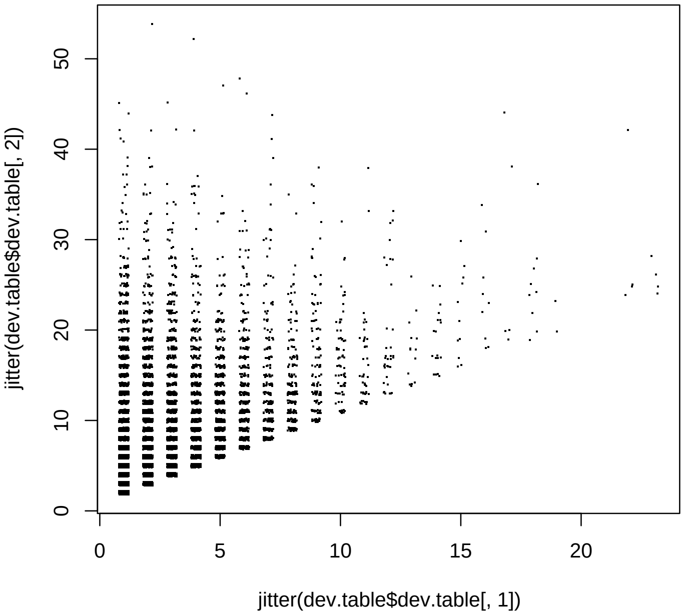
The VDS object has a compact format that is hard to inspect directly. The next step is to turn it into a more understandable development table.

```
> dev.table <- compile.dev.table(VDS)
```

different tables, including total ages of stage transition (the default), "continuous" ages that have not been rounded up to discretize time and age, ages-within-stages of stage transitions, and standard normal deviates used for Gaussian copula modeling of correlated stage durations.

To plot the total age of the maturation from stage 2 to 3 vs maturation from stage 1 to 2:

```
> plot(jitter(dev.table$dev.table[, 1]), jitter(dev.table$dev.table[,
+     2]), pch = ".")
```



Since there are many overlapping integer values I added random jitters to make the figure a bit clearer. Note that stage names can be set up in the "VD.model" object and then used to access results such as columns of the "dev.table$dev.table" data.frame.

4 Expected reproductive output to each age

Calculations such as the Euler-Lotka equation require expected reproductive output to each age, including 0's for death before a given age. A data frame with this information is obtained by:

```
> mean.fec <- calc.average.surv.rep.by.age(dev.table, F = 0.56)
```

Had we created a fecundity in the "VD.model" object, we could have simply passed the "VD.model" object as the F argument. mean.fec has columns "age" and "fecundity.by.age".

5 Long-term population growth rate

Long-term population growth rate can refer to r or $\lambda = e^r$. To obtain r, the solution to the Euler-Lotka equation:

```
> r <- VD.solve.euler(mean.fec)
> r

[1] -0.0001303185
```

In this case the correct $r = -0.00024$, and we are within simulation error of that.

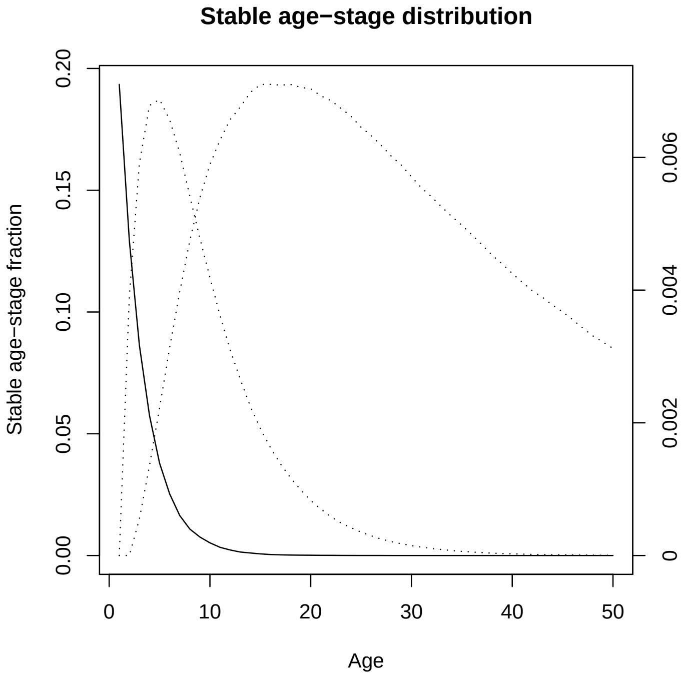
6 Stable age-stage distributions

Obtaining stable age-stage (more generally, stable state) distributions requires the simulation results and r.

```
> SASD <- make.SASD(VDS, r)
> SASD$maturing.frac

[1] 0.02043743 0.13948906 0.00000000

> SASD$dying.frac

[1] 0.31312600 0.01991422 0.03019059

> SASD$SSD

[1] 0.58013215 0.07444084 0.34542701

> VD.mean.age(SASD)

[1]  2.995715  9.274011 42.150974

> VD.mode.age(SASD)

[1]  1  5 16
```

The fractions maturing and dying from each stage and the stable stage distribution match those from the original matrix to within simulation error. We can also plot it with a specialized plot method:

```
> plot(SASD, age.vector = 1:50)

[1] 27.24659
```



See "help(SASD)" for more information on controlling the plots. In this case I limited the horizontal values to make the first two stages more visible. The dashed lines follow the right-hand scale.

7 Finding parameters of other distributions that give the same stable stage distribution as the matrix

7.1 Independent stage durations

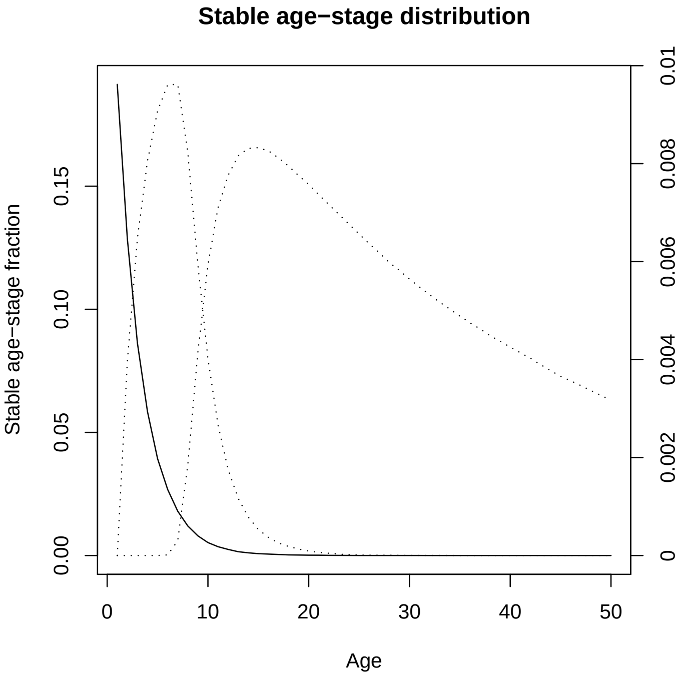
Suppose we want to find parameters of a discretized gamma distribution for the second stage duration that give the same SSD as the matrix. In the case of independent stage durations, we only need to specify the long-term population growth and the stage distribution with one parameter chosen for searching. Note that discretizing of distributions with continuous positive support is done by rounding up, so that the discretized distribution has positive integer support. This is done automatically by "VD.dist" objects, so the user does not need to do anything beyond picking the underlying continuous distribution.

```
> surv2 <- 0.98
> mat2 <- 14/98
> shape2 <- 50
> rate2 <- VD.solve.fraction.maturing.at.SASD.independent(VD.dist("gamma",
+     list(shape = shape2, rate = 1)), VD.dist("geomp1", list(prob = 1 -
+     surv2)), r = -0.00024, fraction.maturing = (mat2), param = "rate",
+     range = c(1e-04, 30), max.age = 200)
> rate2

[1] 8.16705

> VDM <- VD.model(3, marginal.durations = list(VD.dist("geomp1",
+     list(prob = 2/69)), VD.dist("gamma", list(shape = shape2,
+     rate = rate2)), VD.dist("geomp1", list(prob = 0.03))), marginal.death.times = list(VD.
+     list(prob = 0.31)), VD.dist("geomp1", list(prob = 1 - surv2)),
+     VD.dist("infinite")))

Warning: no fecundity specified.  You can simulate development and survival, but you will ne

> VDS <- VD.run(VDM)
> dev.table <- compile.dev.table(VDS)
> mean.fec <- calc.average.surv.rep.by.age(dev.table, F = 0.56)
> r <- VD.solve.euler(mean.fec)
> r

[1] -0.000724299

> SASD <- make.SASD(VDS, r)
> SASD$maturing.frac

[1] 0.01992184 0.14016945 0.00000000

> SASD$dying.frac

[1] 0.30732389 0.02013510 0.03074354

> SASD$SSD

[1] 0.58556621 0.07310137 0.34133241

> VD.mean.age(SASD)

[1]  3.067301  6.790818 43.058053

> VD.mode.age(SASD)

[1]  1  7 15
```

Note the maturing fractions, dying fractions and stable stage distribution do match the original matrix, but the stable age-stage distribution is different:

```
> plot(SASD, age.vector = 1:50)
```

```
[1] 19.89511
```



7.2 Correlated stage durations

Solving for parameters of correlated stage duration distributions that match a given Lefkovitch matrix is more difficult. It requires specification of the model at least up to the stage of interest. Two methods are implemented for this. One does all aspects of the calculations as Monte Carlo integrals. The other does direct summation for survival distributions of the target stage but Monte Carlo integration over the distribution of stage durations. In this example I will first solve for a gamma distribution for stage 1, which can be treated as independent. Then I will define a correlation matrix for the Gaussian copula and solve for a gamma distribution for stage 2, correlated with stage 1, that matches the original matrix in terms of SSD.

```
> surv1 <- 0.69
> mat1 <- 2/69
> shape1 <- 50
> r <- -0.00024
> rate1 <- VD.solve.fraction.maturing.at.SASD.independent(VD.dist("gamma",
+     list(shape = shape1, rate = 1)), VD.dist("geomp1", list(prob = 1 -
+     surv1)), r = r, fraction.maturing = (mat1), param = "rate",
+     range = c(1e-04, 30), max.age = 200)
> rate1

[1] 6.885054

> surv2 <- 0.98
> mat2 <- 14/98
> shape2 <- 50
> my.gauss.cov <- matrix(c(1, 0.75, 0, 0.75, 1, 0, 0, 0, 1), c(3,
+     3))
> VDM <- VD.model(3, marginal.durations = list(VD.dist("gamma",
+     list(shape = shape1, rate = rate1)), VD.dist("gamma", list(shape = 1,
+     rate = 1)), VD.dist("geomp1", list(prob = 0.03))), marginal.death.times = list(VD.dist
+     list(prob = 1 - surv1)), VD.dist("geomp1", list(prob = 1 -
+     surv2)), VD.dist("infinite")), gauss.cov = my.gauss.cov)

Warning: no fecundity specified.  You can simulate development and survival, but you will ne

> rate2 <- VD.solve.fraction.maturing.at.SASD.mc(VDM, iStage = 2,
+     VD.dist("gamma", list(shape = shape2, rate = 1)), r = r,
+     fraction.maturing = mat2, param = "rate", range = c(1e-04,
+         15), max.age = 300, mc.dev.only = TRUE)
> rate2

[1] 7.842234

> marginal.durations(VDM)[[2]] <- VD.dist("gamma", list(shape = shape2,
+     rate = rate2))
> VDS <- VD.run(VDM)
> dev.table <- compile.dev.table(VDS)
> mean.fec <- calc.average.surv.rep.by.age(dev.table, F = 0.56)
> r <- VD.solve.euler(mean.fec)
> r

[1] -0.0001205920

> SASD <- make.SASD(VDS, r)
> SASD$maturing.frac

[1] 0.02014473 0.13989857 0.00000000

> SASD$dying.frac

[1] 0.31041134 0.01954389 0.03006562

> SASD$SSD

[1] 0.58235984 0.07363383 0.34400633
```

```
> VD.mean.age(SASD)
```

```
[1]  2.751187 11.262474 47.905406
```

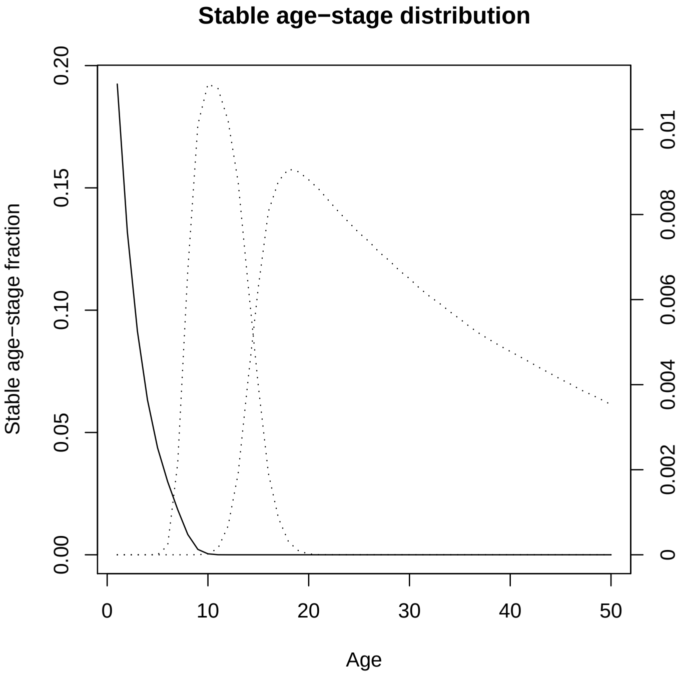
```
> VD.mode.age(SASD)
```

```
[1]  1 10 18
```

Now plot it:

```
> plot(SASD, age.vector = 1:50)
```

```
[1] 17.38877
```



8 Sensitivities and elasticities

Sensitivities are derivatives of λ with respect to model parameters. Elasticities are derivatives of r with respect to the log of model parameters. The derivatives are calculated here as derivatives of r with respect to model parameters, from which either sensitivities or elasticities can be calculated. To get the sensitivities, the calculated derivatives should be multiplied by $\lambda = \exp(r)$. In this example I will define a model with independent stage durations with the gamma distribution for stage 1 obtained from the previous example. I will show sensitivities to the gamma rate parameter for stage 1, the geometric maturation probability for stage 2, and the two survival rates. Note that the “prob” parameter in the death distributions is a mortality, so we multiply by -1 to get the derivative with respect to survival probability. Also note that we need the non-rounded values of stage durations to be included in the dev.table.

```
> VDM <- VD.model(3, marginal.durations = list(VD.dist("gamma",
+     list(shape = shape1, rate = rate1)), VD.dist("geomp1", list(prob = mat2)),
+     VD.dist("geomp1", list(prob = 0.03))), marginal.death.times = list(VD.dist("geomp1",
+     list(prob = 1 - surv1)), VD.dist("geomp1", list(prob = 1 -
+     surv2)), VD.dist("infinite")))

Warning: no fecundity specified.  You can simulate development and survival, but you will ne

> VDS <- VD.run(VDM)
> dev.table <- compile.dev.table(VDS, include.contin = TRUE)
> mean.fec <- calc.average.surv.rep.by.age(dev.table, F = 0.56)
> r <- VD.solve.euler(mean.fec)
> r

[1] -0.0002029075

> SASD <- make.SASD(VDS, r)
> SASD$maturing.frac

[1] 0.02000382 0.14034585 0.00000000

> SASD$dying.frac

[1] 0.30898974 0.01989167 0.03008198

> SASD$SSD

[1] 0.58406877 0.07300675 0.34292448

> VD.mean.age(SASD)

[1]  2.752080 13.597410 46.999571

> VD.mode.age(SASD)

[1]  1  9 20

> F <- 0.56
> -VD.deriv.growth.rate(VDM, r, dev.table, mean.fec, death = 1,
+     param = "prob", make.discrete = FALSE, F = F) * exp(r)

[1] 0.2266652
```

```
> -VD.deriv.growth.rate(VDM, r, dev.table, mean.fec, death = 2,
+     param = "prob", make.discrete = FALSE, F = F) * exp(r)

[1] 0.1358533

> VD.deriv.growth.rate(VDM, r, dev.table, mean.fec, stage = 1,
+     param = "rate", make.discrete = FALSE, F = F) * exp(r)

[1] 0.007881395

> VD.deriv.growth.rate(VDM, r, dev.table, mean.fec, stage = 2,
+     param = "prob", make.discrete = FALSE, F = F) * exp(r)

[1] 0.01830472
```

As of this writing sensitivities for correlated duration models are not implemented, but they can be.

9 This package is preliminary and beta

I cannot guarantee that any results obtained with this package are correct. For the examples I have used to check it, it appears to work. However, this represents relatively limited testing. This is the first release of the package, and many directions for extension exist. If you plan to use this package for real research, please do not hesitate to contact me.

References

Fox, G. A. and Gurevitch, J. (2000). Population numbers count: Tools for near-term demographic analysis. *American Naturalist*, 156(3):242–256.

Haridas, C. V. and Tuljapurkar, S. (2007). Time, transients and elasticity. *Ecology Letters*, 10(12):1143–1153.

Schmalzel, R. J., Reichenbacher, F. W., and Rutman, S. (1995). Demographic study of the rare *Coryphantha robbinsorum* (Cactaceae) in southeastern Arizona. *Madrono*, 42(3):332–348.