

lefko3: a gentle introduction
Creating and analyzing matrix projection models in R

Richard P. Shefferson

2023-07-24

Chapter 8

Population Projection I: Deterministic Analysis

For a moment, nothing happened. Then, after a second or so, nothing continued to happen.

— Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

Deterministic analysis refers to matrix projection conducted assuming that a single matrix encompasses all of the important demographic variation in a population. This single matrix is projected forward an extremely large number of time steps or to a limit of infinity, and the asymptotic properties of the projection are assessed. Projecting forward by the same matrix means that instead of going one time step forward, where $\mathbf{n}_{t+1} = \mathbf{A}\mathbf{n}_t$, we find ourselves analyzing the result of the following equation.

$$\mathbf{n}_t = \mathbf{A}^t \mathbf{n}_0, t \rightarrow \infty \quad (8.1)$$

The asymptotic properties of this projection conform to the results of eigenanalysis, where after some time we see the right eigenvector associated with the dominant eigenvalue yielding the equilibrium stage structure \mathbf{w} , and the left eigenvector associated with the dominant eigenvalue yielding the reproductive value vector \mathbf{v} . The dominant eigenvalue is λ , and is equal to the asymptotic population growth rate, and we also see the following conditions.

$$\mathbf{A}\mathbf{w} = \lambda\mathbf{w} \quad (8.2)$$

$$\mathbf{v}^* \mathbf{A} = \lambda\mathbf{v}^* \quad (8.3)$$

Here, \mathbf{v}^* is the complex conjugate of the left eigenvector and \mathbf{w} is the right eigenvector associated with λ (Caswell, 2001). Once we have the projected population growth rate λ , the stable stage vector, and the reproductive value vector, we can use these terms to derive further values, such as the sensitivity or elasticity of λ to changes in matrix elements.

Package `lefk3` includes a number of functions to aid analysis once matrices are created. These may be of greater utility in some circumstances than established R functions such as `eigen()`, because our functions are made to handle even extremely large, sparse matrices and can handle many kinds of unusual situations that might arise in analysis. Currently, we include functions to estimate element-wise arithmetic mean matrices, population growth rate, stable stage structure, reproductive value, sensitivity and elasticity, and life table response experiments (LTRE and sLTRE). We also include two basic projection functions that show the expected numbers of individuals in each stage forward in time under deterministic, cyclical, and stochastic conditions, with or without density dependence (these two functions will be covered in chapter 10 on General Projection Simulation).

In this chapter, we will utilize the raw MPMs developed in chapter 4 to illustrate the sorts of deterministic analyses that we can perform. Particularly, we will use the `lefkMat` objects named `cypmatrix2r`, `cypmatrix2rp`, `cypmatrix3r`, and `cypmatrix3rp`. However, the function-based MPMs, IPMs, and age-by-stage MPMs that we have created can also be used in all cases.

Let's start off by taking a look at summaries of these four objects.

```
summary(cypmatrix2r)
>
> This ahistorical lefkMat object contains 5 matrices.
>
> Each matrix is square with 11 rows and columns, and a total of 121 elements.
> A total of 120 survival transitions were estimated, with 24 per matrix.
> A total of 40 fecundity transitions were estimated, with 8 per matrix.
> This lefkMat object covers 1 population, 1 patch, and 5 time steps.
>
> The dataset contains a total of 74 unique individuals and 320 unique transitions.
>
> Survival probability sum check (each matrix represented by column in order):
>      [,1] [,2] [,3] [,4] [,5]
> Min.    0.000 0.050 0.050 0.000 0.050
> 1st Qu. 0.100 0.140 0.140 0.100 0.140
> Median  0.689 0.870 0.864 0.610 0.882
> Mean    0.552 0.629 0.629 0.528 0.627
> 3rd Qu. 1.000 1.000 1.000 0.960 1.000
> Max.    1.000 1.000 1.000 1.000 1.000
summary(cypmatrix2rp)
>
> This ahistorical lefkMat object contains 15 matrices.
>
> Each matrix is square with 11 rows and columns, and a total of 121 elements.
> A total of 266 survival transitions were estimated, with 17.733 per matrix.
> A total of 70 fecundity transitions were estimated, with 4.667 per matrix.
> This lefkMat object covers 1 population, 3 patches, and 5 time steps.
>
> The dataset contains a total of 74 unique individuals and 320 unique transitions.
>
> Survival probability sum check (each matrix represented by column in order):
>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
> Min.    0.000 0.000 0.000 0.000 0.000 0.000 0.050 0.050 0.000 0.050 0.000 0.000
> 1st Qu. 0.075 0.025 0.075 0.025 0.075 0.075 0.140 0.140 0.100 0.140 0.100 0.100
> Median  0.180 0.100 0.180 0.100 0.180 0.180 0.909 0.778 0.686 0.857 0.750 0.575
> Mean    0.457 0.361 0.471 0.328 0.417 0.464 0.631 0.611 0.530 0.631 0.562 0.523
> 3rd Qu. 0.955 0.769 1.000 0.592 0.781 1.000 1.000 1.000 0.955 1.000 1.000 1.000
> Max.    1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
>      [,13] [,14] [,15]
> Min.    0.000 0.000 0.000
> 1st Qu. 0.075 0.075 0.100
> Median  0.180 0.180 0.750
> Mean    0.432 0.450 0.562
> 3rd Qu. 0.875 1.000 1.000
> Max.    1.000 1.000 1.000
summary(cypmatrix3r)
```

```

>
> This historical lefkoMat object contains 4 matrices.
>
> Each matrix is square with 121 rows and columns, and a total of 14641 elements.
> A total of 242 survival transitions were estimated, with 60.5 per matrix.
> A total of 54 fecundity transitions were estimated, with 13.5 per matrix.
> This lefkoMat object covers 1 population, 1 patch, and 4 time steps.
>
> The dataset contains a total of 74 unique individuals and 320 unique transitions.
>
> Survival probability sum check (each matrix represented by column in order):
>      [,1] [,2] [,3] [,4]
> Min.   0.000 0.000 0.000 0.000
> 1st Qu. 0.000 0.000 0.000 0.000
> Median 0.000 0.000 0.000 0.000
> Mean   0.173 0.179 0.166 0.198
> 3rd Qu. 0.100 0.100 0.100 0.100
> Max.   1.000 1.000 1.000 1.000
summary(cypmatrix3rp)
>
> This historical lefkoMat object contains 12 matrices.
>
> Each matrix is square with 121 rows and columns, and a total of 14641 elements.
> A total of 516 survival transitions were estimated, with 43 per matrix.
> A total of 70 fecundity transitions were estimated, with 5.833 per matrix.
> This lefkoMat object covers 1 population, 3 patches, and 4 time steps.
>
> The dataset contains a total of 74 unique individuals and 320 unique transitions.
>
> Survival probability sum check (each matrix represented by column in order):
>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
> Min.   0.000 0.0000 0.0000 0.000 0.000 0.000 0.00 0.000 0.000 0.0000 0.000
> 1st Qu. 0.000 0.0000 0.0000 0.000 0.000 0.000 0.00 0.000 0.000 0.0000 0.000
> Median 0.000 0.0000 0.0000 0.000 0.000 0.000 0.00 0.000 0.000 0.0000 0.000
> Mean   0.107 0.0945 0.0851 0.101 0.158 0.158 0.14 0.169 0.119 0.0851 0.119
> 3rd Qu. 0.000 0.0000 0.0000 0.000 0.100 0.100 0.05 0.100 0.000 0.0000 0.000
> Max.   1.000 1.0000 1.0000 1.000 1.000 1.000 1.00 1.000 1.000 1.0000 1.000
>      [,12]
> Min.   0.000
> 1st Qu. 0.000
> Median 0.000
> Mean   0.144
> 3rd Qu. 0.000
> Max.   1.000

```

We see that these four MPMs cover three patches of one population, and that `cypmatrix2r` and `cypmatrix3r` do not distinguish the patches while `cypmatrix2rp` and `cypmatrix3rp` do. Objects `cypmatrix2r` and `cypmatrix2rp` are ahistorical, while `cypmatrix3r` and `cypmatrix3rp` are historical. Because these are raw MPMs and there are six total monitoring occasions covered, we see five matrices estimated in the ahistorical population-level set and four matrices estimated in the historical population-level set. The patch-level MPMs create five and four matrices per patch for the ahistorical and historical cases, respectively. There are 11 life history stages, which we see below.

```

cypmatrix2r$ahstages
>   stage_id stage original_size original_size_b original_size_c min_age max_age
> 1         1   SD             0.0              NA              NA         0   NA
> 2         2   P1             0.0              NA              NA         0   NA
> 3         3   P2             0.0              NA              NA         0   NA
> 4         4   P3             0.0              NA              NA         0   NA
> 5         5   SL             0.0              NA              NA         0   NA
> 6         6   D              0.0              NA              NA         0   NA
> 7         7   XSm            1.0              NA              NA         0   NA
> 8         8   Sm             3.0              NA              NA         0   NA
> 9         9   Md             6.0              NA              NA         0   NA
> 10        10  Lg            11.0              NA              NA         0   NA
> 11        11  XLg           19.5              NA              NA         0   NA
>   repstatus obsstatus propstatus immstatus matstatus entrystage indataset
> 1           0          0           1           0           0           1           0
> 2           0          0           0           1           0           1           0
> 3           0          0           0           1           0           0           0
> 4           0          0           0           1           0           0           0
> 5           0          0           0           1           0           0           0
> 6           0          0           0           0           1           0           1
> 7           1          1           0           0           1           0           1
> 8           1          1           0           0           1           0           1
> 9           1          1           0           0           1           0           1
> 10          1          1           0           0           1           0           1
> 11          1          1           0           0           1           0           1
>   binhalfwidth_raw sizebin_min sizebin_max sizebin_center sizebin_width
> 1                 0.0           0.0           0.0           0.0           0
> 2                 0.0           0.0           0.0           0.0           0
> 3                 0.0           0.0           0.0           0.0           0
> 4                 0.0           0.0           0.0           0.0           0
> 5                 0.0           0.0           0.0           0.0           0
> 6                 0.5          -0.5           0.5           0.0           1
> 7                 0.5           0.5           1.5           1.0           1
> 8                 1.5           1.5           4.5           3.0           3
> 9                 1.5           4.5           7.5           6.0           3
> 10                3.5           7.5          14.5          11.0           7
> 11                5.0          14.5          24.5          19.5          10
>   binhalfwidthb_raw sizebinb_min sizebinb_max sizebinb_center sizebinb_width
> 1                 NA             NA             NA             NA             NA
> 2                 NA             NA             NA             NA             NA
> 3                 NA             NA             NA             NA             NA
> 4                 NA             NA             NA             NA             NA
> 5                 NA             NA             NA             NA             NA
> 6                 NA             NA             NA             NA             NA
> 7                 NA             NA             NA             NA             NA
> 8                 NA             NA             NA             NA             NA
> 9                 NA             NA             NA             NA             NA
> 10                NA             NA             NA             NA             NA
> 11                NA             NA             NA             NA             NA
>   binhalfwidthc_raw sizebinc_min sizebinc_max sizebinc_center sizebinc_width
> 1                 NA             NA             NA             NA             NA

```

```

> 2      NA      NA      NA      NA      NA
> 3      NA      NA      NA      NA      NA
> 4      NA      NA      NA      NA      NA
> 5      NA      NA      NA      NA      NA
> 6      NA      NA      NA      NA      NA
> 7      NA      NA      NA      NA      NA
> 8      NA      NA      NA      NA      NA
> 9      NA      NA      NA      NA      NA
> 10     NA      NA      NA      NA      NA
> 11     NA      NA      NA      NA      NA
>  group      comments alive almostborn
> 1      0      Dormant seed      1      0
> 2      0      1st yr protocorm    1      0
> 3      0      2nd yr protocorm    1      0
> 4      0      3rd yr protocorm    1      0
> 5      0      Seedling            1      0
> 6      0      Dormant adult       1      0
> 7      0      Extra small adult (1 shoot) 1      0
> 8      0      Small adult (2-4 shoots) 1      0
> 9      0      Medium adult (5-7 shoots) 1      0
> 10     0      Large adult (8-14 shoots) 1      0
> 11     0      Extra large adult (>14 shoots) 1      0

```

Let's move on now to our first deterministic analysis: the estimation of the population growth rate λ .

8.1 Population growth rate

The deterministic population growth rate is estimated with function `lambda3()`. This function estimates the population growth rate (λ) as the dominant eigenvalue of each matrix provided (technically, the largest real part of the estimated eigenvalues for each matrix). Where matrices are large, as in most historical or age-by-stage cases, `lambda3()` first converts matrices to sparse format in order to speed up estimation. This is all done under the hood, so the user need not do anything prior to analysis to prepare for this.

First, let's look at the population growth rate estimates at the whole-population level, using the ahistorical `cypmatrix2r`.

```

cyp2r_lam <- lambda3(cypmatrix2r)
cyp2r_lam
>  pop patch year2  lambda
> 1   1   1  2004 1.1204918
> 2   1   1  2005 1.0906393
> 3   1   1  2006 1.1685560
> 4   1   1  2007 0.9519705
> 5   1   1  2008 1.1092330

```

Since we have five matrices in that MPM, we see five estimates of λ . These range from 0.952 in 2007, to 1.169 in 2006. It may be interesting to compare these to the historical case, as below.

```

cyp3r_lam <- lambda3(cypmatrix3r)
cyp3r_lam

```

```
> pop patch year2 lambda
> 1 1 1 2005 0.9794481
> 2 1 1 2006 1.0000000
> 3 1 1 2007 0.9180329
> 4 1 1 2008 1.0570860
```

We see one fewer estimate of λ because there is one fewer matrix. Estimated λ also appears to be somewhat lower in the historical case. Indeed, mean $\lambda \pm 1\text{se}$ for years 2005 through 2008 is 1.088 ± 0.041 and 0.989 ± 0.029 for the ahistorical and historical cases, respectively. Generally, results like this suggest that individual history influences population dynamics via some sort of long-term cost, and so should not be ignored.

Let's move on now and look at λ for the patch-level MPMs. We may wish to compare the ahistorical estimates to the historical estimates. Here, we will use `lambda3()` to develop these estimates for us, and then we will plot them (figure 8.1).

```
cyp2rp_lam <- lambda3(cypmatrix2rp)
cyp3rp_lam <- lambda3(cypmatrix3rp)

plot(lambda ~ year2, data = subset(cyp2rp_lam, patch == "A"),
     ylim = c(0.4, 1.4), type = "l", lwd = 2, bty = "n")
lines(lambda ~ year2, data = subset(cyp2rp_lam, patch == "B"), type = "l",
     lwd = 2, lty = 2)
lines(lambda ~ year2, data = subset(cyp2rp_lam, patch == "C"), type = "l",
     lwd = 2, lty = 3)
lines(lambda ~ year2, data = subset(cyp3rp_lam, patch == "A"), type = "l",
     lwd = 2, lty = 1, col = "red")
lines(lambda ~ year2, data = subset(cyp3rp_lam, patch == "B"), type = "l",
     lwd = 2, lty = 2, col = "red")
lines(lambda ~ year2, data = subset(cyp3rp_lam, patch == "C"), type = "l",
     lwd = 2, lty = 3, col = "red")
legend("bottomleft", c("A ahistorical", "B ahistorical", "C ahistorical",
     "A historical", "B historical", "C historical"), lty = c(1, 2, 3, 1, 2, 3),
     col = c("black", "black", "black", "red", "red", "red"), lwd = 2, cex = 0.7,
     bty = "n")
```

Here we see that λ is quite different between the ahistorical and historical cases in the raw MPMs. This should give us pause and encourage us to consider using historical approaches to correct for the influence of individual history.

We may now wish to compare λ for the mean matrices, as well. Let's do so, as below.

```
cyp2r_mean <- lmean(cypmatrix2r)
cyp2rp_mean <- lmean(cypmatrix2rp)
cyp3r_mean <- lmean(cypmatrix3r)
cyp3rp_mean <- lmean(cypmatrix3rp)

# Overall raw ahistorical lambda
lambda3(cyp2r_mean)
> pop patch lambda
> 1 1 1 1.082189

# Raw ahistorical lambda by patch
lambda3(cyp2rp_mean)
```

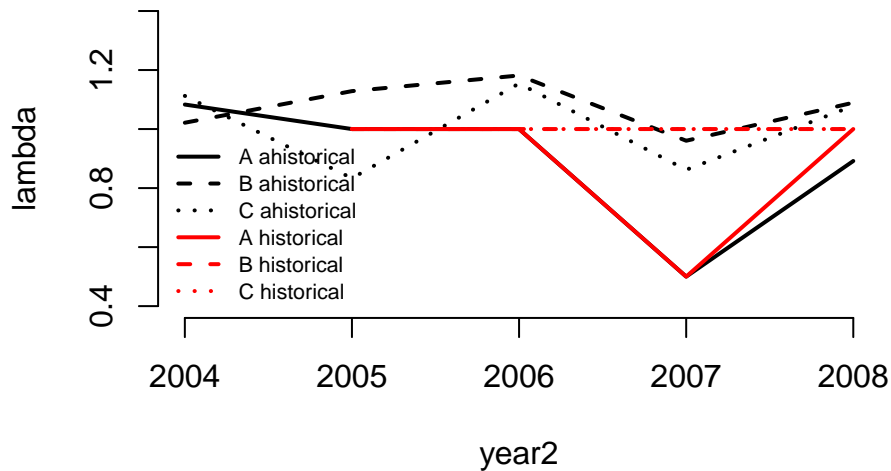


Figure 8.1: Lambda of ahistorical vs. historical raw MPMs

```

> pop patch lambda
> 1 1 A 0.9835717
> 2 1 B 1.1051470
> 3 1 C 1.0717513
> 4 1 0 1.0450267

# Overall raw historical lambda
lambda3(cyp3r_mean)
> pop patch lambda
> 1 1 1 1.011018

# Raw historical lambda by patch
lambda3(cyp3rp_mean)
> pop patch lambda
> 1 1 A 0.8510445
> 2 1 B 0.9481901
> 3 1 C 0.9671030
> 4 1 0 0.8946030

```

As we saw before, our historical estimates are lower than our ahistorical estimates. We also see that λ for the patch-weighted mean matrix (marked as patch 0 in the patch-level output) is lower than the whole-population λ in both ahistorical and historical cases. The latter result suggests that the patches have different levels of influence on overall population dynamics because of different numbers of individuals.

8.2 Stable stage distribution and reproductive value

The stable stage distribution is a vector showing the expected proportion of the population composed of each stage after we have projected the population forward in time sufficiently to create asymptotic conditions. The reproductive value vector gives the expected asymptotic contribution of an individual in each stage to future generations, and so has a relationship (though a complex one) to natural selection (Caswell, 2001). In ahistorical analysis, the stable stage distribution and the reproductive value vector are estimated as the standardized right and left eigenvectors, respectively, associated with the dominant

eigenvalue of the matrix. Standardization of the stable stage distribution is handled by dividing each respective element of the right eigenvector by the sum of the elements in that eigenvector, yielding a vector that sums to 1.0. Standardization of the reproductive value vector is handled by dividing each element in the left eigenvector by the value of the first non-zero element in that eigenvector, which is also often though not always the lowest value. This estimation makes a number of key assumptions, and we urge readers to read Caswell (2001) for further details.

The methods mentioned above apply to historical matrices as well (deVries and Caswell, 2018). However, as described, these methods only provide the stable stage distribution and reproductive values of stage pairs, since each row and column in a historical matrix corresponds to a stage pair rather than an individual stage. We provide two functions, `stablestage3()` and `repvalue3()` to allow the estimation of these vectors for both ahistorical and historical MPMs. When provided with a historical MPM, these functions also estimate **historically-corrected** stable stage distributions and reproductive value vectors, which provide the stable stage distribution and reproductive value vector for the original stages in the associated life history model corrected for individual history. These vectors are comparable to the ahistorical stable stage distributions and reproductive value vectors. The historically-corrected stable stage proportion for stage j is estimated as the sum of all stable stage proportions for stage j in occasion t across all stages in occasion $t-1$.

$$SS_j = \sum_{l=1}^m w_{jl} \quad (8.4)$$

Here, l is stage in occasion $t-1$, m is the number of stages, SS_j is the stable stage proportion of stage j , and w_{jl} is the stable stage proportion of stage pair jl given as the standardized right eigenvector corresponding to the dominant eigenvalue of the hMPM.

The historically-corrected reproductive value of stage j is calculated as the sum of all reproductive values for stage j in occasion t across all stages in occasion $t-1$, weighted by the stable stage proportion of the respective stage pair divided by the stable stage proportion of stage j at occasion t (Ehrlén, 2000).

$$RV_j = \sum_{l=1}^m v_{jl}(w_{jl}/SS_j) \quad (8.5)$$

Here, RV_j refers to reproductive value of stage j , and v_{jl} refers to the reproductive value of the stage pair as given by the standardized left eigenvector of the dominant eigenvalue of the historical MPM. Individual history can alter the stable stage distribution and reproductive value vector quite drastically from those from ahistorical analyses.

Now let's take a look at the stable stage distributions at the population level. We will start by looking just at the ahistorical equilibrium.

```
tm2ss_r <- stablestage3(cyp2r_mean)
tm2ss_r
>   matrix stage_id stage      ss_prop
> 1     1         1   SD 4.709506e-01
> 2     1         2   P1 4.796543e-01
> 3     1         3   P2 4.432261e-02
> 4     1         4   P3 4.095645e-03
> 5     1         5   SL 1.983961e-04
> 6     1         6    D 4.309986e-05
> 7     1         7  XSm 2.759814e-04
> 8     1         8   Sm 3.049081e-04
> 9     1         9   Md 9.705559e-05
> 10    1        10   Lg 4.938104e-05
> 11    1        11  XLg 8.078062e-06
```

The output shows us the stage information and corresponding equilibril proportion (`ss_prop`) for each matrix, of which there is only one. For each matrix, the proportions should sum to 1.0. Here, it appears that 48% of the stable stage structure is 1st year protocorms, and 47% is composed of dormant seeds, with all other stages comprising the remaining 5%.

Now let's compare this output to the historical case.

```
tm3ss_r <- stablestage3(cyp3r_mean)
tm3ss_r
> $hist
>      matrix stage_id_2 stage_id_1 stage_2 stage_1      ss_prop
> 1         1         1         1     SD     SD 4.588724e-02
> 2         1         2         1     P1     SD 4.679498e-02
> 3         1         3         1     P2     SD 0.000000e+00
> 4         1         4         1     P3     SD 0.000000e+00
> 5         1         5         1     SL     SD 0.000000e+00
> 6         1         6         1         D     SD 0.000000e+00
> 7         1         7         1     XSm    SD 0.000000e+00
> 8         1         8         1         Sm    SD 0.000000e+00
> 9         1         9         1         Md    SD 0.000000e+00
> 10        1        10         1         Lg    SD 0.000000e+00
> 11        1        11         1     XLg    SD 0.000000e+00
> 12        1         1         2     SD     P1 0.000000e+00
> 13        1         2         2     P1     P1 0.000000e+00
> 14        1         3         2     P2     P1 4.688477e-02
> 15        1         4         2     P3     P1 0.000000e+00
> 16        1         5         2     SL     P1 0.000000e+00
> 17        1         6         2         D     P1 0.000000e+00
> 18        1         7         2     XSm    P1 0.000000e+00
> 19        1         8         2         Sm    P1 0.000000e+00
> 20        1         9         2         Md    P1 0.000000e+00
> 21        1        10         2         Lg    P1 0.000000e+00
> 22        1        11         2     XLg    P1 0.000000e+00
> 23        1         1         3     SD     P2 0.000000e+00
> 24        1         2         3     P1     P2 0.000000e+00
> 25        1         3         3     P2     P2 0.000000e+00
> 26        1         4         3     P3     P2 4.637381e-03
> 27        1         5         3     SL     P2 0.000000e+00
> 28        1         6         3         D     P2 0.000000e+00
> 29        1         7         3     XSm    P2 0.000000e+00
> 30        1         8         3         Sm    P2 0.000000e+00
> 31        1         9         3         Md    P2 0.000000e+00
> 32        1        10         3         Lg    P2 0.000000e+00
> 33        1        11         3     XLg    P2 0.000000e+00
> 34        1         1         4     SD     P3 0.000000e+00
> 35        1         2         4     P1     P3 0.000000e+00
> 36        1         3         4     P2     P3 0.000000e+00
> 37        1         4         4     P3     P3 0.000000e+00
> 38        1         5         4     SL     P3 2.293422e-04
> 39        1         6         4         D     P3 0.000000e+00
> 40        1         7         4     XSm    P3 0.000000e+00
> 41        1         8         4         Sm    P3 0.000000e+00
> 42        1         9         4         Md    P3 0.000000e+00
```

> 43	1	10	4	Lg	P3	0.000000e+00
> 44	1	11	4	XLg	P3	0.000000e+00
> 45	1	1	5	SD	SL	0.000000e+00
> 46	1	2	5	P1	SL	0.000000e+00
> 47	1	3	5	P2	SL	0.000000e+00
> 48	1	4	5	P3	SL	0.000000e+00
> 49	1	5	5	SL	SL	1.193225e-05
> 50	1	6	5	D	SL	2.075488e-05
> 51	1	7	5	XSm	SL	9.455403e-05
> 52	1	8	5	Sm	SL	3.241594e-05
> 53	1	9	5	Md	SL	0.000000e+00
> 54	1	10	5	Lg	SL	0.000000e+00
> 55	1	11	5	XLg	SL	0.000000e+00
> 56	1	1	6	SD	D	0.000000e+00
> 57	1	2	6	P1	D	0.000000e+00
> 58	1	3	6	P2	D	0.000000e+00
> 59	1	4	6	P3	D	0.000000e+00
> 60	1	5	6	SL	D	0.000000e+00
> 61	1	6	6	D	D	1.952294e-06
> 62	1	7	6	XSm	D	3.097439e-05
> 63	1	8	6	Sm	D	2.853934e-05
> 64	1	9	6	Md	D	0.000000e+00
> 65	1	10	6	Lg	D	1.952294e-06
> 66	1	11	6	XLg	D	0.000000e+00
> 67	1	1	7	SD	XSm	1.585917e-02
> 68	1	2	7	P1	XSm	1.585917e-02
> 69	1	3	7	P2	XSm	0.000000e+00
> 70	1	4	7	P3	XSm	0.000000e+00
> 71	1	5	7	SL	XSm	0.000000e+00
> 72	1	6	7	D	XSm	4.039813e-05
> 73	1	7	7	XSm	XSm	2.341842e-04
> 74	1	8	7	Sm	XSm	1.255182e-04
> 75	1	9	7	Md	XSm	6.421039e-06
> 76	1	10	7	Lg	XSm	0.000000e+00
> 77	1	11	7	XLg	XSm	0.000000e+00
> 78	1	1	8	SD	Sm	1.825976e-01
> 79	1	2	8	P1	Sm	1.825976e-01
> 80	1	3	8	P2	Sm	0.000000e+00
> 81	1	4	8	P3	Sm	0.000000e+00
> 82	1	5	8	SL	Sm	0.000000e+00
> 83	1	6	8	D	Sm	2.368566e-05
> 84	1	7	8	XSm	Sm	1.114763e-04
> 85	1	8	8	Sm	Sm	2.007491e-04
> 86	1	9	8	Md	Sm	4.149712e-05
> 87	1	10	8	Lg	Sm	4.401622e-06
> 88	1	11	8	XLg	Sm	0.000000e+00
> 89	1	1	9	SD	Md	1.909135e-01
> 90	1	2	9	P1	Md	1.909135e-01
> 91	1	3	9	P2	Md	0.000000e+00
> 92	1	4	9	P3	Md	0.000000e+00
> 93	1	5	9	SL	Md	0.000000e+00

```

> 94      1      6      9      D      Md 0.000000e+00
> 95      1      7      9     XSm    Md 1.004197e-05
> 96      1      8      9      Sm     Md 2.949146e-05
> 97      1      9      9      Md     Md 6.204565e-05
> 98      1     10      9      Lg     Md 5.996768e-06
> 99      1     11      9     XLg    Md 0.000000e+00
> 100     1      1     10     SD     Lg 3.337106e-02
> 101     1      2     10     P1     Lg 3.337106e-02
> 102     1      3     10     P2     Lg 0.000000e+00
> 103     1      4     10     P3     Lg 0.000000e+00
> 104     1      5     10     SL     Lg 0.000000e+00
> 105     1      6     10      D     Lg 0.000000e+00
> 106     1      7     10     XSm    Lg 0.000000e+00
> 107     1      8     10      Sm     Lg 1.194258e-06
> 108     1      9     10      Md     Lg 2.677112e-06
> 109     1     10     10     Lg     Lg 5.257711e-06
> 110     1     11     10     XLg    Lg 6.918967e-07
> 111     1      1     11     SD     XLg 4.477104e-03
> 112     1      2     11     P1     XLg 4.477104e-03
> 113     1      3     11     P2     XLg 0.000000e+00
> 114     1      4     11     P3     XLg 0.000000e+00
> 115     1      5     11     SL     XLg 0.000000e+00
> 116     1      6     11      D     XLg 0.000000e+00
> 117     1      7     11     XSm    XLg 0.000000e+00
> 118     1      8     11      Sm     XLg 0.000000e+00
> 119     1      9     11      Md     XLg 0.000000e+00
> 120     1     10     11     Lg     XLg 1.692245e-07
> 121     1     11     11     XLg    XLg 3.421782e-07
>
> $ahist
>   matrix stage_id stage      ss_prop
> 1      1      1      SD 4.731057e-01
> 2      1      2      P1 4.740135e-01
> 3      1      3      P2 4.688477e-02
> 4      1      4      P3 4.637381e-03
> 5      1      5      SL 2.412744e-04
> 6      1      6      D  8.679096e-05
> 7      1      7      XSm 4.812309e-04
> 8      1      8      Sm  4.179082e-04
> 9      1      9      Md  1.126409e-04
> 10     1     10     Lg  1.777762e-05
> 11     1     11     XLg  1.034075e-06

```

We see two data frames, the first covering the expected equilibril distribution of stage pairs, and the second showing the corrected equilibril stage distribution. In both cases, the proportions sum to 1.0 within each matrix, of which there is only one. The amount of information here is difficult to digest at one glance, so let's compare the ahistorical stage distribution to the historically corrected stage distribution via a bar plot (figure 8.2).

```

ss_put_together <- cbind.data.frame(tm2ss_r$ss_prop, tm3ss_r$ahist$ss_prop)
names(ss_put_together) <- c("ahistorical", "historical")
rownames(ss_put_together) <- tm2ss_r$stage

```

```

barplot(t(ss_put_together), beside=T, ylab = "Proportion", xlab = "Stage",
        ylim = c(0, 0.85), col = c("black", "orangered"), bty = "n")
legend("topright", c("ahistorical", "historical"), col = c("black", "orangered"),
       pch = 15, cex = 0.9, bty = "n")

```

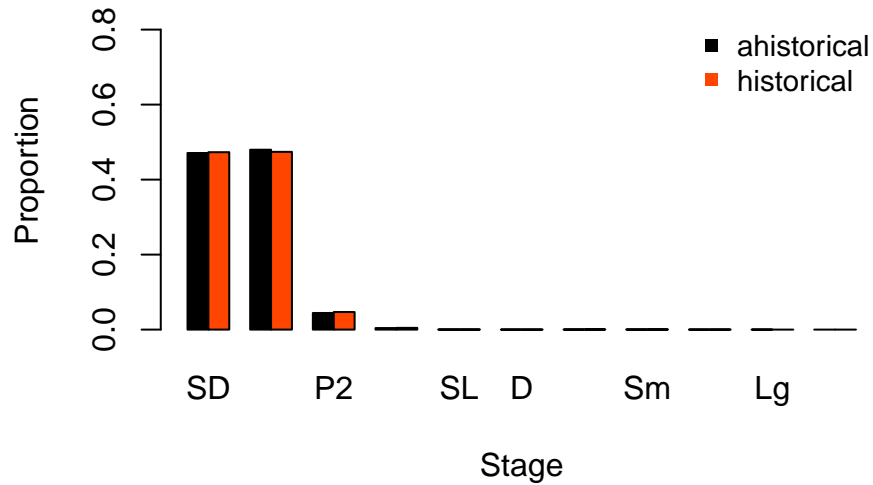


Figure 8.2: Ahistorical vs. historically-corrected stable stage distribution

Overall, these are very similar patterns. Particularly, both ahistorical and historical analyses suggest that dormant seeds and first year protocorms take up the greatest share of the stable stage structure, with second year protocorms coming next. Adults contribute little to the stable stage structure. However, the historical model predicts a slightly greater share of the population composed of dormant seeds and 2nd year protocorms, and a slightly lower share composed of 1st year protocorms.

Next let's look at the reproductive values associated with both ahistorical and historical approaches. We will initially look at just the ahistorical case.

```

tm2rv_r <- repvalue3(cyp2r_mean)
tm2rv_r
>      matrix stage_id stage    rep_value
> 1      1      1      SD      1.00000
> 2      1      2      P1      10.02189
> 3      1      3      P2     108.45573
> 4      1      4      P3    1173.69567
> 5      1      5      SL   25403.20404
> 6      1      6       D   39100.90885
> 7      1      7     XSm   37601.08685
> 8      1      8      Sm   55396.33397
> 9      1      9      Md   85190.53793
> 10     1     10      Lg  128406.01230
> 11     1     11     XLg  179758.08291

```

We have a wide range in reproductive values among the stages, with the smallest value for dormant seeds and the largest for extra large adults. Note that the reproductive value vector is the left eigenvector divided by its first non-zero element, which here is the dormant seed stage. This strongly suggests that the largest adults have a disproportionately large impact on future generations.

Let's now look at the historical reproductive values.

```

tm3rv_r <- repvalue3(cyp3r_mean)
tm3rv_r
> $hist
> matrix stage_id_2 stage_id_1 stage_2 stage_1 rep_value
> 1 1 1 1 SD SD 1.000000e+00
> 2 1 2 1 P1 SD 9.310181e+00
> 3 1 3 1 P2 SD 0.000000e+00
> 4 1 4 1 P3 SD 0.000000e+00
> 5 1 5 1 SL SD 0.000000e+00
> 6 1 6 1 D SD 0.000000e+00
> 7 1 7 1 XSm SD 0.000000e+00
> 8 1 8 1 Sm SD 0.000000e+00
> 9 1 9 1 Md SD 0.000000e+00
> 10 1 10 1 Lg SD 0.000000e+00
> 11 1 11 1 XLg SD 0.000000e+00
> 12 1 1 2 SD P1 0.000000e+00
> 13 1 2 2 P1 P1 0.000000e+00
> 14 1 3 2 P2 P1 9.412762e+01
> 15 1 4 2 P3 P1 0.000000e+00
> 16 1 5 2 SL P1 0.000000e+00
> 17 1 6 2 D P1 0.000000e+00
> 18 1 7 2 XSm P1 0.000000e+00
> 19 1 8 2 Sm P1 0.000000e+00
> 20 1 9 2 Md P1 0.000000e+00
> 21 1 10 2 Lg P1 0.000000e+00
> 22 1 11 2 XLg P1 0.000000e+00
> 23 1 1 3 SD P2 0.000000e+00
> 24 1 2 3 P1 P2 0.000000e+00
> 25 1 3 3 P2 P2 0.000000e+00
> 26 1 4 3 P3 P2 9.516473e+02
> 27 1 5 3 SL P2 0.000000e+00
> 28 1 6 3 D P2 0.000000e+00
> 29 1 7 3 XSm P2 0.000000e+00
> 30 1 8 3 Sm P2 0.000000e+00
> 31 1 9 3 Md P2 0.000000e+00
> 32 1 10 3 Lg P2 0.000000e+00
> 33 1 11 3 XLg P2 0.000000e+00
> 34 1 1 4 SD P3 0.000000e+00
> 35 1 2 4 P1 P3 0.000000e+00
> 36 1 3 4 P2 P3 0.000000e+00
> 37 1 4 4 P3 P3 0.000000e+00
> 38 1 5 4 SL P3 1.924265e+04
> 39 1 6 4 D P3 0.000000e+00
> 40 1 7 4 XSm P3 0.000000e+00
> 41 1 8 4 Sm P3 0.000000e+00
> 42 1 9 4 Md P3 0.000000e+00
> 43 1 10 4 Lg P3 0.000000e+00
> 44 1 11 4 XLg P3 0.000000e+00
> 45 1 1 5 SD SL 0.000000e+00
> 46 1 2 5 P1 SL 0.000000e+00
> 47 1 3 5 P2 SL 0.000000e+00

```

> 48	1	4	5	P3	SL	0.000000e+00
> 49	1	5	5	SL	SL	1.924265e+04
> 50	1	6	5	D	SL	1.600483e+04
> 51	1	7	5	XSm	SL	2.685861e+04
> 52	1	8	5	Sm	SL	4.755014e+04
> 53	1	9	5	Md	SL	0.000000e+00
> 54	1	10	5	Lg	SL	0.000000e+00
> 55	1	11	5	XLg	SL	0.000000e+00
> 56	1	1	6	SD	D	0.000000e+00
> 57	1	2	6	P1	D	0.000000e+00
> 58	1	3	6	P2	D	0.000000e+00
> 59	1	4	6	P3	D	0.000000e+00
> 60	1	5	6	SL	D	0.000000e+00
> 61	1	6	6	D	D	4.034282e+03
> 62	1	7	6	XSm	D	1.631493e+04
> 63	1	8	6	Sm	D	2.683487e+04
> 64	1	9	6	Md	D	0.000000e+00
> 65	1	10	6	Lg	D	3.449945e+04
> 66	1	11	6	XLg	D	0.000000e+00
> 67	1	1	7	SD	XSm	1.019782e+00
> 68	1	2	7	P1	XSm	9.310181e+00
> 69	1	3	7	P2	XSm	0.000000e+00
> 70	1	4	7	P3	XSm	0.000000e+00
> 71	1	5	7	SL	XSm	0.000000e+00
> 72	1	6	7	D	XSm	1.600483e+04
> 73	1	7	7	XSm	XSm	2.728434e+04
> 74	1	8	7	Sm	XSm	5.010449e+04
> 75	1	9	7	Md	XSm	6.833780e+04
> 76	1	10	7	Lg	XSm	1.518934e+04
> 77	1	11	7	XLg	XSm	0.000000e+00
> 78	1	1	8	SD	Sm	1.019782e+00
> 79	1	2	8	P1	Sm	9.310181e+00
> 80	1	3	8	P2	Sm	0.000000e+00
> 81	1	4	8	P3	Sm	0.000000e+00
> 82	1	5	8	SL	Sm	0.000000e+00
> 83	1	6	8	D	Sm	1.519080e+04
> 84	1	7	8	XSm	Sm	3.623157e+04
> 85	1	8	8	Sm	Sm	7.229060e+04
> 86	1	9	8	Md	Sm	1.037423e+05
> 87	1	10	8	Lg	Sm	1.421550e+04
> 88	1	11	8	XLg	Sm	0.000000e+00
> 89	1	1	9	SD	Md	1.019782e+00
> 90	1	2	9	P1	Md	9.310181e+00
> 91	1	3	9	P2	Md	0.000000e+00
> 92	1	4	9	P3	Md	0.000000e+00
> 93	1	5	9	SL	Md	0.000000e+00
> 94	1	6	9	D	Md	0.000000e+00
> 95	1	7	9	XSm	Md	4.524192e+04
> 96	1	8	9	Sm	Md	6.755724e+04
> 97	1	9	9	Md	Md	1.381815e+05
> 98	1	10	9	Lg	Md	8.161812e+04

```

> 99      1      11      9      XLg      Md 0.000000e+00
> 100     1       1      10     SD      Lg 1.019782e+00
> 101     1       2      10     P1      Lg 9.310181e+00
> 102     1       3      10     P2      Lg 0.000000e+00
> 103     1       4      10     P3      Lg 0.000000e+00
> 104     1       5      10     SL      Lg 0.000000e+00
> 105     1       6      10     D       Lg 0.000000e+00
> 106     1       7      10     XSm     Lg 0.000000e+00
> 107     1       8      10     Sm     Lg 5.355023e+04
> 108     1       9      10     Md     Lg 6.142678e+04
> 109     1      10      10     Lg     Lg 1.395183e+05
> 110     1      11      10     XLg    Lg 8.345149e+04
> 111     1       1      11     SD     XLg 1.019782e+00
> 112     1       2      11     P1     XLg 9.310181e+00
> 113     1       3      11     P2     XLg 0.000000e+00
> 114     1       4      11     P3     XLg 0.000000e+00
> 115     1       5      11     SL     XLg 0.000000e+00
> 116     1       6      11     D     XLg 0.000000e+00
> 117     1       7      11     XSm    XLg 0.000000e+00
> 118     1       8      11     Sm     XLg 0.000000e+00
> 119     1       9      11     Md     XLg 0.000000e+00
> 120     1      10      11     Lg     XLg 6.790669e+04
> 121     1      11      11     XLg    XLg 7.189852e+04
>
> $ahist
>   matrix stage_id stage   rep_value
> 1      1         1   SD 1.000000e+00
> 2      1         2   P1 9.146789e+00
> 3      1         3   P2 9.247569e+01
> 4      1         4   P3 9.349460e+02
> 5      1         5   SL 1.890495e+04
> 6      1         6    D 1.524115e+04
> 7      1         7  XSm 2.843405e+04
> 8      1         8   Sm 5.915940e+04
> 9      1         9   Md 1.175879e+05
> 10     1        10   Lg 7.540168e+04
> 11     1        11  XLg 7.823111e+04

```

As in the stable stage output, we see two data frames. The first shows the reproductive values of historical stage pairs, where the has been standardized by dividing the vector against its minimum element. Most reproductive values are zero, generally when referring to an impossible stage pairing (e.g. an individual cannot be a small adult in one year and a 3rd year protocorm in the next). The biologically possible pairs are generally positive, and rather comparable to the ahistorical case. The second data frame shows the historically-corrected reproductive values for each stage in the stageframe. Note that individual history has reduced the reproductive values of most stages. Let's compare the ahistorical reproductive values to the historically-corrected reproductive values, in a side-by-side pair of plots (figure 8.3).

```

tm2rv_r <- repvalue3(cyp2r_mean)
tm3rv_r <- repvalue3(cyp3r_mean)

tm2rv_plot <- as.matrix(tm2rv_r$rep_value)

```



```

rownames(tm2rv_plot) <- tm2rv_r$stage

tm3rv_plot <- as.matrix(tm3rv_r$ahist$rep_value)
rownames(tm3rv_plot) <- tm3rv_r$ahist$stage

par(mfrow=c(1,2))
barplot(t(tm2rv_plot), ylab = "Rep value", xlab = "Stage", col = "black",
        ylim = c(0, 200000), bty = "n")
title("a)", adj = 0)
legend("topleft", c("ahistorical", "historical"), col = c("black", "orangered"),
       pch = 15, bty = "n")

barplot(t(tm3rv_plot), ylab = "", xlab = "Stage", col = "orangered",
        ylim = c(0, 200000), bty = "n")
title("b)", adj = 0)

```

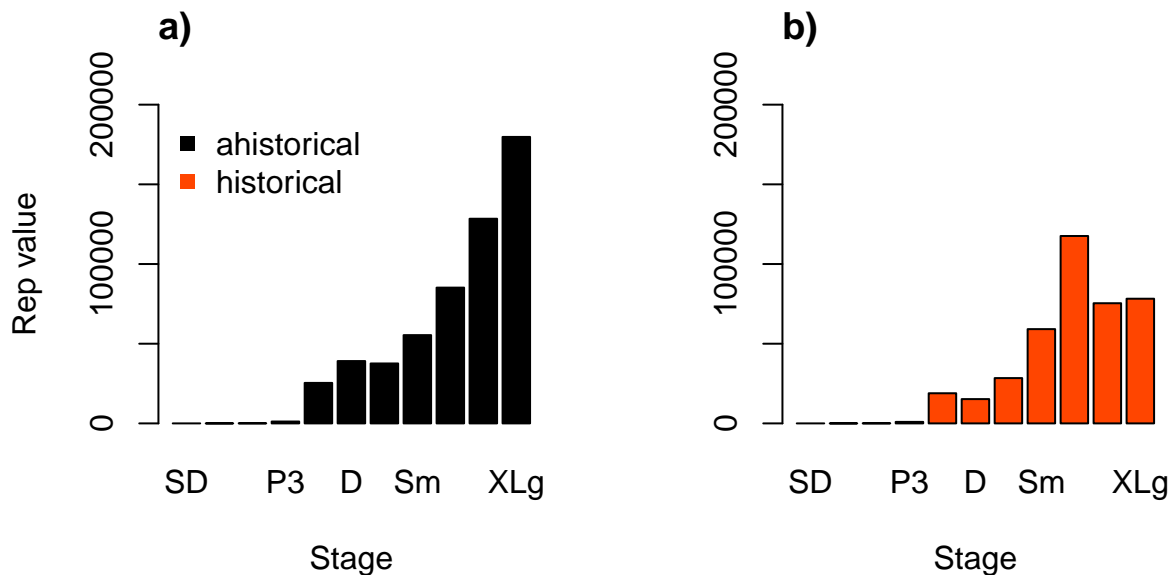


Figure 8.3: Ahistorical (a) vs. historically-corrected (b) deterministic reproductive values

We see some big differences here. In the ahistorical case, reproductive value increases with size, reaching its peak in the largest adults. In contrast, in the historical case, the greatest reproductive values are associated with medium adults, decreasing again in large and extra large adults. So, history appears to have large effects here. Given that reproductive value has implications for natural selection, we might hypothesize that size-related traits might evolve differently under these two models. Interesting results in need of further study!

8.3 Sensitivity analysis

Package `lefk3` contains functions allowing users to conduct deterministic and stochastic sensitivity and elasticity analyses. Sensitivity and elasticity analysis are forms of perturbation analysis, and we urge readers to consult Caswell (2001) and Caswell (2019) to become fully acquainted with the topic. Here, we discuss just the most important aspects to understand these analyses as conducted using `lefk3`.

The sensitivity of λ , the deterministic population growth rate, to a specific element in a projection matrix can be calculated using eigenanalysis, as below.

$$\frac{\partial \lambda}{\partial a_{kj}} = \frac{\bar{v}_k w_j}{\langle \mathbf{w}, \mathbf{v} \rangle} \quad (8.6)$$

Here, \mathbf{w} is the stable stage distribution vector calculated as the right eigenvector of the dominant eigenvalue of the matrix (standardized to sum to 1.0), \mathbf{v} is the reproductive value vector calculated as the associated left eigenvector (standardized by dividing by the value of the first non-zero or minimum value), and \bar{v}_k is the complex conjugate of element k of \mathbf{v} . k refers to stage in occasion $t+1$ and in the ahMPM refers to the corresponding row, while j refers to stage in occasion t and in an ahMPM refers to the corresponding column. The term $\langle \mathbf{w}, \mathbf{v} \rangle$ refers to the scalar product of these vectors.

In the hMPM case, the basic method is essentially the same, although the equation itself changes somewhat to account for stage l in occasion $t-1$. Thus, the sensitivity of λ to each matrix element in an hMPM is given as

$$\frac{\partial \lambda}{\partial a_{kjl}} = \frac{\bar{v}_{kj} w_{jl}}{\langle \mathbf{w}, \mathbf{v} \rangle} \quad (8.7)$$

Here, k and j are as before, l is stage in occasion $t-1$, kj refers both to the stage pair in occasions $t+1$ and t as well as the corresponding row in the historical matrix, and jl refers both to the stage pair in occasions t and $t-1$ as well as the corresponding column in the historical matrix. Historically-corrected sensitivities may also be estimated for the basic stages of the life history model using the historically-corrected stable stage distributions and reproductive values given in equations 8.4 and 8.5.

Let's analyze the sensitivity of λ to the elements of our raw *Cypridium candidum* MPMs. First, let's look at the raw ahistorical MPM.

```
tm2sens_r <- sensitivity3(cyp2r_mean)
tm2sens_r
> $h_sensmats
> NULL
>
> $ah_sensmats
> $ah_sensmats[[1]]
>      [,1]      [,2]      [,3]      [,4]      [,5]
> [1,] 7.251380e-03 7.385393e-03 6.824496e-04 6.306198e-05 3.054770e-06
> [2,] 7.267251e-02 7.401557e-02 6.839433e-03 6.320001e-04 3.061456e-05
> [3,] 7.864537e-01 8.009882e-01 7.401557e-02 6.839433e-03 3.313073e-04
> [4,] 8.510913e+00 8.668204e+00 8.009882e-01 7.401557e-02 3.585370e-03
> [5,] 1.842083e+02 1.876126e+02 1.733641e+01 1.601976e+00 7.760094e-02
> [6,] 2.835355e+02 2.887756e+02 2.668440e+01 2.465781e+00 1.194443e-01
> [7,] 2.726598e+02 2.776988e+02 2.566085e+01 2.371199e+00 1.148627e-01
> [8,] 4.016999e+02 4.091237e+02 3.780521e+01 3.493403e+00 1.692231e-01
> [9,] 6.177489e+02 6.291656e+02 5.813825e+01 5.372284e+00 2.602375e-01
> [10,] 9.311208e+02 9.483289e+02 8.763064e+01 8.097538e+00 3.922508e-01
> [11,] 1.303494e+03 1.327584e+03 1.226758e+02 1.133590e+01 5.491196e-01
>      [,6]      [,7]      [,8]      [,9]      [,10]
> [1,] 6.636226e-07 4.249376e-06 4.694769e-06 1.494397e-06 7.603360e-07
> [2,] 6.650750e-06 4.258676e-05 4.705044e-05 1.497667e-05 7.620002e-06
> [3,] 7.197367e-05 4.608692e-04 5.091746e-04 1.620759e-04 8.246280e-05
> [4,] 7.788909e-04 4.987474e-03 5.510230e-03 1.753967e-03 8.924031e-04
> [5,] 1.685814e-02 1.079478e-01 1.192622e-01 3.796246e-02 1.931497e-02
> [6,] 2.594825e-02 1.661545e-01 1.835697e-01 5.843227e-02 2.972983e-02
```

```

> [7,] 2.495293e-02 1.597811e-01 1.765284e-01 5.619094e-02 2.858946e-02
> [8,] 3.676226e-02 2.353998e-01 2.600730e-01 8.278409e-02 4.211983e-02
> [9,] 5.653436e-02 3.620066e-01 3.999499e-01 1.273085e-01 6.477344e-02
> [10,] 8.521313e-02 5.456454e-01 6.028365e-01 1.918895e-01 9.763172e-02
> [11,] 1.192915e-01 7.638596e-01 8.439226e-01 2.686299e-01 1.366765e-01
>
> [,11]
> [1,] 1.243806e-07
> [2,] 1.246528e-06
> [3,] 1.348978e-05
> [4,] 1.459849e-04
> [5,] 3.159665e-03
> [6,] 4.863393e-03
> [7,] 4.676844e-03
> [8,] 6.890227e-03
> [9,] 1.059605e-02
> [10,] 1.597121e-02
> [11,] 2.235841e-02
>
>
> $hstages
> NULL
>
> $ahstages
>   stage_id stage original_size original_size_b original_size_c min_age max_age
> 1         1   SD             0.0              NA              NA         0     NA
> 2         2   P1             0.0              NA              NA         0     NA
> 3         3   P2             0.0              NA              NA         0     NA
> 4         4   P3             0.0              NA              NA         0     NA
> 5         5   SL             0.0              NA              NA         0     NA
> 6         6   D              0.0              NA              NA         0     NA
> 7         7  XSm             1.0              NA              NA         0     NA
> 8         8   Sm             3.0              NA              NA         0     NA
> 9         9   Md             6.0              NA              NA         0     NA
> 10        10  Lg            11.0              NA              NA         0     NA
> 11        11  XLg            19.5              NA              NA         0     NA
>
>   repstatus obsstatus propstatus immstatus matstatus entrystage indataset
> 1         0         0           1           0           0           1           0
> 2         0         0           0           1           0           1           0
> 3         0         0           0           1           0           0           0
> 4         0         0           0           1           0           0           0
> 5         0         0           0           1           0           0           0
> 6         0         0           0           0           1           0           1
> 7         1         1           0           0           1           0           1
> 8         1         1           0           0           1           0           1
> 9         1         1           0           0           1           0           1
> 10        1         1           0           0           1           0           1
> 11        1         1           0           0           1           0           1
>
>   binhalfwidth_raw sizebin_min sizebin_max sizebin_center sizebin_width
> 1                 0.0           0.0           0.0           0.0           0
> 2                 0.0           0.0           0.0           0.0           0
> 3                 0.0           0.0           0.0           0.0           0

```

```

> 4      0.0      0.0      0.0      0.0      0
> 5      0.0      0.0      0.0      0.0      0
> 6      0.5     -0.5      0.5      0.0      1
> 7      0.5      0.5      1.5      1.0      1
> 8      1.5      1.5      4.5      3.0      3
> 9      1.5      4.5      7.5      6.0      3
> 10     3.5      7.5     14.5     11.0     7
> 11     5.0     14.5     24.5     19.5    10
> binhalfwidthb_raw sizebinb_min sizebinb_max sizebinb_center sizebinb_width
> 1      NA      NA      NA      NA      NA
> 2      NA      NA      NA      NA      NA
> 3      NA      NA      NA      NA      NA
> 4      NA      NA      NA      NA      NA
> 5      NA      NA      NA      NA      NA
> 6      NA      NA      NA      NA      NA
> 7      NA      NA      NA      NA      NA
> 8      NA      NA      NA      NA      NA
> 9      NA      NA      NA      NA      NA
> 10     NA      NA      NA      NA      NA
> 11     NA      NA      NA      NA      NA
> binhalfwidthc_raw sizebinc_min sizebinc_max sizebinc_center sizebinc_width
> 1      NA      NA      NA      NA      NA
> 2      NA      NA      NA      NA      NA
> 3      NA      NA      NA      NA      NA
> 4      NA      NA      NA      NA      NA
> 5      NA      NA      NA      NA      NA
> 6      NA      NA      NA      NA      NA
> 7      NA      NA      NA      NA      NA
> 8      NA      NA      NA      NA      NA
> 9      NA      NA      NA      NA      NA
> 10     NA      NA      NA      NA      NA
> 11     NA      NA      NA      NA      NA
> group              comments alive almostborn
> 1      0              Dormant seed      1      0
> 2      0              1st yr protocorm  1      0
> 3      0              2nd yr protocorm  1      0
> 4      0              3rd yr protocorm  1      0
> 5      0              Seedling      1      0
> 6      0              Dormant adult   1      0
> 7      0      Extra small adult (1 shoot) 1      0
> 8      0      Small adult (2-4 shoots)  1      0
> 9      0      Medium adult (5-7 shoots) 1      0
> 10     0      Large adult (8-14 shoots)  1      0
> 11     0      Extra large adult (>14 shoots) 1      0
>
> attr(,"class")
> [1] "lefkoSens"

```

The output is a list with several elements. The first element is called `h_sensmats`, and it would hold the sensitivity matrices of the A matrices in the `lefkMat` object input if our MPM was historical. Here, our MPM is actually ahistorical, so this element is empty. The second element, `ah_sensmats`, is a list of sensitivity matrices with each matrix corresponding to each matrix of our input MPM in order.

If we had input a historical MPM, then this element would show matrices of the historically-corrected sensitivities. The order of stage pairs would be shown in element `h_stages` if the input MPM was historical. Element `ah_stages` always shows the stageframe underlying the MPM.

Sensitivity matrices can be hard to read, because all of the elements are estimated to have non-zero sensitivities even if the matrix element must equal zero. It usually helps to analyze such matrices with particular questions in mind. For example, what element is λ most sensitive to changes in? We can address this by searching for the maximum sensitivity element, as below.

```
# The highest deterministic sensitivity value
max(tm2sens_r$ah_sensmats[[1]][which(cyp2r_mean$A[[1]] > 0)])
> [1] 1.601976

# This value is associated with element
max_sens <- intersect(which(tm2sens_r$ah_sensmats[[1]] ==
  max(tm2sens_r$ah_sensmats[[1]][which(cyp2r_mean$A[[1]] > 0)])),
  which(cyp2r_mean$A[[1]] > 0))
max_sens
> [1] 38

# This element is associated with column
ceiling(max_sens / dim(tm2sens_r$ahstages)[1])
> [1] 4
# and row
max_sens %% dim(tm2sens_r$ahstages)[1]
> [1] 5
```

The highest sensitivity value appears to be associated with element 38, which is the transition from 3rd year protocorm to seedling. This element is one that we included as a given rate via supplement table, and its high sensitivity means that these assumed, fixed transition probabilities have a great impact on our analyses.

How does all of this compare with the historical sensitivity analysis? Let's take a look, remembering that we will now get a historical sensitivity matrix with 121 rows and columns each corresponding to stage pairs, as well as a historically-corrected sensitivity matrix with 11 rows and columns each corresponding to stages.

```
tm3sens_r <- sensitivity3(cyp3r_mean)

# Highest sensitivity in the historical matrix
max(tm3sens_r$h_sensmats[[1]][which(cyp3r_mean$A[[1]] > 0)])
> [1] 1.210279

# This value is associated with element
max_sens1 <- intersect(which(tm3sens_r$h_sensmats[[1]] ==
  max(tm3sens_r$h_sensmats[[1]][which(cyp3r_mean$A[[1]] > 0)])),
  which(cyp3r_mean$A[[1]] > 0))
max_sens1
> [1] 3063

# This element is associated with column
ceiling(max_sens1 / dim(tm3sens_r$hstages)[1])
> [1] 26
# and row
```

```

max_sens1 %% dim(tm3sens_r$hstages)[1]
> [1] 38

# Highest sensitivity in the historically-corrected matrix
max(tm3sens_r$ah_sensmats[[1]][which(cyp2r_mean$A[[1]] > 0)])
> [1] 1.210279

# This value is associated with element
max_sens2 <- intersect(which(tm3sens_r$ah_sensmats[[1]] ==
  max(tm3sens_r$ah_sensmats[[1]][which(cyp2r_mean$A[[1]] > 0)])),
  which(cyp2r_mean$A[[1]] > 0))
max_sens2
> [1] 38

# This element is associated with column
ceiling(max_sens2 / dim(tm3sens_r$hstages)[1])
> [1] 4
# and row
max_sens2 %% dim(tm3sens_r$hstages)[1]
> [1] 5

```

Here we find that λ in the hMPM is most sensitive to element 3063. This corresponds to the the transition from the 26th stage pair (2nd year protocorm in time $t-1$ and 3rd year protocorm in time t), to the 38th stage pair (3rd year protocorm in time t and seedling in time $t+1$). Further, the historically-corrected highest sensitivity is the same transition as in the ahistorical case. So, once again we find that we find here that early-life transitions have strong additive influences on λ .

Incidentally, the user may wish to know how we can easily understand which a transition each element corresponds to. For example, how do we know that element 3063 corresponds to the transition from the 26th stage pair to the 38th stage pair? The element index 3063 assumes a column vectorized matrix, meaning that we can think of the matrix as a column vector where the left-most column is stacked on top of the second column, and this combined vector is stacked on top of the third column, etc. In the historical matrix, there are 121 rows and columns, so to determine which column we are in, we need to divide the element index by the number of columns and round up to the nearest integer, as below.

```

ceiling(3063 / 121)
> [1] 26

```

To determine the row, we divide the matrix element index by the number of columns and take the remainder.

```

3063 %% 121
> [1] 38

```

8.4 Elasticity analysis

Sensitivities show how much a small, *additive* change in a matrix element can alter λ . They do so by estimating the local slope of λ given in terms of the matrix element (Caswell, 2001). Problematically, sensitivities are also typically non-zero even for impossible transitions represented by zeros in the matrix. Elasticities, in contrast, show the influence of a small, *proportional* change in a matrix element

on λ , essentially given as the local slope of $\log \lambda$ given in terms of the log of the matrix element. This results in impossible transitions yielding elasticity values of zero, although it also yields somewhat different inferences from sensitivity analysis.

In the ahistorical case, the elasticity, e_{kj} , of λ to change in a matrix element a_{kj} is given as the following.

$$e_{kj} = \frac{a_{kj}}{\lambda} \frac{\partial \lambda}{\partial a_{kj}} \quad (8.8)$$

The historical case is just an extension of the above to include stage l in occasion $t-1$.

$$e_{kjl} = \frac{a_{kjl}}{\lambda} \frac{\partial \lambda}{\partial a_{kjl}} \quad (8.9)$$

Here, we use row and column definitions equivalent to those used in the historical sensitivity case. This makes the elasticity a function of the sensitivity of λ to the matrix element, as well as of the value of the matrix element itself. Thus, structural zeros in the hMPM must have elasticity equal to 0, although it is entirely possible that they have non-zero sensitivities.

Elasticities calculated for hMPMs can be compared to those calculated in ahMPMs easily because elasticities may be added by stage or transition type, with all elasticities corresponding to a specific matrix necessarily summing to 1.0. We can calculate the stage pair elasticities resulting from a historical MPM analysis in the following manner, with all symbol definitions as before.

$$e_{kj} = \sum_{i=1}^m e_{kji} \quad (8.10)$$

These stage pair elasticity values are historically-corrected, and so can be added together in matrices of equivalent dimension to the ahistorical matrices and compared against the latter. We have provided a `summary()` function for elasticity output in `lefk03` that automatically groups the resulting elasticity values by the kind of ahistorical or historical transition.

Let's now assess the elasticity of λ to matrix elements in our MPMs, comparing the ahistorical to the historically-corrected case. We'll start off by just looking at the ahistorical case.

```
tm2elas_r <- elasticity3(cyp2r_mean)
tm2elas_r
> $h_elasmats
> NULL
>
> $ah_elasmats
> $ah_elasmats[[1]]
>           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
> [1,] 0.0005360529 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
> [2,] 0.0067153268 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
> [3,] 0.0000000000 0.07401557 0.00000000 0.00000000 0.00000000 0.00000000
> [4,] 0.0000000000 0.00000000 0.07401557 0.00000000 0.00000000 0.00000000
> [5,] 0.0000000000 0.00000000 0.00000000 0.07401557 0.003585370 0.00000000
> [6,] 0.0000000000 0.00000000 0.00000000 0.00000000 0.005105945 0.001598504
> [7,] 0.0000000000 0.00000000 0.00000000 0.00000000 0.038053219 0.007493797
> [8,] 0.0000000000 0.00000000 0.00000000 0.00000000 0.030856409 0.011606513
> [9,] 0.0000000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
> [10,] 0.0000000000 0.00000000 0.00000000 0.00000000 0.00000000 0.005249431
> [11,] 0.0000000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
>           [,7]      [,8]      [,9]     [,10]     [,11]
> [1,] 0.000174604 0.002179009 0.002174921 0.001497188 0.0006896056
```

```

> [2,] 0.001749861 0.021837778 0.021796810 0.015004649 0.0069111490
> [3,] 0.000000000 0.000000000 0.000000000 0.000000000 0.0000000000
> [4,] 0.000000000 0.000000000 0.000000000 0.000000000 0.0000000000
> [5,] 0.000000000 0.000000000 0.000000000 0.000000000 0.0000000000
> [6,] 0.010146698 0.009097098 0.000000000 0.000000000 0.0000000000
> [7,] 0.075620578 0.033709675 0.004903878 0.000000000 0.0000000000
> [8,] 0.061318847 0.126000606 0.025286479 0.005004124 0.0000000000
> [9,] 0.006386165 0.052442610 0.056793876 0.011685801 0.0000000000
> [10,] 0.004384395 0.014806203 0.016352487 0.052411728 0.0044274750
> [11,] 0.000000000 0.000000000 0.000000000 0.012028230 0.0103301811
>
>
> $hstages
> NULL
>
> $agestages
> X1
> 1 NA
>
> $ahstages
>   stage_id stage original_size original_size_b original_size_c min_age max_age
> 1         1   SD             0.0              NA              NA         0     NA
> 2         2   P1             0.0              NA              NA         0     NA
> 3         3   P2             0.0              NA              NA         0     NA
> 4         4   P3             0.0              NA              NA         0     NA
> 5         5   SL             0.0              NA              NA         0     NA
> 6         6   D              0.0              NA              NA         0     NA
> 7         7   XSm            1.0              NA              NA         0     NA
> 8         8   Sm             3.0              NA              NA         0     NA
> 9         9   Md             6.0              NA              NA         0     NA
> 10        10  Lg            11.0              NA              NA         0     NA
> 11        11  XLg            19.5              NA              NA         0     NA
>
>   repstatus obsstatus propstatus immstatus matstatus entrystage indataset
> 1           0          0           1           0           0           1           0
> 2           0          0           0           1           0           1           0
> 3           0          0           0           1           0           0           0
> 4           0          0           0           1           0           0           0
> 5           0          0           0           1           0           0           0
> 6           0          0           0           0           1           0           1
> 7           1          1           0           0           1           0           1
> 8           1          1           0           0           1           0           1
> 9           1          1           0           0           1           0           1
> 10          1          1           0           0           1           0           1
> 11          1          1           0           0           1           0           1
>
>   binhalfwidth_raw sizebin_min sizebin_max sizebin_center sizebin_width
> 1                 0.0           0.0           0.0           0.0           0
> 2                 0.0           0.0           0.0           0.0           0
> 3                 0.0           0.0           0.0           0.0           0
> 4                 0.0           0.0           0.0           0.0           0
> 5                 0.0           0.0           0.0           0.0           0
> 6                 0.5           -0.5           0.5           0.0           1

```



```

> 7          0.5          0.5          1.5          1.0          1
> 8          1.5          1.5          4.5          3.0          3
> 9          1.5          4.5          7.5          6.0          3
> 10         3.5          7.5          14.5         11.0         7
> 11         5.0          14.5         24.5         19.5         10
>   binhalfwidthb_raw sizebinb_min sizebinb_max sizebinb_center sizebinb_width
> 1             NA             NA             NA             NA             NA
> 2             NA             NA             NA             NA             NA
> 3             NA             NA             NA             NA             NA
> 4             NA             NA             NA             NA             NA
> 5             NA             NA             NA             NA             NA
> 6             NA             NA             NA             NA             NA
> 7             NA             NA             NA             NA             NA
> 8             NA             NA             NA             NA             NA
> 9             NA             NA             NA             NA             NA
> 10            NA             NA             NA             NA             NA
> 11            NA             NA             NA             NA             NA
>   binhalfwidthc_raw sizebinc_min sizebinc_max sizebinc_center sizebinc_width
> 1             NA             NA             NA             NA             NA
> 2             NA             NA             NA             NA             NA
> 3             NA             NA             NA             NA             NA
> 4             NA             NA             NA             NA             NA
> 5             NA             NA             NA             NA             NA
> 6             NA             NA             NA             NA             NA
> 7             NA             NA             NA             NA             NA
> 8             NA             NA             NA             NA             NA
> 9             NA             NA             NA             NA             NA
> 10            NA             NA             NA             NA             NA
> 11            NA             NA             NA             NA             NA
>   group          comments alive almostborn
> 1     0          Dormant seed      1         0
> 2     0          1st yr protocorm  1         0
> 3     0          2nd yr protocorm  1         0
> 4     0          3rd yr protocorm  1         0
> 5     0          Seedling          1         0
> 6     0          Dormant adult     1         0
> 7     0  Extra small adult (1 shoot) 1         0
> 8     0    Small adult (2-4 shoots)  1         0
> 9     0   Medium adult (5-7 shoots)  1         0
> 10    0   Large adult (8-14 shoots)  1         0
> 11    0 Extra large adult (>14 shoots) 1         0
>
> attr(,"class")
> [1] "lefkoeLas"

```

The output is a list with a similar order to the output of the `sensitivity3()` function, although here we see elasticity values instead of sensitivity values. The elasticity matrix shown in element `ah_elasmats` has many zeros, and these zeros correspond to zero elements in the original matrix. As before, we may wish to know which element λ is most elastic to.

```

max_elas <- which(tm2elas_r$ah_elasmats[[1]] == max(tm2elas_r$ah_elasmats[[1]]))

# Highest ahistorical elasticity is
tm2elas_r$ah_elasmats[[1]][max_elas]
> [1] 0.1260006

# This value is associated with element
max_elas
> [1] 85

# This element is associated with column
ceiling(max_elas / dim(tm2elas_r$ahstages)[1])
> [1] 8
# and row
max_elas %% dim(tm2elas_r$ahstages)[1]
> [1] 8

```

Interestingly, our ahistorical elasticity analysis suggests that λ is most elastic to element 85, which refers to stasis as a small adult. So, we see greater importance of adult stages in our elasticity analysis, rather than juvenile stages as we say with sensitivity analysis. But let's see how far this pattern holds.

Let's now compare to the historical case. First, we will conduct the elasticity analysis. As the output is far too large for the printed page, we have hashtagged the output, but feel free to remove hashtag in the second line and see what the output looks like.

```

tm3elas_r <- elasticity3(cyp3r_mean)
#tm3elas_r

```

The output is a good deal longer in this case, because the historical MPM yields both a historical elasticity matrix and a historically-corrected matrix. The output must also hold the index to historical stage pairs and the original stageframe. Let's take a look at which elements exert the strongest proportional influence on λ .

```

max_elas1 <- which(tm3elas_r$h_elasmats[[1]] == max(tm3elas_r$h_elasmats[[1]]))
max_elas2 <- which(tm3elas_r$ah_elasmats[[1]] == max(tm3elas_r$ah_elasmats[[1]]))

# Highest historical elasticity is
tm3elas_r$h_elasmats[[1]][max_elas1]
> [1] 0.1238486
#This value is associated with element
max_elas1
> [1] 10249

# This element is associated with column
ceiling(max_elas1 / dim(tm3elas_r$hstages)[1])
> [1] 85
# and row
max_elas1 %% dim(tm3elas_r$hstages)[1]
> [1] 85

# Highest historically-corrected elasticity is
tm3elas_r$ah_elasmats[[1]][max_elas2]
> [1] 0.1968263

```

```

# This value is associated with element
max_elas2
> [1] 85

# This element is associated with column
ceiling(max_elas2 / dim(tm3elas_r$ahstages)[1])
> [1] 8
# and row
max_elas2 %% dim(tm3elas_r$ahstages)[1]
> [1] 8

```

Here we see that ahistorical and historical analyses generally agree about which transitions λ is most elastic in response to: stasis as a small adult. This is a very different result from the ahistorical sensitivity analysis, and is worth exploring.

Now let's compare the elasticity of population growth rate in relation to the core life history stages, via a bar plot comparison (figure 8.4).

```

elas_put_together <- cbind.data.frame(colSums(tm2elas_r$ah_elasmats[[1]]),
  colSums(tm3elas_r$ah_elasmats[[1]]))
names(elas_put_together) <- c("ahistorical", "historical")
rownames(elas_put_together) <- tm2elas_r$ahstages$stage

barplot(t(elas_put_together), beside=T, ylab = "Elasticity", xlab = "Stage",
  col = c("black", "orangered"), bty = "n")
legend("topleft", c("ahistorical", "historical"),
  col = c("black", "orangered"), pch = 15, bty = "n")

```

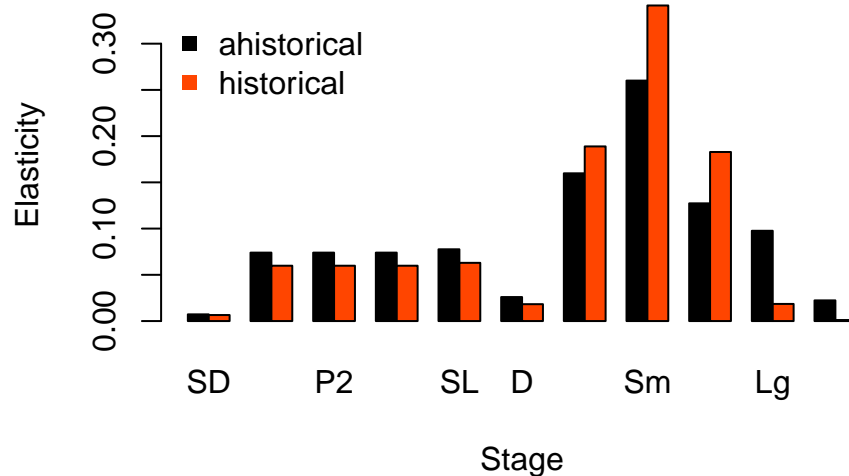


Figure 8.4: Ahistorical vs. historically-corrected deterministic elasticity to stage

Elasticity patterns in the two analyses look somewhat different. Both ahistorical and historical analyses show that λ is most elastic to shifts in transitions from the small adult stage, followed by the extra small and medium adult stages. However, historical analysis suggests a far greater importance to the small adult stage, and generally lower importance to juvenile stages and to large and extra large adults. So, once again, we see an important impact of individual history here.

Finally, let's look at the elasticity sums of different transition types (figure 8.5).

```

tm2elas_r_sums <- summary(tm2elas_r)
tm3elas_r_sums <- summary(tm3elas_r)

elas_sums_together <- cbind.data.frame(tm2elas_r_sums$ahist[,2],
  tm3elas_r_sums$ahist[,2])
names(elas_sums_together) <- c("ahistorical", "historical")
rownames(elas_sums_together) <- tm2elas_r_sums$ahist$category

barplot(t(elas_sums_together), beside = T, ylab = "Elasticity",
  xlab = "Transition", col = c("black", "orangered"), bty = "n")
legend("topright", c("ahistorical", "historical"),
  col = c("black", "orangered"), pch = 15, bty = "n")

```

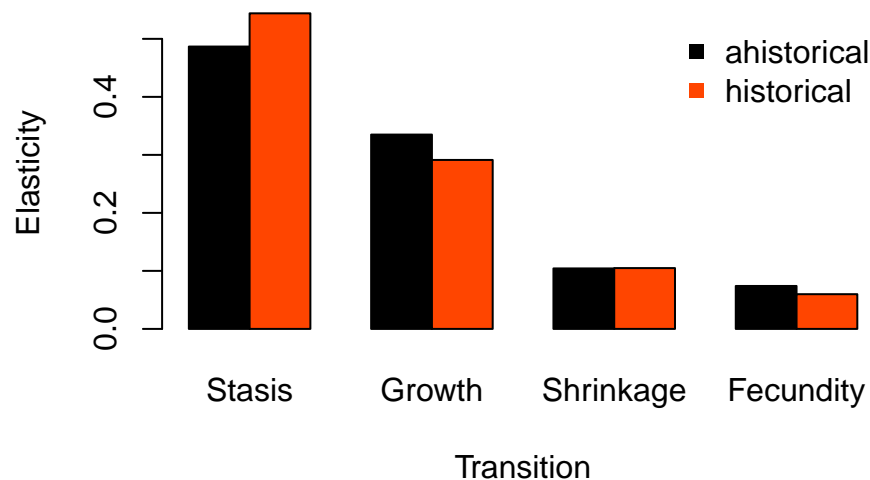


Figure 8.5: Ahistorical vs. historically-corrected elasticity to transitions

Fecundity has the least impact in both cases, although it does influence ahistorical analysis a reasonable amount. Stasis, in contrast, exerts the greatest influence, while growth is more influential than shrinkage and fecundity. We can also look at historical transitions, as below (figure 8.6).

```

elas_hist2plot <- as.matrix(tm3elas_r_sums$hist[,2])
rownames(elas_hist2plot) <- tm3elas_r_sums$hist$category

par(mar = c(7, 4, 2, 2) + 0.2)
barplot(t(elas_hist2plot), ylab = "Elasticity", xlab = "", xaxt = "n",
  col = "orangered", bty = "n")
text(cex=0.6, x=seq(from = -0.3, to = 1.15*length(tm3elas_r_sums$hist$category),
  by = 1.17), y=-0.08, tm3elas_r_sums$hist$category, xpd=TRUE, srt=45)

```

We can see that full stasis (i.e. staying in the same stage from occasion $t-1$ to $t+1$) is associated with the greatest summed elasticity values, and growth from occasion $t-1$ to t followed by stasis to occasion $t+1$ is the next most important. Transitions associated with fecundity are associated with the lowest summed elasticity values, except for fecundity followed by growth.

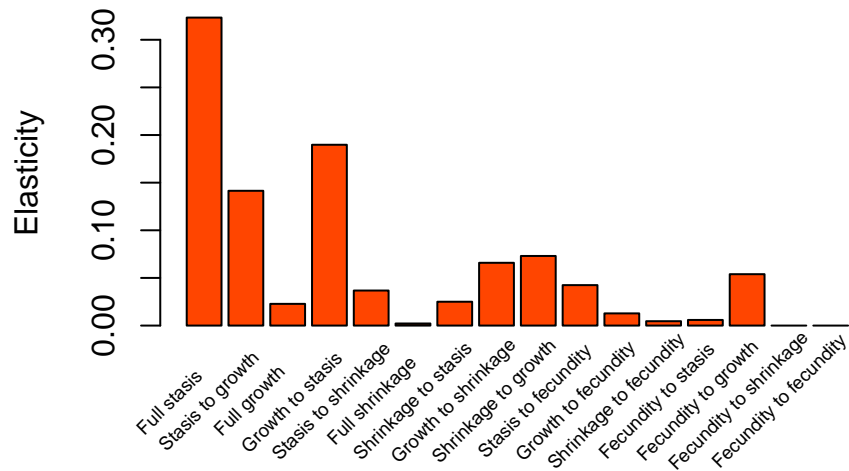


Figure 8.6: Elasticity of lambda to historical transitions

8.5 Further analyses

Deterministic life table response experiments (LTREs) are also possible in `lefk3`, and we describe them in a later chapter 11.

8.6 Points to remember

1. Deterministic analysis refers to matrix projection in which a single matrix is projected forward indefinitely. Its asymptotic properties yield values such as the stable stage structure and reproductive values of each stage, as well as the population growth rate under the stable structure.
2. Package `lefk3` can conduct all major deterministic analyses even for large numbers of massive matrices, such as historical IPMs and age-by-stage MPMs. This includes even sensitivity and elasticity analyses.
3. A simple means of assessing the influence of history on population dynamics is through the comparison of ahistorical projection analyses to the equivalent historically-corrected analyses, since the latter interprets the results of historical analyses but framed within the stageframe and matrix dimensions of the former.