Chapter 12

Further Issues I: Importing Matrices and Editing MPMs

"We all reinvent our pasts."

— John le Carré

Package `lefko3` includes powerful functions to create MPMs of all major kinds. However, suppose that you already have matrices, and wish to use analyze them with `lefko3`. How can these matrices be imported properly? Our package includes functions that allow users to build `lefkoMat` objects with already existing matrices. Once these `lefkoMat` objects are built, they may be applied to the various functions found in the package and analyzed.

To start, we will need to get whatever matrices we have into R itself. This may or may not be easy, depending on what format the matrices are in. Users will find the easiest time with matrices saved in R object files, such as .Rda or .Rdata files. These files may be loaded into memory with the `load()` function. Other users may be working in R with some sort of MPM database, like COMPADRE (Salguero-Gómez et al., 2015). In this case, they can follow the instructions for using that database to extract matrices from the database for their own use. In other cases, users may need to import from a separate file. In these cases, one approach that usually works is to use the `scan()` function together with `matrix()`, for example as `matrix_a <- matrix(scan("myfile.txt"), nrow = 10)`. This function call will import a single matrix in a standard tab-delimited text file as a vector, and then use it to create a matrix with ten rows (this assumes that the right number of elements exists to import into a ten-row matrix).

Regardless of how you accomplish this step, the goal is to take your matrix and turn it into a matrix class object in R. If a matrix or series of matrices is not too large, then perhaps the simplest way to accomplish this is to enter the matrix manually.

12.1 Creating a new lefkoMat object from imported matrices

Let's take an example using a published set of matrices. Here, we will recreate our data object `anthyllis` (section 1.6.3). Davison et al. (2010) reported stochastic contributions made by differences in vital rate means and variances among nine natural populations of the perennial herb *Anthyllis vulneraria*, also known as kidney vetch, which grows in calcareous grasslands in the Viroin Valley of southwestern Belgium. *A. vulneraria* is a grassland specialist and the unique host plant of the Red-listed blue butterfly (*Cupido minimus*). It is a short-lived, rosette-forming legume with a complex life cycle including stasis and retrogression between four stages but no seedbank (seedlings, juveniles, small adults and large adults; figure 12.1).

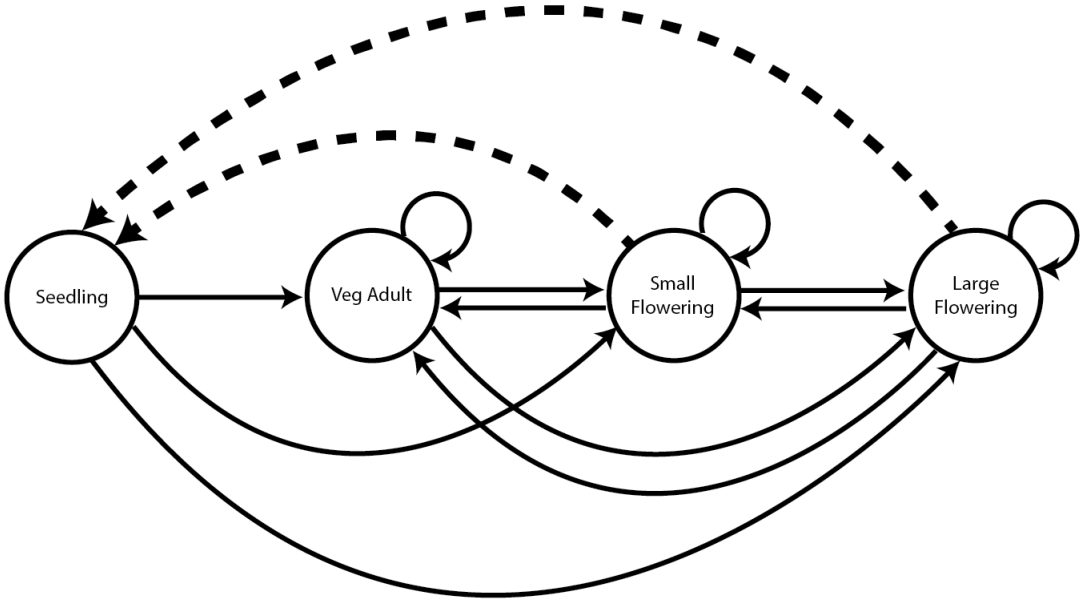

Figure 12.1: Life history model of *Anthyllis vulneraria*. Solid arrows indicate survival transitions while dashed arrows indicate fecundity transitions.

Nine populations (N = 27-50,000) growing in distinct grassland fragments were surveyed between 2003 and 2006, yielding three (4x4) annual transition matrices for each population. The populations occurred within grassland fragments, and were mostly managed as nature reserves through rotational sheep grazing. These surveys coincided with a Summer heat wave (2003), followed by a Spring drought (2005) and an even more extreme heat wave (2006). These populations have been subject to detailed study for aspects of their genetics and demography, and further details on the sites can be obtained through the resulting publications (Krauss et al., 2004; Honnay et al., 2006; Piessens et al., 2009).

Our goal in this exercise will be to import the published MPMs available for these nine populations of *Anthyllis vulneraria*, and to create a `lefkoMat` object for further study.

The most important part of importing matrices into `lefko3` is also to import the underlying life history model. This will require carefully reading whatever material describes the analyses, whether a published research paper, an unpublished thesis, or another source, and carefully listing the stages, their descriptions, and their relationships vis-à-vis survival transitions and fecundity rates. In the case of our *Anthyllis* analysis, this means developing the life history model in the figure above, and then creating a stageframe to match it. Since we do not have the original demographic dataset that produced the published matrices, we do not need to know the exact sizes of plants matching the small and large adult stages. So, we will use proxy values. These proxy values need to be unique and non-negative, and they need non-overlapping bins usable as size classes defining each stage. However, since we are not analyzing size itself here, they do not need any further basis in reality. Other characteristics must be exact and realistic to make sure that the analyses work properly, including all other stage descriptions such as reproductive status, propagule status, and observation status.

```
sizevector <- c(1, 1, 2, 3) # These sizes are not from the original paper
stagevector <- c("Sdl", "Veg", "SmFlo", "LFlo")
repvector <- c(0, 0, 1, 1)
obsvector <- c(1, 1, 1, 1)
```

```
matvector <- c(0, 1, 1, 1)
immvector <- c(1, 0, 0, 0)
propvector <- c(0, 0, 0, 0)
indataset <- c(1, 1, 1, 1)
binvec <- c(0.5, 0.5, 0.5, 0.5)
comments <- c("Seedling", "Vegetative adult", "Small flowering",
  "Large flowering")

anthframe <- sf_create(sizes = sizevector, stagenames = stagevector,
  repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
  immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
  propstatus = propvector, comments = comments)
anthframe
>   stage size size_b size_c min_age max_age repstatus obsstatus propstatus
> 1   Sdl    1     NA     NA      NA      NA         0         1          0
> 2   Veg    1     NA     NA      NA      NA         0         1          0
> 3 SmFlo    2     NA     NA      NA      NA         1         1          0
> 4  LFlo    3     NA     NA      NA      NA         1         1          0
>   immstatus matstatus indataset binhalfwidth_raw sizebin_min sizebin_max
> 1         1         0         1              0.5         0.5         1.5
> 2         0         1         1              0.5         0.5         1.5
> 3         0         1         1              0.5         1.5         2.5
> 4         0         1         1              0.5         2.5         3.5
>   sizebin_center sizebin_width binhalfwidthb_raw sizebinb_min sizebinb_max
> 1              1             1                NA           NA           NA
> 2              1             1                NA           NA           NA
> 3              2             1                NA           NA           NA
> 4              3             1                NA           NA           NA
>   sizebinb_center sizebinb_width binhalfwidthc_raw sizebinc_min sizebinc_max
> 1              NA             NA                NA           NA           NA
> 2              NA             NA                NA           NA           NA
> 3              NA             NA                NA           NA           NA
> 4              NA             NA                NA           NA           NA
>   sizebinc_center sizebinc_width group         comments
> 1              NA             NA     0         Seedling
> 2              NA             NA     0 Vegetative adult
> 3              NA             NA     0  Small flowering
> 4              NA             NA     0  Large flowering
```

Next we can enter the matrices, which were published in Davison et al. (2010). We will enter these matrices as standard R `matrix` class objects. All matrices are square with four columns. I personally find it easier to enter matrices by row when copying from text, and so I have chosen to do so here and have indicated so in the `matrix()` function using `byrow = TRUE` (the default is to fill by column). Users familiar and comfortable with Matlab will likely be quite comfortable with filling matrices by row. Here is the first matrix, covering 2003 to 2004 for population C.

```
XC3 <- matrix(c(0, 0, 1.74, 1.74,
  0.208333333, 0, 0, 0.057142857,
  0.041666667, 0.076923077, 0, 0,
  0.083333333, 0.076923077, 0.066666667, 0.028571429), 4, 4, byrow = TRUE)
XC3
>            [,1]       [,2]       [,3]       [,4]
```

```
> [1,] 0.00000000 0.00000000 1.74000000 1.74000000
> [2,] 0.20833333 0.00000000 0.00000000 0.05714286
> [3,] 0.04166667 0.07692308 0.00000000 0.00000000
> [4,] 0.08333333 0.07692308 0.06666667 0.02857143
```

This is an `A` matrix, meaning that it includes all survival-transitions and fecundity for the population as a whole. The corresponding `U` and `F` matrices were not provided in that paper, although it is most likely that the elements valued at 1.74 in the top right-hand corner are only composed of fecundity values while the rest of the matrix is only composed of survival transitions (this might not be the case if clonal reproduction were possible). The order of rows and columns corresponds to the order of stages in the stageframe `anthframe`. Let's now load the remaining matrices.

```
# POPN C 2004-2005
XC4 <- matrix(c(0, 0, 0.3, 0.6,
  0.32183908, 0.142857143, 0, 0,
  0.16091954, 0.285714286, 0, 0,
  0.252873563, 0.285714286, 0.5, 0.6), 4, 4, byrow = TRUE)

# POPN C 2005-2006
XC5 <- matrix(c(0, 0, 0.50625, 0.675,
  0, 0, 0, 0.035714286,
  0.1, 0.068965517, 0.0625, 0.107142857,
  0.3, 0.137931034, 0, 0.071428571), 4, 4, byrow = TRUE)

# POPN E 2003-2004
XE3 <- matrix(c(0, 0, 2.44, 6.569230769,
  0.196428571, 0, 0, 0,
  0.125, 0.5, 0, 0,
  0.160714286, 0.5, 0.133333333, 0.076923077), 4, 4, byrow = TRUE)

# POPN E 2004-2005
XE4 <- matrix(c(0, 0, 0.45, 0.646153846,
  0.06557377, 0.090909091, 0.125, 0,
  0.032786885, 0, 0.125, 0.076923077,
  0.049180328, 0, 0.125, 0.230769231), 4, 4, byrow = TRUE)

# POPN E 2005-2006
XE5 <- matrix(c(0, 0, 2.85, 3.99,
  0.083333333, 0, 0, 0,
  0, 0, 0, 0,
  0.416666667, 0.1, 0, 0.1), 4, 4, byrow = TRUE)

# POPN F 2003-2004
XF3 <- matrix(c(0, 0, 1.815, 7.058333333,
  0.075949367, 0, 0.05, 0.083333333,
  0.139240506, 0, 0, 0.25,
  0.075949367, 0, 0, 0.083333333), 4, 4, byrow = TRUE)

# POPN F 2004-2005
XF4 <- matrix(c(0, 0, 1.233333333, 7.4,
  0.223880597, 0, 0.111111111, 0.142857143,
  0.134328358, 0.272727273, 0.166666667, 0.142857143,
```

```
  0.119402985, 0.363636364, 0.055555556, 0.142857143), 4, 4, byrow = TRUE)

# POPN F 2005-2006
XF5 <- matrix(c(0, 0, 1.06, 3.372727273,
  0.073170732, 0.025, 0.033333333, 0,
  0.036585366, 0.15, 0.1, 0.136363636,
  0.06097561, 0.225, 0.166666667, 0.272727273), 4, 4, byrow = TRUE)

# POPN G 2003-2004
XG3 <- matrix(c(0, 0, 0.245454545, 2.1,
  0, 0, 0.045454545, 0,
  0.125, 0, 0.090909091, 0,
  0.125, 0, 0.090909091, 0.333333333), 4, 4, byrow = TRUE)

# POPN G 2004-2005
XG4 <- matrix(c(0, 0, 1.1, 1.54,
  0.111111111, 0, 0, 0,
  0, 0, 0, 0,
  0.111111111, 0, 0, 0), 4, 4, byrow = TRUE)

# POPN G 2005-2006
XG5 <- matrix(c(0, 0, 0, 1.5,
  0, 0, 0, 0,
  0.090909091, 0, 0, 0,
  0.545454545, 0.5, 0, 0.5), 4, 4, byrow = TRUE)

# POPN L 2003-2004
XL3 <- matrix(c(0, 0, 1.785365854, 1.856521739,
  0.128571429, 0, 0, 0.010869565,
  0.028571429, 0, 0, 0,
  0.014285714, 0, 0, 0.02173913), 4, 4, byrow = TRUE)

# POPN L 2004-2005
XL4 <- matrix(c(0, 0, 14.25, 16.625,
  0.131443299, 0.057142857, 0, 0.25,
  0.144329897, 0, 0, 0,
  0.092783505, 0.2, 0, 0.25), 4, 4, byrow = TRUE)

# POPN L 2005-2006
XL5 <- matrix(c(0, 0, 0.594642857, 1.765909091,
  0, 0, 0.017857143, 0,
  0.021052632, 0.018518519, 0.035714286, 0.045454545,
  0.021052632, 0.018518519, 0.035714286, 0.068181818), 4, 4, byrow = TRUE)

# POPN O 2003-2004
XO3 <- matrix(c(0, 0, 11.5, 2.775862069,
  0.6, 0.285714286, 0.333333333, 0.24137931,
  0.04, 0.142857143, 0, 0,
  0.16, 0.285714286, 0, 0.172413793), 4, 4, byrow = TRUE)

# POPN O 2004-2005
```

```
XO4 <- matrix(c(0, 0, 3.78, 1.225,
  0.28358209, 0.171052632, 0, 0.166666667,
  0.084577114, 0.026315789, 0, 0.055555556,
  0.139303483, 0.447368421, 0, 0.305555556), 4, 4, byrow = TRUE)

# POPN O 2005-2006
XO5 <- matrix(c(0, 0, 1.542857143, 1.035616438,
  0.126984127, 0.105263158, 0.047619048, 0.054794521,
  0.095238095, 0.157894737, 0.19047619, 0.082191781,
  0.111111111, 0.223684211, 0, 0.356164384), 4, 4, byrow = TRUE)

# POPN Q 2003-2004
XQ3 <- matrix(c(0, 0, 0.15, 0.175,
  0, 0, 0, 0,
  0, 0, 0, 0,
  1, 0, 0, 0), 4, 4, byrow = TRUE)

# POPN Q 2004-2005
XQ4 <- matrix(c(0, 0, 0, 0.25,
  0, 0, 0, 0,
  0, 0, 0, 0,
  1, 0.666666667, 0, 1), 4, 4, byrow = TRUE)

# POPN Q 2005-2006
XQ5 <- matrix(c(0, 0, 0, 1.428571429,
  0, 0, 0, 0.142857143,
  0.25, 0, 0, 0,
  0.25, 0, 0, 0.571428571), 4, 4, byrow = TRUE)

# POPN R 2003-2004
XR3 <- matrix(c(0, 0, 0.7, 0.6125,
  0.25, 0, 0, 0.125,
  0, 0, 0, 0,
  0.25, 0.166666667, 0, 0.25), 4, 4, byrow = TRUE)

# POPN R 2004-2005
XR4 <- matrix(c(0, 0, 0, 0.6,
  0.285714286, 0, 0, 0,
  0.285714286, 0.333333333, 0, 0,
  0.285714286, 0.333333333, 0, 1), 4, 4, byrow = TRUE)

# POPN R 2005-2006
XR5 <- matrix(c(0, 0, 0.7, 0.6125,
  0, 0, 0, 0,
  0, 0, 0, 0,
  0.333333333, 0, 0.333333333, 0.625), 4, 4, byrow = TRUE)

# POPN S 2003-2004
XS3 <- matrix(c(0, 0, 2.1, 0.816666667,
  0.166666667, 0, 0, 0,
  0, 0, 0, 0,
```

```
  0, 0, 0, 0.166666667), 4, 4, byrow = TRUE)

# POPN S 2004-2005
XS4 <- matrix(c(0, 0, 0, 7,
  0.333333333, 0.5, 0, 0,
  0, 0, 0, 0,
  0.333333333, 0, 0, 1), 4, 4, byrow = TRUE)

# POPN S 2005-2006
XS5 <- matrix(c(0, 0, 0, 1.4,
  0, 0, 0, 0,
  0, 0, 0, 0.2,
  0.111111111, 0.75, 0, 0.2), 4, 4, byrow = TRUE)
```

Our next step will be to incorporate these matrices into a single list of matrices. As a reminder, a list is an object in which each element can be of a different class and of different length. Let's build our list and then take a look at it.

```
mats_list <- list(XC3, XC4, XC5, XE3, XE4, XE5, XF3, XF4, XF5, XG3, XG4, XG5,
  XL3, XL4, XL5, XO3, XO4, XO5, XQ3, XQ4, XQ5, XR3, XR4, XR5, XS3, XS4, XS5)

mats_list
> [[1]]
>             [,1]       [,2]       [,3]       [,4]
> [1,] 0.00000000 0.00000000 1.74000000 1.74000000
> [2,] 0.20833333 0.00000000 0.00000000 0.05714286
> [3,] 0.04166667 0.07692308 0.00000000 0.00000000
> [4,] 0.08333333 0.07692308 0.06666667 0.02857143
>
> [[2]]
>           [,1]      [,2] [,3] [,4]
> [1,] 0.0000000 0.0000000  0.3  0.6
> [2,] 0.3218391 0.1428571  0.0  0.0
> [3,] 0.1609195 0.2857143  0.0  0.0
> [4,] 0.2528736 0.2857143  0.5  0.6
>
> [[3]]
>      [,1]       [,2]    [,3]       [,4]
> [1,]  0.0 0.00000000 0.50625 0.67500000
> [2,]  0.0 0.00000000 0.00000 0.03571429
> [3,]  0.1 0.06896552 0.06250 0.10714286
> [4,]  0.3 0.13793103 0.00000 0.07142857
>
> [[4]]
>           [,1] [,2]      [,3]       [,4]
> [1,] 0.0000000  0.0 2.4400000 6.56923077
> [2,] 0.1964286  0.0 0.0000000 0.00000000
> [3,] 0.1250000  0.5 0.0000000 0.00000000
> [4,] 0.1607143  0.5 0.1333333 0.07692308
>
> [[5]]
>             [,1]       [,2]  [,3]       [,4]
```

```
> [1,] 0.00000000 0.00000000 0.450 0.64615385
> [2,] 0.06557377 0.09090909 0.125 0.00000000
> [3,] 0.03278689 0.00000000 0.125 0.07692308
> [4,] 0.04918033 0.00000000 0.125 0.23076923
>
> [[6]]
>            [,1] [,2] [,3] [,4]
> [1,] 0.00000000  0.0 2.85 3.99
> [2,] 0.08333333  0.0 0.00 0.00
> [3,] 0.00000000  0.0 0.00 0.00
> [4,] 0.41666667  0.1 0.00 0.10
>
> [[7]]
>            [,1] [,2]  [,3]       [,4]
> [1,] 0.00000000    0 1.815 7.05833333
> [2,] 0.07594937    0 0.050 0.08333333
> [3,] 0.13924051    0 0.000 0.25000000
> [4,] 0.07594937    0 0.000 0.08333333
>
> [[8]]
>           [,1]      [,2]       [,3]      [,4]
> [1,] 0.0000000 0.0000000 1.23333333 7.4000000
> [2,] 0.2238806 0.0000000 0.11111111 0.1428571
> [3,] 0.1343284 0.2727273 0.16666667 0.1428571
> [4,] 0.1194030 0.3636364 0.05555556 0.1428571
>
> [[9]]
>            [,1]  [,2]       [,3]      [,4]
> [1,] 0.00000000 0.000 1.06000000 3.3727273
> [2,] 0.07317073 0.025 0.03333333 0.0000000
> [3,] 0.03658537 0.150 0.10000000 0.1363636
> [4,] 0.06097561 0.225 0.16666667 0.2727273
>
> [[10]]
>       [,1] [,2]       [,3]      [,4]
> [1,] 0.000    0 0.24545454 2.1000000
> [2,] 0.000    0 0.04545454 0.0000000
> [3,] 0.125    0 0.09090909 0.0000000
> [4,] 0.125    0 0.09090909 0.3333333
>
> [[11]]
>           [,1] [,2] [,3] [,4]
> [1,] 0.0000000    0  1.1 1.54
> [2,] 0.1111111    0  0.0 0.00
> [3,] 0.0000000    0  0.0 0.00
> [4,] 0.1111111    0  0.0 0.00
>
> [[12]]
>            [,1] [,2] [,3] [,4]
> [1,] 0.00000000  0.0    0  1.5
> [2,] 0.00000000  0.0    0  0.0
```

```
> [3,] 0.09090909  0.0    0  0.0
> [4,] 0.54545455  0.5    0  0.5
>
> [[13]]
>            [,1] [,2]     [,3]       [,4]
> [1,] 0.00000000    0 1.785366 1.85652174
> [2,] 0.12857143    0 0.000000 0.01086956
> [3,] 0.02857143    0 0.000000 0.00000000
> [4,] 0.01428571    0 0.000000 0.02173913
>
> [[14]]
>            [,1]       [,2]  [,3]   [,4]
> [1,] 0.00000000 0.00000000 14.25 16.625
> [2,] 0.13144330 0.05714286  0.00  0.250
> [3,] 0.14432990 0.00000000  0.00  0.000
> [4,] 0.09278351 0.20000000  0.00  0.250
>
> [[15]]
>            [,1]       [,2]       [,3]       [,4]
> [1,] 0.00000000 0.00000000 0.59464286 1.76590909
> [2,] 0.00000000 0.00000000 0.01785714 0.00000000
> [3,] 0.02105263 0.01851852 0.03571429 0.04545454
> [4,] 0.02105263 0.01851852 0.03571429 0.06818182
>
> [[16]]
>      [,1]      [,2]       [,3]      [,4]
> [1,] 0.00 0.0000000 11.5000000 2.7758621
> [2,] 0.60 0.2857143  0.3333333 0.2413793
> [3,] 0.04 0.1428571  0.0000000 0.0000000
> [4,] 0.16 0.2857143  0.0000000 0.1724138
>
> [[17]]
>            [,1]       [,2] [,3]       [,4]
> [1,] 0.00000000 0.00000000 3.78 1.22500000
> [2,] 0.28358209 0.17105263 0.00 0.16666667
> [3,] 0.08457711 0.02631579 0.00 0.05555556
> [4,] 0.13930348 0.44736842 0.00 0.30555556
>
> [[18]]
>            [,1]      [,2]       [,3]       [,4]
> [1,] 0.00000000 0.0000000 1.54285714 1.03561644
> [2,] 0.12698413 0.1052632 0.04761905 0.05479452
> [3,] 0.09523809 0.1578947 0.19047619 0.08219178
> [4,] 0.11111111 0.2236842 0.00000000 0.35616438
>
> [[19]]
>      [,1] [,2] [,3]  [,4]
> [1,]    0    0 0.15 0.175
> [2,]    0    0 0.00 0.000
> [3,]    0    0 0.00 0.000
> [4,]    1    0 0.00 0.000
```

```
>
> [[20]]
>      [,1]      [,2] [,3] [,4]
> [1,]    0 0.0000000    0 0.25
> [2,]    0 0.0000000    0 0.00
> [3,]    0 0.0000000    0 0.00
> [4,]    1 0.6666667    0 1.00
>
> [[21]]
>      [,1] [,2] [,3]      [,4]
> [1,] 0.00    0    0 1.4285714
> [2,] 0.00    0    0 0.1428571
> [3,] 0.25    0    0 0.0000000
> [4,] 0.25    0    0 0.5714286
>
> [[22]]
>      [,1]      [,2] [,3]   [,4]
> [1,] 0.00 0.0000000  0.7 0.6125
> [2,] 0.25 0.0000000  0.0 0.1250
> [3,] 0.00 0.0000000  0.0 0.0000
> [4,] 0.25 0.1666667  0.0 0.2500
>
> [[23]]
>           [,1]      [,2] [,3] [,4]
> [1,] 0.0000000 0.0000000    0  0.6
> [2,] 0.2857143 0.0000000    0  0.0
> [3,] 0.2857143 0.3333333    0  0.0
> [4,] 0.2857143 0.3333333    0  1.0
>
> [[24]]
>           [,1] [,2]      [,3]   [,4]
> [1,] 0.0000000    0 0.7000000 0.6125
> [2,] 0.0000000    0 0.0000000 0.0000
> [3,] 0.0000000    0 0.0000000 0.0000
> [4,] 0.3333333    0 0.3333333 0.6250
>
> [[25]]
>           [,1] [,2] [,3]      [,4]
> [1,] 0.0000000    0  2.1 0.8166667
> [2,] 0.1666667    0  0.0 0.0000000
> [3,] 0.0000000    0  0.0 0.0000000
> [4,] 0.0000000    0  0.0 0.1666667
>
> [[26]]
>           [,1] [,2] [,3] [,4]
> [1,] 0.0000000  0.0    0    7
> [2,] 0.3333333  0.5    0    0
> [3,] 0.0000000  0.0    0    0
> [4,] 0.3333333  0.0    0    1
>
> [[27]]
```

```
>           [,1] [,2] [,3] [,4]
> [1,] 0.0000000 0.00    0  1.4
> [2,] 0.0000000 0.00    0  0.0
> [3,] 0.0000000 0.00    0  0.2
> [4,] 0.1111111 0.75    0  0.2
```

We might now wish to check that the top-rightmost two elements are fecundity rates. One way of doing this is to replicate this list into another list, then change the two top-rightmost elements to zero, and then check both the range of values in the final matrices and check the column sums. If we are correct, then both the ranges of values and the column sums will be between 0.0 and 1.0. R provides a number of functions to make working with lists relatively easy, and chief among these are the `lapply()` and `unlist()` functions. Let's try this approach.

```
all_potential_Us <- lapply(mats_list, function(X) {
  X[1, 3] <- 0
  X[1, 4] <- 0

  return(X)
})

summary(unlist(all_potential_Us))
>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
> 0.00000 0.00000 0.00000 0.08003 0.10000 1.00000
all_colSums <- lapply(all_potential_Us, colSums)
summary(unlist(all_colSums))
>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
> 0.00000 0.04803 0.29557 0.32012 0.50000 1.00000
```

We see that we are left with values that look like probabilities (i.e. all range from 0.0 to 1.0), so everything looks good.

Next we will incorporate all of these matrices into a `lefkoMat` object. We will create the `lefkoMat` object to hold these matrices by calling function `create_lM()` with the list object we created. We will also include metadata describing the order of populations (here treated as patches), and the order of monitoring occasions.

```
anth_lefkoMat <- create_lM(mats = mats_list, stageframe = anthframe,
  hstages = NA, historical = FALSE, poporder = 1,
  patchorder = c("C", "C", "C", "E", "E", "E", "F", "F", "F", "G", "G", "G",
    "L", "L", "L", "O", "O", "O", "Q", "Q", "Q", "R", "R", "R", "S", "S", "S"),
  yearorder = c(2003, 2004, 2005, 2003, 2004, 2005, 2003, 2004, 2005, 2003,
    2004, 2005, 2003, 2004, 2005, 2003, 2004, 2005, 2003, 2004, 2005, 2003,
    2004, 2005, 2003, 2004, 2005))

anth_lefkoMat
> $A
> $A[[1]]
>             [,1]       [,2]       [,3]       [,4]
> [1,] 0.00000000 0.00000000 1.74000000 1.74000000
> [2,] 0.20833333 0.00000000 0.00000000 0.05714286
> [3,] 0.04166667 0.07692308 0.00000000 0.00000000
> [4,] 0.08333333 0.07692308 0.06666667 0.02857143
>
```

```
> $A[[2]]
>           [,1]      [,2] [,3] [,4]
> [1,] 0.0000000 0.0000000  0.3  0.6
> [2,] 0.3218391 0.1428571  0.0  0.0
> [3,] 0.1609195 0.2857143  0.0  0.0
> [4,] 0.2528736 0.2857143  0.5  0.6
>
> $A[[3]]
>      [,1]       [,2]    [,3]       [,4]
> [1,]  0.0 0.00000000 0.50625 0.67500000
> [2,]  0.0 0.00000000 0.00000 0.03571429
> [3,]  0.1 0.06896552 0.06250 0.10714286
> [4,]  0.3 0.13793103 0.00000 0.07142857
>
> $A[[4]]
>           [,1] [,2]      [,3]       [,4]
> [1,] 0.0000000  0.0 2.4400000 6.56923077
> [2,] 0.1964286  0.0 0.0000000 0.00000000
> [3,] 0.1250000  0.5 0.0000000 0.00000000
> [4,] 0.1607143  0.5 0.1333333 0.07692308
>
> $A[[5]]
>            [,1]       [,2]  [,3]       [,4]
> [1,] 0.00000000 0.00000000 0.450 0.64615385
> [2,] 0.06557377 0.09090909 0.125 0.00000000
> [3,] 0.03278689 0.00000000 0.125 0.07692308
> [4,] 0.04918033 0.00000000 0.125 0.23076923
>
> $A[[6]]
>            [,1] [,2] [,3] [,4]
> [1,] 0.00000000  0.0 2.85 3.99
> [2,] 0.08333333  0.0 0.00 0.00
> [3,] 0.00000000  0.0 0.00 0.00
> [4,] 0.41666667  0.1 0.00 0.10
>
> $A[[7]]
>            [,1] [,2]  [,3]       [,4]
> [1,] 0.00000000    0 1.815 7.05833333
> [2,] 0.07594937    0 0.050 0.08333333
> [3,] 0.13924051    0 0.000 0.25000000
> [4,] 0.07594937    0 0.000 0.08333333
>
> $A[[8]]
>           [,1]      [,2]       [,3]      [,4]
> [1,] 0.0000000 0.0000000 1.23333333 7.4000000
> [2,] 0.2238806 0.0000000 0.11111111 0.1428571
> [3,] 0.1343284 0.2727273 0.16666667 0.1428571
> [4,] 0.1194030 0.3636364 0.05555556 0.1428571
>
> $A[[9]]
>            [,1]  [,2]       [,3]      [,4]
```

```
> [1,] 0.00000000 0.000 1.06000000 3.3727273
> [2,] 0.07317073 0.025 0.03333333 0.0000000
> [3,] 0.03658537 0.150 0.10000000 0.1363636
> [4,] 0.06097561 0.225 0.16666667 0.2727273
>
> $A[[10]]
>       [,1] [,2]       [,3]      [,4]
> [1,] 0.000    0 0.24545454 2.1000000
> [2,] 0.000    0 0.04545454 0.0000000
> [3,] 0.125    0 0.09090909 0.0000000
> [4,] 0.125    0 0.09090909 0.3333333
>
> $A[[11]]
>           [,1] [,2] [,3] [,4]
> [1,] 0.0000000    0  1.1 1.54
> [2,] 0.1111111    0  0.0 0.00
> [3,] 0.0000000    0  0.0 0.00
> [4,] 0.1111111    0  0.0 0.00
>
> $A[[12]]
>            [,1] [,2] [,3] [,4]
> [1,] 0.00000000  0.0    0  1.5
> [2,] 0.00000000  0.0    0  0.0
> [3,] 0.09090909  0.0    0  0.0
> [4,] 0.54545455  0.5    0  0.5
>
> $A[[13]]
>            [,1] [,2]     [,3]       [,4]
> [1,] 0.00000000    0 1.785366 1.85652174
> [2,] 0.12857143    0 0.000000 0.01086956
> [3,] 0.02857143    0 0.000000 0.00000000
> [4,] 0.01428571    0 0.000000 0.02173913
>
> $A[[14]]
>            [,1]       [,2]  [,3]   [,4]
> [1,] 0.00000000 0.00000000 14.25 16.625
> [2,] 0.13144330 0.05714286  0.00  0.250
> [3,] 0.14432990 0.00000000  0.00  0.000
> [4,] 0.09278351 0.20000000  0.00  0.250
>
> $A[[15]]
>            [,1]       [,2]       [,3]       [,4]
> [1,] 0.00000000 0.00000000 0.59464286 1.76590909
> [2,] 0.00000000 0.00000000 0.01785714 0.00000000
> [3,] 0.02105263 0.01851852 0.03571429 0.04545454
> [4,] 0.02105263 0.01851852 0.03571429 0.06818182
>
> $A[[16]]
>      [,1]      [,2]       [,3]      [,4]
> [1,] 0.00 0.0000000 11.5000000 2.7758621
> [2,] 0.60 0.2857143  0.3333333 0.2413793
```

```
> [3,] 0.04 0.1428571  0.0000000 0.0000000
> [4,] 0.16 0.2857143  0.0000000 0.1724138
>
> $A[[17]]
>            [,1]       [,2] [,3]       [,4]
> [1,] 0.00000000 0.00000000 3.78 1.22500000
> [2,] 0.28358209 0.17105263 0.00 0.16666667
> [3,] 0.08457711 0.02631579 0.00 0.05555556
> [4,] 0.13930348 0.44736842 0.00 0.30555556
>
> $A[[18]]
>            [,1]      [,2]       [,3]       [,4]
> [1,] 0.00000000 0.0000000 1.54285714 1.03561644
> [2,] 0.12698413 0.1052632 0.04761905 0.05479452
> [3,] 0.09523809 0.1578947 0.19047619 0.08219178
> [4,] 0.11111111 0.2236842 0.00000000 0.35616438
>
> $A[[19]]
>      [,1] [,2] [,3]  [,4]
> [1,]    0    0 0.15 0.175
> [2,]    0    0 0.00 0.000
> [3,]    0    0 0.00 0.000
> [4,]    1    0 0.00 0.000
>
> $A[[20]]
>      [,1]      [,2] [,3] [,4]
> [1,]    0 0.0000000    0 0.25
> [2,]    0 0.0000000    0 0.00
> [3,]    0 0.0000000    0 0.00
> [4,]    1 0.6666667    0 1.00
>
> $A[[21]]
>      [,1] [,2] [,3]      [,4]
> [1,] 0.00    0    0 1.4285714
> [2,] 0.00    0    0 0.1428571
> [3,] 0.25    0    0 0.0000000
> [4,] 0.25    0    0 0.5714286
>
> $A[[22]]
>      [,1]      [,2] [,3]   [,4]
> [1,] 0.00 0.0000000  0.7 0.6125
> [2,] 0.25 0.0000000  0.0 0.1250
> [3,] 0.00 0.0000000  0.0 0.0000
> [4,] 0.25 0.1666667  0.0 0.2500
>
> $A[[23]]
>           [,1]      [,2] [,3] [,4]
> [1,] 0.0000000 0.0000000    0  0.6
> [2,] 0.2857143 0.0000000    0  0.0
> [3,] 0.2857143 0.3333333    0  0.0
> [4,] 0.2857143 0.3333333    0  1.0
```

```
>
> $A[[24]]
>           [,1] [,2]      [,3]   [,4]
> [1,] 0.0000000    0 0.7000000 0.6125
> [2,] 0.0000000    0 0.0000000 0.0000
> [3,] 0.0000000    0 0.0000000 0.0000
> [4,] 0.3333333    0 0.3333333 0.6250
>
> $A[[25]]
>           [,1] [,2] [,3]      [,4]
> [1,] 0.0000000    0  2.1 0.8166667
> [2,] 0.1666667    0  0.0 0.0000000
> [3,] 0.0000000    0  0.0 0.0000000
> [4,] 0.0000000    0  0.0 0.1666667
>
> $A[[26]]
>           [,1] [,2] [,3] [,4]
> [1,] 0.0000000  0.0    0    7
> [2,] 0.3333333  0.5    0    0
> [3,] 0.0000000  0.0    0    0
> [4,] 0.3333333  0.0    0    1
>
> $A[[27]]
>           [,1] [,2] [,3] [,4]
> [1,] 0.0000000 0.00    0  1.4
> [2,] 0.0000000 0.00    0  0.0
> [3,] 0.0000000 0.00    0  0.2
> [4,] 0.1111111 0.75    0  0.2
>
>
> $U
> $U[[1]]
>            [,1]       [,2]       [,3]       [,4]
> [1,] 0.00000000 0.00000000 0.00000000 0.00000000
> [2,] 0.20833333 0.00000000 0.00000000 0.05714286
> [3,] 0.04166667 0.07692308 0.00000000 0.00000000
> [4,] 0.08333333 0.07692308 0.06666667 0.02857143
>
> $U[[2]]
>           [,1]      [,2] [,3] [,4]
> [1,] 0.0000000 0.0000000  0.0  0.0
> [2,] 0.3218391 0.1428571  0.0  0.0
> [3,] 0.1609195 0.2857143  0.0  0.0
> [4,] 0.2528736 0.2857143  0.5  0.6
>
> $U[[3]]
>      [,1]       [,2]   [,3]       [,4]
> [1,]  0.0 0.00000000 0.0000 0.00000000
> [2,]  0.0 0.00000000 0.0000 0.03571429
> [3,]  0.1 0.06896552 0.0625 0.10714286
> [4,]  0.3 0.13793103 0.0000 0.07142857
```

```
>
> $U[[4]]
>           [,1] [,2]      [,3]       [,4]
> [1,] 0.0000000  0.0 0.0000000 0.00000000
> [2,] 0.1964286  0.0 0.0000000 0.00000000
> [3,] 0.1250000  0.5 0.0000000 0.00000000
> [4,] 0.1607143  0.5 0.1333333 0.07692308
>
> $U[[5]]
>            [,1]       [,2]  [,3]       [,4]
> [1,] 0.00000000 0.00000000 0.000 0.00000000
> [2,] 0.06557377 0.09090909 0.125 0.00000000
> [3,] 0.03278689 0.00000000 0.125 0.07692308
> [4,] 0.04918033 0.00000000 0.125 0.23076923
>
> $U[[6]]
>            [,1] [,2] [,3] [,4]
> [1,] 0.00000000  0.0    0  0.0
> [2,] 0.08333333  0.0    0  0.0
> [3,] 0.00000000  0.0    0  0.0
> [4,] 0.41666667  0.1    0  0.1
>
> $U[[7]]
>            [,1] [,2] [,3]       [,4]
> [1,] 0.00000000    0 0.00 0.00000000
> [2,] 0.07594937    0 0.05 0.08333333
> [3,] 0.13924051    0 0.00 0.25000000
> [4,] 0.07594937    0 0.00 0.08333333
>
> $U[[8]]
>           [,1]      [,2]       [,3]      [,4]
> [1,] 0.0000000 0.0000000 0.00000000 0.0000000
> [2,] 0.2238806 0.0000000 0.11111111 0.1428571
> [3,] 0.1343284 0.2727273 0.16666667 0.1428571
> [4,] 0.1194030 0.3636364 0.05555556 0.1428571
>
> $U[[9]]
>            [,1]  [,2]       [,3]      [,4]
> [1,] 0.00000000 0.000 0.00000000 0.0000000
> [2,] 0.07317073 0.025 0.03333333 0.0000000
> [3,] 0.03658537 0.150 0.10000000 0.1363636
> [4,] 0.06097561 0.225 0.16666667 0.2727273
>
> $U[[10]]
>       [,1] [,2]       [,3]      [,4]
> [1,] 0.000    0 0.00000000 0.0000000
> [2,] 0.000    0 0.04545454 0.0000000
> [3,] 0.125    0 0.09090909 0.0000000
> [4,] 0.125    0 0.09090909 0.3333333
>
> $U[[11]]
```

```
>           [,1] [,2] [,3] [,4]
> [1,] 0.0000000    0    0    0
> [2,] 0.1111111    0    0    0
> [3,] 0.0000000    0    0    0
> [4,] 0.1111111    0    0    0
>
> $U[[12]]
>            [,1] [,2] [,3] [,4]
> [1,] 0.00000000  0.0    0  0.0
> [2,] 0.00000000  0.0    0  0.0
> [3,] 0.09090909  0.0    0  0.0
> [4,] 0.54545455  0.5    0  0.5
>
> $U[[13]]
>            [,1] [,2] [,3]       [,4]
> [1,] 0.00000000    0    0 0.00000000
> [2,] 0.12857143    0    0 0.01086956
> [3,] 0.02857143    0    0 0.00000000
> [4,] 0.01428571    0    0 0.02173913
>
> $U[[14]]
>            [,1]       [,2] [,3] [,4]
> [1,] 0.00000000 0.00000000    0 0.00
> [2,] 0.13144330 0.05714286    0 0.25
> [3,] 0.14432990 0.00000000    0 0.00
> [4,] 0.09278351 0.20000000    0 0.25
>
> $U[[15]]
>            [,1]       [,2]       [,3]       [,4]
> [1,] 0.00000000 0.00000000 0.00000000 0.00000000
> [2,] 0.00000000 0.00000000 0.01785714 0.00000000
> [3,] 0.02105263 0.01851852 0.03571429 0.04545454
> [4,] 0.02105263 0.01851852 0.03571429 0.06818182
>
> $U[[16]]
>      [,1]      [,2]      [,3]      [,4]
> [1,] 0.00 0.0000000 0.0000000 0.0000000
> [2,] 0.60 0.2857143 0.3333333 0.2413793
> [3,] 0.04 0.1428571 0.0000000 0.0000000
> [4,] 0.16 0.2857143 0.0000000 0.1724138
>
> $U[[17]]
>            [,1]       [,2] [,3]       [,4]
> [1,] 0.00000000 0.00000000    0 0.00000000
> [2,] 0.28358209 0.17105263    0 0.16666667
> [3,] 0.08457711 0.02631579    0 0.05555556
> [4,] 0.13930348 0.44736842    0 0.30555556
>
> $U[[18]]
>            [,1]      [,2]       [,3]       [,4]
> [1,] 0.00000000 0.0000000 0.00000000 0.00000000
```

```
> [2,] 0.12698413 0.1052632 0.04761905 0.05479452
> [3,] 0.09523809 0.1578947 0.19047619 0.08219178
> [4,] 0.11111111 0.2236842 0.00000000 0.35616438
>
> $U[[19]]
>      [,1] [,2] [,3] [,4]
> [1,]    0    0    0    0
> [2,]    0    0    0    0
> [3,]    0    0    0    0
> [4,]    1    0    0    0
>
> $U[[20]]
>      [,1]      [,2] [,3] [,4]
> [1,]    0 0.0000000    0    0
> [2,]    0 0.0000000    0    0
> [3,]    0 0.0000000    0    0
> [4,]    1 0.6666667    0    1
>
> $U[[21]]
>      [,1] [,2] [,3]      [,4]
> [1,] 0.00    0    0 0.0000000
> [2,] 0.00    0    0 0.1428571
> [3,] 0.25    0    0 0.0000000
> [4,] 0.25    0    0 0.5714286
>
> $U[[22]]
>      [,1]      [,2] [,3]  [,4]
> [1,] 0.00 0.0000000    0 0.000
> [2,] 0.25 0.0000000    0 0.125
> [3,] 0.00 0.0000000    0 0.000
> [4,] 0.25 0.1666667    0 0.250
>
> $U[[23]]
>           [,1]      [,2] [,3] [,4]
> [1,] 0.0000000 0.0000000    0    0
> [2,] 0.2857143 0.0000000    0    0
> [3,] 0.2857143 0.3333333    0    0
> [4,] 0.2857143 0.3333333    0    1
>
> $U[[24]]
>           [,1] [,2]      [,3]  [,4]
> [1,] 0.0000000    0 0.0000000 0.000
> [2,] 0.0000000    0 0.0000000 0.000
> [3,] 0.0000000    0 0.0000000 0.000
> [4,] 0.3333333    0 0.3333333 0.625
>
> $U[[25]]
>           [,1] [,2] [,3]      [,4]
> [1,] 0.0000000    0    0 0.0000000
> [2,] 0.1666667    0    0 0.0000000
> [3,] 0.0000000    0    0 0.0000000
```

```
> [4,] 0.0000000    0    0 0.1666667
>
> $U[[26]]
>           [,1] [,2] [,3] [,4]
> [1,] 0.0000000  0.0    0    0
> [2,] 0.3333333  0.5    0    0
> [3,] 0.0000000  0.0    0    0
> [4,] 0.3333333  0.0    0    1
>
> $U[[27]]
>           [,1] [,2] [,3] [,4]
> [1,] 0.0000000 0.00    0  0.0
> [2,] 0.0000000 0.00    0  0.0
> [3,] 0.0000000 0.00    0  0.2
> [4,] 0.1111111 0.75    0  0.2
>
>
> $F
> $F[[1]]
>      [,1] [,2] [,3] [,4]
> [1,]    0    0 1.74 1.74
> [2,]    0    0 0.00 0.00
> [3,]    0    0 0.00 0.00
> [4,]    0    0 0.00 0.00
>
> $F[[2]]
>      [,1] [,2] [,3] [,4]
> [1,]    0    0  0.3  0.6
> [2,]    0    0  0.0  0.0
> [3,]    0    0  0.0  0.0
> [4,]    0    0  0.0  0.0
>
> $F[[3]]
>      [,1] [,2]    [,3]  [,4]
> [1,]    0    0 0.50625 0.675
> [2,]    0    0 0.00000 0.000
> [3,]    0    0 0.00000 0.000
> [4,]    0    0 0.00000 0.000
>
> $F[[4]]
>      [,1] [,2] [,3]     [,4]
> [1,]    0    0 2.44 6.569231
> [2,]    0    0 0.00 0.000000
> [3,]    0    0 0.00 0.000000
> [4,]    0    0 0.00 0.000000
>
> $F[[5]]
>      [,1] [,2] [,3]      [,4]
> [1,]    0    0 0.45 0.6461538
> [2,]    0    0 0.00 0.0000000
> [3,]    0    0 0.00 0.0000000
```

```
> [4,]    0    0 0.00 0.0000000
>
> $F[[6]]
>      [,1] [,2] [,3] [,4]
> [1,]    0    0 2.85 3.99
> [2,]    0    0 0.00 0.00
> [3,]    0    0 0.00 0.00
> [4,]    0    0 0.00 0.00
>
> $F[[7]]
>      [,1] [,2]  [,3]     [,4]
> [1,]    0    0 1.815 7.058333
> [2,]    0    0 0.000 0.000000
> [3,]    0    0 0.000 0.000000
> [4,]    0    0 0.000 0.000000
>
> $F[[8]]
>      [,1] [,2]     [,3] [,4]
> [1,]    0    0 1.233333  7.4
> [2,]    0    0 0.000000  0.0
> [3,]    0    0 0.000000  0.0
> [4,]    0    0 0.000000  0.0
>
> $F[[9]]
>      [,1] [,2] [,3]     [,4]
> [1,]    0    0 1.06 3.372727
> [2,]    0    0 0.00 0.000000
> [3,]    0    0 0.00 0.000000
> [4,]    0    0 0.00 0.000000
>
> $F[[10]]
>      [,1] [,2]      [,3] [,4]
> [1,]    0    0 0.2454545  2.1
> [2,]    0    0 0.0000000  0.0
> [3,]    0    0 0.0000000  0.0
> [4,]    0    0 0.0000000  0.0
>
> $F[[11]]
>      [,1] [,2] [,3] [,4]
> [1,]    0    0  1.1 1.54
> [2,]    0    0  0.0 0.00
> [3,]    0    0  0.0 0.00
> [4,]    0    0  0.0 0.00
>
> $F[[12]]
>      [,1] [,2] [,3] [,4]
> [1,]    0    0    0  1.5
> [2,]    0    0    0  0.0
> [3,]    0    0    0  0.0
> [4,]    0    0    0  0.0
>
```

```
> $F[[13]]
>      [,1] [,2]     [,3]     [,4]
> [1,]    0    0 1.785366 1.856522
> [2,]    0    0 0.000000 0.000000
> [3,]    0    0 0.000000 0.000000
> [4,]    0    0 0.000000 0.000000
>
> $F[[14]]
>      [,1] [,2]  [,3]   [,4]
> [1,]    0    0 14.25 16.625
> [2,]    0    0  0.00  0.000
> [3,]    0    0  0.00  0.000
> [4,]    0    0  0.00  0.000
>
> $F[[15]]
>      [,1] [,2]      [,3]     [,4]
> [1,]    0    0 0.5946429 1.765909
> [2,]    0    0 0.0000000 0.000000
> [3,]    0    0 0.0000000 0.000000
> [4,]    0    0 0.0000000 0.000000
>
> $F[[16]]
>      [,1] [,2] [,3]     [,4]
> [1,]    0    0 11.5 2.775862
> [2,]    0    0  0.0 0.000000
> [3,]    0    0  0.0 0.000000
> [4,]    0    0  0.0 0.000000
>
> $F[[17]]
>      [,1] [,2] [,3]  [,4]
> [1,]    0    0 3.78 1.225
> [2,]    0    0 0.00 0.000
> [3,]    0    0 0.00 0.000
> [4,]    0    0 0.00 0.000
>
> $F[[18]]
>      [,1] [,2]     [,3]     [,4]
> [1,]    0    0 1.542857 1.035616
> [2,]    0    0 0.000000 0.000000
> [3,]    0    0 0.000000 0.000000
> [4,]    0    0 0.000000 0.000000
>
> $F[[19]]
>      [,1] [,2] [,3]  [,4]
> [1,]    0    0 0.15 0.175
> [2,]    0    0 0.00 0.000
> [3,]    0    0 0.00 0.000
> [4,]    0    0 0.00 0.000
>
> $F[[20]]
>      [,1] [,2] [,3] [,4]
```

```
> [1,]    0    0    0 0.25
> [2,]    0    0    0 0.00
> [3,]    0    0    0 0.00
> [4,]    0    0    0 0.00
>
> $F[[21]]
>      [,1] [,2] [,3]     [,4]
> [1,]    0    0    0 1.428571
> [2,]    0    0    0 0.000000
> [3,]    0    0    0 0.000000
> [4,]    0    0    0 0.000000
>
> $F[[22]]
>      [,1] [,2] [,3]   [,4]
> [1,]    0    0  0.7 0.6125
> [2,]    0    0  0.0 0.0000
> [3,]    0    0  0.0 0.0000
> [4,]    0    0  0.0 0.0000
>
> $F[[23]]
>      [,1] [,2] [,3] [,4]
> [1,]    0    0    0  0.6
> [2,]    0    0    0  0.0
> [3,]    0    0    0  0.0
> [4,]    0    0    0  0.0
>
> $F[[24]]
>      [,1] [,2] [,3]   [,4]
> [1,]    0    0  0.7 0.6125
> [2,]    0    0  0.0 0.0000
> [3,]    0    0  0.0 0.0000
> [4,]    0    0  0.0 0.0000
>
> $F[[25]]
>      [,1] [,2] [,3]      [,4]
> [1,]    0    0  2.1 0.8166667
> [2,]    0    0  0.0 0.0000000
> [3,]    0    0  0.0 0.0000000
> [4,]    0    0  0.0 0.0000000
>
> $F[[26]]
>      [,1] [,2] [,3] [,4]
> [1,]    0    0    0    7
> [2,]    0    0    0    0
> [3,]    0    0    0    0
> [4,]    0    0    0    0
>
> $F[[27]]
>      [,1] [,2] [,3] [,4]
> [1,]    0    0    0  1.4
> [2,]    0    0    0  0.0
```

```
> [3,]    0    0    0  0.0
> [4,]    0    0    0  0.0
>
>
> $hstages
> [1] NA
>
> $agestages
> [1] NA
>
> $ahstages
>   stage_id stage_id stage original_size original_size_b original_size_c min_age
> 1        1        1   Sdl             1              NA              NA       0
> 2        2        2   Veg             1              NA              NA       0
> 3        3        3 SmFlo             2              NA              NA       0
> 4        4        4  LFlo             3              NA              NA       0
>   max_age repstatus obsstatus propstatus immstatus matstatus entrystage
> 1      NA         0         1          0         1         0          1
> 2      NA         0         1          0         0         1          0
> 3      NA         1         1          0         0         1          0
> 4      NA         1         1          0         0         1          0
>   indataset binhalfwidth_raw sizebin_min sizebin_max sizebin_center
> 1         1              0.5         0.5         1.5              1
> 2         1              0.5         0.5         1.5              1
> 3         1              0.5         1.5         2.5              2
> 4         1              0.5         2.5         3.5              3
>   sizebin_width binhalfwidthb_raw sizebinb_min sizebinb_max sizebinb_center
> 1             1                NA           NA           NA              NA
> 2             1                NA           NA           NA              NA
> 3             1                NA           NA           NA              NA
> 4             1                NA           NA           NA              NA
>   sizebinb_width binhalfwidthc_raw sizebinc_min sizebinc_max sizebinc_center
> 1             NA                NA           NA           NA              NA
> 2             NA                NA           NA           NA              NA
> 3             NA                NA           NA           NA              NA
> 4             NA                NA           NA           NA              NA
>   sizebinc_width group         comments alive almostborn
> 1             NA     0         Seedling     1          0
> 2             NA     0 Vegetative adult     1          0
> 3             NA     0  Small flowering     1          0
> 4             NA     0  Large flowering     1          0
>
> $labels
>    pop patch year2
> 1    1     C  2003
> 2    1     C  2004
> 3    1     C  2005
> 4    1     E  2003
> 5    1     E  2004
> 6    1     E  2005
> 7    1     F  2003
```

```
> 8    1     F  2004
> 9    1     F  2005
> 10   1     G  2003
> 11   1     G  2004
> 12   1     G  2005
> 13   1     L  2003
> 14   1     L  2004
> 15   1     L  2005
> 16   1     O  2003
> 17   1     O  2004
> 18   1     O  2005
> 19   1     Q  2003
> 20   1     Q  2004
> 21   1     Q  2005
> 22   1     R  2003
> 23   1     R  2004
> 24   1     R  2005
> 25   1     S  2003
> 26   1     S  2004
> 27   1     S  2005
>
> $matrixqc
> [1] 167  48  27
>
> attr(,"class")
> [1] "lefkoMat"
```

The resulting object has all of the elements of a standard `lefkoMat` object except for those elements related to quality control in the demographic dataset and linear modeling. The option `UFdecomp` was left at its default (`UFdecomp = TRUE`), and so `create_lM()` used the stageframe to infer where fecundity values were located in the matrices and created `U` and `F` matrices separating those values. This separation was performed on the basis of the stageframe, which shows which two stages are reproductive and which one stage acts as the entry stage into the population. The default option for `historical` is set to `FALSE`, yielding an `NA` in place of the `hstages` element, which would typically list the order of historical stage pairs.

Let's now take a look at a summary of this `lefkoMat` object.

```
summary(anth_lefkoMat)
>
> This ahistorical lefkoMat object contains 27 matrices.
>
> Each matrix is square with 4 rows and columns, and a total of 16 elements.
> A total of 167 survival transitions were estimated, with 6.185 per matrix.
> A total of 48 fecundity transitions were estimated, with 1.778 per matrix.
> This lefkoMat object covers 1 population, 9 patches, and 3 time steps.
>
> Survival probability sum check (each matrix represented by column in order):
>           [,1]  [,2]   [,3]   [,4]   [,5]  [,6]   [,7]  [,8]  [,9] [,10]  [,11]
> Min.    0.0667 0.500 0.0625 0.0769 0.0909 0.000 0.0000 0.333 0.171 0.000 0.0000
> 1st Qu. 0.0810 0.575 0.1708 0.1192 0.1334 0.075 0.0375 0.405 0.268 0.170 0.0000
> Median  0.1198 0.657 0.2106 0.3077 0.2276 0.100 0.1706 0.453 0.350 0.239 0.0000
> Mean    0.1599 0.637 0.2209 0.4231 0.2303 0.175 0.1895 0.469 0.320 0.203 0.0556
```

```
> 3rd Qu. 0.1987 0.720 0.2607 0.6116 0.3245 0.200 0.3225 0.517 0.402 0.271 0.0556
> Max.    0.3333 0.736 0.4000 1.0000 0.3750 0.500 0.4167 0.636 0.409 0.333 0.2222
>         [,12]  [,13] [,14]  [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
> Min.    0.000 0.0000 0.000 0.0370 0.333 0.000 0.238  0.00 0.000 0.000 0.000
> 1st Qu. 0.375 0.0000 0.193 0.0408 0.394 0.381 0.310  0.00 0.500 0.000 0.125
> Median  0.500 0.0163 0.313 0.0657 0.564 0.518 0.410  0.00 0.833 0.250 0.271
> Mean    0.409 0.0510 0.281 0.0705 0.565 0.420 0.388  0.25 0.667 0.304 0.260
> 3rd Qu. 0.534 0.0673 0.401 0.0954 0.736 0.557 0.488  0.25 1.000 0.554 0.406
> Max.    0.636 0.1714 0.500 0.1136 0.800 0.645 0.493  1.00 1.000 0.714 0.500
>         [,23] [,24]  [,25] [,26]  [,27]
> Min.    0.000 0.000 0.0000 0.000 0.0000
> 1st Qu. 0.500 0.250 0.0000 0.375 0.0833
> Median  0.762 0.333 0.0833 0.583 0.2556
> Mean    0.631 0.323 0.0833 0.542 0.3153
> 3rd Qu. 0.893 0.406 0.1667 0.750 0.4875
> Max.    1.000 0.625 0.1667 1.000 0.7500
```

The summary of this new `lefkoMat` object shows us that we have 27 matrices with four rows and columns each. The estimated numbers of transitions corresponds to the non-zero entries in each matrix. We see that we are covering one population, nine patches, and three time steps. All of the survival probabilities observed fall within the bounds of 0 to 1, so everything seems alright. At this point, we can use this object in `lefko3`'s various projection analyses.

12.2 Subsetting a lefkoMat object

Situations may arise in which a user will wish to create a new `lefkoMat` objects that is itself a subset of another. Such situations can arise, for example, when a `lefkoMat` object covers several patches or populations and the user wishes to create a single object to hold a single patch or population. In these situations, we can use the `subset_lM()` function.

Let's see an example of this function at work. Here, we will create two subset objects. The first will be a `lefkoMat` object with only patch S, and the second will be one with only matrices from 2004.

```
anth_S <- subset_lM(anth_lefkoMat, patch = "S")
anth_2004 <- subset_lM(anth_lefkoMat, year = 2004)

summary(anth_S)
>
> This ahistorical lefkoMat object contains 3 matrices.
>
> Each matrix is square with 4 rows and columns, and a total of 16 elements.
> A total of 10 survival transitions were estimated, with 3.333 per matrix.
> A total of 4 fecundity transitions were estimated, with 1.333 per matrix.
> This lefkoMat object covers 1 population, 1 patch, and 3 time steps.
>
> Survival probability sum check (each matrix represented by column in order):
>           [,1]  [,2]   [,3]
> Min.    0.0000 0.000 0.0000
> 1st Qu. 0.0000 0.375 0.0833
> Median  0.0833 0.583 0.2556
> Mean    0.0833 0.542 0.3153
> 3rd Qu. 0.1667 0.750 0.4875
```

```
> Max.    0.1667 1.000 0.7500
summary(anth_2004)
>
> This ahistorical lefkoMat object contains 9 matrices.
>
> Each matrix is square with 4 rows and columns, and a total of 16 elements.
> A total of 59 survival transitions were estimated, with 6.556 per matrix.
> A total of 15 fecundity transitions were estimated, with 1.667 per matrix.
> This lefkoMat object covers 1 population, 9 patches, and 1 time step.
>
> Survival probability sum check (each matrix represented by column in order):
>          [,1]   [,2]  [,3]   [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
> Min.    0.500 0.0909 0.333 0.0000 0.000 0.000 0.000 0.000 0.000
> 1st Qu. 0.575 0.1334 0.405 0.0000 0.193 0.381 0.500 0.500 0.375
> Median  0.657 0.2276 0.453 0.0000 0.313 0.518 0.833 0.762 0.583
> Mean    0.637 0.2303 0.469 0.0556 0.281 0.420 0.667 0.631 0.542
> 3rd Qu. 0.720 0.3245 0.517 0.0556 0.401 0.557 1.000 0.893 0.750
> Max.    0.736 0.3750 0.636 0.2222 0.500 0.645 1.000 1.000 1.000
```

The first object has three matrices, corresponding only to those of patch S. The second object has nine matrices, corresponding to all of the matrices in which time t is 2004. We can make even more complicated subsets by using the matrix number from the `A` list in the original `lefkoMat` object, as below.

```
anth_complex <- subset_lM(anth_lefkoMat, mat_num = c(1, 5, 20, 21))
summary(anth_complex)
>
> This ahistorical lefkoMat object contains 4 matrices.
>
> Each matrix is square with 4 rows and columns, and a total of 16 elements.
> A total of 24 survival transitions were estimated, with 6 per matrix.
> A total of 6 fecundity transitions were estimated, with 1.5 per matrix.
> This lefkoMat object covers 1 population, 3 patches, and 3 time steps.
>
> Survival probability sum check (each matrix represented by column in order):
>           [,1]   [,2]  [,3]  [,4]
> Min.    0.0667 0.0909 0.000 0.000
> 1st Qu. 0.0810 0.1334 0.500 0.000
> Median  0.1198 0.2276 0.833 0.250
> Mean    0.1599 0.2303 0.667 0.304
> 3rd Qu. 0.1987 0.3245 1.000 0.554
> Max.    0.3333 0.3750 1.000 0.714
```

Here we made a new `lefkoMat` object composed of four matrices, and chose those directly with the `mat_num` option.

12.3 Modifying a lefkoMat object

All `lefkoMat` objects may be edited. We may both add matrices, and delete them. Let's take a look at deleting matrices first.

The `delete_lM()` function is the primary means of removing matrices from a `lefkoMat` object. It takes the same inputs as `subset_lM()` and uses them in similar ways, but to remove matrices rather

than subset them. It removes the matrix and edits the associated `labels` and quality control objects to reflect the change. Let's try one such situation, in which we create a new `lefkoMat` object by creating a copy of `anth_lefkoMat` in which we have removed patch S.

```
anth_noS <- delete_lM(anth_lefkoMat, patch = "S")
summary(anth_noS)
>
> This ahistorical lefkoMat object contains 24 matrices.
>
> Each matrix is square with 4 rows and columns, and a total of 16 elements.
> A total of 157 survival transitions were estimated, with 6.542 per matrix.
> A total of 44 fecundity transitions were estimated, with 1.833 per matrix.
> This lefkoMat object covers 1 population, 8 patches, and 3 time steps.
>
> Survival probability sum check (each matrix represented by column in order):
>           [,1]  [,2]   [,3]   [,4]   [,5]  [,6]   [,7]  [,8]  [,9] [,10]  [,11]
> Min.    0.0667 0.500 0.0625 0.0769 0.0909 0.000 0.0000 0.333 0.171 0.000 0.0000
> 1st Qu. 0.0810 0.575 0.1708 0.1192 0.1334 0.075 0.0375 0.405 0.268 0.170 0.0000
> Median  0.1198 0.657 0.2106 0.3077 0.2276 0.100 0.1706 0.453 0.350 0.239 0.0000
> Mean    0.1599 0.637 0.2209 0.4231 0.2303 0.175 0.1895 0.469 0.320 0.203 0.0556
> 3rd Qu. 0.1987 0.720 0.2607 0.6116 0.3245 0.200 0.3225 0.517 0.402 0.271 0.0556
> Max.    0.3333 0.736 0.4000 1.0000 0.3750 0.500 0.4167 0.636 0.409 0.333 0.2222
>         [,12]  [,13] [,14]  [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
> Min.    0.000 0.0000 0.000 0.0370 0.333 0.000 0.238  0.00 0.000 0.000 0.000
> 1st Qu. 0.375 0.0000 0.193 0.0408 0.394 0.381 0.310  0.00 0.500 0.000 0.125
> Median  0.500 0.0163 0.313 0.0657 0.564 0.518 0.410  0.00 0.833 0.250 0.271
> Mean    0.409 0.0510 0.281 0.0705 0.565 0.420 0.388  0.25 0.667 0.304 0.260
> 3rd Qu. 0.534 0.0673 0.401 0.0954 0.736 0.557 0.488  0.25 1.000 0.554 0.406
> Max.    0.636 0.1714 0.500 0.1136 0.800 0.645 0.493  1.00 1.000 0.714 0.500
>         [,23] [,24]
> Min.    0.000 0.000
> 1st Qu. 0.500 0.250
> Median  0.762 0.333
> Mean    0.631 0.323
> 3rd Qu. 0.893 0.406
> Max.    1.000 0.625
```

We can see that we have a new object with three fewer matrices, and one fewer patch. Let's see the `labels` object to make sure.

```
anth_noS$labels
>    pop patch year2
> 1    1     C  2003
> 2    1     C  2004
> 3    1     C  2005
> 4    1     E  2003
> 5    1     E  2004
> 6    1     E  2005
> 7    1     F  2003
> 8    1     F  2004
> 9    1     F  2005
> 10   1     G  2003
```

```
> 11    1     G  2004
> 12    1     G  2005
> 13    1     L  2003
> 14    1     L  2004
> 15    1     L  2005
> 16    1     O  2003
> 17    1     O  2004
> 18    1     O  2005
> 19    1     Q  2003
> 20    1     Q  2004
> 21    1     Q  2005
> 22    1     R  2003
> 23    1     R  2004
> 24    1     R  2005
```

The last patch is clearly patch R. So, patch S is gone from this object.

We may also add matrices using the `add_lM()` function. In this case, we usually need to add a little information to make sure that the `labels` object is updated properly. Let's see how this works by adding the S patches back to our new `lefkoMat` object.

```
mats_list_S <- list(XS3, XS4, XS5)

anth_noS_addedS <- add_lM(anth_noS, Amats = mats_list_S, UFdecomp = TRUE,
  patch = c("S", "S", "S"), year = c(2003, 2004, 2005))

summary(anth_noS_addedS)
>
> This ahistorical lefkoMat object contains 27 matrices.
>
> Each matrix is square with 4 rows and columns, and a total of 16 elements.
> A total of 167 survival transitions were estimated, with 6.185 per matrix.
> A total of 48 fecundity transitions were estimated, with 1.778 per matrix.
> This lefkoMat object covers 1 population, 9 patches, and 3 time steps.
>
> Survival probability sum check (each matrix represented by column in order):
>           [,1]  [,2]   [,3]   [,4]   [,5]  [,6]   [,7]  [,8]  [,9] [,10]  [,11]
> Min.    0.0667 0.500 0.0625 0.0769 0.0909 0.000 0.0000 0.333 0.171 0.000 0.0000
> 1st Qu. 0.0810 0.575 0.1708 0.1192 0.1334 0.075 0.0375 0.405 0.268 0.170 0.0000
> Median  0.1198 0.657 0.2106 0.3077 0.2276 0.100 0.1706 0.453 0.350 0.239 0.0000
> Mean    0.1599 0.637 0.2209 0.4231 0.2303 0.175 0.1895 0.469 0.320 0.203 0.0556
> 3rd Qu. 0.1987 0.720 0.2607 0.6116 0.3245 0.200 0.3225 0.517 0.402 0.271 0.0556
> Max.    0.3333 0.736 0.4000 1.0000 0.3750 0.500 0.4167 0.636 0.409 0.333 0.2222
>          [,12]  [,13] [,14]  [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
> Min.    0.000 0.0000 0.000 0.0370 0.333 0.000 0.238  0.00 0.000 0.000 0.000
> 1st Qu. 0.375 0.0000 0.193 0.0408 0.394 0.381 0.310  0.00 0.500 0.000 0.125
> Median  0.500 0.0163 0.313 0.0657 0.564 0.518 0.410  0.00 0.833 0.250 0.271
> Mean    0.409 0.0510 0.281 0.0705 0.565 0.420 0.388  0.25 0.667 0.304 0.260
> 3rd Qu. 0.534 0.0673 0.401 0.0954 0.736 0.557 0.488  0.25 1.000 0.554 0.406
> Max.    0.636 0.1714 0.500 0.1136 0.800 0.645 0.493  1.00 1.000 0.714 0.500
>          [,23] [,24]  [,25] [,26]  [,27]
> Min.    0.000 0.000 0.0000 0.000 0.0000
> 1st Qu. 0.500 0.250 0.0000 0.375 0.0833
```

```
> Median  0.762 0.333 0.0833 0.583 0.2556
> Mean    0.631 0.323 0.0833 0.542 0.3153
> 3rd Qu. 0.893 0.406 0.1667 0.750 0.4875
> Max.    1.000 0.625 0.1667 1.000 0.7500
```

Users can compare this summary to that of `anth_lefkoMat` to see that they are the same.

In the options above, we set `UFdecomp = TRUE`. This is an important setting, because it allows R to use th stageframe to infer which elements are the fecundity elements. The function `create_lM()` also has this setting, but the default for that function is `TRUE`. In `add_lM()`, the default is set to `FALSE`, meaning that fecundity terms would not be set to the `F` matrices instead of the `U` matrices without the user explicitly setting it to `TRUE`. Conversely, if matrix elements holding fecundity rates also mix those rates with survival probabilities, then the user may wish to set this option to `FALSE` across the board and edit the `U` and `F` matrices by hand to set the matrices properly.

12.4 Points to remember

1. Matrices may be imported for use in `lefko3` via a variety of approaches, ranging from loading CSV files to direct text input.
2. Properly important matrices into `lefko3` requires the creation of a stageframe that characterizes each stage used. All characteristics other than size need to be properly documented.
3. Function `create_lM()` will also separate fecundity from survival using the stageframe as a guide, creating `U` and `F` matrices in addition to `A` matrices.
4. Functions may be subsetted with `subset_lM()`. Matrices may be added or deleted with `add_lM()` and `delete_lM()`, respectively.