

Chapter 3

Demographic data formatting

Matrix projection models are estimated with demographic data. Regardless of whether they are raw or function-based, the data needs to follow a particular format to make the estimation of elements or vital rate models possible. Package `lefko3` utilizes a standardized format similar to formats required by other protocols, such as the `projection.matrix()` function in package `popbio`. Unlike other packages, `lefko3` includes functions that can standardize a variety of starting dataset formats. These functions likely do not cover all possibilities, but deal with most of the interesting differences among the protocols that demographers use as experienced by the author’s experiences with several tens if not hundreds of collaborators all over the planet.

The format used in analysis is similar to what one would expect in setting up a data frame for a linear model in R or another statistical analysis environment. If we start with a record of observations of a single individual across several years, where each observation was conducted during a single monitoring session per year, then our data needs to be formatted in a way that we can assess at the very least whether the state of an individual at some time depends on the state at the previous time. For a historical model, this dependence extends one further monitoring time back, meaning that we have to be able to assess demographic patterns across groups of 3 consecutive monitoring times. In order to analyze these patterns of dependence, we break the data up into pairs of consecutive observation times (or groups of three consecutive times if analyzing historical models). These broken up pairs are then stacked together into a data frame. For example, if we have data for an individual in years 2010, 2011, 2012, and 2013, then we would break up these data into a data set for the years 2010 and 2011, followed by the years 2011 and 2012, followed by the years 2012 and 2013 (if historical, then we would break up 2010-2011-2012, followed by 2011-2012-2013). In terms of the language of matrix projection models, these standardized dataset are arranged with time t followed by time $t+1$ if ahistorical, and time $t-1$, followed by time t , followed by time $t+1$ in historical format. Thinking of this in terms of the language of linear modeling, the final time in each group can be seen as the response time, while the previous one or two times can be seen as the causal times. We refer to this stacked format as **vertical format**, because it stretches the data for a single individual vertically across rows. This format can be further divided into **historically-formatted vertical (hfv) format** (Shefferson *et al.* 2021), in which groups of three consecutive monitoring times are stacked, and **ahistorically-formatted vertical (ahfv) format**, in which pairs of consecutive years are stacked.

The actual states of individuals can be assessed through multiple variables per observation time. There may be one or more measures of size, an age metric, a measure of reproductive status, and perhaps fecundity, among other variables. Each monitoring occasion should have the same variables for use in analysis. Thus, putting states for consecutive pairs or groups of times in this way will likely result in data frames with many more than two or three columns. Stacking the data in this fashion also ends up leading to multiple rows of data originating from the same individual. Until recently, these rows were treated as independent data points, and the simplest analyses still assume independence. However, if a separate variable is created recording the identity of the individual, then non-independence of data points can be dealt with statistically via the use of individual identity as a random factor in function-based modeling (more on this in later chapters).

Kinds of individual variables that may be included

What sorts of variables can be handled in MPM analysis through `lefko3`? Because this package was originally developed to handle historical size-classified Lefkovitch models, it can of course handle size variables. From version 4.0.0 onward, up to three different size variables can be included in a single analysis. However, it can handle many, many more variables. At the very least, variables denoting *reproductive status* and *fecundity* can be included. These three groups of variables - size, reproductive status, and fecundity - are the classic three

variable types used for classification purposes in stage-based models. Wildlife demographers may be more interested in age than size, and *age* at each time can also be included. In fact, `lefko3` automates the estimation of age and related variables, such as *observed age* at time t and *observed lifespan*. Plant demographers may be particularly interested in variables denoting *observation status*, which is a binomial variable used to denote whether an individual is in an observable stage (`lefko3` can automate the identification of these occurrences within the dataset). They may also be interested in multiple allowed measures of reproductive status and fecundity, since many plants have multiple modes of reproduction, and actual fecundity cannot be accurately known and so must be estimated.

Identity variables may be included and used in analysis. Such variables include *population identity*, *patch identity* or *subpopulation identity*, *individual identity*, and *monitoring occasion*. Package `lefko3` can use these to subset the data accordingly, and also to estimate variables such as *observed age* and *observed lifespan*. If the user has already determined stages for all individuals at all times, then these data may be incorporated as well (although `lefko3` can automatically assign stages). Demographers commonly include *censoring variables* to denote data points that are suspect. These can also be included in `lefko3`, and those data points can also be easily removed from analysis. For example, individuals with damaged tags from which identity cannot be properly read can be marked differently in a censor variable, and analyses can then be performed with and without these potentially biasing data. *Spatial coordinates* can be included in datasets. In `lefko3`, spatial coordinates should be Cartesian, and so paired as X-Y coordinates. Package `lefko3` includes automated density estimation, and so these coordinates can be used to estimate *spatial density*.

Finally, variables can be included covering status as alive or dead, status as observed or unobserved, and up to three separate individual covariates that can be numeric or categorical. Individual covariates may be particularly useful as a means to include environmental variables, or further information on status, such as management regime or presence of injury.

Formatting the demographic data properly

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	plantid	patch	censor	year2	Inf2.2	Inf.2	Veg.2	Pod.2	Inf2.3	Inf.3	Veg.3	Pod.3			
2	164	A	1	2004	0	0	1	0	NA	NA	NA	NA			
3	165	A	1	2004	0	2	1	1	0	2	0	0			
4	240	A	1	2004	0	1	0	0	0	0	0	1			
5	242	A	1	2004	0	0	1	0	0	0	0	1			
6	243	A	1	2004	0	0	5	0	0	0	2	0			
7	246	A	1	2004	0	0	1	0	0	0	0	1			
8	248	A	1	2004	0	0	1	0	0	0	0	1			
9	249	A	1	2004	0	0	1	0	0	1	0	0			
10	251	A	1	2004	0	0	2	0	0	0	2	0			
11	252	A	1	2004	0	0	1	0	0	2	0	0			
12	253	A	1	2004	0	0	1	0	0	1	1	1			
13	255	A	1	2004	0	0	8	0	0	1	3	1			
14	257	A	1	2004	0	0	6	0	0	3	3	1			
15	259	A	1	2004	0	0	2	0	0	1	2	0			
16	263	A	1	2004	0	0	1	0	0	1	0	0			
17	264	A	1	2004	0	0	2	0	0	1	0	0			
18	265	A	1	2004	0	0	1	0	0	2	1	2			
19	391	A	1	2004	0	0	1	0	0	1	0	1			
20	393	A	1	2004	0	2	3	2	0	3	1	1			
21	394	A	1	2004	0	0	3	0	0	0	6	0			
22	431	B	1	2004	0	0	6	0	0	0	4	0			
23	434	B	1	2004	0	0	1	0	0	0	8	0			
24	436	B	1	2004	0	2	8	0	1	8	2	0			
25	437	B	1	2004	0	1	3	1	0	3	1	2			
26	439	B	1	2004	NA	NA	NA	NA	0	0	1	0			
27	440	B	1	2004	0	1	9	0	0	4	4	3			
28	441	B	1	2004	0	0	4	0	0	2	4	2			
29	442	B	1	2004	0	0	4	0	0	0	2	0			

Figure 3.1: Vertically-formatted dataset covering the *Cypripedium candidum* data

Regardless of how the dataset *should* be structured for analysis, demographers themselves often have their own ways of structuring data. In general, most demographers keep their data either in **vertical format** (explained above), or **horizontal format**, in which the data for each single individual spans only a single row that contains all data on state across all monitoring occasions. Above, in Figure 3.1, we see an example of an

ahistorically-formatted vertical dataset. Below, in Figure 3.2, we see an example of a horizontally-formatted dataset.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	plantid	patch	censo	Inf2.00	Inf.00	Veg.00	Pod.00	Inf2.01	Inf.01	Veg.01	Pod.01	Inf2.02	Inf.02	Veg.02	Pod.02	Inf2.03	Inf.03
2	163 A		1	NA	NA	NA	NA	0	0	1	0	NA	NA	NA	NA	NA	NA
3	164 A		1	NA	NA	NA	NA	NA	NA	NA	NA	0	0	0	3	0	0
4	165 A		1	0	3	1	0	0	1	0	0	0	0	3	1	0	0
5	240 A		1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0
6	242 A		1	0	0	3	0	0	1	0	0	0	0	2	0	0	0
7	243 A		1	NA	NA	NA	NA	NA	NA	NA	NA	0	0	3	1	0	NA
8	244 A		1	0	1	0	0	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
9	246 A		1	0	0	3	0	0	3	4	0	0	0	4	3	0	0
10	247 A		1	0	0	1	0	0	0	1	0	NA	NA	NA	NA	0	0
11	248 A		1	0	0	1	0	0	1	0	0	0	0	1	0	0	0
12	249 A		1	NA	NA	NA	NA	0	1	0	0	0	0	1	1	0	0
13	251 A		1	NA	NA	NA	NA	0	0	3	0	0	0	3	0	0	0
14	252 A		1	NA	NA	NA	NA	0	1	1	0	0	0	2	0	0	0
15	253 A		1	0	2	3	0	0	3	4	0	0	0	1	6	0	0
16	254 A		1	0	2	3	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
17	255 A		1	NA	NA	NA	NA	NA	NA	NA	NA	0	0	3	4	0	0
18	256 A		1	NA	NA	NA	NA	0	4	8	0	NA	NA	NA	NA	NA	NA
19	257 A		1	0	0	4	0	NA	NA	NA	NA	0	0	2	1	0	0
20	258 A		1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
21	259 A		1	0	0	2	0	0	1	1	0	0	0	2	0	0	0
22	260 A		1	NA	NA	NA	NA	NA	NA	NA	NA	0	0	2	2	0	NA
23	261 A		1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
24	263 A		1	NA	NA	NA	NA	0	0	1	0	0	0	1	0	0	0
25	264 A		1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0
26	265 A		1	NA	NA	NA	NA	0	1	0	0	NA	NA	NA	NA	0	0
27	391 A		1	NA	NA	NA	NA	NA	NA	NA	NA	0	0	0	1	0	0
28	393 A		1	0	2	1	1	0	2	0	0	0	0	4	0	0	0
29	394 A		1	NA	NA	NA	NA	NA	NA	NA	NA	0	0	0	5	0	0
30	431 B		1	0	1	2	0	0	0	2	0	0	0	3	1	0	0
31	432 B		1	0	0	2	0	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
32	433 B		1	0	2	1	0	NA	NA	NA	NA	0	4	2	0	NA	NA
33	434 B		1	NA	NA	NA	NA	0	2	4	0	0	0	3	5	0	0
34	435 B		1	0	3	3	0	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
35	436 B		1	0	5	0	0	0	5	2	0	0	0	8	1	0	0
36	437 B		1	0	1	1	1	0	2	0	0	0	0	3	0	0	0
37	439 B		1	0	1	0	0	0	0	1	0	0	0	0	1	0	0
38	440 B		1	0	7	2	0	0	3	6	0	0	0	3	6	0	0

Figure 3.2: Horizontally-formatted dataset covering the *Cyripedium candidum* data

Every demographer has their own way of keeping data, and I personally do not feel it is up to me to judge (the level of chaos in my own record-keeping prevents me from making ANY harsh judgments). We have therefore created two powerful functions that can handle most datasets falling into these two general types and standardize them into formats that can be analyzed by `lefko3`. These functions include `verticalize3()`, and `historicalize3()`. We will start by describing the former.

Function `verticalize3()`

Many demographers (including the author) prefer *horizontal* data formats for demographic record-keeping. Here, individual life histories are recorded in a spreadsheet with each unique individual's data organized within a single row. The columns then correspond to descriptor variables and to condition at different times. This format has the advantage that there it is never difficult to find an individual's resighting history, since it must occur all within a single row. In these circumstances, the `verticalize3()` function can standardize these data into *hfv* format.

The `verticalize3()` function takes a number of inputs. Naturally, it needs the data frame being used for analysis (the `data` field). It also needs to know the number of monitoring occasions that the dataset covers (field `noyears`), and which columns identify the population, patch/subpopulation, and individual (fields `popidcol`, `patchidcol`, and `individcol`, respectively).

After these most basic of fields have been entered, the columns conveying each kind of demographic variable need to be identified (this is a very flexible function, so only those variables that actually occur in the dataset need be identified and all others can be ignored). Up to 3 separate size variables can be identified (`sizeacol`, `sizebcol`, and `sizeccol`), as can up to two reproductive status variables (`repstracol` and `restrbcol`), and up to two fecundity variables (`fecacol` and `fecbcol`). These variables will naturally repeat across monitoring occasions, and so each variable represents as many actual columns as monitoring occasions covered by the dataset. To make sure that `R` finds all columns corresponding to a variable, the user may take one of two approaches. The most flexible is to treat each input field as a vector, and to input the

name or column number of each variable in order of monitoring occasion. For example, if we decided to use only one size variable, perhaps the height of a plant in centimeters, and this variable is labelled `ht.2010`, `ht.2011`, `ht.2012`, and `ht.2013` for the years 2010, 2011, 2012, and 2013, then we can input `sizeacol = c("ht.2010", "ht.2011", "ht.2012", "ht.2013")`. All other variables covering status across time need to be input in the same way, yielding equal length vectors in the same temporal order. Alternatively, if the data is arranged such that each monitoring occasion has its data arranged in a strictly repeating order, and each variable is a fixed number of column away from the equivalent variable in the preceding time and in the next time, then only the first of these columns needs to be designated, and the `blocksize` field can be set to the number of columns separating these entries. In the plant height example, if each size variable is 10 columns after the previous size variable and this pattern holds for all other variable types, then this would mean entering only `noyears = 4`, `blocksize = 10`, `sizeacol = "ht.2010"`. Note that the use of the `blocksize` option *absolutely requires* that each instance of a particular variable occurs exactly `blocksize` columns after the previous instance, with no exceptions allowed. Users who occasionally add empty columns here and there in their dataset will likely find mistakes in the resulting standardization if they do not make sure that the exact number of COLUMNS between instances is ALWAYS the same.

While standardizing the data, function `verticalize3()` can also handle stage assignment if given a proper stageframe (the `stageassign` field). To do so properly, `lefko3` also needs to know which size(s) to use for classification purposes. Currently, the options include: `sizea`, `sizeb`, and `sizec` if only one of these size metrics determines stage; `sizeab`, `sizeac`, and `sizebc` if stage assignment is based on a pair of size metrics; `sizeabc` if stage assignment should utilize three size metrics; and `sizeadded`, if stage assignment should proceed on the basis of the sum of all used size metrics. Stage assignment is automated using these size metrics as well as reproductive status, observation status, and status as propagule, immature, and mature. **Note that stage assignment will yield warnings and unassigned stages if any combination of variables is found in the demographic dataset that does not fit exactly one stage in the stage frame.** We also encourage the use of this function without stage assignment on the first pass, since this will allow users to explore the data more fully prior to the development of a life history model and stageframe.

Function `verticalize3()` includes a few more fields of interest. The `repstrrel` and `fecrel` fields provided means of equalizing reproductive status variables and fecundity variables, respectively, that are set on different scales. For example, if two reproductive status variables are used - the first covering single-flowered individuals and the second covering double-flowered individual - then setting `repstrrel = 2` will tell *R* that the second variable counts for twice the reproductive status of the first. The same applies to fecundity, if two different fecundity measures are used.

Demographers using censoring variables will find powerful options in this function. First, `sensorcol` can be set to a static censor variable covering an individual's entire resighting history, or to censor variables across time. This can be set with the `sensorRepeat` field, which defaults to `FALSE` and so assumes that censor values do not vary across time. The value of the censor variable used to denote which individuals to keep can be set to any numeric value (the `sensorkeep` option). The default is `sensorkeep = 0`, meaning that any data point marked 0 are considered OK, while any other value is assumed to be subject to removal. However, even NA can be used as the value to mark data to keep. Finally, if all options are set as described, then function `verticalize3()` will format censor data but will not remove any data points. To force `verticalize3()` to remove censored data points, set `sensor = TRUE`.

Function `verticalize3()` has great power in handling spatial coordinates, and can even estimate spatial density. Cartesian coordinates for each individual are assumed to be a single set. However, if coordinates can change across time and each time point has its own set of coordinates for each individual present, then users can set `coordsRepeat = TRUE`. If the intent in using these coordinates is to estimate spatial density, then this function will calculate the number of individuals within some radius of each focal individual in each monitoring time if given the length of the radius to use option `spacing`).

This function includes five further parameters handling miscellaneous cases. First, demographers often treat missing data as blanks within their spreadsheets. Setting `NAas0 = TRUE` tells *R* to interpret missing data as zeros. This can be useful in many circumstances. For example, if fecundity is estimated as the number of flowers per individual, but the number of flowers per individual is only recorded when flowers actually exist, then setting `NAas0 = TRUE` can allow 0 values to be incorporated in analyses of this fecundity variable (NA values are generally removed prior to analysis). A further scenario concerns the treatment of reproductive status within the dataset relative to the stageframe. If stages are defined such that mature stages

are reproductive but it is nonetheless possible for fecundity to equal 0, then setting `NRasRep = TRUE` tells *R* not to use reproductive status in stage classification. Otherwise, *R* will assume that a mature individual that produces offspring is in a different stage than a mature individual that does not, and this may lead to warnings, unassigned stages, and further difficulties if the stageframe does not include stages to cover the non-reproductive category. The `NAasObs` option provides a means of telling *R* that unobservable stages should be treated as observed during stage classification, if set to `TRUE` (the default is `FALSE`). The `reduce` option, if set to `TRUE`, will tell *R* that any columns in the output dataset with only a single constant value should be removed, as invariant variables yield no influence on response variables. Finally, the `a2check` field should generally be kept to `FALSE` (the default), but if set to `TRUE`, then the resulting standardized dataset will include data for individuals not alive in time *t* (these instances are removed automatically under the default, since it is only individuals alive in time *t* that can transition to anything in time *t*+1).

Now that we have explained the options, let's standardize data frames for the stageframes created in the last chapter. Let's first standardize the vertical dataset for the raw MPM. For this purpose, we will utilize the `verticalize3()` function. Because we are lumping reproductive and non-reproductive individuals into the non-dormant adult classes, we need to set `NRasRep = TRUE`. Otherwise, `verticalize3()` will attempt to use the reproductive status of individuals in classification, and will fail due to the presence of non-reproductive adults. We also need to set `NAas0 = TRUE` to make sure that NA values in size are turned into 0 entries where necessary, and so aid in the assignment of the vegetative dormancy stage. Finally, note that we set up three different size variables here, not as `sizea`, `sizeb`, and `sizec`, and that we tell *R* that we want overall size to be the sum of these (`stagesize = "sizeadded"`).

```
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
  patchidcol = "patch", individcol = "plantid", blocksize = 4,
  sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
  repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
  stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
  NRasRep = TRUE)

summary(cypraw_v1)
>      rowid      popid patchid  individ      year2
> Min.   : 1.00    :320  A: 93   Length:320   Min.   :2004
> 1st Qu.:21.00          B:154   Class :character 1st Qu.:2005
> Median :37.50          C: 73   Mode  :character  Median :2006
> Mean   :38.45                                     Mean   :2006
> 3rd Qu.:56.00                                     3rd Qu.:2007
> Max.   :77.00                                     Max.   :2008
>      firstseen      lastseen      obsage      obslifespan
> Min.   :2004   Min.   :2004   Min.   :0.000   Min.   :0.000
> 1st Qu.:2004   1st Qu.:2009   1st Qu.:1.000   1st Qu.:5.000
> Median :2004   Median :2009   Median :2.000   Median :5.000
> Mean   :2004   Mean   :2009   Mean   :1.853   Mean   :4.556
> 3rd Qu.:2004   3rd Qu.:2009   3rd Qu.:3.000   3rd Qu.:5.000
> Max.   :2008   Max.   :2009   Max.   :4.000   Max.   :5.000
>      sizea1      sizeb1      sizec1      sizeadded
> Min.   :0.000000   Min.   : 0.0000   Min.   : 0.0    Min.   : 0.000
> 1st Qu.:0.000000   1st Qu.: 0.0000   1st Qu.: 0.0    1st Qu.: 0.000
> Median :0.000000   Median : 0.0000   Median : 1.0    Median : 2.000
> Mean   :0.009375   Mean   : 0.7469   Mean   : 1.9    Mean   : 2.656
> 3rd Qu.:0.000000   3rd Qu.: 1.0000   3rd Qu.: 3.0    3rd Qu.: 4.000
> Max.   :1.000000   Max.   :18.0000   Max.   :13.0    Max.   :21.000
>      repstra1      repstrb1      fecal      juvgiven1
> Min.   : 0.0000   Min.   :0.000000   Min.   :0.0000   Min.   :0
> 1st Qu.: 0.0000   1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:0
> Median : 0.0000   Median :0.000000   Median :0.0000   Median :0
```

```

> Mean : 0.7469 Mean :0.009375 Mean :0.2656 Mean :0
> 3rd Qu.: 1.0000 3rd Qu.:0.000000 3rd Qu.:0.0000 3rd Qu.:0
> Max. :18.0000 Max. :1.000000 Max. :7.0000 Max. :0
> obsstatus1 repstatus1 fecstatus1 matstatus1
> Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000
> 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:1.0000
> Median :1.0000 Median :0.0000 Median :0.0000 Median :1.0000
> Mean :0.7469 Mean :0.2875 Mean :0.1344 Mean :0.7688
> 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1.0000
> Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000
> alive1 stage1 stage1index sizea2
> Min. :0.0000 Length:320 Min. : 0.000 Min. :0.000000
> 1st Qu.:1.0000 Class :character 1st Qu.: 6.000 1st Qu.:0.000000
> Median :1.0000 Mode :character Median : 8.000 Median :0.000000
> Mean :0.7688 Mean : 6.144 Mean :0.009375
> 3rd Qu.:1.0000 3rd Qu.: 8.000 3rd Qu.:0.000000
> Max. :1.0000 Max. :11.000 Max. :1.000000
> sizeb2 sizec2 size2added repstra2
> Min. : 0.0000 Min. : 0.000 Min. : 0.000 Min. : 0.0000
> 1st Qu.: 0.0000 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.: 0.0000
> Median : 0.0000 Median : 2.000 Median : 2.000 Median : 0.0000
> Mean : 0.8969 Mean : 2.416 Mean : 3.322 Mean : 0.8969
> 3rd Qu.: 1.0000 3rd Qu.: 3.000 3rd Qu.: 4.000 3rd Qu.: 1.0000
> Max. :18.0000 Max. :13.000 Max. :24.000 Max. :18.0000
> repstrb2 fecca2 juvgiven2 obsstatus2
> Min. :0.000000 Min. :0.0000 Min. :0 Min. :0.0000
> 1st Qu.:0.000000 1st Qu.:0.0000 1st Qu.:0 1st Qu.:1.0000
> Median :0.000000 Median :0.0000 Median :0 Median :1.0000
> Mean :0.009375 Mean :0.2906 Mean :0 Mean :0.9531
> 3rd Qu.:0.000000 3rd Qu.:0.0000 3rd Qu.:0 3rd Qu.:1.0000
> Max. :1.000000 Max. :7.0000 Max. :0 Max. :1.0000
> repstatus2 fecstatus2 matstatus2 alive2 stage2
> Min. :0.0000 Min. :0.0000 Min. :1 Min. :1 Length:320
> 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:1 1st Qu.:1 Class :character
> Median :0.0000 Median :0.0000 Median :1 Median :1 Mode :character
> Mean :0.3688 Mean :0.1562 Mean :1 Mean :1
> 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1 3rd Qu.:1
> Max. :1.0000 Max. :1.0000 Max. :1 Max. :1
> stage2index sizea3 sizeb3 sizec3
> Min. : 6.000 Min. :0.000000 Min. : 0.000 Min. : 0.000
> 1st Qu.: 7.000 1st Qu.:0.000000 1st Qu.: 0.000 1st Qu.: 1.000
> Median : 8.000 Median :0.000000 Median : 0.000 Median : 1.000
> Mean : 7.919 Mean :0.009375 Mean : 1.069 Mean : 2.209
> 3rd Qu.: 8.000 3rd Qu.:0.000000 3rd Qu.: 1.000 3rd Qu.: 3.000
> Max. :11.000 Max. :1.000000 Max. :18.000 Max. :13.000
> size3added repstra3 repstrb3 fecca3
> Min. : 0.000 Min. : 0.000 Min. :0.000000 Min. :0.0000
> 1st Qu.: 1.000 1st Qu.: 0.000 1st Qu.:0.000000 1st Qu.:0.0000
> Median : 2.000 Median : 0.000 Median :0.000000 Median :0.0000
> Mean : 3.288 Mean : 1.069 Mean :0.009375 Mean :0.4562
> 3rd Qu.: 4.000 3rd Qu.: 1.000 3rd Qu.:0.000000 3rd Qu.:0.0000
> Max. :24.000 Max. :18.000 Max. :1.000000 Max. :8.0000
> juvgiven3 obsstatus3 repstatus3 fecstatus3 matstatus3

```



```

> Min. :0 Min. :0.0 Min. :0.0 Min. :0.0000 Min. :1
> 1st Qu.:0 1st Qu.:1.0 1st Qu.:0.0 1st Qu.:0.0000 1st Qu.:1
> Median :0 Median :1.0 Median :0.0 Median :0.0000 Median :1
> Mean :0 Mean :0.9 Mean :0.4 Mean :0.2219 Mean :1
> 3rd Qu.:0 3rd Qu.:1.0 3rd Qu.:1.0 3rd Qu.:0.0000 3rd Qu.:1
> Max. :0 Max. :1.0 Max. :1.0 Max. :1.0000 Max. :1
> alive3 stage3 stage3index
> Min. :0.0000 Length:320 Min. : 0.000
> 1st Qu.:1.0000 Class :character 1st Qu.: 7.000
> Median :1.0000 Mode :character Median : 8.000
> Mean :0.9469 Mean : 7.544
> 3rd Qu.:1.0000 3rd Qu.: 8.000
> Max. :1.0000 Max. :11.000

```

In the summary of the resulting data frame, we see that the first 4 variables are identifying information - they show the data row (`rowid`), the population ID (`popid`), the patch ID (`patchid`), and the individual ID (`indivd`), in order. These are followed by two variables identifying in what monitoring occasion the individual was first seen (`firstseen`), and in what occasion it was last seen (`lastseen`). The next variable, `obsage`, gives the estimated age at time t assuming that the individual began life in the monitoring time in which it was first observed. The next variable, `obslifespan`, gives the full length of time that the individual was observed in the dataset. After that, we see a group of 15 variables typically ending with the number 1. These 15 variables correspond to the state of each individual in time $t-1$ (generally referred to as `time1` in the actual code). Similarly, the next 15 variables typically end with the number 2 and represent state in time t , while the last 15 variables typically end in the number 3 and refer to state in time $t+1$. Most of these variables should be obvious to interpret given the names (definitions are in the help file for this function). Note that the final 7 variables in each group are calculated by `verticalize3()`. These variables represent observation status, reproductive status, fecundity status, maturity status, status as alive or dead, stage name, and stage number (with reference to the stageframe), in order.

Let's also create our standardized data frame for the function-based MPM, as below. Remember that some of the settings need to change here because we will use a different life history model. Particularly, we are now going to separate adults not just by size but by reproductive status. So, we will NOT set `NRasRep = TRUE` here (the default is `NRasRep = FALSE`).

```

cypfb_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
  patchidcol = "patch", individcol = "plantid", blocksize = 4,
  sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
  repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
  stageassign = cypframe_fb, stagesize = "sizeadded", NAas0 = TRUE)

```

```

summary(cypfb_v1)
> rowid popid patchid indivd year2
> Min. : 1.00 :320 A: 93 Length:320 Min. :2004
> 1st Qu.:21.00 B:154 Class :character 1st Qu.:2005
> Median :37.50 C: 73 Mode :character Median :2006
> Mean :38.45 Mean :2006
> 3rd Qu.:56.00 3rd Qu.:2007
> Max. :77.00 Max. :2008
> firstseen lastseen obsage obslifespan
> Min. :2004 Min. :2004 Min. :0.000 Min. :0.000
> 1st Qu.:2004 1st Qu.:2009 1st Qu.:1.000 1st Qu.:5.000
> Median :2004 Median :2009 Median :2.000 Median :5.000
> Mean :2004 Mean :2009 Mean :1.853 Mean :4.556
> 3rd Qu.:2004 3rd Qu.:2009 3rd Qu.:3.000 3rd Qu.:5.000
> Max. :2008 Max. :2009 Max. :4.000 Max. :5.000

```

```

>      sizea1      sizeb1      sizec1      sizedadded
> Min.   :0.000000  Min.   : 0.0000  Min.   : 0.0  Min.   : 0.000
> 1st Qu.:0.000000  1st Qu.: 0.0000  1st Qu.: 0.0  1st Qu.: 0.000
> Median :0.000000  Median : 0.0000  Median : 1.0  Median : 2.000
> Mean   :0.009375  Mean   : 0.7469  Mean   : 1.9  Mean   : 2.656
> 3rd Qu.:0.000000  3rd Qu.: 1.0000  3rd Qu.: 3.0  3rd Qu.: 4.000
> Max.   :1.000000  Max.   :18.0000  Max.   :13.0  Max.   :21.000
>      repstra1      repstrb1      fecal      juvgiven1
> Min.   : 0.0000  Min.   :0.000000  Min.   :0.0000  Min.   :0
> 1st Qu.: 0.0000  1st Qu.:0.000000  1st Qu.:0.0000  1st Qu.:0
> Median : 0.0000  Median :0.000000  Median :0.0000  Median :0
> Mean   : 0.7469  Mean   :0.009375  Mean   :0.2656  Mean   :0
> 3rd Qu.: 1.0000  3rd Qu.:0.000000  3rd Qu.:0.0000  3rd Qu.:0
> Max.   :18.0000  Max.   :1.000000  Max.   :7.0000  Max.   :0
>      obsstatus1      repstatus1      fecstatus1      matstatus1
> Min.   :0.0000  Min.   :0.0000  Min.   :0.0000  Min.   :0.0000
> 1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:1.0000
> Median :1.0000  Median :0.0000  Median :0.0000  Median :1.0000
> Mean   :0.7469  Mean   :0.2875  Mean   :0.1344  Mean   :0.7688
> 3rd Qu.:1.0000  3rd Qu.:1.0000  3rd Qu.:0.0000  3rd Qu.:1.0000
> Max.   :1.0000  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000
>      alive1      stage1      stageindex      sizea2
> Min.   :0.0000  Length:320  Min.   : 0.00  Min.   :0.000000
> 1st Qu.:1.0000  Class :character  1st Qu.: 6.00  1st Qu.:0.000000
> Median :1.0000  Mode  :character  Median : 8.00  Median :0.000000
> Mean   :0.7688  Mean   :14.17  Mean   :14.17  Mean   :0.009375
> 3rd Qu.:1.0000  3rd Qu.:31.00  3rd Qu.:31.00  3rd Qu.:0.000000
> Max.   :1.0000  Max.   :51.00  Max.   :51.00  Max.   :1.000000
>      sizeb2      sizec2      size2added      repstra2
> Min.   : 0.0000  Min.   : 0.000  Min.   : 0.000  Min.   : 0.0000
> 1st Qu.: 0.0000  1st Qu.: 1.000  1st Qu.: 1.000  1st Qu.: 0.0000
> Median : 0.0000  Median : 2.000  Median : 2.000  Median : 0.0000
> Mean   : 0.8969  Mean   : 2.416  Mean   : 3.322  Mean   : 0.8969
> 3rd Qu.: 1.0000  3rd Qu.: 3.000  3rd Qu.: 4.000  3rd Qu.: 1.0000
> Max.   :18.0000  Max.   :13.000  Max.   :24.000  Max.   :18.0000
>      repstrb2      fecal      juvgiven2      obsstatus2
> Min.   :0.000000  Min.   :0.0000  Min.   :0  Min.   :0.0000
> 1st Qu.:0.000000  1st Qu.:0.0000  1st Qu.:0  1st Qu.:1.0000
> Median :0.000000  Median :0.0000  Median :0  Median :1.0000
> Mean   :0.009375  Mean   :0.2906  Mean   :0  Mean   :0.9531
> 3rd Qu.:0.000000  3rd Qu.:0.0000  3rd Qu.:0  3rd Qu.:1.0000
> Max.   :1.000000  Max.   :7.0000  Max.   :0  Max.   :1.0000
>      repstatus2      fecstatus2      matstatus2      alive2      stage2
> Min.   :0.0000  Min.   :0.0000  Min.   :1  Min.   :1  Length:320
> 1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:1  1st Qu.:1  Class :character
> Median :0.0000  Median :0.0000  Median :1  Median :1  Mode  :character
> Mean   :0.3688  Mean   :0.1562  Mean   :1  Mean   :1
> 3rd Qu.:1.0000  3rd Qu.:0.0000  3rd Qu.:1  3rd Qu.:1
> Max.   :1.0000  Max.   :1.0000  Max.   :1  Max.   :1
>      stage2index      sizea3      sizeb3      sizec3
> Min.   : 6.00  Min.   :0.000000  Min.   : 0.000  Min.   : 0.000
> 1st Qu.: 7.00  1st Qu.:0.000000  1st Qu.: 0.000  1st Qu.: 1.000
> Median :10.00  Median :0.000000  Median : 0.000  Median : 1.000

```



```

> Mean :18.17 Mean :0.009375 Mean : 1.069 Mean : 2.209
> 3rd Qu.:32.00 3rd Qu.:0.000000 3rd Qu.: 1.000 3rd Qu.: 3.000
> Max. :54.00 Max. :1.000000 Max. :18.000 Max. :13.000
> size3added repstra3 repstrb3 fecas3
> Min. : 0.000 Min. : 0.000 Min. :0.000000 Min. :0.0000
> 1st Qu.: 1.000 1st Qu.: 0.000 1st Qu.:0.000000 1st Qu.:0.0000
> Median : 2.000 Median : 0.000 Median :0.000000 Median :0.0000
> Mean : 3.288 Mean : 1.069 Mean :0.009375 Mean :0.4562
> 3rd Qu.: 4.000 3rd Qu.: 1.000 3rd Qu.:0.000000 3rd Qu.:0.0000
> Max. :24.000 Max. :18.000 Max. :1.000000 Max. :8.0000
> juvgiven3 obsstatus3 repstatus3 fecstatus3 matstatus3
> Min. :0 Min. :0.0 Min. :0.0 Min. :0.0000 Min. :1
> 1st Qu.:0 1st Qu.:1.0 1st Qu.:0.0 1st Qu.:0.0000 1st Qu.:1
> Median :0 Median :1.0 Median :0.0 Median :0.0000 Median :1
> Mean :0 Mean :0.9 Mean :0.4 Mean :0.2219 Mean :1
> 3rd Qu.:0 3rd Qu.:1.0 3rd Qu.:1.0 3rd Qu.:0.0000 3rd Qu.:1
> Max. :0 Max. :1.0 Max. :1.0 Max. :1.0000 Max. :1
> alive3 stage3 stage3index
> Min. :0.0000 Length:320 Min. : 0.00
> 1st Qu.:1.0000 Class :character 1st Qu.: 7.00
> Median :1.0000 Mode :character Median :10.00
> Mean :0.9469 Mean :18.57
> 3rd Qu.:1.0000 3rd Qu.:33.00
> Max. :1.0000 Max. :54.00

```

The output dataset includes a number of summary variables, but the data is essentially broken down into groups of three consecutive monitoring occasions each (occasions $t+1$, t , and $t-1$, corresponding to `year3`, `year2`, and `year1` in the output, respectively), with individuals spread across multiple rows. The output dataset is further limited to those entries in which the individual is alive in occasion t (`year2`), meaning that all rows in which an individual is dead or not yet recruited in occasion t are dropped. Since the input data is the same, we should see the same numbers of rows and columns in the raw and function-based cases, regardless of the different stageframes used. Thus, we have 320 rows of data and 54 variables in the raw case, and 320 rows of data and 54 variables in the function-based case.

Function `historicalize3()`

If the dataset is in *ahistorical vertical format*, in which an individual's condition across time is recorded across rows in a spreadsheet, with a single row corresponding to either a single observation or to a pair of consecutive observations, then the `historicalize3()` function can standardize the dataset properly. Here, the actual inputs are very similar to `verticalize3()`. However, the `historicalize3()` function assumes that the input data set is organized with rows corresponding to individual status in either only one monitoring occasion, or two consecutive occasions. Inputs include a series of variables more or less equivalent to input options in `verticalize3()`, but some variable names end with `2col`, and others end with `3col`. The former denote status in time t , and are the minimum required for this function to run. If each row includes status in paired consecutive times, then variables ending in `3col` can be used to designate status in time $t+1$. Additionally, this function requires a single variable identifying individuals across rows, so that each individual's resighting history can be inferred and each group of 3 consecutive monitoring times can be put together. All other options work essentially the same as in `verticalize3()`.

Package `lefko3` also includes dataset `cypvert`, which is the same dataset as `cypdata` but set in ahistorical vertical format. Here, we will use the `historicalize3()` function to standardize this dataset, using the `plantid` variable as the individual identity term.

```
data(cypvert)
```

```

cypraw_v2 <- historicalize3(data = cypvert, patchidcol = "patch",
  individcol = "plantid", year2col = "year2", sizea2col = "Inf2.2",
  sizea3col = "Inf2.3", sizeb2col = "Inf.2", sizeb3col = "Inf.3",
  sizec2col = "Veg.2", sizec3col = "Veg.3", repstra2col = "Inf2.2",
  repstra3col = "Inf2.3", repstrb2col = "Inf.2", repstrb3col = "Inf.3",
  fecal2col = "Pod.2", fecal3col = "Pod.3", repstrrel = 2,
  stageassign = cypframe_raw, stagesize = "sizeadded", censorcol = "censor",
  censorkeep = 1, censor = FALSE, NAas0 = TRUE, NRasRep = TRUE, reduce = TRUE)

```

```
summary(cypraw_v2)
```

```

>      rowid          popid          patchid          individ
> Min.   : 0.00   Length:320      Length:320      Length:320
> 1st Qu.: 82.75   Class :character  Class :character  Class :character
> Median :162.50   Mode  :character  Mode  :character  Mode  :character
> Mean   :163.70
> 3rd Qu.:245.25
> Max.   :330.00
>      year2.V1      firstseen.V1      lastseen.V1      obsage.V1
> Min.   :2004.0000  Min.   :2004.0000  Min.   :2004.0000  Min.   :0.000000
> 1st Qu.:2005.0000  1st Qu.:2004.0000  1st Qu.:2009.0000  1st Qu.:1.000000
> Median :2006.0000  Median :2004.0000  Median :2009.0000  Median :2.000000
> Mean   :2005.9562  Mean   :2004.1031  Mean   :2008.6594  Mean   :1.853125
> 3rd Qu.:2007.0000  3rd Qu.:2004.0000  3rd Qu.:2009.0000  3rd Qu.:3.000000
> Max.   :2008.0000  Max.   :2008.0000  Max.   :2009.0000  Max.   :4.000000
>      obslifespan.V1      sizea1      sizeb1      sizec1
> Min.   :0.000000  Min.   :0.000000  Min.   :0.0000  Min.   :0.0
> 1st Qu.:5.000000  1st Qu.:0.000000  1st Qu.:0.0000  1st Qu.:0.0
> Median :5.000000  Median :0.000000  Median :0.0000  Median :1.0
> Mean   :4.55625  Mean   :0.009375  Mean   :0.7469  Mean   :1.9
> 3rd Qu.:5.000000  3rd Qu.:0.000000  3rd Qu.:1.0000  3rd Qu.:3.0
> Max.   :5.000000  Max.   :1.000000  Max.   :18.0000  Max.   :13.0
>      sizedadded      repstra1      repstrb1      fecal
> Min.   :0.000  Min.   :0.000000  Min.   :0.0000  Min.   :0.0000
> 1st Qu.:0.000  1st Qu.:0.000000  1st Qu.:0.0000  1st Qu.:0.0000
> Median :2.000  Median :0.000000  Median :0.0000  Median :0.0000
> Mean   :2.656  Mean   :0.009375  Mean   :0.7469  Mean   :0.2656
> 3rd Qu.:4.000  3rd Qu.:0.000000  3rd Qu.:1.0000  3rd Qu.:0.0000
> Max.   :21.000  Max.   :1.000000  Max.   :18.0000  Max.   :7.0000
>      juvgiven1  obsstatus1  repstatus1  fecstatus1
> Min.   :0  Min.   :0.0000  Min.   :0.0000  Min.   :0.0000
> 1st Qu.:0  1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:0.0000
> Median :0  Median :1.0000  Median :0.0000  Median :0.0000
> Mean   :0  Mean   :0.7469  Mean   :0.2875  Mean   :0.1344
> 3rd Qu.:0  3rd Qu.:1.0000  3rd Qu.:1.0000  3rd Qu.:0.0000
> Max.   :0  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000
>      matstatus1  alive1  stage1  stage1index
> Min.   :0.0000  Min.   :0.0000  Length:320  Min.   :0.000
> 1st Qu.:1.0000  1st Qu.:1.0000  Class :character  1st Qu.:6.000
> Median :1.0000  Median :1.0000  Mode  :character  Median :8.000
> Mean   :0.7688  Mean   :0.7688  Mean   :6.144
> 3rd Qu.:1.0000  3rd Qu.:1.0000  3rd Qu.:8.000
> Max.   :1.0000  Max.   :1.0000  Max.   :11.000
>      sizea2      sizeb2      sizec2      size2added

```

```

> Min. :0.000000 Min. : 0.0000 Min. : 0.000 Min. : 0.000
> 1st Qu.:0.000000 1st Qu.: 0.0000 1st Qu.: 1.000 1st Qu.: 1.000
> Median :0.000000 Median : 0.0000 Median : 2.000 Median : 2.000
> Mean :0.009375 Mean : 0.8969 Mean : 2.416 Mean : 3.322
> 3rd Qu.:0.000000 3rd Qu.: 1.0000 3rd Qu.: 3.000 3rd Qu.: 4.000
> Max. :1.000000 Max. :18.0000 Max. :13.000 Max. :24.000
> repstra2 repstrb2 fecca2 juvgiven2
> Min. :0.000000 Min. : 0.0000 Min. :0.0000 Min. :0
> 1st Qu.:0.000000 1st Qu.: 0.0000 1st Qu.:0.0000 1st Qu.:0
> Median :0.000000 Median : 0.0000 Median :0.0000 Median :0
> Mean :0.009375 Mean : 0.8969 Mean :0.2906 Mean :0
> 3rd Qu.:0.000000 3rd Qu.: 1.0000 3rd Qu.:0.0000 3rd Qu.:0
> Max. :1.000000 Max. :18.0000 Max. :7.0000 Max. :0
> obsstatus2 repstatus2 fecstatus2 matstatus2 alive2
> Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :1 Min. :1
> 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:1 1st Qu.:1
> Median :1.0000 Median :0.0000 Median :0.0000 Median :1 Median :1
> Mean :0.9531 Mean :0.3688 Mean :0.1562 Mean :1 Mean :1
> 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1 3rd Qu.:1
> Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1 Max. :1
> stage2 stage2index sizea3 sizeb3
> Length:320 Min. : 6.000 Min. :0.000000 Min. : 0.000
> Class :character 1st Qu.: 7.000 1st Qu.:0.000000 1st Qu.: 0.000
> Mode :character Median : 8.000 Median :0.000000 Median : 0.000
> Mean : 7.919 Mean :0.009375 Mean : 1.069
> 3rd Qu.: 8.000 3rd Qu.:0.000000 3rd Qu.: 1.000
> Max. :11.000 Max. :1.000000 Max. :18.000
> sizec3 size3added repstra3 repstrb3
> Min. : 0.000 Min. : 0.000 Min. :0.000000 Min. : 0.000
> 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.:0.000000 1st Qu.: 0.000
> Median : 1.000 Median : 2.000 Median :0.000000 Median : 0.000
> Mean : 2.209 Mean : 3.288 Mean :0.009375 Mean : 1.069
> 3rd Qu.: 3.000 3rd Qu.: 4.000 3rd Qu.:0.000000 3rd Qu.: 1.000
> Max. :13.000 Max. :24.000 Max. :1.000000 Max. :18.000
> fecca3 juvgiven3 obsstatus3 repstatus3 fecstatus3
> Min. :0.0000 Min. :0 Min. :0.0 Min. :0.0 Min. :0.00000
> 1st Qu.:0.0000 1st Qu.:0 1st Qu.:1.0 1st Qu.:0.0 1st Qu.:0.00000
> Median :0.0000 Median :0 Median :1.0 Median :0.0 Median :0.00000
> Mean :0.4562 Mean :0 Mean :0.9 Mean :0.4 Mean :0.2219
> 3rd Qu.:0.0000 3rd Qu.:0 3rd Qu.:1.0 3rd Qu.:1.0 3rd Qu.:0.00000
> Max. :8.0000 Max. :0 Max. :1.0 Max. :1.0 Max. :1.00000
> matstatus3 alive3 stage3 stage3index
> Min. :1 Min. :0.0000 Length:320 Min. : 0.000
> 1st Qu.:1 1st Qu.:1.0000 Class :character 1st Qu.: 7.000
> Median :1 Median :1.0000 Mode :character Median : 8.000
> Mean :1 Mean :0.9469 Mean : 7.544
> 3rd Qu.:1 3rd Qu.:1.0000 3rd Qu.: 8.000
> Max. :1 Max. :1.0000 Max. :11.000

```

Let's also create the function-based MPM version.

```

cypfb_v2 <- historicalize3(data = cypvert, patchidcol = "patch",
  individcol = "plantid", year2col = "year2", sizea2col = "Inf2.2",
  sizea3col = "Inf2.3", sizeb2col = "Inf.2", sizeb3col = "Inf.3",

```

```

sizec2col = "Veg.2", sizec3col = "Veg.3", repstra2col = "Inf2.2",
repstra3col = "Inf2.3", repstrb2col = "Inf.2", repstrb3col = "Inf.3",
feca2col = "Pod.2", feca3col = "Pod.3", repstrrel = 2,
stageassign = cypframe_fb, stagesize = "sizeadded", censorcol = "censor",
censorkeep = 1, censor = FALSE, NAas0 = TRUE, reduce = TRUE)

```

```
summary(cypfb_v2)
```

```

>      rowid          popid          patchid          individ
> Min.   : 0.00   Length:320      Length:320      Length:320
> 1st Qu.: 82.75   Class :character  Class :character  Class :character
> Median :162.50   Mode  :character  Mode  :character  Mode  :character
> Mean   :163.70
> 3rd Qu.:245.25
> Max.   :330.00
>      year2.V1      firstseen.V1      lastseen.V1      obsage.V1
> Min.   :2004.0000  Min.   :2004.0000  Min.   :2004.0000  Min.   :0.000000
> 1st Qu.:2005.0000  1st Qu.:2004.0000  1st Qu.:2009.0000  1st Qu.:1.000000
> Median :2006.0000  Median :2004.0000  Median :2009.0000  Median :2.000000
> Mean   :2005.9562  Mean   :2004.1031  Mean   :2008.6594  Mean   :1.853125
> 3rd Qu.:2007.0000  3rd Qu.:2004.0000  3rd Qu.:2009.0000  3rd Qu.:3.000000
> Max.   :2008.0000  Max.   :2008.0000  Max.   :2009.0000  Max.   :4.000000
>      obslifespan.V1      sizea1      sizeb1      sizec1
> Min.   :0.000000  Min.   :0.000000  Min.   : 0.0000  Min.   : 0.0
> 1st Qu.:5.000000  1st Qu.:0.000000  1st Qu.: 0.0000  1st Qu.: 0.0
> Median :5.000000  Median :0.000000  Median : 0.0000  Median : 1.0
> Mean   :4.55625   Mean   :0.009375  Mean   : 0.7469  Mean   : 1.9
> 3rd Qu.:5.000000  3rd Qu.:0.000000  3rd Qu.: 1.0000  3rd Qu.: 3.0
> Max.   :5.000000  Max.   :1.000000  Max.   :18.0000  Max.   :13.0
>      sizedadded      repstra1      repstrb1      fecal
> Min.   : 0.000  Min.   :0.000000  Min.   : 0.0000  Min.   :0.0000
> 1st Qu.: 0.000  1st Qu.:0.000000  1st Qu.: 0.0000  1st Qu.:0.0000
> Median : 2.000  Median :0.000000  Median : 0.0000  Median :0.0000
> Mean   : 2.656  Mean   :0.009375  Mean   : 0.7469  Mean   :0.2656
> 3rd Qu.: 4.000  3rd Qu.:0.000000  3rd Qu.: 1.0000  3rd Qu.:0.0000
> Max.   :21.000  Max.   :1.000000  Max.   :18.0000  Max.   :7.0000
>      juvgiven1  obsstatus1      repstatus1      fecstatus1
> Min.   :0  Min.   :0.0000  Min.   :0.0000  Min.   :0.0000
> 1st Qu.:0  1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:0.0000
> Median :0  Median :1.0000  Median :0.0000  Median :0.0000
> Mean   :0  Mean   :0.7469  Mean   :0.2875  Mean   :0.1344
> 3rd Qu.:0  3rd Qu.:1.0000  3rd Qu.:1.0000  3rd Qu.:0.0000
> Max.   :0  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000
>      matstatus1      alive1      stage1      stage1index
> Min.   :0.0000  Min.   :0.0000  Length:320  Min.   : 0.00
> 1st Qu.:1.0000  1st Qu.:1.0000  Class :character  1st Qu.: 6.00
> Median :1.0000  Median :1.0000  Mode  :character  Median : 8.00
> Mean   :0.7688  Mean   :0.7688  Mean   :14.17
> 3rd Qu.:1.0000  3rd Qu.:1.0000  3rd Qu.:31.00
> Max.   :1.0000  Max.   :1.0000  Max.   :51.00
>      sizea2      sizeb2      sizec2      size2added
> Min.   :0.000000  Min.   : 0.0000  Min.   : 0.000  Min.   : 0.000
> 1st Qu.:0.000000  1st Qu.: 0.0000  1st Qu.: 1.000  1st Qu.: 1.000
> Median :0.000000  Median : 0.0000  Median : 2.000  Median : 2.000

```

```

> Mean :0.009375 Mean : 0.8969 Mean : 2.416 Mean : 3.322
> 3rd Qu.:0.000000 3rd Qu.: 1.0000 3rd Qu.: 3.000 3rd Qu.: 4.000
> Max. :1.000000 Max. :18.0000 Max. :13.000 Max. :24.000
> repstra2 repstrb2 feca2 juvgiven2
> Min. :0.000000 Min. : 0.0000 Min. :0.0000 Min. : 0
> 1st Qu.:0.000000 1st Qu.: 0.0000 1st Qu.:0.0000 1st Qu.:0
> Median :0.000000 Median : 0.0000 Median :0.0000 Median :0
> Mean :0.009375 Mean : 0.8969 Mean :0.2906 Mean : 0
> 3rd Qu.:0.000000 3rd Qu.: 1.0000 3rd Qu.:0.0000 3rd Qu.:0
> Max. :1.000000 Max. :18.0000 Max. :7.0000 Max. : 0
> obsstatus2 repstatus2 fecstatus2 matstatus2 alive2
> Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :1 Min. :1
> 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:1 1st Qu.:1
> Median :1.0000 Median :0.0000 Median :0.0000 Median :1 Median :1
> Mean :0.9531 Mean :0.3688 Mean :0.1562 Mean :1 Mean :1
> 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1 3rd Qu.:1
> Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1 Max. :1
> stage2 stage2index sizea3 sizeb3
> Length:320 Min. : 6.00 Min. :0.000000 Min. : 0.000
> Class :character 1st Qu.: 7.00 1st Qu.:0.000000 1st Qu.: 0.000
> Mode :character Median :10.00 Median :0.000000 Median : 0.000
> Mean :18.17 Mean :0.009375 Mean : 1.069
> 3rd Qu.:32.00 3rd Qu.:0.000000 3rd Qu.: 1.000
> Max. :54.00 Max. :1.000000 Max. :18.000
> sizec3 size3added repstra3 repstrb3
> Min. : 0.000 Min. : 0.000 Min. :0.000000 Min. : 0.000
> 1st Qu.: 1.000 1st Qu.: 1.000 1st Qu.:0.000000 1st Qu.: 0.000
> Median : 1.000 Median : 2.000 Median :0.000000 Median : 0.000
> Mean : 2.209 Mean : 3.288 Mean :0.009375 Mean : 1.069
> 3rd Qu.: 3.000 3rd Qu.: 4.000 3rd Qu.:0.000000 3rd Qu.: 1.000
> Max. :13.000 Max. :24.000 Max. :1.000000 Max. :18.000
> feca3 juvgiven3 obsstatus3 repstatus3 fecstatus3
> Min. :0.0000 Min. :0 Min. :0.0 Min. :0.0 Min. :0.00000
> 1st Qu.:0.0000 1st Qu.:0 1st Qu.:1.0 1st Qu.:0.0 1st Qu.:0.00000
> Median :0.0000 Median :0 Median :1.0 Median :0.0 Median :0.00000
> Mean :0.4562 Mean :0 Mean :0.9 Mean :0.4 Mean :0.2219
> 3rd Qu.:0.0000 3rd Qu.:0 3rd Qu.:1.0 3rd Qu.:1.0 3rd Qu.:0.00000
> Max. :8.0000 Max. :0 Max. :1.0 Max. :1.0 Max. :1.00000
> matstatus3 alive3 stage3 stage3index
> Min. :1 Min. :0.0000 Length:320 Min. : 0.00
> 1st Qu.:1 1st Qu.:1.0000 Class :character 1st Qu.: 7.00
> Median :1 Median :1.0000 Mode :character Median :10.00
> Mean :1 Mean :0.9469 Mean :18.57
> 3rd Qu.:1 3rd Qu.:1.0000 3rd Qu.:33.00
> Max. :1 Max. :1.0000 Max. :54.00

```

One important consideration is the use of censor variables. Censoring a demographic dataset is typically accomplished by including a binary variable in the dataset denoting whether an individual datum is to be kept or excluded. The objects `cypraw_v2` and `cpyfb_v2` were both created without using censoring variables. However, because the datasets actually included censor variables (all data were set to be included, with no suspect data), we wished to incorporate those variables in the final datasets. Hence, although `sensor = FALSE` in both the call to `verticalize3()` and the call to `historicalize3()`, we also noted `sensorcol = "sensor"` and `sensorkeep = 1` in the call to `historicalize3()`. Failing to add these options to the call to `historicalize3()` will produce approximately the same dataset, but with some zeroes entering variables

`sensor1`, `sensor2`, and `sensor3` in the historicalized dataset that do not exist in the first, verticalized dataset (this occurs because the vertical dataset that we standardized only includes entries for each individual actually observed at each time, while the horizontal dataset includes entries for all individuals at all times, regardless of whether they were observed). This would lead to small differences in the resulting datasets. Instead, we have produced datasets that are exactly the same except for variable order. We can see hints of that by looking at the dimensions of our datasets, as below.

```
dim(cypraw_v1)
> [1] 320 54
dim(cypraw_v2)
> [1] 320 54
dim(cypfb_v1)
> [1] 320 54
dim(cypfb_v2)
> [1] 320 54
```

Let's now move on to supplying R with the proxy transitions that we need to properly parameterize our models.

Formatting supplemental data needed for matrix estimation

MPMs are often estimated only partially from available demographic datasets. Some transitions are parameterized using information gathered from other studies, whether through direct input in the matrix or through the development of kernels contingent on external information. Other transitions might also be estimated via proxy transitions elsewhere in the matrix. In `lefko3`, this information can be provided in one of two ways. The preferred, and most recently developed, method is through **supplement tables**, which can be developed using the `supplemental()` function. This function allows users to create a data frame detailing:

- 1) specific transitions to set as constants,
- 2) specific transitions to estimate via proxy transitions, which can also be multiplied by fixed values, and
- 3) specific multipliers for fecundity in cases in which fecundity estimates resulting from linear modeling must be modified to characterize the full transition.

Examples might illustrate where this approach is useful. If I lack my own data on subterranean juvenile stages in a plant species, but I have estimates of survival for those stages from another study, then I might use those estimates as constants in the MPM. Further, if I lack demographic data on the development of germinated seeds to the seedling or earliest adult stage, but I have reason to believe that the survival probabilities should be similar to the survival within an observed stage of seedlings or small adults, then I can use the latter survival-transitions as proxies. Finally, if fecundity is a function of seed production, survival to the next year, and germination probability, then germination probability might be estimated via a separate field germination study. I might wish to incorporate this germination probability both as a constant transition, and as a multiplier on estimated fecundity. Supplement tables provide a means to include all of this information.

Function `supplemental()` was developed to provide a robust, general solution to the addition of these data to MPMs. To use this function, first identify each transition that needs to be altered, and split each transition into stage in times $t+1$, t , and, if conducting a historical analysis, $t-1$. These stages must then be entered in the same order across the `stage3`, `stage2`, and `stage1` fields. For example, if I wished to alter the historical transitions from stage A to B to C, and from A to A to C, then I would enter these two transitions as `stage3 = c("C", "C")`, `stage2 = c("B", "A")`, `stage1 = c("A", "A")`. If I were interested in ahistorical transitions, then I would either set all `stage1` values to NA, such as `stage1 = c(NA, NA)`, or simply not include `stage` in the input (the default is a vector of NA elements). If I wished to replace my two transitions with proxy transitions that `lefko3` will estimate, then I can designate those proxy transitions using the `eststage` fields. For example, `eststage3 = c("F", "F")`, `eststage2 = c("E", "D")`, `eststage1 = c("D", "D")` would replace my original transitions with the values estimated for transitions D-E-F and D-D-F, respectively. Alternatively, if I wished to set these two transitions to fixed

constants, then I could designate those with `givenrate`. For example, `givenrate = c(0.15, 0.25)` would set these two transitions to 0.15 and 0.25, respectively. Estimated fecundity, as well as proxy values, can be modified by a multiplier using the `multiplier` field, and it is important to identify the kind of transition that you wish to modify as a survival transition or a fecundity rate using the `type` field (use `type12` also to designate what kind of transition is occurring between times $t-1$ and t in the historical case). Fecundity multipliers should be identified separately, since they are not proxy values. The function requires the input of the stageframe used to generate the stages.

We will start off by creating two supplement tables taking all of these sorts of data for the function-based MPMs. The first is the supplement table for the historical analysis, and the second table covers the ahistorical analysis. Each row refers to a specific transition, and in the historical case, there are codes for 16 given transitions (11 for the ahistorical case). The first 9 of the historical transitions are set to specific probabilities, and the next 5 are transitions that will be set to other, estimated transitions (these are the non-NA transitions in `eststage` set below). The final 2 terms are fecundity multipliers. Based on the literature, the proxies for entry into the adult classes are transitions from dormancy, as below. However, in the raw dataset, dormancy is not common enough to use as an effective proxy in raw matrix creation. Hence, we can use different proxies for function-based matrix estimation than for raw matrix estimation. Where necessary, we also use `rep` and `mat` as shorthand to code for all reproductive stages and all mature stages, respectively.

Here is the historical supplement table. We will also introduce a variable representing the number of seeds expected per fruit. Note that the number of seeds per fruit is actually likely higher than this, but we are using this number to reflect the relatively stable population dynamics that we see in the population.

```
seeds_per_fruit <- 5000

cypsupp3_fb <- supplemental(stage3 = c("SD", "SD", "P1", "P1", "P2", "P3", "SL",
  "SL", "SL", "D", "V1", "D", "V1", "V2", "SD", "P1"),
  stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL",
  "SL", "SL", "SL", "rep", "rep"),
  stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "SL", "P3", "P3",
  "SL", "SL", "SL", "mat", "mat"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "D", "V1", "D", "V1",
  "V2", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "D", "D", "D", "D",
  "D", NA, NA),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "D", "D", "D", "D",
  "D", NA, NA),
  givenrate = c(0.01, 0.05, 0.10, 0.20, 0.1, 0.1, 0.05, 0.05, 0.05, NA, NA, NA,
  NA, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
  (0.5 * seeds_per_fruit), (0.5 * seeds_per_fruit)),
  type = c("S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S",
  "R", "R"),
  type_t12 = c("S", "F", "S", "F", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S",
  "S", "S", "S"), stageframe = cypframe_fb)

cypsupp3_fb
>   stage3 stage2 stage1 eststage3 eststage2 eststage1 givenrate multiplier
> 1     SD     SD     SD     <NA>     <NA>     <NA>     0.01         1
> 2     SD     SD     rep     <NA>     <NA>     <NA>     0.05         1
> 3     P1     SD     SD     <NA>     <NA>     <NA>     0.10         1
> 4     P1     SD     rep     <NA>     <NA>     <NA>     0.20         1
> 5     P2     P1     SD     <NA>     <NA>     <NA>     0.10         1
> 6     P3     P2     P1     <NA>     <NA>     <NA>     0.10         1
> 7     SL     P3     P2     <NA>     <NA>     <NA>     0.05         1
> 8     SL     SL     P3     <NA>     <NA>     <NA>     0.05         1
```



```

> 9      SL      SL      SL      <NA>      <NA>      <NA>      0.05      1
> 10     D      SL      P3      D          D          D          NA        1
> 11     V1     SL      P3      V1         D          D          NA        1
> 12     D      SL      SL      D          D          D          NA        1
> 13     V1     SL      SL      V1         D          D          NA        1
> 14     V2     SL      SL      V2         D          D          NA        1
> 15     SD     rep     mat     <NA>      <NA>      <NA>      NA        2500
> 16     P1     rep     mat     <NA>      <NA>      <NA>      NA        2500
>      convtype convtype_t12
> 1          1          1
> 2          1          2
> 3          1          1
> 4          1          2
> 5          1          1
> 6          1          1
> 7          1          1
> 8          1          1
> 9          1          1
> 10         1          1
> 11         1          1
> 12         1          1
> 13         1          1
> 14         1          1
> 15         3          1
> 16         3          1

```

Now we can input the ahistorical supplement table.

```

cypsupp2_fb <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
  "V1", "V2", "SD", "P1"),
  stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL", "rep",
  "rep"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "D", "V1", "V2", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "D", "D", "D", NA, NA),
  givenrate = c(0.03, 0.15, 0.1, 0.1, 0.1, 0.05, NA, NA, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, (0.5* seeds_per_fruit),
  (0.5 * seeds_per_fruit)),
  type = c("S", "S", "S", "S", "S", "S", "S", "S", "S", "R", "R"),
  stageframe = cypframe_fb, historical = FALSE)

```

```

cypsupp2_fb
>      stage3 stage2 stage1 eststage3 eststage2 eststage1 givenrate multiplier
> 1      SD      SD      <NA>      <NA>      <NA>      <NA>      0.03      1
> 2      P1      SD      <NA>      <NA>      <NA>      <NA>      0.15      1
> 3      P2      P1      <NA>      <NA>      <NA>      <NA>      0.10      1
> 4      P3      P2      <NA>      <NA>      <NA>      <NA>      0.10      1
> 5      SL      P3      <NA>      <NA>      <NA>      <NA>      0.10      1
> 6      SL      SL      <NA>      <NA>      <NA>      <NA>      0.05      1
> 7      D      SL      <NA>      D          D          <NA>      NA        1
> 8      V1     SL      <NA>      V1         D          <NA>      NA        1
> 9      V2     SL      <NA>      V2         D          <NA>      NA        1
> 10     SD     rep     <NA>      <NA>      <NA>      <NA>      NA        2500
> 11     P1     rep     <NA>      <NA>      <NA>      <NA>      NA        2500
>      convtype convtype_t12

```

```

> 1      1      1
> 2      1      1
> 3      1      1
> 4      1      1
> 5      1      1
> 6      1      1
> 7      1      1
> 8      1      1
> 9      1      1
> 10     3      1
> 11     3      1

```

These supplement tables provide the best means of adding external data to our MPMs because they allow both specific transitions to be isolated, and because they allow the use of shorthand to identify large groups of transitions (e.g. using `mat`, `rep`, `immat`, `prop`, or `all` to signify all mature stages, reproductive stages, immature stages, propagule stages, or simply all stages, respectively). Let's now also supply the supplement tables for the raw MPMs.

```

cypsupp3_raw <- supplemental(stage3 = c("SD", "SD", "P1", "P1", "P2", "P3",
    "SL", "SL", "SL", "D", "D", "SD", "P1"),
  stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL",
    "rep", "rep"),
  stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "SL", "P3",
    "SL", "mat", "mat"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "XSm", "D", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", NA, NA),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", NA, NA),
  givenrate = c(0.01, 0.05, 0.10, 0.20, 0.1, 0.1, 0.05, 0.05, 0.05, NA, NA,
    NA, NA),
  multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
    (0.5 * seeds_per_fruit), (0.5 * seeds_per_fruit)),
  type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
  type_t12 = c(1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1),
  stageframe = cypframe_raw, historical = TRUE)

```

```

cypsupp3_raw
>   stage3 stage2 stage1 eststage3 eststage2 eststage1 givenrate multiplier
> 1     SD     SD     SD     <NA>     <NA>     <NA>     0.01      1
> 2     SD     SD     rep     <NA>     <NA>     <NA>     0.05      1
> 3     P1     SD     SD     <NA>     <NA>     <NA>     0.10      1
> 4     P1     SD     rep     <NA>     <NA>     <NA>     0.20      1
> 5     P2     P1     SD     <NA>     <NA>     <NA>     0.10      1
> 6     P3     P2     P1     <NA>     <NA>     <NA>     0.10      1
> 7     SL     P3     P2     <NA>     <NA>     <NA>     0.05      1
> 8     SL     SL     P3     <NA>     <NA>     <NA>     0.05      1
> 9     SL     SL     SL     <NA>     <NA>     <NA>     0.05      1
> 10    D     SL     P3     XSm     XSm     XSm     NA        1
> 11    D     SL     SL     D     XSm     XSm     NA        1
> 12    SD     rep     mat     <NA>     <NA>     <NA>     NA        2500
> 13    P1     rep     mat     <NA>     <NA>     <NA>     NA        2500
>   convtype convtype_t12
> 1         1         1
> 2         1         2
> 3         1         1

```

```

> 4      1      2
> 5      1      1
> 6      1      1
> 7      1      1
> 8      1      1
> 9      1      1
> 10     1      1
> 11     1      1
> 12     3      1
> 13     3      1

cypsupp2_raw <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
    "XSm", "SD", "P1"),
    stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "rep", "rep"),
    eststage3 = c(NA, NA, NA, NA, NA, NA, "D", "XSm", NA, NA),
    eststage2 = c(NA, NA, NA, NA, NA, NA, "XSm", "XSm", NA, NA),
    givenrate = c(0.03, 0.15, 0.1, 0.1, 0.1, 0.05, NA, NA, NA, NA),
    multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, (0.5 * seeds_per_fruit),
        (0.5 * seeds_per_fruit)),
    type = c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
    stageframe = cypframe_raw, historical = FALSE)

cypsupp2_raw
>   stage3 stage2 stage1 eststage3 eststage2 eststage1 givenrate multiplier
> 1     SD     SD  <NA>     <NA>     <NA>     <NA>     0.03         1
> 2     P1     SD  <NA>     <NA>     <NA>     <NA>     0.15         1
> 3     P2     P1  <NA>     <NA>     <NA>     <NA>     0.10         1
> 4     P3     P2  <NA>     <NA>     <NA>     <NA>     0.10         1
> 5     SL     P3  <NA>     <NA>     <NA>     <NA>     0.10         1
> 6     SL     SL  <NA>     <NA>     <NA>     <NA>     0.05         1
> 7       D     SL  <NA>         D     XSm     <NA>     NA           1
> 8     XSm     SL  <NA>         XSm    XSm     <NA>     NA           1
> 9       SD    rep  <NA>     <NA>     <NA>     <NA>     NA          2500
> 10     P1    rep  <NA>     <NA>     <NA>     <NA>     NA          2500
>   convtype convtype_t12
> 1         1             1
> 2         1             1
> 3         1             1
> 4         1             1
> 5         1             1
> 6         1             1
> 7         1             1
> 8         1             1
> 9         3             1
> 10        3             1

```

An older alternative to the supplement table Supplement tables provide all of the information that we need. However, there is an alternative means of supplying this information, and that is through the creation of a *reproductive matrix* and an *overwrite* table. The *reproductive matrix* is an ahistorical matrix detailing where fecundity rates are set, and at what level. *Cypripedium candidum* produces seeds that germinate by the following growing season (stage P1, or a first year protocorm), or that remain dormant for the next year (stage SD). In the following matrix, we detail that the fecundity of each reproductive stage needs to be split into two between each of these output stages. The actual split places 50% of the fecundity

of a reproductive stage into each category of recruit, where the full fecundity is estimated by linear models that we will create. This reproductive matrix can be used in historical MPM construction, and it is assumed that stage at occasion $t-1$ is set to all.

```

rep.assumptions_fb <- matrix(0, 54, 54)
rep.assumptions_fb[1:2,31:54] <- 0.5 * seeds_per_fruit
rep.assumptions_fb
>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
> [1,]    0    0    0    0    0    0    0    0    0    0    0    0    0
> [2,]    0    0    0    0    0    0    0    0    0    0    0    0    0
> [3,]    0    0    0    0    0    0    0    0    0    0    0    0    0
> [4,]    0    0    0    0    0    0    0    0    0    0    0    0    0
> [5,]    0    0    0    0    0    0    0    0    0    0    0    0    0
> [6,]    0    0    0    0    0    0    0    0    0    0    0    0    0
> [7,]    0    0    0    0    0    0    0    0    0    0    0    0    0
> [8,]    0    0    0    0    0    0    0    0    0    0    0    0    0
> [9,]    0    0    0    0    0    0    0    0    0    0    0    0    0
> [10,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [11,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [12,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [13,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [14,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [15,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [16,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [17,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [18,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [19,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [20,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [21,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [22,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [23,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [24,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [25,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [26,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [27,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [28,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [29,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [30,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [31,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [32,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [33,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [34,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [35,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [36,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [37,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [38,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [39,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [40,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [41,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [42,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [43,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [44,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [45,]   0    0    0    0    0    0    0    0    0    0    0    0    0
> [46,]   0    0    0    0    0    0    0    0    0    0    0    0    0

```

```
> [47,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [48,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [49,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [50,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [51,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [52,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [53,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [54,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
>      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
> [1,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [2,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [3,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [4,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [5,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [6,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [7,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [8,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [9,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [10,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [11,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [12,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [13,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [14,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [15,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [16,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [17,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [18,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [19,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [20,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [21,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [22,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [23,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [24,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [25,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [26,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [27,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [28,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [29,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [30,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [31,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [32,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [33,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [34,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [35,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [36,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [37,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [38,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [39,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [40,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [41,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [42,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [43,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
> [44,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```

> [45,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [46,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [47,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [48,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [49,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [50,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [51,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [52,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [53,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [54,] 0 0 0 0 0 0 0 0 0 0 0 0 0
>      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
> [1,] 0 0 0 0 0 2500 2500 2500 2500 2500 2500 2500
> [2,] 0 0 0 0 0 2500 2500 2500 2500 2500 2500 2500
> [3,] 0 0 0 0 0 0 0 0 0 0 0 0
> [4,] 0 0 0 0 0 0 0 0 0 0 0 0
> [5,] 0 0 0 0 0 0 0 0 0 0 0 0
> [6,] 0 0 0 0 0 0 0 0 0 0 0 0
> [7,] 0 0 0 0 0 0 0 0 0 0 0 0
> [8,] 0 0 0 0 0 0 0 0 0 0 0 0
> [9,] 0 0 0 0 0 0 0 0 0 0 0 0
> [10,] 0 0 0 0 0 0 0 0 0 0 0 0
> [11,] 0 0 0 0 0 0 0 0 0 0 0 0
> [12,] 0 0 0 0 0 0 0 0 0 0 0 0
> [13,] 0 0 0 0 0 0 0 0 0 0 0 0
> [14,] 0 0 0 0 0 0 0 0 0 0 0 0
> [15,] 0 0 0 0 0 0 0 0 0 0 0 0
> [16,] 0 0 0 0 0 0 0 0 0 0 0 0
> [17,] 0 0 0 0 0 0 0 0 0 0 0 0
> [18,] 0 0 0 0 0 0 0 0 0 0 0 0
> [19,] 0 0 0 0 0 0 0 0 0 0 0 0
> [20,] 0 0 0 0 0 0 0 0 0 0 0 0
> [21,] 0 0 0 0 0 0 0 0 0 0 0 0
> [22,] 0 0 0 0 0 0 0 0 0 0 0 0
> [23,] 0 0 0 0 0 0 0 0 0 0 0 0
> [24,] 0 0 0 0 0 0 0 0 0 0 0 0
> [25,] 0 0 0 0 0 0 0 0 0 0 0 0
> [26,] 0 0 0 0 0 0 0 0 0 0 0 0
> [27,] 0 0 0 0 0 0 0 0 0 0 0 0
> [28,] 0 0 0 0 0 0 0 0 0 0 0 0
> [29,] 0 0 0 0 0 0 0 0 0 0 0 0
> [30,] 0 0 0 0 0 0 0 0 0 0 0 0
> [31,] 0 0 0 0 0 0 0 0 0 0 0 0
> [32,] 0 0 0 0 0 0 0 0 0 0 0 0
> [33,] 0 0 0 0 0 0 0 0 0 0 0 0
> [34,] 0 0 0 0 0 0 0 0 0 0 0 0
> [35,] 0 0 0 0 0 0 0 0 0 0 0 0
> [36,] 0 0 0 0 0 0 0 0 0 0 0 0
> [37,] 0 0 0 0 0 0 0 0 0 0 0 0
> [38,] 0 0 0 0 0 0 0 0 0 0 0 0
> [39,] 0 0 0 0 0 0 0 0 0 0 0 0
> [40,] 0 0 0 0 0 0 0 0 0 0 0 0
> [41,] 0 0 0 0 0 0 0 0 0 0 0 0
> [42,] 0 0 0 0 0 0 0 0 0 0 0 0

```

```

> [43,] 0 0 0 0 0 0 0 0 0 0 0 0
> [44,] 0 0 0 0 0 0 0 0 0 0 0 0
> [45,] 0 0 0 0 0 0 0 0 0 0 0 0
> [46,] 0 0 0 0 0 0 0 0 0 0 0 0
> [47,] 0 0 0 0 0 0 0 0 0 0 0 0
> [48,] 0 0 0 0 0 0 0 0 0 0 0 0
> [49,] 0 0 0 0 0 0 0 0 0 0 0 0
> [50,] 0 0 0 0 0 0 0 0 0 0 0 0
> [51,] 0 0 0 0 0 0 0 0 0 0 0 0
> [52,] 0 0 0 0 0 0 0 0 0 0 0 0
> [53,] 0 0 0 0 0 0 0 0 0 0 0 0
> [54,] 0 0 0 0 0 0 0 0 0 0 0 0
>      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49]
> [1,] 2500 2500 2500 2500 2500 2500 2500 2500 2500 2500 2500 2500
> [2,] 2500 2500 2500 2500 2500 2500 2500 2500 2500 2500 2500 2500
> [3,] 0 0 0 0 0 0 0 0 0 0 0 0
> [4,] 0 0 0 0 0 0 0 0 0 0 0 0
> [5,] 0 0 0 0 0 0 0 0 0 0 0 0
> [6,] 0 0 0 0 0 0 0 0 0 0 0 0
> [7,] 0 0 0 0 0 0 0 0 0 0 0 0
> [8,] 0 0 0 0 0 0 0 0 0 0 0 0
> [9,] 0 0 0 0 0 0 0 0 0 0 0 0
> [10,] 0 0 0 0 0 0 0 0 0 0 0 0
> [11,] 0 0 0 0 0 0 0 0 0 0 0 0
> [12,] 0 0 0 0 0 0 0 0 0 0 0 0
> [13,] 0 0 0 0 0 0 0 0 0 0 0 0
> [14,] 0 0 0 0 0 0 0 0 0 0 0 0
> [15,] 0 0 0 0 0 0 0 0 0 0 0 0
> [16,] 0 0 0 0 0 0 0 0 0 0 0 0
> [17,] 0 0 0 0 0 0 0 0 0 0 0 0
> [18,] 0 0 0 0 0 0 0 0 0 0 0 0
> [19,] 0 0 0 0 0 0 0 0 0 0 0 0
> [20,] 0 0 0 0 0 0 0 0 0 0 0 0
> [21,] 0 0 0 0 0 0 0 0 0 0 0 0
> [22,] 0 0 0 0 0 0 0 0 0 0 0 0
> [23,] 0 0 0 0 0 0 0 0 0 0 0 0
> [24,] 0 0 0 0 0 0 0 0 0 0 0 0
> [25,] 0 0 0 0 0 0 0 0 0 0 0 0
> [26,] 0 0 0 0 0 0 0 0 0 0 0 0
> [27,] 0 0 0 0 0 0 0 0 0 0 0 0
> [28,] 0 0 0 0 0 0 0 0 0 0 0 0
> [29,] 0 0 0 0 0 0 0 0 0 0 0 0
> [30,] 0 0 0 0 0 0 0 0 0 0 0 0
> [31,] 0 0 0 0 0 0 0 0 0 0 0 0
> [32,] 0 0 0 0 0 0 0 0 0 0 0 0
> [33,] 0 0 0 0 0 0 0 0 0 0 0 0
> [34,] 0 0 0 0 0 0 0 0 0 0 0 0
> [35,] 0 0 0 0 0 0 0 0 0 0 0 0
> [36,] 0 0 0 0 0 0 0 0 0 0 0 0
> [37,] 0 0 0 0 0 0 0 0 0 0 0 0
> [38,] 0 0 0 0 0 0 0 0 0 0 0 0
> [39,] 0 0 0 0 0 0 0 0 0 0 0 0
> [40,] 0 0 0 0 0 0 0 0 0 0 0 0

```



```
> [41,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [42,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [43,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [44,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [45,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [46,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [47,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [48,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [49,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [50,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [51,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [52,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [53,] 0 0 0 0 0 0 0 0 0 0 0 0 0
> [54,] 0 0 0 0 0 0 0 0 0 0 0 0 0
>      [,50] [,51] [,52] [,53] [,54]
> [1,] 2500 2500 2500 2500 2500
> [2,] 2500 2500 2500 2500 2500
> [3,] 0 0 0 0 0
> [4,] 0 0 0 0 0
> [5,] 0 0 0 0 0
> [6,] 0 0 0 0 0
> [7,] 0 0 0 0 0
> [8,] 0 0 0 0 0
> [9,] 0 0 0 0 0
> [10,] 0 0 0 0 0
> [11,] 0 0 0 0 0
> [12,] 0 0 0 0 0
> [13,] 0 0 0 0 0
> [14,] 0 0 0 0 0
> [15,] 0 0 0 0 0
> [16,] 0 0 0 0 0
> [17,] 0 0 0 0 0
> [18,] 0 0 0 0 0
> [19,] 0 0 0 0 0
> [20,] 0 0 0 0 0
> [21,] 0 0 0 0 0
> [22,] 0 0 0 0 0
> [23,] 0 0 0 0 0
> [24,] 0 0 0 0 0
> [25,] 0 0 0 0 0
> [26,] 0 0 0 0 0
> [27,] 0 0 0 0 0
> [28,] 0 0 0 0 0
> [29,] 0 0 0 0 0
> [30,] 0 0 0 0 0
> [31,] 0 0 0 0 0
> [32,] 0 0 0 0 0
> [33,] 0 0 0 0 0
> [34,] 0 0 0 0 0
> [35,] 0 0 0 0 0
> [36,] 0 0 0 0 0
> [37,] 0 0 0 0 0
> [38,] 0 0 0 0 0
```

```

> [39,]    0    0    0    0    0
> [40,]    0    0    0    0    0
> [41,]    0    0    0    0    0
> [42,]    0    0    0    0    0
> [43,]    0    0    0    0    0
> [44,]    0    0    0    0    0
> [45,]    0    0    0    0    0
> [46,]    0    0    0    0    0
> [47,]    0    0    0    0    0
> [48,]    0    0    0    0    0
> [49,]    0    0    0    0    0
> [50,]    0    0    0    0    0
> [51,]    0    0    0    0    0
> [52,]    0    0    0    0    0
> [53,]    0    0    0    0    0
> [54,]    0    0    0    0    0

```

Next we will create the overwrite tables, which outline transitions that cannot be estimated from the data set and need to be set by other means. Here is an example of two overwrite tables for the *Cypripedium candidum* analysis. Note that they are fundamentally similar to supplement tables, but do not allow the addition of multipliers and transition types from occasion $t-1$ to t , nor do they involve a check of entered stages against the stageframe.

```

cypover3_fb <- overwrite(stage3 = c("SD", "SD", "P1", "P1", "P2", "P3", "SL", "SL",
  "SL", "D", "V1", "D", "V1", "V2"),
  stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL",
  "SL", "SL", "SL"),
  stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "SL", "P3", "P3",
  "SL", "SL", "SL"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "D", "V1", "D",
  "V1", "V2"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "D", "D", "D", "D",
  "D"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "D", "D", "D", "D",
  "D"),
  givenrate = c(0.01, 0.05, 0.10, 0.20, 0.1, 0.1, 0.05, 0.05, 0.05, NA, NA, NA,
  NA, NA),
  type = c("S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S",
  "S"))

```

```

cypover3_fb
>   stage3 stage2 stage1 eststage3 eststage2 eststage1 givenrate convtype
> 1     SD     SD     SD      <NA>      <NA>      <NA>      0.01      1
> 2     SD     SD     rep      <NA>      <NA>      <NA>      0.05      1
> 3     P1     SD     SD      <NA>      <NA>      <NA>      0.10      1
> 4     P1     SD     rep      <NA>      <NA>      <NA>      0.20      1
> 5     P2     P1     SD      <NA>      <NA>      <NA>      0.10      1
> 6     P3     P2     P1      <NA>      <NA>      <NA>      0.10      1
> 7     SL     P3     P2      <NA>      <NA>      <NA>      0.05      1
> 8     SL     SL     P3      <NA>      <NA>      <NA>      0.05      1
> 9     SL     SL     SL      <NA>      <NA>      <NA>      0.05      1
> 10    D      SL     P3         D         D         D         NA       1
> 11    V1     SL     P3         V1        D         D         NA       1
> 12    D      SL     SL         D         D         D         NA       1

```

```

> 13    V1    SL    SL        V1    D        D        NA        1
> 14    V2    SL    SL        V2    D        D        NA        1
>      convtype_t12
> 1          1
> 2          1
> 3          1
> 4          1
> 5          1
> 6          1
> 7          1
> 8          1
> 9          1
> 10         1
> 11         1
> 12         1
> 13         1
> 14         1

cypover2_fb <- overwrite(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "V1",
  "V2"),
  stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "D", "V1", "V2"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "D", "D", "D"),
  givenrate = c(0.03, 0.15, 0.1, 0.1, 0.1, 0.05, NA, NA, NA),
  type = c("S", "S", "S", "S", "S", "S", "S", "S", "S"))

cypover2_fb
>      stage3 stage2 stage1 eststage3 eststage2 eststage1 givenrate convtype
> 1      SD      SD  <NA>      <NA>      <NA>      <NA>      0.03        1
> 2      P1      SD  <NA>      <NA>      <NA>      <NA>      0.15        1
> 3      P2      P1  <NA>      <NA>      <NA>      <NA>      0.10        1
> 4      P3      P2  <NA>      <NA>      <NA>      <NA>      0.10        1
> 5      SL      P3  <NA>      <NA>      <NA>      <NA>      0.10        1
> 6      SL      SL  <NA>      <NA>      <NA>      <NA>      0.05        1
> 7       D      SL  <NA>          D          D          <NA>      NA         1
> 8      V1      SL  <NA>          V1          D          <NA>      NA         1
> 9      V2      SL  <NA>          V2          D          <NA>      NA         1
>      convtype_t12
> 1          1
> 2          1
> 3          1
> 4          1
> 5          1
> 6          1
> 7          1
> 8          1
> 9          1

```

Let's now also supply the overwrite tables and reproductive matrix for the raw MPM approach.

```

rep_cyp_raw <- matrix(0, 11, 11)
rep_cyp_raw[1:2,7:11] <- 0.5 * seeds_per_fruit

rep_cyp_raw

```

```

>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
> [1,]    0    0    0    0    0    0 2500 2500 2500 2500 2500
> [2,]    0    0    0    0    0    0 2500 2500 2500 2500 2500
> [3,]    0    0    0    0    0    0    0    0    0    0    0
> [4,]    0    0    0    0    0    0    0    0    0    0    0
> [5,]    0    0    0    0    0    0    0    0    0    0    0
> [6,]    0    0    0    0    0    0    0    0    0    0    0
> [7,]    0    0    0    0    0    0    0    0    0    0    0
> [8,]    0    0    0    0    0    0    0    0    0    0    0
> [9,]    0    0    0    0    0    0    0    0    0    0    0
> [10,]   0    0    0    0    0    0    0    0    0    0    0
> [11,]   0    0    0    0    0    0    0    0    0    0    0

```

```

cypover2_raw <- overwrite(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
  "XSm", "Sm"),
  stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm"),
  givenrate = c(0.03, 0.15, 0.1, 0.1, 0.1, 0.05, NA, NA, NA),
  type = c("S", "S", "S", "S", "S", "S", "S", "S", "S"))

```

```

cypover2_raw
>   stage3 stage2 stage1 eststage3 eststage2 eststage1 givenrate convtype
> 1     SD     SD  <NA>      <NA>      <NA>      <NA>      0.03      1
> 2     P1     SD  <NA>      <NA>      <NA>      <NA>      0.15      1
> 3     P2     P1  <NA>      <NA>      <NA>      <NA>      0.10      1
> 4     P3     P2  <NA>      <NA>      <NA>      <NA>      0.10      1
> 5     SL     P3  <NA>      <NA>      <NA>      <NA>      0.10      1
> 6     SL     SL  <NA>      <NA>      <NA>      <NA>      0.05      1
> 7       D     SL  <NA>          D        XSm      <NA>      NA        1
> 8     XSm     SL  <NA>      XSm      XSm      <NA>      NA        1
> 9       Sm     SL  <NA>          Sm      XSm      <NA>      NA        1
>   convtype_t12
> 1             1
> 2             1
> 3             1
> 4             1
> 5             1
> 6             1
> 7             1
> 8             1
> 9             1

```

```

cypover3_raw <- overwrite(stage3 = c("SD", "SD", "P1", "P1", "P2", "P3", "SL",
  "SL", "SL", "D", "D", "XSm"),
  stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL",
  "SL"),
  stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "SL", "P3",
  "SL", "SL"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "D", "D", "XSm"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm"),
  givenrate = c(0.01, 0.05, 0.10, 0.20, 0.1, 0.1, 0.05, 0.05, 0.05, NA, NA),

```

```

NA),
type = c("S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S"))

cypover3_raw
>   stage3 stage2 stage1 eststage3 eststage2 eststage1 givenrate convtype
> 1     SD     SD     SD     <NA>     <NA>     <NA>     0.01     1
> 2     SD     SD     rep     <NA>     <NA>     <NA>     0.05     1
> 3     P1     SD     SD     <NA>     <NA>     <NA>     0.10     1
> 4     P1     SD     rep     <NA>     <NA>     <NA>     0.20     1
> 5     P2     P1     SD     <NA>     <NA>     <NA>     0.10     1
> 6     P3     P2     P1     <NA>     <NA>     <NA>     0.10     1
> 7     SL     P3     P2     <NA>     <NA>     <NA>     0.05     1
> 8     SL     SL     P3     <NA>     <NA>     <NA>     0.05     1
> 9     SL     SL     SL     <NA>     <NA>     <NA>     0.05     1
> 10    D     SL     P3     D     XSm     XSm     NA     1
> 11    D     SL     SL     D     XSm     XSm     NA     1
> 12   XSm    SL     SL     XSm    XSm     XSm     NA     1
>   convtype_t12
> 1             1
> 2             1
> 3             1
> 4             1
> 5             1
> 6             1
> 7             1
> 8             1
> 9             1
> 10            1
> 11            1
> 12            1

```

The `overwrite()` approach is much less powerful than the supplement table approach, but some might prefer it.

Handling complex life histories with stage groups and supplement tables

Situations may occur in which the life history of the organism includes two sets of stages with transitions only occurring in one direction between them. For example, let's consider the following life history model.

Raw MPMs have no problem with this situation, provided that each stage has its own, unique set of characteristics. However, function-based MPMs (including IPMs) will have a default tendency to estimate survival transitions from the adult reproductive stages (in blue box) back to the pre-adult non-reproductive stages (in red box). We can deal with this problem using stage grouping.

First, we will create a new stageframe that encapsulates all of the stages above. Note that the primary addition that we will make to the `sf_create()` input is the `group` option.

```

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 3, 1, 3, 6, 11, 20, 31)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "pD", "pXSm", "pSm", "XSm", "Sm",
  "Md", "Lg", "XLg", "XXLg")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1)

```

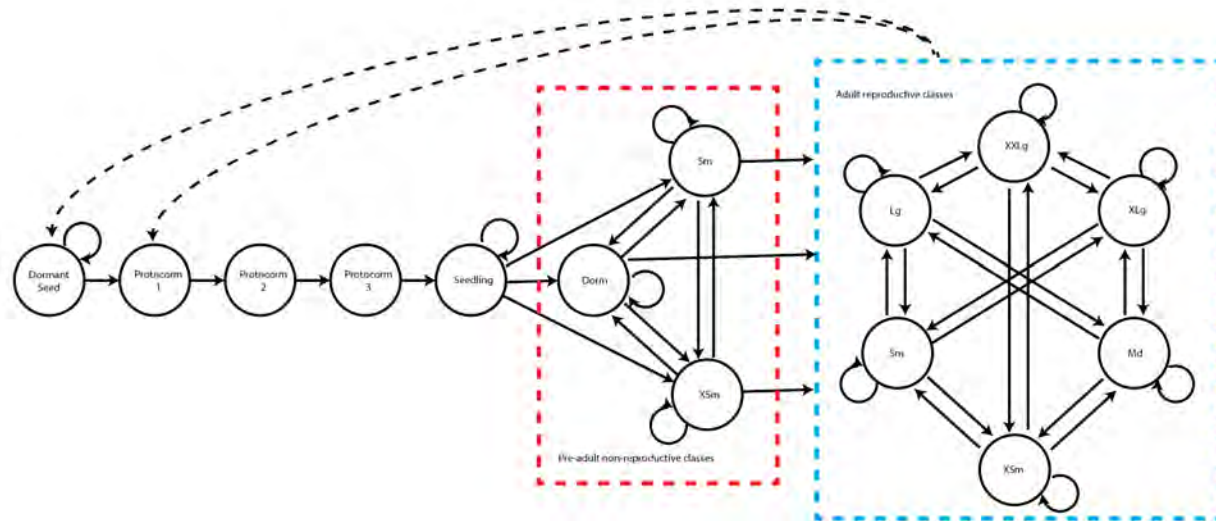


Figure 3.3: Imaginary life history for perennial plant species *X*, with a pre-adult non-reproductive set of 3 stages that eventually leads to a reproductive set of stages, from which the plant cannot return.

```
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1.5, 0.5, 1.5, 1.5, 3.5, 5.5, 5.5)
group <- c(0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 2, 2, 2, 2)

comments <- c("Dormant seed", "1st yr protocorm", "2nd yr protocorm",
              "3rd yr protocorm", "Seedling", "Dormant pre-adult",
              "Extra small pre-adult (1 shoot)", "Small pre-adult (2-4 shoots)",
              "Extra small adult (1 shoot)", "Small adult (2-4 shoots)",
              "Medium adult (5-7 shoots)", "Large adult (8-14 shoots)",
              "Extra large adult (20 shoots)", "Extra extra large adult (>24.5 shoots)")

cypframe_alt <- sf_create(sizes = sizevector, stagenames = stagevector,
                          repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
                          propstatus = propvector, immstatus = immvector, indataset = indataset,
                          binhalfwidth = binvec, group = group, comments = comments)

cypframe_alt
>   stage size size_b size_c min_age max_age repstatus obsstatus propstatus
> 1   SD     0    NA    NA      NA      NA         0         0         1
> 2   P1     0    NA    NA      NA      NA         0         0         0
> 3   P2     0    NA    NA      NA      NA         0         0         0
> 4   P3     0    NA    NA      NA      NA         0         0         0
> 5   SL     0    NA    NA      NA      NA         0         0         0
> 6   pD     0    NA    NA      NA      NA         0         0         0
> 7   pXSm   1    NA    NA      NA      NA         0         1         0
> 8   pSm    3    NA    NA      NA      NA         0         1         0
> 9   XSm    1    NA    NA      NA      NA         1         1         0
> 10  Sm     3    NA    NA      NA      NA         1         1         0
> 11  Md     6    NA    NA      NA      NA         1         1         0
> 12  Lg    11    NA    NA      NA      NA         1         1         0
> 13  XLg   20    NA    NA      NA      NA         1         1         0
> 14  XXLg  31    NA    NA      NA      NA         1         1         0
>   immstatus matstatus indataset binhalfwidth_raw sizebin_min sizebin_max
```

```

> 1      0      0      0      0.0      0.0      0.0
> 2      1      0      0      0.0      0.0      0.0
> 3      1      0      0      0.0      0.0      0.0
> 4      1      0      0      0.0      0.0      0.0
> 5      1      0      0      0.0      0.0      0.0
> 6      1      0      1      0.5      -0.5      0.5
> 7      1      0      1      0.5      0.5      1.5
> 8      1      0      1      1.5      1.5      4.5
> 9      0      1      1      0.5      0.5      1.5
> 10     0      1      1      1.5      1.5      4.5
> 11     0      1      1      1.5      4.5      7.5
> 12     0      1      1      3.5      7.5      14.5
> 13     0      1      1      5.5      14.5      25.5
> 14     0      1      1      5.5      25.5      36.5
>      sizebin_center sizebin_width binhalfwidthb_raw sizebinb_min sizebinb_max
> 1              0              0              NA              NA              NA
> 2              0              0              NA              NA              NA
> 3              0              0              NA              NA              NA
> 4              0              0              NA              NA              NA
> 5              0              0              NA              NA              NA
> 6              0              1              NA              NA              NA
> 7              1              1              NA              NA              NA
> 8              3              3              NA              NA              NA
> 9              1              1              NA              NA              NA
> 10             3              3              NA              NA              NA
> 11             6              3              NA              NA              NA
> 12            11              7              NA              NA              NA
> 13            20             11              NA              NA              NA
> 14            31             11              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
> 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
> 12             NA              NA              NA              NA              NA
> 13             NA              NA              NA              NA              NA
> 14             NA              NA              NA              NA              NA
>      sizebinc_center sizebinc_width group      comments
> 1              NA              NA      0      Dormant seed
> 2              NA              NA      0      1st yr protocorm
> 3              NA              NA      0      2nd yr protocorm
> 4              NA              NA      0      3rd yr protocorm
> 5              NA              NA      0      Seedling
> 6              NA              NA      1      Dormant pre-adult
> 7              NA              NA      1      Extra small pre-adult (1 shoot)
> 8              NA              NA      1      Small pre-adult (2-4 shoots)

```



```

> 9          NA          NA      2      Extra small adult (1 shoot)
> 10         NA          NA      2      Small adult (2-4 shoots)
> 11         NA          NA      2      Medium adult (5-7 shoots)
> 12         NA          NA      2      Large adult (8-14 shoots)
> 13         NA          NA      2      Extra large adult (20 shoots)
> 14         NA          NA      2      Extra extra large adult (>24.5 shoots)

```

Now that we have our stageframe, the only thing we need to do is to specify that survival-transitions from group 2 stages to group 1 stages are equal to 0. We can do this with the `supplemental()` function. Here, we will set these transitions, and also set the fecundity multipliers as before.

```

cypsupp2_alt <- supplemental(stage3 = c("group1", "SD", "P1"),
  stage2 = c("group2", "rep", "rep"), eststage3 = c(NA, NA, NA),
  eststage2 = c(NA, NA, NA), givenrate = c(0, NA, NA),
  multiplier = c(NA, (0.5 * seeds_per_fruit), (0.5 * seeds_per_fruit)),
  type = c(1, 3, 3), stageframe = cypframe_alt, historical = FALSE)

```

```

cypsupp2_alt
>   stage3 stage2 stage1 eststage3 eststage2 eststage1 givenrate multiplier
> 1 group1 group2 <NA>          NA          NA          <NA>          0           1
> 2      SD   rep <NA>          NA          NA          <NA>          NA          2500
> 3      P1   rep <NA>          NA          NA          <NA>          NA          2500
>   convtype convtype_t12
> 1         1           1
> 2         3           1
> 3         3           1

```

This supplement table can now be input into a matrix estimator function to tell that function that these transitions must be 0. We will see how to do this with different styles of MPM in the next four chapters.

We can now proceed in the next chapter to create raw matrices.

Literature cited

Shefferson, R.P., Kurokawa, S. & Ehrlén, J. (2021). *lefko3*: Analysing individual history through size-classified matrix population models. *Methods in Ecology and Evolution*, 12, 378–382.