fishR Vignette - Age-Length Keys to Assign Age from Lengths

Dr. Derek Ogle, Northland College December 16, 2013

The assessment of ages for a large number of fish is very time-consuming, whereas measuring the length of a large number of fish is relatively easy. The age structure for a large number of fish can be estimated by summarizing the relationship between age and length for a relatively small subsample of fish and then applying this summary to the entire sample of fish. This summary is called an *age-length key*. The construction and application of an age-length key and the use of this key to assign ages to individual fish as described in Isermann and Knight (2005) is the focus of this vignette.

Section 1 contains background information for age-length keys in general and the use of the methods described in Isermann and Knight (2005) for assigning ages to un-aged fish. Section 2 shows how to construct an age-length key and Section 3 shows how to use this age-length key to assign ages to individual fish. Section 4 contains additional thoughts on age-length key methods.

This vignette requires functions from the `FSA` and `FSAdata` packages maintained by the author. These packages are loaded into R with

```
> library(FSA)
> library(FSAdata)  # for datafile
> library(plotrix)  # for histStack
```

All analyses in this document use the Lake Ontario Rock Bass (*Ambloplites rupestris*) data set from Wolfert (1980) which is loaded into R and the first and last six rows are viewed with

```
> data(RockBassLO2)
> head(RockBassLO2)  # note that ages are assigned
  age  tl
1   6 218
2   5 184
3   7 211
4   9 223
5   9 245
6   7 181
> tail(RockBassLO2)  # note that ages are not assigned
     age  tl
1283  NA 218
1284  NA 157
1285  NA 200
1286  NA 173
1287  NA 214
1288  NA 194
```

1 Background

1.1 Constructing Age-Length Keys

A subsample of n fish to be aged is selected from the entire sample of fish by randomly selecting fish *from each length interval* (rather than a simple random selection of all fish). The number of fish in each interval can either be fixed at a constant number or proportional to the total number of fish in that length interval. Kimura (1977) notes that proportional selection produces statistically "better" results. However,

proportional selection is uncommon in the field as the length of all fish must be known before taking the subsample. Thus, it is much more common, in practice, to subsample a set number of fish per length interval. More importantly, the range of lengths in the subsample must cover the entire range of lengths in the original sample. This subsample of fish is called the *age sample*, because the relationship between age and length will be determined from this group of fish only.

The measured length and assessed age is recorded for all fish in the age sample. In addition, the length category to which each fish belongs is also recorded. For example, if 5-mm length categories are created that begin on the "0" and "5" units, then a 117 mm fish will be recorded as being in the 115-119 mm length category. Generally, all length categories are of the same width; thus, for simplicity, only the beginning length in the length category is recorded (e.g., "115" mm).

The assigned length category and the assessed age for the fish in the age sample is summarized with a two-way contingency table where the length categories form the rows and age categories form the columns. In other words, the number of fish in the age sample in each length category and age combination is determined. An example of this summary is shown below (with row and column labels to aid interpretation).

	A_1	A_2
L_1	6	2
L_2	3	3
L_3	1	4

In this example, six fish in the age sample belong to the first length category and the first age. Similarly, one fish belongs to the last length category and the first age.

The portion of the entire sample of fish that was not part of the age sample (i.e., fish for which only a length measurement was obtained) is called the *length sample*. Age is "assigned" to fish in the length sample based on the fish's length category and the proportion of fish of that length category of each age as determined by the fish in the age sample. For example, a fish in the length sample that belongs to the first length category has, based on the summary from the age sample, a 75% chance (i.e., $\frac{6}{8}$) of being of the first age and a 25% chance of being of the second age. Furthermore, if there were 20 fish in the length sample in the first length category, then we would expect 15 (i.e., $20 * 0.75$) to be of the first age and five (i.e., $20 * 0.25$) to be of the second age.

Thus, assignment of ages to fish in the length sample is based on the "probability" of each age given the length category that the fish belongs to, as derived from the age-sample. The required conditional probabilities from the age sample are derived from the summary contingency table by dividing each cell of the table by the total number of fish in that length cateogry (i.e., each cell is divided by the sum of its row; thus, a "row-proportions" table is computed). The row-proportions table derived from the summary contingency table shown above is shown below.

	A_1	A_2
L_1	0.75	0.25
L_2	0.50	0.50
L_3	0.20	0.80

This row-proportions table is the so-called *age-length key*, because it relates the conditional probability of an age given a particular length category.

1.2 Age Assignment to Individuals

An age-length key can be used to develop a summary frequency of ages for the length sample. In addition, methods have been developed to find the mean length-at-age for the length sample or the CPE for each age from fish in the length sample (Bettoli and Miranda 2001). However, these methods cannot be used to calculate measures of variability for these summaries (Isermann and Knight 2005). Thus, it is beneficial

to use the age-length key to assign an age to each fish in the length sample and then to summarize those results (Isermann and Knight 2005). There are two methods for using the age-length key to assign ages to specific fish in the length sample: semi-random and completely random methods. The semi-random method is discussed in this section and the completely random method is shown in Section 4.1.

In the semi-random method of assigning ages to individual fish, the exact expected number of fish with a given length interval of a given age will be assigned that age, with exceptions for fractionality discussed below. For example, given the age-length key from above, assume that there are 24 fish in the length sample that are in the first length category. In this case, 75% or eighteen (i.e., $24 * 0.75$) would be assigned the first age and six (i.e., $24 * 0.25$) would be assigned the second age. Which of the 24 fish would be assigned the specific ages is determined randomly, but the number to be assigned each age is set at eighteen and six, respectively.

One difficulty that is soon apparent is that the expected number of individuals in a given length interval assigned a certain age may contain a fraction. For example, supposed that 23 fish in the length sample are in the second length category. In this case, the expected number of fish for each age is 11.5 (i.e., $23 * 0.5$ for each age). Clearly, a half of a fish cannot be assigned to a given age category. This difficulty was called *fractionality* by Isermann and Knight (2005).

Fractionality is handled by first assigning the closest integer ***smaller*** then the expected number of fish to the age-group. In the example above, eleven fish would be assigned each of the two ages. The remaining fish is then randomly assigned an age with a probabilistic weight equal to the expected proportion of fish in each age category. For example, the fish would be assigned the first age with a probability of 0.50 and would be assigned the second age with a probability of 0.50. In other words, in this example, it is essentially a coin-flip which age the "extra" fish is assigned. If the coin flip determines that the "extra" fish should be in the second age, then eleven fish would be assigned the first age and twelve would be assigned the second age. Again, the specific age to be assigned to each individual fish is determined randomly; but eleven fish would be assigned the first age and twelve would be assigned the second age as determined above.

To complete the example, suppose that fourteen fish in the length sample are in the third length category. Given the age-length key above, two fish will be assigned the first age (i.e., $14 * 0.2$) and eleven fish will be assigned the second age (i.e., $14 * 0.8$). The "extra" fish (i.e., $14 - 2 - 11$) will be assigned an age with a probability of being the first age of 0.2 and a probability of being the second age of 0.8. Thus, it is four times more likely that the "extra" fish will be assigned the second age than the first age.

2 Construction of an Age-Length Key in R

In most situations, one data file will contain the lengths of all fish in the sample with corresponding ages only for those fish in the age sample. The ages for the fish not in the age sample will have `NA` in their place which is how R denotes "missing data" (i.e., `NA` stands for "not available"). With this type of data file, the first step in constructing and applying the age-length key is to separate the data file into the age sample of fish with assigned ages and the length sample of fish without assigned ages. This separation requires `Subset()` and `is.na()`. The `Subset()` function, which requires two arguments, creates a new data frame from an existing data frame based on some condition. The first argument is the original data frame from which a subset or part of the data frame should be returned. The second argument is a conditional statement on how that subset should be determined. The `is.na()` function is used to identify positions in a vector where "NA"s occur. This function is used to create the condition for separating the original data frame into age and length samples (i.e., fish with "NA"s in the age variable will be in the length sample, all others will be in the age sample). For example, the age and length samples are constructed for the Lake Ontario Rock Bass data with

```
> rb.age <- Subset(RockBassLO2,!is.na(age))      # get fish without NAs in age variable
> str(rb.age)
'data.frame': 135 obs. of  2 variables:
 $ age: int  6 5 7 9 9 7 8 4 6 8 ...
```

```
 $ tl : int  218 184 211 223 245 181 207 173 201 246 ...
> rb.len <- Subset(RockBassLO2,is.na(age))       # get fish with NAs in age variable
> str(rb.len)
'data.frame': 1153 obs. of  2 variables:
 $ age: int  NA NA NA NA NA NA NA NA NA NA ...
 $ tl : int  172 173 175 171 173 184 203 218 222 185 ...
```

From this, note that the age sample consists of 135 fish with assigned ages and the length sample consists of 1153 fish without assigned ages.

The next step in constructing the age-length key is to create a variable in the age sample that identifies the length category to which each fish belongs. This variable is constructed, with default name *LCat*, and appended to the data frame containing the age-sample with `lencat()`. In this context, `lencat()` requires the following four arguments:

- a formula of the form `~len` where `len` generically represents the variable in the data frame containing the lengths to be categorized,
- the data frame containing the age-sample in `data=`,
- a numeric value identifying the starting length measurement category in `startcat=`, and
- a numeric value identifying the width of the length measurement categories in `w`.

The `lencat()` function returns a data frame that consists of the original data frame plus a variable containing the length interval categories for each fish. The default name of the new variable (*LCat*) can be changed with the `vname=` argument. The `lencat()` function result must be assigned to an object, preferably named differently from the original age sample.

The starting category for 10-mm length categories for the Lake Ontario Rock Bass is determined by finding the minimum length in the age sample with

```
> Summarize(~tl,data=rb.age,digits=1)
      n     mean       sd      min       Q1   median       Q3      max percZero
  135.0    200.7     40.3    111.0    168.0    201.0    234.0    278.0      0.0
```

and then starting the categories with the even-number 10-mm interval just below this value. Construction of the length category variable is completed and the first six rows are viewed with[1]

```
> rb.age1 <- lencat(~tl,data=rb.age,startcat=110,w=10)
> head(rb.age1)
  age  tl LCat
1   6 218  210
2   5 184  180
3   7 211  210
4   9 223  220
5   9 245  240
6   7 181  180
```

Once the length category variable has been added to the age sample data frame, `table()` is used to construct the summary contingency table of numbers of fish in each combined length and age category. The row variable (length category) is the first and the column variable (age) is the second argument to this function. The results of `table()` should be assigned to an object and then submitted as the first argument to `prop.table()`

[1] Notice the new name of the augmented data frame.

along with `margin=1` as a second argument[2] to construct a row-proportions table. The resulting row-proportions table is the actual age-length key determined from the age sample and is ready to be applied to the length sample. The summary contingency table and the row-proportion table (i.e., the age-length key) are constructed with

```
> rb.raw <- with(rb.age1,table(LCat,age))
> rb.key <- prop.table(rb.raw,margin=1)
> round(rb.key,2)      # rounded for display purposes only
     age
LCat    3    4    5    6    7    8    9   10   11
  110 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
  120 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
  130 0.50 0.50 0.00 0.00 0.00 0.00 0.00 0.00 0.00
  140 0.10 0.90 0.00 0.00 0.00 0.00 0.00 0.00 0.00
  150 0.00 0.80 0.20 0.00 0.00 0.00 0.00 0.00 0.00
  160 0.00 0.70 0.20 0.10 0.00 0.00 0.00 0.00 0.00
  170 0.00 0.50 0.30 0.20 0.00 0.00 0.00 0.00 0.00
  180 0.00 0.40 0.40 0.10 0.10 0.00 0.00 0.00 0.00
  190 0.00 0.20 0.30 0.20 0.30 0.00 0.00 0.00 0.00
  200 0.00 0.10 0.20 0.20 0.40 0.10 0.00 0.00 0.00
  210 0.00 0.00 0.10 0.30 0.50 0.00 0.10 0.00 0.00
  220 0.00 0.00 0.00 0.30 0.30 0.20 0.20 0.00 0.00
  230 0.00 0.00 0.00 0.00 0.40 0.10 0.20 0.20 0.10
  240 0.00 0.00 0.00 0.00 0.10 0.50 0.20 0.20 0.00
  250 0.00 0.00 0.00 0.00 0.30 0.20 0.20 0.20 0.10
  260 0.00 0.00 0.00 0.00 0.00 0.25 0.50 0.25 0.00
  270 0.00 0.00 0.00 0.00 0.00 0.20 0.00 0.80 0.00
```

3 Assigning Ages to Individuals with the Age-Length Key in R

3.1 Age Assignment

The semi-random age assignment method described in Section 1.2 is implemented with `ageKey()`. This function requires the numeric matrix containing the age-length key (e.g., as constructed with `prop.table()`) as the first argument, a formula of the form `~len` or `age~len` as the second argument, and the name of the age sample data frame in `data=`. If the age-sample data frame contains a variable to receive the ages then use `age~len`. Alternatively, if the age-sample data frame does not contain a variable to receive the ages then use `~len` and the function will create a new variable called *age* by default.

The `ageKey()` function will determine the length categories to construct based on the age-length key sent in the first argument. The `ageKey()` function returns a data frame that has the same number of rows as the original length sample, but with each fish assigned an age according to the semi-random method. The results of `ageKey()` should be assigned to an object, preferably with a name different from the original length sample. For example, semi-random ages were assigned to the un-aged fish in the length sample with

```
> rb.len1 <- ageKey(rb.key,age~tl,data=rb.len)
> head(rb.len1)
    age  tl
136   5 172
137   6 173
138   6 175
```

[2]The `margin=1` indicates "rows", whereas as `margin=2` would indicate "columns."

```
139   4 171
140   5 173
141   5 184
```

The original (not modified) age sample data frame and the modified length sample data frame (i.e., now containing the ages assigned via the age-length key) can then be row-bound together to construct a data frame that consists of lengths and ages for all fish in the original sample. For example, the original age sample and the modified length sample for the Rock Bass sample are combined to form a complete data frame with

```
> rb.comb <- rbind(rb.age,rb.len1)
> head(rb.comb)
  age  tl
1   6 218
2   5 184
3   7 211
4   9 223
5   9 245
6   7 181
```

3.2 Summary Computations

The assigned ages in the `rb.comb` data frame can then be summarized with[3]

```
> ( rb.sum <- Summarize(tl~age,data=rb.comb,digits=2) )
Warning:  To continue, variable(s) on RHS of formula were converted to a factor.
  age   n  mean    sd min  Q1 median  Q3 max percZero
1   3   5 129.4 12.28 111 125    131 137 143        0
2   4 320 175.7 15.59 130 165    175 187 208        0
3   5 263 187.3 14.13 153 179    187 197 218        0
4   6 215 200.2 17.25 161 188    202 214 228        0
5   7 312 209.6 15.93 180 198    208 217 258        0
6   8  74 226.7 17.89 200 207    224 241 270        0
7   9  63 229.0 13.28 210 220    226 235 265        0
8  10  30 245.0 14.70 230 234    240 251 278        0
9  11   6 237.8 10.03 231 233    234 236 258        0
```

In this summary, note that the *age* and *n* columns represent an age-frequency and the *age* and *mean* columns represent the mean length-at-age for ALL individuals in the entire sample. A plot of the age frequency (Figure 1) is constructed with[4]

```
> hist(~age,data=rb.comb,breaks=3:11,xlim=c(2,12),xlab="Age (yrs)",main="")
```

A plot (Figure 2) of the length-at-age for ALL individuals in the sample[5] with connected mean lengths-at-age superimposed[6] can be constructed with

[3] The "extra" parenthese around this command force R to print the result at the same time it is being assigned to an object.

[4] Note that my figures may look different than yours as I have changed the default settings for the graphics with `par()`. In particular, I used `par(mar=c(3.5,3.5,1,1),mgp=c(2.1,0.4,0),tcl=-0.2)` for most graphics. See the "Basic Plotting in R" vignette.

[5] The `jitter()` function adds a small amount of random noise go the discrete age values so that multiple fish with the same length and age are not plotted directly on top of each other. This makes it easier to visualize the density of fish that have similar age and length values.

[6] The *age* variable in the results from `Summarize()` is a factor variable. To make this plot, the levels must be converted to numeric values with `fact2num()`.

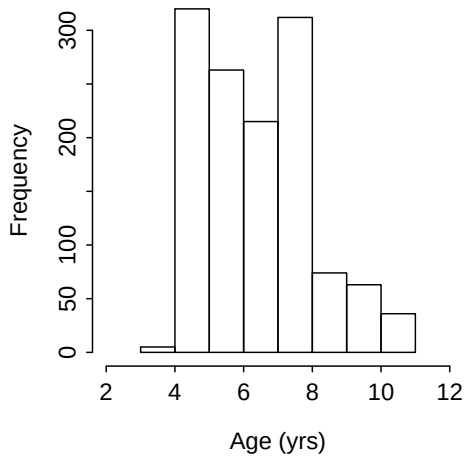

Figure 1. Age frequency for Lake Ontario Rock Bass.

```
> plot(tl~jitter(age),data=rb.comb,ylab="Total Length (mm)",xlab="Age (jittered)")
> lines(mean~fact2num(age),data=rb.sum,col="blue",lwd=2)
```

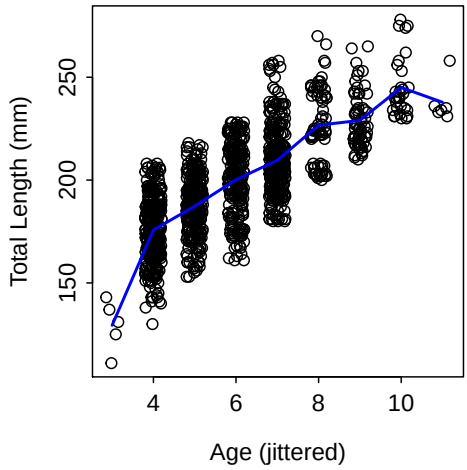

Figure 2. Length-at-age for Lake Ontario Rock Bass with mean lengths-at-age connected by the blue line.

Finally, one can produce a visual summary of the combined age and length frequencies with `histStack()` from the `plotrix` package. This function requires a formula of the form `quant~factor` where `quant` generically represents a quantitative variable that will form the x-axis of the histogram and `factor` generically represents a categorical variable that will form the "stacks" in the bars. The data frame containing these two variables must be included in `data=`. The sequence of breaks to be used in the histogram can be sent to `breaks=`. The user may also control the x-axis label with `xlab=` and may change the color palette with `pal=` (to, for example, `"gray"` to form a gray-scale plot). An example (Figure 3), for the Rock Bass data was created with

```
> histStack(tl~age,data=rb.comb,breaks=seq(110,280,10),xlab="Total Length (mm)")
Warning:  z was converted to a factor
```

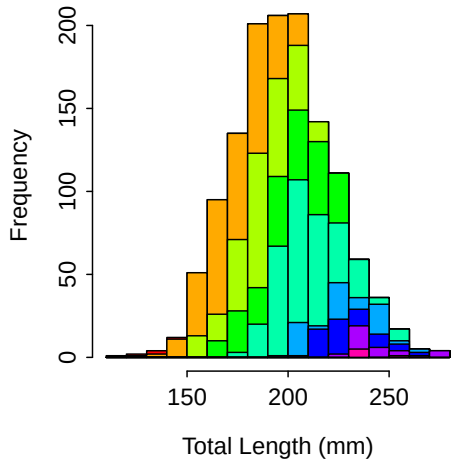

Figure 3. Length frequency for Lake Ontario Rock Bass with different colors for different ages within each length category.

4 Further Thoughts

4.1 Completely Random Age Assignment with Age-Length Key

In the completely random method of assigning ages to individual fish, the age assigned to a fish is randomly assigned with probabilistic weights proportional to the proportion of fish in each age for fish on the age-length key of the same length category. This is essentially the same rule used to assign age to the "exra" fish (from fractionality) in the semi-random method. For example, suppose that 50% of the fish in a length interval in the age-sample were age-1, 25% were age-2, and 25% were age-3. A single fish in the same length interval in the length sample would then have a 50% chance of being assigned an age of 1, a 25% chance of being assigned an age of 2, and a 25% chance of being assigned an age of 3. The specific age it is assigned depends completely on randomness. Thus, in this example, we would expect 11 of 22 fish in this length category in the length-sample to be assigned an age of 1 but in actuality it could be more or it could be less. In the semi-random method, which fish get which age is random but the number of fish that get each age is (primarily) not random. In the completely random method, both the number of fish that get each age and which fish get each age is random.

The completely random method of assigning ages to individual fish is also implemented with `ageKey()`. The arguments to the function are exactly the same as they were for the semi-random method except that a `type="CR"` argument must be included. Use of the completely random method of assigning ages and summary results are computed with

```
> rb.len2 <- ageKey(rb.key,age~tl,data=rb.len,type="CR")
> rb.comb2 <- rbind(rb.age,rb.len2)
> Summarize(tl~age,data=rb.comb2,digits=2)
Warning:  To continue, variable(s) on RHS of formula were converted to a factor.
  age   n  mean    sd min  Q1 median  Q3 max percZero
1   3   5 129.4 12.28 111 125    131 137 143        0
2   4 317 175.3 15.31 130 164    175 186 208        0
3   5 258 187.3 14.38 150 178    187 197 217        0
4   6 197 198.9 17.22 161 187    200 212 228        0
5   7 338 209.6 15.18 181 198    208 217 257        0
6   8  80 227.2 17.56 200 217    226 241 270        0
7   9  59 229.0 13.36 210 221    225 235 265        0
8  10  28 246.4 14.99 230 235    244 254 278        0
```

```
9  11   6 237.7 10.46 230 231    235 238 258        0
```

The semi-random method of assigning ages to individual fish is the preferred method, especially when the age-length key is used to extrapolate the ages of fish in a single length sample. The completely random method is used primarily when studying the theoretical sources of variability inherent in an age-length key analysis. In other words, the semi-random method is the method most often used by the practicing biologist.

References

Bettoli, P. W. and L. E. Miranda. 2001. A cautionary note about estimating mean length at age with subsampled data. North American Journal of Fisheries Management 21:425–428. 2

Isermann, D. A. and C. T. Knight. 2005. A computer program for agelength keys incorporating age assignment to individual fish. North American Journal of Fisheries Management 25:1153–1160. 1, 2, 3

Kimura, D. A. 1977. Statistical assessment of the age-length key. Journal of the Fisheries Research Board of Canada 34:317–324. 1

Wolfert, D. R. 1980. Age and growth of rock bass in eastern Lake Ontario. New York Fish and Game Journal 27:88–90. 1

Reproducibility Information

Version Information

- **Compiled Date:** Mon Dec 16 2013
- **Compiled Time:** 9:45:38 PM
- **Code Execution Time:** 2.46 s

R Information

- **R Version:** R version 3.0.2 (2013-09-25)
- **System:** Windows, i386-w64-mingw32/i386 (32-bit)
- **Base Packages:** base, datasets, graphics, grDevices, methods, stats, utils
- **Other Packages:** FSA_0.4.3, FSAdata_0.1.4, gdata_2.13.2, knitr_1.5.15, plotrix_3.5-2
- **Loaded-Only Packages:** bitops_1.0-6, car_2.0-19, caTools_1.16, cluster_1.14.4, evaluate_0.5.1, formatR_0.10, Formula_1.1-1, gplots_2.12.1, grid_3.0.2, gtools_3.1.1, highr_0.3, Hmisc_3.13-0, KernSmooth_2.23-10, lattice_0.20-24, MASS_7.3-29, multcomp_1.3-1, mvtnorm_0.9-9996, nlme_3.1-113, nnet_7.3-7, quantreg_5.05, sandwich_2.3-0, sciplot_1.1-0, SparseM_1.03, splines_3.0.2, stringr_0.6.2, survival_2.37-4, tools_3.0.2, zoo_1.7-10
- **Required Packages:** FSA, FSAdata, plotrix and their dependencies (car, gdata, gplots, Hmisc, knitr, multcomp, nlme, quantreg, sciplot)