Bootstrapping Annual Biomass Estimates by Taxa

Jeremy Jennings

11/22/2020

Overview

This is a step-by-step review of the draft R code to take raw biomass estimates from Wisseman et al., sample (with replacement) from each set of seasonal replicates, and calculate a bootstrap distribution of annual means by taxon. The code can be found at: https://github.com/JenningsJC/S.Fk.McKenzie_FoodWeb_Study.

1. Read in dummy data: note that each combination of taxon and season's replicates has a unique, but identical, value to aid in evaluating the output of the code at each step.

```
##     site     taxon season replicate biomass
## 1  alpha    mayfly winter         1       1
## 2  alpha    mayfly winter         2       1
## 3  alpha    mayfly winter         3       1
## 4  alpha    mayfly winter         4       1
## 5  alpha    mayfly winter         5       1
## 6  alpha    mayfly spring         1       2
## 7  alpha    mayfly spring         2       2
## 8  alpha    mayfly spring         3       2
## 9  alpha    mayfly spring         4       2
## 10 alpha    mayfly spring         5       2
## 11 alpha    mayfly summer         1       3
## 12 alpha    mayfly summer         2       3
## 13 alpha    mayfly summer         3       3
## 14 alpha    mayfly summer         4       3
## 15 alpha    mayfly summer         5       3
## 16 alpha    mayfly   fall         1       4
## 17 alpha    mayfly   fall         2       4
## 18 alpha    mayfly   fall         3       4
## 19 alpha    mayfly   fall         4       4
## 20 alpha    mayfly   fall         5       4
## 21 alpha caddislfy winter         1       5
## 22 alpha caddislfy winter         2       5
## 23 alpha caddislfy winter         3       5
## 24 alpha caddislfy winter         4       5
## 25 alpha caddislfy winter         5       5
## 26 alpha caddislfy spring         1       6
## 27 alpha caddislfy spring         2       6
## 28 alpha caddislfy spring         3       6
## 29 alpha caddislfy spring         4       6
## 30 alpha caddislfy spring         5       6
## 31 alpha caddislfy summer         1       7
## 32 alpha caddislfy summer         2       7
## 33 alpha caddislfy summer         3       7
## 34 alpha caddislfy summer         4       7
## 35 alpha caddislfy summer         5       7
## 36 alpha caddislfy   fall         1       8
## 37 alpha caddislfy   fall         2       8
## 38 alpha caddislfy   fall         3       8
## 39 alpha caddislfy   fall         4       8
## 40 alpha caddislfy   fall         5       8
## 41 alpha  stonefly winter         1       9
## 42 alpha  stonefly winter         2       9
## 43 alpha  stonefly winter         3       9
## 44 alpha  stonefly winter         4       9
## 45 alpha  stonefly winter         5       9
## 46 alpha  stonefly spring         1      10
## 47 alpha  stonefly spring         2      10
## 48 alpha  stonefly spring         3      10
## 49 alpha  stonefly spring         4      10
## 50 alpha  stonefly spring         5      10
## 51 alpha  stonefly summer         1      11
## 52 alpha  stonefly summer         2      11
## 53 alpha  stonefly summer         3      11
## 54 alpha  stonefly summer         4      11
## 55 alpha  stonefly summer         5      11
## 56 alpha  stonefly   fall         1      12
## 57 alpha  stonefly   fall         2      12
## 58 alpha  stonefly   fall         3      12
## 59 alpha  stonefly   fall         4      12
## 60 alpha  stonefly   fall         5      12
```

2. The R function stratified()

- stratified() takes a stratified random sample from the raw dataset, as specified by the vector of columns given to it (taxon, season, biomass), of the defined size (here it is 1 value of replicate chosen randomly from the set of replicates) and outputs the associated biomasses in an array

```
library(splitstackshape)
random_sample <- stratified(dummy_benth_clean,
                              c("taxon", "season", "biomass"),
                              1 ,
                              replace = TRUE)
print(random_sample)
```

```
##      site     taxon season replicate biomass
##  1: alpha    mayfly winter         4       1
##  2: alpha    mayfly spring         1       2
##  3: alpha    mayfly summer         3       3
##  4: alpha    mayfly   fall         3       4
##  5: alpha caddislfy winter         4       5
##  6: alpha caddislfy spring         3       6
##  7: alpha caddislfy summer         2       7
##  8: alpha caddislfy   fall         2       8
##  9: alpha  stonefly winter         4       9
## 10: alpha  stonefly spring         5      10
## 11: alpha  stonefly summer         5      11
## 12: alpha  stonefly   fall         2      12
```

3. The R function tapply()

- tapply() calculates the mean of the specified column of values (biomass) in the array of biomass values that was output from the stratified() function

```
means <- tapply(random_sample$biomass,
                  list(random_sample$taxon),
                  mean)
print(means)
```

```
## caddislfy    mayfly  stonefly
##       6.5       2.5      10.5
```

4. Looping the functions across the raw data frame

- placing both functions inside a “for loop” allows the user to iterate these operations “n” number of times (“for i in 1:n”)

```
biomass_list <- list()
means_list <- list()
for (i in 1:5) {
  random_sample <- stratified(dummy_benth_clean,
                              c("taxon", "season", "biomass"),
                              1 ,
                              replace = TRUE)
  biomass_list[[i]] <- random_sample

  means <- tapply(random_sample$biomass,
                  list(random_sample$taxon),
                  mean)
  means_list[[i]] <- means
}
## the below code takes the list of means output from the loop and puts
## them together into one table
annual_means <- do.call(rbind, means_list)

## the below code coerces that table into a data frame
annual_benth_means <- as.data.frame(annual_means)
```

- once assembled into a data frame, the set of annual means generated by the loop looks like the below table. Because of the values assigned in the dummy dataset, all means for caddisfly should be 6.5, for mayfly, 2.5, and stonefly 10.5:

```
print(annual_benth_means)
```

```
##   caddislfy mayfly stonefly
## 1       6.5    2.5     10.5
## 2       6.5    2.5     10.5
## 3       6.5    2.5     10.5
## 4       6.5    2.5     10.5
## 5       6.5    2.5     10.5
```

- to inspect the resampled set of biomasses and double-check that the loop is working correctly and the replicate number, season and associated biomass is consistent with the dummy data:

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(purrr)
bio_boot_samples <- biomass_list %>%
  reduce(left_join, by = c("site", "taxon", "season"))
print(bio_boot_samples)
```

```
##      site     taxon season replicate.x biomass.x replicate.y biomass.y
##  1: alpha    mayfly winter           1         1           1         1
##  2: alpha    mayfly spring           5         2           4         2
##  3: alpha    mayfly summer           1         3           1         3
##  4: alpha    mayfly   fall           1         4           3         4
##  5: alpha caddislfy winter           2         5           2         5
##  6: alpha caddislfy spring           5         6           2         6
##  7: alpha caddislfy summer           1         7           4         7
##  8: alpha caddislfy   fall           4         8           4         8
##  9: alpha  stonefly winter           5         9           1         9
## 10: alpha  stonefly spring           5        10           2        10
## 11: alpha  stonefly summer           1        11           3        11
## 12: alpha  stonefly   fall           1        12           4        12
##     replicate.x.x biomass.x.x replicate.y.y biomass.y.y replicate biomass
##  1:             1           1             2           1         4       1
##  2:             3           2             5           2         4       2
##  3:             2           3             1           3         2       3
##  4:             5           4             5           4         1       4
##  5:             2           5             1           5         2       5
##  6:             1           6             3           6         1       6
##  7:             2           7             3           7         4       7
##  8:             4           8             1           8         2       8
##  9:             5           9             1           9         5       9
## 10:             2          10             4          10         3      10
## 11:             1          11             4          11         5      11
## 12:             2          12             5          12         2      12
```