

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:

```
##   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
```

2. Apply the R functions stratified() and tapply() to the raw data in a “for loop”, store the outputs in lists, turn the lists into tables

- 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         2         2
## 3: alpha   mayfly summer         2         3
## 4: alpha   mayfly  fall         2         4
## 5: alpha caddisfly winter         4         5
## 6: alpha caddisfly spring         4         6
## 7: alpha caddisfly summer         5         7
## 8: alpha caddisfly  fall         1         8
## 9: alpha stonefly winter         5         9
## 10: alpha stonefly spring         4        10
## 11: alpha stonefly summer         2        11
## 12: alpha stonefly  fall         3        12
```

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

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

```
## function (x, ...)
## UseMethod("mean")
## <bytecode: 0x000000001563e2e0>
## <environment: namespace:base>
```

- 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 the table into a data frame
annual_benth_means <- as.data.frame(annual_means)
```

- the assembled into a data frame of annual mean estimates looks like:

```
print(annual_benth_means)

##   caddisfly 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 check that the loop is working correctly:

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