

Julia Vineyard

Worked with: Andrew Gordon, Jessica Bonin, Aliza Fassler

Stats Lab 7: The Bootstrap

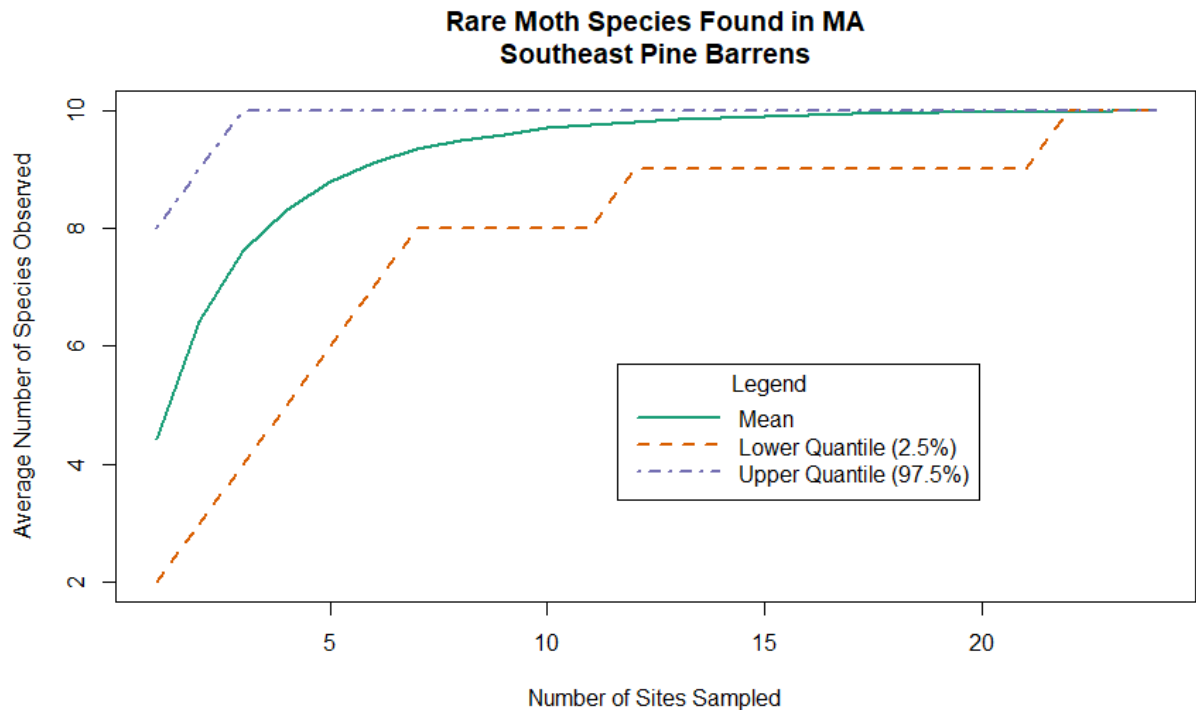
1. `n=123`
`require(palmerpenguins)`
`dat_gentoo_all= subset(penguins, species == "Gentoo", na.rm=TRUE)`
`dat_gentoo_bill_length= na.omit(dat_gentoo_all$bill_length_mm)`
`length(dat_gentoo_bill_length)`
2. `3.081857`
`sd(dat_gentoo_bill_length)`
3. `± 1.9796`
`qt(c(0.025, 0.975), 122)`
4. `0.2778817`
`sse_mean= function(x)`
`{sse= sd(x, na.rm=TRUE)/(sqrt(length(x[! is.na (x)])))`
`return(sse)`
`}`
`sse_mean(dat_gentoo_bill_length)`
5. `47.50488 ± 0.04960036 = 47.45528 47.55448`
`mean(dat_gentoo_bill_length) +`
`qt(c(0.025, 0.975), 122) * (sse_mean(dat_gentoo_bill_length)/sqrt(123))`
6. `2.5%= 46.97724 97.5% = 48.06260`
7. `require(boot)`
`boot_mean= function(x,i)`
`{`
`return(mean(x[i], na.rm=TRUE))`
`}`
`myboot =`
`boot(`
`data = dat_gentoo_bill_length,`
`statistic = boot_mean,`
`R = 10000)`
8. `quantile(myboot$t, c(0.025, 0.975))`

9. rarefaction_sampler = function(input_dat, n_iterations)

```
{  
  n_input_rows = nrow(input_dat)  
  n=nrow(input_dat)  
  
  results_out = matrix(  
    nrow = n_iterations,  
    ncol = n_input_rows)  
  
  for(i in 1:n_iterations)  
  {  
  
    for(j in 1:n)  
    {  
      rows_j = sample(n, size = j, replace=TRUE)  
  
      t1 = input_dat[rows_j, ]  
  
      t2 = apply(t1, 2, sum)  
  
      results_out[i, j] = sum(t2 > 0)  
    }  
  }  
  return(results_out)  
}
```

10. I found debugging the function to be the most difficult part, specifically figuring out what to set n equal to. Andrew and I easily found the issue with the next to last line being text rather than code, but we changed around many things in response to the error message saying that n did not exist. We were initially trying to keep the function coded so that any data input could work, but after having accidentally messed up my working function we worked with Jessica and put the moths data within the function. We later found the fix that is currently incorporated into the function so that it can work with any data.

```
rarefact_moths=rarefaction_sampler(moths[,-1], 10000)  
rare_mean=apply(rarefact_moths, 2, mean)  
rare_quant=apply(rarefact_moths, 2, quantile, probs=c(0.025, 0.975))  
rare=t(rbind(rare_mean, rare_quant))
```



```
matplot(
  rare,
  type="l", lwd=c(2, 2, 2), lty=c(1,2,4),
  col=brewer.pal(n=3, "Dark2"),
  xlab="Number of Sites Sampled",
  ylab="Average Number of Species Observed",
  main="Rare Moth Species Found in MA\nSoutheast Pine Barrens")
```

```
legend(
  "bottomright",
  legend=c("Mean", "Lower Quantile (2.5%)", "Upper Quantile (97.5%)"),
  lty=c(1,2,4), lwd=c(2,2,2), col=brewer.pal(n=3, "Dark2"), inset=c(.2,.2),
  title="Legend", title.col="Black", title.adj = .4,
  seg.len = 4)
```

12. In order to see all of the moth species you would want to visit about 18 sites. The refraction curve I made shows that at around 18 sites is when the mean intercepts the upper quantile.