

Julian Burgoff

11/7/2022

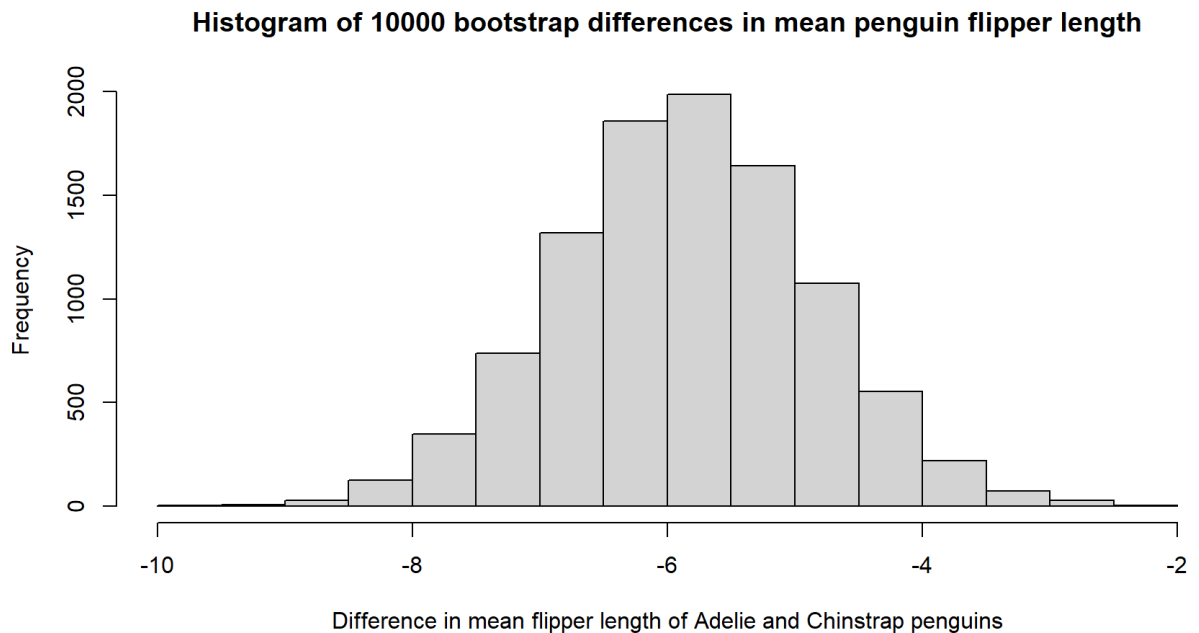
Analysis of Environmental Data

Lab 8

```
1. require(palmerpenguins)
   penguin_dat= droplevels(subset(penguins, species != 'Gentoo'))
   install.packages("simpleboot")
   require(simpleboot)

   adelie= subset(penguin_dat, species == 'Adelie')
   chinstrap= subset(penguin_dat, species== 'Chinstrap')

   pen_boot= two.boot(adelie$flipper_length_mm, chinstrap$flipper_length_mm, mean, R=
10000, na.rm= TRUE)
   sd(pen_boot$t)
[1] 1.0047
```



```
2.

3. quantile(pen_boot$t, c(0.025, 0.975))
   2.5%  97.5%
-7.819310 -3.874961
```

4. The bootstrap resampled differences in means are pretty normally distributed as seen in the histogram, where the highest frequencies are right around -6 declining fairly evenly on both sides of the histogram. Both the mean and median are almost identical (both right around -5.85) which further supports the claim that these data are normally distributed.

5. `pen_ecdf= ecdf(pen_boot$t)`

6. `1- pen_ecdf(-4.5)`
`[1] 0.0872`

7. `pen_ecdf(-8)`
`[1] 0.0161`

8. The null hypothesis is that there is no difference in mean flipper length between the two species of penguins.

The alternative hypothesis is that there is a significant difference in the means of the two species of penguins.

9. `veg<- read.csv("~/environmental_data/data/vegdata.csv")`
`dat_tree = droplevels(subset(veg, treatment %in% c("control", "clipped")))`
`wilcox.test(pine ~ treatment, data = dat_tree, alternative = "less")`

Wilcoxon rank sum test with continuity
correction

data: pine by treatment

W = 48, p-value = 0.9598

alternative hypothesis: true location shift is less than 0

10. `veg<- read.csv("~/environmental_data/data/vegdata.csv")`
`dat_tree = droplevels(subset(veg, treatment %in% c("control", "clipped")))`
`wilcox.test(pine ~ treatment, data = dat_tree, alternative = "less")`
`control= droplevels(subset(dat_tree, treatment == 'control'))`
`clipped= droplevels(subset(dat_tree, treatment == 'clipped'))`
`tree_boot = two.boot(clipped$pine, control$pine, mean, R= 10000, na.rm= TRUE)`
`boot.ci(tree_boot)`

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 10000 bootstrap replicates

CALL :

`boot.ci(boot.out = tree_boot)`

Intervals :

Level	Normal	Basic
95%	(3.25, 28.81)	(2.38, 27.62)

Level	Percentile	BCa
95%	(4.38, 29.62)	(5.00, 30.62)

11. `quantile(tree_boot$t, c(0.025, 0.975))`

2.5% 97.5%

4.375 29.625

12. Simpsons diversity is measure of species richness and evenness which considers the overall number of species and the relative abundance of each species.

13. # Calculate the sample mean and sd:

```
s_sidi_mean = mean(dat_all$s.sidi, na.rm = TRUE)
```

```
s_sidi_sd = sd(dat_all$s.sidi, na.rm = TRUE)
```

```
# Use the subset-by-name symbol ($) to create a
```

```
# new column of z-standardized values.
```

```
dat_all$s.sidi.standardized = (dat_all$s.sidi - s_sidi_mean)/s_sidi_sd
```

14. `dat_1 =`

```
subset(  
  dat_all,  
  select = c(b.sidi, s.sidi))
```

```
m = 10000
```

```
result_mc = numeric(m)
```

```
for(i in 1:m)
```

```
{
```

```
  index_1 = sample(nrow(dat_1), replace = TRUE)
```

```
  index_2 = sample(nrow(dat_1), replace = TRUE)
```

```
  dat_resampled_i =
```

```
  data.frame(  
    b.sidi = dat_1$b.sidi[index_1],
```

```
    s.sidi = dat_1$s.sidi[index_2]
```

```
  )
```

```
  fit_resampled_i = lm(b.sidi ~ s.sidi, data = dat_resampled_i)
```

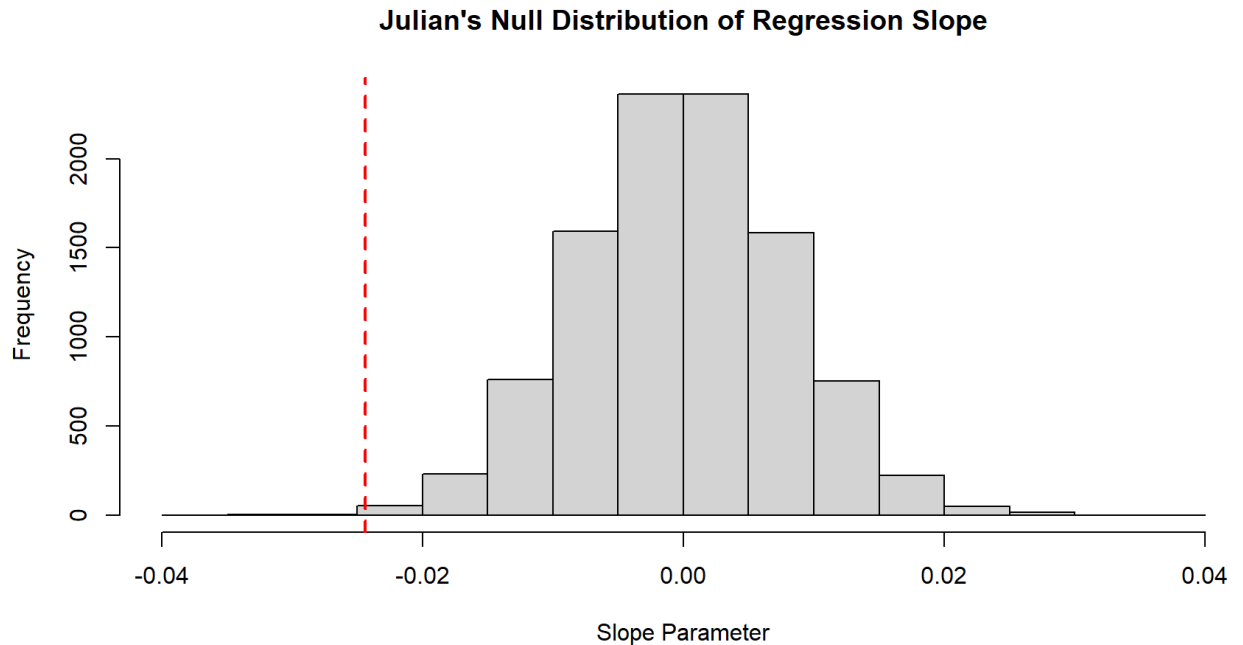
```

slope_resampled_i = coef(fit_resampled_i)[2]

print(slope_resampled_i)
# ... your loop code ...

result_mc[i] = coef(fit_resampled_i)[2]
}

```



15.

16. `quantile(result_mc, c(.05))`

5%

-0.01329749

The observed slope is less than the critical value

17. The results of the simulation suggest weak evidence for a negative relationship between vegetation diversity and bird diversity. The majority of simulations produced slope parameters more positive than the observed slope.

18. `m = 10000`

```
result_boot = numeric(m)
```

```
for(i in 1:m)
```

```
{
```

```
  index_1 = sample(nrow(dat_1), replace = TRUE)
```

```
  dat_boot = dat_1[index_1, ]
```

```
  fit_bs1 = lm(b.sidi ~ s.sidi, data = dat_boot)
```

```
# ... your loop code ...
```

```
    result_boot[i] = coef(fit_bs1)  
}
```

19.