

Biodiversity_Assignment

Jonathan Farr, Payton Arthur, Sammie Foley, and Jules Jones

10/31/2021

Setting up our packages

Seaweed percentages

```
seaweeds = perc %>%
  group_by(vertical_transect, quadrat) %>%
  mutate(unique_ID = as.numeric( paste0(vertical_transect, ".", quadrat))) %>%
  dplyr::select(c(unique_ID, quadrat, per_upright, per_crust)) %>%
  arrange(unique_ID)
```

```
## Adding missing grouping variables: `vertical_transect`
```

```
seaweeds_clean = data.frame(seaweeds[, -c(1,3)])
```

```
# setting up our dataframe so that we can use matrifly to arrange
# our data in long form and then calculate diversity with vegan
cleaned_biodiversity = biodiversity %>% # cleaning data to use matrifly
  drop_na(n) %>%
  group_by(vertical_transect, quadrat) %>%
  mutate(unique_ID = as.numeric( paste0(vertical_transect, ".", quadrat))) %>%
  dplyr::select(unique_ID, species, n)
```

```
## Adding missing grouping variables: `vertical_transect`, `quadrat`
```

```
cleaned_biodiversity = data.frame(cleaned_biodiversity[, -c(1:2)]) # removing extra columns
```

```
# putting it into a matrix format
biodiversity_matrix <- matrifly(cleaned_biodiversity) # using matrifly
```

Calculating diversity indices and linking it to seaweed cover

```
ShannonIndex <- data.frame(
  diversity(biodiversity_matrix, index = "shannon"))
head(ShannonIndex)
```

```
##      diversity.biodiversity_matrix..index....shannon..
## 1.1                                0.7595474
## 1.2                                0.6869616
## 1.3                                0.7529116
## 1.4                                0.0000000
## 2.1                                1.0690737
## 2.2                                0.8184438
```

```
ShannonIndex = ShannonIndex %>%
  rename("ShannonIndex" = diversity.biodiversity_matrix..index....shannon..)
```

```
richness <- biodiversity %>% # calculating richness per quadrat
  drop_na(n) %>%
  group_by(vertical_transect, quadrat) %>%
  summarize(rich = length(unique(species))) %>%
  mutate(unique_ID = as.numeric( paste0(vertical_transect, ".", quadrat)))

## `summarise()` has grouped output by 'vertical_transect'. You can override using the `.groups` argument
Data = cbind(ShannonIndex, seaweeds_clean, richness[,3]) # coming all our data
```

Now, we plot the relationship between diversity and seaweed cover using ggplot and geom_point

```
# visualizing upright percent cover and diversity
p1 = ggplot(data = Data, aes(x=per_upright, y =ShannonIndex))+
  geom_point()+
  geom_smooth(method=lm, se=FALSE)

# visualizing crust percent cover and diversity
p2 = ggplot(data = Data, aes(x=per_crust, y =ShannonIndex))+
  geom_point()+
  geom_smooth(method=lm, se=FALSE)

# visualizing upright percent cover and richness
p3 = ggplot(data = Data, aes(x=per_upright, y =rich))+
  geom_point()+
  geom_smooth(method=lm, se=FALSE)

# visualizing crust percent cover and richness
p4 = ggplot(data = Data, aes(x=per_crust, y =rich))+
  geom_point()+
  geom_smooth(method=lm, se=FALSE)
```

```
ggsave(plot = p1, filename = "upright_diversity.pdf")
```

Saving 6.5 x 4.5 in image

`geom_smooth()` using formula 'y ~ x'

```
ggsave(plot = p2, filename = "crustose_diversity.pdf")
```

Saving 6.5 x 4.5 in image

`geom_smooth()` using formula 'y ~ x'

```
ggsave(plot = p3, filename = "upright_richness.pdf")
```

Saving 6.5 x 4.5 in image

`geom_smooth()` using formula 'y ~ x'

```
ggsave(plot = p4, filename = "crustoes_richness.pdf")
```

Saving 6.5 x 4.5 in image

`geom_smooth()` using formula 'y ~ x'

Using regression to examine the relationships between seaweed % cover and shannon weiner diversity We used poisson distribution because it's more appropriate for data that's never negative

```
# simple linear regression for upright and diversity
upright_diversity = lm(ShannonIndex ~ per_upright, data = Data)
summary(upright_diversity) # not significant
```

```
##
## Call:
## lm(formula = ShannonIndex ~ per_upright, data = Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6695 -0.4279  0.1035  0.3406  0.6323
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.625711    0.120190   5.206 5.96e-05 ***
## per_upright  0.010950    0.007155   1.530  0.143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4608 on 18 degrees of freedom
## Multiple R-squared:  0.1151, Adjusted R-squared:  0.06596
## F-statistic: 2.342 on 1 and 18 DF,  p-value: 0.1433
# simple linear regression for crust and diversity
crust_diversity = lm(ShannonIndex ~ per_crust, data = Data)
summary(crust_diversity) # not significant

##
## Call:
## lm(formula = ShannonIndex ~ per_crust, data = Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.80504 -0.28904  0.07679  0.31905  0.62975
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.69243    0.13030   5.314 4.73e-05 ***
## per_crust    0.01251    0.03187   0.393  0.699
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4877 on 18 degrees of freedom
## Multiple R-squared:  0.008489, Adjusted R-squared: -0.0466
## F-statistic: 0.1541 on 1 and 18 DF,  p-value: 0.6993
# simple linear regression for upright and richness
upright_rich = lm(rich ~ per_upright, data = Data)
summary(upright_rich) # not significant

##
## Call:
## lm(formula = rich ~ per_upright, data = Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0944 -1.2292  0.0394  0.8889  3.6044
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.96055    0.43086   6.871 1.99e-06 ***
## per_upright  0.03346    0.02565   1.305   0.208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.652 on 18 degrees of freedom
## Multiple R-squared:  0.08638,    Adjusted R-squared:  0.03562
## F-statistic: 1.702 on 1 and 18 DF,  p-value: 0.2085
# simple linear regression for crust and richness
crust_rich = lm(rich ~ per_crust, data = Data)
summary(crust_rich) # not significant

##
## Call:
## lm(formula = rich ~ per_crust, data = Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3096 -1.2325 -0.2303  0.7681  3.7653
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.230294    0.461596   6.998 1.56e-06 ***
## per_crust    0.008807    0.112902   0.078   0.939
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.728 on 18 degrees of freedom
## Multiple R-squared:  0.0003379,    Adjusted R-squared:  -0.0552
## F-statistic: 0.006085 on 1 and 18 DF,  p-value: 0.9387
# we have no significant results (teehee) but that's ok! science
```

Table 1: Table 1. Linear regression results of the relationship between algae morphology and both Shannon-Wiener Index and Richness

Analysis	Algae.Type	P.Value	Adjusted.R.squared
Shannon-Wiener Index	upright	0.1433	0.06596
Shannon-Wiener Index	crustose	0.6993	-0.0466
Richness	upright	0.2085	0.03562
Richness	crustose	0.9387	-0.0552