

Diversity statistics

1. Exploring patterns in diversity

We're going to continue with the same NOAA dataset as last week to explore patterns in reef fish diversity at 13 Pacific islands. Remember that we cleaned the dataset - run those commands again (using your old script), and save the new dataset as a csv.

```
write.csv(cred, file="CRED_reef_diversity_clean.csv")
```

Now let's continue to work with the clean dataset. To examine diversity patterns, we need to save our diversity estimates to the workspace. You'll need to load vegan again to use the diversity functions.

```
require(vegan)
abundance<-table(cred$island,cred$species)
shannon<-diversity(abundance, index="shannon") ## Shannon diversity estimates
richness<-rarefy(abundance, min(rowSums(abundance))) ### estimates rarefied by number of individuals

# save the results as a data frame
diversity<-data.frame(shannon, richness)
diversity
```

You'll notice that the rownames are the island names. Let's add this as a variable, and then use match() to bring in the environmental predictor variables. We can create new variables with the \$ sign, like this:

```
diversity$island<-rownames(diversity)
## add in predictor variables using match()
diversity$region<-cred$region[match(rownames(diversity), cred$island)]
diversity$sst<-cred$SST[match(rownames(diversity), cred$island)]
diversity$state<-cred$state[match(rownames(diversity), cred$island)]
diversity$log_population<-cred$log_population[match(rownames(diversity), cred$island)]
diversity$productivity<-cred$productivity[match(rownames(diversity), cred$island)]
```

Now let's create some plots and examine the differences in diversity across islands.

```
par(mfrow=c(1,2)) ### set up a multipanel plot (1 row, 2 columns)
### Boxplots of diversity by region
boxplot(diversity$shannon ~ diversity$region, ylab="Shannon diversity")
boxplot(diversity$richness ~ diversity$region, ylab="Rarefied richness")
### Diversity against SST, coloured by region
plot(diversity$sst, diversity$shannon, col=diversity$region)
plot(diversity$sst, diversity$richness, col=diversity$region)
## Shannon vs. richness - are the diversity indices correlated?
plot(diversity$richness, diversity$shannon, col=diversity$region)
cor(diversity$richness, diversity$shannon)
## Create plot with island names instead of points
plot(diversity$richness, diversity$shannon, col="transparent",xlim=c(25, 60))
text(diversity$richness, diversity$shannon, labels=diversity$island)
```

You should examine the relationships between diversity and all of our predictor variables. Before carrying out statistical analyses, you should be familiar the patterns that you're testing. Which islands are the most diverse? Which islands are the most productive? This works for any ecological dataset - are there spatial or temporal trends in my response variable? How does my response variable change with each of my explanatory variables?

2. Statistical tests in R

R comes with a range of basic statistical tests (t test, anova, regression), and there are many packages for more complex

methods (mixed effect models, principal component analysis, Bayesian stats). As always, google to find a specific package for the statistical test that you're trying to run.

Let's test for differences in mean Shannon diversity between regions.

```
t.test(shannon ~ region, diversity)
## Shannon diversity estimates are significantly different between regions

## remember the assumptions of a t test - parametric: normally distributed
hist(diversity$shannon)
shapiro.test(diversity$shannon)
bartlett.test(diversity$shannon ~ diversity$region)
```

With observational data, we're often interested in modelling relationships with multiple explanatory variables (multiple regression). As we used in previous tutorials, `lm()` is the standard function for linear models. We should also test our model assumptions by looking at residual plots.

```
## Fit linear regression: diversity as function of state + prod + sst
shan_mod<-lm(shannon ~ state + productivity + sst, diversity)
summary(shan_mod)

## Examine model fit + assumptions
plot(shan_mod)
## Residual plots
par(mfrow=c(1,2))
plot(shan_mod$resid)
plot(diversity$productivity, shan_mod$resid)
abline(b=0, a = 0, lty=2)
## Normally distributed residuals?
shapiro.test(shan_mod$resid)
```

We can quickly visualise our model results using the `visreg()` package.

```
install.packages("visreg")
require(visreg)
par(mfrow=c(2,2))
visreg(shan_mod)
```

`visreg` creates partial residual plots - the effect of each explanatory variable is plotted *while holding the model predictions for each other explanatory variable constant*. These are useful for quick interpretations of your model output, but you should be using base plot to present your results in a document.

Try the same tests for species richness - is variation in species richness due to SST, population size, or productivity?

3. Dominance and evenness

Very simply, species diversity = richness + evenness. Evenness is a measure of the relative abundances of species - is one species dominant in a community? Or are there many common species and few rare species?

`vegan` doesn't have an evenness function, but we can use our own. Pielou's evenness = Shannon diversity / log(species richness), therefore:

```
diversity$evenness<-diversity$shannon/log(specnumber(abundance))
diversity
```

Evenness isn't particularly useful on its own - what we really want to see is the relative abundance of different species. Let's go back to the original dataset and create some rank-abundance curves.

```
par(mfrow=c(1,1))
plot(abundance[1,])    ### We need to sort species by decreasing abundance
plot(sort(abundance[1,], decreasing=TRUE), main="Baker", ylab="Abundance", xlab="Rank")
```

We can wrap this plot in a loop to get the rank-abundance curves for every island. A loop uses curly brackets `{}` to repeat a

section of code - in ecology, we mainly use them for bootstrapping data, building population models, and running simulations.

Here, we're going to create a rank-abundance plot for each row of the abundance data frame (i.e. each island). We use "i" to run through the numbers 1 to 8, and the loop finishes when i = 8. Let's use an example loop first:

```
test<-seq(1, 1000, by=35)
for (i in 1:8){
  print(test[i])
}
```

This loop prints each value of test from 1 to 8. Now let's loop a command that creates a rank-abundance plot:

```
par(mfrow=c(2,4)) # set multipanel plot for number of plots (8)
for (i in 1:8){
  plot(sort(abundance[i,], decreasing=TRUE), main=rownames(abundance)[i], ylab="Abundance", xlab="Rank")
}
```

Next week, we'll use some loops to create rank-abundance plots for herbivores and carnivores, and examine how dominance changes between remote and disturbed islands.