BIO 423 - Lab 7

Benjamin Blonder

Spring 2019

Learning outcomes

Content goals:

- Identify important species in food webs (predator-prey networks)
- Compare metrics of network structure across systems
- Test hypotheses about food web structure and habitat

R goals:

- Learn the `igraph` package
- Improve fluency with `lapply` and `sapply` and `inner_join`
- Use anonymous functions

Food webs

We will work with predator-prey food webs arising from studies of multiple stream sites in New Zealand, Maine and North Carolina by Thompson and Townsend over several decades. More information about the data is available at:

`https://www.nceas.ucsb.edu/interactionweb/html/thomps_towns.html`

We also have metadata available for each site (`WebName`), comprising its habitat type (`Habitat`) and location (`Location`).

Each food web is represented as a matrix, with a `0` entry if the species do not interact and a `1` if they do. There are several food webs to read in. We will use the `read_excel` function and the `lapply` function to programmatically load up all the food webs at once.

The `lapply` function takes a `list` or `vector` data structure as input, applies a function of choice to each entry of the list/vector, and returns a list containing the outputs of this function. The `sapply` function does the same thing, but simplifies the output to a vector if possible. Let's see it in action!

If we want to write a more complex function for `lapply`, we can define our own anonymous function as `function(x) {  }`. The last line inside the `{}` should be a `return(...)` statement, telling the function what to pass back to `lapply`.

```r
library(readxl) # install this package if needed
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
```

```r
library(reshape2)
library(ggplot2)

# load metadata for all the sites...
foodweb_metadata = read.csv("thoms_towns_metadata.csv")
names(foodweb_metadata) = c("WebName","Habitat","Location")
str(foodweb_metadata)
```

```
## 'data.frame':    26 obs. of  3 variables:
##  $ WebName : Factor w/ 26 levels "AkatoreA","AkatoreB",..: 1 2 7 3 26 18 4 5 6 12 ...
##  $ Habitat : Factor w/ 4 levels "Broadleaf forest",..: 3 3 3 3 3 3 2 2 2 4 ...
##  $ Location: Factor w/ 3 levels "Maine, USA","North Carolina, USA",..: 3 3 3 3 3 3 3 3 3 3 ...
```

```r
# list all files in the directory
files = dir(path='thoms_towns_excel', pattern='*xls', full.names=TRUE)
head(files)
```

```
## [1] "thoms_towns_excel/AkatoreA.xls"  "thoms_towns_excel/AkatoreB.xls"
## [3] "thoms_towns_excel/Berwick.xls"   "thoms_towns_excel/Blackrock.xls"
## [5] "thoms_towns_excel/Broad.xls"     "thoms_towns_excel/Canton.xls"
```

```r
# also extract the name of the food web from the filename
# first use basename to get the filename with no path
# then use gsub to delete the file extension
file_basename = gsub(".xls", "", sapply(files, basename), fixed=TRUE)
head(file_basename)
```

```
##  thoms_towns_excel/AkatoreA.xls  thoms_towns_excel/AkatoreB.xls
##                      "AkatoreA"                      "AkatoreB"
##   thoms_towns_excel/Berwick.xls thoms_towns_excel/Blackrock.xls
##                       "Berwick"                     "Blackrock"
##     thoms_towns_excel/Broad.xls    thoms_towns_excel/Canton.xls
##                         "Broad"                        "Canton"
```

```r
# run read_excel for every food web
foodwebs_all = lapply(files, read_excel)
```

```
## readxl works best with a newer version of the tibble package.
## You currently have tibble v1.4.2.
## Falling back to column name repair from tibble <= v1.4.2.
## Message displays once per session.
```

```r
# list elements are accessed as [[1]], [[2]], etc. -
# let's look at the first food web
head(foodwebs_all[[1]])
```

```
## # A tibble: 6 x 86
##   X__1  `Unidentified d~ `Terrestrial in~ `Plant material~ Meiofauna
##   <chr>            <dbl>            <dbl>            <dbl>     <dbl>
## 1 Unid~                0                0                0         0
## 2 Terr~                0                0                0         0
## 3 Plan~                0                0                0         0
## 4 Meio~                0                0                0         0
## 5 Achn~                0                0                0         0
## 6 Achn~                0                0                0         0
## # ... with 81 more variables: `Achnanthes inflata` <dbl>, `Achnanthes
## #   lanceolata` <dbl>, `Achnanthes linearis` <dbl>, `Achnanthes
## #   minutissima` <dbl>, `Ankitodesmus sp.` <dbl>, Batrachospermum <dbl>,
## #   `Blue-green algae` <dbl>, Calothrix <dbl>, `Cocconeis
## #   placentula` <dbl>, `Cymbella kappi` <dbl>, `Cymbella kappii` <dbl>,
## #   `Cymbella mulleri` <dbl>, `Diatoma heimale` <dbl>, `Epithemia
## #   sorex` <dbl>, `Eunotia pectinalis` <dbl>, `Fragilaria
## #   vaucheriae` <dbl>, `Frustulia rhomboides` <dbl>, `Gomphoneis
## #   herculeana` <dbl>, `Gomphonema accuminatum` <dbl>, `Gomphonema
## #   angustatum` <dbl>, `Gomphonema intricatum` <dbl>, `Gomphonema
## #   parvulum` <dbl>, `Gomphonema sp. III` <dbl>, `Gomphonema sp.
## #   unk` <dbl>, `Gomphonema truncatum` <dbl>, `Green algae` <dbl>,
## #   Gyrosigma <dbl>, `Melosira varians` <dbl>, `Navicula avenacea` <dbl>,
## #   `Navicula rhynocephala` <dbl>, `Nitzschia dissipata` <dbl>, `Nitzschia
## #   dubia` <dbl>, `Nitzschia linearis` <dbl>, `Pinnularia spp.` <dbl>,
## #   `Rhoicosphenia curvata` <dbl>, Staurostratum <dbl>, `Surirella
## #   elegans` <dbl>, `Synedra ulna` <dbl>, `Tabellaria flocculosa` <dbl>,
## #   Ulothrix <dbl>, `Ameletopsis perscitus (N=13)` <dbl>, `Aoteapsyche
## #   raruraru (N=8)` <dbl>, `Austroperla cyrene (N=15)` <dbl>,
## #   `Austrosimulium australense (N=4)` <dbl>, `Chironomini
## #   (N=Cricotopus)` <dbl>, `Coloburiscus humeralis (N=5)` <dbl>,
## #   `Costachorema xanthoptera (N=3)` <dbl>, `Cricotopus sp I (N=6)` <dbl>,
## #   `Cricotopus sp II (N=1)` <dbl>, `Cristaperla (N=15)` <dbl>,
## #   `Deleatidium (N=15)` <dbl>, `Eriopterini (N=6)` <dbl>, `Eye forward
## #   chiron (N=1)` <dbl>, `Helicopsyche albescens (N=3)` <dbl>, `Hudsonema
## #   aliena (N=2)` <dbl>, `Hudsonema amabilis (N=12)` <dbl>, `Hydora nitida
## #   (ad) (N=15)` <dbl>, `Hydora nitida (l) (N=15)` <dbl>, `Hydrobiosella
## #   stenocerca (N=15)` <dbl>, `Hydrobiosis parumbripennis
## #   (N=Chostachorema)` <dbl>, `Mischoderus (N=1)` <dbl>, `Nannochorista
## #   phillpotti (N=1)` <dbl>, `Neozephlebia scita (N=10)` <dbl>, `Oligo II
## #   (N=3)` <dbl>, `Oligo Lumbri pink (N=10)` <dbl>, `Oligo skinny
## #   (N=10)` <dbl>, `Paracalliope fluviatalus (N=10)` <dbl>, `Paracalliope
## #   pale (N=10)` <dbl>, `Paralimnophila skuseii (N=3)` <dbl>,
## #   `Paucispinigera approximata (N=1)` <dbl>, `Polypedellum (N=4)` <dbl>,
## #   `Polypedellum II (N=1)` <dbl>, `Polyplectropus puerilis (N=1)` <dbl>,
## #   `Scirtid broad spp. (N=2)` <dbl>, `Sphaerid (N=1)` <dbl>, `Stenoperla
## #   prasinia (N=10)` <dbl>, `Stictocladius (N=2)` <dbl>, `Zelandobius
## #   (N=2)` <dbl>, `Zelandoperla fenestrata (N=10)` <dbl>, `Zelandotipula
## #   (N=1)` <dbl>, `Anguilla dieffenbachii (N=1)` <dbl>
```

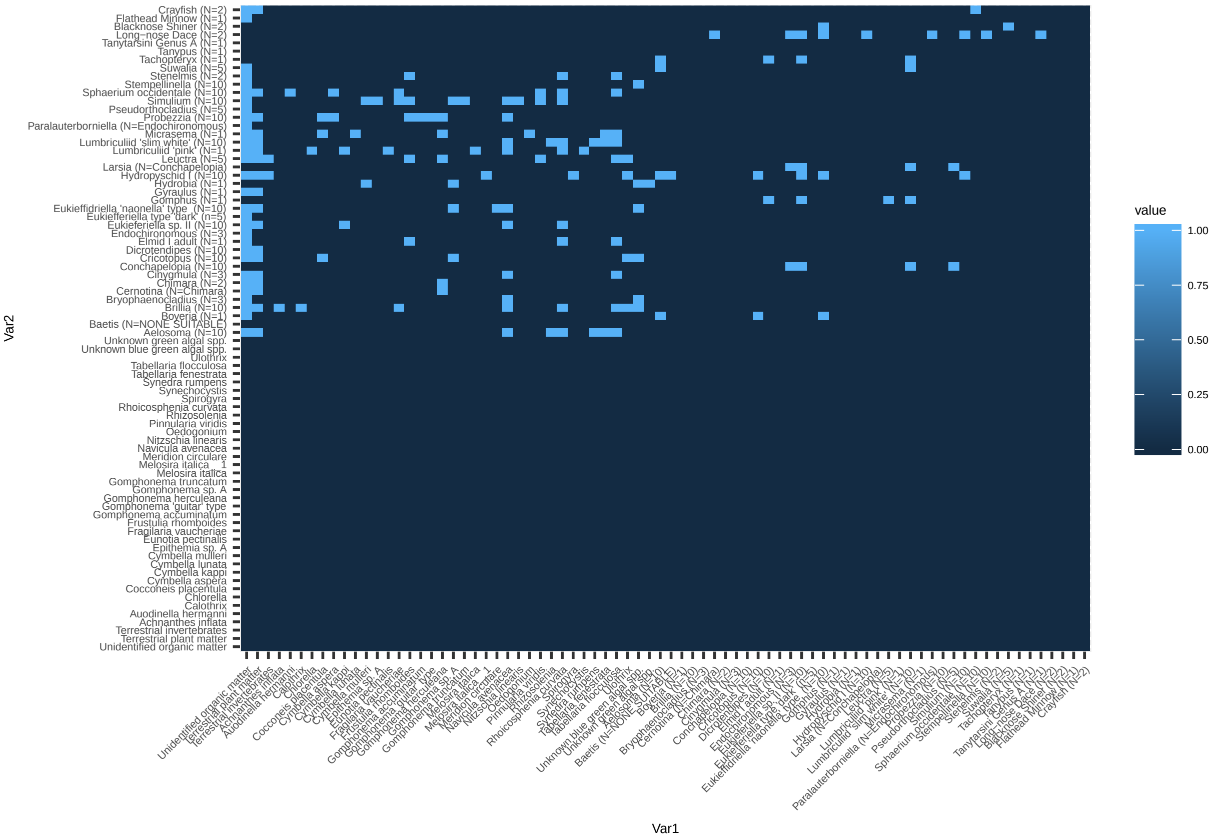
```
# the first column is incorrectly set to the names of the species
# so we should drop the 1st column of each data frame
# this also relies on using an `anonynmous function`, as the second
# argument to the lapply function.

foodwebs_all_clean = lapply(foodwebs_all, function(x) {
  # drop the first column
  x = x[,-1]

  # make the data frame square by removing extra columns
  mindim = min(dim(x))
  x = x[1:mindim, 1:mindim]

  # keep the species names
  speciesnames = names(x)
  # convert to matrix (needed later on...)
  x_matrix = as.matrix(x)
  dimnames(x_matrix) = list(speciesnames, speciesnames)

  # return only the processed matrix
  return(x_matrix)
})
sapply(foodwebs_all_clean, dim)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]   85   58   79   87   95  109   49   58   71    86    97   107    86
## [2,]   85   58   79   87   95  109   49   58   71    86    97   107    86
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## [1,]    96    98    78   105    71    78    78   113    83    79    92
## [2,]    96    98    78   105    71    78    78   113    83    79    92
##      [,25] [,26]
## [1,]    78    69
## [2,]    78    69
```

```
# the foodwebs are now square matrices
# with each row and column corresponding to a species pair

# last, we need to set the names of the list to the names of the food webs
names(foodwebs_all_clean) = file_basename
```

We are now in a position to begin visualizing and analyzing these networks. The simplest way to visualize the networks is as a heatmap. Here is an example of how to do this for one of the food webs:

```
# choose an example web
myweb = foodwebs_all_clean$Troy
myweb_melted = melt(as.array(myweb))
ggplot(myweb_melted,aes(x=Var1,y=Var2,fill=value)) +
  geom_tile() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(text = element_text(size=5))
```



Another more powerful way to visualize the food web is as a graph. We need the `igraph` package for this.

```
library(igraph)
```

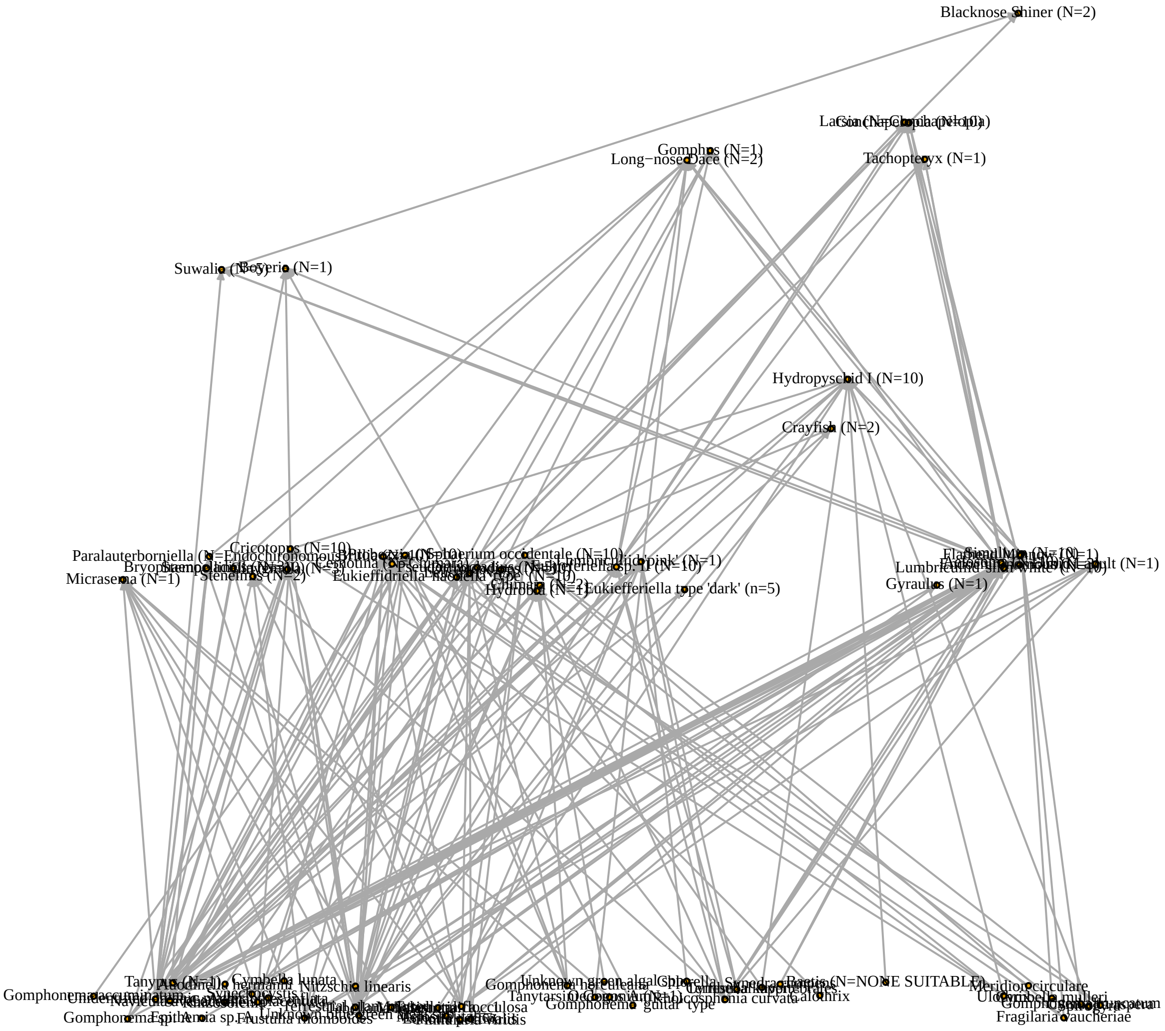
```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
library(NetIndices)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
# pick a web again
myweb = foodwebs_all_clean$Troy
```

```r
myweb_graph = graph.adjacency(myweb,mode='directed')

# calculate a trophic index based on the total flow through to each species
tl <- TrophInd(myweb)
# arrange species randomly
layout_visual <- layout.random(myweb_graph)
# set y-coordinate to the trophic position
# add a bit of jitter to the y-axis coordinate
layout_visual[,2] <- jitter(tl$TL,amount=0.05)

# igraph plotting is different than ggplot2...
plot(myweb_graph,
     layout=layout_visual,
     vertex.size=0,
     edge.arrow.size=0.25,
     vertex.label.color='black',
     vertex.label.cex=0.5)
```



It is also possible to summarize other information on a species-level basis. One key statistic is the degree, i.e. the number of linkages per species. We can count the in-degree and the out-degree, which respectively reflect the # of species that this species predates or is predated by.

```
degreevals_out = degree(myweb_graph,mode='out')
degreevals_in = degree(myweb_graph,mode='in')
head(sort(degreevals_in,decreasing=TRUE))
```

```
##     Hydropyschid I (N=10)           Simulium (N=10)
##                        12                        11
##            Brillia (N=10) Lumbriculiid 'pink' (N=1)
##                        10                         9
##          Probezzia (N=10)       Long-nose Dace (N=2)
##                         9                         9
```

Questions

1. Make a graph of the `Canton` network.
2. In the `Canton` food web, what species is the top predator? Give its Latin name from the data and also look up its common name.
3. In the `Canton` food web, how many species are primary producers (hint: look in `tl`, and consider making a histogram. You can also check the help for `TrophInd` to understand what you are seeing). Also report the names of two primary producers, and explain what type of organism they are.
4. One species in the `Canton` network has a loop. You can find it by determining which entry of myweb has a non-zero diagonal entry. (Hint: `which(diag(x)==1)`). What species is this? What does the loop represent biologically?
5. In the `Canton` network, which 'species' is predated by the most other species? Which species predates the most other species?
6. One hypothesis is that more specialized species should be predators, i.e. have a higher trophic level. Evaluate support for this hypothesis in the `Canton` food web by by plotting trophic level vs. in-degree. Make a linear regression using `lm` to evaluate the hypothesis, but also show a graph and evaluate the data visually.

Network level metrics

It is also possible to calculate various indices for the overall network. For example, we can calculate the mean in-degree of species for each network to summarize the characteristic number of prey species per species.

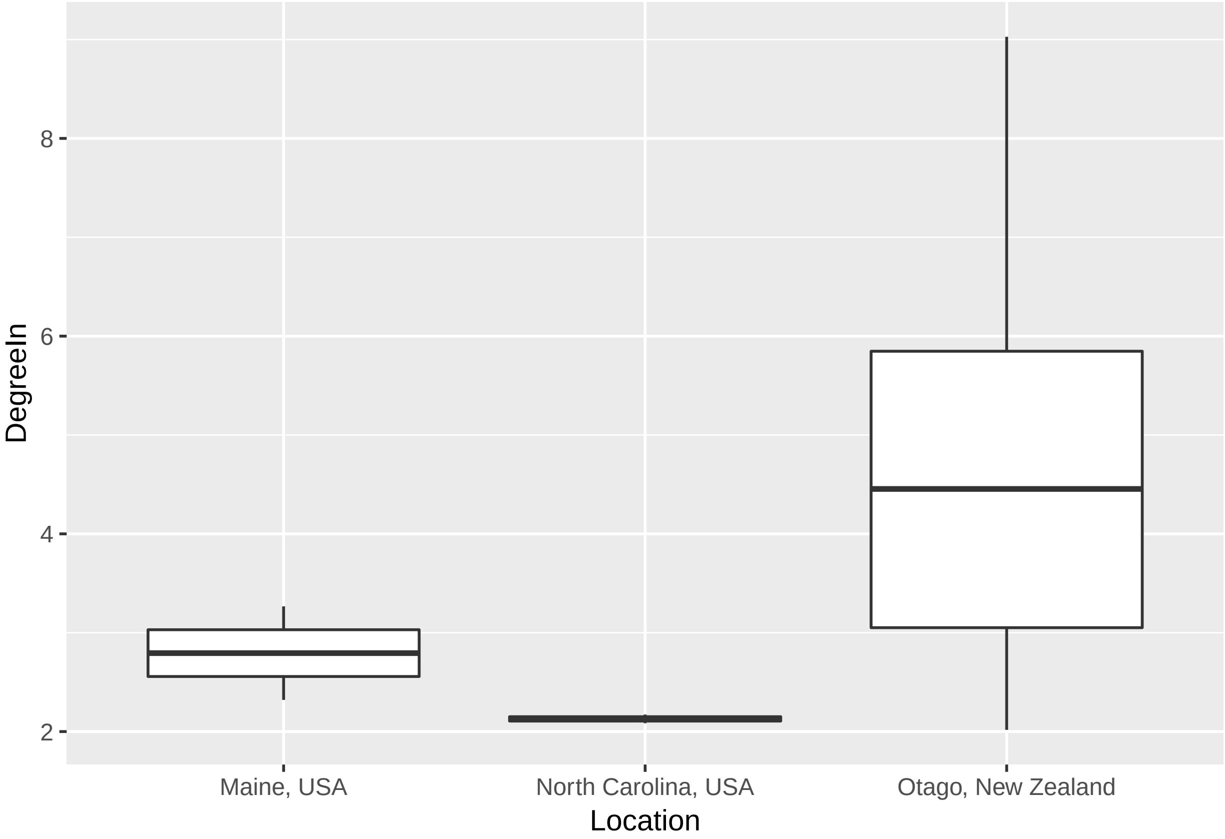
```
mean_degree_in_all = sapply(foodwebs_all_clean, function(x) {
  myweb_graph_this = graph.adjacency(x,mode='directed')
  degree_in_vals = degree(myweb_graph_this, mode='in')
  mean_degree_in = mean(degree_in_vals)
  return(mean_degree_in)
  })
# convert from vector to data frame
mean_degree_in_all_df = data.frame(
                          WebName=names(mean_degree_in_all),
                          DegreeIn=mean_degree_in_all)

head(mean_degree_in_all_df)
```

```
##              WebName DegreeIn
## AkatoreA   AkatoreA 2.670588
## AkatoreB   AkatoreB 2.017241
## Berwick     Berwick 3.037975
## Blackrock Blackrock 4.310345
## Broad         Broad 5.947368
## Canton       Canton 6.495413
```

```
# join the degree data with the metadata using the WebName column
mean_degree_joined = inner_join(foodweb_metadata, mean_degree_in_all_df, by='WebName')

head(mean_degree_joined)
```

```
##      WebName     Habitat           Location DegreeIn
## 1   AkatoreA Pine forest Otago, New Zealand 2.670588
## 2   AkatoreB Pine forest Otago, New Zealand 2.017241
## 3    Catlins Pine forest Otago, New Zealand 2.244898
## 4    Berwick Pine forest Otago, New Zealand 3.037975
## 5     Venlaw Pine forest Otago, New Zealand 2.710145
## 6 Narrowdale Pine forest Otago, New Zealand 2.183099
```

```
ggplot(mean_degree_joined,aes(x=Location,y=DegreeIn)) +
  geom_boxplot()
```



Questions

7. A key metric of food webs is connectance, which can be defined as L/S^2 where L is the number of links and S is the number of species in the web. You can get L as `length(E(myweb_graph))` and S as `length(V(myweb_graph))` (`E` for edges, `V` for vertices). Calculate the connectance for all the different sites and store it as a new column in a data frame, then join with the metadata. Report the mean and standard deviation of connectance across all sites.

8. Different habitat types may have different connectances, reflecting effects of environment on species interactions. One hypothesis is that food webs in broadleaf forest have higher connectance than those in pine forest because they have more diverse species. Assess this hypothesis by a) making a boxplot graph of `Habitat` vs `L/S^2` and b) running a t-test to compare connectance between the two forest types (hint: remember to subset the data first). Is the hypothesis supported?

Optional questions for graduate students

- Another key property of food webs is their stability when species go extinct. For example, extinctions may cascade if a species loses all of its prey species. You can simulate this process by re-generating food webs after removing species *i* from the dataset (e.g. by removing row *i* and column *i* from the underlying matrix), then further removing all species that have an in-degree of zero. A good way to summarize the effect of this removal is the network connectance. Under a scenario of random species extinction, how does network connectance vary with # of species removed? You can report results for a single food web, or explore this for all the food webs in a comparative approach. In either case you will have to summarize results across a distribution of random species extinctions. On average, how many species have to go extinct for the connectance to drop by fifty percent? (Hint: you can either write the code yourself, or check out the `NetworkExtinction` R package)
- What if extinction is non-random, and biased towards species at higher trophic levels? Repeat the above analysis, explaining your choices for how to incorporate this non-randomness into the simulation.
- Is it realistic that the food web does not rewire itself when a species is simulated to go extinct? Explain your rationale. Consider writing a simulation to address rewiring after extinction, e.g. based on trophic level.

An optional further topic (bipartite networks)

- Bipartite networks can be used to represent systems where the species fall into two mutually exclusive categories, e.g. plants and pollinators, disease vectors and hosts, and so on. We don't have time to do a full analysis of such networks in this class, but if you want to learn more, check out and follow a great tutorial at `https://fukamilab.github.io/BIO202/09-B-networks.html`. Follow the tutorial.

What to hand in

- A single Word Document including:
- written answers (1-2 sentences) and figures for each question above
- A copy of your R script (the contents of your `.R` file pasted into the Word document)
- Author contribution statement