

McGill University, BIOL 310 - Scaling in ecology

Riva, Federico

03/02/2022

Introduction

This is a tutorial complementing the lecture on scales, available at <https://github.com/FedericoRiva>. We will focus on three main parts: 1) spatial data preparation; 2) scale dependency in species rarity; and 3) scale dependency in biodiversity trends

Spatial data preparation

```
# set up
# packages needed in the tutorial
packages = c("data.table", "raster", "sp", "maptools", "dplyr", "vegan")

# load the packages; if missing, install & load
package.check <- lapply(
  packages,
  FUN = function(x) {
    if (!require(x, character.only = TRUE)) {
      install.packages(x, dependencies = TRUE)
      library(x, character.only = TRUE)
    }
  }
)
```

```
## Loading required package: data.table
```

```
## Warning: package 'data.table' was built under R version 4.1.1
```

```
## Loading required package: raster
```

```
## Loading required package: sp
```

```
##
```

```
## Attaching package: 'raster'
```

```
## The following object is masked from 'package:data.table':
```

```
##
```

```
##      shift
```

```

## Loading required package: maptools

## Checking rgeos availability: TRUE

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:raster':
##
##     intersect, select, union

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: vegan

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.5-7

```

```

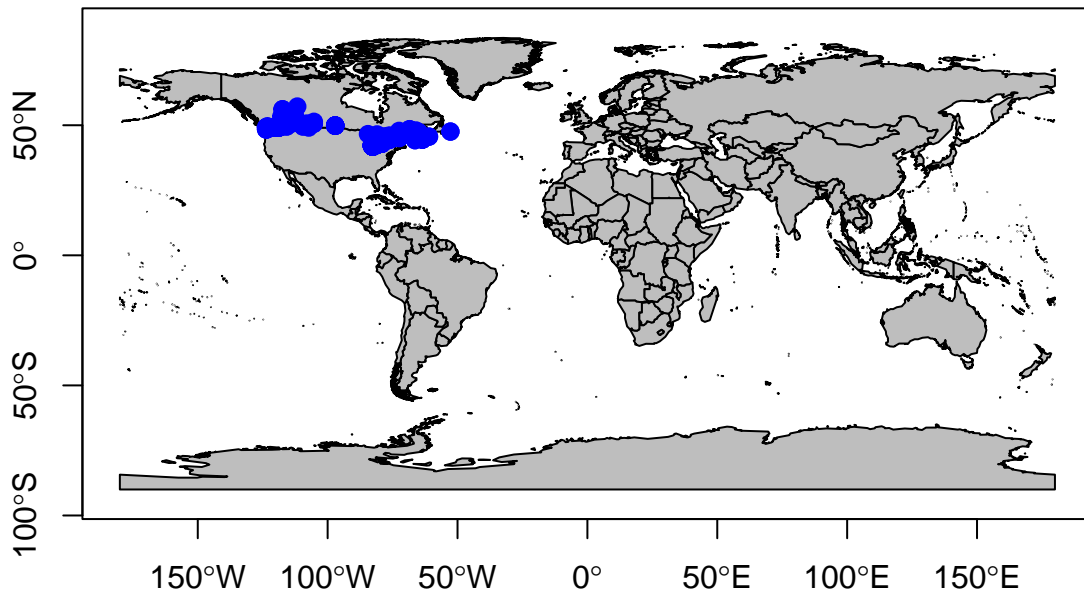
# # tpen and clean data
# library(data.table)
#
# # to handle raster and vector spatial data
# library(raster)
# library(sp)
#
# # to get the world map when plotting original data
# library(maptools)
#
# # data wrangling
# library(dplyr)
#
# # analysis of diversity
# library(vegan)

```

```
query_gbif <- data.table::fread("Data\\query_gbif_2013.csv", header = TRUE) # data downloaded from GBIF
```

Always look at your data, particularly when dealing with spatial data. Let's check if the cabbage white (*Pieris rapae*), a generalist species common in Europe, NA and Oceania, occurs only in CA

```
example <- query_gbif[query_gbif$species == "Pieris rapae"]
data(wrld_simpl)
plot(wrld_simpl,
      #xlim=c(min(example$decimalLongitude)-1, max(example$decimalLatitude)+1), # you can adjust the extent
      #ylim=c(min(example$decimalLongitude)-1, max(example$decimalLatitude)+1),
      axes=TRUE, col="grey")
points(example$decimalLongitude, example$decimalLatitude, col="blue", pch=20, cex=1.7)
```



All good - Data restricted to Canada, no clear outliers and/or occurrences in the ocean, etc. Note that this plot is based on geographic coordinates, expressed in degrees. To evaluate issues of spatial scales, one needs to account for the area of different raster cells, which require projected coordinates, expressed in metric system.

Open spatial data

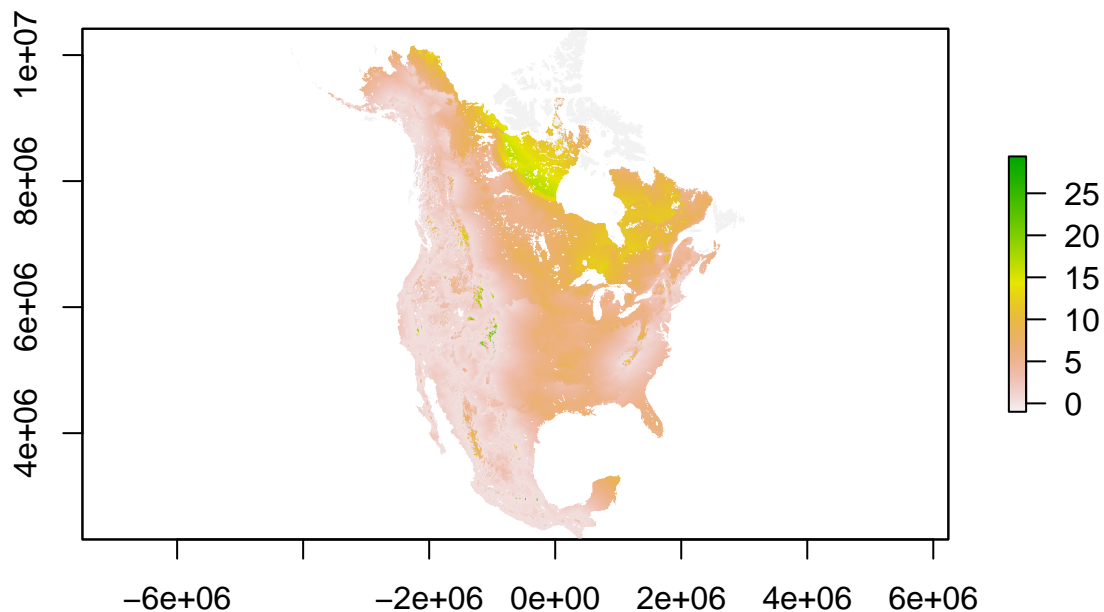
Next we will explore some spatial datasets, both in vectorial and raster formats, to familiarize with R spatial; you can find more at <https://rspatial.org/>. We begin by opening a dataset of climatic velocity, approximated using the euclidean distance (i.e., shortest straight line) between two cells with analog climates. The dataset is available at <https://adaptwest.databasin.org/pages/adaptwest-velocitymed/>.

```
# open euclidean-distance based velocity;
VEL_ED <- raster::raster("Data\\velocity\\velocity.ED.tif") #Velocity Euclidean Distance
```

```
VEL_ED
```

```
## class      : RasterLayer
## dimensions : 1621, 1456, 2360176 (nrow, ncol, ncell)
## resolution : 5000, 5000 (x, y)
## extent     : -4272000, 3008000, 2313000, 10418000 (xmin, xmax, ymin, ymax)
## crs        : +proj=lcc +lat_0=0 +lon_0=-95 +lat_1=49 +lat_2=77 +x_0=0 +y_0=0 +datum=WGS84 +units=m +no_defs
## source     : velocity.ED.tif
## names      : velocity.ED
```

```
plot(VEL_ED)
```



Two considerations: (i) the spatial grain of this raster is 5 km, and the coordinate system is a projected system in World Geodetic System 1984 (WGS84; https://en.wikipedia.org/wiki/World_Geodetic_System)

The plot shows up to 25 km/year migration needed to track future climates

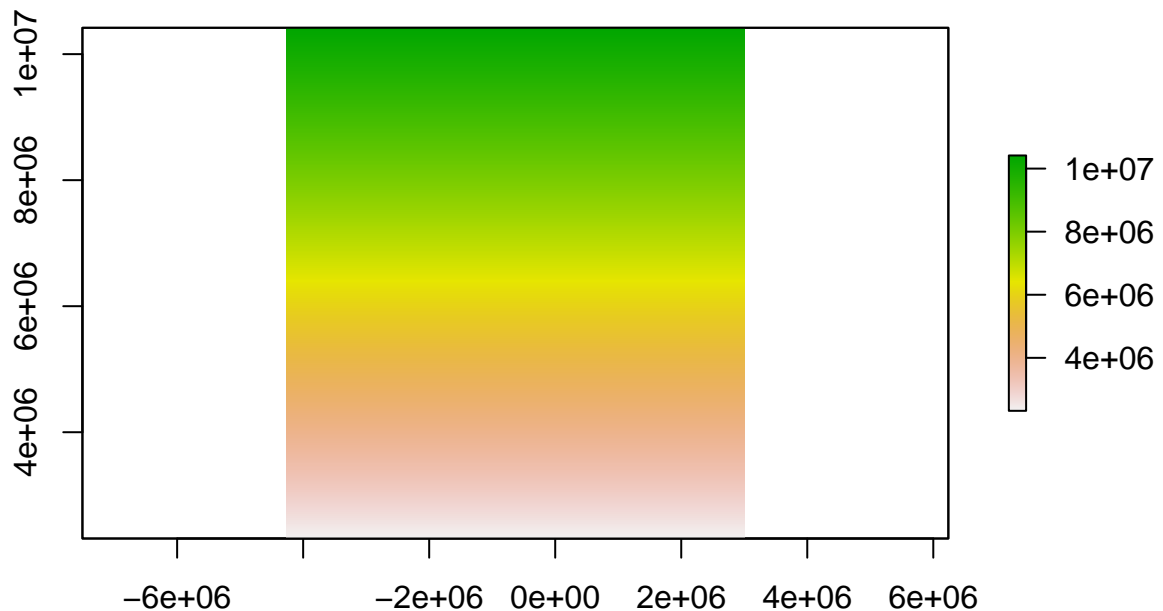
```
crs <- crs(VEL_ED)
crs
```

```
## CRS arguments:
## +proj=lcc +lat_0=0 +lon_0=-95 +lat_1=49 +lat_2=77 +x_0=0 +y_0=0
## +datum=WGS84 +units=m +no_defs
```

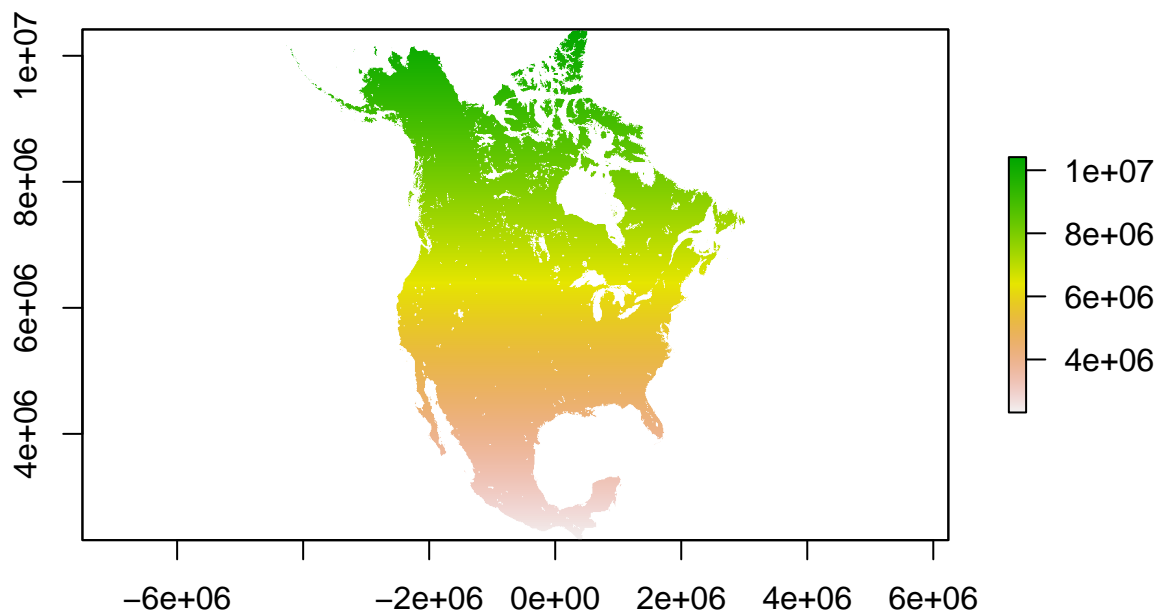
We will establish the coordinate reference system (crs) from VEL_ED as the reference system in our analysis.

Next, let's create a raster of latitudinal values for all the cells present in VEL_ED

```
# created a raster having the same extent of VEL_ED, but containing latitude values instead  
  
LAT <- VEL_ED # copy VEL_ED into a new object, LAT, which will  
raster_coordinates <- coordinates(VEL_ED) # extract the coordinates of every cell  
LAT[] <- raster_coordinates[, 2] # replace the content of every cell in LAT with its latitude  
  
plot(LAT)
```



```
# remove pixels around NA for LAT  
# create a mask  
mask <- VEL_ED # create a mask based on VEL_ED  
mask[!is.na(mask[])] <- 1 # every value in VEL_ED that is not NA becomes a 1  
LAT <- LAT*mask # multiply the mask for LAT to remove all NAs in VEL_ED  
  
plot(LAT)
```



Aggregate and observe the results

```
# aggregate to evaluate the same environmental gradients at different spatial scales
VEL_ED_50 <- raster::aggregate(VEL_ED, 10)
VEL_ED_500 <- raster::aggregate(VEL_ED_50, 10)

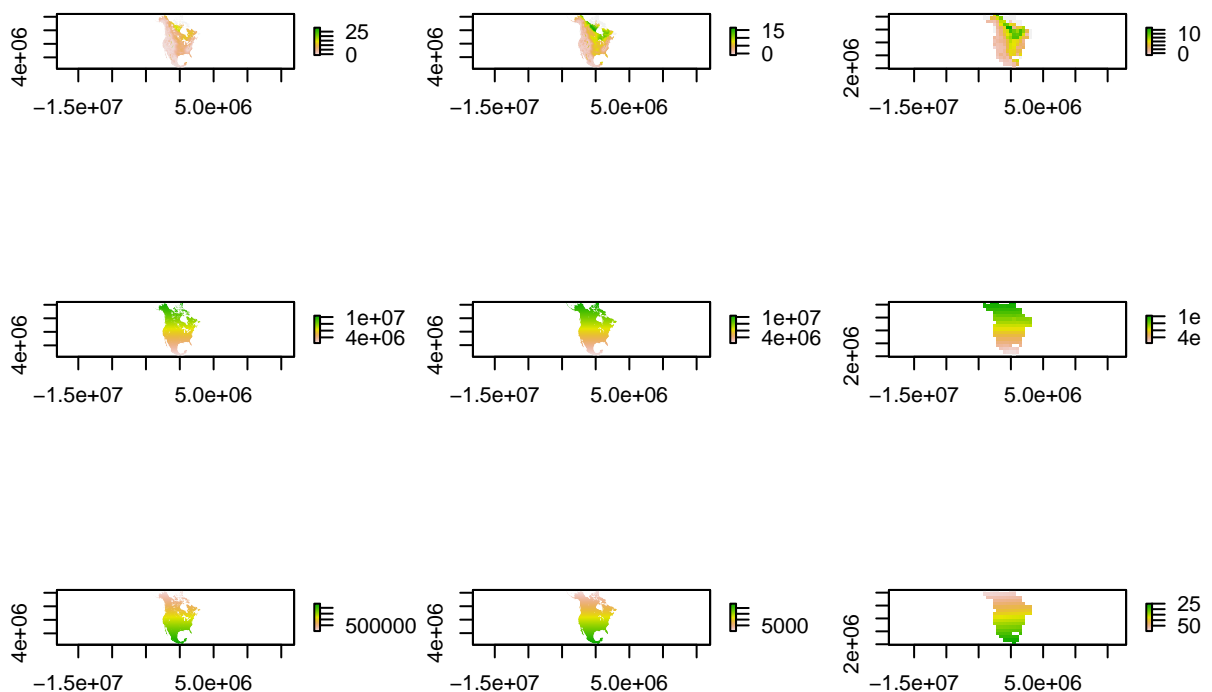
LAT_50 <- raster::aggregate(LAT, 10)
LAT_500 <- raster::aggregate(LAT_50, 10)

# create unique ID cells for every 5, 50 and 500 km cell
ID <- VEL_ED
ID_50 <- VEL_ED_50
ID_500 <- VEL_ED_500

ID[] <- seq(1, ncell(VEL_ED))
ID_50[] <- seq(1, ncell(VEL_ED_50))
ID_500[] <- seq(1, ncell(VEL_ED_500))

# crop NA cells
ID <- ID*mask
ID_50 <- ID_50*(raster::aggregate(mask,10))
ID_500 <- ID_500*(raster::aggregate(mask,100))
```

```
# check visually the data
par(mfrow=c(3,3))
plot(VEL_ED)
plot(VEL_ED_50)
plot(VEL_ED_500)
plot(LAT)
plot(LAT_50)
plot(LAT_500)
plot(ID)
plot(ID_50)
plot(ID_500)
```



```
dev.off() # clean setting for 9 plots
```

```
## null device
##      1
```

How does aggregation affects the covariates? How does aggregation affects the area of North America?

Project the data downloaded from GBIF

```

# project geographic coordinates (from degrees to metric units)
coordinates(query_gbif) <- c("decimalLongitude", "decimalLatitude")
proj4string(query_gbif) <- CRS("+init=epsg:4326") # GBIF geographic coordinates "+init=epsg:4326 +proj=

query_gbif <- sp::spTransform(query_gbif, crs) # project according to adaptwest raster data
query_gbif <- as.data.frame(query_gbif)

# replace name of columns after projection
names(query_gbif)[names(query_gbif) == 'decimalLongitude'] <- 'x' # changed from degrees to metric
names(query_gbif)[names(query_gbif) == 'decimalLatitude'] <- 'y' # changed from degrees to metric

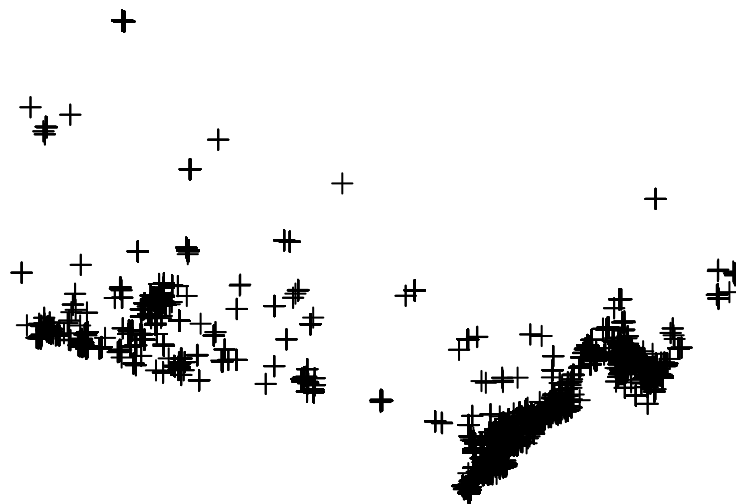
# prepare coordinates and data for spatial dataframe
coordinates_query_gbif <- query_gbif[, c("x", "y")] # coordinates; equivalent to query_gbif[, 49:50]
data_query_gbif <- query_gbif[, c(9:10, 31)] # data you want to keep

# make a SpatialPointsDataFrame object
spdf <- sp::SpatialPointsDataFrame(coords = coordinates_query_gbif,
                                   data = data_query_gbif,
                                   proj4string = crs)

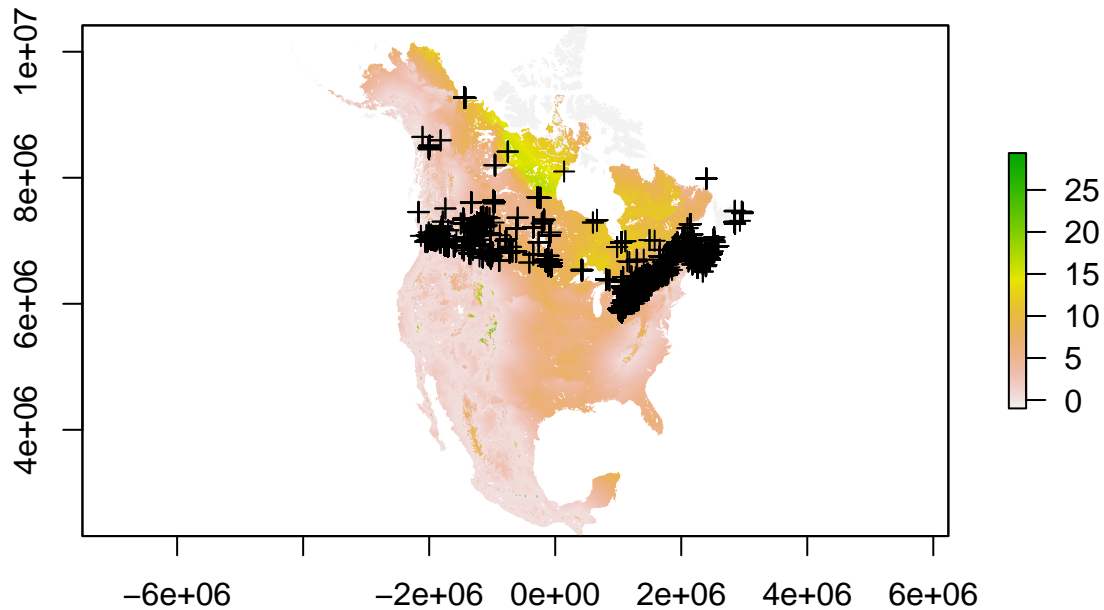
```

Now the coordinates in degrees from GBIF are expressed in meters from the origin of the projected reference system. Plot them over the raster

```
plot(spdf)
```




```
# note that point() does not overlay points properly
# double-check with Laura and Janaina
plot(VEL_ED)
plot(spdf,
      #col = spdf$year,
      add = TRUE)
```



Very weird. Let's check the coordinate reference system

```
crs(VEL_ED)
```

```
## CRS arguments:
## +proj=lcc +lat_0=0 +lon_0=-95 +lat_1=49 +lat_2=77 +x_0=0 +y_0=0
## +datum=WGS84 +units=m +no_defs
```

```
crs(spdf)
```

```
## CRS arguments:
## +proj=lcc +lat_0=0 +lon_0=-95 +lat_1=49 +lat_2=77 +x_0=0 +y_0=0
## +datum=WGS84 +units=m +no_defs
```

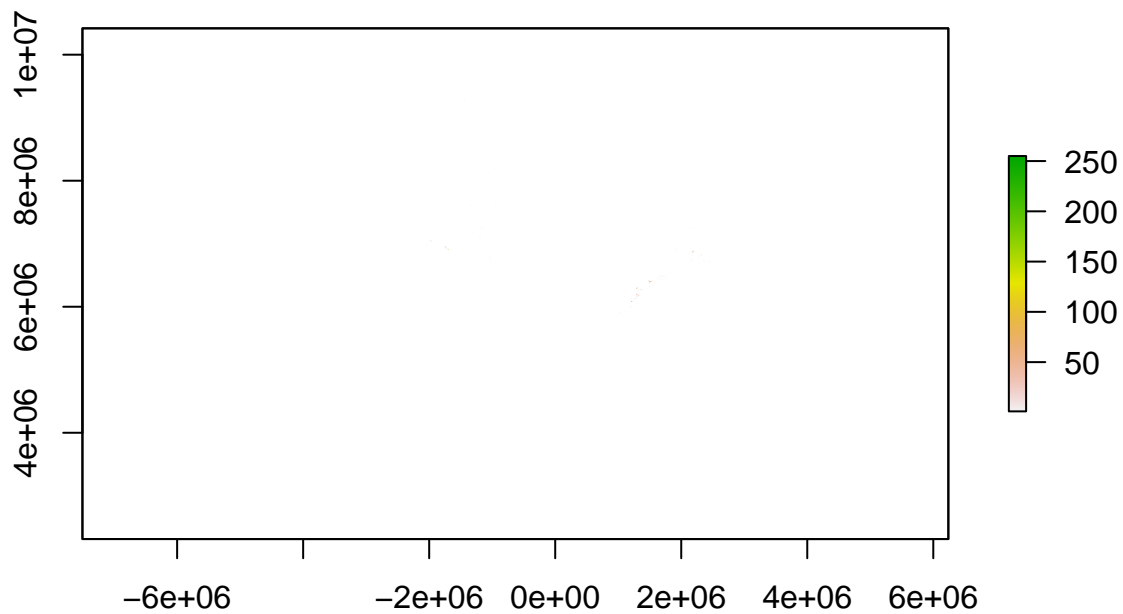
Same crs. This is a very annoying visual glitch in R, but it does not affect point extraction; see <https://stackoverflow.com/questions/28371270/resizing-plot-output-causes-raster-and-points-to-become-misaligned> for a solution

Check the distribution of checklists across Canada

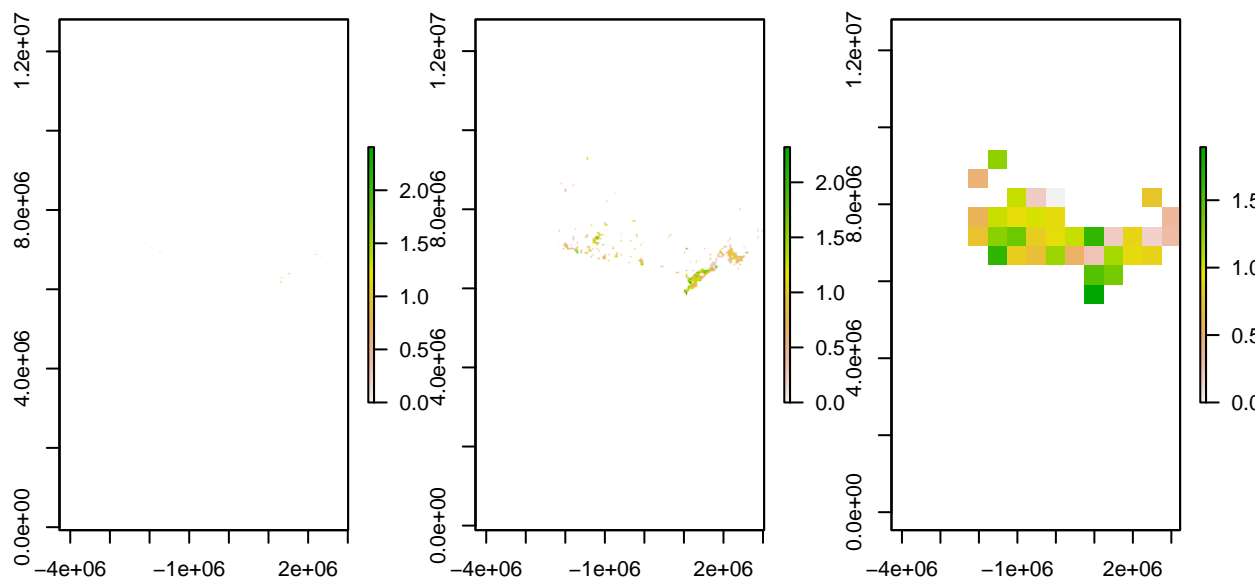
Now that the occurrence data is projected, we can look at its distribution across Canada with more attention. First, we transform points into a raster, counting the number of occurrences in each 5 km cell. Then, we aggregate our spatial units to evaluate the degree to which community science efforts covered Canada

```
# look at how many cells have multiple observations
checklists <- raster::rasterize(coordinates_query_gbif, ID, fun='count')

plot(checklists) # difficult to see, 5 km cells are very small at the continental scale
```



```
par(mfrow=c(1,3))
plot(log10(checklists))
plot(log10(aggregate(checklists,10))) # 50 km resolution
plot(log10(aggregate(checklists,100))) # 500 km resolution
```



The effects of scaling on sampling coverage are substantial.

Last, we extract our covariates using the locations of occurrence of butterflies.

```
# extract covariates
coordinates_query_gbif$extract_ID <- extract(ID, coordinates_query_gbif[, 1:2])
coordinates_query_gbif$extract_ID_50 <- extract(ID_50, coordinates_query_gbif[, 1:2])
coordinates_query_gbif$extract_ID_500 <- extract(ID_500, coordinates_query_gbif[, 1:2])

coordinates_query_gbif$extract_LAT <- extract(LAT, coordinates_query_gbif[, 1:2])
coordinates_query_gbif$extract_LAT_50 <- extract(LAT_50, coordinates_query_gbif[, 1:2])
coordinates_query_gbif$extract_LAT_500 <- extract(LAT_500, coordinates_query_gbif[, 1:2])

coordinates_query_gbif$extract_VEL_ED <- extract(VEL_ED, coordinates_query_gbif[, 1:2])
coordinates_query_gbif$extract_VEL_ED_50 <- extract(VEL_ED_50, coordinates_query_gbif[, 1:2])
coordinates_query_gbif$extract_VEL_ED_500 <- extract(VEL_ED_500, coordinates_query_gbif[, 1:2])

## check relationship between LAT and VEL_ED across scales
## might take a while; select the next three rows and press CTRL + SHIFT + C to run the code
# par(mfrow=c(1,3))
# plot(coordinates_query_gbif$extract_VEL_ED ~ coordinates_query_gbif$extract_LAT)
# plot(coordinates_query_gbif$extract_VEL_ED_50 ~ coordinates_query_gbif$extract_LAT_50)
# plot(coordinates_query_gbif$extract_VEL_ED_500 ~ coordinates_query_gbif$extract_LAT_500)

# create a unique coord field to merge with original data
coordinates_query_gbif$pasted_coord <- paste(coordinates_query_gbif$x, coordinates_query_gbif$y)
query_gbif$pasted_coord <- paste(query_gbif$x, query_gbif$y)
```

Rarity and its relationship to spatial grain

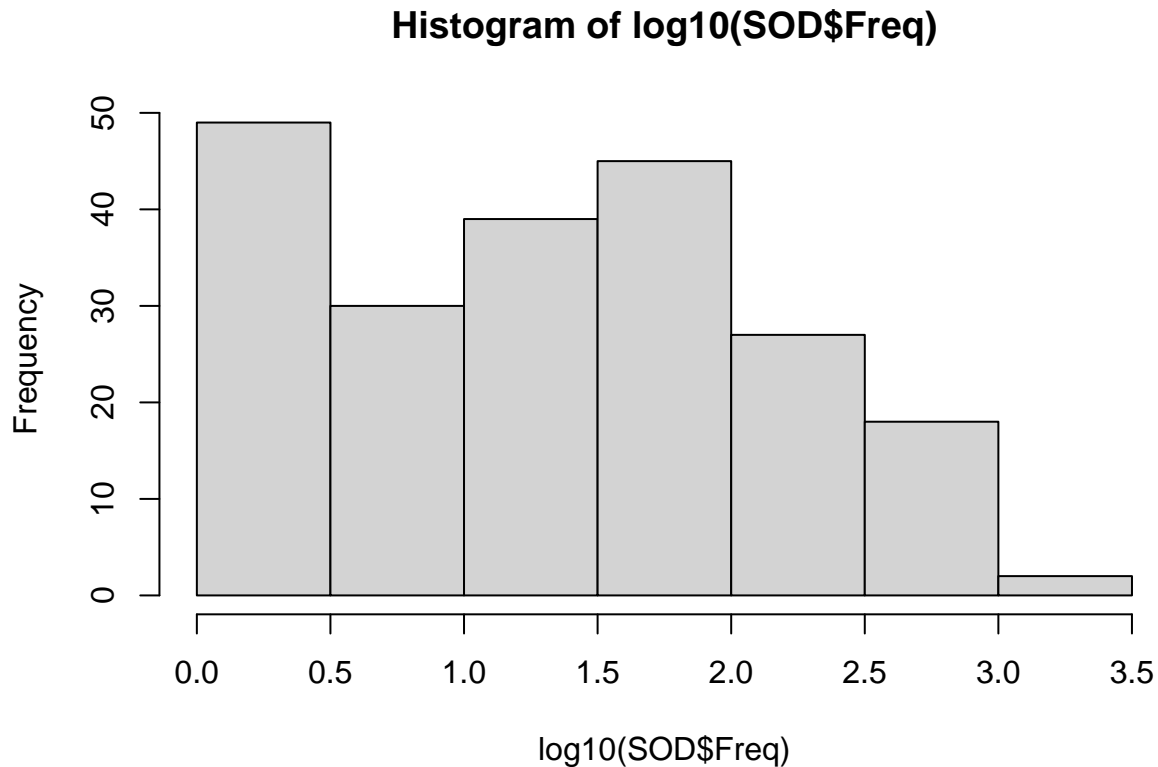
In the second part of the tutorial, we will look at how spatial scales affect the relative rarity of species in a community. We will contrast patterns of rarity in Canada vs. patterns of rarity in three Provinces (Alberta, Ontario and Quebec) based on our entire dataset. Then, we will resample an equal number of occurrences from the four strata, and assess whether effects are due to are or tu sampling.

```
# inspecting the species occurrence distribution
SOD <- as.data.frame(table(query_gbif$species))
head(SOD)
```

```
##           Var1 Freq
## 1   Aglais milberti 205
## 2 Ancyloxypha numitor 311
## 3   Anthocharis sara  40
## 4   Apodemia mormo   1
## 5 Asterocampa celtis  22
## 6 Asterocampa clyton  26
```

```
# as expected the cabbage white (Pieris rapae) is the most common species in these samples (check your
```

```
hist(log10(SOD$Freq))
```



```

# most species have been observed between 10 and 1000 occurrence data

# inspecting the species occurrence distribution at lower spatial levels (Provinces)
## which provinces we have in the dataframe?
levels(as.factor(query_gbif$stateProvince))

## [1] "Alberta"           "British Columbia"
## [3] "Manitoba"          "New Brunswick"
## [5] "Newfoundland and Labrador" "Northwest Territories"
## [7] "Nova Scotia"       "Nunavut"
## [9] "Ontario"           "Prince Edward Island"
## [11] "Quebec"            "Saskatchewan"
## [13] "Yukon"

# provinces
query_gbif_subset_SOD_AB <- query_gbif[which(query_gbif$stateProvince == "Alberta"), ]
query_gbif_subset_SOD_QC <- query_gbif[which(query_gbif$stateProvince == "Quebec"), ]
query_gbif_subset_SOD_ON <- query_gbif[which(query_gbif$stateProvince == "Ontario"), ]

SOD_AB <- as.data.frame(table(query_gbif_subset_SOD_AB$species))
SOD_QC <- as.data.frame(table(query_gbif_subset_SOD_QC$species))
SOD_ON <- as.data.frame(table(query_gbif_subset_SOD_ON$species))

## divide by the total number of individuals in each dataset, so as to transform counts into proportion
SOD$Freq <- SOD$Freq/sum(SOD$Freq)
SOD_AB$Freq <- SOD_AB$Freq/sum(SOD_AB$Freq)
SOD_QC$Freq <- SOD_QC$Freq/sum(SOD_QC$Freq)
SOD_ON$Freq <- SOD_ON$Freq/sum(SOD_ON$Freq)

# plot the curves
plot(SOD$Freq[order(SOD$Freq)], xaxt='n', frame.plot=TRUE,
     main="Species-occurrence ranking plot",
     ylab="Frequency of occurrence", xlab="Species ranked by frequency of occurrence", pch = 16,
     ylim = c(0, 0.10))

par(new=TRUE)
plot(SOD_AB$Freq[order(SOD_AB$Freq)], col = "red", axes=FALSE, frame.plot=TRUE,
     ylab="", xlab="", pch = 16,
     ylim = c(0, 0.10))

par(new=TRUE)
plot(SOD_QC$Freq[order(SOD_QC$Freq)], col = "blue", axes=FALSE, frame.plot=TRUE,
     ylab="", xlab="", pch = 16,
     ylim = c(0, 0.10))

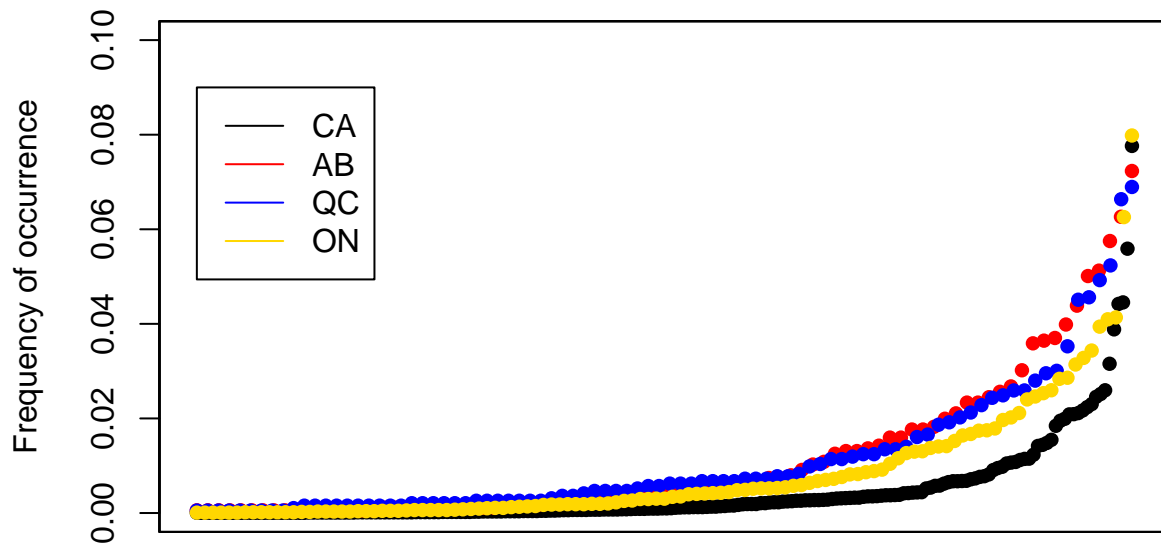
par(new=TRUE)
plot(SOD_ON$Freq[order(SOD_ON$Freq)], col = "gold", axes=FALSE, frame.plot=TRUE,
     ylab="", xlab="", pch = 16,
     ylim = c(0, 0.10))

legend(1, 0.09, # position of the legend; the x axis goes from 0 to 0.1 (10% Of the occurrences observed)
      legend=c("CA", "AB", "QC", "ON"),

```

```
col=c("black", "red", "blue", "gold"), lty=1, cex=1)
```

Species–occurrence ranking plot



Species ranked by frequency of occurrence

These are typical SODs, where many species are rare and a handful tend to dominate the community. What do we infer from this pattern? It could be that, when considering larger spatial scales, there is a higher dominance on the SOD of very common species (black line always under colored lines). However, it could be that differences in sampling effort across region dictate this pattern. Therefore, we repeat the analysis randomly selecting the same number of individuals in each dataset.

```
# how many observations in different datasets?  
nrow(query_gbif)
```

```
## [1] 21578
```

```
nrow(query_gbif_subset_SOD_AB)
```

```
## [1] 1756
```

```
nrow(query_gbif_subset_SOD_QC)
```

```
## [1] 1929
```

```
nrow(query_gbif_subset_SOD_ON)
```

```
## [1] 12865
```

```
# let's simulate 1700 individuals, the limiting factor being the spatial unit with the lowest number of
SOD <- query_gbif[sample(nrow(query_gbif), 1700), ]
query_gbif_subset_SOD_AB <- query_gbif_subset_SOD_AB[sample(nrow(query_gbif_subset_SOD_AB), 1700), ]
query_gbif_subset_SOD_QC <- query_gbif_subset_SOD_QC[sample(nrow(query_gbif_subset_SOD_QC), 1700), ]
query_gbif_subset_SOD_ON <- query_gbif_subset_SOD_ON[sample(nrow(query_gbif_subset_SOD_ON), 1700), ]

SOD <- as.data.frame(table(query_gbif$species))
SOD_AB <- as.data.frame(table(query_gbif_subset_SOD_AB$species))
SOD_QC <- as.data.frame(table(query_gbif_subset_SOD_QC$species))
SOD_ON <- as.data.frame(table(query_gbif_subset_SOD_ON$species))

##
SOD$Freq <- SOD$Freq/sum(SOD$Freq) # repeat for all other SODs
SOD_AB$Freq <- SOD_AB$Freq/sum(SOD_AB$Freq) # repeat for all other SODs
SOD_QC$Freq <- SOD_QC$Freq/sum(SOD_QC$Freq) # repeat for all other SODs
SOD_ON$Freq <- SOD_ON$Freq/sum(SOD_ON$Freq) # repeat for all other SODs

#par(mfrow=c(1,5))
plot(SOD$Freq[order(SOD$Freq)], xaxt='n', frame.plot=TRUE,
     main="Species-occurrence ranking plot",
     ylab="Frequency of occurrence", xlab="Species ranked by frequency of occurrence", pch = 16,
     ylim = c(0, 0.10))

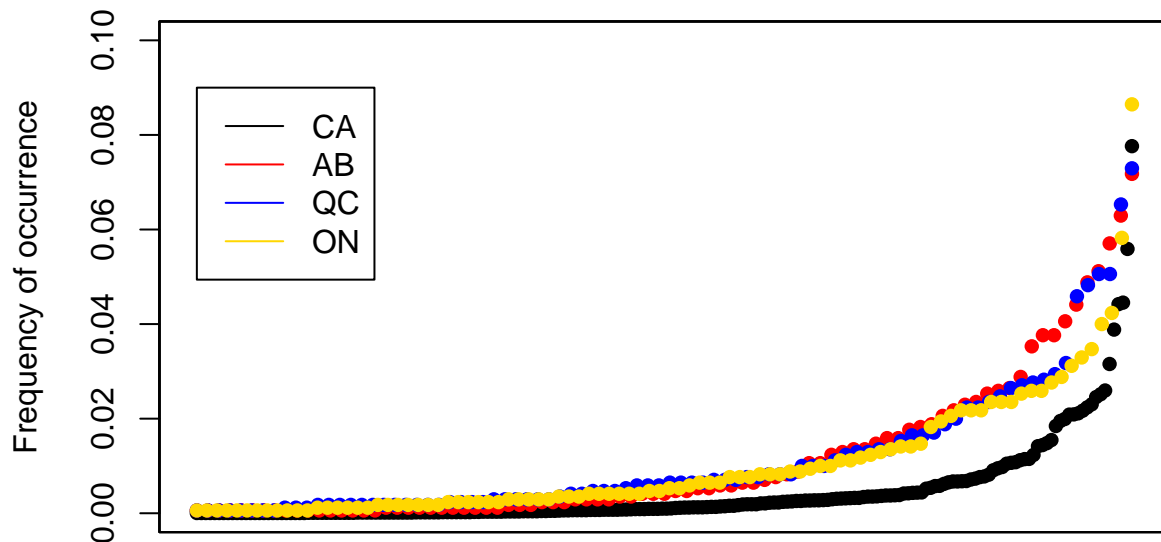
par(new=TRUE)
plot(SOD_AB$Freq[order(SOD_AB$Freq)], col = "red", axes=FALSE, frame.plot=TRUE,
     ylab="", xlab="", pch = 16,
     ylim = c(0, 0.10))

par(new=TRUE)
plot(SOD_QC$Freq[order(SOD_QC$Freq)], col = "blue", axes=FALSE, frame.plot=TRUE,
     ylab="", xlab="", pch = 16,
     ylim = c(0, 0.10))

par(new=TRUE)
plot(SOD_ON$Freq[order(SOD_ON$Freq)], col = "gold", axes=FALSE, frame.plot=TRUE,
     ylab="", xlab="", pch = 16,
     ylim = c(0, 0.10))

legend(1, 0.09, # position of the legend; the x axis goes from 0 to 0.1 (10% Of the occurrences observe
      legend=c("CA", "AB", "QC", "ON"),
      col=c("black", "red", "blue", "gold"), lty=1, cex=1)
```

Species–occurrence ranking plot



Species ranked by frequency of occurrence

The pattern is consistent even when accounting for sampling effort; the hypothesized area effects on the species frequency occurrence rank plot are supported by our dataset.

Part 3: Spatiotemporal trends in biodiversity

I prepared a table where the 262 butterfly species of Canada are recorded in ~ 30,000 checklists between 2010 and 2021. For each checklist, I also provided spatial and temporal information (columns 262-280).

```
final_table <- data.table::fread("Data\\final_table.csv", header = TRUE)
head(final_table$`Polygonia comma`)
```

```
## [1] 0 0 0 0 0 0
```

```
head(final_table$extract_LAT)
```

```
## [1] 6175500 6925500 6925500 6170500 7245500 6835500
```

```
head(final_table$year)
```

```
## [1] 2010 2010 2010 2010 2010 2010
```

In the next lines, I created a script that (i) breaks the 30,000 rows table into multiple lists, each containing all the observations occurred at the same site. With “site” we used the “extract_ID” field representing each of the 5-km cell in which checklist were observed earlier. You can choose a different scale by replacing “extract_ID” with “extract_ID_50” or “extract_ID_500”.


```
## split into the 5-km cell communities
split_5 <- split(final_table, final_table$extract_ID)
# 4674 sites with at least one checklist
```

Then, I removed all sites with less than 10 checklists in 12 years

```
# we keep only sites with at least 10 checklists
split_5 <- Filter(function(x) nrow(x) > 10, split_5)
# 465 sites
```

Then, we retained only years with at least 5 checklists, and removed sites with less than two years

```
# how many transects per year
## split each cell in the 12 years
split_5 <- lapply(split_5, function(x) split(x, x$year))
for(i in 1:length(split_5)){
  split_5[[i]] <- Filter(function(x) nrow(x) > 5, split_5[[i]])
}

# remove sites with less than 2 years
split_5 <- Filter(function(x) length(x) > 2, split_5) # we assume that visits with only one observation.
```

To account for differences in sampling effort, we resample the same number of checklists in each year in each site. We also remove sites with less than 3 years of data.

```
## randomly select in each site an equal number of checklists
for(i in 1:length(split_5)){
  for(j in 1:length(split_5[[i]])){
    split_5[[i]][[j]] <- split_5[[i]][[j]][sample(nrow(split_5[[i]][[j]]), 5, replace = FALSE), ]
  }
}

# remove sites with only one or two years
split_5 <- Filter(function(x) length(x) > 2, split_5) # we assume that visits with only one observation.
```

Now the data is ready to be analyzed. In the next section, we calculate species richness in each of the years for which we had at least 5 checklists, in each of the sites. I also calculated the average Jaccard similarity (https://en.wikipedia.org/wiki/Jaccard_index) across years in a site

```
## calculate species richness
split_5_richness <- split_5
for(i in 1:length(split_5_richness)){
  for(j in 1:length(split_5_richness[[i]])){
    split_5_richness[[i]][[j]] <- sum(colSums(split_5[[i]][[j]][,1:262]) != 0) # the sum of all columns
  }
}

for(i in 1:length(split_5_richness)){
  split_5_richness[[i]] <- do.call(rbind.data.frame, split_5_richness[[i]])
}

# community table
```

```

split_5_comm <- split_5
for(i in 1:length(split_5_comm)){
  for(j in 1:length(split_5_comm[[i]])){
    split_5_comm[[i]][[j]] <- as.data.frame(t(colSums(split_5_comm[[i]][[j]][,1:262])))
    split_5_comm[[i]][[j]][split_5_comm[[i]][[j]] > 0] <- 1
  }
}

for(i in 1:length(split_5_comm)){
  split_5_comm[[i]] <- do.call(rbind.data.frame, split_5_comm[[i]])
}

# calculate the mean Jaccard similarity across the species found in a site in different years
split_5_similarity <- split_5_comm
for(i in 1:length(split_5_similarity)){
  split_5_similarity[[i]] <- mean(vegdist(split_5_similarity[[i]], method = "jaccard"))
}

## years relative to each list
years <- list()
for(i in 1:length(split_5)){
  years[[i]] <- as.numeric(names(split_5[[i]]))
}

```

Last, we fit some models predicting trends in species richness at each site

```

# trends in richness
models <- list()
for(i in 1:length(years)){
  models[[i]] <- lm(split_5_richness[[i]][,1] ~ years[[i]]) # model where richness at a site is predict
  models[[i]] <- models[[i]]$coefficients[2] # keep only the year effect
}

# for(i in 1:length(years)){
#   models[[i]] <- models[[i]]$coefficients[2]
# }

# velocity
split_5_velocity <- split_5

for(i in 1:length(split_5_velocity)){
  split_5_velocity[[i]] <- split_5_velocity[[i]][[1]]$extract_VEL_ED[[1]]
}

split_5_latitude <- split_5

for(i in 1:length(split_5_latitude)){
  split_5_latitude[[i]] <- split_5_latitude[[i]][[1]]$extract_LAT[[1]]
}

# clean the datasets
models <- do.call(rbind.data.frame, models)
colnames(models)[1] <- "slope"

```

```

similarity <-do.call(rbind.data.frame, split_5_similarity)
colnames(similarity)[1] <- "similarity"
velocity <- do.call(rbind.data.frame, split_5_velocity)
colnames(velocity)[1] <- "velocity"
latitude <- do.call(rbind.data.frame, split_5_latitude)
colnames(latitude)[1] <- "latitude"

summary(lm(models$slope ~ latitude$latitude))

##
## Call:
## lm(formula = models$slope ~ latitude$latitude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.7835 -1.0936 -0.2601  0.7831  5.1897
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.837e+00  2.950e+00  -1.640   0.1038
## latitude$latitude  7.778e-07  4.610e-07   1.687   0.0943 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.677 on 113 degrees of freedom
## Multiple R-squared:  0.02457,    Adjusted R-squared:  0.01594
## F-statistic: 2.847 on 1 and 113 DF,  p-value: 0.09431

# # check other relationships
# plot(models$slope ~ velocity$velocity)
#
# plot(similarity$similarity ~ latitude$latitude)
# plot(models$slope ~ similarity$similarity)

```

The relationship between richness changes at a site (“slope”) and latitude tend to be positive, and at time significant. Because the resampling of checklist in every site is stochastic, you will find results different from your classmates.

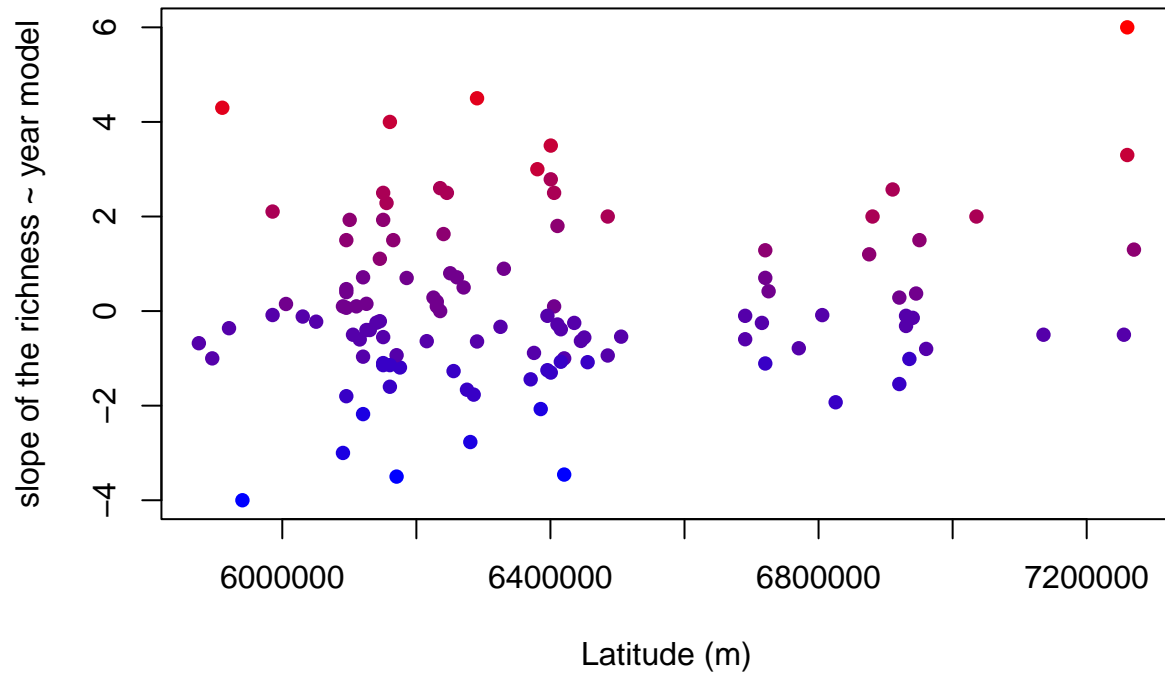
```

#Create a function to generate a continuous color palette
rbPal <- colorRampPalette(c('blue','red'))

#This adds a column of color values
# based on the y values
models$Col <- rbPal(10)[as.numeric(cut(models$slope,breaks = 10))]
plot(models$slope ~ latitude$latitude,
     col = models$Col,
     pch = 16,
     main="Relationship between biodiversity trends and latitude",
     xlab="Latitude (m)", ylab="slope of the richness ~ year model")

```

Relationship between biodiversity trends and latitude



You can see positive relationships between richness and time in red, and negative in blue. It seems like many sites are characterized by a loss of diversity, except for a few northern sites that seem to have positive diversity trends.