Marine Spatial PLanning

Moegamad Uzair Jack

2025-06-02

## Loading packages

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(ggpubr)  
library(vegan)

## Loading required package: permute  
## Loading required package: lattice  
## This is vegan 2.6-6.1

## Reading File

Reading csv as a data frame

EF <- read.csv('EstuaryFishSurveyData.csv')  
class(EF)

## [1] "data.frame"

Replace NA values with 0

EF[is.na(EF)] <- 0

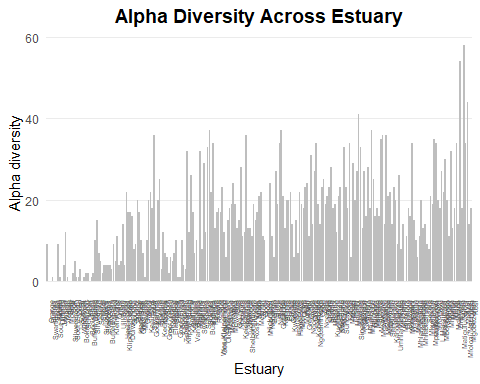
## Plot Species Number

Creating a separate column

EF$alpha <- specnumber(EF[4: 148])

Plotting alpha diversity (species number) per estuary

EF$Estuary <- factor(EF$Estuary, levels = unique(EF$Estuary)) # Ensure Estuary is a factor for ordered x-axis labels  
  
g1 <- ggplot(EF, aes(x = Estuary, y = alpha)) +  
 geom\_bar(stat = "identity", fill = "grey") +  
 theme\_minimal() +  
 labs(  
 title = "Alpha Diversity Across Estuary",  
 x = "Estuary",  
 y = "Alpha diversity"  
 ) +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1, size = 5),  
 plot.title = element\_text(size = 14, face = "bold", hjust = 0.5),  
 panel.grid.major.x = element\_blank(),  
 panel.grid.minor = element\_blank()  
 )  
g1



## Calculating Diversity

EF$SW <- diversity(EF[4:148], index = "shannon")

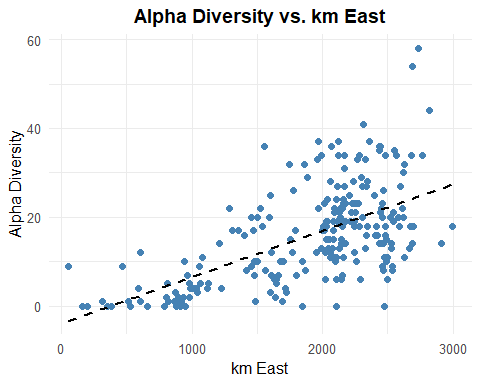
## Creating kmEast column

EF$kmEast <- 3000 - EF$kmWest

## Plotting alpha diveristy vs. km East

g2 <- ggplot(EF, aes(x = kmEast, y = alpha)) +  
 geom\_point(color = "steelblue", size = 2) +  
 geom\_smooth(method = "lm", se = FALSE, color = "black", linetype = "dashed") +  
 theme\_minimal(base\_size = 12) +  
 labs(  
 title = "Alpha Diversity vs. km East",  
 x = "km East",  
 y = "Alpha Diversity"  
 ) +  
 theme(  
 plot.title = element\_text(face = "bold", hjust = 0.5),  
 axis.title = element\_text(size = 12),  
 axis.text = element\_text(size = 10)  
 )  
g2

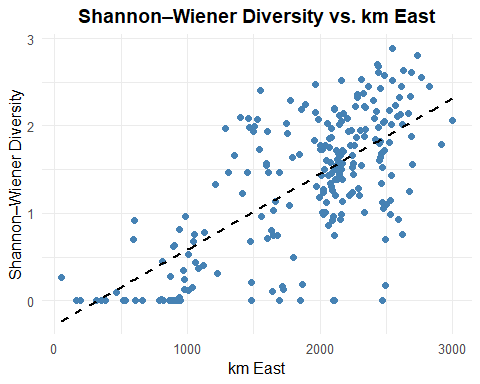
## `geom\_smooth()` using formula = 'y ~ x'



## Plotting Shannon–Wiener Diversity vs. km East

g3 <- ggplot(EF, aes(x = kmEast, y = SW)) +  
 geom\_point(color = "steelblue", size = 2) +  
 geom\_smooth(method = "lm", se = FALSE, color = "black", linetype = "dashed") +  
 theme\_minimal(base\_size = 12) +  
 labs(  
 title = "Shannon–Wiener Diversity vs. km East",  
 x = "km East",  
 y = "Shannon–Wiener Diversity"  
 ) +  
 theme(  
 plot.title = element\_text(face = "bold", hjust = 0.5),  
 axis.title = element\_text(size = 12),  
 axis.text = element\_text(size = 10)  
 )  
g3

## `geom\_smooth()` using formula = 'y ~ x'



## Reorder data frame

EF <- EF[order(-EF$alpha), ]

## Creating new variable

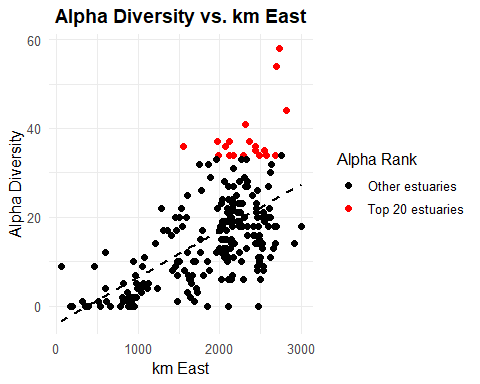
EF$AlphaList <- 0  
EF$AlphaList[1:20] <- 1

## Plot top 20 in RED and the rest in BLACK

Plot the alpha diversity against km East with colour factor based on the alpha rank. Here the top 20 are represented in red and the rest in black.

g4 <- ggplot(EF, aes(x = kmEast, y = alpha)) +  
 geom\_point(aes(color = factor(AlphaList)), size = 2) +  
 geom\_smooth(method = "lm", se = FALSE, color = "black", linetype = "dashed") +  
 scale\_color\_manual(  
 values = c("0" = "black", "1" = "red"),  
 labels = c("Other estuaries", "Top 20 estuaries"),  
 name = "Alpha Rank"  
 ) +  
 theme\_minimal(base\_size = 12) +  
 labs(  
 title = "Alpha Diversity vs. km East",  
 x = "km East",  
 y = "Alpha Diversity"  
 ) +  
 theme(  
 plot.title = element\_text(face = "bold", hjust = 0.5),  
 axis.title = element\_text(size = 12),  
 axis.text = element\_text(size = 10),  
 legend.position = "right"  
 )  
  
g4

## `geom\_smooth()` using formula = 'y ~ x'



## Alpha Method

Now we need to list the top 20 esturaries and save it into a variable

AlphaList <- EF[1:20, 1]  
AlphaList

## [1] Mlalazi Matigulu/Nyoni St Lucia Mngazana Kariega   
## [6] Kwelera Mntafufu Knysna Tyolomnqa Mzamba   
## [11] Mtamvuna Mtentu Mkomazi Kowie Gqunube   
## [16] Ngqusi/Inxaxo Mtata Mzimkulu Lovu Zinkwasi   
## 231 Levels: Orange Buffels Swartlintjies Spoeg Groen Sout (North) ... Kosi

We want the list and the number of species. 0 represents the species not in the list; 1 represents the species that is in the list.

AlphaSpp <- specnumber(EF[,4:148] , EF$AlphaList)  
AlphaSpp

## 0 1   
## 121 116

There are 116 species in the list and 145 species in total. So, we calculate the percentage of species within the list.

AlphaPerc <- (116/145) \*100  
AlphaPerc

## [1] 80

## Change the metric

Now do the same for the SW index. Start by reordering the dataframe

EF <- EF[order(-EF$SW), ]

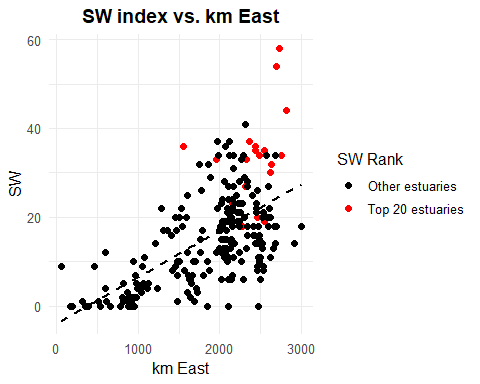
Creating new variable

EF$SW\_list <- 0  
EF$SW\_list[1:20] <- 1

Plot top 20 in RED and the rest in BLACK Plot the SW against km East with colour factor based on the SW rank. Here the top 20 are represented in red and the rest in black. Keep alpha to compare between the 2.

g5 <- ggplot(EF, aes(x = kmEast, y = alpha)) +  
 geom\_point(aes(color = factor(SW\_list)), size = 2) +  
 geom\_smooth(method = "lm", se = FALSE, color = "black", linetype = "dashed") +  
 scale\_color\_manual(  
 values = c("0" = "black", "1" = "red"),  
 labels = c("Other estuaries", "Top 20 estuaries"),  
 name = "SW Rank"  
 ) +  
 theme\_minimal(base\_size = 12) +  
 labs(  
 title = "SW index vs. km East",  
 x = "km East",  
 y = "SW"  
 ) +  
 theme(  
 plot.title = element\_text(face = "bold", hjust = 0.5),  
 axis.title = element\_text(size = 12),  
 axis.text = element\_text(size = 10),  
 legend.position = "right"  
 )  
  
g5

## `geom\_smooth()` using formula = 'y ~ x'

 ## SW Method Now we need to list the top 20 esturaries and save it into a variable

SW\_list <- EF[1:20, 1]  
SW\_list

## [1] Mkomazi Mlalazi Mtentu Mzamba   
## [5] Mhlali Matigulu/Nyoni Mtamvuna Mzimkulu   
## [9] Mfolozi/Msunduzi Sinangwana Great Kei Mahlongwa   
## [13] Bushmans Mgeni Mntafufu St Lucia   
## [17] Umhlangankulu Knysna Mngazi Mapuzi   
## 231 Levels: Orange Buffels Swartlintjies Spoeg Groen Sout (North) ... Kosi

We want the list and the number of species. 0 represents the species not in the list; 1 represents the species that is in the list.

SW\_spp <- specnumber(EF[,4:148] , EF$SW\_list)  
SW\_spp

## 0 1   
## 125 110

There are 110 species in the list and 145 species in total. So, we calculate the percentage of species within the list.

SW\_perc <- (110/145) \*100  
SW\_perc

## [1] 75.86207

## Biogeographic Zonation

Use specnumber to calculate the species number per biogeographic zone (BZ)

specnumber(EF[,4:148], EF$BZ)

## E S W   
## 111 79 21

In each of the 3 zones, choose the most diverse estuary in each zone. Choose 6 estuaries from each zone (2 spare). 6 from W; 7 from S; 7 from E. Start ordering the data frame by BZ then by alpha.

EF <- EF[order(EF$BZ, -EF$alpha), ]  
head(EF)

## Estuary kmWest BZ Spot.damsel Estuarine.bream Bald.glassy  
## 228 Mlalazi 267 E 0 25 375  
## 226 Matigulu/Nyoni 308 E 0 25 517  
## 230 St Lucia 179 E 0 4 437  
## 171 Mngazana 685 E 0 1 8  
## 178 Mntafufu 638 E 0 2 126  
## 184 Mzamba 564 E 0 9 0  
## Slender.glassy Longspine.glassy Evileye.blaasop Longfin.eel Striped.angler  
## 228 69 8 3 0 0  
## 226 16 11 0 0 0  
## 230 8 1 0 0 0  
## 171 1 0 0 0 0  
## 178 0 0 9 0 0  
## 184 11 2 2 0 0  
## Silver.kob Dusky.kob Blackedged.blaasop Cape.silverside Freshwater.goby  
## 228 0 3 2 0 0  
## 226 0 11 2 0 0  
## 230 0 1 0 0 0  
## 171 0 10 0 63 0  
## 178 0 5 0 0 0  
## 184 0 1 0 0 0  
## Smallmouth.yellowfish Scaly Prison.goby Baldy Barehead.goby Giant.kingfish  
## 228 0 0 0 0 0 40  
## 226 0 0 0 0 0 22  
## 230 0 0 0 0 0 4  
## 171 0 0 14 0 0 2  
## 178 0 0 133 0 0 39  
## 184 0 0 0 0 0 7  
## Blacktip.kingfish Brassy.kingfish Bigeye.kingfish Kingfish  
## 228 0 1 13 0  
## 226 0 0 11 0  
## 230 0 0 0 0  
## 171 1 2 6 0  
## 178 0 0 25 0  
## 184 0 0 23 0  
## Doubledash.butterflyfish Milkfish Cape.gurnard Bluespotted.blaasop  
## 228 0 0 0 14  
## 226 0 0 0 0  
## 230 0 0 0 0  
## 171 0 0 0 0  
## 178 0 0 0 1  
## 184 0 0 0 1  
## Sharptooth.catfish Estuary.klipfish Super.klipfish Karanteen  
## 228 4 0 0 0  
## 226 3 0 0 0  
## 230 0 0 0 1  
## 171 0 0 0 0  
## 178 0 0 0 0  
## 184 0 0 0 0  
## Fringelip.mullet Carp Bluespotted.stingray Zebra Blacktail Dusky.sleeper  
## 228 0 0 0 0 0 0  
## 226 0 0 0 0 0 1  
## 230 0 0 0 0 0 0  
## 171 0 0 0 0 0 0  
## 178 0 0 0 0 0 0  
## 184 0 0 0 0 0 0  
## Ladyfish Anchovies Cape.anchovy Orangespotted.rockcod Malabar.rockcod  
## 228 4 0 23 0 1  
## 226 4 0 0 0 2  
## 230 1 0 1 1 0  
## 171 18 0 0 0 0  
## 178 4 0 0 0 0  
## 184 6 0 0 0 0  
## Greasy.rockcod Redeye.roundherring Spotted.ragged.tooth Bblackthroat.goby  
## 228 0 0 0 0  
## 226 0 0 0 0  
## 230 1 0 0 0  
## 171 0 0 0 0  
## 178 0 0 0 0  
## 184 0 0 0 0  
## Spotted.sandgoby Cape.galaxias White.seacatfish Mosquitofish  
## 228 0 0 0 0  
## 226 0 0 0 0  
## 230 0 0 0 0  
## 171 0 0 0 0  
## 178 0 0 0 0  
## 184 1 0 0 0  
## Smallscale.pursemouth Longspine.pursemouth Evenfin.pursemouth  
## 228 12 2 64  
## 226 0 0 64  
## 230 0 0 5  
## 171 0 0 0  
## 178 0 0 0  
## 184 0 0 4  
## Oblong.pursemouth Estuarine.roundherring Sleepy.goby River.goby Tank.goby  
## 228 0 3 0 2 10  
## 226 0 4 0 4 1  
## 230 0 14 5 15 1  
## 171 0 1834 0 22 2  
## 178 0 367 0 61 0  
## 184 0 92 0 12 0  
## Gobies Janbruin Dark.shyshark Spotted.halfbeak Cape.sole Kelee.shad  
## 228 0 0 0 0 0 49  
## 226 0 0 0 0 0 32  
## 230 0 0 0 0 0 78  
## 171 0 0 0 0 0 15  
## 178 0 0 0 0 0 3  
## 184 0 0 0 0 0 4  
## Belly.pipefish Bellybarred.pipefish Knysna.seahorse Small.kob Slimy  
## 228 0 0 0 0 149  
## 226 1 0 0 0 206  
## 230 0 0 0 1 65  
## 171 0 0 0 0 7  
## 178 0 1 0 0 2  
## 184 0 0 0 0 1  
## Bluegill.sunfish Garrick White.steenbras Sand.steenbras Diamond.mullet  
## 228 0 0 0 0 16  
## 226 0 0 0 0 4  
## 230 0 0 0 0 0  
## 171 0 3 1 0 3  
## 178 0 1 0 0 4  
## 184 0 0 0 0 1  
## Groovy.mullet Largescale.mullet Giantscale.mullet Southern.mullet Mullet  
## 228 76 58 1 0 6  
## 226 198 46 1 0 19  
## 230 41 47 0 0 7  
## 171 115 32 0 0 188  
## 178 211 36 0 0 107  
## 184 103 69 0 0 13  
## Striped.mullet Mangrove.snapper Dory.snapper Oxeye.tarpon Spotted.bass  
## 228 1 2 0 5 0  
## 226 1 2 1 3 0  
## 230 0 0 0 2 0  
## 171 52 1 0 0 0  
## 178 1 2 0 0 0  
## 184 0 1 0 0 0  
## Largemouth.bass Round.moony Oval.moony Flathead.mullet Mullet.1 Durban.goby  
## 228 0 1 1 12 23 0  
## 226 0 3 0 16 5 0  
## 230 0 0 0 1 12 0  
## 171 0 0 1 15 23 0  
## 178 0 0 0 12 5 0  
## 184 0 0 2 9 3 0  
## Eagleray Freshwater.mullet Sharptail.goby Speartail.goby Kappie.blenny  
## 228 0 1 2 15 0  
## 226 0 1 2 6 0  
## 230 0 0 148 19 0  
## 171 0 25 0 2 0  
## 178 0 7 2 10 0  
## 184 0 8 3 1 0  
## Rainbow.trout Mozambique.tilapia Eyebrow.goby Mud.blenny African.mudskipper  
## 228 0 9 0 0 1  
## 226 0 16 0 0 1  
## 230 0 0 40 0 0  
## 171 0 0 0 0 0  
## 178 0 0 0 0 0  
## 184 0 0 0 0 0  
## Bartail.flathead Guppy Striped.threadfin Spotted.grunter Javelin.grunter  
## 228 12 0 0 32 24  
## 226 4 0 0 120 12  
## 230 1 0 0 6 16  
## 171 0 0 1 14 1  
## 178 0 0 0 45 0  
## 184 0 0 0 14 0  
## Cock.grunter Piggy Elf Speckled.sandgoby Southern.mouthbrooder  
## 228 0 0 2 0 0  
## 226 0 0 0 0 0  
## 230 9 0 0 0 0  
## 171 0 0 6 1 0  
## 178 0 0 0 2 0  
## 184 0 0 0 0 0  
## Largetooth.flounder Twineye.skate Checked.goby White.stumpnose  
## 228 0 0 0 0  
## 226 8 0 0 0  
## 230 0 0 0 0  
## 171 0 0 0 0  
## 178 0 0 0 0  
## 184 0 0 0 0  
## Cape.stumpnose Tropical.stumpnose Pilchard Strepie Doublespotted.queenfish  
## 228 2 1 0 0 5  
## 226 17 3 0 0 13  
## 230 4 0 0 0 0  
## 171 181 0 0 0 0  
## 178 186 0 0 0 1  
## 184 42 0 0 0 66  
## Pugnose.soapy Whitespotted.rabbitfish Barebreast.goby Silver.sillago  
## 228 0 0 0 10  
## 226 0 0 7 2  
## 230 0 0 0 1  
## 171 0 0 0 0  
## 178 0 0 0 0  
## 184 0 0 0 0  
## Blackhand.sole Pickhandle.barracuda Thorny.anchovy Longsnout.pipefish  
## 228 3 1 16 0  
## 226 5 3 0 0  
## 230 9 1 18 0  
## 171 0 0 0 0  
## 178 9 0 0 0  
## 184 3 0 1 0  
## Estuarine.pipefish Thornfish Longjaw.glassnose Orangemouth.glassnose  
## 228 0 11 1 59  
## 226 0 9 2 4  
## 230 0 15 3 132  
## 171 0 9 0 33  
## 178 0 14 0 0  
## 184 0 18 0 0  
## Slender.giant.moray Redbreast.tilapia Banded.tilapia  
## 228 0 0 0  
## 226 0 2 0  
## 230 3 0 0  
## 171 0 0 0  
## 178 0 0 0  
## 184 0 0 0  
## Blackspotted.electric.ray Marbled.electric.ray Pompano Maasbanker Comb.goby  
## 228 0 0 0 0 0  
## 226 0 0 0 0 0  
## 230 0 0 0 0 1  
## 171 0 0 0 0 0  
## 178 0 0 0 0 0  
## 184 0 0 0 0 0  
## Yellowbanded.goatfish Bluetail.mullet Longarm.mullet Robust.mullet  
## 228 0 14 353 3  
## 226 0 18 104 0  
## 230 0 7 31 0  
## 171 0 11 8 26  
## 178 0 4 7 2  
## 184 3 6 3 6  
## Bluespot.mullet Mullet.2 alpha SW kmEast AlphaList SW\_list  
## 228 0 5 58 2.800256 2733 1 1  
## 226 0 75 54 2.614774 2692 1 1  
## 230 0 0 44 2.445971 2821 1 1  
## 171 0 2 41 1.543088 2315 1 0  
## 178 1 5 37 2.451720 2362 1 1  
## 184 0 0 36 2.675707 2436 1 1

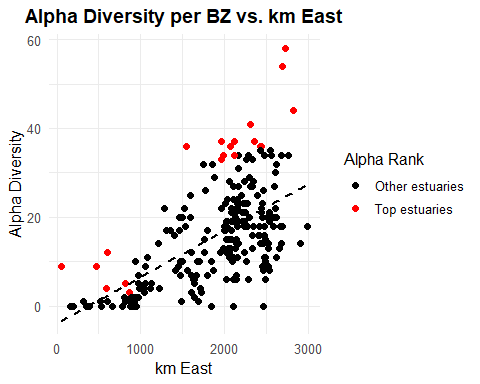
Creating a list

EF$BZ\_list <- 0  
  
top\_n <- c(W = 6, S = 7, E = 7)  
  
for (z in names(top\_n)) {  
 zone\_rows <- which(EF$BZ == z)  
 EF$BZ\_list[head(zone\_rows, top\_n[z])] <- 1  
}

Creating the plot

g6 <- ggplot(EF, aes(x = kmEast, y = alpha)) +  
 geom\_point(aes(color = factor(BZ\_list)), size = 2) +  
 geom\_smooth(method = "lm", se = FALSE, color = "black", linetype = "dashed") +  
 scale\_color\_manual(  
 values = c("0" = "black", "1" = "red"),  
 labels = c("Other estuaries", "Top estuaries"),  
 name = "Alpha Rank"  
 ) +  
 theme\_minimal(base\_size = 12) +  
 labs(  
 title = "Alpha Diversity per BZ vs. km East",  
 x = "km East",  
 y = "Alpha Diversity"  
 ) +  
 theme(  
 plot.title = element\_text(face = "bold", hjust = 0.5),  
 axis.title = element\_text(size = 12),  
 axis.text = element\_text(size = 10),  
 legend.position = "right"  
 )  
  
g6

## `geom\_smooth()` using formula = 'y ~ x'

 We want the list and the number of species. 0 represents the species not in the list; 1 represents the species that is in the list.

BZ\_spp <- specnumber(EF[, 4:148], EF$BZ\_list)  
BZ\_spp

## 0 1   
## 120 122

There are 122 species in the list and 145 species in total. So, we calculate the percentage of species within the list.

BZ\_perc <- (122/145) \*100  
BZ\_perc

## [1] 84.13793

## Dendrogam

We have things called biogeographical zones, each with their own sets of species. When 2 zones come together (overlap), that is where you get the most species. But that is not good to protect. The edges have more diversity but we need to protect the centre. The edges have high diversity and are not good for protection because they are a mix and not atrue measure of diversity. We do multivariate. We find if the ecosystem falls into groups and then we identify the estuaries that fall into these ecossytem groups. To do this, we make use of a dendrogram.

Create a dendrogram based on species composition. ‘vegan’ plays a key role in this process. Pairwise comparisons between each estuary to develop an index of similarity in species composition.

EF <- subset(EF, alpha > 0) #subset the data due to the 0s  
  
EF\_deco <- decostand(EF[,4:148], method = 'total')

## Bray-Curtis

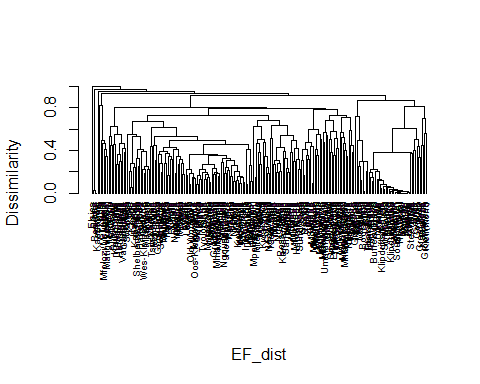
Bray-Curtis most commonly used index for similarity. Especially when there are many 0s, Bray-CUrtis is a good fit.

EF\_dist <- vegdist(EF\_deco, method = 'bray')

## H-clust and V-clust

H-clust is horizontal and v-clust is vertical.

EF\_clust <- hclust(EF\_dist, method = "average") # run clustering  
  
plot(  
 EF\_clust,  
 labels = EF$Estuary,  
 hang = -1,  
 cex = 0.6,  
 main = NULL,  
 ylab = "Dissimilarity",  
 sub = "",  
 col = "black"  
)

 Reduce the tree

slice <- cutree(EF\_clust,  
 h= 0.8)  
slice

## 228 226 230 171 178 184 186 212 183 200 229 224 166 213 148 172 163 221 145 218   
## 1 1 1 2 2 2 2 2 2 3 1 2 2 2 2 2 2 2 2 2   
## 168 176 156 170 216 174 147 193 179 152 143 151 164 190 155 161 217 188 210 187   
## 2 2 3 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2   
## 160 154 191 205 214 219 169 211 153 149 158 165 232 198 177 181 157 227 223 215   
## 2 2 2 2 2 2 2 2 2 2 2 2 5 2 2 2 2 2 2 2   
## 175 182 180 199 201 185 225 231 150 189 195 146 207 222 206 173 159 204 197 144   
## 2 2 2 2 2 2 1 2 2 2 2 2 2 3 2 2 2 2 2 2   
## 220 162 202 192 208 209 194 167 203 90 129 59 110 92 128 89 85 77 87 107   
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2   
## 126 80 62 123 103 142 97 57 44 134 139 91 118 130 117 50 56 133 132 61   
## 2 2 3 2 2 4 3 2 6 2 2 2 2 2 2 2 6 2 2 2   
## 114 140 102 127 104 95 58 141 101 51 116 94 46 81 45 96 47 137 78 106   
## 2 2 2 2 2 2 2 2 2 6 2 2 6 3 3 2 6 2 2 2   
## 115 100 135 42 111 112 131 93 105 64 88 79 98 109 108 113 119 124 39 55   
## 2 2 2 3 2 2 2 2 2 2 3 2 2 2 2 2 2 4 2 7   
## 83 74 120 27 71 52 49 37 48 60 86 65 70 138 82 29 53 125 136 99   
## 7 2 2 2 2 2 2 2 3 2 2 7 3 2 4 6 6 2 2 4   
## 68 66 41 69 38 30 32 34 40 43 75 35 33 63 36 76 31 67 54 72   
## 6 6 6 8 6 6 6 6 6 6 6 6 6 4 7 9 6 2 4 9   
## 73 11 1 7 16 10 19 23 22 26 15 4 8 12 17 18 21 25   
## 9 6 6 6 6 6 6 4 4 6 6 6 6 6 6 6 6 10

EF$Slice80 <- slice #adding into data frame

Ordering data by slice then alpha

EF <- EF[order(EF$Slice80, -EF$alpha), ]

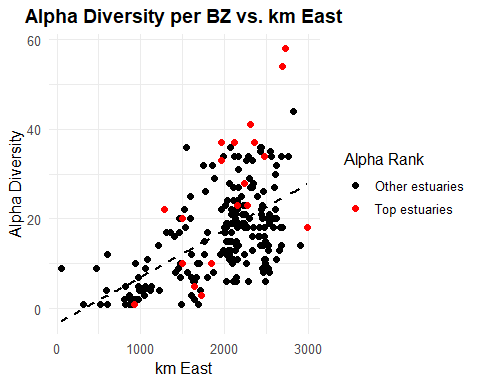
Creating list and score the top 2 in each Slice80 group as 1 and the rest as 0

# Initialize the Slice80\_list column to 0  
EF$Slice80\_list <- 0  
  
# Loop through each unique Slice80 group  
for (g in unique(EF$Slice80)) {  
   
 # Find all rows in the group  
 group <- which(EF$Slice80 == g)  
   
 # Determine how many rows to score  
 top\_n <- if (g == "2") {  
 4  
 } else if (g == "3") {  
 3  
 } else {  
 2  
 }  
   
 # Always score at least 1 row (if only 1 available)  
 EF$Slice80\_list[group[1:min(top\_n, length(group))]] <- 1  
}

## Plot the Slice80 list

g7 <- ggplot(EF, aes(x = kmEast, y = alpha)) +  
 geom\_point(aes(color = factor(Slice80\_list)), size = 2) +  
 geom\_smooth(method = "lm", se = FALSE, color = "black", linetype = "dashed") +  
 scale\_color\_manual(  
 values = c("0" = "black", "1" = "red"),  
 labels = c("Other estuaries", "Top estuaries"),  
 name = "Alpha Rank"  
 ) +  
 theme\_minimal(base\_size = 12) +  
 labs(  
 title = "Alpha Diversity per BZ vs. km East",  
 x = "km East",  
 y = "Alpha Diversity"  
 ) +  
 theme(  
 plot.title = element\_text(face = "bold", hjust = 0.5),  
 axis.title = element\_text(size = 12),  
 axis.text = element\_text(size = 10),  
 legend.position = "right"  
 )  
  
g7

## `geom\_smooth()` using formula = 'y ~ x'

 List the species in the groups

Slice80\_spp <- specnumber(EF[,4:148], EF$Slice80\_list)  
Slice80\_spp

## 0 1   
## 129 100

There are 100 species in the list and 145 species in total. So, we calculate the percentage of species within the list.

Slice80\_perc <- (100/145) \*100  
Slice80\_perc

## [1] 68.96552

## The Range Centre Method

Calculate the range

# species abundance data  
spp <- EF[, 4:148]  
  
# distance from eastern boundary  
coords <- EF$kmEast

Now we need to calculate the range for each species. We can create a function to do this.

range\_stats <- function(abundance, coords) {  
 present\_coords <- coords[abundance > 0] # get kmEast values where species is present  
   
 if (length(present\_coords) == 0) {  
 # species not present anywhere  
 return(c(west\_limit = NA, east\_limit = NA, center = NA))  
 } else {  
 east <- min(present\_coords) # furthest west point  
 west <- max(present\_coords) # furthest east point  
 centre <- mean(c(east, west)) # center of range  
   
 return(c(east\_limit = east, west\_limit = west, centre = centre))  
 }  
}  
  
# Apply the function across all species (columns)  
spp\_range <- t(apply(spp, 2, range\_stats, coords = coords))

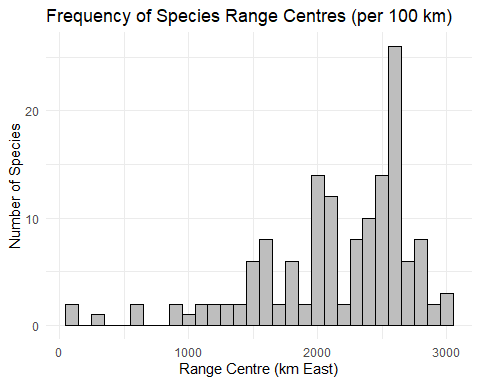
Now we just need to plop our results into a dataframe which contains the species names as well as their west\_limit, east\_limit and centre values.

# convert result to a clean data frame  
spp\_range\_list <- data.frame(  
 Species = colnames(spp), # species names from column headers  
 spp\_range # range stats: west, east, center  
)  
  
# preview first few rows of the result  
head(spp\_range\_list)

## Species east\_limit west\_limit centre  
## Spot.damsel Spot.damsel 2997 2997 2997.0  
## Estuarine.bream Estuarine.bream 1966 2821 2393.5  
## Bald.glassy Bald.glassy 1861 2821 2341.0  
## Slender.glassy Slender.glassy 2255 2997 2626.0  
## Longspine.glassy Longspine.glassy 2119 2997 2558.0  
## Evileye.blaasop Evileye.blaasop 969 2733 1851.0

Now we can plot a histogram with the frequency of range centres per 100 km.

g8<- ggplot(spp\_range\_list, aes(x = centre)) +  
 geom\_histogram(binwidth = 100, fill = "grey", color = "black") +  
 xlab("Range Centre (km East)") +  
 ylab("Number of Species") +  
 ggtitle("Frequency of Species Range Centres (per 100 km)") +  
 theme\_minimal()  
  
g8

 Extract frequencies from histogram and add that to the original data frame

EF$block <- ceiling(EF$kmEast/100)  
  
EF$RC\_list <- 1  
  
EF <- EF[order(EF$kmEast),]  
  
interval\_table <- table(cut(spp\_range\_list$centre,seq(0,3000,100)))  
print(order(interval\_table, decreasing = TRUE))

## [1] 26 25 27 20 21 16 24 22 29 18 15 17 19 23 28 30 1 7 10 12 13 14 3 9 11  
## [26] 2 4 5 6 8

# Define blocks to score as 0  
zero\_blocks <- c(13, 14, 3, 9, 11, 2, 4, 5, 6, 8)  
  
# Loop through each row  
for (i in 1:nrow(EF)) {  
 if (EF$block[i] %in% zero\_blocks) {  
 EF$RC\_list[i] <- 0  
 }  
}

Subset data and remove the zeros

EF <- subset(EF, RC\_list > 0)

Order data by block and alpha

EF <- EF[order(EF$block, -EF$alpha), ]

Assigning max

max <- EF$alpha[1] # Initialize max with the first alpha value  
EF$RC\_list <- 0   
EF$RC\_list[1] <- 1  
  
for (j in 2:nrow(EF)) {  
 if (EF$block[j] > EF$block[j-1]) {  
 max <- 0  
 if (EF$alpha[j] > max) {  
 EF$RC\_list[j] <- 1   
 max <- EF$alpha[j]  
 }  
 } else if (EF$block[j] == EF$block[j-1]) {  
 if (EF$alpha[j] > max) {  
 EF$RC\_list[j] <- 1  
 EF$RC\_list[j-1] <- 0  
 max <- EF$alpha[j]  
 }  
 }  
}  
  
sum(EF$RC\_list)

## [1] 20

List the species in the groups

RC\_spp <- specnumber(EF[,4:148], EF$RC\_list)  
RC\_spp

## 0 1   
## 117 121

There are 121 species in the list and 145 species in total. So, we calculate the percentage of species within the list.

RC\_perc <- (121/145) \*100  
RC\_perc

## [1] 83.44828

## Plot the RC list

g9 <- ggplot(EF, aes(x = kmEast, y = alpha)) +  
 geom\_point(aes(color = factor(RC\_list)), size = 2) +  
 geom\_smooth(method = "lm", se = FALSE, color = "black", linetype = "dashed") +  
 scale\_color\_manual(  
 values = c("0" = "black", "1" = "red"),  
 labels = c("Other estuaries", "Top estuaries"),  
 name = "Top 20 of method"  
 ) +  
 theme\_minimal(base\_size = 12) +  
 labs(  
 title = "Alpha Diversity vs. km East",  
 x = "km East",  
 y = "Alpha Diversity"  
 ) +  
 theme(  
 plot.title = element\_text(face = "bold", hjust = 0.5),  
 axis.title = element\_text(size = 12),  
 axis.text = element\_text(size = 10),  
 legend.position = "right"  
 )  
  
g9

## `geom\_smooth()` using formula = 'y ~ x'

