Fishery definition for YFT in dolphin sets

Haikun Xu

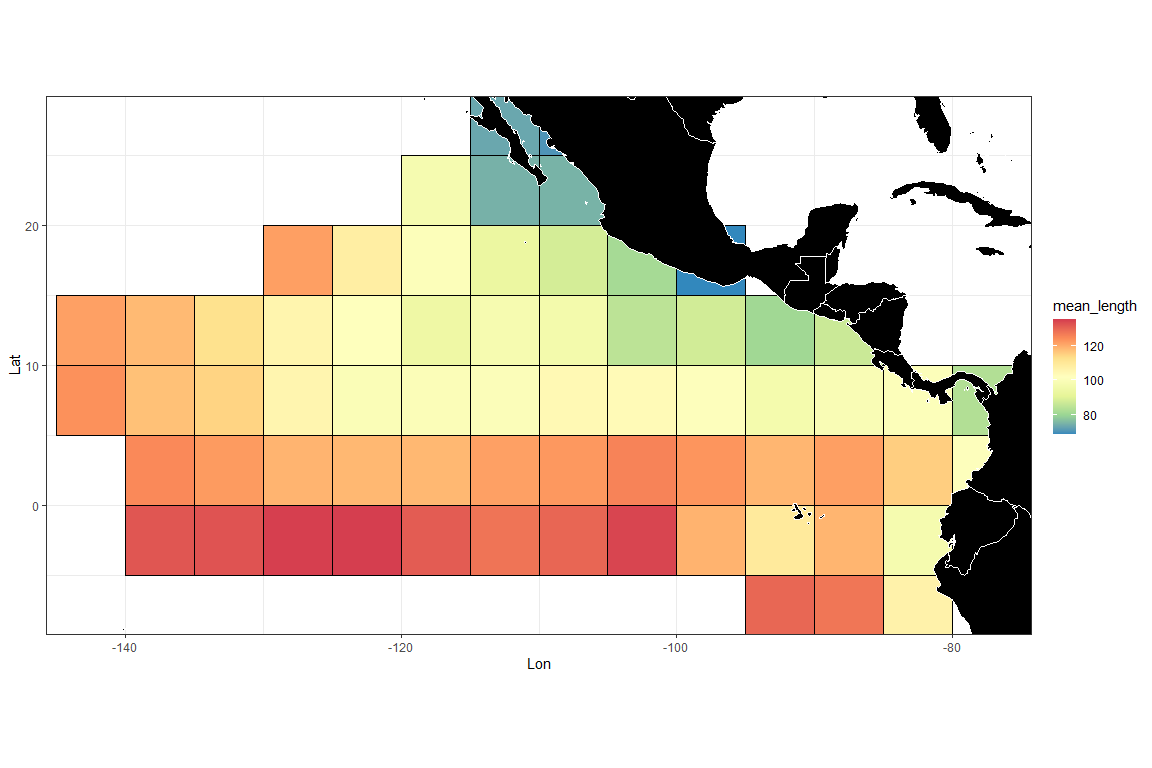
2024-07-19

library(FishFreqClustering)  
library(FishFreqTree)  
library(tidyverse)

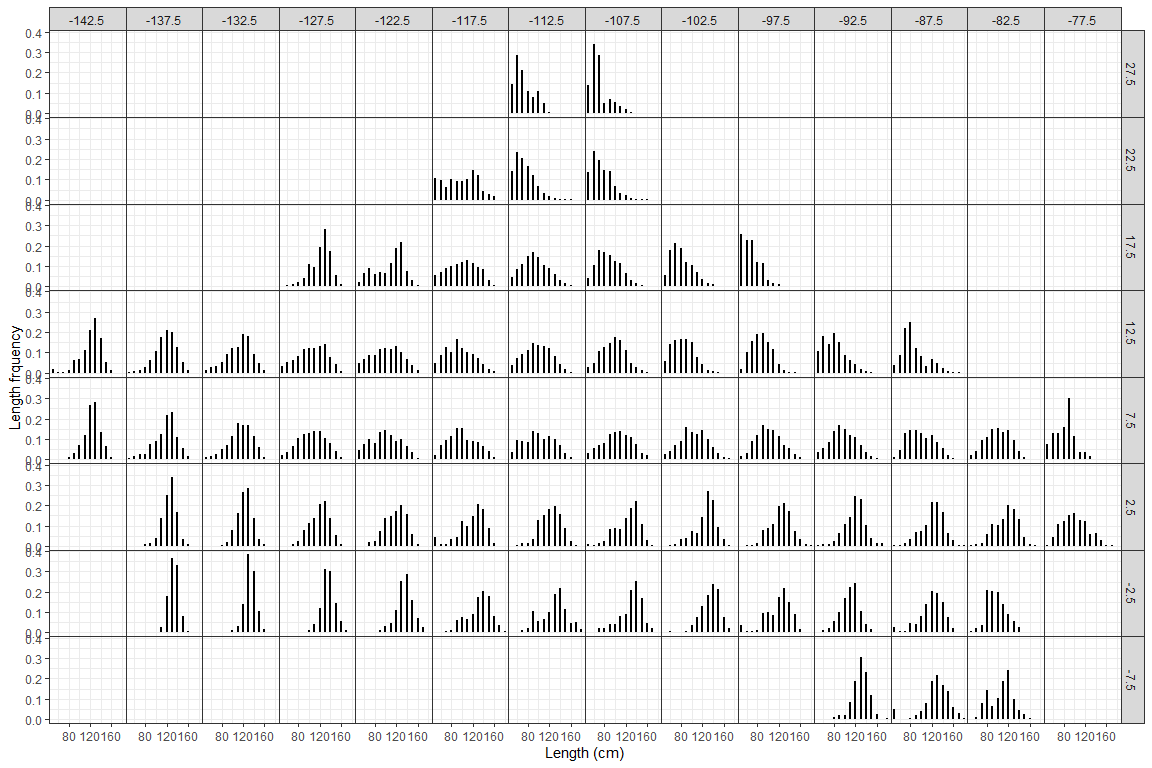
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── 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

directory <- "D:/OneDrive - IATTC/IATTC/2024/Irregular clustering/YFT DEL/"  
setwd(directory)  
  
Raw <- read.csv("yft\_lf\_2000\_to\_2022.csv")  
Raw$quarter = ceiling(Raw$month / 3)  
Raw$lat = Raw$lat.5deg + 2.5  
Raw$lon = Raw$lon.5deg - 2.5

LF.DEL <- Raw %>% filter(class == 6, setype == 1) # 1=DEL; 4=NOA; 5=OBJ  
LF <- LF.DEL[, c("year", "quarter", "lat", "lon", paste0("X", 1:201))] %>%  
 group\_by(lat, lon) %>%  
 mutate(N = length(unique(paste0(year, "-", quarter)))) %>%  
 filter(N > 4, lat > -10) # remove the cells with less than 4 quarters of data since 2000  
  
bins <- seq(1, 201, 1) # data length bins  
new\_bins <- seq(50, 180, 10) # bins to be used in the clustering analysis  
  
# first aggregate the raw LF to the new bins by quarter  
LF1 <-  
 lf.aggregate(  
 LF,  
 fcol = 5,  
 lcol = 205,  
 bins,  
 new\_bins,  
 LengthOnly = FALSE  
 )  
  
# Chekcing the data by making two plots  
bins <- new\_bins # use the new bins  
nbins <- length(bins)  
fcol = 5  
lcol = 4 + length(bins)  
save\_dir=directory  
  
make.meanl.map(LF1,fcol,lcol,bins,save\_dir,width=10,height=10)



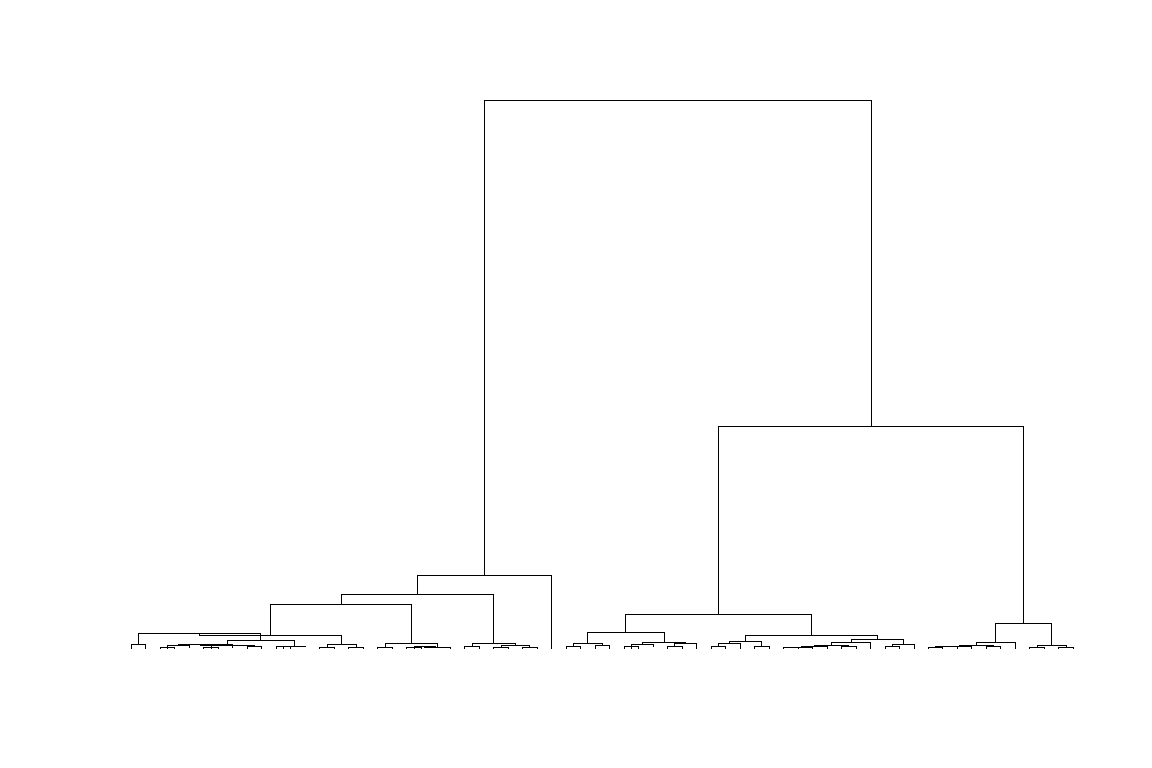
make.lf.map(LF1,fcol,lcol,bins,save\_dir)



# divide the LF by the mean LF for the year-quarter  
LF2 <- lf.demean(LF1, fcol, lcol, bins)

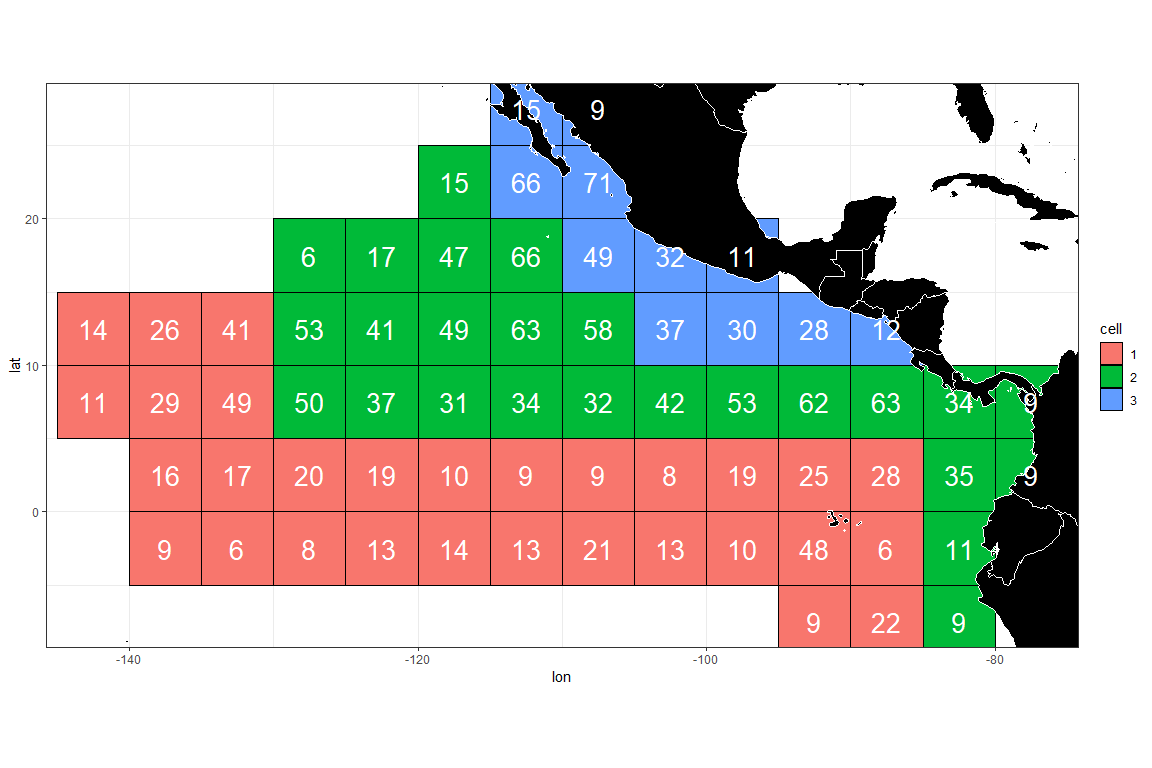
mmd <- LF2[,c(2,4:(lcol+1))] # mmd is the input data for the clustering analysis - it should have year, lat, lon, and bin numbers  
  
# setting up input data frames for clustering algorithm  
temp = packbylatlon(mmd, 5, 5, nbins) # aggregate the input LF across time for each grid cell  
packedmmd3 = temp$table1  
packedpdf3 = topdf(packedmmd3, 4, 3 + nbins)  
packedcdf3 = tocdf(packedpdf3, 4, 3 + nbins)  
mmdt = packedmmd3[packedmmd3[, 4 + nbins] > 0,]  
rrs = mmdt[, 4 + nbins]  
mmdtpdf = packedpdf3[packedmmd3[, 4 + nbins] > 0,]  
mmdtpdf[, 4 + nbins] = mmdt[, 4 + nbins]  
mmdtcdf = packedcdf3[packedmmd3[, 4 + nbins] > 0,]  
mmdtcdf[, 4 + nbins] = mmdt[, 4 + nbins]  
  
densmatx = matrix(0, nrow(mmdt), nbins)  
densmaty = matrix(0, nrow(mmdt), nbins)  
for (i in 1:nrow(mmdt)) {  
 weightvec = t(mmdt[i, 4:(3 + nbins)])  
 weightvec = weightvec / sum(weightvec)  
 densmatx[i, ] = bins  
 densmaty[i, ] = t(weightvec)  
}  
  
# run distributional clustering with adjacency criterion  
adjmat <- adjinf(mmdtpdf[, 2], mmdtpdf[, 3]) # the matrix specifying adjacency   
alydens.spatial23 <-  
 hclust.regionsmm(  
 as.matrix(densmaty),  
 adj = TRUE, # adjacent areas  
 adjmat = adjmat,  
 rr = (rrs) # rr is the weighting factor  
 )

# making maps of the clusters and corresponding L-F density curves  
# Look at the tree structure  
cplotu(alydens.spatial23$merges, alydens.spatial23$distseq, hopt = 'dist')



## NULL

# kk is the number of clusters to use  
kk = 3  
# save clustering results  
temp <- putcolor(alydens.spatial23$merges, kk)  
cluster <- cbind(mmdt[,2:3], factor(temp-1), rrs)  
names(cluster) <- c("lat", "lon", "cell", "Nsamp")  
write.csv(cluster, file = paste0(save\_dir, "cluster\_YFT", kk, ".csv"), row.names = FALSE)  
  
# # map of clusters  
wmap <- map\_data("world")  
ggplot(data = cluster) +  
 geom\_tile(aes(x = lon, y = lat, fill = cell), color = "black") +  
 geom\_polygon(  
 data = wmap,  
 aes(long, lat, group = group),  
 fill = "black",  
 colour = "white",  
 lwd = 0.5  
 ) +  
 geom\_text(aes(x = lon, y = lat, label = Nsamp), color = "white",size=7) +  
 coord\_quickmap(ylim = c(min(cluster$lat), max(cluster$lat)),  
 xlim = c(min(cluster$lon), max(cluster$lon))) +  
 theme\_bw()



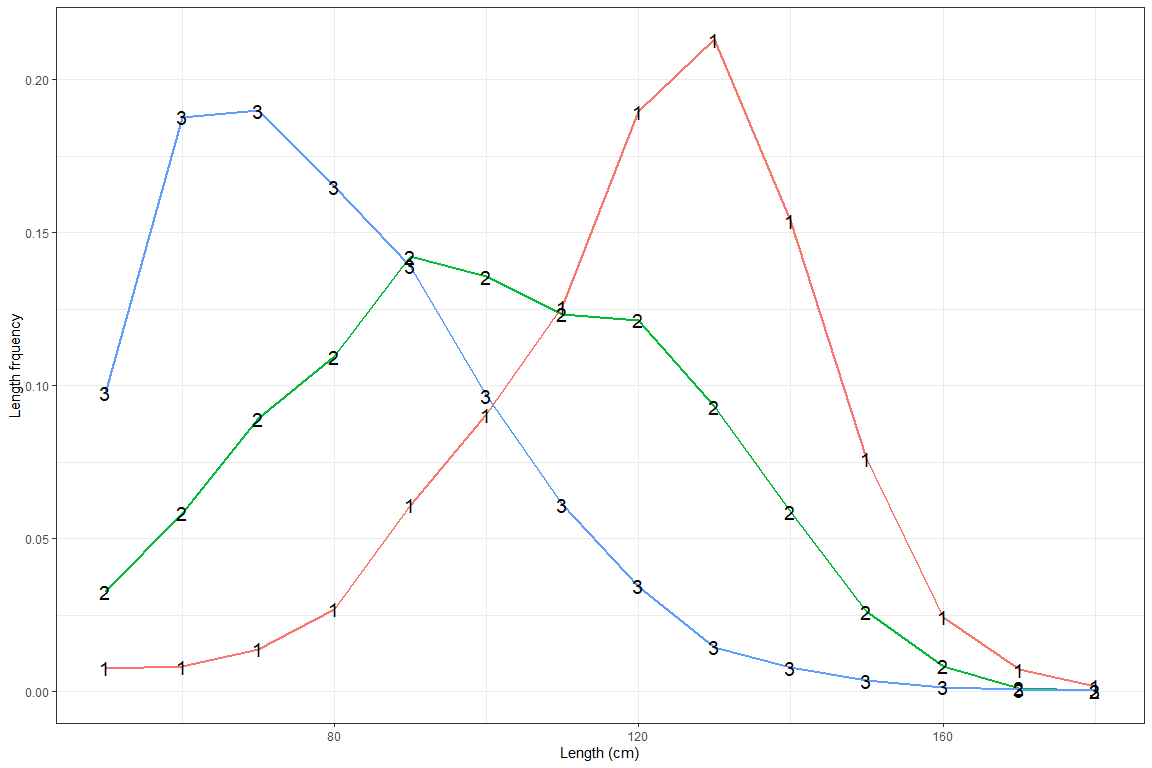
ggsave(file = paste0(save\_dir, "Clustering\_map", kk, ".png"))

## Saving 12 x 8 in image

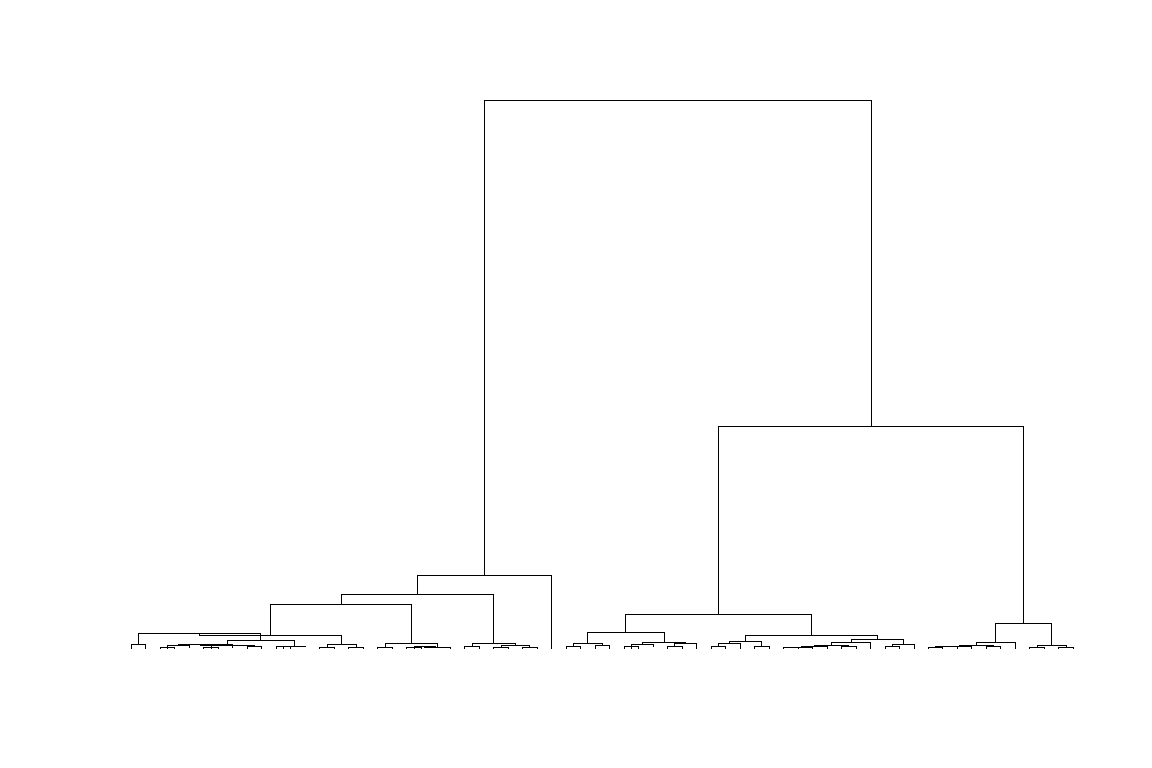
# mean LF in each area  
LF1\_cluster <- left\_join(LF1, cluster) %>% rename(Flag = cell)

## Joining with `by = join\_by(lat, lon)`

make.lf.cell(LF1\_cluster, fcol, lcol, bins, save\_dir, plot\_name = paste0("NewLF2", kk))

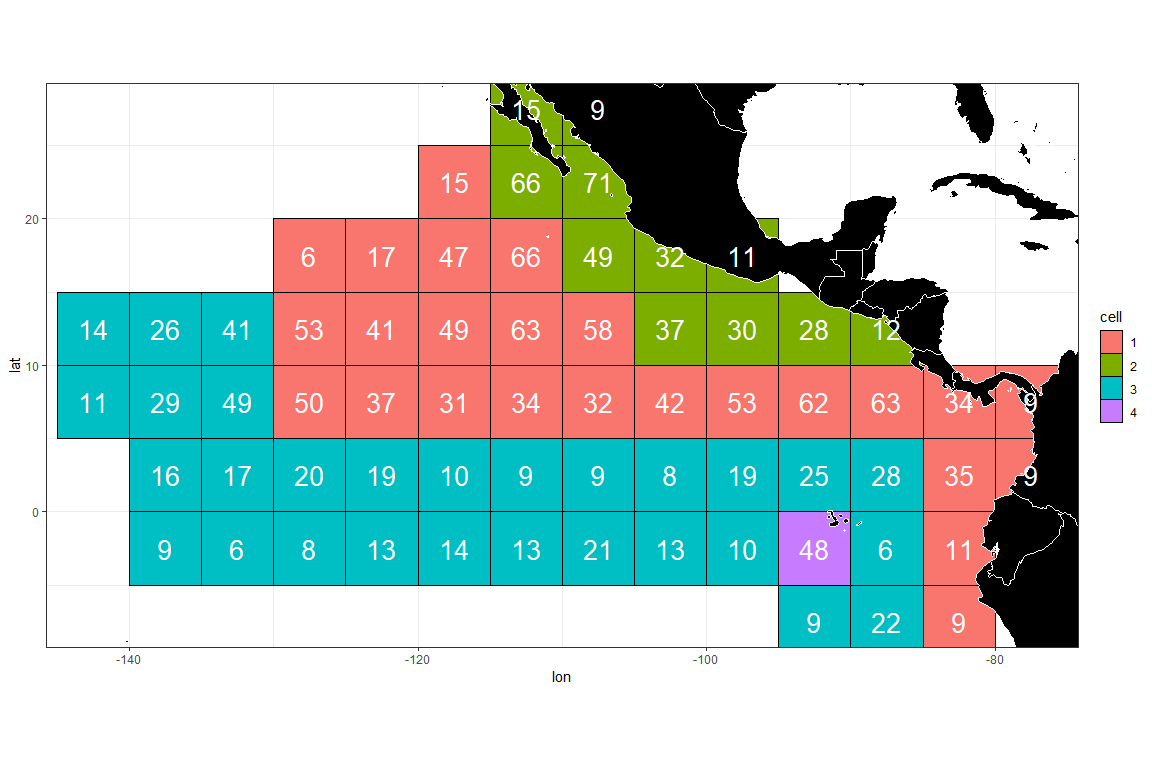


# making maps of the clusters and corresponding L-F density curves  
# Look at the tree structure  
cplotu(alydens.spatial23$merges, alydens.spatial23$distseq, hopt = 'dist')



## NULL

# kk is the number of clusters to use  
kk = 4  
# save clustering results  
temp <- putcolor(alydens.spatial23$merges, kk)  
cluster <- cbind(mmdt[,2:3], factor(temp-1), rrs)  
names(cluster) <- c("lat", "lon", "cell", "Nsamp")  
write.csv(cluster, file = paste0(save\_dir, "cluster\_YFT", kk, ".csv"), row.names = FALSE)  
  
# # map of clusters  
wmap <- map\_data("world")  
ggplot(data = cluster) +  
 geom\_tile(aes(x = lon, y = lat, fill = cell), color = "black") +  
 geom\_polygon(  
 data = wmap,  
 aes(long, lat, group = group),  
 fill = "black",  
 colour = "white",  
 lwd = 0.5  
 ) +  
 geom\_text(aes(x = lon, y = lat, label = Nsamp), color = "white",size=7) +  
 coord\_quickmap(ylim = c(min(cluster$lat), max(cluster$lat)),  
 xlim = c(min(cluster$lon), max(cluster$lon))) +  
 theme\_bw()



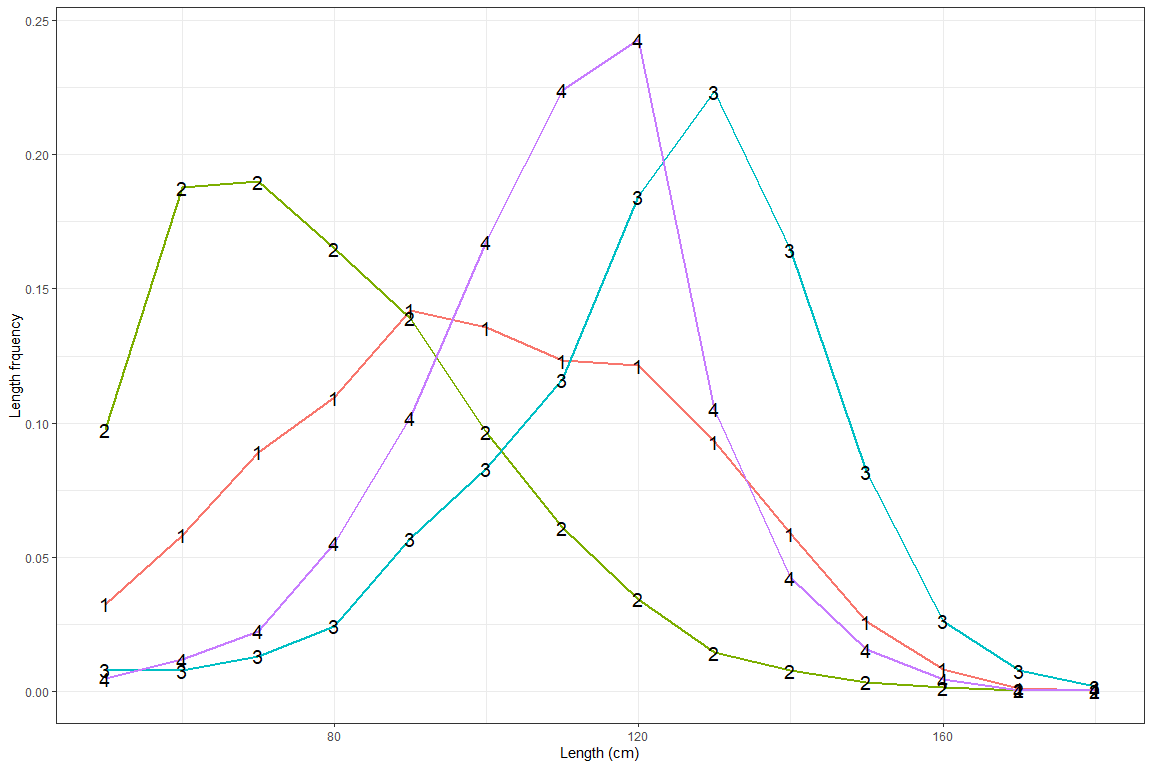
ggsave(file = paste0(save\_dir, "Clustering\_map", kk, ".png"))

## Saving 12 x 8 in image

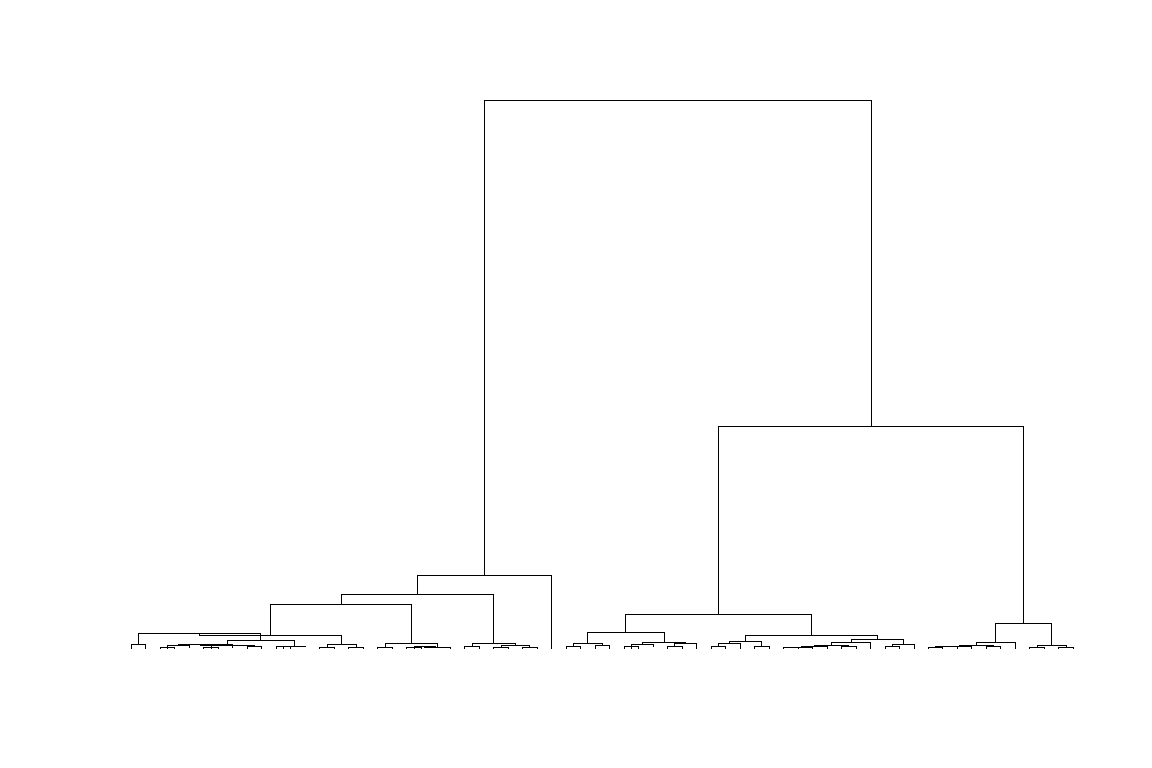
# mean LF in each area  
LF1\_cluster <- left\_join(LF1, cluster) %>% rename(Flag = cell)

## Joining with `by = join\_by(lat, lon)`

make.lf.cell(LF1\_cluster, fcol, lcol, bins, save\_dir, plot\_name = paste0("NewLF2", kk))

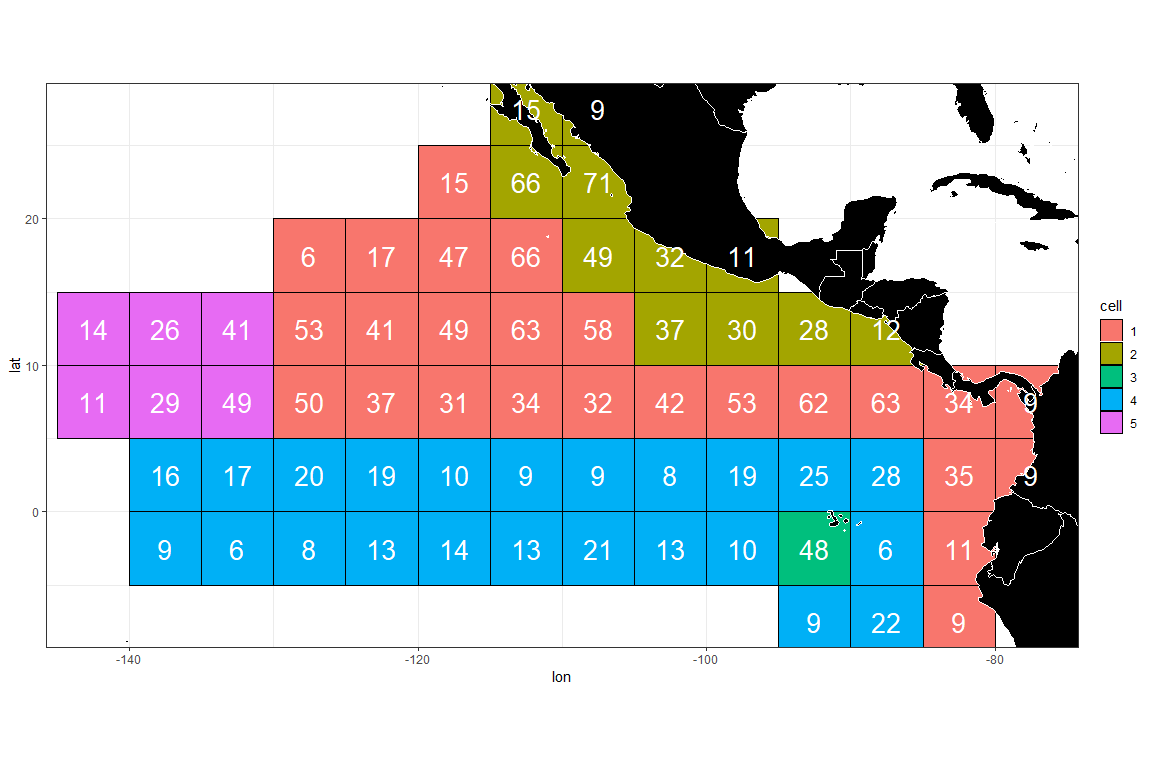


# making maps of the clusters and corresponding L-F density curves  
# Look at the tree structure  
cplotu(alydens.spatial23$merges, alydens.spatial23$distseq, hopt = 'dist')



## NULL

# kk is the number of clusters to use  
kk = 5  
# save clustering results  
temp <- putcolor(alydens.spatial23$merges, kk)  
cluster <- cbind(mmdt[,2:3], factor(temp-1), rrs)  
names(cluster) <- c("lat", "lon", "cell", "Nsamp")  
write.csv(cluster, file = paste0(save\_dir, "cluster\_YFT", kk, ".csv"), row.names = FALSE)  
  
# # map of clusters  
wmap <- map\_data("world")  
ggplot(data = cluster) +  
 geom\_tile(aes(x = lon, y = lat, fill = cell), color = "black") +  
 geom\_polygon(  
 data = wmap,  
 aes(long, lat, group = group),  
 fill = "black",  
 colour = "white",  
 lwd = 0.5  
 ) +  
 geom\_text(aes(x = lon, y = lat, label = Nsamp), color = "white",size=7) +  
 coord\_quickmap(ylim = c(min(cluster$lat), max(cluster$lat)),  
 xlim = c(min(cluster$lon), max(cluster$lon))) +  
 theme\_bw()



ggsave(file = paste0(save\_dir, "Clustering\_map", kk, ".png"))

## Saving 12 x 8 in image

# mean LF in each area  
LF1\_cluster <- left\_join(LF1, cluster) %>% rename(Flag = cell)

## Joining with `by = join\_by(lat, lon)`

make.lf.cell(LF1\_cluster, fcol, lcol, bins, save\_dir, plot\_name = paste0("NewLF2", kk))

