# Fishery definition for YFT in dolphin sets

Haikun Xu

2024-08-09

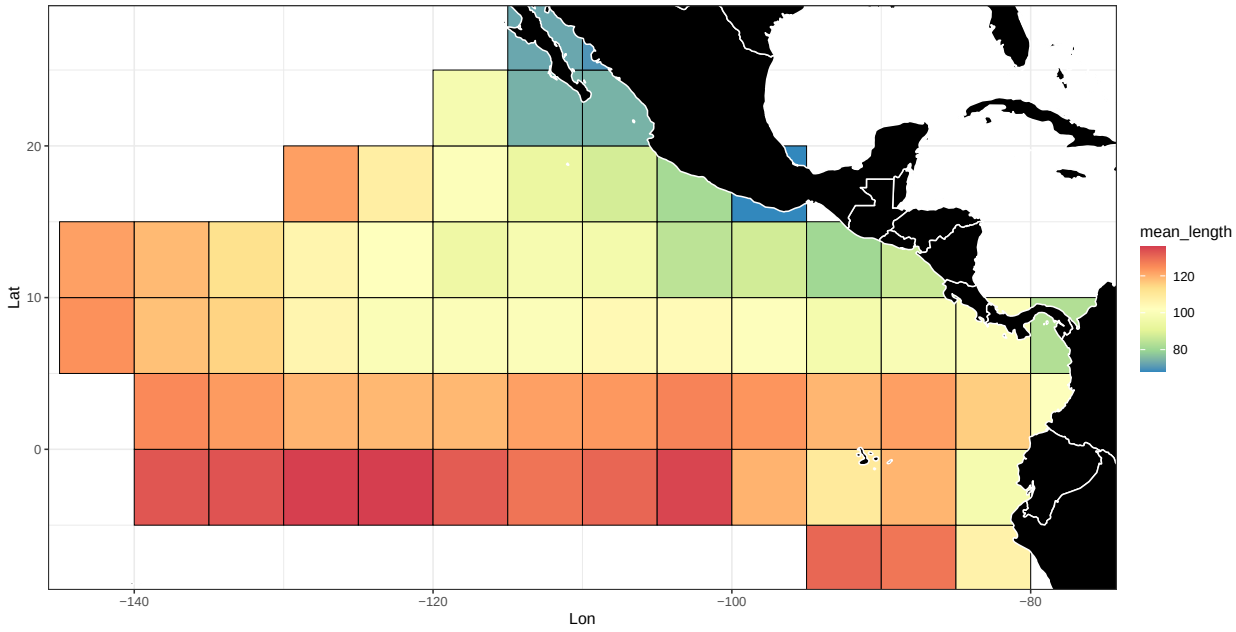
```
library(FishFreqClustering)
library(FishFreqTree)
library(tidyverse)
```

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

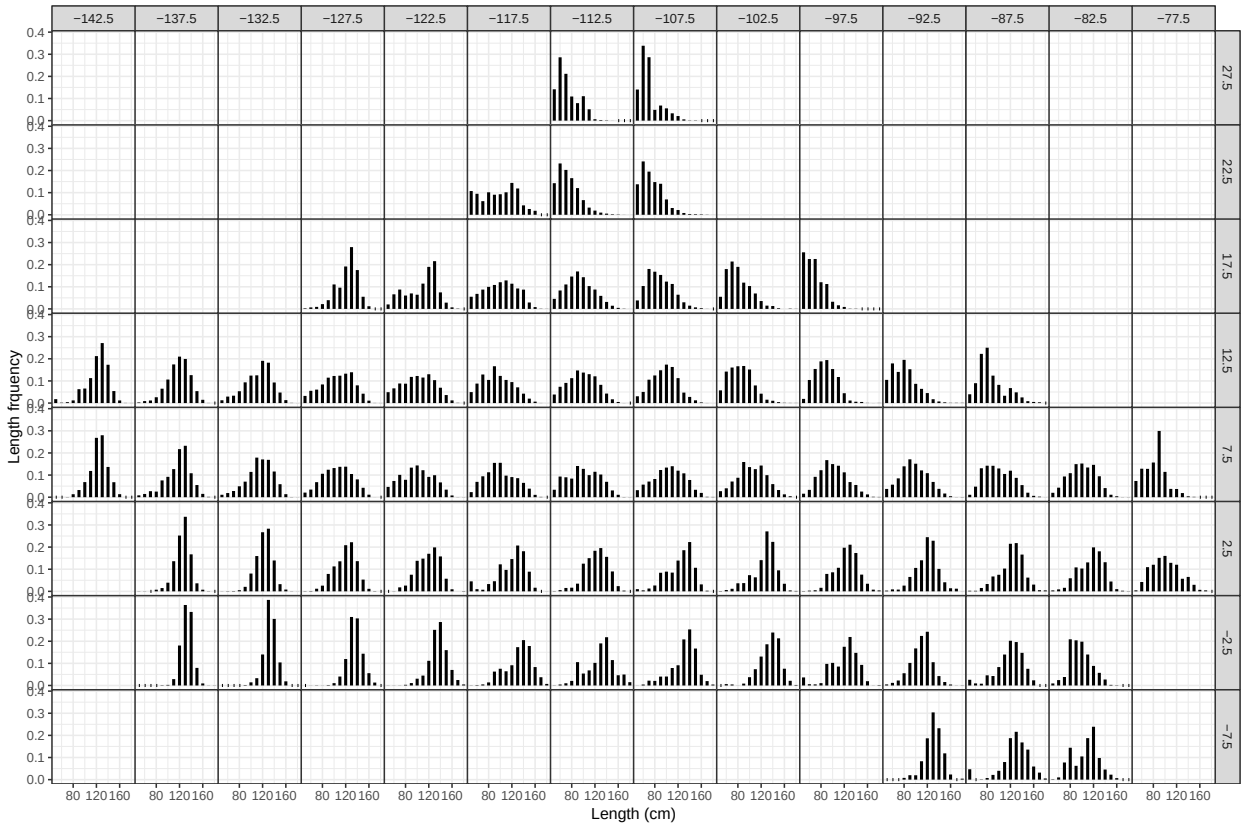
```
directory <- "D:/OneDrive - IATTC/IATTC/2024/Irregular clustering/YFT DEL/"
setwd(directory)

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)
```

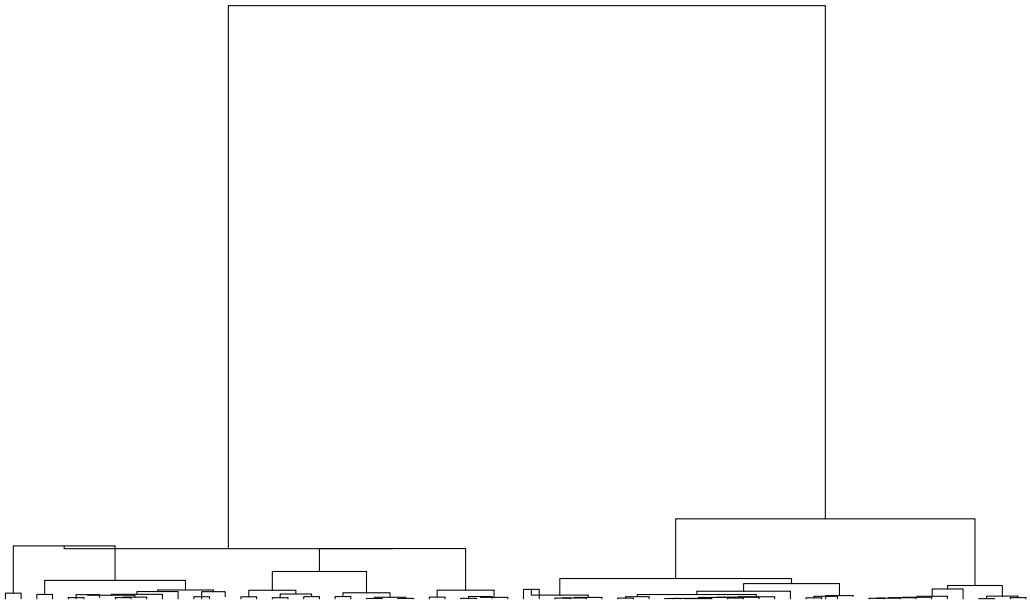



```
make.lf.map(LF1,fcol,lcol,bins,save_dir)
```




```
# 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,

# 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] # sample size
mmdtpdf = packedpdf3[packedmmd3[, 4 + nbins] > 0,] # PDF sums to 1 for each grid
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(mmdtpdf[i, 4:(3 + nbins)])
  # weightvec = weightvec / sum(weightvec)
  densmatx[i, ] = bins
  densmaty[i, ] = t(weightvec)
}

# run distributional clustering with adjacency criterion
```

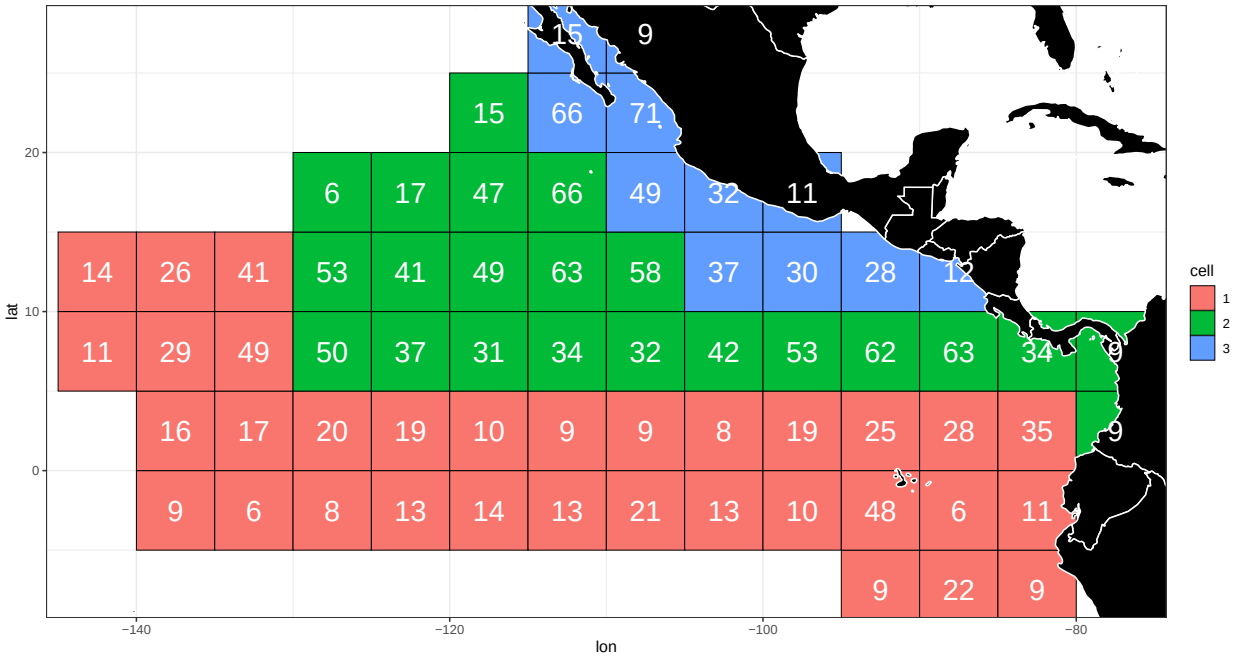
```
adjmat <- adjinf(mmdtpdf[, 2], mmdtpdf[, 3]) # the matrix specifying adjacency
alydens.spatial23 <-
  hclust.regionsmm(
    as.matrix(densmaty),
    adj = TRUE, # adjacent areas
    adjmat = adjmat,
    rr = sign(rrs) # rr is the weighting factor; equal weighting is used in this case
  )

# Look at the tree structure
cplotu(alydens.spatial23$merges, alydens.spatial23$distseq, hopt = 'dist')
```



```
## NULL
```

```
# making maps of the clusters and corresponding L-F density curves

#     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()
```



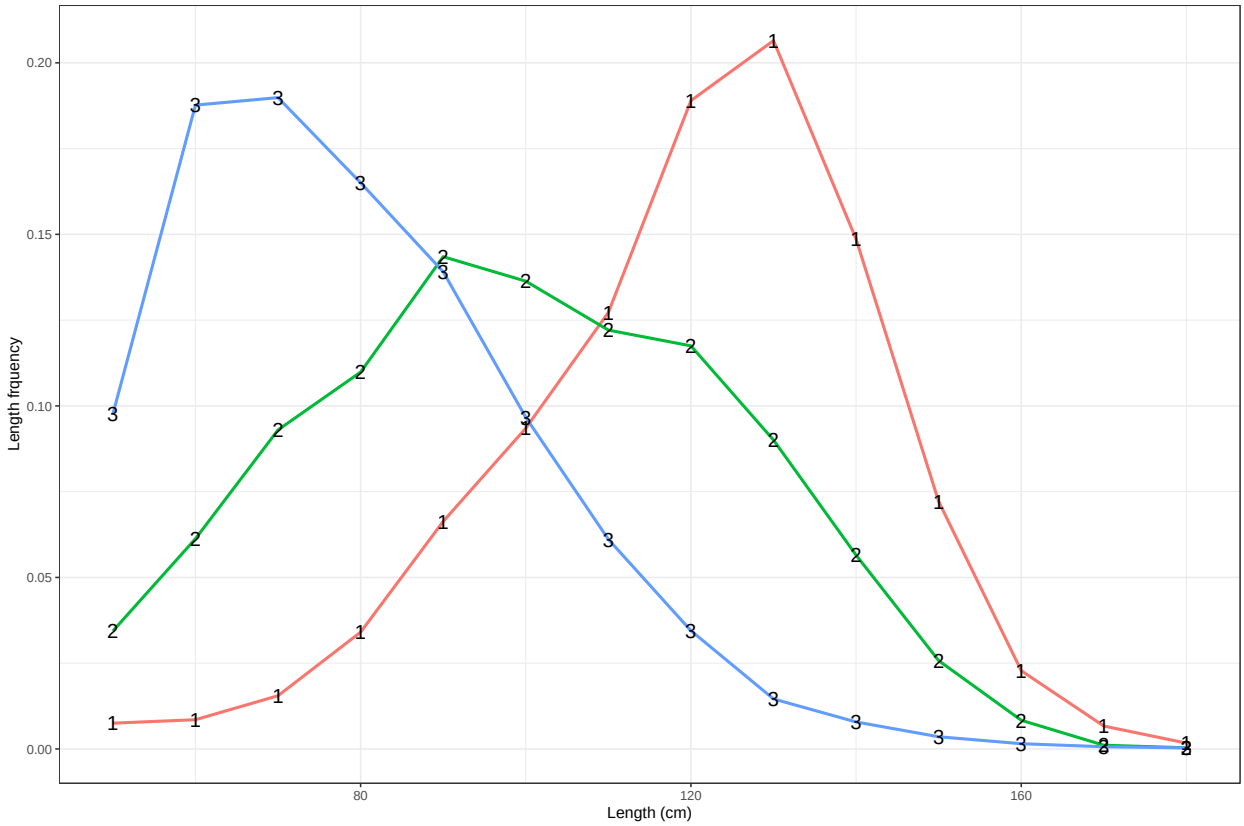

```
ggsave(file = paste0(save_dir, "Clustering_map", kk, ".png"))
```

```
## Saving 12 x 8 in image
```

```
# mean LF in each area
LF1_cluster <- left_join(LF1, cluster) %>% rename(Flag = cell)
```

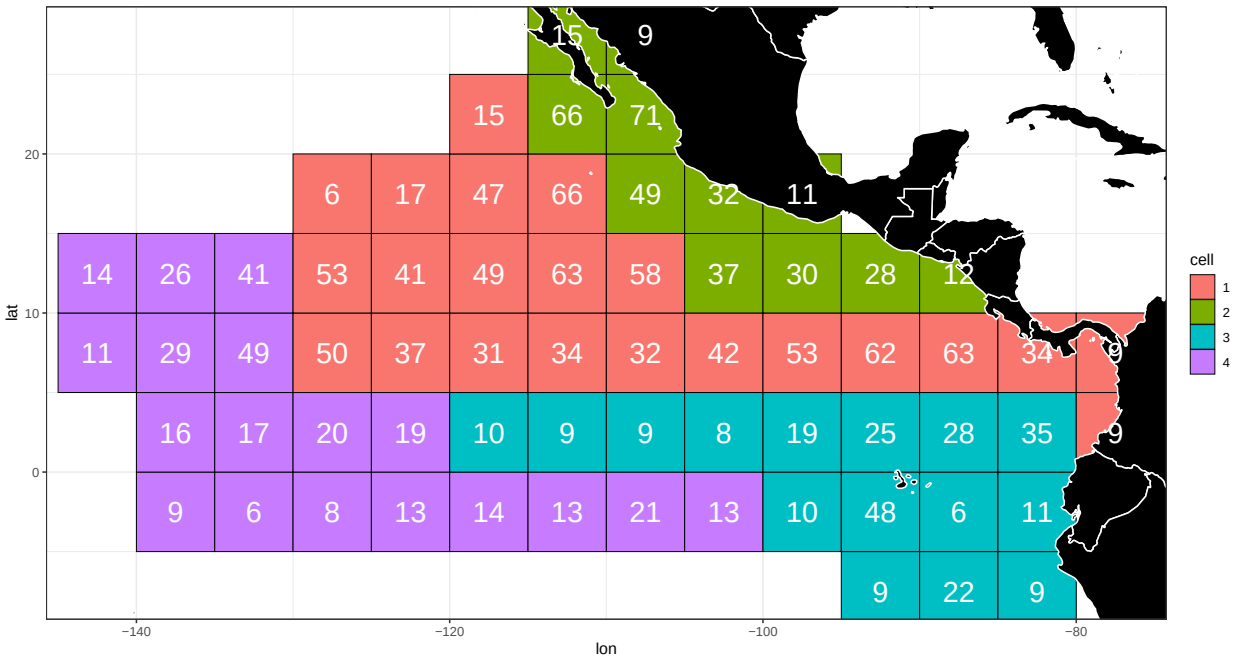
```
## Joining with `by = join_by(lat, lon)`
```

```
make.lf.cell(LF1_cluster, fcol, lcol, bins, save_dir, plot_name = paste0("NewLF", kk))
```




```
# making maps of the clusters and corresponding L-F density curves

#     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()
```



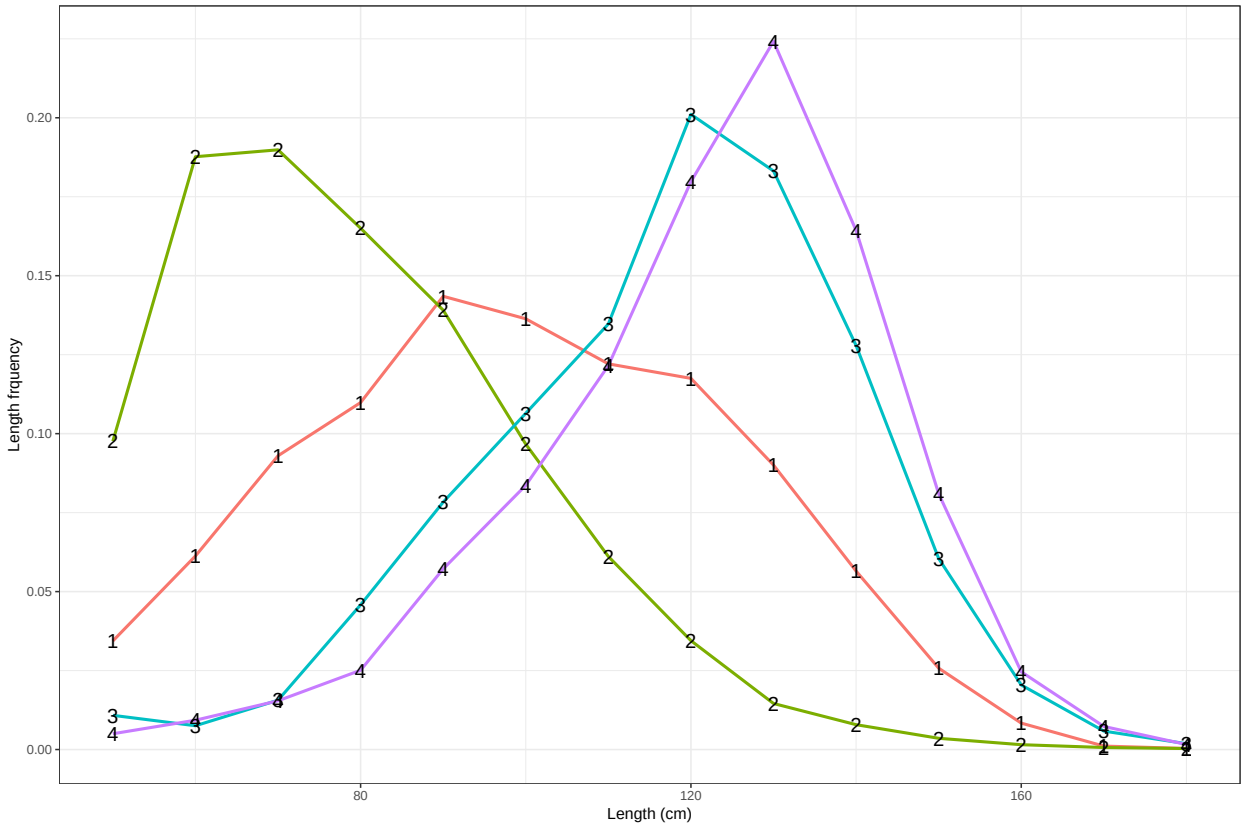

```
ggsave(file = paste0(save_dir, "Clustering_map", kk, ".png"))
```

```
## Saving 12 x 8 in image
```

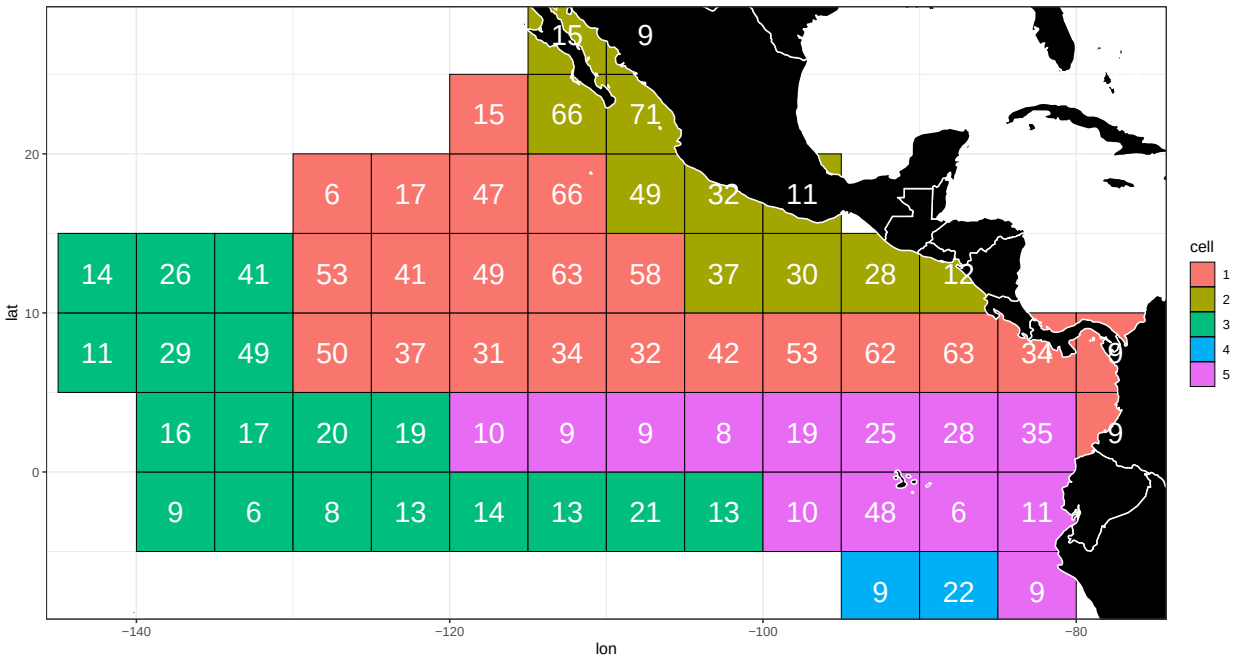
```
# mean LF in each area
LF1_cluster <- left_join(LF1, cluster) %>% rename(Flag = cell)
```

```
## Joining with `by = join_by(lat, lon)`
```

```
make.lf.cell(LF1_cluster, fcol, lcol, bins, save_dir, plot_name = paste0("NewLF", kk))
```




```
# making maps of the clusters and corresponding L-F density curves

#     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()
```



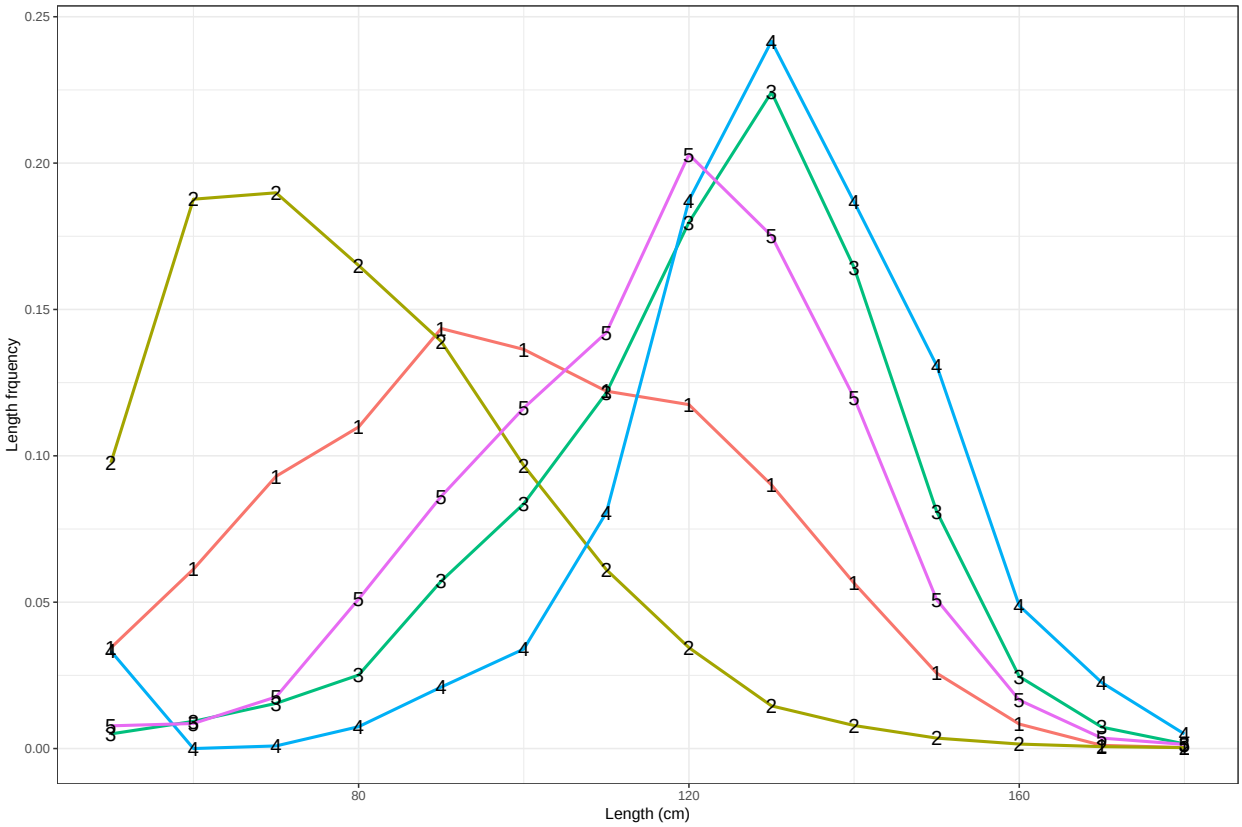

```
ggsave(file = paste0(save_dir, "Clustering_map", kk, ".png"))
```

```
## 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("NewLF", kk))
```


0.25
0.20
0.15
0.10
0.05
0.00
Length frquency
80
120
160
Length (cm)