02_Frequencies

2025-03-11

Contents


```
library(CCA)
library(dplyr)
library(ggplot2)
library(reshape2)
library(ggridges)
library(gridExtra)
library(patchwork)
library(FactoMineR)
```

Load

Loading Pre-processed data provided

```
## load
load(here::here("Data", "ProcessedData", "processed_AnalysisData.Rdata"))
```

```
processed_data
```

```
## # A tibble: 14,575 x 484
##    fishNum dateSample dateTimeSample      dateProcessed species   spCode
##    <chr>   <date>     <dttm>              <date>        <chr>      <dbl>
##  1 LT001   2022-07-21 2022-07-21 16:56:00 2022-07-27    lakeTrout     81
##  2 LT001   2022-07-21 2022-07-21 16:56:00 2022-07-27    lakeTrout     81
##  3 LT001   2022-07-21 2022-07-21 16:56:00 2022-07-27    lakeTrout     81
##  4 LT001   2022-07-21 2022-07-21 16:56:00 2022-07-27    lakeTrout     81
##  5 LT001   2022-07-21 2022-07-21 16:56:00 2022-07-27    lakeTrout     81
##  6 LT001   2022-07-21 2022-07-21 16:56:00 2022-07-27    lakeTrout     81
##  7 LT001   2022-07-21 2022-07-21 16:56:00 2022-07-27    lakeTrout     81
##  8 LT001   2022-07-21 2022-07-21 16:56:00 2022-07-27    lakeTrout     81
##  9 LT001   2022-07-21 2022-07-21 16:56:00 2022-07-27    lakeTrout     81
## 10 LT001   2022-07-21 2022-07-21 16:56:00 2022-07-27    lakeTrout     81
## # i 14,565 more rows
## # i 478 more variables: totalLength <dbl>, forkLength <dbl>, weight <dbl>,
## #   girth <dbl>, dorsoLatHeight <dbl>, clipTag <chr>, sex <chr>, mat <dbl>,
## #   airbladderTotalLength <dbl>, airBladderWidth <dbl>, airbladderWeight <dbl>,
## #   airBladderWeightCond <dbl>, agingStructure <chr>, tissueSample <chr>,
## #   Region_name <chr>, FishTrack <chr>, MaxTSdiff <dbl>, Ping_time <chr>,
## #   deltaRange <dbl>, deltaMinAng <dbl>, deltaMajAng <dbl>, ...
```

```
## variable names in data
names(processed_data)
```

```
##   [1] "fishNum"                      "dateSample"
##   [3] "dateTimeSample"               "dateProcessed"
##   [5] "species"                      "spCode"
##   [7] "totalLength"                  "forkLength"
##   [9] "weight"                       "girth"
##  [11] "dorsoLatHeight"               "clipTag"
##  [13] "sex"                          "mat"
##  [15] "airbladderTotalLength"        "airBladderWidth"
##  [17] "airbladderWeight"             "airBladderWeightCond"
##  [19] "agingStructure"               "tissueSample"
##  [21] "Region_name"                  "FishTrack"
##  [23] "MaxTSdiff"                    "Ping_time"
##  [25] "deltaRange"                   "deltaMinAng"
##  [27] "deltaMajAng"                  "aspectAngle"
##  [29] "Target_range"                 "Angle_minor_axis"
##  [31] "Angle_major_axis"             "Distance_minor_axis"
##  [33] "Distance_major_axis"          "StandDev_Angles_Minor_Axis"
##  [35] "StandDev_Angles_Major_Axis"   "Target_true_depth"
##  [37] "pingNumber"                   "Ping_S"
##  [39] "Ping_E"                       "Num_targets"
##  [41] "TS_mean"                      "Target_range_mean"
##  [43] "Speed_4D_mean_unsmoothed"     "Fish_track_change_in_range"
##  [45] "Time_in_beam"                 "Distance_3D_unsmoothed"
##  [47] "Thickness_mean"               "Exclude_below_line_range_mean"
##  [49] "Target_depth_mean"            "Target_depth_max"
##  [51] "Target_depth_min"             "Fish_track_change_in_depth"
##  [53] "Region_bottom_altitude_min"   "Region_bottom_altitude_max"
##  [55] "Region_bottom_altitude_mean"  "Region_top_altitude_min"
##  [57] "Region_top_altitude_max"      "Region_top_altitude_mean"
##  [59] "F45"                          "F45.5"
##  [61] "F46"                          "F46.5"
##  [63] "F47"                          "F47.5"
##  [65] "F48"                          "F48.5"
##  [67] "F49"                          "F49.5"
##  [69] "F50"                          "F50.5"
##  [71] "F51"                          "F51.5"
##  [73] "F52"                          "F52.5"
##  [75] "F53"                          "F53.5"
##  [77] "F54"                          "F54.5"
##  [79] "F55"                          "F55.5"
##  [81] "F56"                          "F56.5"
##  [83] "F57"                          "F57.5"
##  [85] "F58"                          "F58.5"
##  [87] "F59"                          "F59.5"
##  [89] "F60"                          "F60.5"
##  [91] "F61"                          "F61.5"
##  [93] "F62"                          "F62.5"
##  [95] "F63"                          "F63.5"
##  [97] "F64"                          "F64.5"
##  [99] "F65"                          "F65.5"
## [101] "F66"                          "F66.5"
## [103] "F67"                          "F67.5"
## [105] "F68"                          "F68.5"
## [107] "F69"                          "F69.5"
```

```
## [109] "F70"                                "F70.5"
## [111] "F71"                                "F71.5"
## [113] "F72"                                "F72.5"
## [115] "F73"                                "F73.5"
## [117] "F74"                                "F74.5"
## [119] "F75"                                "F75.5"
## [121] "F76"                                "F76.5"
## [123] "F77"                                "F77.5"
## [125] "F78"                                "F78.5"
## [127] "F79"                                "F79.5"
## [129] "F80"                                "F80.5"
## [131] "F81"                                "F81.5"
## [133] "F82"                                "F82.5"
## [135] "F83"                                "F83.5"
## [137] "F84"                                "F84.5"
## [139] "F85"                                "F85.5"
## [141] "F86"                                "F86.5"
## [143] "F87"                                "F87.5"
## [145] "F88"                                "F88.5"
## [147] "F89"                                "F89.5"
## [149] "F90"                                "F90.5"
## [151] "F91"                                "F91.5"
## [153] "F92"                                "F92.5"
## [155] "F93"                                "F93.5"
## [157] "F94"                                "F94.5"
## [159] "F95"                                "F95.5"
## [161] "F96"                                "F96.5"
## [163] "F97"                                "F97.5"
## [165] "F98"                                "F98.5"
## [167] "F99"                                "F99.5"
## [169] "F100"                               "F100.5"
## [171] "F101"                               "F101.5"
## [173] "F102"                               "F102.5"
## [175] "F103"                               "F103.5"
## [177] "F104"                               "F104.5"
## [179] "F105"                               "F105.5"
## [181] "F106"                               "F106.5"
## [183] "F107"                               "F107.5"
## [185] "F108"                               "F108.5"
## [187] "F109"                               "F109.5"
## [189] "F110"                               "F110.5"
## [191] "F111"                               "F111.5"
## [193] "F112"                               "F112.5"
## [195] "F113"                               "F113.5"
## [197] "F114"                               "F114.5"
## [199] "F115"                               "F115.5"
## [201] "F116"                               "F116.5"
## [203] "F117"                               "F117.5"
## [205] "F118"                               "F118.5"
## [207] "F119"                               "F119.5"
## [209] "F120"                               "F120.5"
## [211] "F121"                               "F121.5"
## [213] "F122"                               "F122.5"
## [215] "F123"                               "F123.5"
```

```
## [217] "F124"                              "F124.5"
## [219] "F125"                              "F125.5"
## [221] "F126"                              "F126.5"
## [223] "F127"                              "F127.5"
## [225] "F128"                              "F128.5"
## [227] "F129"                              "F129.5"
## [229] "F130"                              "F130.5"
## [231] "F131"                              "F131.5"
## [233] "F132"                              "F132.5"
## [235] "F133"                              "F133.5"
## [237] "F134"                              "F134.5"
## [239] "F135"                              "F135.5"
## [241] "F136"                              "F136.5"
## [243] "F137"                              "F137.5"
## [245] "F138"                              "F138.5"
## [247] "F139"                              "F139.5"
## [249] "F140"                              "F140.5"
## [251] "F141"                              "F141.5"
## [253] "F142"                              "F142.5"
## [255] "F143"                              "F143.5"
## [257] "F144"                              "F144.5"
## [259] "F145"                              "F145.5"
## [261] "F146"                              "F146.5"
## [263] "F147"                              "F147.5"
## [265] "F148"                              "F148.5"
## [267] "F149"                              "F149.5"
## [269] "F150"                              "F150.5"
## [271] "F151"                              "F151.5"
## [273] "F152"                              "F152.5"
## [275] "F153"                              "F153.5"
## [277] "F154"                              "F154.5"
## [279] "F155"                              "F155.5"
## [281] "F156"                              "F156.5"
## [283] "F157"                              "F157.5"
## [285] "F158"                              "F158.5"
## [287] "F159"                              "F159.5"
## [289] "F160"                              "F160.5"
## [291] "F161"                              "F161.5"
## [293] "F162"                              "F162.5"
## [295] "F163"                              "F163.5"
## [297] "F164"                              "F164.5"
## [299] "F165"                              "F165.5"
## [301] "F166"                              "F166.5"
## [303] "F167"                              "F167.5"
## [305] "F168"                              "F168.5"
## [307] "F169"                              "F169.5"
## [309] "F170"                              "F173"
## [311] "F173.5"                            "F174"
## [313] "F174.5"                            "F175"
## [315] "F175.5"                            "F176"
## [317] "F176.5"                            "F177"
## [319] "F177.5"                            "F178"
## [321] "F178.5"                            "F179"
## [323] "F179.5"                            "F180"
```

```
## [325] "F180.5"                     "F181"
## [327] "F181.5"                     "F182"
## [329] "F182.5"                     "F183"
## [331] "F183.5"                     "F184"
## [333] "F184.5"                     "F185"
## [335] "F185.5"                     "F186"
## [337] "F186.5"                     "F187"
## [339] "F187.5"                     "F188"
## [341] "F188.5"                     "F189"
## [343] "F189.5"                     "F190"
## [345] "F190.5"                     "F191"
## [347] "F191.5"                     "F192"
## [349] "F192.5"                     "F193"
## [351] "F193.5"                     "F194"
## [353] "F194.5"                     "F195"
## [355] "F195.5"                     "F196"
## [357] "F196.5"                     "F197"
## [359] "F197.5"                     "F198"
## [361] "F198.5"                     "F199"
## [363] "F199.5"                     "F200"
## [365] "F200.5"                     "F201"
## [367] "F201.5"                     "F202"
## [369] "F202.5"                     "F203"
## [371] "F203.5"                     "F204"
## [373] "F204.5"                     "F205"
## [375] "F205.5"                     "F206"
## [377] "F206.5"                     "F207"
## [379] "F207.5"                     "F208"
## [381] "F208.5"                     "F209"
## [383] "F209.5"                     "F210"
## [385] "F210.5"                     "F211"
## [387] "F211.5"                     "F212"
## [389] "F212.5"                     "F213"
## [391] "F213.5"                     "F214"
## [393] "F214.5"                     "F215"
## [395] "F215.5"                     "F216"
## [397] "F216.5"                     "F217"
## [399] "F217.5"                     "F218"
## [401] "F218.5"                     "F219"
## [403] "F219.5"                     "F220"
## [405] "F220.5"                     "F221"
## [407] "F221.5"                     "F222"
## [409] "F222.5"                     "F223"
## [411] "F223.5"                     "F224"
## [413] "F224.5"                     "F225"
## [415] "F225.5"                     "F226"
## [417] "F226.5"                     "F227"
## [419] "F227.5"                     "F228"
## [421] "F228.5"                     "F229"
## [423] "F229.5"                     "F230"
## [425] "F230.5"                     "F231"
## [427] "F231.5"                     "F232"
## [429] "F232.5"                     "F233"
## [431] "F233.5"                     "F234"
```

```
## [433] "F234.5"                       "F235"
## [435] "F235.5"                       "F236"
## [437] "F236.5"                       "F237"
## [439] "F237.5"                       "F238"
## [441] "F238.5"                       "F239"
## [443] "F239.5"                       "F240"
## [445] "F240.5"                       "F241"
## [447] "F241.5"                       "F242"
## [449] "F242.5"                       "F243"
## [451] "F243.5"                       "F244"
## [453] "F244.5"                       "F245"
## [455] "F245.5"                       "F246"
## [457] "F246.5"                       "F247"
## [459] "F247.5"                       "F248"
## [461] "F248.5"                       "F249"
## [463] "F249.5"                       "F250"
## [465] "F250.5"                       "F251"
## [467] "F251.5"                       "F252"
## [469] "F252.5"                       "F253"
## [471] "F253.5"                       "F254"
## [473] "F254.5"                       "F255"
## [475] "F255.5"                       "F256"
## [477] "F256.5"                       "F257"
## [479] "F257.5"                       "F258"
## [481] "F258.5"                       "F259"
## [483] "F259.5"                       "F260"
```

Create a dataframe only containing frequencies.

```
frequency_data <- (
  processed_data
  |> select(1, 5, 59:481)
)
frequency_data |> head()
```

```
## # A tibble: 6 x 425
##   fishNum species     F45 F45.5   F46 F46.5   F47 F47.5   F48 F48.5   F49 F49.5
##   <chr>   <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 LT001   lakeTrout -48.9 -47.5 -45.9 -44.2 -43.0 -41.7 -40.6 -39.6 -38.6 -37.7
## 2 LT001   lakeTrout -47.4 -47.6 -47.1 -45.5 -43.6 -41.6 -40.1 -39.0 -38.0 -37.2
## 3 LT001   lakeTrout -47.9 -49.3 -49.3 -47.2 -44.9 -42.7 -41.0 -39.5 -38.3 -37.2
## 4 LT001   lakeTrout -44.3 -45.7 -47.6 -48.8 -48.6 -46.7 -44.7 -42.9 -41.5 -40.3
## 5 LT001   lakeTrout -41.4 -42.3 -43.6 -44.7 -45.9 -46.6 -46.9 -46.7 -46.0 -45.1
## 6 LT001   lakeTrout -33.8 -33.7 -33.7 -33.5 -33.7 -33.8 -34.0 -34.1 -34.2 -34.2
## # i 413 more variables: F50 <dbl>, F50.5 <dbl>, F51 <dbl>, F51.5 <dbl>,
## #   F52 <dbl>, F52.5 <dbl>, F53 <dbl>, F53.5 <dbl>, F54 <dbl>, F54.5 <dbl>,
## #   F55 <dbl>, F55.5 <dbl>, F56 <dbl>, F56.5 <dbl>, F57 <dbl>, F57.5 <dbl>,
## #   F58 <dbl>, F58.5 <dbl>, F59 <dbl>, F59.5 <dbl>, F60 <dbl>, F60.5 <dbl>,
## #   F61 <dbl>, F61.5 <dbl>, F62 <dbl>, F62.5 <dbl>, F63 <dbl>, F63.5 <dbl>,
## #   F64 <dbl>, F64.5 <dbl>, F65 <dbl>, F65.5 <dbl>, F66 <dbl>, F66.5 <dbl>,
## #   F67 <dbl>, F67.5 <dbl>, F68 <dbl>, F68.5 <dbl>, F69 <dbl>, F69.5 <dbl>, ...
```

Basic Frequencies

Data cleaning

Since we notice the data of F90 to F170 are missing, we want to remove these frequencies columns from our dataset.

```
## removing columns 93(F90) to 253(F170)
frequency_data <- frequency_data |> select(-c(93:253))
## name(frequency_data)
```

Separate into three dataset for each species

```
LakeTrout <- frequency_data[frequency_data$species == "lakeTrout", ]
LakeWhiteFish <- frequency_data[frequency_data$species == "lakeWhitefish", ]
SmallmouthBass <- frequency_data[frequency_data$species == "smallmouthBass", ]
```

TS Mean Between Fish Species

We want to use the mean of target strength across all ping times of each fish across all frequencies to explore the dataset by fish species. We first create a dataframe that contains all mean values as above.

```
ts_mean_allTime_data <- (
  frequency_data
  |> group_by(fishNum, species)
  |> mutate(species = recode(species,
                             "lakeTrout" = "Lake Trout",
                             "lakeWhitefish" = "Lake Whitefish",
                             "smallmouthBass" = "Smallmouth Bass"))
  |> summarize(across(starts_with("F"), ~mean(., na.rm = TRUE), .names = "{.col}_mean"))
)
```

```
## `summarise()` has grouped output by 'fishNum'. You can override using the
## `.groups` argument.
```

```
ts_mean_allTime_data |> head()
```

```
## # A tibble: 6 x 264
## # Groups:   fishNum [6]
##   fishNum species    F45_mean F45.5_mean F46_mean F46.5_mean F47_mean F47.5_mean
##   <chr>   <chr>         <dbl>      <dbl>    <dbl>      <dbl>    <dbl>      <dbl>
## 1 LT001   Lake Trout    -46.3      -46.2    -46.3      -45.9    -45.4      -44.9
## 2 LT002   Lake Trout    -47.7      -47.3    -47.0      -46.6    -46.6      -46.4
## 3 LT003   Lake Trout    -47.3      -47.0    -46.8      -46.4    -46.3      -46.1
## 4 LT004   Lake Trout    -40.4      -40.3    -40.3      -40.0    -40.0      -39.8
## 5 LT005   Lake Trout    -60.5      -60.2    -60.1      -59.6    -59.7      -59.6
## 6 LT006   Lake Trout    -39.3      -39.2    -39.2      -38.9    -38.8      -38.6
## # i 256 more variables: F48_mean <dbl>, F48.5_mean <dbl>, F49_mean <dbl>,
## #   F49.5_mean <dbl>, F50_mean <dbl>, F50.5_mean <dbl>, F51_mean <dbl>,
## #   F51.5_mean <dbl>, F52_mean <dbl>, F52.5_mean <dbl>, F53_mean <dbl>,
## #   F53.5_mean <dbl>, F54_mean <dbl>, F54.5_mean <dbl>, F55_mean <dbl>,
## #   F55.5_mean <dbl>, F56_mean <dbl>, F56.5_mean <dbl>, F57_mean <dbl>,
## #   F57.5_mean <dbl>, F58_mean <dbl>, F58.5_mean <dbl>, F59_mean <dbl>,
## #   F59.5_mean <dbl>, F60_mean <dbl>, F60.5_mean <dbl>, F61_mean <dbl>, ...
```

We then melt the wide format dataframe into a long format dataframe for easy visualization.

```
ts_mean_allTime_data_long <- (
  ts_mean_allTime_data
```

```r
  |> melt(
    id.vars = c("fishNum", "species"),
    variable.name = "Frequency",
    value.name = "TS_mean_allTime"
  )
)

ts_mean_allTime_data_long$Frequency <- as.numeric(gsub("F([0-9.]+)_mean", "\\1",
                                                       ts_mean_allTime_data_long$Frequency))
ts_mean_allTime_data_long <- ts_mean_allTime_data_long |> arrange(fishNum, Frequency)
ts_mean_allTime_data_long |> head()
```

```
##   fishNum    species Frequency TS_mean_allTime
## 1   LT001 Lake Trout      45.0       -46.26963
## 2   LT001 Lake Trout      45.5       -46.18908
## 3   LT001 Lake Trout      46.0       -46.33088
## 4   LT001 Lake Trout      46.5       -45.91361
## 5   LT001 Lake Trout      47.0       -45.44261
## 6   LT001 Lake Trout      47.5       -44.92863
```

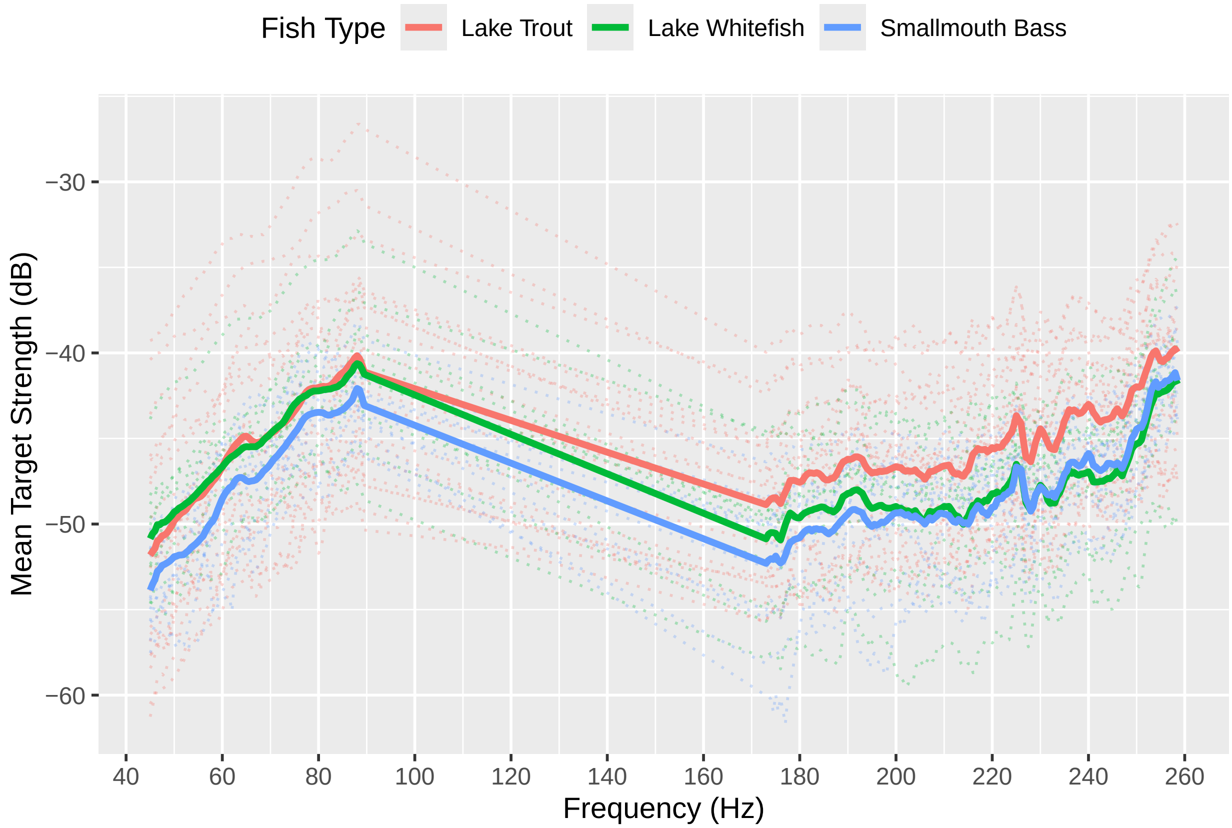
Plots We want to show different type of plots to discover the frequency patterns of each fish species.

First we want to plot an frequency response plot showing the mean target strength across different frequencies for different fish species with each fish response in the background.

```r
(
  ggplot(ts_mean_allTime_data_long, aes(x = Frequency, y = TS_mean_allTime, color = species))
  ## Indiviual fish (dotted lines)
  + geom_line(
    aes(group = interaction(fishNum, species)),
    linetype = "dotted",
    alpha = 0.3,
    linewidth = 0.5
  )
  ## Mean trends (bold lines)
  + stat_summary(
    fun = mean,
    geom = "line",
    aes(group = species),
    linewidth = 1.2
  )
  ## Aesthetics
  + scale_x_continuous(breaks = seq(40, 260, by = 20))
  + labs(
    x = "Frequency (Hz)",
    y = "Mean Target Strength (dB)",
    color = "Fish Type"
  )
  + theme(legend.position = "top")
)
```

```
## Warning: Removed 172 rows containing non-finite outside the scale range
## (`stat_summary()`).

## Warning: Removed 172 rows containing missing values or values outside the scale range
```

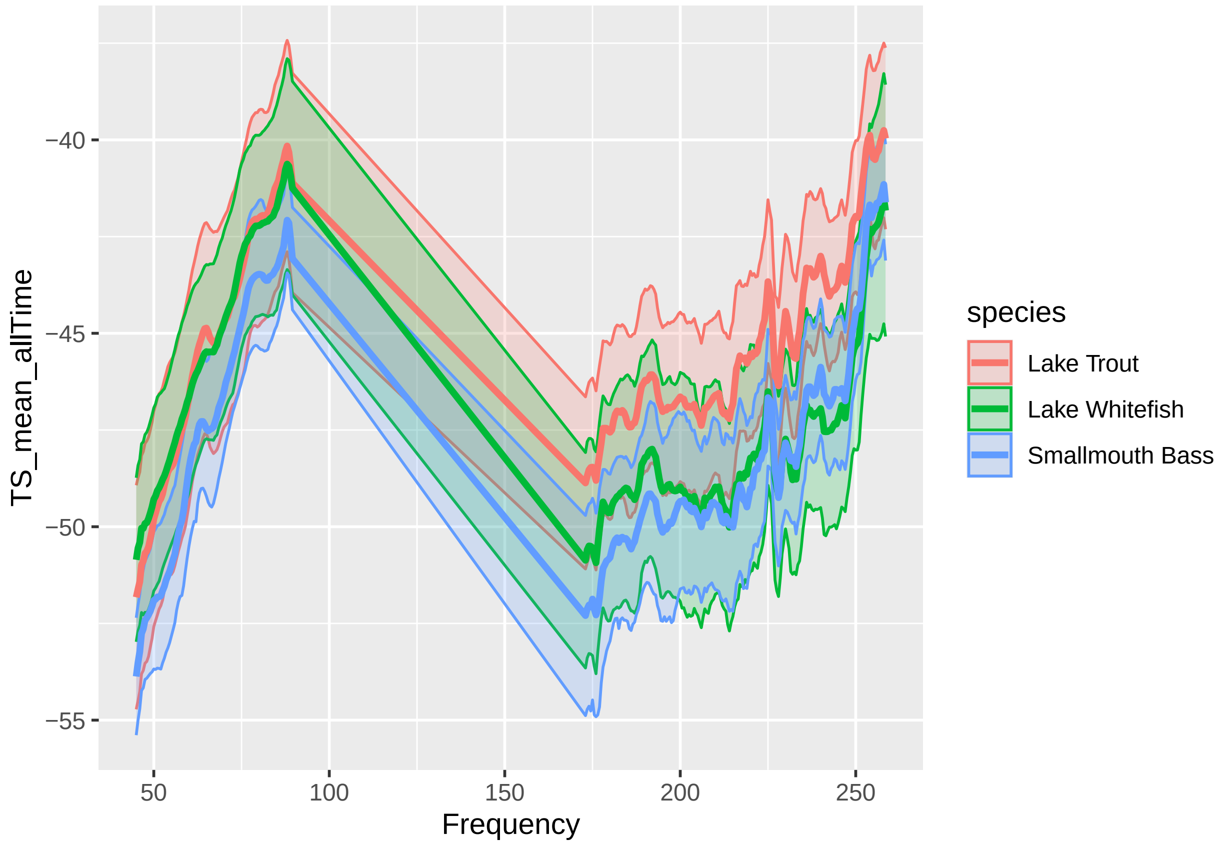
```
## (`geom_line()`).
```



Plot to observe the range of frequncies for each individual fish for each species.

```
(
  ggplot(ts_mean_allTime_data_long, aes(x = Frequency, y = TS_mean_allTime, color = species))
  + stat_summary(fun.data = mean_cl_normal, geom = "ribbon", alpha = 0.2, aes(fill = species))
  + stat_summary(fun = mean, geom = "line", linewidth = 1.2)
)
```

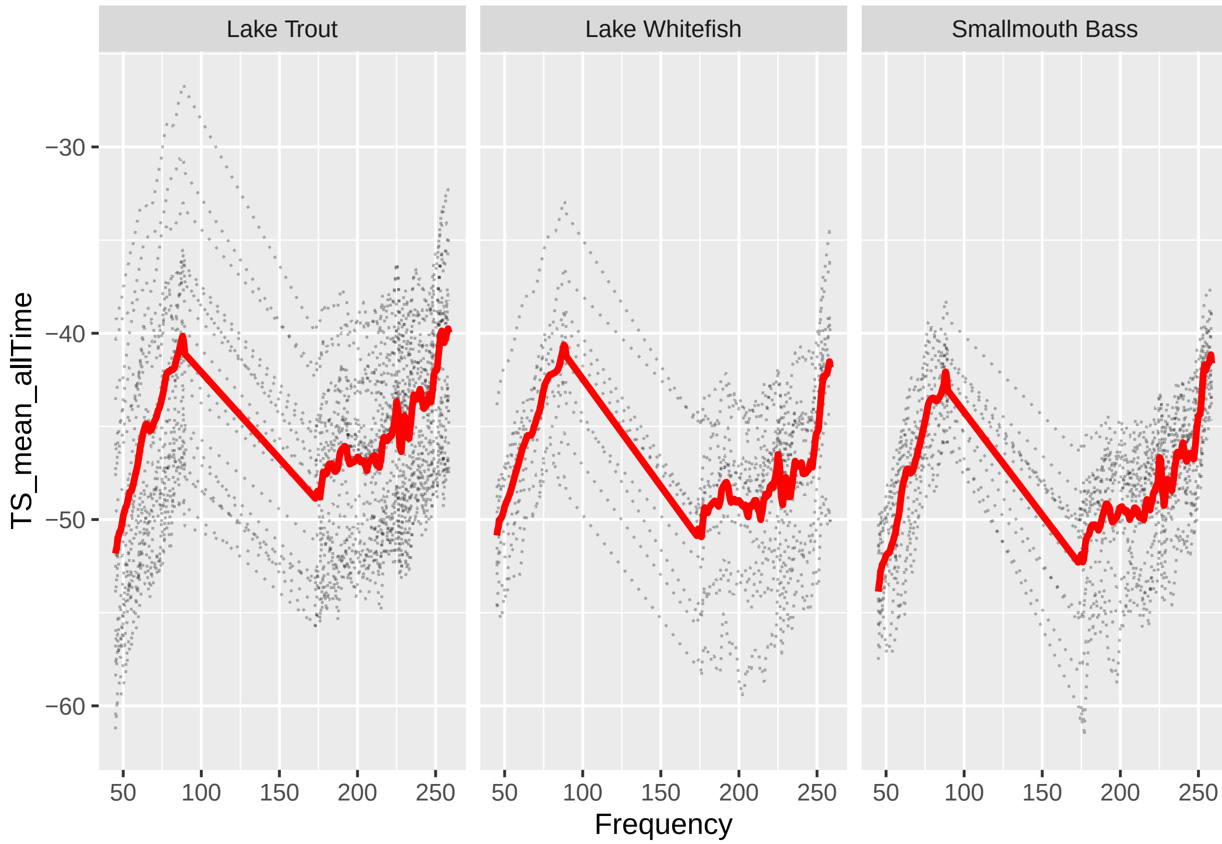
```
## Warning: Removed 172 rows containing non-finite outside the scale range
## (`stat_summary()`).
## Removed 172 rows containing non-finite outside the scale range
## (`stat_summary()`).
```



Plot to observe the range of frequencies of each species separtely.

```
(
  ggplot(ts_mean_allTime_data_long, aes(x = Frequency, y = TS_mean_allTime))
  + geom_line(aes(group = fishNum), linetype = "dotted", alpha = 0.3)
  + stat_summary(fun = mean, geom = "line", color = "red", linewidth = 1.2)
  + facet_wrap(~species)
)
```

```
## Warning: Removed 172 rows containing non-finite outside the scale range
## (`stat_summary()`).

## Warning: Removed 172 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



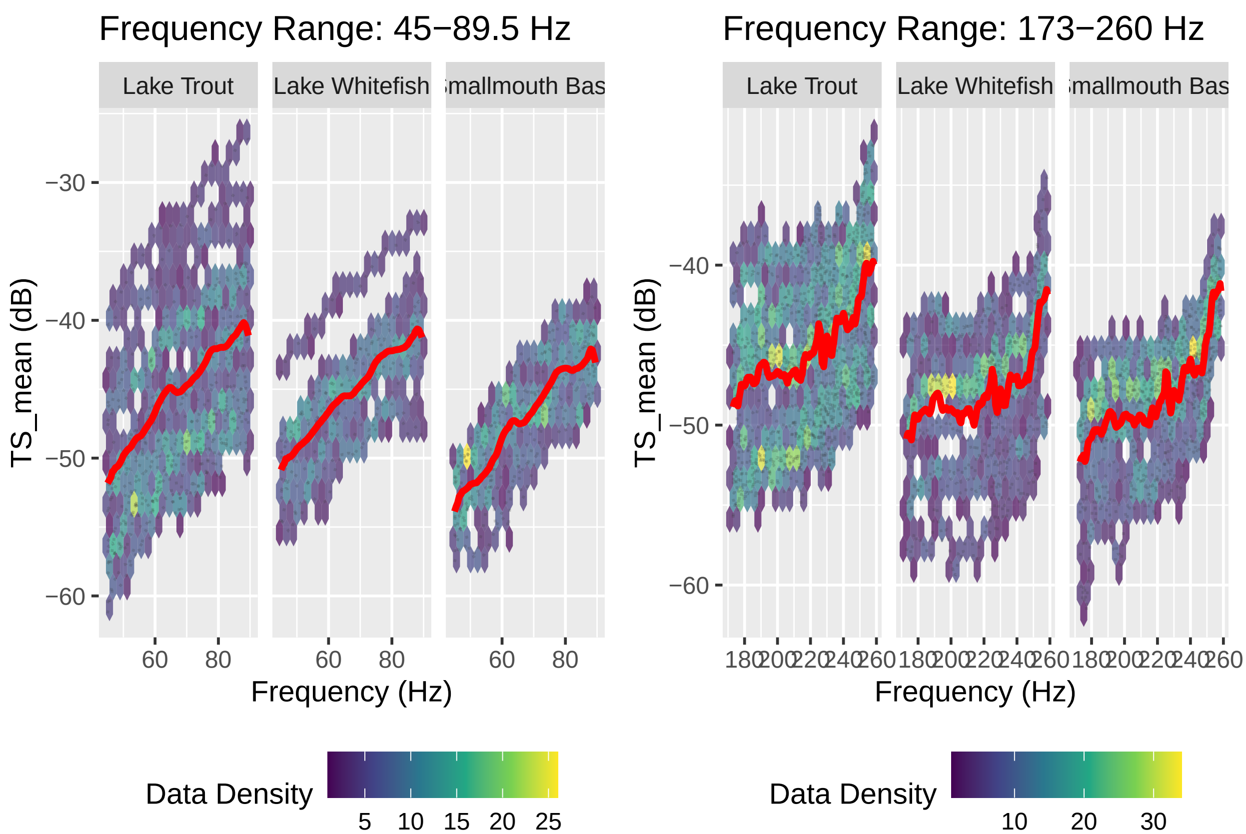
Density contour plot to show target strength measurements across differnt frequency ranges for each fish species.

```
## function to create density contour plot
create_frequency_density_plot <- function(data, title){
#' Create Target Strength Frequency Response Plots
#'
#' This function creates visualization of acoustic target strength data across frequency ranges for fis
#' The plot combines multiple visualization elements:
#'   - Hexagonal binning to show data density distribution
#'   - Individual fish measurements as dotted gray lines
#'   - Mean target strength trend line in red
#'
#' @param data A long format data frame containing columns in this order: fishNum, species, Frequency,
#' @param title String for the plot title

  colnames(data)[4] <- "TS_mean"

  (
    ggplot(data, aes(x = Frequency, y = TS_mean))
    ## hexbin density layer
    + geom_hex(aes(fill = after_stat(count)), bins = 20, alpha = 0.7)
    ## Indiviiual fish lines
    + geom_line(
      aes(group = interaction(fishNum, species)),
      linetype = "dotted",
      alpha = 0.2,
```

```r
      color = "gray20"
    )
    ## mean trend
    + stat_summary(
      fun = mean,
      geom = "line",
      color = "red",
      linewidth = 1.2
    )
    ## facet by fish type
    + facet_wrap(~species, nrow = 1)
    ## Aesthetics
    + scale_fill_viridis_c(name = "Data Density")
    + scale_x_continuous(breaks = seq(0, 260, by = 20))
    + labs(title = title, x = "Frequency (Hz)", y = "TS_mean (dB)")
    # + theme_minimal()
    + theme(legend.position = "bottom")
  )
}
```

```r
## split data into two frequency groups
ts_mean_allTime_low_long <- ts_mean_allTime_data_long |> filter(Frequency >= 45 & Frequency <= 89.5)
ts_mean_allTime_high_long <- ts_mean_allTime_data_long |> filter(Frequency >= 173 & Frequency <= 260)

# Create plots for both frequency ranges
plot_frequency_density_low <- create_frequency_density_plot(ts_mean_allTime_low_long, "Frequency Range:
plot_frequency_density_high <- create_frequency_density_plot(ts_mean_allTime_high_long, "Frequency Rang

# Arrange side-by-side
grid.arrange(plot_frequency_density_low, plot_frequency_density_high, ncol = 2)
```

```
## Warning: Removed 172 rows containing non-finite outside the scale range
## (`stat_binhex()`).

## Warning: Removed 172 rows containing non-finite outside the scale range
## (`stat_summary()`).

## Warning: Removed 172 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

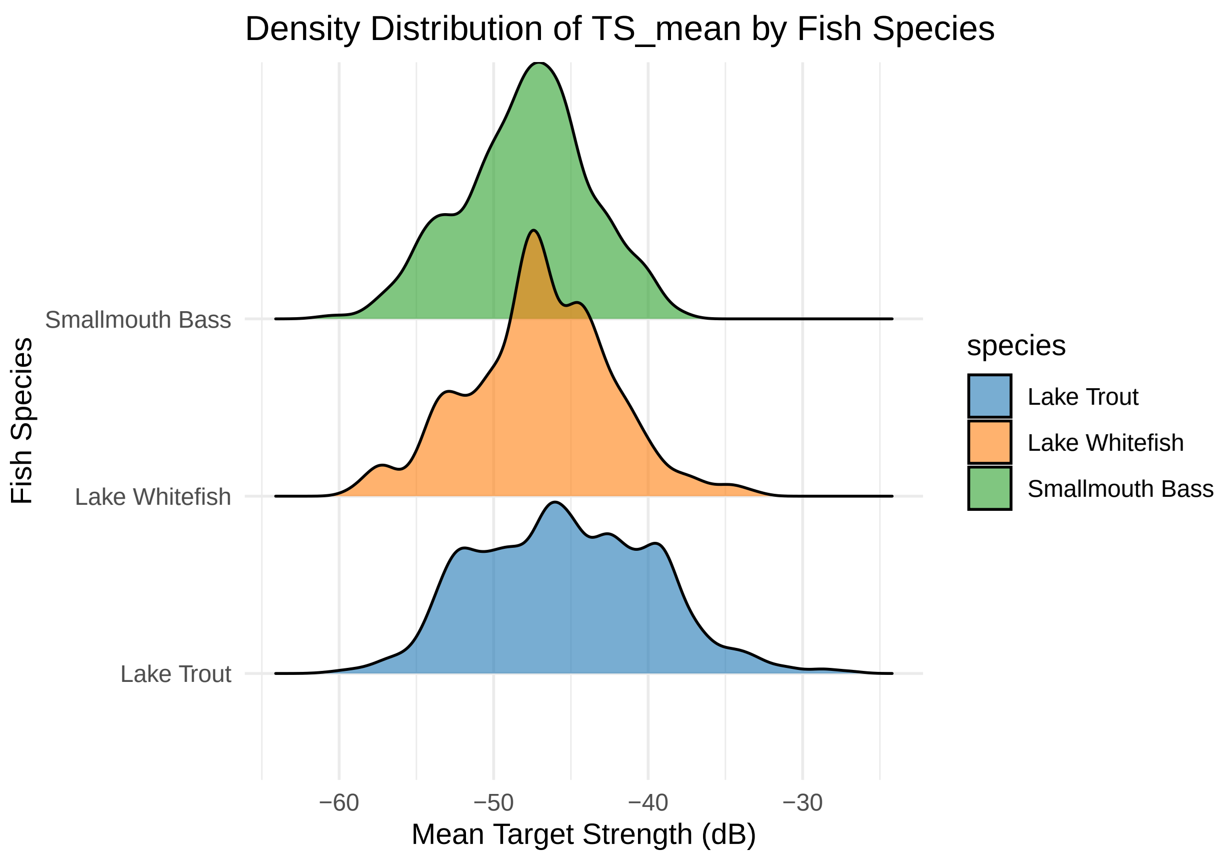


Create plot of density distribution of target strength by fish species.

```
(
  ggplot(ts_mean_allTime_data_long, aes(
    x = TS_mean_allTime,        # Numeric variable for density
    y = species,      # Categorical variable (fish species)
    fill = species    # Color by species
  ))
  ## density ridges
  + geom_density_ridges(alpha = 0.6, scale = 1.5)
  + labs(
    title = "Density Distribution of TS_mean by Fish Species",
    x = "Mean Target Strength (dB)",
    y = "Fish Species"
  )
  + theme_minimal()
  + scale_fill_manual(values = c("#1f77b4", "#ff7f0e", "#2ca02c"))
)
```

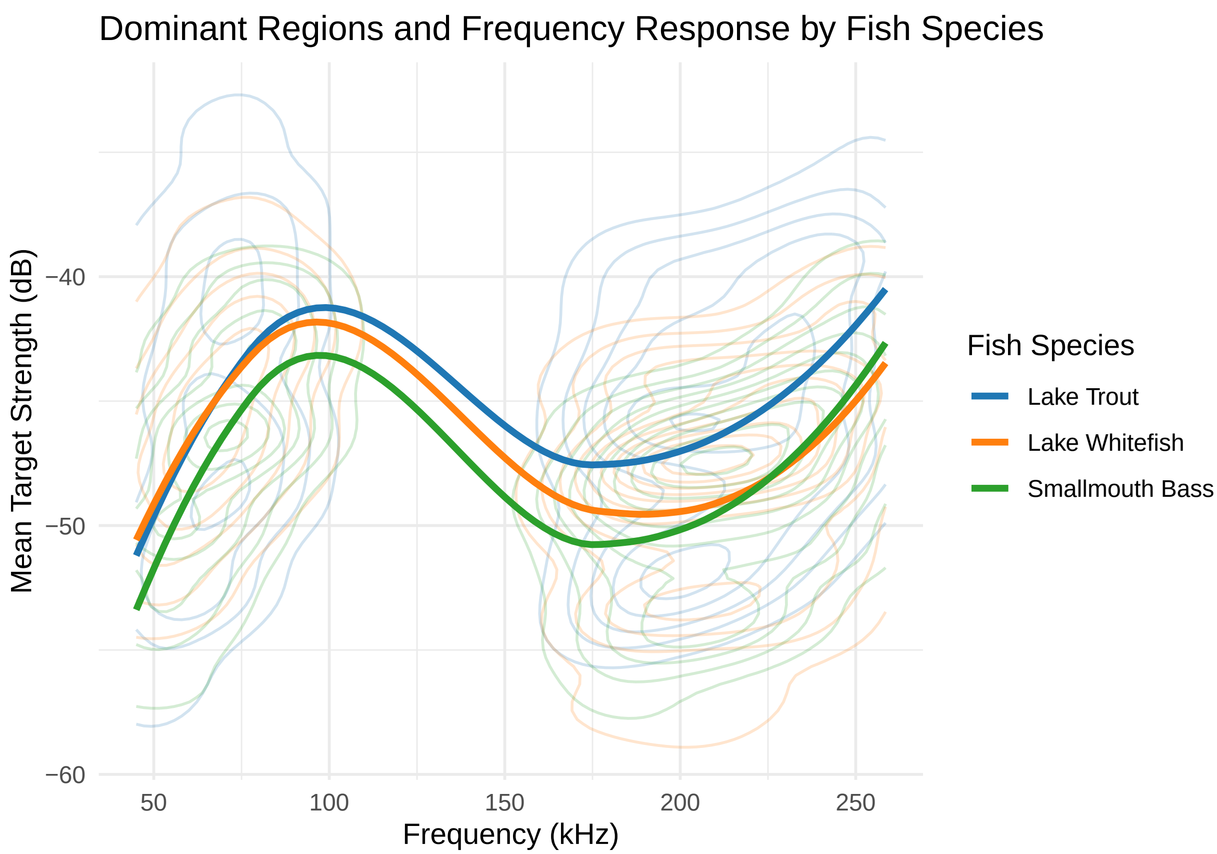
```
## Picking joint bandwidth of 0.808
```

```
## Warning: Removed 172 rows containing non-finite outside the scale range
## (`stat_density_ridges()`).
```



We create another density contour plot and we want to observe if there any potential dominant frequency response region by each fish species.

```
(
  ggplot(ts_mean_allTime_data_long, aes(x = Frequency, y = TS_mean_allTime, color = species))
  ## density contours
  + geom_density_2d(aes(fill = species), alpha = 0.2, contour_var = "density")
  ## frequency response trend lines
  + geom_smooth(
    method = "loess",
    formula = y ~ x,
    se = FALSE,  # Remove confidence bands
    linewidth = 1.2
  )
  ## Aesthetics
  + labs(
    title = "Dominant Regions and Frequency Response by Fish Species",
    x = "Frequency (kHz)",
    y = "Mean Target Strength (dB)",
    color = "Fish Species"
  )
  + scale_color_manual(values = c("#1f77b4", "#ff7f0e", "#2ca02c"))
  + scale_fill_manual(values = c("#1f77b4", "#ff7f0e", "#2ca02c"))
  + theme(legend.position = "bottom")
  + theme_minimal()
)
```



TS Mean In Fish Species

This section we want to investigate the potential trend / pattern in the frequency response in an indiviual fish species.

Lake Trout We first want to investigate the frequency response across different ping time for a single fish. We want to focus on `fishNum = LT001`.

```
LT001_frequency <- LakeTrout[LakeTrout$fishNum == "LT004", ]
# LT001_frequency <- LakeWhiteFish[LakeWhiteFish$fishNum == "LWF003", ]
# LT001_frequency <- SmallmouthBass[SmallmouthBass$fishNum == "SMB005", ]
```

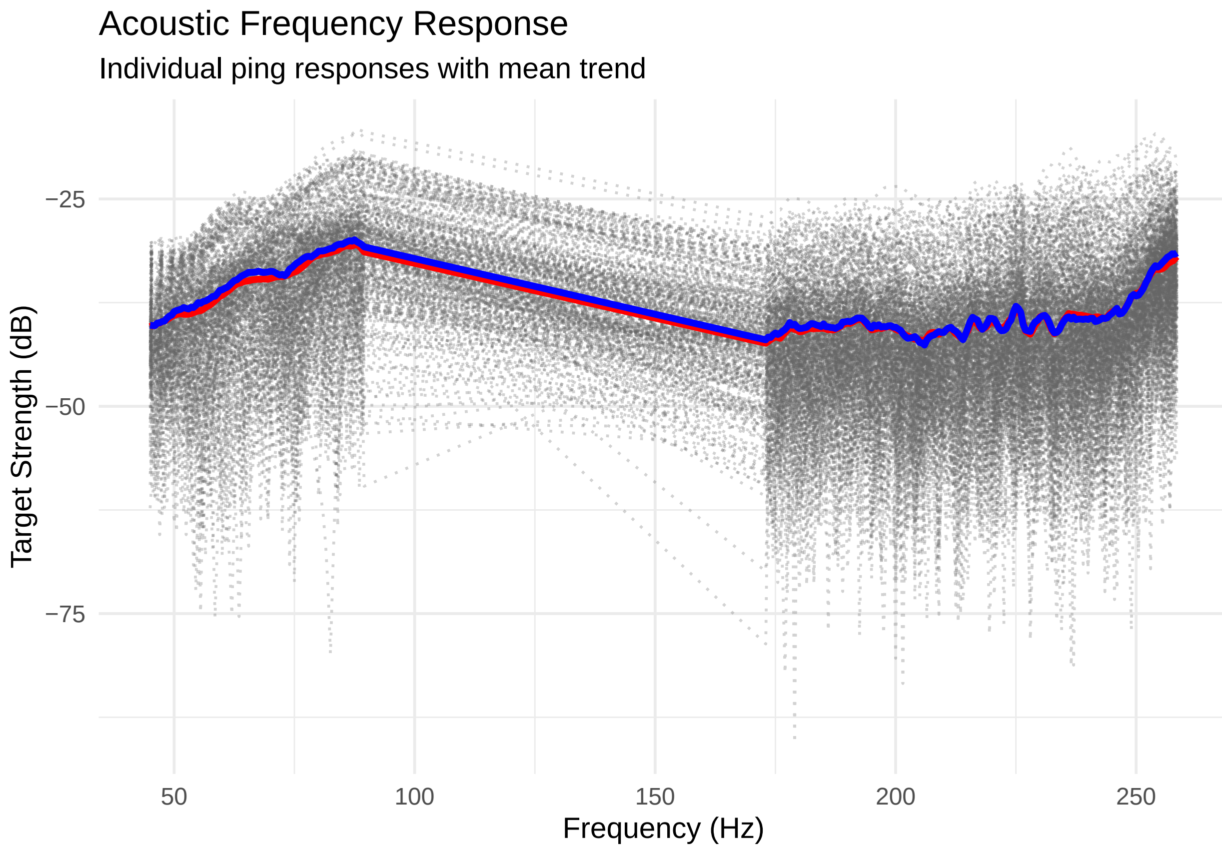
```
## melt dataframe into plottable format
LT001_frequency$ping_id <- 1:nrow(LT001_frequency)
LT001_frequency_long <- melt(
  LT001_frequency,
  id.vars = c("fishNum", "species", "ping_id"),
  variable.name = "Frequency",
  value.name = "TS"
)
LT001_frequency_long$Frequency <- as.numeric(gsub("F", "",
                                                  LT001_frequency_long$Frequency))
LT001_frequency_long |> head()
```

```
##   fishNum   species ping_id Frequency        TS
## 1   LT004 lakeTrout       1        45 -41.64990
## 2   LT004 lakeTrout       2        45 -44.13450
```

```
## 3   LT004 lakeTrout       3        45 -47.54812
## 4   LT004 lakeTrout       4        45 -36.01969
## 5   LT004 lakeTrout       5        45 -35.59207
## 6   LT004 lakeTrout       6        45 -34.82420
```

Plot to create visualization of frequency response of fish `LT001` across ping time.

```
(
  ggplot(LT001_frequency_long, aes(x = Frequency, y = TS))
  ## Indiviual ping time response as dotted lines
  + geom_line(aes(group = ping_id), linetype = "dotted", alpha = 0.3, color = "gray40")
  ## mean respinse as solid line
  + stat_summary(fun = mean, geom = "line", color = "red", linewidth = 1.2)
  + stat_summary(fun = median, geom = "line", color = "blue", linewidth = 1.2)
  ## confidence interval for mean
  # + stat_summary(fun.data = mean_cl_normal, geom = "ribbon", alpha = 0.2, fill = "red")
  + labs(
    title = "Acoustic Frequency Response",
    subtitle = "Individual ping responses with mean trend",
    x = "Frequency (Hz)",
    y = "Target Strength (dB)"
  )
  + theme(
    panel.grid.minor = element_blank(),
    legend.position = "none"
  )
  + theme_minimal()
)
```

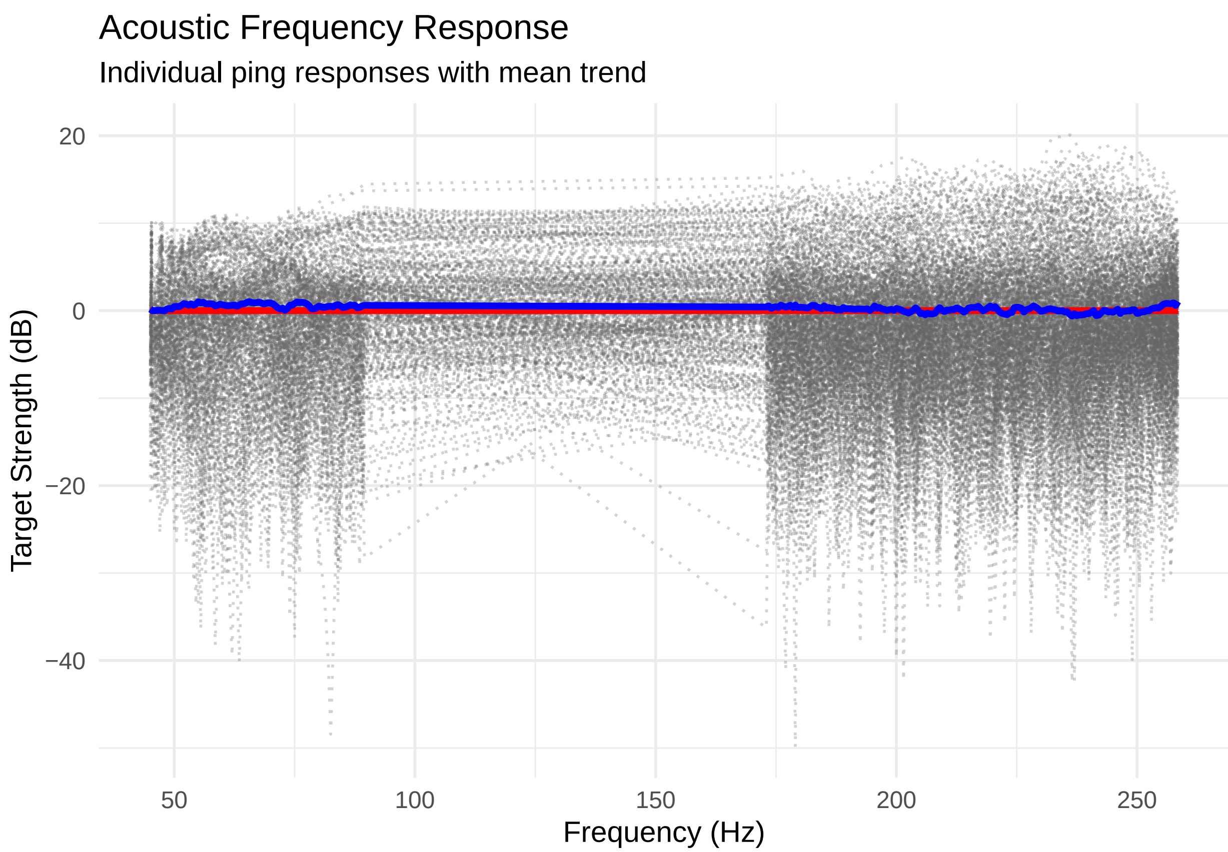


Next, we want to compte the difference between the frequency response from the mean from different ping time.

```
## compute mean
LT001_frequency_mean <- (
  LT001_frequency_long
  |> group_by(Frequency)
  |> summarize(TS_mean = mean(TS, na.rm = TRUE))
)

## left join and compute difference
LT001_frequency_long <- (
  LT001_frequency_long
  |> left_join(LT001_frequency_mean, by = "Frequency")
  |> mutate(diff_from_mean = TS - TS_mean)
)
LT001_frequency_long |> head()
```

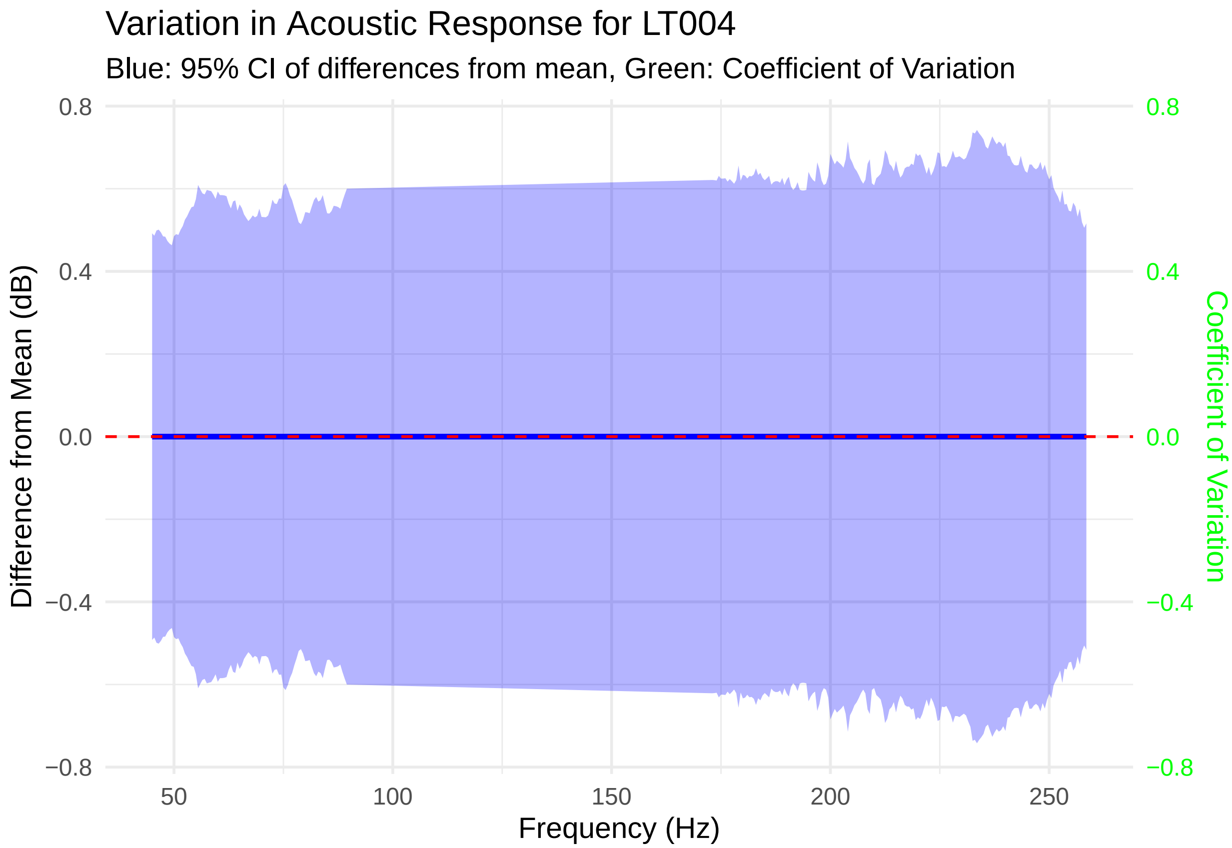
```
##   fishNum   species ping_id Frequency        TS   TS_mean diff_from_mean
## 1   LT004 lakeTrout       1        45 -41.64990 -40.39746      -1.252438
## 2   LT004 lakeTrout       2        45 -44.13450 -40.39746      -3.737038
## 3   LT004 lakeTrout       3        45 -47.54812 -40.39746      -7.150657
## 4   LT004 lakeTrout       4        45 -36.01969 -40.39746       4.377772
## 5   LT004 lakeTrout       5        45 -35.59207 -40.39746       4.805394
## 6   LT004 lakeTrout       6        45 -34.82420 -40.39746       5.573259
```

```
(
  ggplot(LT001_frequency_long, aes(x = Frequency, y = diff_from_mean))
  ## Indiviual ping time response as dotted lines
  + geom_line(aes(group = ping_id), linetype = "dotted", alpha = 0.3, color = "gray40")
  ## mean respinse as solid line
  + stat_summary(fun = mean, geom = "line", color = "red", linewidth = 1.2)
  + stat_summary(fun = median, geom = "line", color = "blue", linewidth = 1.2)
  ## confidence interval for mean
  # + stat_summary(fun.data = mean_cl_normal, geom = "ribbon", alpha = 0.2, fill = "red")
  + labs(
    title = "Acoustic Frequency Response",
    subtitle = "Individual ping responses with mean trend",
    x = "Frequency (Hz)",
    y = "Target Strength (dB)"
  )
  + theme(
    panel.grid.minor = element_blank(),
    legend.position = "none"
  )
  + theme_minimal()
)
```




```
# Calculate confidence intervals for the differences from mean
ci_data <- LT001_frequency_long %>%
  group_by(Frequency) %>%
  summarize(
```

```
    mean_diff = mean(diff_from_mean, na.rm = TRUE),
    sd_diff = sd(diff_from_mean, na.rm = TRUE),
    n = n(),
    # Calculate 95% confidence interval
    ci_lower = mean_diff - qt(0.975, n-1) * sd_diff / sqrt(n),
    ci_upper = mean_diff + qt(0.975, n-1) * sd_diff / sqrt(n),
    # Calculate variability metrics
    cv = sd_diff / abs(mean_diff + 0.0001), # Coefficient of variation (adding small constant to avoid
    range = max(diff_from_mean, na.rm = TRUE) - min(diff_from_mean, na.rm = TRUE)
  )

# Create a visualization of the confidence intervals
ggplot(ci_data, aes(x = Frequency)) +
  # Add confidence interval as ribbon
  geom_ribbon(aes(ymin = ci_lower, ymax = ci_upper), alpha = 0.3, fill = "blue") +
  # Add mean line
  geom_line(aes(y = mean_diff), color = "blue", linewidth = 1) +
  # Add zero reference line
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  # Add variability plot
  # geom_line(aes(y = cv), color = "green", linewidth = 1) +
  # Create second y-axis for coefficient of variation
  scale_y_continuous(
    name = "Difference from Mean (dB)",
    sec.axis = sec_axis(~ ., name = "Coefficient of Variation")
  ) +
  labs(
    title = paste("Variation in Acoustic Response for", unique(LT001_frequency_long$fishNum)),
    subtitle = "Blue: 95% CI of differences from mean, Green: Coefficient of Variation",
    x = "Frequency (Hz)"
  ) +
  theme_minimal() +
  theme(
    axis.title.y.right = element_text(color = "green"),
    axis.text.y.right = element_text(color = "green")
  )
```



Principal Component Analysis We want to perform PCA in Lake Trout and we want to see if specified frequencies is contributing strongly to the variance of Lake Trout.

```
## aggregating dataframe into mean frquencies by each fish and scale
LakeTrout_agg <- (
  LakeTrout
  |> group_by(fishNum)
  |> filter(fishNum != "LT008") ## contains missing bal
  |> summarise(across(starts_with("F"), mean, na.rm = TRUE))
  |> ungroup()
  |> dplyr::select(-fishNum)
  |> scale()
)
```

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(starts_with("F"), mean, na.rm = TRUE)`.
## i In group 1: `fishNum = "LT001"`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```r
LakeTrout_pca <- PCA(LakeTrout_agg, graph = FALSE)
LakeTrout_pca$eig
```

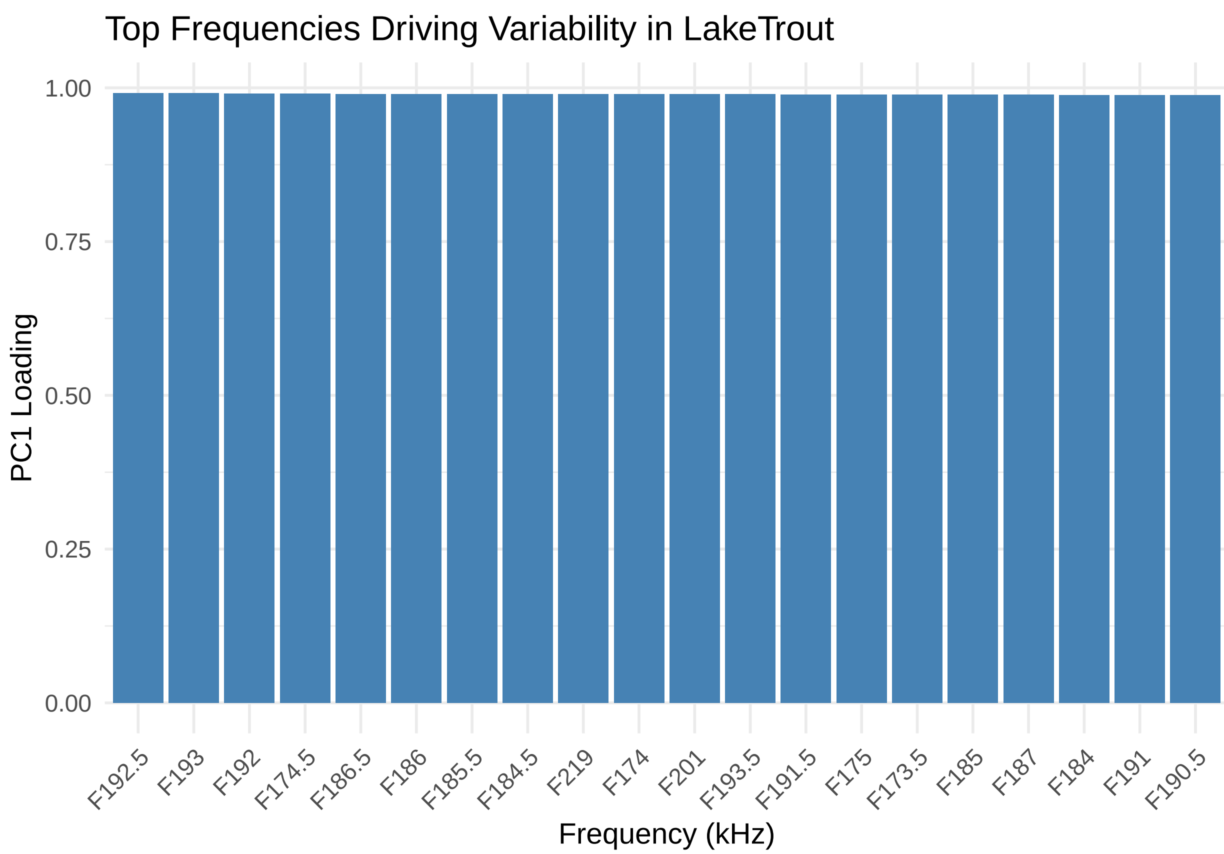
```
##          eigenvalue percentage of variance cumulative percentage of variance
## comp 1  244.21489472          93.211791879                          93.21179
## comp 2   11.98462387           4.574283918                          97.78608
## comp 3    1.27549045           0.486828415                          98.27290
## comp 4    1.05141984           0.401305283                          98.67421
## comp 5    0.91561696           0.349472122                          99.02368
## comp 6    0.60554067           0.231122395                          99.25480
## comp 7    0.41031229           0.156607745                          99.41141
## comp 8    0.26497132           0.101134090                          99.51255
## comp 9    0.23101183           0.088172453                          99.60072
## comp 10   0.22164218           0.084596254                          99.68531
## comp 11   0.17298864           0.066026199                          99.75134
## comp 12   0.13547558           0.051708235                          99.80305
## comp 13   0.12137783           0.046327415                          99.84938
## comp 14   0.10342138           0.039473811                          99.88885
## comp 15   0.09071791           0.034625156                          99.92348
## comp 16   0.07759171           0.029615158                          99.95309
## comp 17   0.06594847           0.025171171                          99.97826
## comp 18   0.03107329           0.011860036                          99.99012
## comp 19   0.02588105           0.009878265                         100.00000
```

```r
LakeTrout_loadings_pc1 <-(
  LakeTrout_pca$var$coord[, 1]
  |> as.data.frame()
  |> rename(Loading = "LakeTrout_pca$var$coord[, 1]")
  |> mutate(Frequency = rownames(LakeTrout_pca$var$coord))
  |> arrange(desc(abs(Loading)))
)
# loadings_pc1
LakeTrout_loadings_pc1_top_frequencies <- LakeTrout_loadings_pc1 |> head(20)
print(LakeTrout_loadings_pc1_top_frequencies)
```

```
##          Loading Frequency
## F192.5 0.9917246    F192.5
## F193   0.9911145      F193
## F192   0.9906835      F192
## F174.5 0.9906645    F174.5
## F186.5 0.9901207    F186.5
## F186   0.9898355      F186
## F185.5 0.9898325    F185.5
## F184.5 0.9898172    F184.5
## F219   0.9897545      F219
## F174   0.9895466      F174
## F201   0.9895126      F201
## F193.5 0.9894501    F193.5
## F191.5 0.9893288    F191.5
## F175   0.9888396      F175
## F173.5 0.9887690    F173.5
## F185   0.9887283      F185
## F187   0.9886458      F187
## F184   0.9884787      F184
```

```
## F191   0.9879446      F191
## F190.5 0.9878550    F190.5
```

```
ggplot(LakeTrout_loadings_pc1_top_frequencies, aes(x = reorder(Frequency, -abs(Loading)), y = Loading))
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Top Frequencies Driving Variability in LakeTrout",
       x = "Frequency (kHz)", y = "PC1 Loading") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
cor_mat <- cor(LakeTrout_agg) |> as.data.frame()
cor_mat |> head()
```

```
##             F45     F45.5       F46     F46.5       F47     F47.5       F48
## F45   1.0000000 0.9996056 0.9984781 0.9963892 0.9938490 0.9907588 0.9873195
## F45.5 0.9996056 1.0000000 0.9994420 0.9977076 0.9952274 0.9922555 0.9889508
## F46   0.9984781 0.9994420 1.0000000 0.9991627 0.9970995 0.9944152 0.9912062
## F46.5 0.9963892 0.9977076 0.9991627 1.0000000 0.9992304 0.9974915 0.9950857
## F47   0.9938490 0.9952274 0.9970995 0.9992304 1.0000000 0.9993983 0.9979255
## F47.5 0.9907588 0.9922555 0.9944152 0.9974915 0.9993983 1.0000000 0.9995045
##           F48.5       F49     F49.5       F50     F50.5       F51     F51.5
## F45   0.9854044 0.9867684 0.9875213 0.9879263 0.9864481 0.9846682 0.9823516
## F45.5 0.9872332 0.9884595 0.9890266 0.9890938 0.9875503 0.9859041 0.9838920
## F46   0.9899854 0.9910707 0.9913655 0.9911944 0.9896404 0.9880390 0.9861063
## F46.5 0.9942902 0.9948713 0.9945583 0.9936412 0.9917328 0.9898139 0.9879700
## F47   0.9972634 0.9976006 0.9969827 0.9954719 0.9931085 0.9906675 0.9892975
## F47.5 0.9990207 0.9990787 0.9982728 0.9964518 0.9938612 0.9912456 0.9897838
```

```
##              F52     F52.5       F53     F53.5       F54     F54.5       F55
## F45   0.9784427 0.9752323 0.9749124 0.9758274 0.9793099 0.9835452 0.9865828
## F45.5 0.9802075 0.9773009 0.9769559 0.9776729 0.9810197 0.9851049 0.9878685
## F46   0.9829682 0.9805702 0.9800609 0.9805476 0.9836110 0.9872657 0.9894391
## F46.5 0.9852680 0.9833675 0.9826165 0.9825025 0.9848524 0.9877872 0.9891125
## F47   0.9868246 0.9849949 0.9844540 0.9838143 0.9851698 0.9869363 0.9871556
## F47.5 0.9870537 0.9850853 0.9841743 0.9827655 0.9833158 0.9843635 0.9839844
##            F55.5       F56     F56.5       F57     F57.5       F58     F58.5
## F45   0.9882379 0.9880204 0.9868946 0.9849491 0.9826938 0.9790282 0.9742466
## F45.5 0.9893801 0.9891469 0.9880760 0.9865137 0.9845111 0.9811511 0.9767838
## F46   0.9904948 0.9904018 0.9896441 0.9886293 0.9868956 0.9840928 0.9805374
## F46.5 0.9894972 0.9889457 0.9883278 0.9880547 0.9869264 0.9850078 0.9824732
## F47   0.9869529 0.9858366 0.9851733 0.9853552 0.9846066 0.9832537 0.9813668
## F47.5 0.9833049 0.9818024 0.9810931 0.9815788 0.9811626 0.9803480 0.9789828
##              F59     F59.5       F60     F60.5       F61     F61.5       F62
## F45   0.9726974 0.9729084 0.9747031 0.9755038 0.9760930 0.9759473 0.9755610
## F45.5 0.9755277 0.9758662 0.9774892 0.9781902 0.9786538 0.9784071 0.9779916
## F46   0.9798032 0.9803538 0.9819924 0.9827455 0.9832965 0.9831439 0.9827446
## F46.5 0.9826682 0.9838174 0.9856276 0.9865472 0.9872740 0.9870454 0.9864016
## F47   0.9822194 0.9841360 0.9863297 0.9876743 0.9888113 0.9887047 0.9880671
## F47.5 0.9805389 0.9831913 0.9858067 0.9875771 0.9891117 0.9890382 0.9884401
##            F62.5       F63     F63.5       F64     F64.5       F65     F65.5
## F45   0.9749474 0.9763192 0.9770825 0.9752035 0.9721714 0.9702102 0.9690829
## F45.5 0.9773644 0.9785647 0.9791646 0.9773056 0.9743185 0.9721607 0.9707896
## F46   0.9819044 0.9827168 0.9829475 0.9809075 0.9778756 0.9758360 0.9744647
## F46.5 0.9853464 0.9856095 0.9853228 0.9830642 0.9798341 0.9779481 0.9766652
## F47   0.9870246 0.9868557 0.9862236 0.9840216 0.9806383 0.9788223 0.9773040
## F47.5 0.9876186 0.9873080 0.9864360 0.9839550 0.9801991 0.9782156 0.9766181
##              F66     F66.5       F67     F67.5       F68     F68.5       F69
## F45   0.9693286 0.9694006 0.9693635 0.9701633 0.9729604 0.9749580 0.9735983
## F45.5 0.9705929 0.9702044 0.9698984 0.9707058 0.9736817 0.9761107 0.9755301
## F46   0.9743693 0.9737714 0.9735222 0.9744489 0.9774257 0.9796496 0.9793211
## F46.5 0.9770049 0.9764910 0.9762480 0.9772222 0.9800893 0.9819148 0.9816792
## F47   0.9775724 0.9774012 0.9769491 0.9778995 0.9807999 0.9823260 0.9818747
## F47.5 0.9771173 0.9776056 0.9775672 0.9787093 0.9817307 0.9832829 0.9828455
##            F69.5       F70     F70.5       F71     F71.5       F72     F72.5
## F45   0.9713382 0.9687595 0.9662977 0.9653687 0.9674539 0.9686952 0.9681880
## F45.5 0.9738804 0.9715984 0.9692356 0.9684252 0.9705783 0.9715785 0.9708675
## F46   0.9777119 0.9753733 0.9730581 0.9723520 0.9745443 0.9755351 0.9748343
## F46.5 0.9798350 0.9772485 0.9749231 0.9742504 0.9764004 0.9774216 0.9770064
## F47   0.9796851 0.9768184 0.9742553 0.9732771 0.9755322 0.9765071 0.9761753
## F47.5 0.9804991 0.9774544 0.9747710 0.9737119 0.9758441 0.9766067 0.9762177
##              F73     F73.5       F74     F74.5       F75     F75.5       F76
## F45   0.9643414 0.9665943 0.9708638 0.9712180 0.9683276 0.9643715 0.9647953
## F45.5 0.9668647 0.9686072 0.9723308 0.9721909 0.9689943 0.9648905 0.9653424
## F46   0.9709055 0.9720941 0.9751185 0.9743950 0.9712244 0.9674893 0.9683512
## F46.5 0.9733277 0.9742328 0.9767563 0.9756761 0.9725105 0.9692312 0.9706053
## F47   0.9720753 0.9730024 0.9757517 0.9748336 0.9720097 0.9691752 0.9708732
## F47.5 0.9719202 0.9726238 0.9750131 0.9737850 0.9708668 0.9682587 0.9703683
##            F76.5       F77     F77.5       F78     F78.5       F79     F79.5
## F45   0.9670242 0.9686722 0.9695534 0.9688760 0.9672709 0.9637791 0.9592755
## F45.5 0.9680940 0.9699244 0.9710981 0.9702755 0.9682266 0.9642796 0.9592677
## F46   0.9718798 0.9737178 0.9749261 0.9740485 0.9719117 0.9679660 0.9626831
## F46.5 0.9742707 0.9756156 0.9766900 0.9757167 0.9735335 0.9697745 0.9645636
```

```
## F47   0.9746522 0.9757820 0.9768712 0.9759442 0.9737483 0.9699725 0.9651319
## F47.5 0.9742646 0.9754484 0.9765460 0.9756221 0.9733571 0.9695004 0.9644536
##             F80     F80.5       F81     F81.5       F82     F82.5       F83
## F45   0.9532876 0.9511268 0.9517823 0.9556073 0.9574697 0.9581440 0.9602334
## F45.5 0.9528631 0.9508473 0.9519910 0.9560693 0.9583415 0.9594043 0.9619342
## F46   0.9556484 0.9535186 0.9545732 0.9583147 0.9600952 0.9605580 0.9625559
## F46.5 0.9573167 0.9552776 0.9561451 0.9593051 0.9602870 0.9599690 0.9610987
## F47   0.9582841 0.9566196 0.9573339 0.9599221 0.9599652 0.9587307 0.9587519
## F47.5 0.9575909 0.9559968 0.9564101 0.9584435 0.9579054 0.9564316 0.9562671
##           F83.5       F84     F84.5       F85     F85.5       F86     F86.5
## F45   0.9590305 0.9571245 0.9573435 0.9556413 0.9552736 0.9567014 0.9565901
## F45.5 0.9612836 0.9596192 0.9601070 0.9586652 0.9583425 0.9598657 0.9597721
## F46   0.9620461 0.9602928 0.9609014 0.9597421 0.9595353 0.9612628 0.9612409
## F46.5 0.9603609 0.9585003 0.9593386 0.9586490 0.9585151 0.9606123 0.9607602
## F47   0.9574204 0.9554590 0.9562851 0.9555293 0.9550756 0.9575098 0.9580271
## F47.5 0.9549700 0.9530404 0.9540247 0.9535225 0.9531281 0.9556724 0.9565174
##             F87     F87.5       F88     F88.5       F89     F89.5      F173
## F45   0.9574097 0.9598496 0.9656046 0.9687327 0.9695074 0.9686576 0.8813356
## F45.5 0.9604769 0.9624892 0.9676919 0.9703363 0.9702609 0.9686341 0.8836910
## F46   0.9620685 0.9640473 0.9690843 0.9719469 0.9715406 0.9696071 0.8889647
## F46.5 0.9616690 0.9637103 0.9686105 0.9718969 0.9714925 0.9696770 0.8917619
## F47   0.9591011 0.9610838 0.9665572 0.9701155 0.9699025 0.9681438 0.8913108
## F47.5 0.9578091 0.9601245 0.9658643 0.9696870 0.9695886 0.9676789 0.8911367
##          F173.5      F174    F174.5      F175    F175.5      F176    F176.5
## F45   0.8814772 0.8810421 0.8834045 0.8798306 0.8670267 0.8489260 0.8490983
## F45.5 0.8839726 0.8833285 0.8849828 0.8815669 0.8688293 0.8506633 0.8509417
## F46   0.8891655 0.8878578 0.8880548 0.8845918 0.8722406 0.8541429 0.8545605
## F46.5 0.8915497 0.8893972 0.8882559 0.8849893 0.8730501 0.8546588 0.8552488
## F47   0.8899080 0.8869906 0.8858323 0.8827736 0.8707320 0.8517708 0.8524922
## F47.5 0.8891889 0.8856715 0.8842160 0.8813939 0.8693702 0.8499375 0.8507694
##            F177    F177.5      F178    F178.5      F179    F179.5      F180
## F45   0.8640953 0.8766028 0.8846390 0.8806816 0.8717291 0.8623332 0.8631477
## F45.5 0.8655766 0.8782449 0.8862591 0.8829334 0.8746567 0.8658677 0.8669211
## F46   0.8685362 0.8809136 0.8886960 0.8864997 0.8795427 0.8717874 0.8733244
## F46.5 0.8685224 0.8803432 0.8876107 0.8862746 0.8805027 0.8738929 0.8755586
## F47   0.8650202 0.8762372 0.8834710 0.8823050 0.8766323 0.8703765 0.8722364
## F47.5 0.8627065 0.8735401 0.8806376 0.8795769 0.8739760 0.8683094 0.8704206
##          F180.5      F181    F181.5      F182    F182.5      F183    F183.5
## F45   0.8654168 0.8704839 0.8677580 0.8695321 0.8748813 0.8826885 0.8861074
## F45.5 0.8691107 0.8734816 0.8702658 0.8715584 0.8764993 0.8842940 0.8875695
## F46   0.8757515 0.8794224 0.8759184 0.8769556 0.8814191 0.8893540 0.8920466
## F46.5 0.8779132 0.8808335 0.8775290 0.8784588 0.8828894 0.8914265 0.8936175
## F47   0.8743899 0.8776247 0.8748340 0.8762460 0.8812647 0.8900543 0.8924706
## F47.5 0.8724474 0.8753421 0.8726361 0.8740840 0.8792534 0.8882148 0.8909228
##            F184    F184.5      F185    F185.5      F186    F186.5      F187
## F45   0.8902087 0.8899975 0.8860738 0.8879036 0.8827378 0.8788590 0.8749909
## F45.5 0.8913532 0.8910028 0.8871659 0.8889536 0.8842905 0.8805274 0.8769116
## F46   0.8955622 0.8951811 0.8913020 0.8928568 0.8886567 0.8848546 0.8813356
## F46.5 0.8968817 0.8968364 0.8932051 0.8943968 0.8906447 0.8863434 0.8827183
## F47   0.8954086 0.8951577 0.8910980 0.8922472 0.8889687 0.8847599 0.8815075
## F47.5 0.8939633 0.8940610 0.8902659 0.8914994 0.8885209 0.8842388 0.8811923
##          F187.5      F188    F188.5      F189    F189.5      F190    F190.5
## F45   0.8670948 0.8736199 0.8741425 0.8752081 0.8774032 0.8768977 0.8797257
## F45.5 0.8688889 0.8750852 0.8753203 0.8762507 0.8786040 0.8783831 0.8814973
```

```
## F46   0.8731630 0.8788583 0.8789595 0.8798426 0.8823442 0.8824717 0.8853695
## F46.5 0.8743315 0.8797977 0.8801899 0.8811792 0.8840614 0.8852015 0.8881963
## F47   0.8732634 0.8784382 0.8789759 0.8800912 0.8832836 0.8851057 0.8886781
## F47.5 0.8730249 0.8782978 0.8790229 0.8801188 0.8835117 0.8858142 0.8896689
##            F191    F191.5      F192    F192.5      F193    F193.5      F194
## F45   0.8781818 0.8836352 0.8860715 0.8896868 0.8883366 0.8853000 0.8757910
## F45.5 0.8801050 0.8854476 0.8875401 0.8909004 0.8897272 0.8869828 0.8776980
## F46   0.8839128 0.8886737 0.8903256 0.8939006 0.8934861 0.8914922 0.8822523
## F46.5 0.8867526 0.8911166 0.8924067 0.8961781 0.8963447 0.8947025 0.8854790
## F47   0.8869798 0.8906492 0.8915331 0.8956676 0.8963286 0.8947016 0.8854759
## F47.5 0.8879203 0.8911790 0.8916840 0.8957233 0.8961188 0.8939614 0.8850069
##          F194.5      F195    F195.5      F196    F196.5      F197    F197.5
## F45   0.8725687 0.8690538 0.8647146 0.8586165 0.8520763 0.8417781 0.8461424
## F45.5 0.8740107 0.8700650 0.8657652 0.8592377 0.8529951 0.8426675 0.8470977
## F46   0.8778125 0.8728617 0.8683224 0.8611408 0.8556435 0.8453604 0.8497123
## F46.5 0.8805018 0.8747703 0.8699064 0.8619751 0.8569878 0.8464986 0.8506163
## F47   0.8807893 0.8750064 0.8700973 0.8616713 0.8564983 0.8453100 0.8489539
## F47.5 0.8805758 0.8749140 0.8702178 0.8616769 0.8562981 0.8446120 0.8480549
##            F198    F198.5      F199    F199.5      F200    F200.5      F201
## F45   0.8469048 0.8442648 0.8436048 0.8546200 0.8658122 0.8707601 0.8738943
## F45.5 0.8479169 0.8452724 0.8445146 0.8554607 0.8666075 0.8718054 0.8750523
## F46   0.8505671 0.8479337 0.8470889 0.8577636 0.8687639 0.8742796 0.8777576
## F46.5 0.8514519 0.8481006 0.8466847 0.8567185 0.8680714 0.8745390 0.8786997
## F47   0.8497251 0.8456834 0.8440536 0.8534941 0.8653302 0.8727755 0.8773325
## F47.5 0.8488437 0.8443403 0.8425504 0.8515378 0.8633237 0.8711308 0.8757898
##          F201.5      F202    F202.5      F203    F203.5      F204    F204.5
## F45   0.8672311 0.8678472 0.8590229 0.8540326 0.8499832 0.8479032 0.8438723
## F45.5 0.8683216 0.8689468 0.8602023 0.8549402 0.8508695 0.8486307 0.8442134
## F46   0.8713431 0.8722081 0.8634012 0.8579811 0.8538991 0.8514670 0.8465810
## F46.5 0.8730778 0.8743564 0.8651726 0.8597230 0.8556393 0.8530446 0.8477358
## F47   0.8720325 0.8732063 0.8632490 0.8576906 0.8533077 0.8511095 0.8464261
## F47.5 0.8705795 0.8716900 0.8613167 0.8558843 0.8515709 0.8492915 0.8446808
##            F205    F205.5      F206    F206.5      F207    F207.5      F208
## F45   0.8438744 0.8530104 0.8576317 0.8559934 0.8598971 0.8641325 0.8581053
## F45.5 0.8439907 0.8531335 0.8576581 0.8560001 0.8596715 0.8641711 0.8584839
## F46   0.8460521 0.8550152 0.8593349 0.8573716 0.8603399 0.8649209 0.8596328
## F46.5 0.8472091 0.8558373 0.8594970 0.8572746 0.8601430 0.8653929 0.8607746
## F47   0.8461299 0.8545386 0.8575653 0.8550196 0.8583274 0.8641634 0.8598480
## F47.5 0.8445502 0.8526318 0.8553434 0.8526969 0.8560408 0.8622156 0.8585043
##          F208.5      F209    F209.5      F210    F210.5      F211    F211.5
## F45   0.8436658 0.8337528 0.8364794 0.8436289 0.8498493 0.8563687 0.8546032
## F45.5 0.8439564 0.8338900 0.8366601 0.8440762 0.8505136 0.8573123 0.8558820
## F46   0.8450302 0.8343373 0.8364582 0.8437715 0.8500952 0.8577998 0.8572976
## F46.5 0.8463106 0.8345945 0.8351880 0.8421351 0.8485884 0.8575730 0.8581897
## F47   0.8455187 0.8334624 0.8332370 0.8399656 0.8463606 0.8558189 0.8566819
## F47.5 0.8445140 0.8322650 0.8314311 0.8379433 0.8444859 0.8542300 0.8552172
##            F212    F212.5      F213    F213.5      F214    F214.5      F215
## F45   0.8574469 0.8608665 0.8598258 0.8585295 0.8571250 0.8631257 0.8639994
## F45.5 0.8588150 0.8621803 0.8606818 0.8591382 0.8575760 0.8634090 0.8644789
## F46   0.8607082 0.8641004 0.8612378 0.8588554 0.8568647 0.8626664 0.8643587
## F46.5 0.8621732 0.8654905 0.8610791 0.8572028 0.8542674 0.8596574 0.8617882
## F47   0.8607755 0.8641384 0.8593709 0.8545897 0.8505908 0.8554287 0.8572417
## F47.5 0.8594094 0.8626875 0.8575466 0.8520458 0.8470016 0.8509378 0.8524917
##          F215.5      F216    F216.5      F217    F217.5      F218    F218.5
```

```
## F45   0.8699774 0.8655652 0.8544398 0.8476136 0.8406149 0.8480860 0.8670758
## F45.5 0.8707068 0.8665095 0.8552435 0.8485549 0.8418781 0.8490780 0.8675653
## F46   0.8717291 0.8687689 0.8577830 0.8512354 0.8448554 0.8513976 0.8686805
## F46.5 0.8706073 0.8692267 0.8585103 0.8521968 0.8458881 0.8520777 0.8686840
## F47   0.8666128 0.8659400 0.8557386 0.8503158 0.8445178 0.8507204 0.8669878
## F47.5 0.8624892 0.8628803 0.8533305 0.8485553 0.8432201 0.8493403 0.8651829
##            F219    F219.5      F220    F220.5      F221    F221.5      F222
## F45   0.8802872 0.8748360 0.8632657 0.8626737 0.8535032 0.8512038 0.8523668
## F45.5 0.8805856 0.8748207 0.8630290 0.8624480 0.8539021 0.8517935 0.8532623
## F46   0.8813972 0.8758788 0.8644312 0.8642142 0.8567709 0.8545042 0.8554348
## F46.5 0.8816106 0.8763989 0.8651434 0.8653597 0.8588058 0.8562988 0.8565390
## F47   0.8797898 0.8743640 0.8628216 0.8633505 0.8570355 0.8547846 0.8552688
## F47.5 0.8778475 0.8722983 0.8607982 0.8617637 0.8557516 0.8537733 0.8541533
##          F222.5      F223    F223.5      F224    F224.5      F225    F225.5
## F45   0.8552939 0.8669005 0.8771203 0.8898932 0.8924879 0.8868608 0.8736940
## F45.5 0.8557416 0.8663970 0.8763824 0.8891025 0.8920151 0.8866512 0.8738698
## F46   0.8563034 0.8647374 0.8745288 0.8873953 0.8910152 0.8866663 0.8746342
## F46.5 0.8559894 0.8623433 0.8719194 0.8852852 0.8901199 0.8873201 0.8760995
## F47   0.8538928 0.8594401 0.8685292 0.8819444 0.8876229 0.8856241 0.8747964
## F47.5 0.8521490 0.8565710 0.8649505 0.8786048 0.8846443 0.8830639 0.8725628
##            F226    F226.5      F227    F227.5      F228    F228.5      F229
## F45   0.8633489 0.8433197 0.8393526 0.8378730 0.8374162 0.8377274 0.8339260
## F45.5 0.8635477 0.8433070 0.8394750 0.8382265 0.8373129 0.8372811 0.8334922
## F46   0.8640279 0.8430699 0.8385424 0.8369201 0.8355700 0.8355429 0.8313964
## F46.5 0.8648117 0.8424705 0.8368232 0.8350905 0.8342845 0.8352030 0.8308267
## F47   0.8630437 0.8391946 0.8322364 0.8305419 0.8309879 0.8337894 0.8302466
## F47.5 0.8607421 0.8363091 0.8290740 0.8277729 0.8288704 0.8322977 0.8287450
##          F229.5      F230    F230.5      F231    F231.5      F232    F232.5
## F45   0.8467275 0.8565762 0.8656352 0.8576426 0.8609871 0.8572262 0.8440392
## F45.5 0.8463118 0.8562155 0.8656927 0.8583998 0.8618968 0.8579669 0.8445246
## F46   0.8438914 0.8539772 0.8642258 0.8586509 0.8623284 0.8579843 0.8437248
## F46.5 0.8424901 0.8525663 0.8632583 0.8591185 0.8628905 0.8585533 0.8429876
## F47   0.8412986 0.8507444 0.8603071 0.8552141 0.8594089 0.8558589 0.8397491
## F47.5 0.8391313 0.8481947 0.8577755 0.8530665 0.8572351 0.8538578 0.8377520
##            F233    F233.5      F234    F234.5      F235    F235.5      F236
## F45   0.8333317 0.8366802 0.8408595 0.8462262 0.8568916 0.8535567 0.8490322
## F45.5 0.8338303 0.8366961 0.8407625 0.8457496 0.8561975 0.8528897 0.8481310
## F46   0.8324186 0.8341979 0.8380735 0.8428337 0.8530254 0.8497737 0.8452907
## F46.5 0.8300360 0.8302295 0.8338234 0.8393436 0.8500462 0.8474654 0.8438331
## F47   0.8260897 0.8254907 0.8288292 0.8354098 0.8469268 0.8450107 0.8414716
## F47.5 0.8236486 0.8222212 0.8251759 0.8324872 0.8442446 0.8426725 0.8393617
##          F236.5      F237    F237.5      F238    F238.5      F239    F239.5
## F45   0.8407288 0.8307361 0.8302941 0.8371227 0.8303338 0.8195819 0.8048591
## F45.5 0.8398633 0.8299363 0.8297264 0.8370725 0.8308804 0.8199862 0.8049946
## F46   0.8380717 0.8287120 0.8286068 0.8365459 0.8310531 0.8199924 0.8047792
## F46.5 0.8373178 0.8286419 0.8283989 0.8363450 0.8307988 0.8195812 0.8042413
## F47   0.8346809 0.8263534 0.8260764 0.8336427 0.8280510 0.8174253 0.8024139
## F47.5 0.8325299 0.8244403 0.8244350 0.8318175 0.8260560 0.8151364 0.7999009
##            F240    F240.5      F241    F241.5      F242    F242.5      F243
## F45   0.8012034 0.8119111 0.8232119 0.8205556 0.8241461 0.8246914 0.8189326
## F45.5 0.8008836 0.8112852 0.8221050 0.8196869 0.8238432 0.8253657 0.8195798
## F46   0.7999548 0.8093506 0.8196557 0.8183505 0.8240588 0.8271032 0.8206720
## F46.5 0.7987836 0.8072055 0.8173106 0.8170406 0.8239599 0.8282860 0.8207863
## F47   0.7962648 0.8037351 0.8139576 0.8133122 0.8206538 0.8250512 0.8175594
```

```
## F47.5 0.7929423 0.7998048 0.8097374 0.8094554 0.8172251 0.8220156 0.8143599
##           F243.5      F244    F244.5      F245    F245.5      F246    F246.5
## F45    0.8008462 0.7913604 0.7917053 0.7797381 0.8038966 0.7972928 0.7809050
## F45.5 0.8018708 0.7921430 0.7922611 0.7796716 0.8041698 0.7975864 0.7815122
## F46    0.8036701 0.7936316 0.7927343 0.7795146 0.8037592 0.7972902 0.7816752
## F46.5 0.8044560 0.7945117 0.7924711 0.7785146 0.8018015 0.7952315 0.7799219
## F47    0.8011025 0.7901081 0.7875208 0.7741376 0.7977175 0.7916288 0.7761487
## F47.5 0.7981415 0.7873565 0.7845421 0.7708189 0.7941973 0.7884395 0.7729537
##             F247    F247.5      F248    F248.5      F249    F249.5      F250
## F45    0.7592057 0.7496167 0.7466897 0.7476320 0.7650295 0.7769755 0.7854903
## F45.5 0.7594649 0.7497131 0.7468825 0.7476831 0.7649499 0.7769557 0.7853718
## F46    0.7598120 0.7499402 0.7474876 0.7476179 0.7640064 0.7764159 0.7849735
## F46.5 0.7584944 0.7483636 0.7469218 0.7475968 0.7632353 0.7763789 0.7849754
## F47    0.7548691 0.7442494 0.7426250 0.7435024 0.7593330 0.7722924 0.7810309
## F47.5 0.7520089 0.7415516 0.7405823 0.7419201 0.7573149 0.7703213 0.7791222
##           F250.5      F251    F251.5      F252    F252.5      F253    F253.5
## F45    0.7966254 0.8005155 0.8128418 0.8139394 0.8129073 0.8135229 0.8364870
## F45.5 0.7967630 0.8008432 0.8132548 0.8152107 0.8139524 0.8146988 0.8372672
## F46    0.7972893 0.8017937 0.8140174 0.8158160 0.8140244 0.8146342 0.8359513
## F46.5 0.7978170 0.8019768 0.8136590 0.8152507 0.8135780 0.8144136 0.8344865
## F47    0.7940566 0.7976597 0.8100188 0.8120208 0.8101070 0.8116255 0.8323072
## F47.5 0.7925026 0.7959519 0.8081004 0.8105999 0.8087133 0.8106671 0.8311265
##             F254    F254.5      F255    F255.5      F256    F256.5      F257
## F45    0.8500803 0.8512987 0.8462635 0.8490578 0.8640935 0.8672562 0.8587325
## F45.5 0.8508589 0.8523015 0.8478659 0.8511738 0.8655596 0.8681721 0.8602313
## F46    0.8507433 0.8533112 0.8498270 0.8531286 0.8659914 0.8677359 0.8602365
## F46.5 0.8505427 0.8535687 0.8508017 0.8532012 0.8638956 0.8651321 0.8586462
## F47    0.8478591 0.8506880 0.8468212 0.8487535 0.8594512 0.8613242 0.8562778
## F47.5 0.8469944 0.8498602 0.8456638 0.8467361 0.8564972 0.8589619 0.8551134
##           F257.5      F258    F258.5
## F45    0.8309907 0.8401535 0.8497166
## F45.5 0.8328851 0.8414124 0.8504737
## F46    0.8333260 0.8414542 0.8501398
## F46.5 0.8323266 0.8405863 0.8494949
## F47    0.8315986 0.8391294 0.8475065
## F47.5 0.8312648 0.8386555 0.8468001
```

Lake White Fish

```
is.na(LakeWhiteFish) |> sum()
```

PCA

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

```
## aggregating dataframe into mean frquencies by each fish and scale
LakeWhiteFish_agg <- (
  LakeWhiteFish
  |> group_by(fishNum)
  |> summarise(across(starts_with("F"), mean, na.rm = TRUE))
  |> ungroup()
  |> dplyr::select(-fishNum)
  |> scale()
)
```

```r
LakeWhiteFish_pca <- PCA(LakeWhiteFish_agg, graph = FALSE)
LakeWhiteFish_pca$eig
```

```
##          eigenvalue percentage of variance cumulative percentage of variance
## comp 1  231.9991409            88.54929044                          88.54929
## comp 2   22.5857938             8.62053199                          97.16982
## comp 3    2.9915858             1.14182662                          98.31165
## comp 4    1.2695144             0.48454748                          98.79620
## comp 5    1.2381232             0.47256612                          99.26876
## comp 6    0.6835889             0.26091178                          99.52967
## comp 7    0.5099489             0.19463698                          99.72431
## comp 8    0.3544067             0.13526972                          99.85958
## comp 9    0.2432860             0.09285724                          99.95244
## comp 10   0.1246115             0.04756163                         100.00000
```

```r
LakeWhiteFish_loadings_pc1 <-(
  LakeWhiteFish_pca$var$coord[, 1]
  |> as.data.frame()
  |> rename(Loading = "LakeWhiteFish_pca$var$coord[, 1]")
  |> mutate(Frequency = rownames(LakeWhiteFish_pca$var$coord))
  |> arrange(desc(abs(Loading)))
)
# loadings_pc1
LakeWhiteFish_loadings_pc1_top_frequencies <- LakeWhiteFish_loadings_pc1 |> head(20)
print(LakeWhiteFish_loadings_pc1_top_frequencies)
```

```
##          Loading Frequency
## F177   0.9925150      F177
## F177.5 0.9923810    F177.5
## F181.5 0.9917162    F181.5
## F179   0.9916320      F179
## F176.5 0.9913896    F176.5
## F179.5 0.9913677    F179.5
## F178   0.9911983      F178
## F191.5 0.9910568    F191.5
## F192   0.9909740      F192
## F178.5 0.9909690    F178.5
## F182   0.9903404      F182
## F181   0.9902244      F181
## F187   0.9899987      F187
## F192.5 0.9897638    F192.5
## F189   0.9895014      F189
## F191   0.9894162      F191
## F180   0.9893366      F180
## F189.5 0.9893036    F189.5
## F180.5 0.9886951    F180.5
## F176   0.9886713      F176
```

```r
is.na(SmallmouthBass) |> sum()
```

Smallmouth Bass

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

```r
## aggregating dataframe into mean frquencies by each fish and scale
SmallmouthBass_agg <- (
  SmallmouthBass
  |> group_by(fishNum)
  |> summarise(across(starts_with("F"), mean, na.rm = TRUE))
  |> ungroup()
  |> dplyr::select(-fishNum)
  |> scale()
)

SmallmouthBass_pca <- PCA(SmallmouthBass_agg, graph = FALSE)
SmallmouthBass_pca$eig
```

```
##          eigenvalue percentage of variance cumulative percentage of variance
## comp 1  213.3640082            81.4366443                          81.43664
## comp 2   23.9921634             9.1573143                          90.59396
## comp 3   11.5097038             4.3930167                          94.98698
## comp 4    3.9931711             1.5241111                          96.51109
## comp 5    2.5775514             0.9837982                          97.49488
## comp 6    2.0178735             0.7701807                          98.26507
## comp 7    1.3838672             0.5281936                          98.79326
## comp 8    1.1034893             0.4211791                          99.21444
## comp 9    0.9147164             0.3491284                          99.56357
## comp 10   0.6858171             0.2617623                          99.82533
## comp 11   0.4576387             0.1746713                         100.00000
```

```r
SmallmouthBass_loadings_pc1 <-(
  SmallmouthBass_pca$var$coord[, 1]
  |> as.data.frame()
  |> rename(Loading = "SmallmouthBass_pca$var$coord[, 1]")
  |> mutate(Frequency = rownames(SmallmouthBass_pca$var$coord))
  |> arrange(desc(abs(Loading)))
)
# loadings_pc1
SmallmouthBass_loadings_pc1_top_frequencies <- SmallmouthBass_loadings_pc1 |> head(20)
print(SmallmouthBass_loadings_pc1_top_frequencies)
```

```
##          Loading Frequency
## F208   0.9897487      F208
## F209   0.9890043      F209
## F209.5 0.9878411    F209.5
## F208.5 0.9869398    F208.5
## F184.5 0.9859878    F184.5
## F210.5 0.9857557    F210.5
## F210   0.9845922      F210
## F212   0.9834752      F212
## F184   0.9829311      F184
## F212.5 0.9822498    F212.5
## F211   0.9818086      F211
## F185   0.9815360      F185
## F207.5 0.9813842    F207.5
## F183.5 0.9804217    F183.5
## F211.5 0.9798649    F211.5
## F205   0.9793735      F205
```

```
## F205.5 0.9790389    F205.5
## F213.5 0.9784301    F213.5
## F207   0.9775466      F207
## F214   0.9772022      F214
```

```
(SmallmouthBass
  |> group_by(fishNum)
  |> summarise(across(starts_with("F"), mean, na.rm = TRUE))
  |> ungroup()
  |> dplyr::select(-fishNum)
 |> summarise(across(starts_with("F"), mean, na.rm = TRUE))
 )
```

```
## # A tibble: 1 x 262
##     F45 F45.5   F46 F46.5   F47 F47.5   F48 F48.5   F49 F49.5   F50 F50.5   F51
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 -53.9 -53.5 -53.2 -52.8 -52.6 -52.4 -52.4 -52.3 -52.2 -52.1 -51.9 -51.9 -51.8
## # i 249 more variables: F51.5 <dbl>, F52 <dbl>, F52.5 <dbl>, F53 <dbl>,
## #   F53.5 <dbl>, F54 <dbl>, F54.5 <dbl>, F55 <dbl>, F55.5 <dbl>, F56 <dbl>,
## #   F56.5 <dbl>, F57 <dbl>, F57.5 <dbl>, F58 <dbl>, F58.5 <dbl>, F59 <dbl>,
## #   F59.5 <dbl>, F60 <dbl>, F60.5 <dbl>, F61 <dbl>, F61.5 <dbl>, F62 <dbl>,
## #   F62.5 <dbl>, F63 <dbl>, F63.5 <dbl>, F64 <dbl>, F64.5 <dbl>, F65 <dbl>,
## #   F65.5 <dbl>, F66 <dbl>, F66.5 <dbl>, F67 <dbl>, F67.5 <dbl>, F68 <dbl>,
## #   F68.5 <dbl>, F69 <dbl>, F69.5 <dbl>, F70 <dbl>, F70.5 <dbl>, F71 <dbl>, ...
```

Overall PCA

```
LT_mean_frequency <- (
  LakeTrout
  |> group_by(fishNum)
  |> filter(fishNum != "LT008")
  |> dplyr::select(matches("^F(1[7-9][0-9](\\.[0-9])?|2[0-5][0-9](\\.[0-9])?|260(\\.[0-9])?)$"))
  |> summarise(across(starts_with("F"), mean, na.rm = TRUE))
  |> ungroup()
  |> dplyr::select(-fishNum)
  |> summarise(across(starts_with("F"), mean, na.rm = TRUE))
)
```

```
## Adding missing grouping variables: `fishNum`
```

```
LT_mean_frequency |> head()
```

```
## # A tibble: 1 x 172
##    F173 F173.5  F174 F174.5  F175 F175.5  F176 F176.5  F177 F177.5  F178 F178.5
##   <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl>
## 1 -48.9  -48.7 -48.5  -48.5 -48.5  -48.6 -48.8  -48.4 -48.1  -47.8 -47.5  -47.5
## # i 160 more variables: F179 <dbl>, F179.5 <dbl>, F180 <dbl>, F180.5 <dbl>,
## #   F181 <dbl>, F181.5 <dbl>, F182 <dbl>, F182.5 <dbl>, F183 <dbl>,
## #   F183.5 <dbl>, F184 <dbl>, F184.5 <dbl>, F185 <dbl>, F185.5 <dbl>,
## #   F186 <dbl>, F186.5 <dbl>, F187 <dbl>, F187.5 <dbl>, F188 <dbl>,
## #   F188.5 <dbl>, F189 <dbl>, F189.5 <dbl>, F190 <dbl>, F190.5 <dbl>,
## #   F191 <dbl>, F191.5 <dbl>, F192 <dbl>, F192.5 <dbl>, F193 <dbl>,
## #   F193.5 <dbl>, F194 <dbl>, F194.5 <dbl>, F195 <dbl>, F195.5 <dbl>, ...
```

```
LWF_mean_frequency <- (
  LakeWhiteFish
```

```r
  |> group_by(fishNum)
  |> dplyr::select(matches("^F(1[7-9][0-9](\\.[0-9])?|2[0-5][0-9](\\.[0-9])?|260(\\.[0-9])?)$"))
  |> summarise(across(starts_with("F"), mean, na.rm = TRUE))
  |> ungroup()
  |> dplyr::select(-fishNum)
  |> summarise(across(starts_with("F"), mean, na.rm = TRUE))
)
```

```
## Adding missing grouping variables: `fishNum`
```

```r
LWF_mean_frequency |> head()
```

```
## # A tibble: 1 x 172
##    F173 F173.5  F174 F174.5  F175 F175.5  F176 F176.5  F177 F177.5  F178 F178.5
##   <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl>
## 1 -50.9  -50.6 -50.5  -50.5 -50.5  -50.8 -50.9  -50.5 -50.0  -49.7 -49.4  -49.5
## # i 160 more variables: F179 <dbl>, F179.5 <dbl>, F180 <dbl>, F180.5 <dbl>,
## #   F181 <dbl>, F181.5 <dbl>, F182 <dbl>, F182.5 <dbl>, F183 <dbl>,
## #   F183.5 <dbl>, F184 <dbl>, F184.5 <dbl>, F185 <dbl>, F185.5 <dbl>,
## #   F186 <dbl>, F186.5 <dbl>, F187 <dbl>, F187.5 <dbl>, F188 <dbl>,
## #   F188.5 <dbl>, F189 <dbl>, F189.5 <dbl>, F190 <dbl>, F190.5 <dbl>,
## #   F191 <dbl>, F191.5 <dbl>, F192 <dbl>, F192.5 <dbl>, F193 <dbl>,
## #   F193.5 <dbl>, F194 <dbl>, F194.5 <dbl>, F195 <dbl>, F195.5 <dbl>, ...
```

```r
SB_mean_frequency <- (
  SmallmouthBass
  |> group_by(fishNum)
  |> dplyr::select(matches("^F(1[7-9][0-9](\\.[0-9])?|2[0-5][0-9](\\.[0-9])?|260(\\.[0-9])?)$"))
  |> summarise(across(starts_with("F"), mean, na.rm = TRUE))
  |> ungroup()
  |> dplyr::select(-fishNum)
  |> summarise(across(starts_with("F"), mean, na.rm = TRUE))
)
```

```
## Adding missing grouping variables: `fishNum`
```

```r
SB_mean_frequency |> head()
```

```
## # A tibble: 1 x 172
##    F173 F173.5  F174 F174.5  F175 F175.5  F176 F176.5  F177 F177.5  F178 F178.5
##   <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>  <dbl>
## 1 -52.3  -52.1 -52.0  -52.1 -51.9  -52.1 -52.3  -52.2 -51.8  -51.4 -51.1  -51.0
## # i 160 more variables: F179 <dbl>, F179.5 <dbl>, F180 <dbl>, F180.5 <dbl>,
## #   F181 <dbl>, F181.5 <dbl>, F182 <dbl>, F182.5 <dbl>, F183 <dbl>,
## #   F183.5 <dbl>, F184 <dbl>, F184.5 <dbl>, F185 <dbl>, F185.5 <dbl>,
## #   F186 <dbl>, F186.5 <dbl>, F187 <dbl>, F187.5 <dbl>, F188 <dbl>,
## #   F188.5 <dbl>, F189 <dbl>, F189.5 <dbl>, F190 <dbl>, F190.5 <dbl>,
## #   F191 <dbl>, F191.5 <dbl>, F192 <dbl>, F192.5 <dbl>, F193 <dbl>,
## #   F193.5 <dbl>, F194 <dbl>, F194.5 <dbl>, F195 <dbl>, F195.5 <dbl>, ...
```

```r
## prepare fish species PCA comparison data
prepare_species_PCA_data <- function(species, mean_freq_df, loadings_df) {

  ## convery mean frequency data to long format
  species_long <- melt(
    mean_freq_df,
    variable.name = "Frequency",
```

```r
    value.name = "TS"

  )
  ## removing "F" from frequency and make it numerical
  species_long$Frequency <- as.numeric(gsub("F", "",
                                            species_long$Frequency))
  loadings_df$Frequency <- as.numeric(gsub("F", "",
                                           loadings_df$Frequency))

  ## Add species name
  species_long$Species <- species

  ## join with loadings
  species_combined <- merge(
    species_long,
    loadings_df[, c("Frequency", "Loading")],
    by = "Frequency",
    all.x = TRUE
  )

  return(species_combined)
}
```

```r
## prepare PCA data for each species
LakeTrout_PCA_data <- prepare_species_PCA_data(
  "Lake Trout",
  LT_mean_frequency,
  LakeTrout_loadings_pc1
)
```

```
## No id variables; using all as measure variables
```

```r
LakeWhiteFish_PCA_data <- prepare_species_PCA_data(
  "Lake Whitefish",
  LWF_mean_frequency,
  LakeWhiteFish_loadings_pc1
)
```

```
## No id variables; using all as measure variables
```

```r
SmallmouthBass_PCA_data <- prepare_species_PCA_data(
  "Smallmouth Bass",
  SB_mean_frequency,
  SmallmouthBass_loadings_pc1
)
```

```
## No id variables; using all as measure variables
```

```r
all_species_PCA_data <- rbind(
  LakeTrout_PCA_data,
  LakeWhiteFish_PCA_data,
  SmallmouthBass_PCA_data
)
```

```r
all_species_PCA_data |> head()
```

```
##   Frequency        TS     Species    Loading
```

```
## 1     173.0 -48.86743 Lake Trout 0.9871391
## 2     173.5 -48.68770 Lake Trout 0.9887690
## 3     174.0 -48.53021 Lake Trout 0.9895466
## 4     174.5 -48.46775 Lake Trout 0.9906645
## 5     175.0 -48.45928 Lake Trout 0.9888396
## 6     175.5 -48.63499 Lake Trout 0.9829606
```

```r
## create plots to show PCA laodings and TS_mean to visulize the PCA data
species_colors <- c(
  "Lake Trout" = "#1f77b4",
  "Lake Whitefish" = "#2ca02c",
  "Smallmouth Bass" = "#ff7f0e"
)

## PCA loading plots
all_species_PCA_plot <- (
  ggplot(all_species_PCA_data,
         aes(x = Frequency, y = Loading, color = Species))
  + geom_line(linewidth = 0.5)
  + scale_color_manual(values = species_colors)
  + labs(
    title = "PC1 Loadings by species",
    y = "PC1 Loading"
  )
  + theme(
    axis.title.x = element_blank(),
    axis.text.x = element_blank(),
    legend.position = "none"
  )
  + theme_minimal()
)

## TS_mean plots
all_species_TS_mean_plot <- (
  ggplot(all_species_PCA_data,
         aes(x = Frequency, y = TS, color = Species))
  + geom_line(linewidth = 0.8)
  + scale_color_manual(values = species_colors)
  + labs(
    title = "Mean Target Strength by Species",
    y = "Target Strength",
    x = "Frequency"
  )
  + theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "bottom"
  )
  + theme_minimal()
)

## combined plots
all_species_PCA_combined_plot <- (
  all_species_PCA_plot / all_species_TS_mean_plot
  + plot_layout(heights = c(1, 1.2))
```

```
)
all_species_PCA_combined_plot
```