MUSSEL ABM - Cluster Output Summary

Serra Örey

Last compiled on 15 March, 2023

Contents

On 15 March, 2023 I worked on(and if you are at the html possibly knitted) this script.

Here is a R Markdown script(or the output) for:
summary document for Orey et.al. outputs to use in Mussel ABM

Please follow the *table of content (outline)* for keeping track of the steps. It must be on the left side.

Statistical tests for the explanatory variables.

1. Housekeeping

Check the needed packages:

```
#check Rversion:
R.version.string
```

```
## [1] "R version 4.0.2 (2020-06-22)"
```

```
#important for rmarkdown
knitr::opts_chunk$set(echo = T, eval = T, fig.keep="all",cache = T) #DEFAULT:echo = T, eval = T, #eval=
knitr::opts_knit$set(root.dir = "M:/01_Projects/")
#getwd()
```

```r
#housekeeping
##packages####
options(rlib_downstream_check = FALSE) #dont check package downstream

library(scales) #for show_col
library(patchwork)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --v ggplot2 3.4.0
## v tibble  3.1.6      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.2
```

```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()    masks scales::discard()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
```

```r
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:patchwork':
##
##     align_plots
```

```r
#library(tidyr, lib.loc = "/usr/lib/R/site-library")

#require(parallel)
#require(doParallel)
#require(foreach)

#require(sp)
#require(sf)
#require(rnaturalearth)

library(MetBrewer)
```

```
## Warning: package 'MetBrewer' was built under R version 4.0.5
```

```
library(knitr)

#library(factoextra)
library(RColorBrewer)

#library(cluster)



#require(cluster)
#require(chron)
require("factoextra") #visualization with clustering and pca
```

```
## Loading required package: factoextra
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
#require(maps)

library(ggcorrplot)
library(jmv) #for frequency table
```

```
## Warning: package 'jmv' was built under R version 4.0.5
```

```
library(nnet)

library(ggstatsplot)
```

```
## You can cite this package as:
##      Patil, I. (2021). Visualizations with statistical details: The 'ggstatsplot' approach.
##      Journal of Open Source Software, 6(61), 3167, doi:10.21105/joss.03167
```

```
library(Rcpp)
```

```
## Warning: package 'Rcpp' was built under R version 4.0.5
```

```
sessionInfo() #for reporting on package versions
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
```

```
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] Rcpp_1.0.8.3       ggstatsplot_0.10.0 nnet_7.3-18        jmv_2.3.4
##  [5] ggcorrplot_0.1.4   factoextra_1.0.7   RColorBrewer_1.1-2 knitr_1.30
##  [9] MetBrewer_0.2.0    cowplot_1.1.0      forcats_0.5.2      stringr_1.4.0
## [13] dplyr_1.0.10       purrr_0.3.4        readr_2.1.2        tidyr_1.2.0
## [17] tibble_3.1.6       ggplot2_3.4.0      tidyverse_1.3.2    patchwork_1.0.1
## [21] scales_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.2             jsonlite_1.7.2         modelr_0.1.8
##  [4] paletteer_1.5.0        datawizard_0.6.5       assertthat_0.2.1
##  [7] googlesheets4_1.0.1    cellranger_1.1.0       yaml_2.2.1
## [10] bayestestR_0.13.0      ggrepel_0.8.2          pillar_1.8.1
## [13] backports_1.1.10       glue_1.6.2             digest_0.6.27
## [16] rvest_1.0.3            colorspace_1.4-1       htmltools_0.5.0
## [19] pkgconfig_2.0.3        broom_1.0.1            haven_2.5.0
## [22] jmvcore_2.3.19         tzdb_0.3.0             googledrive_2.0.0
## [25] generics_0.0.2         ellipsis_0.3.2         withr_2.5.0
## [28] cli_3.4.1              magrittr_2.0.1         crayon_1.5.2
## [31] readxl_1.3.1           evaluate_0.14          fs_1.5.0
## [34] fansi_0.4.2            xml2_1.3.3             tools_4.0.2
## [37] hms_1.1.2              gargle_1.2.1           lifecycle_1.0.3
## [40] munsell_0.5.0          reprex_2.0.2           statsExpressions_1.3.5
## [43] compiler_4.0.2         rlang_1.0.6            grid_4.0.2
## [46] parameters_0.20.0      rstudioapi_0.13        rmarkdown_2.5.3
## [49] gtable_0.3.0           DBI_1.1.0              rematch2_2.1.2
## [52] correlation_0.8.3      R6_2.5.0               lubridate_1.8.0
## [55] performance_0.10.2     utf8_1.2.1             zeallot_0.1.0
## [58] insight_0.19.0         stringi_1.5.3          vctrs_0.5.1
## [61] dbplyr_2.2.1           tidyselect_1.2.0       xfun_0.22
```

Paths, parameters…

```
my_project<-"221122-1stmanu_stat_trial"

#paths####
#data_main<- "~/shared_vmsdata/"; #update for local laptop

data_in<- paste0("M:/01_Projects/",my_project,"/data_in/")
data_out<-paste0("M:/01_Projects/",my_project,"/data_out/")

#parameters####
#prj<-"+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0" #map projection
#prj<-"+proj=utm +zone=32 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs +type=crs" #map project
#prj2<-sf::st_crs("EPSG:25832")
#proj4string = CRS(prj)

t_today=format(Sys.time(), '%y%m%d')


#latlon to zoom
```

```r
#zolon = c(5, 9)
#zolat = c(53.2, 55)

#latlon to super zoom
#zozolon = c(7, 9)
#zozolat = c(53.5, 54.3)

#my_lon = c(2.5, 9.3)
#my_lat = c(51, 57)
```

Fancy color picking chunk.

```r
#display_all(sequential = FALSE, colorblind_only = TRUE)
#met.brewer("Archambault", 9)

#met.brewer("Veronese", 12) #

serra<-c(met.brewer("Veronese", 12)[1],
         met.brewer("Veronese", 12)[5],
         met.brewer("Veronese", 12)[9],
         met.brewer("Veronese", 12)[12])

#I am assigning cluster ID names to colors! (this was designed in a previous script)
names(serra)<-factor(c(4,3,2,1))

#scales::show_col(serra)


#serra<-c('#332288', '#AA4499', '#44AA99')
#require(scales)
#q_colors =  4 # for no particular reason
#v_colors =  viridis(q_colors, option = "B")
#rev(v_colors)
```

2. Data Loading

Loading part:
I am loading R files from previous step.

I calculated the aggregated values already in project: 220525-repating_HR_calculations.
They are per year per vessel level. And I copied that file to data_in of my new project “221022-clustering_pca”.

It is already merged on ***221124*** and ***221125***.

These data files aggregated individual vessel information and also mean values are used for interannual aggregation of 13 years of the study period (2009-2013). Details of the previous aggregation can be found in:

file:///R:/R/01_Projects/221022-clustering_pca/code/3-1-plot-final_graphs

parameter name explains the aggregation for parameters used in clustering:

perves_f_trait_df <- perves_f_trait_df %>% rename(mean.core_area_index=mean.mean_hr50, sd.core_area_index=sd.mean_hr50, mean.area_flexibility=mean.mean_hrchange_rel, sd.area_flexibility=sd.mean_hrchan

mean.total__fishinghours=mean.sum__fishinghours, sd.total__fishinghours=sd.sum__fishinghours, mean.triplength=mean.mean
sd.triplength=sd.mean__triplength, mean.distance__to__port=mean.dist__to__DHAR, sd.distance__to__port=sd.dist__to__DHAI

```
df_to_explain<-readRDS(file = paste0(data_in,"perves-merged_explanatory-","221124",".rds"))
#f_trait_df-perves-ves_clusters-221125
df_to_cluster<-readRDS(file = paste0(data_in,"f_trait_df-perves-ves_clusters-","221125",".rds"))
```

```
ports<- read.csv(paste(data_in,"UN_LOCODE-230302.csv",sep="")) #UNlist of harbours #(ISO3_Country_Code,
```

3. Data Wrangling

3.cluster

3.explain

```
colnames(df_to_explain)
```

```
##  [1] "VE_REF"             "sub_grp"            "LE_KG_CSH"
##  [4] "pp_tot_kgs"         "eff_hrs"            "LPUE"
##  [7] "other_catch"        "perc_csh"           "Other Gear"
## [10] "TBB_CRU_16-31_0_0" "per_metcra"         "VE_LEN"
## [13] "VE_KW"              "VE_TON"             "fav_dep"
## [16] "fav_lan"            "VE_HOME"
```

```
#all
my_parameters<-c("VE_REF","sub_grp",
                 "LE_KG_CSH","pp_tot_kgs","eff_hrs","LPUE","other_catch","perc_csh",
                 "Other Gear","TBB_CRU_16-31_0_0","per_metcra",
                 "VE_LEN","VE_KW","VE_TON","fav_dep","fav_lan","VE_HOME")

#only continuous without harbors
my_parameters<-c("VE_REF","sub_grp",
                 "LE_KG_CSH","pp_tot_kgs","eff_hrs","LPUE","other_catch","perc_csh",
                 "Other Gear","TBB_CRU_16-31_0_0","per_metcra",
                 "VE_LEN","VE_KW","VE_TON")

df_merged_sub <- df_to_explain %>% dplyr::select(all_of(my_parameters))
```

Check correlation.

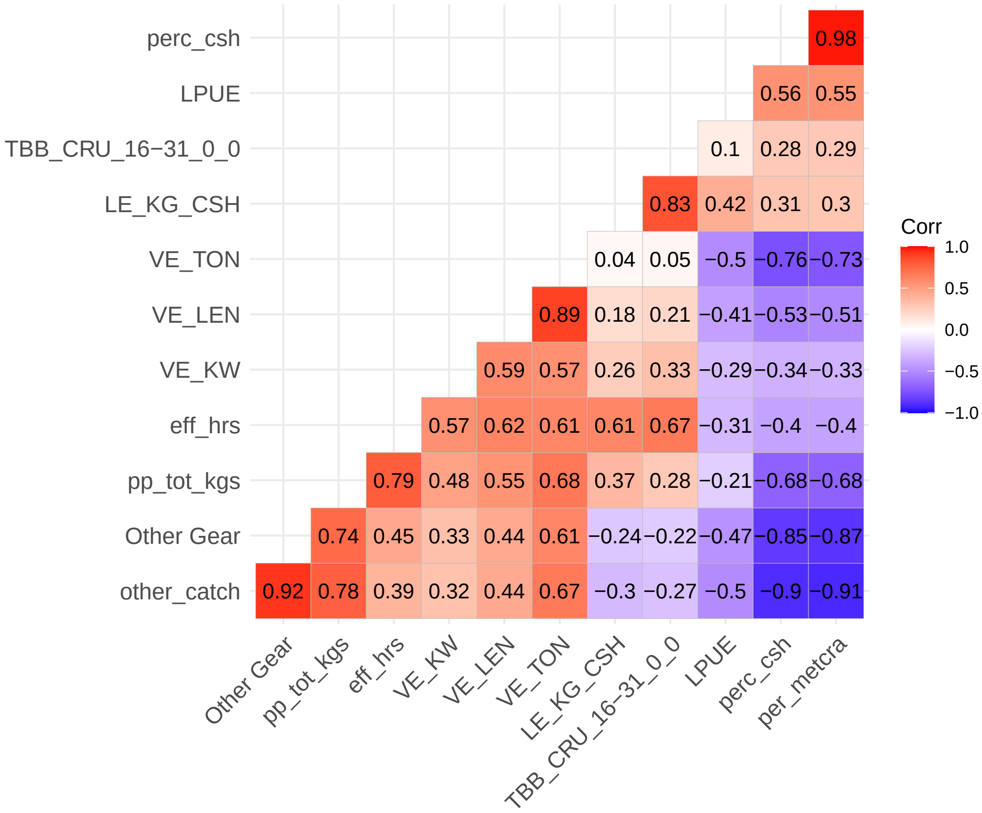
```
perves_f_trait_df<-df_merged_sub

#str(perves_f_trait_df)
summary(perves_f_trait_df)
```

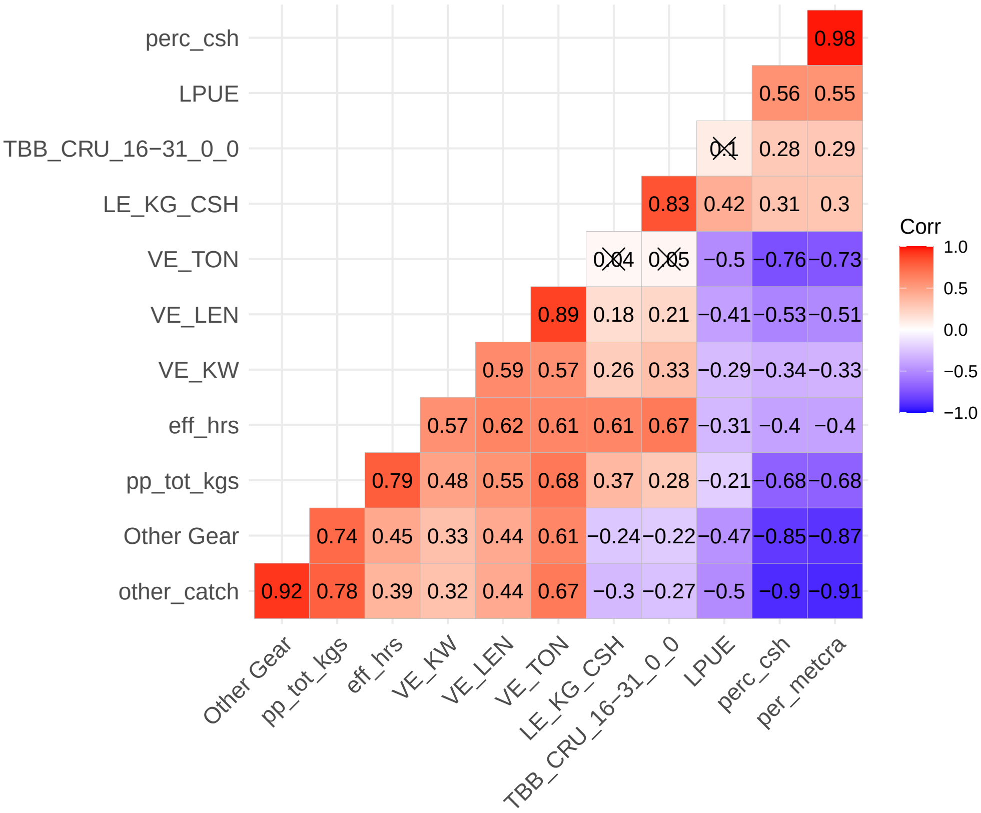
```
##     VE_REF          sub_grp   LE_KG_CSH          pp_tot_kgs
##  Length:211         4: 12   Min.   :   385.2   Min.   :   385.2
##  Class :character   3:119   1st Qu.: 38158.2   1st Qu.: 42566.9
##  Mode  :character   2: 17   Median : 60788.4   Median : 64374.5
##                     1: 63   Mean   : 62183.3   Mean   : 71923.5
##                             3rd Qu.: 82508.8   3rd Qu.: 86809.9
##                             Max.   :151993.0   Max.   :359796.8
##     eff_hrs             LPUE           other_catch          perc_csh
##  Min.   :  48.63   Min.   :  0.9254   Min.   :     0.0   Min.   :0.01918
##  1st Qu.: 878.76   1st Qu.: 37.7449   1st Qu.:     0.0   1st Qu.:0.99864
##  Median :1351.27   Median : 46.7274   Median :     4.2   Median :0.99993
##  Mean   :1401.46   Mean   : 48.3771   Mean   :  9740.3   Mean   :0.95024
##  3rd Qu.:1814.53   3rd Qu.: 57.8626   3rd Qu.:    88.9   3rd Qu.:1.00000
##  Max.   :3685.15   Max.   :111.5714   Max.   :357240.7   Max.   :1.00000
##    Other Gear       TBB_CRU_16-31_0_0   per_metcra          VE_LEN
##  Min.   :    0.00   Min.   :  145.9   Min.   :0.02774   Min.   :12.35
##  1st Qu.:    0.00   1st Qu.: 7783.8   1st Qu.:0.99386   1st Qu.:15.95
##  Median :    7.67   Median :14462.3   Median :0.99940   Median :17.24
##  Mean   : 1233.04   Mean   :14784.6   Mean   :0.94863   Mean   :18.09
##  3rd Qu.:  109.12   3rd Qu.:21952.0   3rd Qu.:1.00000   3rd Qu.:19.36
##  Max.   :44714.63   Max.   :41026.4   Max.   :1.00000   Max.   :25.27
##      VE_KW           VE_TON
##  Min.   :110.0   Min.   : 10.00
##  1st Qu.:184.5   1st Qu.: 28.00
##  Median :216.9   Median : 38.00
##  Mean   :203.3   Mean   : 49.03
##  3rd Qu.:221.0   3rd Qu.: 59.00
##  Max.   :304.8   Max.   :167.25
```

```r
#correlation matrix
my_corr <- cor(perves_f_trait_df[,3:ncol(perves_f_trait_df)],method = "pearson")


ggcorrplot(my_corr, hc.order = TRUE, type = "lower",
   lab = TRUE)
```



```r
# Compute a matrix of correlation p-values
p.mat <- cor_pmat(perves_f_trait_df[,3:ncol(perves_f_trait_df)])
ggcorrplot(my_corr, hc.order = TRUE, type = "lower",
   lab = TRUE, p.mat = p.mat)
```



deciding what parameters to put in cluster

```
#selected
my_parameters<-c("VE_REF","sub_grp",
                 "LE_KG_CSH","LPUE","other_catch",
                 "VE_LEN","VE_KW")

df_merged_sub <- df_to_explain %>% dplyr::select(my_parameters)
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect
## 1.1.0.

## Warning: Please use ‘all_of()‘ or ‘any_of()‘ instead.

## Warning: # Was:

## Warning: data %>% select(my_parameters)

## Warning:

## Warning: # Now:

## Warning: data %>% select(all_of(my_parameters))

## Warning:

## Warning: See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

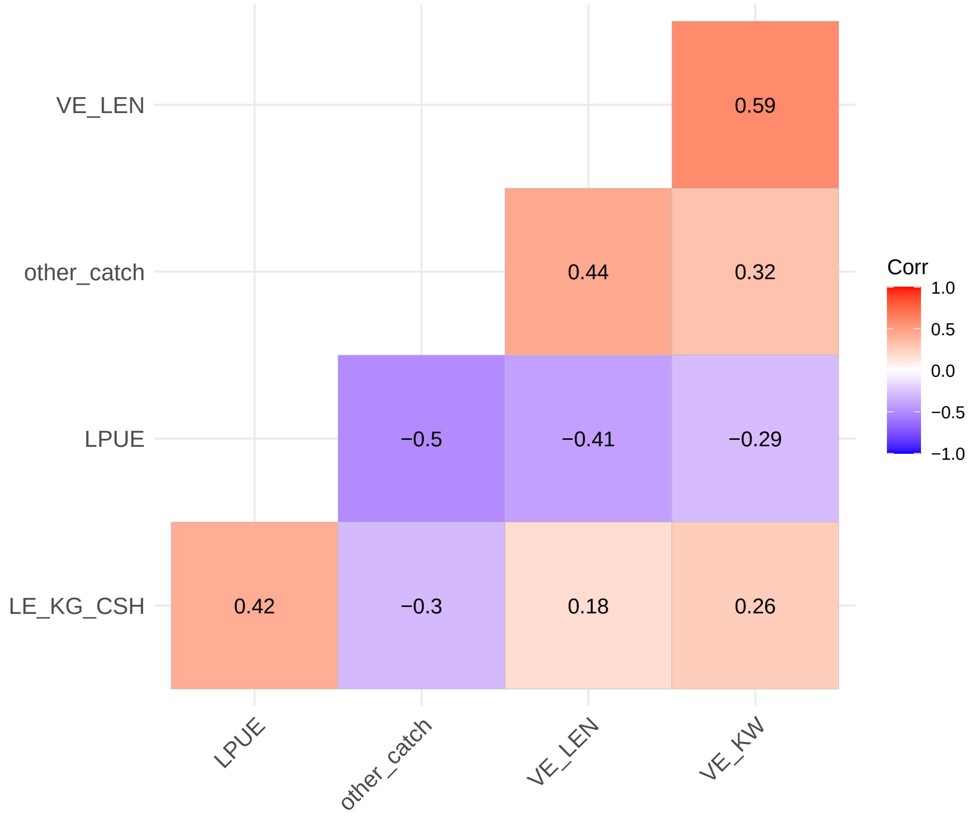
```r
perves_f_trait_df<-df_merged_sub


#str(perves_f_trait_df)
summary(perves_f_trait_df)
```

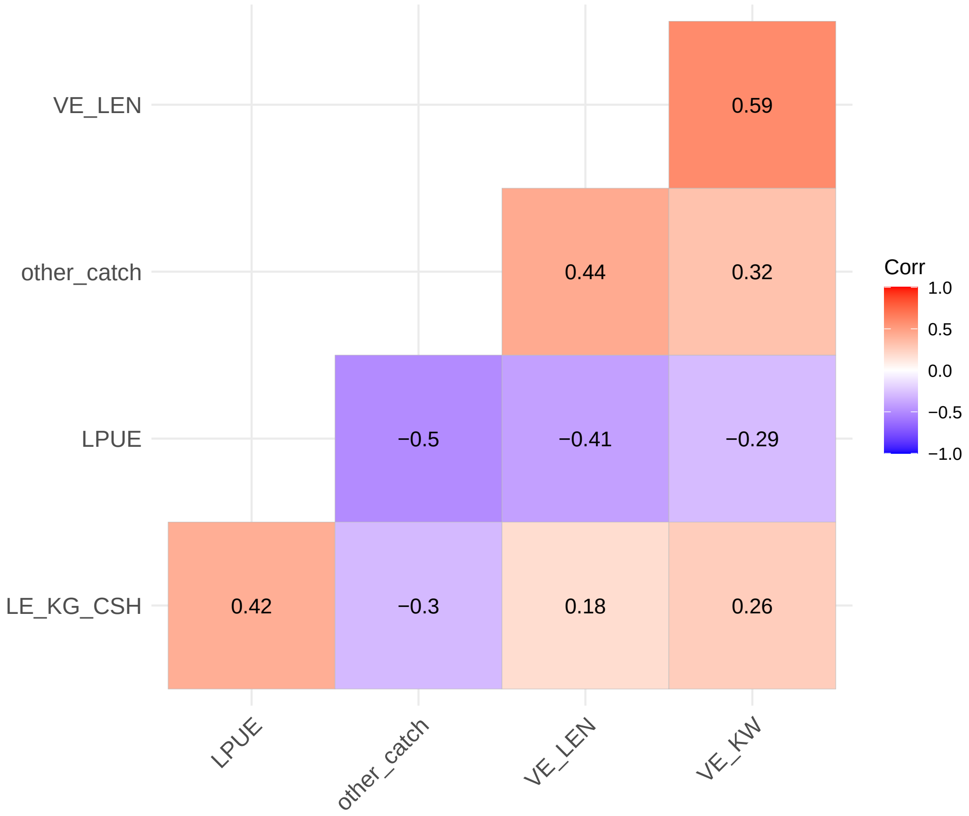
```
##     VE_REF          sub_grp   LE_KG_CSH             LPUE
##  Length:211         4: 12   Min.   :   385.2   Min.   :  0.9254
##  Class :character   3:119   1st Qu.: 38158.2   1st Qu.: 37.7449
##  Mode  :character   2: 17   Median : 60788.4   Median : 46.7274
##                     1: 63   Mean   : 62183.3   Mean   : 48.3771
##                             3rd Qu.: 82508.8   3rd Qu.: 57.8626
##                             Max.   :151993.0   Max.   :111.5714
##   other_catch           VE_LEN          VE_KW
##  Min.   :     0.0   Min.   :12.35   Min.   :110.0
##  1st Qu.:     0.0   1st Qu.:15.95   1st Qu.:184.5
##  Median :     4.2   Median :17.24   Median :216.9
##  Mean   :  9740.3   Mean   :18.09   Mean   :203.3
##  3rd Qu.:    88.9   3rd Qu.:19.36   3rd Qu.:221.0
##  Max.   :357240.7   Max.   :25.27   Max.   :304.8
```

```r
#correlation matrix
my_corr <- cor(perves_f_trait_df[,3:ncol(perves_f_trait_df)],method = "pearson")


ggcorrplot(my_corr, hc.order = TRUE, type = "lower",
   lab = TRUE)
```



```r
# Compute a matrix of correlation p-values
p.mat <- cor_pmat(perves_f_trait_df[,3:ncol(perves_f_trait_df)])
ggcorrplot(my_corr, hc.order = TRUE, type = "lower",
   lab = TRUE, p.mat = p.mat)
```



I am omitting harbor for the first try choosing only:LE_KG_CSH,LPUE, other_catch, VE_LEN, VE_KW

It would be nice to have vessel built year. but I do not have it at the moment.

```
df_stat<-df_to_explain %>% dplyr::select(c(sub_grp,LE_KG_CSH,LPUE, other_catch, VE_LEN, VE_KW))

#order levels to plot  1234
df_stat$sub_grp <- factor(df_stat$sub_grp, levels = 1:4)
```

##3.harbors harbor aggragation is recorded. number of vessels per cluster are calculated.

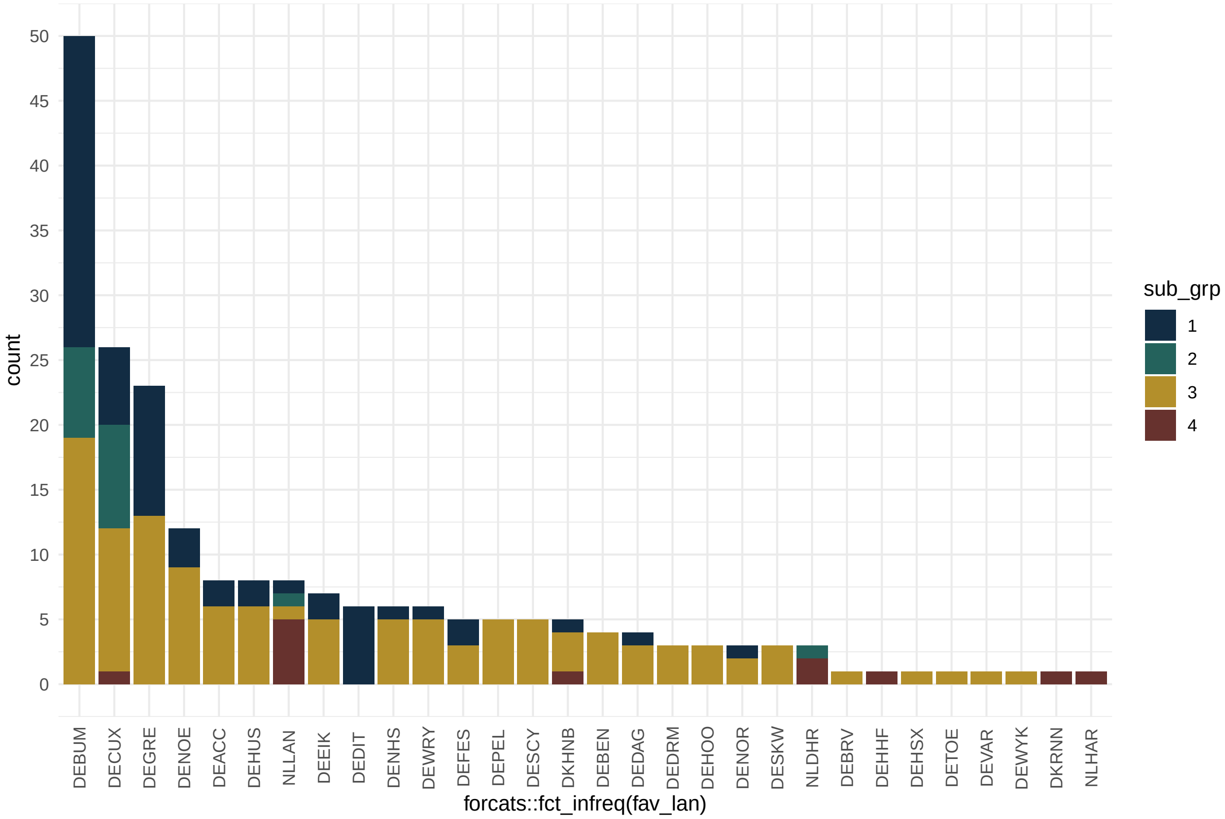
```
df_merged<-df_to_explain
df_merged$fav_lan<-as.factor(df_merged$fav_lan)
levels(df_merged$fav_lan)
```

```
##  [1] "DEACC" "DEBEN" "DEBRV" "DEBUM" "DECUX" "DEDAG" "DEDIT" "DEDRM" "DEEIK"
## [10] "DEFES" "DEGRE" "DEHHF" "DEHOO" "DEHSX" "DEHUS" "DENHS" "DENOE" "DENOR"
## [19] "DEPEL" "DESCY" "DESKW" "DETOE" "DEVAR" "DEWRY" "DEWYK" "DKHNB" "DKRNN"
## [28] "NLDHR" "NLHAR" "NLLAN"
```

```
#str(df_merged)

#order levels to plot  1234
df_merged$sub_grp <- factor(df_merged$sub_grp, levels = 1:4)

ggplot()+
  geom_bar(data=df_merged,aes(x=forcats::fct_infreq(fav_lan),fill=sub_grp))+
  scale_fill_manual(values = serra) +
  scale_y_continuous(breaks = seq(0, 50, by = 5))+
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



```
#get an output table for number of vessels per port
temp_df<-df_merged %>%
  group_by(sub_grp,fav_lan) %>%
  summarise(n_vessels = n())
```

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

```
temp_df <- temp_df %>% left_join(ports,by=c("fav_lan"="FAO_code"))

#it is good enough to print as such with our wrapping the column names
knitr::kable(colnames(temp_df))
```

x
sub_grp
fav_lan
n_vessels

x
X
ISO3_Country_Code
full_name
Coordinates
Latitude
Longitude
EU_Fish_Port
Port
Region

```
knitr::kable(temp_df)
```

sub_grp	fav_lan	n_vessels	X	ISO3_Country_Code	full_name	Coordinates	Latitude	Longitude	EU_Fish_Port	Port	Region
1	DEACC	2	1283	DEU	Accumersiel	5368N 00849E	53.679877	7.485630	Y	Y	Niedersachsen
1	DEBUM	24	1359	DEU	Buesum	5413N 00885E	54.133333	8.850000	Y	Y	Schleswig-Holstein-Nordsee
1	DECUX	6	1371	DEU	Cuxhaven	5387N 00870E	53.866667	8.700000	Y	Y	Niedersachsen
1	DEDAG	1	1372	DEU	Dagebuell	5473N 00869E	54.731667	8.691667	Y	Y	Schleswig-Holstein-Nordsee
1	DEDIT	6	1381	DEU	Ditzum	5331N 00728E	53.308333	7.275000	Y	Y	Niedersachsen
1	DEEIK	2	1402	DEU	Eidersperrwerk	5427N 00885E	54.266667	8.850000	Y	Y	Schleswig-Holstein-Nordsee
1	DEFES	2	1411	DEU	Fedderwardersiel	5360N 00836E	53.600000	8.358333	Y	Y	Niedersachsen
1	DEGRE	10	1439	DEU	Greetsiel	5350N 00710E	53.500000	7.100000	Y	Y	Niedersachsen
1	DEHUS	2	1485	DEU	Husum	5448N 00904E	54.475000	9.045000	Y	Y	Schleswig-Holstein-Nordsee
1	DENHS	1	1595	DEU	Neuharlingersiel	5370N 00770E	53.700000	7.703333	Y	Y	Niedersachsen
1	DENOE	3	1600	DEU	Norddeich	5362N 00716E	53.625000	7.158333	Y	Y	Niedersachsen
1	DENOR	1	1601	DEU	Nordstrand	5450N 00881E	54.498333	8.808333	Y	Y	Schleswig-Holstein-Nordsee
1	DEWRY	1	1745	DEU	Wremen	5365N 00850E	53.650000	8.500000	Y	Y	Niedersachsen
1	DKHNB	1	1947	DNK	Havneby	5509N 00855E	55.087300	8.554500	Y	Y	Daenemark-Nordsee
1	NLLAN	1	7550	NLD	Lauwersoog	5342N 00620E	53.416667	6.200000	Y	Y	Niederlande
2	DEBUM	7	1359	DEU	Buesum	5413N 00885E	54.133333	8.850000	Y	Y	Schleswig-Holstein-Nordsee

sub_grp	fav_lan	n_vessels	X	ISO3_Country_Code	full_name	Coordinates	Latitude	Longitude	EU_Fish_Port	Port	Region
2	DECUX	8	1371	DEU	Cuxhaven	5387N 00870E	53.86667	8.700000	Y	Y	Niedersachsen
2	NLDHR	1	7522	NLD	Den Helder	5297N 00477E	52.96667	4.766667	Y	Y	Niederlande
2	NLLAN	1	7550	NLD	Lauwersoog	5342N 00620E	53.41667	6.200000	Y	Y	Niederlande
3	DEACC	6	1283	DEU	Accumersiel	5368N 00849E	53.67987	7.485630	Y	Y	Niedersachsen
3	DEBEN	4	1313	DEU	Bensersiel	5368N 00758E	53.67500	7.575000	Y	Y	Niedersachsen
3	DEBRV	1	1354	DEU	Bremerhaven	5354N 00857E	53.53667	8.573333	Y	Y	Niedersachsen
3	DEBUM	19	1359	DEU	Buesum	5413N 00885E	54.13333	8.850000	Y	Y	Schleswig-Holstein-Nordsee
3	DECUX	11	1371	DEU	Cuxhaven	5387N 00870E	53.86667	8.700000	Y	Y	Niedersachsen
3	DEDAG	3	1372	DEU	Dagebuell	5473N 00869E	54.73167	8.691667	Y	Y	Schleswig-Holstein-Nordsee
3	DEDRM	3	1388	DEU	Dorum	5374N 00853E	53.73833	8.533333	Y	Y	Niedersachsen
3	DEEIK	5	1402	DEU	Eidersperrwerk	5427N 00885E	54.26667	8.850000	Y	Y	Schleswig-Holstein-Nordsee
3	DEFES	3	1411	DEU	Fedderwardersiel	5360N 00836E	53.60000	8.358333	Y	Y	Niedersachsen
3	DEGRE	13	1439	DEU	Greetsiel	5350N 00710E	53.50000	7.100000	Y	Y	Niedersachsen
3	DEHOO	3	1474	DEU	Hooksiel	5364N 00808E	53.64167	8.083333	Y	Y	Niedersachsen
3	DEHSX	1	1483	DEU	Harlesiel	5371N 00781E	53.70667	7.808333	Y	Y	Niedersachsen
3	DEHUS	6	1485	DEU	Husum	5448N 00904E	54.47500	9.045000	Y	Y	Schleswig-Holstein-Nordsee
3	DENHS	5	1595	DEU	Neuharlingersiel	5370N 00770E	53.70000	7.703333	Y	Y	Niedersachsen
3	DENOE	9	1600	DEU	Norddeich	5362N 00716E	53.62500	7.158333	Y	Y	Niedersachsen
3	DENOR	2	1601	DEU	Nordstrand	5450N 00881E	54.49833	8.808333	Y	Y	Schleswig-Holstein-Nordsee
3	DEPEL	5	1625	DEU	Pellworm	5452N 00869E	54.52167	8.688333	Y	Y	Schleswig-Holstein-Nordsee
3	DESCY	5	1664	DEU	Schluettsiel	5468N 00875E	54.68167	8.753333	Y	Y	Schleswig-Holstein-Nordsee
3	DESKW	3	1674	DEU	Spiekaneufeld	5379N 00855E	53.79000	8.548333	Y	Y	Niedersachsen

sub_grp	fav_lan	n_vessels	X	ISO3_Country_Code	full_name	Coordinates	Latitude	Longitude	EU_Fish_Port	Port	Region
3	DETOE	1	1698	DEU	Toenning	5432N 00895E	54.31667	8.950000	Y	Y	Schleswig-Holstein-Nordsee
3	DEVAR	1	1714	DEU	Varel	5341N 00818E	53.41000	8.183333	Y	Y	Niedersachsen
3	DEWRY	5	1745	DEU	Wremen	5365N 00850E	53.65000	8.500000	Y	Y	Niedersachsen
3	DEWYK	1	1755	DEU	Wyk auf Foehr	5469N 00858E	54.69333	8.575000	Y	Y	Schleswig-Holstein-Nordsee
3	DKHNB	3	1947	DNK	Havneby	5509N 00855E	55.08730	8.554500	Y	Y	Daenemark-Nordsee
3	NLLAN	1	7550	NLD	Lauwersoog	5342N 00620E	53.41667	6.200000	Y	Y	Niederlande
4	DECUX	1	1371	DEU	Cuxhaven	5387N 00870E	53.86667	8.700000	Y	Y	Niedersachsen
4	DEHHF	1	1464	DEU	Heiligenhafen	5437N 01098E	54.37333	10.983333	Y	Y	Schleswig-Holstein-Ostsee
4	DKHNB	1	1947	DNK	Havneby	5509N 00855E	55.08730	8.554500	Y	Y	Daenemark-Nordsee
4	DKRNN	1	2147	DNK	Ronne	5510N 01470E	55.09931	14.697600	Y	Y	Daenemark-Ostsee
4	NLDHR	2	7522	NLD	Den Helder	5297N 00477E	52.96667	4.766667	Y	Y	Niederlande
4	NLHAR	1	7536	NLD	Harlingen	5318N 00542E	53.18333	5.416667	Y	Y	Niederlande
4	NLLAN	5	7550	NLD	Lauwersoog	5342N 00620E	53.41667	6.200000	Y	Y	Niederlande

```
write_csv(temp_df,file = paste0(data_out,"musselabm-","harbor_vessel",".csv"))
```

4. Plots

detailed example

4.cluster

is data normally dist?

```
df_to_cluster
```

```
## # A tibble: 211 x 7
##    VE_REF       core_area_index area_flexibility total~1 tripl~2 dista~3 sub_grp
##    <chr>                  <dbl>            <dbl>   <dbl>   <dbl>   <dbl> <fct>
##  1 BEL011101959            919.             3.18   1252.   26.0   82443. 1
##  2 DEU000070400            515.             3.41    690.   40.6  102810. 2
##  3 DEU000070500            164.             3.06    716.    9.61  23696. 3
##  4 DEU000160300            109.             2.99    587.    6.60  19190. 3
##  5 DEU000200500            338.             3.45    912.   28.9  166331. 4
##  6 DEU000210300            108.             2.92    331.    6.87  20154. 3
##  7 DEU000270300            878.             3.31   1607.   41.8  110384. 2
##  8 DEU000290300           2644.             3.53   3515.   69.0  153246. 2
##  9 DEU000340500           1388.             3.30    937.   40.0  112494. 2
## 10 DEU000350300            165.             2.98    730.    8.65  29170. 3
## # ... with 201 more rows, and abbreviated variable names 1: total_fishinghours,
## #   2: triplength, 3: distance_to_port
```

```
shapiro.test(df_to_cluster$core_area_index)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  df_to_cluster$core_area_index
## W = 0.55363, p-value < 2.2e-16
```

```
shapiro.test(df_to_cluster$area_flexibility)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  df_to_cluster$area_flexibility
## W = 0.97738, p-value = 0.001784
```

```
shapiro.test(df_to_cluster$total_fishinghours)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  df_to_cluster$total_fishinghours
## W = 0.97756, p-value = 0.001892
```

```
shapiro.test(df_to_cluster$triplength)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  df_to_cluster$triplength
## W = 0.92417, p-value = 5.836e-09
```

```
shapiro.test(df_to_cluster$distance_to_port)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  df_to_cluster$distance_to_port
## W = 0.72211, p-value < 2.2e-16
```

Choose NON-PARAMETRIC TEST.

```
summary(df_to_cluster)
```

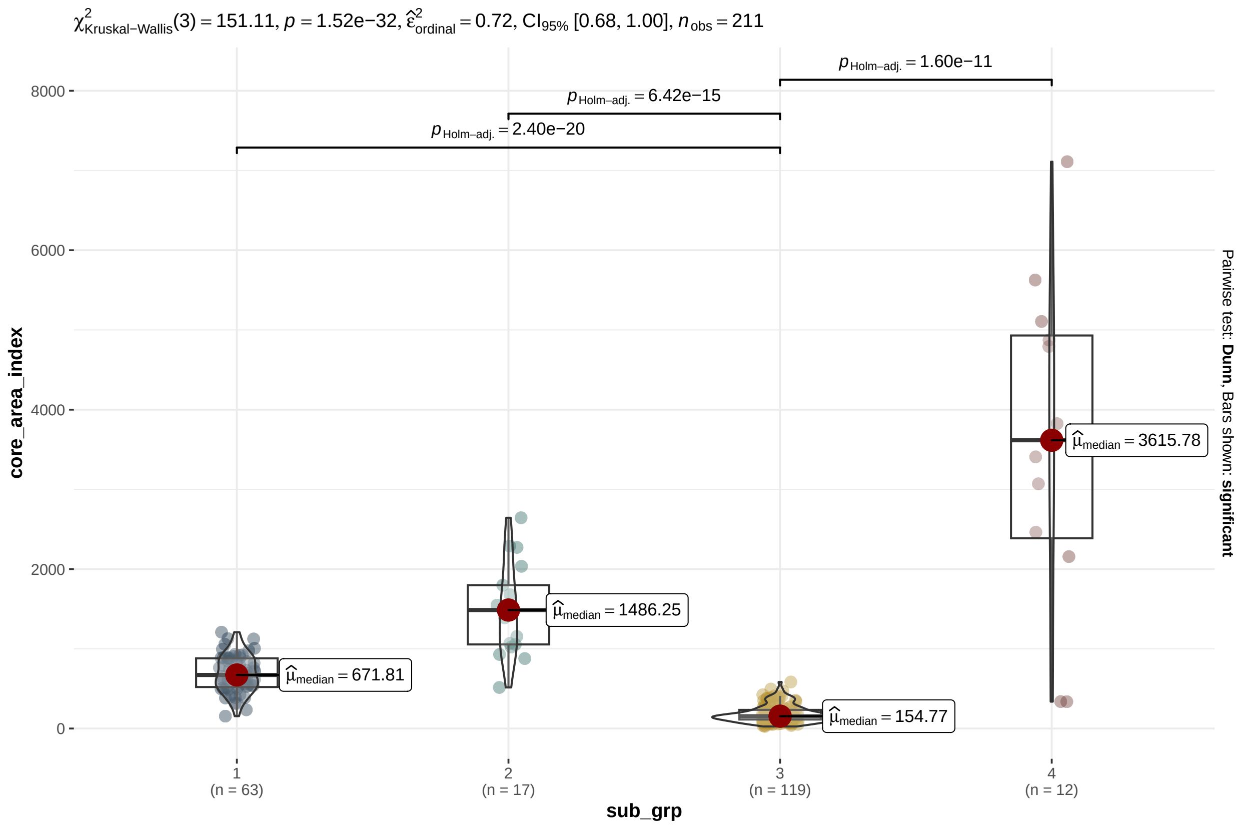
```
##     VE_REF          core_area_index   area_flexibility total_fishinghours
##  Length:211         Min.   :  25.77   Min.   :2.759    Min.   :  48.63   
##  Class :character   1st Qu.: 149.98   1st Qu.:3.143    1st Qu.: 878.76   
##  Mode  :character   Median : 315.38   Median :3.281    Median :1351.27   
##                     Mean   : 632.43   Mean   :3.283    Mean   :1401.46   
##                     3rd Qu.: 727.29   3rd Qu.:3.396    3rd Qu.:1814.53   
##                     Max.   :7110.31   Max.   :4.032    Max.   :3685.15   
##    triplength     distance_to_port sub_grp
##  Min.   : 3.956   Min.   : 11809   1: 63  
##  1st Qu.:14.466   1st Qu.: 35448   2: 17  
##  Median :24.207   Median : 46652   3:119  
##  Mean   :26.352   Mean   : 67452   4: 12  
##  3rd Qu.:35.462   3rd Qu.: 78863          
##  Max.   :82.349   Max.   :305945
```

```
#df_to_cluster %>%
#  group_by(sub_grp) %>%
#  summarise(across(2:6,
#                   list(median = median, mean = mean, sd = sd, min=min,max=max),
#                   .names = "{.col}-{.fn}"))


#
ggbetweenstats(
  data = df_to_cluster,
  x = sub_grp,
  y = core_area_index,
  type = "nonparametric"
)+
  scale_colour_manual(
  values = serra,
  )
```

```
## Warning in (function (mapping = NULL, data = NULL, stat = "identity", position =
## "identity", : Ignoring unknown parameters: `segment.linetype`
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```



```
temp_df <- df_to_cluster %>%
  group_by(sub_grp) %>%
  summarise_at(c("core_area_index"), list(median = median, mean = mean, sd = sd, min=min,max=max))
knitr::kable(temp_df)
```

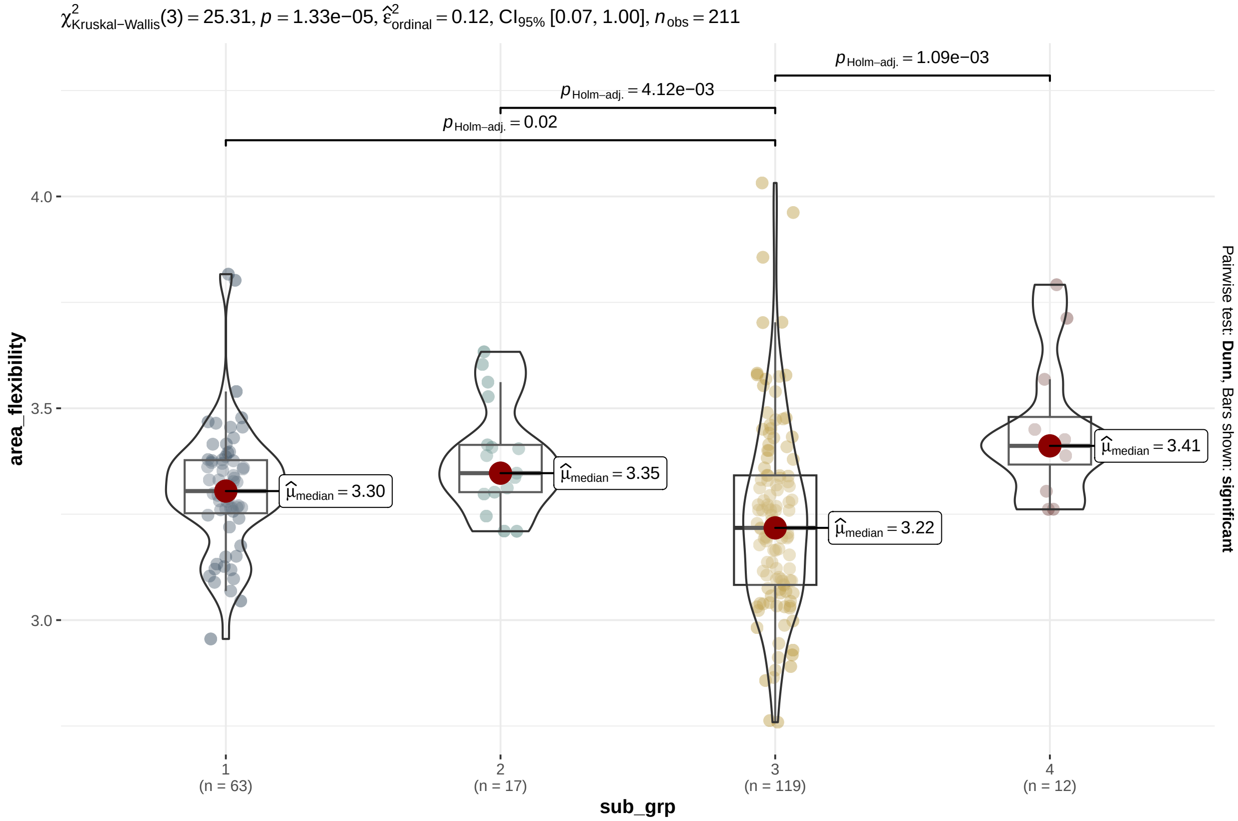
sub_grp	median	mean	sd	min	max
1	671.8138	683.3898	230.6545	154.82579	1208.2514
2	1486.2515	1486.9644	578.8590	515.18585	2643.7759
3	154.7685	184.9535	109.5209	25.77415	583.7084
4	3615.7793	3591.7230	2061.9764	336.71010	7110.3125

```
write_csv(temp_df,file = paste0(data_out,"musselabm-","core_area_index",".csv"))

#
ggbetweenstats(
  data = df_to_cluster,
  x = sub_grp,
  y = area_flexibility,
  type = "nonparametric"
)+
  scale_colour_manual(
  values = serra,
  )
```

```
## Warning in (function (mapping = NULL, data = NULL, stat = "identity", position =
## "identity", : Ignoring unknown parameters: ‘segment.linetype‘

## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```



```
temp_df <- df_to_cluster %>%
  group_by(sub_grp) %>%
  summarise_at(c("area_flexibility"), list(median = median, mean = mean, sd = sd, min=min,max=max))
knitr::kable(temp_df)
```

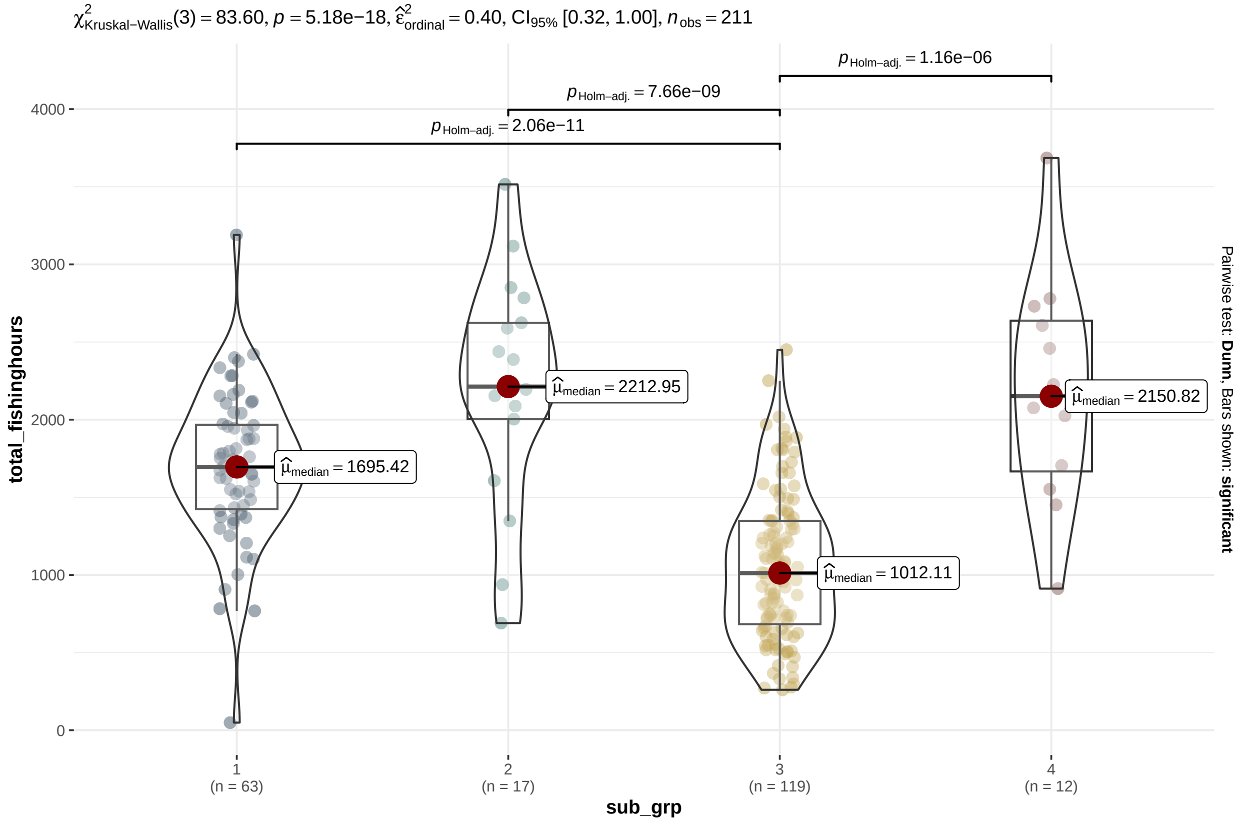
sub_grp	median	mean	sd	min	max
1	3.304568	3.308126	0.1502837	2.955536	3.816552
2	3.346852	3.385128	0.1295437	3.209747	3.633262
3	3.217691	3.238134	0.2246333	2.758949	4.031913
4	3.411284	3.449094	0.1655735	3.261499	3.791583

```
write_csv(temp_df,file = paste0(data_out,"musselabm-","area_flexibility",".csv"))

#
ggbetweenstats(
  data = df_to_cluster,
  x = sub_grp,
  y = total_fishinghours,
  type = "nonparametric"
)+
  scale_colour_manual(
  values = serra,
  )
```

```
## Warning in (function (mapping = NULL, data = NULL, stat = "identity", position =
## "identity", : Ignoring unknown parameters: ‘segment.linetype‘

## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```



```
temp_df <- df_to_cluster %>%
  group_by(sub_grp) %>%
  summarise_at(c("total_fishinghours"), list(median = median, mean = mean, sd = sd, min=min,max=max))
knitr::kable(temp_df)
```

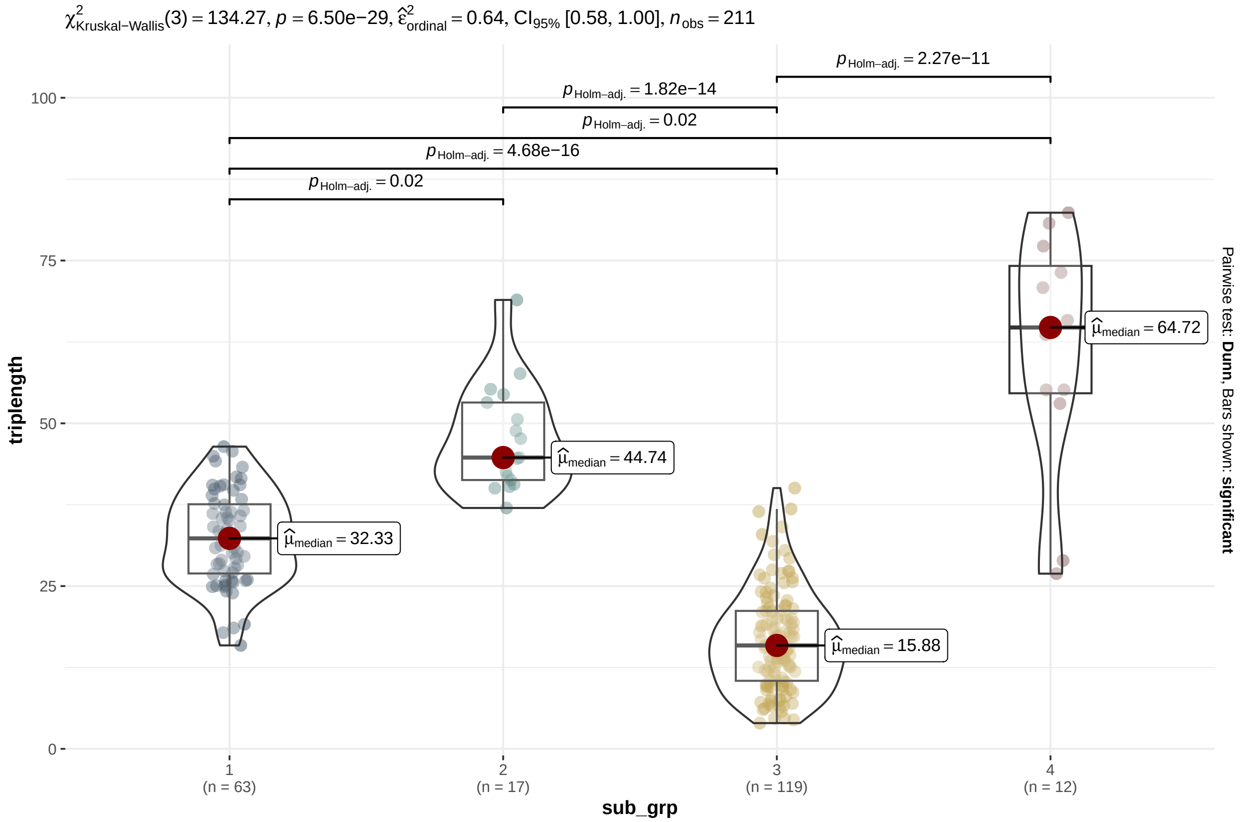
sub_grp	median	mean	sd	min	max
1	1695.418	1695.687	482.9011	48.63333	3189.592
2	2212.949	2208.319	740.1988	689.68974	3515.372
3	1012.106	1051.525	471.8420	260.28333	2450.369
4	2150.825	2183.944	737.6611	911.71000	3685.154

```
write_csv(temp_df,file = paste0(data_out,"musselabm-","total_fishinghours",".csv"))

#
ggbetweenstats(
  data = df_to_cluster,
  x = sub_grp,
  y = triplength,
  type = "nonparametric"
)+
  scale_colour_manual(
  values = serra,
  )
```

```
## Warning in (function (mapping = NULL, data = NULL, stat = "identity", position =
## "identity", : Ignoring unknown parameters: ‘segment.linetype‘

## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```



```
temp_df <- df_to_cluster %>%
  group_by(sub_grp) %>%
  summarise_at(c("triplength"), list(median = median, mean = mean, sd = sd, min=min,max=max))
knitr::kable(temp_df)
```

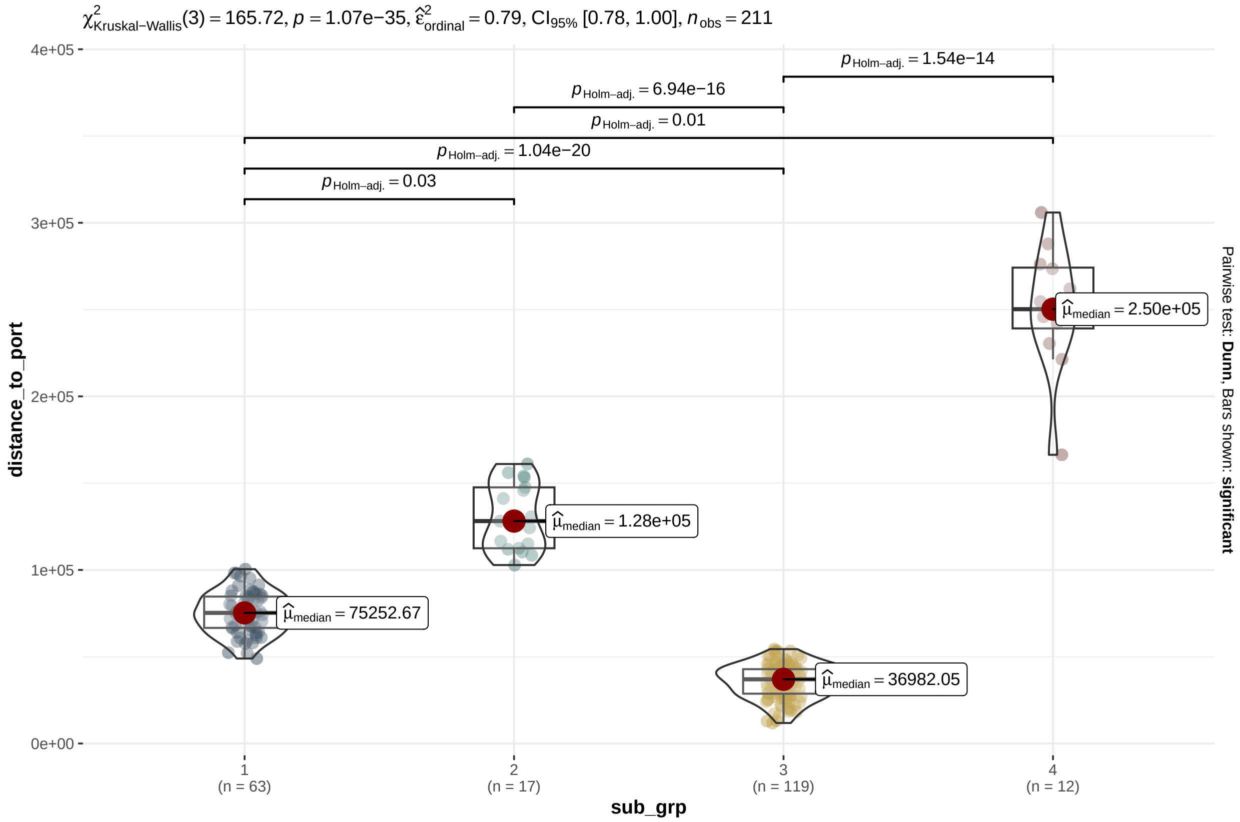
sub_grp	median	mean	sd	min	max
1	32.33045	32.36189	7.186190	15.892593	46.42073
2	44.73920	47.61776	8.259382	36.997584	68.95257
3	15.88431	16.63042	7.771796	3.955714	40.05543
4	64.72111	61.07376	18.352791	26.908756	82.34921

```
write_csv(temp_df,file = paste0(data_out,"musselabm-","triplength",".csv"))

#
ggbetweenstats(
  data = df_to_cluster,
  x = sub_grp,
  y = distance_to_port,
  type = "nonparametric"
)+
  scale_colour_manual(
  values = serra,
  )
```

```
## Warning in (function (mapping = NULL, data = NULL, stat = "identity", position =
## "identity", : Ignoring unknown parameters: ‘segment.linetype‘

## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```



```
temp_df <- df_to_cluster %>%
  group_by(sub_grp) %>%
  summarise_at(c("distance_to_port"), list(median = median, mean = mean, sd = sd, min=min,max=max))
knitr::kable(temp_df)
```

sub_grp	median	mean	sd	min	max
1	75252.67	75177.00	12195.914	48947.98	100426.79
2	128148.22	130562.80	19536.906	102810.47	161044.84
3	36982.05	35852.34	9810.115	11808.97	54369.99
4	250235.58	250854.21	36025.651	166330.75	305944.86

```
write_csv(temp_df,file = paste0(data_out,"musselabm-","distance_to_port",".csv"))
```

4.explanatory

is data normally dist?

```
summary(df_stat)
```

```
##  sub_grp   LE_KG_CSH            LPUE           other_catch
##  1: 63   Min.   :   385.2   Min.   :  0.9254   Min.   :     0.0
##  2: 17   1st Qu.: 38158.2   1st Qu.: 37.7449   1st Qu.:     0.0
##  3:119   Median : 60788.4   Median : 46.7274   Median :     4.2
##  4: 12   Mean   : 62183.3   Mean   : 48.3771   Mean   :  9740.3
##          3rd Qu.: 82508.8   3rd Qu.: 57.8626   3rd Qu.:    88.9
##          Max.   :151993.0   Max.   :111.5714   Max.   :357240.7
##      VE_LEN          VE_KW
##  Min.   :12.35   Min.   :110.0
##  1st Qu.:15.95   1st Qu.:184.5
##  Median :17.24   Median :216.9
##  Mean   :18.09   Mean   :203.3
##  3rd Qu.:19.36   3rd Qu.:221.0
##  Max.   :25.27   Max.   :304.8
```

```
shapiro.test(df_stat$LE_KG_CSH)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  df_stat$LE_KG_CSH
## W = 0.98396, p-value = 0.01697
```

```
shapiro.test(df_stat$LPUE)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  df_stat$LPUE
## W = 0.9701, p-value = 0.000187
```

```
shapiro.test(df_stat$other_catch)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_stat$other_catch
## W = 0.21393, p-value < 2.2e-16
```

```
shapiro.test(df_stat$VE_LEN)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_stat$VE_LEN
## W = 0.89524, p-value = 5.594e-11
```

```
shapiro.test(df_stat$VE_KW)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_stat$VE_KW
## W = 0.81532, p-value = 4.312e-15
```

Choose NON-PARAMETRIC TEST.

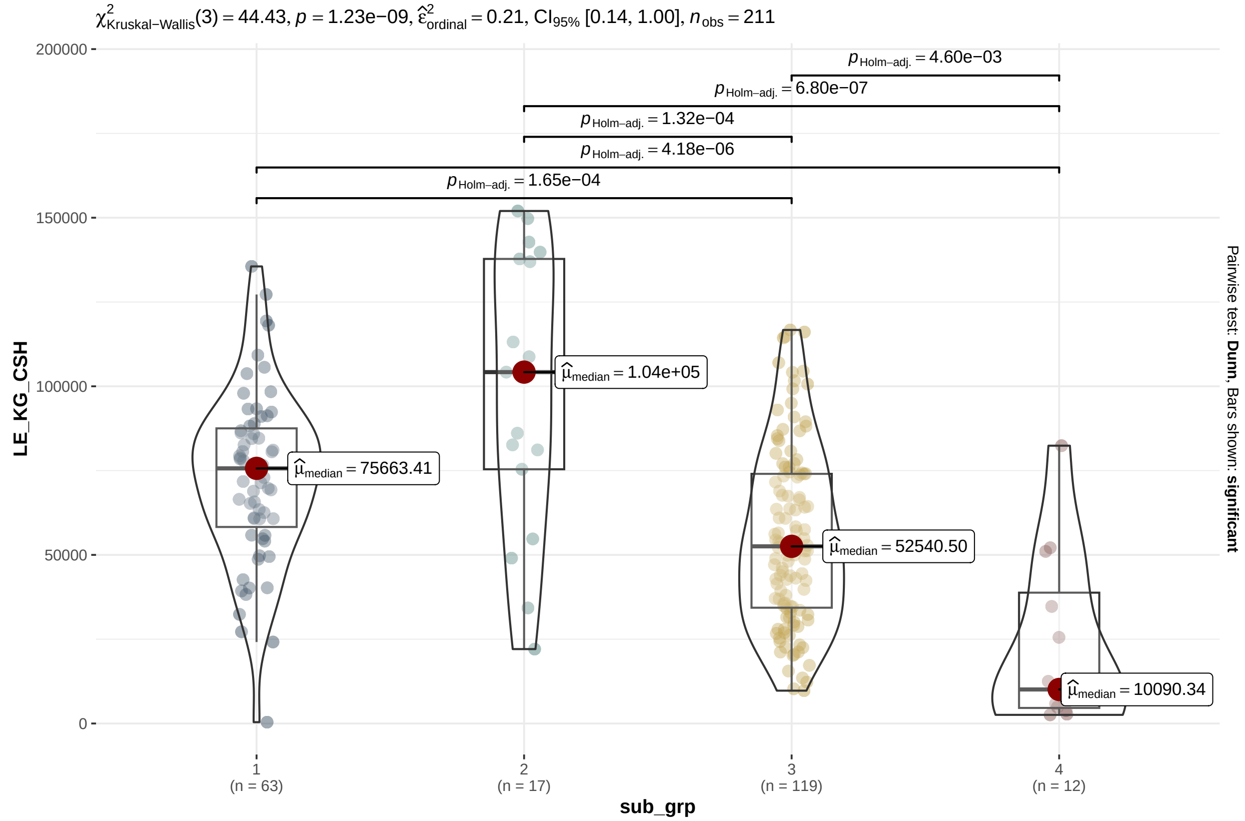
```
colnames(df_stat)
```

```
## [1] "sub_grp"     "LE_KG_CSH"   "LPUE"        "other_catch" "VE_LEN"
## [6] "VE_KW"
```

```
ggbetweenstats(
  data = df_stat,
  x = sub_grp,
  y = LE_KG_CSH,
  type = "nonparametric"
)+
  scale_colour_manual(
  values = serra,
  )
```

```
## Warning in (function (mapping = NULL, data = NULL, stat = "identity", position =
## "identity", : Ignoring unknown parameters: ‘segment.linetype‘
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```



```
temp_df <- df_stat %>%
  group_by(sub_grp) %>%
  summarise_at(c("LE_KG_CSH"), list(median = median, mean = mean, sd = sd, min=min,max=max))
knitr::kable(temp_df)
```

sub_grp	median	mean	sd	min	max
1	75663.41	72918.38	25395.82	385.2318	135555.4
2	104207.80	98262.09	41902.33	22066.5758	151993.0
3	52540.50	55214.27	26379.00	9734.0182	116709.8
4	10090.34	23821.43	26042.62	2556.0547	82392.4

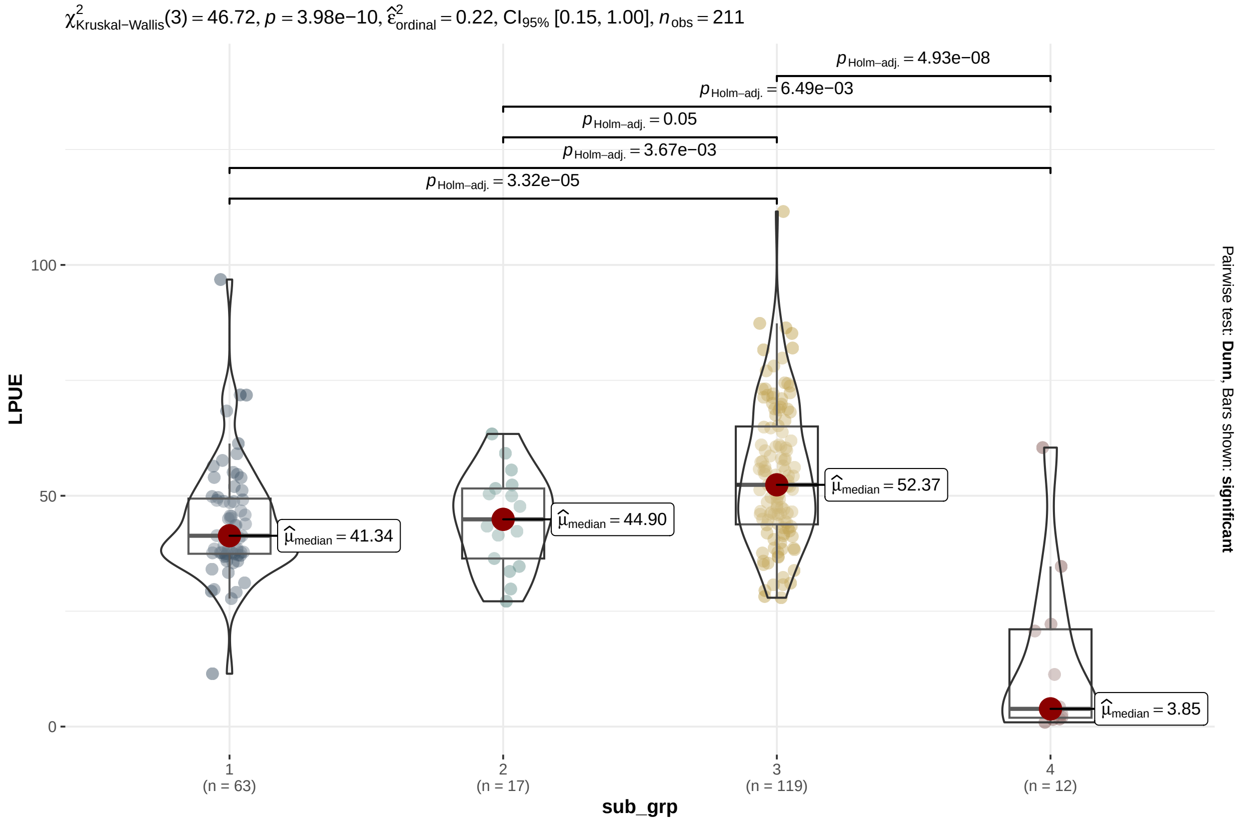
```
write_csv(temp_df,file = paste0(data_out,"musselabm-","LE_KG_CSH",".csv"))



ggbetweenstats(
  data = df_stat,
  x = sub_grp,
  y = LPUE,
  type = "nonparametric"
)+
  scale_colour_manual(
  values = serra,
  )
```

```
## Warning in (function (mapping = NULL, data = NULL, stat = "identity", position =
## "identity", : Ignoring unknown parameters: ‘segment.linetype‘

## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```



```
temp_df <- df_stat %>%
  group_by(sub_grp) %>%
  summarise_at(c("LPUE"), list(median = median, mean = mean, sd = sd, min=min,max=max))
knitr::kable(temp_df)
```

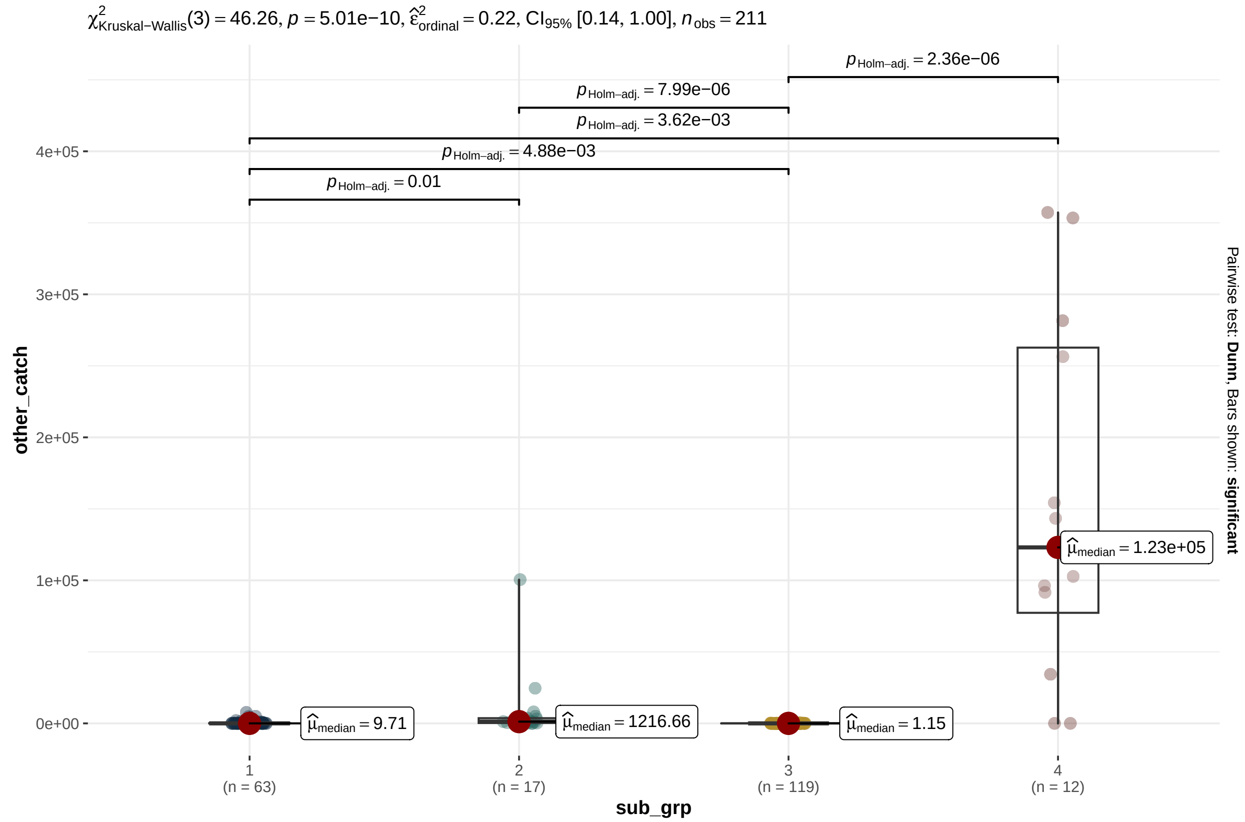
sub_grp	median	mean	sd	min	max
1	41.335974	44.29354	12.59282	11.4565721	96.83555
2	44.896110	44.93154	10.28313	27.1406127	63.39054
3	52.369771	54.51565	15.09858	27.9123237	111.57142
4	3.845739	13.82290	18.23795	0.9253936	60.43421

```
write_csv(temp_df,file = paste0(data_out,"musselabm-","LPUE",".csv"))
```

```r
ggbetweenstats(
  data = df_stat,
  x = sub_grp,
  y = other_catch,
  type = "nonparametric"
)+
  scale_colour_manual(
  values = serra,
  )
```

```
## Warning in (function (mapping = NULL, data = NULL, stat = "identity", position =
## "identity", : Ignoring unknown parameters: ‘segment.linetype‘
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```



```r
temp_df <- df_stat %>%
  group_by(sub_grp) %>%
  summarise_at(c("other_catch"), list(median = median, mean = mean, sd = sd, min=min,max=max))
knitr::kable(temp_df)
```

sub_grp	median	mean	sd	min	max
1	9.714286e+00	441.01037	1301.7533	0	7685.1947
2	1.216664e+03	8958.99278	24327.3201	0	100528.7291
3	1.153846e+00	32.69259	92.1262	0	645.9944
4	1.230500e+05	155935.26802	127749.5169	0	357240.7333

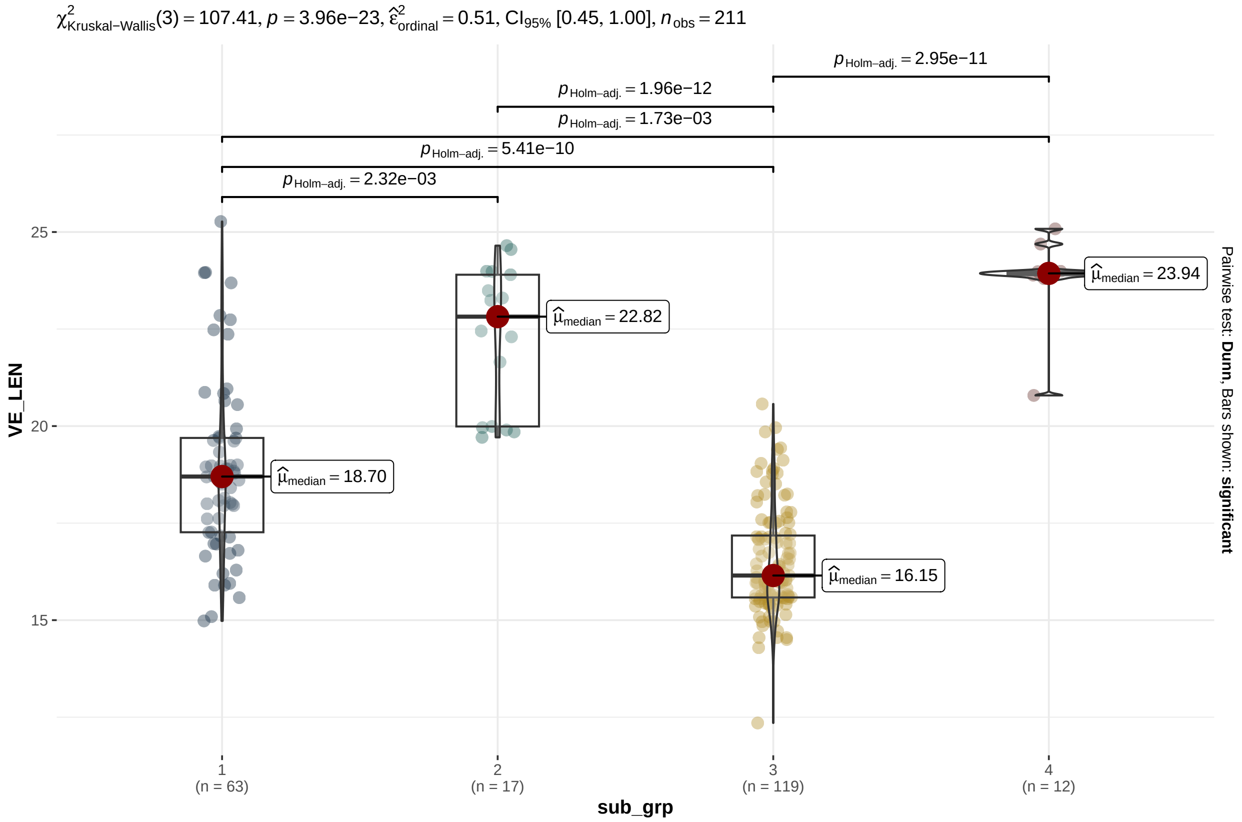
```
write_csv(temp_df,file = paste0(data_out,"musselabm-","other_catch",".csv"))



ggbetweenstats(
  data = df_stat,
  x = sub_grp,
  y = VE_LEN,
  type = "nonparametric"
)+
  scale_colour_manual(
  values = serra,
  )
```

```
## Warning in (function (mapping = NULL, data = NULL, stat = "identity", position =
## "identity", : Ignoring unknown parameters: `segment.linetype`
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```



```
temp_df <- df_stat %>%
  group_by(sub_grp) %>%
  summarise_at(c("VE_LEN"), list(median = median, mean = mean, sd = sd, min=min,max=max))
knitr::kable(temp_df)
```

sub_grp	median	mean	sd	min	max
1	18.700	18.86219	2.257809	14.98	25.2700
2	22.820	22.33722	1.804621	19.71	24.6500
3	16.150	16.49427	1.377352	12.35	20.5700
4	23.935	23.82021	1.029144	20.79	25.0825

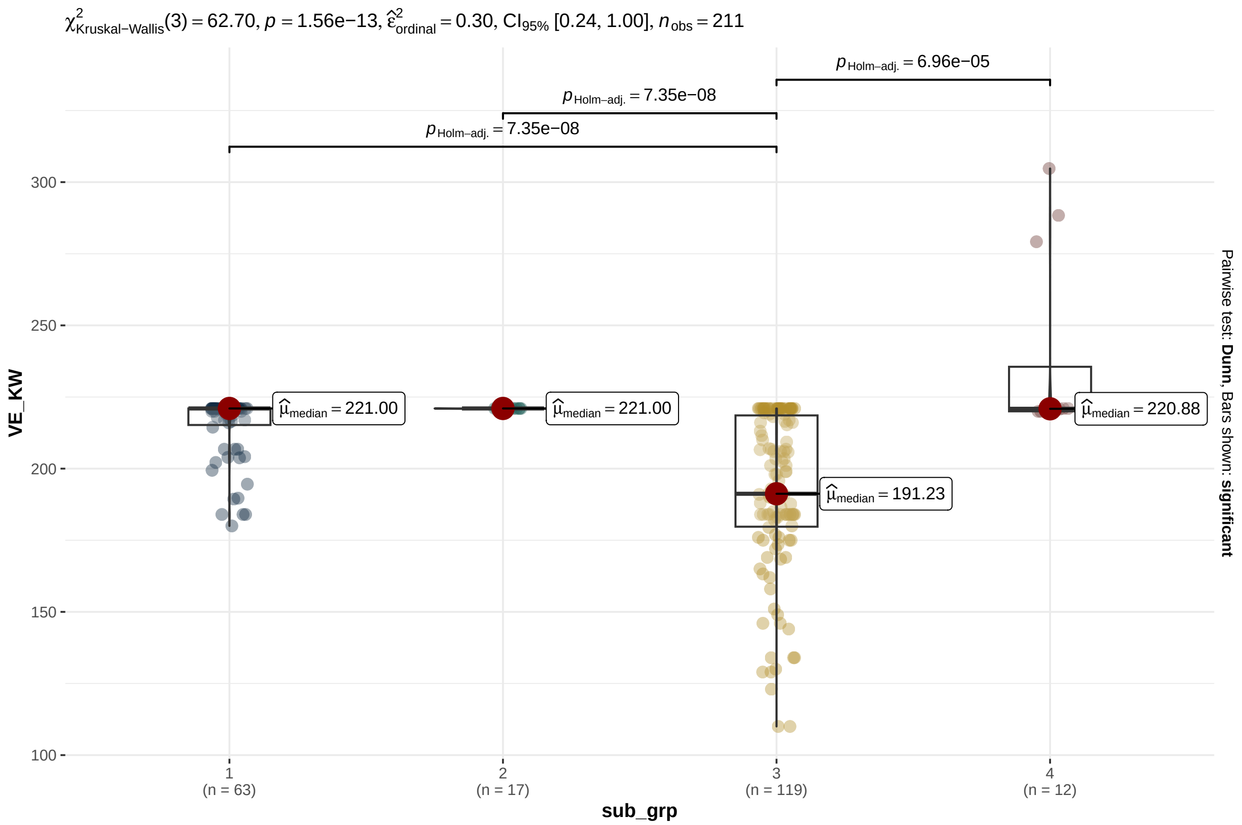
```
write_csv(temp_df,file = paste0(data_out,"musselabm-","VE_LEN",".csv"))




ggbetweenstats(
  data = df_stat,
  x = sub_grp,
  y = VE_KW,
  type = "nonparametric"
)+
  scale_colour_manual(
  values = serra,
  )
```

```
## Warning in (function (mapping = NULL, data = NULL, stat = "identity", position =
## "identity", : Ignoring unknown parameters: ‘segment.linetype‘
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```



```
temp_df <- df_stat %>%
  group_by(sub_grp) %>%
  summarise_at(c("VE_KW"), list(median = median, mean = mean, sd = sd, min=min,max=max))
knitr::kable(temp_df)
```

sub_grp	median	mean	sd	min	max
1	221.0000	214.4229	11.4967368	180	221.00
2	221.0000	220.9412	0.2425356	220	221.00
3	191.2308	191.4593	27.8762502	110	221.00
4	220.8846	238.0663	32.2645524	220	304.75

```
write_csv(temp_df,file = paste0(data_out,"musselabm-","VE_KW",".csv"))
```

End of the document. - Serra Örey