4.2_Visualisation_distribution.R

Iris Van Paemel

2024-06-13

4.2_Visualisation_distribution.R

Input

- Output/Species_Location.csv => header consists of "Specieslist", "BelgianCoast", "Getxo", "Koster", "Laeso", "Limfjord", "Plymouth", "Roscoff", "SwedishWestCoast", "TZS", "Vigo" /! replace with your own data from preparation file, these consist of amount of samples per ARMS location
- test_outputs/sea_distances or test_outputs/fly_distances => consists of files with distances and filenames: [species_name]*distancesTo*[location_name].csv => headers of these files consist of "x", "year", "month", "country"

Output

- Different directories are made to save the graphs in
- first graph: distribution of sea distances
- second graph: distribution of sea and fly distances
- third graph: distribution of sea distances colored by country
- fourth graph: distribution of sea distances colored by year

script + descriptions

- Load libraries (install if necessary)
- Set working directory to directory where the R-script is saved
- load in species_location file
- Reshape data from wide to long format
- Filter rows where 'value' is greater than 0

```
#####################################################
### LOAD LIBRARIES

library("tidyr")
library("ggplot2")
library("dplyr")
library("poliscidata")
#####################################################

#####################################################
### PREPARE LOCATION FILE
# Set working directory to directory where the R-script is saved
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

```r
# first load in species_location file
species_location <- read.csv("Output/1_Species_Location.csv")
# Reshape data from wide to long format
long <- pivot_longer(species_location, !Specieslist)
# Filter rows where 'value' is greater than 0
long <- long[long$value > 0, ]

##################################################
```

DISTRIBUTION SEA DISTANCES GRAPH (line 23 in 4.2_Visualisation_distribution.R)

- start the function for making graphs for each species and ARMS location separately
- print the species name and location as a check (iteration will show when the next species is processed)
- save the filename of the distances in 'distance_file' /! change if filename is different
- Read the file
- clean dataframe from rows with Inf in them
- convert distances from meters to kilometers
- select the distances that are below 40000km
- make the histogram with ggplot
- recommend keeping print(plot) in comments when iterating (R can't handle making a lot of graphs in an iteration)
- create a directory to save the plots in /! change directory name if needed
- function ends
- initiate error list to store error messages in
- start iterating over the long file and use function
- write error files

```r
# iterate over species_location file again using function and map()
# and read csv file per species

Distribution_seadistance <- function(species_name, species_location){
  # read csv file per species
  print(paste0("species_name: ", species_name))
  print(paste0("species_location: ", species_location))
  distance_file <- paste0("test_outputs/sea_distances/", species_name, "_distancesTo_", species_locatio

  distances <- read.table(distance_file, header = TRUE)
  # clean dataframe from rows with Inf in them
  distances <- distances[is.finite(distances$x), ]
  distances$x <- distances$x/1000
  distances <- subset(distances, x < 40000)

  ##############################################################################
  # make histograms of distances per species, with filtering on distance limit 40000
  ##############################################################################

  dist_plot <- ggplot(distances, aes(x = x)) +
    geom_histogram(binwidth = 50, fill = "blue", color = "black", boundary = 0) +
    labs(title = "Histogram of Distances", x = "Distance in km", y = "Frequency of species") +
    theme_bw() +
    #scale_fill_brewer(palette = "Set1") +
    ggtitle(paste0("Distribution of ", species_name, " from ", species_location)) +
    theme(plot.title = element_text(hjust = 0.5, size = 20, face = "bold"), # set title font size, plac
```

```
          plot.margin = margin(0.3, 0.3, 0.4, 0.4, "cm"),
          axis.text = element_text(size = 16),           # Set font size for axis numbers
          axis.title = element_text(size = 20)) +         # Set font size for axis titles
    scale_x_continuous(breaks = seq(0, 3000, by = 250), expand = c(0, 0)) +
    scale_y_continuous(expand = c(0, 0)) +
    coord_cartesian(xlim = c(0, 3000)) # Use coord_cartesian for setting limits

  # print(dist_plot)
  # create a directory to save the plots in

  if(!dir.exists("test_outputs/sea_distribution_plots")) {
    dir.create("test_outputs/sea_distribution_plots")
    ggsave(filename = paste0("test_outputs/sea_distribution_plots/plot_", species_name, "_from_", speci
           plot = dist_plot, width = 50, height = 30, units = "cm", dpi = 900)
  } else {
    ggsave(filename = paste0("test_outputs/sea_distribution_plots/plot_", species_name, "_from_", speci
           plot = dist_plot, width = 50, height = 30, units = "cm", dpi = 900)
  }
}
################################# END FUNCTION

# initiate error list to store error messages in
error <- c()

# executing iteration over the long file
plot <- Map(Distribution_seadistance, long$Specieslist, long$name)

# write error files
file_error <- paste0("test_outputs/errors_graph_sea_distances.csv",)
write.table(error, file = file_error, append = TRUE, quote = FALSE,
            col.names = FALSE, row.names = FALSE)
```

DISTRIBUTION SEA + FLY DISTANCES COMBINED GRAPH (line 87 in 4.2_Visualisation_distribution.R)

- start function
- make variables with filenames in which distances are saved /! change if filename is different
- read the files
- transform Inf to NA
- convert each dataframe from m to km and select the distances smaller than 4000
- make a combined dataframe to use for the graph
- put print in comments when iterating over function
- check if directory exists, save the plot
- function ending
- map() usage to iterate over the function

```
#############################################################
# Make combined histogram of sea distances and fly distances
#############################################################

Distribution_combDistance <- function(species_name, species_location){
  # make variables with filenames in which distances are saved
  sea_distance_file <- paste0("theoretical_data/sea_distances/", species_name, "_distancesTo_", species
  fly_distance_file <- paste0("theoretical_data/fly_distances/", species_name, "_distancesTo_", species
```

```r
# read the files
sea_distances <- read.table(sea_distance_file)
fly_distances <- read.table(fly_distance_file)
# transform Inf to NA
sea_distances$x[is.infinite(sea_distances$x)] <- NA
fly_distances$x[is.infinite(fly_distances$x)] <- NA

# convert each dataframe from m to km and select the distances smaller than 40000km
sea_distances$x <- sea_distances$x/1000
sea_distances <- subset(sea_distances, x < 40000)
sea_distances$type <- "sea_distance" # type will be a column in combined dataframe
fly_distances$x <- fly_distances$x/1000
fly_distances <- subset(fly_distances, x < 40000)
fly_distances$type <- "fly_distance" # type will be a column in combined dataframe

# make a combined dataframe to use for the graph
combined_distances <- rbind(sea_distances, fly_distances)

p <- combined_distances %>%
  ggplot( aes(x=x, fill=type)) +
  geom_histogram(binwidth = 50, color="#e9ecef", alpha=0.6, position = 'identity') +
  theme_bw() +
  #scale_fill_brewer(palette = "Set1") +
  labs(x = "Distance in km", y = "Frequency") +
  ggtitle(paste0("Distribution of ", species_name, " from ", species_location)) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 20, face = "bold"), # set title font size, placemen
    plot.margin = margin(0.3, 0.3, 0.4, 0.4, "cm"),
    axis.text = element_text(size = 16),  # Set font size for axis numbers
    axis.title = element_text(size = 20), # Set font size for title
    legend.title = element_text(size = 18, face="bold"), # Settings for legend title
    legend.text = element_text(size = 16)) +  # settings for legend text
  scale_x_continuous(breaks = seq(0, 3000, by = 250), expand = c(0, 0)) +  # settings for x axis
  scale_y_continuous(expand = c(0, 0)) +
  # used expand to make sure the axes are on the lines of the axes and not above them floating
  coord_cartesian(xlim = c(0, 3000)) + # Use coord_cartesian for setting limits
  # set legend title and labels
  scale_fill_discrete(
    name = "Distance type",
    breaks = c("sea_distance", "fly_distance"),
    labels = c("Sea distance", "Fly distance")) +
  labs(x = "Distance in km", y = "Frequency")

#print(p)

# check if directory exists, save the plot
if(!dir.exists(paste0("test_outputs/combined_distribution_plots"))) {
  dir.create(paste0("test_outputs/combined_distribution_plots"))
  ggsave(filename = paste0("test_outputs/combined_distribution_plots/plotcomb_", species_name, "_from_
         plot = p, width = 60, height = 30, units = "cm", dpi = 900)
} else {
  ggsave(filename = paste0("test_outputs/combined_distribution_plots/plotcomb_", species_name, "_from_
         plot = p, width = 60, height = 30, units = "cm", dpi = 900)
```

```
  }
}

# executing iteration for the long file!
plot <- Map(Distribution_combDistance, long$Specieslist, long$name)
```

DISTRIBUTION SEA DISTANCES GRAPH + LOCATIONS (line 155 in 4.2_Visualisation_distribution.R)

- start the function
- read csv files per species and print species name and ARMS location name /! change if filename is different
- save the filename in the variable 'distance_file'
- read the distance file and convert m to km
- select distances that are below 40000km
- make the plot, don't print when iterating
- create a directory to save the plots in
- execute the iteration with mapt() over the function

```
############################################################
# Make histograms of locations
############################################################

Location_histograms <- function(species_name, species_location){

  print(paste0("species_name: ", species_name))
  print(paste0("species_location: ", species_location))
  # make variable with filename
  distance_file <- paste0("test_outputs/sea_distances/", species_name, "_distancesTo_", species_locatio
  # read csv file per species
  distance_file <- read.table(distance_file, header = TRUE, sep = ",")
  # convert to meters
  distance_file$x <- distance_file$x/1000
  # select only distances below 40000km
  distances <- subset(distance_file, x < 40000)

  distances <- subset(distance_file, x < 40000)

  ### PLOT

  country_plot <- ggplot(distances, aes(x = x, fill = country)) +
    geom_histogram(binwidth = 50, boundary = 0, position = "stack") +  # adjust the binwidth to persona
    labs(title = paste0("Frequencies of Sea distances/country for ", species_name," in ", species_locati
                        x = "Sea distance in km", y = "Frequency of species") +
    theme_bw() +
    #scale_fill_brewer(palette = "Set1") +  # You can choose a different palette if you like
    theme(plot.title = element_text(hjust = 0.5, size = 20, face = "bold"), # set title font size, plac
          plot.margin = margin(0.3, 0.3, 0.4, 0.4, "cm"),
          axis.text = element_text(size = 16),           # Set font size for axis numbers
          axis.title = element_text(size = 20),
          legend.title = element_text(size = 14),    # Increase legend title size
          legend.text = element_text(size = 12),     # Increase legend text size
          legend.key.size = unit(1.5, "lines")) +    # Increase legend key size
    scale_x_continuous(breaks = seq(0, 3000, by = 250), expand = c(0, 0)) +
```

```
    scale_y_continuous(expand = c(0, 0)) +
    coord_cartesian(xlim = c(0, 3000)) # Use coord_cartesian for setting limits

  #print(country_plot)
  # create a directory to save the plots in
  if(!dir.exists("test_outputs/sea_distribution_country_plots")) {
    dir.create("test_outputs/sea_distribution_country_plots")
    ggsave(filename = paste0("theoretical_data/plot_", species_name, "_from_", species_location, ".png")
           plot = country_plot, width = 60, height = 30, units = "cm", dpi = 900)
  } else {
    ggsave(filename = paste0("test_outputs/sea_distribution_country_plots/plot_", species_name, "_from_
           plot = country_plot, width = 60, height = 30, units = "cm", dpi = 900)
  }
}

# executing iteration for the long file!
plot <- Map(Location_histograms, long$Specieslist, long$name)
```

DISTRIBUTION SEA DISTANCES GRAPH + YEARS (line 208 in 4.2_Visualisation_distribution.R)

- start the function
- print the species name and species_location
- save the filename in a variable /! change if filename is different
- read the file
- convert the distances to km
- select the distances below 40000km
- make a function which assigns years to specific year categories
- make a variable with preferenced year categories
- make the plot, don't print the plot when iterating
- create a directory to save the plots in
- start iterating over the function with map()

```
############################################################
# Make histograms of year categories
############################################################

Year_histograms <- function(species_name, species_location){

  print(paste0("species_name: ", species_name))
  print(paste0("species_location: ", species_location))
  # make variable with filename
  distance_file <- paste0("theoretical_data/sea_distances/", species_name,
                          "_distancesTo_", species_location, ".csv")

  # read the csv files
  distance_file <- read.table(distance_file, header = TRUE)
  # convert to meters
  distance_file$x <- distance_file$x/1000
  # select only distances below 40000km
  distances <- subset(distance_file, x < 40000)

  # make a function which assigns years to specific year categories
  assign_category <- function(year) {
```

```
    if (is.na(year)) {
      return(NA)   # return NA when year is not present
    }
    for (category in year_categories) {
      range <- as.numeric(unlist(strsplit(category, "-"))) # save years as numeric without "-"
      if (year >= range[1] & year < range[2]) {  # if the year falls into this category
        return(category) # return this category
      }
    }
    return(NA) # If year doesn't fall into any category, return NA
  }
  # make a variable with preferenced year categories
  year_categories <- c("1965-1985","1985-1990",
                       "1990-1995","1995-2000","2000-2005","2005-2010","2010-2015",
                       "2015-2020","2020-2025")
  # Apply function to create new column with year categories
  distances$year_category <- sapply(distances$year, assign_category)

  ##################################
  ### PLOT

  year_plot <- ggplot(distances, aes(x = x, fill = year_category)) +
    geom_histogram(binwidth = 50, boundary = 0, position = "stack") +  # adjust the binwidth to persona
    labs(title = paste0("Frequencies of Sea distances/year for ", species_name," in ", species_location
         x = "Sea distance in km", y = "Frequency of species") +
    theme_bw() +
    scale_fill_brewer(palette = "YlOrRd", na.value = "black") + # You can choose a different palette if
    theme(plot.title = element_text(hjust = 0.5, size = 20, face = "bold"), # set title font size, plac
          plot.margin = margin(0.3, 0.3, 0.4, 0.4, "cm"),
          axis.text = element_text(size = 16),           # Set font size for axis numbers
          axis.title = element_text(size = 20),
          legend.title = element_text(size = 14),    # Increase legend title size
          legend.text = element_text(size = 12),     # Increase legend text size
          legend.key.size = unit(1.5, "lines")) +    # Increase legend key size
    scale_x_continuous(breaks = seq(0, 3000, by = 250), expand = c(0, 0)) +
    scale_y_continuous(expand = c(0, 0)) +
    coord_cartesian(xlim = c(0, 3000)) # Use coord_cartesian for setting limits

  #print(year_plot)
  # create a directory to save the plots in
  if(!dir.exists("test_outputs/sea_distribution_year_plots")) {
    dir.create("test_outputs/sea_distribution_year_plots")
    ggsave(filename = paste0("test_outputs/sea_distribution_year_plots/plot_", species_name, "_from_", s
           plot = year_plot, width = 60, height = 30, units = "cm", dpi = 900)
  } else {
    ggsave(filename = paste0("test_outputs/sea_distribution_year_plots/plot_", species_name, "_from_", s
           plot = year_plot, width = 60, height = 30, units = "cm", dpi = 900)
  }
}

# executing iteration for the long file!
plot <- Map(Year_histograms, long$Specieslist, long$name)
```