

Script to validate points in ReSurvey database using RS data

Alicia Valdés

19 febrero 2025

This R script is used to make a first attempt to validate the points in the ReSurvey database using RS indicators (NDVI and NDMI).

Load libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## vforcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyrr    1.3.1
## v purrr    1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(here)
```

```
## here() starts at C:/Users/jimenezalfaro/OneDrive - Universidad de Oviedo/IMIB/Analyses/MOTIVATE/MOTIV
```

```
library(sf)
```

```
## Linking to GEOS 3.13.0, GDAL 3.10.1, PROJ 9.5.1; sf_use_s2() is TRUE
```

```
library(gridExtra)
```

```
##
## Adjuntando el paquete: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(sf)
library(rnaturalearth)
library(readxl)
```

Define printall function

```
printall <- function(tibble) {
  print(tibble, width = Inf)
}
```

Read ReSurvey data with RS indicators

```
db_resurv_RS<-read_tsv(
  here("data", "clean", "db_resurv_RS.csv"),
  col_types = cols(
    # Dynamically specify EUNIS columns as character
    .default = col_guess(), # Default guessing for other columns
    EUNISa = col_character(),
    EUNISb = col_character(),
    EUNISc = col_character(),
    EUNISd = col_character(),
    EUNISa_1 = col_character(),
    EUNISa_2 = col_character(),
    EUNISa_3 = col_character(),
    EUNISa_4 = col_character(),
    EUNISb_1 = col_character(),
    EUNISb_2 = col_character(),
    EUNISb_3 = col_character(),
    EUNISb_4 = col_character(),
    EUNISc_1 = col_character(),
    EUNISc_2 = col_character(),
    EUNISc_3 = col_character(),
    EUNISc_4 = col_character(),
    EUNISd_1 = col_character(),
    EUNISd_2 = col_character(),
    EUNISd_3 = col_character(),
    EUNISd_4 = col_character(),
    EUNISa_1_descr = col_character(),
    EUNISb_1_descr = col_character(),
    EUNISc_1_descr = col_character(),
    EUNISd_1_descr = col_character(),
    EUNIS_assignment = col_character(),
    EUNISa_2_descr = col_character(),
    EUNISa_3_descr = col_character(),
    EUNISa_4_descr = col_character(),
    EUNISb_2_descr = col_character(),
    EUNISb_3_descr = col_character(),
    EUNISb_4_descr = col_character(),
```

```

    EUNISc_2_descr = col_character(),
    EUNISc_3_descr = col_character(),
    EUNISc_4_descr = col_character(),
    EUNISd_2_descr = col_character(),
    EUNISd_3_descr = col_character(),
    EUNISd_4_descr = col_character()
  )
)

```

No parsing issues!

Several EUNIS level 1 assigned

Number of rows where there is more than one EUNIS 1 assigned, and they are different among them. See what to do with these later! So far I take EUNISa_1.

```

nrow(db_resurv_RS %>%
      # Rows with more than one EUNIS 1 assigned
      filter(!is.na(EUNISb_1)) %>%
      filter(EUNISa_1 != EUNISb_1 | EUNISb_1 != EUNISc_1 | EUNISa_1 != EUNISc_1))

## [1] 102

```

Tibble with selected columns

```

db_resurv_RS_short <- db_resurv_RS %>%
  select(PlotObservationID, Country, RS_CODE, `ReSurvey site`, `ReSurvey plot`,
         `Manipulate (y/n)`, `Type of manipulation`, Lon_updated, Lat_updated,
         `Location method`, `Location uncertainty (m)`, EUNISa_1,
         EUNISa_1_descr, EUNISa_2, EUNISa_2_descr, EUNISa_3, EUNISa_3_descr,
         EUNISa_4, EUNISa_4_descr, date, year, biogeo, unit, year_RS, source,
         Lon_RS, Lat_RS, NDVI, NDMI, RS_data)

```

Missing data checks

Do when all RS data is ready!

Flag when year is different between RS data and ReSurvey db

```

db_resurv_RS_short <- db_resurv_RS_short %>%
  mutate(year_diff = year != year_RS)

```

```
db_resurv_RS_short %>% count(year_diff)
```

```
## # A tibble: 3 x 2
##   year_diff     n
##   <lg1>     <int>
## 1 FALSE      20896
## 2 TRUE       2
## 3 NA        402682
```

Flag when coordinates are different between RS data and ReSurvey db

```
db_resurv_RS_short <- db_resurv_RS_short %>%
  mutate(Lon_diff = Lon_updated != Lon_RS,
        Lat_diff = Lat_updated != Lat_RS)
```

```
db_resurv_RS_short %>% count(Lon_diff)
```

```
## # A tibble: 3 x 2
##   Lon_diff     n
##   <lg1>     <int>
## 1 FALSE      20621
## 2 TRUE       277
## 3 NA        402682
```

```
db_resurv_RS_short %>% count(Lat_diff)
```

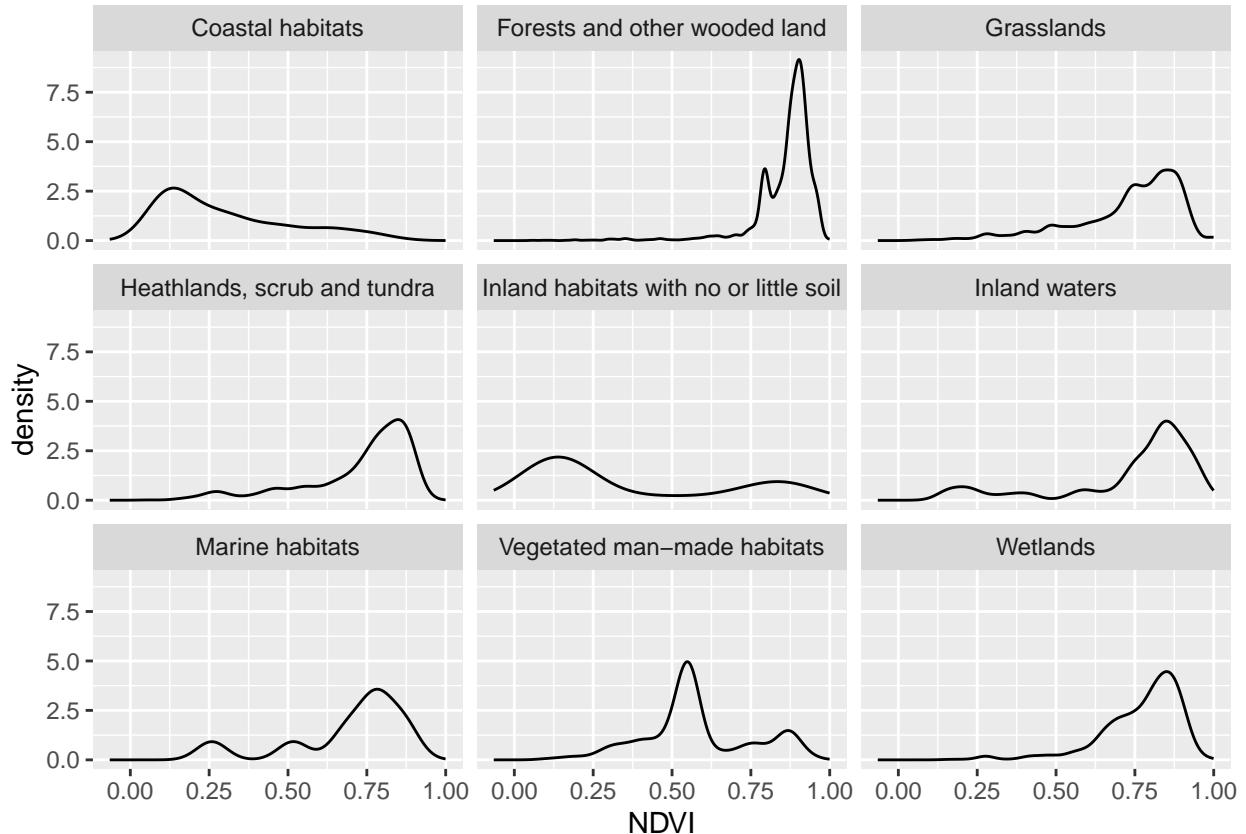
```
## # A tibble: 3 x 2
##   Lat_diff     n
##   <lg1>     <int>
## 1 FALSE      20586
## 2 TRUE       312
## 3 NA        402682
```

NDVI and NDMI would need to be calculated again for those?

Distributions NDVI and NDMI for all bioregions

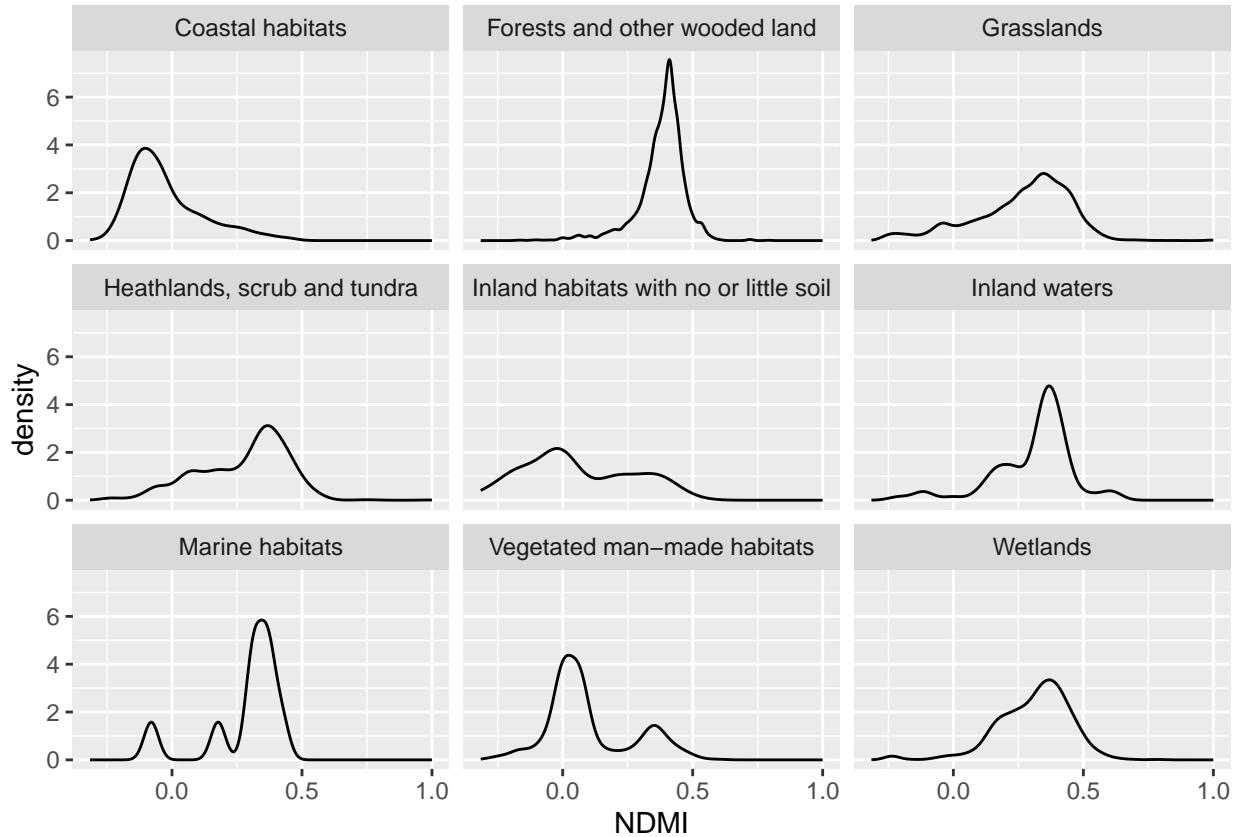
```
ggplot(db_resurv_RS_short %>% filter(!is.na(EUNISa_1_descr)), aes(x = NDVI)) +
  facet_wrap(~ EUNISa_1_descr) +
  geom_density()
```

```
## Warning: Removed 233475 rows containing non-finite outside the scale range
## ('stat_density()'').
```



```
ggplot(db_resurv_RS_short %>% filter(!is.na(EUNISa_1_descr)), aes(x = NDMI)) +
  facet_wrap(~ EUNISa_1_descr) +
  geom_density()
```

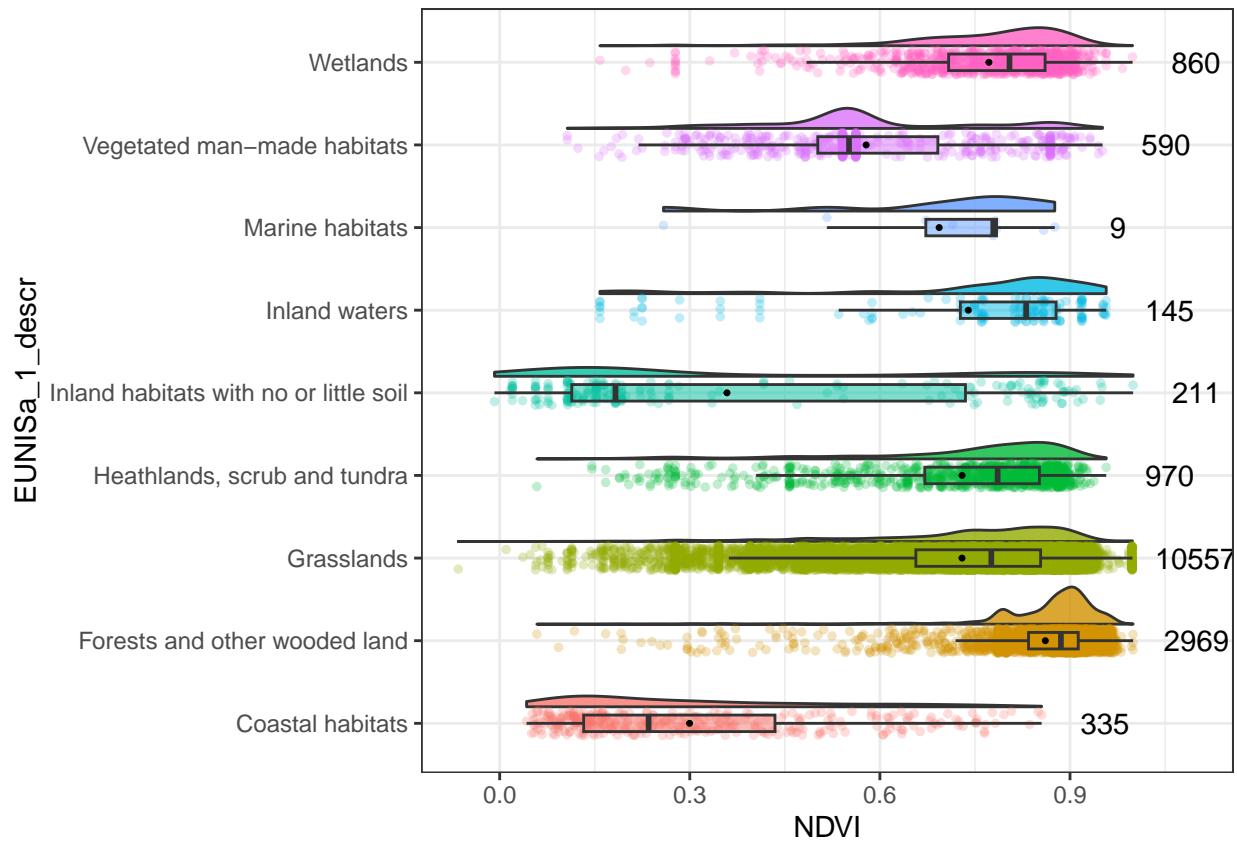
```
## Warning: Removed 233475 rows containing non-finite outside the scale range
## ('stat_density()').
```



Violin + boxplot + points

```
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd97121f7f9ce947837e")

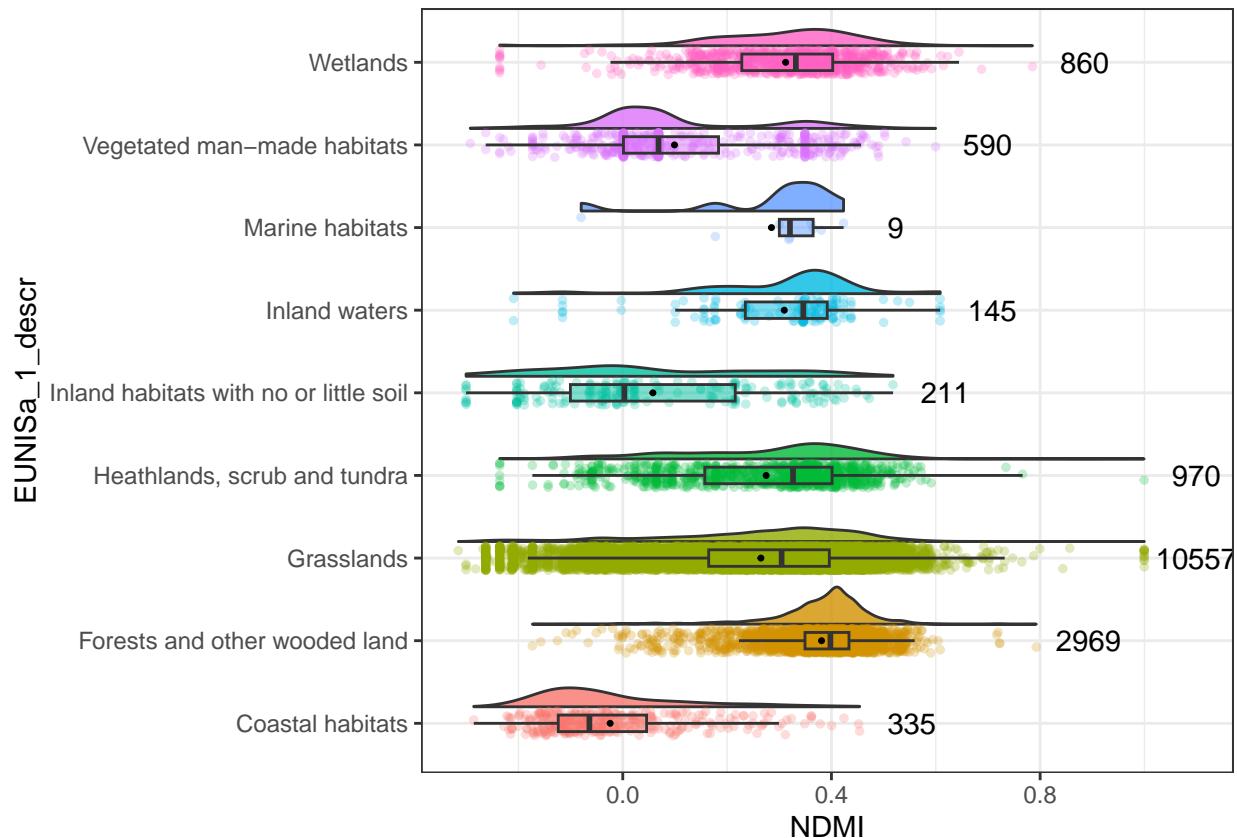
ggplot(data = filter(db_resurv_RS_short %>%
    filter(RS_data == T & !is.na(EUNISa_1_descr))),
    aes(x = EUNISa_1_descr, y = NDVI, fill = EUNISa_1_descr)) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = NDVI, color = EUNISa_1_descr),
             position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 1) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                label = length(x)),
              geom = "text", aes(label = ..label..), vjust = 0.5) +
  labs(y = "NDVI", x = "EUNISa_1_descr") +
  guides(fill = FALSE, color = FALSE) +
  theme_bw() + coord_flip()
```



```

ggsave(
  here("output", "figures", "NDVI.tif"),
  width = 21, height = 29.7, units = "cm", dpi = 300)
ggplot(data = filter(db_resurv_RS_short %>%
  filter(RS_data == T & !is.na(EUNISa_1_descr))),
  aes(x = EUNISa_1_descr, y = NDMI, fill = EUNISa_1_descr)) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = NDMI, color = EUNISa_1_descr),
             position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 1) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                label = length(x)),
              geom = "text", aes(label = ..label..), vjust = 0.5) +
  labs(y = "NDMI", x = "EUNISa_1_descr") +
  guides(fill = FALSE, color = FALSE) +
  theme_bw() + coord_flip()

```



```
ggsave(
  here("output", "figures", "NDMI.tif"),
  width = 21, height = 29.7, units = "cm", dpi = 300)
```

Distributions NDVI and NDMI for each bioregion

```
# Get unique values of EUNISa_1_descr, excluding NA
eunis_levels <- unique(db_resurv_RS_short$EUNISa_1_descr
                         [!is.na(db_resurv_RS_short$EUNISa_1_descr)])
```

```
# Generate plots using purrr::map
eunis_levels %>%
  set_names() %>% # Set names to access plots by EUNISa_1_descr
  map(~
    ggplot() +
      # Density plot for each biogeog (filled)
      geom_density(data = filter(db_resurv_RS_short, EUNISa_1_descr == .x),
                   aes(x = NDVI, fill = biogeog), alpha = 0.5) +
      # Density plot for NDVI in red with thicker line
      geom_density(data = filter(db_resurv_RS_short, EUNISa_1_descr == .x),
                   aes(x = NDVI),
                   color = "red", alpha = 0.5, size = 1,
                   show.legend = FALSE) +
```

```

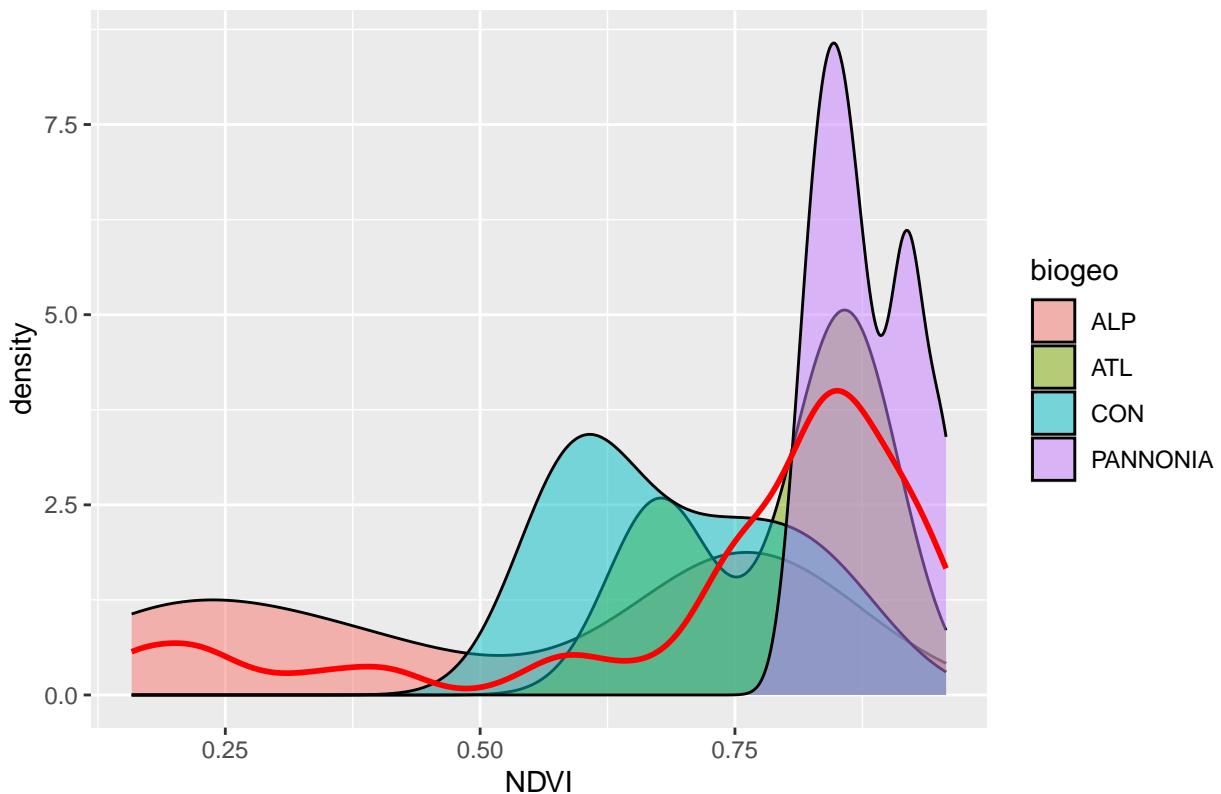
    labs(title = .x)
) %>%
walk(print)

## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

## Warning: Removed 5253 rows containing non-finite outside the scale range
## ('stat_density()').
## Removed 5253 rows containing non-finite outside the scale range
## ('stat_density()').

```

Inland waters



```

## Warning: Removed 27808 rows containing non-finite outside the scale range
## ('stat_density()').

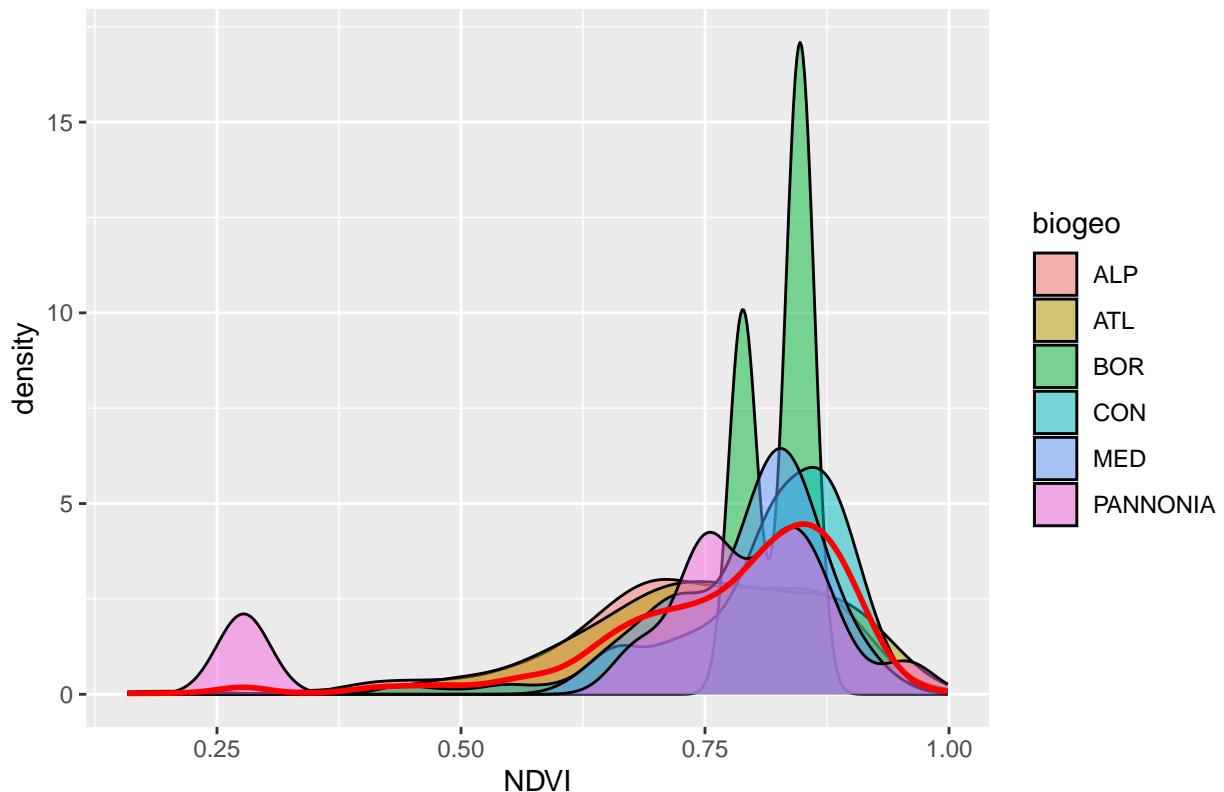
```

```

## Warning: Removed 27808 rows containing non-finite outside the scale range
## ('stat_density()').

```

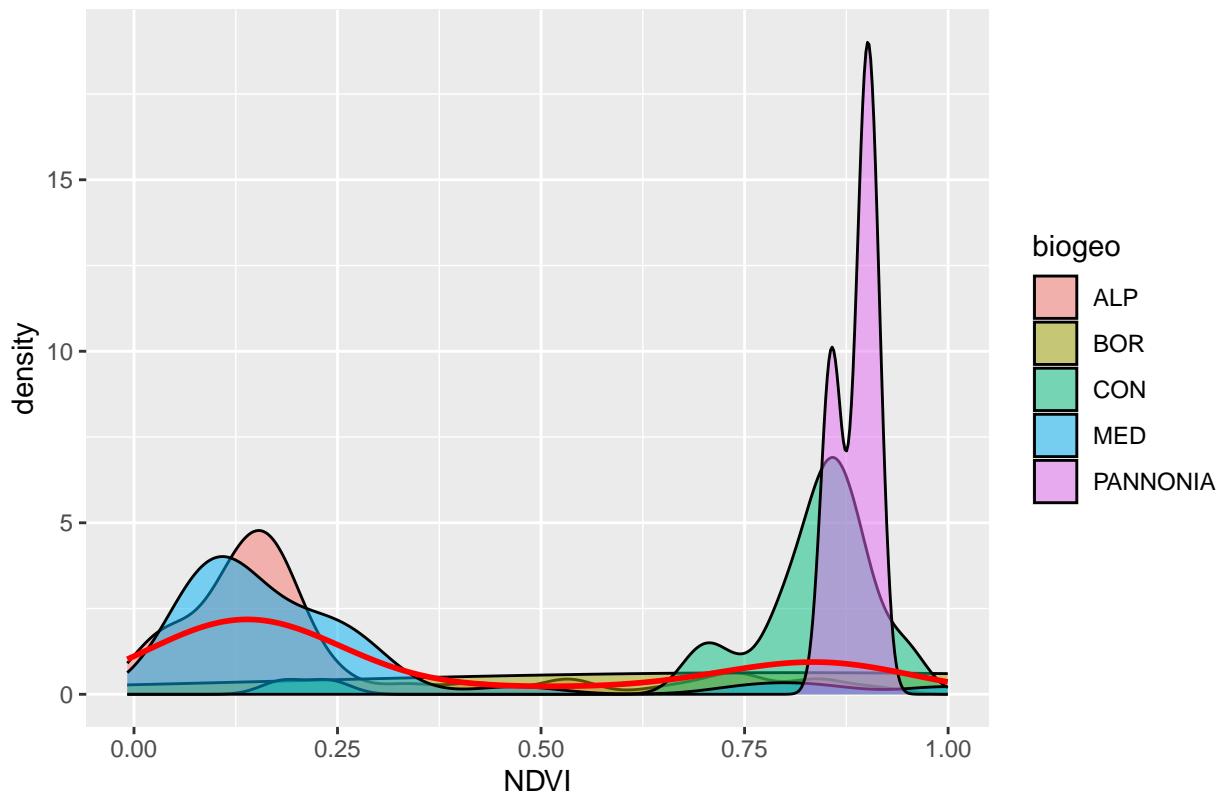
Wetlands



```
## Warning: Removed 3686 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 3686 rows containing non-finite outside the scale range
## ('stat_density()').
```

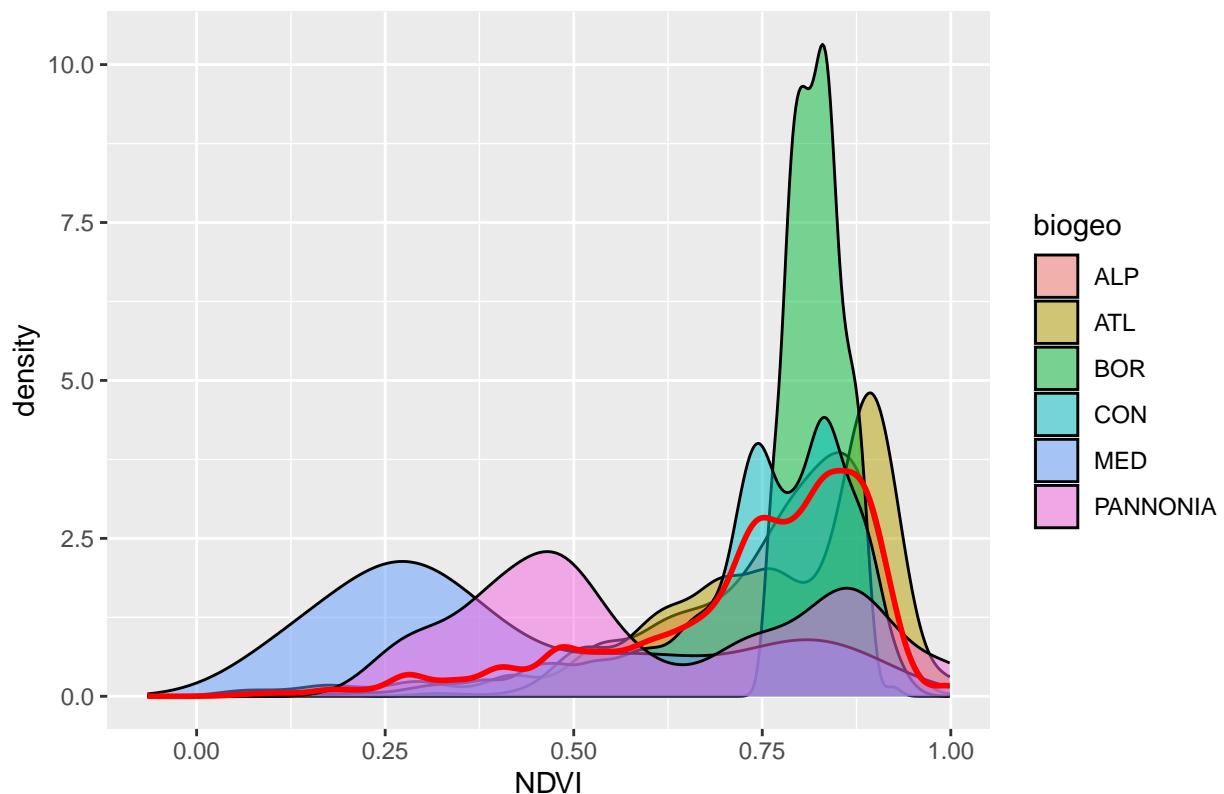
Inland habitats with no or little soil



```
## Warning: Removed 100537 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 100537 rows containing non-finite outside the scale range
## ('stat_density()').
```

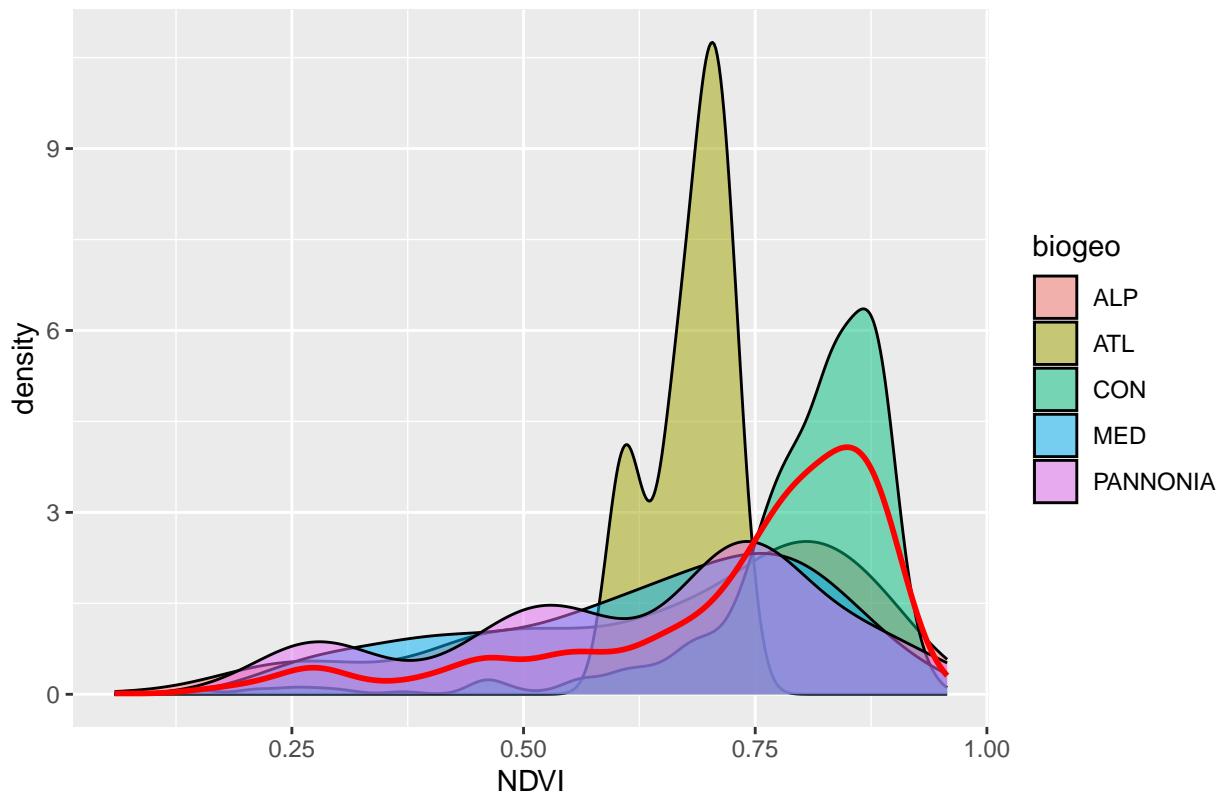
Grasslands



```
## Warning: Removed 23363 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 23363 rows containing non-finite outside the scale range
## ('stat_density()').
```

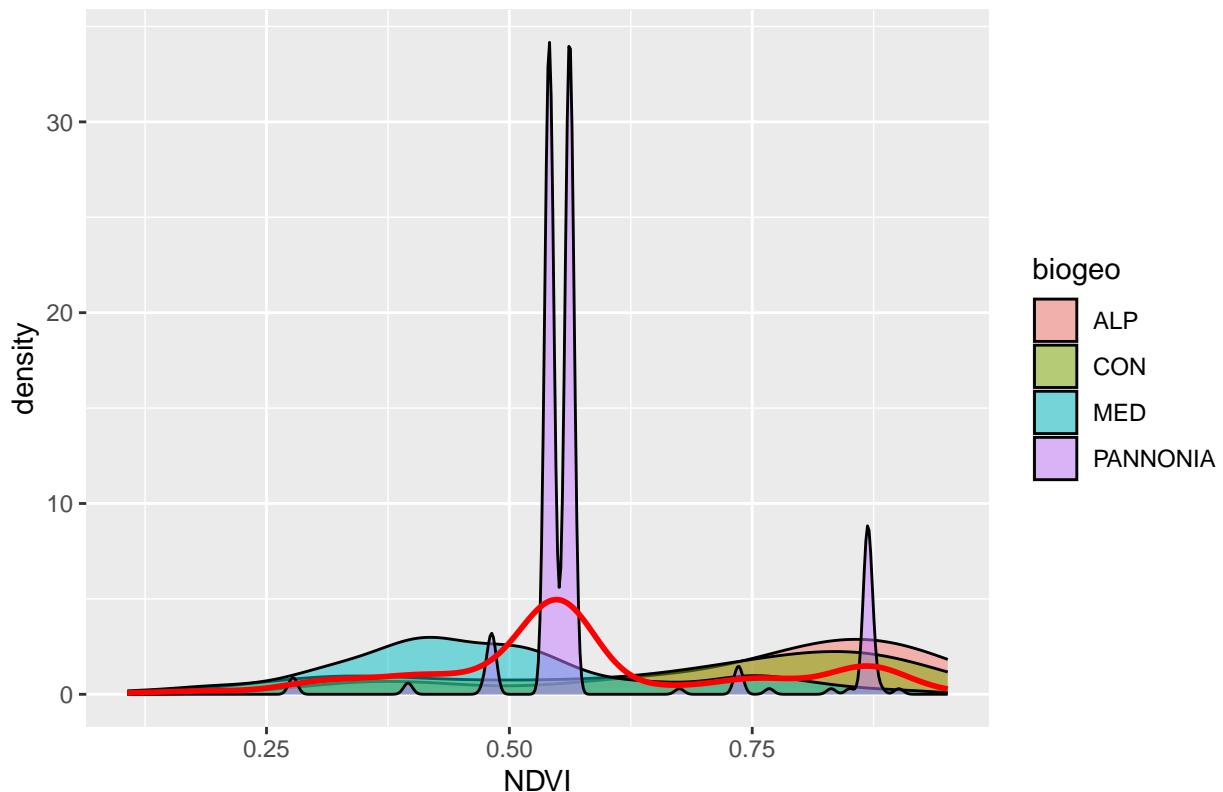
Heathlands, scrub and tundra



```
## Warning: Removed 7465 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 7465 rows containing non-finite outside the scale range
## ('stat_density()').
```

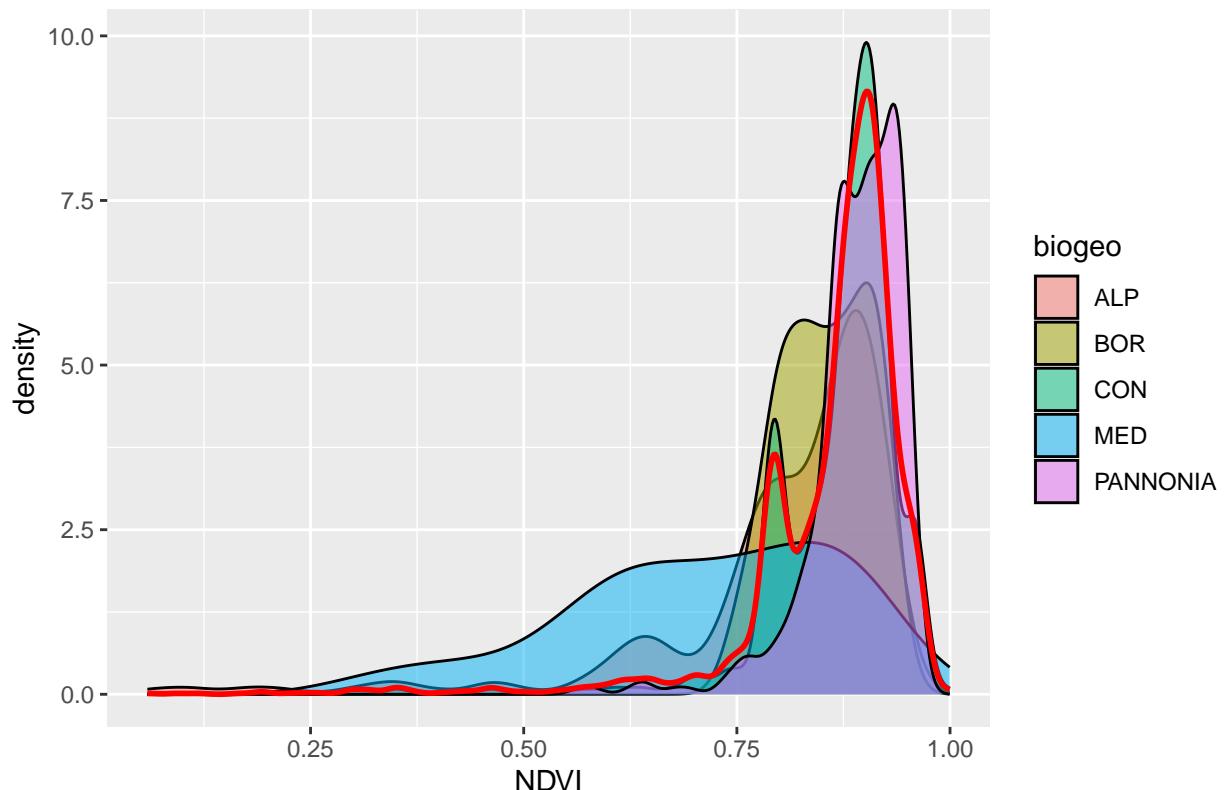
Vegetated man-made habitats



```
## Warning: Removed 21413 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 21413 rows containing non-finite outside the scale range
## ('stat_density()').
```

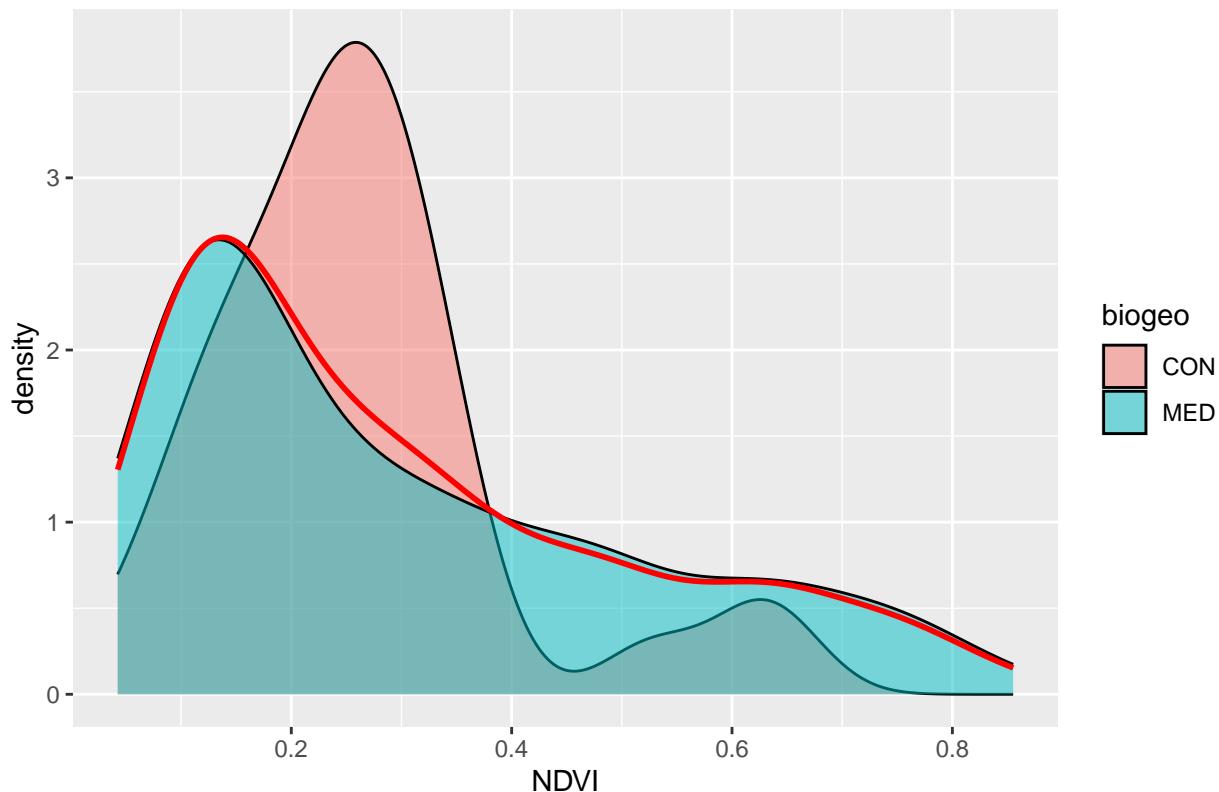
Forests and other wooded land



```
## Warning: Removed 31477 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 31477 rows containing non-finite outside the scale range
## ('stat_density()').
```

Coastal habitats



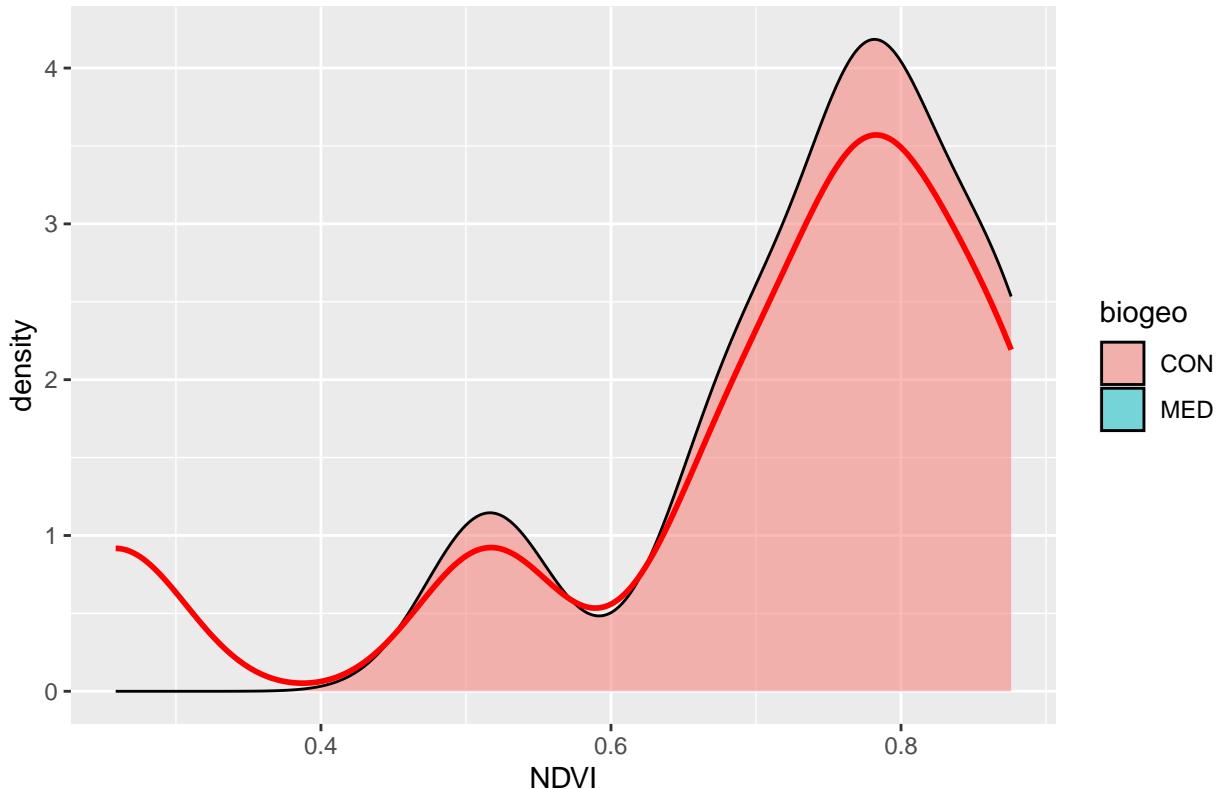
```
## Warning: Removed 12473 rows containing non-finite outside the scale range
## ('stat_density()').

## Warning: Groups with fewer than two data points have been dropped.

## Warning: Removed 12473 rows containing non-finite outside the scale range
## ('stat_density()').

## Warning in max(ids, na.rm = TRUE): ningun argumento finito para max; retornando
## -Inf
```

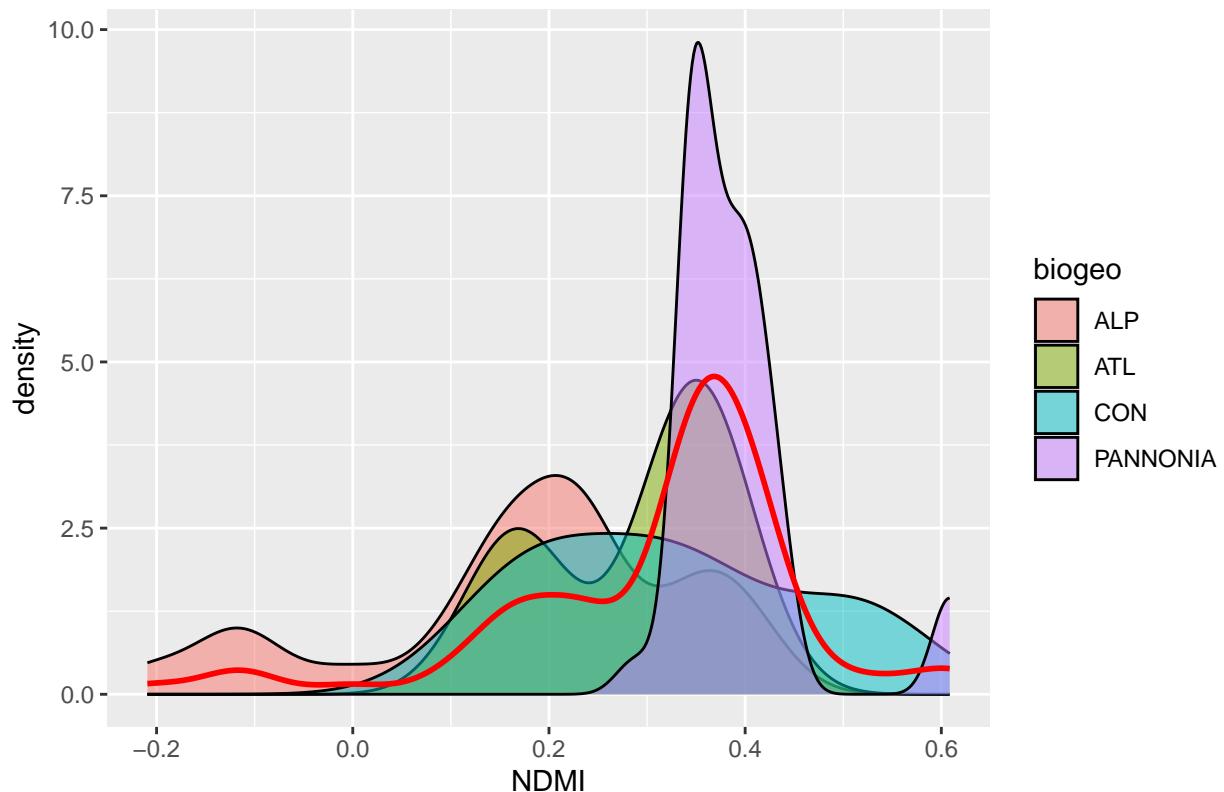
Marine habitats



```
eunis_levels %>%
  set_names() %>% # Set names to access plots by EUNISa_1_descr
  map(~
    ggplot() +
      # Density plot for each biogeographic region (filled)
      geom_density(data = filter(db_resurv_RS_short, EUNISa_1_descr == .x),
                   aes(x = NDMI, fill = biogeo), alpha = 0.5) +
      # Density plot for NDMI in red with thicker line
      geom_density(data = filter(db_resurv_RS_short, EUNISa_1_descr == .x),
                   aes(x = NDMI),
                   color = "red", alpha = 0.5, size = 1,
                   show.legend = FALSE) +
      labs(title = .x)
  ) %>%
  walk(print)

## Warning: Removed 5253 rows containing non-finite outside the scale range
## ('stat_density()').
## Removed 5253 rows containing non-finite outside the scale range
## ('stat_density()').
```

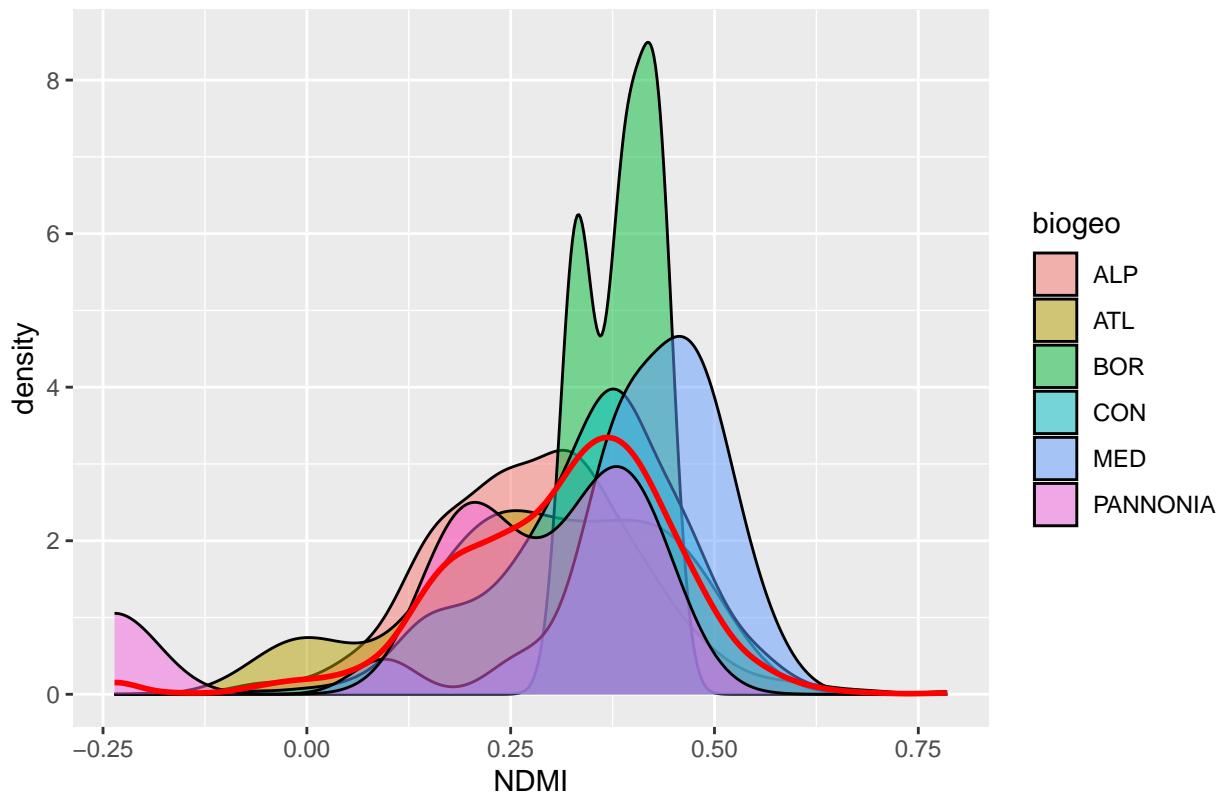
Inland waters



```
## Warning: Removed 27808 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 27808 rows containing non-finite outside the scale range
## ('stat_density()').
```

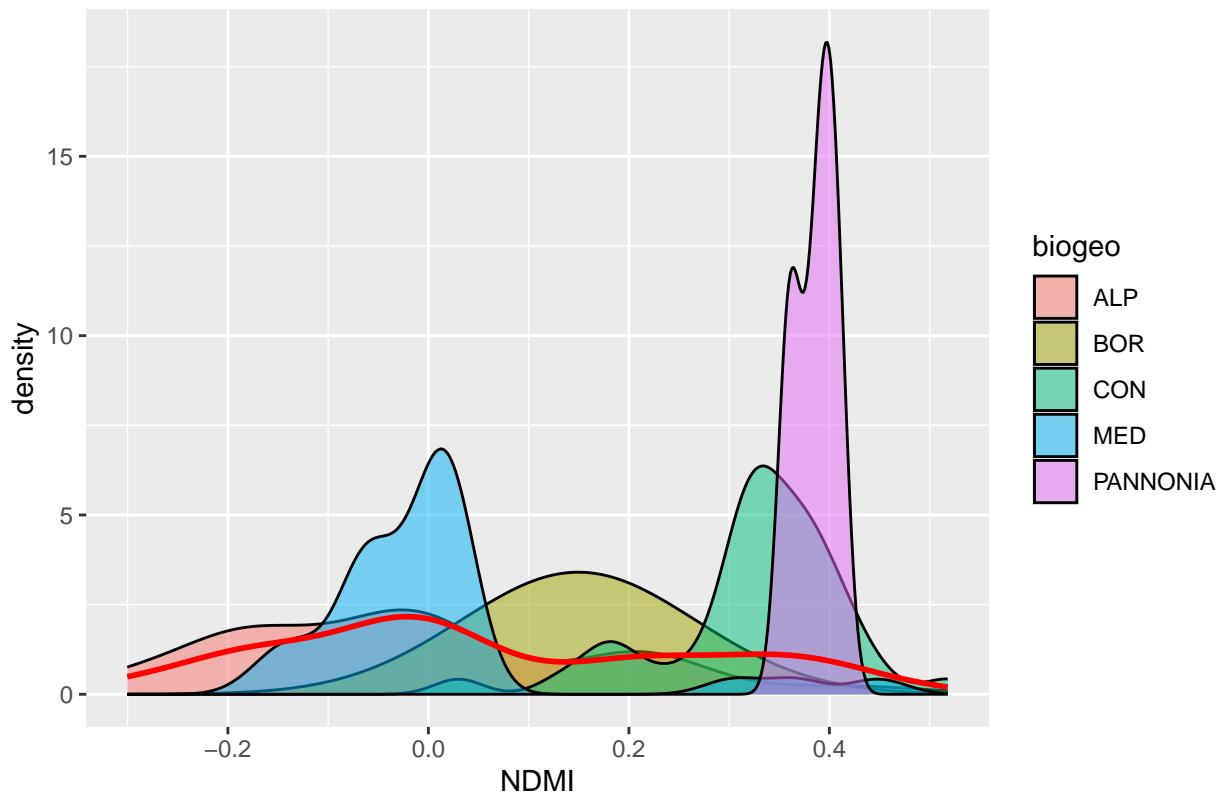
Wetlands



```
## Warning: Removed 3686 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 3686 rows containing non-finite outside the scale range
## ('stat_density()').
```

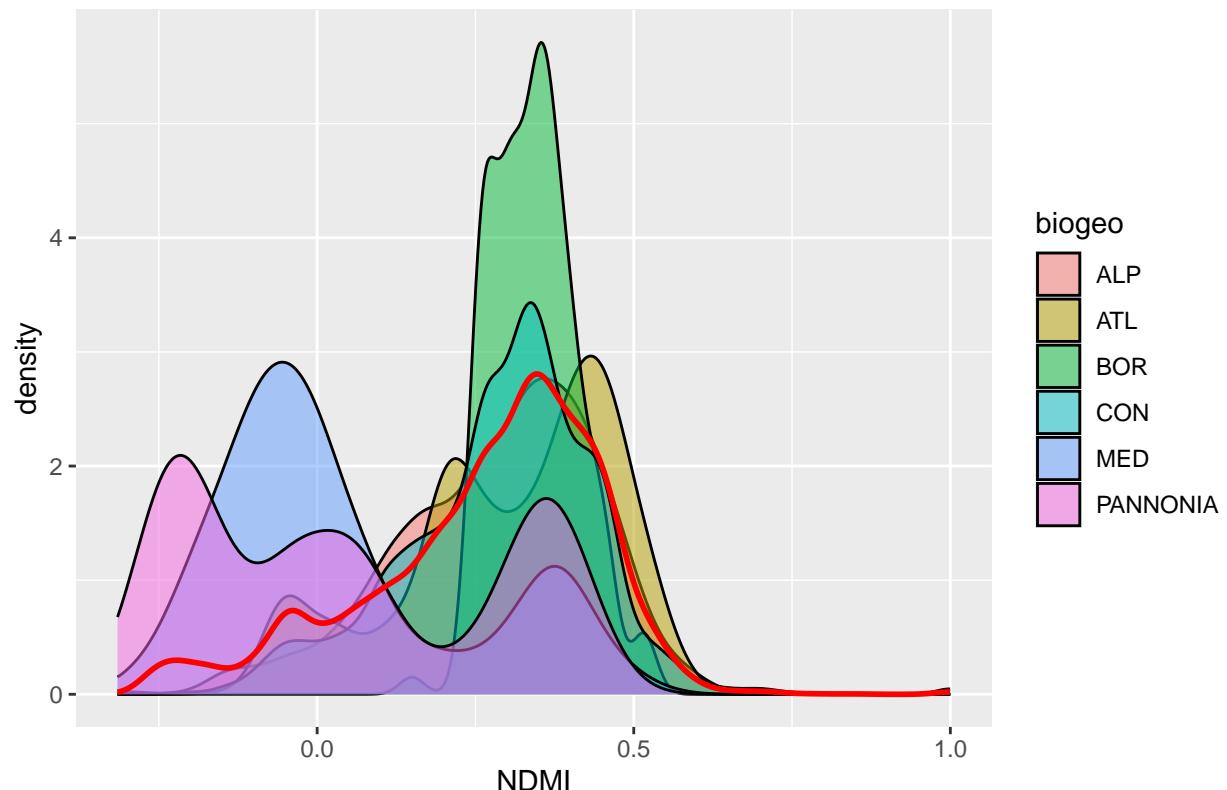
Inland habitats with no or little soil



```
## Warning: Removed 100537 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 100537 rows containing non-finite outside the scale range
## ('stat_density()').
```

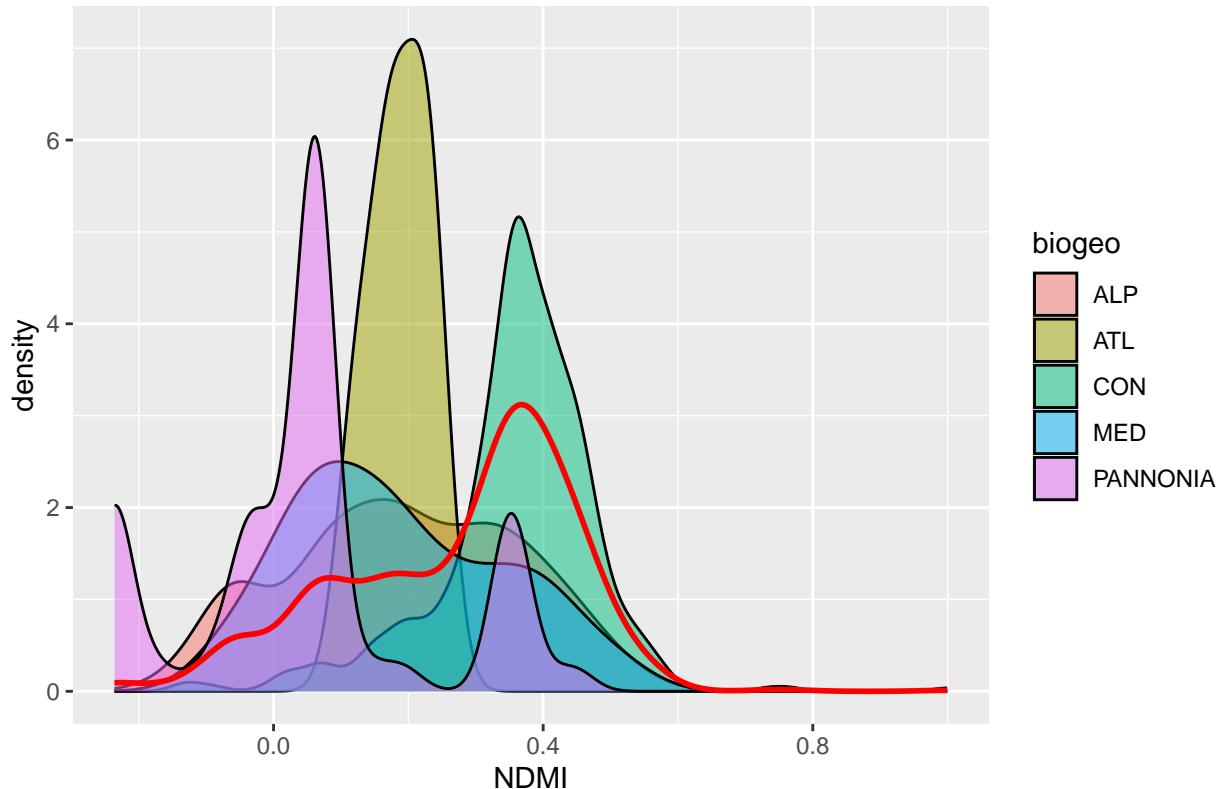
Grasslands



```
## Warning: Removed 23363 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 23363 rows containing non-finite outside the scale range
## ('stat_density()').
```

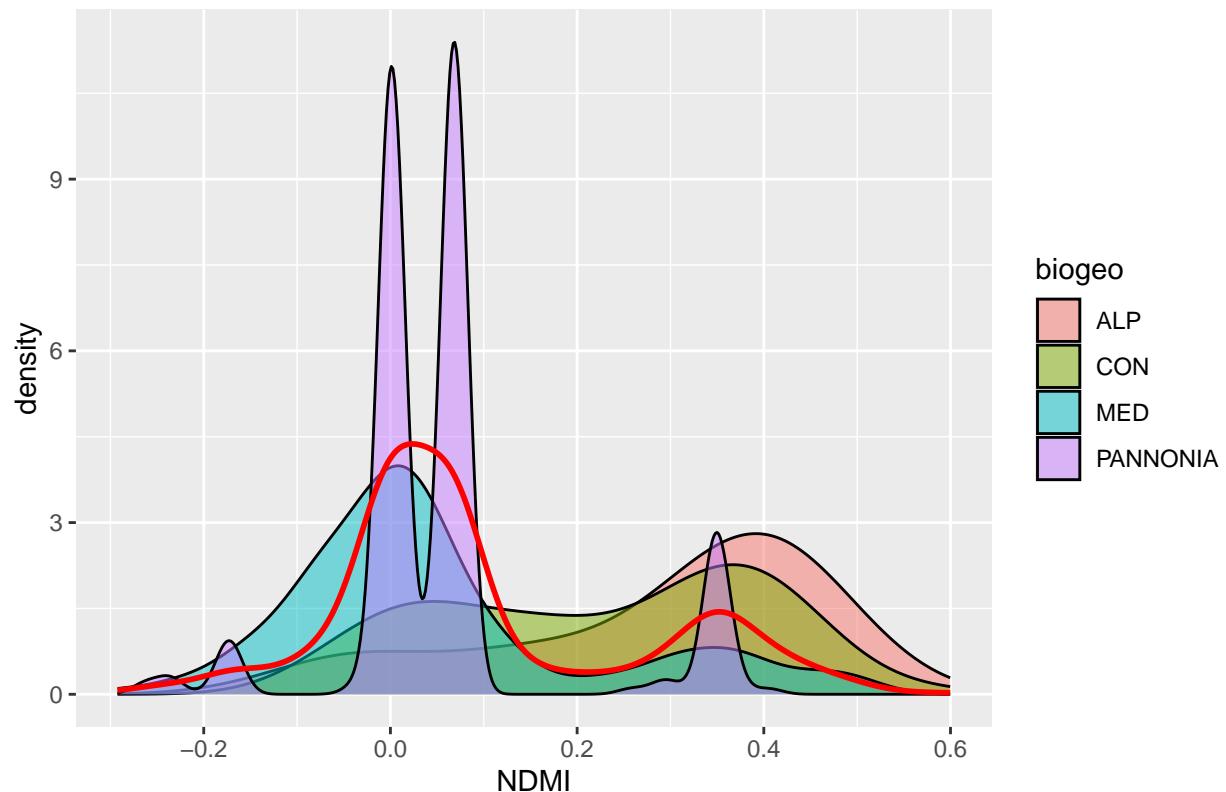
Heathlands, scrub and tundra



```
## Warning: Removed 7465 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 7465 rows containing non-finite outside the scale range
## ('stat_density()').
```

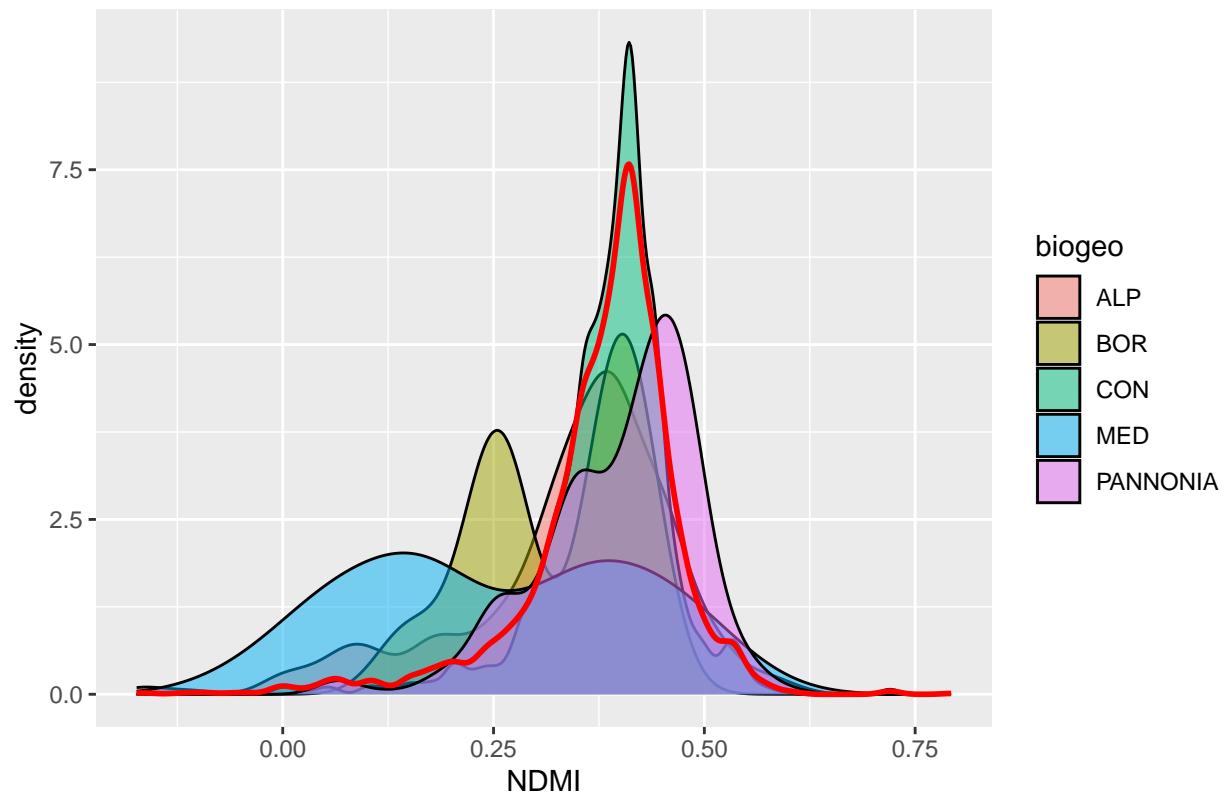
Vegetated man-made habitats



```
## Warning: Removed 21413 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 21413 rows containing non-finite outside the scale range
## ('stat_density()').
```

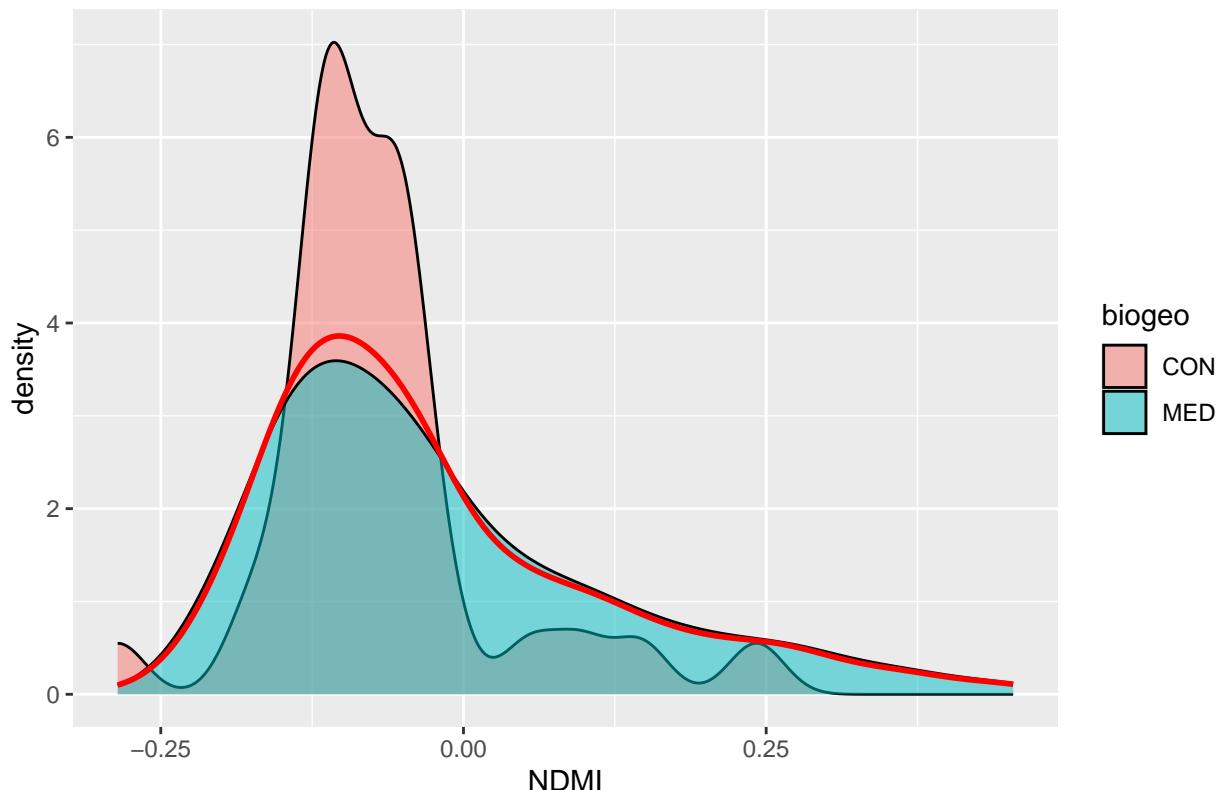
Forests and other wooded land



```
## Warning: Removed 31477 rows containing non-finite outside the scale range
## ('stat_density()').
```

```
## Warning: Removed 31477 rows containing non-finite outside the scale range
## ('stat_density()').
```

Coastal habitats



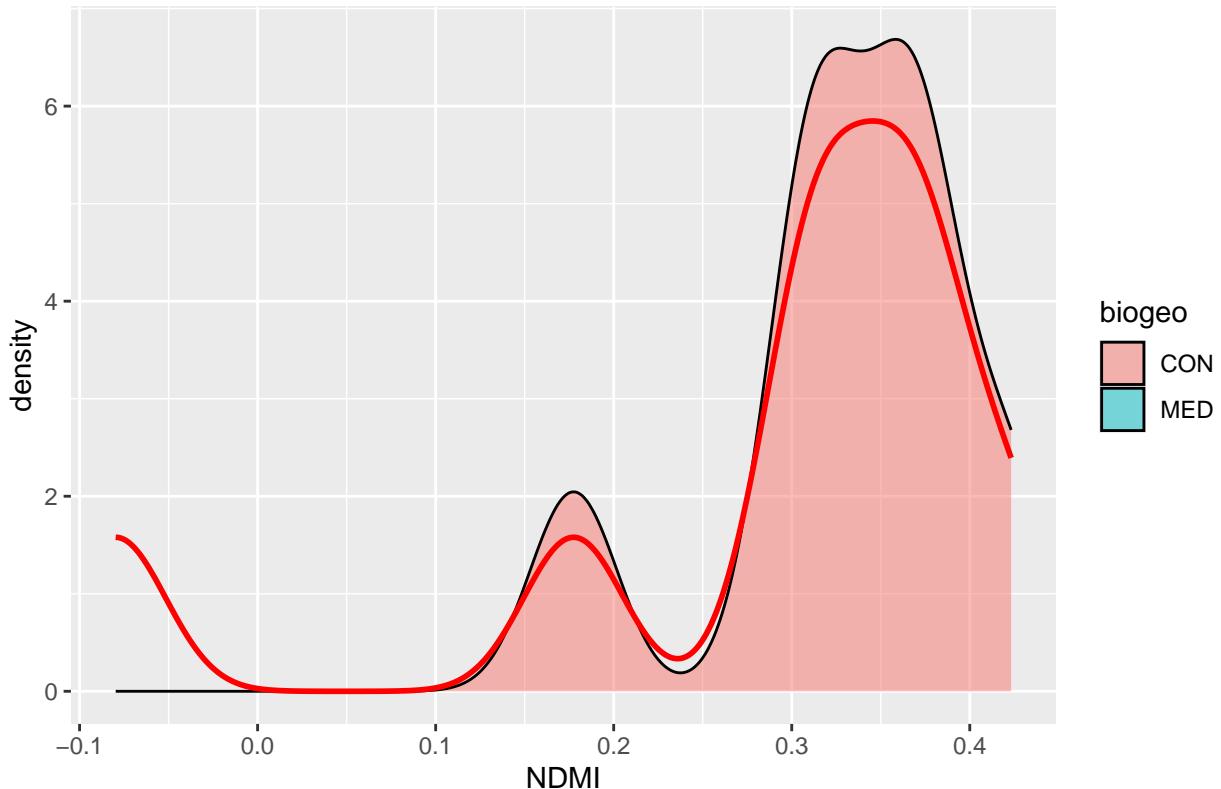
```
## Warning: Removed 12473 rows containing non-finite outside the scale range
## ('stat_density()').

## Warning: Groups with fewer than two data points have been dropped.

## Warning: Removed 12473 rows containing non-finite outside the scale range
## ('stat_density()').

## Warning in max(ids, na.rm = TRUE): ningun argumento finito para max; retornando
## -Inf
```

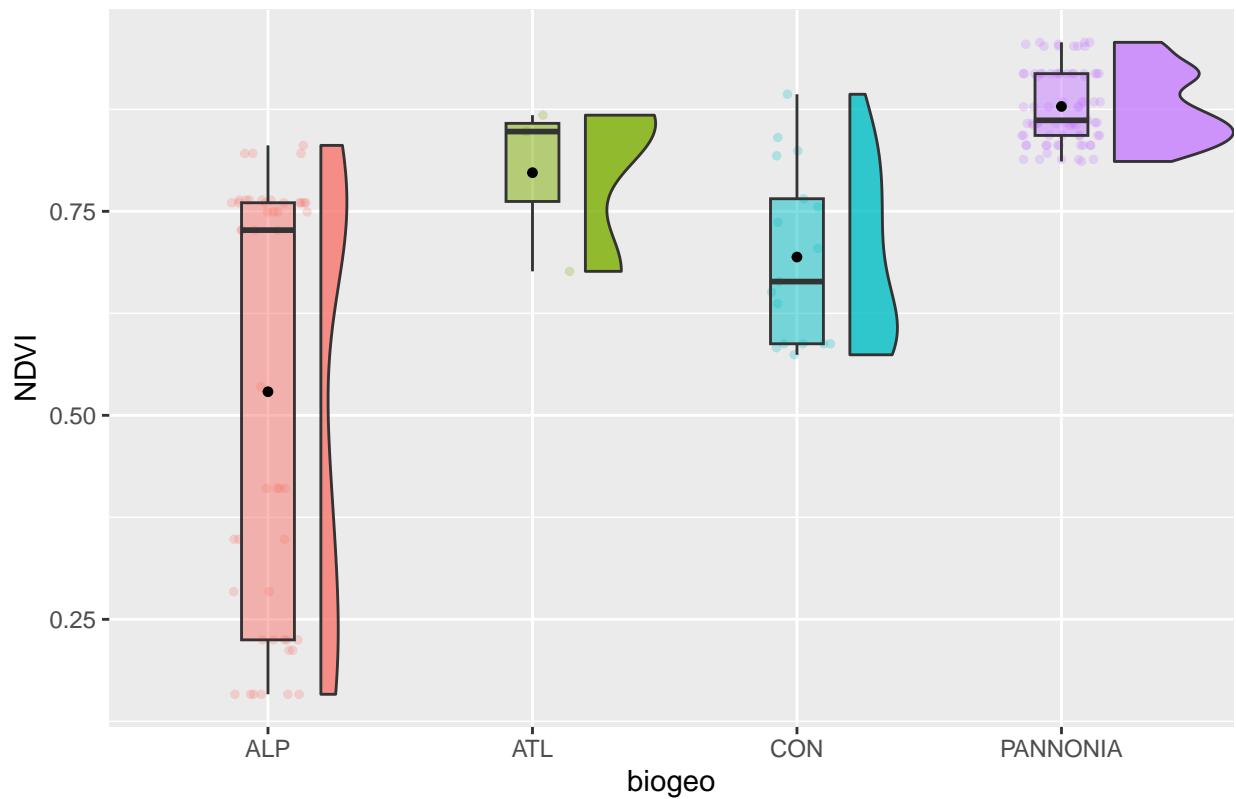
Marine habitats



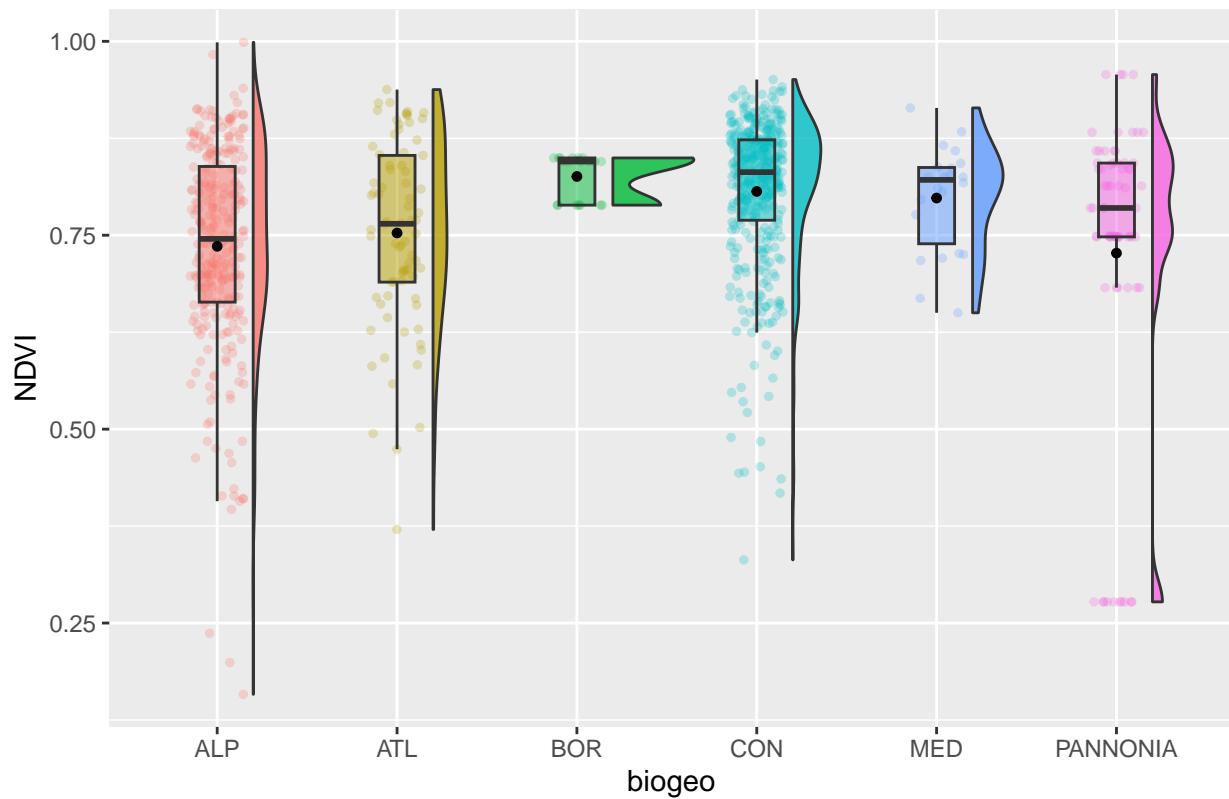
Violin + boxplot + points for each bioregion

```
eunis_levels %>%
  set_names() %>% # Set names to access plots by EUNISa_1_descr
  map(~
    ggplot(data = filter(db_resurv_RS_short %>% filter(RS_data == T),
                        EUNISa_1_descr == .x),
           aes(x = biogeo, y = NDVI, fill = biogeo)) +
    geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
    geom_point(aes(y = NDVI, color = biogeo),
               position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
    geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
    stat_summary(fun.y=mean, geom="point", shape = 20, size=2) +
    labs(y = "NDVI", x = "biogeo") +
    guides(fill = FALSE, color = FALSE) +
    labs(title = .x)
  ) %>%
  walk(print)
```

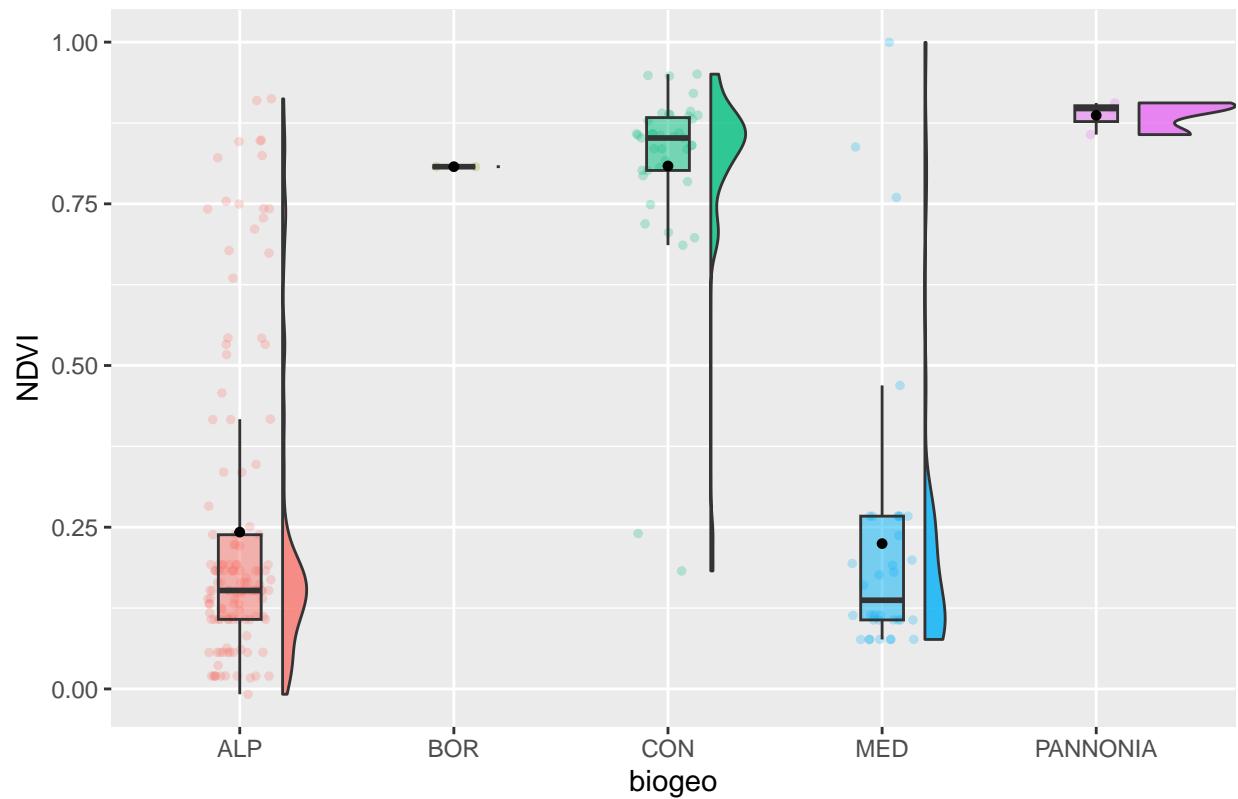
Inland waters



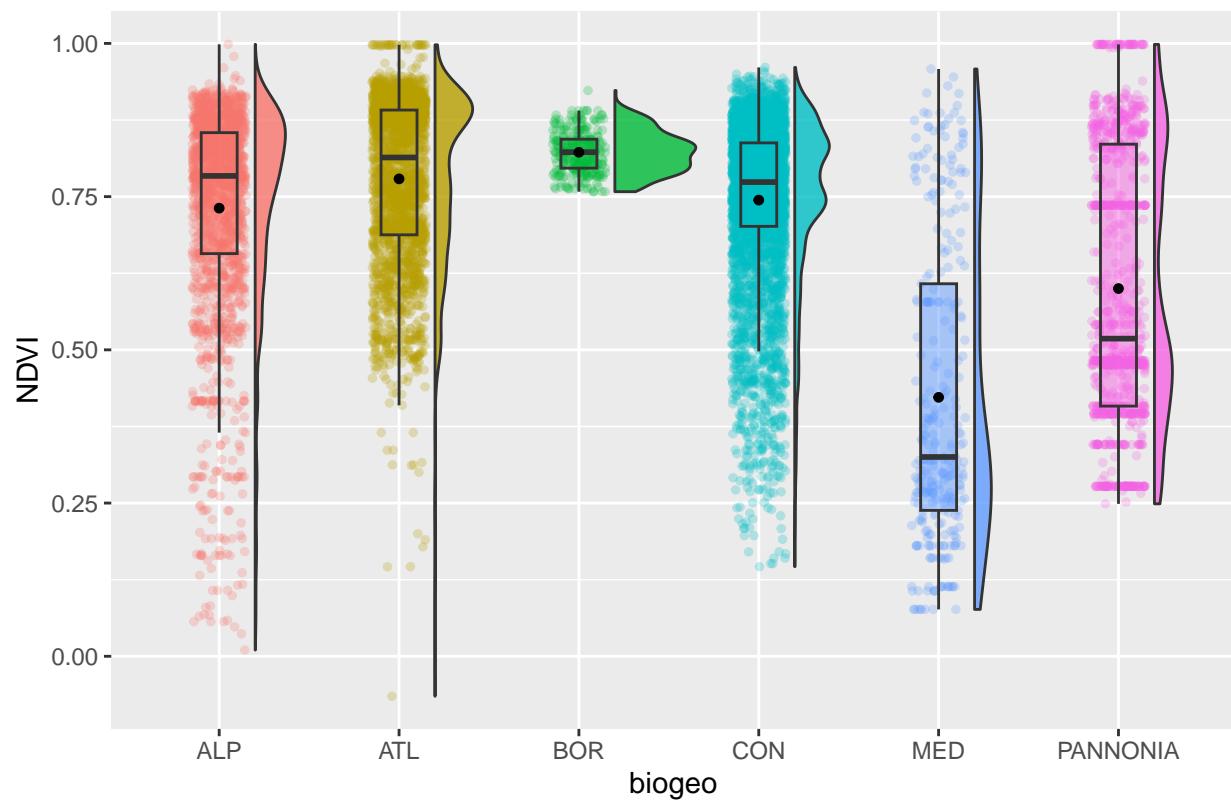
Wetlands



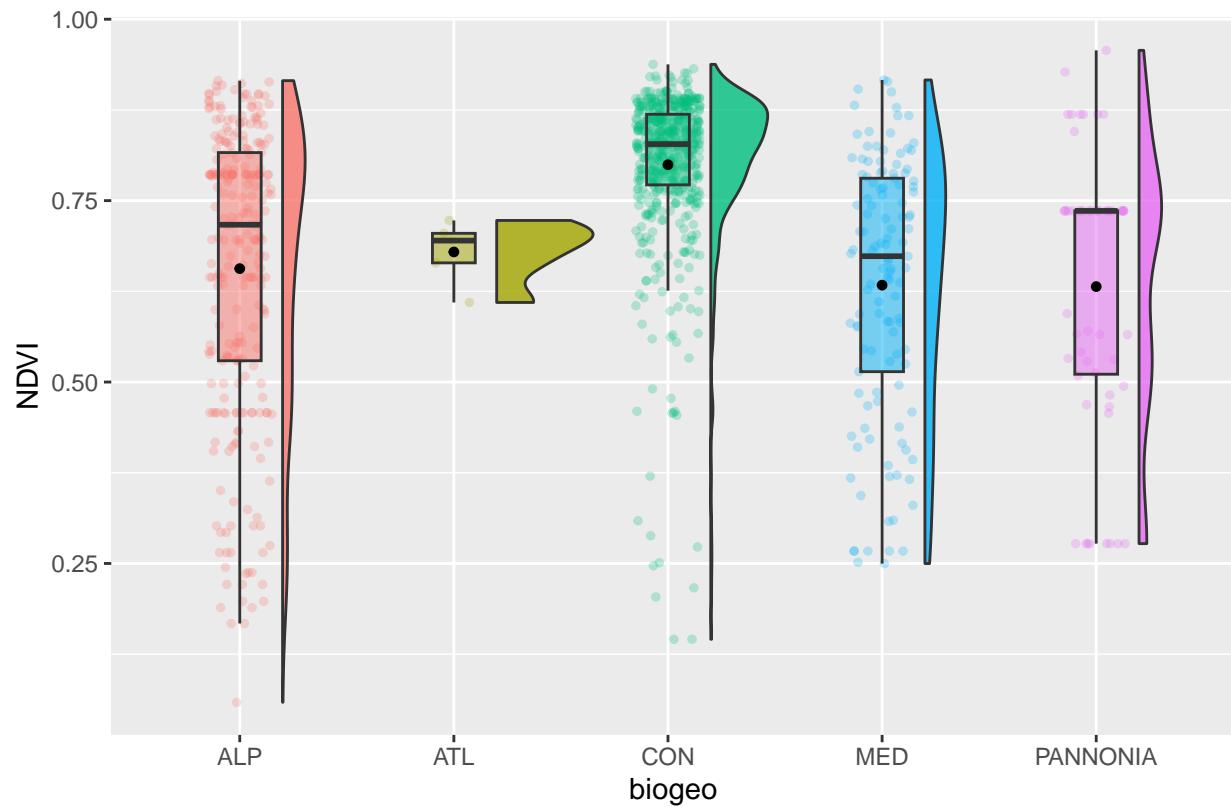
Inland habitats with no or little soil



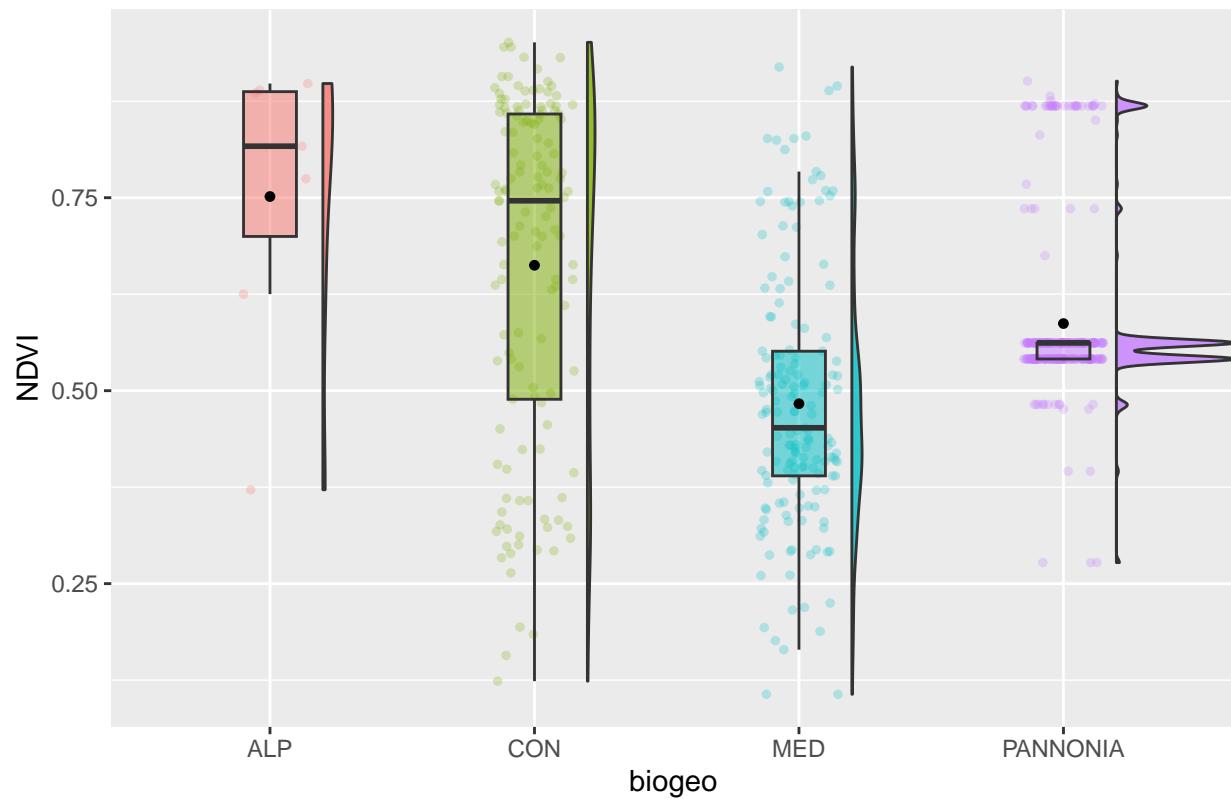
Grasslands



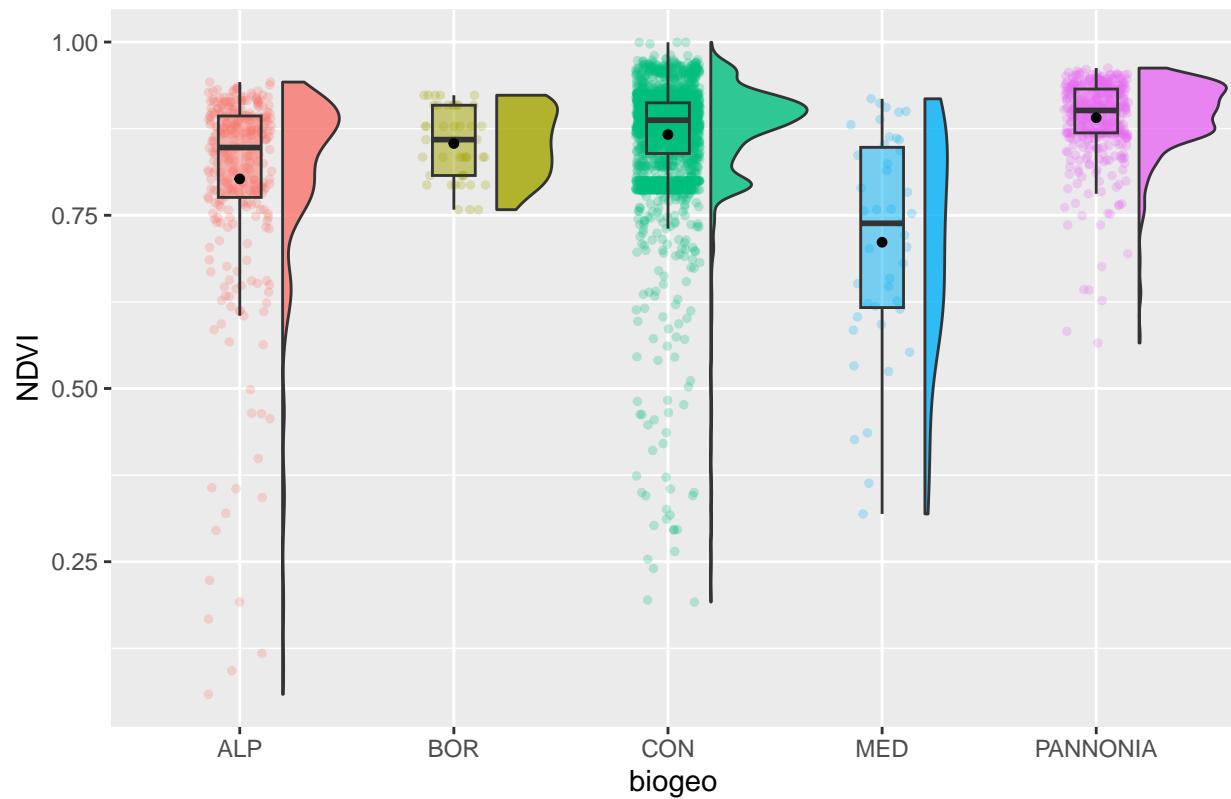
Heathlands, scrub and tundra



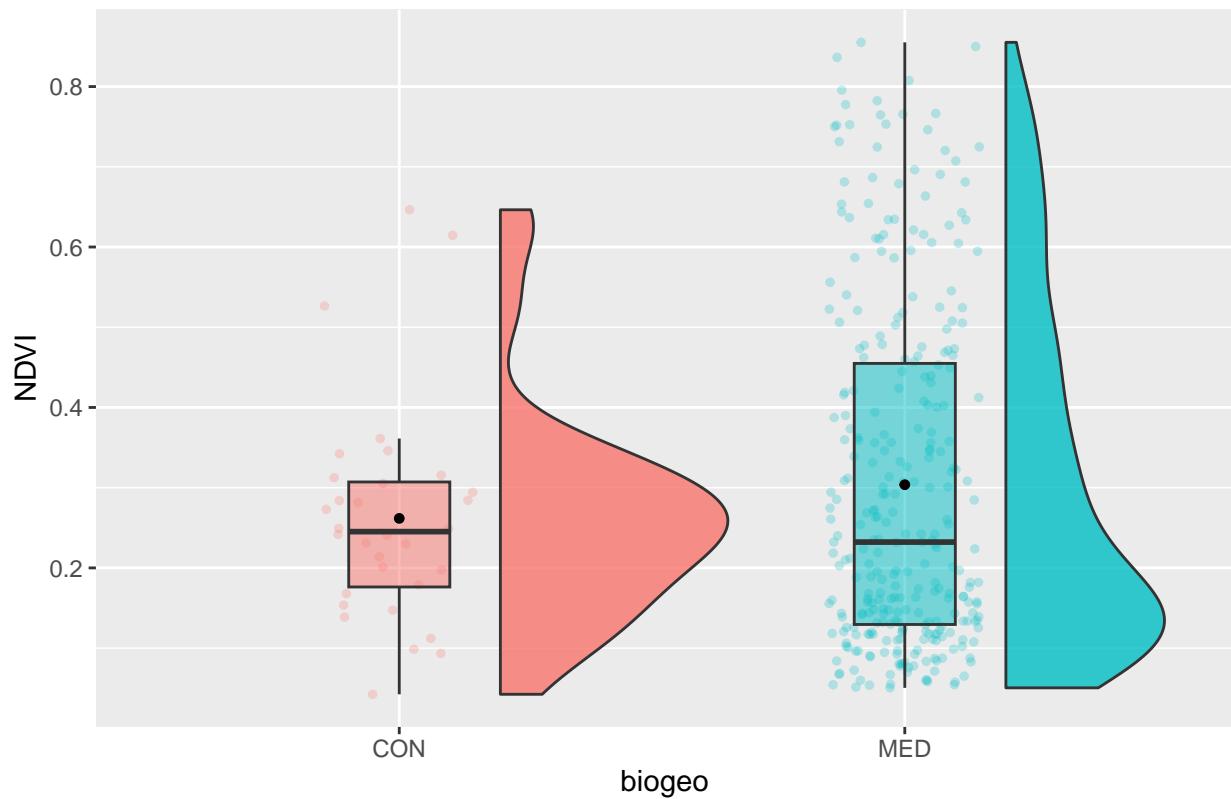
Vegetated man-made habitats



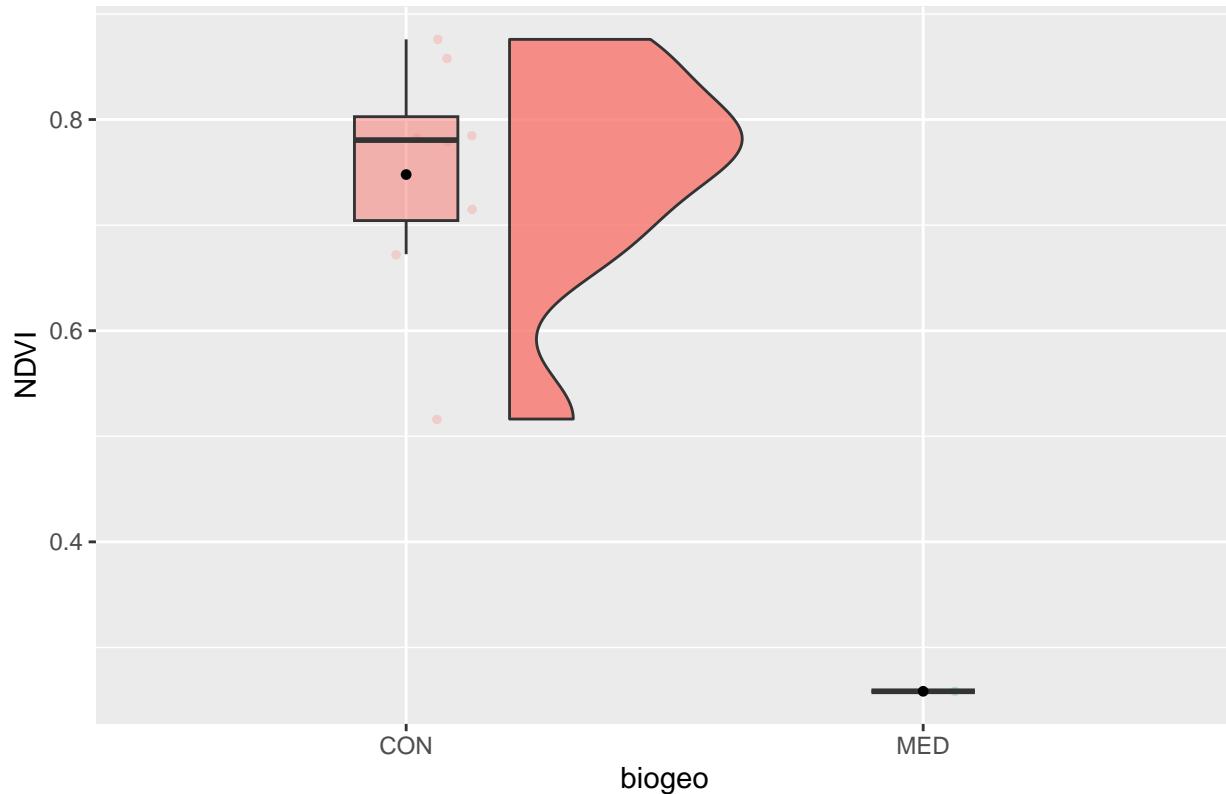
Forests and other wooded land



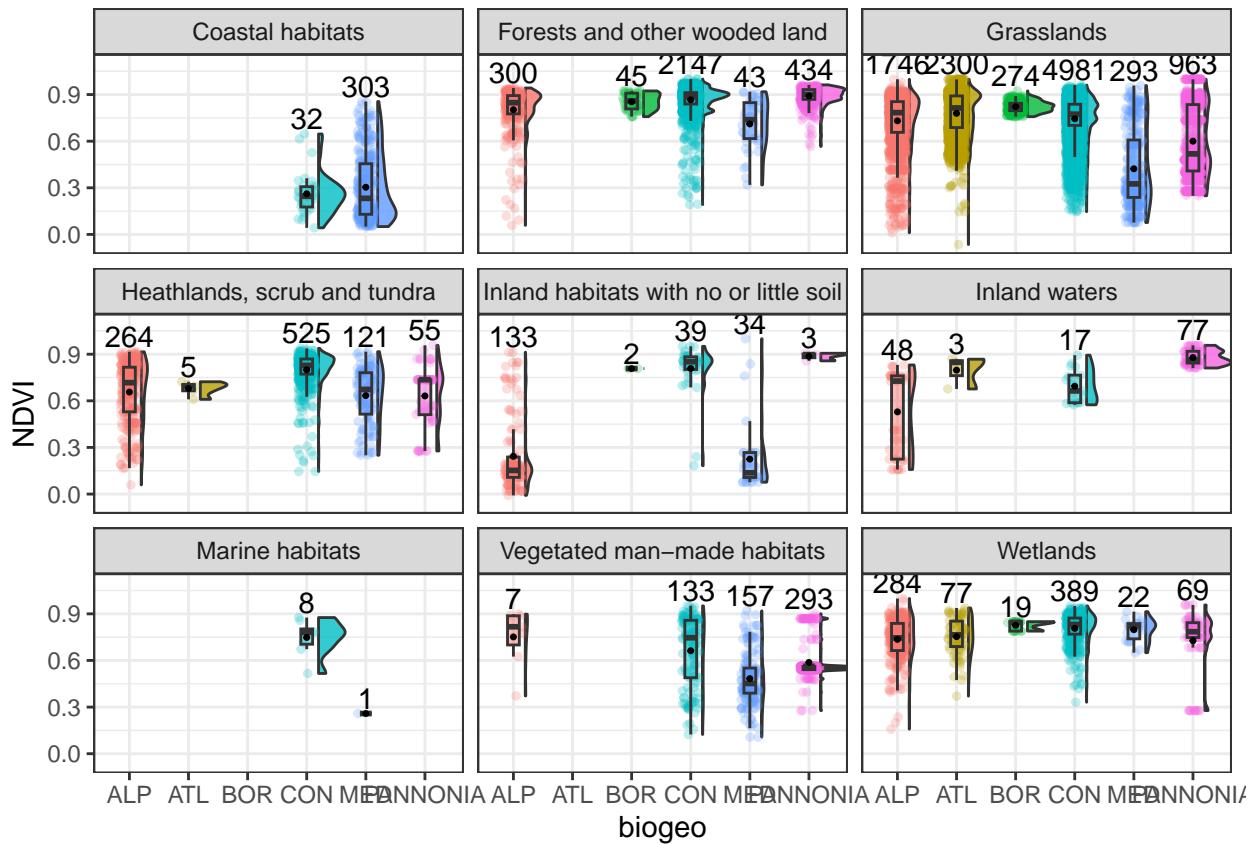
Coastal habitats



Marine habitats



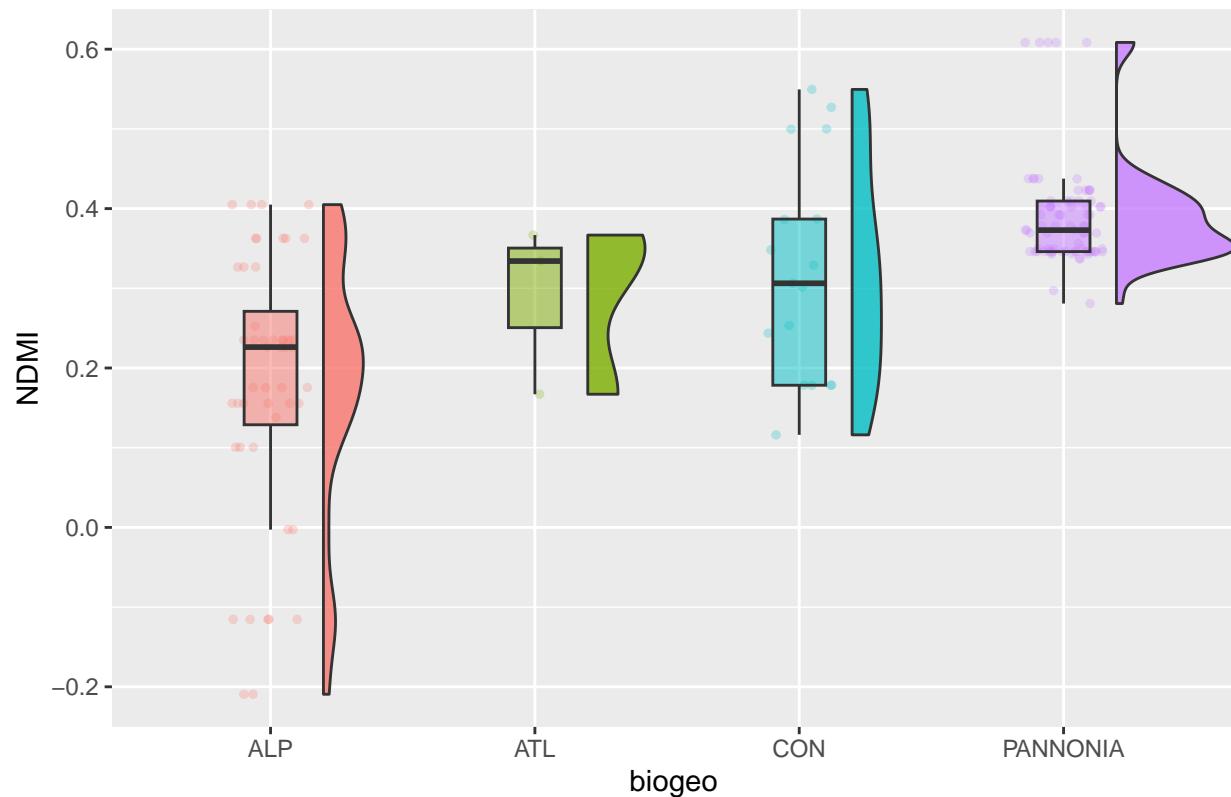
```
ggplot(data = filter(db_resurv_RS_short %>%
                     filter(RS_data == T & !is.na(EUNISa_1_descr))),
       aes(x = biogeo, y = NDVI, fill = biogeo)) +
  facet_wrap(~ EUNISa_1_descr) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = NDVI, color = biogeo),
             position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 1) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                 label = length(x)),
              geom = "text", aes(label = ..label..), vjust = 0.5) +
  labs(y = "NDVI", x = "biogeo") +
  guides(fill = FALSE, color = FALSE) +
  theme_bw()
```



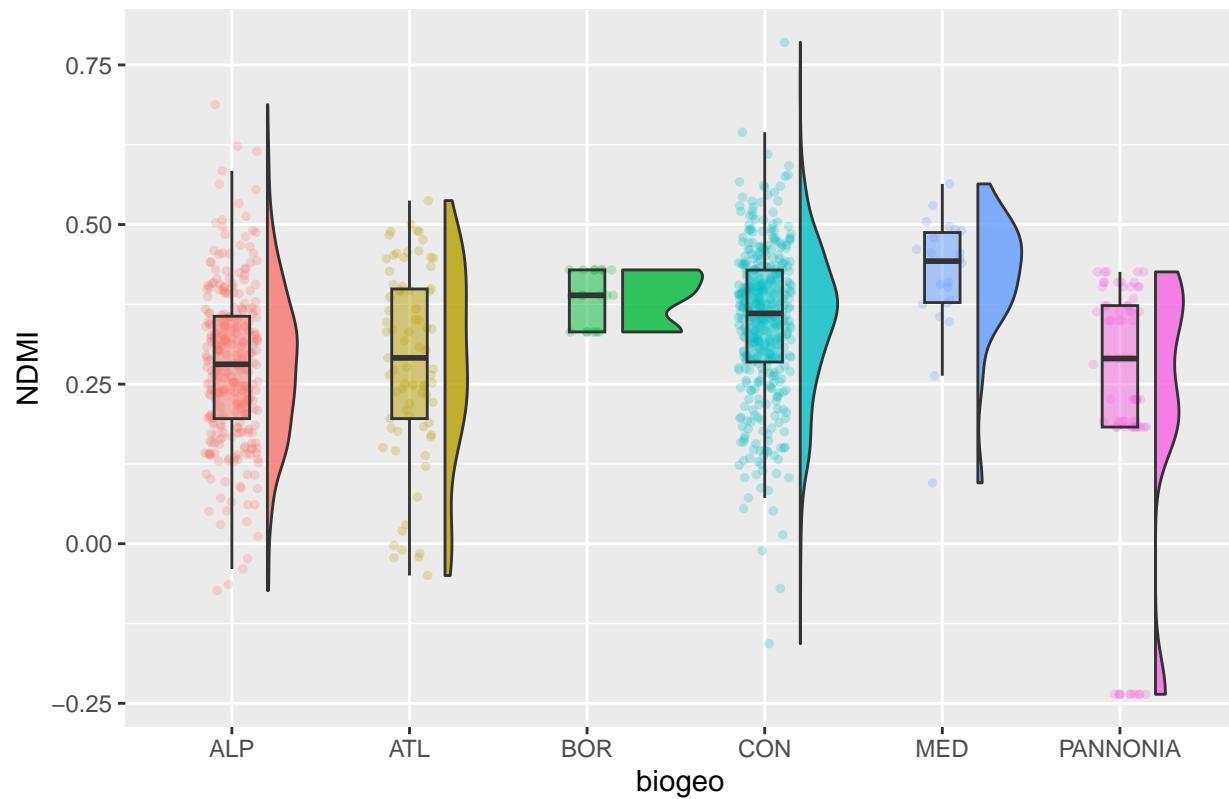
```
ggsave(
  here("output", "figures", "NDVI_biogeo.tif"),
  width = 29.7, height = 21, units = "cm", dpi = 300)
```

```
eunis_levels %>%
  set_names() %>% # Set names to access plots by EUNISa_1_descr
  map(~
    ggplot(data = filter(db_resurv_RS_short %>% filter(RS_data == T),
                        EUNISa_1_descr == .x),
           aes(x = biogeo, y = NDMI, fill = biogeo)) +
      geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
      geom_point(aes(y = NDMI, color = biogeo),
                 position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
      geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
      labs(y = "NDMI", x = "biogeo") +
      guides(fill = FALSE, color = FALSE) +
      labs(title = .x)
  ) %>%
  walk(print)
```

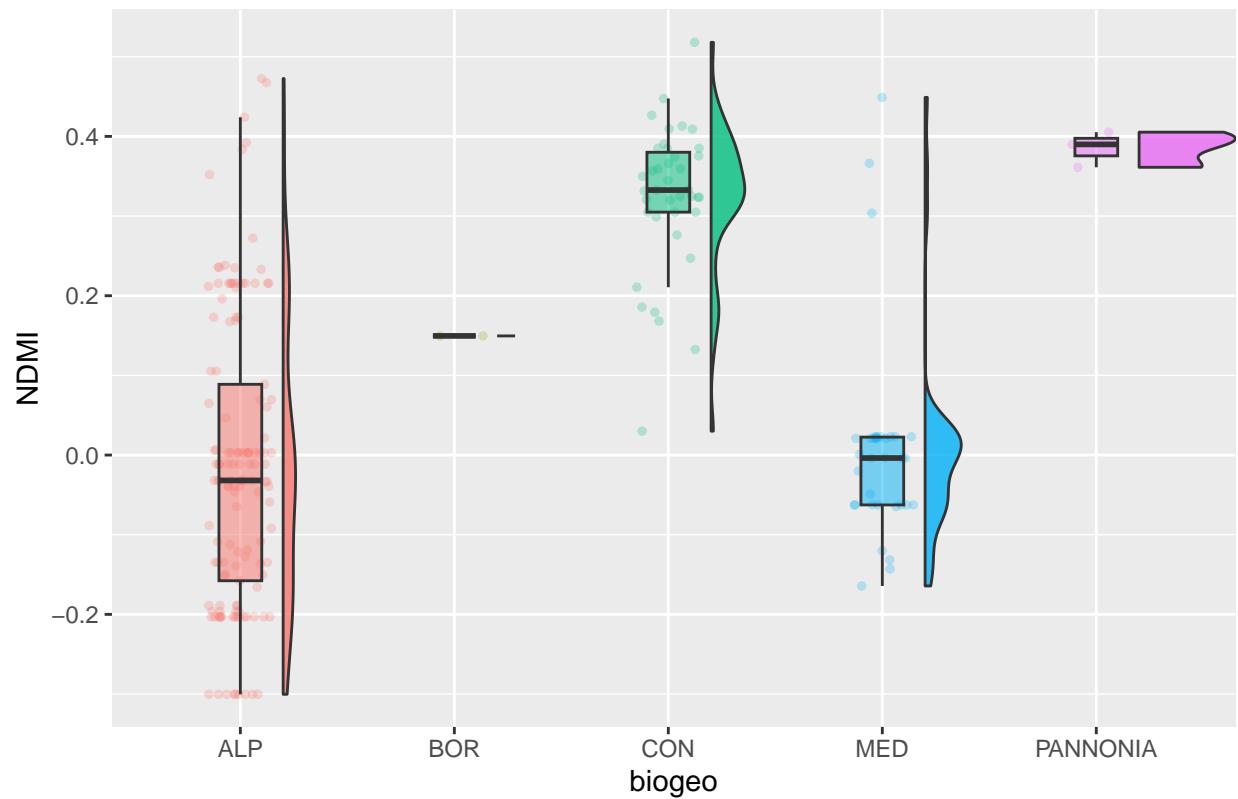
Inland waters



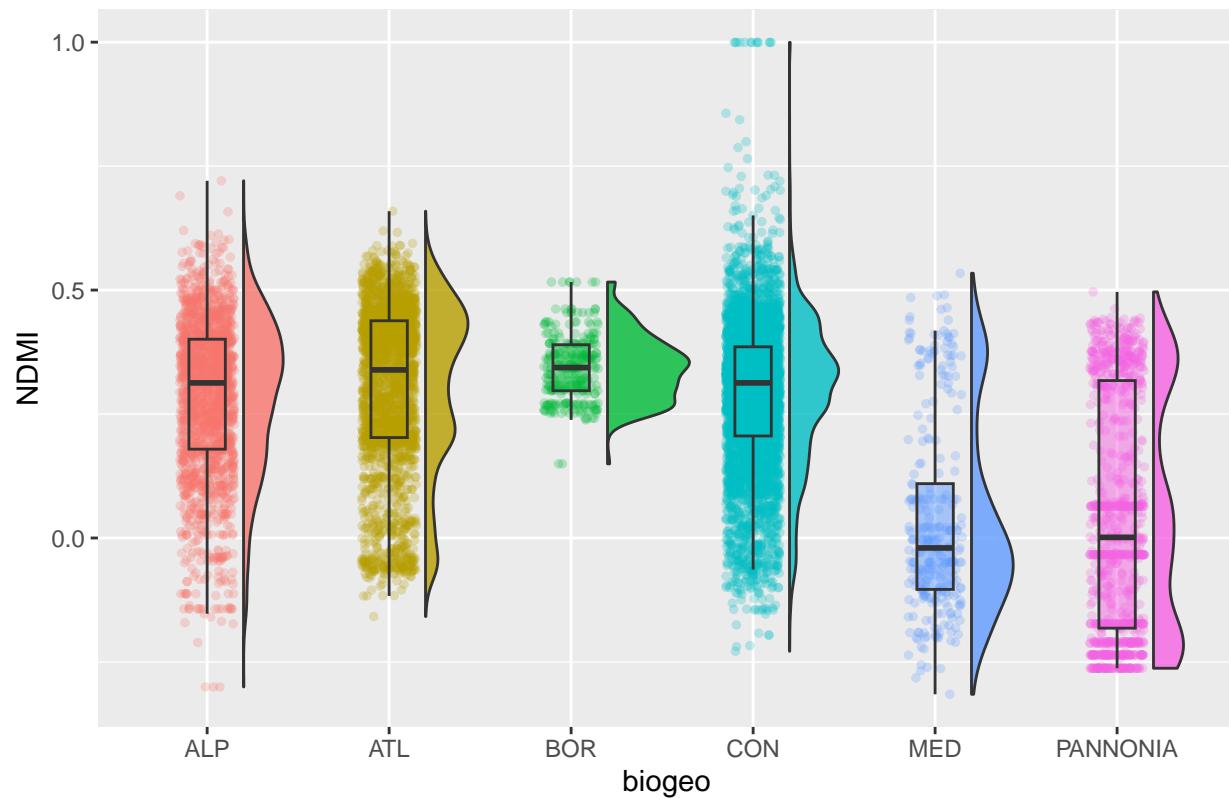
Wetlands



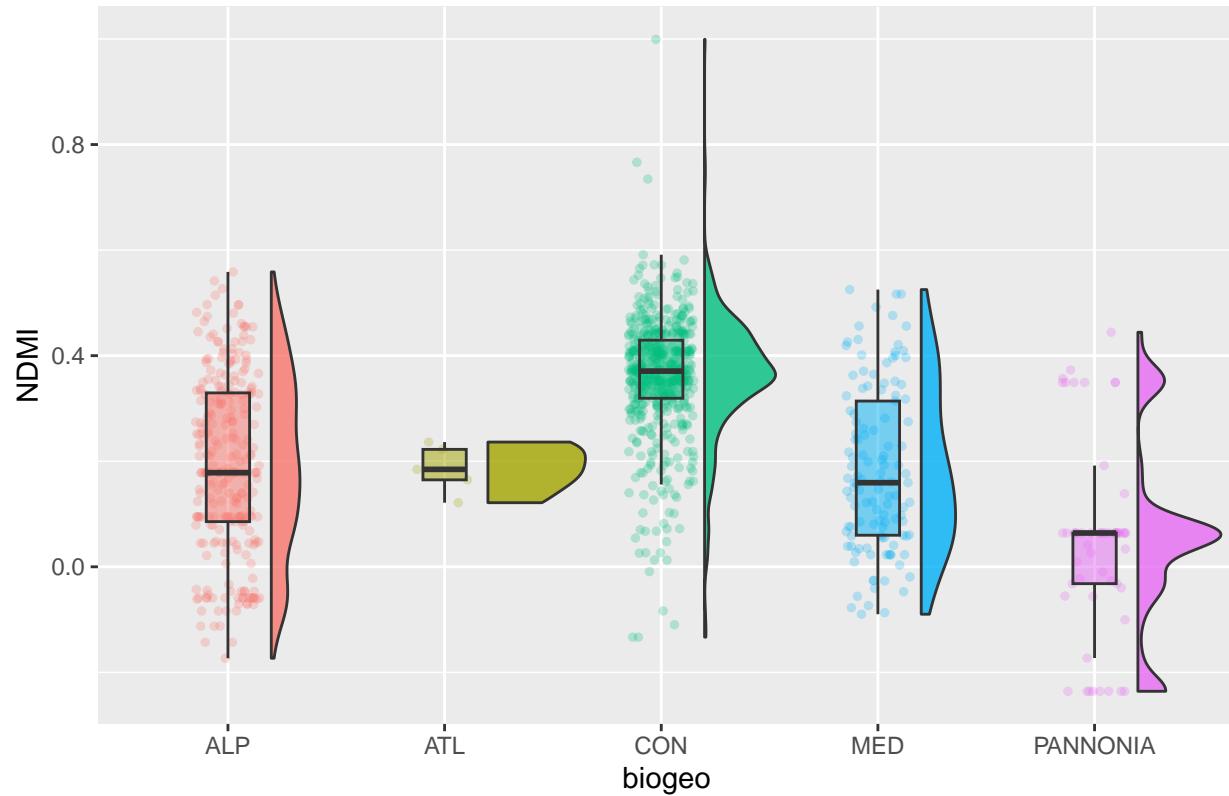
Inland habitats with no or little soil



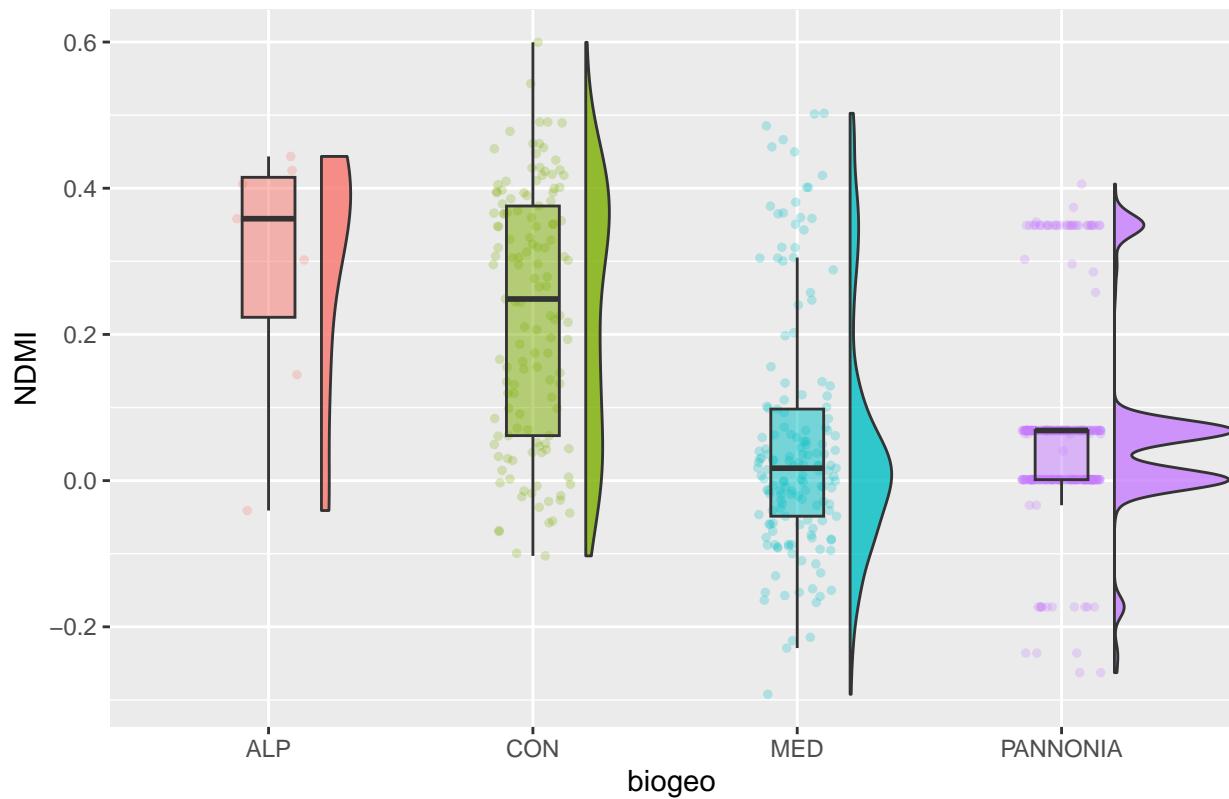
Grasslands



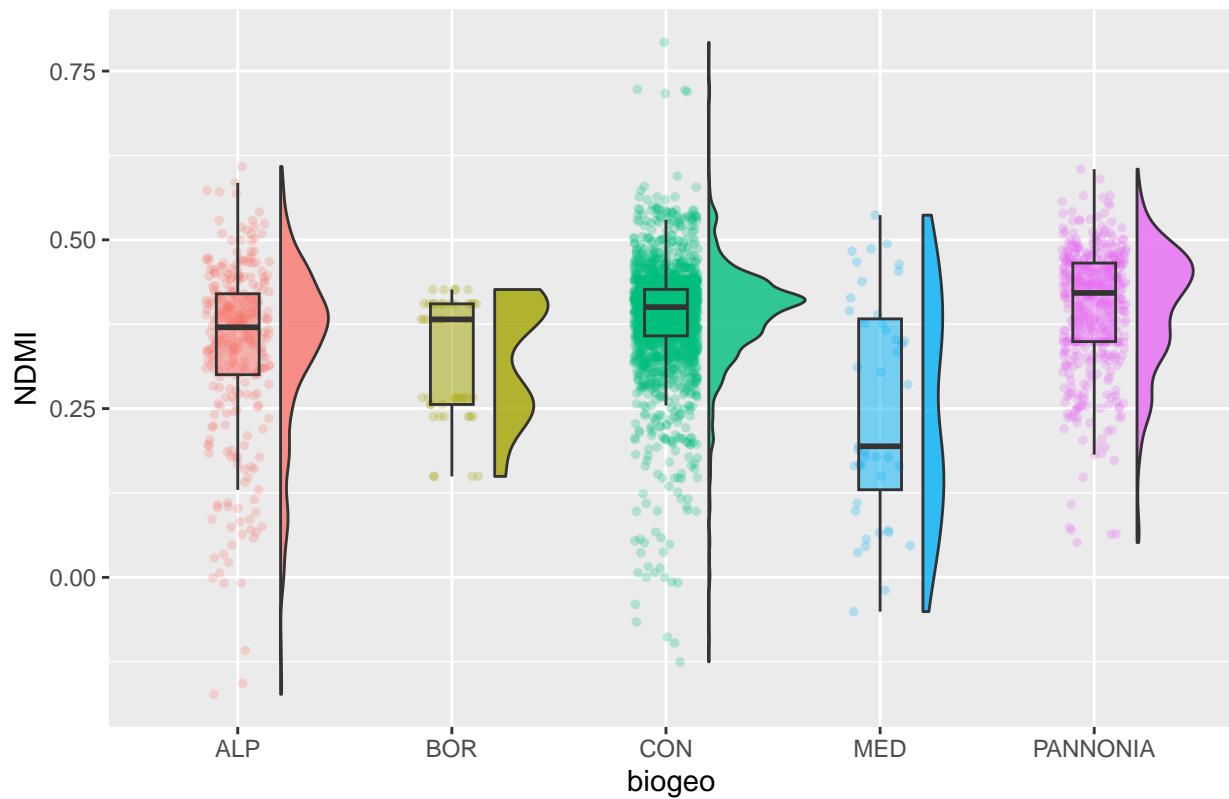
Heathlands, scrub and tundra



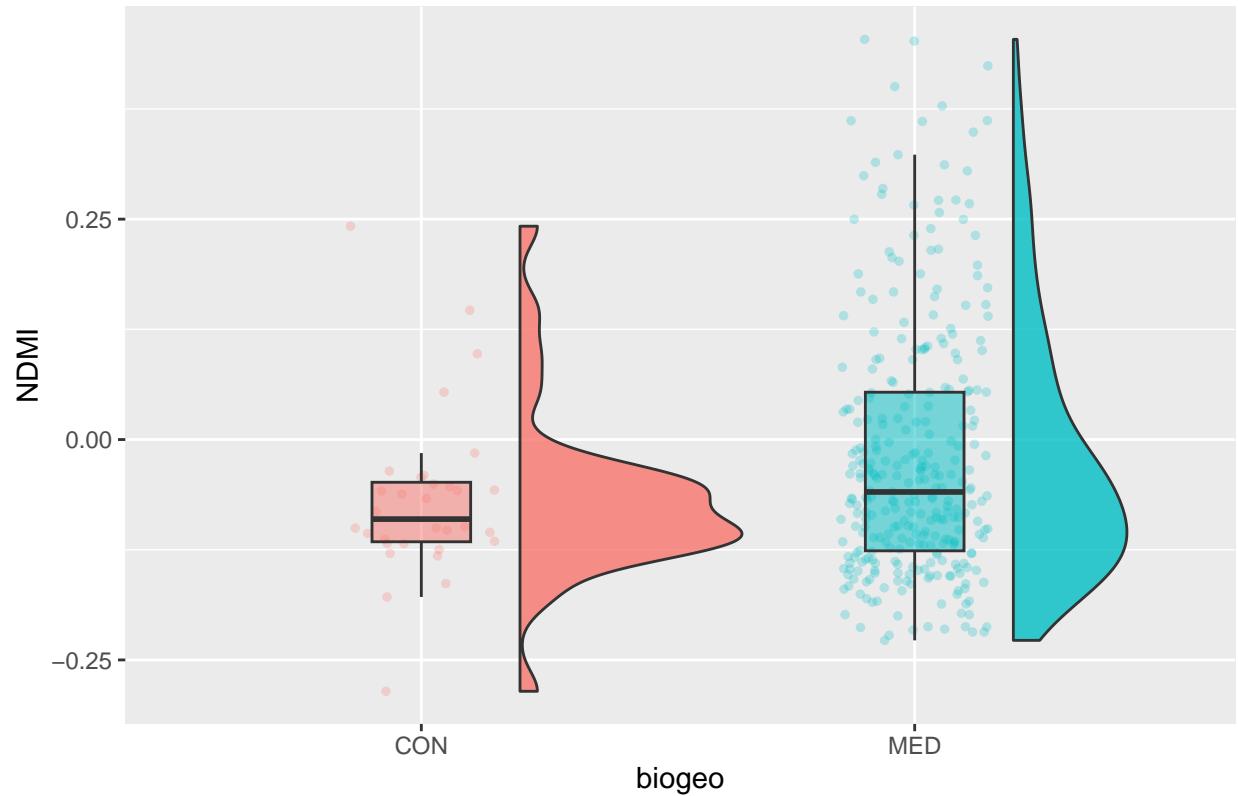
Vegetated man-made habitats



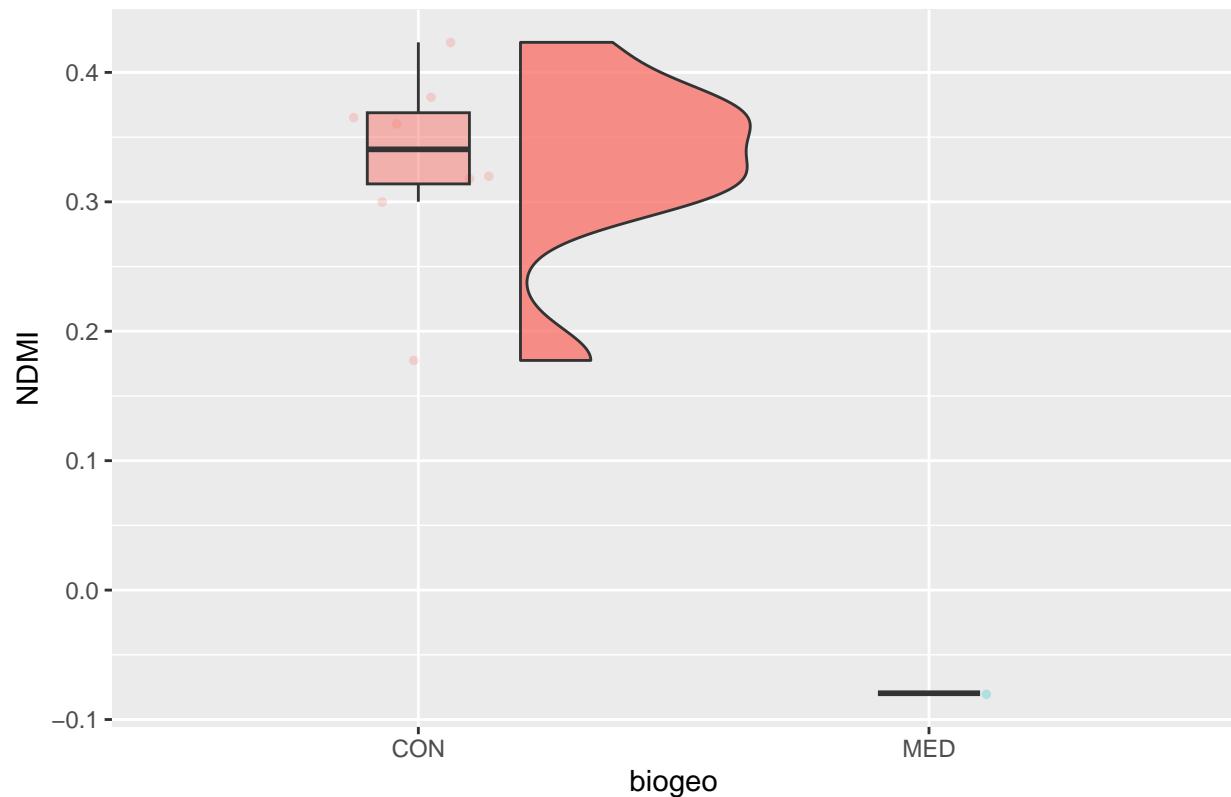
Forests and other wooded land



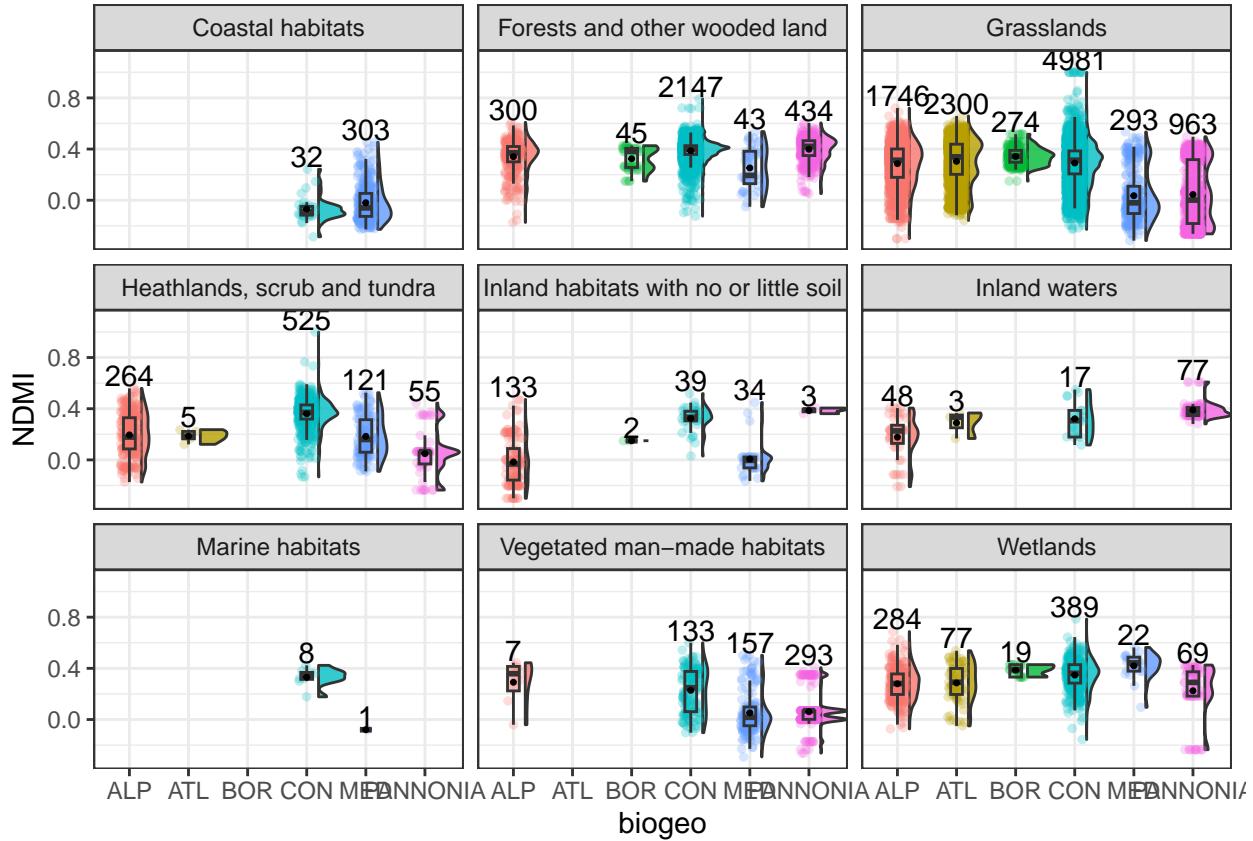
Coastal habitats



Marine habitats



```
ggplot(data = filter(db_resurv_RS_short %>%
                     filter(RS_data == T & !is.na(EUNISa_1_descr))),
       aes(x = biogeo, y = NDMI, fill = biogeo)) +
  facet_wrap(~ EUNISa_1_descr) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = NDMI, color = biogeo),
             position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 1) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                label = length(x)),
              geom = "text", aes(label = ..label..), vjust = 0.5) +
  labs(y = "NDMI", x = "biogeo") +
  guides(fill = FALSE, color = FALSE) +
  theme_bw()
```



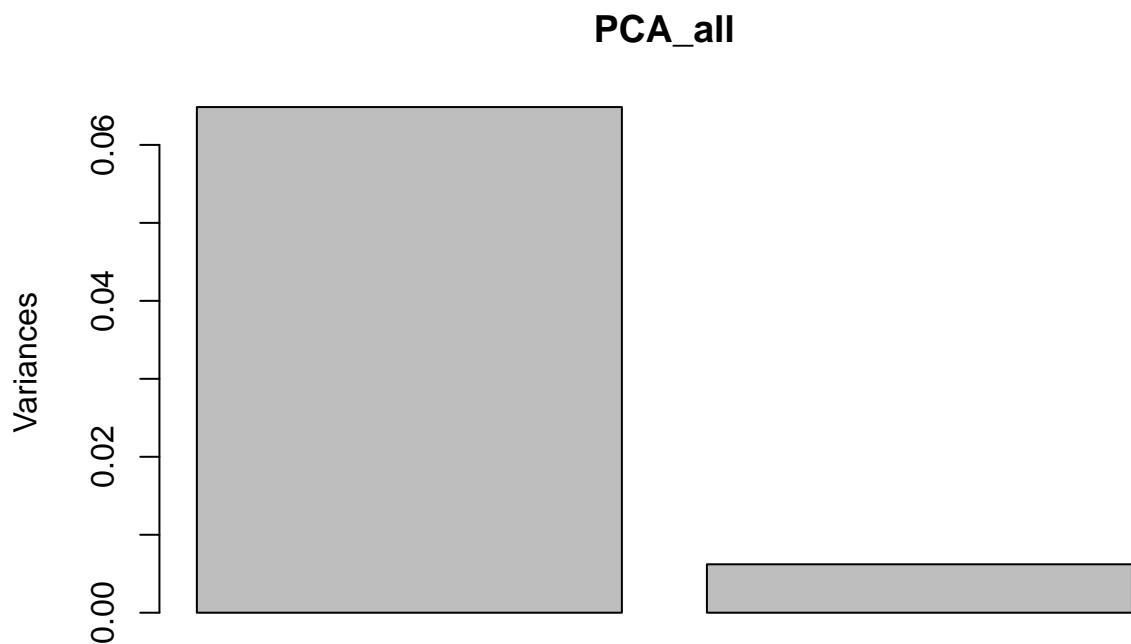
```
ggsave(
  here("output", "figures", "NDMI_biogeo.tiff"),
  width = 29.7, height = 21, units = "cm", dpi = 300)
```

PCA

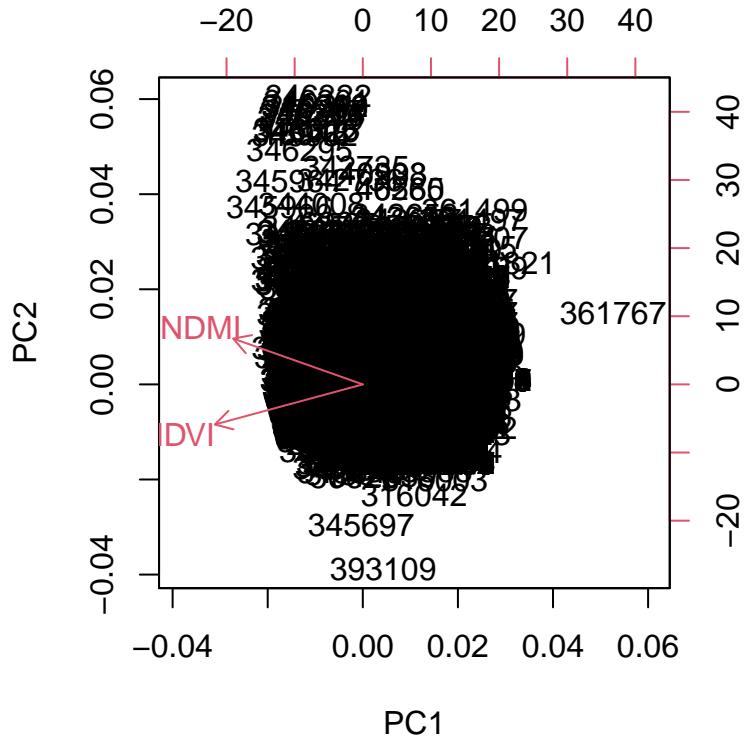
```
PCA_all <- prcomp(~ NDVI + NDMI, data = db_resurv_RS_short)
```

```
PCA_ALP <- prcomp(~ NDVI + NDMI,
                     data = db_resurv_RS_short %>% filter(biogeo == "ALP"))
PCA_CON <- prcomp(~ NDVI + NDMI,
                     data = db_resurv_RS_short %>% filter(biogeo == "CON"))
PCA_ATL <- prcomp(~ NDVI + NDMI,
                     data = db_resurv_RS_short %>% filter(biogeo == "ATL"))
PCA_PANNONIA <- prcomp(~ NDVI + NDMI,
                         data = db_resurv_RS_short %>% filter(biogeo == "PANNONIA"))
PCA_MED <- prcomp(~ NDVI + NDMI,
                     data = db_resurv_RS_short %>% filter(biogeo == "MED"))
PCA_BOR <- prcomp(~ NDVI + NDMI,
                     data = db_resurv_RS_short %>% filter(biogeo == "BOR"))
```

```
plot(PCA_all)
```



```
biplot(PCA_all)
```

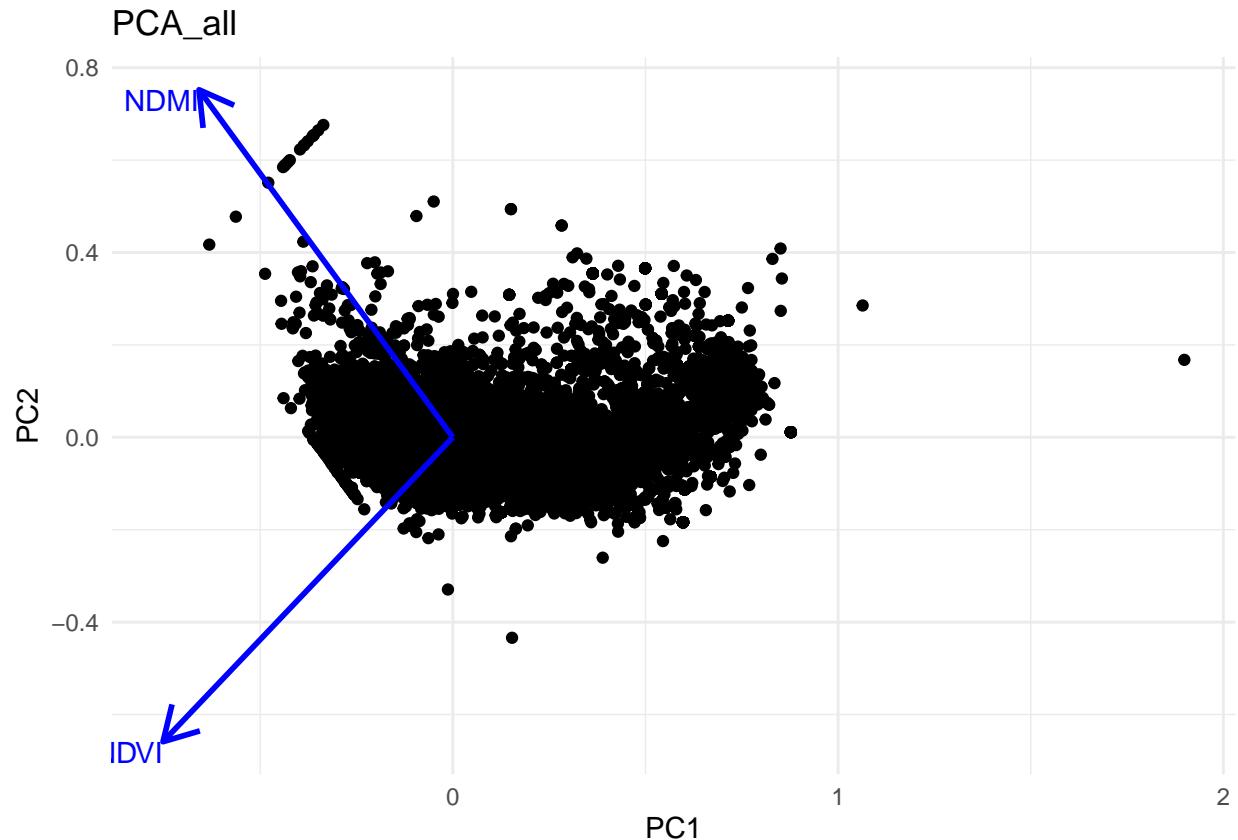


```
# Define a function to generate PCA biplots
generate_pca_biplot <- function(pca_result, title) {
  ggplot(as.data.frame(pca_result$x), aes(x = PC1, y = PC2)) +
    geom_point() +
    geom_segment(data = as.data.frame(pca_result$rotation),
                 aes(x = 0, y = 0, xend = PC1, yend = PC2),
                 arrow = arrow(length = unit(0.2, "inches")),
                 color = "blue", size = 1) +
    geom_text(data = as.data.frame(pca_result$rotation),
              aes(x = PC1, y = PC2,
                  label = rownames(as.data.frame(pca_result$rotation))),
              vjust = 1, hjust = 1, color = "blue") +
    labs(title = title, x = "PC1", y = "PC2") +
    theme_minimal()
}
```

```
# Function to generate variance explained bar plot
generate_variance_plot <- function(pca_result, title) {
  ggplot(
    data.frame(PC = names(summary(pca_result)$importance[2, ]),
               Variance = summary(pca_result)$importance[2, ]),
    aes(x = PC, y = Variance)
  ) +
    geom_bar(stat = "identity") +
    labs(x = "Principal Component", y = "Variance Explained (%)", title = title) +
    theme_minimal()
```

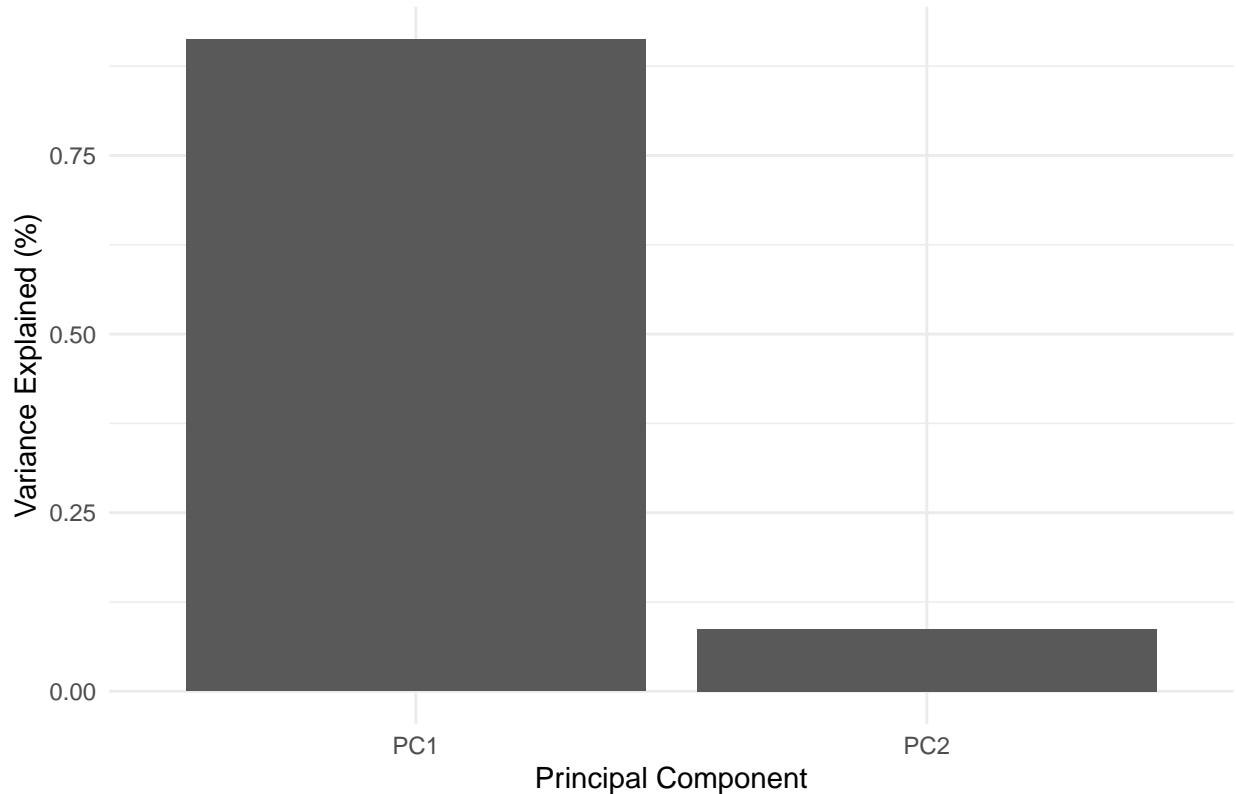
```
}
```

```
generate_pca_biplot(PCA_all, "PCA_all")
```



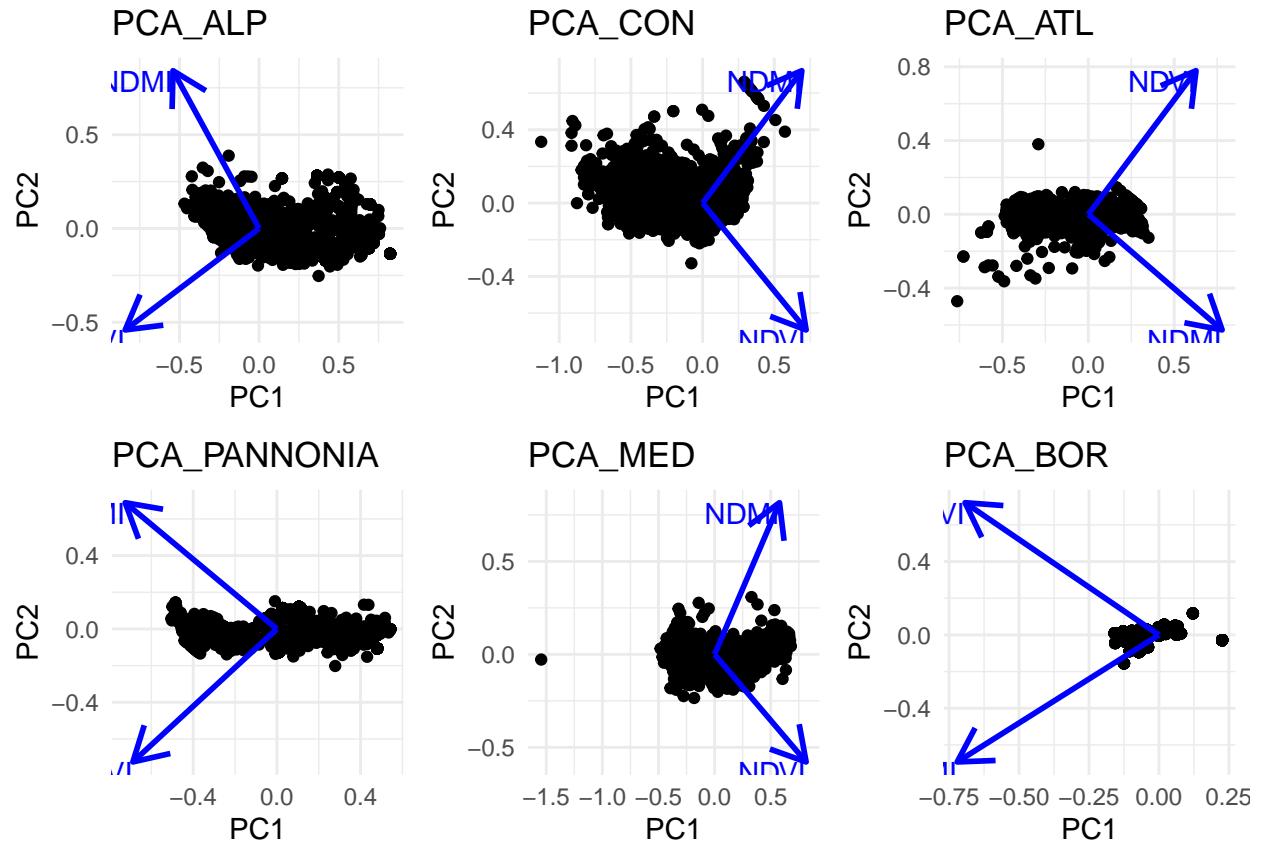
```
generate_variance_plot(PCA_all, "PCA_all")
```

PCA_all



```
pca_list <- list(PCA_ALP = PCA_ALP, PCA_CON = PCA_CON, PCA_ATL = PCA_ATL,
                  PCA_PANNONIA = PCA_PANNONIA, PCA_MED = PCA_MED,
                  PCA_BOR = PCA_BOR)
```

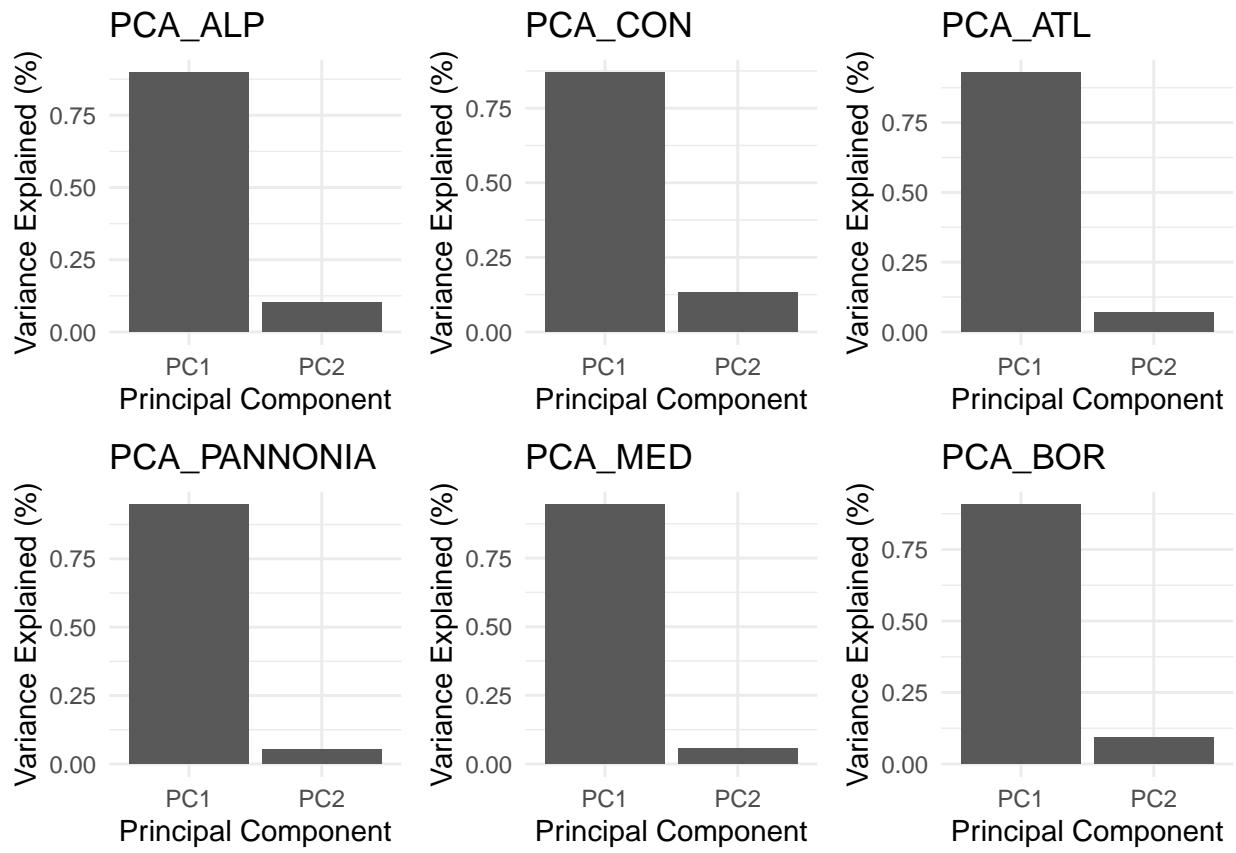
```
pca_biplots <- lapply(names(pca_list),
                      function(pca_name) {
                        generate_pca_biplot(pca_list[[pca_name]],
                                             pca_name)
                      })
grid.arrange(grobs = pca_biplots, ncol = 3)
```



```

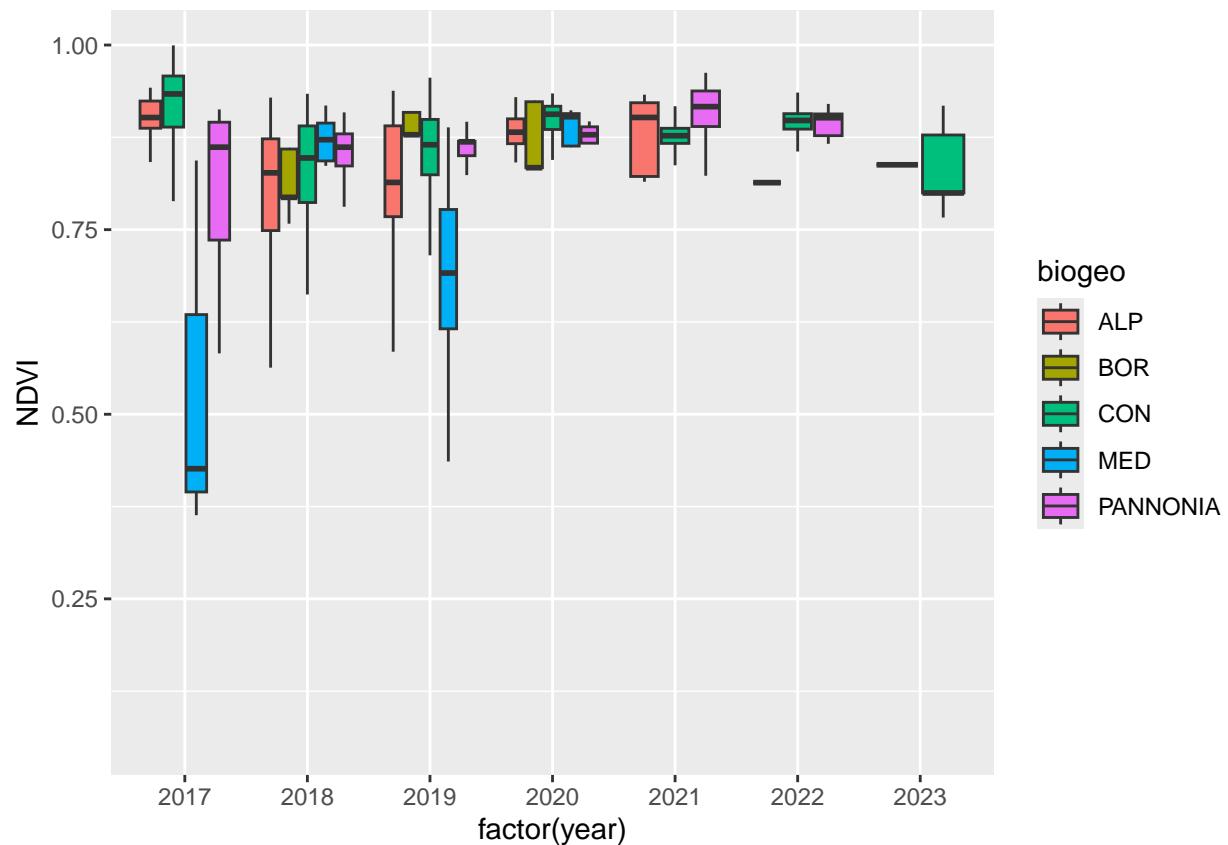
variance_plots <- lapply(names(pca_list), function(pca_name) {
  generate_variance_plot(pca_list[[pca_name]], pca_name)
})
grid.arrange(grobs = variance_plots, ncol = 3)

```

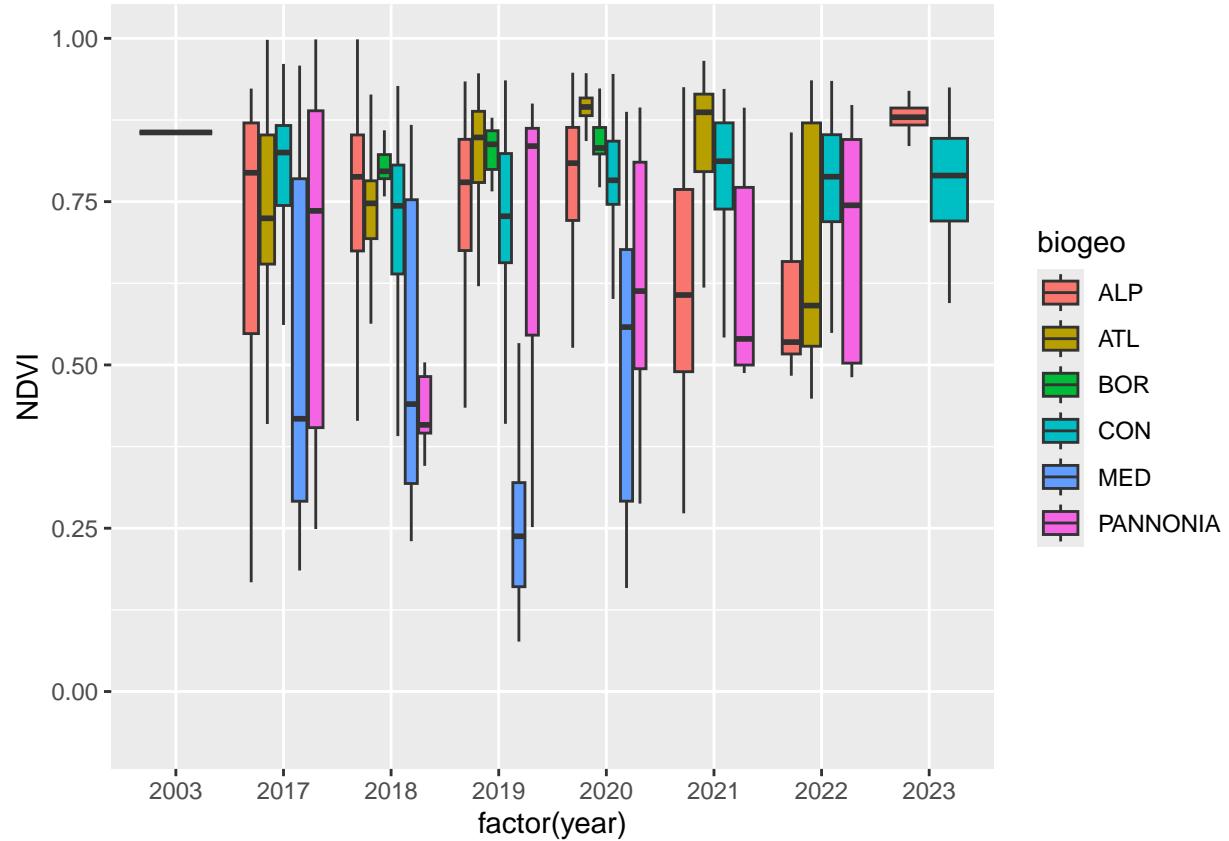


Different years: Forests and grasslands

```
db_resurv_RS_short %>% filter(EUNISa_1 == "T" & RS_data == T) %>%
  ggplot(aes(x = factor(year), y = NDVI, fill = biogeo)) +
  geom_boxplot(outlier.shape = NA)
```



```
db_resurv_RS_short %>% filter(EUNISa_1 == "R" & RS_data == T) %>%
  ggplot(aes(x = factor(year), y = NDVI, fill = biogeo)) +
  geom_boxplot(outlier.shape = NA)
```



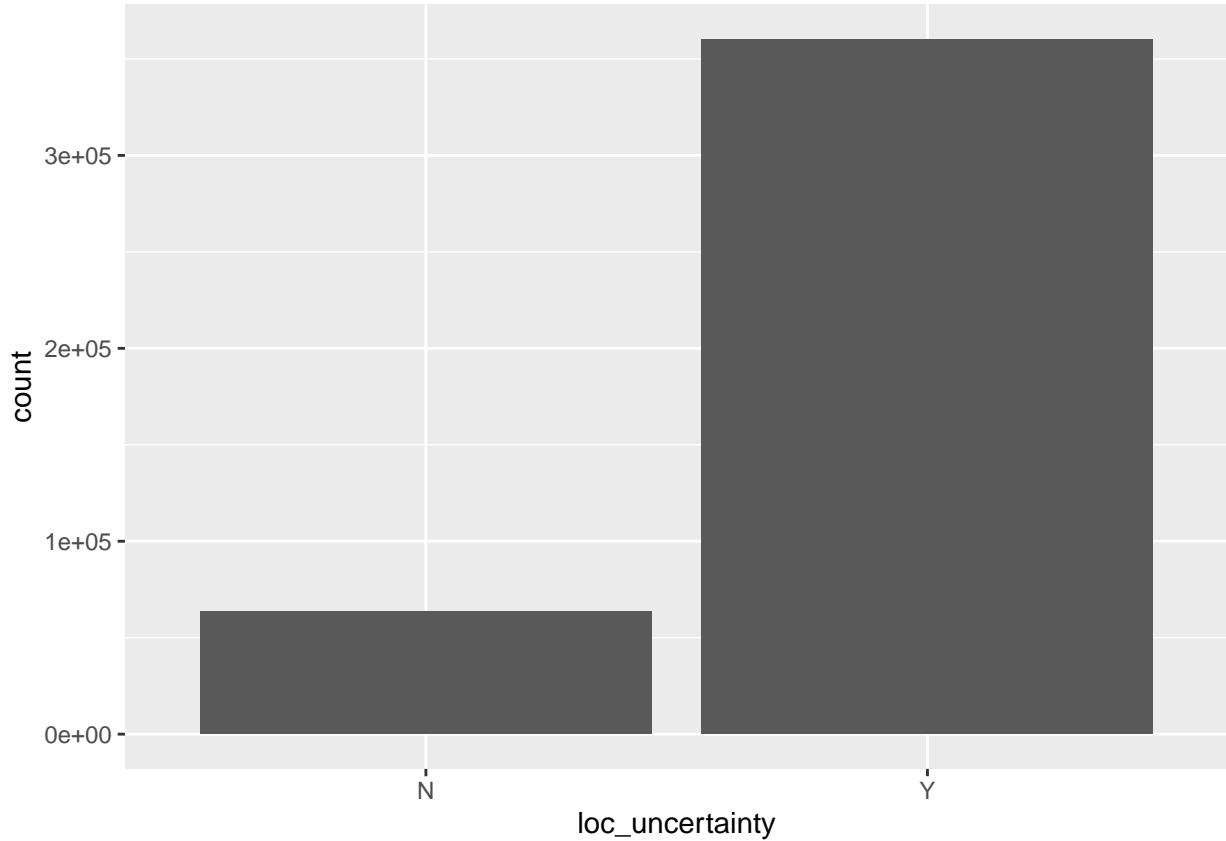
Save as shapefile

```
# st_write(db_resurv_RS_short %>% filter(!is.na(Lon_updated)) %>%
#   st_as_sf(coords = c("Lon_updated", "Lat_updated"), crs = 4326),
#   here("../", "..", "..", "GIS", "shapefiles", "db_resurv_RS_short.shp"))
```

Keep only “good” points

How many points with data about Location uncertainty?

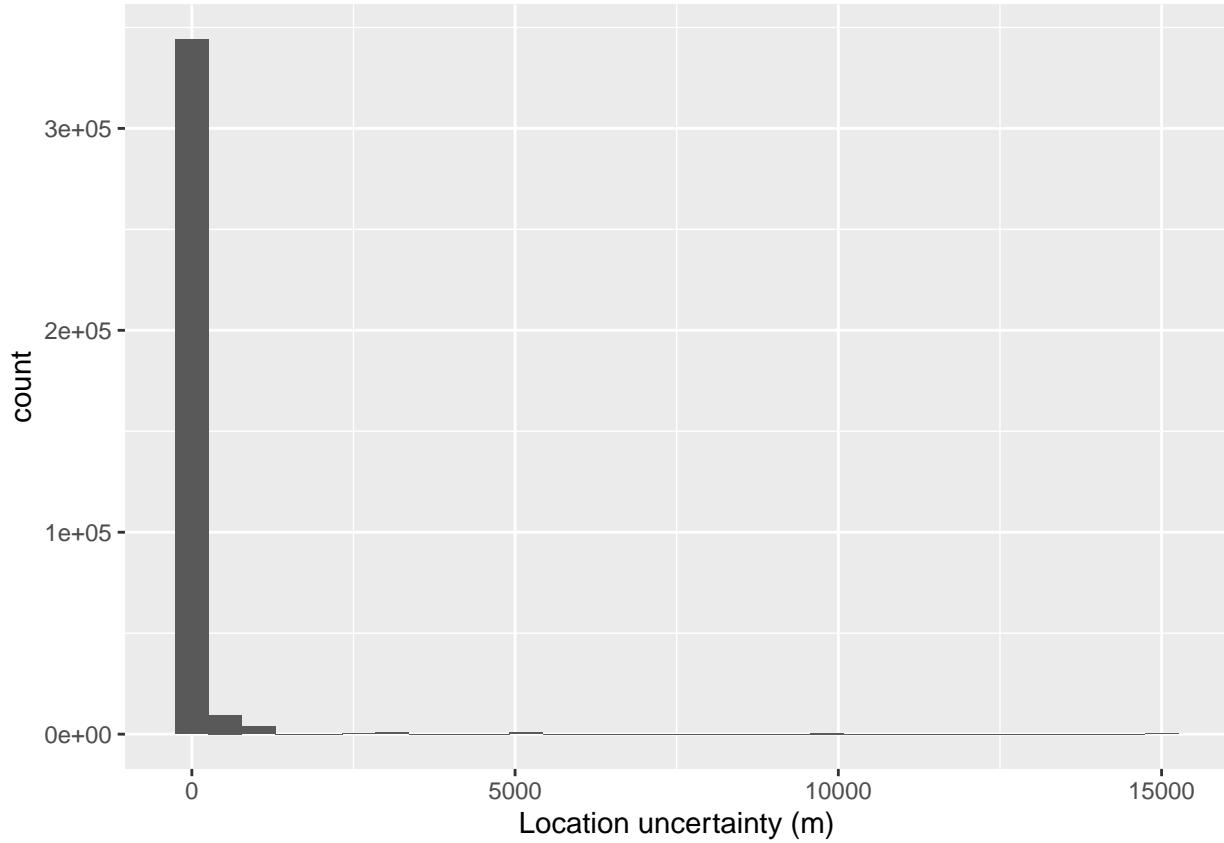
```
db_resurv_RS_short %>%
  mutate(loc_uncertainty = ifelse(is.na(`Location uncertainty (m)`), "N", "Y")) %>%
  ggplot(aes(x = loc_uncertainty)) + geom_bar()
```



Distribution of location uncertainty when it is available (i.e. no NA):

```
db_resurv_RS_short %>%
  filter(!is.na(`Location uncertainty (m)`)) %>%
  ggplot(aes(x = `Location uncertainty (m)`)) + geom_histogram()

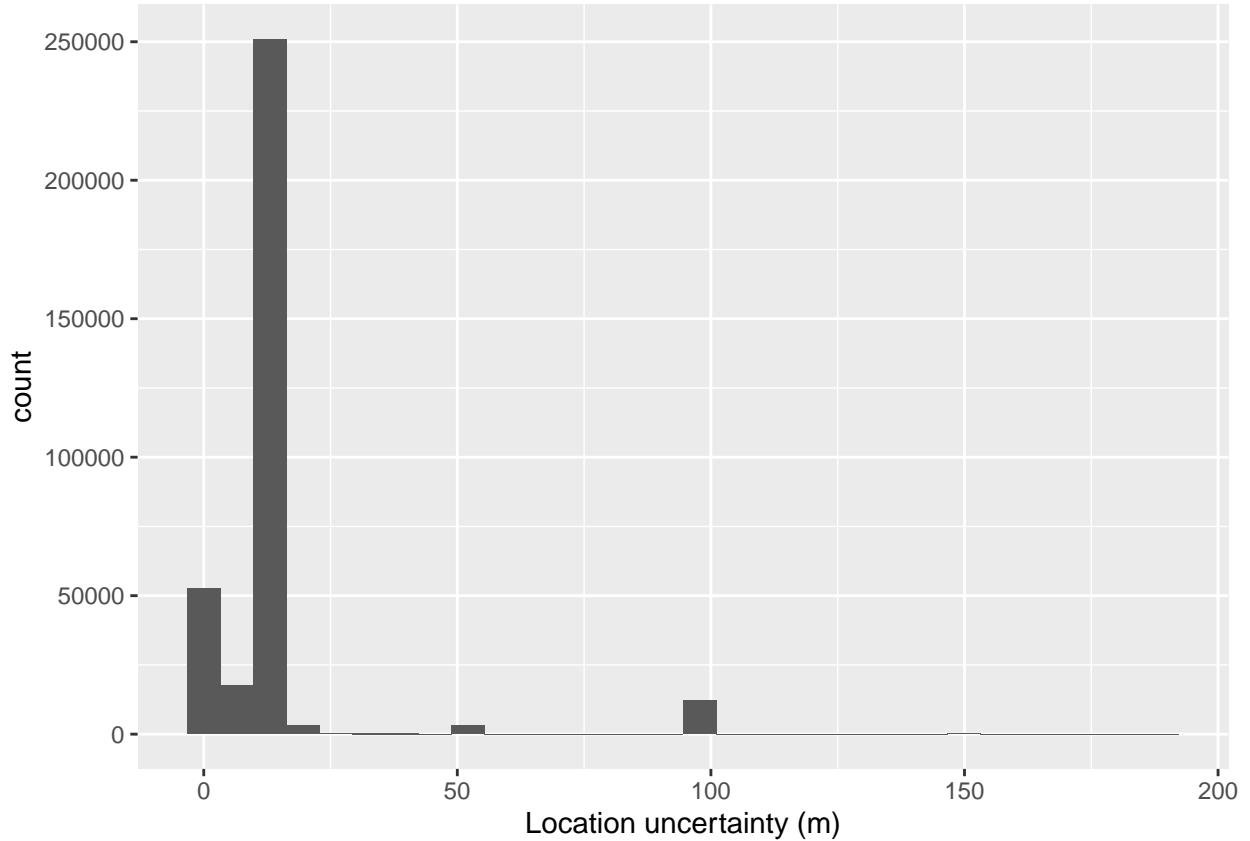
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



Distribution of location uncertainty when it is < 200 m:

```
db_resurv_RS_short %>%
  filter(`Location uncertainty (m)` < 200) %>%
  ggplot(aes(x = `Location uncertainty (m)`)) + geom_histogram()

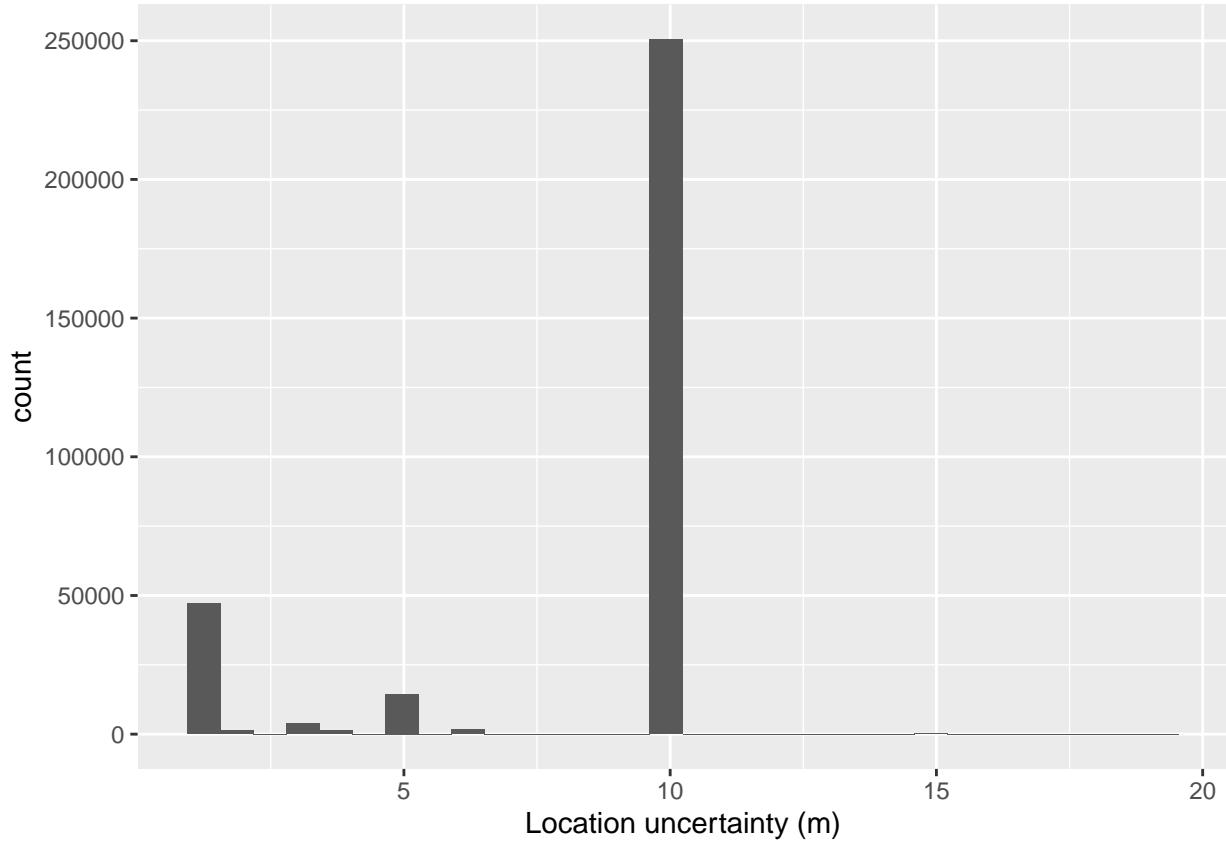
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Distribution of location uncertainty when it is < 20 m:

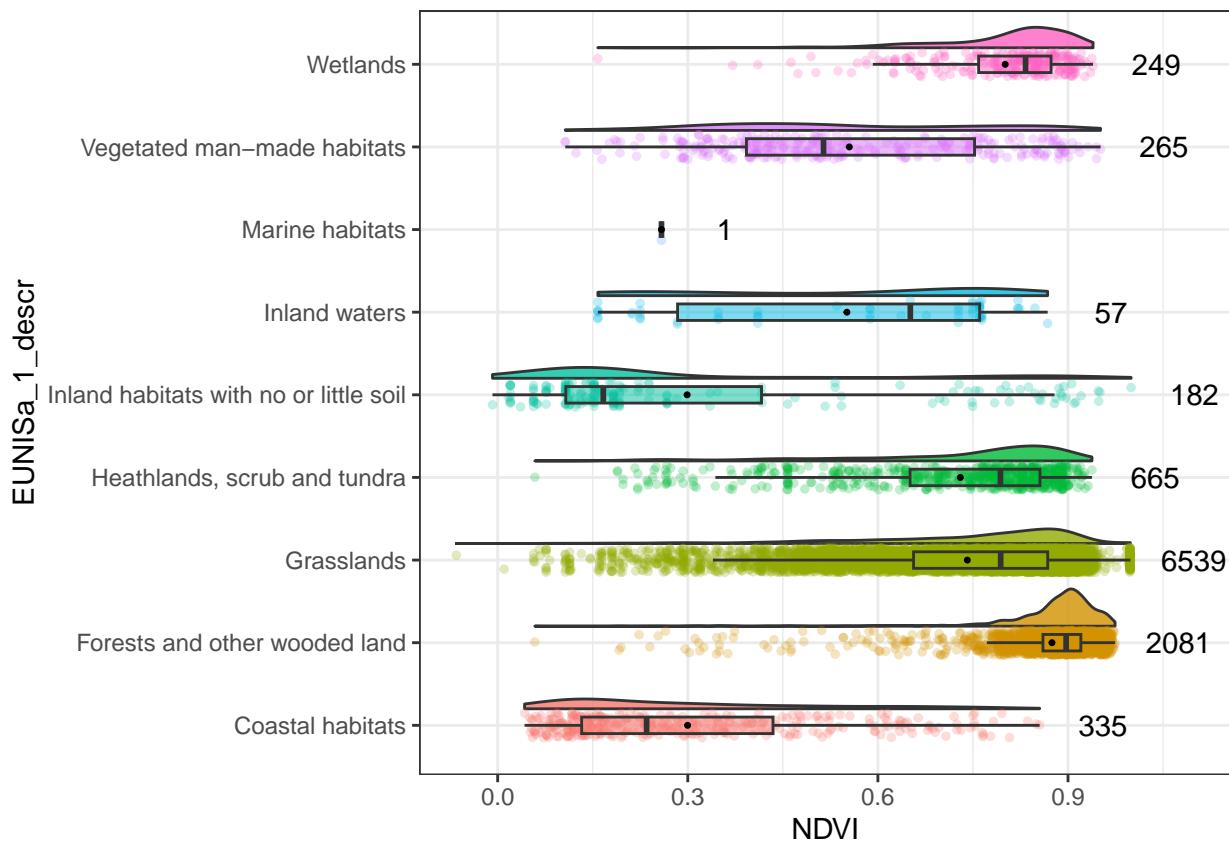
```
db_resurv_RS_short %>%
  filter(`Location uncertainty (m)` < 20) %>%
  ggplot(aes(x = `Location uncertainty (m)`)) + geom_histogram()
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



Distributions with uncertainty < 20 m

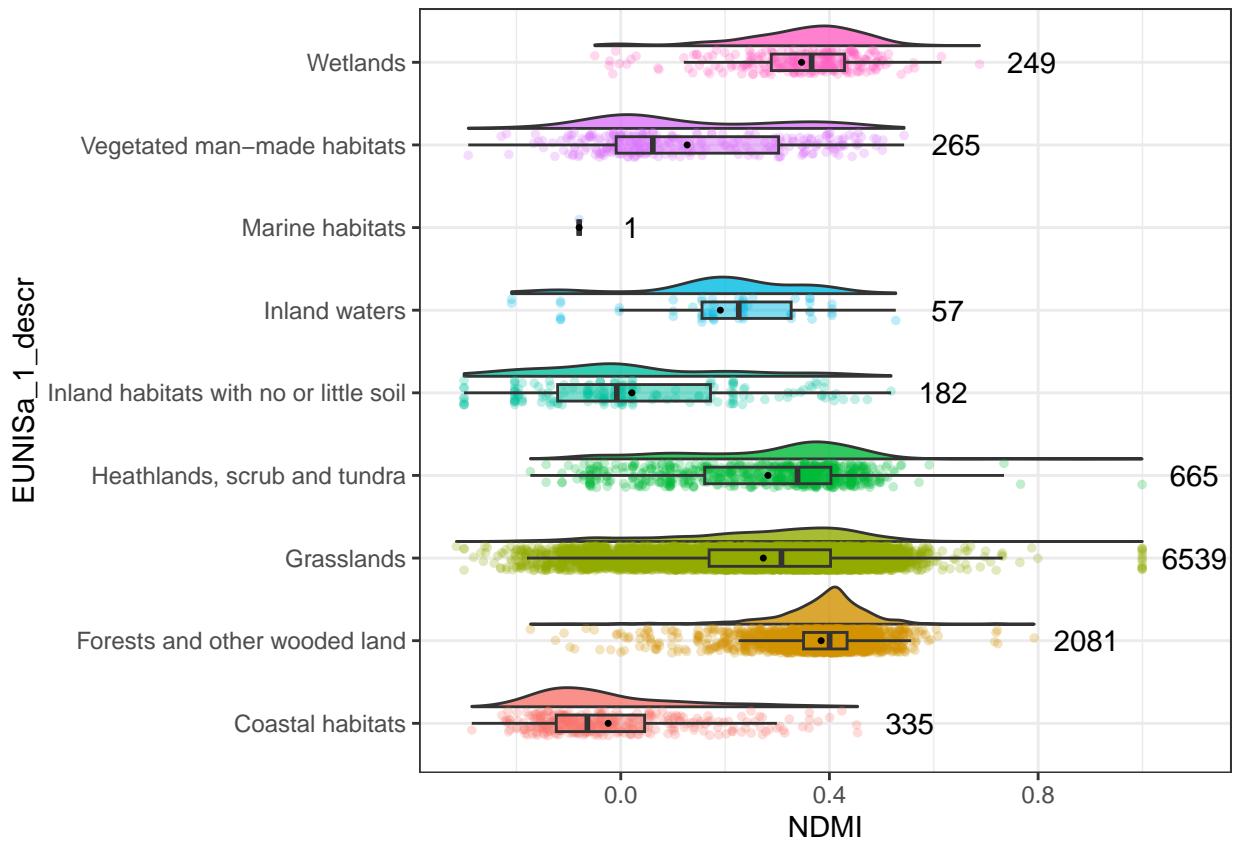
```
ggplot(data = db_resurv_RS_short %>%
  filter(`Location uncertainty (m)` < 20) %>%
  filter(RS_data == T & !is.na(EUNISA_1_descr)),
  aes(x = EUNISA_1_descr, y = NDVI, fill = EUNISA_1_descr)) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = NDVI, color = EUNISA_1_descr),
             position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 1) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                label = length(x)),
               geom = "text", aes(label = ..label..), vjust = 0.5) +
  labs(y = "NDVI", x = "EUNISA_1_descr") +
  guides(fill = FALSE, color = FALSE) +
  theme_bw() + coord_flip()
```



```

ggsave(
  here("output", "figures", "NDVI_20.tif"),
  width = 21, height = 29.7, units = "cm", dpi = 300)
ggplot(data = db_resurv_RS_short %>%
  filter(`Location uncertainty (m)` < 20) %>%
  filter(RS_data == T & !is.na(EUNISa_1_descr)),
  aes(x = EUNISa_1_descr, y = NDMI, fill = EUNISa_1_descr)) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = NDMI, color = EUNISa_1_descr),
             position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 1) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                label = length(x)),
              geom = "text", aes(label = ..label..), vjust = 0.5) +
  labs(y = "NDMI", x = "EUNISa_1_descr") +
  guides(fill = FALSE, color = FALSE) +
  theme_bw() + coord_flip()

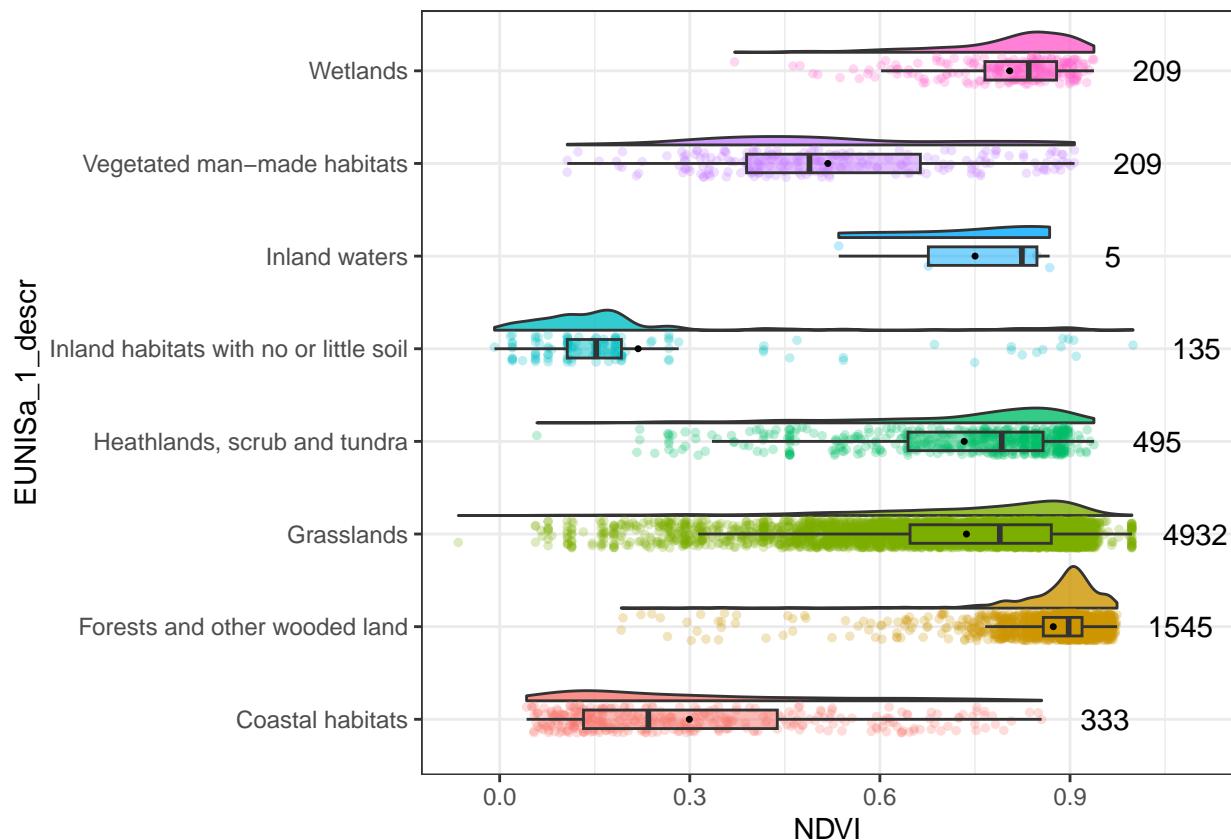
```



```
ggsave(
  here("output", "figures", "NDMI_20.tif"),
  width = 21, height = 29.7, units = "cm", dpi = 300)
```

Distributions with uncertainty < 5 m

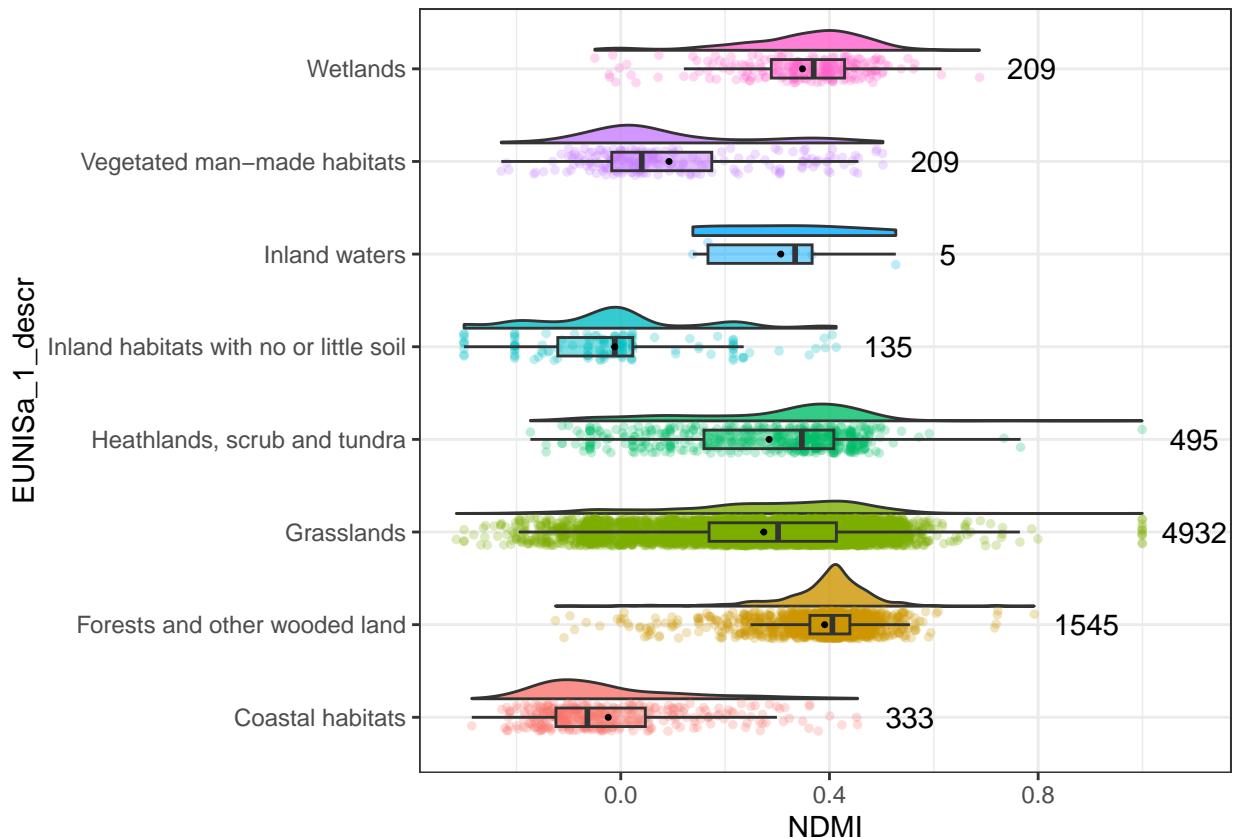
```
ggplot(data = db_resurv_RS_short %>%
  filter(`Location uncertainty (m)` < 5) %>%
  filter(RS_data == T & !is.na(EUNISa_1_descr)),
  aes(x = EUNISa_1_descr, y = NDVI, fill = EUNISa_1_descr)) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = NDVI, color = EUNISa_1_descr),
             position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 1) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                label = length(x)),
              geom = "text", aes(label = ..label..), vjust = 0.5) +
  labs(y = "NDVI", x = "EUNISa_1_descr") +
  guides(fill = FALSE, color = FALSE) +
  theme_bw() + coord_flip()
```



```

ggsave(
  here("output", "figures", "NDVI_5.tif"),
  width = 21, height = 29.7, units = "cm", dpi = 300)
ggplot(data = db_resurv_RS_short %>%
  filter(`Location uncertainty (m)` < 5) %>%
  filter(RS_data == T & !is.na(EUNISa_1_descr)),
  aes(x = EUNISa_1_descr, y = NDMI, fill = EUNISa_1_descr)) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = NDMI, color = EUNISa_1_descr),
             position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 1) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                label = length(x)),
              geom = "text", aes(label = ..label..), vjust = 0.5) +
  labs(y = "NDMI", x = "EUNISa_1_descr") +
  guides(fill = FALSE, color = FALSE) +
  theme_bw() + coord_flip()

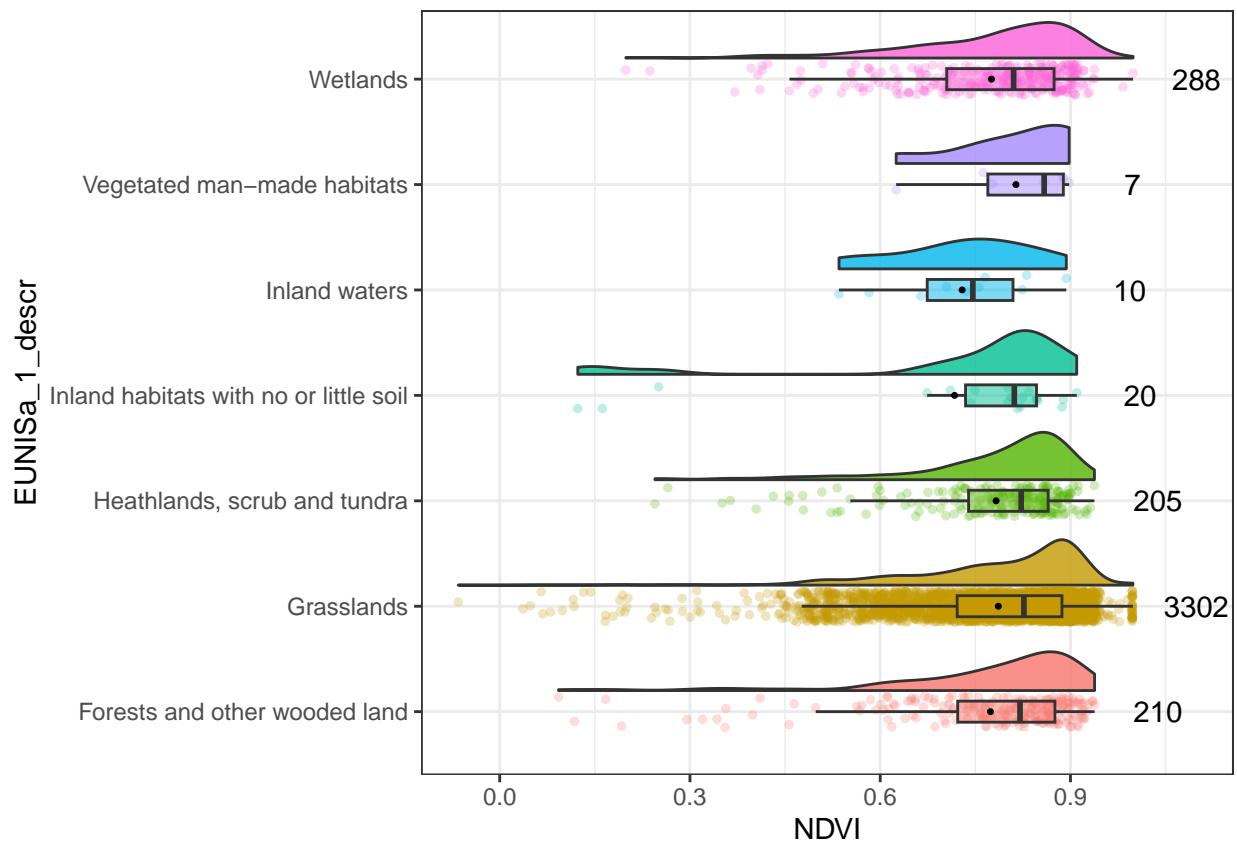
```



```
ggsave(
  here("output", "figures", "NDVI_5.tiff"),
  width = 21, height = 29.7, units = "cm", dpi = 300)
```

Distributions with Location method = “Location with differential GPS”

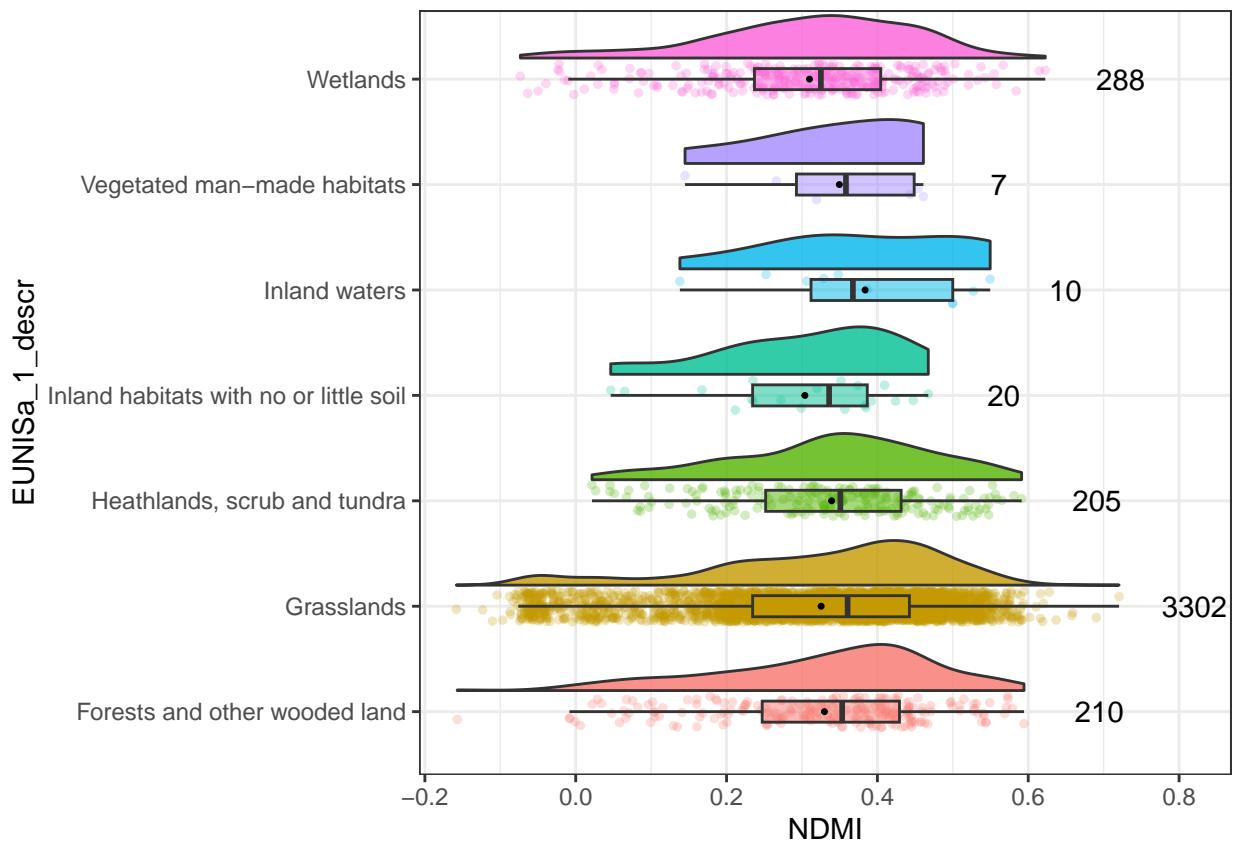
```
ggplot(data = db_resurv_RS_short %>%
  filter(`Location method` == "Location with differential GPS") %>%
  filter(RS_data == T & !is.na(EUNISa_1_descr)),
  aes(x = EUNISa_1_descr, y = NDVI, fill = EUNISa_1_descr)) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = NDVI, color = EUNISa_1_descr),
             position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 1) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                label = length(x)),
              geom = "text", aes(label = ..label..), vjust = 0.5) +
  labs(y = "NDVI", x = "EUNISa_1_descr") +
  guides(fill = FALSE, color = FALSE) +
  theme_bw() + coord_flip()
```



```

ggsave(
  here("output", "figures", "NDVI_differential_GPS.tif"),
  width = 21, height = 29.7, units = "cm", dpi = 300)
ggplot(data = db_resurv_RS_short %>%
  filter(`Location method` == "Location with differential GPS") %>%
  filter(RS_data == T & !is.na(EUNISA_1_descr)),
  aes(x = EUNISA_1_descr, y = NDMI, fill = EUNISA_1_descr)) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = NDMI, color = EUNISA_1_descr),
             position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 1) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                label = length(x)),
               geom = "text", aes(label = ..label..), vjust = 0.5) +
  labs(y = "NDMI", x = "EUNISA_1_descr") +
  guides(fill = FALSE, color = FALSE) +
  theme_bw() + coord_flip()

```



```
ggsave(
  here("output", "figures", "NDVI_differential_GPS.tif"),
  width = 21, height = 29.7, units = "cm", dpi = 300)
```

```
ggplot(data = filter(db_resurv_RS_short %>%
                      filter(`Location uncertainty (m)` < 50) %>%
                      filter(RS_data == T & !is.na(EUNISa_1_descr))),
       aes(x = biogeo, y = NDVI, fill = biogeo)) +
  facet_wrap(~ EUNISa_1_descr) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = NDVI, color = biogeo),
             position = position_jitter(width = 0.15), size = 1, alpha = 0.25) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 1) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                label = length(x)),
              geom = "text", aes(label = ..label..), vjust = 0.5) +
  labs(y = "NDVI", x = "biogeo") +
  guides(fill = FALSE, color = FALSE) +
  theme_bw()
```

```
## Warning: Groups with fewer than two datapoints have been dropped.
## i Set 'drop = FALSE' to consider such groups for position adjustment purposes.
```

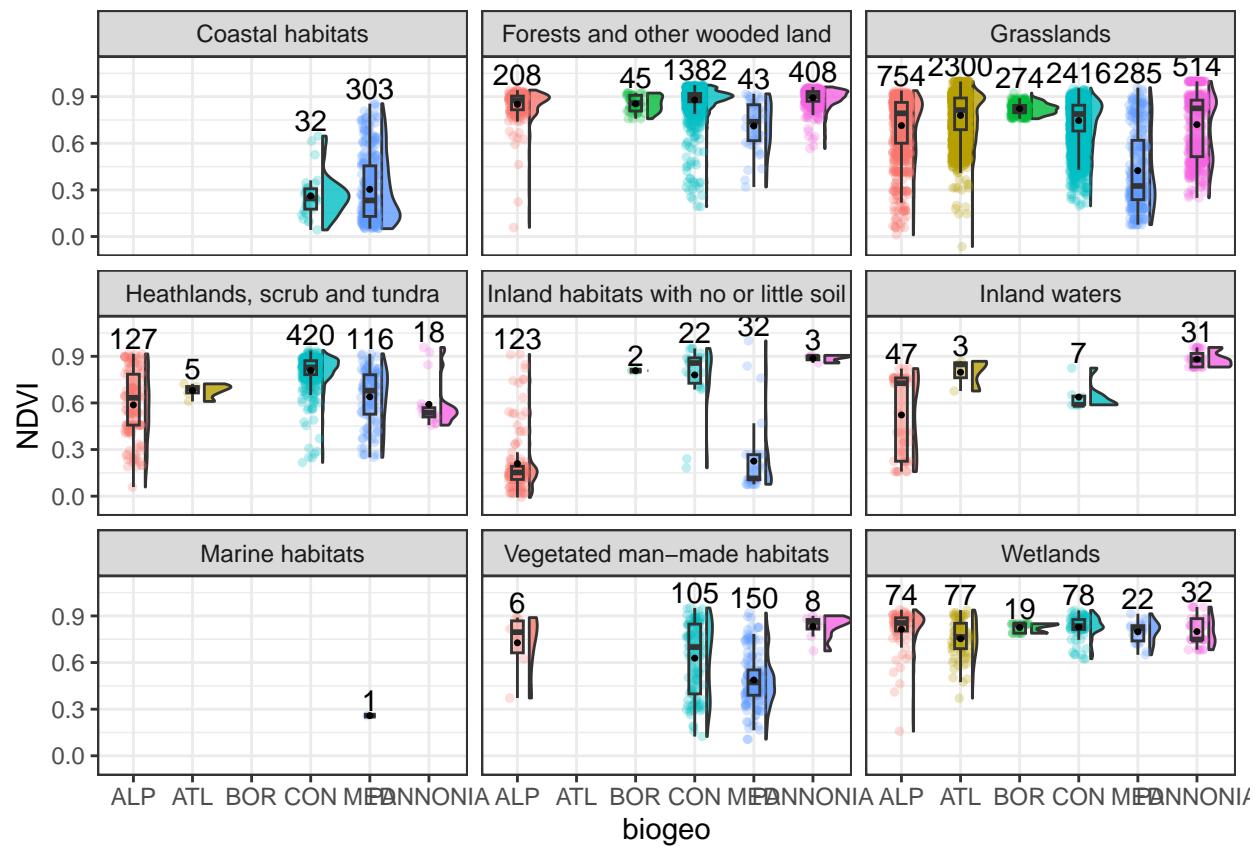
```
## Warning in max(data$density, na.rm = TRUE): ningun argumento finito para max;
```

```

## retornando -Inf

## Warning: Computation failed in 'stat_ydensity()'.
## Caused by error in '$<- .data.frame':
## ! replacement has 1 row, data has 0

```



Save csv for SONATA workshop

Points with uncertainty < 50 m, last resurvey for each plot.

```

points SONATA <- db_resurv_RS_short %>%
  filter(`Location uncertainty (m)` < 50 & RS_data == T &
         !is.na(EUNISa_1_descr) & Lon_diff == F & Lat_diff == F) %>%
  group_by(RS_CODE, `ReSurvey site`, `ReSurvey plot`) %>%
  mutate(max_year = max(year)) %>%
  filter(year == max_year) %>%
  # In some cases there is more than one row per group,
  # i.e. more than one row for the last year
  slice_sample(n = 1) %>%
  ungroup() %>%
  select(-max_year, -year_RS, -source, -Lon_RS, -Lat_RS, -RS_data, -year_diff,
         -Lon_diff, -Lat_diff)

```

Plot points in a map:

```

# Load world boundaries
world <- ne_countries(scale = "medium", returnclass = "sf")

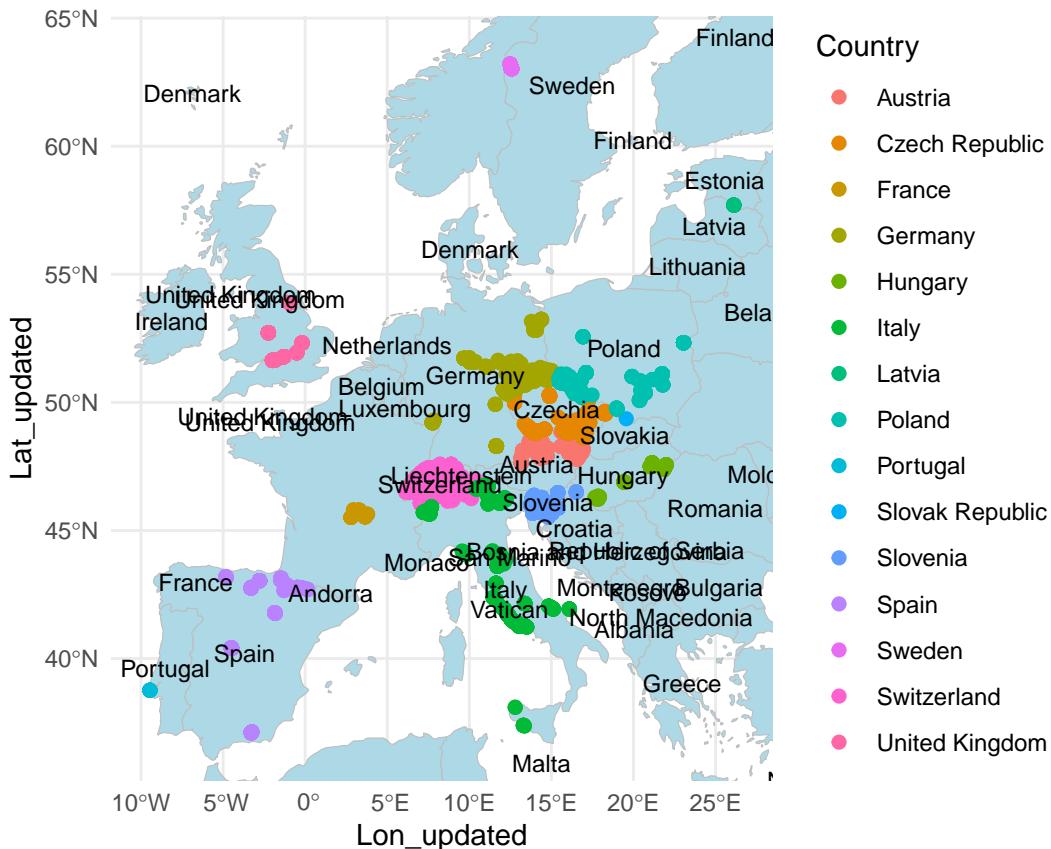
# Calculate the extent of the points
extent <- points SONATA %>%
  summarise(
    lon_min = min(Lon_updated, na.rm = TRUE),
    lon_max = max(Lon_updated, na.rm = TRUE),
    lat_min = min(Lat_updated, na.rm = TRUE),
    lat_max = max(Lat_updated, na.rm = TRUE)
  )

# Add padding to the extent (adjust as needed)
padding <- 0.5 # Adjust padding to your preference
x_limits <- c(extent$lon_min - padding, extent$lon_max + padding)
y_limits <- c(extent$lat_min - padding, extent$lat_max + padding)

# Compute centroids for labeling
world_centroids <- world %>%
  mutate(centroid = st_centroid(geometry)) %>%
  mutate(lon = st_coordinates(centroid)[, 1],
         lat = st_coordinates(centroid)[, 2])

# Create the zoomed map
ggplot() +
  geom_sf(data = world, fill = "lightblue", color = "gray") +
  geom_point(data = points SONATA %>%
    filter(`Location uncertainty (m)` < 20),
    aes(x = Lon_updated, y = Lat_updated, color = Country),
    size = 2) +
  geom_text(data = world_centroids,
            aes(x = lon, y = lat, label = sovereign),
            size = 3, color = "black") + # Adjust size and color as needed
  coord_sf(xlim = x_limits, ylim = y_limits) +
  theme_minimal()

```



Save csv:

```
write_tsv(points SONATA, here("data", "clean", "points SONATA.csv"))
```

Puntos Cordillera

Read and merge RS data

Read sheets from Excel file sent by Gonzalo (and edited by me to create new sheets). Rename identifiers (Codigo and Plot_ID to id to be able to bind rows of both datasets).

```
cordi1 <- read_excel(
  here("data", "edited", "NDVI_NDMI_Pastizales_Bosques_2_AV.xlsx"),
  sheet = "pastizal_AV") %>%
  rename(id = "Relevé_num") %>% mutate(id = as.character(id))
cordi2 <- read_excel(
  here("data", "edited", "NDVI_NDMI_Pastizales_Bosques_AV.xlsx"),
  sheet = "bosque_AV") %>%
  rename(id = "Plot_ID")
```

Merge both datasets and get max NDVI and NDMI for each point:

```

cordi <- bind_rows(cordi1, cordi2) %>% select(-date, -mes) %>%
  group_by(id) %>%
  summarise(max_NDVI = max(NDVI, na.rm = TRUE),
            max_NDMI = max(NDMI, na.rm = TRUE))

```

Read and merge shapefiles to get vegetation / habitat info

```

alpine_grasslands <- st_read(
  here("../", "..", "GIS", "shapefiles", "EVEREST",
    "AlpineGrasslands_WGS84.shp")) %>%
  rename(id = "Relevé_num", hab = "Hábitat") %>% select (id, hab) %>% as_tibble() %>%
  mutate(id = as.character(id))

## Reading layer 'AlpineGrasslands_WGS84' from data source
##   'C:\Users\jimenezalfaro\OneDrive - Universidad de Oviedo\IMIB\GIS\shapefiles\EVEREST\AlpineGrasslan
##   using driver 'ESRI Shapefile'
## Simple feature collection with 40 features and 24 fields
## Geometry type: POINT
## Dimension:      XY
## Bounding box:  xmin: -6.34826 ymin: 42.78827 xmax: -6.037457 ymax: 43.0219
## Geodetic CRS:  WGS 84

vegetation_types <- st_read(
  here("../", "..", "GIS", "shapefiles", "EVEREST",
    "VegetationTypes_WGS84.shp")) %>%
  rename(id = "Plot_ID", hab = "Vegetation") %>% select(id, hab) %>% as_tibble()

## Reading layer 'VegetationTypes_WGS84' from data source
##   'C:\Users\jimenezalfaro\OneDrive - Universidad de Oviedo\IMIB\GIS\shapefiles\EVEREST\VegetationType
##   using driver 'ESRI Shapefile'
## Simple feature collection with 30 features and 5 fields
## Geometry type: POINT
## Dimension:      XY
## Bounding box:  xmin: -3.307347 ymin: 43.07503 xmax: -3.095917 ymax: 43.23479
## Geodetic CRS:  WGS 84

habitats <- bind_rows(alpine_grasslands, vegetation_types)

```

Merge RS and habitat data

```

cordi <- habitats %>%
  left_join(cordi)

## Joining with 'by = join_by(id)'

```

See the different info on hab:

```

cordi %>% distinct(hab)

## # A tibble: 37 x 1
##   hab
##   <chr>
## 1 Pastizal alpino
## 2 Cervunal
## 3 Suelo pedregoso crioturbado
## 4 Pastizal en claro de matorral poco pastado
## 5 Pastizal alpino en zona de nevero
## 6 Pastizal en suelo crioturbado, pedregoso
## 7 <NA>
## 8 Pastizal alpino en suelo crioturbado
## 9 Pastizal alpino pedregoso
## 10 Pastizal posiblemente pastado
## # i 27 more rows

```

Assign to EUNIS level 1:

```

cordi <- cordi %>%
  mutate(EUNIS_1 = case_when(hab == str_detect(hab, "Pastizal|Cervunal|grassland|meadow") ~ "R",
                             hab == str_detect(hab, "forest") ~ "T",
                             hab == str_detect(hab, "Scrub|scrub|Shrubland|shrubland|shrub|Heathland") ~ "S",
                             hab == str_detect(hab, "Suelo") ~ "U",
                             TRUE ~ NA_character_))

```

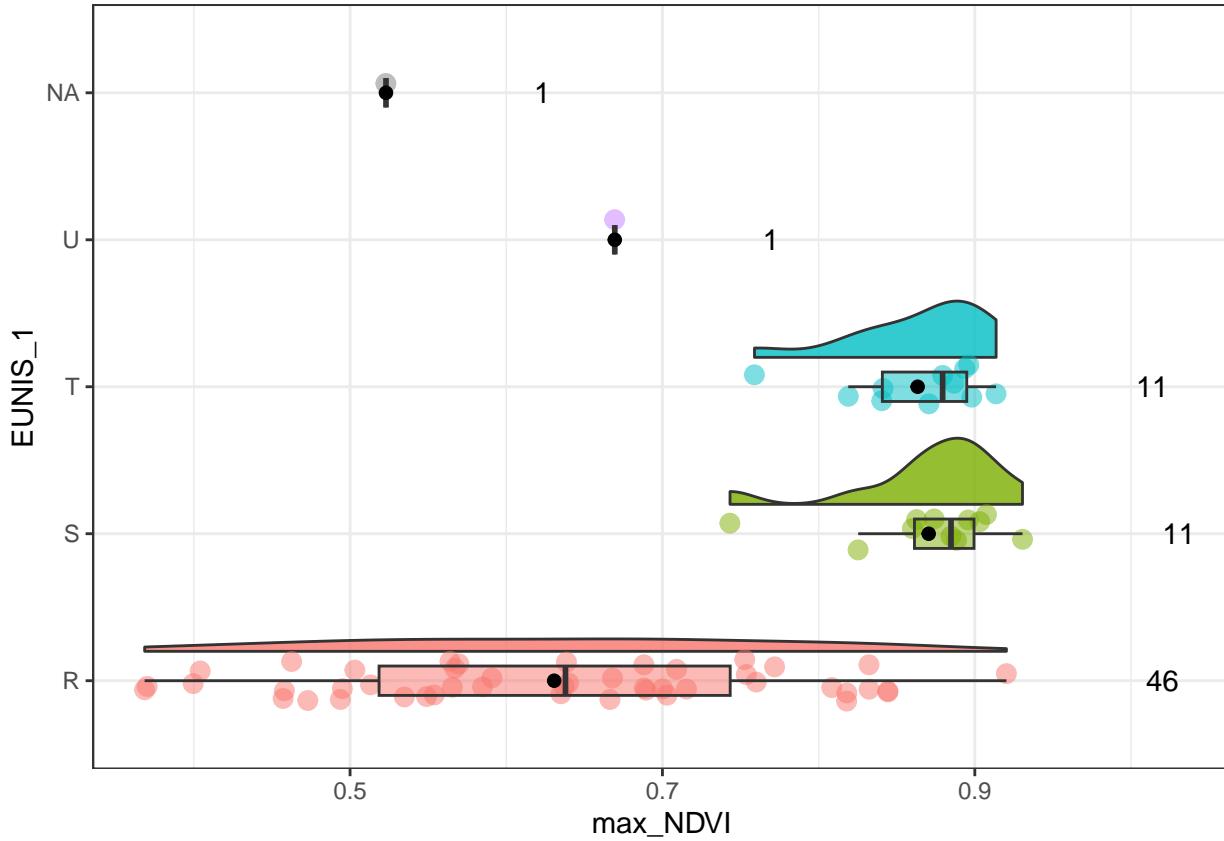
Plots NDVI and NDMI

```

ggplot(data = cordi %>% filter(!is.na(max_NDVI)),
       aes(x = EUNIS_1, y = max_NDVI, fill = EUNIS_1)) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = max_NDVI, color = EUNIS_1),
             position = position_jitter(width = 0.15), size = 3,
             alpha = 0.5) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 3) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                 label = length(x)),
               geom = "text", aes(label = ..label..), vjust = 0.5) +
  guides(fill = FALSE, color = FALSE) + theme_bw() + coord_flip()

## Warning: Groups with fewer than two datapoints have been dropped.
## i Set 'drop = FALSE' to consider such groups for position adjustment purposes.
## Groups with fewer than two datapoints have been dropped.
## i Set 'drop = FALSE' to consider such groups for position adjustment purposes.

```

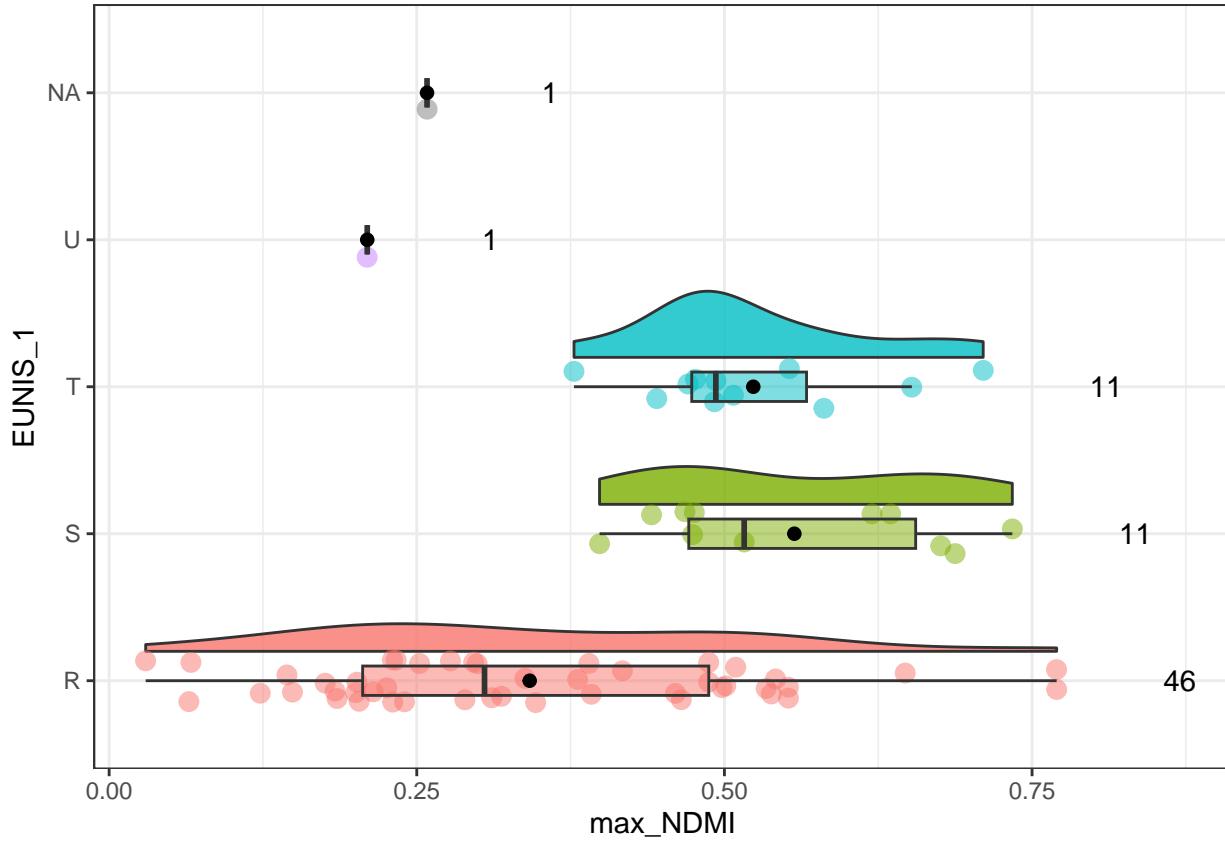


```
ggsave(here("output", "figures", "cordi_NDMI.tif"),
       width = 21, height = 29.7, units = "cm", dpi = 300)
```

```
## Warning: Groups with fewer than two datapoints have been dropped.
## i Set 'drop = FALSE' to consider such groups for position adjustment purposes.
## Groups with fewer than two datapoints have been dropped.
## i Set 'drop = FALSE' to consider such groups for position adjustment purposes.
```

```
ggplot(data = cordi %>% filter(!is.na(max_NDMI)),
       aes(x = EUNIS_1, y = max_NDMI, fill = EUNIS_1)) +
  geom_flat_violin(position = position_nudge(x = 0.2, y = 0), alpha = 0.8) +
  geom_point(aes(y = max_NDMI, color = EUNIS_1),
             position = position_jitter(width = 0.15), size = 3,
             alpha = 0.5) +
  geom_boxplot(width = 0.2, outlier.shape = NA, alpha = 0.5) +
  stat_summary(fun.y=mean, geom="point", shape = 20, size = 3) +
  stat_summary(fun.data = function(x) data.frame(y = max(x) + 0.1,
                                                label = length(x)),
              geom = "text", aes(label = ..label..), vjust = 0.5) +
  guides(fill = FALSE, color = FALSE) + theme_bw() + coord_flip()
```

```
## Warning: Groups with fewer than two datapoints have been dropped.
## i Set 'drop = FALSE' to consider such groups for position adjustment purposes.
## Groups with fewer than two datapoints have been dropped.
## i Set 'drop = FALSE' to consider such groups for position adjustment purposes.
```



```
ggsave(here("output", "figures", "cordi_NDMI.tiff"),
       width = 21, height = 29.7, units = "cm", dpi = 300)
```

```
## Warning: Groups with fewer than two datapoints have been dropped.
## i Set 'drop = FALSE' to consider such groups for position adjustment purposes.
## Groups with fewer than two datapoints have been dropped.
## i Set 'drop = FALSE' to consider such groups for position adjustment purposes.
```

Session info

```
sessionInfo()

## R version 4.4.2 (2024-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Spanish_Spain.utf8  LC_CTYPE=Spanish_Spain.utf8
## [3] LC_MONETARY=Spanish_Spain.utf8 LC_NUMERIC=C
```

```

## [5] LC_TIME=Spanish_Spain.utf8
##
## time zone: Europe/Madrid
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics   grDevices utils      datasets   methods    base
##
## other attached packages:
## [1] readxl_1.4.3          rnaturalearth_1.0.1 gridExtra_2.3
## [4] sf_1.0-19              here_1.0.1           lubridate_1.9.4
## [7] forcats_1.0.0          stringr_1.5.1        dplyr_1.1.4
## [10] purrrr_1.0.2          readr_2.1.5          tidyverse_2.0.0
## [13] tibble_3.2.1           ggplot2_3.5.1        tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
## [1] gtable_0.3.6            xfun_0.50             tzdb_0.4.0
## [4] vctrs_0.6.5              tools_4.4.2            generics_0.1.3
## [7] parallel_4.4.2           proxy_0.4-27          pkgconfig_2.0.3
## [10] KernSmooth_2.23-24       lifecycle_1.0.4       farver_2.1.2
## [13] compiler_4.4.2           textshaping_1.0.0     munsell_0.5.1
## [16] terra_1.8-15            codetools_0.2-20      htmltools_0.5.8.1
## [19] class_7.3-22             yaml_2.3.10           pillar_1.10.1
## [22] crayon_1.5.3             classInt_0.4-11       wk_0.9.4
## [25] rnaturalearthdata_1.0.0  tidyselect_1.2.1     digest_0.6.37
## [28] stringi_1.8.4            labeling_0.4.3         rprojroot_2.0.4
## [31] fastmap_1.2.0            grid_4.4.2             colorspace_2.1-1
## [34] cli_3.6.3                magrittr_2.0.3         utf8_1.2.4
## [37] e1071_1.7-16             withr_3.0.2            scales_1.3.0
## [40] bit64_4.6.0-1            timechange_0.3.0      rmarkdown_2.29
## [43] httr_1.4.7                bit_4.5.0.1            cellranger_1.1.0
## [46] ragg_1.3.3                hms_1.1.3              evaluate_1.0.3
## [49] knitr_1.49                s2_1.1.7               rlang_1.1.5
## [52] Rcpp_1.0.14               glue_1.8.0              DBI_1.2.3
## [55] rstudioapi_0.17.1         vroom_1.6.5            jsonlite_1.8.9
## [58] plyr_1.8.9                R6_2.5.1               systemfonts_1.2.1
## [61] units_0.8-5

```