Using spatially biased variables in ecosystem condition accounting with a GIS based workflow

A Short Subtitle

Anders Lorentzen Kolstad[a,*], Matthew Grainger[a]

[a] *Norwegian Institute for Nature Research, Department of Terrestrial Ecology, Pb 5685 Torgarden, Trondheim, 7485*

Abstract

This is the abstract. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum augue turpis, dictum non malesuada a, volutpat eget velit. Nam placerat turpis purus, eu tristique ex tincidunt et. Mauris sed augue eget turpis ultrices tincidunt. Sed et mi in leo porta egestas. Aliquam non laoreet velit. Nunc quis ex vitae eros aliquet auctor nec ac libero. Duis laoreet sapien eu mi luctus, in bibendum leo molestie. Sed hendrerit diam diam, ac dapibus nisl volutpat vitae. Aliquam bibendum varius libero, eu efficitur justo rutrum at. Sed at tempus elit.

Keywords: alien species, disturbance, indicator, ecosystem condition, wetlands, mire, peatlands, ecosystem accounting, SEEA EA

```
library(tidyverse)
library(knitr)
library(sf)
library(tmap)
library(tmaptools)
library(stars)
library(terra)
library(tidyterra)
library(ggtext)
library(cowplot)
library(units)
library(rnaturalearth)
library(rnaturalearthdata)
library(ggmagnify)
library(ggridges)
library(eaTools) #https://ninanor.github.io/eaTools/ version 0.0.0.9000

myCRS <- 25832
```

```
# Conditional file directory
dir <- substr(getwd(), 1, 2)

# Some directories
```

*Corresponding author

Email addresses: anders.kolstad@nina.no (Anders Lorentzen Kolstad), matthew.grainger@nina.no (Matthew Grainger)

```r
# Ecosystem delineation map
path_mire <- "P:/41201785_okologisk_tilstand_2022_2023/data/Myrmodell/myrmodell90pros.tif"

# infrastructure index
path_infrastructure <- ifelse(dir == "C:",
  "R:/GeoSpatialData/Utility_governmentalServices/Norway_Infrastructure_Index/Original/Infrastrukturin
  "/data/R/GeoSpatialData/Utility_governmentalServices/Norway_Infrastructure_Index/Original/Infrastruk
)

# field survey
# # downloaded from https://kartkatalog.geonorge.no/metadata/naturtyper-miljoedirektoratets-instruks/e
path_naturetypes <- "../data/survey.gdb"

# municipality outline
path_muni <- "../data/Basisdata_0000_Norge_25833_Kommuner_FGDB.gdb"

# path to local caching folder
path_temp <- ifelse(dir == "C:",
  "P:/41201785_okologisk_tilstand_2022_2023/data/cache/",
  "/data/P-Prosjekter2/41201785_okologisk_tilstand_2022_2023/data/cache/"
)
```

```r
# I already did some work to identify the relevant nature types
# summary file (https://github.com/NINAnor/ecosystemCondition/blob/main/data/naturetypes/natureType_su
naturetypes_summary <- readRDS("../data/natureType_summary.rds")

# Survey data
# The data is too big to be stored on GitHub
# Import polygon data set
st_layers(path_naturetypes)
```

```
Driver: OpenFileGDB
Available layers:
              layer_name geometry_type features fields              crs_name
1 naturtyper_nin_dekning Multi Polygon     3519     13 ETRS89 / UTM zone 33N
2     naturtyper_nin_omr Multi Polygon   142031     35 ETRS89 / UTM zone 33N
```

```r
naturetypes <- sf::st_read(dsn = path_naturetypes, layer = "naturtyper_nin_omr")
```

```
Reading layer `naturtyper_nin_omr' from data source
  `C:\Users\anders.kolstad\Github\HIAs\data\survey.gdb' using driver `OpenFileGDB'
Simple feature collection with 142031 features and 35 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -74953.52 ymin: 6448986 xmax: 1079858 ymax: 7921284
Projected CRS: ETRS89 / UTM zone 33N
```

```r
# 142k polygons (2023)

# Impart survey coverage map
coverage <- sf::st_read(dsn = path_naturetypes, layer = "naturtyper_nin_dekning") |>
  st_transform(myCRS)
```

```
Reading layer `naturtyper_nin_dekning' from data source
  `C:\Users\anders.kolstad\Github\HIAs\data\survey.gdb' using driver `OpenFileGDB'
Simple feature collection with 3519 features and 13 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -75023.32 ymin: 6448486 xmax: 1079866 ymax: 7921376
Projected CRS: ETRS89 / UTM zone 33N
```

```r
# Outline of norway (coastline)
outline <- sf::read_sf("../data/outlineOfNorway_EPSG25833.shp") |>
  st_transform(myCRS)

# Municipalities
# find the correct layer
st_layers(path_muni)
```

```
Driver: OpenFileGDB
Available layers:
                                             layer_name     geometry_type
1                            territorialgrense_informasjon                NA
2                                   riksgrense_informasjon                NA
3                            avtaltavgrensningslinje_land                NA
4                                  dokumentasjonsreferanse                NA
5                     avtaltavgrensningslinje_informasjon                NA
6                                   administrativenhetnavn                NA
7                                  avtaltavgrensningslinje Multi Line String
8                                                  kommune     Multi Polygon
9                                           kommunegrense Multi Line String
10                                             riksgrense Multi Line String
11                                      territorialgrense Multi Line String
12                                           fylkesgrense Multi Line String
13 avtaltavgrensningslinje_avtaltavgrensningslinje_land                NA
14     avtaltavgrensningslinje_dokumentasjonsreferansse                NA
   features fields               crs_name
1         0      2                   <NA>
2         0      2                   <NA>
3         6      2                   <NA>
4        83     14                   <NA>
5         3      2                   <NA>
6       389      5                   <NA>
7         6     18 ETRS89 / UTM zone 33N
8       363     13 ETRS89 / UTM zone 33N
9     18096     20 ETRS89 / UTM zone 33N
10      228     18 ETRS89 / UTM zone 33N
11       77     19 ETRS89 / UTM zone 33N
12     2793     20 ETRS89 / UTM zone 33N
13        0      2                   <NA>
14        0      2                   <NA>
```

```r
# read inn data and transform
muni <- sf::read_sf(path_muni, layer = "kommune") |>
  st_transform(myCRS)
```

```r
# Infrastructure index (read proxy)
infra <- stars::read_stars(path_infrastructure)
```

```r
# Mire data
# Spat rasters cannot be cached
mire_terra <- terra::rast(path_mire)
```

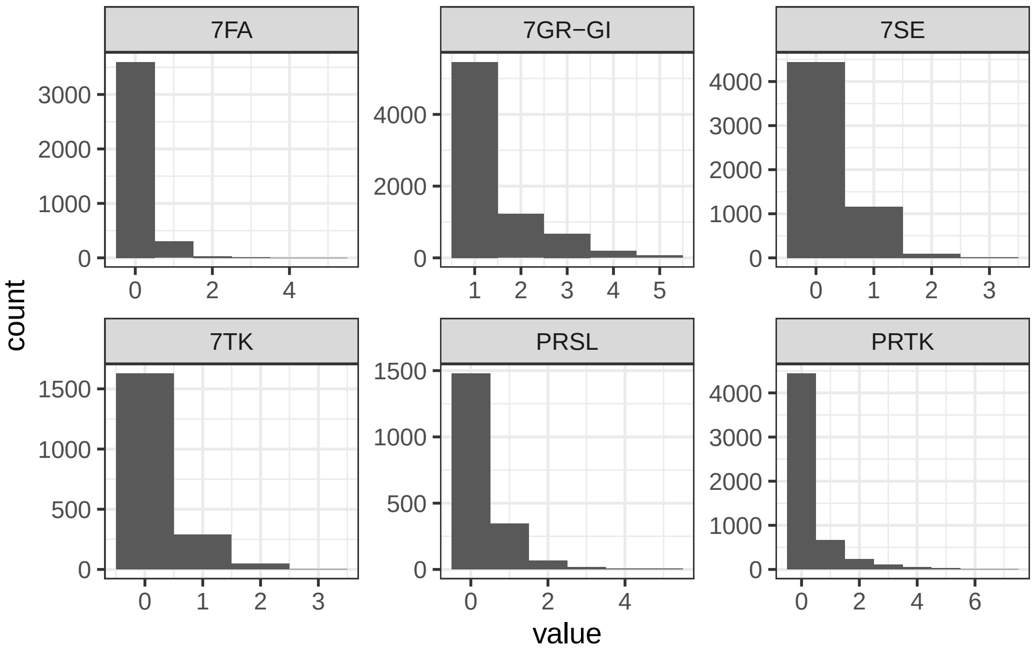
```r
myVars <- c("7TK", "7SE", "PRTK", "PRSL", "7FA", "7GR-GI")
nts <- naturetypes_summary %>%
  rowwise() %>%
  mutate(keepers = sum(c_across(
    all_of(myVars))>0, na.rm=T)) |>
  filter(
    keepers >0,
    Ecosystem == "våtmark"
    ) |>
  pull(Nature_type)
```

```r
# Clean the survey data

naturetypes <- naturetypes |>
  # keep only wetlands
  filter(
    hovedøkosystem == "våtmark",
    naturtype %in% nts,
    naturtype != "Kalkrik helofyttsump"
  ) |>
  # calculate the areas (m2) of the polygons
  mutate(area = SHAPE |> st_area()) |>
  # the variable codes and values are all in the same column
  separate_rows(ninBeskrivelsesvariable, sep = ",") |>
  separate(
    col = ninBeskrivelsesvariable,
    into = c("NiN_variable_code", "NiN_variable_value"),
    sep = "_",
    remove = F
  ) |>
  mutate(NiN_variable_value = as.numeric(NiN_variable_value)) |>
  filter(NiN_variable_code %in% myVars) |>
  select(
    id = identifikasjon_lokalId,
    municipality = kommunenummer,
    year = kartleggingsår,
    mosaic = mosaikk,
    quality = lokalitetskvalitet,
    biodiversity = naturmangfold,
    condition = tilstand,
    natureType = naturtype,
    variable = NiN_variable_code,
    value = NiN_variable_value,
    area
  ) |>
```

```r
  st_transform(myCRS) # Choosing this to match the EDM (se further down)
# 19k obs.
```

```r
# Plot to show what the most common nature types in the data set are
naturetypes |>
  as_tibble() |>
  count(natureType, sort=T) |>
  mutate(natureType = fct_reorder(natureType, n)) |>
  ggplot(aes(x = natureType, y = n))+
  geom_col()+
  coord_flip()
```

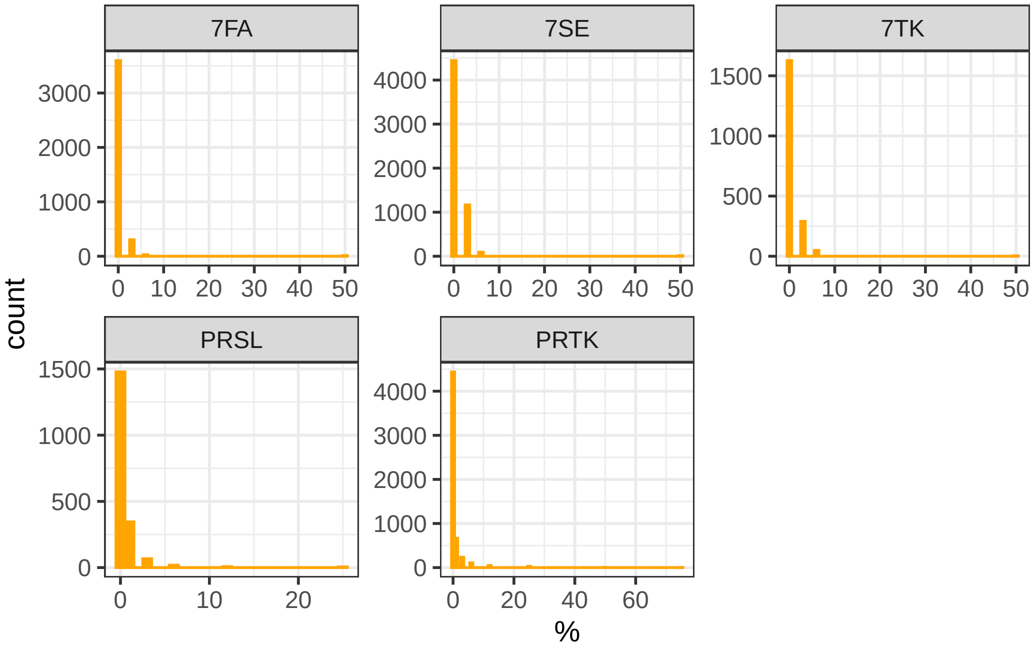
```r
# I now want to take the variables and normalise them before I can then combine
# them despite them being on different scales.
# I will first normalise by converting into % (not for 7GR-GI).
# Remember the ordinal categories represents frequency ranges
# The data is strongly right skewed, so simply taking the center value of each
# bin will not work:
naturetypes %>%
  ggplot() +
  theme_bw() +
  geom_histogram(aes(x = value),
    binwidth = 1
  ) +
  facet_wrap(. ~ variable,
    scales = "free"
  )
```



59

```r
# I will use the lower bound for each bin instead.
# The exception in when the variable is 1, because then the lower bound
```

```r
# is 0, same as when the variable is 0.
# For these I will set manually a slightly higher value.

naturetypes <- naturetypes %>%
  mutate(value = case_when(
    # selecting the variables that follow the same 4 step scale
    variable %in% c("7TK", "7SE", "7FA") ~
      case_match(
        value,
        0 ~ 0,
        1 ~ mean(c(0, 1 / 16)) * 100,
        2 ~ 1 / 16 * 100,
        3 ~ 50
      ), # note that it is not possible to get a value of 1
    # selecting the eight step variables
    variable %in% c("PRTK", "PRSL") ~
      case_match(
        value,
        0 ~ 0,
        1 ~ 1.5,
        2 ~ 3,
        3 ~ 6.25,
        4 ~ 12.5,
        5 ~ 25,
        6 ~ 50,
        7 ~ 75
      ),
    .default = value
  ))

naturetypes %>%
  filter(variable != "7GR-GI") |>
  ggplot() +
  theme_bw() +
  geom_histogram(aes(x = value),
    binwidth = 1,
    color = "orange",
    fill = "orange"
  ) +
  xlab("%") +
  facet_wrap(. ~ variable,
    scales = "free"
  )
```



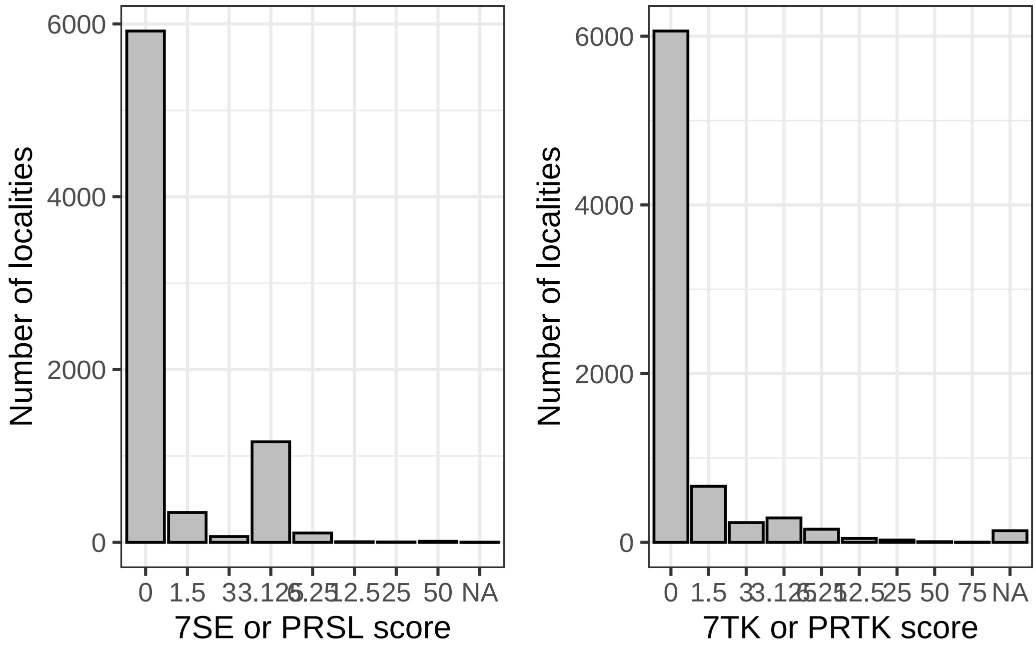

```
# Now I make the data wide, and remove 7TK and 7SE if PRTK or PRSL are present,
# respectively

naturetypes_wide <- naturetypes |>
  filter(variable %in% c("7TK", "7SE", "PRTK", "PRSL")) |>
  # Column names starting with a number is problematic, so adding a prefix
  mutate(variable = paste0("var_", variable)) |>
  pivot_wider(
    names_from = "variable",
    values_from = "value",
    id_cols = "id") |>
    as_tibble()

head(naturetypes_wide, 10)
```

```
# A tibble: 10 x 5
   id                var_7SE var_PRTK var_7TK var_PRSL
   <chr>               <dbl>    <dbl>   <dbl>    <dbl>
 1 NINFP2310142406     0          0        NA       NA
 2 NINFP2010014477     3.12       0        NA       NA
 3 NINFP1910019329     0          1.5      NA       NA
 4 NINFP2010021023    NA         NA         0        1.5
 5 NINFP2010016336     0          0        NA       NA
 6 NINFP1910006253     0          0        NA       NA
 7 NINFP2210112856     0          0        NA       NA
 8 NINFP2010007785    NA         NA         0        0
 9 NINFP2010027022    NA         NA         0        0
10 NINFP2210099311     0          0        NA       NA
```

```
# First I will combine 7TK and PRTK, and also 7SE and PRSL.
naturetypes_wide <- naturetypes_wide %>%
```

```
  mutate(
    TK = if_else(
      is.na(var_PRTK), var_7TK, var_PRTK
    ),
    SE = if_else(
      is.na(var_PRSL), var_7SE, var_PRSL
    )
  )

plot_grid(
  naturetypes_wide %>%
    as_tibble() |>
    count(SE,
      name = "sum"
    ) |>
    ggplot(
      aes(
        x = factor(SE),
        y = sum
      )
    ) +
    geom_bar(
      stat = "identity",
      fill = "grey",
      colour = "black"
    ) +
    theme_bw(base_size = 12) +
    labs(
      x = "7SE or PRSL score",
      y = "Number of localities"
    ),
  naturetypes_wide %>%
    as_tibble() |>
    count(TK,
      name = "sum"
    ) |>
    ggplot(
      aes(
        x = factor(TK),
        y = sum
      )
    ) +
    geom_bar(
      stat = "identity",
      fill = "grey",
      colour = "black"
    ) +
    theme_bw(base_size = 12) +
    labs(
      x = "7TK or PRTK score",
      y = "Number of localities"
```

```
    )
)
```

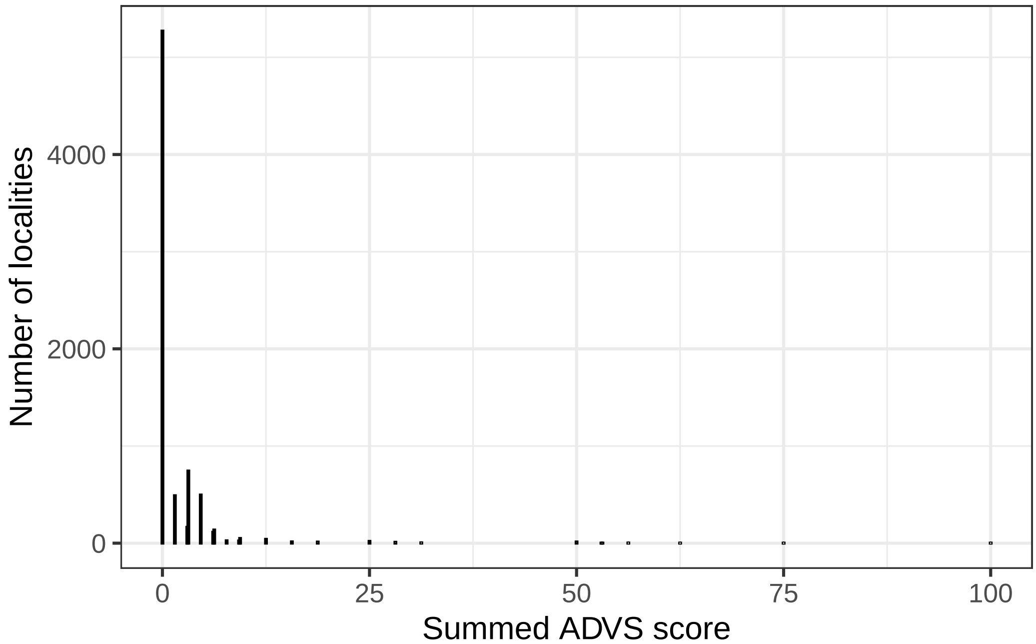


74

```
# The NA's represents localities where just one of the two variables
# (then thinking 7SE and PRSL as the same variable)
# is recorded. To combine these into one metric ADSV, I could take the
# one with the highest value (worst-rule) og the sum.
# Sum is problematic is not all locations have two values to sum together.
# But the other option is problematic since I think field workers often
# tend to split the effects over two variables is they have that option.
# And if we have 50% vehicle damage and 50% hiking damage, that is no doubt
# worst than just having 50% of either. So I will use the sum, despite its issues.
```

```
# Taking the sum of 7SE and 7TK (incl the PR.. variables)
naturetypes_wide <- naturetypes_wide |>
  rowwise() |>
  mutate(ADSV = sum(c(SE, TK), na.rm = TRUE))

naturetypes_wide %>%
  as_tibble() |>
  count(ADSV,
    name = "sum"
  ) |>
  ggplot(
    aes(
      x = ADSV,
      y = sum
    )
  ) +
  geom_bar(
```

```
    stat = "identity",
    fill = "grey",
    colour = "black"
  ) +
  theme_bw(base_size = 12) +
  labs(
    x = "Summed ADVS score",
    y = "Number of localities"
  ) +
  scale_x_continuous(
    labels = scales::label_number(accuracy = 1)
  )
```

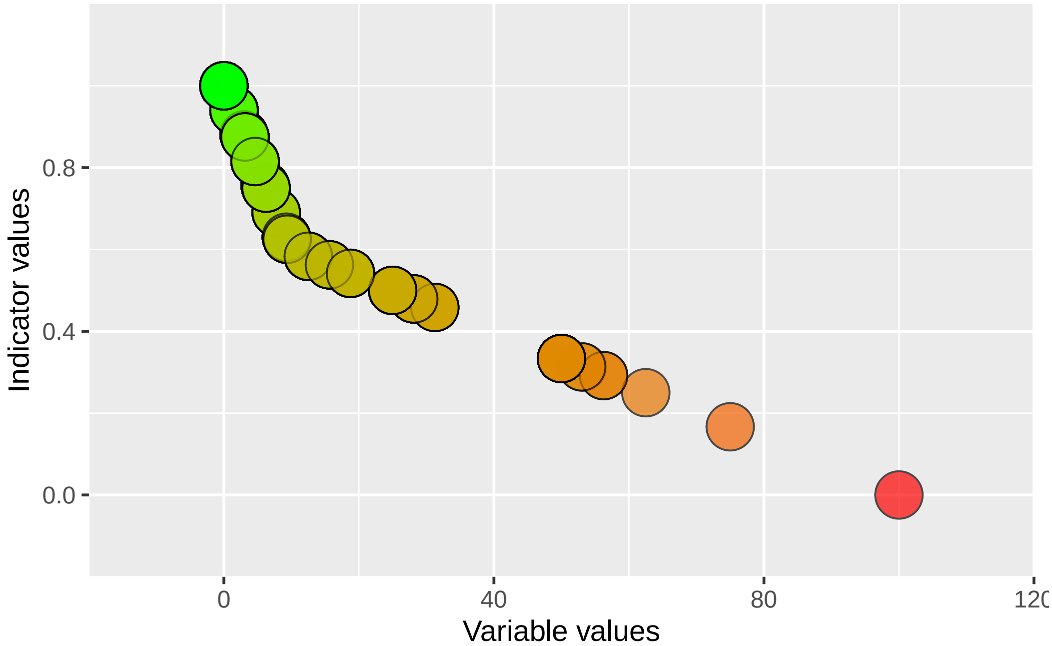



```
# Now I will copy these ADVS-values into the sf object again, keeping things in wide format
naturetypes <- naturetypes |>
  pivot_wider(
    names_from = "variable",
    values_from = "value"
  ) |>
  left_join(naturetypes_wide |> select(id, ADSV), by = "id") |>
  select(!c("7TK", "7SE", "PRSL", "PRTK"))

head(naturetypes)
```

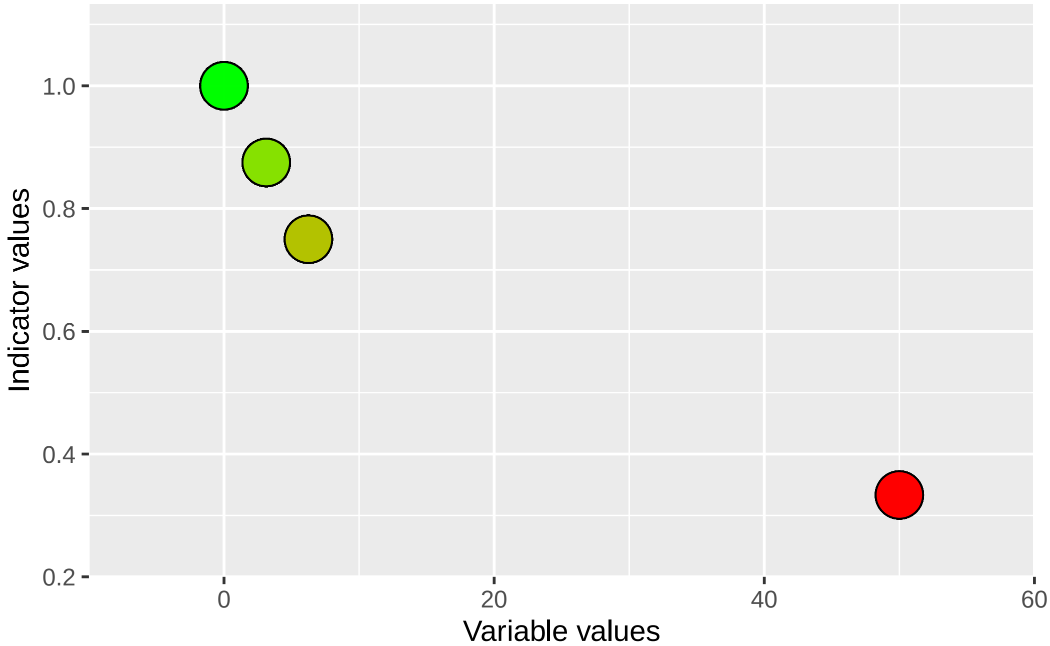
```
Simple feature collection with 6 features and 12 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 285378.4 ymin: 6611808 xmax: 811876.8 ymax: 7623720
Projected CRS: ETRS89 / UTM zone 32N
# A tibble: 6 x 13
```

```
  id         municipality year  mosaic quality biodiversity condition natureType
  <chr>      <chr>        <chr> <chr>  <chr>   <chr>        <chr>     <chr>
1 NINFP2310~ 1508         2023  nei    lavKva~ "lite"       dårlig    Sørlig ne~
2 NINFP2010~ 1848         2020  nei    modera~ "lite"       god       Terrengde~
3 NINFP1910~ 5512         2019  nei    modera~ "lite"       god       Rik åpen ~
4 NINFP2010~ 5054         2020  nei    sværtL~ ""           sværtRed~ Semi-natu~
5 NINFP2010~ 4613         2020  nei    modera~ "moderat"    moderat   Rik åpen ~
6 NINFP1910~ 5014         2019  nei    modera~ "lite"       god       Sørlig ne~
# i 5 more variables: area [m^2], SHAPE <MULTIPOLYGON [m]>, `7GR-GI` <dbl>,
#   `7FA` <dbl>, ADSV <dbl>
```

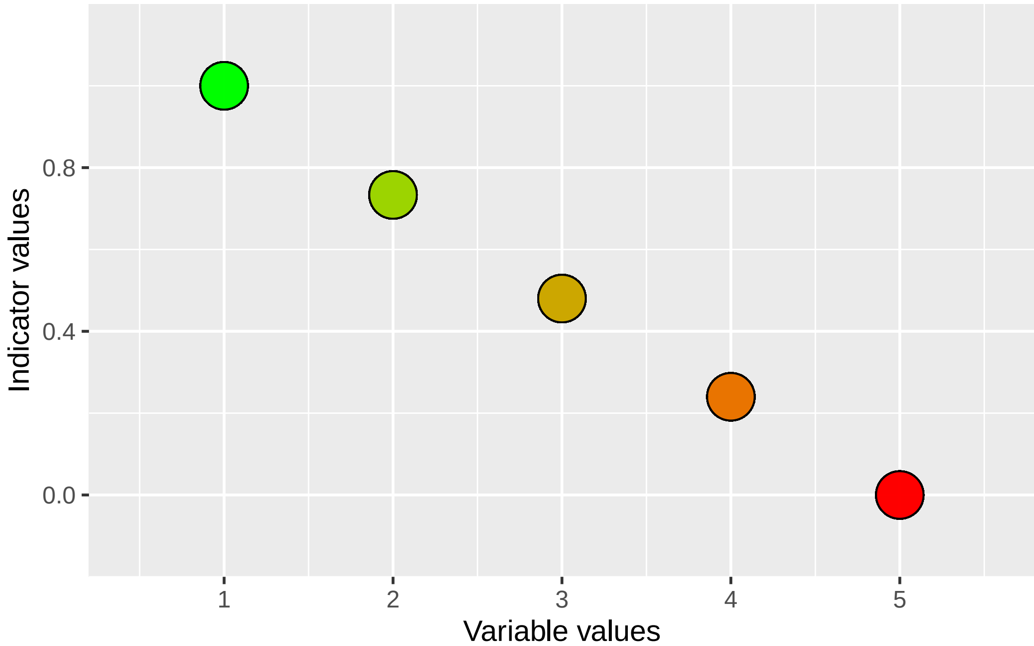
```r
# Now I rescale the now continuous variables using reference and threshold values
# I will use the same reference levels/values for all of Norway for ADSV and alien species:

upper <- 0
lower <- 100
threshold <- 10

# For 7GR-GI I use this
upper2 <- 1
lower2 <- 5
threshold2 <- 2.5 # = observable effect. Value 3 indicates a shift to a new type (grunntype)


eaTools::ea_normalise(data = naturetypes,
  vector = "ADSV",
  upper_reference_level = lower,
  lower_reference_level = upper,
  break_point = threshold,
  plot=T,
  reverse = T
  )
```




```
# There is no point yet making this a time series
# I will assign all the indicator value to the same time (2018-2022)

# same for 7FK
eaTools::ea_normalise(data = naturetypes,
  vector = "7FA",
  upper_reference_level = lower,
  lower_reference_level = upper,
  break_point = threshold,
  plot=T,
  reverse = T
  )
```




```
# The variables are really coarse

eaTools::ea_normalise(data = naturetypes,
  vector = "7GR-GI",
  upper_reference_level = lower2,
  lower_reference_level = upper2,
  break_point = threshold2,
  plot=T,
  reverse = T
  )
```



94

```
# Adding scaled indicator values to the dataset
# Same code as above, but with plot=F.
naturetypes$i_ADSV <- eaTools::ea_normalise(
  data = naturetypes,
  vector = "ADSV",
  upper_reference_level = lower,
  lower_reference_level = upper,
  break_point = threshold,
  reverse = T
)

naturetypes$i_alien <- eaTools::ea_normalise(
  data = naturetypes,
  vector = "7FA",
  upper_reference_level = lower,
  lower_reference_level = upper,
  break_point = threshold,
  reverse = T
)

naturetypes$i_ditch <- eaTools::ea_normalise(
  data = naturetypes,
  vector = "7GR-GI",
  upper_reference_level = lower2,
  lower_reference_level = upper2,
  break_point = threshold2,
  reverse = T
)
```

```r
# Preparing the outlines for the three municipalieties

# The data contains some multisurfaces
# table(st_geometry_type(muni))
# Here is a function to make sure that multipolygons are returned
ensure_multipolygons <- function(X) {
  tmp1 <- tempfile(fileext = ".gpkg")
  tmp2 <- tempfile(fileext = ".gpkg")
  st_write(X, tmp1)
  gdalUtilities::ogr2ogr(tmp1, tmp2, f = "GPKG", nlt = "MULTIPOLYGON")
  Y <- st_read(tmp2)
  st_sf(st_drop_geometry(X), geom = st_geometry(Y))
}

muni <- ensure_multipolygons(muni)
```

```
Writing layer `file399445a57d0' to data source
  `C:\Users\ANDERS~1.KOL\AppData\Local\Temp\Rtmp27B1xV\file399445a57d0.gpkg' using driver `GPKG'
Writing 363 features with 13 fields and geometry type Unknown (any).
Reading layer `file399445a57d0' from data source
  `C:\Users\anders.kolstad\AppData\Local\Temp\Rtmp27B1xV\file39945612674c.gpkg'
  using driver `GPKG'
Simple feature collection with 363 features and 13 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 234758.4 ymin: 6402776 xmax: 1339069 ymax: 8020511
Projected CRS: ETRS89 / UTM zone 32N
```

```r
# table(st_geometry_type(muni)) #OK

# subset of the three target municipalities
muni3 <- muni |>
  filter(kommunenummer %in% c(
    "3020", # Nordre Follo
    "3451", # Nord-Aurdal
    "3446" # Gran
  )) |>
  mutate(Municipality = case_when(
    kommunenummer == "3020" ~ "Nordre Follo",
    kommunenummer == "3451" ~ "Nord-Aurdal",
    kommunenummer == "3446" ~ "Gran"
  ))

# To crop EDM, I need the three municipalities seprately.
nf <- muni3 |>
  filter(kommunenummer == "3020")
na <- muni3 |>
  filter(kommunenummer == "3451")
gr <- muni3 |>
  filter(kommunenummer == "3446")
```

```r
# I need to intersect the naturetypes data with the municipalities
nature3 <- naturetypes |>
  st_intersection(muni3)

nature3 |>
  as_tibble() |>
  count(municipality,
    sort = TRUE,
    name = "Number of polygons")
```

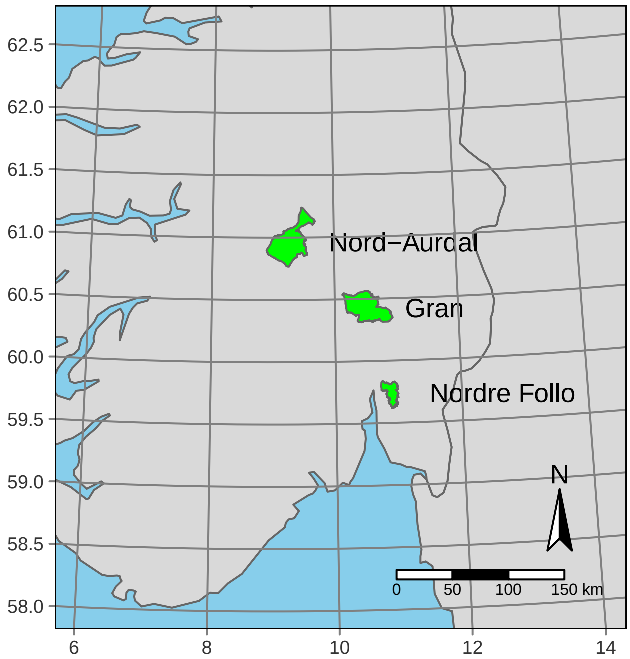
```
# A tibble: 5 x 2
  municipality `Number of polygons`
  <chr>                       <int>
1 3451                          236
2 3446                           54
3 3207                            5
4 3234,3446                       2
5 3207,3218                       1
```

```r
# There where some polygons that spanned municipal borders.
# It's not a problem

# and also to get the data coverage polygon.
coverage3 <- coverage |>
  st_intersection(muni3)
```

```r
# Simplified coastline / terrestrial area
terrestrial <- outline |>
  st_intersection(muni3)

# Polygons for the oceans in each municipality
ocean <- muni3 |>
  st_difference(outline)

# calculate stats - terrestrial area
terrestrial <- terrestrial |>
  mutate(
    area_t = geometry |> st_area(),
    t_area_km =
      round(units::drop_units(area_t * 1e-6))
  )
```

```r
# Make map to show where the three municipalities are
world <- ne_countries(scale = "medium", returnclass = "sf") |>
  st_transform(myCRS) |>
  filter(admin %in% c("Norway", "Sweden")) |>
  st_make_valid()

# get centroids
centroids <- muni3 |>
  st_centroid()
```

```
inc <- 200000
myBbox <- st_bbox(centroids)
myBbox[1:2] <- myBbox[1:2]-inc
myBbox[3:4] <- myBbox[3:4]+inc

(positionMap <-
  tm_shape(world,
           bbox = myBbox) +
    tm_polygons() +
  tm_shape(muni3) +
    tm_polygons(col = "green") +
  tm_shape(centroids) +
  tm_text(
    text = "Municipality",
    just= "left",
    size = .8,
    xmod = 1,
    ymod = 0
  ) +
  tm_grid(projection = 4326) +
  tm_layout(
    bg.color = "skyblue",
    outer.margins = c(0.01, .02, .02, .02))+
  tm_compass()+
  tm_scale_bar()
)
```




```
# what is the distance between Nordre Follo and Nord-Aurdal
(km_distance <- centroids |>
  st_distance() |>
```

```r
  max() |>
  set_units("km") |>
  drop_units() |>
  round())
```

```
[1] 162
```

```r
# I first tried to import and crop the mire data using stars,
# but that failed (see pre 21 feb 2023).
# Trying { terra } instead

# convert municipal outline to vect via st
nf_vect <- as(nf, "Spatial") |>
  terra::vect()
gr_vect <- as(gr, "Spatial") |>
  terra::vect()
na_vect <- as(na, "Spatial") |>
  terra::vect()

# crop and mask (very fast!)
mire_terra_nf <- mire_terra |>
  terra::crop(nf_vect) |>
  terra::mask(nf_vect)

mire_terra_gr <- mire_terra |>
  terra::crop(gr_vect) |>
  terra::mask(gr_vect)

mire_terra_na <- mire_terra |>
  terra::crop(na_vect) |>
  terra::mask(na_vect)

# Plot to check overlap
# ggplot()+
#  geom_spatraster(data = mire_nf_terra)+
#  geom_spatvector(data = nf_vect,
#                  fill = NA)
# The cropping and masking worked.

# # I like the stars, sf and tmap combo better, so I return to stars
mire_stars_nf <- mire_terra_nf |>
  st_as_stars()
mire_stars_gr <- mire_terra_gr |>
  st_as_stars()
mire_stars_na <- mire_terra_na |>
  st_as_stars()

par(mfrow=c(3,1))
plot(mire_stars_nf)
plot(mire_stars_gr)
plot(mire_stars_na)
```

```
saveRDS(mire_stars_nf, "manual_cache/mire_stars_nf.RDS")
saveRDS(mire_stars_gr, "manual_cache/mire_stars_gr.RDS")
saveRDS(mire_stars_na, "manual_cache/mire_stars_na.RDS")
```

```
mire_stars_nf <- readRDS("manual_cache/mire_stars_nf.RDS")
mire_stars_gr <- readRDS("manual_cache/mire_stars_gr.RDS")
mire_stars_na <- readRDS("manual_cache/mire_stars_na.RDS")
mire_terra_nf <- rast(mire_stars_nf)
mire_terra_gr <- rast(mire_stars_gr)
mire_terra_na <- rast(mire_stars_na)
```

```
# calculate area of survey coverage maps
# values goes into summary table in the ms
dk2 <- coverage3 |>
  group_by(Municipality) |>
  summarise(SHAPE = st_union(SHAPE)) |>
  mutate(
    dk_area_km = SHAPE |> st_area(),
    dk_area_km = round(units::drop_units(dk_area_km * 1e-6))
  )
```

```
# calculate area of the mires in each municipality
# -- Nordre Follo
(mireArea <- mire_terra_nf |>
  global(c("mean", "sum"), na.rm = T) |>
  add_column("Municipality" = "Nordre Follo") |>
  mutate(
    mirePercent = round(mean * 100, 1),
    mire_km2 = sum / 1e+4
  ))
```

```
              mean  sum Municipality mirePercent mire_km2
Myr153 0.002926816 6202 Nordre Follo         0.3   0.6202
```

```
# -- Gran
mireArea2 <- mire_terra_gr |>
  global(c("mean", "sum"), na.rm = T) |>
  add_column("Municipality" = "Gran") |>
  mutate(
    mirePercent = round(mean * 100, 1),
    mire_km2 = sum / 1e+4
  )

# -- Nord-Aurdal
mireArea3 <- mire_terra_na |>
  global(c("mean", "sum"), na.rm = T) |>
  add_column("Municipality" = "Nord-Aurdal") |>
  mutate(
    mirePercent = round(mean * 100, 1),
    mire_km2 = sum / 1e+4
  )

mireArea <- mireArea |>
```

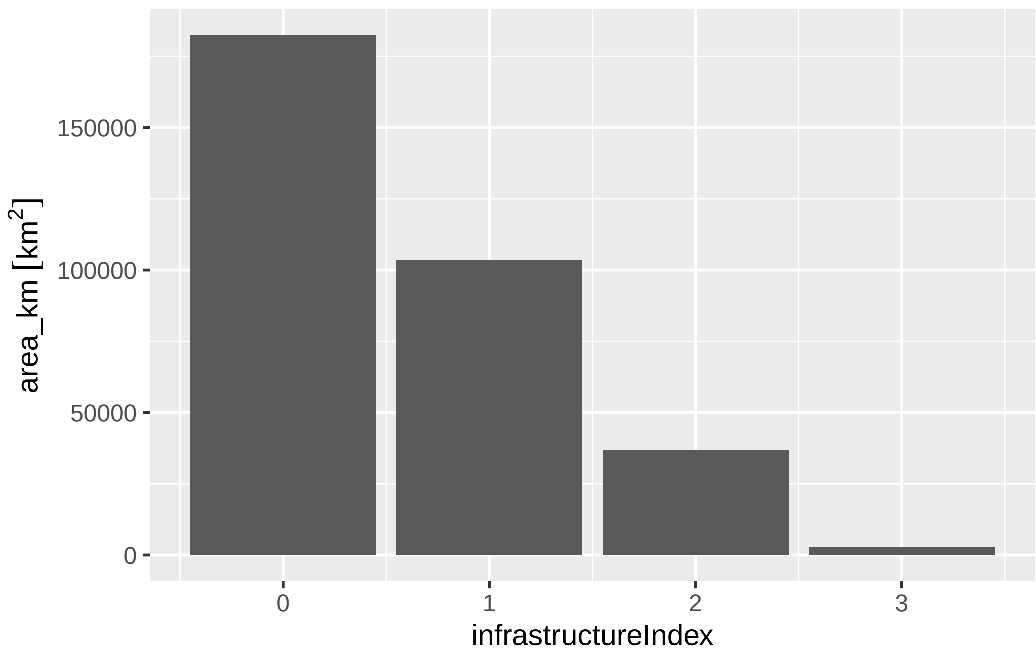
```
  rbind(mireArea2, mireArea3)

# Calculate the area of mire inside the coverage maps
# -- Nordre Follo
mire_in_dk <- mire_terra_nf |>
  terra::mask(dk2 |> filter(Municipality == "Nordre Follo")) |>
  global("sum", na.rm = T) |>
  mutate(mireInSurvey_km2 = sum / 1e+4) |>
  add_column(Municipality = "Nordre Follo")

mire_in_dk2 <- mire_terra_gr |>
  terra::mask(dk2 |> filter(Municipality == "Gran")) |>
  global("sum", na.rm = T) |>
  mutate(mireInSurvey_km2 = sum / 1e+4) |>
  add_column(Municipality = "Gran")
mire_in_dk3 <- mire_terra_na |>
  terra::mask(dk2 |> filter(Municipality == "Nord-Aurdal")) |>
  global("sum", na.rm = T) |>
  mutate(mireInSurvey_km2 = sum / 1e+4) |>
  add_column(Municipality = "Nord-Aurdal")

mire_in_dk <- mire_in_dk |>
  rbind(mire_in_dk2, mire_in_dk3)
```

```
# this data is on a 100x100m grid
infra <- infra |>
  # , which is more then we need - warp it to 1x1km
  st_warp(
    cellsize = c(1000, 1000),
    crs = st_crs(nf),
    use_gdal = TRUE,
    method = "average"
  ) |>
  setNames("infrastructureIndex") |>
  st_transform(myCRS) |>
  mutate(infrastructureIndex = case_when(
    infrastructureIndex < 1 ~ 0,
    infrastructureIndex < 6 ~ 1,
    infrastructureIndex < 12 ~ 2,
    infrastructureIndex >= 12 ~ 3
  )) |>
  # taking away point in the sea
  st_crop(outline)

# This step might seem rather stupid. We want to vectorize a rather large
# raster. This makes it a quite big data object. The reason is that there is no
# really good way to burn polygon data on to raster grid cells after the disuse
# of the raster package. It was not straight forward then either. But
# calculating intersections between polygons is very fast and easy.

infra <- eaTools::ea_homogeneous_area(infra,
```

```
  groups = infrastructureIndex
)

saveRDS(infra, paste0(path_temp, "infrastructureIndex_discrete_vectorized.rds"))
```

```
# read cached vetorized infrastructure data
infra <- readRDS(paste0(path_temp, "infrastructureIndex_discrete_vectorized.rds"))
```

```
# Calculate area
infra <- infra |>
  mutate(
    area = geometry |> st_area(),
    area_km = area |> set_units("km2")
  )

# show the summered area per HIA
infra |>
  as_tibble() |>
  group_by(infrastructureIndex) |>
  summarise(area_km = sum(area_km)) |>
  ggplot(aes(
    x = infrastructureIndex,
    y = area_km
  )) +
  geom_col()
```



118

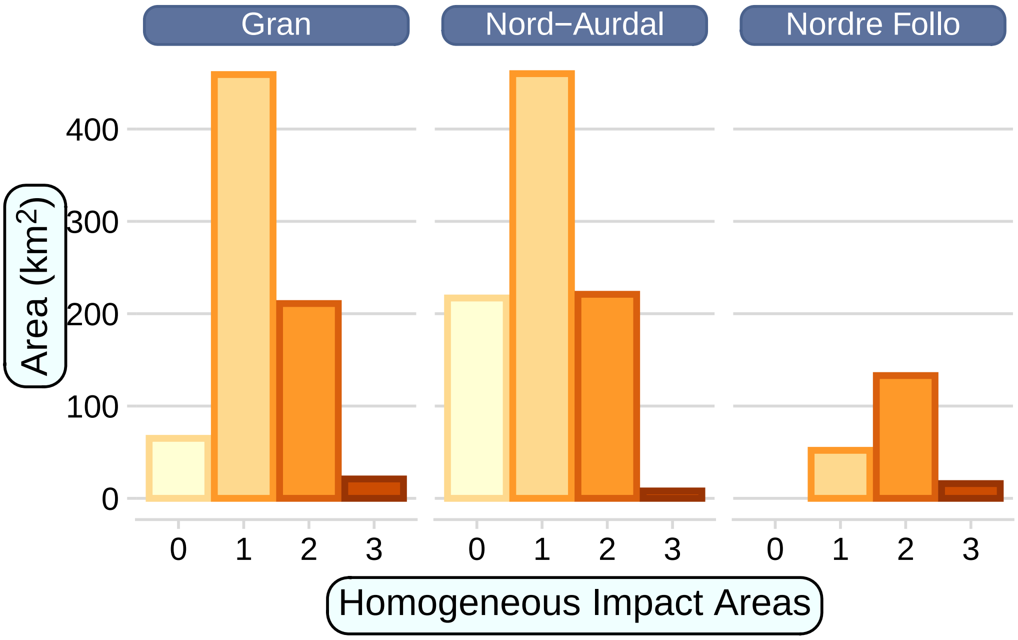
```
# intersect with the three municipalities
# and calculate area
infraMuni3 <- infra |>
  st_intersection(muni3) |>
```

```r
  mutate(area = geometry |> st_area())

# Turn m2 into km2
# and sum the total area per HIA
(infraMuni3_tbl <- infraMuni3 |>
  as.data.frame() |>
  mutate(area_HIA_km2 = units::drop_units(area) * 1e-6) |>
  group_by(Municipality, infrastructureIndex) |>
  summarise(total_area_HIAs_km2 = round(sum(area_HIA_km2))))
```

```
# A tibble: 11 x 3
# Groups:   Municipality [3]
   Municipality infrastructureIndex total_area_HIAs_km2
   <chr>                      <dbl>               <dbl>
 1 Gran                           0                  65
 2 Gran                           1                 459
 3 Gran                           2                 211
 4 Gran                           3                  21
 5 Nord-Aurdal                    0                 217
 6 Nord-Aurdal                    1                 460
 7 Nord-Aurdal                    2                 221
 8 Nord-Aurdal                    3                   8
 9 Nordre Follo                   1                  52
10 Nordre Follo                   2                 133
11 Nordre Follo                   3                  16
```

```r
# Calculate the area weighted mean HIA value per municipality
infraMuni3_summary <- infraMuni3_tbl |>
  group_by(Municipality) |>
  summarise(
    meanHIA =
      round(
        weighted.mean(
          infrastructureIndex, total_area_HIAs_km2
        ), 2
      )
  )

# Make a plot to check that it has worked
(infra_dist_plot <- infraMuni3_tbl |>
  ggplot() +
  geom_bar(
    aes(
      x = infrastructureIndex,
      y = total_area_HIAs_km2,
      fill = factor(infrastructureIndex),
      colour = factor(infrastructureIndex)
    ),
    stat = "identity",
    lwd = 1.2
  ) +
  scale_fill_manual(values = RColorBrewer::brewer.pal(4, "YlOrBr")) +
```

```
  scale_color_manual(values = RColorBrewer::brewer.pal(5, "YlOrBr")[-1]) +
  theme_minimal_hgrid() +
  labs(
    x = "Homogeneous Impact Areas",
    y = "Area (km<sup>2</sup>)"
  ) +
  theme(
    axis.title.x = element_textbox_simple(
      width = NULL,
      padding = margin(4, 4, 4, 4),
      margin = margin(4, 0, 0, 0),
      linetype = 1,
      r = grid::unit(8, "pt"),
      fill = "azure1"
    ),
    axis.title.y = element_textbox_simple(
      width = NULL,
      padding = margin(4, 4, 4, 4),
      margin = margin(4, 0, 0, 0),
      linetype = 1,
      orientation = "left-rotated",
      r = grid::unit(8, "pt"),
      fill = "azure1"
    ),
    strip.background = element_blank(),
    strip.text = element_textbox(
      size = 12,
      color = "white", fill = "#5D729D", box.color = "#4A618C",
      halign = 0.5, linetype = 1, r = unit(5, "pt"), width = unit(1, "npc"),
      padding = margin(2, 0, 1, 0), margin = margin(3, 3, 3, 3)
    )
  ) +
  guides(fill = "none", colour = "none") +
  #scale_y_log10() +
  facet_grid(cols = vars(Municipality))
)
```



134

```
#ggsave(plot = infra_dist_plot,
#       "../images/infra-dist-plot.jpg")
```

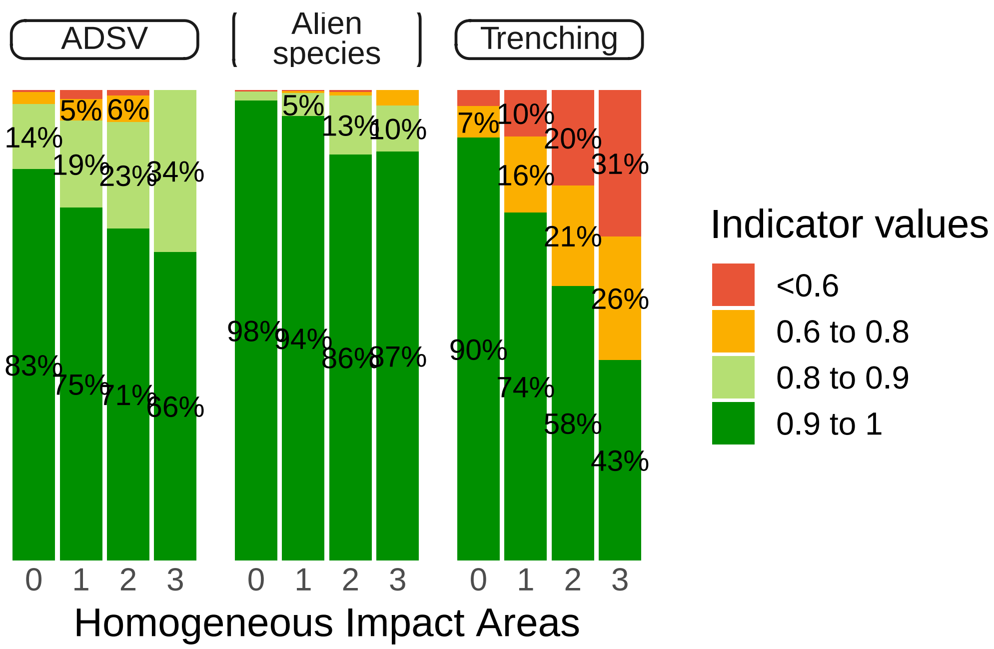
```
# Now I want to see if the indicator values depend on the HIA is a predictable
# way to justify the stratification
corrCheck <- st_intersection(naturetypes, infra)
```

```
# A first look
corrCheck |>
  mutate(
    i_ADSV_fct = floor(round(i_ADSV * 10, 2)) / 10,
    i_alien_fct = floor(round(i_alien * 10, 2)) / 10,
    i_ditch_fct = floor(round(i_ditch * 10, 2)) / 10
  ) |>
  pivot_longer(
    cols = c(i_ADSV_fct, i_alien_fct, i_ditch_fct),
    values_to = "indicatorValue",
    names_to = "indicator",
    values_drop_na = T
  ) |>
  ggplot(aes(
    x = factor(infrastructureIndex),
    fill = factor(indicatorValue)
  )) +
  geom_bar(
    position = "fill"
  ) +
  theme_bw(base_size = 12) +
  guides(fill = guide_legend("Scaled indicator values")) +
  ylab("Fraction of data points") +
```

```
  xlab("HIA") +
  scale_fill_brewer(palette = "RdYlGn") +
  facet_grid(indicator ~ year)

# After this I also tried chaning the color gradient,
# and the number of categories. I tries discrete colors and log-transformation.
# These variants are interpretted slightly differetly by the brain.
# See versions pre 27.02.2023
```

```
# Using a color gradient emphasizes the first color (dark green).
# Lets try discrete colors, and merge some classes to simplify
(validationPlot <- corrCheck |>
  pivot_longer(
    cols = c(i_ADSV, i_alien, i_ditch),
    values_to = "indicatorValue",
    names_to = "indicator",
    values_drop_na = T
  ) |>
  mutate(
    condition = case_when(
      indicatorValue < 0.6 ~ "<0.6",
      indicatorValue < 0.8 ~ "0.6 to 0.8",
      indicatorValue < 0.91 ~ "0.8 to 0.9",
      .default = "0.9 to 1"
    ),
    condition = fct_reorder(condition, indicatorValue),
    indicator = case_when(
      indicator == "i_ADSV" ~ "ADSV",
      indicator == "i_alien" ~ "Alien species",
      indicator == "i_ditch" ~ "Trenching"
    )
  ) |>
  as_tibble() |>
  group_by(indicator, infrastructureIndex, condition) |>
  summarise(n = n()) |>
  ungroup() |>
  group_by(indicator, infrastructureIndex) |>
  mutate(lab = round(n/sum(n)*100),
         lab = case_when(
           lab < 5 ~ NA,
           .default = paste0(lab, "%")
         )) |>
  ggplot(aes(
    x = infrastructureIndex,
    y = n,
    fill = condition
  )) +
  geom_bar(
    position = "fill",
    stat = "identity"
  ) +
```

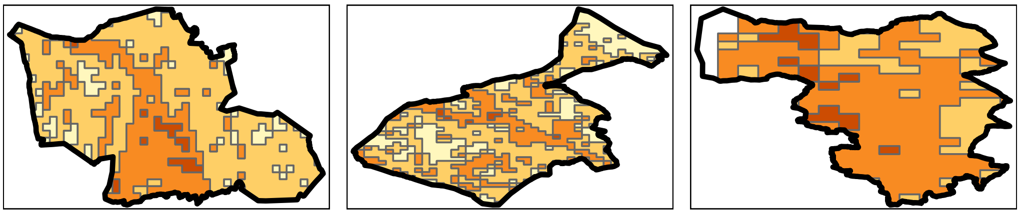
```
  geom_text(aes(label = lab),
    position = position_fill(vjust = 0.5),
    color= "black", vjust = 0.5, size = 4) +
  theme_minimal(base_size = 15) +
  theme(
    panel.grid = element_blank(),
    axis.text.x = element_text(margin = margin(t = -10)),
    axis.text.y = element_blank(),
    axis.title.y = element_blank(),
    strip.text = element_textbox(
      size = 12,
      halign = 0.5, linetype = 1, r = unit(5, "pt"), width = unit(1, "npc"),
      padding = margin(2, 0, 1, 0), margin = margin(3, 3, 3, 3))) +
  guides(fill = guide_legend("Indicator values")) +
  xlab("Homogeneous Impact Areas") +
  scale_fill_manual(values = c("#E85437","#FBAF00", "#B5DF73", "#009000")) +
  facet_grid(~indicator)
)
```




```
ggsave("../images/validation-plot.jpg",
       plot=validationPlot,
       width = 9,
  height = 5)
```

```
# Infratructure in each municipality
(infraMuniMap <- tm_shape(muni3) +
  tm_borders() +
  tm_shape(infraMuni3) +
  tm_polygons(
    col = "infrastructureIndex",
```

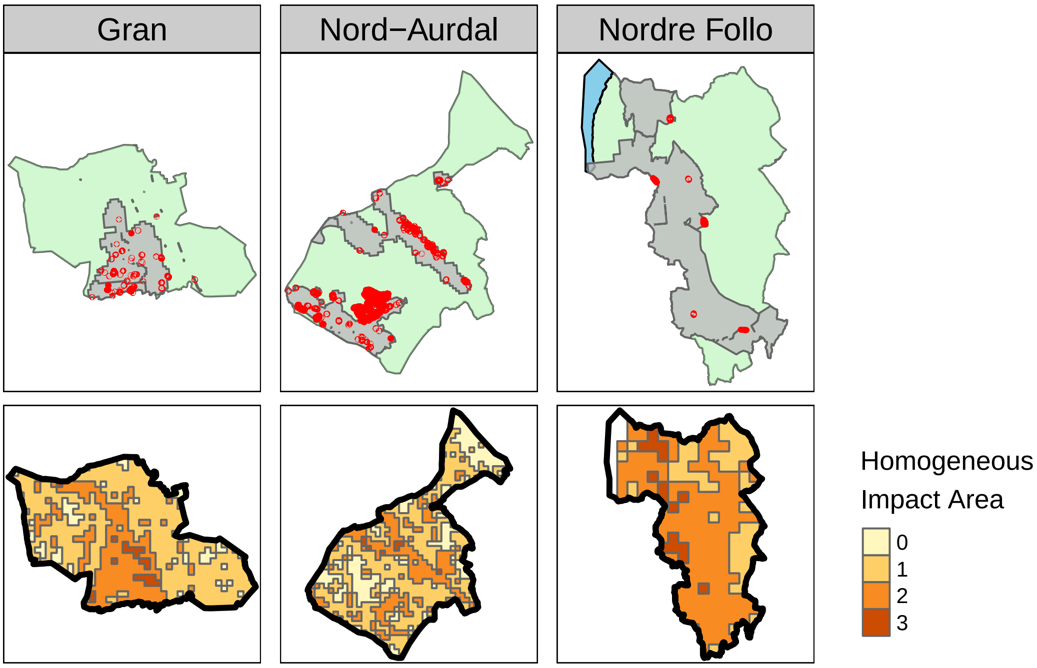
```
    style = "cat",
    title = "Homogeneous Impact Areas"
  ) +
  tm_layout(
    legend.show = F,
    panel.label.height = 0) +
  tm_shape(muni3) +
  tm_borders(lwd = 3, col = "black") +
  tm_facets(by = "Municipality"))
```



136

```
# A figure with just the legend
infraMuniMap_l <- tm_shape(muni3) +
  tm_borders() +
  tm_shape(infraMuni3) +
  tm_polygons(
    col = "infrastructureIndex",
    style = "cat",
    title = "Homogeneous\nImpact Area"
  ) +
  tm_layout(legend.only = TRUE,
            legend.position = c("left", "bottom"),
            legend.outside = F)


empty <- tm_shape(muni3) +
  tm_borders(col="white") +
  tm_layout(frame = F)

# Survey coverage and other colour overlayed the municipalities
muniPlot <- tm_shape(muni3) +
```

```
  tm_borders() +
  tm_facets(
    by = "Municipality",
    ncol = 3
  ) +
  tm_shape(terrestrial) +
  tm_fill(
    col = "lightgreen",
    alpha = .4
  ) +
  tm_shape(ocean) +
  tm_polygons(
    col = "skyblue",
    border.col = "black"
  ) +
  tm_shape(dk2) +
  tm_polygons(
    col = "grey",
    alpha = .8
  ) +
  tm_shape(nature3) +
  tm_polygons(
    col = "red",
    border.col = "red",
    lwd = 3
  )



(methodsMap <- tmap_arrange(
  muniPlot,
  empty,
  infraMuniMap,
  infraMuniMap_l,
  ncol = 2,
  widths = c(.8, .2),
  heights = c(.6, .4),
  outer.margins = NULL
))
```



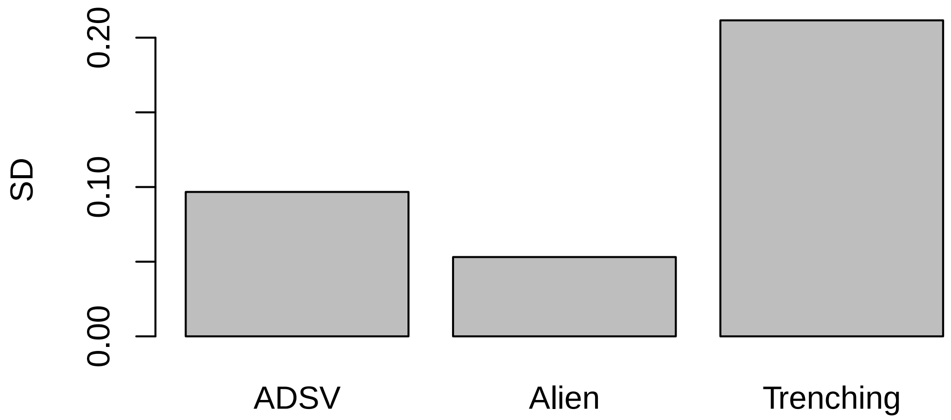
137

```
#tmap_save(methodsMap, "../images/studyLocations.tiff",
#           dpi= 1000,
#           units = "cm",
#           width = 18,
#           height = 10)
#tmap_save(methodsMap, "../images/studyLocations.jpg",
#           units = "cm",
#           width = 18,
#           height = 10)
#
#saveRDS(methodsMap, "../figures/studyLocation.RDS")
```

```
# Calculate some more stats for the table with municipality stats

# Number of mire polygons per municipality
nature3_tbl <- nature3 |>
  group_by(Municipality) |>
  summarise(n = n()) |>
  as_tibble()

# Take muni3 and add all sorts of other data to it
# using left_join.
muni_tbl <- muni3 |>
  # calculate total area
  mutate(
    area_km =
      round(
        units::drop_units(
          geom |> st_area() * 1e-6
        )
```

```
      )
  ) |>
  # make tibble for the coming join
  as_tibble() |>
  # paste inn terrestrial area
  left_join(
    terrestrial |>
      as_tibble() |>
      select(kommunenummer, t_area_km),
    keep = F
  ) |>
  # area of survey
  left_join(dk2 |> select(Municipality, dk_area_km)) |>
  mutate(dk_percent = round((dk_area_km / t_area_km) * 100)) |>
  # number of polygons
  left_join(nature3_tbl |> select(Municipality, n)) |>
  # total mire area and %
  left_join(mireArea |> select(Municipality, mirePercent, mire_km2)) |>
  # % mire inside survey coverage map
  left_join(mire_in_dk |> select(Municipality, mireInSurvey_km2)) |>
  mutate(mireInSurvery_percent = round(mireInSurvey_km2 / mire_km2 * 100, 2)) |>
  left_join(infraMuni3_summary)
```

```
# Now I need to spatially (horizontally) aggregate the indicator values for each
# HIA. Rather than taking the arithmetic mean, I will use a Bayesian updating
# approach. The point estimate (central tendency) will probably be the same
# more or less, with the two approaches, but with the updating approach I can
# get a honest measure for the uncertainty even with very small sample sizes.
# For this I need a 'true' value for the variation in the indicator
# I will use the entire national data set to determine this number

national_sd_ADSV <- sd(naturetypes$i_ADSV, na.rm=T)
national_sd_alien <- sd(naturetypes$i_alien, na.rm=T)
national_sd_ditch <- sd(naturetypes$i_ditch, na.rm=T)

barplot(c(national_sd_ADSV, national_sd_alien, national_sd_ditch),
        names.arg = c("ADSV", "Alien", "Trenching"),
        ylab="SD")
```



138

```
# The figure shows that the Trenching indicator is more spatially variable
```

```
# This function is modified from Bolstad::normdp
# It updates a flat prior based on a sample of values (indicator values)
# and weight (polygon area) and returns a distribution for the weighted mean
# that has a gaussian distribution. It assumes the sampled population is also
# Gaussian. The variance is estimated from the full sample of indicator values.


wgt_mean <- function(x, weights,
                     sigma.x = NULL,
                     mu = seq(0, 1, length.out=1000),
                     mu.prior = rep(1/length(mu), times=length(mu)),
                     stat = "mean",
                     ...) {

  mx <- weighted.mean(x, weights)
  if (round(sum(mu.prior), 7) != 1) {
    warning("The prior probabilities did not sum to 1, therefore the prior has been normalized")
    mu.prior <- mu.prior / sum(mu.prior)
  }
  n.mu <- length(mu)
  nx <- length(x)
  snx <- sigma.x^2 / nx
  likelihood <- exp(-0.5 * (mx - mu)^2 / snx)
  posterior <- likelihood * mu.prior / sum(likelihood * mu.prior)
  mx <- sum(mu * posterior)
  vx <- sum((mu - mx)^2 * posterior)

  # draw 1k samples from the posterior to calculate the quantiles and sd from
```

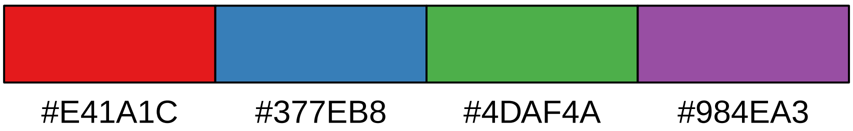
```
  sample <- sample(mu, size = 1000, prob = posterior, replace = T)
  lower <- quantile(sample, probs = 0.025)
  upper <- quantile(sample, probs = 0.975)
  # we can assume that the distrubutin for the mean is gaussian, so we take a
  # symetrical sd from here and use it, along with the mean, to recreate a
  # normal distribution later which we can sample from
  sdx <- sd(sample)
  results <- list(
    name = "mu",
    param.x = mu,
    prior = mu.prior,
    likelihood = likelihood,
    posterior = posterior,
    weighted_mean = mx,
    var = vx
  )
  if(stat == "mean") return(mx)
  if(stat == "lower") return(lower)
  if(stat == "upper") return(upper)
  if(stat == "sd") return(sdx)
}
```

```
# Intersect nature3 with the HIA
stats_tbl <- nature3 |>
  st_intersection(infraMuni3) |>
  pivot_longer(cols = c(
    i_ADSV,
    i_alien,
    i_ditch),
    names_to = "indicator",
    values_to = "indicatorValue") |>
  filter(!is.na(indicatorValue)) |>
  mutate(area = drop_units(area)) |>
  group_by(indicator, Municipality, infrastructureIndex) |>
  summarise(sd = wgt_mean(indicatorValue,
      weights = area,
      sigma.x = case_when(
        indicator == "i_alien" ~ national_sd_alien,
        indicator == "i_ditch" ~ national_sd_ditch,
        indicator == "i_ADSV" ~ national_sd_ADSV,
        .default = NULL
      ),
      stat = "sd"),
    mean = wgt_mean(indicatorValue,
      weights = area,
      sigma.x = case_when(
        indicator == "i_alien" ~ national_sd_alien,
        indicator == "i_ditch" ~ national_sd_ditch,
        indicator == "i_ADSV" ~ national_sd_ADSV,
        .default = NULL
      ),
```

```r
      stat = "mean"),
    n = n())
stats_tbl
```

```
Simple feature collection with 20 features and 6 fields
Geometry type: GEOMETRY
Dimension:     XY
Bounding box:  xmin: 494489.4 ymin: 6614554 xmax: 608777.4 ymax: 6770831
Projected CRS: ETRS89 / UTM zone 32N
# A tibble: 20 x 7
# Groups:   indicator, Municipality [9]
   indicator Municipality infrastructureIndex      sd  mean     n
   <chr>     <chr>                      <dbl>   <dbl> <dbl> <int>
 1 i_ADSV    Gran                           1 0.0279  0.783    12
 2 i_ADSV    Gran                           2 0.0226  0.916    18
 3 i_ADSV    Nord-Aurdal                    0 0.0266  0.836    13
 4 i_ADSV    Nord-Aurdal                    1 0.00837 0.867   145
 5 i_ADSV    Nord-Aurdal                    2 0.0111  0.833    75
 6 i_ADSV    Nordre Follo                   2 0.0375  0.949     3
 7 i_ADSV    Nordre Follo                   3 0.0590  0.923     1
 8 i_alien   Gran                           1 0.0196  0.930     7
 9 i_alien   Gran                           2 0.00920 0.935    33
10 i_alien   Nord-Aurdal                    1 0.0212  0.938     6
11 i_alien   Nord-Aurdal                    2 0.0102  0.979    27
12 i_alien   Nordre Follo                   2 0.0300  0.898     3
13 i_alien   Nordre Follo                   3 0.0365  0.898     2
14 i_ditch   Gran                           1 0.0606  0.859    12
15 i_ditch   Gran                           2 0.0528  0.636    17
16 i_ditch   Nord-Aurdal                    0 0.0453  0.932    13
17 i_ditch   Nord-Aurdal                    1 0.0177  0.858   145
18 i_ditch   Nord-Aurdal                    2 0.0252  0.900    75
19 i_ditch   Nordre Follo                   2 0.115   0.244     3
20 i_ditch   Nordre Follo                   3 0.174   0.691     1
# i 1 more variable: SHAPE <GEOMETRY [m]>
```

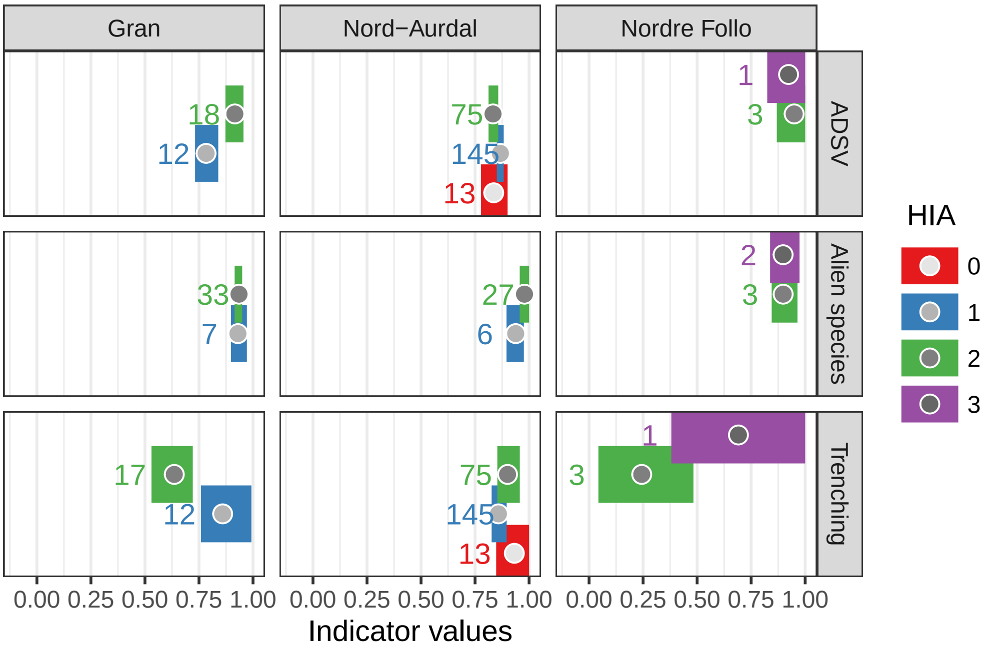
```r
# Forest plot

# #define colours for dots and bars
dotCOLS = c("grey90","grey70", "grey50", "grey40")
barCOLS <- tmaptools::get_brewer_pal("Set1", n = 4)
```



169

```
(forest_plot <- stats_tbl |>
  mutate(
    indicator = case_when(
      indicator == "i_ADSV" ~ "ADSV",
      indicator == "i_alien" ~ "Alien species",
      indicator == "i_ditch" ~ "Trenching"
    )
  ) |>
  rowwise() |>
  mutate(
    low = quantile(rnorm(200, mean, sd), probs = 0.025),
    high = quantile(rnorm(200, mean, sd), probs = 0.975),
    high = ifelse(high > 1, 1, high)
  ) |>
  ggplot(aes(x=infrastructureIndex,
             y=mean,
             ymin=low,
             ymax=high,
             col=factor(infrastructureIndex),
             fill=factor(infrastructureIndex))) +
  geom_linerange(
    size=10) +
  geom_point(
    size=3,
    shape=21,
    colour="white",
    stroke = 0.5,
    ) +
  geom_text(
    aes(y = low, label = n),
```

```
    nudge_y = -0.1,
    show.legend = F
  ) +
  scale_fill_manual(values=dotCOLS)+
  scale_color_manual(values=barCOLS)+
  scale_x_discrete(name="") +
  scale_y_continuous(name="Indicator values", limits = c(-0.1, 1)) +
  coord_flip() +
  theme_bw() +
  labs(fill = "HIA",
       col = "HIA") +
  facet_grid(indicator ~ Municipality)
)
```



170

```
#ggsave("../images/forest-plot.jpg",
#       plot=forest_plot)
```

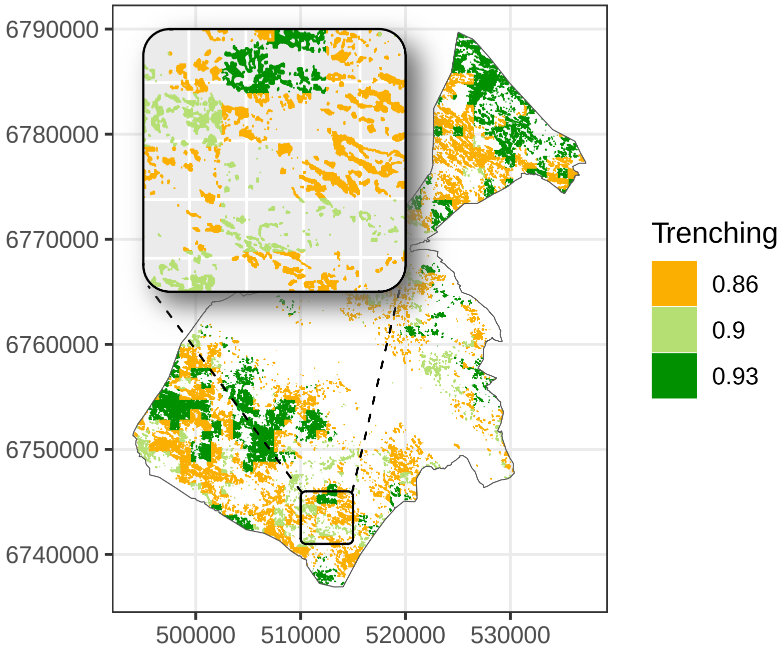
```
# Next I need to
#  - copy the weighted means and SDs over to  HIA map and then over to the
#    ecosystem delineation map polygons.
#  - show intermittent resulting map with ggmagnify
#  - sample from these distributions (ignore missing HIAs)
```

```
# Copy the weighted means and SDs over to HIA map and then over to the
# ecosystem delineation map polygons.
spread_na <- mire_stars_na |>
  st_as_sf(merge=T) |>
  filter(Myr153 == 1) |>
  st_intersection(infraMuni3 |> select(infrastructureIndex)) |>
  mutate(area = geometry |> st_area()) |>
```

```
  left_join(stats_tbl |>
              as_tibble() |>
              filter(Municipality == "Nord-Aurdal") |>
              select(mean, sd, indicator, n, infrastructureIndex, Municipality),
            by = "infrastructureIndex")

spread_nf <- mire_stars_nf |>
  st_as_sf(merge=T) |>
  filter(Myr153 == 1) |>
  st_intersection(infraMuni3 |> select(infrastructureIndex)) |>
  mutate(area = geometry |> st_area()) |>
  left_join(stats_tbl |>
              as_tibble() |>
              filter(Municipality == "Nordre Follo") |>
              select(mean, sd, indicator, n, infrastructureIndex, Municipality),
            by = "infrastructureIndex")

spread_gr <- mire_stars_gr |>
  st_as_sf(merge=T) |>
  filter(Myr153 == 1) |>
  st_intersection(infraMuni3 |> select(infrastructureIndex)) |>
  mutate(area = geometry |> st_area()) |>
  left_join(stats_tbl |>
              as_tibble() |>
              filter(Municipality == "Gran") |>
              select(mean, sd, indicator, n, infrastructureIndex, Municipality),
            by = "infrastructureIndex")
```

```
# A plot of Nord-Aurdal with the mean indicator values per HIA spread over the
# EDM
from <- c(xmin = 510000, xmax = 515000, ymin = 6741000, ymax = 6746000)
to <-   c(xmin = 495000, xmax = 520000, ymin = 6765000, ymax = 6790000)
myCols <- c(
  #"#E85437",
  "#FBAF00",
  "#B5DF73",
  "#009000"
  )


(spread_na_map <- spread_na |>
  filter(indicator == "i_ditch") |>
  mutate(Trenching = factor(round(mean, 2))) |>
  ggplot() +
  geom_sf(aes(fill=Trenching,
              color = Trenching)) +
  geom_sf(data = na,
          alpha=0) +
  scale_fill_manual(values = myCols) +
  scale_color_manual(values = myCols) +
  coord_sf(
```

```
    datum = st_crs(myCRS),
    xlim = c(494174.8 , 537114.7 ),
    ylim = c(6737092 , 6789676)) +
  ggmagnify::geom_magnify(from = from, to = to,
                          expand = 0,
                          shadow =T,
                          corners = 0.1) +
  theme_bw()
)
```



171

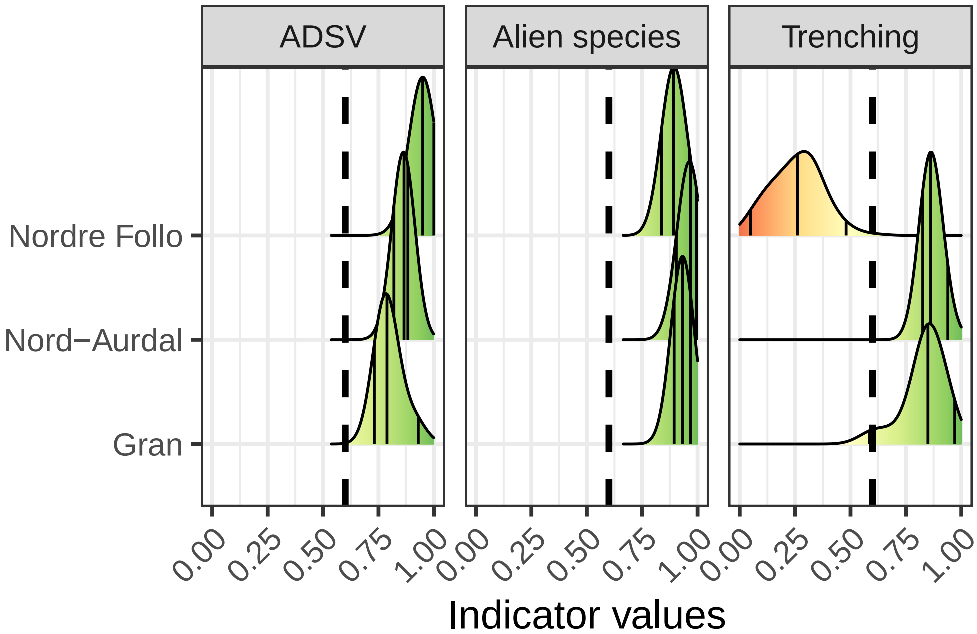
```
#ggsave("../images/spread-na.tiff",
#        plot = spread_na_map)
#ggsave("../images/spread-na-small.tiff",
#        plot = spread_na_map,
#        dpi=150)
#ggsave("../images/spread-na-small.jpg",
#        plot = spread_na_map)
```

```
# Now I sample indicator values from the EDM and create a new distribution for
# indicator values in the EAAs (the municipalities). First I sample the individual
# distributions for each mire polygon with n defined by the polygon area.
# The distribution for the polygon areas is strongly right skewed, meaning
# some polygons will contribute much more to the EAA value than others.
# I will keep this design, in-line with SEEA EA guidelines for area weighting,
# but it's worth noting that this could be solved in other ways.
#
combineAll <- rbind(
  spread_nf,
  spread_gr,
  spread_na
```

```
) |>
  as_tibble() |>
  drop_na() |>
  group_by(Municipality, indicator) |>
  rowwise() |>
  mutate(
    # this draws one sample pre m2 from a normal distribution:
    i_sample = list(sample(rnorm(n, mean, sd)))) |>
  select(-geometry) |>
  group_by(Municipality, indicator) |>
  reframe(i_sample =
  # this samples randomly from i_sample, ie large polygons are more likely
  # to contibute:
            sample(
              unlist(i_sample),
              size = 1000,
              replace = TRUE)) |>
  # truncation, since the tails from the normal distribution can go beyond 0 and 1
  mutate(i_sample = case_when(
    i_sample < 0 ~ 0,
    i_sample >1  ~ 1,
    .default = i_sample
  ))
```

```
(ridgepPlot <- combineAll |>
  mutate(
    indicator = case_when(
      indicator == "i_ADSV" ~ "ADSV",
      indicator == "i_alien" ~ "Alien species",
      indicator == "i_ditch" ~ "Trenching"
    )) |>
  ggplot(
    aes(
      x = i_sample,
      y = Municipality,
      fill = after_stat(x))
  ) +
  geom_density_ridges_gradient(
    bandwidth=.05,
    quantile_lines = TRUE,
    quantiles = c(0.025, .5, 0.975)
  ) +
  #scale_fill_viridis_c(option = "D") +
  scale_fill_distiller(palette = "RdYlGn",
    direction = 1) +
  theme_bw(base_size = 15) +
  guides(fill = "none") +
  geom_vline(xintercept = .6,
    size=1.2,
    lty=2) +
  theme(axis.text.x = element_text(angle=45, hjust=1, vjust=1))+
```

```r
  labs(y = "", x = "Indicator values")+
  xlim(c(0,1)) +
  facet_wrap(.~indicator)
)
```




```r
#ggsave(plot = ridgepPlot,
#       "../images/ridgePlot.jpg",
#       width=10,
#       height=5)
```

```r
(EEA_tbl <- combineAll |>
  mutate(
    indicator = case_when(
      indicator == "i_ADSV" ~ "ADSV",
      indicator == "i_alien" ~ "Alien species",
      indicator == "i_ditch" ~ "Trenching"
    )) |>
  group_by(Municipality, indicator) |>
  summarise(mean = round(mean(i_sample), 2),
            median = round(median(i_sample), 2),
            percentile_025 = round(quantile(i_sample, probs = 0.025), 2),
            percentile_975 = round(quantile(i_sample, probs = 0.975), 2))
)
```

```
# A tibble: 9 x 6
# Groups:   Municipality [3]
  Municipality indicator      mean median percentile_025 percentile_975
  <chr>        <chr>         <dbl>  <dbl>          <dbl>          <dbl>
1 Gran         ADSV           0.8    0.79           0.73           0.93
2 Gran         Alien species  0.93   0.93           0.89           0.97
```

```
3 Gran          Trenching     0.83   0.85          0.58          0.97
4 Nord-Aurdal   ADSV          0.86   0.87          0.82          0.88
5 Nord-Aurdal   Alien species 0.96   0.97          0.9           0.99
6 Nord-Aurdal   Trenching     0.87   0.86          0.83          0.94
7 Nordre Follo ADSV           0.95   0.95          0.87          1
8 Nordre Follo Alien species  0.89   0.89          0.84          0.96
9 Nordre Follo Trenching      0.26   0.26          0.05          0.48
```

```r
library(kableExtra)

# saveRDS(EEA_tbl, "../output/EEA-table.RDS")
```

```r
(EEA_tbl_out <- EEA_tbl |>
  ungroup() |>
  mutate(value =
    paste0(
      format(median, 2),
      " [",
      format(percentile_025,2),
      " - ",
      format(percentile_975,2),
      "] ")) |>
  rename(Indicator = indicator) |>
  select(
    -mean,
    -median,
    -percentile_025,
    -percentile_975,
    -Municipality
  )|>
  kbl(table.attr = "style = \"color: black;\"",
      align = "lr") |>
  kable_classic("striped",
    full_width = F) |>
  row_spec(0, bold=T) %>%
    pack_rows("Gran", 1, 3,
      label_row_css = "background-color: #cef598; color: #000000;") |>
  pack_rows("Nord-Aurdal", 4, 6,
      label_row_css = "background-color: #cef598; color: #000000;") |>
  pack_rows("Nordre Follo", 7, 9,
      label_row_css = "background-color: #cef598; color: #000000;")
)
```

1. Introduction

Ecosystem condition accounting is the game of compiling relevant data on the status, trends and qualities of ecosystems (i.e. nature) and communicating this in a structured format. Its purpose is to make it easier to account for nature in policy by making the environmental costs of certain policies and practices visible to decision makers. As natural capital keeps declining all over the world, it is becoming increasingly urgent to make the message clear to decision makers about. A statistical standard for ecosystem accounting, including ecosystem condition accounting, was developed by the UN and adopted by the UN Statistical Commission in 2021 and is called SEEA EA (United Nations (2021); System of Environmental-Economic Accounting

- Ecosystem Accounting). The standard, or framework, is a set of rules, principles and best practices for compiling Ecosystem accounts, mainly aimed at national accounts.

Central to ecosystem condition accounts are variables and indicators. These are parameters chosen to reflect the central condition characteristics of the ecosystems, and that can be quantified and ideally monitored over time to reflect the status and trends in condition. Indicators are (data) variables that are normalised (rescaled) against upper and lower reference values to become bound between the values 0 and 1. This normalisation ensures that indicators are more comparable because an indicator value of 1 will mean the same for all indicators, i.e. that the variable equals the upper reference value which again reflect the value of the variables under the reference condition. Similarly, a value of 0 mean that the variable is the worst possible state. The reference condition needs to be defined for each Ecosystem Condition Assessment separately, but SEEA EA gives some suggestion, such as an an ecosystem with no or minimal anthropogenic disturbance.

A general requirement for indicators in the SEEA EA framework is that they should give an unbiased representation of the condition inside the ecosystem assets (Czúcz et al., 2021b, table 1; see also United Nations, 2021, §2.87) Ecosystem assets are defined as "ecological entities [meaning areas] about which information is sought and about which statistics are ultimately compiled (United Nations, 2021). This requirement for indicator validity means that spatially biased data are ill suited, especially if sampling intensity varies along gradient of anthropogenic pressures and hence ecosystem condition. SEEA EA is spatially explicit, which in practice means that indicators that are in some way sampled (i.e. not complete wall-to-wall data like remotely sensed imagery), the values are projected on to areas on the map, so that all areas inside the ecosystem accounting area get assigned value for that indicator. There are at least 3 general ways to achieve this complete areal coverage of indicator values:

a. Using wall-to-wall data (e.g. remotely sensed data)
b. Predict values using a model (e.g. by accounting for environmental variation)
c. Simple projection of some best estimate, typically a central tendency from area representative data

The need for an unbiased estimation of indicator values is unquestionable, but nonetheless, this requirement puts a large limitation on what types of data one can use. Ecosystem condition assessments are generally limited by data availability, and the choice of variables and indicators to include in assessments are more often or not a pragmatic and opportunistic one which is unlikely to reflect the full scope of the ecosystem condition characteristics. Note that the same is true for thematic biases. For example reflected in the scarcity of data included on insects or soil biota, even though most will agree they represent key ecosystem characteristics. Also having data from only one or a subset of nature types inside what is defined as the ecosystem in the assessment, is a typical thematic bias in ecosystem accounting. However, in this paper we chose to focus on spatial bias.

Being able to make use of spatially biased data would greatly alleviate data shortage problems in ecosystem condition accounts. One way to achieve this is modelling (option b in the list above). Models can describe the general associations between the spatially sampled data and the context (e.g. the set of environmental variables) where it was samples, and use these relationships to predict and project indicator values to areas that where not originally sampled. Depending on the data that goes into these models, they can be very reliable and make good indicators. This is especially true when the ecosystem assets are large (e.g. regions or nations). But when they are small, like the scale of a municipality, and when the indicator is more likely to be used as the evidence base in concrete physical land use planning, then the inherent level of uncertainty from such models becomes unacceptable.

In this study we explore the potential for using a stratified aggregation technique to make use of spatially biased field data in ecosystem condition accounting. We demonstrate this technique using a generic GIS-based workflow for compiling ecosystem condition accounts that can be applied at any spatial scale, and we highlight the opportunities for local use-cases of this workflow by contrasting our findings across three neighboring municipalities in Norway (Figure 1). The main question is how much generalisation can we perform on the data we have before the resulting indicator looses its practical value in local governance processes. We end by interviewing end-users from the relevant municipalities about the perceived benefits and shortcomings of our condition indicators.

```
positionMap
```

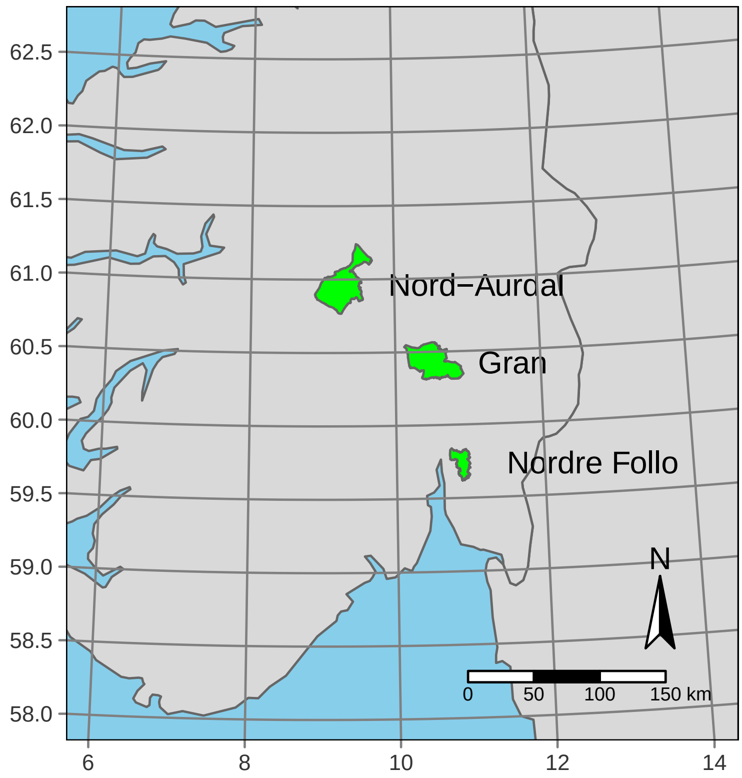


Figure 1: Postion of the three example municipalities in Norway.

2. Material and Methods

This study makes use of a data set from from a standardised field survey of nature types in Norway that started in 2018 and which is still ongoing (Norwegian Environmental Agency, 2024). We included data from 2018 to 2023. In this survey, selected nature types are delineated on a map, and each locality is scored on a range of variables relevant for describing the state and quality of nature. The surveys are commissioned with the goal of producing data relevant for immediate land-use decisions, and is therefore spatially biased, typically towards areas with high human impact or expected impact. In addition there is a thematic and size bias in the sampling protocol. For examples, for the forest ecosystem, rare, endangered or calcareous forest types are delineated, whereas more common or ordinary forest types are not. For that reason we we focused on open mire ecosystems in Norway where the thematic bias is less severe. Of all possible mire types, the survey only maps the following:

- Southern ombrotrophic mires > 2500 m^2
- Northern ombrotrophic mires > 10.000 m^2
- All semi-natural mires (minerotrophic fens)
- Calcareous southern fens >500 m^2
- Calcareous northern fens >1000 m^2

In the above, *southern* refers to boreonemoral and southboreal zones, and *northern* refers to mid-boreal, north boreal, and alpine zones **?** . In addition, the northern fens need to be even more calcareous than the southern fens in order to be surveyed.

In this paper we assume the survey is representative for the entire mire ecosystem in Norway. Although it is possible that smaller or less calcareous mires will score systematically different than the ones that are surveyed we do not think this is so much the case for our variables. It may also be that alien plants are more or less common on bogs relative to fens. For this we have no assumption.

For the mire ecosystem we have an ecosystem delineation map produced using remotely sensed data and a deep learning model (Bakkestuen et al., 2023). This model, which has a 90.9% precision when tested against independent field data, estimates 12.7% of the area in southern Norway is mire (Bakkestuen et al., 2023). Mires are ecologically and socially important in Norway simply doe to its large extent, and due to its role in climate mitigation, as mires store a large amount of carbon (REF?). There has not been a national assessment of the ecosystem condition of mires in Norway, but we recent contributed on a report which presented several new indicators that can be used in future assessment (Nybø et al., 2023; see also Kolstad et al., 2023)). This study builds on the work in that report.

We started by defining the reference state as one where ecosystems are subject to little or no human influence, with a climate as in the period 1961-1990 and a native species pool similar as today (i.e no mammoths allowed). We then identified two *ecosystem condition characteristics* that, along with many other characteristics that we do not specify here, define the typical behavior of open mire ecosystems in the reference condition: species composition, and; vegetation intactness. To describe these two characteristics we use six condition variables from the same data set (Table 1). See Miljødirektoratet (2022) for the sampling protocol (in Norwegian).

```
knitr::include_graphics("../images/studyLocations.jpg")
```

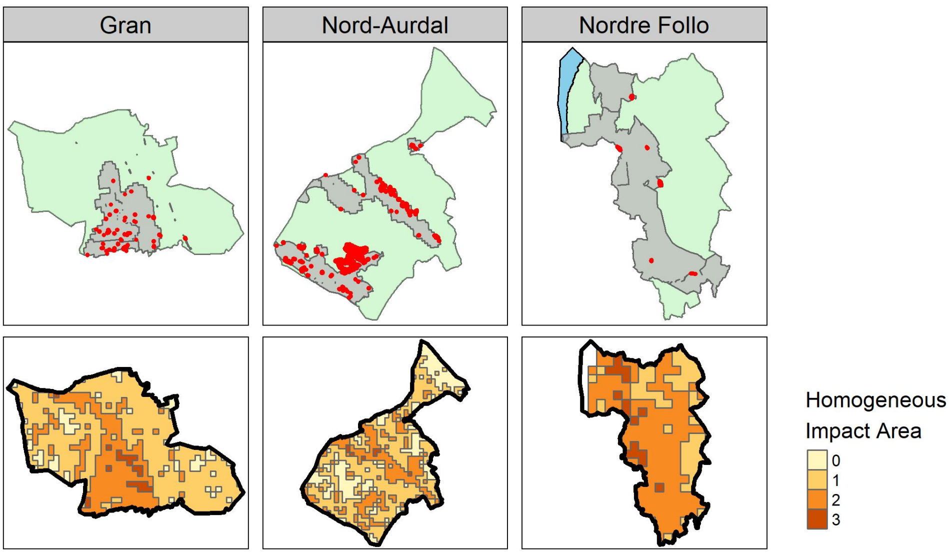


Figure 2: Map of the three focal municiaplities. For each municipality the map shows ocean in blue and non-ocean in green. The survey coverage maps are in grey, and the mapped mires are in red, with polygon borderes made extra thich to make them visible, but then also exagerating their size.

Table 1: Variables used in this study

id	Variable code	Variable name	Measurement unit	Description	Reference
1	7FK	Prevalence of alien species	Unitless, ordinal, non-linear scale from 1 (no alien species) to 7 (only alien species)	The fraction of the species composition made up from alien species	Halvorsen and Bratli (2019)
2	7SE	Human caused abration or abration-caused erosion	Unitless, ordinal, non-linear scale from 1 to 4.	Measures the frequency of imagined 4 m^2 quadrats layed over the area that has some sign of abration	Halvorsen and Bratli (2019)
3	PRSL	*as above*	Unitless, ordinal, non-linear scale from 0 to 7.	Same as 7SE, but recorded at a higher resolution	Miljødirektoratet (2022)
4	7TK	Tracks from large vehicles	Unitless, ordinal, non-linear scale from 1 to 4.	Measures the frequency of imagined 100 m^2 quadrats layed over the area that has some signs of vehicle tracks	Rune Halvorsen and Harald Bratli (2019)
5	PRTK	*as above*	Unitless, ordinal, non-linear scale from 0 to 7.	Same as 7TK, but recorded at a higher resolution	Miljødirektoratet (2022)
6	7GR-GI	Ditching intensity	Unitless, ordinal scale from 1 to 5	Describes the effect that drainage ditches is estimate to have on the species composition and environmental variables ones the system reached its new equilibrium	Miljødirektoratet (2022)
7	Infrastructure Index	Infrastructure Index	Unitelss linear scale from 0 to 13.2	Unitelss index ranging from from 0 to 13.2	Erikstad et al. (2023)
8	HIA	Human Impact area	Ordinal, non-linear scale from 1 to 4	A categorical representation of the infrastructure index	this paper

Variables 1 -5 were 7FK were originally recorded along frequency ranges. We the lower limit for each frequency range to convert them into percentages. This was done because the data was strongly right skewed . 7FK was developed into a single indicator (i.e. a normalised variable) named *Alien plant cover*. Variables 2-5 describe very related aspects and so they where combined into one indicator called antropogenic disturbance to soil and vegetation, or ADSV for short. This was done by summing the two variables after they had been converted to percentages. This was not a perfect solution, especially since some localities only had one of the variables recorded, but we chose this, rather than for example using a *worst rule* principle, in order to better separate the localities in terms of their indicator values. For both indicators (alien species and ADSV), the lower and upper reference values, i.e. the worst and best possible condition that the variables can be in, is defined as 100% and 0%, respectively. The threshold for *good ecosystem condition* was defined as 10%, and this was mapped to the value 0.6 on the rescaled indicators. Variable 7GR-GI was rescaled into an indicator named *ditching intensity* by using the lower and upper reference values 1 and 5, respectively, and a threshold value of 2.5. A variable value of 1 indicates an intact mire, and a value of 5 indicates a mire transitioning away from a wetland. A value of 2 indicator observable change within the range expected for the same mapping unit, and a value of 3 indicates a mire transitioning into a neighboring (ecologically speaking) mapping unit. See Figure 3 for a schematic workflow for the indicator development.

The Alien plant cover was assigned to ECT class B1 - Compositional state characteristics, and ADSV was assigned to ECT class A1 - Physical state characteristics (Czúcz et al., 2021a).

```
knitr::include_graphics("../images/workflow.jpg")
```

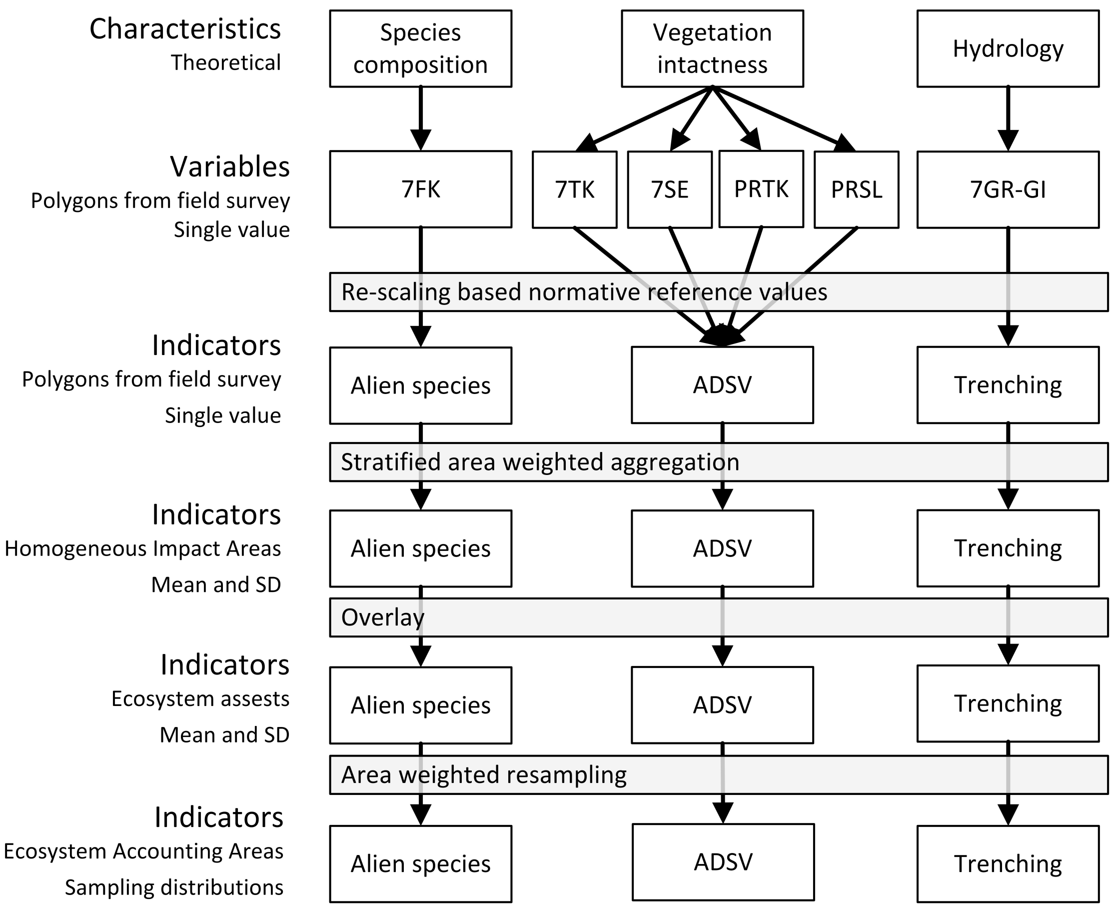


Figure 3: Schematic workflow followed in this paper. Sub-index A is calculated using spatially explicit aggregation.

We chose three municipalities in Norway to test out the indicators (Figure 1). The differ in several aspect, such as the amount of mire area, the total area surveyed, and the prevalence of infrastructure (Table 1).

Table 2: Information for the three target municipalities in Norway

Municipality	Total terrestrial area (km^2)	% of terrestrial area surveyed	% open mires in relation to total terrestrial area	Total mire area (km^2)	%of mire area inside survey area	Number of mire polygons in survey	Mean Infrastructure Index value
Nordre Follo	203	40	0.3	0.6202	18.61	6	1.82
`Gran`	756	21	2.7	20.2422	0.53	56	1.25
`Nord-Aurdal`	906	26	11.1	100.7177	18.14	236	1.02

Validation/sensitivity analyses (varying HIA resolution and minimum n?)

Aggregation

3. Results

Data availability/coverage

Validation

```
knitr::include_graphics("../images/validation-plot.jpg")
```

Figure 4: Proportion of localities with different scaled indicator values for three ecosystem condition indicators along a gradient of infrastructure densities.

```
knitr::include_graphics("../images/forest-plot.jpg")
```

Figure 5: Indicator values (circles = mean; bars = 95% credible intervals) for three mire ecosystem condition indicators in three municipalities. The indicator values are calculated uniquely for each Homogeneous Impact Area (HIA) in each municipality. The numbers to the left of each bar is the sample size, i.e. the number of surveyed mires.

```
knitr::include_graphics("../images/spread-na-small.jpg")
```

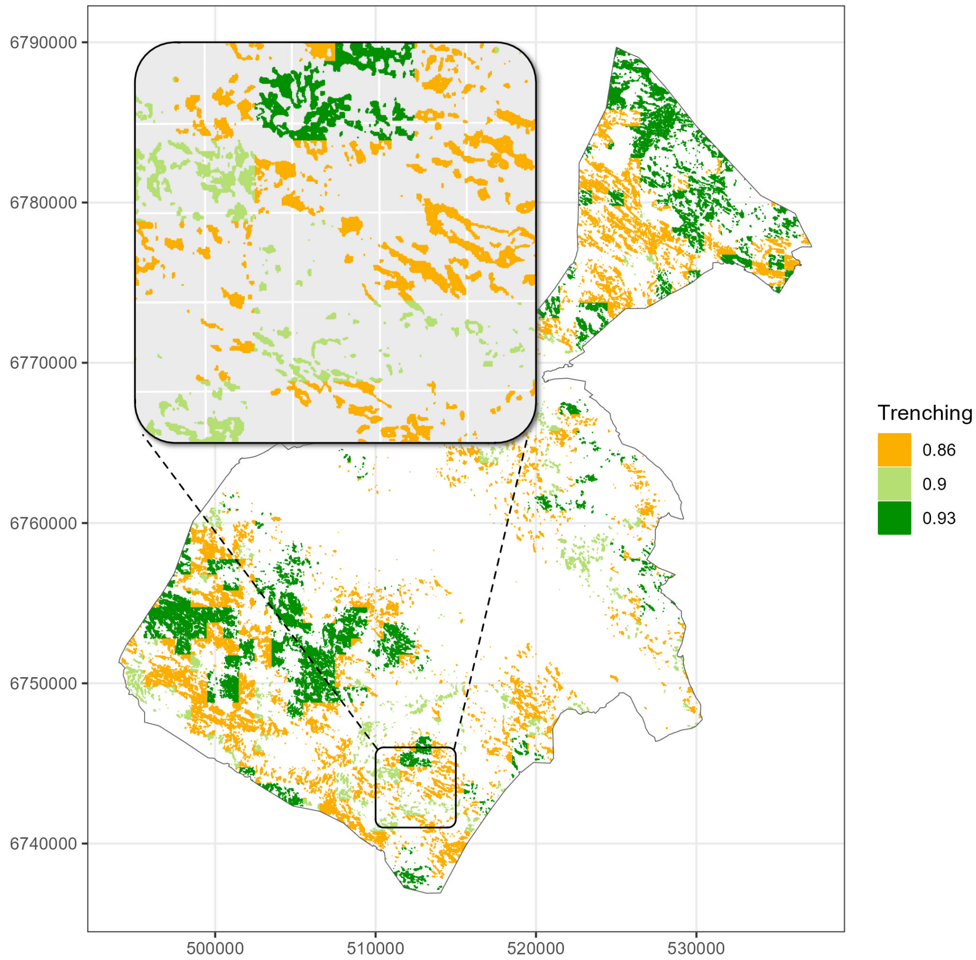


Figure 6: An indicator for mire trenching shown for Nord-Aurdal municipality. Individual mire polygons are colored by the mean indicator values for the homogenous impact area where it lies. Colors are chosen to best reflect categorical differences and exagerates the absolute difference between areas. The inset it just a visual aid.

```
knitr::include_graphics("../images/ridgePlot.jpg")
```

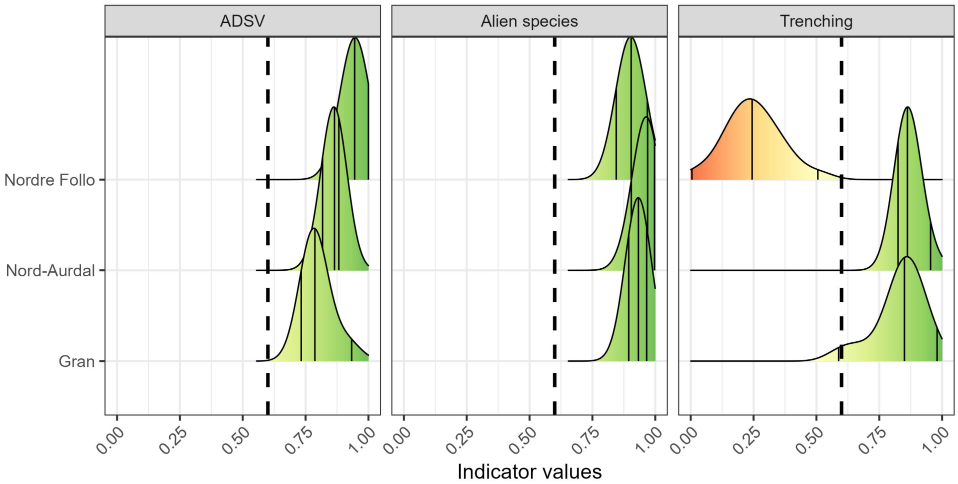


Figure 7: Distributions for three ecosystem condition indicators in the Norwegian Municiaplities. The color gradient reflects the same as the x-axis. The dotted vertical line represnet the threshold for what is considered reduced ecosystem condtion (< 0.6).

4. Discussion

5. Conclusion

Credit authorship contribution statement

Declaration of Competing Interest

Acknowledgements

Data availability

References {.unnumbered}

References

Bakkestuen, V., Venter, Z., Ganerød, A.J., Framstad, E., 2023. Delineation of Wetland Areas in South Norway from Sentinel-2 Imagery and LiDAR Using TensorFlow, U-Net, and Google Earth Engine. Remote Sensing 15, 1203. URL: https://www.mdpi.com/2072-4292/15/5/1203, doi:10.3390/rs15051203. number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

Czúcz, B., Keith, H., Driver, A., Jackson, B., Nicholson, E., Maes, J., 2021a. A common typology for ecosystem characteristics and ecosystem condition variables. One Ecosystem 6, 1–16. doi:10.3897/oneeco.6.e58218. publisher: Pensoft Publishers.

Czúcz, B., Keith, H., Maes, J., Driver, A., Jackson, B., Nicholson, E., Kiss, M., Obst, C., 2021b. Selection criteria for ecosystem condition indicators. Ecological Indicators 133, 108376. URL: https://linkinghub.elsevier.com/retrieve/pii/S1470160X21010414, doi:10.1016/j.ecolind.2021.108376. publisher: Elsevier B.V.

Erikstad, L., Simensen, T., Bakkestuen, V., Halvorsen, R., 2023. Index Measuring Land Use Intensity—A Gradient-Based Approach. Geomatics 3, 188–204. URL: https://www.mdpi.com/2673-7418/3/1/10, doi:10.3390/geomatics3010010. number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

Halvorsen, R., Bratli, H., 2019. Dokumentasjon av NiN versjon 2.2 tilrettelagt for praktisk naturkartlegging: utvalgte variabler fra beskrivelsessystemet. – Natur i Norge, Artikkel 11 (versjon 2.2.0). Technical Report. Trondheim.

Kolstad, A.L., Grainger, M., Sandvik, H., Venter, Z., Speed, J.D., Nater, C., Evju, M., Åstrøm, J., 2023. Indicators for ecosystem condition in norway. URL: https://ninanor.github.io/ecosystemCondition/index.html.

Miljødirektoratet, 2022. Naturtyper - miljødirektoratets instruks.

Norwegian Environmental Agency, 2024. Naturtyper etter Miljødirektoratets Instruks. URL: https://kartkatalog.miljodirektoratet.no/Dataset/Details/2031.

Nybø, S., Kolstad, A.L., Sandvik, H., Bakkestuen, V., Evju, M., Framstad, E., Grainger, M., Jepsen, J.U., Nater, C.R., Speed, J.D., Töpper, J.P., Venter, Z., Åström, J., Åström, S.C., Bargmann, T.B., Silvennoinen, H.M., Sydenham, M.A., 2023. Indikatorer for økologisk tilstand i våtmark, semi-naturlig mark og naturlig åpne områder under skoggrensa. Technical Report. Norwegian Institute for Nature Research. URL: https://brage.nina.no/nina-xmlui/handle/11250/3096724.

Rune Halvorsen, Harald Bratli, 2019. Dokumentasjon av NiN versjon 2.2 tilrettelagt for praktisk naturkartlegging: utvalgte variabler fra beskrivelsessystemet. – Natur i Norge, Artikkel 11 (versjon 2.2.0). Technical Report. Artsdatabanken. Trondheim.

United Nations, 2021. System of Environmental-Economic Accounting-Ecosystem Accounting White cover (pre-edited) version. Technical Report. URL: https://seea.un.org/ecosystem-accounting.