Human Influence in SDMs: Literature Review (Part III)

Veronica F. Frans (email: verofrans@gmail.com)

February 1, 2024

Contents

1 Summary

This is the third R script of the literature review and synthesis for the article entitled, "Gaps and opportunities in modeling human influence on species distributions in the Anthropocene," by Veronica F. Frans and Jianguo Liu.

Here, in Part III of the qualitative synthesis, the following is accomplished:

(1) Cleanup, simplification, and synthesis of human predictor names across articles
(2) Summary of predictor use across articles, categories, and data types
(3) Summary of ambiguous predictors
(4) Summary of buffered predictors
(5) Summary of study vs. human predictor time frames
(6) Plot of first and last years of predictor use
(7) CSV file export of predictor name list

The next script (Part IV) uses the CSV file of the systematic review to get a global context for human predictor use in SDMs through maps.

2 R Setup

We are using R version 4.3.0 (R Core Team 2023).

2.1 Libraries

Load libraries

```r
# load libraries
  library("dplyr")         # for table manipulations
  library("scales")        # for scales and formatting
  library("kableExtra")    # for table viewing in Rmarkdown
  library("tidyr")         # for table manipulations
  library("plyr")          # for table manipulations
  library("tidyverse")     # for graphics/table management
  library("ggplot2")       # for graphics
  library("RColorBrewer")  # for graphics
  library("alluvial")      # for graphics
  library("ggforce")       # for graphics (speeds up ggplot)
  library("ggalluvial")    # for graphics
  library("ggbreak")       # for graphics
  library("patchwork")     # for graphics
  library("migest")        # for graphics (chord diagram)
  library("circlize")      # for graphics (chord diagram)
  library("chorddiag")     # for graphics (chord diagram)
  library("raster")        # for mapping
  library("rgdal")         # for mapping
  library("sp")            # for mapping
  library("ggmap")         # for mapping and graphics
  library("maps")          # for mapping
  library("plotfunctions") # for data visualization
  library("svglite")       # for saving graphics in svg format
```

2.2 Directories

The primary directory is the folder where the `hum_sdm_litrv_r.Rproj` is stored.

```
# create image folder and its directory
  dir.create(paste0("images"))
  image.dir <- paste0("images\\")

# create data folder and its directory
  dir.create(paste0("data"))
  data.dir <- paste0("data\\")
```

2.3 Load data

Upload the data table from the abstract screening and review, and subset to only the articles that are accepted. We will also need a few saved CSV files from Part II.

```
# full article screening and review table
  # read CSV
    rev.df <- read.csv(paste0(data.dir,"hum_sdm_lit_review_RAW.csv"),
                       header=T, sep=",")

  # subset only accepted papers (after year 2000)
    yes.df <- rev.df[(rev.df$relevant=="yes"),]
    yes.df <- yes.df[(yes.df$year>=2000),]

# study domain, taxa, and focus
  domtaxfoc.df <- read.csv(paste0(data.dir,"domain_taxa_focus_count_papers.csv"),
                           header=T, sep=",")
```

3 Compiling a synthesized list of human predictors used in SDMs

In this section, we will go over the entire list of human-related predictors that were recorded during the full article review. These are predictors for past, present, and future time frame studies. The predictor names are based on descriptions of verbatim names given by the authors. This full list of unique predictor names will be synthesized to have a more holistic overview of human predictor use in SDMs. Below, edits to predictor names are made.

3.1 Predictor name table setup

First, the table of predictors are stratified across time frame, and then turned into a longer table, as the raw data table has all predictors per paper and per time frame semi-colon separated within one row/column coordinate. The long table will have a unique row per paper, time frame, domain, and predictor name.

```
# Extract table and predictors from relevant papers
  preds.list.df <- subset(yes.df,
                          select = c("uid", "domain","past_hum_preds",
                                     "present_hum_preds","future_hum_preds"))

# Separate by past/present/future
```

```
  preds.past.df <- subset(preds.list.df, select = c("uid","past_hum_preds"))
  preds.pres.df <- subset(preds.list.df, select = c("uid", "present_hum_preds"))
  preds.fut.df <- subset(preds.list.df, select = c("uid","future_hum_preds"))

# Split multiple predictors contained in one row into multiple other new rows
  preds.past.df <- separate_rows(preds.past.df, past_hum_preds,sep="; ",convert = TRUE)
  preds.pres.df <- separate_rows(preds.pres.df, present_hum_preds,sep="; ", convert = TRUE)
  preds.fut.df <- separate_rows(preds.fut.df, future_hum_preds,sep="; ", convert = TRUE)

# Remove all NA's and "NONE"
  # Identify the patterns to remove
    patterns_to_remove <- c("", NA,"NONE","none")

  # Remove rows with the specified patterns
    preds.past.df <- preds.past.df[!(preds.past.df$past_hum_preds %in% patterns_to_remove),]
    preds.past.df <- preds.past.df[rowSums(is.na(preds.past.df)) == 0,] # redo row numbers
    preds.pres.df <- preds.pres.df[!(preds.pres.df$present_hum_preds %in% patterns_to_remove),]
    preds.pres.df <- preds.pres.df[rowSums(is.na(preds.pres.df)) == 0,] # redo row numbers
    preds.fut.df <- preds.fut.df[!(preds.fut.df$future_hum_preds %in% patterns_to_remove),]
    preds.fut.df <- preds.fut.df[rowSums(is.na(preds.fut.df)) == 0,]     # redo row numbers

# Convert all to factors
  preds.past.df$past_hum_preds <- as.factor(as.character(preds.past.df$past_hum_preds))
  preds.pres.df$present_hum_preds <- as.factor(as.character(preds.pres.df$present_hum_preds))
  preds.fut.df$future_hum_preds <- as.factor(as.character(preds.fut.df$future_hum_preds))

# Change from wide to long-format dataframe and change column names
  preds.past.df$timeframe <- "past"
  preds.pres.df$timeframe <- "present"
  preds.fut.df$timeframe <- "future"
  colnames(preds.past.df)[which(names(preds.past.df)=="past_hum_preds")] <-"predictor"
  colnames(preds.pres.df)[which(names(preds.pres.df)=="present_hum_preds")] <-"predictor"
  colnames(preds.fut.df)[which(names(preds.fut.df)=="future_hum_preds")] <-"predictor"

# Bind again into one list (overwriting original list of predictors from above)
  preds.list.df <- rbind(preds.past.df, preds.pres.df, preds.fut.df)

# Change time frame to factor
  preds.list.df$timeframe <- as.factor(as.character(preds.list.df$timeframe))

# save as CSV
  write.csv(preds.list.df,paste0(data.dir,"predictor_list_RAW.csv"),
            row.names = FALSE)

# preview
  head(preds.list.df); tail(preds.list.df)

# summary
  #summary(preds.list.df$predictor)
  #summary(preds.list.df)
```

```
## # A tibble: 6 x 3
##     uid predictor                                 timeframe
```

```
##   <int> <fct>                                      <fct>
## 1   180 developed_open_space_percent_165m          past
## 2   180 developed_light_intensity_percent_165m     past
## 3   180 developed_moderate_intensity_percent_165m past
## 4   180 agricultural_land_percent_165m             past
## 5   180 developed_open_space_percent_315m          past
## 6   180 developed_light_intensity_percent_315m     past
## # A tibble: 6 x 3
##     uid predictor                             timeframe
##   <int> <fct>                                 <fct>
## 1 11339 non-irrigated_vineyards               future
## 2 11339 built-up_areas                        future
## 3 11339 urban_areas_distance                  future
## 4 11339 land_cover_class_sum                  future
## 5 11686 land_cover                            future
## 6 11686 urban_and_forest_percent_750m_buffer future
```

Count how many unique predictor names there are in the list.

```
# inspect
  paste("Unique predictor names to be synthesized:",
        length(unique(preds.list.df$predictor)))
```

```
## [1] "Unique predictor names to be synthesized: 2746"
```

3.2 Predictor name synthesis

3.2.1 Synthesizing food/agriculture predictor names

Edit predictor names related to agriculture (farming, cultivating, rearing, animals, soil types).

```
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    # first, fix small spaces
      "^ " = "",
      "_ " = "_",
      " " = "_",
    # agricultural terms
      "algricultural" = "agricultural",
      "agricultura_" = "agricultural_",
      "agricultural_land" = "agricultural_areas",
      "^agricultural_area$" = "agricultural_areas",
      "agricultural_area_" = "agricultural_areas_",
      "agricultral" = "agricultural",
      "^cultivated_fields$" = "cultivated_areas",
      "^cropland_area_2050$" = "cropland_areas",
      "^cropland_area_2070$" = "cropland_areas",
      "^cropland$" = "cropland_areas",
      "^croplands$" = "cropland_areas",
      "^pastures$" = "pasture_areas",
      "^pasture$" = "pasture_areas",
      "^percent_agricultural_land$" = "agricultural_areas_percent",
```

```
    "agriculture_" = "agricultural_areas_",
    "^ agricultural_areas_patches_mean_size$" = "agricultural_areas_patches_mean_size",
    "^cropland_area$" = "cropland_areas",
    "cropland_area-" = "cropland_areas_area-",
    "cropland_proportion" = "cropland_areas_proportion",
    "farmland_" = "farmlands_",
    "farmlands_areas_" = "farmlands_",
    "_percent_cover" = "_percent",
    "croplands" = "cropland_areas",
    "^pasture_areas$" = "pastures",
    "pasture_area_" = "pastures_",
    "^pastureland$" = "pastures",
    "pasturelands_" = "pastures_",
    "pasture_" = "pastures_",
    "pastures_areas_h" = "pastures_",
    "pasturesm" = "pastures",
    "scrub_pasturessize" = "pastures_scrub_area_size",
    "shrub_and_pastures_" = "pastures_and_shrub_",
    "percent_agricultural_areas" = "agricultural_areas_percent",
    "agricultural_areas_heterogenous" = "agricultural_areas_heterogeneous",
    "cows_density" = "cattle_density",
    "FAO_cattle_density" = "cattle_density",
    "dryland_crops_percent" = "crops_dryland_percent",
    "dry_cropland_percent" = "cropland_dry_percent",
    "dry_grass_cropland_percent" = "cropland_dry_grass_percent",
    "dry_farming_frequency" = "farming_dry_frequency",
    "dry_farm_distance" = "farms_dry_distance",
    "dry_herbaceous_crops" = "crops_dry_herbaceous_present_absent",
    "irrigated_farming_" = "irrigated_farms_",
    "irrigated_farm_" = "irrigated_farms_",
    "harvest_instensity_wild_yams" = "harvest_wild_yams_intensity",
    "grazing_area_" = "grazing_areas_",
    "^vineyard$" = "vineyards",
    "vineyard_" = "vineyards_",
    "small_ruminant_" = "small_ruminants_",
    "agricultural_areas_ha" = "agricultural_areas_area_ha",
    "arable_lands_" = "arable_land_",
    "irrigation_area_" = "irrigated_areas_",
    "residual_pastoral_areas" = "pastoral_areas_residual",
    "mixed_cropland_areas_" = "cropland_areas_mixed_",
    "cropland_percent" = "cropland_areas_percent",
    "fruit_tree_crops" = "crops_fruit_tree",
    "rainfed_crops" = "crops_rainfed",
    "^pastureschange_" = "pastures_change_",
    "plantation_proportio" = "plantation_proportion",
    "horses_" = "horse_",
    "pig_density" = "pig_livestock_density",
    "cropland_density" = "cropland_areas_density",
    "cropland_area_" = "cropland_areas_",
    "cropland_areas_change_" = "cropland_areas_area_change_",
    "non-irrigated arable land" = "non-irrigated_arable_land",
    "acacia_plantations" = "plantations_acacia",
    "large_livestock" = "livestock_large",
```

```
    "irrigated_area_" = "irrigated_areas_",
    "agriculture_area_" = "agricultural_areas_",
    "winter_grain_" = "crop_winter_grain_",
    "winter_cereals" = "crop_winter_cereals",
    "summer_cereals" = "crop_summer_cereals",
    "winter_wheat" = "crop_winter_wheat",
    "^spring_grain" = "crop_spring_grain",
    "wheat_frequency" = "crop_wheat_frequency",
    "^wheat_crops_" = "crop_wheat_",
    "^grain_crops_" = "crop_grain_",
    "^grape_crops_" = "crop_grape_",
    "^grapevine_crops_" = "crop_grapevine_",
    "^walnut_crops_" = "crop_walnut_",
    "sugar_beet_crops_" = "crop_sugar_beet_",
    "sugar_cane_cover_" = "crop_sugar_cane_",
    "vineyards_ha" = "vineyards_area_size",
    "vineyards_m2" = "vineyards_area_size",
    "vineyards_size" = "vineyards_area_size",
    "cows" = "cattle",
    "^cattle_" = "livestock_cattle_",
    "^livestock_area_" = "livestock_areas_",
    "^fodder_" = "livestock_fodder_",
    "agricultural_areas_irrigated" = "irrigated_agricultural_areas",
    "agricultural_areas_patches_mean" = "agricultural_areas_mean",
    "wooded" = "woody",
    "areass" = "areas",
    "aquaculture_facility" = "aquaculture_",
    "agricultural_establishments" = "agricultural_areas",
    "agricultural_grassland_" = "agricultural_grasslands_",
    "^agriculture$" = "agricultural_areas",
    "agriculturr" = "agricultur",
    "^maize_" = "crop_maize_",
    "^alfalfa_crop_" = "crop_alfalfa_",
    "^alfalfa_" = "crop_alfalfa_",
    "^almond_crop_" = "crop_almond_",
    "^almond_" = "crop_almond_",
    "^annual_crops_" = "crops_annual_",
    "^annual_days_grazed$" = "grazing_annual_days",
    "^meadows_orchards" = "meadows_and_orchards",
    "^low-intensity_agricultural_areas" = "agricultural_aeas_low-intensity",
    "^arable_lands$" = "arable_land",
    "^artichokes_frequency$" = "crop_artichoke_frequency",
    "^harvested_artichokes_frequency$" = "crop_artichokes_harvested_frequency",
    "^mixed_agricultural_areas$" = "agricultural_areas_mixed",
    "mowing_meadow" = "meadow_mowed",
    "broadleaved_deciduous_orchards_percent" = "orchards_broadleaved_deciduous_percent",
    "broadleaved_evergreen_orchards_percent" = "orchards_broadleaved_evergreen_percent",
    "needle-leaved_evergreen_orchards_percent" = "orchards_needle-leaved_evergreen_percent",
    "oilseed_rape" = "crop_oilseed_rape",
    "oil_seed_rape_" = "crop_oilseed_rape_",
    "fields_rapeseed" = "crop_oilseed_rape",
    "rape_crop_" = "crop_oilseed_rape_",
    "_number$" = "_count",
```

```
    "number_annual_fish_stock_events" = "annual_fish_stock_events_count",
    "old_deciduous_conifer_plantations_percent" = "plantations_old_deciduous_conifer_percent",
    "old_evergreen_conifer_plantations_percent" = "plantations_old_evergreen_conifer_percent",
    "_grooves" = "_groves",
    "olive_groves" = "orchards_olives",
    "orchard_" = "orchards_",
    "_olive_tree_groves" = "_olive_tree_orchards",
    "olive_cultivations" = "orchards_olives",
    "olive_plantations" = "orchards_olives",
    "olive_and_fruit_groves" = "orchards_olives_and_fruit_groves",
    "fruit_trees_and_olive_groves" = "orchards_olives_and_fruit_groves",
    "fruit_tree_plantation" = "plantations_fruit_tree",
    "fruit_trees_and_berry_plantation" = "plantations_fruit_tree_and_berry",
    "fruit_and_berry_plantations" = "plantations_fruit_and_berry",
    "fruit_plantations" = "plantations_fruit",
    "fruit_trees_percent" = "plantations_fruit_trees_percent",
    "coconut_plantations" = "plantations_coconut",
    "coffee_plantations" = "plantations_coffee",
    "^tree_plantation$" = "tree_plantations",
    "tree_cultures_" = "tree_plantations_",
    "complex_cultivation_areas" = "complex_cultivation_patterns",
    "cultivation_complex_" = "complex_cultivation_patterns_",
    "complex_cultivations_" = "complex_cultivation_patterns_",
    "pastures_land" = "pastures_",
    "livestock_cattle_presence_absence" = "livestock_cattle_presence",
    "^winter_cereals" = "crop_cereal_winter",
    "^summer_cereals" = "crop_cereal_summer",
    "^fruit_crops_" = "crop_fruit_",
    "cereal_crops" = "crop_cereal",
    "cereal_land_cover" = "crop_cereal",
    "fields_cereals_" = "crop_cereal_",
    "^cereal_" = "crop_cereal_",
    "cereals_" = "crop_cereal_",
    "dry_cereal_cultures" = "crop_cereal_dry",
    "paddy_agriculture" = "paddy_fields",
    "paddy_areas" = "paddy_fields",
    "paddy_field_" = "paddy_fields_",
    "paddy_fields_size" = "paddy_fields_area_size",
    "^rice$" = "rice_paddy",
    "^rice_crop$" = "rice_paddy",
    "^rice_field_" = "rice_paddy_",
    "^rice_fields_" = "rice_paddy_",
    "rice_paddy_cover_km2" = "rice_paddy_area_size",
    "^rice_percent" = "rice_paddy_percent",
    "palm_oil_plantations" = "plantations_palm_oil",
    "^soybean" = "crop_soybean",
    "strawberry_crops" = "crop_strawberry",
    "specialized_crop" = "crop_specialized",
    "pastureland" = "pastures",
    "catchment_percent" = "_percent_catchment",
    "cropland_areas" = "cropland",
    "50percent" = "50_percent",
    "cropland_sum" = "cropland_count",
```

```
    "_sum_length" = "_length_sum",
    "^permanent_culture$" = "permanent_cultures",
    "cultivated_area_size" = "cultivated_areas_area_size",
    "cultivated_areas_area_size" = "cultivated_areas_size",
    "cultivated_land_" = "cultivated_areas_",
    "cultivated_lands_" = "cultivated_areas_",
    "pig_livestock_" = "livestock_pig_",
    "porcine" = "livestock_pig",
    "cultivated_proportion" = "cultivated_areas_percent",
    "potato_crops" = "crop_potato",
    "ranchos" = "rangeland",
    "rain-fed_crops" = "crops_rainfed",
    "rainfed_agriculture" = "cropland_areas_rainfed",
    "rainfed_cropland_distance" = "cropland_areas_rainfed_distance",
    "^rangelands" = "rangeland",
    "potato_crops" = "crop_potato",
    "dry_cropping" = "crop_dry",
    "dry_crops_" = "crop_dry_",
    "dryland_crops" = "crop_dry",
    "dry_herbaceous_crops" = "crop_dry_herbaceous",
    "dry_heterogeneous_crops" = "crop_dry_heterogeneous",
    "dry_field_crops" = "crop_dry",
    "dry_farm_" = "farmlands_dry_",
    "farms_dry" = "farmlands_dry",
    "dry_tree_crops" = "crop_dry_tree",
    "fallow_fields" = "fallow_land",
    "eucalyptus_forest" = "plantations_eucalyptus",
    "pine_and_eucalyptus_plantations" = "plantations_pine_and_eucalyptus",
    "farmlands_area_percent" = "farmands_percent",
    "^farmlands_areas$" = "farmlands",
    "farms_distance" = "farmlands_distance",
    "tonns" = "tons",
    "fields_maize" = "crop_maize",
    "corn_presence" = "crop_maize_presence",
    "corn_field_percent" = "crop_maize_percent",
    "finfish_aquaculture" = "aquaculture_finfish",
    "^salmon_farm" = "aquaculture_salmon",
    "safflower_crops" = "crop_safflower",
    "shaded_coffee_crops" = "crop_shaded_coffee",
    "sheep_and_goat" = "sheep_goat",
    "sheep_or_goat" = "sheep_goat",
    "grazed_land" = "grazing_areas",
    "grazing_land" = "grazing_areas",
    "grazing_presence" = "grazing_areas",
    "grazing_presence_absence" = "grazing_areas",
    "grazing_nongrazing_land" = "grazing_areas",
    "uncoverted_to_maize_land_type" = "unconverted_maize",
    "hedge_p" = "hedgerows_p",
    "hedge_rows" = "hedgerows",
    "hedgerow_" = "hedgerows_",
    "irrigated_agriculture" = "irrigated_agricultural_areas",
    "heterogeneous_agriculture" = "agricultural_areas_heterogeneous",
    "heterogeneous_agricultural_areas" = "agricultural_areas_heterogeneous",
```

```
    "cultivated_crop_percent" = "cultivated_crops_percent",
    "^sheep_abundance_class" = "livestock_sheep_abundance_class",
    "^sheep_density" = "livestock_sheep_density",
    "^sheep_goat_density" = "livestock_sheep_goat_density",
    "^sheep_grazing_alpine_percent" = "grazing_sheep_alpine_percent",
    "^sheep_farm_distance" = "livestock_sheep_farm_distance",
    "livestock_density_cattle" = "livestock_cattle_density",
    "livestock_density_goats" = "livestock_goat_density",
    "livestock_density_sheep" = "livestock_sheep_density",
    "agricultural_areas_20km_radius_percent" = "agricultural_areas_percent_20km_radius",
    "fruit_trees_and_orchards_olives" = "orchards_fruit_and_olives",
    "orchards_olives_and_fruit_groves_percent" = "orchards_fruit_and_olives",
    "^groves" = "orchards",
    "plantations_fruit_trees" = "orchards_fruit_tree",
    "plantations_fruit_tree" = "orchards_fruit_tree",
    "berrys" = "berries",
    "cow_density" = "livestock_cattle_density",
    "deer_density" = "livestock_deer_density",
    "goat_density" = "livestock_goat_density",
    "hedges" = "hedgerows",
    "opland_vegetation_mosaic" = "cropland_vegetation_mosaic",
    "cropland_areas" = "croplands",
    "crop_drypercent" = "crop_dry_percent",
    "barley_crops" = "crop_barley",
    "olive_percent" = "olives_percent",
    "olive_orchards_percent" = "orchards_olives_percent",
    "paddy_fields" = "rice_paddy",
    "pine_plantations" = "plantations_pine",
    "eucalyptus_plantations" = "plantations_eucalyptus",
    "livestock_sheep_livestock_goat" = "livestock_sheep_goat",
    "livestock_cattle_livestock" = "livestock_cattle",
    "livestock_livestock_goat" = "livestock_goat",
    "cattle_abundance" = "cattle_density",
    "horse_abundance" = "horse_density",
    "sheep_abundance" = "sheep_density",
    "horticulural" = "horticultural",
    "young_evergreen_conifer_plantations" = "plantations_young_evergreen_conifer"
  )

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2621
```

3.2.2 Synthesizing silviculture predictor names

Edit predictor names related to silviculture (growing/cutting trees, agroforestry, clearcut areas)

```
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    "\\(cut_blocks\\)" = "",
    "^logging_areas$" = "cut-block_areas",
    "^logging_percent_\\(cut-blocks\\)$" = "cut-blocks_percent",
    "^logging_percent$" = "cut-blocks_percent",
    "cut_block" = "cut-block",
    "cut_blocks" = "cut-block",
    "cutblock" = "cut-block",
    "\\(\\)" = "",
    "clear-cut" = "clear_cut",
    "^cut-blocks_" = "logging_cut-block_",
    "^cut-block_" = "logging_cut-block_",
    "cut-block_presence" = "cut-block_areas",
    "cut-blocks" = "cut-block_areas",
    "block_features" = "logging_cut-block_areas",
    "logging_cut_size_ha" = "logging_cut-block_areas_size",
    "logging_cut-logging cut-block areas" = "logging cut-block areas",
    "^cut-block_areas$" = "logging cut-block areas",
    "clearcut_areas" =  "clear_cut_areas",
    "land_clearance" = "clear_cut_areas",
    "logging_cut-logging_cut-block_areas" = "logging_cut-block_areas",
    "saw_mills" = "sawmills",
    "industrial_logging" = "logging_industrial"
  )

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2616
```

3.2.3 Synthesizing energy/fuel/raw material predictor names

Edit predictor names related to energy, fuels and raw materials.

```
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    "coal_mines_" = "mines_",
    "lead_mines_" = "mines_",
    "unconventional oil and gas well pads" = "oil_gas_well_pads_unconventional",
    "mine_lands_" = "mines_",
    "mine_" = "mines_",
```

```
    "mining_areas" = "mines",
    "mining_sites" = "mines",
    "freatures" = "features",
    "transmission_line_" = "transmission_lines_",
    "electric_transmission_lines" = "electric_lines",
    "electric_wiring" = "electric_lines",
    "electric_line_" = "electric_lines_",
    "power_lines" = "powerlines",
    "precipitation_corrected_irrigation" =
      "precipitation_evaporation_corrected_irrigation",
    "submarine_pipelines_cables_extent" = "pipeline_submarine_cables_extent",
    "well_pads_percent" = "oil_well_pads_percent",
    "mines_lands" = "mines",
    "mines_sites" = "mines",
    "^pipelines_density$" = "oil_gas_pipeline_density",
    "oil_well_sites" = "oil_gas_well_sites",
    "oil_and_gas_well_density" = "oil_gas_well_density",
    "dams_distance_downstream" = "dams_downstream_distance",
    "powerlines_presence" = "powerlines",
    "petroleum" = "oil",
    "seismic_pipelines" = "seismic_lines"
  )

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2601
```

3.2.4 Synthesizing recreation/tourism predictor names

Edit predictor names related to recreation and tourism.

```
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    "trail_" = "trails_",
    "_trail$" = "_trails",
    "^path$" = "paths",
    "paths" = "trails",
    "footpath_presence" = "trails",
    "campsites" = "campground",
    "recreation_features" = "recreational_areas",
    "^gardens$" = "garden",
    "ski-lifts" = "ski-lift",
    "skilift" = "ski-lift",
    "ski_tracks_lifts_m" = "ski_tracks_and_lifts_length",
```

```
    "cableways_" = "cableway_"
  )

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2594
```

3.2.5 Synthesizing human habitat/infrastructure predictor names

Edit predictor names related to human habitation and infrastructure.

```
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    "human_area_" = "human_areas_",
    "urban_areas_ha" = "urban_areas_size",
    "urban_areas_m2" = "urban_areas_size",
    "urban_land_use" = "urban_areas",
    "urban_land_cover" = "urban_areas",
    "urban_land" = "urban_areas",
    "urban_fraction" = "urban_areas_percent",
    "urban_percent" = "urban_areas_percent",
    "urban_count" = "urban_areas_count",
    "^rban_area_" = "urban_areas_",
    "^urban_area_" = "urban_areas_",
    "^urban_areas_presence" = "urban_areas",
    "urban_areas_size_km" = "urban_areas_size",
    "^urban_zone_percent$" = "urban_areas",
    "^urban_areas_proportion$" = "urban_areas_percent",
    "^urban_settlement$" = "settlements_urban",
    "urban_settlements" = "settlements_urban",
    "^human_settlements" = "settlements",
    "settelments" = "settlements",
    "town_" = "towns_",
    "city_" = "cities_",
    "village_" = "villages_",
    "^ developed_exposed_land_percent$" = "developed_land_exposed_percent",
    "^garden$" = "garden_presence_absence",
    "^developed_land$" = "developed_areas",
    "^developed_high_intensity$" = "developed_areas_intensity_high",
    "^developed_med_intensity$" = "developed_areas_intensity_medium",
    "^developed_low_intensity$" = "developed_areas_intensity_low",
    "^development_intensity_high$" = "developed_areas_intensity_high",
    "^developedment_intensity_low$" = "developed_areas_intensity_low",
    "^distance_village$" = "village_distance",
```

```
    "^rural_settlements_" = "settlements_rural_",
    "^rural_village$" = "villages_rural",
    "small_villages" = "villages_small",
    "small_settlements" = "settlements_small",
    "^settlement_distance$" = "settlements_distance",
    "^distance_settlements$" = "settlements_distance",
    "^built-up_area$" = "built-up_areas",
    "^built-up_area_" = "built-up_areas_",
    "^built-up_area__density_50m_buffer$" = "built-up_areas_density_50m_buffer",
    "village_" = "villages_",
    "urban_area_" = "urban_areas_",
    "^urban$" = "urban_areas",
    "^urban_area$" = "urban_areas",
    "buildings_" = "buildings_",
    "protected_area_" = "protected_areas_",
    "non-linear_footprint" = "human_footprint_non-linear",
    "non-agricultural_footprint" = "human_footprint_agricultural",
    "building_" = "buildings_",
    "buildings_number_per_km" = "buildings_count",
    "isolated_houses_and_roads_" = "houses_and_roads_isolated_",
    "number_separate_parcels_used_per_household" = "household_separate_parcels_used_count",
    "distance_villages_classes" = "villages_distance_class",
    "land-use_built-up" = "built-up",
    "main_cities_" = "cities_main_",
    " \\(gain or loss\\)" = "_gain_loss",
    "settlement_" = "settlements_",
    "south-facing_walls" = "walls_south-facing",
    "native_american" = "Native_American",
    "_american_" = "_American_",
    "proportion_human_land_use" = "human_land_use_proportion",
    "transfromation" = "transformation",
    "urbanization_" = "urban_areas_",
    "small_cities_" = "cities_small_",
    "domestic_garden_cover_" = "garden_",
    "conventration_impervious_surfaces" = "impervious_surfaces_percent",
    "developed_area_percent" = "developed_areas_percent",
    "residential_area_percent" = "residential_areas_percent",
    "restricted_areas_military" = "military_restricted_areas",
    "weighted-mean" = "weighted_mean",
    "adjacent_land_cover" = "land_cover_adjacent",
    "^light_pollution$" = "nighttime_light_intensity",
    "^light_pollution_degree$" = "nighttime_light_intensity",
    "^wells_" = "well_",
    "wasteland_size" = "wasteland_area_size",
    "worst_housing_conditions_" = "housing_conditions_worst_",
    "best_housing_conditions_" = "housing_conditions_best_",
    "local_situation_urban_or_landscape" = "urban_areas_or_landscape_local",
    "low-density_urban_areas" = "urban_areas_low-density",
    "low-intensity_urban_areas" = "urban_areas_low-intensity",
    "low-intensity_urban_percent" = "urban_areas_low-intensity_percent",
    "low-intensity_developed_area_" = "developed_areas_low-intensity_",
    "low-intensity_developed_areas_" = "developed_areas_low-intensity_",
    "developed_area_low_intensity_" = "developed_areas_low-intensity_",
```

```
    "low-intensity_development_" = "developed_areas_low-intensity_",
    "low-intensity_land_use_" = "land_use_low-intensity_",
    "anthropogenic_land_use" = "anthropogenic_land",
    "antropogenic_features" = "anthropogenic_structures",
    "arable_area_size" = "arable_land_area_size",
    "arable_area" = "arable_land",
    "arable_fields_presence" = "arable_fields",
    "arable_fields" = "arable_land",
    "medium-intensity_urban_areas" = "urban_areas_medium-intensity",
    "medium-intensity_urban_percent" = "urban_areas_medium-intensity_percent",
    "medium-intensity_developed_area_" = "developed_areas_medium-intensity_",
    "medium-intensity_developed_areas_" = "developed_areas_medium-intensity_",
    "developed_area_medium_intensity_" = "developed_areas_medium-intensity_",
    "medium_or_high-intensity_development_" = "developed_areas_medium_or_high-intensity_",
    "medium_or_high-intensity_land_use_" = "land_use_medium_or_high-intensity_",
    "artificial_light_intensity" = "nighttime_light_intensity",
    "nightlight" = "nighttime_light",
    "artificial_light_intensity" = "nighttime_light_intensity",
    "nightime_light_intensity" = "nighttime_light_intensity",
    "anthropogenic_night_lights" = "nighttime_light_intensity",
    "^nighttime_light$" = "nighttime_light_intensity",
    "^nightlight$" = "nighttime_light_intensity",
    "urban_brownfield_" = "urban_brownfields_",
    "natural_or_unnatural_burn" = "burn_natural_or_unnatural",
    "build-up" = "built-up",
    "built_area_" = "built-up_areas_",
    "buildings_number_per_km" = "buildings_frequency_1km_radius",
    "built-up_and_urban_areas_distance" = "urban_and_built-up_areas_distance",
    "dense_urban_areas" = "urban_areas_high-intensity",
    "developed_area_" = "developed_areas_",
    "intensity_high" = "high-intensity",
    "intensity_medium" = "medium-intensity",
    "intensity_low" = "low_high-intensity",
    "developed_land" = "developed_areas",
    "developed_exposed_land" = "developed_areas_exposed",
    "developed_high-intensity" = "developed_areas_high-intensity",
    "high-intensity_developed_area_" = "developed_areas_high-intensity_",
    "^high-intensity_developed_areas" = "developed_areas_high-intensity",
    "high-intensity_development" = "developed_areas_high-intensity",
    "high-intensity_urban_areas" = "urban_areas_high-intensity",
    "high-intensity_urban_percent" = "urban_areas_high-intensity_percent",
    "developed_medium-intensity" = "developed_areas_medium-intensity",
    "developed_low-intensity" = "developed_areas_low-intensity",
    "^development_" = "developed_areas",
    "high-intensity_development_" = "developed_areas_high-intensity_",
    "development_medium_area_" = "developed_areas_medium-intensity_area",
    "developmed" = "developed",
    "^developed_p" = "developed_areas_p",
    "^developed_moderate_intensity" = "developed_areas_medium-intensity",
    "diffuse_urban_areas" = "urban_areas_diffuse",
    "discontinous" = "discontinuous",
    "urban_areas_discontinuous" = "discontinuous_urban_fabric",
    "^settlements_areas$" = "settlements",
```

```
    "settlments_" = "settlements_",
    "settlment_" = "settlements_",
    "short-cut_lawn" = "lawn_short_cut",
    "green_urban_areas" = "urban_green_space",
    "^unihabited_villages" = "villages_uninhabited",
    "human_activities" = "human_activity",
    "human-dominated_area_" = "human-dominated_areas_",
    "human_dominated_areas_" = "human-dominated_areas_",
    "industrial_land" = "industrial_areas",
    "human_habitation" = "settlements",
    "inhabited_areas" = "settlements",
    "human_inhabited_areas" = "settlements",
    "urban_park_" = "urban_parks_",
    "human_populated_area_" = "human_populated_areas_",
    "human_population_center_" = "human_population_centers_",
    "human_use_area" = "human_use",
    "impervious_land_cover" = "impervious_surfaces",
    "impervious_surface_" = "impervious_surfaces_",
    "imperviousness_percent" = "imperviousness",
    "imperviousness_index" = "imperviousness",
    "impervous" = "impervious",
    "suburban_d" = "suburban_areas_d",
    "suburban_p" = "suburban_areas_p",
    "^settlement$" = "settlements",
    "residential_area_density" = "residential_areas_density",
    "developed_areaspercent" = "developed_areas_percent",
    "developed_areasopen" = "developed_areas_open",
    "developed_areasmedium" = "developed_areas_medium",
    "developed_areaslow" = "developed_areas_low",
    "developed_areashigh" = "developed_areas_high",
    "lit-up_areas" = "artificial_illumination",
    "rural_land" = "rural_areas",
    "cemetaries" = "cemeteries",
    "^human_settlement$" = "settlements",
    "human_use" = "human_land_use",
    "^human_footprint$" = "human_footprint_index",
    "^human_settlements$" = "settlements"
    )


# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2487
```

3.2.6 Synthesizing transportation/human movement predictor names

Edit predictor names related to transportation infrastructure and human movement (e.g. bridges, roads, highways, airports, railways, canals, linear features, boat traffic, traffic, shipping, streets).

```
# create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    "railroads_" = "railways_",
    "railroad_" = "railways_",
    "railway_" = "railways_",
    "^railway$" = "railways",
    "railways_tracks" = "railway_tracks",
    "^railways_length_per_cell" = "railways_length",
    "^motorway_length$" = "highways_length",
    "^secondary_roads_distance$" = "roads_secondary_distance",
    "^distance_roads$" = "roads_distance",
    "^road_distance$" = "roads_distance",
    "^distance_minor_roads$" = "roads_minor_distance",
    "^distance_major_roads$" = "roads_major_distance",
    "^road_length_per_cell$" = "roads_length",
    "^road_proximity$" = "roads_distance",
    "^road_density$" = "roads_density",
    "autonomic_roads_" = "roads_autonomic_",
    "main_roads" = "roads_main",
    "major_roads" = "roads_major",
    "national_roads" = "roads_national",
    "primary_road_" = "roads_primary_",
    "road_" = "roads_",
    "roads_access_number" = "roads_access_count",
    "roads_distancec" = "roads_distance",
    "winter_roads" = "roads_winter",
    "number_roads_access" = "roads_access_number",
    "secondary_roads_" = "roads_secondary_",
    "^sealed_roads_" = "roads_paved_",
    "unsealed_roads_" = "roads_unpaved_",
    "highway_" = "highways_",
    "^interstate_highways_" = "highways_interstate_",
    "dam_" = "dams_",
    "port_" = "ports_",
    "ports_proximity" = "ports_distance",
    "conventional_roads" = "roads_conventional",
    "number_roads_lake_perimeter" = "roads_lake_perimeter_count",
    "concentration_impervious_surfaces" = "impervious_surfaces_percent",
    "manmade_surfaces_percent" = "artificial_surfaces_percent",
    "concrete_areas_percent" = "impervious_surfaces_percent",
    "impervious_surface_percent" = "impervious_surfaces_percent",
    "tarred_areas_percent" = "impervious_surfaces_percent",
    "roads_frequency" = "roads_count",
    "^accessibility$" = "human_accessibility",
    "wide_roads" = "roads_wide",
    "infrastructures" = "infrastructure",
    "^local_roads_" = "roads_local_",
    "^main_roads_" = "roads_main_",
    "^major_and_local_roads_" = "roads_major_and_local_",
```

```
    "^major_roads_" = "roads_major_",
    "^minor_roads_" = "roads_minor_",
    "^narrow_roads_" = "roads_narrow_",
    "^minor_street_" = "street_minor_",
    "^major_street_" = "street_major_",
    "street_length_m" = "street_length",
    "asphalt_roads" = "roads_paved",
    "roads_non-asphalted" = "roads_unpaved",
    "^paved_road_" = "roads_paved_",
    "^paved_roads_" = "roads_paved_",
    "^pavement_area" = "paved_area",
    "boat_ramp_" = "boat_launch_",
    "number_boat_launch" = "boat_launch_count",
    "county_roads" = "roads_county",
    "^primary_roads" = "roads_primary",
    "^primative_roads" = "roads_primative",
    "roads_20km_buffer_percent" = "roads_percent_20km_radius",
    "roads_length_upaved" = "roads_upaved_length",
    "expressway" = "highway",
    "scrub_to_roadway_distance" = "road_to_scrub_distance",
    "^track_distance" = "tracks_distance",
    "tertiary_roads" = "roads_tertiary",
    "gravel_roads" = "roads_gravel",
    "highways_and_roads" = "roads_and_highways",
    "roads_km" = "roads_length",
    "unpaved_track" = "roads_unpaved",
    "federal_avian" = "federal_aviation",
    "motorway" = "highway",
    "railways_and_roads_" = "roads_and_railways_",
    "^unpaved_roads_" = "roads_unpaved_",
    "railway_tracks" = "railways",
    "railway_track_" = "railways_",
    "railways_track_" = "railways_",
    "roads_length_unpaved" = "roads_unpaved_length"
  )

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2426
```

3.2.7 Synthesizing socio-economic predictor names

Edit predictor names related to economics/growth (education, poverty, unemployment, population, retired, renters, homeowners).

```r
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    "^human_population$" = "human_population_density",
    "^human_populationn" = "human_population",
    "^human_populationN" = "human_population",
    "^humans_count_" = "human_population_density_",
    "^humans_per_km2_" = "human_population_density_1km_radius",
    "rural_human_population_density" = "human_population_density_rural",
    "^populated_areas_" = "human_populated_areas_",
    "number_inhabitants_nearest_village" = "inhabitants_nearest_village_count",
    "^own_>100k" = "town_>100k",
    "^>65yrs_percent$" = "residents_>65yrs_percent",
    "farm_forestry_fishing_profession_percent" = "profession_farm_forestry_fishing_percent",
    "^population_" = "human_population_",
    "^year_housing_" = "housing_year_",
    "^year_moved_" = "housing_year_moved_",
    "^income" = "household_income",
    "white_households_" = "households_white_",
    "residents_with_bachelor_degrees_" = "education_bachelors_above_",
    "bachelors_above_" = "bachelors_and_above_",
    "school_below_" = "school_and_below_",
    "_km2reef" = "_km2_reef",
    "\\(gainorloss\\)" = "_gain_or_loss",
    "_number$" = "_count",
    "poopulation" = "population",
    "town_>100k_residents_distance" = "towns_>100k_residents_distance",
    "towns_>100k_residents_distance" = "towns_>100k_inhabitants_distance",
    "towns_distance_>100k_inhabitants" = "towns_>100k_inhabitants_distance",
    "towns_>500k_residents_distance" = "towns_distance_>500k_inhabitants",
    "towns_distance_>500k_inhabitants" = "towns_>500k_inhabitants_distance",
    "full-time_farmers" = "farmers_full-time",
    "part-time_farmers" = "farmers_part-time",
    "rutal" = "rural",
    "inhabitants_density" = "residents_density",
    "inhabitant_density" = "residents_density",
    "infuence" = "influence",
    "global_human" = "human",
    "industrian" = "industrial",
    "human_population_settlements" = "human_population_density_settlements",
    "urban_and_transport_land_cover_percent" = "urban_and_transport_percent"
  )

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2410
```

3.2.8 Synthesizing land loss/degradation/abandonment predictor names

Edit predictors related to land loss/degradation/abandonment.

```
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c("disturbed_forest_cover" = "forest_cover_disturbed",
                "^forest$" = "forest_presence_absence",
                "forest_fragmentations" = "forest_fragmentation",
                "semi_natural_" = "semi-natural_",
                "perterbation" = "perturbation",
                "forest_patch_" = "forest_",
                "past_deforestation_area_" = "deforestation_area_historic",
                "forested_non_forested_deforested_class" = "deforested_area",
                "forested_non_forested" = "forest_non-forest",
                "^unexploited_area$" = "unexploited_areas",
                "forested_non-forested" = "forest_non-forest",
                "deforestation_area_historicdistance" = "deforestation_historic_distance"
  )

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2408
```

3.2.9 Synthesizing conservation/management predictor names

Edit predictor names related to protection, conservation, and management.

```
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    "distance_protected_areas" = "protected_areas_distance",
    "distance_non-hunting_reserve" = "non-hunting_reserve_distance",
    "no-hunting_area_distance" = "non-hunting_area_distance",
    "hunting_area_distance" = "hunting_areas_distance",
    "regions" = "areas",
    "^protected_area$" = "protected_areas",
    "^protected_areas_presence$" = "protected_areas",
    "^release_point$" = "release_site",
    "^release_distance$" = "release_site_distance",
    "^protected_areas_presence$" = "protected_areas",
    "introduced_site" = "species_introduction_site",
    "reintroduction_site_nucleus_distance" = "species_introduction_site_distance",
    "introduction_locus_distance" = "species_introduction_site_distance",
    "introduced_site_distance" = "species_introduction_site_distance",
    "release_point_distance" = "species_introduction_site_distance",
    "release_site_distance" = "species_introduction_site_distance"
```

```
  )

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2399
```

3.2.10 Synthesizing pollution predictor names

Edit predictor names related to pollution.

```
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    "nighttime_" = "night_",
    "anthropogenic_night_lights" = "night_light_intensity",
    "nightime_light_intensity" = "night_light_intensity",
    " polyaromatic_" = "polyaromatic_",
    "maximum_chlorophyll-a" = "chlorophyll-a_maximum",
    "minimum_chlorophyll-a" = "chlorophyll-a_minimum",
    "\\(streamflow\\)" = "",
    "_incidents_number_" = "_incidents_count_",
    "_runoff\\(\\)$" = "_runoff",
    "barrier_current" = "barrier")

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2398
```

3.2.11 Synthesizing ambiguous use/cover predictor names

Edit predictors related to generalized use/cover.

```
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    "land_cover_land_use" = "land_use/land_cover",
    "land_use_land_cover" = "land_use/land_cover",
```

```
    "land_use/land_cover_mode" = "land_use/land_cover",
    "^land_use_land_cover$" = "land_use/land_cover",
    "^land_use$" = "land_use/land_cover",
    "^land_cover$" = "land_use/land_cover",
    "^land_Cover$" = "land_use/land_cover",
    "^land_use_type$" = "land_use/land_cover",
    "^land_use_1976$" = "land_use/land_cover",
    "^land_use_1990$" = "land_use/land_cover",
    "^land_use_1996$" = "land_use/land_cover",
    "^land_use_2000$" = "land_use/land_cover",
    "^land_use_2003$" = "land_use/land_cover",
    "^land_use_2050$" = "land_use/land_cover",
    "^land_use_2070$" = "land_use/land_cover",
    "^land_cover_type$" = "land_use/land_cover",
    "land_use_type" = "land_use/land_cover",
    "land_cover_patches" = "land_cover_patch",
    "land_use_change" = "land_use_change_percent",
    "land_use_count" = "land_use_class_count",
    "land_use_sum" = "land_use_richness",
    "historic_19" = "historic_yr19",
    "land_use_yr" = "land_use_historic_yr",
    "landcover_type" = "land_use/land_cover",
    "land_use_type" = "land_use/land_cover",
    "land-use_" = "land_use_",
    "habitat_type" = "land_use/land_cover",
    "^land_cover_change$" = "land_cover_change_rate",
    "_contagion_index" = "contagion",
    "landscape_condition$" = "landscape_condition_index",
    "diveristy" = "diversity",
    "diversity_index" = "diversity",
    "_heteogeneity" = "_heterogeneity",
    "land_cover_heterogeneity" = "land_cover_diversity",
    "historic_land_use" = "land_use_historic",
    "land_cover_type_dominant" = "land_cover_dominant",
    "land_cover_dominant_class" = "land_cover_dominant",
    "^non-forested$" = "forest_non-forest",
    "primary_land_cover" = "land_cover_dominant",
    "land_cover_and_land_use/land_cover" = "land_use/land_cover",
    "land_use_class" = "land_use/land_cover",
    "land_covercontagion" = "land_cover_contagion",
    "^land_cover_" = "land_use/land_cover_",
    "^land_Cover_" = "land_use/land_cover_"
    )

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2371
```

3.2.12 Synthesizing additional predictor name patterns

Edit predictors with years in names.

```r
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    "_1900" = "_yr1900",
    "_1950" = "_yr1950",
    "_1985" = "_yr1985",
    "_2021" = "_yr2021",
    "_2035" = "_yr2035",
    "_2050" = "_yr2050",
    "_2060" = "_yr2060",
    "_2070" = "_yr2070",
    "_2100" = "_yr2100",
  # deleting future years because years are implied in SDM
    "_yr2021" = "",
    "_yr2035" = "",
    "_yr2050" = "",
    "_yr2060" = "",
    "_yr2070" = "",
    "_yr2100" = "")

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2357
```

Edit predictors using radii, buffers, and other size indicators.

```r
# Create a vector of patterns to search and replace (search on left, replace on right)
  patterns <- c(
    "buffer" = "radius",
    "100ha" = "100ha_radius",
    "_10m$" = "_10m_radius",
    "_25m$" = "_25m_radius",
    "_50m$" = "_50m_radius",
    "_100m$" = "_100m_radius",
    "_165m$" = "_165m_radius",
    "_315m$" = "_315m_radius",
    "_500m$" = "_500m_radius",
    "_615m$" = "_615m_radius",
    "_1215m$" = "_1215m_radius",
```

```
    "_4.5km$" = "_4.5km_radius",
    "_5km$" = "_5km_radius",
    "_20km$" = "_20km_radius",
    "proportion" = "percent",
    "fraction" = "percent",
    "_m2$" = "_area_size",
    "_m$" = "_length",
    "_meters$" = "_length",
    "_m^3" = "_volume",
    "_size_ha$" = "_area_size",
    "_area_ha$" = "_area_size",
    "average" = "mean",
    "patches" = "patch",
    "quantity" = "count",
    "hectares" = "area_size",
    "_presence_absence" = "",
    "_presence-absence" = "",
    "distane" = "distance",
    "desnity" = "density",
    "aeas" = "areas",
    "_countes$" = "count",
    "country_boundry" = "country_boundary",
    "areaspercent" = "areas_percent",
    "percentn" = "percent",
    "m^3" = "volume",
    "distubance" = "disturbance",
    "radius_radius" = "radius",
    "disconinuous" = "discontinuous"
    )

# for-loop of edits
  for (pattern in names(patterns)) {
    preds.list.df <- data.frame(lapply(preds.list.df, function(x) {
      gsub(pattern, patterns[pattern], x)
    }))
  }

# get new count of predictor list
  length(unique(preds.list.df$predictor))
```

```
## [1] 2311
```

Edit lines with typos in semi-colons.

```
preds.list.df <- separate_rows(preds.list.df,predictor,sep=';')
```

Delete rows that are only blank spaces.

```
# delete extra rows with blanks
  preds.list.df <- preds.list.df[!(is.na(preds.list.df$predictor) |
                                   preds.list.df$predictor==""), ]
```

Convert all underscores (_) to spaces.

```
# remove extra spaces in front of strings
  preds.list.df <- data.frame(lapply(preds.list.df, function(x) {gsub("^ ","", x)}))

# convert underscores to spaces
  preds.list.df <- data.frame(lapply(preds.list.df, function(x) {gsub("_"," ", x)}))

# remove extra spaces in between strings
  preds.list.df <- data.frame(lapply(preds.list.df, function(x) {gsub("  "," ", x)}))
  preds.list.df <- data.frame(lapply(preds.list.df, function(x) {gsub("  "," ", x)}))
```

Get *final count* of unique predictor names.

```
# inspect
  length(unique(preds.list.df$predictor))
```

```
## [1] 2307
```

3.3 Summary table of predictors used

```
# Get a count of predictors, sorted by past/present/future
  library("plyr")
  preds.list.short <- ddply(preds.list.df, .(timeframe, predictor), summarize,
                            count=length(predictor))
  str(preds.list.short)
```

```
## 'data.frame':    2535 obs. of  3 variables:
##  $ timeframe: chr  "future" "future" "future" "future" ...
##  $ predictor: chr  "abandoned areas percent" "agricultural areas heterogeneous percent" "agricultural
##  $ count    : int  1 1 5 1 1 1 2 3 1 1 ...
```

```
# Remove repeated predictors for each UID and time frame
  preds.list.uniq <- preds.list.df[!duplicated(
                              preds.list.df[,c('uid','predictor','timeframe')]),]

# Get a list of unique predictors, timeframes, and papers
  preds.list.shorter <- ddply(preds.list.uniq, .(predictor),
                             summarize,
                             # list of paper UIDs that used the predictors
                             papers=paste(unique(uid),collapse="; "),
                             # list of time frames for which they were used
                             timeframes=paste(unique(timeframe),collapse="; "),
                             # count number of papers using each predictor
                             count=paste(length(unlist(strsplit(papers, ";"))))
                             )

# get structure
  options(width=85) # ensure width
  str(preds.list.shorter)
```

```
## 'data.frame':    2307 obs. of  4 variables:
##  $ predictor : chr  "abandoned agricultural areas new distance" "abandoned agricultural areas old di
##  $ papers    : chr  "8233" "8233" "11080" "9901" ...
##  $ timeframes: chr  "present" "present" "past; present; future" "present" ...
##  $ count     : chr  "1" "1" "1" "1" ...
```

```
# inspect as needed
  #summary(preds.list.shorter$predictor)

# save
  write.csv(preds.list.shorter, paste0(data.dir,"predictor_list_summary.csv"),
            row.names = FALSE)
```

3.4 Top 10 human predictors

Here, we show how to extract the top 10 human predictors being used in SDM studies, but the full list is available in the Supplementary Materials corresponding to this published article.

```
# read again (no row names anymore)
  preds.list.shorter <- read.csv(paste0(data.dir,"predictor_list_summary.csv"),
                                 header = TRUE)

# show table of predictors used more than twice
  preds.list.short2 <- subset(preds.list.shorter[preds.list.shorter$count>=2,])

# Sort the predictors by most frequent, followed by name
  preds.list.short2 <- preds.list.short2[order(-preds.list.short2$count,
                                               preds.list.short2$predictor),]

# Show the top 10
  kableExtra::kbl(preds.list.short2[1:10,], booktabs=T, longtable=T) %>%
    kable_styling(latex_options = c("striped","repeat_header")) %>%
    column_spec(1, width="2em") %>%
    column_spec(2, bold=F, color="black", border_right=F, width="10em") %>%
    column_spec(3, width="18em") %>%
    column_spec(4, width="4em") %>%
    column_spec(5, width="3em")
```

(continued)

	predictor	papers	timeframes	count
	predictor	papers	timeframes	count
1135	land use/land cover	1465; 1842; 4528; 4683; 5096; 6083; 6597; 8067; 8225; 8345; 8489; 11123; 11290; 11370; 92; 106; 129; 144; 178; 191; 200; 204; 254; 279; 326; 408; 430; 498; 499; 521; 651; 655; 667; 682; 706; 707; 760; 762; 774; 851; 855; 869; 875; 899; 912; 922; 936; 940; 952; 956; 985; 996; 1099; 1110; 1121; 1130; 1159; 1184; 1201; 1208; 1222; 1230; 1271; 1297; 1366; 1397; 1462; 1527; 1534; 1554; 1573; 1583; 1589; 1616; 1620; 1654; 1669; 1686; 1726; 1735; 1751; 1752; 1765; 1777; 1798; 1924; 1965; 1990; 1993; 2010; 2026; 2029; 2092; 2104; 2109; 2129; 2141; 2154; 2163; 2186; 2207; 2222; 2233; 2283; 2338; 2366; 2367; 2400; 2419; 2468; 2472; 2478; 2484; 2513; 2610; 2658; 2669; 2736; 2758; 2760; 2859; 2864; 2879; 2929; 2987; 2998; 3073; 3104; 3112; 3142; 3208; 3279; 3303; 3325; 3327; 3344; 3352; 3354; 3366; 3382; 3405; 3424; 3531; 3540; 3601; 3612; 3732; 3734; 3781; 3793; 3854; 4021; 4123; 4133; 4159; 4188; 4190; 4224; 4226; 4234; 4243; 4373; 4396; 4474; 4537; 4601; 4651; 4656; 4671; 4692; 4762; 4798; 4846; 4850; 4861; 4886; 4893; 4905; 4913; 4934; 4943; 4944; 4953; 5098; 5125; 5228; 5284; 5308; 5389; 5393; 5442; 5470; 5490; 5512; 5542; 5558; 5589; 5643; 5648; 5671; 5680; 5690; 5761; 5775; 5783; 5833; 5836; 5846; 5896; 5936; 5948; 5968; 6051; 6102; 6142; 6211; 6224; 6282; 6296; 6347; 6349; 6359; 6421; 6471; 6484; 6527; 6544; 6599; 6613; 6721; 6864; 6869; 6877; 6879; 6939; 6959; 7021; 7082; 7144; 7188; 7190; 7192; 7195; 7216; 7240; 7273; 7306; 7318; 7338; 7386; 7387; 7435; 7449; 7450; 7483; 7512; 7537; 7564; 7573; 7575; 7612; 7823; 7879; 7912; 7959; 7963; 7973; 8011; 8079; 8103; 8105; 8108; 8149; 8192; 8193; 8261; 8287; 8342; 8370; 8384; 8407; 8409; 8420; 8464; 8482; 8504; 8509; 8544; 8565; 8577; 8670; 8691; 8712; 8780; 8802; 8810; 8822; 8832; 8836; 8846; 8864; 8935; 8936; 8970; 8976; 9024; 9036; 9042; 9071; 9072; 9082; 9107; 9117; 9148; 9174; 9199; 9235; 9278; 9295; 9358; 9383; 9412; 9416; 9420; 9458; 9545; 9546; 9576; 9579; 9641; 9660; 9685; 9710; 9714; 9733; 9760; 9785; 9808; 9810; 9854; 9864; 9941; 9973; 9986; 10035; 10065; 10069; 10148; 10185; 10342; 10402; 10493; 10505; 10524; 10636; 10657; 10688; 10692; 10706; 10707; 10738; 10747; 10807; 10813; 10898; 10920; 11008; 11032; 11037; 11057; 11075; 11117; 11143; 11217; 11226; 11244; 11271; 11348; 11355; 11364; 11386;	past; present; future	397

(continued)

	predictor	papers	timeframes	count
1728	roads distance	1465; 4528; 6083; 6597; 8067; 8345; 10897; 92; 106; 107; 144; 198; 297; 298; 352; 523; 643; 682; 684; 706; 753; 773; 922; 944; 995; 1017; 1159; 1213; 1228; 1230; 1324; 1446; 1516; 1725; 1752; 1880; 1924; 1964; 2152; 2186; 2274; 2278; 2294; 2317; 2326; 2461; 2479; 2736; 2798; 2905; 2984; 2991; 3019; 3077; 3112; 3125; 3142; 3219; 3279; 3283; 3356; 3424; 3438; 3540; 3550; 3589; 3613; 3688; 3694; 3781; 3782; 3783; 3786; 3850; 3859; 3979; 3988; 4006; 4011; 4023; 4164; 4174; 4230; 4237; 4263; 4396; 4592; 4620; 4722; 4784; 4798; 4850; 4866; 5098; 5124; 5228; 5323; 5334; 5382; 5389; 5431; 5476; 5490; 5592; 5642; 5717; 5784; 5785; 5833; 5835; 5836; 5848; 5885; 5992; 6037; 6079; 6156; 6167; 6422; 6485; 6527; 6542; 6573; 6578; 6586; 6926; 6940; 6979; 7216; 7219; 7273; 7387; 7582; 7605; 7612; 7633; 7798; 7802; 7912; 7944; 7950; 7962; 7964; 7993; 8149; 8153; 8173; 8261; 8351; 8400; 8404; 8420; 8516; 8548; 8691; 8708; 8712; 8713; 8724; 8768; 8864; 8976; 9163; 9182; 9185; 9186; 9190; 9252; 9341; 9412; 9487; 9532; 9552; 9562; 9584; 9675; 9710; 9807; 9842; 9854; 9908; 9941; 9973; 10008; 10118; 10187; 10195; 10342; 10349; 10478; 10493; 10505; 10585; 10607; 10636; 10649; 10657; 10692; 10814; 10853; 10924; 10959; 10975; 11093; 11217; 11270; 11345; 11379; 11393; 11397; 11404; 11538; 11544; 11794; 11871; 11882; 12019; 12036; 12037; 12097; 12200; 12241; 12243; 12480; 12481	past; present; future	225

(continued)

	predictor	papers	timeframes	count
1001	human population density	3535; 5555; 6083; 6569; 6966; 10738; 11290; 11401; 18; 171; 188; 212; 297; 298; 373; 408; 410; 415; 510; 552; 622; 640; 643; 684; 706; 760; 827; 848; 952; 1044; 1127; 1143; 1171; 1228; 1249; 1336; 1418; 1611; 1776; 1880; 1882; 1939; 1992; 2087; 2154; 2259; 2313; 2376; 2410; 2445; 2479; 2484; 2509; 2510; 2517; 2670; 2736; 2863; 2888; 2976; 3023; 3028; 3112; 3130; 3257; 3279; 3283; 3525; 3585; 3617; 3675; 3684; 3729; 3732; 3808; 3979; 4034; 4039; 4050; 4143; 4153; 4164; 4243; 4441; 4479; 4537; 4557; 4748; 4782; 4787; 4900; 4905; 4949; 5192; 5228; 5264; 5644; 5672; 5680; 5717; 5763; 5805; 5846; 5896; 5903; 5948; 5976; 6343; 6360; 6444; 6471; 6585; 6586; 6622; 6869; 6940; 6978; 7016; 7176; 7195; 7226; 7306; 7483; 7537; 7605; 7841; 7884; 7944; 7973; 7991; 8016; 8093; 8192; 8261; 8342; 8428; 8464; 8486; 8524; 8603; 8654; 8691; 8840; 8858; 8935; 8953; 8970; 9022; 9024; 9190; 9199; 9412; 9534; 9569; 9579; 9673; 9733; 9785; 9938; 9939; 10051; 10118; 10315; 10478; 10686; 10747; 10813; 10827; 10920; 10951; 11008; 11226; 11309; 11515; 11659; 11877; 12044; 12060; 12109; 12181; 12322; 12434; 12439	past; present; future	183
80	agricultural areas percent	691; 2791; 5713; 6569; 6966; 45; 61; 66; 327; 720; 730; 848; 863; 1173; 1245; 1433; 1664; 2095; 2335; 2338; 2395; 2484; 2730; 2885; 2991; 3152; 3585; 3609; 3805; 3850; 3912; 4179; 4355; 4724; 4748; 4903; 5124; 5126; 5264; 5390; 5721; 5728; 5763; 5863; 5948; 6126; 6375; 6485; 6533; 6538; 6586; 6769; 6900; 6901; 6929; 6974; 7172; 7492; 7880; 7884; 7993; 8084; 8147; 8231; 8722; 8750; 9025; 9125; 9415; 9426; 9534; 9552; 9557; 9562; 9590; 9748; 9938; 10039; 10515; 10742; 10827; 10899; 10935; 11125; 11439; 11508; 11543; 11794; 11931; 12008; 12246; 12283	past; present; future	92

(continued)

	predictor	papers	timeframes	count
1721	roads density	1465; 11290; 11400; 45; 566; 643; 848; 863; 932; 1245; 1324; 1408; 1583; 1722; 2271; 2349; 2758; 2984; 3130; 3551; 3612; 3613; 3617; 3638; 3642; 3675; 3684; 4021; 4034; 4143; 4188; 4230; 4396; 4563; 4898; 5098; 5449; 5555; 5590; 5599; 5680; 5763; 6282; 6362; 6536; 6865; 7393; 7424; 7461; 7483; 7677; 8059; 8173; 8404; 8428; 8565; 8691; 8736; 8780; 8860; 8935; 8953; 9022; 9036; 9190; 9534; 9675; 9760; 9878; 9909; 10118; 10126; 10314; 10390; 10730; 10742; 10813; 10827; 10921; 11272; 11538; 11539; 11546; 11877; 12060; 12067; 12243; 12380; 12439	past; present	89
2112	urban areas percent	691; 2791; 6569; 6966; 61; 175; 552; 566; 673; 730; 848; 1244; 1484; 1658; 1683; 1711; 1895; 2015; 2147; 2196; 2670; 3510; 3581; 3585; 3780; 3804; 3805; 3850; 3961; 4032; 4065; 4215; 4241; 4511; 4903; 5264; 5294; 5390; 5512; 5644; 5648; 5721; 5863; 5948; 5977; 6445; 6769; 6974; 7185; 7207; 7265; 7605; 7646; 7677; 7898; 7991; 8087; 8231; 8429; 8446; 8648; 8852; 8858; 8974; 9125; 9426; 9748; 9878; 9991; 10035; 10529; 10707; 10935; 11063; 11125; 11422; 11871; 12044; 12109; 12175; 12256; 12283; 12434	past; present; future	83
979	human footprint index	1466; 1626; 71; 95; 200; 254; 297; 326; 373; 408; 554; 566; 577; 651; 718; 753; 1044; 1099; 1109; 1201; 1202; 1210; 1217; 1249; 1462; 1669; 1742; 1762; 1992; 2213; 2267; 2456; 2500; 2530; 2630; 2935; 3290; 3303; 3405; 3682; 3781; 3843; 3951; 4143; 4350; 4731; 5114; 5358; 5457; 5554; 5562; 5775; 5903; 5935; 5999; 6375; 6439; 6485; 7252; 7432; 7874; 7889; 7913; 8222; 8502; 8750; 9035; 9278; 9521; 9957; 10051; 10666; 10706; 10984	past; present; future	74
1889	settlements distance	1465; 3950; 106; 279; 297; 706; 800; 869; 1017; 1027; 1213; 1228; 1295; 1725; 1891; 1924; 2274; 2278; 2787; 3112; 3142; 3219; 3283; 3424; 3912; 3960; 4172; 4441; 4717; 4722; 4739; 4798; 5382; 5489; 5781; 5784; 5835; 5992; 6079; 6156; 6522; 6536; 6792; 6926; 6947; 7273; 7387; 7618; 7912; 7944; 7950; 8016; 8955; 9182; 9185; 9186; 9458; 9579; 9748; 10008; 10813; 11057; 11345; 11474; 11625	past; present; future	65

(continued)

	predictor	papers	timeframes	count
2097	urban areas distance	107; 204; 375; 552; 709; 753; 800; 813; 827; 1130; 1222; 1516; 2010; 2046; 2087; 2259; 2475; 2670; 3019; 3077; 3428; 3688; 3694; 3941; 3957; 4480; 4932; 4989; 5206; 5308; 5512; 5590; 5885; 6224; 6343; 6485; 6767; 6879; 6940; 7192; 7612; 7798; 8516; 8712; 9854; 10468; 10640; 10829; 10959; 11037; 11339; 11404; 11425; 11522; 11692; 11855; 11882; 11931; 12037; 12097; 12181; 12277	present; future	62
439	cropland percent	1444; 3542; 357; 410; 552; 683; 684; 827; 1322; 1484; 1531; 1891; 2015; 2044; 2313; 2946; 3415; 3452; 4034; 4877; 4882; 5334; 5422; 5631; 5644; 5977; 6002; 6079; 6478; 6785; 7185; 7195; 7265; 7605; 7612; 7733; 7889; 8974; 9171; 9374; 10126; 10977; 11063; 12332	past; present; future	44

Out of curiosity, we also want to know how many human predictors have been used by at least more than one paper.

```
# get length
  paste("A total of",dim(preds.list.short2)[1],
        "human predictors are used in at least more than one paper.")
```

```
## [1] "A total of 371 human predictors are used in at least more than one paper."
```

We also want to know how many human predictors have been used in only one paper.

```
# get length
  paste(nrow(preds.list.shorter[preds.list.shorter$count==1,]),
        "out of",
        nrow(preds.list.shorter),
        "human predictors have only been used in only one paper.")
```

```
## [1] "1936 out of 2307 human predictors have only been used in only one paper."
```

3.5 Sorting predictors by data type

Add a blank column for data type, and then fill in this field by text mining predictor names and assigning accordingly.

```
preds.list.shorter$data_type <- NA
```

3.5.1 Predictors relating to density/count

```r
# make list of search terms
  cat_list <- c("* average$","* mean$","*sum$","*total$","abundance","concentration",
                "count","death","density","frequency","growth","income","individual*",
                "killing","loss","maximum","mean annual", "minimum","number","percent",
                "poisoning","precipitation","proportion","quantity","range",
                "tons per hectare","total annual","total dissolved","volume",
                "tonnage","sum imports","watts","wastewater discharge"
                )

# inspect list
  #preds.list.shorter$predictor[grepl(paste0(cat_list, collapse = "|"),
  #                                 preds.list.shorter$predictor)]

# search and append
  preds.list.shorter$data_type <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                              preds.list.shorter$predictor),
                                        'density/count', preds.list.shorter$data_type)

# inspect (note NA's remaining)
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$data_type))
```

```
## density/count          NA's
##          1120          1187
```

3.5.2 Predictors using indices, ratios or intensities

```r
# make list of search terms
  cat_list <- c("index", "intensity", "boat traffic","fire frequency",
                "* fragmentation$", "* poverty$", "footprint",
                "accessibility", "cohesion", "activeness","contagion",
                "dynamics","heterogeneity","value","* ratio$",
                "impact","* level$", "acidification"," activity ",
                "pressure","capacity","withdrawal"," rate$"," PCA",
                "gross domestic","effort","financial returns",
                "human land transformation","land use change",
                "wealth","pastures change yr1900-2005","annual daily",
                "pesticide application rate kg km2","forest loss 10yr mean",
                "harvest interannual SD","^land cover diversity$",
                "agricultural modification","reservoir capacities ",
                "imperviousness","clumpiness","disturbance geomorphological",
                "risk","diffusion","change","household movement",
                "interspersion","diversity","evenness","richness",
                "transformation","integrity","probability","velocity",
                "contrast","connectivity","productivity","noise"
                )

# search and append
  preds.list.shorter$data_type <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                              preds.list.shorter$predictor),
                                        'index', preds.list.shorter$data_type)
```

```
# inspect (note NA's remaining)
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$data_type))
```

```
## density/count          index          NA's
##          1063           234          1010
```

3.5.3 Predictors referring to size

```
# make list of search terms
  cat_list <- c("size","length","* m$","height","* ha$","hectares",
                "surface area$","area change","* area$","area-weighted mean",
                "area-weighted-mean","* m2$","DBH","width"
                )

# search and append
  preds.list.shorter$data_type <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                               preds.list.shorter$predictor),
                                         'size', preds.list.shorter$data_type)

# inspect (note NA's remaining)
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$data_type))
```

```
## density/count          index          size          NA's
##          1043           233           183           848
```

3.5.4 Predictors that are descriptive

Descriptive refers to e.g. presence/absence, 1/0s, distribution, types, areas, status or state of a feature or category. These are typically categorical data types.

```
# make list of search terms
  cat_list <- c("presence","absence","type*","units","* areas$", "class",
                "* edges","undisturbed","disturbed forest$","sites$",
                "*conventional$","status",
                "^parks$"," reserves","biome","anthropogenic land",
                "arable and farming lands","gyrate$","^vineyards$",
                "mixed","land-use","station","^buildings$"," crops$",
                "human-dominated landscape","^highways$","filter",
                " disturbance$"," disturbance ","saline",
                " crops$","forest management approaches","forest non-forest",
                "human settlements","agricultural areas heterogeneous",
                "agricultural areas intensive","agricultural areas natural",
                "built-up subbasin","arable and farming lands","built-up upstream",
                "plantations","systems","ownership"," groves$","diked",
                "excavated","walls s*","wasteland","^town$","^tracks$","^trails$",
                "^shipwrecks$","advisor*","^roads$","verge"," winter","^railways$",
                "^roads main$","^roads primary$","^roads secondary$",
                "recent burn","^powerlines$","excavated","permanent crops",
```

```
                 "restricted","^mines$","mines historic","artificial surfaces",
                 "residual"," open low","nongrazing","human influence and ",
                 "distribution","registration","^pastures$","^row_crops$",
                 "land use/land cover","land use historic","vegetation$",
                 "Native American land","*passable stream barrier",
                 "hydrocarbons high","hydrocarbons low","developed open space",
                 "historical yr1900","scenic locations","seismic lines","^arable land$",
                 "^oil gas pipeline$","developed areas roads and deciduous woodland",
                 "crops fruit tree","cropland and grassland","cropland and pastures",
                 "crops dry herbaceous present absent","scenic locations","* trails$",
                 "* pipelines$","* cutlines$","* well pads$","^villages$","* rainfed$",
                 "^landscape condition$","* slums$","cut-block features","state name",
                 "* abandoned$", "* abandoned_areas","* woody$","* greenhouses$",
                 "tire storage depots","forest harvested","needs unmet","* mosaic$",
                 "land use previous year","country boundary","* green space$",
                 "agricultural areas 500m radius","agricultural areas 10m radius",
                 "latitude","longitude"
                 )

# search and append
  preds.list.shorter$data_type <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                               preds.list.shorter$predictor),
                                         'descriptive', preds.list.shorter$data_type)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$data_type))
```

```
## density/count   descriptive         index          size           NA's
##           954           463           194           153            543
```

3.5.5 Predictors relating to distance

```
# make list of search terms
  cat_list <- c("*distance*", "* depth", "* proximity", "* extent", "adjacency")

# search and append
  preds.list.shorter$data_type <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                               preds.list.shorter$predictor),
                                         'distance', preds.list.shorter$data_type)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$data_type))
```

```
## density/count   descriptive      distance         index          size           NA's
##           942           437           393           187           153            195
```

3.5.6 Predictors using time

Time refers to e.g. those listing years, or length of time.

```
# make list of search terms
  cat_list <- c("year","period"," time","time since","annual days","date","duration",
                "age class","* age$")

# search and append
  preds.list.shorter$data_type <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                               preds.list.shorter$predictor),
                                         'time', preds.list.shorter$data_type)

# inspect (note NA's remaining)
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$data_type))
```

```
## density/count   descriptive      distance         index          size          time
##           941           433           393           187           153            28
##          NA's
##           172
```

3.5.7 Small manual changes

```
# change to density/count
  cat_list <- c('developed open space percent','artificial surfaces percent',
                'permanent crops percent','gallons')
  preds.list.shorter$data_type <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                               preds.list.shorter$predictor),
                                         'density/count', preds.list.shorter$data_type)

# change to index
  cat_list <- c('*high-low ratio$','burning frequency','logging frequency',
                "diversity")
  preds.list.shorter$data_type <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                               preds.list.shorter$predictor),
                                         'index', preds.list.shorter$data_type)

# inspect
  summary(as.factor(preds.list.shorter$data_type))
```

```
## density/count   descriptive      distance         index          size          time
##           954           413           393           195           153            27
##          NA's
##           172
```

3.6 Table inspection

Show any remaining NAs, and edit above. It was found that all remaining NAs (after visual inspection) qualify as "descriptive", and are reclassified to this data type.

```
# preview remaining NA's (activate as needed)
  #preds.list.shorter[is.na(preds.list.shorter$data_type),]
```

Reclassify remaining NA's to descriptive data type.

```
# preview remaining NA's
  preds.list.shorter$data_type[is.na(preds.list.shorter$data_type)] <- "descriptive"
```

Save and get final summary.

```
# save
  write.csv(preds.list.shorter, paste0(data.dir,"predictor_list_summary_SHORT.csv"),
            row.names = FALSE)

# inspect
  summary(as.factor(preds.list.shorter$data_type))
```

```
## density/count   descriptive      distance         index          size          time
##           954           585           393           195           153            27
```

3.7 Sorting predictors by category

Add a new column for the categories.

```
preds.list.shorter$category <- NA
```

3.7.1 Predictors relating to barriers/access

```
# make list of search terms
  cat_list <- c("accessibility", "barrier", "fenc*", "wall",
                "water flow obstacles","boundary","hedgerows","human access")

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                              preds.list.shorter$predictor),
                                        'barriers/access', preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
## barriers/access            NA's
##              31            2276
```

3.7.2 Predictors relating to transportation

```
# make list of search terms
  cat_list <- c("airport", "railway", "aviation", "traffic","train",
                "shipping","roads","shipwrecks","ports","tracks",
                "highway","boat ramp","canal","intersections",
                "transportation","navigable","linear features",
```

```
                "airstrips","waterway","boat launch","path","transit",
                "watercourses","ship ","ships ","road to scrub","mooring"
                )

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                               preds.list.shorter$predictor),
                                      'transportation', preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
## barriers/access  transportation           NA's
##              30             272           2005
```

3.7.3 Predictors relating to human presence (general)

```
# make list of search terms
  cat_list <- c("human influence","human activity","human areas",
                "human footprint","human populated areas",
                "human population_center","moved in","human use",
                "anthropogenic biome","anthropogenic land",
                "human land use","human land transformation",
                "human features","human-dominated landscape",
                "anthropogenic","anthrome","anthropic",
                "human presence","human-dominated")

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                               preds.list.shorter$predictor),
                                      'human presence', preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
## barriers/access  human presence  transportation            NA's
##              30              47             272            1958
```

3.7.4 Predictors relating to food and agriculture

```
# make list of search terms
  cat_list <- c("aquaculture","fish","aviculture","winery",
                "plantation","agricultural","pasture","viticulture",
                "allotment","arable","cultiva*","cropland","donkey",
                "crops","horticulture","artichokes","farm","grazing",
                "cattle","feeding","ruminant","fields","sheep","goat",
                "yams","harvest","human footprint agricultural","hay",
```

```
                 "livestock","groves","poultry","irrigation",
                 "husbandry","vineyards","wheat","platanus","tilled",
                 "agroforestry","field activities","food source type",
                 "horse","irrigated areas","orchards","pastoral",
                 "permanent crop","pig ","ranchos","rangelands","tractors",
                 "seed","crop ","crop damage","fallow","corn","maize",
                 "fertilizer","irrigated","tillage","tree nursery",
                 "rice paddy","rangeland","permanent cultures",
                 "planted pine","plough","meadow","productive lands without trees")

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                              preds.list.shorter$predictor),
                                       'food/agriculture', preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
##  barriers/access food/agriculture   human presence   transportation             NA's
##               29              863               44              267             1104
```

3.7.5 Predictors relating to pollution

```
# make list of search terms
  cat_list <- c("andosol", "toxic", "poison", "pesticide",
                 "hydrocarbon","chlorophyll","light","wastewater",
                 "night light","salinity","runoff","acidification",
                 "brownfield","pollution","pollutant","concentration",
                 "artificial illumination","contaminated","dumping site",
                 "waste dumping","dump","inorganic","rubble","insecticide",
                 "superfund sites","phosphorus total","dissolved")

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                              preds.list.shorter$predictor),
                                       'pollution', preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
##  barriers/access food/agriculture   human presence        pollution   transportation
##               29              779               44              144              267
##             NA's
##             1044
```

3.7.6 Predictors relating to tourism/recreation

```r
# make list of search terms
  cat_list <- c("bike","garden","trail","ski-resort","campground","golf",
                "parks","hunting","pet shop","hunter","fishing camp","dog",
                "recreation","scenic","game guard","ski","artisanal fishing",
                "tourism","tourist")

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                              preds.list.shorter$predictor),
                                        'recreation/tourism', preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
##    barriers/access   food/agriculture     human presence          pollution
##                 29                775                 44                144
## recreation/tourism     transportation               NA's
##                 72                263                980
```

3.7.7 Predictors relating to energy/raw materials

Not that this excludes wood products, which were classified under disturbance, since it is more related to deforestation, etc.

```r
# make list of search terms
  cat_list <- c("dams", "electric", "hydropower", "energy","damming",
                "hydraulic","utility","well pads","clear cut","^wells ",
                "^wind *","mines","mining","oil gas","oil camp","oil well",
                "seismic","forest cut","dredging and","pipeline"," cutlines",
                "transmission lines","powerlines","velocity","withdrawl",
                "well","reservoir","collection","excavate","extract",
                "dredging","forest processing","dripline","production forest",
                "logged forest","surface fuels type"
                )

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                              preds.list.shorter$predictor),
                                        'energy/raw materials', preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
##      barriers/access energy/raw materials     food/agriculture       human presence
##                   29                  137                  769                   44
##            pollution   recreation/tourism       transportation                 NA's
##                  143                   70                  260                  855
```

3.7.8 Predictors relating to socio-economics

```
# make list of search terms
  cat_list <- c("financial","gross domestic","property size",
                "healthcare","household","human population",
                "poverty","wealth","land value","owner","police",
                "profession","renters ","working","unemployment",
                "residents*65","killings","education","65yrs","^residents",
                "marijuana","opium","community associations","workers",
                "retired people","water withdrawal","travel time",
                "income","inhabitants nearest","citizen","land tenure",
                "land value","state name","basic needs","inhabitant density",
                "cleared vegetation","commune","communit","industry density",
                "postal address forwards","^inhabitants"
                )

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                             preds.list.shorter$predictor),
                                       'socio-economic', preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
##      barriers/access energy/raw materials     food/agriculture       human presence
##                   29                  136                  764                   44
##            pollution   recreation/tourism       socio-economic       transportation
##                  143                   70                  107                  259
##                 NA's
##                  755
```

3.7.9 Predictors relating to disturbance/fragmentation

```
# make list of search terms
  cat_list <- c("fragmentation","artificial areas","artificial surfaces",
                "degraded","deforestation"," burn ","forest burns","disturbed",
                "burnt","direct human pressure","disturbance","forest loss",
                "removal","habitat loss","perturbation","logging","threat",
                "avoidance","marine human impact","naturalness","undisturbed",
                "cut block","cut-block","fire","exotic species","semi-natural",
                "destroyed","bare ground ","clearcut","forest cut","canopy loss",
                "bare land","water risk","extirpation","cutovers and burns",
                "pressure","noise","transformation",
                "meadow exploited","modified habitat","impact","poaching",
                "stream crossings","alteration","road-stream density crossings",
                "human modification")

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                             preds.list.shorter$predictor),
```

```
                                     'disturbance', preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
##      barriers/access          disturbance energy/raw materials     food/agriculture
##                   29                  171                  133                  758
##       human presence            pollution   recreation/tourism       socio-economic
##                   38                  142                   68                  106
##       transportation                 NA's
##                  250                  612
```

3.7.10 Predictors relating to infrastructure

```
# make list of search terms
  cat_list <- c("bridges","buildings","town","ditch","drain",
                "land ownership","bathing","landfill","car wash",
                "infrastr","built-up","coastline type","apartments",
                "developed","manmade","street","settlements",
                "filling distance","easement","gravel","residential area",
                "wasteland","houses","housing","household density",
                "cities","communication towers","impervious",
                "development intensity","construction activities",
                "industrial sites","military","urban areas","slums",
                "urban land","urban rural","villages","lawn",
                "wetlands excavated","dike","urban polygons","universities",
                "human coast type","polder","sparsely populated areas",
                "urban center","inundation","abstraction","tower ",
                "roof sheet","depot","water supply","structures","church",
                "weir","urban","artificial land","artificial water",
                "artificial open","artificial flooding","property size",
                "production of property","commercial units","commercial area",
                "manufacturer","commercial plant nursery","services",
                "construction sites","cottage","factory","house distance",
                "house density","industrial area","development","rural areas",
                "rural land","residence distance","residences count",
                "public facility","paved area","navy exercise areas",
                "settling lagoons","industrial facility")

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                             preds.list.shorter$predictor),
                                     'infrastructure', preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
##      barriers/access          disturbance energy/raw materials     food/agriculture
##                   29                  117                  129                  750
```

```
##       human presence       infrastructure            pollution   recreation/tourism
##                   36                  624                   55                   57
##       socio-economic       transportation                 NA's
##                   97                  228                  185
```

3.7.11 Predictors relating to management/interventions

```
# make list of search terms
  cat_list <- c("management","managed","protected areas",
                "reintroduction ","wilderness","first record",
                "forest distance","forest presence","habitat filter",
                "Native American","nature reserves","non-hunting","introduced site",
                "urban forest","regulated areas","introduction locus",
                "advisories","water ecological status","artificial regeneration",
                "artificial reef","baiting treatment","research camp",
                "conservation","regulation","stocking","reserve","protection",
                "improved grassland","unprotected","tribal land",
                "introduction","silvicultur","site preparation","scientific",
                "reforest","ranger station","protected","park security",
                "marine park","nest box","intensive grasslands",
                "snare hotspots"
                )

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                          preds.list.shorter$predictor),
                                      'management/interventions',
                                      preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
##          barriers/access              disturbance     energy/raw materials
##                       29                      116                      127
##         food/agriculture           human presence           infrastructure
##                      738                       36                      621
## management/interventions                pollution       recreation/tourism
##                      107                       53                       53
##           socio-economic           transportation                     NA's
##                       96                      227                      104
```

3.7.12 Predictors that are ambiguous

```
# make list of search terms
  cat_list <- c("land cover","land-use other","^land use ","diversity",
                "herbaceous areas","forested","landscape PCA","intactness",
                "open areas","wetland types","landscape condition","integrity",
                "forest yr2050","forest distance","time since land abandonment",
                "abandoned areas with vegetation","abandoned areas percent",
```

```
                 "wildness","forests natural and commercial","burn natural or unnatural",
                 "artificial or natural water","forest non-forest","historic vegetation",
                 "land condition index","unexploited areas","forest stands 15-30yrs",
                 "secondary land","remnant native habitat distance","remnant vegetation",
                 "pond isolation","openness"
                 )

# search and append
  preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                              preds.list.shorter$predictor),
                                        'ambiguous',
                                        preds.list.shorter$category)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
##                ambiguous           barriers/access               disturbance
##                      117                        29                       115
##     energy/raw materials          food/agriculture            human presence
##                      127                       734                        36
##           infrastructure management/interventions                 pollution
##                      618                       102                        53
##       recreation/tourism             socio-economic            transportation
##                       53                        96                       227
```

Show any remaining NAs.

```
# show NAs
  preds.list.shorter[is.na(preds.list.shorter$category),]
```

```
## [1] predictor  papers     timeframes count      data_type  category
## <0 rows> (or 0-length row.names)
```

3.7.13 Small manual changes

```
# change to disturbance
  cat_list <- c("deforested","pressure","direct human pressure")

  # search and append
    preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                                preds.list.shorter$predictor),
                                         'disturbance',
                                         preds.list.shorter$category)

# change to pollution
  cat_list <- c("night light development index","herbicide pressure")

  # search and append
    preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
```

```
                                               preds.list.shorter$predictor),
                                      'pollution',
                                      preds.list.shorter$category)

# change to transportation
  cat_list <- c("aviation structure distance")

  # search and append
    preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                               preds.list.shorter$predictor),
                                      'transportation',
                                      preds.list.shorter$category)

# change to management/interventions
  cat_list <- c("reforest")

  # search and append
    preds.list.shorter$category <- ifelse(grepl(paste0(cat_list, collapse = "|"),
                                               preds.list.shorter$predictor),
                                      'management/interventions',
                                      preds.list.shorter$category)
```

Show any remaining NAs.

```
# show NAs
  preds.list.shorter[is.na(preds.list.shorter$category),]
```

```
## [1] predictor  papers     timeframes count      data_type  category
## <0 rows> (or 0-length row.names)
```

Save by overwriting previous summary table.

```
# save
  write.csv(preds.list.shorter,
            paste0(data.dir,"predictor_list_summary_SHORT.csv"),
            row.names = FALSE)

# inspect
  options(width = 85) #ensure width
  summary(as.factor(preds.list.shorter$category))
```

```
##                ambiguous          barriers/access              disturbance
##                      115                       29                      115
##      energy/raw materials        food/agriculture           human presence
##                      127                      734                       36
##           infrastructure management/interventions                pollution
##                      617                      103                       55
##        recreation/tourism           socio-economic           transportation
##                       53                       96                      227
```

4 Nested pie chart of predictor use

Next, we summarize data and create pie charts with the following three layers:

- Inner pie: data types
- 1st outer: categories
- 2nd outer: sum of unique articles per predictor

4.0.1 Make summaries and labels for each pie layer

Inner pie: data type

```
# Make a summary of each data type for the inner pie
  type_totals = ddply(preds.list.shorter,
                    .(data_type), summarize, count=length(data_type))

# show here
  kableExtra::kbl(type_totals,booktabs=T, longtable=T) %>%
    kable_styling(latex_options = c("striped","repeat_header"))
```

data_type	count
density/count	954
descriptive	585
distance	393
index	195
size	153
time	27

```
# change labels for proper fitting
  type_totals$labs <- type_totals$data_type
  type_totals$labs <- as.factor(type_totals$labs)
  levels(type_totals$labs)[1] <- "density/count      "
  levels(type_totals$labs)[2] <- "descriptive"
  levels(type_totals$labs)[3] <- "\n       distance"
  levels(type_totals$labs)[4] <- "\nindex"
  levels(type_totals$labs)[5] <- "      size"
  levels(type_totals$labs)[6] <- "      time"

# change labels for proper fitting
  type_totals$numlabs <- type_totals$count
  #type_totals$numlabs <- as.factor(type_totals$numlabs)
  type_totals$numlabs[1] <- paste0("                ",type_totals$numlabs[1])
  type_totals$numlabs[2] <- paste0("              ",type_totals$numlabs[2])
  type_totals$numlabs[3] <- paste0("\n\n",type_totals$numlabs[3])
  type_totals$numlabs[4] <- paste0("\n\n\n",type_totals$numlabs[4])
  type_totals$numlabs[5] <- paste0("\n\n",type_totals$numlabs[5])
  type_totals$numlabs[6] <- paste0("\n\n",type_totals$numlabs[6])

# show here
  options(width = 85) #ensure width
  type_totals
```

```
##       data_type count                 labs            numlabs
## 1 density/count   954 density/count                      954
## 2   descriptive   585         descriptive                585
## 3      distance   393   \n       distance             \n\n393
## 4         index   195             \nindex           \n\n\n195
## 5          size   153                size             \n\n153
## 6          time    27                time              \n\n27
```

1st outer pie: predictor categories

```
# Make a summary of each category for 1st outer pie
  cat_totals = ddply(preds.list.shorter,
                     .(data_type,category), summarize, count=length(category))

# show here
  kableExtra::kbl(cat_totals,booktabs=T, longtable=T) %>%
    kable_styling(latex_options = c("striped","repeat_header"))
```

data_type	category	count
density/count	ambiguous	13
density/count	barriers/access	6
density/count	disturbance	33
density/count	energy/raw materials	52
density/count	food/agriculture	361
density/count	human presence	7
density/count	infrastructure	276
density/count	management/interventions	33
density/count	pollution	24
density/count	recreation/tourism	11
density/count	socio-economic	53
density/count	transportation	85
descriptive	ambiguous	69
descriptive	barriers/access	9
descriptive	disturbance	31
descriptive	energy/raw materials	34
descriptive	food/agriculture	190
descriptive	human presence	13
descriptive	infrastructure	109
descriptive	management/interventions	39
descriptive	pollution	18
descriptive	recreation/tourism	12
descriptive	socio-economic	10
descriptive	transportation	51
distance	ambiguous	8
distance	barriers/access	8
distance	disturbance	10
distance	energy/raw materials	29
distance	food/agriculture	82
distance	human presence	10
distance	infrastructure	127

(continued)

data_type	category	count
distance	management/interventions	21
distance	pollution	4
distance	recreation/tourism	25
distance	socio-economic	10
distance	transportation	59
index	ambiguous	21
index	barriers/access	2
index	disturbance	31
index	energy/raw materials	2
index	food/agriculture	28
index	human presence	6
index	infrastructure	62
index	management/interventions	8
index	pollution	9
index	recreation/tourism	3
index	socio-economic	16
index	transportation	7
size	ambiguous	2
size	barriers/access	4
size	disturbance	6
size	energy/raw materials	9
size	food/agriculture	67
size	infrastructure	31
size	management/interventions	1
size	recreation/tourism	2
size	socio-economic	6
size	transportation	25
time	ambiguous	2
time	disturbance	4
time	energy/raw materials	1
time	food/agriculture	6
time	infrastructure	12
time	management/interventions	1
time	socio-economic	1

```
# change labels for count
# (we are removing labels for any predictors <=9 for sake of space)
  cat_totals$labs <- cat_totals$count
  cat_totals$labs[cat_totals$labs<=9] <- '' # make blank

# change labels for legend
  cat_totals$legend <- cat_totals$category
  cat_totals$legend <- as.factor(cat_totals$legend)

# Sort frequency and assign colors
  freqs <- unique(preds.list.shorter$count)[order(unique(preds.list.shorter$count))]
  require(classInt)
  col_ints <- classIntervals(preds.list.shorter$count, 10, style = "jenks")
```

```r
# add color (colorblind-friendly)
  cols <- colorRampPalette(c('#C2E3D2','#B5DDD8','#A8D8DC','#9BD2E1','#8DCBE4',
                             '#7BBCE7','#7EB2E4','#88A5DD','#9398D2','#9D7DB2'))
  cols <- cols(10)
  #plot(rep(1,10),col=cols,pch=19,cex=3) # activate to inspect color palette

# append matching colors for each interval
  new_cols <- findColours(col_ints, cols)
  preds.list.shorter$cols <- new_cols
```

2nd outer pie: sum of unique articles per predictor

```r
# Sort by predictor, category and number of unique articles for 3rd outer pie
  pred_totals = ddply(preds.list.shorter,
                      .(data_type,category,predictor), summarize,
                      count=count, hex_col=cols, ones=1)

# sort in order of frequency
  pred_totals <- arrange(pred_totals,data_type,category,desc(count))

# preview here
  head(pred_totals)
```

```
##       data_type  category                           predictor count hex_col ones
## 1 density/count ambiguous             abandoned areas percent     1 #C2E3D2    1
## 2 density/count ambiguous herbaceous areas density 50m radius     1 #C2E3D2    1
## 3 density/count ambiguous                  intactness percent     1 #C2E3D2    1
## 4 density/count ambiguous                land use low percent     1 #C2E3D2    1
## 5 density/count ambiguous   non-forest secondary land percent     1 #C2E3D2    1
## 6 density/count ambiguous  open areas percent historic yr1830     1 #C2E3D2    1
```

4.0.2 Pie chart

Using a custom pie function, acquired from a stackoverflow help page (https://stackoverflow.com/questions/25880110/r-put-labels-inside-pie-chart):

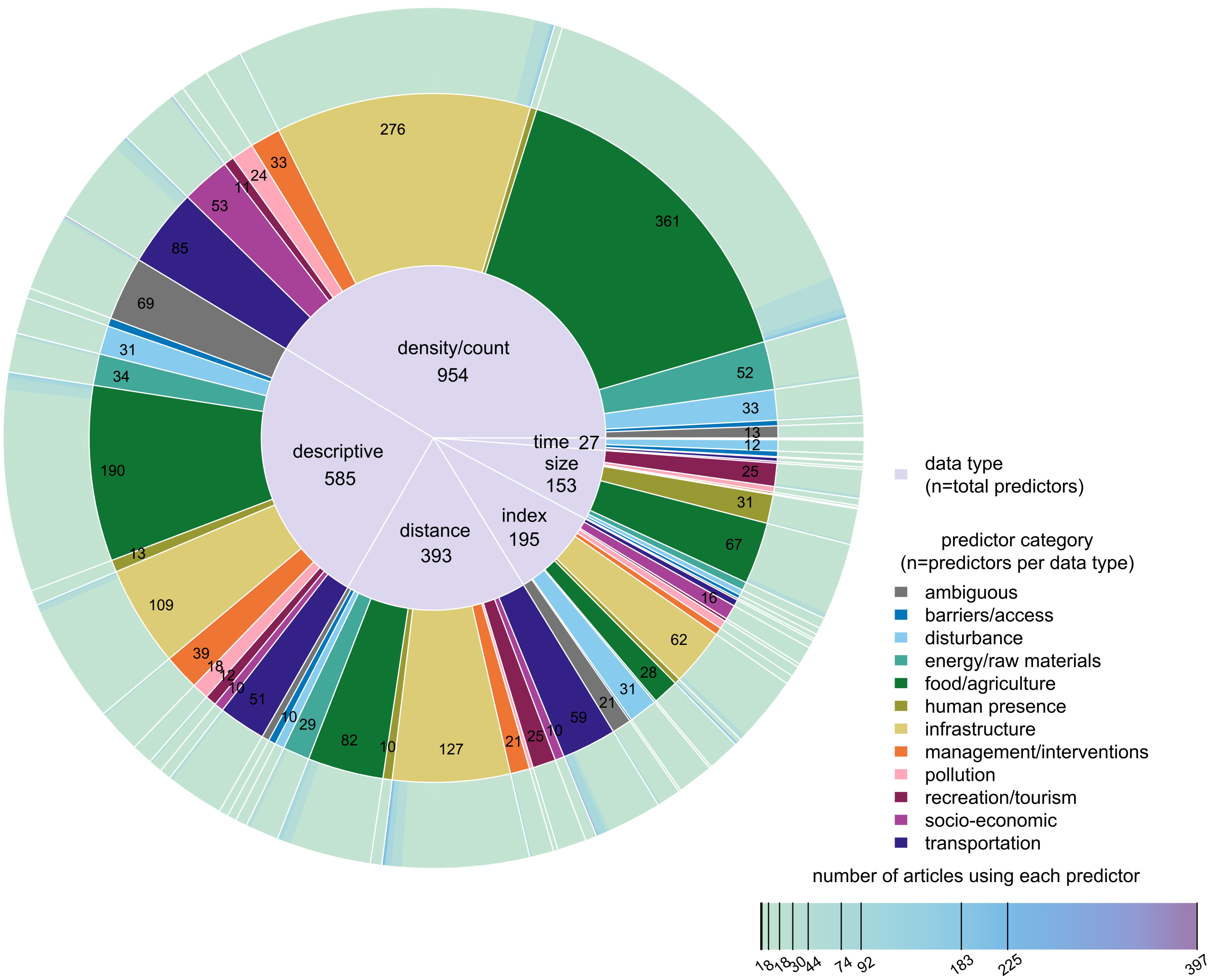
Then we use these functions to make a pie.

```r
# Make a 3-level pie
  #' x      numeric vector for each slice
  #' group  vector identifying the group for each slice
  #' labels vector of labels for individual slices
  #' col    colors for each group
  #' radius radius for inner and outer pie (usually in [0,1])
  # png(paste0(image.dir,"predictor_pie_RAW.png"),
  #     height=10,width=14,units='in',res=600)
  svg(paste0(image.dir,"predictor_pie_RAW.svg"),
      height=10,width=14)
  plot.new()
    { #white outlines for pies
      par(new = TRUE)
      pie(pred_totals$ones, border = FALSE, radius = 1,
```

```
        col = pred_totals$hex_col, #edges = 10,
        labels = NA)

    par(new = TRUE)
    newpie2(cat_totals$count, border = "white", radius = 1,
        col = NA,
        labels = NA)

    par(new = TRUE)
    newpie2(cat_totals$count, border = "white", radius = .8,
        #colorblind-friendly
        col=c('#777777','#0077BB','#88CCEE','#44AA99','#117733','#999933',
              '#DDCC77','#EE7733','#FFAABB','#882255','#AA4499','#332288'),
        labels = cat_totals$labs, cex=0.8)

    # fills of each pie
    par(new = TRUE)
    newpie2(cat_totals$count,border = NA, radius = .8,
        col = NA,
        labels = cat_totals$labs, cex=0.8)

    par(new = TRUE)
    newpie(type_totals$count, border = "white", radius = .4,
        col='#DDD8EF',
        labels = type_totals$labs)

    par(new = TRUE)
    newpie(type_totals$count, border = NA, radius = .4,
        col=NA,
        labels = type_totals$numlabs)

    legend(x=1.03,y=-0.22,legend=unique(cat_totals$legend),
           bty="n",border='white',
           fill=c('#777777','#0077BB','#88CCEE','#44AA99','#117733','#999933',
              '#DDCC77','#EE7733','#FFAABB','#882255','#AA4499','#332288'),
           title='predictor category\n(n=predictors per data type)')

    legend(x=1.03,y=0,legend='data type\n(n=total predictors)',
           fill='#DDD8EF', border = 'white', bty="n")

    gradientLegend(valRange=col_ints$brks,length = .3,
                   n.seg = col_ints$brks, side=1, color=cols,
                   title='number of articles using predictor',
                   border.col = NA, tick.col = 'black')

    text(x=1.26,y=-1.02,
         labels = 'number of articles using each predictor')
  }
  dev.off()
```

```
## pdf
##   2
```

Some manual edits were done to move labels for better visibility using Inkscape. The final image is uploaded here:



5 Comparing predictor use by context (study focus and taxa)

It's possible that the use of these different predictors/categories is context-specific. We can assess that by appending the lists from the alluvial plots with this updated predictor list and making bar plots.

5.1 Combining dataframes

First, we combine the taxa/domain/focus tables and preview the new dataframe.

```
# first, rename some columns
  names(domtaxfoc.df)[names(domtaxfoc.df) == 'taxon_group'] <- 'taxa'
  names(domtaxfoc.df)[names(domtaxfoc.df) == 'count'] <- 'count_studies'

# preview
  head(domtaxfoc.df)
```

```
##       domain       taxa                  study_focus count_studies
```

```
## 1 freshwater amphibians                conservation             4
## 2 freshwater amphibians disturbance/habitat change             3
## 3 freshwater amphibians                 exploratory             3
## 4 freshwater amphibians                   invasions             4
## 5 freshwater amphibians reintroduction/restoration             1
## 6 freshwater      birds                conservation             3
##                  papers count_papers
## 1  66; 2338; 7489; 10468            4
## 2      6957; 9483; 11546            3
## 3       863; 5512; 10170            3
## 4 3452; 3617; 6978; 8147            4
## 5                   9360            1
## 6      4581; 5829; 11087            3
```

Show some of the predictor list.

```
# preview
  options(width = 85) # ensure width
  head(preds.list.shorter)
```

```
##                                   predictor papers            timeframes count
## 1 abandoned agricultural areas new distance   8233               present     1
## 2 abandoned agricultural areas old distance   8233               present     1
## 3                    abandoned areas percent  11080 past; present; future     1
## 4            abandoned areas with vegetation   9901               present     1
## 5                 abandoned cropland percent   2946               present     1
## 6                 abandoned pastures percent   2946               present     1
##        data_type         category    cols
## 1      distance food/agriculture #C2E3D2
## 2      distance food/agriculture #C2E3D2
## 3 density/count        ambiguous #C2E3D2
## 4   descriptive        ambiguous #C2E3D2
## 5 density/count food/agriculture #C2E3D2
## 6 density/count food/agriculture #C2E3D2
```

Lengthen the predictor and taxa lists, where there's a new row per semi-colon-separated paper.

```
# lengthen predictor list
  # mutate data, with repeated rows of papers and predictors
    preds.list.long <- preds.list.shorter %>%
                        rownames_to_column() %>%
                        mutate(uid = strsplit(papers, "; ")) %>%
                        unnest %>%
                        group_by(uid)

  # double-check count of unique papers (should match PRISMA)
    paste("number of unique papers:",
          summarise(preds.list.long,count = length(unique(uid))))


# lengthen taxa/domain/focus list
  # mutate data, with repeated rows of papers per taxa and domain
    dotf.list.long <- domtaxfoc.df %>%
```

```
                        rownames_to_column() %>%
                        mutate(uid = strsplit(papers, "; ")) %>%
                        unnest %>%
                        group_by(uid)

  # double-check count of unique papers (should match PRISMA)
    paste("number of unique papers:",
          summarise(dotf.list.long,count = length(unique(uid))))
```

```
## [1] "number of unique papers: 1429"
## [1] "number of unique papers: 1429"
```

Next, join the two dataframes by paper ID (UID). This list will be longer than the original `preds.list.long` since there are multiples of taxa and domains per paper in `domtaxfoc.df`.

```
# left join
  prdotf.list.long <- left_join(preds.list.long,dotf.list.long, by='uid')
```

We **save this list** for use later in the synthesis of edited data fields (**see Part IV and V).

```
# save as a CSV
  write.csv(prdotf.list.long ,paste0(data.dir,"predictor_domain_taxa_focus_long.csv"),
            row.names = FALSE)
```

We then shorten this list to make it unique to only the predictors per paper UID, for us to target the focus of each study. The length is expected to be the same as the first `predictor.list.long` that we made. Then, we will make a second list, where it will be unique to the predictors per taxa per paper ID. This table will be as long as there are taxa.

```
# predictor list for study focus
  foc.preds.list <- ddply(prdotf.list.long,.(study_focus,category,uid,predictor),
                          summarize,
                          count_studies=length(predictor))

# predictor list for focus
  tax.preds.list <- ddply(prdotf.list.long,.(taxa,category,uid,predictor),
                          summarize,
                          count_studies=length(predictor))
```

We summarize these tables again, getting a count of predictors per category per study focus, and count for predictors per category per taxa. For the former, the count should sum to the total number of articles using all the predictors (the sum of the outer pie from above). For the latter, the sum would be much larger, and should roughly equal the number of taxa studied per article, multiplied by the sum of the outer pie (the exact number may differ by domain). We will also calculate the relative percents.

```
# summaries for study focus
  # get counts
    foc.preds.list2 <- ddply(foc.preds.list,.(study_focus,category),
                             summarize,
                             count=length(predictor))
  # get total, join, and calculate percent
```

```
    foc.preds.tot <-  ddply(foc.preds.list2,.(study_focus),
                            summarize,
                            total=sum(count))
    foc.preds.list2 <- left_join(foc.preds.list2, foc.preds.tot, by='study_focus')
    foc.preds.list2$perc <- foc.preds.list2$count/foc.preds.list2$total

# save table
  write.csv(foc.preds.list2,
            paste0(data.dir,"focus_predictor_percent_list.csv"),
            row.names = FALSE)

# display here
  kableExtra::kbl(foc.preds.list2,booktabs=T, longtable=T) %>%
    kable_styling(latex_options = c("striped","repeat_header")) %>%
    column_spec(1, width="10em") %>%
    column_spec(2, width="10em") %>%
    column_spec(3, width="5em") %>%
    column_spec(4, width="5em") %>%
    column_spec(5, width="5em")
```

study_focus	category	count	total	perc
conflict/collisions	ambiguous	24	208	0.1153846
conflict/collisions	barriers/access	1	208	0.0048077
conflict/collisions	disturbance	4	208	0.0192308
conflict/collisions	energy/raw materials	4	208	0.0192308
conflict/collisions	food/agriculture	53	208	0.2548077
conflict/collisions	human presence	3	208	0.0144231
conflict/collisions	infrastructure	37	208	0.1778846
conflict/collisions	management/interventions	9	208	0.0432692
conflict/collisions	pollution	4	208	0.0192308
conflict/collisions	recreation/tourism	5	208	0.0240385
conflict/collisions	socio-economic	18	208	0.0865385
conflict/collisions	transportation	46	208	0.2211538
conservation	ambiguous	149	1309	0.1138273
conservation	barriers/access	6	1309	0.0045837
conservation	disturbance	38	1309	0.0290298
conservation	energy/raw materials	56	1309	0.0427807
conservation	food/agriculture	322	1309	0.2459893
conservation	human presence	38	1309	0.0290298
conservation	infrastructure	334	1309	0.2551566
conservation	management/interventions	40	1309	0.0305577
conservation	pollution	26	1309	0.0198625
conservation	recreation/tourism	17	1309	0.0129870
conservation	socio-economic	61	1309	0.0466005
conservation	transportation	222	1309	0.1695951
disturbance/habitat change	ambiguous	89	733	0.1214188
disturbance/habitat change	barriers/access	5	733	0.0068213

(continued)

study_focus	category	count	total	perc
disturbance/habitat change	disturbance	36	733	0.0491132
disturbance/habitat change	energy/raw materials	33	733	0.0450205
disturbance/habitat change	food/agriculture	231	733	0.3151432
disturbance/habitat change	human presence	15	733	0.0204638
disturbance/habitat change	infrastructure	152	733	0.2073670
disturbance/habitat change	management/interventions	24	733	0.0327422
disturbance/habitat change	pollution	19	733	0.0259209
disturbance/habitat change	recreation/tourism	9	733	0.0122783
disturbance/habitat change	socio-economic	27	733	0.0368349
disturbance/habitat change	transportation	93	733	0.1268759
exploratory	ambiguous	132	1207	0.1093621
exploratory	barriers/access	15	1207	0.0124275
exploratory	disturbance	32	1207	0.0265120
exploratory	energy/raw materials	32	1207	0.0265120
exploratory	food/agriculture	418	1207	0.3463132
exploratory	human presence	26	1207	0.0215410
exploratory	infrastructure	313	1207	0.2593206
exploratory	management/interventions	29	1207	0.0240265
exploratory	pollution	12	1207	0.0099420
exploratory	recreation/tourism	12	1207	0.0099420
exploratory	socio-economic	44	1207	0.0364540
exploratory	transportation	142	1207	0.1176471
food/economics	ambiguous	24	238	0.1008403
food/economics	barriers/access	1	238	0.0042017
food/economics	disturbance	1	238	0.0042017
food/economics	energy/raw materials	9	238	0.0378151
food/economics	food/agriculture	92	238	0.3865546
food/economics	human presence	9	238	0.0378151
food/economics	infrastructure	38	238	0.1596639
food/economics	management/interventions	5	238	0.0210084
food/economics	pollution	4	238	0.0168067
food/economics	recreation/tourism	5	238	0.0210084
food/economics	socio-economic	21	238	0.0882353
food/economics	transportation	29	238	0.1218487
human health/safety	ambiguous	38	217	0.1751152
human health/safety	barriers/access	2	217	0.0092166
human health/safety	disturbance	4	217	0.0184332
human health/safety	energy/raw materials	3	217	0.0138249

(continued)

study_focus	category	count	total	perc
human health/safety	food/agriculture	53	217	0.2442396
human health/safety	human presence	9	217	0.0414747
human health/safety	infrastructure	60	217	0.2764977
human health/safety	management/interventions	4	217	0.0184332
human health/safety	pollution	4	217	0.0184332
human health/safety	socio-economic	27	217	0.1244240
human health/safety	transportation	13	217	0.0599078
invasions	ambiguous	66	663	0.0995475
invasions	barriers/access	5	663	0.0075415
invasions	disturbance	16	663	0.0241327
invasions	energy/raw materials	13	663	0.0196078
invasions	food/agriculture	109	663	0.1644042
invasions	human presence	48	663	0.0723982
invasions	infrastructure	150	663	0.2262443
invasions	management/interventio	14	663	0.0211161
invasions	pollution	6	663	0.0090498
invasions	recreation/tourism	22	663	0.0331825
invasions	socio-economic	67	663	0.1010558
invasions	transportation	147	663	0.2217195
reintroduction/restoration	ambiguous	38	371	0.1024259
reintroduction/restoratic	barriers/access	3	371	0.0080863
reintroduction/restoration	disturbance	11	371	0.0296496
reintroduction/restoratic	energy/raw materials	12	371	0.0323450
reintroduction/restoration	food/agriculture	94	371	0.2533693
reintroduction/restoratic	human presence	7	371	0.0188679
reintroduction/restoration	infrastructure	80	371	0.2156334
reintroduction/restoratic	management/interventio	20	371	0.0539084
reintroduction/restoration	pollution	5	371	0.0134771
reintroduction/restoratic	recreation/tourism	7	371	0.0188679
reintroduction/restoration	socio-economic	24	371	0.0646900
reintroduction/restoratic	transportation	70	371	0.1886792

```
# summaries for taxa
  # get counts
    tax.preds.list2 <- ddply(tax.preds.list,.(taxa,category),
                             summarize,
                             count=length(predictor))
  # get totals, join, and calculate percent
    tax.preds.tot <-  ddply(tax.preds.list2,.(taxa),
                             summarize,
                             total=sum(count))
    tax.preds.list2 <- left_join(tax.preds.list2, tax.preds.tot, by='taxa')
    tax.preds.list2$perc <- tax.preds.list2$count/tax.preds.list2$total

# save table
  write.csv(tax.preds.list2,
           paste0(data.dir,"taxa_predictor_percent_list.csv"),
           row.names = FALSE)
```

```r
# display here
  kableExtra::kbl(tax.preds.list2,booktabs=T, longtable=T) %>%
    kable_styling(latex_options = c("striped","repeat_header")) %>%
    column_spec(1, width="10em") %>%
    column_spec(2, width="10em") %>%
    column_spec(3, width="5em") %>%
    column_spec(4, width="5em") %>%
    column_spec(5, width="5em")
```

taxa	category	count	total	perc
amphibians	ambiguous	31	213	0.1455399
amphibians	barriers/access	1	213	0.0046948
amphibians	disturbance	3	213	0.0140845
amphibians	energy/raw materials	5	213	0.0234742
amphibians	food/agriculture	65	213	0.3051643
amphibians	human presence	8	213	0.0375587
amphibians	infrastructure	58	213	0.2723005
amphibians	management/interventions	5	213	0.0234742
amphibians	recreation/tourism	1	213	0.0046948
amphibians	socio-economic	9	213	0.0422535
amphibians	transportation	27	213	0.1267606
birds	ambiguous	140	1320	0.1060606
birds	barriers/access	13	1320	0.0098485
birds	disturbance	38	1320	0.0287879
birds	energy/raw materials	44	1320	0.0333333
birds	food/agriculture	466	1320	0.3530303
birds	human presence	23	1320	0.0174242
birds	infrastructure	285	1320	0.2159091
birds	management/interventio	33	1320	0.0250000
birds	pollution	16	1320	0.0121212
birds	recreation/tourism	15	1320	0.0113636
birds	socio-economic	56	1320	0.0424242
birds	transportation	191	1320	0.1446970
fish	ambiguous	19	318	0.0597484
fish	barriers/access	4	318	0.0125786
fish	disturbance	10	318	0.0314465
fish	energy/raw materials	49	318	0.1540881
fish	food/agriculture	82	318	0.2578616
fish	human presence	4	318	0.0125786
fish	infrastructure	83	318	0.2610063
fish	management/interventio	9	318	0.0283019
fish	pollution	8	318	0.0251572
fish	recreation/tourism	1	318	0.0031447
fish	socio-economic	19	318	0.0597484
fish	transportation	30	318	0.0943396
herbaceous plants	ambiguous	76	512	0.1484375
herbaceous plants	barriers/access	3	512	0.0058594
herbaceous plants	disturbance	12	512	0.0234375
herbaceous plants	energy/raw materials	4	512	0.0078125

(continued)

taxa	category	count	total	perc
herbaceous plants	food/agriculture	96	512	0.1875000
herbaceous plants	human presence	25	512	0.0488281
herbaceous plants	infrastructure	153	512	0.2988281
herbaceous plants	management/interventio	10	512	0.0195312
herbaceous plants	pollution	9	512	0.0175781
herbaceous plants	recreation/tourism	10	512	0.0195312
herbaceous plants	socio-economic	31	512	0.0605469
herbaceous plants	transportation	83	512	0.1621094
invertebrates	ambiguous	95	808	0.1175743
invertebrates	barriers/access	4	808	0.0049505
invertebrates	disturbance	18	808	0.0222772
invertebrates	energy/raw materials	16	808	0.0198020
invertebrates	food/agriculture	212	808	0.2623762
invertebrates	human presence	21	808	0.0259901
invertebrates	infrastructure	259	808	0.3205446
invertebrates	management/interventio	22	808	0.0272277
invertebrates	pollution	31	808	0.0383663
invertebrates	recreation/tourism	8	808	0.0099010
invertebrates	socio-economic	51	808	0.0631188
invertebrates	transportation	71	808	0.0878713
mammals	ambiguous	185	1772	0.1044018
mammals	barriers/access	16	1772	0.0090293
mammals	disturbance	59	1772	0.0332957
mammals	energy/raw materials	42	1772	0.0237020
mammals	food/agriculture	439	1772	0.2477427
mammals	human presence	60	1772	0.0338600
mammals	infrastructure	369	1772	0.2082393
mammals	management/interventio	62	1772	0.0349887
mammals	pollution	20	1772	0.0112867
mammals	recreation/tourism	36	1772	0.0203160
mammals	socio-economic	123	1772	0.0694131
mammals	transportation	361	1772	0.2037246
microorganisms	ambiguous	16	95	0.1684211
microorganisms	barriers/access	1	95	0.0105263
microorganisms	disturbance	11	95	0.1157895
microorganisms	food/agriculture	21	95	0.2210526
microorganisms	human presence	6	95	0.0631579
microorganisms	infrastructure	8	95	0.0842105
microorganisms	management/interventions	1	95	0.0105263
microorganisms	pollution	6	95	0.0631579
microorganisms	recreation/tourism	2	95	0.0210526
microorganisms	socio-economic	6	95	0.0631579
microorganisms	transportation	17	95	0.1789474
reptiles	ambiguous	29	225	0.1288889
reptiles	barriers/access	3	225	0.0133333
reptiles	disturbance	4	225	0.0177778
reptiles	energy/raw materials	6	225	0.0266667

(continued)

taxa	category	count	total	perc
reptiles	food/agriculture	58	225	0.2577778
reptiles	human presence	9	225	0.0400000
reptiles	infrastructure	50	225	0.2222222
reptiles	management/interventions	6	225	0.0266667
reptiles	pollution	4	225	0.0177778
reptiles	recreation/tourism	5	225	0.0222222
reptiles	socio-economic	11	225	0.0488889
reptiles	transportation	40	225	0.1777778
trees/shrubs	ambiguous	27	183	0.1475410
trees/shrubs	barriers/access	1	183	0.0054645
trees/shrubs	disturbance	4	183	0.0218579
trees/shrubs	energy/raw materials	3	183	0.0163934
trees/shrubs	food/agriculture	30	183	0.1639344
trees/shrubs	human presence	17	183	0.0928962
trees/shrubs	infrastructure	38	183	0.2076503
trees/shrubs	management/interventions	4	183	0.0218579
trees/shrubs	pollution	2	183	0.0109290
trees/shrubs	recreation/tourism	2	183	0.0109290
trees/shrubs	socio-economic	30	183	0.1639344
trees/shrubs	transportation	25	183	0.1366120

5.2 Visualizing predictor use against study context

5.2.1 Predictor by study focus

```r
# study focus/category plot
  foc_plt <- ggplot(foc.preds.list) +
               aes(x=study_focus, fill=category) +
               geom_bar(position = 'fill', width = .95) +
               ylim(0,1.1) +
               xlab('') + ylab('') +
               scale_y_continuous(breaks = seq(0,1,.1)) +
               scale_x_discrete(labels=c('conflict/\ncollisions','conservation',
                                          'disturbance/\nhabitat change','exploratory',
                                          'food/\neconomics','human\nhealth/safety',
                                          'invasions','reintroduction/\nrestoration')) +
               scale_fill_manual("",
                                 values = c('#777777','#0077BB','#88CCEE','#44AA99',
                                            '#117733','#999933','#DDCC77','#EE7733',
                                            '#FFAABB','#882255','#AA4499','#332288')) +
               theme_classic() +
               theme(legend.position = 'none',
                     axis.text.x = element_text(size=rel(1.09))) +
               geom_text(aes(study_focus, 1.03,
                             label = total, fill=NULL), data = foc.preds.tot) +
               labs(tag=' sum of predictors across articles:') +
               theme(plot.tag.position = c(.2,.94),plot.tag = element_text(size=10)) +
```

```
                ggtitle('Study focus')
```

5.2.2 Predictor use by taxa

```
# study taxa/category plot
  tax_plt <- ggplot(tax.preds.list) +
                aes(x=taxa, fill=category) +
                geom_bar(position = 'fill', width = .95) +
                ylim(0,1.1) +
                xlab('') + ylab('') +
                scale_y_continuous(breaks = seq(0,1,.1)) +
                scale_x_discrete(labels=c("amphibians","birds","fish","herbaceous\nplants",
                                          "invertebrates","mammals","micro-\norganisms",
                                          "reptiles","trees/shrubs")) +
                scale_fill_manual("predictor category",
                                  values = c('#777777','#0077BB','#88CCEE','#44AA99',
                                             '#117733','#999933','#DDCC77','#EE7733',
                                             '#FFAABB','#882255','#AA4499','#332288')) +
                theme_classic() +
                theme(legend.position = 'none',
                      axis.text.x = element_text(size=rel(1.09))) +
                geom_text(aes(taxa, 1.03,
                              label = total, fill=NULL), data = tax.preds.tot) +
                labs(tag=' sum of predictors across articles:') +
                theme(plot.tag.position = c(.2,.94),plot.tag = element_text(size=10)) +
                ggtitle('Taxa')
```

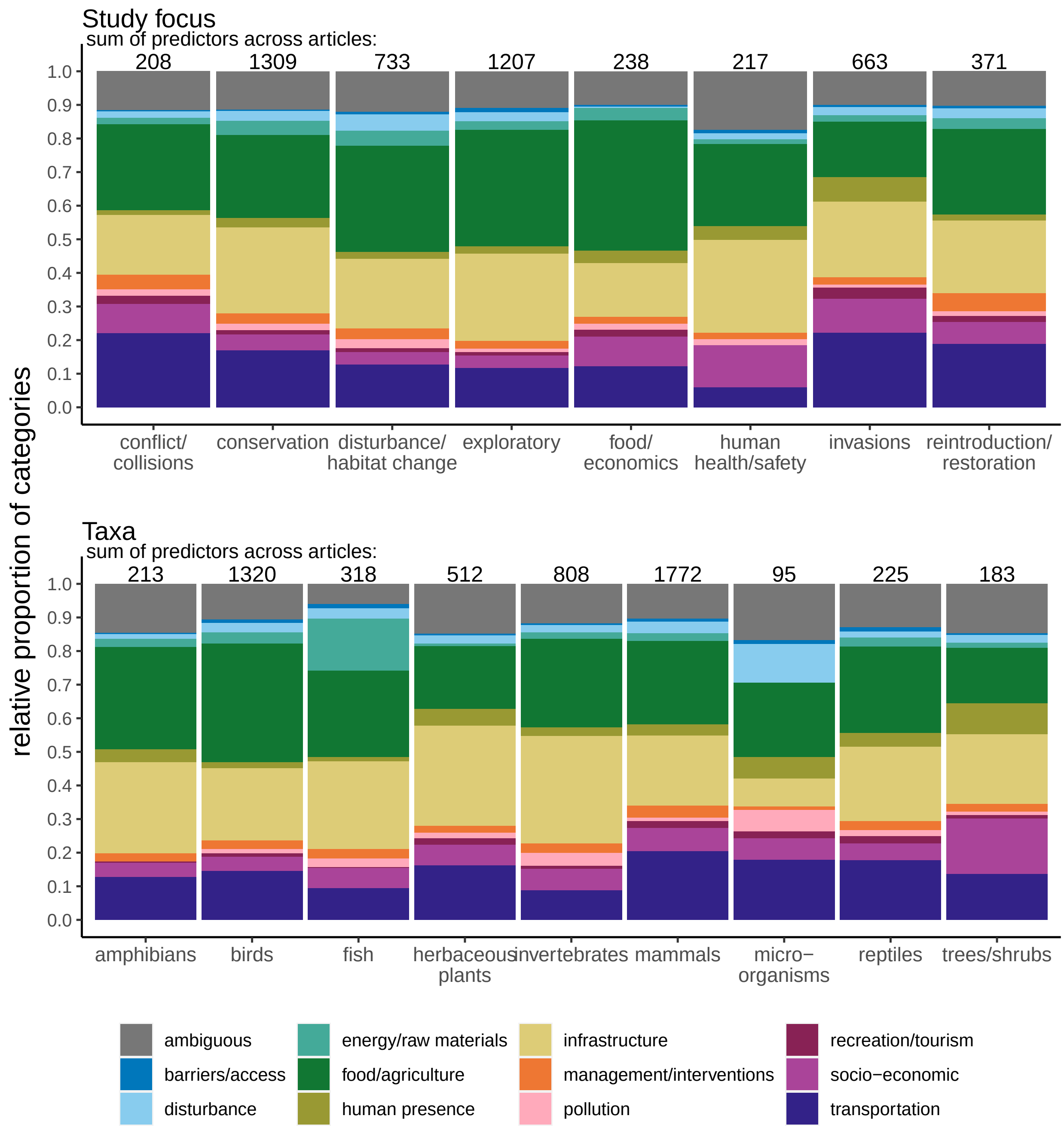
```
# legend only
  leg <- ggplot(tax.preds.list) +
            aes(x=taxa, fill=category) +
            geom_bar(position = 'fill') +
            scale_fill_manual('',
                              values = c('#777777','#0077BB','#88CCEE','#44AA99',
                                         '#117733','#999933','#DDCC77','#EE7733',
                                         '#FFAABB','#882255','#AA4499','#332288')) +
            theme(legend.position = 'bottom')

# extract legend for multi-plot
  require(ggpubr)
  leg <- ggpubr::get_legend(leg)

# preview here
  #as_ggplot(leg)
```

5.3 Multiplot

Put all plots into a 2x2 grid.

```
require(cowplot)
```

```
# multiplot
  cow_AB <- plot_grid(foc_plt + ylab('') + xlab(''),
                      tax_plt + ylab('') + xlab(''),
            ncol = 1,
            rel_heights = c(1,1),# A negative rel_height shrinks space between elements
            axis='bt',
            label_y='proportion of category use')
            #label_size = 18,
            #label_fontfamily = "sans")
  # add legend
    cow_AB <- plot_grid(cow_AB, leg, ncol = 1, rel_heights = c(1, .1),rel_widths = c(1,1))
  # add y-axis label
    cow_AB <- cow_AB +
             theme(plot.margin = unit(c(.1,.1,.2,.1), "cm")) +
             draw_label("relative proportion of categories",
                        x=-.01, y=0.5, vjust= 1.5, angle=90)

# save
  ggsave(plot=cow_AB, filename = paste0(image.dir,'predictor_category_use2.png'),
         height = 8, width = 7.5, units = 'in', dpi = 600)
  ggsave(plot=cow_AB, filename = paste0(image.dir,'predictor_category_use2.svg'),
         height = 8, width = 7.5, units = 'in')

# show here
  cow_AB
```



6 Understanding ambiguous predictor use

Next, we investigate the trends in usage of ambiguous predictors.

Get total count of articles using ambiguous predictors.

```
# subset ambiguous predictors
  amb_only <- preds.list.shorter[preds.list.shorter$category=='ambiguous',]

# get count of unique papers using ambiguous predictors
```

```r
  amb_only$papers2 = as.character(amb_only$papers)

# mutate data, with repeated rows of papers and predictors
  amb_only_long <- amb_only %>%
                    rownames_to_column() %>%
                    mutate(string = strsplit(papers2, "; ")) %>%
                    unnest %>%
                    group_by(string)

# get vector list of unique papers (for matching later)
  amb_only_papers <- summarise(amb_only_long,
                               uid = unique(string))

# save list
  write.csv(amb_only_papers,
            paste0(data.dir,"ambiguous_predictor_paper_list.csv"),
            row.names = FALSE)

# get count of unique papers
  paste(summarise(amb_only_long,count = length(unique(string))),
        "papers use ambiguous predictors")
```

```
## [1] "490 papers use ambiguous predictors"
```

Compare total number of ambiguous predictors used by each article with the total number of human predictors they used in their SDMs.

```r
# get table of predictors
  # Pull up original table and subset
    preds.df <- data.frame(subset(yes.df,
                                  select = c("uid","domain","num_env_preds","num_hum_preds")))

  # calculate total predictors across studies (this will help include past-only papers)
    preds.df$total_preds <- as.integer(preds.df$num_env_preds) + as.integer(preds.df$num_hum_preds)

  # set up factors
    preds.df$domain <- as.factor(preds.df$domain)
    preds.df$domain <- factor(preds.df$domain, levels = c("terrestrial", "freshwater", "marine"))

# get count of ambiguous predictors per paper
  amb.df <- ddply(amb_only_long,.(string), summarize,amb_count=length(predictor))
  amb.df <- data.frame(uid = amb.df$string, amb_count = amb.df$amb_count)

# extract rows matching paper IDs
  hum_preds_cts <- data.frame(uid = as.character(preds.df$uid),
                              hum_count = as.integer(preds.df$num_hum_preds))
  amb.df <- left_join(amb.df,hum_preds_cts, by='uid')

# count instances where amb_count >= hum_count ('>' is OK since diff time frames here)
  paste(nrow(amb.df[amb.df$amb_count >= amb.df$hum_count,]),
        "papers use only ambiguous predictors as the human predictors in their SDMs")
```

```
## [1] "197 papers use only ambiguous predictors as the human predictors in their SDMs"
```

Save

```
# save list
  write.csv(amb.df,
            paste0(data.dir,"ambiguous_predictor_dataframe.csv"),
            row.names = FALSE)
```

7 Understanding buffered predictors

```
# Make a subset
  buffer.preds <- preds.list.shorter[grepl('radius',preds.list.shorter$predictor)|
                                       grepl('buffer',preds.list.shorter$predictor),]

# lengthen list
  buffer.preds <- separate_rows(buffer.preds, papers, sep="; ",convert = TRUE)

# total number of articles using buffered predictors
  length(unique(buffer.preds$papers))
```

```
## [1] 68
```

```
# total number of predictors
  length(unique(buffer.preds$predictor))
```

```
## [1] 409
```

```
# get summaries per predictor category
  buffer.sum <- ddply(buffer.preds,.(category),
                     summarize,
                   # get number of papers using buffers per category
                     count_preds = length(unique(predictor)),
                    # list of taxa for each predictor
                     count_papers = length(unique(papers)),
                    # list of study focus for each predictor
                     predictor = paste(unique(predictor),collapse="; ")
                   )

# save table
  write.csv(buffer.sum,
            paste0(data.dir,"buffer_predictor_dataframe.csv"),
            row.names = FALSE)

# show summary (only first three columns)
  kableExtra::kbl(buffer.sum[,1:3],booktabs=T, longtable=T) %>%
    kable_styling(latex_options = c("striped","repeat_header"))
```

category	count_preds	count_papers
ambiguous	16	12

(continued)

category	count_preds	count_papers
barriers/access	1	1
disturbance	7	3
energy/raw materials	8	4
food/agriculture	129	38
infrastructure	205	42
management/interventions	5	3
pollution	2	2
recreation/tourism	2	1
socio-economic	11	5
transportation	23	14

8 Article time frames compared to human predictor time frames

```
# Create a subset of relevant studies with only the paper ID and the study time frames
  time.df <- subset(yes.df, select = c("uid", "time", "hum_time"))

# Ensure that duplicates are removed (uid is duplicated for multiple domains)
  colnm <- c("uid", "time","hum_time")
  time.df[colnm] <- lapply(time.df[colnm], factor)
  time.df <- time.df[!duplicated(time.df[,c("uid")]),]

# Calculate (note that this is number of papers, not studies)
  time.list <- ddply(time.df, .(time, hum_time), summarize,
                              count=length(hum_time))

# Show table
  kableExtra::kbl(time.list,booktabs=T, longtable=T) %>%
    kable_styling(latex_options = c("striped","repeat_header"))
```

time	hum_time	count
past-future	past-future	1
past-future	past-only	1
past-only	past-only	6
past-present	past-present	40
past-present	present-only	14
past-present-future	past-present	3
past-present-future	past-present-future	7
past-present-future	present-future	1
present-future	present-future	86
present-future	present-only	112
present-only	present-only	1148
present-past	present-past	5
present-past-future	present-only	5

A good number of papers have a mismatch in time frames. We can show the mismatch more clearly by doing another summary.

```
# extract past papers only
  time.list_past <- time.list[grepl('past',time.list$time),]

# extract past/present-future papers only
  time.list_futr <- time.list[grepl('future',time.list$time),]

# add indicator for mismatches
  time.list_past <- ddply(time.list_past, .(time, hum_time, count), summarize,
                                         indicator=sum(hum_time != time))
  time.list_futr <- ddply(time.list_futr, .(time, hum_time, count), summarize,
                                         indicator=sum(hum_time != time))
  time.list_cts <- ddply(time.list, .(time, hum_time, count), summarize,
                                         indicator=sum(hum_time != time))

# get counts
  paste('studies with past mismatches:',
         sum(time.list_past$count[time.list_past$indicator==1]));
  paste('studies with future mismatches:',
         sum(time.list_futr$count[time.list_futr$indicator==1]));
  paste('total mismatches:',
        sum(time.list_cts$count[time.list_cts$indicator==1]))
```

```
## [1] "studies with past mismatches: 24"
## [1] "studies with future mismatches: 122"
## [1] "total mismatches: 136"
```

Note that the sum of past and future mismatches has overlap due to e.g. "past-present-future" studies.

8.1 Time frame chord diagram

Next, we make a cord diagram showing the study time frames compared to the human predictor time frames. This is to get a better visualization of the proportion of matches and mismatches across studies and across the time frame varieties.

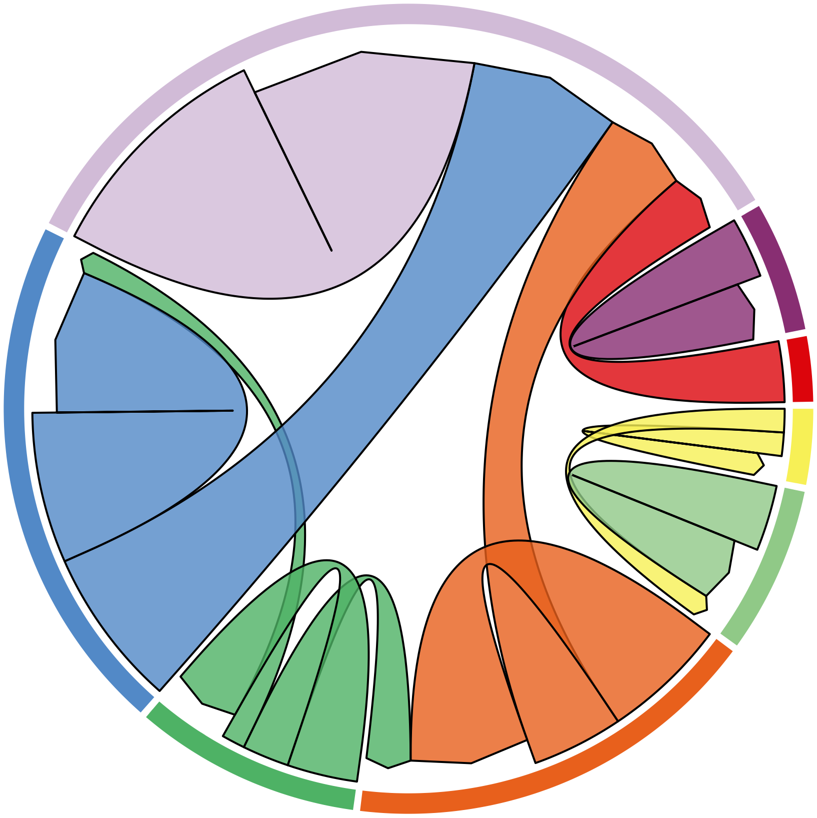
```
# remove "only" from time.list
  time.list <- data.frame(lapply(time.list, function(x) {gsub("-only","", x)}))

# change factor levels
  time.list$time <- factor(time.list$time,
                         levels = c("past",
                                    "past-present",
                                    "past-present-future",
                                    "past-future",
                                    "present",
                                    "present-past",
                                    "present-past-future",
                                    "present-future"
                                    ))

  time.list$hum_time <- factor(time.list$hum_time,
```

```
                              levels = c("past",
                                         "past-present",
                                         "past-present-future",
                                         "past-future",
                                         "present",
                                         "present-past",
                                         "present-future"
                                         ))

# revert to numeric
  time.list$count <- as.numeric(time.list$count)

# make labels (optional)
  stu.lab <- ddply(time.list, .(time), summarize,
                             time_lab=paste0(time,' (',sum(count),')'))
  hum.lab <- ddply(time.list, .(hum_time), summarize,
                             hum_lab=paste0(hum_time,' (',sum(count),')'))

# join labels to table
  time.list <- left_join(time.list,stu.lab,by='time')
  time.list <- left_join(time.list,hum.lab,by='hum_time')
  time.list <- distinct(time.list, .keep_all = TRUE) # ensure no repeated rows

# copy time.list
  time.list2 <- time.list

# change values to log + 1 scale for easier visualization
  time.list2$count <- log(time.list$count + 1)

  time.col <- c('#F7F056','#90C987','#E8601C','#4EB265',
              '#5289C7','#D1BBD7','#882E72','#DC050C')


  svg(paste0(image.dir,"time_frames_circle_1-30-2024.svg"),
      height=8,width=8)
  plot.new()
  # base chord diagram
    chordDiagram(
      x = time.list2,
      grid.col = time.col,
      link.border = TRUE,
      transparency = 0.2,
      directional = 1,
      direction.type = c("arrows", "diffHeight"),
      diffHeight  = -0.04,
      keep.diagonal = TRUE,
      annotationTrack = "grid",
      annotationTrackHeight = c(0.05, 0.1),
      link.arr.type = "big.arrow",
      link.sort = TRUE,
      link.largest.ontop = TRUE)

  # # add text and axis (activate to check)
```

```
#   circos.trackPlotRegion(
#     track.index = 1,
#     bg.border = NA,
#     panel.fun = function(x, y) {
#
#       xlim = get.cell.meta.data("xlim")
#       sector.index = get.cell.meta.data("sector.index")
#
#       # # Add names to the sector
#       # circos.text(
#       #   x = mean(xlim),
#       #   y = 2,
#       #   labels = sector.index,
#       #   facing = "clockwise",
#       #   #facing = "bending",
#       #   cex = 1
#       #   )

 # }
#)
dev.off()
```

```
## pdf
##   2
```

```
# plot again to show here
chordDiagram(
  x = time.list2,
  grid.col = time.col,
  link.border = TRUE,
  transparency = 0.2,
  directional = 1,
  direction.type = c("arrows", "diffHeight"),
  diffHeight  = -0.04,
  keep.diagonal = TRUE,
  annotationTrack = "grid",
  annotationTrackHeight = c(0.05, 0.1),
  link.arr.type = "big.arrow",
  link.sort = TRUE,
  link.largest.ontop = TRUE)
```



This figure was exported as an SVG and edited in PowerPoint, with labels done manually.

9 Assessing predictor use over time

It is possible that human predictor use/selection is also dependent on data availability–especially if modelers are not creating these predictors themselves. Here, we make figures to evaluate when human predictors have begun being used, using first years of publication per predictor as an indicator.

First, we add the years of publication to the predictor dataset (`prdotf.list.long`), matching the paper UID for each predictor.

```
# subset yes.df to only UID, study area, scale, and publication year
  predtime.df <- subset(yes.df, select = c('uid','year'))

# set structure
  predtime.df$uid <- as.character(predtime.df$uid)
```

```
# left-join to predictor list
  predtime.df <- left_join(prdotf.list.long, predtime.df, by='uid')
  predtime.df <- distinct(predtime.df, .keep_all = TRUE) # ensure no repeated rows

# subset predictor list
  predtime.df <- subset(predtime.df,
                        select = c('category', 'data_type', 'predictor', 'uid', 'year'))
```

Next, we summarize the predictor list, extracting the first and last years of publication and frequency of unique articles.

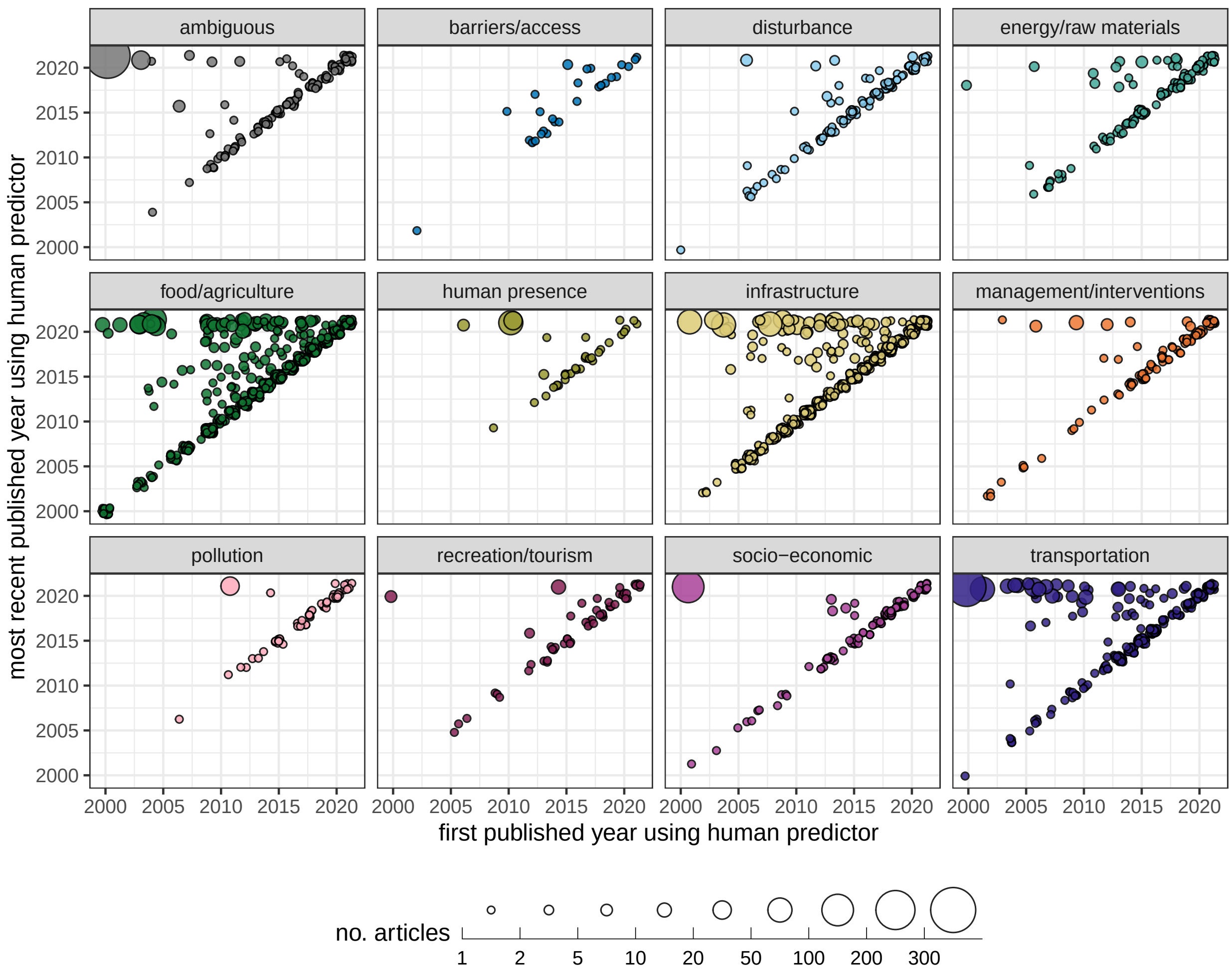
```
# make summary table
  predtime.df <- ddply(predtime.df,
                        .(category, data_type, predictor),
                        summarize,
                        # list of papers for each predictor
                        uid=paste(unique(uid),collapse="; "),
                        # count number of papers using each predictor
                        count=as.integer(paste(length(unlist(strsplit(uid,";"))))),
                        # get first published year of predictor use
                        first_year=min(year),
                        # get last published year of predictor use
                        last_year=max(year)
                        )
```

9.1 Predictor selection over time

Create a bubble plot showing the first and last year of use of a human predictor across the various predictor categories.

```
# set colors for predictor categories
  cat.col <- c('#777777','#0077BB','#88CCEE','#44AA99','#117733','#999933',
               '#DDCC77','#EE7733','#FFAABB','#882255','#AA4499','#332288')

# bubble scatterplot
  bub <- ggplot(predtime.df,
                aes(x=first_year, y=last_year,
                    size=count, fill=category)) +
                geom_jitter(shape=21, alpha=0.85) +
                # increasing point sizes for total number of variables used
                #scale_size_area(name="no. articles",max_size = 10)+
                scale_size_binned(range = c(1, 10),
                                  n.breaks = 10,
                                  breaks = c(1,2,5,10,20,50,100,200,300,400),
                                  name="no. articles",
                                  ) +
                scale_fill_manual(name="domain",values=cat.col,
                                  guide=FALSE) +
                xlab("first published year using human predictor")+
                ylab("most recent published year using human predictor")+
                theme_bw() +
                facet_wrap(~category, scales = "fixed") +
```

```
                 theme(legend.position = 'bottom')

# save
  ggsave(filename=paste0(image.dir,"predictor_time_bubbleplot.png"),
         plot=bub, height=6.5, width=8, units = 'in', dpi = 600)
  ggsave(filename=paste0(image.dir,"predictor_time_bubbleplot.svg"),
         plot=bub, height=6.5, width=8, units = 'in')

# plot
  bub
```



For each category, how many predictors have persisted beyond their first year of publication? We can summarize this as a table here.

```
# make summary table
  predsums.df <- ddply(predtime.df,
                       .(category),
                       summarize,
                       # count persistence
                       count_once=sum(first_year == last_year),
                       count_kept=sum(first_year != last_year),
```

```
                         # percent persistence
                         perc_once=round(count_once/sum(count_kept,count_once),2),
                         perc_kept=round(count_kept/sum(count_kept,count_once),2)
                         )

# display here
  kableExtra::kbl(predsums.df, booktabs=T, longtable=T) %>%
    kable_styling(latex_options = c("striped","repeat_header"))
```

category	count_once	count_kept	perc_once	perc_kept
ambiguous	100	15	0.87	0.13
barriers/access	22	7	0.76	0.24
disturbance	102	13	0.89	0.11
energy/raw materials	109	18	0.86	0.14
food/agriculture	609	125	0.83	0.17
human presence	29	7	0.81	0.19
infrastructure	532	85	0.86	0.14
management/interventions	92	11	0.89	0.11
pollution	52	3	0.95	0.05
recreation/tourism	46	7	0.87	0.13
socio-economic	89	7	0.93	0.07
transportation	179	48	0.79	0.21

It is also possible that the years of use will vary across countries. We visualize this in Part IV.

10 Save final predictor table, including study context

We also want to add this information to the predictor list table, for use in the supplementary materials of the corresponding article.

```
# append years to predictor list
  years.df <- subset(yes.df, select = c('uid','year'))
  years.df$uid <- as.character(years.df$uid)

# left-join to predictor list
  prdotf.list.long <- left_join(prdotf.list.long, years.df, by='uid')
  prdotf.list.long <- distinct(prdotf.list.long, .keep_all = TRUE) # ensure no repeated rows

# Get a count of predictors in general
  preds.list.export <- ddply(prdotf.list.long,
                             .(category, data_type, predictor, timeframes),
                             summarize,
                             # list of taxa for each predictor
                             taxa=paste(unique(taxa),collapse="; "),
                             # list of study focus for each predictor
                             study_focus=paste(unique(study_focus),collapse="; "),
                             # list of papers for each predictor
                             uid=paste(unique(uid),collapse="; "),
                             # count number of papers using each predictor
```

```
                            count=paste(length(unlist(strsplit(uid,";")))),
                            # get first published year of predictor use
                            first_year=min(year),
                            # get last published year of predictor use
                            last_year=max(year)
                            )

# save table
  write.csv(preds.list.export,
            paste0(data.dir,"predictor_list_summary_FINAL.csv"),
            row.names = FALSE)
```

11 Save

```
# save progress
  save.image("SDMs_human_lit_review_III.RData")
```

THIS IS THE END OF THE SCRIPT.

See "Human Influence in SDMs: Literature Review (Part IV)" for next steps.
