Spatio Temporal Analysis Of Sustainability Data in Central Europe

Bachelor Thesis in Computational Statistics

Marijana Petojevic

26.09.2024

Table of Contents

Introduction

Sustainability is a critical challenge facing today's society, particularly in Central Europe, where land use, environmental degradation, and industrial activities are significant factors. To address these issues, it is essential to monitor and analyze spatio-temporal data over time, uncovering patterns that can inform policy and decision-making. This report focuses on the spatio-temporal analysis of sustainability data, combining diverse datasets on land cover, mining locations, burned areas, and emissions across multiple years (2018-2023). These datasets, gathered from various sources such as the *Land Cover Data Set*, *Global Mining Locations Data*, and the *Global Wildfire Information System*, offer a comprehensive view of environmental changes in the region. Additional datasets from *Global Forest Watch Dashboards*, covering 2001–2023, are utilized to offer deeper insights into factors influencing tree cover changes in Central Europe.

The project is structured into three main parts. The first part, **Data Collection and Preparation**, involves gathering and processing relevant spatial data. The second part, **Data Description**, provides a visual and descriptive analysis of the data, examining its spatio-temporal distribution and identifying the impact of various environmental factors on tree cover changes. Finally, the third part employs different **Clustering Methods** to group the spatio-

temporal data into clusters, providing meaningful insights into sustainability trends across the region.

This approach enables a nuanced understanding of how environmental changes are evolving over time and space, allowing for more effective and targeted actions to improve sustainability in Central Europe.

Data Collection and Preparation

Land Cover Data Set

To perform a spatio-temporal analysis of tree cover change in a specific country from 2018 to 2023, we sourced geospatial data from the *Living Atlas*. This dataset provides annual land cover information, including tree cover, in a tile-based GeoTiff format. The *Living Atlas* offers users the ability to explore various land cover types through satellite imagery, covering categories such as *Trees, Water, Crops, Snow/Ice, Built Areas, Clouds, Flooded Vegetation, Bare Ground, and Rangeland* for the period between 2018 and 2023. More detailed information about the dataset is available *here*.

Since the dataset is available in a tile-based GeoTIFF format, allowing for high-resolution geographic data to be analyzed over time, to efficiently collect the relevant data for the Central European region, I developed a custom function to automate the process of generating download links and retrieving the necessary files.

Downloading Land Cover Data

The `generate_links` function creates download links for each year by constructing URLs that point to the tiles covering Central Europe. These tiles correspond to specific grid codes, ensuring full coverage of the region. The data spans from January 1st of one year to January 1st of the next, and this process was repeated for each year from 2018 to 2023.

```
generate_links <- function(year) {
  base_url <- "https://lulctimeseries.blob.core.windows.net/lulctimeseriesv003/lc"
  grid_codes <- c("32U", "33U", "34U", "32T", "33T", "34T")

  links <- paste0(
    base_url, year, "/",
    grid_codes, "_",
    year, "0101-", year + 1, "0101.tif"
  )

  return(links)
}
```

After constructing the links for the years 2018–2023, the files were downloaded and organized into directories for each year.

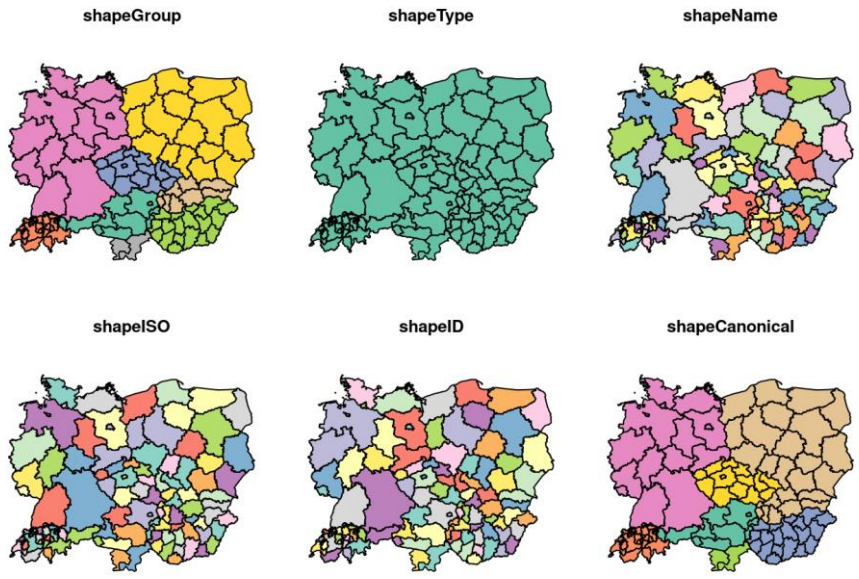
```
for (year in names(download_links)) {
  dir.create(year, showWarnings = FALSE)

  for (l in download_links[[year]]) {
    download.file(
```

```
      url = l,
      destfile = paste0(year, "/", basename(l)),
      mode = "wb",
      method = "wget"
    )
  }
}
```

Loading Spatial Data

To focus the analysis on specific countries in Central Europe, I retrieved spatial boundary data for Austria, Germany, Czech Republic, Poland, Slovakia, Hungary, Switzerland, and Slovenia. These country boundaries were combined into a single spatial object to represent the geographic extent of the study. Each country is represented by its administrative regions of the first level.

```
central_europe <- rbind(gb_adm1("Austria"), gb_adm1("Germany"),
                        gb_adm1("Czech Republic"), gb_adm1("Poland"),
                        gb_adm1("Slovakia"), gb_adm1("Hungary"),
                        gb_adm1("Switzerland"), gb_adm1("Slovenia"))
```



Boundaries Of Central Europe Conducted Through `rgeoboundaries` *package*

Retrieving and Rasterizing GeoTIFF Files

The *get_files* function retrieves the previously downloaded GeoTIFF images for each year, organizing them into lists that can be processed further.

```
get_files <- function(year) {
  path <- file.path(as.character(year))
  pattern <- paste0(year + 1, "0101.tif")
  list.files(path = path, pattern = pattern, full.names = TRUE)
}

files_list <- lapply(years, get_files)
names(files_list) <- years
```

Once the files were loaded into R, they were rasterized using the *terra::rast* function, preparing them for further geo-spatial operations.

```
rasters_list <- lapply(files_list, function(file_paths) {
  lapply(file_paths, terra::rast)
})
```

Cropping and Masking Rasters

To ensure that the analysis focuses solely on Central Europe, the raster data was cropped and masked based on the spatial boundaries of the selected countries. The *terra:aggregate* function reduces the resolution for computational efficiency, while the *terra::project* function ensures that the rasters have a consistent coordinate reference system (CRS). The resulting rasters were saved in *GeoTIFF* format, enabling the creation of virtual raster layers. These virtual layers are more computationally efficient and better suited for handling large-scale geospatial data during analysis.

```
crs <- "EPSG:4326"

for (index in seq_along(rasters_list)) {
  year <- years[index]
  rasters <- rasters_list[[index]]

  for (i in seq_along(rasters)) {
    raster <- rasters[[i]]

    country <- central_europe %>%
      sf::st_transform(crs = terra::crs(raster))

    land_cover <- raster %>%
      crop(vect(country), snap = "in") %>%
      mask(vect(country)) %>%
      aggregate(fact = 5, fun = "modal") %>%
      terra::project(crs)

    output_dir <- paste0(year)
    dir.create(output_dir, showWarnings = FALSE, recursive = TRUE)

    terra::writeRaster(
      land_cover,
      paste0(output_dir, "/", i, "_central_europe_", year, ".tif"),
      overwrite = TRUE
    )
  }
}
```

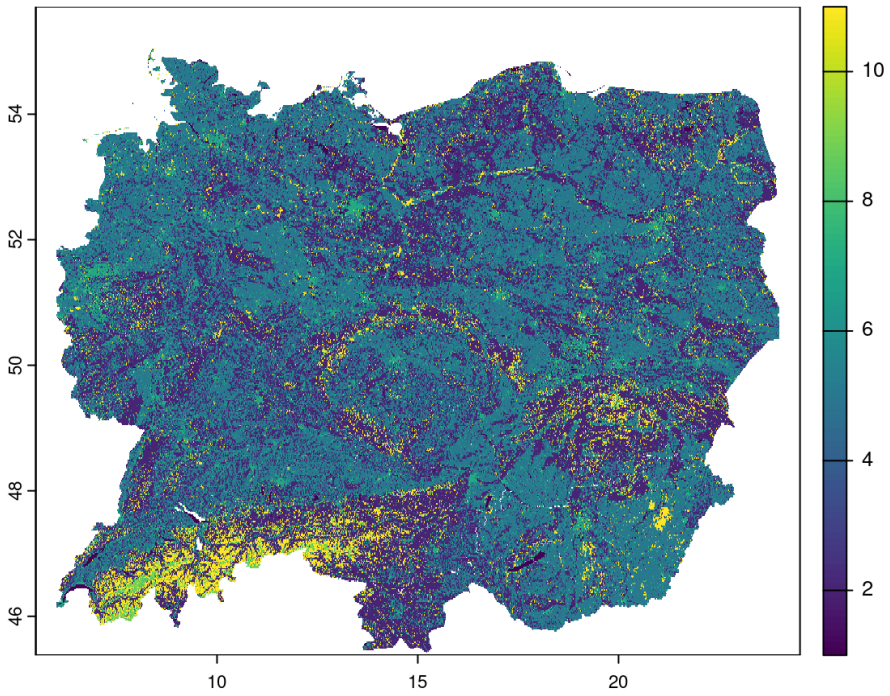
Creating Virtual Raster Layers

To facilitate visualization and analysis, I created virtual raster layers for each year by combining the individual tiles into a single continuous raster. The *create_vrt* function generates these virtual layers, allowing for efficient handling of large spatial data. The virtual rasters represent the land cover for each year across the entirety of Central Europe, providing a streamlined view of the data.

```
create_vrt <- function(year) {
  r_list <- list.files(
```

```
    path = paste0(year),
    pattern = paste0("_central_europe_", year),
    full.names = TRUE
  )
  vrt_name <- paste0("central_europe_vrt", year, ".vrt")
  terra::vrt(r_list, vrt_name, overwrite = TRUE)
}

virtual_layers <- lapply(years, create_vrt)
```

By plotting one of these virtual layers, I was able to confirm that the cropping and rasterization processes were executed correctly.



Virtual Layer of Central Europe for 2018

Extracting Meaningful Data From Rasters

In the Land Cover dataset, each coordinate point is assigned one of 10 possible values, representing different land cover types. These values correspond to categories such as *water* (`value 1`), *Trees* (`value 2`), *Flooded vegetation* (`value 4`), *Crops* (`value 5`), *Built area* (`value 7`), *Bare Ground* (`value 8`), *Snow/Ice* (`value 9`), *Clouds* (`value 10`) or *Rangeland* (`value 11`). The visual representation of these points is determined by their color on the raster, which indicates the type of land cover. Further details about these variables can be found *here*. For more information on how the values correspond to their original colors, refer to my submission on *GitHub*, where I first explored the Land Cover dataset.

To analyze land cover data across different years, we used the `exact_extract` function from the `exactextractr` package. This function calculates the total area (in km²) of each administrative region in Central Europe and determines what percentage of that area is covered by specific land

cover types, such as water, trees, and built areas. The function `calculate_coverage_percentage` was created to perform these calculations for each region and store the results in separate data frames for each year. Each data frame contains the total area of a region and the percentage of that region covered by various land cover types.

```r
calculate_coverage_percentage <- function(raster, polygons) {
  exactextractr::exact_extract(
    raster,
    polygons %>% sf::st_transform(crs = terra::crs(raster)),
    function(df) {
      df %>%
        dplyr::group_by(shapeName) %>%
        dplyr::summarize(
          total_area_km2 = sum(coverage_area / 1e6, na.rm = TRUE),

          water_percentage = sum(coverage_area[value == 1] / 1e6, na.rm = TRUE) /
            total_area_km2 * 100,

          tree_percentage = sum(coverage_area[value == 2] / 1e6, na.rm = TRUE) /
            total_area_km2 * 100,

          flooded_vegetation_percentage = sum(coverage_area[value == 4] / 1e6, na.rm = TRUE) /
            total_area_km2 * 100,

          crops_percentage = sum(coverage_area[value == 5] / 1e6, na.rm = TRUE) /
            total_area_km2 * 100,

          built_area_percentage = sum(coverage_area[value == 7] / 1e6, na.rm = TRUE) /
            total_area_km2 * 100,

          bare_ground_percentage = sum(coverage_area[value == 8] / 1e6, na.rm = TRUE) /
            total_area_km2 * 100,

          snow_ice_percentage = sum(coverage_area[value == 9] / 1e6, na.rm = TRUE) /
            total_area_km2 * 100,

          clouds_percentage = sum(coverage_area[value == 10] / 1e6, na.rm = TRUE) /
            total_area_km2 * 100,

          rangeland_percentage = sum(coverage_area[value == 11] / 1e6, na.rm = TRUE) /
            total_area_km2 * 100
        )
    },
    summarize_df = TRUE,
    coverage_area = TRUE,
    include_cols = "shapeName"
  )
}
```

Applying function:

```r
extracted_values <- list()

for (i in seq_along(virtual_layers)) {
  rasters <- virtual_layers[[i]]

  lc <- calculate_coverage_percentage(
    raster = rasters,
```

```
    polygons = central_europe
  )

  extracted_values[[i]] <- lc
}
names(extracted_values) <- years
```

Renaming and Merging Data

Once the data frames were generated for each year, the next step involved renaming the columns to include the corresponding year in the column name. This is done for all land cover percentage columns, ensuring that the data from different years can be easily distinguished and later merged into a single data frame.

```
for (i in seq_along(extracted_values)) {
  year <- names(extracted_values)[i]
  df <- extracted_values[[i]]

  col_names <- colnames(df)
  start_index <- which(col_names == "water_percentage")

  colnames(df)[start_index:length(col_names)] <-
        paste0(col_names[start_index:length(col_names)],  "_", year)

  extracted_values[[i]] <- df
}
```

Finally, all the data frames for the different years were merged into one comprehensive data frame. This allows for a unified view of land cover changes across Central Europe from 2018 to 2023, where each region's total area and percentage coverage for different land cover types are tracked year by year:

```
combined_df <- extracted_values[[1]]

for (i in 2:length(extracted_values)) {
  combined_df <- merge(
    combined_df,
    extracted_values[[i]],
    by = c("shapeName", "total_area_km2"),
    all = TRUE
  )
}
```

Mining Data

The mining location data was obtained from *data.world*, and I chose to use the GeoTIFF image in 30 arcsecond resolution for the best spatial accuracy. The data file can be downloaded from *this link*. Since the GeoTIFF image contains global mining locations, the raster was cropped to focus on Central Europe for the purpose of this analysis. Upon examining the dataset, I found that raster points with values greater than zero represent mining areas. Using this information, we extracted the coverage of mining areas in each administrative region in Central Europe. Additionally, we calculated the percentage of each region's area covered by mines.

The following code extracts mining area coverage for each region:

```r
mine_cover_central_europe_df <- exactextractr::exact_extract(
  mining_raster,
  central_europe %>% sf::st_transform(crs = terra::crs(mining_raster)),
  function(df) {
    df %>%
      dplyr::group_by(shapeName) %>%
      dplyr::summarize(
        total_area_km2 = sum(coverage_area / 1e6, na.rm = TRUE),
        total_mine_cover_km2 = sum(coverage_area[value > 0] / 1e6, na.rm = TRUE),
        mine_cover_of_state_percentage = if (any(!is.na(value) & value > 0)) {
          sum(coverage_area[value > 0] / 1e6, na.rm = TRUE) / total_area_km2 * 100
        } else {
          NA_real_
        }
      )
  },
  summarize_df = TRUE,
  coverage_area = TRUE,
  include_cols = "shapeName"
)
```

After extracting the mining data, the new columns were integrated into our existing dataset. For more detailed information on this process and the mining locations dataset, refer to the *Central Europe Data Collection* document available on my GitHub page.

Burned Areas Data

The *Global Monthly Burned Area [2002 - 2023]* dataset was sourced from the *Global Wildfire Information System* and can be downloaded from this *link*. It provides information on the monthly burned areas (in hectares) for each country and its administrative regions. The data spans from 2002 to 2023. For this project, I filtered the dataset to only include records from 2018 to 2023, as these were the years relevant to my analysis.

```r
{r}
names(burned_areas)
```

```
 [1] "year"                "month"               "gid_0"               "country"
"gid_1"                "region"              "forest"
 [8] "savannas"            "shrublands_grasslands" "croplands"           "other"
```

Columns in Burned Areas Data Set

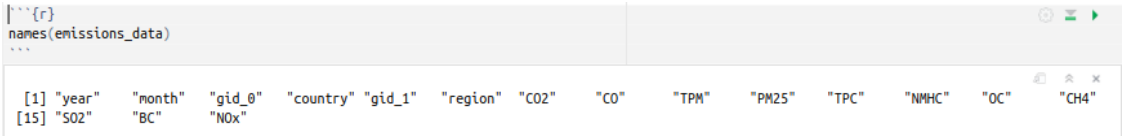
The dataset differentiates between various land types affected by burning, such as *forests, savannas, shrubland/grasslands, croplands,* and *other* categories. I summarized this data by year and region to calculate the total burned area for each region between 2018 and 2023:

```r
burned_areas_summary <- burned_areas %>%
  group_by(year, region) %>%
  summarize(
    total_burned_area = sum(forest, savannas, shrublands_grasslands, croplands, other,
                        na.rm = TRUE))
```

To merge this dataset with our existing data, I cleaned the data and transformed it into a wide format. More information on this process can be found again in the *Central Europe Data Collection* document on my GitHub Page.

Central Europe Emissions Data

GFED Global Monthly Emissions [2002 - 2023] data set can also be found on *Global Wildfire Information System* and can be downloaded following this *link*. provides monthly emissions data for various pollutants, such as CO2, CO, PM25, and CH4, across the globe. This dataset contains emission observations for first-level administrative regions of each country in the world.



Emission Data Set Columns

For this project, I extracted only the relevant data for Central European regions between 2018 and 2023. Since the original dataset contains monthly emissions, I summarized the data to obtain yearly emission totals for each region. Afterward, the data was transformed into a wide format to facilitate integration with the main dataset.Details on the entire process of handling and summarizing the emissions data can be found in the *Central Europe Data Collection* document on my GitHub Page.

Dataset Representation

Final data set can be viewed and downloaded *here*.

Global Forest Watch Dashboard Data

For the Shiny App, which will be discussed in detail later, I also sourced data from the *Global Forest Dashboards*. The dashboard provides downloadable data for each country for various purposes. The datasets I used include **Tree Cover Loss (2001-2020)**, **Annual Tree Cover Loss by Dominant Driver**, **Tree Cover Loss Due to Fires**, and **Components of Net Change in Tree Cover**. This data was downloaded separately for each country and then merged into separate data frames to cover the entire Central European region. These data frames can be accessed on my GitHub page *here* and code for on data collection process can be found in *this file*.

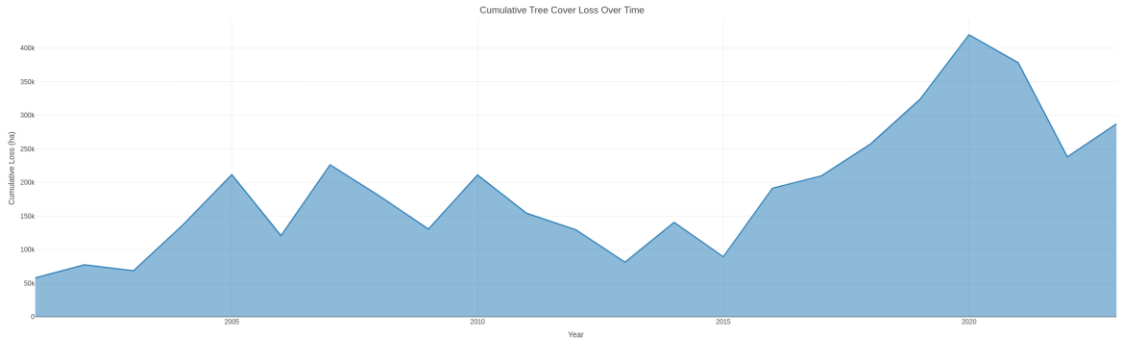
Sample Description

The following section will outline the general structure of the datasets described in the previous section. We will examine the metrics that characterize this data and highlight trends in tree percentage changes, alongside changes in other features that may have influenced tree changes over the course of the last five years.

To enable users to explore and interact with data conducted in previous step, a **Shiny App** was developed. The app allows users to visually inspect the dataset and interact with the various plots, enabling them to analyze tree cover trends, identify regional patterns, and examine the key drivers of forest change. With this interactive tool, users can better understand the spatial and temporal aspects of tree cover loss across Central Europe. App with used datasets needed for it to run can be downloaded from *this link*.

General Data Description

One of the key metrics derived from the data is the **cumulative tree cover loss**, which captures how much forest cover has been lost over time across the study regions. The following plot illustrates the cumulative tree cover loss (measured in hectares) between 2000 and 2023:



The plot highlights several periods of significant tree loss, notably around 2010 and 2020, with peaks reaching over 400,000 hectares of cumulative loss. Understanding these losses is essential for analyzing the effectiveness of forest management practices and identifying regions most vulnerable to deforestation.

This cumulative tree loss can be further explored by examining the spatial distribution of changes in tree cover, as shown in subsequent sections. Moreover, the Shiny App allows users to break down these losses by region and correlate them with other factors like built area expansion, crop percentage changes, rangeland percentage and gas emissions.

Tree cover Loss in Central European Countries

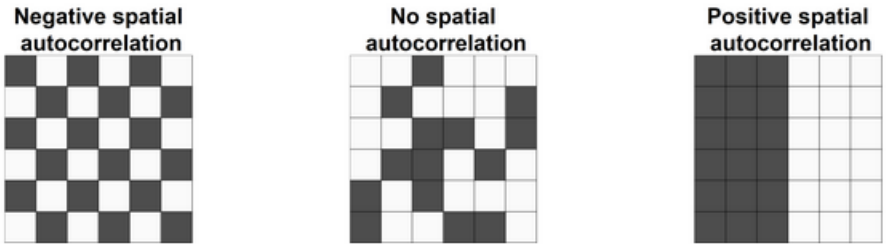
Tree cover loss is significant across all Central European countries, but the factors driving these changes and the influence of other environmental variables remain unclear. To better understand what affects tree cover change in the Central European region, we will conduct a descriptive statistical analysis and aim to identify evident patterns in tree cover change.

Spatial Autocorrelation

Note: This and following sections will include some code snippets, the whole code which you can run and play with it, can be found on my Github page in *this document.*

Spatial autocorrelation describes the extent to which a variable is correlated with itself across space. A similar concept is found in *Tobler's First Law of Geography*, which states that *"everything is related to everything else, but near things are more related than distant things"*.

Positive spatial autocorrelation occurs when similar values are clustered together, while negative spatial autocorrelation occurs when dissimilar values are located near each other. Spatial autocorrelation can be assessed using indices that summarize the degree to which similar observations occur near each other within a dataset.



Configurations showing different types of spatial autocorrelation

As we are working with spatio-temporal data, we will further examine the spatial autocorrelation of our variable of interest, `tree_percentage`, for each of the five years from 2018 to 2023. To understand how the spatial dimension affects tree growth percentages, this section will employ **Moran's I** statistic to visualize tree percentage in relation to spatially lagged tree percentage. We will then conduct Moran's I tests and run **Monte Carlo simulations**, which will allow us to further cluster states within Central Europe based on their tree percentage and spatial location.

Global Moran's I

The Global Moran's I can be calculated as follows:

$$I = \frac{n \sum_i \sum_j w_{ij} (Y_i - \bar{Y})(Y_j - \bar{Y})}{(\sum_{i \neq j} w_{ij}) \sum_i (Y_i - \bar{Y})^2},$$

In the formula n denotes number of regions, Y_i is the observed value of the variable of interest in region i and $\bar{Y}$ is the mean of all values. Spatial weights that denote the spatial proximity between regions i and j are w_{ij}, with $w_{ii} = 0$ and $i, j = 1, \ldots, n$.

The statistic tests how similar each region is to its neighbors. Under the null hypothesis of no spatial autocorrelation, observations Y_i are independent and identically distributed. Moran's I follows a normal distribution with the mean and variance:

$$E[I] = -\frac{1}{n-1}$$

$$Var[I] = \frac{n^2(n-1)S_1 - n(n-1)S_2 - 2S_0^2}{(n+1)(n-1)^2 S_0^2}$$

where:

- $S_0 = \sum_{i \neq j} w_{ij}$,
- $S_1 = \frac{1}{2}\sum_{i \neq j} \left(w_{ij} + w_{ji}\right)^2$,
- $S_2 = \sum_k \left(\sum_j w_{kj} + \sum_i w_{ik}\right)^2$.

Interpretation of Moran's I:

- **Positive Spatial Autocorrelation (Clustering):** Values significantly above the expected $E[I]$ indicate positive spatial autocorrelation, where neighboring regions tend to have similar values.
- **Negative Spatial Autocorrelation (Dispersion):** Values significantly below $E[I]$ indicate negative spatial autocorrelation, where neighboring regions tend to have dissimilar values.
- **No Spatial Autocorrelation (Randomness):** Values around $E[I]$ indicate no spatial pattern.

The test statistic for Moran's I is calculated as:

$$z = \frac{I - E(I)}{\sqrt{Var(I)}}$$

This z-score is compared to the standard normal distribution to assess significance.

Alternatively, Monte Carlo randomization can generate a distribution of Moran's I under random patterns. This method provides a randomization distribution for Moran's I by reassigning the observed values among regions and calculating the Moran's I for each of the patterns.

To test spatial autocorrelation, we follow these steps:

1. **Null Hypothesis** (H_0): $I = E[I]$ (no spatial autocorrelation).
2. **Alternative Hypothesis** (H_1): $I \neq E[I]$ (spatial autocorrelation).
3. **Test Statistic**: Calculate the z-score $z = \frac{I - E(I)}{\sqrt{Var(I)}}$.
4. **p-value**: Compare the z-score to the standard normal distribution or use Monte Carlo randomization to obtain a p-value.

Decision:

- If p-value < α (usually $\alpha = 0.05$), we reject the null hypothesis and conclude there is spatial autocorrelation.

- If p-value $\geq \alpha$, we fail to reject the null hypothesis and conclude there is no evidence of spatial autocorrelation.

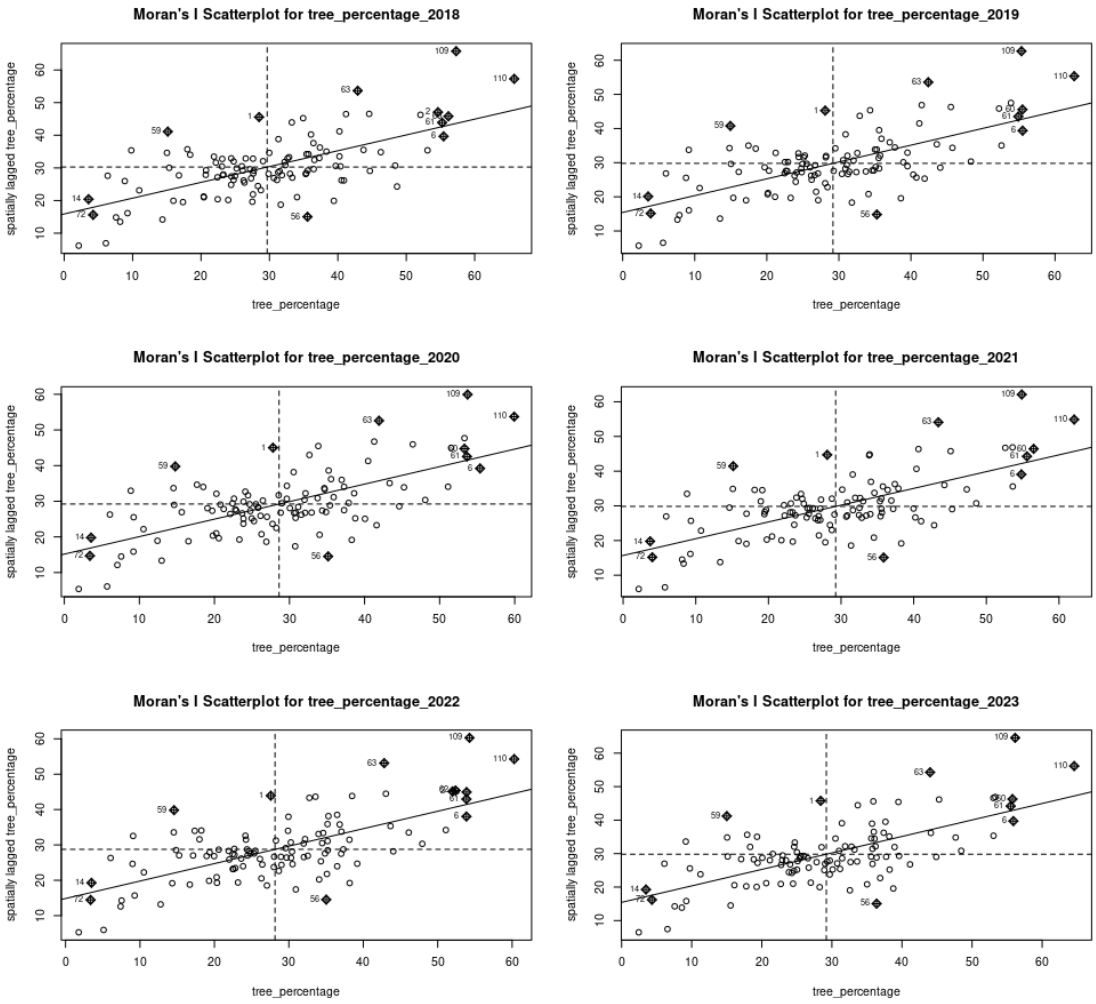
In order to gain first impressions about spatial autocorrelation of *tree_percentage* over the years, we will examine Moran's I scatterplots for each year. The *moran.plot()* function is used to create these scatterplots, which visualize the relationship between each observation and its spatially lagged values (i.e., the weighted average of its neighboring values). Using the *lag.listw()* function from the *spdep* package, we compute the spatial weights that help highlight this relationship and that are also used by other functions of *spdep* package which are going to be discussed in further work.

```
nb <- poly2nb(data_with_state_boarders$geometry, queen = TRUE) # 'queen' contiguity for neighbo
rs, meaning two neighbours share point of boarder
lw <- nb2listw(nb, style = "W")  # Spatial weights matrix, row-standardized
```

The Moran's Plots are divided into four quadrants:

- **Quadrant I (top-right)**: High values surrounded by high values (High-High cluster).
- **Quadrant II (top-left)**: Low values surrounded by high values (Low-High outliers).
- **Quadrant III (bottom-left)**: Low values surrounded by low values (Low-Low cluster).
- **Quadrant IV (bottom-right)**: High values surrounded by low values (High-Low outliers).

```
par(mfrow = c(3, 2))
for (year_col in year_columns) {
  tree_percentage <- data_with_state_boarders[[year_col]]
  moran.plot(tree_percentage, lw, main = paste("Moran's I Scatterplot for", year_col))
}
```



Moran's I Scatter Plot showing observations against lagged values

All scatterplots show a **positive linear relationship**, meaning that states with higher tree percentages tend to be surrounded by states with similarly high values (and vice versa for low values). This suggests **positive spatial autocorrelation**, indicating that similar values are clustered together rather than randomly distributed. **High-High clusters** (top-right quadrant) contain states with higher-than-average tree percentages surrounded by similarly high tree percentage neighbors. **Low-Low clusters** (bottom-left quadrant) contain states with lower-than-average tree percentages surrounded by neighbors with similarly low tree percentages. There are a few states that appear in the **top-left or bottom-right quadrants**, indicating potential **spatial outliers** (e.g., a high tree percentage surrounded by low tree percentage neighbors, or vice versa). These could be regions where local conditions differ significantly from their neighbors. Across all six plots, there is a consistent positive trend, meaning that the spatial autocorrelation of tree percentage remains stable over time. Each year shows similar clustering behavior, with relatively few outliers.

As we have gained an idea about spatial autocorrelation of our variable of interest, we aim to test our hypothesis of positive spatial autocorrelation. At the first step we are going to employ function `moran.test()` of the `spdep` package to test spatial autocorrelation using Moran's I. this function allows us to set argument called `alternative` which denotes alternative hypothesis and

can be set to *greater*, *less* or *two.sided*. For this test we are going to set *alternative="greater"* which specifies hypothesis as follows:

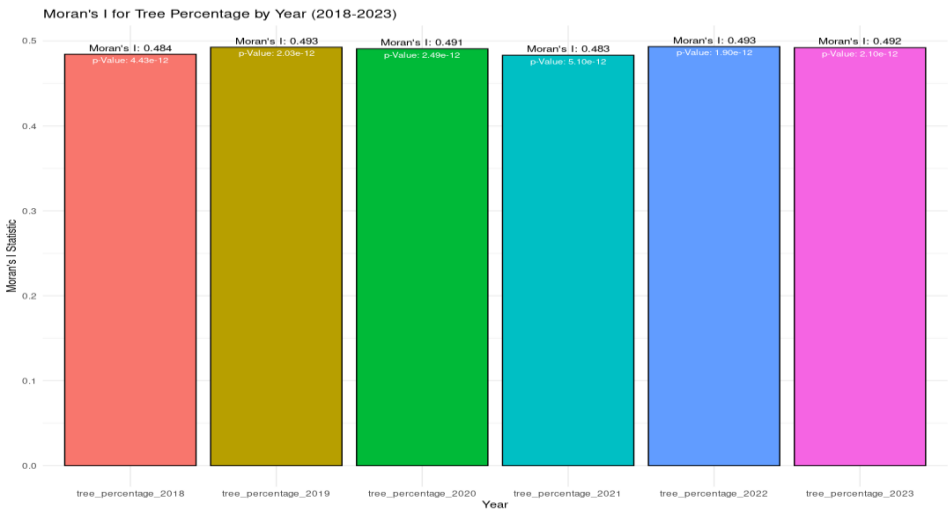
- $H_0: I \leq E[i]$ (negative spatial autocorrelation or no spatial autocorrelation)
- $H_1: I > E[i]$ (positive spatial autocorrelation)

As *moran.test()* doesn't differentiate between temporal observations, we are going to create a function that will calculate Moran's I test for each year of retrospective *tree_percentage*.

Calculating and storing results of Moran's I tests:

```
calculate_morans_i_for_year <- function(data, value_column, lw) {
    # Extract the tree percentage for the specific year
    tree_percentage <- data[[value_column]]
    morans_i <- moran.test(tree_percentage, lw)
    return(morans_i)
}

year_columns <- c("tree_percentage_2018", "tree_percentage_2019", "tree_percentage_2020",
                  "tree_percentage_2021", "tree_percentage_2022", "tree_percentage_2023")
for (year_col in year_columns) {
  morans_i_results[[year_col]] <- calculate_morans_i_for_year(
    data_with_state_boarders,
    value_column = year_col,
    lw = lw
  )
}
```

Next, we extract Moran's I statistic and corresponding p-values for each year to visualize the results and determine whether the null hypothesis (H_0) of no spatial autocorrelation can be rejected:



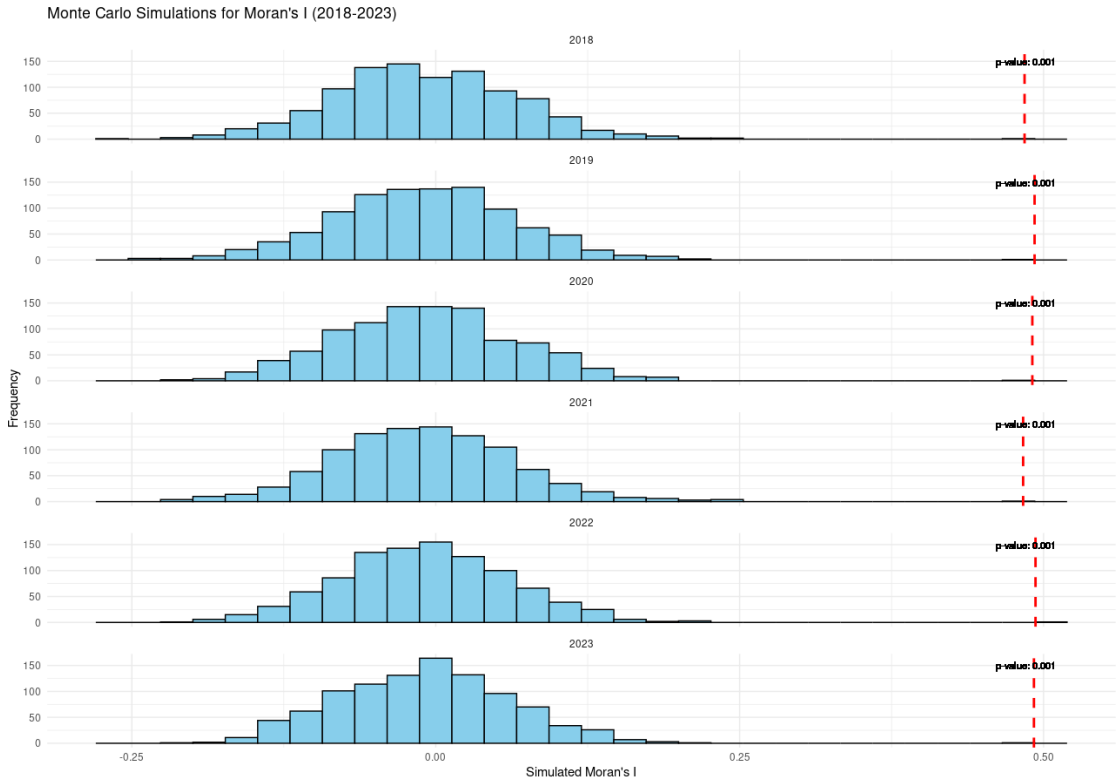
Moran's I Statistics and p-Values for tree percentage variables from 2018 to 2023

As seen in the computed values, the p-values for each year are below the 0.05 significance level, allowing us to reject the null hypothesis. This indicates strong evidence of positive spatial autocorrelation in the *tree_percentage* variable across all years from 2018 to 2023.

To further assess the significance, we can utilize a Monte Carlo approach. This method creates random patterns by reassigning the observed values among fixed areas and calculates Moran's I for each randomized pattern. The p-values are then computed as the proportion of simulated values that are as extreme or more extreme than the observed statistic, thereby providing a randomization-based significance test. The *spdep* package provides the *moran.mc()* function to perform this Monte Carlo simulation.

We are going to calculate Moran's I statistic using *moran.mc()* employing 999 simulations and extract the results into a data frame in order to make plots that reveal a pattern and let us decide on our hypothesis:

```
monte_carlo_results <- list()

for (year_col in year_columns) {
  tree_percentage <- data_with_state_boarders[[year_col]]
  monte_carlo_results[[year_col]] <- moran.mc(tree_percentage, lw, nsim = 999)
}
```



Histograms of Moran's I values for each simulated pattern in Monte Carlo randomization approach.

The red dashed line at the position of the observed Moran's I statistic for each year represents where the actual Moran's I falls in relation to the distribution of the simulated Moran's I values. Its purpose is to visually indicate how extreme the observed Moran's I is compared to the simulated values. If the observed Moran's I is located far in the tail of the simulated distribution, it suggests a significant result.

In our case, this holds true, as the Monte Carlo randomization approach reinforces the findings of the previously employed Moran's I test. The calculated *p*-values for each year are below the significance level of 0.05, which allows us to reject the null hypothesis. This provides strong

evidence that positive spatial autocorrelation is present in the `tree_percentage` variable for each year.

Local Moran's I and Local Indicators of Spatial Association (LISA)

The Global Moran's I provides a measure of spatial autocorrelation for the entire study area. However, it is often useful to analyze spatial clustering at a more localized level using *Local Indicators of Spatial Association (LISA)*. LISA measures the degree of spatial clustering around each observation. One key property of LISA is that the sum of the local Moran's I values across all regions is proportional to the global Moran's I. This allows us to break the global statistic into local components, offering more granular insights.

Widely used LISA metrics is the **local Moran's I**. For region i, the local Moran's I is defined as:

$$I_i = \frac{n(Y_i - \bar{Y}) \sum_j (Y_j - \bar{Y}) w_{ij}}{\sum_j w_{ij} \sum_i (Y_i - \bar{Y})^2}$$

where n is the number of regions, Y_i is the observed value for region i, $\bar{Y}$ is the mean value across all regions, w_{ij} are the spatial weights between regions i and j.

The global Moran's I is proportional to the sum of the local Moran's I values:

$$I = \frac{1}{\sum_i \sum_j w_{ij}} \sum_i I_i$$

The local Moran's I indicates whether a region is part of a cluster of similar values or if it is an outlier. A high value of I_i suggests that region i is surrounded by regions with similar values (high-high or low-low clusters), while a low value indicates the region is an outlier (high-low or low-high).

To interpret the local Moran's I for each region, we create a map of **p-values**. These p-values represent the probability of observing the given value under the null hypothesis of no spatial autocorrelation. The p-values are computed using a simulation-based approach, where the observed value in each region is fixed, and the remaining values are randomly shuffled. This process enables us to identify statistically significant clusters of spatial association or outliers.

"Greater Local Moran's I Test"

In this section, we calculate the **Local Moran's I** statistic for each year from 2018 to 2023 to explore the local spatial autocorrelation patterns for the variable `tree_percentage`. We use the `localmoran()` function from the `spdep` package, which computes Local Moran's I based on the spatial weights for the data. we again set alternative hypothesis to "greater", where:

- H_0: No or negative spatial autocorrelation.
- H_1: Positive autocorrelation.

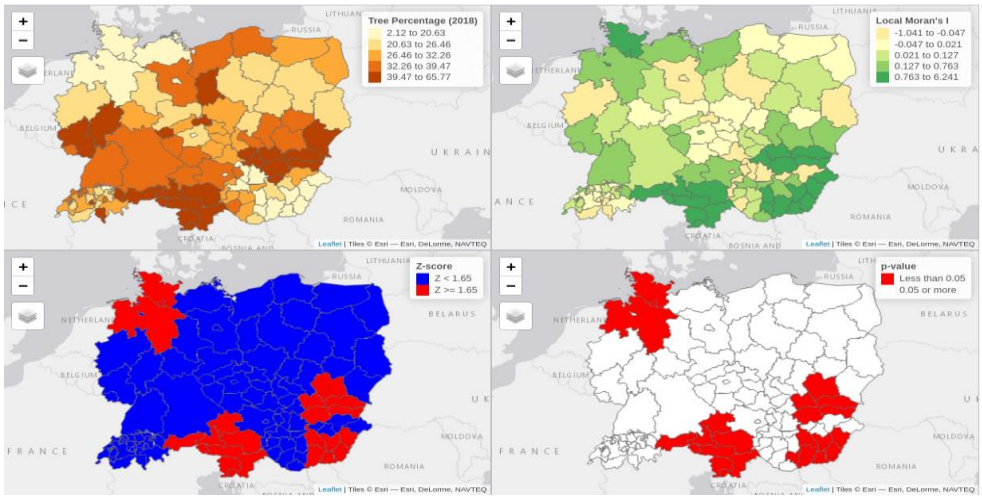
The function `localmoran()` returns:

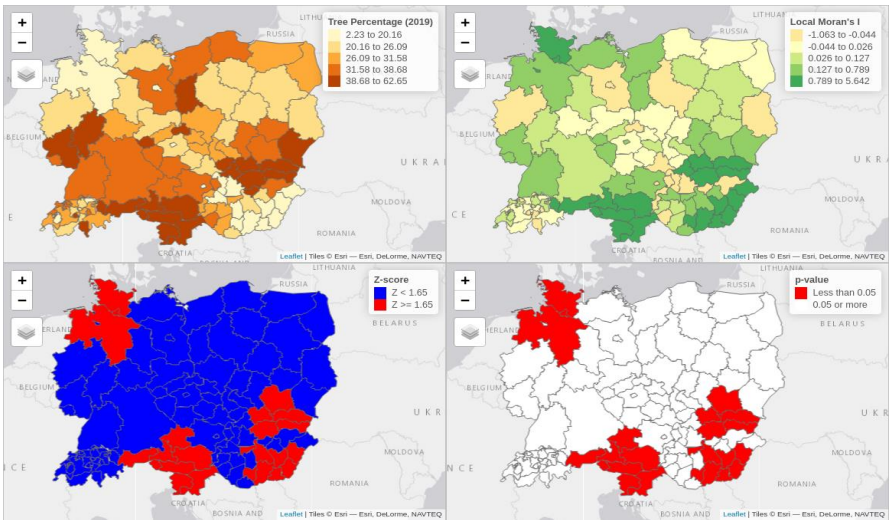
- I_i: Local Moran's I statistic for each area.
- $E[I_i]$: Expected value under H_0.

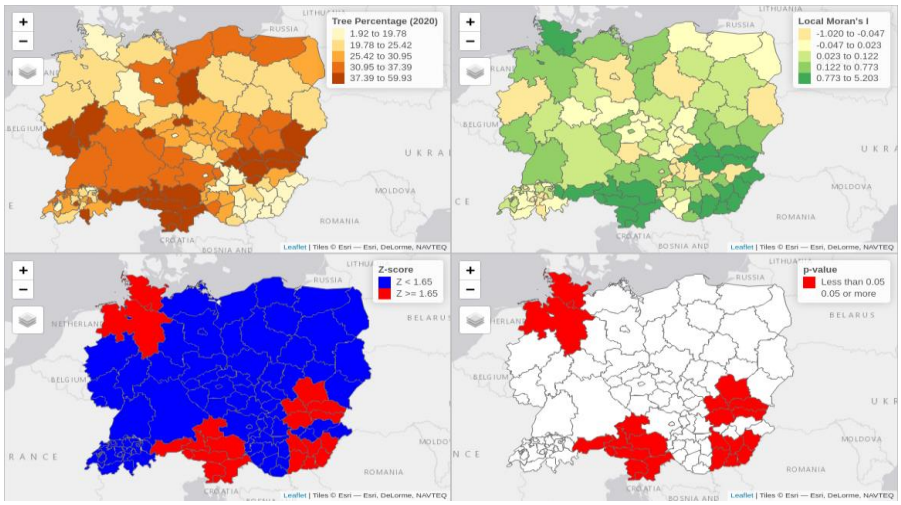
- $Var[I_i]$: Variance of Local Moran's I.
- $Z[I_i]$: Z-Scores to access significance.
- $Pr(z > E[I_i])$: p-values indicating the significance of the spatial autocorrelation.

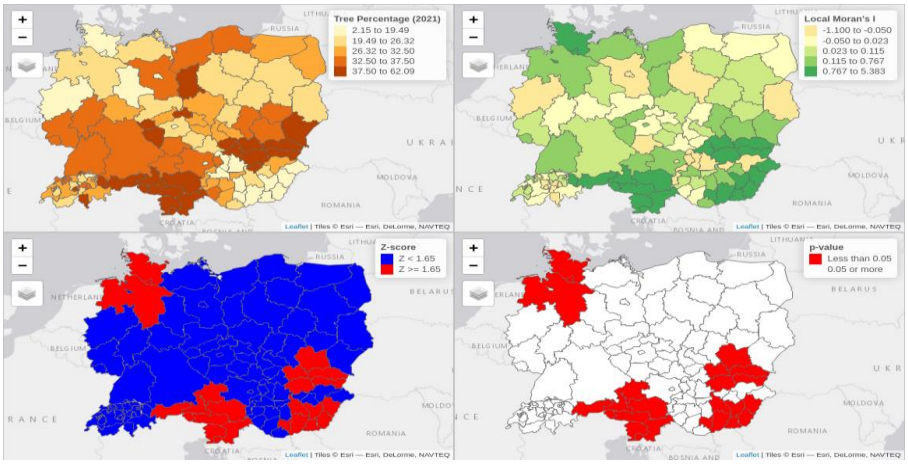
```
calculate_local_morans_i_for_year <- function(data, value_column, lw) {
  tree_percentage <- data[[value_column]]

  # Calculate Local Moran's I with alternative hypothesis "greater"
  lmoran <- localmoran(tree_percentage, lw, alternative = "greater")

  return(data.frame(
    Ii = lmoran[, "Ii"],                # Local Moran's I statistic
    EIi = lmoran[, "E.Ii"],             # Expected value of Ii
    VarIi = lmoran[, "Var.Ii"],         # Variance of Ii
    ZIi = lmoran[, "Z.Ii"],             # Z-scores
    p_value = lmoran[, "Pr(z > E(Ii))"]  # P-values for greater test
  ))
}

local_morans_i_results <- list()
for (year_col in year_columns) {
  local_morans_i_results[[year_col]] <- calculate_local_morans_i_for_year(
    data_with_state_boarders,
    value_column = year_col,
    lw = lw
  )
}
```

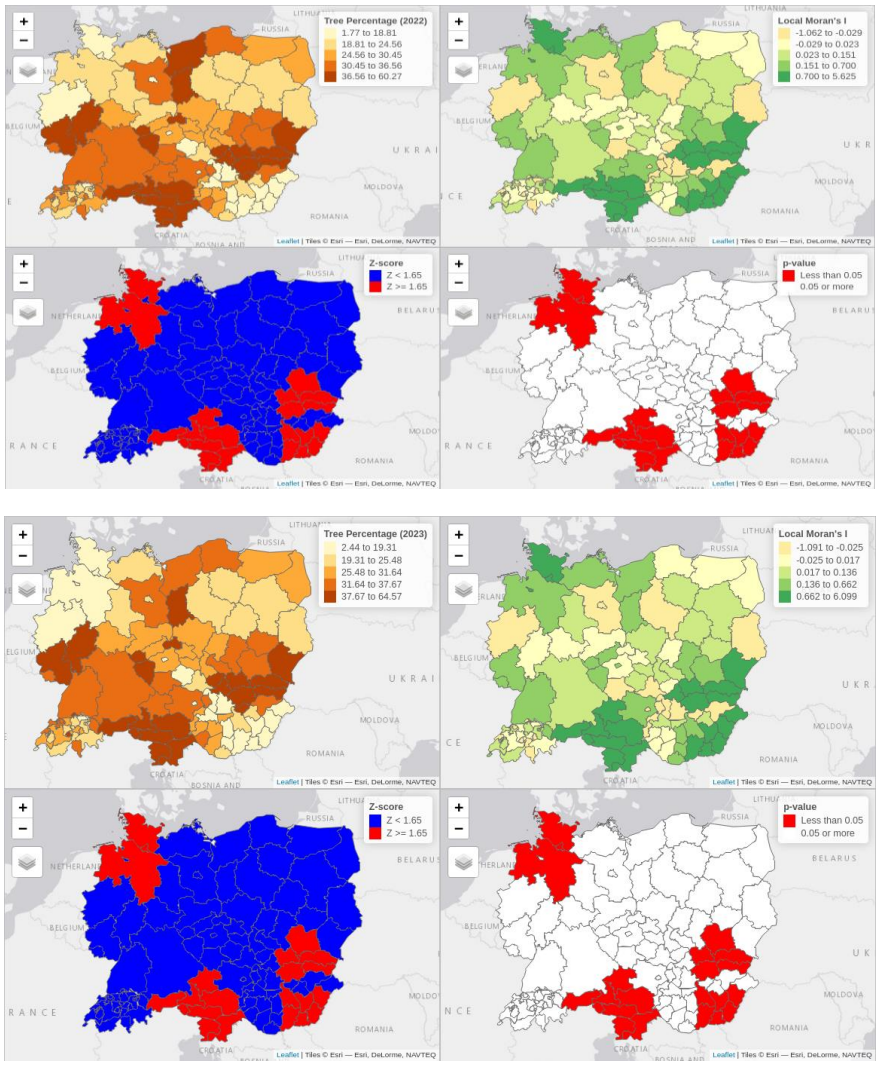
We will visualize results for each year using `tmap` package. Areas in which p-value is less than the significance level 0.05 (in other words areas where z-scores are higher than `qnorm(0.95) = 1.65`) are the ones for which we reject null hypothesis H_0 and conclude they present positive spatial autocorrelation. We will visualize Tree percentage of each year and Local Moran's I statistic as well, to also observe how these changed over time.




Tree Percentage (2019)
2.23 to 20.16
20.16 to 26.09
26.09 to 31.58
31.58 to 38.68
38.68 to 62.65
Local Moran's I
-1.063 to -0.044
-0.044 to 0.026
0.026 to 0.127
0.127 to 0.789
0.789 to 5.642
Z-score
Z < 1.65
Z >= 1.65
p-value
Less than 0.05
0.05 or more
Leaflet | Tiles © Esri — Esri, DeLorme, NAVTEQ


Tree Percentage (2020)
1.92 to 19.78
19.78 to 25.42
25.42 to 30.95
30.95 to 37.39
37.39 to 59.93
Local Moran's I
-1.020 to -0.047
-0.047 to 0.023
0.023 to 0.122
0.122 to 0.773
0.773 to 5.203
Z-score
Z < 1.65
Z >= 1.65
p-value
Less than 0.05
0.05 or more
Leaflet | Tiles © Esri — Esri, DeLorme, NAVTEQ


Tree Percentage (2021)
2.15 to 19.49
19.49 to 26.32
26.32 to 32.50
32.50 to 37.50
37.50 to 62.09
Local Moran's I
-1.100 to -0.050
-0.050 to 0.023
0.023 to 0.115
0.115 to 0.767
0.767 to 5.383
Z-score
Z < 1.65
Z >= 1.65
p-value
Less than 0.05
0.05 or more
Leaflet | Tiles © Esri — Esri, DeLorme, NAVTEQ



As the *tree_percentage* variable shows, there are slight changes in the tree percentage for each area over the years. Similarly, the Moran's I statistics calculated for each year also reflect these differences. The plots for each year illustrate areas with positive spatial autocorrelation, with some neighboring states joining a cluster of positive spatial autocorrelation one year and then no longer being part of that group the next year.

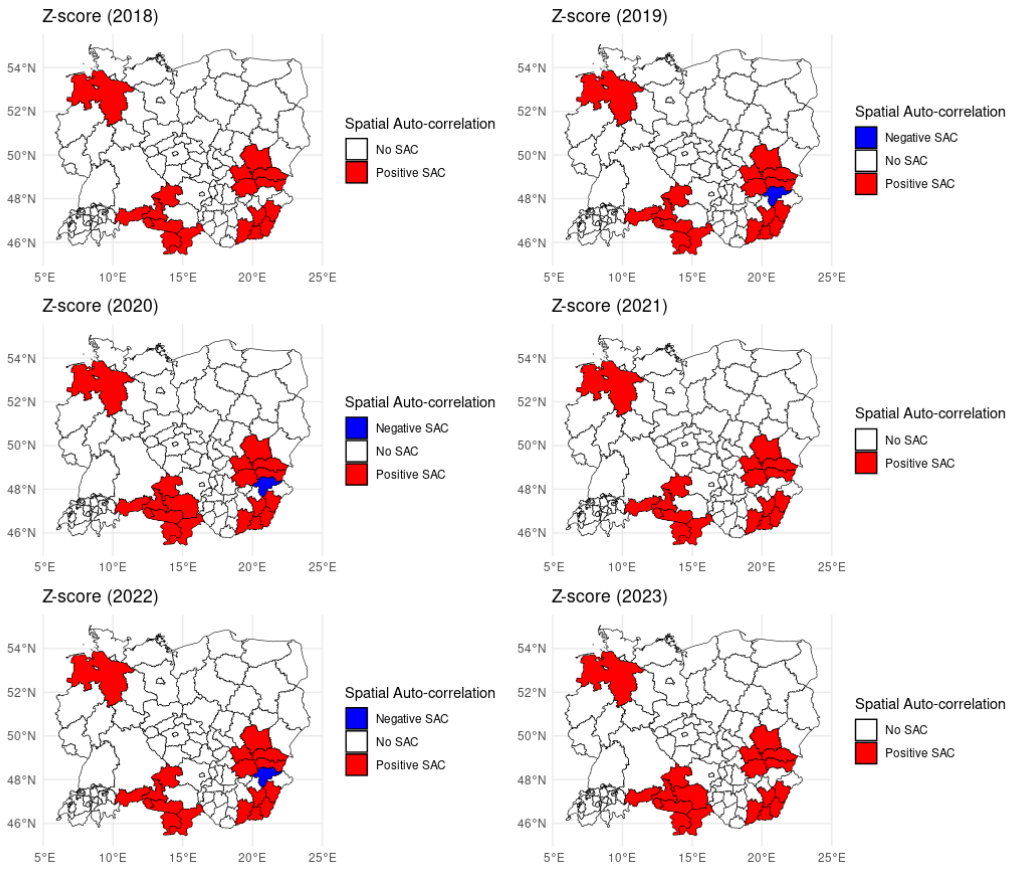
In Hungary, states with positive spatial autocorrelation consistently show very low tree percentages (between 2% and 10%). This makes sense, as this region is largely covered by fertile arable land (as indicated by the crops in our dataset). The significance value is below 0.05 for southern Austrian states and the entirety of Slovenia, where tree percentages remain high compared to neighboring areas. The Slovenian states of Zahodna and Vzhodna, with tree percentages rising between 55% and 65%, contrast with Austrian states, where tree percentages are between 40% and 55%. A few Slovakian states, such as Zilina and Banska Bystrica, exhibit similar patterns to Austria and Slovenia, with high tree coverage (~55%) throughout the observed years. The observed positive spatial autocorrelation in these groups makes sence, as these regions in Austria, Slovenia, and Slovakia are mountainous and naturally have higher tree percentages.

In contrast, the northeastern territories of Germany show behavior similar to Hungary, with lower tree percentages compared to their neighboring states and regions geographically close to them. Those states are cities with higher built area density where it's expected to see lower tree percentage. Other states, however, do not exhibit a clear pattern, likely because their tree percentages are influenced by various factors beyond just geographical ones.

Two sided Local Moran's Test

Previous Local Moran's I statistic was calculated using hypothesis alternative "greater", but if we set the hypothesis alternative to *two.sided* this might reveal new patterns in the data, as H_0 would be: no spatial autocorrelation vs. H_1: positive or negative spatial autocorrelation. In the two-sided test, z-score values lower than -1.96 indicate negative spatial autocorrelation and those greater than 1.96 positive one. We are going to conduct this test in order to break down the H_0 hypothesis from the previous section with "greater" hypothesis alternative (which was no or negative spatial autocorrelation) to see if any states for any of the captured years show negative spatial auto correlation.

```r
calculate_local_morans_i_for_year_two_sided <- function(data,
                                                        value_column,
                                                        lw) {

    tree_percentage <- data[[value_column]]

  # Calculate Local Moran's I two sided test
  lmoran <- localmoran(tree_percentage, lw, alternative = "two.sided")

  return(data.frame(
    Ii = lmoran[, "Ii"],                  # Local Moran's I statistic
    EIi = lmoran[, "E.Ii"],               # Expected value of Ii
    VarIi = lmoran[, "Var.Ii"],           # Variance of Ii
    ZIi = lmoran[, "Z.Ii"],               # Z-scores
    p_value = lmoran[, "Pr(z != E(Ii))"]  # P-values for two.sided test
  ))
}
```



Areas with negative, no and positive spatial autocorrelation conducted from two-sided Local Moran's I Statistic

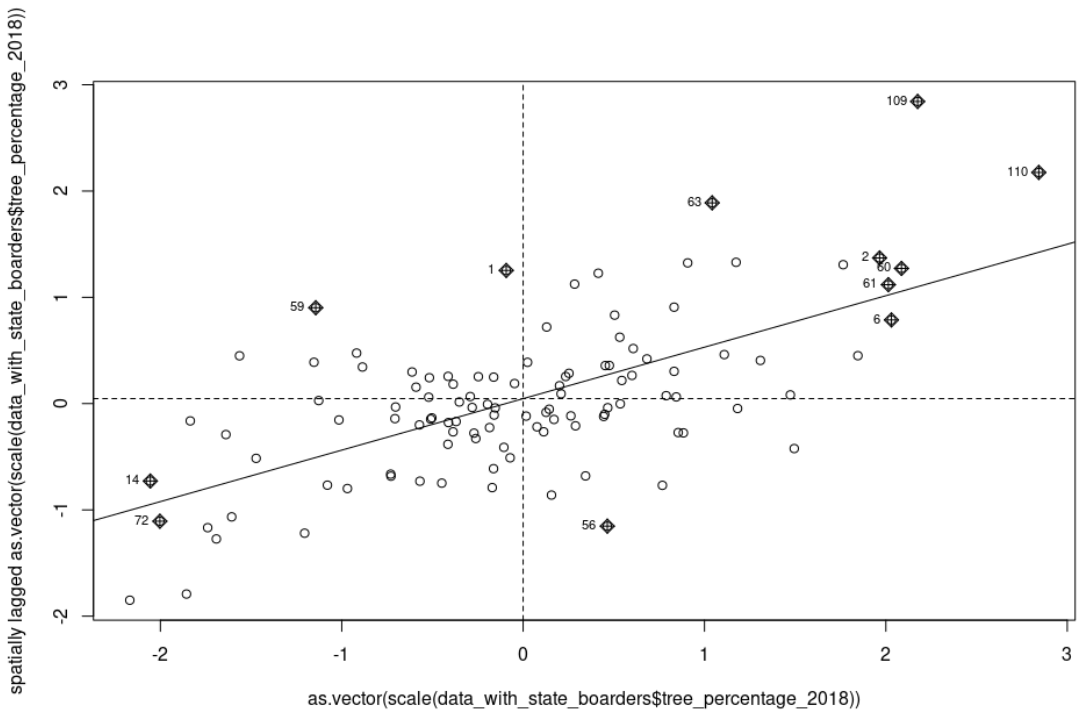
The groups with positive spatial autocorrelation under the `"two-sided"` alternative hypothesis revealed the same patterns as those obtained from the one-sided `"greater"` alternative hypothesis. From 2019 to 2022 (excluding 2021), negative spatial autocorrelation was observed in the state of Borsod-Abaúj-Zemplén in Hungary. Negative spatial autocorrelation means that surrounding areas have dissimilar values. This state is bordered on one side by Slovakian states with high tree coverage and on the other side by Hungarian states with the lowest tree percentage values in the dataset.

Moran's I Clusters

As we have seen in the section about Moran's plots, they divide information into four quadrants. Based on Moran's I statistics and information provided through the plot, we are able to separate our data points into four clusters: **Low-Low**, **High-High**, **High-Low** and **Low-High**. To detect clusters, we first have to run `localmoran()` function as we did in previous step, we are again going to access the `"two.sided"` alternative hypothesis and to obtain clusters of each type we are going going to use information provided by the Moran's I scatterplot given by previously mentioned function `moran.plot()`.

We are going to run `moran.plot()` for tree percentage varioable for only one year to see now the resulting data frame looks like for it.

```
mp <- moran.plot(as.vector(scale(data_with_state_boarders$tree_percentage_2018)), lw)
head(mp)
```



Moran's I Scatterplot showing the scaled values against spatially lagged values for tree percentage in 2018

	x	wx	is_inf	labels	dfb.1_	dfb.x	dffit	cov.r	cook.d	hat
1	-0.09309701	1.2529849	TRUE	1	0.21507713	-0.020114677	0.21601568	0.9377539	0.0224897420	0.009170423
2	1.96533880	1.3707091	TRUE	2	0.06491272	0.128159359	0.14366100	1.0574846	0.0103727508	0.044527208
3	0.47535268	0.3584624	FALSE	3	0.01369774	0.006541056	0.01517937	1.0298880	0.0001162612	0.011163938
4	0.41441512	1.2267081	FALSE	4	0.16704307	0.069541995	0.18094053	0.9737113	0.0160667556	0.010666504
5	1.17415442	1.3294317	FALSE	5	0.12236194	0.144329355	0.18921788	1.0107501	0.0178009693	0.021738970
6	2.03061711	0.7872328	TRUE	6	-0.04251125	-0.086719151	-0.09657856	1.0651581	0.0046989743	0.046920321

We identify cluster types using the quadrants of the scaled values *(mp$x)* and their spatially lagged values *(mp$wx)*, along with the p-values from the local Moran's I for each area

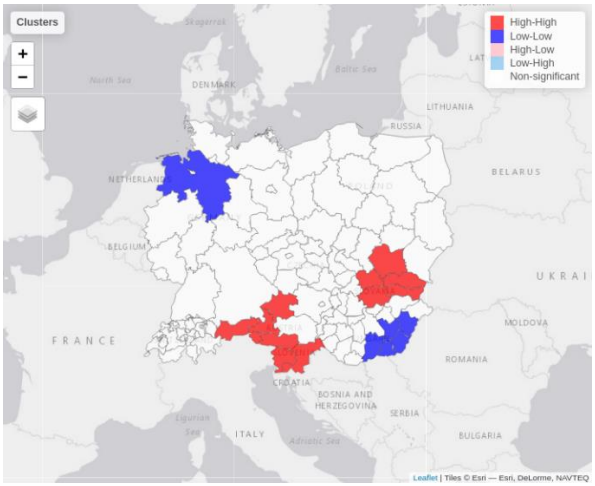
(data_with_state_boarders$p_value_ts_year_column). The clusters are classified as follows: areas with a significant local Moran's I are classified as *high-high* if both the value and its corresponding spatially lagged value are positive, *low-low* if both are negative, *high-low* if the value is positive but the spatially lagged value is negative, and *low-high* if the value is negative but the spatially lagged value is positive.

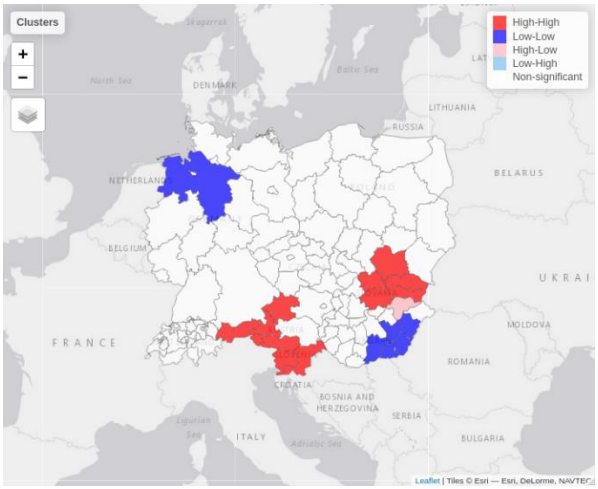
To represent these cluster types, we create a *quadrant* variable for each area, based on its value, spatially lagged value, and p-value. Specifically, quadrants 1, 2, 3, and 4 correspond to *high-high*, *low-low*, *high-low*, and *low-high* clusters, respectively, while quadrant 5 represents non-significant areas.

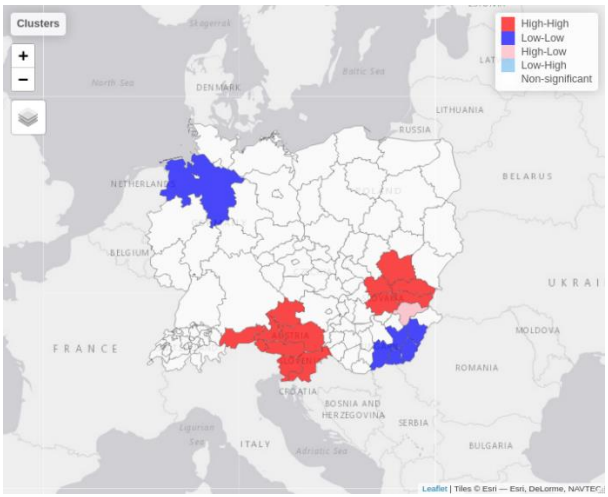
```
data_with_state_boarders$quadrant_2018 <- NA

# high-high
data_with_state_boarders[(mp$x >= 0 & mp$wx >= 0) &
```

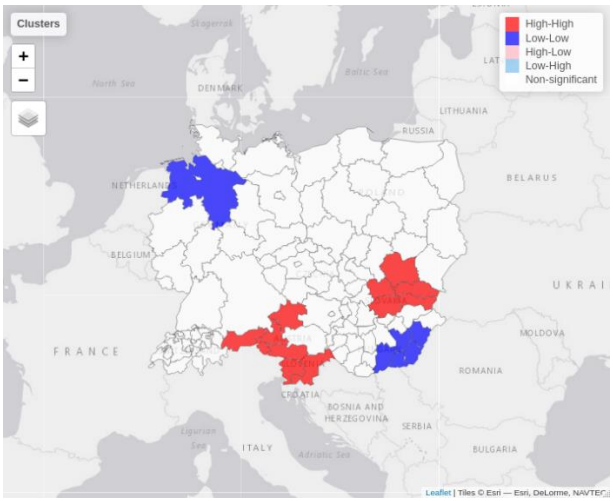
```
    (data_with_state_boarders$p_value_ts_tree_percentage_2018 <= 0.05), "quadrant_2018"] <- 1

# low-low
data_with_state_boarders[(mp$x <= 0 & mp$wx <= 0) &
    (data_with_state_boarders$p_value_ts_tree_percentage_2018 <= 0.05), "quadrant_2018"] <- 2

# high-low
data_with_state_boarders[(mp$x >= 0 & mp$wx <= 0) &
    (data_with_state_boarders$p_value_ts_tree_percentage_2018 <= 0.05), "quadrant_2018"] <- 3

# low-high
data_with_state_boarders[(mp$x <= 0 & mp$wx >= 0) &
    (data_with_state_boarders$p_value_ts_tree_percentage_2018 <= 0.05), "quadrant_2018"] <- 4

# non-significant
data_with_state_boarders[(data_with_state_boarders$p_value_ts_tree_percentage_2018 > 0.05),
                         "quadrant_2018"] <- 5
```

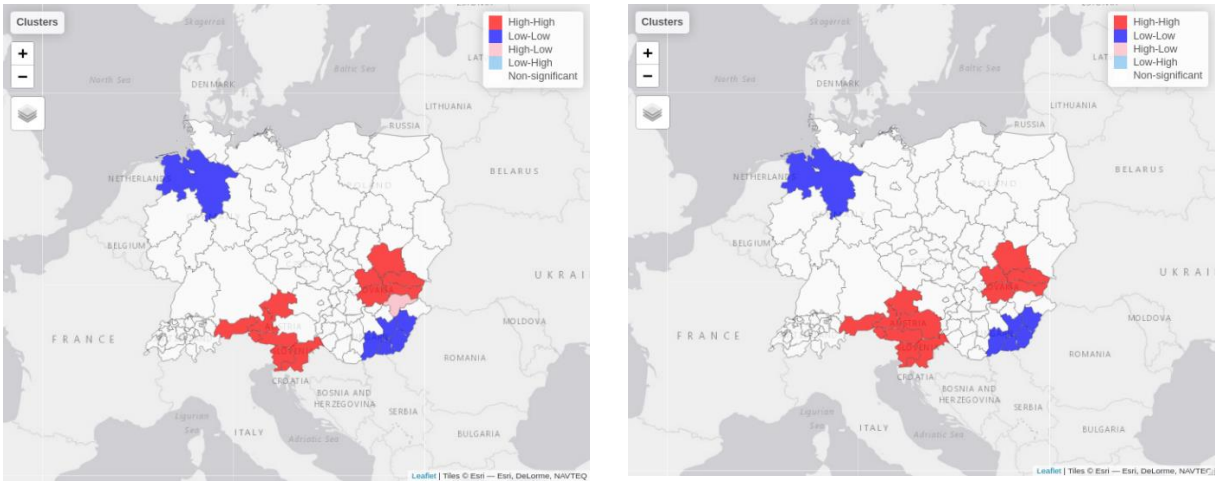
We will repeat this process for each year and we than get corresponding plots:











The plots reveal similar patterns to those observed in the previous section, where we used the `Localmoran()` test to assess positive, negative, and no spatial autocorrelation. This time, we can see the specific reasons why each region exhibits positive or negative spatial autocorrelation. The red areas in the plots represent regions with positive spatial autocorrelation, where high-value areas are surrounded by other high-value areas. These include regions in southern Austria, Slovenia, and Slovakia which all experience high tree coverage throughout the years.

The blue areas also indicate positive spatial autocorrelation, but in this case, they represent regions where states with low tree percentages are surrounded by other low-value states. As observed previously, these regions are located in Hungary and northern Germany.

The only state with negative spatial autocorrelation, as identified in the previous section, is Borsod-Abaúj-Zemplén in Hungary, which showed negative spatial autocorrelation in 2019, 2020, and 2022. Now, we see that this is due to its *High-Low* classification, meaning that the tree percentage in the state itself is high, but its spatially lagged value is low, as it is surrounded by neighboring Hungarian states with significantly lower tree percentages. This prevents it from being clustered with those lower-value states. Similarly, it cannot be grouped with the Slovakian territories to its north, which have much higher tree percentages (around 50%) compared to the 30% observed in Borsod-Abaúj-Zemplén.

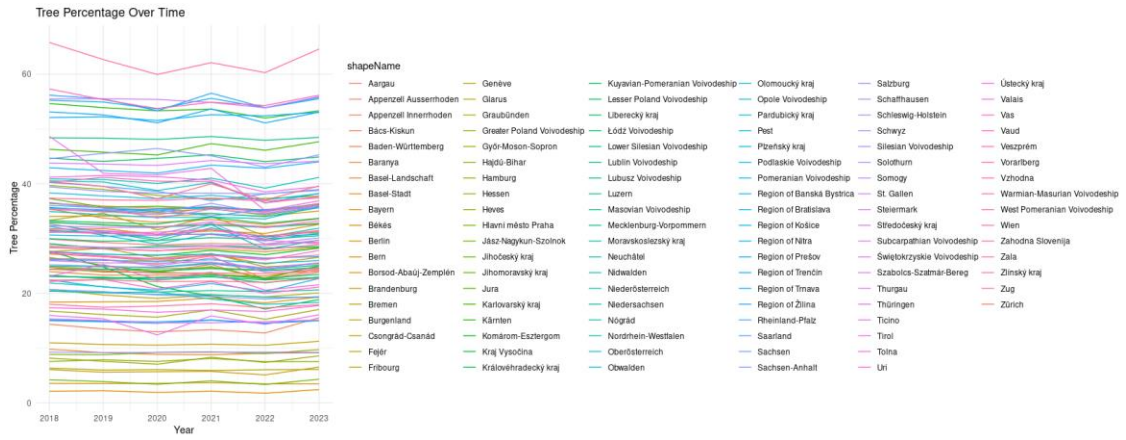
Temporal autocorrelation

As we have seen in previous section, some regions of Central Europe experience strong spatial autocorrelation, now we want to also analyze the temporal autocorrelation of tree percentage to understand the stability and consistency of tree cover across different regions over time. **Temporal autocorrelation** refers to the relationship between a variable's values at different time points—in this case, the tree percentage from one year to the next.

By examining how the tree percentage in one year is correlated with the tree percentage in the previous year, we can assess whether regions with high (or low) tree cover maintain their levels over time. This helps to identify trends and patterns in tree cover stability, as well as detect regions that may experience significant changes between consecutive years.

In order to work with this data as time series, we first have to convert it from wide format to long format, so that we can access simple visualization techniques:

```
time_series_data <- data_with_state_boarders %>%
  select(shapeName, starts_with("water_percentage"), starts_with("tree_percentage"),
         starts_with("flooded_vegetation_percentage"), starts_with("crops_percentage"),
         starts_with("built_area_percentage"), starts_with("bare_ground_percentage"),
         starts_with("snow_ice_percentage"), starts_with("clouds_percentage"),
         starts_with("rangeland_percentage"), starts_with("burned_area"),
         starts_with("CO2_total"), starts_with("PM25_total"),
         starts_with("TPC_total"), starts_with("NMHC_total"),
         starts_with("OC_total"), starts_with("CH4_total"),
         starts_with("SO2_total"), starts_with("BC_total")) %>%
  pivot_longer(cols = -shapeName,
               names_to = "parameter",
               values_to = "value") %>%
  mutate(year = as.integer(str_extract(parameter, "\\d{4}")),
         parameter = str_remove(parameter, "_\\d{4}")) %>%
  pivot_wider(names_from = parameter, values_from = value) %>%
  group_by(shapeName, year) %>%
  summarize(across(everything(), mean, na.rm = TRUE), .groups = 'drop')


time_series_data <- data_with_state_boarders %>%
  select(shapeName, shapeGroup, total_area_km2) %>%
  left_join(time_series_data, by = "shapeName")
```



Tree Percentage changes in central Europe over time

Just by looking at the time series data of *tree_percentage* in our data set, we get an idea of its consistency over time and slight changes that each state experienced, but to understand the dynamics of tree percentage over time, we calculated the correlation between the tree percentage values in one year and the corresponding values from the previous year (i.e., the **lagged values**). This approach helps quantify how much a region's tree cover in a given year is influenced by its tree cover in the preceding year. The rationale behind this is that if a region's tree percentage is strongly correlated with its value from the previous year, it implies a level of **stability** or **persistence** in tree coverage. Conversely, a weak or negative correlation would suggest potential **fluctuations** or **changes** in forestation.

The correlation between the tree percentage values and their lagged values is computed using **Pearson correlation**. Pearson correlation quantifies the linear relationship between two

variables—in this case, the tree percentage for a given year and the tree percentage from the previous year (lagged values). The formula for Pearson correlation between two variables X (tree percentage) and Y (lagged tree percentage) is:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}}$$

Where:

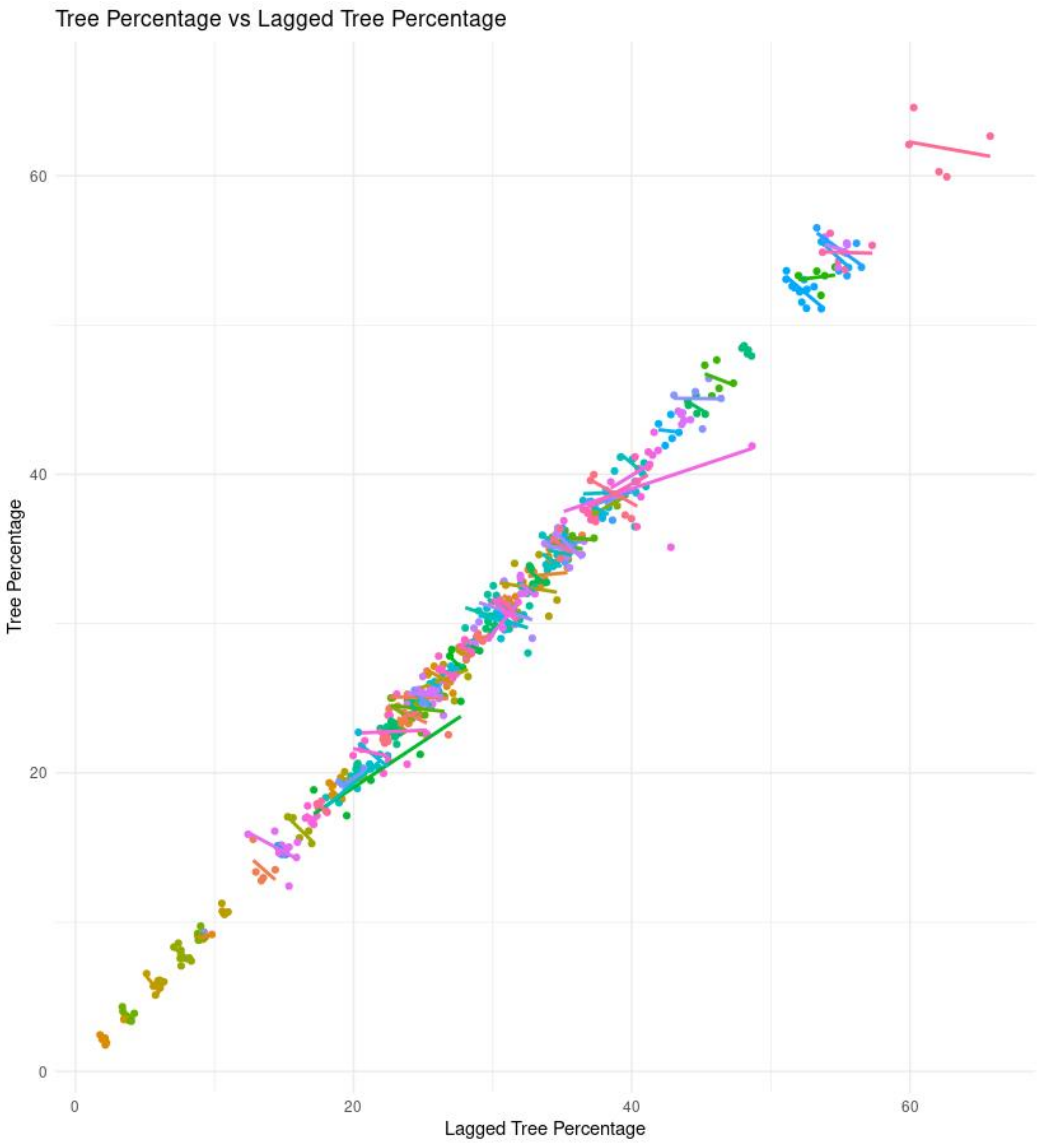
- r is the Pearson correlation coefficient, ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation).
- X_i and Y_i are the individual data points of tree percentage and lagged tree percentage, respectively.
- $\bar{X}$ and $\bar{Y}$ are the mean values of tree percentage and lagged tree percentage.

A value of $r \approx 1$ indicates a strong positive correlation, meaning that regions with high tree percentages in one year are likely to have high tree percentages in the following year and vice versa for regions with low tree percentage. A value of $r \approx -1$ suggests an inverse relationship, where regions with high tree percentages in one year have low tree percentages in the following year, while $r \approx 0$ suggests no relationship between the values.

The process:

1. We introduced a lag by creating a new variable representing the tree percentage of the previous year. This lagging operation shifts the data by one time unit (year) for each region.
2. For each region, the Pearson correlation between the tree percentage values in a given year and the values from the previous year (lagged values) was computed. This process captures the extent to which the tree percentage in year t is dependent on the tree percentage in year $t - 1$.
3. A high positive correlation for a region implies that the tree coverage in that region indicates relatively same patterns in tree cover growth/loss over time. A low or negative correlation suggests that the region may experience more dynamic changes in tree cover from one year to the next - either tree cover growth or loss without clear pattern.

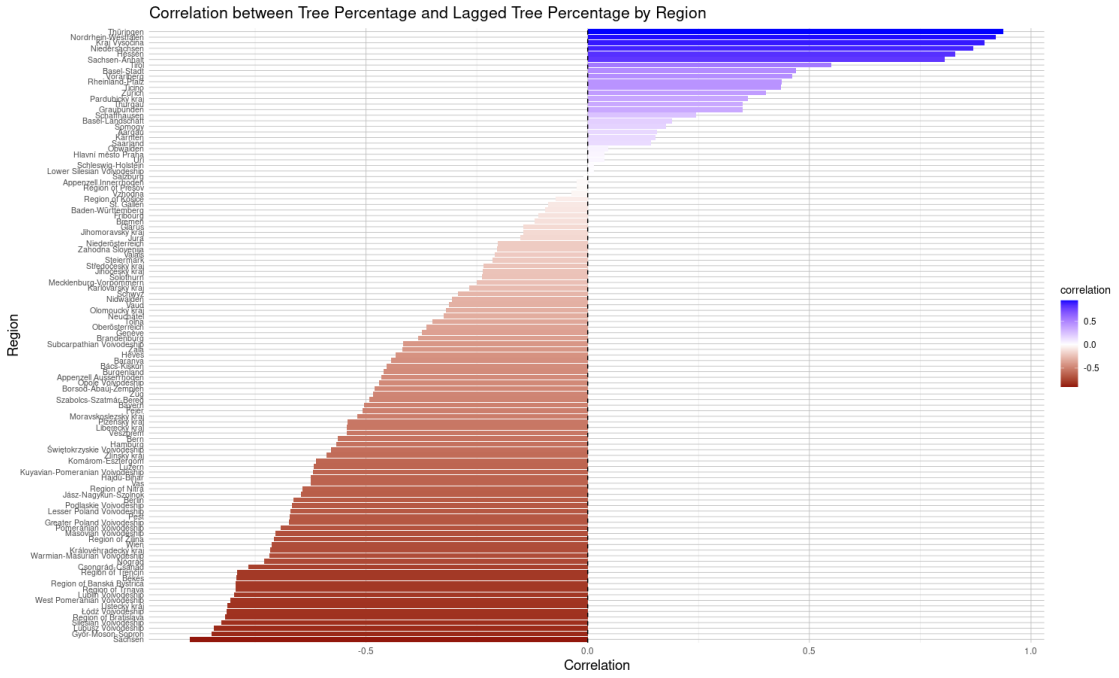
```
time_series_data <- time_series_data %>%
  arrange(shapeName, year)  # Arrange data by region and year

# Create lagged variable for tree percentage
time_series_data <- time_series_data %>%
  group_by(shapeName) %>%
  mutate(lag_tree_percentage = lag(tree_percentage))

# Compute correlation between tree_percentage and lag_tree_percentage
correlation_results <- time_series_data %>%
  summarise(correlation = cor(tree_percentage, lag_tree_percentage, use = "complete.obs"))
```



Scatter plot showing the relationship between tree percentage and its lagged tree percentage for each region. Each point represents a region's tree percentage in one year versus its percentage in the previous year. The color represents different regions, and the trend lines show the linear relationship for each region.

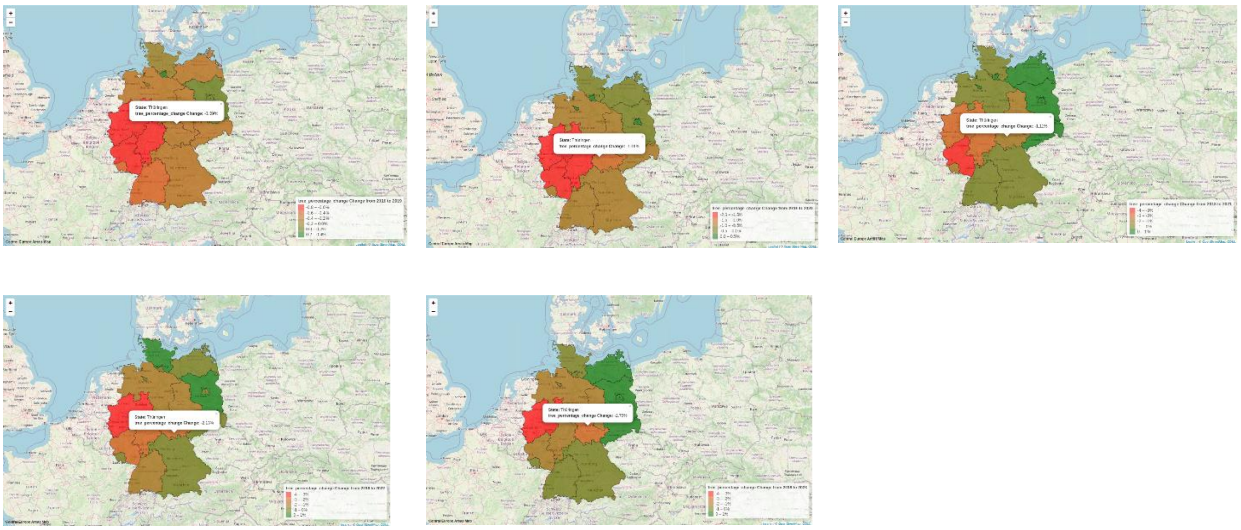
As we can see on the plot, there is a clear **positive linear relationship** between tree percentage and the lagged tree percentage. This suggests that if a region had a high tree percentage in one year, it likely had a similarly high percentage in the previous year (and vice versa for low tree percentages). The points follow a nearly perfect diagonal line, indicating that the tree percentage in any given year is highly correlated with the tree percentage in the previous year. This means that regions tend to have **consistent tree cover over time**. The small deviations from the diagonal line indicate slight year-to-year fluctuations in tree percentages. However, these fluctuations appear to be minor, reinforcing the idea that tree cover tends to be stable over time for most regions. There are a few regions where the points are slightly farther from the diagonal line, indicating that these regions experienced more significant changes in tree percentage between consecutive years.

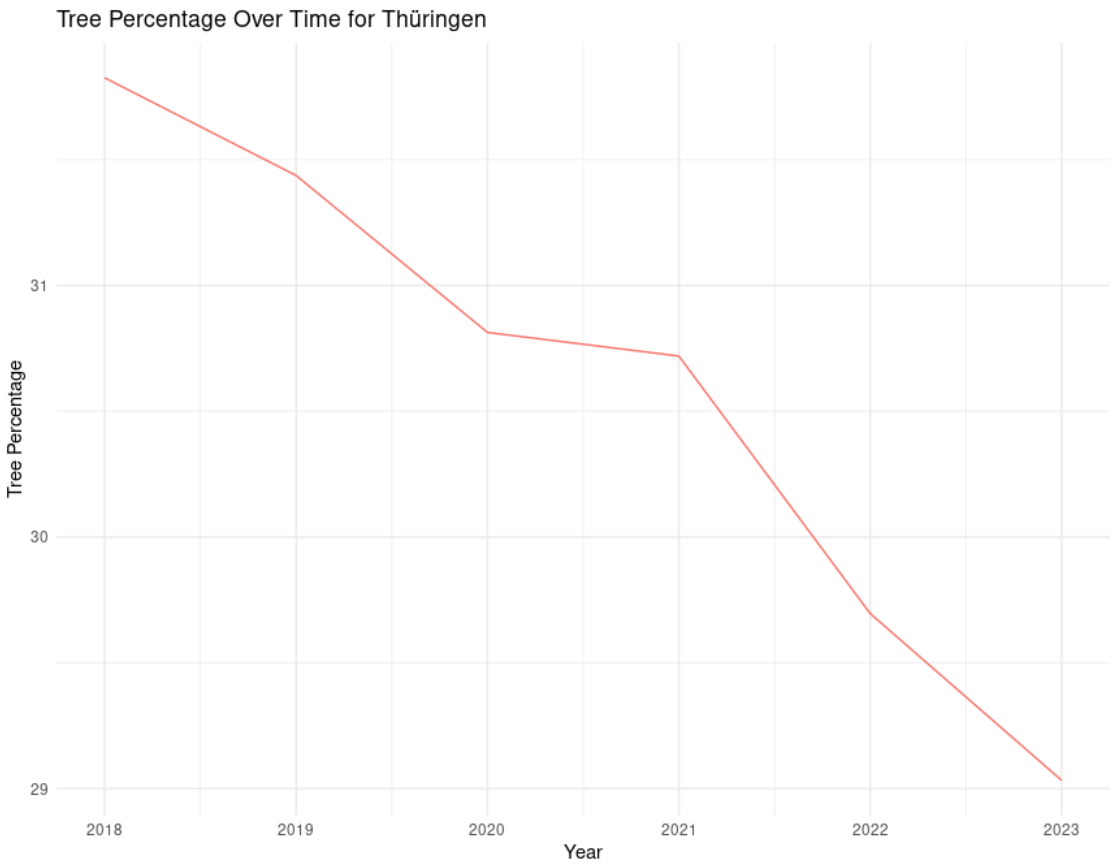
To further investigate the trend of this strong linear relationship, we will closely examine each region to determine whether they exhibit a positive or negative correlation.



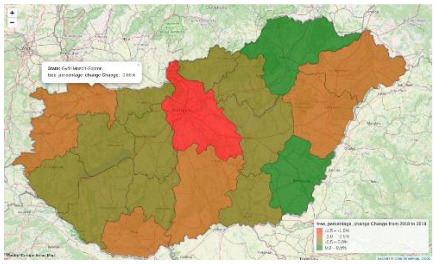
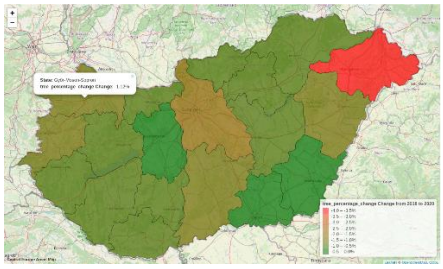
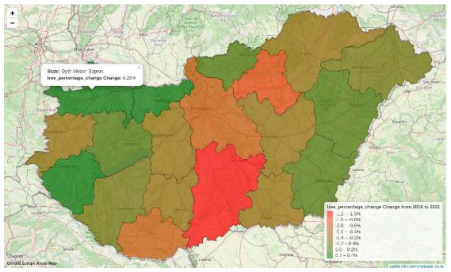
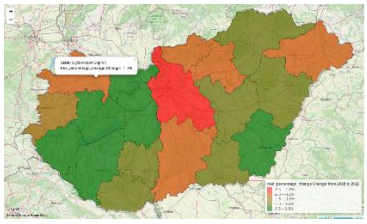
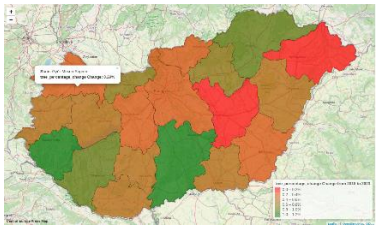
Correlation between tree percentage and its lagged values for each region.

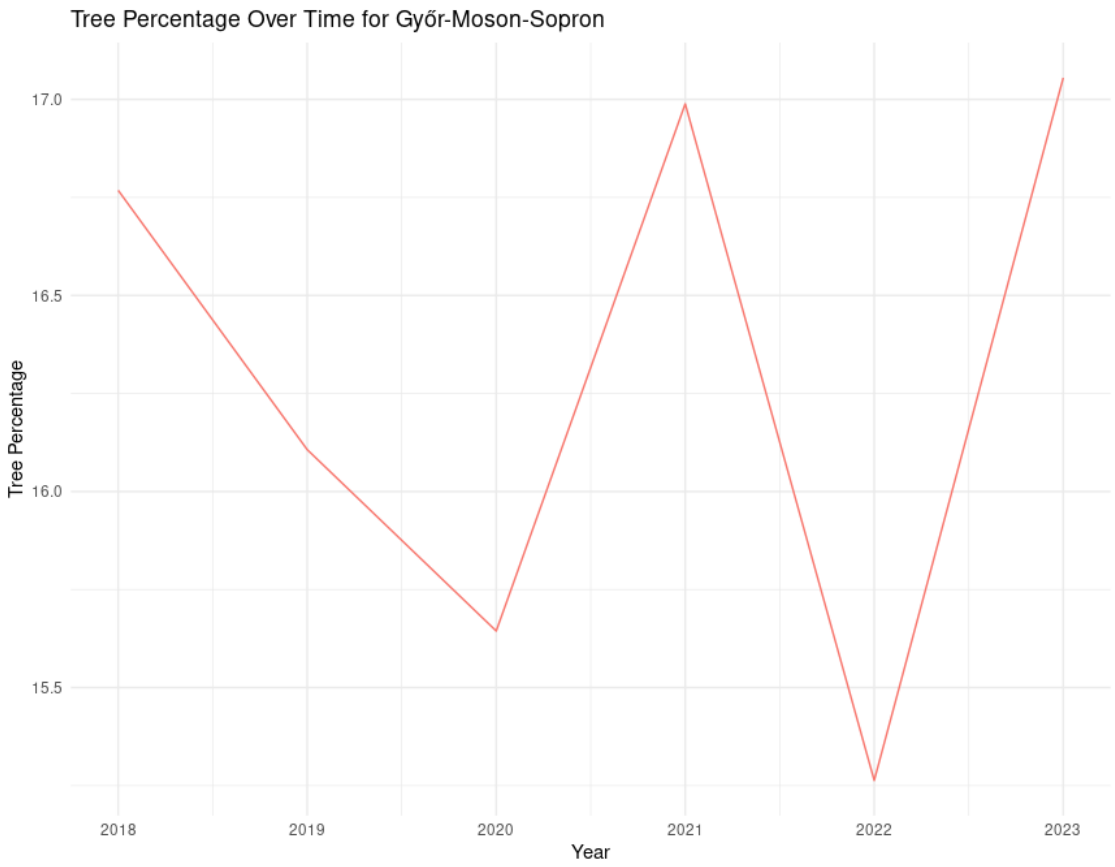
The regions with blue bars represent a **positive correlation** between tree percentage and its lagged value. This means that these regions tend to stay stable in their tree cover growth/loss over time. A prime example of this is **Thüringen** in Germany, where the tree percentage change has shown stableness year after year. Following plots from the Shiny-App illustrates tree change behaviour of this state:





On the other hand, the red bars represent regions with a **negative correlation**, indicating that the tree percentage has been experiencing inconsistent changes over time, whether it was a tree growth or tree loss. Region of Gyor in Hungary has shown such strong behaviour in experiencing such different patterns over the years as shown in plots below:



Most regions fall within the correlation range of **-0.5 to 0.5**, suggesting that, for these regions, changes in tree percentage from year to year are not as pronounced. This indicates that tree cover remains relatively stable over time for many areas. This stability is further supported by the earlier scatterplot, where tree percentages showed a strong linear relationship with their lagged values, confirming that significant fluctuations are not common across the dataset, although they can happen, but when that is the case they are not so rigorous.

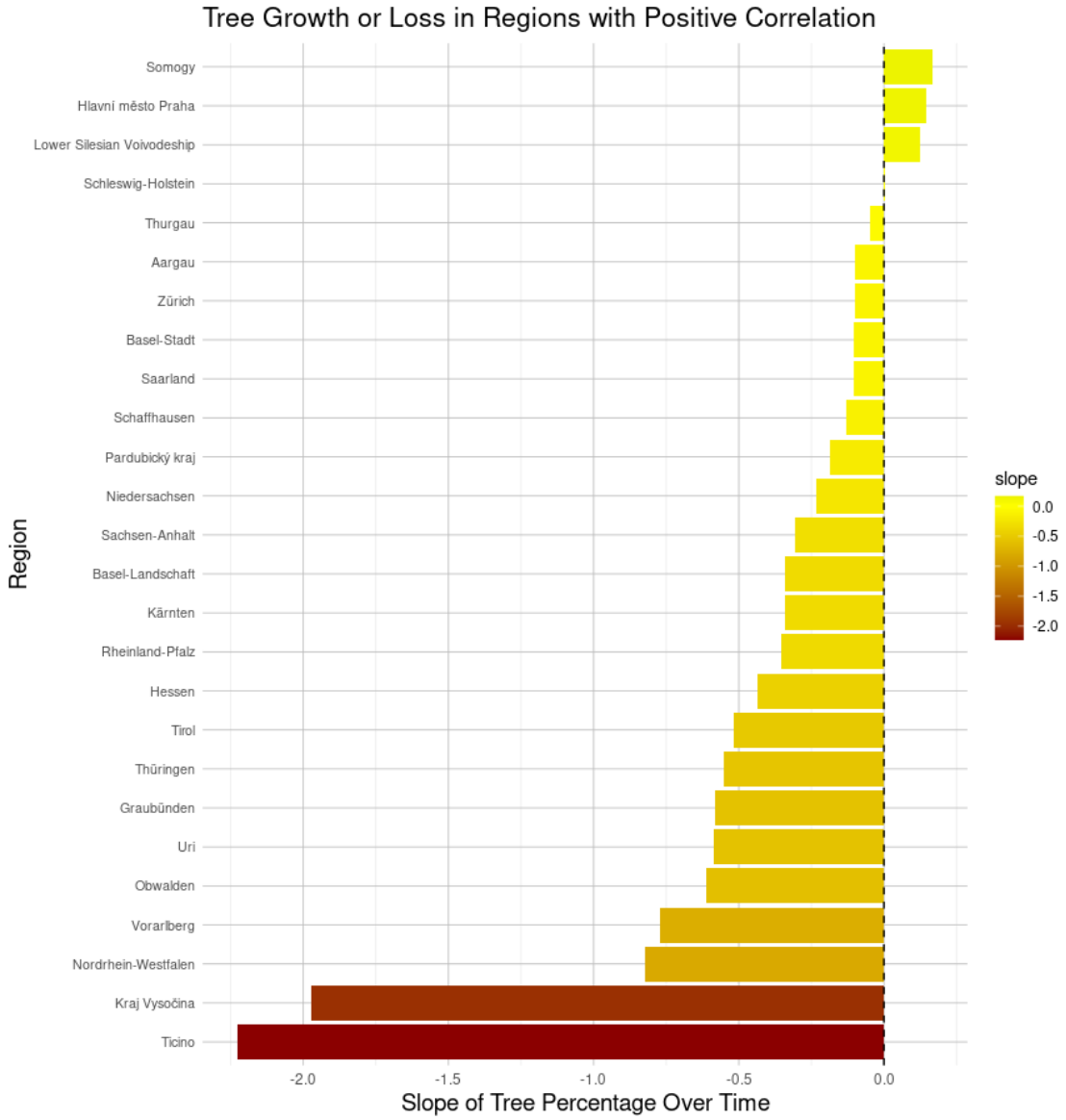
Positive correlation alone doesn't reveal whether changes in *tree_percentage* are due to tree growth or loss. To address this, we further analyze whether regions with positive correlation are experiencing an increase or decrease in tree coverage over time. The goal is to determine if regions with positive correlation have seen constant tree growth or loss. To do this, we calculate the **slope of a linear regression** between tree percentage and time (year) for each positively correlated region. The slope of linear regression reveals how the dependent variable changes with each unit increase in the independent variable. In our case this would mean, how much the tree cover increases/decreases over time. For example with every passing year, tree cover in a specific region decreases two times.

- **Positive Slope**: Tree growth, indicating an increase in tree coverage.
- **Negative Slope**: Tree loss, showing a decrease in tree coverage.

```
positive_corr <- correlation_results %>%
  filter(correlation > 0)

trend_results <- time_series_data %>%
  filter(shapeName %in% positive_corr$shapeName) %>%
```

```
  group_by(shapeName) %>%
  summarise(slope = coef(lm(tree_percentage ~ year))[2])  # Extract the slope from linear regre
ssion

positive_corr_trend <- positive_corr %>%
  left_join(trend_results, by = "shapeName")
```

The results are presented in the plot, where regions with negative correlation are sorted by their slope values. Regions with a positive slope indicate regions that experienced slight constant tree growth. On the other hand, regions with a negative slope reflect areas that have experienced tree loss over the observed years.



Slope of Tree Percentage Over Time in Regions with Positive Correlation.

We observe a clear trend of decreasing tree percentage year after year in most regions. Only a few regions, such as Somogy in Hungary and the Czech capital, Prague, experienced tree growth during this period.

Significant tree loss was observed in the regions of Kraj Vysočina (Czech Republic) and Ticino (Switzerland) where negative slope was around -2.. In 2018, Ticino had 48.6% tree cover, which decreased to around 41% in 2019 and 2020. Although there was a slight recovery with a 1% increase in 2021, this was followed by a sharp 7% decline in 2022, and a small 1% gain in 2023. Overall, Ticino has experienced a consistent decline in tree cover compared to its 2018 levels, and the previously observed negative correlation of -0.43 aligns with this downward trend.

Tree Cover Change in Ticino, Switzerland

Correlation Between Tree Percentage and Feature Variables

In addition to analyzing the changes in tree percentage over time, it is important to consider other environmental variables that may have influenced these trends. Several factors, such as `crops_percentage`, `rangeland_percentage`, and `built_area_percentage`, could have an impact on tree cover. Understanding the relationship between these variables and tree percentage will provide a more comprehensive view of the factors contributing to tree growth or loss in various regions.

One way to assess these relationships is by examining the correlations between `tree_percentage` and other land use variables. Specifically, we can explore whether increases in cropland or built-up areas are associated with decreases in tree cover, or if regions with more

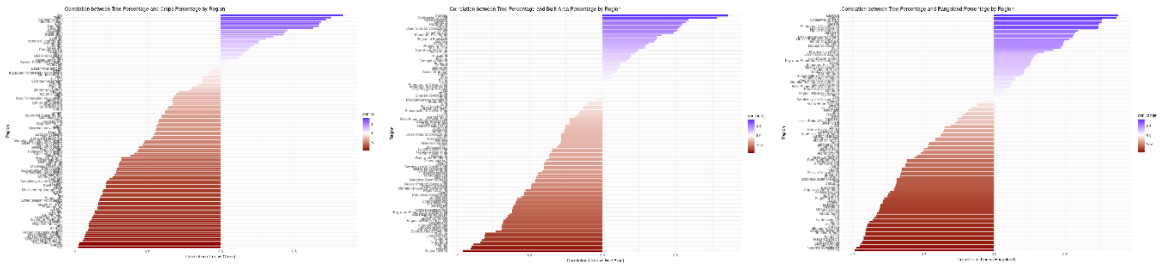
rangeland show different patterns. A **negative correlation** between tree cover and any of these variables would indicate an inverse relationship, where increases in one variable are associated with decreases in tree percentage. A **positive correlation** between tree cover and other variables would mean, that increase in one variable would affect increase in the other variable and vice versa for decrease.

By examining these correlations, we aim to better understand how land use changes, such as agricultural expansion or urbanization, may have affected forest and tree cover in the regions we are analyzing. This will help us identify the broader environmental trends that are contributing to changes in tree coverage over the observed period.

```
correlation_results <- time_series_data %>%
  group_by(shapeName) %>%
  summarise(
    cor_crops = cor(tree_percentage, crops_percentage, use = "complete.obs"),
    cor_rangeland = cor(tree_percentage, rangeland_percentage, use = "complete.obs"),
    cor_built_area = cor(tree_percentage, built_area_percentage, use = "complete.obs")
  )
```

Accessing Correlation

Correlation of Tree Percentage and Crops percentage, Rangeland Percentage and Built Area Percentage.



Correlation Between Tree Percentage and Built Area Percentage

The correlation between tree percentage and built area percentage reveals diverse regional trends:

- **Positive Correlation**: Regions such as **Schleswig-Holstein** and **Thurgau** exhibit positive correlations (blue bars). In these areas, urbanization seems to coexist with tree coverage, indicating that efforts may be in place to preserve trees despite urban expansion.

	shapeName	country	year	tree_percentage	built_area_percentage	crops_percentage	rangeland_percentage
1	Thurgau	Switzerland	2018	15.35109	7.889567	38.45092	0.6719250
2	Thurgau	Switzerland	2019	15.02519	7.595731	38.82270	0.5422335
3	Thurgau	Switzerland	2020	14.63463	7.404295	39.95911	0.5772814
4	Thurgau	Switzerland	2021	14.63261	7.368612	39.66233	0.4535823
5	Thurgau	Switzerland	2022	14.83731	7.760879	38.61622	0.6062741
6	Thurgau	Switzerland	2023	15.13217	7.692033	38.58274	0.5224204

Tree Percentage and Built Area Percentage Over years in Region of Thurgau, Switzerland

	shapeName	country	year	tree_percentage	built_area_percentage	crops_percentage	rangeland_percentage
1	Schleswig-Holstein	Germany	2018	9.257344	7.050101	63.23616	1.603168
2	Schleswig-Holstein	Germany	2019	9.187268	6.895954	63.91047	1.448659
3	Schleswig-Holstein	Germany	2020	9.212480	7.001733	63.23837	1.591543
4	Schleswig-Holstein	Germany	2021	9.253899	6.969419	62.56089	2.024999
5	Schleswig-Holstein	Germany	2022	9.282854	7.202088	62.02129	1.894059
6	Schleswig-Holstein	Germany	2023	9.225530	7.072624	63.14644	1.607380

Tree Percentage and Built Area percentage in Schleswig-Holstein, Germany

As indicated by the highly positive correlation between tree percentage and built area percentage, regions where built areas expanded did not experience tree loss. Instead, these regions actually saw tree growth during those years. Conversely, in cases where built area slightly decreased, tree cover also experienced some loss. This may be explained by broader land-use or economic changes. For instance, a reduction in built area could reflect urban decay, infrastructure degradation, or economic shifts that not only reduce construction but also lead to neglect of green spaces. It is worth to mention that in these areas tree cover is also not the main type of land cover but rater crops and expansion of this variable could be the one that in the same time has affected built area decrease and tree cover decrease.

- **Negative Correlation**: On the other hand, regions like **Zahodna Slovenia** and the **Region of Nitra** show strong negative correlations (red bars). This suggests that tree cover is being replaced by built-up areas, likely due to urban development or infrastructure expansion. Observations for the Region of Nitra in Slovakia indicate that in years where there was slight growth in built areas, tree cover experienced a corresponding slight decrease. Conversely, in years where a reduction in built area percentage was observed, the region saw a subsequent increase in tree cover.

	shapeName	country	year	tree_percentage	built_area_percentage
1	Region of Nitra	Slovakia	2018	15.14285	5.824435
2	Region of Nitra	Slovakia	2019	14.94010	5.870430
3	Region of Nitra	Slovakia	2020	14.78335	5.915031
4	Region of Nitra	Slovakia	2021	15.15076	5.802580
5	Region of Nitra	Slovakia	2022	14.57394	5.973271
6	Region of Nitra	Slovakia	2023	15.00607	5.804226

Correlation Between Tree Percentage and Crops Percentage

The correlation between tree percentage and crop percentage shows a generally negative relationship:

- **Positive Correlation**: A few regions, such as **Zug** and **Sachsen-Anhalt**, exhibit a slight positive correlation, suggesting that these areas are able to maintain or increase tree cover while expanding agricultural activity.

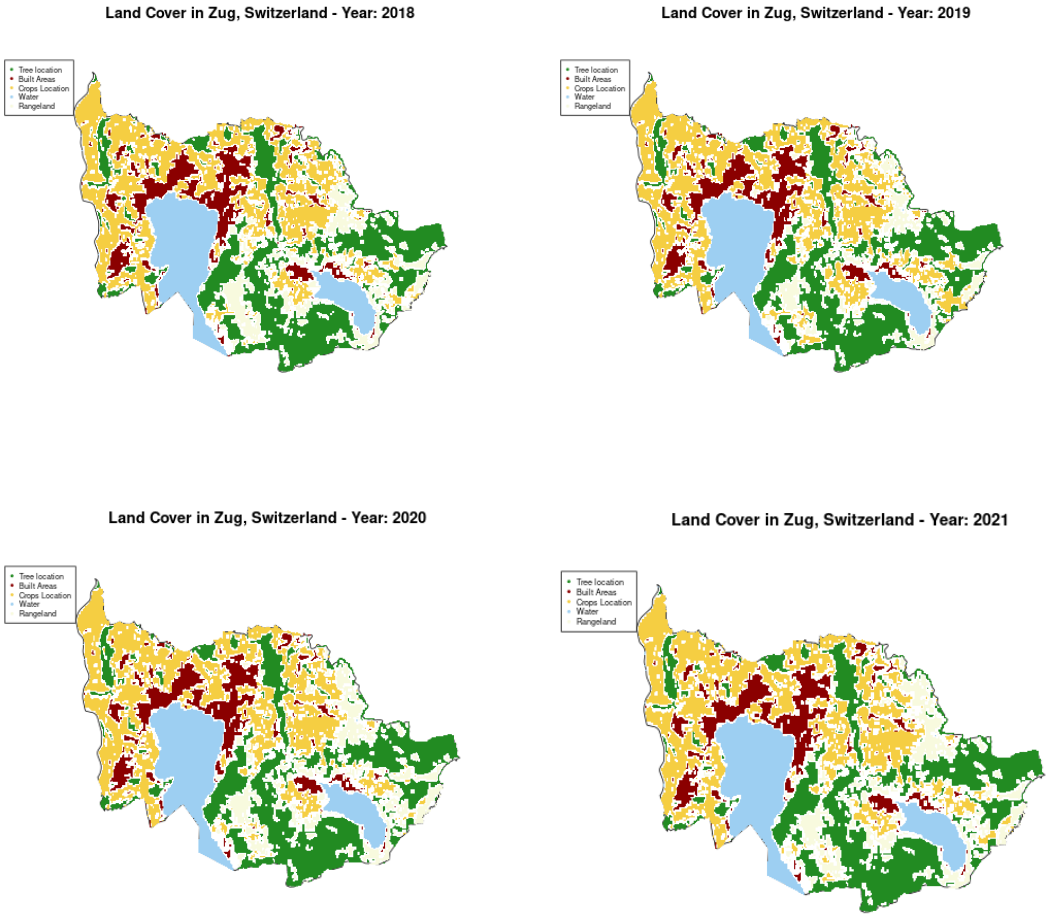
	shapeName	country	year	tree_percentage	built_area_percentage	crops_percentage	rangeland_percentage
1	Zug	Switzerland	2018	22.40392	8.209416	22.24890	6.000536
2	Zug	Switzerland	2019	22.47480	8.060644	23.29709	4.457411
3	Zug	Switzerland	2020	22.11453	8.083602	22.15078	6.392855
4	Zug	Switzerland	2021	22.23154	8.035330	21.67452	6.242256
5	Zug	Switzerland	2022	21.98976	8.367403	21.44413	6.375805
6	Zug	Switzerland	2023	22.75936	8.379360	23.05153	4.751911

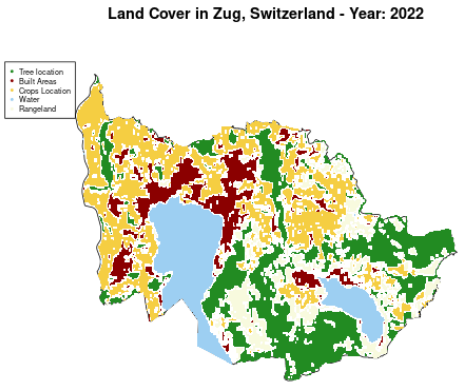
Tree Percentage and Built Area percentage in Zug, Switzerelnd

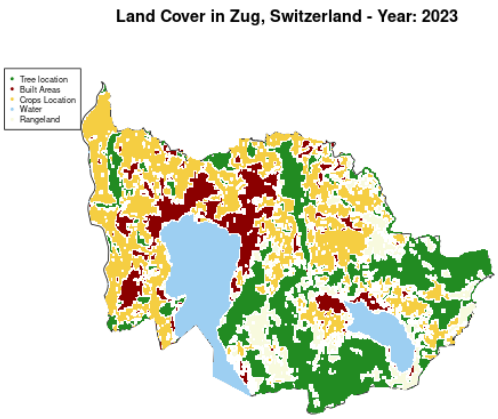
When examining the values for Zug over those years, we observe that the percentage of land used for agricultural activities and the percentage of tree cover are nearly identical. Both tree percentage and crops percentage experienced growth and loss in the same years. However, the changes in built area contradicted the trends for both variables. In years when crops and tree cover were growing, built areas slightly decreased (with small changes), and when built areas expanded, both tree and crops cover decreased.

This behavior can be explained by examining the tree cover raster for each year, which shows that tree cover and crop areas in Zug are almost "*disjoint*" (between areas with tree cover and crops cover, there is always some area of rangeland cover). In some parts of Zug, these areas border each other, but over the years, any growth in tree or crop areas did not occur where they border each other, but rather in regions where they do not affect one another. On the other hand, built areas are primarily located within the crops and rangeland areas (while tree cover is mostly in the mountainous Alpine regions, where settlements are rare). Thus, the expansion of built areas negatively affected the percentage of agricultural land use (crops).

Land Cover Change in Zug over the course of years 2018 to 2023. As these changes are best visible when animated, you can view an animated image that displays changes in land cover of Zug here.



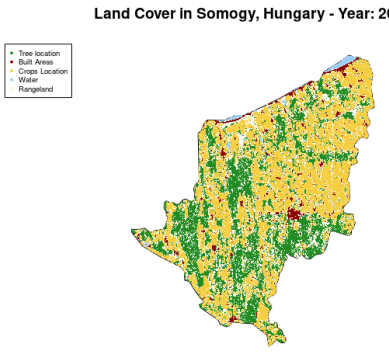


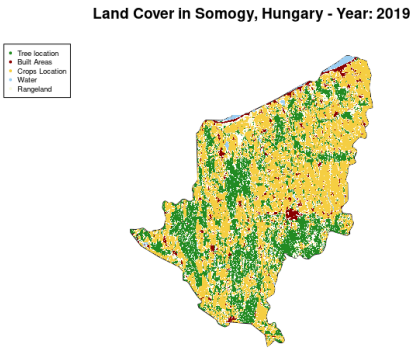


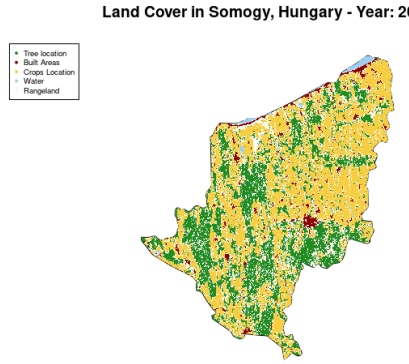
- **Negative Correlation**: A far more common trend is a negative correlation, seen in regions like **Somogy** and **Hlavní město Praha**. In these areas, as crop cultivation expands, tree coverage decreases, likely due to land conversion from forest to agricultural use.

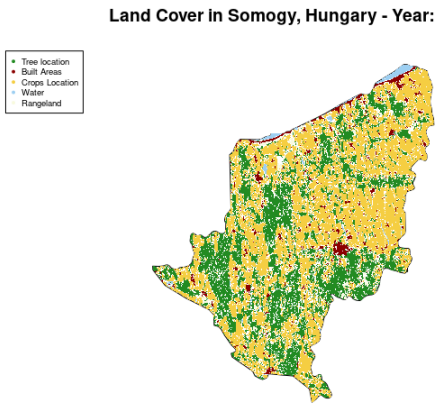
As observed from the land cover rasters of the Somogy area in Hungary, the two predominant land cover types are tree and crop areas, which are adjacent to each other. In this case, it is logical that changes in these two parameters significantly affect each other, with the positive expansion of crop areas contributing to the decrease in tree cover in the region.

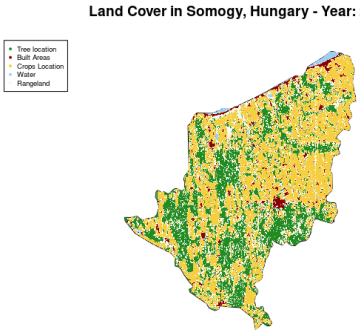
Land Cover Change in Somogy over the course of years 2018 to 2023. As these changes are better visible when animated, you can view how crop cover and tree cover replace each other in animated manner here.

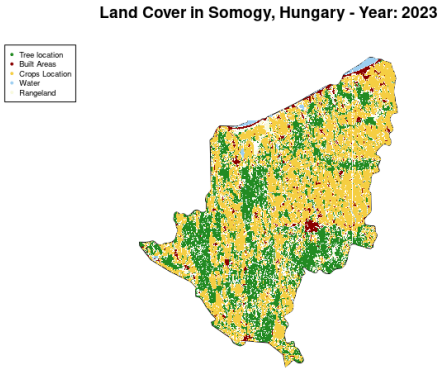




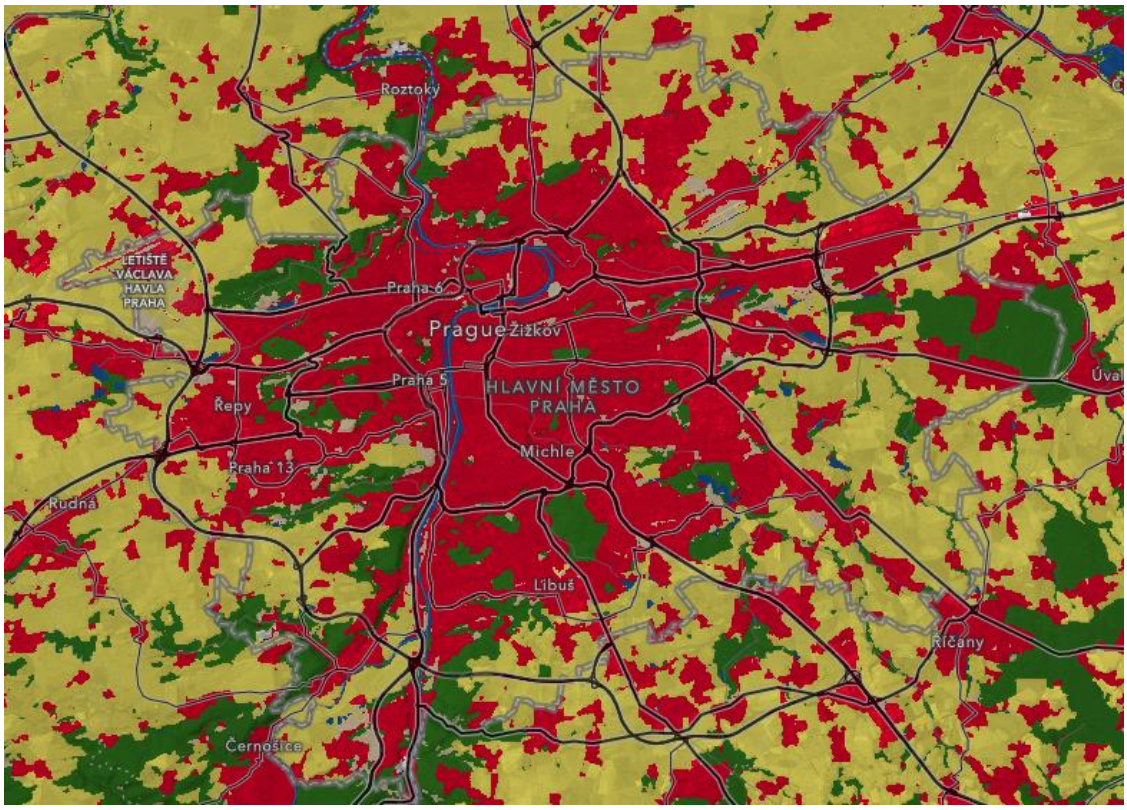








Similarly, in Prague, the capital of the Czech Republic, crop and tree areas are found along the city borders in less urbanized areas. Once again, it is evident that as agricultural land increases, the tree percentage decreases.



Land Use in Prague in 2018, where can be observed how crops and tree areas are satured alongside city boarders.

As negative correlation trends are mostly present in correlation table of tree percentage and crops percentage, this indicates pressure that agricultural activities can place on forested areas, with **most regions showing that expanding crop production leads to a reduction in tree cover**.

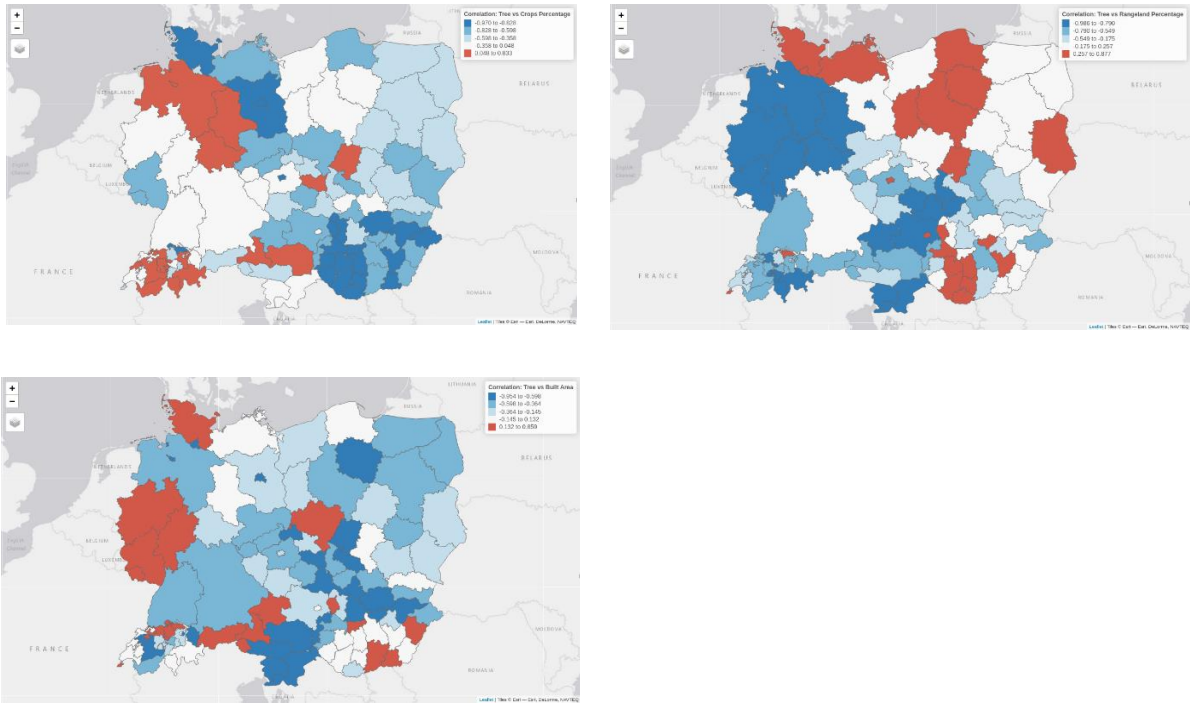
Correlation Between Tree Percentage and Rangeland Percentage

Similar to the crops percentage, the correlation between tree percentage and rangeland percentage shows largely negative relationships:

- **Positive Correlation**: Some regions, such as **Baranya** and **Nograd Voivodeship**, show positive correlations, where tree cover and rangeland areas appear to coexist. This may indicate effective land management that accommodates both grazing and forest conservation.
- **Negative Correlation**: However, the majority of regions, like **Hessen** and **Nordrhein-Westfalen**, exhibit strong negative correlations, implying that as rangeland expands, tree cover diminishes, likely due to forest clearing for grazing purposes.

Spatial distribution of correlation between tree percentage and feature variables

Spatial distribution of correlation between tree percentage and feature variables.



From the maps in the figure above, it can be observed that areas with high negative or high positive correlation for each category (tree percentage vs. crops percentage, tree percentage vs. built area percentage, and tree percentage vs. rangeland percentage) are mostly spatially grouped, as neighboring states tend to have similar land cover usage. In regions where a negative correlation between tree percentage and other variables was observed, it is due to competition for land resources, meaning crop areas, rangeland areas, or built-up areas are directly adjacent to tree cover. Conversely, in areas where a positive correlation was observed between tree percentage and another variable, there is typically less spatial proximity between tree cover and the other variable. Instead, both are separated by a third type of land use. When this third land cover expands or decreases, it affects both tree cover and the comparison variable, leading them to increase or decrease together.

Conclusion

The relationships between various land use covers—such as crops, rangeland, and built areas—and tree percentage across different regions predominantly show negative correlations. This suggests that as agricultural activities, urban expansion, or other land uses increase, tree cover tends to decrease. This inverse relationship is most evident in regions where tree and crop areas or built-up zones are **spatially adjacent** or competing for land resources.

In areas where agricultural expansion is necessary, such as in Somogy and suburbs of Prague, increased land use for crops has directly contributed to the loss of tree cover. Similarly, in urbanized regions, the growth of built areas often encroaches upon green spaces, further diminishing tree coverage. These dynamics highlight the trade-offs between development and

environmental conservation, where land used for urbanization or agriculture tends to result in a reduction in tree-covered areas, particularly in regions where available land is already limited.

Model Fitting

We aim to model and predict tree percentage based on multiple land-use and pollution-related variables. Understanding how different variables such as built area percentage, crops percentage, rangeland percentage, and pollution levels influence tree coverage is essential to guide environmental policies and land-use decisions. The primary goal is to fit a model that can accurately predict tree percentage using these variables.

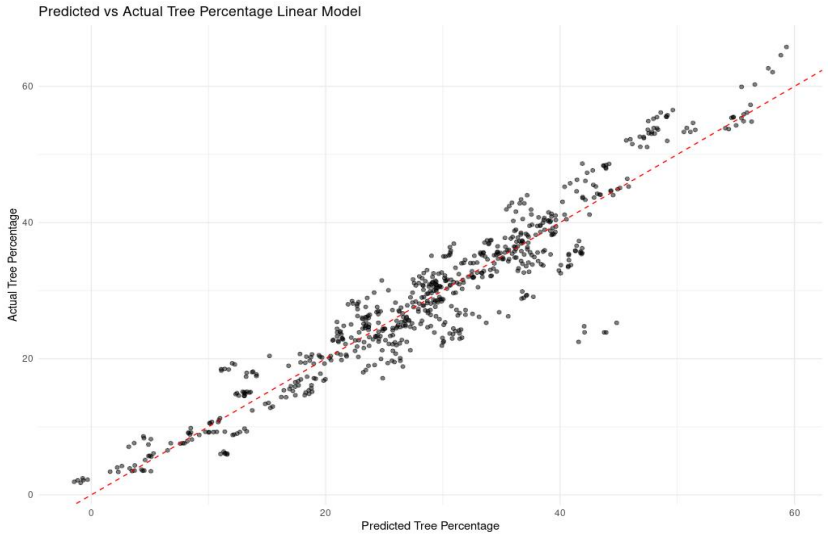
Model Fitting Approaches

We began by fitting several models to predict tree percentage, each of which was evaluated for performance:

Linear Regression Model

First we tried to fit a simple linear model where tree percentage is assumed to have a linear relationship with land-use and pollution-related variables.

```
model <- lm(tree_percentage ~ built_area_percentage + crops_percentage
            + rangeland_percentage + water_percentage
            + flooded_vegetation_percentage + bare_ground_percentage
            + snow_ice_percentage + burned_area
            + CO2_total + PM25_total + TPC_total + NMHC_total
            + OC_total + CH4_total + SO2_total
            + BC_total, data = time_series_data)
time_series_data$predicted_lm <- predict(model)
```

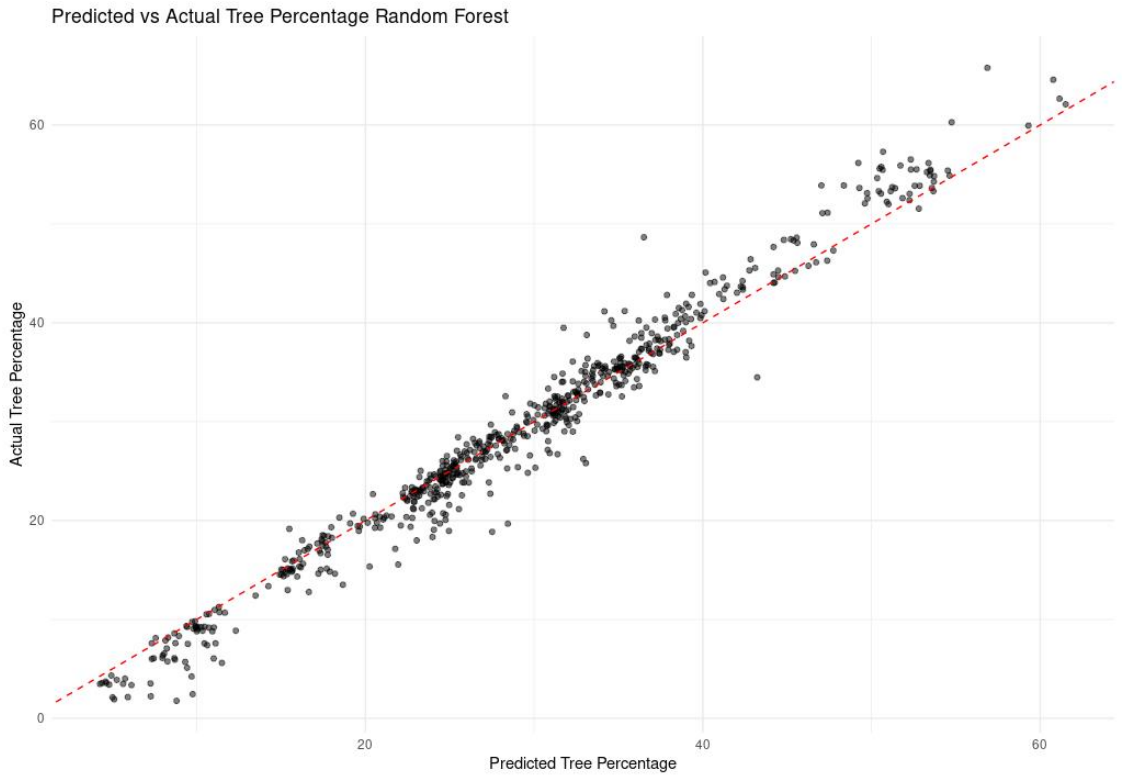


Predicted vs. Actual Tree Percentage from Linear regression Model. *The linear regression model provided an initial understanding but showed not the best fit and high residuals, meaning the predictions were often far off from the actual values and grouped in clusters that overestimated or underestimated the actual tree percentage values. As observed in the previous section, where we analyzed the correlation between tree percentage and various other feature variables, tree percentage did not show positive correlations with most variables across regions. Negative correlations were predominant, and the values rarely followed a linear pattern, which explains why a linear model is not best-suited to our dataset.*

Random Forest Regression

Random Forest Regression Model combines multiple decision trees to create a single model. Each tree in the forest builds from different subset of the data and makes its own independent prediction, whereas the final prediction for the input is based on the average or weighted average of all the individual trees' predictions. This introduces a more flexible non-linear model which could help us capture more complex relationships between the tree percentage and predictor variables.

```
model_rf <- randomForest(tree_percentage ~ built_area_percentage + crops_percentage
                         + rangeland_percentage + water_percentage
                         + flooded_vegetation_percentage + bare_ground_percentage
                         + snow_ice_percentage + burned_area
                         + CO2_total + PM25_total + TPC_total
                         + NMHC_total + OC_total + CH4_total + SO2_total
                         + BC_total, data = time_series_data)
time_series_data$predicted_rf <- predict(model_rf)
```



Predicted vs. Actual Tree Percentage from Random Forest Regression Model. *As many points closely follow the ideal line, it is suggested that Random Forest model generally performs well in predicting tree percentage. However, there are deviations, especially at higher and lower tree percentages, indicating that the model may have some difficulty accurately predicting very high or very high or very low tree percentage values. While the model seems to capture the overall trend, further tuning or more complex models might improve performance, especially in these extreme cases.*

There are two main clusters of points below and under the ideal (red) line. The one for tree percentages being between 0% and 10% tree percentage, where the model for these low values overestimated the actual tree percentage. The other one is grouping high tree percentage values (between 45 and 60% tree percentage) over the ideal line, meaning that the model has

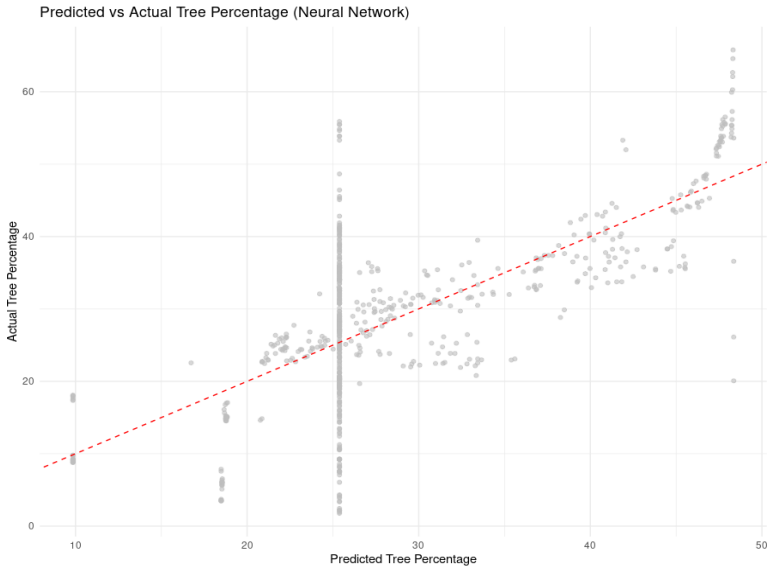
underestimated those values and predicted lower tree percentage than it actually should be the case.

Neural Network Model

Neural networks are a class of machine learning models inspired by the structure and function of the human brain, making them highly effective at capturing complex, non-linear relationships in data. Given the diverse and multi-dimensional nature of my dataset, which includes variables like built area percentage, crops percentage, and pollution levels, a neural network was a promising candidate for predicting tree percentage. By employing multiple hidden layers and non-linear transformations, the model has the potential to uncover intricate interactions between these variables that traditional linear models might overlook.

```
nn_model <- nnet(tree_percentage ~ built_area_percentage + crops_percentage
                 + rangeland_percentage + water_percentage
                 + flooded_vegetation_percentage + bare_ground_percentage
                 + snow_ice_percentage + burned_area
                 + CO2_total + PM25_total + TPC_total + NMHC_total
                 + OC_total + CH4_total + SO2_total + BC_total,
                 data = time_series_data, size = 10, linout = TRUE)

time_series_data$predicted_nnet <- predict(nn_model, time_series_data)
```

Although neural networks are powerful, they can also be prone to overfitting, especially in cases where the data may not be sufficient or complex enough to warrant such a sophisticated approach. The neural network, though capable of handling the multi-dimensional nature of the data, struggled with accurate predictions, as reflected in the scatterplot comparing actual and predicted values.

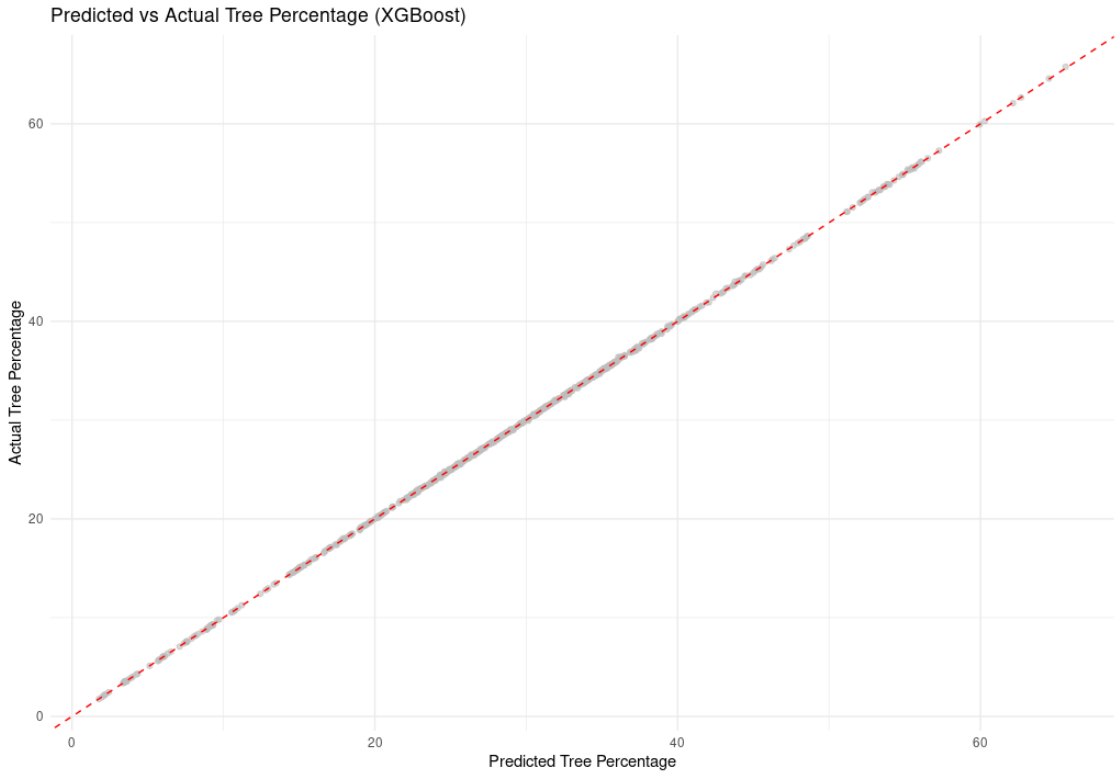


Actual vs. Predicted Tree Percentage From Neural Network Model. *There is a noticeable vertical clustering of points, which suggests that the model is assigning similar predicted values regardless of the actual variation in tree percentages. This indicates that the neural network is struggling to capture the variance in the data, leading to poor predictions across various regions. While some points are close to the ideal line showing decent predictions, there is a wide scatter of points, particularly in areas where the tree percentage is abow 20%. This spread indicates that the neural network is not accurately capturing the relationship between the predictors and tree percentage, leading to suboptimal results.*

XGBoost Regression Model

XGBoost is an advanced ensemble learning technique that efficiently handles non-linearities and complex interactions in the data. This model uses similar decision tree approach as Random Forest Regression, with the difference, that XGBoost develops one tree at a time, correcting faults caused by previously trained trees, where as Random Forest generates each tree independently and aggregates results at the end. The gradient decent algorithm is used to minimize the loss when adding new trees.

```
data_matrix <- model.matrix(tree_percentage ~ built_area_percentage + crops_percentage
                            + rangeland_percentage + water_percentage
                            + flooded_vegetation_percentage + bare_ground_percentage
                            + snow_ice_percentage + burned_area
                            + CO2_total + PM25_total + TPC_total
                            + NMHC_total + OC_total + CH4_total + SO2_total + BC_total,
                            data = time_series_data)

xgb_model <- xgboost(data = data_matrix, label = time_series_data$tree_percentage,
                     nrounds = 100, objective = "reg:squarederror")

time_series_data$predicted_xgboost <- predict(xgb_model, data_matrix)
```



Actual vs. Predicted Tree Percentages from XGBoost. *XGBoost shows an almost perfect alignment between the predicted and actual tree percentages, as the points tightly follow the ideal line, which represents the ideal $x = y$ relationship. Model appears to have effectively captured underlying relationships between the tree percentage and predictor variables and there is very little deviation between actual and predicted values.*

Model Comparison

We will now compare the four models once again to ensure that XGBoost is indeed the best fitting model. To do this, we can use the R-squared metric, also known as the coefficient of

determination, which measures how well the model's predictions align with the actual data. The R-squared value ranges from 0 to 1, where 1 indicates that the model explains all the variability in the response variable, while 0 means that the model fails to explain any of the variability. This comparison will provide a clearer understanding of each model's performance.

```
rsquared <- function(actual, predicted) {
  cor(actual, predicted) ^ 2
}

# R² for linear model
rsquared_lm <- rsquared(time_series_data$tree_percentage, time_series_data$predicted_lm)

# R² for nnet model
rsquared_nnet <- rsquared(time_series_data$tree_percentage, time_series_data$predicted_nnet)

# R² for Random Forest model
rsquared_rf <- rsquared(time_series_data$tree_percentage, time_series_data$predicted_rf)

# R² for XGBoost model
rsquared_xgb <- rsquared(time_series_data$tree_percentage, time_series_data$predicted_xgboost)
```

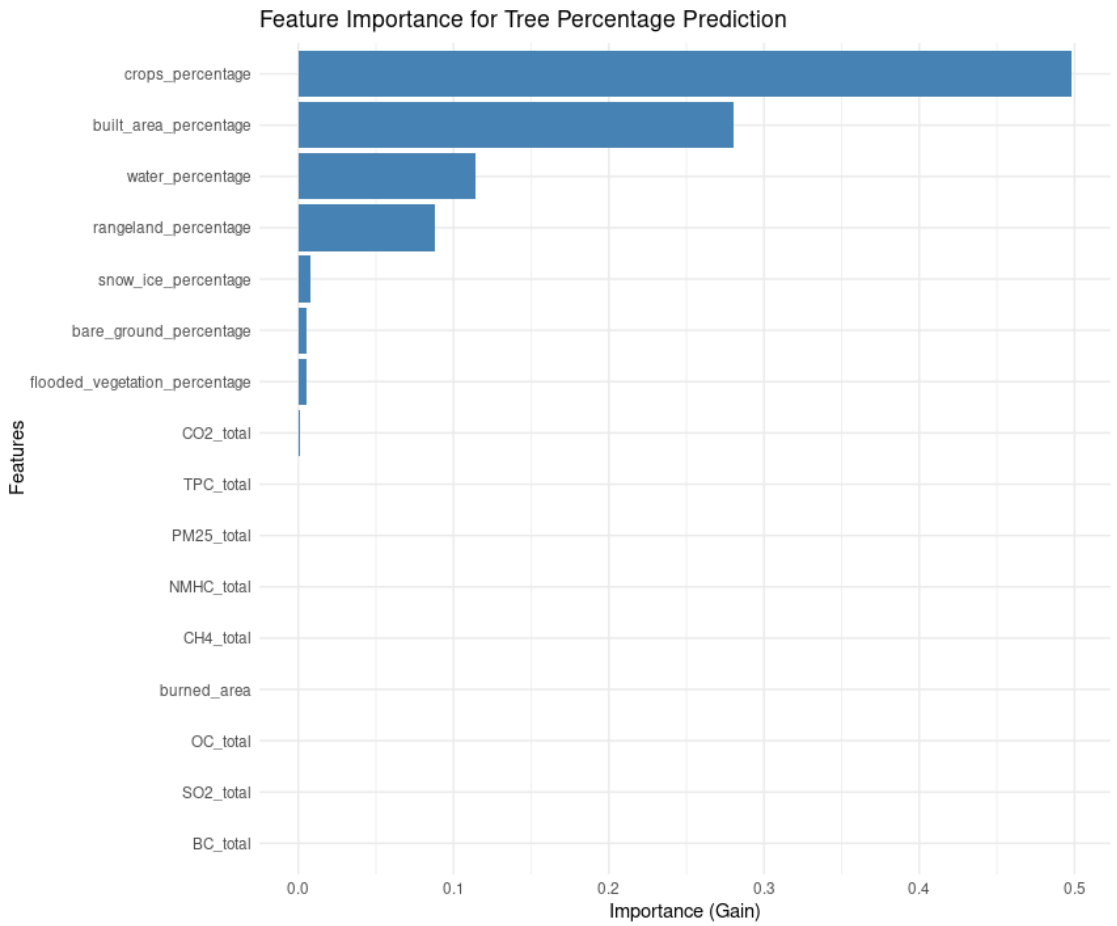
```
> cat("R² - Linear Model:", rsquared_lm, "\n")
R² - Linear Model: 0.9037152
> cat("R² - Random Forest:", rsquared_rf, "\n")
R² - Random Forest: 0.974837
> cat("R² - NNet Model:", rsquared_nnet, "\n")
R² - NNet Model: 0.5159236
> cat("R² - XGBoost:", rsquared_xgb, "\n")
R² - XGBoost: 0.9999732
```

R-squared Results.

The results show that XGBoost is the best-performing model compared to the others tested, as demonstrated by its R-squared value of nearly 1, indicating a near-perfect fit. This suggests that XGBoost is able to capture almost all the variability in the tree percentage, making it the most reliable for predictions in our dataset. Random Forest model also indicates high R-squared metric but as we have seen from its predictions, XGBoost stil outperforms it.

Importance of Predictor Variables

With XGBoost performing so well, it becomes important to understand which features contribute most to ist predictions. The next step is to analyze the feature importance, which helps us identify which variables have the greatest influence on tree percentage.



Importance of Predictor Variables for XGBoost Model.

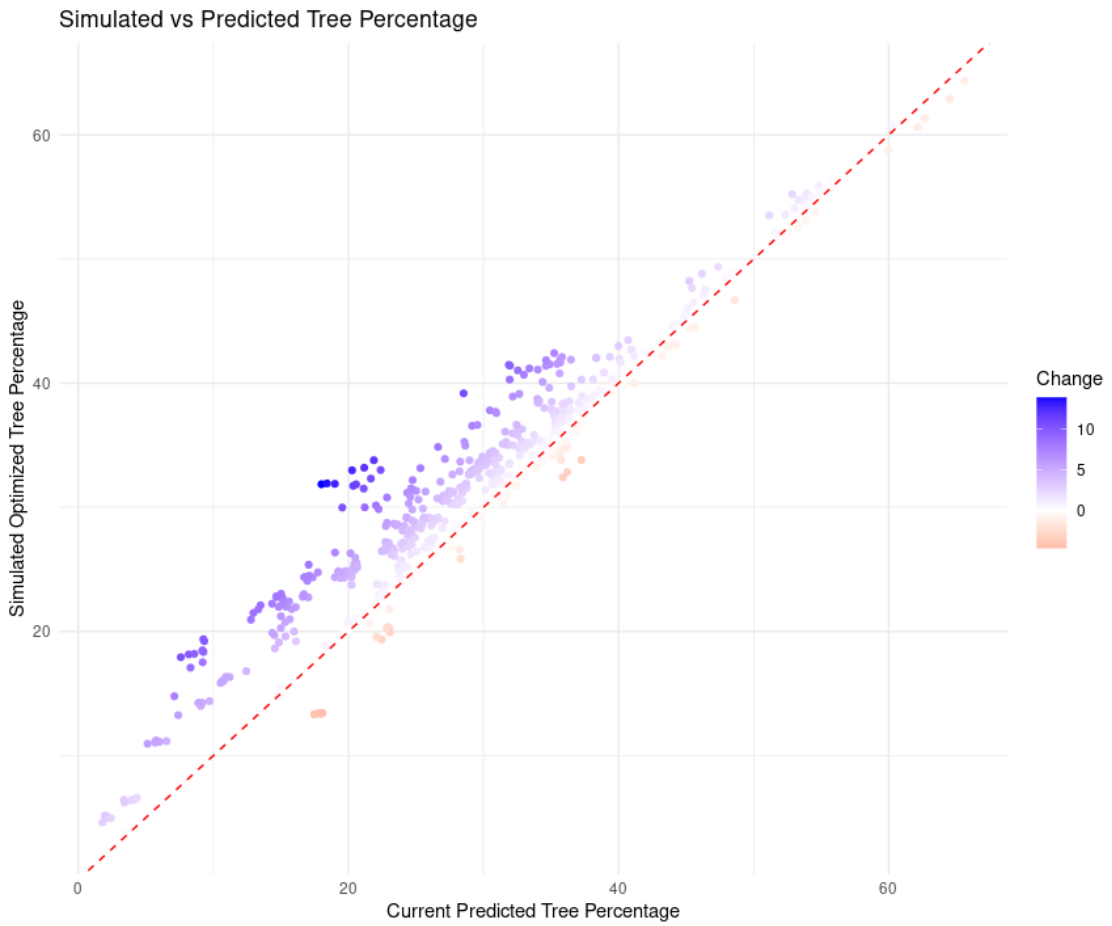
The importance plot reveals that `crops_percentage` and `built_area_percentage` are the most significant drivers influencing tree coverage, followed by `water_percentage` and `rangeland_percentage`. This makes sense because crops and built areas are the primary competitors for land use, often located in geographically adjacent areas. Interestingly, gas emission variables have almost no effect on tree percentage. While the plot suggests their impact is negligible, closer inspection of the data frame shows that these values are not exactly zero, but very close to it. This could be due to the fact that emission measurements were taken during years with wildfires and are not consistently present in the data. As a result, their overall influence is lower compared to other variables that exhibit consistent and permanent changes contributing to tree cover fluctuations, whether positive or negative.

Simulating the Impact of Land-Use and Pollution Reductions on Tree Coverage

In this section, we aim to simulate how changes in key predictor variables could affect tree percentage across different regions. The goal is to understand how adjustments in land use and pollution levels could potentially lead to optimized tree coverage. This type of simulation helps provide actionable insights into land management and environmental policy-making by showing the direct impact of reducing urban sprawl, agricultural expansion, or pollution levels on tree cover.

To achieve this, we define a simulation function that applies percentage-based reductions to key variables such as *built_area_percentage*, *crops_percentage*, and *CO2_total* (as a representative of pollution, since other gas emissions showed importance for the model very near zero). For example, reducing built areas by 10%, crops by 10%, and pollution by 20% helps us explore how these changes can optimize tree coverage. Although *water_percentage* and *rangeland_percentage* play crucial roles in predicting tree percentage, it wouldn't be realistic to simulate changes to these values. Increasing water percentage, which supports tree growth, or decreasing rangeland percentage, which primarily reduces tree cover, would involve altering natural land cover types. Reducing rangeland would result in the destruction of natural habitats, and it is not feasible for land-use management to artificially increase water percentage without negatively impacting other environmental factors. Water must reach the land through natural processes.

```
simulate_tree_percentage <- function(data, model,
                                     built_area_change, crops_change,
                                     pollution_reduction) {

  simulated_data <- data
  simulated_data$built_area_percentage <- simulated_data$built_area_percentage *
                                          (1 - built_area_change)
  simulated_data$crops_percentage <- simulated_data$crops_percentage * (1 - crops_change)
  simulated_data$CO2_total <- simulated_data$CO2_total * (1 - pollution_reduction)

  simulated_data_matrix <- model.matrix(tree_percentage ~ built_area_percentage
                                        + crops_percentage + rangeland_percentage
                                        + water_percentage + flooded_vegetation_percentage
                                        + bare_ground_percentage + snow_ice_percentage
                                        + burned_area + CO2_total + PM25_total
                                        + TPC_total + NMHC_total + OC_total
                                        + CH4_total + SO2_total + BC_total,
                                        data = simulated_data)

  simulated_data$predicted_tree_percentage <- predict(model, simulated_data_matrix)

  return(simulated_data)
}

# Reduce built area by 10%, crops by 10%, and pollution by 20%
simulated_scenario <- simulate_tree_percentage(time_series_data, xgb_model,
                                               built_area_change = 0.10,
                                               crops_change = 0.10,
                                               pollution_reduction = 0.20)
```



Simulated Optimized Tree Percentage vs Predicted Tree Percentage.

As we can see from the comparison of predicted and simulated optimized tree percentages, reduction of built areas, crops and CO2 pollution would result in the most observations having higher tree percentage. Since our data is currently in long format, where each observation stands for a single Central European region for a given year, we will summarize the results for each region to take a closer look at winnings we observed for each region over the period of six years from 2018 to 2023.

We created a comparison dataset that calculates both the **predicted tree percentage** from the XGBoost model and the **simulated tree percentage** after applying land-use policy changes. The difference between the two values allows us to identify whether regions experienced improvements or declines in tree coverage based on the simulation. To better understand the effect of these simulated changes at the **regional level**, we computed the average predicted and simulated tree percentages for each region across the six years. This aggregation helped to see broader trends in tree coverage changes, ensuring that the analysis wasn't overly granular.

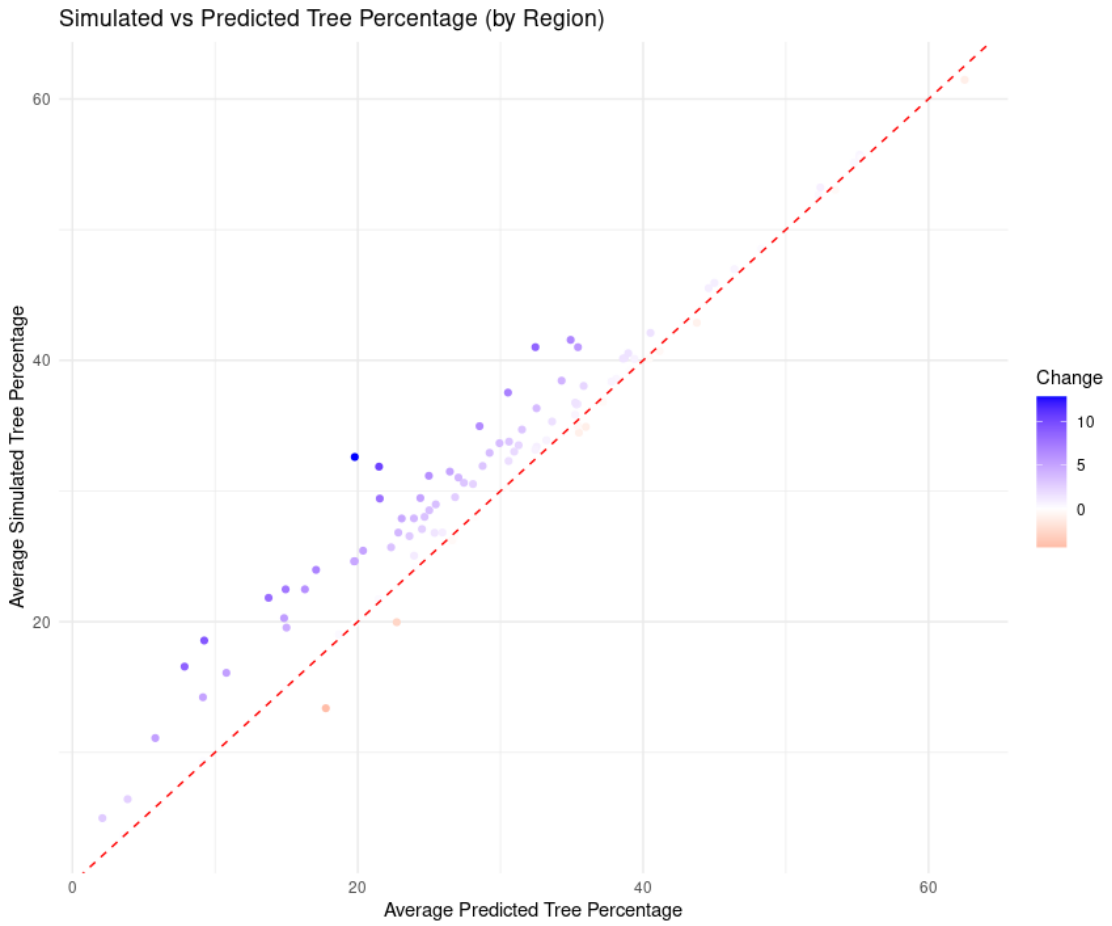
```
regional_comparison <- comparison %>%
  group_by(shapeName) %>%
  summarise(
    avg_predicted_tree_percentage = mean(predicted_xgboost, na.rm = TRUE),
    avg_simulated_tree_percentage = mean(simulated_tree_percentage, na.rm = TRUE)
  ) %>%
  mutate(difference = avg_simulated_tree_percentage - avg_predicted_tree_percentage)
```

```r
summary_by_region <- regional_comparison %>%
  summarise(
    total_improved = sum(difference > 0),
    total_declined = sum(difference < 0),
    avg_improvement = mean(difference[difference > 0], na.rm = TRUE),
    avg_decline = mean(difference[difference < 0], na.rm = TRUE)
  )
```

```
# A tibble: 1 × 4
  total_improved total_declined avg_improvement avg_decline
           <int>          <int>           <dbl>       <dbl>
1             91             19            3.06      -0.767
```

Predictor Optimization Results by Region.

As seen from the output, **91 regions** experienced improvements in tree percentage after the simulated changes in predictor variables. The **average yearly improvement** in regions where tree percentage increase was simulated was 3.06%. On the other hand **19 regions** experienced slight decline in tree cover percentage where yearly decline was -0.767%.



Simulated vs Predicted Tree Percentage by Region. *Almost all regions in our dataset fall in the upper part of the scatterplot, indicating tree cover growth. However, there are a few regions where the average simulated tree growth remained unchanged, and some regions where the simulated tree growth, with adjusted predictor variables, was lower than expected. It's important to note that in some regions, tree percentage correlates positively with crops and built area percentages. This means that a reduction in these two variables can lead to a reduction in tree percentage. This typically occurs in regions where built area and crops are geographically disjoint from tree cover, separated by other types of land use.*

In conclusion, land use management will need to carefully consider whether an average yearly tree cover growth of 3% is a desirable goal, or if the trade-off is too steep given the reduction in crops and settlement areas needed to achieve it. While increasing tree cover can be beneficial for the environment, it comes at the cost of important land use for citizens, agriculture, and infrastructure. However, with the implementation of the simulation function, there is flexibility to experiment with different scenarios and values. This allows for informed decision-making on how to best balance tree cover growth with other land use priorities.

Spatio-temporal Clustering

Spatio-temporal analysis is the final stage of this research, where we explore the changes in tree percentage not only across regions but also over time. This approach allows researchers to detect patterns, trends, and anomalies across geographical locations as well as time periods. By combining spatial data with temporal trends, we aim to uncover the dynamic relationships between environmental factors and land use practices, helping to identify regions that are undergoing rapid changes and those that remain relatively stable.

Clustering methods generally focus either on spatial proximity or temporal behavior, but in many real-world applications, both dimensions are crucial. **Spatio-temporal clustering** accounts for the fact that regions can be similar based on where they are located (spatial) and how they behave over time (temporal). Combining these two aspects provides a more holistic understanding of how a region's attributes evolve in relation to its neighbors and over time.

The Challenges of Spatio-Temporal Clustering

The main challenge in spatio-temporal clustering was finding the right algorithm capable of detecting spatially close regions that exhibit similar trends in tree percentage change over time. Existing literature primarily focuses on the concept of hotspots, where increasing values are observed in particular regions over a specific time span.

For my research on sustainability data in Central Europe, I explored various spatio-temporal clustering approaches. One of the most popular methods for this type of analysis is ST-DBSCAN (Spatial-Temporal Density-Based Spatial Clustering of Applications with Noise). As a density-based algorithm, DBSCAN partitions data into clusters based on the distance between points. When a temporal dimension is also added, the algorithm assesses both spatial and temporal distances to cluster the data accordingly. It uses two key parameters: `eps`, which represents the minimum spatial distance between two objects, and `eps2`, which defines the minimum temporal distance. Since the data must be properly scaled for the algorithm to work, finding the appropriate parameters that reveal meaningful clustering can be quite challenging and I while trying to make a use of this algorithm I wasn't able to find proper parameters that would reveal meaningful clustering. The algorithm can be found *here*.

While searching for a more suitable approach that could better capture our data and make more informed clustering decisions, I came across a novel spatio-temporal clustering algorithm introduced by *Soudeep and Sayar in 2023*. Their algorithm was successfully applied to data capturing the development of COVID-19 in the United States, effectively identifying similar trends in neighboring regions. Given our goal of identifying spatially close regions that show similar trends in tree change over time, I decided to apply their algorithm, which leverages a weighted combination of spatial and temporal distance matrices. This approach also utilizes gap statistics

to determine the optimal number of clusters and uses PAM Clustering to make final decision based on combined distance matrix that includes both spatial and temporal distances.

PAM (Partitioning Around Medoids) For Spatio-Temporal Clustering

General Idea of PAM Clustering

The PAM (Partitioning Around Medoids) clustering algorithm is well-suited for spatio-temporal data because it can efficiently handle mixed types of distances (both spatial and temporal) and is robust against noise. Unlike K-means, which uses centroids, PAM works by selecting actual observations (medoids) to represent clusters.

The core idea behind PAM is to:

1. **Select Medoids**: Choose actual data points as representative objects for each cluster (called medoids).
2. **Assign Clusters**: Each data point is assigned to the nearest medoid based on the combined distance metric (spatial and temporal).
3. **Optimize Medoids**: Iteratively improve the choice of medoids to minimize the overall dissimilarity within clusters.

Why PAM for This Study?

In my project, I explored various clustering techniques, including those designed for purely spatial or purely temporal data. However, these approaches failed to produce meaningful results, especially in terms of capturing the joint spatial-temporal behavior of tree coverage across different regions.

The algorithm introduced in the paper *A novel spatio-temporal clustering algorithm with applications on COVID-19 the United States* provided above presents a powerful approach for combining spatial and temporal distances in a meaningful way. It allowed me to blend spatial proximity and temporal trends into a single combined distance matrix, ensuring that regions were clustered not just by where they were located, but also by how their tree coverage changed over time.

PAM clustering was chosen because:

1. It handles **non-Euclidean distance matrices** (like the combined spatial-temporal matrix).
2. It is robust to **outliers and noise** in the data.
3. It provides **interpretable medoids**, which are actual data points representing clusters.
4. It allows for flexibility balancing the importance of spatial and temporal factors through the use of a weighted parameter, α. This enables us to experiment with how much influence each component (spatial or temporal) has on the clustering results, ensuring that we can capture meaningful patterns in the data that reflect both the geographical and temporal aspects of tree coverage changes.

The Approach

The spatio-temporal clustering in this study was carried out by constructing two distance matrices: one representing spatial distances between regions and the other representing

temporal differences in tree percentage over the years. These distances were then normalized using the Frobenius norm to ensure that both spatial and temporal distances contributed comparably to the final clustering.

```
polygon_distances <- st_distance(data_with_state_boarders$geometry)
spatial_distances <- as.matrix(polygon_distances)

temporal_data <- data %>%
  select(starts_with("tree_percentage")) %>%
  as.matrix()

temporal_distances <- as.matrix(dist(temporal_data)) # Eucledian distance between time series

normalize <- function(mat) {
  mat / sqrt(sum(mat^2))  # Frobenius norm
}

spatial_distances_norm <- normalize(spatial_distances)
temporal_distances_norm <- normalize(temporal_distances)
```

Adjusting the weighting parameter α, allows us to explore various configurations of spatio-temporal balance. When $\alpha = 1$, only temporal differences are considered, and when $\alpha = 0$, only spatial distances are considered. When α (i.e. $\alpha = 0.1$) is close to zero, geographically close regions are more likely to be clustered together regardless of how their tree percentage changes over time. On the other hand, when α is close to one (i.e. $\alpha = 0.9$) emporal trends dominate, meaning regions with similar changes in tree percentage over time are clustered together, even if they are geographically distant.

The general formula used to compute the **combined distance matrix** for the actuall clustering is:

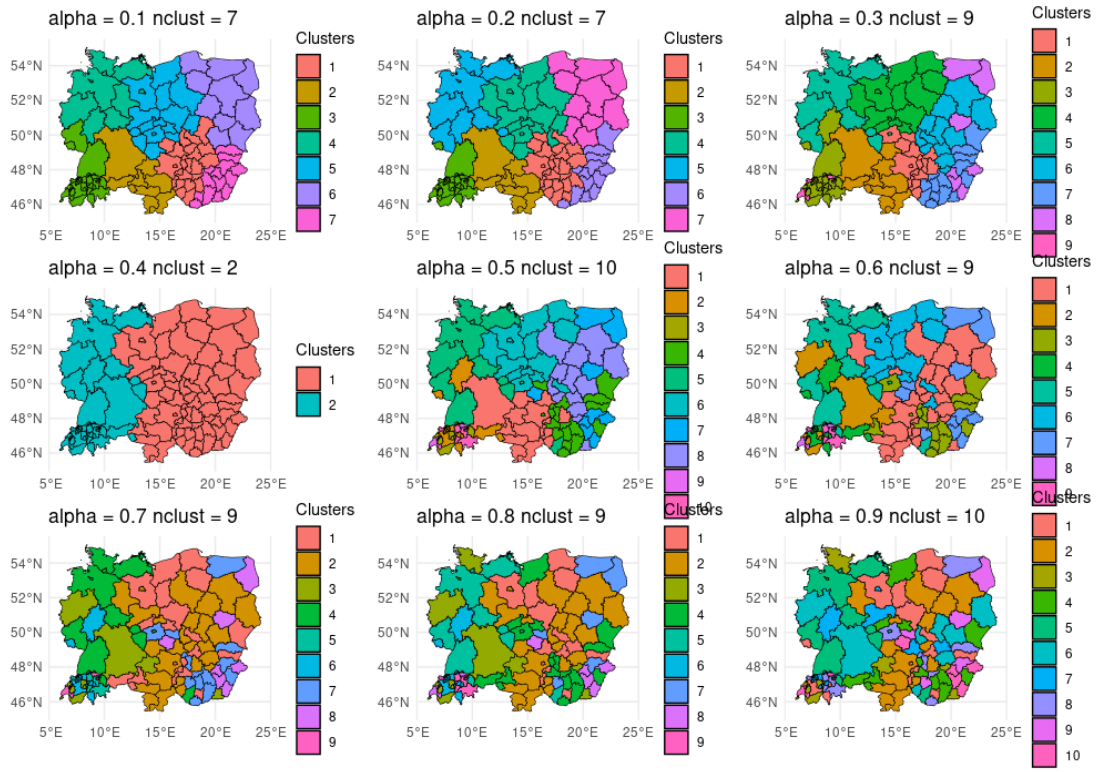
$$D_{combined} = \alpha \times D_{temporal} + (1 - \alpha) \times D_{spatial}$$

$D_{temporal}$ is in our case distance matrix representing differences in tree percentage over time, but can be calculated for any combination of variables. In the further section, we are going to cluster data using this approach based on temporal data about tree cover and other predictor variables, but right now, we are going to focus on clustering based only on temporal data of tree percentage and its spatial distribution. $D_{spatial}$ is matrix of spatial distances between regions.

The **gap statistic** was used to determine the optimal number of clusters for each value of α by comparing the within-cluster variance for a given number of clusters to that of a reference distribution, helping to find the number of clusters that best fits the data without overfitting.

```
for (alpha in seq(0.1, 0.9, by = 0.1)) {

  combined_distances <- alpha * temporal_distances_norm + (1 - alpha) * spatial_distances_norm

  gap_stat <- clusGap(combined_distances, FUN = pam, K.max = 10, B = 50)

  optimal_clusters <- maxSE(f = gap_stat$Tab[, "gap"], SE.f = gap_stat$Tab[, "SE.sim"])

  set.seed(123)
  pam_result <- pam(combined_distances, k = optimal_clusters)

  data_with_state_boarders$cluster <- pam_result$clustering
```

```
  p <- ggplot(data_with_state_boarders) +
    geom_sf(aes(fill = as.factor(cluster)), color = "black", size = 0.5) +
    labs(
      title = paste("alpha =", alpha, "nclust =", optimal_clusters),
      fill = "Clusters"
    ) +
    theme_minimal()

  plots_list[[paste("alpha_", alpha, sep = "")]] <- p
}
```

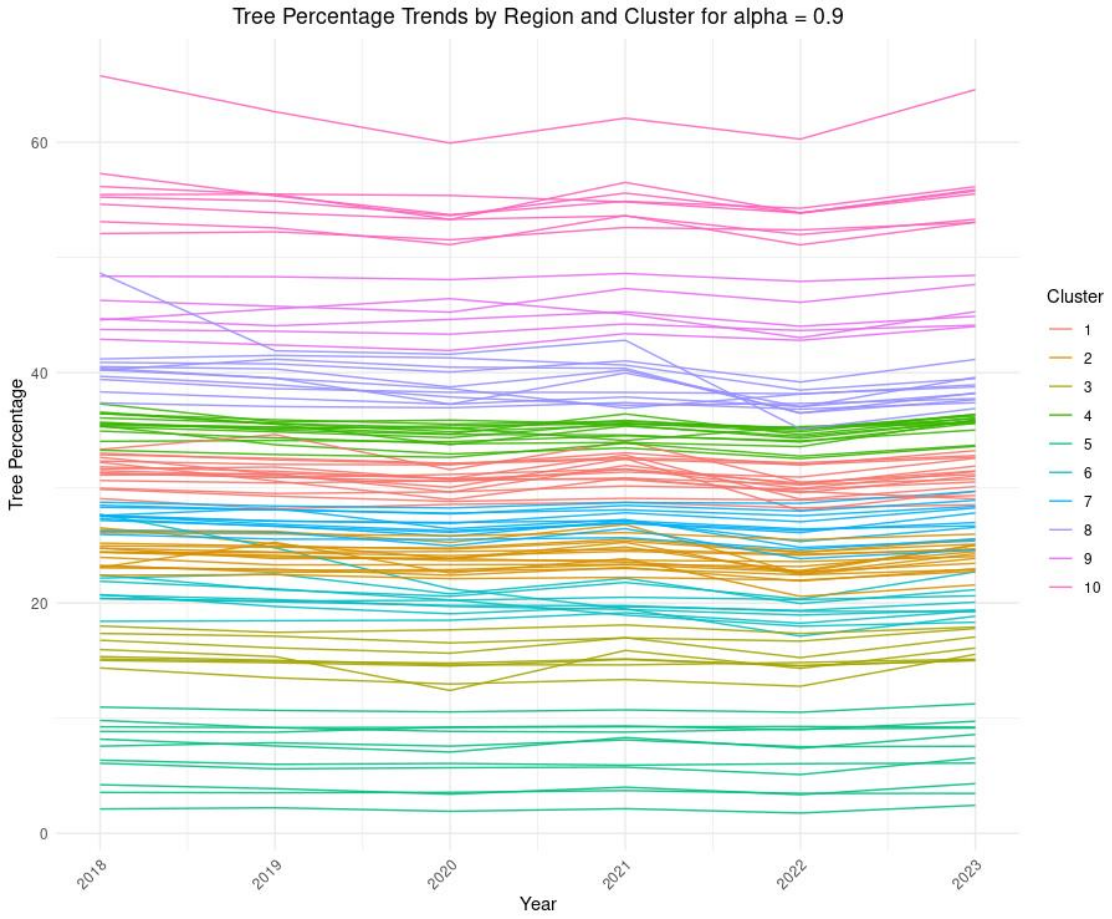


Clustering Results for Varying Alpha Values.

At lower alpha values, such as 0.1 or 0.2, regions are clustered more heavily based on geographic proximity. This is evident as neighboring regions are more likely to share the same cluster, despite having different temporal tree percentage trends. As alpha increases to 0.5 and beyond, the temporal trends in tree percentage change become more dominant. This results in geographically distant regions, such as northern and southern parts of the map, being grouped together because they show similar tree percentage trends over time.

Interestingly, the number of clusters fluctuates with different alpha values. For example, at alpha = 0.4, there are only two clusters, suggesting that either spatial or temporal distances alone are not enough to differentiate regions meaningfully. In contrast, at alpha values like 0.5 or 0.9, we observe 10 distinct clusters, indicating that both spatial and temporal information are contributing more evenly to the clustering, leading to a more detailed segmentation. If we take a look at tree percentage trends by regions for high alpha value, we will observe that regions that

showed mostly identical trends in tree cover change, have been clustered together into clusters that barely have overlapping points.



Tree Percentage Trends Over Years By Cluster for $\alpha = 0.9$.

Including All Feature Variables to Clustering

In this section, we aim to compare the results of **spatio-temporal clustering** based on two distinct approaches: clustering using **only tree percentage** as the temporal variable and clustering using **multiple land-use variables**, including tree percentage, water percentage, crops percentage, built area percentage, and others. By analyzing these two different clustering setups, we can explore how additional land-use features impact the formation of clusters and whether they provide further insights compared to using tree percentage alone.

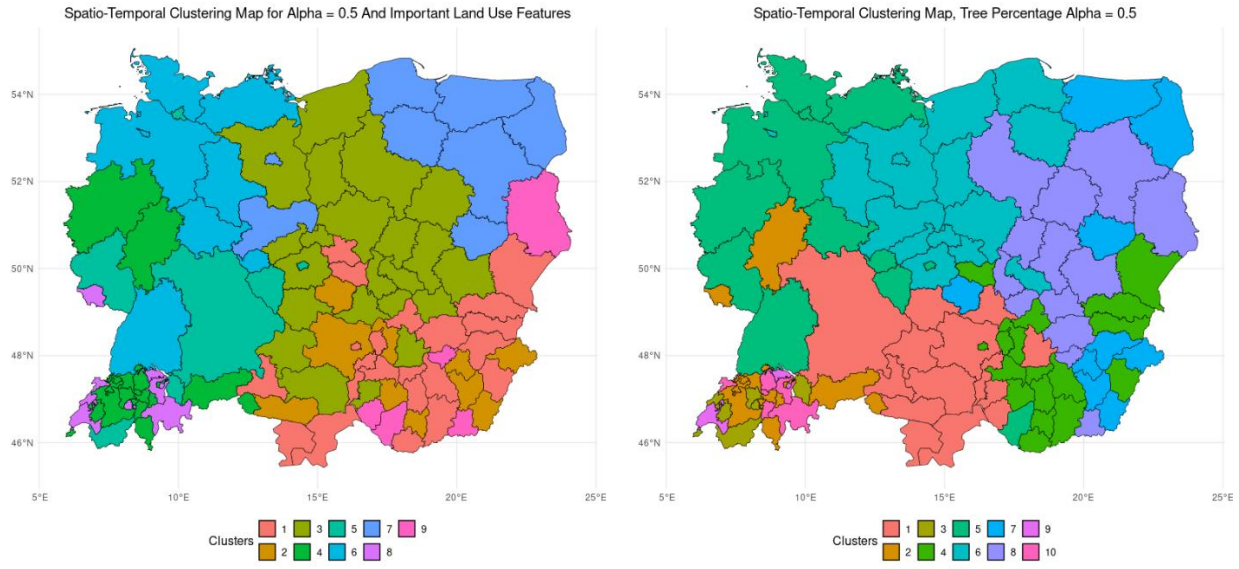
I also experimented with incorporating **pollution variables** (such as CO2 and PM25 levels) into the clustering process. However, as these variables are not consistently present in the data over time and are subject to variability (such as wildfire events), they proved less reliable for our clustering task. As we observed in the **XGBoost feature importance analysis**, pollution variables had almost no impact on the tree percentage prediction. Consequently, including them in the clustering process resulted in less informative clusters, often causing all regions to be grouped into the same cluster, offering no meaningful distinctions.

```
temporal_data <- data %>%
  select(starts_with(c("tree_percentage", "water_percentage", "crops_percentage",
```

```
                    "built_area_percentage", "rangeland_percentage",
                    "bare_ground", "snow_ice", "flooded"))) %>%
  mutate(across(everything(), as.numeric)) %>%
  as.matrix()
```

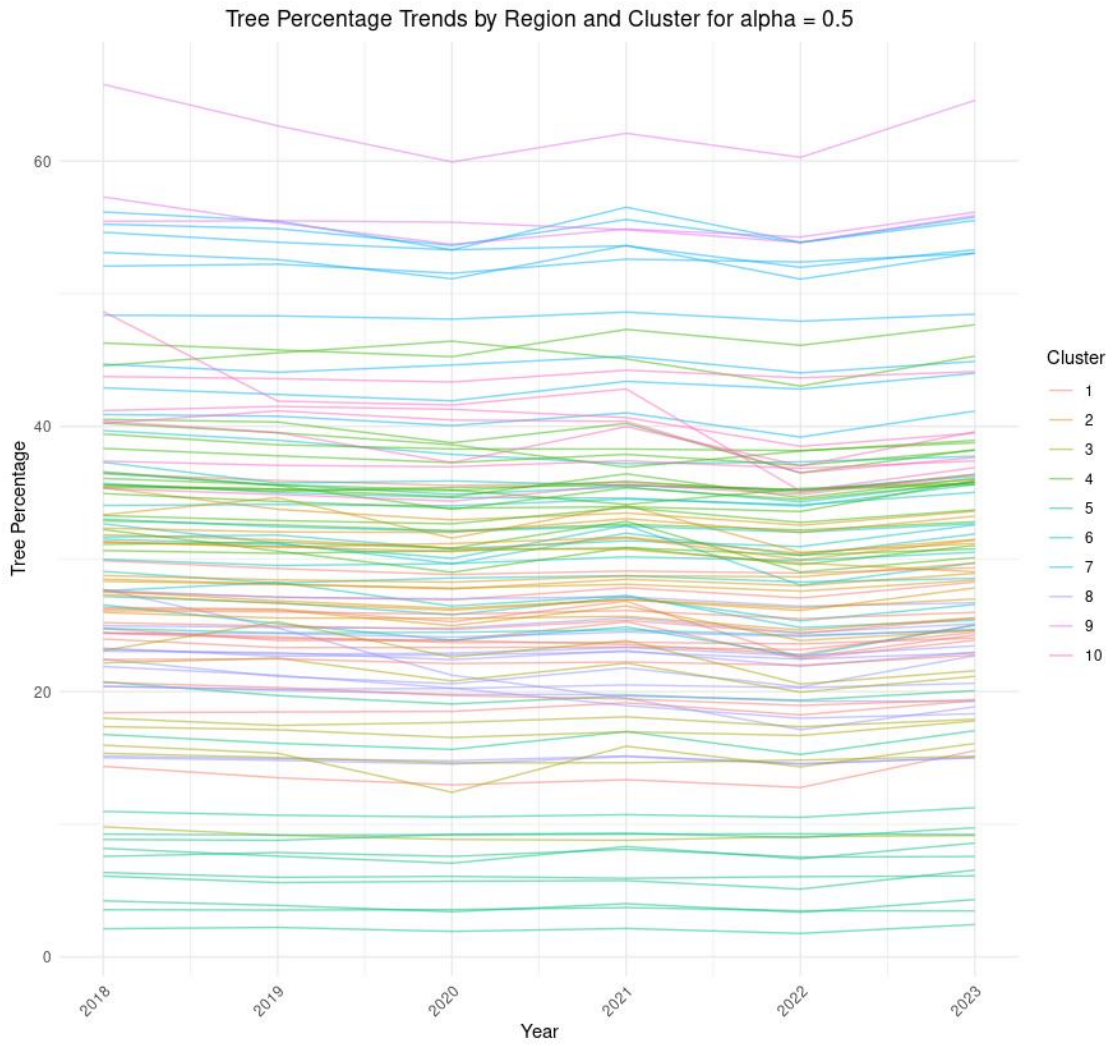
The goal of this comparison is to determine the extent to which tree coverage trends alone can capture spatial and temporal patterns in our data versus the added benefit of incorporating other critical land-use factors. These additional variables play a significant role in the competition for land use, which may directly influence the change in tree cover over time. We will explore clustering with $\alpha = 0.5$ and give temporal and spatial dimensions identical importance.

Spatio-Temporal clustering based on only tree percentage as temporal dimension vs. multiple land-use variables as temporal dimension.



Clustering Comparison

- **Tree Percentage Only**: The regions tend to cluster into larger, contiguous areas. This continuity suggests that neighboring regions with similar tree coverage changes tend to be grouped together. Spatial distribution of regions also contributes to the clustering results attempting to group geographically close regions with similar trends together.

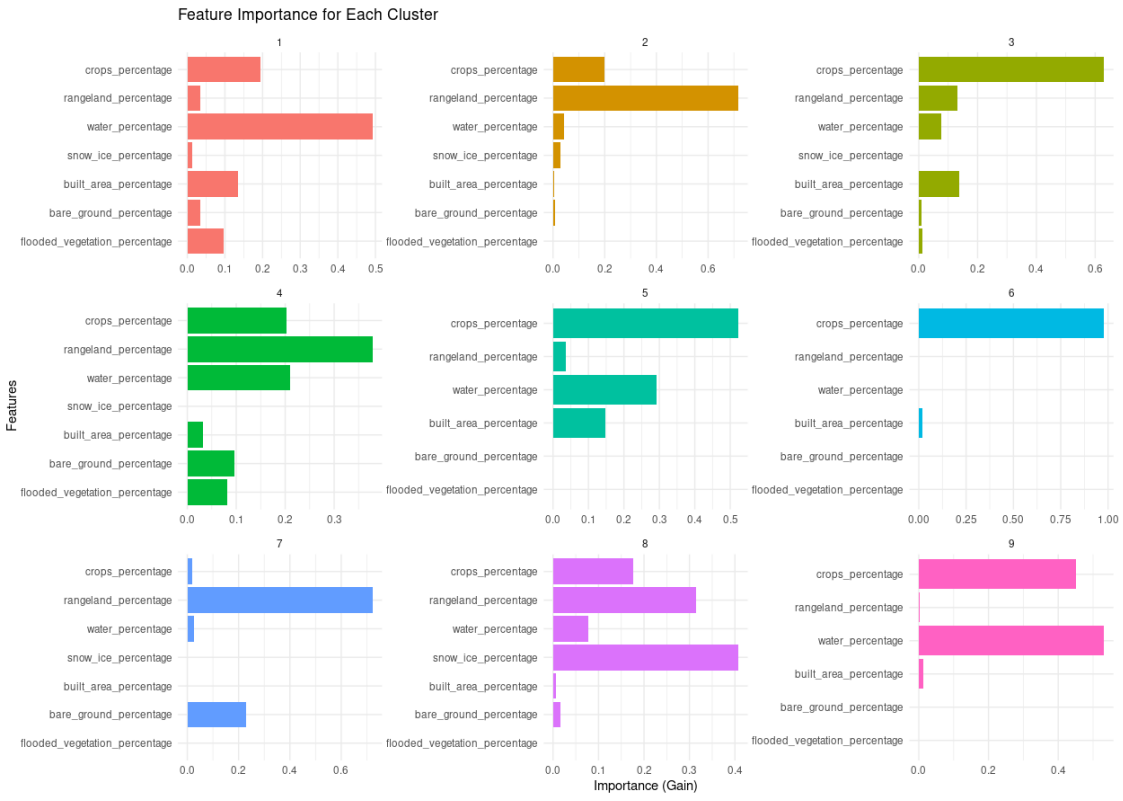


- **Multiple Variables**: The clusters become more **fragmented** and **localized**. This fragmentation likely reflects the more complex land-use dynamics in these regions. Regions where built-up areas and agricultural activities are significant now influence cluster membership, showing that the competition for land use plays a larger role than just changes in tree cover. Interestingly, a smaller cluster emerges in Switzerland, grouping a few regions together, which presents an unusual and unexpected pattern in the analysis and needs further inspection to why this assignment would make sense.

Feature Importance For Clustering

As noted before, small clusters emerge in our data set, taht are mainly surrounded by members of larger clusters and it remained unclear why certain regions were clustered together as the ones in Switzerland, where quite a few neigbouring regions were assigned to different clusters. To investigate this further, we applied the XGBoost algorithm to each cluster individually. The objective was to evaluate the importance of land-use features, such as crop percentage, built area percentage, and water percentage, in determining cluster membership, providing insights into the drivers behind the clustering patterns we observed.

```r
importance_list <- list()
for (clust in unique(data_with_state_boarders$cluster)) {

  subset_data <- time_series_data %>% filter(cluster == clust)

  model <- xgboost(data = model.matrix(~ crops_percentage + built_area_percentage
                                       + water_percentage + rangeland_percentage
                                       + flooded_vegetation_percentage
                                       + bare_ground_percentage +  snow_ice_percentage - 1,
                                       subset_data),
                   label = subset_data$tree_percentage,
                   nrounds = 50, objective = "reg:squarederror", verbose = 0)

  importance <- xgb.importance(feature_names = colnames(model.matrix(~ crops_percentage
                                + built_area_percentage + water_percentage
                                + rangeland_percentage + flooded_vegetation_percentage
                                +bare_ground_percentage +  snow_ice_percentage - 1,
                                subset_data)), model = model)

  importance$Cluster <- as.factor(clust)

  importance_list[[paste("Cluster", clust)]] <- importance
}
```



Feature Importance For Assigning Regions to Clusters

One of the most notable clusters is Cluster 8, which includes regions in Switzerland. The key factors here are *snow_ice_percentage* and *rangeland_percentage*, which strongly influence tree coverage in these regions. This aligns well with the geography of Switzerland, known for its mountainous terrain where rangelands dominate and snow/ice cover is prevalent, especially in

higher altitudes. Crops percentage plays a significant role in Cluster 8, which explains why the western regions of Switzerland and Saarland in Germany were assigned to this cluster. Clusters 2, 4, and 7 are primarily dominated by the rangeland factor, which is expected given their location in mountainous regions across Germany, Austria, and Switzerland. In addition to rangelands, crops also have a strong influence, particularly in Cluster 2, where several Hungarian regions were grouped due to their agricultural activities. On the other hand, Clusters 3, 5, and 6 are primarily driven by crops percentage, with Cluster 3 largely encompassing agricultural land in Poland and Cluster 5 land of same use in Germany. For Cluster 5 it's also important to note that `built_are_percentage` has heavy influence to it, explaining why region of Czech capital Prague which is mainly completely built with crops on city boarders is enclosed by this cluster.

Conclusion and Discussion

In this research project, we explored the spatio-temporal dynamics of sustainability data across Central Europe, focusing on the interplay between tree coverage and other land-use and environmental variables. The journey started with data collection and processing, where we brought together datasets covering tree percentages, water bodies, agricultural land, built areas, and other land cover types. This data was further enriched with information on emissions and pollution levels to provide a holistic view of environmental conditions across the region.

The first stage involved understanding the temporal and spatial trends in tree coverage. We looked at descriptive statistics and created visualizations that highlighted how tree percentages have evolved over time in different regions. This gave a sense of where tree cover was declining and which areas remained relatively stable. As the analysis progressed, I found that many of the trends in tree coverage could be linked to competing land uses such as agriculture, urban expansion, and even natural features like rangelands and snow/ice cover in the mountainous regions.

We then delved into spatio-temporal clustering to uncover deeper patterns. We were able to cluster regions not only based on their spatial proximity but also considering their temporal trends in key environmental features. This approach helped to identify regions that shared similar land-use dynamics over time, revealing clusters of areas facing similar challenges or undergoing similar transformations. The feature importance analysis revealed why certain regions grouped together, such as those in Switzerland where snow and ice coverage, as well as rangelands, played dominant roles. Meanwhile, regions focused on agriculture, particularly in Poland, were shaped primarily by the influence of crops percentage.

However, not all variables contributed meaningfully. When I incorporated pollution and emissions data, the results were less informative, and many regions ended up being grouped into a single large cluster. This was consistent with the earlier finding in the XGBoost analysis, where pollution variables had minimal importance, likely due to the sporadic nature of the data or the lack of consistent emissions records across the years.

There is potential to build on this work by incorporating socioeconomic factors such as population density or economic activity, which would provide a more comprehensive picture of human influence on land use. It would also be beneficial to find better suited polution data and discover its influences on the changes in tree percentage. Future research could refine the clustering approach by testing more advanced algorithms capable of handling the complex relationships between environmental features. There is also room to build predictive models that

forecast how regions might evolve based on current land-use patterns and emerging environmental challenges.

Acknowledgment

I would like to thank my supervisor, Prof. Dr. Laura Vana Gür, for giving me the opportunity to work on this topic and for guiding me through the process of my first scientific research.