Social-ecological Research in Economics (SRE) Discussion Paper 03/2024

The Diffusion of Green Building Certifications in Europe - A preliminary descriptive analysis

by

Gunther Maier, Katarzyna Reyman, Michał Głuszak

Institute for Multi-Level Governance & Development
Department of Socio-Economics
Vienna University of Economics and Business

03 / 2024

WIRTSCHAFTSUNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

The Diffusion of Green Building Certifications in Europe - A preliminary descriptive analysis

by

Gunther Maier, Katarzyna Reyman, Michał Głuszak

Abstract

This paper provides a preliminary descriptive analysis of the spatial and temporal diffusion of green building certifications in Europe. The analysis is preliminary because it does not include all the major green building certification schemas in Europe; just BREEAM and LEED. The main aim of this analysis is to show that there is a strong diffusion process of green building certifications in Europe and that this process has marked spatial and temporal dynamics. The paper also aims to demonstrate that this process can be analyzed at different spatial scales from countries to NUTS3 regions to municipalities.

Keywords:
green buildings; LEED; BREEAM; diffusion;

JEL:

The Diffusion of Green Building Certifications in Europe - A preliminary descriptive analysis

Gunther Maier, Katarzyna Reyman, and Michał Głuszak

1 Introduction

Our motivation for writing this paper is the growing concern and global action to mitigate climate change and the fact that the building sector is responsible for 35% of the EU's energy-related emissions in 2021 and for over 35% of the EU's total waste generation. The green buildings movement aims to reduce the negative impact of the buildings sector on the environment. Eco-buildings use resources effectively, create healthy and comfortable living environment, and live in harmony with the environment (Shi and Liu 2019). Green buildings play an important role in environmental protection and resource conservation (Wang et al. 2018). As sustainable buildings provide economic, environmental and socio-economic benefits, it is important to understand what factors influence their adoption, how they are diffused across cities, regions and countries, and what the future trends of their dissemination may be.

This exploratory analysis of the spatio-temporal diffusion of office eco-buildings in Europe is a step towards a larger project that will continue with the full data and in addition address the key issues outlined above.

This paper uses data from BREEAM (Building Research Establishment Environmental Assessment Methodology) and LEED (Leadership in Energy and Environmental Design) certificates of office buildings in 2006-24 in 33 European countries. The green building certificate is the proxy for its sustainability. The data come from the websites of the certification organisations[1]. We merged data on certificates with data on NUTS3 (source: Eurostat) using the shapefile on NUTS3 boundaries. In addition, we applied data on construction workers in 2018 (Eurostat) as a proxy for the size of the construction sector in NUTS3 regions. To explore the data, we employed descriptive statistics, Theil's concentration index, spatial analysis (Moran's I statistics and local Moran's I) and temporal centres of gravities. To investigate the factors that influence the spatial distribution of eco-buildings across countries, we constructed count models and cross-sectional analysis. This preliminary analysis uses partial data on certificates of green buildings. To get the full picture, data on HQE (Haute Qualité Environnementale) and

[1] https://tools.breeam.com/projects/explore/buildings.jsp and https://www.usgbc.org/projects.

DGNB (Deutsche Gesellschaft für Nachhaltiges Bauen) certificates are needed. We therefore refrain from interpreting the analyses, apart from descriptive statistics.

With this analysis, we intend to demonstrate that this spatio-temporal analysis is feasible with the data currently used. The use of Quarto software allows the results to be shared, reproduced and discussed with a scientific audience. Since we are working on a project proposal at the moment, we currently do not provide access to the green building data we have collected. This will be changed as soon as possible. However, the remaining analytical workflow is fully transparent in this publication.

In our exploratory analysis we proceeded as follows: in the first step, we performed descriptive statistics by country and by year for BREEAM and LEED certificates for the number and stock of observations (green buildings). We estimated the average annual increase in the number of green building certificates. We examined the distribution of certificates across countries and per construction worker. We calculated the Theil concentration index per country and per year. Secondly, we performed descriptive statistics on the merged data at NUTS3 level. We measured spatial autocorrelation using Moran's I statistics and local Moran's I at the level of European regions. By calculating the centroid of all green buildings and estimating growth trajectories, we added the time dimension to the analysis. We then used regression analysis to examine the growth process of certified building development. We fitted three types of model fit curves (exponential, a non-linear cubic, a non-linear logistic) to the number and stock of green buildings for all European countries and for each country separately. We assume that the number of certified green buildings over time follows a growth process that eventually reaches saturation. We checked whether the fitted model curve resembles the s-shaped innovation curve. We then run regression models with the number of green buildings as a dependent variable and countries, countries and year, and interactions. Finally, a cross-sectional analysis with the number of construction workers in each country as a proxy for the size of the construction sector in a NUTS region.

2 Reading all the data

We need the packages `tidyverse`, `eurostat`, `gmodels`, `forcats`, `sf`, `spdep`, `tmap`, and `gt` for these operations. Make sure you have installed them. In the following code chunk we load these packages.

```
library(tidyverse)
library(eurostat)
library(gmodels)
library(forcats)
library(sf)
library(spdep)
library(tmap)
```

```r
library(ggplot2)
library(plotly)
```

In the next sections, we load all the data that we need for our analysis.

2.1 Reading the Green Building data.

The first step is to read the green building data. We construct a URL with the respective query string to extract the correct observations from the database. We then feed the encoded version of the URL to function `read.delim()`, which reads directly from the internet.

Since we are currently working on a project proposal, this step is currently suppressed in the output. We will make it transparently available as soon as possible so that the whole analysis will be perfectly reproducable.

This dataframe has 29864 observations and 10 variables. Since the year 2024 is incomplete, we filter only observations with a certification date before January 1st 2024.

Each observation represents one green building certification. Note that these are **flow** numbers - awarded certifications. They are not the **stock** of certified green buildings.

2.2 Reading Construction Employment data from Eurostat

See https://ropengov.github.io/eurostat/articles/eurostat_tutorial.html for information on downloading data from Eurostat into R. This section uses the R package `eurostat` which is described in the mentioned tutorial.

The Eurostat table_id `sbs_r_nuts06_r2` is for employment data of all sectors in all NUTS2 regions in all years 2008 to 2020.

```r
emp_eurostat_id = "sbs_r_nuts06_r2"
```

We get the full dataset from Eurostat. To get cross-sectional information about employment in the construction sector for as many countries as possible, we filter the data by the following criteria

- the construction sector
- year: 2018 (last year with UK information)
- full countries
- persons employed

We rename the column `geo` to `country` to ease later merging. We also rename "UK" to "GB" as it is used in the green building dataset.

```
constructionEmploymentByCountry <- get_eurostat(emp_eurostat_id,
          time_format = "num", stringsAsFactors = TRUE) %>%
  filter(nace_r2 == "F", indic_sb == "V16110", str_length(geo) == 2,
         TIME_PERIOD == 2018) %>%
  rename(country = geo) %>%
  mutate(country = case_match(country, "UK" ~ "GB", .default = country))
```

2.3 Reading Population data from Eurostat

In a similar way we read population data for all countries from Eurostat. The Eurostat table_id `demo_r_pjangrp3` is for population data in all NUTS3 regions in all years 2008 to 2020. We select the year 2019, the last year with data for the UK. Again, we use the R package `eurostat`.

```
eurostat_id = "demo_r_pjangrp3"
#head(search_eurostat(eurostat_id, column = "code"))
```

```
Population <- get_eurostat(eurostat_id,
          time_format = "num", stringsAsFactors = TRUE) %>%
  filter(str_length(geo) == 5,
         TIME_PERIOD == 2019,
         age == "TOTAL",
         sex == "T")
```

We aggregate the population from the NUTS3 level to countries and store the results in data-frame `PopulationByCountry`. We will use this data-frame later.

```
PopulationByCountry <- Population %>%
  group_by(country = str_sub(geo, 1, 2)) %>%
  summarize(pop = sum(values)) %>%
  mutate(country = case_match(country, "UK" ~ "GB", .default = country)) %>%
  mutate(country = case_match(country, "EL" ~ "GR", .default = country))
```

2.4 Reading the NUTS codes and names

The following code chunk reads the shapefile with the NUTS3 boundaries and extracts the NUTS3 codes and names. It sorts the data-frame by NUTS3 codes. This information is essential for mapping and for the spatial analysis.

The shapefile is read from the local drive. We downloaded this file using the web-tool https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units/territorial-units-statistics.

```
nuts3 <- st_read("NUTS_RG_01M_2016_3857.shp")
nuts3_sf <- st_as_sf(nuts3, coords = c("longitude", "latitude"), crs = 4326)
nuts3_sf <- st_transform(nuts3_sf, crs= 4326)
nuts3_sf <- nuts3_sf %>%
            filter(nchar(NUTS_ID) >= 5)

nuts3 <- as_tibble(nuts3 %>%
            filter(nchar(NUTS_ID) >= 5))
nuts3 <- select(nuts3, "NUTS_ID", "NAME_LATN")
nuts3 <- nuts3[order(nuts3$NUTS_ID),]
```

2.5 Some useful data structures

For later use we also generate vectors of the years and of the country codes as well as a list of the corresponding country names.

```
years <- sort(unique(green$CertYear))
countries <- sort(unique(green$country))
countryNames <- list(
  "AT" = "Austria",        "BE" = "Belgium",         "BG" = "Bulgaria", "CH" = "Switzerland",
  "CY" = "Cyprus",         "CZ" = "Czech Republic", "DE" = "Germany",  "DK" = "Denmark",
  "EE" = "Estonia",        "ES" = "Spain",           "FI" = "Finland",  "FR" = "France",
  "GB" = "Great Britain", "GR" = "Greece",          "HR" = "Croatia",  "HU" = "Hungary",
  "IE" = "Ireland",        "IS" = "Iceland",         "IT" = "Italy",    "LT" = "Lithuania",
  "LU" = "Luxembourg",     "LV" = "Latvia",          "MT" = "Malta",    "NL" = "Netherlands",
  "NO" = "Norway",         "PL" = "Poland",          "PT" = "Portugal", "RO" = "Romania",
  "RS" = "Serbia",         "SE" = "Sweden",          "SI" = "Slovenia", "SK" = "Slovakia",
  "TR" = "Turkije")
```

With all the required data loaded, we can now concentrate on the descriptive analysis of the green building certifications.

3 Descriptive analysis by years and countries

3.1 Grouping the observations

As a first step, we group the data by year and produce a dataframe (`byYear`) with the count of certifications by year. In a second dataframe (`byYearAggr`) we aggregate the number of certifications over the years to get the numbers of the certified green buildings.

```r
byYear <- green %>%
  count(CertYear)

# Calculate the aggregates
byYearAggr = byYear
byYearAggr$nAggr = sapply(byYearAggr$CertYear,
                          function(x){
                            sum(byYearAggr$n[byYearAggr$CertYear <= x])
                          })
```

3.2 Do the numbers of certifications increase over time?

The numbers of certifications by Certification Schema and by Year are:

```r
ct <- CrossTable(green$CertYear, green$Schema, prop.t = F, prop.chisq = F)
```

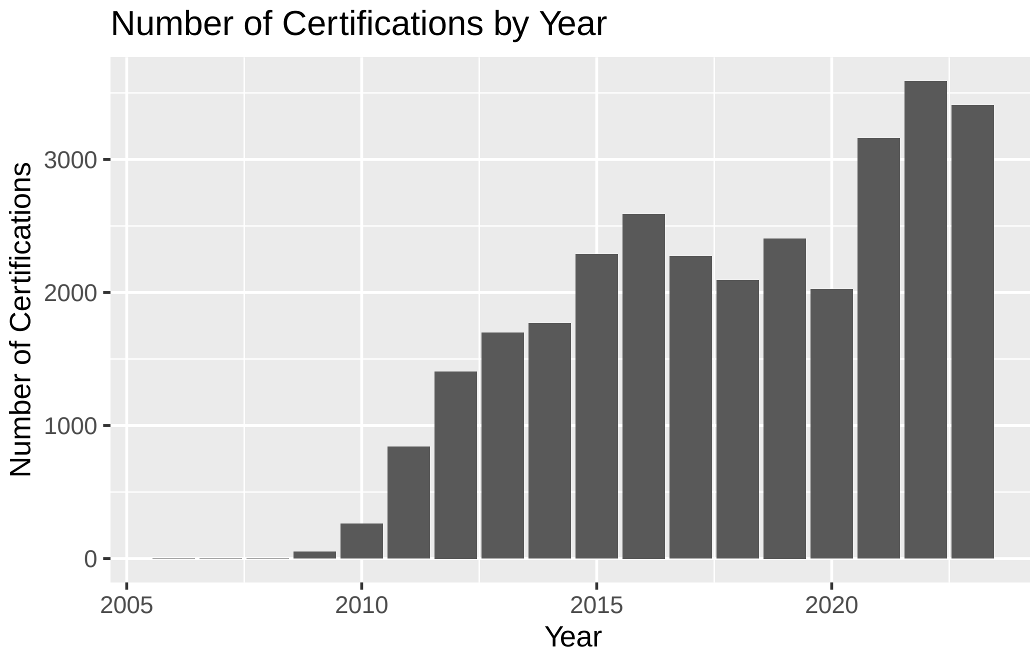
```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|-------------------------|


Total Observations in Table:  29864


               | green$Schema
green$CertYear |    BREEAM |      LEED | Row Total |
---------------|-----------|-----------|-----------|
          2006 |         0 |         2 |         2 |
               |     0.000 |     1.000 |     0.000 |
               |     0.000 |     0.001 |           |
---------------|-----------|-----------|-----------|
          2007 |         0 |         1 |         1 |
               |     0.000 |     1.000 |     0.000 |
               |     0.000 |     0.000 |           |
---------------|-----------|-----------|-----------|
          2008 |         0 |         1 |         1 |
```

```
               |     0.000 |     1.000 |     0.000 |
               |     0.000 |     0.000 |           |
---------------|-----------|-----------|-----------|
          2009 |        42 |        10 |        52 |
               |     0.808 |     0.192 |     0.002 |
               |     0.002 |     0.003 |           |
---------------|-----------|-----------|-----------|
          2010 |       240 |        22 |       262 |
               |     0.916 |     0.084 |     0.009 |
               |     0.009 |     0.008 |           |
---------------|-----------|-----------|-----------|
          2011 |       773 |        66 |       839 |
               |     0.921 |     0.079 |     0.028 |
               |     0.029 |     0.023 |           |
---------------|-----------|-----------|-----------|
          2012 |      1307 |        99 |      1406 |
               |     0.930 |     0.070 |     0.047 |
               |     0.048 |     0.034 |           |
---------------|-----------|-----------|-----------|
          2013 |      1552 |       146 |      1698 |
               |     0.914 |     0.086 |     0.057 |
               |     0.058 |     0.051 |           |
---------------|-----------|-----------|-----------|
          2014 |      1578 |       190 |      1768 |
               |     0.893 |     0.107 |     0.059 |
               |     0.058 |     0.066 |           |
---------------|-----------|-----------|-----------|
          2015 |      2069 |       219 |      2288 |
               |     0.904 |     0.096 |     0.077 |
               |     0.077 |     0.076 |           |
---------------|-----------|-----------|-----------|
          2016 |      2383 |       207 |      2590 |
               |     0.920 |     0.080 |     0.087 |
               |     0.088 |     0.072 |           |
---------------|-----------|-----------|-----------|
          2017 |      2007 |       266 |      2273 |
               |     0.883 |     0.117 |     0.076 |
               |     0.074 |     0.092 |           |
---------------|-----------|-----------|-----------|
          2018 |      1794 |       299 |      2093 |
               |     0.857 |     0.143 |     0.070 |
               |     0.066 |     0.104 |           |
---------------|-----------|-----------|-----------|
```

```
           2019 |      2128 |       278 |      2406 |
                |     0.884 |     0.116 |     0.081 |
                |     0.079 |     0.096 |           |
----------------|-----------|-----------|-----------|
           2020 |      1736 |       290 |      2026 |
                |     0.857 |     0.143 |     0.068 |
                |     0.064 |     0.101 |           |
----------------|-----------|-----------|-----------|
           2021 |      2870 |       291 |      3161 |
                |     0.908 |     0.092 |     0.106 |
                |     0.106 |     0.101 |           |
----------------|-----------|-----------|-----------|
           2022 |      3386 |       203 |      3589 |
                |     0.943 |     0.057 |     0.120 |
                |     0.125 |     0.070 |           |
----------------|-----------|-----------|-----------|
           2023 |      3116 |       293 |      3409 |
                |     0.914 |     0.086 |     0.114 |
                |     0.115 |     0.102 |           |
----------------|-----------|-----------|-----------|
   Column Total |     26981 |      2883 |     29864 |
                |     0.903 |     0.097 |           |
----------------|-----------|-----------|-----------|
```

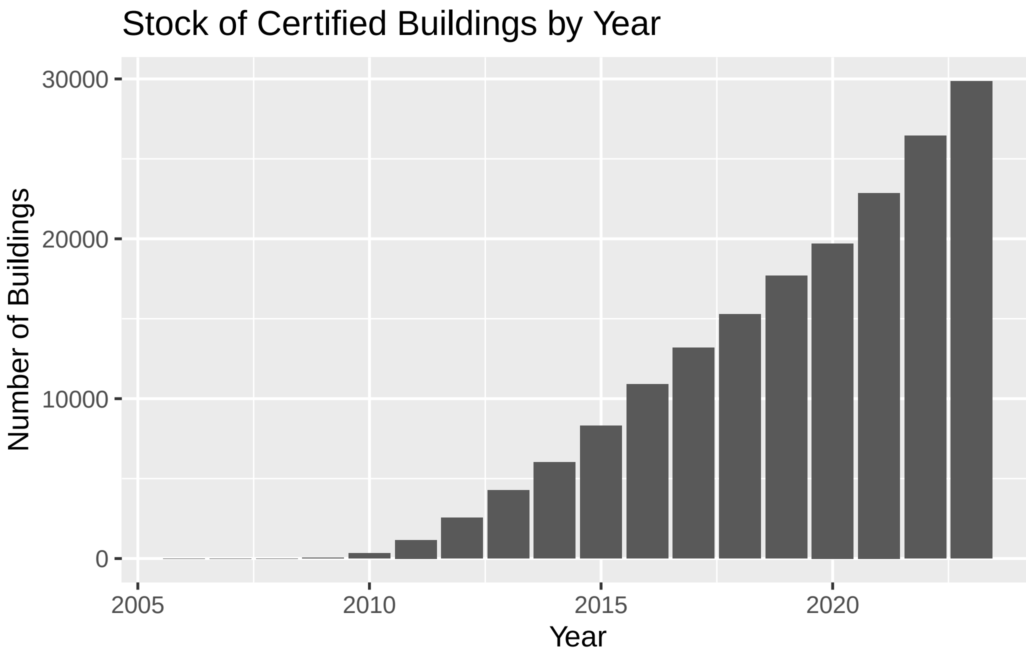
When we plot the number of certifications by year, we see a clear increase.

```
byYear %>%
  ggplot(aes(x=CertYear, y=n)) +
    geom_col() +
    labs(title="Number of Certifications by Year",
         x="Year", y = "Number of Certifications")
```



Since the certified buildings accumulate over the years, the increase in the number of certified green buildings (the "Stock"), is much more pronounces.

```
byYearAggr %>%
  ggplot(aes(x=CertYear, y=nAggr)) +
    geom_col() +
    labs(title="Stock of Certified Buildings by Year",
         x="Year", y = "Number of Buildings")
```



As the above table shows, the increase in the number of certifications is to a large extent due to the certifications by BREEAM, which account for over 90% of all certifications. To estimate the average annual increase in the number of green building certifications and to check for statistical significance, we run a log-linear regression of the number of certifications by year.

```
reg <- lm(log(n)~CertYear, data=green %>%
  count(CertYear))
(lm <- summary(reg))
```

```
Call:
lm(formula = log(n) ~ CertYear, data = green %>% count(CertYear))

Residuals:
    Min      1Q  Median      3Q     Max 
-3.2548 -1.1278  0.1339  1.4311  2.2409 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -876.64741  160.48476  -5.462 5.21e-05 ***
CertYear       0.43820    0.07966   5.501 4.84e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.754 on 16 degrees of freedom
Multiple R-squared:  0.6541,    Adjusted R-squared:  0.6325
F-statistic: 30.26 on 1 and 16 DF,  p-value: 4.84e-05
```

The estimation shows an average annual increase of 43.82%. With a t-value of 5.5, this relation is highly significant. The lower and upper bounds of the 95%-confidence interval are 26.93 and 60.71, respectively.

3.3 How do certifications distribute over countries?

In this section, we do the same analysis across countries. First, we tabulate the numbers and shares of certifications by Certification Schema and Country:

```r
ct <- CrossTable(green$country, green$Schema, prop.t = F, prop.chisq = F)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|-------------------------|


Total Observations in Table:  29864


               | green$Schema
green$country |    BREEAM |      LEED | Row Total |
--------------|-----------|-----------|-----------|
           AT |        12 |        39 |        51 |
              |     0.235 |     0.765 |     0.002 |
              |     0.000 |     0.014 |           |
--------------|-----------|-----------|-----------|
           BE |       327 |        18 |       345 |
              |     0.948 |     0.052 |     0.012 |
              |     0.012 |     0.006 |           |
--------------|-----------|-----------|-----------|
           BG |        36 |        14 |        50 |
```

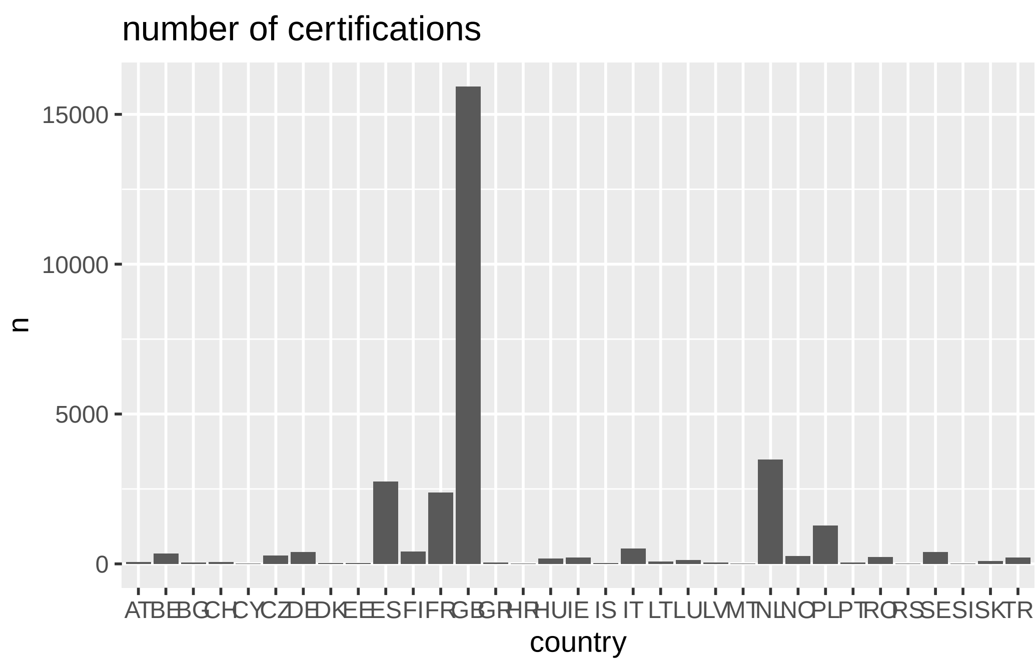
	0.720	0.280	0.002
	0.001	0.005	
CH	9	56	65
	0.138	0.862	0.002
	0.000	0.019	
CY	1	0	1
	1.000	0.000	0.000
	0.000	0.000	
CZ	201	66	267
	0.753	0.247	0.009
	0.007	0.023	
DE	53	335	388
	0.137	0.863	0.013
	0.002	0.116	
DK	10	17	27
	0.370	0.630	0.001
	0.000	0.006	
EE	3	16	19
	0.158	0.842	0.001
	0.000	0.006	
ES	2169	568	2737
	0.792	0.208	0.092
	0.080	0.197	
FI	180	228	408
	0.441	0.559	0.014
	0.007	0.079	
FR	2316	63	2379
	0.974	0.026	0.080
	0.086	0.022	
GB	15820	112	15932
	0.993	0.007	0.533
	0.586	0.039	

GR	6	28	34
	0.176	0.824	0.001
	0.000	0.010	
HR	1	5	6
	0.167	0.833	0.000
	0.000	0.002	
HU	123	59	182
	0.676	0.324	0.006
	0.005	0.020	
IE	88	125	213
	0.413	0.587	0.007
	0.003	0.043	
IS	29	0	29
	1.000	0.000	0.001
	0.001	0.000	
IT	77	434	511
	0.151	0.849	0.017
	0.003	0.151	
LT	65	18	83
	0.783	0.217	0.003
	0.002	0.006	
LU	127	2	129
	0.984	0.016	0.004
	0.005	0.001	
LV	33	3	36
	0.917	0.083	0.001
	0.001	0.001	
MT	2	0	2
	1.000	0.000	0.000
	0.000	0.000	
NL	3442	33	3475
	0.991	0.009	0.116
	0.128	0.011	

NO	248	9	257
	0.965	0.035	0.009
	0.009	0.003	
PL	1048	232	1280
	0.819	0.181	0.043
	0.039	0.080	
PT	31	10	41
	0.756	0.244	0.001
	0.001	0.003	
RO	174	50	224
	0.777	0.223	0.008
	0.006	0.017	
RS	2	0	2
	1.000	0.000	0.000
	0.000	0.000	
SE	241	156	397
	0.607	0.393	0.013
	0.009	0.054	
SI	3	1	4
	0.750	0.250	0.000
	0.000	0.000	
SK	67	22	89
	0.753	0.247	0.003
	0.002	0.008	
TR	37	164	201
	0.184	0.816	0.007
	0.001	0.057	
Column Total	26981	2883	29864
	0.903	0.097	

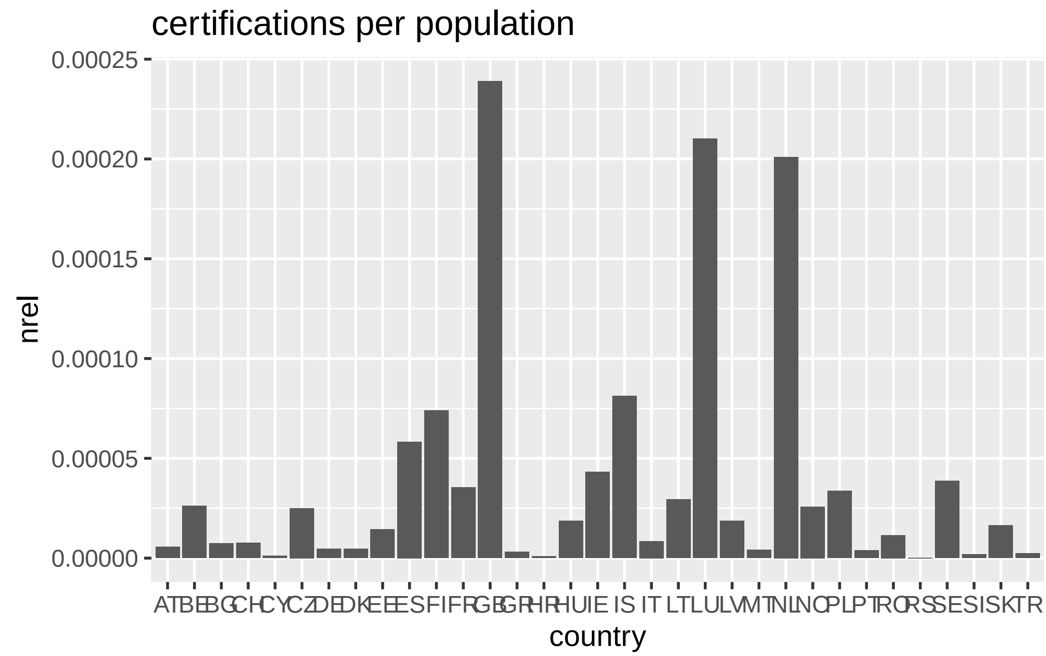
Then, we plot the total number of certifications, which is equal to the number of certified buildings, by countries.

```
byCountryTotals <- green %>%
  count(country)

green %>%
  count(country) %>%
  ggplot(aes(x=country,y=n)) +
  geom_col() +
  ggtitle("number of certifications")
```



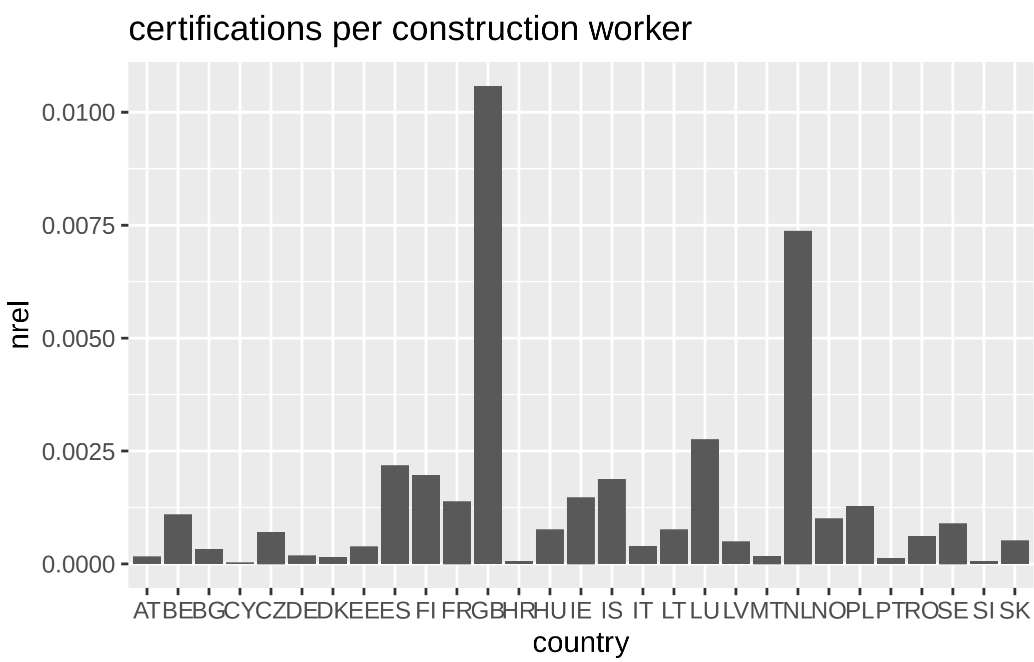
We see the strong dominance of the UK. Since the countries in Europe are of very different size, we set them in relation to some size indicators. First, we plot the numbers of certifications relative to the coutry population in 2019.

```
green %>%
  count(country) %>%
  merge(PopulationByCountry, by = "country") %>%
  mutate(nrel = n/pop) %>%
  ggplot(aes(x=country,y=nrel)) +
  geom_col() +
  ggtitle("certifications per population")
```



Then, we set the number of certifications in relation to the number of employees in construction in 2018.

```
green %>%
  count(country) %>%
  merge(constructionEmploymentByCountry, by = "country") %>%
  mutate(nrel = n/values) %>%
  ggplot(aes(x=country,y=nrel)) +
  geom_col() +
  ggtitle("certifications per construction worker")
```



3.4 Are the numbers of certifications concentrated in certain countries?

To see how strong the concentration of certifications is over countries, we calculate concentration indices (Theil indices) for the numbers of certifications by LEED and BREEAM and by country. These indices range from 0 (most concentrated) to 3.4965 (evenly distributed over 33 countries). Note that smaller values mean a stronger concentration. Code listing Listing 1 implements the computation of the Theil indices.

Listing 1 Computation of Theil indices

```
theil <- function(x) {
  x <- x/sum(x)
  y <- x*log(x)
  y[is.nan(y)] <- 0
  return(round(-1*sum(y),5))
}
```

The Theil indices for the certification schemas and for the total areas follows:

```
TheilTable <- function() {
  cat("Theil indices by Certification Schema\n\n")
```

```r
  cat("  Certification Schema   Theil index\n")
  cat("  ====================   ===========\n")
  cat(paste0("  BREEAM","                    ",
             round(theil(ct$prop.col[,"BREEAM"]),3),"\n"))
  cat(paste0("  LEED","                      ",
             round(theil(ct$prop.col[,"LEED"]),3),"\n"))
  cat("  --------------------   -----------\n")
  cat(paste0("  TOTAL","                     ",
             round(theil(byCountryTotals$n),3),"\n"))
}
TheilTable()
```

```
Theil indices by Certification Schema

  Certification Schema   Theil index
  ====================   ===========
  BREEAM                    1.54
  LEED                      2.642
  --------------------   -----------
  TOTAL                     1.794
```

All three indices show a considerable level of concentration across European countries. The concentration of the total figure is strongly attributable to the cross-country concentration of the BREEAM certifications.

To see where the certifications are concentrating, We list all the countries with a share of more than 5% of the respective Certification Schema. For BREEAM we get (in percent):

```r
round(ct$prop.col[,"BREEAM"][ct$prop.col[,"BREEAM"]>0.05]*100,3)
```

```
    ES     FR     GB     NL
 8.039  8.584 58.634 12.757
```

For LEED, the corresponding list of countries is (in percent):

```r
round(ct$prop.col[,"LEED"][ct$prop.col[,"LEED"]>0.05]*100,3)
```

```
    DE     ES     FI     IT     PL     SE     TR
11.620 19.702  7.908 15.054  8.047  5.411  5.689
```

As the indices show, BREEAM is considerably more concentrated. 58.6% of the BREEAM certifications are in GB. Besides GB, only NL, ES, and FR reach shares of over 5%. For LEED, the highest share in Europe is in ES (19.4%). Shares above 5% are also found in DE, FI, IT, PL, SE, TR.

4 Analysis by NUTS3 regions

In this chapter, we analyze the distribution of LEED and BREEAM certified green buildings at a regional level; by the NUTS3 regions of Europe. After some data preparation we first characterize the regional distribution of green buildings. We will see that some regions contain a considerable number of green buildings, while almost half the NUTS3 regions do not contain any green buildings at all. Then, we again calculate Theil indices, this time with NUTS3 regions as units of observation. Since we again find substantial concentration of green buildings, we then calculate spatial indices, which tell us how spatially concentrated the pattern is. We calculate Moran's I indices and local Moran indicators. According to this analysis, the distribution of green buildings over NUTS3 regions shows significant spatial autocorrelation. This means that NUTS3 regions with many (few) green buildings tend to be neighboring regions with many (few) green buildings.

4.1 Preparatory steps

In this section we generate the count data by NUTS3 region and by year. The aim is to calculate concentration measures (Theil-indices) across the NUTS3 regions for each of the years. This is a non-spatial concentration measure.

We aggregate the count of buildings by NUTS3 region. First, we create a data-frame with the total number (`n`).

```
NutsCount <- green %>%
  count(NUTS) %>%
  filter(nchar(NUTS)==5)
```

Then, we generate the counts year by year (2006-2023) and merge the result to the data-frame. The following function does all the necessary calculations. It gets the year and the data-frame `NutsCount` as input and returns the data-frame with the new column (number of certifications in this year in each of the NUTS3 regions).

```
countsByYear <- function(yr, NutsCount) {
  yrname <- paste0("n", yr)

  NutsCountTemp <- green %>%
```

```r
    filter(CertYear == yr) %>%
    count(NUTS, name = "m") %>%
    filter(nchar(NUTS)==5)
  NutsCount <- merge(NutsCount, NutsCountTemp, by.x="NUTS",
                     by.y="NUTS", all = T)
  NutsCount$m[is.na(NutsCount$m)] <- 0
  NutsCount <- NutsCount %>%
    rename({{yrname}} := m)
  return(NutsCount)
}
```

In the following code chunk we add year after year to the data-frame.

```r
for (y in years) {
  NutsCount <- countsByYear(y, NutsCount)
}
```

Then, we create the stock of green buildings by adding the respective flow variables.

```r
NutsCount <- NutsCount %>%
  mutate(
    a2006 = n2006,
    a2007 = a2006 + n2007,
    a2008 = a2007 + n2008,
    a2009 = a2008 + n2009,
    a2010 = a2009 + n2010,
    a2011 = a2010 + n2011,
    a2012 = a2011 + n2012,
    a2013 = a2012 + n2013,
    a2014 = a2013 + n2014,
    a2015 = a2014 + n2015,
    a2016 = a2015 + n2016,
    a2017 = a2016 + n2017,
    a2018 = a2017 + n2018,
    a2019 = a2018 + n2019,
    a2020 = a2019 + n2020,
    a2021 = a2020 + n2021,
    a2022 = a2021 + n2022,
    a2023 = a2022 + n2023
  )
```

Now we merge the list of NUTS3 codes and names with the NutsCounts. We set parameter `all` to `TRUE` so that all regions stay in the data-frame. For regions without green buildings the respective values are set to `NA`. Then, we replace all the `NA`s with zeros.

```
nutsFull <- merge(nuts3, NutsCount, by.x="NUTS_ID", by.y = "NUTS", all = TRUE)
nutsFull$n[is.na(nutsFull$n)] <- 0
for (y in years) {
  n <- paste0("n",y)
  a <- paste0("a",y)
  nutsFull[n][is.na(nutsFull[n])] <- 0
  nutsFull[a][is.na(nutsFull[a])] <- 0
}
```

At the end we merge the flow and stock counts to the data-frame `nuts3_sf` so that we can use these numbers in maps.

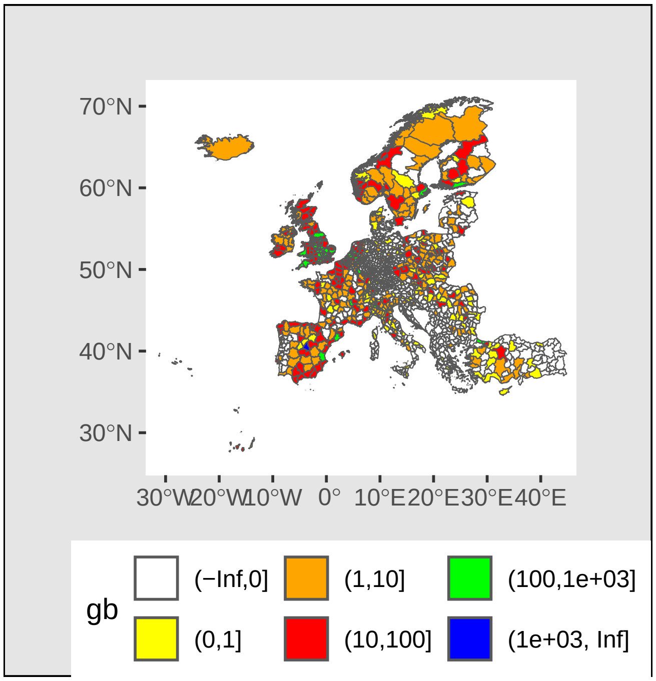
```
nuts3_sf <- merge(nuts3_sf, nutsFull, by.x="NUTS_ID", by.y="NUTS_ID")
```

4.2 Descriptive Statistics by NUTS3

There are 768 NUTS3 regions with at least one Green Building. In the data-frame `nutsFull` there are 1522 observations. Thus there are 754 regions without any certified green building.

The following map shows all the NUTS3 regions color coded by their number of green buildings. All the white regions do not contain any LEED or BREEAM certified green building at all. The pattern we see in this map, in part results from the limited number of certification schemas that we use.

```
p <- nuts3_sf %>%
  mutate(gb = cut(nuts3_sf$a2023, breaks=c(-Inf, 0, 1, 10, 100, 1000, Inf))) %>%
  ggplot() +
    geom_sf(aes(fill=gb)) +
    xlim(-30,  43) +
    ylim(27, 71) +
    scale_fill_manual(values=c("white","yellow","orange","red","green","blue"))+
    theme(legend.position = "bottom",
          plot.margin = margin(1, 1, 0, 1, "cm"),
          panel.background = element_rect(fill = "white"),
          plot.background = element_rect(
            fill = "grey90",
            colour = "black"
          )
    )
#ggplotly(p)
p
```



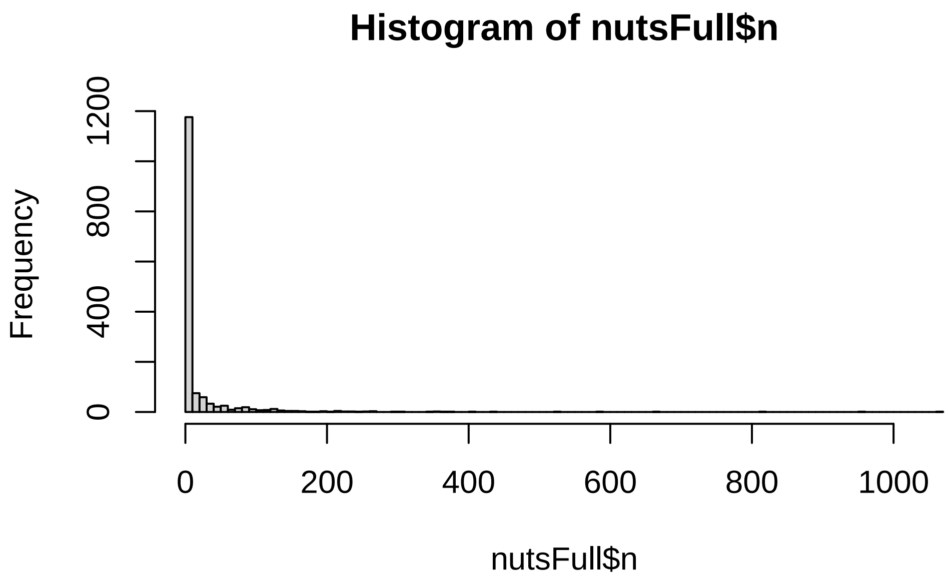
The descriptive statistics of the total numbers of green buildings per region show that we have a power-law type distribution with few regions with large numbers and many with very small numbers. While in average there are 19.62 certified green buildings per region, more than half of the regions have just one green building or none. Three quarters of the region have 8 such buildings or fewer.

```
summary(nutsFull$n)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00    0.00    1.00   19.62    8.00 1065.00 
```

The histogram illustrates this situation clearly.

```
hist(nutsFull$n, breaks = 100)
```



At the top end, there are just 10 NUTS3 regions with more than 300 certified green buildings. Only Madrid, the Spanish capital, has over 1,000 green buildings certified by LEED or BREEAM. In general, we see most top European cities represented in this list: Madrid, London, Amsterdam, Paris, Barcelona, etc.

```r
nutsFull %>%
  filter(n > 300) %>%
  select(NUTS_ID, NAME_LATN, n) %>%
  arrange(-n)
```

```
   NUTS_ID                 NAME_LATN    n
1    ES300                    Madrid 1065
2    UKI32               Westminster  953
3    NL329            Groot-Amsterdam  812
4    UKI31 Camden and City of London  665
5    ES511                 Barcelona  590
6    FR101                     Paris  528
7    NL310                   Utrecht  431
8    UKJ11                 Berkshire  410
9    FR105            Hauts-de-Seine  376
10   PL911           Miasto Warszawa  366
11   UKI44     Lewisham and Southwark  354
12   UKI41         Hackney and Newham  353
```

```
13   UKI42              Tower Hamlets  341
14   NL33C             Groot-Rijnmond  309
```

4.3 Theil indices across NUTS3

Now we have a data-frame with all NUTS3 by all certified buildings (`n`), certified buildings per year (`nyyyy`) and the stock of green buildings per year (`ayyyy`). This allows us to calculate Theil-indices for the years (lower values mean stronger concentration). Since there are 1522 NUTS3 regions in the dataset, the maximum Theil value in the case of equal distribution is 7.3277805.

```
printTheil <- function() {
  cat("Theil concentration indices\n\n")
  cat("  YEAR    FLOWS    STOCK\n")
  cat("  ----    -----    -----\n")
  for (x in years) {
    n <- paste0("n",x)
    a <- paste0("a",x)
    cat(paste0("  ",x,": ",
        format(theil(nutsFull[[n]]), width=7, digits=3, nsmall=3),"  ",
        format(theil(nutsFull[[a]]), width=7, digits=3, nsmall=3),"\n"))
  }
}

printTheil()
```

```
Theil concentration indices

  YEAR    FLOWS    STOCK
  ----    -----    -----
  2006:   0.693    0.693
  2007:   0.000    1.099
  2008:   0.000    1.386
  2009:   3.451    3.561
  2010:   4.658    4.757
  2011:   5.011    5.091
  2012:   5.110    5.200
  2013:   5.187    5.278
  2014:   5.260    5.335
  2015:   5.209    5.356
  2016:   5.185    5.358
```

```
2017:   5.194    5.375
2018:   5.167    5.384
2019:   5.240    5.407
2020:   5.328    5.442
2021:   5.159    5.470
2022:   4.930    5.475
2023:   5.070    5.469
```

With the exception of the year 2023, the Theil index values of the stock increase year after year. This means that with the growing stock of Green Buildings they are becoming less concentrated. As the descriptive statistics by NUTS3 show, also by the end of the observation period the distribution of green buildings over NUTS3 regions is strongly concentrated.

4.4 Moran's I Analysis

This observed concentration is measured across the NUTS3 regions, irrespective of their location. The regions with a large number of green buildings could be spread randomly across Europe or cluster in a certain small area. To check the spatial relation between NUTS3 regions with many green buildings, we turn to Moran's I analysis.

Moran's I analysis takes into account the spatial configuration of Europe's NUTS3 regions. From the shapefile we extract a neighborhood matrix that indicates which NUTS3 regions have a common border. Based on this neighborhood matrix, we can then calculate for every NUTS3 region the average number of green buildings in its neighboring regions. This generates the spatially lagged value. The Moran's I statistic shows the relationship between the values and the respective spatially lagged values across all NUTS3 regions. A significantly positive value shows that regions with many (few) green buildings tend to be neighboring regions with many (few) green buildings.

First, we prepare the spatial weight matrix.

```
nb <- poly2nb(nuts3_sf, queen=TRUE)
lw <- nb2listw(nb, style="W", zero.policy=TRUE)
```

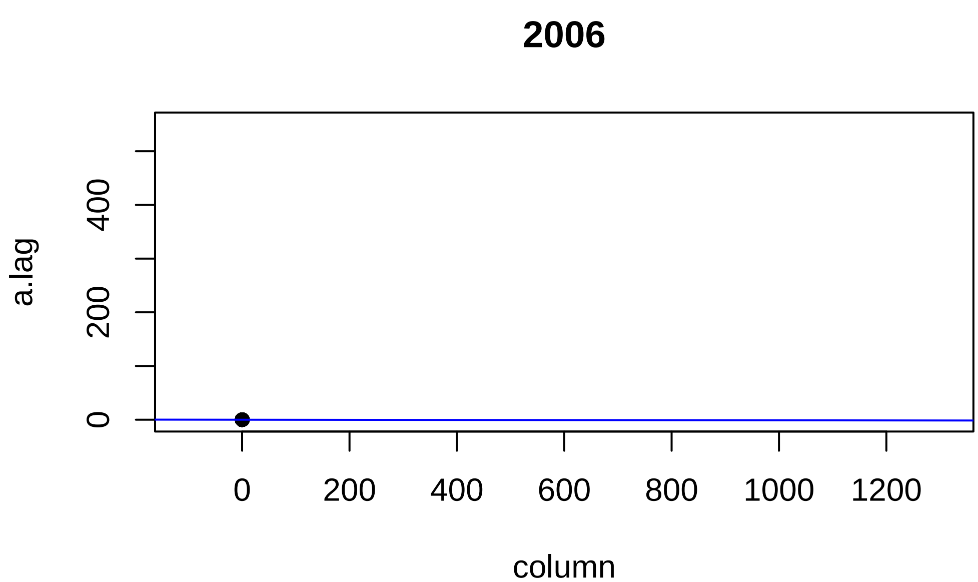
Then, we define a function that does all the relevant calculations: it computes the spatially lagged values and plots them against the raw values. Then, it computes Moran's I, and runs the moran test to find whether or not the relation is statistically significant.

```
doMoran <- function(yr, column) {
  a.lag <- lag.listw(lw, column)
  plot(a.lag ~ column, pch=16, asp=1, main=yr, xlim=c(0,1200), ylim=c(0,550))
  M1 <- lm(a.lag ~ column)
```

```
  abline(M1, col="blue")
  I <- moran(column, lw, length(nb), Szero(lw))[1]
  cat(paste("YEAR: ", yr),"\n")
  cat(paste("NUTS3 regions with Green Building(s):",
    length(which(column != 0))),"\n")
  print(moran.test(column, lw, alternative = "greater"))
}
```

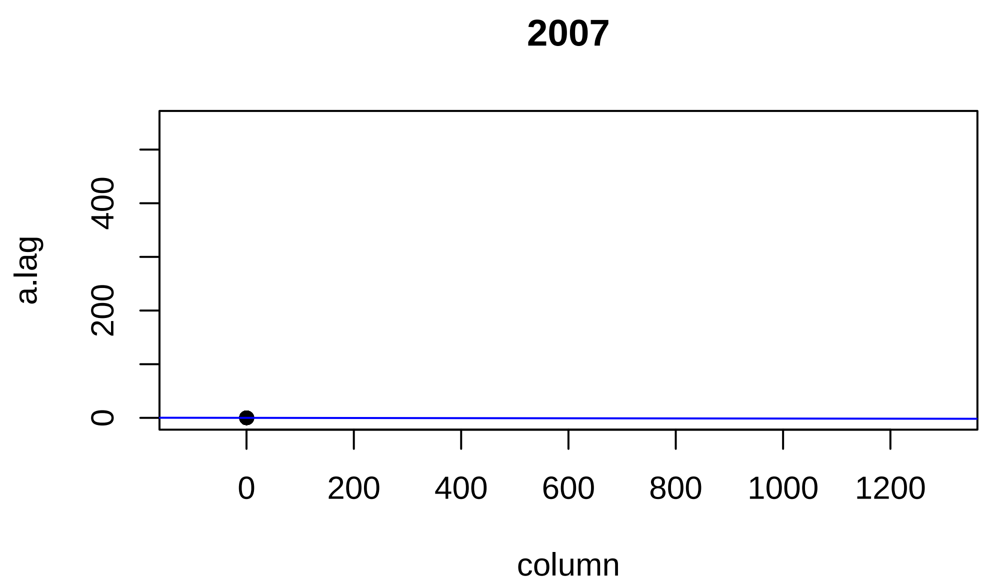
Now, we run the Moran functions for each of the years. To be able to compare the scatterplots over the years, we have fixed their axes to the identical limits.

```
for (y in years) {
  a <- paste0("a", y)
  doMoran(y, nuts3_sf[[a]])
}
```

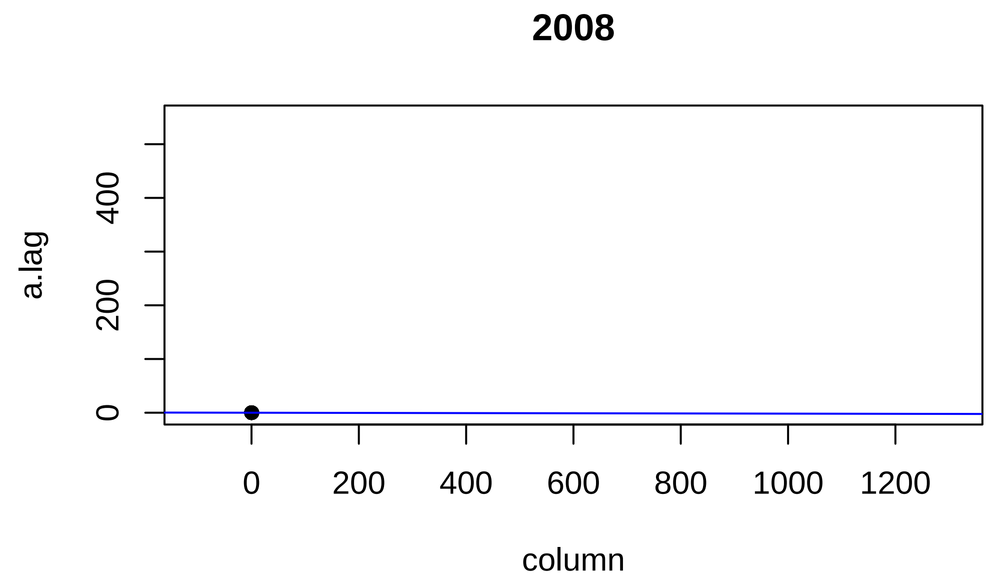



```
YEAR:  2006
NUTS3 regions with Green Building(s): 2

    Moran I test under randomisation

data:  column
```

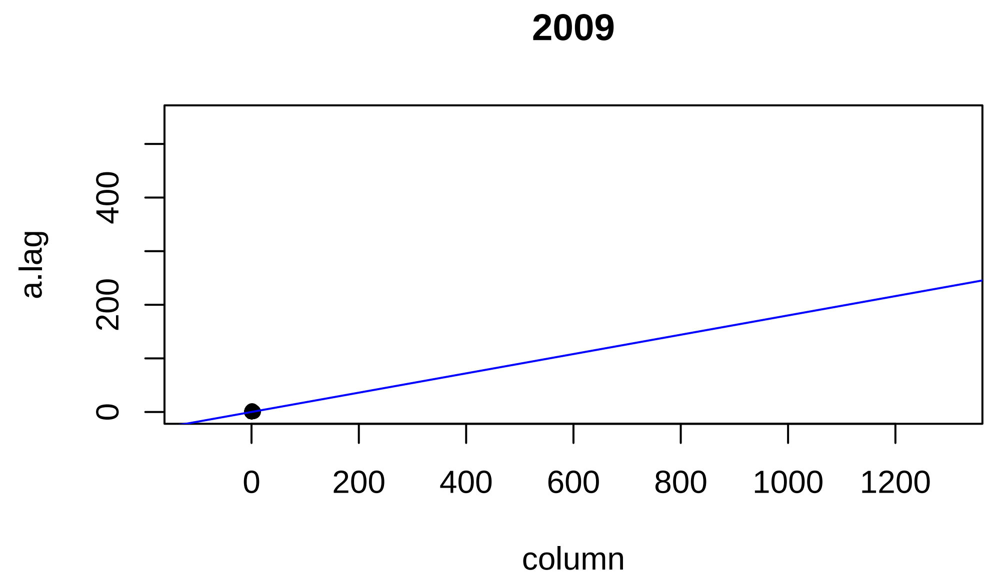
```
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = -0.037772, p-value = 0.5151
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
    -0.0011318550     -0.0006752194      0.0001461467
```



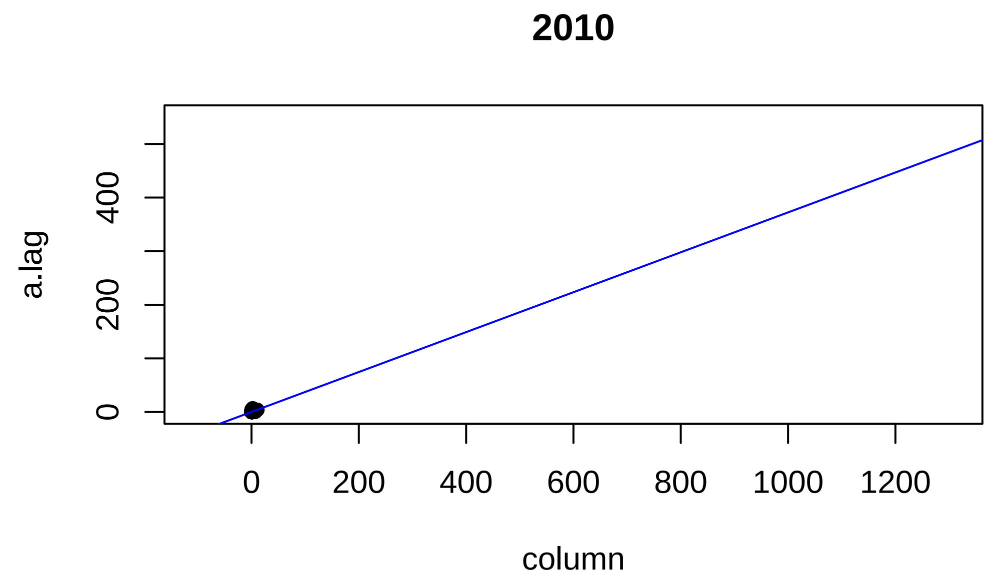

```
YEAR:  2007
NUTS3 regions with Green Building(s): 3

    Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = -0.046265, p-value = 0.5185
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
    -0.0013254559     -0.0006752194      0.0001975327
```



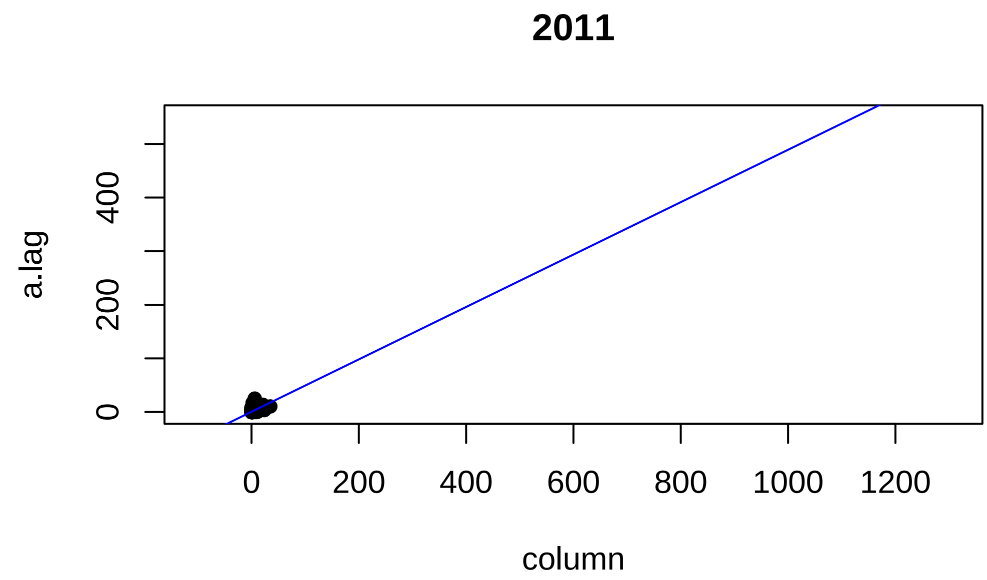

```
YEAR:  2008
NUTS3 regions with Green Building(s): 4

    Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = -0.073398, p-value = 0.5293
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
    -0.0017718371     -0.0006752194      0.0002232256
```



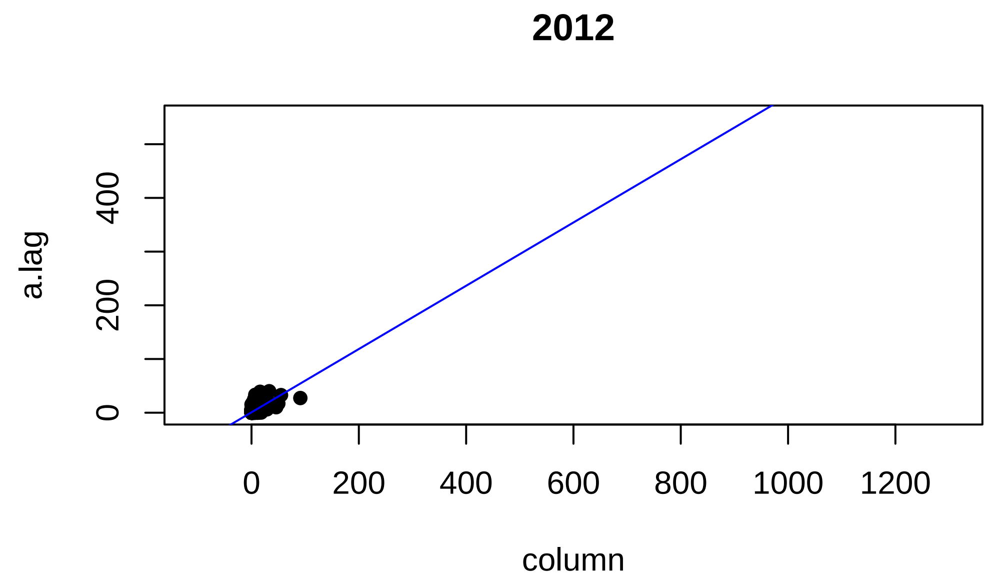

```
YEAR:  2009
NUTS3 regions with Green Building(s): 40

	Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 10.846, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.1799289580     -0.0006752194      0.0002772749
```



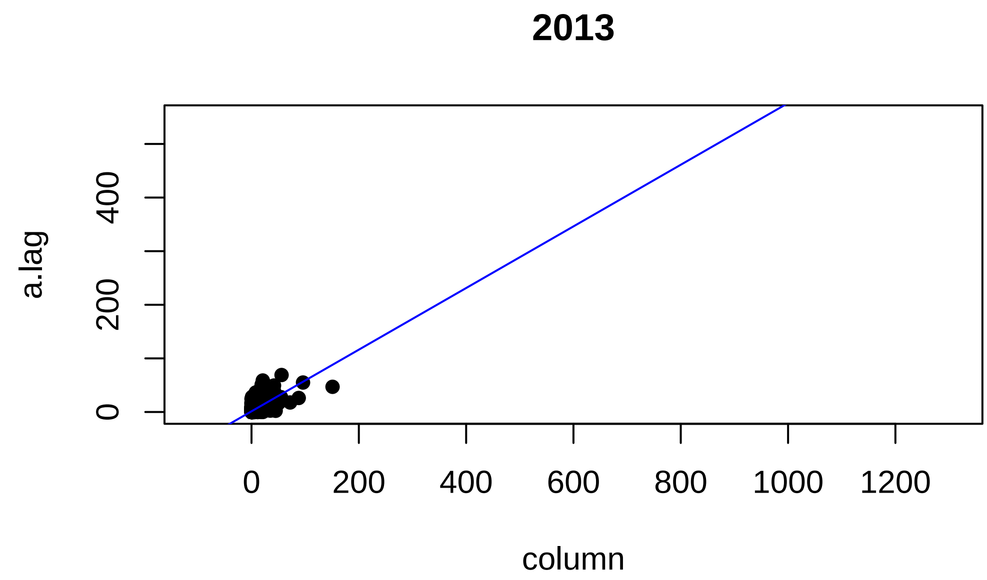

```
YEAR:  2010
NUTS3 regions with Green Building(s): 149

    Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 21.858, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.3709419005     -0.0006752194      0.0002890583
```



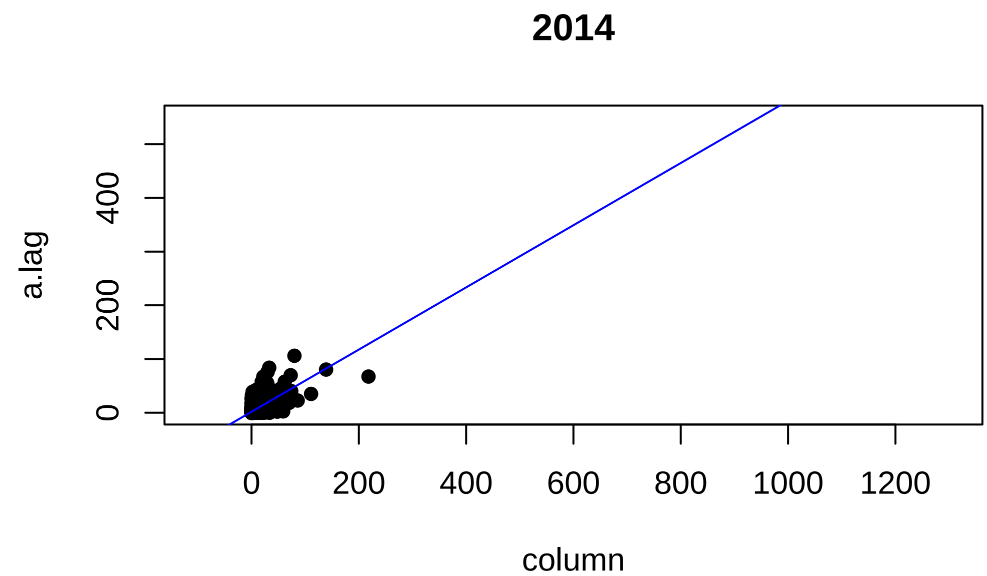

```
YEAR:  2011
NUTS3 regions with Green Building(s): 257

	Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 28.635, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.4873153573     -0.0006752194      0.0002904218
```



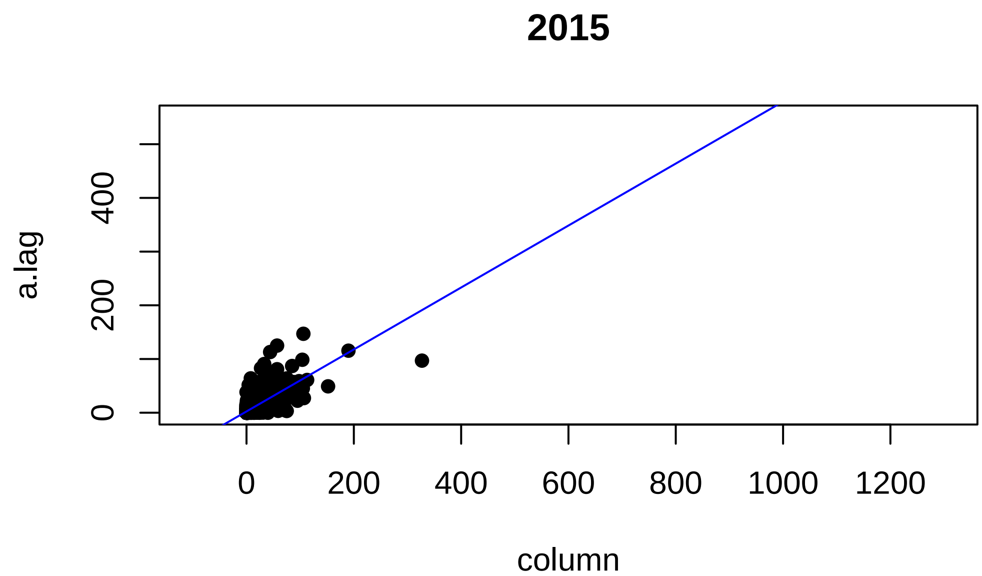

```
YEAR:  2012
NUTS3 regions with Green Building(s): 317

    Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 34.702, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.5874608500     -0.0006752194      0.0002872357
```



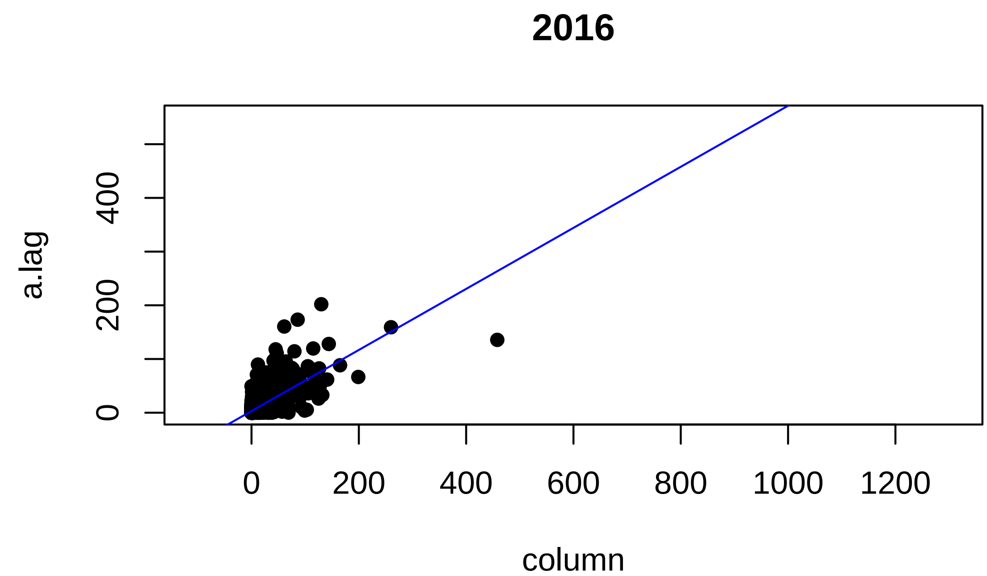

```
YEAR:  2013
NUTS3 regions with Green Building(s): 380

	Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 33.917, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.5730391120     -0.0006752194      0.0002861286
```



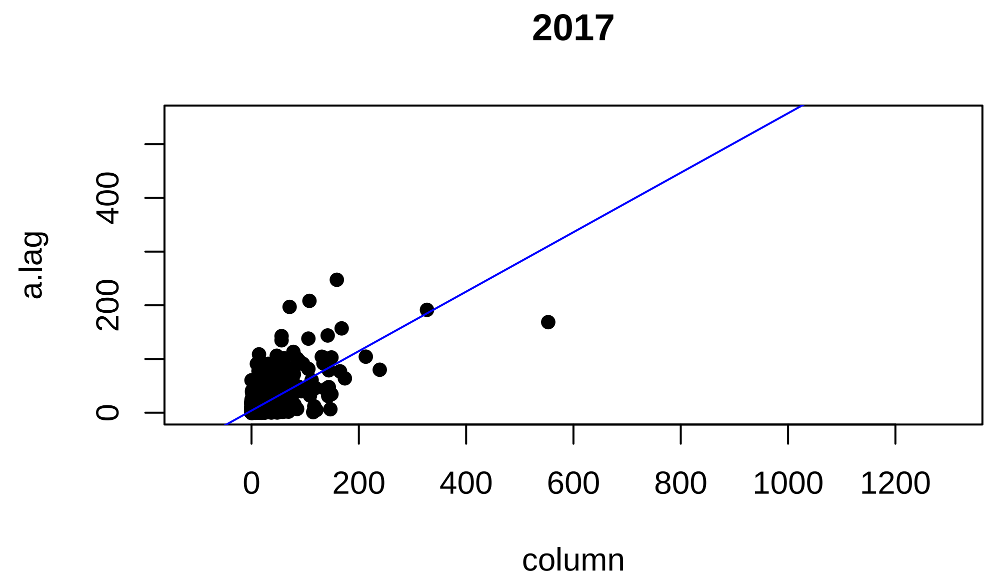

```
YEAR:  2014
NUTS3 regions with Green Building(s): 446

    Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 34.218, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.5770210092     -0.0006752194      0.0002850301
```



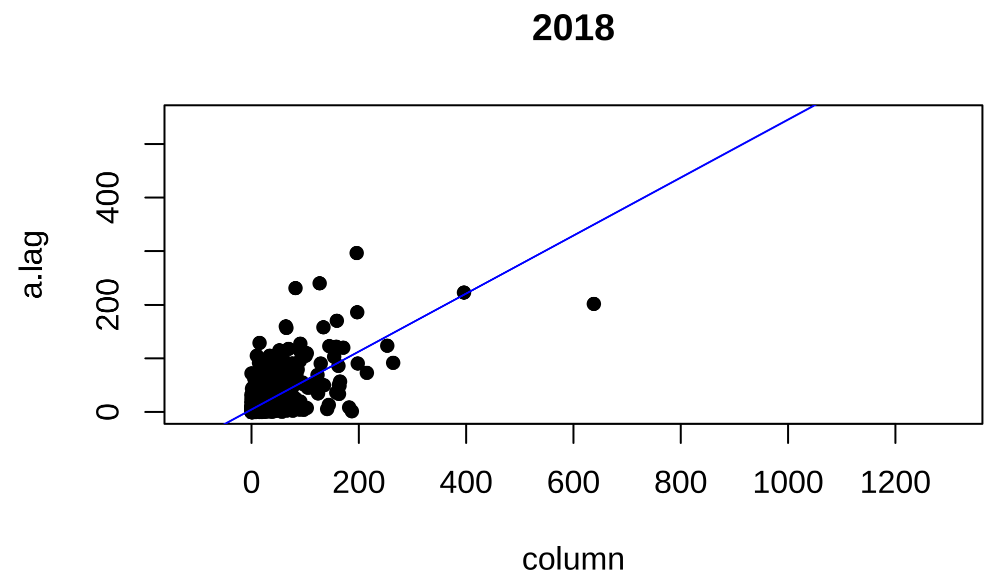

```
YEAR:  2015
NUTS3 regions with Green Building(s): 502

    Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 34.296, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.5748726185     -0.0006752194      0.0002816291
```



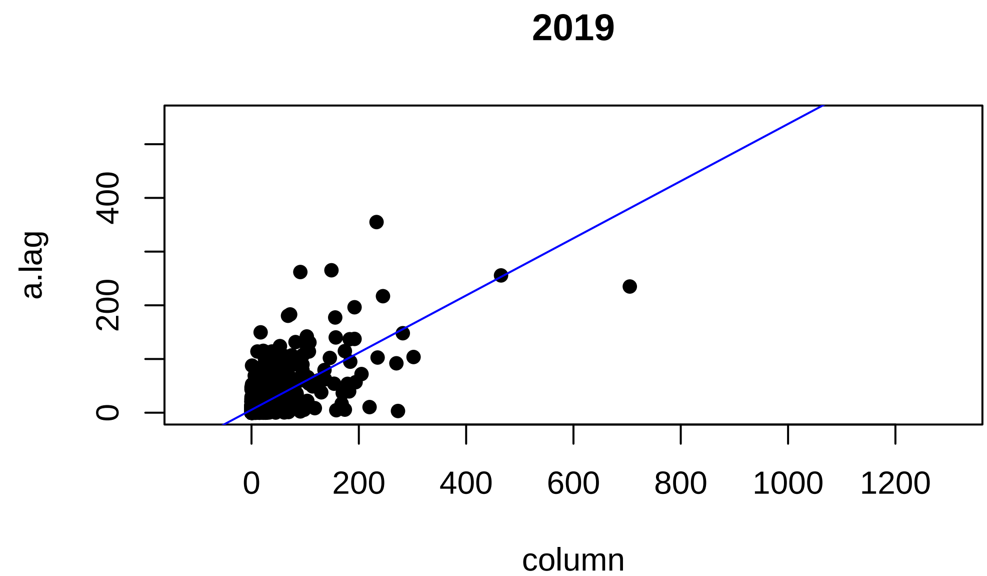

```
YEAR:  2016
NUTS3 regions with Green Building(s): 542

	Moran I test under randomisation

data:  column  
weights: lw  
n reduced by no-neighbour observations
  

Moran I statistic standard deviate = 33.981, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance 
     0.5664289216     -0.0006752194      0.0002785099 
```



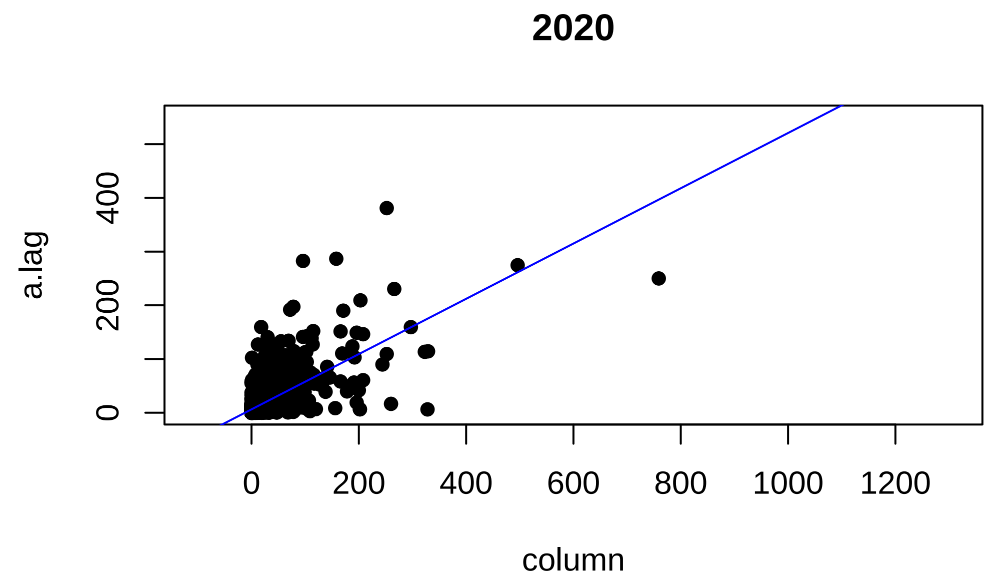

```
YEAR:  2017
NUTS3 regions with Green Building(s): 584

    Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 33.103, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.5516760462     -0.0006752194      0.0002784163
```



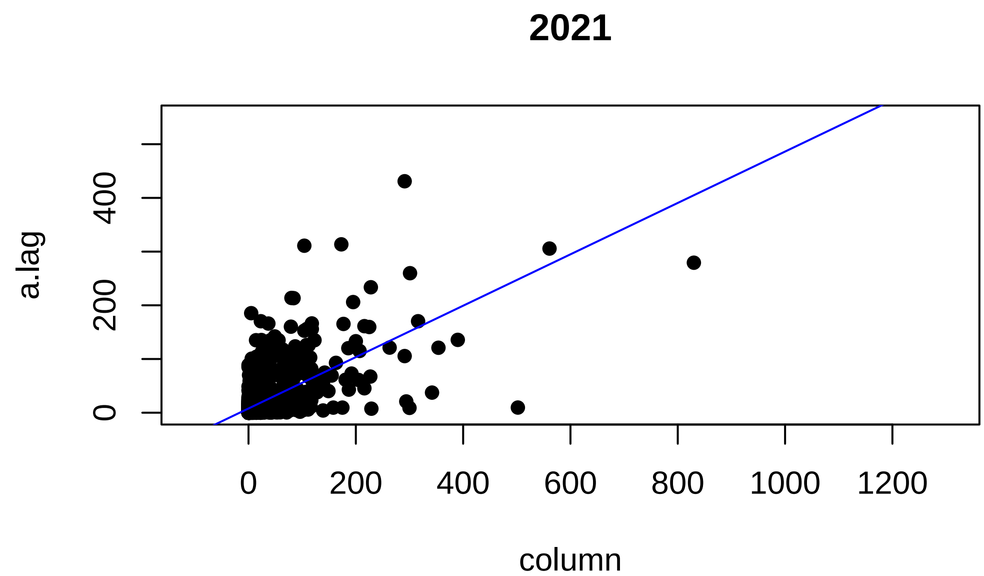

```
YEAR:  2018
NUTS3 regions with Green Building(s): 615

    Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 32.312, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.5386625321     -0.0006752194      0.0002786052
```



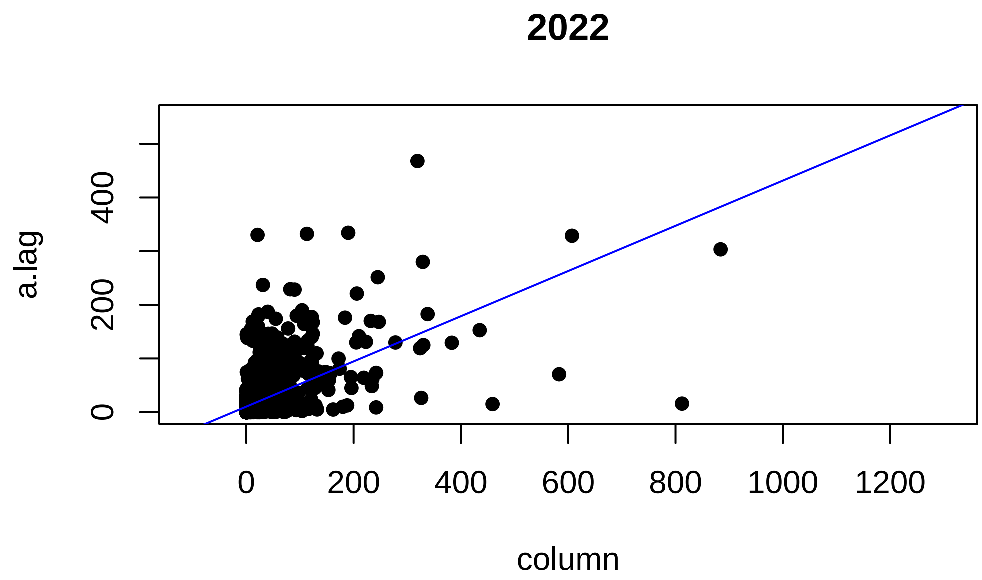

```
YEAR:  2019
NUTS3 regions with Green Building(s): 647

	Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 31.697, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.5301958800     -0.0006752194      0.0002805053
```



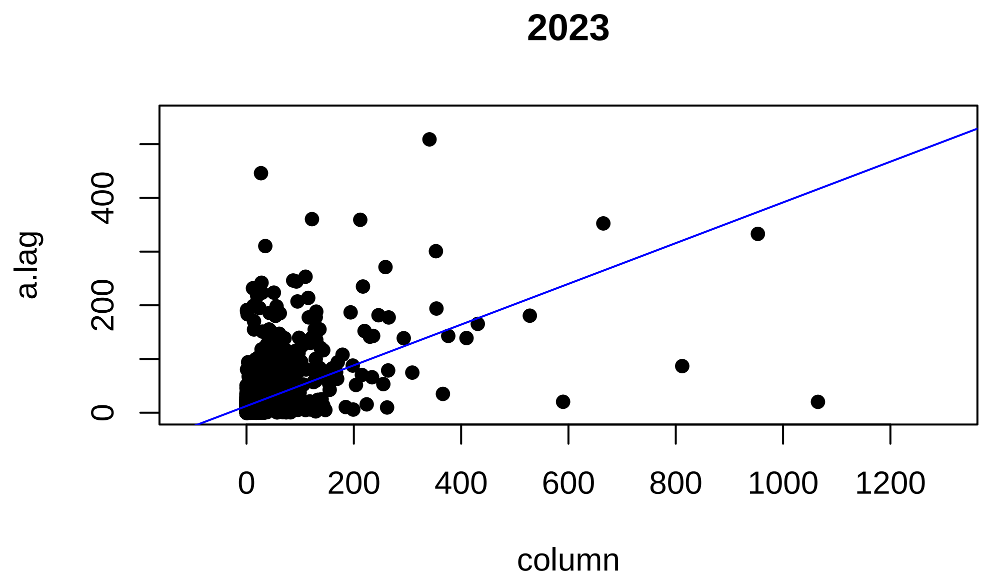

```
YEAR:  2020
NUTS3 regions with Green Building(s): 679

    Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 30.58, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.5124652617     -0.0006752194      0.0002815805
```




```
YEAR:  2021
NUTS3 regions with Green Building(s): 714

    Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 28.345, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.4762978943     -0.0006752194      0.0002831657
```




```
YEAR:  2022
NUTS3 regions with Green Building(s): 748

	Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 24.968, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.4192940260     -0.0006752194      0.0002829199
```




```
YEAR:  2023
NUTS3 regions with Green Building(s): 768

    Moran I test under randomisation

data:  column
weights: lw
n reduced by no-neighbour observations

Moran I statistic standard deviate = 22.578, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.3773527178     -0.0006752194      0.0002803293
```

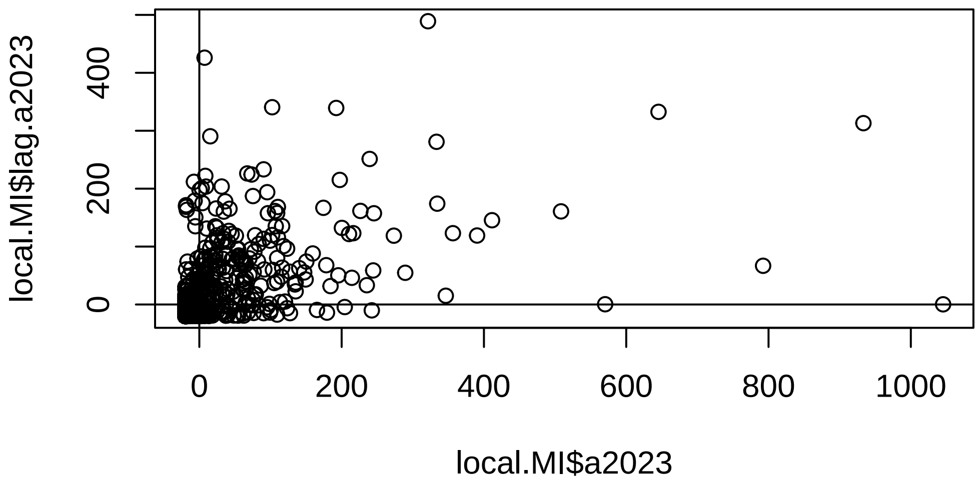
We see that in the first three years (2006-2008) certified green buildings pop up randomly. For these years the Moran's I is close to zero and clearly not significant. In 2008, however, there were only four NUTS3 regions in Europe with - a total number of four - certified green buildings. In 2009, 52 green buildings were certified, the number of NUTS3 regions with certified green buildings jumped to 40, and we begin to see a highly significant positive spatial relation in the spatial distribution of Green Buildings. From 2009 onward, Moran's I values are significantly more positive than expected, indicating a clear spatial concentration of certified green buildings. Moran's I reaches its highest value in 2012. It stays on a high level until 2018. Then, it declines continuously until the year 2023. We can interpret this as an indicator for an early period of growth when new green buildings were mainly certified in the neighborhood of existing green buildings. In the later period, green buildings seem to have diffused out to new NUTS3 regions.

4.5 Local Moran's I

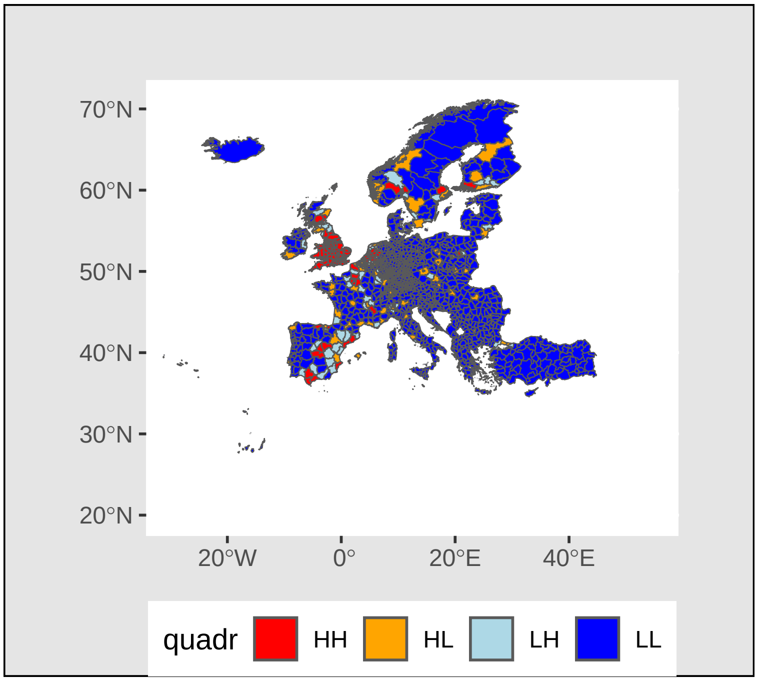
In the previous series of scatterplots we place the number of green buildings on the x-axis and the average number of green buildings of the respective neighbors on the y-axis. When we subtract the respective average from both variables, we can place the regions into four quadrants: the top right quadrant with above average values of the region and of its neighbors (called "high-high" or "HH" in exploratory spatial analysis); at the bottom left regions with below average values in both dimensions (called "low-low" or "LL"); "low-high" or "LH" regions are in the top left quadrant, "high-low" or "HL" in the bottom right quadrant.

The next code chunk calculates these indicators and puts them into a scatter plot and a map.

```
local.MI <- cbind(nuts3_sf, localmoran(nuts3_sf$a2023, lw))
local.MI$Ii <- local.MI$Ii - mean(local.MI$Ii, na.rm = TRUE)

local.MI$lag.a2023<-  lag.listw(lw,local.MI$a2023, NAOK = TRUE)

# centers the variable of interest around its mean

local.MI$a2023 <- local.MI$a2023 - mean(local.MI$a2023, na.rm = TRUE)
local.MI$lag.a2023 <- local.MI$lag.a2023 - mean(local.MI$lag.a2023, na.rm = TRUE)
local.MI$quadr <- "HH"
local.MI$quadr[local.MI$a2023 < 0 & local.MI$lag.a2023 < 0] <- "LL"
local.MI$quadr[local.MI$a2023 < 0 & local.MI$lag.a2023 >= 0] <- "LH"
local.MI$quadr[local.MI$a2023 >= 0 & local.MI$lag.a2023 < 0] <- "HL"

plot(local.MI$a2023, local.MI$lag.a2023)
abline(h = 0)
abline(v = 0)
```




```
p <- ggplot(local.MI) +
  geom_sf(aes(fill=quadr)) +
  xlim(-30,  55) +
  ylim(20, 71) +
  scale_fill_manual(values=c("red","orange","lightblue","blue")) +
  theme(legend.position = "bottom",
          plot.margin = margin(1, 1, 0, 1, "cm"),
          panel.background = element_rect(fill = "white"),
          plot.background = element_rect(
            fill = "grey90",
            colour = "black"
          )
  )
#ggplotly(p)
p
```



Let us count, how many regions fall into each of the quadrants.

```
(lm.counts <- summary(factor(local.MI$quadr)))
```

```
 HH   HL   LH   LL
206   72   88 1156
```

As the summary shows, most of the regions (1156) fall into the LL (bottom-left) quadrant. The second largest category are regions falling into HH (206). The HL and LH quadrants contain the smallest numbers of regions (72 and 88, respectively).

5 Calculating the point of gravity of all green buildings

From the analysis so far we know

1. that the numbers of certifications and of certified green buildings have grown massively over the observation period and
2. that the certifications and the building stock are concentrated at the country and at the NUTS3 level, and
3. that they occur in spatial clusters.

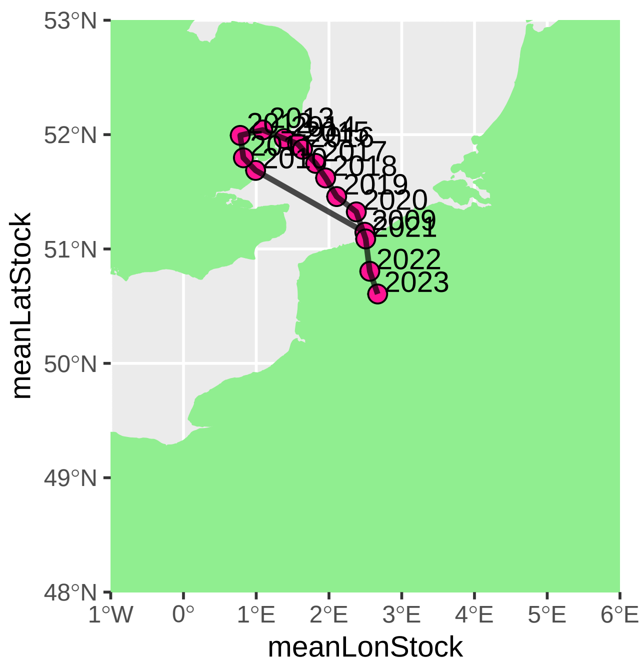
In this section we want to answer the question, how this spatial pattern of green buildings and of green building certifications evolves over time. Since we know the exact locations of all certified green buildings, we can calculate the points of gravity year by year. The coordinates of the respective point of gravity are just the average coordinates of all the respective buildings or certifications.

So, in the following code chunk we compute the mean latitude and mean longitude values for every year and store the result in the data-frame `centroid`. Since the centroids for the first three years are based on very few buildings, we exclude these years in the final step of the code chunk.

```
# First, we store the number of observations and the sum the latitudes
# and longitudes for every year.
centroids <- green %>%
  group_by(CertYear) %>%
  summarize(n = n(), sumLat = sum(lat), sumLon = sum(lon))

# Second, we calculate and add means and columns of zeros
centroids <- centroids %>%
  mutate(meanLat = sumLat/n, meanLon = sumLon/n, nStock = 0,
         sumLatStock = 0, sumLonStock =0)

# For every year, we calculate the sum of counts and the sum of
# lat and lon UP TO THAT YEAR
# This is what "sum(head(..., i))" does.
for (i in 1:nrow(centroids)) {
  centroids$nStock[i] <- sum(head(centroids$n ,i))
  centroids$sumLatStock[i] <- sum(head(centroids$sumLat ,i))
  centroids$sumLonStock[i] <- sum(head(centroids$sumLon ,i))
}

# Fourth, we calculate the means for every year from the sums.
centroids <- centroids %>%
  mutate(meanLatStock = sumLatStock/nStock,
         meanLonStock = sumLonStock/nStock)

centroids <- centroids %>% filter(nStock > 50)
```

In a first step, we generate a base map that does not show any boundaries. Then, we put the stock-centroids on that map, to see, how the centroids of the stock of the certified green buildings changes from year to year. We focus the map to the Channel and parts of Great Britain, France, Belgium, and the Netherlands.

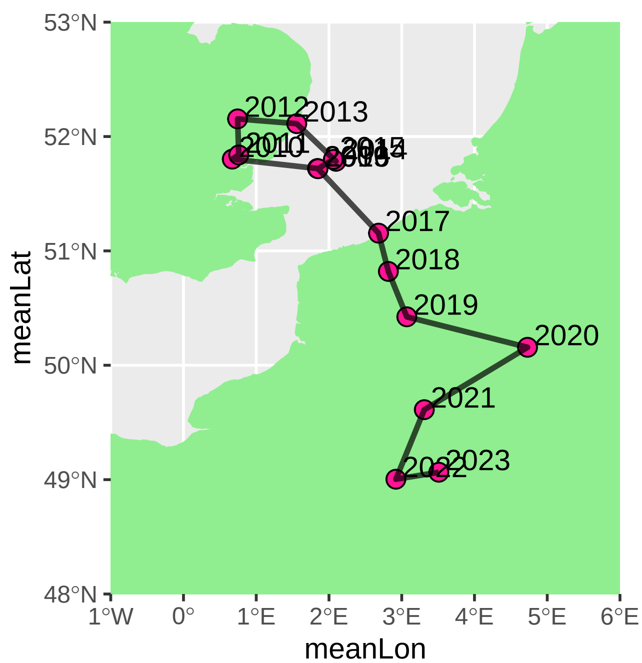
```
base_map <- ggplot(nuts3_sf) +
  geom_sf(colour="light green", fill="light green")

map_stock_data <-
  base_map  +
  geom_point(data=centroids,
             aes(x=meanLonStock, y=meanLatStock), colour="Black",
             fill="Deep Pink", pch=21, size=3, alpha=I(1.0)) +
  geom_path(data=centroids,
             aes(x=meanLonStock, y=meanLatStock), linewidth=1,
             alpha=I(0.7)) +
  geom_text(data=centroids,aes(x=meanLonStock, y=meanLatStock,
             label=CertYear), hjust=-0.1, vjust=-0.1)

map_stock_data +
  coord_sf(xlim = c(-1, 6), ylim = c(48, 53), expand = FALSE)
```



While the centroid was at the French coast in 2009, it then jumped across the Channel the coast of Essex, North-East of London. From there, it gradually looped around and moved South-East back to the continent. Since we know that certifications in Great Britain are mainly BREEAM, this illustrates the early dominance of this certification schema and the gradual increase of certification activities by LEED on the European continent.

This is clearly confirmed when we look at the centroids of the certifications year by year.

```
map_flow_data <-
  base_map  +

  geom_point(data=centroids,
             aes(x=meanLon, y=meanLat), colour="Black",
             fill="Deep Pink",pch=21, size=3, alpha=I(1.0)) +
  geom_path(data=centroids,
             aes(x=meanLon, y=meanLat), linewidth=1, alpha=I(0.7)) +
  geom_text(data=centroids,aes(x=meanLon, y=meanLat, label=CertYear),
            hjust=-0.1, vjust=-0.1)

map_flow_data +
  coord_sf(xlim = c(-1, 6), ylim = c(48, 53), expand = FALSE)
```



This map shows, how the certification activities year after year in average move from Great Britain toward the South to France. The centroids of 2009 and 2016 almost perfectly coincide. This creates this perceived “loop” over Essex and the North Sea.

6 Estimating the growth trajectory of the number of green buildings

In this chapter, we want to take a first step toward modeling the growth trajectories of green guilding certifications.

We assume that the number of certified green buildings over time follows a growth process that becomes saturated at some point. Such a relation can be captured by the logistic curve (see chapter 24 of (Childs, Hindle, and Warren, n.d.)).

$$y = \phi_1 * \frac{1}{1 + \exp((\phi_2 - x)/\phi_3)}$$

This logistic curve has three parameters that we can estimate with non-linear regression. The three parameters and their meanings are:

1. ϕ_1 measures the upper limit (asymptote) of the S-curve
2. ϕ_2 measures the inflection point of the S-curve, i.e., the value of x where y has reached half its maximum height. Here, the slope switches from increasing to decreasing.
3. ϕ_3 is a scale parameter that measures how quickly the function increases. The larger this parameter, the more spread out is the S-curve.

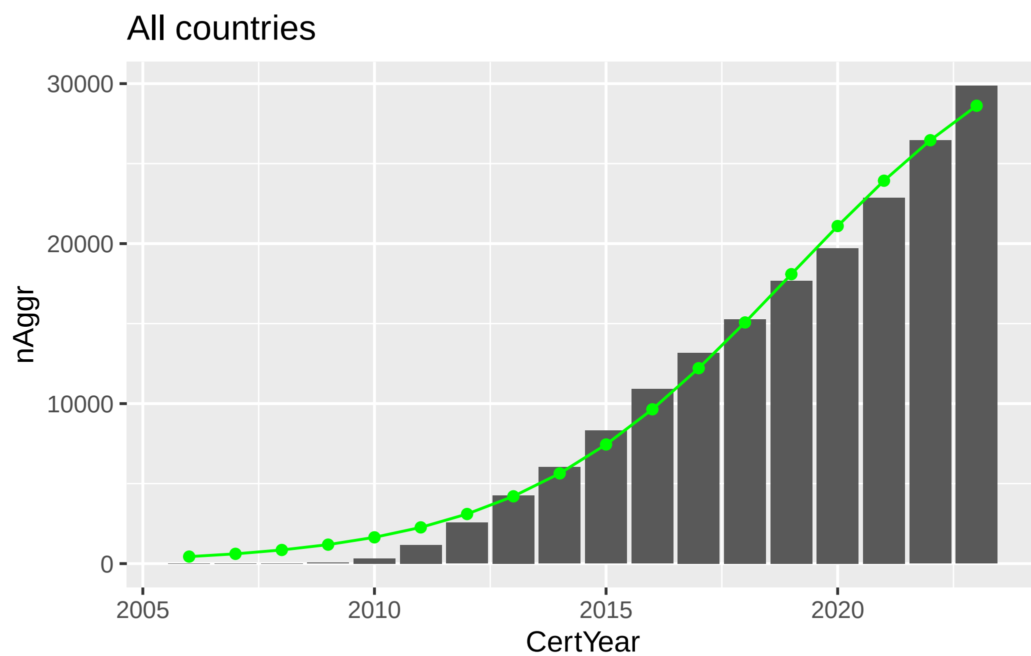
6.1 Growth trajectory for the whole set of countries

```
nonlinAggr <- nls(nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3),
                  data=byYearAggr)
(totalSum <- summary(nonlinAggr))
```

```
Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1 3.584e+04  2.969e+03   12.07 3.99e-09 ***
phi_2 2.019e+03  5.969e-01 3382.57  < 2e-16 ***
phi_3 2.946e+00  2.584e-01   11.40 8.68e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 969.8 on 15 degrees of freedom
```

```
Number of iterations to convergence: 0
Achieved convergence tolerance: 6.821e-07
```

```r
byYearAggr$fittedNL = fitted(nonlinAggr)
ggplot(byYearAggr) +
  geom_col(aes(x=CertYear,y=nAggr)) +
  geom_line(aes(x=CertYear, y=fittedNL), color="green") +
  geom_point(aes(x=CertYear, y=fittedNL), color="green") +
  ggtitle("All countries")
```



For later use, we create a new data-frame `PopulationByCountryPlus` with a row with the total population of all the countries. We will use this information later.

```r
PopulationByCountryPlus <- PopulationByCountry %>%
  add_row(country = "TOTAL", pop = sum(PopulationByCountry$pop))
```

To be better able to compare these results to those for the individual countries, we compute the following indicators:

1. **Phi1ByPop**, the parameter ϕ_1 standardized by the total population($=100000$*phi1/pop);

2. **Slope**, the slope at the inflection point of the function, standardized to the same saturation level. Since the slope at the inflection point ($x = \phi_2$) is $\frac{\phi_1}{2\phi_3}$, the standardized slope is just $\frac{1}{2\phi_3}$.
3. **Tens**, the year when the curve reaches 10% of the saturation level. This is calculated by $x = \phi_2 - \phi_3 \ln(\frac{1}{a} - 1)$ with $a = 0.1$.

We calculate the same set of indicators for each of the countries as well below. The corresponding values for all countries are shown below:

```
totalParams <- function(){
  Phi1ByPop <- 100000*totalSum$coefficients[1,1]/sum(PopulationByCountry$pop)
  slope <- 1/(2*totalSum$coefficients[3,1])
  tens <- totalSum$coefficients[2,1]-totalSum$coefficients[3,1]*log(9)
  cat(paste0("Phi1ByPop = ",round(Phi1ByPop,4),"\n"))
  cat(paste0("Slope     = ",round(slope,4),"\n"))
  cat(paste0("Tens      = ",round(tens,4),"\n"))
}
totalParams()
```

```
Phi1ByPop = 5.6829
Slope     = 0.1697
Tens      = 2012.4713
```

For the full set of countries, the model estimates a saturation level of 3.584×10^4 certified green buildings. The inflection point of the growth process is estimated for the year 2018.94. As compared to the individual countries, the overall distribution is more spread out than most of the individual countries. Most likely, this is the effect of the aggregation of countries with different growth trajectories.

6.2 Analysis by countries

Now, we apply the same analysis for each of the countries separately. To prepare the required data, we first count the certificates by country and by year. This data-frame is the basis for the following analyses.

```
byCountryYear <- green %>%
  count(country, CertYear)
```

We create an empty dataframe to collect all the parameter estimates.

```
parameters <- data.frame(country = "TOTAL",
  phi1 = totalSum$coefficients[1,1],
  phi2 = totalSum$coefficients[2,1],
  phi3 = totalSum$coefficients[3,1])
```

To be able to loop over all the countries, we create the following function, which gets the country code and based on that filters the country data, calculates the stock of the green buildings per year, estimates the logistic equation, stores the parameter estimates in a data-frame, and plots the bar chart of the green buildings and the green line of the estimated logistic function.

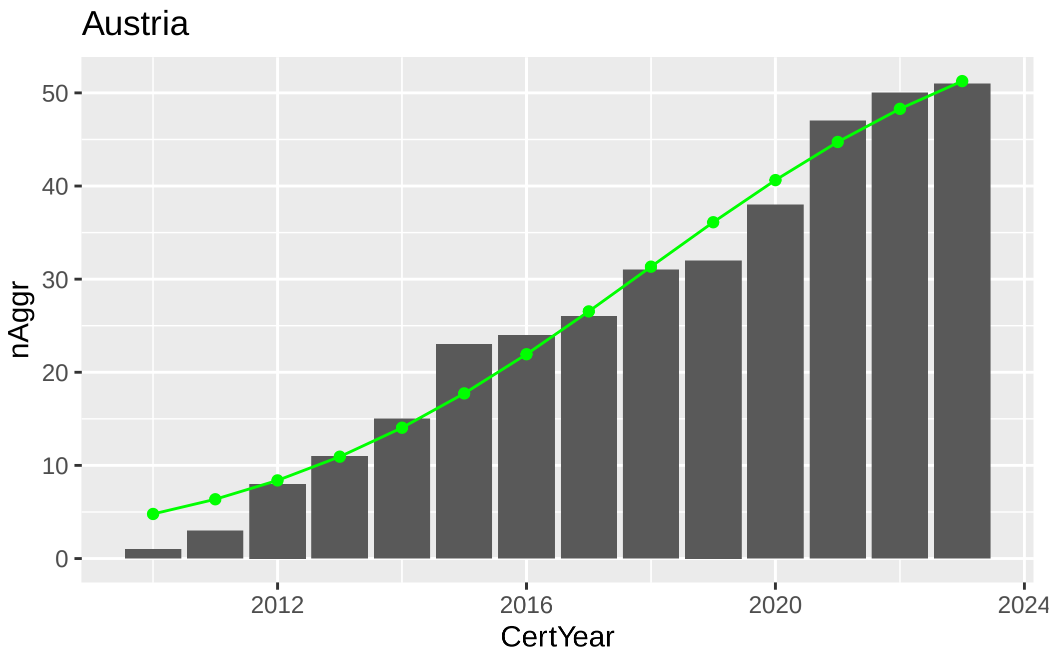
```
trajectoryByCountry <- function(cntry) {
  byYear <- byCountryYear %>%
    filter(country == cntry)

  byYear$nAggr = sapply(byYear$CertYear,  function(x){
    sum(byYear$n[byYear$CertYear <= x])})

  nonlin <- nls(nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3), data=byYear)
  reg2 <- summary(nonlin)
  df <- data.frame(country = cntry,
                   phi1 = reg2$coefficients[1,1],
                   phi2 = reg2$coefficients[2,1],
                   phi3 = reg2$coefficients[3,1])

  byYear$fittedNL = fitted(nonlin)
  plot2 <- ggplot(byYear) +
    geom_col(aes(x=CertYear,y=nAggr)) +
    geom_line(aes(x=CertYear, y=fittedNL), color="green") +
    geom_point(aes(x=CertYear, y=fittedNL), color="green") +
    ggtitle(countryNames[cntry])
  return(list(reg2,plot2,df))
}
```

As it turns out, the estimation procedure does not converge for 8 of the 33 the countries. These countries are eliminated from the following analysis. The reason for their exclusion will be discussed below.

```
for (c in countries) {
  if ( c == "CY" | c == "EE" | c == "HR" | c == "IS" | c == "MT" |
       c == "NL" | c == "RS" | c == "SI" ) {
```

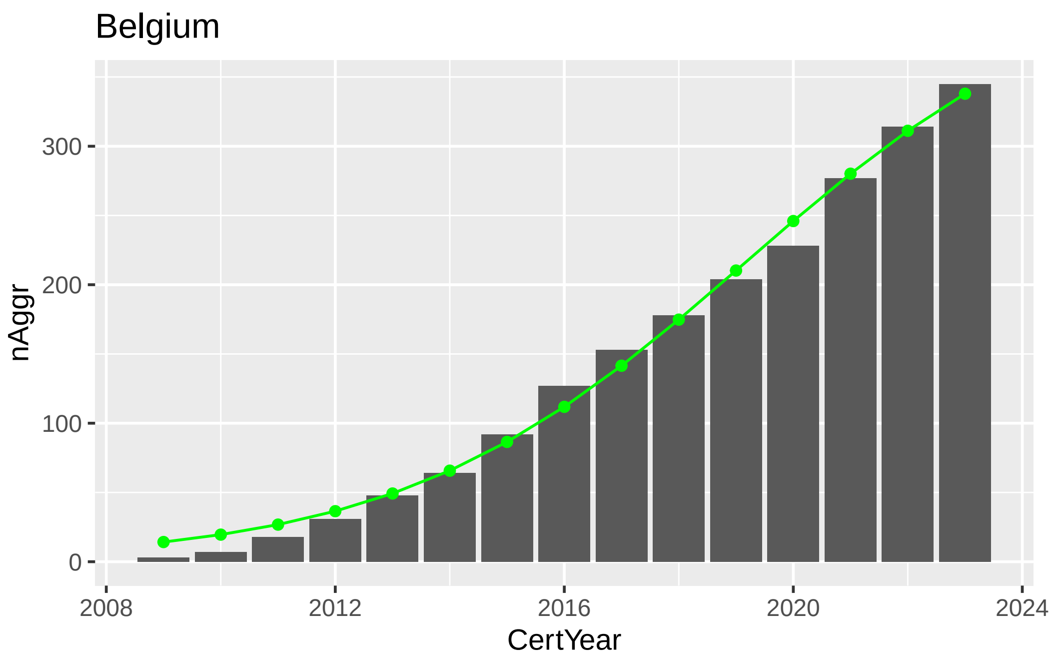
```
    next
  }
  out <- trajectoryByCountry(c)
  cat(paste0("\n",countryNames[c]),"\n\n")
  cat("Certified Buildings (STOCK)\n")
  print(out[1])
  print(out[2])
  parameters <- rbind(parameters, data.frame(out[3]))
}
```

```
Austria

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error  t value Pr(>|t|)
phi_1   61.5686     7.4990    8.210 5.10e-06 ***
phi_2 2017.8853     0.9755 2068.498  < 2e-16 ***
phi_3    3.1875     0.4910    6.492 4.48e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.879 on 11 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 4.941e-06


[[1]]
```




```
Belgium

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1  432.1417    34.9085   12.38 3.42e-08 ***
phi_2 2019.1623     0.5824 3466.87  < 2e-16 ***
phi_3    3.0048     0.2459   12.22 3.95e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.12 on 12 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 3.849e-06
```

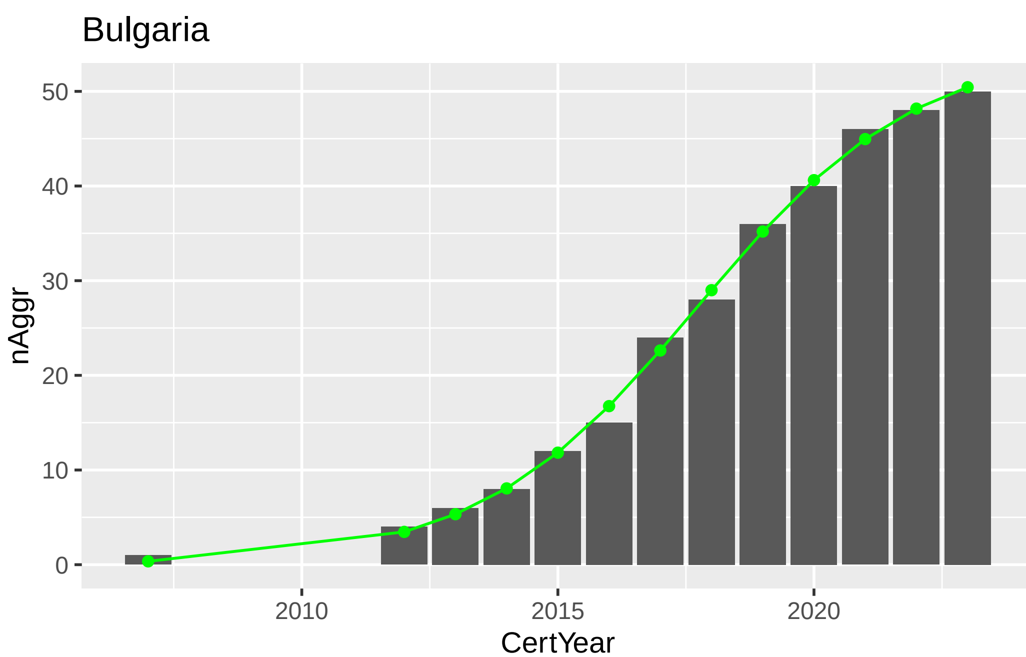
[[1]]




```
Bulgaria

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error  t value Pr(>|t|)
phi_1   54.7200     1.4412    37.97 3.83e-12 ***
phi_2 2017.7448     0.1644 12273.05  < 2e-16 ***
phi_3    2.1317     0.1068    19.97 2.18e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9728 on 10 degrees of freedom

Number of iterations to convergence: 0
```

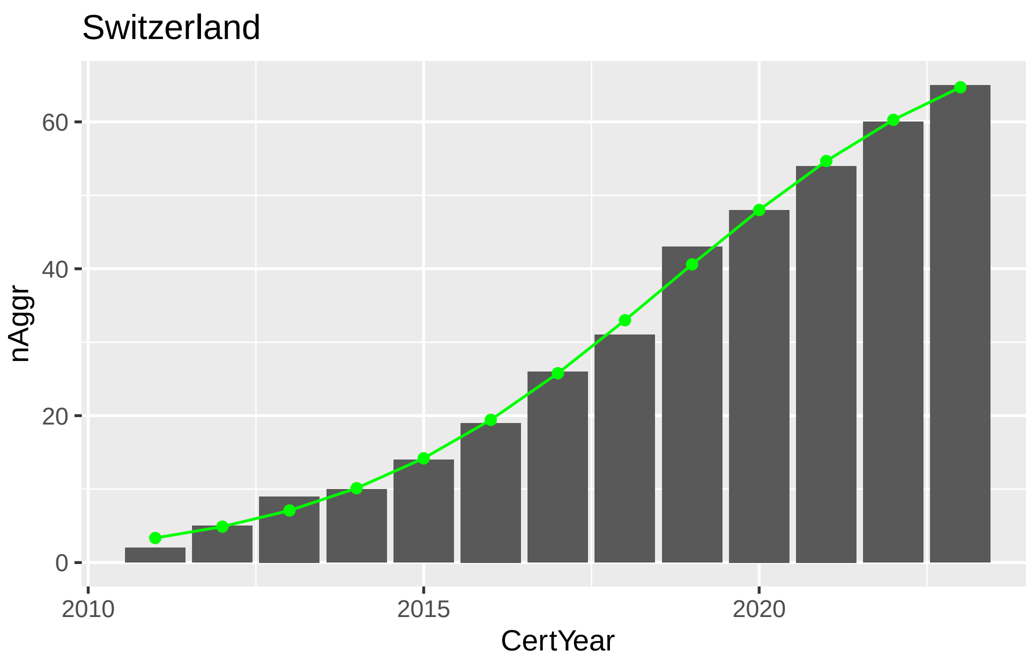
```
Achieved convergence tolerance: 8.539e-07
```

```
[[1]]
```




```
Switzerland

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1   76.0368     3.0384   25.02 2.38e-10 ***
phi_2 2018.6618     0.2591 7792.31  < 2e-16 ***
phi_3    2.4886     0.1331   18.70 4.13e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

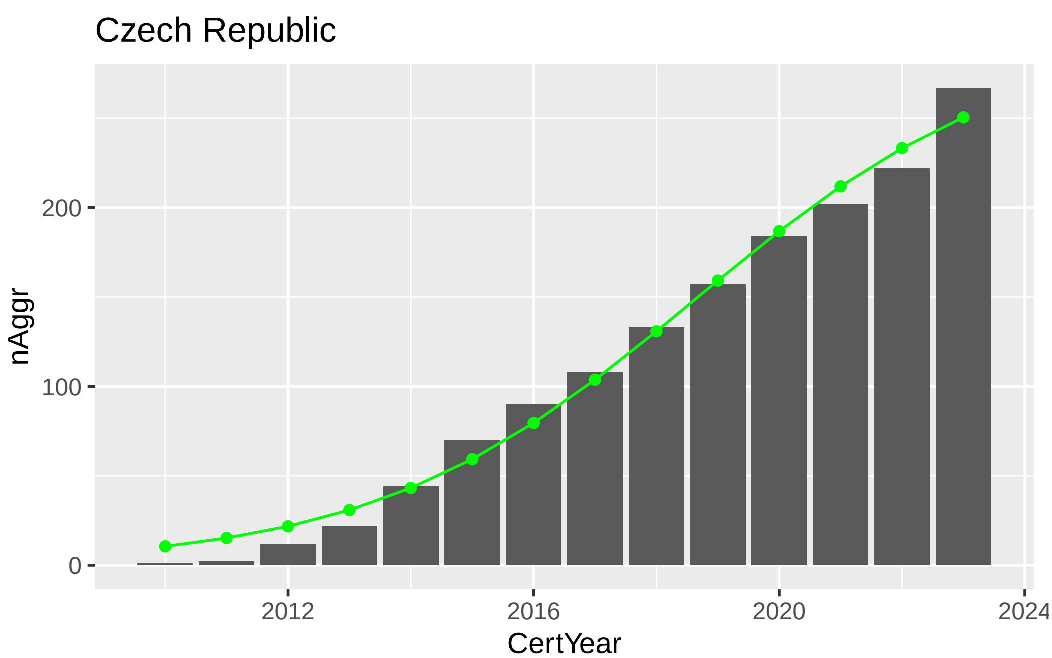
```
Residual standard error: 1.269 on 10 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 2.625e-07


[[1]]
```




```
Czech Republic

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error  t value Pr(>|t|)
phi_1  297.5600    25.9227   11.479 1.83e-07 ***
phi_2 2018.6343     0.5858 3445.756  < 2e-16 ***
phi_3    2.6124     0.2903    8.999 2.10e-06 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.43 on 11 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 9.028e-07
```

[[1]]



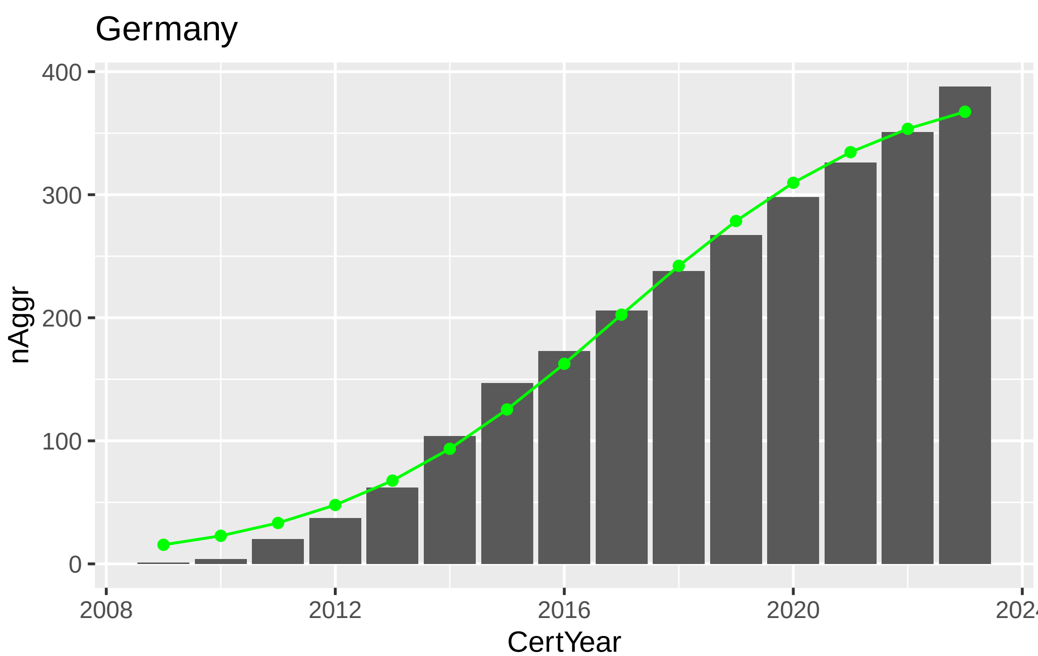
Germany

Certified Buildings (STOCK)
[[1]]

```
Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
```

```
phi_1  398.7815    19.4930   20.46 1.07e-10 ***
phi_2 2016.9225     0.3537 5702.90  < 2e-16 ***
phi_3    2.4699     0.2226   11.09 1.15e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.07 on 12 degrees of freedom

Number of iterations to convergence: 1
Achieved convergence tolerance: 2.978e-06
```

[[1]]

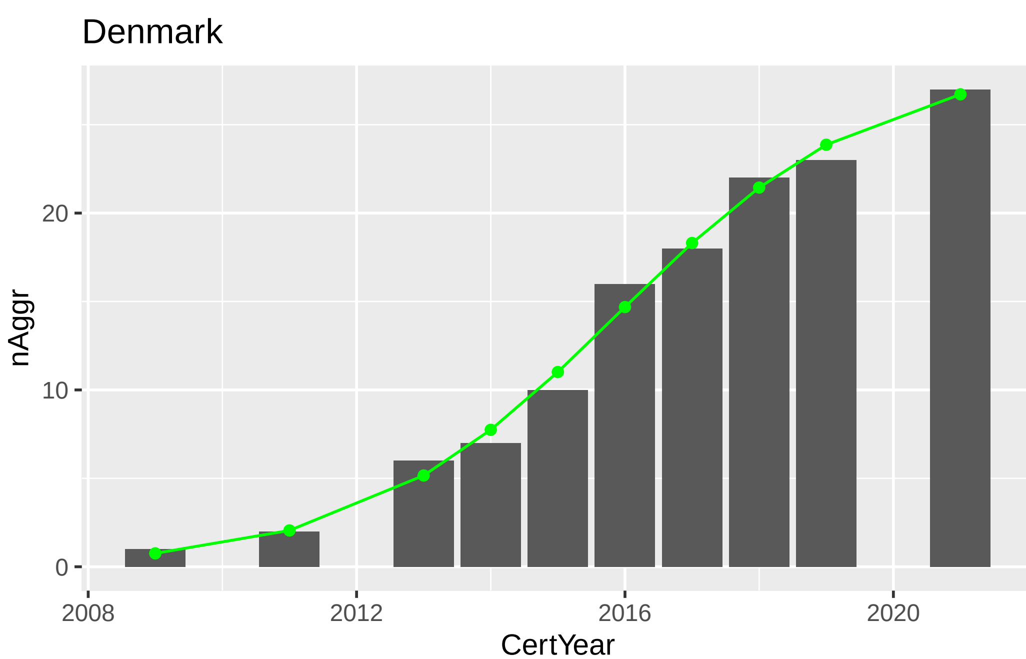


Denmark

Certified Buildings (STOCK)
[[1]]

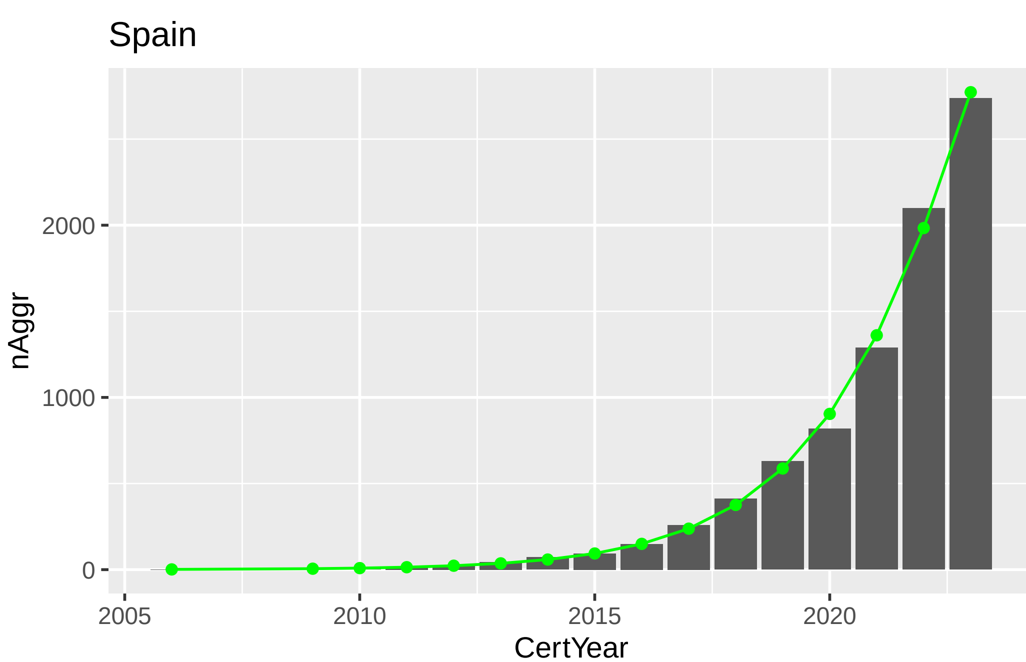
```
Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)
```

```
Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1   28.5638     1.3512   21.14 1.33e-07 ***
phi_2 2015.8916     0.2572 7838.83  < 2e-16 ***
phi_3    1.9123     0.1801   10.62 1.44e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8705 on 7 degrees of freedom

Number of iterations to convergence: 1
Achieved convergence tolerance: 7.089e-06


[[1]]
```

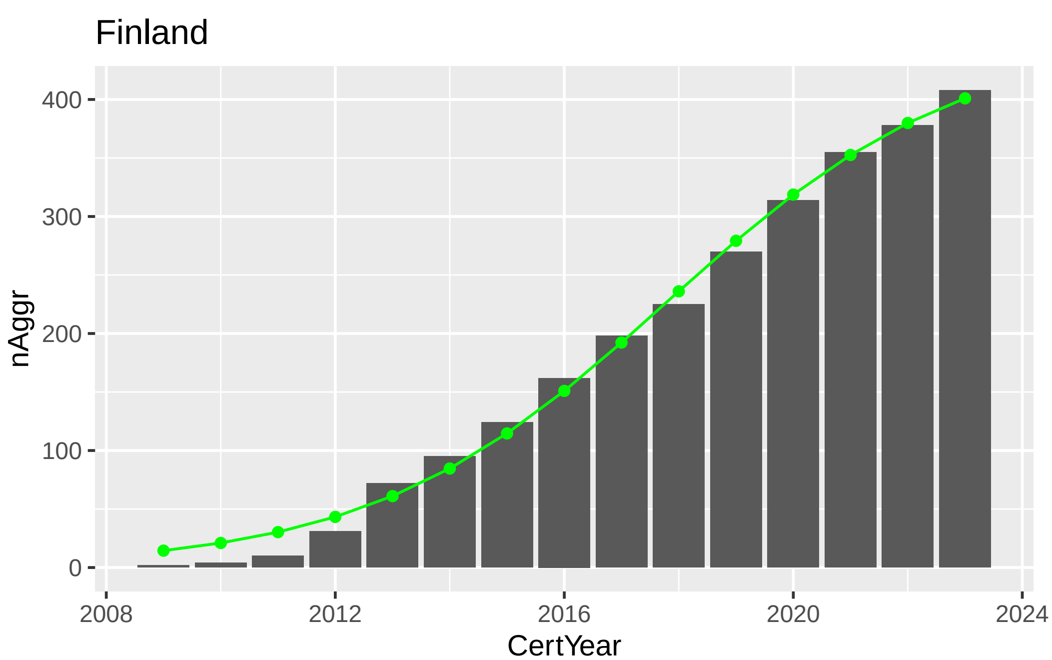



```
Spain

Certified Buildings (STOCK)
```

```
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error  t value Pr(>|t|)
phi_1 8014.5408  2440.0484    3.285  0.00592 **
phi_2 2024.3411     1.0510 1926.079  < 2e-16 ***
phi_3    2.1052     0.1607   13.102 7.24e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 48.89 on 13 degrees of freedom

Number of iterations to convergence: 1
Achieved convergence tolerance: 1.403e-06


[[1]]
```



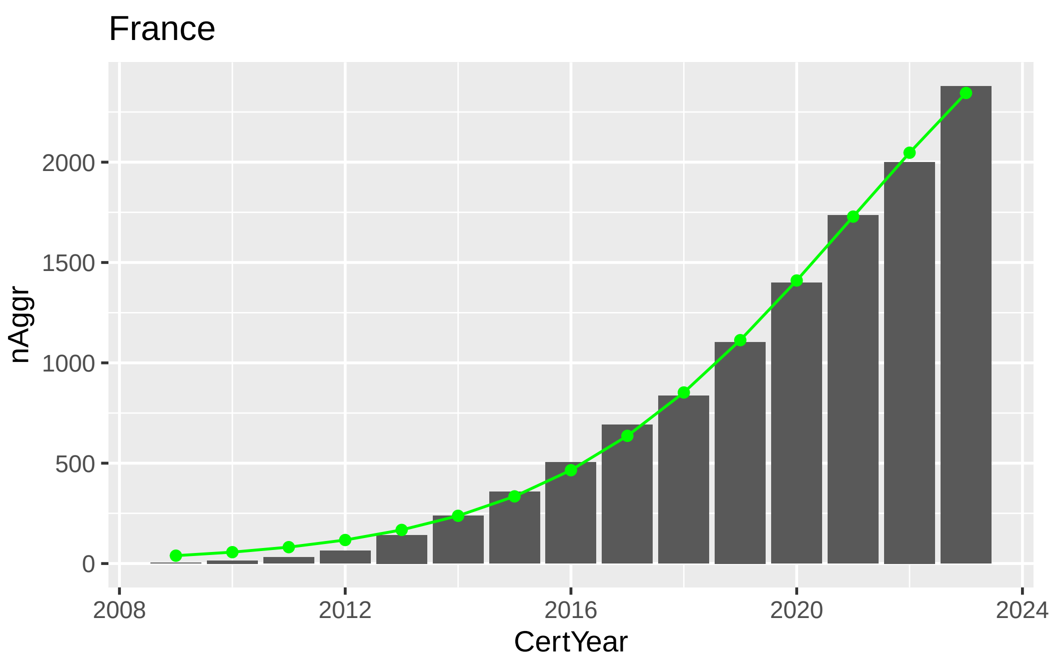
```
Finland

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1  454.0996    22.5202   20.16 1.27e-10 ***
phi_2 2017.7956     0.3484 5791.09  < 2e-16 ***
phi_3    2.5747     0.1932   13.33 1.49e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.12 on 12 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 2.808e-06


[[1]]
```



France

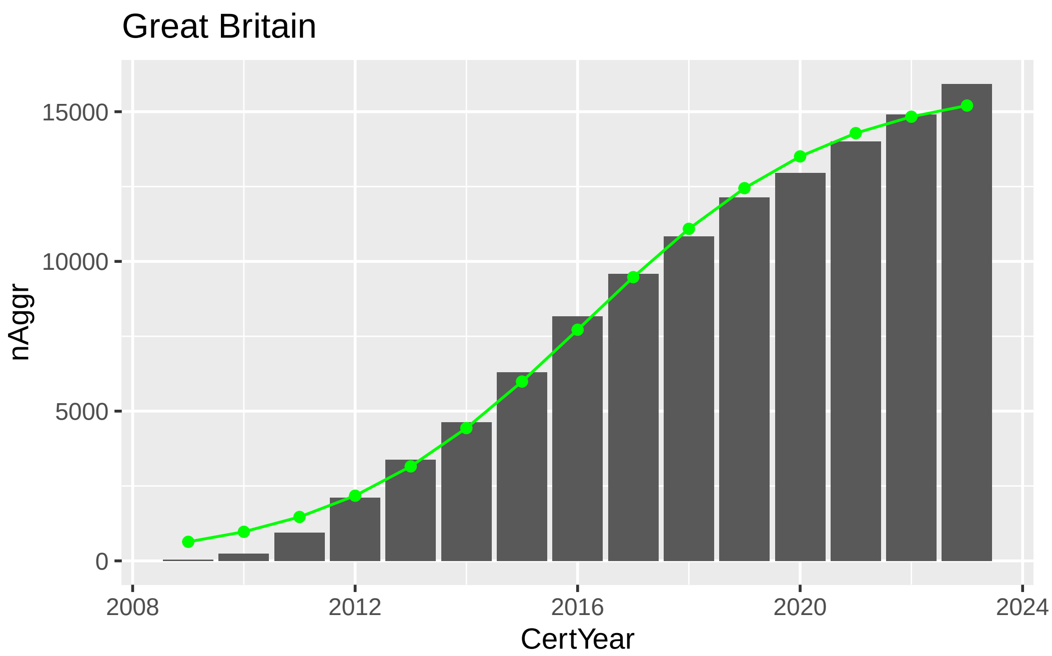
Certified Buildings (STOCK)

```
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1 3461.4082   247.0853   14.01 8.47e-09 ***
phi_2 2021.0063     0.4188 4826.12  < 2e-16 ***
phi_3    2.6875     0.1379   19.48 1.89e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 39.02 on 12 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 6.629e-06


[[1]]
```




```
Great Britain

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1 1.592e+04  4.761e+02   33.44 3.25e-13 ***
phi_2 2.016e+03  2.174e-01 9274.40  < 2e-16 ***
phi_3 2.244e+00  1.548e-01   14.50 5.73e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 468.5 on 12 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 8e-06
```

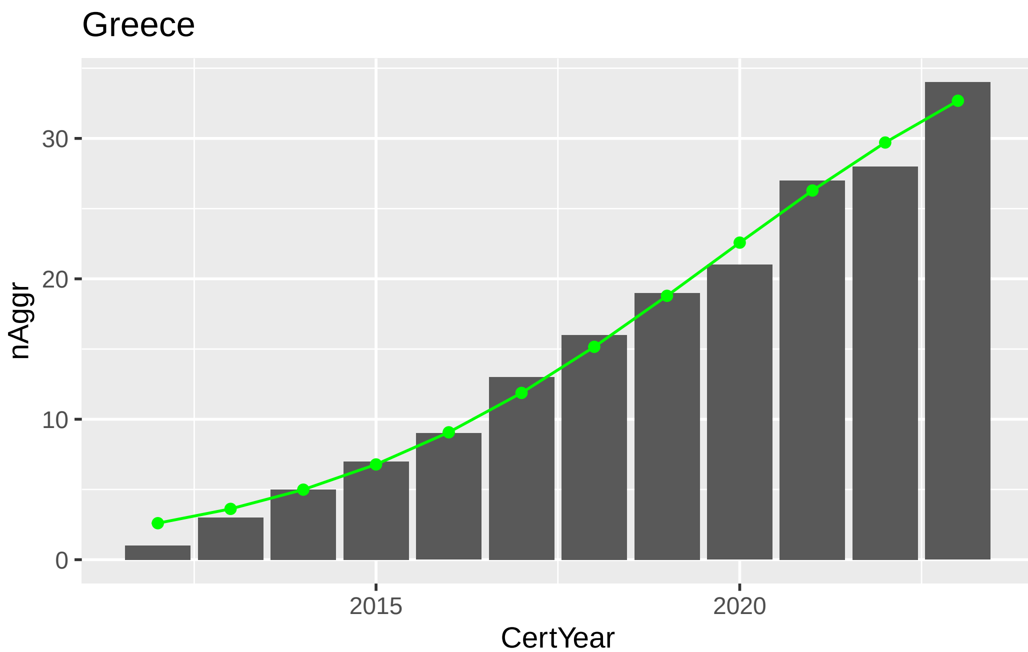
[[1]]




```
Greece

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error  t value Pr(>|t|)    
phi_1   42.6452     4.8176    8.852 9.78e-06 ***
phi_2 2019.6693     0.7482 2699.526  < 2e-16 ***
phi_3    2.8039     0.3149    8.905 9.31e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.186 on 9 degrees of freedom

Number of iterations to convergence: 0
```

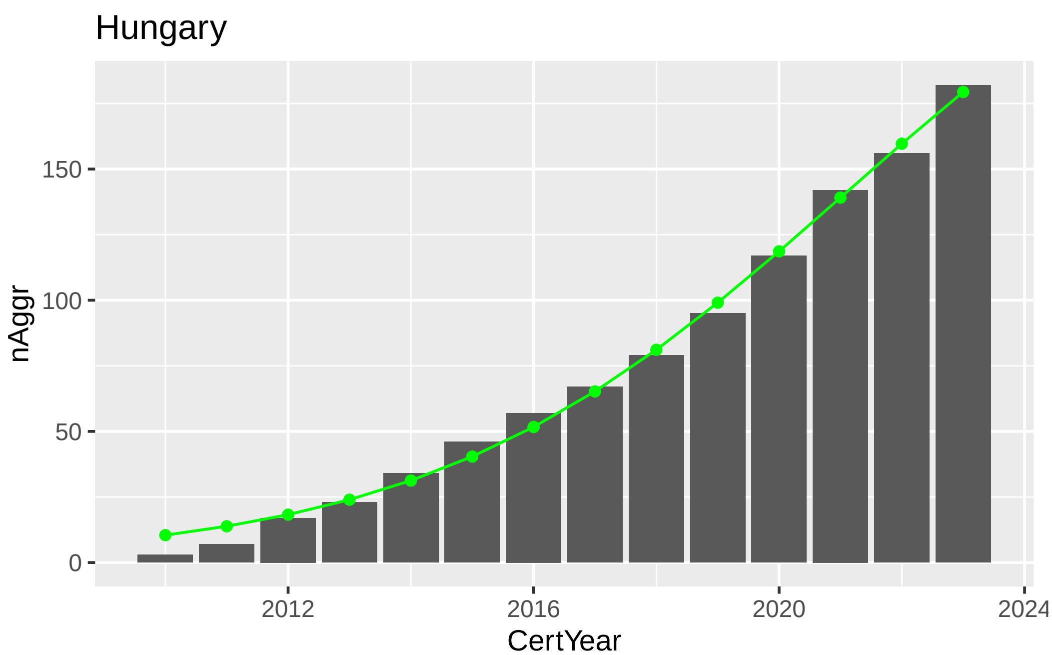
```
Achieved convergence tolerance: 4.802e-06
```

```
[[1]]
```



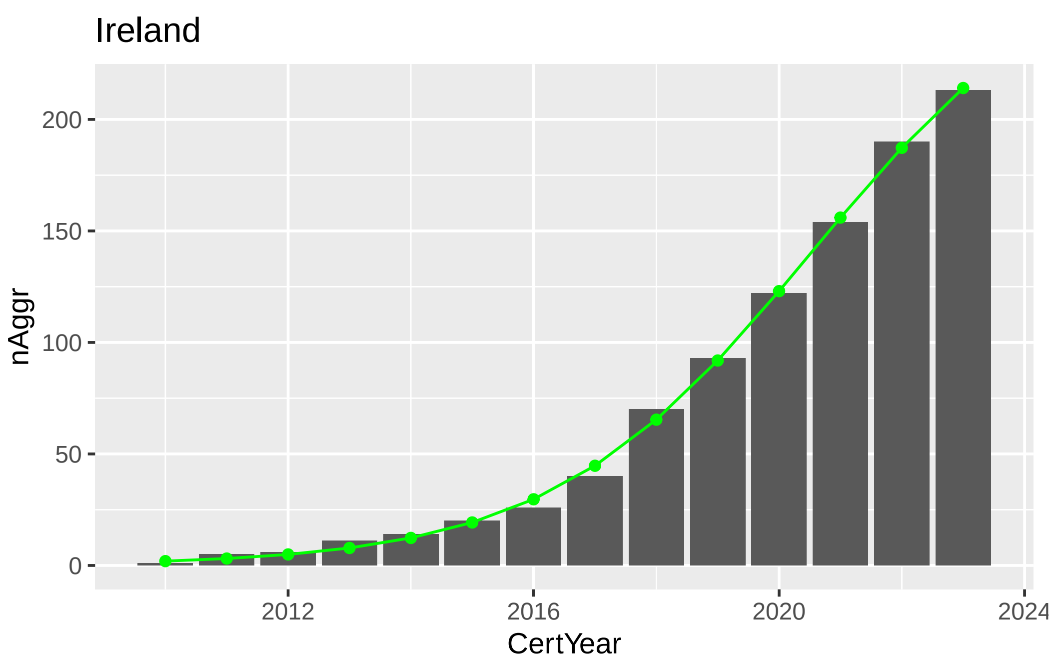

```
Hungary

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error  t value Pr(>|t|)
phi_1  281.0927    34.9492    8.043 6.21e-06 ***
phi_2 2021.0693     0.8879 2276.151  < 2e-16 ***
phi_3    3.4010     0.2712   12.540 7.39e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 4.543 on 11 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 5.113e-06


[[1]]
```




```
Ireland

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1 2.807e+02  1.209e+01   23.23 1.07e-10 ***
phi_2 2.021e+03  2.174e-01 9294.62  < 2e-16 ***
phi_3 2.120e+00  8.807e-02   24.07 7.27e-11 ***
```

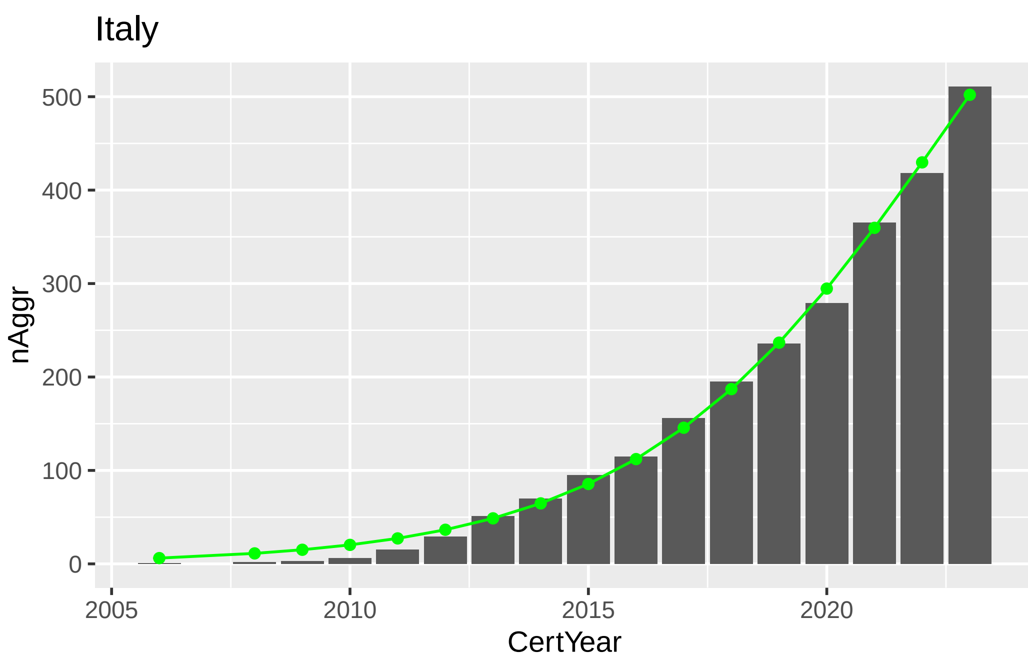
```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.87 on 11 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 8.055e-06


[[1]]
```




```
Italy

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error  t value Pr(>|t|)
```

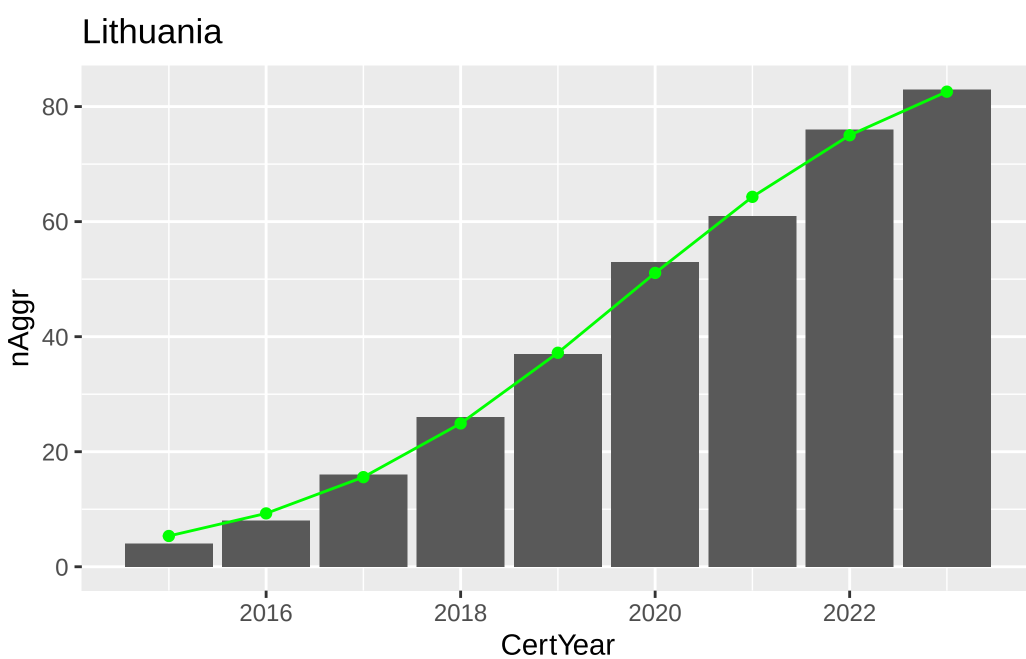
```
phi_1  960.5215   143.2141    6.707 9.99e-06 ***
phi_2 2022.6996     0.9375 2157.483  < 2e-16 ***
phi_3    3.3106     0.2170   15.258 4.06e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.21 on 14 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 8.285e-06
```

```
[[1]]
```



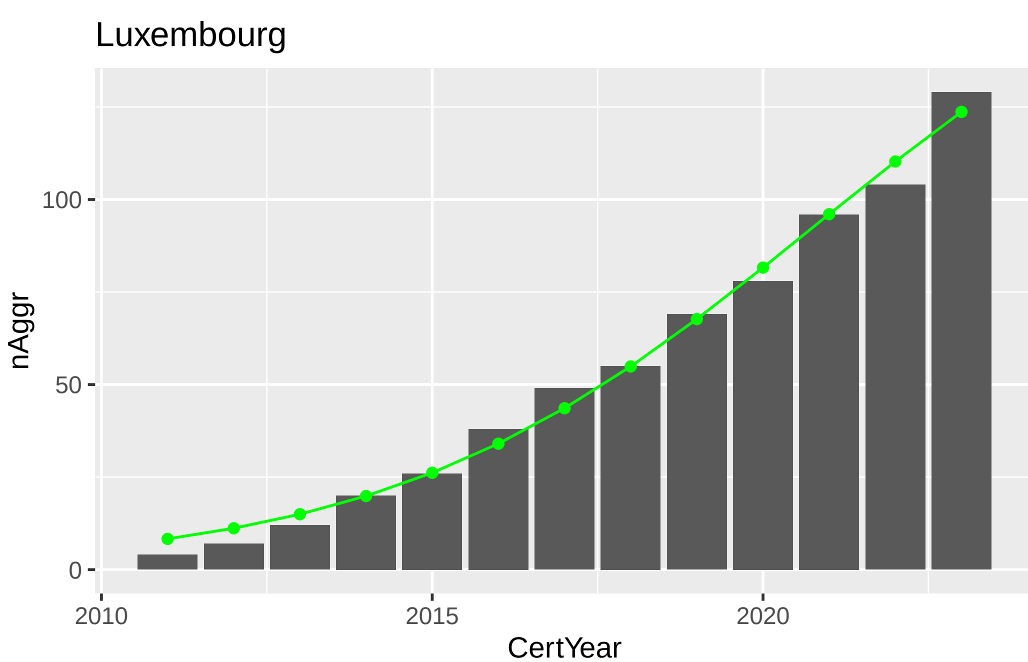

```
Lithuania

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)
```

```
Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1   94.2928     4.2677   22.09 5.62e-07 ***
phi_2 2019.7192     0.2075 9735.87  < 2e-16 ***
phi_3    1.6787     0.1181   14.21 7.59e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.853 on 6 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 3.649e-07


[[1]]
```

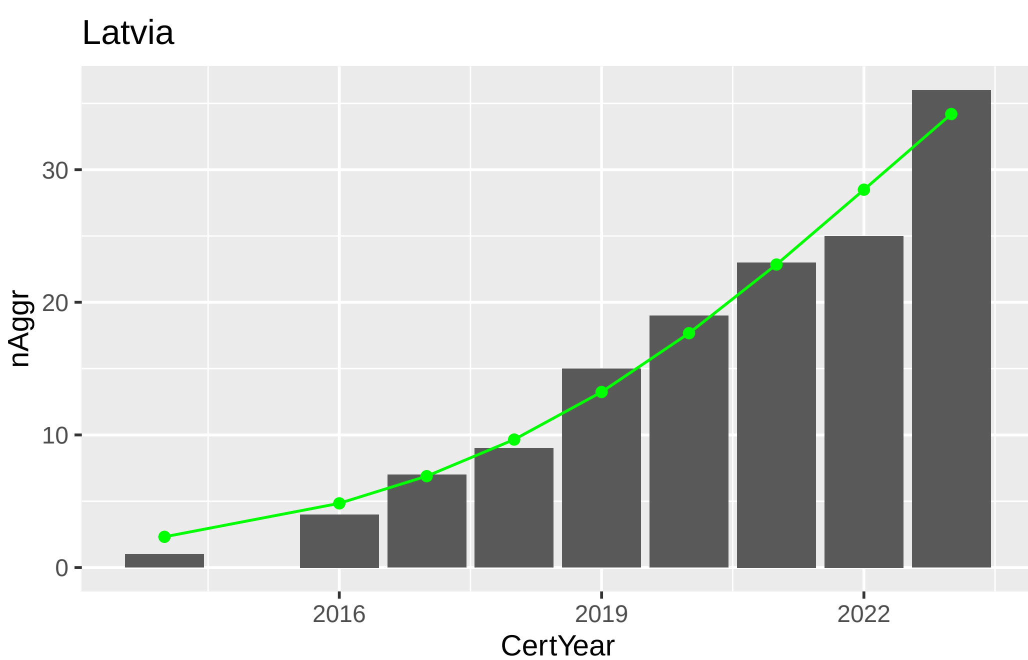



```
Luxembourg

Certified Buildings (STOCK)
```

```
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
      Estimate Std. Error  t value Pr(>|t|)
phi_1  184.578     27.656    6.674 5.54e-05 ***
phi_2 2020.741      1.033 1956.783  < 2e-16 ***
phi_3    3.188      0.342    9.324 3.01e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.149 on 10 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 1.359e-06


[[1]]
```



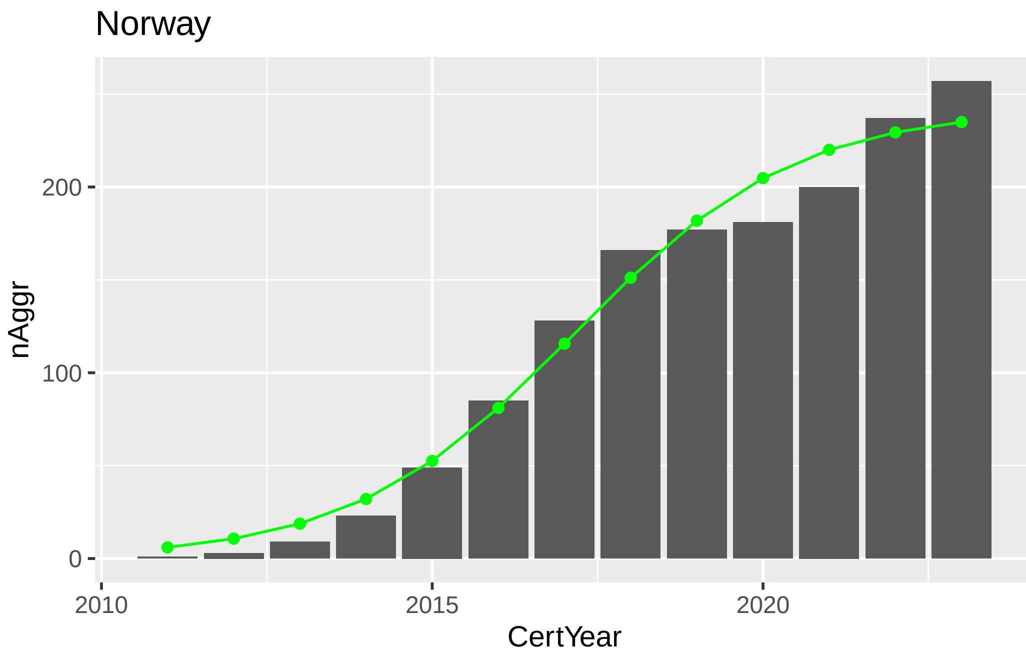
```
Latvia

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1   58.7439    24.4767    2.40  0.05330 .
phi_2 2022.1536     2.1553  938.24  < 2e-16 ***
phi_3    2.5534     0.6138    4.16  0.00594 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.967 on 6 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 1.648e-06


[[1]]
```



Norway

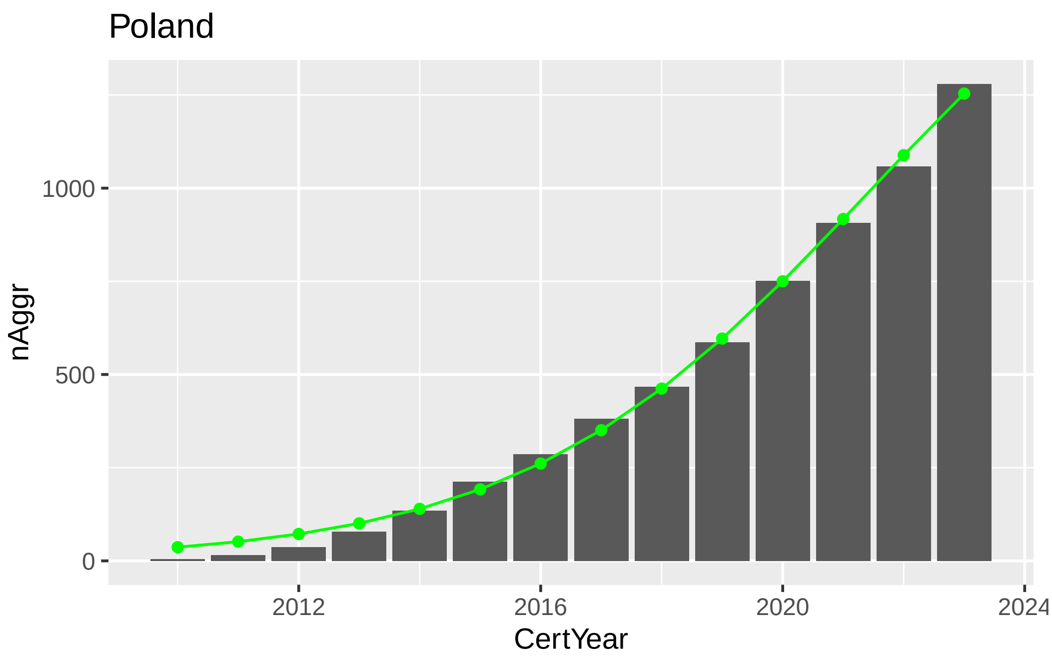
Certified Buildings (STOCK)

```
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error  t value Pr(>|t|)
phi_1  242.0613    13.7008   17.668 7.19e-09 ***
phi_2 2017.1486     0.3350 6021.403  < 2e-16 ***
phi_3    1.6746     0.2516    6.657 5.66e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.81 on 10 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 5.778e-06


[[1]]
```




```
Poland

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error  t value Pr(>|t|)
phi_1 1974.0715   212.3291    9.297 1.52e-06 ***
phi_2 2021.4071     0.6494 3112.930  < 2e-16 ***
phi_3    2.8742     0.1952   14.721 1.39e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.76 on 11 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 1.188e-06
```

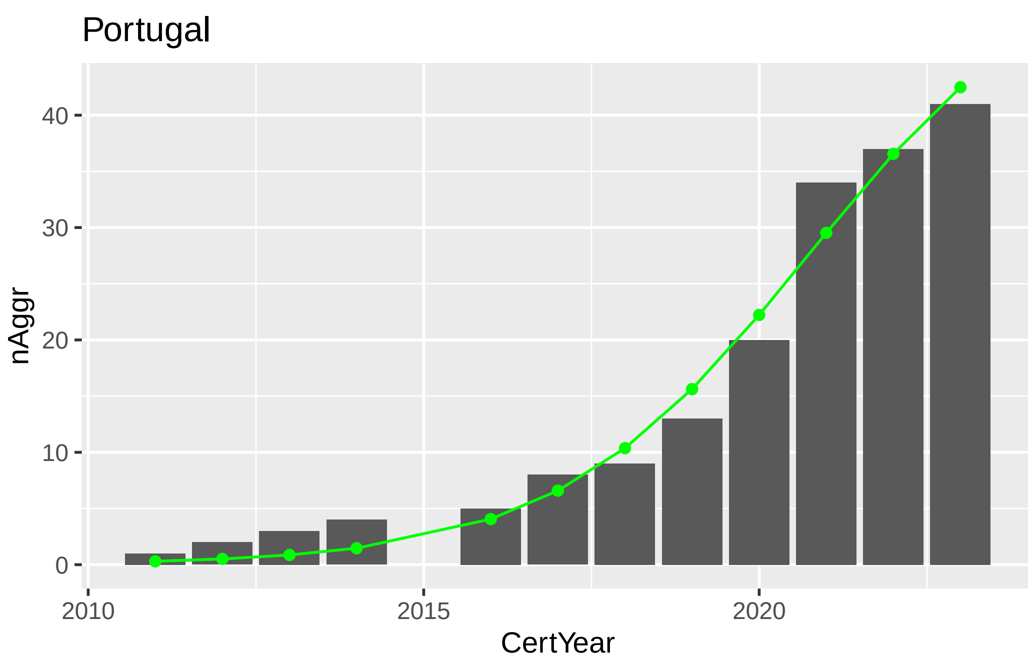
[[1]]




```
Portugal

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error  t value Pr(>|t|)
phi_1   55.1096    10.4333    5.282 0.000506 ***
phi_2 2020.7321     0.8518 2372.323  < 2e-16 ***
phi_3    1.8685     0.3654    5.114 0.000633 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.421 on 9 degrees of freedom

Number of iterations to convergence: 0
```

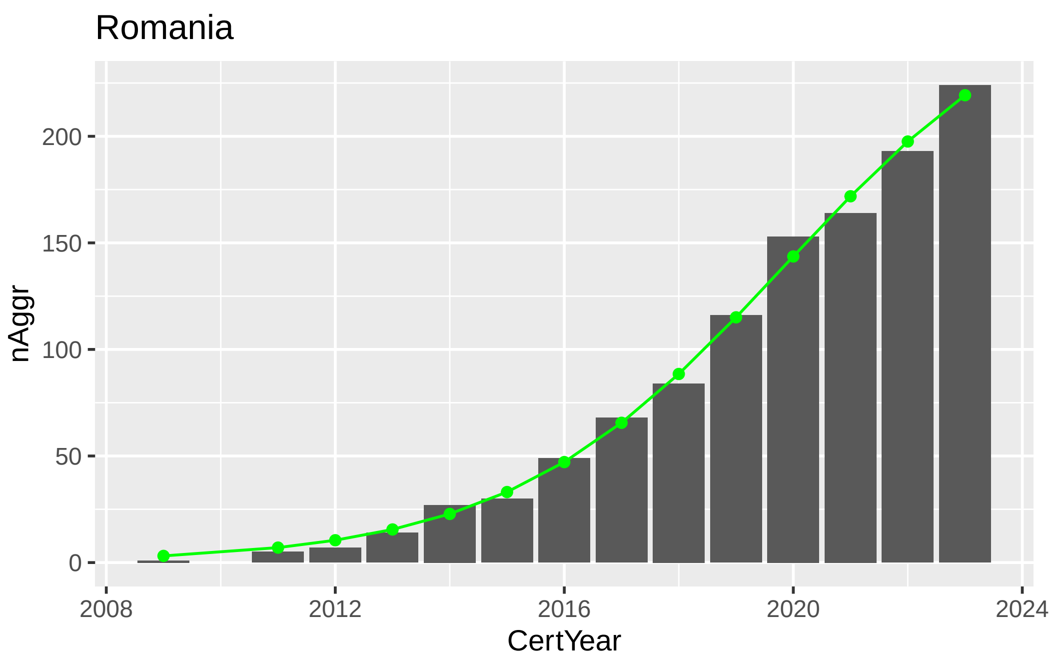
```
Achieved convergence tolerance: 9.104e-06
```

```
[[1]]
```




```
Romania

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1  279.3309    17.8569   15.64 7.32e-09 ***
phi_2 2019.8626     0.3755 5379.68  < 2e-16 ***
phi_3    2.4227     0.1601   15.13 1.04e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

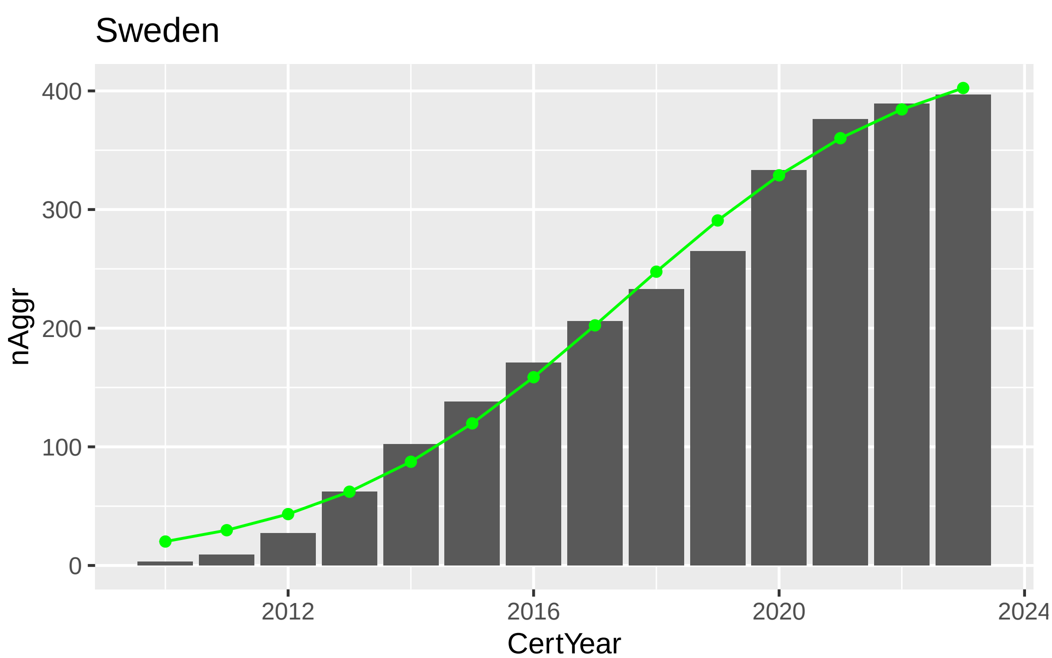
```
Residual standard error: 4.983 on 11 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 8.057e-07


[[1]]
```




```
Sweden

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
phi_1  443.4380    25.6124   17.31  2.5e-09 ***
phi_2 2017.4270     0.4008 5034.02  < 2e-16 ***
phi_3    2.4404     0.2431   10.04  7.1e-07 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.24 on 11 degrees of freedom

Number of iterations to convergence: 1
Achieved convergence tolerance: 1.826e-06
```

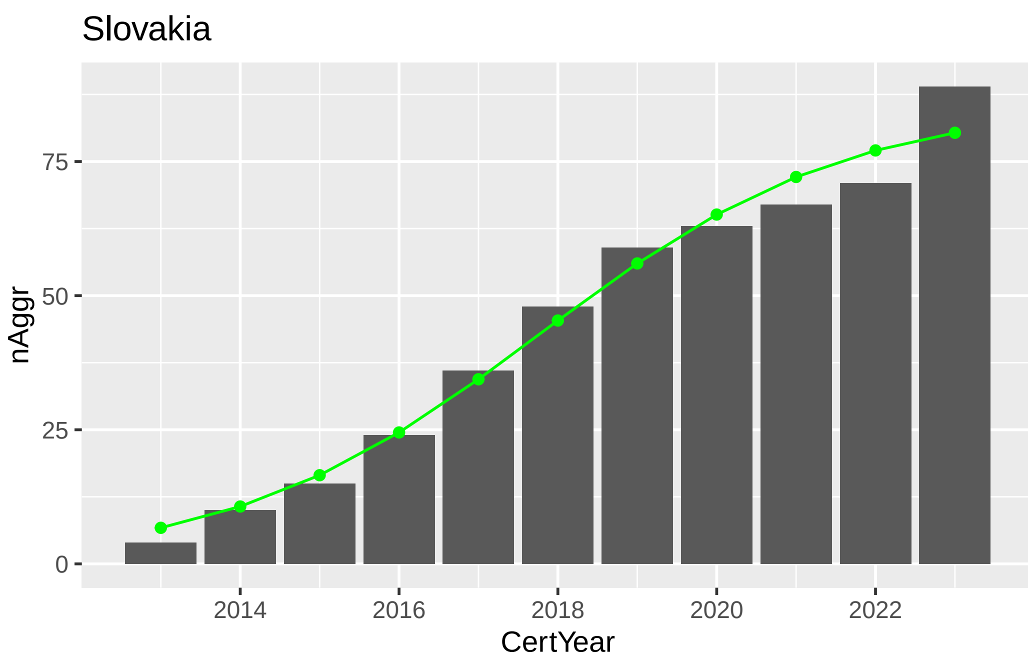
```
[[1]]
```




```
Slovakia

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)

Parameters:
       Estimate Std. Error  t value Pr(>|t|)
```

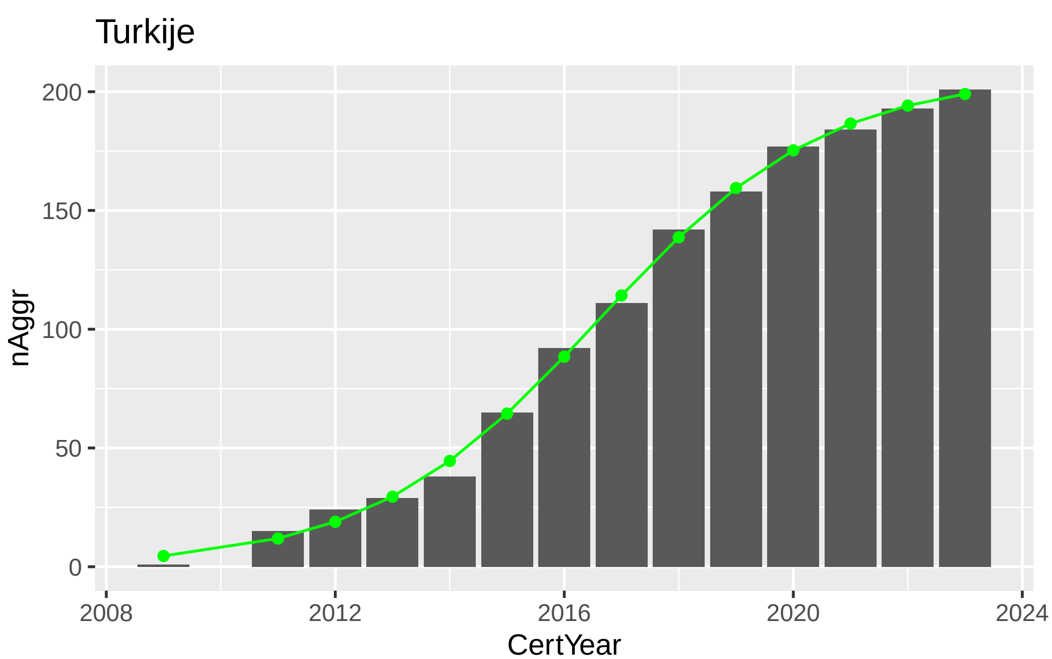
```
phi_1   85.7769     6.2093   13.814 7.28e-07 ***
phi_2 2017.7766     0.4255 4741.848  < 2e-16 ***
phi_3    1.9371     0.2973    6.516 0.000185 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.625 on 8 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 3.652e-06
```

```
[[1]]
```




```
Turkije

Certified Buildings (STOCK)
[[1]]

Formula: nAggr ~ SSlogis(CertYear, phi_1, phi_2, phi_3)
```

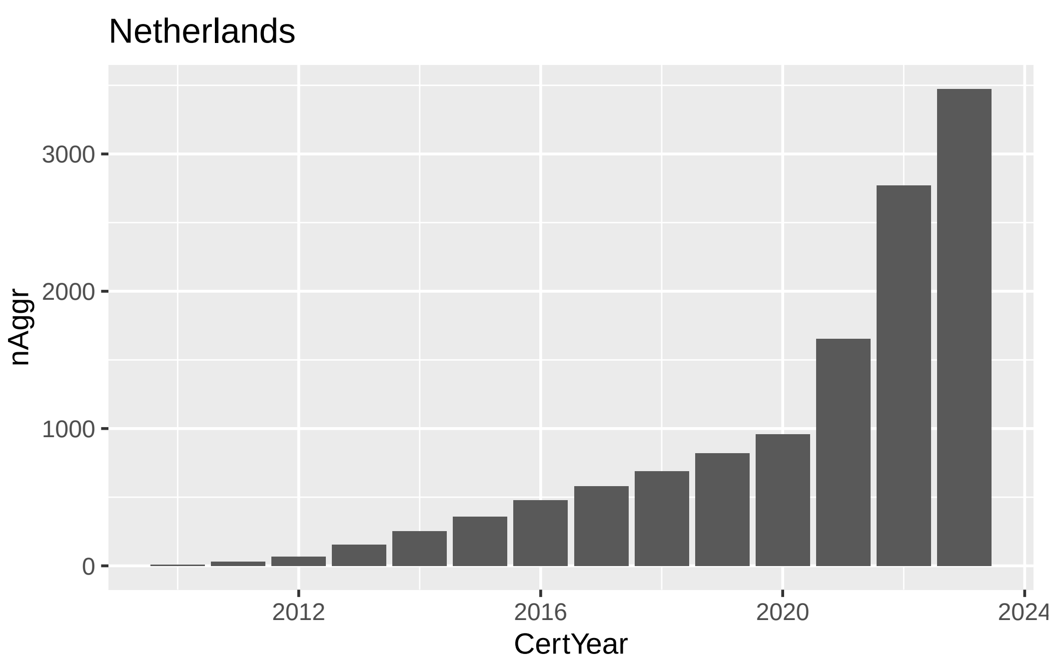
```
Parameters:
       Estimate Std. Error  t value Pr(>|t|)
phi_1 2.070e+02  3.545e+00    58.41 4.58e-15 ***
phi_2 2.017e+03  1.139e-01 17707.25  < 2e-16 ***
phi_3 1.997e+00  8.373e-02    23.85 8.01e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.586 on 11 degrees of freedom

Number of iterations to convergence: 1
Achieved convergence tolerance: 1.151e-06


[[1]]
```



6.2.1 Excluded Countries

The countries “Cyprus”, “Estonia”, “Croatia”, “Iceland”, “Malta”, “Serbia”, and “Slovenia” all had fewer than 30 certified buildings. This turns out to be too few observations for the estimation and we had to skip these countries. A special case is “The Netherlands”. With 3475 this country has a substantial number of certified green buildings. The plot, however shows that up to 2020, the increase is fairly moderate. After 2020, the numbers increase much more strongly. When we try to estimate a logistic curve for the Netherlands, the procedure fails. The parameter estimates do not converge but grow beyond bounds. We can only estimate two of the three parameters and have to fix the third one.

```
byYear <- byCountryYear %>%
  filter(country == "NL")
byYear$nAggr = sapply(byYear$CertYear,  function(x){
  sum(byYear$n[byYear$CertYear <= x])})

nonlinNL <- nls(nAggr ~ 3407*(1/(1+exp((phi2-CertYear)/phi3))),start = list(phi2=2024, phi3=
sum <- summary(nonlinNL)

byYear$fittedNL = fitted(nonlinNL)
ggplot(byYear) +
  geom_col(aes(x=CertYear,y=nAggr)) +
  ggtitle(countryNames["NL"])
```

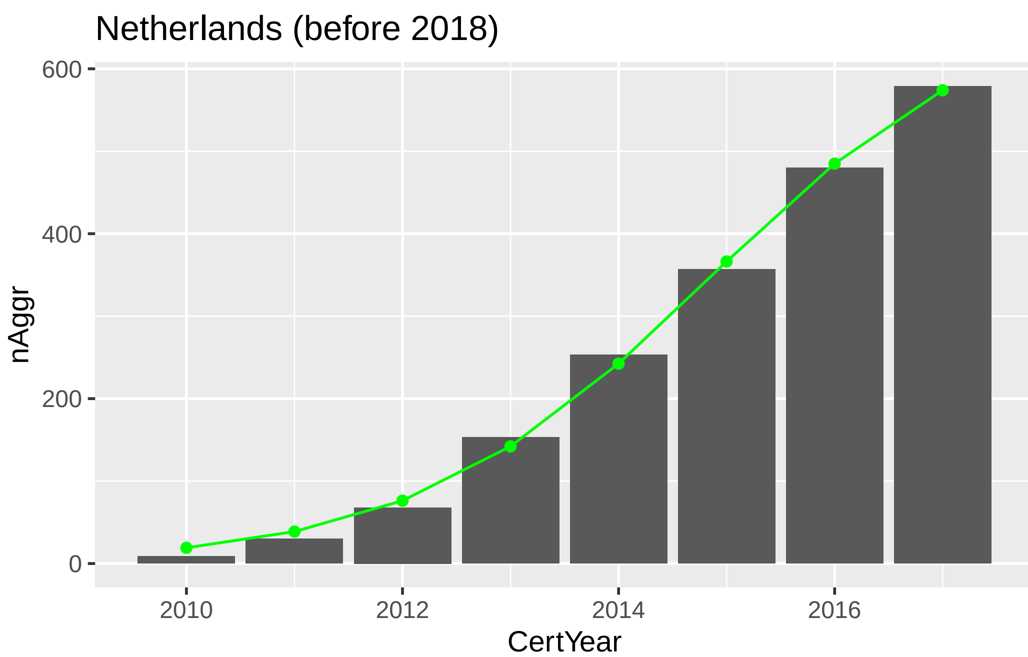


When we split the dataset into two period, before 2018 and 2018 and later, the model converges for each of the periods:

```r
byYear <- byCountryYear %>%
  filter(country == "NL") %>%
  filter(CertYear < 2018)
byYear$nAggr = sapply(byYear$CertYear,  function(x){
  sum(byYear$n[byYear$CertYear <= x])})

nonlinNL <- nls(nAggr ~ SSlogis(CertYear, phi1, phi2, phi3), data=byYear)
sum <- summary(nonlinNL)
sum
```

```
Formula: nAggr ~ SSlogis(CertYear, phi1, phi2, phi3)

Parameters:
      Estimate Std. Error  t value Pr(>|t|)
phi1 6.906e+02  3.560e+01    19.40 6.72e-06 ***
phi2 2.015e+03  1.830e-01 11008.47  < 2e-16 ***
phi3 1.358e+00  9.198e-02    14.77 2.58e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.07 on 5 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 1.783e-06
```

```r
byYear$fittedNL = fitted(nonlinNL)
ggplot(byYear) +
  geom_col(aes(x=CertYear,y=nAggr)) +
  geom_line(aes(x=CertYear, y=fittedNL), color="green") +
  geom_point(aes(x=CertYear, y=fittedNL), color="green") +
  ggtitle(paste(countryNames["NL"],"(before 2018)"))
```

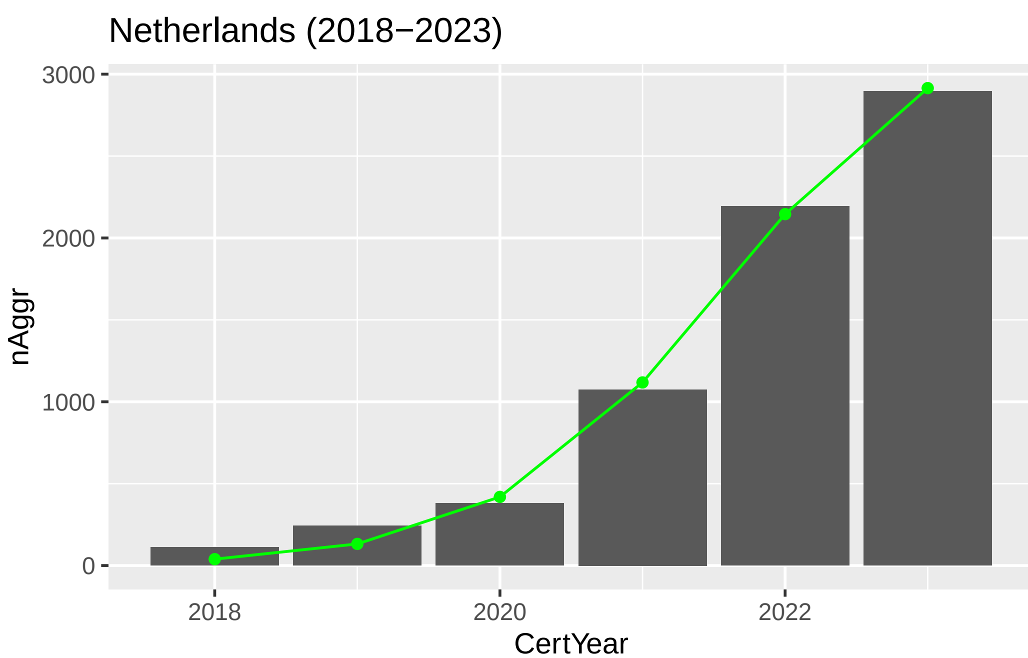



```r
byYear <- byCountryYear %>%
  filter(country == "NL") %>%
  filter(CertYear >= 2018)
byYear$nAggr = sapply(byYear$CertYear,  function(x){
  sum(byYear$n[byYear$CertYear <= x])})

nonlinNL <- nls(nAggr ~ SSlogis(CertYear, phi1, phi2, phi3), data=byYear)
sum <- summary(nonlinNL)
sum
```

```
Formula: nAggr ~ SSlogis(CertYear, phi1, phi2, phi3)

Parameters:
      Estimate Std. Error   t value Pr(>|t|)
phi1 3.407e+03  2.691e+02    12.664  0.00106 **
phi2 2.022e+03  1.774e-01 11393.552 1.49e-12 ***
phi3 8.016e-01  9.998e-02     8.017  0.00405 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 88.12 on 3 degrees of freedom
```

```
Number of iterations to convergence: 1
Achieved convergence tolerance: 2.937e-06
```

```
byYear$fittedNL = fitted(nonlinNL)
ggplot(byYear) +
  geom_col(aes(x=CertYear,y=nAggr)) +
  geom_line(aes(x=CertYear, y=fittedNL), color="green") +
  geom_point(aes(x=CertYear, y=fittedNL), color="green") +
  ggtitle(paste(countryNames["NL"],"(2018-2023)"))
```



We get very different results for the two time periods: For the first period the estimated upper limit is 690 certified green buildings, for the second one it is 3407, almost five times as many.

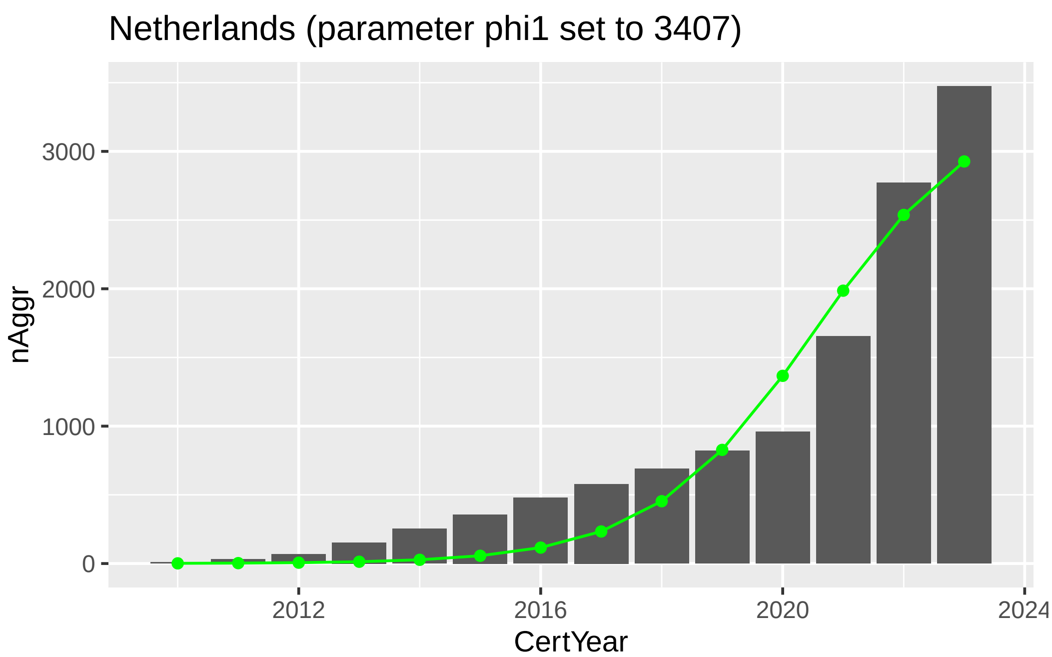
When we explicitly set the parameter `phi1` to the value 3407, that we got from the second time period, and only estimate the other two parameters,the model converges, but does not fit the observations very well. Therefore, we leave out The Netherlands from the further analysis and hope that data from additional years and from additional certification schemas will help resolve this problem.

```
byYear <- byCountryYear %>%
  filter(country == "NL")
byYear$nAggr = sapply(byYear$CertYear,  function(x){
  sum(byYear$n[byYear$CertYear <= x])})
```

```r
nonlinNL <- nls(nAggr ~ 3407*(1/(1+exp((phi2-CertYear)/phi3))),
  start = list(phi2=2024, phi3=2), data=byYear, trace=FALSE)
sum <- summary(nonlinNL)
sum
```

```
Formula: nAggr ~ 3407 * (1/(1 + exp((phi2 - CertYear)/phi3)))

Parameters:
      Estimate Std. Error  t value Pr(>|t|)
phi2 2020.5451     0.2586 7814.596  < 2e-16 ***
phi3    1.3585     0.2466    5.509 0.000134 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 303.5 on 12 degrees of freedom

Number of iterations to convergence: 17
Achieved convergence tolerance: 7.595e-06
```

```r
byYear$fittedNL = fitted(nonlinNL)
ggplot(byYear) +
  geom_col(aes(x=CertYear,y=nAggr)) +
  geom_line(aes(x=CertYear, y=fittedNL), color="green") +
  geom_point(aes(x=CertYear, y=fittedNL), color="green") +
  ggtitle(paste(countryNames["NL"],"(parameter phi1 set to 3407)"))
```



6.3 Interpreting the parameters

The estimated parameter values differ considerably between the countries. We estimate a saturation point (parameter ϕ_1) of just 28.56 certified green buildings for Denmark, but over 8,000 for Spain and almost 16,000 for Great Britain. On the other hand, for Denmark and Great Britain we estimate the lowest turning points (late 2015 and early 2016, respectively), for Spain with 2024 the highest one.

```
parameters
```

```
   country        phi1     phi2     phi3
1    TOTAL 35839.79192 2018.944 2.945705
2       AT    61.56865 2017.885 3.187544
3       BE   432.14174 2019.162 3.004766
4       BG    54.72004 2017.745 2.131665
5       CH    76.03678 2018.662 2.488630
6       CZ   297.56003 2018.634 2.612407
7       DE   398.78153 2016.922 2.469876
8       DK    28.56384 2015.892 1.912345
9       ES  8014.54081 2024.341 2.105183
10      FI   454.09965 2017.796 2.574699
11      FR  3461.40816 2021.006 2.687529
```

```
12        GB 15918.06740 2016.135 2.243569
13        GR    42.64520 2019.669 2.803938
14        HU   281.09266 2021.069 3.400984
15        IE   280.73159 2020.527 2.119614
16        IT   960.52149 2022.700 3.310613
17        LT    94.29277 2019.719 1.678673
18        LU   184.57788 2020.741 3.188375
19        LV    58.74388 2022.154 2.553400
20        NO   242.06128 2017.149 1.674614
21        PL  1974.07145 2021.407 2.874246
22        PT    55.10958 2020.732 1.868463
23        RO   279.33095 2019.863 2.422696
24        SE   443.43797 2017.427 2.440390
25        SK    85.77695 2017.777 1.937094
26        TR   207.04588 2016.586 1.997266
```

As we mentioned above, parameter `phi1` estimates the upper limit of the logistic function. We can interpret this as the estimated maximum number of certified green buildings. Of course, the maximum number of green buildings in a country is strongly influenced by the size of the country. Exactly the same growth process will yield a larger estimate for `phi1` in a larger country than in a smaller one. To make the results comparable, we divide the estimated values of `phi1` by the population of the country. In the following table we show `phi1ByPop`, the value of `phi1` per 100,000 inhabitants of the country. The table is in descending order of this variable.

```r
parameters %>%
  merge(PopulationByCountryPlus, by = "country") %>%
  mutate(phi1ByPop = 100000*phi1/pop) %>%
  arrange(desc(phi1ByPop)) %>%
  select(country, phi1, pop, phi1ByPop)
```

```
   country        phi1       pop  phi1ByPop
1       LU   184.57788    613894 30.0667353
2       GB 15918.06740  66647112 23.8841068
3       ES  8014.54081  46918951 17.0816709
4       FI   454.09965   5517919  8.2295454
5    TOTAL 35839.79192 630663666  5.6828693
6       IE   280.73159   4940311  5.6824679
7       PL  1974.07145  37972812  5.1986444
8       FR  3461.40816  67290471  5.1439797
9       SE   443.43797  10230185  4.3346036
10      LT    94.29277   2812200  3.3529894
```

```
11        BE   432.14174  13213979  3.2703377
12        LV    58.74388   1919968  3.0596280
13        HU   281.09266   9700272  2.8977812
14        CZ   297.56003  10649800  2.7940434
15        NO   242.06128   9951007  2.4325305
16        IT   960.52149  59816673  1.6057755
17        SK    85.77695   5450421  1.5737674
18        RO   279.33095  19414458  1.4387780
19        CH    76.03678   8544527  0.8898887
20        BG    54.72004   6664177  0.8211072
21        AT    61.56865   8858775  0.6950018
22        PT    55.10958  10333496  0.5333101
23        DK    28.56384   5806081  0.4919643
24        DE   398.78153  83019213  0.4803485
25        GR    42.64520  10724599  0.3976391
26        TR   207.04588  82003882  0.2524830
```

Because of its small size, Luxembourg with just 129 certified green buildings has by far the largest value of `phi1ByPop`. In some countries at the end of the ranking, like Austria, Germany, and Denmark, green building certification is dominated by the DGNB system, which is not yet included in our analysis. With a more complete dataset, these countries will most likely reach considerably higher values.

The slope of the function at the inflection point ($x = \phi_2$) is $\frac{\phi_1}{2\phi_3}$. Standardized to the same saturation level, this slope is just $\frac{1}{2\phi_3}$. We calculate this standardized slope for each of the countries and store it in the column `slope` of the data frame `parameters`. In addition, we also calculate at what year the curve reaches 10% of the saturation level. This is calculated by $x = \phi_2 - \phi_3 \ln(\frac{1}{a} - 1)$ with $a = 0.1$ and stored in column `tens`.

```
parameters$slope <- 1/(2*parameters$phi3)
parameters$tens <- parameters$phi2-parameters$phi3*log(9)
```

The parameter estimates and these indicators are shown in the following table.

```
parameters %>%
  arrange(tens) %>%
  select(country, phi3, slope, tens)
```

```
   country     phi3     slope     tens
1       AT 3.187544 0.1568606 2010.882
2       GB 2.243569 0.2228592 2011.206
3       DE 2.469876 0.2024393 2011.496
```

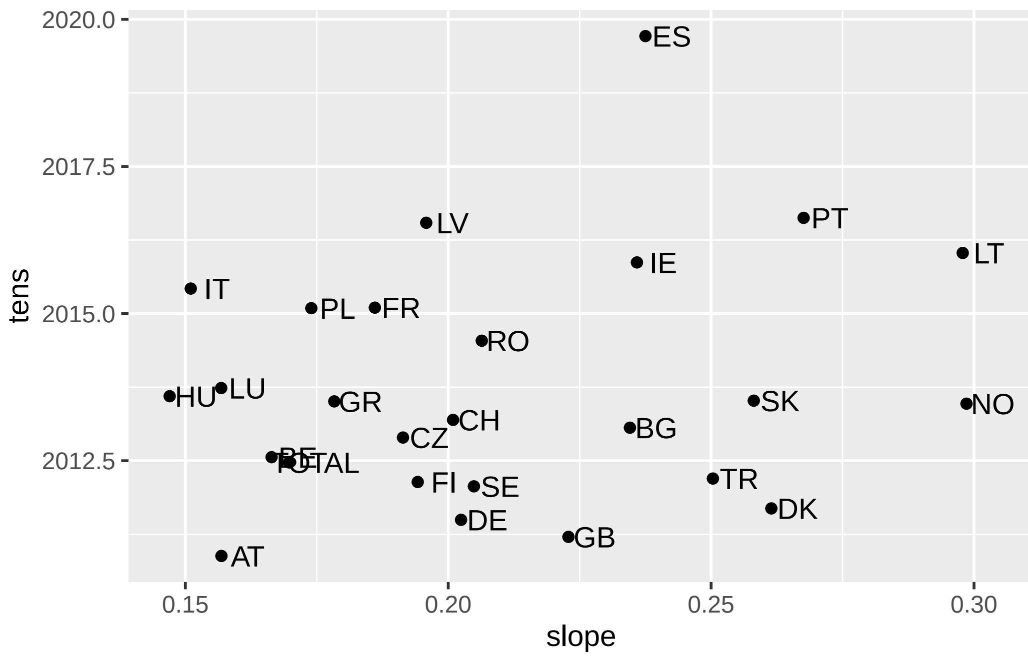
```
4        DK 1.912345 0.2614591 2011.690
5        SE 2.440390 0.2048853 2012.065
6        FI 2.574699 0.1941975 2012.138
7        TR 1.997266 0.2503422 2012.197
8     TOTAL 2.945705 0.1697386 2012.471
9        BE 3.004766 0.1664023 2012.560
10       CZ 2.612407 0.1913944 2012.894
11       BG 2.131665 0.2345584 2013.061
12       CH 2.488630 0.2009138 2013.194
13       NO 1.674614 0.2985763 2013.469
14       GR 2.803938 0.1783206 2013.508
15       SK 1.937094 0.2581186 2013.520
16       HU 3.400984 0.1470163 2013.597
17       LU 3.188375 0.1568197 2013.735
18       RO 2.422696 0.2063817 2014.539
19       PL 2.874246 0.1739587 2015.092
20       FR 2.687529 0.1860445 2015.101
21       IT 3.310613 0.1510294 2015.425
22       IE 2.119614 0.2358920 2015.869
23       LT 1.678673 0.2978544 2016.031
24       LV 2.553400 0.1958174 2016.543
25       PT 1.868463 0.2675997 2016.627
26       ES 2.105183 0.2375090 2019.716
```

We see clear differences between the countries in the estimation results. Our measure for the standardized saturation level, `phi1ByPop`, shows that LEED and BREEAM certifications are strong in a small number of countries. Only Luxembourg, Great Britain, Spain, and Finland show higher standardized saturation levels than all the countries in average. The other 21 countries all show smaller values. It will be interesting to see when we use a wider set of certification schemas.

The indicator `tens` shows, when a country reached 10% of its estimated saturation level. `slope` shows how strongly the growth process is. The table again shows quite different growth processes. Austria, for example, is the first country that reached the 10% mark, but also has one of the smallest slopes. Countries like Spain, Portugal, and Lithuania, on the other hand, reach the 10% mark much later, but have a much steeper slope.

The following scatterplot positions the countries in these two dimensions.

```r
ggplot(parameters, aes(x=slope, y=tens)) +
  geom_point() +
  geom_text(label=parameters$country, nudge_x=0.005)
```



7 Conclusion

As has been said in the introduction, the aim of this discussion paper is to demonstrate that a spatio-temporal analysis of the diffusion of green building certifications in Europe is feasible. Since we only have data for LEED and BREEAM available at this point, we have to refrain from more substantial interpretations of the results. The analysis shows, however, that such an analysis will most likely lead to valuable new insights. As we did show, there was a massive growth in the number of green buildings in Europe during the observation period. This process was not uniform, but resulted in a concentrated pattern of certified green buildings with significant spatial clusters. More in depth analysis seems needed to clearly understand the dynamics of this spatio-temporal process, and to identify policy instruments that can help to stimulate a stronger and more wide spread application of green technologies in the construction sector and in the buildings we use. Since buildings and their construction are critical elements for the transition to more sustainability, this knowledge is of great importance, in our view.

References

Childs, Dylan Z, Bethan J Hindle, and Philip H Warren. n.d. *Introductory Biostatistics with r*. https://tuos-bio-data-skills.github.io/intro-stats-book/.

Shi, Yingling, and Xinping Liu. 2019. "Research on the Literature of Green Building Based on the Web of Science: A Scientometric Analysis in CiteSpace (2002–2018)." *Sustainability* 11 (13): 3716. https://doi.org/10.3390/su11133716.

Wang, Wei, Shoujian Zhang, Yikun Su, and Xinyang Deng. 2018. "Key Factors to Green Building Technologies Adoption in Developing Countries: The Perspective of Chinese Designers." *Sustainability* 10 (11): 4135. https://doi.org/10.3390/su10114135.

Session Info

```r
sessioninfo::session_info()
```

```
- Session info ---------------------------------------------------------------
 setting  value
 version  R version 4.4.1 (2024-06-14 ucrt)
 os       Windows 11 x64 (build 26100)
 system   x86_64, mingw32
 ui       RTerm
 language (EN)
 collate  English_Austria.utf8
 ctype    English_Austria.utf8
 tz       Europe/Vienna
 date     2024-11-12
 pandoc   3.2 @ C:/Program Files/RStudio/resources/app/bin/quarto/bin/tools/ (via rmarkdown)

- Packages -------------------------------------------------------------------
 package      * version  date (UTC) lib source
 abind          1.4-8    2024-09-12 [1] CRAN (R 4.4.1)
 assertthat     0.2.1    2019-03-21 [1] CRAN (R 4.4.1)
 backports      1.5.0    2024-05-23 [1] CRAN (R 4.4.0)
 base64enc      0.1-3    2015-07-28 [1] CRAN (R 4.4.0)
 bibtex         0.5.1    2023-01-26 [1] CRAN (R 4.4.1)
 boot           1.3-30   2024-02-26 [2] CRAN (R 4.4.1)
 cellranger     1.1.0    2016-07-27 [1] CRAN (R 4.4.1)
 class          7.3-22   2023-05-03 [2] CRAN (R 4.4.1)
 classInt       0.4-10   2023-09-05 [1] CRAN (R 4.4.1)
 cli            3.6.3    2024-06-21 [1] CRAN (R 4.4.1)
 codetools      0.2-20   2024-03-31 [2] CRAN (R 4.4.1)
 colorspace     2.1-1    2024-07-26 [1] CRAN (R 4.4.1)
 countrycode    1.6.0    2024-03-22 [1] CRAN (R 4.4.1)
 crosstalk      1.2.1    2023-11-23 [1] CRAN (R 4.4.1)
```

```
curl           5.2.3    2024-09-20 [1] CRAN (R 4.4.1)
data.table     1.16.0   2024-08-27 [1] CRAN (R 4.4.1)
DBI            1.2.3    2024-06-02 [1] CRAN (R 4.4.1)
deldir         2.0-4    2024-02-28 [1] CRAN (R 4.4.0)
dichromat      2.0-0.1  2022-05-02 [1] CRAN (R 4.4.0)
digest         0.6.37   2024-08-19 [1] CRAN (R 4.4.1)
dplyr        * 1.1.4    2023-11-17 [1] CRAN (R 4.4.1)
e1071          1.7-16   2024-09-16 [1] CRAN (R 4.4.1)
eurostat     * 4.0.0    2023-12-19 [1] CRAN (R 4.4.1)
evaluate       1.0.1    2024-10-10 [1] CRAN (R 4.4.1)
fansi          1.0.6    2023-12-08 [1] CRAN (R 4.4.1)
farver         2.1.2    2024-05-13 [1] CRAN (R 4.4.1)
fastmap        1.2.0    2024-05-15 [1] CRAN (R 4.4.1)
forcats      * 1.0.0    2023-01-29 [1] CRAN (R 4.4.1)
gdata          3.0.1    2024-10-22 [1] CRAN (R 4.4.1)
generics       0.1.3    2022-07-05 [1] CRAN (R 4.4.1)
ggplot2      * 3.5.1    2024-04-23 [1] CRAN (R 4.4.1)
glue           1.8.0    2024-09-30 [1] CRAN (R 4.4.1)
gmodels      * 2.19.1   2024-03-06 [1] CRAN (R 4.4.1)
gtable         0.3.6    2024-10-25 [1] CRAN (R 4.4.1)
gtools         3.9.5    2023-11-20 [1] CRAN (R 4.4.1)
here           1.0.1    2020-12-13 [1] CRAN (R 4.4.1)
hms            1.1.3    2023-03-21 [1] CRAN (R 4.4.1)
htmltools      0.5.8.1  2024-04-04 [1] CRAN (R 4.4.1)
htmlwidgets    1.6.4    2023-12-06 [1] CRAN (R 4.4.1)
httr           1.4.7    2023-08-15 [1] CRAN (R 4.4.1)
httr2          1.0.6    2024-11-04 [1] CRAN (R 4.4.1)
ISOweek        0.6-2    2011-09-07 [1] CRAN (R 4.4.1)
jsonlite       1.8.9    2024-09-20 [1] CRAN (R 4.4.1)
KernSmooth     2.23-24  2024-05-17 [2] CRAN (R 4.4.1)
knitr          1.48     2024-07-07 [1] CRAN (R 4.4.1)
labeling       0.4.3    2023-08-29 [1] CRAN (R 4.4.0)
lattice        0.22-6   2024-03-20 [2] CRAN (R 4.4.1)
lazyeval       0.2.2    2019-03-15 [1] CRAN (R 4.4.1)
leafem         0.2.3    2023-09-17 [1] CRAN (R 4.4.1)
leaflet        2.2.2    2024-03-26 [1] CRAN (R 4.4.1)
leafsync       0.1.0    2019-03-05 [1] CRAN (R 4.4.1)
lifecycle      1.0.4    2023-11-07 [1] CRAN (R 4.4.1)
lubridate    * 1.9.3    2023-09-27 [1] CRAN (R 4.4.1)
lwgeom         0.2-14   2024-02-21 [1] CRAN (R 4.4.1)
magrittr       2.0.3    2022-03-30 [1] CRAN (R 4.4.1)
MASS           7.3-60.2 2024-04-26 [2] CRAN (R 4.4.1)
munsell        0.5.1    2024-04-01 [1] CRAN (R 4.4.1)
```

```
pillar         1.9.0    2023-03-22 [1] CRAN (R 4.4.1)
pkgconfig      2.0.3    2019-09-22 [1] CRAN (R 4.4.1)
plotly       * 4.10.4   2024-01-13 [1] CRAN (R 4.4.1)
plyr           1.8.9    2023-10-02 [1] CRAN (R 4.4.1)
png            0.1-8    2022-11-29 [1] CRAN (R 4.4.0)
proxy          0.4-27   2022-06-09 [1] CRAN (R 4.4.1)
purrr        * 1.0.2    2023-08-10 [1] CRAN (R 4.4.1)
R6             2.5.1    2021-08-19 [1] CRAN (R 4.4.1)
rappdirs       0.3.3    2021-01-31 [1] CRAN (R 4.4.1)
raster         3.6-26   2023-10-14 [1] CRAN (R 4.4.1)
RColorBrewer   1.1-3    2022-04-03 [1] CRAN (R 4.4.0)
Rcpp           1.0.13-1 2024-11-02 [1] CRAN (R 4.4.1)
readr        * 2.1.5    2024-01-10 [1] CRAN (R 4.4.1)
readxl         1.4.3    2023-07-06 [1] CRAN (R 4.4.1)
RefManageR     1.4.0    2022-09-30 [1] CRAN (R 4.4.1)
regions        0.1.8    2021-06-21 [1] CRAN (R 4.4.1)
rlang          1.1.4    2024-06-04 [1] CRAN (R 4.4.1)
rmarkdown      2.29     2024-11-04 [1] CRAN (R 4.4.1)
rprojroot      2.0.4    2023-11-05 [1] CRAN (R 4.4.1)
rstudioapi     0.17.1   2024-10-22 [1] CRAN (R 4.4.1)
s2             1.1.7    2024-07-17 [1] CRAN (R 4.4.1)
scales         1.3.0    2023-11-28 [1] CRAN (R 4.4.1)
sessioninfo    1.2.2    2021-12-06 [1] CRAN (R 4.4.1)
sf           * 1.0-18   2024-10-11 [1] CRAN (R 4.4.1)
sp             2.1-4    2024-04-30 [1] CRAN (R 4.4.1)
spData       * 2.3.3    2024-09-02 [1] CRAN (R 4.4.1)
spdep        * 1.3-6    2024-09-13 [1] CRAN (R 4.4.1)
stars          0.6-6    2024-07-16 [1] CRAN (R 4.4.1)
stringi        1.8.4    2024-05-06 [1] CRAN (R 4.4.0)
stringr      * 1.5.1    2023-11-14 [1] CRAN (R 4.4.1)
terra          1.7-83   2024-10-14 [1] CRAN (R 4.4.1)
tibble       * 3.2.1    2023-03-20 [1] CRAN (R 4.4.1)
tidyr        * 1.3.1    2024-01-24 [1] CRAN (R 4.4.1)
tidyselect     1.2.1    2024-03-11 [1] CRAN (R 4.4.1)
tidyverse    * 2.0.0    2023-02-22 [1] CRAN (R 4.4.1)
timechange     0.3.0    2024-01-18 [1] CRAN (R 4.4.1)
tmap         * 3.3-4    2023-09-12 [1] CRAN (R 4.4.1)
tmaptools      3.1-1    2021-01-19 [1] CRAN (R 4.4.1)
tzdb           0.4.0    2023-05-12 [1] CRAN (R 4.4.1)
units          0.8-5    2023-11-28 [1] CRAN (R 4.4.1)
utf8           1.2.4    2023-10-22 [1] CRAN (R 4.4.1)
vctrs          0.6.5    2023-12-01 [1] CRAN (R 4.4.1)
viridisLite    0.4.2    2023-05-02 [1] CRAN (R 4.4.1)
```

```
withr          3.0.2     2024-10-28 [1] CRAN (R 4.4.1)
wk             0.9.4     2024-10-11 [1] CRAN (R 4.4.1)
xfun           0.49      2024-10-31 [1] CRAN (R 4.4.1)
XML            3.99-0.17 2024-06-25 [1] CRAN (R 4.4.1)
xml2           1.3.6     2023-12-04 [1] CRAN (R 4.4.1)
yaml           2.3.10    2024-07-26 [1] CRAN (R 4.4.1)

[1] C:/Users/gmaier/AppData/Local/R/win-library/4.4
[2] C:/Program Files/R/R-4.4.1/library

------------------------------------------------------------------------------
```