University of Padua
Departments of Mathematics

Master Degree in Data Science

Low-Carbon Electricity Generation: A Comprehensive Analysis

Valerio Rocca, 2094861
Giacomo Virginio, 2076681

Academic Year 2022/2023

Contents

Abstract

Given the escalating concerns surrounding climate change and the ever-increasing demand for electricity, the generation of low-carbon electricity has assumed utmost importance in curbing greenhouse gas emissions and fostering a cleaner and more sustainable future. This report presents a comprehensive analysis of low-carbon electricity generation, focusing on its history, current trends, and future prospects.

The analysis begins by examining quantitatively the historical context of low-carbon electricity generation, tracing its origins and evolution over time and by proposing insights on different sources and world areas.

Next, the report delves into the current trends and status of low-carbon electricity generation. By exploiting a tailored yet simple measure, the Green Score, it provides a detailed overview of various sources such as nuclear, solar, wind, and hydropower, along with their contributions to the global energy mix.

Finally, various models with different transformations on the data are proposed, in order to find the best performing ones: models' coefficients are analyzed to learn the impact of the different independent variables on features that measure pollution and the adoption of different renewable energy sources in the countries over the last two decades.

Chapter 1: Obtaining data

We could not find a single dataset containing all the information of interest. Thus, the project's first step is merging multiple datasets into one. The primary one is the *Energy dataset* by Our World in Data (from now onwards referred to as "OWID") [1], which contains various time series for each world country regarding energy and electricity production and consumption.

```
# Import the libraries
library(dplyr)
library(tidyr)
library(ggplot2)
library(viridis)
library(rworldmap)
library(glmnet)
library(readxl)
library(gridExtra)
library(corrplot)
library(knitr)
```

```
# Import the Energy dataset
main = read.csv("datasets//Total_energy_data.csv")
```

We then merge the following datasets into it:

- *GDP (constant 2015 US$)* by World Bank [2], which contains the time series of the GDP in each country from 1960 to 2021, measured in constant 2015 USA dollars.
- *Land Area* by OWID [3], which contains the time series of the land area of each country from 1961 to 2021, measured in squared kilometers;
- *Agricultural land* by OWID [4], which contains the time series of the share of land area used for agriculture in each country from 1961 to 2018;
- *Urbanization rate* by OWID [5], which contains the time series of the share of people living in urban areas in each country from 1960 to 2020;
- *Human Development Index* by OWID [6], which contains the time series of the HDI for each country from 1990 to 2021;
- *Death rate from air pollution* by OWID [7], which after filtering contains the time series of the number of deaths from outdoor particulate matter per 100,000 population in each country from 1990 to 2019;
- *Coal proved reserves* by OWID [8], which contains the reserves of coal in each country in 2021, measured in tonnes;
- *Oil proved reserves* by OWID [9], which contains the reserves of oil in each country in 2020, measured in tonnes;
- *Natural gas proved reserves* by OWID [10], which contains the time series of the reserves of natural gas in each country from 1980 to 2020;
- *Uranium proved reserves* by OECD [11], which contains uranium reserves in each country in 2019, measured in tonnes.

Merging presents **three critical issues** listed below, together with the implemented solutions.

1. Time series are recorded for **different years**. We tackled this problem by merging through left join: all the rows of the *Energy dataset* are included, while rows from the other datasets are included if there is a match; otherwise, a NA value is added. In order to perform correctly the left join, we need to remove the countries in the *Energy dataset* without an ISO code. This is the case for some semi-autonomous territories inside of a country (e.g., Wake Island), countries that no longer exist (e.g., Yugoslavia), and country groupings (e.g., OPEC countries). Therefore, we decide to remove those observations.

2. Coal, oil, and uranium reserves are **stationary values**, as time series for those variables are not publicly available. Therefore, we approached the issue by considering the reserves fixed through time, as it does not affect the quality of the analyses.
3. *GDP (constant 2015 US$)* dataset contains **a column for each year**, while the other time series datasets format the years using a specific variable. Therefore, we modify the structure of *GDP* to fit the others'.

```
# Delete units from "main" without an ISO code
main = main[main$iso_code!='',]

# Creation of a function able to automatically join datasets from OWID
join_owid = function(main_data, secondary_dataset_link){
  place_data = read.csv(paste(secondary_dataset_link))
  main_data = left_join(main_data, select(place_data, -c("Entity")),
                 by = c("iso_code" = "Code", "year" = "Year"))
  rm(place_data)
  return(main_data)
}

# Import and merging of the following OWID variables: country area, HDI and urbanization rate
main = join_owid(main, "datasets//country_areas.csv")
main = join_owid(main, "datasets//human_development_index.csv")
main = join_owid(main, "datasets//urbanization_rate.csv")

# Import, selection and merging of deaths from air pollution
 death_rates = read.csv("datasets//death_rates_from_air_pollution.csv")[, c(2,3,5)]
 colnames(death_rates) = c("Code", "Year", "particulate_pollution")
 main = left_join(main, death_rates,
                  by = c("iso_code" = "Code", "year" = "Year"))
rm(death_rates)

# Import and merging share of land for agricultural use
main = join_owid(main, "datasets//share_of_land_area_used_for_agriculture.csv")


# Import, filtering and merging of coal proved reserves
coal_res = read.csv("datasets//coal_proved_reserves.csv")
coal_res = coal_res[coal_res$Year!="2020",]
colnames(coal_res) = c("Entity", "Code", "Year", "coal_reserves_2021")
main = left_join(main, select(coal_res, -c("Entity","Year")),
                 by = c("iso_code" = "Code"))
rm(coal_res)

# Import and merging oil proved reserves
main = join_owid(main, "datasets//oil_proved_reserves.csv")

# Import and merging of uranium reserves
uranium_res = read.csv("datasets//uranium_proved_reserves.txt", sep = "\t")
colnames(uranium_res) = c("V1","V2", "uranium_reserves_2019", "V4")
uranium_res$uranium_reserves_2019 = as.numeric(gsub(",","",uranium_res$uranium_reserves_2019))
uranium_res$V1 = sub(".","",uranium_res$V1)
main = left_join(main, select(uranium_res, -c("V2", "V4")),
                 by = c("country" = "V1"))
rm(uranium_res)
```

```r
# Import and merging natural gas proved reserves
main = join_owid(main, "datasets//natural_gas_proved_reserves.csv")

# Import the GDP dataset
gdp = read_excel("datasets//gdp_constant_2015_dollars.xlsx")[,4:66]
# Structure modification
colnames(gdp) = c("code", 1960:2021)
gdp = gather(gdp, key = "year", value = "gdp", -code)
gdp$year = as.integer(gdp$year)
# Merging
main = left_join(main, gdp, by = c("iso_code" = "code", "year" = "year"))
rm(gdp)
```

Chapter 2: Data pre-processing

In this section, we present the pre-processing activities performed.

1. **Units selection**, computed over the *Energy dataset*. Already partially computed and explained in the previous section, in this phase we also removed two regions with too many missing values: Antarctica and Western Sahara.
2. **Feature selection**, computed over the *Energy dataset*. From the original 129 variables, we kept only 36 relevant for the analyses.
3. **Feature renaming**, computed over *main*, as the features merged to the *Energy dataset* have inconvenient names.
4. **Feature addition**, computed over *main* in paragraph 4.3. The new categorical variable groups the world countries into six macroregions.
5. **Cleaning NA values**, computed over *main*. It consists of the substitution of NA values for reserves data to 0 and, for GDP, of '..' to NA.

```r
# 1. Units selection: remove Antarctica and Western Sahara
main = filter(main, iso_code != "ATA", iso_code != "ESH")

# 2. Feature selection
main = select(main, -c("gdp.x", "biofuel_cons_change_pct",
                       "biofuel_cons_change_twh", "biofuel_cons_per_capita",
                       "biofuel_elec_per_capita", "biofuel_consumption",
                       "biofuel_electricity", "biofuel_share_elec",
                       "biofuel_share_energy", "coal_cons_change_pct",
                       "coal_cons_change_twh", "coal_cons_per_capita",
                       "coal_elec_per_capita", "coal_prod_change_pct",
                       "coal_prod_change_twh", "coal_prod_per_capita",
                       "coal_consumption", "coal_share_energy",
                       "energy_cons_change_pct", "energy_per_capita",
                       "energy_per_gdp", "electricity_share_energy",
                       "fossil_cons_change_pct", "fossil_cons_change_twh",
                       "fossil_elec_per_capita", "fossil_fuel_consumption",
                       "fossil_share_energy", "gas_cons_change_pct",
                       "gas_cons_change_twh", "gas_elec_per_capita",
                       "gas_prod_change_pct", "gas_prod_change_twh",
                       "gas_prod_per_capita", "gas_consumption",
                       "gas_share_energy", "hydro_cons_change_pct",
                       "hydro_cons_change_twh", "hydro_elec_per_capita",
                       "fossil_energy_per_capita", "hydro_energy_per_capita",
                       "hydro_consumption", "hydro_share_energy",
                       "low_carbon_cons_change_pct", "low_carbon_cons_change_twh",
                       "low_carbon_elec_per_capita", "low_carbon_energy_per_capita",
                       "low_carbon_consumption", "low_carbon_share_energy",
                       "net_elec_imports_share_demand", "nuclear_cons_change_pct",
                       "nuclear_cons_change_twh", "nuclear_elec_per_capita",
                       "nuclear_energy_per_capita", "nuclear_consumption",
                       "nuclear_share_energy", "oil_prod_per_capita",
                       "gas_energy_per_capita", "oil_elec_per_capita",
                       "oil_prod_change_pct", "oil_prod_change_twh",
                       "oil_consumption", "oil_share_energy",
                       "other_renewable_exc_biofuel_electricity",
                       "other_renewables_cons_change_pct",
```

```
                        "other_renewables_cons_change_twh",
                        "other_renewables_elec_per_capita",
                        "other_renewables_elec_per_capita_exc_biofuel",
                        "other_renewables_energy_per_capita",
                        "other_renewables_share_elec_exc_biofuel",
                        "other_renewable_consumption",
                        "other_renewables_share_energy", "per_capita_electricity",
                        "renewables_cons_change_pct", "renewables_cons_change_twh",
                        "renewables_elec_per_capita", "renewables_energy_per_capita",
                        "renewables_consumption", "renewables_share_energy",
                        "solar_cons_change_pct", "solar_cons_change_twh",
                        "solar_elec_per_capita", "solar_consumption",
                        "solar_share_energy", "wind_cons_change_pct",
                        "wind_cons_change_twh", "wind_consumption",
                        "wind_share_energy", "solar_energy_per_capita",
                        "wind_elec_per_capita", "wind_energy_per_capita",
                        "oil_cons_change_pct", "oil_cons_change_twh",
                        "oil_energy_per_capita"))

# 3. Feature renaming
colnames(main) = c(colnames(main[,1:36]), "land_area", "hdi", "urbaniz_rate",
                            "particulate_pollution", "agri_land_rate",
                            "coal_reserves_2021", "oil_reserves_2020",
                            "uranium_reserves_2019","gas_reserves", "gdp")

# 4. Cleaning NA and 0 values
main = main %>% mutate(
  oil_reserves_2020 = coalesce(oil_reserves_2020, 0),
  uranium_reserves_2019 = coalesce(uranium_reserves_2019, 0),
  gas_reserves = coalesce(gas_reserves, 0),
  coal_reserves_2021 = coalesce(coal_reserves_2021, 0))

main = mutate(main, gdp = na_if(gdp, ".."))
main$gdp = as.numeric(main$gdp)
```

Chapter 3: Exploratory analyses

3.1 Global exploratory analyses

The first step of exploratory analysis is getting a summary of our dataset and describing each feature.

```
summary(main)
```

```
##    country               year        iso_code           population
##  Length:16338       Min.   :1900   Length:16338       Min.   :1.833e+03
##  Class :character   1st Qu.:1944   Class :character   1st Qu.:1.286e+06
##  Mode  :character   Median :1983   Mode  :character   Median :5.683e+06
##                     Mean   :1973                      Mean   :2.688e+07
##                     3rd Qu.:2003                      3rd Qu.:1.717e+07
##                     Max.   :2022                      Max.   :1.426e+09
##                                                       NA's   :65
##  carbon_intensity_elec coal_electricity  coal_production    coal_share_elec
##  Min.   :   0.0        Min.   :   0.00   Min.   :    0.00   Min.   :  0.00
##  1st Qu.: 266.7        1st Qu.:   0.00   1st Qu.:    0.00   1st Qu.:  0.00
##  Median : 488.6        Median :   0.00   Median :    0.00   Median :  0.00
##  Mean   : 439.3        Mean   :  46.78   Mean   :  163.40   Mean   : 13.83
##  3rd Qu.: 629.6        3rd Qu.:   7.66   3rd Qu.:   12.35   3rd Qu.: 19.52
##  Max.   :1000.0        Max.   :5339.14   Max.   :23651.39   Max.   :100.00
##  NA's   :11776         NA's   :11033     NA's   :3447       NA's   :11056
##  electricity_demand electricity_generation energy_cons_change_twh
##  Min.   :   0.00    Min.   :   0.000       Min.   :-1978.438
##  1st Qu.:   0.93    1st Qu.:   1.155       1st Qu.:   -0.083
##  Median :   8.21    Median :  12.140       Median :    0.378
##  Mean   :  99.07    Mean   : 109.935       Mean   :   12.128
##  3rd Qu.:  46.96    3rd Qu.:  57.050       3rd Qu.:    6.700
##  Max.   :8466.32    Max.   :8484.020       Max.   : 2796.320
##  NA's   :11297      NA's   :10446          NA's   :7078
##  fossil_electricity fossil_share_elec gas_electricity   gas_production
##  Min.   :   0.00    Min.   :  0.00    Min.   :   0.00   Min.   :   0.000
##  1st Qu.:   0.30    1st Qu.: 38.65    1st Qu.:   0.00   1st Qu.:   0.000
##  Median :   3.65    Median : 72.70    Median :   0.38   Median :   0.000
##  Mean   :  76.33    Mean   : 64.67    Mean   :  23.93   Mean   :  96.610
##  3rd Qu.:  34.47    3rd Qu.: 96.67    3rd Qu.:  13.10   3rd Qu.:   7.249
##  Max.   :5623.99    Max.   :100.00    Max.   :1624.17   Max.   :9342.032
##  NA's   :10927      NA's   :11056     NA's   :11033     NA's   :3280
##  gas_share_elec    greenhouse_gas_emissions hydro_electricity
##  Min.   :  0.000   Min.   :   0.00          Min.   :   0.000
##  1st Qu.:  0.000   1st Qu.:   0.21          1st Qu.:   0.010
##  Median :  1.124   Median :   1.68          Median :   1.490
##  Mean   : 18.181   Mean   :  46.68          Mean   :  18.392
##  3rd Qu.: 27.882   3rd Qu.:  15.49          3rd Qu.:   9.482
##  Max.   :100.000   Max.   :4618.32          Max.   :1321.710
##  NA's   :11056     NA's   :11647            NA's   :9173
##  hydro_share_elec  low_carbon_electricity low_carbon_share_elec
##  Min.   :  0.000   Min.   :   0.000       Min.   :  0.000
##  1st Qu.:  0.028   1st Qu.:   0.047       1st Qu.:  2.196
##  Median : 10.664   Median :   2.340       Median : 26.051
##  Mean   : 25.646   Mean   :  35.803       Mean   : 35.046
```

```
##  3rd Qu.: 45.233   3rd Qu.:  16.330       3rd Qu.: 61.852
##  Max.   :100.000   Max.   :2860.030       Max.   :100.000
##  NA's   :10576     NA's   :9132           NA's   :10575
##  net_elec_imports  nuclear_electricity nuclear_share_elec oil_electricity
##  Min.   :-77.030   Min.   :  0.00      Min.   : 0.000     Min.   :  0.000
##  1st Qu.:  0.000   1st Qu.:  0.00      1st Qu.: 0.000     1st Qu.:  0.080
##  Median :  0.000   Median :  0.00      Median : 0.000     Median :  0.820
##  Mean   :  0.051   Mean   : 13.42      Mean   : 5.104     Mean   :  7.147
##  3rd Qu.:  0.350   3rd Qu.:  0.00      3rd Qu.: 0.000     3rd Qu.:  4.640
##  Max.   : 66.670   Max.   :809.41      Max.   :88.138     Max.   :287.538
##  NA's   :11297     NA's   :9137        NA's   :10580      NA's   :11033
##  oil_production    oil_share_elec    other_renewable_electricity
##  Min.   :   0.00   Min.   :  0.000   Min.   :  0.000
##  1st Qu.:   0.00   1st Qu.:  1.864   1st Qu.:  0.000
##  Median :   0.00   Median : 12.078   Median :  0.000
##  Mean   : 170.85   Mean   : 32.657   Mean   :  1.734
##  3rd Qu.:  25.12   3rd Qu.: 60.370   3rd Qu.:  0.340
##  Max.   :8721.28   Max.   :100.000   Max.   :169.932
##  NA's   :2817      NA's   :11056     NA's   :9288
##  other_renewables_share_elec primary_energy_consumption renewables_electricity
##  Min.   : 0.000              Min.   :    0.00          Min.   :   0.000
##  1st Qu.: 0.000              1st Qu.:    5.48          1st Qu.:   0.050
##  Median : 0.000              Median :   47.70          Median :   1.882
##  Mean   : 2.562              Mean   :  590.00          Mean   :  22.546
##  3rd Qu.: 1.613              3rd Qu.:  294.78          3rd Qu.:  11.663
##  Max.   :71.429              Max.   :43790.89          Max.   :2452.530
##  NA's   :10625               NA's   :6862              NA's   :9182
##  renewables_share_elec solar_electricity solar_share_elec wind_electricity
##  Min.   :  0.000       Min.   :  0.000   Min.   : 0.000   Min.   :  0.000
##  1st Qu.:  1.475       1st Qu.:  0.000   1st Qu.: 0.000   1st Qu.:  0.000
##  Median : 16.768       Median :  0.000   Median : 0.000   Median :  0.000
##  Mean   : 30.209       Mean   :  0.698   Mean   : 0.585   Mean   :  1.884
##  3rd Qu.: 54.417       3rd Qu.:  0.000   3rd Qu.: 0.026   3rd Qu.:  0.010
##  Max.   :100.000       Max.   :327.000   Max.   :40.000   Max.   :655.600
##  NA's   :10625         NA's   :9212      NA's   :10625    NA's   :9222
##  wind_share_elec    land_area             hdi          urbaniz_rate
##  Min.   : 0.000   Min.   :       10   Min.   :0.216   Min.   :  2.077
##  1st Qu.: 0.000   1st Qu.:    23180   1st Qu.:0.542   1st Qu.: 33.295
##  Median : 0.000   Median :   143000   Median :0.692   Median : 53.485
##  Mean   : 1.313   Mean   :   703350   Mean   :0.668   Mean   : 53.562
##  3rd Qu.: 0.089   3rd Qu.:   566730   3rd Qu.:0.796   3rd Qu.: 73.799
##  Max.   :56.840   Max.   :16389950   Max.   :0.962   Max.   :100.000
##  NA's   :10625    NA's   :6217       NA's   :10866   NA's   :6279
##  particulate_pollution agri_land_rate   coal_reserves_2021  oil_reserves_2020
##  Min.   :  2.48        Min.   : 0.263   Min.   :0.000e+00   Min.   :0.000e+00
##  1st Qu.: 21.91        1st Qu.:18.678   1st Qu.:0.000e+00   1st Qu.:0.000e+00
##  Median : 35.03        Median :37.621   Median :0.000e+00   Median :0.000e+00
##  Mean   : 45.28        Mean   :37.257   Mean   :6.845e+09   Mean   :4.251e+08
##  3rd Qu.: 59.42        3rd Qu.:55.376   3rd Qu.:0.000e+00   3rd Qu.:0.000e+00
##  Max.   :205.58        Max.   :90.556   Max.   :2.489e+11   Max.   :4.144e+10
##  NA's   :10471         NA's   :7104
##  uranium_reserves_2019  gas_reserves            gdp
##  Min.   :      0       Min.   :0.000e+00   Min.   :2.156e+07
##  1st Qu.:      0       1st Qu.:0.000e+00   1st Qu.:5.099e+09
```

```
##  Median :      0       Median :0.000e+00   Median :2.179e+10
##  Mean   :  46446       Mean   :3.221e+11   Mean   :2.748e+11
##  3rd Qu.:   6100       3rd Qu.:0.000e+00   3rd Qu.:1.296e+11
##  Max.   :2049400       Max.   :3.789e+13   Max.   :2.053e+13
##                                            NA's   :7579
```

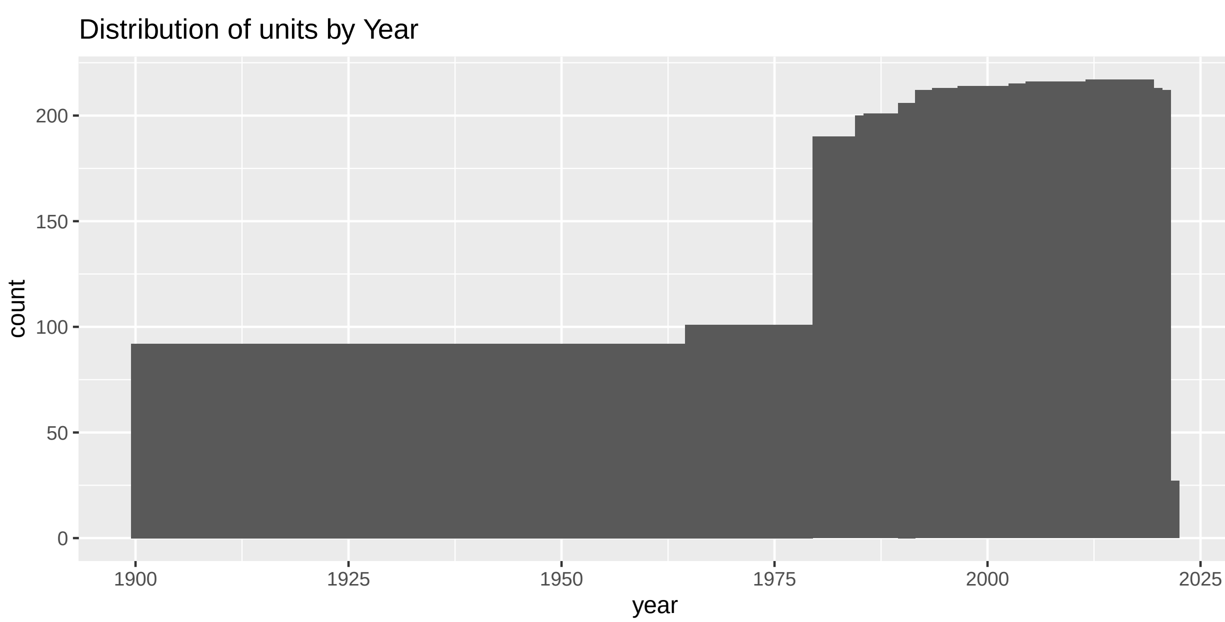
Each unit represent a country in a given year; the dataset has two character variables, country and ISO code, and 45 numerical variables.

10 of them are have been presented in the data obtaining step; the other 35 features, belonging to the initial energy dataset, are:

- **Year** and **Population**;
- Pollution measurements: **Carbon intensity of electricity** (which measures how many grams of CO2 are released to produce a kWh of electricity) and **Greenhouse gas emissions**;
- Overall energy and electricity measurements: **Electricity demand** (which is the amount of electricity consumed), **Electricity generation** (the amount of electricity produced), **Energy consumption change in tWh** (change in energy consumed compared to the previous year), **Net electricity imports** (electricity imported minus electricity exported) and **Primary energy consumption** (Energy consumed by the country);
- Variables related to electricity consumption and production for each source: **renewables**, that are divided in hydro, solar, wind, other renewables; **fossil**, divided in coal, oil and gas; and finally **low-carbon**, which is the aggregation of renewables with nuclear. For each source the dataset presents a variable for **electricity production** and **share of electricity production**, and for each single fossil source the **production** is also present.

The following is the distribution of units by Year:

```
ggplot(main, aes(x = year)) + geom_histogram(bins = 123) +
  ggtitle(paste("Distribution of units by Year"))
```

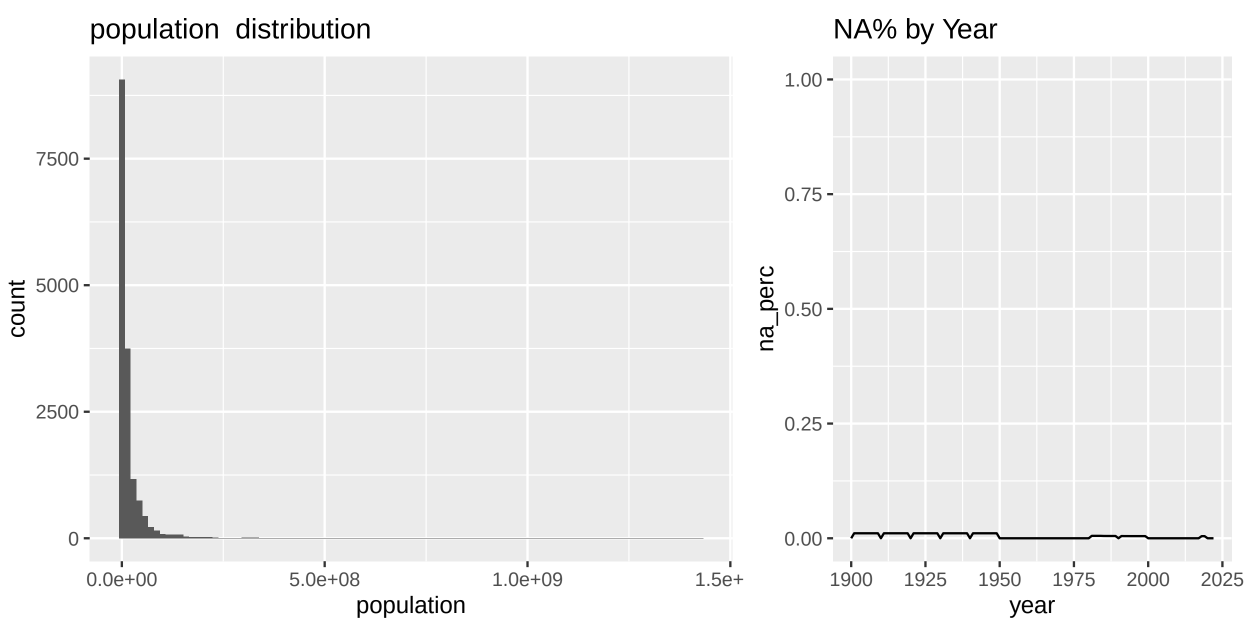


3.2 Variable transformation

Then, we explore more in depth each variable by plotting its distribution and by viewing the share of NA values, since for some years the dataset is very sparse.

However, as is already noticeable from the summaries, the data is very skewed and has many outliers. This is further shown by plotting for Population.

```
i=4
i1 = colnames(main)[i]
p11 = ggplot(main, aes_string(x = i1)) + geom_histogram(bins = 100) +
  ggtitle(paste(i1, " distribution"))
nacount = main %>%
  group_by(year) %>%
  summarize(na_perc = sum(is.na(!!sym(i1)))/n())
p12 = ggplot(nacount, aes(x = year, y = na_perc)) + geom_line() + ylim(0, 1) +
  ggtitle("NA% by Year")
grid.arrange(p11, p12, widths = c(0.6, 0.4), ncol = 2)
```



Therefore, we decide to **transform data** for the exploratory analysis: first by dividing all variables that are not shares, except HDI, by the population (as millions of inhabitants); then by applying a logarithm transformation to the following features: population, land area, GDP and the four reserves variables.

Also, we create a subset of our dataset containing data from a single year, 2016.

```
cols_pc = c(6,7,9,10,11,12,14,15,17,18,20,22,23,25,26,28,29,30,31,33,35,37,42,43,44,45,46)
cols_log = c(4, 37, 42, 43, 44, 45, 46)

mainlog = main %>% mutate(across(all_of(cols_pc), .fns = ~.*1000000/population))
mainlog = mainlog %>% mutate(across(all_of(c(cols_log)), .fns = ~ log(.+1)))

#single year visualization, 2016
mainlog2016 = mainlog[mainlog$year==2016,]
```

3.3 Plotting function and division in groups

As we have many variables, we write a function to plot for a given variable:

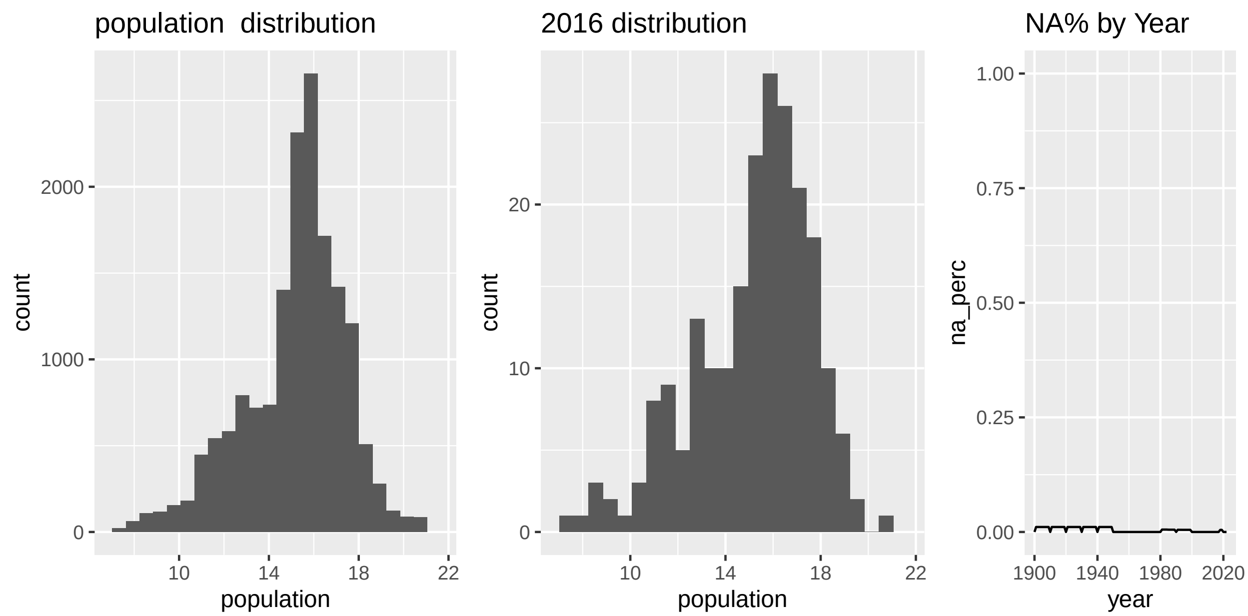
- its distribution;
- its distribution only for the year 2016;
- and its NA percentage by Year.

The function also prints the three countries with the highest measurement for the variable in 2016.

```
do_plots = function(i){
  i1 = colnames(mainlog)[i]
  x_min = min(mainlog[i1], na.rm=TRUE)
  x_max = max(mainlog[i1], na.rm=TRUE)
  x_diff = x_max-x_min

  # Plot of the whole distribution
  p11 = ggplot(mainlog, aes_string(x = i1)) + geom_histogram(bins = 25) +
    ggtitle(paste(i1, " distribution")) + xlim(x_min-x_diff/25, x_max+x_diff/25)

  # Plot of 2016 distribution
  p12 = ggplot(mainlog2016, aes_string(x = i1)) + geom_histogram(bins = 25) +
    ggtitle("2016 distribution") + xlim(x_min-x_diff/25, x_max+x_diff/25)

  # Plot of % of NA by year
  nacount = mainlog %>%
    group_by(year) %>%
    summarize(na_perc = sum(is.na(!!sym(i1)))/n())
  p13 = ggplot(nacount, aes(x = year, y = na_perc)) + geom_line() + ylim(0, 1) +
    ggtitle("NA% by Year") + scale_x_continuous(breaks = seq(1900, 2022, by = 40))
  grid.arrange(p11, p12, p13, widths = c(3,3,2), ncol = 3)

  # Print of the three countries with the highest measurement in 2016
  i1_ord = mainlog2016[order(mainlog2016[i1], decreasing=TRUE),1]
  print(paste("Top three countries in 2016 for",i1,":"))
  print(paste(i1_ord[1], ",", i1_ord[2], ",", i1_ord[3]))
}
```

We can plot again for Population, and we can now see population has a log-normal distribution. For 2016 it is similarly distributed, with an overall shift to the right.

```
do_plots(4)
```




```
## [1] "Top three countries in 2016 for population :"
## [1] "China , India , United States"
```

To make the exploration easier and more intuitive, we divide variables into five groups, each containing:

1. variables about the reserves;
2. variables that regard low-carbon sources;
3. variables regarding fossil sources;
4. other variables that belong to the *Energy dataset*, which are variables in that dataset not about a specific source;
5. other variables that do not belong to the *Energy dataset*.

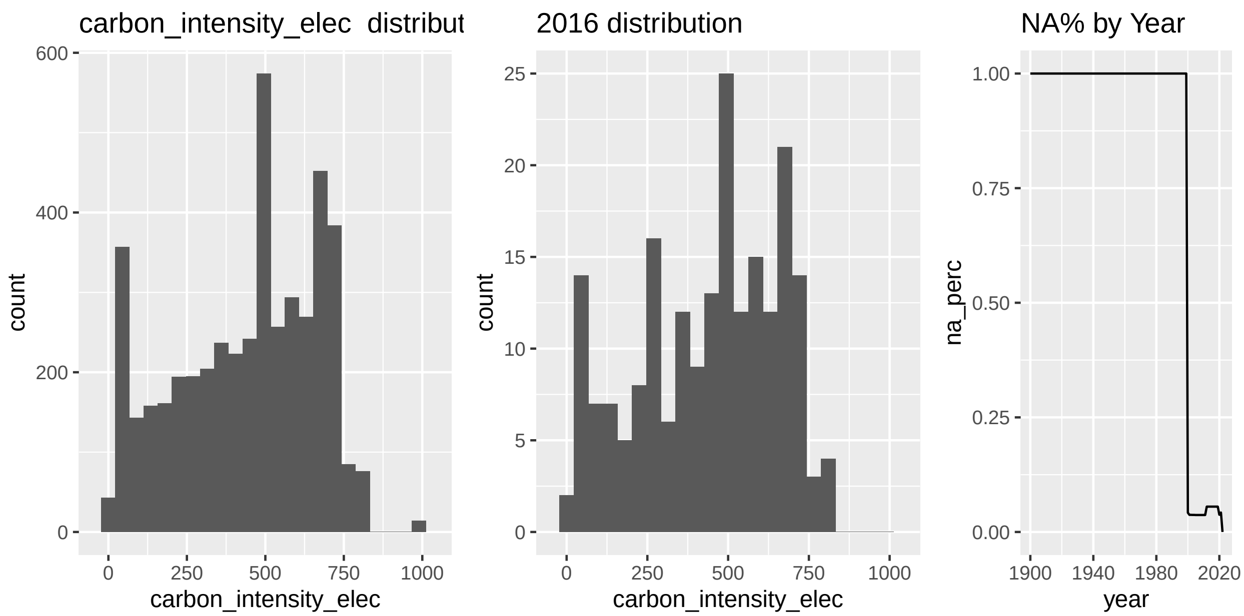
```
other_measures = c(5,17,9,10,11,22,30)
reserves = c(42,43,44,45)
ext_measures = c(37,38,39,40,41,46)
lowcarb = c(18,19,33,34,35,36,28,29,31,32,23,24,20,21)
highcarb = c(6,7,8,14,15,16,25,26,27,12,13)
```

Now we plot for each group using the function we created, and we also plot the correlation between variables inside each group.

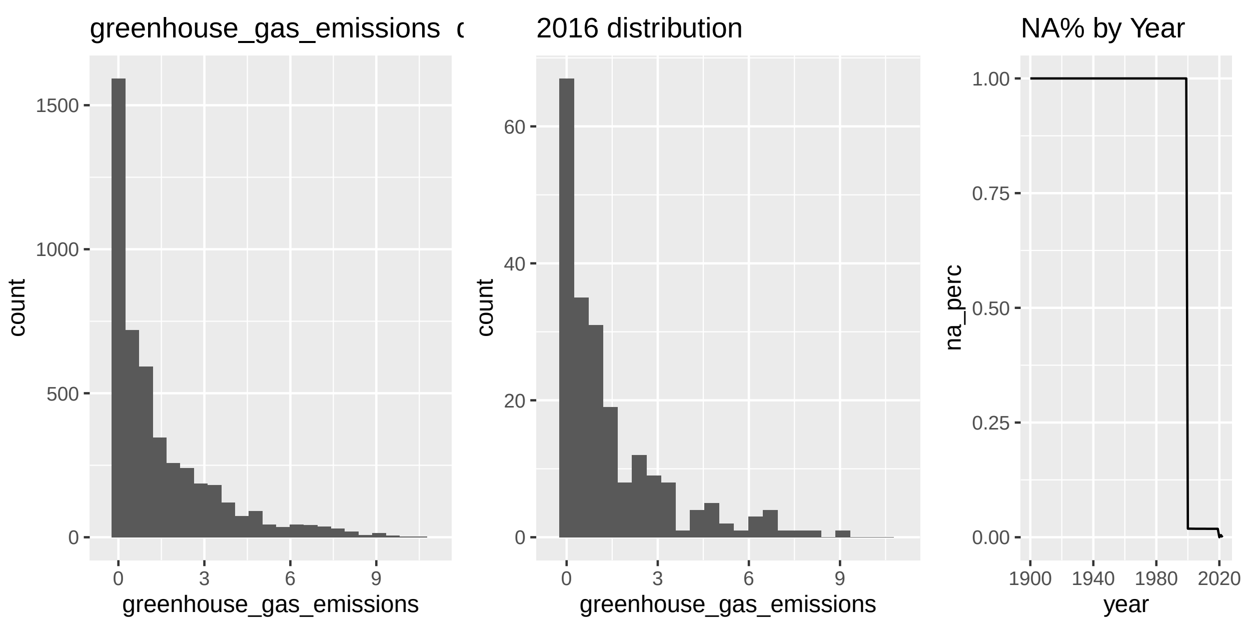
3.4 Analysis on groups of variables

3.4.1 Analysis on Energy dataset non-source specific

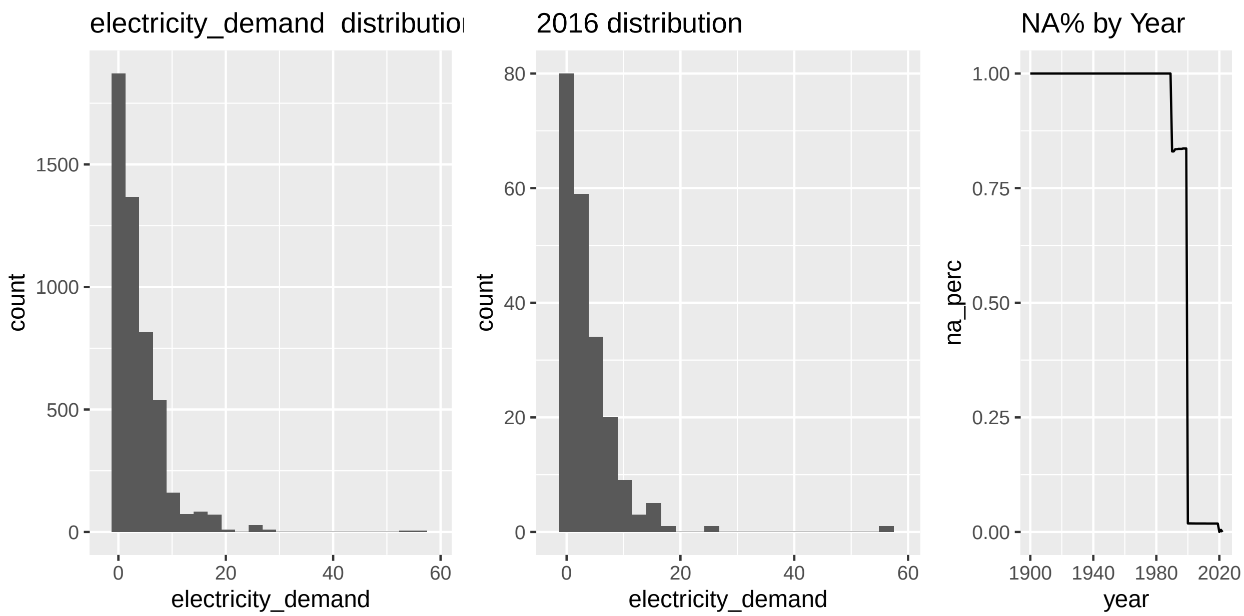
```
for (i in other_measures){
  do_plots(i)
}
```



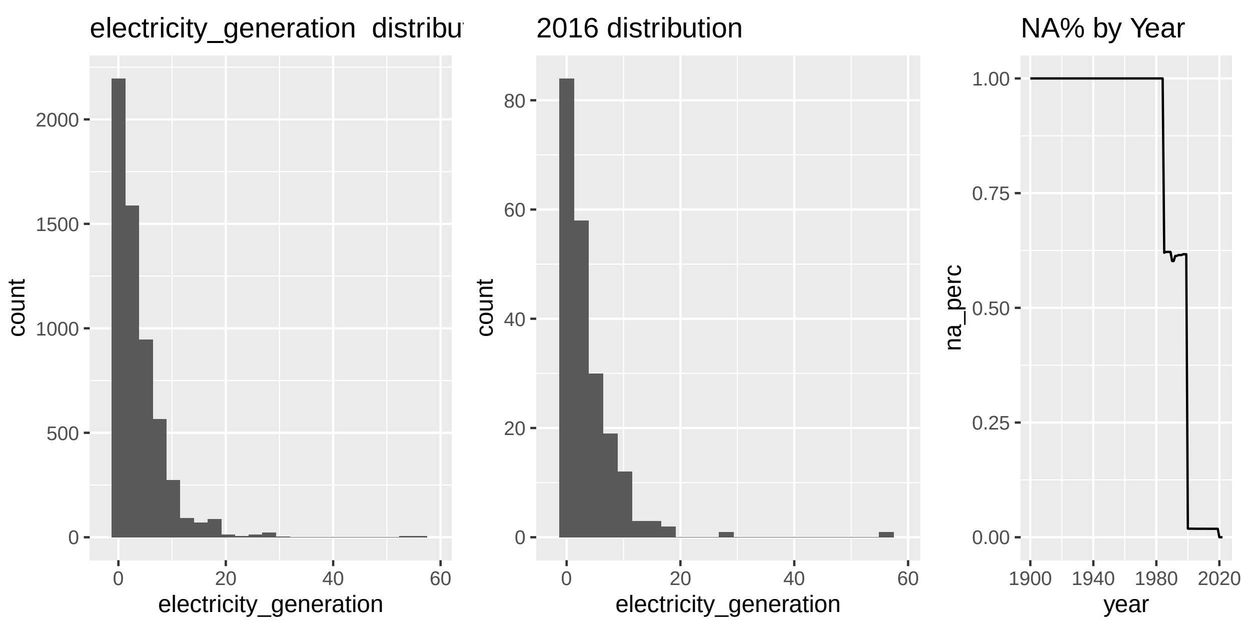

```
## [1] "Top three countries in 2016 for carbon_intensity_elec :"
## [1] "Botswana , Comoros , Saint Pierre and Miquelon"
```



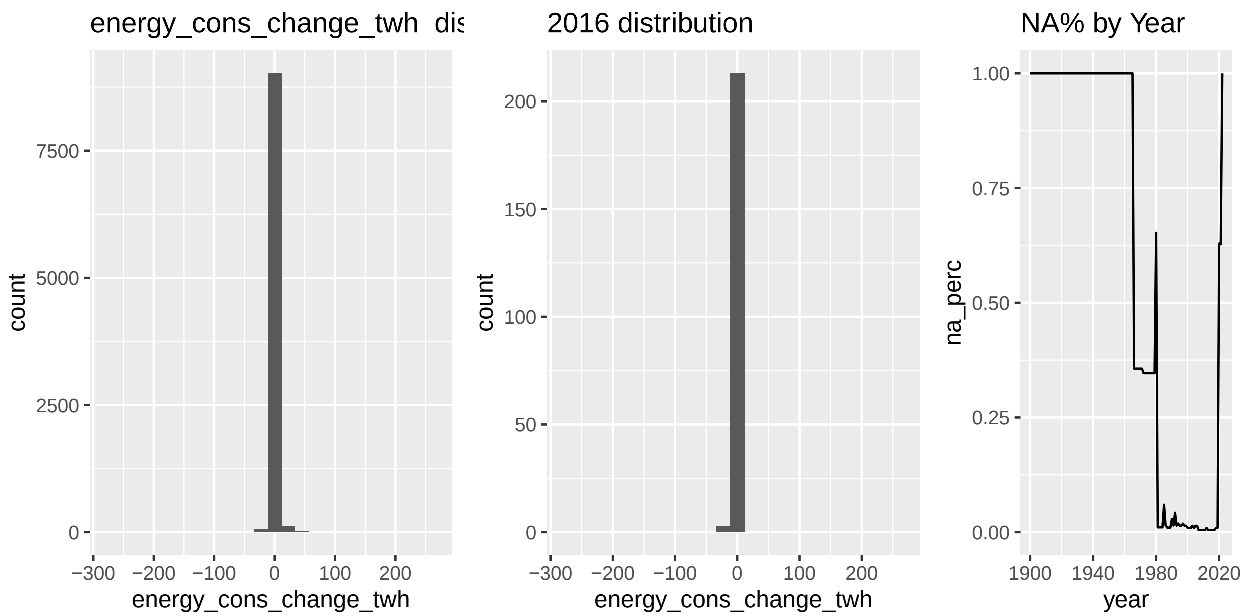

```
## [1] "Top three countries in 2016 for greenhouse_gas_emissions :"
## [1] "Bahrain , Kuwait , Qatar"
```



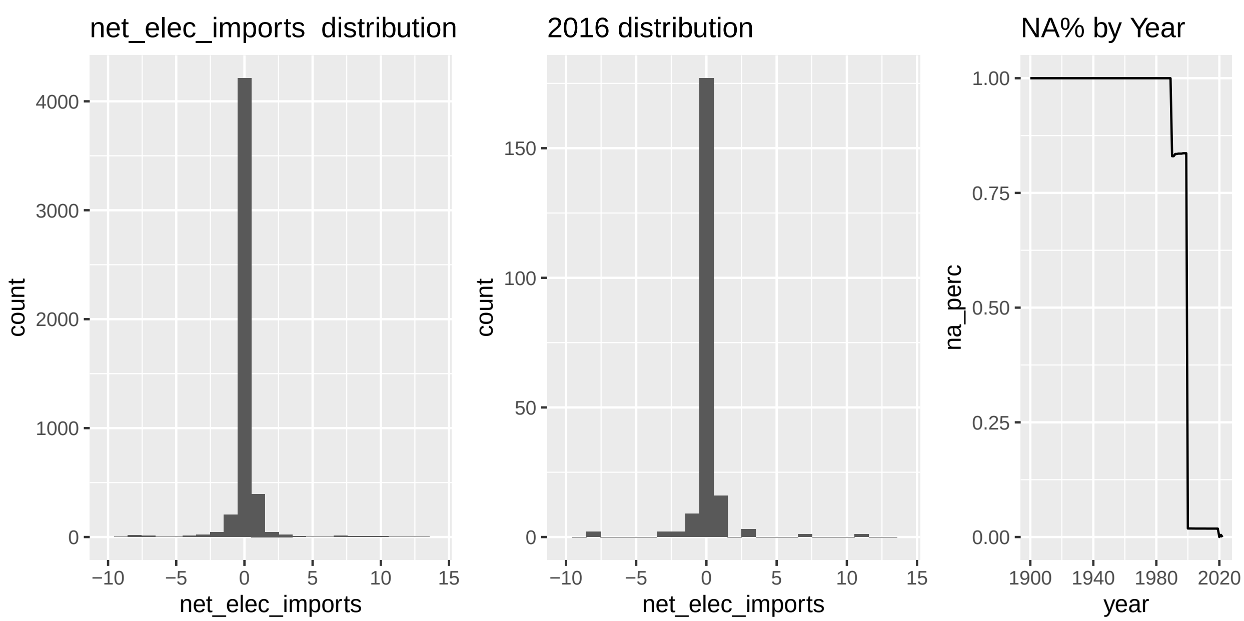

```
## [1] "Top three countries in 2016 for electricity_demand :"
## [1] "Iceland , Norway , Bahrain"
```



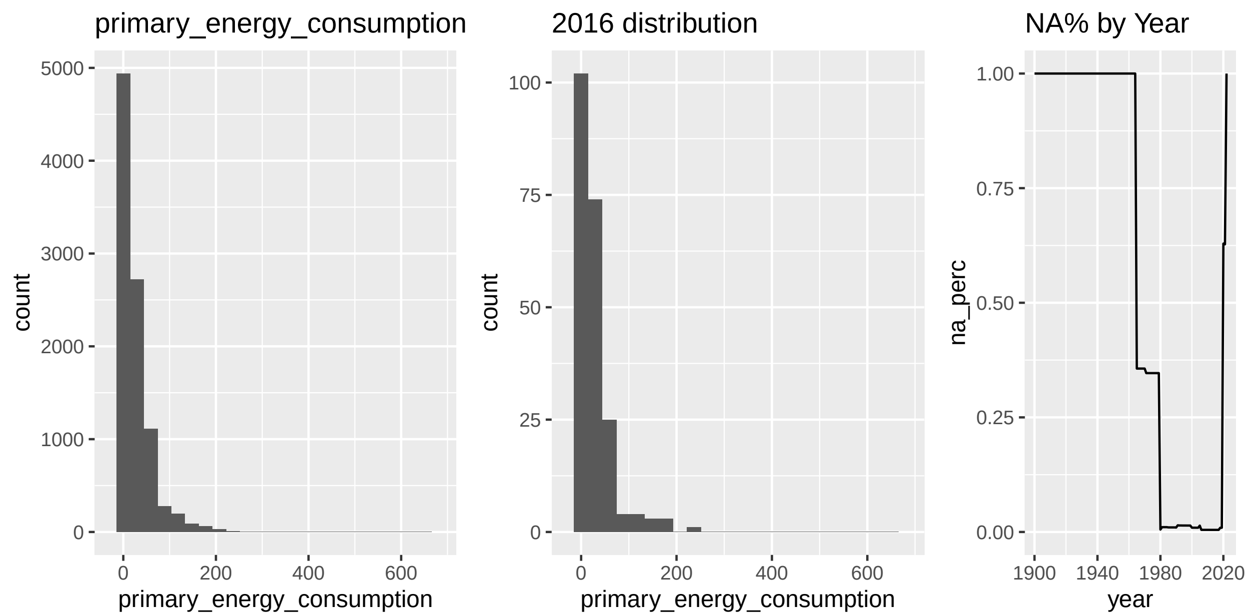

```
## [1] "Top three countries in 2016 for electricity_generation :"
## [1] "Iceland , Norway , Bahrain"
```




```
## [1] "Top three countries in 2016 for energy_cons_change_twh :"
## [1] "Bermuda , Laos , Malta"
```

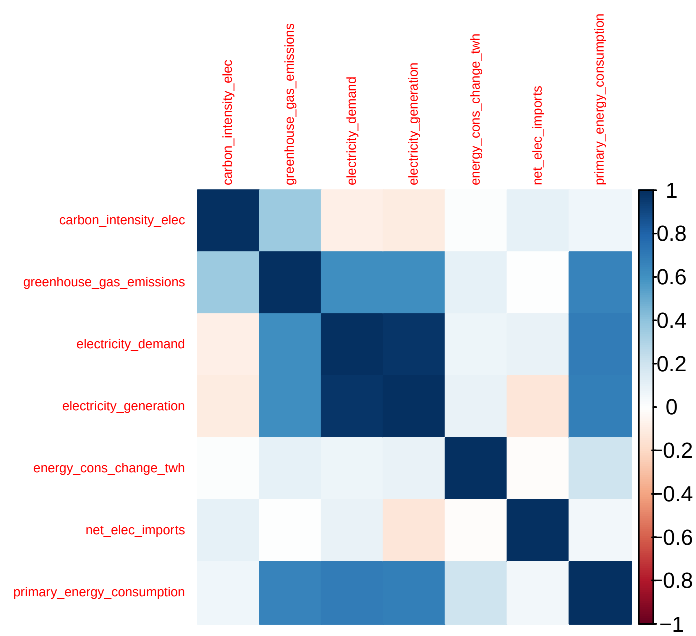



```
## [1] "Top three countries in 2016 for net_elec_imports :"
## [1] "Luxembourg , Macao , Finland"
```

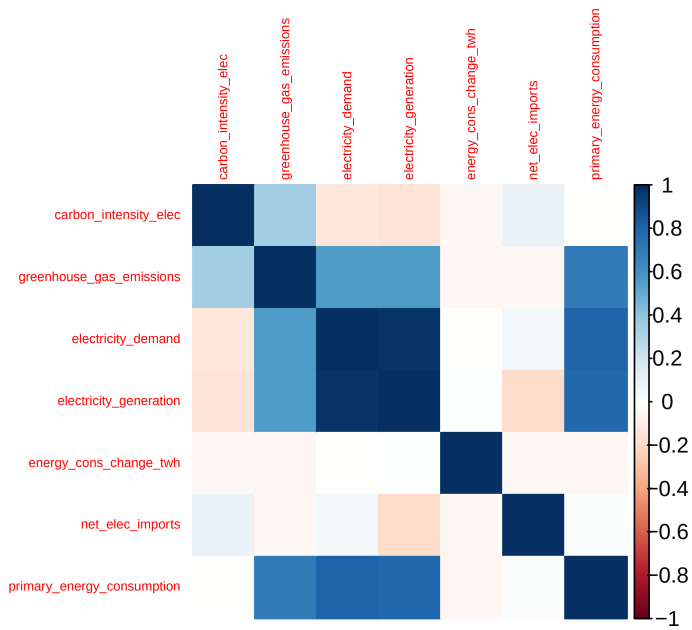



```
## [1] "Top three countries in 2016 for primary_energy_consumption :"
## [1] "Qatar , Iceland , Netherlands Antilles"
```

```
corrplot(cor(mainlog[,other_measures], use="pairwise.complete.obs"),
         method="color", tl.cex = .5)
```




```
corrplot(cor(mainlog2016[,other_measures], use="pairwise.complete.obs"),
         method="color", tl.cex = .5)
```



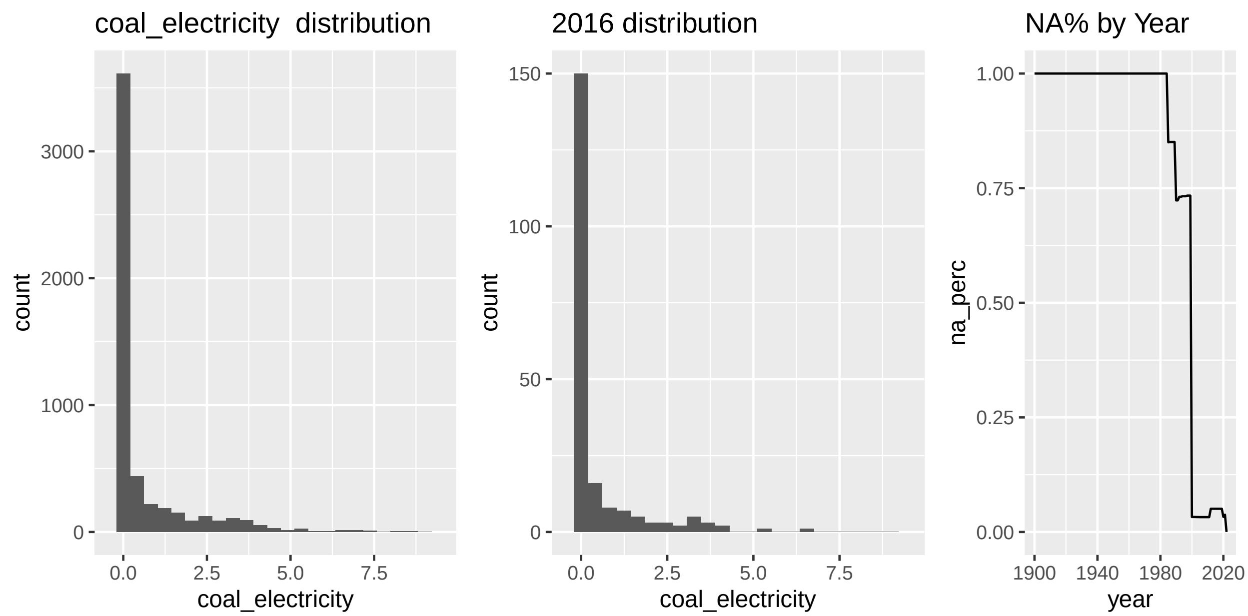
From the NA% plot we notice that some variables are only recorded since 2000, which is important to highlight, especially for data modeling.

There is a noticeable difference between **carbon intensity of electricity** and **greenhouse gas emissions** (even though the two have a slight positive correlation): the first one has a distribution similar to a Gaussian, with left skewing, while the second looks more like a log-Gaussian. However, the most important difference can be noticed in the countries with the highest measurements: carbon intensity in fact measures only the CO2 pollution from electricity, with some small countries being the highest polluters (for each kWh), with these countries probably fully relying on coal for electricity production; greenhouse gas emissions instead consider the emissions made during energy generation, so it considers also primary energy (therefore the countries with the highest scores are oil producers from Middle East & North Africa). Also **Primary energy consumption** is correlated to greenhouse gas emissions, but not to carbon intensity of electricity.

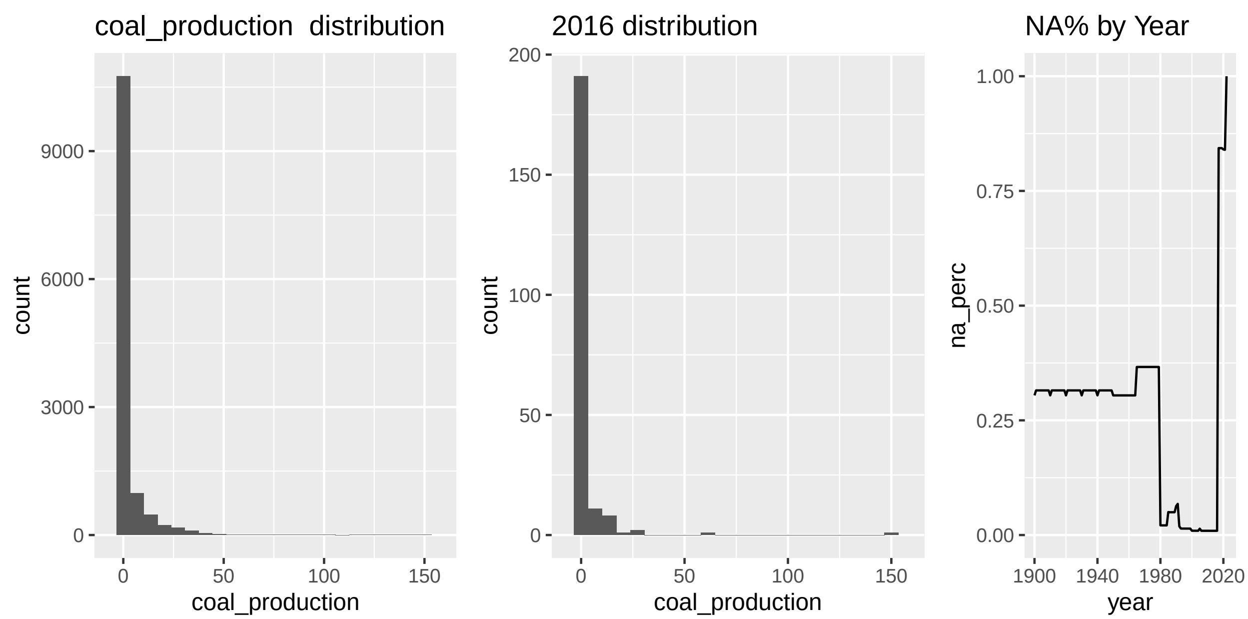
The countries with the highest **electricity demands and generation** (two measurements that are almost collinear, as expected) are rich countries that are either very cold or very hot.

3.4.2 Analysis on Energy dataset for fossil sources

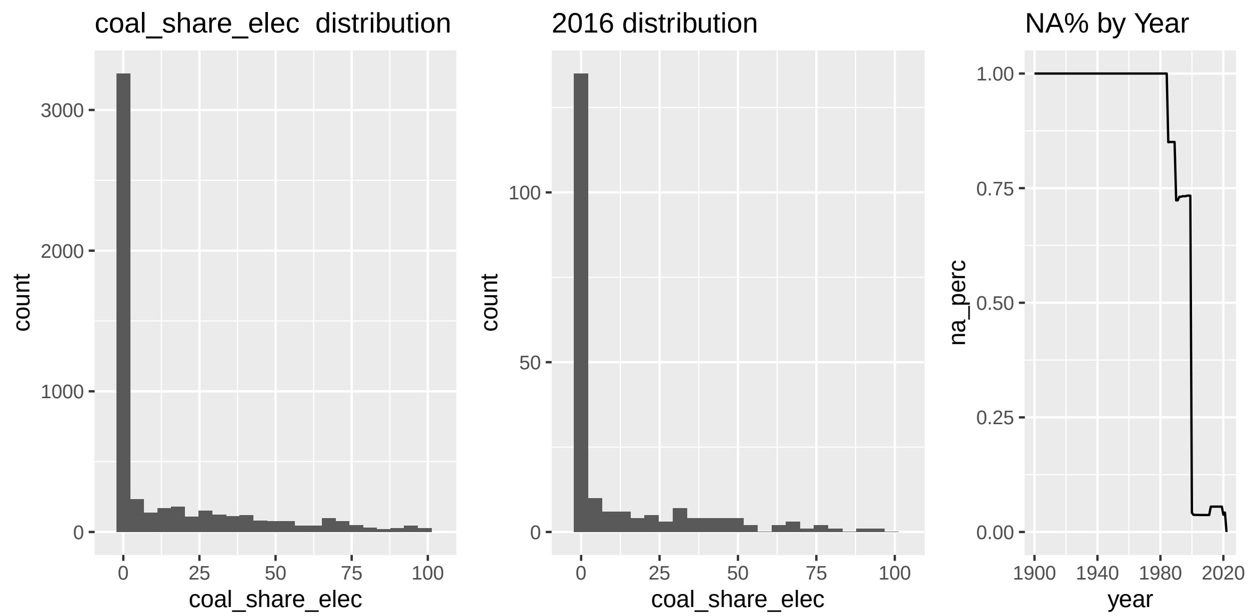
```
for (i in highcarb){
  do_plots(i)
}
```



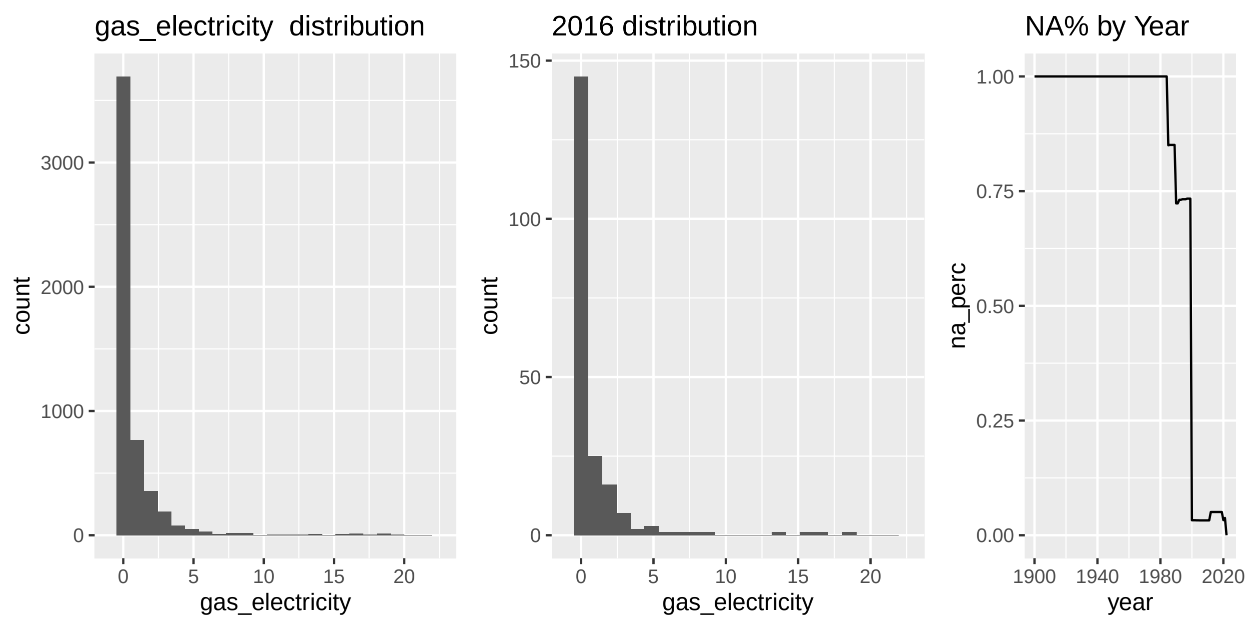

```
## [1] "Top three countries in 2016 for coal_electricity :"
## [1] "Australia , Taiwan , South Korea"
```



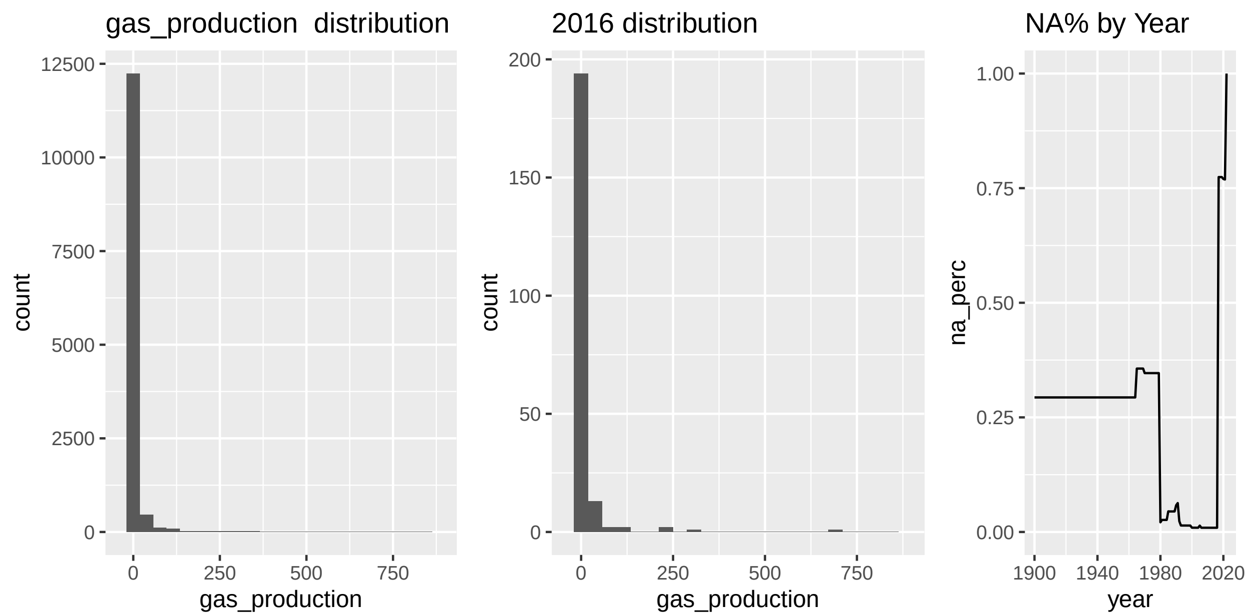

```
## [1] "Top three countries in 2016 for coal_production :"
## [1] "Australia , Mongolia , South Africa"
```



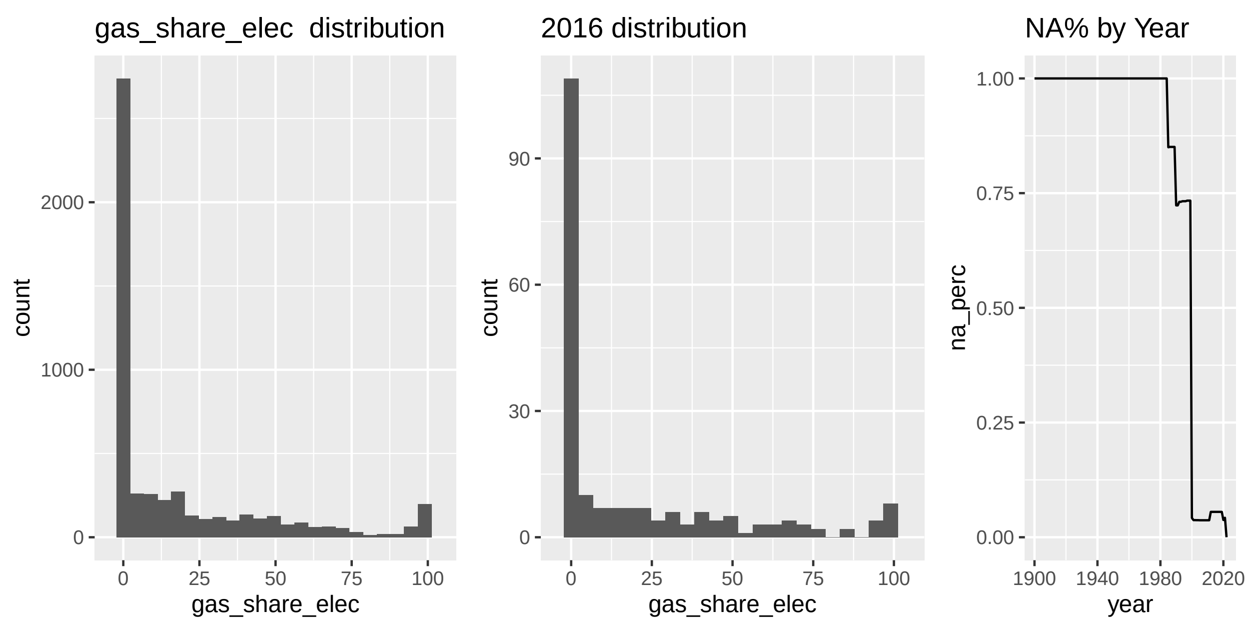

```
## [1] "Top three countries in 2016 for coal_share_elec :"
## [1] "Mongolia , South Africa , Botswana"
```



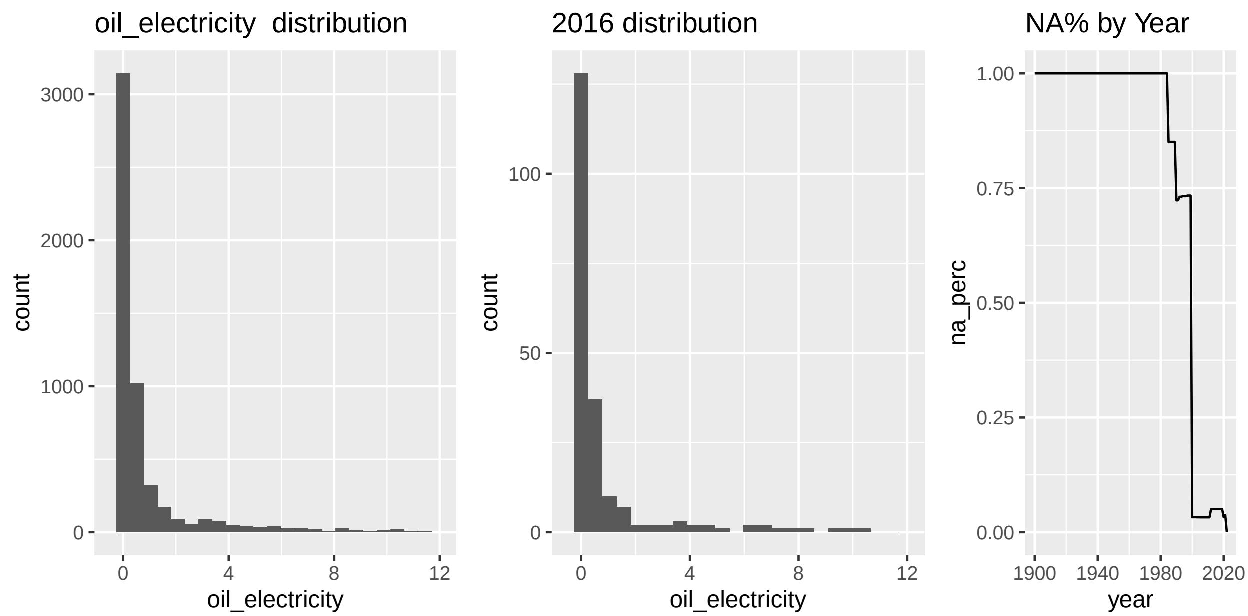

```
## [1] "Top three countries in 2016 for gas_electricity :"
## [1] "Bahrain , Kuwait , Qatar"
```



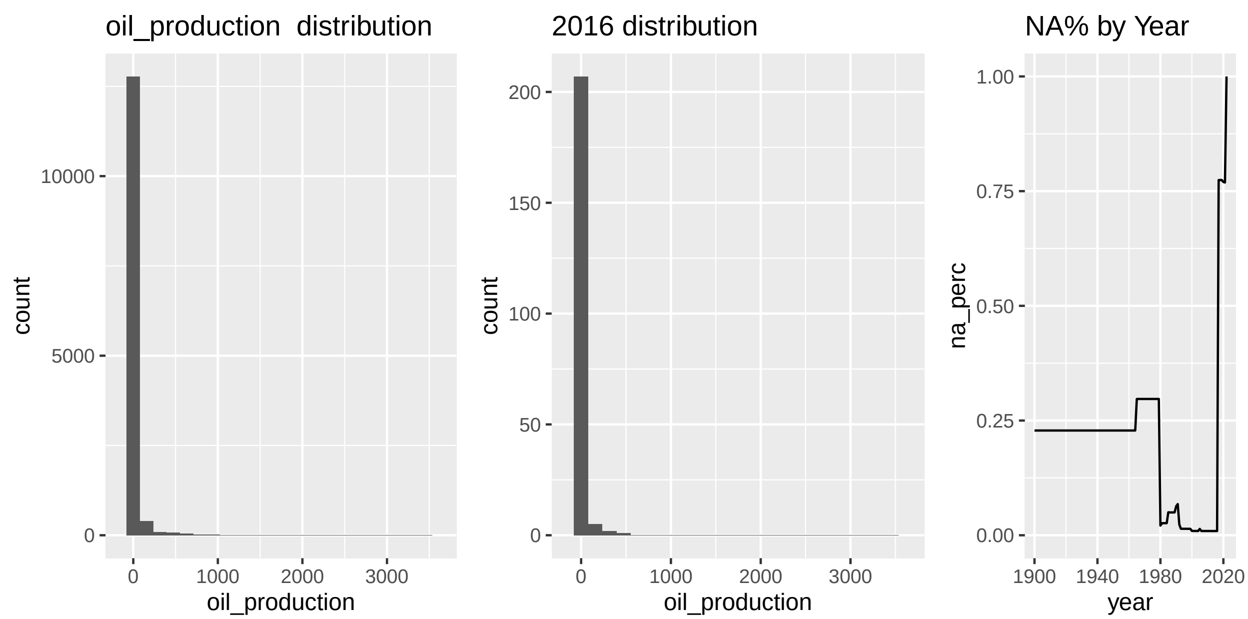

```
## [1] "Top three countries in 2016 for gas_production :"
## [1] "Qatar , Brunei , Norway"
```



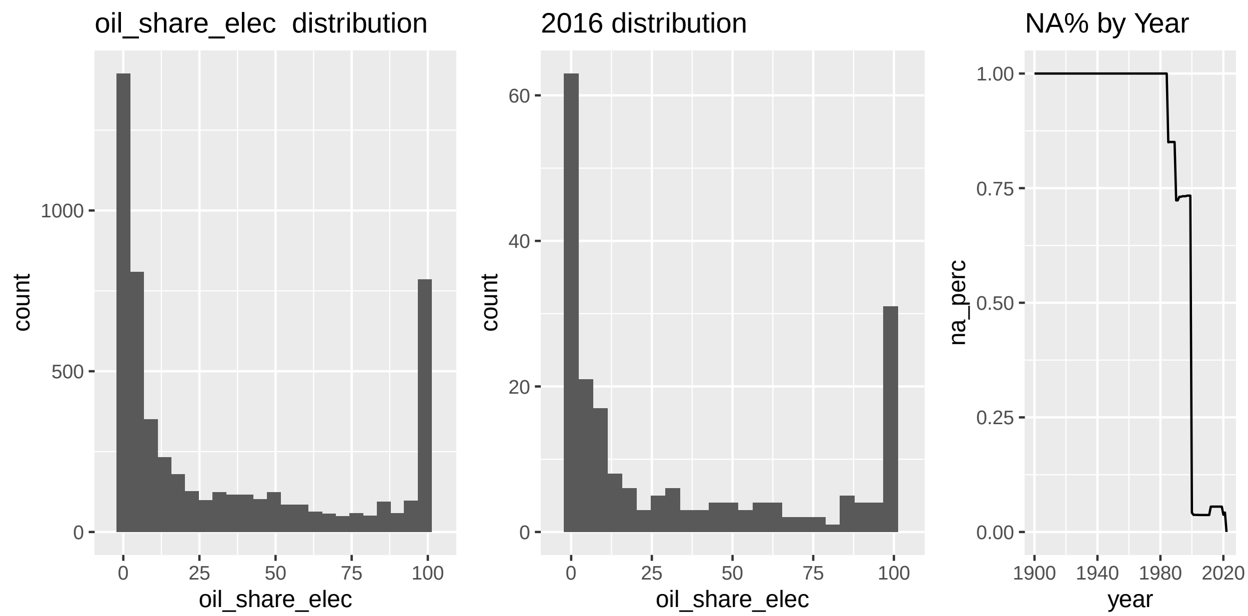

```
## [1] "Top three countries in 2016 for gas_share_elec :"
## [1] "Macao , Oman , Kuwait"
```



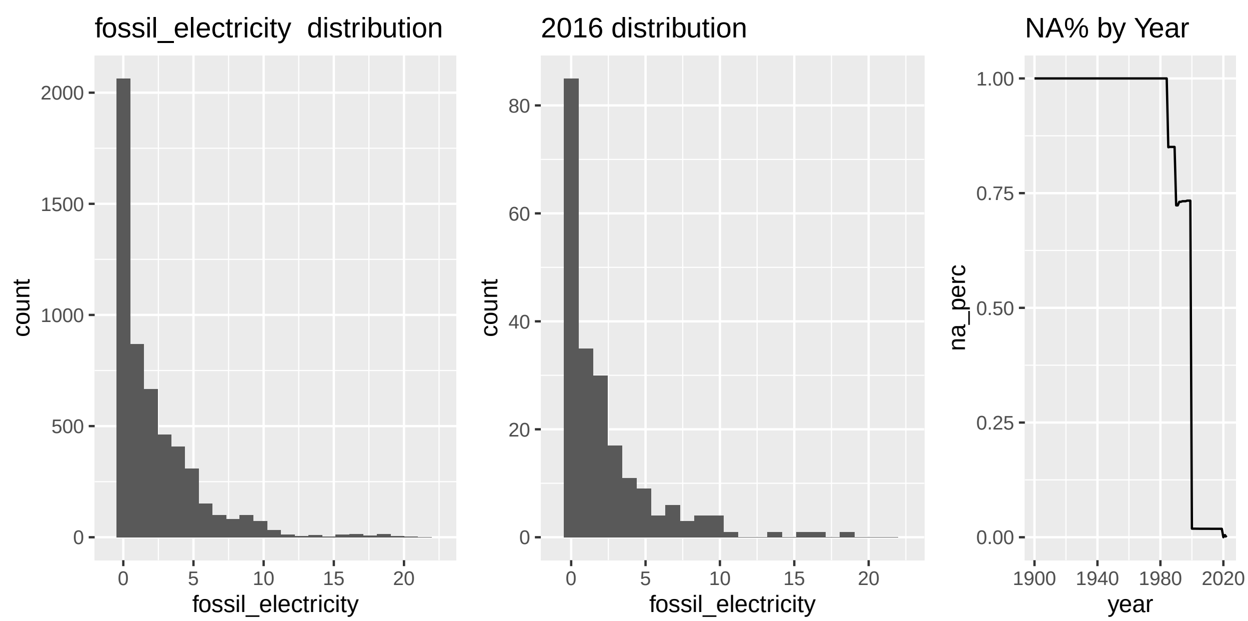

```
## [1] "Top three countries in 2016 for oil_electricity :"
## [1] "Cayman Islands , Guam , New Caledonia"
```



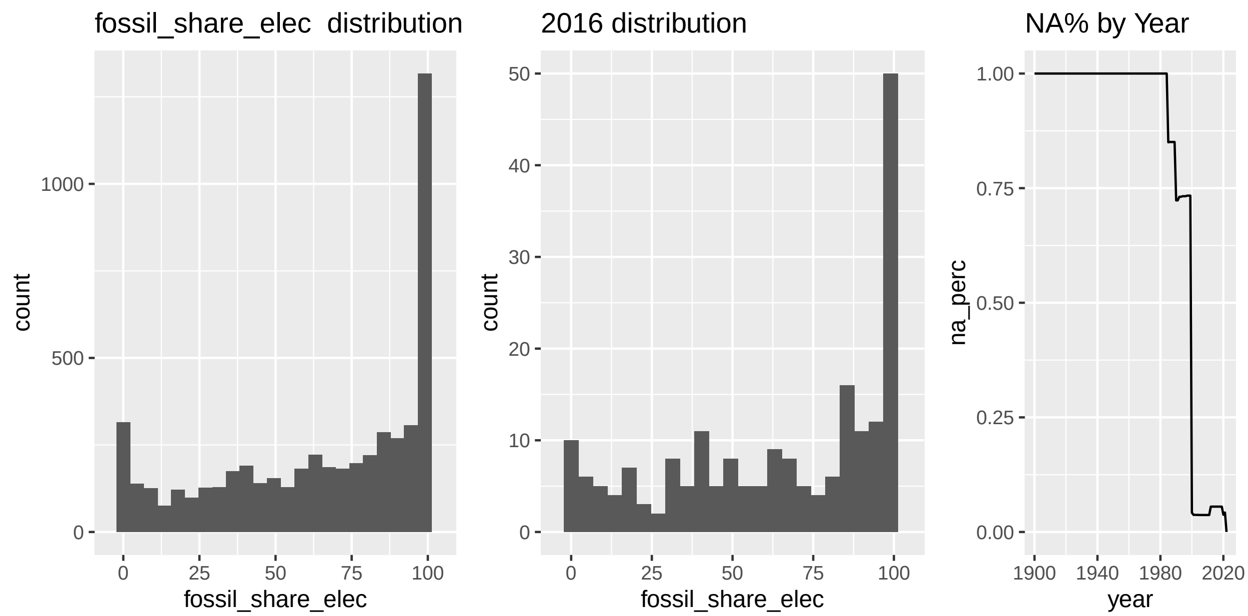

```
## [1] "Top three countries in 2016 for oil_production :"
## [1] "Kuwait , Qatar , United Arab Emirates"
```




```
## [1] "Top three countries in 2016 for oil_share_elec :"
## [1] "American Samoa , Antigua and Barbuda , Bahamas"
```

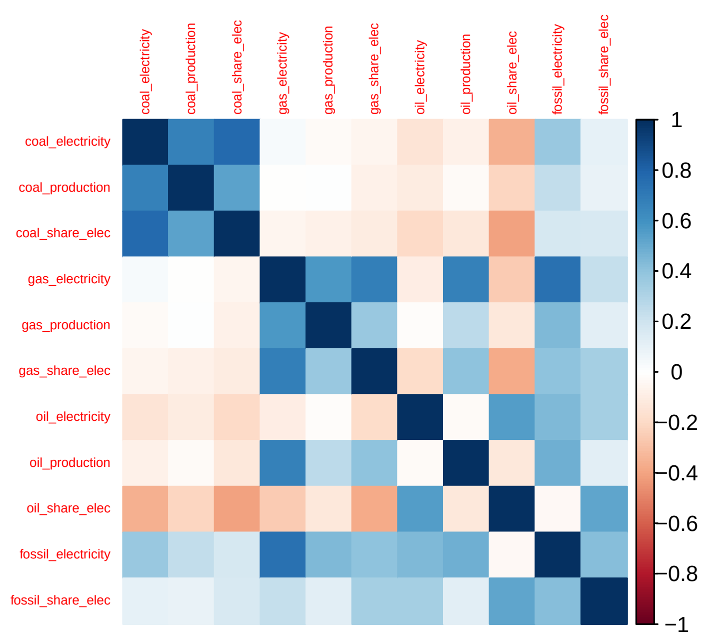



```
## [1] "Top three countries in 2016 for fossil_electricity :"
## [1] "Bahrain , Kuwait , Qatar"
```

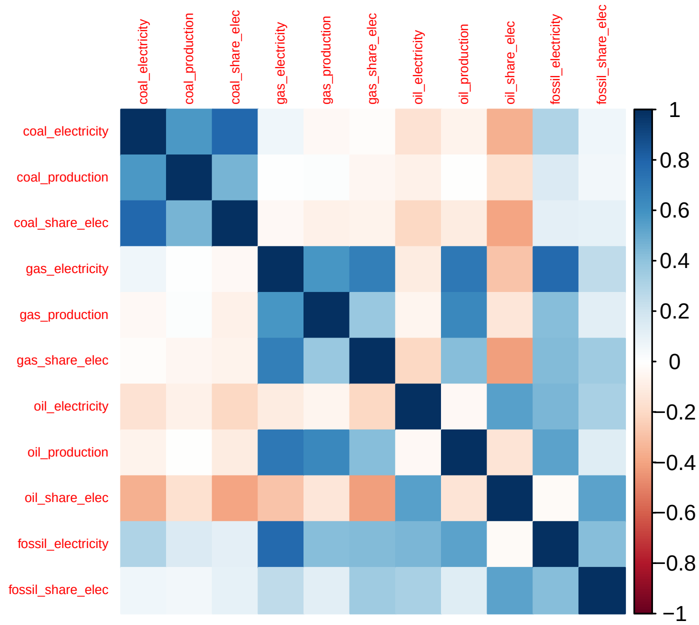



```
## [1] "Top three countries in 2016 for fossil_share_elec :"
## [1] "American Samoa , Antigua and Barbuda , Bahamas"
```

```
corrplot(cor(mainlog[,highcarb], use="pairwise.complete.obs"),
         method="color", tl.cex = .5)
```




```
corrplot(cor(mainlog2016[,highcarb], use="pairwise.complete.obs"),
         method="color", tl.cex = .5)
```

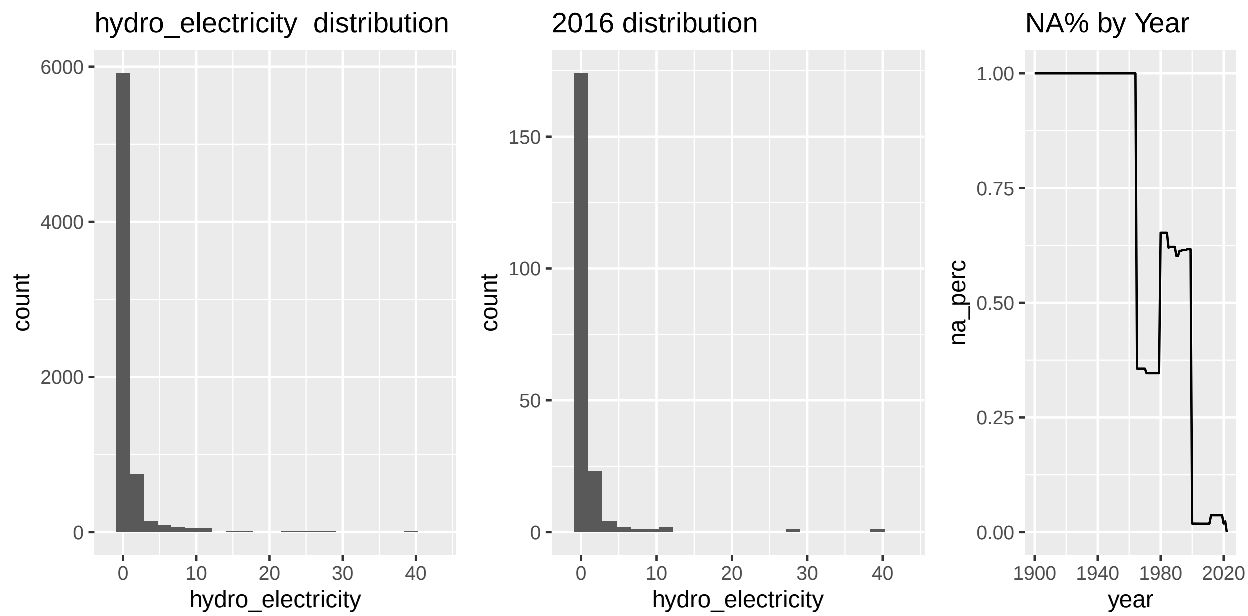


Moving on to fossil sources, we notice that **oil** overall is the most used source, but curiously oil **production** and **electricity** from oil are not correlated (and even have a negative correlation when considering the share of electricity). In contrast, for both **coal** and **gas**, the production and the electricity generation have a strong correlation, meaning that countries that use them tend to be producers; the same cannot be said for **oil**.

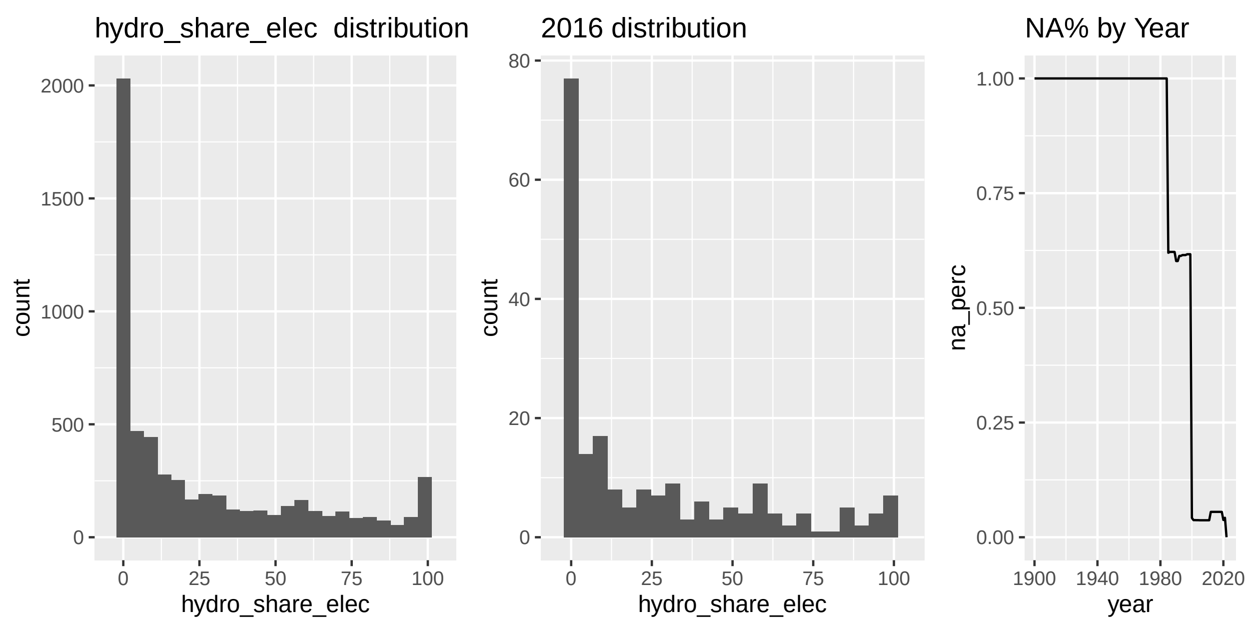
Comparing the graphs with all years versus 2016, we can notice a very slight decrease over time of use for all fossil sources.

3.4.3 Analysis on Energy dataset for low-carbon sources

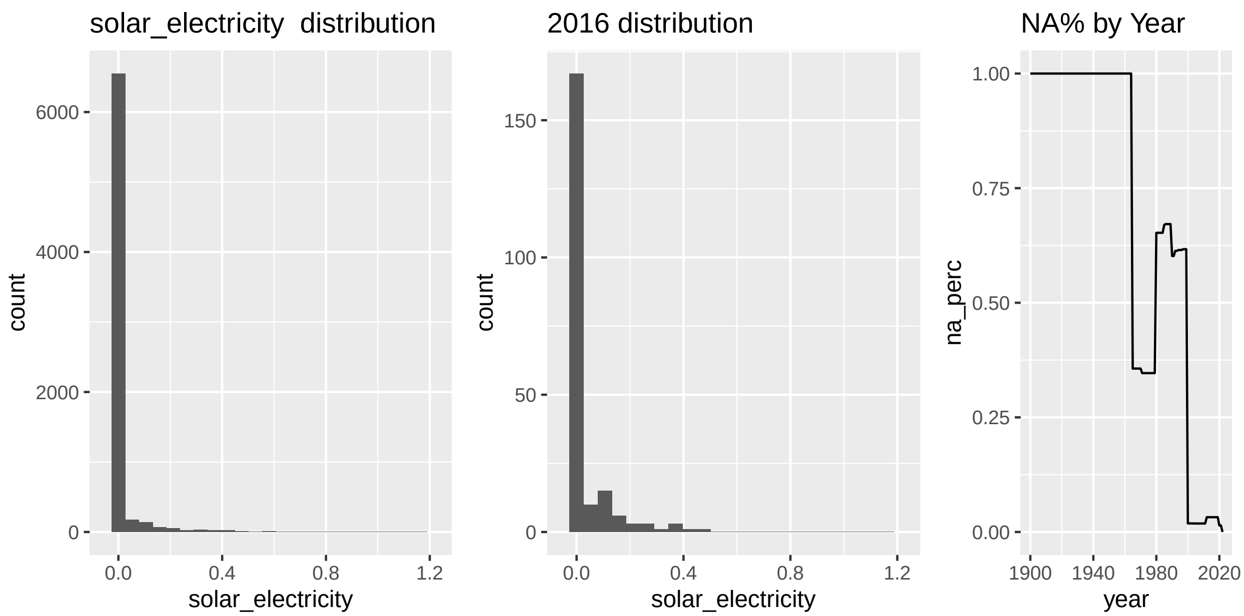
```
for (i in lowcarb){
  do_plots(i)
}
```



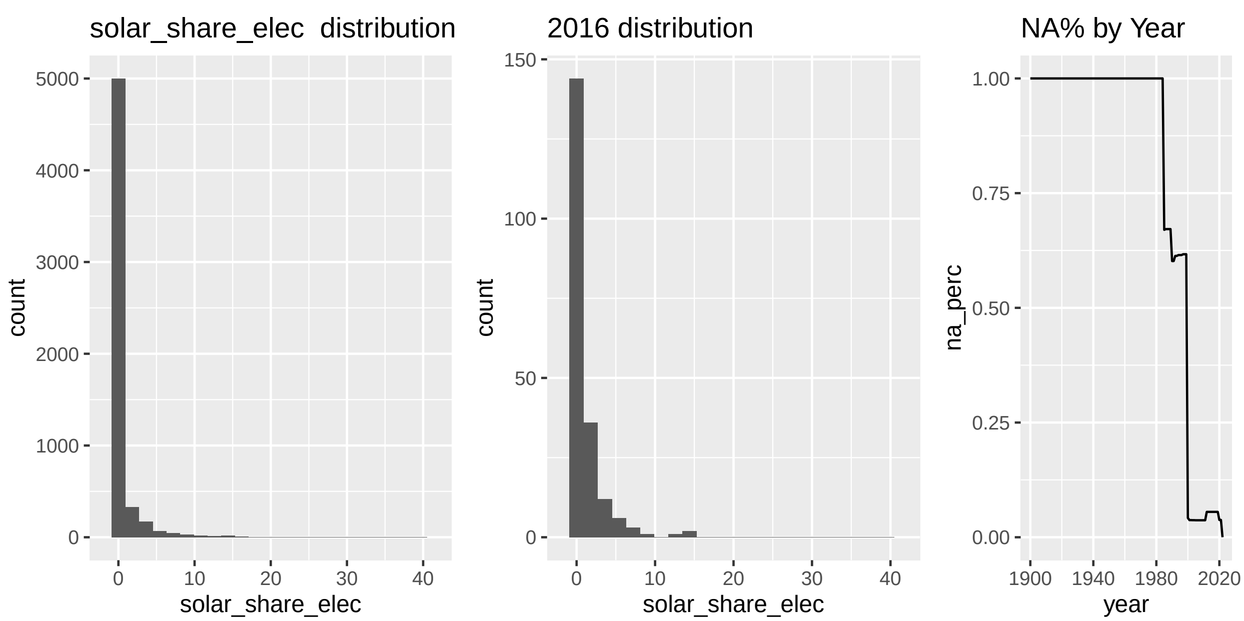

```
## [1] "Top three countries in 2016 for hydro_electricity :"
## [1] "Iceland , Norway , Canada"
```



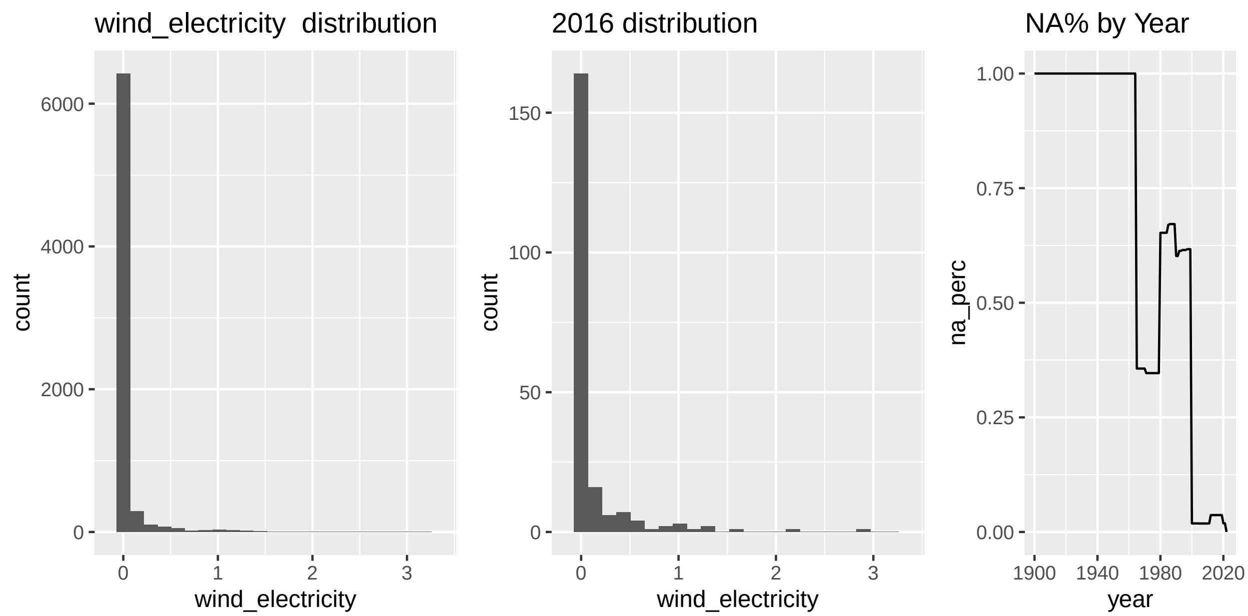

```
## [1] "Top three countries in 2016 for hydro_share_elec :"
## [1] "Albania , Bhutan , Central African Republic"
```



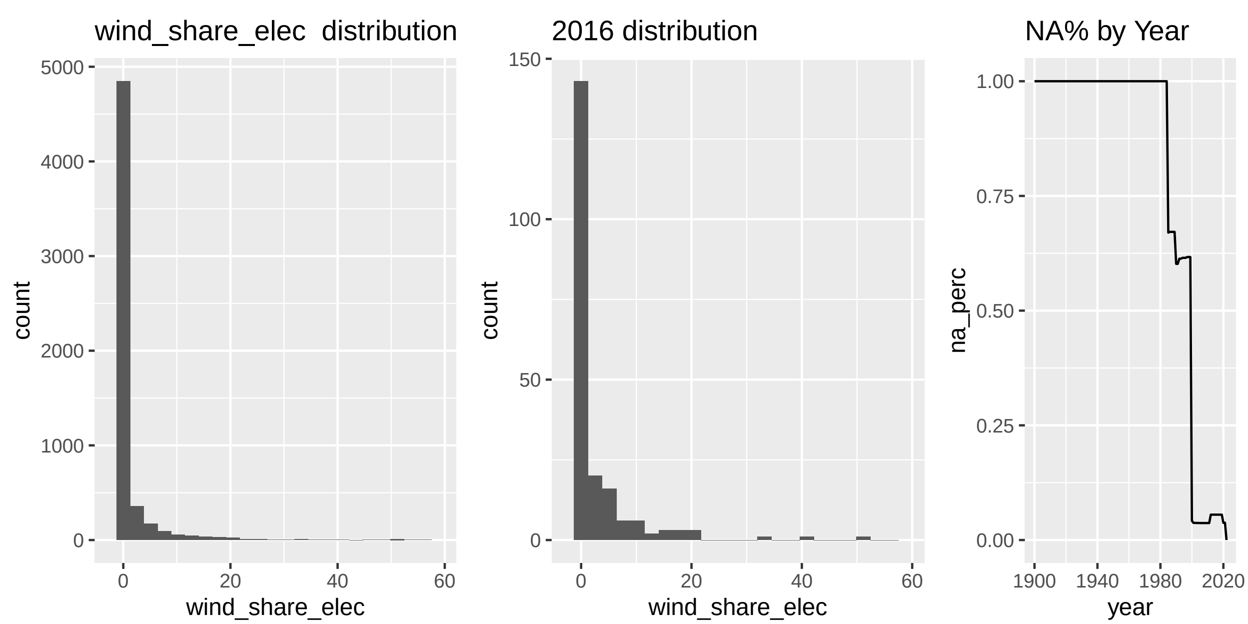

```
## [1] "Top three countries in 2016 for solar_electricity :"
## [1] "Germany , Guam , Italy"
```



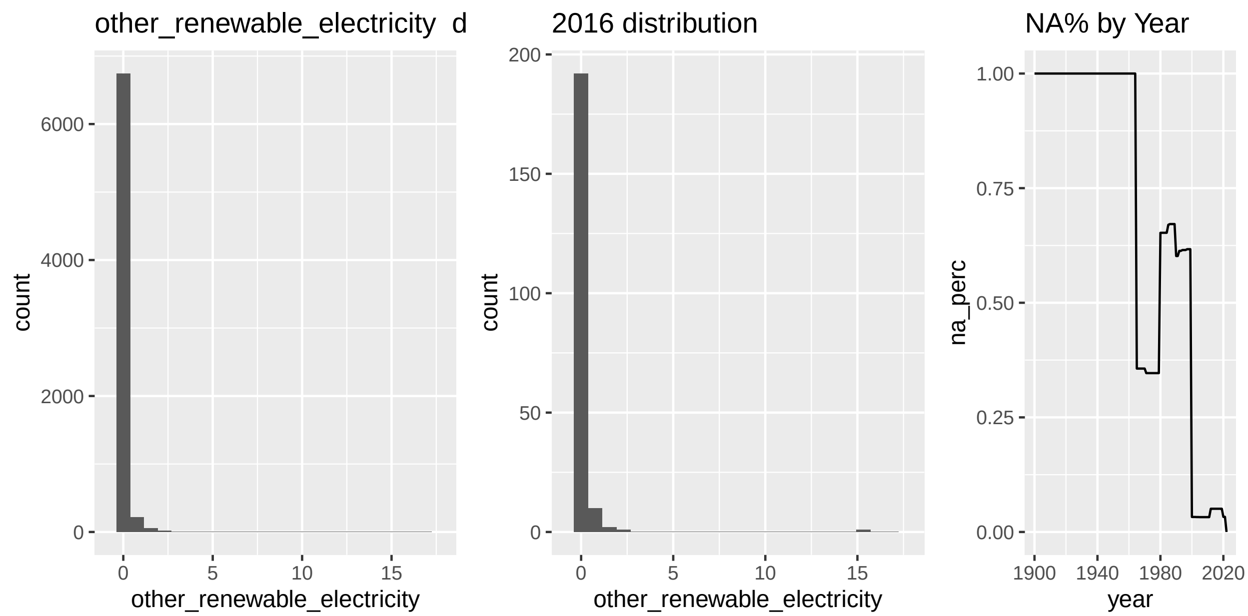

```
## [1] "Top three countries in 2016 for solar_share_elec :"
## [1] "Malta , Samoa , Luxembourg"
```



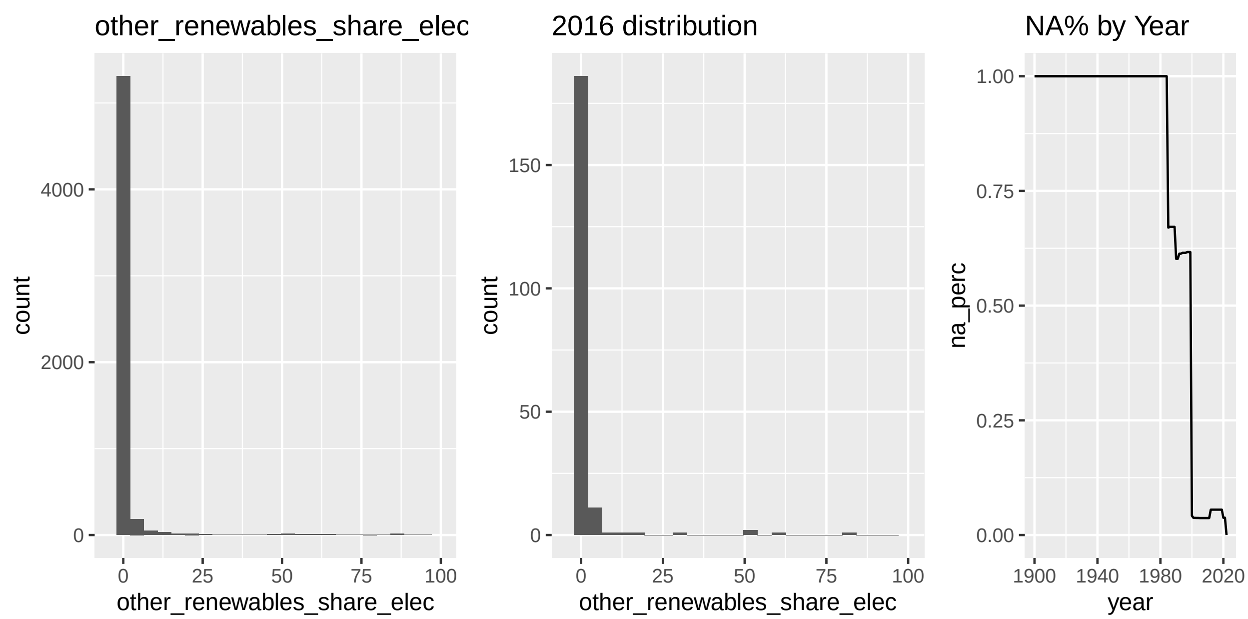

```
## [1] "Top three countries in 2016 for wind_electricity :"
## [1] "Falkland Islands , Denmark , Sweden"
```



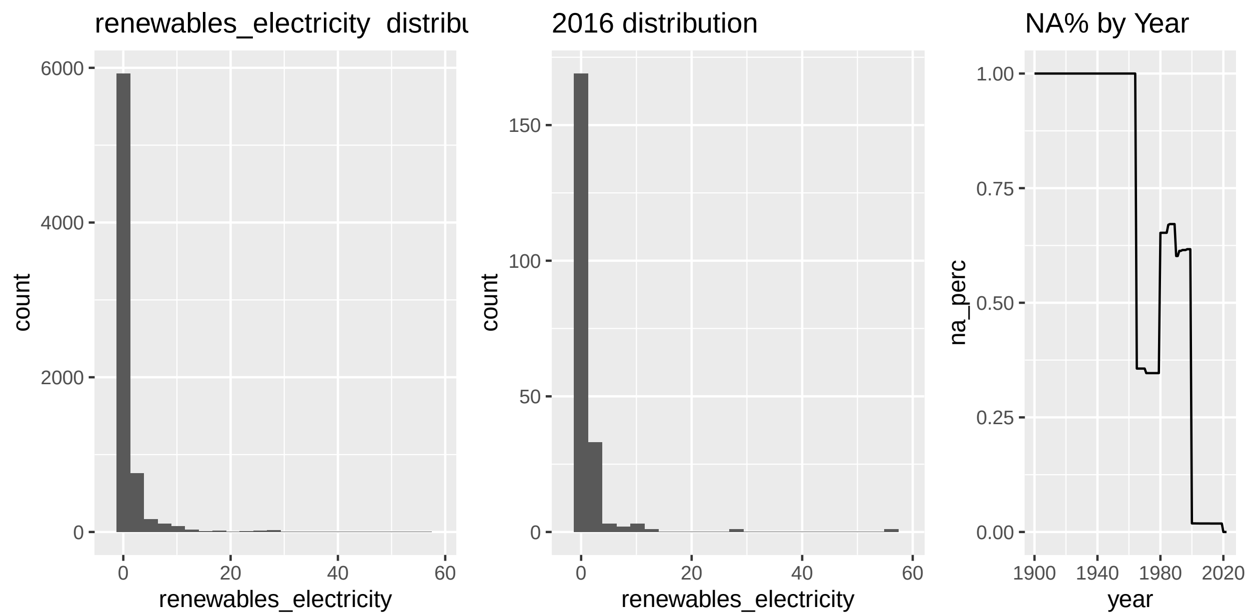

```
## [1] "Top three countries in 2016 for wind_share_elec :"
## [1] "Falkland Islands , Denmark , Lithuania"
```



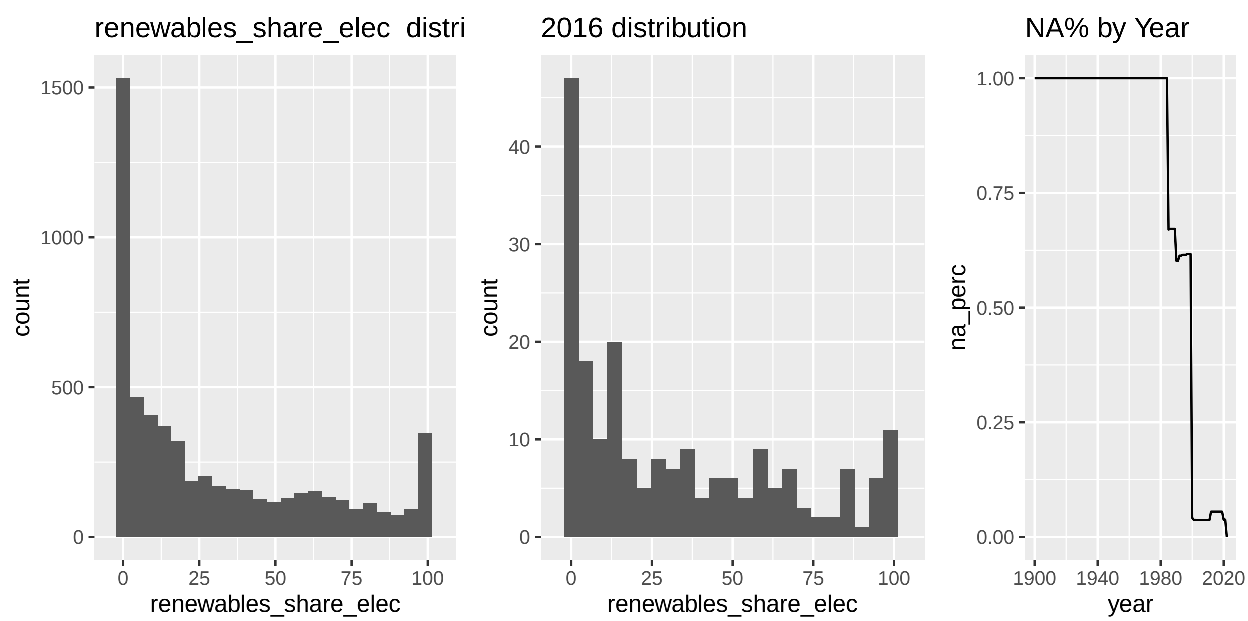

```
## [1] "Top three countries in 2016 for other_renewable_electricity :"
## [1] "Iceland , Finland , New Zealand"
```



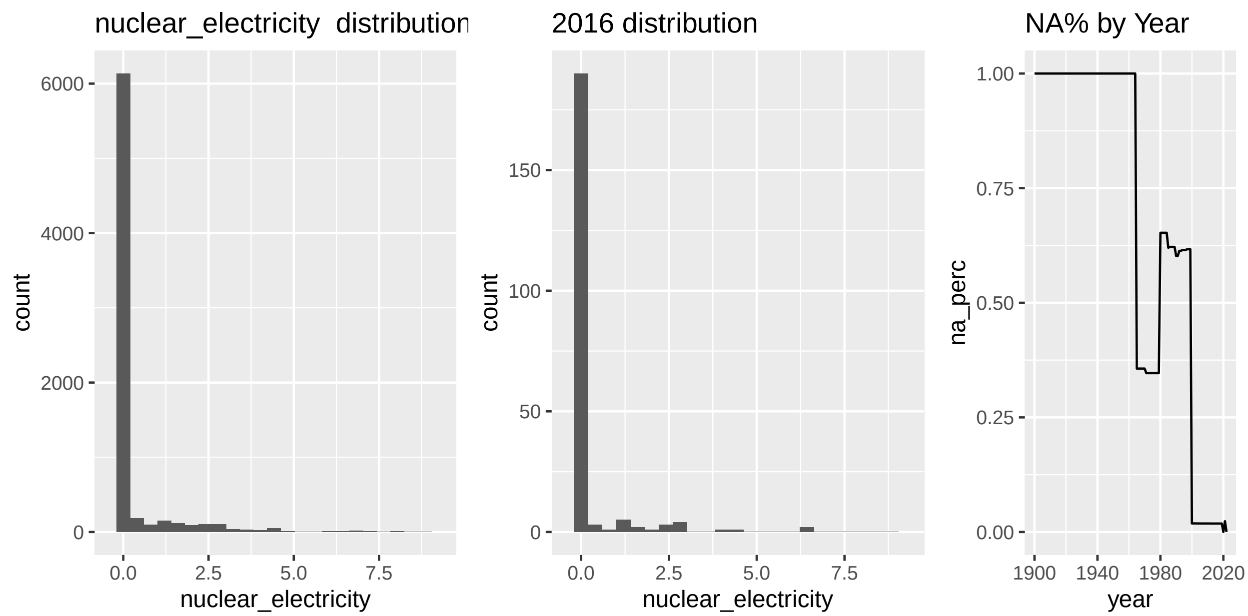

```
## [1] "Top three countries in 2016 for other_renewables_share_elec :"
## [1] "Iceland , Eswatini , Belize"
```



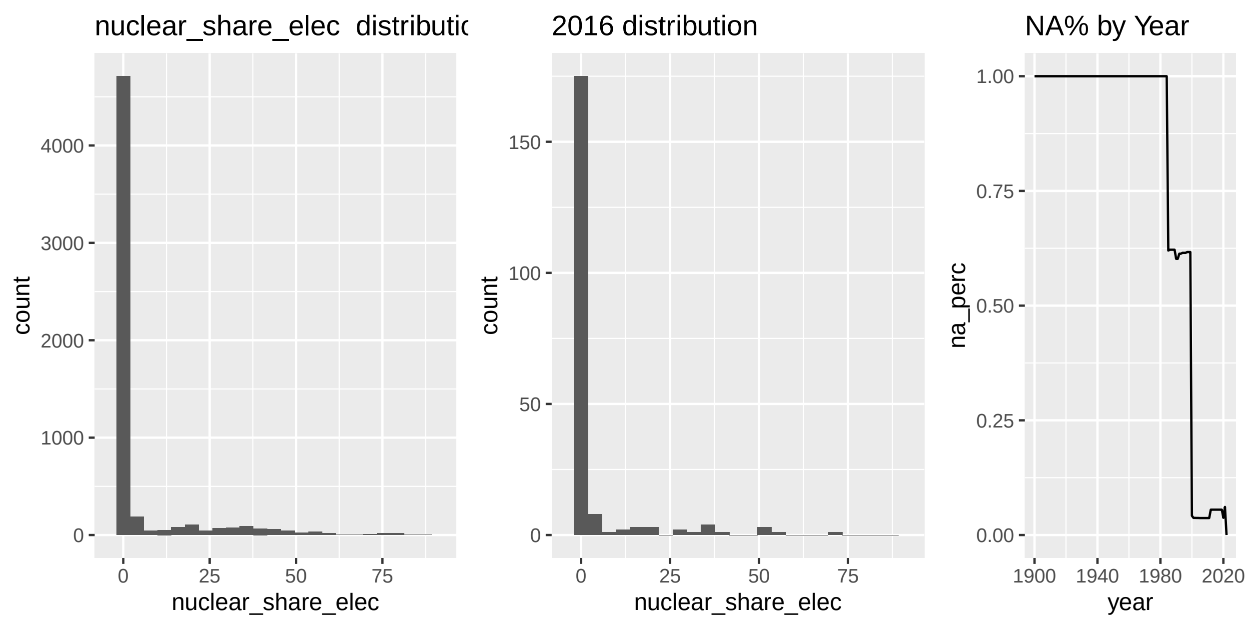

```
## [1] "Top three countries in 2016 for renewables_electricity :"
## [1] "Iceland , Norway , Canada"
```



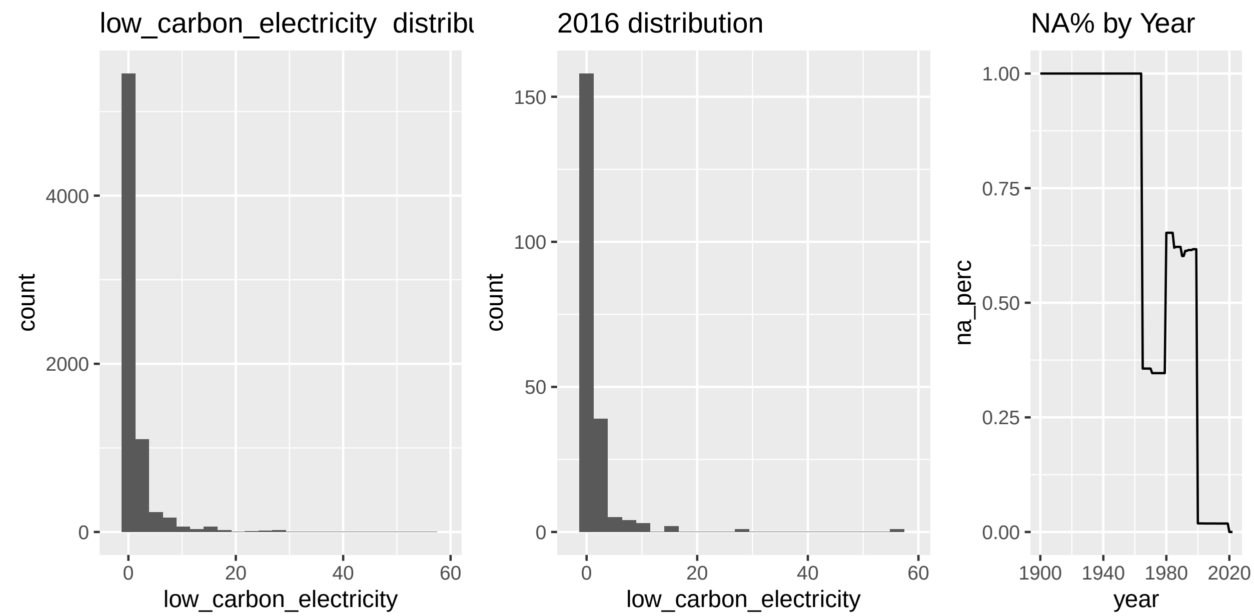

```
## [1] "Top three countries in 2016 for renewables_share_elec :"
## [1] "Albania , Bhutan , Central African Republic"
```



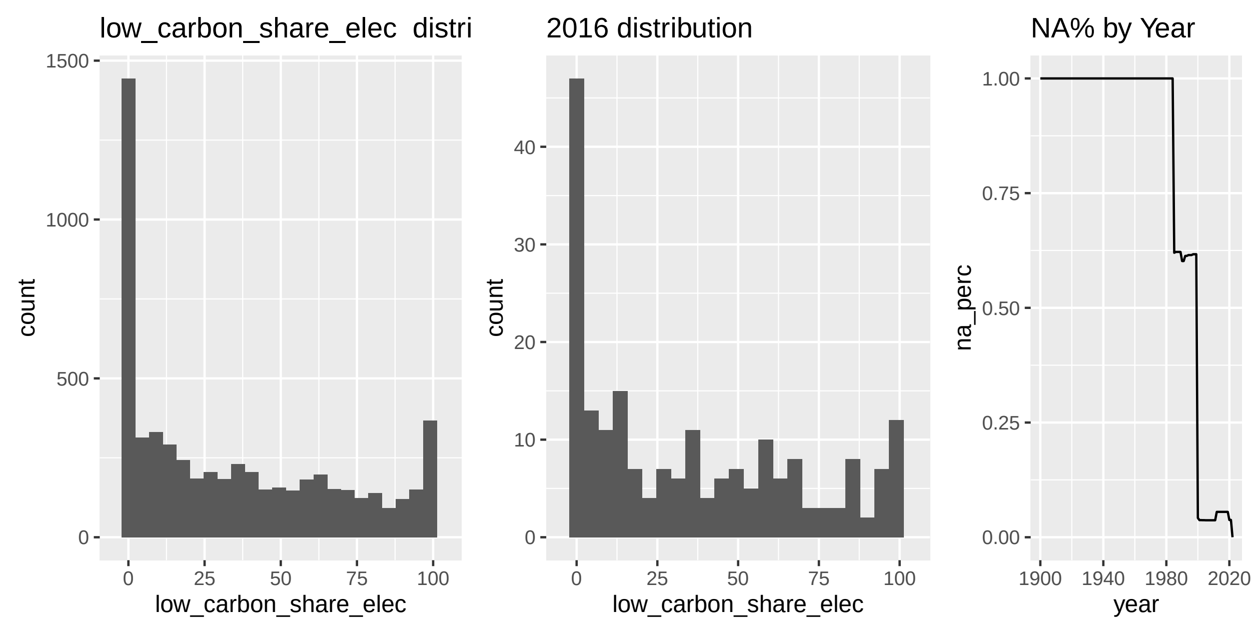

```
## [1] "Top three countries in 2016 for nuclear_electricity :"
## [1] "Sweden , France , Finland"
```




```
## [1] "Top three countries in 2016 for nuclear_share_elec :"
## [1] "France , Slovakia , Belgium"
```

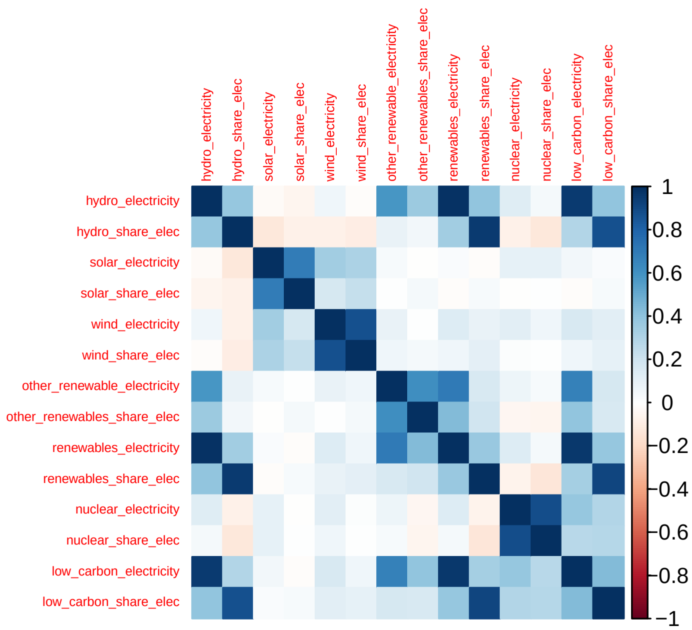


[1] "Top three countries in 2016 for low_carbon_electricity :"
[1] "Iceland , Norway , Sweden"

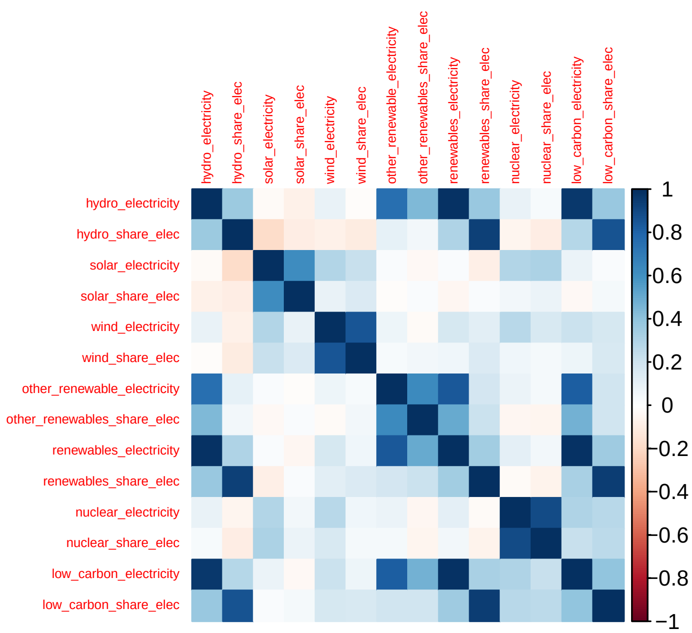


[1] "Top three countries in 2016 for low_carbon_share_elec :"
[1] "Albania , Bhutan , Central African Republic"

```
corrplot(cor(mainlog[,lowcarb], use="pairwise.complete.obs"),
         method="color", tl.cex = .5)
```




```
corrplot(cor(mainlog2016[,lowcarb], use="pairwise.complete.obs"),
         method="color", tl.cex = .5)
```



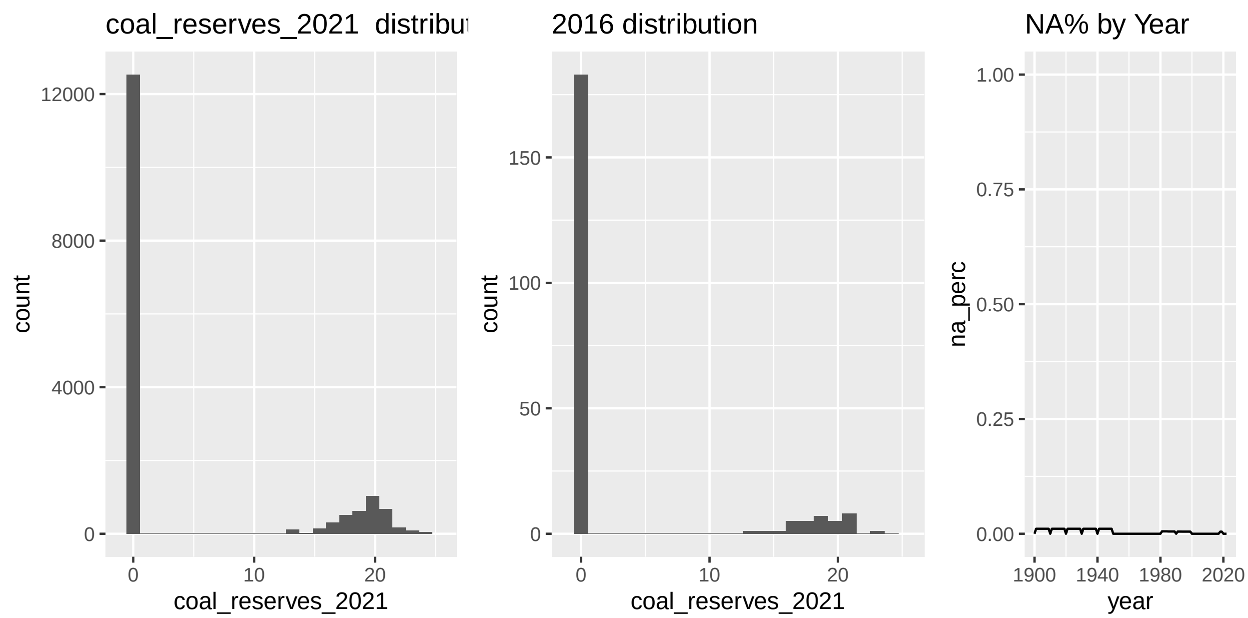
Again, for **low-carbon sources**, we see that most countries only share records from 2000 onwards.

Electricity from low-carbon sources mainly comes from **hydropower**, with all other sources having distributions close to 0, with some significant outliers. In fact, hydro is almost collinear with overall electricity production and share for renewables and low carbon. The main reason is that hydro is historically the most used renewable source, as it is extremely cheap compared to the other low-carbon sources and has a different purpose other than electricity production.

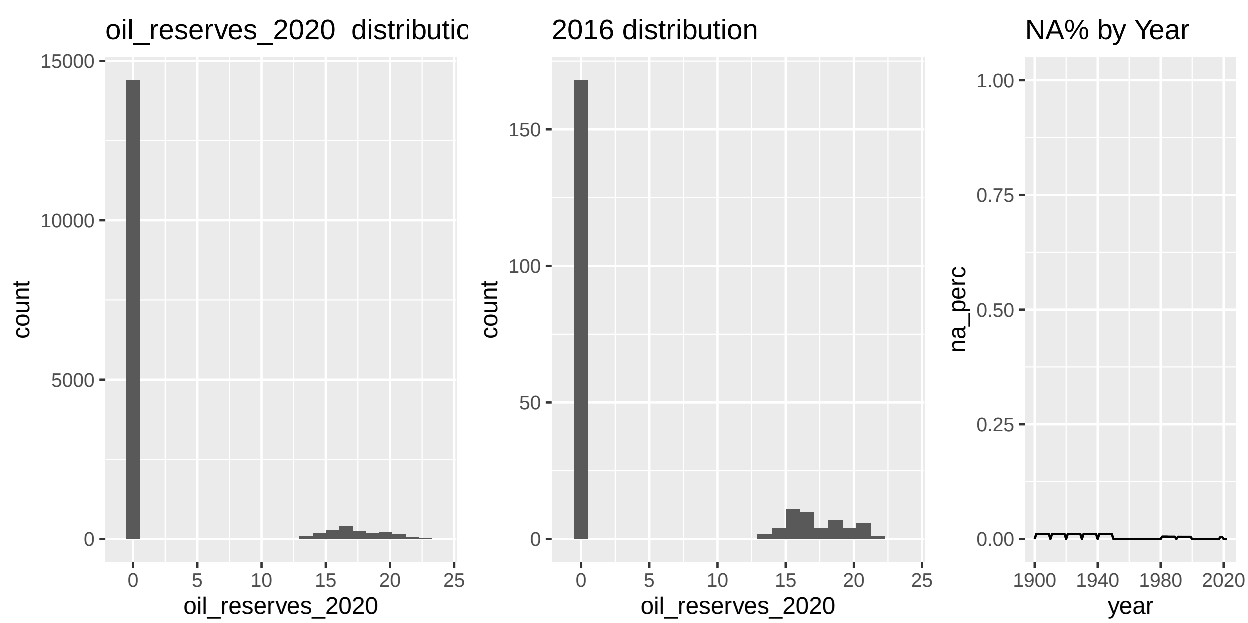
Solar and **Wind** are correlated, however, the correlation has gotten weaker over time; this is probably due to the cost of the sources, which shrunk over the years, so it became feasible for more countries to invest in renewable sources and to do so in the way that best fits the country availability.

3.4.4 Analysis on external reserves variables

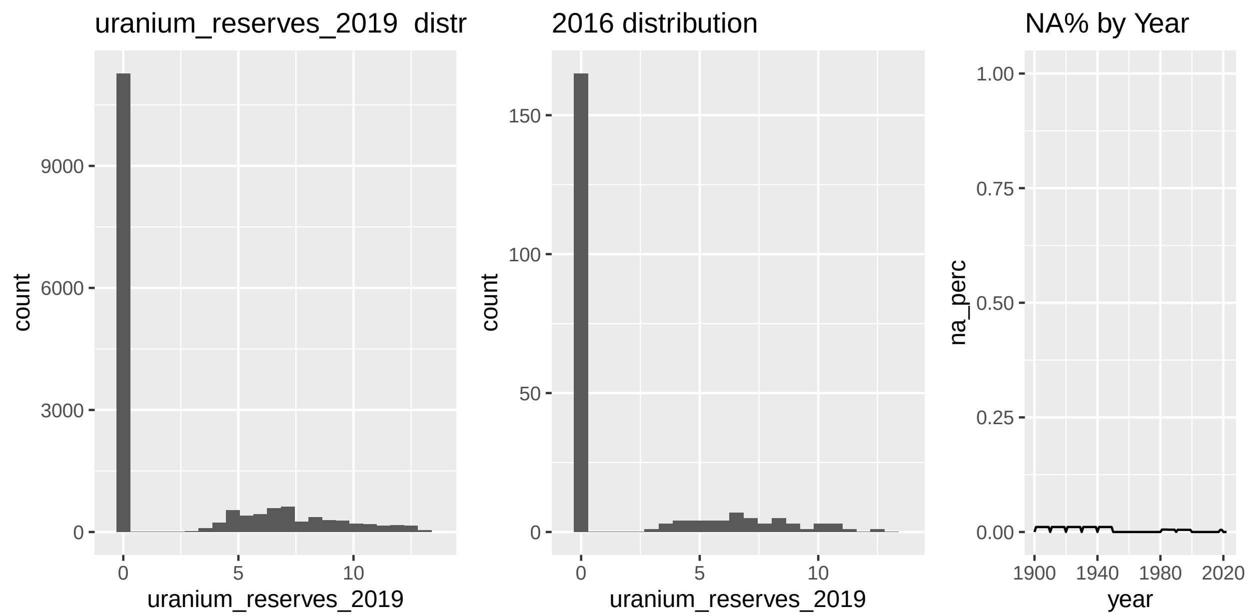
```
for (i in reserves){
  do_plots(i)
}
```



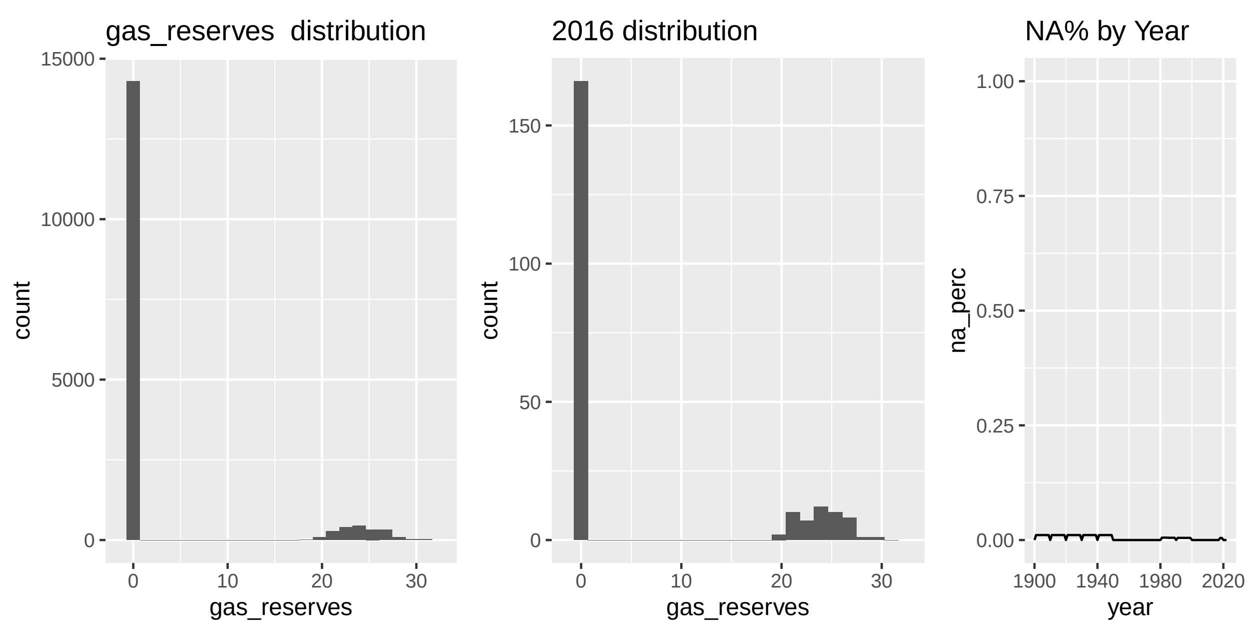

```
## [1] "Top three countries in 2016 for coal_reserves_2021 :"
## [1] "Australia , New Zealand , Kazakhstan"
```




```
## [1] "Top three countries in 2016 for oil_reserves_2020 :"
## [1] "Kuwait , United Arab Emirates , Venezuela"
```

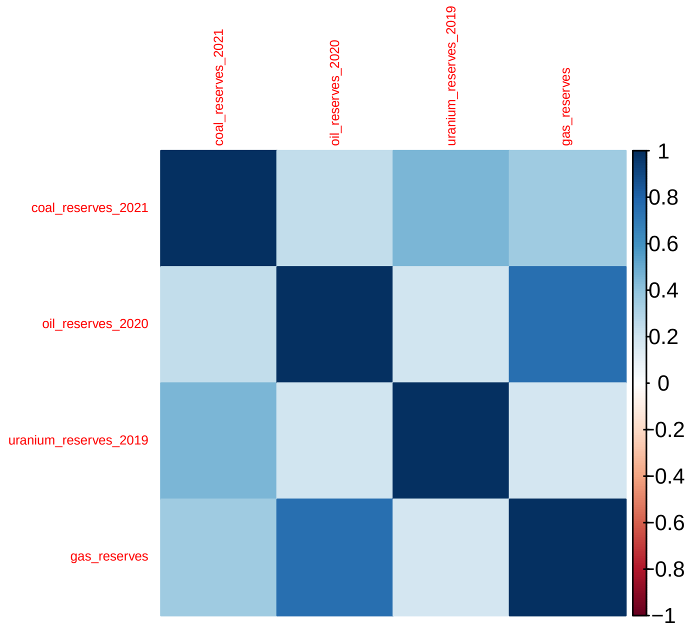



```
## [1] "Top three countries in 2016 for uranium_reserves_2019 :"
## [1] "Namibia , Australia , Kazakhstan"
```




```
## [1] "Top three countries in 2016 for gas_reserves :"
## [1] "Qatar , Turkmenistan , United Arab Emirates"
```

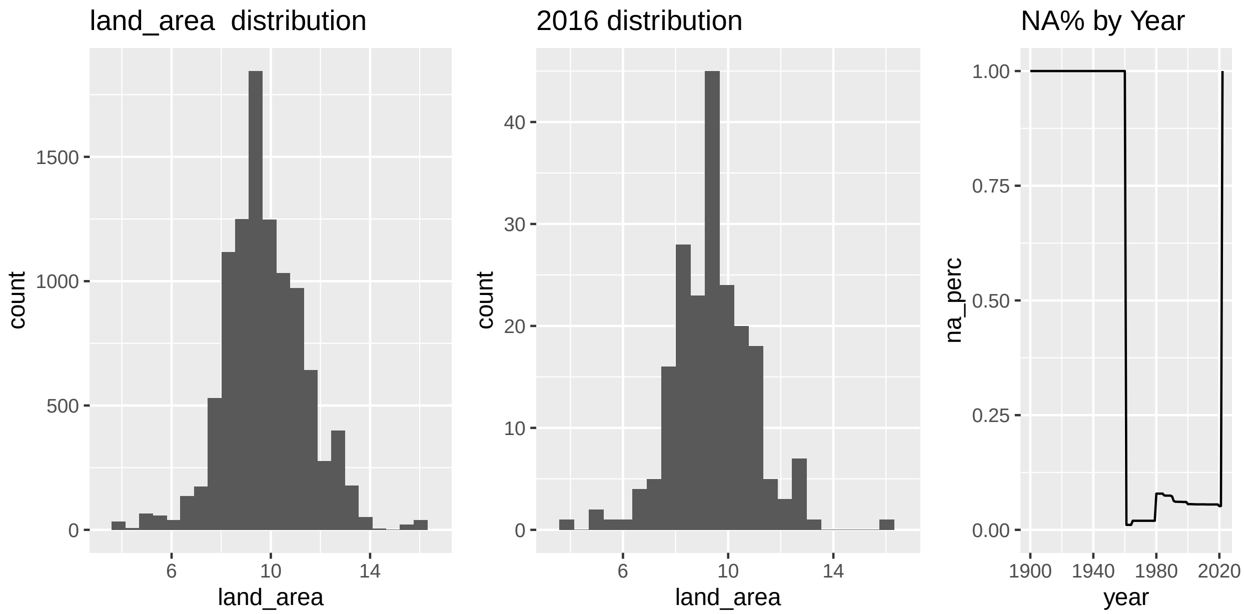
```
corrplot(cor(mainlog2016[,reserves], use="pairwise.complete.obs"),
         method="color", tl.cex = .5)
```



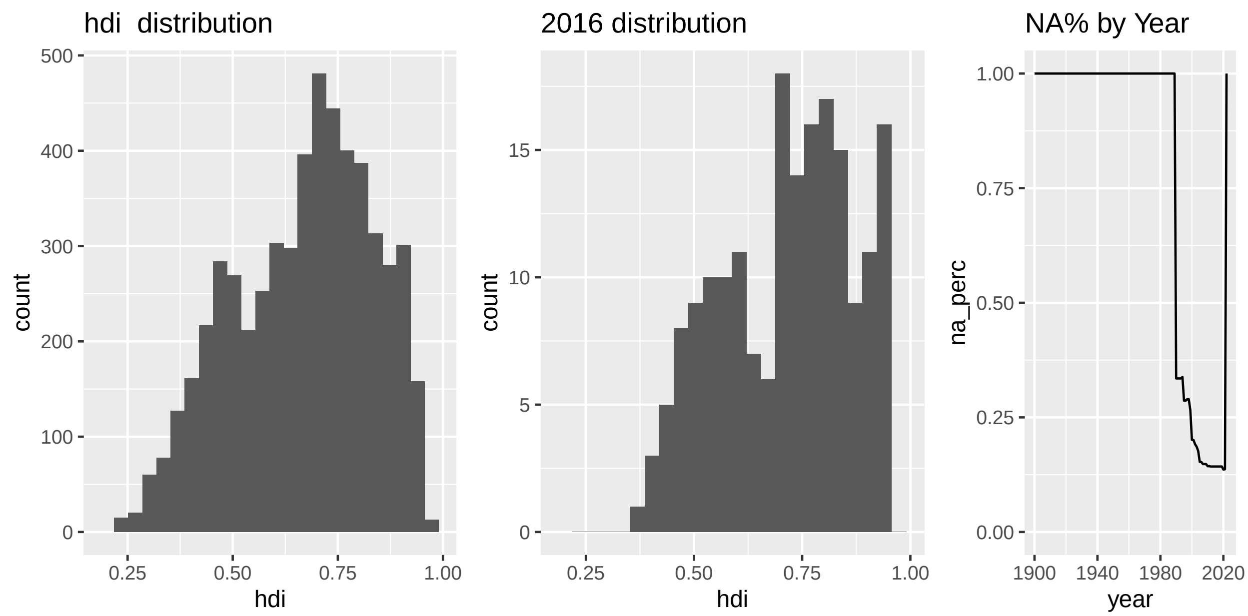
Most countries have no reserves; however, the distribution of reserves for the countries that do have them seem to be normally shaped (after transformations). A really interesting insight is that the different types of reserves are correlated between each others.

3.4.5 Analysis on other external variables

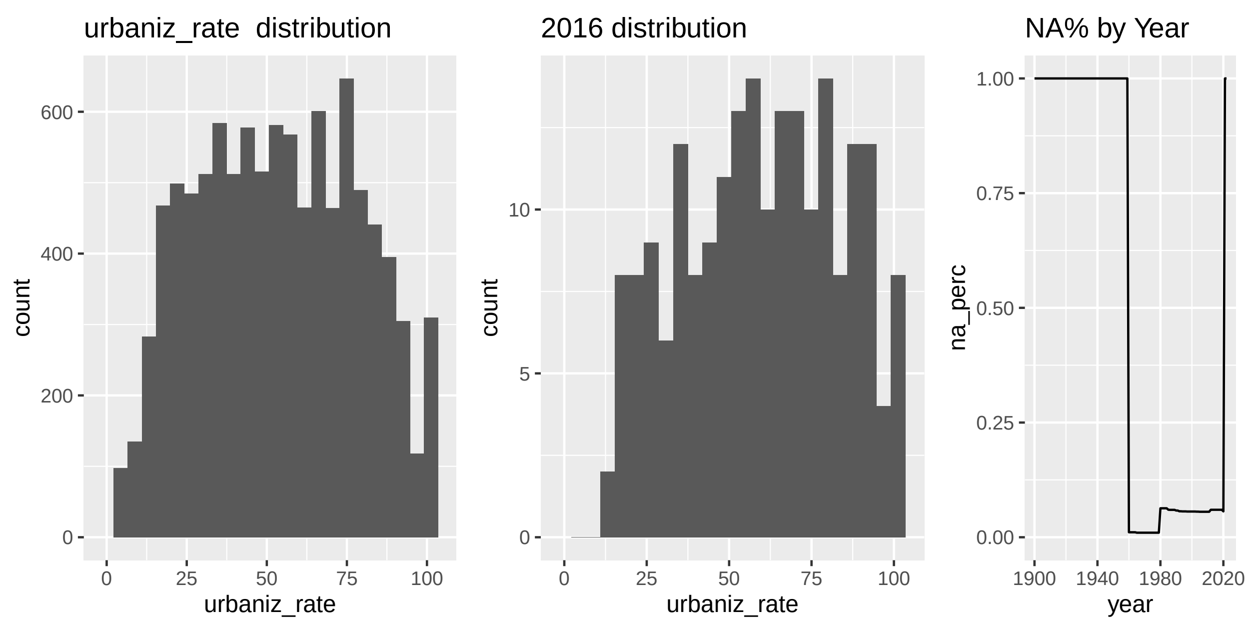
```
for (i in ext_measures){
  do_plots(i)
}
```



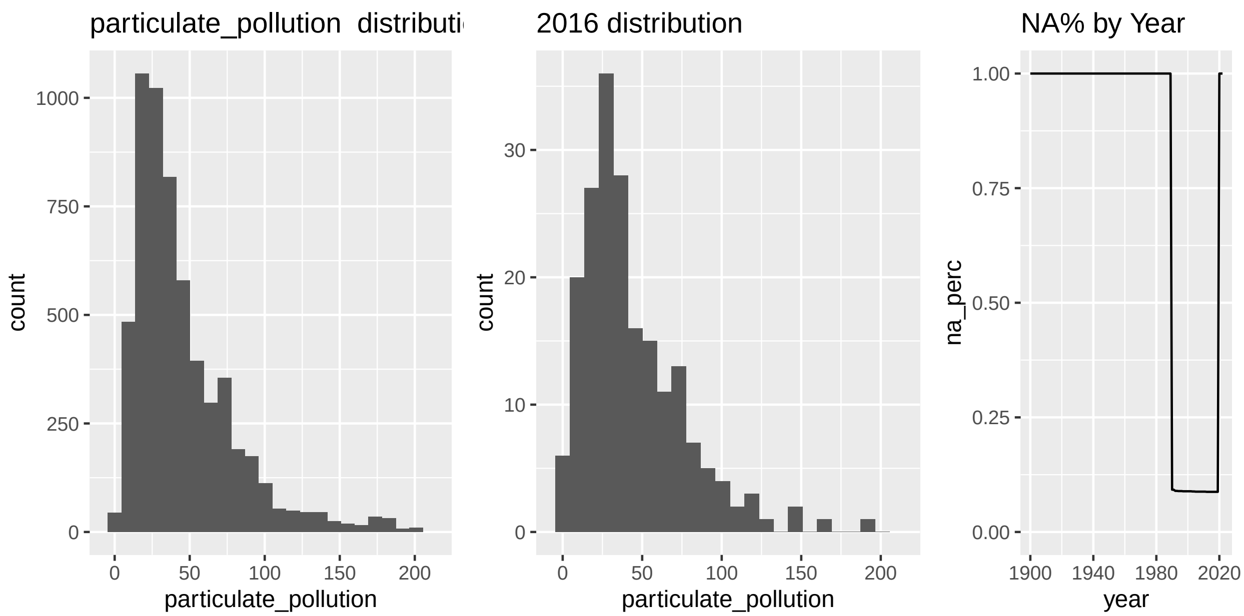

```
## [1] "Top three countries in 2016 for land_area :"
## [1] "Greenland , Mongolia , Namibia"
```



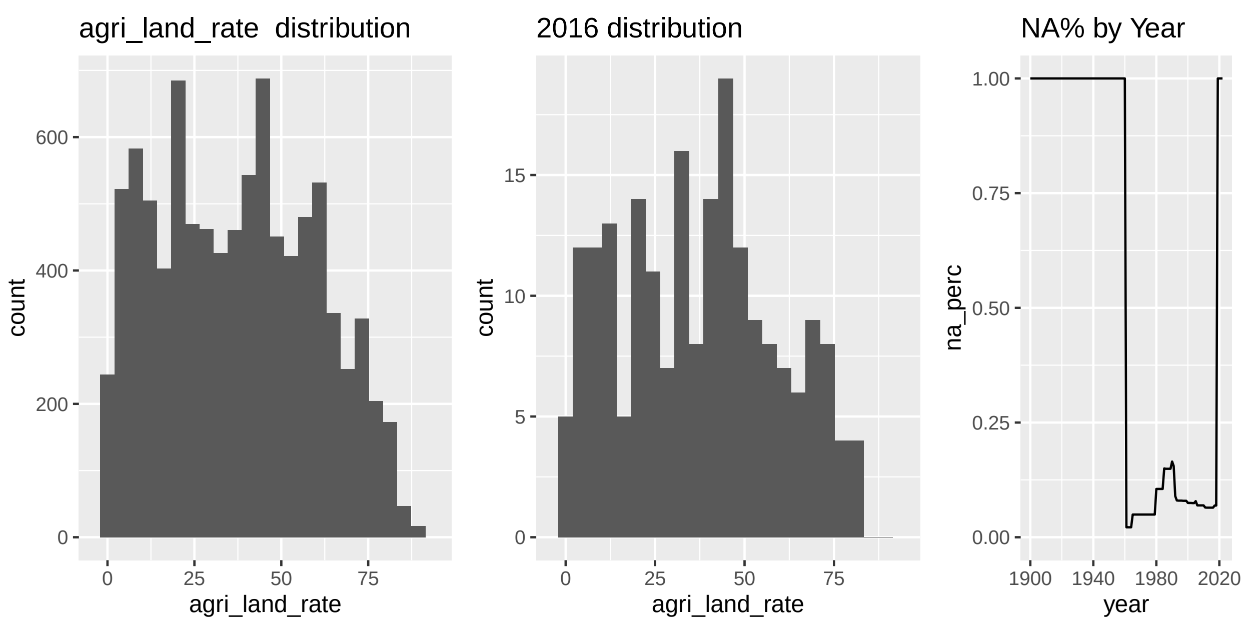

```
## [1] "Top three countries in 2016 for hdi :"
## [1] "Switzerland , Norway , Iceland"
```



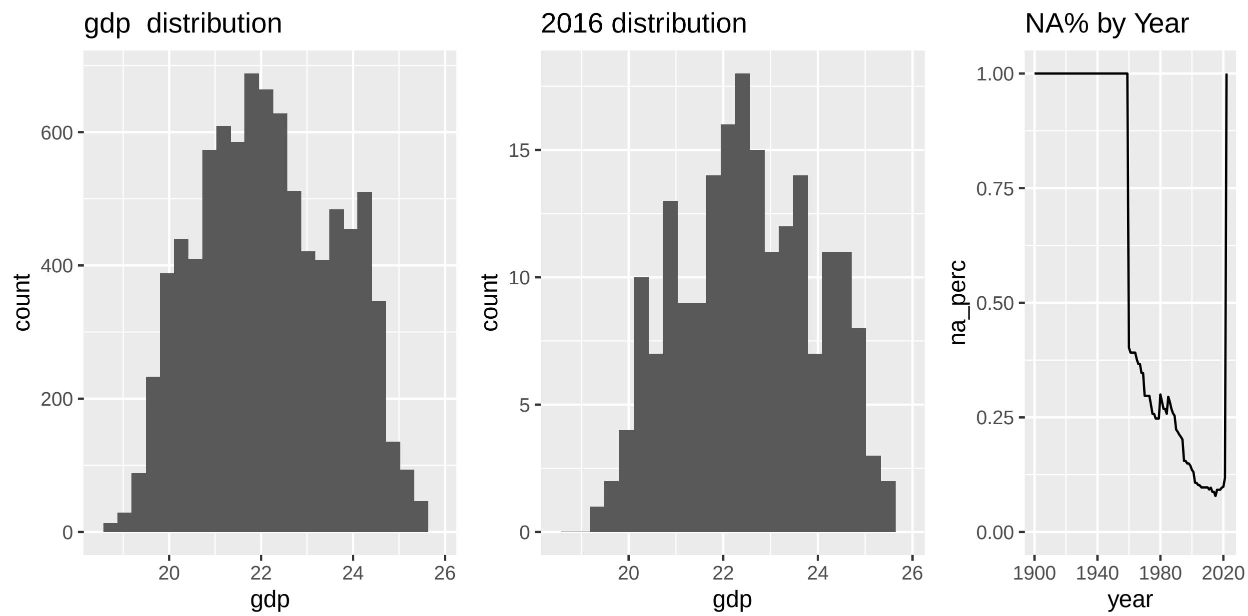

```
## [1] "Top three countries in 2016 for urbaniz_rate :"
## [1] "Bermuda , Cayman Islands , Gibraltar"
```




```
## [1] "Top three countries in 2016 for particulate_pollution :"
## [1] "Uzbekistan , Egypt , Oman"
```

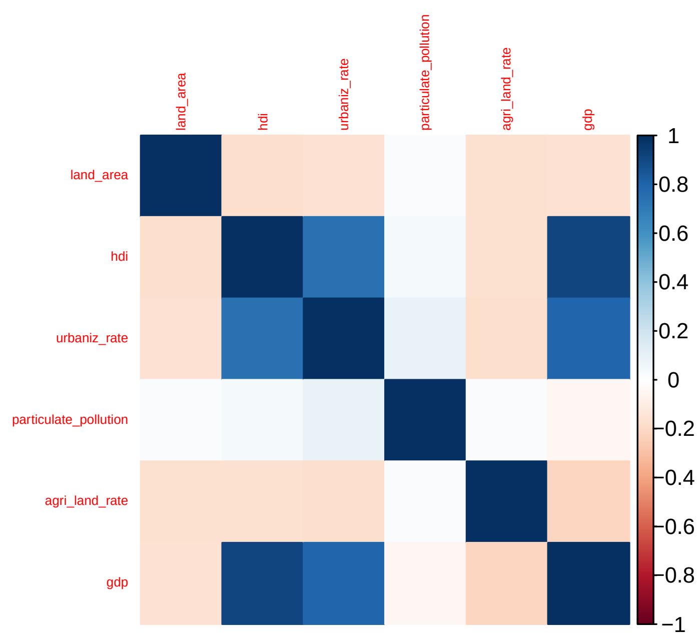



```
## [1] "Top three countries in 2016 for agri_land_rate :"
## [1] "Saudi Arabia , Uruguay , Kazakhstan"
```

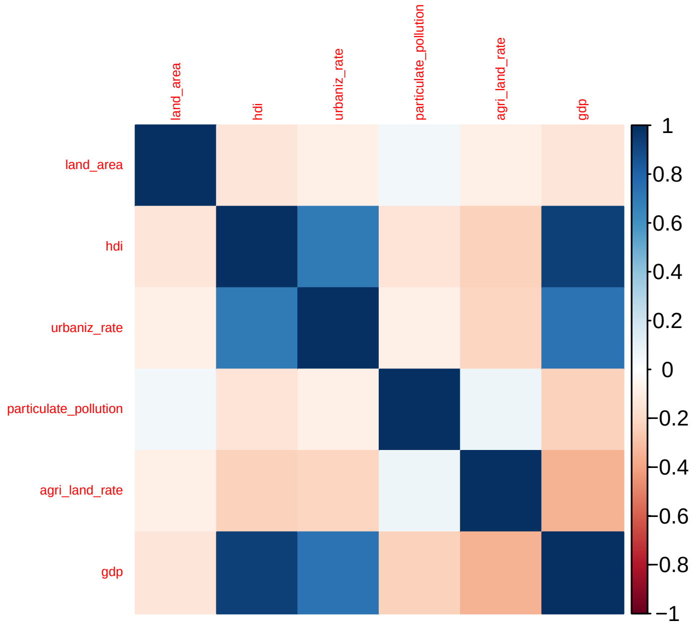



```
## [1] "Top three countries in 2016 for gdp :"
## [1] "Luxembourg , Bermuda , Switzerland"
```

```
corrplot(cor(mainlog[,ext_measures], use="pairwise.complete.obs"),
         method="color", tl.cex = .5)
```




```
corrplot(cor(mainlog2016[,ext_measures], use="pairwise.complete.obs"),
         method="color", tl.cex = .5)
```

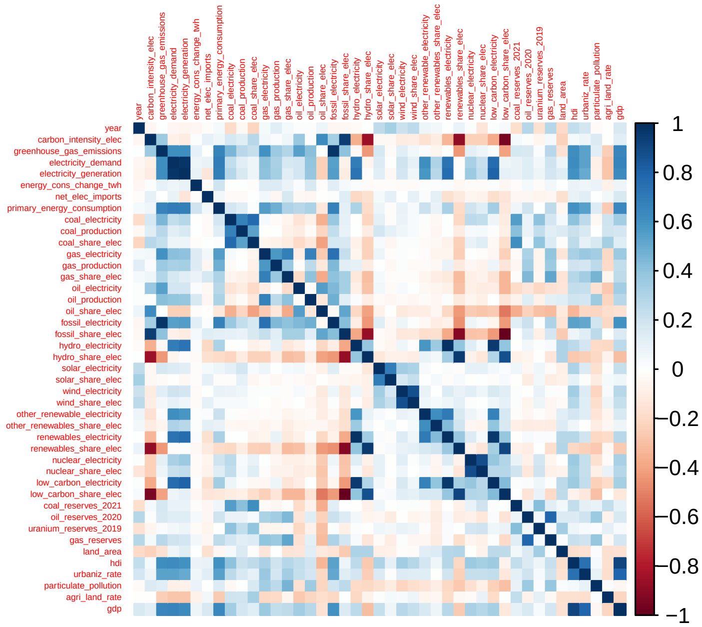


After being transformed, the variables external to the Energy dataset all have a Gaussian-like distribution.

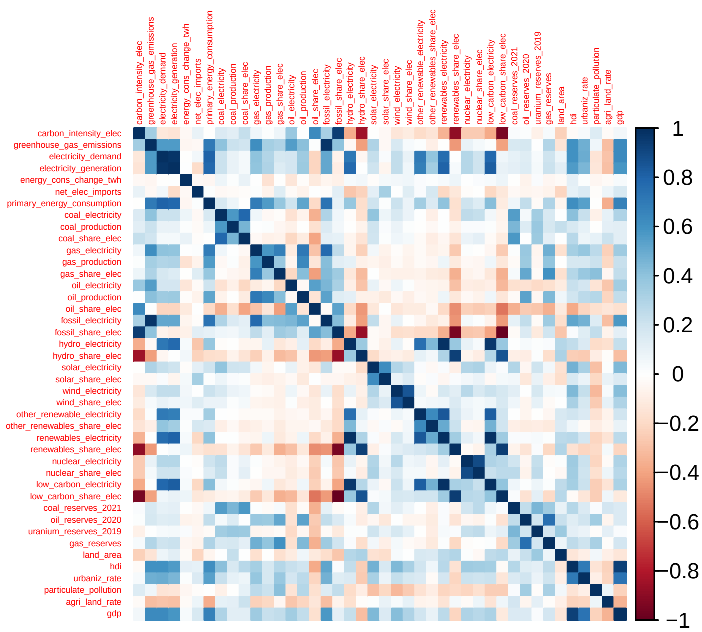
As expected, **GDP** and **HDI** are strongly correlated; also, urbanization rate has a good correlation with those two variables; all three are slightly negatively correlated to **land area** and **agricultural land rate**. **Particulate pollution** is not correlated to any other variables.

3.4.6 Inter-groups correlations

```
subsetdf = mainlog[,c(2,other_measures,highcarb,lowcarb,reserves,ext_measures)]
subsetdf2016 = mainlog2016[,c(other_measures,highcarb,lowcarb,reserves,ext_measures)]

corrplot(cor(subsetdf, use="pairwise.complete.obs"),
         method="color", tl.cex = .3)
```




```
corrplot(cor(subsetdf2016, use="pairwise.complete.obs"),
         method="color", tl.cex = .3)
```



Finally, we plot correlation among all variables, considering year as one of those. The main findings are the following:

1. **Year** has a slightly positive correlation with **solar** and **wind**, and a slight negative correlation with **coal**.
2. Some noticeable correlations between variables in different groups are the ones between renewables (hydro) and fossil sources to the variables measuring pollution.
3. **Coal** and **gas electricity** productions are correlated to their respective **reserves**, while **oil electricity** has a negative correlation to its ones.

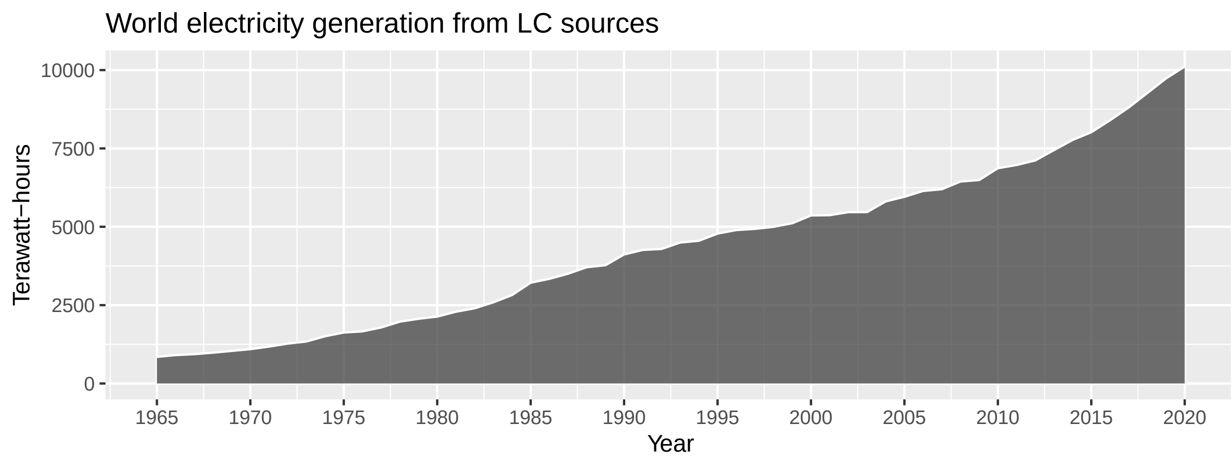
4. **Nuclear** is slightly positively correlated to **uranium reserves**, but it is equally correlated to the economic indexes (**GDP**, **HDI**).

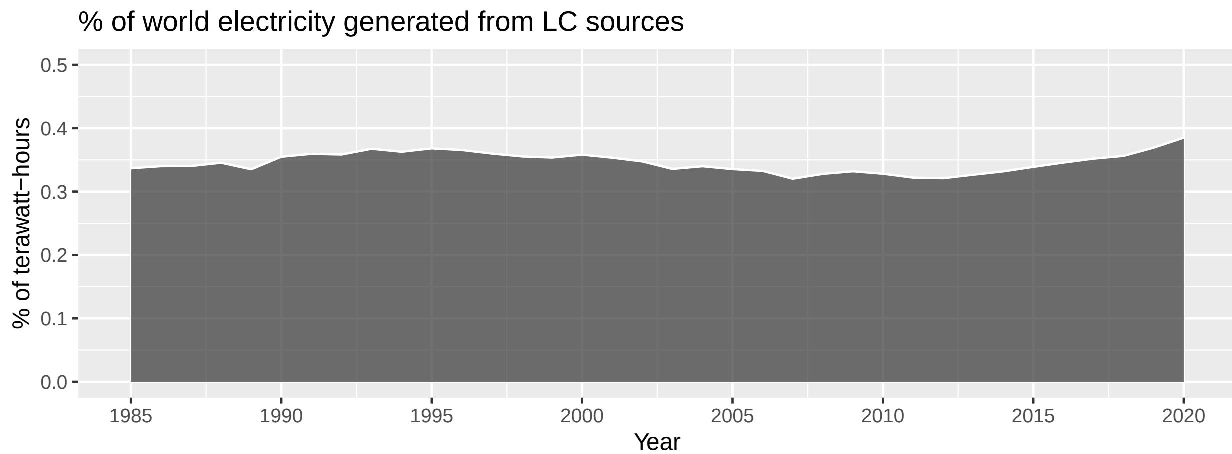
Chapter 4: Descriptive analyses

4.1 Global analyses

We start the descriptive analyses by looking at the global low-carbon electricity generation trends. By **low-carbon**, we refer to the electricity produced with substantially lower greenhouse gas emissions than conventional fossil fuel power generation [12]. In other words, the term low-carbon includes renewable and nuclear sources. We will now refer to it with the acronym **LC**.

```r
# 1. World electricity generation from LC sources

# a. Area plot of the total generation
# Creation of the dataset
place = main[c("year", "low_carbon_electricity")] %>%
  mutate_all(~replace_na(.,0)) %>%
  group_by(year) %>%
  summarize(sum_lc = sum(low_carbon_electricity))

# Creation of the plot
gg1 = ggplot(place, aes(year, sum_lc)) +
  geom_area(alpha = 0.7, colour="white") +
  scale_x_continuous(limits = c(1965,2020), breaks = seq(1965, 2020, by = 5)) +
  labs(title = "World electricity generation from LC sources",
       x = "Year",
       y = "Terawatt-hours")

# b. Area plot of the ratio between LC generation and total electricity generation
# Creation of the dataset
place = main[c("year", "low_carbon_electricity", "electricity_generation")] %>%
  mutate_all(~replace_na(.,0)) %>%
  group_by(year) %>%
  summarize(sum_lc = sum(low_carbon_electricity)/sum(electricity_generation))

# Creation of the plot
gg2 = ggplot(place, aes(year, sum_lc)) +
  geom_area(alpha = 0.7, colour="white") +
  scale_x_continuous(limits = c(1985,2020), breaks = seq(1985, 2020, by = 5)) +
  scale_y_continuous(limits = c(0,0.5)) +
  labs(title = "% of world electricity generated from LC sources",
       x = "Year",
       y = "% of terawatt-hours")

# Visualization of gg1 and gg2
grid.arrange(gg1,gg2)
```



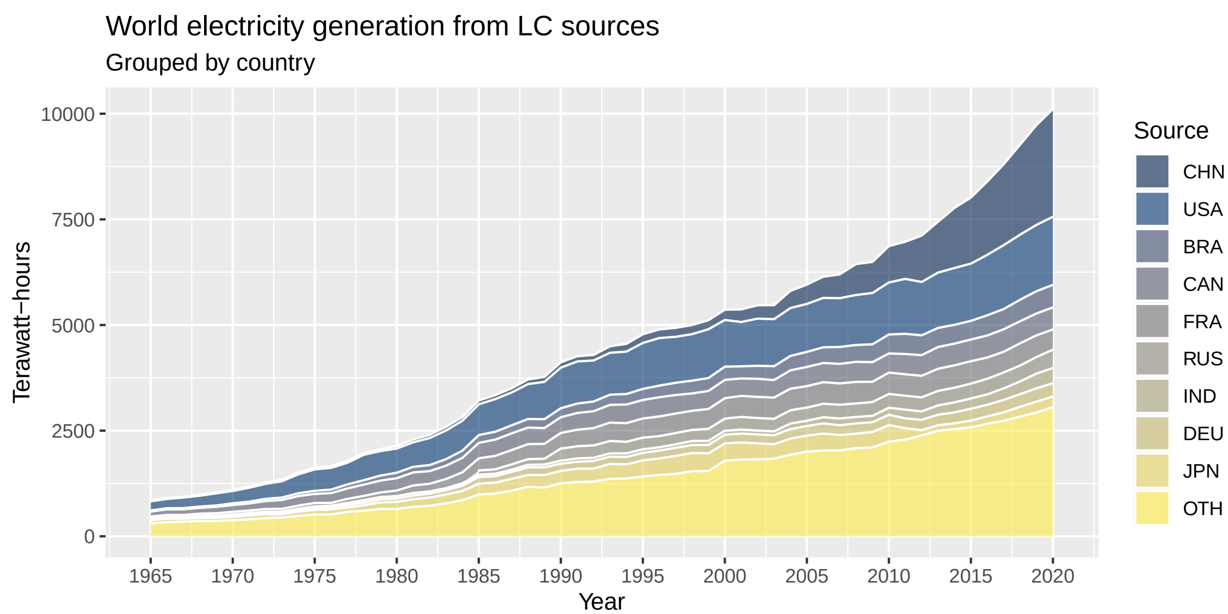


World's electricity production from LC sources constantly grew, going from a generation of less than 1000 TwH in 1965 to more than 10000 TwH in 2020, with an impressive average yearly growth of 21.6%.

Nonetheless, the second image clarifies an important point: even if the electricity generation from LC sources increased, the share of world electricity generated from LC sources has been stationary over the years, except for a timid increase from 2013 to 2020.

```
# 2. World electricity generation from LC sources, grouped by countries

# a. Creation of a vector containing the ISO codes of the nine countries with the highest
# LC electricity production in 2020
place = main[c("year", "low_carbon_electricity")] %>% mutate_all(~replace_na(.,0))
place = cbind(iso_code = main$iso_code, place) %>%
  filter(year == 2020) %>%
  arrange(desc(low_carbon_electricity))
place_2 = place$iso_code[1:9]

# b. Group the other countries in a single class called "OTH" ("others")
place = main[c("year", "low_carbon_electricity")] %>% mutate_all(~replace_na(.,0))
place = cbind(iso_code = main$iso_code, place)
for(i in 1:nrow(place)){
  place$iso_code[i] = ifelse(place$iso_code[i] %in% place_2, place$iso_code[i], "OTH")
}
place = group_by(place, iso_code, year) %>%
  summarize(sum_lc = sum(low_carbon_electricity))
```

```r
place$iso_code = factor(place$iso_code,
                        levels = c("CHN", "USA", "BRA", "CAN", "FRA", "RUS",
                                   "IND", "DEU", "JPN", "OTH"))

#  c. Plot the graph
gg1 = ggplot(place, aes(year, sum_lc, fill = iso_code)) +
  geom_area(alpha=0.6, colour="white") +
  scale_x_continuous(limits = c(1965,2020), breaks = seq(1965, 2020, by = 5)) +
  labs(title = "World electricity generation from LC sources",
       subtitle = "Grouped by country",
       x = "Year",
       y = "Terawatt-hours") +
  scale_fill_viridis_d(name = "Source", option = "E")

# d. Share of electricity production by country in 2020
place = filter(place, year == 2020)
place$sum_lc = round(place$sum_lc / sum(place$sum_lc),2)
place = place[,c(1,3)] %>% arrange(desc(sum_lc)) %>% as.data.frame()
colnames(place) = c("ISO code","% LC generation")

# Plot gg1 and table
gg1
```



```
kable(cbind(place[1:5,], place[6:10,]),
      caption = "Share of LC electricity generation by country, 2020")
```

Table 1: Share of LC electricity generation by country, 2020

ISO code	% LC generation	ISO code	% LC generation
OTH	0.30	FRA	0.05
CHN	0.25	IND	0.04
USA	0.16	RUS	0.04
BRA	0.05	DEU	0.03
CAN	0.05	JPN	0.02

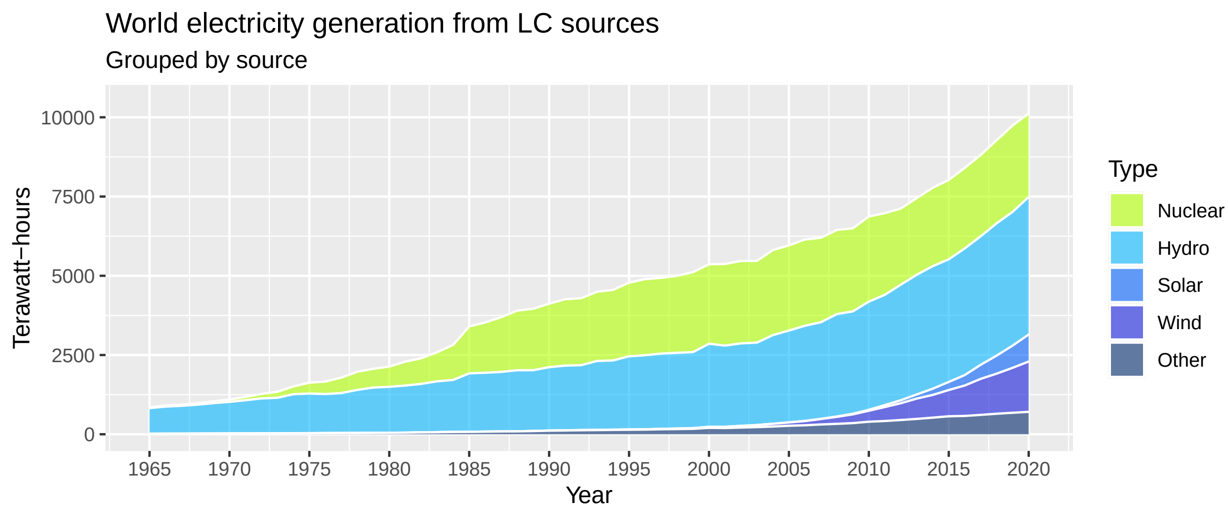
Unsurprisingly, the generation of electricity from LC sources is not homogeneous between the countries: China and the USA produce 41% of the world's LC electricity generated in 2020; the nine nations with the highest electricity generation from LC sources account for 69%.

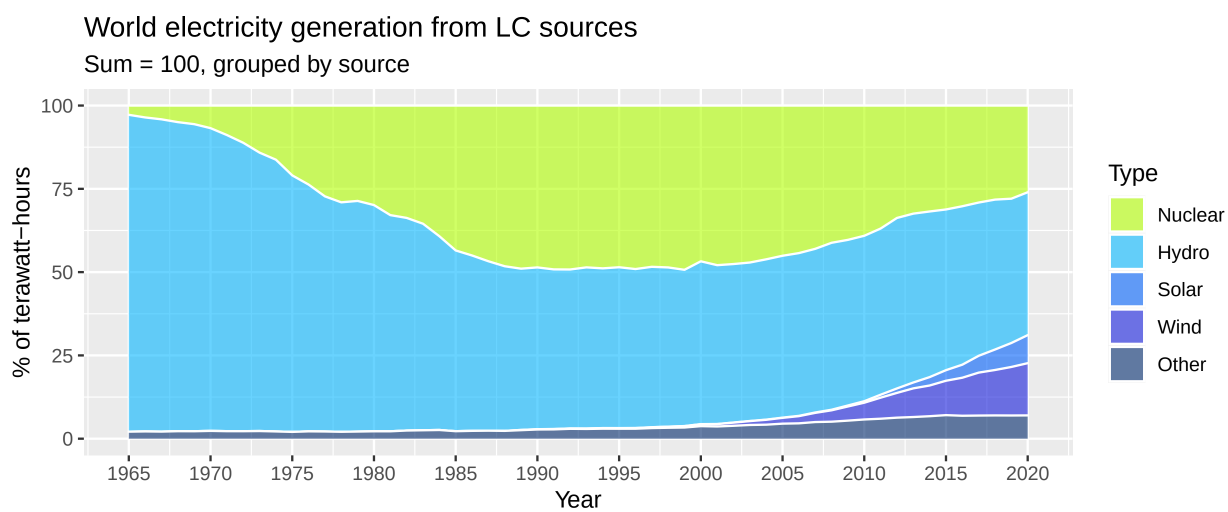
4.2 Analyses by source

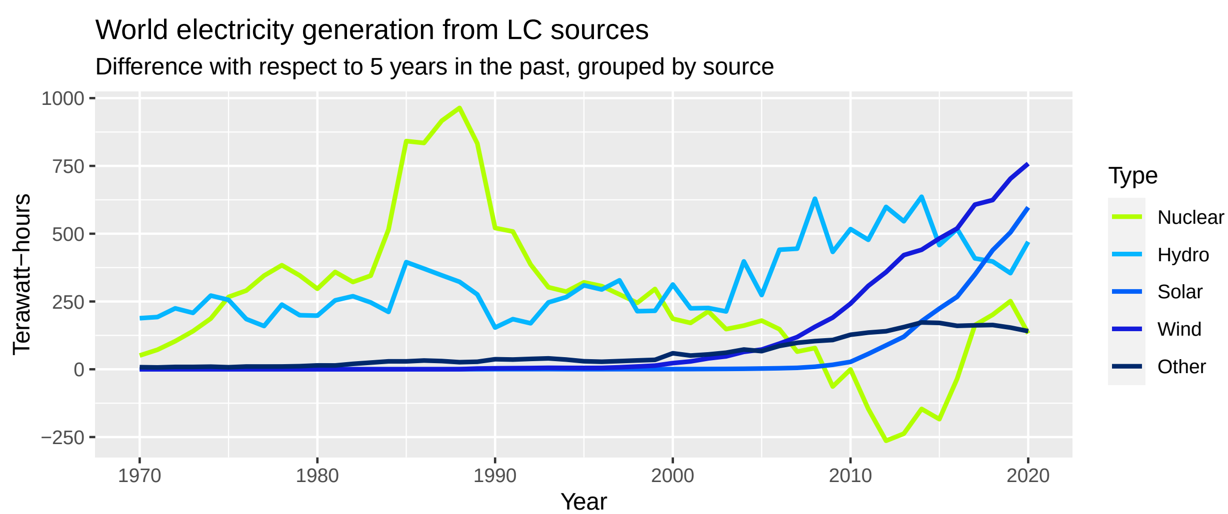
```
# 3. World electricity generation from LC sources, grouped by source

# a. Area plot of the total electricity generation
# Creation of the dataset
place = main[c("year", "hydro_electricity",
               "nuclear_electricity", "solar_electricity",
               "wind_electricity", "other_renewable_electricity")] %>%
  mutate_all(~replace_na(.,0)) %>%
  group_by(year) %>%
  summarize(Nuclear = sum(nuclear_electricity),
            Hydro = sum(hydro_electricity),
            Wind = sum(wind_electricity),
            Solar = sum(solar_electricity),
            Other = sum(other_renewable_electricity)) %>%
  gather(key = "Type",
         value = "elect",
         -year)
place$Type = factor(place$Type, levels = c("Nuclear", "Hydro", "Solar",
                                           "Wind", "Other"))

# Creation of the plot
gg1 = ggplot(place, aes(year, elect, fill = Type)) +
  geom_area(alpha=0.6 , size=.5, colour="white")+
  scale_x_continuous(limits = c(1965,2020), breaks = seq(1965, 2020, by = 5)) +
  scale_y_continuous(limits = c(0,10500)) +
  scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
  labs(title = "World electricity generation from LC sources",
       subtitle = "Grouped by source",
       x = "Year",
       y = "Terawatt-hours")

# b. Area plot of the total electricity generation, with sum = 100
```

```r
# Creation of the dataset
place = main[c("year","hydro_electricity", "wind_electricity",
               "nuclear_electricity", "solar_electricity",
               "other_renewable_electricity")] %>%
  mutate_all(~replace_na(.,0)) %>%
  group_by(year) %>%
  summarize(Nuclear = sum(nuclear_electricity),
            Hydro = sum(hydro_electricity),
            Wind = sum(wind_electricity),
            Solar = sum(solar_electricity),
            Other = sum(other_renewable_electricity)) %>%
  select(Nuclear, Hydro, Wind, Solar,
         Other, Nuclear) %>%
  mutate(year = 1900:2022, row_total = rowSums(.)) %>%
  mutate(across(Nuclear:Other, ~ . / row_total * 100)) %>%
  select(-row_total) %>%
  gather(key = "Type",
         value = "elect",
         -year)
place$Type = factor(place$Type, levels = c("Nuclear", "Hydro", "Solar",
                                           "Wind", "Other"))

# Creation of the plot
gg2 = ggplot(place, aes(year, elect, fill = Type)) +
  geom_area(alpha=0.6 , size=.5, colour="white")+
  scale_x_continuous(limits = c(1965,2020), breaks = seq(1965, 2020, by = 5)) +
    scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
      labs(title = "World electricity generation from LC sources",
       subtitle = "Sum = 100, grouped by source",
       x = "Year",
       y = "% of terawatt-hours")

# c. Line plot of the difference of generation with respect to 5 years in the past
# Creation of the dataset
place = main[c("year", "hydro_electricity", "nuclear_electricity",
               "solar_electricity", "wind_electricity",
               "other_renewable_electricity")] %>%
  mutate_all(~replace_na(.,0)) %>%
  filter(year <= 2020) %>%
  group_by(year) %>%
  summarize(Nuclear = sum(nuclear_electricity),
            Hydro = sum(hydro_electricity),
            Wind = sum(wind_electricity),
            Solar = sum(solar_electricity),
            Other = sum(other_renewable_electricity)) %>%
  mutate(Nuclear = (Nuclear - dplyr::lag(Nuclear,5)),
         Hydro = (Hydro - dplyr::lag(Hydro,5)),
         Wind = (Wind - dplyr::lag(Wind,5)),
         Solar = (Solar - dplyr::lag(Solar,5)),
         Other = (Other - dplyr::lag(Other,5))) %>%
  gather(key = "Type",
         value = "world_elect",
         -year)
```

```r
place$Type = factor(place$Type, levels = c("Nuclear", "Hydro", "Solar",
                                            "Wind", "Other"))

# Creation of the plot
gg3 = ggplot(place, aes(year, world_elect, colour = Type)) +
  geom_line(size = 1) +
  scale_x_continuous(limits = c(1970,2020)) +
  scale_y_continuous() +
  scale_color_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
    labs(title = "World electricity generation from LC sources",
       subtitle = "Difference with respect to 5 years in the past, grouped by source",
       x = "Year",
       y = "Terawatt-hours")

# Visualization of gg1, gg2 and gg3
grid.arrange(gg1, gg2, gg3)
```







The plots allow us to identify three different phases in the history of LC electricity.

1. **Dawn of nuclear electricity** (up to the mid-'80s). In this era, there are two different trends: on one side, the electricity generated by hydropower grows linearly with respect to the past generation; on the other side, the civil usage of nuclear power takes the first steps.

2. **Golden age of nuclear electricity** (from the mid-'80s to mid-'00s). The electricity generation from nuclear reaches its peak, while the production of hydropower plants continues to grow linearly.

3. **Golden age of renewables** (from the mid-'00s to nowadays). Nuclear power generation declines and gives way to renewables. In particular, solar and wind generations skyrocket.

4.3 Analyses by macroregion

We are now interested in analyzing electricity generation by LC sources in different areas. To do so, we aggregate world countries into six macroregions based on geographical, economic, and cultural factors.

1. **Developed countries**: Western Europe, Israel, USA, Canada, Australia, New Zealand, Japan, South Korea, Taiwan, Hong Kong, and Macao.

2. **Latin America and the Caribbean**: North and South America's countries, except for USA and Canada.

3. **Eastern Europe**: former members of the Warsaw Pact (excluding Kazakhstan, Turkmenistan, Uzbekistan, Tajikistan, and Kyrgyzstan) and former Yugoslavia.

4. **Middle East and Northern Africa**: Morocco, Algeria, Tunisia, Libya, Egypt, Jordan, Palestine, Lebanon, Syria, Turkey, Iraq, Iran, Kuwait, Saudi Arabia, Yemen, Oman, United Arab Emirates, Bahrain, and Qatar.

5. **Sub-Saharan Africa**: non-aforementioned African countries.

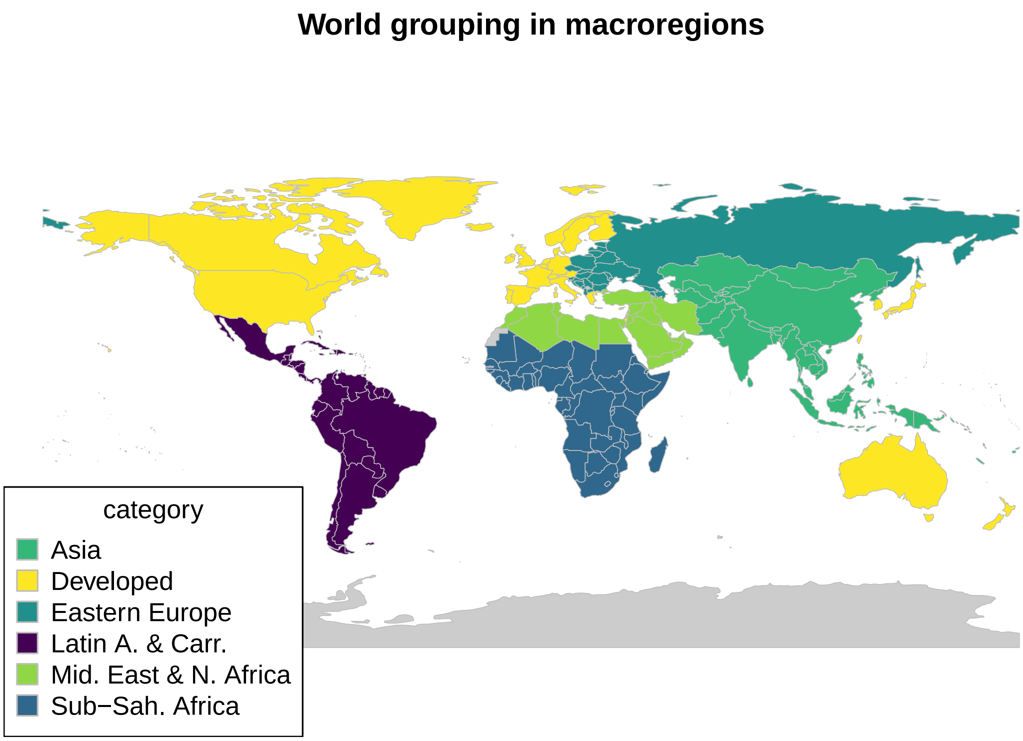
6. **Asia**: non-aforementioned Asian countries.

```
# a. Creation of a vector for each macroregion, containing the ISO-codes
developed_countries = c("AUS", "AUT", "BEL", "CAN", "CYP", "DNK", "FRO", "FIN",
                        "FRA", "DEU", "GRC", "GRL", "HKG", "ISL", "IRL", "ISR",
                        "ITA", "JPN", "LUX", "MAC", "MLT", "NLD", "NZL", "NOR",
                        "PRT", "SPM", "KOR", "ESP", "SWE", "CHE", "TWN", "GBR",
                        "USA", "REU", "GIB")

latin_countries = c("ATG", "ARG", "ABW", "BHS", "BRB", "BLZ", "BMU", "BOL",
                    "BRA", "CYM", "CHL", "COL", "CRI", "CUB", "DMA", "DOM",
                    "ECU", "SLV", "FLK", "GUF", "GRD", "GLP", "GTM", "GUY",
                    "HTI", "HND", "JAM", "MTQ", "MEX", "MSR", "NIC", "PAN",
                    "PRY", "PER", "PRI", "KNA", "LCA", "VCT", "SUR", "TTO",
                    "TCA", "VIR", "URY", "VEN", "VGB", "ANT")

east_europe_countries = c("ALB", "ARM", "AZE", "BLR", "BIH", "BGR", "HRV",
                          "CZE", "EST", "GEO", "HUN", "LVA", "LTU", "MDA",
                          "MNE", "MKD", "POL", "ROU", "RUS", "SRB", "SVK",
                          "SVN", "UKR")

sub_african_countries = c("AGO", "BEN", "BWA", "BFA", "BDI", "CPV", "CMR",
                          "CAF", "TCD", "COM", "COG", "CIV", "COD", "DJI",
                          "GNQ", "ERI", "SWZ", "ETH", "GAB", "GMB", "GHA",
                          "GIN", "GNB", "KEN", "LSO", "LBR", "MDG", "MWI",
                          "MLI", "MRT", "MUS", "MOZ", "NAM", "NER", "NGA",
                          "RWA", "STP", "SEN", "SLE", "SOM", "ZAF", "SSD",
                          "SDN", "TZA", "TGO", "UGA", "ZMB", "ZWE", "SHN")
```

```r
middle_east_countries = c("DZA", "BHR", "EGY", "IRN", "IRQ", "JOR", "KWT",
                          "LBN", "LBY", "MAR", "OMN", "PSE", "QAT", "SAU",
                          "SYR", "TUR", "ARE", "YEM", "TUN")

asian_countries = c("AFG", "ASM", "BGD", "BTN", "BRN", "KHM", "CHN", "COK",
                    "FJI", "PYF", "GUM", "IND", "IDN", "KAZ", "KIR", "KGZ",
                    "LAO", "MYS", "MDV", "FSM", "MNG", "MMR", "NRU", "NPL",
                    "NCL", "PRK", "MNP", "PAK", "PNG", "PHL", "WSM", "VNM",
                    "SYC", "SGP", "SLB", "LKA", "TJK", "THA", "TLS", "TON",
                    "TKM", "TUV", "UZB", "VUT", "NIU")

#b. Assign the grouping to each observation in "main"
tag = rep(0, nrow(main))

for(i in 1:length(tag)){
  if(main$iso_code[i] %in% developed_countries){
    tag[i] = "developed"
  }
  else if(main$iso_code[i] %in% latin_countries){
    tag[i] = "latin"
  }
  else if(main$iso_code[i] %in% east_europe_countries){
    tag[i] = "east_europe"
  }
  else if(main$iso_code[i] %in% sub_african_countries){
    tag[i] = "sub_african"
  }
  else if(main$iso_code[i] %in% middle_east_countries){
    tag[i] = "middle_east"
  }
  else if(main$iso_code[i] %in% asian_countries){
    tag[i] = "asian"
  }
}
main = cbind(main, tag)

# c.Plot of the world map
df_asian = data.frame(region = "Asia", tag = asian_countries)
df_east = data.frame(region = "Eastern Europe", tag = east_europe_countries)
df_middle = data.frame(region = "Mid. East & N. Africa", tag = middle_east_countries)
df_dev = data.frame(region = "Developed", tag = developed_countries)
df_africa = data.frame(region ="Sub-Sah. Africa", tag = sub_african_countries)
df_latin = data.frame(region = "Latin A. & Carr.", tag = latin_countries)
df_world = rbind(df_asian, df_east, df_middle, df_dev, df_africa, df_latin)


map = joinCountryData2Map(df_world, joinCode = "ISO3",
                          nameJoinColumn = "tag")
```

```
## 212 codes from your data successfully matched countries in the map
## 5 codes from your data failed to match with a country code in the map
## 31 codes from the map weren't represented in your data
```

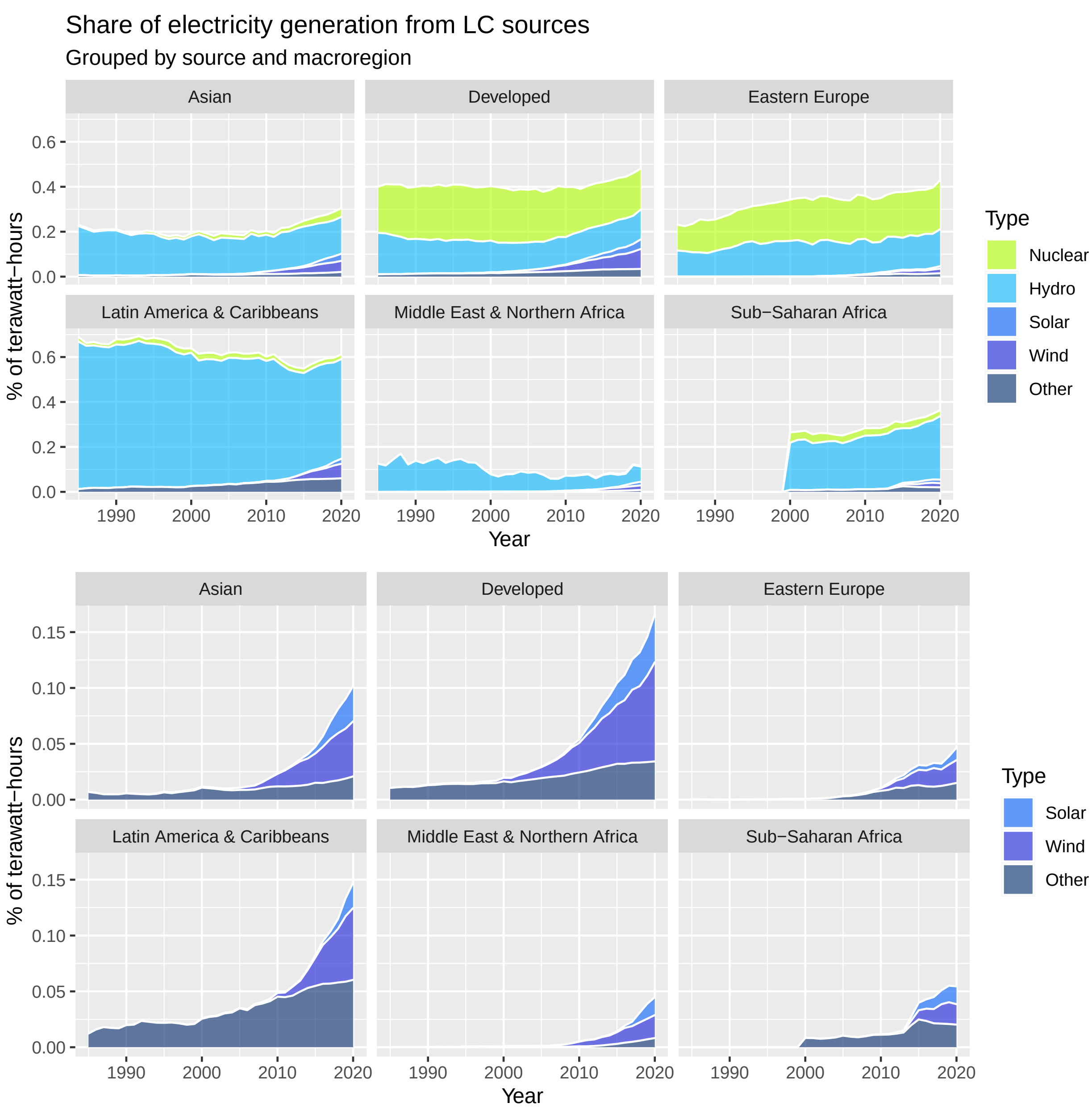
```
mapCountryData(map, nameColumnToPlot = "region", catMethod = "categorical",
               missingCountryCol = gray(.8),
               colourPalette = c("#35B779", "#FDE725", "#21908C", "#440154",
                                 "#8FD744", "#30678D"),
               mapTitle = "World grouping in macroregions")
```




```
# 3. World electricity generation from LC sources, grouped by macroregion

# a. LC generation by macroregion
# Creation of the dataset
place = main[c("year","hydro_electricity", "wind_electricity",
               "nuclear_electricity", "solar_electricity",
               "other_renewable_electricity", "electricity_generation")] %>%
  mutate_all(~replace_na(.,0)) %>%
  cbind(., tag = main$tag) %>%
  group_by(year, tag) %>%
  summarize(Nuclear = sum(nuclear_electricity)/sum(electricity_generation),
            Hydro = sum(hydro_electricity)/sum(electricity_generation),
            Wind = sum(wind_electricity)/sum(electricity_generation),
            Solar = sum(solar_electricity)/sum(electricity_generation),
            Other = sum(other_renewable_electricity)/sum(electricity_generation)) %>%
  gather(key = "Type",
         value = "elect",
```

```r
         -year, -tag)
place$Type = factor(place$Type, levels = c("Nuclear", "Hydro", "Solar",
                                            "Wind", "Other"))
# Here we remove data for Sub-Saharan countries from 2000, as data is not available
# for most of the countries
for(i in 1:nrow(place)){
  if(place$year[i] < 2000 & place$tag[i] == "sub_african"){
    place$elect[i] = 0
  }
}

# This functions change the labels of the macroregions and of the low-carbon
# sources respectively, in order to obtain neater plots
tag_modifier = function(place_data){
  place_data[place_data$tag  == "asian", "tag"] = "Asian"
  place_data[place_data$tag  == "developed", "tag"] = "Developed"
  place_data[place_data$tag  == "east_europe", "tag"] = "Eastern Europe"
  place_data[place_data$tag  == "latin", "tag"] = "Latin America & Caribbeans"
  place_data[place_data$tag  == "middle_east", "tag"] = "Middle East & Northern Africa"
  place_data[place_data$tag  == "sub_african", "tag"] = "Sub-Saharan Africa"
  return(place_data)
}

source_modifier = function(place_data){
  place_data[place_data$Source  == "nuclear_share_elec", "Source"] = "Nuclear"
  place_data[place_data$Source  == "hydro_share_elec", "Source"] = "Hydro"
  place_data[place_data$Source  == "solar_share_elec", "Source"] = "Solar"
  place_data[place_data$Source  == "wind_share_elec", "Source"] = "Wind"
  place_data[place_data$Source  == "other_renewables_share_elec", "Source"] = "Other"

  place_data$Source = factor(place_data$Source, levels = c("Nuclear", "Hydro", "Solar",
                                                            "Wind", "Other"))
  return(place_data)
}

place = tag_modifier(place)


# Creation of the plot
gg1 = ggplot(place, aes(year, elect, fill = Type)) +
  geom_area(alpha=0.6 , size=.5, colour="white")+
  scale_x_continuous(limits = c(1985,2020)) +
  scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
  facet_wrap(~ tag, nrow = 2) +
  labs(title = "Share of electricity generation from LC sources",
       subtitle = "Grouped by source and macroregion",
       x = "Year",
       y = "% of terawatt-hours")

# b. Plot of renewables generation by macroregion, excluding hydropower
# Creation of the dataset
place = filter(place, Type == "Solar" | Type == "Wind" | Type == "Other")
```

```r
# Creation of the plot
gg2 = ggplot(place, aes(year, elect, fill = Type)) +
  geom_area(alpha=0.6 , size=.5, colour="white")+
  scale_x_continuous(limits = c(1985,2020)) +
  scale_fill_manual(values = c("#0060FA", "#141BDB", "#00296B")) +
  facet_wrap(~ tag, nrow = 2) +
  labs(x = "Year",
       y = "% of terawatt-hours")

# Visualization of gg1, gg2 and gg3
grid.arrange(gg1, gg2, ncol=1)
```



A variety of conclusions can be drawn from the previous graphs. Here we present the four main findings.

1. The different areas are **not homogeneous** in electricity generation from LC sources: Latin American countries have a more significant share; Developed, Eastern European and Sub-Saharan and Asian follow; Middle-East generation is negligible.

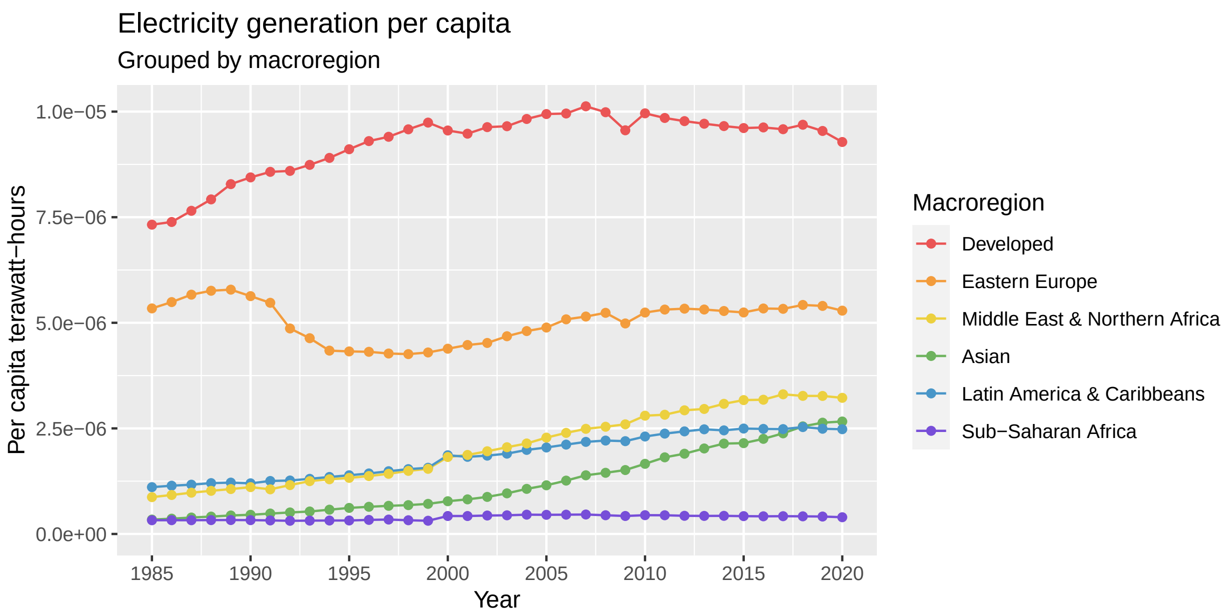
2. **Hydropower** is an essential source of electricity in all the considered macroregions.

3. **Nuclear electricity** is significant only in Developed countries and Eastern Europe; its importance is comparable to hydropower generation in those macroregions.

4. Developed countries drive **non-hydropower renewable** production. Nonetheless, it is also true that those sources are also rapidly becoming more relevant in Latin America & Caribbeans and Asia.

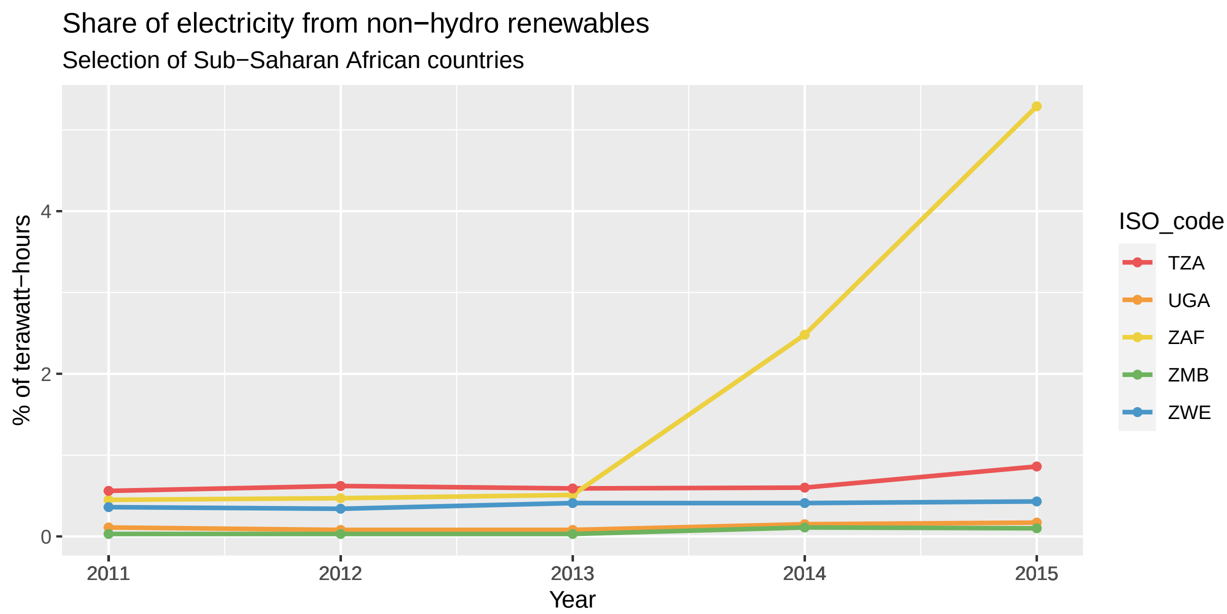
It is important to highlight also that a lower total electricity generation heavily influences Asian, Latin American, and (especially) Sub-Saharan generation rates, as the graph below shows.

```r
# Focus on electricity generation per capita
# Creation of the dataset
place = main[c("year","electricity_generation", "population")] %>%
  mutate_all(~replace_na(.,0)) %>%
  cbind(., tag = main$tag) %>%
  group_by(tag,year) %>%
  summarize(gen_per_capita = sum(electricity_generation)/sum(population))

place = tag_modifier(place)
place$tag = factor(place$tag, levels = c("Developed", "Eastern Europe",
                                         "Middle East & Northern Africa",
                                         "Asian", "Latin America & Caribbeans",
                                         "Sub-Saharan Africa"))
colnames(place) = c("Macroregion","year","gen_per_capita")


# Creation of the plot
ggplot(place, aes(year, gen_per_capita, color = Macroregion)) +
  geom_line() +
  geom_point() +
  scale_x_continuous(limits = c(1985,2020), breaks = seq(1985, 2020, by = 5)) +
  scale_color_manual(values = c("#EA5555", "#F39C3C", "#ECD03F", "#6EB35E", "#4996C8",
                                "#774ED8")) +
  labs(title = "Electricity generation per capita",
       subtitle = "Grouped by macroregion",
       x = "Year",
       y = "Per capita terawatt-hours")
```



The electricity generation from non-hydro renewable sources in Sub-Saharan Africa increases rapidly between 2011 and 2015. Therefore, we studied the behavior of the five countries in the region with the highest non-hydro renewable electricity generation in 2015. As the plot shows, the steep increase is simply due to an exploding generation from solar and wind sources in South Africa.

```
# Extract the LC production grouped by source from 2011 to 2015 of the five countries
# with the highest renewables production in 2015

#a. Extract the ISO of five countries with the highest non-hydro renewables generation
#   in 2015
place = select(main, iso_code, year, renewables_electricity, hydro_electricity) %>%
  filter(year == 2015 & tag == "sub_african") %>%
  mutate(non_hydro_elec = renewables_electricity - hydro_electricity) %>%
  arrange(desc(non_hydro_elec)) %>%
  select(iso_code) %>%
  top_n(5) %>%
  as.data.frame()

# b. Creation of the dataset
place = select(main, year, iso_code, renewables_electricity, hydro_electricity) %>%
  filter(year <= 2015 & year >= 2011 & iso_code %in% place$iso_code) %>%
  mutate(non_hydro_elec = renewables_electricity - hydro_electricity)
colnames(place) = c(colnames(place)[1], "ISO_code", colnames(place)[3:5])

# c. Creation the plot
ggplot(place, aes(year, non_hydro_elec, color = ISO_code)) +
  geom_line(size = 1) +
  geom_point(size = 1.5) +
  scale_x_continuous(limits = c(2011,2015), breaks = place$year) +
  scale_color_manual(values = c("#EA5555", "#F39C3C", "#ECD03F", "#6EB35E", "#4996C8")) +
    labs(title = "Share of electricity from non-hydro renewables",
       subtitle = "Selection of Sub-Saharan African countries",
       x = "Year",
       y = "% of terawatt-hours")
```



4.4 Green Score with focus on the sources

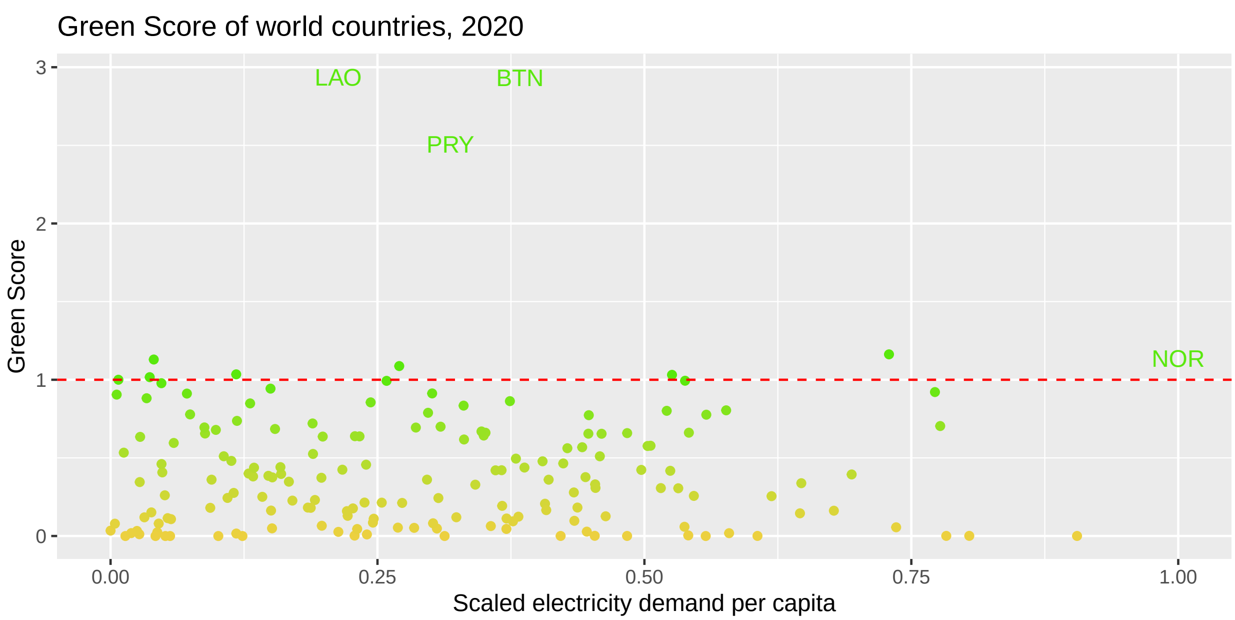
In this section, we aim to study which countries are nearer to the full LC target (i.e., to produce from LC sources all the electricity they consume). To do so, we create a **Green Score**, defined as the following ratio:

$$\mathrm{GS} = \frac{\text{electricity generated from LC sources}}{\text{electricity demand}}$$

Note that in paragraphs 4.4 and 4.5 analyses, we removed the countries with a smaller population than 500,000. The reason is that we expect those countries to be too small to have a significant independent electricity policy with respect to their neighbors.

```
# Note. x is computed as the squared root of the electricity demand per capita, scaled in
# the interval [0,1]. We plot the electricity demand per capita to distinguish if the
# high Green Score is due to low electricity consumption or good green policies. We
# choose the described scaling because it allows for spacing more points in the graph.

# Scatterplot of the Green Score in 2020 in each country
# Creation of the dataset
place = filter(main, (year == 2020 & population > 500000 & iso_code != "REU")) %>%
  transform(elec_demand_per_capita = (
    sqrt(electricity_demand / population) -
      min(sqrt(electricity_demand / population))) /
      (max(sqrt(electricity_demand / population)) -
         min(sqrt(electricity_demand / population))),
    green_score_renew = renewables_electricity / electricity_demand,
    green_score_lc = low_carbon_electricity / electricity_demand) %>%
  select(iso_code, elec_demand_per_capita, green_score_renew, green_score_lc)

# Creation of the plot
ggplot()+
  geom_point(data = filter(place, iso_code != "LAO", iso_code != "BTN",
                           iso_code != "PRY", iso_code != "NOR"),
```

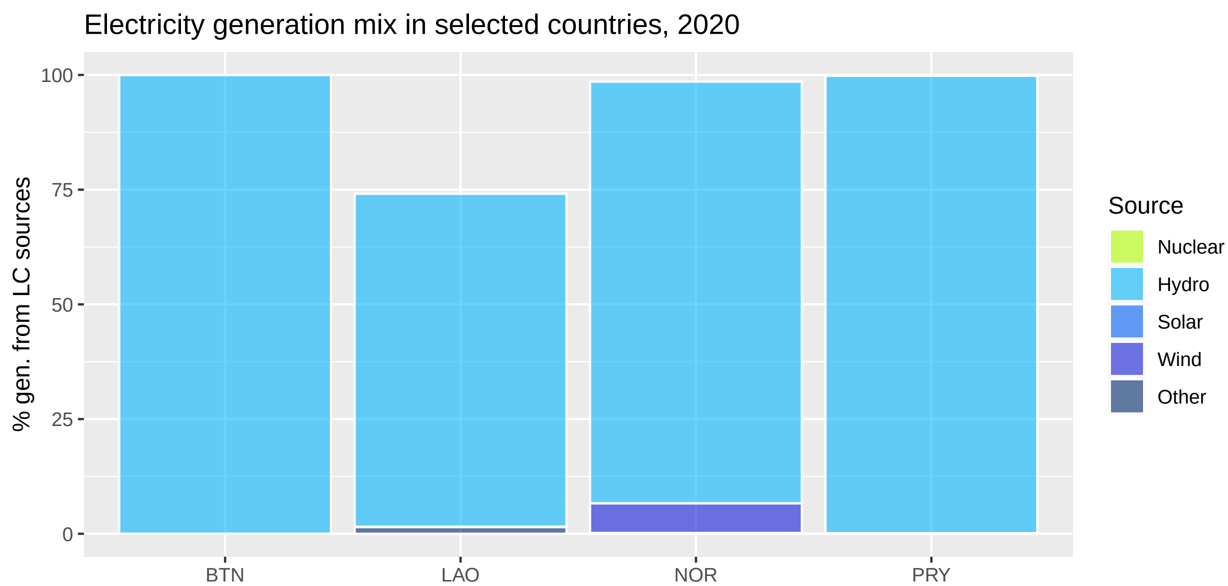
```
           mapping = aes(elec_demand_per_capita, green_score_lc,
                         color = green_score_lc)) +
  geom_text(data = filter(place, iso_code == "LAO" | iso_code == "BTN" |
                          iso_code == "PRY" | iso_code == "NOR"),
          mapping = aes(elec_demand_per_capita, green_score_lc,
                        label = iso_code, color = green_score_lc)) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  scale_color_gradient(low = "#ECD03F", high = "#59E80C", na.value = "#59E80C",
                       limits = c(0,1), guide = "none") +
  labs(title = "Green Score of world countries, 2020",
       x = "Scaled electricity demand per capita",
       y = "Green Score")
```



The plot highlights **four main outliers**: Laos, Bhutan, and Paraguay, with an overscaled Green Score and a low electricity demand per capita; Norway, with a positive Green Score while being the country with the highest electricity demand per capita. We want to understand if they have shared features that allow us to explain their outperformance.

```
# Barplot of the electricity generation mix in LAO, BTN, PRY and NOR
# Creation of the dataset
place = filter(main, (year == 2020 & (iso_code == "LAO" | iso_code == "BTN" |
                                        iso_code == "PRY" | iso_code == "NOR"))) %>%
  select(iso_code, solar_share_elec, wind_share_elec, hydro_share_elec,
         nuclear_share_elec, other_renewables_share_elec) %>%
  gather(key = "Source", value = "value", -iso_code)

# The following function modify the names of the sources to enchance the visualization
source_modifier = function(place_data){
  place_data[place_data$Source  == "nuclear_share_elec", "Source"] = "Nuclear"
  place_data[place_data$Source  == "hydro_share_elec", "Source"] = "Hydro"
  place_data[place_data$Source  == "solar_share_elec", "Source"] = "Solar"
  place_data[place_data$Source  == "wind_share_elec", "Source"] = "Wind"
  place_data[place_data$Source  == "other_renewables_share_elec", "Source"] = "Other"
```

```r
  place_data$Source = factor(place_data$Source, levels = c("Nuclear", "Hydro", "Solar", "Wind",
                                                           "Other"))
  return(place_data)
}

place = source_modifier(place)


# Creation of the plot
ggplot(place, aes(x = iso_code, y = value, fill = Source)) +
  geom_bar(position = "stack", stat = "identity", alpha = 0.6, colour = "white") +
  scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
  labs(title = "Electricity generation mix in selected countries, 2020",
       x = "",
       y = "% gen. from LC sources")
```



The high performance of the countries is due to a **dominant hydroelectric generation**. For example, further research on the topic highlights that Laos aims to become the "Battery of Southeast Asia" by further exploiting its impressive hydropower potential [13]. So those countries can achieve such an impressive result because of a resource not available everywhere.

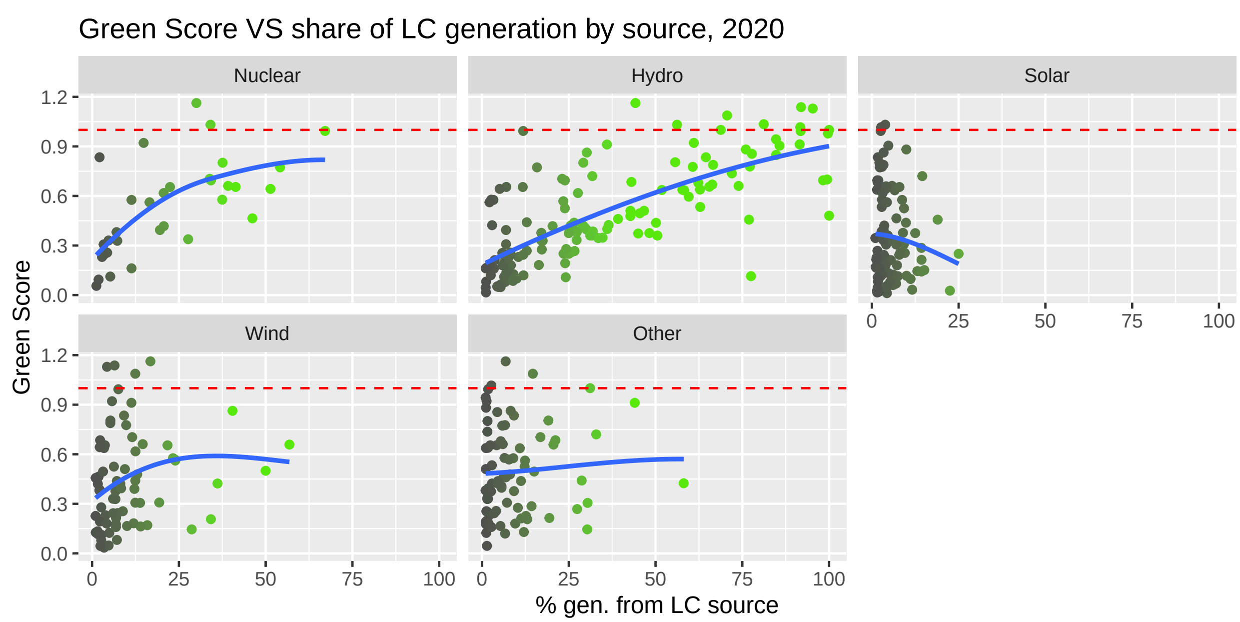
Nonetheless, some countries have great hydropower generation but do not exploit it: for instance, the Democratic Republic of Congo has a potential of 100.000 MW (more than four times the biggest hydropower plant in the world, the Three Gorges Dam in China [14]), but uses only 2.5% of it due to political instability and lack of investments [15].

We then analyze further the correlation between each LC source and the Green Score.

```r
# Scatterplot of the green Score VS share of electricity produced from each LC source
# Creation of the dataset
place = filter(main, year == 2020) %>%
  transform(green_score_lc = (low_carbon_electricity / electricity_demand)) %>%
  select(iso_code, green_score_lc, nuclear_share_elec, hydro_share_elec,
         solar_share_elec, wind_share_elec, other_renewables_share_elec,
         green_score_lc) %>%
```

```
  filter(complete.cases(.)) %>%
  gather(key = "Source", value = "value", -iso_code, -green_score_lc) %>%
  # We excluded countries with an irrelevant production from each source
  filter(value > 1)

place = source_modifier(place)

# Creation of the plot
ggplot(filter(place, iso_code != "LAO", iso_code != "BTN", iso_code != "PRY"),
       aes(value, green_score_lc, color = value)) +
  geom_point() +
  geom_smooth(method = "loess", span = 2, se = FALSE, size = 1) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  scale_color_gradient(low = "grey30", high = "#59E80C", na.value = "#59E80C",
                       limits = c(0,40),
                       guide = "none") +
  facet_wrap(~Source, nrow = 2) +
  labs(title = "Green Score VS share of LC generation by source, 2020",
       x = "% gen. from LC source",
       y = "Green Score")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



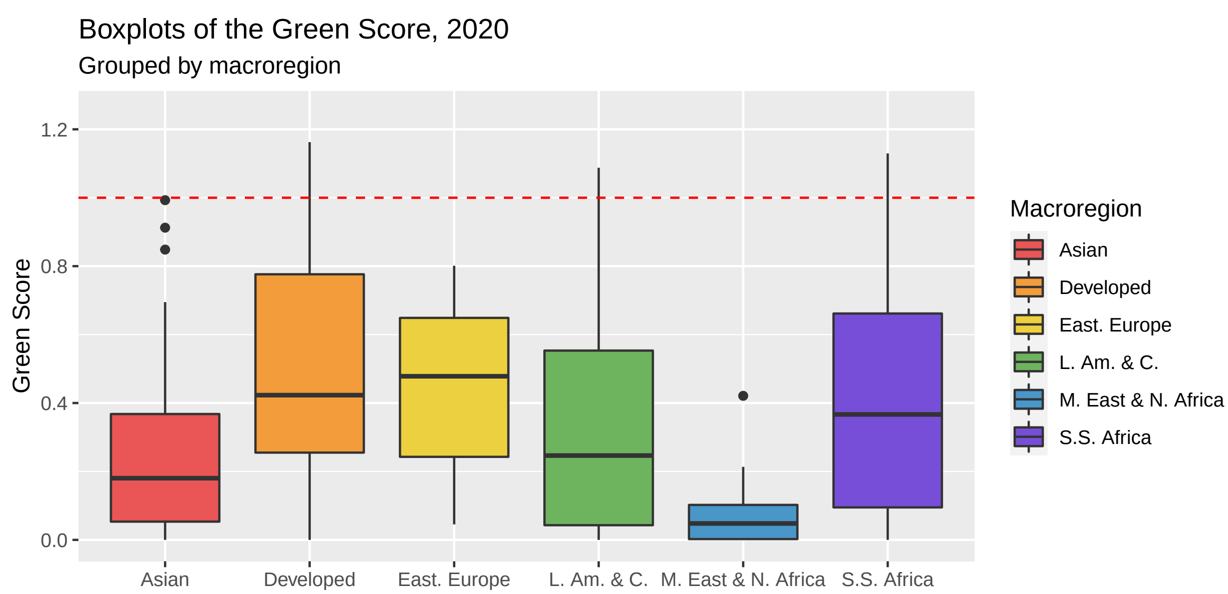

```
# Note. "loess" is a statistical technique used for estimating smooth curves in
# scatterplot data. It works by fitting multiple local regression models
# to different subsets of the data, allowing it to capture non-linear patterns and
# relationships between variables. It was introduced in the following plot only to
# highlight better the trends from a graphical point of view.
```

The plot confirms the correlation between the Green Score and the electricity generation from hydropower. It also shows an important link with nuclear power production but not with solar, wind, and other sources.

4.5 Green Score with focus on the macroregions

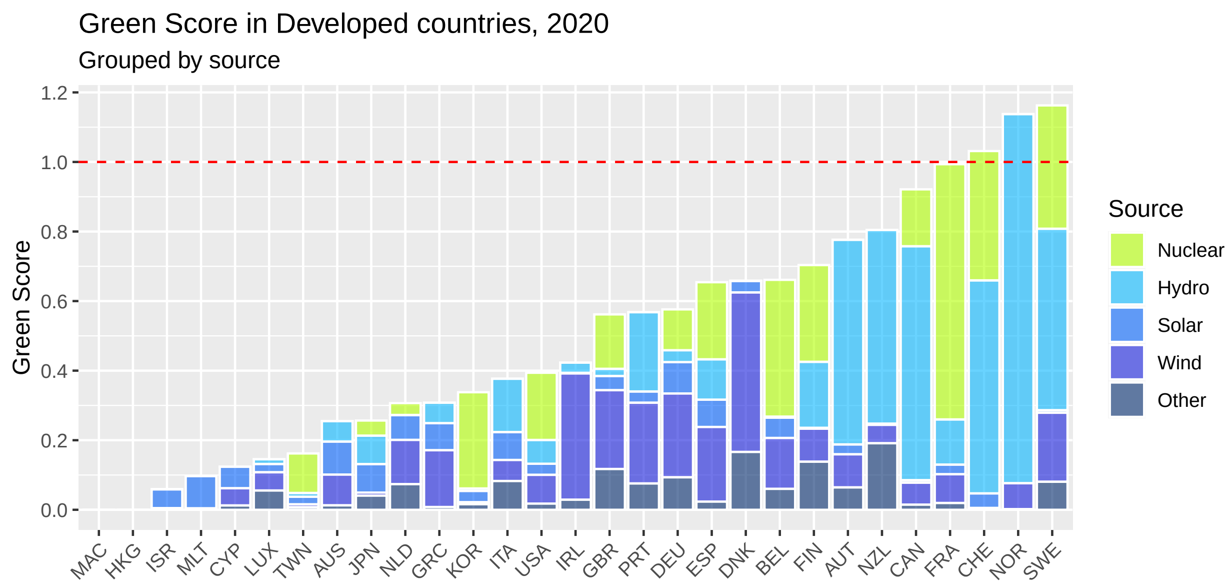
In this final section of descriptive analyses, we focus on the analysis of the Green Score in each country.

```r
# Boxplot of the Green Score
# Creation of the dataset
place = filter(main, year == 2020) %>%
  transform(green_score_lc = (low_carbon_electricity / electricity_demand)) %>%
  select(iso_code, tag, green_score_lc)

place[place$tag  == "asian", "tag"] = "Asian"
place[place$tag  == "developed", "tag"] = "Developed"
place[place$tag  == "east_europe", "tag"] = "East. Europe"
place[place$tag  == "latin", "tag"] = "L. Am. & C."
place[place$tag  == "middle_east", "tag"] = "M. East & N. Africa"
place[place$tag  == "sub_african", "tag"] = "S.S. Africa"

colnames(place) = c("iso_code", "Macroregion", "green_score_lc")

# Creation of the plot
ggplot(place, aes(x = Macroregion, y = green_score_lc)) +
  geom_boxplot(aes(fill = Macroregion)) +
  ylim(0,1.25) +
  scale_fill_manual(values = c("#EA5555", "#F39C3C", "#ECD03F", "#6EB35E", "#4996C8",
                                "#774ED8")) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
    labs(title = "Boxplots of the Green Score, 2020",
         subtitle = "Grouped by macroregion",
       x = "",
       y = "Green Score")
```



The leading macroregion by Green Score comprises the **Developed countries**, followed by Eastern Europe, Sub-Saharian Africa, Asia, and Latin America & Caribbeans. The Middle East & Northern Africa has significantly lower score compared to other areas. Let us now explore each area separately.

```
# Barplot of the Green Score
# Creation of the dataset
place = filter(main, year == 2020, !is.na(other_renewables_share_elec),
               population >= 500000) %>%
  transform(ratio = (low_carbon_electricity / (electricity_demand * low_carbon_share_elec))) %>%
  transform(solar_share_elec = ratio * solar_share_elec,
            wind_share_elec = ratio * wind_share_elec,
            hydro_share_elec = ratio * hydro_share_elec,
            nuclear_share_elec = ratio * nuclear_share_elec,
            other_renewables_share_elec = ratio * other_renewables_share_elec) %>%
  select(iso_code, tag, solar_share_elec, wind_share_elec,
         hydro_share_elec, nuclear_share_elec, other_renewables_share_elec) %>%
  # There are NaN values obtain because of division by zero. We want them to be 0
  mutate(across(where(is.numeric), ~ ifelse(is.nan(.), 0, .))) %>%
  # There are NA values. We want to remove them
  gather(key = "Source", value = "value", -iso_code, -tag)

place = source_modifier(place)

# Creation of the plot for the Developed countries
ggplot(filter(place, tag == "developed"),
       aes(x = reorder(iso_code, value), y = value, fill = Source)) +
  geom_bar(position = "stack", stat = "identity", alpha = 0.6, colour = "white") +
  scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
  scale_y_continuous(breaks = seq(0, 1.2, by = 0.2)) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(title = "Green Score in Developed countries, 2020",
       subtitle = "Grouped by source",
       x = "",
       y = "Green Score") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

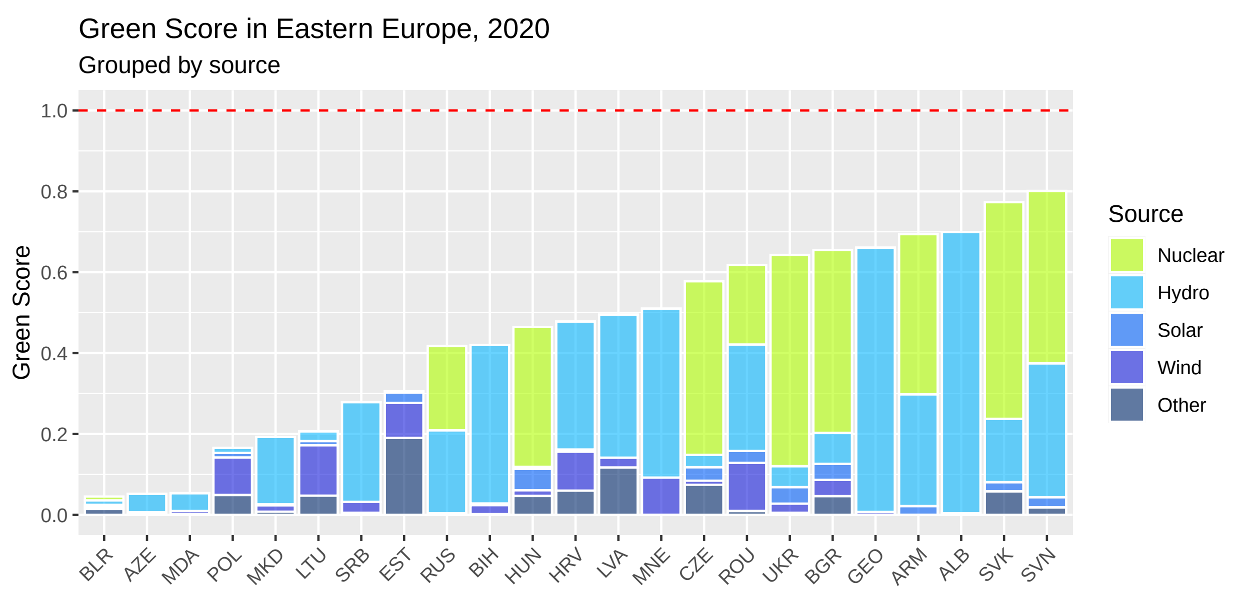


The Developed countries with the highest Green Scores are **Sweden**, Norway, Switzerland, and France. In contrast, the ones with the lowest values are Macao, Hong Kong, Israel, and Malta.

The LC sources are heterogeneous: while countries like Norway, Switzerland, and Canada are mainly driven by hydropower, others like France and Belgium mainly generate electricity from nuclear power, and others still, like Denmark and Ireland, are mainly driven by non-hydro renewables.

It is also interesting to highlight that the Developed Asian countries tend to have a lower score than the others: Japan is the second-best-performing Asian country, but it only outperforms Australia, Luxembourg, Cyprus, and Malta.

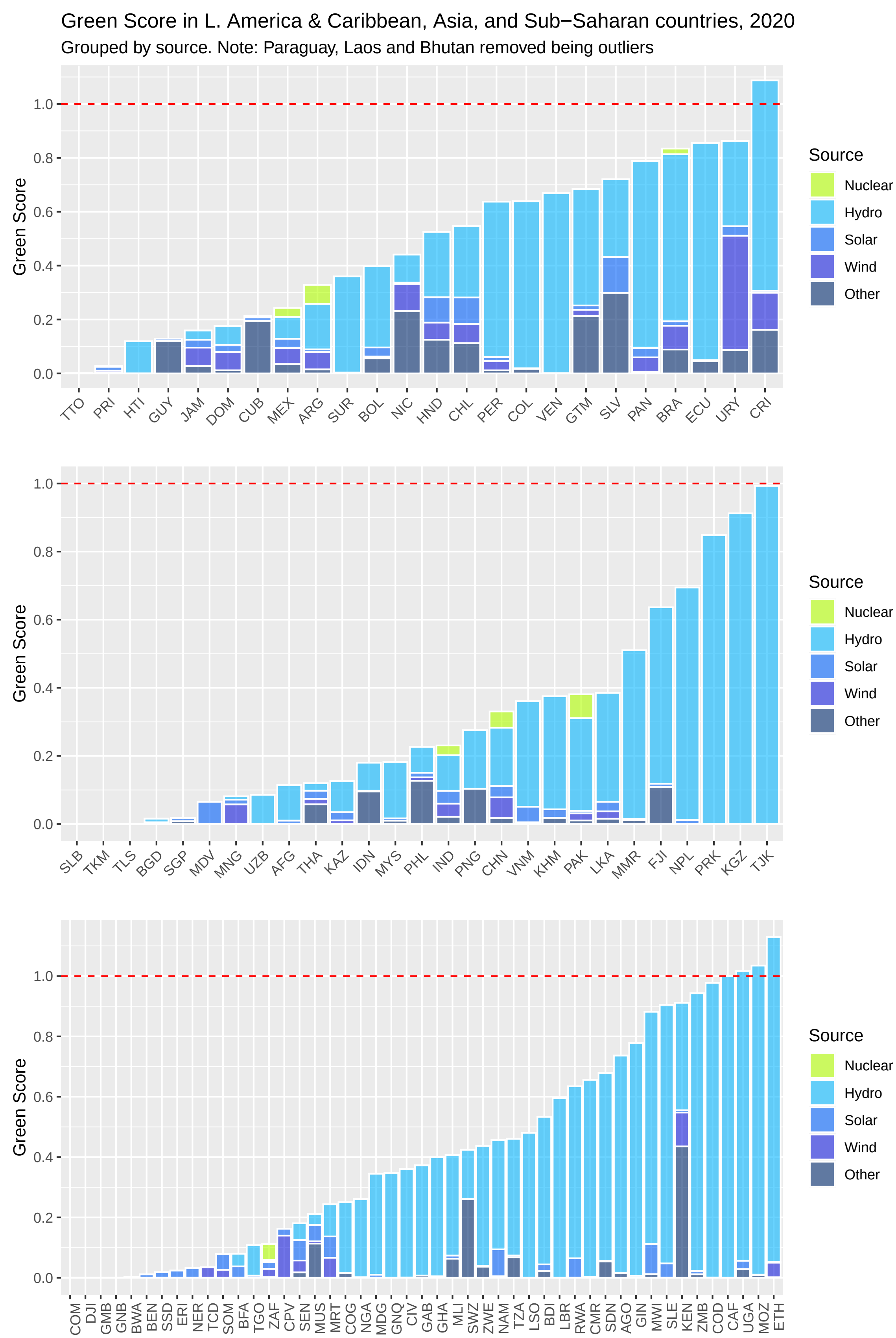
```r
# Creation of the plot for Eastern Europe
ggplot(filter(place, tag == "east_europe"),
       aes(x = reorder(iso_code, value), y = value, fill = Source)) +
  geom_bar(position = "stack", stat = "identity", alpha = 0.6, colour = "white") +
  scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
    scale_y_continuous(breaks = seq(0, 1, by = 0.2)) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(title = "Green Score in Eastern Europe, 2020",
       subtitle = "Grouped by source",
       x = "",
       y = "Green Score") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



No country in Eastern Europe is near a Green Score equal to 1: the best-performing one is **Slovenia**, with a score of 0.72. Follow Slovakia, Albania, and Armenia. There are also three countries with a score near zero in the region: Belarus, Azerbaijan, and Moldova. As the barplot shows, Eastern European countries mainly produce LC electricity through hydro and nuclear sources.

```r
# Creation of the plot for Latin America & Caribbean
gg1 = ggplot(filter(place, tag == "latin", iso_code != "PRY"),
       aes(x = reorder(iso_code, value), y = value, fill = Source)) +
  geom_bar(position = "stack", stat = "identity", alpha = 0.6, colour = "white") +
  scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
    scale_y_continuous(breaks = seq(0, 1.2, by = 0.2)) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(title = "Green Score in L. America & Caribbean, Asia, and Sub-Saharan countries, 2020",
       subtitle = "Grouped by source. Note: Paraguay, Laos and Bhutan removed being outliers",
```

```r
       x = "",
       y = "Green Score") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Creation of the plot for the Asian countries
gg2 = ggplot(filter(place, tag == "asian", iso_code != "LAO", iso_code != "BTN"),
       aes(x = reorder(iso_code, value), y = value, fill = Source)) +
  geom_bar(position = "stack", stat = "identity", alpha = 0.6, colour = "white") +
  scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
    scale_y_continuous(breaks = seq(0, 1, by = 0.2)) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(x = "",
       y = "Green Score") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# creation of the plot for the Sub-Saharan Africa
gg3 = ggplot(filter(place, tag == "sub_african"),
       aes(x = reorder(iso_code, value), y = value, fill = Source)) +
  geom_bar(position = "stack", stat = "identity", alpha = 0.6, colour = "white") +
  scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
    scale_y_continuous(breaks = seq(0, 1.2, by = 0.2)) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(x = "",
       y = "Green Score") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# Visualization of the plots
grid.arrange(gg1, gg2, gg3, ncol=1)
```


Green Score in L. America & Caribbean, Asia, and Sub−Saharan countries, 2020
Grouped by source. Note: Paraguay, Laos and Bhutan removed being outliers
Green Score
0.0
0.2
0.4
0.6
0.8
1.0
TTO PRI HTI GUY JAM DOM CUB MEX ARG SUR BOL NIC HND CHL PER COL VEN GTM SLV PAN BRA ECU URY CRI
SLB TKM TLS BGD SGP MDV MNG UZB AFG THA KAZ IDN MYS PHL IND PNG CHN VNM KHM PAK LKA MMR FJI NPL PRK KGZ TJK
COM DJI GMB GNB BWA BEN SSD ERI NER TCD SOM BFA TGO ZAF CPV SEN MUS MRT COG NGA MDG GNQ CIV GAB GHA MLI SWZ ZWE NAM TZA LSO BDI LBR RWA CMR SDN AGO GIN MWI SLE KEN ZMB COD CAF UGA MOZ ETH
Source
Nuclear
Hydro
Solar
Wind
Other

We grouped the findings for Latin America & Caribbean, Asia, and Sub-Saharan countries because their electricity mixes are similar and mainly driven by hydropower.

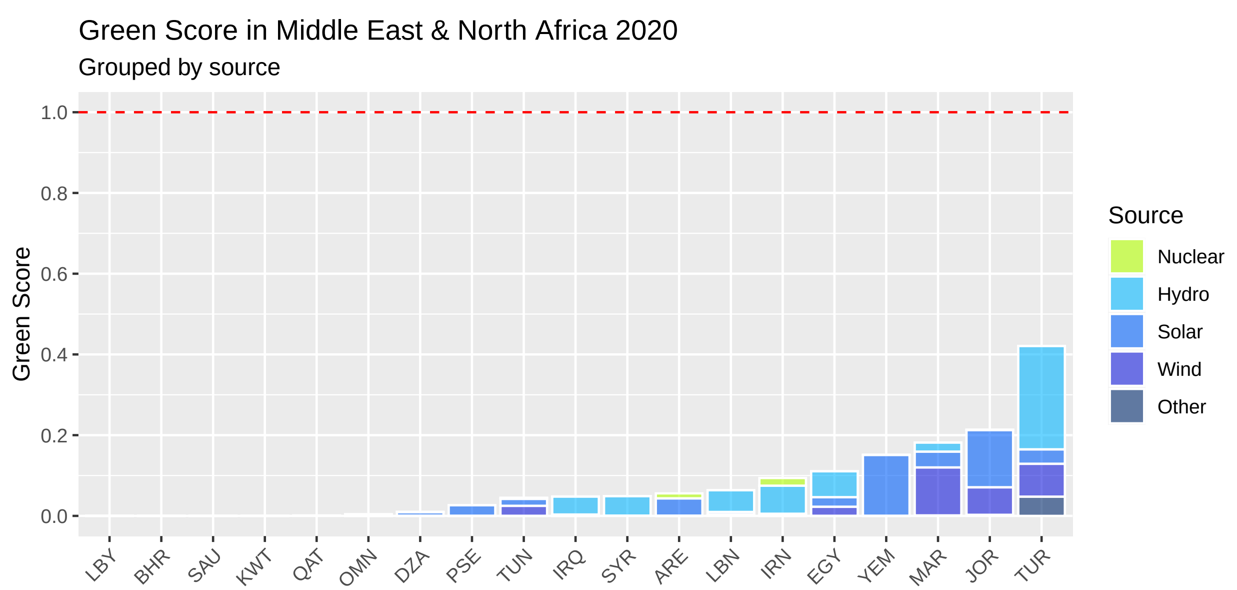
The **best-performing** countries in each macroregion are Costa Rica, Uruguay, and Ecuador for Latin America & Caribbean; Tajikistan, Kyrgyzstan, and North Korea for Asia; Ethiopia, Mozambique, and Uganda for Sub-Saharan Africa. Instead, the countries with the lowest Green Score are respectively: Haiti, Porto Rico, and Trinidad & Tobago; Timor Est, Turkmenistan, and Solomon Islands; Botswana, Guinea-Bissau, Gambia, Djibouti, and Comoros (tied with totally fossil-dependent electricity generation).

Here are the other main observations:

1. some Latin American and Caribbean countries also have significant production from other sources, notably **Uruguay**, which mainly produces electricity through wind power;

2. **Kenya** is the only Sub-Saharan country with a good Green Score that generates significantly from non-hydropower sources. The reason is that the country exploits the incredibly cost-effective geothermic capacity of the Rift Valley. [16]

```
# Creation of the plot for the North African & Middle-East countries
ggplot(filter(place, tag == "middle_east"),
       aes(x = reorder(iso_code, value), y = value, fill = Source)) +
  geom_bar(position = "stack", stat = "identity", alpha = 0.6, colour = "white") +
  scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
    scale_y_continuous(breaks = seq(0, 1, by = 0.2)) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(title = "Green Score in Middle East & North Africa 2020",
       subtitle = "Grouped by source",
       x = "",
       y = "Green Score") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



We conclude the descriptive analyses by examining the performance of Northern African and Middle East countries. Consistently with the findings in paragraph 4.3, the barplot shows that all the countries fail to reach a good performance: the best performing one, **Turkey**, has a Green Score smaller than 0.5. As the following table shows, this is mainly due to their large availability of oil and gas.

```r
# Creation of a dataset containing per capita average fossil reserves in Middle-East
# and Northern African countries VS the rest of the world
place = filter(main, year == 2020, tag == "middle_east") %>%
  select(population, oil_reserves_2020, gas_reserves) %>%
  mutate_all(~replace_na(.,0)) %>%
  summarize(oil_reserves_2020 = round(mean(oil_reserves_2020 / population),2),
            gas_reserves = round(mean(gas_reserves / population),2))

place2 = filter(main, year == 2020, tag != "middle_east") %>%
  select(population, oil_reserves_2020, gas_reserves) %>%
  mutate_all(~replace_na(.,0)) %>%
  summarize(oil_reserves_2020 = round(mean(oil_reserves_2020 / population),2),
            gas_reserves = round(mean(gas_reserves / population),2))

place = rbind(place, place2)
place = cbind(c("Middle East & North Africa", "Other countries"), place)
colnames(place) = c("Macroregion", "Oil (2020)", "Gas (2020)")
kable(place, caption = "Per capita reserves of fossil electricity sources")
```

Table 2: Per capita reserves of fossil electricity sources

Macroregion	Oil (2020)	Gas (2020)
Middle East & North Africa	469.81	583180.3
Other countries	18.40	22920.9

Chapter 5: Modeling

5.1 Initial data preparation for Modeling

To build the models, we start by preparing the data for it.

We take the dataset transformed in the exploratory analysis, i.e. per capita (every million inhabitants) and/or logarithmic-transformed data. We add a column containing the **tag** variable to this dataset, as done in the descriptive analysis. This time we insert both the normal feature and a dummy version of it. The reason is that linear models and stepwise selection work with the first one, while lasso and ridge models need the second one.

We then filter the dataset by considering only units that have year's variable between 2000 and 2019, and then we get a subset that contains only complete cases of the dataset.

```
# Adding tags here as well
mainlog = cbind(mainlog, tag)
mainlog = cbind(mainlog, model.matrix(~-1+tag, data=mainlog))
# Keep only years between 2000 and 2019
mainlogm = mainlog[mainlog$year %in% 2000:2019,]

mainlogm=mainlogm[complete.cases(mainlogm),]
```

5.2 Functions definition

In order to make multiple models, we create tailored functions to make the code less repetitive.

We first define two simple functions to transform the variables: one applies min-max normalization; the other contains the logit function, modified to avoids infinite values (since our data lies in the interval [0,1]).

We also define what columns we will later apply the functions to (in this case all numerical columns except **year**).

```
# Normalizing columns
normalize = function(x) {
  return((x- min(x)) /(max(x)-min(x)))
}

# Applying logit to normalized column
logify = function(x){
  return(qlogis((x/1.00001)+0.000005))
}

normcols=c(4:46)
```

In the following step, we define the function *selectvar_nl*, which allows us to select which columns to include in the model as independent variables.

The arguments are all boolean except the last one:

- *Bi*, identifies whether we want to include *Energy dataset*'s variables which regard the specific energy sources electricity production;
- *Bt*, identifies whether we want to include **tag**;

- *By*, identifies whether the functions defined above to normalize and apply the logit function is applied over the column grouped by year, or on the whole column;
- *Bl*, identifies whether to apply the logit function or not (normalization is always applied);
- *mlm*, the dataset to be transformed.

The function returns a list with two items: the modified dataset and a sub-list with the names of the variables used.

```
selectvar_nl = function(Bi=Bindepsource, Bt=Btag, By=Byearwise, Bl=Blogify, mlm=mainlogm){
  if (By){
    #year-wise normalization
    for (i in unique(mlm$year)){
      mlm[mlm$year==i,normcols] = lapply(mlm[mlm$year==i,normcols], normalize)
    }
  } else {
    #overall normalization
    mlm[normcols] = lapply(mlm[normcols], normalize)
  }
  #normalize year
  mlm["year"] = lapply(mlm["year"], normalize)

  if (Bl){
    if(By){
      #year-wise logify
      for (i in unique(mlm$year)){
        mlm[mlm$year==i,normcols] = lapply(mlm[mlm$year==i,normcols], logify)
      }
    } else {
      #overall logify
      mlm[normcols] = lapply(mlm[normcols], logify)
    }
    #logify year
    mlm["year"] = lapply(mlm["year"], logify)
  }

  if (Bi){
    independent = colnames(mlm)[c(2,4,6,14,18,23,25,28,33,35,37,38,39,41,42,43,44,45,46)]
  } else {
    independent = colnames(mlm)[c(2,4,37,38,39,41,42,43,44,45,46)]
  }
  if (Bt){
    independent = append(independent, colnames(mlm[47]))
  }
  return(list("mlm"=mlm, "ind"=independent))
}
```

Next, we define the function *setup_lr*, which has four arguments:

- *mlm*, the dataset used for the model;
- *dep*, the name of the model's dependent variable in the model;
- *ind*, a list containing the model's independent variables names;

- *Bt*, boolean, identifies whether the model should have **tag** as an independent variable.

The function returns a list of two elements:

1. the list of values of the dependent variable;
2. a matrix containing the values of the features used as independent variables (the transformation to matrix is needed to use the function that create lasso and ridge models).

```
#Setup for Lasso and Ridge
setup_lr = function(mlm=model_data, dep=dependent, ind=independent, Bt=Btag){
  y = mlm[,dep]
  if (Bt){
    x = data.matrix(select(mlm,c(head(ind,-1),tail(colnames(mlm),6))))
  } else {
    x = data.matrix(select(mlm,ind))
  }
  return(list("y"=y, "x"=x))
}
```

Then we define the function *rs_models*, which gets the same arguments as *setup_lr*.

The function runs the linear model with the dependent and independent variables passed in the arguments; then performs a stepwise variables selection using BIC as criterion (k=log(n)).

Subsequently, it calls the function *setup_lr* to get the matrices to run *glmnet*; it uses cross-validation to find the best lambda and then uses it to build the elastic-net models (lasso with alpha = 1, ridge with alpha=0).

Finally, the function returns a list of 5 sub-lists:

- *rs*, which contains the R squared of each model obtained (linear, stepwise, lasso, and ridge);
- *coeff_lm*, which contains the coefficients in the linear model;
- *coeff_sm*, which contains the coefficients in the linear model after stepwise variable selection;
- *coeff_lasso*, which contains the coefficients in the lasso model;
- *coeff_ridge*, which contains the coefficients in the ridge model.

All the coefficients' variables are ordered by their descending absolute value.

```
rs_models = function(mlm=model_data, dep=dependent, ind=independent, Bt=Btag){
  my_formula = as.formula(paste(paste(dep), " ~ ", paste(ind, collapse = " + ")))
  model = lm(my_formula, mlm)
  step_model=step(model,direction=c("both"), trace=FALSE, k=log(nrow(mainlogm)))
  xy = setup_lr(mlm, dep, ind, Bt)
  x = xy$x
  y = xy$y
  cv_model = cv.glmnet(x, y, alpha = 1)
  best_lasso = glmnet(x, y, alpha = 1, lambda = cv_model$lambda.min)
  cv_model = cv.glmnet(x, y, alpha = 0)
  best_ridge = glmnet(x, y, alpha = 0, lambda = cv_model$lambda.min)
  names_coeff_lasso = dimnames(coef(best_lasso))[[1]][which(coef(best_lasso) != 0)]
  names_coeff_ridge = dimnames(coef(best_ridge))[[1]][which(coef(best_ridge) != 0)]
```

```
  values_coeff_lasso = coef(best_lasso)[which(coef(best_lasso) != 0)]
  values_coeff_ridge = coef(best_ridge)[which(coef(best_ridge) != 0)]
  coeff_lasso = setNames(values_coeff_lasso, names_coeff_lasso)
  coeff_ridge = setNames(values_coeff_ridge, names_coeff_ridge)
  coeff_lm = model$coefficients
  coeff_sm = step_model$coefficients
  rsq = c(summary(model)$r.squared, summary(step_model)$r.squared,
          best_lasso$dev.ratio, best_ridge$dev.ratio)
  return(list("rs" = setNames(rsq, c("lm","sm","lasso","ridge")),
              "coeff_lm" = coeff_lm[order(-abs(sapply(coeff_lm,'[[',1)))],
              "coeff_sm" = coeff_sm[order(-abs(sapply(coeff_sm,'[[',1)))],
              "coeff_lasso" = coeff_lasso[order(-abs(sapply(coeff_lasso,'[[',1)))],
              "coeff_ridge" = coeff_ridge[order(-abs(sapply(coeff_ridge,'[[',1)))]))
}
```

The last function defined is *do_models*.

It only has one argument, *depen*, which is the name of the dependent variable we want to have in the model.

The function builds a data frame with eight columns: the first four contain the boolean values used as arguments by *selectvar_nl*; the last four are the values returned by *rs_models* in the *rs* item (the R squared value for the four models).

For each possible combination of True-False values requested as input by *selectvar_nl*, the function calls *select var* and then *rs_models*, giving as arguments the data and list of independent variables returned by *selectvar_nl*; the value are then inserted in the data frame.

The function then prints which arguments are to be passed to get the best models (to be precise, the model with the best R squared for the linear with stepwise selection) and its scores, together with the coefficients of the stepwise selection model with such arguments.

Then the same is printed again, but this time only considering the models that do not contain the energy source data as independent variables.

Finally, the whole data frame is printed.

This approach allows us to compare the performance of all different models to learn how to build the best model for the data.

```
do_models = function(depen = dependent){
  dependent = depen

  results_df = data.frame(Bindepsource = logical(), Btag = logical(),
                          Byearwise = logical(), Blogify = logical(),
                          lm = numeric(), sm = numeric(),
                          lasso = numeric(), ridge = numeric())

  for (i in c(TRUE, FALSE)) {
    for (j in c(TRUE, FALSE)) {
      for (k in c(TRUE, FALSE)) {
        for (l in c(TRUE, FALSE)) {
          tmp = selectvar_nl(Bi=i, Bt=j, By=k, Bl=l)
          model_data = tmp$mlm
          independent = tmp$ind
          results = rs_models(mlm=model_data, dep=dependent, ind=independent, Bt=j)
          results_df = rbind(results_df, data.frame(Bindepsource = i,
                                                    Btag = j,
```

```
                                                        Byearwise = k,
                                                        Blogify = l,
                                                        lm = results$rs['lm'],
                                                        sm = results$rs['sm'],
                                                        lasso = results$rs['lasso'],
                                                        ridge = results$rs['ridge']))
        }
      }
    }
  }

  max_row = results_df[which.max(results_df$sm), ]

  max_row_nind = results_df[which.max(results_df[9:16,]$sm)+8, ]

  tmp = selectvar_nl(Bi=max_row$Bindepsource, Bt=max_row$Btag,
                     By=max_row$Byearwise, Bl=max_row$Blogify)
  model_data = tmp$mlm
  independent = tmp$ind
  results = rs_models(mlm=model_data, dep=dependent,
                      ind=independent, Bt=max_row$Btag)

  tmp = selectvar_nl(Bi=max_row_nind$Bindepsource, Bt=max_row_nind$Btag,
                     By=max_row_nind$Byearwise, Bl=max_row_nind$Blogify)
  model_data = tmp$mlm
  independent = tmp$ind
  results_nind = rs_models(mlm=model_data, dep=dependent,
                           ind=independent, Bt=max_row_nind$Btag)

  print(paste("Models for the variable:", dependent))
  print("")
  print(kable(as.data.frame(max_row), caption = "Best performing models"))
  print(kable(as.data.frame(results$coeff_sm),
              caption = "Coefficients for best stepwise lm"))
  print(kable(as.data.frame(max_row_nind),
              caption = "Best performing models with only external data"))
  print(kable(as.data.frame(results_nind$coeff_sm),
              caption = "Coefficients for best stepwise lm with only external data"))
  print(kable(results_df, caption = "Performance on all models"))

}
```

5.3 Obtaining Models

The last step involves only calling the **do_models** function, passing the dependent variable.

Ten dependent variables are passed, all included in the initial *Energy dataset*: *carbon intensity of electricity*, *greenhouse gas emissions*, *share of electricity of fossil electricity*, and the share of electricity of all LC electricity sources.

Notice that for the last eight features, we will mainly focus on analyzing the model with only external independent variables since when analyzing the share of electricity of a given source, the electricity production of such source has a high correlation, giving us a high R squared, but only trivial results.

It is crucial to notice also that, when applying the logit function, the numerical variables in the dataset are all distributed between roughly -12 and +12, so their coefficients are comparable among them but are not directly comparable to the coefficients for the tag variables; instead, when the logit function is not applied, they can be directly compared.

GDP and HDI, which are significantly correlated, often appear with similar but opposite coefficients, which indicates rich countries where investment in human development and freedom is not as high when GDP has a positive coefficient and HDI a negative one, and the opposite when GDP is negative and HDI positive.

Here we briefly summarized the obtained results, based on the arguments of the functions:

- In all the best models obtained, the **tag** feature is kept during the stepwise selection, meaning it is significant;
- The *yearwise* argument in the best models is almost always False (with the exception of the best model for *other renewables*; however, even in that case with the argument being True, R squared is almost identical);
- Applying the logit function instead has mixed results: in some models in fact it halves the R squared, while in others it greatly improves it.

5.3.1 Carbon intensity of electricity

```r
do_models(depen="carbon_intensity_elec")
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(ind)` instead of `ind` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
## [1] "Models for the variable: carbon_intensity_elec"
## [1] ""
##
##
## Table: Best performing models
##
## |    |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:---|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm2 |TRUE         |TRUE |FALSE     |TRUE    | 0.5345208| 0.5312796| 0.5343709| 0.5306843|
##
##
## Table: Coefficients for best stepwise lm
##
## |                    | results$coeff_sm|
## |:-------------------|----------------:|
## |(Intercept)         |       -0.6665385|
## |tagdeveloped        |       -0.6303496|
## |tageast_europe      |        0.4009654|
## |gdp                 |        0.3890607|
## |tagsub_african      |       -0.3671303|
## |hdi                 |       -0.2808178|
## |hydro_electricity   |       -0.2599964|
## |taglatin            |       -0.2274012|
## |oil_electricity     |        0.1660544|
```

```
## |coal_electricity    |        0.1180649|
## |land_area           |        0.1062362|
## |nuclear_electricity |       -0.0779265|
## |agri_land_rate      |        0.0735105|
## |urbaniz_rate        |        0.0642560|
## |coal_reserves_2021  |        0.0250738|
## |tagmiddle_east      |       -0.0152201|
##
##
## Table: Best performing models with only external data
##
## |     |Bindepsource |Btag |Byearwise |Blogify |       lm|        sm|     lasso|     ridge|
## |:----|:------------|:----|:---------|:-------|--------:|---------:|---------:|---------:|
## |lm11 |FALSE        |TRUE |FALSE     |FALSE   | 0.260983| 0.2595082| 0.2609641| 0.2590556|
##
##
## Table: Coefficients for best stepwise lm with only external data
##
## |                      | results_nind$coeff_sm|
## |:---------------------|---------------------:|
## |(Intercept)           |             0.8868181|
## |population            |            -0.5452103|
## |hdi                   |            -0.3692295|
## |land_area             |            -0.3586299|
## |gdp                   |             0.3352992|
## |coal_reserves_2021    |             0.1954342|
## |tagdeveloped          |            -0.1184115|
## |agri_land_rate        |             0.1027159|
## |gas_reserves          |             0.0958774|
## |uranium_reserves_2019 |             0.0947377|
## |tageast_europe        |            -0.0805312|
## |tagmiddle_east        |             0.0749988|
## |tagsub_african        |            -0.0697385|
## |taglatin              |            -0.0626034|
##
##
## Table: Performance on all models
##
## |     |Bindepsource |Btag  |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:-----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm   |TRUE         |TRUE  |TRUE      |TRUE    | 0.5050993| 0.5032466| 0.5039708| 0.5025404|
## |lm1  |TRUE         |TRUE  |TRUE      |FALSE   | 0.4671707| 0.4650698| 0.4671266| 0.4635382|
## |lm2  |TRUE         |TRUE  |FALSE     |TRUE    | 0.5345208| 0.5312796| 0.5343709| 0.5306843|
## |lm3  |TRUE         |TRUE  |FALSE     |FALSE   | 0.4832769| 0.4816792| 0.4832405| 0.4786653|
## |lm4  |TRUE         |FALSE |TRUE      |TRUE    | 0.4896714| 0.4879791| 0.4889842| 0.4872723|
## |lm5  |TRUE         |FALSE |TRUE      |FALSE   | 0.4265291| 0.4245340| 0.4265070| 0.4241049|
## |lm6  |TRUE         |FALSE |FALSE     |TRUE    | 0.5088925| 0.5039002| 0.5088646| 0.5058196|
## |lm7  |TRUE         |FALSE |FALSE     |FALSE   | 0.4413045| 0.4386144| 0.4412839| 0.4387359|
## |lm8  |FALSE        |TRUE  |TRUE      |TRUE    | 0.1550579| 0.1522984| 0.1550360| 0.1547510|
## |lm9  |FALSE        |TRUE  |TRUE      |FALSE   | 0.2501506| 0.2500491| 0.2500588| 0.2488066|
## |lm10 |FALSE        |TRUE  |FALSE     |TRUE    | 0.1782024| 0.1750655| 0.1781695| 0.1770559|
## |lm11 |FALSE        |TRUE  |FALSE     |FALSE   | 0.2609830| 0.2595082| 0.2609641| 0.2590556|
## |lm12 |FALSE        |FALSE |TRUE      |TRUE    | 0.1114156| 0.1065295| 0.1114116| 0.1111432|
## |lm13 |FALSE        |FALSE |TRUE      |FALSE   | 0.2077355| 0.2077320| 0.2076402| 0.2069483|
```

```
## |lm14 |FALSE        |FALSE |FALSE     |TRUE    | 0.1418332| 0.1358737| 0.1417035| 0.1406853|
## |lm15 |FALSE        |FALSE |FALSE     |FALSE   | 0.2203351| 0.2189623| 0.2203175| 0.2192978|
```

Performance on the carbon intensity of electricity is relatively low, even with the energy source variables in the model, at ~0.53.

GDP and HDI have opposite weights, indicating that high carbon intensity is related to rich countries with lower HDI. However, the HDI coefficient is only ~.75 of the GDP, indicating that richer countries' inhabitants still pollute more than poorer countries.

The tag for Eastern Europe is positive, while the others are negative, with the Developed countries tag having a high coefficient (which also balances out with the score for GDP), relating to the fact that people in many Developed countries pollute less, thanks to the high share of renewables in their electricity mix.

As expected, coefficients related to the source of electricity are positive for fossil sources and negative for hydro and nuclear, solar, and wind. Therefore, they are instead not kept in the stepwise model.

Regarding the model excluding source data, the performance is ~0.26, and does not use logit on the data.

Now population and land area have high negative coefficients, indicating that populations in smaller but significantly less populated countries pollute way more.

The coefficients on the tags are different, with Eastern countries having a negative coefficient while middle-east having a positive coefficient. This also relates to the influence of population and land area above: Eastern European countries tend to be extensive and populated, while in the middle east this is not necessarily true; thus, we probably have outliers of small countries that pollute a lot due to oil production.

5.3.2 Greenhouse gas emissions

```
do_models(depen="greenhouse_gas_emissions")
```

```
## [1] "Models for the variable: greenhouse_gas_emissions"
## [1] ""
##
##
## Table: Best performing models
##
## |    |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|    lasso|     ridge|
## |:---|:------------|:----|:---------|:-------|---------:|---------:|--------:|---------:|
## |lm3 |TRUE         |TRUE |FALSE     |FALSE   | 0.9997319| 0.9997304| 0.999026| 0.9900342|
##
##
## Table: Coefficients for best stepwise lm
##
## |                            | results$coeff_sm|
## |:---------------------------|----------------:|
## |gas_electricity             |        0.9969829|
## |coal_electricity            |        0.7230451|
## |oil_electricity             |        0.6358749|
## |hydro_electricity           |        0.0912293|
## |other_renewable_electricity |        0.0759251|
## |nuclear_electricity         |        0.0218374|
## |wind_electricity            |        0.0177594|
## |hdi                         |        0.0029589|
## |land_area                   |        0.0019338|
```

```
## |(Intercept)               |       -0.0013812|
## |coal_reserves_2021        |       -0.0012207|
## |urbaniz_rate              |       -0.0011272|
## |tageast_europe            |       -0.0008518|
## |agri_land_rate            |       -0.0007515|
## |tagmiddle_east            |       -0.0005090|
## |tagdeveloped              |        0.0004676|
## |tagsub_african            |        0.0003568|
## |taglatin                  |        0.0001462|
##
##
## Table: Best performing models with only external data
##
## |     |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm10 |FALSE        |TRUE |FALSE     |TRUE    | 0.7778692| 0.7776551| 0.7778094| 0.7748152|
##
##
## Table: Coefficients for best stepwise lm with only external data
##
## |                      | results_nind$coeff_sm|
## |:---------------------|---------------------:|
## |(Intercept)           |            -1.9531608|
## |tagmiddle_east        |             0.9106724|
## |gdp                   |             0.8684919|
## |tagsub_african        |            -0.7702664|
## |tagdeveloped          |            -0.6339248|
## |tageast_europe        |             0.4767372|
## |hdi                   |             0.2615497|
## |population            |            -0.2452021|
## |urbaniz_rate          |             0.1770140|
## |taglatin              |            -0.1219884|
## |land_area             |            -0.0787747|
## |coal_reserves_2021    |             0.0710509|
## |gas_reserves          |             0.0599651|
## |oil_reserves_2020     |            -0.0285059|
## |uranium_reserves_2019 |             0.0196204|
##
##
## Table: Performance on all models
##
## |     |Bindepsource |Btag  |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:-----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm   |TRUE         |TRUE  |TRUE      |TRUE    | 0.7472059| 0.7465403| 0.7471697| 0.7451801|
## |lm1  |TRUE         |TRUE  |TRUE      |FALSE   | 0.9601086| 0.9600636| 0.9600697| 0.9532482|
## |lm2  |TRUE         |TRUE  |FALSE     |TRUE    | 0.8426872| 0.8424920| 0.8426383| 0.8395067|
## |lm3  |TRUE         |TRUE  |FALSE     |FALSE   | 0.9997319| 0.9997304| 0.9990260| 0.9900342|
## |lm4  |TRUE         |FALSE |TRUE      |TRUE    | 0.7256124| 0.7245526| 0.7255821| 0.7230750|
## |lm5  |TRUE         |FALSE |TRUE      |FALSE   | 0.9578822| 0.9577008| 0.9578303| 0.9506877|
## |lm6  |TRUE         |FALSE |FALSE     |TRUE    | 0.8156265| 0.8153165| 0.8155867| 0.8126405|
## |lm7  |TRUE         |FALSE |FALSE     |FALSE   | 0.9997277| 0.9997252| 0.9990152| 0.9906047|
## |lm8  |FALSE        |TRUE  |TRUE      |TRUE    | 0.6470084| 0.6461660| 0.6469782| 0.6458597|
## |lm9  |FALSE        |TRUE  |TRUE      |FALSE   | 0.6217946| 0.6200317| 0.6217469| 0.6135361|
## |lm10 |FALSE        |TRUE  |FALSE     |TRUE    | 0.7778692| 0.7776551| 0.7778094| 0.7748152|
```

```
## |lm11 |FALSE        |TRUE  |FALSE     |FALSE   | 0.6209938| 0.6196816| 0.6209481| 0.6129978|
## |lm12 |FALSE        |FALSE |TRUE      |TRUE    | 0.5845940| 0.5827457| 0.5845649| 0.5831239|
## |lm13 |FALSE        |FALSE |TRUE      |FALSE   | 0.5748852| 0.5748006| 0.5748509| 0.5663684|
## |lm14 |FALSE        |FALSE |FALSE     |TRUE    | 0.7319622| 0.7311077| 0.7319295| 0.7300817|
## |lm15 |FALSE        |FALSE |FALSE     |FALSE   | 0.5724844| 0.5724219| 0.5724514| 0.5639957|
```

Greenhouse gas emission models have a very high R squared, with the best model ~1.

Gas, coal, and oil electricity explain most of the independent feature. However, curiously also hydro, other renewables, and nuclear have a positive correlation.

The model without sources still has a very high R squared ~0.77.

In this case, there is a significant influence of GDP with positive coefficients, and HDI also has a positive coefficient, indicating greenhouse gas emissions are very correlated to rich countries,

The tags are positive for the Middle East & North Africa and Eastern Europe and negative for the others, especially Sub-Saharan Africa and Developed countries.

5.3.3 Hydro share of electricity

```
do_models(depen="hydro_share_elec")
```

```
## [1] "Models for the variable: hydro_share_elec"
## [1] ""
##
##
## Table: Best performing models
##
## |    |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:---|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm2 |TRUE         |TRUE |FALSE     |TRUE    | 0.8722087| 0.8701415| 0.8721735| 0.8631823|
##
##
## Table: Coefficients for best stepwise lm
##
## |                            | results$coeff_sm|
## |:---------------------------|----------------:|
## |(Intercept)                 |        4.0400231|
## |hydro_electricity           |        1.5816282|
## |tageast_europe              |       -1.3485628|
## |tagsub_african              |        0.9492310|
## |gdp                         |       -0.7860944|
## |tagmiddle_east              |       -0.6490157|
## |land_area                   |       -0.5314326|
## |taglatin                    |        0.4566842|
## |oil_electricity             |       -0.2621500|
## |urbaniz_rate                |       -0.2119081|
## |other_renewable_electricity |       -0.1684825|
## |population                  |        0.1617658|
## |coal_electricity            |       -0.1481847|
## |wind_electricity            |        0.0601382|
## |tagdeveloped                |       -0.0107972|
##
```

```
##
## Table: Best performing models with only external data
##
## |     |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm11 |FALSE        |TRUE |FALSE     |FALSE   | 0.3396587| 0.3395835| 0.3396108| 0.3330256|
##
##
## Table: Coefficients for best stepwise lm with only external data
##
## |                     | results_nind$coeff_sm|
## |:--------------------|---------------------:|
## |hdi                  |             0.8046347|
## |gdp                  |            -0.7955483|
## |land_area            |             0.6642048|
## |population           |             0.5931483|
## |(Intercept)          |            -0.2410406|
## |tagmiddle_east       |            -0.1716373|
## |uranium_reserves_2019 |           -0.1701178|
## |coal_reserves_2021    |           -0.1686447|
## |agri_land_rate       |            -0.1684242|
## |urbaniz_rate         |            -0.1511656|
## |tagsub_african       |             0.1509692|
## |gas_reserves         |            -0.1468586|
## |taglatin             |             0.1042733|
## |year                 |            -0.0909412|
## |tagdeveloped         |             0.0730765|
## |tageast_europe       |            -0.0184525|
##
##
## Table: Performance on all models
##
## |     |Bindepsource |Btag  |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:-----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm   |TRUE         |TRUE  |TRUE      |TRUE    | 0.8163382| 0.8147588| 0.8163042| 0.8076685|
## |lm1  |TRUE         |TRUE  |TRUE      |FALSE   | 0.4736208| 0.4729328| 0.4735482| 0.4684607|
## |lm2  |TRUE         |TRUE  |FALSE     |TRUE    | 0.8722087| 0.8701415| 0.8721735| 0.8631823|
## |lm3  |TRUE         |TRUE  |FALSE     |FALSE   | 0.4946946| 0.4941418| 0.4946709| 0.4886738|
## |lm4  |TRUE         |FALSE |TRUE      |TRUE    | 0.8046176| 0.8043559| 0.8045764| 0.7965902|
## |lm5  |TRUE         |FALSE |TRUE      |FALSE   | 0.4228089| 0.4204118| 0.4227709| 0.4202896|
## |lm6  |TRUE         |FALSE |FALSE     |TRUE    | 0.8654082| 0.8645582| 0.8653555| 0.8569189|
## |lm7  |TRUE         |FALSE |FALSE     |FALSE   | 0.4458070| 0.4437792| 0.4457821| 0.4431344|
## |lm8  |FALSE        |TRUE  |TRUE      |TRUE    | 0.2400780| 0.2381664| 0.2400263| 0.2397156|
## |lm9  |FALSE        |TRUE  |TRUE      |FALSE   | 0.3263678| 0.3257687| 0.3263387| 0.3211063|
## |lm10 |FALSE        |TRUE  |FALSE     |TRUE    | 0.2964711| 0.2945068| 0.2964342| 0.2949013|
## |lm11 |FALSE        |TRUE  |FALSE     |FALSE   | 0.3396587| 0.3395835| 0.3396108| 0.3330256|
## |lm12 |FALSE        |FALSE |TRUE      |TRUE    | 0.1990178| 0.1989606| 0.1987186| 0.1987068|
## |lm13 |FALSE        |FALSE |TRUE      |FALSE   | 0.2697632| 0.2691073| 0.2697512| 0.2677978|
## |lm14 |FALSE        |FALSE |FALSE     |TRUE    | 0.2512429| 0.2493177| 0.2511527| 0.2494011|
## |lm15 |FALSE        |FALSE |FALSE     |FALSE   | 0.2829128| 0.2824542| 0.2828892| 0.2805562|
```

Hydroelectricity is obviously a good indicator when sources are in the model. However, GDP and land area are also very influential, which might make us think that this clean energy source is also easily accessible to poorer and smaller countries; nonetheless, these are just an indication of total electricity production (especially regarding GDP: richer countries use more electricity pro capita).

In fact, the model excluding sources, land area, and population has negative coefficients, with HDI having a positive coefficient and GDP a negative one, indicating that for a high share of hydroelectricity, the country needs to be able to make sizeable investments.

In the model excluding source, we also have this time year included, with a small yet negative coefficient, indicating that the share of hydro in the electricity mix is getting lower each year if the other parameters are fixed.

5.3.4 Solar share of electricity

```
do_models(depen="solar_share_elec")
```

```
## [1] "Models for the variable: solar_share_elec"
## [1] ""
##
##
## Table: Best performing models
##
## |    |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:---|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm2 |TRUE         |TRUE |FALSE     |TRUE    | 0.9532925| 0.9529513| 0.9532219| 0.9406927|
##
##
## Table: Coefficients for best stepwise lm
##
## |                    | results$coeff_sm|
## |:-------------------|----------------:|
## |solar_electricity   |        1.0521981|
## |tagdeveloped        |       -0.7459863|
## |tagmiddle_east      |       -0.2681365|
## |tagsub_african      |       -0.1778307|
## |taglatin            |       -0.1768368|
## |(Intercept)         |        0.1255167|
## |tageast_europe      |       -0.1223003|
## |population          |        0.1171737|
## |hdi                 |       -0.0956029|
## |agri_land_rate      |        0.0394254|
## |nuclear_electricity |       -0.0393302|
## |coal_reserves_2021  |       -0.0231408|
## |gas_electricity     |       -0.0212437|
##
##
## Table: Best performing models with only external data
##
## |     |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm10 |FALSE        |TRUE |FALSE     |TRUE    | 0.3338985| 0.3322466| 0.3338052| 0.3334057|
##
##
## Table: Coefficients for best stepwise lm with only external data
##
```

```
## |                       | results_nind$coeff_sm|
## |:----------------------|---------------------:|
## |(Intercept)            |            -9.8037091|
## |tagdeveloped           |             1.1769652|
## |tageast_europe         |            -0.8854891|
## |population             |             0.8332436|
## |tagsub_african         |             0.7714459|
## |hdi                    |             0.6952542|
## |land_area              |            -0.4253591|
## |gdp                    |             0.3400174|
## |year                   |             0.3058422|
## |urbaniz_rate           |            -0.2534462|
## |taglatin               |            -0.0729929|
## |oil_reserves_2020      |            -0.0575208|
## |uranium_reserves_2019  |             0.0494156|
## |coal_reserves_2021     |             0.0477924|
## |tagmiddle_east         |            -0.0407383|
##
##
## Table: Performance on all models
##
## |     |Bindepsource |Btag  |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:-----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm   |TRUE         |TRUE  |TRUE      |TRUE    | 0.9092341| 0.9085198| 0.9091595| 0.8979058|
## |lm1  |TRUE         |TRUE  |TRUE      |FALSE   | 0.6512108| 0.6473243| 0.6511782| 0.6462251|
## |lm2  |TRUE         |TRUE  |FALSE     |TRUE    | 0.9532925| 0.9529513| 0.9532219| 0.9406927|
## |lm3  |TRUE         |TRUE  |FALSE     |FALSE   | 0.6318986| 0.6262037| 0.6314521| 0.6272721|
## |lm4  |TRUE         |FALSE |TRUE      |TRUE    | 0.9074616| 0.9074010| 0.9073140| 0.8969142|
## |lm5  |TRUE         |FALSE |TRUE      |FALSE   | 0.6499696| 0.6473243| 0.6498515| 0.6450456|
## |lm6  |TRUE         |FALSE |FALSE     |TRUE    | 0.9517905| 0.9514997| 0.9517403| 0.9395924|
## |lm7  |TRUE         |FALSE |FALSE     |FALSE   | 0.6285994| 0.6262037| 0.6285413| 0.6241299|
## |lm8  |FALSE        |TRUE  |TRUE      |TRUE    | 0.3250663| 0.3241064| 0.3250531| 0.3245504|
## |lm9  |FALSE        |TRUE  |TRUE      |FALSE   | 0.1853965| 0.1821087| 0.1853850| 0.1848927|
## |lm10 |FALSE        |TRUE  |FALSE     |TRUE    | 0.3338985| 0.3322466| 0.3338052| 0.3334057|
## |lm11 |FALSE        |TRUE  |FALSE     |FALSE   | 0.1776729| 0.1740659| 0.1761799| 0.1767252|
## |lm12 |FALSE        |FALSE |TRUE      |TRUE    | 0.2626138| 0.2625994| 0.2624104| 0.2622873|
## |lm13 |FALSE        |FALSE |TRUE      |FALSE   | 0.1444018| 0.1427092| 0.1443954| 0.1436379|
## |lm14 |FALSE        |FALSE |FALSE     |TRUE    | 0.3075653| 0.3056911| 0.3075516| 0.3071665|
## |lm15 |FALSE        |FALSE |FALSE     |FALSE   | 0.1468499| 0.1419095| 0.1468195| 0.1459880|
```

Regarding solar generation, the model with sources is trivial, with most of the share explained by solar electricity.

It is interesting to analyze the model excluding source, in which HDI, GDP, and population have positive scores, indicating solar is accessible mainly to more prosperous and highly populated countries.

Here, year appears with a noticeable weight, indicating that solar energy has quickly developed over the last two decades.

5.3.5 Wind share of electricity

```
do_models(depen="wind_share_elec")
```

```
## [1] "Models for the variable: wind_share_elec"
## [1] ""
##
##
## Table: Best performing models
##
## |    |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:---|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm2 |TRUE         |TRUE |FALSE     |TRUE    | 0.9844039| 0.9843127| 0.9843542| 0.9656907|
##
##
## Table: Coefficients for best stepwise lm
##
## |                    | results$coeff_sm|
## |:-------------------|----------------:|
## |tagdeveloped        |       -1.1416358|
## |wind_electricity    |        1.1100925|
## |(Intercept)         |        0.9686533|
## |tagsub_african      |       -0.2693459|
## |tageast_europe      |       -0.1501065|
## |hdi                 |       -0.1404808|
## |tagmiddle_east      |        0.0924604|
## |land_area           |       -0.0875162|
## |population          |        0.0869981|
## |gdp                 |        0.0626305|
## |nuclear_electricity |       -0.0224733|
## |gas_electricity     |       -0.0210091|
## |agri_land_rate      |        0.0182419|
## |hydro_electricity   |       -0.0118503|
## |oil_electricity     |       -0.0107341|
## |taglatin            |        0.0093200|
## |gas_reserves        |       -0.0068677|
##
##
## Table: Best performing models with only external data
##
## |     |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm10 |FALSE        |TRUE |FALSE     |TRUE    | 0.4903061| 0.4898799| 0.4902841| 0.4894546|
##
##
## Table: Coefficients for best stepwise lm with only external data
##
## |                      | results_nind$coeff_sm|
## |:---------------------|---------------------:|
## |(Intercept)           |           -10.1135085|
## |tagdeveloped          |             3.9109605|
## |taglatin              |             1.2642224|
## |tagmiddle_east        |             1.1993442|
## |population            |             0.8410582|
## |tageast_europe        |             0.7715965|
## |hdi                   |             0.7639646|
## |gdp                   |             0.3506423|
## |urbaniz_rate          |            -0.2312611|
```

```
## |land_area            |            -0.2032410|
## |year                 |             0.1710774|
## |agri_land_rate       |             0.1597170|
## |gas_reserves         |            -0.0817925|
## |coal_reserves_2021   |             0.0722841|
## |uranium_reserves_2019 |            0.0582874|
## |tagsub_african       |            -0.0408035|
##
##
## Table: Performance on all models
##
## |     |Bindepsource |Btag  |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:-----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm   |TRUE         |TRUE  |TRUE      |TRUE    | 0.9776968| 0.9776120| 0.9776440| 0.9609463|
## |lm1  |TRUE         |TRUE  |TRUE      |FALSE   | 0.8180183| 0.8150051| 0.8178047| 0.8091617|
## |lm2  |TRUE         |TRUE  |FALSE     |TRUE    | 0.9844039| 0.9843127| 0.9843542| 0.9656907|
## |lm3  |TRUE         |TRUE  |FALSE     |FALSE   | 0.7663574| 0.7638306| 0.7663231| 0.7590068|
## |lm4  |TRUE         |FALSE |TRUE      |TRUE    | 0.9737698| 0.9735686| 0.9737047| 0.9590652|
## |lm5  |TRUE         |FALSE |TRUE      |FALSE   | 0.8164373| 0.8150051| 0.8164004| 0.8083559|
## |lm6  |TRUE         |FALSE |FALSE     |TRUE    | 0.9790619| 0.9790006| 0.9790034| 0.9624837|
## |lm7  |TRUE         |FALSE |FALSE     |FALSE   | 0.7649547| 0.7638306| 0.7644308| 0.7582635|
## |lm8  |FALSE        |TRUE  |TRUE      |TRUE    | 0.4696318| 0.4681342| 0.4696054| 0.4686405|
## |lm9  |FALSE        |TRUE  |TRUE      |FALSE   | 0.2580726| 0.2565171| 0.2579735| 0.2565790|
## |lm10 |FALSE        |TRUE  |FALSE     |TRUE    | 0.4903061| 0.4898799| 0.4902841| 0.4894546|
## |lm11 |FALSE        |TRUE  |FALSE     |FALSE   | 0.2295707| 0.2246848| 0.2295570| 0.2284636|
## |lm12 |FALSE        |FALSE |TRUE      |TRUE    | 0.3626293| 0.3621573| 0.3626135| 0.3620319|
## |lm13 |FALSE        |FALSE |TRUE      |FALSE   | 0.1847001| 0.1804610| 0.1846845| 0.1835520|
## |lm14 |FALSE        |FALSE |FALSE     |TRUE    | 0.4486921| 0.4482159| 0.4486711| 0.4472531|
## |lm15 |FALSE        |FALSE |FALSE     |FALSE   | 0.1775404| 0.1751734| 0.1774840| 0.1766446|
```

Similarly, wind's share of electricity generation is mainly explained by the electricity production from wind, so it is not worth spending time on the best model.

The model excluding sources has an R squared ~0.49 and behaves very similarly to solar share. The major difference is in the tag *sub-african*, which for obvious geographical reasons has a positive coefficient in the "solar model", while here it has a negative one.

5.3.6 Other renewables share of electricity

```r
do_models(depen="other_renewables_share_elec")
```

```
## [1] "Models for the variable: other_renewables_share_elec"
## [1] ""
##
##
## Table: Best performing models
##
## |   |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:--|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm |TRUE         |TRUE |TRUE      |TRUE    | 0.9221327| 0.9214762| 0.9220396| 0.9062341|
##
##
```

```
## Table: Coefficients for best stepwise lm
##
## |                            | results$coeff_sm|
## |:---------------------------|----------------:|
## |(Intercept)                 |       -2.0156652|
## |tagdeveloped                |       -1.1702433|
## |other_renewable_electricity |        1.1096296|
## |tageast_europe              |        0.3979516|
## |taglatin                    |       -0.3697240|
## |population                  |       -0.1741052|
## |tagsub_african              |        0.1428462|
## |nuclear_electricity         |       -0.1076176|
## |land_area                   |        0.0767394|
## |coal_electricity            |       -0.0696384|
## |tagmiddle_east              |       -0.0580652|
## |coal_reserves_2021          |       -0.0562147|
## |urbaniz_rate                |       -0.0532090|
## |hydro_electricity           |       -0.0448959|
## |gas_reserves                |       -0.0326719|
## |year                        |        0.0250557|
##
##
## Table: Best performing models with only external data
##
## |     |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm10 |FALSE        |TRUE |FALSE     |TRUE    | 0.2600053| 0.2581676| 0.2598897| 0.2594965|
##
##
## Table: Coefficients for best stepwise lm with only external data
##
## |                   | results_nind$coeff_sm|
## |:------------------|---------------------:|
## |(Intercept)        |           -11.8484518|
## |taglatin           |             2.0748584|
## |tagmiddle_east     |            -1.3826720|
## |tagdeveloped       |             1.2478274|
## |population         |             0.7329831|
## |tagsub_african     |             0.6975479|
## |gdp                |             0.6221787|
## |hdi                |             0.5145639|
## |land_area          |             0.3980233|
## |urbaniz_rate       |            -0.1495560|
## |oil_reserves_2020  |            -0.1220423|
## |coal_reserves_2021 |            -0.0764886|
## |tageast_europe     |             0.0595342|
##
##
## Table: Performance on all models
##
## |     |Bindepsource |Btag  |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:-----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm   |TRUE         |TRUE  |TRUE      |TRUE    | 0.9221327| 0.9214762| 0.9220396| 0.9062341|
## |lm1  |TRUE         |TRUE  |TRUE      |FALSE   | 0.5515052| 0.5480883| 0.5514682| 0.5431949|
```

```
## |lm2  |TRUE         |TRUE  |FALSE     |TRUE    | 0.9216102| 0.9210009| 0.9215252| 0.9036343|
## |lm3  |TRUE         |TRUE  |FALSE     |FALSE   | 0.5220947| 0.5186935| 0.5220570| 0.5142534|
## |lm4  |TRUE         |FALSE |TRUE      |TRUE    | 0.9150720| 0.9145832| 0.9149931| 0.9012033|
## |lm5  |TRUE         |FALSE |TRUE      |FALSE   | 0.5499624| 0.5480883| 0.5499310| 0.5420196|
## |lm6  |TRUE         |FALSE |FALSE     |TRUE    | 0.9152209| 0.9148620| 0.9151793| 0.8993687|
## |lm7  |TRUE         |FALSE |FALSE     |FALSE   | 0.5209920| 0.5186935| 0.5209588| 0.5130419|
## |lm8  |FALSE        |TRUE  |TRUE      |TRUE    | 0.2176780| 0.2112594| 0.2176697| 0.2174437|
## |lm9  |FALSE        |TRUE  |TRUE      |FALSE   | 0.1261415| 0.1151474| 0.1261311| 0.1252381|
## |lm10 |FALSE        |TRUE  |FALSE     |TRUE    | 0.2600053| 0.2581676| 0.2598897| 0.2594965|
## |lm11 |FALSE        |TRUE  |FALSE     |FALSE   | 0.1394144| 0.1267532| 0.1394015| 0.1387185|
## |lm12 |FALSE        |FALSE |TRUE      |TRUE    | 0.1355634| 0.1331777| 0.1355558| 0.1354489|
## |lm13 |FALSE        |FALSE |TRUE      |FALSE   | 0.1160682| 0.1151474| 0.1160610| 0.1148401|
## |lm14 |FALSE        |FALSE |FALSE     |TRUE    | 0.1928867| 0.1917813| 0.1927802| 0.1924009|
## |lm15 |FALSE        |FALSE |FALSE     |FALSE   | 0.1320864| 0.1267532| 0.1320750| 0.1309237|
```

Other renewables' behavior is also very similar to solar. However, in the model without sources (which has the lowest score of ~0.26, indicating it is hard to explain it with our variables), we now see a high coefficient for Latin America & Caribbeans instead, indicating that this measure is also highly related to local availability and geography.

5.3.7 Nuclear share of electricity

```r
do_models(depen="nuclear_share_elec")
```

```
## [1] "Models for the variable: nuclear_share_elec"
## [1] ""
##
##
## Table: Best performing models
##
## |    |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|    lasso|     ridge|
## |:---|:------------|:----|:---------|:-------|---------:|---------:|--------:|---------:|
## |lm2 |TRUE         |TRUE |FALSE     |TRUE    | 0.9861162| 0.9859826| 0.986057| 0.9738284|
##
##
## Table: Coefficients for best stepwise lm
##
## |                    | results$coeff_sm|
## |:-------------------|----------------:|
## |(Intercept)         |        1.1275109|
## |nuclear_electricity |        1.0704264|
## |tagdeveloped        |       -0.3782303|
## |tagmiddle_east      |       -0.2109865|
## |tagsub_african      |       -0.1880907|
## |tageast_europe      |        0.1712895|
## |hdi                 |       -0.0884279|
## |population          |        0.0697808|
## |gdp                 |        0.0672300|
## |land_area           |       -0.0559275|
## |agri_land_rate      |        0.0546028|
## |coal_electricity    |       -0.0226589|
## |coal_reserves_2021  |        0.0155275|
```

```
## |gas_reserves        |        0.0088970|
## |taglatin            |       -0.0053640|
##
##
## Table: Best performing models with only external data
##
## |     |Bindepsource |Btag |Byearwise |Blogify |       lm|        sm|     lasso|     ridge|
## |:----|:------------|:----|:---------|:-------|--------:|---------:|---------:|---------:|
## |lm10 |FALSE        |TRUE |FALSE     |TRUE    | 0.459583| 0.4574956| 0.4595567| 0.4591493|
##
##
## Table: Coefficients for best stepwise lm with only external data
##
## |                      | results_nind$coeff_sm|
## |:---------------------|---------------------:|
## |(Intercept)           |            -9.4508164|
## |tageast_europe        |             3.6719535|
## |tagdeveloped          |             2.7903841|
## |taglatin              |             0.9463899|
## |population            |             0.7846926|
## |tagsub_african        |             0.6207119|
## |hdi                   |             0.6149629|
## |land_area             |            -0.4893601|
## |tagmiddle_east        |            -0.1663075|
## |coal_reserves_2021    |             0.1551027|
## |uranium_reserves_2019 |             0.1359981|
## |year                  |            -0.0569535|
## |oil_reserves_2020     |            -0.0505826|
##
##
## Table: Performance on all models
##
## |     |Bindepsource |Btag  |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:-----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm   |TRUE         |TRUE  |TRUE      |TRUE    | 0.9517532| 0.9514434| 0.9516967| 0.9412235|
## |lm1  |TRUE         |TRUE  |TRUE      |FALSE   | 0.8549568| 0.8541355| 0.8549163| 0.8471012|
## |lm2  |TRUE         |TRUE  |FALSE     |TRUE    | 0.9861162| 0.9859826| 0.9860570| 0.9738284|
## |lm3  |TRUE         |TRUE  |FALSE     |FALSE   | 0.8597170| 0.8590185| 0.8596738| 0.8520994|
## |lm4  |TRUE         |FALSE |TRUE      |TRUE    | 0.9495076| 0.9490532| 0.9494549| 0.9392815|
## |lm5  |TRUE         |FALSE |TRUE      |FALSE   | 0.8229483| 0.8224376| 0.8229203| 0.8143804|
## |lm6  |TRUE         |FALSE |FALSE     |TRUE    | 0.9848376| 0.9847507| 0.9847741| 0.9731689|
## |lm7  |TRUE         |FALSE |FALSE     |FALSE   | 0.8307860| 0.8296260| 0.8307482| 0.8225923|
## |lm8  |FALSE        |TRUE  |TRUE      |TRUE    | 0.4057491| 0.4033973| 0.4053289| 0.4053933|
## |lm9  |FALSE        |TRUE  |TRUE      |FALSE   | 0.3464401| 0.3417525| 0.3464212| 0.3443844|
## |lm10 |FALSE        |TRUE  |FALSE     |TRUE    | 0.4595830| 0.4574956| 0.4595567| 0.4591493|
## |lm11 |FALSE        |TRUE  |FALSE     |FALSE   | 0.3561215| 0.3545879| 0.3560845| 0.3545575|
## |lm12 |FALSE        |FALSE |TRUE      |TRUE    | 0.3171428| 0.3165326| 0.3167487| 0.3167390|
## |lm13 |FALSE        |FALSE |TRUE      |FALSE   | 0.2518207| 0.2482337| 0.2516937| 0.2513190|
## |lm14 |FALSE        |FALSE |FALSE     |TRUE    | 0.4036612| 0.4006403| 0.4036417| 0.4028174|
## |lm15 |FALSE        |FALSE |FALSE     |FALSE   | 0.2700016| 0.2665232| 0.2699901| 0.2691938|
```

Considering nuclear share as dependent, we focus on the model excluding sources, which has a score of ~0.46 for the best model.

The tags variable shows a high preference for nuclear mainly in Eastern Europe and Developed countries.

Population and HDI are the leading positive indicators, indicating that also nuclear is mainly accessible to richer and more populated countries; also, there is a negative coefficient for land area, showing its energy production efficiency in terms of electricity produced compared to the area needed to operate the nuclear plants.

Year's variable is also present in this model, this time with a negative coefficient, indicating that countries are progressively abandoning nuclear, or at least they stopped investing in it.

Uranium reserves have a positive coefficient, indicating they can be a reason to invest in nuclear energy. However, it is not very high, indicating it is not a main concern, probably because of the low amount of material needed to produce electricity from nuclear, which makes transporting the material a small cost without the need of building infrastructure for it.

5.3.8 Renewables, Low-carbon and Fossil share of electricity

```r
do_models(depen="renewables_share_elec")
```

```
## [1] "Models for the variable: renewables_share_elec"
## [1] ""
##
##
## Table: Best performing models
##
## |    |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:---|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm2 |TRUE         |TRUE |FALSE     |TRUE    | 0.7924768| 0.7909105| 0.7922676| 0.7858865|
##
##
## Table: Coefficients for best stepwise lm
##
## |                           | results$coeff_sm|
## |:--------------------------|----------------:|
## |(Intercept)                |        5.1854476|
## |hydro_electricity          |        1.3302253|
## |gdp                        |       -0.9694312|
## |tagsub_african             |        0.5898365|
## |tagdeveloped               |        0.5494048|
## |tagmiddle_east             |       -0.4992895|
## |tageast_europe             |       -0.4278137|
## |land_area                  |       -0.3815821|
## |oil_electricity            |       -0.3268700|
## |taglatin                   |        0.2620910|
## |coal_electricity           |       -0.1894783|
## |other_renewable_electricity |        0.1600416|
## |gas_electricity            |       -0.1410298|
## |wind_electricity           |        0.1380852|
## |solar_electricity          |        0.1122998|
## |nuclear_electricity        |       -0.0896496|
## |gas_reserves               |        0.0465484|
## |coal_reserves_2021         |       -0.0365110|
##
##
## Table: Best performing models with only external data
```

```
##
## |     |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm11 |FALSE        |TRUE |FALSE     |FALSE   | 0.3398832| 0.3393901| 0.3398533| 0.3357615|
##
##
## Table: Coefficients for best stepwise lm with only external data
##
## |                      | results_nind$coeff_sm|
## |:---------------------|---------------------:|
## |land_area             |             0.7005778|
## |hdi                   |             0.6770543|
## |population            |             0.6265031|
## |gdp                   |            -0.6261981|
## |(Intercept)           |            -0.2861480|
## |urbaniz_rate          |            -0.1917985|
## |tagmiddle_east        |            -0.1910241|
## |coal_reserves_2021    |            -0.1881796|
## |uranium_reserves_2019 |            -0.1765060|
## |gas_reserves          |            -0.1693572|
## |tagsub_african        |             0.1364242|
## |taglatin              |             0.1314459|
## |tagdeveloped          |             0.1227851|
## |agri_land_rate        |            -0.1089031|
## |tageast_europe        |            -0.0180961|
##
##
## Table: Performance on all models
##
## |     |Bindepsource |Btag  |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:-----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm   |TRUE         |TRUE  |TRUE      |TRUE    | 0.7545738| 0.7519330| 0.7545404| 0.7488644|
## |lm1  |TRUE         |TRUE  |TRUE      |FALSE   | 0.4872912| 0.4861149| 0.4872319| 0.4833779|
## |lm2  |TRUE         |TRUE  |FALSE     |TRUE    | 0.7924768| 0.7909105| 0.7922676| 0.7858865|
## |lm3  |TRUE         |TRUE  |FALSE     |FALSE   | 0.5006437| 0.4994569| 0.5005965| 0.4960226|
## |lm4  |TRUE         |FALSE |TRUE      |TRUE    | 0.7492522| 0.7473281| 0.7492186| 0.7435600|
## |lm5  |TRUE         |FALSE |TRUE      |FALSE   | 0.4295078| 0.4287511| 0.4294687| 0.4279092|
## |lm6  |TRUE         |FALSE |FALSE     |TRUE    | 0.7893778| 0.7886939| 0.7893390| 0.7828918|
## |lm7  |TRUE         |FALSE |FALSE     |FALSE   | 0.4439032| 0.4396907| 0.4438812| 0.4421156|
## |lm8  |FALSE        |TRUE  |TRUE      |TRUE    | 0.2244147| 0.2233488| 0.2244073| 0.2240558|
## |lm9  |FALSE        |TRUE  |TRUE      |FALSE   | 0.3301242| 0.3291873| 0.3300746| 0.3266750|
## |lm10 |FALSE        |TRUE  |FALSE     |TRUE    | 0.2801782| 0.2798356| 0.2790848| 0.2787673|
## |lm11 |FALSE        |TRUE  |FALSE     |FALSE   | 0.3398832| 0.3393901| 0.3398533| 0.3357615|
## |lm12 |FALSE        |FALSE |TRUE      |TRUE    | 0.1713937| 0.1698180| 0.1713865| 0.1711820|
## |lm13 |FALSE        |FALSE |TRUE      |FALSE   | 0.2597227| 0.2577533| 0.2597038| 0.2587265|
## |lm14 |FALSE        |FALSE |FALSE     |TRUE    | 0.2249855| 0.2239086| 0.2245653| 0.2234740|
## |lm15 |FALSE        |FALSE |FALSE     |FALSE   | 0.2703876| 0.2703417| 0.2703692| 0.2691814|
```

```r
do_models(depen="low_carbon_share_elec")
```

```
## [1] "Models for the variable: low_carbon_share_elec"
## [1] ""
##
##
```

```
## Table: Best performing models
##
## |    |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:---|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm2 |TRUE         |TRUE |FALSE     |TRUE    | 0.7906304| 0.7895204| 0.7903835| 0.7840845|
##
##
## Table: Coefficients for best stepwise lm
##
## |                            | results$coeff_sm|
## |:---------------------------|----------------:|
## |(Intercept)                 |        6.9463657|
## |hydro_electricity           |        1.3185006|
## |gdp                         |       -0.9436698|
## |tagdeveloped                |        0.5209017|
## |tagsub_african              |        0.5022337|
## |tagmiddle_east              |       -0.4534741|
## |tageast_europe              |       -0.4272291|
## |land_area                   |       -0.3731762|
## |oil_electricity             |       -0.3273684|
## |coal_electricity            |       -0.1897854|
## |other_renewable_electricity |        0.1577925|
## |taglatin                    |        0.1302026|
## |wind_electricity            |        0.1226428|
## |solar_electricity           |        0.1193776|
## |gas_electricity             |       -0.1143774|
## |nuclear_electricity         |        0.0893336|
## |year                        |       -0.0482198|
## |coal_reserves_2021          |       -0.0400540|
##
##
## Table: Best performing models with only external data
##
## |     |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm11 |FALSE        |TRUE |FALSE     |FALSE   | 0.3426829| 0.3403904| 0.3426269| 0.3391884|
##
##
## Table: Coefficients for best stepwise lm with only external data
##
## |                     | results_nind$coeff_sm|
## |:--------------------|---------------------:|
## |population           |             0.7842394|
## |land_area            |             0.6182214|
## |hdi                  |             0.5718938|
## |gdp                  |            -0.4473895|
## |(Intercept)          |            -0.3187620|
## |gas_reserves         |            -0.2536834|
## |tagmiddle_east       |            -0.2045186|
## |tagdeveloped         |             0.1984907|
## |coal_reserves_2021   |            -0.1917409|
## |urbaniz_rate         |            -0.1619410|
## |tageast_europe       |             0.1215758|
## |taglatin             |             0.1169517|
```

```
## |agri_land_rate        |            -0.1131481|
## |tagsub_african        |             0.1131203|
## |uranium_reserves_2019 |            -0.1097553|
##
##
## Table: Performance on all models
##
## |     |Bindepsource |Btag  |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:-----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm   |TRUE         |TRUE  |TRUE      |TRUE    | 0.7543023| 0.7522752| 0.7542695| 0.7487435|
## |lm1  |TRUE         |TRUE  |TRUE      |FALSE   | 0.5033493| 0.5019087| 0.5033210| 0.5002793|
## |lm2  |TRUE         |TRUE  |FALSE     |TRUE    | 0.7906304| 0.7895204| 0.7903835| 0.7840845|
## |lm3  |TRUE         |TRUE  |FALSE     |FALSE   | 0.5183225| 0.5164649| 0.5182896| 0.5144857|
## |lm4  |TRUE         |FALSE |TRUE      |TRUE    | 0.7499102| 0.7479253| 0.7498747| 0.7444595|
## |lm5  |TRUE         |FALSE |TRUE      |FALSE   | 0.4460168| 0.4441488| 0.4459505| 0.4443492|
## |lm6  |TRUE         |FALSE |FALSE     |TRUE    | 0.7877550| 0.7864344| 0.7875890| 0.7814279|
## |lm7  |TRUE         |FALSE |FALSE     |FALSE   | 0.4612759| 0.4582377| 0.4612499| 0.4594244|
## |lm8  |FALSE        |TRUE  |TRUE      |TRUE    | 0.2494761| 0.2489060| 0.2494680| 0.2490676|
## |lm9  |FALSE        |TRUE  |TRUE      |FALSE   | 0.3301761| 0.3296410| 0.3301557| 0.3274576|
## |lm10 |FALSE        |TRUE  |FALSE     |TRUE    | 0.3037926| 0.3034158| 0.3034522| 0.3022777|
## |lm11 |FALSE        |TRUE  |FALSE     |FALSE   | 0.3426829| 0.3403904| 0.3426269| 0.3391884|
## |lm12 |FALSE        |FALSE |TRUE      |TRUE    | 0.1895932| 0.1885575| 0.1893929| 0.1893335|
## |lm13 |FALSE        |FALSE |TRUE      |FALSE   | 0.2557127| 0.2533985| 0.2556604| 0.2544855|
## |lm14 |FALSE        |FALSE |FALSE     |TRUE    | 0.2449189| 0.2407402| 0.2446466| 0.2431050|
## |lm15 |FALSE        |FALSE |FALSE     |FALSE   | 0.2712288| 0.2683122| 0.2712108| 0.2697490|
```

```
do_models(depen="fossil_share_elec")
```

```
## [1] "Models for the variable: fossil_share_elec"
## [1] ""
##
##
## Table: Best performing models
##
## |    |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:---|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm2 |TRUE         |TRUE |FALSE     |TRUE    | 0.7906306| 0.7895206| 0.7903837| 0.7840847|
##
##
## Table: Coefficients for best stepwise lm
##
## |                            | results$coeff_sm|
## |:---------------------------|----------------:|
## |(Intercept)                 |       -6.9463806|
## |hydro_electricity           |       -1.3185018|
## |gdp                         |        0.9436708|
## |tagdeveloped                |       -0.5208989|
## |tagsub_african              |       -0.5022372|
## |tagmiddle_east              |        0.4534814|
## |tageast_europe              |        0.4272278|
## |land_area                   |        0.3731771|
## |oil_electricity             |        0.3273681|
## |coal_electricity            |        0.1897851|
## |other_renewable_electricity |       -0.1577927|
```

```
## |taglatin                    |       -0.1302043|
## |wind_electricity            |       -0.1226431|
## |solar_electricity           |       -0.1193779|
## |gas_electricity             |        0.1143771|
## |nuclear_electricity         |       -0.0893335|
## |year                        |        0.0482195|
## |coal_reserves_2021          |        0.0400541|
##
##
## Table: Best performing models with only external data
##
## |     |Bindepsource |Btag |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm11 |FALSE        |TRUE |FALSE     |FALSE   | 0.3426829| 0.3403904| 0.3426652| 0.3391884|
##
##
## Table: Coefficients for best stepwise lm with only external data
##
## |                      | results_nind$coeff_sm|
## |:---------------------|---------------------:|
## |(Intercept)           |             1.3187621|
## |population            |            -0.7842394|
## |land_area             |            -0.6182213|
## |hdi                   |            -0.5718936|
## |gdp                   |             0.4473892|
## |gas_reserves          |             0.2536834|
## |tagmiddle_east        |             0.2045187|
## |tagdeveloped          |            -0.1984907|
## |coal_reserves_2021    |             0.1917409|
## |urbaniz_rate          |             0.1619410|
## |tageast_europe        |            -0.1215758|
## |taglatin              |            -0.1169517|
## |agri_land_rate        |             0.1131481|
## |tagsub_african        |            -0.1131203|
## |uranium_reserves_2019 |             0.1097554|
##
##
## Table: Performance on all models
##
## |     |Bindepsource |Btag  |Byearwise |Blogify |        lm|        sm|     lasso|     ridge|
## |:----|:------------|:-----|:---------|:-------|---------:|---------:|---------:|---------:|
## |lm   |TRUE         |TRUE  |TRUE      |TRUE    | 0.7543024| 0.7522754| 0.7542618| 0.7487437|
## |lm1  |TRUE         |TRUE  |TRUE      |FALSE   | 0.5033493| 0.5019087| 0.5033091| 0.5002794|
## |lm2  |TRUE         |TRUE  |FALSE     |TRUE    | 0.7906306| 0.7895206| 0.7903837| 0.7840847|
## |lm3  |TRUE         |TRUE  |FALSE     |FALSE   | 0.5183225| 0.5164649| 0.5182946| 0.5144857|
## |lm4  |TRUE         |FALSE |TRUE      |TRUE    | 0.7499103| 0.7479254| 0.7498748| 0.7444596|
## |lm5  |TRUE         |FALSE |TRUE      |FALSE   | 0.4460167| 0.4441488| 0.4459612| 0.4443492|
## |lm6  |TRUE         |FALSE |FALSE     |TRUE    | 0.7877552| 0.7864346| 0.7876923| 0.7814281|
## |lm7  |TRUE         |FALSE |FALSE     |FALSE   | 0.4612759| 0.4582377| 0.4612499| 0.4594244|
## |lm8  |FALSE        |TRUE  |TRUE      |TRUE    | 0.2494764| 0.2489062| 0.2494682| 0.2490678|
## |lm9  |FALSE        |TRUE  |TRUE      |FALSE   | 0.3301761| 0.3296410| 0.3301593| 0.3274576|
## |lm10 |FALSE        |TRUE  |FALSE     |TRUE    | 0.3037929| 0.3034161| 0.3033989| 0.3022780|
## |lm11 |FALSE        |TRUE  |FALSE     |FALSE   | 0.3426829| 0.3403904| 0.3426652| 0.3391884|
## |lm12 |FALSE        |FALSE |TRUE      |TRUE    | 0.1895932| 0.1885576| 0.1895423| 0.1893336|
```

```
## |lm13 |FALSE        |FALSE |TRUE      |FALSE   | 0.2557127| 0.2533985| 0.2556969| 0.2544855|
## |lm14 |FALSE        |FALSE |FALSE     |TRUE    | 0.2449190| 0.2407403| 0.2448660| 0.2431050|
## |lm15 |FALSE        |FALSE |FALSE     |FALSE   | 0.2712288| 0.2683122| 0.2712108| 0.2697490|
```

Renewables, LC and fossil are grouped, since the share of LC is equal to 1 minus the share of fossil, and LC and renewables' shares are very similar, due to the small impact of nuclear in the mix. The following statements regard the LC model:

The best model has R square of ~0.79, which is low with respect to our expectations, considering it has the variables for electricity from each source; it returns trivial coefficients, with a high impact of hydro (making up most of the LC electricity) and also fossil fuel sources on the opposite end. GDP also has a high coefficient, and HDI is not in the model.

The model without sources only has an R squared of ~0.34, but it is much more interesting, being highly impacted by population and land area, confirming that smaller countries are the ones that rely mostly on fossil fuels, probably because of smaller infrastructure investments and research costs.

Reserves also have quite high coefficients, as countries with fossil fuel reserves are less incentivized to invest in clean electricity sources.

Sitography

[1] https://github.com/owid/energy-data

[2] https://data.worldbank.org/indicator/NY.GDP.MKTP.KD

[3] https://ourworldindata.org/grapher/land-area-km

[4] https://ourworldindata.org/grapher/share-of-land-area-used-for-agriculture

[5] https://ourworldindata.org/grapher/share-of-population-urban

[6] https://ourworldindata.org/grapher/human-development-index

[7] https://ourworldindata.org/grapher/death-rates-from-air-pollution

[8] https://ourworldindata.org/grapher/coal-proved-reserves

[9] https://ourworldindata.org/grapher/oil-proved-reserves

[10] https://www.oecd.org/publications/uranium-20725310.htm

[11] https://ourworldindata.org/grapher/natural-gas-proved-reserves

[12] https://en.wikipedia.org/wiki/Low-carbon_power

[13] https://www.andritz.com/hydro-en/hydronews/hydro-news-asia/laos

[14] https://en.wikipedia.org/wiki/List_of_largest_hydroelectric_power_stations

[15] https://www.andritz.com/hydro-en/hydronews/hydropower-africa/democratic-rep-congo

[16] https://www.imf.org/en/Publications/fandd/issues/2022/12/country-case-kenya-taps-the-earth-heat