Titolo provvisorio

Rocca Valerio, Virginio Giacomo

2023-05-18

TODO list

1) Abstract
2) Obtaining data
3) Data cleaning and filtering (including pre-processing)
4) Exploratory analysis
5) Descriptive analysis
6) Data modeling (RICORDATI DI CAMBIARE OIL DI CIA CON OWID)
7) Usare report Pietro come riferimento
8) Check errori: low-carbon col trattino, aggiungere "Figure x".

Contents

Abstract

This is the abstract

Obtaining data

We could not find a single dataset containing all the information of interest. Thus, the project's first step was 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.

```r
# Import the Energy dataset
main = read.csv("C://Users//valer//Documents//GitHub//Project_Statistical_Learning//datasets//Total_ene
```

We then merged 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 (CHECK BC THE LINK HAS ONLY 2020);
- *Uranium proved reserves* by OECD [11], which contains uranium reserves in each country in 2019, measured in tonnes.

Merging presents three critical issues, which are 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 *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). 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. We modify the structure of *GDP* to fit the others'.

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

# Import the GDP dataset
gdp = read_excel("C://Users//valer//Documents//GitHub//Project_Statistical_Learning//datasets//gdp_cons
# 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)
```

```r
# Import and merging of country area
country_areas = read.csv("C://Users//valer//Documents//GitHub//Project_Statistical_Learning//datasets//c
main = left_join(main, select(country_areas, -c("Entity")),
                 by = c("iso_code" = "Code", "year" = "Year"))
rm(country_areas)

# Import and merging of agricultural share area
agri_share = read.csv("C://Users//valer//Documents//GitHub//Project_Statistical_Learning//datasets//sha
main = left_join(main, select(agri_share, -c("Entity")),
                 by = c("iso_code" = "Code", "year" = "Year"))
rm(agri_share)

# Import and merging of urbanization rate
urbanization_rate = read.csv("C://Users//valer//Documents//GitHub//Project_Statistical_Learning//datase
main = left_join(main, select(urbanization_rate, -c("Entity")),
                 by = c("iso_code" = "Code", "year" = "Year"))
rm(urbanization_rate)

# Import and merging of HDI
hdi = read.csv("C://Users//valer//Documents//GitHub//Project_Statistical_Learning//datasets//human_devel
main = left_join(main, select(hdi, -c("Entity")),
                 by=c("iso_code" = "Code", "year" = "Year"))
rm(hdi)

# Import, selection and merging of deaths from air pollution
death_rates = read.csv("C://Users//valer//Documents//GitHub//Project_Statistical_Learning//datasets//dea
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, filtering and merging of coal proved reserves
coal_res = read.csv("C://Users//valer//Documents//GitHub//Project_Statistical_Learning//datasets//coal_p
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 of oil proved reserves
oil_res = read.csv("C://Users//valer//Documents//GitHub//Project_Statistical_Learning//datasets//oil_pr
main = left_join(main, select(oil_res, -c("Entity")),
                 by = c("iso_code" = "Code", "year" = "Year"))
rm(oil_res)

# Import and merging of natural gas reserves
gas_res = read.csv("C://Users//valer//Documents//GitHub//Project_Statistical_Learning//datasets//natural
main = left_join(main, select(gas_res, -c("Entity",)),
                 by = c("iso_code" = "Code", "year" = "Year"))
rm(gas_res)

# Import and merging of uranium reserves
uranium_res = read.csv("C://Users//valer//Documents//GitHub//Project_Statistical_Learning//datasets//ura
```

```r
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)
```

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: Anctartica 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.

```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",
            "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",
```

```r
            "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",
            "biofuel_consumption",
            "biofuel_electricity",
            "biofuel_share_elec",
            "biofuel_share_energy"))

# 3. Feature renaming
# Print of the names of the features
print(colnames(main[,37:46]))
```

```
##  [1] "gdp.y"
##  [2] "Land.area..sq..km."
##  [3] "Agricultural.land....of.land.area."
##  [4] "Urban.population....of.total.population."
##  [5] "Human.Development.Index"
##  [6] "particulate_pollution"
##  [7] "coal_reserves_2021"
##  [8] "Oil...Proved.reserves"
##  [9] "gas_reserves"
## [10] "uranium_reserves_2019"
```

```r
# Proper feature renaming
colnames(main) = c(colnames(main[,1:36]), "gdp", "land_area", "agri_land_rate",
                   "urbaniz_rate", "hdi", "particulate_pollution",
                   "coal_reserves_2021", "oil_reserves_2020", "gas_reserves",
                   "uranium_reserves_2019")
```

4. Descriptive analyses

We start the descriptive analyses by looking at the general renewable and 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], i.e., to the renewable and nuclear electricities. From now onward, we will refer to it with the acronym "LC."

```r
# 1. World electricity generation from low carbon 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
```

```r
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 = "Figure 1. World electricity generation from low carbon 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 = "Figure 2. % of world electricity generated from LC sources",
       x = "Year",
       y = "% of terawatt-hours")

grid.arrange(gg1,gg2)
```

Figure 1. World electricity generation from low carbon sources

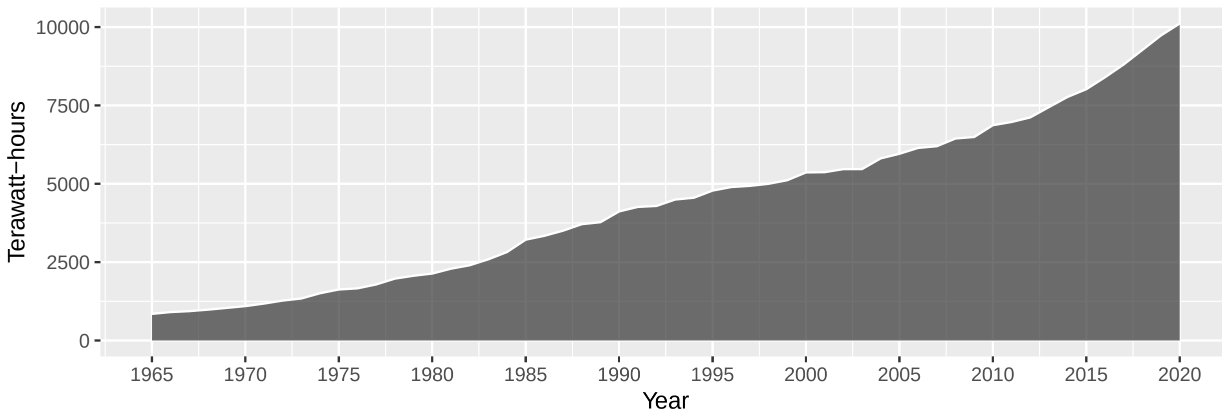
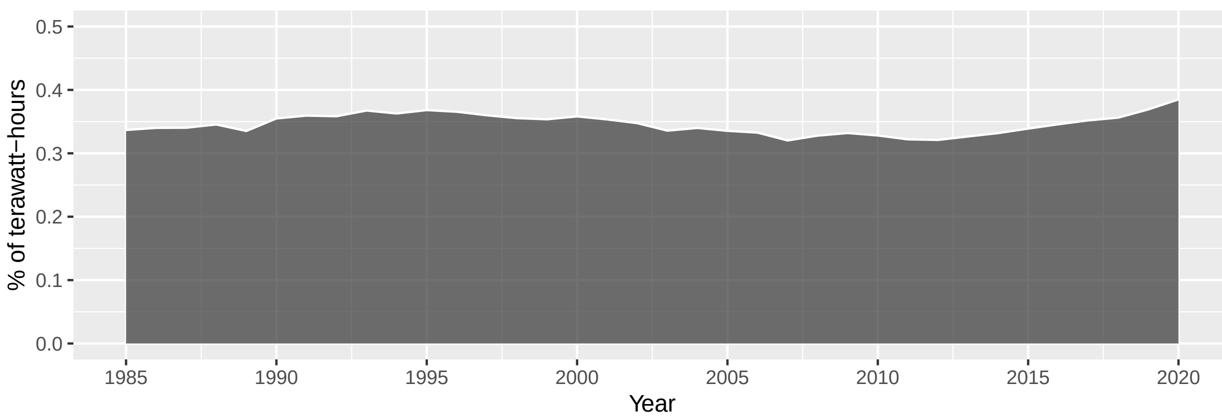


Figure 2. % of world electricity generated from LC sources



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%. So let us now break down data by country.

As it is clear from *Figure 2*, a higher production

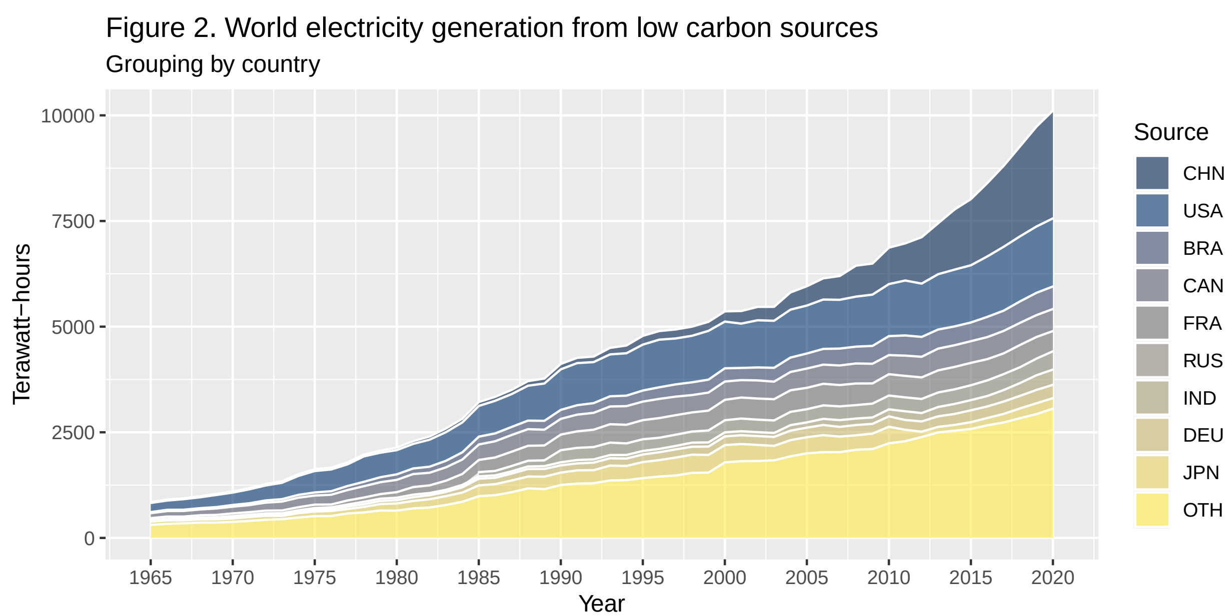
```
# 2. World electricity generation from low carbon sources, grouped by countries

# a. Create a vector containing the ISO codes of the nine countries with the highest LC electricity pro
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))
```

```
## `summarise()` has grouped output by 'iso_code'. You can override using the
```

```
## `.groups` argument.
```

```
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 = "Figure 2. World electricity generation from low carbon sources",
       subtitle = "Grouping 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)
colnames(place) = c(colnames(place[,1:2]),"perc_lc")


gg1
```

```
## Warning: Removed 598 rows containing missing values (position_stack).
```




```
place[,c(1,3)] %>% arrange(desc(perc_lc)) %>% as_tibble()
```

```
## # A tibble: 10 x 2
##    iso_code perc_lc
##    <fct>      <dbl>
##  1 OTH         0.3
```

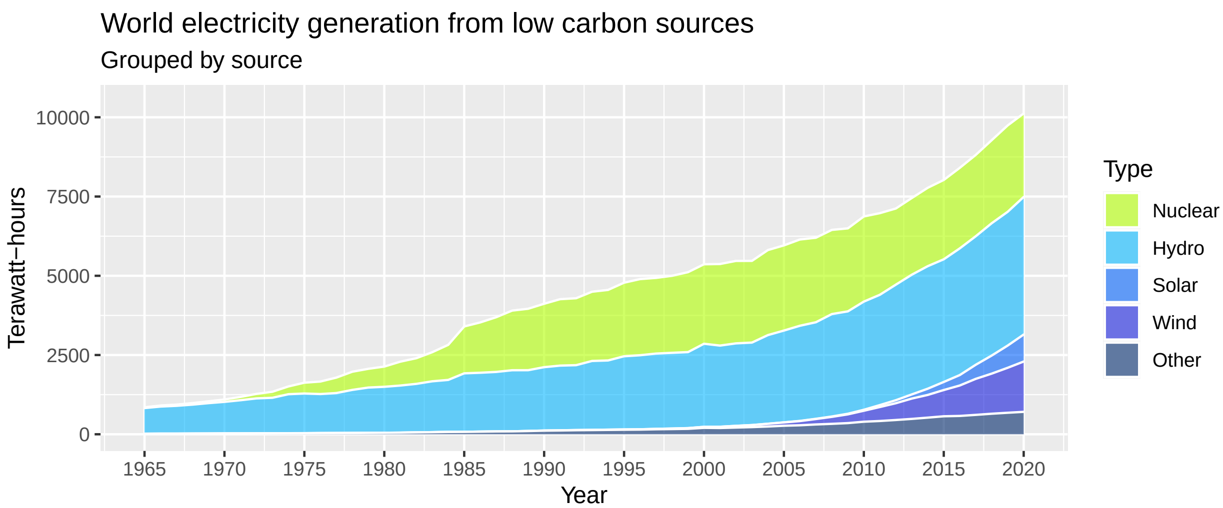
```
##   2 CHN         0.25
##   3 USA         0.16
##   4 BRA         0.05
##   5 CAN         0.05
##   6 FRA         0.05
##   7 IND         0.04
##   8 RUS         0.04
##   9 DEU         0.03
## 10 JPN         0.02
```

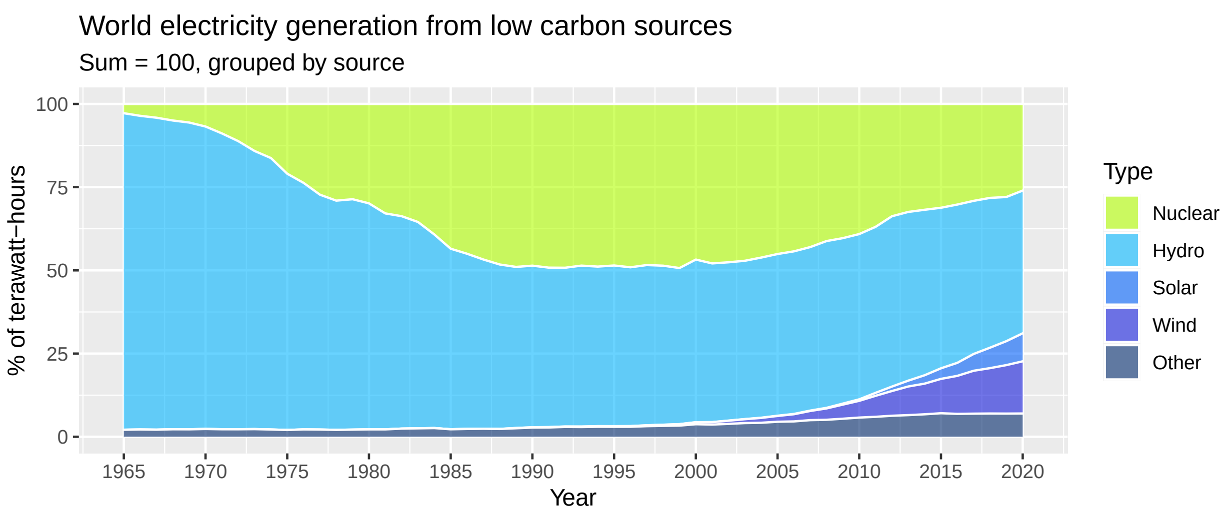
Not surprisingly, the generation of electricity from LC sources is not homogeneous between the countries: China and the USA own 41% of the electricity generated in 2020; the nine nations with the highest electricity generation from LC sources account for 69%.

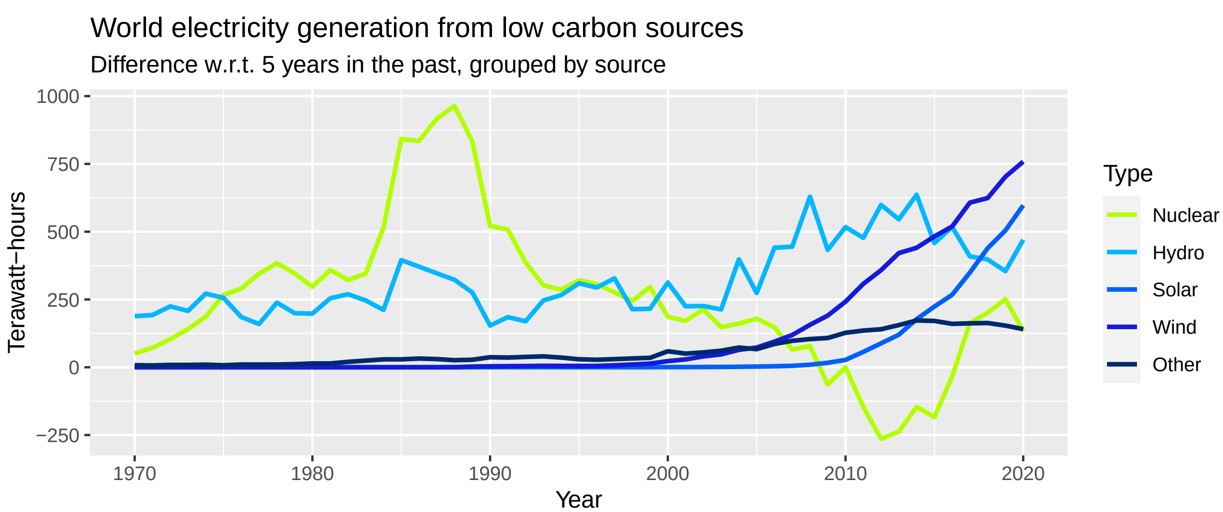
```
# 3. World electricity generation from low carbon sources, grouped by source

# 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)) +
  #xlim(1975,2020)??
  scale_y_continuous(limits = c(0,10500)) +
  scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
  labs(title = "World electricity generation from low carbon sources",
       subtitle = "Grouped by source",
       x = "Year",
       y = "Terawatt-hours")

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

```r
            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"))

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 low carbon 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)
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)) +
```

```r
  scale_y_continuous() +
  scale_color_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
    labs(title = "World electricity generation from low carbon sources",
       subtitle = "Difference w.r.t. 5 years in the past, grouped by source",
       x = "Year",
       y = "Terawatt-hours")

grid.arrange(gg1, gg2, gg3)
```







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

1. **Dawn of LC 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 declined and gave way to renewables. In particular, the solar and wind generation skyrocketed.

We are now interested in analizying the generation of electricity by LC sources in different areas. To do so, we aggregate world countries in six groups on the basis of geographical, economical 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 Caribbeans**: North and South America's countries, except of 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.

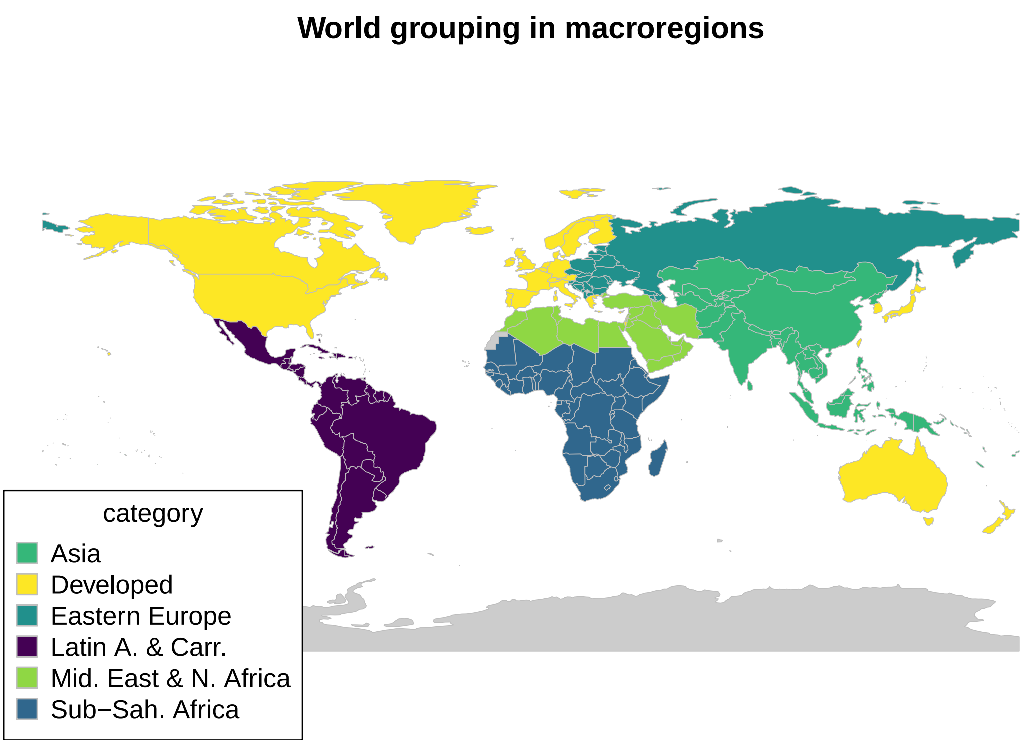
```
# a. Create a vector for each group, containing the ISO-codes
developed_countries = c("AUS", "AUT", "BEL", "CAN", "CYP", "DNK", "FRO", "FIN",
                        "FRA", "DEU", "GRC", "GRL", "HKG", "ISL", "IRL", "ISR",
                        "ITA", "JPN", "LUX", "MAC", "MLT", "NLD", "NZL", "NOR",
                        "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")

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

```r
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)
```

```r
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
```

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




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

# a. Scatter plot %2010 vs %2020
# Creation of the dataset
place = select(main, year, low_carbon_electricity, renewables_electricity,
               electricity_generation) %>%
  mutate_all(~replace_na(.,0)) %>%
  cbind(., tag = main$tag) %>%
  filter(year == 2010 | year == 2020 ) %>%
```

```r
  gather(key = "type", value = "lc_generation", -year, -electricity_generation, -tag) %>%
  group_by(year, tag, type) %>%
  summarize(perc_elec_from_lc = sum(lc_generation)*100 /sum(electricity_generation)) %>%
  spread(key = year, value = perc_elec_from_lc)
```

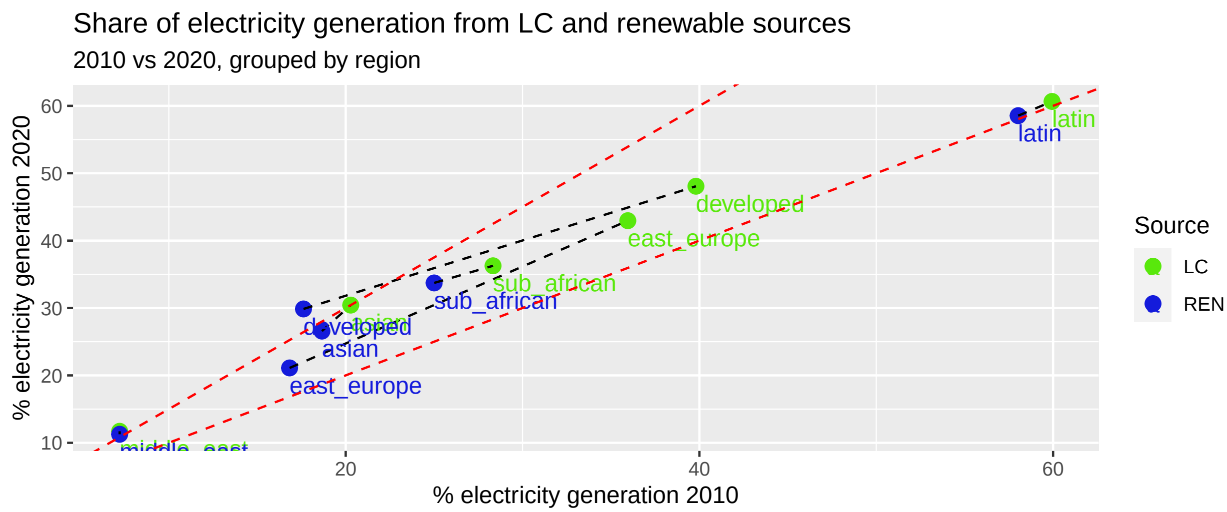
```
## `summarise()` has grouped output by 'year', 'tag'. You can override using the
## `.groups` argument.
```

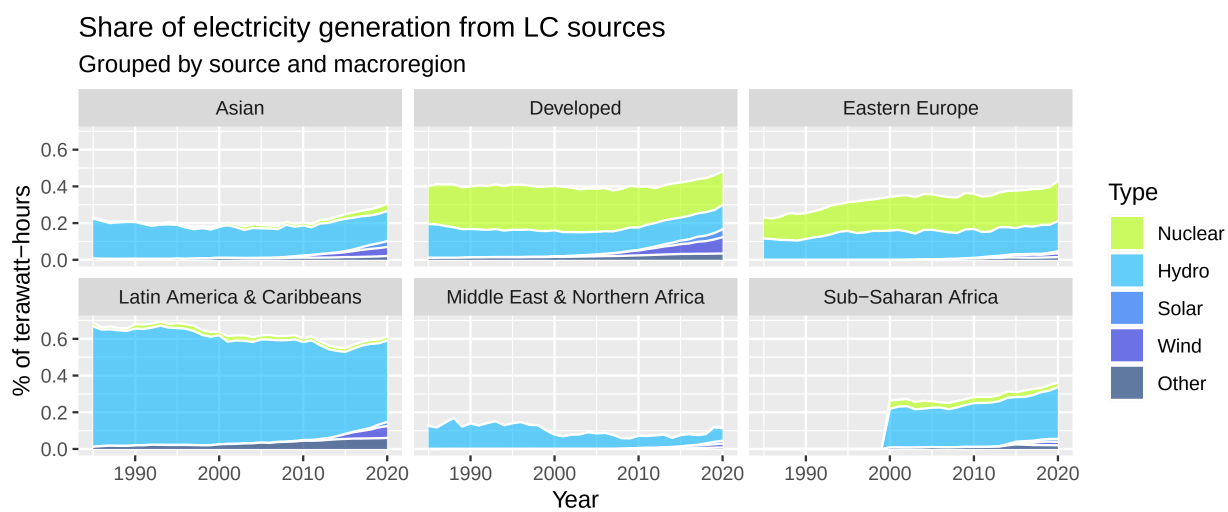
```r
colnames(place) = c("tag", "Source", "ten", "twenty")

place[place$Source  == "low_carbon_electricity", "Source"] = "LC"
place[place$Source  == "renewables_electricity", "Source"] = "REN"

# Creation of the plot
gg1 = ggplot(place, aes(ten, twenty, col = Source)) +
  geom_point(size = 3) +
  geom_line(aes(group = tag), color = "black", linetype = "dashed") +
  geom_text(aes(label = tag), hjust = 0, vjust = 1.5) +
  scale_color_manual(values = c("#59E80C", "#141BDB")) +
  labs(title = "Share of electricity generation from LC and renewable sources",
       subtitle = "2010 vs 2020, grouped by region",
       x = "% electricity generation 2010",
       y = "% electricity generation 2020") +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  geom_abline(intercept = 0, slope = 1.5, linetype = "dashed", color = "red")

# b. LC generation by region
# 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",
         -year, -tag)
```

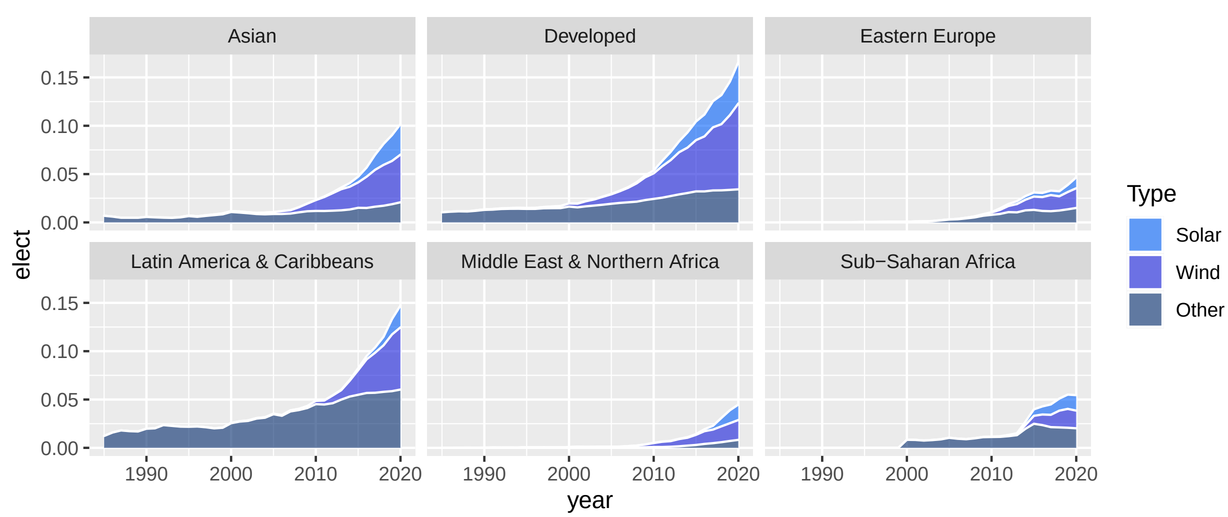
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```r
place$Type = factor(place$Type, levels = c("Nuclear", "Hydro", "Solar",
                                           "Wind", "Other"))
for(i in 1:nrow(place)){
  if(place$year[i] < 2000 & place$tag[i] == "sub_african"){
    place$elect[i] = 0
  }
```

```r
}

# Creation of the plot
place[place$tag  == "asian", "tag"] = "Asian"
place[place$tag  == "developed", "tag"] = "Developed"
place[place$tag  == "east_europe", "tag"] = "Eastern Europe"
place[place$tag  == "latin", "tag"] = "Latin America & Caribbeans"
place[place$tag  == "middle_east", "tag"] = "Middle East & Northern Africa"
place[place$tag  == "sub_african", "tag"] = "Sub-Saharan Africa"

gg2 = ggplot(place, aes(year, elect, fill = Type)) +
  geom_area(alpha=0.6 , size=.5, colour="white")+
  scale_x_continuous(limits = c(1985,2020)) +
  #xlim(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")

# c. focus on renewables excluding hydroelectric
# Creation of the dataset
place = filter(place, Type == "Solar" | Type == "Wind" | Type == "Other")

# Creation of the plot
gg3 = ggplot(place, aes(year, elect, fill = Type)) +
  geom_area(alpha=0.6 , size=.5, colour="white")+
  scale_x_continuous(limits = c(1985,2020)) +
  #xlim(1985,2020)??
  scale_fill_manual(values = c("#0060FA", "#141BDB", "#00296B")) +
  facet_wrap(~ tag, nrow = 2)


grid.arrange(gg1, gg2, gg3, 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 low-carbon sources: Latin American countries have a more significant share; developed, Eastern European and Sub-Saharan and Asian follow; Middle-East generation is negligible.

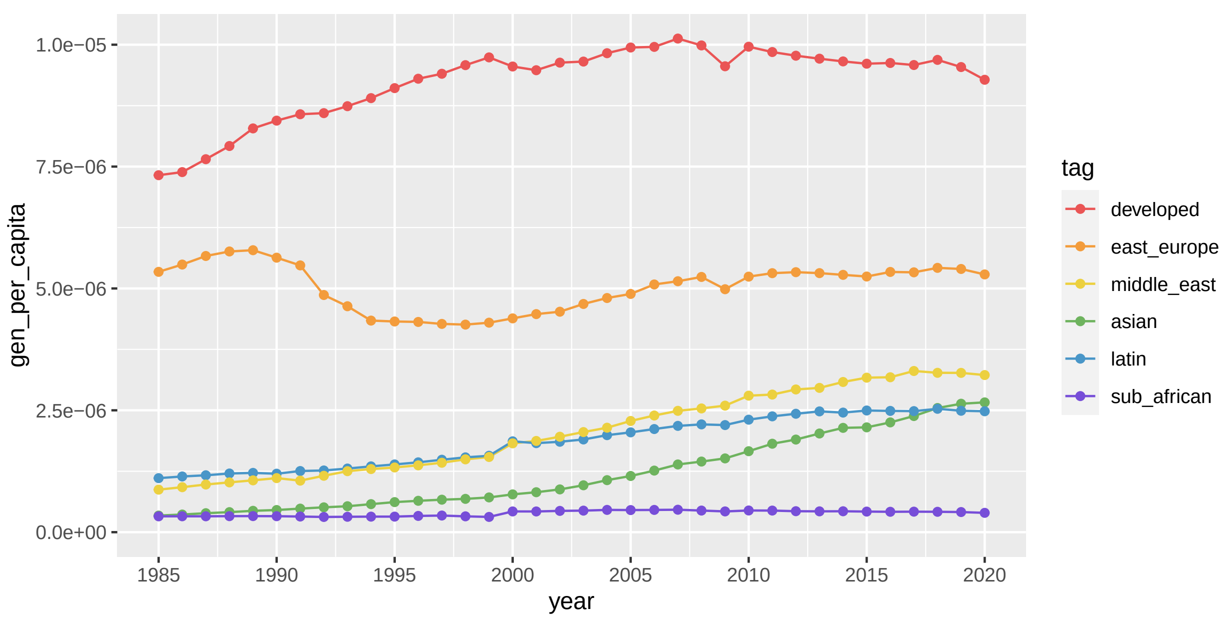
2. Hydropower is an essential source of electricity in all the considered regions.
3. Nuclear electricity is significant only in developed countries and Eastern Europe; its importance is comparable to hydropower generation in those regions.
4. Developed countries drive non-hydropower renewable production. Nonetheless, it is also true that those sources are also rapidly becoming more relevant in Latin America and Asia.

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

```
# d. Focus on 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))
```

```
## `summarise()` has grouped output by 'tag'. You can override using the `.groups`
## argument.
```

```
place$tag = factor(place$tag, levels = c("developed", "east_europe",
                                         "middle_east", "asian", "latin",
                                         "sub_african"))

# Creation of the plot
ggplot(place, aes(year, gen_per_capita, color = tag)) +
  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"))
```

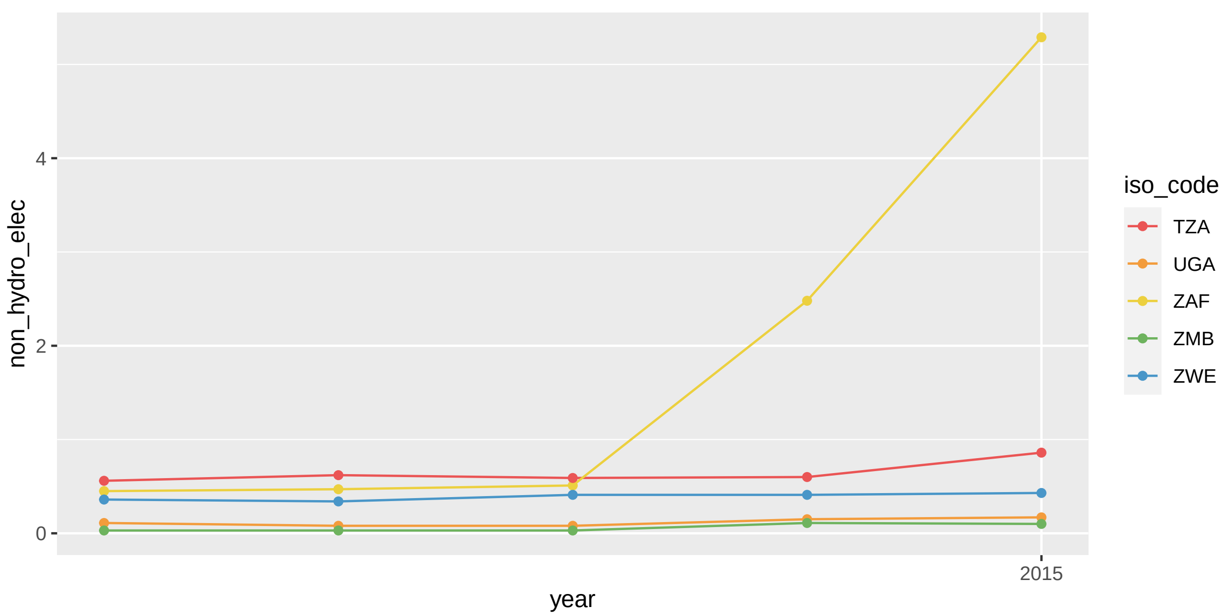


The electricity generation from non-hydro renewable sources in Sub-Saharan Africa increases rapidly between 2011 and 2015. Therefore, we studied the behavior of the five countries in the region with the highest non-hydro renewable electricity generation in 2015. As the plot shows, the steep increase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

#a. Extract ISO of 5 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()
```

```
## Selecting by iso_code
```

```
# b. Create 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)

# c. Create the plot
ggplot(place, aes(year, non_hydro_elec, color = iso_code)) +
  geom_line() +
  geom_point() +
  scale_x_continuous(limits = c(2011,2015), breaks = seq(1985, 2020, by = 5)) +
  scale_color_manual(values = c("#EA5555", "#F39C3C", "#ECD03F", "#6EB35E", "#4996C8"))
```



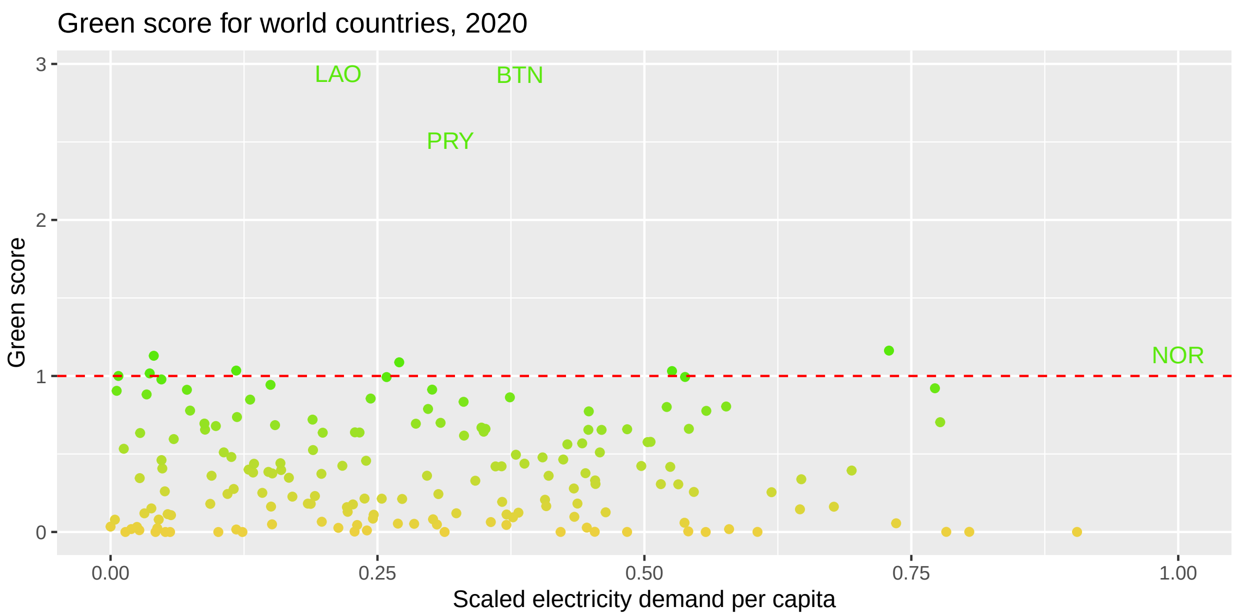
Green score - Focus on source

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 created a Green Score, defined as the following ratio:

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

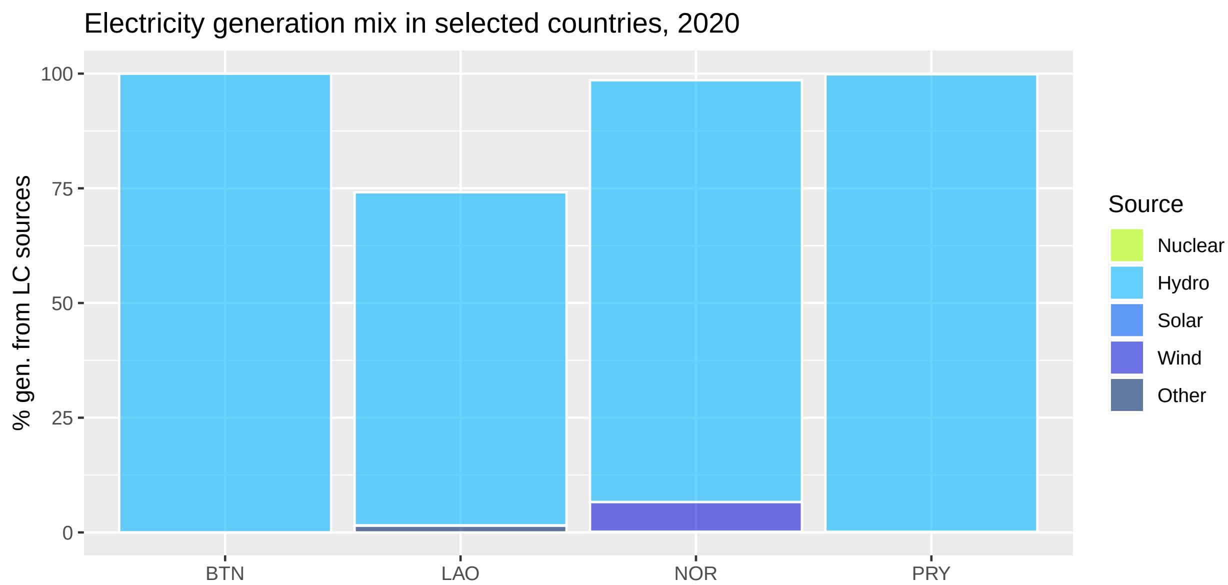
Note that in this section, descriptive analyses, we removed the countries with a smaller population than 500 000. The reason is that we expect those countries to be too small to have a significant independent electricity policy with respect to their neighbors.

```r
# Create the dataset
# Note. x is computed as the squared root of the electricity demand per capita, scaled in the interval
place = filter(main, (year == 2020 & population > 500000 & iso_code != "REU")) %>%
  transform(elec_demand_per_capita = (sqrt(electricity_demand / population) - min(sqrt(electricity_dema
         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)

# Create 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 for 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.

```
# Create 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)

place[place$Source  == "nuclear_share_elec", "Source"] = "Nuclear"
place[place$Source  == "hydro_share_elec", "Source"] = "Hydro"
place[place$Source  == "solar_share_elec", "Source"] = "Solar"
place[place$Source  == "wind_share_elec", "Source"] = "Wind"
place[place$Source  == "other_renewables_share_elec", "Source"] = "Other"

place$Source = factor(place$Source, levels = c("Nuclear", "Hydro", "Solar", "Wind",
                                               "Other"))

# Create 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")
```



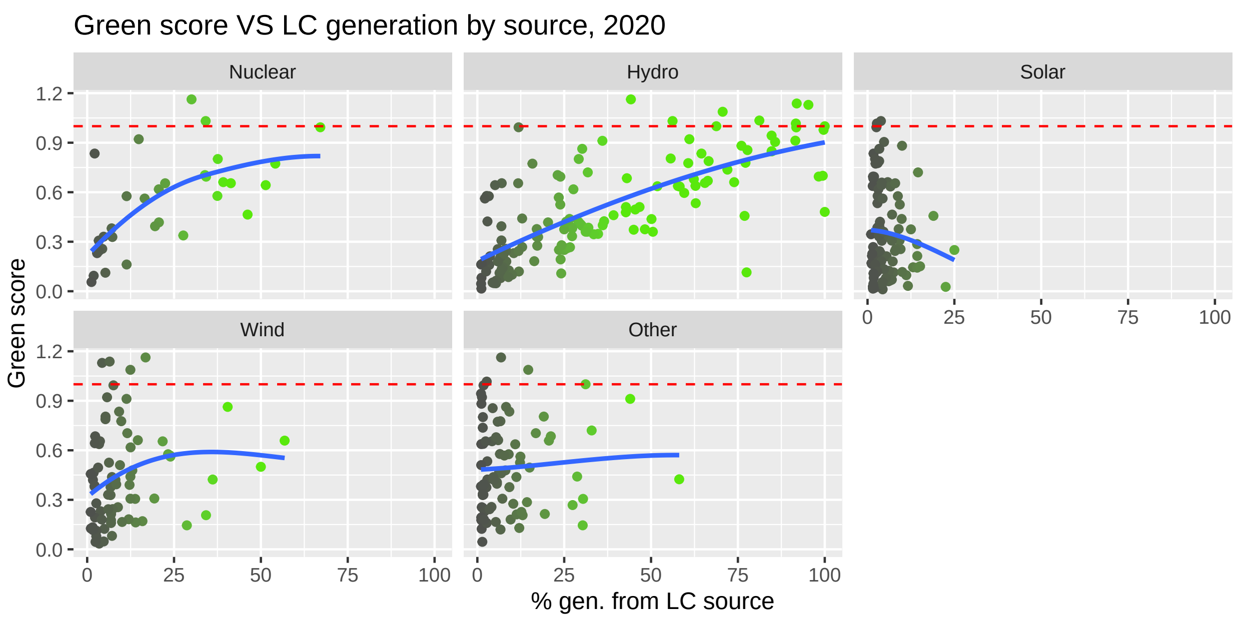
The high performance of the countries is due to a dominant hydroelectric generation. For example, Laos aims to become the "Battery of Southeast Asia" by further exploiting its impressive hydropower potential [13]. So those countries can achieve such an impressive result because of a resource not available 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 [14]), but exploit 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.

```
# Create 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[place$Source  == "nuclear_share_elec", "Source"] = "Nuclear"
place[place$Source  == "hydro_share_elec", "Source"] = "Hydro"
place[place$Source  == "solar_share_elec", "Source"] = "Solar"
place[place$Source  == "wind_share_elec", "Source"] = "Wind"
place[place$Source  == "other_renewables_share_elec", "Source"] = "Other"

place$Source = factor(place$Source, levels = c("Nuclear", "Hydro", "Solar", "Wind",
                                               "Other"))

# Create 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) +
```

```r
  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 LC generation by source, 2020",
       x = "% gen. from LC source",
       y = "Green score")
```

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



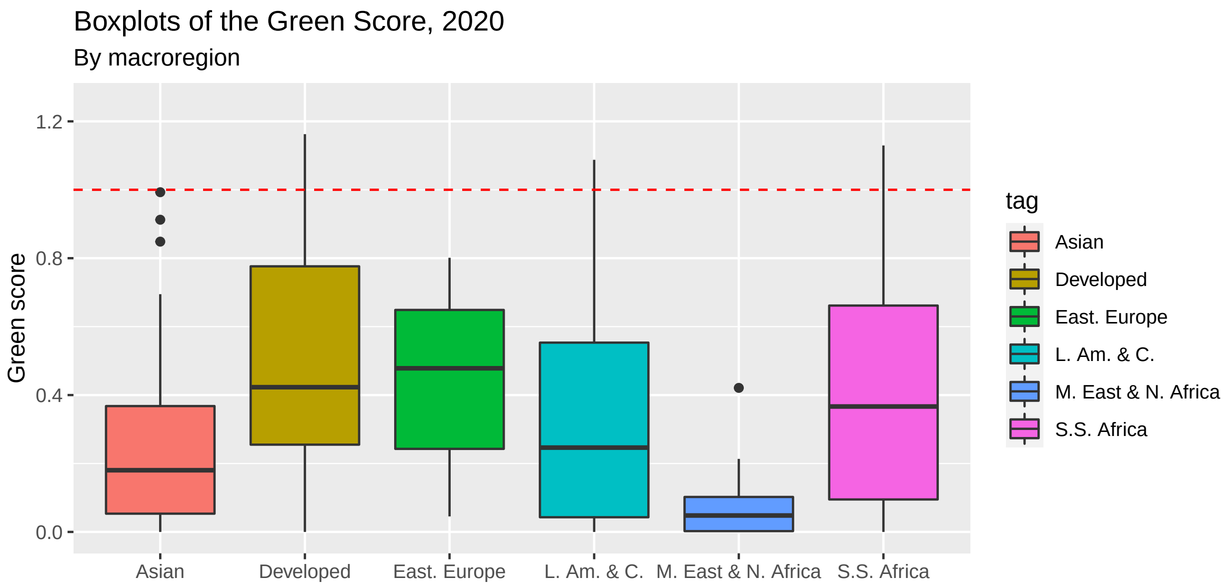
The plot confirms the correlation between the green score and the electricity generation from hydropower. It also shows an important link with nuclear power production but not with solar, wind, and other sources.

Green score - Focus on macroregions

In this final section of descriptive analyses we focus on the analysis of the green score in each country.

```r
# 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"

# Creation of the plot
ggplot(place, aes(x = tag, y = green_score_lc)) +
```

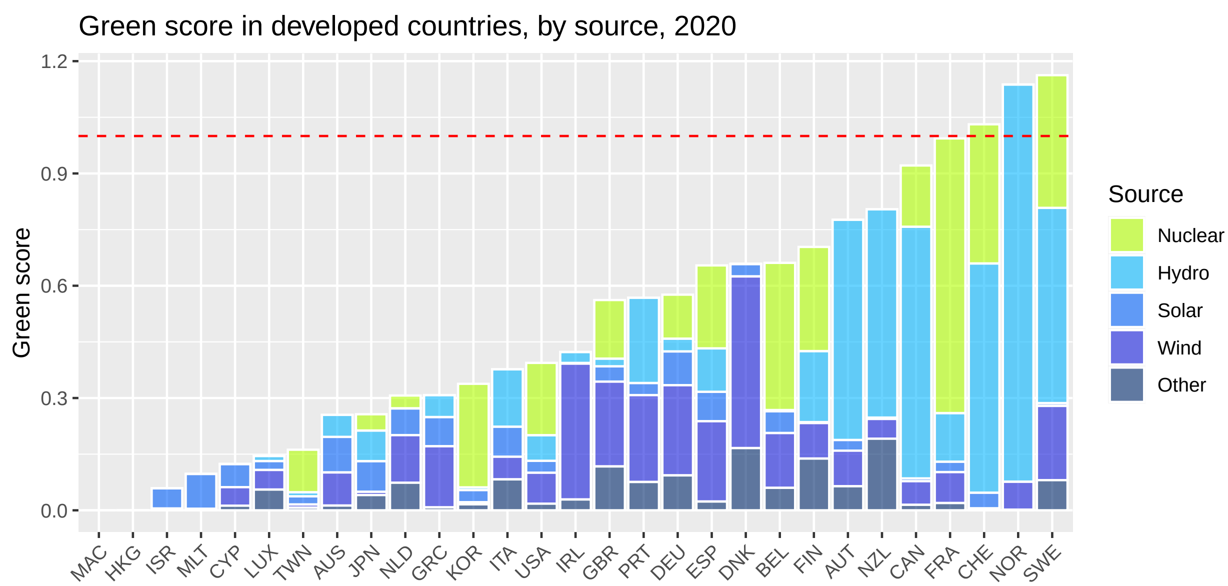
```r
  geom_boxplot(aes(fill = tag)) +
  ylim(0,1.25) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
    labs(title = "Boxplots of the Green Score, 2020",
         subtitle = "By macroregion",
       x = "",
       y = "Green score")
```



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 the countries of each area separately.

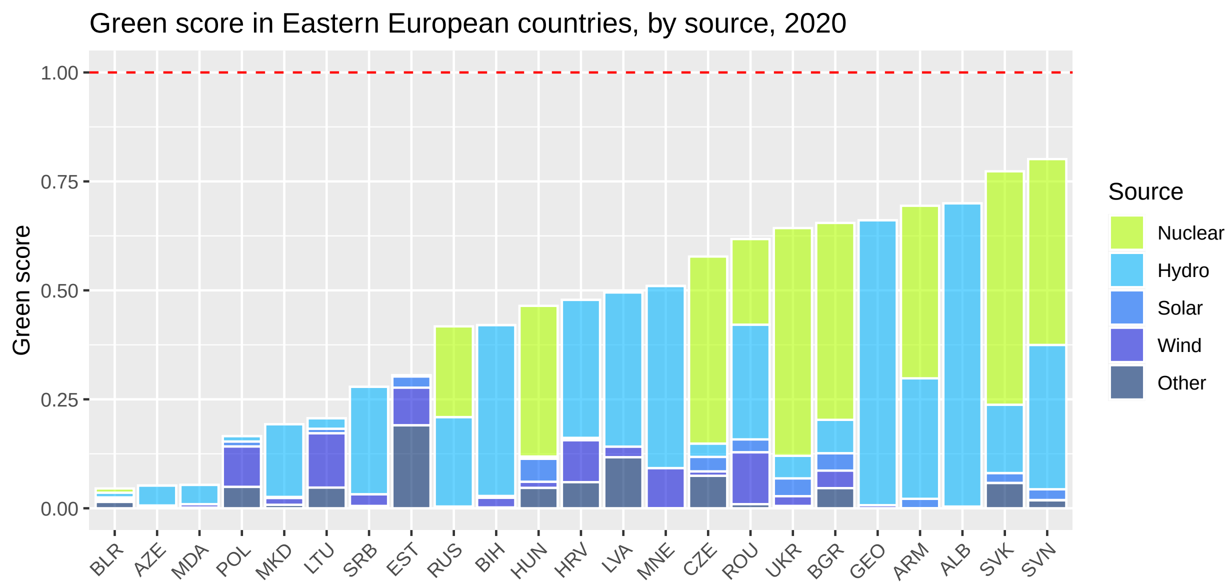
```r
# Creation of the dataset
place = filter(main, year == 2020, !is.na(other_renewables_share_elec),
               population >= 500000) %>%
  transform(ratio = (low_carbon_electricity / (electricity_demand * low_carbon_share_elec))) %>%
  transform(green_solar = ratio * solar_share_elec,
            green_wind = ratio * wind_share_elec,
            green_hydro = ratio * hydro_share_elec,
            green_nuclear = ratio * nuclear_share_elec,
            green_other = ratio * other_renewables_share_elec) %>%
  select(iso_code, tag, green_solar, green_wind,
         green_hydro, green_nuclear, green_other) %>%
  # 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[place$Source  == "green_nuclear", "Source"] = "Nuclear"
place[place$Source  == "green_hydro", "Source"] = "Hydro"
place[place$Source  == "green_solar", "Source"] = "Solar"
place[place$Source  == "green_wind", "Source"] = "Wind"
place[place$Source  == "green_other", "Source"] = "Other"
```

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

# Creation of the plot
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")) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(title = "Green score in developed countries, by source, 2020",
       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. The 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 other sources.

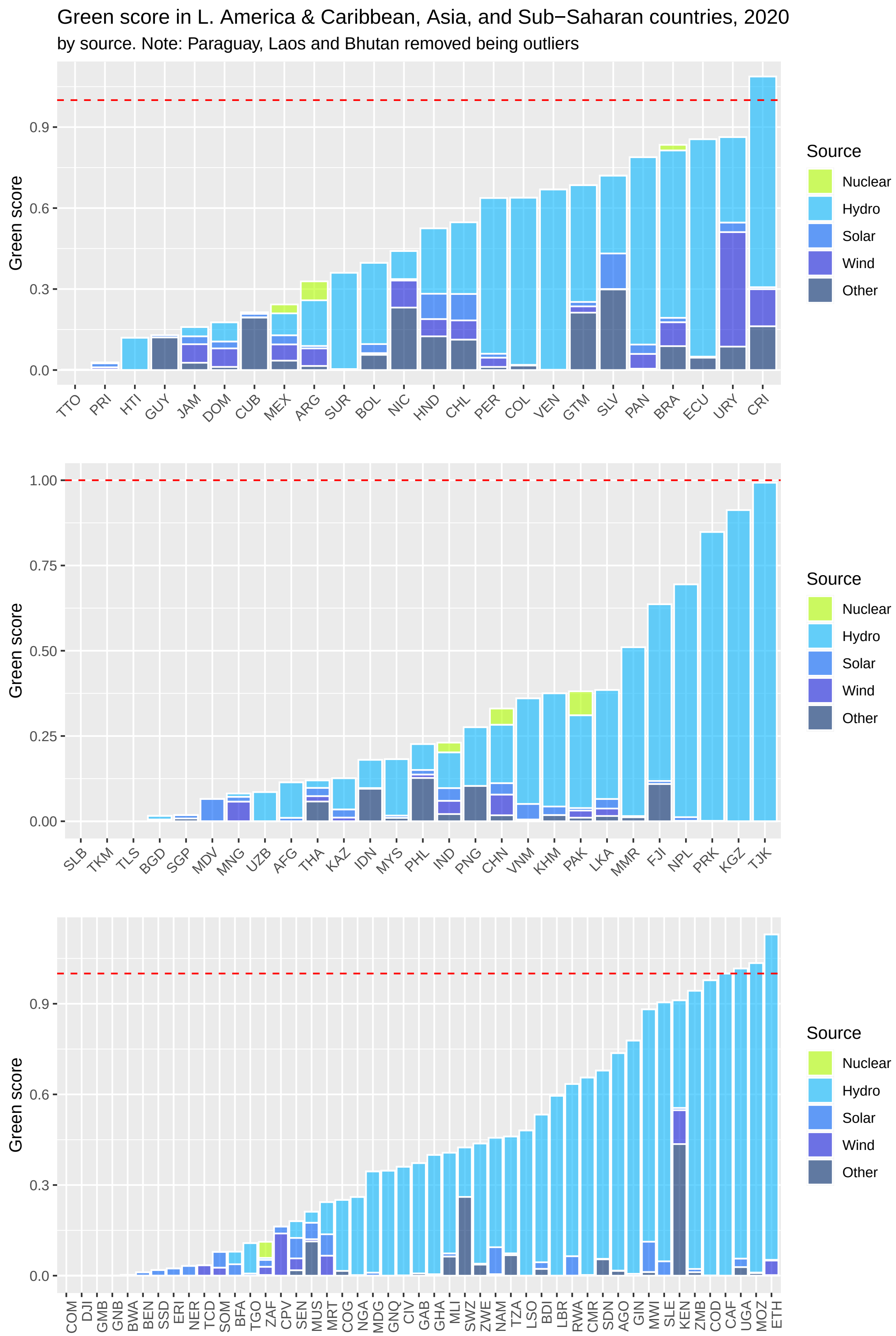
```r
ggplot(filter(place, tag == "east_europe"),
       aes(x = reorder(iso_code, value), y = value, fill = Source)) +
  geom_bar(position = "stack", stat = "identity", alpha = 0.6, colour = "white") +
  scale_fill_manual(values = c("#B2FF00", "#05B6FF", "#0060FA", "#141BDB", "#00296B")) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(title = "Green score in Eastern European countries, by source, 2020",
       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. The countries in the region mainly produce LC electricity through hydro and nuclear sources.

```
# 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")) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(title = "Green score in L. America & Caribbean, Asia, and Sub-Saharan countries, 2020",
       subtitle = "by source. Note: Paraguay, Laos and Bhutan removed being outliers",
       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")) +
  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")) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(x = "",
       y = "Green score") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

```
print(grid.arrange(gg1, gg2, gg3, ncol=1))
```


Green score in L. America & Caribbean, Asia, and Sub−Saharan countries, 2020
by source. Note: Paraguay, Laos and Bhutan removed being outliers
Green score
0.0
0.3
0.6
0.9
TTO PRI HTI GUY JAM DOM CUB MEX ARG SUR BOL NIC HND CHL PER COL VEN GTM SLV PAN BRA ECU URY CRI
Source
Nuclear
Hydro
Solar
Wind
Other
Green score
0.00
0.25
0.50
0.75
1.00
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
Source
Nuclear
Hydro
Solar
Wind
Other
Green score
0.0
0.3
0.6
0.9
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

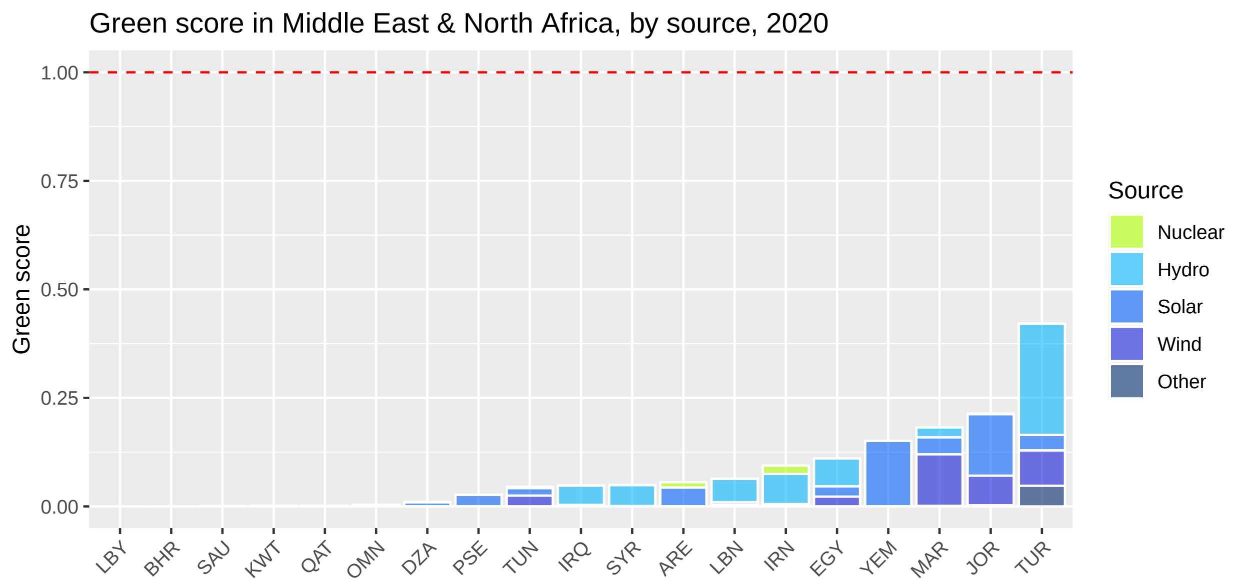
```
## TableGrob (3 x 1) "arrange": 3 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (2-2,1-1) arrange gtable[layout]
## 3 3 (3-3,1-1) arrange gtable[layout]
```

```
# Print electricity production in Tajikistan, Kysrgikistan, and North Korea compared to China
place2 = filter(main, year == 2020 &
                 (iso_code == "TJK" | iso_code == "KGZ" | iso_code == "PRK" | iso_code == "CHN")) %>%
  select(iso_code, electricity_demand, population) %>%
  summarize(iso_code, demand_percapita = electricity_demand / population)
print(place2)
```

```
##   iso_code demand_percapita
## 1      CHN     5.416569e-06
## 2      KGZ     2.524561e-06
## 3      PRK     5.841313e-07
## 4      TJK     1.911306e-06
```

We grouped the findings for Latin America & Caribbean, Asia, and Sub-Saharan countries because their electricity mixes are similar and mainly driven by hydropower. Here are the 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; 3. The Asian countries with a high green score are those with the lowest electricity generation per capita: Tajikistan, Kyrgyzstan, and North Korea.

```
# Creation of the plot
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")) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  labs(title = "Green score in Middle East & North Africa, by source, 2020",
       x = "",
       y = "Green score") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



We conclude the descriptive analyses by looking at the performance of Northern Africa and Middle East countries. The best performing country, Turkey, has a green score smaller than 0.5. As the following table shows, this is mainly due to an heavy exploitation of fossil sources.

```
# Creation of the dataset and of the table
filter(main, tag == "middle_east", year == 2020) %>%
  select(iso_code, fossil_share_elec) %>%
  arrange(desc(fossil_share_elec))
```

```
##    iso_code fossil_share_elec
## 1       BHR            99.967
## 2       LBY            99.967
## 3       SAU            99.938
## 4       KWT            99.925
## 5       QAT            99.910
## 6       OMN            99.621
## 7       DZA            99.010
## 8       TUN            95.659
## 9       SYR            95.193
## 10      ARE            94.421
## 11      IRQ            94.223
## 12      LBN            93.634
## 13      IRN            90.715
## 14      EGY            88.996
## 15      YEM            84.877
## 16      MAR            81.709
## 17      JOR            78.443
## 18      PSE            77.500
## 19      TUR            57.979
```

Bibliografia e link

[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

Prova

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

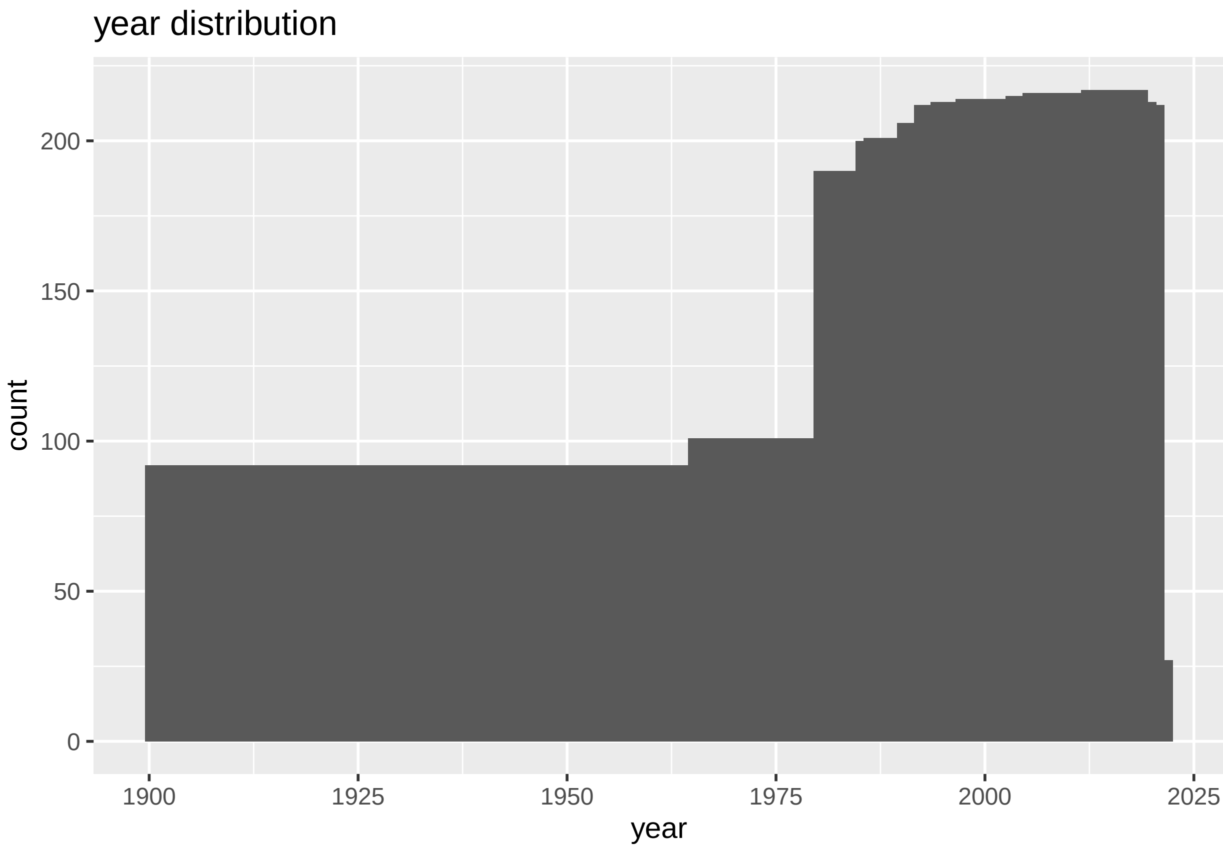
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

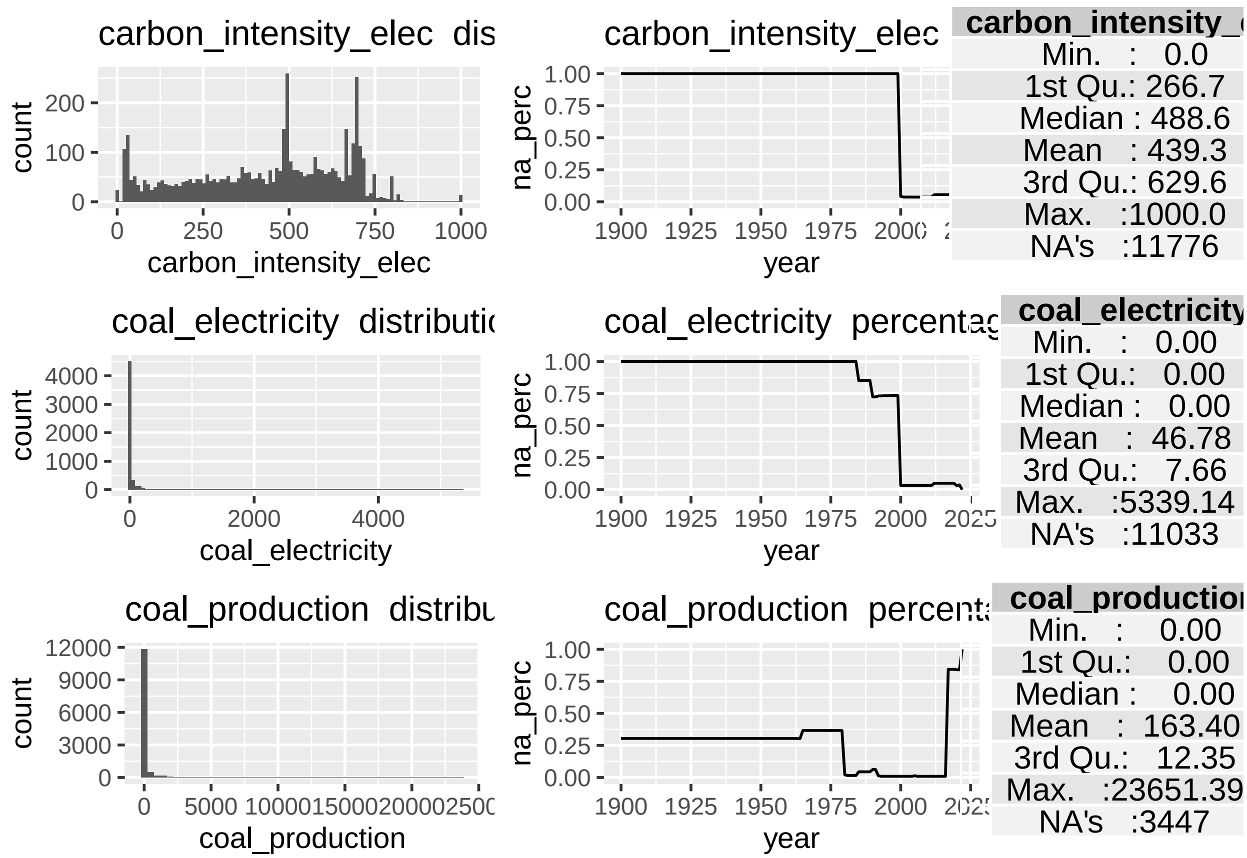
```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

La prova del plot di Giacomo:




```
## Warning: Removed 11776 rows containing non-finite values (stat_bin).

## Warning: Removed 11033 rows containing non-finite values (stat_bin).

## Warning: Removed 3447 rows containing non-finite values (stat_bin).
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.