UN Sustainable Development Goals Data Analysis

Fredrik Randsborg Bølstad

June 16, 2019

Contents

1 Introduction

The purpose of this project is to explore the global progress towards the United Nations (UN) Sustainable Development Goals (SDG). The main dataset is the UN SDG Indicators dataset provided by the UN Statistics Division. It is complimented by data from the Gapminder Foundation and the World Bank. Finding powerful ways of visualizing the dataset will be of particular importance to the project. With the help of the maps and mapdata R packages, a key ambition is to visualize reported progress for each country on a world map. Finally, the report will touch on what the current data may say about future results. Key goals:

- Programmatically download data from the relevant sources
- Clean and merge data into unified datasets
- Communicate reported results graphically
- Use trend analysis to predict future results

2 Data analysis

2.1 Downloading data

The main data for this project is provided by the UN Statistics Division via Google BigQuery. The geographic areas in the UN dataset includes countries, regions and continents. The geographical scope of this analysis is limited to the country list used by the Gapminder Foundation. The UN dataset will thus have to be filtered to only keep the countries that are in the Gapminder's data. Population data is separately downloaded from The World Bank's repository. Country coordinate data is extracted from the maps and mapsdata R packages.

Table 1: Data sources

Data source	Fetch method
UN Sustainable Development Goals Indicators	bigrquery - R interface to BigQuery
Gapminder geography data	googlesheets
World Bank population data	wbstats - R interface to World Bank API
Spatial map data	maps and mapdata

The raw data is downloaded and saved for processing. The raw and clean data are thus stored separately to avoid having to download the raw data over again, in case an unexpected or undesired result flow from modification of the dataset during data cleaning.

The code used to download and save the above data from each source can be found below. The code is commented to explain exactly what it does.

2.1.1 UN SDG Indicators data from Google BigQuery

```
library(bigrquery)    # Query Google's BigQuery database
library(DBI)          # Interface for communication with relational database systems

# Local destination for downloaded raw data
un_sdg_file = "./raw_data/bigquery_un_sdg.rda"

# Google Cloud project ID required for BigQuery interactions
```

```r
# Create your own project ID at https://console.cloud.google.com/
google_cloud_project_id <- "edx-ds-capstone-2019"

# BigQuery connection interface
dbi_connection <- dbConnect(
  bigrquery::bigquery(),
  project = "publicdata",
  dataset = "un_sdg",
  billing = google_cloud_project_id
)

# SQL query for entire dataset (~300 MB)
sql_query <- "SELECT * FROM `bigquery-public-data.un_sdg.indicators`"

# Download dataset
un_sdg_data <- dbGetQuery(dbi_connection, sql_query)

# Save dataframe to file
save(un_sdg_data, file = un_sdg_file)
```

2.1.2 Gapminder country list

```r
library(googlesheets)   # Google spreadsheet API

# Local destination for downloaded raw data
gapminder_file = "./raw_data/gapminder_country_list.rda"

# Connect to spreadsheet (URL from https://www.gapminder.org/data/geo/)
  gapminder_spreadsheet <- gs_url(paste("https://docs.google.com/spreadsheets/",
                                        "d/1qHalit8sXC0R8oVXibc2wa2gY7bkwGzOybEMTWp-08o/",
                                        sep = "") # formatted to fit on page

# Download list-of-countries worksheet
gapminder_country_list <- gapminder_spreadsheet %>%
  gs_read(ws = "list-of-countries-etc")

# Keep only country and region.
#> Mutate colname to 'geoareaname' to match UN SDG data colname
#> Make geo uppercase to match World Bank data
gapminder_country_list <- gapminder_country_list %>%
  mutate(geoareaname = name, region = four_regions) %>%
  select(geo, geoareaname, region) %>%
  mutate(geo = toupper(geo))

# Save dataframe to file
save(gapminder_country_list, file = gapminder_file)
```

2.1.3 World Bank population data

```
library(wbstats)  # World Bank API interface

# Local destination for downloaded raw data
wb_pop_file = "./raw_data/world_bank_population_data.rda"

# Create list of ISO 3 country codes from Gapminder
# which will be used to query World Bank's data
iso3c_codes <- gapminder_country_list$geo

# Remove HOS (Holy See) which isn't in World Bank data
# https://datahelpdesk.worldbank.org/knowledgebase/articles/898590-country-api-queries
iso3c_codes <- iso3c_codes[-which(iso3c_codes == 'HOS', )]

# Get year interval for which there is UN SDG data
if(!exists("un_sdg_data")) load(un_sdg_file);
first_year <- min(un_sdg_data$timeperiod)
last_year <- max(un_sdg_data$timeperiod)

# Download population size from the World Bank
wb_pop <- wb(country = iso3c_codes,
  indicator = 'SP.POP.TOTL',
  startdate = first_year, enddate = last_year)

# Save dataframe to file
save(wb_pop, file = wb_pop_file)
```

2.2 Cleaning data

2.2.1 Merging datasets

The data from the UN, Gapminder Foundation and the World Bank are merged into one unified dataset. There are two main goals for this process:

1. Join UN SDG data with Gapminder data, such that all rows relating to geographical areas not in the Gapminder country list are dropped.
2. Append the World Bank population data for each country and the respective years in the dataset.

```
### Join UN SDG and Gapminder dataset such that:
# > Drop from UN SDG all countries not in Gapminder
# > Append to UN SDG the region column from Gapminder
dat <- inner_join(un_sdg_data, gapminder_country_list)

### Append World Bank population data, by country and year
# Rename and select World Bank columns to match UN SDG dataset
wb_pop <- wb_pop %>%
  rename(geo = iso3c, timeperiod = date, population = value) %>%
  select(geo, timeperiod, population)

# Append population column to UN SDG dataset
dat <- left_join(dat, wb_pop)
```

2.2.2 Selecting relevant data

After preliminary exploration of the dataset, including manual inspection, it became clear that the reported UN data were severely lacking. Many indicators had no, little or inconsistent data reporting. Only a minority of indicators were consistently or semi-consistently reported on by a broad community of countries. A selection of indicators with sufficient data were therefore handpicked to be used for the continued analysis. The target description for each indicator were manually added to the dataset for convenience (so that the goal of the indicator can quickly be queried). The complete description of all indicators in the dataset can be found in the annex of the UN General Assembly Resolution A/RES/71/313. This is also the source of the text descriptions copied into the the code below.

Table 2: Selected UN SDG Indicators

Indicator	Description
6.1.1	Population proportion using safe drinking water services
7.1.1	Population proportion with access to electricity
7.2.1	Renewable energy (as share of total consumption)
8.1.1	Annual growth rate, real GDP per capita
9.4.1	CO_2 emission per unit of value added
12.2.1	Material footprint
12.2.2	Domestic material consumption
17.6.2	Population proportion with broadband subscriptions
17.8.1	Population proportion using the internet

```
### Indicators to keep
indicators <- c("6.1.1", "7.1.1", "7.2.1", "8.1.1", "9.4.1",
                "12.2.1", "12.2.2", "17.6.2", "17.8.1")

### Text description of targets for indicators (https://undocs.org/A/RES/71/313)
target_description <-
  c('By 2030, achieve universal and equitable access to
    safe and affordable drinking water for all',
    'By 2030, ensure universal access to affordable, reliable and modern energy services',
    'By 2030, increase substantially the share of
    renewable energy in the global energy mix',
    'Sustain per capita economic growth in accordance with
    national circumstances and, in particular, at least 7 per cent
    gross domestic product growth per annum in the least developed countries',
    'By 2030, upgrade infrastructure and retrofit industries to make them sustainable,
    with increased resource-use efficiency and greater adoption of clean and
    environmentally sound technologies and industrial processes, with all countries
    taking action in accordance with their respective capabilities',
    'By 2030, achieve the sustainable management and efficient use of natural resources',
    'By 2030, achieve the sustainable management and efficient use of natural resources',
    'Enhance North-South, South-South and triangular regional and international
    cooperation on and access to science, technology and innovation and
    enhance knowledge-sharing on mutually agreed terms, including through
    improved coordination among existing mechanisms, in particular at the
    United Nations level, and through a global technology facilitation mechanism',
    'Fully operationalize the technology bank and science, technology and innovation
    apacity-building mechanism for least developed countries by 2017 and enhance the
    use of enabling technology, in particular information and communications technology')
```

```r
# Data frame of indicators and text descriptions (later appended to the main dataset)
descriptions_dat <- data.frame(indicators, target_description) %>%
  rename(indicator = indicators)
```

2.2.3 Dropping, renaming and coercing data

The entire UN SDG Indicators dataset were downloaded to begin with. After the preliminary exploration of the data and the selection of indicators, many variables are irrelevant to this project. Those are dropped from the data frame that will later be used for visualization and analysis. A few columns are also renamed to something more descriptive (e.g. *geoareaname* has been filtered to only include countries and is thus renamed to *country*). Finally, each column is coerced into either a numeric (e.g. reported values), integer (e.g. years) or factor (e.g. region) to optimize performance.

```r
### Manipulate columns with dplyr
dat <- dat %>%

  # Filter for chosen indicators
  filter(indicator %in% indicators) %>%

  # Remove seriescode, geoareacode, time_detail, freq
  select(-c('goal', 'seriescode', 'geoareacode', 'time_detail', 'freq')) %>%

  # Rename geoareaname to country, timeperiod to year
  rename(country = geoareaname, year = timeperiod) %>%

  # Capitalize region column
  mutate(region = sub("(.)", "\\U\\1", region, perl=TRUE)) %>%

  # Remove empty columns
  select_if(~sum(!is.na(.)) > 0) %>%

  # Coerce the type of each column
  mutate(target = as.factor(target),
         indicator = as.factor(indicator),
         year = as.integer(year),
         value = as.numeric(value),
         nature = as.factor(nature),
         location = as.factor(location),
         type_of_product = as.factor(type_of_product),
         type_of_speed = as.factor(type_of_speed),
         units = as.factor(units),
         region = as.factor(region))
```

2.2.4 Calculate absolute population numbers

The UN SDG data only lists a value for each indicator, e.g. the population proportion that has access to safe drinking water. With the population data from the World Bank, it is also possible to estimate the absolute number of people with access to safe drinking water. This is interesting in itself and could be relevant for future analysis. Future analysis could e.g. weigh global progress according to population numbers instead of treating each country as equally important to aggregate global progress.

```r
# Use value and population to create column containing the absolute number of people
dat <- dat %>%
  mutate(nr_people = as.integer(floor(population*value/100)))
```

2.2.5 Append target descriptions for SDG indicators

The target descriptions for each indicator, which were previously saved in a separate data frame, is appended to the main data frame.

```r
# Append target descriptions to dat
dat <- left_join(dat, descriptions_dat)
```

2.2.6 Harmonizing map data with the main dataset

Map data from the *map_data* R package is saved to a variable and the country column is renamed to match with the main dataset.

```r
### Save map coordinates in dataframe
map_world <- map_data('world')

# Rename region as country to harmonize with 'dat'
map_world <- map_world %>%
  rename(country = region)
```

Find and list all instances of spelling discrepancies for country names.

```r
# List all mismatches in country names between dat and map_world
dat %>%
  anti_join(map_world, by = 'country') %>%
  select(country) %>%
  unique()
```

Rename countries for which there are spelling discrepancies in order to cooperate fully with the main dataset.

```r
# Rename countries in dat to match those of map_world
dat <- dat %>%
  mutate(country = recode(country,
                          'Antigua and Barbuda' = 'Antigua',
                          'Holy See' = 'Vatican',
                          'Trinidad and Tobago' = 'Trinidad')) %>%

  # Remove rows with country Tuvalu
  filter(country != 'Tuvalu')
```

2.2.7 Save tidy data to disk

```r
### Save data frame to file
save(dat, file = tidy_file)
save(map_world, file = world_map_file)
```

2.3 Exploring and visualizing data

Most of the work on this data analysis project ended up being dedicated to visualizing the data. A significant amount of thought and research went into different options. A key achievement of this project was the successful creation of a function that generates a progress heatmap of the world, given any UN SDG indicator and year. Moreover, data for population proportion indicators were separated into the rural and urban population; a side-by-side comparison of the world heatmaps for each population category, and with a scatterplot showing progress per year, was the culmination of this visualization effort. Several functions were programmed to achieve this result in am intuitive and user friendly manner.

Functions for generating GIFs of these maps were also created. This enables a year-by-year animation of progress. Because the GIFs can't be displayed in this PDF report, the code is not included here. Examples of GIF animations are however uploaded to this project's github repository and linked to in the appropriate section below. The full code for the GIF creation is in the main code file (also available on github).

Titles, subtitles, units, year and regions are programmatically added to the plots.

2.3.1 World heatmaps

The functions needed to generate a world heatmap for any indicator:

```
### Function to filter for given indicator and year; delete empty columns
select_ind_year <- function(data, ind, yr) {

  data %>%
    filter(indicator == ind, year %in% yr) %>%
    select_if(~sum(!is.na(.)) > 0)

}

### Function to create world heatmap
plot_world_heatmap <- function(data, ind, year) {

  # Create filtered dataset
  data_filtered <- select_ind_year(data, ind, year)

  # Join filtered dataset with world map
  map_data <- left_join(map_world, data_filtered, by = 'country')

  # Text for title and legend
  title <- map_data$seriesdescription %>% na.omit() %>% unique()
  unit <- map_data$units %>% na.omit() %>% unique() %>% as.character()

  # Plot graph
  ggplot(map_data, aes(x = long, y = lat, group = group)) +
    geom_polygon(aes(fill = value)) +

    # Titles and axis
    labs(title = paste(title),
         subtitle = paste('Year', year),
         fill = paste(unit),
         x = NULL, y = NULL) +

    # Legend
```

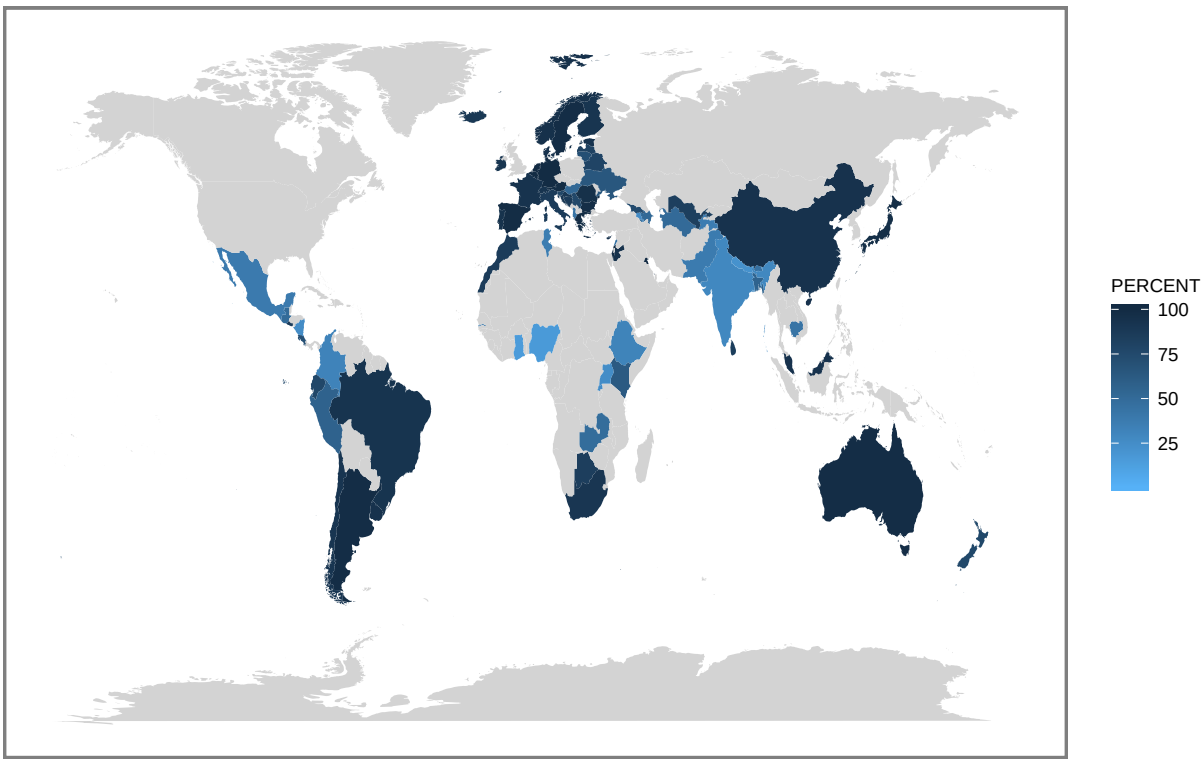
```
    scale_fill_gradient(low = "#56B1F7", high = "#132B43", na.value = 'lightgrey') +

    # Theme adjustments
    theme(plot.title = element_text(size = 9, face = 'bold'),
          plot.subtitle = element_text(size = 8, hjust = 0),
          axis.ticks = element_blank(),
          axis.text = element_blank(),
          legend.key.size = unit(14, 'pt'),
          legend.title = element_text(size = 7),
          legend.text = element_text(size = 7),
          panel.background = element_rect(fill = 'white'),
          plot.background = element_rect(fill = 'white'),
          panel.border = element_rect(size = 1, colour = 'gray50', fill = NA))

}
```

Looping through the above function for all selected indicators:

```
for (ind in indicators) {
  heatmap <- plot_world_heatmap(dat, ind, 2000)
  print(heatmap)
}
```

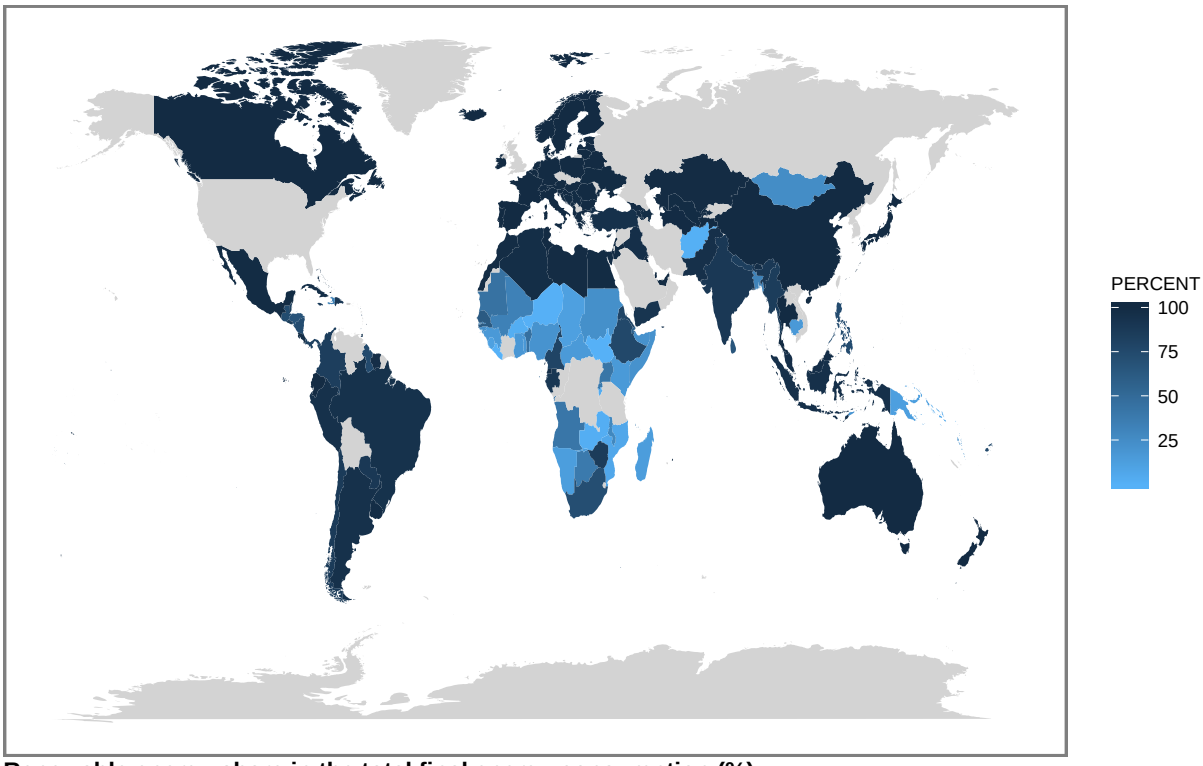
Proportion of population using safely managed drinking water services, by urban/rural (%)

Year 2000



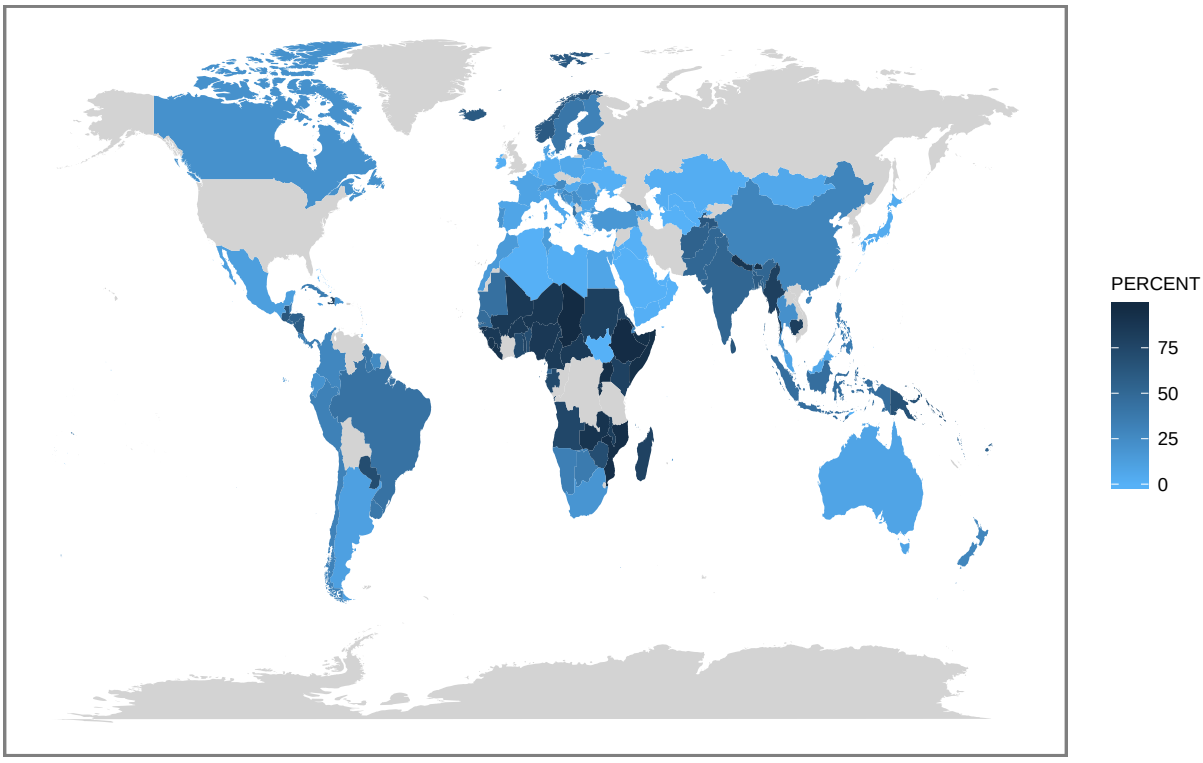
Proportion of population with access to electricity, by urban/rural (%)

Year 2000



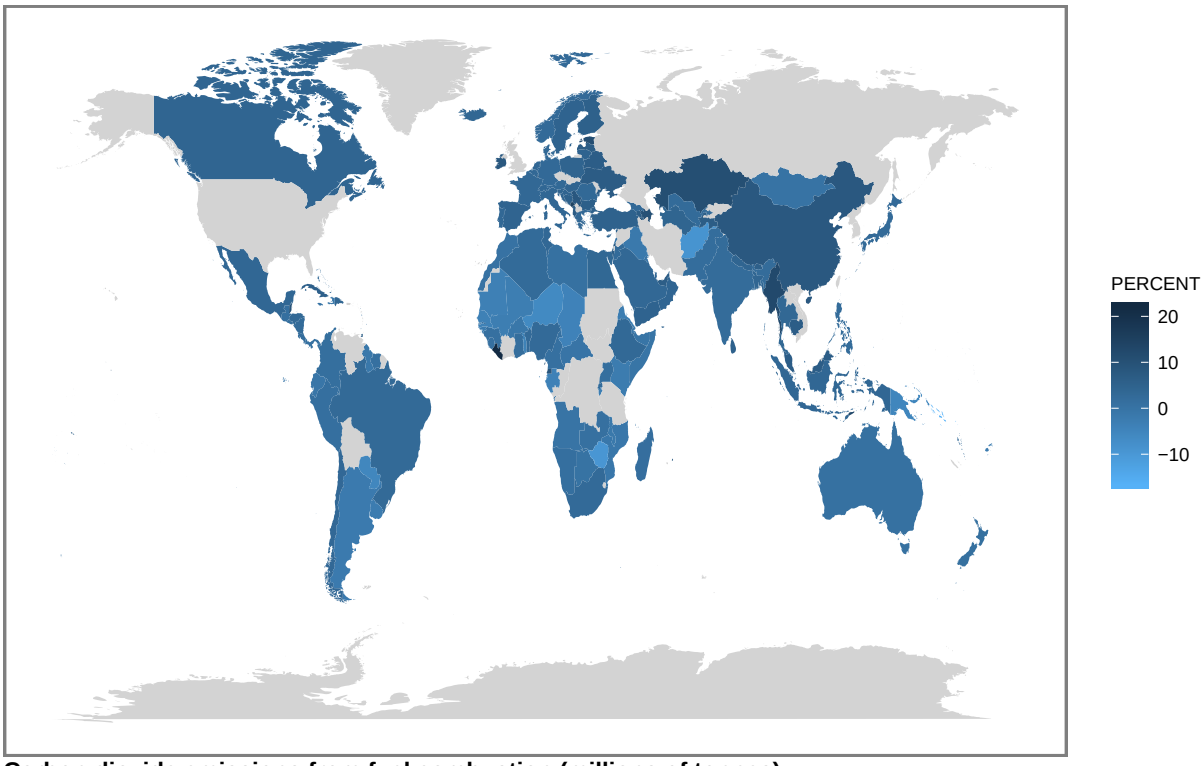
Renewable energy share in the total final energy consumption (%)

Year 2000



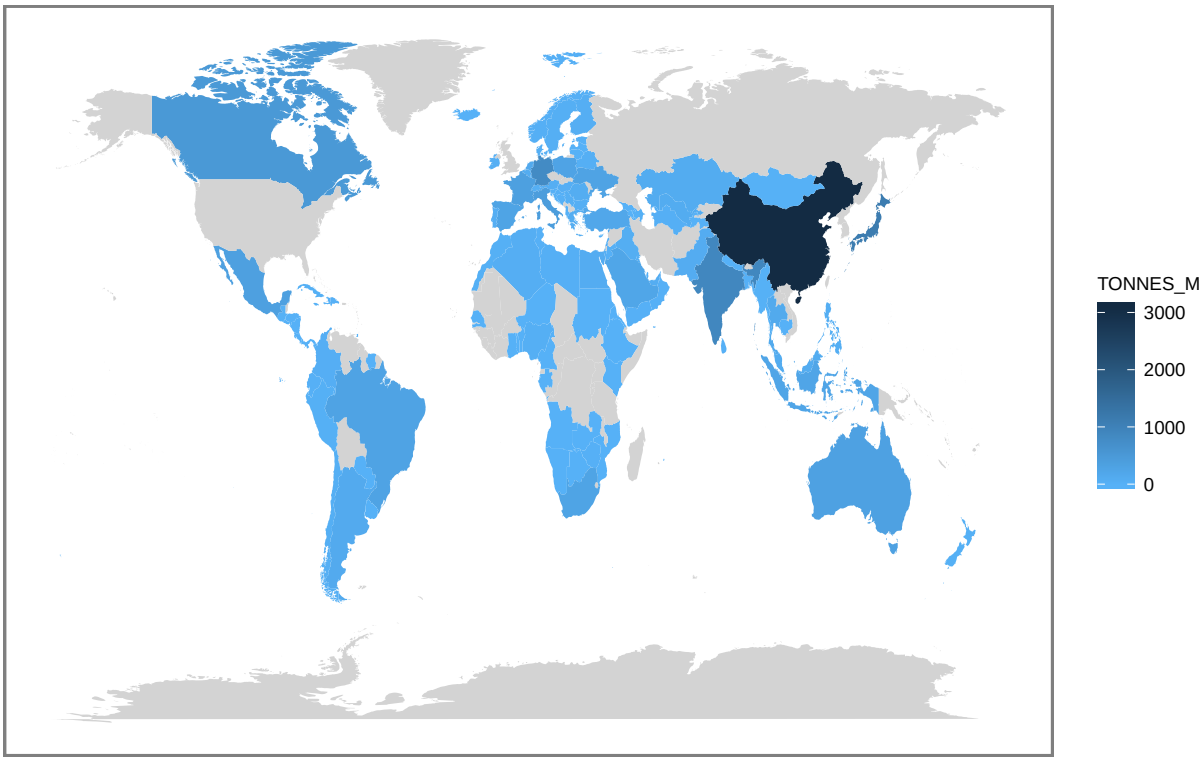
Annual growth rate of real GDP per capita (%)

Year 2000



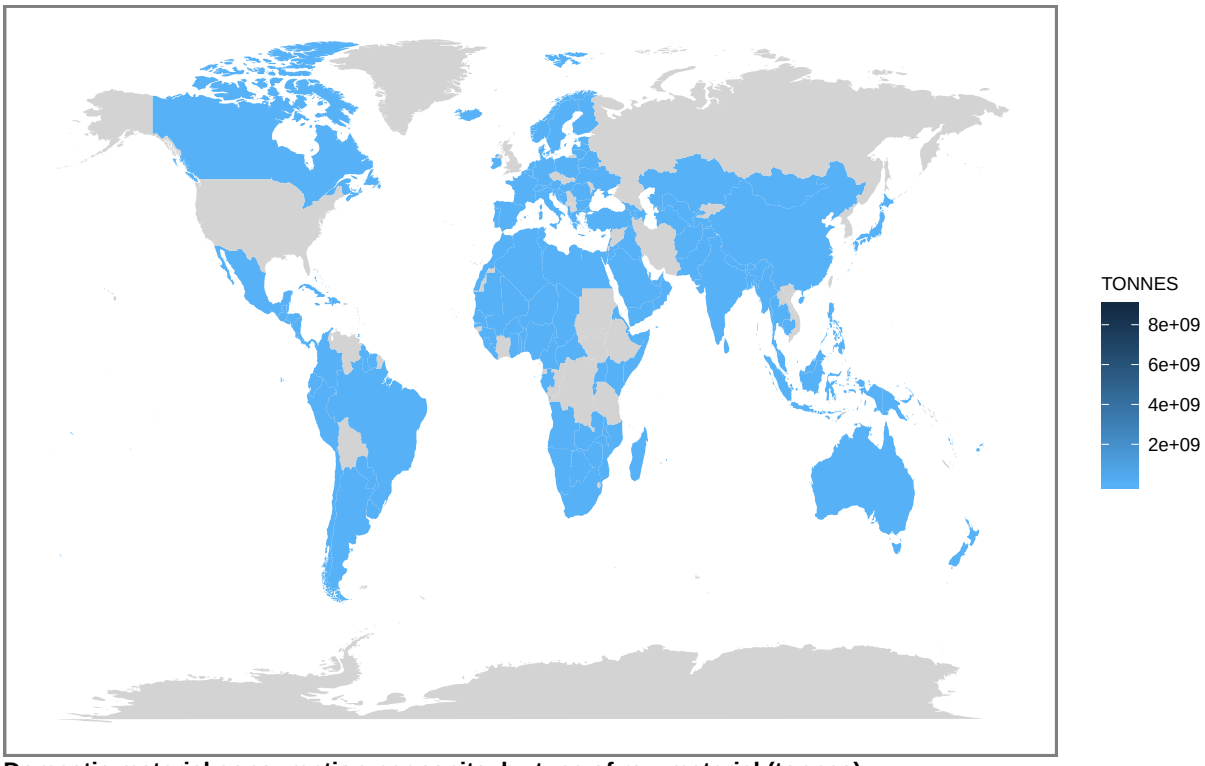
Carbon dioxide emissions from fuel combustion (millions of tonnes)

Year 2000



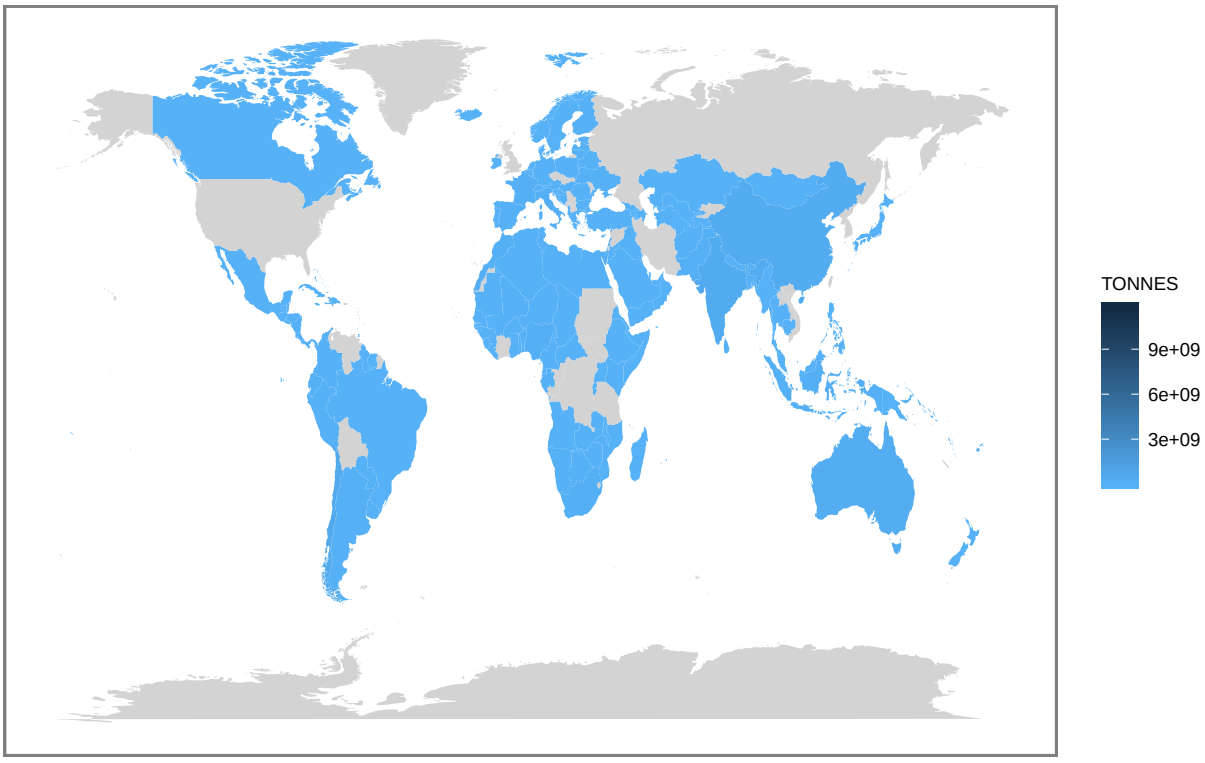
Material footprint per capita, by type of raw material (tonnes)

Year 2000



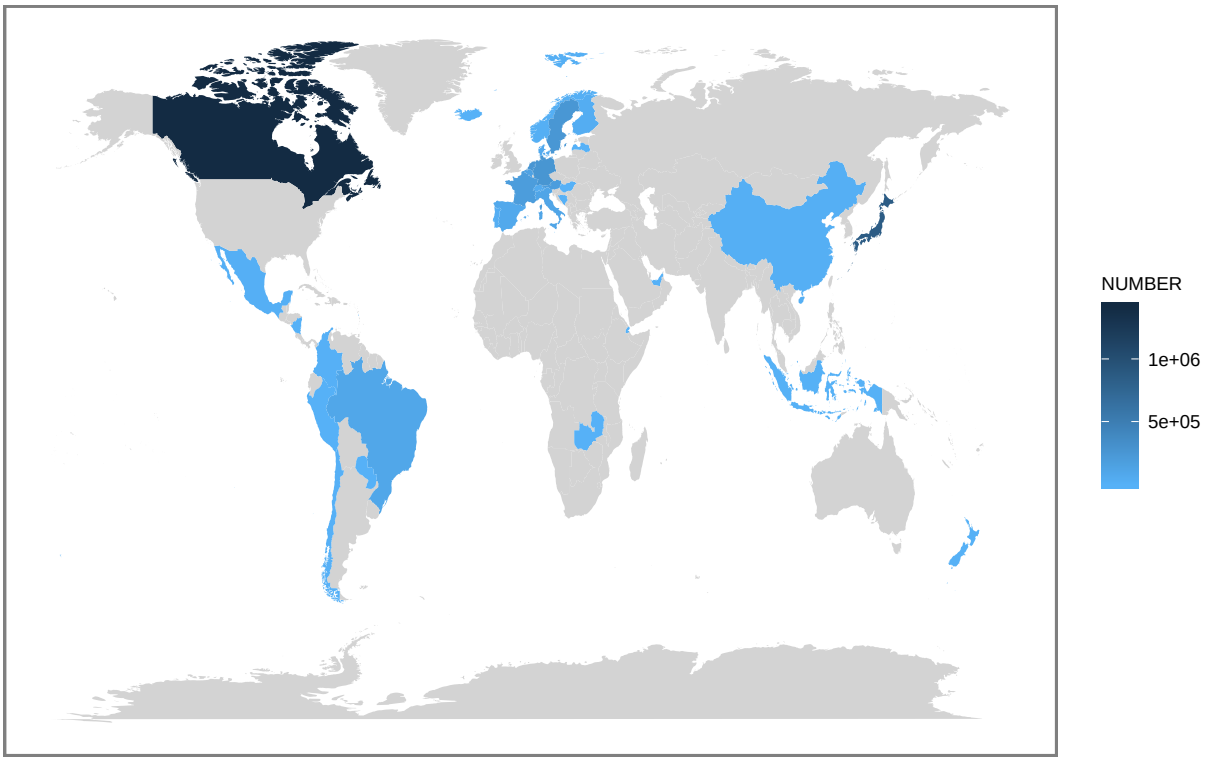
Domestic material consumption per capita, by type of raw material (tonnes)

Year 2000



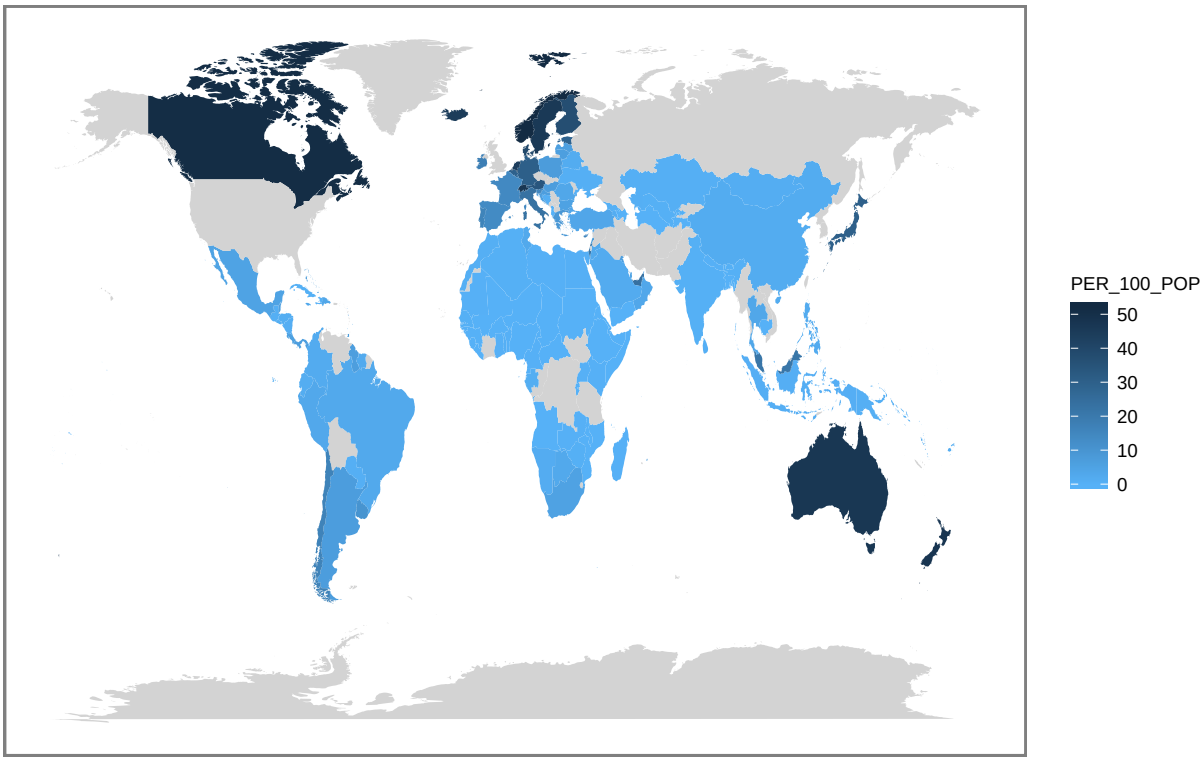
Number of fixed Internet broadband subscriptions, by speed (number)

Year 2000



Internet users per 100 inhabitants

Year 2000



2.3.2 World heatmap - rural vs urban grid

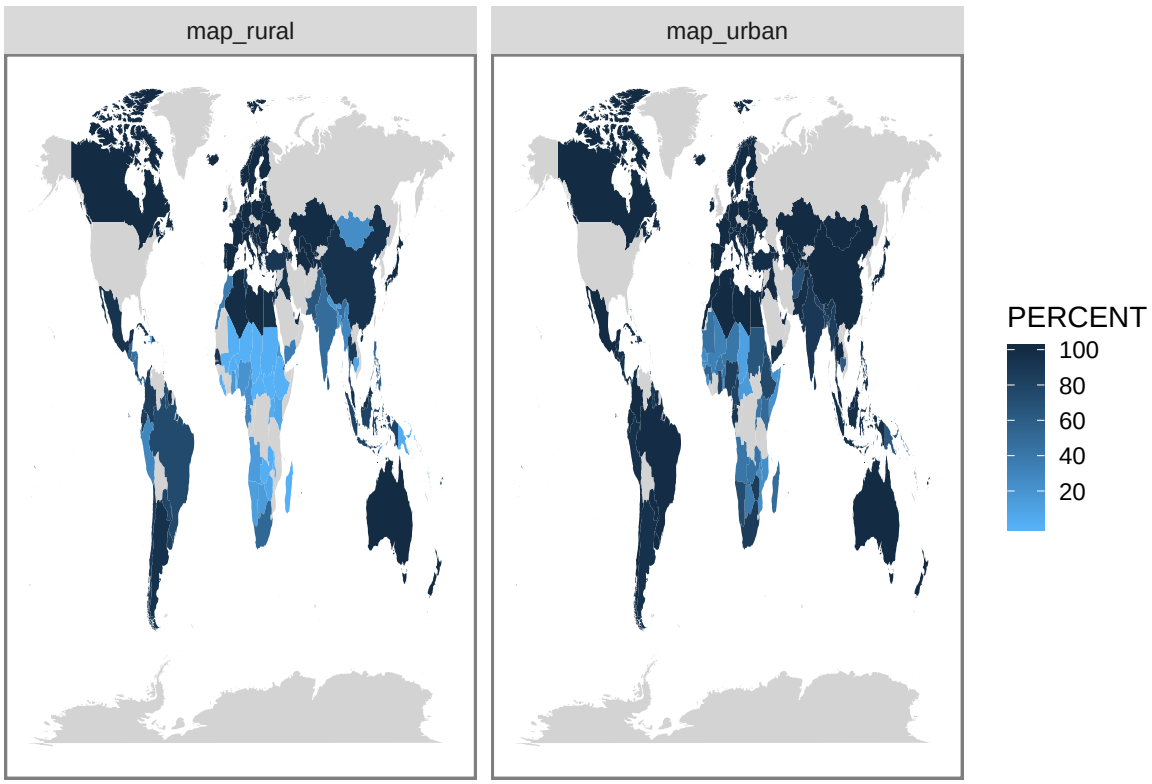
Function to create the world heatmap grid of urban vs rural.

```
### Filter for given indicator, year and loc; delete empty columns
select_ind_year_loc <- function(data, ind, yr, loc) {

  data %>%
    filter(indicator == ind, year %in% yr, location == loc) %>%
    select_if(~sum(!is.na(.)) > 0)

}

### Create world heatmap, rural vs. urban 1x2 grid
facet_heatmap <- function(data, ind, yr) {

  # Create filtered datasets for each location
  data_urban <- select_ind_year_loc(data, ind, yr, "URBAN")
  data_rural <- select_ind_year_loc(data, ind, yr, "RURAL")

  # Joined filtered data with world map
  map_urban <- left_join(map_world, data_urban, by = 'country')
  map_rural <- left_join(map_world, data_rural, by = 'country')

  # Bind all data together
  map <- rbind(map_urban, map_rural)

  # Create data frame with facet property
  map <- data.frame(map, Facet = rep(c("map_urban","map_rural"),
                                     times=c(nrow(map_urban),nrow(map_rural))))

  # Text for plot title
  title <- map$seriesdescription %>% na.omit() %>% unique()
  unit <- map$units %>% na.omit() %>% unique() %>% as.character()

  # Plot graph
  ggplot(map, aes(x = long, y = lat, group = group)) +
    geom_polygon(aes(fill = value)) +

    # Labels and legend
    labs(title = paste(title),
         subtitle = paste('Year', yr),
         fill = paste(unit),
         x = NULL, y = NULL) +

    scale_fill_gradient(low = "#56B1F7", high = "#132B43", na.value = 'lightgrey',
                        breaks = seq(0, 100, 20)) +

    # Theme adjustments
    theme(plot.title = element_text(size = 12, face = 'bold'),
          plot.subtitle = element_text(size = 9, hjust = 0),
          axis.ticks = element_blank(),
          axis.text = element_blank(),
          legend.key.size = unit(14, 'pt'),
```

```
          panel.background = element_rect(fill = 'white'),
          panel.border = element_rect(size = 1, colour = 'gray50', fill = NA),
          plot.margin = unit(c(0,0,0,11),"mm")) + # top, right, bottom, left

    # Faceting
    facet_wrap(.~Facet)

}
```

Example of world heatmap grid (this plot is more useful in larger dimensions but the concept is still illustrated).

```
facet_heatmap(dat, "7.1.1", 2000)
```



2.3.3 Scatterplot - rural vs urban grid

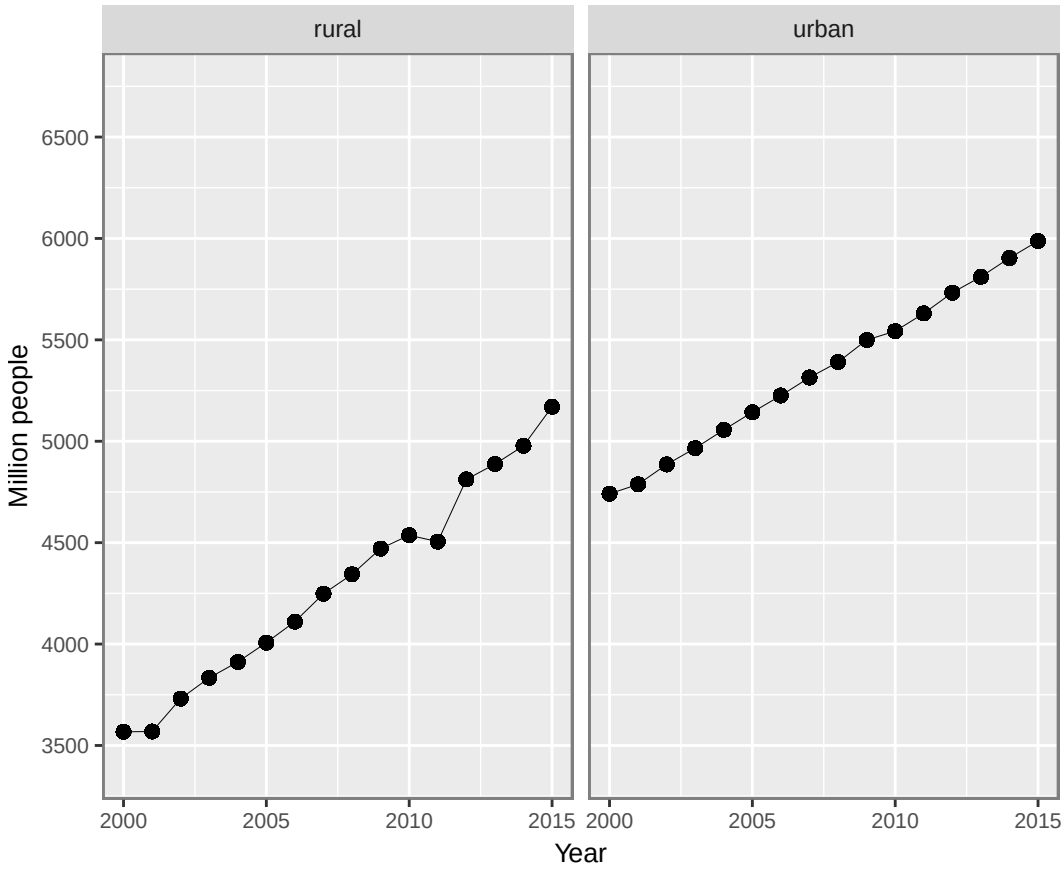
Function for year-to-year scatterplot of the above development, comparing rural to urban. Requires both a start year and an end year as input.

```
### Create scatterplot, rural vs. urban 1x2 grid
facet_scatterplot <- function(data, ind, start, end, x_max = end) {

  # Create function that filters data used for plotting
  filter_data <- function(data, ind, start, end, loc) {
```

```
    dat %>%

      # Filter for indicator, given year interval and location
      filter(indicator == ind, year %in% c(start:end), location == loc) %>%

      # Add column for nr. of people covered (in millions) for all countries
      group_by(year) %>%
      mutate(total_people_million = sum(nr_people/10^6, na.rm = TRUE))

  }

  # Create filtered datasets for each location
  rural <- filter_data(data, ind, start, end, "RURAL")
  urban <- filter_data(data, ind, start, end, "URBAN")

  # Joined filtered data with world map
  plot <- rbind(rural, urban)

  # Create data frame with facet property
  plot <- data.frame(plot, Facet = rep(c("rural","urban"),
                                       times=c(nrow(rural),nrow(urban))))

  # Specify limits for plot
  x_min <- start
  #x_max defaults to 'end' but can be overriden via user argument
  y_min <- 3400
  y_max <- 6750

  # Plot graph
  plot %>%
    ggplot(aes(year, total_people_million)) +
    geom_point(size = 2) +
    geom_line(size = 0.1) +

    # Set limits
    scale_x_continuous("Year", breaks = seq(2000, 2015, by = 5), limits = c(x_min, x_max)) +
    scale_y_continuous("Million people", breaks = seq(3500, 6500, by = 500), limits = c(y_min, y_max)) +

    # Labels
    labs(x = "Year") +

    # Theme adjustments
    theme(panel.background = element_rect(linetype = 5),
          panel.border = element_rect(size = 1, colour = 'gray50', fill = NA),
          plot.margin = unit(c(0,26,0,0),"mm"),  # top, right, bottom, left
          axis.title = element_text(size = 10),
          axis.text = element_text(size = 8)) +

    # Faceting
    facet_grid(~Facet)

}
```

Example of scatterplot.

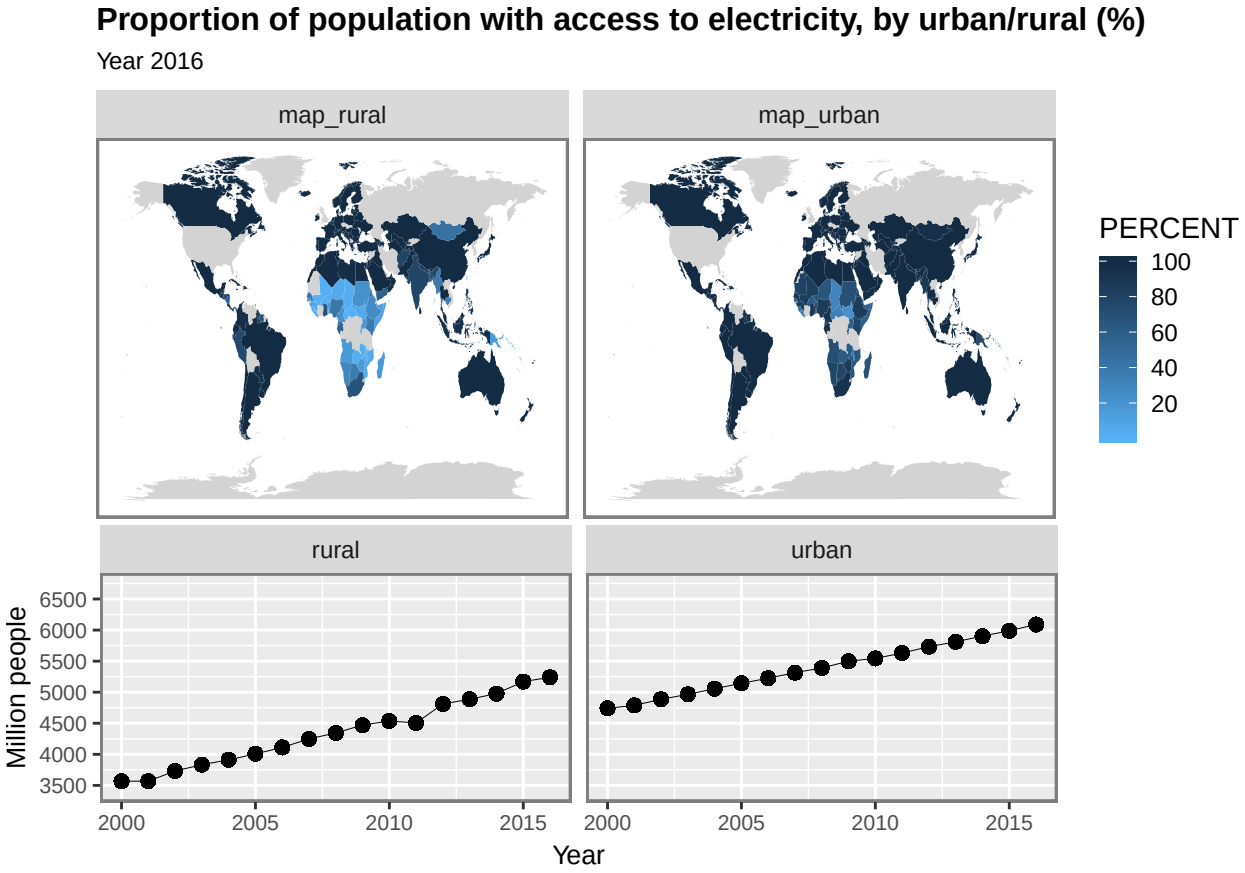
```
facet_scatterplot(dat, "7.1.1", 2000, 2015)
```



2.3.4 Combined heatmap and scatterplot - rural vs urban grid

It is possible to combine the two grids of plots. Here follows an example for the year 2016 and with the scatterplot starting from year 2000.

```
heatmap <- facet_heatmap(dat, "7.1.1", 2016)
scatterplot <- facet_scatterplot(dat, "7.1.1", 2000, 2016)
grid.arrange(heatmap, scatterplot, nrow = 2, heights = c(0.6, 0.4))
```



2.3.5 Animated heatmaps and scatterplots

It is possible to stitch together the above plot to create a time series animation. This is accomplished by stitching the PNGs together into a GIF. An example of the above plot animated in high resolution can be found here. While the functions can be used to create many more animations, a larger selection of examples are found here.

2.3.6 Country bar chart comparison

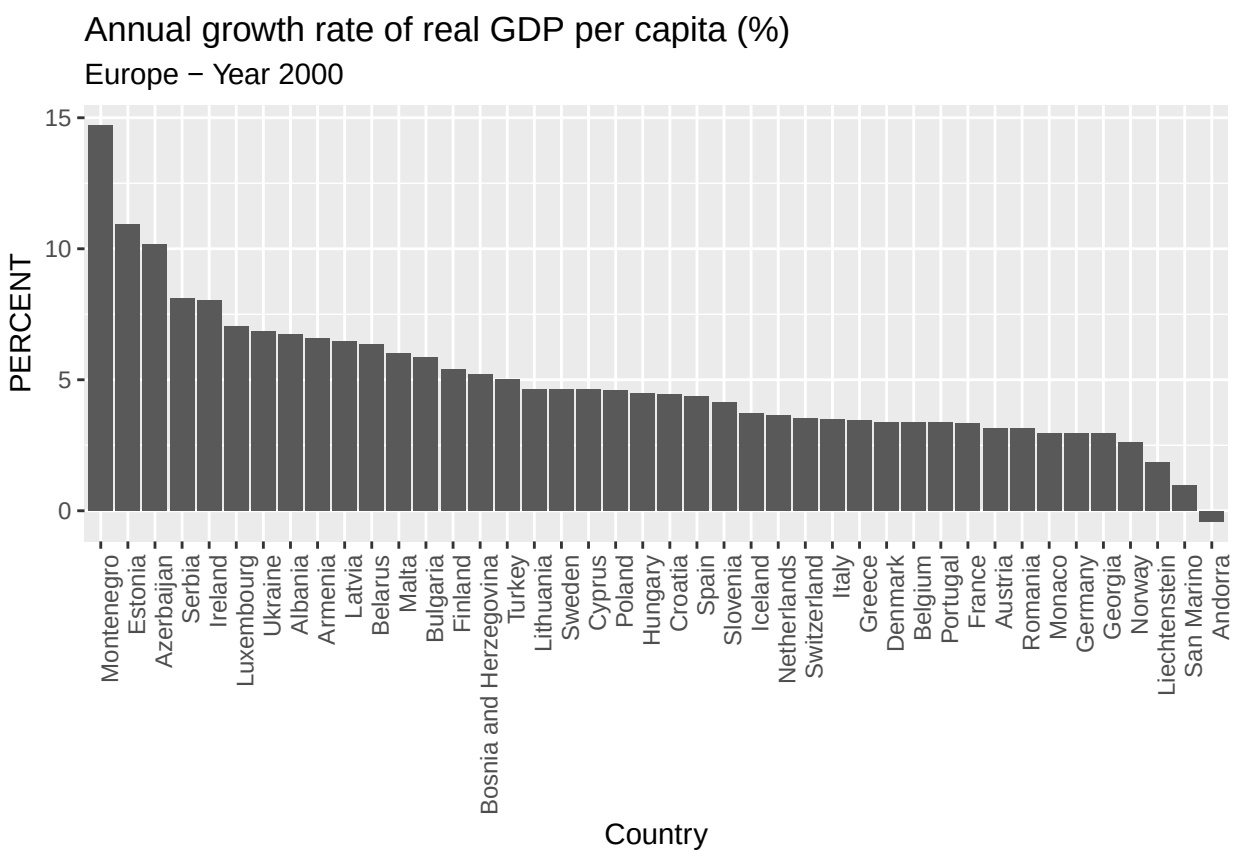
Below is a function that takes indicator, year and continent as inputs and returns an ordered bar chart of country performance (with the best performing countries first).

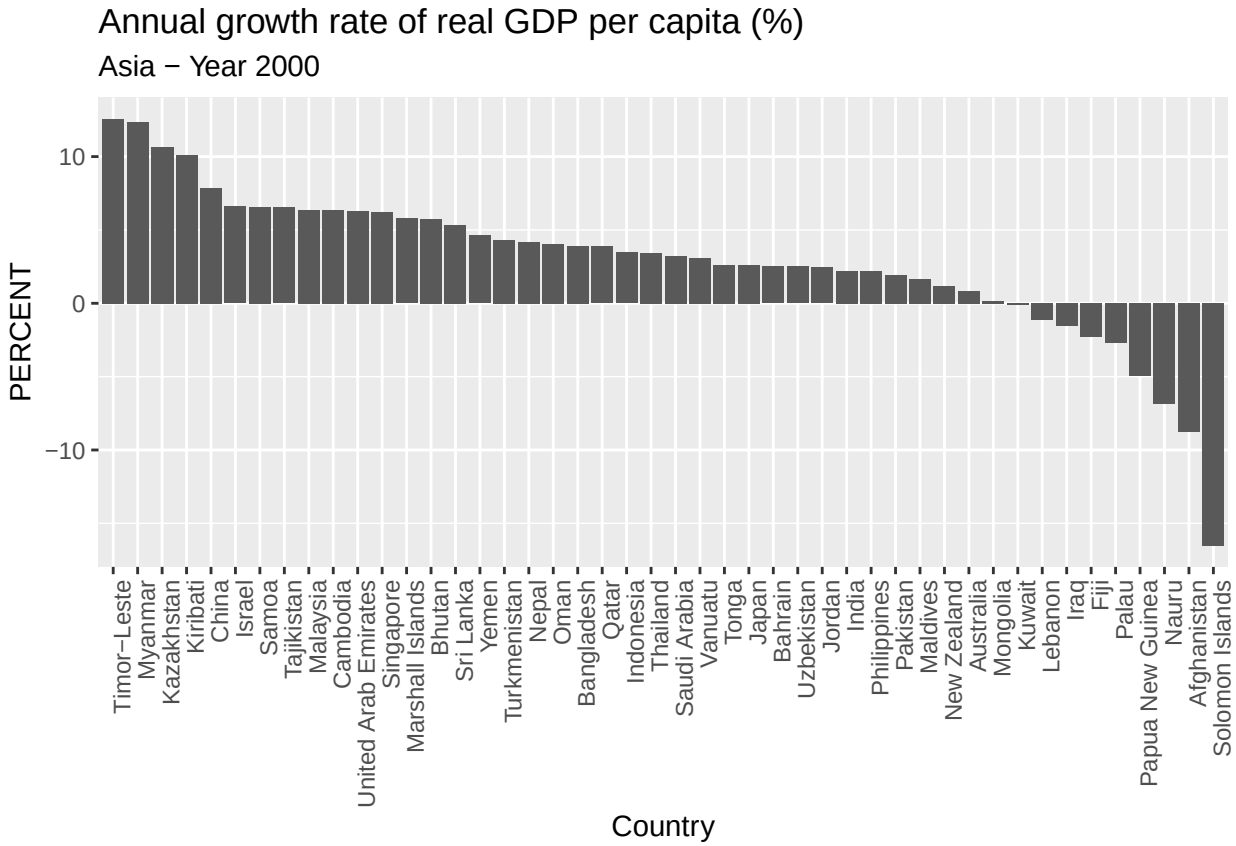
```
country_barchart <- function(data, ind, yr, continent) {

  # Filter data accoring to input arguments and select relevant columns
  filtered_data <- filter(data, year == yr, indicator == ind, region == continent) %>%
    select(c(indicator, seriesdescription, country, year, value, units, location, region))

  # Select only ALLAREA so as to not add up RURAL and URBAN for indicators 6.1.1 and 7.1.1.
  if (ind %in% c("6.1.1", "7.1.1")) {
    filtered_data <- filtered_data %>% filter(location == "ALLAREA")
  }

  # Get bar chart title and unit
```

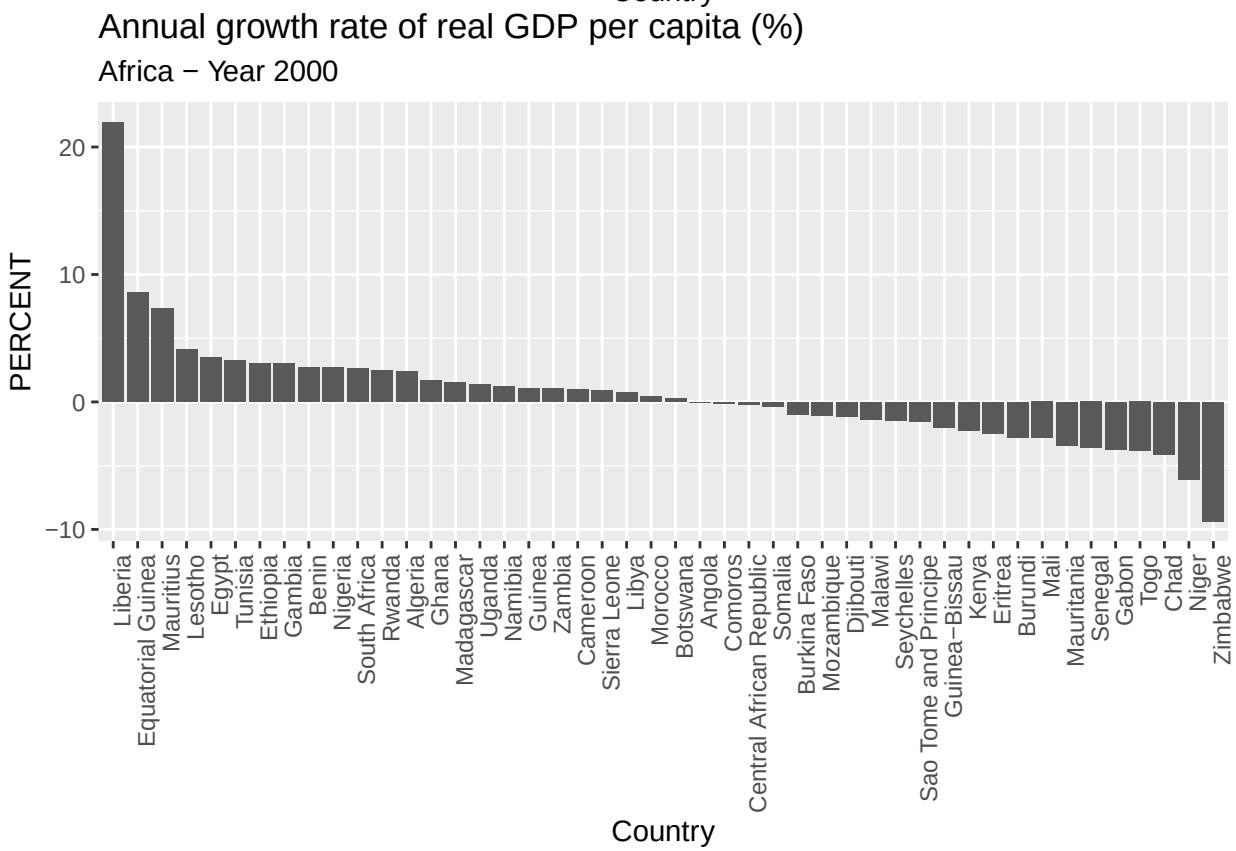
```
  title <- filtered_data$seriesdescription %>% na.omit() %>% unique()
  unit <- filtered_data$units %>% na.omit() %>% unique()

  # Plot graph
  filtered_data %>%
    ggplot(aes(x = reorder(country, -value), y = value)) +
    geom_bar(stat = "identity") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
    labs(title = paste(title),
         subtitle = paste(continent, '- Year', yr),
         x = "Country",
         y = paste(unit))

}
```

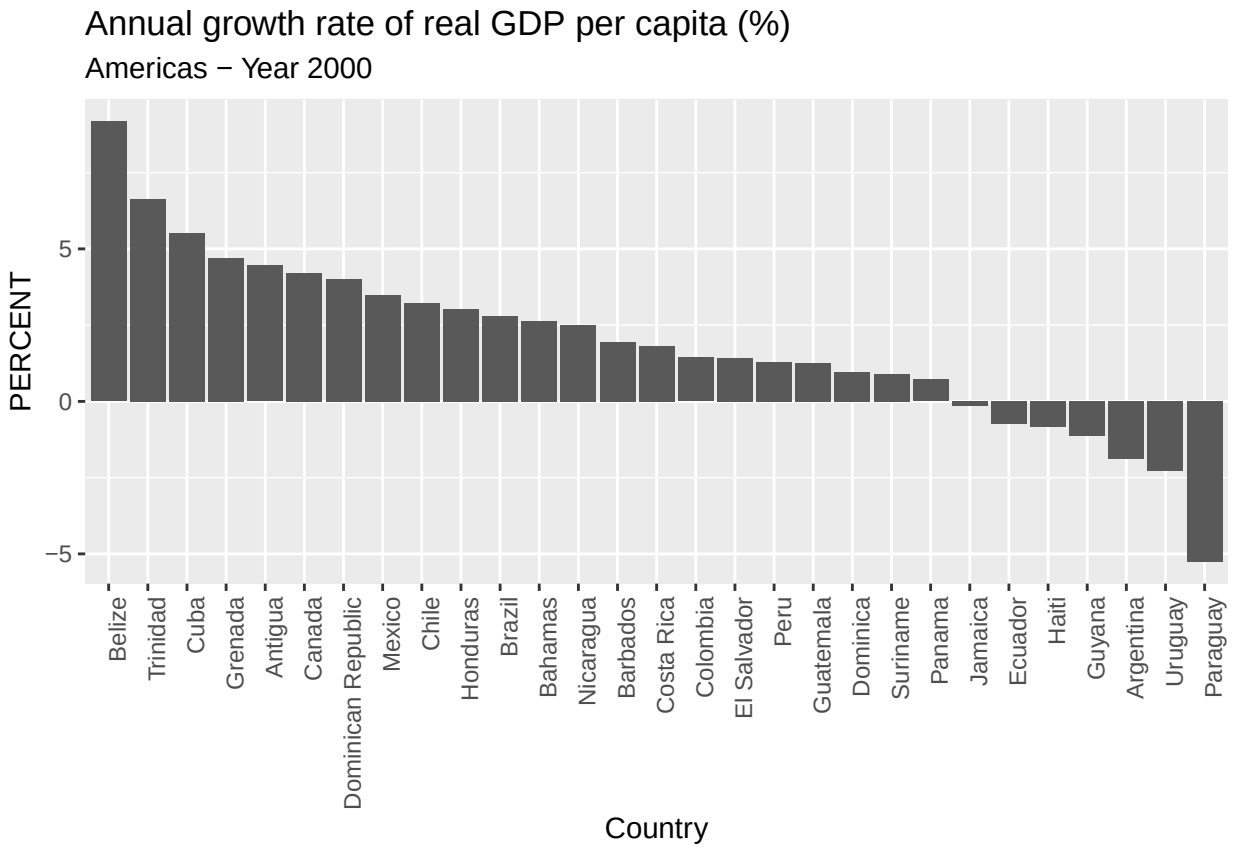
Here are the example charts generated for indicator 8.1.1 (annual RGDP growth) for year 2000, for each continent.

```
continents <- dat$region %>% unique()

for (cont in continents) {
  plot <- country_barchart(dat, "8.1.1", 2000, cont)
  print(plot)
}
```




Annual growth rate of real GDP per capita (%)
Asia − Year 2000
PERCENT
10
0
−10
Timor−Leste
Myanmar
Kazakhstan
Kiribati
China
Israel
Samoa
Tajikistan
Malaysia
Cambodia
United Arab Emirates
Singapore
Marshall Islands
Bhutan
Sri Lanka
Yemen
Turkmenistan
Nepal
Oman
Bangladesh
Qatar
Indonesia
Thailand
Saudi Arabia
Vanuatu
Tonga
Japan
Bahrain
Uzbekistan
Jordan
India
Philippines
Pakistan
Maldives
New Zealand
Australia
Mongolia
Kuwait
Lebanon
Iraq
Fiji
Palau
Papua New Guinea
Nauru
Afghanistan
Solomon Islands
Country


Annual growth rate of real GDP per capita (%)
Africa − Year 2000
PERCENT
20
10
0
−10
Liberia
Equatorial Guinea
Mauritius
Lesotho
Egypt
Tunisia
Ethiopia
Gambia
Benin
Nigeria
South Africa
Rwanda
Algeria
Ghana
Madagascar
Uganda
Namibia
Guinea
Zambia
Cameroon
Sierra Leone
Libya
Morocco
Botswana
Angola
Comoros
Central African Republic
Somalia
Burkina Faso
Mozambique
Djibouti
Malawi
Seychelles
Sao Tome and Principe
Guinea−Bissau
Kenya
Eritrea
Burundi
Mali
Mauritania
Senegal
Gabon
Togo
Chad
Niger
Zimbabwe
Country



2.3.7 Wide form time-series table

To inspect indicators numerically, a function was created that takes indicator and continent as input, returning a table in long form for all years there is data.

```
long_form_table <- function(data, ind, continent) {

  # Filter data accoring to input arguments and select relevant columns
  filtered_data <- data %>%
    select(indicator, seriesdescription, country, year, value, units, location, region) %>%
    filter(indicator == ind, region == continent)

  # Select only ALLAREA so as to not add up RURAL and URBAN for indicators 6.1.1 and 7.1.1.
  if (ind %in% c("6.1.1", "7.1.1")) {
    filtered_data <- filtered_data %>% filter(location == "ALLAREA")
  }

  # Cast data frame to wide form
  dcast(filtered_data, country ~ year)

}
```

Below follows a selection of examples using this function. While all years are recorded in the variable, a smaller selection of years is displayed in the report such that the tables fit on the page.

Proportion of individuals using the Internet in the Americas

```
internet_long_americas <- long_form_table(dat, "17.8.1", "Americas")
internet_long_americas[,c(1, 7:17)]
```

country	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Antigua	27.0	30.0	34.0	38.0	42.0	47.0	52.0	58.0	63.4	67.8	70.0
Argentina	17.7	20.9	25.9	28.1	34.0	45.0	51.0	55.8	59.9	64.7	68.0
Bahamas	25.0	26.0	27.0	31.5	33.9	43.0	65.0	71.7	72.0	76.9	78.0
Barbados	52.5	55.3	58.2	61.4	64.7	65.1	66.5	71.2	71.8	75.2	76.1
Belize	17.0	24.0	24.6	26.3	27.2	28.2	30.7	31.0	33.6	38.7	41.6
Brazil	21.0	28.2	30.9	33.8	39.2	40.7	45.7	48.6	51.0	54.6	58.3
Canada	71.7	72.4	73.2	76.7	80.3	80.3	83.0	83.0	85.8	87.1	88.5
Chile	31.2	34.5	35.9	37.3	41.6	45.0	52.2	55.1	58.0	61.1	64.3
Colombia	11.0	15.3	21.8	25.6	30.0	36.5	40.4	49.0	51.7	52.6	55.9
Costa Rica	22.1	25.1	28.4	32.3	34.3	36.5	39.2	47.5	46.0	53.0	59.8
Cuba	9.7	11.2	11.7	12.9	14.3	15.9	16.0	21.2	27.9	29.1	37.3
Dominica	38.5	39.4	40.3	41.2	42.0	47.5	48.6	49.8	51.0	57.5	65.0
Dominican Republic	11.5	14.8	17.7	20.8	27.7	31.4	38.0	42.3	45.9	49.6	54.2
Ecuador	6.0	7.2	10.8	18.8	24.6	29.0	31.4	35.1	40.3	45.6	48.9
El Salvador	4.2	5.5	6.1	10.1	12.1	15.9	18.9	20.3	23.1	24.8	26.8
Grenada	20.5	21.4	22.3	23.2	24.1	27.0	30.0	32.0	35.0	51.6	53.8
Guatemala	5.7	6.5	7.3	8.3	9.3	10.5	12.3	16.0	19.7	23.4	28.8
Guyana	NA	NA	13.8	18.2	23.9	29.9	30.0	30.5	31.0	32.0	34.0
Haiti	6.4	6.8	7.2	7.6	8.1	8.4	9.0	9.8	10.6	11.4	12.2
Honduras	6.5	7.8	9.4	9.6	9.8	11.1	15.9	18.1	17.8	19.1	27.6
Jamaica	12.8	16.4	21.1	23.6	24.3	27.7	37.4	33.8	37.1	40.4	42.2
Mexico	17.2	19.5	20.8	21.7	26.3	31.1	37.2	39.8	43.5	44.4	57.4
Nicaragua	2.6	2.8	3.9	5.3	7.3	10.0	10.6	13.5	15.5	17.6	19.7
Panama	11.5	17.3	22.3	33.8	39.1	40.1	42.7	40.3	44.0	44.9	51.2
Paraguay	7.9	8.0	11.2	14.3	18.9	19.8	24.8	29.3	36.9	43.0	48.4
Peru	17.1	20.7	25.2	30.6	31.4	34.8	36.0	38.2	39.2	40.2	40.9
Suriname	6.4	9.5	14.1	21.1	31.4	31.6	32.0	34.7	37.4	40.1	42.8
Trinidad	29.0	30.0	32.3	34.8	44.3	48.5	55.2	59.5	63.8	65.1	69.2
Uruguay	20.1	29.4	34.0	39.3	41.8	46.4	51.4	54.5	57.7	61.5	64.6

Proportion of individuals with access to electricity in Asia

```
electricity_long_asia <- long_form_table(dat, "7.1.1", "Asia")
electricity_long_asia[,c(1, 7:17)]
```

country	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Afghanistan	23.00	28.23	33.75	42.40	44.85	42.70	43.22	69.10	67.26	89.50	71.50
Australia	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Bahrain	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Bangladesh	44.23	50.53	46.50	51.25	53.63	55.26	59.60	60.88	61.50	62.40	68.20
Bhutan	59.81	61.18	71.80	70.01	74.46	73.28	83.41	91.50	92.40	96.91	98.42
Cambodia	20.50	28.08	30.18	26.40	34.44	31.10	38.78	40.96	43.16	56.10	47.57
China	97.74	98.07	98.40	98.75	99.12	99.70	99.75	99.92	99.98	100.00	100.00
Fiji	82.34	83.74	88.90	86.16	88.07	89.55	91.05	92.55	91.75	95.59	97.12
India	66.93	67.90	69.99	71.55	75.00	76.30	67.60	79.90	79.60	81.24	88.00
Indonesia	86.19	90.62	91.10	92.73	93.55	94.15	94.83	96.00	96.46	97.01	97.54

country	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Iraq	97.29	97.62	98.10	97.94	98.21	98.49	98.00	99.30	99.38	99.66	99.86
Israel	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Japan	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Jordan	99.05	99.12	98.80	99.31	99.40	99.57	99.72	99.50	99.94	99.99	100.00
Kazakhstan	99.22	99.77	99.33	99.42	100.00	99.64	99.80	99.87	99.95	99.99	100.00
Kiribati	69.68	64.19	66.19	68.22	64.19	63.17	74.41	76.51	78.61	80.72	90.56
Kuwait	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Lebanon	99.57	99.38	99.90	99.90	99.88	99.95	99.68	100.00	100.00	100.00	100.00
Malaysia	98.02	98.24	98.48	98.73	99.30	99.29	99.57	99.80	99.93	99.99	100.00
Maldives	91.18	92.32	93.49	94.68	99.90	97.12	98.25	99.16	99.71	100.00	99.99
Marshall Islands	75.77	77.27	72.30	80.32	81.88	83.46	89.92	86.66	88.27	89.89	91.51
Mongolia	86.20	76.30	76.78	77.28	77.80	78.50	72.06	79.45	81.20	80.61	81.19
Myanmar	47.82	48.58	49.35	50.15	50.96	48.80	52.64	53.51	54.38	52.00	60.50
Nauru	99.51	99.41	99.80	99.25	99.21	99.18	99.00	99.16	99.16	99.17	99.00
Nepal	47.14	51.20	55.02	58.99	62.98	66.99	76.30	75.06	79.10	84.90	87.21
New Zealand	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Oman	NA	NA	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Pakistan	83.85	83.85	89.20	86.79	90.73	89.81	91.37	92.90	93.60	96.02	93.50
Palau	98.92	98.37	98.39	98.44	98.50	98.58	98.68	97.63	98.91	99.81	99.16
Papua New Guinea	15.25	12.40	16.50	17.16	17.84	19.50	19.25	19.97	20.71	21.45	22.19
Philippines	78.62	79.66	80.72	83.30	82.91	84.03	85.17	86.32	87.50	88.65	89.08
Qatar	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Samoa	91.98	96.37	93.77	94.69	97.90	96.61	96.67	98.50	99.26	97.90	99.94
Saudi Arabia	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Singapore	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Solomon Islands	19.17	14.40	12.90	26.83	21.20	32.04	34.66	37.30	44.70	42.60	55.10
Sri Lanka	77.40	78.97	80.00	82.17	83.81	85.30	87.76	88.81	90.20	92.19	93.89
Tajikistan	99.30	98.91	99.02	99.15	99.00	99.46	99.64	99.10	99.92	99.98	100.00
Thailand	92.33	99.15	94.42	95.50	98.96	99.70	98.72	99.11	99.86	99.98	99.60
Timor-Leste	32.99	35.68	36.60	41.10	43.84	38.00	49.39	52.17	54.97	57.78	67.28
Tonga	88.71	92.30	90.07	90.78	91.51	92.26	92.18	92.80	94.59	95.38	96.18
Turkmenistan	99.60	99.84	99.64	99.68	99.75	100.00	99.91	99.96	99.99	100.00	100.00
United Arab Emirates	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Uzbekistan	99.76	99.49	99.84	99.90	99.96	99.99	100.00	100.00	100.00	100.00	100.00
Vanuatu	29.49	26.44	33.62	33.87	33.29	36.90	38.43	39.98	31.70	43.10	44.67
Yemen	56.42	55.80	59.04	60.39	61.75	63.13	64.53	65.94	75.60	66.10	70.21

Renewable energy share as percentage of total consumption in Africa

```
renewables_long_africa <- long_form_table(dat, "7.2.1", "Africa")
renewables_long_africa[,c(1, 7:17)]
```

country	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Algeria	0.58	0.41	0.41	0.30	0.31	0.26	0.18	0.19	0.14	0.07	0.06
Angola	70.95	65.02	61.60	58.11	55.75	54.19	52.72	52.25	50.69	50.80	49.57
Benin	59.19	57.30	54.49	54.76	52.83	48.12	49.45	51.15	52.71	52.54	50.86
Botswana	30.76	29.37	27.67	27.16	28.28	30.19	30.00	29.75	29.22	28.80	28.88
Burkina Faso	86.54	84.62	82.43	82.94	83.60	81.45	80.57	77.18	75.43	75.24	74.17
Burundi	97.29	96.86	96.89	96.85	96.92	96.76	96.59	93.96	94.07	94.60	95.68
Cameroon	86.23	85.43	80.71	80.71	79.09	78.60	78.37	78.05	77.61	76.98	76.54

country	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Central African Republic	85.65	84.80	80.86	80.58	80.58	79.81	78.77	78.03	77.08	76.90	76.57
Chad	91.73	91.81	90.71	90.07	91.05	90.79	90.60	90.65	89.46	89.24	89.36
Comoros	44.28	43.37	50.11	50.95	49.82	46.41	47.79	48.01	44.08	46.62	45.33
Djibouti	33.09	33.01	32.52	33.14	33.31	34.43	34.44	34.07	34.01	34.15	15.38
Egypt	6.49	6.23	6.09	5.97	5.76	5.72	5.58	5.55	5.71	5.87	5.71
Equatorial Guinea	7.86	6.63	5.68	6.31	6.10	5.95	5.64	5.35	5.96	7.64	7.82
Eritrea	74.86	78.56	80.02	81.90	81.81	81.25	80.75	80.23	80.23	79.93	79.77
Ethiopia	95.13	94.94	94.57	94.52	94.60	94.52	94.01	93.75	93.00	92.07	92.16
Gabon	85.24	86.15	87.35	87.38	88.10	85.88	79.77	75.51	82.23	81.35	82.01
Gambia	60.04	59.27	56.73	55.75	53.40	54.71	54.16	54.22	56.64	51.93	51.51
Ghana	61.43	58.56	54.53	55.86	49.91	49.78	48.01	44.48	44.04	45.05	41.41
Guinea	81.56	81.06	80.36	80.15	79.25	75.71	74.75	76.37	78.53	78.52	76.27
Guinea-Bissau	88.53	88.29	87.90	88.42	87.89	87.81	87.72	87.61	87.42	87.06	86.85
Kenya	81.23	79.77	80.26	80.48	78.15	76.27	77.08	78.50	77.14	75.52	72.66
Lesotho	55.31	55.46	55.33	54.99	53.98	53.45	53.14	52.41	52.30	51.39	52.14
Liberia	88.23	87.63	89.85	91.49	91.51	89.21	87.37	85.18	86.99	83.94	83.85
Libya	1.81	1.88	2.00	1.88	1.72	1.57	2.23	1.91	1.72	1.75	1.97
Madagascar	80.47	84.34	83.34	82.33	83.18	81.93	78.84	76.40	72.74	72.03	70.17
Malawi	83.80	83.34	84.09	81.53	83.36	79.47	79.66	80.37	79.87	79.95	83.65
Mali	85.80	82.67	78.89	76.24	72.71	69.13	66.65	64.93	62.41	64.59	61.53
Mauritania	38.77	39.12	35.63	35.80	33.43	34.00	33.28	31.84	34.20	32.02	32.16
Mauritius	17.74	16.49	15.77	12.82	12.87	13.66	12.14	11.61	11.44	10.63	11.54
Morocco	20.66	18.65	16.97	15.12	14.76	14.41	12.35	11.40	12.05	11.72	11.32
Mozambique	93.62	93.29	92.14	92.43	91.67	91.30	90.46	90.82	90.32	88.86	86.40
Namibia	32.06	32.86	32.77	27.03	27.08	26.37	27.94	28.56	28.27	27.62	26.47
Niger	87.60	87.48	86.22	86.11	83.30	80.71	77.27	72.75	78.73	78.14	78.94
Nigeria	84.07	86.14	87.38	86.35	88.83	86.78	86.15	86.45	86.78	87.30	86.64
Rwanda	89.24	90.59	90.31	90.63	90.54	90.66	89.64	88.79	88.47	88.18	86.66
Sao Tome and Principe	47.94	47.33	47.69	46.04	45.41	43.76	43.12	41.25	41.40	41.81	41.06
Senegal	41.23	44.20	40.63	40.47	51.96	50.26	48.66	50.37	43.59	43.36	42.71
Seychelles	0.38	0.35	0.43	0.38	0.51	0.63	0.96	0.63	1.29	1.30	1.35
Sierra Leone	87.25	83.39	85.56	85.27	85.39	84.18	80.35	78.43	75.22	73.05	77.66
Somalia	93.30	93.47	93.02	93.37	93.63	93.57	93.65	93.89	94.23	94.43	94.29
South Africa	16.25	16.37	15.81	15.99	15.57	17.08	16.98	16.63	16.10	16.59	17.15
South Sudan	0.00	0.00	0.00	0.00	0.00	0.00	0.00	30.20	29.70	29.83	39.07
Sudan	72.43	69.35	65.72	63.66	63.28	61.44	63.97	59.82	61.31	62.44	61.60
Togo	77.22	79.66	79.94	76.52	63.40	65.83	69.16	72.74	72.40	72.16	71.26
Tunisia	14.18	14.32	14.32	14.61	16.07	12.69	13.33	13.08	13.25	12.93	12.56
Uganda	92.81	92.42	92.31	92.28	92.44	91.61	91.09	91.27	90.75	90.22	89.06
Zambia	89.00	89.84	92.96	92.18	92.09	92.10	90.90	88.63	88.48	88.03	87.99
Zimbabwe	80.15	78.65	78.08	82.03	83.15	82.88	79.57	78.02	79.56	81.05	81.80

2.4 Forecasting

Due to the perhaps over-ambitious visualization efforts given my skills and limited time, this report regrettably has not contributed much in terms of forecasting. I believe a solid foundation has been created to enable powerful forecasting techniques for future analysis. It is certainly my intention to keep improving this project, especially with regard to forecasting values, as well as visualizing those forecasts. A simple example, and perhaps the most obvious starting point, has been included in the report to demonstrate some of the elementary forecasting techniques available.

2.4.1 Fitting a linear model using `lm`

Example for Afghanistan on indicator 7.1.1 (population proportion with access to electricity).

```
lm_711_afg <- dat %>%
  filter(country == "Afghanistan", indicator == "7.1.1", location == "ALLAREA") %>%
  select(country, indicator, year, value)

fit <- lm(value ~ year, data = lm_711_afg)

summary(fit)$coeff
```

```
##                   Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -11011.825490 654.756545 -16.81820 3.821371e-11
## year             5.503799   0.326073  16.87904 3.629168e-11
```

We can use this linear model to predict the future values until 2030, under the assumption of linearity:

```
new <- data.frame(year = 2001:2030)

predict(fit, new)
```

```
##          1          2          3          4          5          6
##   1.276348   6.780147  12.283946  17.787745  23.291544  28.795343
##          7          8          9         10         11         12
##  34.299142  39.802941  45.306740  50.810539  56.314338  61.818137
##         13         14         15         16         17         18
##  67.321936  72.825735  78.329534  83.833333  89.337132  94.840931
##         19         20         21         22         23         24
## 100.344730 105.848529 111.352328 116.856127 122.359926 127.863725
##         25         26         27         28         29         30
## 133.367525 138.871324 144.375123 149.878922 155.382721 160.886520
```

The best fit of the model suggests that 100% of Afghanistan's will have access to electricity by 2019. Using the lower confidence bound gives us a more conservative prediction of 2023:

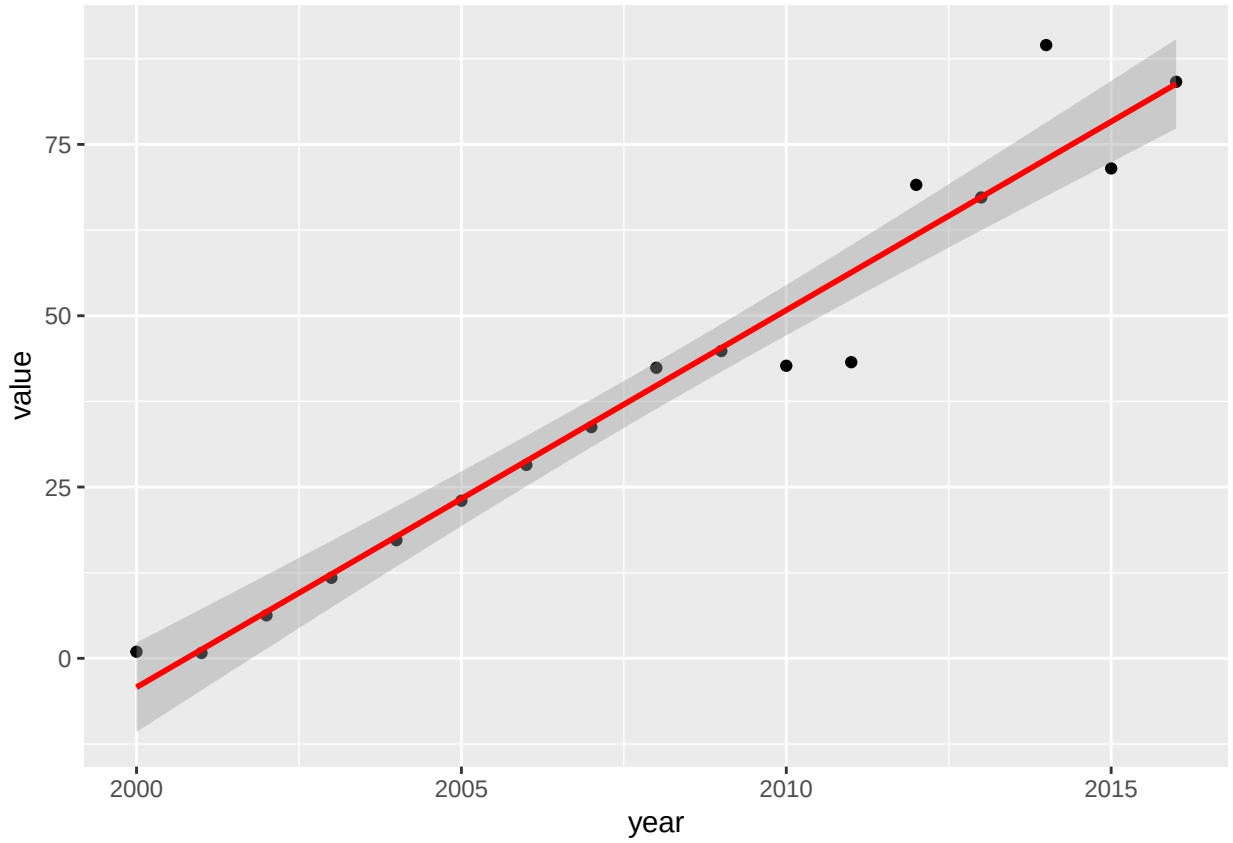
```
predict(fit, new, interval = "prediction")
```

```
##           fit        lwr       upr
## 1    1.276348 -13.966371  16.51907
## 2    6.780147  -8.255178  21.81547
## 3   12.283946  -2.573631  27.14152
## 4   17.787745   3.077195  32.49830
## 5   23.291544   8.696373  37.88672
## 6   28.795343  14.283147  43.30754
## 7   34.299142  19.836959  48.76133
## 8   39.802941  25.357468  54.24841
## 9   45.306740  30.844557  59.76892
## 10  50.810539  36.298343  65.32274
## 11  56.314338  41.719167  70.90951
## 12  61.818137  47.107587  76.52869
```

```
## 13  67.321936  52.464359  82.17951
## 14  72.825735  57.790410  87.86106
## 15  78.329534  63.086815  93.57225
## 16  83.833333  68.354767  99.31190
## 17  89.337132  73.595543 105.07872
## 18  94.840931  78.810481 110.87138
## 19 100.344730  84.000952 116.68851
## 20 105.848529  89.168334 122.52873
## 21 111.352328  94.313994 128.39066
## 22 116.856127  99.439273 134.27298
## 23 122.359926 104.545469 140.17438
## 24 127.863725 109.633832 146.09362
## 25 133.367525 114.705552 152.02950
## 26 138.871324 119.761759 157.98089
## 27 144.375123 124.803515 163.94673
## 28 149.878922 129.831821 169.92602
## 29 155.382721 134.847611 175.91783
## 30 160.886520 139.851756 181.92128
```

A plot of the `lm` model can be created with `ggplot`.

```
ggplot(lm_711_afg, aes(x = year, y = value)) +
  geom_point() +
  stat_smooth(method = "lm", col = "red")
```



3 Results

This project has successfully fetched data from all the desired sources and created a unified dataset that is convenient for exploration and analysis. In my opinion, the key achievement of this project is the creation of functions that visualize the results from the dataset. These functions also serves as a baseline for communicating the results of analysis and forecasting. Due to time constraints and the significant effort I have needed to put into the project, I have unfortunately not been able to produce much forecasting. One of the main goals of the projects were to use trend analysis to predict future results. While a simple example of such analysis is included, there is much potential to improve this aspect of the project.

Table 6: Achievement of goals

Goal	**Status**
Programmatically download data from the relevant sources	Achieved
Clean and merge data into unified datasets	Achieved
Communicate reported results graphically	Achieved[(1)]
Use trend analysis to predict future results	Lacking

[(1)] More visualization techniques can always be added.

4 Conclusion

The desired data has been downloaded, cleaned and explored in detail. Visualization of this dataset were considered important from the outset and much effort has been spent on creating good graphics. This has come at the expense of forecasting, which will require more work. The project required more time than anticipated but has been very fun and immensely rewarding; I have learned a lot and only regret not having more time to further the project before submission. I intend to continue working on improving and adding visualizations, and developing forecasting methods including the visualization of those forecasts. I also plan to add a function to pull new data and record the accuracy of the forecasting methods against the new data.