Ggplot2 Lab – a tribute to Hans Rosling

Marcin Kierczak

23 October 2016

Hans Rosling (27 July 1948 – 7 February 2017) was a Swedish physician, academic, statistician, and public speaker. He was the Professor of International Health at Karolinska Institutet and was the co-founder and chairman of the Gapminder Foundation, which developed the Trendalyzer software system. He held presentations around the world, including several TED Talks in which he promoted the use of data to explore development issues. source: Wikipedia

The Lab

In this excercise we will plot a Hans Rosling's Gapminder-style plot using *ggplot2* library. It will not be as nice as the original, but it will be a good starting point for making more advanced graphics.

We have chosen to plot the so-called Preston curve, a curve reflecting life expectancy vs. *per capita* gross domestic product (GDP). To make the task more interesting, we will obtain the necessary data straight from Wikipedia, using a technique that is broadly called *web scraping*.

The first thing is to load a couple of packages that are necessary to do the analyses:

- rvest – will provide tools for harvesting (scraping) Wikipedia pages,
- stringr – will provide tools for string operations and it also imports the *magrittr* package that provides a nice new operator %>% called the pipe operator. More about it below,
- ggrepel – provides a mechanism of resolving overlapping text on ggplot2 plots,
- ggplot2 – for obvious reasons – plotting.

```
library(rvest)
library(stringr)
library(ggrepel)
library(ggplot2)
```

Now, we need to import some data. Let's start with GDP by country.

```
# URL to the Wikipedia entry
url <- "https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)_per_capita"

# Use pipe operator to, in sequence, read  HTML source for the Wikipedia entry
# from the provided URL and fish-out all tables from there. The result, a list of
# tables will be assigned to variable data
data <- url %>% read_html() %>% html_nodes('table')

# Now, just see what the data variable contains. We see, we want the 3rd table.
gdp <- html_table(data[[3]], dec='.', header=T)

# Country names are a bit messy due to encoding, we fix this with the iconv
# function
gdp$Country <- iconv(gdp$Country, "UTF-8", "ascii", sub='')

# Name all the rows by the country
rownames(gdp) <- gdp$Country
```

```
# Rename the 3rd column for the sake of simplicity
colnames(gdp)[3] <- 'GDP'

# Convert GDP to numeric (remove non-decimal commas too)
gdp$GDP <- as.numeric(gsub(',', '', gdp$GDP))
```

Note the function of the %>% (pipe) operator. The **left() %>% right() -> var2** is equivalent to two lines of code:

```
var1 <- left()
var2 <- right(var1)
```

You simply feed the result returned by the function to the left to the input of the function to the right.

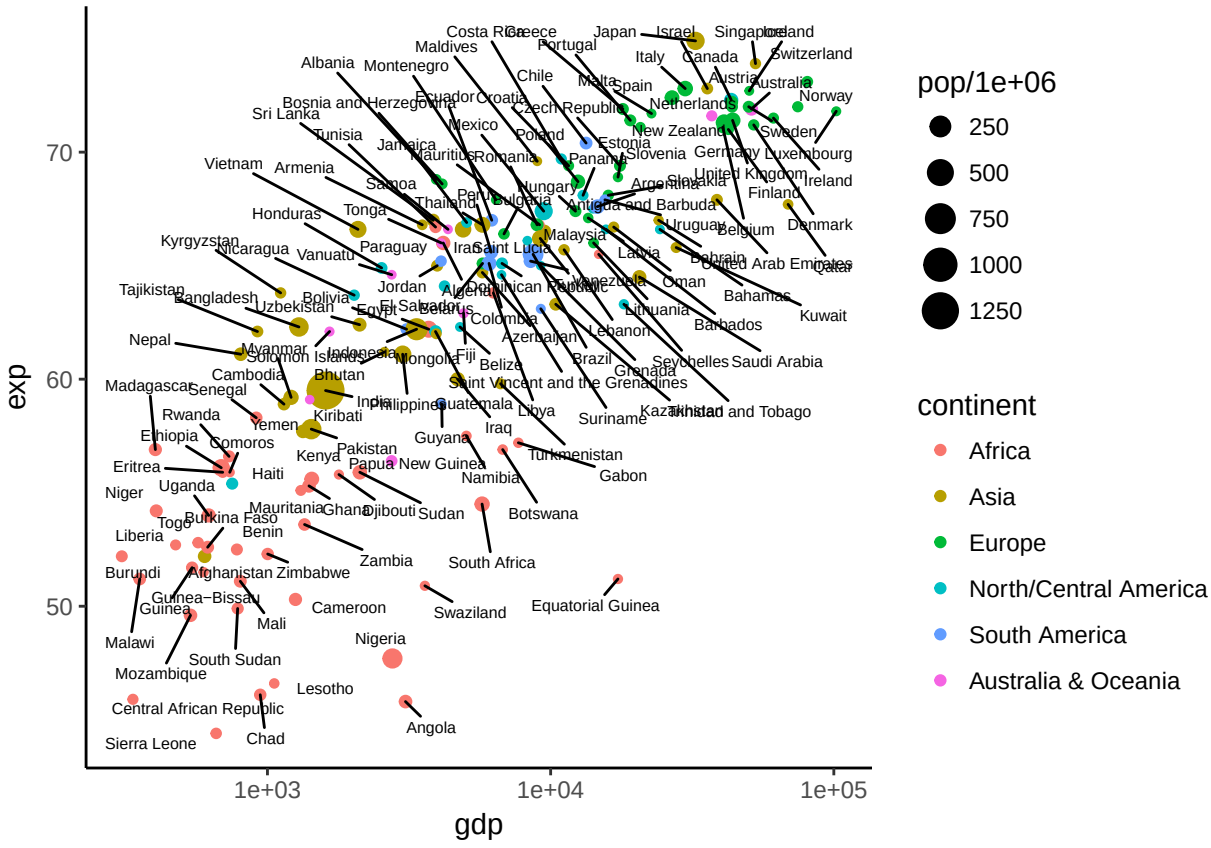
```
url <- "https://en.wikipedia.org/wiki/List_of_countries_by_life_expectancy"
data <- url %>% read_html() %>% html_nodes('table')
life.exp <- html_table(data[[1]], dec='.', header=T)
life.exp$Country <- iconv(life.exp$Country, "UTF-8", "ascii",sub='')
colnames(life.exp) <- gsub('\n', '', colnames(life.exp))

url <- 'https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population'
data <- url %>% read_html() %>% html_nodes('table')
pop <- html_table(data[[2]], dec='.', header=T)
pop$`Country (or dependent territory)` <- iconv(pop$`Country (or dependent territory)`, "UTF-8", "ascii
colnames(pop) <- gsub('\n', '', colnames(pop))
colnames(pop)[2] <- 'Country'
pop$Population <- as.numeric(gsub(',', '', pop$Population))

# Continent
url <- 'https://en.wikipedia.org/wiki/List_of_sovereign_states_and_dependent_territories_by_continent'
data <- url %>% read_html() %>% html_nodes('table')
africa <- data.frame(continent="Africa", Country=html_table(data[[4]], dec='.', header=T)[,2])
asia <- data.frame(continent="Asia", Country=html_table(data[[6]], dec='.', header=T)[,2])
europe <- data.frame(continent="Europe", Country=html_table(data[[8]], dec='.', header=T)[,2])
americaNC <- data.frame(continent="North/Central America", Country=html_table(data[[10]], dec='.', head
americaS <- data.frame(continent="South America", Country=html_table(data[[12]], dec='.', header=T)[,2]
ausocean <- data.frame(continent="Australia & Oceania", Country=html_table(data[[14]], dec='.', header=
continents <- rbind(africa, asia, europe, americaNC, americaS, ausocean)
rownames(continents) <- continents$Country

# Merge all data
dat <- merge(gdp[,c(2:3)], life.exp[,c(1,5,7,9)], by='Country')
dat <- merge(dat, pop[,c(2,3)], by='Country')
dat$continent <- continents[dat$Country,1]

dat2 <- data.frame(country=dat$Country, gdp=(dat$GDP), exp=dat$`Both sexes lifeexpectancy (HALE)`, pop=
g <- ggplot(dat2, aes(x=gdp, y=exp, label=country, size=pop/1e6)) +
  geom_point(aes(col=continent)) +
  geom_text_repel(size=2, box.padding = unit(0.5, "lines"), force=1, max.iter = 100) +
  scale_x_log10() +
  theme_classic()
g
```


exp
gdp
70
60
50
1e+03
1e+04
1e+05
Costa Rica
Greece
Japan
Israel
Singapore
Iceland
Maldives
Portugal
Italy
Canada
Switzerland
Albania
Montenegro
Chile
Malta
Spain
Austria
Australia
Bosnia and Herzegovina
Ecuador
Croatia
Czech Republic
Netherlands
Norway
Sri Lanka
Tunisia
Mexico
New Zealand
Sweden
Jamaica
Mauritius
Poland
Estonia
Romania
Panama
Slovenia
Germany
Luxembourg
Vietnam
Armenia
United Kingdom
Ireland
Samoa
Thailand
Peru
Hungary
Bulgaria
Argentina
Slovakia
Finland
Honduras
Tonga
Antigua and Barbuda
Uruguay
Belgium
Denmark
Kyrgyzstan
Nicaragua
Vanuatu
Paraguay
Iran
Saint Lucia
Malaysia
Latvia
Bahrain
United Arab Emirates
Qatar
Tajikistan
Jordan
Dominican Republic
Venezuela
Oman
Bangladesh
Bolivia
Uzbekistan
El Salvador
Belarus
Lithuania
Bahamas
Egypt
Colombia
Barbados
Kuwait
Nepal
Azerbaijan
Lebanon
Myanmar
Solomon Islands
Indonesia
Mongolia
Fiji
Belize
Brazil
Grenada
Seychelles
Saudi Arabia
Cambodia
Bhutan
Saint Vincent and the Grenadines
Madagascar
Senegal
India
Philippines
Guatemala
Libya
Suriname
Kazakhstan
Trinidad and Tobago
Rwanda
Kiribati
Yemen
Iraq
Ethiopia
Comoros
Guyana
Pakistan
Kenya
Turkmenistan
Eritrea
Haiti
Papua New Guinea
Namibia
Gabon
Niger
Uganda
Mauritania
Ghana
Djibouti
Sudan
Botswana
Togo
Burkina Faso
Liberia
Benin
Zambia
South Africa
Burundi
Afghanistan
Zimbabwe
Guinea−Bissau
Guinea
Cameroon
Swaziland
Equatorial Guinea
Mali
Malawi
Nigeria
South Sudan
Mozambique
Lesotho
Central African Republic
Angola
Sierra Leone
Chad
pop/1e+06
250
500
750
1000
1250
continent
Africa
Asia
Europe
North/Central America
South America
Australia & Oceania