R is an incredible tool for reproducible research. In the present series of blog posts I want to show how one can easily acquire data within an R session, documenting every step in a fully reproducible way. There are numerous data acquisition options for R users. Of course, I do not attempt to show all the data possibilities and tend to focus mostly on demographic data. If your prime interest lies outside human population statistics, it’s worth checking the amazing [Open Data Task View](https://github.com/ropensci/opendata).

The series consists of four posts:

* [Loading prepared datasets](https://ikashnitsky.github.io/2017/data-acquisition-one/)
* [Accessing popular statistical databases](https://ikashnitsky.github.io/2017/data-acquisition-two)
* **Demographic data sources**
* Getting spatial data

For each of the data acquisition options I provide a small visualization use case.

**Human Mortality Database**

When it comes to testing the big questions of human population dynamics, there is no more reliable data source than [Human Mortality Database](http://www.mortality.org/). This database is run by demographers who use state-of-the-art methodology to overcome issues in the data. As the result, the estimates are as precise as possible. Their [methods protocol](http://www.mortality.org/Public/Docs/MethodsProtocol.pdf) is a masterpiece of demographic data processing. On the down side, the data of decent enough quality is available for only a bunch of countries. To explore the data I highly recommend [Human Mortality Database Explorer][exp] by [Jonas Schöley](https://twitter.com/jschoeley).

Thanks to [Tim Riffe](https://twitter.com/timriffe1)’s HMDHFDplus package, one can now download HMD data with just a couple of lines of R code. Please note that an account at [mortality.org](http://www.mortality.org/) is needed in order to download data. As one may guess from the package name, it also helps to grab data from equally brilliant [Human Fertility Database](http://www.humanfertility.org/).

The following example is taken from my [earlier post](https://ikashnitsky.github.io/2017/hmd-all-sex-ratio/) (and updated a bit). I think it illustrates nicely the power of automated data acquisition in R. Here I am going to download one year population population structures for both males and females for each single country of HMD. If you want to reproduce the result, beware that the script will download a couple of dozens megabits of data. Then I will calculate and visualize as small multiples sex ratios in all countries along age dimension. Sex ratios reflect the two basic regularities of human demographics: 1) there are always more boys being born; 2) males experience higher mortality throughout their life-course. Besides some artificial and well known exceptions, sex ratio at birth does not vary dramatically and is more or less constant at the level of 105-106 boys per 100 girls. Hence, differences in the sex ratio profiles of countries mainly reflect gender gap in mortality.

# load required packages

library(HMDHFDplus)

library(tidyverse)

library(purrr)

# help function to list the available countries

country <- getHMDcountries()

# remove optional populations

opt\_pop <- c("FRACNP", "DEUTE", "DEUTW", "GBRCENW", "GBR\_NP")

country <- country[!country %in% opt\_pop]

# temporary function to download HMD data for a simgle county (dot = input)

tempf\_get\_hmd <- . %>% readHMDweb("Exposures\_1x1", ik\_user\_hmd, ik\_pass\_hmd)

# download the data iteratively for all countries using purrr::map()

exposures <- country %>% map(tempf\_get\_hmd)

# data transformation to apply to each county dataframe

tempf\_trans\_data <- . %>%

select(Year, Age, Female, Male) %>%

filter(Year %in% 2012) %>%

select(-Year) %>%

transmute(age = Age, ratio = Male / Female \* 100)

# perform transformation

df\_hmd <- exposures %>%

map(tempf\_trans\_data) %>%

bind\_rows(.id = "country")

# summarize all ages older than 90 (too jerky)

df\_hmd\_90 <- df\_hmd %>%

filter(age %in% 90:110) %>%

group\_by(country) %>%

summarise(ratio = ratio %>% mean(na.rm = T)) %>%

ungroup() %>%

transmute(country, age = 90, ratio)

# insert summarized 90+

df\_hmd\_fin <- bind\_rows(df\_hmd %>% filter(!age %in% 90:110), df\_hmd\_90)

# finaly - plot

df\_hmd\_fin %>%

ggplot(aes(age, ratio, color = country, group = country))+

geom\_hline(yintercept = 100, color = "grey50", size = 1)+

geom\_line(size = 1)+

scale\_y\_continuous(limits = c(0, 120),

expand = c(0, 0),

breaks = seq(0, 120, 20))+

scale\_x\_continuous(limits = c(0, 90),

expand = c(0, 0),

breaks = seq(0, 80, 20))+

facet\_wrap(~country, ncol = 6)+

theme\_minimal(base\_family = "mono", base\_size = 15)+

theme(legend.position = "none",

panel.border = element\_rect(size = .5, fill = NA,

color = "grey50"))+

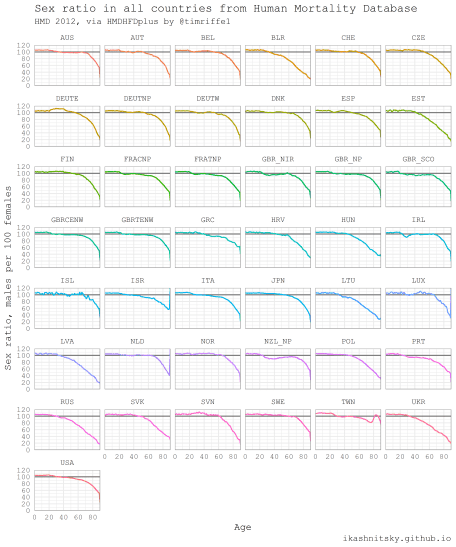
labs(x = "Age",

y = "Sex ratio, males per 100 females",

title = "Sex ratio in all countries from Human Mortality Database",

subtitle = "HMD 2012, via HMDHFDplus by @timriffe1",

caption = "ikashnitsky.github.io")

[](https://i1.wp.com/ikashnitsky.github.io/images/171210/hmd.png?ssl=1)

**United Nations World Population Prospects**

Population Department of the United Nations provides high quality population estimates for all countries of the world. They update estimates every 2-3 years and publish openly as an interactive report [World Population Prospects](https://esa.un.org/unpd/wpp/). One may find in these reports key highlights and, of course, rich data. The data is later wrapped in R packages called wpp20xx. Currently, the available packages are for the estimate updates 2008, 2010, 2012, 2015, and 2017. I will give here an example of wpp2015 use adapted from my earlier [post](https://ikashnitsky.github.io/2017/global-male-life-expectancy-convergence/).

Using ridgeplot, the amazing type of dataviz promoted ggridges package by [Claus Wilke](https://twitter.com/ClausWilke), I am going to show the impressive reduction of global inequality in male mortality that took place since 1950.

library(wpp2015)

library(tidyverse)

library(ggridges)

library(viridis)

# get the UN country names

data(UNlocations)

countries <- UNlocations %>% pull(name) %>% paste

# data on male life expectancy at birth

data(e0M)

e0M %>%

filter(country %in% countries) %>%

select(-last.observed) %>%

gather(period, value, 3:15) %>%

ggplot(aes(x = value, y = period %>% fct\_rev()))+

geom\_density\_ridges(aes(fill = period))+

scale\_fill\_viridis(discrete = T, option = "B", direction = -1,

begin = .1, end = .9)+

labs(x = "Male life expectancy at birth",

y = "Period",

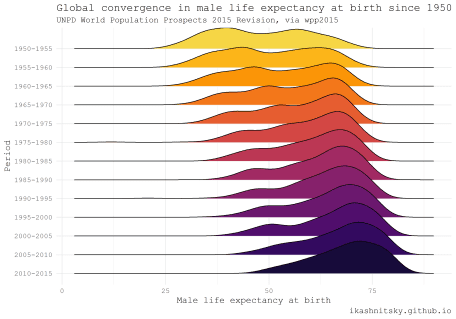
title = "Global convergence in male life expectancy at birth since 1950",

subtitle = "UNPD World Population Prospects 2015 Revision, via wpp2015",

caption = "ikashnitsky.github.io")+

theme\_minimal(base\_family = "mono")+

theme(legend.position = "none")

[](https://i2.wp.com/ikashnitsky.github.io/images/171210/wpp2015.png?ssl=1)

**European Social Survey (ESS)**

[European Social Survey](http://www.europeansocialsurvey.org/about/) provides uniquely rich nationally representative cross-county comparable information on the values of Europeans. Every two years a cross-sectional sample is taken in each participating country. All the data is easily available [upon registration](http://www.europeansocialsurvey.org/user/new). Datasets are distributed as SAS, SPSS, or STATA files. Thanks to [Jorge Cimentada](https://twitter.com/cimentadaj), these datasets are now easily available via ess package. I am going to visualize how respondents assessed their level of trust in police in all available countries at the latest round of the survey.

library(ess)

library(tidyverse)

# help gunction to see the available countries

show\_countries()

# check the available rounds for a selected country

show\_country\_rounds("Netherlands")

# get the full dataset of the last (8) round

df\_ess <- ess\_rounds(8, your\_email = ik\_email)

# select a variable and calculate mean value

df\_ess\_select <- df\_ess %>%

bind\_rows() %>%

select(idno, cntry, trstplc) %>%

group\_by(cntry) %>%

mutate(avg = trstplc %>% mean(na.rm = T)) %>%

ungroup() %>%

mutate(cntry = cntry %>% as\_factor() %>% fct\_reorder(avg))

df\_ess\_select %>%

ggplot(aes(trstplc, fill = avg))+

geom\_histogram()+

scale\_x\_continuous(limits = c(0, 11), breaks = seq(2, 10, 2))+

scale\_fill\_gradient("Average\ntrust\nscore",

low = "black", high = "aquamarine")+

facet\_wrap(~cntry, ncol = 6)+

theme\_minimal(base\_family = "mono")+

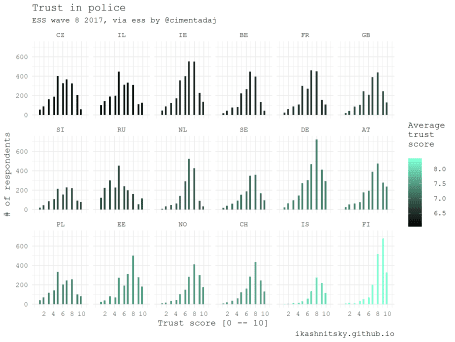
labs(x = "Trust score [0 -- 10]",

y = "# of respondents",

title = "Trust in police",

subtitle = "ESS wave 8 2017, via ess by @cimentadaj",

caption = "ikashnitsky.github.io")

[](https://i1.wp.com/ikashnitsky.github.io/images/171210/ess.png?ssl=1)

**American Community Survey and Census**

There are several packages that provide access to the US Census and ACS data. Perhaps the most convenient one is the recent tidycensus package by [Kyle Walker](https://twitter.com/kyle_e_walker). One extremely useful feature of this approach is the ability to download geodata along with stats in the form of simple features. Simple features, a revolutionary approach to deal with spatial data in R implemented in sf package by [Edzer Pebesma](https://twitter.com/edzerpebesma), allow to manage and visualize geodata tidy and efficiently. Note that in order to reproduce the following example one would have to install the development version of ggplot2.

Below I map median ages of census tracts population in Chicago based on the ACS estimates in 2015. To use tidycensus, an API key is required. API is instantly provided [upon registration](https://api.census.gov/data/key_signup.html).

library(tidycensus)

library(tidyverse)

library(viridis)

library(janitor)

library(sf)

# to use geom\_sf we need the latest development version of ggplot2

devtools::install\_github("tidyverse/ggplot2", "develop")

library(ggplot2)

# you need a personal API key, available free at

# https://api.census.gov/data/key\_signup.html

# normally, this key is to be stored in .Renviron

# see state and county codes and names

fips\_codes %>% View

# the available variables

load\_variables(year = 2015, dataset = "acs5") %>% View

# data on median age of population in Chicago

df\_acs <- get\_acs(

geography = "tract",

county = "Cook County",

state = "IL",

variables = "B01002\_001E",

year = 2015,

key = ik\_api\_acs,

geometry = TRUE

) %>% clean\_names()

# map the data

df\_acs %>%

ggplot()+

geom\_sf(aes(fill = estimate %>%

cut(breaks = seq(20, 60, 10))),

color = NA)+

scale\_fill\_viridis\_d("Median age", begin = .4)+

coord\_sf(datum = NA)+

theme\_void(base\_family = "mono")+

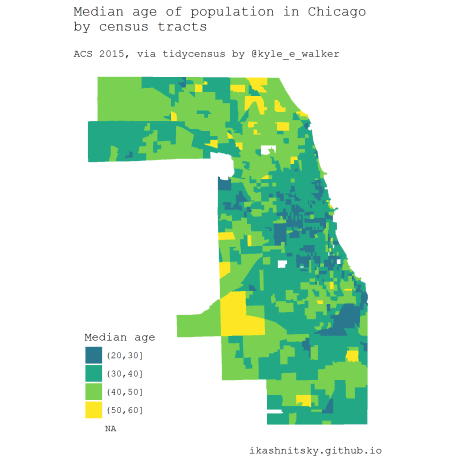
theme(legend.position = c(.15, .15))+

labs(title = "Median age of population in Chicago\nby census tracts\n",

subtitle = "ACS 2015, via tidycensus by @kyle\_e\_walker",

caption = "ikashnitsky.github.io",

x = NULL, y = NULL)

[](https://i0.wp.com/ikashnitsky.github.io/images/171210/tidycensus.png?ssl=1)

**Full reproducibility**

All the code chunks together can be found in [this gist](https://gist.github.com/ikashnitsky/2e4bb16e097b1a264eeeae13ca6d7ce3).