Nowadays (I’ve seen that word used *so* much in journal articles lately that I could not resist using it) I’m using world tile grids more frequently as the need arises to convey the state of exposure of various services at a global (country) scale. Given that necessity fosters invention it seemed that having a ggplot2 geom for world tile grids would make my life easier and also make work more efficient.

To that end, there’s a nascent ggplot2 extension package for making world tile grids called (uncreatively) [worldtilegrid[](https://gitlab.com/hrbrmstr/worldtilegrid)](https://gitlab.com/hrbrmstr/worldtilegrid). It’s also at GitHub if you’re more comfortable working from code from there.

The [README](https://gitlab.com/hrbrmstr/worldtilegrid#readme) has examples but we’ll walk through another one here and work with life expectancy data curated by [Our World in Data](https://ourworldindata.org/life-expectancy). Rather than draw out this post to a less tenable length with an explanation of how to pull the XHR JSON data into R, just grab the CSV linked with the visualization on that page and substitute the path in the code for where you stored it.

We do need to clean up this data a bit since it has some issues. Let’s do that and carve out some slices into two new data frames so we can work with the most recent curated year and some historical data:

library(hrbrthemes) # gitlab/github :: hrbrmstr/hrbrthemes (use devtools installers)

library(worldtilegrid) # gitlab/github :: hrbrmstr/worldtilegrid (use devtools installers)

library(tidyverse)

cols(

Entity = col\_character(),

Code = col\_character(),

Year = col\_integer(),

`Life expectancy (Clio-Infra up to 1949; UN Population Division for 1950 to 2015)` = col\_double()

) -> le\_cols

read\_csv("~/Downloads/life-expectancy.csv", col\_types = le\_cols) %>%

set\_names(c("country", "iso3c", "year", "life\_expectancy")) -> lifexp

# clean it up a bit since it's not great data and bust it into groups

# that match the vis on the site vs use continuous raw data, esp since

# the data is really just estimates

filter(lifexp, !is.na(iso3c)) %>%

filter(nchar(iso3c) == 3) %>%

mutate(

grp = cut(

x = life\_expectancy,

breaks = c(10, 20, 30, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85),

include.lowest = TRUE

)

) -> lifexp

last\_year <- filter(lifexp, year == last(year))

a\_few\_years <- filter(lifexp, year %in% c(seq(1900, 2015, 20), 2015))

Now we can use it to make some cartograms.

**Using the World Tile Grid geom**

You can [reference a previous post](https://rud.is/b/2018/06/07/making-world-tile-grid-grids/) for what it would normally take to create these tile grids. The idea was to simplify the API down to the caller just needing to specify some country names/ISO[23]Cs (to a country aesthetic) and a value (to a fill aesthetic) and let the package do the rest. To that end, here's how to display the life expectancy data for 2015:

ggplot(last\_year) +

geom\_wtg(aes(country = iso3c, fill = grp), border\_col = "#2b2b2b") + # yep, you can sum up the entire blog post to this one line

viridis::scale\_fill\_viridis(

name = "Life Expectancy", discrete = TRUE, na.value=alpha(ft\_cols$gray, 1/10),

drop = FALSE

) +

coord\_equal() +

labs(

x=NULL, y=NULL,

title = "Life expectancy, 2015",

subtitle = "Shown is period life expectancy at birth. This corresponds to an estimate of the average number\nof years a newborn infant would live if prevailing patterns of mortality at the time of its birth were to\nstay the same throughout its life",

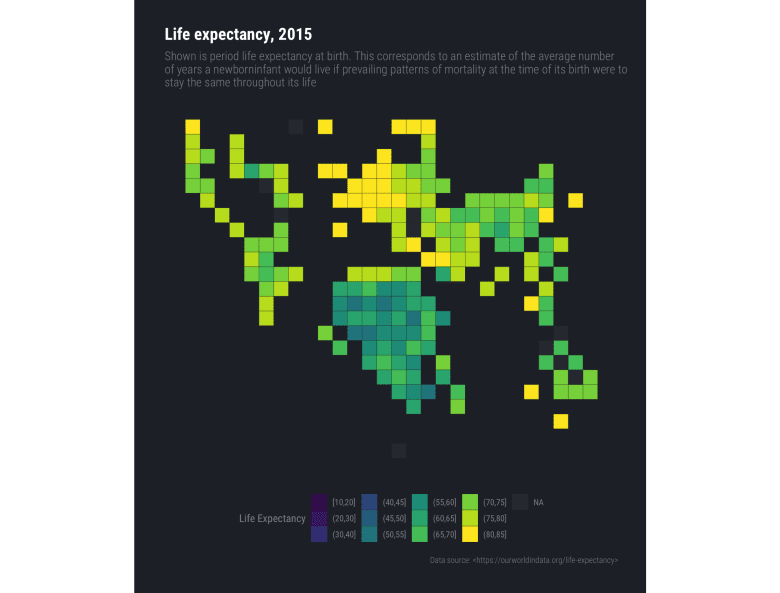
caption = "Data source: "

) +

theme\_ft\_rc(grid="") +

theme(axis.text=element\_blank()) +

theme(legend.position = "bottom")

[](https://rud.is/b/2018/08/27/simplifying-world-tile-grid-creation-with-geom_wtg/le-wtg-01/)

You can add labels via geom\_text() and use the wtg stat for it as it provides a number of computed variables you can work with:

* x,y: the X,Y position of the tile
* name: Country name (e.g. Afghanistan)
* country.code: ISO2C country code abbreviation (e.g. AF)
* iso\_3166.2: Full ISO 3166 2-letter abbreviation code (e.g. ISO 3166-2:AF)
* region: Region name (e.g. Asia)
* sub.region: Sub-region name (e.g. Southern Asia)
* region.code: Region code (e.g. 142)
* sub.region.code: Sub-region code (e.g. 034)

Labeling should be a deliberate decision and used sparingly/with care.

**Easier World Tile Grid Facets**

Making faceted charts is also straightforward. Just use ggplot2's faceting subsytem:

ggplot(a\_few\_years) +

geom\_wtg(aes(country = iso3c, fill = grp), border\_col = "#2b2b2b") +

viridis::scale\_fill\_viridis(

name = "Life Expectancy", discrete = TRUE, na.value=alpha(ft\_cols$gray, 1/10),

drop = FALSE

) +

coord\_equal() +

labs(

x=NULL, y=NULL,

title = "Life expectancy, 1900-2015 (selected years)",

subtitle = "Shown is period life expectancy at birth. This corresponds to an estimate of the average number of years a newborn infant\nwould live if prevailing patterns of mortality at the time of its birth were to stay the same throughout its life",

caption = "Data source: "

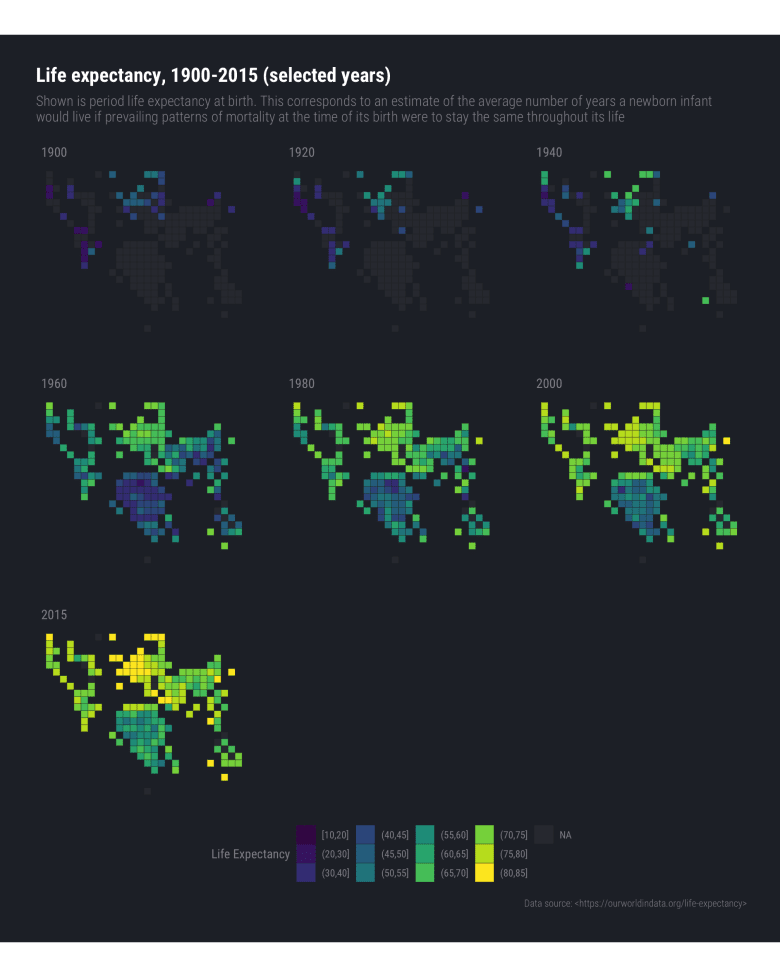
) +

facet\_wrap(~year) +

theme\_ft\_rc(grid="") +

theme(axis.text=element\_blank()) +

theme(legend.position = "bottom")

[](https://rud.is/b/2018/08/27/simplifying-world-tile-grid-creation-with-geom_wtg/le-wtg-02/)

(You may want to open that one up in a separate tab/window.)

Animation might have been a better choice than facets (which is an exercise left to the reader ).

**FIN**

The package API is still in "rapidly changing" mode, but feel free to kick the tyres and file issues or PRs on your community coding platform of choice as needed.

As noted in the package, the world tile grid concept credit goes to Jon Schwabish and the CSV data used to create the package exposed wtg data frame is the work of Maarten Lambrechts.