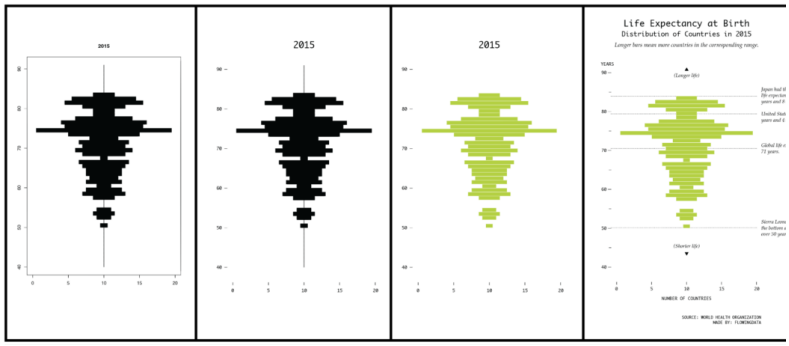
How to Edit R Charts in Adobe Illustrator

By **Nathan Yau**

A detailed guide for R users who want to polish their charts in the popular graphic design app for readability and aesthetics.

Download Source



When it comes to to visualization in R, most people stay completely in R. This is fine when your charts are specifically for analysis, because you're the only who looks at them. You don't need to add context, explain visual encodings, or make things look nice. The goal is to produce graphs in rapid fashion so that you can make sense of your data.

However, when it comes time to produce graphics for a wider audience, it can be useful and more efficient to export your chart as a PDF and edit in vector software such as Adobe Illustrator or its open source alternative Inkscape. We covered the latter already. Inkscape is free, but the usability and setup process isn't as straightforward as Illustrator.

On the other hand, Illustrator costs you around twenty bucks per month, depending on your subscription plan, and it can be overkill for some who just want to edit a handful of charts every now and then. So naturally, each has it advantages and disadvantages.

I personally use Illustrator. In this tutorial, I walk you through my general workflow, starting from a default chart made in R and to publication-ready graphic.

Setup

This tutorial uses base R and Adobe Illustrator CC 2015. If you can get your hands on an old version of Illustrator for a flat fee, that should work too. The interface will just look a little different. I switched to CC 2015 about a year ago, because I noticed my older version wasn't running great on a newer computer. In any case, as this is an Illustrator tutorial, you need some running version of it.

You only use R to generate a default plot so that you can edit it. If you want to skip the R step and go right into Illustrator, I included a default plot to edit in the tutorial download.

Generate Chart in R

Start with the data. Load life expectancy data from the World Health Organization using `read.csv()`. It provides estimates from 2000 through 2015 for over 200 countries.

```
# Life expectancy estimates from the World Health Organization
life <- read.csv("data/life-expectancy-who.tsv", sep="\t",
stringsAsFactors=FALSE)
```

This will be more clear once you see the chart.Then draw a somewhat non-traditional chart, which uses the same binning as a histogram but rotates the chart by 90 degrees and aligns the bars horizontally in the center.

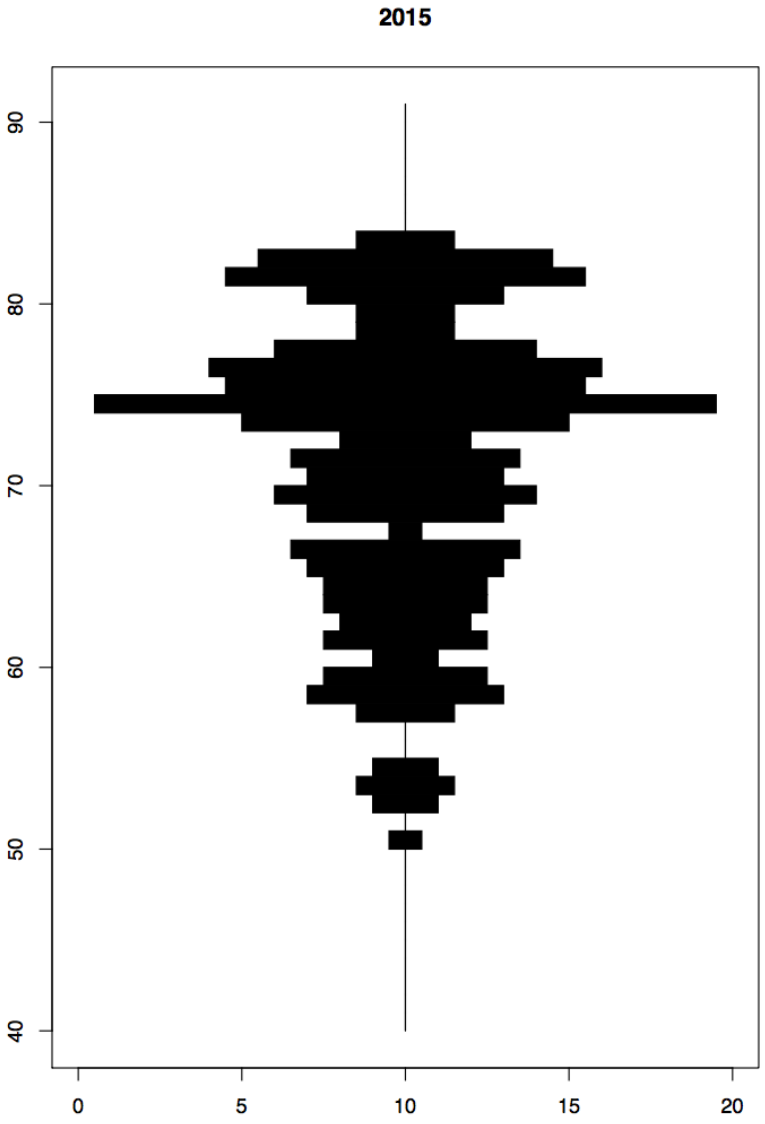
Calculate the coordinates for each rectangle:

```
# Calculate coordinates.
year <- 2015
curr <- life[life$Year == year, ]
cnts <- table( cut(curr$life_exp_birth_all, breaks=40:90) )
max_cnt <- max(cnts)
xleft <- (20 - cnts) / 2
xright <- xleft   cnts
ybottom <- 40:90
ytop <- 41:91
```

Then draw the plot:

```
# Draw plot.
plot(0, 0, type="n", xlim=c(0, 20), ylim=c(40,91), xlab="", ylab="",
main=year)
rect(xleft, ybottom, xright, ytop, col="black")
```

Here's what it looks like:



I give you this chart type so that you get a chance to annotate it and explain how to read it. You get into this later but first, let's take care of some boilerplate stuff.

Export the chart as a PDF. Depending on how you use R — R base, RStudio, or command line — the process is a little bit different. You can either save as PDF via menus or use the `pdf()` function to save output in a PDF file. You can also just use the PDF file titled "default-plot-from-R.pdf" in the tutorial download.

Assuming you have your PDF file, open it in Illustrator.

Quick Illustrator Rundown

One of the toughest challenges with getting started with Illustrator is simply knowing what tools to use for each task. There are a ton of options and menus, and it can be a confusing endeavor to find a tool the you need that you don't even know the name of. So here's a quick rundown of what you use in this tutorial, which is a small subset of all of the things available.

There is the Default Tools panel, which provides quick access to the commonly used tools:

The black arrow is the icon for the Selection Tool. As the name suggests, it lets you select elements. The white arrow is for the Direct Selection Tool. It provides a similar function, except when you select an element that is “grouped” with other elements, it will only highlight the single element. In contrast, if you were to select an element that is grouped using the Selection Tool (black arrow), the whole group is selected.

In case you’re confused, this grouping stuff (and other concepts) should be more clear when you put it into practice in the next sections.Why is this useful? Grouping lets you keep elements together so that you can edit them almost as if they were one element. For example, say you have a chart. The set of rectangles in a bar chart usually share the same aesthetic and coordinate system, so it makes sense to group them.

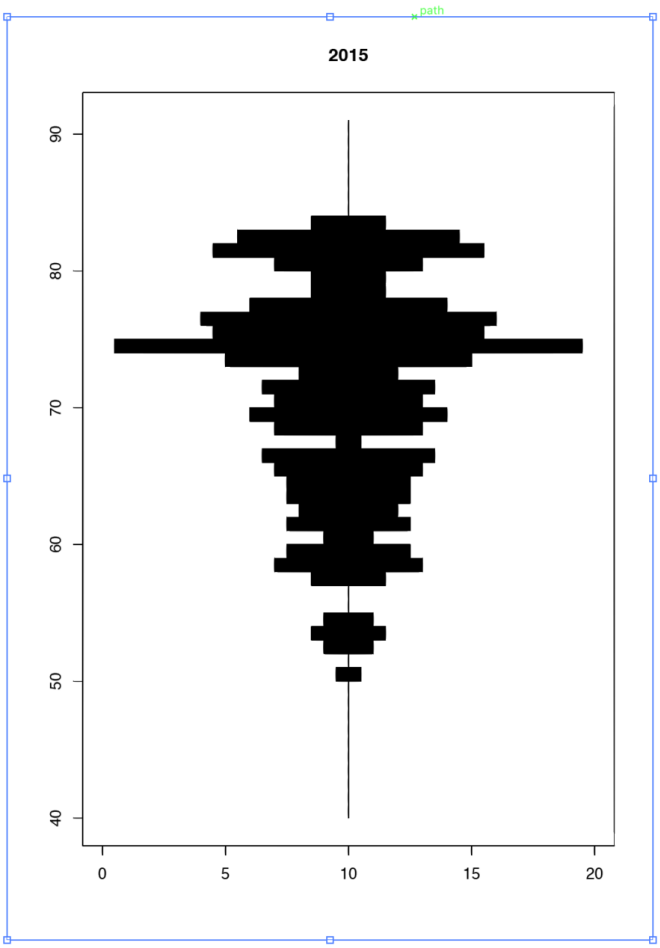
Moving down the Tools panel, there is a “T” which is for the Type Tool. You use this one to add text to your graphics, or in the case of visualization, annotation. You also use it edit existing text.

That’s pretty much all you need for now. I’ll explain anything else that comes up along the way.

Remove Extraneous Elements

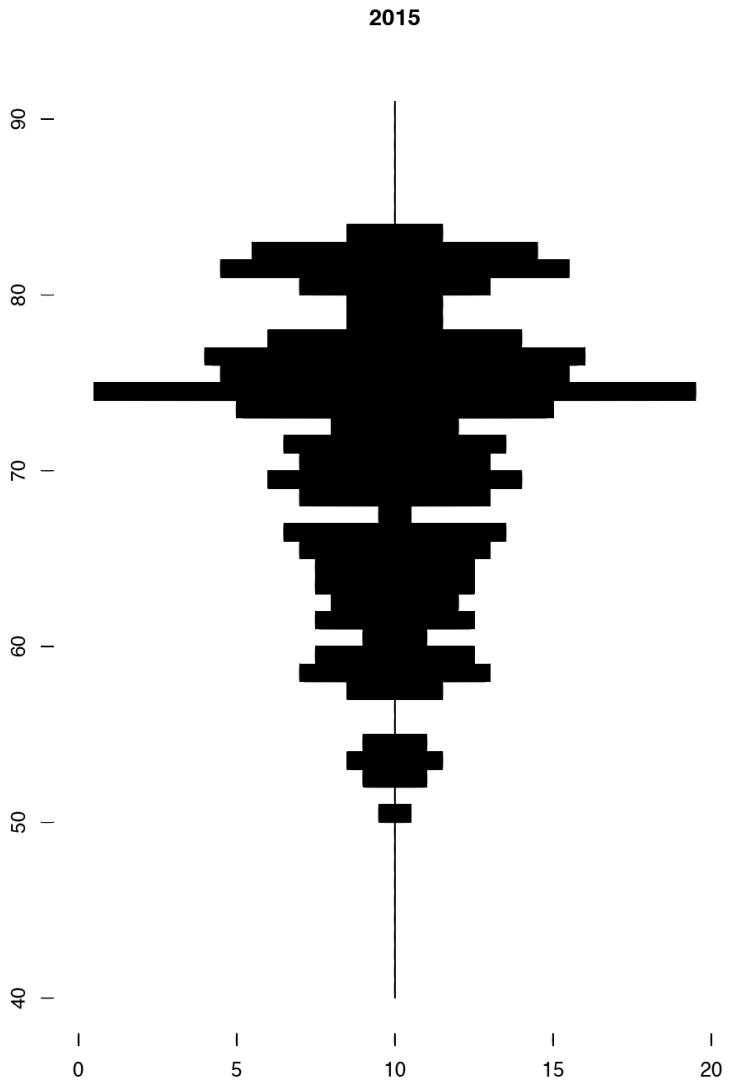
When you export a graph from R as a PDF file, there are a handful of elements that I always delete to make editing in later steps easier: unnecessary clipping masks, the axis baselines, and the box border around a chart.

Using the Direct Selection tool, click and drag around the outside border of the graphic. You can’t actually see the clipping mask until you select it. Your selection will highlight. Hit the delete button on your keyboard to remove.



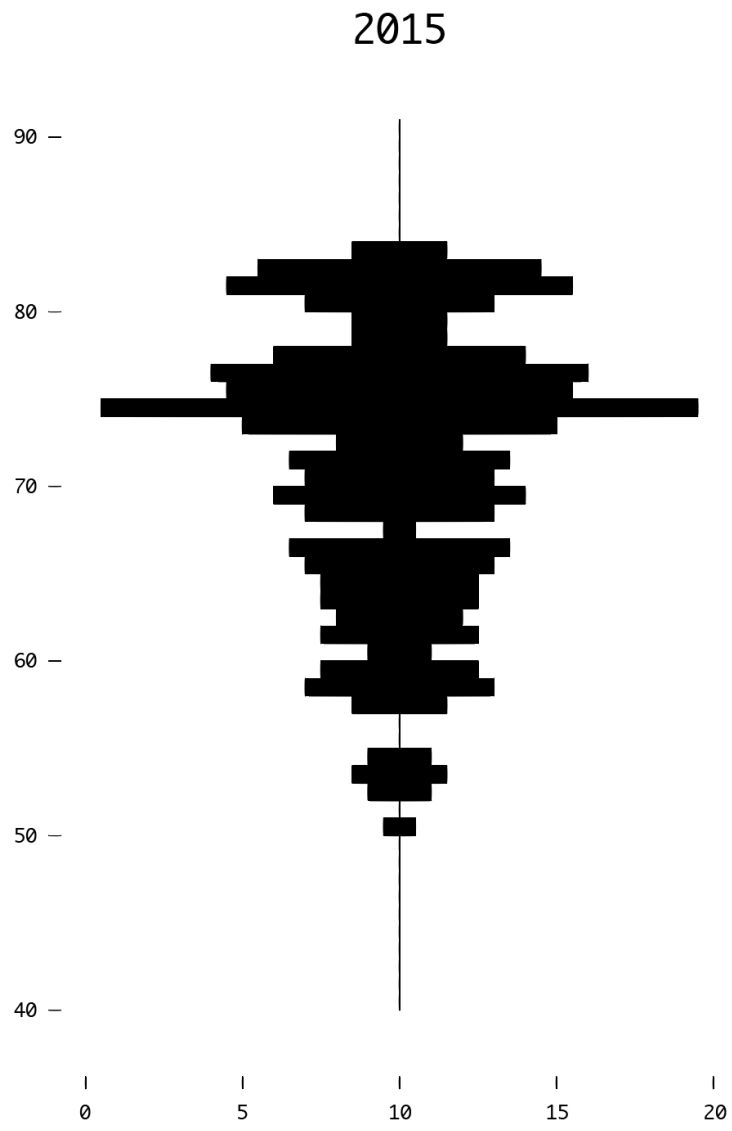
Remove the box border around the chart in the same way. Then click on the baseline for the horizontal axis (still using the Direction Selection tool) and hit delete. Do the same for the Vertical axis baseline.

Here's what you should have so far:



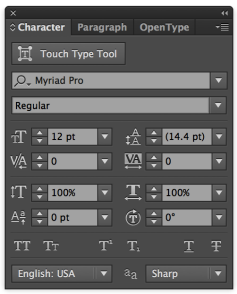
Fix Labels

Now adjust the labels to make them more readable and to fit within your preferred style. A simple change in font can make a big difference. Here's what we're after:



Again, using the Direct Selection tool, click and drag over the vertical axis labels. From here, there are multiple ways to change the font: You can right click on the text and select font from the context menu, you can change character style via the Character window, or you can edit via the Control bar. It depends on what you're comfortable with. I never use the right-click method (for no reason in particular), and go with the Character window or the Control bar instead.

Here's what the Character window looks like:



You can enable it via Illustrator's main menu, Window > Type > Character. Change typeface, size, spacing, and orientation from here.

The Control bar, found at the top of the screen by default changes depending on what you select. Here's what it looks like when you select text:

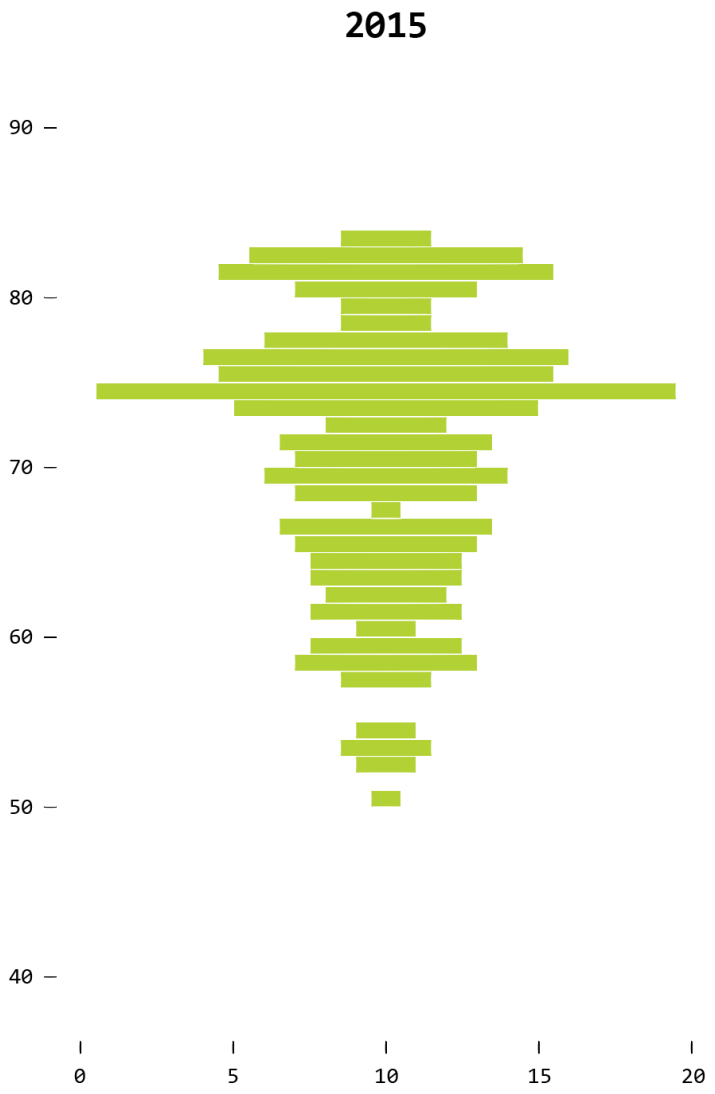
It offers similar options as the Character window. Try to change the typeface and font style for the labels and title. On my own computer, I changed the typeface to Consolas, but it may or may not be available on your own system. In this case, choose whatever typeface you like.

When you use Transform Each… instead of just Transform, each element is transformed individually. If instead you did a plain transform, it would rotate the full selection by -90 degrees.To rotate the labels on the vertical axis so that they read horizontally, select them again with the Direct Selection tool. Then via the main menu Transform > Transform Each, set the angle of rotation to -90 degrees. Click OK.

Adjust Visual Elements

It already looks pretty different with these small changes. Next, you adjust the visual elements — the things that actually encode the data. The main thing to keep in mind: Don't mess up the geometry, because if you screw that up, you misrepresent the data.

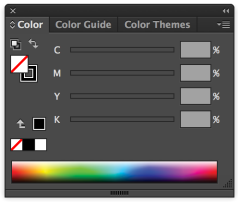
Here is what you will do:



It's a new color for the bars and a white border around each to provide visual separation.

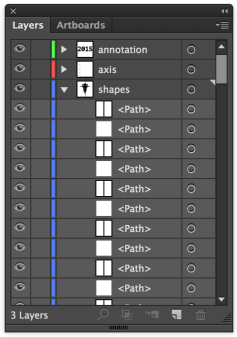
In the exported PDF from R, each rectangle is actually a set of two rectangles. One is a filled rectangle, and the other is an empty rectangle that serves as a border. In this example, no border color (stroke) was explicitly specified in R, so the border rectangles' stroke color match the fill color. However, in Illustrator, you can specify both the fill and the stroke color on a single rectangle, so the border rectangle from R is extraneous.

Again, use the Direct Selection tool. Mouse over the border of a rectangle until you see it highlight. Click. Look at the Color window to confirm that you selected the border rectangle and not the fill one. It should look like this:



Use the menu Select > Same > Fill & Stroke to select elements with no fill and a black stroke. This selects all the empty black rectangles. However, it also selects the tick marks, and you don't want to delete those.

This is where layers comes in. De-select the rectangles for now by clicking elsewhere on the page. Create a new layer called axis via the Layers window:

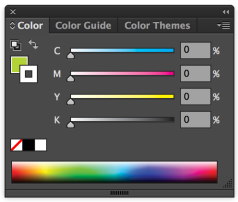


In the bottom right, there is an icon to create a new layer. It looks like a square page with a folded corner. It's next to the trash can icon. Click on it and create a new layer called "axis". Select your new layer in the Layers window.

Then move your mouse back to the graphic and select both the horizontal and vertical tick marks with the Direct Selection tool. Right click and in the context menu, select Arrange > Send to Current Layer. This moves all of those elements to the axis layer (assuming you selected it in the Layers window).

Now you can lock the axis layer so that they can't be selected. Click on the empty square next to layer window (next to the eyeball). Then try doing the same thing again with Select > Same > Fill & Stroke. This time around, only the rectangles highlight. Delete these empty-fill rectangles.

This might seem like a lot of steps right now, but once you know where all of this stuff is, the process only takes a few seconds.This leaves you with filled rectangles with no border. Select them with click-and-drag, and change their fill color via the Colors window. Change their stroke color too to white. Your color selections should look similar to this:



Annotate

Annotation plays a big role in making a graphic readable to a wide audience. Some argue that annotation is the most important layer, as it points out what's important, why a graph exists, explains how to read a graph, and clarifies ambiguities.

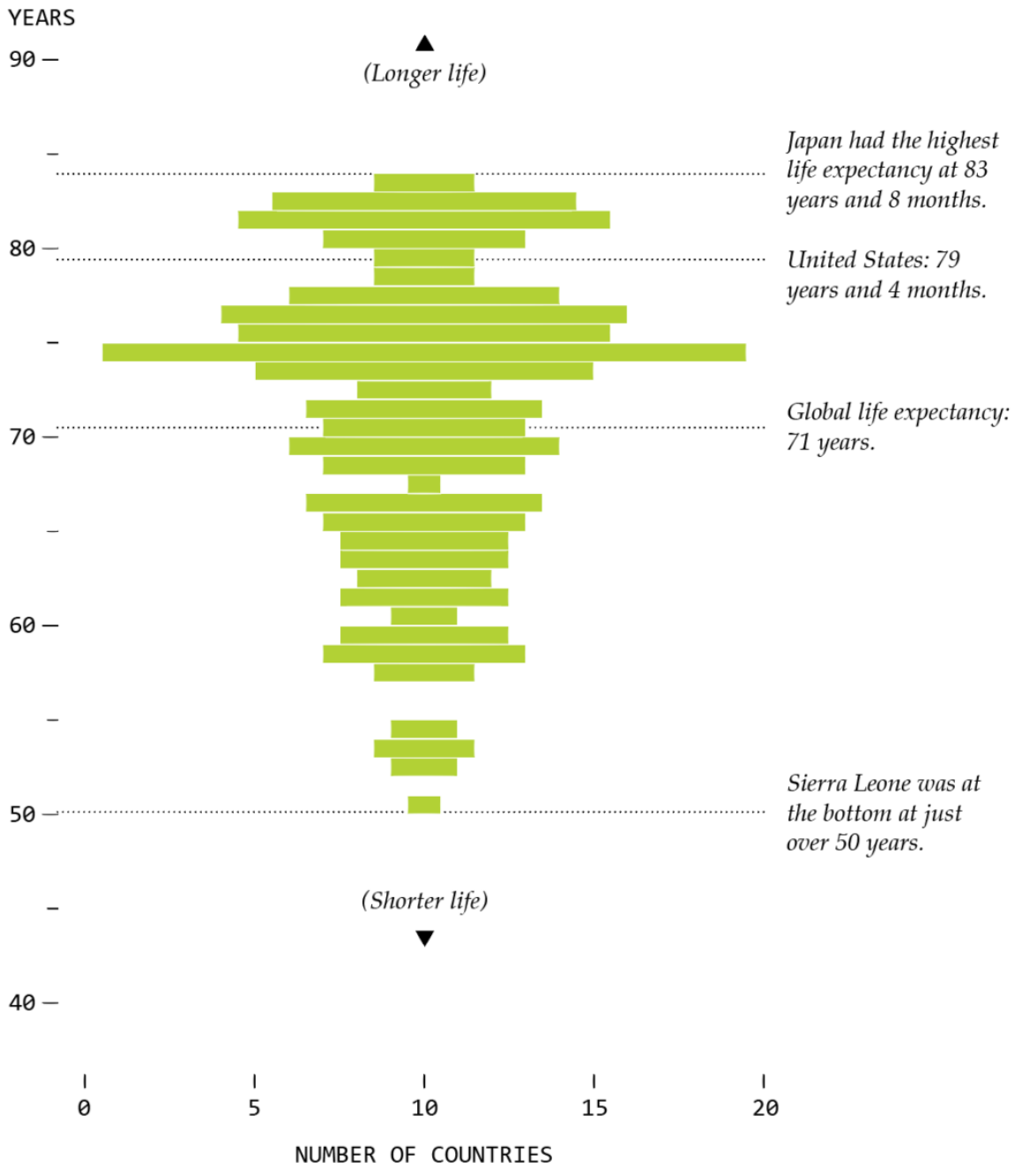
It's fairly straightforward to add text in R, but you have to mess around with coordinates and layouts to make sure it looks right. Text is finicky though, so I find it's much easier to edit in Illustrator than trial-and-error with R. You can see changes instantly

Here's the previous chart annotated:

Life Expectancy at Birth

Distribution of Countries in 2015

Longer bars mean more countries in the corresponding range.

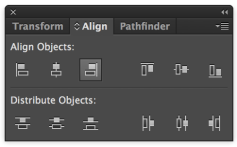


SOURCE: WORLD HEALTH ORGANIZATION
MADE BY: SOMEONE

Again, you use the Type tool to add and remove text. If you just click with the Type tool, a cursor shows up so that you can type. All of your words appear on a single line by default, but you can just hit Enter on the keyboard to create a new line. Alternatively, you can click and drag with the Type tool, which creates an area where your words go. The advantage of this route is that word-wrapping and word breaks happen automatically.

By the way, it can be useful to create a new layer for the annotation to keep things organized.Once you create text on top of your chart, you can change fonts in the same way you did before through the Character window. I find it's usually easier to enter all of your labels and notes first and then adjust later. The content is most important part after all.

That said, with all of your notes on the screen, adjust their position and size so that the labeling is less hodge dodge and more order. Select text with the Direct Selection tool, and then use alignment options through the Align panel, as shown below:



These let you align text left, right, center, top, bottom, and middle. The key is organization so that your annotation is easy to follow. You also want to create a visual hierarchy, creating visual differences between headers, sub-headers, labels, and notes. If something is more important, you can make the type bigger or bolder. If it's less important, you can make it smaller or faded. The style you choose evolves with use, and you get a feel for what you want with practice.

Wrapping Up

For me, this seemed like a lot to take in at first. So many icons. So many menus. But like R, it gets easier with practice. And you're really using only a small subset of what Illustrator has to offer: selection, color, alignment, and text. Figure out how those work, and it's much easier to edit your charts, because you can directly move all of the elements.

About the Author

Nathan Yau is a statistician who works primarily with visualization. He earned his PhD in statistics from UCLA, is the author of two best-selling books — *Data Points* and *Visualize This* — and runs FlowingData. Introvert. Likes food. Likes beer.

7 Comments (Add Yours)

- *Mara* — February 8, 2017 at 3:19 am

  After wrestling with some text placement in R for far too long, I decided to add Illustrator into my workflow for a specific project. However, I'm of two minds about it, bc I plan on sharing the code, and it's not something I scripted. Any advice for how to use illustrator in the context of reproducible research? Thanks, Nathan, I get so much out of your tutorials.

  Reply

  - *Nathan Yau* — February 13, 2017 at 2:42 pm

    Tough one. I always use Illustrator as a way to edit and annotate. So the actual mapping visual elements to data is done in R. Does polish need to be reproduced?

    Reply

    - *Mara* — March 19, 2017 at 4:56 pm

      I guess not— I'm always curious how far people go with that. Like, I know this code -> viz -> code project exists: https://github.com/alex-r-bigelow/hanpuku

    - *Nathan Yau* — March 22, 2017 at 10:08 am

      Interesting. I know there's a programmatic way into Illustrator with JavaScript, so that might be one way to go about reproducibility. I haven't used it enough yet though to know for sure.

- *Minh Bui* — March 2, 2017 at 4:51 pm

  Thanks Nathan for the great tutorial, I found one typo in the R script:

  plot(0, 0, type="n", xlim=c(0, 20), ylim=c(40,91), xlab="", ylab="", main=y)

  I think you mean: main = year

  Reply

  - *Nathan Yau* — March 22, 2017 at 10:05 am

    Thanks, Minh. Fixed.

    Reply

- *Nathaniel Black* — March 2, 2019 at 11:19 am

  "+" is missing in the line below:

  xright <- xleft cnts

  Reply

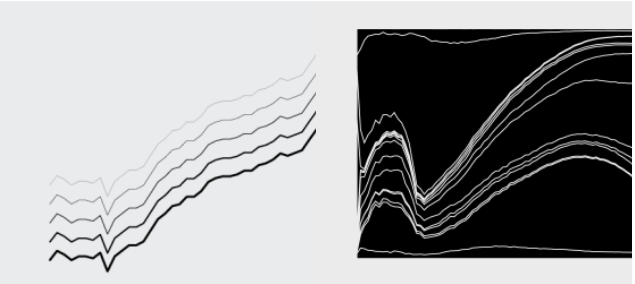
Add Comment

Logged in as webbedfeet. Logout »

Comment

Submit Query

More Tutorials See All →

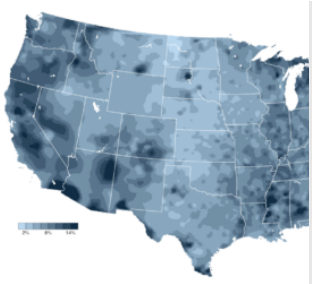


How to Make Line Charts in R

Learn to draw lines wherever and however you want, and you've got yourself some flexibility.

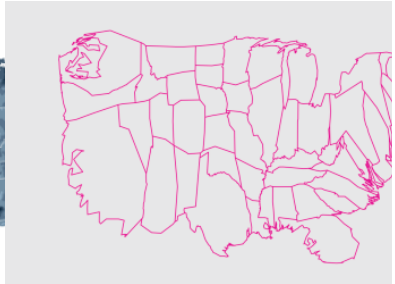
How to Make an Interactive Stacked Area Chart

Stacked area charts let you see categorical data over time. Interaction allows you to focus on specific categories without losing sight of the big picture.



Transitioning Map, Part 2: Refining the Format and Layout

How to make a more readable and more visually accurate map, before you dive into the big transitions.



Mapping with Diffusion-based Cartograms

Sometimes these cartograms can distort areas beyond recognition, but they can also provide a better visual representation for a region with a wide range of subregions. At the least, they're fun to look at.