ॐ Sketchbook

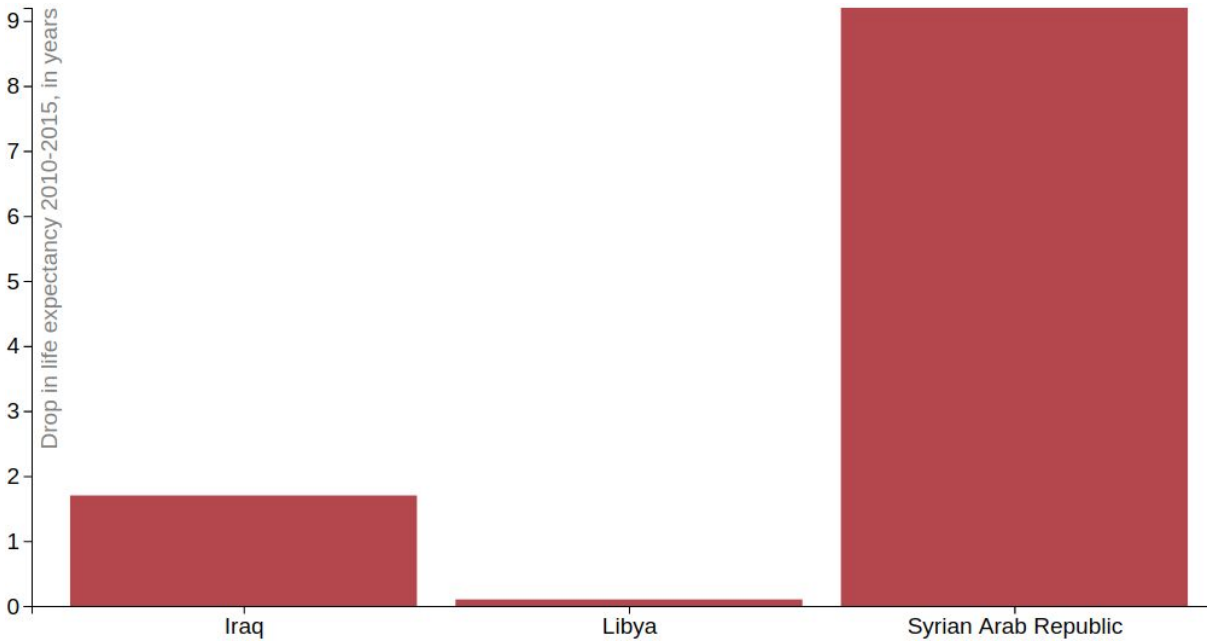
Information Vis. course, FHNW 2017

by Alexandr Shchelov

(https://github.com/ChadoNihi, @ChadoNihi, https://codepen.io/electroFish/)

Exercise 01 (a Linux guy ∴ no Tableau)

Which countries have decreased life expectancy in 2015 compared to 2010? Which had the biggest drop?



The chart was generated with JS library D3.js. The code can be found here.

As seen above, out of almost 200 provided countries in the WHO's dataset, only three -- Iraq, Libya, and the Syrian Arab Republic (aka Syria) -- had decreased life expectancy in 2015 compared to 2010. This may be partly explained by the fact that both of the Middle Eastern countries as well as African Libya are in a state of civil (religious?) wars.

Exercise 02

The raw CSV-file was compared to the goal file using online tool on https://www.diffchecker.com/.

In order to 'clean' the raw data a simple Python CLI tool was written. The script reads a headers row of the goal file, add it to the clean data, then steps through the raw CSV data, building the output.

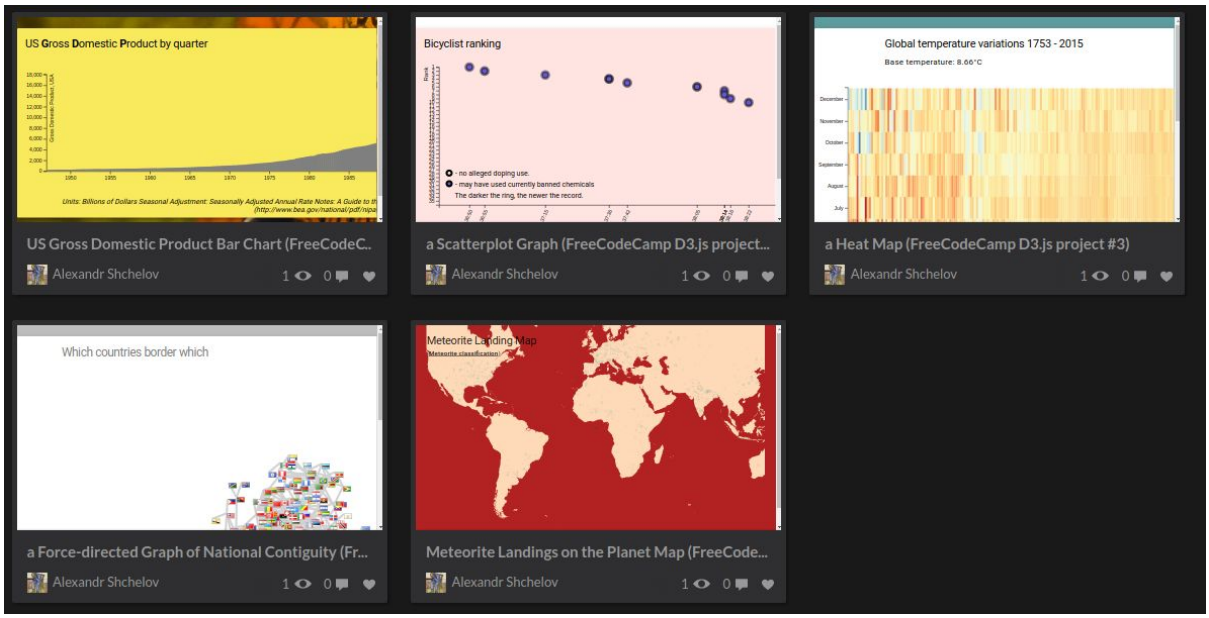
The first working version of the script revealed entries without IDs and some repeated names. The latter version deletes those rows, to be consistent with the goal file.

Why haven't I used Trifacta Wrangler and Apache Zeppelin, which the mentor suggested that we use? The former I didn't plan to use from the beginning due to its commercial distribution, DSL, and lack of Linux support.

Apache Zeppelin (as well its alternative Jupyter) was forgone as for the current exercise I only needed simple data transformation, with no visualisation. Later I could adapt my Python script to work with either of those 'notebooks'.

Exercise 03

As a part of Free Code Camp's data visualisation certificate ('Chado Nihi' is my web name), I have done several small projects with D3 v4. I've also used the library in lieu of Tableau for the first exercise.



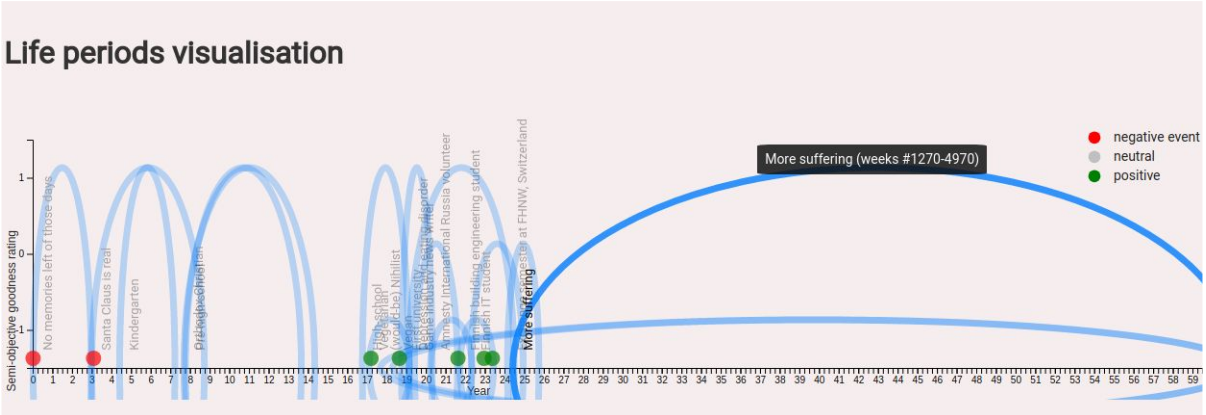
My D3.js collection on CodePen.

Exercise 04-05

From the time I had read the requirements, I wanted to forgo the referred visualisation and meet (most of) the requirements with a different visualisation.

A prefered idea I came up with involved visualising life periods with sets (ellipses), using color for an additional dimension. I wanted to utilize the fact that life periods, like sets, can overlap or be "subsetted" by longer life episodes.

However, given my D3 skill, I didn’t have enough time to properly implement that idea. Still, the idea of overlapping sets can still be seen in the [final prototype]:



Two CSV files were generated for the task:

- one describing life periods: period in weeks, description, and a “goodness” rating
- another describing life events (circle “points” on the visualisation): week number, description, and the rating

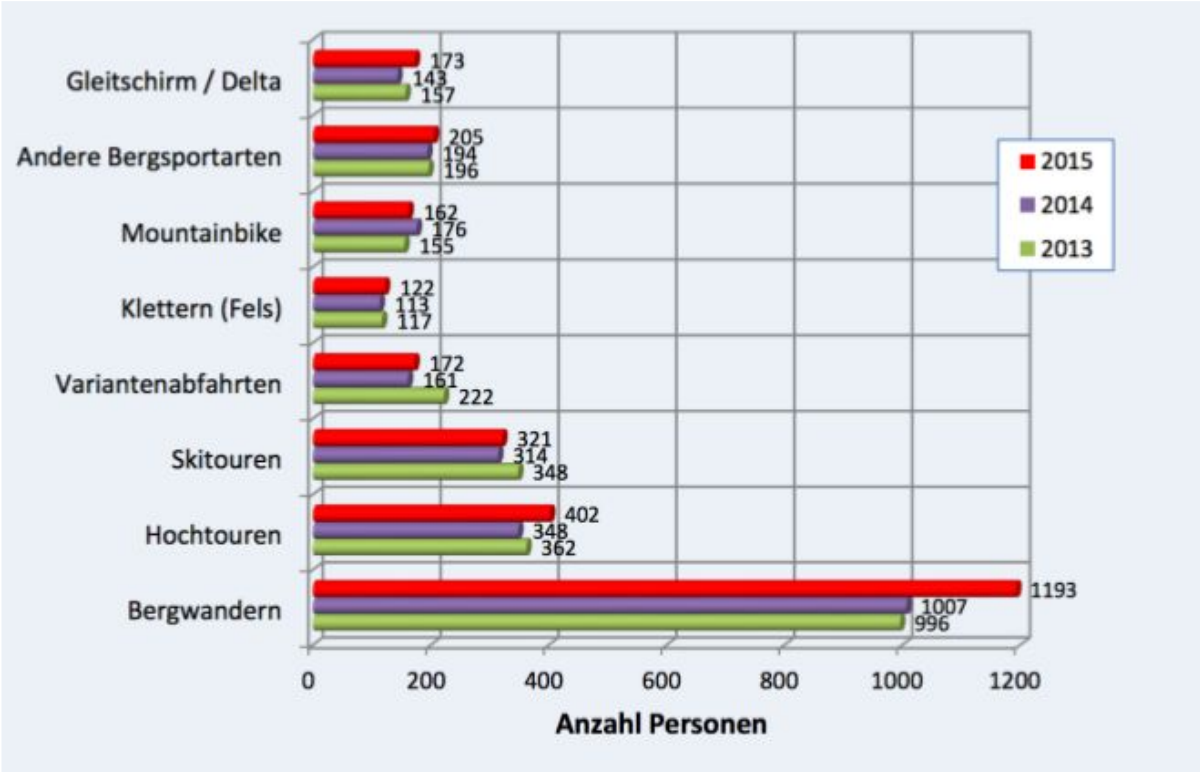
Since depicting weeks on one line was impractical, I used years instead, with a quarter scaling.

Many questions could be asked regarding quality of the visualisation: ”Why is it so ugly?”, “Could it be less ‘noisy’?”, “Why not use the events’ color convention for life periods?”, etc. But since I deem it to be a prototype, I would leave those questions open.

Exercise 06

See Exercise 01

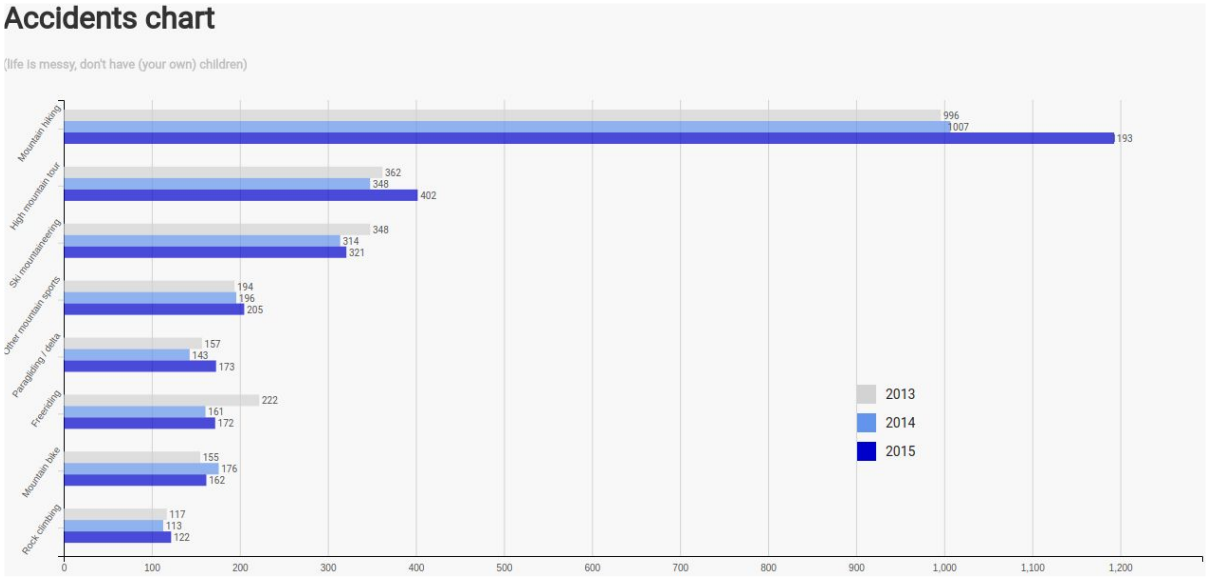
Exercise 07



Possible problems with the graph:

- Presence of both green and red colors (often a problem for 'color-blind' people)
- Label overlap
- Redundant x label
- Bold black (font, grid)
- Unsorted groups
- Unnecessary shadows
- Too saturated colors
- The graph's BG may not match the rest of the page

My refactored version of the graph is accessible on CodePen and looks like this:



The general strategy was to rid the graph of unnecessary visuals and make it “color-blind friendly”.

Bars, for example, became “flat” and shadowless. Colors were chosen of different brightness and without mixing colors that persons with certain kind of color-blindness would perceive as nearly the same (e.g. see blue-yellow color blindness).

Background color became that of the parent element, nearly white.

Since the top bar group seems to me too far from the x axis, I thought of either using subtle guiding vertical grid lines or putting the x axis additionally at the top. In the end the former seemed to me less obtrusive and more effective.

To construct the graph D3.js library was prefered to:

- more high level programming tools since I wanted more control for parameters (although I didn’t have time to fully utilise that power);
- graphical editors for I’m currently honing my programming skills to find relevant job.

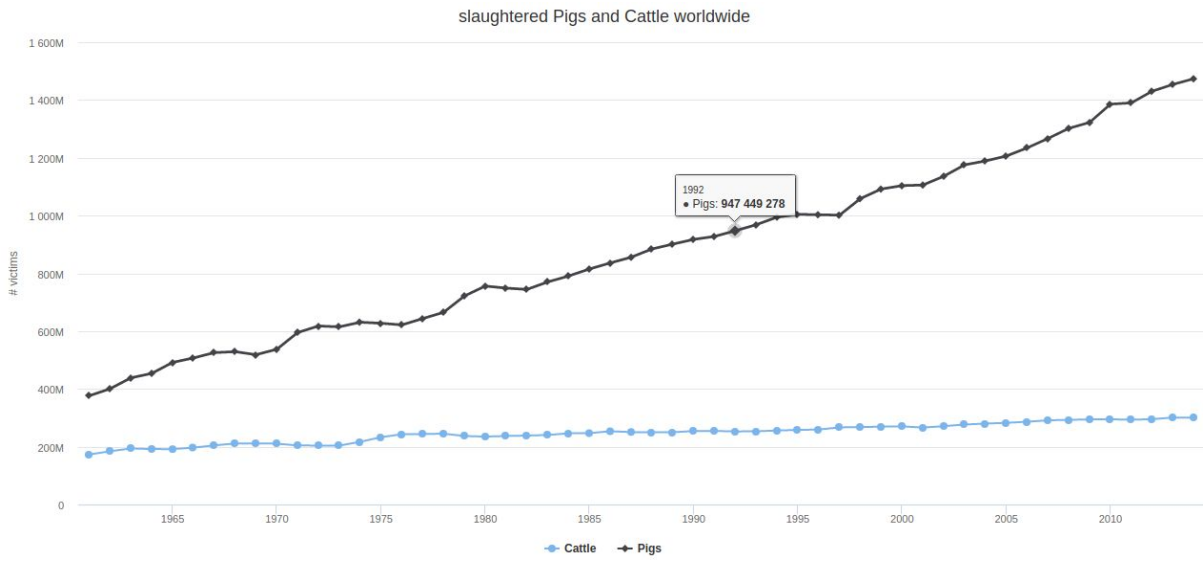
Exercise 08

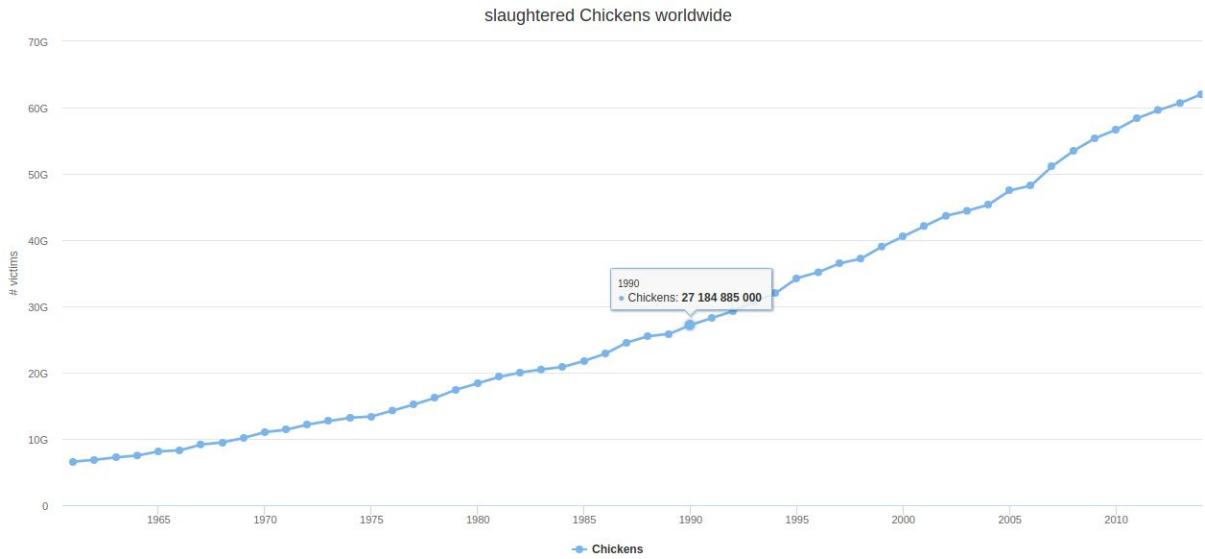
I have few doubts which FAO’s data to visualise: suffering of tens of billions of farmed animals is, I would argue, one of the most urgent problems of our times. In this regard, I’ve chosen to visualise the number of slaughtered non-human animals (although, in many cases slaughtering may be seen as a final deliverance from the misery), particularly pigs, ‘cattle’, and chickens (as prototypical factory farmed animals).

My initial plan was to use statistics for China, Germany, and USA. But then I thought that what matters is the total number of the abused animals, and the per-country statistics would draw attention from a worldwide situation. I thus ended up with a worldwide dataset.

For this visualisation I had decided to forgo D3 and use something unfamiliar. The choice fell on Highcharts, a free (open-source?) JS chart library. It seemed like a good addition to my toolset, especially for time-constrained tasks since Highcharts trades detailed customisation for producing nicely looking results *quickly*.

Since the number of slaughtered chickens outnumbers that of cows and pigs by a factor of thousand, I had a problem of visualising the data in one chart. An initial solution was to use a logarithmic scale for the number of victims. That way, however, the discouraging increase in the numbers was smoothed to a horizontal line. So in the end I constructed two line charts: one for chickens and one for pigs and cows:




slaughtered Chickens worldwide
70G
60G
50G
40G
30G
20G
10G
0
victims
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
1990
Chickens: 27 184 885 000
Chickens