Data Domain

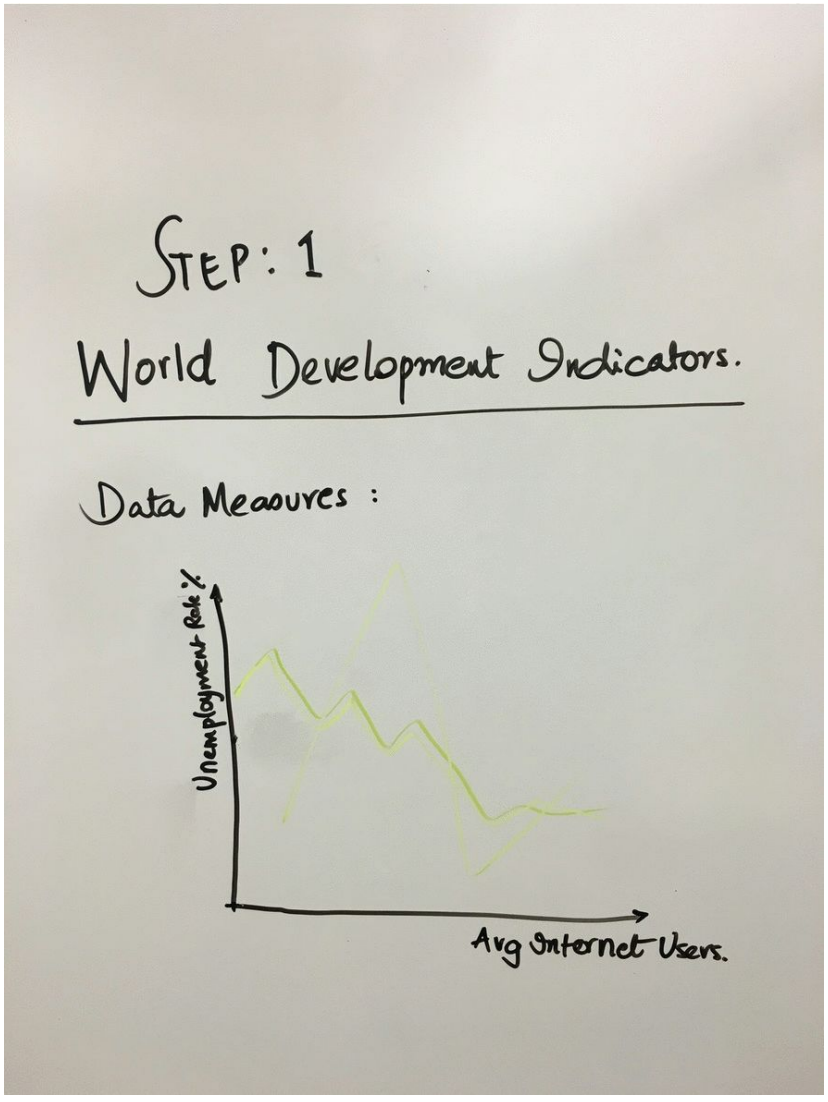
World Development Indicators: Exploring Unemployment Rates Vs. Average Internet Users (per 100 people)

Interactive Visualization Technique Implemented

Multidimensional Visual Analysis Using Cross-Filtered Views.

About Data Domain and Interactive Visualization Technique: Getting Started

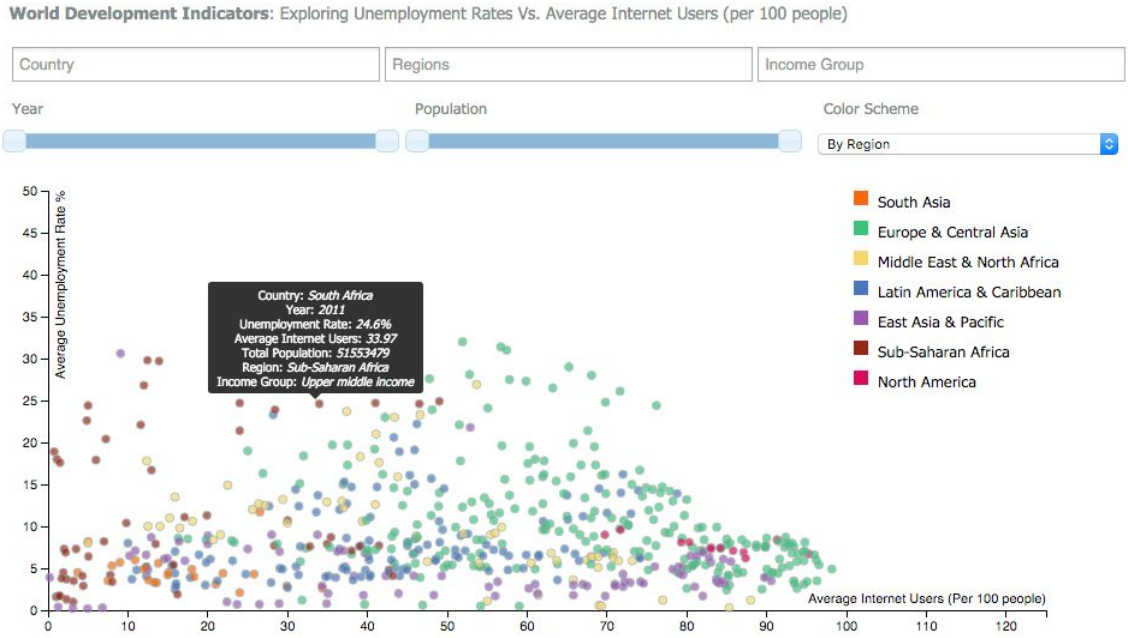
- The idea is interactively expressing sequences of multidimensional data by selecting and filtering unique data values across pairs of views (Weaver, C. 2008).
- The data domain in this given context is the exploration of a few world development indicators as presented by The World Bank. In particular we scope down the analysis to the following parameters: (i) Average Unemployment Rate %, (ii) Average Internet Users (Per 100 people).
- The different dimensions/views that we want to explore in this project are: Countries, Regions, Income Groups, Population across 5 years. Before getting started with the implementation of the code, we envisioned the software to solve this by allowing user to slice and dice across any of these dimensions and filter data across pairs of views.
- We started off with the whiteboard to brainstorm possible ideas to move forward with the implementation narrowing down the data measures, dimensions and the possible visual representation methods.

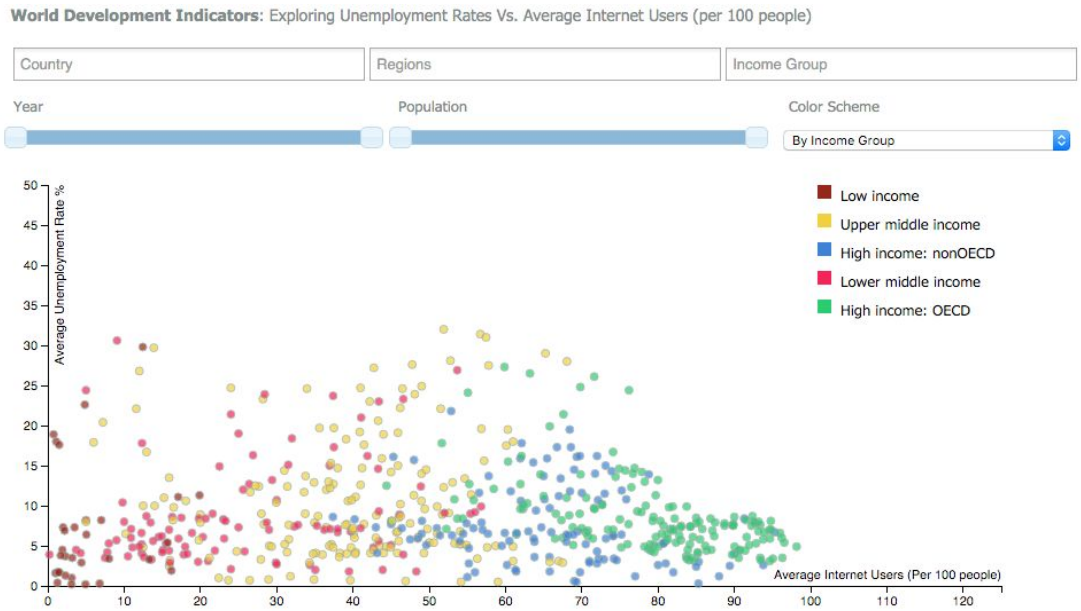


Storyboard - Images

Design Implementation and Feature Description

We made the obvious choice of selecting D3 for its flexible implementation options and customizability. Initial steps of getting used to the development environment was challenging, but we caught pace soon. The final interactive visualization application features the following filters: Country, Region, Income Group (Multiple select dropdown menus) and Year, Population (Sliders for brushing and linking). The data is represented as a scatter plot with the unemployment rate on the Y axis and the average number of internet users. The idea of this exercise is to not arrive at any conclusions or draw causal interpretations but to simply explore patterns in data across different slices and cross-filtered views (no spurious correlation). The filters work on an "AND" operator logic where all conditions selected by the user need to be satisfied for the data to populate based on the desired selection. The most unique feature of our visualization is the dynamic color coding scheme that can be selected by the user: we explored the market leader in data visualization (Tableau) and discovered that this feature is not available on its current platform (Tableau allows color coding -- but it cannot be changed dynamically for a given visualization on the fly by the user once the coding scheme is implemented while building the visualization). When the user selects the color coding scheme from the dropdown the legend is dynamically updated based on the selection.





We can spot several patterns making use of the cross-filtered views in the interactive visualization built: For instance, it is visibly clear that the countries with lower development indices as per the World Bank are the ones with least employment and mostly lie in the Sub Saharan Africa. Also, the high income countries have the most internet users and the lowest unemployment rates. It is also visible that the employment rates seemed to have improved over the period from 2010-14 reflecting a strong economic growth across the board.

Translating storyboard brainstorming into implementation (Changes and learnings)

1. The main concern that came up during the whiteboard exercise to work on the storyboard: Which color coding scheme would uncover the most valuable insights hidden in the data? We performed some exploratory analysis to conclude that it would be extremely helpful to let the user decide for himself/herself. This resulted in our decision to go ahead and implement a dynamic color coding method that would allow the user to select different encoding depending on the context of the filtered data. We also selected the colors and the scales carefully to ensure the visualization followed Bertin's levels of organization and Tufte's data-ink ratio principals.
2. Additionally, we also agreed to the fact that the user might want to make multiple selections in the data for each filter - We identified this requirement for multiple select filtering.
3. Allow user flexible selections of all possible combinations of cross-filtered views (The selections can be done in any order).

Collaboration and Synergy (Sharing Coding Responsibility)

Working in a team of 2 was an intense experience - both of us were relatively new to D3 and web programming. This gave us a chance to share our learnings as we proceeded. Initially we decided to work independently and come up with possible ideas and the interactive techniques that we could possibly execute - this allowed us to bring unique perspectives and ideas to the table and then merge them. There was no explicit division of the amount of work or tasks to be completed but as a team we ended up contributing equally. Kunal coded the data plotting and implemented the logic of the sliders to work together, Akash worked on the multiple selection filters and the dynamic color coding implementation. We worked as an agile team building many iterations and optimizing the code and the functionality as we grew. We also identified many bugs on the way and split responsibility to fix them. We collaborated and integrated our code to get all the elements working together specially the filters. Github served as the repository for the source code.

Development Process and Key learnings

We met almost daily in the MGH Collab to work together and share the updates on the progress made from the previous day. Ramping up initially was the biggest challenge and learning the basics of D3 took substantial time. The development process at times overwhelmed the both of us when we hit the wall - every time we ran into a road block we stepped back and tried to understand the cause of the issue. The development time was around 30-35 hours (per person) including many more hours just trying to fix code breaks and testing the filters under all use cases. The most challenging feature to code was integrating multiple filters and making sure they worked in sync as envisioned during the storyboard brainstorming process. One bug that involved the code breaking at the income filter took more than 1 day to fix: it was just a minus sign we missed in one piece of the logic. We learned the importance of writing clean code as the scale of the project grew to be able to maintain it. The other feature that drained a lot of time and effort was the dynamic color coding mechanism: the challenge was to ensure the filtered data was retained on the graph on color coding selection. With no resources available online about dynamic color coding implementation so we had to push the limits here. Our code was extremely modular and written in the form of reusable functions: this simplified a lot of things and improved code readability. We also arranged the files in the bundled manner to make it easy for any person to locate respective files in the CSS, Scripts and the data folders.

References

[1] Weaver, C. (2008, October). Multidimensional visual analysis using cross-filtered views. In Visual Analytics Science and Technology, 2008. VAST'08. IEEE Symposium on (pp. 163-170). IEEE.

Appendix

1. Project Live Link: http://akashjaswal.github.io/world-unemployment
2. Github Repository: https://github.com/akashjaswal/world-unemployment
3. Storyboard: https://drive.google.com/a/uw.edu/folderview?id=0B1kiZ_Y09c16M1B6RVFVSktxd2c&usp=sharing