

Objectives

With these assignments you will learn how to

- add basic interaction techniques to visualizations
- create basic layouts using D3.js
- evaluate the effectiveness and quality of a given visualization

For your programming tasks, you will use D3.js together with JavaScript integrated in a HTML file. First install node1 and a simple web server by typing

- `npm install http-server -g`

in your command line. In order to load CSV files you need to start a local server by typing

- `http-server -a 127.0.0.1 -o`

in the directory where your files are located. In each subtask you need to modify the provided HTML file, e.g., modify `Ex2c.html` for Exercise 2 c), images below a description demonstrate possible solutions. The sizes of the DOM objects, e.g., the radii of circles or color schemes, should be chosen in an appropriate way but are not restricted to one single possibility. Color schemes can be loaded from [colorbrewer¹](#), for details about D3.js you can refer to its [documentation²](#). Your solutions must be uploaded via Moodle till **December 09, 2020, 9am (UTC+1)** as one ZIP file that contains all HTML files. The naming convention for this ZIP file is `sheet2__<Matrikelnummer1>__<Matrikelnummer2>.zip`.

Exercise 2.1: Interactive Scatter Plot (10 Points)

On the first exercise sheet the data set *UnRegionsGdp.csv* was visualized using a scatter plot. In this exercise we will extend a baseline implementation provided by the file *ScatterUnRegionsGdp.html* by various interaction techniques.

- A third visual variable can be added in a two-dimensional scatter plot by mapping data onto the radii of the circles. Apply this idea to capture the size of population in the visualization. (2 Points)
- Interaction enables the user to capture more information from a single visualization. Extend the implementation from part a) to enable details-on-demand for single data points, which show the GDP and the HDI. (2 Points)
- Further extend the results from b) to filter all points, which belong to the same region provided by the legend. When a region is selected, all data points should be highlighted in an appropriate way. (2 Points)
- Extend the version of a) by adding zoom and brush functionalities. (4 Points)

Exercise 2.2: Two Layouts for ETF Data (4 Points)

The data set *Turnover.csv* contains the turnover in million Euro of 10 large Exchange Traded Funds (ETF) traded at Xetra³ for the months July, August, September and October. The goal of this exercise is to apply two layouts to visualize the distribution.

- Create a pie chart to capture the division of the traded ETFs for October. Make sure that the visualization applies an appropriate color scheme, shows labels, and numbers in an effective way. (4 Points)

¹Colorbrewer2

²D3.js Documentation

³Deutsche Börse Xetra

- b) A stacked bar chart can be used to capture the division at various points in time. Create a stacked bar chart for four months that display the division. Make sure that the visualization applies an appropriate color scheme, shows labels, and numbers in an effective way. (4 Points)

Exercise 2.3: Dissemination Data set (10 Points)

The data set *Covid19Worldwide.csv*⁴ contains the latest available public data on COVID-19 including a daily situation update, the epidemiological curve and the global geographical distribution (EU/EEA and the UK, worldwide). In this exercise your task is to create an expressive visualization that envisions interesting relations in the data. The minimal criteria are listed below

- At least one interaction technique
- A reasonable use of a color scheme
- At least two points in time

Your submission is evaluated based on its implementation and its theoretical foundation.

Exercise 2.4: Discussion of given visualizations (2 Points)

In the zip-file of this exercise sheet you can find four visualizations from different domains. Describe a negative aspect of each.

Instructions

Pair Programming On these assignments, you are encouraged (not required) to work with a partner provided you practice pair programming. Pair programming „is a practice in which two programmers work side-by-side at one computer, continuously collaborating on the same design, algorithm, code, or test.“ One partner is driving (designing and typing the code) while the other is navigating (reviewing the work, identifying bugs, and asking questions). The two partners switch roles every 30–40 minutes, and on demand, brainstorm.

Violation of Rules A violation of rules results in grading the affected assignments with 0 points.

- Writing code with a partner without following the pair programming instructions listed above (e.g., if one partner does not participate in the process) is a serious violation of the course collaboration policy.
- Plagiarism represents a serious violation of the course policy.

⁴EU Open Data Portal: COVID-19 Coronavirus data