

Objectives

With these assignments you will learn

- design and implement a visualization for weighted directed graphs;
- apply and implement Sankey diagrams using *D3.js*;
- visualize social networks using force-directed graph layouts.

For your programming tasks, you will use JavaScript and its library *D3.js* for creating visualizations. You are expected to comment and document your code in a focused and clear manner.

Your solutions must be uploaded via Moodle by **February 3, 2022, 9am (UTC+1)** as one ZIP file that contains all answers and source files. The naming convention for this ZIP file is **sheet5_<group_name>.zip**.

Instructions

Implementation As visualization framework we use *D3.js* and its respective language JavaScript; for details we refer to its documentation¹.

The implementations with *D3.js* should be implemented as a web page based using the framework. Make sure that relative paths are used. Supplementary exercise-specific data should be organized in a sub-folder called “data”. Color schemes, in particular, can be explored and loaded via colorbrewer2.org². Local provisioning for debugging and testing can be done using a local web server such as the *http-server* by *node*³. The server can be installed with the following command:

- `npm install http-server -g`

In order to load e.g. CSV files you need to start a local server by

- `http-server -a 127.0.0.1 -o`

in your working directory containing both the sources and the data.

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.

¹D3.js documentation

²COLORBREWER 2.0

³Node.js®

Exercise 5.1: Evaluation of Visualization Examples (2 Points, Theory)

In the zip file of this exercise sheet you will find four screenshots of visualization examples from different applications and domains. Describe and explain a selected aspect that conceptually represents a weak point in the respective visualization example.

Your submission should contain a single PDF-file.

Exercise 5.2: Visualizing International Migration Flows (6 Points)

Data on migration flows are based on the number of those people who migrate to (immigration) or leave a country (emigration) within a certain period of time (e.g., within a year)⁴. The dataset *MigrationFlow.csv* contains the number of emigrants and immigrants between two countries in the year 2017. This relationship can be represented by a directed, weighted graph.

Create a visualization that shows migration between countries based on the underlying graph structure. Not all countries need to be included. However, ensure that the visualization covers the 30 countries with the most immigrants and allows for at least one meaningful interaction by the user. Take care of appropriate labeling and choosing an adequate color scheme.

Exercise 5.3: Visualizing Napoleon's March on Moscow (6 Points)

A very well-known visualization is the illustration of Napoleon's Russian campaign *Carte figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813* by the French civil engineer Charles Minard in 1869. In fact, it is a *Sankey Diagram*; the graphical representation of quantity flows. This visualization technique is used for directed acyclic graphs, by representing the width of the lines connecting two nodes as a quantity.

The visualization framework *D3.js* provides a layout generator for creating Sankey diagrams. The *d3-sankey* module is not part of the *D3*'s default bundle, but can be imported using:

```
<script src="https://unpkg.com/d3-sankey"></script>
```

Apply the *d3-sankey* module to visualize Napoleon's March on Moscow in an interactive Sankey diagram. The data describing the respective quantities of the French army can be found in the attached file *NapoleonMarch.csv*. Use an appropriate color scheme to indicate the direction of each edge, which is encoded in the *direction* column. Further add one interaction technique, that provides details on demand.

Exercise 5.4: Visualizing the Social Graph of "A Song of Ice and Fire" (6 Points)

Game of Thrones is a very popular fantasy series with 73 episodes, divided into 8 seasons, based on George R. R. Martin's "A Song of Ice and Fire" saga. Its social network is described in the data set *GoT.csv* as an undirected graph⁵. In detail it contains the columns

- *Source* and *Target* contain the names of the characters, i.e., the names as nodes of the graph;
- *weight* represents the number of interactions between two characters.

Create a force-directed graph using *D3.js*. Define an appropriate mapping for the weights of the edges and the order of each node using adequate visual variables. In addition, labels should be included in the visualization.

⁴UN Handbook on Measuring International Migration through Population Censuses, 2017

⁵<https://github.com/mathbeveridge/asoiat>