JBI100 Visualization

Anna Vilanova & Stef van den Elzen

Assignment 2

Exercise 1 –Programming Language

You have to choose as a group the platform in which you develop the visualization tool. Making visualization easily accessible is relevant, for example, to achieve impact. Web-based tools facilitate this accessibility (e.g., JavaScript, D3, python, Dash). However, it is also more than fine if you build a tool, for example with JAVA, Python, JavaScript, C++, C#, or whatever your project group finds suitable. This course is not a programming course, however, programming is needed as a means to implement the visualization concepts. You have to adapt your choice to your programming expertise and level. Some choices like using C++ or JavaScript and D3 give you a lot of flexibility on the visualizations you can design, however, they have a steeper learning curve. This year we are giving a short introduction to some of these possibilities such that you can have some starting points.

We provide you with a framework using python and dash. https://gitlab.tue.nl/JBI100/dashframework

You can use this framework as a starting point, however, you are also welcome to use another programming language that suits your background and expectations better. The internet is full of basic information that can help you create an initial basic framework. This will be valued in the project but you should not be spending most of your time on it, since it is not the core of the course.

Different steps:

(a) Find a programming language that is suitable for each group member and discuss why.

(b) Discuss, what are the benefits of this programming language with respect to this visualization project? What are the drawbacks?

(c) Make a choice that suits your group concerning which programming language you will be using for the visualization project.

Exercise 2 – Tasks/ Questions (Why) (Domain specific)

Once we have the goal we need to divide it into tasks. Tasks the user will want to achieve with our visual design. Most tasks have to be adequate for a visualization solution, but they might be combined (especially at a lower level) with automatic/derived data tasks.

(a) (Task Analysis) Divide the goal into different tasks (at least 4 to 5) about the data set (around one sentence for each)! Examples of such tasks could be: "Understand the distribution of accidents across vehicle types". Try to find a balance between being vague or too specific. Make sure that some of the tasks are complex enough, and involve multiple attributes, with multiple meanings beyond two attributes.

(b) (Task Analysis) Another useful way to define tasks is by formulating questions. Define questions that the users might formulate to develop the tasks you have defined. Questions that your tool should provide an answer to, e.g., "How is the distribution of accidents per vehicle type?". Questions should be related and linked to the tasks.

Exercise 3 – Task Abstraction

Once the goal, data, and tasks are understood from the domain point of view. We enter into the abstraction phase such that we can identify the most adequate designs later on.

(a) (Task (Why)) Develop and write the task abstraction according to the "why" presented in Tamara Munzner's framework from the tasks that you have described in Exercise 2.