

Data Visualization Report

Thomas Vroom

i6291496, t.vroom@student.maastrichtuniversity.nl

I. CHOSEN DATASET

I chose the dataset "Nabijheid voorzieningen; afstand locatie, regionale cijfers" from the CBS as dataset for this project. This dataset contains information on average distances to common facilities such as hospitals, schools, train stations etc. To not make things overly complicated I only used the data from 2023 (the most recent year for which the data is complete), considered the data at municipality-scale (i.e. each municipality is its own spatial area), and used the following subset of columns:

Avg. Distance to Closest {GP, Pharmacy, Hospital, Supermarket, Primary School, High School, Highway, Train Station, Fire Station}

I used the "Wijk- en buurtkaart 2023" as the shape file for this data, also by the CBS. This shape file uses the same municipality codes as the dataset, making the transformation from the original dataset to geospatial data very straightforward. The dataset was preprocessed in the following way:

- 1) Select the relevant columns from the original dataset.
- 2) Remove rows that contain NaN (municipalities that do not exist anymore).
- 3) Merge the shape file and the dataset on the region codes.
- 4) Remove water areas from the shape file
- 5) Convert the coordinate system to EPSG:4326.
- 6) Export the merged shape file as GeoJSON.

The final dataset has 13 columns (features) and 342 rows (spatial areas; one per municipality).

II. MULTIVARIATE VISUALIZATION

The first task is as follows: "Compare the features of one selected spatial area to a set of other selected spatial areas". For the set of spatial areas I selected all municipalities from the province Limburg, with the goal of comparing them in a customizable subset of features.

A. Multivariate sketches

The first sketches for solving this task can be seen in Figure 1. If the image is not large enough, you can also find the original scan in the source code under `report/images/`.

B. Best 3 sketches

The first sketch considered can be seen in Figure 2. The **advantages** of this design include that it is the easiest to understand and use out of all my ideas, allows for a lot of customization, and allows for comparison of specific features for specific areas at a glance. The **disadvantages** of this

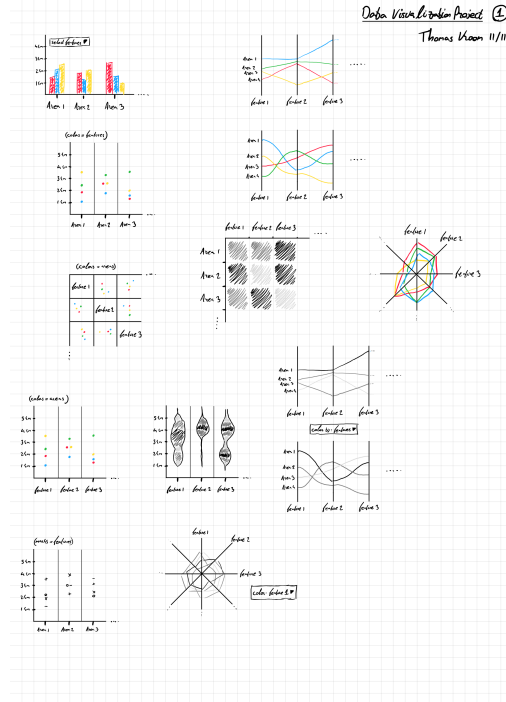


Fig. 1. First sketches for Task 1.

design include that it does not scale well with more areas and / or features, with more of either potentially leading to an overwhelming chart.

The second sketch considered can be seen in Figure 3. The **advantages** of this design include that it scales well with more areas and columns (up to a certain point), and it offers a lot of flexibility in what to visualize and how to do it. The **disadvantages** of this design include that you cannot find specific areas at a glance, and it is also difficult to compare specific areas.

The third sketch considered can be seen in Figure 4. The **advantages** of this design include that it adds new information (area under the line), scales well with more areas, and offers a lot of flexibility in what to visualize and how to do it. The **disadvantages** of this design include that it does not scale well with more columns, can feel very overwhelming, and is potentially more difficult to code.

In the end I opted for the second design, as it strikes a good balance with the flexibility of the first design, while not being as overwhelming to use as the third design. The disadvantage of it being difficult to compare specific areas should be mitigated in the future by allowing customizable



Fig. 2. The first in-depth sketch for Task 1. This design shows a grouped barchart with a shared y-axis, customizable selection of features through a drop-down menu, and highlightable bars.

area selection, and since parallel coordinate plots are quite common in multivariate visualization there should be enough resources online to realize the design in practice.

C. Implementation

The full source code can be found in the attached .zip file. The code for Task 1 can mostly be found in `src/multivariate_vis.js`. As described in there, I used existing code from the following sources: [1] [2] [3]. Unfortunately, there are still some things missing that will be implemented in the next week:

- Dragging to select multiple areas.
- Filtering between features.

D. Performing task 1

Screenshots of the visualization can be found in Figures 5 and 6. Figure 5 shows the parallel coordinates plot with 9 axes for the 9 different features. For ease of comparison, all axes share the same units and scale. The color-encoding of the lines can be customized by a drop-down menu in the top left. Hovering over a line highlights it and shows the name of the area in the form of a tooltip (see Figure 6).

E. Self Reflection

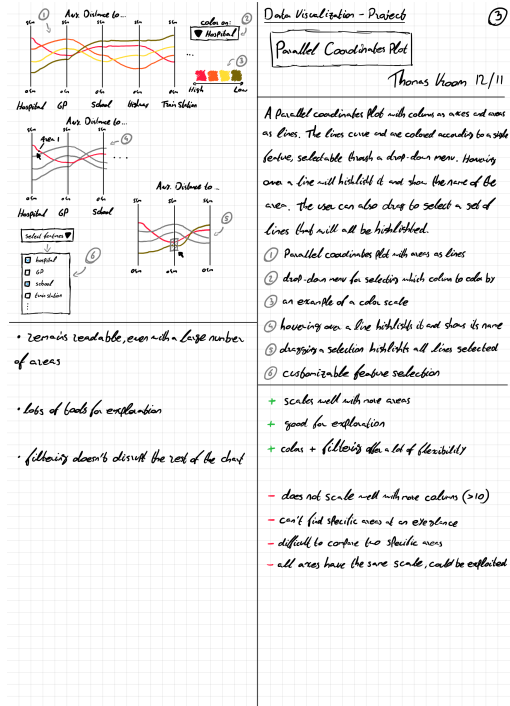


Fig. 3. The second in-depth sketch for Task 1. This design shows a parallel coordinates plot with curved lines, highlightable areas, customizable color-encoding through a drop-down menu, and customizable selection of features through a drop-down menu.

Points	Sub component
2 / 2	There are at least 10 different rough sketches of nD visualizations that could potentially solve Task 1.
2 / 2	There are at least 3 different rough sketches of nD visualizations not shown directly in the lecture that could potentially solve Task 1.
2 / 2	The three different detailed sketches can solve Task 1.
2 / 2	The detailed sketches could be implemented without confusion by a third-party.
2 / 2	The tradeoffs of the detailed sketches for Task 1 are discussed well.
1 / 1	The chosen visualization sketch can solve Task 1 well.
2 / 2	It is well argued why the chosen visualization sketch was chosen.
3 / 5	The final visualization sketch is fully implemented and suitable for Task 1.
4 / 4	Labels, colors and other design elements have been included in the implementation and are suitable.
0 / 5	Interactivity for filtering and selecting variables has been built into the visualization and is fully functional
2 / 2	Task 1 has been performed well with the implemented visualization.
1 / 1	You have self-reflected on this rubric, and filled in your expected amount of points for each sub component.

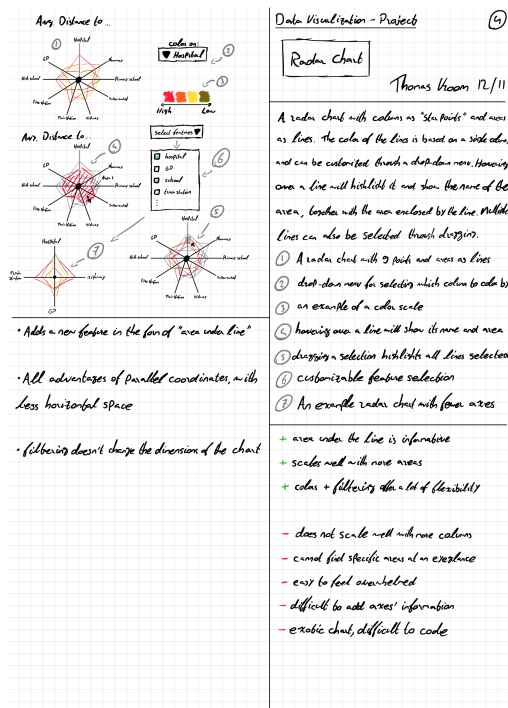


Fig. 4. The third in-depth sketch for Task 1. This design shows a radar chart with areas as lines, customizable selection of features through a drop-down menu, customizable color-encoding through a drop-down menu, and highlightable lines with colored-in areas.

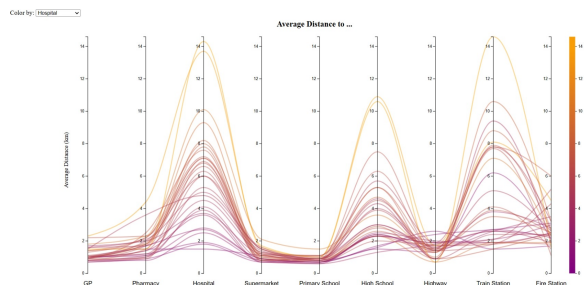


Fig. 5. A screenshot of the implemented visualization for Task 1.

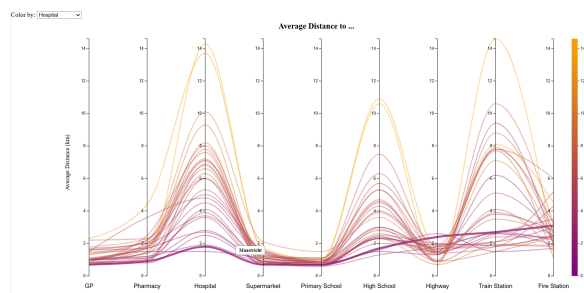


Fig. 6. A screenshot of the implemented visualization for Task 1, highlighting the entry for *Maastricht*.