CSC544 - Advanced Information Visualization
Assignment 4 - Parallel Coordinates

Problem 1:

Initial Concept:
Following decisions were taken initially:

1. Parallel coordinates are much better for displaying quantitative data than categorical data. Therefore, we do not display any of the categorical attributes as axes in the visualization.
2. We chose not to ignore missing values and they are visualized by a special horizontal line below all the other axis. If there is any missing value, a line is drawn between one of the attributes (one point on an axis) to the corresponding point located below the missing value attribute's axis on the horizontal line. This is illustrated in the sketches listed in this document.
3. Lines are colored to represent categories. If the dataset has just one categorical attribute, then we group the categories on that and assign a qualitative color map (generated using ColorBrewer) to the categories and encode the lines using these colors. If it has more than one categorical attribute, we group on the column with least number of string values.
4. We have a legend indicating the color and corresponding category and also a rectangular bar whose size is indicative of the number of instances of that category.
5. Labels and ticks must be displayed for each axis along with the values at specific points. However, since overplotting can obscure the range values, it was decided only the min and max values will be shown for each axis.
6. The distance between the axes should adapt to window resizing
7. The filtering box should be light gray to indicate selection but also not to obscure the lines flowing through it. Filtering should also be real time with lines being updated across the axes based on the filtering window(s).
8. Reordering axes should be through dragging the selected axis to the nearest column with which we want to swap with. This be intuitive through smooth transitions.
9. Finally, the background canvas was chosen to be white, to ensure no distraction or interference.

Below are some of the sketches that were considered before implementation:

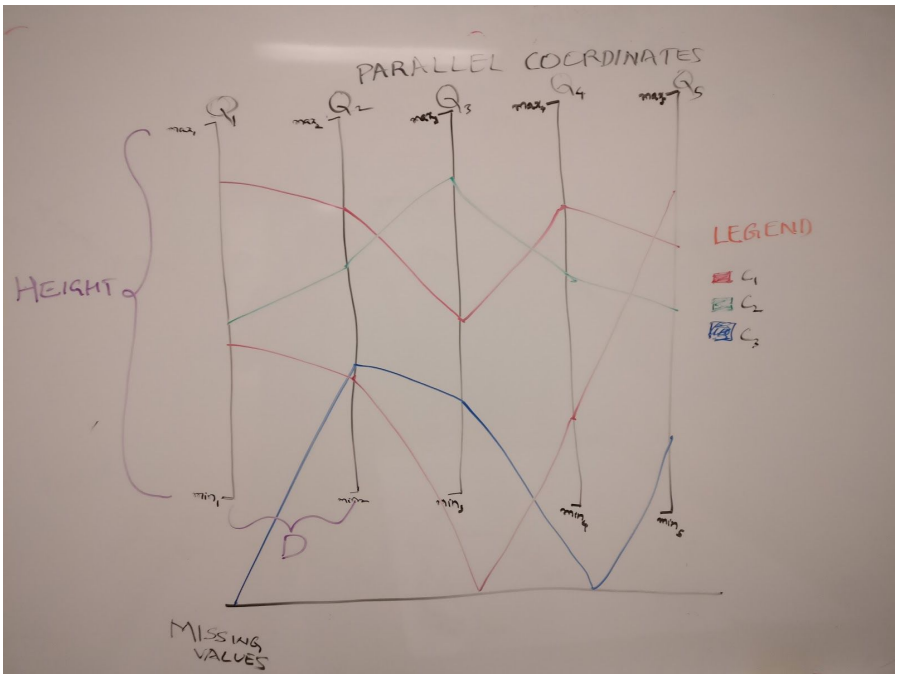

Figure 1: Initial PC viz concept

In Figure 1, all qualitative coordinates Q1,Q2,...Qn have a vertical axis each. Colored lines indicate categories listed in the legend. We can see how the “Missing Values” axis works from the figure as well. We also have (min,max) ranges drawn for each axis rather than multiple scale units.

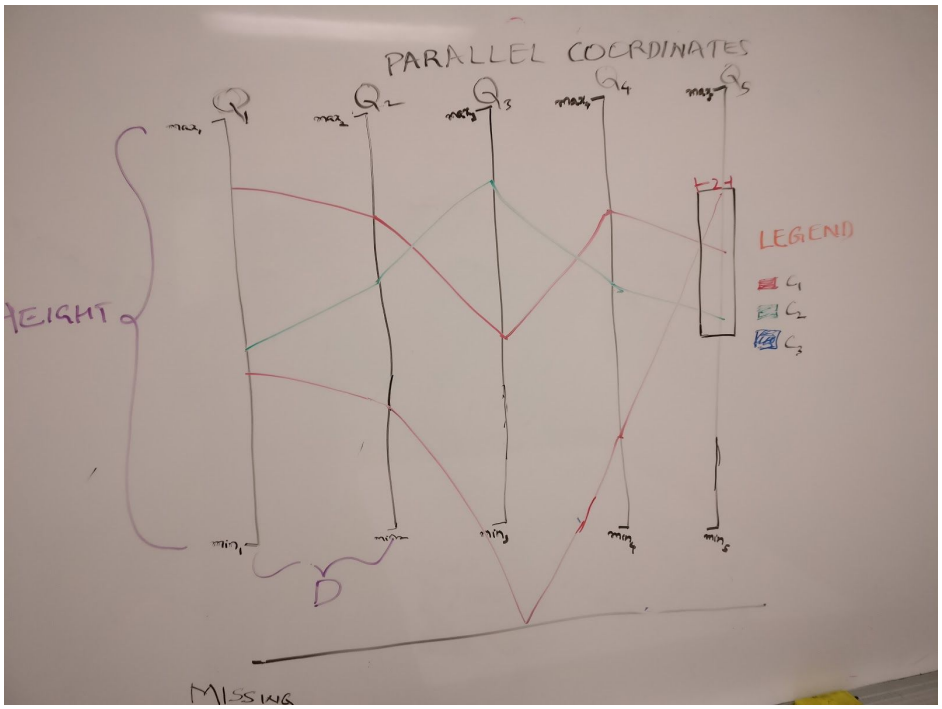

Figure 2: Sketch illustrating filtering

Figure 2 shows the sketch indicating how filtering should possibly work. A rectangle of upto a certain maximum width can be drawn and only all lines flowing through that shall be shown.

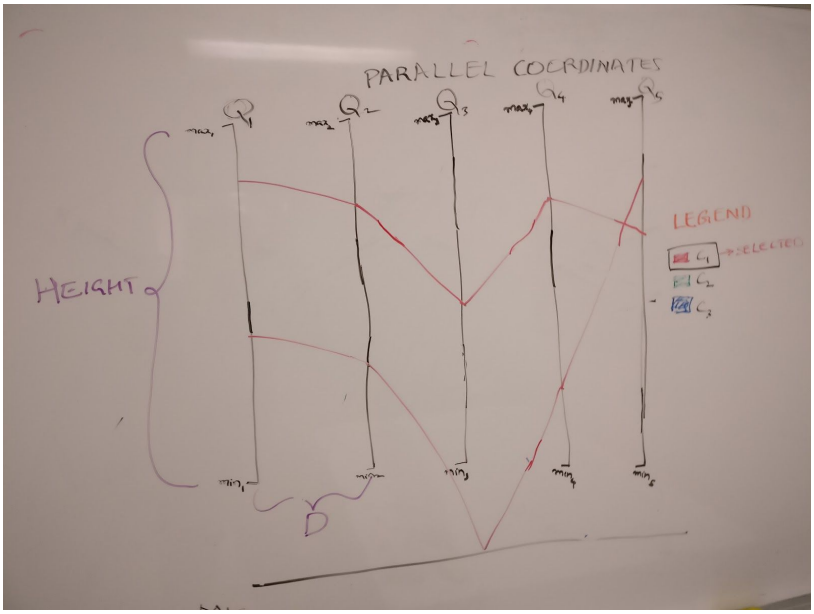

Figure 3: Filtering through legends

Figure 3 shows a sketch showing filtering by selecting a category in the legend. This was considered in the implementation but was not implemented because of the complexity arising through significant number of interaction actions for a mouse click depending on the screen position.

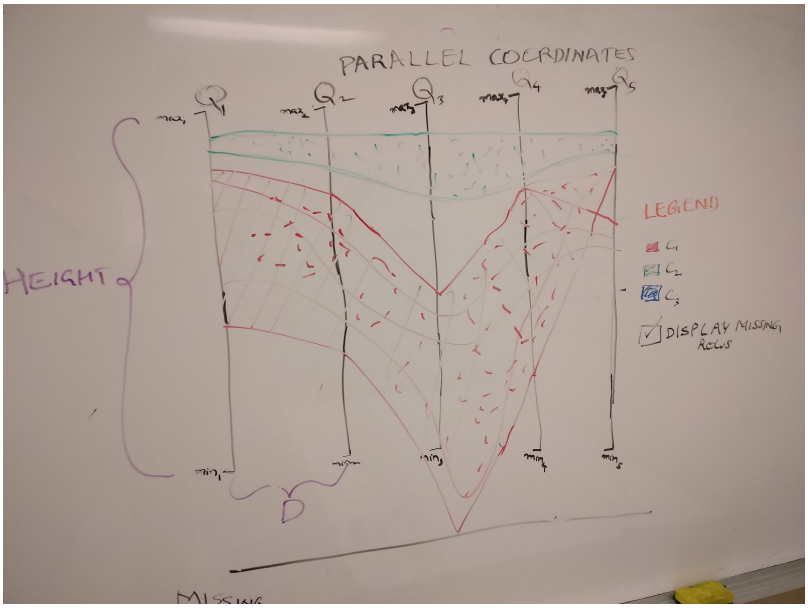

Figure 4: Sketch illustrating possible clustering view

Figure 4 shows a possible view of how lines would look when a clustering scheme is applied on top of them. In the initial concept, we show possible cluster labels using the legend. However, in the actual implementation we avoid showing these labels because:

1. Clustering is unsupervised, so clusters do not posses any pre-defined labels.
2. Color encodes the clusters, so that suffices as number of clusters will be fairly limited (Depending on the user input)

We also don't consider missing values in the clustering scheme, as they create ambiguity. Hence, no missing values axes nor the legend is available in the Problem 2. However, other interactions like filtering, reordering still exist.

Things that changed in the actual implementation:

1. It was possible to actually display all the scale units along an axis rather than just the min and max. Overplotting was avoided by reducing the opacity of the lines drawn.
2. In the sketches, the legends are shown vertically ordered. However, this is not scalable as Cars2 and Nutrients have fairly large number of categories. Instead we try to position in a two dimensional representation. 8 per row, and if exceeded, position the categories on the next column. The legend is arranged below the right rather than to the side of the plot. This creates a nice small multiples view of various categories, making it fairly easy to compare.
3. Displaying axis labels created a conflict in some datasets as they were overlapping due to lengthy strings. To avoid this, we can follow a zig zag order illustrated in the implementation for displaying the labels, thereby avoid overlap.
4. I performed a certain amount of data pre-processing manually to reduced the number of categories. For example, Cars2 had misspellings of Volkswagen (as vokswagen) or abbreviated (VW). We could consider this attribute value aggregation to a point.

5. We have an interactive legend, that gets updated with filtering across multiple attributes. This becomes very useful to identify some trends that can be traced to labels.
6. Inverting the axes works by clicking on a particular axis label. The range of the axes and the lines flowing through it get updated in real time.
7. Reordering the axis is intuitive by dragging after holding onto an axis label to nearest axis with which we want to swap with.
8. Use right click to return to initial state.
9. Clustered data doesn't require legend since color coding suffices. And rows with missing values are avoided in the clustering process.

Additional features for consideration:

1. Legend is updated in real time, showing the number of instances for particular category, with histogram indicating the size, apart from text showing the count. The categories are arranged as small multiples for easy comparison.
2. Colormap for categories is a qualitative color map borrowed from ColorBrewer with further colors considered from the pre-defined ofColor suite.
3. Text labels for axis are arranged in zig zag order to avoid interference. This becomes important in a dataset with lengthy attribute labels like nutrients.tsv
4. Line colors have low opacity to reduce impact due to overplotting.
5. The plot is fairly adaptive to window resizing.

Instructions on usage:

Switch between datasets by entering the key corresponding to the dataset. The instructions are displayed on the console.

Screenshots from actual implementation:

Figure 5 shows my implementation for Cars dataset. We observe the presence of the Missing axes, and below it small multiple views of the number of instances for each color coded category.

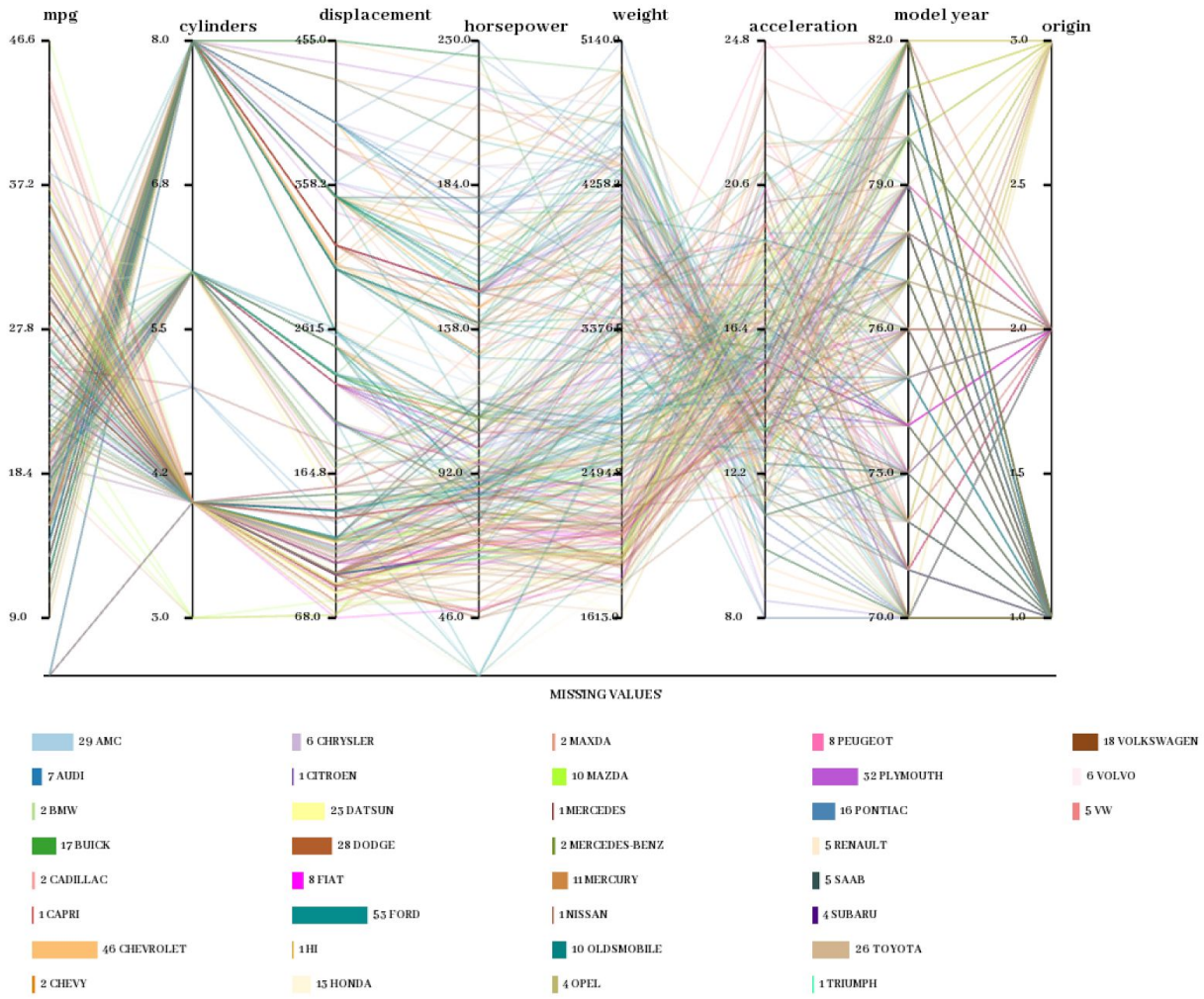

Figure 5: A snapshot of PC implementation for cars.tsv

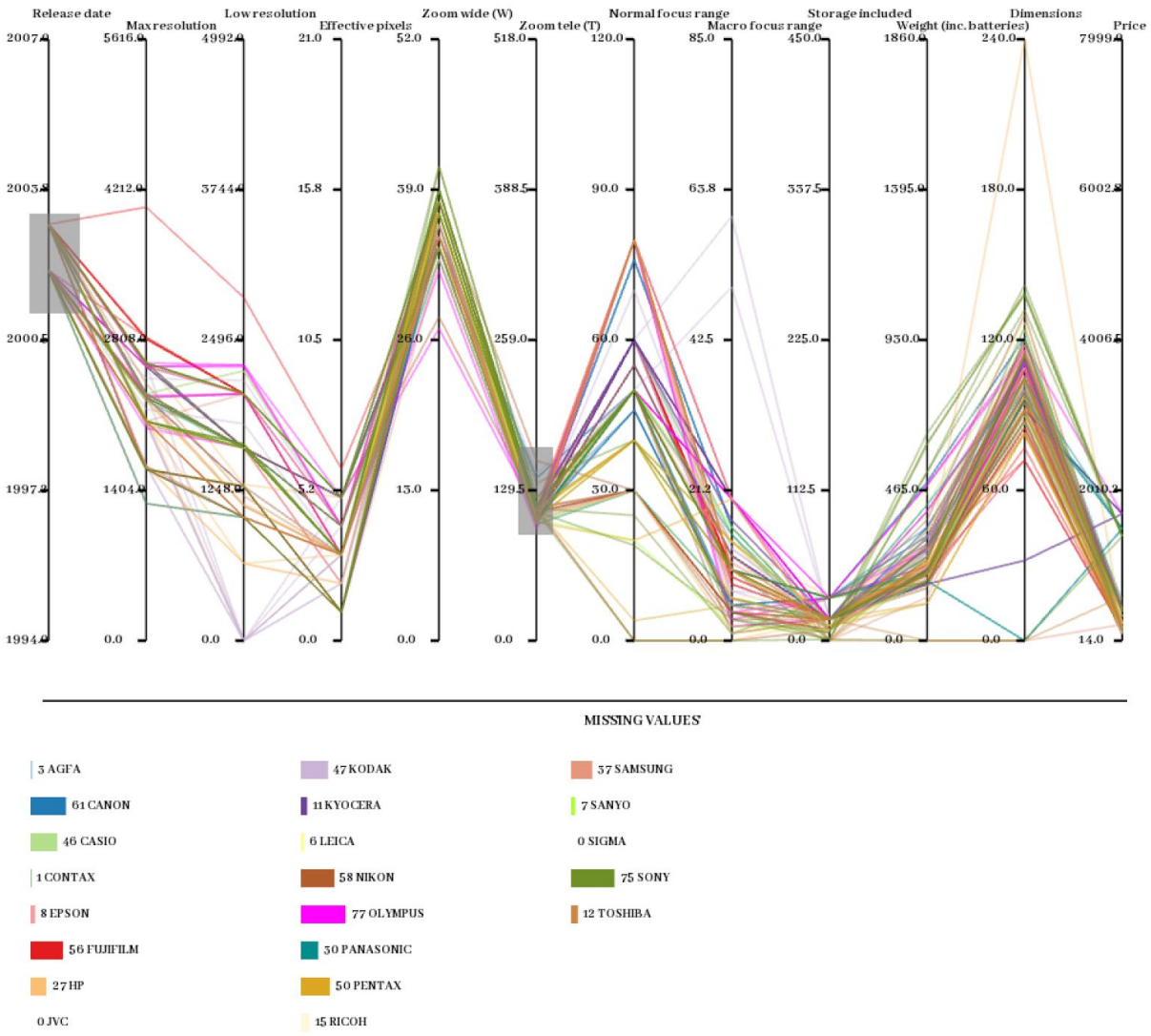

Figure 6: Illustrating filtering across multiple attributes and legend filtering

Figure 6 shows filtering across attributes “Release Date” and “Zoom Tele” attributes for the Cars2 dataset. The filtering is noticeably slower on the nutrients.tsv dataset due to large number of data points.

I illustrate further patterns and relations between attributes in report of A04P02.