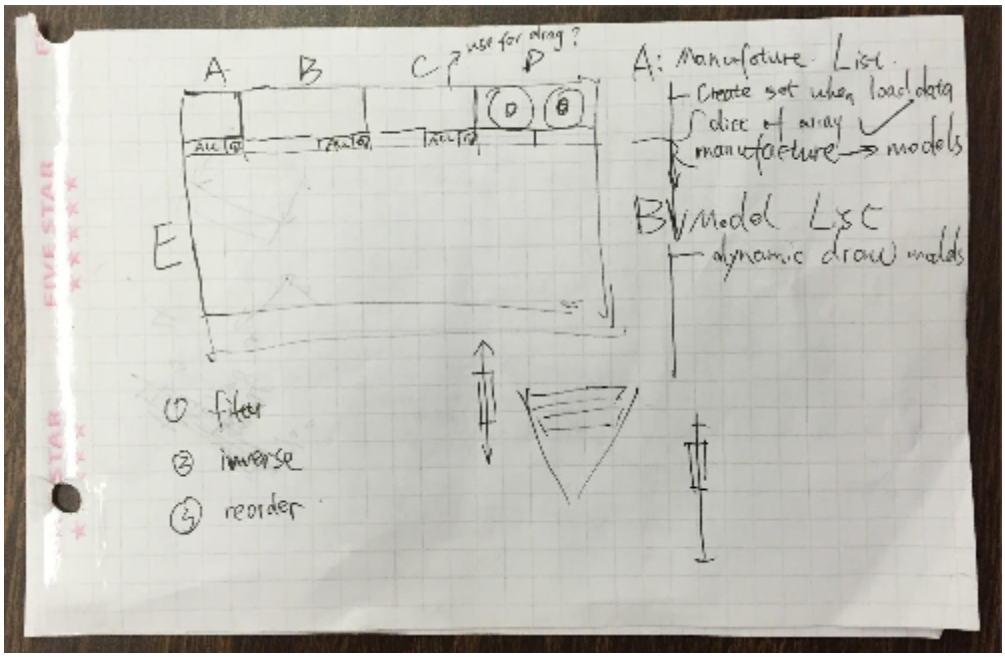
Lab 3 — Parallel Coordinates | CPSC 8810.001 Data Visualization
Name: Chaoren Li
username: chaorel

Part 1.

- I wrote a python script to change NA values in the car table to 0, so that the FloatTable class can load it by changing the delimiter from TAB to comma.

Part 2.

- This is my original sketch for the interface.



- Section A : a list of tags of brands that parsed out from the data set, user should be able to filter the displayed data by clicking on tags
- Section B : a list of models that belong to selected tags, more than one brand selected results in the combination of models displayed here, only for display purpose
- Section C : a list of attributes red from the data set, by clicking on them, one can choose to color the lines according the this attribute
- Section D : to display statistic informations, including selected brands, models, and two pie charts for showing the percentage of selected data.
- Section E : where to draw the parallel coordinates, greatly inspired by the nutrient content example. The functions I wanted achieve include: dynamically load, arrange and label axes; user can filter data by creating boxes on each axis; user can flip and reorder the axes.

- For the issue of overlapping lines, I thought about carefully choosing color and using clustering data as user specified.

Part 3.

- I approximately achieved 80% function in my original design.
- I achieved dynamically loading data and plotting axes and labels by carefully processing the units.
- One can flip the axes by click the I button on the top of each axis.
- I achieved filtering on axes by creating boxes, and this operation can be combined with other filtering and functions such as flipping axes and reordering them.
- Originally, I want to have scroll-menus for displaying different attributes, but I realized it's hard to implement by just using processing.
- The initial ways to reorder the axes is by dragging them. But after I realized the cost behind it, I compromised to just use a plus button at the bottom of each axis.
- I couldn't add the section D in time, will do in the future.

Part 4.

- filtering data is done by creating slidable boxes on axes
- reordering axes is done by clicking the + button at the bottom of each axis
- inverting axes is done by clicking the I button at the top of each axis

- Using a slidable box to filter data is intuitive and easy to use, however, one might feel hard to operate if he/she wants to filter at a precise range.
- Reordering axes could be done by dragging them directly, then you will face more technical issues at backend, for example, how to make the transition effective and expressive during the dragging action, and how to transfer necessary data with this axis to the new logical and physical position.

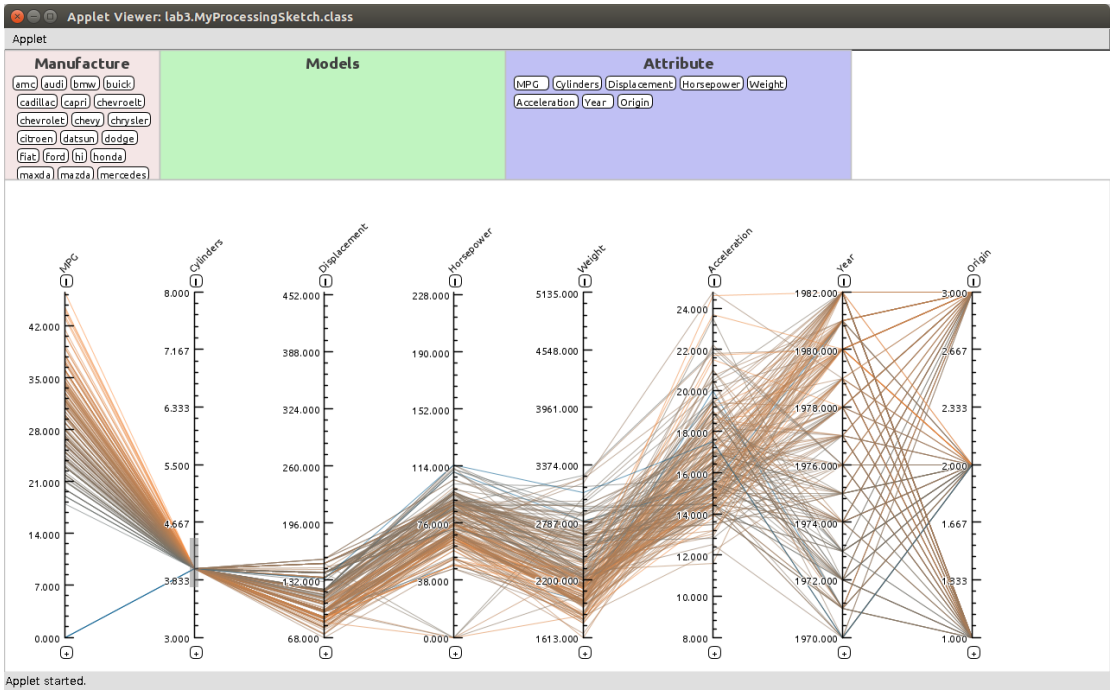
Part 5.

- First thing I did for improving is for the tick labels, labels overlapped with lines are hard to read. So I added a small white contour.
- When first adding the slidable box filter, the interface displays all lines when dragging along so that user can make clear where they want to stop. Later I decided, by using better data structure, to only draw dynamically lines in the real time range, this helps user to determine better.
- I eventually give up the ideal using image to save drawing time. Partially because I added the dynamic adding when dragging the box, also, you don't want to display the all lines when user only choose several brand to display, then there will be a lot situations you need to buffer images.
- TIPS, for large number of items, creating a box under a tag, then switch back boosts the speed, somehow relates to my implementation, hasn't updated all.

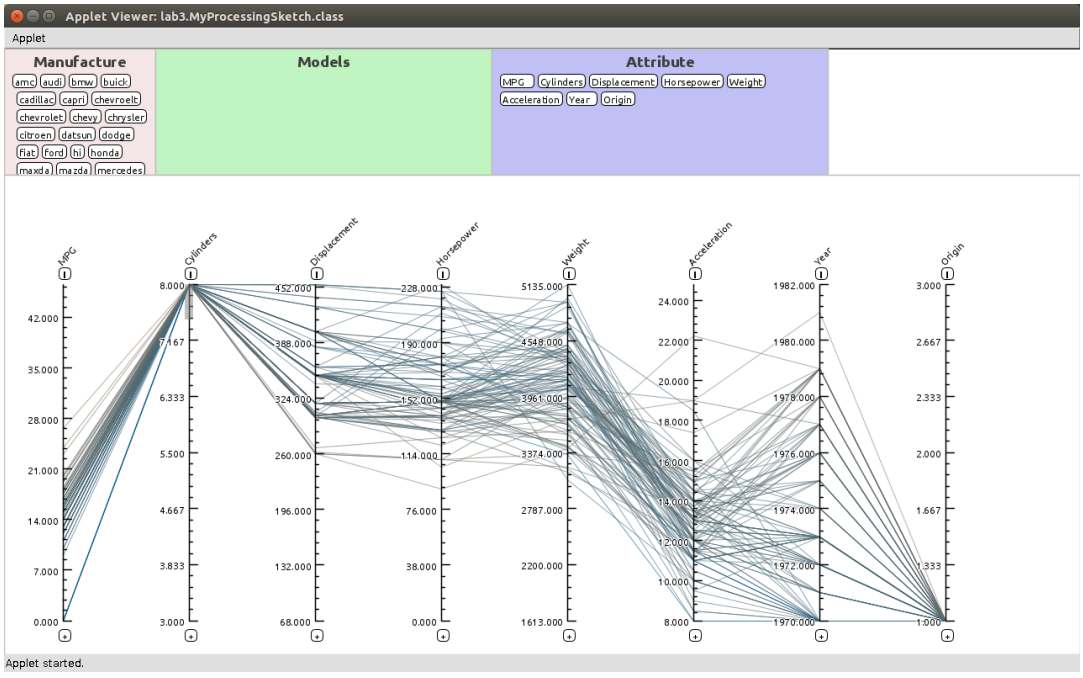
Part 6.

Since the data is huge, the conclusion could be expanded much more if exploring them carefully and more specifically. The following are just some first glance conclusion.

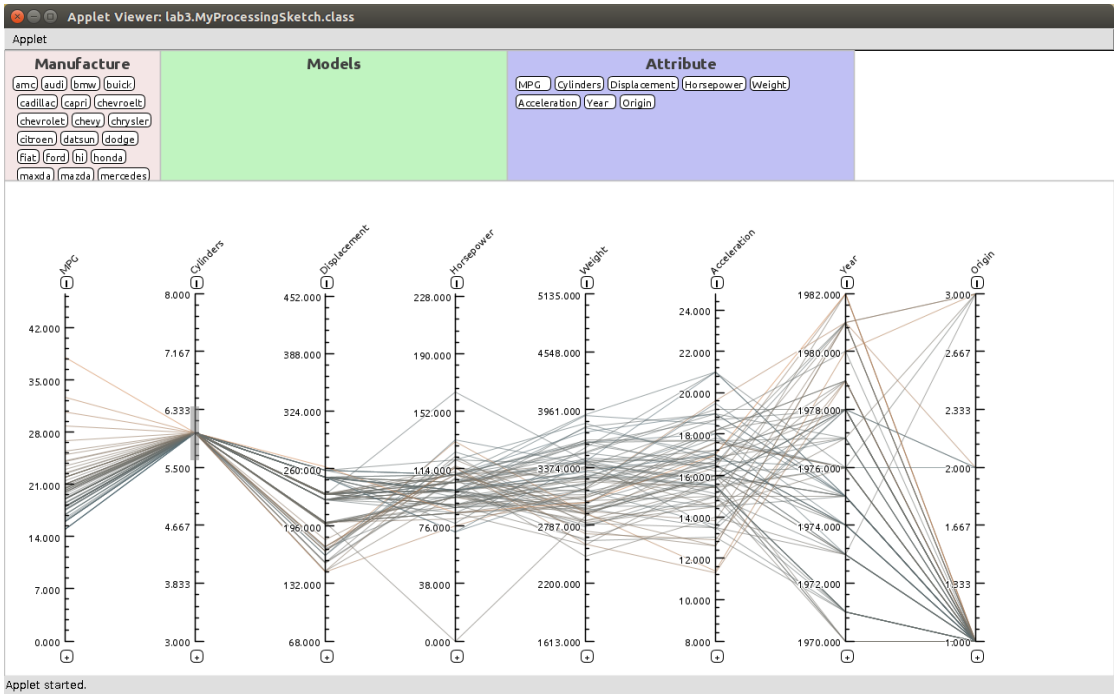
For the car data:



- Cars have higher MPG usually have less cylinders and horsepower, the weight of them is around 2400 pounds and the acceleration is around 16.

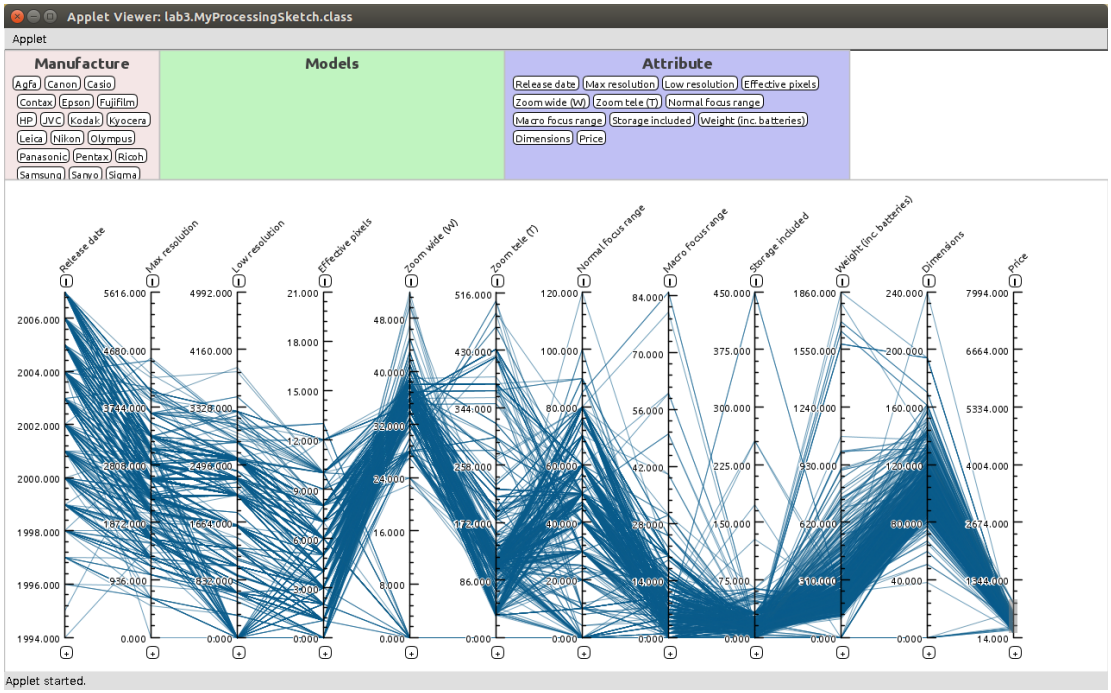


- 8 cylinder cars have lower than 20 MPG and only 1 origin, they have higher than average horsepower and weight, and lower acceleration.

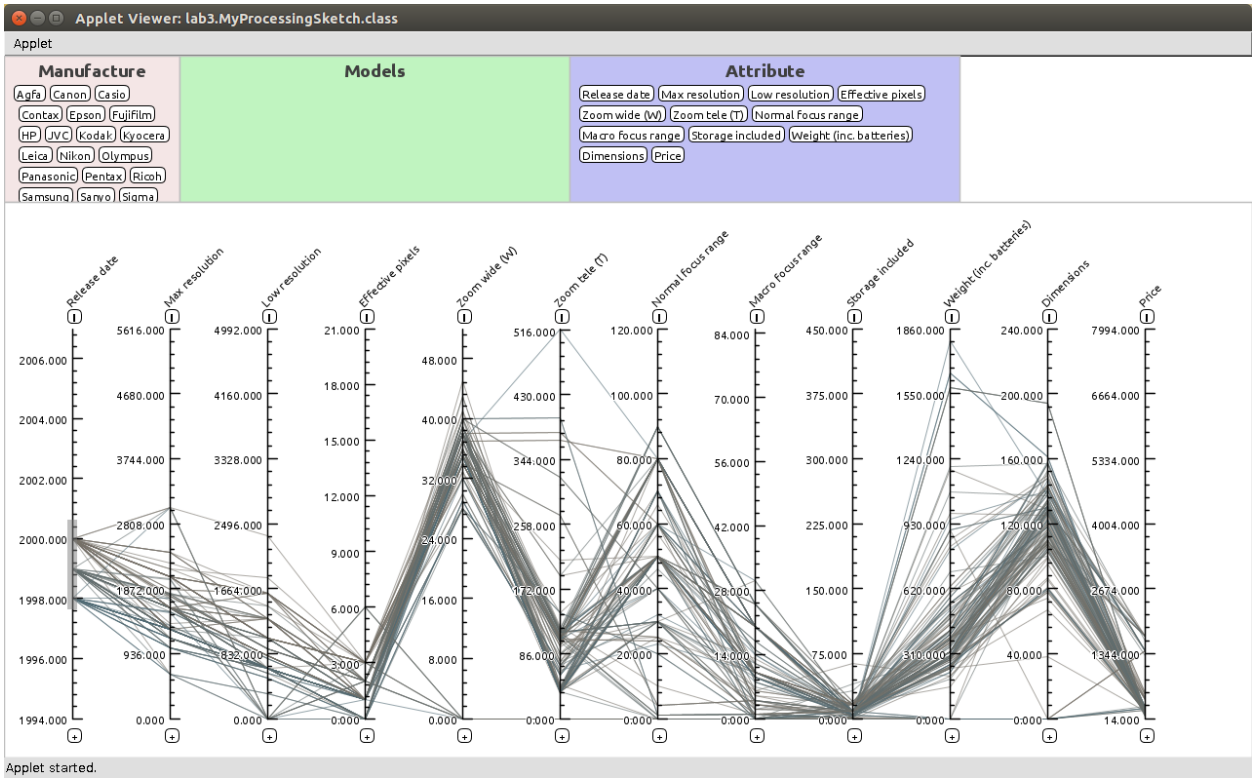


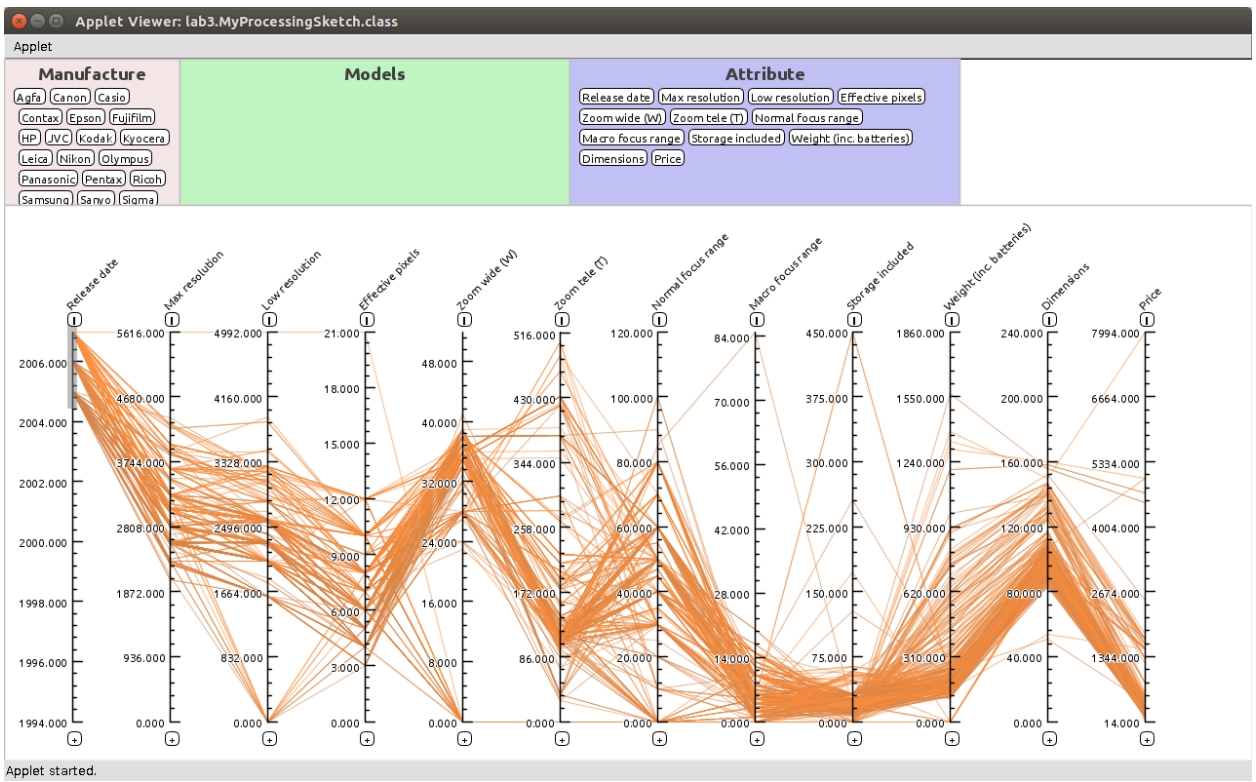
- 6 Cylinders cars have a middle range MPG, average values on horsepower, weight and acceleration, but most of them have one origin.

For cameras



The major price range is from 200 - 850, the zoom wide and zoom tele haven't changed much through the years





The major spec has kept changing through the year is the resolution, other attributes might slightly changed but maintains a table value.