Data Vizualisation - Proposal

Overview, motivation, target audience. Related work and inspiration. What am I trying to show in the data viz?

For this project, we choose to look at a movie dataset. The aim is to have an overview of some movies (from a The movie Database dataset) by looking at how they are linked together. To link them, we would like to look at usual features like actors, directors, year, type or score, but also to consider the work of crew members that are neither actors nor directors but are present in a lot of movies, like hairdressers, CGI techniciens, sound effects directors and so on. We hope that it will highlight new relations and propose a new perspective on movie dataset. We want to integrate those different dimension in a unique interactive visualisation, addressed to anyone who want to have a better understanding of the relation that exist between well known movies.

This idea came to us by looking at some visualization that show how some entities were linked in a particular universe or space. For instance, we saw network visualisation about Star Wars [1], Star Trek [2] or even Subreddits [3]. We found this pretty cool and interesting and looked at which universe/space we could focus on. We decided to look at movies in general and not to focus in a particular universe. We found our dataset and realized that we may have more informations than expected (not just actors and directors), which can be even more interesting and original. A lot of visualization and analysis of movie dataset already exist. For instance [4] propose cool interactive visualization and there are a lot like this. However, we didn't find any visualization that show link made with the crew members. We would like to let the user explore the data by letting him go through different point of view, highlighting different feature that he could choose.

[1] http://www.kirellbenzi.com/blog/exploring-the-star-wars-expanded-universe/
[2] https://datascopeanalytics.com/startrekviz/
[3] http://rhiever.github.io/redditviz/
[4] https://ngchwanlii.github.io/datavis/imdb-genres.html

Dataset : https://www.kaggle.com/tmdb/tmdb-movie-metadata/data
A dataset of 5000 extracted from The movie Database (https://www.themoviedb.org/?language=fr). It contains generic informations about the movies (title, language, production country/companies, genres, budget and revenue, and grades). For each movie, we also have complete cast and the crew. We may want to get more data from other dataset/website (ImDB).

GitHub: https://github.com/qkuenlin/DataViz

Bollé Timothy - 11413325 (UNIL)
Kuenlin Quentin - 234033
Neu Virgile - 224138

CODE BOOK

With this project, we propose an overview on the movie industry, more specifically by looking at the relations between well known movies. Usually, this is done by looking at the actors or directors they have in common, or by looking at features like budget, revenue or ranking. Here, the idea is also to point out the work of crew members that are neither actors nor directors but are present in a lot of movies, like hairdressers, CGI techniciens, sound effects directors and so on.
We will use an unique interactive visualization to integrate those different dimensions. It is addressed to anyone who want to explore the various relations and have a better understanding of the movie industry.

This idea came to us by looking at some visualization that show how some entities were linked in a particular universe or space. For instance, we saw network visualisation about Star Wars [1], Star Trek [2] or even Subreddits [3]. We found this pretty cool and interesting and looked at which universe/space we could focus on. We decided to look at movies in general and not to focus in a particular universe.

We looked at various movie dataset and found one that have more informations than expected (not just actors and directors), which can be even more interesting and original. A lot of visualization and analysis of movie dataset already exist. For instance [4] propose cool interactive visualization and there are a lot like this. However, we didn't find any visualization that show link made with the crew members. We would like to let the user explore the data by letting him go through different point of view, highlighting different feature that he could choose.

The dataset we chose contains 5000 movies extracted from The movie Database (https://www.themoviedb.org/?language=fr). It contains generic informations about the movies (title, language, production country/companies, genres, budget and revenue, and grades). For each movie, we also have complete cast and the crew.
The dataset is available here : https://www.kaggle.com/tmdb/tmdb-movie-metadata/data

To increase the interactivity and the information available in the visualization, we may want to combine data from other sources, like wikipedia or ImDB.

The data was explored using a Jupyter Notebook and Python. We started to clean the data and to integrate it into a sqlite database. The advantages of this format is that it facilitates information retrieval with simple sql queries, and that the database is a simple file that do not need to run on a particular server.

We examined the possibility to make requests directly from javascript, in a similar way we would have opened a file, but it seems complicated to do in this particular language. In fact, there exist a script to make SQL request to a sqlite file in javascript (sql.js). The first main issue was to get the sqlite file from into the browser. As it is a web page, there is no possibility to load files that are not JSON, CSV or other separated data file, for security reasons. The first work around we explored was to embedded the sqlite database into a JSON file to be able to load it, but there was problem because it is a binary file, and JSON is read as a string, so some reserved characters broke the JSON. Even using a base64 encoding (binary to string) didn't work, the database was broken when decoding the string. Then we managed to create a DB inside the browser, not by directly importing the sqlite file, but by recreating it with SQL commands "CREATE TABLE" and "INSERT INTO". But after that, the second main issue arises. It was super slow. It took nearly 10 seconds to load the page, whereas with simply JSONs it took 1. And the third and last issue that made us give up this idea was the way the data was returned by queries. With JSONs we are able to do filter operation that returns us another JSON array. With SQL, the result is an array of arrays, not an array of JSON, so the selection of attributes become way harder and we have to use indices instead of name, that complicates the code.

Once the data was put in the database, we had 104842 person, cast and crew mixed, 4803 movies and 235665 links between movies and people. The crew was splitted into different department and jobs. All the person from the cast were assigned in the same department and job: Cast > Actor. The problem here is that the main actors and extras/stunt doubles are in the same category, which results in a huge category that may be hard to handle later in the visualization. An idea to separate some of the person, could be to extract a list of well known actors.

As a first try, and to develop the visualization, we chose to take only movies with a popularity greater to 50 (we don't know exactly here what it represent but it range from 875 to 0). [BOXPLOT] We also removed the links and the associated person where a person was linked to only one movie. For now, we also removed the person that were under the department CREW, because it wasn't really clear what it meant and because we want to initially focus our visualisation on the department level (and not the job level).

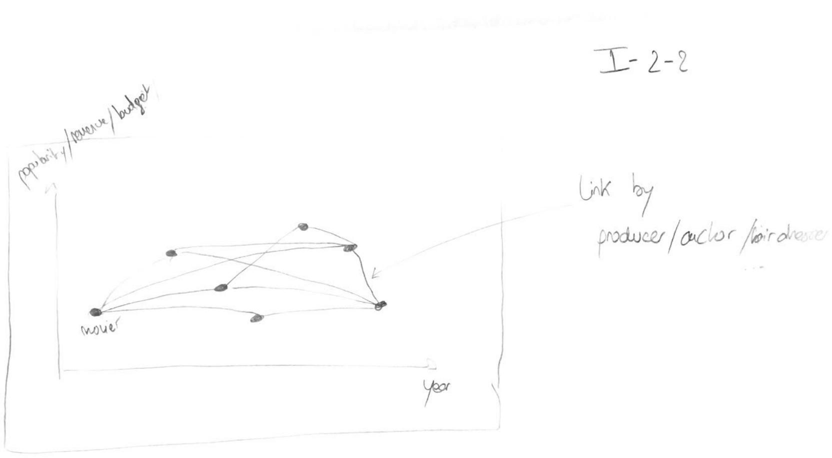
The next step of the project is to establish a first visualization design and experiment it.
To do this, we tried to follow the steps proposed by McKenna et al. [https://design-worksheets.github.io/], the first step (Understand) had been already done.

To resume, we want to:

- Show relations between well known movies and present general information/statistics
    - relations at the crew level => highlight less known jobs
    - relations at the cast level (maybe limited to famous person)
- Present informations about particular movies

The user should be able to explore the data, find interesting links and think about new aspects of movies. It could also be useful to find movies similar to one movie the user liked.

Because we want to display relations, we first thought about a classic node-link diagram.

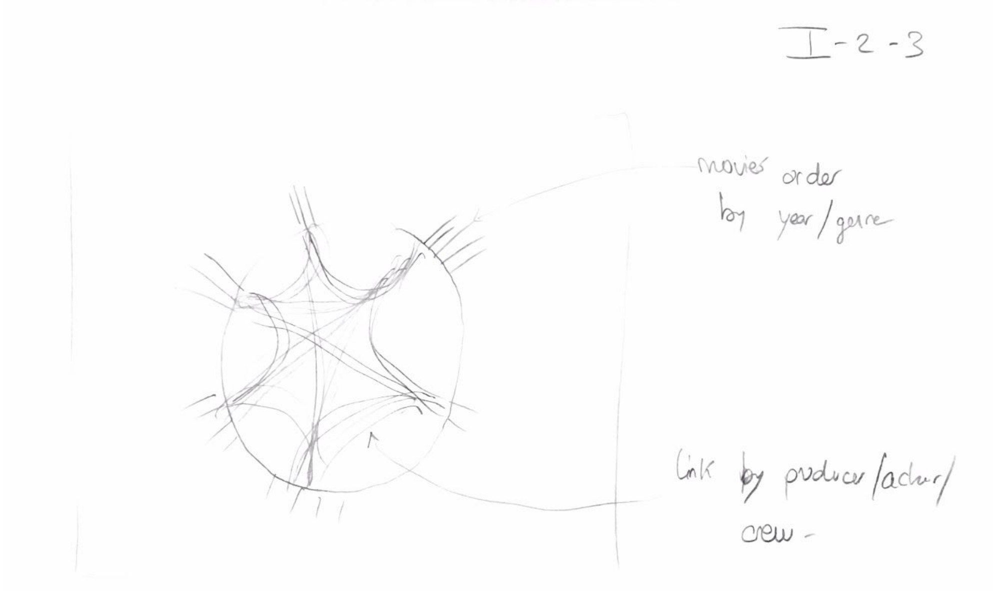


Sketch 1: First visualization idea

The idea was to display the different nodes on a two dimensional diagram where the x-axis could have been the year, and the y-axis the popularity, the revenue or the budget of the movie. Then, edges would have been drawn between nodes linked by same crew members or cast members.

However this kind of visualization would have been instinctive and quite easy to order, we fear that the potentially high number of links make the visualization hard to read and to navigate. Here a force-directed layout seems not adapted.

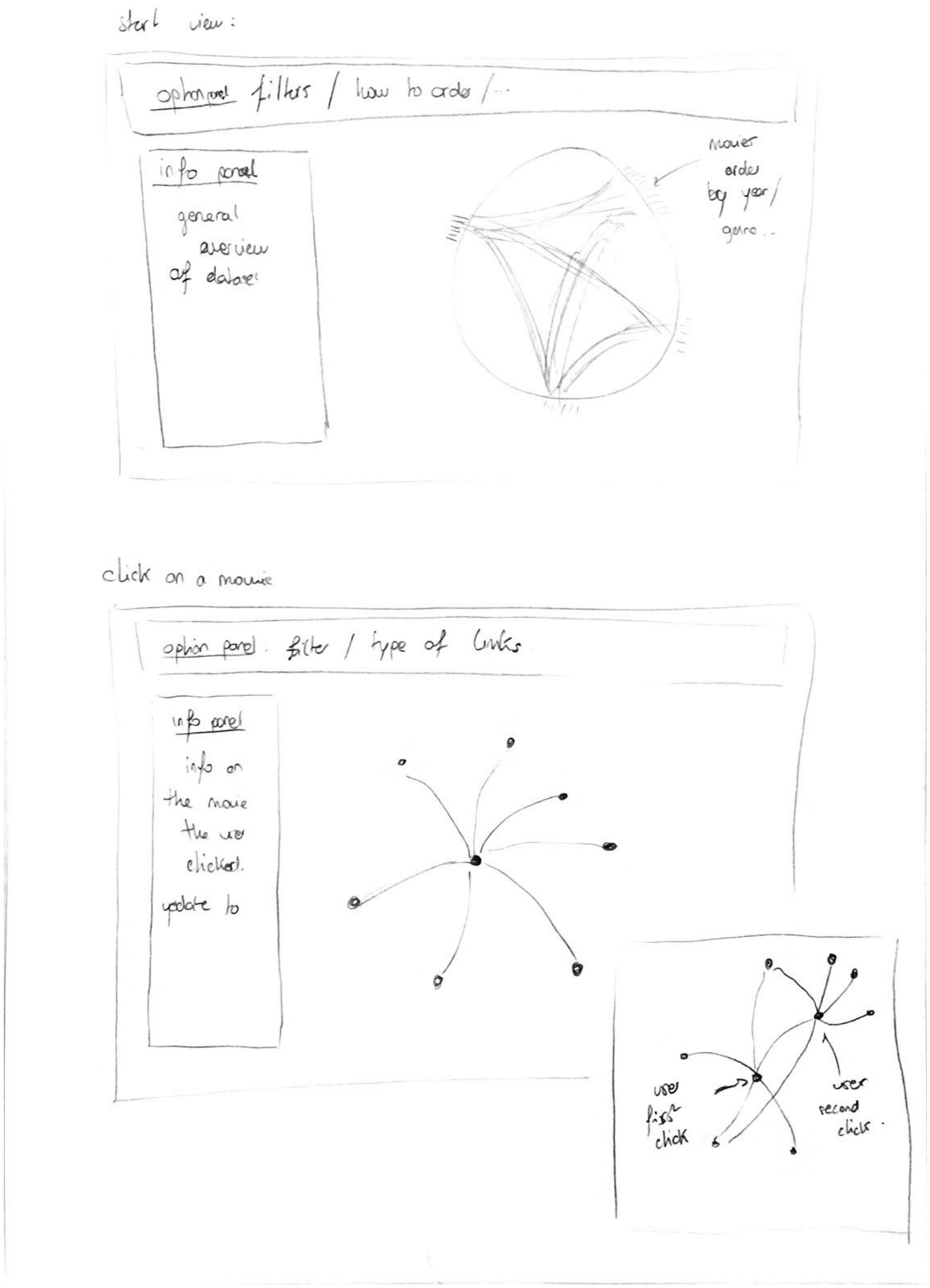
The second idea was to use circle layout with edge bundling.



Sketch 2: second visualization idea

Here, it could be possible to order the movies by year or genre (or even maybe by revenue, budget or popularity). Then, as for the first layout, the links would have been made if two movies share a crew or cast member. With this layout, the general overview is cleaner but we find this type of layout less instinctive and maybe harder to interact with.

Finally, we decided to try to integrate both layout, depending of the level of detail the user want, in order to keep the benefits of both layouts.


start view:
option panel filters / how to order / ...
info panel
general overview of dataset
movies order by year / genre ..
click on a movie
option panel. filter / type of links
info panel
info on the movie the user clicked.
update to
user first click
user second click.

Sketch 3: Final visualization idea

To have an overview of the data, we display a circular layout with edge bundling. When a user clicks on a particular movie, we switch to traditional node-link diagram and display only the movies connected to it. If the user clicks on a second node, the movies linked to this one would be displayed. Here the layout could be a collapsible force layout or an organized layout in a two dimensional diagram, with x-axis as the year and y-axis as the popularity or revenue. We will try those different possibilities and see which one is more adapted.

This way, we keep the advantages of the circular layout to have an overview of the data, and the advantages of a more traditional layout, which is to be instinctive and easy to manipulate. In both views (general and detailed), we will have an option panel at the top that will allow to filter the data and select which type of link the user want to display, and an information panel where general or particular information and statistics will be displayed.