Process book: DataMoviz

Quentin de Longraye, Aymen Gannouni, Victor Le

December 19, 2017

Figure 1: Datamoviz mockup

Contents

You may zoom on all figures using your pdf reader.

1 Presentation of DataMoviz

1.1 Overview, motivation and target audience

Overview, motivation, target audience + Questions: What am I trying to show this my viz?

DataMoviz is a web application, that aims at involving the user in a unique experience to discover all the movies from The Movie DB in a very interactive way. This project allows to get impressive insights on the filmmaking scence scince the beginnings. What motivates us the most is the opportunity that we have to deliver fascinating observations from raw data through data visualization. There is no restrictions in term of prerequisites for our target audience. Anyone who has interest in movies is heartly welcomed to visit DataMoviz!

1.2 Related work and inspiration

Related work and inspiration

2 Exploratory analysis

2.1 Dataset

Presentation of the dataset. Dataset: where does it come from, what are you processing steps?

The dataset comes from The Movie DB (TMDB). Data was imported using a list of all IMDB movie IDs by querying the dedicated API provided by TMDB. From this data, we built a MongoDB database. Usage of a NoSQL database was extremely useful as it allows to write very permissive queries and allows data to be into an inconsistent state (if some data is missing).

We exposed this database through an API we wrote using ExpressJS. We created dedicated endpoint for all concerned components requiring to query the data in a specific way. From the subset of filters sent by the frontend application, the backend creates query to get, aggregate or count data, and returns the result.

2.2 Data preview

Exploratory data analysis: What viz have you used to gain insights on the data?

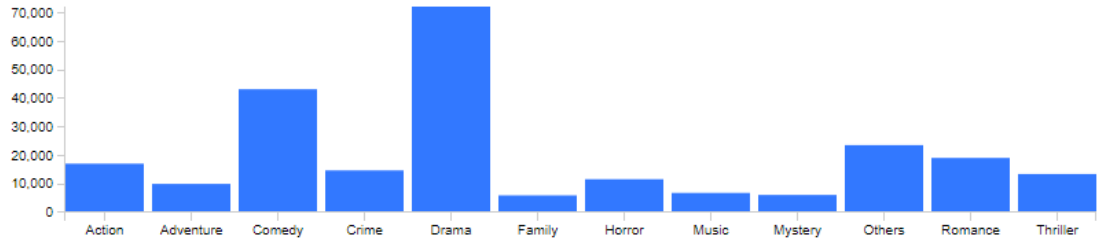
2.2.1 Existing visualizations

- IMDB itself - The movie db - Other graphs

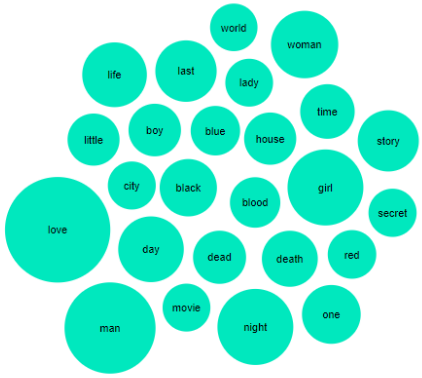
2.2.2 Data previsualization

RAWGraphs Description TBD (useful to compare the kind of visualization pie VS bar chart, bar chart VS bubble chart, ...).

Tableau (screens + explanations).



(a) Raw graph for genres



(b) Raw graph for movie title keywords

Figure 2: Two RAWGraphs which helped to choose relevant visualizations

3 Solution's build-up

3.1 Considered visualizations

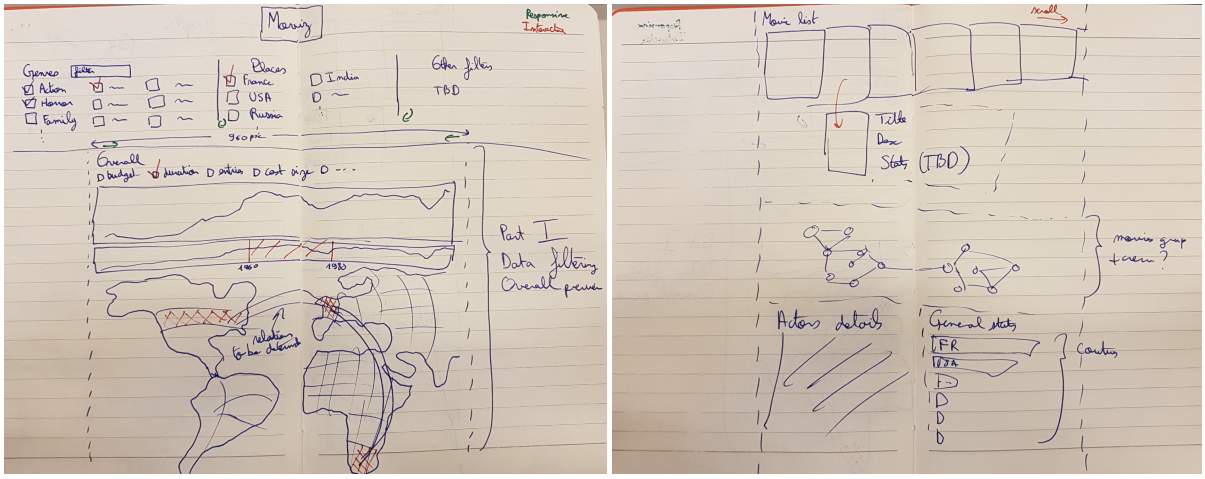
Designs: What are the different visualizations you considered? Justify the design decisions you made using the perceptual and design principles. Did you deviate from your initial proposal?

3.1.1 Initial ideas

The current implementation of the website is already very close to the initial idea. We wanted to build an application which allows to see how the movies production system (e.g. Hollywood) has evolved over time. To do so, we wanted to import and use movies data to show, for instance, which actors are the biggest influencers of the movie scene. We wanted to focus on a map and graph display to outline evolution and repartition of movies accross countries and over time, and show the relationship between them.

We eventually discovered that the data itself brings a lot of different insights about movies repartition and evolution over time. We started to build a lot of different visualizations and decided to shift our project to a filter-oriented application in which you can (re-)discover the movie scene based on the filters you choosed. This allows to learn a lot from specific genres, time periods or countries, ... Thanks to all these filters, users can filter data to thousand of different ways and make easy comparisons. We decided consequently to change from a single view to a more modular and decoupled application with a lot of different components displaying one specific subset of the movie scene. The main idea is to bring an overview and allow the user to deep into the data to get more details about movies (fig. 3).

We improved our dataset by calculating the more used words in movie titles, and added a movie slider so users can see which popular movies are matching the entered criteria. This allows to learn how movie distribution evolved over time, but also to discover new movies.



(a) First page with movies overview

(b) Second page with movie details

Figure 3: A first draft of the application

3.1.2 Incremental improvements

TBD

3.2 Implementation

Implementation: Describe the intent and functionality of the interactive visualizations you implemented. Provide clear and well-referenced images showing the key design and interaction elements.

3.2.1 Presentation of the visualizations

In the section, we will present each visualizations and the insights it brings about the data.

The filters bar and panel The filter panel allows user to see which filters are currently applied on page's data (fig. 4). It permits to give an explicit explanation about page current's sate. Users are allowed to remove a filter by clicking on the small cross near the filter description. They can also click on *more filters* to have access to three more page filters: filtering by keywords in movie titles, by genre and by content rating.

Figure 4: The filters bar with applied filters

This panel also supports keyboard shortcuts such as *esc* to empty the search field and to close the filters section.

The interactive map The interactive map is the first visible visualization when going into the Data-Moviz website. This visualization allows to understand which countries are more involved into producing movies, considering all the applied filters by other application components (see section 3.2.2 for more details). When the user selects a country, the maps colors changes smoothly to let the user understand

the transition between the two states (unfiltered to filtered state). The figure 5 shows what happened after selecting a country (here, we click on France).

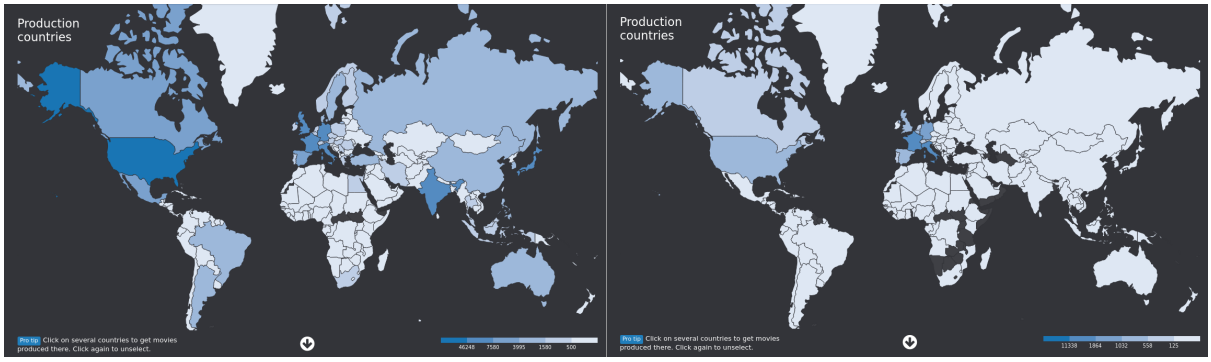


Figure 5: Initial draft of the filtering system

As we can see, all colors intensity decreases. We still have a lot of colored countries because all of them participated into the production of some movies together with french production companies. The color intensity allows the user to understand which countries were more involved in movie production based on current filters, but also which countries co-producted movies with France the most (here: Italy and Germany).

The time evolution The second visualization is a time evolution display and selector. It allows to see how the number of produced movies evolved over time, considering the selected filters. Users can interact with this visualization by selecting a time range in the subchart (fig. 6). This will create a new filter and restrict all gathered data for a specific time range. All the other visualizations will be updated consequently.

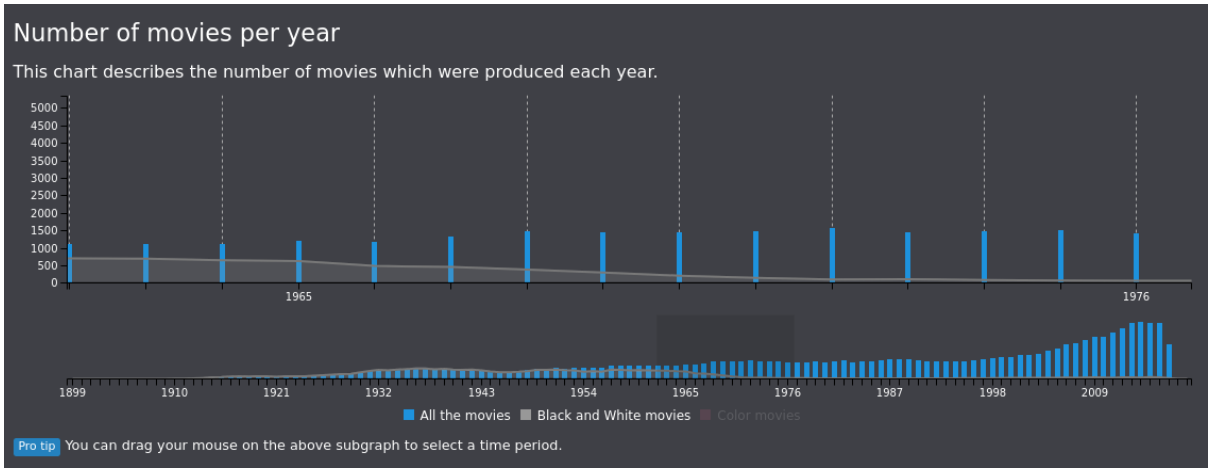


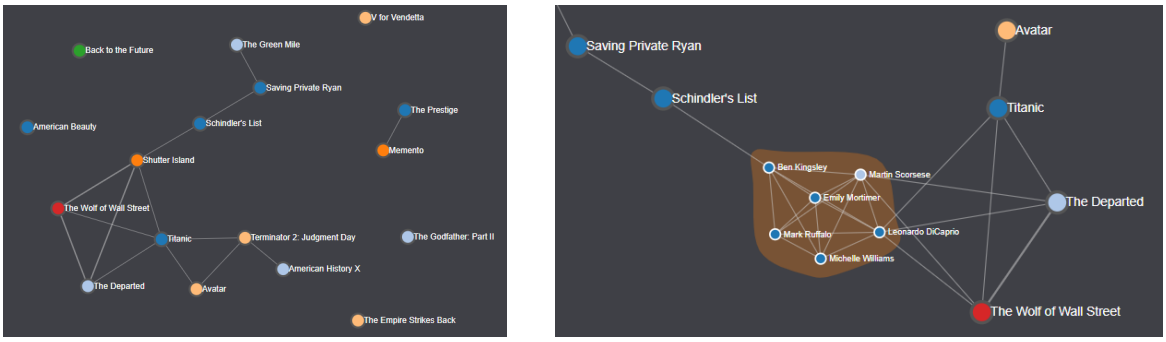
Figure 6: The time evolution with 1970-1985 period selected

In figure 6, we can see that black and white movies were also selected. This allows to see how black and white movie distribution evolved for the time range.

The movies statistics

The actors network The actor network is a vizualization that shows how the movies are related (fig. 7a). The relation is based on the shared crew members. Each movie and actor are represented as a node in the graph. The color of this one indicates its genre or role respectively. A edge is defined between two

movies if and only if there is at least one actor who plays in both of them. Its width is correlated to the number of actor in common. By default, the graph only shows the movies : to show the crew who plays for one, we can just click on the corresponding node. For example in figure 7b, we clicked on the *Shutter Island* node to make its actors visible. These latters are fully connected because they plays on the same movies. There is edge too from actor to movie which indicated that he also plays on linked movie.



(a) The initial network

(b) The network after clicking on Shutter Island movie

Figure 7: Two views of the network

The movies details The network is highly interactive and customizable. We can change the number of actors, crew members and movies to show thanks to some sliders (see figure 8). This vizualization offers a way to navigate freely through the network : it supports panning, and zooming. If the user got lost, he can reset easily to default zoom by pressing a button, no need to reload the page. Moreover, if we are not interested to having the movies name and actors name shown, we can turn it off (figure 9). Anyway, a hover on a node shows this information in addition to indicating the role or genre respectively for actors and movies.

Figure 8: Sliders allowing to change the quantity of displayed data

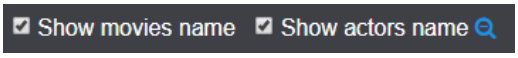


Figure 9: Checkboxes allowing to tweak the display

3.2.2 The filtering system

The filtering system is one of the most important features of this visualization. It allows to interact with the page and asserts all graphs are in the same, coherent state. Without this system, we would not have been able to provide an visually interactive web application. It was built by creating a single shared state between all components. Each time a component needs to update the filters, it triggers an event which is relayed to the other components. The following draft (fig. 10) describes how the system works. It was designed before implementation.

When the user interacts with the page (for instance by clicking on a country of the map visualization (1)), a filters update event is triggered into an event bus (2). This event contains all the filters that are currently applied to get the data for each component of the application, with a small addition or deletion (the example shows a constraint in which the user wants to filter to get only movies produced in France). The event bus redirects this filter to all the subscribed components (3). Based on the new filters, they may query the API (4) to get refreshed data, and can update the visualization consequently (5).

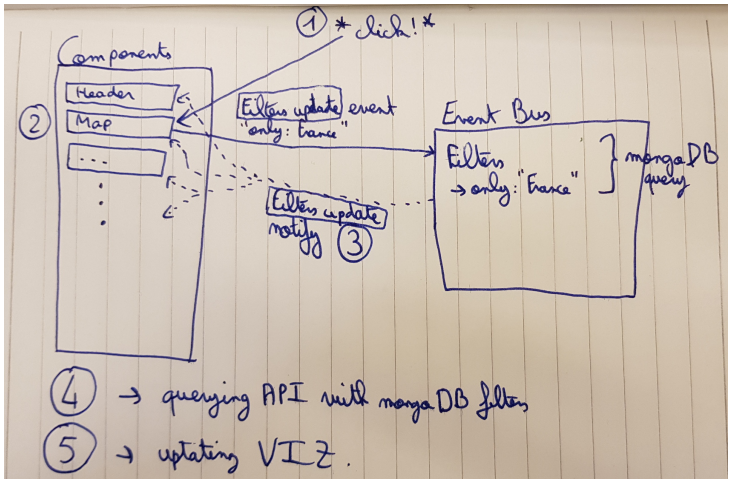


Figure 10: Initial draft of the filtering system

This system was designed following simple computational thinking concepts, by reducing the filtering complexity into a single, atomic and context-independent (stateless) set of filters which can be directly passed to MongoDB to filter data. We also take leverage of HTTP cache to allow the browser to cache all previous requests, so we potentially avoid overwhelming processing on the server.

The final implementation is presented in the figure 11. This event happened after clicking on France country. We can see in the event payload that an object is stored. This object describes a MongoDB filtering subquery which may be used by the server to restrict which data should be used during data selection, aggregation or counting.

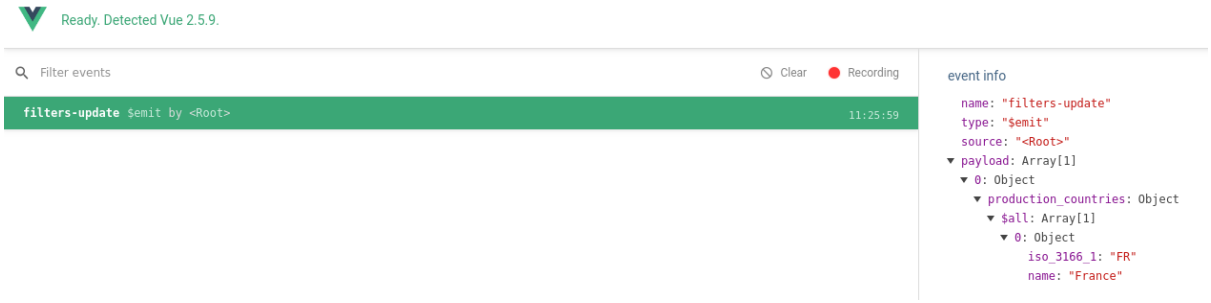


Figure 11: An emitted event when selecting a country

This system is extremely flexible as it allows to add new components without impacting other parts of the application. Each new component should only define two behaviors (if required): filters update triggering after user interaction, and filters update handling when another component updates the filters.

3.3 Evaluation

Evaluation: What did you learn about the data by using your visualizations? How did you answer your questions? How well does your visualization work, and how could you further improve it?

We discovered a lot of insights about movies when working on the visualization. The first interesting fact was about the most used words in movie titles. It was interesting to see which words are more used in titles, and how it evolved over time. We also discovered how each decade of the twentieth century influenced the movie genres, such as westerns during the 40's or porn movies during the 70's. Our visualization also brought some interesting informations about outliers, such as the four movies from Greenland.

Our questions was to better understand how movies distribution and the movie scene in general evolved over time. Thanks to the plenty of implemented filters, we were able to understand how countries were involved in producing movies over time or by comparing time ranges, and the graph permitted to outline some unexpected relationships between movies through its casting.

The final visualization works as expected. Some further improvements could be to improve the overall performances, by implementing a requests aggretator in charge to delay and aggregate the requests to the server into an unique one. This would reduce data traffic between the page and the server, and would improve overall performances.

The application was built using mobile-first methodology. This allowed us to bring a fully responsive and working application on mobile phones. Expected performances are however high and some smarthpones may have some difficulties to render the full page. This may be improved by limiting the number of queries to the server as stated in the previous paragraph, or limiting SVG usage (which would require some important changes). Finally, some minor improvements could be done to enhance compatibility with other browsers. On the visualization side, more visualization could be added, and more features such as movies suggestions, random filters selections and more interactions with the charts.

4 Peer assessment

- Preparation – were they prepared during team meetings?
- Contribution – did they contribute productively to the team discussion and work?
- Respect for others' ideas – did they encourage others to contribute their ideas?
- Flexibility – were they flexible when disagreements occurred?

List of Figures