RTS: The Swiss TV Mirror

A Data Visualization Project
for
EPFL CS-480

KÖSE Süha Kağan
YAGHINI Mohammad
AJALOEIAN Ahmad

Fall 2017

1. Abstract

The popular media is an (imperfect) mirror of the society. In this project, we want to study the effect of social trends and events of the past century on TV/Radio broadcasts. The 20th century saw the invention and ubiquity of the broadcast media: first radio and then television. At the same time, many international events (World Wars, the rise and the fall of Communism, the formation of European Union, etc) as well as national incidents (sexual revolution of 1960's, the establishment of women voting right) took place in this period. All of these events had profound impact on the society, and as a result on broadcast media such as TV and Radio. Having access to a wealth of information about TV/Radio emissions all the way from 1930's until today, we believe we can track these events and by means of a data story and a convincing visualization, highlight the events of the decades past.

2. Project Overview

2.1. Motivation

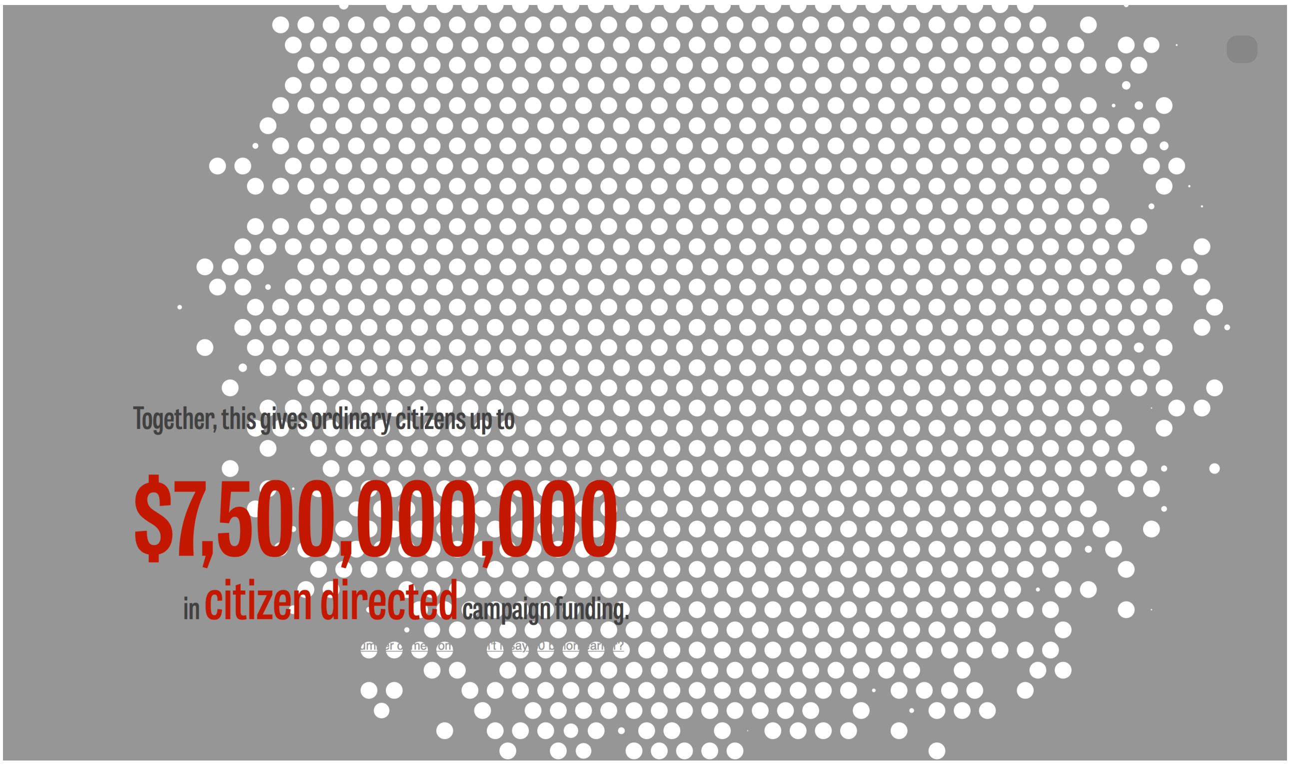
Our main motivation is the sheer amount and scope of the structured data provided by the Swiss public broadcasting organization, RTS. Although the majority of the data is understandably recorded in recent years, it covers an expansive temporal span (from 1900's until today).
More importantly, given that in March 2017 the Swiss will decide by popular vote whether to abolish the "Billag" (which also supports RTS), such a study is very much relevant and timely. An interactive and easy-to-understand visualization has the potential to showcase the added value of this public organization over the course of the past century.

2.2. Target Audience

Given the last point, our target audience is the general public. Ideally, we want our results to be accessible to the electorate, while being accurate and meaningful for the informed citizen.

2.3. Inspirations

This (of which the below figure is a screen shot) is an inspiring visualization/data story made for a campaign. But since, we have to dealt with geographical and temporal data, something like this would be ideal example.



2.4. Visualization

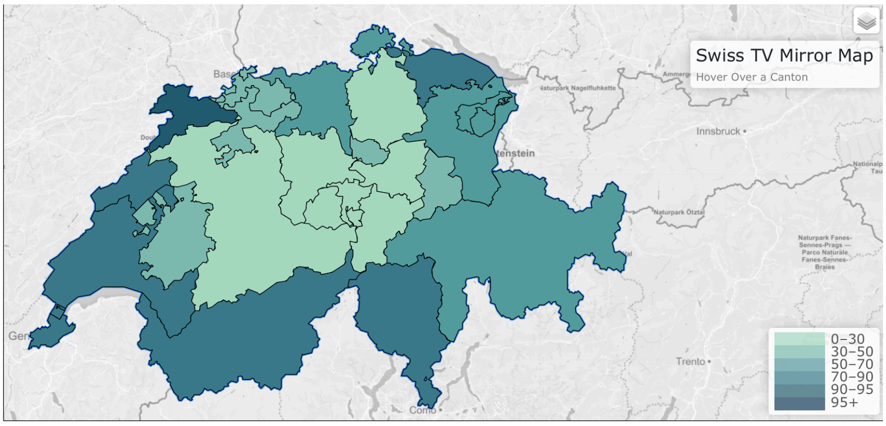
We would like to have an interactive map with a timeline (to keep track of evolution of the map through time) that shows the how broadcasts connect different places together. In order to emphasize the importance of RTS/SRG/SSR in bringing the Swiss together, we have thought of a map where edges (a field map) connect the areas of interest together. (Note that such visualization is only possible for the records that have more than one "geographical theme" (which is a feature in the dataset) associated with them.)

Also, we are planing to demonstrate with which topics that cantons are connected, by displaying the key words related to the topics for each of the connection. By doing that, we also plan to demonstrate which topics are mentioned in broadcasts through years.

3. Dataset

As we mentioned earlier, we are going to mainly use the RTS dataset. Highly convenient API of the RTS dataset can be found here. The page also contains information about how to use the API.

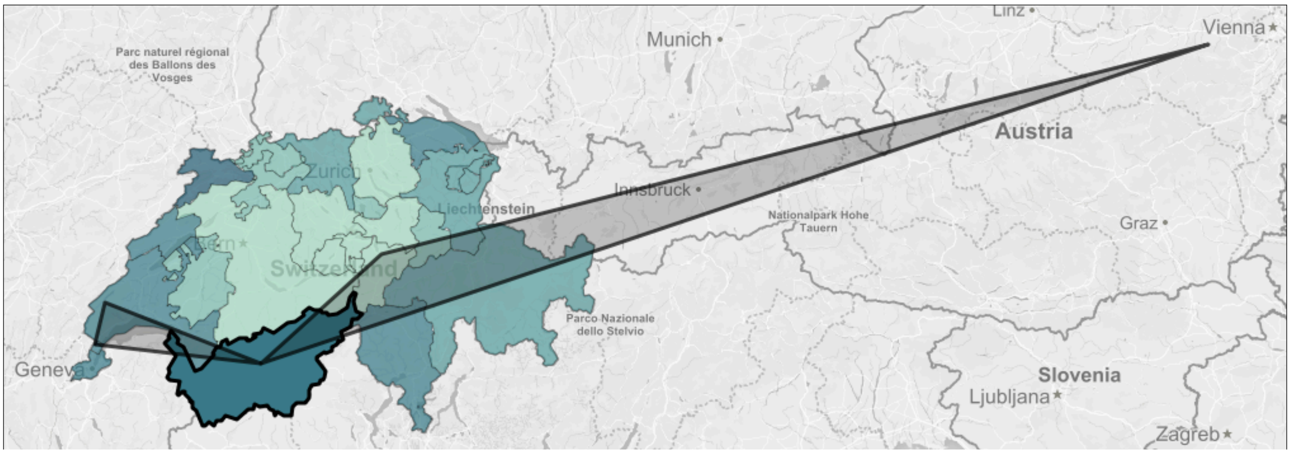
The required data for our project is that cantons related to each of the broadcast (from 1900's to 2000's) and the content (or category) of these broadcasts both of which are provided within RTS dataset. For data accusation we are working with another team (one of our members is also part of this team) from the class Applied Data Analysis. They agreed on providing us the clean data of broadcasts. There might be some missing or ambiguous fields related to broadcasts so, it requires cleaning and filling the missing by data analysis techniques before the data can become useful for our visualization purposes. (Since the Data Visualization class doesn't require us to do these kind of analysis, it is a win-win for both teams.) Collaborating team's Github repository can be found here.



There are couple more steps to mention about data processing. Although the name of the cantons related to each broadcast is provided, we require to retrieve the corresponding geographical locations to use in our map. It is also required to have a .geojson file of to retrieve the border of Switzerland and cantons.

4. Development Process

We've organized the project (and the process book) such that it consist of two major



parts which are map and timeline construction. Data acquisition, cleaning and repurposing was also simultaneously done throughout the project.

4.1 Map Construction

The first step to decide was which tool to use for building the map. We decide to use Leaflet since it is one of the leading open-source JavaScript library for interactive maps (and also the one we've talked about in the class). Creating the initial map with Leaflet was highly convenient. We created a layer and appended to the map along with the coordinates of Switzerland.

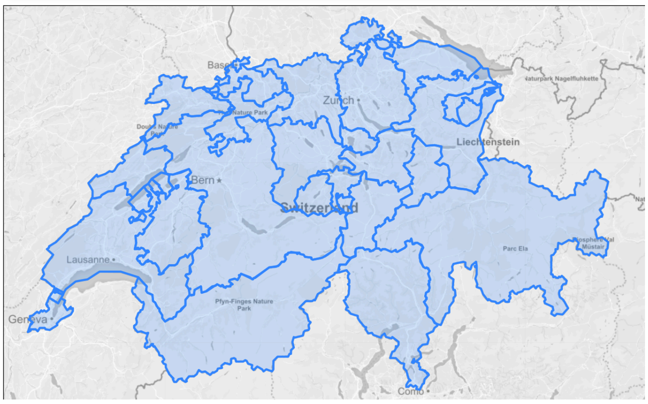
Then first step was to build plain map of Switzerland which only displays the borders

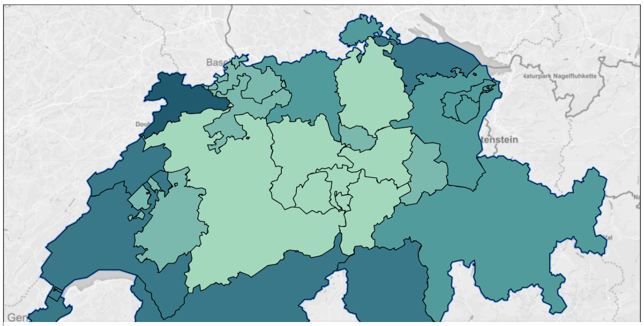
```
var initialLayer =
  L.tileLayer('https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access_token=' + mapboxAccessToken, {
    id: 'mapbox.light'});
var map = L.map('map', {center: [46.818, 8.227], zoom: 8, layers: [initialLayer]});
```

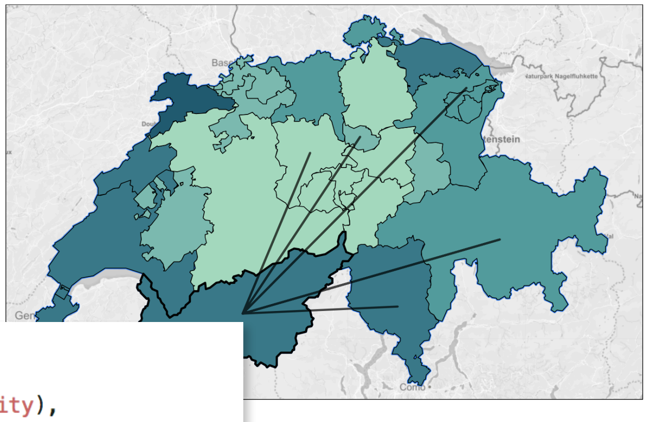
of its cantons as shown in Figure 1. Utilised .geojson data can be found here and the code was fairly simple.

Figure 1: Plain Switzerland Map

```
var geojson = L.geoJson(geojsonData).addTo(map);
```







```
function styleFunction(feature) {
  return {
      fillColor: getColor(feature.properties.density),
      weight: 1,
      opacity: 1,
      color: 'black',
      fillOpacity: 1
  };
}
geojson = L.geoJson(geojsonData, {style: styleFunction}).addTo(map);
```

The next step is to color the map since color schemes might be a good indicator of how well a cantons is connected with others (via broadcasts). In this stage, we didn't have the real data so we created an arbitrary data and color the map accordingly. The resulting map is in Figure 2. We coded a function (called getColor) to map values into corresponding colors. In order to find a suitable color scheme, we used a pretty helpful website called Carto. Then we added the style function to the map.
One of the main property of our visualization will be the lines connecting cantons according to broadcasts. But again, since we didn't have the data yet, we randomly connected each canton to five others. We used PolyLine function of Leaflet to create the lines and added them to the map (as in Figure 3). We though that the most convenient way to show the connection lines is when a canton clicked on.

Figure 2: Colored Switzerland Map

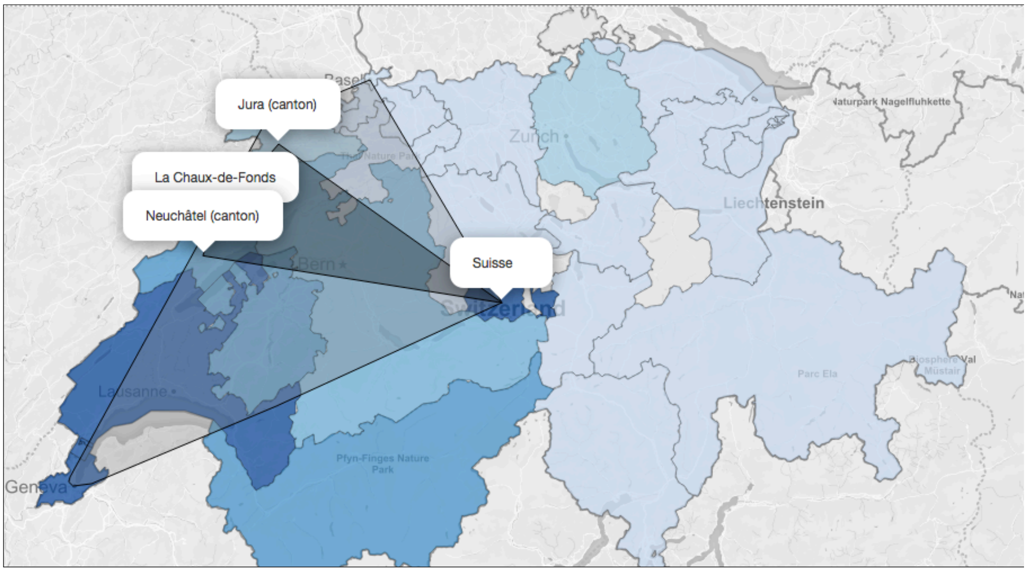
Figure 3: Lines Connecting Cantons

We are planing to serve some information (e.g. excerpt) about broadcasts that connect cantons. Our plan is to display this information inside a separate box when a cantons is clicked but at this point we are not yet decided how exactly we do that. As a prototype, we created an information box on top-right of the map for information display purposes. In addition to that, we made a legend for color mapping. The initial prototype of the map resulted as demonstrated in Figure 4.
Figure 4: Initial Version of the Map Prototype

As mentioned earlier, in order to emphasize the importance of RTS on connecting Swiss together, we want to draw a map where edges (a field map) connect mentioned areas together for each broadcast. So far, we've only created a map with basic edges. As the next step, we want extend this edges and create polygons that connects multiple cantons together. First, we've created a .json data (from RTS dataset) that maps locations to their coordinates in the map. We created a listener so that when a canton is clicked, some other random locations are randomly chosen then, these location were connecting together using polygons. At this point we still don't have the real data so we just try to simulate real connections.

After this stage, we got the real data from ADA group so, we now have actual data of broadcasts and which locations are mentioned within each broadcast. So now, we were able to draw actual connections between locations and we tried to do that for some broadcasts. The resulting map is demonstrated in Figure 5. Note that a broadcast in RTS dataset can be related to not only cantons of Switzerland but any other place on earth and therefore, our initial polygon drawings are not restricted to Switzerland. Since the purpose of our visualization is to illustrate the connectedness of Switzerland via RTS, the next step is to prune the dataset so that it only contains broadcast within Swiss cantons.

Figure 5: Polygons Showing Connection between Places



As it is seen from the map, we only demonstrated 2 broadcasts that canton Valais mentioned in it since we didn't prune the dataset at this point. The obvious next step was to prune the dataset so we did that. Now, we wanted to visualize all the broadcast related to each canton but since we didn't have timeline slider yet, we decided to use broadcasts data only from one specific year. Map with these changes is displayed in Figure 6.

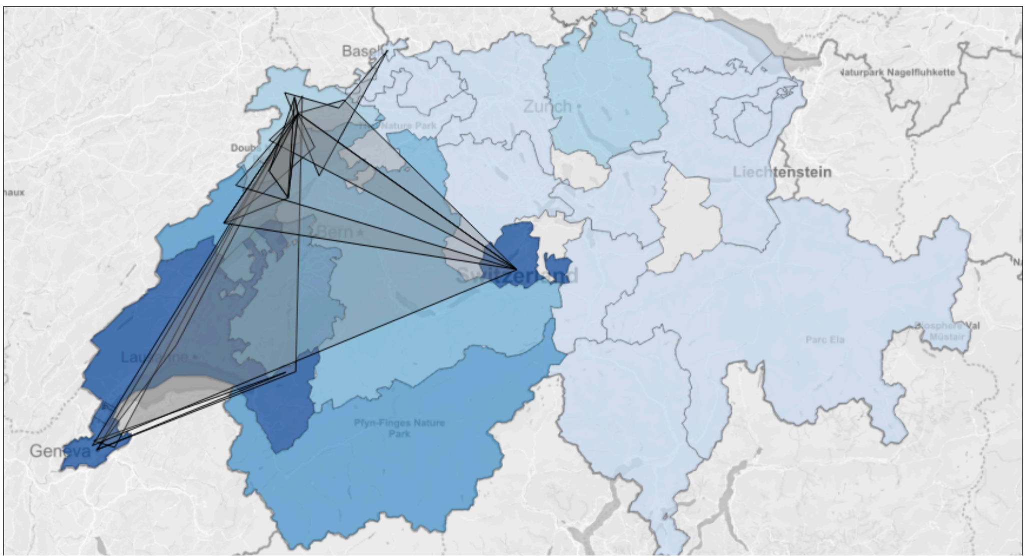


Figure 6: Polygons showing all the connections between Jura and all other cantons for year 2006.

It is worth mentioning that at this point, we started using real data to colour the map, the more a canton have related connections the darker it gets. And if a canton has no connections, we made it naturally, un-clickable. About the polygon visualization, there were two very clear problems. These polygons were too mixed with each other, it was not possible to extract meaning out of them. And the other problem was the corner points. It was not clear where exactly they were at, especially for the ones who don't now Switzerland very well. The first solution that we though of to solve the confusion with polygons was to draw a big concave hull instead of polygons. That way, it would be possible to have a better understanding of the overall area that are connected with broadcasts. About the implementation, we've used a function called ConcaveHull that works basically like standard Polygon function of Leaflet, the code can be found here. About the second problem, we decided to display actual names of mentioned locations. To do that we decided to use Popup function since names will be popup when canton is clicked. The resulting map can be seen in Figure 7.

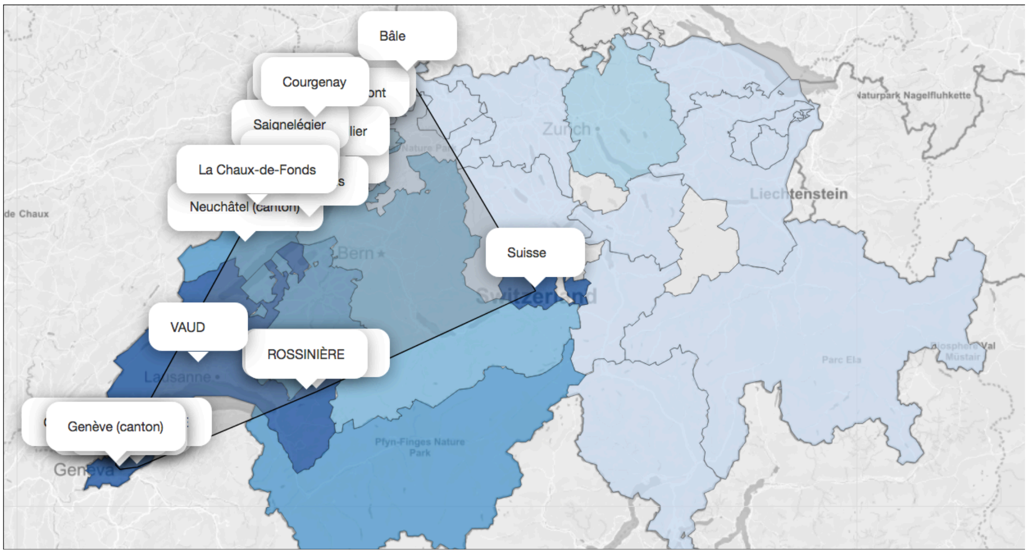
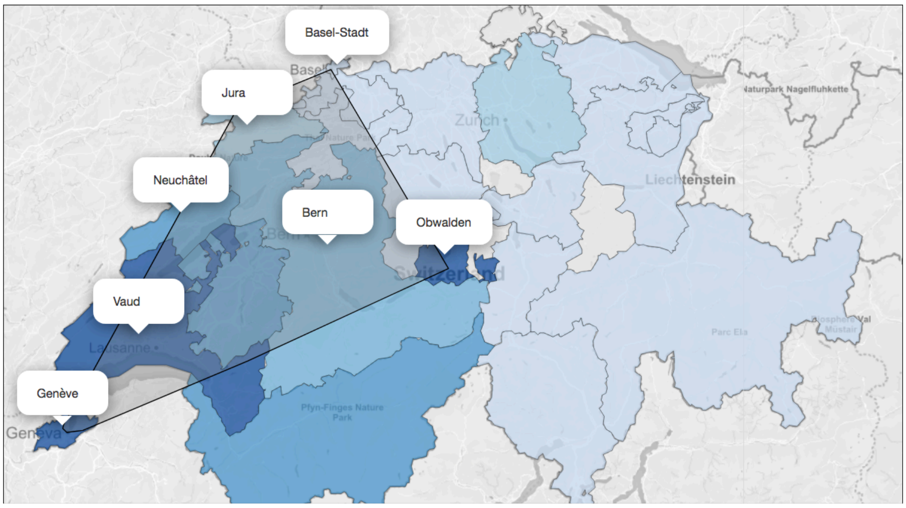


Figure 7: Concave hull representation of connections with actual location names

Although it was more clear that potential area that the broadcasts were effective of, displaying location names obviously was something we need to change.They were on top of each other and some location were mentioned in couple broadcast and displaying couple of times. As a solution to that, we decided to display only corresponding canton names (Figure 8). We though that doing os would be informative enough to have a general understanding of locations that are influenced by broadcasts. For mapping each location to its canton name we used shapely library in python. With this library given the geographical coordinates of a point we can check if it is inside a polygon (in this case a canton) or not. We have the coordinates of Switzerland canton boarders from .geojson file of Swiss cantons.

Figure 8: Connection displayed by canton names

By drawing Concave Hull instead of polygons and displaying canton names instead of actual locations, we were able to obtain more clean and descriptive map that gives the overall idea about which areas were influenced by broadcasts in general. By doing so, we basically compromise individual informations about each broadcasts. We needed to



find a way to display the overall picture without giving up informations related to individual broadcasts. The solution that we found was to display the concave hull with cantons names when a canton is clicked and along with that, show the list of actual broadcasts in a separate menu in a way that they can be clicked. And, when they are the polygon that connects the locations mentioned in this broadcast are drawn on the map with actual location names. The resulting map is in Figure 9. As a small technical details, each of these (concave hull, polygons, names and map itself) is implemented as separate layers to be able to attach and detach them freely. This was especially useful while experimenting on the map to improve it.

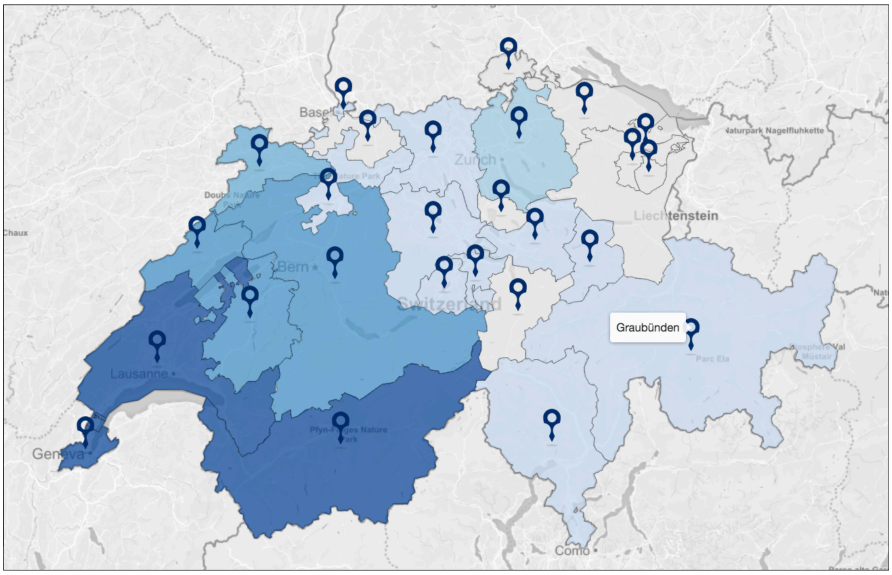
Figure 9: Polygon related to an individual broadcast along with concave hull that shows the are of influence for all broadcasts for a clicked canton

At this point, we were sceptical about displaying names with popups since it is often the case that they fit into one another. They occupy more place than required and shadow the actual map. Also, they lack aesthetic appeal. Consequently, and with certain feedbacks, we decided to change these display box. To overcome mentioned shortcomings, we brainstormed together and come up with couple of ideas. First, we though that displaying names only on hover is much better in terms of visibility of the actual map. In addition to that, we shrink the display box by also removing the unnecessary white space around the text. The resulting map has become much more elegant and can be seen in the following figure 10.1.

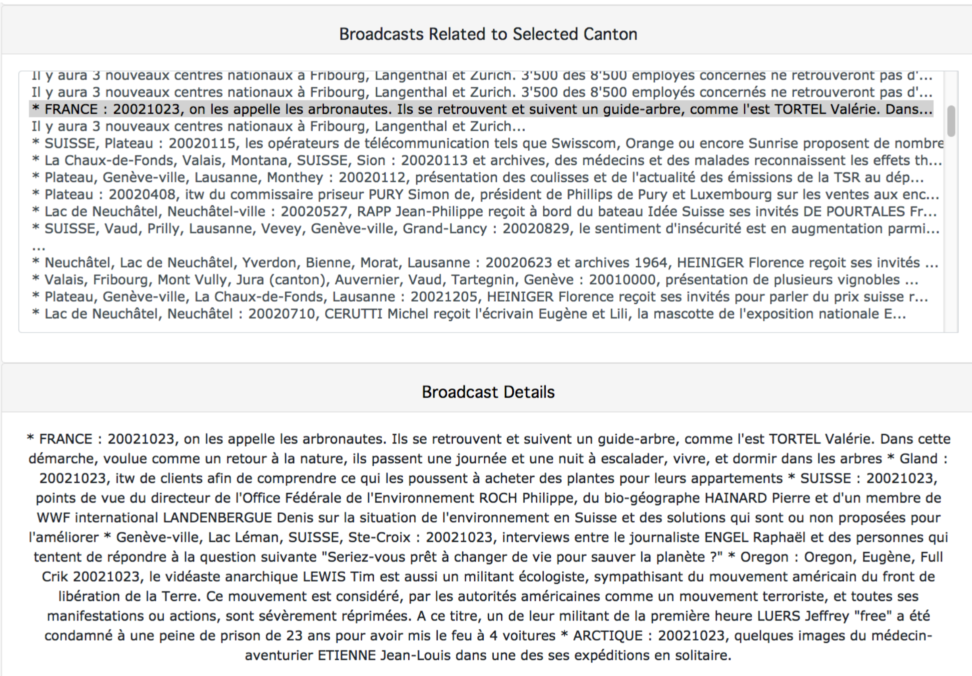
By displaying location names in this way, we reduce the instant visibility of cantons especially, for those who does not know Switzerland very well. To make up for this drawback we decided to give users an option to display canton markers. By doing so, users can check cantons names whenever they want by just hovering over markers as it is seen on Figure 10.2. We've done that by using layer control capability of Leaflet. Showing and hiding these marker is as simple as checking a box.

Figure 10.1: Location names are displayed as minimalistic boxes and only on hover.
Figure 10.2: Optional markers to show canton names

We mentioned that we wanted to serve some information related to broadcasts and since we already in this stage where individual broadcasts can be selected, we decided to create a text area to show excerpt of the selected broadcast (Screenshot 1). For



extracting excerpts related to a canton we make use of the unique broadcast IDs. In the RTS data each broadcast is associated with an ID. So when a canton is selected we can extract IDs related to that canton in that year and show it in the broadcast pane. We choose to design both the list where broadcasts can be chosen and the



information field as foldable so that they can be closed when not in use. Since use purposes of our visualization can vary from user to user so, we though that modular design would be much more convenient. Each user can see what they are interested in.

Screenshot 1: Foldable menus for broadcast and details of them

Along with these menus, we also wanted to give users a way of filtering broadcasts. In RTS data, most of the broadcast is associated with a genre and we though users might just, be interested in certain genres. For this reason, we decided to design a genre filter. RTS dataset is not standardized much and contains about 100 genres for this reason we required to find a convenient way of choosing concerned genres. We come up with the idea of using fuzzy search where it is possible to relate a typed search argument with existing genres even when the argument does not exactly corresponds to them. We devised the filtering in a way that when a specific genre is chosen, the map itself is updated along with all of the corresponding broadcast informations. Thanks to that, it is even possible to use filter when canton is clicked and corresponding polygons are drawn. An example of a filtered map can be seen in Figure 10.3. Basically, these polygons are re-drawn in case a filter is applied. One important issue worth mentioning is that it might be the case where there is no broadcast present for chosen filter. In such situation, we decided to guide user (with popups) to go to different year or change the filter itself. Note that, it was not wise to filter according to two genres since, broadcasts with more than one genres are really rare in RTS dataset. Filtering with more than one genres leaves the map empty for almost all of the years.

It is also, important to mention that along with genres, RTS contains themes associated with broadcasts. However, as emphasized, RTS dataset is in need of improvements since like genres, themes are also not standardized. There are almost 2000 different times in the dataset and not viable for grouping broadcasts together. That is the reason, we did not filtered according to themes. Although we consider using some kind of machine learning algorithm to cluster similar themes together, our lack of information (related to these algorithms) at this point, prevented us from doing so.

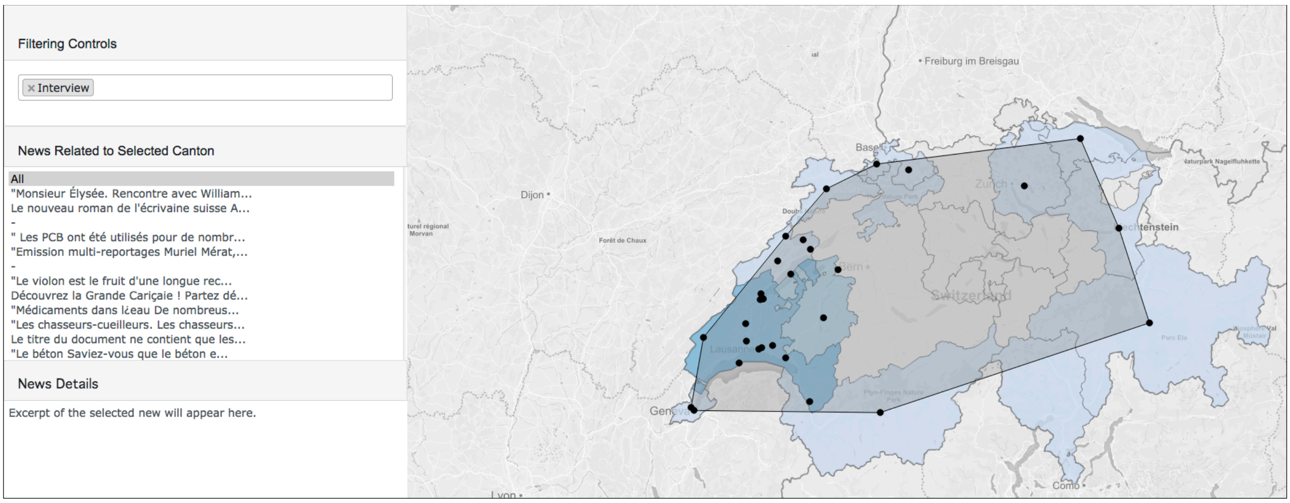
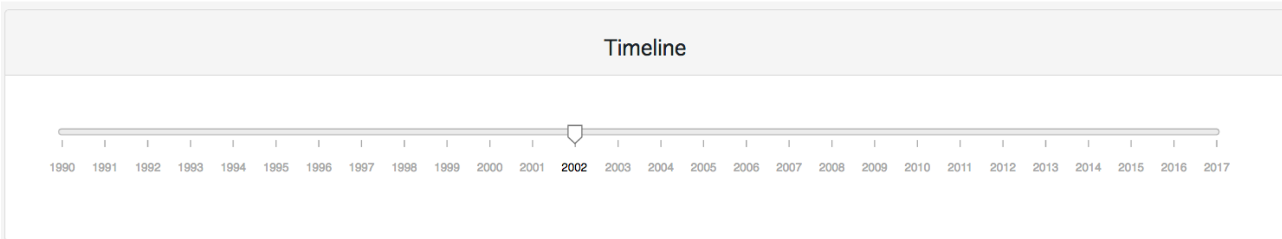


Figure 10.3: Filtered broadcasts according to genre "Interview"

4.2 Timeline Construction

Until this state we've only worked one year worth of data but since we are trying to show how RTS connects locations and people together throughout years, a timeline slider was a must-have. So, the next step was to build a timeline slider. We started with a simple slider called d3.horizontalSlider and made it to change text of the heading according. Detailed information and the code of this slider can be found here. It was simple as it is seen in Screenshot 2.



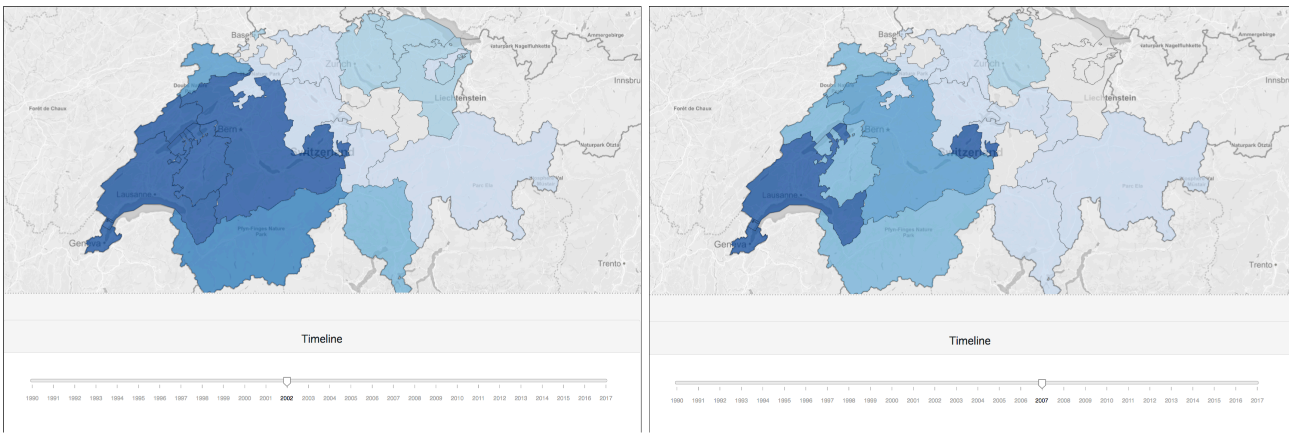
Screenshot 2: Basic Timeline Slider

Note that it is also designed as foldable field since a user might want to work on a single year and won't need a slider. Next, we actually utilized the slider so that connections (concave hulls and polygons) were drawn according to year. We basically defined a variable called YEAR and made this slider change this variable. According to this variable data related to broadcasts were retrieved. The code to get broadcast for the clicked canton was like the following.

```
var canton_name = e.target.feature.properties.name;
var broadcasts = cantonConnections[canton_name][YEAR]
```

We follow the same logic to color the map. This time the layer were re-drawn each time value of the slider changed. As mentioned earlier, the color tone of a canton is determined according to number of connections it has on a specific year. We showed the change of the choropleth map in Figure 10.

Figure 10: The change of choropleth map from 2002 (left) to 2007 (right)



Timeline was highly responsive and convenient to use, especially for those who interested in change in the connectedness of a canton. However, the actual information about where each of the broadcasts happen were not present in this map. Again, we were generalizing (mapping each location that a broadcast mention to the corresponding canton) to give an overall picture of the connection patterns. As we did when drawing super-edges, we wanted find a way to visualize individual location that a broadcast mentioned without loosing the big picture. The solution we found was to put a dot for actual coordinates of locations mentioned in broadcast along with choropleth itself. In this way, it is easy to see where is the exact places that broadcasts mentioned and how often an area is mentioned (by looking the frequency of dots in near a specific area). Placing these dots were highly simple with `Marker` object of Leaflet. The resulting map is in Figure 11.

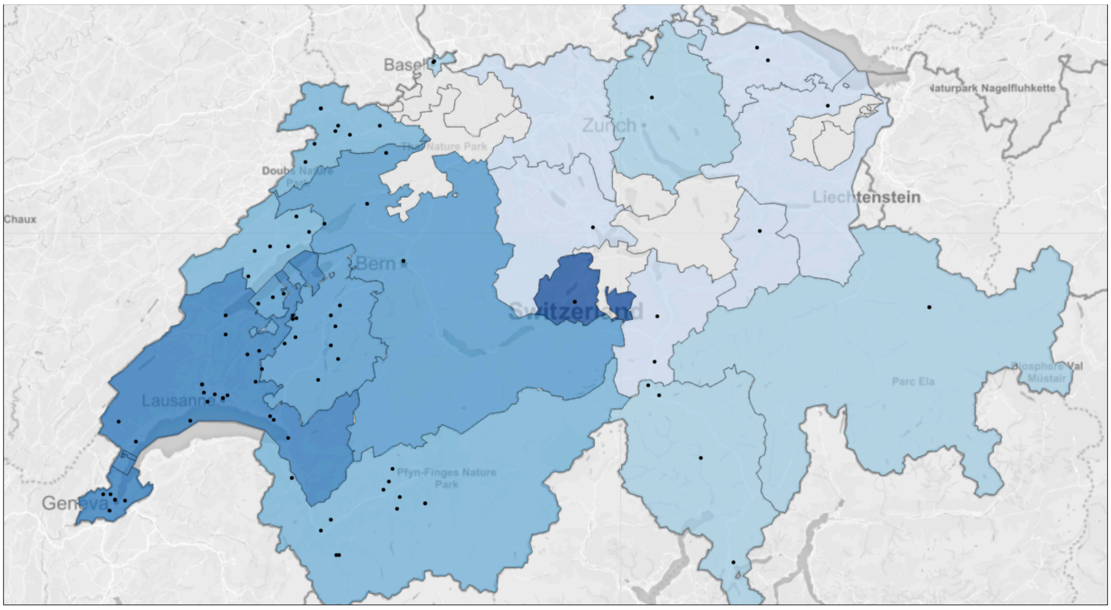


Figure 11: Dots are drawn for each location mentioned in broadcasts on a year

We choose to put small dots by following our logic that our visualization can be used for different purposes. For the ones who are not interested exact location, the dots should be bothering so, we set their sizes just visible enough for who is interested in.

Further thinking and feedbacks made us change our minds about location dots. We decided to increase their size and and make them clickable. We though that showing the corresponding polygons when a particular dot is clicked is a natural way of utilizing these dots. Polygons (and concave hulls) are only drawn when corresponding dots are clicked. This way, user would be able to see the exact location appeared in broadcasts and very conveniently go to broadcasts that they want to examine juts by clicking a dot. Note that, according to clicked dots, corresponding broadcasts are also, highlighted and details are shown on the left menu.

Although, showing a dot for each location mentioned in broadcasts and displaying a polygon between locations that are mentioned together, is an effective way of guiding users. Practically, it didn't work in our case since many dots appear on top of each other and block themselves. In order to prevent this cluttering, we decided to show corresponding dots when a particular canton is clicked. This way, users can still exact location appeared in broadcasts without overlapping. As for deciding the size of individual dots, we experimented with different values and came into the conclusion that the size should be big enough for clicking and small enough to not the intervene with other dots. Consequently, we choose somewhere in the between as the size of these dots. The resulting map after applying all of these ideas can be seen in to following figure.

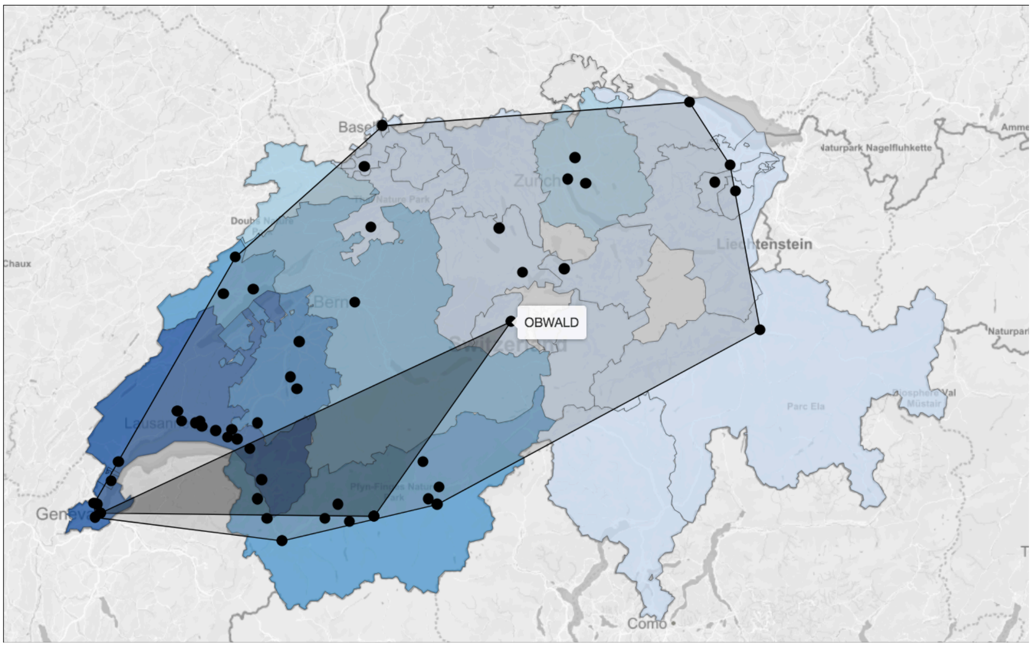
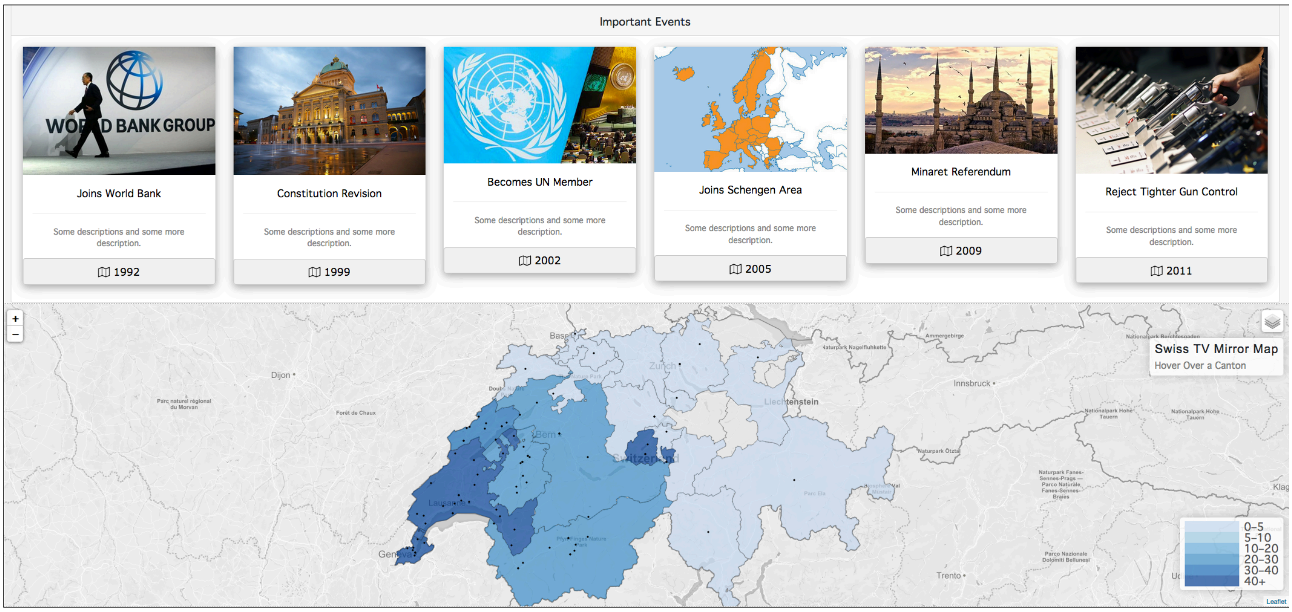


Figure 11.1: Map with bigger and clickable dots corresponding to mentioned locations in broadcasts related to clicked canton within the year selected in timeline slider

The next thought that comes to our mind was to give users a convenient way of exploring the map in the periods of important event (e.g. when the right to vote given to woman). It is especially, useful to see how RTS brings the nation together during these important events. We decided to design a foldable area where each event is represented by a clickable card and when it is clicked map goes to corresponding year. We made this happen by changing the timeline slider and since, the map is connected to timeline slider, it changes automatically. The resulting design is shown in Screenshot 3.

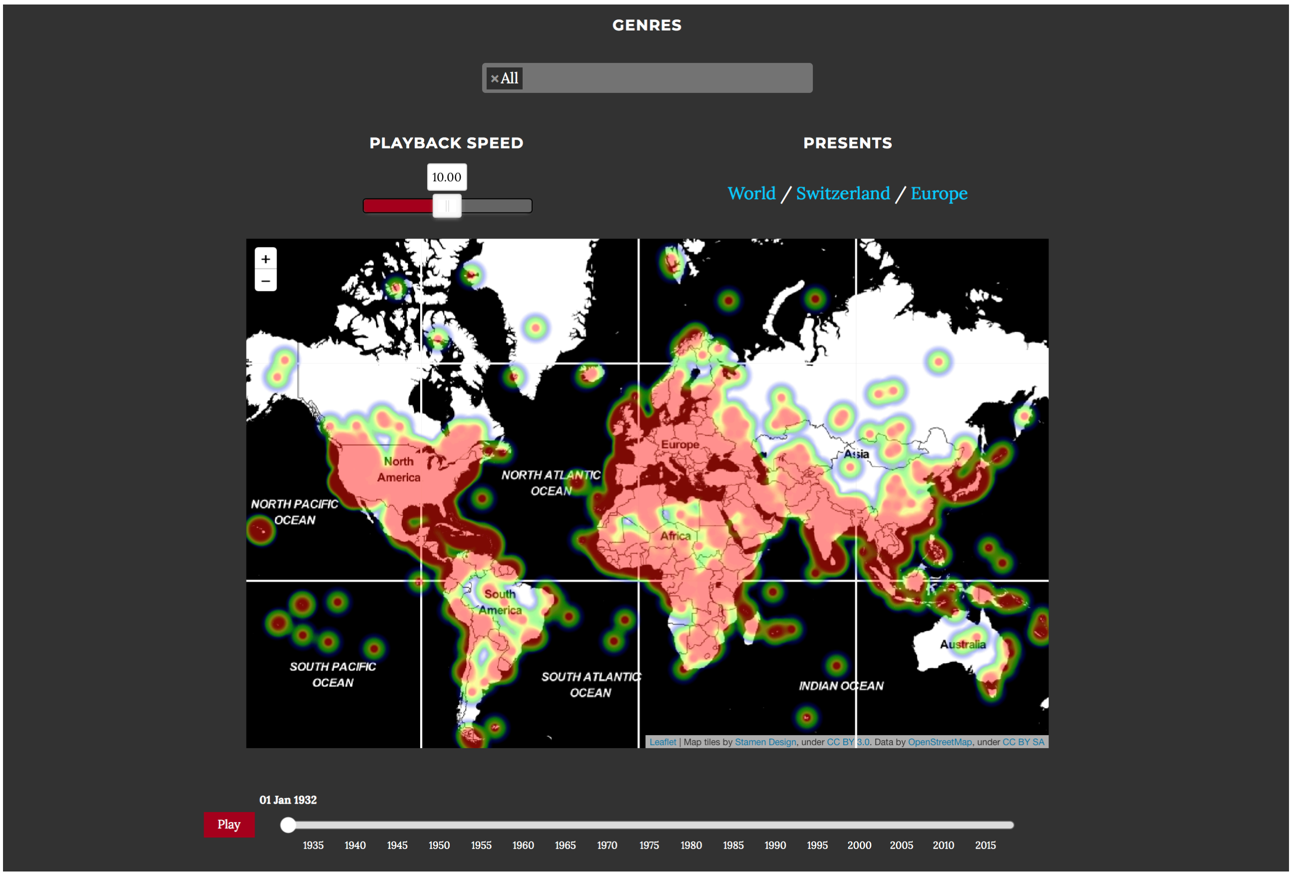


Screenshot 3: Important event cards that give means of easily accessing corresponding dates

One important point worth mentioning about design is that the main tool that we choose to use in order to have more stylish, easy to use elements (e.g. important events cards, foldable areas) was the Bootstrap which is an open source toolkit for developing website.

5. Heatmap and News Cooccurrences

As mentioned before, this project was carried out in conjunction with a EPFL Applied Data Analysis Course project group (Leo Wirz, Sina Fakheri and Mohammad Yaghini). That project led to this data story:



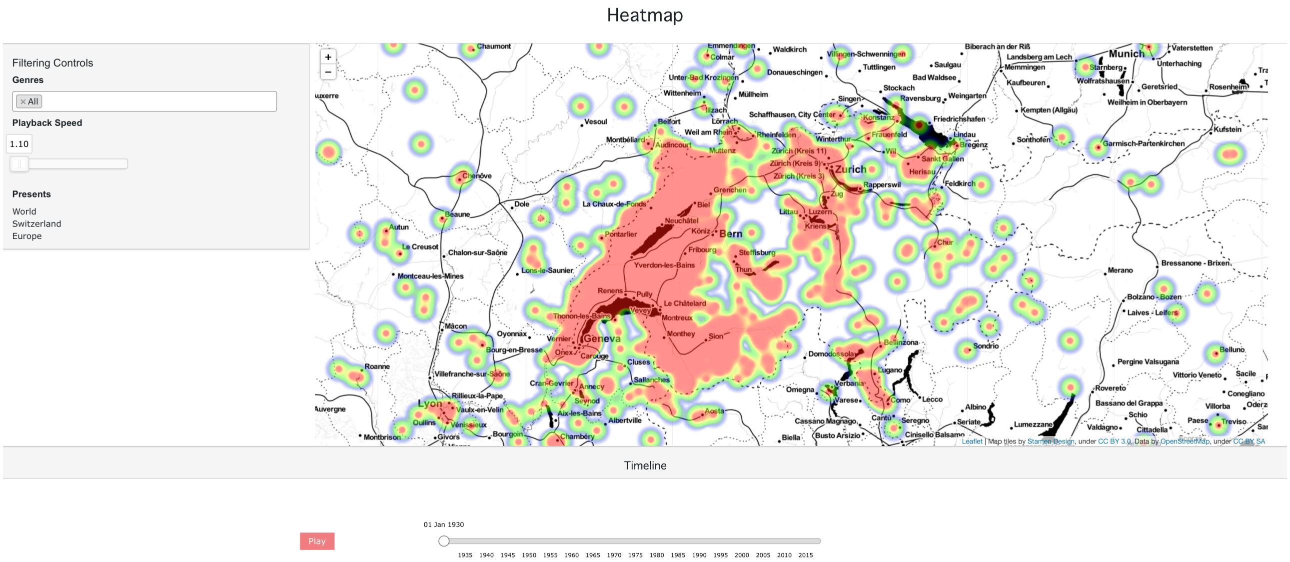
The above figure shows the frequency of news ("heat") in different locations, together with filtering abilities on the genre and an animation feature that allows the user to appreciate the temporal features of the data as well as its spacial ones. To this end, playback controls are provided as well as a set of "presents" (pre-set configurations) to facilitate the navigation of the visualization.

The interesting aspect about this visualization is that since each news contains *a set* of related locations, in the visualization (and especially while playing the animation), the locations that appear together light up at the same time. This naturally gave us the idea of showing the connections between different locations in Switzerland by reusing the same tool.

5.1 Changes and Adaptations

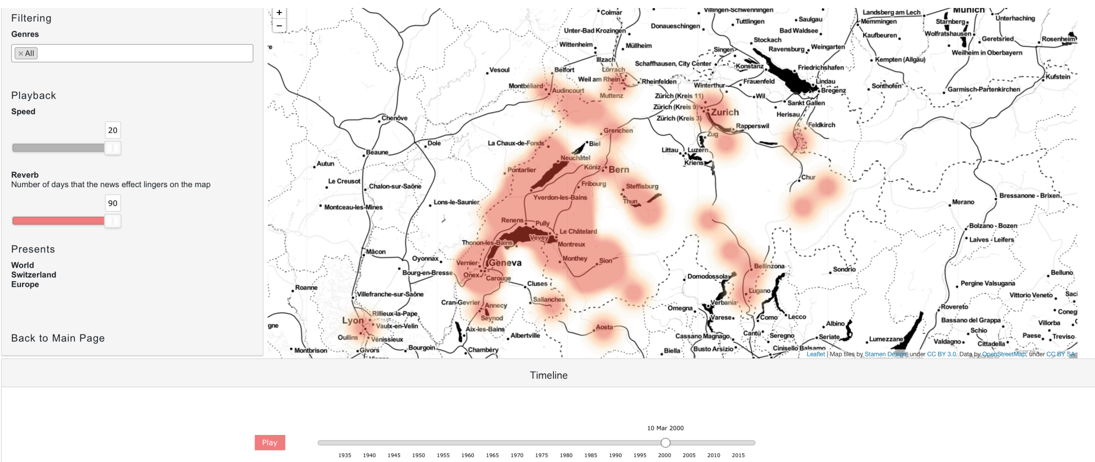
Our initial attempt was this:

The feedback that we got form the professor said that the original coloring scheme (the “JET”) is problematic because of its inconsistent luminance. Indeed, since the only thing that we measure is the frequency of the news, having such a gradient of colors did not seem correct. So using the *colorbrewer2* we selected a single hue 3 class coloring scheme for the heatmap, which is implemented by heatmap.js (a leaflet plugin).



The next issue that we faced was that — especially when filters were applied — the animations were not smooth (the points quickly appeared and disappeared). For this we needed to have a buffering window that showed the news with some lingering effect, a reverb.

Adding these features brought us to our final design:



6. Conclusion

By developing this visualization, we mainly aimed to show how Switzerland is connected via broadcast throughout years. Also, we tried to make the data related to all the broadcasts from 1900s until today for anyone who is curious about them. We gave means of exploring Switzerland's history and present through broadcast. And once again, we wanted show the importance of RTS and the work that has been done under it.