Sacramento Crime Visualisation

## Where to find:

The whole visualization project including all source code etc. is located on a *GitHub* repo.

## Dataset:

* Origin: [*https://github.com/pkamin/sacramentocrime/blob/master/SacramentocrimeJanuary2006.csv*](https://github.com/pkamin/sacramentocrime/blob/master/SacramentocrimeJanuary2006.csv)
* Name : Sacramento Crime
* Nr. Items: 7585
* Nr. Attributes: 9
* Size: 0.78mb

## About:

This dataset shows crime statistics from January 2006 in the city of Sacramento USA.

## Tasks:

1. Find out the most dangerous areas around the city of Sacramento.
   1. To get an overview of the dangerous areas in the city, the distribution of all crimes in the dataset should be visualized on a map background.
   2. Distribution of committed crimes should be shown subject to a certain district.
2. Does a correlation exist between the time crimes were committed and the district they were in? Is one district more dangerous than another?
   1. **Addition**: This task has changed a little bit in matters of our first handed in task description. The huge amount of different attribute (crimedescr) values, 300+, caused some troubles and made the visualization mostly confusing. So we decided to change the plan a little and see how the relation between certain districts and a specific time is. The district attribute has only 6 different values and was a little bit easier to handle and visualize.
   2. For the visualization of this task a Stacked Bar Chart visualization should be used to get a quick overview of the different districts and the time the crimes were commited.
   3. Certain values (number of crimes) should be shown for each part of a bar and time.

## Source:

The source can be found at github at <https://github.com/anotherrandomname/sacramento-crime-visualization>

### Approach:

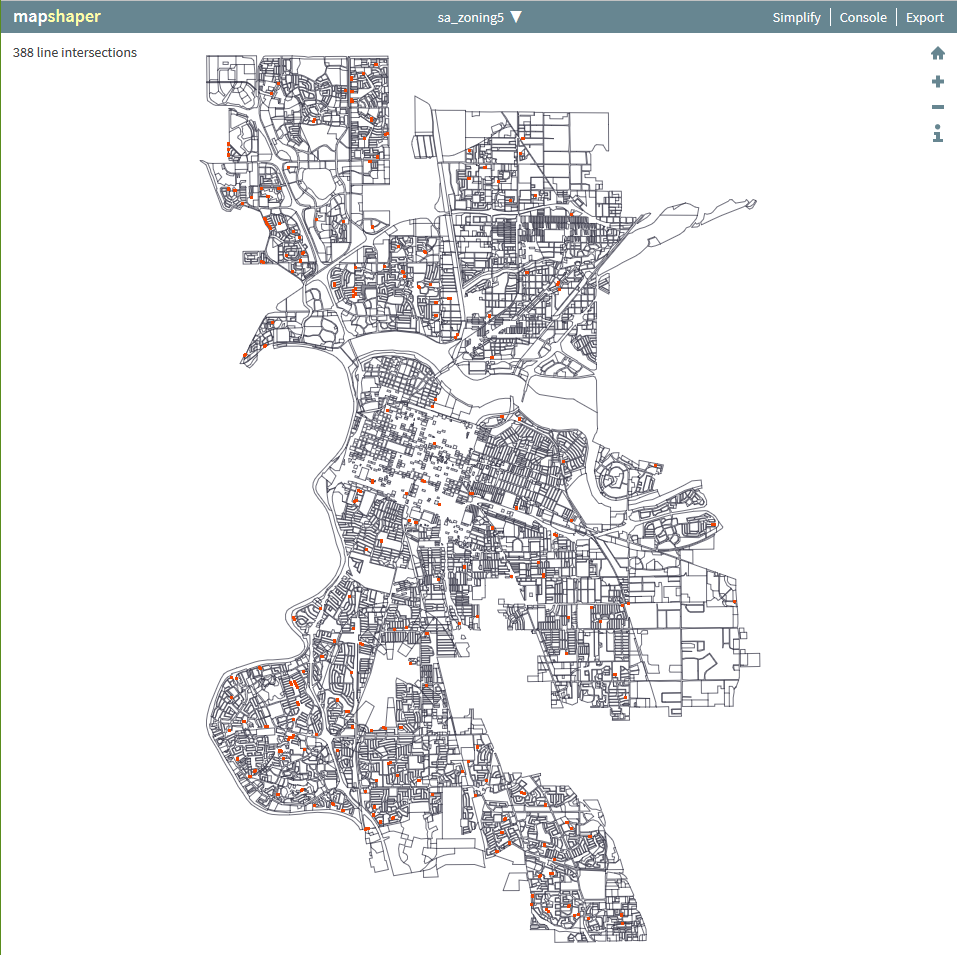
#### Task 1 Map:

The map visualization was very tricky, mostly because it was difficult to find proper working map-data from Sacramento.

To understand how to get the appropriate data and how to use it with D3, we read a lot of different tutorials. The most useful was “Let’s make a map” (<https://bost.ocks.org/mike/map/>), even it is a bit outdated because it is using D3v3 and old versions of topojson and so on. But recently Mike Bostock published a new tutorial, using new versions of the technologies. So we ended up using shp2json to convert map-data from us-governments to geoJSON. This worked perfectly with an US-Map and the California map, which can be found in the tutorial, but it won’t work with the shp-files we found for Sacramento.

We tried several maps provided by the city of Sacramento (<https://www.cityofsacramento.org/GIS/Data>). For example the zoning file would fit very good for our purpose, but we also tried the city boundary file and several others.

When we converted the zoning file to geoJSON and tried it out in mapshaper.org, it looked very promising:

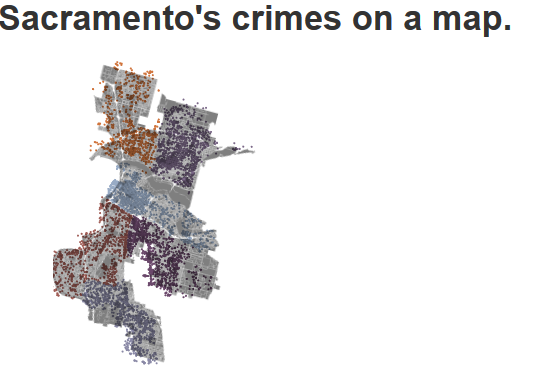


But it just could not be rendered properly in D3, using several different projections. We only got results like this:



We also tried to export several different formats in mapshaper, with no results. When we looked at the generated json-file and we found coordinate files like this: 6840262.589940444,1948386.9988849312. We think that’s the reason why it isn’t working, but we are sure you have a solution to this. Maybe it is only a beginner fault.

After a lot of hours of trying, we converted the zoning-file to an svg and used it directly as background for the mapping of our dataset.

The final result looks like this (with and without svg as background):



The projection is done with the following code snippets:

var w = 800;

var h = 400;

var projection = d3.geoAlbers()

.translate([w \* 25, h \* 6.88])

.scale([60000]);

var path = d3.geoPath()

.projection(projection);

var svgMap = d3.select('#svg1HereMap')

.append("svg")

.attr("width", w)

.attr("height", h);

The svg-image is added as following:

d3.select("#svg1HereMap").append("image")

.attr("xlink:href", "data/sa\_zoning.svg")

.attr("width", 250)

.attr("height", 250)

.attr("opacity", 0.5)

.attr("id", "svg-image");

Finally the crimes are printed on the map as followed:

const rect\_enter = rect.enter().append("circle", "rect")

.attr("r", 1)

.attr("fill", function(d, i){

return z(d.district);

})

.attr("transform", function(d) {

return "translate(" + projection([

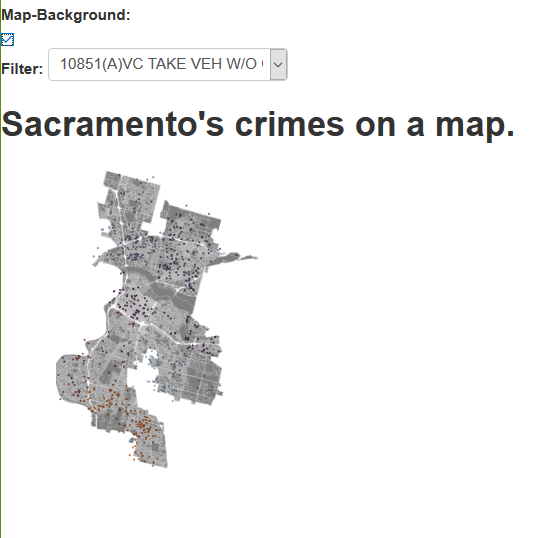
d.longitude,

d.latitude

]) + ")";

});

The different crimes in the different districts are mapped with different colors, according to the stacked bar chart. The crimes on the map and on the bar chart could also be filtered with a dropdown as you can see in the following screenshot. The map can be shown by clicking on the checkbox:



#### Task 2 Bars:

So at first the dataset is imported via *d3.csv()* import and during that a few datatypes of attributes like, cdatetime, district, latitude, etc. are set correctly. The datetime needed also a little bit of *d3.timeParse()*, that means we just kept the time values not the date. In order to get the data group by the time and also by the district for a better usage afterwards, we nested the date with *d3.nest()*.

To use the data in a stacked bar chart we had to remap the structure of the given data a little bit, because we had some troubles to get this running with our data specifically with the structure of our data. The remapping also includes the *d3.stack()* operation of the collection for the usage afterwards. Also we changed from our first attempt to show all the different crime descriptions in this visualization to a different approach with the districts which is also very useful.  
  
Next thing was to create the layers for each district a collection of single bars with the correct color respectively and let d3 stack the correctly. These bars are also always showing a tooltip with the appropriate value for better understanding, when hovering over with the mouse. Also the corresponding entry on the legend is highlighted. Other way round the corresponding layer for each time value is highlighted when moving the mouse cursor over an entry of the legend.

Interactions:

* Higlight BarChart -> Legend
* Legend -> BarChart
* Barchart tooltips

Pics:

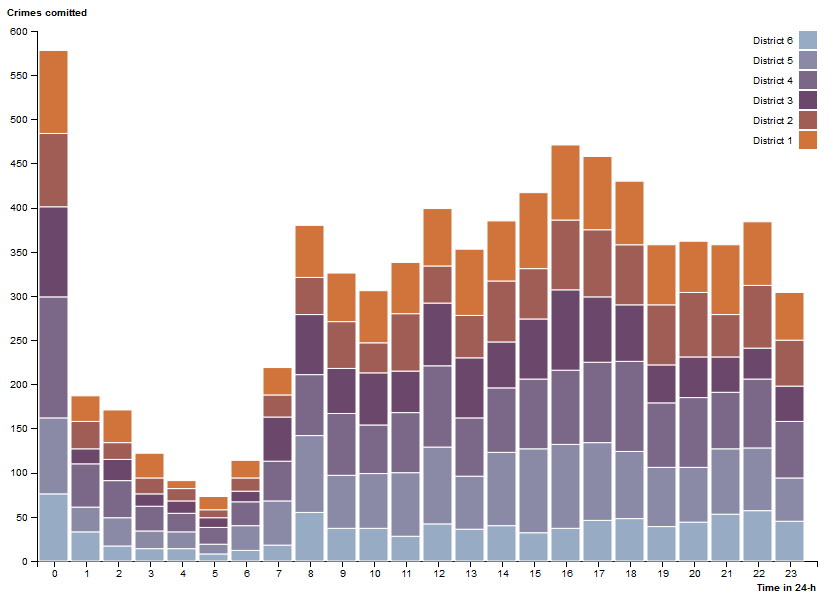


Abbildung 1: Stacked Bar Chart crimes/district/time

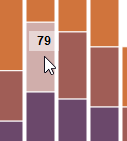


Abbildung 2: Bar tooltip

#### Cross Interactions

Interactions between the bar chart and the map:

You can filter the crimes shown on the map

* by hour, by hovering over a bar of the barchart
* by district, by hovering over the district legend
* by choosing a crime in the dropdown

You can filter the crimes shown in the barchart

* by choosing a crime in the dropdown

You can also apply multiple filters.