Project report (Group 9)

Rik de Graaff and Travis Rivera Petit

University of Basel
Databases lecture (cs244)
Autumn Semester 2019

1 Introduction

The rise of social media has given people from all different backgrounds a plattform with which to share their thoughts and feelings online leaving them avaiable for anyone to read. This phenomenon gives society yet another tool for analysing people's mass behavior; after all, if many individuals share similar posts online, it may be a sign that something is going on. This becomes particualrly interesting when important events that affect millons of people take palce.

After the last US elections some were hearly displeased and others were satisfied, this friction has been pronounced by the fact that the controvesial figure Donald Trump has taken over the presiency of the United States since 2016. In fact many opposers of Trump did not stop at Twitter or Reddit, they took it up to themselves to go out and protest on the streets.

One must not forget how much impact such events can have over the mental well-being of people.

Our goal with this project is to answer the question "To what extent have the results of the US 2016 presidential elections and its consequences affected the suicide rates in the United States?" To do so we use information about protests, riots, and related events together with the outcomes of the elections and reports regarding suicide rates all neatly stored in a database to make a county-level investigation regarding the question at hand.

2 Source material

The choice of sources is paramount when it comes to doing projects such as this one. Our datasets are:

1. **A US county level presidatial resul for the years 2012 to 2016** (add citaton) The reason for which we use the outcomes of the 2012 and 2016 elections instead of just the latter is that eventrs from 2012 onwards are used as a control group, i.e. we ...
2. **The GDELT2 events dataset** GDELT, which stands for Global Database of Events, Language and Tone is a database which monitors and manages political episodes gobally. It is one of the largest open souce indexes of global

society citaton and it keeps track of ocurrances such as diplomatic summits, wars, news, protests et cetera as well as the relationships between them.
The GDELT 2 dataset is consists of three parts: an event dataset, a mentions dataset and a global knowledge graph. For this project work with the events dataset which is itself divided into one hundred and sixty three thousand csv files separated by timestamps which cover global events ranging from the year 2015 onwards. Because of the specifications of the project, we decided to work with the eighty nine thousand of them, namely those in the interval starting from January of 2015 to September of 2017, nine months after Donald Trump was declared president of the United States.

3. **Centers for disease and control prevention's suicide rate per US county reports**

All of the sources combined amount to about 85 GB of data.

Note: For the P1 milestone we presented a slightly different version of sources: we sated that we were going to use the GDELT1 (as opposed to the GDELT2) events dataset. This was because we felt that since both of thsese sources have more data than we need, going for GDELT2 would be an overkill, however we realized later on that unlike for GDELT1, the GDELT2 csv files contain a record called ADM2Code, which can be used to link counties and events together (which is called for, details on this are covered on the INTEGRATION section). This change rocketed our file sizes from 23 GiB to more than 80 GiB. Because of this, we decided to drop another source: The Crowd Counting Consortium's protest activity dataset, since the information found thre is redundant now that we have GDELT2. The GDELT data subset we use in this project consists of sixty three thousand csv files, where each file takes somewhere between 800 KB and 1.5 MB of storage.

On our P2 hand-in we include a script (/gdelt/download.py) that downloads each csv file and edits the filenames for further processing. If a file is found to be corrupt, it is not downloaded and the index of the file is saved in a .txt file (/gdelt/bad_indices.txt).

The dataset consists of sixty attributes and the domain of each attribute is in detail on the official handbook citation.

Moving on, the suicide rate dataset can be obtained from the Centers for Disease Control and Prevention website [1].

The election results dataset is straightforward, it can be obtained on todo.

Putting it all together: The election results and the Suicide Rate data sets have the county attribute in common. A county lies in a state, and both a county and a state are spesializations of a "geography". Some other attributes can be removed since they are redundant (e.g. MonthYear makes the Year attribute irrelevant). The end result can be seen on figure 4.

Technical parts On the inegration process, we tried using the counties' FIPS codes present on each of the datasets to link the different entities together. However we quickly learned that each dataset's FIPS code contradicts the FIPS code each other data set.

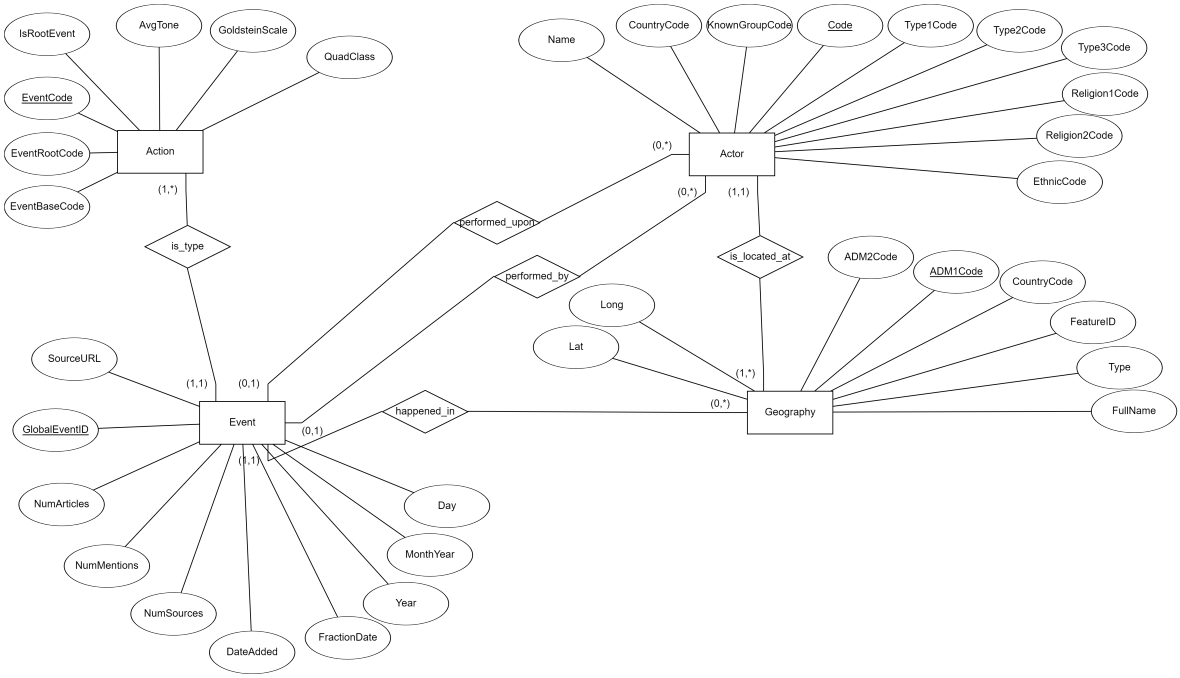

Fig. 1: Our GDELT schema integration

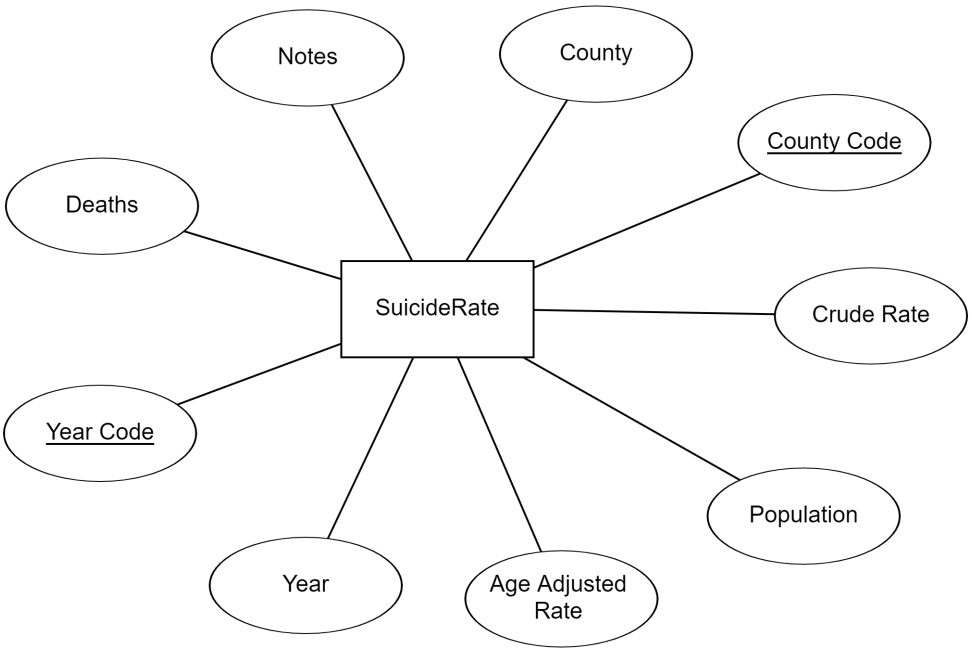

Fig. 2: Our suicide rate schema integration

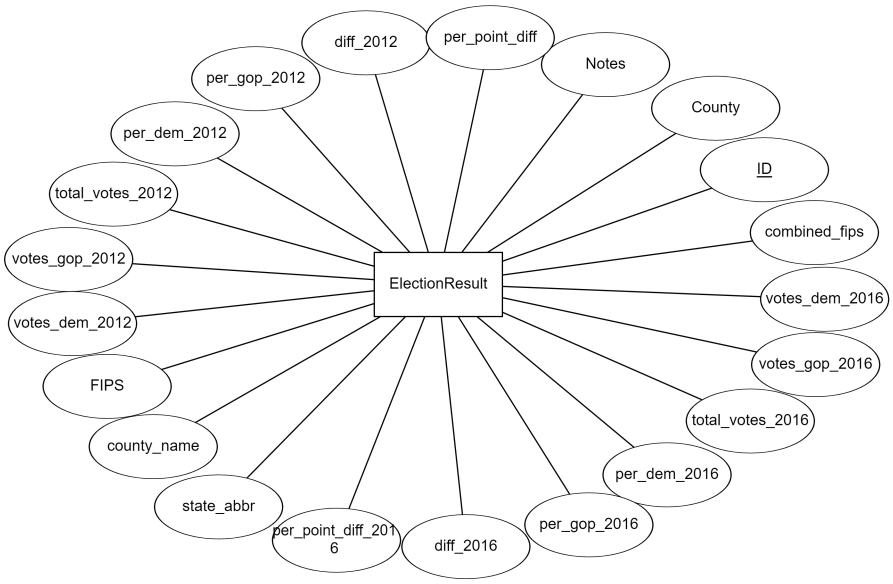

Fig. 3: Our election results integration

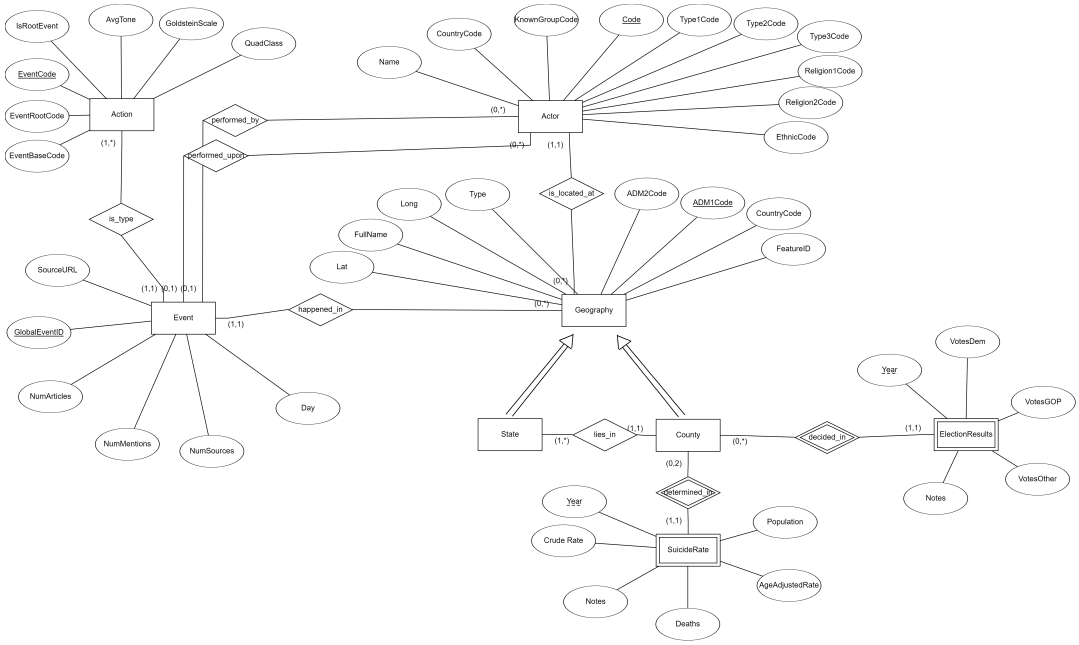

Fig. 4: Schema over the whole database

Our solution has been to identify each county (and state) by a geoID code than can be derived from todo https://community.esri.com/thread/24614 ESRI .

The only other hazard occurs when ElectionResults generates an ID that does not match any County, in that case the corresponding row is simply skipped and also printed to STDOUT for reference.

3 Analysis

The main focus of our project was finding out whether the election of Donald Trump had an impact on suicide rates in the United States of America. To that end, the first diagramm we created was simply a bar chart with the total amount of suicides in the United States of America per year for the years we had data on: 1999 through 2017.

This diagramm can be found in Figure 5. As can be plainly seen, there was a marked uptick in suicides between 2016 and 2017. Since Donald Trump was elected president near the end of 2016 on the 8th of November and assumed the presidency in the beginning of 2017 on the 20th of January, this noticeable jump provides us with some tentative evidence in support of our hypothesis.

However, the diagramm in Figure 6 shows that there was no notable correlation on a county level between how much better Trump performed than Mitt

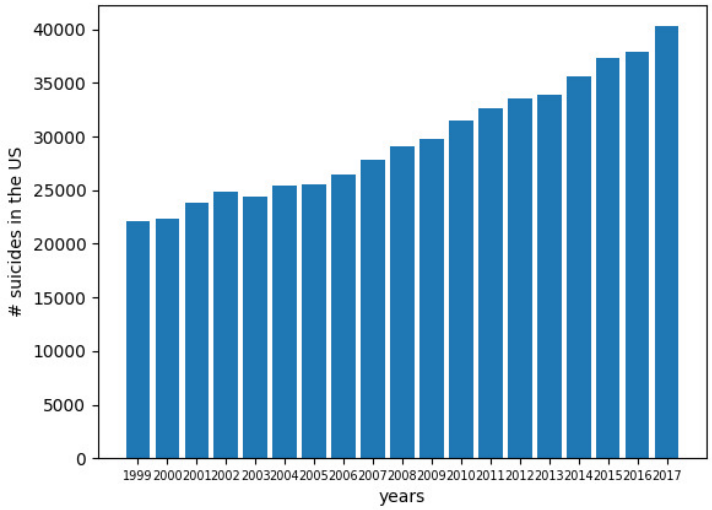

Fig. 5: Total yearly suicides in the US from 1999 to 2017.

Romney did in 2012 and the increase in suicides in said county. Even looking at the counties which flipped during the 2016 election yields no discernable correlation. What can be made out is that, in most counties, Trump did better than Romney did in 2012 and suicide rates rose. These findings are not in the least bit surprising, given that Trump won the election and Romney did not and that the total amount of suicides rose markedly, but it's good to see that our data reflects reality and is consistent.

Figure 7 shows a similar picture, but uses the age adjusted suicide rate in 2016 instead of the increase between 2016 and 2017. In this diagramm we see a noticeable, if slight, correlation. It seems like Trump did unexpectedly well compared to establishment republican candidate Romney in counties with high suicide rates. [2] came to a similar finding when they examined the relation between life expectancy and republican gains in the 2016 election. This is an interesting finding in its own right, but does not support our hypothesis. It is however worth noting, that it in no way explains the rise in suicides between 2016 and 2017.

Figure 8 contains a single frame of an animated scatter plot showing both the change in age adjusted suicide rate and the election results dynamically. The animation did not reveal any unexpected or noteworthy insights into the data. This still image, showing the election results in 2016 plotted against the suicide rate in 2017 does show, however, that there is a clear correlation between counties with a large portion of republican voters and high suicide rates. The picture looks virtually the same when comparing the suicide rates in 2016 and

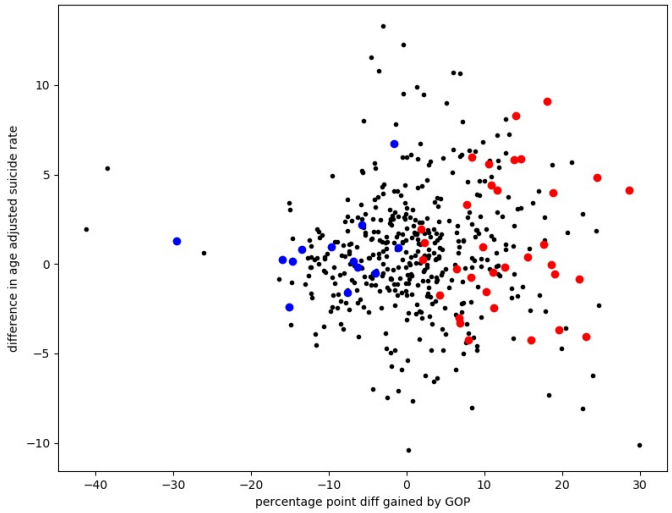

Fig. 6: Difference in age adjusted suicide rate per US county between 2016 and 2017 plotted against the gain by the republican party over the democratic party from 2012 to 2016. Counties that flipped from democratic to republican are highlighted in red, those where the reverse happened are marked blue.

the election results from 2012. This tells us that Trumps success in counties with high suicide rates does not necessarily set him apart as a candidate. It also suggests that the increased suicide rate since 2012 might explain some of his success, and not the other way around.

3.1 Methods

We generated all diagramms by performing SQL queries on our database and used python with matplotlib do visualize the data. The data for Figure 5 was selected using the following straightforward query:

```
SELECT
    year, SUM(deaths)
FROM
    suiciderate
GROUP BY
    year
ORDER BY
    year
;
```

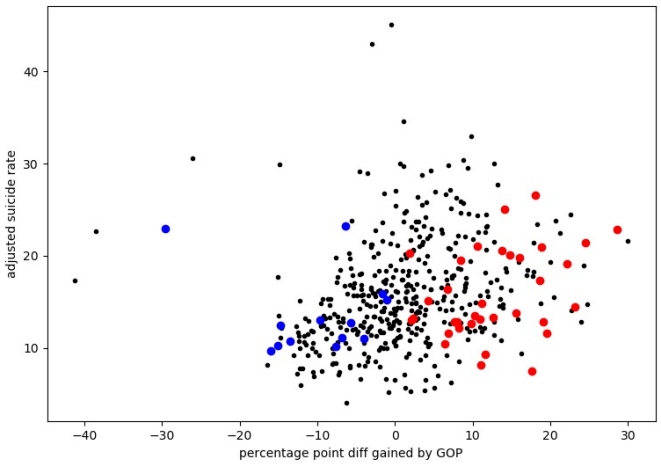

Fig. 7: Age adjusted suicide rate per US county in 2016 plotted against the gain by the republican party over the democratic party from 2012 to 2016. Counties that flipped from democratic to republican are highlighted in red, those where the reverse happened are marked blue.

Figure 6, Figure 7, Figure 8 were all created using the results of the same query:

```
SELECT
    county.name, county.state_name,
    100 * cast(e2012.votesgop - e2012.votesdem as decimal)
        /(e2012.votesgop + e2012.votesdem + e2012.votesother)
        AS perdiff2012,
    100 * cast(e2016.votesgop - e2016.votesdem as decimal)
        /(e2016.votesgop + e2016.votesdem + e2016.votesother)
        AS perdiff2016,
    s2016.ageadjustedrate AS rate2016,
    s2017.ageadjustedrate AS rate2017
FROM
    county
    JOIN electionresult e2012 ON
        e2012.countygeoid = county.gid AND e2012.year = 2012
    JOIN electionresult e2016 ON
        e2016.countygeoid = county.gid AND e2016.year = 2016
    JOIN suiciderate s2016 ON
        s2016.countygeoid = county.gid AND s2016.year = 2016
    JOIN suiciderate s2017 ON
```

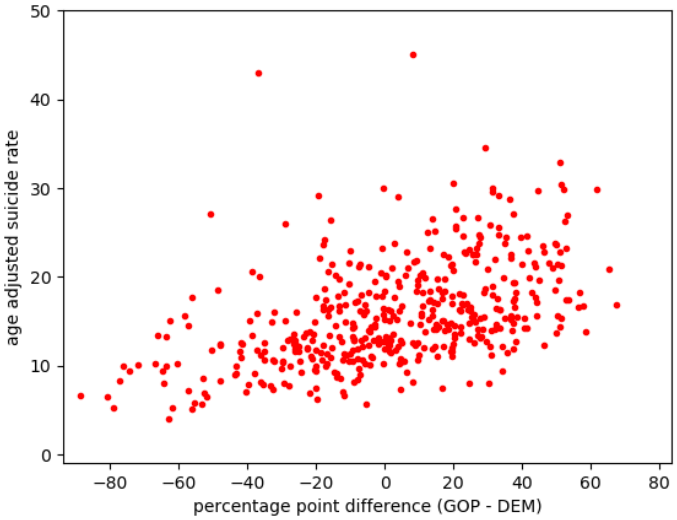

Fig. 8: Age adjusted suicide rate per US county in 2017 plotted against the percetage point difference between the GOP and the DNC in the 2016 presidential election.

```
        s2017.countygeoid = county.gid AND s2017.year = 2017
WHERE
    s2016.ageadjustedrate IS NOT NULL
    AND s2017.ageadjustedrate IS NOT NULL
;
```

This query selects every US county for which we have associated suicide rate data and lection data for both 2012 and 2016. Some counties had too few suicides or were perhaps not surveyed, so our source did not report an age adjusted suicide rate for them. Using the results of this query, we could plot any combination of the election results in 2012, those in 2016 or the difference between the two and the age adjusted suicide rate in 2016, 2017 or the difference between the two. In order to make the animation, we interpolated between the scatter plot of the 2012 election results and 2016 suicide rates and that of 2016 and 2016 repectively.

We also ran a similar query to find out to what degree the data in GDELT was able to explain the suicide rates:

```
SELECT
    county.name, county.state_name, ageadjustedrate, (
        SELECT
            AVG(goldsteinScale)
        FROM
```

```
                city
                JOIN "Event" e ON
                    e.actor2geoid = city.geoid
                    OR e.geoid = city.geoid
                JOIN "Action" a ON
                    a.id = e.actionid
            WHERE
                city.countygid = county.gid
                AND EXTRACT(year FROM e.dateoccurred)
                    = sr.year
    )
FROM
    county
    JOIN suiciderate sr ON
        sr.countygeoid = county.gid
WHERE
    sr.ageadjustedrate IS NOT NULL
;
```

Unfortunately, this query proved entirely too time consuming and was not finished running by the time the deadline of this report was approaching. After the query had been running for several minutes, we took to using postgreSQL's EXPLAIN and ANALYZE functionality to look at the execution plan of the query and estimate how long it might take to finish. The figure we can up with comparing the estimated cost by postgreSQL and comparing the cost of smaller queries to their actual runtime was about 250 hours. The culprit is more than likely a missing index on the dateoccurred field of the Event relation. Because of this missing index, the execution plan involves performing a sequential scan of our biggest table with hundreds of millions of entries inside a nested loop. We considered adding this missing index or possibly extracting the year into its own column and adding an index on that instead, but that would take too much time too.

4 Lessons Learned

Describe your lessons learned.

References

[1] *CDC Wonder*. `https://wonder.cdc.gov/`. [Online; accessed 8-January-2020].

[2] Lee Goldman. "Shorter life expectancy linked to 2016 presidential election outcome". In: (2018).