Laboratory of Data Science

Davide Piccoli, 657519
Gabriele Gori, 599135

Contents

1 Introduction

The aim of this project is to create and populate a database starting from different files and then perform some operations on it.

2 Datawarehouse building

The starting point is represented by a file called *Police.csv*, which contains the main body of data. This data regards gun violence incidents occurred between January 2013 and March 2018 in the US, including information about the victims, the guns and the locations.
There are also some additional files. The file *dates.xml* maps each date id from the *Police.csv* file to a real date. Furthermore, the files *participant age.json* (F1), *participant type.json* (F2), and *participant status.json* (F3) contain three dictionaries whose information will be used to compute the crime gravity attribute.

2.1 Database Schema

The database schema can be observed below.

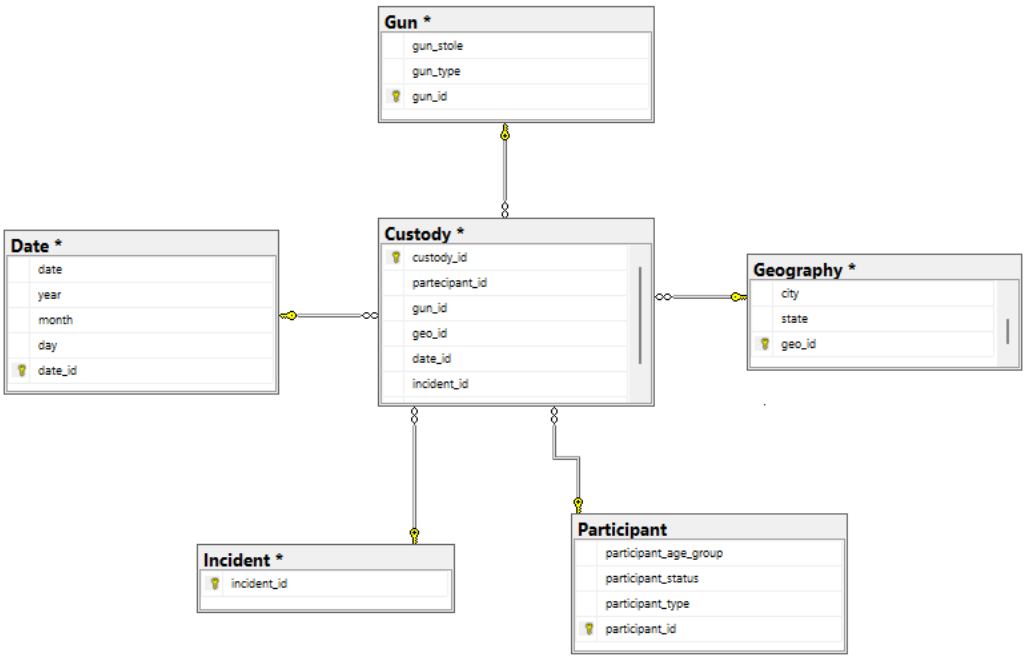

Figure 1: DB Schema obtained in Microsoft SQL Server Management Studio

2.2 Data Understanding and Pre-processing

Some basic data understanding was performed on the main table Police, in order to detect possible missing values and/or duplicates.
This table has 170928 rows and 11 attributes, which are the following:

- *custody id*: with 170928 unique values.
- *participant age group*: with 3 unique values (Adult 18+, Teen 12-17, Child 0-11).
- *participant gender*: with 2 unique values.
- *participant status*: with 4 unique values (Arrested, Unharmed, Killed, Injured).
- *participant type*: with 2 unique values (Suspect, Victim).
- *latitude*: with 65553 unique values.
- *longitude*: with 68607 unique values.
- *gun stolen*: with 4 unique values (Stolen, Irrelevant, Unknown, Not-Stolen).
- *gun type*: with 28 unique values.
- *incident id*: with 105168 unique values.
- *date fk*: with 1634 unique values.

There are no missing values in the table.

2.3 Tables

The database includes the following tables:

1. **Geography**: with attributes *latitude, longitude, city, state, geo id*. The first two features were already provided by the main table Police. The city and the state were found through two methods: the reverse_geocode library and the free US cities dataset available at *https://simplemaps.com/data/us-cities*. The first attempt was with reverse_geocode, whose search function takes a tuple of coordinates as input and returns that info. Even though cities were pretty much correct and computational time was low, the research didn't provide us the result we wanted, since it could not distinguish between US states. There are several cities with the same name but different states, and without this last piece of information we couldn't answer following assignments as we intended. To fix these issues we decided to use the US cities dataset: for every pair of coordinates we checked the closest city in the dataset based on euclidean distance,

and we attached the city and the corresponding state to our table. We did some tests to check whether this method worked. After several attempts we concluded that accuracy was pretty high. This second method took around 25 minutes of computational time. The values in the geo_id column were generated such that every unique row could have a distinct id.

2. **Date**: with attributes *date, year, month, day, date id*. The first four features were found through the file dates.xml, starting from the date_fk values already provided by the Police table. As for the id, the same process previously followed while building the Geography table was implemented here.

3. **Gun**: with attributes *gun stole, gun type, gun id*. The first two features were extracted from the Police table, while the id was generated the same way as before.

4. **Participant**: with attributes *participant age group, participant status, participant type, participant id*. The first three features were extracted from the Police table, while the id was generated the same way as before.

5. **Incident**: with attribute *incident id* generated for each unique row.

6. **Custody**: with attributes *custody id, participant id, gun id, geo id, date id, incident id, crime gravity*. Custody ids were generated the same way as the previous ids, while the others were simply extracted from the existing tables. Finally, the crime rate attribute was computed considering the participant age, type and status of each incident and thus exploiting the dictionaries included in the files *participant age.json* (F1), *participant type.json* (F2), and *partcipant status.json* (F3).

2.4 Tables upload

Once all the tables above were inserted in distinct csv files, they were read and loaded by executing a standard insert query. The data was inserted into the tables row by row until all the changes were committed. The process for some of them has been split into several sessions, exploiting the itertools library. This is due to the fact that tables like Custody and Geography were really slow to be uploaded. The upload was done through pyodbc, in the pre-existing tables created on SQL Management studio.

3 SQL Server Integration Services (SSIS)

3.1 Assignment 0

For every year, the participant ordered by the total number of custodies.

Figure 2: Assignment 0

For this task we used 5 nodes in a data flow. After the OLE DB source for the custody table, we used a Lookup node connected to the table Date to merge information exploiting the date_id. The crucial node is the Aggregation, where we grouped data firstly by year, then by participant_id, and finally we used the COUNT for custody_id. The output alias for the COUNT operation is custody_number. Then we ordered by both year and custody_number and sent data flow to the flat file output.

3.2 Assignment 1

For each state, compute the stolen gravity index defined as the ratio between the total gravity of custodies involving stolen guns divided by the overall gravity of custodies.

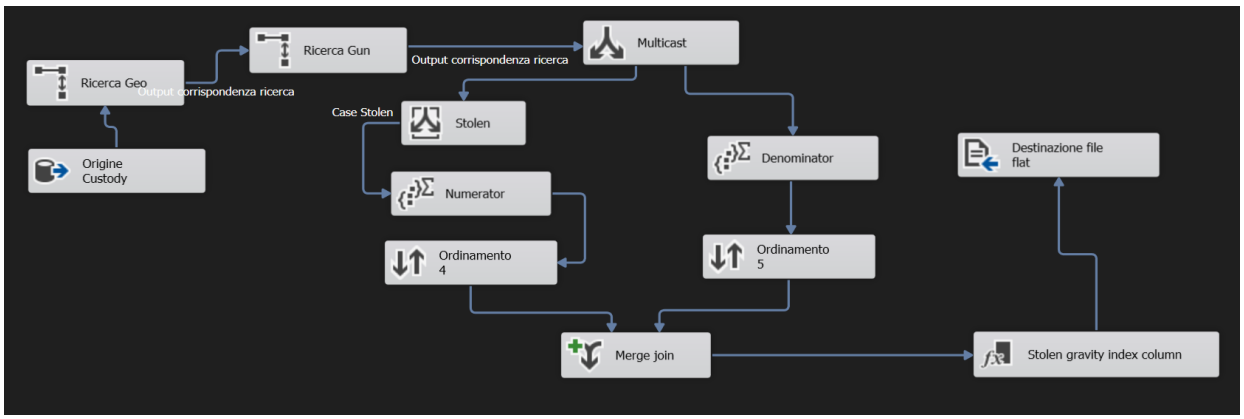

Figure 3: Assignment 1

Similarly to what we did in assignment 0, we started with Custody OLE DB source with two lookup nodes, involving Geography and Gun. Since we needed a numerator and a denominator, we exploited the multicast node to duplicate the flow. Then we computed both terms:

- Numerator: we only considered 'Stolen' records with a conditional split, in order to group by state and sum the crime gravity consequently. We had all the sums we needed, so we ordered by state (this was be useful for the merge join).
- Denominator: no data filtering was needed here, thus we repeated what we did in the previous Aggregation node to sum up all the crime gravity by state.

After these new measures were computed, we performed a merge join; we chose this node because we couldn't perform a lookup with the processed data that was generated after the previous operations. As for the the new index column creation, we applied the formula contained in the assignment using the new measures we have created, ensuring that we have multiplied the denominator by 1.0 to get a float result.

3.3 Assignment 2

For each month, compute the total gravity in percentage with respect to the annual total.

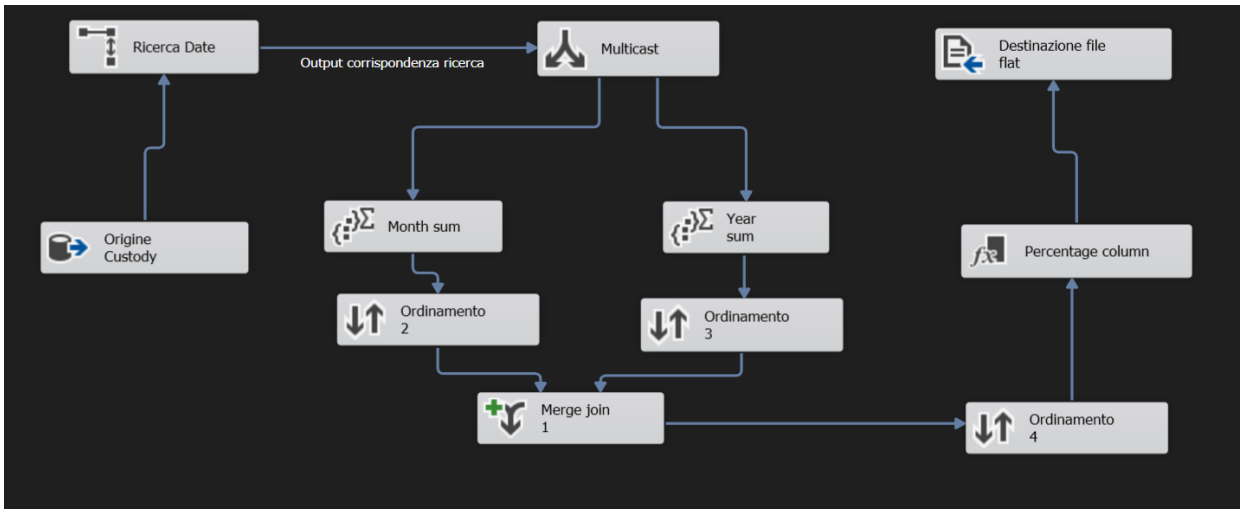

Figure 4: Assignment 2

We merged Custody and Date information as in the previous assignments. Then, after the multicast, we computed:

- Month sum: GROUP BY Year, GROUP BY Month, SUM of crime gravity
- Year sum: GROUP BY Year, SUM of crime gravity

After the ordering and the merge join, we ordered data once again by year and month. The new percentage column was made by the formula:

$$((crime\_gravity\_month * 1.0)/crime\_gravity\_year) * 100$$

Finally, we reached the flat file destination.

4 SQL Server Analysis Services (SSAS) and MDX

From now on, we use SQL Server Analysis Services.

4.1 Assignment 0

In this section we will discuss the Datacube creation. In Visual Studio we selected a new Analysis Services project.

- The first thing to do is the data source creation. We connected to our schema.
- Then, we created our data view exploiting the data source.
- The next passage regarded the actual cube creation. We imported every table in our schema.

An important task here was the creation of hierarchies in our dimensions: these will be useful for the next queries. Two of them were created:

- DayMonthyear: a hierarchy from Year to Day which is the deepest level of detail.
- CityByState: state is the lowest level.

4.2 Assignment 1

Show the total crime gravity for each city and the grand total with respect to the state.

```
WITH MEMBER ratio AS
    IIF([Measures].[Crime Gravity] = null, 0,
    ([Measures].[Crime Gravity]/([Geography].[City].CurrentMember.parent,[Measures].[Crime Gravity]))),
    format_string='Percent'
    member state_grandtotal as
    aggregate({[Geography].[City].CurrentMember.parent}, [Measures].[Crime Gravity])

SELECT {[Measures].[Crime Gravity], state_grandtotal, ratio} ON COLUMNS,
    CROSSJOIN([Geography].[City].[City], [Geography].[State].[State])  ON ROWS
FROM [Group ID 7 DB]
```

First of all we created two new variables: "ratio" and "state_grandtotal". The variable "ratio" considers the crime gravity of a given city: if it is a null value, "ratio" is equal to 0, otherwise it computes the ratio between the crime gravity of that city and the crime gravity of the state the city belongs to, converting it into a percentage value. Since we also wanted to show this latter value in the final output, the variable "state_grandtotal" was created as well, considering indeed the grand total of each state crime gravity.
The query then simply selects the measures we are interested in, thus crime gravity, state grand total and ratio. Each city and the related state are selected on rows in order to provide the final result and a crossjoin makes sure that these two measures are selected together despite being on the same hierarchy level.

4.3 Assignment 2

Show the percentage increase or decrease in total crime gravity answers with respect to the previous year for each age group.

```
WITH MEMBER difference AS
  IIF(([Time].[Year].CurrentMember.PrevMember, [Measures].[Crime Gravity])=null, 0,
  (([Time].[Year].CurrentMember, [Measures].[Crime Gravity]) - ([Time].[Year].CurrentMember.PrevMember,
  [Measures].[Crime Gravity]))/([Time].[Year].CurrentMember.PrevMember, [Measures].[Crime Gravity])),
  FORMAT_STRING='Percent'

SELECT {[Measures].[Crime Gravity], difference} ON COLUMNS,
CROSSJOIN([Participant].[Participant Age Group].[Participant Age Group], [Time].[Year].[Year])  ON ROWS
FROM [Group ID 7 DB]
```

Here the measure "difference" is created. It basically represents the increase or decrease in total crime gravity between subsequent years. Of course, if the previous year is not included in our data, this measure is equal to 0. If the previous year is available, the percentage increase or decrease in total crime gravity is computed.
The final output is obtained by selecting the total crime gravity for each age group in each year, together with the related "difference" measure.

4.4 Assignment 3

Show the ratio between the total crime gravity of each year w.r.t the previous year.

```
WITH MEMBER ratio AS
  IIF(([Time].[Year].CurrentMember.PrevMember, [Measures].[Crime Gravity])=null, 0,
  [Measures].[Crime Gravity]/([Time].[Year].CurrentMember.PrevMember, [Measures].[Crime Gravity]))

SELECT {[Measures].[Crime Gravity], ratio} ON COLUMNS,
       ([Time].[Year].[Year].Members) ON ROWS
FROM [Group ID 7 DB]
```

Also in this case a new measure is created: "ratio". Similarly as before, if the previous year is not present in the data, this measure is equal to 0. Otherwise, it computes the ratio between the total crime gravity of a given year and the one of the year before.
Then we proceeded to the selection of crime gravity and ratio on columns, followed by the selection of "Year" instances on rows.

4.5 Assignment 4

Create a dashboard that shows the geographical distribution of the total crime gravity in each age group.

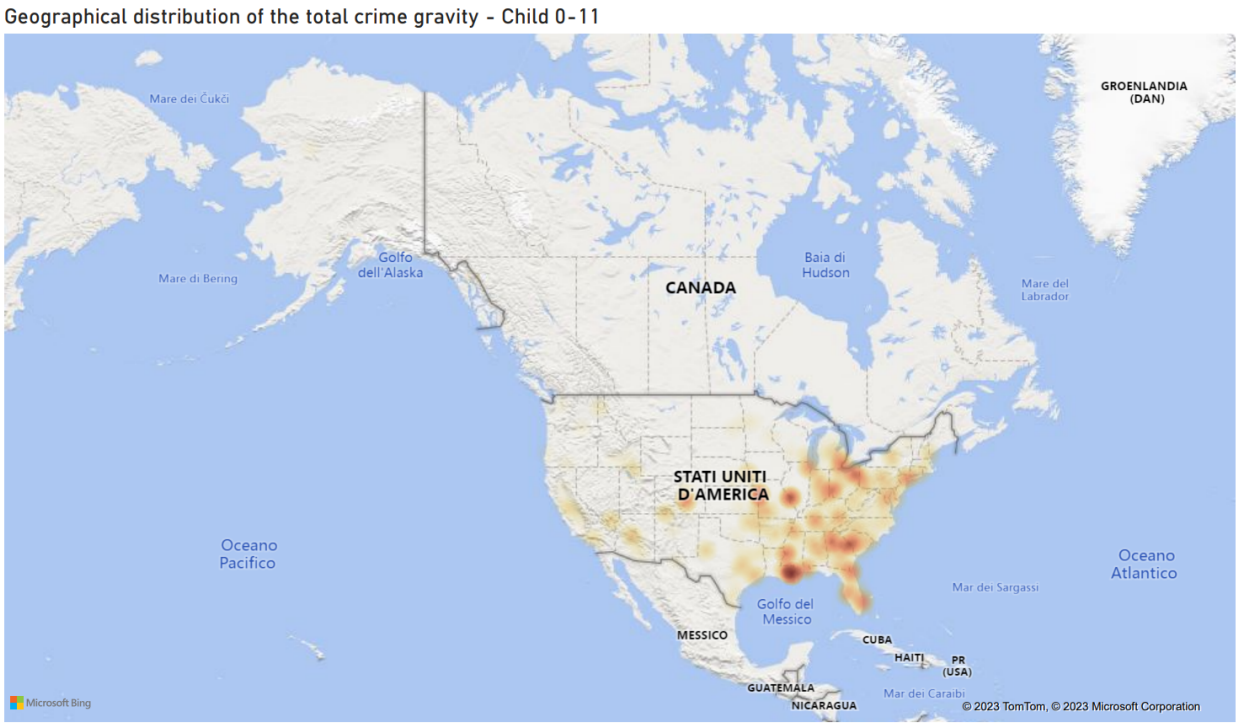

Figure 5: Geographical distribution of the total crime gravity within the age group Child 0-11.

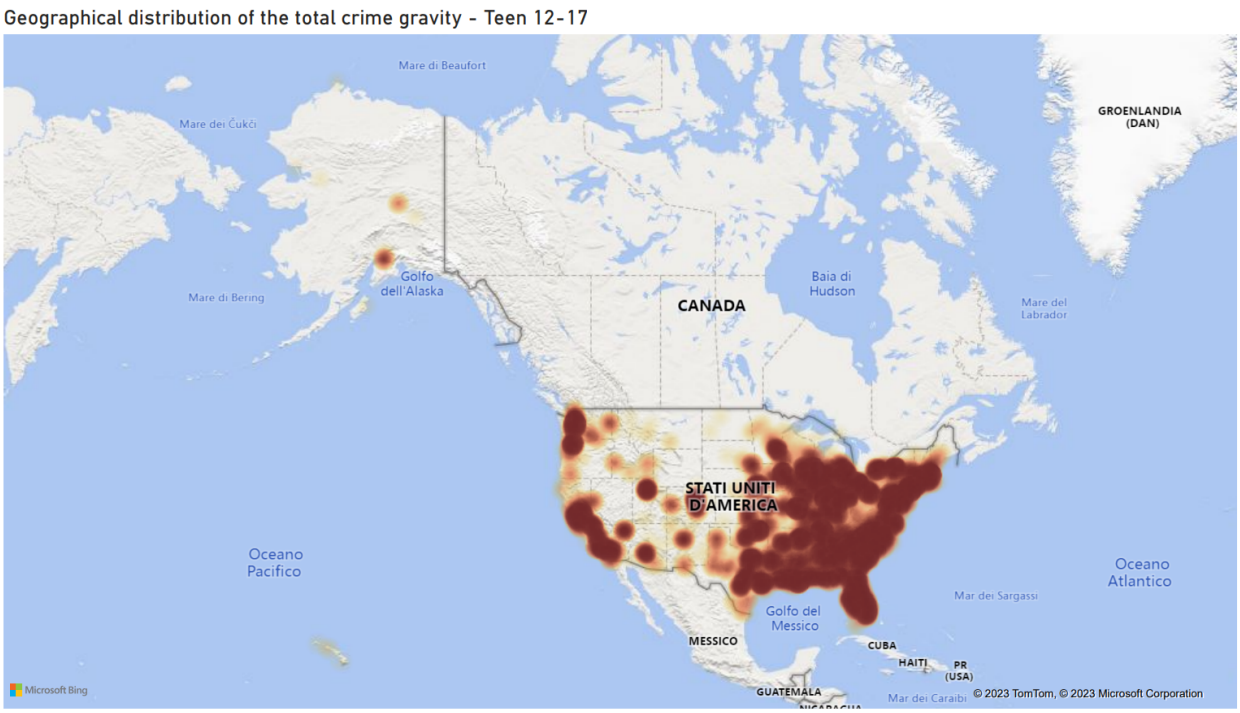

Figure 6: Geographical distribution of the total crime gravity within the age group Teen 12-17.

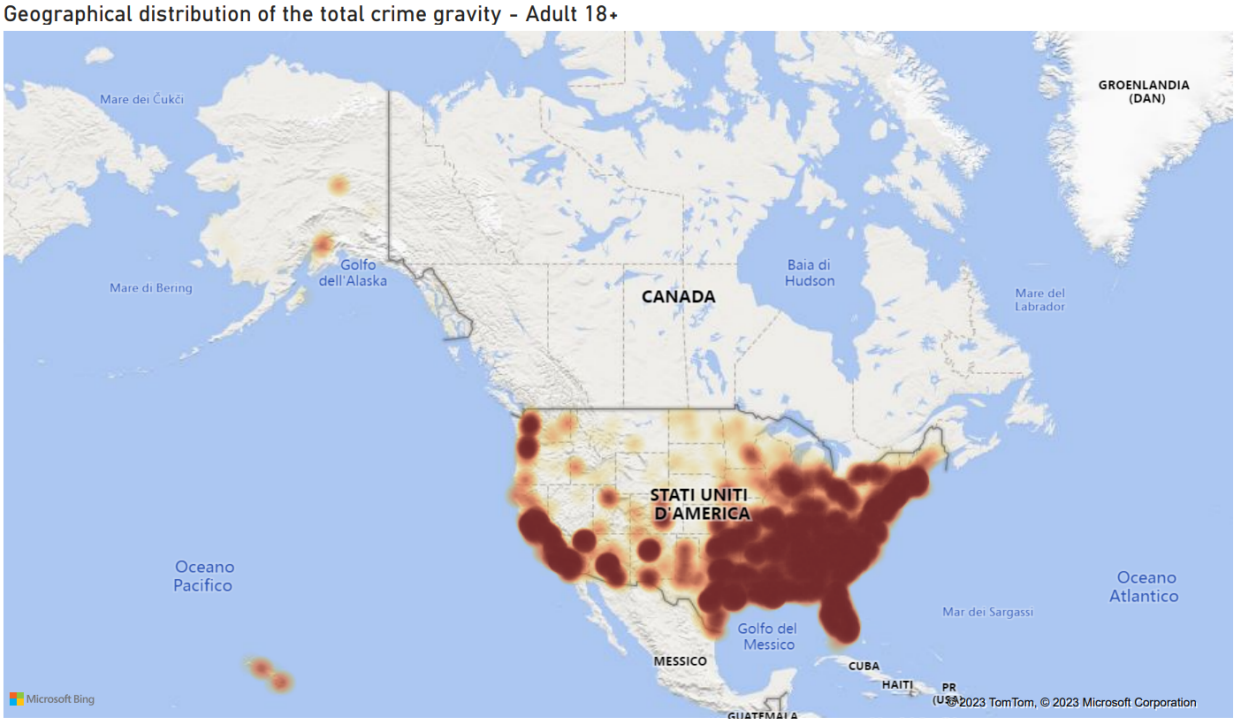

Figure 7: Geographical distribution of the total crime gravity within the age group Adult 18+.

Three heatmaps were employed for this dashboard, one for each age group. These maps actually highlight a strong correlation among the crime gravity within different age group *Adult 18+* and *Teen 12-17*. They look very similar from this perspective, but some minor differences can be spotted by zooming in.
Overall, the east side of the country seems to register many more incidents compared to the west side, where only the area along the coast and some inland cities are highlighted. Of course, the crime gravity distribution also reflects the urban distribution in the US: for example, the Rocky Mountains area on the west side can be easily noticed. Also, the very low population density in Alaska causes this state to exhibit a distinctive pattern in its crime gravity distribution despite its vastity. In Figure 8 a recap for this assignment is reported. Bigger bubbles mean a higher overall crime gravity. Furthermore, it is easy to spot a high concentration of records in Fairbanks, Alaska. Some bubbles in the east side show an high ratio of child involved.

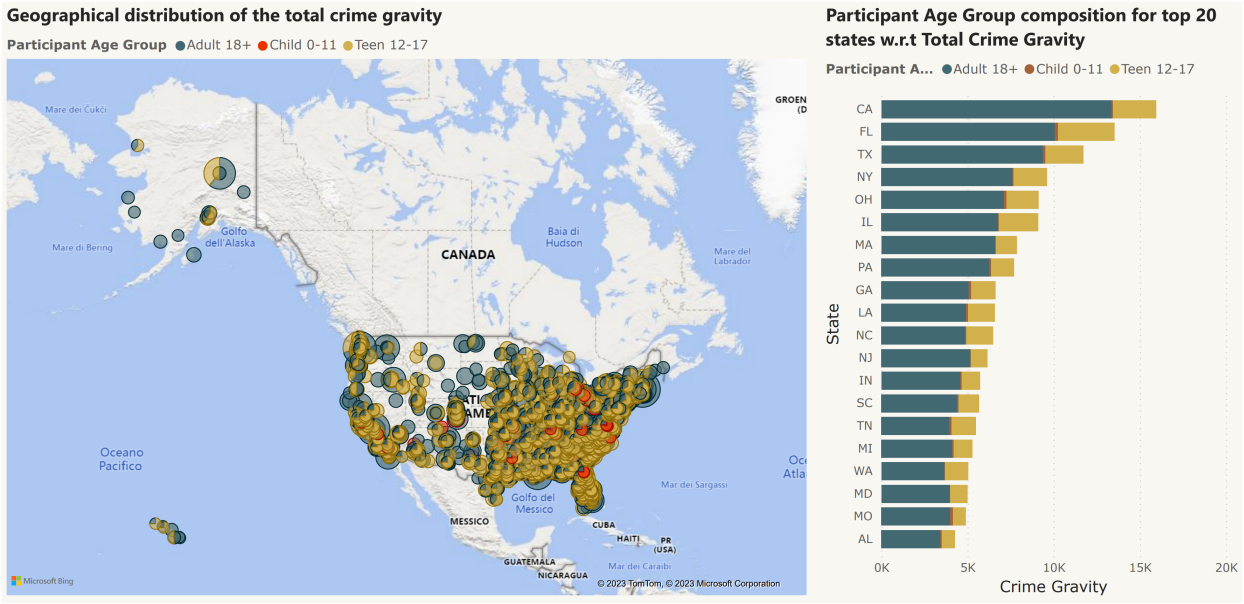

Figure 8: Assignment 4 recap dashboard

4.6 Assignment 5

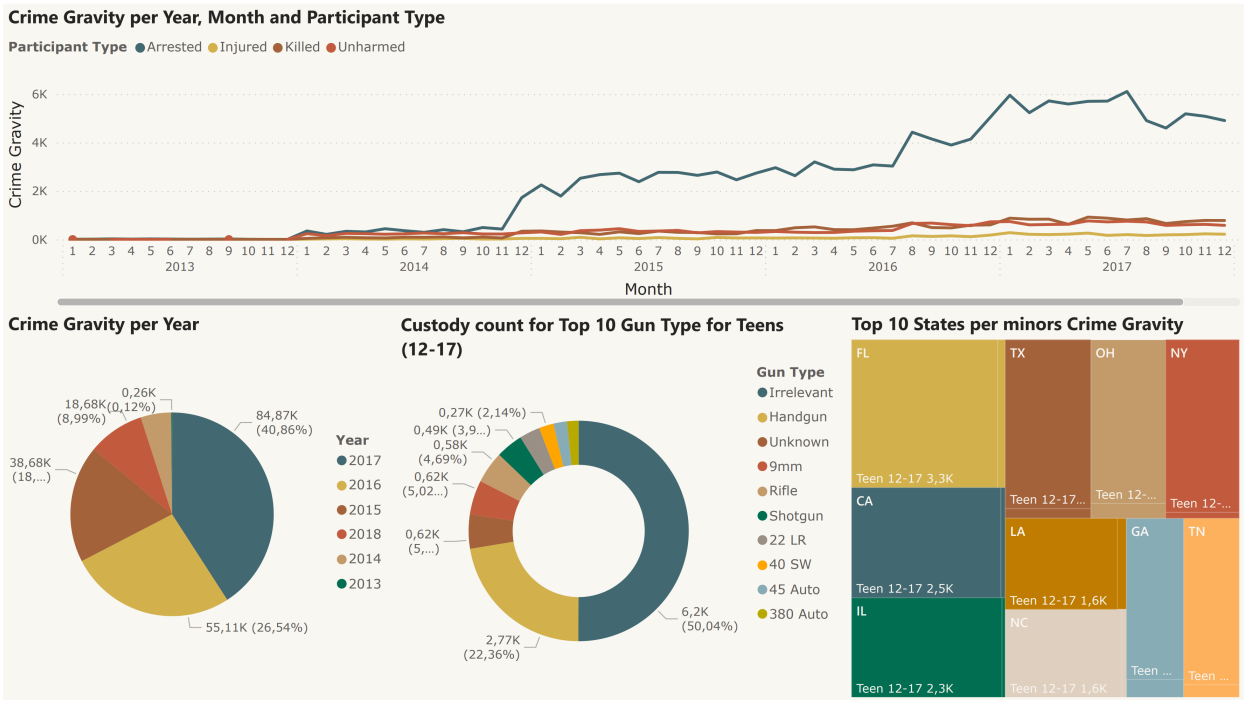

Figure 9: Dashboard

One interesting fact to notice is that total crime gravity has a major peak in 2017, following the growing trend from 2016 and 2015. The pattern for 2013 and 2014 is also worth a mention, showing that crime gravity for arrested people seems to be really low with respect to following years. Furthermore, all the categories follow an increasing trend until 2018. In the lower center box we reported the top 10 gun types for teen participant, where we can spot a noticeable majority of handguns within actual guns, while the top 50% of gun types in this plot were categorized as irrelevant. The other kinds of gun types show an heterogeneous distribution. If we focus only on minor subjects (child and teens), we can easily see that Florida occupies the highest spot on the podium for total crime gravity, followed by California and Illinois (lower right plot). Illinois, considering the overall crime gravity, is just on 5th place (figure 8), meaning that a significant part of crimes was carried out by underage subjects.