Impact estimation of earthquakes with a Twitter analysis

Final mid-term report

**School:** Lucerne University of Applied Sciences and Arts  
**Program:** Master of Science in Applied Information and Data Science (MScIDS)  
**Course:** Data Warehouse and Data Lake Systems 1  
**Team:** Earthquake  
**Place/Date:** Lucerne, 21.11.2021

**Supervisors:** José Mancera and Luis Teràn

**Authors**:

|  |  |  |
| --- | --- | --- |
| Sandro Huber Spitalackerstrasse 20a 3013 Bern +4178 840 09 09 sandro.huber@stud.hslu.ch | Thomas Schwendimann Spitalstrasse 8b 6210 Sursee +41 79 129 19 43 thomas.schwendimann@stud.hslu.ch | Lea Senn Treppenweg 19 5300 Turgi +4179 960 74 79 lea.senn@stud.hslu.ch |
|  |  |  |



List of Contents

[1 Background 3](#_Toc86850936)

[Problem statement 3](#_Toc86850937)

[Business questions 3](#_Toc86850938)

[Methodology 3](#_Toc86850939)

[1.1 Data sources 3](#_Toc86850940)

[1.1.1 Twitter API 3](#_Toc86850941)

[1.1.2 Earthquake API 4](#_Toc86850942)

[1.2 Procedure 4](#_Toc86850943)

[1.3 Data visualisation 5](#_Toc86850944)

[2 Sources 7](#_Toc86850945)

[2.1 Literature 7](#_Toc86850946)

[2.2 Images 7](#_Toc86850947)

[3 7](#_Toc86850948)

[Declaration 9](#_Toc86850949)

[4 References 10](#_Toc86850950)

[5 Appendix 11](#_Toc86850951)

List of Figures

[Figure 1‑1 - Insured losses for earthquakes/tsunamis, 1970-2020, in USD billion at 2020 prices (Source: Swiss Re Institute) 3](#_Toc88289892)

[Figure 1‑2 - number of victims depending on catastrophes (excluding US), 2015-2020, (Source Swiss Re Institute) 4](#_Toc88289893)

[Figure 2‑1 - Screenshot of the dashboard by USGS (Elaine, 2015 6](#_Toc88289894)

List of Tables

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

# Background

As a global reinsurance company, our client SwissRe is affected by global catastrophes on a broad scale. In 2020, SwissRe estimated the global insured catastrophe losses to be USD 83 billion (SwissRe, 2020). One such cost factor can be caused by earthquakes, which makes it all the more important to get as much information and as quickly as possible about the scale of these natural disasters.

## Problem statement

While the cost for catastrophes from earthquakes is not relevant in every year (Figure 1‑1), the influence on both insurance company as well as their clients is quite large, both in costs as well as number of victims (Figure 1‑2).

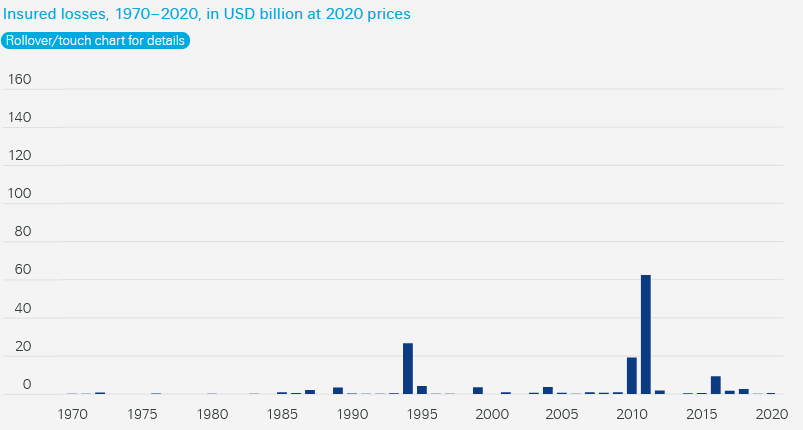


Figure 1‑1 - Insured losses for earthquakes/tsunamis, 1970-2020, in USD billion at 2020 prices (Source: Swiss Re Institute)

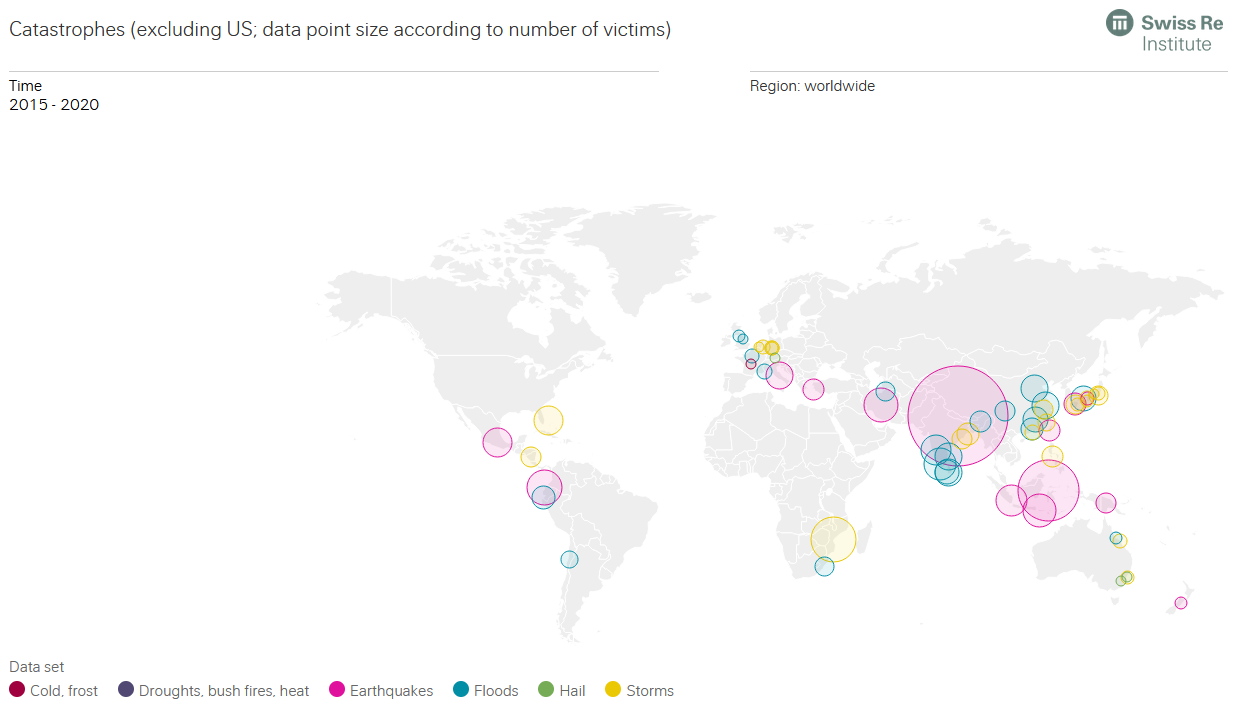


Figure 1‑2 - number of victims depending on catastrophes (excluding US), 2015-2020, (Source Swiss Re Institute)

While seismographs around the world record earthquakes – even if they are barely perceptible to humans – and make the data available in real time, the effects of damage to buildings and people are often only visible later when the damage reports arrived. Additionally, the seismographs are reporting earthquakes all around the world, which means also at very remote places where no costs for SwissRe arise. Therefore, the information from the seismographs should be enriched with qualitative data in order to better estimate an earthquake.

This also enables SwissRe to be there for their affected clients immediately, improving customer experience and possibly even reducing the damage. In January 2019, SwissRe wrote a lengthy report, indicating how the future business models for insurance companies may change due to digitalisation. One such change is how insurers interact with their clients. As of now, the customer journey is mostly limited to the signing of the contract and the claims, which may or may not arise. In the future, insurance companies might try to be involved more closely with their clients, even utilizing touchpoints such as mobile phone apps, where the insurance company can warn their clients ahead of possibly damaging events (Avramakis, Anchen, & Raverkar, 2019).

## Business questions

Hundreds of earthquakes are registered worldwide every day. Fortunately, most of them cause little damage to buildings and people. But for larger earthquakes that produce a lot of damage, SwissRe’s claims experts want to know as soon as possible what costs they need to expect.

We assume that bigger damages also lead to a bigger response in social media. To estimate the damage, we want to try to analyse the "Twitter echo" after big earthquakes and compare the amount of Tweets with the numbers from comparable earthquakes in the past.

We plan to make this comparison available to SwissRe's claims experts on a dashboard in an intuitive way. The claims experts from SwissRe can then use this information and match it with their historical cost data. They might find a relationship between the number of tweets and the costs and can use the information to forecast future costs for earthquakes.

For this comprehensive analysis, we will analyse stronger earthquakes (i.e. magnitude of 5 or more on the Richter scale) from the past. We suspect that relevant Twitter messages can be found for the last five to ten years. In the period before that, Twitter was probably not widespread enough.

Further, a sentiment analysis of the content of the tweets could be interesting to get more insights about how large the damage is. Since NLP is not the main topic of this course, we see this part as optional.

We decided to answer the following research questions:

* *How many Tweets including the word “earthquake” were sent within 24 hours after major earthquakes in the past 5 years?*
* *What is the correlation between the magnitude of the earthquake and the number of Tweets posted?*
* *How did the sentiment in the Tweets change based on the magnitude?*

# State of the art

The U.S. Geological Survey of the USGS (which also provides the API used in this work) has an article examining the influence of earthquakes on tweets. The researchers come to the conclusion that “the tweet-frequency time series constructed from tweets containing the word "earthquake" clearly shows large peaks correlated with the origin times of widely felt events.” Also important to our work is the USGS's recognition that the response to Twitter comes very quickly. About 75 percent of tweets are registered within two minutes of the quake. The USGS refers to this as “considerably faster than seismographic detections in poorly instrumented regions of the world.” (Earle et al., 2011)

The finding that Twitter reports a faster response time than the USGS was also evident in the 2008 Sichuan earthquake, in part because the USGS operated most of its seismographs in the U.S. at the time and areas such as China were poorly equipped. This fact drove the USGS to continue working with data from Twitter. This also created dashboards with various data on earthquakes on Twitter (Figure 2‑1). Details about this project are available on Twitter's blog (Elaine, 2015).

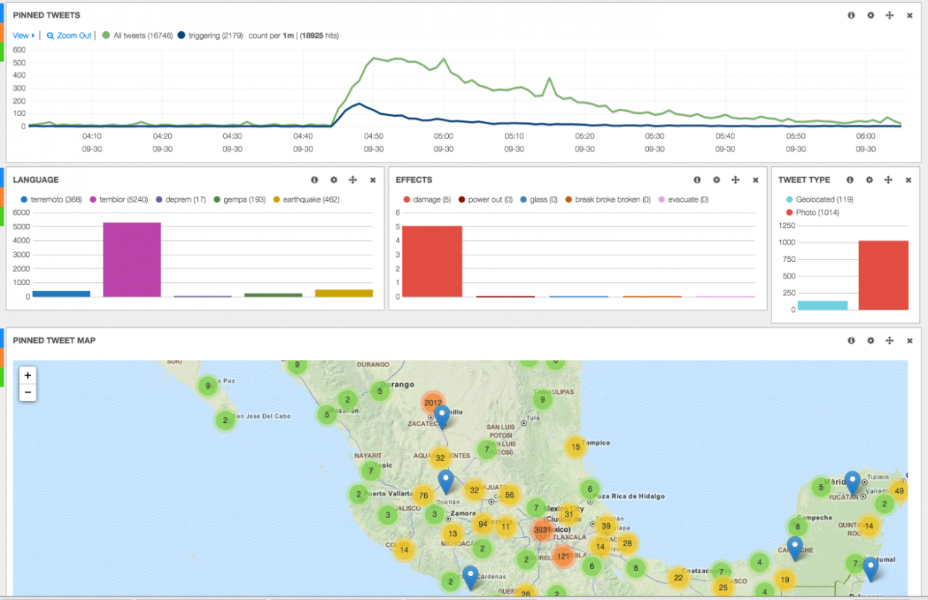


Figure 2‑1 - Screenshot of the dashboard by USGS (Elaine, 2015

# Methodology

The aim is to quantify the twitter echo after a major earthquake by counting the tweets and retweets including the word “earthquake” on twitter within 24 hours after an event. It is planned to do this for a selection of the most severe earthquakes in terms of their magnitude, probably for earthquakes with a magnitude of 6 or more on the Richter scale, as these are described as “Damage to a moderate number of well-built structures in populated areas. Earthquake-resistant structures survive with slight to moderate damage. Poorly designed structures receive moderate to severe damage. Felt in wider areas; up to hundreds of kilometers from the epicenter. Strong to violent shaking in epicentral area.” (USGS, 2010). We suspect that the greater the damage and therefore the greater the costs, the more earthquake-related messages are posted on Twitter.

This chapter describes the data scources used in this project and explains the used technology and procedure.

## Data sources

Two different APIs are going to be used for this project.

### Twitter API

Twitter is a social media platform with roughly 500 million tweets every day, which accumulates to 6’000 tweets a second (brandwatch, 2020). As such, Twitter is quick to pick up on trends and events that are happening right at this moment, all around the globe. The Twitter API gives access to historic tweets as well as streaming them in real time. To be able to use these services to their full extent, an academic research account is required, for which we applied for.

From our first tests we know that around 10‘000 to 15‘000 tweets including the word „earthquake“ are posted on Twitter every day. A significant part of them are posted by bots, that do not react to human perception or even damanges.

Information about the Twitter API are published under <https://developer.twitter.com/en/docs/twitter-api>

### Earthquake API

The National Institute of Standards and Technology leads the USGS Earthquake Hazards Program of the U.S. Geological Survey (USGS). On their website, they provide an API with relevant earthquake science information and knowledge, including the time, location, magnitude and others. The API is updated every minute.

Details can be found under <https://earthquake.usgs.gov/fdsnws/event/1/>.

# Procedure

## Twitter Data

In a first step, the Twitter API is used to access the historical data for as far back as possible (max. 10 years) with the limit of 10 million tweets a month. Python’s library ‘requests’ is used for all GET requests to the API. The code is executed in a Directed Acyclic Graph (DAG) from Apache Airflow. This chapter provides an overview of the steps involved in extracting the data and storing it in a postgreSQL database – the data lake – using a DAG.

### Preparation

Due to the monthly limit of tweets that can be accessed, it was not possible to create a query which returns every tweet containing the word ‘earthquake’. One day returns roughly over 10’000 tweets, which would limit the time frame that is accessible for this project. In order to increase this time frame, test queries have been executed to optimize the final search criteria. Retweets, replies and the word ‘minor’ are excluded from the search. Furthermore, the number of tweets one account posts a day were analyzed for a short time period. In order to further reduce the number of results we obtain, the accounts with the highest number of tweets were extracted from the data and excluded from the query, thus removing some of the bots which tweet about every earthquake measured, including those of no relevance, such as earthquakes with a low value on the richter scale. This also helps to only get the actual human responses. Since the length of the query is limited to 1’024 characters, only around 40 usernames have been excluded. Once the data is in the data lake, the other bots will be removed when transforming the data before it is loaded into the data warehouse.

### Data Lake

For the data storage, a postgreSQL database has been created in Thomas Schwendimann’s Amazon Relational Database Service (RDS). This database serves as a data lake, where all the Twitter data will be stored in its raw format. This includes for one the tweets matching the criteria, but also information about the users which posted them. These data will be stored in separate tables.

### Extracting the Data

Figure 4‑1 represents the DAG for the process of extracting and loading the Twitter Data into the data lake. Due to the access limitation of the API, such as the number of allowed requests every 15 minutes, this DAG is executed every six hours and will extract half a year worth of data. The defined time interval is 1 January 2010 to 31 October 2021.

Diagram

Description automatically generated

Figure 4‑1 - DAG Twitter Data

The first tasks checks if the corresponding tables (‘tweets’ and ‘users’) in the data lake exist, and if not creates them. The tables are therefore ideally only created with the first execution. Next up are the query parameters, which are defined in a separate task. Arguably, this step could be combined with the extracting of the data, but was chosen to be separate for a better overview. Two variables were added to the Apache Airflow environment, allowing to access them without writing the values as code. First, the bearer token required for the authentication when making the API requests. Second, the filter for the query, which consists of the prefix ‘-from:’ and the usernames that are excluded from the search. These variables are accessed and used in the set up of the query. The query parameters, URL and headers required for the request are saved using XComs (cross-communications), a mechanism that allows to send data between tasks.

The next step pulls the XComs with the relevant variables and accesses the API and requests the Twitter data matching the query. Only 500 tweets can be accessed per request from the API. With every request the API returns a token in the metadata – assuming more results are there but were not returned due to the limit. Using this token, one can then request again and will get the data beginning from where the previous request left off. This is done in an infinite while loop, which stops if there is no token in the metadata, indicating that all the results have been delivered. For the tweets, the fields ‘created\_at’, ‘text’, ‘author\_id’ and ‘id’ are returned as per query defined. For the users, the fields ‘name’, ‘username’, ‘location’ (if available) and ‘id’ (which matches the author id). All the responses are stored in lists containing a dictionary (json format) for each tweet and user respectively – and then once again saved using XComs.

### Loading the Data

The loading of the tweets and user information into the database is done in two separate tasks. In both cases, an iterator is created from the list containing the data – in order to save memory. Since not all user entries contain the key ‘location’, the key and an empty string as value is added in its place if not in the dictionary. This allows for easy insertion into the database. To connect to said database, a PostgresHook accesses the connection details saved in Apache Airflow – once again to not have sensitive data visible in the code. Due to the volume of data, the execute\_batch function from psycopg2’s library is used to insert the rows into the tables. This requires a separate import psycopg2.extras. If the data volume were even higher, the copy\_from method would have been used, which has the fastest execution time when combined with an iterator (Benita, 2019). Finally, once the data has been loaded, a simple SELECT request is done for each table to validate if the process has been successful or not. In the tables below is the output of a simple SELECT statement using pgAdmin 4, a software with a GUI for postgreSQL databases.

Graphical user interface, text, application

Description automatically generated

4‑1 - Abstract table tweets

Graphical user interface, text, application, chat or text message, email

Description automatically generated

4‑2 - Abstract table tweets\_user

## Earthquake data

The earthquake data was retrieved from the earthquake.usgs.gov API. The API is provided by the United States Geological Survey (USGS) and can directly be accessed over the URL without any tokens or other verifications needed. The data can already be filtered via setting of parameters in the URL of the API. To safe computational power and space on our web server, this filtering method was used to pre-filter all earthquakes that are not relevant for the project. This was done with the definition of the following parameters:

* Starttime
* Endtime
* Minimum magnitude

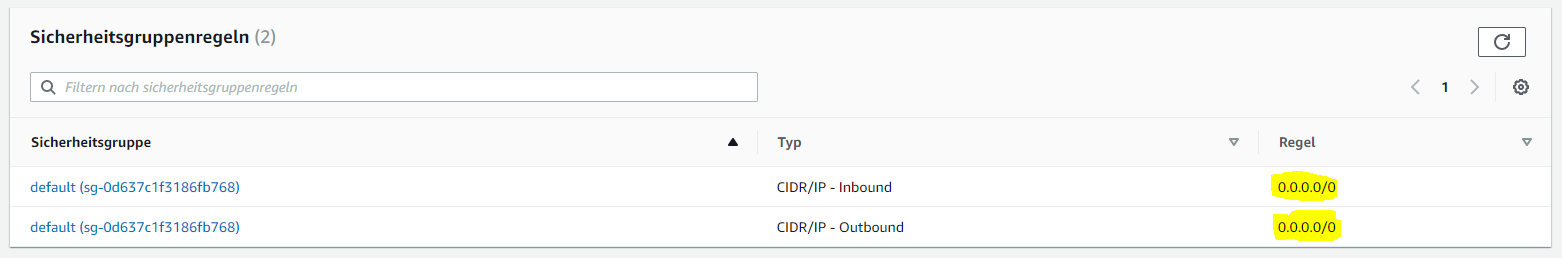


The time frame of the observation is between the first of January 2015 until the 31.10.2021. This dates were chosen in order to have a big sample size but also to limit ourselves to a certain time frame that is possible to cover with the twitter data. Additionally, the minimum magnitude of six has been defined, because of the above already described reasons in the elaboration of the background of the project.

With help of this first filtering of the data, the API delivers 932 relevant earthquakes. Obviously, this list has again to be cleaned, for example in terms of their location, because of the irrelevancy of earthquakes that are happening in the ocean or other remote places. This duty will be done during the next project stage in the second half of the semester.

### Data storage

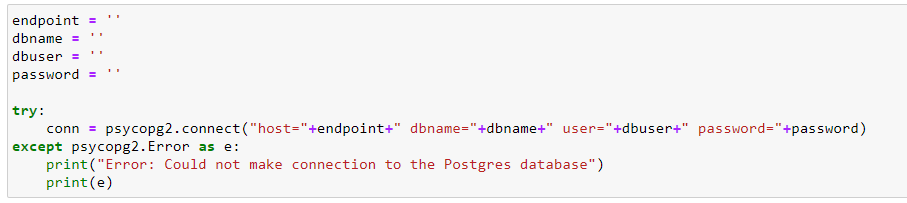
To have an uniform data storage, the earthquake data is stored in a postgreSQL database, exactly like the twitter data. For this purpose, the database first needed to be created in the amazon rds. When it comes to safety aspects in terms of limited access to the database from specific IP-addresses, our group could not follow the best practice of a limited access. It was necessary to leave the access open to all IP-addresses since no one of our group member has a static IP-address. Therefore, this option was not possible for us. However, the data is not sensitive, therefore, safety is not a big issue in this case. Nevertheless, the database is secured with help of a password, which is sufficient for this project.



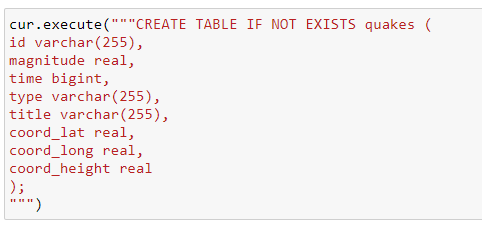
### Data loading

This step was first tried to be accomplished with help of a small sample dataset with help of a jupyter notebook. This was done with help of the following steps:

* Downloading the data via USGS API
* Small data preparation step
  + Unfolding of the geodata points
* Connecting to the database with the according credentials



* Creation of a table in the PostgreSQL database with the following code snippet

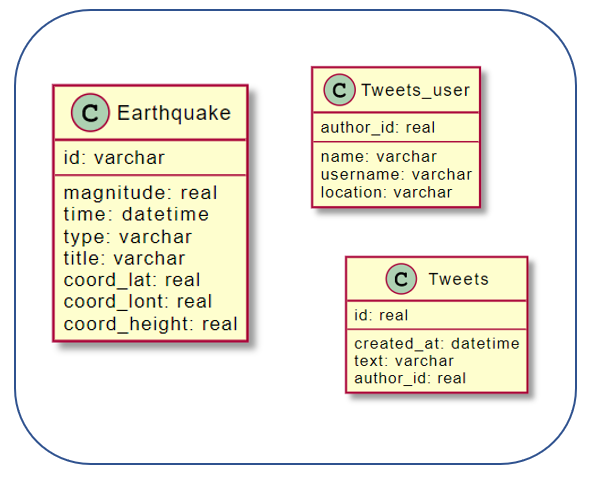


* Implementation of the data
* Close the connection again

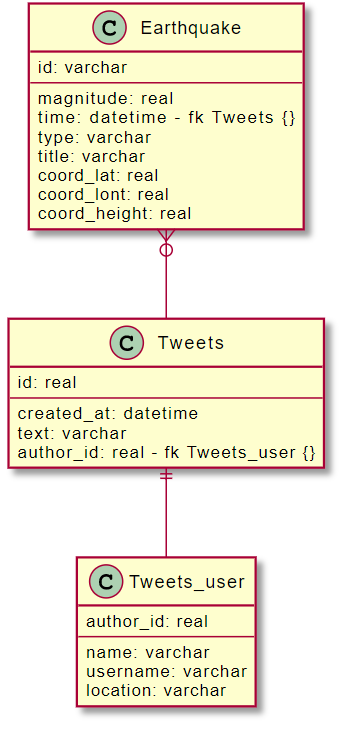
After the successful implementation of the data into the PostgreSQL server with help of this little local prototype, the implementation was automated with help of Apache airflow.

## Data lake storage overview

Finally, the created data lake includes the following information:



Until this point, the data is floating in the database in loose tables. This is the goal of the first stage of this project and has already been reached. However, it is important to have an outlook on how the data can be prepared and rearranged to combine these raw data tables. Therefore, the following relationship model has been drawn:



After the processing of the data, the tweets can be associated with the tweets according to the “time” of the earthquake and the “created\_at” of the tweets. An earthquake can have zero or multiple tweets. Additionally, a twitter user can directly be assigned to a tweet with help of the “author\_id”. One tweet must have exactly one user who created the tweet.

## Project organisation

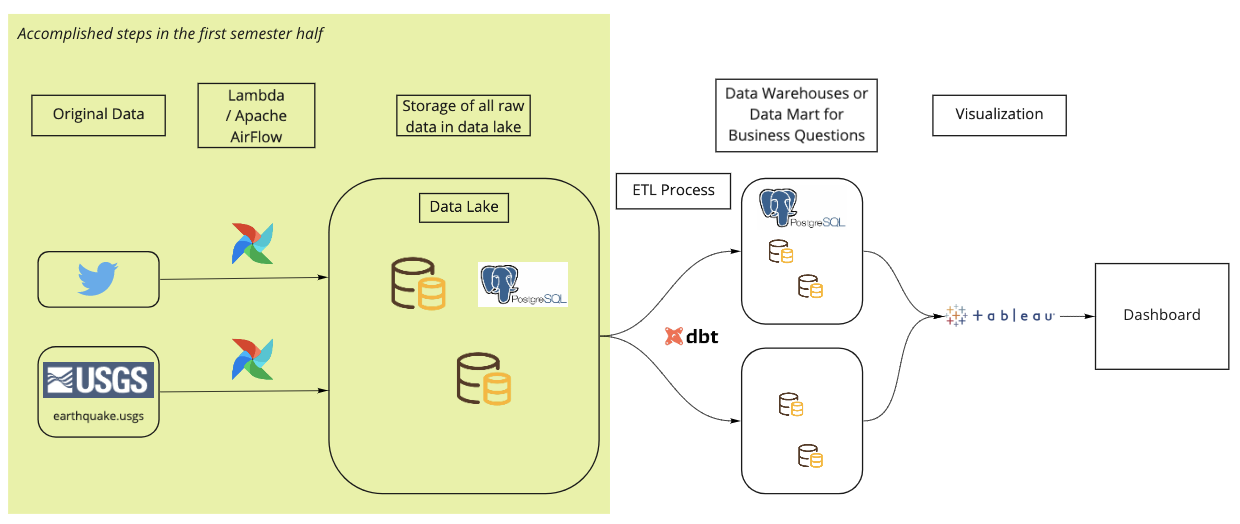
According to the recommendation of the lecturers, a local prototype in Jupyter Notebook is created first. This allows the students to familiarize themselves with the APIs. In addition, for the time being it is only possible to get a certain extract from the complete data set. This is especially important for Twitter, where the number of tweets per developer account is limited.

The local prototype will later be emigrated to Apache Airflow, where it will regularly (presumably once a day) retrieve the latest data from the API and store it in the database on Amazon Webservices.

## Used technology

In the following illustration, the projects high level architecture can be seen. Beside a rough overview of the steps that the data is aiming to take, also the tools that are planned to be applied are visualized.

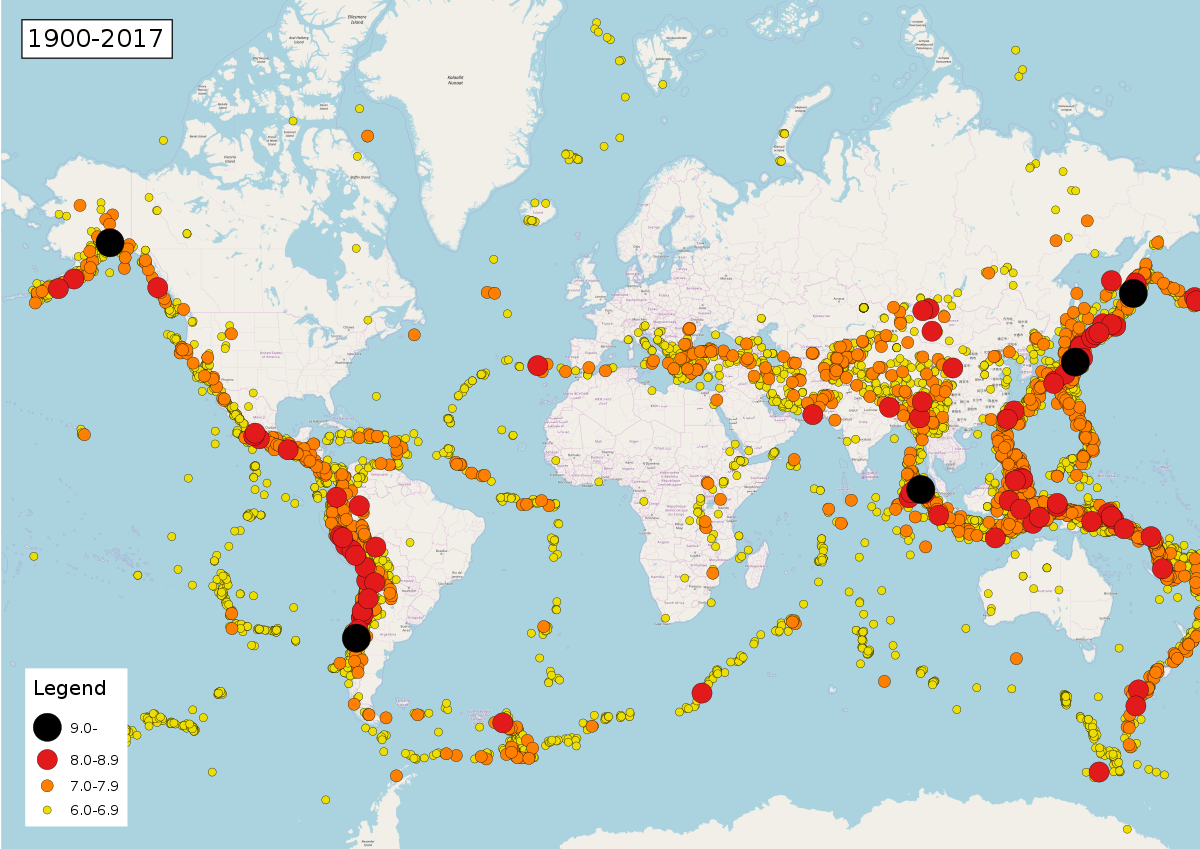
The highlighted part shows the accomplished steps in the first half of the semester.



*Image 3: High level architecture*

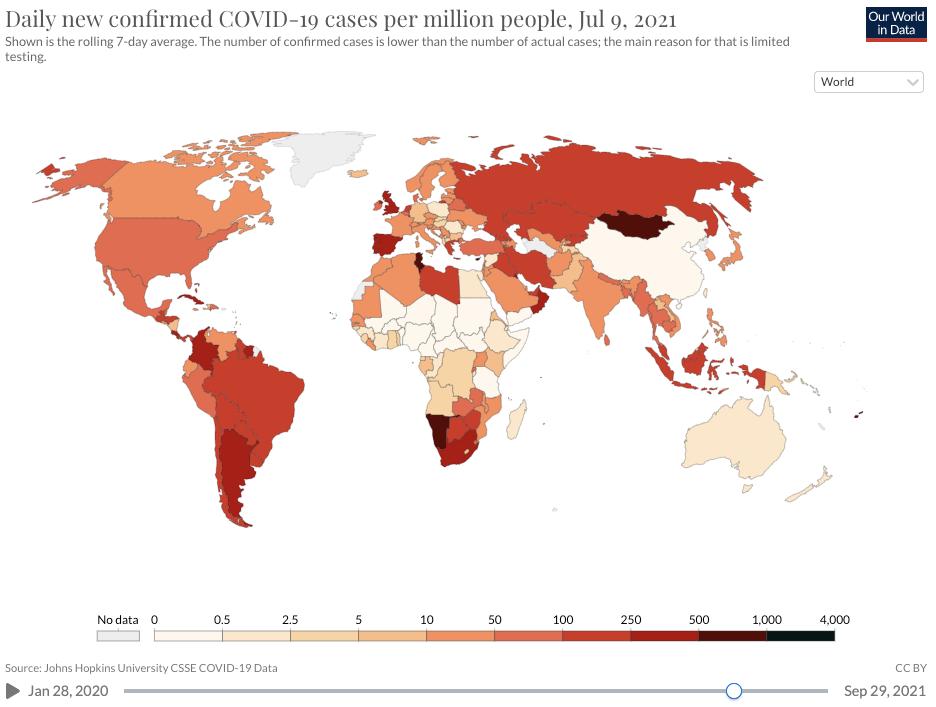
## Data visualisation

We plan to visualise the earthquakes on a map. It should ideally show the magnitude (the heavier an earthquake was, the bigger the symbol on the map). If possible, we would like to show the Twitter reaction on the same map (also the bigger the Twitter reaction, the bigger the symbol).



*Image 4: Wikipedia Map of Earthquakes 1900 - 2017*

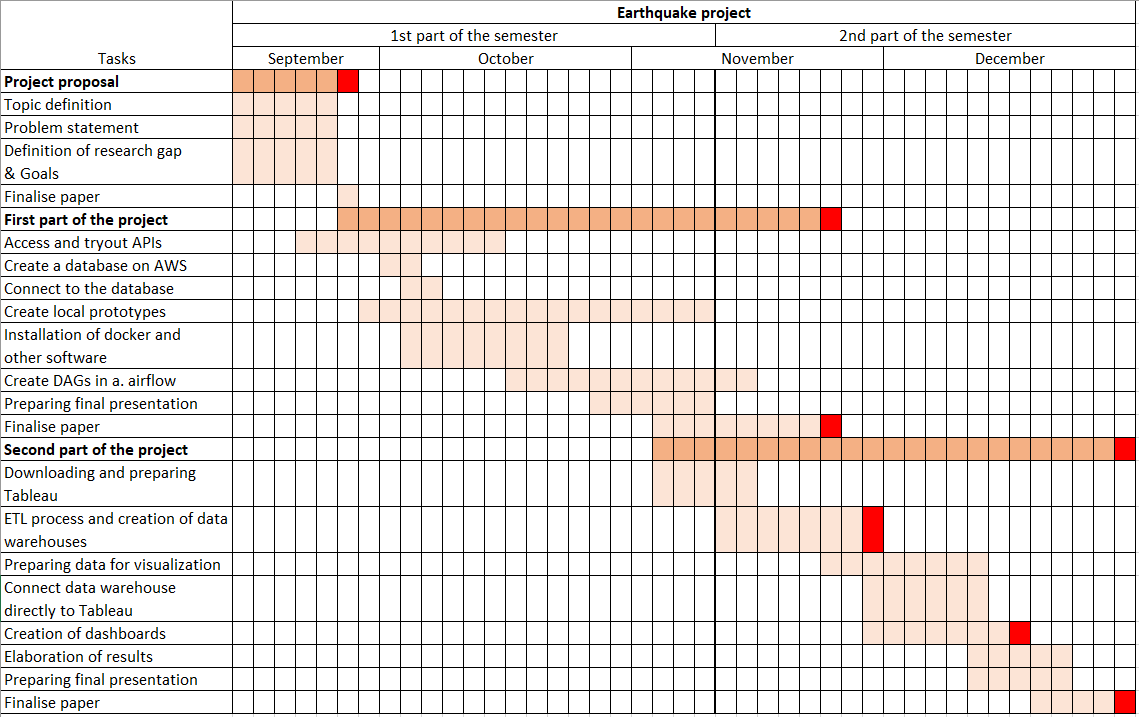
We also thought about an option to show the time component. A possible solution could be an animated timeline, similar to an example on the website «Our world in data», that shows the covid infections per million people by country over time. It has a play button on the bottom, that starts the visualisation of the pandemic over time. But it is also possible to drag and drop the slider to a certain point of time. Alternatively, an animated GIF could be an option as well to show the earthquakes and their corresponding Twitter reaction over time.



*Image 5: Animated timeline of covid infections on «Our world in data»*

# Project plan

The following gantt-chart provides a broad overview about the tasks that have been achieved so far and that are aimed to be accomplished throughout this project. The red points are showing important milestones in this project, such as the finalisation of the paper, the creation of data warehouse, dashboards and more.



# Collaboration

The project is driven by the three team members simultaneously. The Scrum framework helps with this: During regular meetings on site or online, current challenges and upcoming steps are discussed. Although everyone is driving their own sub-projects, great importance is attached to ensuring that all members come into contact with all technologies. This is not the most efficient way to reach the goal, but because all team members have little experience with the technologies presented in this module, it seems to be the best learning path.

Project organization files - namely the proposals, documentation and schedules - are stored on a shared folder in the Switch-Drive cloud environment and backed up regularly.

Code and data are stored and versioned in a Github repository. This allows the individual work steps to be documented and tracked as needed. The recommended steps covered in class were included, such as creating a .gitignore file. The README.md file provides a quick overview of the project.

# Project limitations

In keeping with the curriculum of this course, the focus will be on exploring new technologies, such as Apache Airflow, Amazon Web Services, Lambda Functions, DBT, and so on. In answering the research questions, some limitations will therefore be accepted.

For example, only tweets containing the word "earthquake" will be intercepted. Of course, there are also tweets in languages other than English, which then contain corresponding translations of this term. In addition, there are various synonyms or related terms, such as "shake" or "quake", which are also not included in the analysis.

Furthermore, the focus is not on sentiment analysis, because that would be out of scope of this course. With a short analysis, the potential of this methodology is hinted at, but further evaluations are postponed to a later date.

# Sources

## Literature

## Images

Figure 1-1: Swiss Re’s insured losses chart <https://www.swissre.com/media/news-releases/nr-20201215-sigma-full-year-2020-preliminary-natcat-loss-estimates.html>

Image 1: Earthquake dashboard by USGS, <https://blog.twitter.com/en_us/a/2015/usgs-twitter-data-earthquake-detection>

Image 2: CRISP-DM, Slides from Lecture 2, DWL1

Image 3: High level architecture

Image 4: Wikipedia Map of Earthquakes 1900 - 2017

<https://commons.wikimedia.org/wiki/File:Map_of_earthquakes_1900-.svg>

Image 5: Animated timeline of covid infections on «Our world in data»

<https://ourworldindata.org/explorers/coronavirus-data-explorer?tab=map&zoomToSelection=true&time=2021-07-09&facet=none&pickerSort=asc&pickerMetric=location&Metric=Confirmed+cases&Interval=7-day+rolling+average&Relative+to+Population=true&Align+outbreaks=false&country=USA~GBR~CAN~DEU~ITA~IND>

brandwatch, 60 Incredible and Interesting Twitter Stats and Statistics, 2020

<https://www.brandwatch.com/blog/twitter-stats-and-statistics>

# Declaration

We hereby confirm that we independently have written this paper. Any text passages which were not written by us are quoted as citations and specific references to the origin are made.

All used sources (including images, graphics, etc.) are included in the Bibliography.

Lucerne, 12th of November 2021

**Signature:**



Sandro Huber Thomas Schwendimann



Lea Senn

# References

Avramakis, E., Anchen, J., & Raverkar, A. K. (2019, January). Digital ecosystems: extending the boundaries of value creation in insurance. Swiss Re Management Ltds.

Elaine (2015, October 7). How the USGS uses Twitter data to track earthquakes. <https://blog.twitter.com/en_us/a/2015/usgs-twitter-data-earthquake-detection>

Earle, Paul. Bowden, Daniel. Guy, Michelle (2011). Twitter earthquake detection: Earthquake monitoring in a social world. <https://pubs.er.usgs.gov/publication/70006356>

SwissRe. (2020, December 15). SwissRe Institute estimates USD 83 billion global insured catastrophe losses in 2020, the fif. Retrieved from <https://www.swissre.com/media/news-releases/nr-20201215-sigma-full-year-2020-preliminary-natcat-loss-estimates.html>

USGS (2010, May 24). Earthquake Facts and Statistics. <https://web.archive.org/web/20100524161817/http://earthquake.usgs.gov/earthquakes/eqarchives/year/eqstats.php>

# References

Avramakis, E., Anchen, J., & Raverkar, A. K. (2019, January). Digital ecosystems: extending the boundaries of value creation in insurance. Swiss Re Management Ltds.

SwissRe. (2020, December 15). *SwissRe*. Retrieved from https://www.swissre.com/media/news-releases/nr-20201215-sigma-full-year-2020-preliminary-natcat-loss-estimates.html

# Appendix