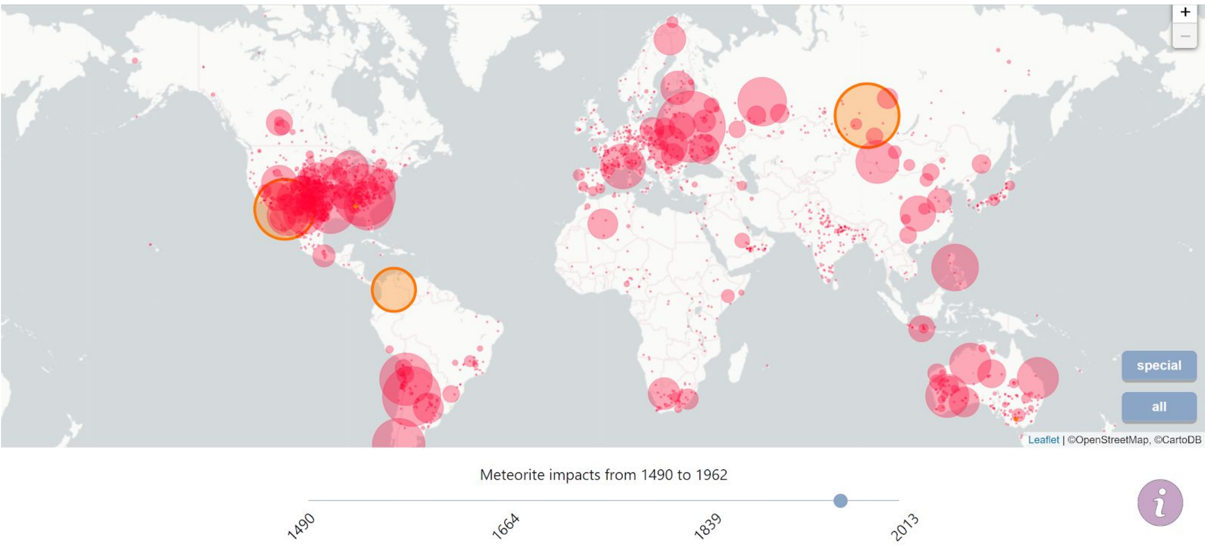

Meteorite Landings

Documentation

IVIS - HS19

By
Alain Baumann
Sebastian Renner

Content

1. Lead-In / Project Idea

Over the centuries thousands of meteorites have been recorded by humanity. The further our species developed and our technologies advanced the better we got at documenting them. It is estimated, that around 20’000 meteorites hit earth every year. With a circumference of 40’007 km and an area of 510,064,472 km^2 how big is the chance of a human getting hit by a meteorite?

While this certainly is an interesting question the answer is so hypothetical that it basically does not matter. So instead we want to show all the remarkable incidents during meteorite landings that have been recorded over the last 500 years. These include people, animals, houses and cars that have been hit or affected by meteorite strikes. We’ll let the observer get a feeling for himself of the insignificance of such a small probability.

2. Dataset

2.1. Source

The baseline for our project is a dataset about meteorite impacts on earth provided by NASA: https://data.nasa.gov/Space-Science/Meteorite-Landings/gh4g-9sfh

The raw data comes in form of a CSV-file. It consists of around 45.7k rows, each one of them representing a meteorite impact on earth. The columns build the attributes for each meteorite:

Column name	Data type
name	String
id	Integer
recclass	String
mass (g)	Decimal
fall	String
year	Datetime
reclat	String
reclong	String
GeoLocation	Location

2.2. Data quality

We ensured that the overall quality of the data was sufficient. A first indicator was, that all other datasets regarding this topic referred to the one from NASA. It seems that everybody else trusts in the authenticity of the data that NASA provides and so do we.
Over the course of the project we could make sure of that ourselfs. When we were researching meteorites with an interesting backstory we could compare if the information we found matched with the attributes of our meteorites from the dataset.

3. Data transformation

3.1. Tools

Although we worked with an official dataset, there was still a lot of work to do as many entries weren't formatted properly, certain values were missing or had nonsense content. For cleaning up this big chunk of data we used Trifacta[1] a data transformation tool we got to know during an exercise in class.

3.2. Cleanup

There were many iterations necessary before we reached our final dataset. At first, we just eliminated entries where values like geolocation, year or mass were missing, as all of these attributes are critical for our data visualisation. Also we tried to fix mismatches where possible.
After applying this first filter, we determined that many different meteorites where located at the same location - GeoLat == (0,0). This couldn't be a coincidence. It is just another way to show that the data collector didn't know where the meteorite landed. So we omitted all entries with this particular geolocation. We also realized that the year column was a datetime field, meaning that in addition to the year it also contained a date and a timestamp. Almost all entries were dated to the 1st of January at 00:00:00. Due to this discovery we only extracted the year and changed the field type to an integer.
As the dataset was still quite big and caused annoying lags in development, we decided to delete the generic id and nametype column as we had no use for the generic id and the nametype column was always set to 'valid'.
When we first used the meteorites mass as the radius for its marker, we quickly saw that some values couldn't be realistic. A quick inspection gave as the answer: some entries used a comma instead of a full-stop as an indicator for a floating point. A quick normalization took care of this minor issue.
Last but not least we formed clusters out of meteorites which descended from the same meteor and landed roughly on the same spot. This deduplication made the application way faster at no loss of information.

[1] https://www.trifacta.com/

4. Technology Stack

We used a single site web page implemented in Javascript with HTML as markup language and CSS for styling. The core component of our site is displayed on the leaflet map engine, which provides seamless integration of a number of independent map providers like OpenStreetMap or the Environmental Systems Research Institute. We used the simplest map design by OpenStreetMaps, which focuses mainly on country borders.

Supported Browsers

While developing we focused mainly on the most used desktop web browsers:

- Chrome: 67 %
- Firefox: 8 %
- Edge: 6 %

Project Structure

On the most upper level there is the index.html file. It will load everything it needs from the content of the supplied meteorite-landings project folder. Most notably are the files JSON.js which contains all information to the meteorite landings and the file: meteorinfo.js which contains the special information to the select few interesting impacts.

5. Visualization

To let the observer get a good idea of where these meteorites all touched down we visualized them on a map. How big the mass of a meteorite is, influences the size of the marker that appears on the map. Since there were a lot of meteorite markers the rest of the website had to look as clean and simple as it could get, or it would hinder the users experience. This is also why we chose a map overlay that only shows the country borders and doesn't add additional complexity.

Although red is a risky color since a lot of colorblind people have a lot of trouble distinguishing it, we chose it as our color for the markers. Since there aren't a lot of different colors on the map, the red acts as an accent and helps to highlight the impact locations as important. Es a secondary color we chose orange. Another color that really stands out. It is meant to mark some of the impact locations as interesting and prompt the user to hover and click the marker.

As mentioned before, we really want the impacts to draw the observer's eye. All other components had to be a lot simpler and inconspicuous. This influenced the color and thickness of the slider as well as the buttons on the map. We settled on a light blue to be just interesting enough for the observer to look at and notice that there is functionality behind it. In the case that the observer wouldn't know what the site is about and start to scan it visually, we put in an infobutton that distinguishes itself through its shape and color from all other buttons on the site. It is supposed to catch the observer's eye and make him click the button.

The rule, less is more, also applies to the infobar which slides in from the left. The movement and the fact that it covers half the screen makes it noticeable enough and eliminates all need for a special color or shape.

6. Interaction

To not overcrowd the visualization we decided to show the basic information about a meteorite only if you hover over it in form of a pop-up. All the meteorites, where additional information is deposited, are colored in orange. If you click on one of those a sidebar will open. Besides the basic information it will display the story behind this particular meteorite and a picture, either of the meteorite itself or relating capture of the incident.
As it may be too annoying to search for all the orange markers, we implemented a layer, which will only show all meteorites with an interesting backstory. This layer can be reached by clicking on the *special* button on the bottom right of the map. Clicking on the *all* button will switch back to the original view with all meteorites.

With the slider on the bottom of the screen you can control the interval from our first recorded meteorite impact (in the year 1490) to the year of your choice. This allows you to filter the visualization up to certain events (like when meteorite impacts where recorded by local interest groups or when NASA was founded).

7. Reflection

While developing we certainly learned a lot of new things. Probably one of the most important lessons we had to learn was that handling a big amount of data is very taxing on your systems ressources. This creates a lot of downtime when you have to wait for the site to reload, or your development environment slows down so much that developing becomes a nightmare.
Furthermore we realized that changing a small detail on a site can change the whole look and feel of said site. For us this happened when we changed the design of your slider to be less thick. Suddenly the whole site just started to look a lot more elegant.