Meteorite Landing Project

Exploring the sky

In this document we present our work designing, preparing, and implementing a visualization on registered meteorite landings on the earth's land surface.

Hoba: the largest registered meteorite to land on the earth surface

Table of Contents

Team members

Implementer team	Christian Pala, Andrea Wey
Customer team	Dyuman Bulloni, Paolo Gübeli

Audience and questions

Audience	The main target users are journalists and their audiences. They may want to use the dashboard to generate infographics, could be for a specific audience (ex: National Geographic readers) or the public (ex: reporting in case of a real event).
Questions	The dashboard aims to answer the following questions: - How many meteorites are landing yearly on the earth and are there any trends? - Where do they land? - Do they land in heavily populated areas? - What are they made of? - How big are they? - Did we manage to spot more meteorites falling as the years go by? - How are meteorites classified?

Tasks	
Map	By means of a map we show the geospatial distribution of meteorites, points on the map are characterized by color and shape to better explain the size difference between the rocks. By hovering over an observation the details are provided, and by clicking on it we filter out the rest.
Bubble chart	This graph helps us categorize the different meteorite types using the size channel. We can divide into clusters for comparisons, and by means of a filter we can also see compositions of the different classifications of meteorites. By hovering over a group, we get a brief scientific description of the type of rock.
Scatter plot	We show the trend over the years of meteorite landings with a graph. We also have a comparison between fell and found meteorites. All this together gives us a nice distribution of the landings over the years.
Dashboard	Our dashboard supports categorizing and clustering We can filter everything for a certain period of time, for a determined meteorite mass range and by asteroid type classification. By hovering over an observation of a meteorite we always get details on demand, and by selecting it we display its characteristics on all three graphs.

Description of dataset and data	
Dataset	The main dataset is hosted online on NASA's Open Data Portal:
	The observations have been collected by the Meteoritical Society: https://meteoritical.org/
	For demographic data, we planned to use the World Bank API: http://api.worldbank.org/v2/en/indicator/EN.POP.DNST
	For the background map we used: - GPW v4 Population density 2020 - ESRI Administrative Boundaries level 1 - cartographic national boundaries https://sedac.ciesin.columbia.edu/ The last two map layers are important to show the Antarctic.
Main dataset	**Variables:** • **name**: the name of the meteorite (typically a location, often modified with a number, year, composition, etc) • **id**: a unique identifier for the meteorite • **nametype**: one of: -- *valid*: a typical meteorite -- *relict*: a meteorite that has been highly degraded by weather on Earth • **recclass**: the class of the meteorite; one of a large number of classes based on physical, chemical, and other characteristics • **mass**: the mass of the meteorite, in grams • **fall**: whether the meteorite was seen falling, or was discovered after its impact; one of: -- *Fell*: the meteorite's fall was observed -- *Found*: the meteorite's fall was not observed • **year**: the year the meteorite fell, or the year it was found (depending on the value of **fell**) • **reclat**: the latitude of the meteorite's landing • **reclong**: the longitude of the meteorite's landing • **GeoLocation**: a parentheses-enclose, comma-separated tuple that combines **reclat** and **reclong**

Related works

NASA	The American space agency NASA has a website on the topic: https://data.nasa.gov/Space-Science/Meteorite-Landings/gh4g-9sfh
The meteorological society	The meteorological society is responsible for collecting most of the publicly available data: https://meteoritical.org/
Tableau viz of the day	Our professor kindly sent us this visualization which represents a similar dataset: https://public.tableau.com/app/profile/yash.sakhuja/viz/MeteoriteLandings1583-2013/Meteorites
Map formats	From the lecture 12 on Maps: - Dot map on slide 22 - Choropleth map on slide 25

Design phase

FDS 1: Our initial brainstorming where we explored different kinds of visualizations.

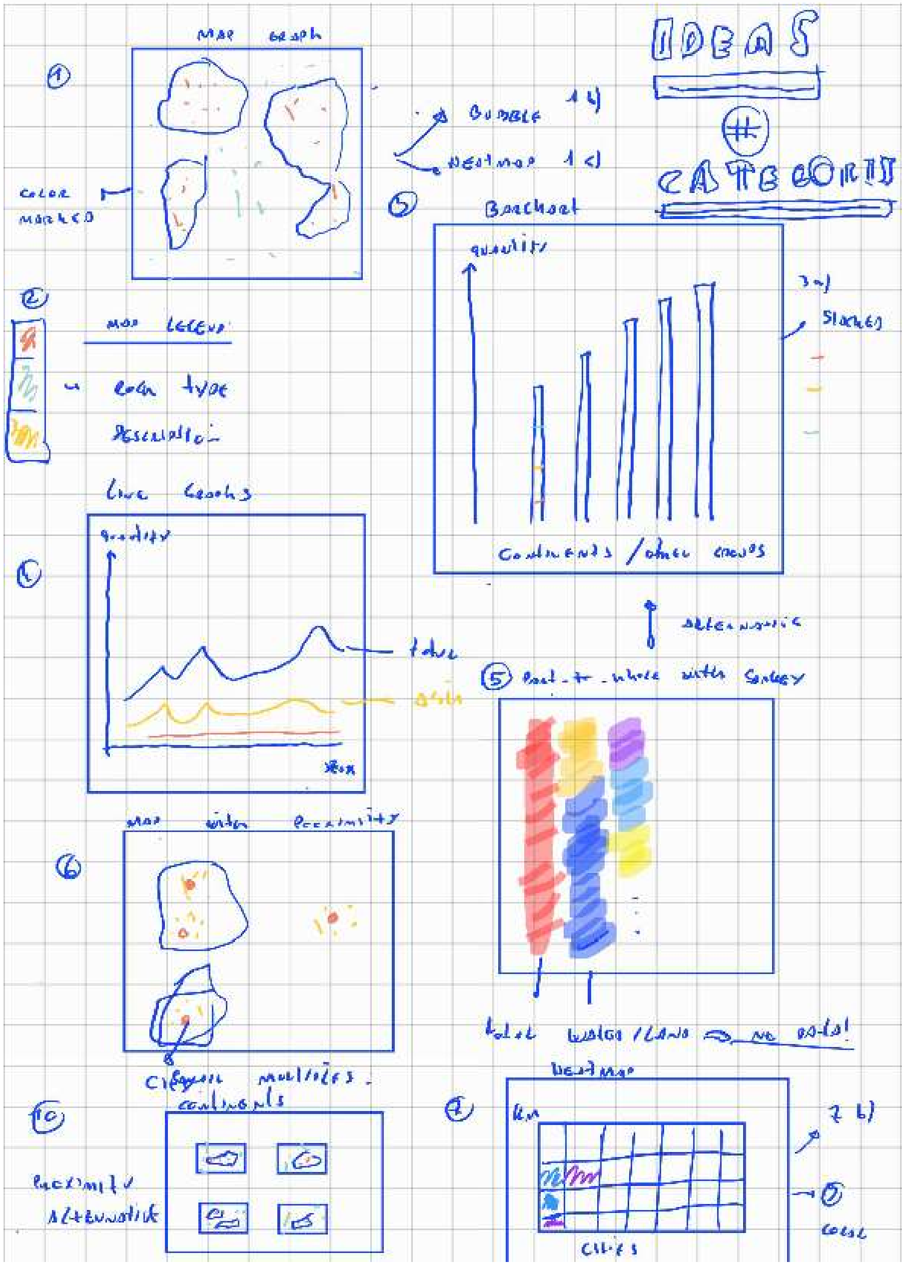

Figure 1: FDS 1

FDS2: First client proposal, trying to avoid using a map. Identified asteroid type as an important variable.

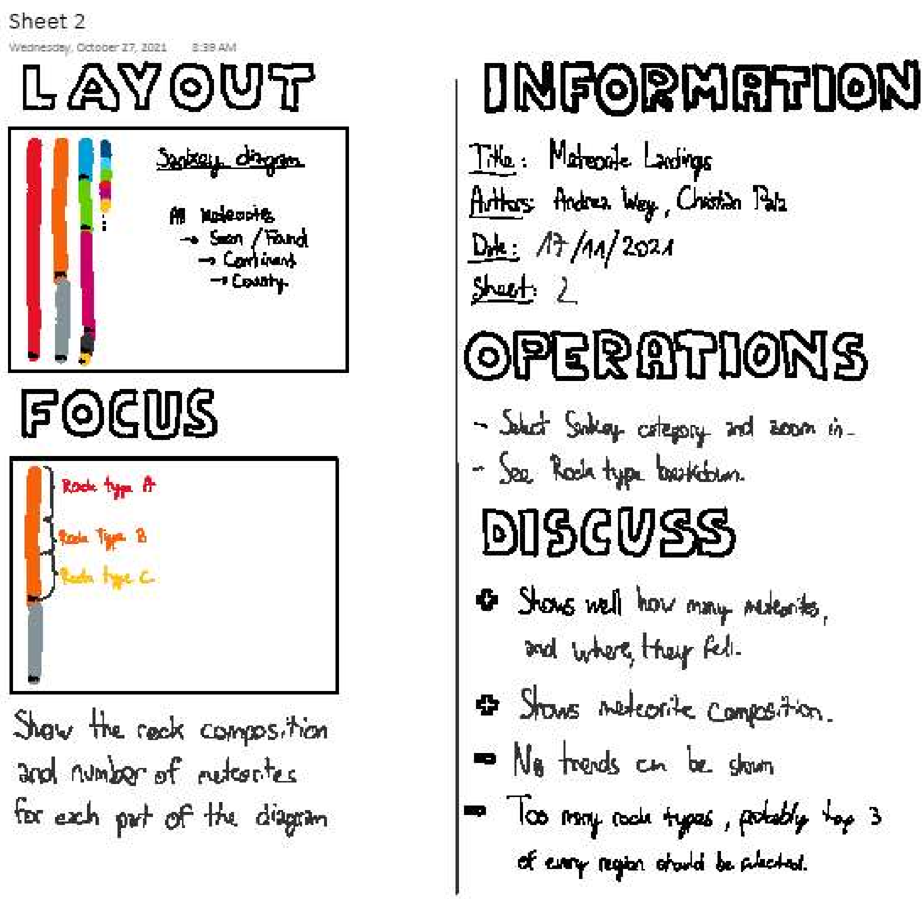

Figure 2: FDS 2

FDS3: Second client proposal, identified map as essential. Further studies on classifications with violin plots, a nice idea we substituted with a bubble chart.

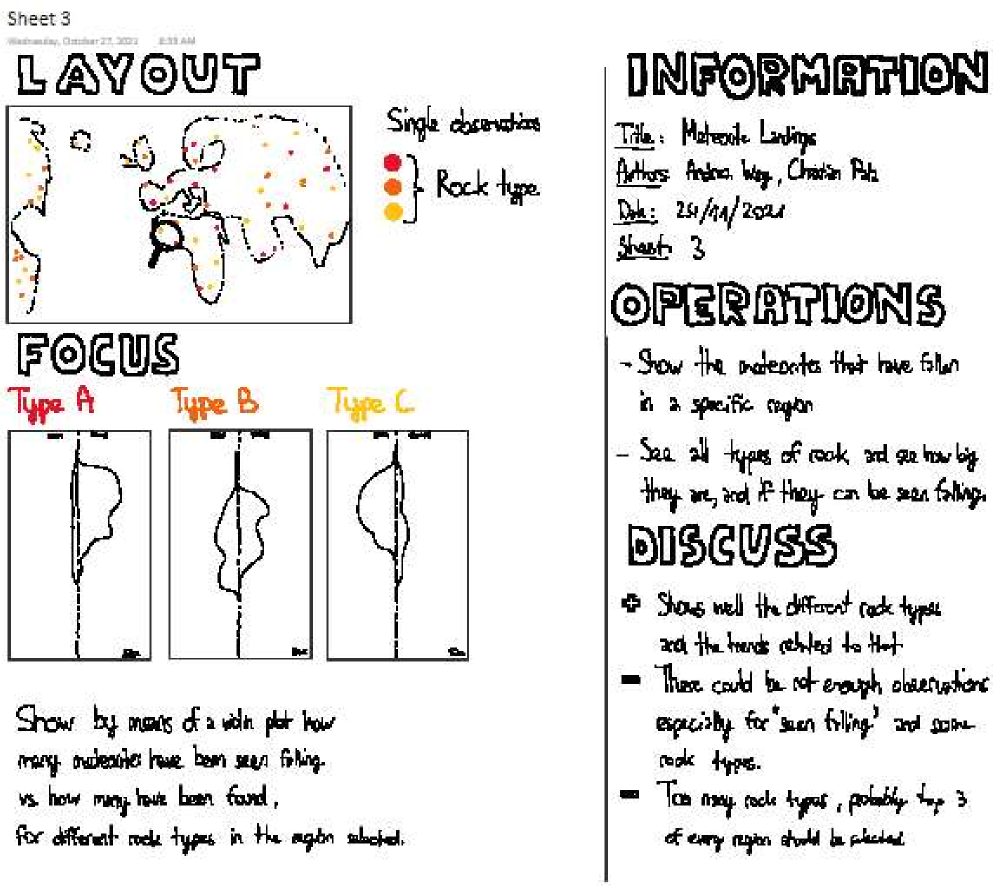

Figure 3: FDS 3

FDS4: Started to work on time series with the bar chart of the years, on the population density with the city proposal and on dashboard functionalities.

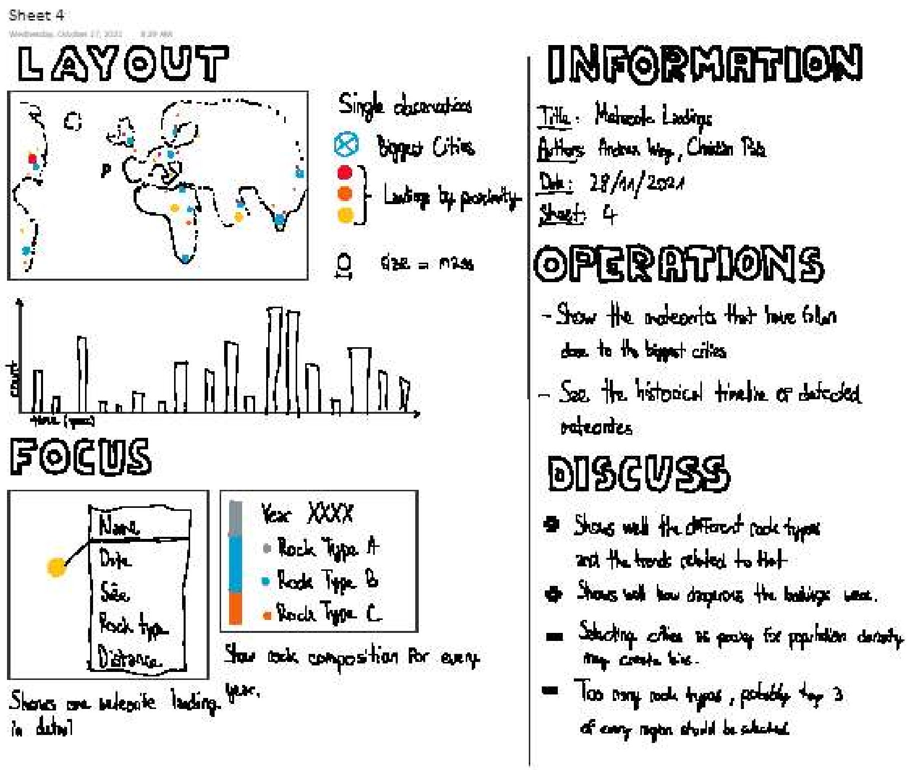

Figure 4: FDS 4

FDS5: Identified the choropleth map as our design choice, elaborated on the dashboard functionalities, we explain in the implementation why we chose a scatterplot.

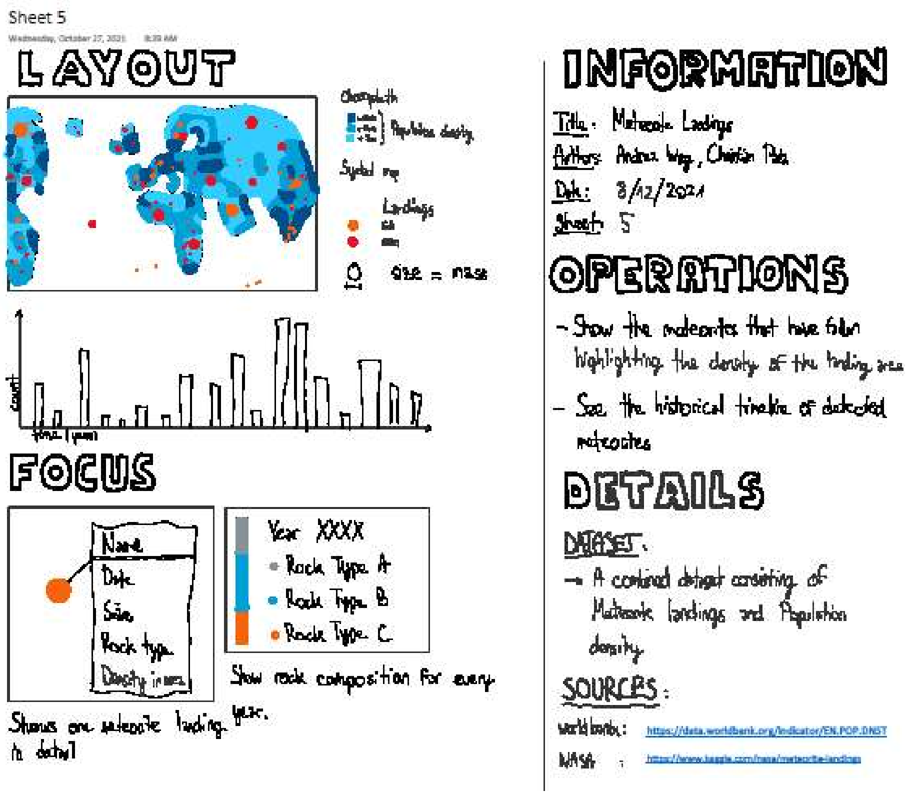

Figure 5: FDS 5

Implementation

Our first step was to download and pre-process the meteorite landing dataset from the NASA API. Below a snippet of the Jupyter notebook we worked on:

```
100 print(df.shape)
    df.dtypes

    (45716, 10)
100 name           object
    id              int64
    nametype       object
    recclass       object
    mass (g)      float64
    fall           object
    year           object
    reclat        float64
    reclong       float64
    GeoLocation    object
    dtype: object

101 df.head()
```

	name	id	nametype	recclass	mass (g)	fall	year	reclat	reclong	GeoLocation
0	Aachen	1	Valid	L5	21.0	Fell	01/01/1880 12:00:00 AM	50.77500	6.08333	(50.775, 6.08333)
1	Aarhus	2	Valid	H6	720.0	Fell	01/01/1951 12:00:00 AM	56.18333	10.23333	(56.18333, 10.23333)
2	Abee	6	Valid	EH4	107000.0	Fell	01/01/1952 12:00:00 AM	54.21667	-113.00000	(54.21667, -113.0)
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	01/01/1976 12:00:00 AM	16.88333	-99.90000	(16.88333, -99.9)
4	Achiras	370	Valid	L6	780.0	Fell	01/01/1902 12:00:00 AM	-33.16667	-64.95000	(-33.16667, -64.95)

Figure 6: Jupyter notebook preprocessing code snippet.

See the attachments for the full code. Besides dealing with values stored in the wrong format for our visualization, we also noticed how the observations were divided by detection method, either the meteorite was seen falling or it was later found on the ground. One of our client requests was to track the yearly trend of meteorites falling present in the dataset, this inclusion in the same year of both fallen and found meteorites is a strong confounder and we therefore decided to separate the yearly trends in the visualization.

The client also required a visualization encompassing both the meteorite landings and some measure of population density, with the goal of showing how potentially dangerous undetected meteorite landings can be.

Our first attempt, since we had the latitude and longitude coordinates available, but not the addresses of where the meteorites landed, was to use the python library Geopy and an API, like Google or later Mapbox, to reverse geolocate the landing sites. This is a code snippet of our work:

```
65  from typing import Tuple
    from typing import List
    import time


    mapbox: MapBox = MapBox('pk.eyJ1IjoidXR0aSIsImEiOiJja3llNXJzcHcwNGhnMm5wbWszYXc3cDVvIn0.jmunv4h49Qe6bqSvRhqUdw')
    points: pd.DataFrame = df[['latitude','longitude']]
    point_list: List[Tuple] = [tuple(x) for x in points.to_numpy()]

66  country = list()

    def reverse_geo_locate():
        for i in range(0, len(point_list)):  # len(point_list)):
            # do not exceed API calls per second
            time.sleep(0.005)
            with open('countries.csv', 'a', encoding='utf-8') as f:
                location = mapbox.reverse(point_list[i])
                country.append(location if location else ("no address","no country"))

67  new_df = pd.DataFrame(country)

68  new_df.to_csv('countries_last_attempt')
```

After checking the quality of the geo-located data, in particular the number of missing nations, we decided to use another strategy to represent the population density.

Figure 7: Batch geolocating with Mapbox code snippet.

Google's free API proved to be too slow, following professor Mazza's suggestion we switched to Mapbox and managed to download the data. The quality however was very poor, hence we decided to look for other solutions.

The second attempt was to make a custom map on Mapbox using a GeoJSON file with the world boundaries, which we found on the World Atlas Github:
https://github.com/topojson/world-atlas
and integrate it on http://geojson.io/#map=2/20.0/0.0 with the information on the countries density from the World Bank API we used for Activity 13:
http://api.worldbank.org/v2/en/indicator/EN.POP.DNST.

Below a snippet of the data pre-processing for this task:

Preprocessing of world population dataset

import the dataset

```
wp: pd.DataFrame = pd.read_csv("world_population.csv")
```

brief exploration

```
print(wp.shape)
wp.dtypes
```

```
(266, 65)
```

Figure 8: World Population Dataset pre-processing.

This also did not work out well due to the size of the files in question.

During our research Andrea noticed that the open geospatial consortium: https://www.ogc.org/standards/wms already provided maps with population density, together with many other options, and that we could therefore use a ready-made solution for the density map, the data provided is also relatively new (2020) so that's what we finally settled on.

Below the first visualization we created integrating meteorite landings and population density:

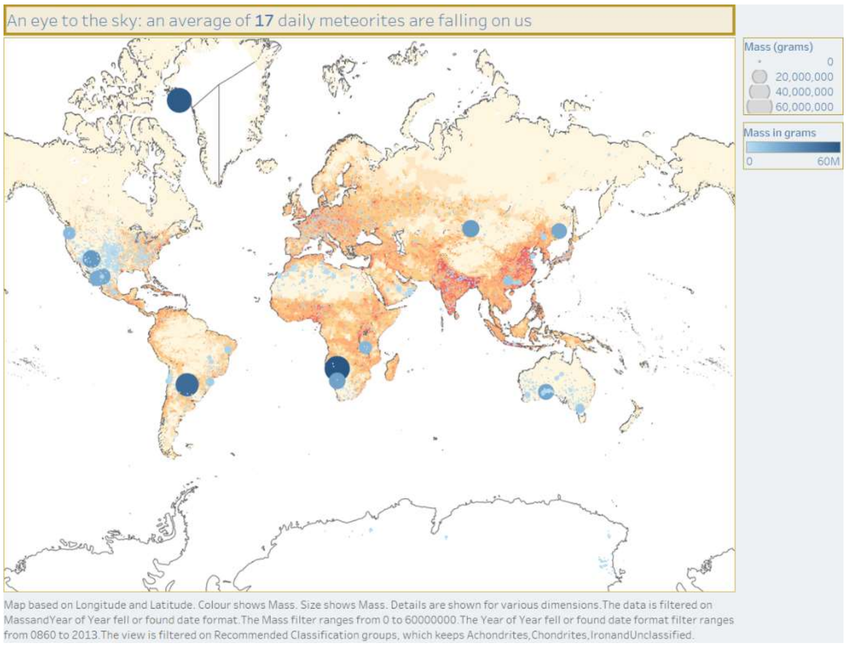

Figure 9: Overlapped meteor landings and density maps

Our clients were very satisfied with the visualization.

The next part of the project we explored was the way the meteorites are classified and how we could integrate it in the visualization, we mentioned this to the clients during the design phase, and although this was not a requirement, they were happy with the suggestion. We used a simplified version of Rubin's meteorite catalog (A., 2021) to group meteorites by category which we used for the color selection, we then arranged them by circle size based on the number of detected landings compared to their specific classification, this is the result:

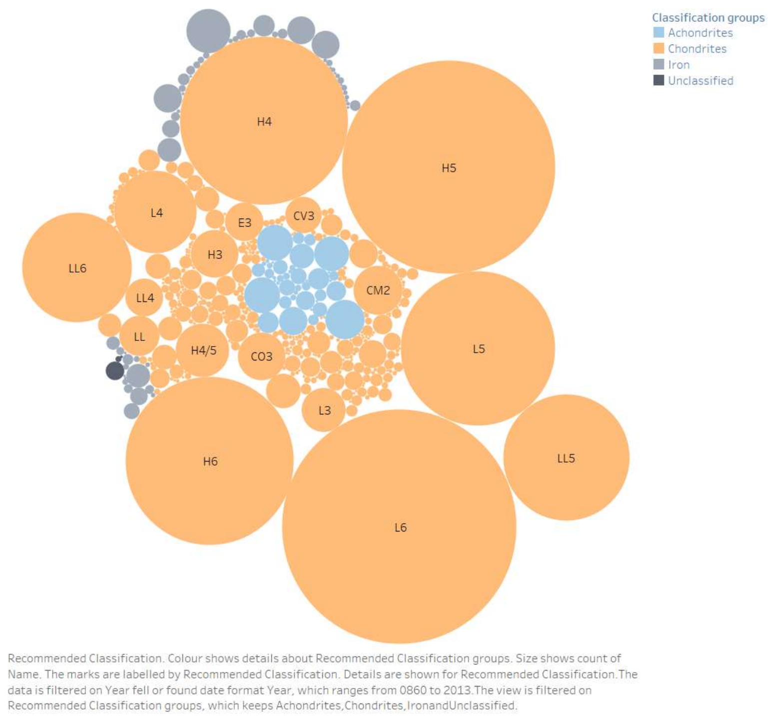

Figure 10: Meteorites classification overview and grouping via bubble chart.

We then created a calculated field to use in the tooltip to explain what each classification means:

- Achondrites, early asteroids and meteorites in the solar system which took part in planet formation and therefore lost their chondrites, planetary building blocks.

- Chondrites, similar to achondrites but which did not participate in planetary or asteroid formation and therefore conserve their chondrites, used to study how planets and asteroids form in early solar systems.

- Iron, mineral rich asteroids which comprise most of the heavy and dangerous meteorites to land on our planet.

- Unclassified meteorites, those were just a few we could not clearly put in one of the other three groups or which was already marked in the dataset as unknown.

The final separate visualization to show is the one mentioned earlier, where we separated the fallen and the found meteorites by year to satisfy our clients request regarding a time series on falling meteorites with two scatter plots describing the yearly trend, using some elements of gestalt theory like similarity, proximity and in the upper part of the chart continuity, which makes the single observations look like a continuous line.

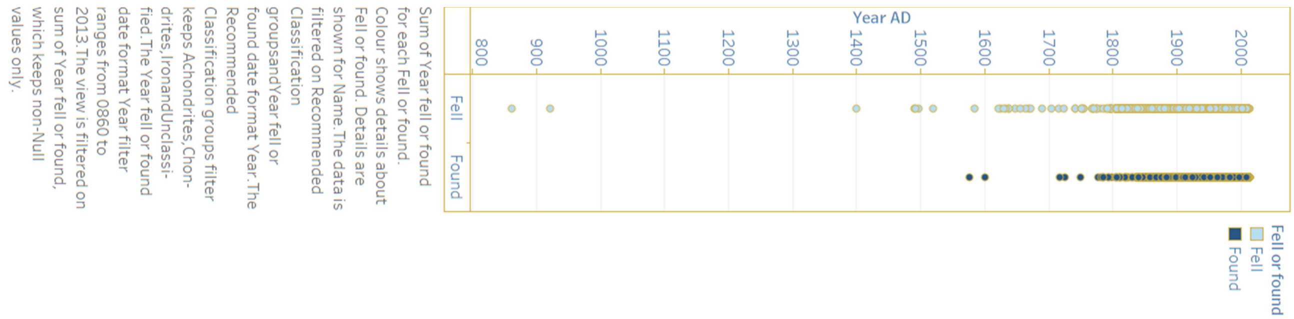

Figure 11: rotated view of fell and found meteorites with their timelines.

The view is rotated to keep the document readable. We debated whether to include this visualization or a more standard bar chart like the one below:

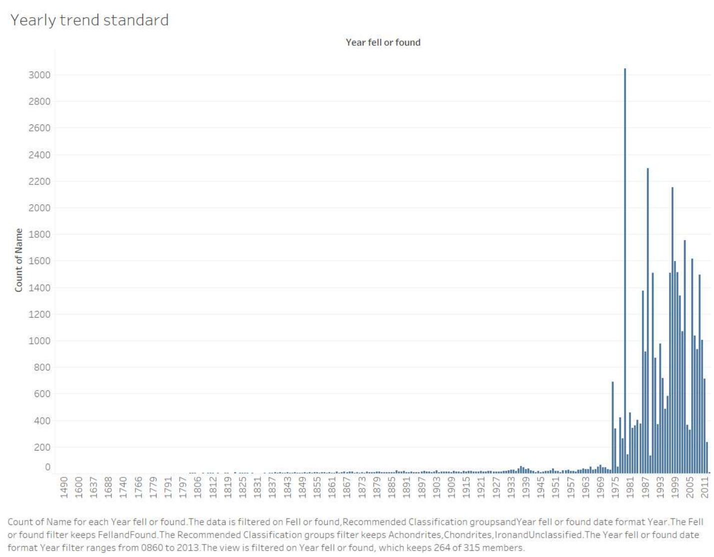

Figure 12: Bar chart of yearly meteorite landings.

Both have their pros and cons, we like how the scatter plot clearly separates fell and found, we also found a nice way to integrate this visualization in the dashboard, allowing a user to see graphically were in the timeline the meteorite they are interested in lies. We do however lose a clear view of the number of meteorites per year, which instead is obvious with the alternative visualization.

Finally, we present the dashboard we created putting together all our work:

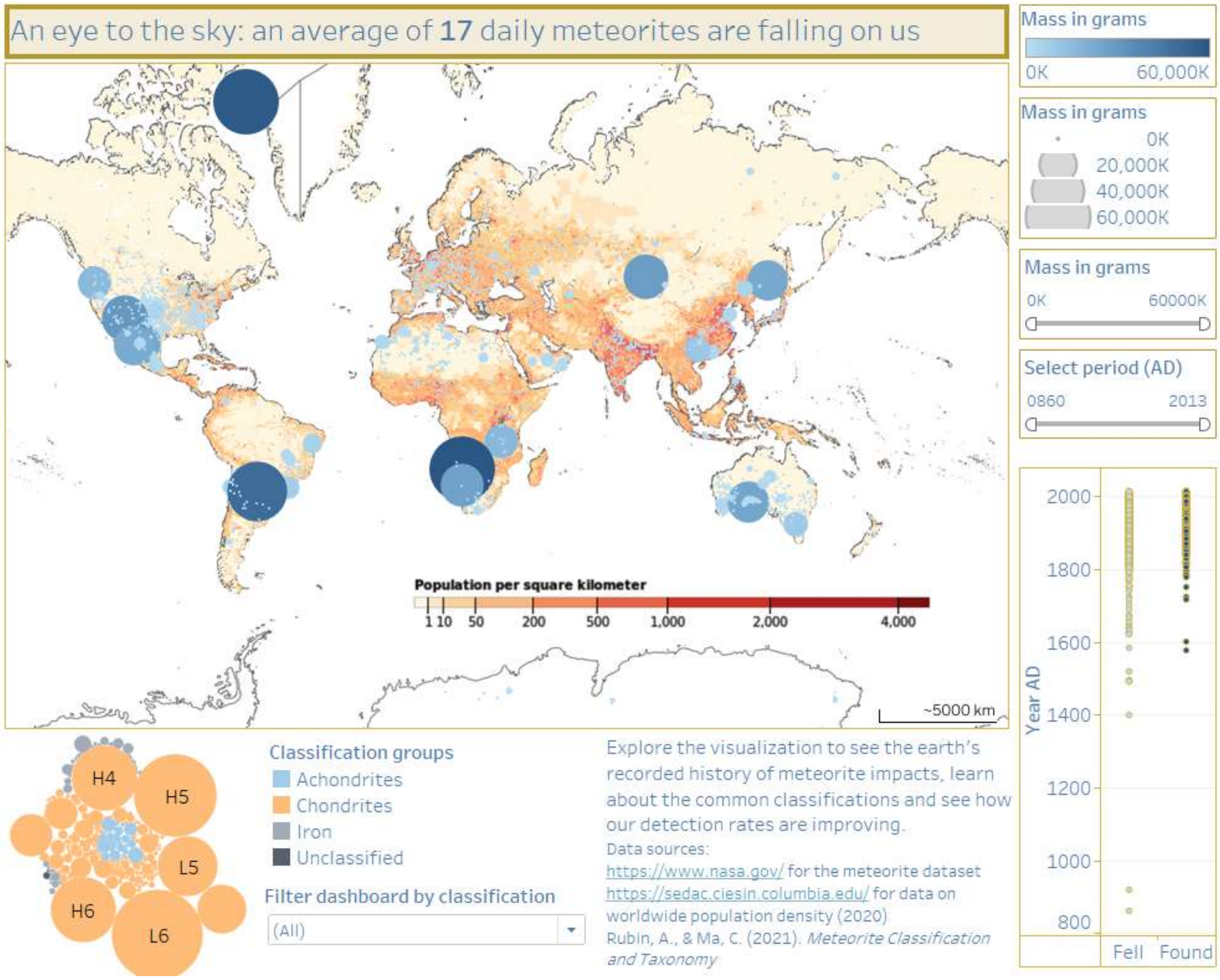

Figure 13: Meteorite landings dashboard.

We connected all visualizations and filters, so if the client is interested in examining a single meteorite, they will be able to graphically see when it fell, which classification it belongs to and what that means. Clients are also able to filter by period of interest and mass of the meteorites, features that go beyond what the client requested but add in our opinion to the usability of the dashboard. We also produced informative tooltips for all meteorites with references and sources enabling someone interested in the topic to delve more into the aspect they are interested in. We selected the title based on the client's request to make it eye catching. Finally, we tried to provide a uniform color scheme in the way we presented the information, going for a color blind friendly palette selection when possible.

A weak point, hard in our opinion to circumnavigate due to the number of information requested is the richness of color and how this may overload sensory perception, especially for people with color blindness, working with geospatial data should mitigate this concern as we are all quite familiar with maps, but still there may be room for improvement here.

Final conclusion

We received very positive feedback from the client team, we believe they will be able to use the visualization for the intended purposes.

We were able to address all the original goals and objectives, although the time series of falling meteorites is not as clear as we would like it to be, an animation could be a solution for this problem.

We struggled a lot with the population density map, as noted in the project description, and on all the filters we wanted to provide the client with.

We enjoyed working together on the project, it allowed us to explore more advanced Tableau features, as well as applying our knowledge from the data visualization course and other modules.

Table of figures

Bibliography

1. Rubin, A., & Ma, C. (2021). Meteorite Classification and Taxonomy. In Meteorite Mineralogy (Cambridge Planetary Science, pp. 101-108). Cambridge: Cambridge University Press. doi:10.1017/9781108613767.007

2. R. Mazza. 12 - Maps. 13.12.2021. p-22-25

3. Y. Sakhuja. Meteorite Landings 1583-2013. Tableau Visualization of the day. Retrieved 15.01.2022