# Process book
# App data visualization

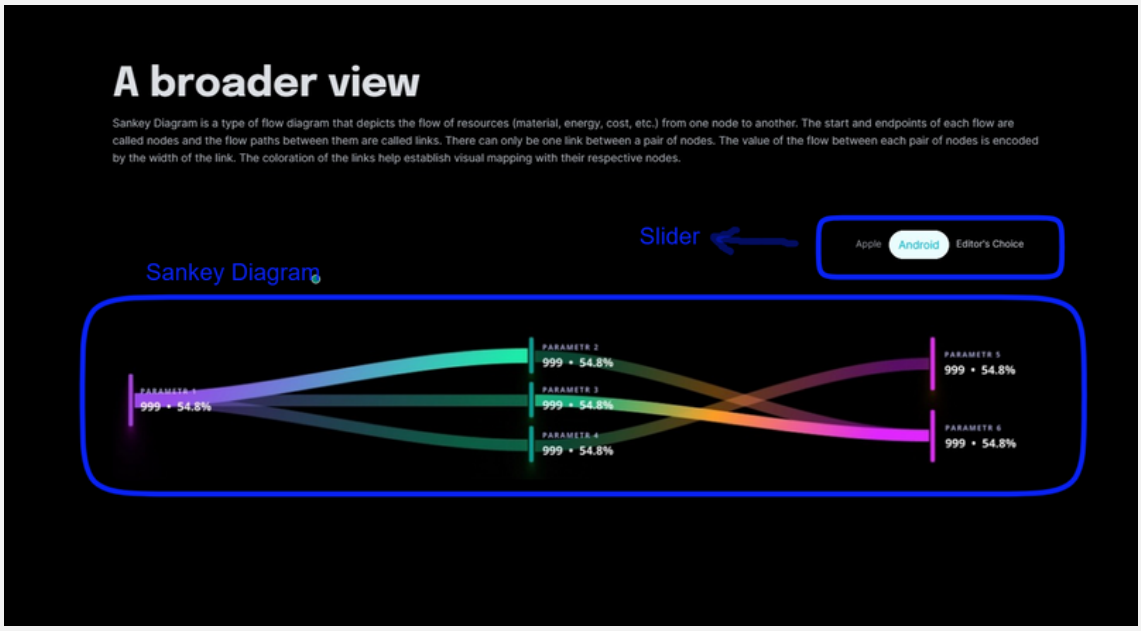
Elia Fantini, Konrad Litwiński, Adrien Vauthey

# Introduction

Nowadays, most people own a smartphone hence **mobile applications** are widely spread across the globe. The **App Store** and **Google Play Store** are widely recognized platforms where developers can showcase and distribute their mobile applications to millions of users worldwide. These stores host a vast array of apps spanning various categories, catering to diverse user interests and needs. Aside some articles that showcase the *Top 10 apps for better productivity* and so on, we wanted to dive deeper and show some **interesting visualizations** that are not only bar charts, and which are **comprehensive** to most people.

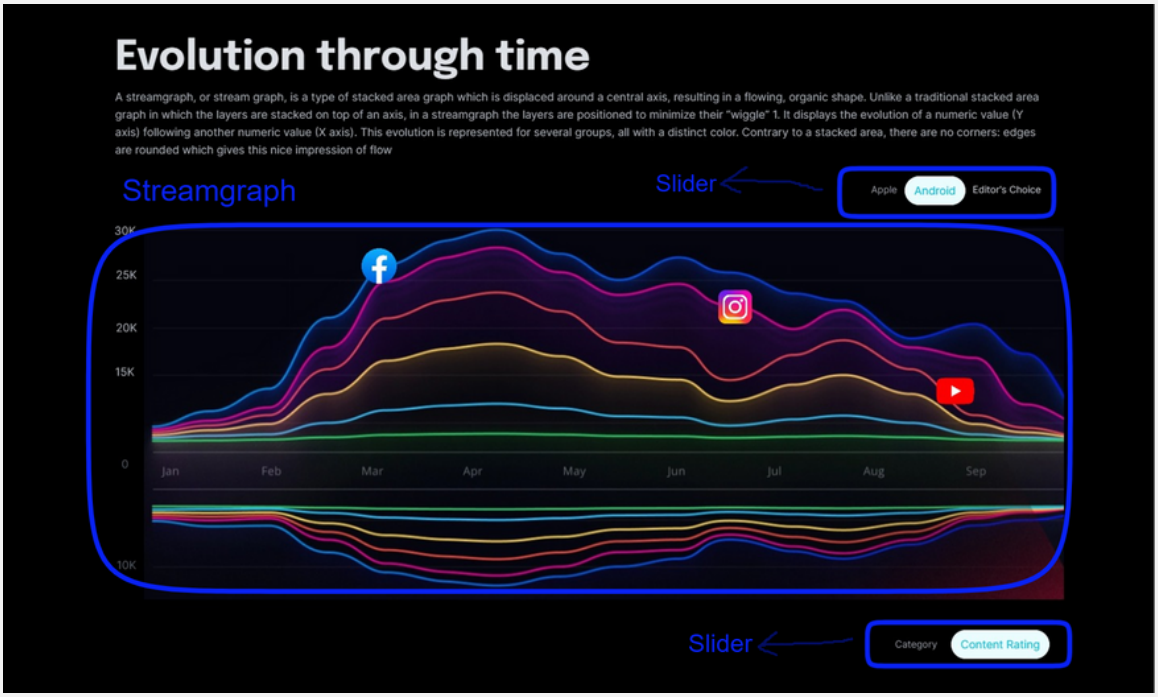
# Challenges

We have a **Sankey diagram**, which provides a wider and more general visual representation of the dataset and gives insight into the revenue model and success of different categories.




Our diagram ended up as expected, however we used google charts instead of d3.js for that particular chart we encountered some issues for its creation. Otherwise, we added a drop down menu to choose which app categories one wants to see. Each time the visualization is updated, either by changing categories or by changing the dataset, it will randomly take a color palette.

In the **stream graph**, we had a similar problem as for the treemap where we had to store each data into a variable, but it was not fast enough to use them when we load the website. Hence we created a function that is called at the beginning which load the data directly into the chart instead of taking the corresponding variables. For following updates of the chart, we can use the stored data without any problem.




The last part contains a handful of charts about each category. As previously stated, we had a problem to load data into d3.js on launch, for which we have found a workaround. We didn't have room to display other charts on the right of the **tree map**. We added them below.

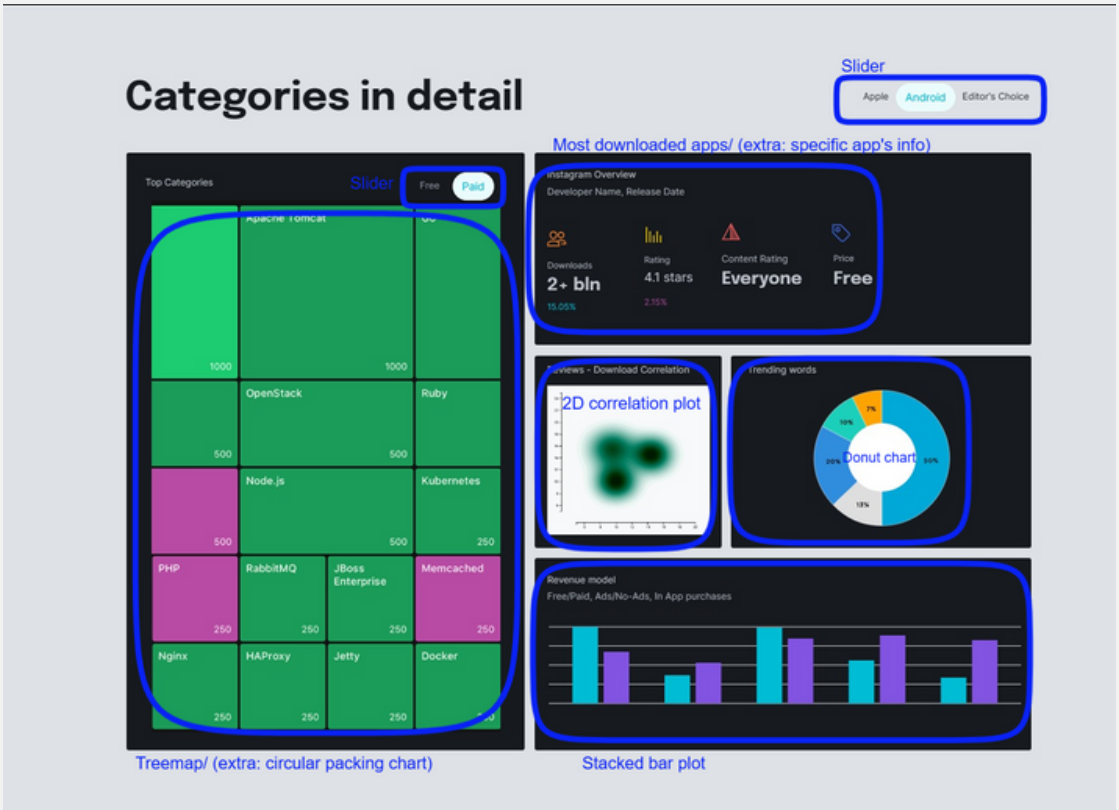
For the **tree map**, we had to create two helper functions when adding categories name on the tiles. The first one create a hidden *svg* with the text to compute its length in pixels given the font size. The second one compute the sub string of the category name in case its length is higher than the tile's width. Then we also had to check whether the height of the text fits in the tile's height.

We tried to create a smooth transition when creating the **tree map**, unfortunately we had some issues to display the name on each tile when the transition was done. Hence we only changed the opacity when changing between datasets, it is not perfect but it is less rough on the eyes.

As extra, we wished to implement a slider that change the color of the tile according to the percentage of free/paid application in that category. Unfortunately, it was a pain to deal with d3.js as we only managed to create a gradient, which was not what we expected to show.

Adding the **top apps** and **app details** did not take long to implement. We just had to reprocess the datasets to obtain the information we needed.

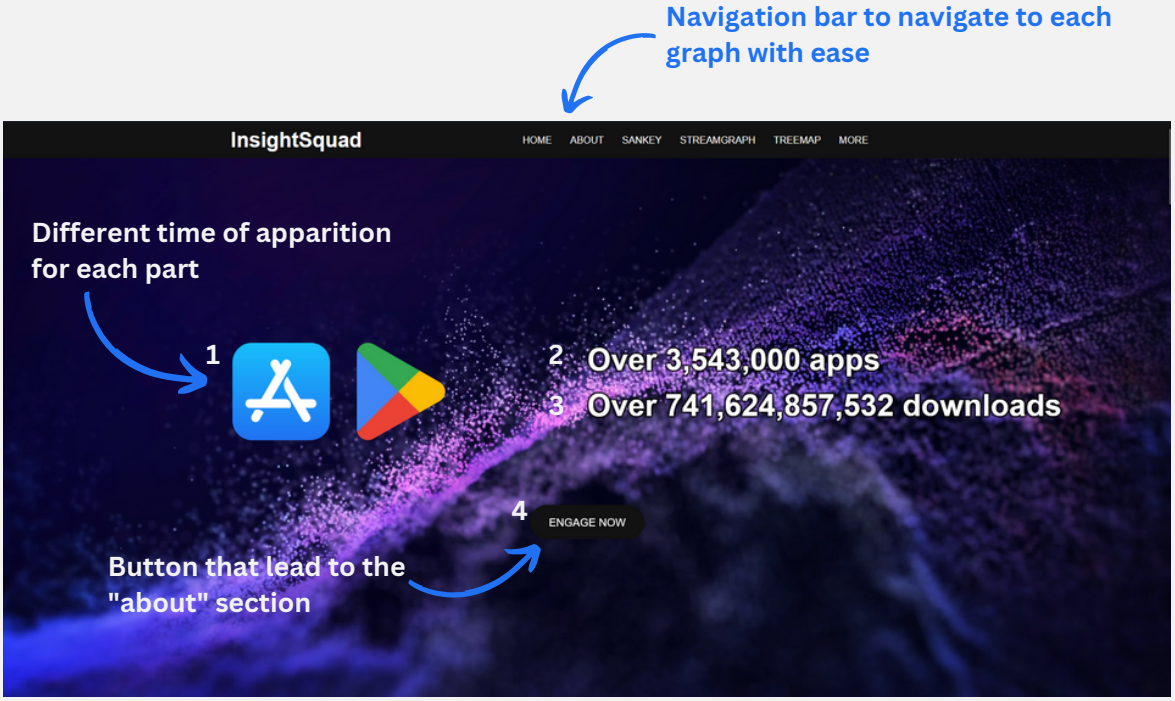
The creation of the **donut chart** with the ten most used words went pretty smoothly. We have a nice transition of each part when changing between categories and we added an opacity transition for the lines and labels. It also has a drop down menu to choose the category

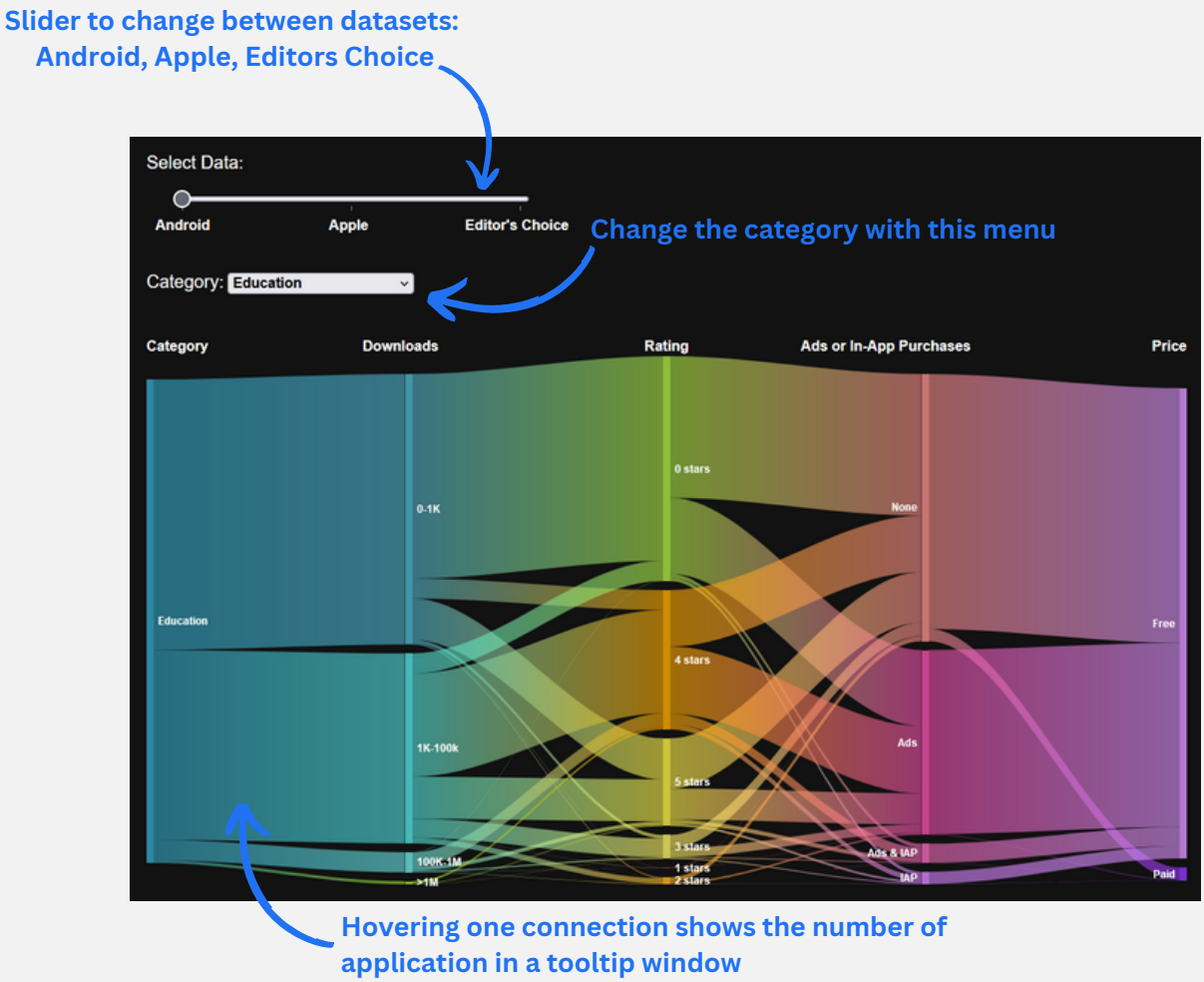



Initially, we had a **2D correlation plot** using **density**. But we had some issues when computing them, hence we used a **heat map** which does the work for what we intended to show. Due to the high disparity between the number of applications in each cell, the grid was mostly blank with small change of colors. We first try to take the logarithm of each value, but when we repeatedly change between datasets, the grid was filled with blue. Then, we use a logarithmic scale for the color and it worked fine while also showing the correct value.

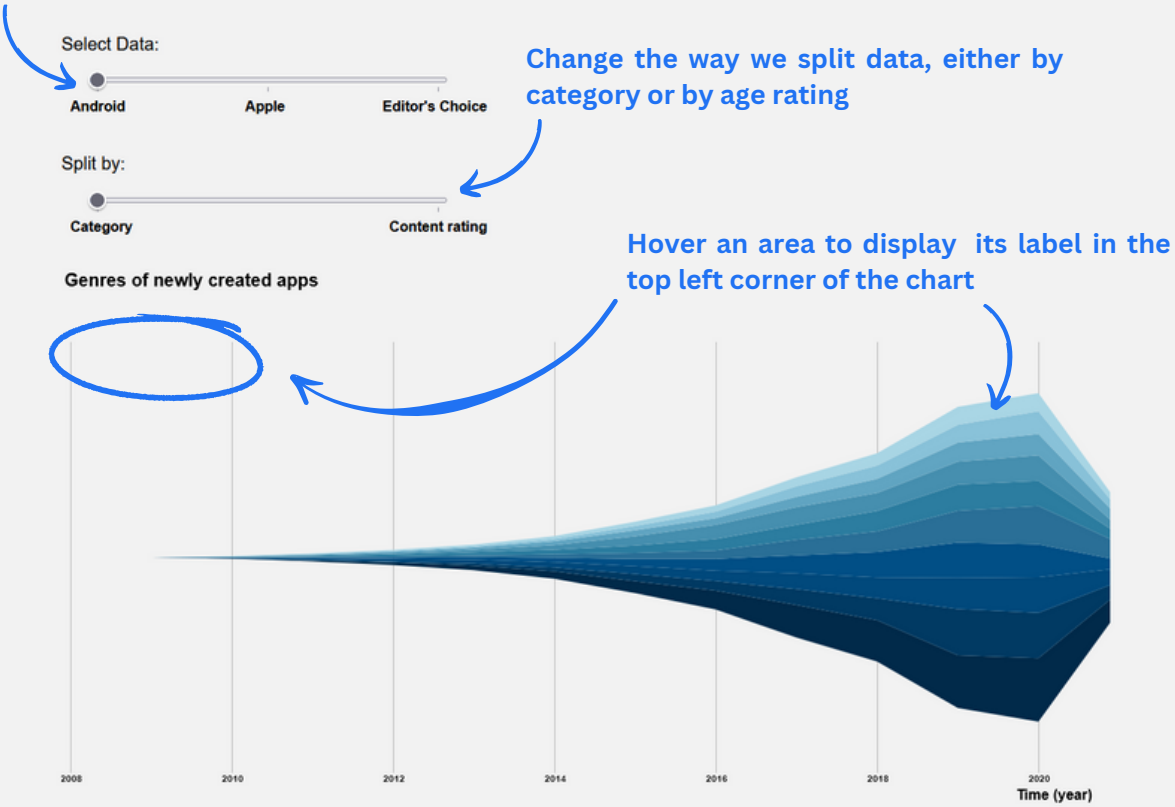
The **stacked bar plot** was built without any issues. We have the hue stacked on top of each other instead of being side by side. It was built using google charts.

# Functionality



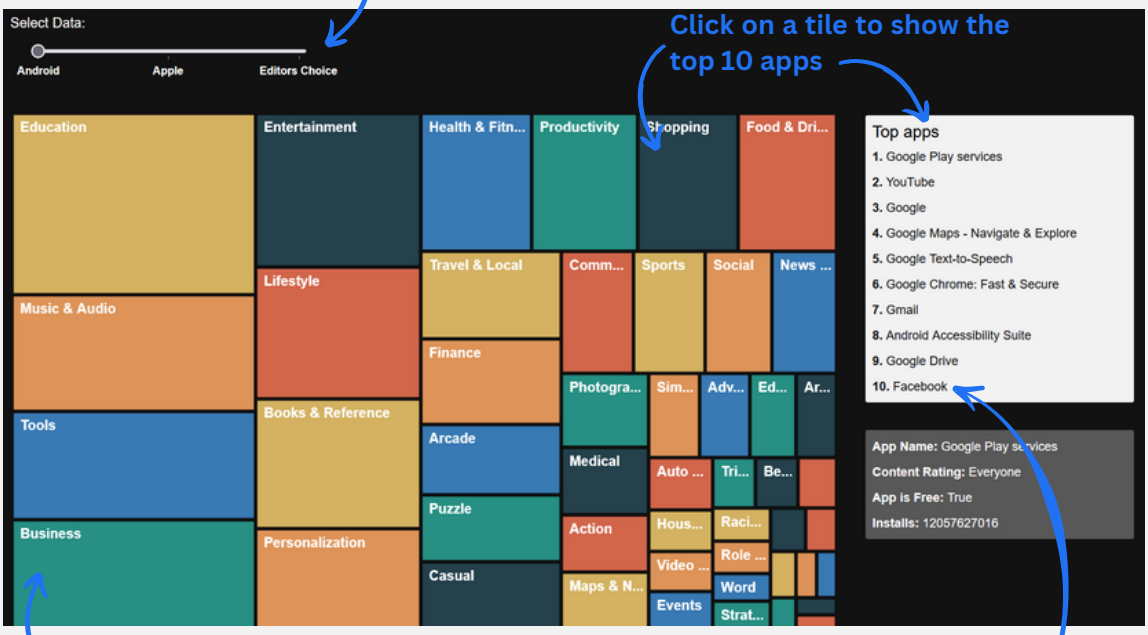
Change between datasets




Change the way we split data, either by category or by age rating

Hover an area to display its label in the top left corner of the chart

Change between datasets



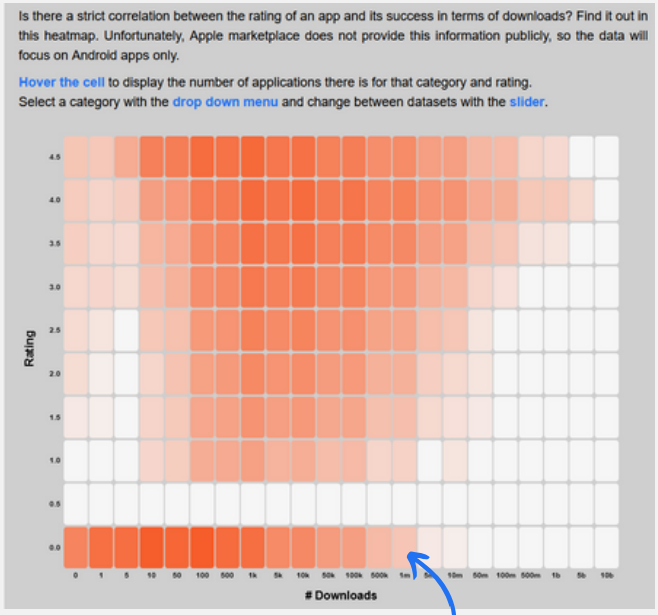

Click on a tile to show the top 10 apps

Hover a tile to show more information about it

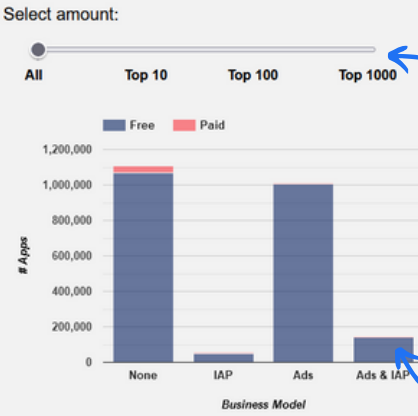
Click on an app name to show more information about it

Change between datasets


Change the category with this menu




Hover a cell to show the number of apps




Change how many applications you want to take into account

Hover an area to show the number of apps

Note: changing the dataset and category affect all 3 charts

# Peer assessment

### Elia Fantini

- Proposed main idea and some visualizations
- Partially wrote Milestone 1 and 2
- Created website's draft
- Sankey diagram
- Navigation bar
- Heatmap
- Stacked Bar Plot
- Plots purpose description
- CSS for layout and colors
- Screencast

### Konrad Litwiński

- Exploratory data analysis
- Created website skeleton
- Streamgraph
- Treemap (app details)
- Refactoring

### Adrien Vauthey

- Partially wrote Milestone 1 and 2
- Treemap
- Donutchart
- Navigation bar
- Home page
- Plot functionality description
- CSS for some charts placement
- Process book