INF552 Project - Brazilian Flights

ALBUQUERQUE SILVA, Igor
igor.albuquerque-silva@polytechnique.edu

GALVÃO LOPES, Aloysio
aloysio.galvao-lopes@polytechnique.edu

December 2020

1 Important

We've developed a fairly complex project. This way, this report will focus in describing in more details our design decisions, the insights that can be inferred from the visualization and the technical aspects. Information about how to use the visualization is already provided in the visualization itself.

We recommend that you first see the video provided with this report (if you can't access or find it please let us know) to get a general view of the project.

Useful resources:

- Visualization link: http://brazil-flights.herokuapp.com/
- Instructions: provided in the visualization's webpage.
- Video: provided alongside the report, you can also use this link to view it on Youtube.
- The code is available at https://github.com/aloysiogl/inf552-project
- This report.

2 Introduction

We've decided to make an interactive visualization of Brazilian flights ranging January 2000 to October 2020. In this visualization you have access to three

main windows that interact with each other to show different aspects of flights data.

This project is based on a dataset freely available at ANAC's (Brazil's National Agency of Civil Aviation) website. We chose to do this visualization as we both are Brazilians so we decided to visualize something related to Brazil and, at the same time, flights data offer plenty of useful information to create rich visualizations. In addition to this, the Brazilian flights dataset was the most complete flights dataset we found on the web.

3 Design Decisions

Our focus was to have a clean and elegant look, and have the user interact with the data as much as possible. The main goal of the application is to provide liberty to obtain different insights from the data. Figure 1 shows the initial view of the application.

We'll comment each main window that we developed and explain the design decisions that lead us in the design of each view.

3.1 Map window

In this window we're visualizing information about flights routes, thus, a network with geographical data. Therefore, we chose to show the airports as points in a map. This way the nodes are the airports and the links are the routes. Those nodes and links can vary based on the selections on the other view.

It's also important to highlight that we simplified the map, as well as it's colors, to make the flights information stand out. The map is mostly dark and, when necessary, some parts of it are highlighted to show selections, for example if you click in Brazil, it will be highlighted with a lighter grey and only flights in Brazil will be considered. Overall, the map colors were chosen to not distract the viewers attention from the flights information, so only the countries and Brazilian states divisions are shown.

As the two main channels for quantitative data were used (for encoding the position of the planes and the routes) we chose to inform additional data using color, size and animation.

As a lot information can be displayed, color, size and animation are used in a complementary and also redundant way to ensure that the information is well transmitted. Since conveying absolute quantities using these channels is very hard, these channels are always used in a comparative way, for example

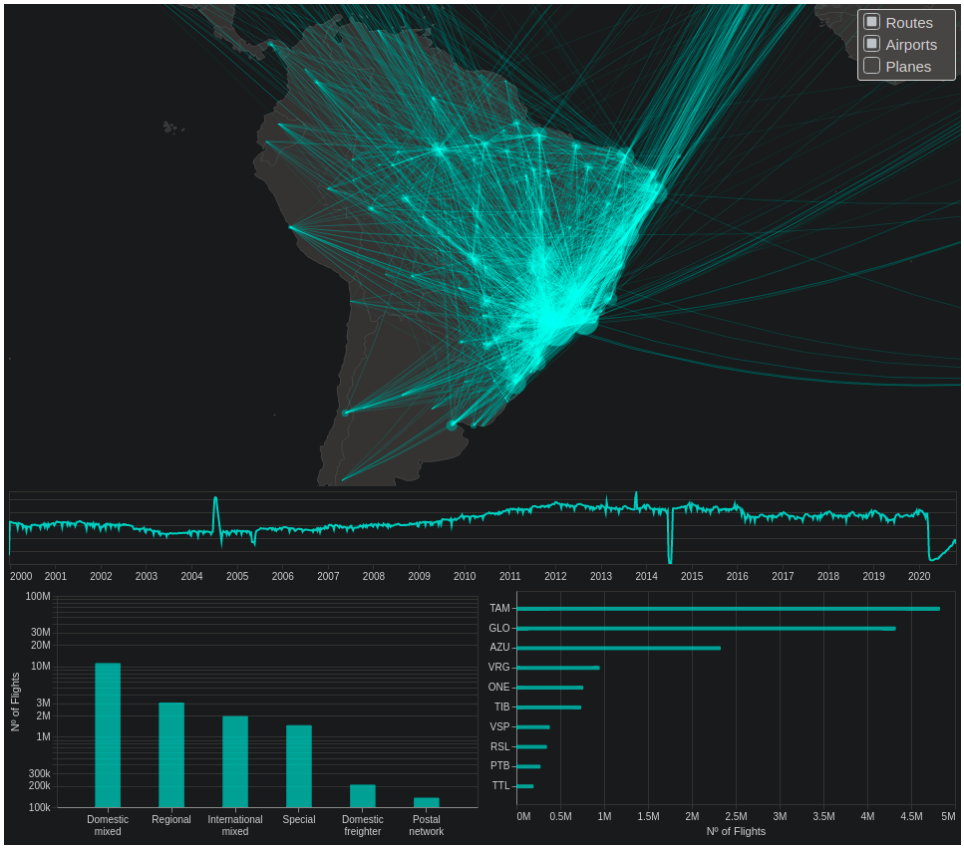

Figure 1: Initial view of the application, with no filters applied.

you can see circles over the airports, but their area represent a comparative measure between all the selected airports.

3.1.1 Color

You are able to chose to view the routes in the map via a checkbox. When this option is selected the routes are shown by lines in the map. In this case the color brilliance encodes the number of flights in a given route in relation to all of the other routes in the map. As it's not appropriate to use this channel for measuring absolute values, these intensities are used only to compere different routes and no scale is given. We also add some opacity to the routes so that when they overlap their brilliance appears stronger.

3.1.2 Area

You are able to view the airports in the map via a checkbox. When you select this option a circle appears in each selected airport and their area represent a comparison between all of the selected airports in flights flow. Again, we don't convey the notion of quantity as this channel is only appropriate to convey order. If you hover over a specific airport a tooltip will appear and show you specific information about a given airport, such as flights flow, name etc.

3.1.3 Animation

In this last channel you are able to view animated airplanes in the map by selecting a checkbox called planes. By activating this mode, planes will be generated on the map according to the routes flow. We use this additional channel to encode the routes' average speed, this way if you see fast planes in the map you'll be able to compare routes in speed. Again it's possible to convey the notion of order but not the notion of quantity. Some planes are red, this means that they are delayed of more than 15 minutes (default delay by the FAA), this way if a route has lots of red planes you know that flights are often delayed in a given route. If you hover over a plane you can get numerical data about a route. This way, by sampling planes from the routes we can get more precise information without having to show every route at the same time.

3.1.4 Wrap up and some other decisions

All of the channels described above can be shown at the same time or not. Depending on the number of states that are selected and the selection of the other windows, it'll be more interesting to show just some of them or even all of them. They're complementary and redundant, this way the main information (routes utilisation) can be transmitted more efficiently.

We've decided to use a simple color pallet with mostly dark colors to represent the map, some lighter colors for textual data, and a light blue color for flights data, this way the most relevant information stands out easily. We took a lot of care in the color selection to make it always aesthetically pleasing and at the same time make the relevant data more visible. We've also tried to use as few colors as possible to not fall in the trap of making lots of elements stand out at the same time. As we're visualizing quite a lot of data using a lot of different colors would not be feasible and it'd even be confusing.

We took some time to find candidates of colors to use alongside the main light blue color. We concluded that white, yellow, green, orange, light red and light purple would be the best candidates. When adding the planes to the visualization we've concluded that they'd need to have a different color as blue would mix with the routes, therefore we chose white, as that color doesn't stand out much and don't mix a lot with our main blue. We chose yellow for the selected plane (if you hover over a plane you'll see specific information about it and it's route) because green and purple can mix more easily with our blue and finally we chose red for the delayed planes because it conveys something negative about that plane and also it wouldn't mix with the main light blue. Finally we emphasise that, as our visualization uses a rather small set of colors it's more friendly to those who are color blind.

3.2 The slider window

The objective of this window is to simply act as a selector for the considered time period, this way we didn't want it to stand out in relative to the map or the plots. With that in mind we decided to use the same darkest color of the map for the background as well as the light blue for a plot. This way we keep the same visual identity, as well as we keep it discrete. The plot is a line plot that shows the quantity of flights by date but no scale is given, this way the line is just used as a reference for changing the time window as as a semi-quantitative measure. The reason for not putting a scale was that the objective of this plot was to act as a slider and not as a plot, adding a scale would make the overall design more asymmetric and would not add a lot of extra value as the vertical space occupied by this window is small, making a scale hard to read and even drawing unintended attention.

3.3 The bottom plots

We've chosen to inform some categorical data in the form of plots, so we chose to use bar plots, as they're generally a good choice to present quantitative data and compare different classes. Again, we chose to keep the color pallet of the map to keep the visual identity and to not draw unwanted attention.

The bars in both plots can be selected to filter the data shown in all of the other views. To show to the user that a bar has been selected or hovered we change it's opacity.

3.3.1 The bottom left plot

In this plot we show information about flights types. As we found that the number of flights vary considerably between classes, we chose to use a logarithmic scale. We also chose to use vertical bars mainly because the other view was composed of horizontal bars, this way it's easier to comprehend that both plots are not related. These classes are also not competing, this way one class being above the other (in the horizontal base case) would have no meaning, using vertical bars helps to convey the sense that the classes don't have any superiority order. It's important to note that those classes are provided by our dataset and, thus, were not created by us.

3.4 The bottom right plot

In this plot we show information about the number of flights by company for the top ten companies in the current selection. We chose to use a linear scale because the difference of the number of flights per company usually falls only between one order of magnitude or two. The horizontal bars are sorted by number of flights, therefore the leading company is at the top of the plot. This way the natural association between top (physically) and superiority is explored in this view as well.

3.5 Final considerations about the design

The whole project was designed with interactivity in mind, this way every view interacts with every other and acts a a filter for the others. Therefore, you can interpret this visualization in the following way:

- The map has three modes of visualization that can overlap if needed and acts as a filter by geographic location.
- The slider shows the flight intensity in a period and acts as a filter by period.
- The bottom left plot shows the flights by type and acts as a filter by type.
- The bottom right plot show the flights by airline and acts as a filter by airline.

4 Code tools and data processing

The website was developed in JavaScript using d3 and Vega-Lite. The map was done in d3 and the three auxiliary plots in Vega-Lite. To make it run smoother and faster, we query the necessary data for each plot on a back-end server. Therefore, every time the user changes a selection in the application, each of the four sections requests filtered data to the server and displays it, in an asynchronous manner.

This back-end server was written in Node.js and is hosted in Heroku, because of the easy deployment process. In addition to feeding the html and front-end code to the browser, communicates with a Google BigQuery warehouse to obtain the necessary data for each query. This warehouse contains an SQL server and the database which will be explained in the next section, so it's able to process each query in a short time, and group the results based on what the plots require. For example, the airlines plot requires the 10 airlines with most flights given the current filters.

We decided to use BigQuery and a back-end server because our dataset is relatively big. It is an 85Mb file which required a considerable initial loading time in the website, and doing the queries locally also took a lot of time.

4.1 Data processing

Our data comes from the flights history database in ANAC's website, which is the national civil aviation agency in Brazil. An extended cleaning and preprocessing step was required to use this data, because it was provided in multiple tables for each month, which have different encoding, missing and renamed columns, many notations for invalid elements, and other problems. This preprocessing was done in Python using pandas.

We used three different airports datasets to obtain data about them, because we needed coordinates for all flights. We tested their coordinates by plotting their coordinates and grouping by state using Plotly, and manually fixed wrong coordinates. Figure 2 shows the results of this test.

Our cleaned dataset contains a table for each year, in a total of 3.5Gb of csv files and around 20 million flights. Each flight contains the following columns: `airline`, `di`, `type`, `origin_airport`, `destination_airport`, `scheduled_departure`, `real_departure`, `scheduled_arrival`, `real_arrival`, `status`, `reason`, `origin_latitude`, `origin_longitude`, `destination_latitude`, `destination_longitude`, `origin_state`, `destination_state`, `domestic`.

We filtered these flights by those that involved at least one Brazilian airport that had more than 1000 flights in total, and were operated by an

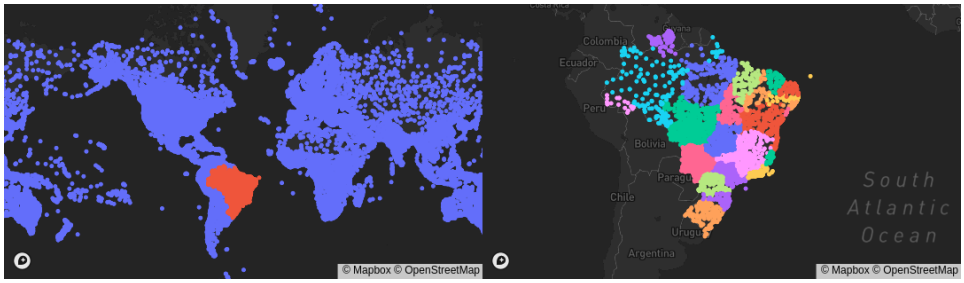

Figure 2: Airports dataset test. Note that airports in the same state are correctly positioned grouped with other airports from that state.

airline that had more than 1000 flights. With these filters we still contain around 90% of the initial dataset.

We then grouped these flights by destination, origin, week, airline and type. For each of these groups we calculate the count, average flight duration and average delay. We created a final table with these groups, which forms an 85Mb file with around 2 million routes.

5 Interesting insights

A lot of interesting insights can be taken from our application. We collected some and present them in this section.

5.1 COVID-19 crisis

A first insight from the data is the number of flights drop during the COVID-19 crisis. Figures 3 and 4 show the number of domestic and international flights before and during the crisis. We can see huge drops in both plots, but a slow recovery can be seen in the domestic flights, while the international flights level is still nonexistent.

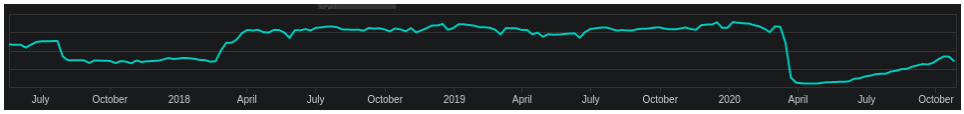

Figure 3: Number of domestic commercial flights distribution from July 2018 to October 2020.

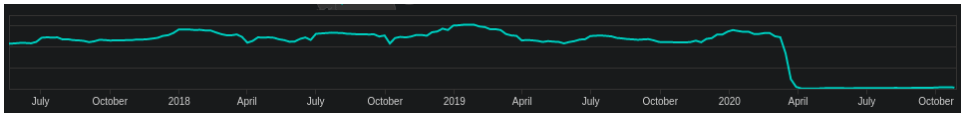

Figure 4: Number of international commercial flights distribution from July 2018 to October 2020.

Figure 5 shows the distribution of freighter flights in the same time period, and we can see that there isn't any major change during the crisis, which is interesting.

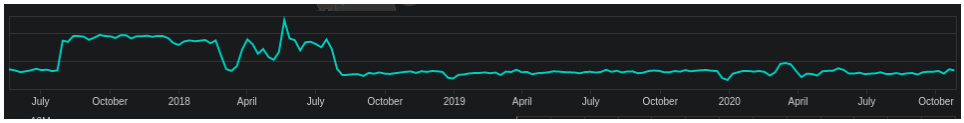

Figure 5: Number of freighter flights distribution from July 2018 to October 2020.

5.2 Airlines evolution

We can also visualize airlines evolutions/bankruptcies. For example, Figures 6 and 7 show the evolution of Gol Airlines, today the second airline with the most flights, and the bankruptcy of Varig, formerly the largest airline in latin america.

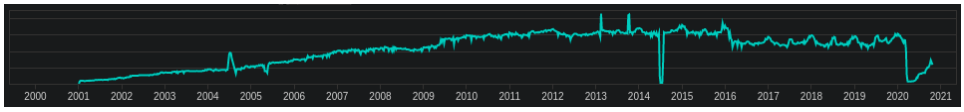

Figure 6: Evolution of GOL Airlines since 2000.

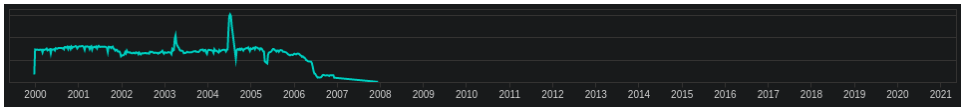

Figure 7: Evolution of Varig since 2000.

We can also see the evolution of international flights in general in Figure 8. The total number of international flights has doubled since 2000.

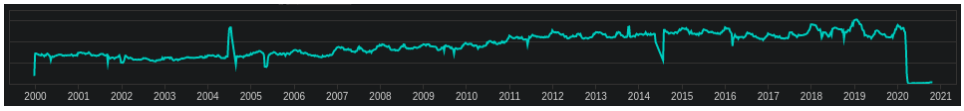

Figure 8: Evolution of international flights since 2000.

5.3 Delays by region

We can see patterns in delayed flights in some routes and regions. For example, Figure 9 shows how in the state of Pará delayed flights are more common than in the states of São Paulo and Minas Gerais.

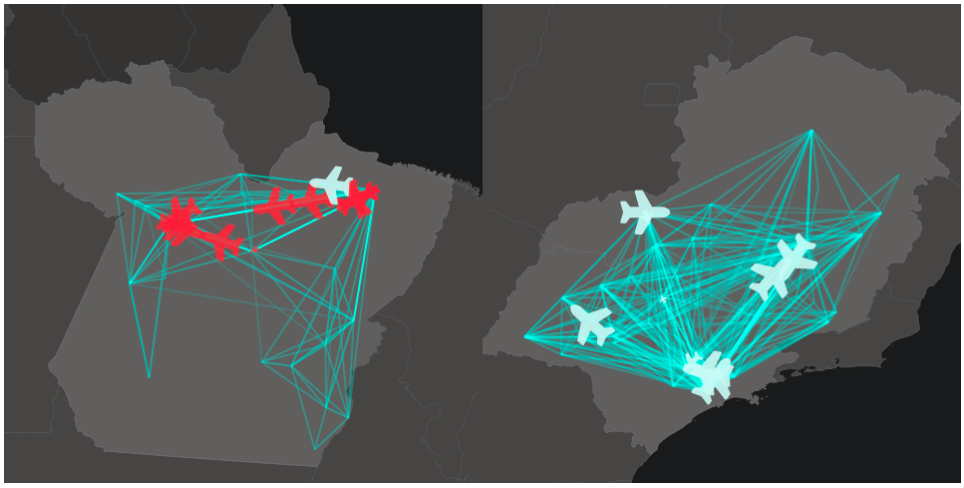

Figure 9: Delayed flights between the north and the south-east regions.

5.4 Flights networks

Different flights types contain very different networks. Figure 10 contains commercial, freighter and postal networks showing this effect. Note how there is a big difference between different commercial routes, while in the postal network there are only a few routes equally important.

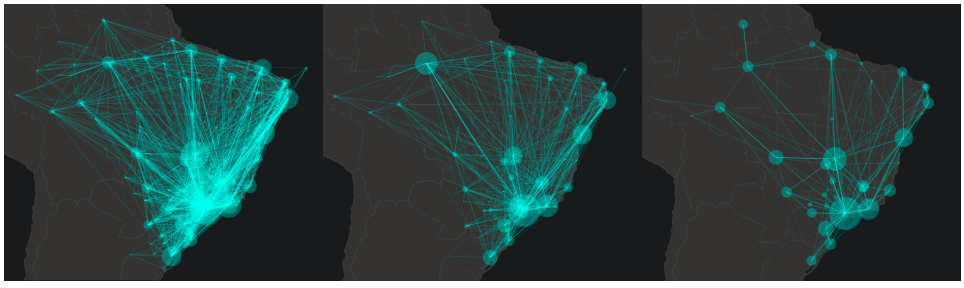

Figure 10: Commercial, freighter and postal routes, respectively.