IS428 - Visual Analytics for Business Intelligence AY2016/2017 Term 1

Topic:

Visualising Air Travel within United States

Aaron Mak Kang Sheng
Ong Ming Hao

Introduction

Air travel has been growing at a constant rate during the past 20 years even when economic growth has been stagnant.[1] Recently decreasing oil prices have also decreased the price of jet fuel. Together, they have increased the airline industry's profitability by a whopping 12%[2].

With the increasing competition that comes with a more profitable industry, airlines are on the lookout for better ways to optimize their routes to ensure a full load of passengers. There is an increasing trend of airlines finding ways to use big data[3] to better tailor to customer's' needs[4] or to better allocation of resources. Due to these benefits, airlines are starting to realize this competitive edge of data analytics and one aspect where data analytics is useful is route optimization.

Motivation & Objectives

One of the largest cost drivers in US airlines is labour. In the two largest airlines, labor unit cost increase exceeding 7% year on year. Over 2 years, labor costs have increased 16.4% and 17.5% at JetBlue.[5] In addition, there is increasing capacity, combined with significant fare competition in the US domestic market, resulting in lower yields. Systemwide passenger yield declined 5.1% during the second quarter 2016, compared with the same period a year earlier. One way to reduce labor costs and increase passenger yield is to ensure that the airline routes are optimized to ensure that the flight is as full as possible before taking off.

Even though we know that could be a potential solution, tools in the market are sorely lacking. There is no precise, business oriented tool that allows the user to explore

[1] http://www.iata.org/whatwedo/Documents/economics/Economic-Performance-of-the-Airline-Industry-mid-year-2016-forecast-slides.pdf

[2] https://www.theguardian.com/business/2016/jun/02/airline-industry-profits-expected-to-increase-2016-iata

[3] http://www.futuretravelexperience.com/2016/01/5-technology-trends-that-airlines-and-airports-should-be-prepared-for-in-2016/

[4] http://www.wns.com/insights/articles/articledetail/62/5-trends-for-the-global-airline-industry

[5] http://www.oliverwyman.com/content/dam/oliver-wyman/global/en/2016/jan/oliver-wyman-airline-economic-analysis-2015-2016.pdf

flight patterns visually using a map. This could be because such a tool is very niche and would be costly to customize.

Therefore, we aim to create a tool that allows airlines to easily visualize existing routes based on historical passenger data so they can find the optimum routes for profitability.

The objectives of this tool are as follows:

- To explore the flow of passengers from each airport and state
- To discover flight trends between states and airports
- To reveal passenger trends over time

Background Research

This section discusses about visualisations created by various people to help in this project. From visualisation that illustrates the different aspects of the airplane industry, to various visualisation that would enable our visualisation that we create to gain more insights or better filtering options for users.

Flight Stream

The first visualisation is from flight stream (http://callumprentice.github.io/apps/flight_stream/index.html#), which visualises the flights happening worldwide. What amazed us was the ability for the app to allow the user to adjust the speed and size of how the flights are represented and the opacity of the trails they leave as they traverse the globe.

Though it looks amazing, little insight can be obtained from this visualisation. We are only able to know which are the more popular routes that flights take as well as places where flights are highly dense. If you want to look more in depth into the data, we are unable to. This visualisation is lacking in going into detail, such as the number of flights to each airport, or the number of passengers travelling to and from the different airports.

Using Tableau to visualize performance of major airline carriers

The second visualisation is from Ben Jones, which uses Tableau to visualise the flight delays that happened within the United States (https://public.tableau.com/en-us/s/blog/2015/05/visualizing-more-five-million-flights). It uses a node-link diagram, where you are able to visualise the the delay times for flights that leave a specific airport and the airline associated to the delay. This is a more business-oriented approach that is created using tableau dashboard. Unlike the first visualisation, it goes into incredible details, by allowing users to select the state where flights are leaving, as well as the airline. In addition, it would show the user an overview of the total time it delay for all flights in a table format, making it incredibly detailed.

Thus, insights such as which airline to take in order which has the least probability of delays are able to be obtained from this visualisation.

However, this visualisation fall short in the area of user friendliness. It has hard to navigate using this visualisation, and the loading of data is very slow. In addition, it does not breakdown into the various airports, which could bring even more detail to the visualisation,

Co-occurrence Matrix

The third visualisation is a co-occurrence matrix created by Mike Bostock. (https://bost.ocks.org/mike/miserables/) This visualisation illustrates character co-occurrences in the movie, Les Miserables”. Each coloured cell represents when both characters appeared in the same chapter, with cells that are darker indicate a higher occurrence.

This visualisation is brilliant in it’s ability to show an overview of many nodes in just a single screen. If you want to know if a certain character appear alongside another, you can just use this co-occurrence matrix.

However, this visualisation is poorly visualised. There is no legends to indicate what the different colours mean, how much times each person appear with each other, etc. There is no detailed information when you hover over each square, etc.

Calendar View

The last visualisation is a “Calendar View” created by Luis Carli (http://luiscarli.com/fitbit/). A calendar view is a heatmap which shows the breakdown of a specific data. This enables the users to have an overview of that data over a long time period. In this visualization, it visualises the amount of steps a person has walked throughout the year according to his fitbit.

This calendar view is other calendar view, because you are able to hover over each day to see a detailed breakdown of the number of steps taken for that day. In addition, it also shows a graph to visualise the average amount of steps taken for each day and for each week of the year.

This overall approach of going into detail as well as giving an overview, is something that our visualisation must achieve in order to give a great visualisation.

Design Framework

Data Preparation

Our data source is the US Department of Transportation. Since we are only focusing on domestic flights, we did not need to source the data from elsewhere. Flight history and airport details are taken from www.transtats.bts.gov/.

Airports

The raw airport data has a lot of unnecessary data. The first step was to remove airports that were not in the US. After all, we are only analyzing domestic flights. Using excel to filter the csv file, we deleted all airports that were outside US.

There are also duplicates of airports as seen in the above image. On closer inspection, we realized that the details of the airports were updated so we had to take the latest one. Looking at the column, “AIRPORT_THRU_DATE”, we can see which airports have ceased to exist so we used that column to remove the duplicates.

Fields such as AIRPORT_SEQ_ID, AIRPORT, AIRPORT_WAC_SEQ_ID2, AIRPORT_WAC, AIRPORT_COUNTRY_NAME, LON_DEGREES, LON_HEMISPHERE, LON_MINUTES, LON_SECONDS, CITY_MARKET_WAC_SEQ_ID2, CITY_MARKET_WAC, LAT_DEGREES, LAT_HEMISPHERE, LAT_MINUTES, LAT_SECONDS, AIRPORT_STATE_FIPS were removed since they are not needed for analysis.

Last of all, we transformed the .csv file to a .geojson file so that it is easier for d3 to process.

Flights

Similarly, the raw flights data for 2016 came with unnecessary data. It’s file size is a whopping 33.9MB so we have to reduce the data set for quicker loading on the web.

The flight data is summarized by month, route and carrier. Even though it is not the master data, we will have to make do because this dataset is as detailed as possible for public data.

Since we are only concerned with routes that have passengers, we removed flight routes that had 0 passengers that month.

We also removed fields that we would not use for our analysis such as UNIQUE_CARRIER, AIRLINE_ID, UNIQUE_CARRIER_ENTITY, REGION, CARRIER, CARRIER_NAME, CARRIER_GROUP, CARRIER_GROUP_NEW, ORIGIN_AIRPORT_SEQ_ID, ORIGIN_CITY_MARKET_ID, ORIGIN, ORIGIN_STATE_FIPS, etc.

Storage Method

Since we are not writing any data, we figured that the best way to store the data is on the client. Consequently, we load the data when the browser loads the page. This also ensures that the app quickly responds to any changes in input since no additional data is needed from a backend.

We stored the data in geojson and csv format which are readable by the d3 library.

Final Application Design

Technology Used

The application is built with a few libraries. Namely, d3.js[6], d3-legend.js[7], crossfilter.js[8], selectize.js[9], twitter bootstrap[10], noUISlider[11], d3-tip.js[12].

Crossfilter.js is essential because it helps us filter through the large dataset within a fraction of a second.

D3 is our main charting library as it renders the svg and parses the .csv files and .json files.

The rest of the libraries are for aesthetic purposes.

With the required considerations in mind, we came up with the final application design that consists of an almost ideal layout and interactivity that we planned for. The remaining of this section aims to uncover our reasons of choice for the selected visualizations.

[6] https://d3js.org

[7] http://d3-legend.susielu.com

[8] https://github.com/square/crossfilter

[9] https://selectize.github.io/selectize.js/

[10] http://getbootstrap.com

[11] https://refreshless.com/nouislider/

[12] https://github.com/Caged/d3-tip

Month View

Unfortunately, the data that we obtained only had month, we are unable to do calendar view. Instead, we decided to adopt a unique approach called “Month view”. In this month view, we are able to have an overview of all the passengers for any month in the year 2015.

Hovering on the months will display the number of passengers and it’s month. In addition, the color scale has 6 shades with the lightest shade being the minimum flights per month and the darkest shade being the maximum flights per month. This enables the user to immediately have an idea of which month has the most number of passengers.

However, this Month View is not only used for giving the user an overview of the number of passengers for each month in 2015. It is also able to filter the flights to specific months, either by clicking on a specific month that you wish to filter or by using the slider bar to filter specific consecutive months.

From Month View, we are able to gain insights such as which months are the more popular months to travel. This would help the airline industry to better price their tickets in the future. For example, since October and November has the most number of passengers, Southwest airline can lower their fares in order to attract more customers.

Network Map

The network map shows the airports and its corresponding passengers. The passengers are visualized in 2 ways. The size of the circle shows the outgoing/incoming passengers for that airport. The arcs, which appear upon clicking the airport, shows the routes originating from the clicked airport. The sum of passengers on the route are linearly scaled to the thickness of the arc.

Hovering on the circle would reveal the name of the airport, the city and state and the sum of outgoing/incoming passengers. Similarly, hovering on the arc would reveal the origin and destination airport together with the sum of the number of passengers.

The incoming/outgoing passenger filter on the left of the network map will result in the airports being filtered accordingly to the range being displayed. Similarly, the toggle switch will allow the user to switch between viewing the outgoing and incoming passengers.

In order to filter by state, the user has to add a state into the dynamic dropdown. This dropdown allows for multiple inputs so more than one state can be selected.

We used a network map so that the user can easily visualize the number of passengers geographically. The network map also shows the flow of passengers from one airport to another. Filters also help the user to quickly navigate to a specific class of airport or state.

Passengers Matrix

The passengers matrix is a matrix table that gives the users incredible detail in the number of passengers moving from one state to another. In addition, it would give an overview of the different flights for a specific state. For example, we are able to identify that Delaware only have flights to 8 states. This matrix table is affected by the month that you choose.

By hovering over each square you are able to give a detailed breakdown of the number of passengers flying between the origin state to the destination state as well as the origin state and destination states. In addition, we are also able to identify the more popular states people travel to, within the United States.

This visualisation would give insights such as which state is a popular place to go to, and where do most of the visitors come from, and how much visitors travel to that specific state from another specific state. For example, we are able to see from the picture below that

50,790 passengers travel from Illinois to Arkansas in the year 2015. Lastly, we are also able to view the different breakdown of passengers of specific month(s).

Challenges

One of the initial challenges was filtering the data quickly. If we used our own simple functions which do not involve indexing in any form, the function will take forever to parse through the 20,000 rows of flight data. By using crossfilter.js, this is done in a fraction of a second, which greatly increases the speed of the application.

Another challenge that we had was the drawing of the arcs. Since there are so many arcs, the browser will take some time to draw them out. To give the illusion of responsiveness, all the arcs are drawn when the application is first loaded. Then when the airport is clicked, it will toggle the ‘display’ attribute of the arc based on the origin state encoded into the arc attribute using jquery.

The reset filters button would originally draw the entire network graph twice since the outgoing and incoming passengers sliders were reset causing it to be redrawn once for each slider. Since it took too long, we changed the event listener to wait until both sliders are set, then drawing the graph once, reducing the loading time dramatically.

Insights

1. Alaska has many airports for a small state and only a small fraction of flights go out of Alaska. In other words, most of its flights are intra state.
2. Majority of passengers are concentrated in the coastal areas.
3. Generally, a larger amount of passengers travel from a large airport to another similarly sized one.
4. A large amount of passengers travel from east to west and vice-versa.
5. October and November are the busiest months to travel.

Future Work

There are a few improvements and suggestions we can consider for future versions of of this application.

By allowing the user to be able to zoom in further to the network map, the user can understand and explore the data better. This is especially helpful when there is a cluster of airports such as in Alaska.

Another helpful feature could be the individual filtering of airports. Likewise, it would help when there is a cluster of airports and the user might find it difficult to view smaller airports if the filter is not used.

Pricing could also be added as another dimension for analysis alongside passengers. This could help airlines to price competitively by considering what the average price is for each airline on that route.

Lastly, gathering more flight data than just one year. This could give the airline industry a more accurate view of passengers flying within the United States. It would also bring about more opportunities for data visualisation since there is only limited visualisations that you can do when your data is only limited to one year, such as finding the average passengers travelling for that year.

Acknowledgement

We would like to thank Dr. Kam Tin Seong, Associate Professor of Information Systems (Practice) and our peers from IS428 Visual Analytics for Business Intelligence 2016- 2017 T1 for their support and feedback.