Casey Lee
Euan McInally

Airport Statistics Visualizations

Motivation:

Our project was about statistics regarding delays within airports, focusing primarily on weather, but also on other aspects. Our main objective was to create an interesting visualization that still allowed the user to accurately and easily obtain data without having too many extra features. The intent for this project was for people to use and reference the data easily, and to detect trends within the data for use in actual flight plans. Our hypothesis was that the data would be best represented in a ranking format to allow the users to interact with it and find specific flights within certain months in order to check the chances of a flight being cancelled or delayed.

References:

[1]
Caviglia, Giorgio, et al. *RAWGraphs*, DensityDesign Research Lab, 2013, app.rawgraphs.io/.

This is an online tool we used in order to implement create custom streamgraphs.

[2]
Dey, Tanujit, David Phillips, and Patrick Steele. “A Graphical Tool to Visualize Predicted Minimum Delay Flights.” *Semanticscholar.org*, pdfs.semanticscholar.org/1887/7e423ae25146eb28fb9506921338ce6a4912.pdf.

This paper discusses a tool that was created to visualize flight delays as well as predict flight delays that are likely to occur to help people who are travelling make decisions. This paper attempts to tackle the same general goal as us of visualizing flight delays however they take it a step further to predicting future delays as well. There are significant differences however seeing as their tools looks at individual flights while ours attempts to give overall trends for various different airports.

[3]
Havre, S., Hetzler, B., Nowell, L. (2000) ThemeRiver: Visualizing Theme Changes over Time. Proceedings of the IEEE Symposium on Information Visualization. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=885098

An early paper on a streamgraph type visualization and the benefits such a visual would have. This is quite relevant to our project as a large chunk of our project was utilizing streamgraphs as a means to visualize airport delays.

[4]
Jones, Ben. “Visualizing More Than Five Million Flights.” *Tableau Public*, 31 May 2015, public.tableau.com/en-us/s/blog/2015/05/visualizing-more-five-million-flights.

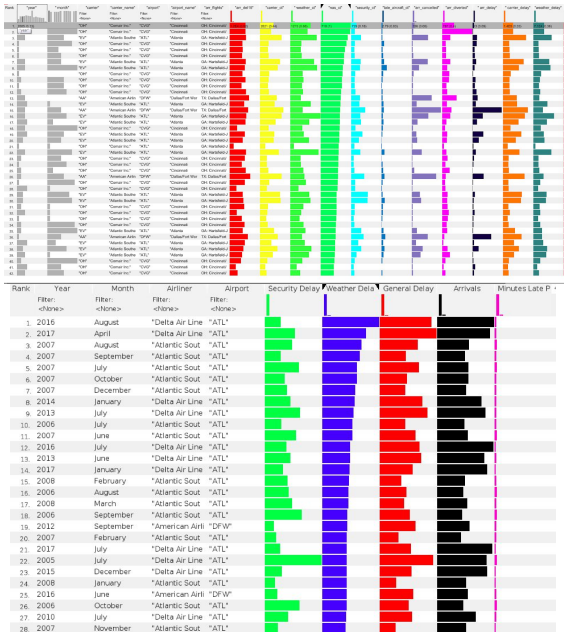
This is a website we found that includes a tool to visualize flight data for U.S. flights. It allows you to select an airport and an airline and it dynamically updates to show where that airline has flights to from the chosen city. This is an example of a visualization that is intended to express similar information to what our project is doing but it does it in a very different way. They even got their data from the same location as we did.

[5]
Transtats.gov, United States Department of Transportation, www.transtats.bts.gov/Tables.asp?DB_ID=120.

This is the online database where we acquired all of our data. This is where the U.S. Department of Transportation stores all of their records for all different types of transport.

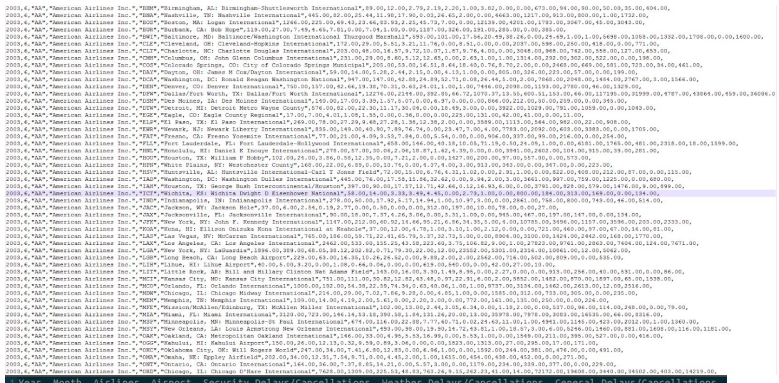
Design Evolution:

Our original idea focused around gathering data regarding school closures, especially with RPI. We had intended to gather school closure and delay for various schools in the area as well as local weather data in order to analyze the relationship between weather patterns and likelihood of school closures. The first week and a half of our project revolved around gathering that data. With that in mind, we worked to gather data regarding this information. However, neither RPI nor Russell Sage College seemed to record their snow days/closures, and none of this information was easily obtainable online. After a long chase through administration lines, we decided to give up on the idea. As a result, we switched over to the airport idea, as it seemed to line up on the same vein.



For our visualization, we initially decided upon using LineUp as our primary source of visualizations, as we wanted our data to be as clear as possible. We avoided D3, since we didn’t need the tools that D3 provides in order to create the final product we were looking for and neither of us had much experience at all with javascript and thus a D3 implementation of this would likely be outside the scope of this project. Additionally, we quickly realized that with the amount of information we were wanting to analyze and display, it would be almost impossible to have a dynamically updating tool that a user could interact with seeing as it

takes almost 20 seconds for the tool we are using to generate a streamgraph for even our most limited set of data. As a result we opted to instead create a series of static visualizations that could give a wide overview of the data we wanted to display, with the possibility of adding in interactivity in future if we decide to further develop this idea after this project. We started using LineUp as we thought this would be the most effective tool for this data and later, we chose to add in stream graphs, as we wanted the user to be able to see trends as well as specific data. As a result, we decided to stick with LineUp and stream graphs for this project. For the LineUp portion of our visualization, we first preprocessed the data, and visualized multiple sets of data to see the different variations, ie. we used only NY airports, or checked delays with a specific airliner. Later, we realized that it was inconvenient to spread apart the data, so we decided rather to do a large amount of preprocessing in order to make the data more agreeable. Previously, the data looked something like what is on the right. Later, we cleaned it up and made it look much more organized



```
Year, Month, Airliner, Airport, Security Delays/Cancellations, Weather Delays/Cancellations, General Delays/Cancellations,
 Arrivals, Minutes Late Per Arrival
2003,June,"American Airlines Inc.","ABQ",1326.88,524.68,3322.44,307,16.8534
2003,June,"American Airlines Inc.","ANC",676.75,278.09,1879.16,90,31.4889
2003,June,"American Airlines Inc.","ATL",5671.58,1400.99,9932.43,752,22.613
2003,June,"American Airlines Inc.","AUS",2935.32,3100.24,11010.4,842,20.2447
2003,June,"American Airlines Inc.","BDL",1152.48,829.9,4401.61,383,16.6684
2003,June,"American Airlines Inc.","BHM",144.39,96.79,1128.82,89,15.3933
2003,June,"American Airlines Inc.","BNA",1742.88,1242.44,6586.68,445,21.3383
2003,June,"American Airlines Inc.","BOS",4957.59,4270.43,15507,1266,19.5379
2003,June,"American Airlines Inc.","BUR",488.46,333.49,1607.04,119,20.4117
2003,June,"American Airlines Inc.","BWI",3098.75,1075.56,7425.69,593,19.5616
2003,June,"American Airlines Inc.","CLE",682.97,603.51,2845.51,172,24.8232
2003,June,"American Airlines Inc.","CLT",1319.79,984.57,3891.63,283,30.5221
2003,June,"American Airlines Inc.","CMH",841.77,300.6,1545.63,231,11.6364
2003,June,"American Airlines Inc.","COS",1331.04,685.51,3027.46,205,24.6049
2003,June,"American Airlines Inc.","DAY",284.59,331.28,1023.13,59,27.7797
2003,June,"American Airlines Inc.","DCA",4328.6,2090.88,9617.52,947,16.9345
2003,June,"American Airlines Inc.","DEN",4062.69,2140.66,9004.64,759,20.2773
2003,June,"American Airlines Inc.","DFW",49788.1,32391.8,156717,12276,19.4605
2003,June,"American Airlines Inc.","DSM",315.64,215.39,1234.97,140,12.6143
2003,June,"American Airlines Inc.","DTW",1891.21,1051.3,5868.49,576,13.908
2003,June,"American Airlines Inc.","EGE",85.63,135.01,243.36,17,27.2941
2003,June,"American Airlines Inc.","ELP",1582.96,1140.29,4612.76,269,27.2714
2003,June,"American Airlines Inc.","EWR",4070.63,2132.9,9671.47,835,19.012
2003,June,"American Airlines Inc.","FAT",326.37,341.09,1186.54,77,24.0779
2003,June,"American Airlines Inc.","FLL",2880.24,1805.18,7969.59,658,19.2325
2003,June,"American Airlines Inc.","HNL",1039.93,2032.06,4324.01,278,28.7626
2003,June,"American Airlines Inc.","HOU",666.93,203.86,2033.21,102,28.4706
2003,June,"American Airlines Inc.","HPN",357.74,349.89,1169.37,168,11.1726
2003,June,"American Airlines Inc.","HSV",304.33,414.76,955.91,72,23.2639
2003,June,"American Airlines Inc.","IAD",2032.48,1014.58,4431.94,445,16.8067
2003,June,"American Airlines Inc.","IAH",2110.37,845.57,4812.05,397,19.5667
2003,June,"American Airlines Inc.","ICT",489.94,187.33,951.73,58,28.0862
2003,June,"American Airlines Inc.","IND",1563.11,775.92,3485.97,278,28.9532
2003,June,"American Airlines Inc.","JAC",90.96,199.54,345.5,37,17.1892
2003,June,"American Airlines Inc.","JAX",351.32,474.37,1101.31,90,21.4111
2003,June,"American Airlines Inc.","JFK",4862.87,3556.92,13583.2,1147,19.1831
2003,June,"American Airlines Inc.","KOA",168.1,464.78,833.12,37,39.6216
```

By doing this we were able to clean up the data to make it look much neater, allowing for easier reading of the data, as well as less clutter for the user to have to look through.

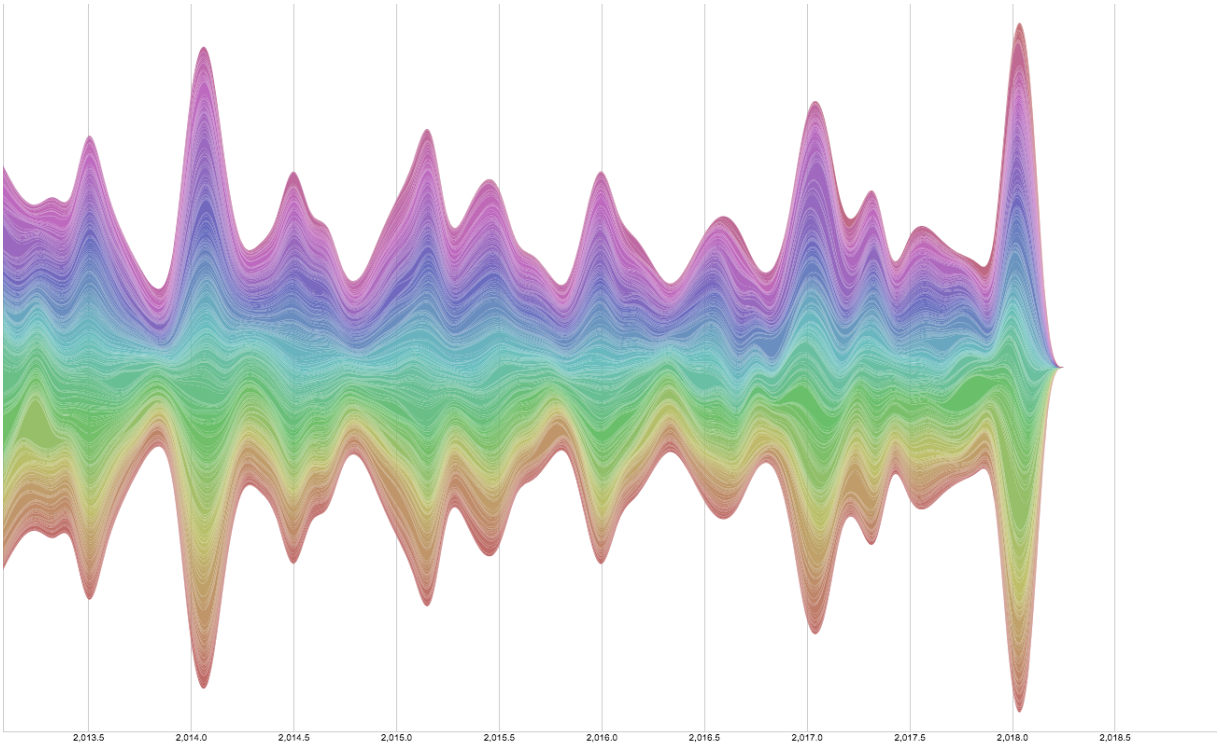
On the streamgraph end of things the first step was to go through 60 csv data files (one for each month of data being analyzed) each containing information for about 500,000 individual flights. A sample of the data files analyzed for the streamgraphs is included below.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Year	Quarter	Month	DayofMonth	DayOfWeek	FlightDate	UniqueCarrie	AirlineID	Carrier	TailNum	FlightNum	OriginAirport	OriginAirport	OriginCityMa	Origin	OriginCityNa	OriginState	OriginStateFi	OriginStateN
2	2013	1	2	17	7	2/17/13	DL	19790	DL	N718TW	2344	12478	1247802	31703	JFK	New York, N	NY	36	New York
3	2013	1	2	17	7	2/17/13	DL	19790	DL	N667DN	2347	12953	1295302	31703	LGA	New York, N	NY	36	New York
4	2013	1	2	17	7	2/17/13	DL	19790	DL	N614DL	2348	14869	1486903	34614	SLC	Salt Lake Cit	UT	49	Utah
5	2013	1	2	17	7	2/17/13	DL	19790	DL	N613DL	2349	10397	1039705	30397	ATL	Atlanta, GA	GA	13	Georgia
6	2013	1	2	17	7	2/17/13	DL	19790	DL	N922DL	2351	11193	1119302	33105	CVG	Cincinnati, O	KY	21	Kentucky
7	2013	1	2	17	7	2/17/13	DL	19790	DL	N919DE	2352	14635	1463502	31714	RSW	Fort Myers, F	FL	12	Florida
8	2013	1	2	17	7	2/17/13	DL	19790	DL	N927DA	2353	10874	1087402	30647	CAK	Akron, OH	OH	39	Ohio
9	2013	1	2	17	7	2/17/13	DL	19790	DL	N129DL	2355	10397	1039705	30397	ATL	Atlanta, GA	GA	13	Georgia
10	2013	1	2	17	7	2/17/13	DL	19790	DL	N332NB	2358	14027	1402702	34027	PBI	West Palm B	FL	12	Florida
11	2013	1	2	17	7	2/17/13	DL	19790	DL	N637DL	2359	11278	1127802	30852	DCA	Washington,	VA	51	Virginia
12	2013	1	2	17	7	2/17/13	DL	19790	DL	N195DN	2364	12173	1217301	32134	HNL	Honolulu, HI	HI	15	Hawaii
13	2013	1	2	17	7	2/17/13	DL	19790	DL	N192DN	2365	12892	1289203	32575	LAX	Los Angeles,	CA	6	California
14	2013	1	2	17	7	2/17/13	DL	19790	DL	N773NC	2366	15323	1532302	35323	TRI	Bristol/Johns	TN	47	Tennessee

We wrote a parser to go through and pick out what information was actually needed for the streamgraphs and print out to a file the month by month total flight statistics for every U.S. airport. As you may expect, this process ran very slowly with so much data to process and as a result this step of the process took a long time to get working correctly. A sample of the total month by month statistics data that was printed out is included below.

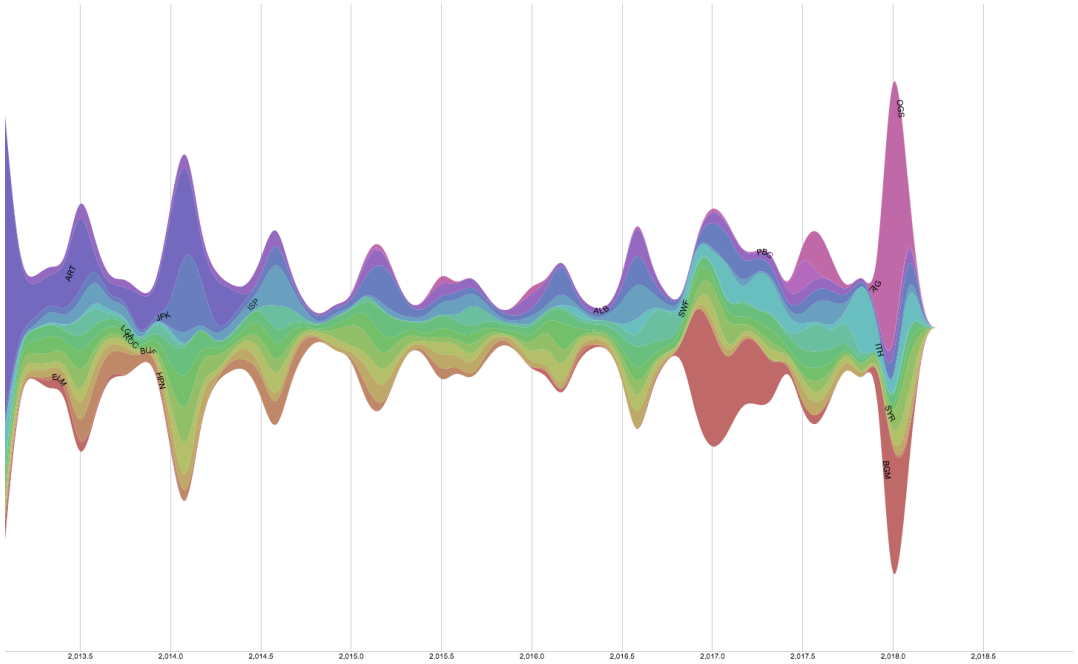
```
Airport:SIT:Sitka, AK

        Year:2013
        Month:1
                Cancellations:0
                Weather Cancellations:0
                Departure Delays:21
                Total Dep. Delay Time:1163
                Arrival Delays:34
                Total Arr. Delay Time:1112
                Total Weather Delay Time:586
                Departure Count:92
                Arrival Count:93
        Month:2
                Cancellations:0
                Weather Cancellations:0
                Departure Delays:12
                Total Dep. Delay Time:585
                Arrival Delays:12
                Total Arr. Delay Time:270
                Total Weather Delay Time:61
                Departure Count:84
                Arrival Count:84
        Month:3
                Cancellations:0
                Weather Cancellations:0
                Departure Delays:21
                Total Dep. Delay Time:459
                Arrival Delays:25
```

Once this slow processing step had been completed, the next step was to simply take all of this information that we have calculated and format it into csv files that can be plugged into the streamgraph tool in order to create the final visualizations. We wrote another parser to convert from the above format into several different csv files for different visualizations we wanted to make (we ended up with multiple streamgraphs in order to analyze various different aspects of the data as well as using several different subsets of the data to allow for more careful analysis of individual data. We began by creating a streamgraph depicting the average weather delay time per flight over time for every airport in the U.S. This ended up creating quite an interesting visualization (shown below) that was very useful for analyzing general trends in the data, however it is essentially impossible to pick out information regarding specific airports in order to compare them to each other.

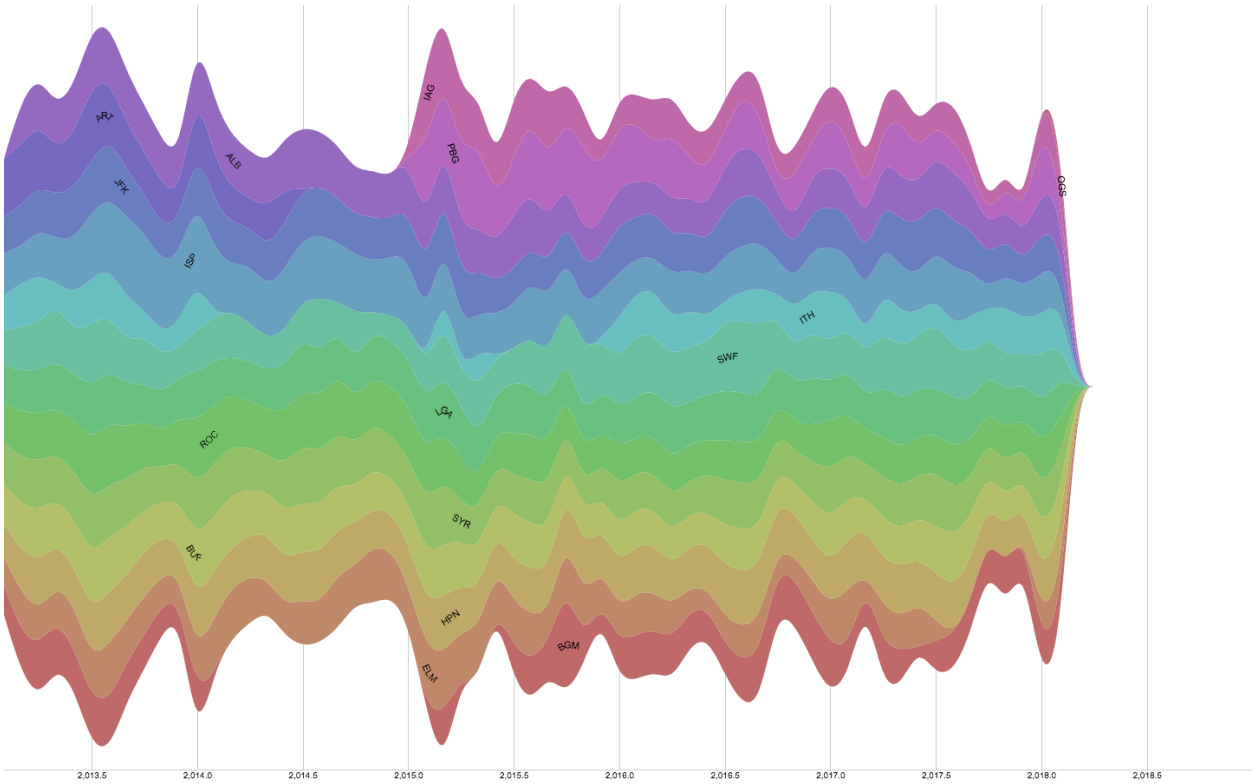


Our initial streamgraph visualization showing average delay time per flight over time for airports using data for all airports in the U.S. As you can see it is very clear to pick out general trends in the data (i.e. there is a large spike in delays every winter and a smaller spike every summer) however it is impossible to differentiate the colors from one another in order to get information on any particular airport.

After seeing this visualization, we decided that visualizations of the entire data set are useful for general trends, but in order to allow users to discern information about individual airports we would need to use a more limited data set. As such we created some visualizations using only New York airports in order to give an idea of how these visualizations could be used as we figured that people are mostly going to want to compare airports in the same general area to one another so a visualization showing all airports in NY state would likely be handy when deciding which to go to. Two of these New York specific visualizations have been included below.

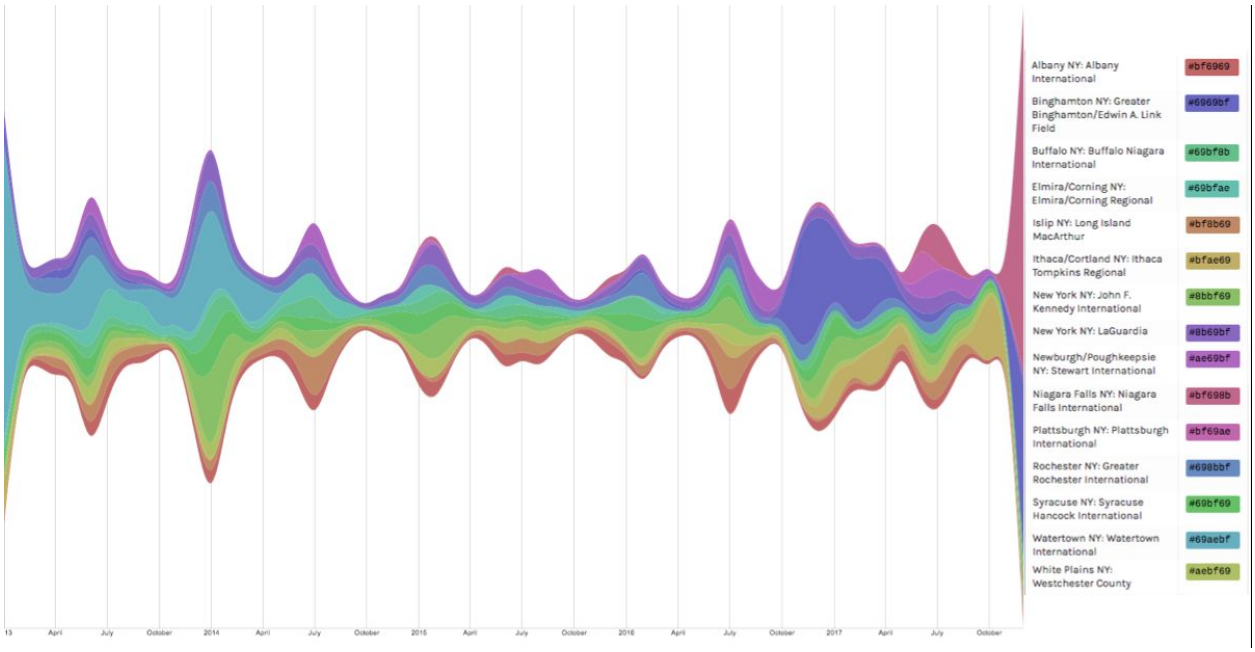


Our next version streamgraph visualization showing average delay time per flight over time for airports using data for only New York airports.. As you can see it is much easier to see information for individual airports in this version.

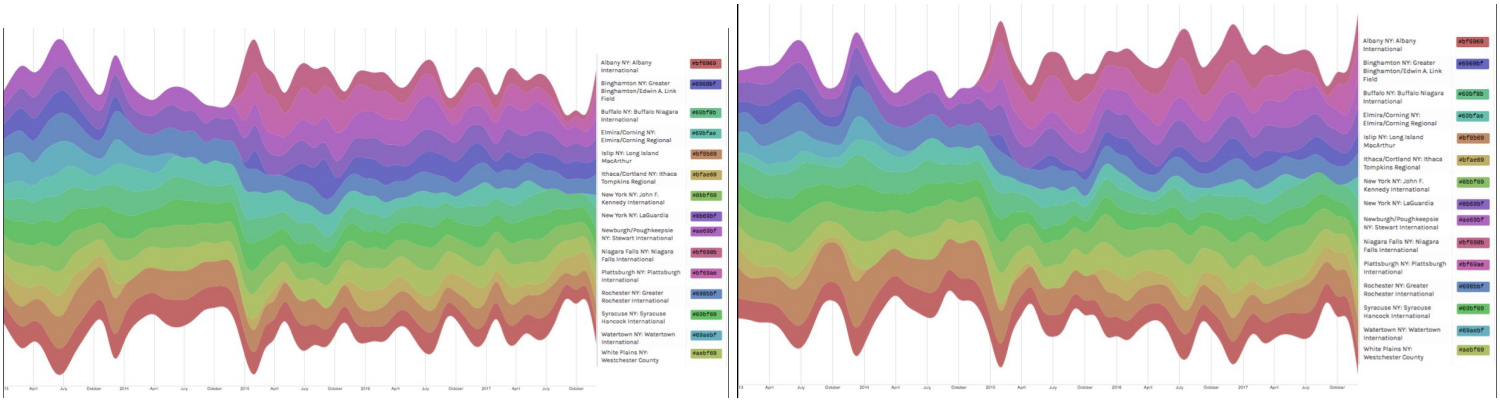


A streamgraph depicting percentage of arriving flights that are delayed for each airport in New York.

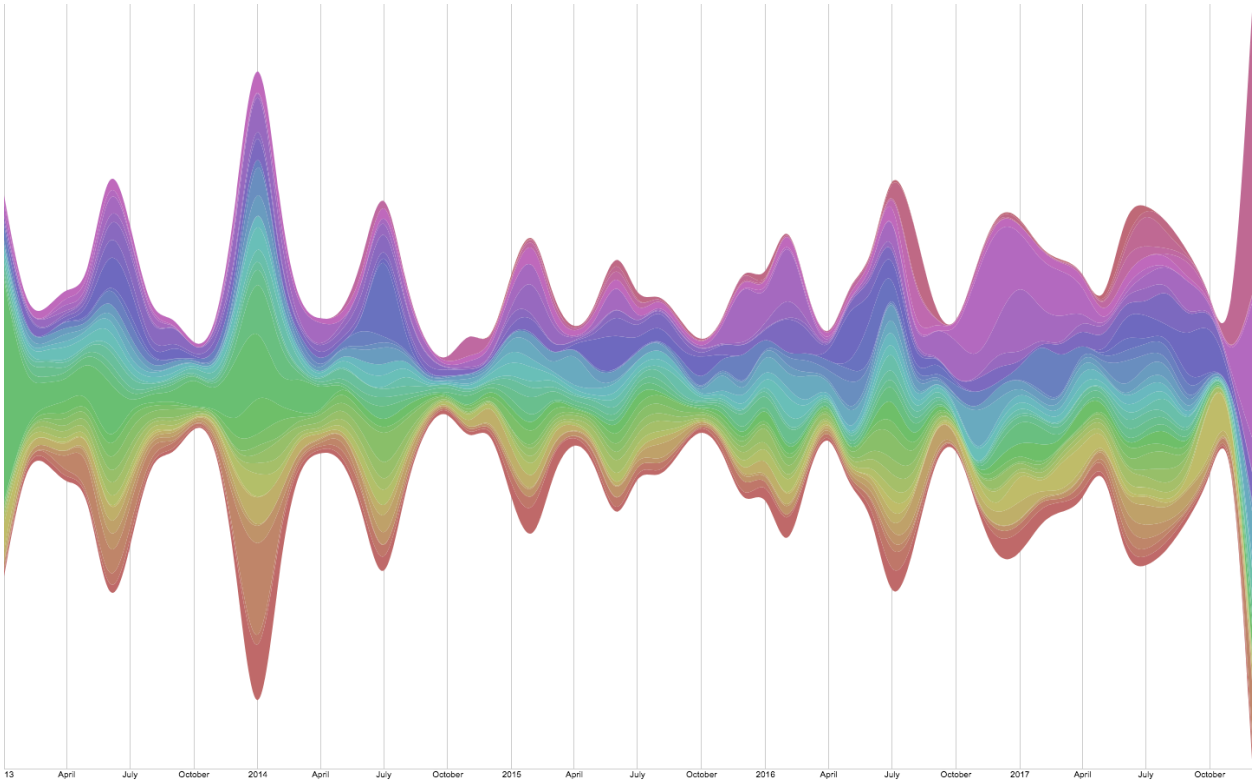
At this point in the process we received in class feedback regarding our streamgraph design and made several finalized visualizations using this feedback. We won't include all our images however they can all be found on our GIT repository under Streamgraphs -> Final Visuals.



Our final version streamgraph visualization showing average delay time per flight over time for airports using data for only New York airports. As you can see it is easier to find which airport goes with each color and aldo much simpler to see which months of the year the various spikes/dips occur in.



Streamgraphs comparing percentage of arrivals delayed over time (left) to percentage of departures delayed over time (right) for airports in New York. As you can see, the streamgraphs are almost identical which makes a lot of sense since presumably if an arrival comes in late, the next flight using that plane will also need to be delayed.



Our final version streamgraph visualization showing average delay time per flight over time for airports using data for all Northeast airports (Northeast = ME, VT, NH, MA, RI, CT, NY, PA, NJ). This makes for a good middle ground visualization as far as quantity of data goes since it gives a good general idea of overall trends like the one depicting all U.S. airports but it also is still possible to see how much impact individual airports have on overall trends, even if you can't really tell which airports in particular they are.

Peer Feedback:

Feedback was divided into two parts, one involving the streamgraphs and the other involving ranking through LineUp. With the LineUp review, most of the people interviewed were very interested in the data, as many people fly to get to the area. With the main focus being the preservation of the data, most people were very happy with what we had. The main issue many people had was the numbers being incredibly all of the place, with 15 different categories of delays and cancellations cluttering up much of the page. In addition, many groups asked for clarification on what certain categories meant, or other details. Using this information, I heavily processed the data, condensing down a large amount of the numbers into three distinct categories, while also properly labeling most of the numbers and data in a clear and distinct way. I also discovered many faults with the data through this further preprocessing step, creating something I can actually talk about as a large source of error.

As for the streamgraph side of things, we were mainly asking people for feedback regarding streamgraph layout, design, and readability. People had some very helpful advice as far as this goes to help improve the quality of the streamgraphs. The biggest piece of advice people gave was that the x-axis was confusing to interpret and follow. At that time, the visualization we had included an x axis with intervals every half year and year values listed as 2013, 2013.5, 2014, 2014.5 and so on. This labelling

scheme proved very confusing for people and they had a lot of difficulty determining when in the year each point on the streamgraph was. We modified to x-axis to make it clearer by labelling the start of each year (January) with the year number, and then including intervals every 3 months with the month name (so April, July, and October) and we found that this made it generally much simpler to follow the data. By far the biggest piece of feedback we received was that it was difficult to determine which airport corresponded with what color and that we should include a legend to simplify this. As we had it before, the streamgraph itself was labelled with the 3 letter airport code on each colored section. People had issues with this since the labels were scattered throughout the visualization rather than all being in one place and some of them could get difficult to read when they were in thin sections of the streamgraph. People also disliked the 3 letter airport codes since people don't typically know the abbreviation for their local airports off the top of their head. So, we added a legend in on the side of the visualizations and that could easily be followed to find which color represented which airport. We also opted to remove the labels on the streamgraph itself entirely since they made it look fairly messy and they were redundant with the legend there anyway. To get the names of the airports rather than simply the abbreviations, we added code to the parser that went through the file listing which abbreviation was for which airport and used this info to create a more intuitive legend that had actual airport names and not just 3 letter codes. Obviously though, we didn't add a legend for the all airport visualization since the legend would be enormous and it would do very little seeing as there are about 50 shades of each color in that visualization. Since the bigger visualizations were only intended to pick out overall trends, we opted to not include a legend for those and to use the legend just for the New York airport visuals.

Implementation:

The technical implementations did not utilize D3, as we wished to avoid using D3 due to prior experiences. In addition, since our data wasn't being dynamically updated (last update was beginning of this year), we felt as though we did not need to make the data pull information from the website, as it would only need to update once every year. Most of the data parsing and preprocessing was done in a C++ script for the LineUp app, as many of the lines randomly were missing data or simply created lines of data without any information behind it. The script I wrote filtered all of those lines with bad data (perhaps creating a source of error), and sorted many of the numbers into larger categories, clearing up a large amount of clutter and making the whole ranking look much more visually appealing. Otherwise, LineUp was a fairly simple program to use, and the data was provided by a government website, which needed a decent amount of preprocessing in order to be useful, especially with the numerous amounts of errors within the file itself.

The technical implementation of the streamgraph section was slightly more intricate. This involved first writing 2 python scripts, one that slowly went through almost 30 million lines of individual flight data and recording the necessary information on a month by month basis for each airport, and another that took that information and converted it into the various csv files that would go into the streamgraph generation tool. Once that was done, each visualization had to be run through the streamgraph generation tool and then the legend had to be manually edited into the image to make the final visualization. The most challenging part of this section was the data collection and manipulation. The data was all available from the website mentioned in the references section, so we had to first go

through and download the 60 csv files (1 for each month of data we were using) and then write the parsers necessary to process all that information. There were several lines of incomplete data that caused parsing issues and they were slow to find because of how slow it was to run the code each time to test it.

Division of Work:

For the division of work in this project the main way we split the work was each of us took charge of one of the ways we wanted to visualize this dataset. Casey took charge of the LineUp visualizations and was responsible for most of the work there and Euan took charge of the streamgraph visualizations and was responsible for most of the work in that section of the project. Other than that, we spent time working side by side to find the necessary data and write up the proposal at the start as well as for finalizing the visualizations and preparing the report and presentation at the end. Additionally, even though our project idea eventually changed, during the first week and a half, Casey spent a large amount of time trying to collect data on school closures and delays while Euan started working on finding and collecting data regarding weather in the Albany area over a long period of time. As mentioned we eventually had to abandon this idea but we figured we'd mention who did what work on that idea anyway.