Project 4 – Interactive Visualization using D3

Nathan McIntosh

Due November 3, 2020

1 Project Description

For this project, I picked three datasets from Kaggle Datasets. To create visualizations with the D3 library, I chose to use the python plotly library, and in particular, the plotly express sub-library. I chose python for this project because it is the language I have the most experience with. I chose Plotly as it seems to have an excellent interface to the underlying D3 library, and thorough documentation.

2 Datasets

2.1 Dataset 1: Udemy Development Course Data

id	avg_rating	num_published_lectures	discount_price_currency
title	avg_rating_recent	num_published_practice_tests	discount_price_price_string
url	rating	created	price_detail_amount
is_paid	num_reviews	published_time	price_detail_currency
num_subscribers	is_wishlisted	discount_price_amount	price_detail_price_string

Table 1: Dataset 1 Columns

This is a relatively straightforward dataset; all of the columns seem self-explanatory. Some of the perhaps more interesting columns here are `title`, `is_paid`, `num_subscribers`, `avg_rating`, `avg_rating_recent`, `num_reviews`, `num_published_lectures`, `num_published_practice_tests`, and `discount_price_amount`.

2.2 Dataset 2: Amazon Top 50 best selling books from 2009 to 2019

Name	Price
Author	Year
User Rating	Genre
Reviews	

Table 2: Dataset 2 Columns

The `name` and `author` columns are fairly self-explanatory. `User rating` refers to the average rating out of 5. `Reviews` is the total number of reviews. `Price` is in USD. The `Year` goes between 2009 and 2019. `Genre` is either fiction or non-fiction.

2.3 Dataset 3: COVID's Impact on Airport Traffic

The columns `Date`, `AirportName`, `City`, `State`, and `Country` are all self-explanatory. `AggregationMethod` refers to the smallest unit of time aggregated, in this case, all data points are aggregated at the `daily` level. `Version` is the version of the dataset, in this case, version 1. `PercentOfBaseline` is the proportion of trips

AggregationMethod	AirportName	City	Country
Date	PercentOfBaseline	State	Geography
Version	Centroid	ISO_3166_2	

Table 3: Dataset 3 Columns

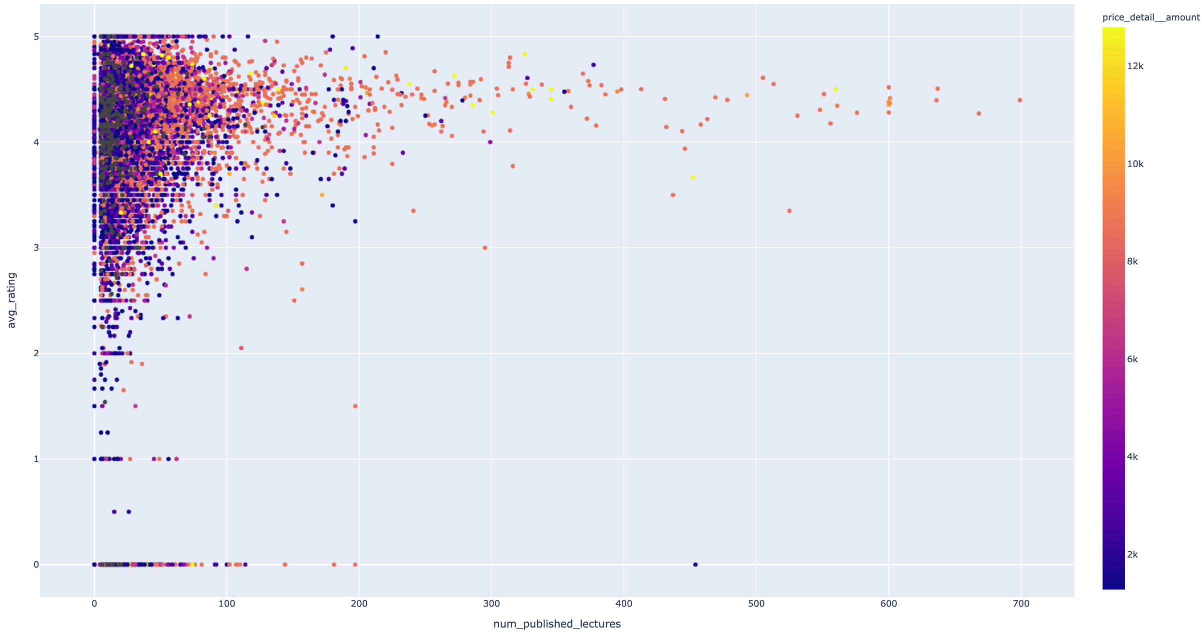
on this date as compared to average number of trips on the same day of week in baseline period of 1st February 2020 - 15th March 2020. `Centroid` is the geography representing centroid of the Airport polygon. `ISO_3166_2` is the ISO-3166-2 code representing Country and Subdivision. `Geography` is the polygon of the Airport that is used to compute this metric.

3 Visualizations

3.1 Visualization 1

For this dataset on Udemy courses, I chose to investigate how the availability of lectures affected the average rating of a course. As a third "dimension", I also chose to plot the course price as a color. These went from free, to over 12 thousand Indian Rupees. There are several types of interactivity available to the viewer: hovering over a point will show all data about it; the axes are rescalable; and the user can zoom by drawing boxes or clicking and dragging to zoom vertically or horizontally.

Figure 1: Visualization for Dataset 1: Udemy Course Reviews

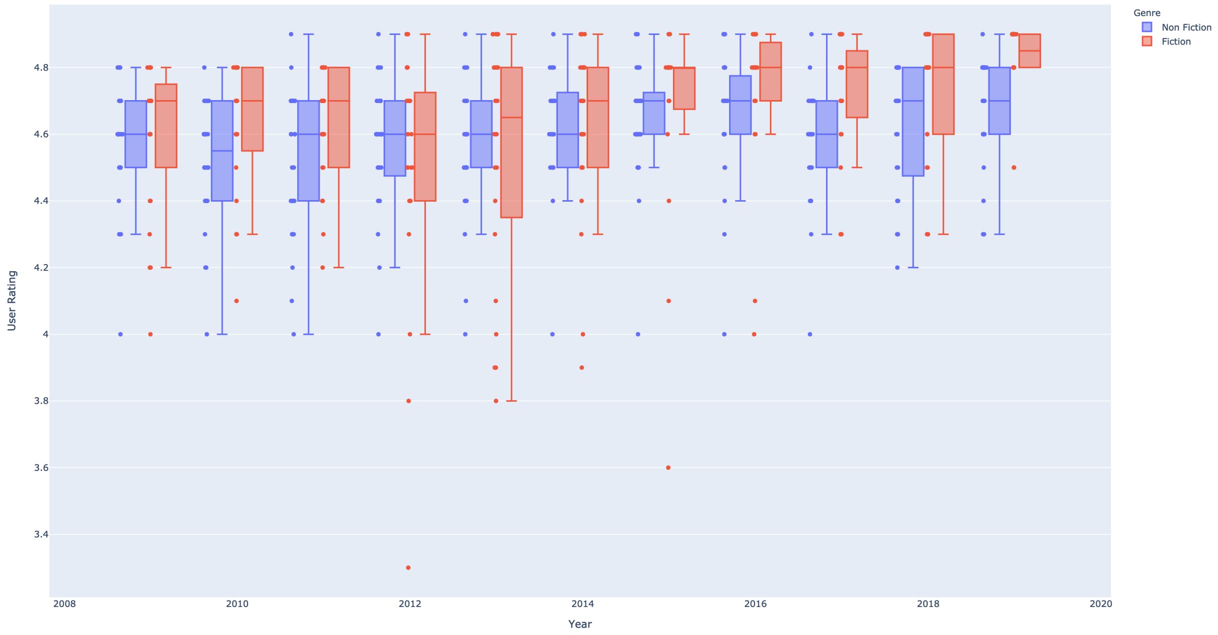


3.2 Visualization 2

For this dataset on the best selling books on Amazon from 2009 to 2019, I chose to make a grouped box plot. Since this is a timeseries dataset, I wanted to display that on the x-axis, and see the general trend of reviews. There are also two well defined groups: fiction and non-fiction, and I wanted to see if there were any noticeable disparities between the two. The end result is a grouped box plot. I like this because it shows statistical measures of user rating. For instance, we can see that fiction book are almost always rated more highly than non-fiction. Along with the boxes, there are also strip plots. These dots show the individual data points of the books. The user can hover a point to see all the information about a book. Like visualization 1, the user can also zoom and re-size as earlier mentioned. Finally, the user can click on a genre listed in the legend to filter it out: clicking on fiction will convert the plot to one of only non-fiction. Clicking on it again will bring it back.

One of the things I found most interesting about this plot was that many of the best selling books had not been published the same year, but rather may have been from many years ago. For instance, one of the best selling books of 2016 was To Kill a Mockingbird, which was in fact published in 1960. From manual inspection of a few years, I was unable to find this pattern in any non-fiction books. Perhaps there are many classic fiction books which sell well in any year, and relatively few classic non-fiction books that continue to sell well.

Figure 2: Visualization for Dataset 2: Amazon Best Selling Books 2009-2019



3.3 Visualization 3

This dataset contains the geographic data of airports, and their relative traffic levels to pre-COVID times over 215 days. Time was again a primary dimension of this dataset, but I wanted the user to interact with the time dimension more viscerally. To do this, I created an animation of the baseline traffic over time. As time progresses and the level of traffic changes, the size of the dot representing each airport changes.

This visualization leads to a few insights.

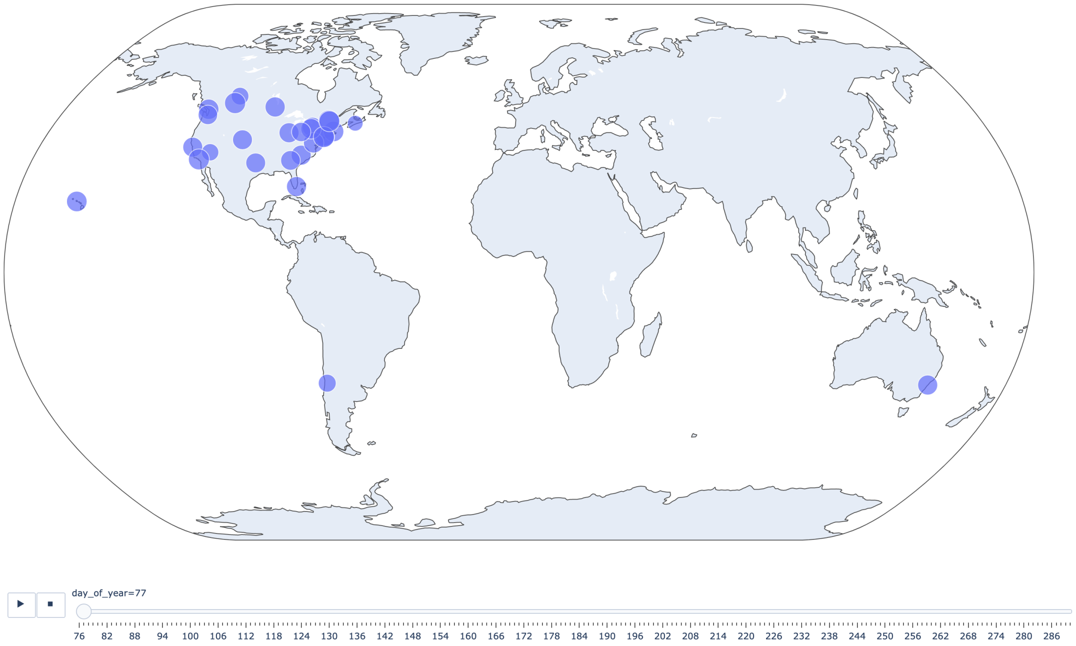
1. Most airports' traffic fluctuates a fair amount over time. Could this be due to weekend vs. weekday travel differences?

2. Hawaii's Daniel K. Inouye International Airport fluctuated much less than most other airports. Is leisure travel relatively steady during COVID? I was under the impression that business travel was less affected than leisure travel.

3. On day of year 251 (September 7) almost all airports had drastically reduced traffic compared to the days around them. I was not able to spot this pattern again while watching the data. What happened

on September 7?

Figure 3: Visualization for Dataset 3: COVID19's Impact on Airport Traffic


▶
■
day_of_year=77
76
82
88
94
100
106
112
118
124
130
136
142
148
154
160
166
172
178
184
190
196
202
208
214
220
226
232
238
244
250
256
262
268
274
280
286