Report for 3651 Films in Almost 3651 Words

For my project I used the data set provided at: <https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset> and a data set showing inflation rates over the years: <https://inflationdata.com/Inflation/Consumer_Price_Index/HistoricalCPI.aspx?reloaded=true> to create data visualisations with the focus on the genres in the film datasets provided.

My first task was to clean the data set to make it usable, more of this is documented below. To clean the data set I had to correct character changes where letters outside the English alphabet had been altered inside the scraping process, these can be seen in the original data set. I then had to filter out data sets that weren’t complete, duplicate films and then to remove any films that weren’t English language, as the budget was not in US dollars.   
  
Now I had a clean data set, I outlined what I would like to achieve with the data to find patterns. I thought genre and how it related to the other columns would be interesting to explore. This meant I would need a way to count the occurrence of each genre within each film. I did this using the Pandas library for python in the CountGenre.py file to create a new column for each genre which would be filled in with either a 1 or a 0 depending if that genre was present in the film. This basically created a new dataset for me to work with from the one column ‘genre’ in the original data set which had a list of different genres present in one cell for each film.

I began now by outlining what graphs I wanted to make in order to explore the data. I decided on:

1. A bar graph that illustrated which genres occurred the most in the data set.
2. A bar graph that illustrated the gross, I ended up also including the budget   
   as I felt this was important to representing the information.
3. A line graph that showed how genres changed over time, this ended up as an area graph for reasons I will explain later.
4. A Chord or Sankey diagram to show how different genres were linked, as films normally have more than one genre.

There were a couple of things that I needed to do for all the graphs, one was to find a colour scheme which allowed me unique colours for each genre that all looked good together and the other was to make sure all of the genres shown had a big enough representation. I cut Film-Noir, Biography and Mystery as there wasn’t a big enough sample size for them. For solving my colour issue I used the generator here: <http://tristen.ca/hcl-picker/#/hlc/6/1.05/CAF270/453B52> it allowed me to put a lot of different variables. I accepted by having 18 variables a rainbow theme was going to be something hard to avoid if I wanted distinct colours, and to be honest I quite like the slightly subdued rainbow colours it gave. With 18 genres it was going to be confusing whatever colours I chose and was going to have to implement interactivity to gain clarity of information.  
  
I started of with organising all the data for the graphs, but will go through the process for each graph.

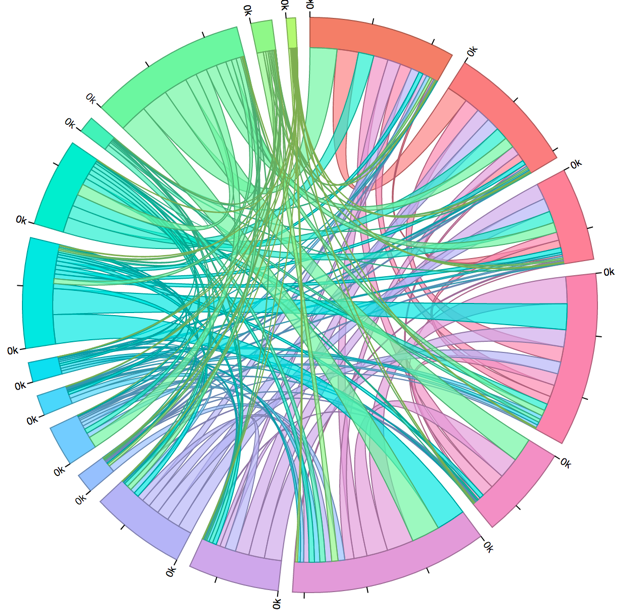
Chord Diagram

To start with I wasn’t sure whether to create a Chord diagram for my work or a Sankey diagram. The Sankey diagram didn’t give itself much understanding as the chord diagram, I thought, but I had only ever seen chord diagrams which showed flow in one direction, but the data I had flowed both ways from the source and the target arc. I decided to try and chord diagram and find a solution for the problems I had raised.

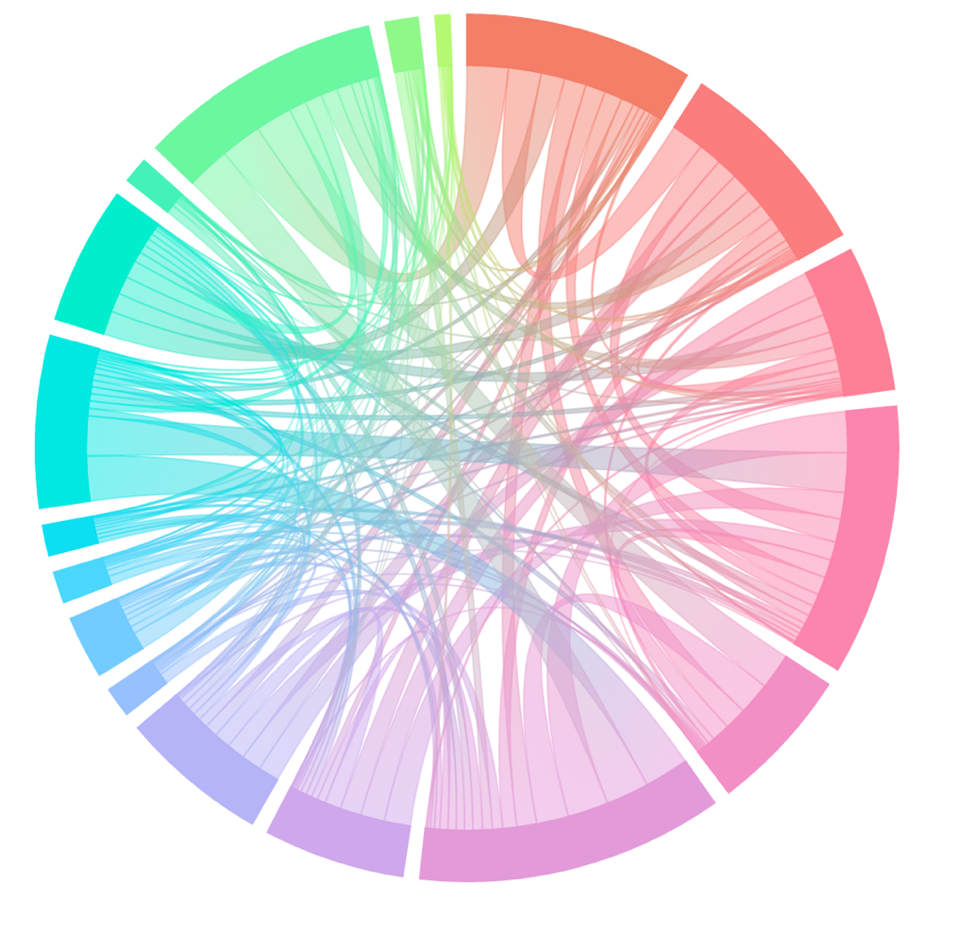
To start with I looking at what format I needed the data to be in inside the .csv file, I thought something like this was the solution:

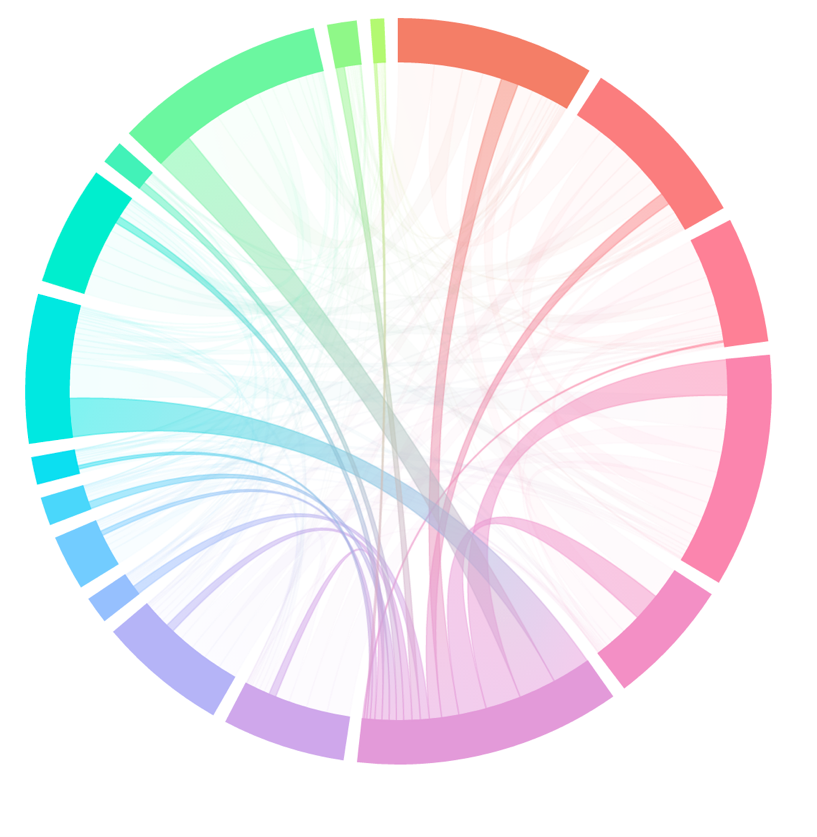
|  |  |  |
| --- | --- | --- |
| Action | War | 1 |
| Action | Romance | 2 |
| War | Romance | 3 |

I started to try and create a .py file to sort this, but realised I could do it fairly easily in excel by using an IF function to see if the adventure column = 1 and the action column = 1 for that film, then to write 1 into its own column. I took this data for each genre and added to the chord matrix.csv file.   
  
I soon realised after looking at this example by Mike Bostock: <https://bl.ocks.org/mbostock/4062006> that I would need to reformat the data into a matrix, you can see this process in the excel sheet. I hard coded the matrix into the chord diagram.  
  
Mike Bostock’s example was great to letting me understand how a chord diagram works, how it was a series of arcs with chords linking from a source to a target. I changed numbers on his example to see how it all worked then used bits from it as a basis for my own diagram.

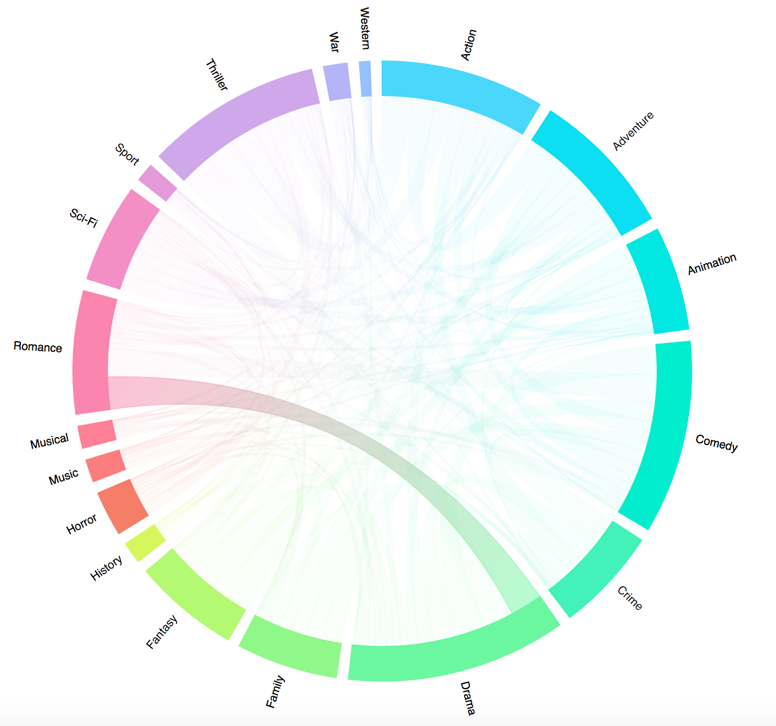
  
Here you can see my first attempt at the diagram, it is messy, you can’t really tell what is going on. The colours of the chord don’t make sense as it isn’t following. I also don’t think having ticks around the outside make sense as you can’t tell an exact value for each chord even with them.

The first thing I did was to strip is back and try and make it more visually appealing, if it was going to be interactive, it needed to be pretty enough to perk the curiosity of the user to play with the data.

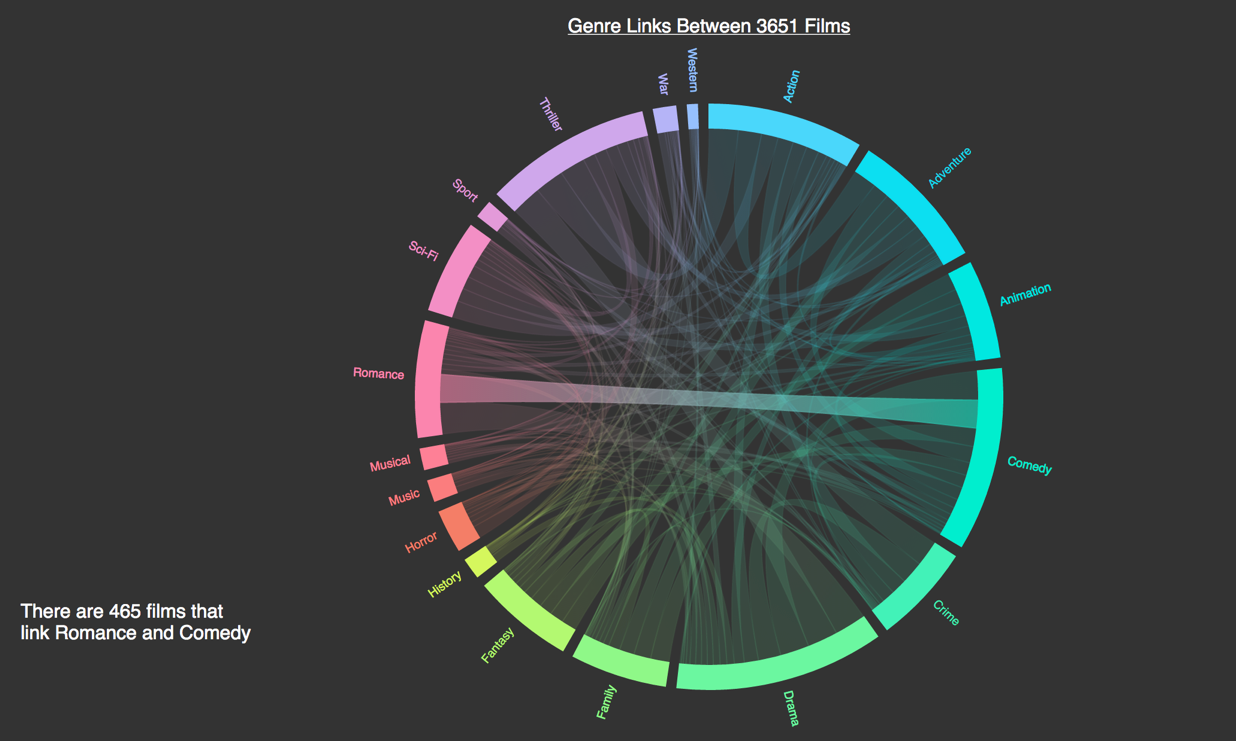
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
The most notable change here is the discovery of the tutorial here: <https://www.visualcinnamon.com/2016/06/orientation-gradient-d3-chord-diagram.html> by Nadieh Bremer who has since become a D3 idol for me. She creates a colour gradient which follows the arc of the chord by figuring out the angles, meaning that now my source and target can remain the colour of their own genre.  
  
There graph was still quite busy but I didn’t want to remove any other genres, as it would be deleting the very data I wanted to display. I set myself two goals, one was to add opacity so when you hover on a group arc it would make any chords not in that group opaque and to make it so if you hover over a particular chord, everything else will become more opaque and the value of that chord will be shown.

Here I created a function which uses the index values in the matrix to make anything not in a particular index when hovering opaque – this adds clarity and lets the user interact with the graph.



I also played around with the idea of having a grey background as I think it makes the colours pop out more. I went back to white but finally decided on a darker grey as you will see later.

In this image I have made it so that when selecting a particular bar it will keep its opacity, while reducing the other arcs.   
  
The other thing you will notice is that the colours have changed. Although I wanted them to be in the order of the colour spectrum in the circle, I also want the genres to relate somewhat to the colours. I thought it would add to understanding if romance was in the pink/red spectrum and so cycled through the colours until this was the case.

I finally added a tooltip so that when you hover over a chord its value is displayed and it tells you what films are linked. Hopefully, this will also help people see what the graph is displaying.  
I also reverted back to a dark grey colour as I think it makes the graph more visually appealing and stand out more. I have also changed the genre titles for this reason.

Bar Graph to Display Gross per Genre

The Stacked Area Graph started off as a Line Graph, before I decided that wasn’t the best way to display the data.  
  
Before I could think of creating a graph, I had a lot of work to do with the data. First I would have to adjust all the budgets and gross’s to reflect the impact of inflation, it wouldn’t be fair to compare a film made in the 60’s to the 70’s. As all the prices were in USD as long as they were English language films, I haven’t included the prices as adjusted for exchange rates at the time. I use the inflation CPI index here: <https://inflationdata.com/Inflation/Consumer_Price_Index/HistoricalCPI.aspx?reloaded=true> as another data set.

I needed to use Pandas to join the data sets together to calculate the inflation trends. I wanted the CPI to appear after each film for that films year so I could compare it to the CPI for 2017 as it stands at the moment.   
  
After first running the python code used to map the two data sets I got a wall of errors:

Traceback (most recent call last):  
  
 File "/ Documents/Data Vis/Coursework2/MappingCSV.py", line 2, in <module>

film = pd.read\_csv("/Documents/Data Vis/Coursework2/movie\_metadata.csv")

File "/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-packages/pandas/io/parsers.py", line 646, in parser\_f

return \_read(filepath\_or\_buffer, kwds)

File "/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-packages/pandas/io/parsers.py", line 401, in \_read

data = parser.read()

File "/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-packages/pandas/io/parsers.py", line 939, in read

ret = self.\_engine.read(nrows)

File "/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/site-packages/pandas/io/parsers.py", line 1508, in read

data = self.\_reader.read(nrows)

File "pandas/parser.pyx", line 848, in pandas.parser.TextReader.read (pandas/parser.c:10415)

File "pandas/parser.pyx", line 870, in pandas.parser.TextReader.\_read\_low\_memory (pandas/parser.c:10691)

File "pandas/parser.pyx", line 947, in pandas.parser.TextReader.\_read\_rows (pandas/parser.c:11728)

File "pandas/parser.pyx", line 1049, in pandas.parser.TextReader.\_convert\_column\_data (pandas/parser.c:13162)

File "pandas/parser.pyx", line 1108, in pandas.parser.TextReader.\_convert\_tokens (pandas/parser.c:14116)

File "pandas/parser.pyx", line 1206, in pandas.parser.TextReader.\_convert\_with\_dtype (pandas/parser.c:16172)

File "pandas/parser.pyx", line 1222, in pandas.parser.TextReader.\_string\_convert (pandas/parser.c:16400)

File "pandas/parser.pyx", line 1458, in pandas.parser.\_string\_box\_utf8 (pandas/parser.c:22072)

UnicodeDecodeError: 'utf-8' codec can't decode byte 0x92 in position 4: invalid start byte  
  
I thought this might, at least be in part down to the random characters in the file such as Œ æ which is for some reason at the end of each of the film titles. I cleared this up, and then realised a lot of the names when using characters such as

È, which aren’t in the English alphabet, it replaced it with a ©, excel makes it easy to find and correct these changes.

It seems the ‘utf-8’ codec error was because of the characters in names with characters outside of the English language characters. I found a fix by setting the encoding:  
film = pd.read\_csv("movie\_metadata.csv", encoding='mac\_roman')  
The file is Mapping CSV.py if you would like to look at the code closer.

To find out the value after being adjusted for inflation for the value in 2017. For the USD the Avg CPI was 243.22 so the equation is:

Where x is the budget or gross in 2017 figures, y is the budget or gross in the year of release and z is the CPI on the year of release.

I was starting to get concerned about the data at this point, as other sources reported The Polar Express’ gross at around $162,775,358, while my data reported it much lower – looking at the iMDB website where the data is scraped from it seems this is a one off internal error on their part as it is the value they account on the website.

Now that I had an adjusted budget and gross for each film I needed to find out a way of seeing which genre is the most profitable. I started out in excel by creating a simple if function which says if the Action cell for example has a 1, meaning the film has the action genre then fill the cell with the adjusted gross value. GenreCountBudgetCorrect.csv has the process for this as I repeated the same thing for gross and budget.

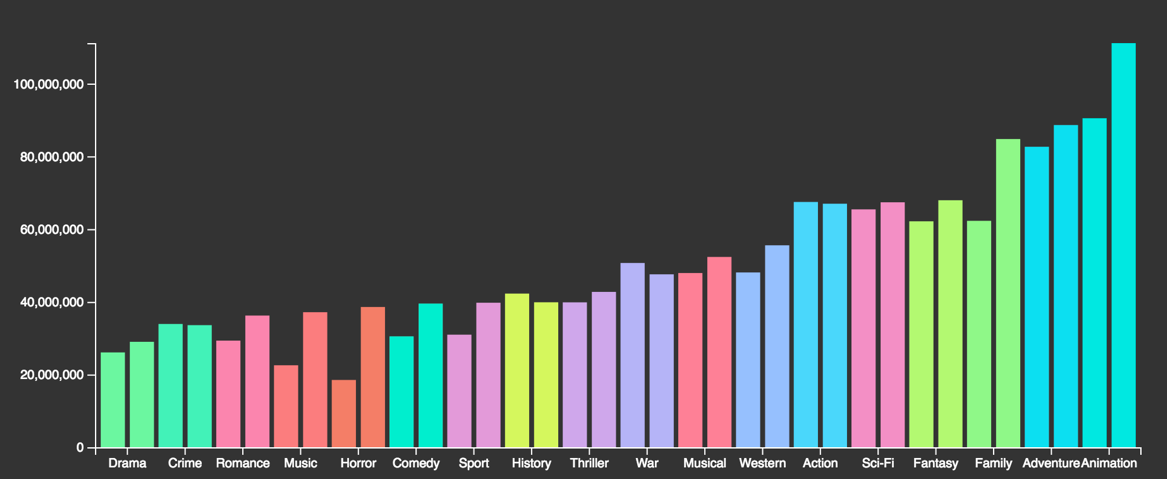
Now I could simply add all the values together and divide them by the amount for each genre to find the mean, but I decided to create a boxplot to see what would be the best method.

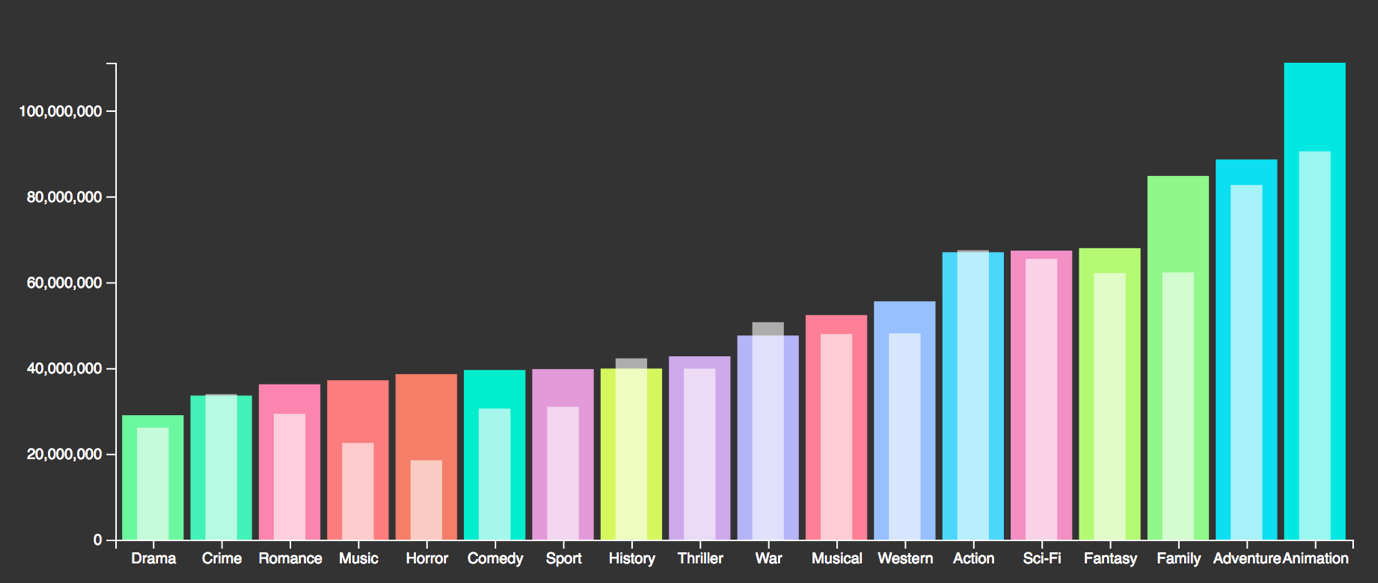
Now I had the data I could start actually making the bar chart.

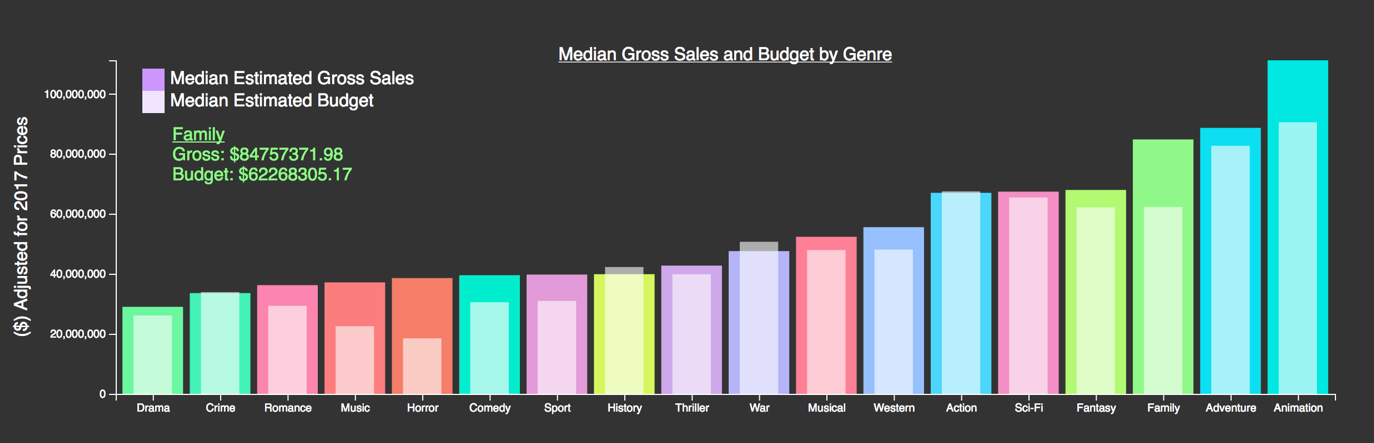
I wanted the hexadecimal value for each genre to stay the same through all the charts, so I am adding it to the .csv file before I order the bars by value.

I soon realised Musical hasn’t been added to the chart, I corrected this and found the median. Luckily by keeping the GenreGrossBoxPlot.py files it was fairly quick to find the correct medians.

I plotted two bars showing both the median estimated budget and median gross sales for each Film Genre, by changing the widths I can show them alongside each other easily. I didn’t think this is the best way to display them, as the gross is the most important factor and it makes it hard to differentiated between them.

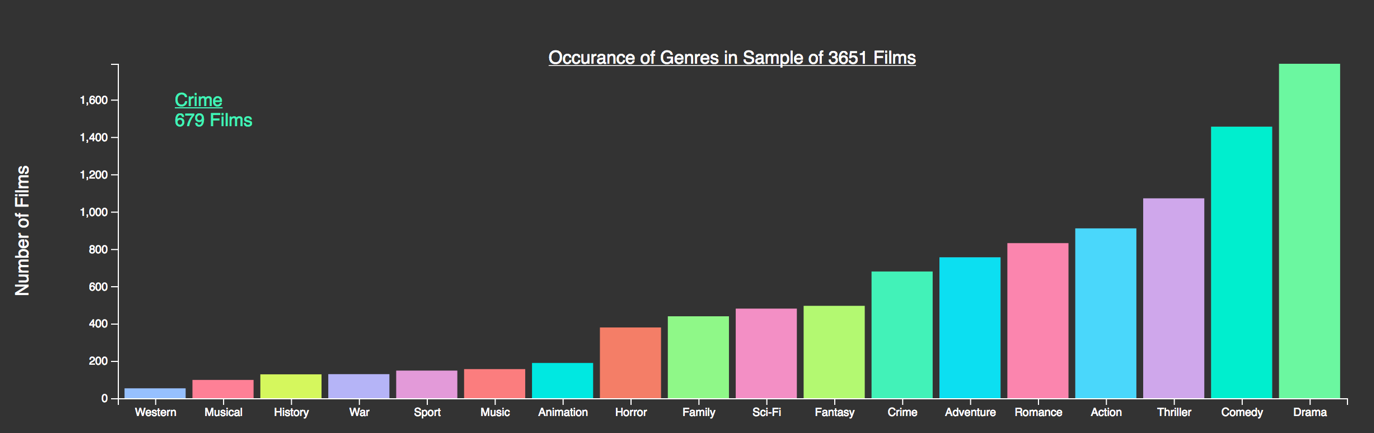


By playing with the opacity the budget is now shown as white with opacity over the top of the gross bars. The main focus of the graph is the gross of the films, that is why the genres are ordered by gross. I think it is important to see the budget of the films to see it in perspective of how much was spent

I added a tooltip and some other extra features to the graph that you can see below.  
The legend shows what the pale white means, but it isn’t any of the colours in the graph to try and not be confusing. The value for the budget and growth is also displayed when you hover over.

Bar Graph to Display Most Popular Film

Creating the popularity bar graph was fairly simple, as I could just recycle most of the code from my other bar graph.

It shows how many times each genre occurred in the data set. I think this graph is important, not just for what it shows, but for clarity for those viewing the data as they can see the sample sizes for the median values so can understand that the value for drama has a bigger dataset than westerns, for example, so the median will be more accurate for the whole genre for drama than it is western.

Line Graph/Area Graph To Display Changes in Genre over Time

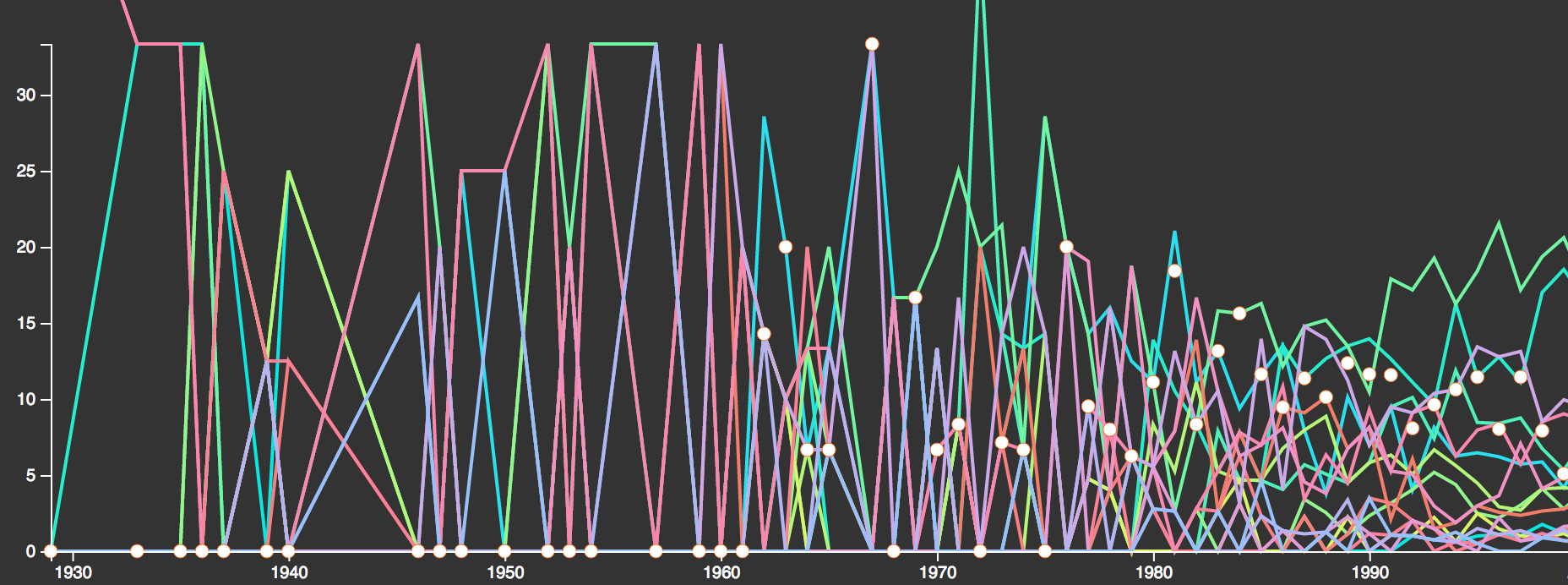
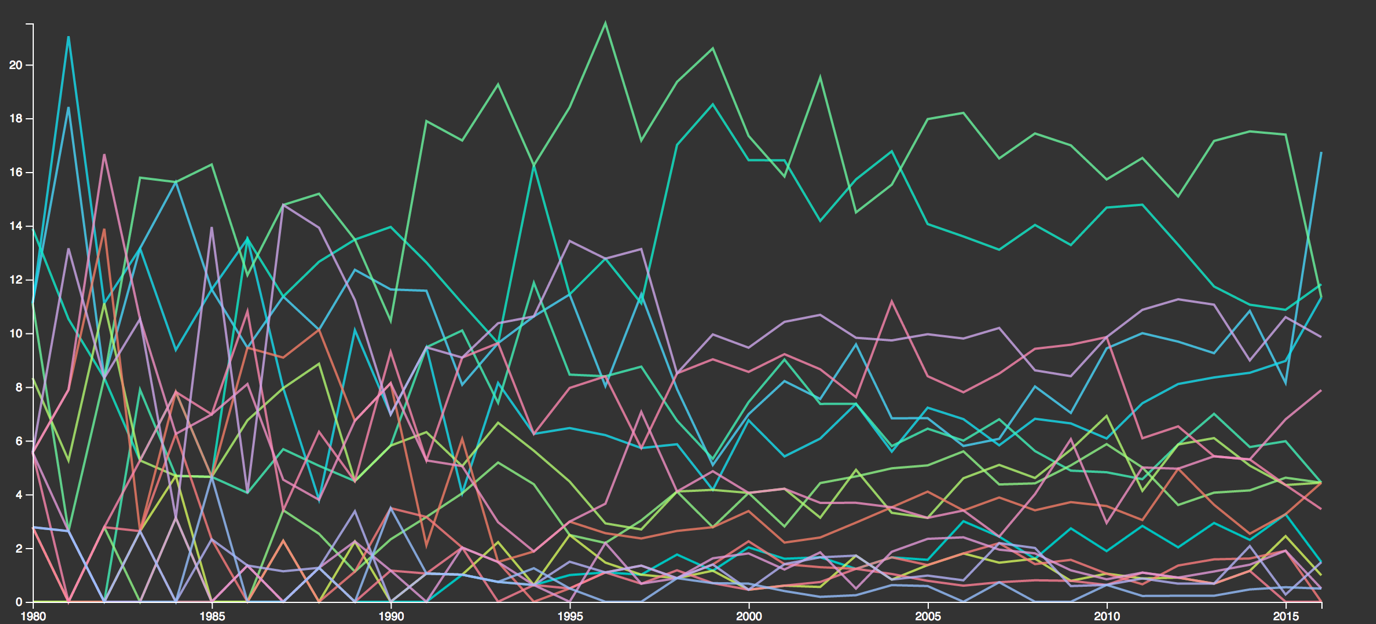
To do this I first had to find the total genre for each year. I made a script TotalGenreByYear.py which creates a sum of all the data for each year, in TotalGenreByYear.csv one of the problems is that more recent films have a lot more entries.

One potential solution for this is to do it by decade instead of year and to do it by distribution of genres within that period. I understood that the sample still makes the data problematic statistically as the earlier decades have a lot smaller sample.

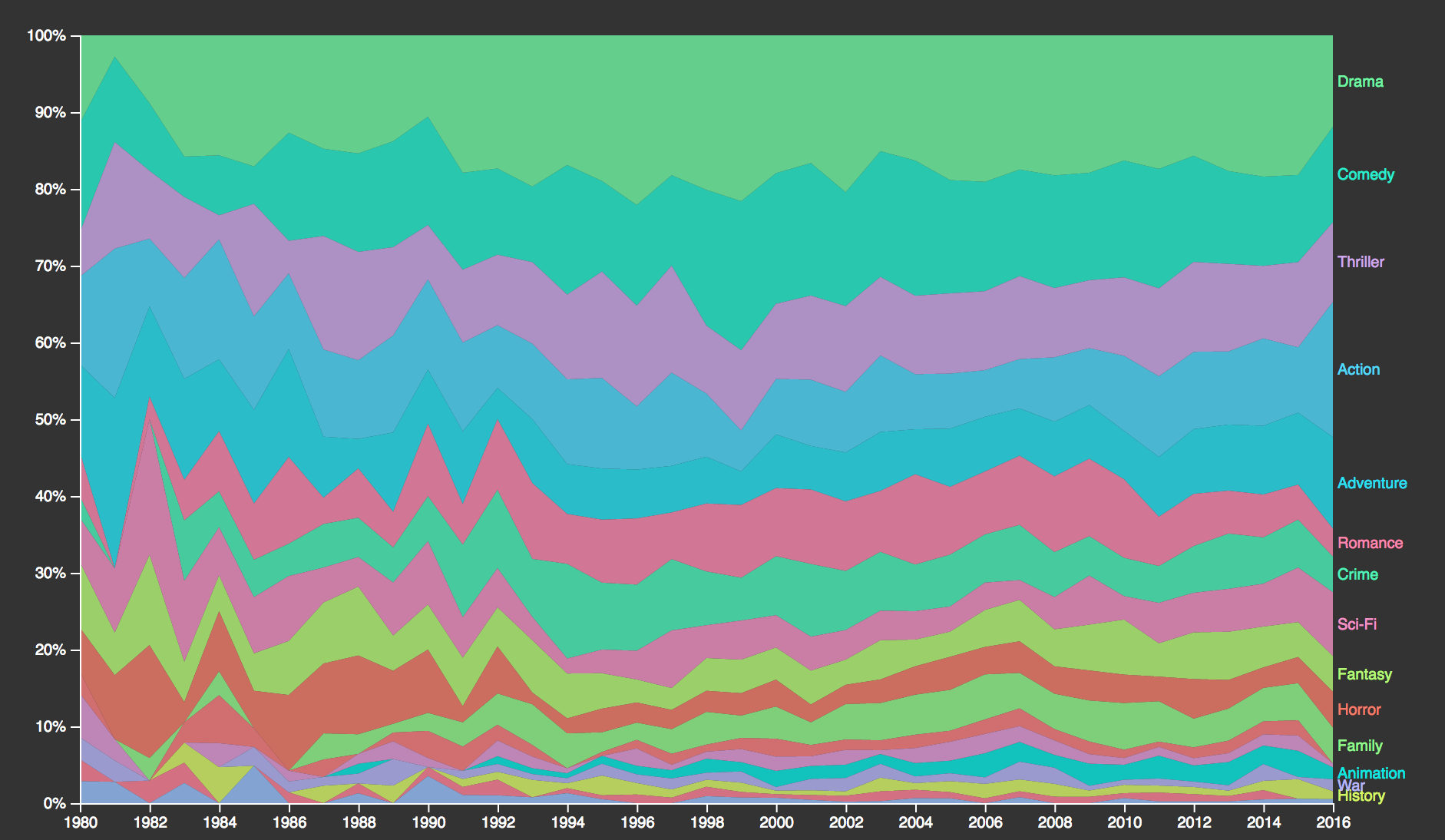
Another option is to just look closely at trends from the 80s-2010 in years as there is more data so I can find out what the Annual Breakdown of the 3651 films by their genre by working out what the total different numbers of genres there were for that year and then what percentage each genre made up of that.

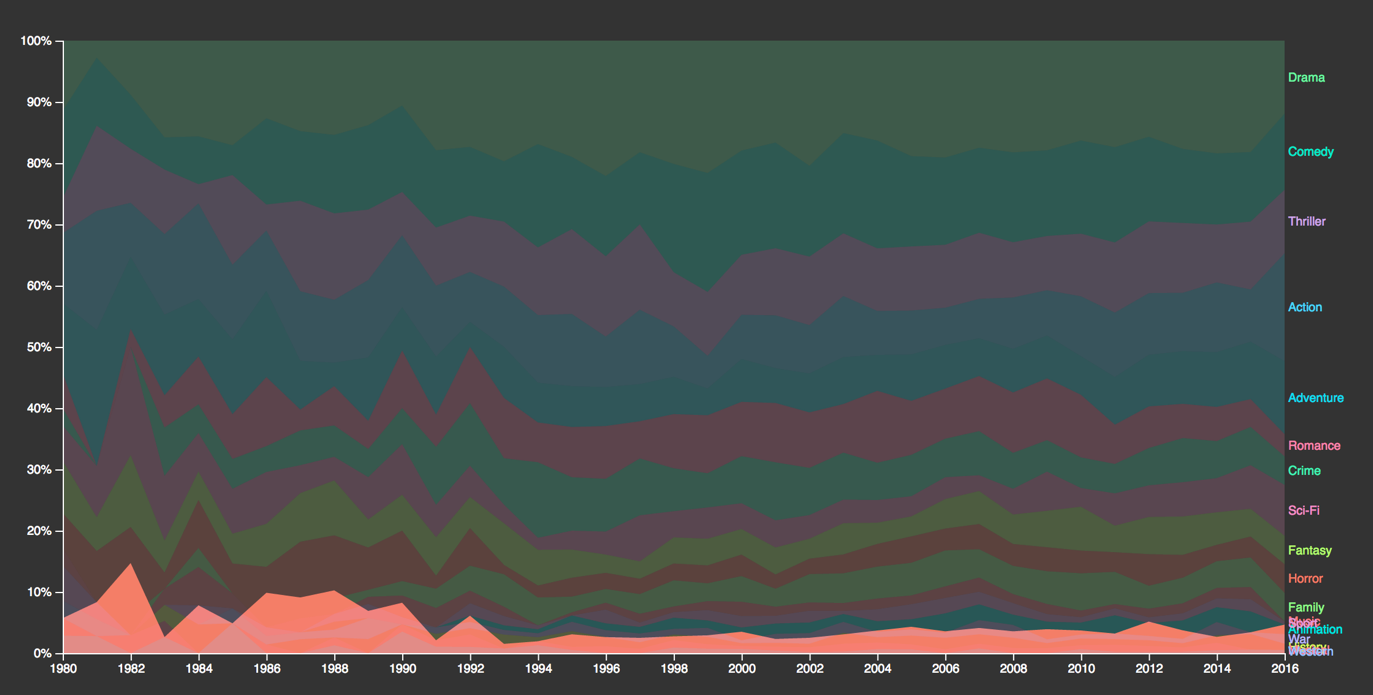
You can see the progression of these calculations from TotalByDecadeWorking.csv to TotalByYearProportion.csv to TotalByYearProportion1980Up.csv. TotalByYearProportion.csv illustrates why I couldn’t use the lower years, it was important to understand the limitations and flaws of my data.

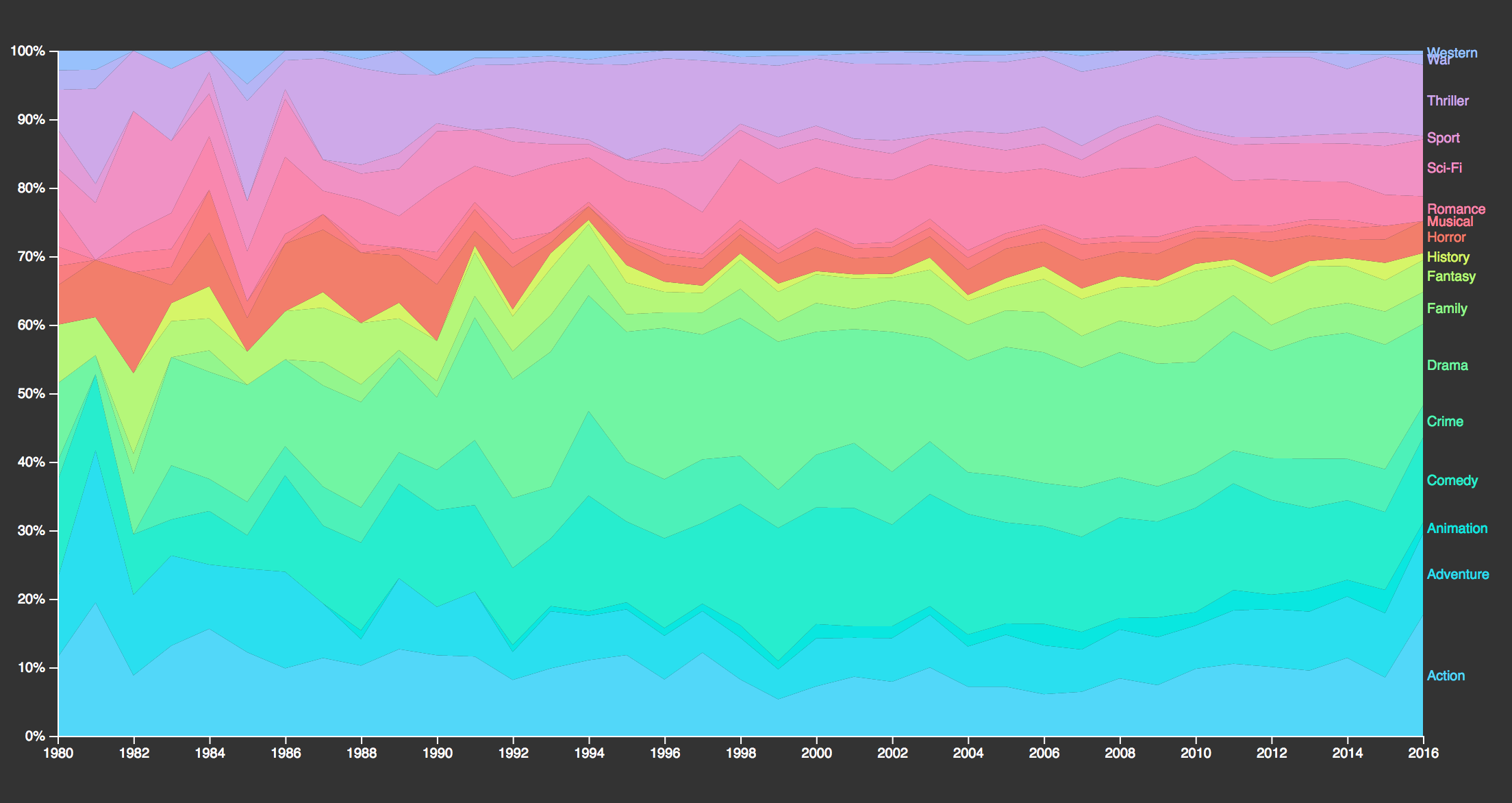
I started making the charts using all the years to check, as you can see by the graph below there wasn’t enough data to show any trends as the lines become a lot steadier as time goes on.

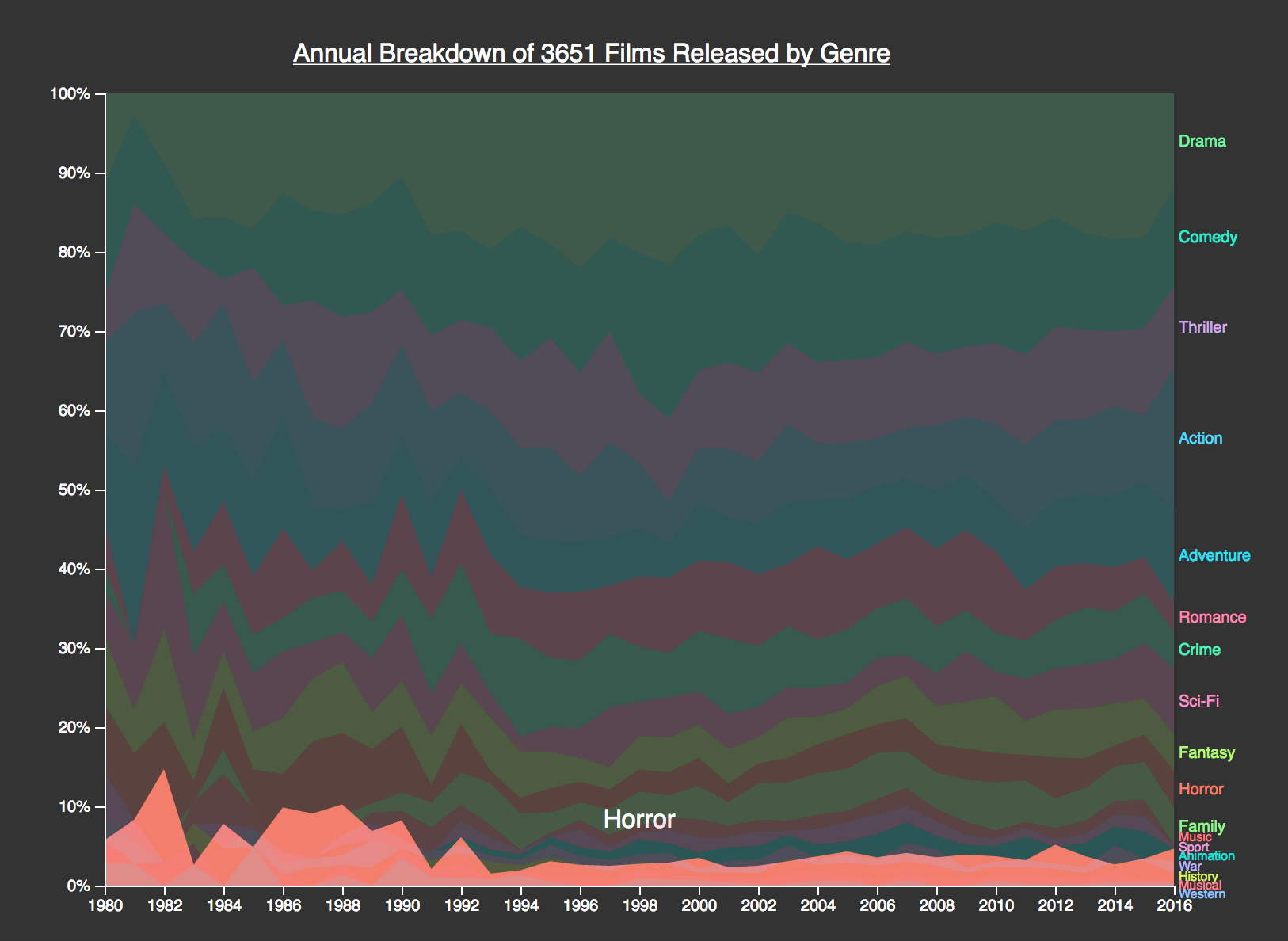
   
  
  
I then edited it to show the data from the 1980s onwards.

To solve the problem of overcrowding presented before you, I have decided that an stacked area graph would better present my data, as they all add up to 100%.

I have made a stacked area graph following the tutorial here: <https://bl.ocks.org/mbostock/3885211>  
  


One of the problems you can see here is that the baseline is affected and so it makes the values difficult to see how they have actually changed over time. I did some searching to find ways to combat this and found a good solution by Zache Thomae: <http://bl.ocks.org/zthomae/6384271>. The problem now is his version is in V3. How it works is by setting all of the area’s invisibility to low when hovered over and then drawing another another path onto the map. I attempted to do the same with my graph by seeing what process his functions went through and trying to use his solutions along with my own.

I adapted my graph so that you can hover over any part of if and see it with the baseline set to 0. One problem is that in ordering it by descending the text overlaps as all the low values are in one place. Without an order this isn’t so much of a problem, but the order adds more understanding to the graph. Below is the unordered version.

I decided to adjust the font size on the lower portion of the ordered stacked graph. This doesn’t look great and may be hard to read, but is the best solution I can come up with. I will add text when hovering so that it is still clear what part of that graph is showing.

This is my final version of the graph. I think it is a lot better as an area and by hovering over each genre it is clear to see how they have changed over time, for example the boom of horror films in the 80s following on from the 70s as shown above. Having the text appear as you hover also makes up for the small text on the side in my opinion.

Evaluation

Overall, I am pleased with how the project went. The resulting visualisations are fairly pretty to look at so should attract a user’s attention, particularly with the chord graph which came out better than I could have hoped.

One of the key issues I realised straight away was the data itself. It came from a scraper of IMDb and so had its own problems mentioned above. Through have to clean up the data I was becoming increasingly apprehensive, but by realising what the problems were I have adapted the focus of the project early on to solely be about genre. Understanding the limitations of my own data was vital in this project. Though it is a large sample of films, it is by no means exhaustive so I decided to avoid putting individual films or actors in, I cannot say that one film is the highest grossing of all time, because my data set doesn’t include all films. By understanding my data’s strengths, being that it was a large sample size I was able to focus my attention to its strengths. By extracting the genre data and making it into data which could then be analysed, when it couldn’t before, I was able to find interesting trends in data. Some of the trends aren’t ground breaking, and are what you would expect in some cases – nobody is going to be surprised at the ‘RomCom’ links, but they might be able to find some information they didn’t know, or encourage them to think more about why films are paired the way they are. Despite my data’s flaws I am glad I have made something with it that most people would be interested in looking at.

With the graphs, I think one of the biggest challenges has been setting new challenges for myself beyond what we learned in class. I wanted to show the visualisations in the best possible way and to create something that wasn’t the usual so that it would grab people’s attention. One of the problems I found were trying to understand how the different pieces of code worked by looking at the documentation. I found the best way was to play with other people’s code to figure out what everything did and then apply my own knowledge of D3. I think this flexibility of knowing that if I could think of something to do, there was probably a way to do it has let my data be presented in a way that people will be interested in it.

If I was to do the project again I would like to test the webpage on more people to see if they know that the graphs are interactive. I didn’t know whether to explicitly write on the page or to just hope people hover over it. I would also like to put more work in the webpage as it doesn’t work perfectly on lower screen sizes with the tooltips, so I will edit this in the future too.

One of the other things I would like to do is to explore the genre when including all the non English language films too and see how this varies between the different languages. I would also like to be able to look at what film ratings are most commonly linked to different genres and see what the most profitable ratings are, similar to the bar chart on genres. It would also be interesting to see how things like duration affect the budget and average ratings. Despite its flaws, the data set presents a number of opportunities which I haven’t had time to explore yet, but would like to do in the future.

After Note

During the process of creating my data project, I kept a word document called Data Diary which I referred back to a lot for this report, I have attached it alongside my code zip, but it isn’t edited so the structure is lacking and it is a bit rambling. Most of what is included there I have documented here.

Bibliography

Brown, Tristen. *Colorpicker for data.* N.p., 2017. Web. 25 Apr. 2017.

Bremer, Nadieh. "Data-Based Orientation For A D3.Js Gradient - Visual Cinnamon". *Visual Cinnamon*. N.p., 2017. Web. 25 Apr. 2017.

Thomae, Zach. "Stacked Area Chart With Focus On Hover.". *Bl.ocks.org*. N.p., 2017. Web. 25 Apr. 2017.

McMahon, Tim. "Historical Consumer Price Index (CPI)". *Inflationdata.com*. N.p., 2017. Web. 25 Apr. 2017.

Bostock, Mike. "Chord Diagram". *Bl.ocks.org*. N.p., 2017. Web. 25 Apr. 2017.