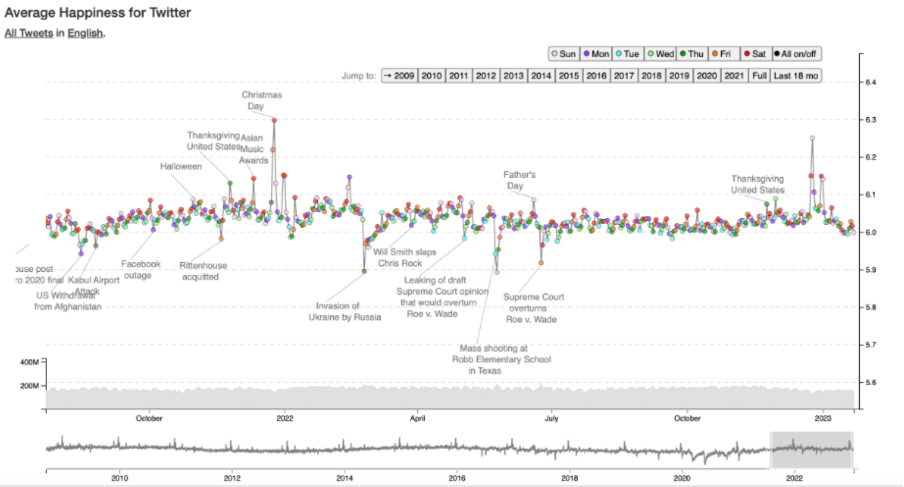
MySQL Assignment

Jenni Hutson

January 23, 2023

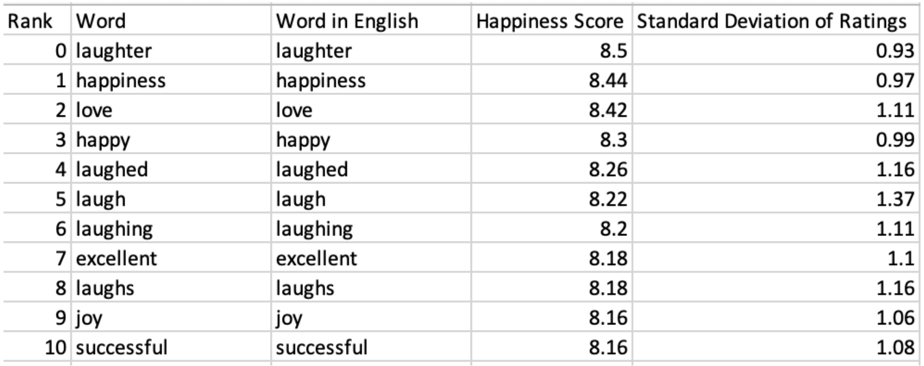
1 Description of Concept

I was inspired by the Hedonometer for my project, which is a project out of the Computational Story Lab at the University of Vermont. The Hedonometer takes a random sample of 10% of all tweets everyday and strips them for English words. These words are matched against a list of about 10,000 words with associated happiness ratings between 1-9, with 1 being the saddest and 9 the happiness. The word rankings were averaged from rankings given by Amazon Mechanical Turk workers. In this way, the Hedonometer can get the overall happiness level of Twitter on a given day. This is not based on context at the moment, but purely on the happiness of each individual word. Despite this, the Hedonometer does a pretty good job of capturing the sentiment of a large group of people, and their happiness map has clear inflection points for tragic and joyous events experienced at a large scale. From their website:



I was curious if happiness levels could also be detected in checkouts from the Seattle Public Library, and if they would also correspond to large events happening in the United States. My prediction was that mood would not respond as quickly as it did on Twitter to large events, but large events might have a more sustained impact on what titles were being checked out.

Luckily, the Hedonometer project provides their list of words with happiness rankings as a downloadable CSV:



Rank	Word	Word in English	Happiness Score	Standard Deviation of Ratings
0	laughter	laughter	8.5	0.93
1	happiness	happiness	8.44	0.97
2	love	love	8.42	1.11
3	happy	happy	8.3	0.99
4	laughed	laughed	8.26	1.16
5	laugh	laugh	8.22	1.37
6	laughing	laughing	8.2	1.11
7	excellent	excellent	8.18	1.1
8	laughs	laughs	8.18	1.16
9	joy	joy	8.16	1.06
10	successful	successful	8.16	1.08

and so on.

2 MySQL Query

My SQL query was very simple, as my primary interest was gathering as much time-based data as I could on titles checked out in the year 2022. The general query was as follows, and I later made more specific queries to see the impact on data:

```
SELECT
    cout, title, itemtype
FROM
    ‘spl_2016‘.‘outraw‘
WHERE
    cout LIKE ’2022%’
        AND itemType LIKE ’%bk’;
```

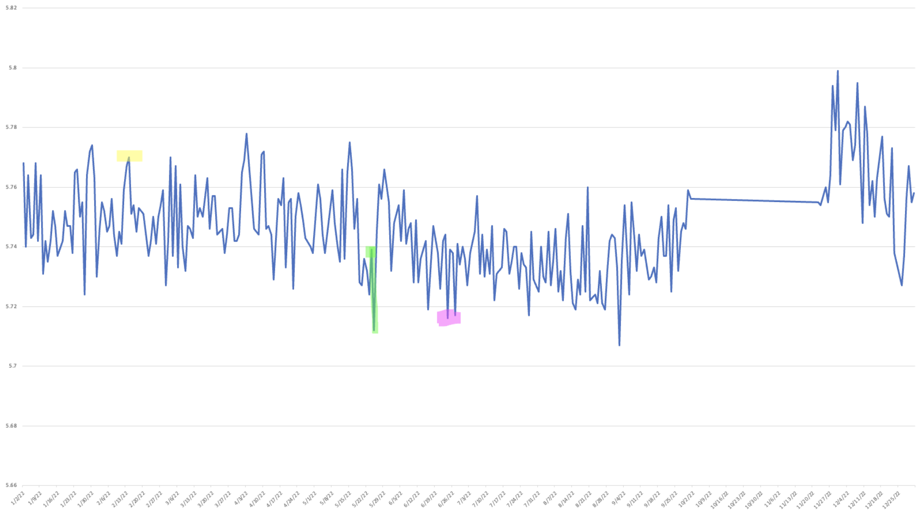
This query returned all the titles of books checked out by date in 2022, which I downloaded as a CSV.

3 Analysis

I then wrote a simple Python script which, for each date in the CSV, went through each title, looked up its individual words' happiness rankings, averaged per title and then averaged per day. In this way I was able to get an average happiness ranking per day, and save this into a new CSV. Like the Hedonometer, I ignored some words which are difficult to ascribe a happiness value. I also noticed there was outlier data on days that the library was closed, i.e. holidays. I removed those days from my analysis.

The script is printed in full at the end of the document.

From the resulting CSV, I generated this line graph of 2022:



Weirdly, there seems to be no checkout data from 10/2/22 to 11/21/22.

Like the Hedonometer data above, the happiness rankings tend to hover around an average happiness ranking, although book titles are slightly less happy overall. However, we can still see some trends. On both charts, the final months of the year after Thanksgiving are happier overall.

I've highlighted some other similarities in responses to significant days that I noticed.

Valentine's Day is highlighted in yellow, and it does seem like happiness spiked somewhat from previous days.

The mass shooting at Robb Elementary School was on May 24 and is highlighted in green, the chart hits one of it's lowest points the following day.

On June 24, the Supreme Court overturned Roe v. Wade, which is highlighted in pink. This and the next couple of days mark a downturn in happiness.

Overall, although the range of values was small, I was surprised that there was some real variation in sentiment, some of which may seem to be in response to large events. There also seem to be longer periods of mood shift, such as a sadder spell in the summer and a spike in mood in the winter. I wonder if this may actually reflect the opposite in general mood–perhaps people are actually sadder in the winter, and checking out cheerful books to try to improve their mood.

Out of curiosity, I also analyzed subsections of checkouts such as fiction or nonfiction, but the trends remained remarkably similar. I would be interested to try this again with subjects instead of titles, but I do feel broad subjects would map less cleanly to sentiment values of happiness.

4 Analysis Script

```
HEDONODATA = "Hedonometer.csv"
COUTDATA = "2022bookcouts.csv"

happyDict = {}

with open(HEDONODATA, encoding=’utf-8’) as csvf:
    csvReader = csv.DictReader(csvf)

    for row in csvReader:
        key = row[’Word in English’]
        val = row[’Happiness Score’]
        happyDict[key] = val

coutDict = {}

with open(COUTDATA, encoding=’utf-8’) as csvf:
    csvReader = csv.DictReader(csvf)

    for row in csvReader:
        datetime = row[’cout’]
        date = datetime[0:10]
        title = row[’title’]
        if date in coutDict and coutDict[date] is not None:
            coutDict[date].append(title)
        else:
            coutDict[date] = [title]

dates = list(coutDict.keys())
dates.sort()

dateRatings = []

irrelevant_words = ["the", "a", "and", "then", "miami","pearl", "santa",
                  "atlantic", "grand", "green", "falls", "haven", "sin",
                      "con",
                  "gren","springfield","falling","international","terminal","mad",
                  "al","ak","az","ar","ca","co","ct","de","fl","ga","hi","id","il",
                  "in","ia","ks","ky","la","me","md","ma","mi","mn","ms","mo","mt",
                  "ne","nv","nh","nj","nm","ny","nc","nd","oh","ok","or","pa","ri",
                  "sc","sd","tn","tx","ut","vt","va","wa","wv","wi","wy",
                      "alabama",
                  "alaska", "arizona", "arkansas", "california",
                      "colorado", "connecticut",
                  "delaware", "florida", "georgia", "hawaii", "idaho",
                      "illinois", "indiana",
                  "iowa", "kansas", "kentucky", "louisiana", "maine",
                      "maryland",
                  "massachusetts", "michigan", "minnesota",
                      "mississippi", "missouri",
                  "montana", "nebraska", "nevada", "new", "hampshire",
                      "jersey", "mexico",
                  "york", "north", "carolina", "dakota", "ohio",
                      "oklahoma", "oregon", "pennsylvania",
                  "rhode", "island", "south", "carolina", "dakota",
                      "tennessee", "texas",
                  "utah", "vermont", "virginia", "washington", "west",
                      "virginia", "wisconsin",]

holidays = ["2022-12-25", "2022-09-05", "2022-07-04", "2022-06-19",
           "2022-01-01", "2022-01-17", "2022-02-21", "2022-05-30",
           "2022-10-10", "2022-11-11", "2022-11-24", "2022-12-24"]

for date in dates:
    if date not in holidays:
       titles = coutDict[date]
       titleCount = 0
       dateHappy = 0
       for title in titles:
           titleScore = 0
           noHit = True
           words = title.split(" ")
           wordCount = 0
           for word in words:
               word = word.lower()
               if word not in irrelevant_words:
                  if word in happyDict:
                      noHit = False
                      titleScore += float(happyDict[word])
                      wordCount +=1
           if not noHit:
               titleCount += 1
               avgHappy = titleScore / wordCount
               dateHappy += avgHappy
       if not noHit:
           dateHappy = dateHappy/titleCount
           dateRatings.append({"Date": date, "Hedonometer":
               round(dateHappy, 3)})


field_names = ["Date", "Hedonometer"]
```

```
with open('HappinessPerDay.csv', 'w') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames=field_names)
    writer.writeheader()
    writer.writerows(dateRatings)
```