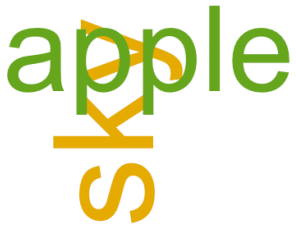
Lucas Sipos

Problem 1:

a. For some weird reason I only got the wordcloud like this 😳.

b. The words included are: institutions, yet, glorious, thousand, continue, compeers, upon, children's, benefits, united, children, rejoice, bequeathed, may, cause, conferred, country, enjoy, generations, Washington.
The words that are excluded are: our, and, to, a, the, on, us, by, have, under, those, his.
I think the words included are the one that produce the content of a text such as: nouns, verbs, adjectives and adverbs, and the ones excluded are the words that make a sentence function such as: articles, conjunctions, prepositions and pronouns.

c. Command use: wordcloud("apple apple apple apple apple sky sky sky sky sky sky a a a a a a car car", colors=brewer.pal(6,"Dark2"), random.order=FALSE)



I would say my guess was correct, but for the test that I'll be doing for point "c" and "d", I'll use another strategy to see even deeper into this wordcloud package.
The thing that I want to point out as you'll see in the example that I gave is that there is an average frequency rate that a work should have to appear in the wordcloud. Also, if I would change the 'a' with an "aaa", the wordcloud would print the "aaa", so I think that even unknown words that have high frequency would appear in the wordcloud.

d. Command used: wordcloud("sky car car sky sky sky sky sky sky sky sky sky hello hello hello hello hello hello hello bye bye bye bye bye bye why why why why why apple apple apple apple hi hi hi hi a a a a a a a a a a mine mine mine mine mine work", colors=brewer.pal(6,"Dark2"),random.order=FALSE)

As you can't see even though 'why' appears multiple times, it doesn't show up in the wordcloud because it's an excluded word, as well as 'a' and 'hi'.

After trying to make a python program that would give me an accurate answer about how the wordcloud work (code below):

```
from collections import Counter
✓ [60] < 10 ms
```

```
text = "sky car car sky sky sky sky sky sky sky sky hello hello hello hello hello hello hello bye bye bye bye bye bye why why why why why apple
 apple apple apple hi hi hi hi a a a a a a a a a a mine mine mine mine mine work".split(" ")
word_counter = Counter(text)
number_of_words = len(text)
print(f"Number of words in text: {number_of_words}")
print(word_counter)
✓ [61] < 10 ms
```

```
Number of words in text: 54
Counter({'sky': 10, 'a': 10, 'hello': 7, 'bye': 6, 'why': 5, 'mine': 5, 'apple': 4, 'hi': 4, 'car': 2, 'work': 1})
```

Now, I'll get the frequency of the words over the total number of words in text, without the words that are ***not included***

```
excluded_words = ["a", "why", "hi"]
frequency = []
for word, freq in word_counter.items():
    if word in excluded_words:
        number_of_words -= freq
        continue
    frequency.append((word, freq))
frequency = [(word, f/number_of_words) for word, f in frequency]
print(f"Number of words in text without excluded: {number_of_words}")
print(f"Frequency of words in text: {sorted([(w, f) for w, f in word_counter.items() if w not in excluded_words], key=lambda x: -x[1])}")
print(f"Frequency of words in text: {sorted(frequency, key=lambda x: -x[1])}")
print(f"Minimum frequency to be accepted in wordcloud: {sum([f for w, f in frequency])/len(frequency)}")     frequency
✓ [62] < 10 ms
```

```
Number of words in text without excluded: 35
Frequency of words in text: [('sky', 10), ('hello', 7), ('bye', 6), ('mine', 5), ('apple', 4), ('car', 2), ('work', 1)]
Frequency of words in text: [('sky', 0.2857142857142857), ('hello', 0.2), ('bye', 0.1714285714285143), ('mine', 0.1428571428571428), ('apple',
 0.11428571428571428), ('car', 0.05714285714285714), ('work', 0.02857142857142857)]
Minimum frequency to be accepted in wordcloud: 0.14285714285714285
```

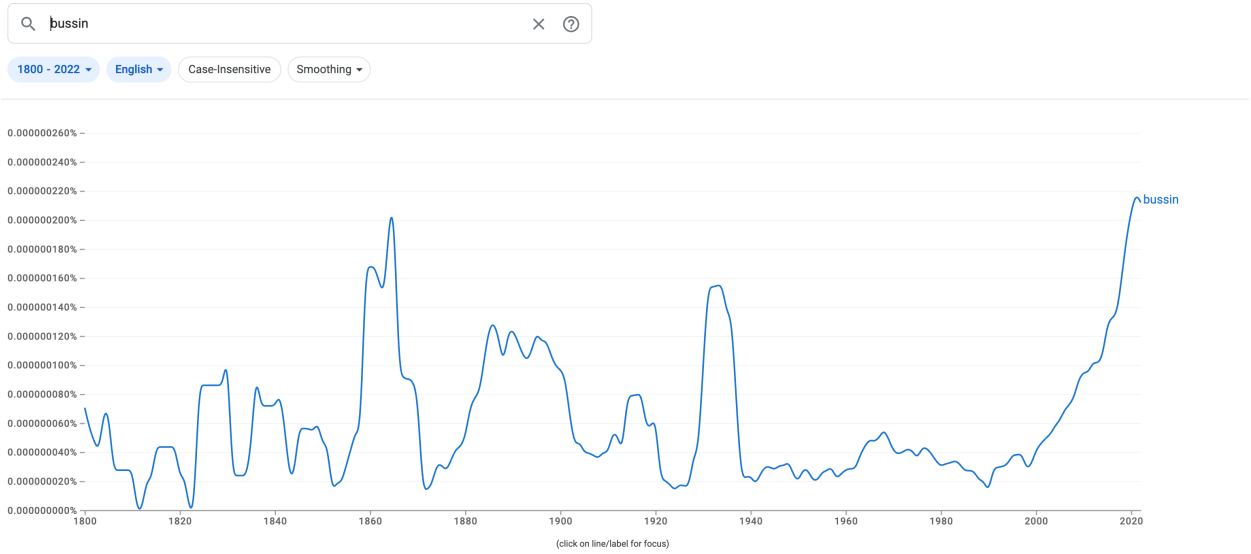
I realised that maybe wordcloud only gets the words that have the frequency = max_frequency * 0.4, so basically 40% out the maximum frequency from the included words, of course 😅. I've tested out the same input but I've added an included word with frequency = 4 and it worked.

Problem 2:

a. Over time the books that refer to Mark Keane had some really small peaks in the 1817 – 1822 and 1833 – 1838, after that the named wasn't heard of in the book industry until 1908 – 1914 and even then too little to be noticed. We can see the real spikes coming in 1955 and then having a little downfall, but not for long, because, in 1971 it touch its peak popularity. The next ~20 years it was in a continuous fall, but came back a little in 2001, again went down until 2013 and up until 2022 it's still growing.

b. If I type my whole name, then it wouldn't give me anything, for Lucas, there is actually a very good graph ragarding books and articles having the book title or article information regarding my name. The lowest points were in 1808 and 1825 but from then on there was a very good scaling until 1902 and it went down until 1973 but then it starting growing so much that in 2017, Lucas hit his peak.

c. I don't think the word is a recent introduction into English, but every gen Z kid uses this word. Nevertheless, I never knew that this word was actually used in various articles during the 1860's and the 1930's.

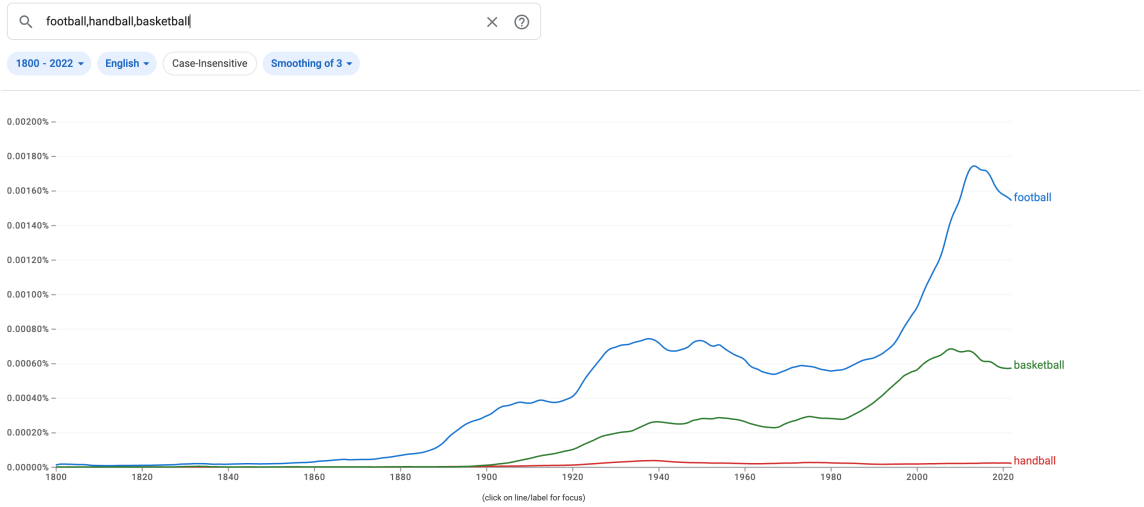


d. For the lower values:

- Closer to the original data, so you can see the real differences between years
- The line will be less smooth and have more peaks
- Harder to see the long term trends

For the higer values:

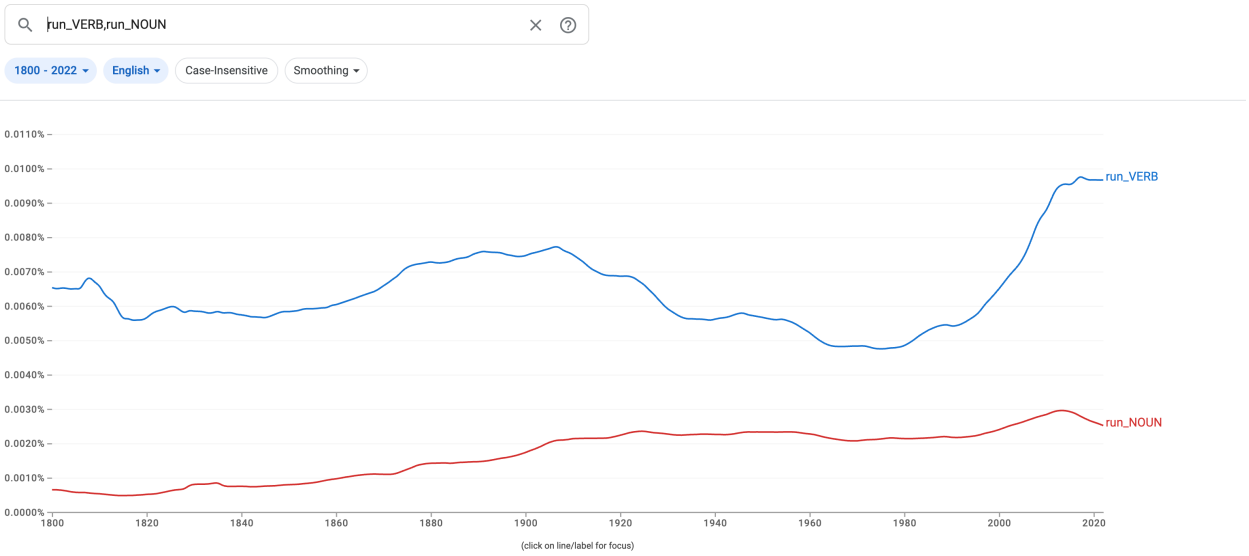
- You cannot see the spike of those trends
- The line will be smoother, getting rid of the peaks and becoming hills
- Easier to identify ling term trends



e.

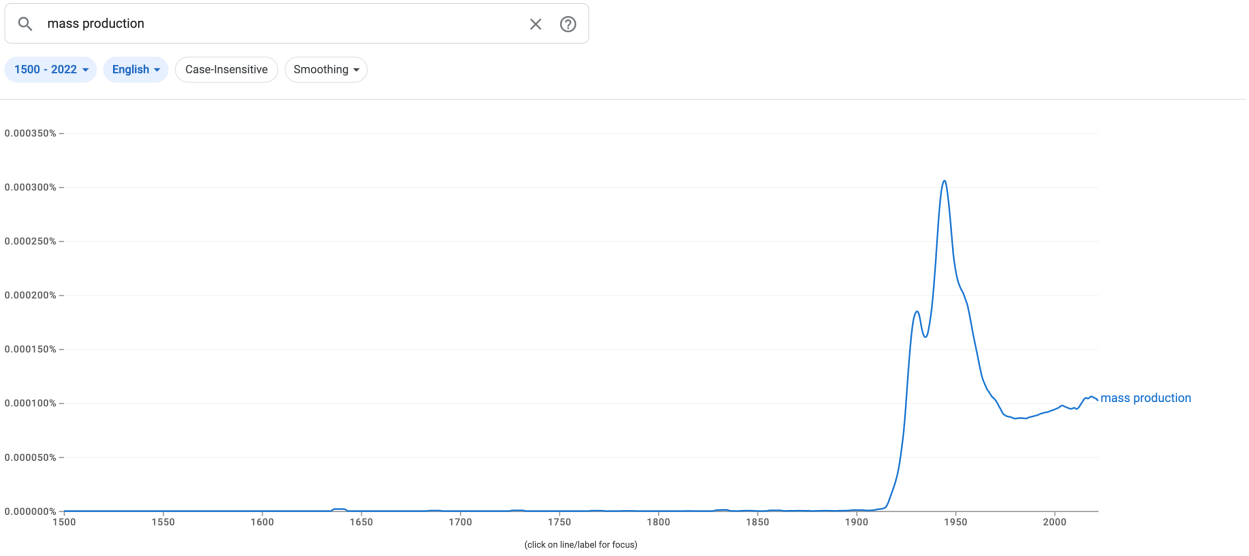
Lucas Sipos

I chose the 3 major sports that all have in common a ball about the same size and pretty similar to each other. There is nothing interesting about it, only the part where football wasn't as popular as is today, even though I thought that football was popular sincer the 60'. Everyone knows that this contest is fairly display because the majority of the population watches football and there is a good percent which watch basketball. The interesting thing is that handball has such a low score in perspective with the other ones. Nevertheless, there will probably be an increase in both football and basketball in 2023 and onwards, but I don't think that handball will ever recover.



f.

As expected the word run is more popular as a verb rather than a noun. Fun thing that I founf about it is that the verb run (if we ignore the 80's+) had its peak right before the first World War, as if they knew something is about to go down and the same goes for the second one. After that it became more popular maybe because people are trying to get fitter these days. As for the noun I think it's because of the "let's go for a run" sentence.



g.

Lucas Sipos

As expected, mass production wasn't something to talk about 500 years ago, but it started to get popular at the beginning of the first World War, maybe because they needed chemical weapons and normal ones to win the was. But the real spike happens around the discovery of nuclear weapons and I think everybody was in a race for the production of the best nuclear weapon. After that the productions of them slowed down, but the term started to grow again because now there is a need of mass production aproximately in every indutstry.

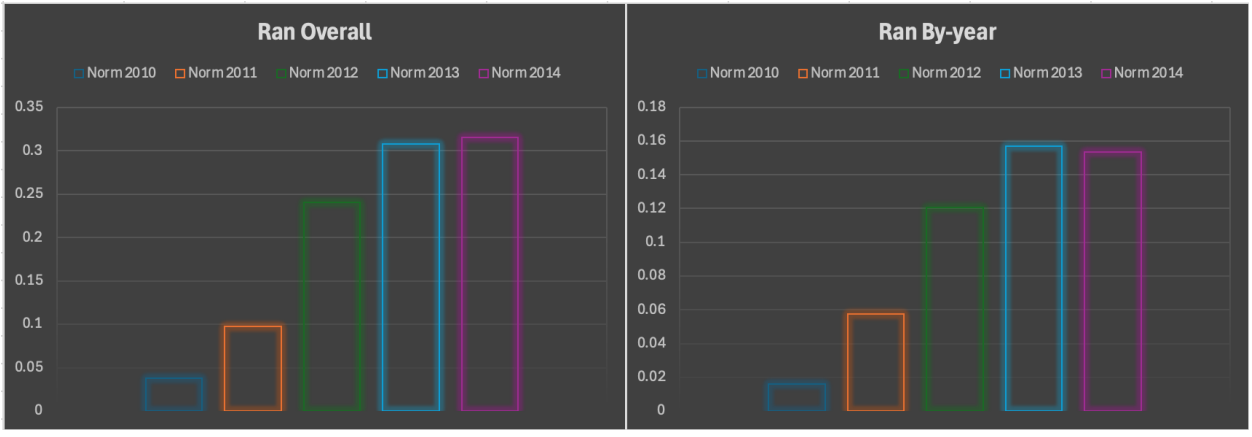
Problem 3:

a/b.

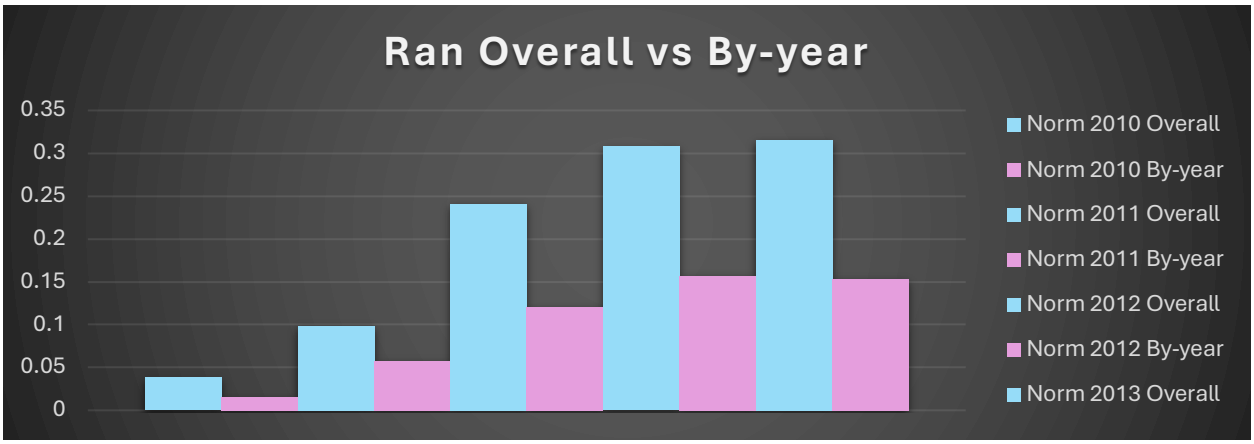
Words	2010	2011	2012	2013	2014	Total
ran	208	533	1313	1679	1719	5452
out of	1786	863	399	282	265	3595
ideas	1362	557	714	1209	1417	5259
when	1391	1179	1284	1476	447	5777
started	1811	451	1443	687	1718	6110
to write	53	945	1993	1444	1612	6047
table	1871	238	1449	480	394	4432
columns	789	937	957	586	1831	5100
rows	1914	1775	96	1228	1435	6448
frequency	1701	1806	1226	1615	355	6703
Total	12886	9284	10874	10686	11193	54923

	Norm 2010		Norm 2011		Norm 2012		Norm 2013		Norm 2014	
	Overall	By-year	Overall	By-year	Overall	By-year	Overall	By-year	Overall	By-year
ran	0.03815114	0.01614155	0.09776229	0.0574106	0.24082905	0.12074674	0.30796038	0.15712147	0.31529714	0.15357813
out of	0.49680111	0.13860003	0.24005563	0.09295562	0.11098748	0.03669303	0.07844228	0.02638967	0.07371349	0.02367551
ideas	0.2589846	0.1056961	0.10591367	0.05999569	0.13576726	0.06566121	0.22989161	0.11313869	0.26944286	0.12659698
when	0.24078241	0.10794661	0.20408517	0.12699268	0.22226069	0.11807982	0.25549593	0.13812465	0.0773758	0.03993567
started	0.29639935	0.14054012	0.07381342	0.0485782	0.23617021	0.13270186	0.11243863	0.06428972	0.2811784	0.15348879
to write	0.00876468	0.00411299	0.15627584	0.10178802	0.32958492	0.18328122	0.2387961	0.13513008	0.26657847	0.14401858
table	0.42215704	0.14519634	0.05370036	0.0256355	0.32694043	0.13325363	0.10830325	0.04491859	0.08889892	0.03520057
columns	0.15470588	0.06122924	0.18372549	0.10092632	0.18764706	0.08800809	0.11490196	0.05483811	0.35901961	0.16358438
rows	0.29683623	0.14853329	0.27527916	0.19118914	0.01488834	0.0088284	0.19044665	0.11491671	0.22254963	0.12820513
frequency	0.25376697	0.13200372	0.2694316	0.19452822	0.18290318	0.112746	0.24093689	0.15113232	0.05296136	0.03171625

c.



Lucas Sipos



The overall normalization gives a wider perspective on when the word was used the most throughout his whole life span. Unlike the by-year normalization which shows you the popularity of that word compared to everything else said that year.