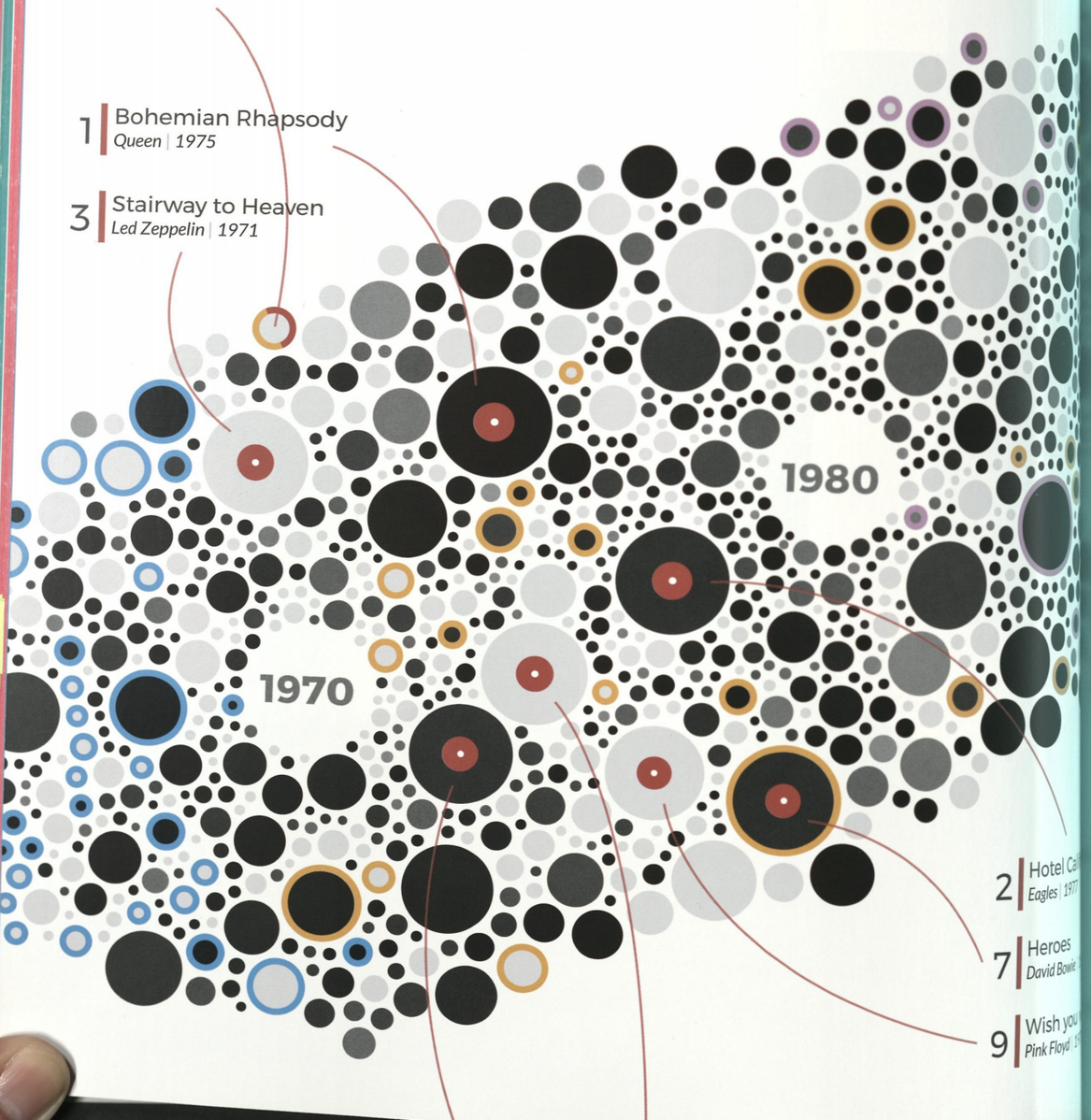
Newly discovered

Although already released in 1972, *Starman* from David Bowie is the highest new song in the list. It never appeared in the previous 17 editions of the Top 2000 and entered in 2016 on position 270.

Prince

Another legend who passed away in 2016 (on April 21st). It seems that new people discovered his works, with all 9 songs that were in 2015's list rising significantly and 8 more songs joining in 2016.

6 Avond
Boudewijn



The Top 2000 ❤ the 70s & 80

NADIEH

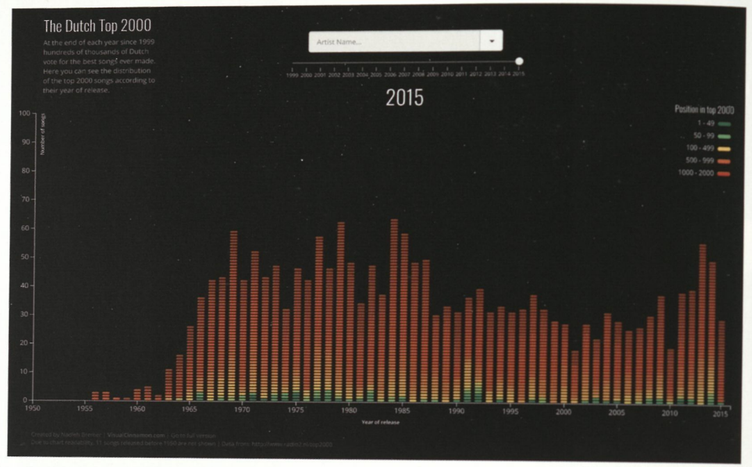
When you say "Music in December" to somebody from the Netherlands, there's a very likely chance that they'll think of the Top 2000. The Top 2000 is an annual list of 2,000 songs, chosen by listeners, that airs on Dutch Radio NPO2 and is played between Christmas and New Year's Eve. I actually played with this data before in 2014, when I was still very new to D3.js (Figure 6.1).

As artists sometimes revisit a past artwork to see how their style has evolved (which I always love to see), I thought it would be fitting to try and do the same myself. So, two years after my first attempt, I decided to look at the Top 2000 songs again and visualize which decade was the most popular in terms of the years the songs were released.

Not to worry if you're not Dutch; roughly 90% of the songs in the Top 2000 are sung in English (with Queen usually in first place) so the songs should seem familiar.

Fig.6.1

The result of visualizing the Top 2000 data in a 2014 personal project made with D3.js.



Data

Thankfully, the Top 2000 website publishes an Excel file of the 2,000 songs for that year, containing the song title, artist name, and year of release. On December 19, 2016, that file was released. However, I wanted another important variable—the highest rank ever reached in the normal weekly charts—because adding this to the visual would provide more context on the songs. There are a few of these in the Netherlands, and I eventually went with the Top 40 chart because this had been going non-stop since 1965 and because the Top 40 website seemed scrapeable. Next, I wrote a small scraper in R that would go through the ±50 years of music chart lists and save the artist's name, song title, song URL (as the unique ID), and chart position. This data was then aggregated to make it unique per song. I also saved some extra information per song, such as the highest position ever reached and number of weeks in the Top 40.

The tricky part was to match the artists and songs from the Top 40 list I created to those in the Top 2000 list. I first tried a merge on an exact match of artist and title. That matched about 60% of the songs between both lists. Not bad, but I had actually expected more matches, thinking that artist and song names were rather fixed.

Browsing through the remaining songs, I noticed that one of the two lists would sometimes use more words than the other, such as "John Lennon-Plastic Ono Band" versus just "John Lennon." Therefore, I also searched for partial matches between the lists, as long as all the words of one song and artist were contained in the other list. That helped match 10% more.

Next came the fuzzy part. Sometimes words were apparently written slightly different, such as "Don't Stop '*Til* You Get Enough" versus "Don't Stop '*Till* You Get Enough." Using R's `stringdist` package, I applied the Full Damerau-Levenshtein distance to compare titles and artists. However, I was quite strict; only two changes were allowed on both the title and artist to create a match (otherwise, the song "Bad" from U2 could be turned into any other three-letter song title or 2–3 letter artist name). Sadly, that only gave me 2.5% more matches, and I manually checked all the matched songs after each step to correct a handful of wrong matches.

The Top 40 chart URLs had a year/week logic to them: www.top40.nl/top40/2015/week-42

The Full Damerau-Levenshtein distance counts the number of deletions, insertions, substitutions, and transposition of adjacent characters necessary to turn string "a" into string "b."

I also tried something with the "Tips of the Week" list to check against, searching for songs that were tipped but that never made the Top 40, which gave me a few more matches. For the remaining songs I manually went through each list searching for artists or song titles with variations in how they were spelled, such as "Andrea Bocelli & Sarah Brightman" in the Top 2000 list versus "Sarah Brightman & Andrea Bocelli" in the Top 40. For the remaining 380 songs, I wasn't able to find exactly how many actually appeared in the Top 40, but after all the data processing I did along the way, I'd guess it's less than 10%.

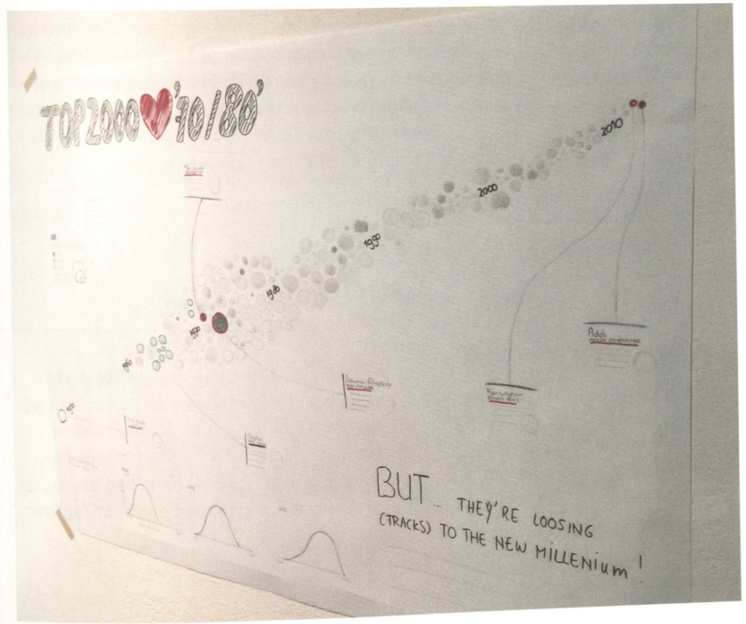
Data

The idea for visualizing this particular dataset had been in the works for some time. During the spring, I attended a very interesting data visualization workshop given by Juan Velasco on "Information Graphics for Print and Online." Part of the workshop was to come up with an idea for an infographic. And although my small team of three people came up with 40 possible ideas, we were all intrigued by the Top 2000 songs.

We decided to have the most recent list of 2,000 songs take center stage and visualized them in a "beeswarm" manner that grouped them around their year of release. Each circle (i.e., a song) would be sized according to their highest position in the Top 40 and colored according to their rank in the Top 2000. Some of these songs would then be highlighted with color and annotations, such as "highest newcomer in the Top 2000 list."

Fig.6.2

The general idea for visualizing the Top 2000 song using a "beeswarm" clustering to place songs at their year of release.



↳ Add Context Using Remaining Visual Channels

Even if getting the main insights from your data across to your audience is of the utmost importance, try to keep an open mind by adding extra details to create additional context about the information that you want to convey. This can create a more visually pleasing result, while also giving the truly interested reader even more ways to dive into and understand the information.

A way for me to think about adding extra details is to think about which "visual channels" are still free after I have the main chart standing; visual channels being those components of a data visualization that can be used to encode data, such as position, color, and size. For example, with the Top 2000 infographic during the workshop, our team knew that we wanted to use a beeswarm clustering to place all the song circles near their year of release. This would define the main visual's shape and also answer the original "Which decade is most popular in terms of song release year?" question. And while size and color are pretty common visual encodings to use with data, there are so many more visual channels that make it more interesting!

In terms of *remaining* visual channels, we chose to use a colored stroke to highlight the "interesting" songs (such as the highest riser, newcomer, or the Pokémon song), which we also annotated with text.

Finally, in the bottom section we decided to place some mini-charts that highlighted the distribution of the songs (arranged by release year) that were featured in the 1999, 2008, and 2016 editions of the Top 2000. These would highlight the fact that the bulk of the 2,000 songs from the 1999 edition of the Top 2000 were released in the 70s, but that this has slowly been moving towards newer decades for every new edition of the Top 2000.

On the second day of the workshop we also made a mobile version of the concept. This time we thought of creating a long scrollable beeswarm visual where you could theoretically listen to bits of each song and see extra information (Figure 6.3).

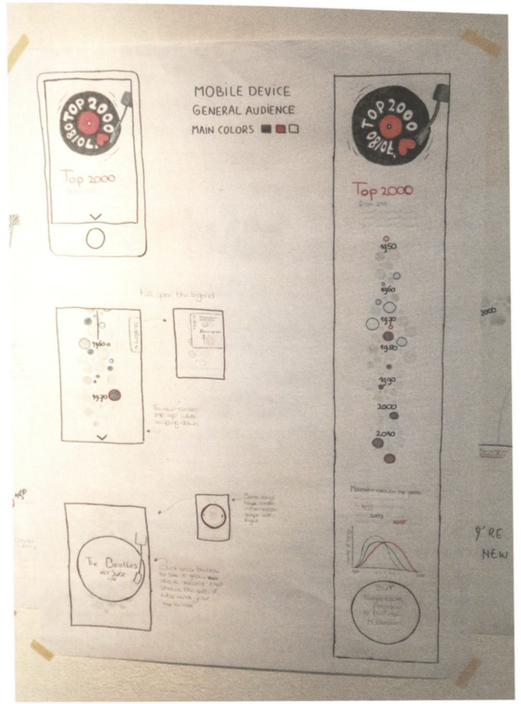
Code

This time I finally made something primarily static: a poster. Nevertheless, I still had to use `D3.js` to build the beeswarm centerpiece. In this case I needed a *force* along the horizontal x-axis that would cluster the songs based on their year of release, starting from the 1950s at the left all the way to 2016 at the other end. It took me several iterations to figure out the right balance of settings before it filled the region nicely around the horizontal axis, without the songs being moved away too far from their actual release year.

This is very similar to what I talked about in my "Royal Constellations" project when I used the birth date to pull the network apart along one axis.

Fig.6.3

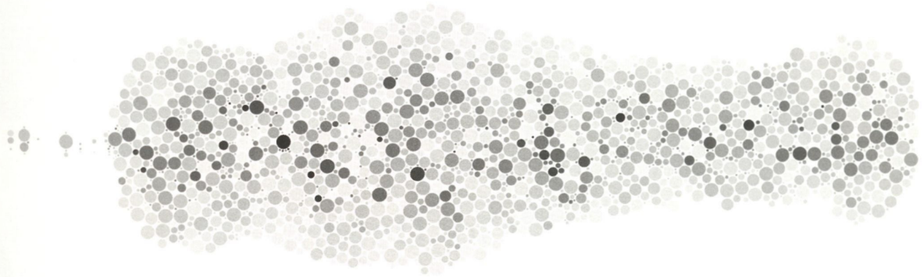
For the mobile version we converted the poster to a very thin and long beeswarm where you could listen to small snippets of each song.



In my first attempts, the circles were *sized* according to the highest position they had reached in the Top 40 and *colored* according to their position in the Top 2000. This is what we'd come up with during the workshop. But this gave some difficulty in songs that never appeared in the Top 40; I still needed a size for these. I therefore made the "unmatched" all the same size, but that resulted in many light grey circles of about the same size and it didn't look appealing.

Fig.6.4

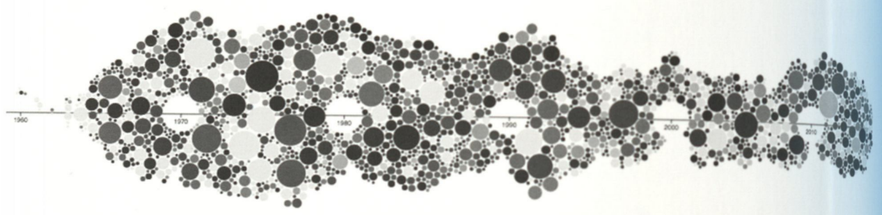
Using Top 40 information for circle size and Top 2000 ranking for color didn't create an appealing image.



In hindsight, this choice of size and color wasn't optimal in another way as well. My visual was about the Top 2000, but I was using circle size for the "nice-to-have" variable of the Top 40 position. It made a lot more sense to switch the two; the Top 2000 rank determined the circle size and the Top 40 rank became a color. Now, the biggest circles—which were easier to locate—actually represented the highest ranking songs. And I could use a very light grey (a color more often used for "missing" data) for songs that never appeared in the Top 40. Plus, it immediately gave a great visual improvement and only needed a few tweaks to the code.

Fig.6.5

Switching the scales for the Top 2000 and Top 40 rankings immediately made the visual more visually appealing and effective.



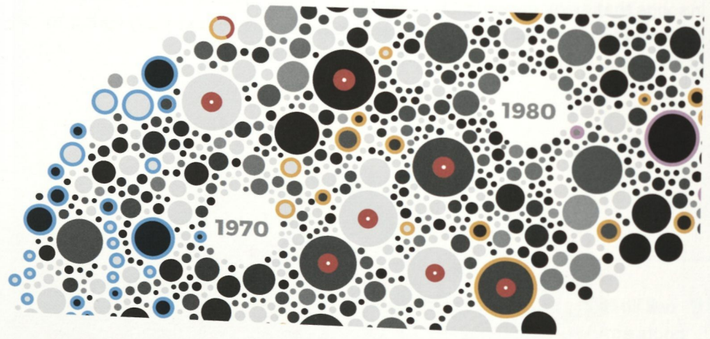
I then started to mark out the circles (songs) that I wanted to annotate later. Already during the workshop we decided to keep the visual very black and white, inspired by the intense blackness of vinyl records. Using red, the color of the Top 2000 logo, to mark songs that had something interesting about them and blue for the artist/band with most songs in the list. I also highlighted all the songs by David Bowie and Prince, who passed away in 2016, by adding yellow and purple strokes around their songs, respectively.

Since the top 10 songs from the list were the biggest circles, I thought it would look nice to mark these as small vinyl records to make them stand out even more.

The "vinyls" are nothing more than a very small white circle on top of a small red circle.

Fig.6.6

Using colored strokes to mark certain artists, and turning the top 10 songs into tiny vinyl records with a red and white circle on top.



↳ Design to Maximize for Delight

Those top 10 songs didn't have to look like tiny vinyl records to make them stand out. However, by adding a touch that fits with the topic that's visualized, it made the total visual just a little bit more fun to look at.

Outside Strokes with SVG

Although possible in certain vector drawing programs, such as Adobe Illustrator, you cannot do an *outside* stroke on SVG elements, such as circles or rectangles, in the browser. Thus when you stroke an element, the width of that stroke is centered on the outline of the element. However, for data visualizations (and especially for smaller circles) it's quite important that part of the circle's area isn't "hidden" behind a stroke.

Thankfully, an outside stroke can easily be mimicked; plot a circle in the color that you have in mind for the stroke. The radius of this circle should be just as big as your "actual" circle *plus* how wide you want the stroke to be. Next plot your actual circle on top and it will look like the background circle is an outside stroke (Figure 6.7).

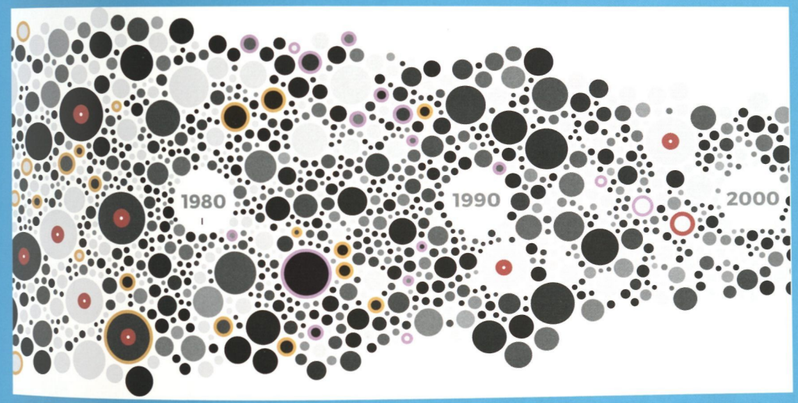


Fig.6.7

The colored strokes are colored circles behind the grey circles that are just a little bigger.

With those relatively simple elements done and being sure I wouldn't change anything anymore, I copied the SVG element of the beeswarm from my browser and pasted it into Adobe Illustrator. There I started adjusting it to look like the poster that our little group had made during the workshop (from Figure 6.2). Such as turning the beeswarm 25 degrees, just for the effect of making it look a bit more interesting, and placing annotations around it. For the red "notable" songs I used the data itself together with the Top 2000 website to search for some interesting facts, like Justin Timberlake having the highest ranking song from 2016. I placed these texts using an underlying grid to keep things nicely aligned in columns and rows (Figure 6.8). After finishing the beeswarm/top part of the infographic, I capitulated on keeping this visual totally static and made a small interactive version online just to be able to hover over each circle and see which song it is (Figure 6.9).

You can either use the free "NYT SVG Crowbar" tool, or literally copy the SVG element from the Chrome devTools into Illustrator.

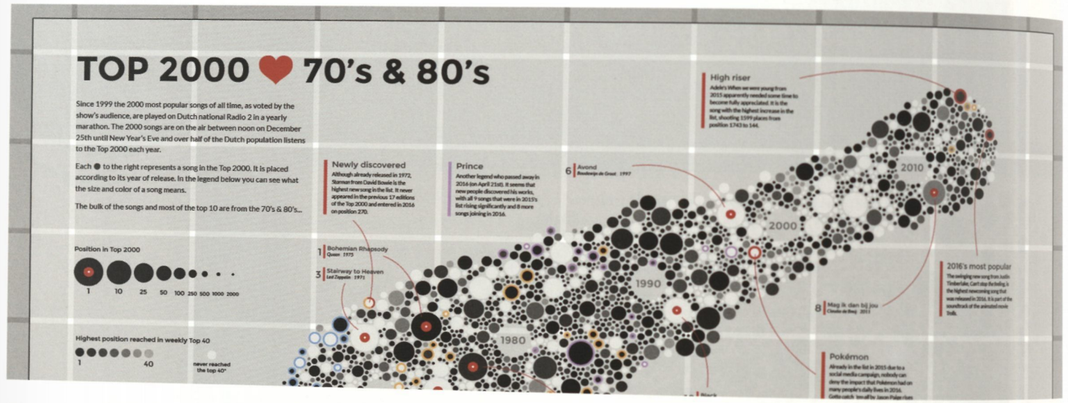


Fig.6.8

The underlying grid that I used in Illustrator to lay out all the text.

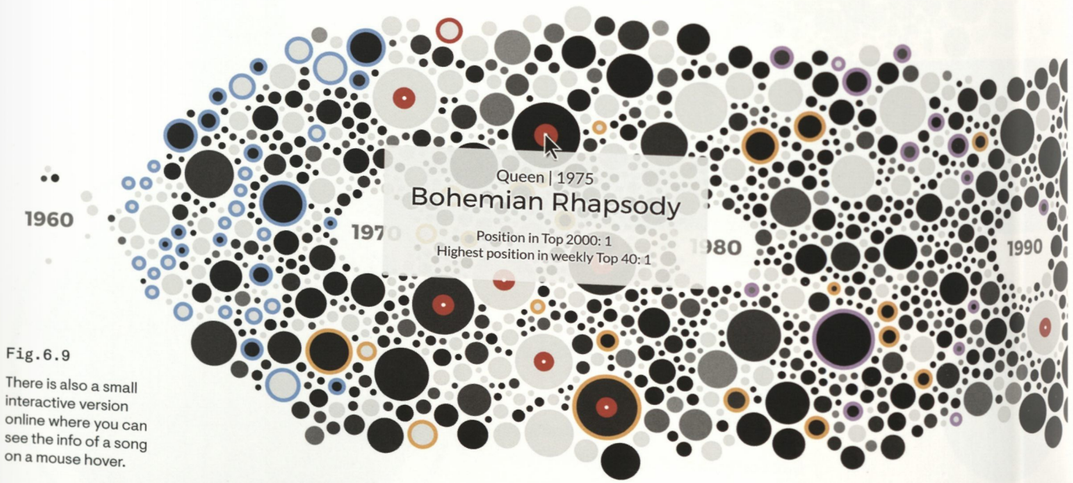


Fig.6.9

There is also a small interactive version online where you can see the info of a song on a mouse hover.

Finally, I wanted to incorporate the fact that the distribution of the songs across release year has been changing towards the 90s and 2000s. I had mentioned this fact to my team during the workshop, and we'd placed three simple line charts in the lower left of our design to highlight it (Figure 6.2). But now that I wanted to *actually* create the charts, it wasn't quite clear what visual form would convey the idea best. I already had the full history of every Top 2000 since the first one aired in 1999 from my previous visual on the topic from two years ago. I appended the 2016 data and started making some simple plots in R using `ggplot2`. That it should probably be a histogram or something similar was clear to me from the start, but should I smooth it down? How many years to show? Should they overlap or be displayed as "small multiples"? (Figure 6.10).

In the end I chose to go with a small multiple histogram of four editions picked from the past 18 years, but overplotted with a smoothed density curve to make the general shape more easily comparable between the four charts.

In Figure 6.11 you can see what I took straight from R. I played with the color to also encode the height. Eventually, however, I made them all the same grey on the poster, since I didn't want the histograms to draw too much attention.

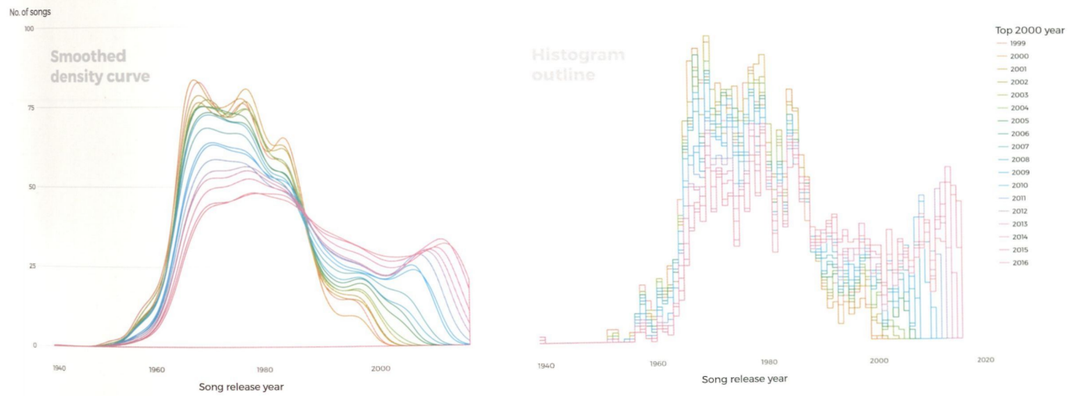


Fig.6.10

Comparing the trend of song release year across all 18 editions of the Top 2000.

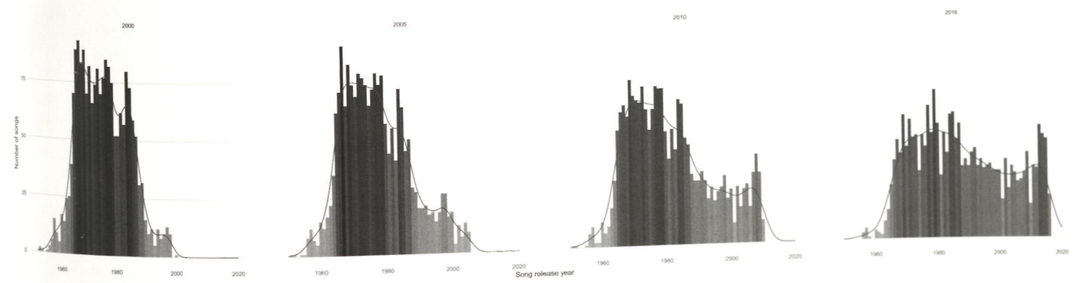


Fig.6.11

Four histograms from four different years of the Top 2000 showing the distribution of song release year and its steady move towards more recent decades.

↳ Combine Tools

It's perfectly fine if you create your visual with several different tools. Just like people, finding a unicorn tool that can do everything you need "the best way every time" doesn't exist. Personally, I always start with R; I use it to read in the data, do some initial statistics, clean and transform the data, and make (lots of) simple plots to get a sense of the data. However, although R's `ggplot2` package can do a lot to visualize data, I need more creative control for the final result, and it often has to be interactive and on the web. I find that creating my visuals in JavaScript with the help of `D3.js` provides what I need.

For static visuals, I always end with Adobe Illustrator, or other vector-based tools, to add some final touches (such as legends and annotations—things that can take a lot of time to add with code). Of course, not all projects lend themselves to using your tools of choice, but just keep in mind that you don't need to always create a visual from scratch with *one tool*. Try to learn more tools and programs and acquaint yourself with the strengths and weaknesses of each.

Reflections

One of the main reasons why I decided to create a poster instead of an interactive visual came down to other time commitments. Making a static visualization is always *much* faster for me to create, even if it's partly based on something that I started in `D3.js`. For static visuals you just don't have to consider browser bugs, performance, responsive design, *and* interactivity!

Preparing and working on the data scraping and cleaning took about 20 hours of time, the ideation and sketching about three, and the coding/creating approximately 20–30 hours. Finishing a static visual after creating so many interactive ones always reminds me how much I like making printable static visuals.

A funny thing happened though. I generally share my (personal) projects only on Twitter, maybe Instagram if I'm in a good mood. But I decided to also post this infographic on my LinkedIn account, due to most of my connections there being Dutch at the time. And somehow, it got picked up beyond my direct connections and went a little viral in the Netherlands. Totally unexpected! I thought this visual was too complex for that. Majorly exciting to experience though! A few hundred thousand views and thousands of likes later, I even had some businesses reach out to me to create a data visualization for them. Which was *perfect* timing, since I was officially a freelancer from the moment the Top 2000 finished airing the number 1 song at the very start of 2017.

The Top 2000 ❤ the 70s & 80

↳ Top2000.VisualCinnamon.com

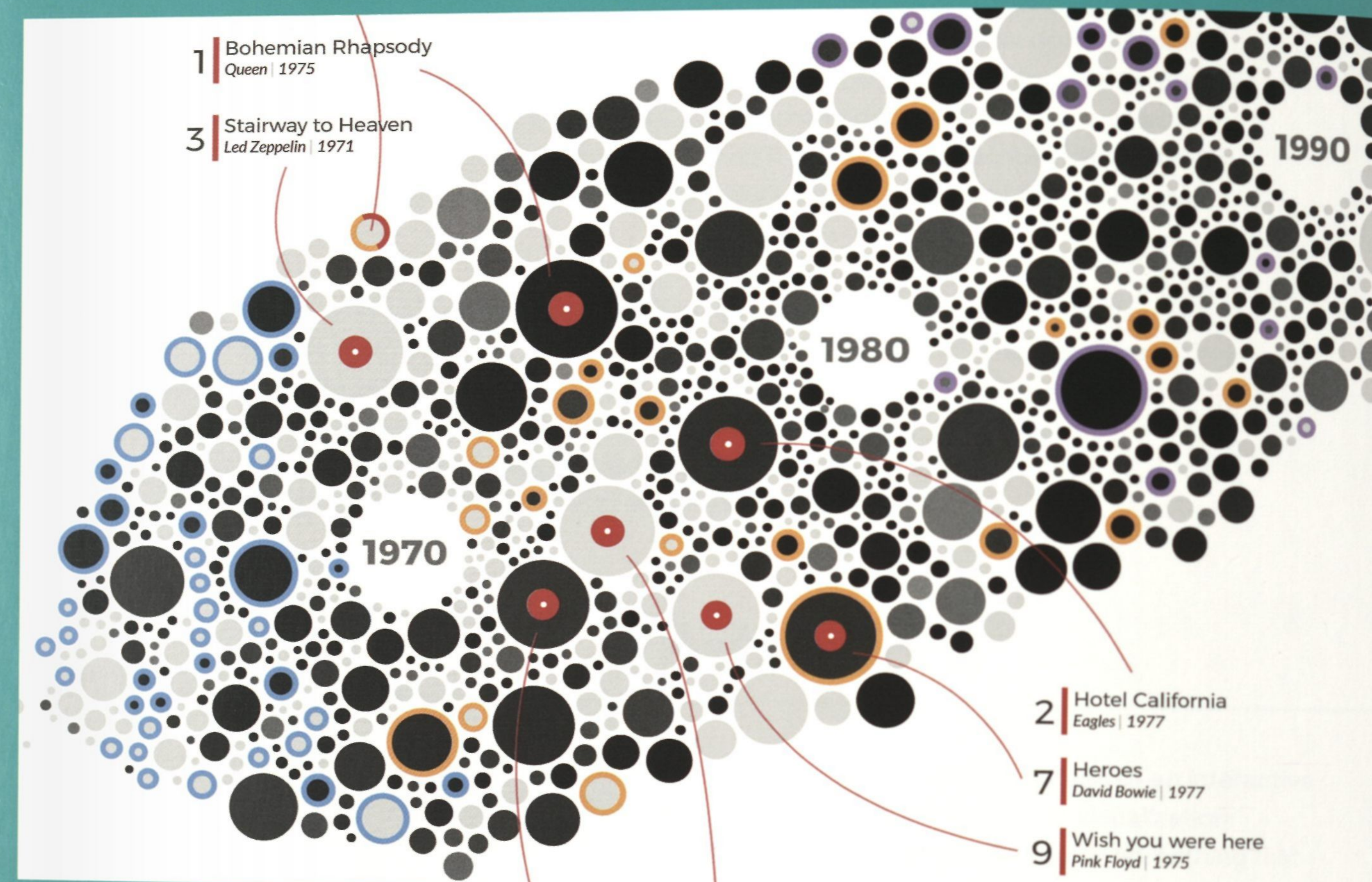


Fig.6.12

The most popular decades in terms of song release year are the 70s and 80s.

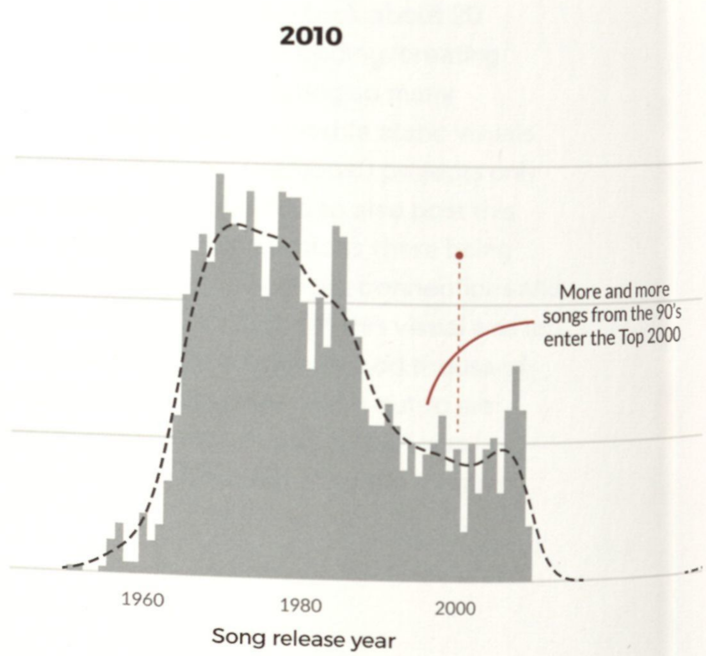
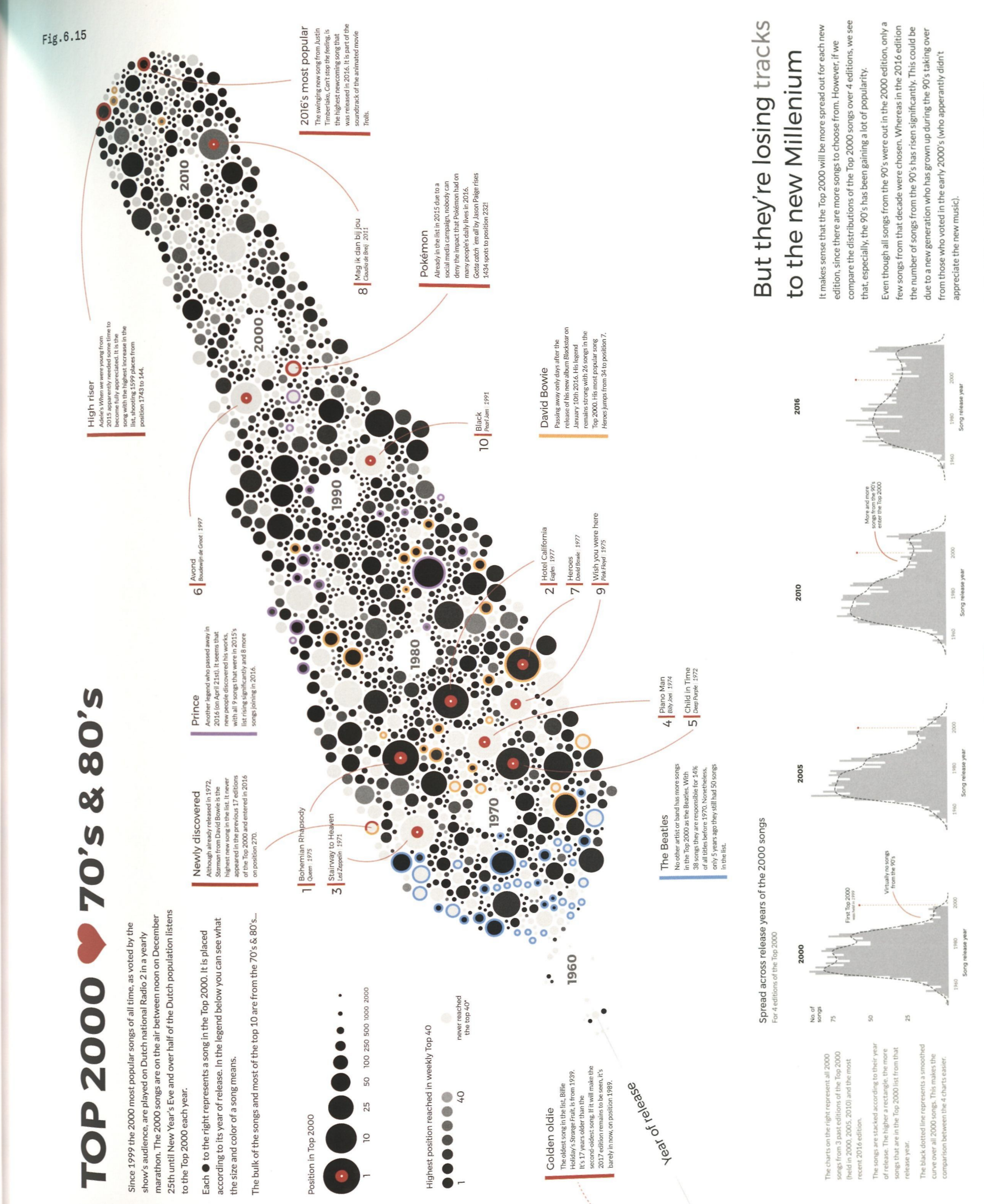


Fig.6.13

Highlighting all of Prince's songs in the list with a purple stroke.

Fig.6.14

With each new Top 2000, more songs from the 90s enter the list, even though the first edition was aired in 1999.



Data-Driven Revolutions

SHIRLEY

When Nadieh and I first agreed on the topic "Music," I felt quite lost. I didn't know what I wanted to do, except to perhaps explore something related to K-Pop. But K-Pop was just too broad of a topic and I didn't know it well enough (anymore) to find a good angle. I was lamenting this lack of an angle with a friend when I suddenly remembered the game *Dance Dance Revolution* (DDR).

DDR originally started as an arcade game in Japan and eventually was released as video games for home consoles. Basically, the player would stand on a mat or a platform that served as the "controller" and step on any of the four arrows on the mat to "press" the controller buttons. A combination of arrows would appear on the TV screen in front of them, and the players would have to step on (or stomp on) the same arrow(s) on the mat, timing their steps with the arrows scrolling up the screen. It was a popular game across the world and a huge part of my teenage life.

I first came across DDR in 2001 at a friend's house. Since there were no arcades with DDR anywhere near where I lived, I begged my parents to buy me the game. (This was easier said than done; there were absolutely *no* video games in our house at the time, so a DDR set meant not only buying the game itself, but also the PS2 and the mat that came along with it.) It took me two years to convince my parents, and the summer I got it, I was on it every day for hours. I was the type that played the same song on the same difficulty over and over until I mastered it, and I played it regularly until I left for college five years later.