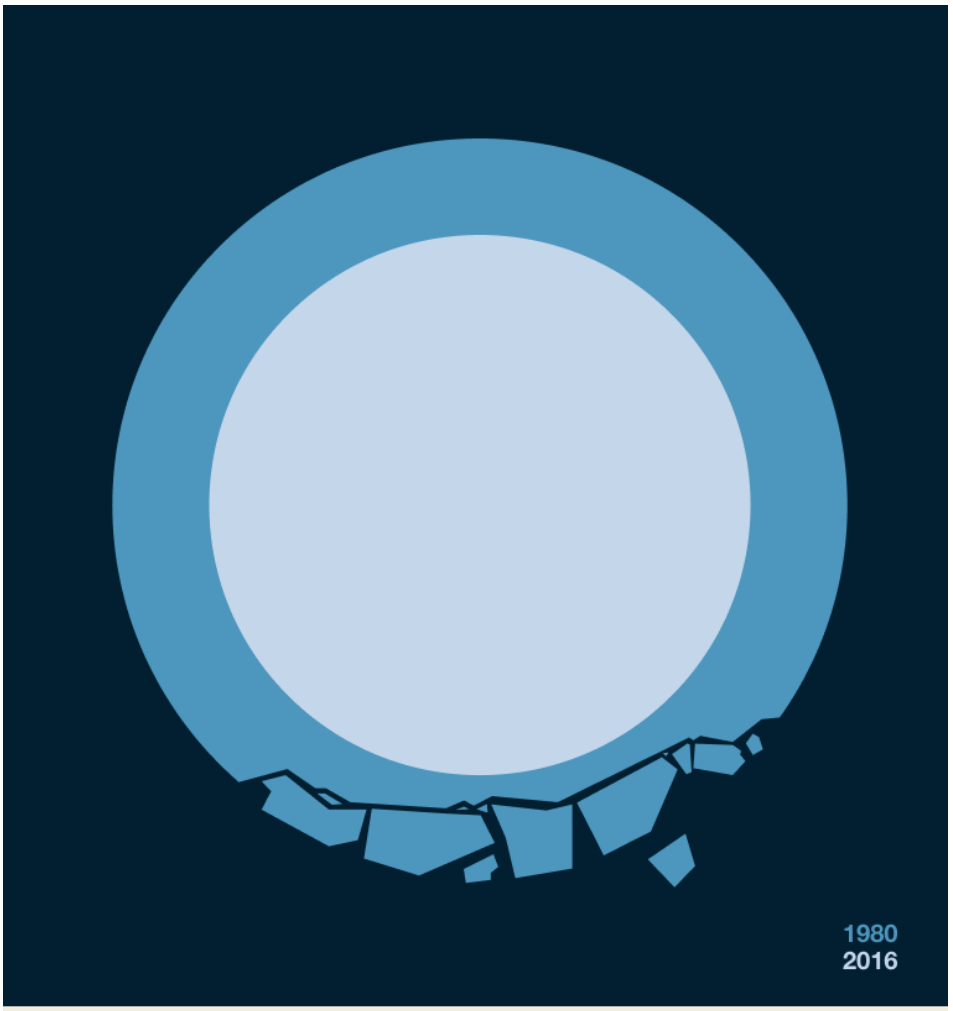
The Making Of: Cracked Sea Ice Loss



sea ice loss

1980 – 2016

This simple graphic depicting the reduction of Arctic sea ice between 1980 and 2016, highlights the urgency of the Arctic sea ice loss crisis. As our planet's temperature continues to rise due to climate change, the Arctic is losing sea ice.

Hello fellow data visualizer!

This little tutorial will guide you through all necessary steps to reproduce the little picture “Arctic Sea Ice Loss” that is shown above. Depending on how experienced you are in the field of data visualisations we have produced individual step-by-step tutorials for several different platforms for you to choose from. Be advised that there is no strict right or wrong when visualising the data as long as the data itself and the message derived from it is not changed. Oh and please have fun on the way!

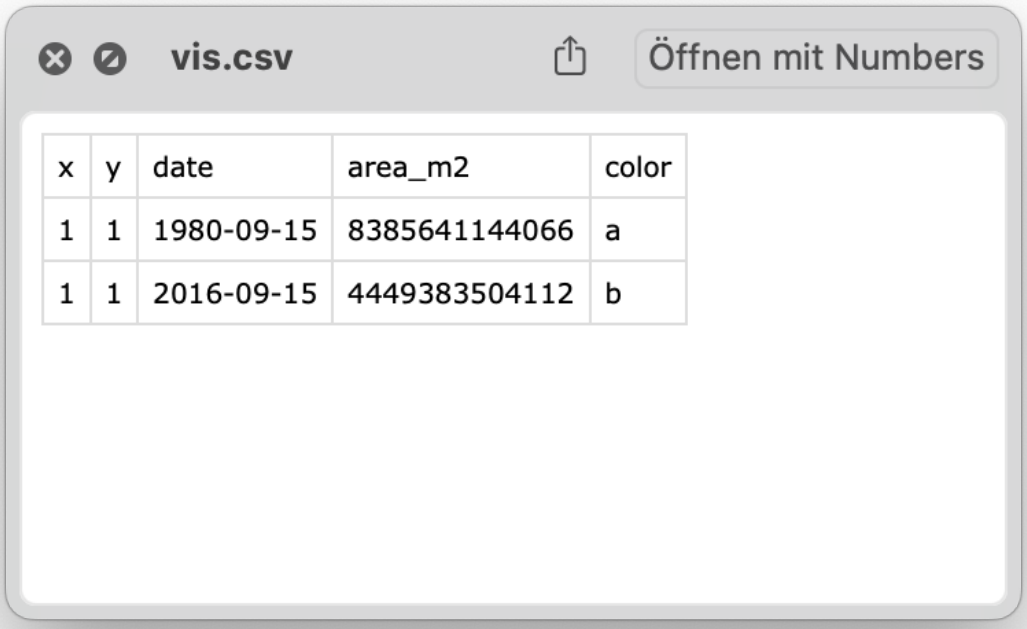
You will Need

A computer with internet access and depending on the tutorial you choose the installation of a vector editing software like open source Inkscape or commercially available Adobe Illustrator. If you go down the “programming charts” path you’ll need to install a python and jupyter notebook environment or use a hosted environment like Google Colab or noteable.io .

Where to get the data

1. Go to the site climate.esa.int/littlepictures. Find the "Greatest Hits" section. This is where you will find all the data you need.

2. Download SeaIceLossTheIceBreakerLittlePicture.csv. Then open it up in a text editor of your choice. You will find 5 columns in the file and two rows. The x and y columns are placeholders for the visualisation. The column “area m2” is the extent or arctic sea ice in m2. The first row contains the data from September 1980 and the second row is the extent from 2016. The last column is used for layering the data correctly.

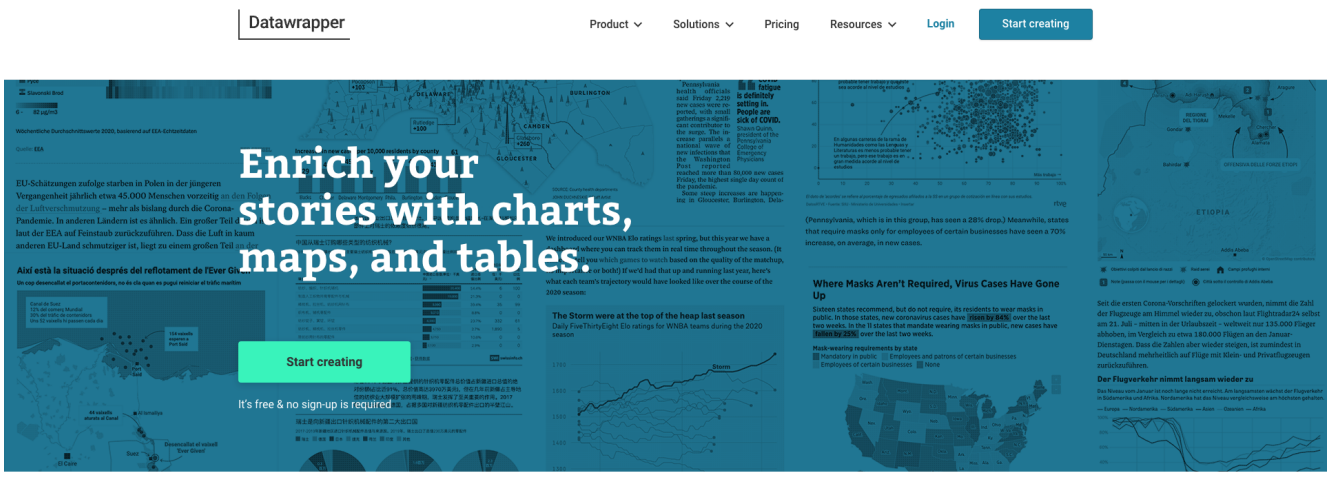


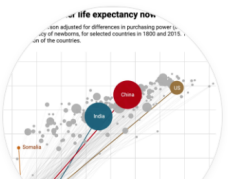
x	y	date	area_m2	color
1	1	1980-09-15	8385641144066	a
1	1	2016-09-15	4449383504112	b

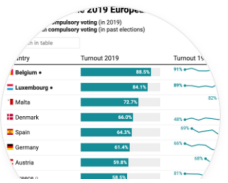
Starting fresh? Try Datawrapper!

Using tools like Datawrapper is a good and no-code way for getting into the topic of data visualisation. It has a lot of styling options but a really well documented and easy to follow step by step process as well. The tool is web-based so you will not need to install any additional software.

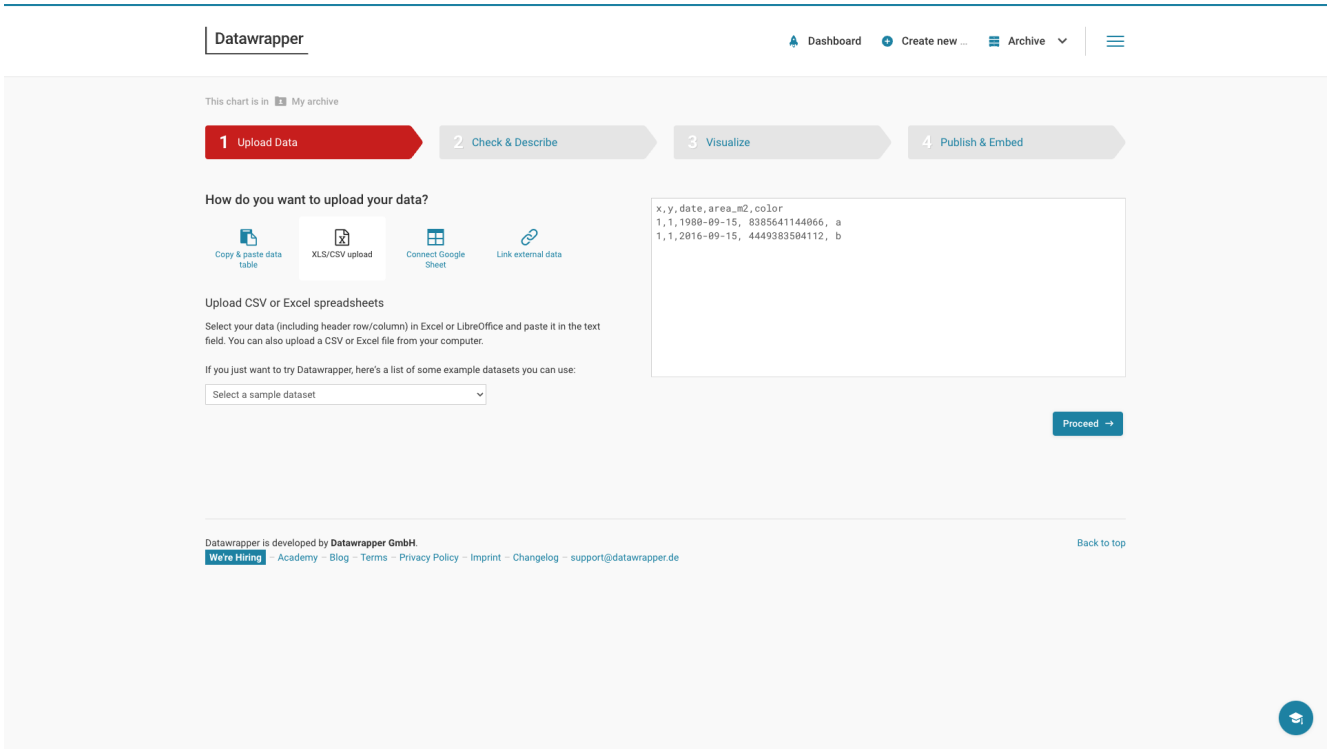
1. Let's create a simple visualisation of the dataset in Datawrapper. Open up https://www.datawrapper.de/ and click on "start creating". It's helpful to create a free account first but you can start right away without doing so.



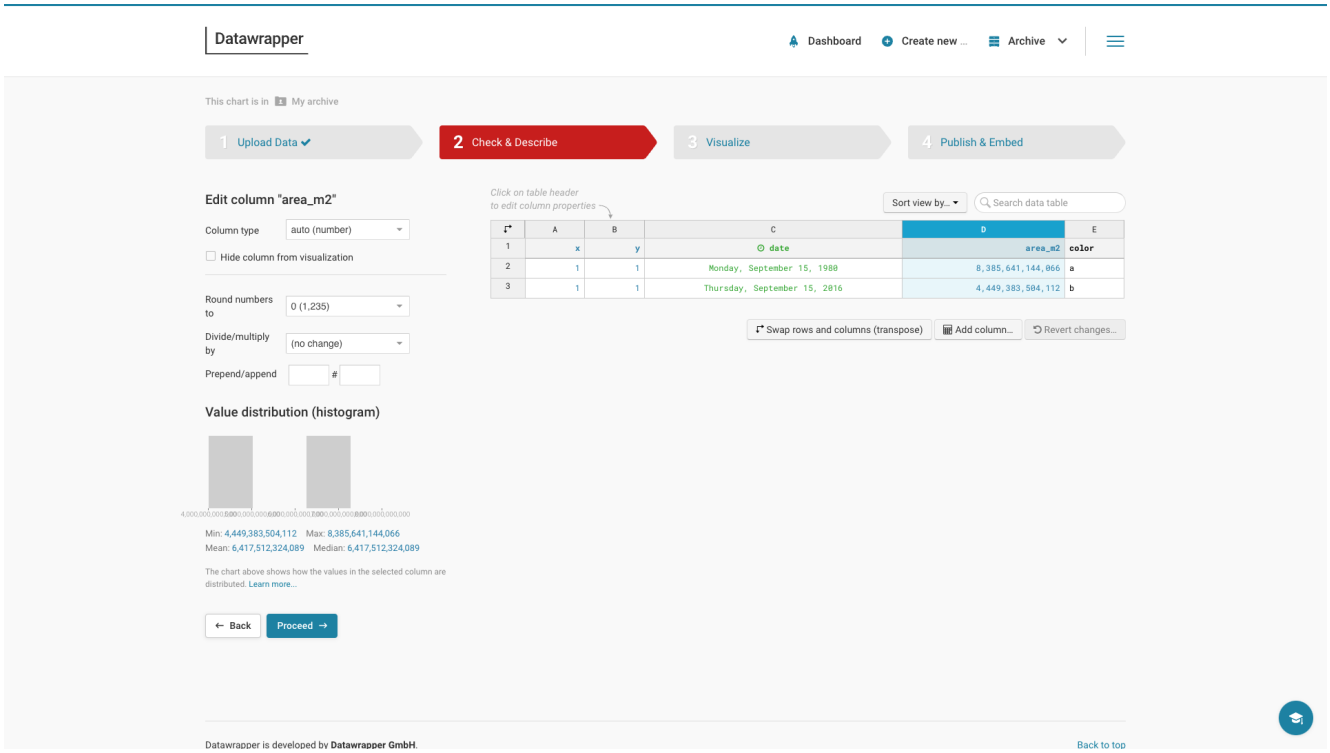




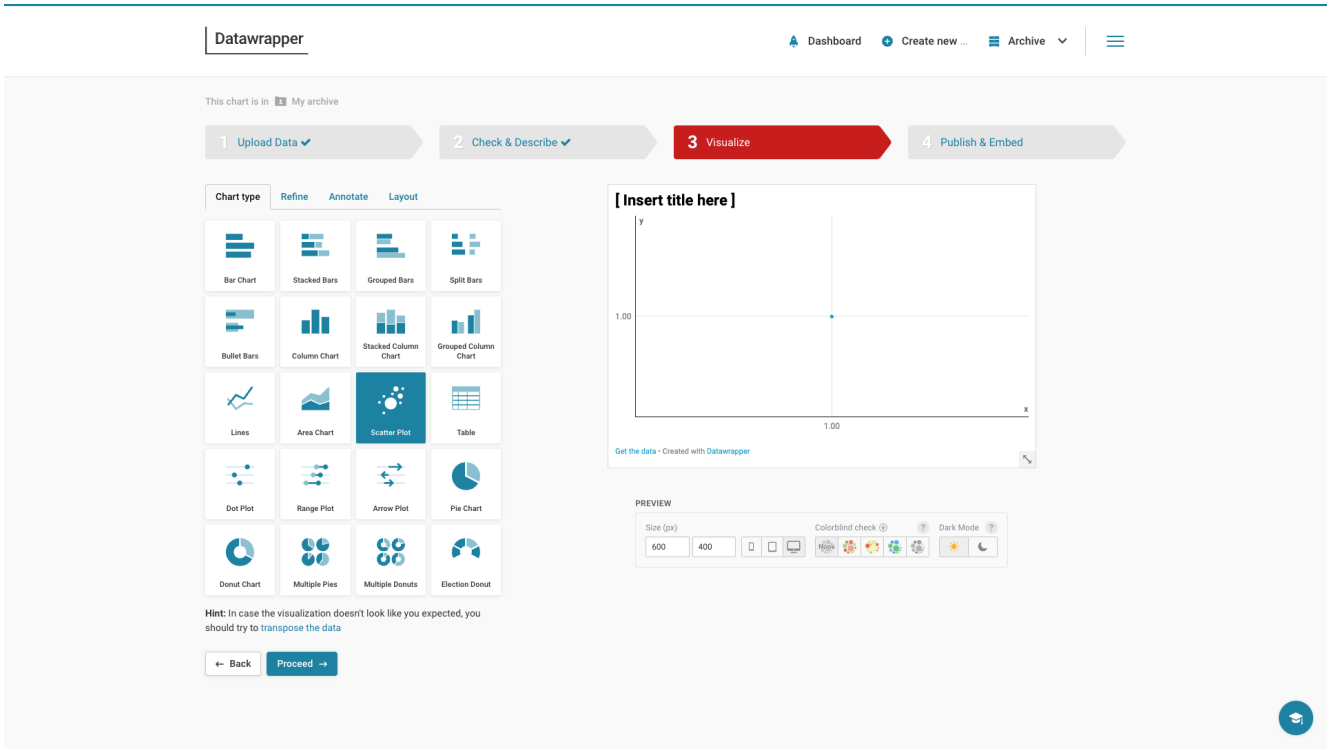
2. Datawrapper has a 4-step approach to creating your chart so first upload the csv file or copy and paste it in the white box. You can always switch back and forth between the individual steps.



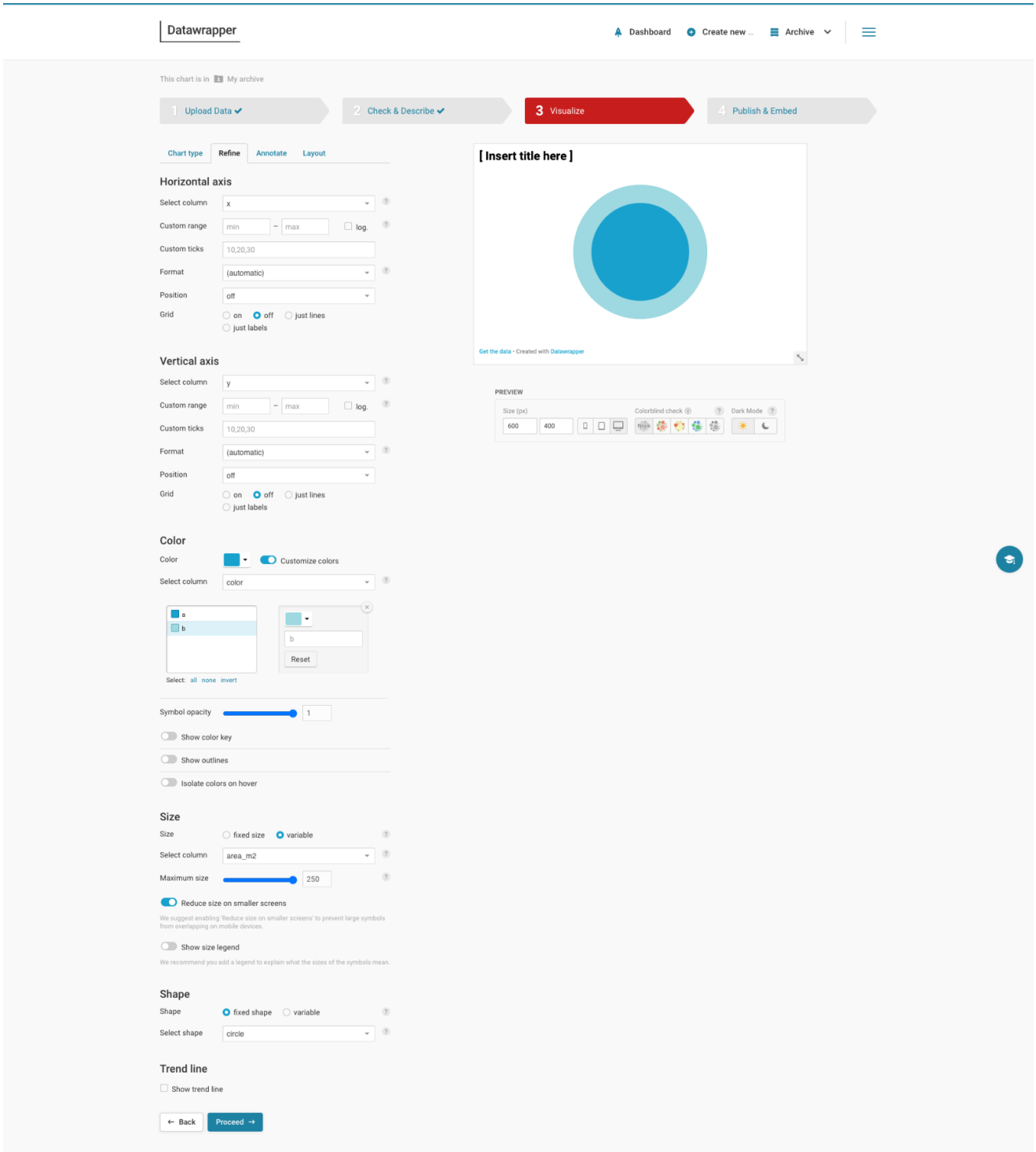
3. Click on “Check & Describe” for a quick look at your dataset. The data should have been parsed correctly into separate columns and the data types will have been inferred, too.



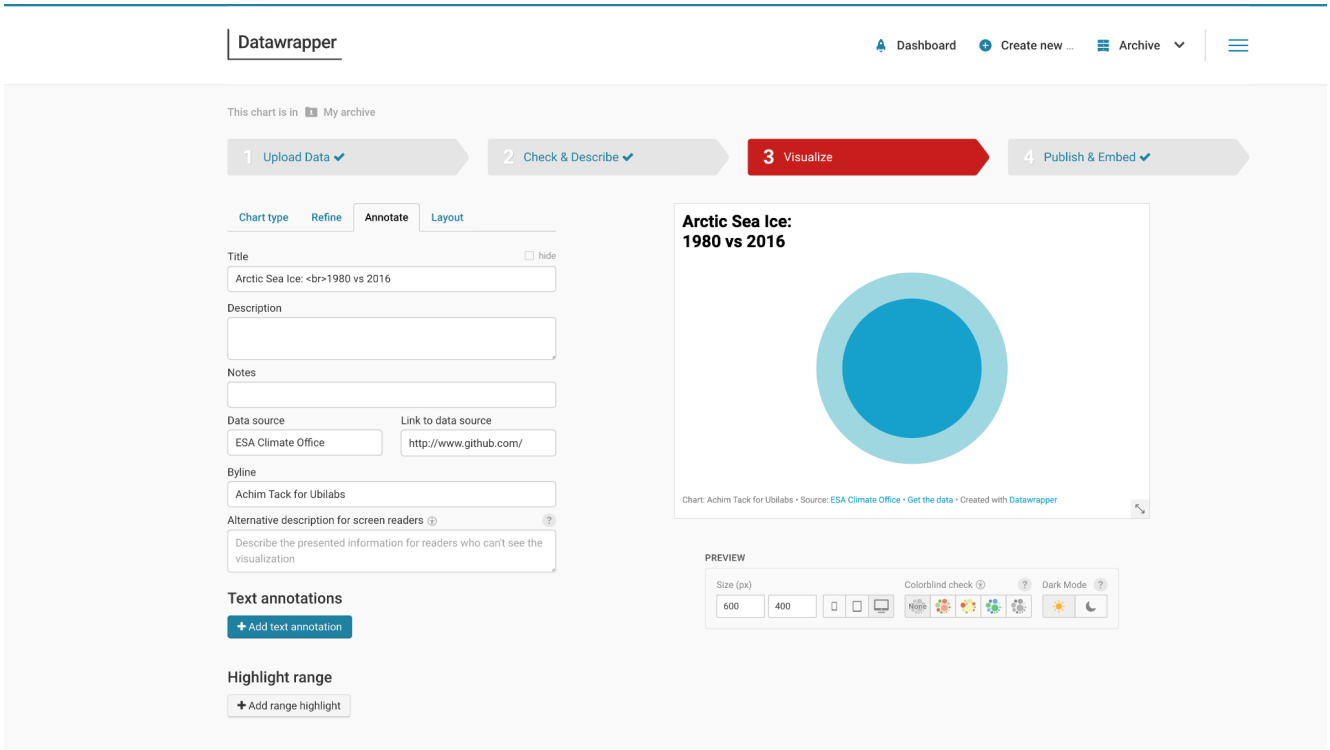
4. Now the boring part is over and you can actually start creating something. Click on the “Visualize” button and choose “Scatter Plot” as the chart type. The resulting chart should look like this. Note that Datawrapper inferred the point coordinates from the x and y columns (both set to 1) so that now we have 2 points, one for each year displayed on top of each other.



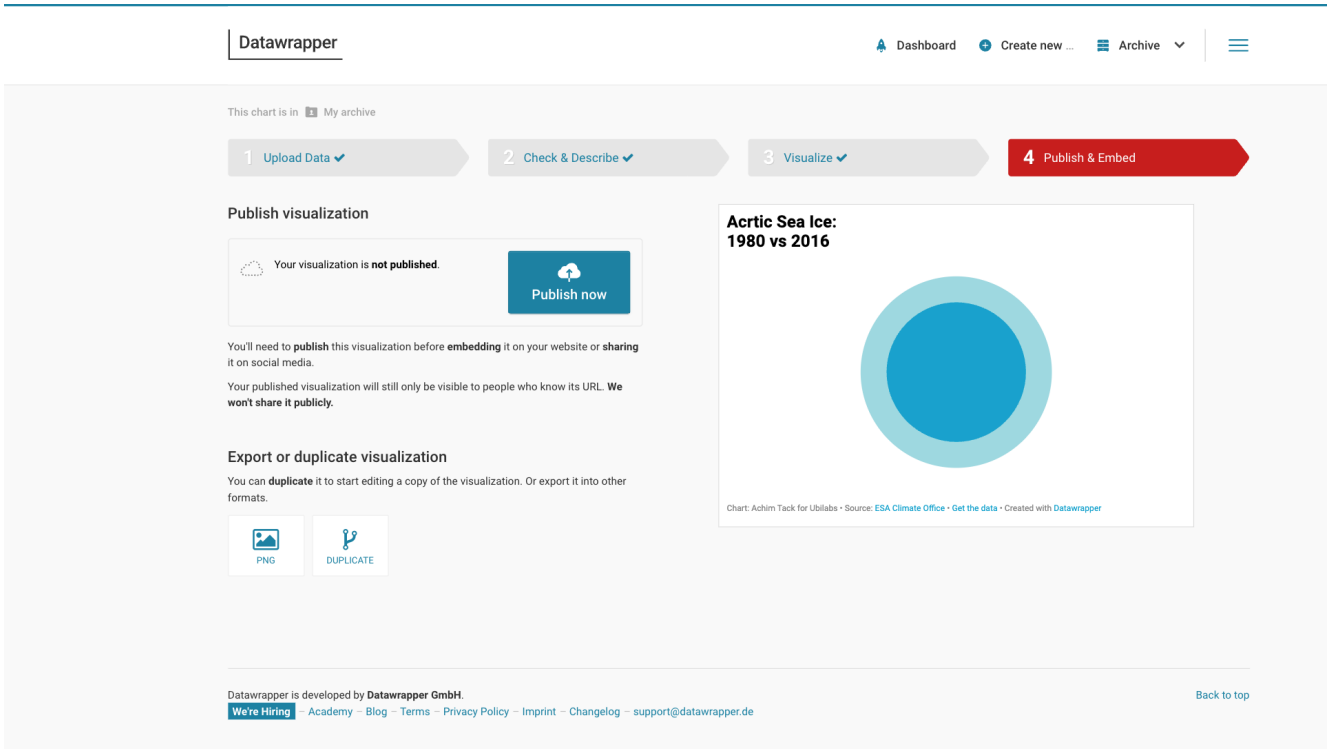
5. Click on “Refine” and change the settings for the chart as shown in the screenshot below. The most important settings for you to recreate the little picture are in the “Size” and the “Color” sections. Experiment with other settings to create different results.



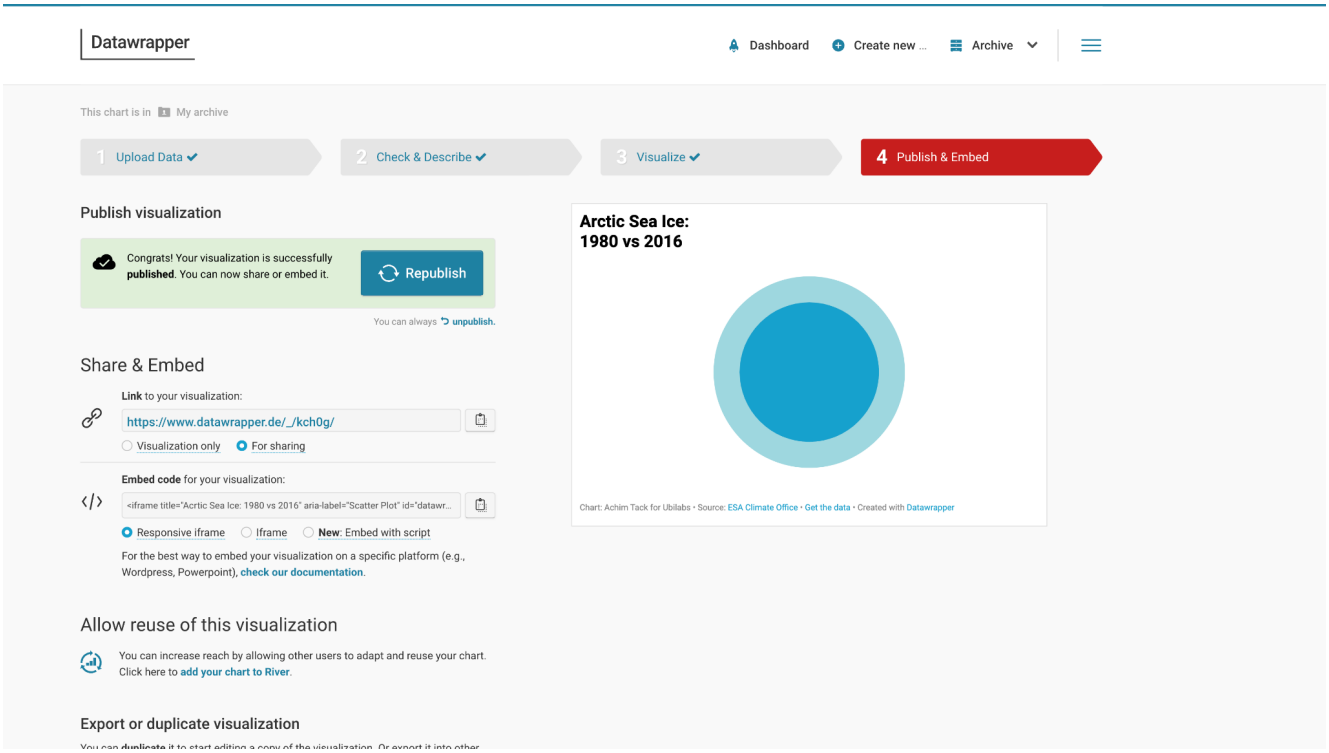
6. Great! You are nearly there. Now click on “Annotate”. In this part of the process you can add a title, links to the datasource and a byline. You can also experiment with different labelling options. For example you could add big numbers for the two years in the circles.



7. Your little picture is complete - the last step is to publish it. Click on “Publish & Embed” and check your little picture for the last time. Does it look ok? Then click on the “Publish Now” button. Note: You can always come back, change the chart and republish it again.



8. Once published you get few options on sharing your work. For example you can copy a link, use an embed to integrate it in your website or download a png file for use in social media.



<u>A few tips we found helpful:</u>

Datawrapper sometimes might have issues with the order of the circles.
Try switching between colors and drag the elements of the color column until it looks right.

When choosing colors it's helpful to use the different color blind options presented below the chart on the "Visualize" tab.

If you really want to explore the options go to https://academy.datawrapper.de/ for more ideas on how to tweak and tune your chart.

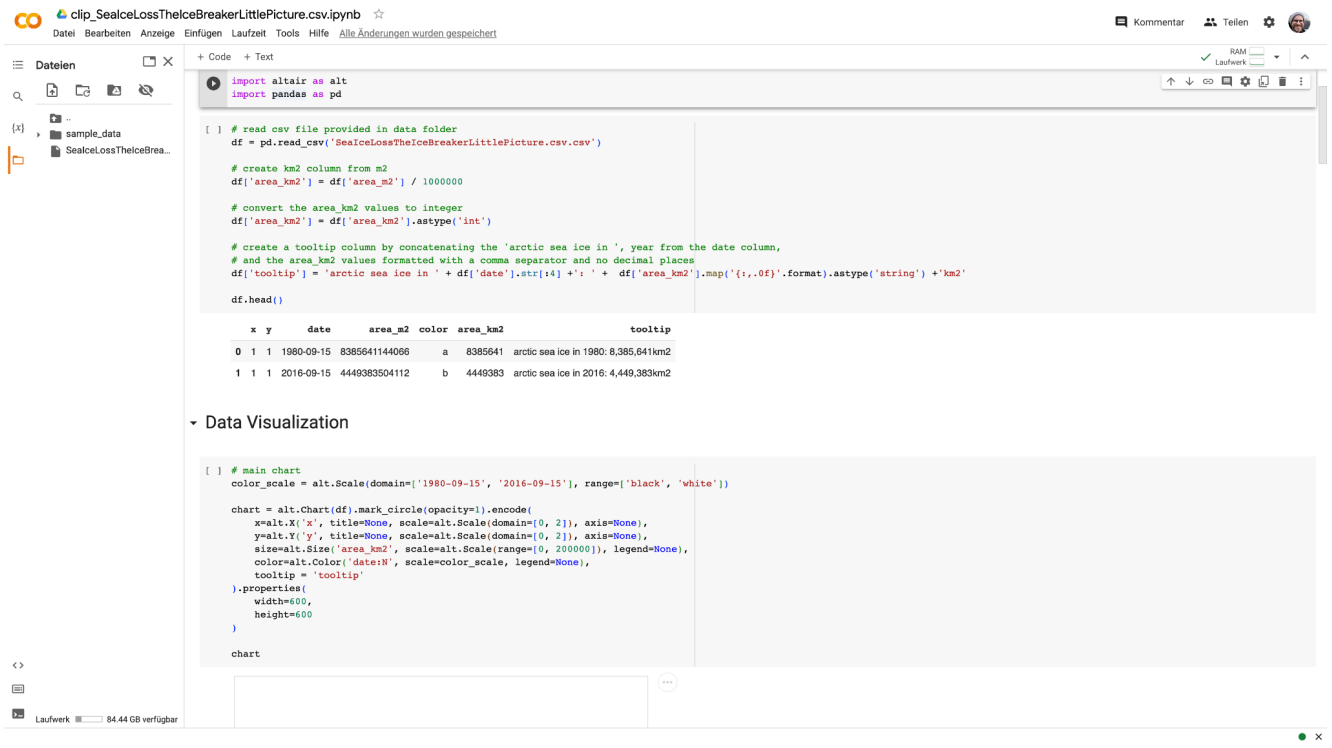
Like to code things? Use a Python Script!

Using off the shelf tools like Datawrapper is a good way to start creating charts and diagrams to be published on the web. If you want a little more control over what you are creating it can be a good idea to use a programming language like python or r to “program your chart”. Using the example above we created a tutorial for python. We created a little tutorial on how to set-up a programming environment on your computer.

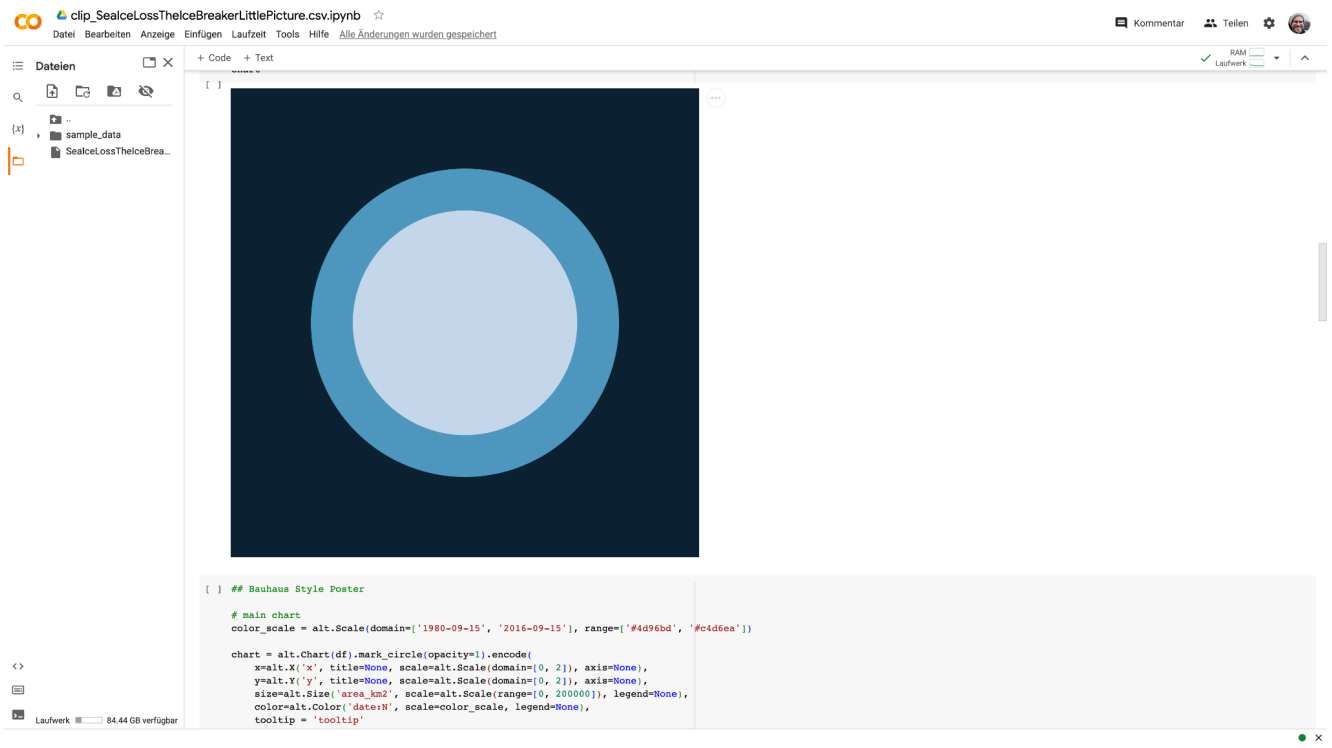
1. Download and open up the jupyter notebook provided for this little picture here: URL

You should be able to start it on most local or hosted environments like Google Colab or noteable.io .

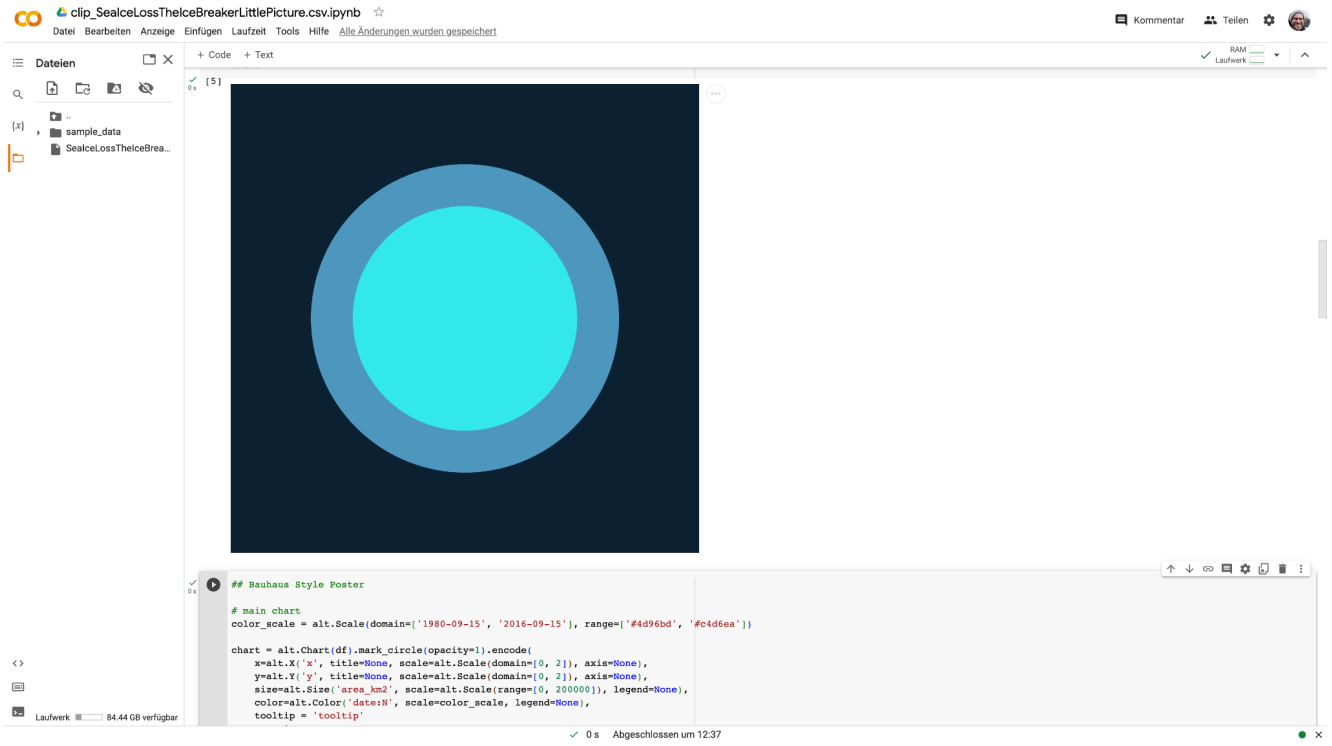
2. Place the csv file in the same folder where your notebook is running from. If you use a hosted version like Google Colab or noteable.io upload the file to the main data folder. Use drag and drop to upload the file.



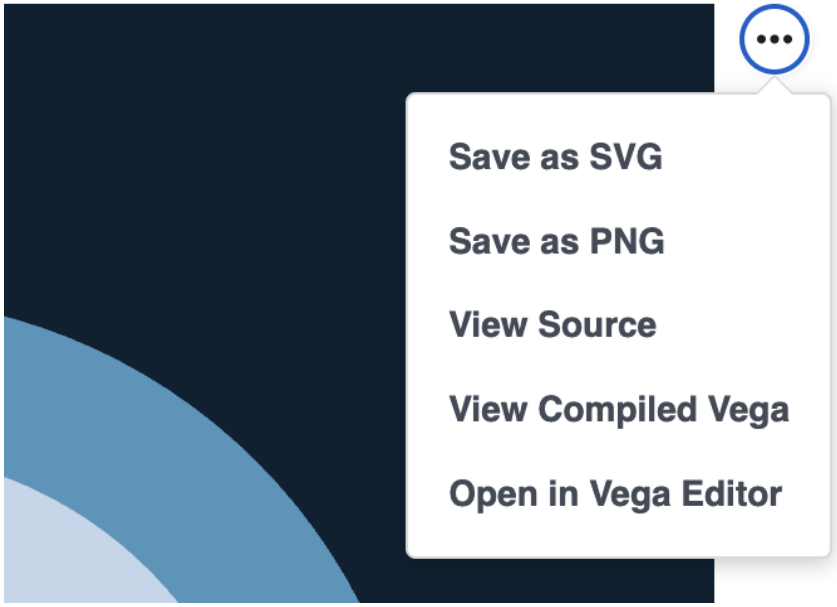
3. The notebook uses the [altair python library] for creating and tweaking the charts. Depending on your environment you might have to install it first. Depending on your environment there should be a menu option called “run all cells” or something named similar. Run the complete notebook and you should be able to see several charts in it. The notebook is using inline comments and markdown formatted cells to explain how the charts are created.



4. The altair library encodes different columns of your dataset into different aspects of your chart. If you are just getting started try to tweak the colors and parameters to your liking. Colors are presented in hex format which can be generated using a [hex color picker].



5. Once finished you can use the three-button menu to the top right of each generated chart to download your little picture as a svg or png. Note that altair is using the vega declarative visualisation language so you can open up the generated code in a separate editor if you like. The svg can be imported into a vector editing software if you want to touch up your little picture a little more.



A few tips we found helpful:

The scripts might take some time to run, especially if you are on a hosted environment.

Jupyter notebooks consist of individual code or markdown cells. You can run each cell by clicking into a cell and pressing “shift” and “enter”. Be aware that cells are meant to be executed from top to bottom.

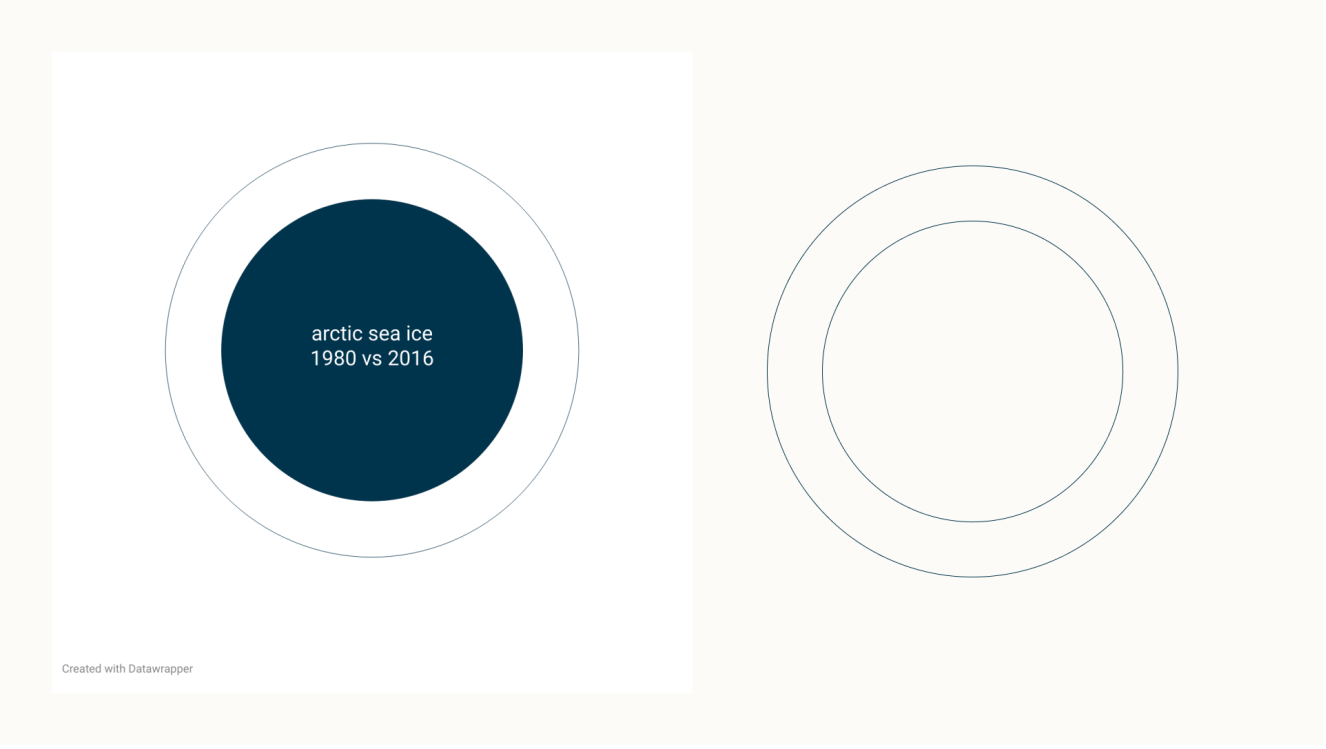
If you want to dive into the code, try changing colors first. Find the line containing the colors in hex format and try to change some of the colors: (`range=['#fbfc93','#f0af3c','#e27711','#b93e04','#7f2a0c','#451104']`)

Want to spice things up a little?

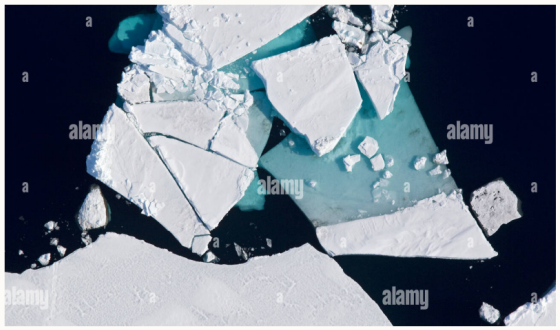
Use vector design software!

Creating little pictures using no-code hosted solutions like Datawrapper or scripting them using a programming language is great but if you want to emotionally charge and polish your little picture a bit more it is helpful to use a vector editing software on top of your previous outputs. In the following steps we are writing down the steps agnostic to a specific software.

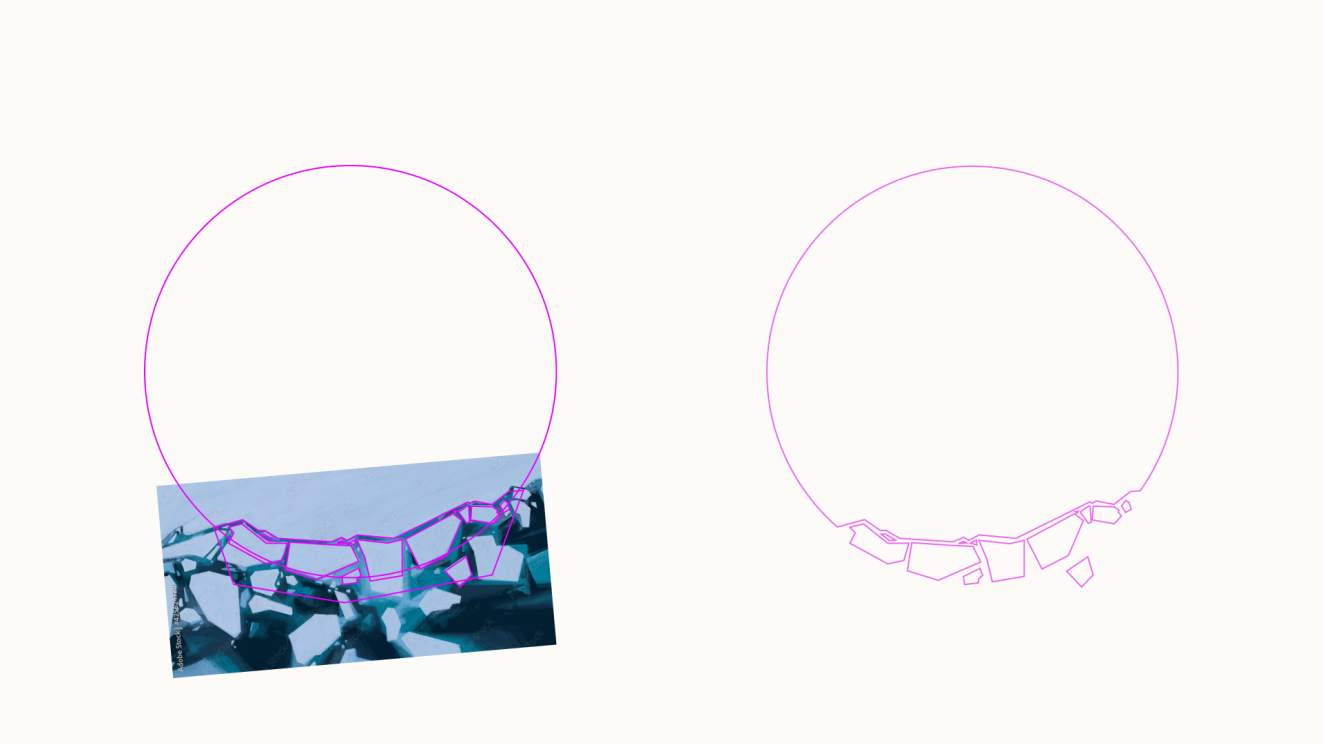
1. Open up your preferred vector editing software and start by dragging the .svg exported from the python script in the last section into a new document or open/import the .svg.
Extract the shapes by ungrouping all elements (Strg Shift G). If you added any text or other elements in python delete those unneeded shapes. The result should look like the right of the two circles in the picture below.



2. Search on the web for images that describe your little picture the best way possible. In this case we used ”arctic ice top view” as a search.

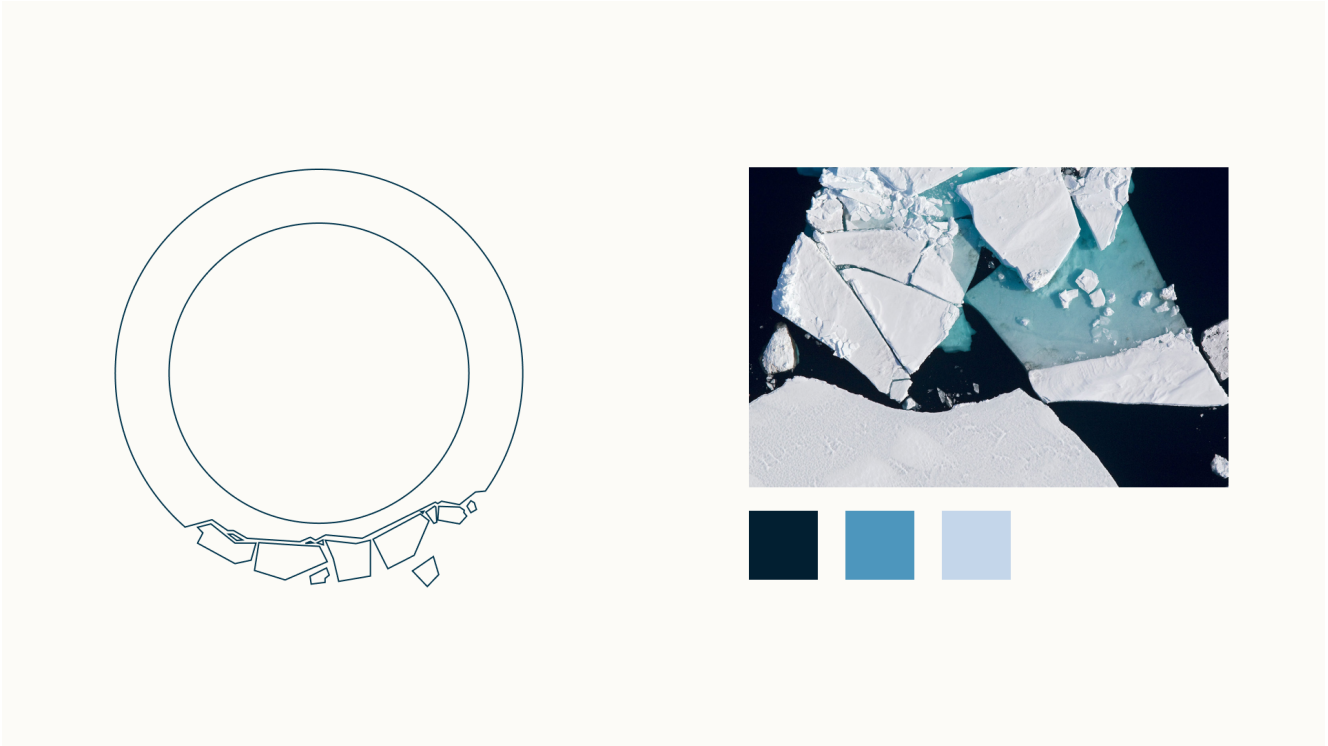


3. Drag/import the image into the software and place it below the svg lines. Now it get's a little tricky: Scale and place the image underneath the .svg so that it fits the size of the circles. Use the pen tool to retrace elements from the image as part of the outer circle. Keep the inner circle as-is.

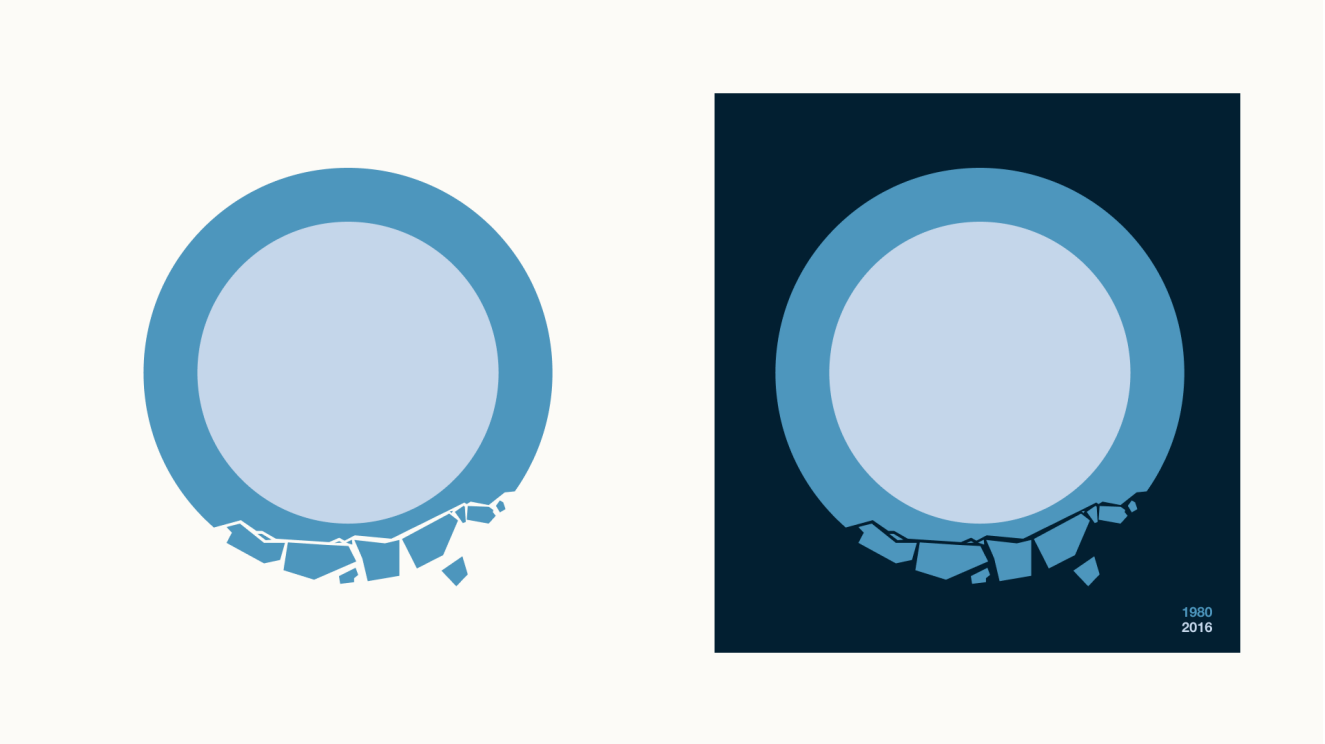


4. Next step will be the coloring. It's helpful to extract color schemes from the images searched before so that ice sheet fragments have a realistic, yet bold coloring. You can use

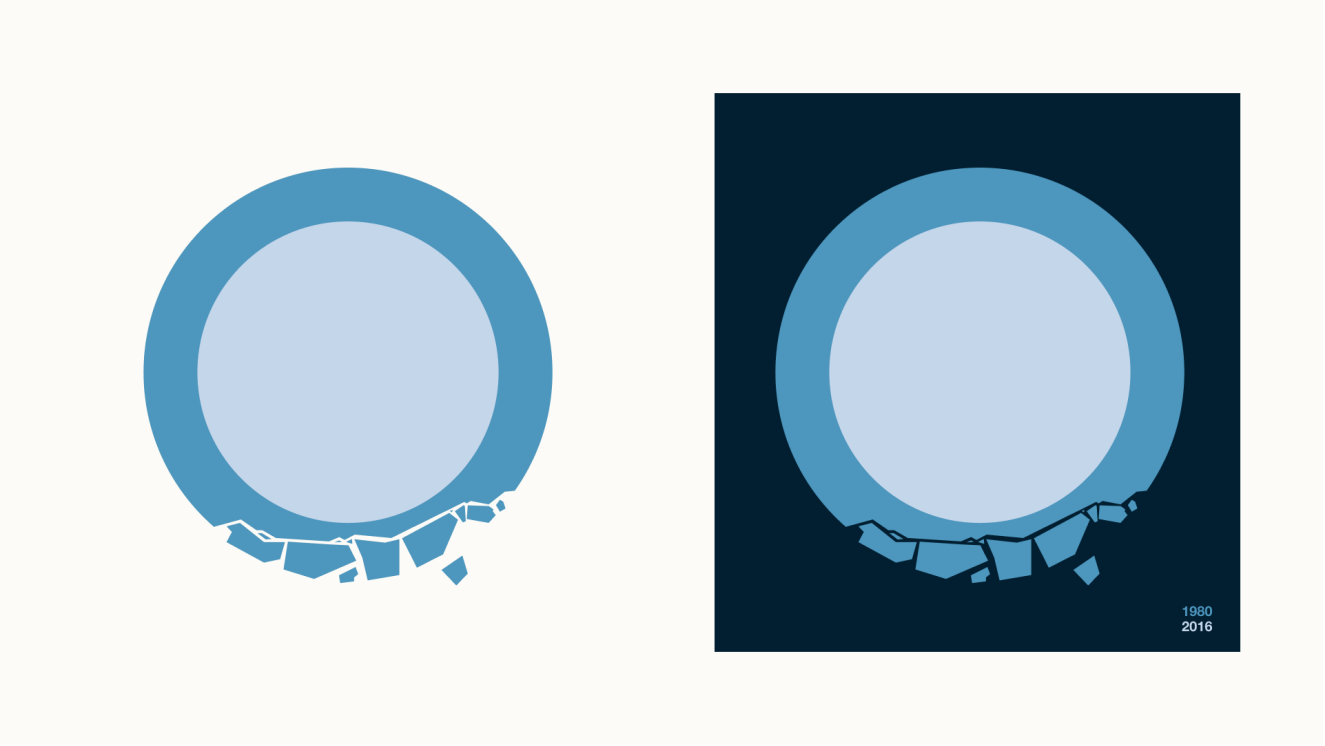
tools like https://coolors.co/image-picker to extract palettes from the images searched. In this example we came up with the colors #0d2039, #6d9ec7 and #cfdaee .



5. Select all elements from the outer circle and set them to the color chosen. As the outer ring is supposed to represent the ice already melted we chose a darker shade of blue (#6d9ec7). Use a lighter blue (#cfdaee) for the inner ring. We added a square as an outline element (#0d2039) as well as color coded legend elements using the text tool. Save the left of the two images as a new .svg file.



6. Now you are almost finished. It's helpful to add a title and some more info to the little image by adding a footer. For this you can open our template footer provided here. Open up the footer file, and drag the .svg file saved in the step before into the template. Change the title and subtitle and save the now complete little picture in a format of your choosing. Congratulations, you have successfully completed the tutorial!



A few tips we found helpful:

- ➔ Not every image fits your needs for the little picture .
  Try different options to see how the results might look like with different approaches.
- ➔ When you use an image as a reference for the colors, make sure that the colors you pick differentiate enough.

If you really want to explore the options go to https://academy.datawrapper.de/ for more ideas on how to tweak and tune your chart.