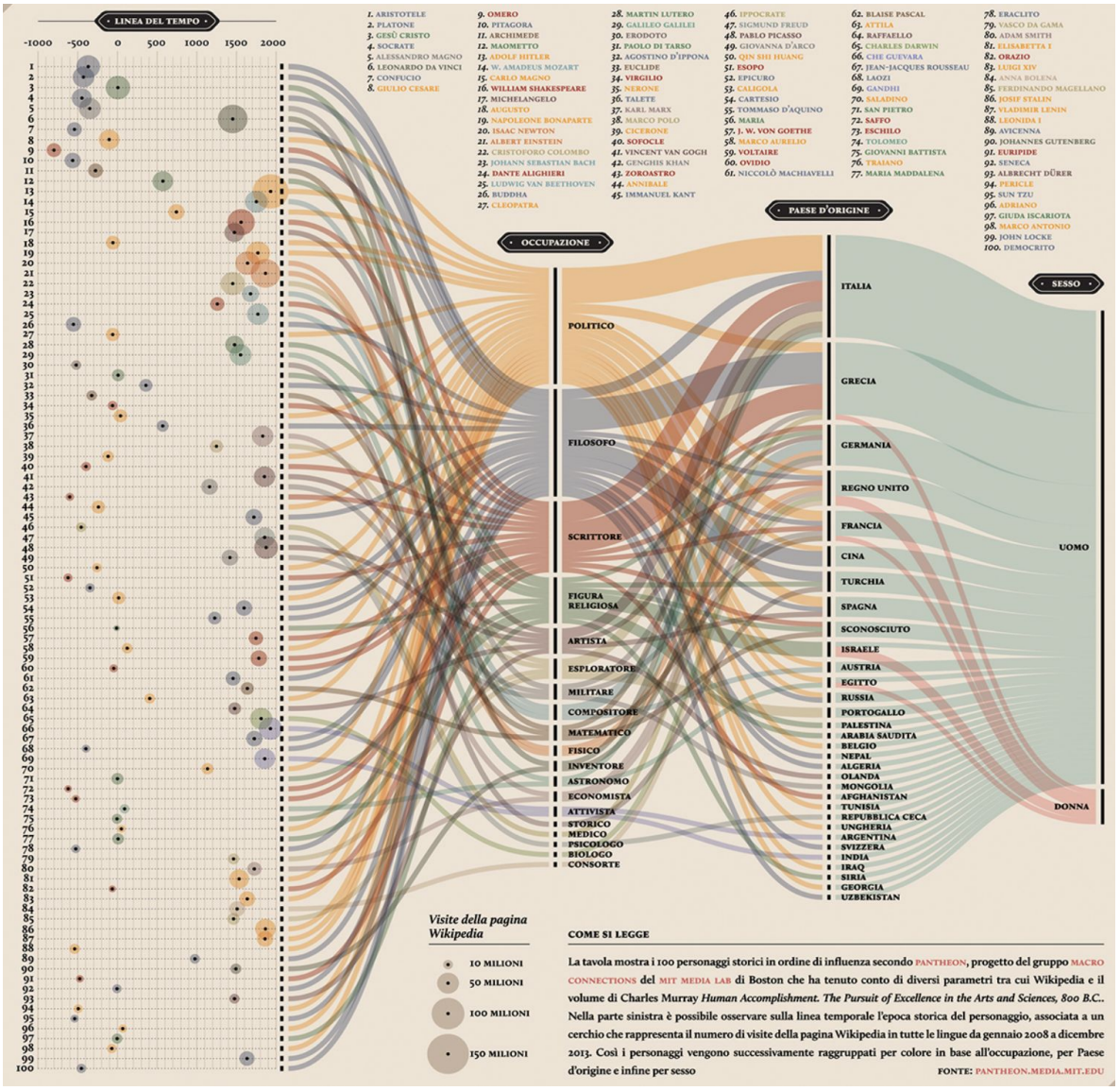
CSCI E-79: The Art and Design of Information

Spring 2019

LAB 2

Creating visualizations with tools

Part 1: RAWGraphs



Source: Behance

Index

-- Introduction
01 / What is *RawGraphs*
02 / How to produce visualizations with *RawGraphs*
-- Credits and Additional Resources

Introduction

Welcome to E-79 Lab 2!

The second lab will guide you through the creation of visualizations with a tool named *RawGraphs*. This type of tools help in generating unpolished visualizations, ready for further implementation and/or redesign. It will help you familiarizing with different visual models, data, and visual variables. With *RawGraphs* you can create less or more complex visualizations. We find it very interesting for creating visually pleasing and complex layouts in a short amount of time.

Every design process starts with a sketch. As you have learned, sketch helps clarifying design ideas and determining the fundamentals of the project. This lab is the second step in the information design process. After knowing how to define you visual goal, you can proceed with further exploration and implementation.

Note! The teaser picture was created using *RawGraphs* and then redesigned using *Adobe Illustrator*.

Topics and Learning Objectives

- transform your concept from a sketch to a vector drawing
- preparing dataset for *RawGraphs*
- creating visualization using *RawGraphs*
- understand and combine data with visual variables
- experiment with visual models
- customize your visualization
- export an SVG file

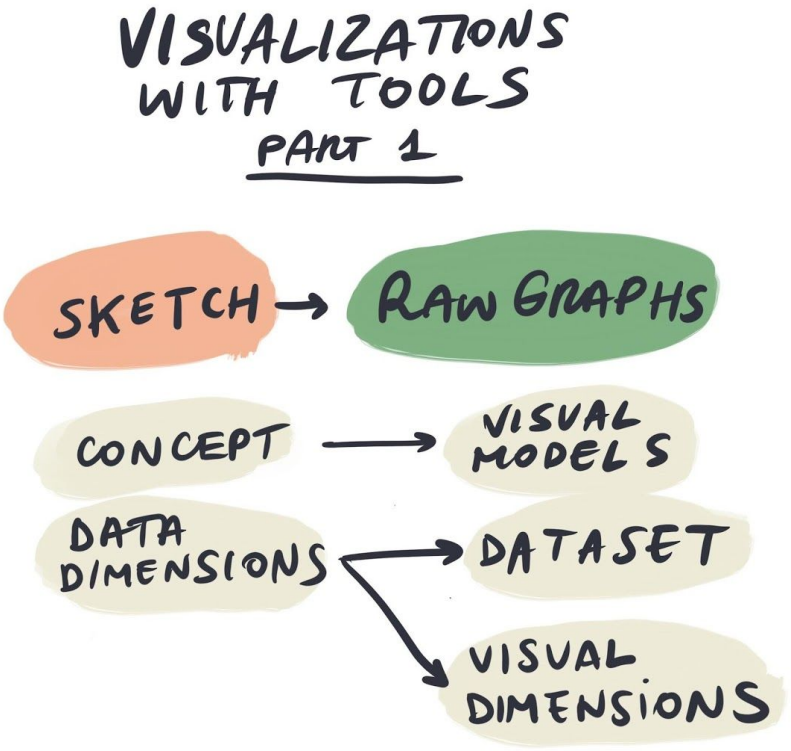


Figure 1: Lab 2 topics and learning objectives

What is *RawGraphs*?

RawGraphs is an online open tool created by *DensityDesign* and *Calibro* with the purpose of creating visualizations quickly, those difficult or time-consuming to implement using script languages. It is an easy to learn tool, enabling fast creation of very complex visual layouts. The final output is a semi-finished or *raw* visualization, that can be exported to other tools (e.g. *Illustrator*) for further redesign and/or implementation.

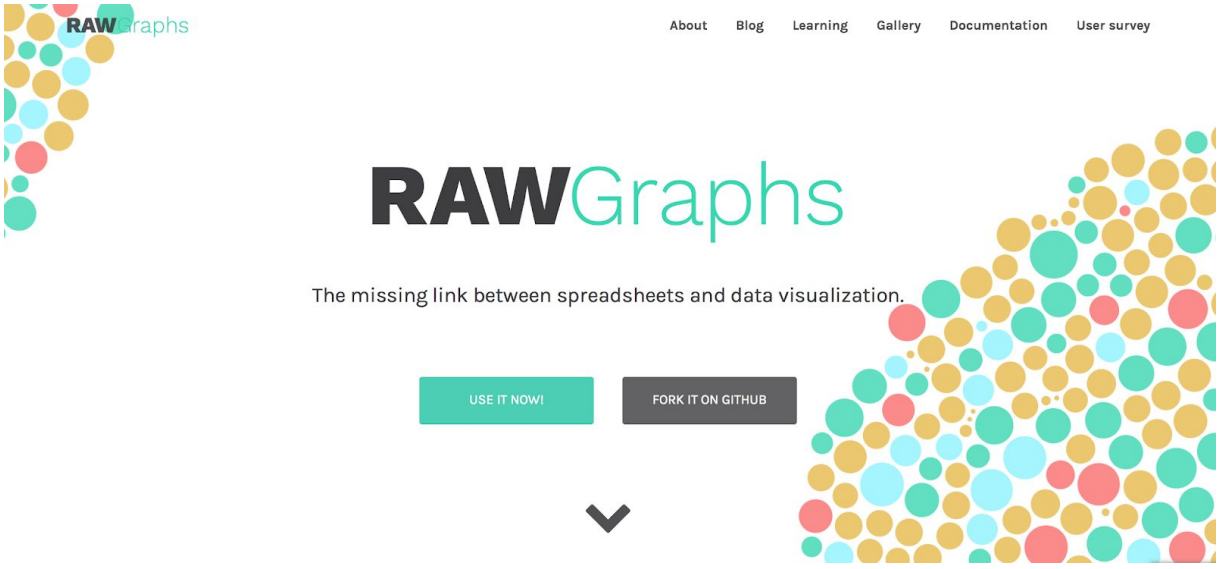


Figure 2: rawgraphs.io

PROS of *RawGraphs*:

- creates complex visual models
- very easy to learn and fast in generating new visualizations
- the variety of models proposed is always growing
- open source project! everyone can contribute to its development
- exported SVG files are completely editable

CONS (or **something to be aware of**):

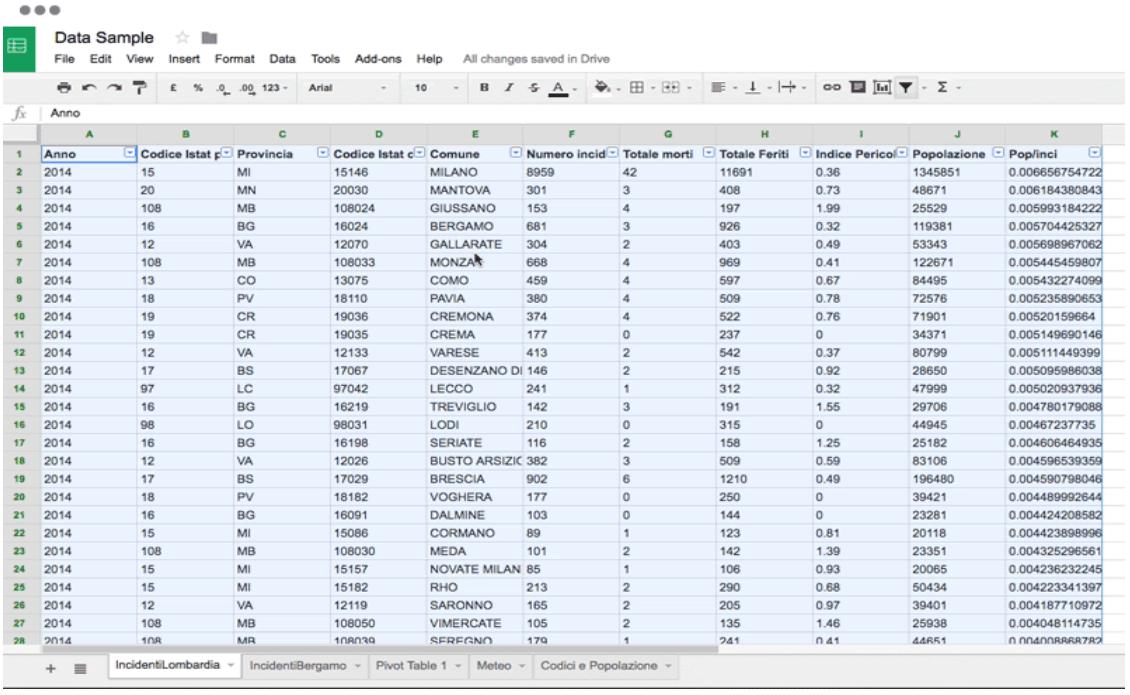
- the dataset should be formatted in a very specific way
- the need for an appropriate data variable available
- it can be very inefficient due to a specific data structure constraint

The process in *RawGraphs* starts with understanding your data and transforming features to better fit chosen visual models. Why do you have to take care of the data formatting? Because, *RawGraphs* can produce the desired visual model, but you would need to help in telling how to do it and what features to take into consideration.

For example, you have a dataset representing a time series. Having know that line graphs might be the best way to visualize the time series will give you a clue how to reorganize the dataset as *RawGraphs* expects it. Afterwards, you can experiment with additional models inside the *RawGraphs* choosing different time series representations (e.g. bump chart, streamgraph... etc). At the end you can easily export final solution for further customization or implementation.

How to produce visualizations with *RawGraphs*

1/ Loading data



	A	B	C	D	E	F	G	H	I	J	K
1	Anno	Codice Istat p	Provincia	Codice Istat c	Comune	Numero incid	Totale morti	Totale Feriti	Indice Pericol	Popolazione	Pop/inci
2	2014	15	MI	15146	MILANO	8959	42	11691	0.36	1345851	0.006656754722
3	2014	20	MN	20030	MANTOVA	301	3	408	0.73	48671	0.006184380843
4	2014	108	MB	108024	GIUSSANO	153	4	197	1.99	25529	0.005993184222
5	2014	16	BG	16024	BERGAMO	681	3	926	0.32	119381	0.005704425327
6	2014	12	VA	12070	GALLARATE	304	2	403	0.49	53343	0.005698967062
7	2014	108	MB	108033	MONZA	668	4	969	0.41	122671	0.005445459807
8	2014	13	CO	13075	COMO	459	4	597	0.67	84495	0.005432274099
9	2014	18	PV	18110	PAVIA	380	4	509	0.78	72576	0.005235890653
10	2014	19	CR	19036	CREMONA	374	4	522	0.76	71901	0.0052015966
11	2014	19	CR	19035	CREMA	177	0	237	0	34371	0.005149690146
12	2014	12	VA	12133	VARESE	413	2	542	0.37	80799	0.005111449399
13	2014	17	BS	17067	DESENZANO DI	146	2	215	0.92	28650	0.005095986038
14	2014	97	LC	97042	LECCO	241	1	312	0.32	47999	0.005020937936
15	2014	16	BG	16219	TREVIGLIO	142	3	191	1.55	29706	0.004780179088
16	2014	98	LO	98031	LODI	210	0	315	0	44945	0.00467237735
17	2014	16	BG	16198	SERIATE	116	2	158	1.25	25182	0.004606464935
18	2014	12	VA	12026	BUSTO ARSIZIO	382	3	509	0.59	83106	0.004596539359
19	2014	17	BS	17029	BRESCIA	902	6	1210	0.49	196480	0.004590798046
20	2014	18	PV	18182	VOGHERA	177	0	250	0	39421	0.004489992644
21	2014	16	BG	16091	DALMINE	103	0	144	0	23281	0.004424208582
22	2014	15	MI	15086	CORMANO	89	1	123	0.81	20118	0.004423898996
23	2014	108	MB	108030	MEDA	101	2	142	1.39	23351	0.004325296561
24	2014	15	MI	15157	NOVATE MILAN	85	1	106	0.93	20065	0.004236232245
25	2014	15	MI	15182	RHO	213	2	290	0.68	50434	0.004223341397
26	2014	12	VA	12119	SARONNO	165	2	205	0.97	39401	0.004187710972
27	2014	108	MB	108050	VIMERCATE	105	2	135	1.46	25938	0.004048114735
28	2014	108	MB	108039	SEREGNO	179	1	241	0.41	44651	0.004008868782

Figure 3: Import data from Excel
click on the image to see animated steps

RawGraphs works with a set of different file types:

- delimiter-separated (e.g. csv or tsv)
- copied-and-pasted texts from other applications (e.g. Microsoft Excel, Google Spreadsheets or TextEdit)
- CORS-enabled endpoints (APIs)

RAWGraphs is a web app and your data will be processed by the web browser. No server-side operations or storages are performed. There are a few different ways to select your data:

A. Use samples datasets
Samples data is very useful to understand what kind of data structures *RawGraphs* exects as well as what typologies of data you can play with.

B. Copy-paste your data from Excel
C. Upload dataset from your computer
D. Upload data using URL

Bare in mind to format of your data before exporting!

Put some time and take a look at the data samples provided by *RawGraphs*. Apply the same structure. If the data is not organized in the right way, you will not be able to proceed with Lab 2 exercises.

2/ Pick a visual model

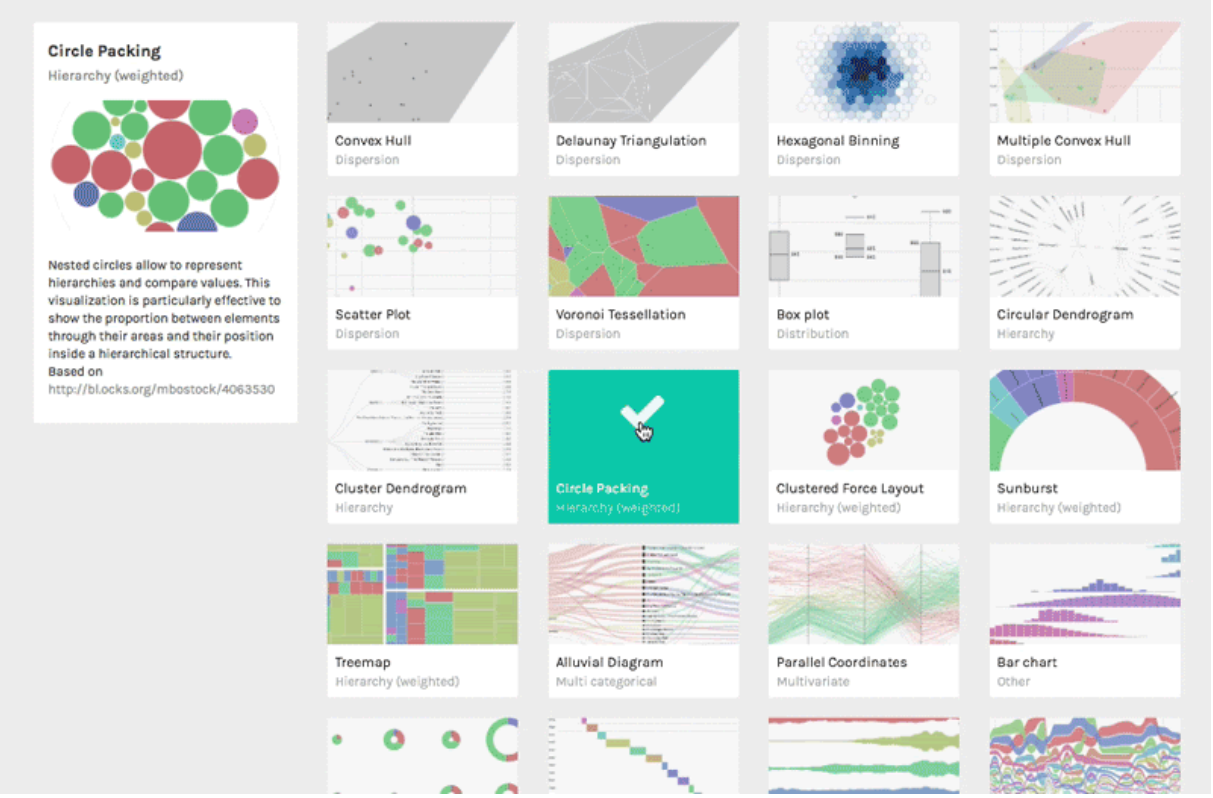


Figure 4: Choosing visual models
click on the image to see animated steps

In *RawGraphs* you will find conventional and unconventional models, from barcharts to alluvial charts. This tool makes easy generation of charts based on your data that in general are not easy to produce with other tools. Since *RawGraphs* is an open source tool, everyone can contribute to the development of new models. Great thing is that the library of visual models is growing with every new day. What makes this tool even more attractive is that all models are built on top of d3.js library!

Note! Before proceeding we would recommend to play with importing the data and/or to look at the data samples provided by *RawGraphs* in order to familiarize yourself with a demanded data structures. After having an idea of how to pre-process and import the data, you may proceed with the following examples.

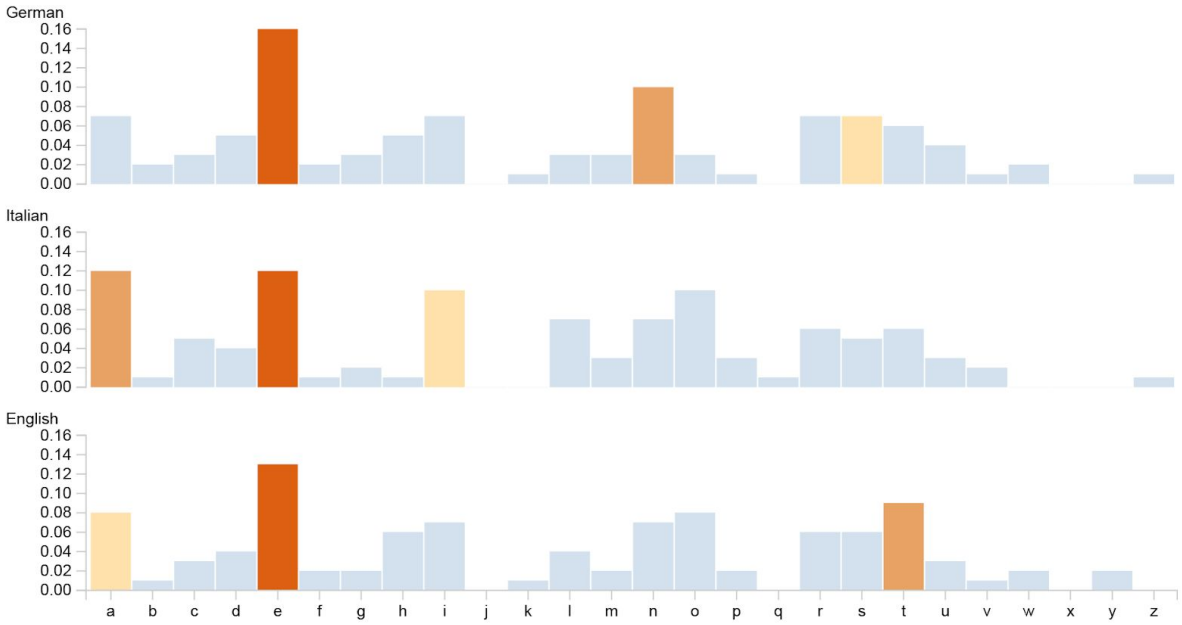


Figure 5: Bar Charts

Example 1 - Barchart

Model: barchart
Dataset (from samples): "Most frequent letters per language"

Take a look at the Figure 6 and than look back to your data. From the dataset you must choose at least 2 columns (features):

1. **Letters:** name of the categories
2. **Frequency:** values
3. **Language**: name of the group (only for multiple bar chart)
4. **Rank:** bar color

Select these 2 columns (features) and in a few seconds you should be able to see the flash message:

"78 records in your data have been successfully parsed!"

In case the process does not go your way, we would suggest taking a second look at the Figure 6 for a possible missing detail.

Note 1 — You should always look for green flash message after uploading your data and before proceeding with the next steps.

Note 2 — The dataset structure is often referred as 'stacked' or 'narrow'. If you have data as wide/unstacked form, you can use the "stack/unstack" button.

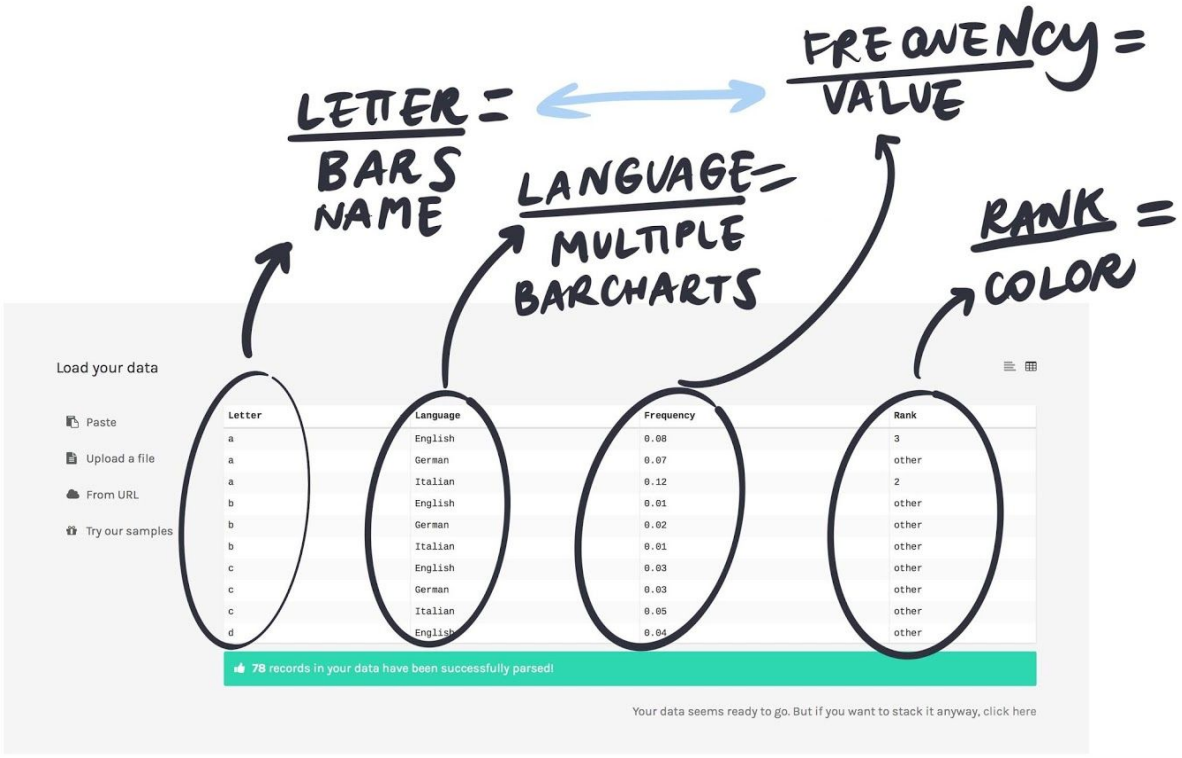


Figure 6: Details regarding data loading

Example 2 - Alluvial Diagram

Model: Alluvial Diagram
Dataset (from samples): Titanic's Passengers

The alluvial diagrams are composed of the three main elements:

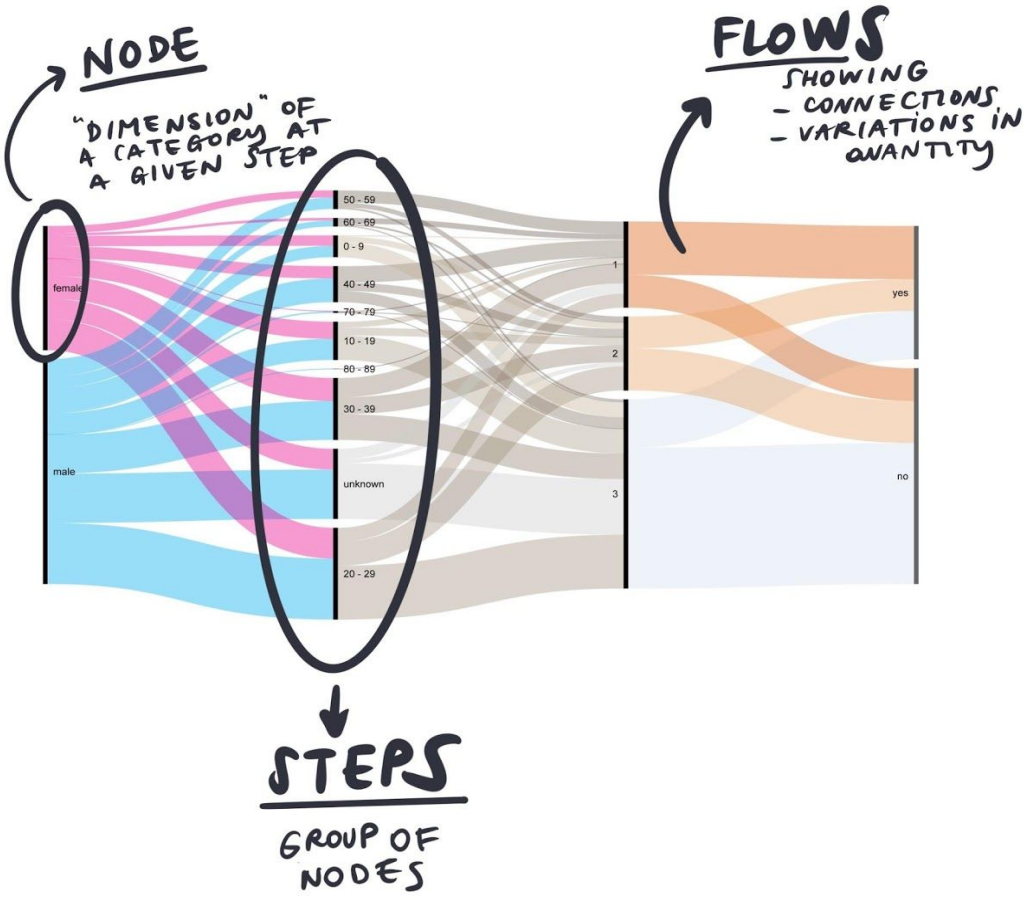


Figure 7: Alluvial Diagram components

Alluvial diagrams represents weighted flows among nodes. They are a specific kind of *Sankey* diagrams using the same logic to show how the same set of items can be regrouped according to a different dimensions.

In *RawGraphs* the size of edges among nodes isn't defined directly. Each line of the dataset becomes a series of flows and each column defines the step. The size of edges and of flows is therefore linked to the number of lines in the dataset containing the same group of nodes. Flows coming from and going to the same nodes are grouped into same categories.

The data we are going to work with must be structured in a following way:

- each line of the dataset is an item
- each column is a property of such item

An example:

NAME	AGE	CITY
John	30	London
Richard	20	Berlin
Frank	20	Berlin
Gabriele	30	Milan

The data must contain at least two columns (features). Each additional column will define a new step in the alluvial diagram. An additional column can be used to define the line weight.

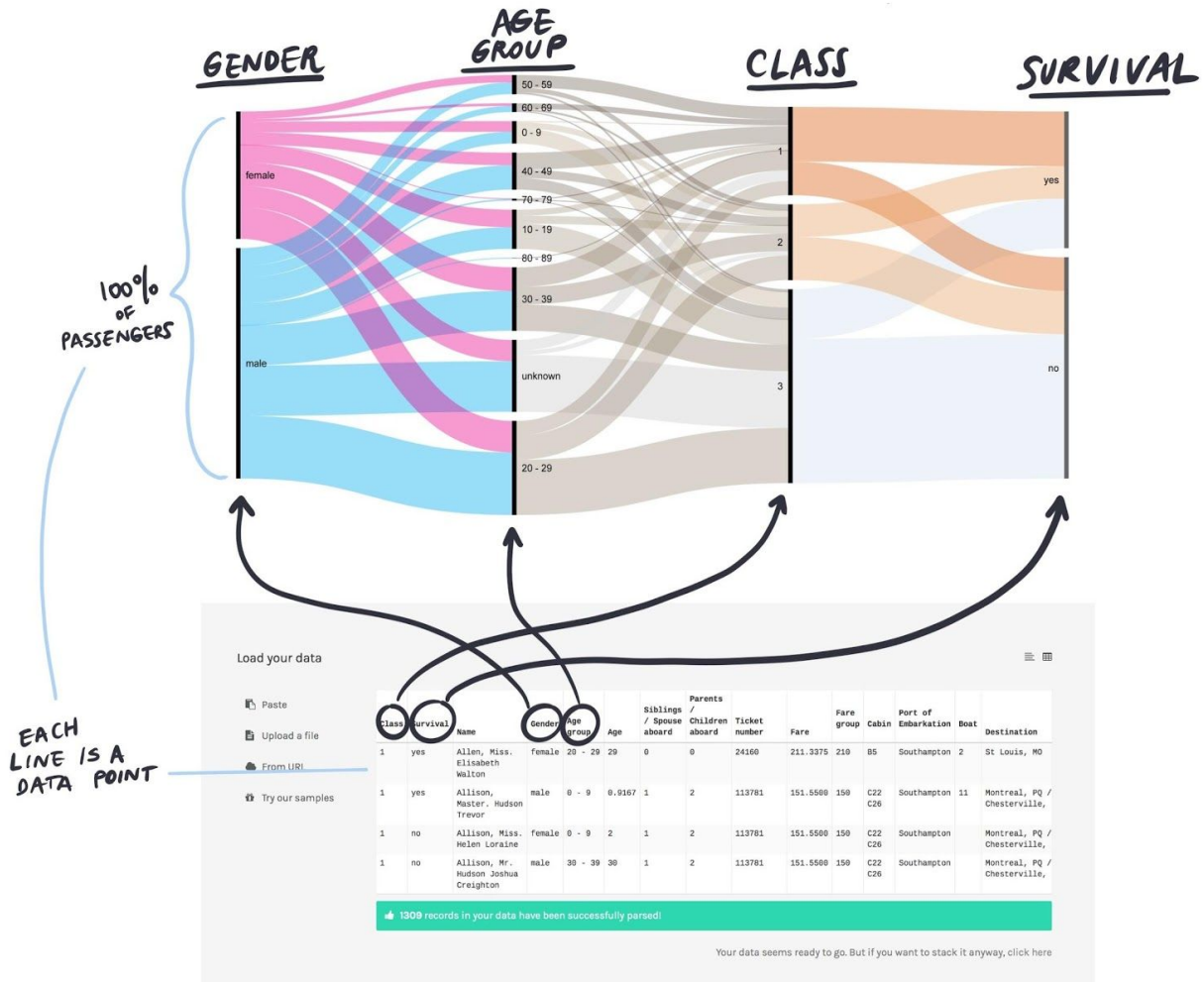


Figure 8: Alluvial Diagram features

Look at the data and keep in mind the structure if you want to work with this type of visual models. If you want to create 4 steps as presented with Figure 8, you need to have at least 4 columns (features) selected. In that case you will have:

- **Gender** with two nodes, male and female. The size of the nodes is proportional to the number of nodes containing the value.

- **Age** group composed of 9 nodes:

    - the first 8 for each decade
    - last one for unknown age or missing data

    In this case, the size of each node is proportional to the number of rows containing the value. The flows among nodes represents the number of lines in the dataset sharing the same values (for example, the largest flow is from the node 'male' in the Gender and the '20-29' node in the Age group).

- **Class** contains three nodes (first, second, third).

- **Survival** contains two nodes (yes for survived, no for not survived).

Take a look at the Figure 8 once again and make sure you have accomplished the previous 4 steps.

Exercise - Select a set of data to visualize

In the first lab you came up with a sketch of visualization representing some data. We asked you to pick one or more visual models, combine them, and create a visual layout. That sketch can be considered the starting point for the creation of a visualization using tools.

However, before opening the *RawGraphs* and choosing a visual model, you have to "rethink" your sketch. In other words, you have to "decompose" the visualization using different elements and to understand:

- what visual models you are going to use
- how many visual models will be used
- what features you will be using (from the entire dataset), and
- how the data has been combined

At this point you can start exploring the visual models you need in *RawGraphs* or start organizing your dataset/s in Excel, respecting the requested structure.

Note! In order to understand what's the right format in which you have to parse your data, just select a sample dataset and experiment with visual models. At the end you will see how the data are organized and what data will work the best for the models chosen.

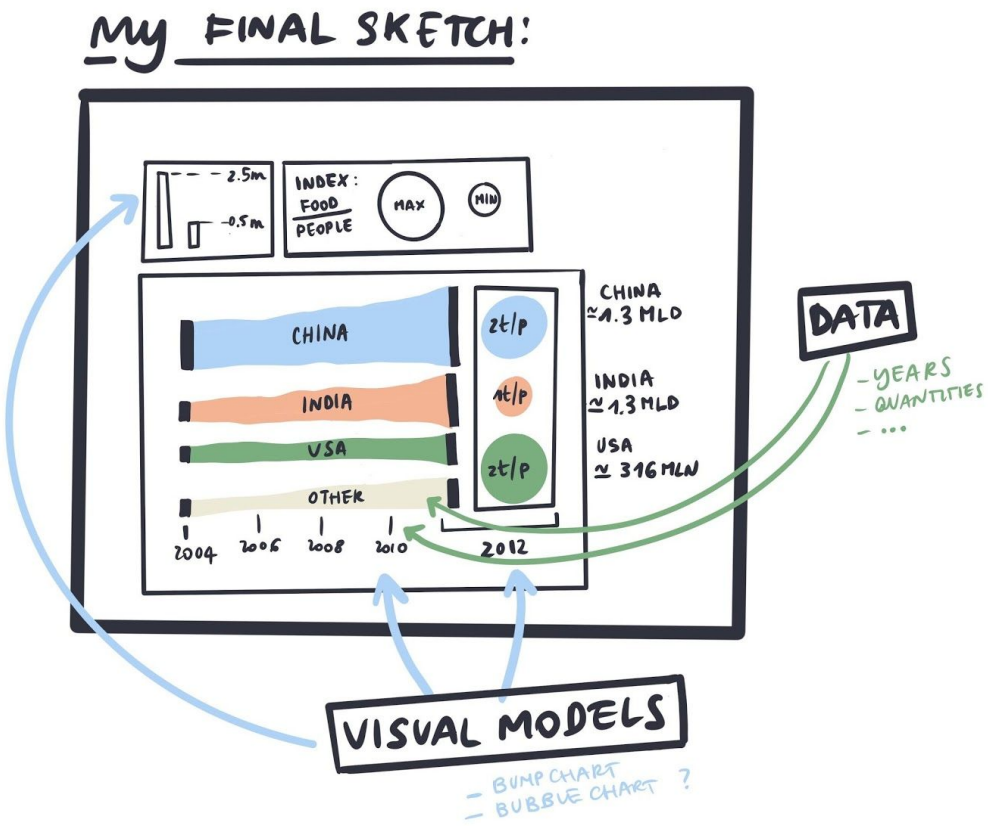


Figure 9: Potential result - visual layout sketch - from Lab 1

3/ Map your data dimensions

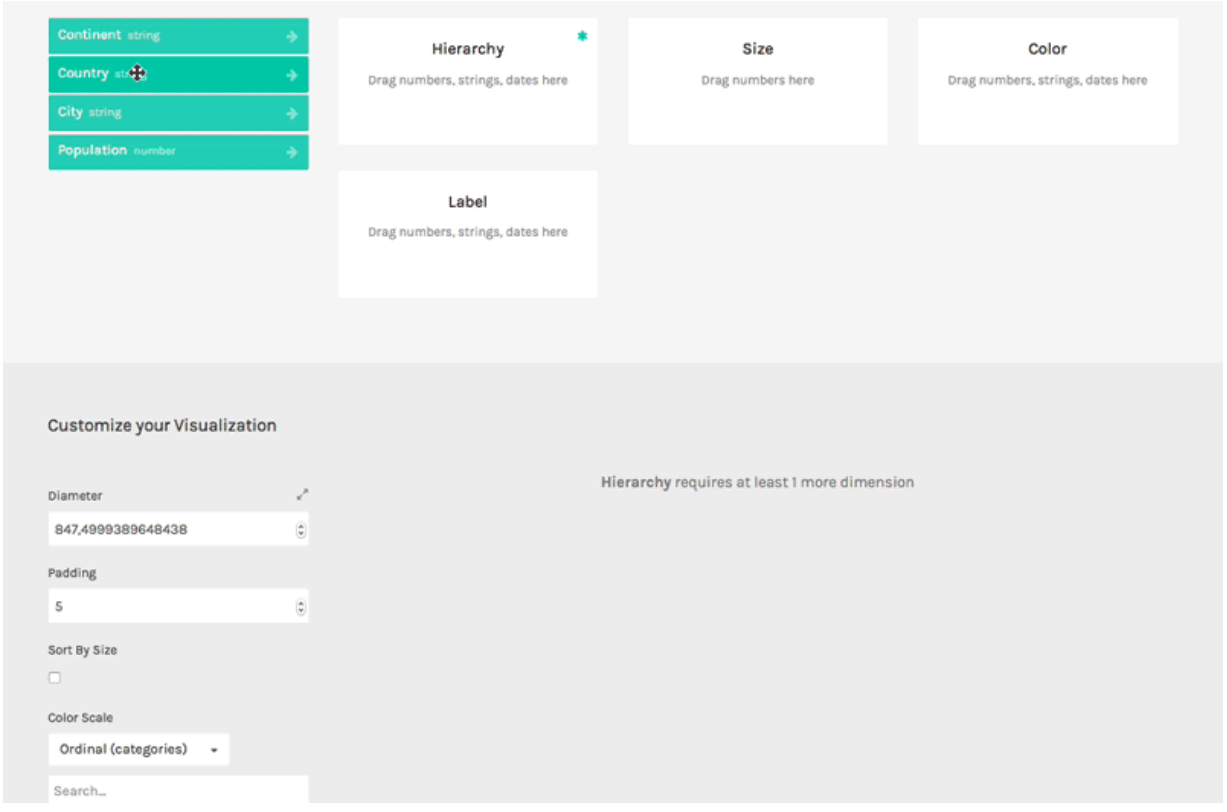


Figure 10: Map data dimensions
click on the image to see animated steps

Map the dimensions of your dataset with the visual variables of the selected layout. As soon as the dataset dimensions are mapped you'll see the visualization.

If something "blocks" you from dropping more than one variable that usually means: no more than single variable exists in the the dataset; you have dropped the wrong variable/s; your data is structured (rows and columns) the wrong way.

Example 1 - Barchart

The barchart layout has four visual variables:

VARIABLE	TYPE	REQUIRED	MULTIPLE	DESCRIPTION
X Axis	string, numbers			For each unique value found in the column, a group (a new bar chart) is created.
Group	string, numbers			For each unique value found in the column, a bar is created.
Size	number			Accepts only columns containing numbers. The value will define the bar height.
Color	string, numbers			Can accept both number and strings. A color will be defined for each unique value found in the list.

It's time to map your data and create the visualization. On the left hand side of your dataset you have the **list of dimensions**, and on the right-hand side you will find available **visual variables** for the barchart model. In order to proceed, follow the next steps:

- Drag and drop the dimensions
- Drag Letter into X Axis
- Drag Frequency into Height
- Drag Language into Groups
- Drag Rank into Color

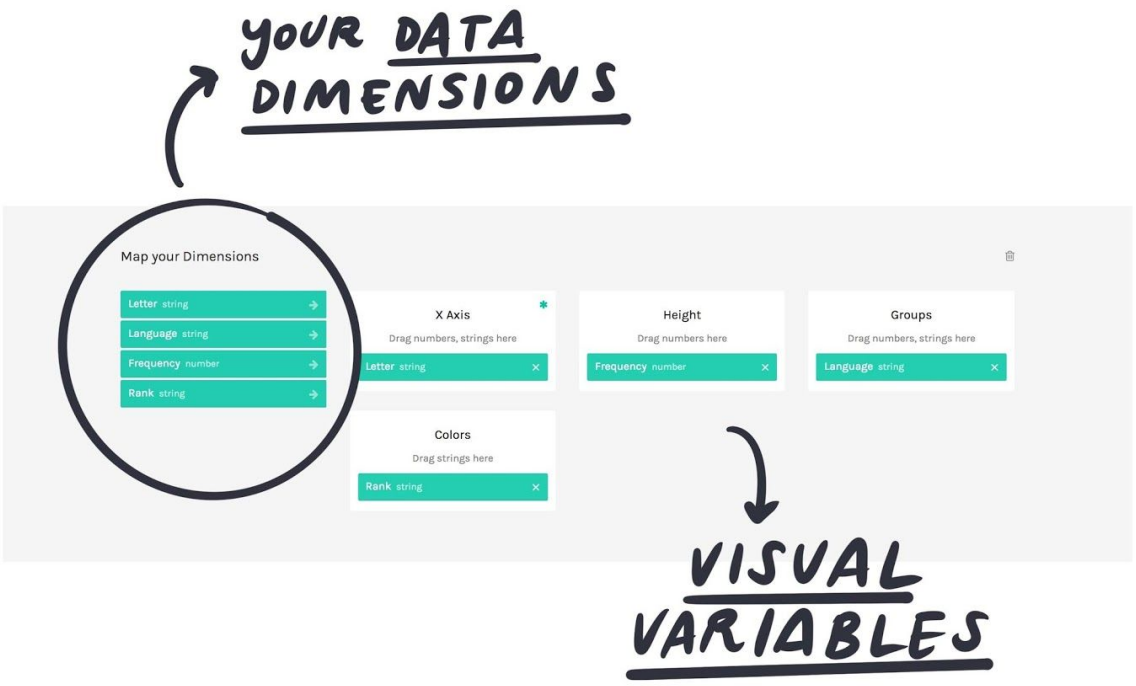


Figure 10: Data dimensions and visual variables

This way you can create a bar for each letter (mapping it on the X Axis). Than, we will divide the visualization in three barcharts, one per language, mapping languages as Groups.

The next step defines the height of the bars, mapping the frequency of each letter. Finally, we will generate one color for each unique value, that can be found in the Rank column.

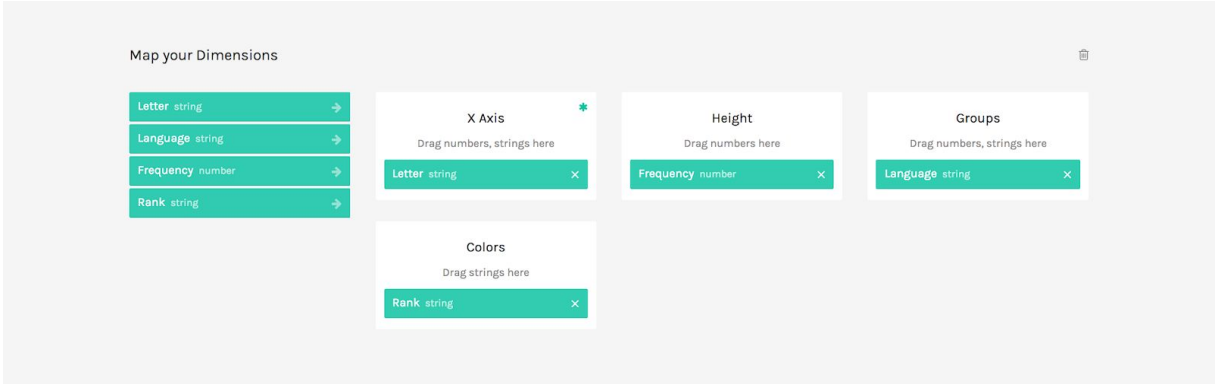


Figure 11: Mapping dimensions

Example 2 - Alluvial Diagram

The alluvial diagram has two visual variables.

VARIABLE	TYPE	REQUIRED	MULTIPLE	DESCRIPTION
Steps	String, numbers			It accept multiple values, at least two column must be selected. Each dragged column will define a step (a vertical group of nodes). The dragging order is also the visualization order. For each unique value found in each column a node will be created.
Size	Numbers			Defines the weight of each line of the dataset. If not defined, all the lines will have the same weight.

Let us map the data and create the visualization. On the left-hand side you will see the list of dimensions of your dataset, and on the right-hand the available visual variable of the bar chart.

Drag and drop the dimensions as following:

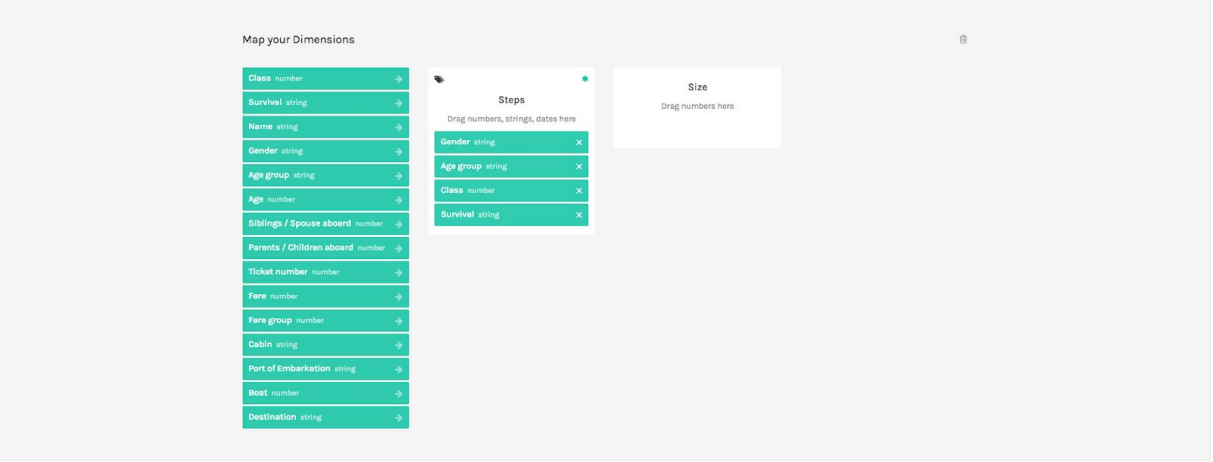


Figure 12: Mapping dimensions

This way you will accomplish all 4 steps as explained before.

Exercise - Choose your visual model

Once you have processed your dataset and decided what visual models you will be using, copy/paste the selected data, and choose the final visual model. Than, drag

and drop the data dimensions in the corresponding variables and create your visualization.

If you don't like the final result, try to switch between different models, playing until you obtain the desirable result. The purpose of this step is to reproduce the visualization you sketched, combining data with exact proportions.

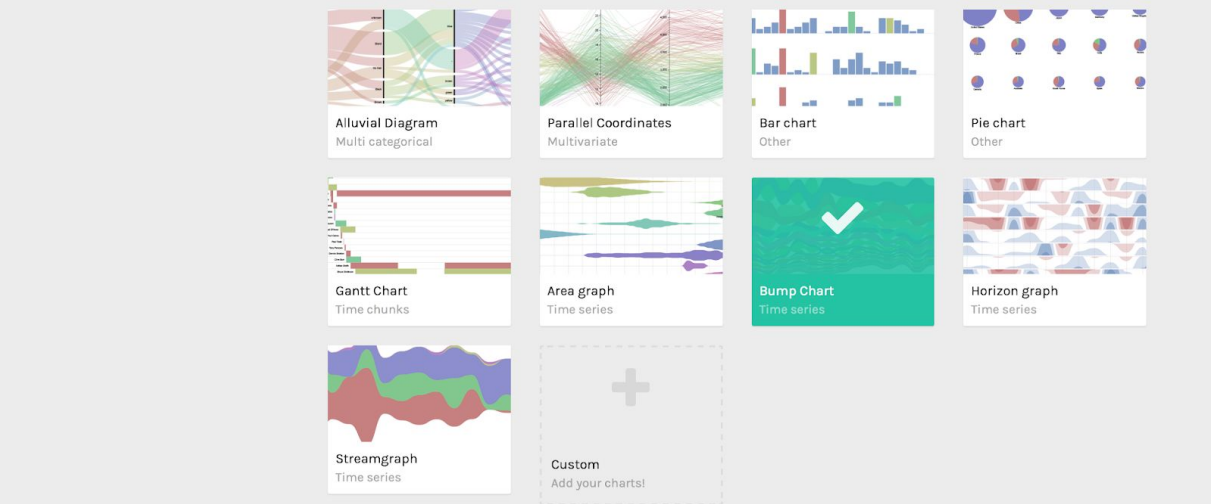


Figure 13: Visual models library

4/ Refine and export

For the purpose of refining, customize the visualization with some of the below listed parameters. Take a look at the following section for more details.

Example 1 - Barchart

The bar chart the parameters:

OPTION	DESCRIPTION
Width	artboard width in pixels
Height	artboard height in pixels
Vertical Padding	distance among bar charts, in pixel
Horizontal Padding	distance between bars, in percentage of the size of the bar (0 = 0%, 1 = 100%)

Use Same Scale	If set, every barchart element will have the same scale
Colour Scale	list of uniques values in the dimension mapped as “color”. If set to ordinal, you can set a color for each value. If set to linear, the app will try to find the minimum and maximum value contained in the dimension, and then creating a gradient among those two values

You can export visualizations as:

- **vector** (**SVG**) - You can open the SVG with *Illustrator* or other vector graphics editors and customize the visualization.
- **raster** (**PNG**) images

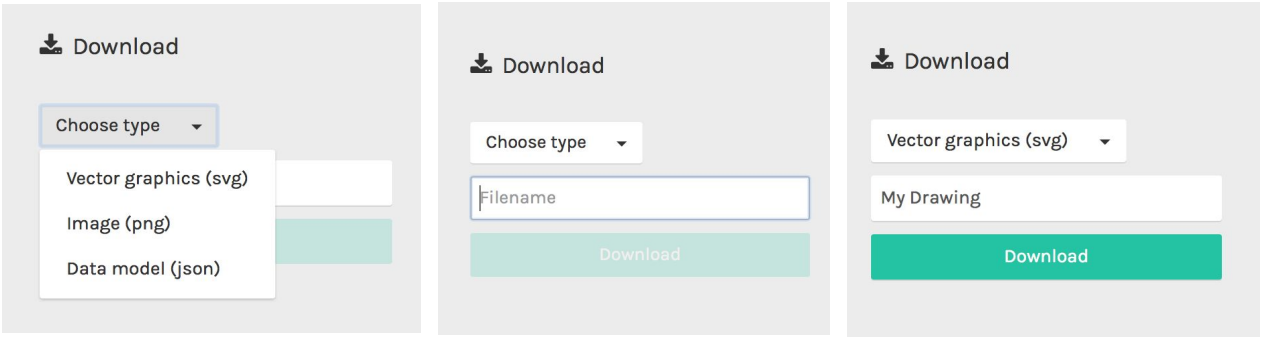


Figure 14: Download visualization

- **embed** visualization into your web page:

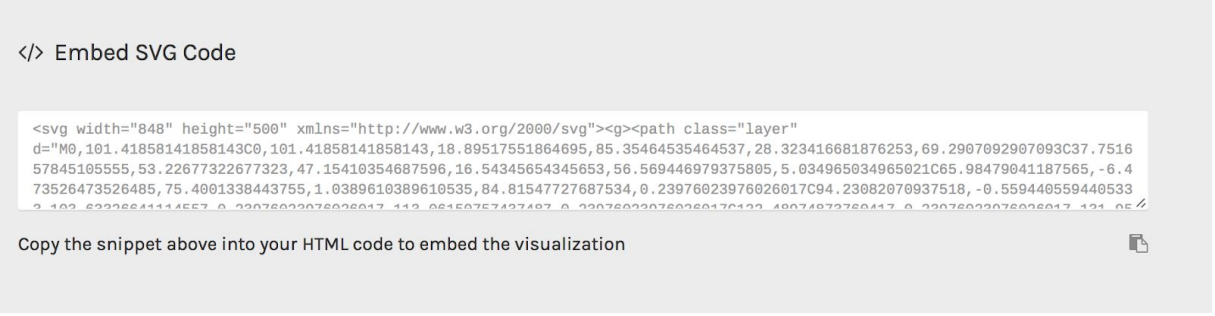


Figure 15: Embed visualization into webpage

Exercise - Choose your visual model

Compose your model, play with the different parameters, and customize the final visualization. Once you are done, download your work as SVG or PNG file.

If you have designed more than one visual model in your sketched draft, create all of them using *RawGraphs*. Pay attention to maintain visual consistency between different variables (e.g if you choose to associate colors to countries, always use the same color code).

Experiment with creating at least two visualizations for each typology you selected.

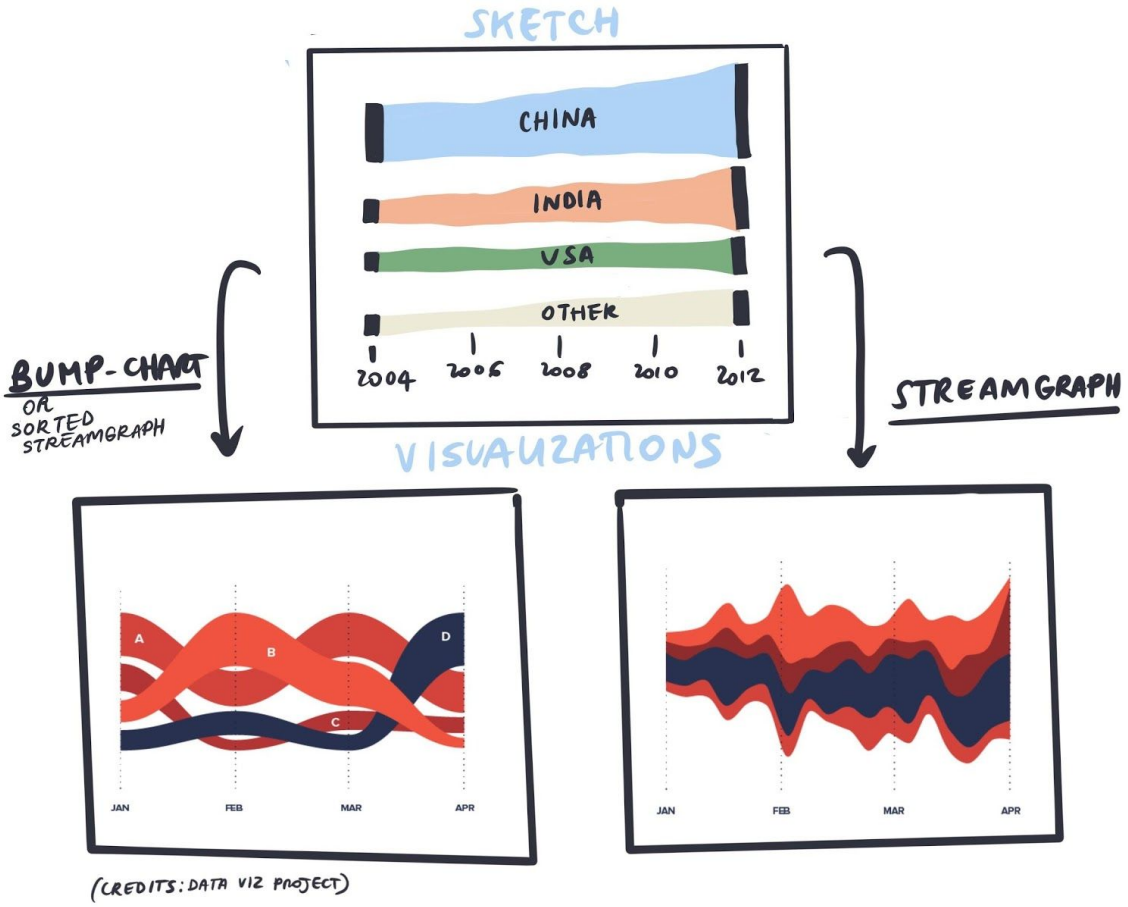


Figure 16: Generate different visual models based on the same sketch

Credits and Additional Resources

01.
RawGraphs
RawGraphs web app
Url: here

02.
RawGraphs GitHub
RawGraphs github
Url: here

03.
RawGraphs tutorials
RawGraphs tutorials at rawgraphs.io
Url: here