Foundations
Tools for Data Visualization

DataVis 2020
http://datavis2020.github.io
Dr. Benjamin Bach

THE UNIVERSITY *of* EDINBURGH


Power BI


Gephi


polestar


Data Illustrator


Graphy


THE VISTORIAN
(BETA)


RAWGraphs


mapbox


Data-Driven Documents

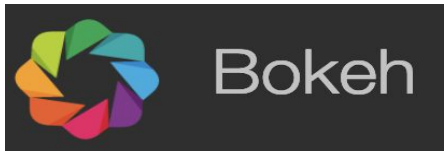

Leaflet


Processing


i want hue


Charticulator


+ableau


Bokeh


plotly


COLORBREWER 2.0
color advice for cartography

Data

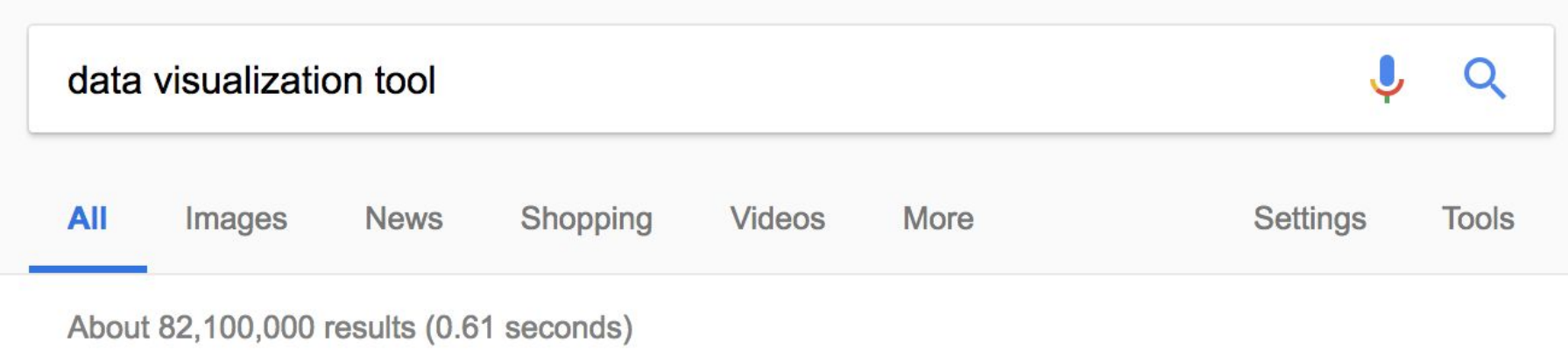
Creation

Exploration

Presentation

Visualization

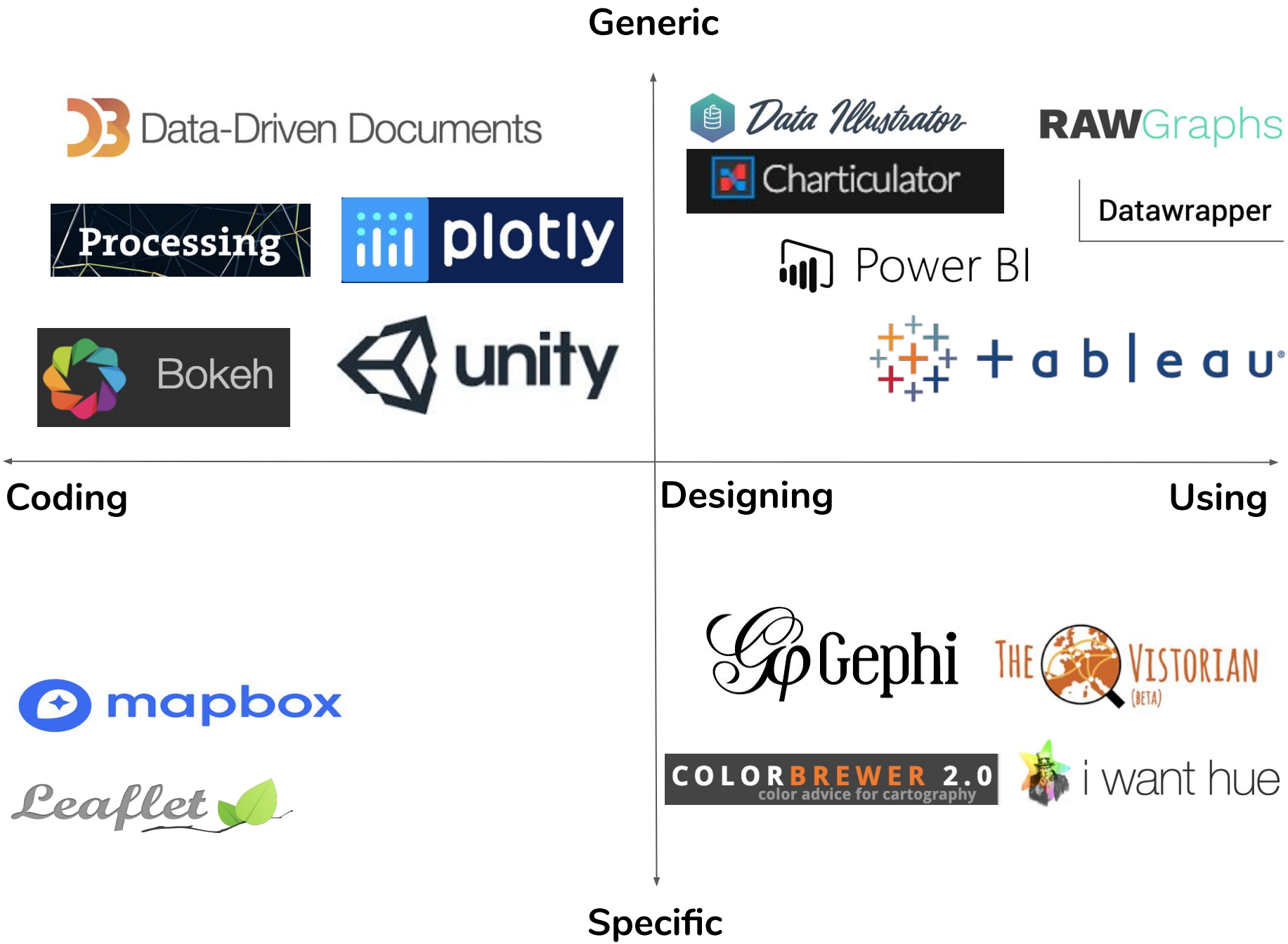
Data Visualization Tools



- Visualizations
- Accessibility
- Learnability
- Data format(s)
- Aesthetics
- Interaction, publication, recycling,...
- Performance
- Programming skills

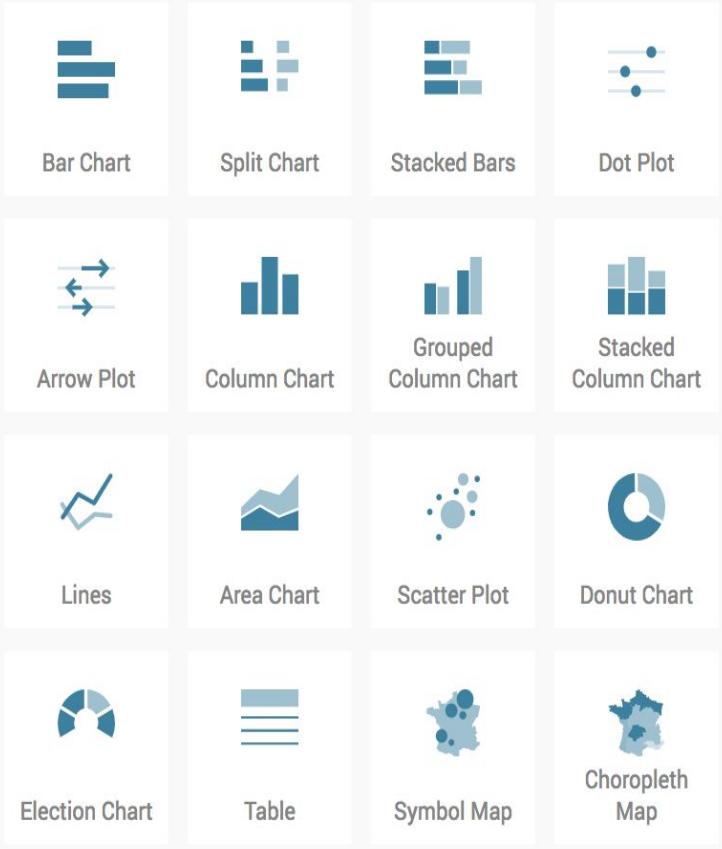
Choosing a tool?

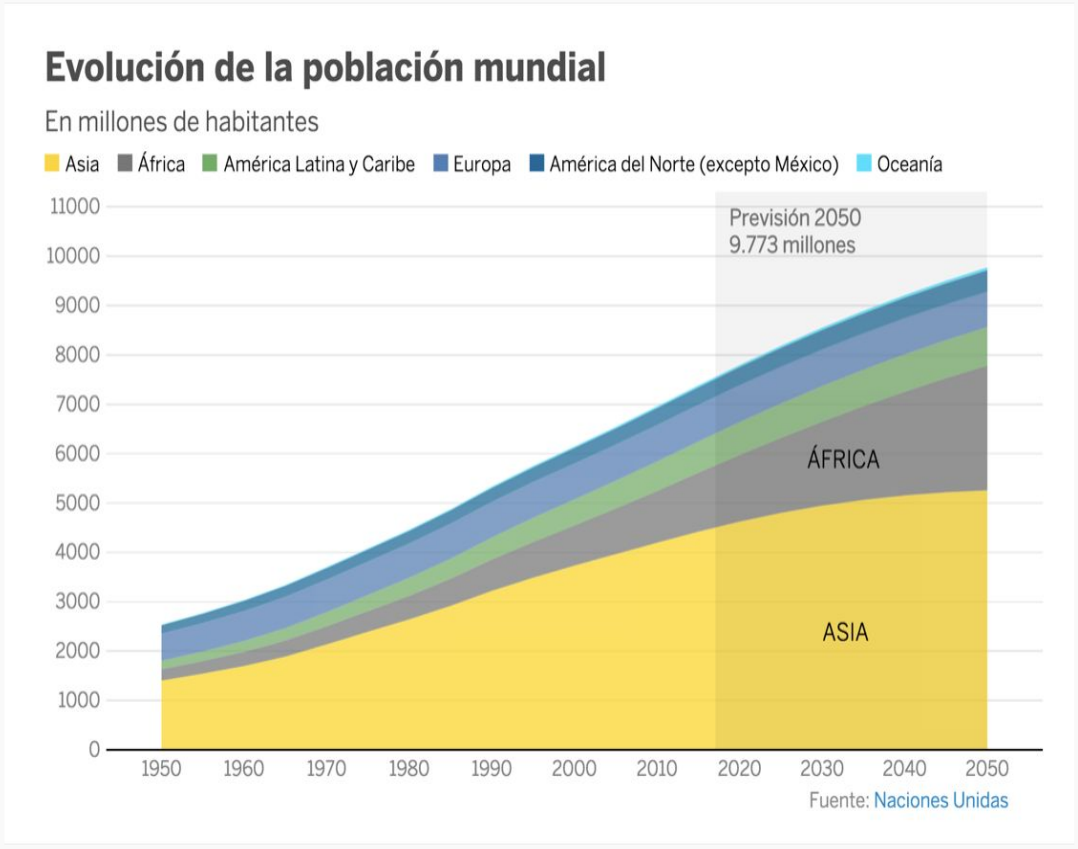
- **What** can the tool do?
- How much do **I** have to do?


Generic
Data-Driven Documents
Data Illustrator
RAWGraphs
Charticulator
Datawrapper
Processing
plotly
Power BI
Bokeh
unity
tableau
Coding
Designing
Using
mapbox
Gephi
THE VISTORIAN (BETA)
COLORBREWER 2.0
color advice for cartography
i want hue
Leaflet
Specific

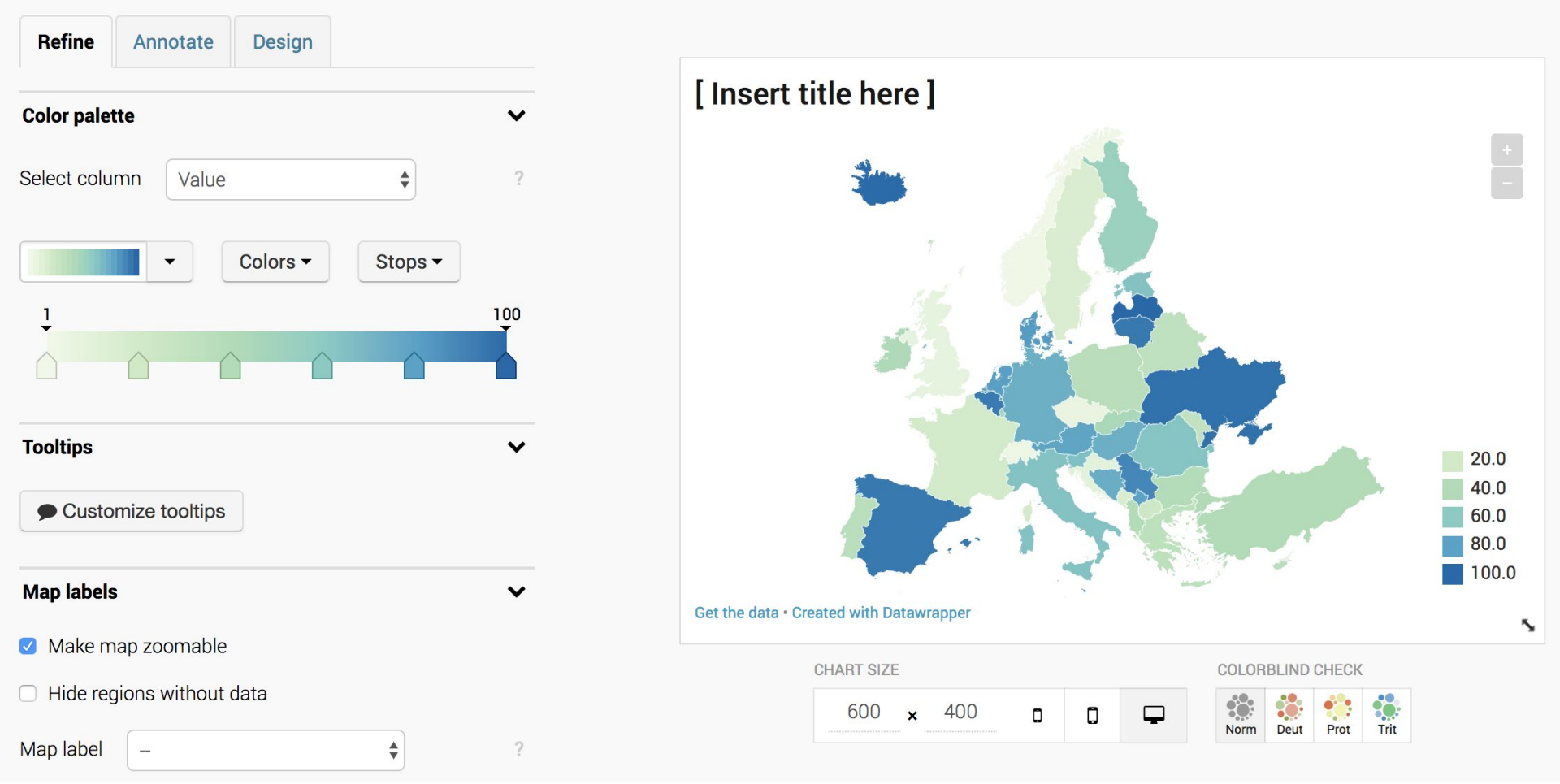
DataWrapper

https://www.datawrapper.de/





DataWrapper

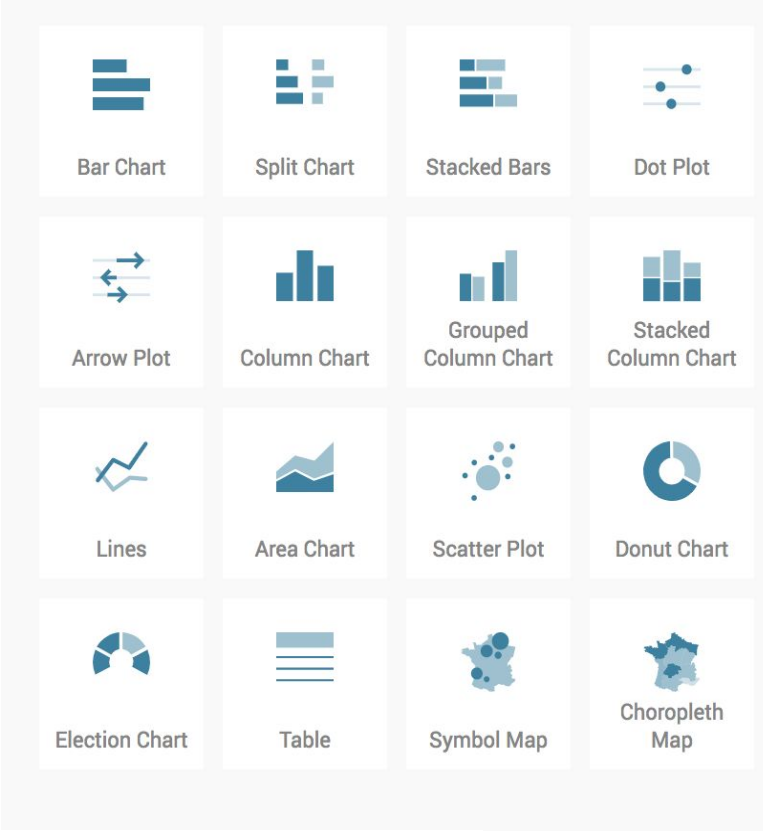


DataWrapper

https://www.datawrapper.de/

DataWrapper

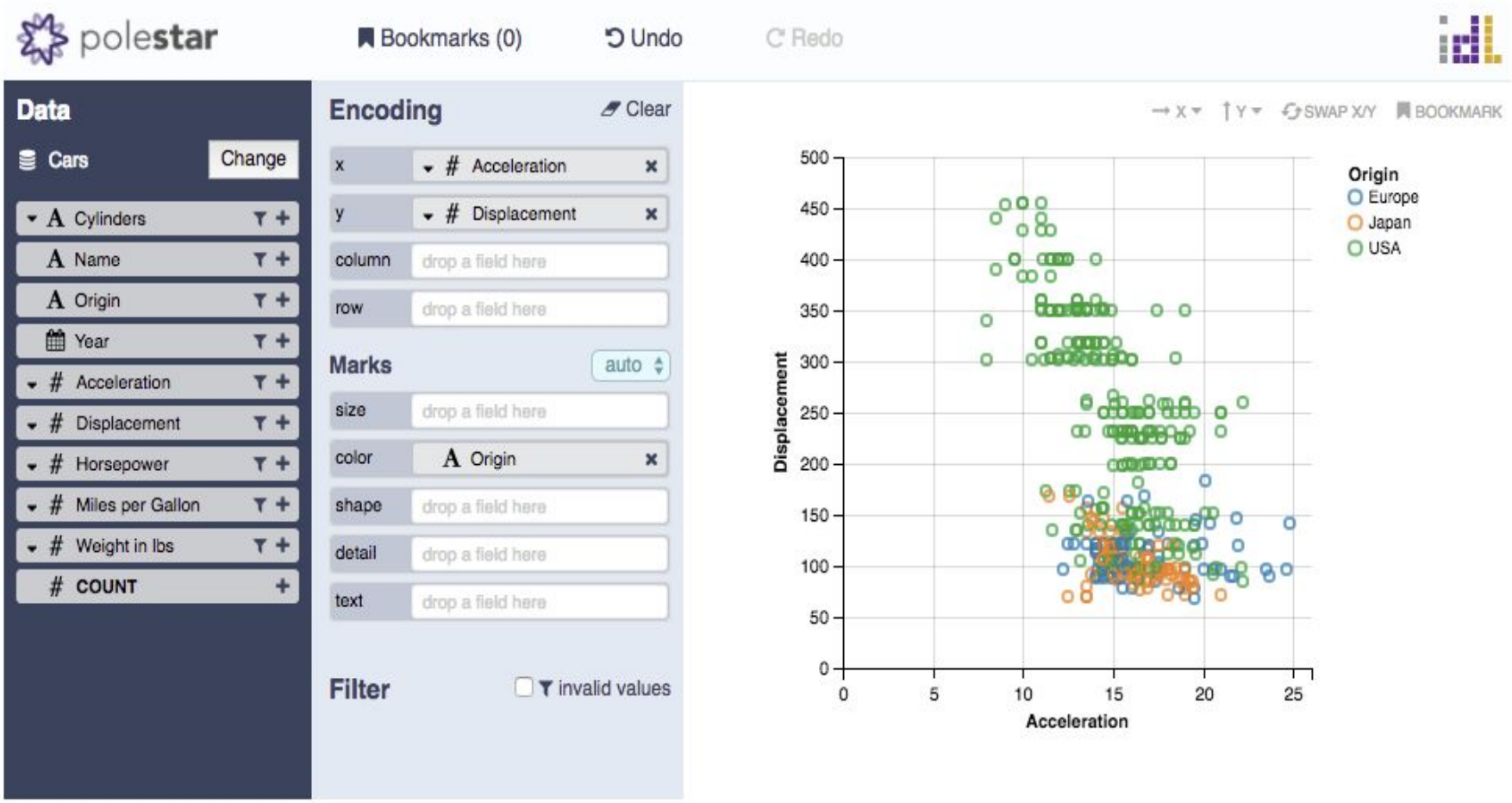
- Good choice of standard graphics
- Maps
- Simple styling and coloring options
- Easily embed graphic
- But: only trial version!



	Basic free	**Single** 29€/month	**Team** 99€/month	**Custom** 499€/month	**Enterprise** Contact us
User type	Just want to try	Use daily	Professional use in a team	Full solution with custom branding	

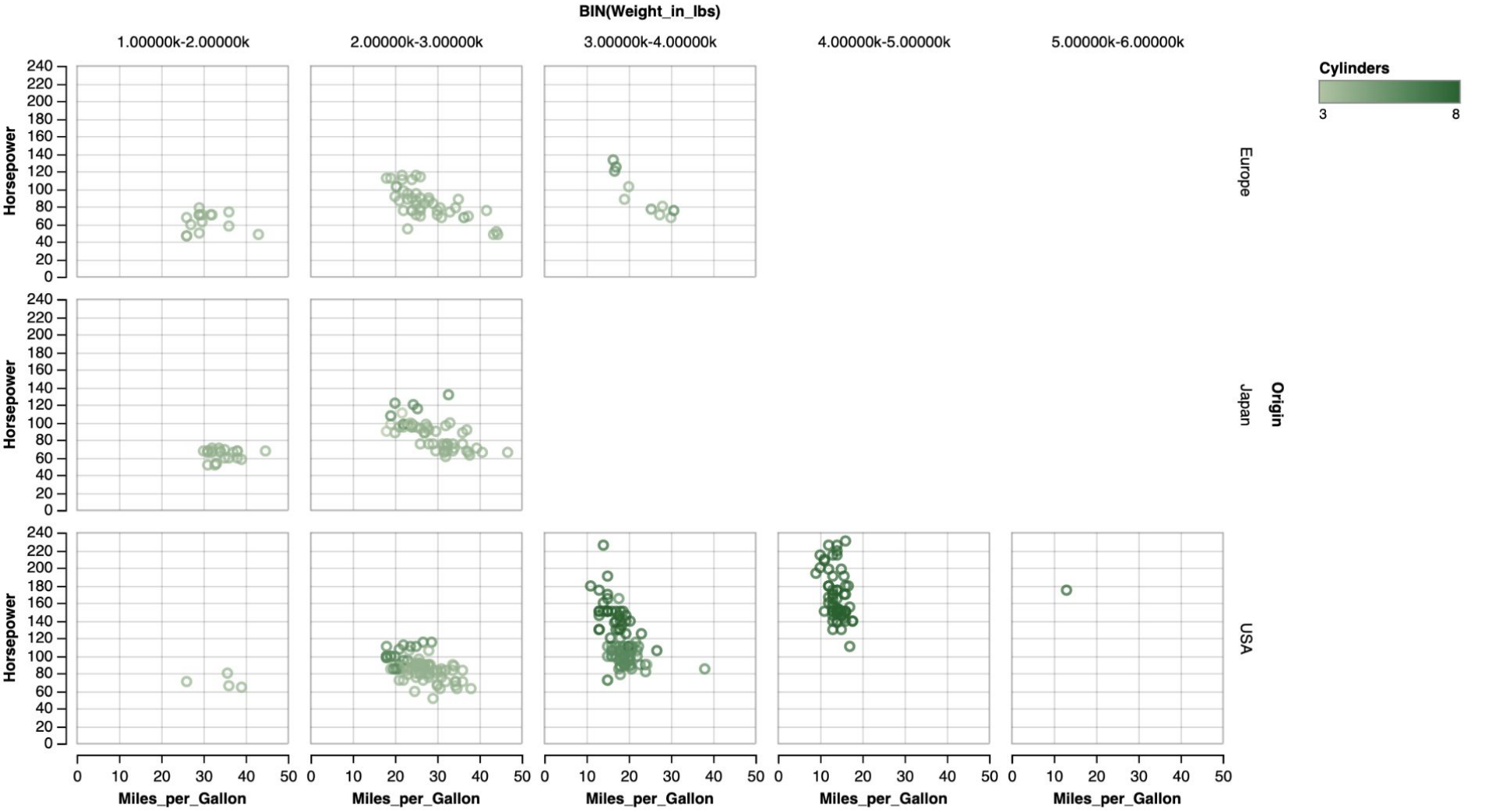
PoleStar / Vega

https://vega.github.io/polestar



PoleStar / Vega

https://vega.github.io/polestar

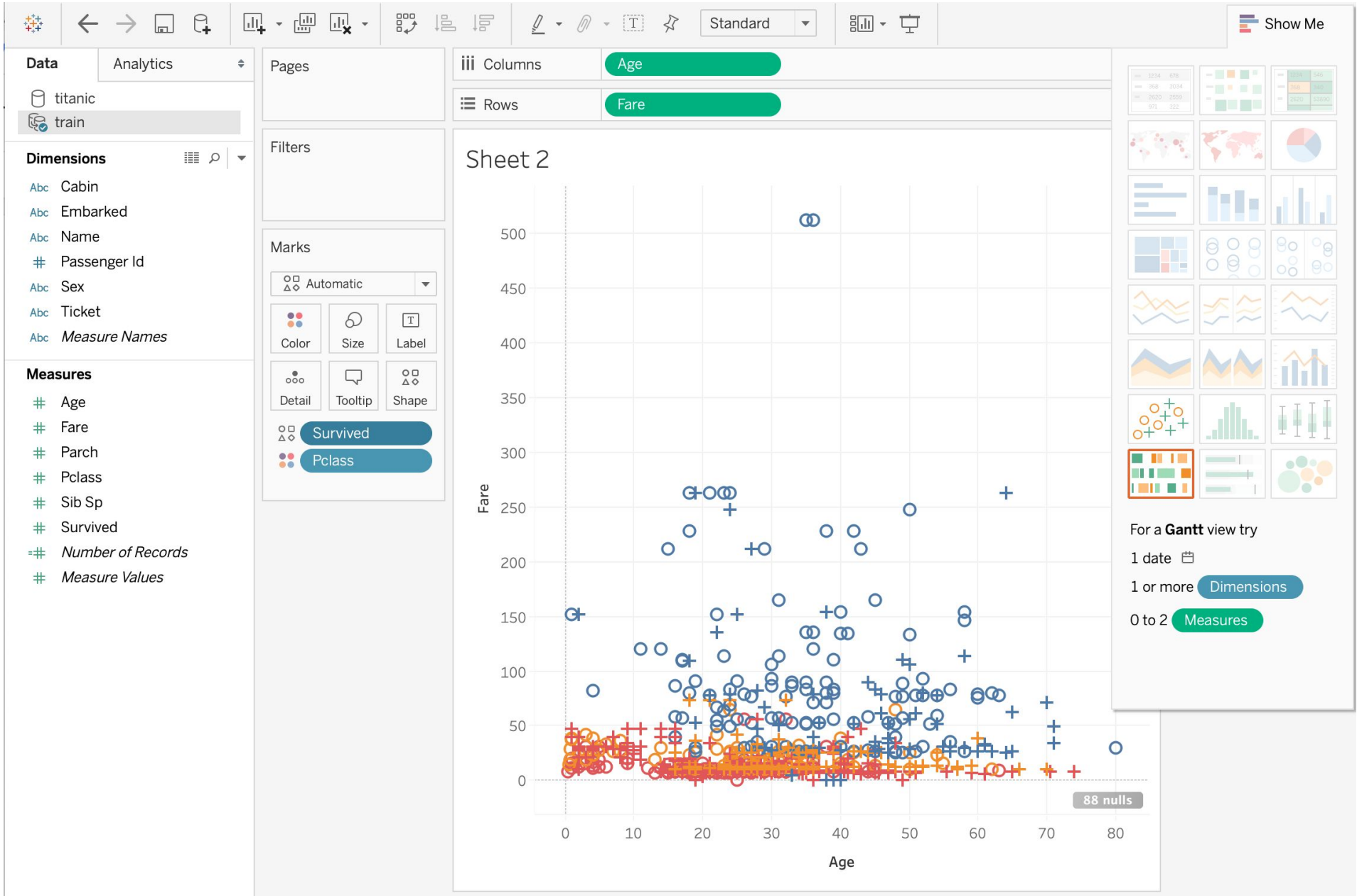


Polestar Vega

- Multivariate data
- Very easy to use
- Little interaction
- Exports VegaLite specification

Tablau

https://www.tableau.com

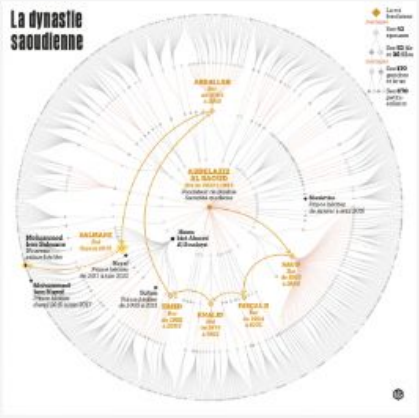


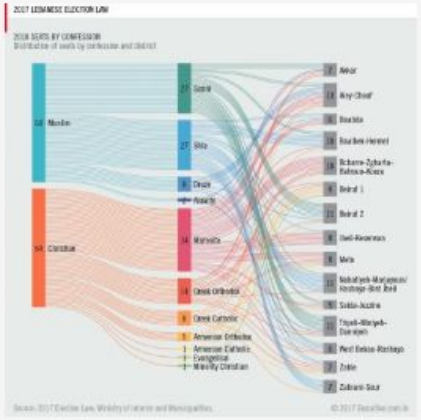
Tableau

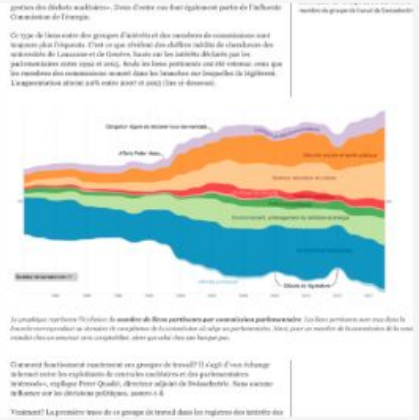
- Simple drag and drop
- Many functions (too many?)
- Many standard charts like DataWrapper
- Video tutorials
- Free trail and online version
- Targeted towards businesses

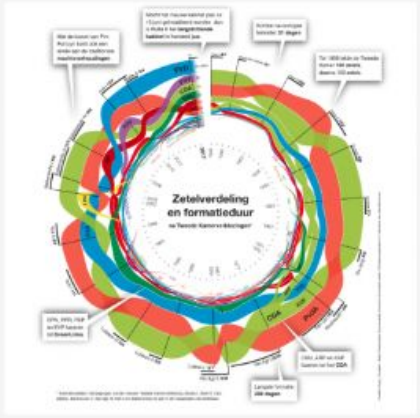
RawGraphs

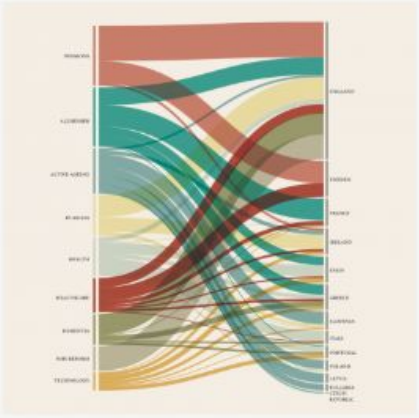
https://www.rawgraphs.io

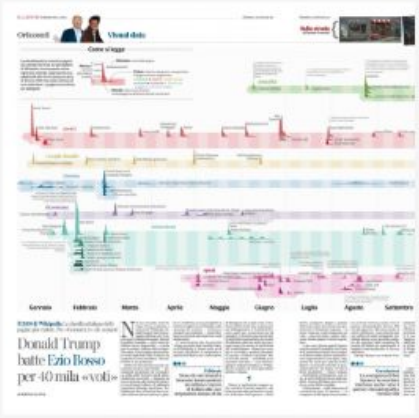


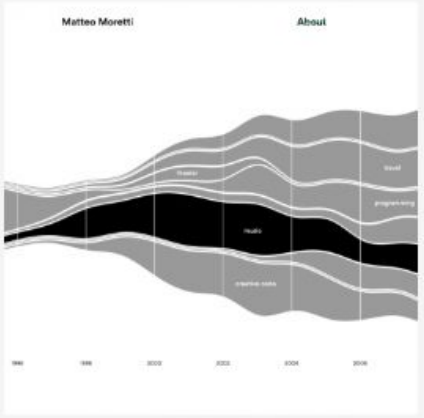


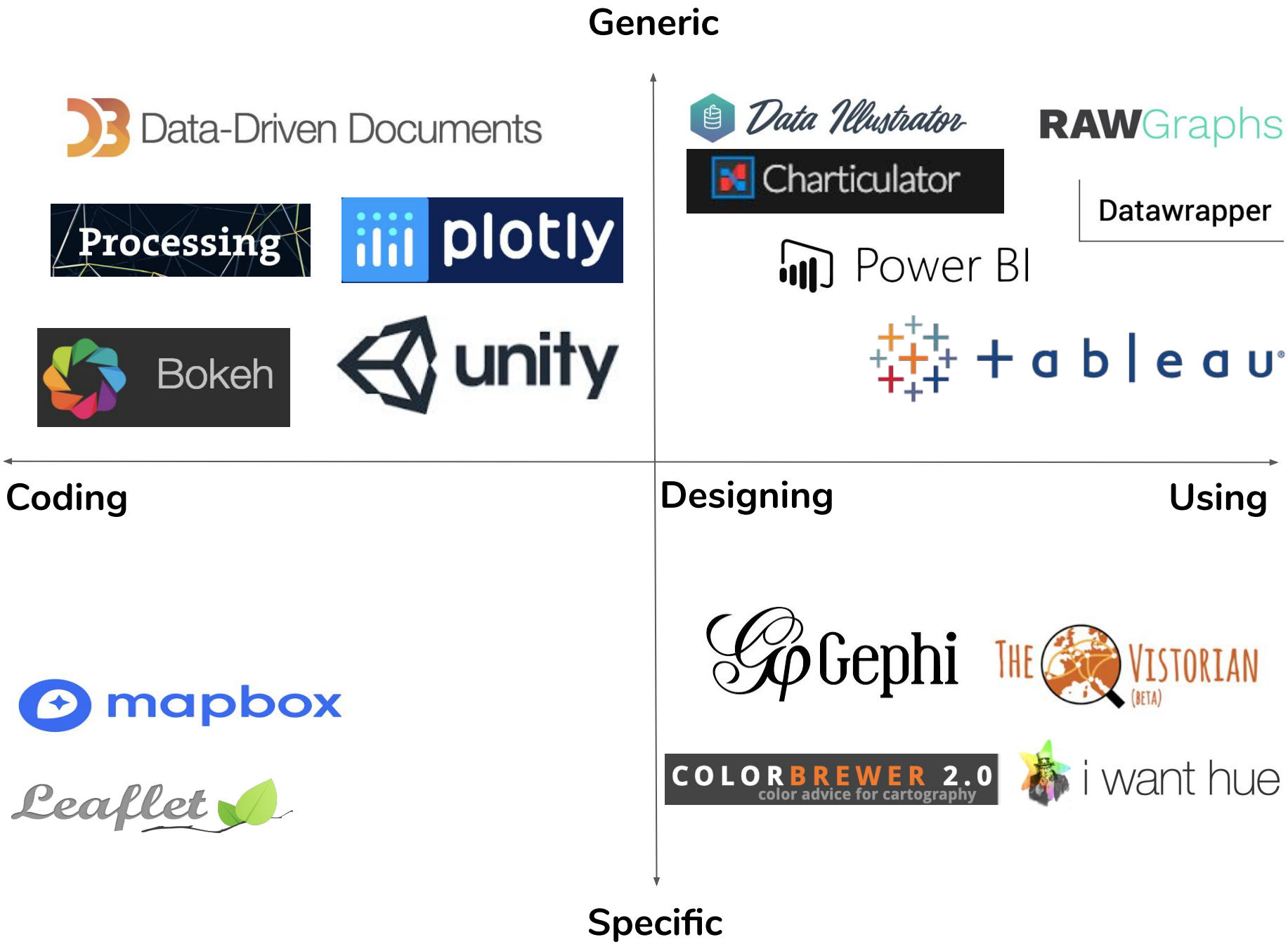




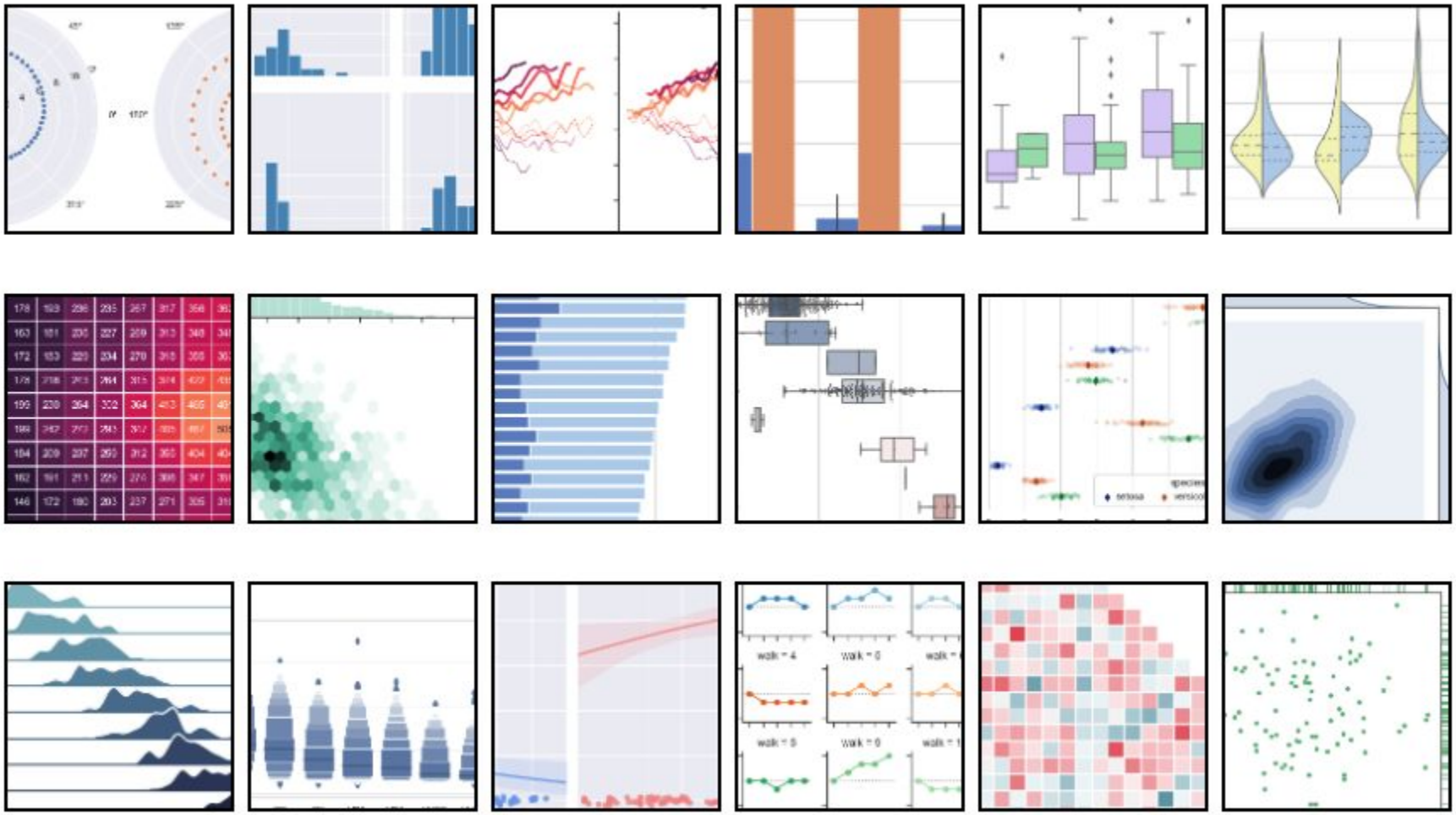







Generic
Data-Driven Documents
Data Illustrator
RAWGraphs
Charticulator
Datawrapper
Processing
plotly
Power BI
Bokeh
unity
tableau
Coding
Designing
Using
mapbox
Gephi
THE VISTORIAN (BETA)
COLORBREWER 2.0
color advice for cartography
i want hue
Leaflet
Specific

Python: **Seaborn**

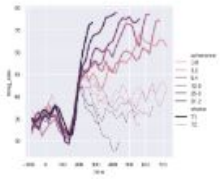


- Static images
- Easy to program
- Good choice of charts

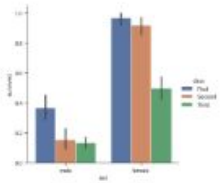
Python: **Seaborn**

Official seaborn tutorial

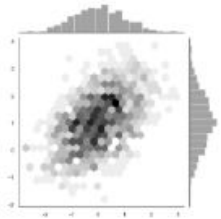
Plotting functions



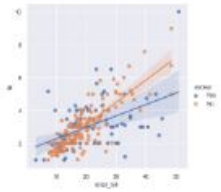
- Visualizing statistical relationships
  - Relating variables with scatter plots
  - Emphasizing continuity with line plots
  - Showing multiple relationships with facets



- Plotting with categorical data
  - Categorical scatterplots
  - Distributions of observations within categories
  - Statistical estimation within categories
  - Plotting "wide-form" data
  - Showing multiple relationships with facets

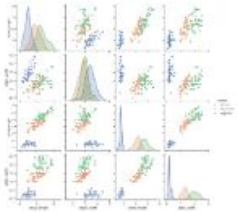


- Visualizing the distribution of a dataset
  - Plotting univariate distributions
  - Plotting bivariate distributions
  - Visualizing pairwise relationships in a dataset



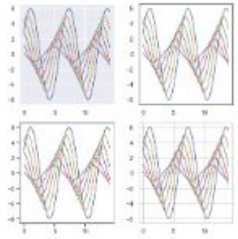
- Visualizing linear relationships
  - Functions to draw linear regression models
  - Fitting different kinds of models
  - Conditioning on other variables
  - Controlling the size and shape of the plot
  - Plotting a regression in other contexts

Multi-plot grids

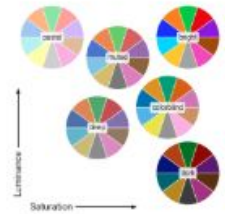


- Building structured multi-plot grids
  - Conditional small multiples
  - Using custom functions
  - Plotting pairwise data relationships

Plot aesthetics

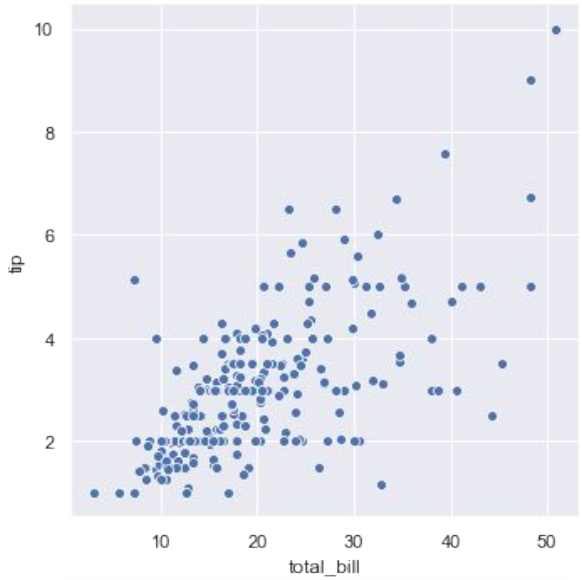


- Controlling figure aesthetics
  - Seaborn figure styles
  - Removing axes spines
  - Temporarily setting figure style
  - Overriding elements of the seaborn styles
  - Scaling plot elements

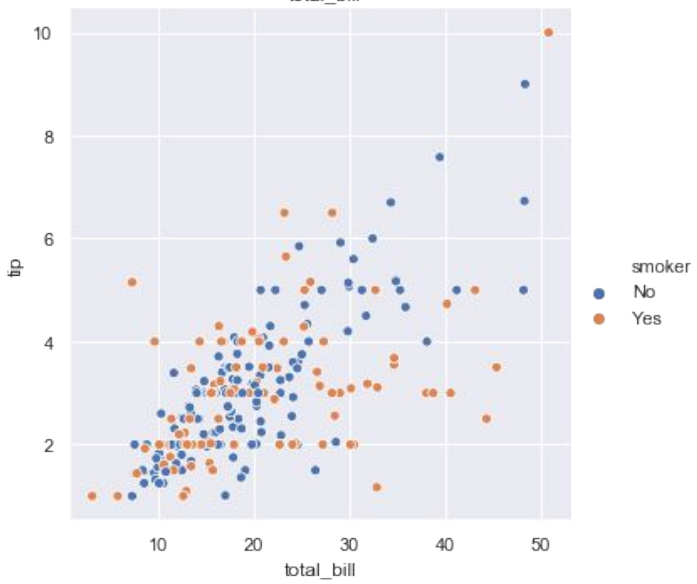


- Choosing color palettes
  - Building color palettes
  - Qualitative color palettes
  - Sequential color palettes
  - Diverging color palettes
  - Setting the default color palette

Python: **Seaborn**




```
tips = sns.load_dataset("tips")
sns.relplot(x="total_bill", y="tip", data=tips)
```

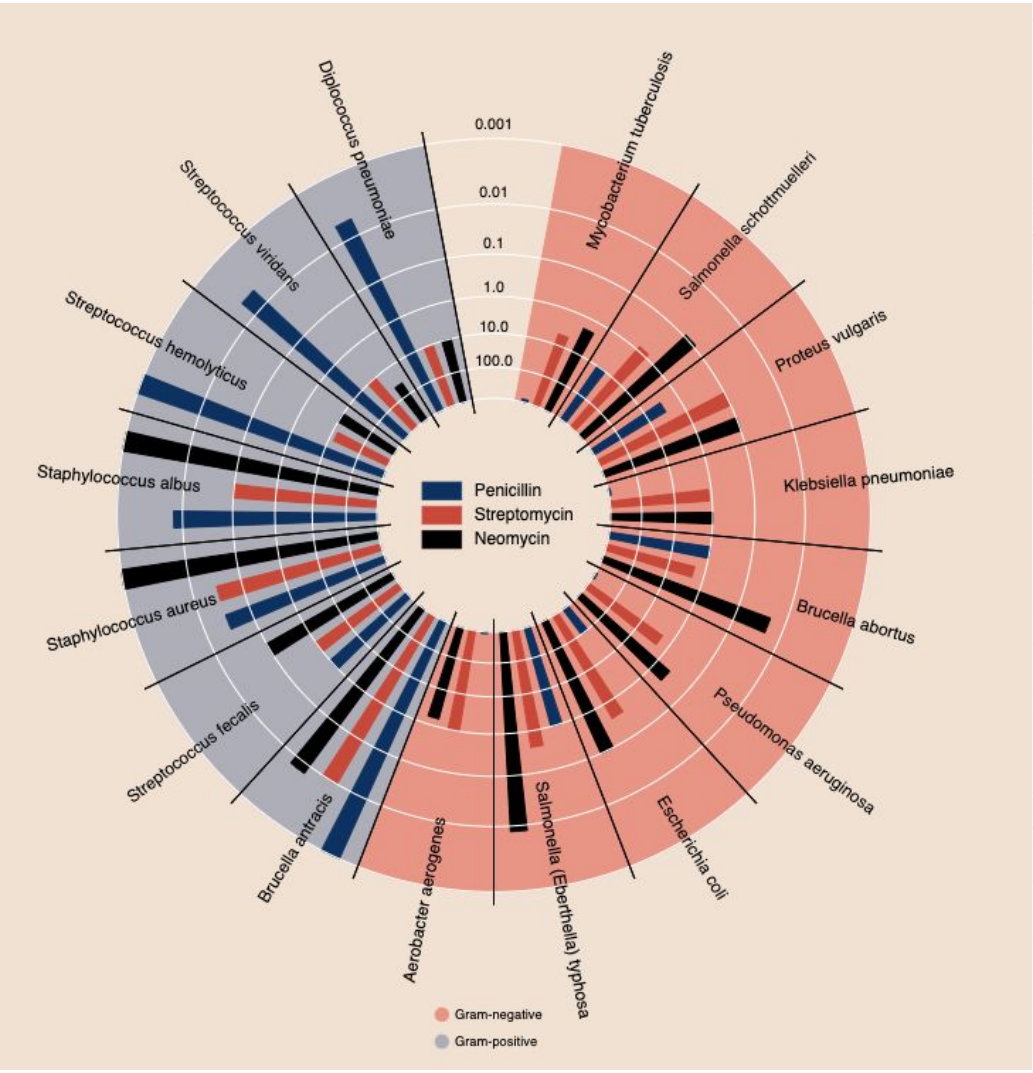



```
sns.relplot(x="total_bill", y="tip",
, y="tip", hue="smoker", data=tips);
```

Python: **Bokeh**

- Interactive graphics
- Produces JavaScript and runs in browsers
- Low-level control > sophistication
- Interactive applications

Python: **Bokeh**




```
minr = sqrt(log(.001 * 1E4))
maxr = sqrt(log(1000 * 1E4))
a = (outer_radius - inner_radius) / (minr - maxr)
b = inner_radius - a * maxr

def rad(mic):
    return a * np.sqrt(np.log(mic * 1E4)) + b

big_angle = 2.0 * np.pi / (len(df) + 1)
small_angle = big_angle / 7

p = figure(plot_width=width, plot_height=height, title="",
    x_axis_type=None, y_axis_type=None,
    x_range=(-420, 420), y_range=(-420, 420),
    min_border=0, outline_line_color="black",
    background_fill_color="#f0e1d2")

p.xgrid.grid_line_color = None
p.ygrid.grid_line_color = None

# annular wedges
angles = np.pi/2 - big_angle/2 - df.index.to_series()*big_angle
colors = [gram_color[gram] for gram in df.gram]
p.annular_wedge(
    0, 0, inner_radius, outer_radius, -big_angle+angles, angles, color=colors,
)

# small wedges
p.annular_wedge(0, 0, inner_radius, rad(df.penicillin),
                -big_angle+angles+5*small_angle, -big_angle+angles+6*small_angle,
                color=drug_color['Penicillin'])
p.annular_wedge(0, 0, inner_radius, rad(df.streptomycin),
                -big_angle+angles+3*small_angle, -big_angle+angles+4*small_angle,
                color=drug_color['Streptomycin'])
p.annular_wedge(0, 0, inner_radius, rad(df.neomycin),
                -big_angle+angles+1*small_angle, -big_angle+angles+2*small_angle,
                color=drug_color['Neomycin'])

# circular axes and lables
labels = np.power(10.0, np.arange(-3, 4))
radii = a * np.sqrt(np.log(labels * 1E4)) + b
p.circle(0, 0, radius=radii, fill_color=None, line_color="white")
p.text(0, radii[:-1], [str(r) for r in labels[:-1]],
       text_font_size="8pt", text_align="center", text_baseline="middle")

# radial axes
p.annular_wedge(0, 0, inner_radius-10, outer_radius+10,
                -big_angle+angles, -big_angle+angles, color="black")

# bacteria labels
xr = radii[0]*np.cos(np.array(-big_angle/2 + angles))
yr = radii[0]*np.sin(np.array(-big_angle/2 + angles))
label_angle=np.array(-big_angle/2+angles)
label_angle[label_angle < -np.pi/2] += np.pi # easier to read labels on the left s
p.text(xr, yr, df.bacteria, angle=label_angle,
       text_font_size="9pt", text_align="center", text_baseline="middle")

# OK, these hand drawn legends are pretty clunky, will be improved in future relea
p.circle([-40, -40], [-370, -390], color=list(gram_color.values()), radius=5)
p.text([-30, -30], [-370, -390], text=["Gram-" + gr for gr in gram_color.keys()],
       text_font_size="7pt", text_align="left", text_baseline="middle")

p.rect([-40, -40, -40], [18, 0, -18], width=30, height=13,
       color=list(drug_color.values()))
p.text([-15, -15, -15], [18, 0, -18], text=list(drug_color),
       text_font_size="9pt", text_align="left", text_baseline="middle")

output_file("burtin.html", title="burtin.py example")

show(p)
```

Python: **Bokeh**

AN INTERACTIVE EXPLORER FOR MOVIE DATA

Interact with the widgets on the left to query a subset of movies to plot. Hover over the circles to see more information about each movie.

Inspired by the **Shiny Movie Explorer**. *(Information from **OMDB**)*

Minimum number of reviews: **80**

Dollars at Box Office (millions): **0**

Genre
All

Year released: **1970**

End Year released: **2014**

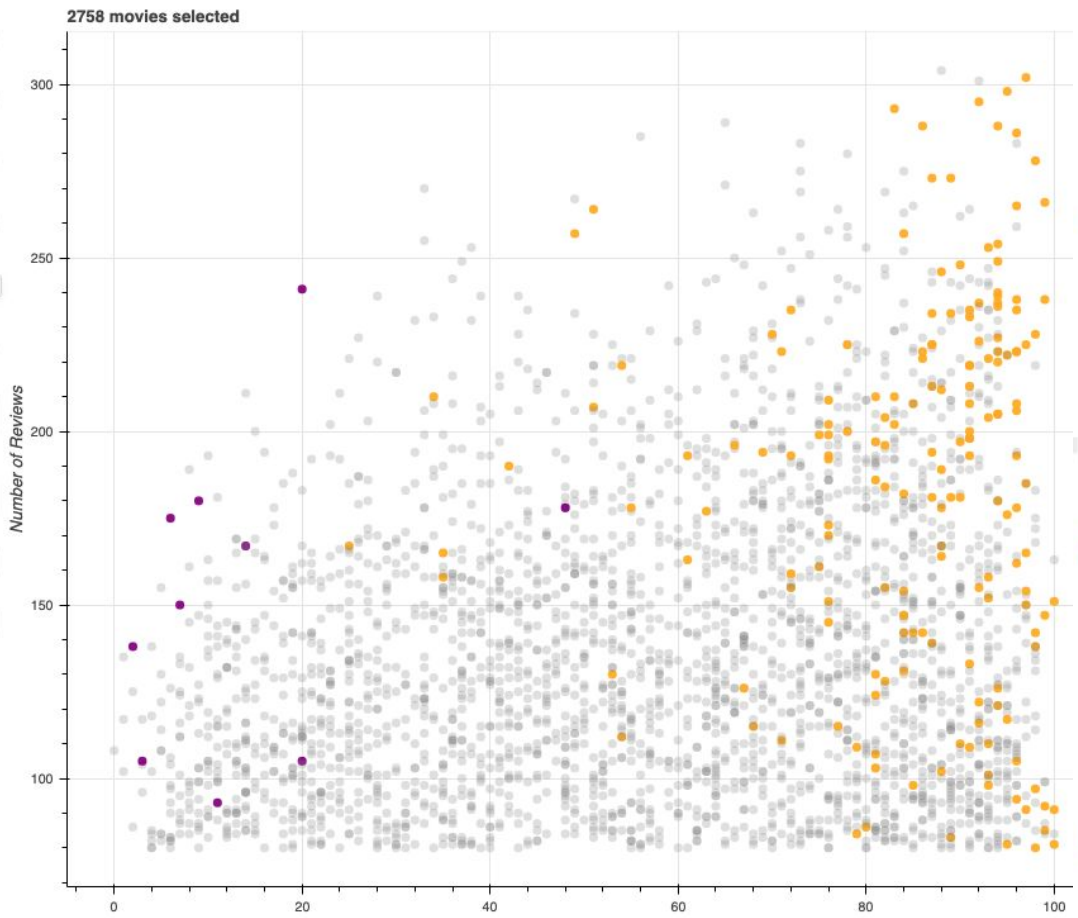
Minimum number of Oscar wins: **0**

Director name contains

Cast names contains

X Axis
Tomato Meter

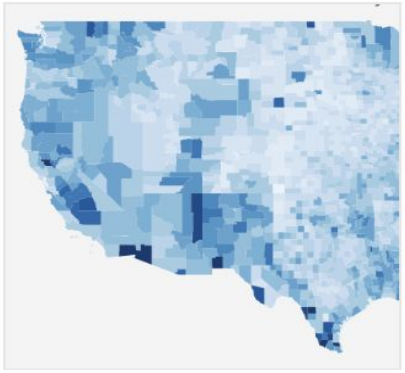
Y Axis
Number of Reviews

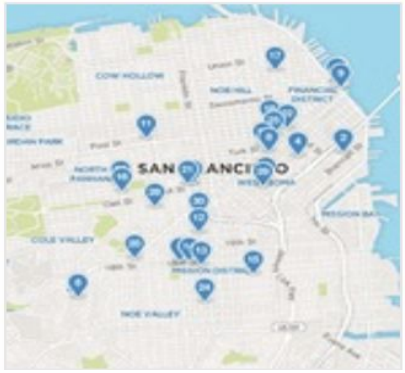
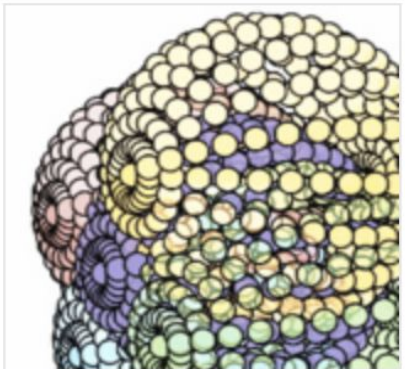
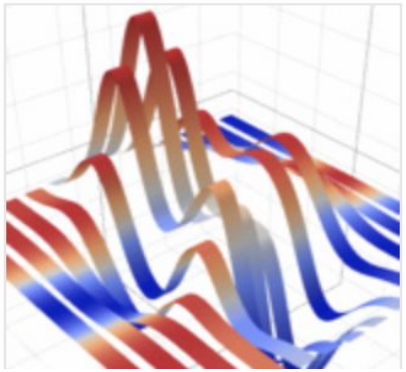
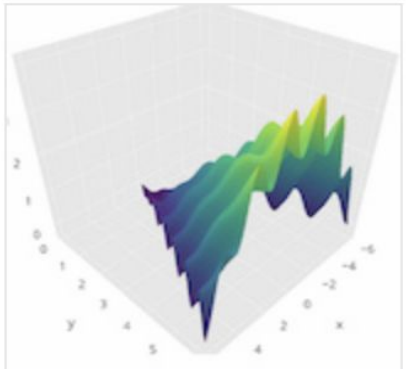
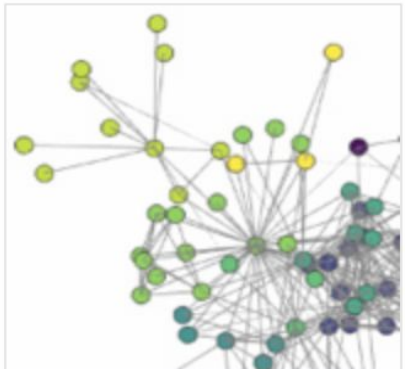
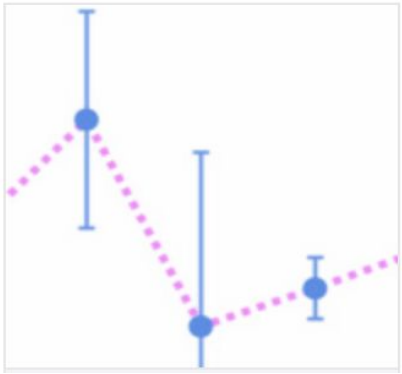
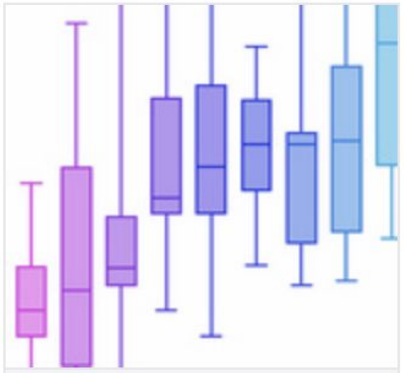



```
axis_map = {
    "Tomato Meter": "Meter",
    "Numeric Rating": "numericRating",
    "Number of Reviews": "Reviews",
    "Box Office (dollars)": "BoxOffice",
    "Length (minutes)": "Runtime",
    "Year": "Year",
}

desc = Div(text=open(join(dirname(__file__), "descrip

# Create Input controls
reviews = Slider(title="Minimum number of reviews", v
min_year = Slider(title="Year released", start=1940,
max_year = Slider(title="End Year released", start=19
oscars = Slider(title="Minimum number of Oscar wins",
boxoffice = Slider(title="Dollars at Box Office (mill
genre = Select(title="Genre", value="All",
               options=open(join(dirname(__file__),
director = TextInput(title="Director name contains")
cast = TextInput(title="Cast names contains")
x_axis = Select(title="X Axis", options=sorted(axis_m
y_axis = Select(title="Y Axis", options=sorted(axis_m

# Create Column Data Source that will be used by the
source = ColumnDataSource(data=dict(x=[], y=[], color

TOOLTIPS=[
    ("Title", "@title"),
    ("Year", "@year"),
    ("$", "@revenue")
]

p = figure(plot_height=600, plot_width=700, title="",
p.circle(x="x", y="y", source=source, size=7, color="


def select_movies():
    genre_val = genre.value
    director_val = director.value.strip()
    cast_val = cast.value.strip()
    selected = movies[
        (movies.Reviews >= reviews.value) &
        (movies.BoxOffice >= (boxoffice.value * 1e6)) &
        (movies.Year >= min_year.value) &
        (movies.Year <= max_year.value) &
        (movies.Oscars >= oscars.value)
    ]
    if (genre_val != "All"):
        selected = selected[selected.Genre.str.contai
    if (director_val != ""):
        selected = selected[selected.Director.str.con
    if (cast_val != ""):
        selected = selected[selected.Cast.str.contain
    return selected


def update():
    df = select_movies()
    x_name = axis_map[x_axis.value]
    y_name = axis_map[y_axis.value]

    p.xaxis.axis_label = x_axis.value
    p.yaxis.axis_label = y_axis.value
    p.title.text = "%d movies selected" % len(df)
    source.data = dict(
        x=df[x_name],
        y=df[y_name],
        color=df["color"],
        title=df["Title"],
        year=df["Year"],
        revenue=df["revenue"],
        alpha=df["alpha"],
    )

controls = [reviews, boxoffice, genre, min_year, max_
for control in controls:
    control.on_change('value', lambda attr, old, new:

inputs = column(*controls, width=320, height=1000)
inputs.sizing_mode = "fixed"
l = layout([
    [desc],
    [inputs, p],
], sizing_mode="scale_both")

update()  # initial load of the data

curdoc().add_root(l)
curdoc().title = "Movies"
```

https://demo.bokeh.org/movies

Plotly (Python, R, MatLab, Perl..) > JavaScript

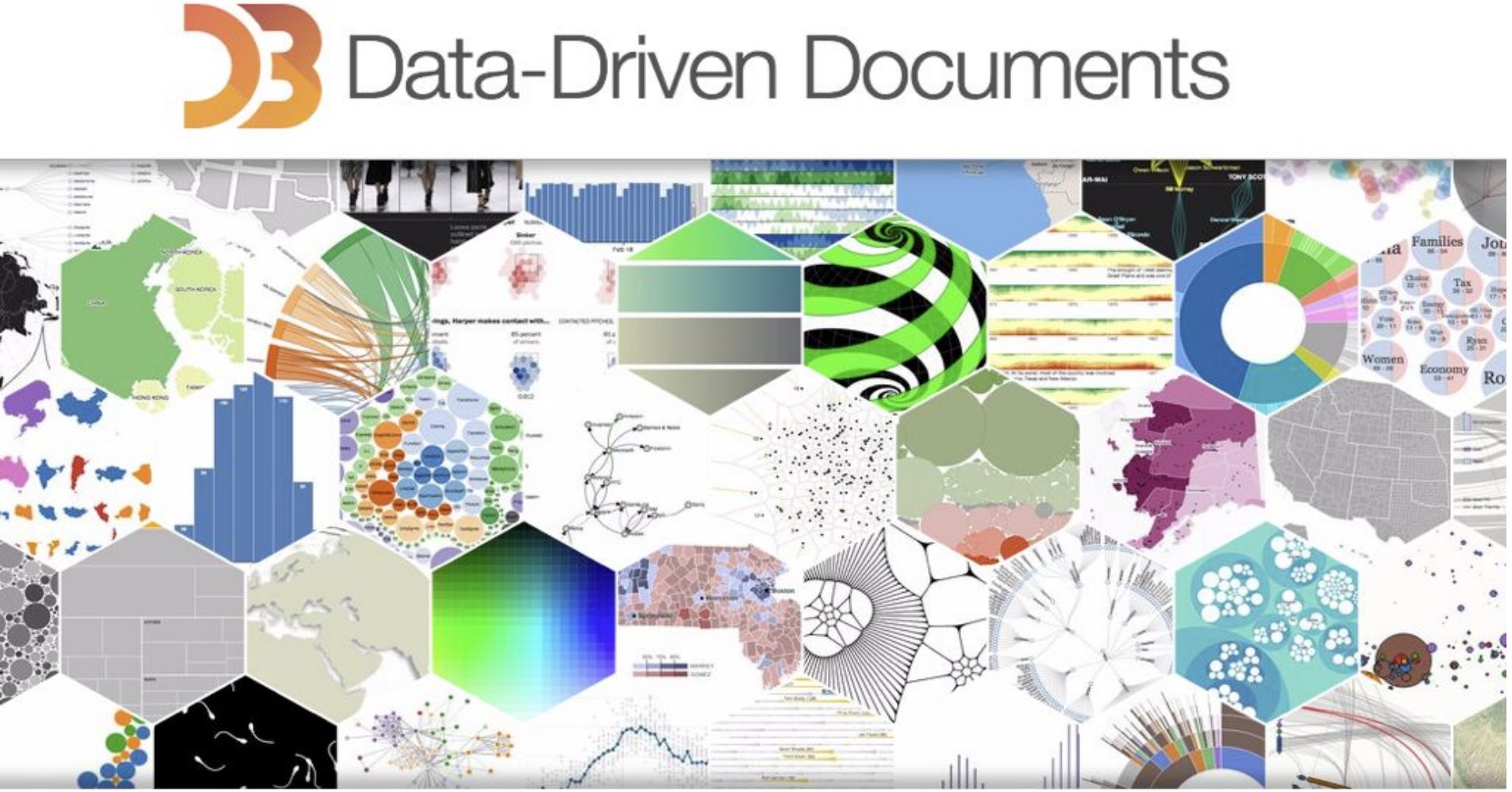


- Cross-languages
- Interactive > JavaScript

JavaScript: **D3**

http://d3.js



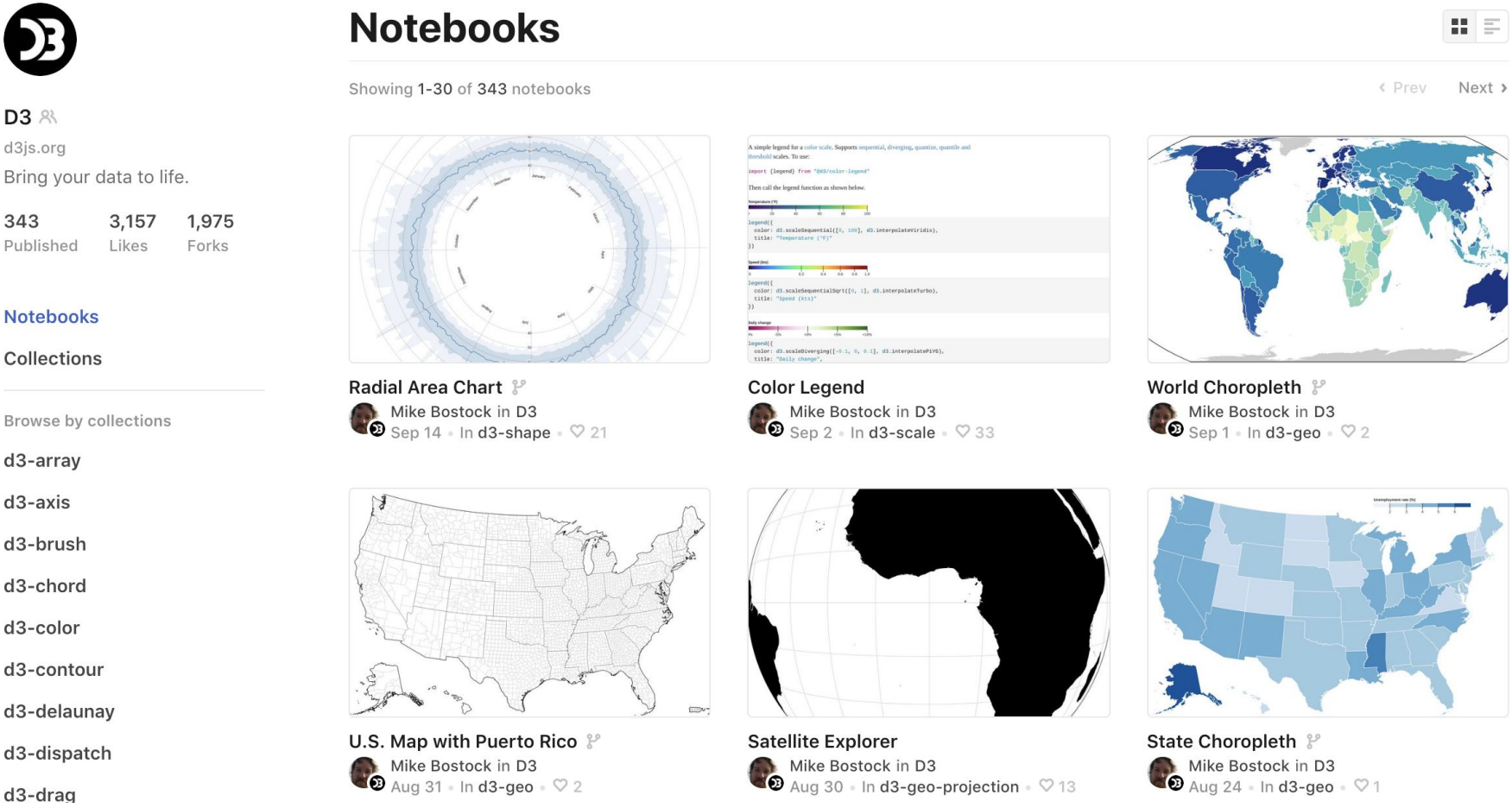
JavaScript: **D3**

http://d3.js

- **Importing** (CSV, Json, etc..)
- **Objects** (shapes) and visual attributes
- **JQuery** based
- **Functions:** interactivity, data transformation, layouts, geography, streaming, networks, scales, time, transitions, zoom, stats
- **Color scales**
- Good **examples** and huge **community**

- Tricky to learn

JavaScript: **D3—Observable**

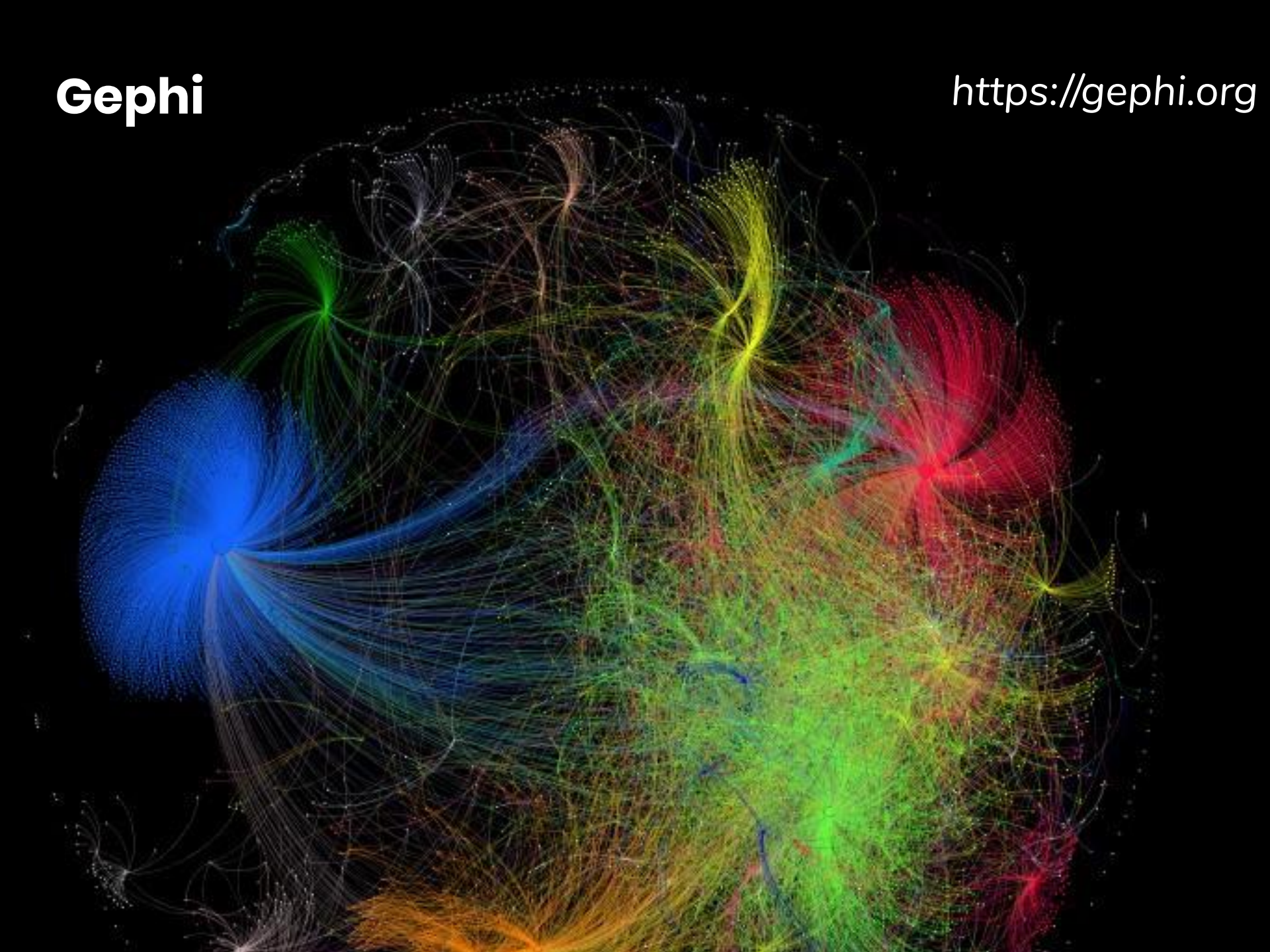


- D3 Live coding (like python notebooks)

https://observablehq.com/

Gephi

https://gephi.org



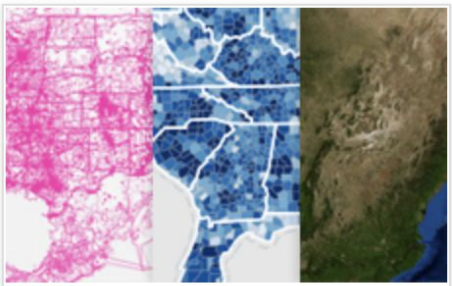
Geographical

Leaflet

A lightweight JavaScript library for making tile-based interactive maps for desktop and mobile browsers.

CartoDB

A web service for mapping, analyzing and building applications with data.



Polymaps

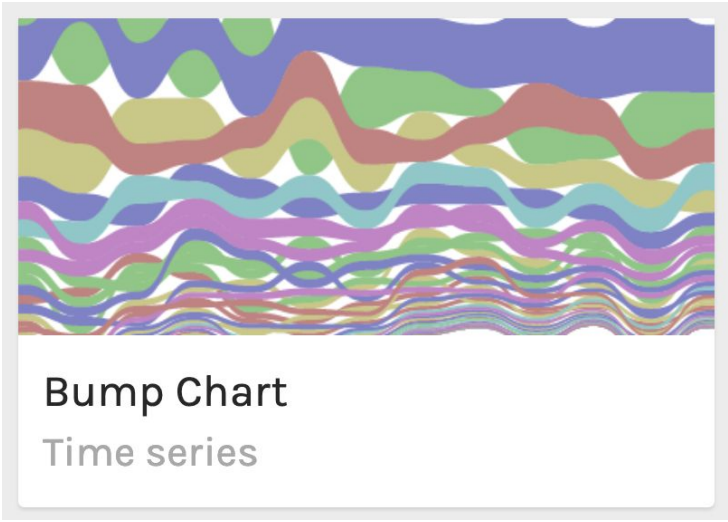
A library for making dynamic, interactive maps with image- and vector-based tiles.

MapBox

A web platform for hosting custom designed map tiles and a set of open source tools to produce them.

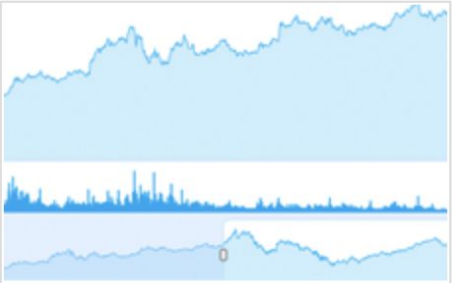
Taken from: http://datavisualization.ch

Temporal



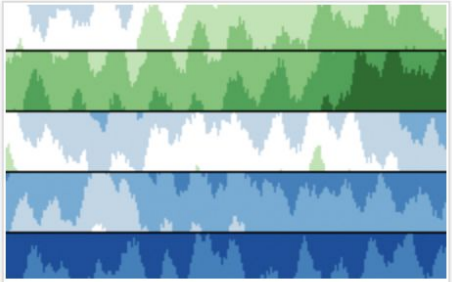
Bump Chart

Time series



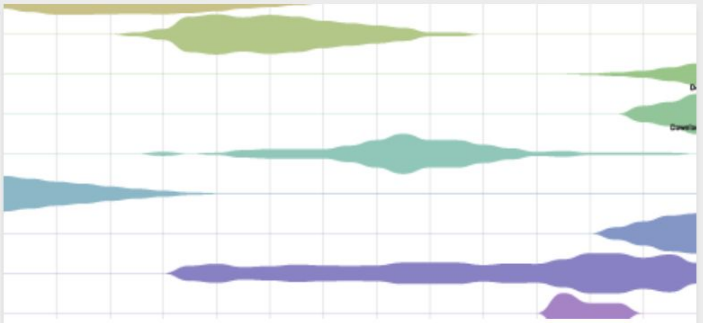
Envision.js

A library for creating fast, dynamic and interactive time series visualizations.



Cubism.js

A library for creating interactive time series and horizon graphs based on D3.js



Area graph

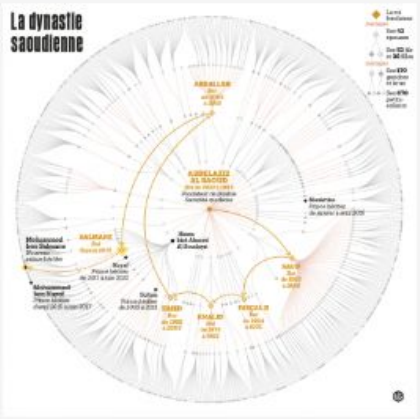
Time series

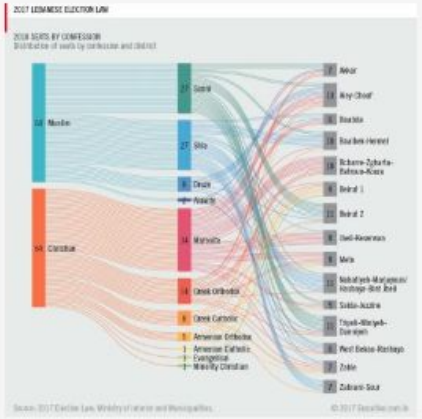
Taken from: http://datavisualization.ch

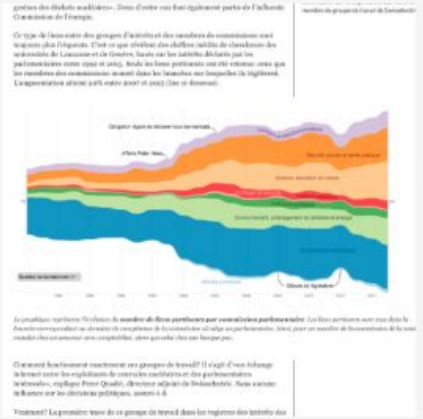
Presentation & Design

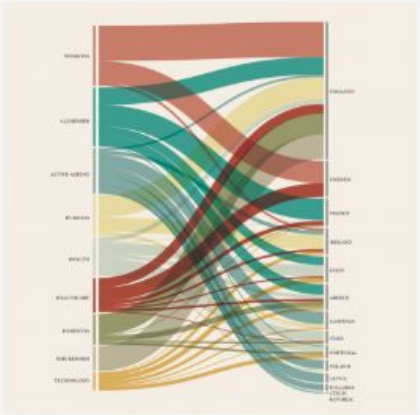
RawGraphs

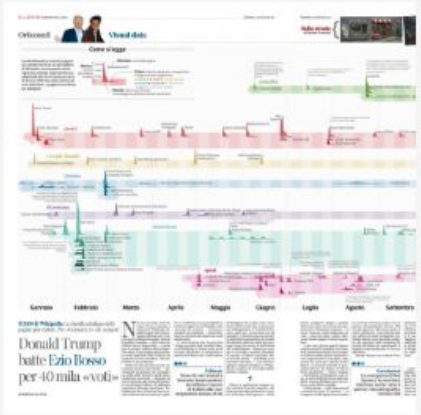
https://www.rawgraphs.io

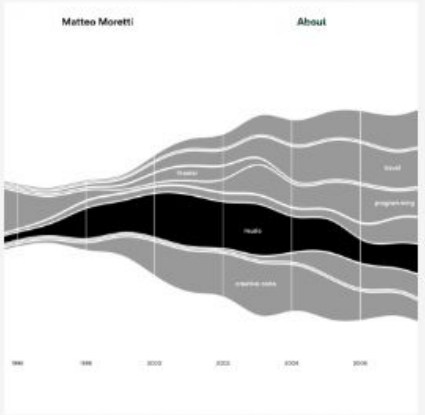






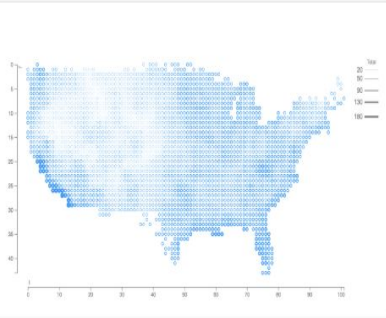






Data Illustrator

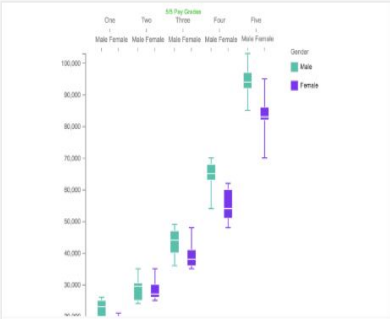
http://data-illustrator.com



The Pleasant Places to Live

Binned map showing pleasant weather days in the US.

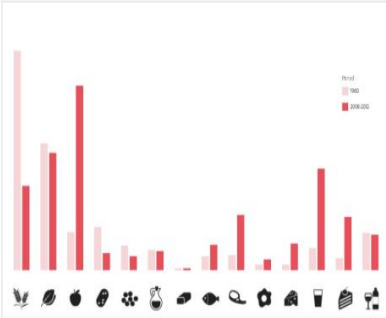
Open Example | Watch Demo



Gender Pay Gap - Box Plot

A box and whisker plot demonstrating the gender pay gap across salary grades.

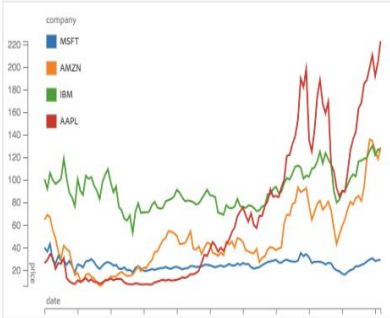
Open Example | Watch Demo



How Consumption Has Changed

How consumption of different types of food has changed since 1960

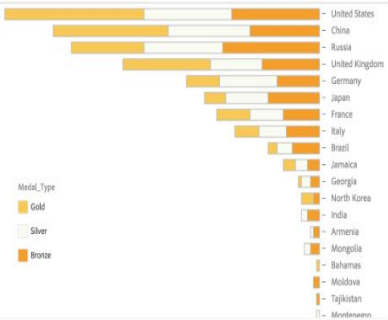
Open Example | Watch Demo



Stock Market

Monthly stock prices for four companies from 2000 to 2010

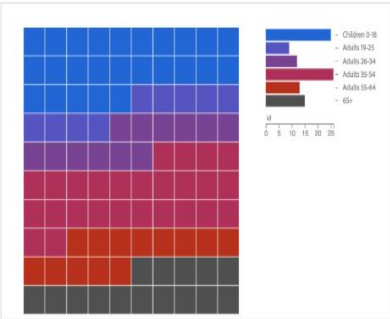
Open Example | Watch Demo



2012 Summer Olympic Medals

Stacked bar chart on the number of gold, silver and bronze medals by country

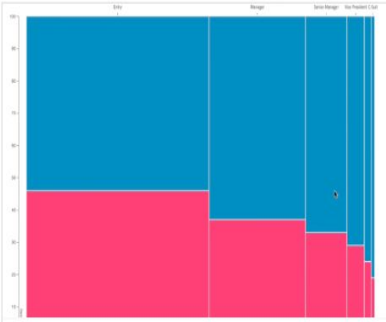
Open Example | Watch Demo



Population Distribution by Age

The distribution of population by age groups in the United States in 2016

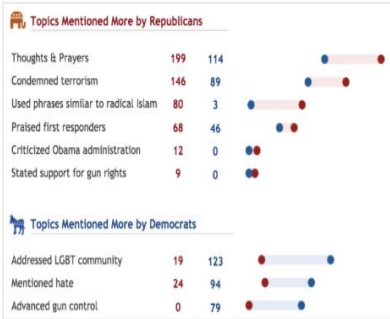
Open Example | Watch Demo



Share of Women across Job Levels

The proportion of women declines in higher job titles.

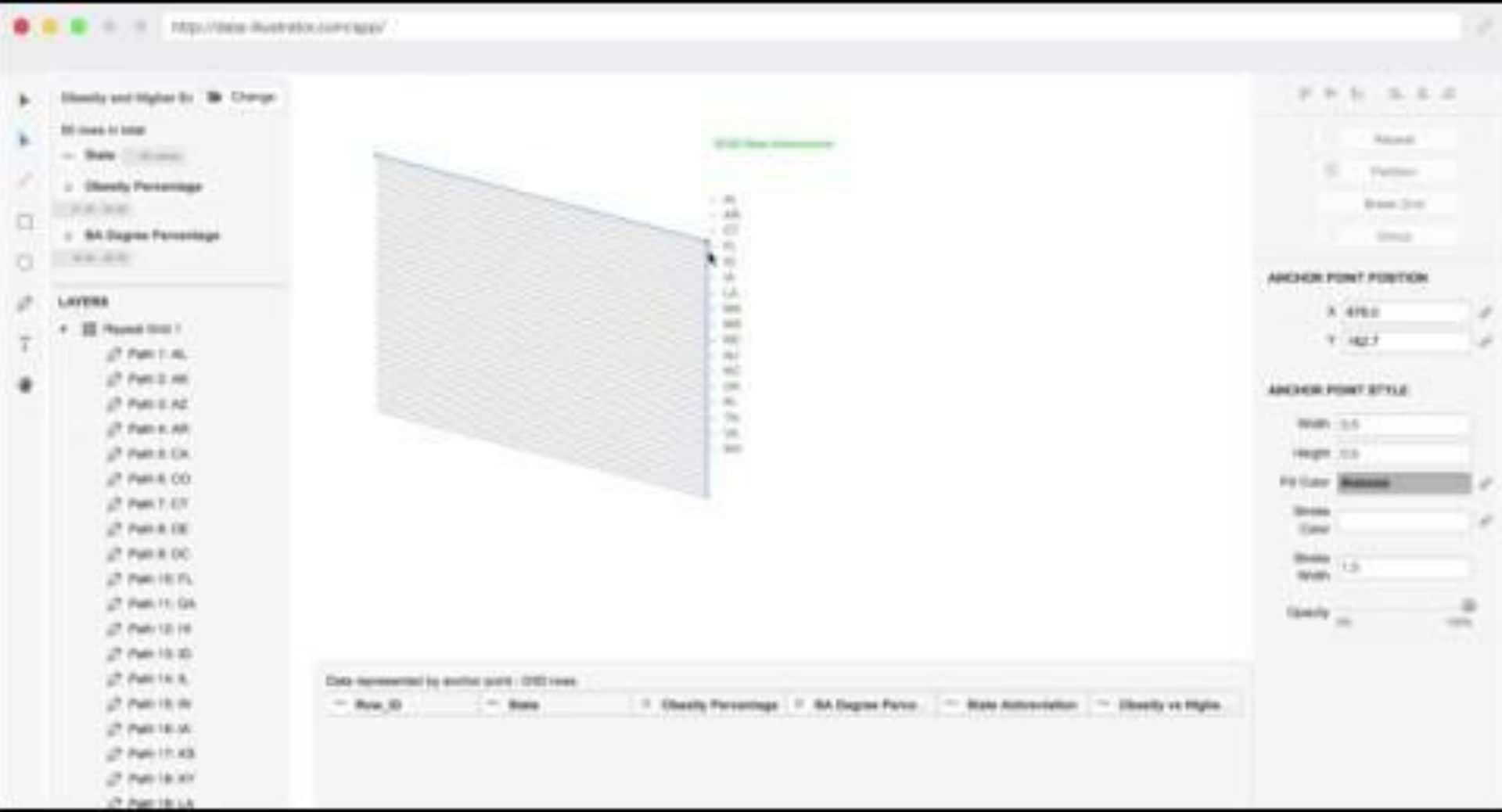
Open Example | Watch Demo



Partisan Reactions on Mass Shooting

Topics mentioned by the two parties after the Orlando nightclub shooting

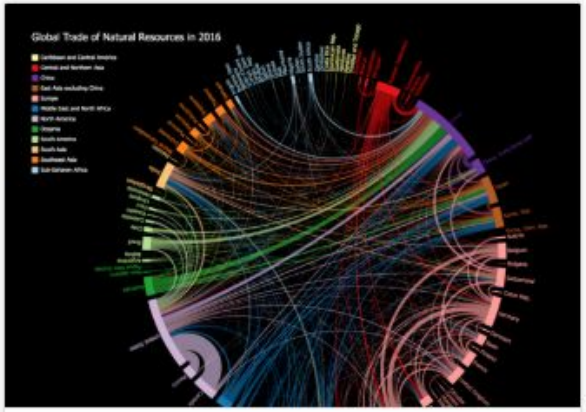
Open Example | Watch Demo



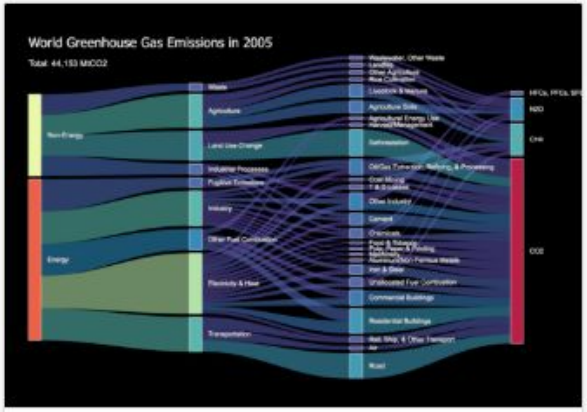
Charticulator

https://charticulator.com

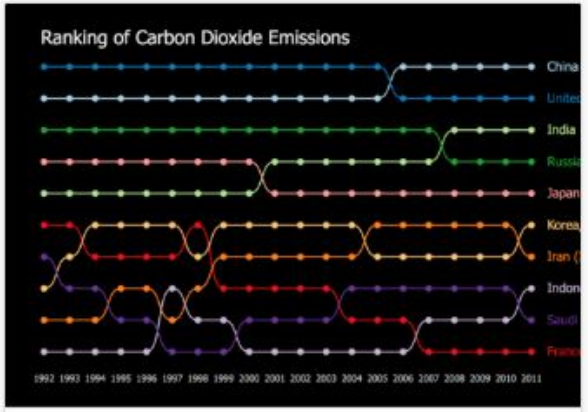
Chart & Video



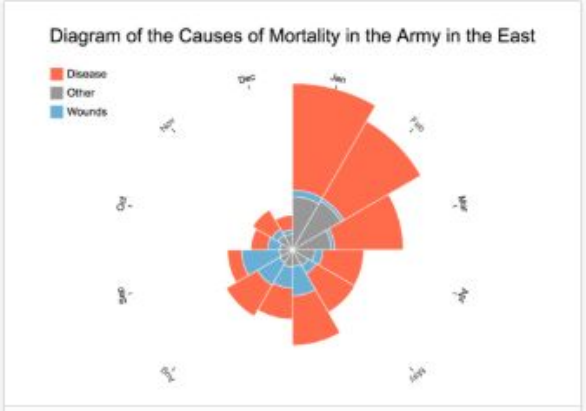
Global trade of natural resources in 2016



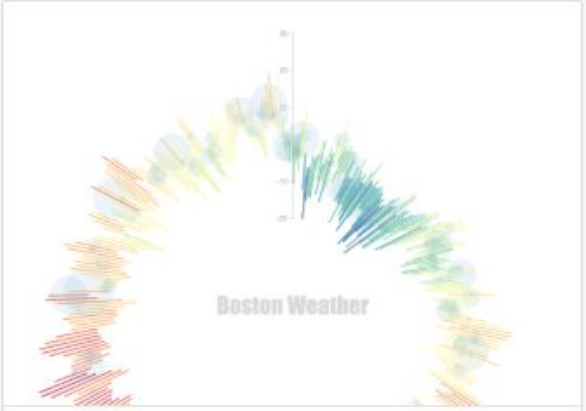
World greenhouse gas emissions



Ranking of carbon dioxide emissions of selected countries



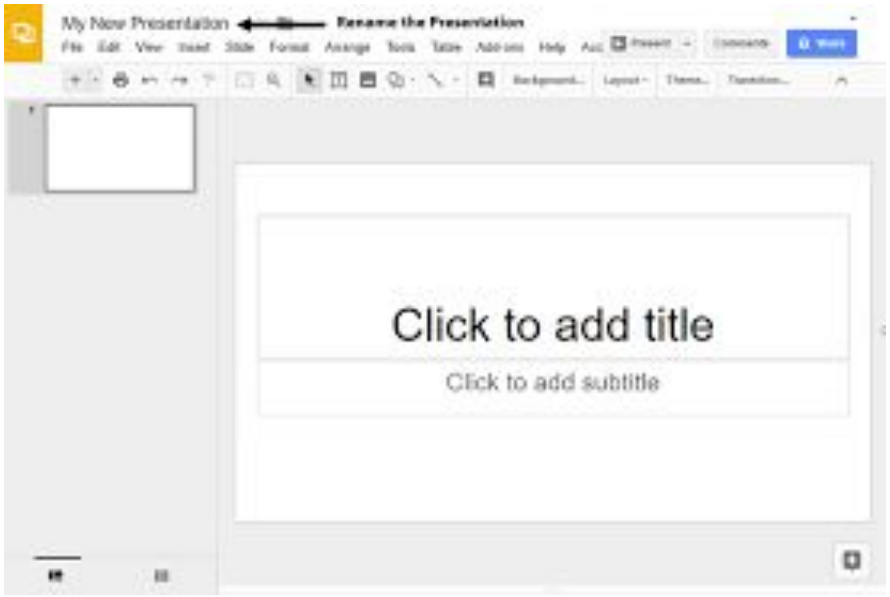
Causes of Mortality over time (Reproduction of the Nightingale chart)



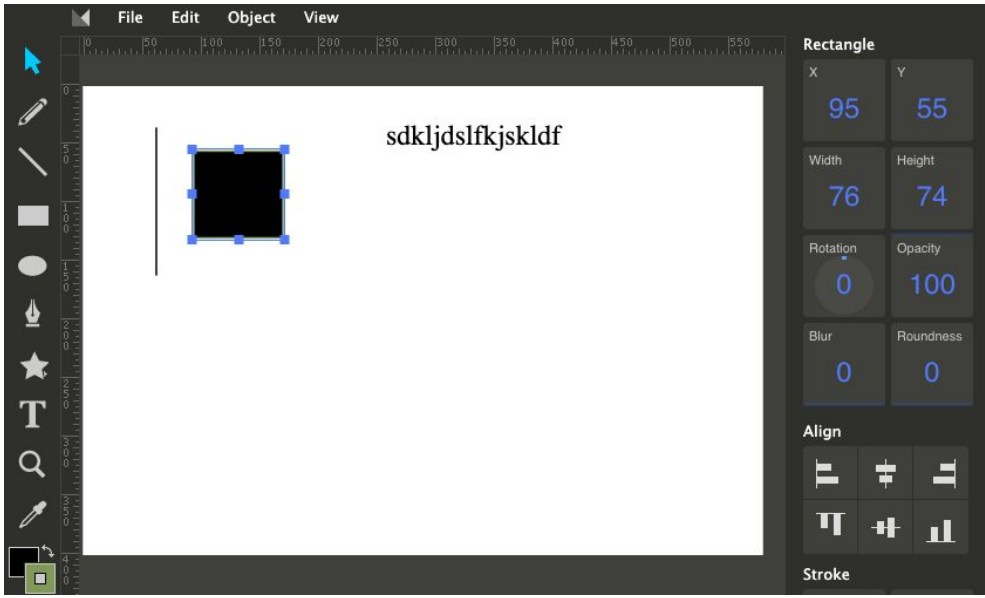
Boston weather in a year

Spiral Bar Chart

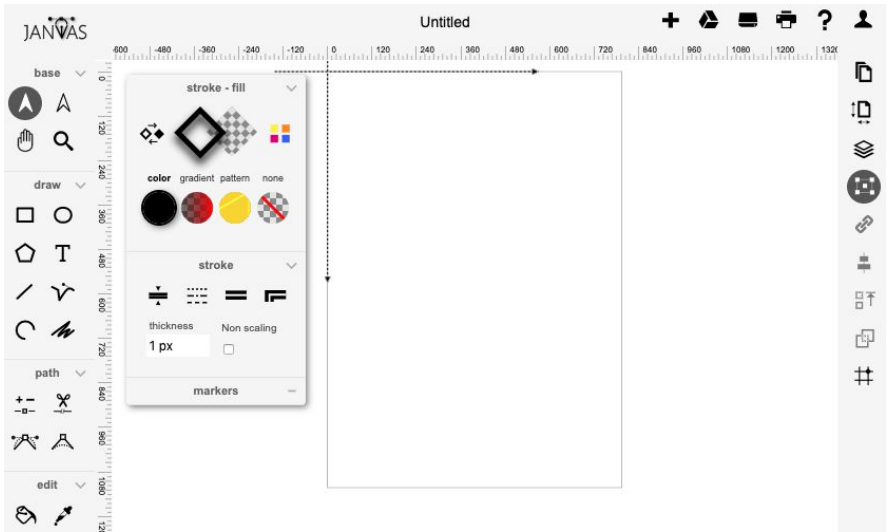
Vector Drawing



Google Slides



Method Draw
(https://editor.method.ac/)



www.janvas.com

Recap

- Simple & quick: Data Wrapper
- Scatterplots + distributions: Polestar
- Alternatively Raw graphs, but requires Illustrator pass
- Python: Seaborn (static), Bokeh or Plotly (interactive)
- Powerful & customized: D3 (start early!)
- Graphs: Gephi, Palladio, Vistorian
- Customized & no-coding: Data Illustrator, Charticulator
- Vector tools

Datavisualization.ch

DATAVISUALIZATION.CH SELECTED TOOLS Sear

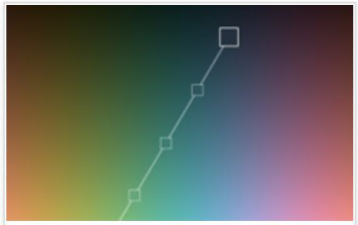
All Maps Charts Data Color

Arbor.js

A library of force-directed layout algorithms plus abstractions for graph organization and refresh handling.

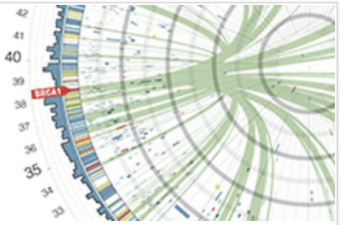
CartoDB

A web service for mapping, analyzing and building applications with data.



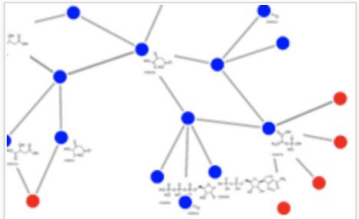
Chroma.js

Interactive color space explorer that allows to preview a set of linear interpolated equidistant colors.



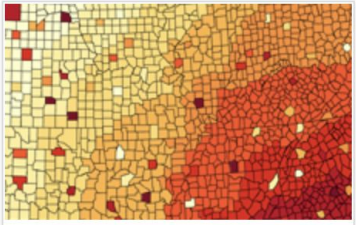
Circos

A software package for visualizing data in a circular layout.



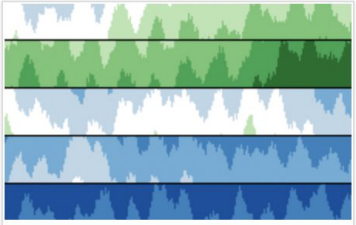
Cola.js

A library for arranging networks using constraint-based optimization techniques.



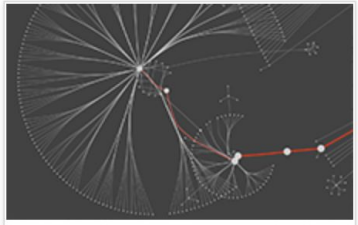
ColorBrewer

A web tool for selecting colors for maps.



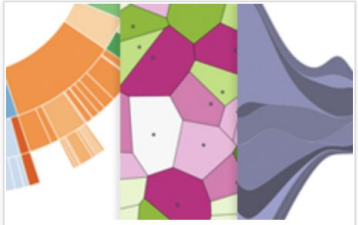
Cubism.js

A library for creating interactive time series and horizon graphs based on D3.js



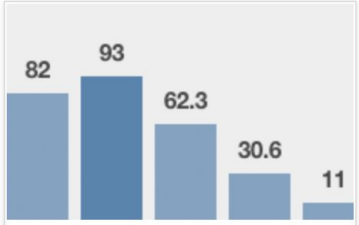
Cytoscape

An application for visualizing complex networks and integrating these with any type of attribute data.



D3.js

An small, flexible and efficient library to create and manipulate interactive documents based on data.



Dance.js

A simple data-driven visualization framework based on Data.js and Underscore.js

https://visualinteractivedata.github.io/res-tools.html

visual | interactive
data | School of Informatics, Institut for Design Informatics, University of Edinburgh

design informatics | informatics

Latest

12/19+++**Visual+interactive data** contributes towards **four** papers for CHI 2020! Congrats everyone!

10/19+++**Ryan Bowler** joins as a PhD student, working on Temporal Design.

10/19+++**Mashael Hamad Alkadi** joins as a PhD student, working on Learning Analytics and understanding visualizations.

09/19+++Tobias Kauer joins as a PhD student, working on Discursive Visualizations.

07/19+++Our survey on geographic networks goes online.

Visualization Tools

General Purpose

- Google Fusion tables: for very simple graphs, networks, and maps; supports some data cleaning and transformation (e.g., automatic recognition of location names and transformation into latitude and longitude).
- Tableau: A powerful commercial tool for visualizing data; does not support network or text visualization. Free 2-week trial and free versions for students available.
- Raw graphs: Free and online tool to import and map data to standard visualizations, but also parallel coordinates plot http://rawgraphs.io
- [PoleStar: Simple webinterface that automatically creates visualizations based on the data input (Mainly scatterplots and distributions with various style attributes). Based on * [Vega. https://vega.github.io/polestar
- [The Gamma: A JavaScript library that lets anyone create transparent and open data visualizations that are linked to the original data source and encourages the reader to further explore data and find interesting facts on their own. https://thegamma.net
- [iVisDesigner (Scatterplots, Nodelink, Matrices, PCP, timelines): https://donghaoren.org/ivisdesigner/

Visualization in R

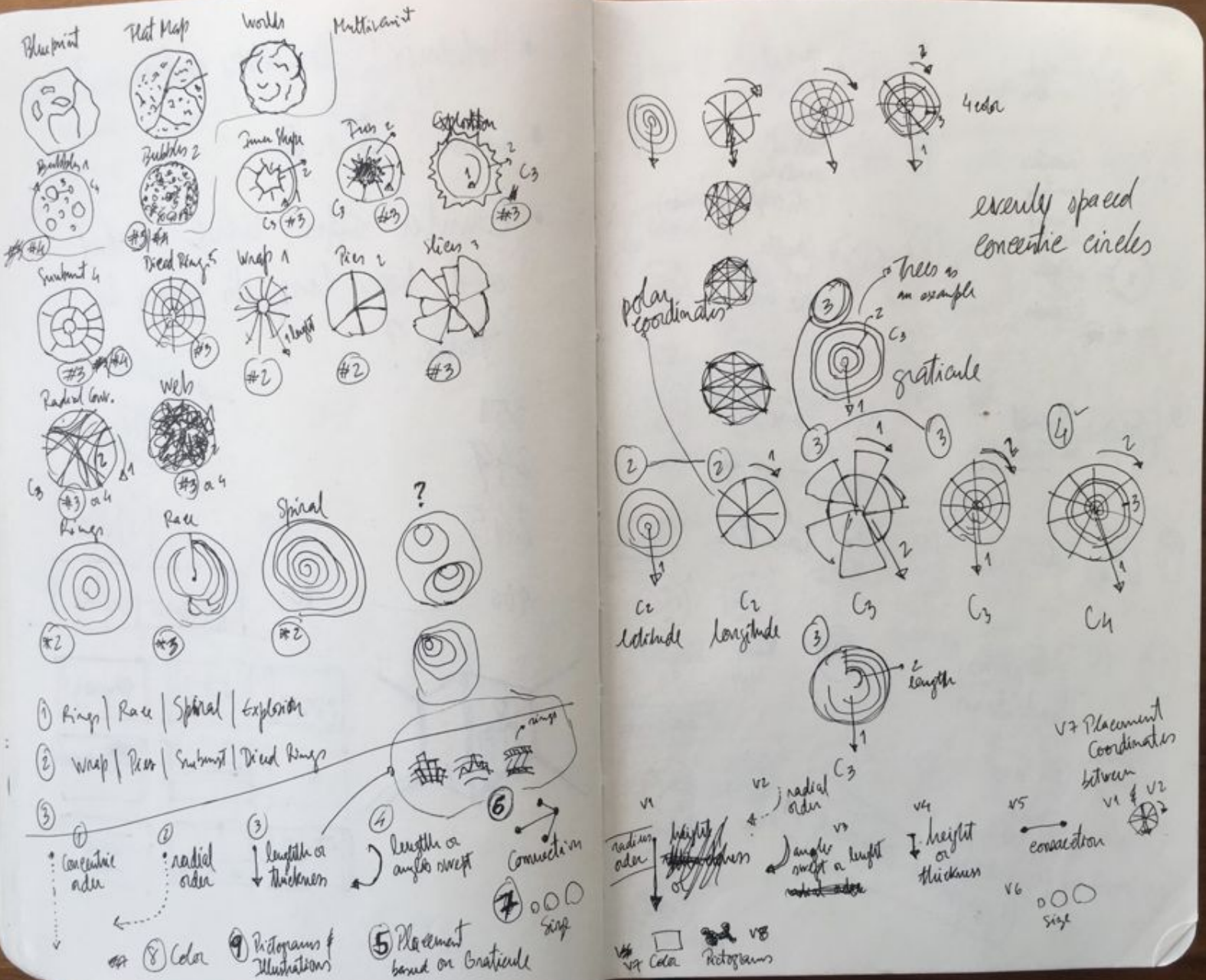
- [https://www.r-bloggers.com/7-visualizations-you-should-learn-in-r
- [Ggplot2
- [R Graph gallery: http://www.r-graph-gallery.com/

Visualization in Python

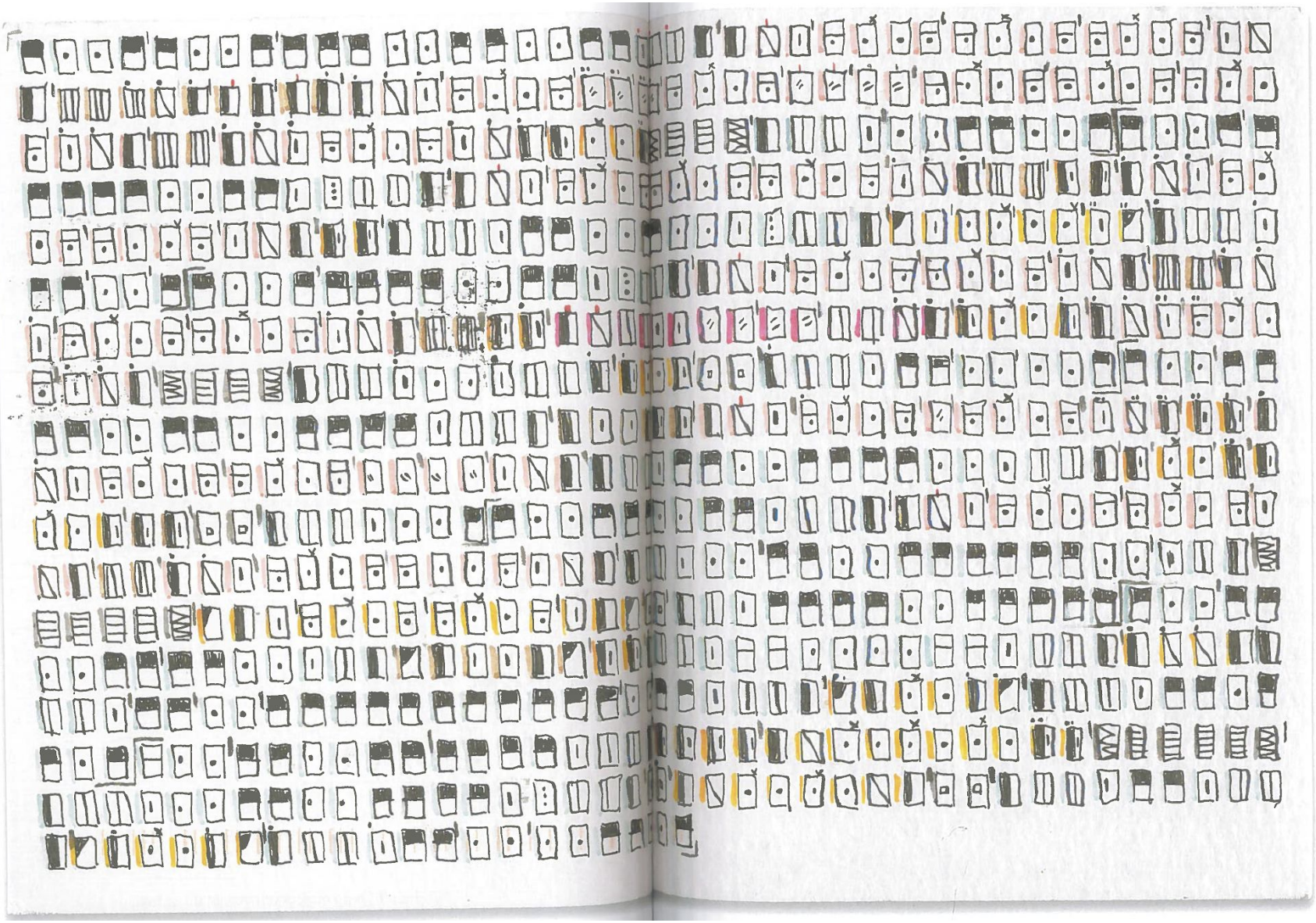
- [General library of python charts: https://python-graph-gallery.com

Suggestions & experiences?

Sketching

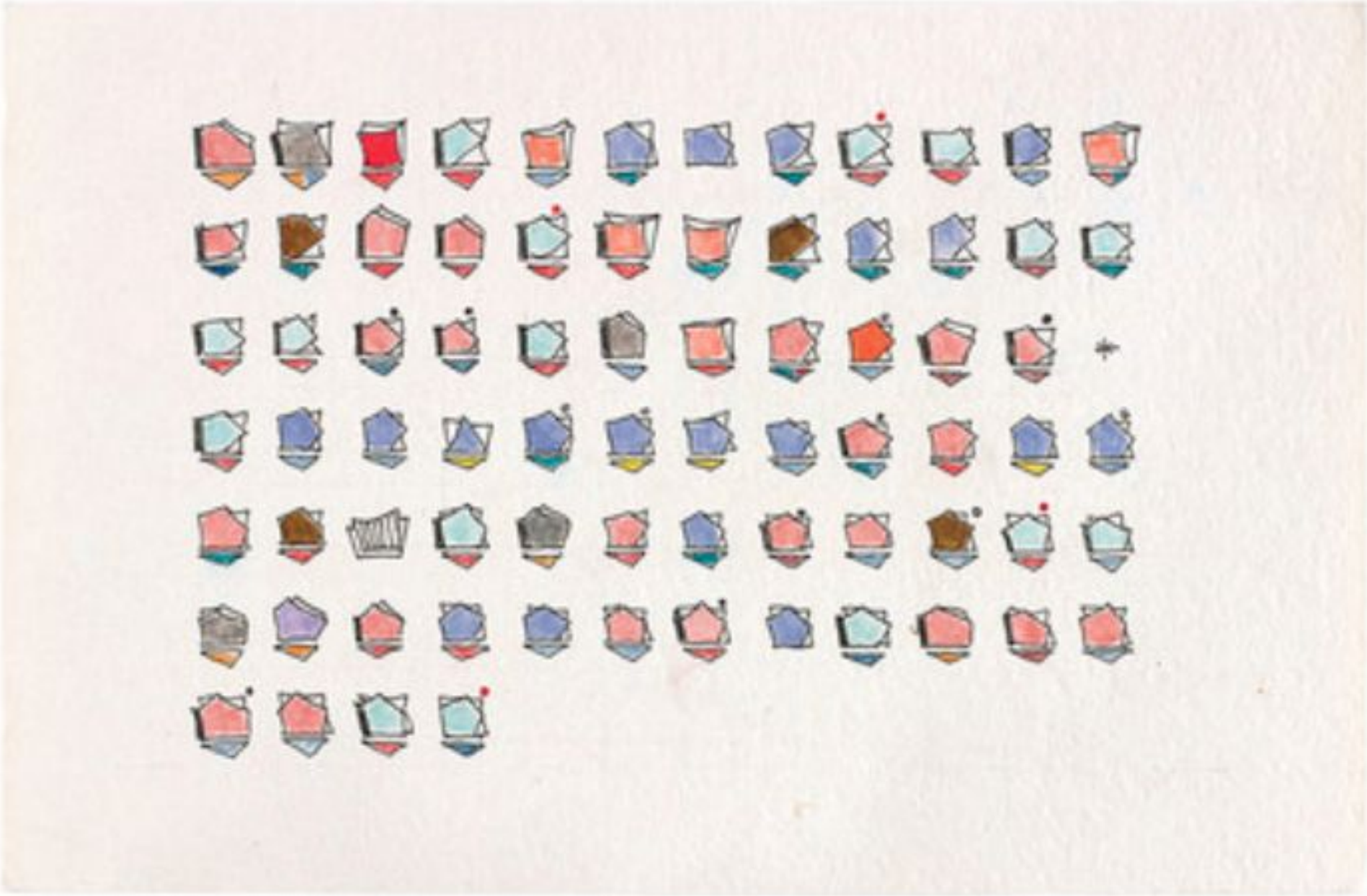

Blueprint
Heat Map
Worlds
Multivariant
Bubbles 1
Bubbles 2
Inner Shape
Dots
Exploration
Sunburst
Dried Rings
Wrap
Piers
Slices
Radial Grid
Web
Rings
Race
Spiral
?
1 Rings | Race | Spiral | Exploration
2 Wrap | Piers | Sunburst | Dried Rings
3
Concentric order
radial order
Length or thickness
Length or angle sweep
Connection
Size
8 Color
9 Pictograms & Illustrations
5 Placement based on Graticule
4 cols
evenly spaced concentric circles
polar coordinates
Trees as an example
graticule
C2 latitude
C2 longitude
C3
C3
C4
length
radial order
V1
V2
V3
V4
V5
V6
V7 Placement Coordinates between V1 & V2
connection
height or thickness
Size
V7 Color
V8 Pictograms

Sketching: **Visual Marks**

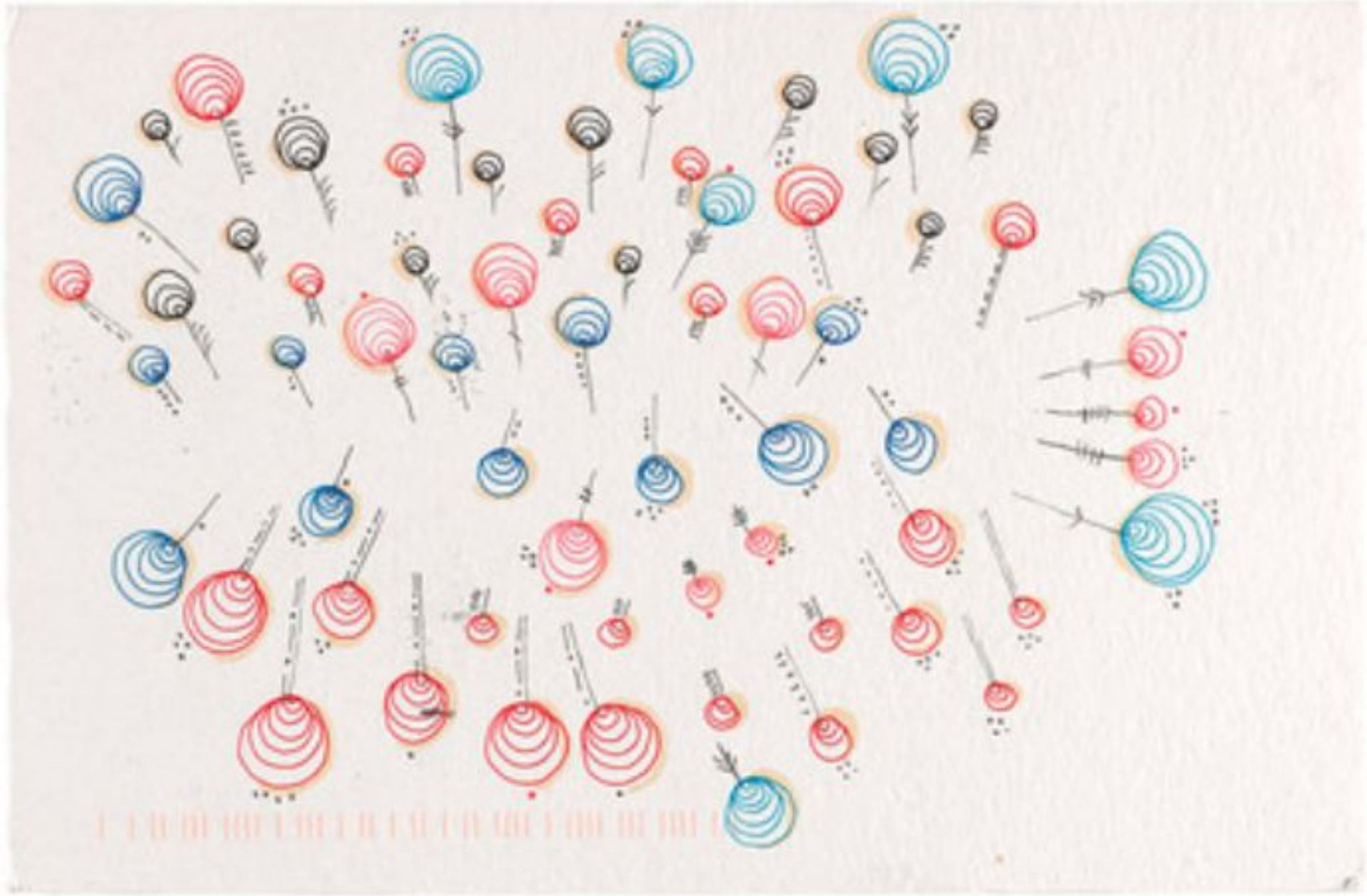


Dear Data

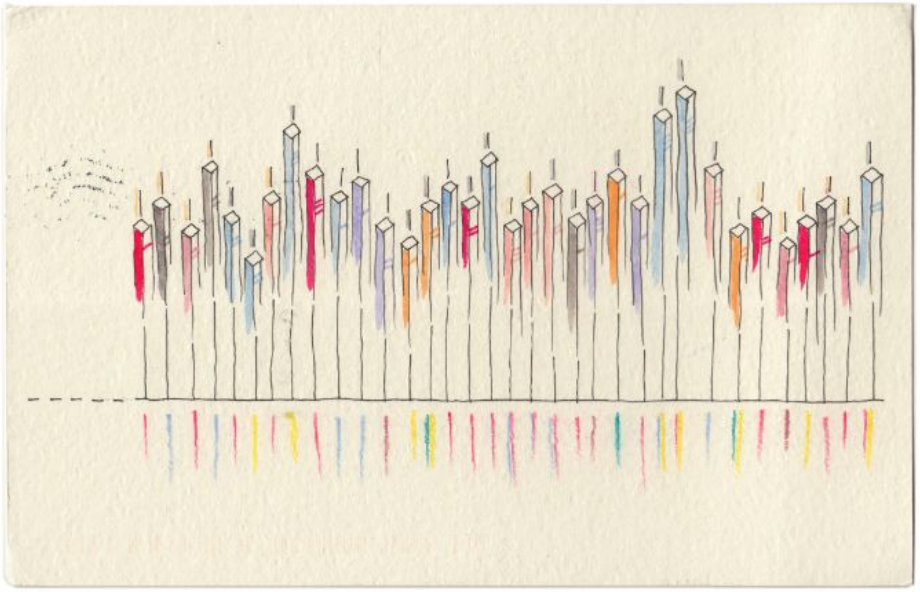
http://www.dear-data.com

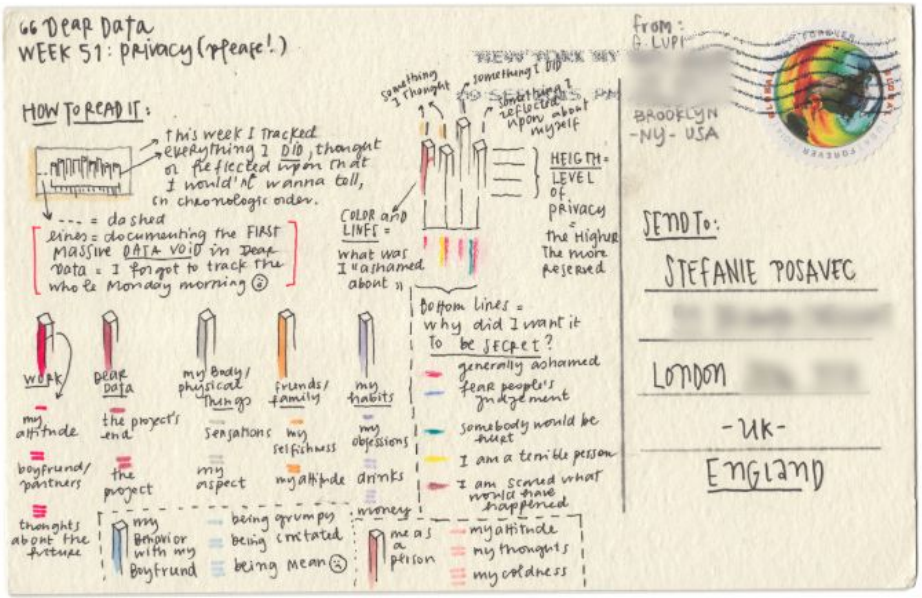


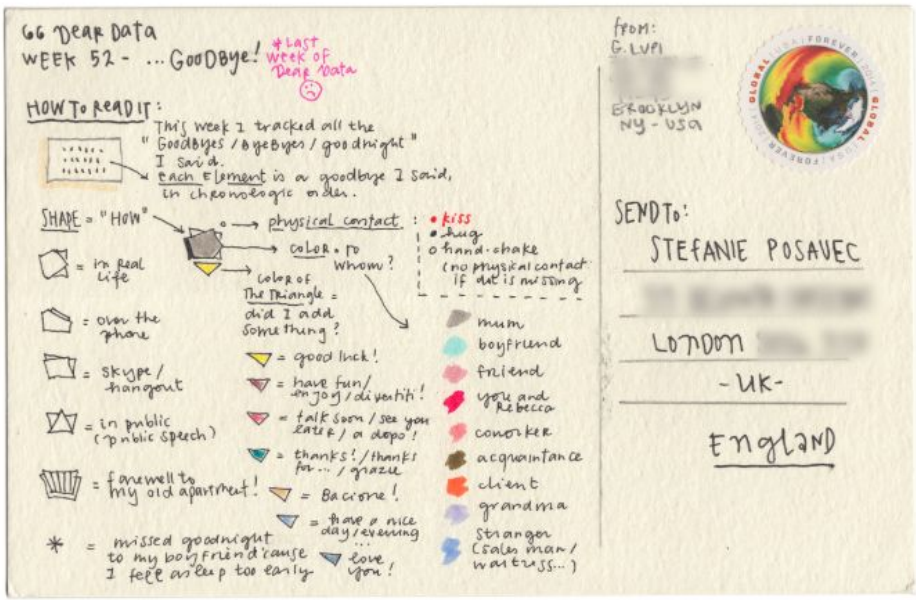
Dear Data



Dear Data



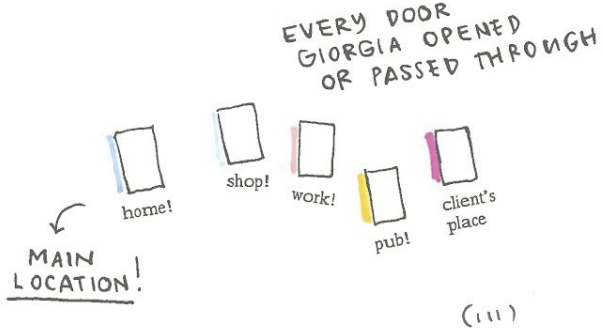

66 Dear Data
WEEK 51: privacy (ppeace!)
HOW TO READ IT:
this week I tracked everything I DID, thought or reflected upon that I would'nt wanna tell, in cheonologic order.
--- = dashed lines = documenting the FIRST MASSIVE DATA VOID in Dear Data = I forgot to track the whole Monday morning
COLOR AND LINES = what was I "ashamed about"
HEIGTH = LEVEL OF PRIVACY = the Higher the more reserved
Bottom lines = why did I want it to be SECRET?
generally ashamed
fear people's judgement
somebody would be hurt
I am a terrible person
I am scared what would have happened
work
dear data
my Body/ physical things
friends/ family
my habits
my attitude
the project's end
sensations
my selfishness
my obsessions
boyfriend/ partners
the project
my aspect
my attitude
drinks
thoughts about the future
my behavior with my Boyfriend
being grumpy
being irritated
being mean
money
me as a person
my attitude
my thoughts
my coldness
from: G. LUPI
BROOKLYN -NY- USA
SEND TO:
STEFANIE POSAVEC
LONDON
-UK-
ENGLAND

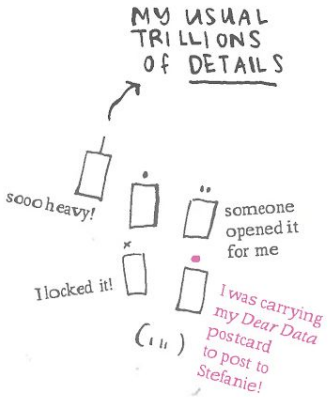

66 Dear Data
WEEK 52 - ... GOODBYE!
#LAST WEEK OF Dear Data
HOW TO READ IT:
This week I tracked all the "GoodByes / ByeByes / goodnight" I said.
Each Element is a goodbye I said, in chronologic order.
SHAPE = "HOW"
= in real life
= over the phone
= skype / hangout
= in public (=public speech)
= farewell to my old apartment!
= missed goodnight to my boyfriend cause I fell asleep too early
physical contact
COLOR = to Whom?
color of The triangle = did I add something?
= good luck!
= have fun / enjoy / divertiti!
= talk soon / see you later / a dopo!
= thanks! / thanks for... / grazie
= Bacione!
= have a nice day / evening ...
= love you!
kiss
hug
hand-shake (no physical contact if dat is missing)
mum
boyfriend
friend
you and Rebecca
coworker
acquaintance
client
grandma
Stranger (Scalo man / waitress...)
from: G. LUPI
BROOKLYN NY - USA
SEND TO:
STEFANIE POSAVEC
LONDON
-UK-
ENGLAND

Sketching: **Visual Marks**

This week Giorgia and Stefanie were tracking the doors they passed through. As much as it sounds an unusual dataset, it was a pretext to show each other the pace of their days through their external and internal environments.

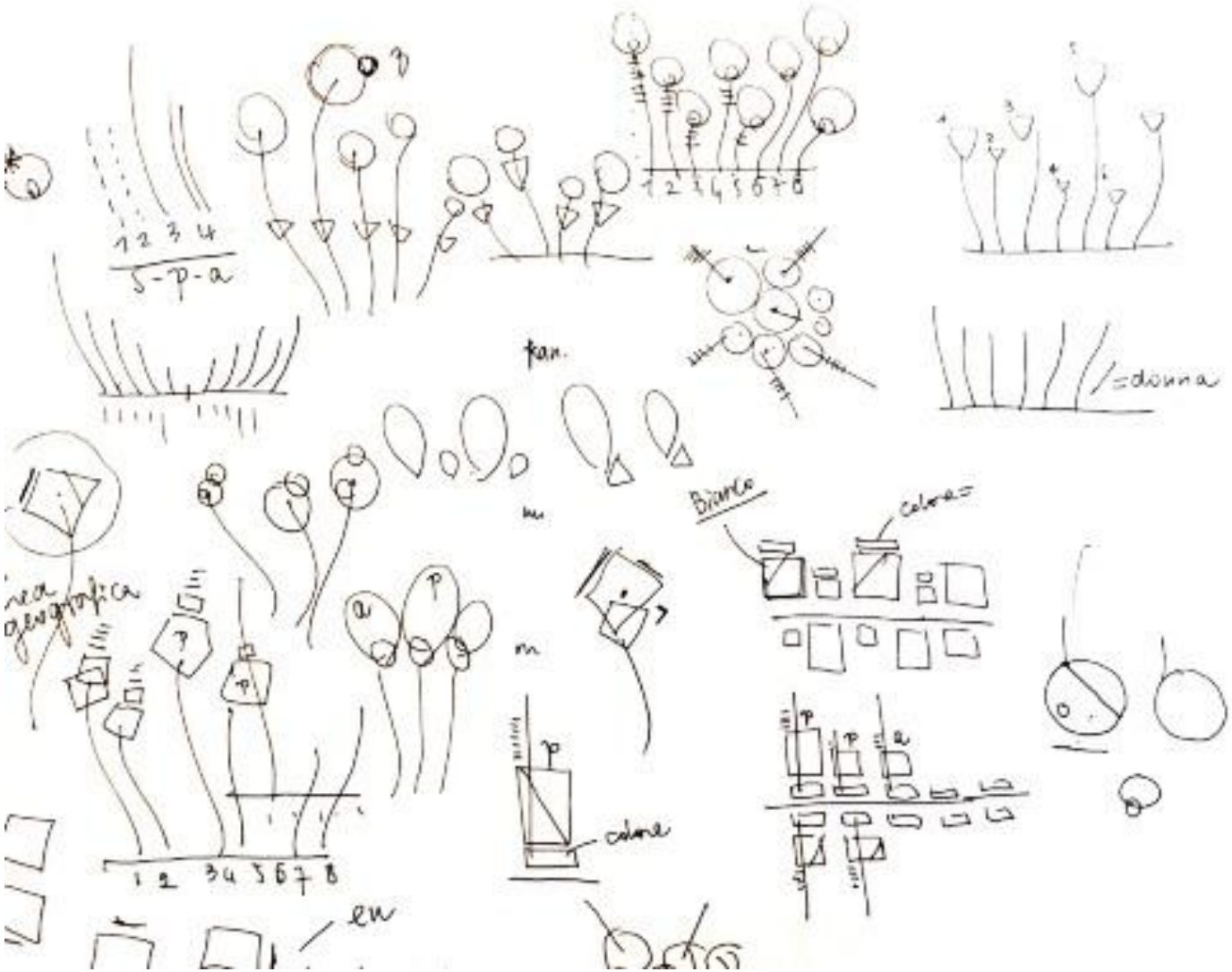
It is a reminder that you can still see the story of a life lived, even in the most uncommon types of data tracking, if you add the right details to your gathering.



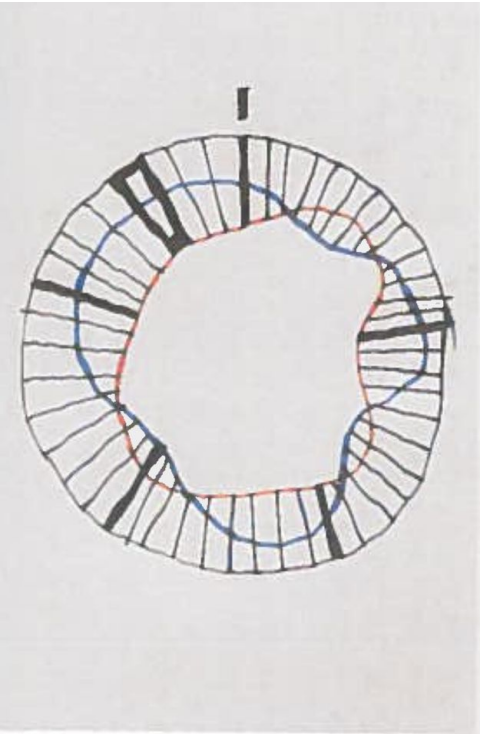


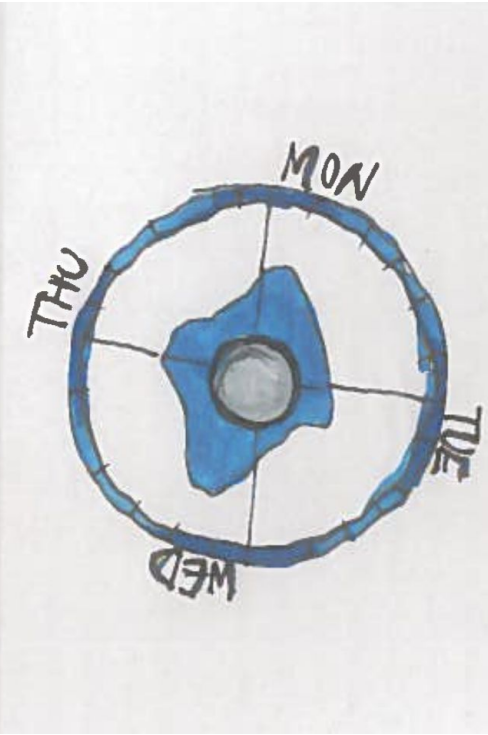
Why can't I help ooooooooverdetailing my data collections???

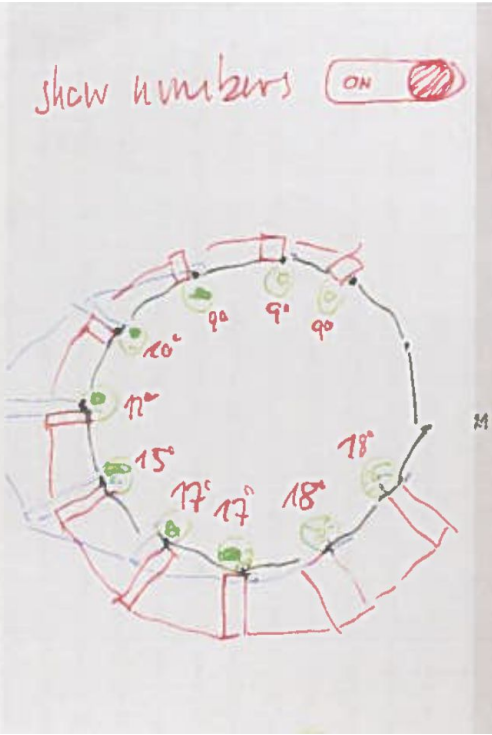
Sketching: **Shapes and forms**

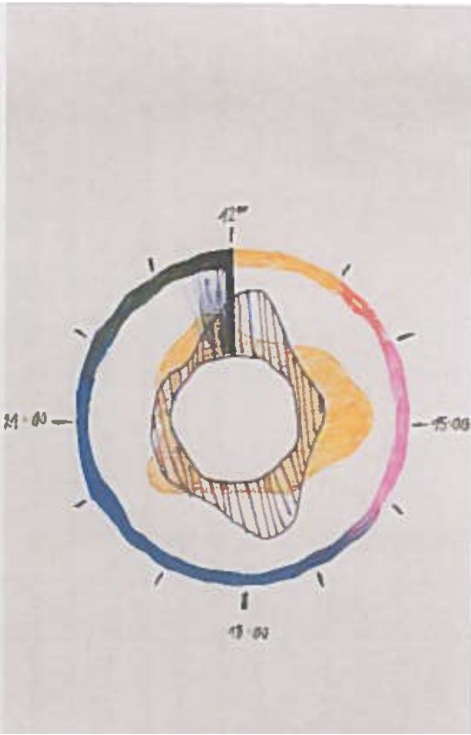


Sketching: **Alternatives**

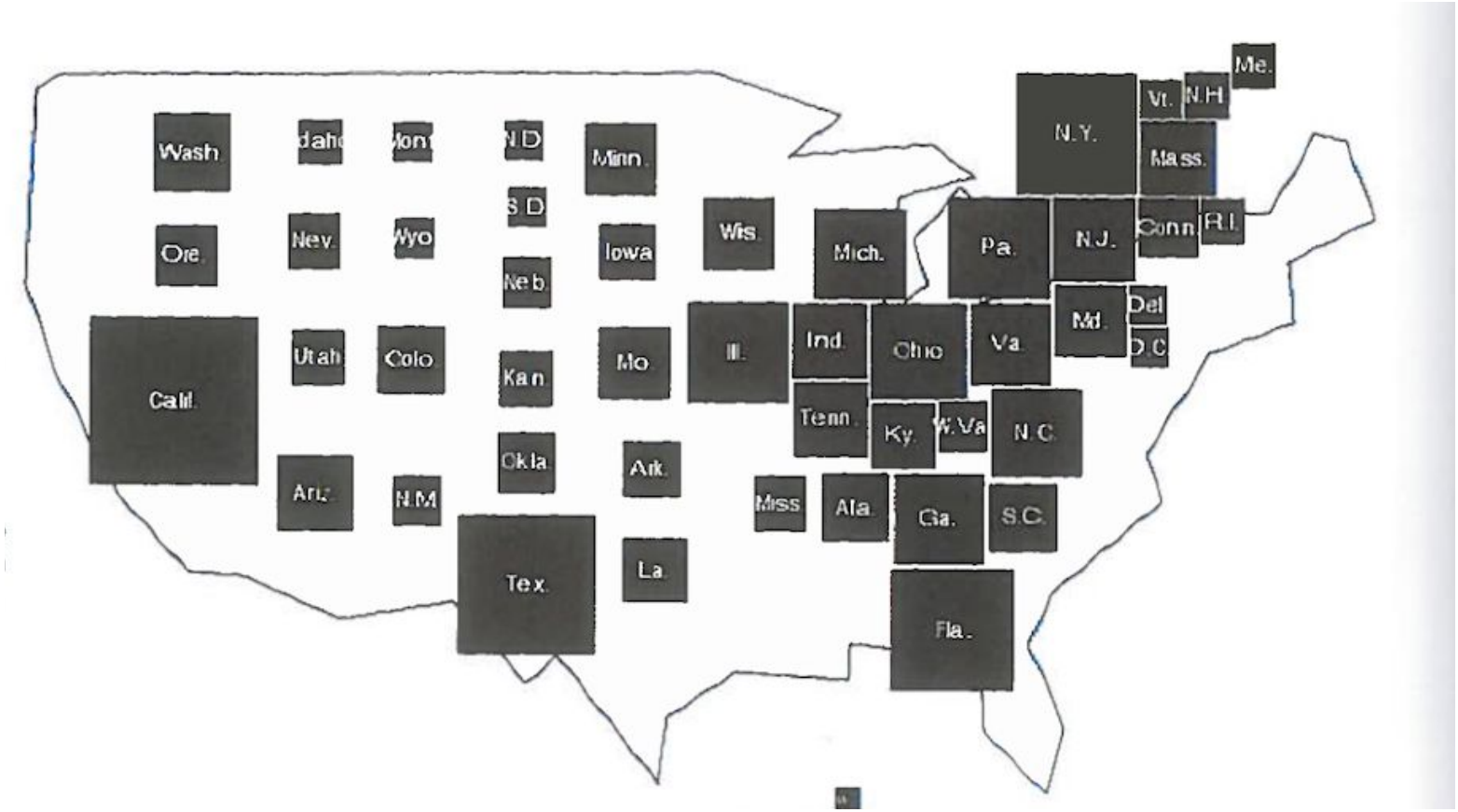






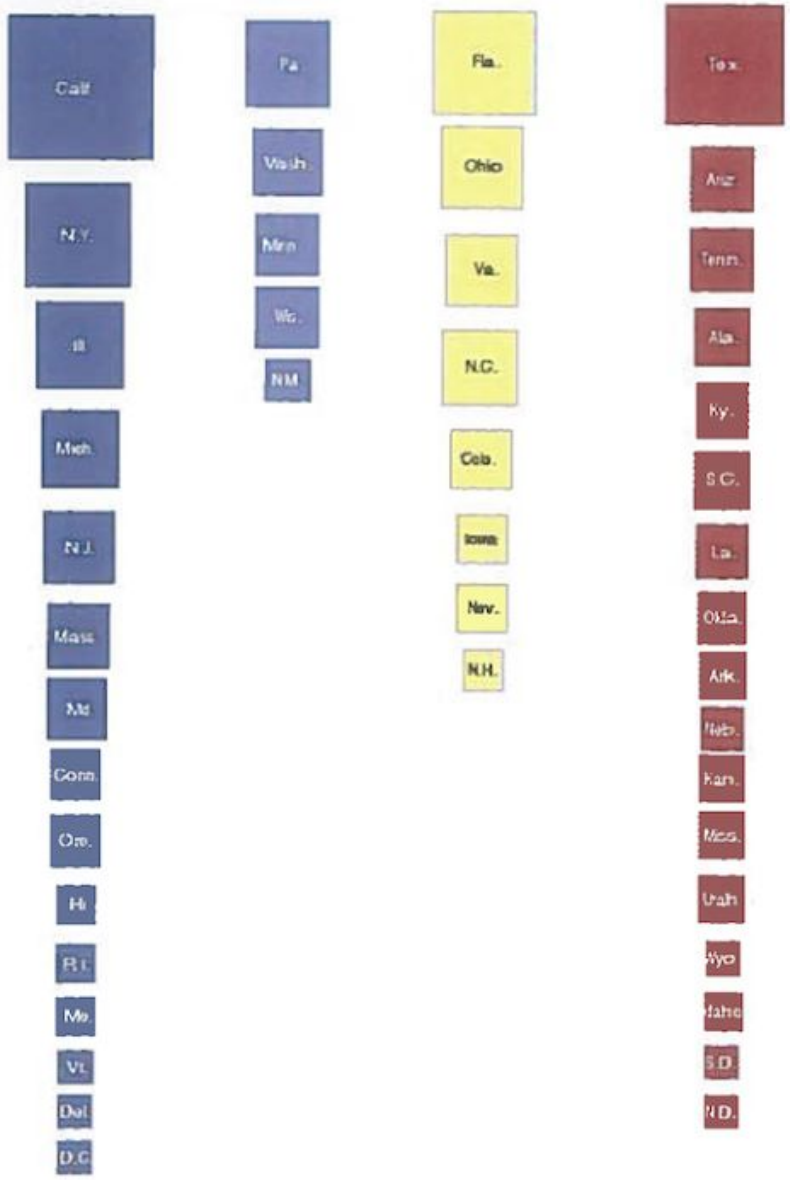


Evolution

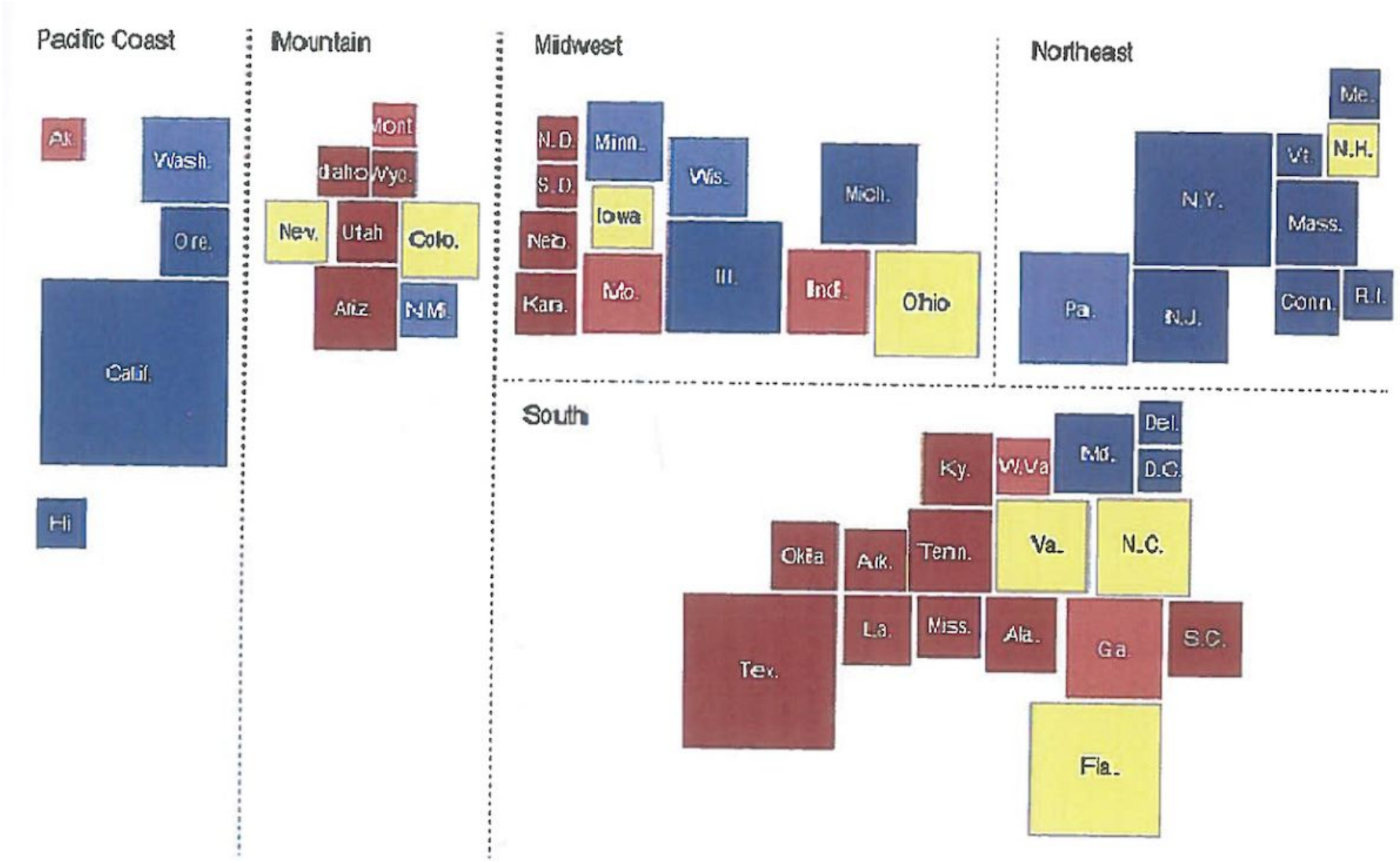

Wash.
Ore.
Calif.
Nev.
Utah
Ariz.
Colo.
N.M.
Wyo.
N.D.
S.D.
Neb.
Kan.
Okla.
Tex.
Minn.
Iowa
Mo.
Ark.
La.
Wis.
Ill.
Mich.
Ind.
Ohio
Ky.
Tenn.
Miss.
Ala.
Ga.
Fla.
S.C.
N.C.
Va.
W.Va.
Pa.
N.Y.
N.J.
Md.
Del.
D.C.
Conn.
R.I.
Mass.
Vt.
N.H.
Me.

Evolution

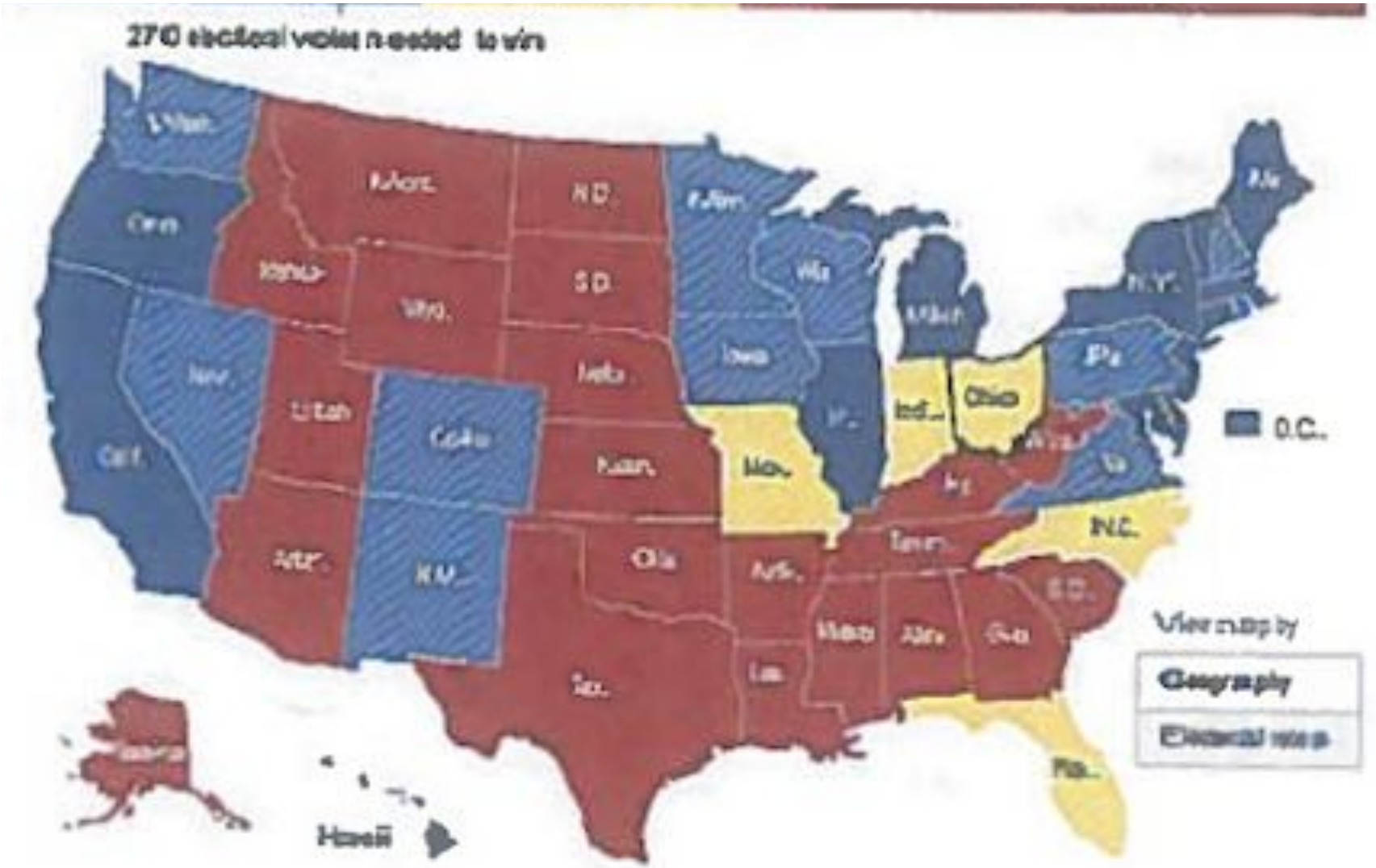
Evolution


Calif
N.Y.
Ill
Mich
N.J.
Mass
Md
Conn
Ore.
Hi
R.I.
Me.
Vt.
Del
D.C
Pa
Wash.
Minn
Wis.
NM
Fla.
Ohio
Va.
N.C.
Colo.
Iowa
Nev.
N.H.
Tex.
Ariz
Tenn.
Ala.
Ky.
S.C.
La.
Okla.
Ark.
Nebr.
Kan.
Miss.
Utah
Wyo
Idaho
S.D.
N.D.

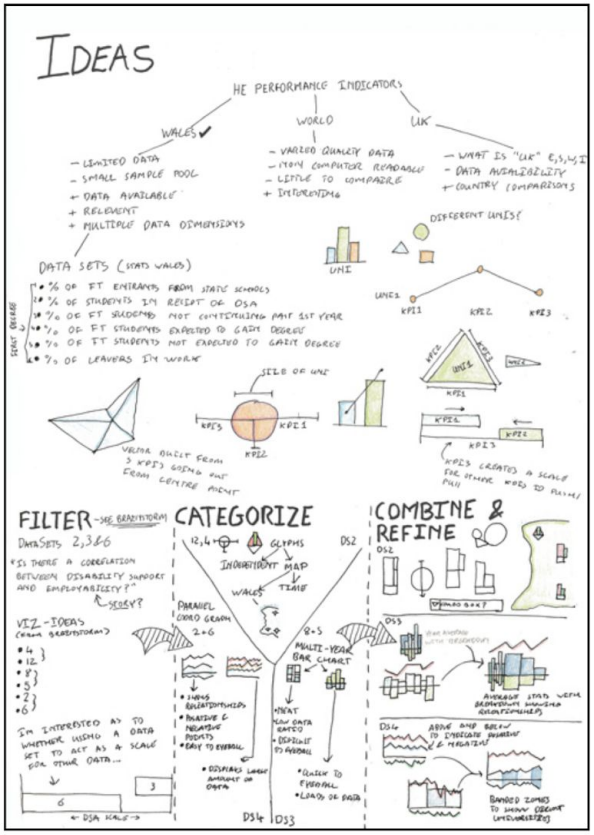
Evolution

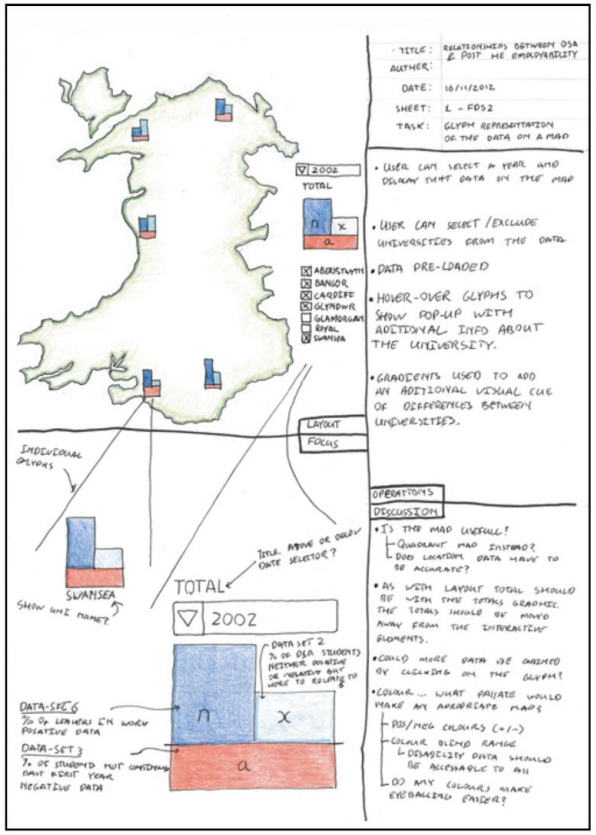

Pacific Coast
Mountain
Midwest
Northeast
South
Ak.
Wash.
Ore.
Calif.
Hi
Mont.
Idaho
Wyo.
Nev.
Utah
Colo.
Ariz.
N.M.
N.D.
Minn.
S.D.
Wis.
Iowa
Mich.
Neb.
Kans.
Mo.
Ill.
Ind.
Ohio
Me.
N.H.
Vt.
N.Y.
Mass.
Pa.
N.J.
Conn.
R.I.
Del.
Md.
D.C.
Ky.
W.Va.
Okla.
Ark.
Tenn.
Va.
N.C.
Tex.
La.
Miss.
Ala.
Ga.
S.C.
Fla.

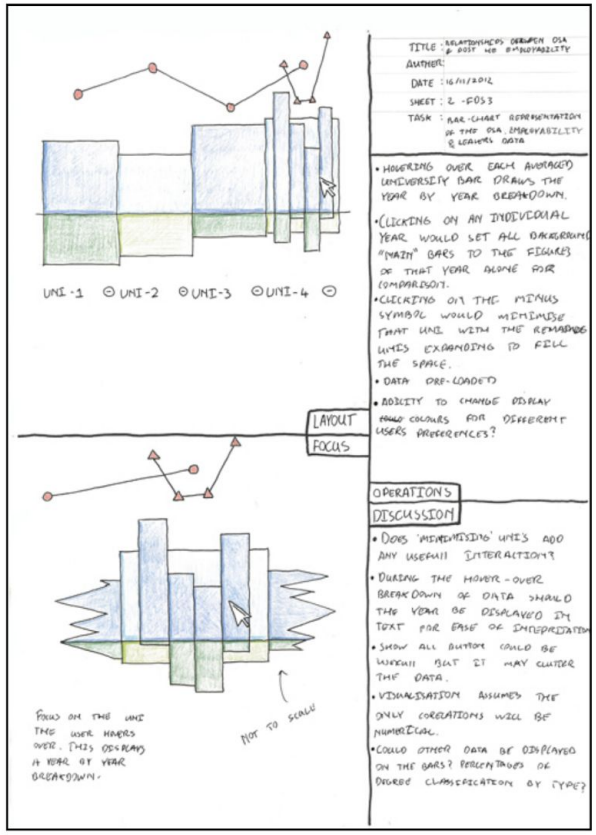
Evolution



Sketching: **Views and Layouts**

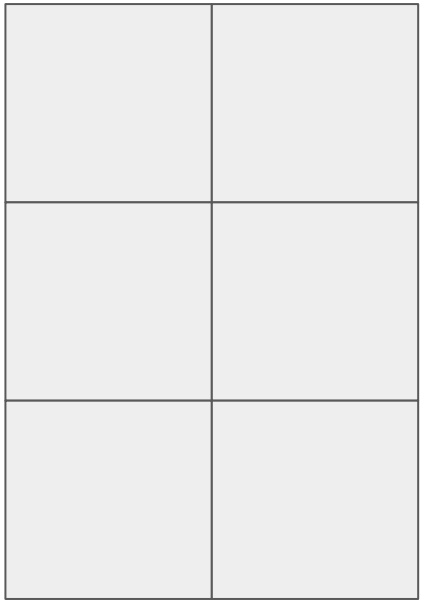






Data Challenge

1. Data & Structure
2. Context & Audience
3. Questions & Messages



total_bill
16.99
10.34
21.01
23.68
24.59
25.29
8.77
26.88
15.04
14.78
10.27
35.26

total_bill	tip	sex	smoker	day	time	size
16.99	1.01	Female	No	Sun	Dinner	2
10.34	1.66	Male	No	Sun	Dinner	3
21.01	3.5	Male	No	Sun	Dinner	3
23.68	3.31	Male	No	Sun	Dinner	2
24.59	3.61	Female	Yes	Sun	Lunch	4
25.29	4.71	Male	No	Sun	Lunch	4
8.77	2	Male	No	Sun	Dinner	2
26.88	3.12	Male	No	Sun	Dinner	4
15.04	1.96	Male	Yes	Sun	Dinner	2
14.78	3.23	Male	Yes	Sun	Dinner	2
10.27	1.71	Male	Yes	Sun	Dinner	2
35.26	5	Female	No	Sun	Lunch	4

Sketching

- 1st Sheet: **Idea** collection (4 * 3min)
- **Explanation & Feedback** (10min)
- 2nd Sheet: **Elaboration** (20min)