Session 2.2
Visualization Tools

Online Course
Data Visualization for Professionals

THE UNIVERSITY of EDINBURGH

Benjamin Bach
June 2020
http://benjbach.me
https://datavis-online.github.io

-- Not for external use --

Outline

How to navigate the jungle of visualization tools?

1. Tool landscape
2. Generic & usage
3. Generic & coding
4. Specific applications
5. Tools for presentation and design

Session 2.2
Tool Landscape

Online Course
Data Visualization for Professionals

THE UNIVERSITY of EDINBURGH

Benjamin Bach
June 2020
http://benjbach.me
https://datavis-online.github.io

-- Not for external use --

Some visualization tools

Power BI

Gephi

polestar

Data Illustrator

Graphy

THE VISTORIAN (BETA)

RAWGraphs

mapbox

Data-Driven Documents

Leaflet

i want hue

Processing

Charticulator

tableau

COLORBREWER 2.0
color advice for cartography

Bokeh

plotly

Criteria

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

Choosing a tool?

- **What** can the tool do?
- What **data** and **visualizations**?
- What do I have to **learn**?
- How much do I have to **do**?
- Is the tool **available**?
- What's the **platform?**

Choosing a tool?

- **What** can the tool do?
- What **data** and **visualizations**?
- What do I have to **learn**?
- How much do I have to **do**?
- Is the tool **available**?
- What's the **platform?**

Choosing a tool?

- **What** can the tool do?
- What **data** and **visualizations**?
- What do I have to **learn**?
- How much do I have to **do**?
- Is the tool **available**?
- What's the **platform?**

Choosing a tool?

- **What** can the tool do?
- What **data** and **visualizations**?
- What do I have to **learn**?
- How much do I have to **do**?
- Is the tool **available**?
- What's the **platform?**

Choosing a tool?

- **What** can the tool do?
- What **data** and **visualizations**?
- What do I have to **learn**?
- How much do I have to **do**?
- Is the tool **available**?
- What's the **platform?**

Choosing a tool?

- **What** can the tool do?
- What **data** and **visualizations**?
- What do I have to **learn**?
- How much do I have to **do**?
- Is the tool **available**?
- What's the **platform?**

Visbrowser: www.vistools.net

AVAILABILITY	PROGRAMMING SKILLS	PLATFORM	FEATURES	TYPE OF DATA
13 Free	8 None	7 Apple	6 Web-publishing	15 Numeric
16 Open source	6 Javascript	6 Windows	9 File exports	7 Geographic
7 $ Paid	1 Python	4 Linux	2 Other	10 Temporal
	1 Java 3 Other	10 Web	4 Wizard	0 Text 4 Other
	1 Matlab 2 R	4 Library	1 PDF Export	2 Network

D3.JS

https://d3js.org/
D3.js is a JavaScript library for manipulating documents based on data.

DATAWRAPPER

https://www.datawrapper.de/
Datawrapper is an open-source web tool that you can use to create basic interactive charts.

RAW GRAPHS

https://rawgraphs.io/
RAW Graphs is an open-source data visualization framework designed to make visualising complex data easy for everyone.

RSTUDIO

https://rstudio.com/
RStudio is an integrated development environment for R, a programming language for statistical computing and graphics.

Coding Designing Using

Bokeh

Coding

Designing

Using

Bokeh

tableau

Coding

Designing

Using

Generic
Bokeh
tableau
Coding
Designing
Using
Specific

Generic
Data Illustrator
RAWGraphs
Charticulator
Datawrapper
Power BI
Bokeh
tableau
Coding
Designing
Using
Specific

Generic
Data-Driven Documents
Processing
plotly
Bokeh
unity
Data Illustrator
RAWGraphs
Charticulator
Datawrapper
Power BI
tableau
Coding
Designing
Using
Specific

Generic

Data-Driven Documents

Processing

plotly

Bokeh

unity

Data Illustrator

RAWGraphs

Charticulator

Datawrapper

Power BI

tableau

Coding

Designing

Using

Gephi

THE VISTORIAN (BETA)

COLORBREWER 2.0
color advice for cartography

i want hue

Specific

Generic

Data-Driven Documents

Processing

plotly

Bokeh

unity

Data Illustrator

RAWGraphs

Charticulator

Datawrapper

Power BI

tableau

Coding

Designing

Using

mapbox

Leaflet

Gephi

THE VISTORIAN (BETA)

COLORBREWER 2.0 color advice for cartography

i want hue

Specific

Session 2.2

Tools: Generic + Usage

Online Course
Data Visualization for Professionals

THE UNIVERSITY of EDINBURGH

Benjamin Bach
June 2020
http://benjbach.me
https://datavis-online.github.io

-- Not for external use --

DataWrapper

https://www.datawrapper.de/

By David Alameda for elpais.com

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

https://vega.github.io/polestar

Polestar

https://vega.github.io/polestar

Polestar

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

Tableau

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

http://rawgraphs.io

Session 2.2

Tools: Generic + Coding

Online Course
Data Visualization for Professionals

THE UNIVERSITY of EDINBURGH

Benjamin Bach
June 2020
http://benjbach.me
https://datavis-online.github.io

-- Not for external use --

Python: **Seaborn**

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

Python: **Seaborn**

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)
```

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**

```python
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

2758 movies selected

```
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

Data-Driven Documents

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

d3js.org

Bring your data to life.

343 Published | 3,157 Likes | 1,975 Forks

Notebooks

Collections

Browse by collections

d3-array

d3-axis

d3-brush

d3-chord

d3-color

d3-contour

d3-delaunay

d3-dispatch

d3-drag

Notebooks

Showing 1-30 of 343 notebooks

Prev Next

Radial Area Chart

Mike Bostock in D3

Sep 14 • In d3-shape • 21

Color Legend

Mike Bostock in D3

Sep 2 • In d3-scale • 33

World Choropleth

Mike Bostock in D3

Sep 1 • In d3-geo • 2

U.S. Map with Puerto Rico

Mike Bostock in D3

Aug 31 • In d3-geo • 2

Satellite Explorer

Mike Bostock in D3

Aug 30 • In d3-geo-projection • 13

State Choropleth

Mike Bostock in D3

Aug 24 • In d3-geo • 1

- D3 Live coding (like python notebooks)

https://observablehq.com/

D3 Chart library

https://www.d3-graph-gallery.com/

Steps:

- A ridgeline plot basically displays many densities. You probably want to visit the density section of the gallery if this is new for you.
- Here, two Y axis are built. The first is for density, using `scaleLinear()`. The second for the groups, with `scaleBand()`

← Edit me!

```
<!DOCTYPE html>
<meta charset="utf-8">

<!-- Load d3.js -->
<script src="https://d3js.org/d3.v4.js"></script>

<!-- Create a div where the graph will take place -->
<div id="my_dataviz"></div>
```

```
<script>

// set the dimensions and margins of the graph
var margin = {top: 60, right: 30, bottom: 20, left:110},
    width = 460 - margin.left - margin.right,
    height = 400 - margin.top - margin.bottom;

// append the svg object to the body of the page
var svg = d3.select("#my_dataviz")
  .append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
  .append("g")
    .attr("transform",
          "translate(" + margin.left + "," + margin.top + ")");

//read data
d3.csv("https://raw.githubusercontent.com/zonination/perceptions/master/pr

  // Get the different categories and count them
```

Session 2.2

Tools: Specific

Online Course
Data Visualization for Professionals

THE UNIVERSITY of EDINBURGH

Benjamin Bach
June 2020
http://benjbach.me
https://datavis-online.github.io

-- Not for external use --

Gephi

https://gephi.org

The Vistorian: *vistorian.net*

THE VISTORIAN (BETA)

Interactive Visualizations for Dynamic and Multivariate Networks.
Free, online, and open source.

Visualizations Demo My Session

Visualization Manual Data Formatting Github Contact

The University of Edinburgh · Microsoft Research - Inria Joint Centre · Aviz · Inria · Microsoft Research

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

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

Bump Chart

Time series

Area graph

Time series

Taken from: http://datavisualization.ch

Session 2.2

Tools for Presentation and Design

Online Course
Data Visualization for Professionals

THE UNIVERSITY of EDINBURGH

Benjamin Bach
June 2020
http://benjbach.me
https://datavis-online.github.io

-- Not for external use --

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

Sketch: https://www.sketch.com/

Session 2.2

Tools: Wrap up

Online Course
Data Visualization for Professionals

THE UNIVERSITY of EDINBURGH

Benjamin Bach
June 2020
http://benjbach.me
https://datavis-online.github.io

-- Not for external use --

Generic

Data-Driven Documents

Processing

plotly

Bokeh

unity

Data Illustrator

RAWGraphs

Charticulator

Datawrapper

Power BI

tableau

Coding

Designing

Using

mapbox

Leaflet

Gephi

THE VISTORIAN (BETA)

COLORBREWER 2.0 color advice for cartography

i want hue

Specific

Visbrowser: www.vistools.net

AVAILABILITY	PROGRAMMING SKILLS	PLATFORM	FEATURES	TYPE OF DATA
13 Free	8 None	7 Apple	6 Web-publishing	15 Numeric
16 Open source	6 Javascript	6 Windows	9 File exports	7 Geographic
7 $ Paid	1 Python	4 Linux	2 Other	10 Temporal
	1 Java 3 Other	10 Web	4 Wizard	0 Text 4 Other
	1 Matlab 2 R	4 Library	1 PDF Export	2 Network

D3.JS

https://d3js.org/
D3.js is a JavaScript library for manipulating documents based on data.

DATAWRAPPER

https://www.datawrapper.de/
Datawrapper is an open-source web tool that you can use to create basic interactive charts.

RAW GRAPHS

https://rawgraphs.io/
RAW Graphs is an open-source data visualization framework designed to make visualising complex data easy for everyone.

RSTUDIO

https://rstudio.com/
RStudio is an integrated development environment for R, a programming language for statistical computing and graphics.

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

Andy Kirk

DATA HANDLING | APPLICATIONS | PROGRAMMING | WEB-BASED | QUALITATIVE | MAPPING | SPECIALIST | COLOUR

0 TO 255

ABBYY

ABLE2EXTRACT

ADIOMA

ADOBE AFTER EFFECTS

ADOBE ANIMATE

ADOBE COLOR

ADOBE EDGE

ADOBE ILLUSTRATOR

AESOP STORY ENGINE

AFFINITY DESIGNER

AFFINITY DESIGNER

AI2HTML

ALTERYX

AMAZON QUICKSIGHT

AMCHARTS

ANIMAPS

ANYCHART

APPS FOR EXCEL

ARBOR.JS

ARCGIS

AUTODRAW

AXURE

BALSAMIQ

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

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

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

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

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

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

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

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