scieneers
DRIVEN BY DATA

Visual data exploration with python

Nico Kreiling

scieneers
DRIVEN BY DATA

Hi, I'm Nico Kreiling

- Data Scientist @ Scieneers
- Host of the Techtiefen podcast
- Major focus fields:
  - Natural Language Processing (NLP)
  - Recommender Systems (RecSys)
  - Machine Learning Operations (MLOps)

NicoKreiling

Conference Talk Archive

2 Options for Setup

Only 👩‍💻 on my machine

- Clone https://github.com/krlng/py-dashboarding
- Install dependencies with poetry (recommended) or pip
- Use your favorite notebook (Jupyter, VSCode...)

Go to the ☁️

- Go to https://github.com/krlng/py-dashboarding
- Click on the binder link

scieneers
DRIVEN BY DATA

Goals

- Learning some best practices for EDAs
- Differentiation between Data Visualization and Visual Data Exploration
- An overview and introduction to common plotting libraries
  - Pandas-Plotting-Backend
  - Plotly
  - Altair
- An overview of libraries for interactive dashboards
  - Voila
  - Streamlit
  - Panel

Agenda

- Data Visualisation Basics
- Specialities of Data Exploration
- Plotting Libraries
- Dashboarding Libraries
- Summary

What does Explorative Data Analysis (EDA) mean?

"The EDA approach is [...] an attitude/philosophy about how a data analysis should be carried out."

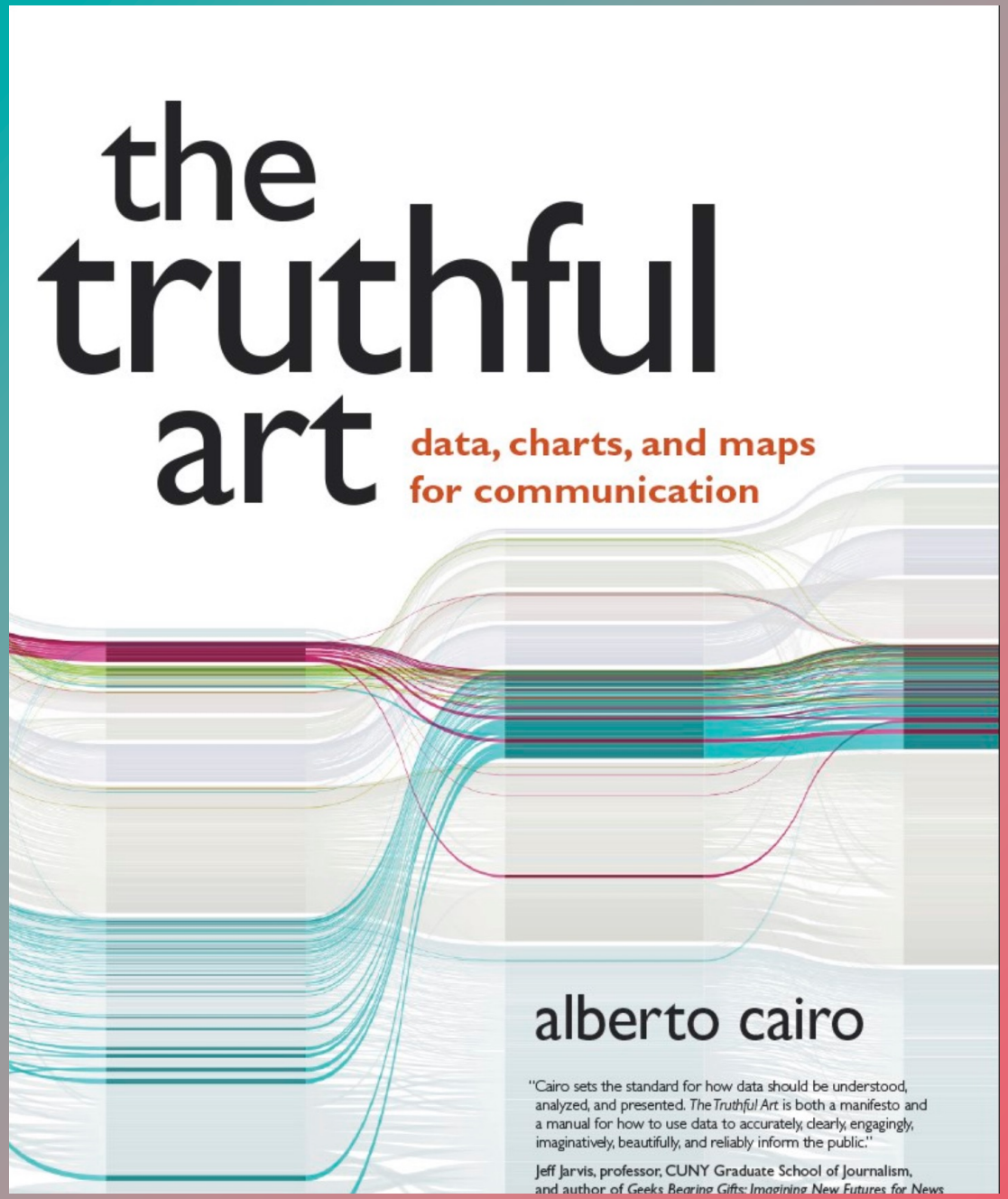
Filliben, 2004

"The role of the data analyst is to listen to the data in as many ways as possible until a plausible "story" of the data is apparent."

Tukey, 1977

John Tukey [1915 - 2000]

https://infovis-wiki.net/wiki/Exploratory_Data_Analysis_(EDA)

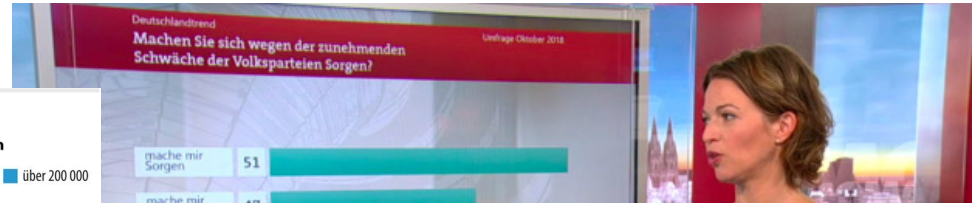


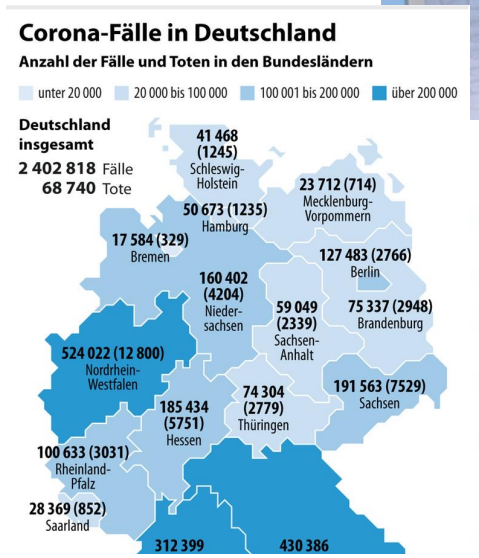
Data Visualization

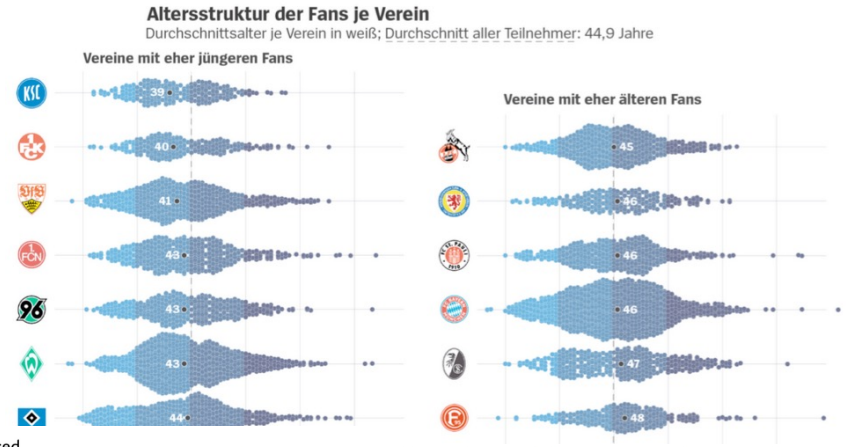
No matter how clever the choice of the information, and no matter how technologically impressive the encoding, a visualization fails if the decoding fails

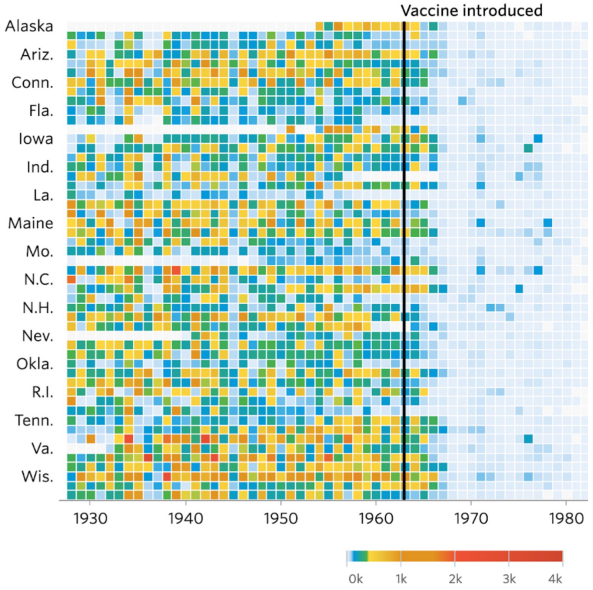
5 Qualities of Data Visualization

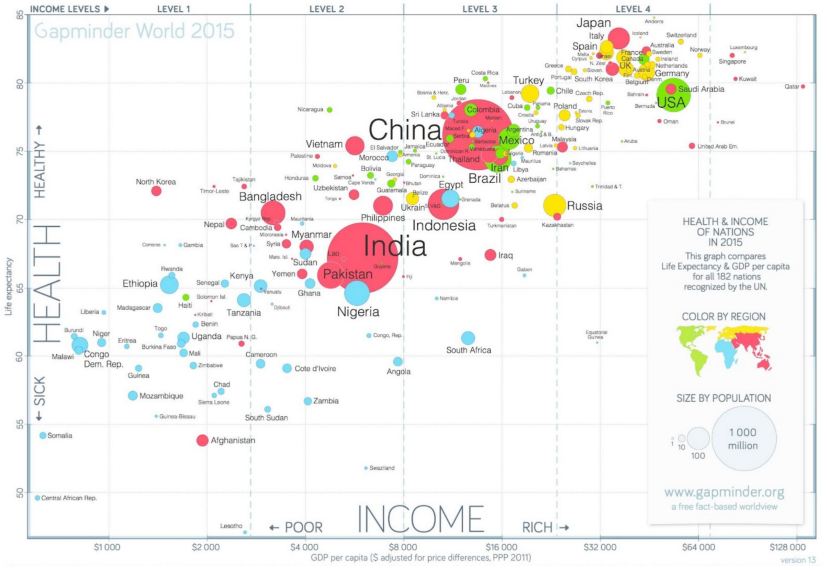
- **Truthful:** based on thorough and honest research.
- **Functional:** constitute an accurate depiction of the data to let people do meaningful operations based on it.
- **Beautiful:** attractive, intriguing, and even aesthetically pleasing for its intended audience
- **Insightful:** Reveals evidence that would be hard to see otherwise
- **Enlightenment:** Great visualizations change people's minds for the better











Data Visualization

Result presentation

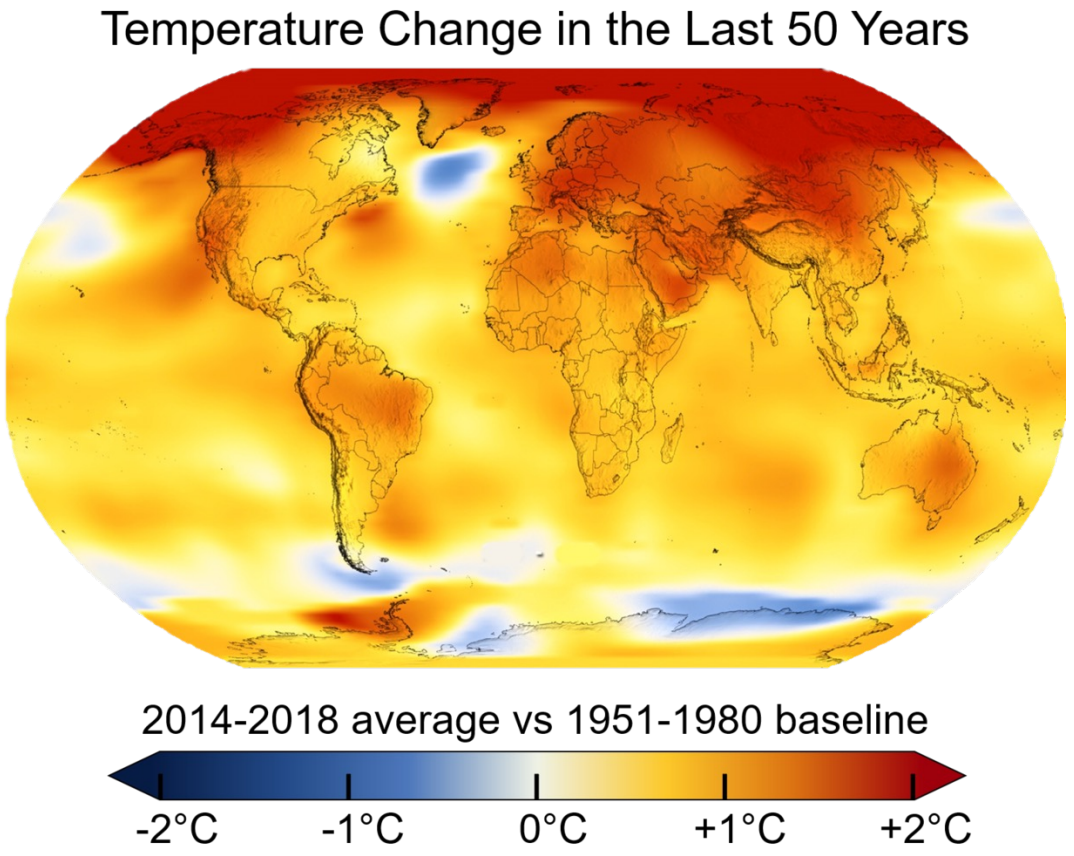
- Which insight should be presented?
- As little information as possible, as many as necessary
- Common use-case:
  - Diverse audience
  - Unkown context
  - Short attention period

Data Exploration

- Which insights can be discovered?
- Highlight data in as many angles as possible
- Common use-cases:
  - Small audience
  - Own workstation
  - Focused analysis

About the Dataset – CO2 emissions worldwide

- Reasons to pick this dataset:
  - Climate change will be (one of) the dominating topics for the upcoming centuries
  - Big enough to discover many insides, but no parallelization required
  - Contains both: temporal and geographic information
- Our world in data site with many visualizations
- Github page with aggregated and aligned data from multiple sources



EDA Checklist

- Dataset size
- Gain an overview of columns
- Check datatypes
- Identify and analyse ID columns
- (Check time- and geographic dimensions)
- Analyse missing values
- Detect outliers

Practice

Notebook 1: Data Validation and Checks

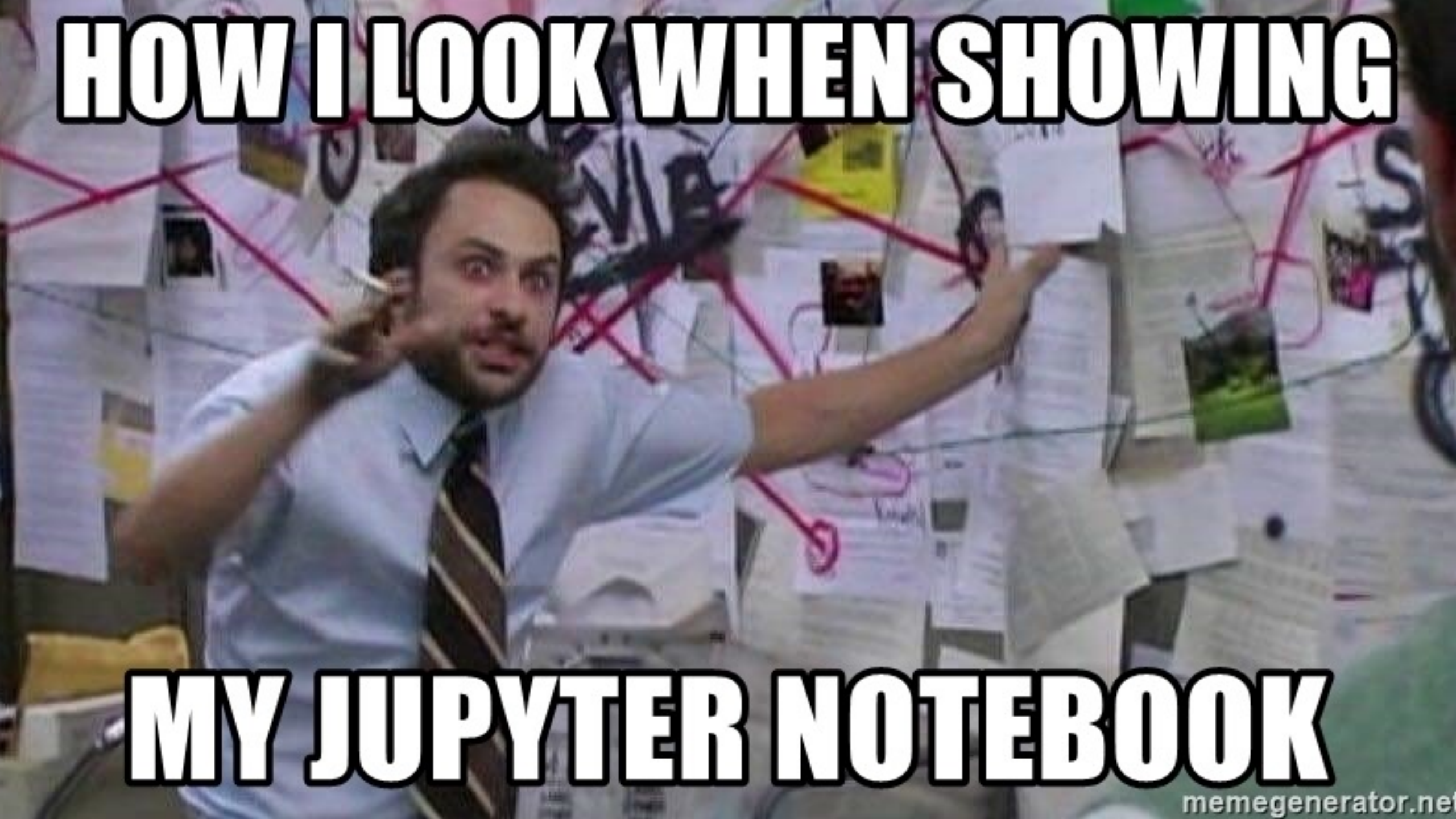
Notebook 2: Relationships and Hierarchy

Notebook 3: Analysing time

Notebook 4: Geographic information

Notebook 5: Feature Discovery

Notebook 6: Dashboards


HOW I LOOK WHEN SHOWING
MY JUPYTER NOTEBOOK
memegenerator.net


UNTITLED25.IPYNB
UNTITLED24.IPYNB

scieneers
DRIVEN BY DATA

Notebook-EDA best practices

- Variable namings:
  - Prevent duplicate names for re-used objects / functions
  - Group by prefix to use auto-completion
- Use Python packages for often used or complicated code
  - Make editable install (pip install –e my-package)
  - Use auto-reloading (%load_ext autoreload %autoreload 2)
- Keep structure
  - Give notebooks proper names
  - Use Markdown Headings for structure within notebooks
  - Place shared imports / dataload in a common notebook that is part of every notebook using the %run magic

scieneers
DRIVEN BY DATA

Visualizations – Techniques & Tools

Visual vocabulary

Deviation

Emphasise variations (+/-) from a fixed reference point. Typically the reference point is zero but it can also be a target or a long-term average. Can also be used to show sentiment (positive/neutral/negative).

Example FT uses
Trade surplus/deficit, climate change

Diverging bar

A simple standard bar chart that can handle both negative and positive magnitude values.

Diverging stacked bar

Perfect for presenting survey results which involve sentiment (eg disagree/neutral/agree).

Spine

Splits a single value into two contrasting components (eg male/female).

Surplus/deficit filled line

The shaded area of these charts allows a balance to be shown – either against a baseline or between two series.

Correlation

Show the relationship between two or more variables. Be mindful that, unless you tell them otherwise, many readers will assume the relationships you show them to be causal (i.e. one causes the other).

Example FT uses
Inflation and unemployment, income and life expectancy

Scatterplot

The standard way to show the relationship between two continuous variables, each of which has its own axis.

Column + line timeline

A good way of showing the relationship between an amount (columns) and a rate (line).

Connected scatterplot

Usually used to show how the relationship between 2 variables has changed over time.

Bubble

Like a scatterplot, but adds additional detail by sizing the circles according to a third variable.

XY heatmap

A good way of showing the patterns between 2 categories of data, less effective at showing fine differences in amounts.

Ranking

Use where an item's position in an ordered list is more important than its absolute or relative value. Don't be afraid to highlight the points of interest.

Example FT uses
Wealth, deprivation, league tables, constituency election results

Ordered bar

Standard bar charts display the ranks of values much more easily when sorted into order.

Ordered column

See above.

Ordered proportional symbol

Use when there are big variations between values and/or seeing fine differences between data is not so important.

Dot strip plot

Dots placed in order on a strip are a space-efficient method of laying out ranks across multiple categories.

Slope

Perfect for showing how ranks have changed over time or vary between categories.

Lollipop

Lollipops draw more attention to the data value than standard bar/column and can also show rank and value effectively.

Bump

Effective for showing changing rankings across multiple dates. For large datasets, consider grouping lines using colour.

Distribution

Show values in a dataset and how often they occur. The shape (or 'skew') of a distribution can be a memorable way of highlighting the lack of uniformity or equality in the data.

Example FT uses
Income distribution, population (age/sex) distribution, revealing inequality

Histogram

The standard way to show a statistical distribution - keep the gaps between columns small to highlight the 'shape' of the data.

Dot plot

A simple way of showing the change or range (min/max) of data across multiple categories.

Dot strip plot

Good for showing individual values in a distribution, can be a problem when too many dots have the same value.

Barcode plot

Like dot strip plots, good for displaying all the data in a table, they work best when highlighting individual values.

Boxplot

Summarise multiple distributions by showing the median (centre) and range of the data

Violin plot

Similar to a box plot but more effective with complex distributions (data that cannot be summarised with simple average).

Population pyramid

A standard way for showing the age and sex breakdown of a population distribution; effectively, back to back histograms.

Cumulative curve

A good way of showing how unequal a distribution is: y axis is always cumulative frequency, x axis is always a measure.

Frequency polygons

For displaying multiple distributions of data. Like a regular line chart, best limited to a maximum of 3 or 4 datasets.

Beeswarm

Use to emphasise individual points in a distribution. Points can be sized to an additional variable. Best with medium-sized datasets

Change over Time

Give emphasis to changing trends. These can be short (intra-day) movements or extended series traversing decades or centuries: Choosing the correct time period is important to provide suitable context for the reader.

Example FT uses
Share price movements, economic time series, sectoral changes in a market

Line

The standard way to show a changing time series. If data are irregular, consider markers to represent data points.

Column

Columns work well for showing change over time - but usually best with only one series of data at a time.

Column + line timeline

A good way of showing the relationship over time between an amount (columns) and a rate (line).

Slope

Good for showing changing data as long as the data can be simplified into 2 or 3 points without missing a key part of story.

Area chart

Use with care – these are good at showing changes to total, but seeing change in components can be very difficult.

Candlestick

Usually focused on day-to-day activity, these charts show opening/closing and high/low points of each day.

Fan chart (projections)

Use to show the uncertainty in future projections - usually this grows the further forward to projection.

Connected scatterplot

A good way of showing changing data for two variables whenever there is a relatively clear pattern of progression.

Calendar heatmap

A great way of showing temporal patterns (daily, weekly, monthly) – at the expense of showing precision in quantity.

Priestley timeline

Great when date and duration are key elements of the story in the data.

Magnitude

Show size comparisons. These can be relative (just being able to see larger/bigger) or absolute (need to see fine differences). Usually these show a 'counted' number (for example, barrels, dollars or people) rather than a calculated rate or per cent.

Example FT uses
Commodity production, market capitalisation, volumes in general

Column

The standard way to compare the size of things. Must always start at 0 on the axis.

Bar

See above. Good when the data are not time series and labels have long category names.

Paired column

As per standard column but allows for multiple series. Can become tricky to read with more than 2 series.

Paired bar

See above.

Marimekko

A good way of showing the size and proportion of data at the same time – as long as the data are not too complicated.

Proportional symbol

Use when there are big variations between values and/or seeing fine differences between data is not so important.

Isotype (pictogram)

Excellent solution in some instances – use only with whole numbers (do not slice off an arm to represent a decimal).

Lollipop

Lollipop charts draw more attention to the data value than standard bar/column – does not have to start at zero (but preferable).

Radar

A space-efficient way of showing value of multiple variables– but make sure they are organised in a way that makes sense to reader.

Parallel coordinates

An alternative to radar charts – again, the arrangement of the variables is important. Usually benefits from highlighting values.

Part-to-whole

Show how a single entity can be broken down into its component elements. If the reader's interest is solely in the size of the components, consider a magnitude-type chart instead.

Example FT uses
Fiscal budgets, company structures, national election results

Stacked column/bar

A simple way of showing part-to-whole relationships but can be difficult to read with more than a few components.

Marimekko

A good way of showing the size and proportion of data at the same time – as long as the data are not too complicated.

Pie

A common way of showing part-to-whole data – but be aware that it's difficult to accurately compare the size of the segments.

Donut

Similar to a pie chart – but the centre can be a good way of making space to include more information about the data (eg total).

Treemap

Use for hierarchical part-to-whole relationships; can be difficult to read when there are many small segments.

Voronoi

A way of turning points into areas – any point within each area is closer to the central point than any other centroid.

Arc

A hemicycle, often used for visualising parliamentary composition by number of seats.

Gridplot

Good for showing % information, they work best when used on whole numbers and work well in small multiple layout form.

Venn

Generally only used for schematic representation.

Waterfall

Can be useful for showing part-to-whole relationships where some of the components are negative.

Spatial

Aside from locator maps only used when precise locations or geographical patterns in data are more important to the reader than anything else.

Example FT uses
Population density, natural resource locations, natural disaster risk/impact, catchment areas, variation in election results

Basic choropleth (rate/ratio)

The standard approach for putting data on a map – should always be rates rather than totals and use a sensible base geography.

Proportional symbol (count/magnitude)

Use for totals rather than rates – be wary that small differences in data will be hard to see.

Flow map

For showing unambiguous movement across a map.

Contour map

For showing areas of equal value on a map. Can use deviation colour schemes for showing +/- values

Equalised cartogram

Converting each unit on a map to a regular and equally-sized shape – good for representing voting regions with equal value.

Scaled cartogram (value)

Stretching and shrinking a map so that each area is sized according to a particular value.

Dot density

Used to show the location of individual events/locations – make sure to annotate any patterns the reader should see.

Heat map

Grid-based data values mapped with an intensity colour scale. As choropleth map – but not snapped to an admin/political unit.

Flow

Show the reader volumes or intensity of movement between two or more states or conditions. These might be logical sequences or geographical locations.

Example FT uses
Movement of funds, trade, migrants, lawsuits, information; relationship graphs.

Sankey

Shows changes in flows from one condition to at least one other; good for tracing the eventual outcome of a complex process.

Waterfall

Designed to show the sequencing of data through a flow process, typically budgets. Can include +/- components.

Chord

A complex but powerful diagram which can illustrate 2-way flows (and net winner) in a matrix.

Network

Used for showing the strength and inter-connectedness of relationships of varying types.

https://raw.githubusercontent.com/Financial-Times/chart-doctor/main/visual-vocabulary/Visual-vocabulary.pdf

scieneers
DRIVEN BY DATA

Two approaches to data visualization APIs

- **Imperative approach**: specifies the **"how"**
  - e.g. matplotlib
  - Pros: **fine-graned control** over chart specification
  - Cons: **Verbose**, makes you think about low-level visualization details

- **Declarative approach**: specifies the **"what"**
  - e.g., altair, plotnine (ggplot2)
  - Pros: fast development, conciseness, and lets you **concentrate on visualization content**
  - Cons: **less control** over chart details, trust someone elses decisions about low level graph properties

scieneers
DRIVEN BY DATA

Tooling

Core

Python libraries on which multiple higher-level libraries are built.

Name	Stars	Contributors	Downloads	License	Docs	PyPI	Conda	Sponsors	Built on
matplotlib	16k	427	pypi 28M/month, conda 1M/month	PSF	up	v3.5.2	anaconda v3.5.1	NUMFOCUS	matplotlib
plotly.py	12k	176	pypi 7.5M/month, conda 229k/month	MIT	up	v5.8.0	anaconda v5.6.0	plotly	plotly
bokeh	16k	393	pypi 2.5M/month, conda 541k/month	BSD-3-Clause	up	v2.4.3	anaconda v2.4.2	NUMFOCUS ANACONDA.	bokeh

High-Level

InfoVis Libraries focusing on high-level operations for working with data visually, built upon the core Python or JS libraries.

Name	Stars	Contributors	Downloads	License	Docs	PyPI	Conda	Sponsors	Built on
seaborn	9.5k	148	pypi 9.5M/month, conda 373k/month	BSD (3-clause)	up	v0.11.2	anaconda v0.11.2	-	matplotlib
altair	7.5k	120	pypi 8.4M/month, conda 45k/month	BSD 3-clause	up	v4.2.0	conda-forge v4.2.0	-	
holoviews	2.2k	109	pypi 310k/month, conda 58k/month	BSD	up	v1.14.9	anaconda v1.14.8	ANACONDA.	bokeh matplotlib plotly
plotly_express	669	6	pypi 189k/month, conda 2k/month	MIT	up	v0.4.1	conda-forge v0.4.1	plotly	plotly
chartify	3.2k	15	pypi 8.6k/month, conda 199/month	Apache 2	up	v3.0.3	conda-forge v3.0.3	Spotify	bokeh

Source: https://pyviz.org/tools#dashboarding

scieneers
DRIVEN BY DATA

plotly | Express

Plotly Express

- High-Level API to Plotly
- Enables easy creation of some commonly used, very powerful plots
- Personal Highlights:
  - Nice coloring by default
  - Nice interactive features
  - Animation-frame optiojn for some charts

https://plotly.com/python/plotly-express/#gallery

scieneers
DRIVEN BY DATA

How does plotley Express work?

- Similar to pandas / seaborn API
- `px.<chart-type>(`

```
    data_frame=None,
    x=None,
    y=None,
    other-chart-type-specific-attributes
)
```

Practice

Notebook 1: Data Validation and Checks

Notebook 2: Relationships and Hierarchy

Notebook 3: Analysing time

Notebook 4: Geographic information

Notebook 5: Feature Discovery

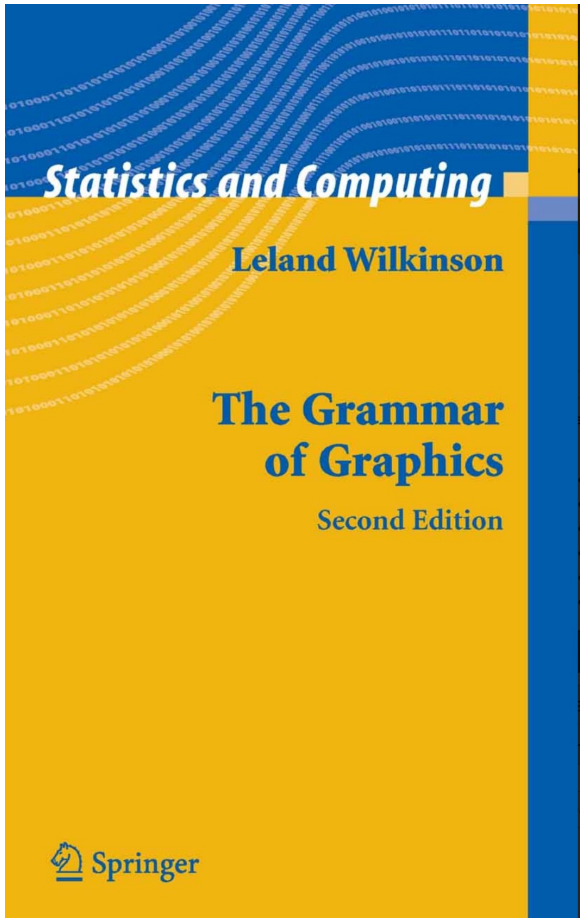
Notebook 6: Dashboards

scieneers
DRIVEN BY DATA

Altair

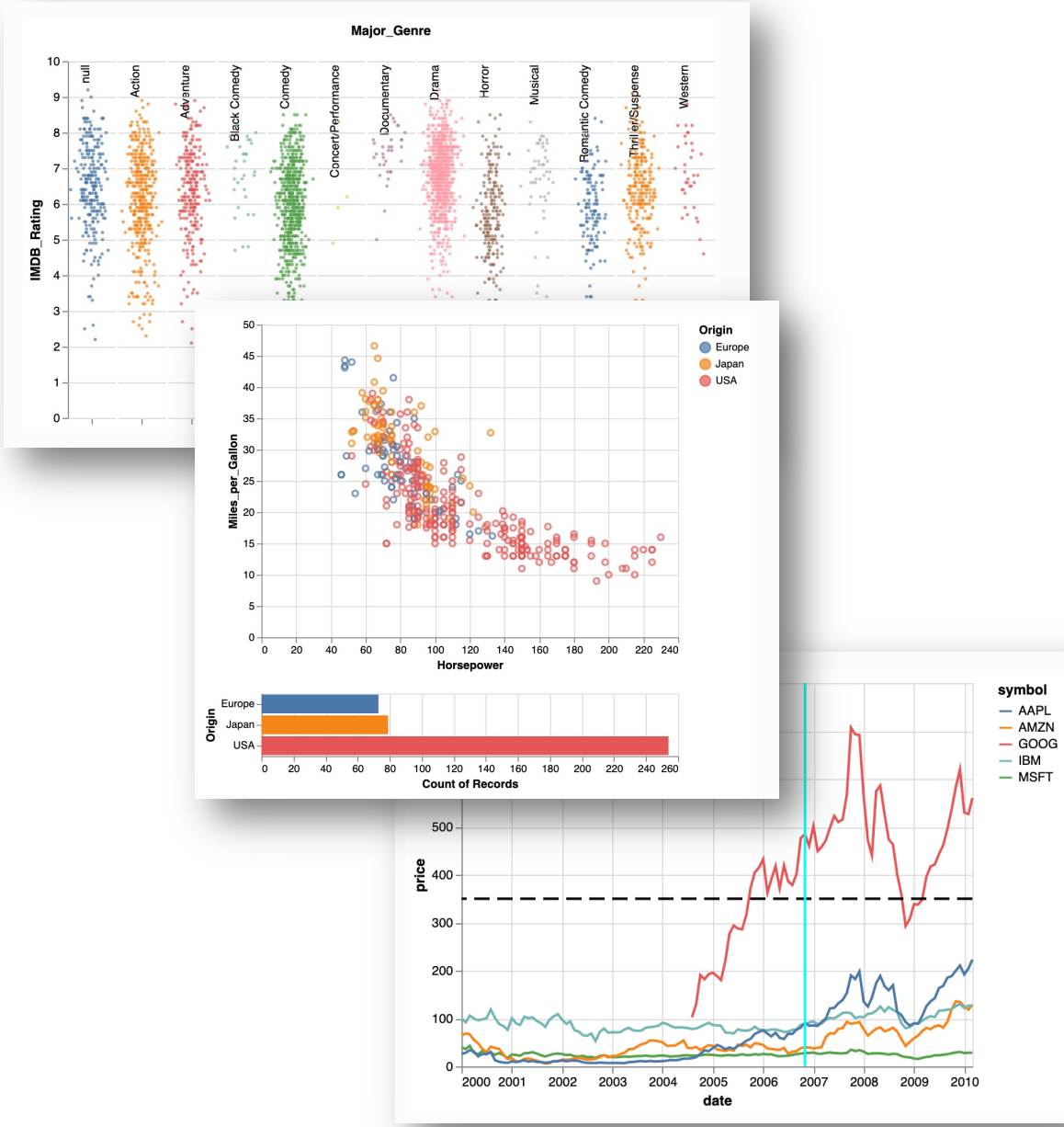
Excurse: Grammar of graphics:

- Mathematical foundations for statistical graphics
- Proposed a grammar to formalize his idea
- Grammar of graphics: a system of rules and primitives to generate meaningful perceivable graphics
- Later adapted by Hadley Wickham - the author of ggplot - as layered grammar of graphics



Altair

- Python implementation of the grammer of graphics
- Well-structured and powerful way to create flexible charts
- Uses VegaLight and Vega under the hood
- Personal Highlights:
  - Easy Datatype Encoding
  - Plug- and adaptable charts
  - Selections, Bindings and Filters



Altair Chart Declaration

```
alt.Chart(countries, width=1000).mark_line().encode(
    x=alt.X("year", scale=alt.Scale(zero=False)),
    y=alt.Y(field='co2', aggregate='sum', type='quantitative'),
    color="continent:N"
)
```

Data

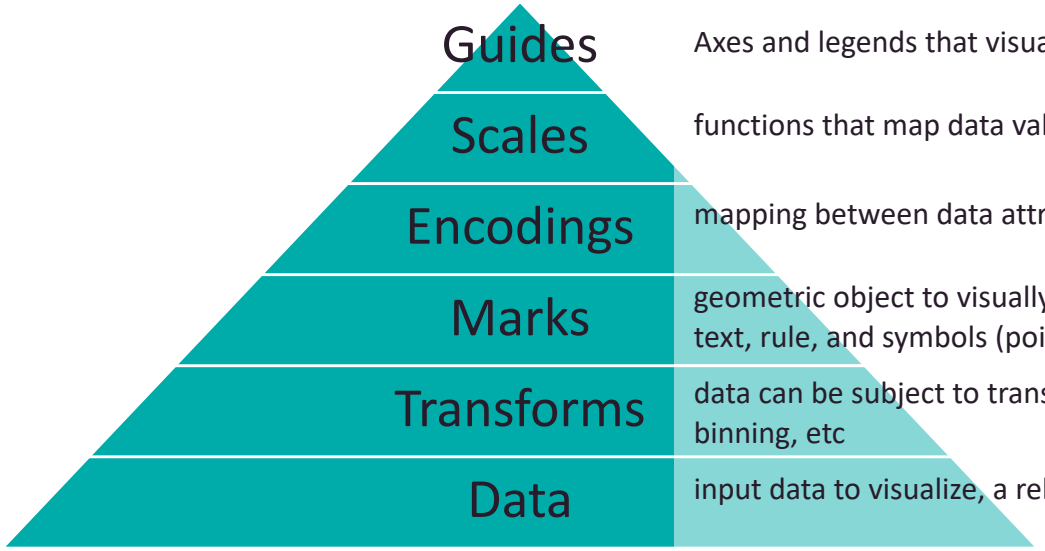
Marks

Scales

Transforms

Encodings

Guides



Axes and legends that visualize scales (ordinal, quantitative, nominal)

functions that map data values to visual values

mapping between data attributes (cols) and properties of visual marks

geometric object to visually encode data-records such as bar, line, area, text, rule, and symbols (point and tick)

data can be subject to transformations such as filter, aggregation, binning, etc

input data to visualize, a relational table consisting of rows and columns.

Practice

Notebook 1: Data Validation and Checks

Notebook 2: Relationships and Hierarchy

Notebook 3: Analysing time

Notebook 4: Geographic information

Notebook 5: Feature Discovery

Notebook 6: Dashboards

Practice

Notebook 1: Data Validation and Checks

Notebook 2: Relationships and Hierarchy

Notebook 3: Analysing time

Notebook 4: Geographic information

Notebook 5: Feature Discovery

Notebook 6: Dashboards

Practice

Notebook 1: Data Validation and Checks

Notebook 2: Relationships and Hierarchy

Notebook 3: Analysing time

Notebook 4: Geographic information

Notebook 5: Feature Discovery

Notebook 6: Dashboards

scieneers
DRIVEN BY DATA

Dashboards

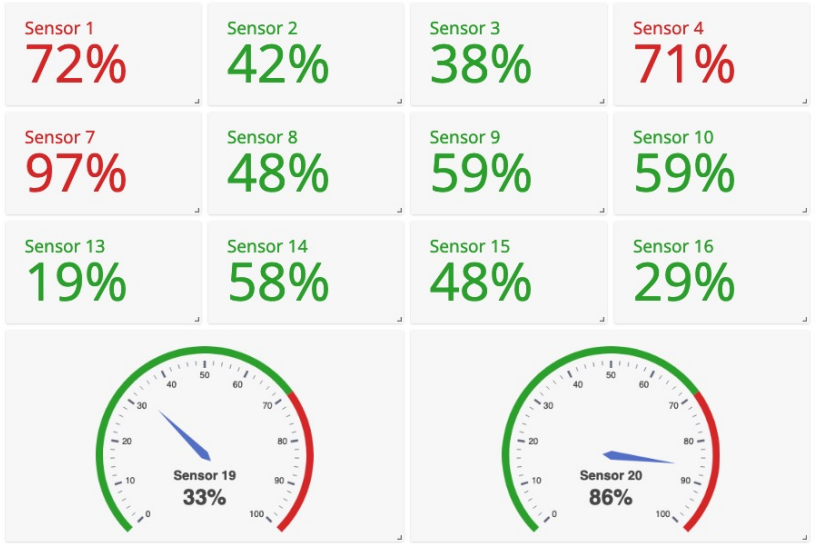
What is a Dashboard?

A **dashboard** is a type of graphical user interface which often provides at-a-glance views of key performance indicators (KPIs) relevant to a particular objective or business process.

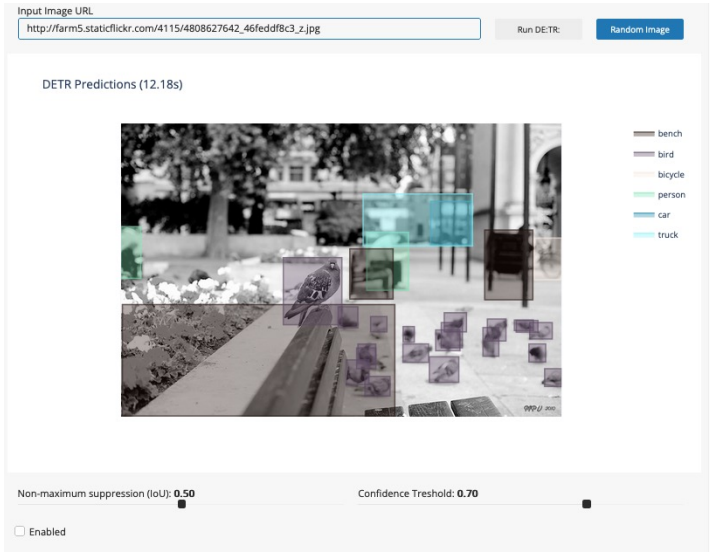
In other usage, "dashboard" is another name for "progress report" or "report" and considered a form of data visualization. In providing this overview, business owners can save time and improve their decision making by utilizing dashboards.

https://en.wikipedia.org/wiki/Dashboard_(business)

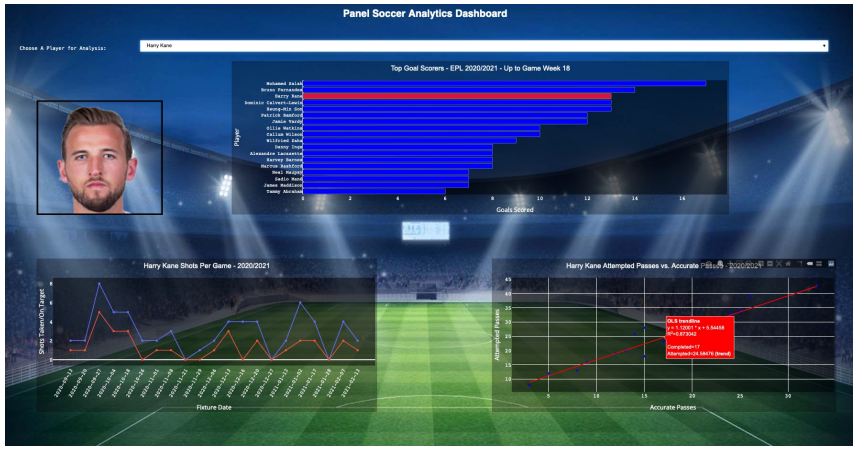
Use Cases for Dashboards



Monitoring Streaming Data



Interactive (ML) Demos



Analytical Dashboards

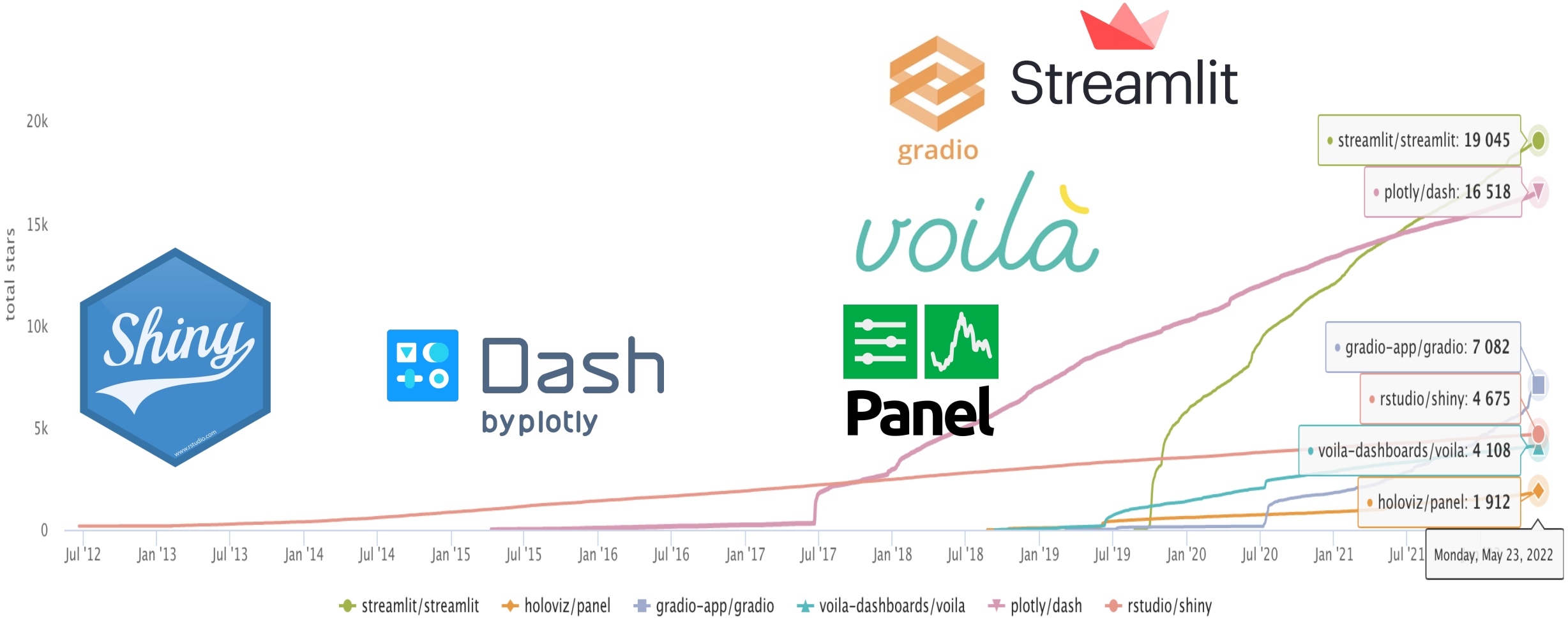
scieneers
DRIVEN BY DATA

Notebooks / Dashboards Pro's and Con's

	Notebooks are great at…	Notebooks fail…
Dashboards +	Combining data with general information (markdown) Data visualizations Data exploration	To ensure reproducability Providing a restricted access to data For non-technical users
Dashboards -	As interactive computing environment teaching to code	For proper software engineering

scieneers
DRIVEN BY DATA

What dashboards to exist



scieneers
DRIVEN BY DATA

Which dashboard tools are out there?

Name	Stars	Contributors	Downloads	License	Sponsors	Built on
bokeh	16k	393	pypi 3M/month · conda 541k/month	BSD-3-Clause	NUMFOCUS, ANACONDA.	bokeh
dash	17k	98	pypi 862k/month · conda 26k/month	MIT	plotly	plotly
streamlit	19k	129	pypi 1M/month · conda 6k/month	Apache 2	Streamlit	-
panel	1.9k	93	pypi 464k/month · conda 63k/month	BSD	ANACONDA.	bokeh
gradio	7k	60	pypi 265k/month · conda 0/month	Apache License 2.0	Hugging Face	-
visdom	9.1k	94	pypi 64k/month · conda 1k/month	CC-BY-NC-4.0	-	plotly
voila	4.1k	55	pypi 39k/month · conda 5k/month	BSD	QuantStack	-

Source: https://pyviz.org/tools#dashboarding

scieneers
DRIVEN BY DATA

scieneers
DRIVEN BY DATA

Voila

- Open Source library to transform any Jupyter notebook into a dashboard
- Also works for some non-python kernels
- Mostly used together with Jupyter widgets
- Minimal learning curve
- Sponsored by QuantStack

scieneers
DRIVEN BY DATA

Voila Workflow

- Create a notebook using ipython-widgets
- Convert the notebook into a dashboard using `voila notebook.ipynb`

scieneers
DRIVEN BY DATA

Streamlit

Many examples at Awesome streamlit

- Python library (open source)
- But also a start-up which was aquired by snowflake recently
- Easy to use with enhanced caching and storage options
- Provides an own cloud for deployment

Streamlit workflow

- Create a dashboard.py file and `import streamlit as st`
- Use streamlit components for inputs and outputs
- Use `streamlit run dashboard.py`

scieneers
DRIVEN BY DATA

Panel

scieneers
DRIVEN BY DATA

Panel

Many examples at Awesome panel

- Python library on top of param
- Allows usage within a notebook
- Provides multiple APIs
  - Interact: automatifcally create UI controls by using type inference
  - Widgets, Templates & Pipelines: Fine grained control to build complicated dashboards
- Part of a bigger eco-system, fully sponsored by Anaconda

HoloViz-maintained libraries

scieneers
DRIVEN BY DATA

Panel Workflow

- Use-case 1: Easy Notebook usage
  - Call `pn.extension()`
  - Use `pn.interact` to transform any function into a minimal dashboard
  - Use `pn.serve()` to start a full-screen dashboard
- Use-case 2: Create a production dashboard
  - Create a file such as dashboard.py and import panel
  - Create a class that inherits from `param.Parameterized`
  - Instantiate that class
  - Run `panel serve app.py`

scieneers
DRIVEN BY DATA

Comparison & Summary

scieneers
DRIVEN BY DATA

Comparison

	Voila	Streamlit	Panel
Programming Languages	Python, C++, Julia	Python	Python
Notebook Support	Yes	No	Yes
Design flexiablity	Templating system, but not the main focus	Limited but easy theme editor	Powerfull Templating system using Bootstrap / Material-Design
Deployment	Documentation for multiple providers (GCP, Binder, Heroku..)	Streamlit cloud	Documentation for multiple providers (Azure, GCP, Binder, Anaconda, Herokus..)
Community + Support	Popular, but bit less active	Very popular and active	Small, very personal community
Strength	Multi-Language Support Tight Jupyter Coupling	Ease-of-use Beautiful by default	Very flexibel Multi-Page-Support

Practice

Notebook 1: Data Validation and Checks

Notebook 2: Relationships and Hierarchy

Notebook 3: Analysing time

Notebook 4: Geographic information

Notebook 5: Feature Discovery

Notebook 6: Dashboards

More information

- Read: A detailed comparison of Voila, Streamlit, Panel and Dash
- Video PyData Workshop with the authors of those Dash, Voila, Panel and Streamlit
- Code can be found at: https://github.com/krlng/py-dashboarding/tree/main/training

- Or reach out to me:


NicoKreiling


Nico Kreiling

Listen to my podcast Techtiefen