

2025/5/16

PyData.Fukuoka Meetup #24

Tableau から **PyGWalker** への移行を検討してみた



Satoshi Koda

ec22s @ Connpass, GitHub, Instagram

Ocean Solution Technology Co.,Ltd.



話したいこと

PyGWalker って何

可視化のスコープ

Alternative になる？

- Kanaries社
- Graphic Walker
- PyGWalker
- 自分と会社のこと
- Tableauの問題点*
- 組織内 or 顧客向け
- EDA or データViz
- 作れるグラフ
- 総合的に見て

* 個人の主観です

PyGWalker って何

可視化のスコープ

Alternative になる？



The screenshot shows the GitHub repository page for 'Kanaries'. The README.md file contains the following content:

Welcome to Kanaries Open Source Community 🎉

Kanaries is an open source startup that focus on data exploration and data visualization. We build softwares that automates the workflow of data exploration to help users to understand data better and faster.

Open Source Family

Solution	Description	Image
Graphic Walker	Embeddable analytic solution, embed an interactive visual analytic component in your system.	
PyGWalker	Turn your data into an interactive UI for data exploration. It allows you to visualize your data with simple drag-and-drop operations.	

Kanaries <https://github.com/Kanaries>

"an open source startup that focus on data exploration and data visualization"

→ Graphic Walker

"a different open-source alternative to Tableau"

Reactコンポーネント

→ PyGWalker

'an abbreviation of "Python binding of Graphic Walker"'

Graphic Walker を Python の各種フレームワークに組み込むもの

→ GWalkR, Rath, etc.

PyGWalker って何

可視化のスコープ

Alternative になる？



ec22s / docker-gwalker

Code Issues Pull requests Actions Projects Security Insights Settings

Files

main Go to file

Compose file

Code Blame 54 lines (49 loc) · 1.46 KB

```
1 # resources
2 # https://github.com/Kanaries/pygwalker > examples
3 # https://github.com/Kanaries/pygwalker-in-streamlit
4
5 # build: docker-compose build --no-cache
6 # start: docker-compose up -d
7 # open:
8 # http://localhost:8001/pyg_html/test0 最もシンプル
9 # http://localhost:8888 (jupyter) tokenは起動時コンソールに表示される
10 # http://localhost:2718 (marimo)
11 # http://localhost:8501 (streamlit) 色々あり
12 # end: docker-compose down
13 # remove: docker rmi -f docker-pygwalker-image && docker buildx his
14
15 services:
16   image:
17     build: .
```

コンテナ作りました

<https://github.com/ec22s/docker-gwalker>

Pythonパッケージのインストールで
gccがないとエラー出たら
slimでないイメージ使って下さい

Dockerfile

Code Blame 26 lines (23 loc) · 771 Bytes

```
1 # ref. https://github.com/0msys/streamlit-template-devcontainer
2
3 FROM python:3.12-slim as base
4 # FROM python:3.12 as base
5 # for 'gcc': No such file or directory
6 ARG USERNAME=pyuser
7 ARG USER_UID=1000
8 ARG USER_GID=$USER_UID
```

PyGWalker って何

可視化のスコープ

Alternative になる？



Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Container CPU usage 0.58% / 800% (8 CPUs available)

Container memory usage 1.65GB / 7.47GB

Search Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)
<input type="checkbox"/>	gwalker	-	-	-
<input type="checkbox"/>	dev	60ae081e1d7b	gwalker-image	2002:2002
<input type="checkbox"/>	pygwalker	-	-	-
<input type="checkbox"/>	web_server_demo	1d6c075960d7	pygwalker-image	8001:8001
<input type="checkbox"/>	jupyter_demo	010952949c9f	pygwalker-image	8888:8888
<input type="checkbox"/>	marimo_demo	4699a84bb49a	pygwalker-image	2718:2718
<input type="checkbox"/>	streamlit_demo	101be4582ffc	pygwalker-image	8501:8501

PyGWalkerのコンテナは4つ起動します

<https://github.com/ec22s/docker-gwalker>

Pythonのslimイメージが使えない場合、
イメージサイズは2GBになります

Images [Give feedback](#)

View and manage your local and Docker Hub images. [Learn more](#)

Local Docker Hub repositories

2.55 GB / 0 Bytes in use 2 images Last refresh

Search

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size
<input type="checkbox"/>	pygwalker-image	latest	52a006a5df65	14 hours ago	2 GB
<input type="checkbox"/>	gwalker-image	latest	63be0c15bce9	15 hours ago	547.13 MB

PyGWalker って何

可視化のスコープ

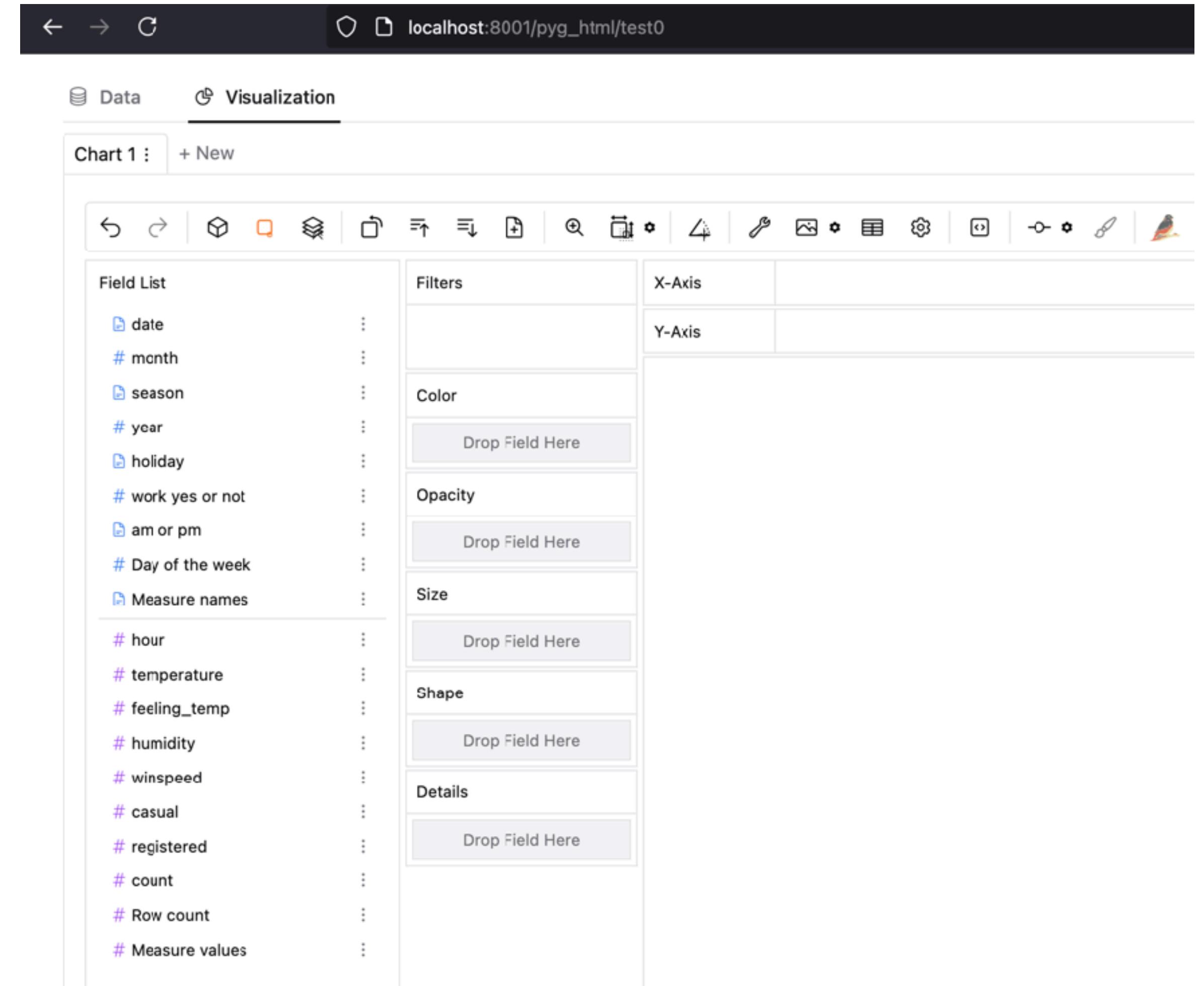
Alternative になる？



pygwalker			
web_server_demo	1d6c075960d7	pygwalker-image	8001:8001
jupyter_demo	010952949c9f	pygwalker-image	8888:8888
marimo_demo	4699a84bb49a	pygwalker-image	2718:2718
streamlit_demo	101be4582ffc	pygwalker-image	8501:8501

ポート8001のコンテナ：最もシンプルなPyGWalker画面

- Tableauに似たUI
- 最左部にデータの各列（メジャーネーム／メジャーバリュー）
- 各列 → X軸／Y軸にD&Dしてグラフを作成
- 各列 → フィルタや色にD&Dして詳細を設定



PyGWalker って何

可視化のスコープ

Alternative になる？



pygwalker			
web_server_demo	1d6c075960d7	pygwalker-image	8001:8001 ↗
jupyter_demo	010952949c9f	pygwalker-image	8888:8888 ↗
marimo_demo	4699a84bb49a	pygwalker-image	2718:2718 ↗
streamlit_demo	101be4582ffc	pygwalker-image	8501:8501 ↗

ポート8888のコンテナ : Jupyter Labへ組み込み

- pygwalkerをインポート
- walkメソッドにpandasのデータフレームを渡せば
データ読み込み済みのPyGWalkerが表示される
- セル内のPyGWalkerでグラフ作成などが可能

The screenshot shows a Jupyter Lab environment with the URL `localhost:8888/doc/tree/jupyter_demo.ipynb`. A file named `jupyter_demo.ipynb` is selected in the sidebar. The main area contains Python code for reading a CSV file and using PyGWalker:

```
[1]: import pygwalker as pyg
import pandas as pd

df = pd.read_csv("/data/bike_sharing_dc.csv")

walker = pyg.walk(df, spec="./gw_config.json")
```

Below the code, there are tabs for "Data", "Visualization", and "Chat". The "Visualization" tab is active, showing a configuration panel for creating a chart. The "Field List" section lists various columns from the dataset: date, # month, # season, # year, # holiday, # work yes or not, # am or pm, # Day of the week, Measure names, # hour, # temperature, and # feeling_temp. To the right, there are sections for "Filters", "X-Axis", "Y-Axis", "Color", "Opacity", "Size", and "Shape".

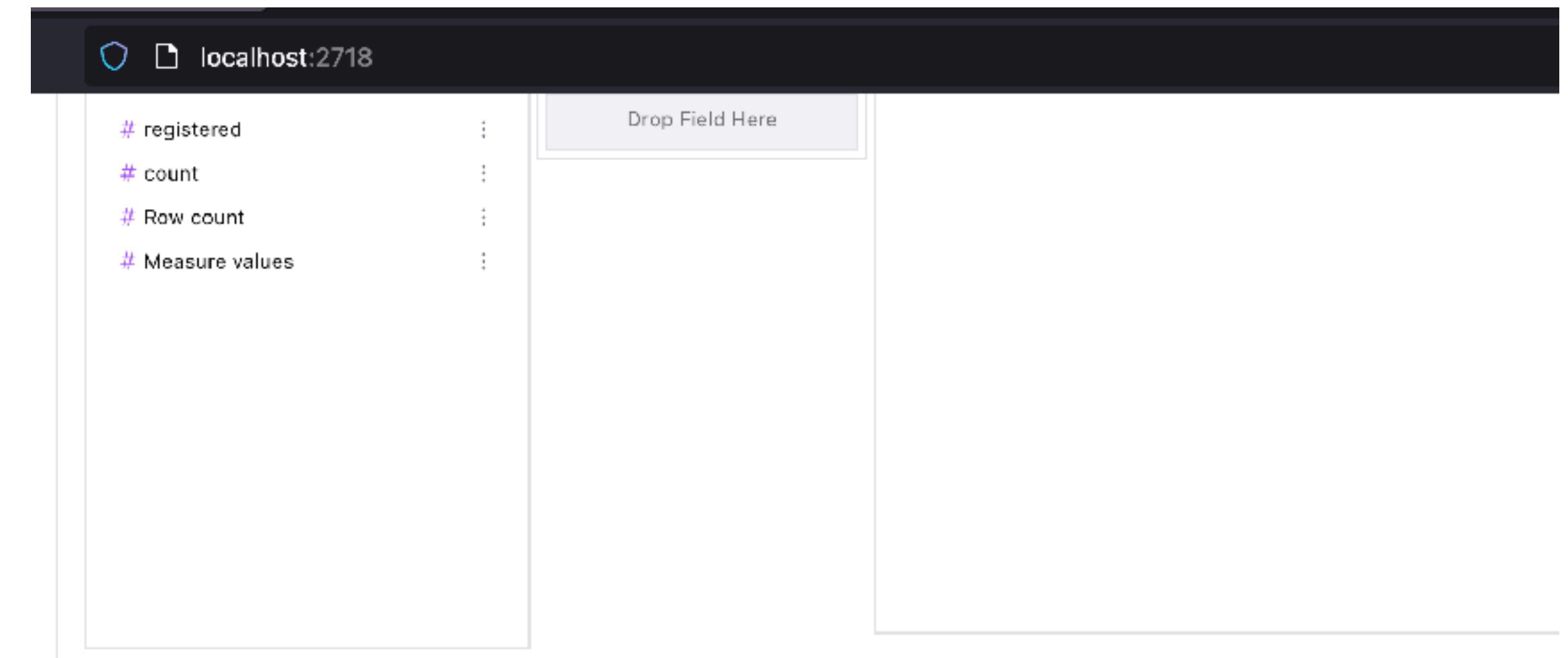
PyGWalker って何

可視化のスコープ

Alternative になる？

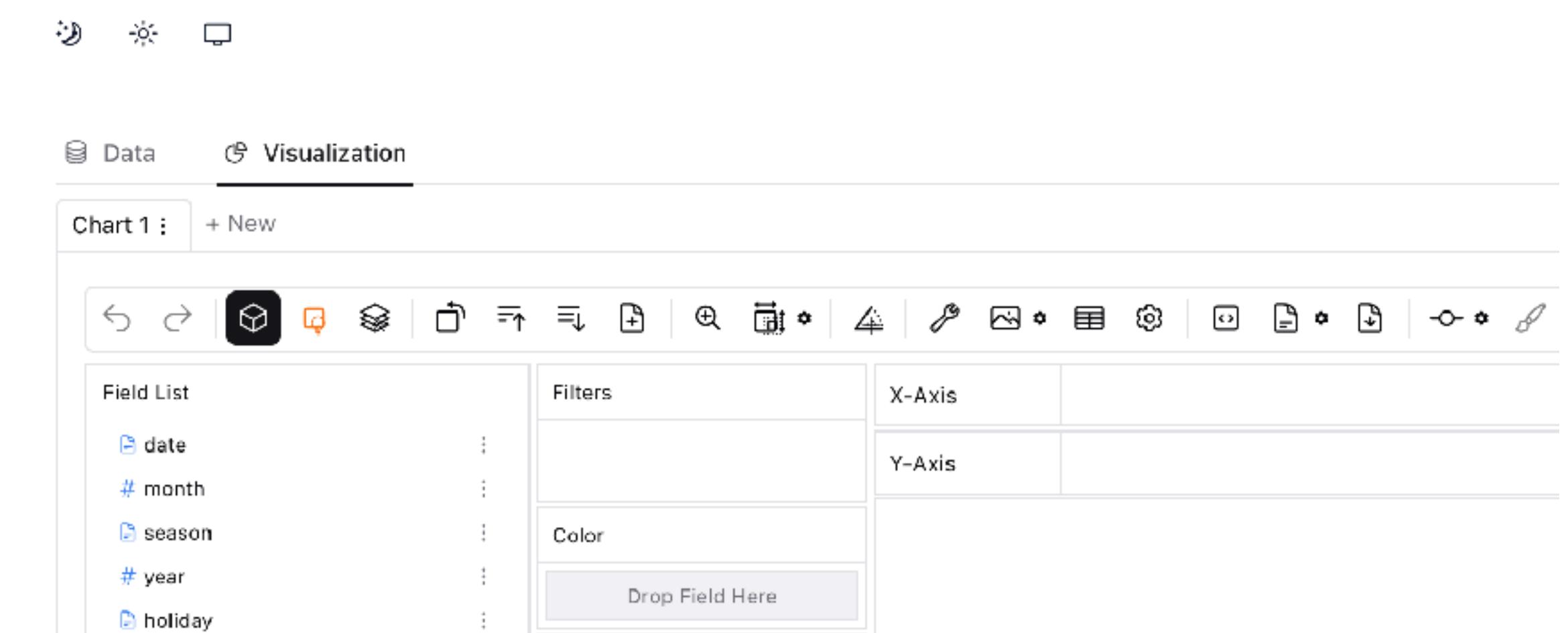


pygwalker			
web_server_demo	1d6c075960d7	pygwalker-image	8001:8001
jupyter_demo	010952949c9f	pygwalker-image	8888:8888
marimo_demo	4699a84bb49a	pygwalker-image	2718:2718
streamlit_demo	101be4582ffc	pygwalker-image	8501:8501



ポート2718のコンテナ : marimoに組み込み

- Jupyterと同じことができる
- また、シンプルなWebも作れる
- このコンテナは後者で、2つのPyGWalkerを上下に並べた



PyGWalker って何

可視化のスコープ

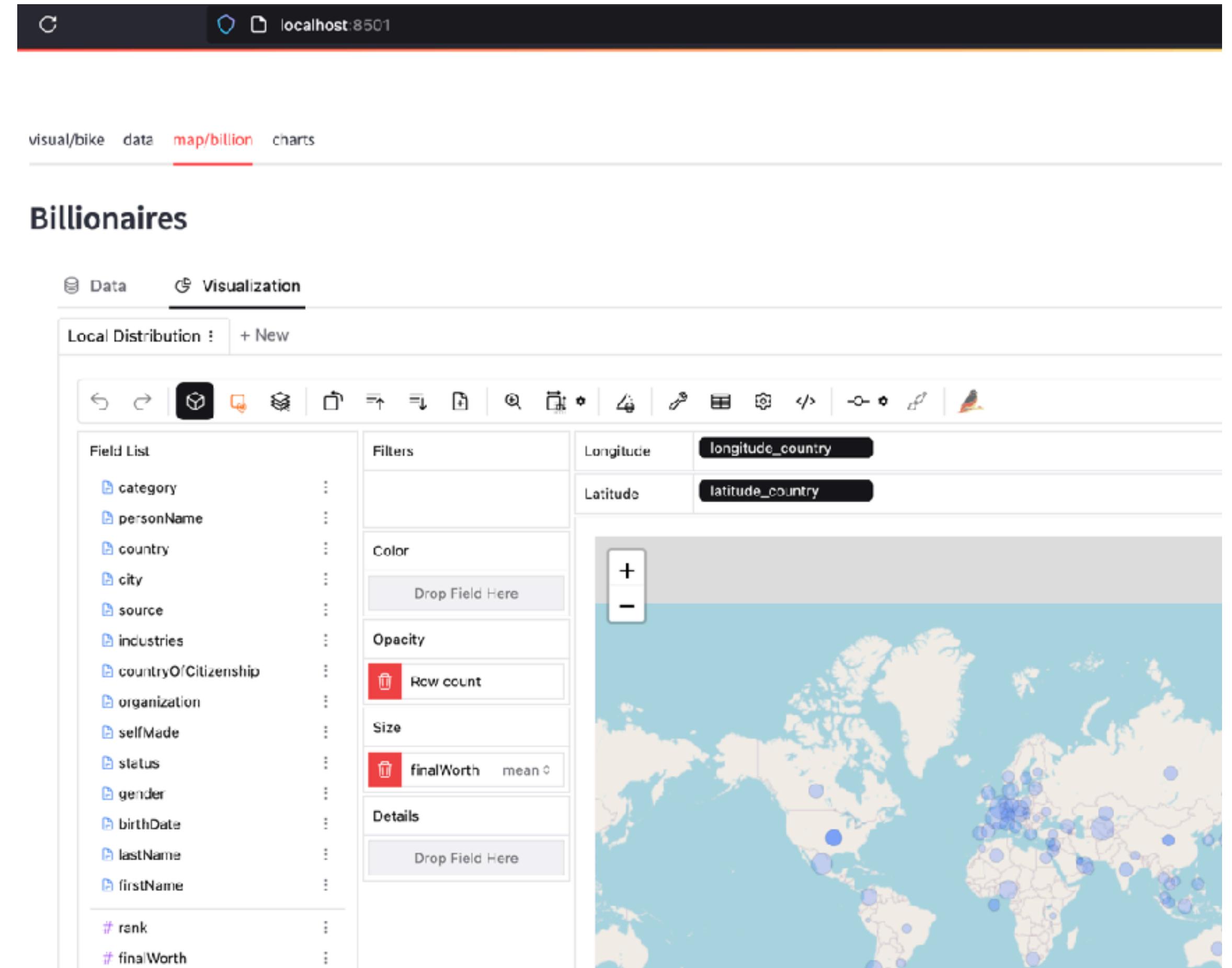
Alternative になる？



pygwalker			
web_server_demo	1d6c075960d7	pygwalker-image	8001:8001
jupyter_demo	010952949c9f	pygwalker-image	8888:8888
marimo_demo	4699a84bb49a	pygwalker-image	2718:2718
streamlit_demo	101be4582ffc	pygwalker-image	8501:8501

ポート8501のコンテナ : StreamlitでWebアプリ化

- データはKaggleのパブリックなBillionairesなど
- Streamlitでタブを作り、各タブにPyGWalkerを設置
- chartタブは個々のグラフを取り出して表示する例



PyGWalker って何

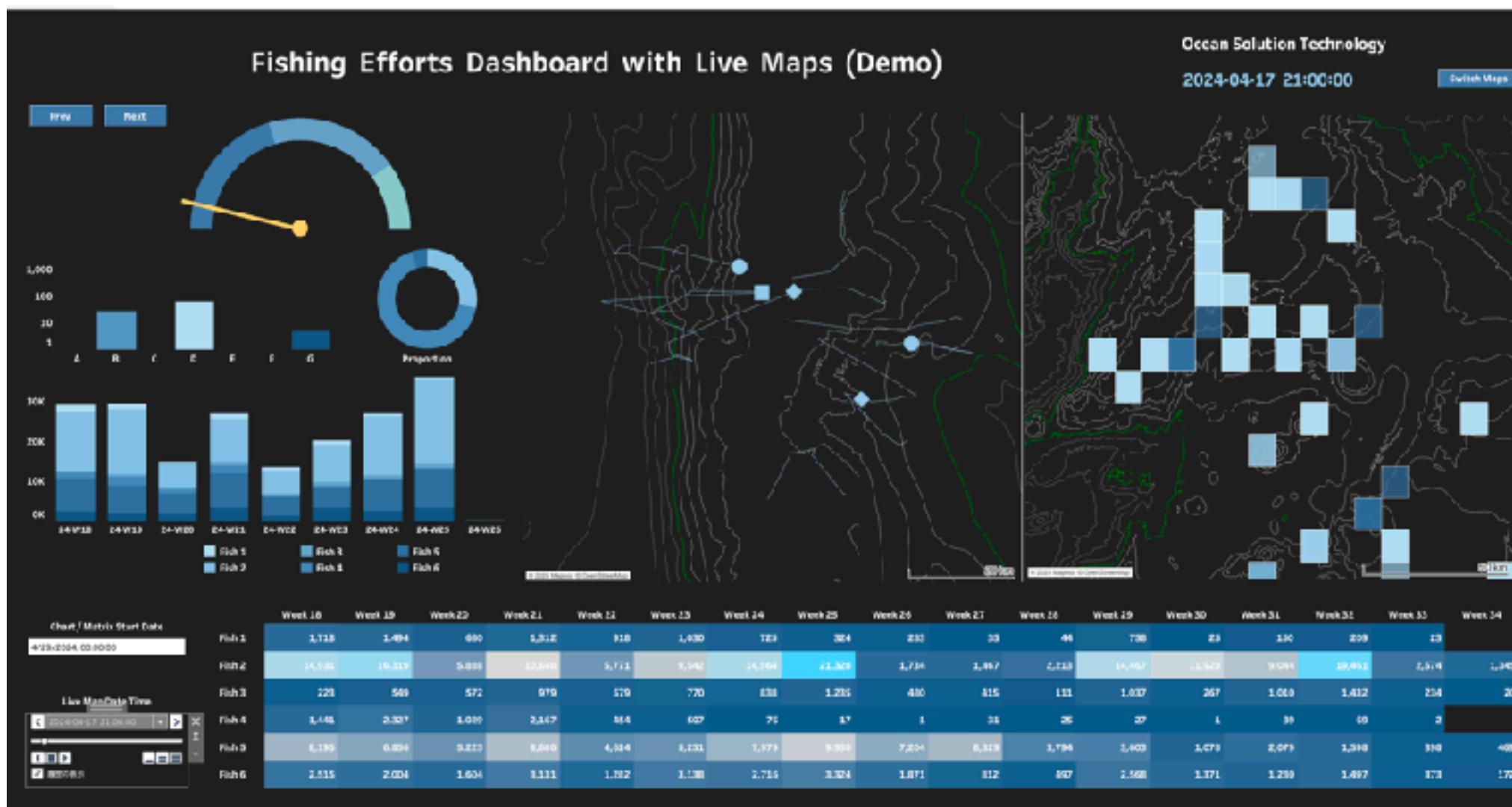
可視化のスコープ

Alternative になる？



<https://www.ocean5.co.jp>

オーシャンソリューションテクノロジー株式会社
本社＝長崎県佐世保市
漁業者支援システム「トリトンの矛」など展開



Tableauで感じた問題点（個人の主観）

- 使う側の経済的コスト, 作る側の時間的コスト
- 画面サイズ／環境の違いによるレイアウト崩れ
- 計算フィールドを多用せざるを得ず, その弊害
- 標準UIが古く, 自作が大変 (普通のボタンパーツすら無い)
- 機能拡張が可能だが, 外部／自前のサーバ依存になる
- 性能問題 (製品やライセンス間の差もあり得るか) etc.

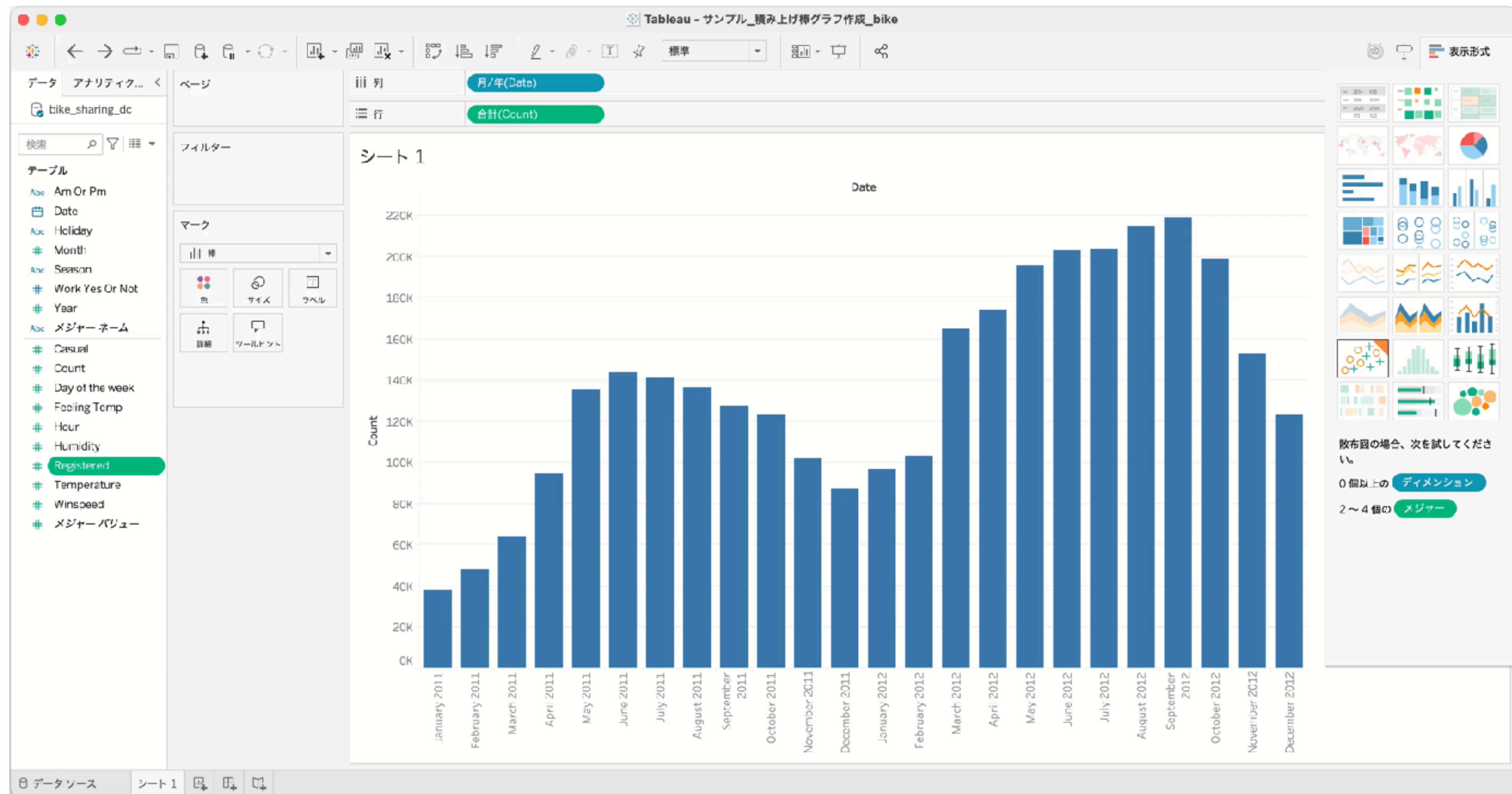
PyGWalker って何

可視化のスコープ

Alternative になる？



Tableau 積み上げ棒グラフを作るのに戸惑う動画



PyGWalker って何

可視化のスコープ

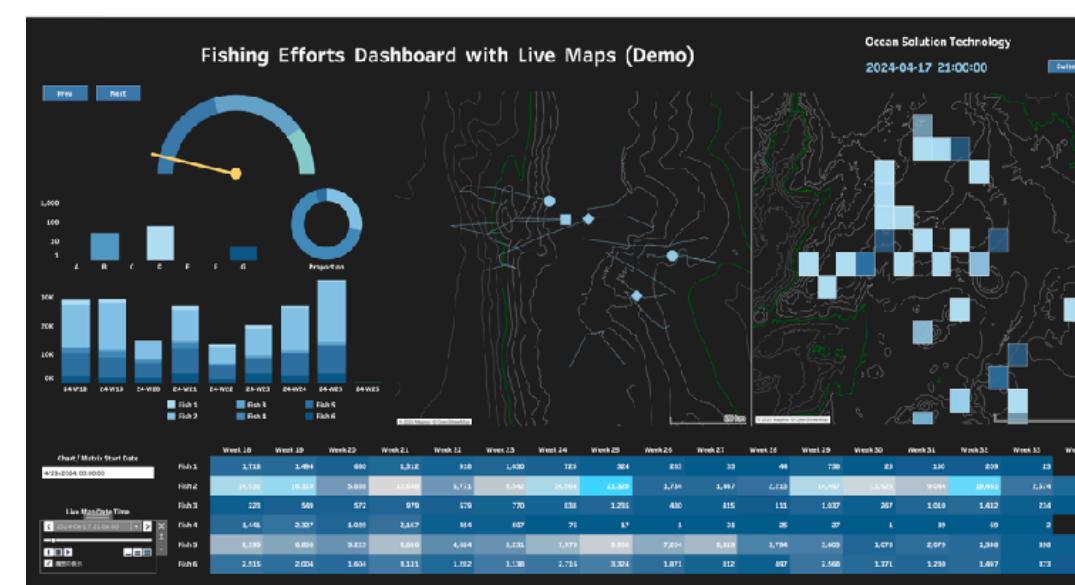
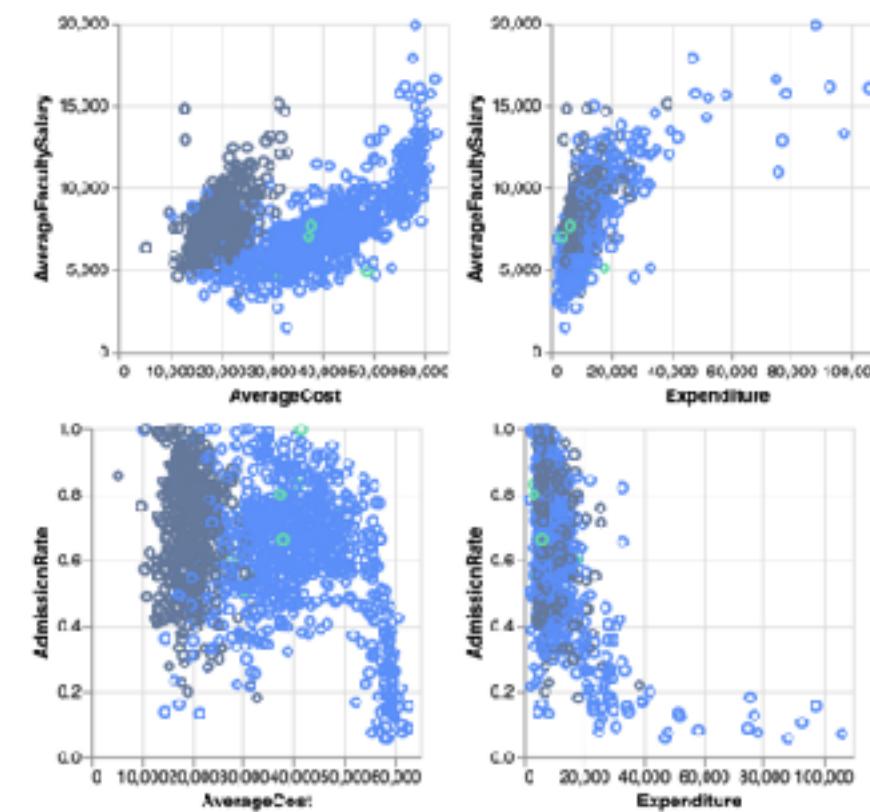
Alternative になる？

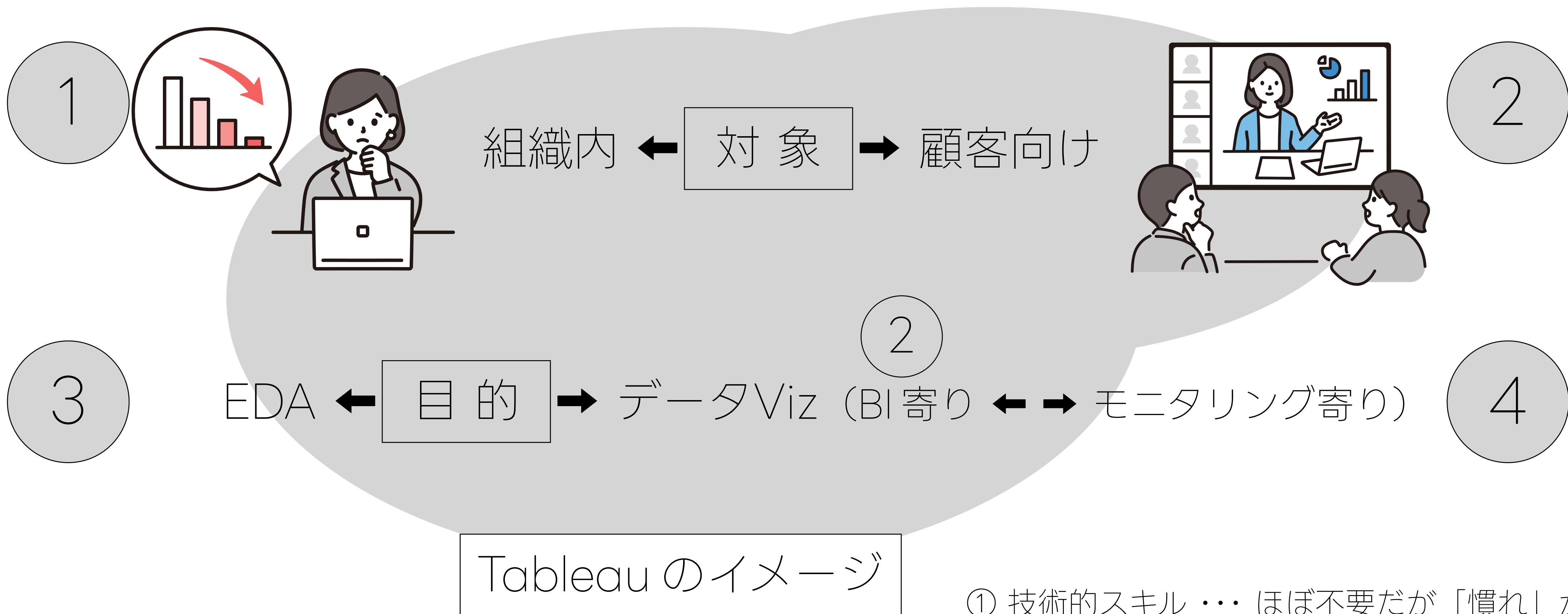


組織内 ← 対象 → 顧客向け



EDA ← 目的 → データViz (BI寄り) ← → モニタリング寄り)





- ① 技術的スキル … ほぼ不要だが「慣れ」が必要
- ② デザイン … それなりにできるが限界も多い
- ③ 統計学的分析 … ほぼ別ツールが必要
- ④ リアルタイムのデータ／画面更新 … 難しそう

PyGWalker って何

可視化のスコープ

Alternative になる？



- 作れるグラフ

Talbeau



PyGWalker



PyGWalker って何

可視化のスコープ

Alternative になる？

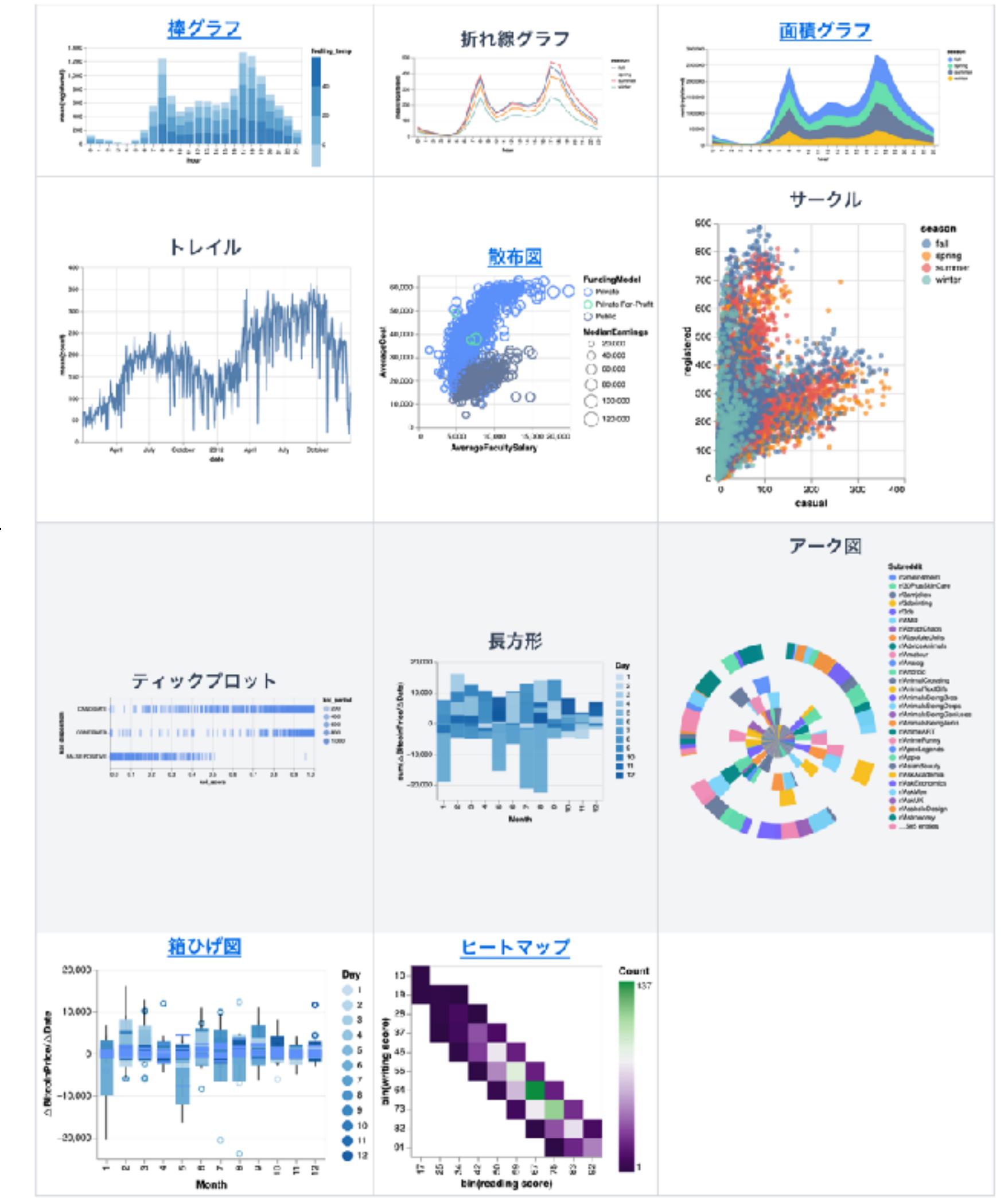


- 作れるグラフ

Talbeau



(Py)GWalker
公式サイトの
ギャラリーより



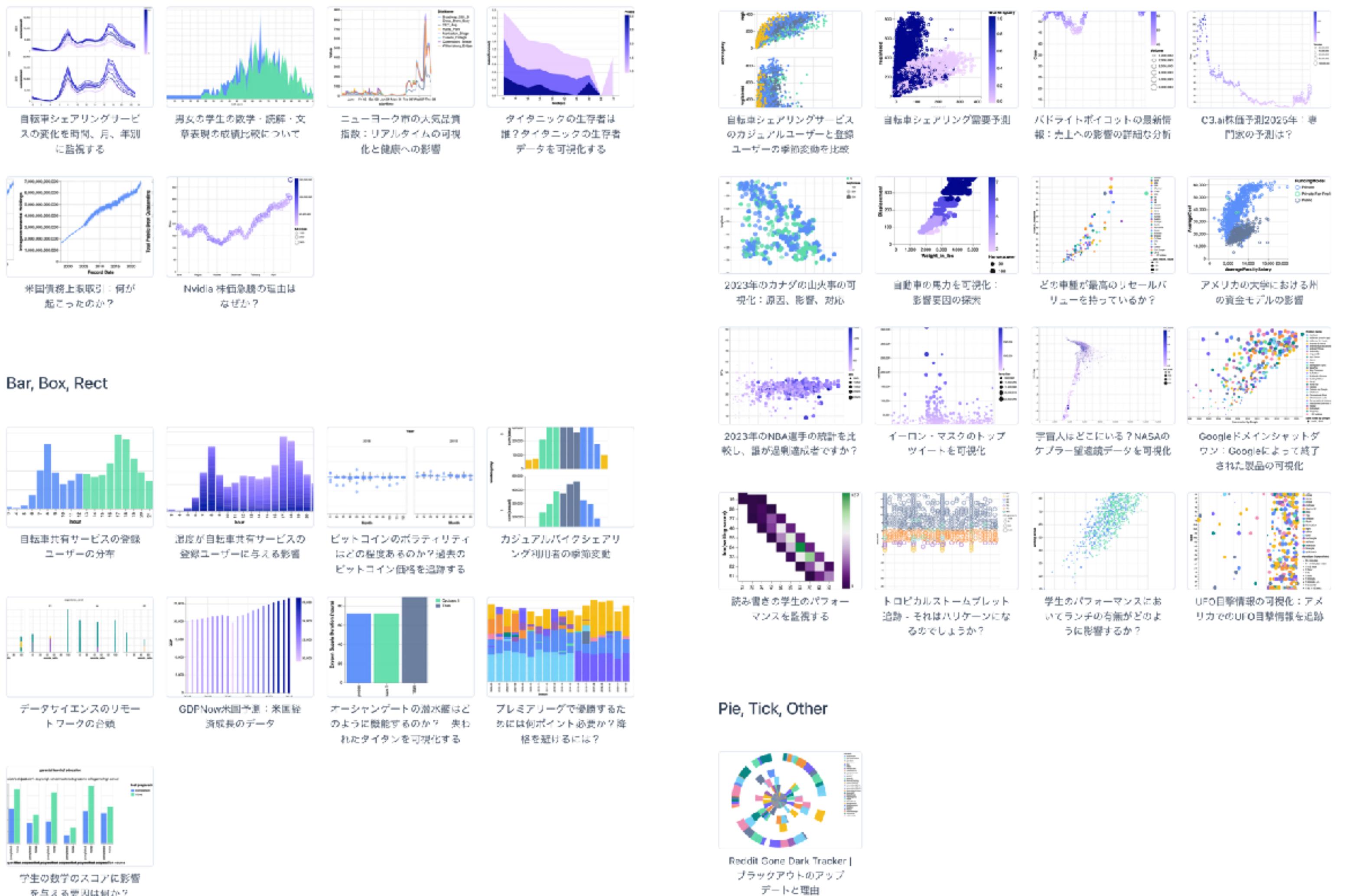
PyGWalker って何

可視化のスコープ

Alternative になる？



- 作れるグラフ





- 基本的なグラフは同等 (棒, 線, エリアチャート, 円, 散布図, ボックスプロット)

- Tableau でのみ描けるもの ➔

- PyGWalker にあってほしい...

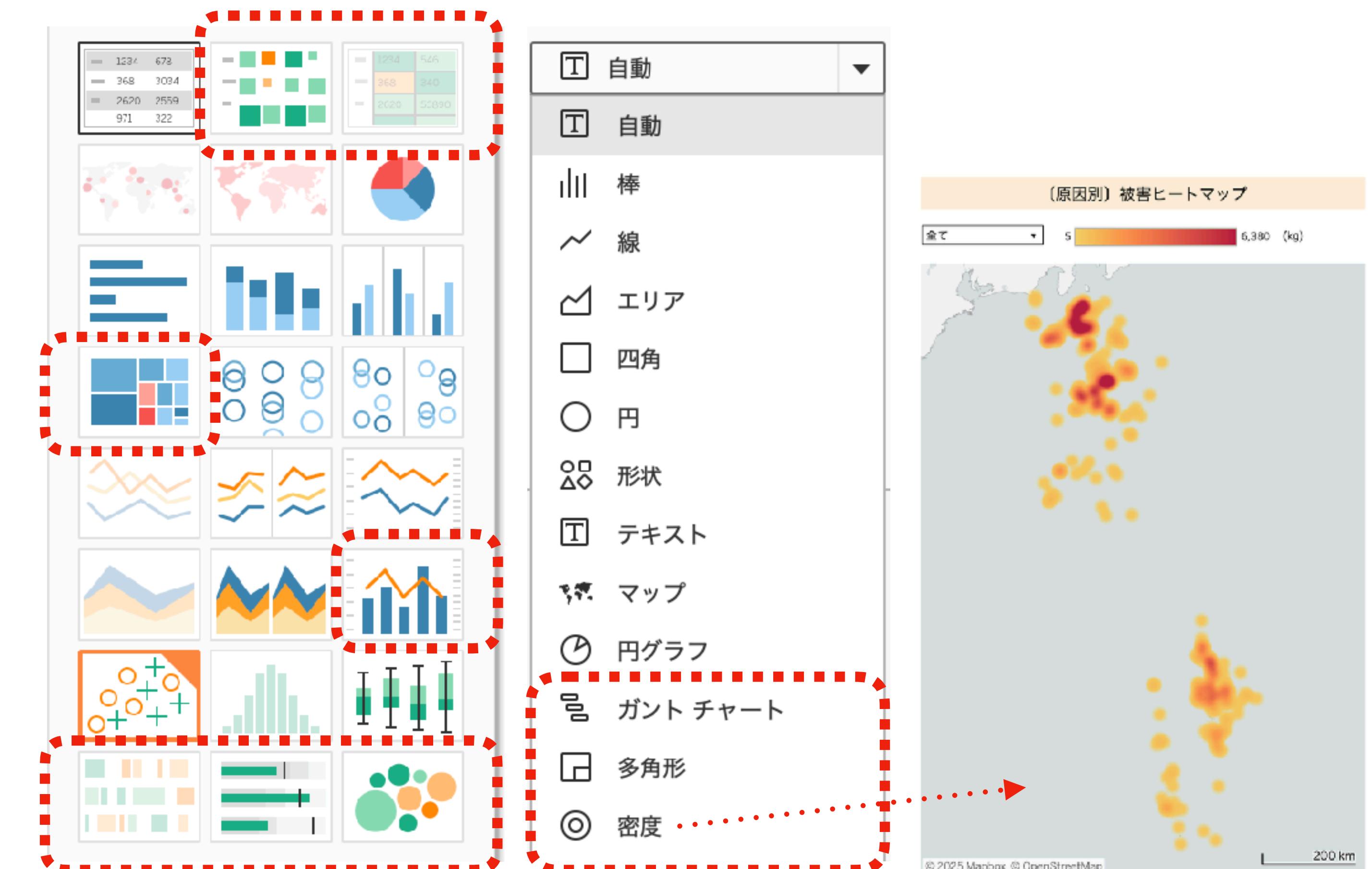
2軸組み合わせグラフ

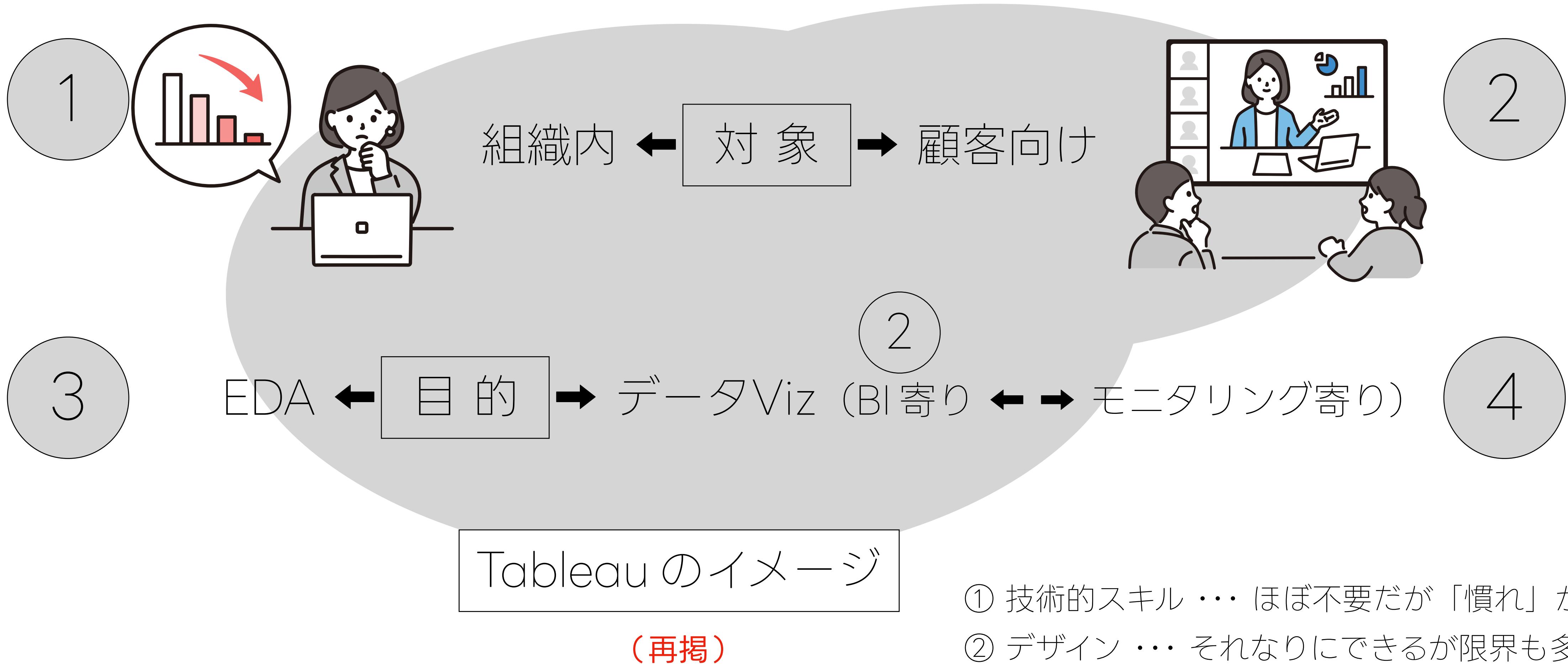
表セルの数値に応じた着色

密度マップ

異種グラフの複数レイアウト

ぐらい？ その意味では PyGWalker すごい



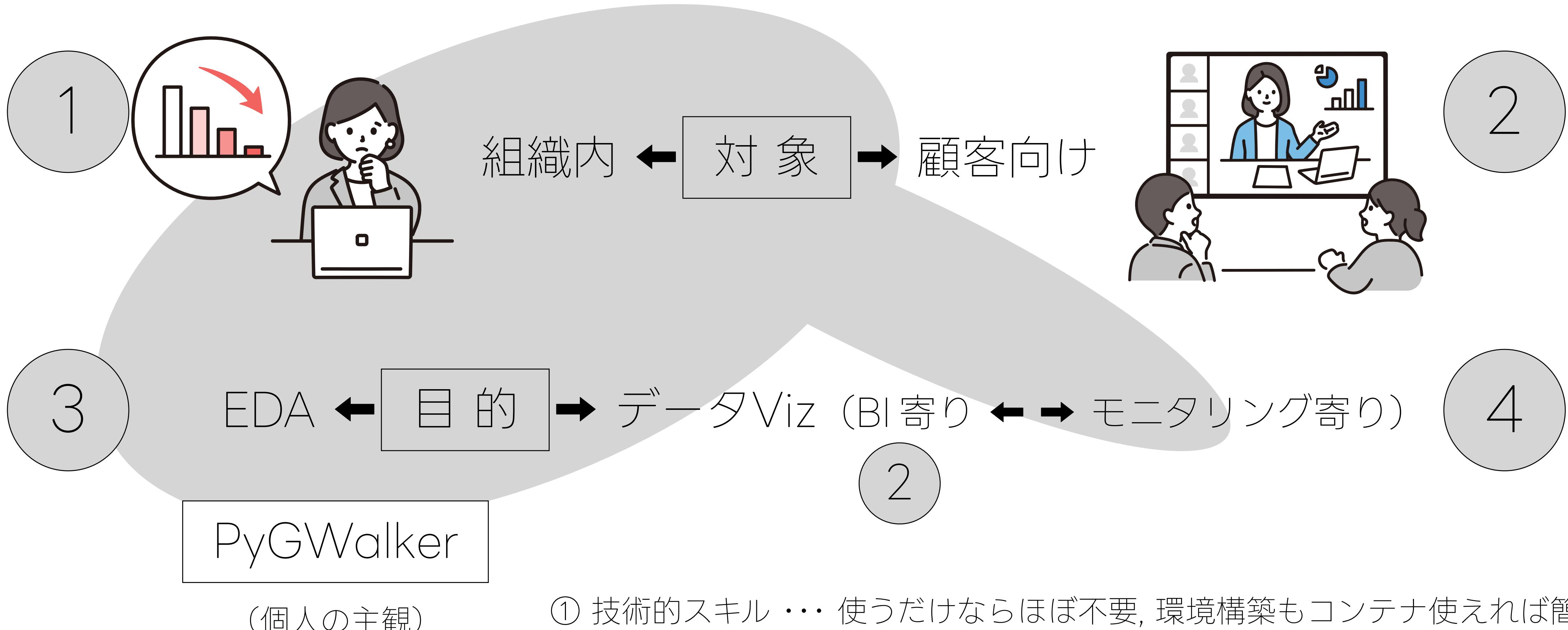


- ① 技術的スキル …… ほぼ不要だが「慣れ」が必要
- ② デザイン …… それなりにできるが限界も多い
- ③ 統計学的分析 …… ほぼ別ツールが必要
- ④ リアルタイムのデータ／画面更新 …… 難しそう

PyGWalker って何

可視化のスコープ

Alternative になる？



- ① 技術的スキル … 使うだけならほぼ不要, 環境構築もコンテナ使えば簡単
- ② 異種グラフのレイアウト, 多様なデザイン … 難しい
- ③ **統計学的分析 … Jupyter等と簡単に連携できる**
- ④ リアルタイムのデータ／画面更新 … Webフレームワークと連携すれば可能？



結論

組織内の EDA 向けなら十分ありだが、BI ダッシュボードは厳しい

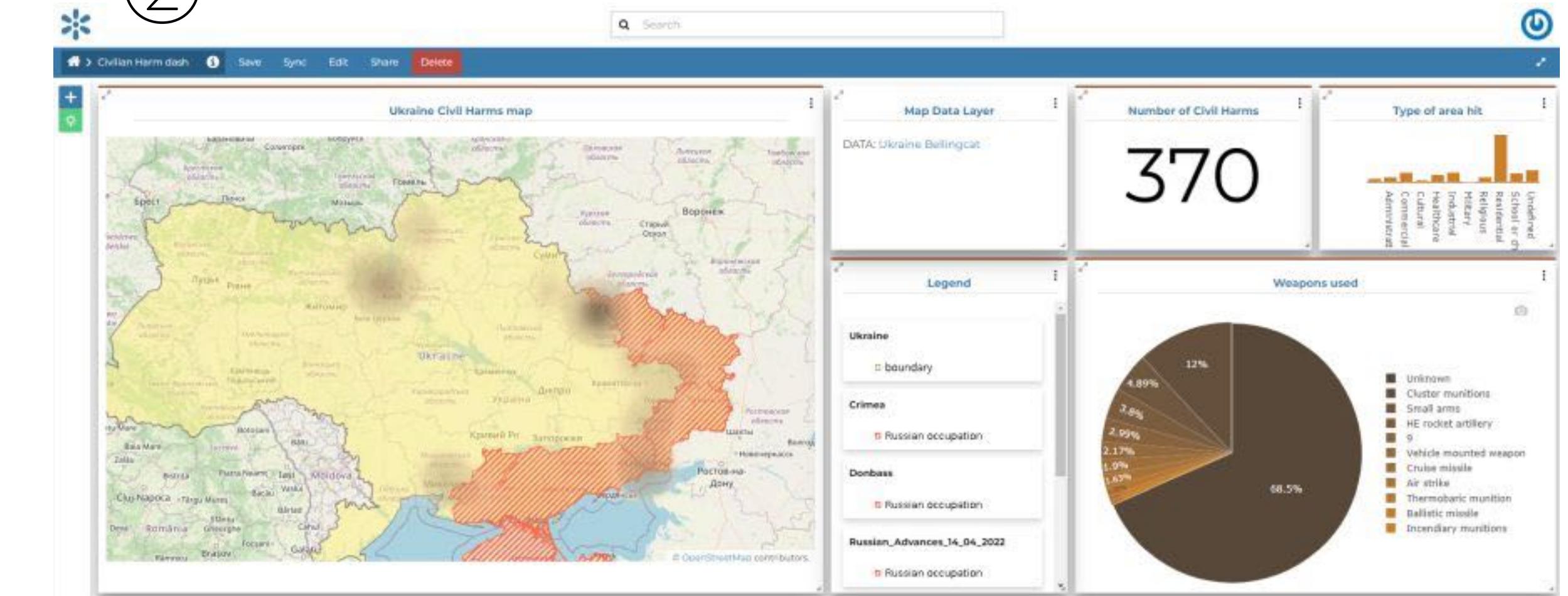
- Web フレームワークで相当作り込んでも…同等になるか不明
- Graphic Walker の有償デスクトップ版①のダッシュボードも、まだ機能が少ない
- この分野の open source Alternative はブルーオーシャン（最も近いのは② GeoNode ?）

①

ダウンロードは<https://kanaries.net/download>から

The screenshot shows the Graphic Walker desktop application interface. At the top, there are tabs for Data, Visualization, and Dashboard. The Dashboard tab is selected. Below the tabs, there's a button for 'New Dashboard'. On the left, a sidebar titled 'Dashboard Items' lists 'Charts' and 'Elements'. Under 'Charts', there are two large numerical displays: one showing 'random from SQL' with the value '0.8222794886667' and another showing 'Average temperature' with the value '15.358396915817'. The main area is labeled 'Title'.

②



以上です。ご清聴ありがとうございました！

スライドと Docker コンテナ



Instagram

