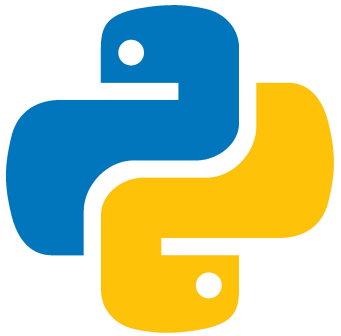
Outils génériques de visualisation de données pour Python



Par Vincent Roger

23/11/2023

À propos de moi

PhD en informatique.

Data scientist de Kiviak instrument et spécialiste de l'apprentissage automatique.

LinkedIn

https://www.linkedin.com/in/vroger11/

Site web

https://website.vincent-roger.fr/

Pourquoi la visualisation de données avec Python ?

- L'extraction et la préparation des données sont simplifiées par des outils tels que Pandas, Dask et d'autres.
- Il s'agit d'un langage incontournable pour les scientifiques des données, l'apprentissage automatique et les personnes travaillant sur les MLOPS.
- Alors pourquoi apprendre un nouveau langage juste pour la visualisation de données ?

Quels types d'outils

Spécifiques (pour graphes)

- NetworkX
- Graph-tool
- PyGraphviz → Graphviz
- Pyvis

Générique

Objectif de cette présentation !

Plan

- Non orienté web
- Orienté web

Données que nous allons utiliser

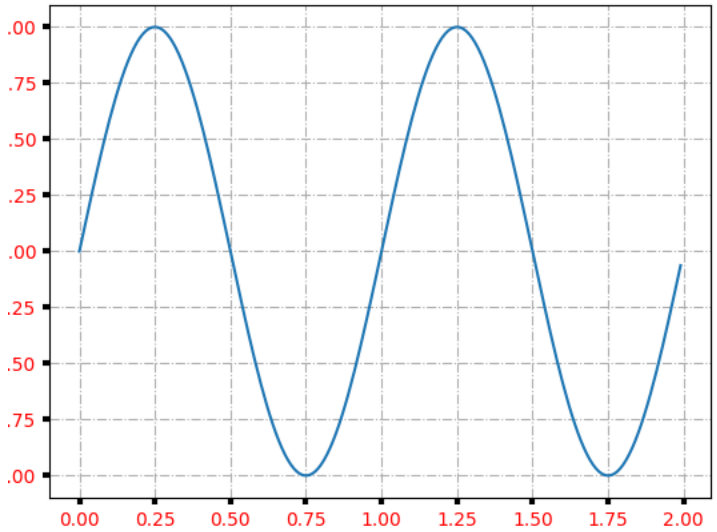
```
import pandas as pd

df_iris = pd.read_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/iris.csv')

df_iris
```

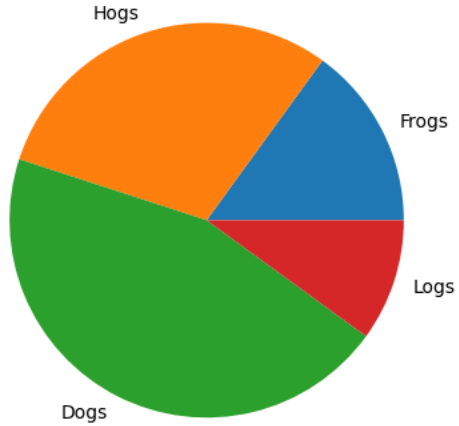
```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

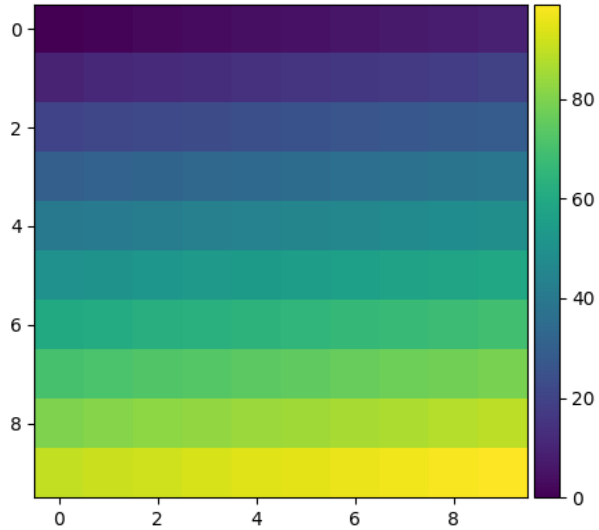
Non orienté web

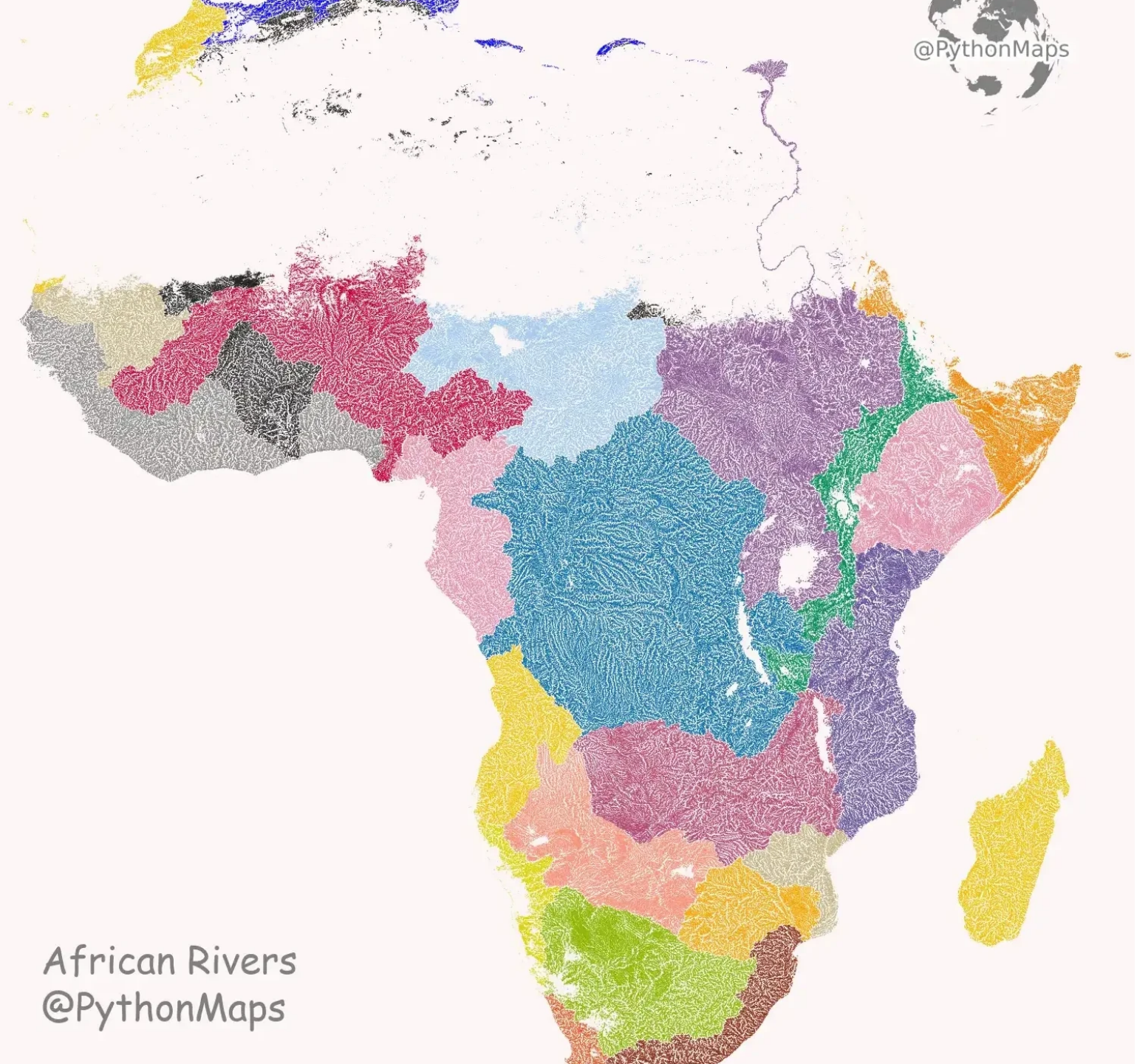
Matplotlib

- Basé sur Qt.
- Bibliothèque standard.
- Par défaut pour de nombreux outils comme Pandas.









Matplotlib

- Puissant si maîtrisé !

Matplotlib - diagramme de dispersion des sépales de l'iris

```
import matplotlib.pyplot as plt

# Create a scatter plot with legends based on species
plt.figure(figsize=(8, 6))

# Iterate over each species and plot scatter points
for species in df_iris["species"].unique():
    species_data = df_iris[df_iris['species'] == species]
    plt.scatter(species_data['sepal_length'], species_data['sepal_width'], label=species)

# Add labels and title
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.title('Scatter Plot of Iris Dataset')

# Add legend
plt.legend()

# Show the plot
plt.show()
```

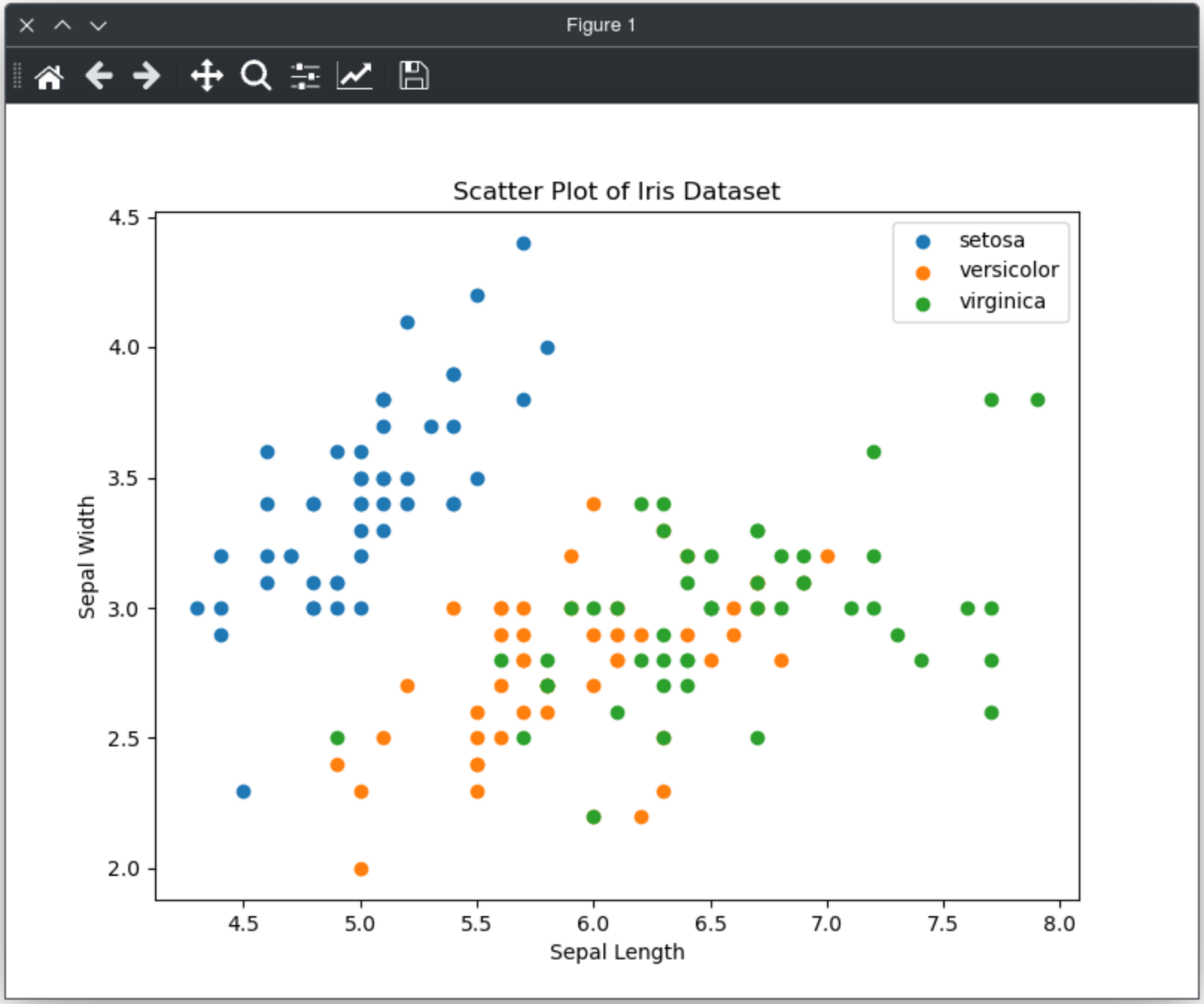


Figure 1
Scatter Plot of Iris Dataset
setosa
versicolor
virginica
Sepal Width
Sepal Length
4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

Matplotlib

Avantages :

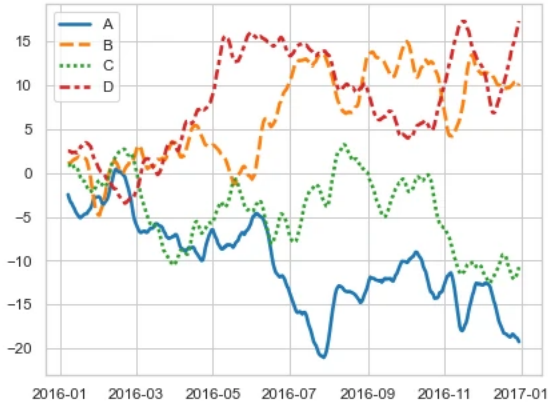
- La librairie la plus utilisée.
- Travailler localement.
- Grande communauté.

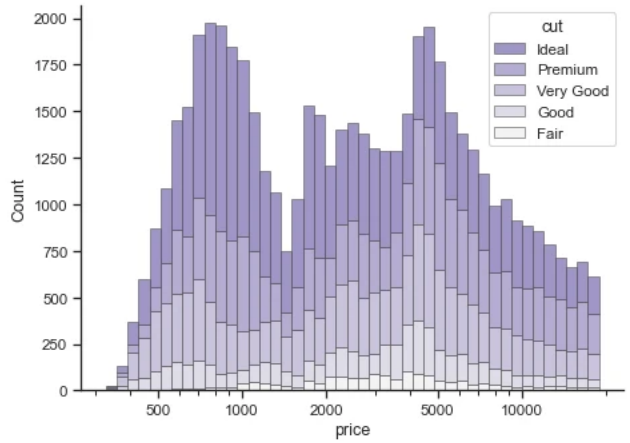
Inconvénients :

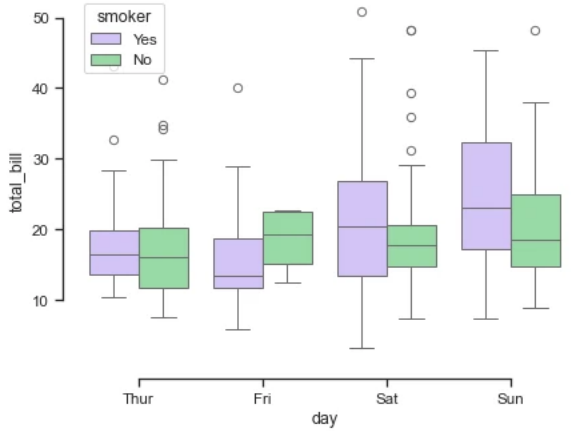
- Verbeux.
- N'est pas bien adapté aux tracés dynamiques (en particulier avec de nombreux points de données).
- Il peut être difficile de réaliser certains tracés spécifiques.

Seaborn

- Basé sur Matplotlib.
- Conçu pour les graphiques statistiques.







Seaborn - diagramme de dispersion des sépales de l'iris

```
import matplotlib.pyplot as plt
import seaborn as sns

# Create a scatter plot
sns.scatterplot(data=df_iris, x="sepal_length", y="sepal_width", hue="species")

# Add labels and title
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.title('Scatter Plot of Sepal')

# Display the plot
plt.show()
```

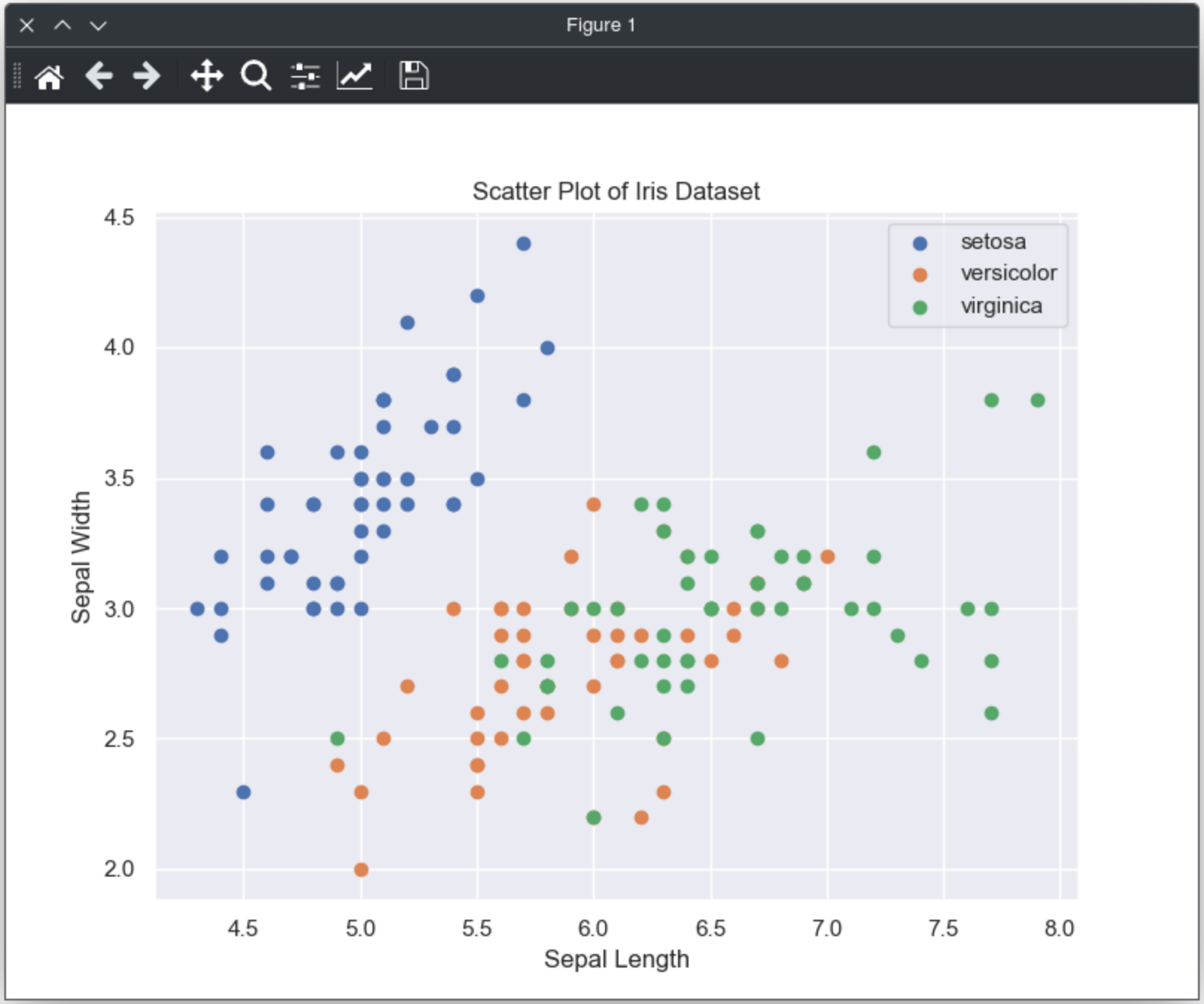
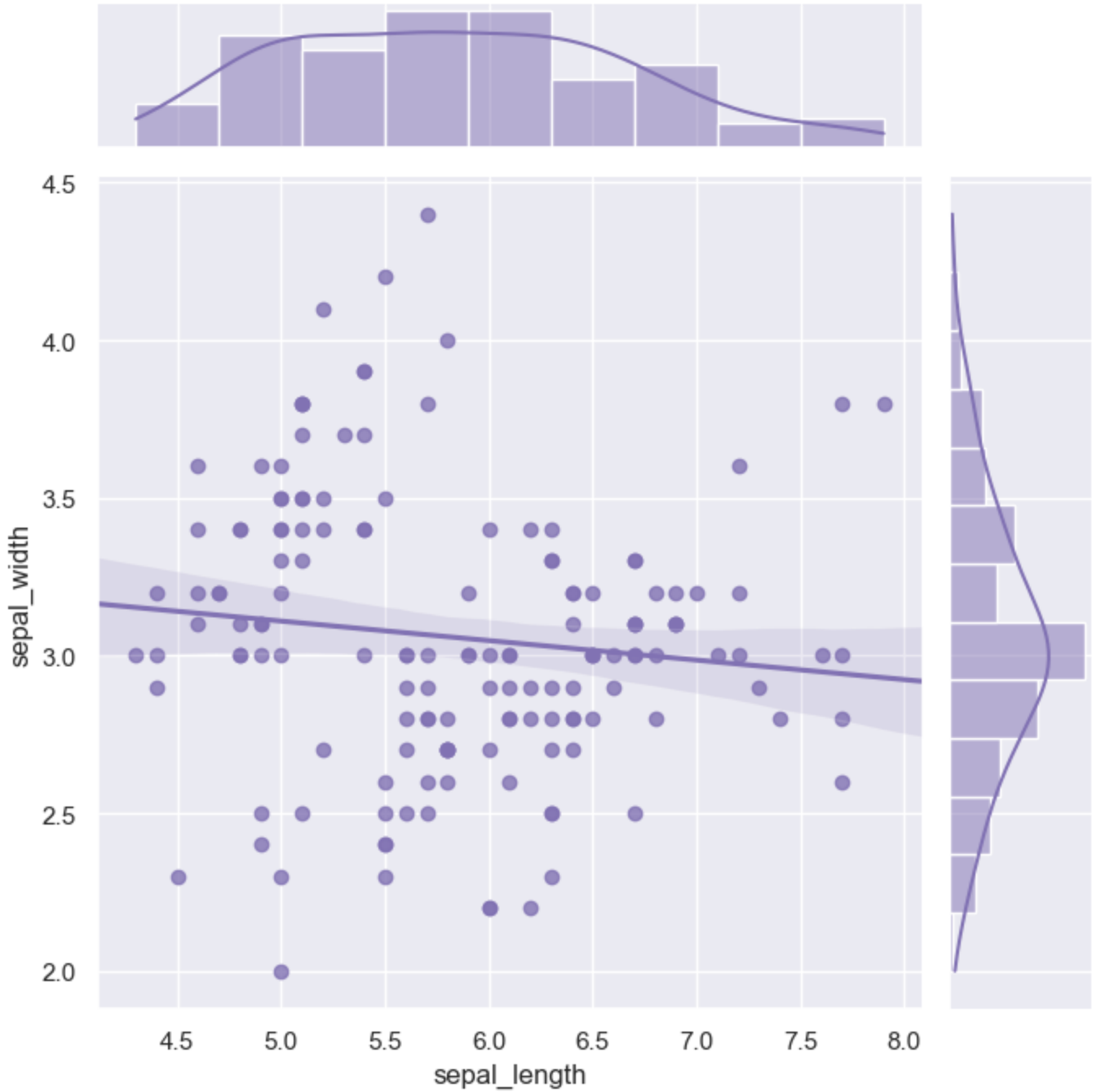


Figure 1
Scatter Plot of Iris Dataset
setosa
versicolor
virginica
Sepal Width
Sepal Length
4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

Seaborn - faire rapidement des statistiques

```
import matplotlib.pyplot as plt
import seaborn as sns

# Compute jointplot
sns.set_theme(style="darkgrid")
g = sns.jointplot(data=df_iris, x="sepal_length", y="sepal_width",
                  kind="reg", truncate=False,
                  color="m", height=7)

# Display the plot
plt.show()
```


4.5
4.0
3.5
3.0
2.5
2.0
sepal_width
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
sepal_length

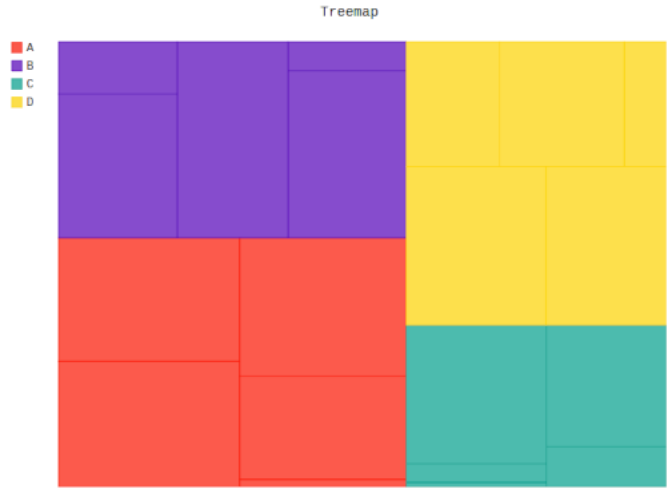
Seaborn

Avantages :

- Fournit des thèmes.
- Fournit des fonctions de haut niveau pour l'analyse statistique afin de faciliter certains tracés.

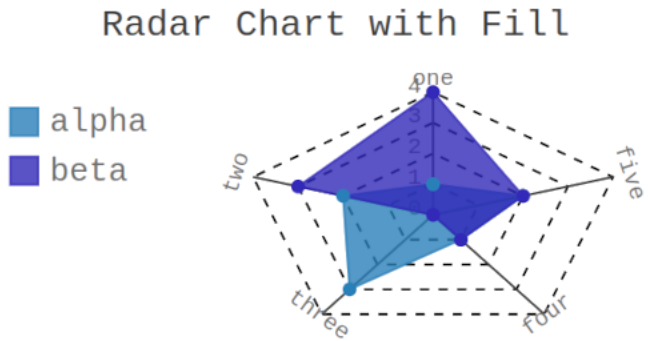
Inconvénients :

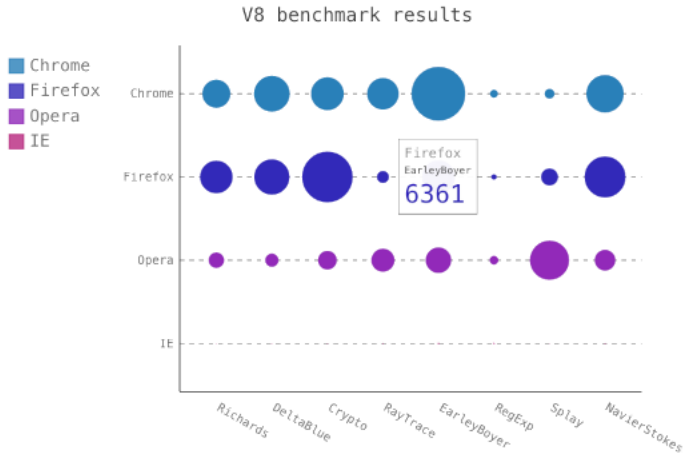
- Toujours un peu verbeux lors de la modification de certains tracés.
- N'est pas conçu pour les tracés dynamiques.
- Ne permet pas de générer tous les types de tracés.



Pygal

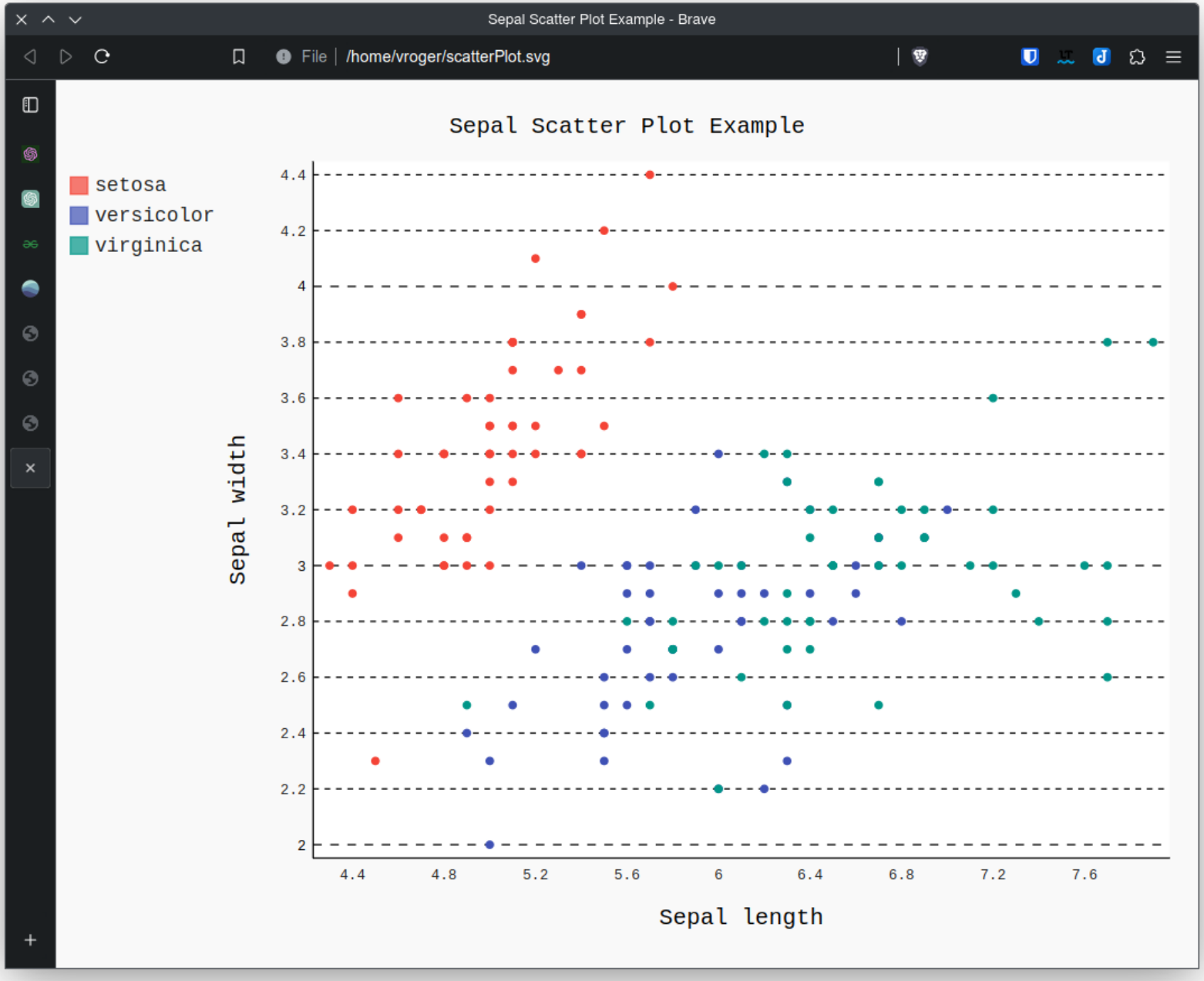
- Produire des SVG dynamiques.
- Écrit en Python.





Pygal - diagramme de dispersion des sépales de l'iris

```
import pygal

scatter_chart = pygal.XY(stroke=False)
for specie in df_iris["species"].unique():
    df_s = df_iris[df_iris["species"] == specie]
    scatter_chart.add(f'{specie}', list(zip(df_s["sepal_length"], df_s["sepal_width"])))

scatter_chart.title = 'Sepal Scatter Plot Example'
scatter_chart.x_title = 'Sepal length'
scatter_chart.y_title = 'Sepal width'
scatter_chart.render_to_file('scatterPlot.svg')
```


Sepal Scatter Plot Example - Brave
File | /home/vroger/scatterPlot.svg
Sepal Scatter Plot Example
setosa
versicolor
virginica
Sepal width
4.4
4.2
4
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
4.4
4.8
5.2
5.6
6
6.4
6.8
7.2
7.6
Sepal length

Pygal

Avantages

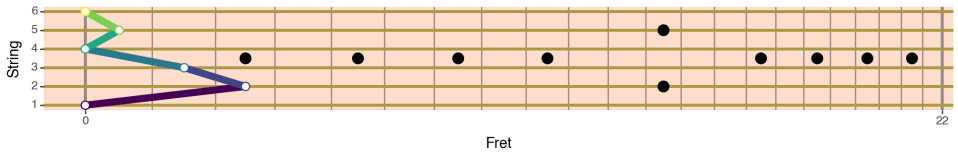
- Même si le résultat est un fichier SVG, il est interactif par défaut !
- Grande variété de graphiques.

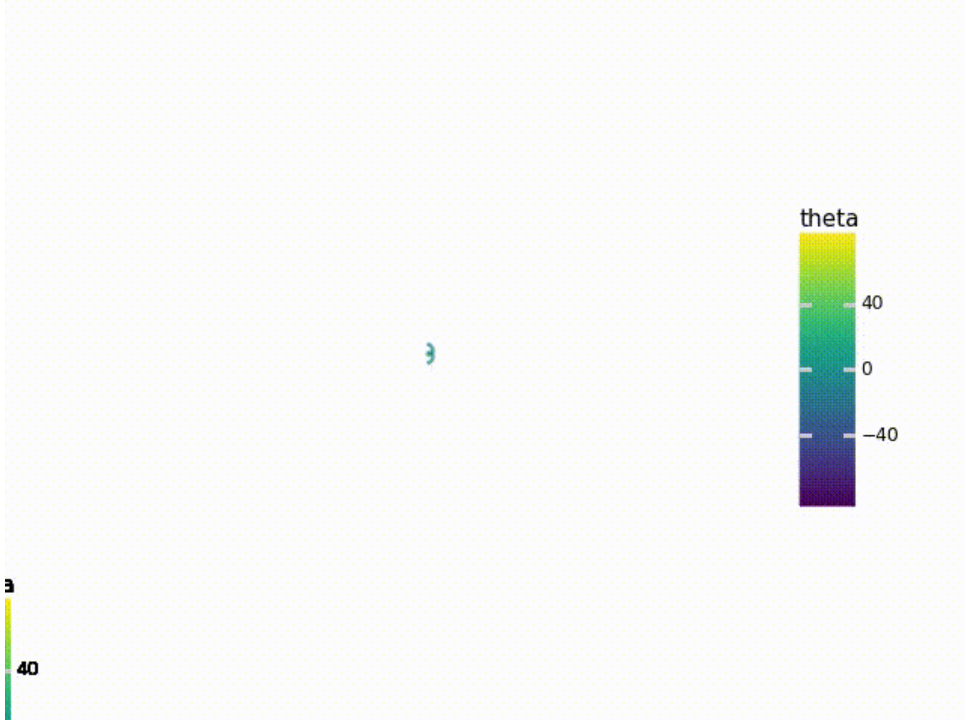
Inconvénients

- SVG compatible avec les navigateurs web, mais pas avec des logiciels comme Inkscape ou notebook (même avec rendu HTML).
- Les graphiques ne sont pas clairs avec les valeurs par défaut (en particulier la grille).

Plotnine

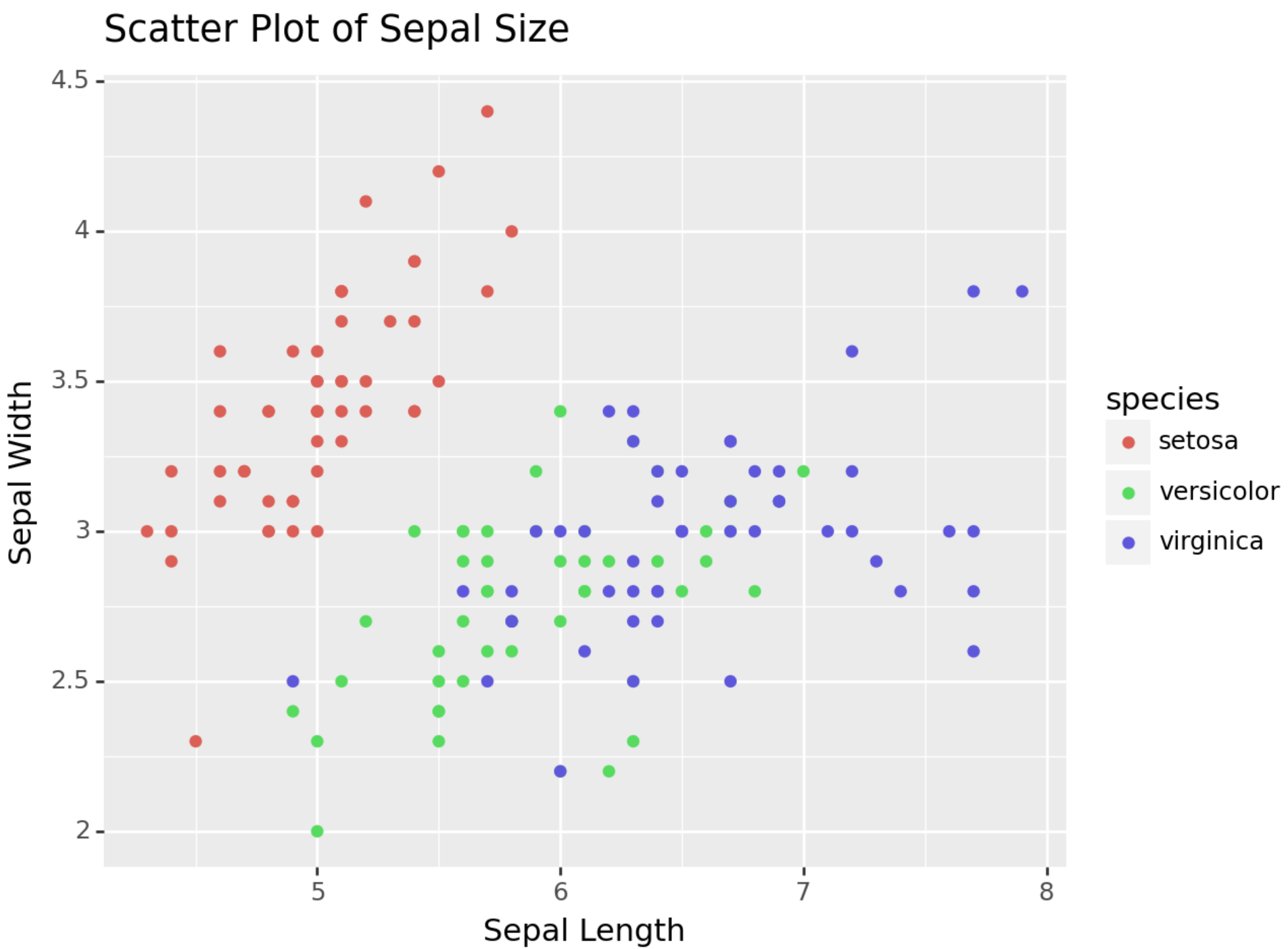
- Implémentation de la grammaire de ggplot2 en Python.





Plotnine - diagramme de dispersion des sépales de l'iris

```
from plotnine import ggplot, aes, geom_point, labs

# Create a scatter plot of sepal size
scatter_plot = (
    ggplot(df_iris, aes(x='sepal_length', y='sepal_width', color='species')) +
    geom_point() +
    labs(title='Iris Dataset Scatter Plot', x='Sepal Length', y='Sepal Width')
)

# Display the plot
print(scatter_plot)
```


Scatter Plot of Sepal Size
Sepal Width
Sepal Length
4.5
4
3.5
3
2.5
2
5
6
7
8
species
setosa
versicolor
virginica

Plotnine

Avantages

- Aussi puissant que ggplot2.
- Outil adapté aux utilisateurs de R utilisant Python.
- Grande variété de tracés possibles.

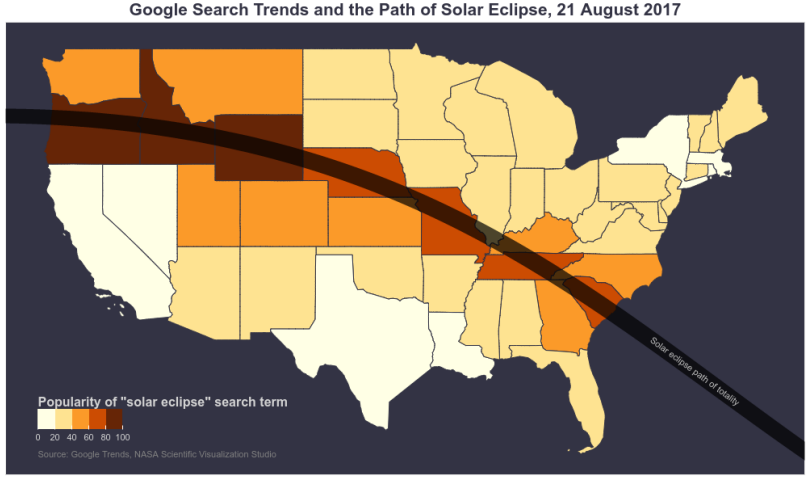
Inconvénients

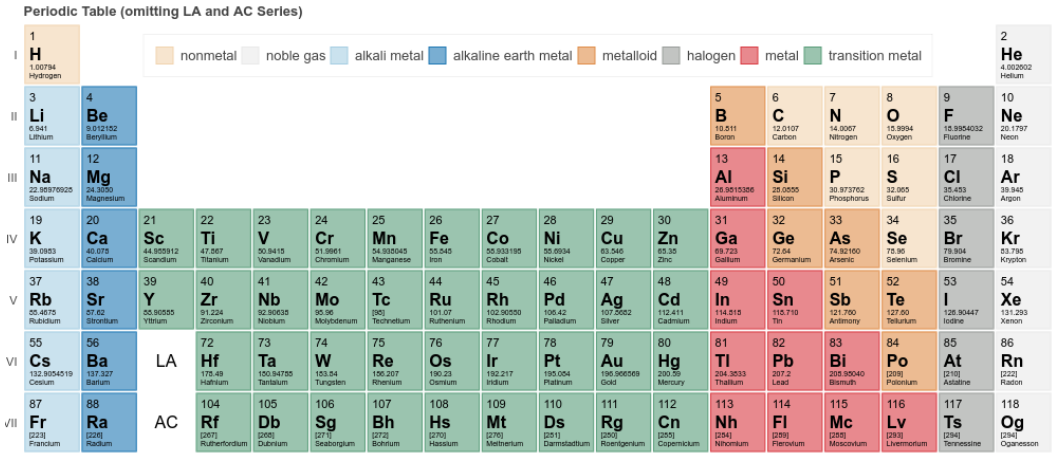
- La syntaxe n'est pas Python.
- Ne permet pas de tracer de grands ensembles de données.
- L'interactivité des tracés est limitée.

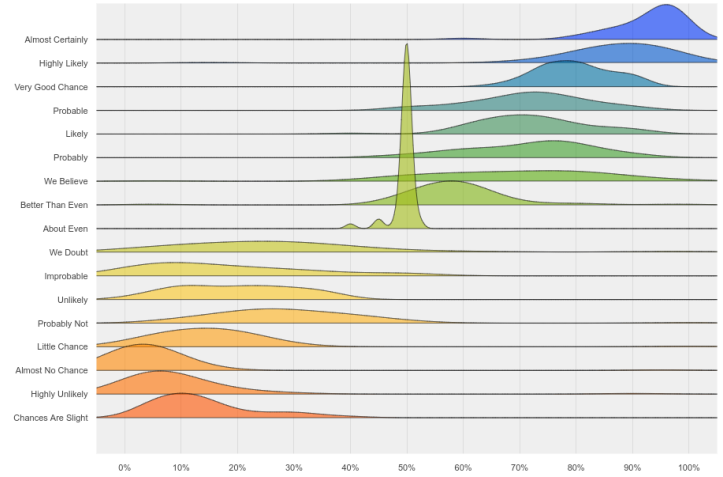
Orienté Web

Bokeh

- Utilisable dans Jupyter et IPython (sortie HTML par défaut).
- Capacité de créer des tableaux de bord.
- L'utilisation de WebGL est possible.

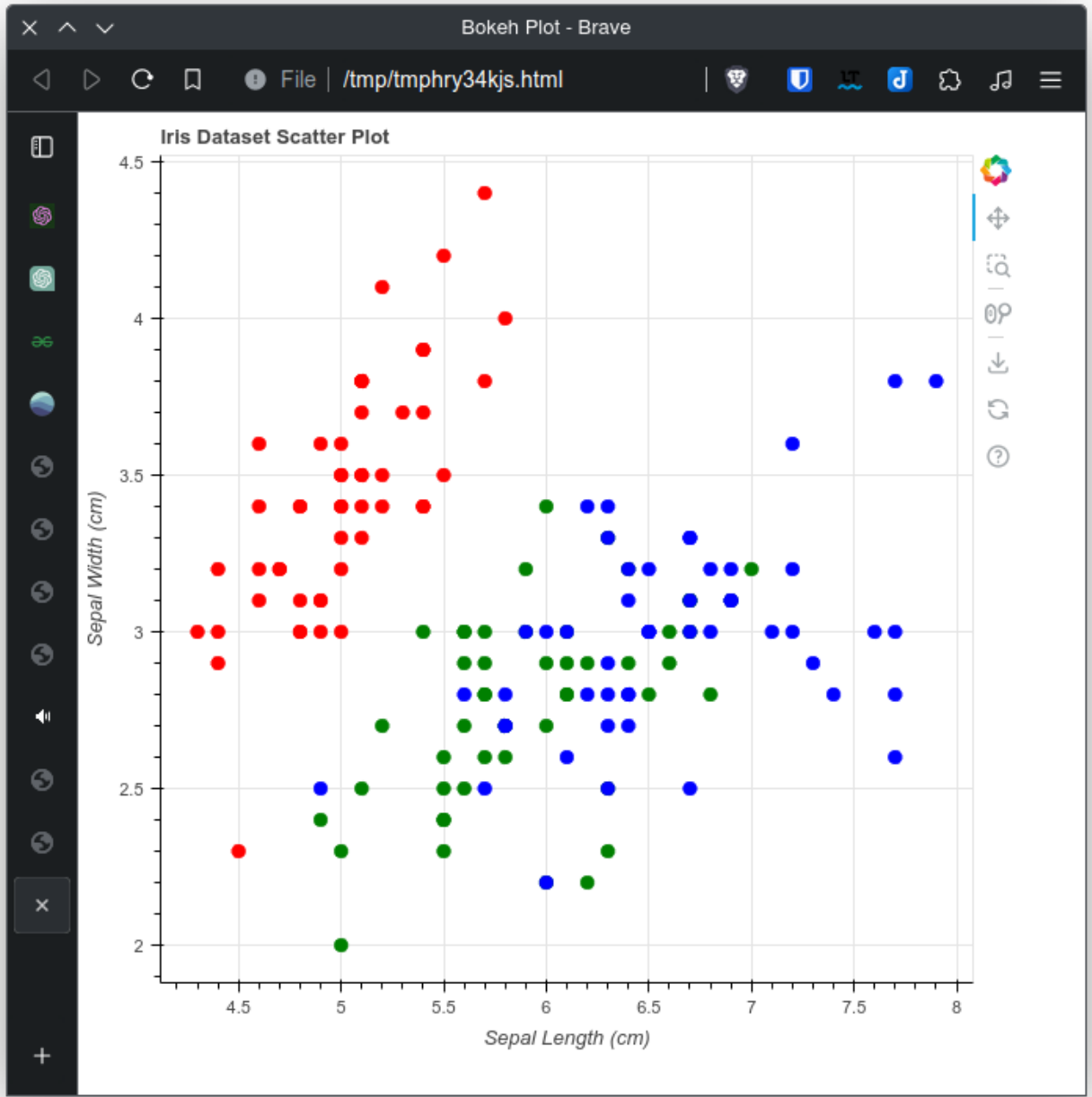






Bokeh - diagramme de dispersion des sépales de l'iris

```
from bokeh.plotting import figure, show
from bokeh.transform import factor_cmap
from bokeh.models import ColumnDataSource

# Create a ColumnDataSource
source = ColumnDataSource(df_iris)

# Create a scatter plot with titles
p = figure(title='Iris Dataset Scatter Plot', x_axis_label='Sepal Length (cm)', y_axis_label='Sepal Width (cm)')

p.circle(x='sepal_length', y='sepal_width', source=source,
                size=8,
              color=factor_cmap(
                         'species',
                          palette=['red', 'green', 'blue'],
                          factors=df_iris["species"].unique()
               )
)

# Show the plot
show(p)
```


Bokeh Plot - Brave
File | /tmp/tmphry34kjs.html
Iris Dataset Scatter Plot
Sepal Width (cm)
Sepal Length (cm)
4.5
4
3.5
3
2.5
2
4.5
5
5.5
6
6.5
7
7.5
8

Bokeh

Avantages

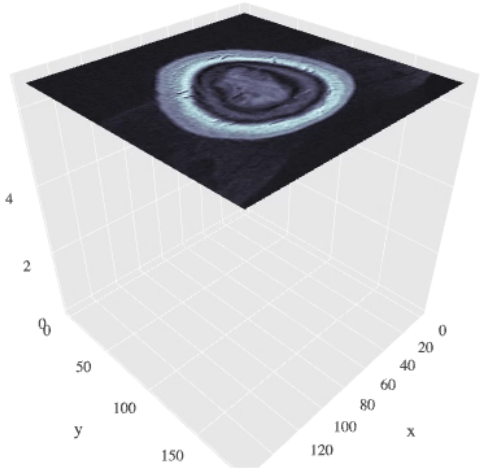
- API de bas niveau et de haut niveau.
- Peut facilement gérer un million de points.
- Interactivité (widgets et tableaux de bord).
- Capacité à réaliser de nombreux tracés différents.

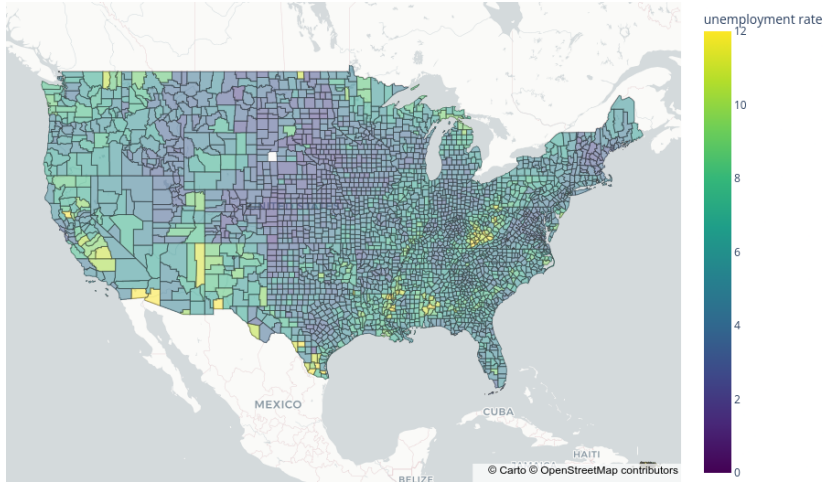
Inconvénients

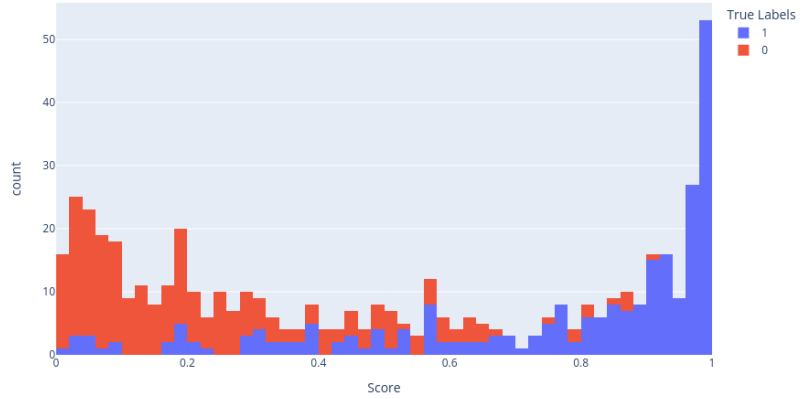
- Plus difficile à apprendre que d'autres bibliothèques.
- Moins complet pour les tracés en 3D.
- Si les serveurs de Bokeh sont hors service, vos tracés peuvent ne pas s'afficher correctement (ou pas du tout).

Plotly

- Conçu pour les visualisations interactives.
- L'utilisation de WebGL est possible.
- Utilisable avec Python, R et JavaScript.







Plotly - diagramme de dispersion des sépales de l'iris

```
import plotly.express as px

fig = px.scatter(df_iris, x='sepal_length', y='sepal_width', color='species',
                 title='Iris Dataset Scatter Plot', labels={'sepal_length': 'Sepal Length', 'sepal_width': 'Sepal Width'})

# Show the plot
fig.show()
```


127.0.0.1:35611 - Brave
127.0.0.1:35611
Iris Dataset Scatter Plot
Sepal Width
Sepal Length
4.5
4
3.5
3
2.5
2
4.5
5
5.5
6
6.5
7
7.5
8
species
setosa
versicolor
virginica

Plotly

Avantages

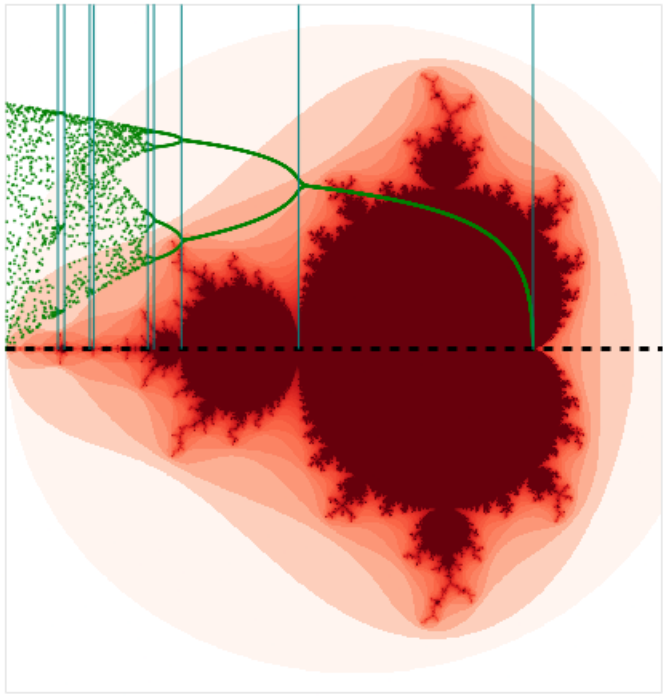
- La documentation est claire comme de l'eau de roche pour les tracés simples.
- Tableau de bord possible avec Dash (bien intégré).
- Redimensionnement automatique du graphique par défaut.

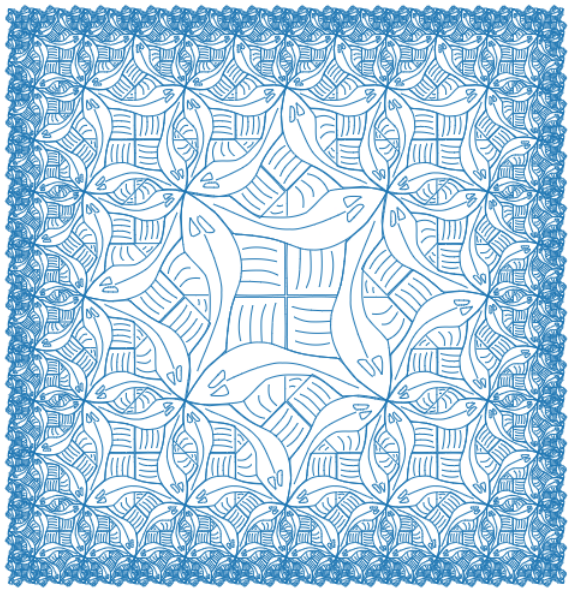
Inconvénients

- Si les serveurs de Plotly sont en panne, vos tracés peuvent ne pas s'afficher correctement (ou pas du tout).
- Difficile d'utiliser des images comme tiques.

Holoviews

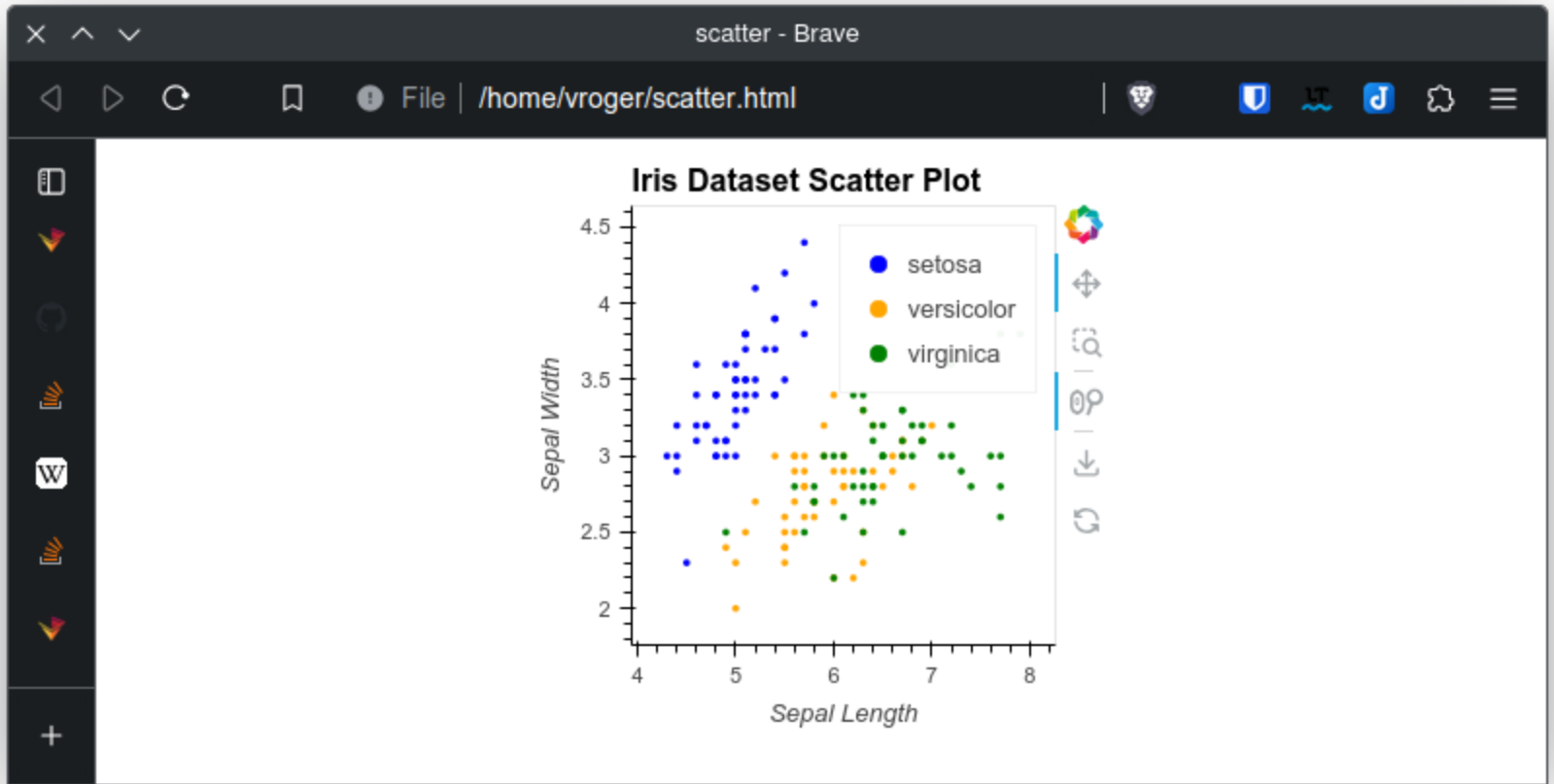
Une bibliothèque de haut niveau pour Bokeh, Matplotlib et Plotly.





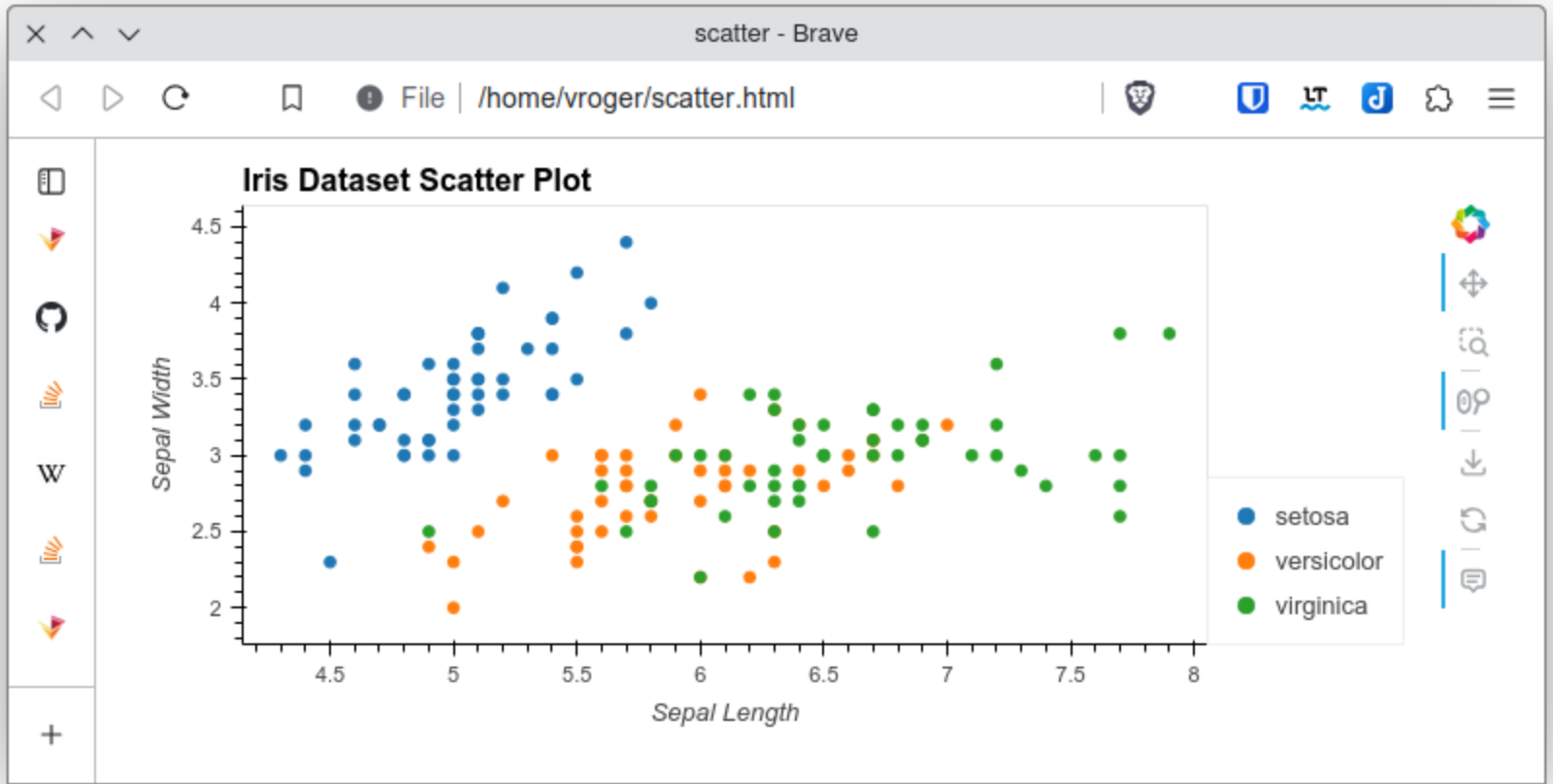
Holoviews - diagramme de dispersion des sépales de l'iris

```
import holoviews as hv

scatter = hv.Scatter(
    data=df_iris, kdims=['sepal_length'], vdims=['sepal_width', 'species'],
).opts(color='species', cmap=['blue', 'orange', 'green'])

scatter = scatter.opts(
    title="Iris Dataset Scatter Plot",
    xlabel='Sepal Length',
    ylabel='Sepal Width',
)

hv.save(scatter, 'scatter.html', backend='bokeh')
```

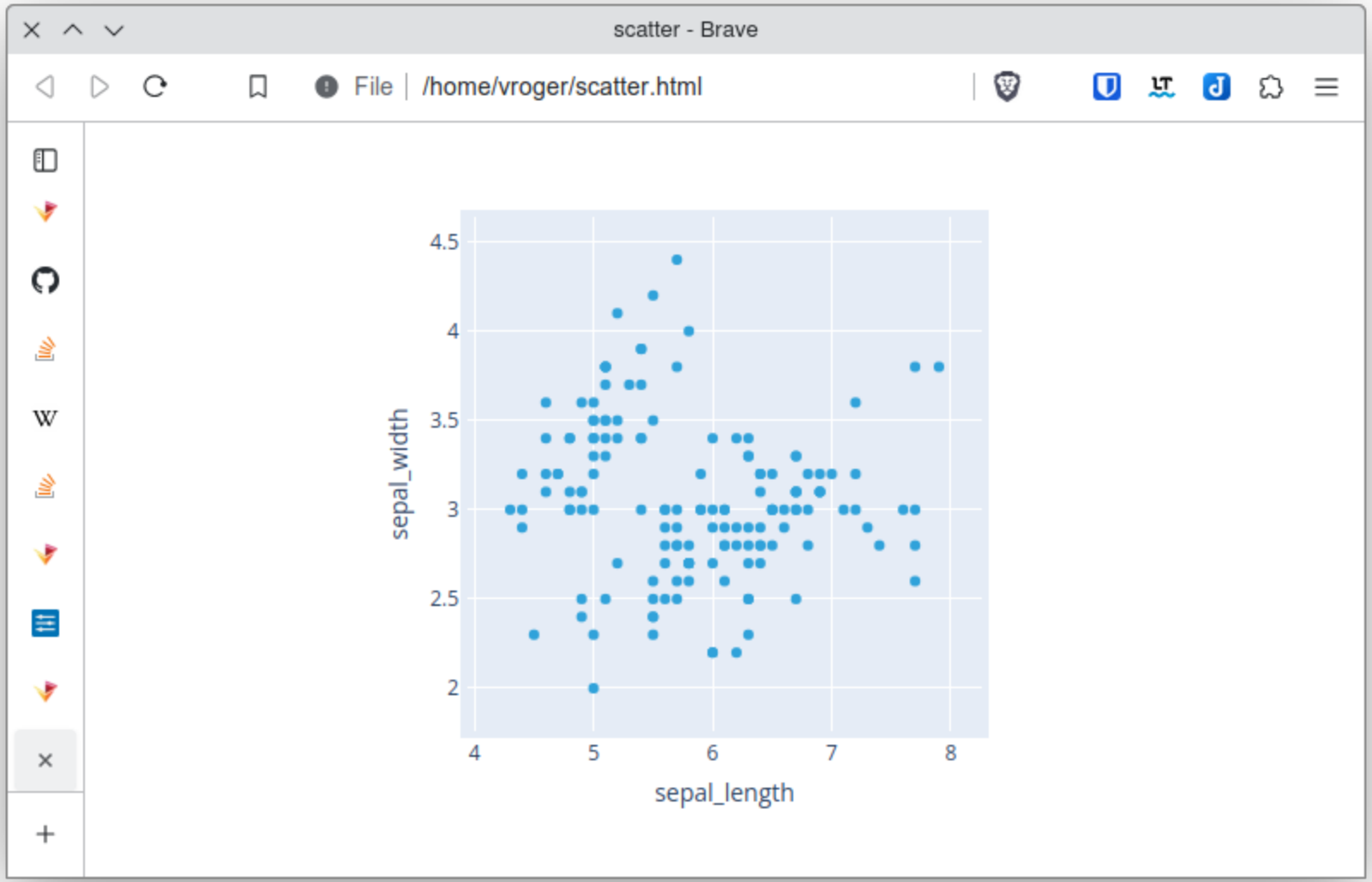

scatter - Brave
File | /home/vroger/scatter.html
Iris Dataset Scatter Plot
setosa
versicolor
virginica
Sepal Width
Sepal Length
4.5
4
3.5
3
2.5
2
4
5
6
7
8

Holoviews - diagramme de dispersion des sépales de l'iris avec l'API Pandas

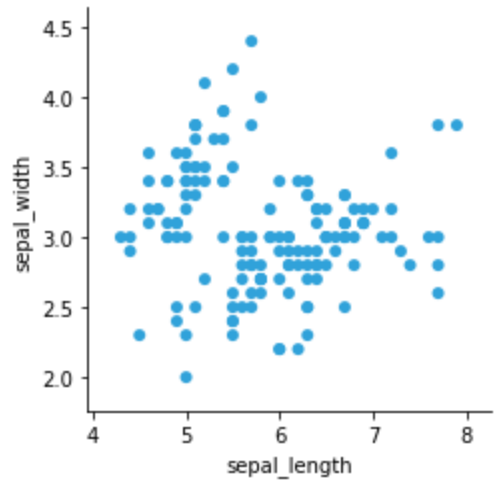
```
import hvplot
import hvplot.pandas
import holoviews as hv

scatter = df_iris.hvplot(kind='scatter', x='sepal_length', y='sepal_width', color='species')

scatter = scatter.opts(
    title="Iris Dataset Scatter Plot",
    xlabel='Sepal Length',
    ylabel='Sepal Width',
)

hv.save(scatter, 'scatter.html', backend='bokeh')
```


scatter - Brave
File | /home/vroger/scatter.html
Iris Dataset Scatter Plot
Sepal Width
Sepal Length
4.5
4
3.5
3
2.5
2
4.5
5
5.5
6
6.5
7
7.5
8
setosa
versicolor
virginica

```
hv.save(scatter, 'scatter.html', backend='plotly')
```


scatter - Brave
File | /home/vroger/scatter.html
4.5
4
3.5
3
2.5
2
sepal_width
4
5
6
7
8
sepal_length

```
hv.save(scatter, 'scatter.png', backend='matplotlib')
```


4.5
4.0
3.5
3.0
2.5
2.0
sepal_width
4
5
6
7
8
sepal_length


PAS SI GÉNÉRIQUE!
imgflip.com

Holoviews

Avantages

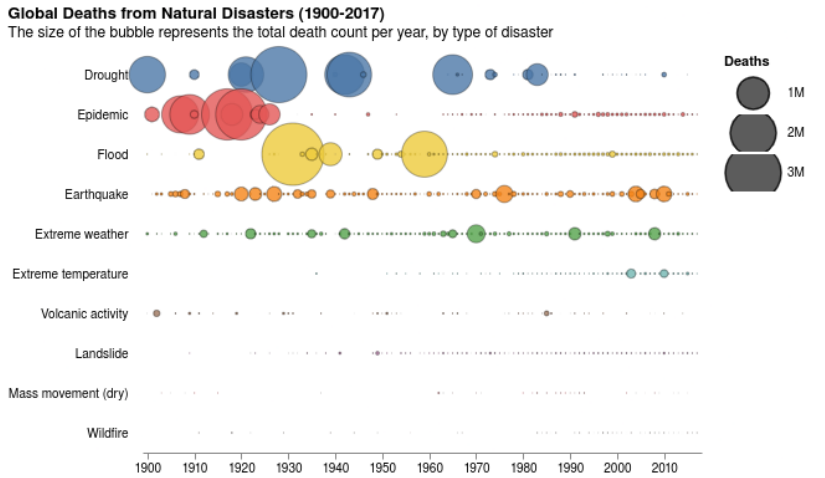
- Fonctionne bien avec Bokeh.
- API de haut niveau.

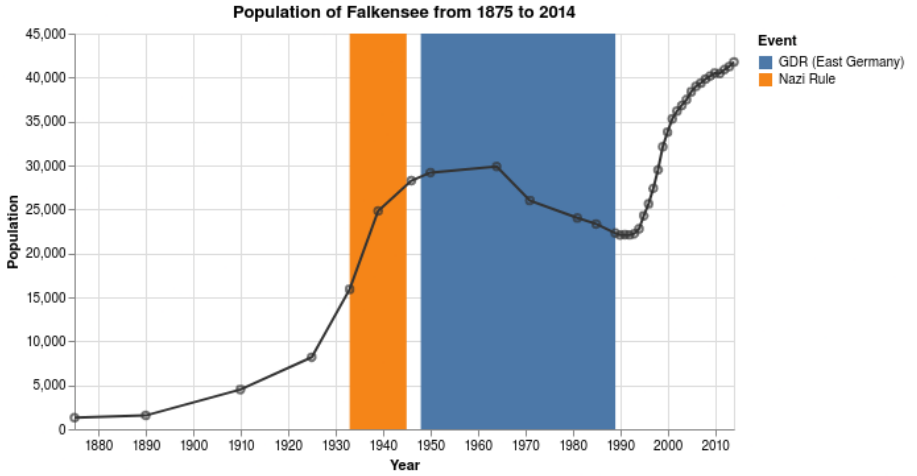
Désavantages

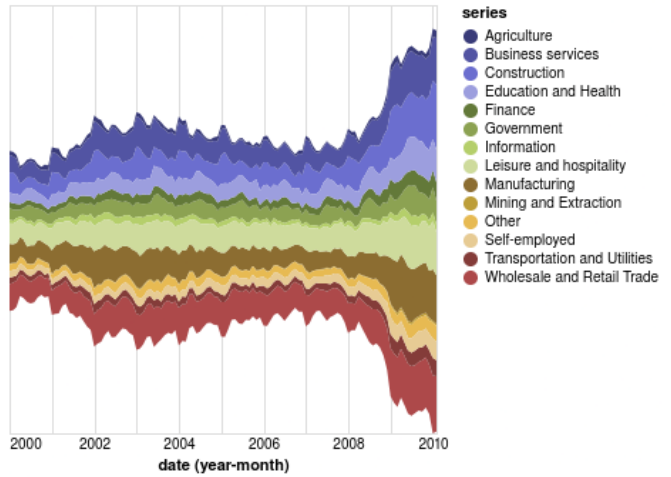
- Pas générique (du moins pour l'instant).

Altair

- Conçu pour les ordinateurs portables.
- Utilise le moteur Vega (qui produit du code JavaScript).

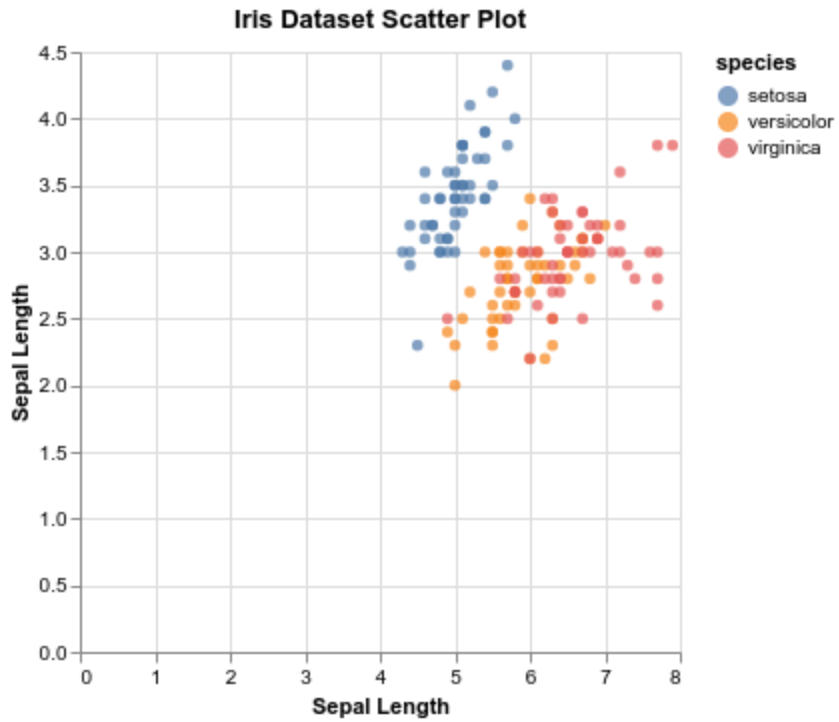






Altair - diagramme de dispersion des sépales de l'iris

```
import altair as alt

# Create the scatter plot
scatter_plot = alt.Chart(df_iris).mark_circle().encode(
    x=alt.X('sepal_length', title='Sepal Length'),
    y=alt.Y('sepal_width', title='Sepal Length'),
    color='species'
).properties(
    title='Iris Dataset Scatter Plot'
)

# Show the plot
scatter_plot
```


Iris Dataset Scatter Plot
Sepal Length
Sepal Length
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5
6
7
8
species
setosa
versicolor
virginica

Altair

Avantages

- Utilisation simple des dataframes.
- Performances plus agréables qu'un Plotly dans un notebook VS Code.

Inconvénients

- Les valeurs par défaut ne sont pas très bonnes pour les graphiques statistiques.
- Pas de visualiseur pour les dernières versions (5.x).
- Personnalisation limitée des graphiques.

Conclusion

Mon choix

Plotly

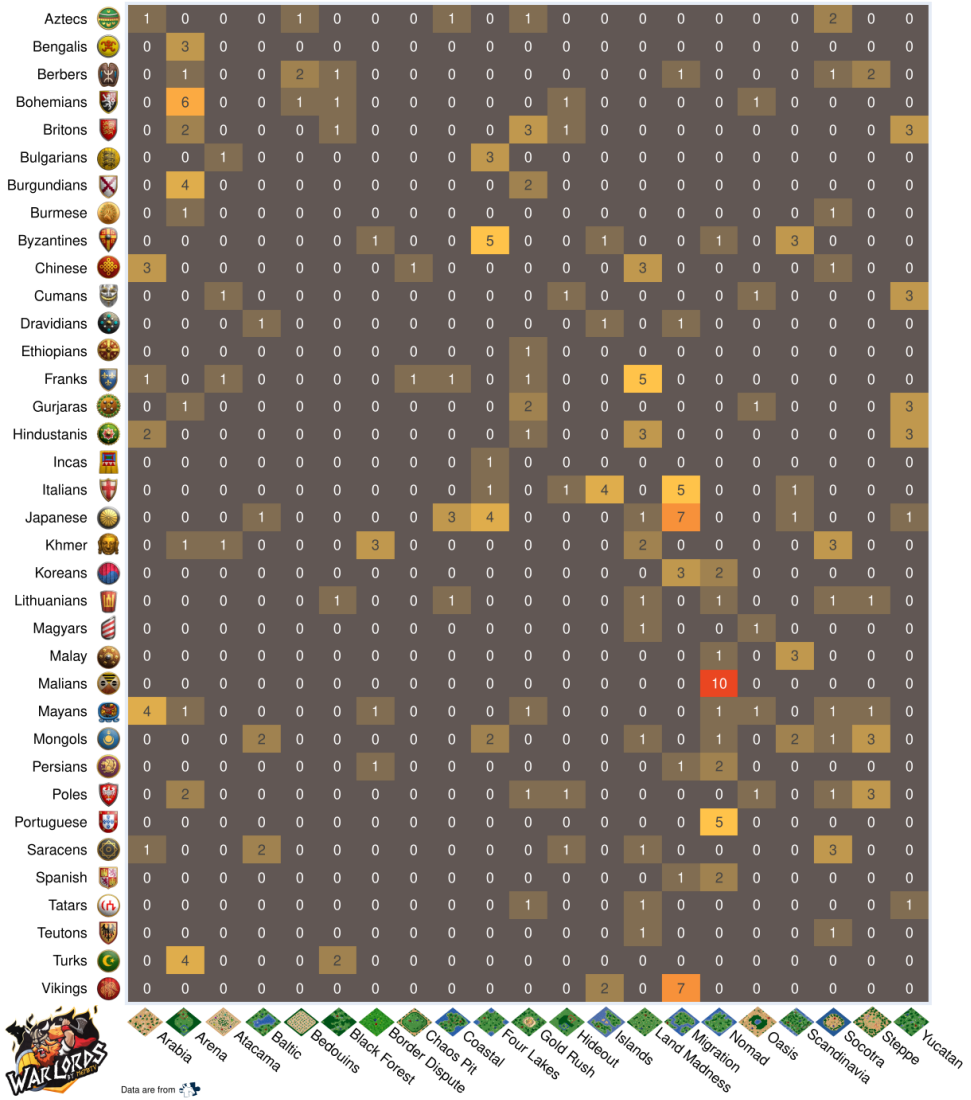
Avantages

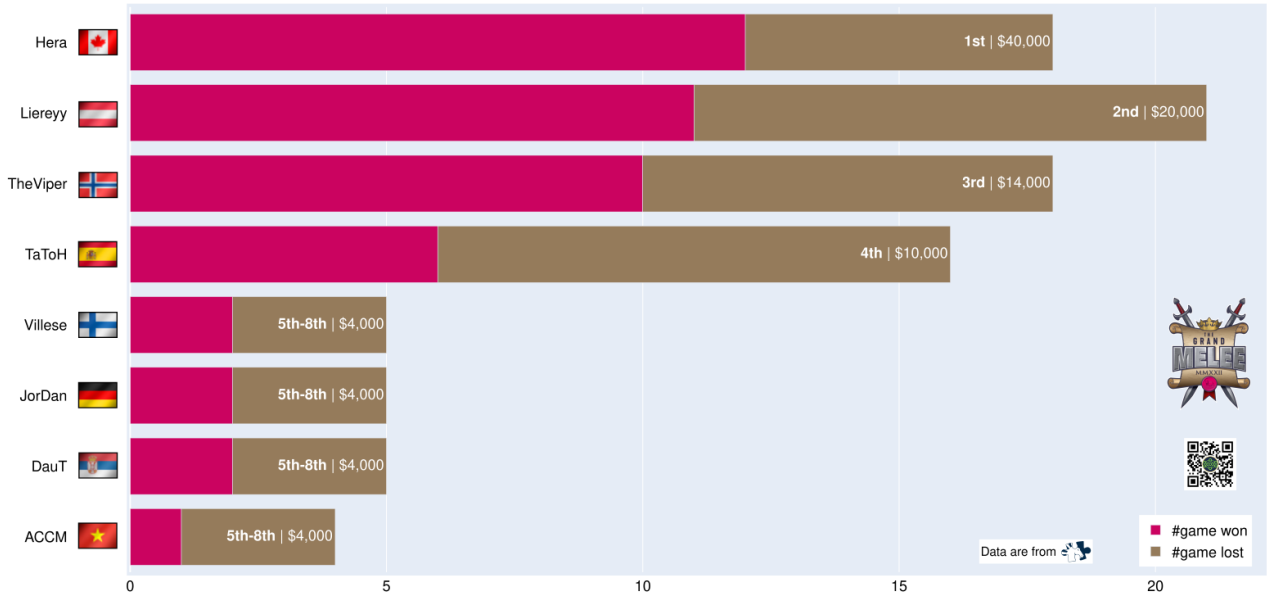
- Excellentes valeurs par défaut qui réduisent mes codes.
- Excellente documentation.
- Supporté par MLFlow.

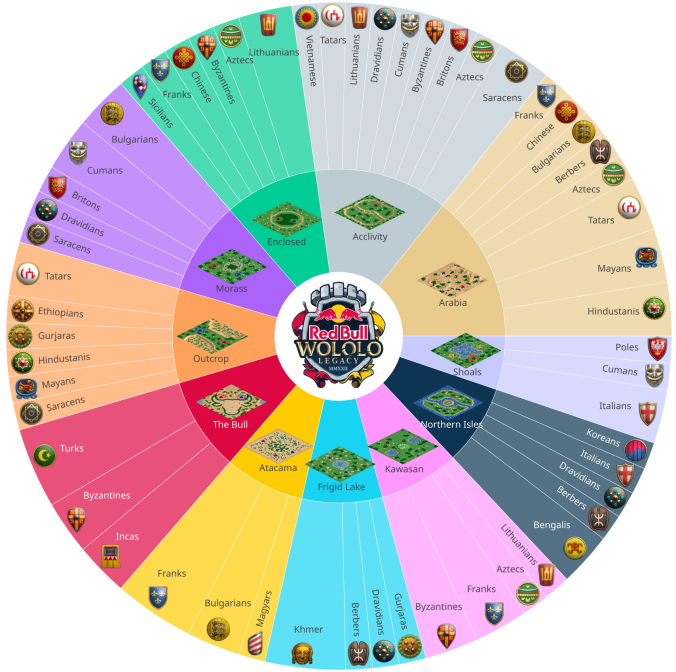
Inconvénients

- Beaucoup de tracés dans les notebooks VSCode implique des ralentissements.
- Avoir des images en tant que ticks est difficile à mettre en œuvre (si vous connaissez un moyen simple, n'hésitez pas).

Certains de mes travaux

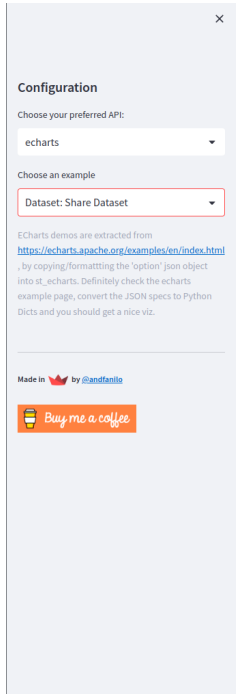


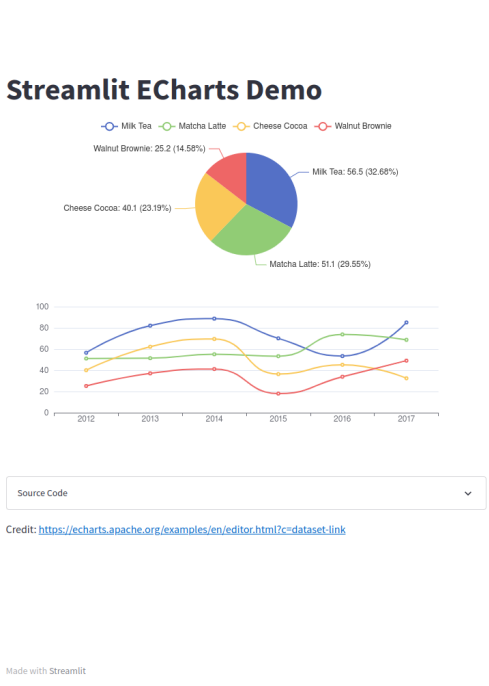




RedBull Wololo Legacy - Winning Civilizations on the Main Event

Bonus dashboard/webapp avec Streamlit





Merci d'avoir écouté !