

Informationen Visualisieren mit Python

Matplotlib & Bokeh

Informatik 1 für Biomedical Engineering

May 22, 2017

Institute for Interactive Systems and Data Science

Warum Visualisierungen?

1) Um Algorithmen und Formeln zu verstehen

$$\left(\left(\frac{x}{7} \right)^2 \sqrt{\frac{||x| - 3|}{|x| - 3}} + \left(\frac{y}{3} \right)^2 \sqrt{\frac{|y + \frac{3\sqrt{33}}{7}|}{y + \frac{3\sqrt{33}}{7}}} - 1 \right) \quad (1)$$

$$\left(\left| \frac{x}{2} \right| - \left(\frac{3\sqrt{33} - 7}{112} \right) x^2 - 3 + \sqrt{1 - (||x| - 2| - 1)^2} - y \right) \quad (2)$$

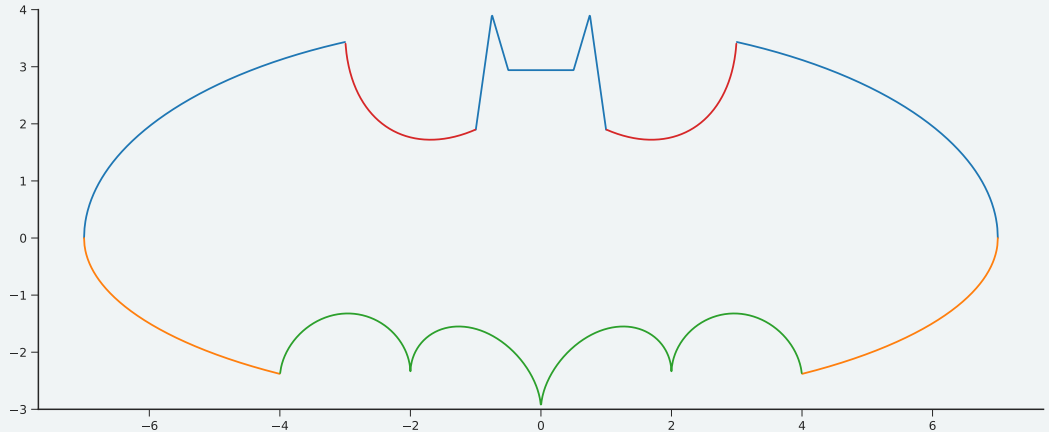
$$\left(3\sqrt{\frac{|(|x| - 1)(|x| - .75)|}{(1 - |x|)(|x| - .75)}} - 8|x| - y \right) \left(3|x| + .75\sqrt{\frac{|(|x| - .75)(|x| - .5)|}{(.75 - |x|)(|x| - .5)}} - y \right) \quad (3)$$

$$\left(2.25\sqrt{\frac{(x - .5)(x + .5)}{(.5 - x)(.5 + x)}} - y \right) \quad (4)$$

$$\left(\frac{6\sqrt{10}}{7} + (1.5 - .5|x|)\sqrt{\frac{||x| - 1|}{|x| - 1}} - \frac{6\sqrt{10}}{14}\sqrt{4 - (|x| - 1)^2} - y \right) = 0 \quad (5)$$

Warum Visualisierungen?

1) Um Algorithmen und Formeln zu verstehen



Warum Visualisierungen?

2) Um Daten zu verstehen

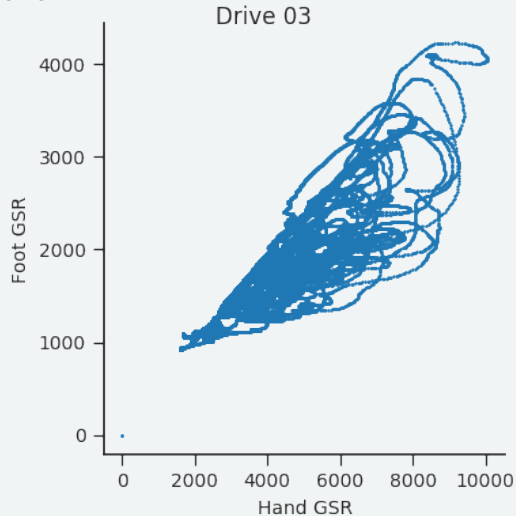
- Grundverständnis erlangen
- Erster Schritt explorativer Analysen

Beispiel: Assignment 2 - Korrelationsanalyse (`--gsr`)

- Korrelieren die Werte GSR-Werte für Hand und Fuß?
- zwei Möglichkeiten
 1. Correlation Coefficient Berechnen (z.B.: Pearson, Spearman, Kendall-Tau)
 2. In Plot Daten gegenüberstellen

Warum Visualisierungen?

2) Um Daten zu verstehen



3) Damit andere Ihre Daten verstehen

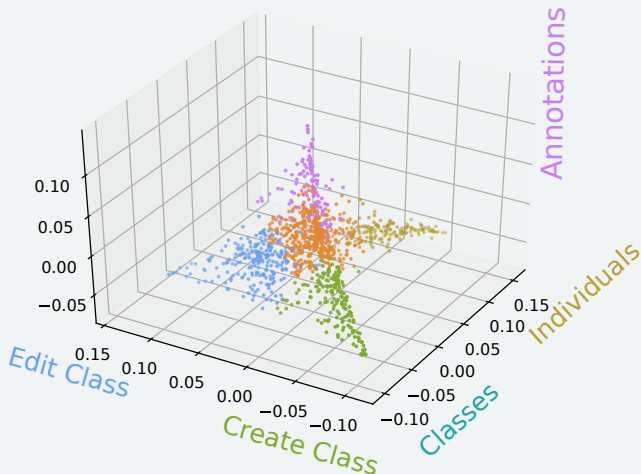
- Ergebnisse zu kommunizieren ist schwer
- Andere haben:
 - Kein Vorwissen von Ihren Daten
 - Kein Vorwissen zu dem, was Sie getan haben

3) Damit andere Ihre Daten verstehen

- Ergebnisse zu kommunizieren ist schwer
- Andere haben:
 - Kein Vorwissen von Ihren Daten
 - Kein Vorwissen zu dem, was Sie getan haben
 - **Wenig Zeit!**

Warum Visualisierungen?

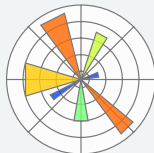
3) Damit andere Ihre Daten verstehen



- Python hat kein eingebautes Package (anders als R)

- Python hat kein eingebautes Package (anders als R)
- Viele offene Packages füllen dieses Loch

- Python hat kein eingebautes Package (anders als R)
- Viele offene Packages füllen dieses Loch
 - Bindings für darunterliegendes Framework



- Python hat kein eingebautes Package (anders als R)
- Viele offene Packages füllen dieses Loch
 - Bindings für darunterliegendes Framework
 - jupyter Support

Typische Vorgangsweise

Einfacher 3-Step-Plan

1

2

3

[illegible]

Some random Data

Think how to visualize

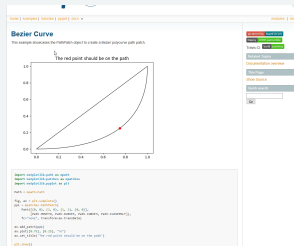
Typische Vorgangsweise

Einfacher 3-Step-Plan

1

2

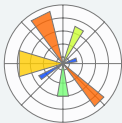
3

[illegible]

Some random Data

Think how to visualize

[Go to documentation](#)



Matplotlib

<https://matplotlib.org>

- + Weit verbreitet
- + Umfangreiche Dokumentation
- + Viele Tutorials
- + Viele Erweiterungen
 - z.B.: Seaborn
- Wirkt zT altmodisch
- Nativ kaum Interaktivität
- Feinanpassung umständlich

```
1 pip install matplotlib
```



Bokeh

<https://bokeh.pydata.org>

- + Sehr modern
- + Auf Web spezialisiert
 - Overlays
 - Interaktionen
- + Modernes Plot Design
- + Wenig anpassung notwendig
- Noch sehr *"in development"*
- Tutorials noch unausgereift

```
1 pip install bokeh
```



Matplotlib - Basics

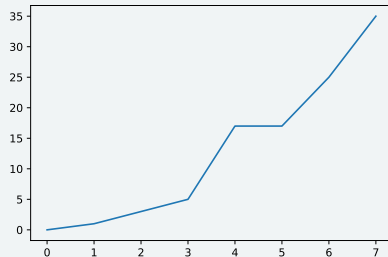
Import in Python:

```
1 from matplotlib import pyplot as plt
```



"My First Plot:"

```
1 # Uncomment below for notebooks
2 # %matplotlib inline
3
4 from matplotlib import pyplot as plt
5
6 plt.plot([0, 1, 3, 5, 17, 17, 25, 35])
7 plt.show()
```



Matplotlib - Basics

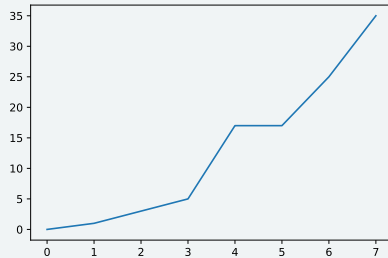
Import in Python:

```
1 from matplotlib import pyplot as plt
```



“My First Plot:” (Clean Version)

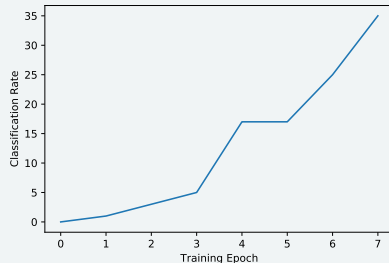
```
1 # Uncomment below for notebooks
2 # %matplotlib inline
3
4 from matplotlib import pyplot as plt
5
6 figure, axis = plt.subplots()
7 ys = [1,2,3,5,6,3]
8 xs = np.arange(len(ys))
9 axis.plot(xs, ys)
10 figure.show()
```



Plots ohne Achsenbeschriftung sind wertlos!

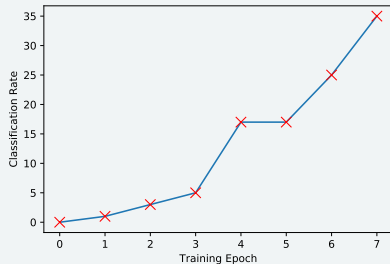
- Jedes axis Objekt kann eine Beschriftung auf X und Y Achse haben
- `set_xlabel()` und `set_ylabel()`
 - String mit Bezeichnung der Achse als Parameter
- Unterstützt $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ - `math` Encoding mit `$<text>$`
 - `μ` für μ
 - `σ` für σ

```
1 # ...  
2 axis.plot(xs, ys)  
3 axis.set_xlabel("Training_Epoch")  
4 axis.set_ylabel("Classification_Rate")  
5 figure.show()
```



- Markieren die einzelnen Werte bei Linienplots
- Helfen mehrere Linien zu unterscheiden
- Optionen: https://matplotlib.org/api/markers_api.html

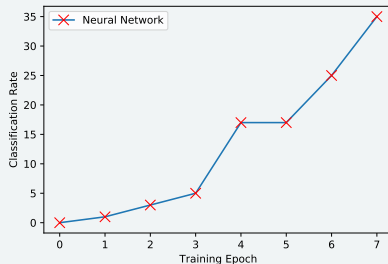
```
1 # ...  
2 axis.plot(xs, ys,  
3           marker="x",  
4           markersize=10,  
5           markeredgcolor="red")  
6 axis.set_xlabel("Training_Epoch")  
7 axis.set_ylabel("Classification_Rate")  
8 figure.show()
```



Matplotlib - Legende

- Mit Legenden geben Sie jeder Linie eine Bezeichnung
- Helfen mehrere Linien zu unterscheiden
- Automatische Platzierung dort wo matplotlib es für am Besten hält
 - Außerhalb des Plots zu platzieren ist umständlich!

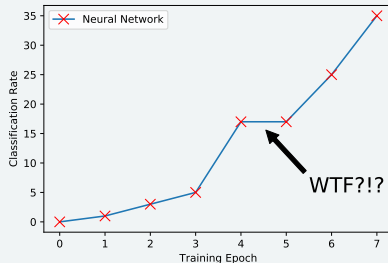
```
1 # ...  
2 axis.plot(xs, ys,  
3           marker="x",  
4           markersize=10,  
5           markeredgecolor="red",  
6           label="Neural_Network")  
7 axis.set_xlabel("Training_Epoch")  
8 axis.set_ylabel("Classification_Rate")  
9 axis.legend()  
10 figure.show()
```



Matplotlib - Annotierungen

- Text und Pfeile am Plot
- Helfen Besonderheiten hervorzuheben

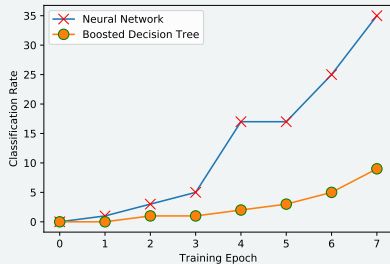
```
1 # ...
2 axis.plot(xs, ys,
3           marker="x",
4           markersize=10,
5           markeredgecolor="red",
6           label="Neural_Network")
7 axis.annotate('WTF?!?',
8              xy=(4.5, 16),
9              xytext=(5.5, 5),
10             arrowprops=dict(facecolor='black',
11                             shrink=0.05),
12             fontsize=20 )
13 axis.set_xlabel("Training_Epoch")
14 axis.set_ylabel("Classification_Rate")
15 axis.legend()
16 figure.show()
```



Matplotlib - Mehrere Plots auf einer Achse

- Achsen unterstützen beliebig viele plots übereinander
- Z-Reihenfolge je nach Reihenfolge in Code
 - Marker am Ende erneut Plotten?
- Übersichtlichkeit beachten

```
1 # ...
2 axis.plot(xs, ys,
3           marker="x",
4           markersize=10,
5           markeredgcolor="red",
6           label="Neural_Network")
7 axis.plot(xs, [0, 0, 1, 1, 2, 3, 5, 9],
8           marker="o",
9           markersize=10,
10          markeredgcolor="green",
11          label="Boosted_Decision_Tree")
12 # ...
```



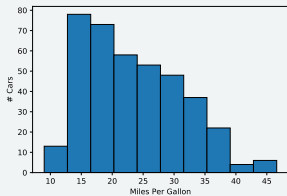
Speichern von Plots

- Speichern über **figure**-Objekt
- Typ definiert durch Dateiname
- Tipp: Speichern als PDF und als PNG
- Bokeh speichert als HTML(+ JavaScript)
 - Typ definiert durch `figure.output_backend("png", "svg")`
 - Direkter export zu Bilddatei benötigt PhantomJS

```
1  # ...
2  figure.savefig("rate_over_epochs.png",
3                transparent=True,
4                bbox_inches="tight")
5  figure.savefig("rate_over_epochs.pdf",
6                transparent=True,
7                bbox_inches="tight")
```

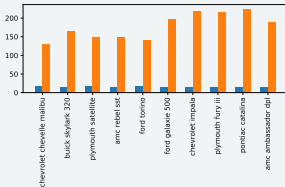


Andere häufige Plottypen



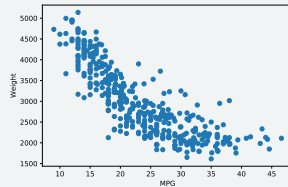
Histogramme

`matplotlib.pyplot.hist()`



Barcharts

`matplotlib.pyplot.bar()`



Scatterplots

`matplotlib.pyplot.scatter()`

- Korrekte Farbwahl ist wichtig
- Beschränkte Farben
 - Schwarz-Weiß Druck
 - Photokopierer
 - Schlechte Displays
 - Farbenblind Menschen

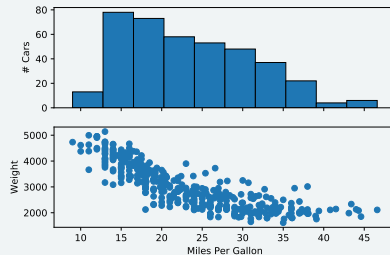
Einfache Lösung:

1. Colorbrewer <http://colorbrewer2.org/>
2. Matplotlib Colormaps: <https://matplotlib.org/users/colormaps.html>

Subplots

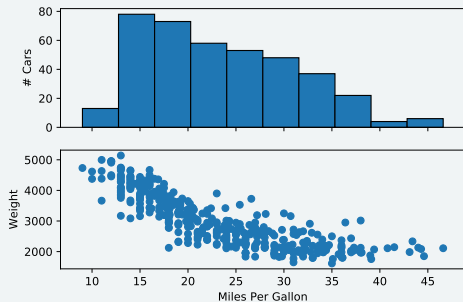
- Mehrere Plots in einer Figure
- Parameter in `subplots()`
 - Achsen Objekt ist nun Array!
- X und Y Achsen können geteilt werden
 - `sharex` und `sharey` Keyword-Parameter

```
1 # ...
2 figure, axes = plt.subplots((2),
3                             sharex=True)
4 axes[0].hist(mpgs)
5 axes[0].set_ylabel("$\#Cars")
6
7 axes[1].scatter(mpgs, weights)
8 axes[1].set_xlabel("Miles Per Gallon")
9 axes[1].set_ylabel("Weight")
10 figure.tight_layout()
11 # ...
```

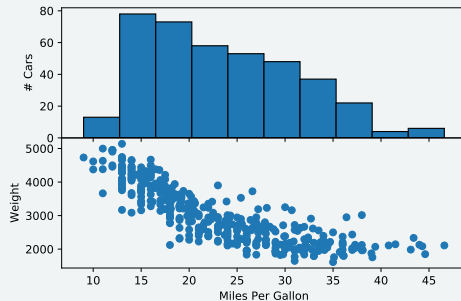


Subplots - Spacing

Mithilfe von `subplots_adjust()`¹



Default



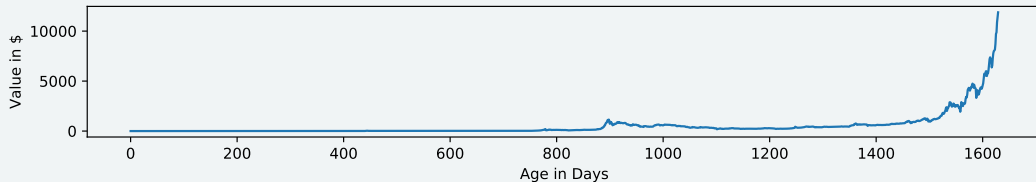
`subplot_adjust(hspace=0)`

¹ Documentation `matplotlib.pyplot.subplots_adjust(*args, **kwargs)`

- Definiert die numerischen Enden der Achsen
- Erlauben Fokus auf einen Ausschnitt
- Oft verwendet um Eindrücke zu manipulieren
 - Gute Plots beinhalten fast immer den Nullpunkt!
- Achsenfunktionen
 - `.set_xlim(<start>, <stop>)`
 - `.set_ylim(<start>, <stop>)`

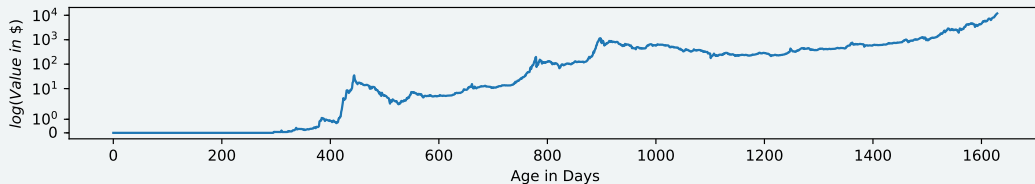
Subplots - Log-Scaling

- Ändert die Skalierung der Achsen auf \log_{10}
- Gut, um große Bandbreiten darzustellen
- Achsenlabels müssen immer darauf hinweisen!



Subplots - Log-Scaling

- Ändert die Skalierung der Achsen auf \log_{10}
- Gut, um große Bandbreiten darzustellen
- Achsenlabels müssen immer darauf hinweisen!



Ein letzter Trick: Seaborn

- Seaborn <http://seaborn.pydata.org/>
- Eigenes python package `pip install seaborn`
- Überschreibt einige Matplotlib Default Settings
- Viele neue Plot-Typen
- Funktioniert mit pandas dataframes und datagrids
 - In Zukunft relevant...

“matplotlib tries to make easy things easy and hard things possible”