

# 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

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

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

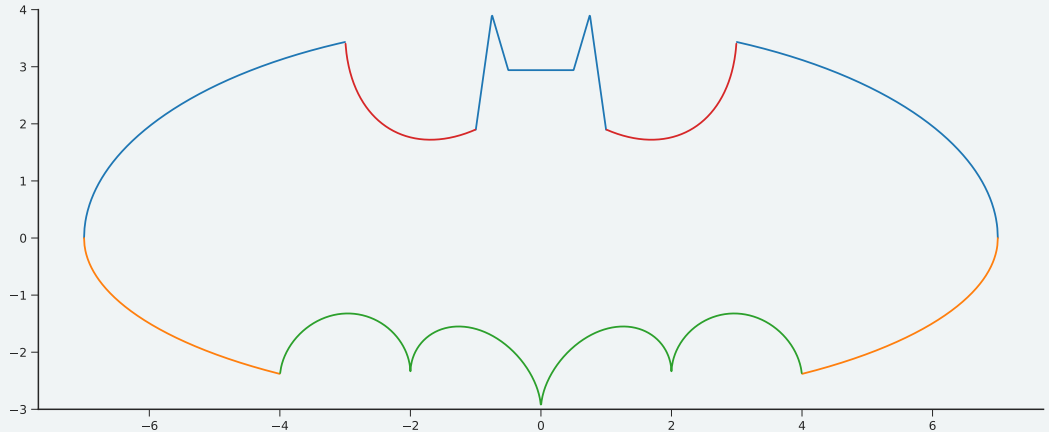
$$\left( 3\sqrt{\frac{|(|x| - 1)(|x| - .75)|}{(1 - |x|)(|x| - .75)}} - 8|x| - y \right) \left( 3|x| + .75\sqrt{\frac{|(|x| - .75)(|x| - .5)|}{(.75 - |x|)(|x| - .5)}} - y \right) \quad (3)$$

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

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

# Warum Visualisierungen?

## 1) Um Algorithmen und Formeln zu verstehen



# Warum Visualisierungen?

## 2) Um Daten zu verstehen

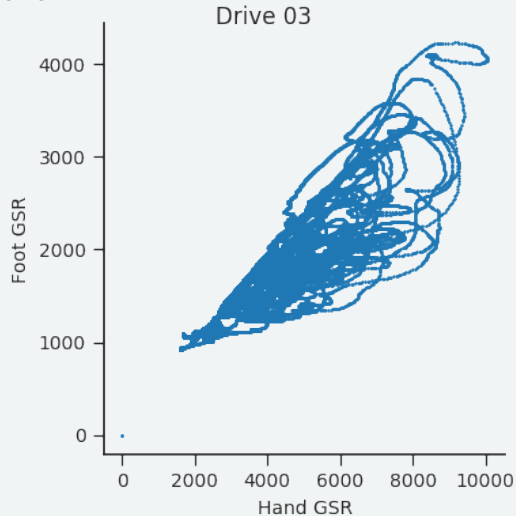
- Grundverständnis erlangen
- Erster Schritt explorativer Analysen

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

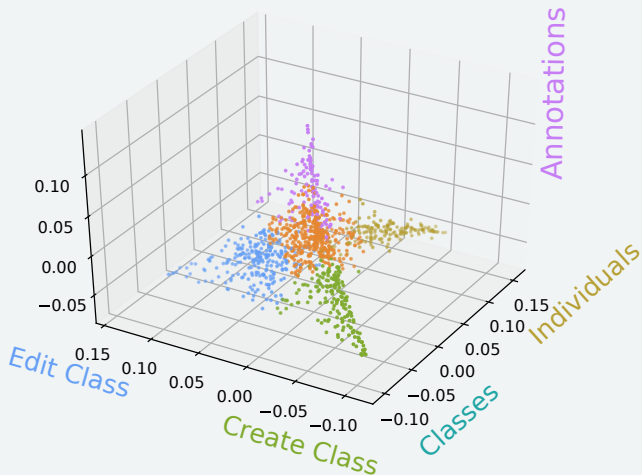
- Ergebnisse zu kommunizieren ist schwer
- Andere haben:
  - Kein Vorwissen von Ihren Daten
  - Kein Vorwissen zu dem, was Sie getan haben

## 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

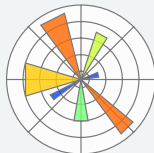




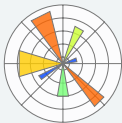
- Python hat kein eingebautes Package (anders als R)

- 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



## 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

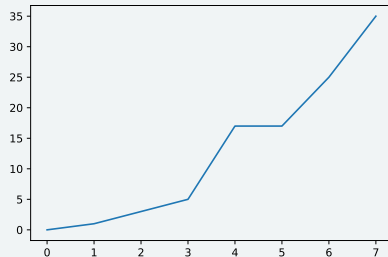
Import in Python:

```
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()
```



## Typische Vorgangsweise

## Einfacher 3-Step-Plan

1

2

3

[illegible]

## Some random Data

## 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

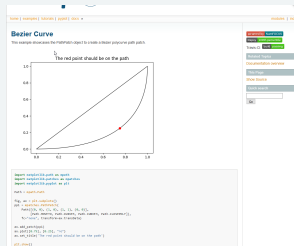


Imported Values

Temp	50.0	55.0	60.0	65.0	70.0	75.0	80.0	85.0	90.0	95.0	100.0
0.00	2045.00	2050.00	2100.00	2150.00	2200.00	2250.00	2300.00	2350.00	2400.00	2450.00	2500.00
0.05	2047.00	2052.00	2102.00	2152.00	2202.00	2252.00	2302.00	2352.00	2402.00	2452.00	2502.00
0.10	2049.00	2054.00	2104.00	2154.00	2204.00	2254.00	2304.00	2354.00	2404.00	2454.00	2504.00
0.15	2051.00	2056.00	2106.00	2156.00	2206.00	2256.00	2306.00	2356.00	2406.00	2456.00	2506.00
0.20	2053.00	2058.00	2108.00	2158.00	2208.00	2258.00	2308.00	2358.00	2408.00	2458.00	2508.00
0.25	2055.00	2060.00	2110.00	2160.00	2210.00	2260.00	2310.00	2360.00	2410.00	2460.00	2510.00
0.30	2057.00	2062.00	2112.00	2162.00	2212.00	2262.00	2312.00	2362.00	2412.00	2462.00	2512.00
0.35	2059.00	2064.00	2114.00	2164.00	2214.00	2264.00	2314.00	2364.00	2414.00	2464.00	2514.00
0.40	2061.00	2066.00	2116.00	2166.00	2216.00	2266.00	2316.00	2366.00	2416.00	2466.00	2516.00
0.45	2063.00	2068.00	2118.00	2168.00	2218.00	2268.00	2318.00	2368.00	2418.00	2468.00	2518.00
0.50	2065.00	2070.00	2120.00	2170.00	2220.00	2270.00	2320.00	2370.00	2420.00	2470.00	2520.00
0.55	2067.00	2072.00	2122.00	2172.00	2222.00	2272.00	2322.00	2372.00	2422.00	2472.00	2522.00
0.60	2069.00	2074.00	2124.00	2174.00	2224.00	2274.00	2324.00	2374.00	2424.00	2474.00	2524.00
0.65	2071.00	2076.00	2126.00	2176.00	2226.00	2276.00	2326.00	2376.00	2426.00	2476.00	2526.00
0.70	2073.00	2078.00	2128.00	2178.00	2228.00	2278.00	2328.00	2378.00	2428.00	2478.00	2528.00
0.75	2075.00	2080.00	2130.00	2180.00	2230.00	2280.00	2330.00	2380.00	2430.00	2480.00	2530.00
0.80	2077.00	2082.00	2132.00	2182.00	2232.00	2282.00	2332.00	2382.00	2432.00	2482.00	2532.00
0.85	2079.00	2084.00	2134.00	2184.00	2234.00	2284.00	2334.00	2384.00	2434.00	2484.00	2534.00
0.90	2081.00	2086.00	2136.00	2186.00	2236.00	2286.00	2336.00	2386.00	2436.00	2486.00	2536.00
0.95	2083.00	2088.00	2138.00	2188.00	2238.00	2288.00	2338.00	2388.00	2438.00	2488.00	2538.00
1.00	2085.00	2090.00	2140.00	2190.00	2240.00	2290.00	2340.00	2390.00	2440.00	2490.00	2540.00

Some random Data

Think how to visualize



Go to documentation

# Matplotlib - Basics

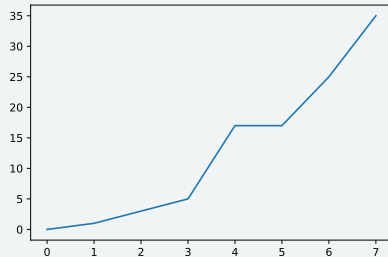
Import in Python:

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



*“My First Plot:” (Clean Version)*

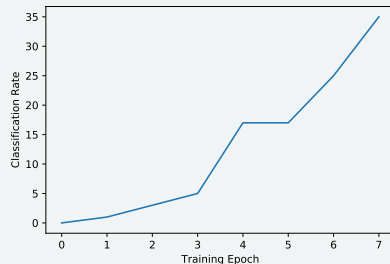
```
1 # Uncomment below for notebooks
2 # %matplotlib inline
3
4 from matplotlib import pyplot as plt
5
6 figure, axis = plt.subplots()
7 ys = [1,2,3,5,6,3]
8 xs = np.arange(len(ys))
9 axis.plot(xs, ys)
10 figure.show()
```



## Plots ohne Achsenbeschriftung sind wertlos!

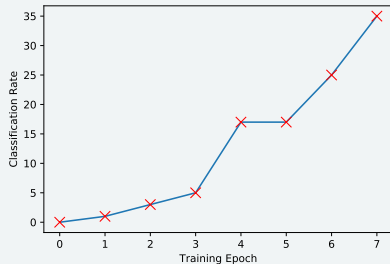
- Jedes axis Objekt kann eine Beschriftung auf X und Y Achse haben
- `set_xlabel()` und `set_ylabel()`
  - String mit Bezeichnung der Achse als Parameter
- Unterstützt  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ - `math` Encoding mit `$<text>$`
  - `$\mu$` für  $\mu$
  - `$\sigma$` für  $\sigma$

```
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](https://matplotlib.org/api/markers_api.html)

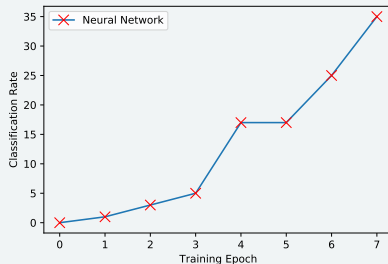
```
1 # ...  
2 axis.plot(xs, ys,  
3           marker="x",  
4           markersize=10,  
5           markeredgcolor="red")  
6 axis.set_xlabel("Training_Epoch")  
7 axis.set_ylabel("Classification_Rate")  
8 figure.show()
```



# 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!

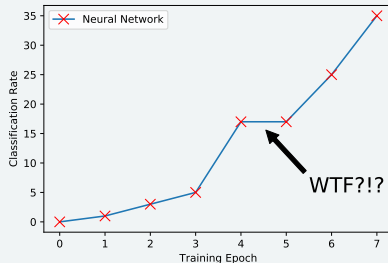
```
1 # ...
2 axis.plot(xs, ys,
3           marker="x",
4           markersize=10,
5           markeredgecolor="red",
6           label="Neural_Network")
7 axis.set_xlabel("Training_Epoch")
8 axis.set_ylabel("Classification_Rate")
9 axis.legend()
10 figure.show()
```



# Matplotlib - Annotierungen

- Text und Pfeile am Plot
- Helfen Besonderheiten hervorzuheben

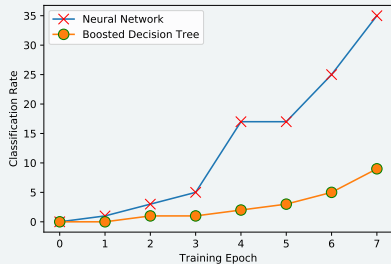
```
1 # ...
2 axis.plot(xs, ys,
3           marker="x",
4           markersize=10,
5           markeredgecolor="red",
6           label="Neural_Network")
7 axis.annotate('WTF?!?',
8              xy=(4.5, 16),
9              xytext=(5.5, 5),
10             arrowprops=dict(facecolor='black',
11                             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

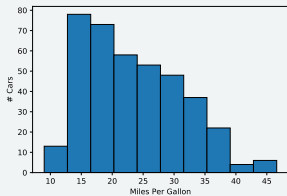
- Speichern über **figure**-Objekt
- Typ definiert durch Dateiname
- Tipp: Speichern als PDF und als PNG
- Bokeh speichert als HTML(+ JavaScript)
  - Typ definiert durch `figure.output_backend("png", "svg")`
  - Direkter export zu Bilddatei benötigt PhantomJS

```
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")
```



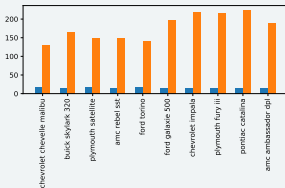


# Andere häufige Plottypen



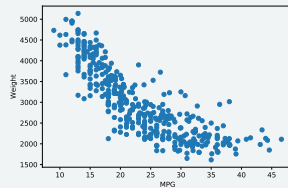
Histogramme

`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

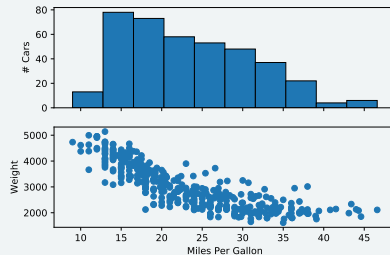
Einfache Lösung:

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

# Subplots

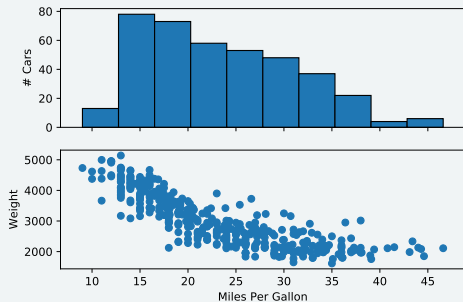
- Mehrere Plots in einer Figure
- Parameter in `subplots()`
  - Achsen Objekt ist nun Array!
- X und Y Achsen können geteilt werden
  - `sharex` und `sharey` Keyword-Parameter

```
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 # ...
```

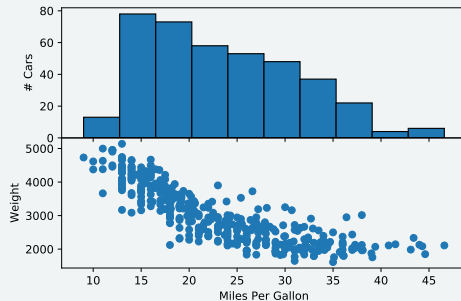


# Subplots - Spacing

Mithilfe von `subplots_adjust()`<sup>1</sup>



Default



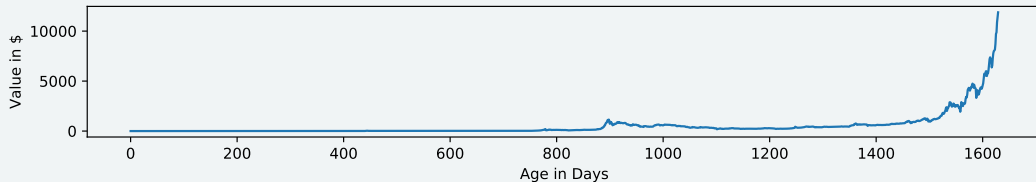
`subplot_adjust(hspace=0)`

<sup>1</sup> 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>)`

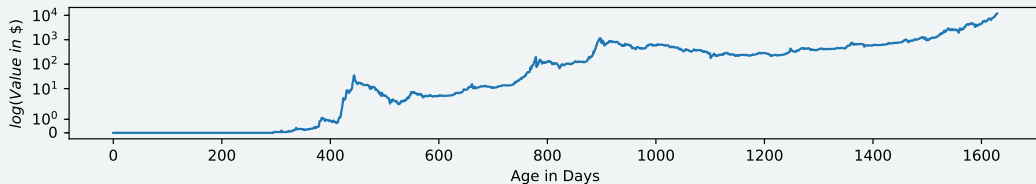
## Subplots - Log-Scaling

- 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”*