

Python Basics

<code>import numpy as np</code>	
<code>%matplotlib notebook, %matplotlib inline</code>	interaktive Plots
<code>a = np.array([0, 1, 2, 3, 4])</code>	
<code>data[a,b]</code>	zweidimensionales Array, gibt Werte mit Koordinaten (a,b) zurück
<code>print(a[1:3])</code>	[start :ende]
<code>b = a + 1</code>	mult/add array
<code>a = np.zeros(5)</code>	5 x 0.0
<code>a = np.ones(5)</code>	
<code>a = np.linspace(a_min, a_max, n_a)</code>	$d_a = 1/(n_a - 1)$
<code>n_points = np.linspace(1, np.log10(len(data)), 100, dtype=int)</code>	logarithmisches array # wir wollen 100 zahlen
<code>a = np.arange(a_min, a_max, d_a)</code>	$n_a = (a_{max} - a_{min}) / d_a$ (aufgerundet, array ohne a_{max})
<code>a = [0, math.pi, 0] b = np.exp(a)</code>	Funktion auf Array anwenden
<code>np.loadtxt?</code>	Dokumentation einer Funktion
<code>print('Die Summe der Zahlen {} und {} ist gleich {}'.format(3, 5, 8))</code>	{:d} : Integer {:f} : Fließkomma {:0.2f}: Fließkomma mit 2 Nachkommastellen
<code>np.array([np.fromstring('1', dtype=float, sep=',')])</code>	String Daten in Array (l ist String)
<code>np.stack((a, b))</code>	Matrix mit Zeilen (a,b)
<code>a.to_numpy()</code>	Array zu np-Array



By **moeregrind8645**

Not published yet.

Last updated 10th August, 2023.

Page 1 of 13.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

Python Basics (cont)

```
X = np.reshape(df['x'].values, newshape=(-1, 1)) # (-1, 1) bedeutet: eine Spalte mit sovielen Zeilen
nötig
```

Iteration und Enumerate

```
for c_i in c:
    print(c_i)
for index, wert in enumerate(a):
    b[index] = wert**2
```

0.0 0.25 0.5 0.75 1.0

Wir möchten jedem b[i] den Wert a[i]**2 zuweisen.

Plotten eines Datensatzes

```
import matplotlib.pyplot as plt
# Umrechnung von Zoll in cm (1 Zoll = 2.54 cm)
cm = 0.393701
# Definition von globalen Parametern, hier die Schriftgröße
plt.rcParams.update({'font.size': 16})
plt.rcParams.update({'axes.labelsize': 16})
# Definition der Figur, mit Größenangabe
fig = plt.figure(figsize=(20, 17))
ax1 = fig.add_subplot(1, 1, 1) (anzahl, (x,y))
# Definition der zu plottenden Daten mit Label
ax1.plot(x0, y0, 'bo', label='Data') #b = blue, o = Punkte
ax1.plot(x0, y0, 'r-', label=r'$y = x^2$') #Rote Striche
# Achsenbeschriftung, die Angegebene Schriftgröße überschreibt die globale Schriftgröße
```



By [moeregrind8645](https://cheatography.com/moeregrind8645/)

Not published yet.

Last updated 10th August, 2023.

Page 2 of 13.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>

Plotten eines Datensatzes (cont)

```
> ax1.set_xlabel('x (x unit)')
ax1.set_ylabel('y (y unit)')
ax1.set_title("Titel")
# Definition des Bereichs der geplottet werden soll
ax1.set_xlim(0, 2)
ax1.set_ylim(0, 4)
# Erzeugen einer Legende, hier werden die Label aus der Plotfunktion übernommen
ax1.legend(fontsize=20, frameon=False)
# Abspeichern der Figur
fig.savefig(r'Figur.png', dpi=400, transparent=True)
```

Plotbefehle

<code>ax3.bar(bins[:-1], hist, width= 0.8 *bin size)</code>	Balkendiagramm
<code>ax.axvline(V_mean, color='r')</code>	Linie einzeichnen
<code>fig.tight_layout()</code>	schönes Layout
<code>ax.plot(timel, signall, 'x-')</code>	gemessene Datenpunkte markieren
<code>plt.scatter(t_mess, U_mess, zorder=2, color = " red ", marker ='o')</code>	Punkte
<code>plt.axes().set_aspect ('auto', 'datalim')</code>	
<code>ax.legend()</code>	Legende erstellen
<code>ax.semilogx(n_points, mean_n)</code>	logarithmische x-achse
<code>ax.set_yscale("log")</code>	# Logarithmische Skalierung der y-Achse
<code>ax1.errorbar(x, y, x_error, y_error)</code>	Graph mit Fehler
<code>ax3.plot(t, z, '.')</code>	Punkte plotten



By **moeregrind8645**

Not published yet.

Last updated 10th August, 2023.

Page 3 of 13.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

Abspeichern und Laden von Daten

SPEICHERN

```
np.savetxt('Dat aOu t.txt', (x0, y0), delimiter = ',') #Speichert Daten als ASCII (txt)
np.savetxt('Dat aOu t.txt', (x,y,z), delimiter=',', header='B eis pie ldatei Datena nalyse,
14.2.2023. Format: x, y, z') #Für Metadaten
```

LADEN

```
(x2, y2) = np.loadtxt('Dat aOu t.txt', delimiter = ', ', comments = '#') #lädt Daten, blendet # aus
data = np.loadtxt('Dat aOu t.txt', delimiter = ',') #Daten von allg. Datenarray
print(data.shape) # a=np.array([[ [0 ], [ 0]]) hat shape (1,2,1) (3-dimensional)
x3 = data[0, :]
y3 = data[1, :]
plt.plot(x3, y3)
zeilen, spalten = a.shape
```

Mit 'shape' erfahren wir die Form. Diese Funktion gibt ein Tupel zurück mit den Längen der einzelnen Dimensionen des Arrays

zeilenweise Lesen und Speichern

LADEN

```
with open(file, 'r') as f:
    lines = f.readlines()
#lines ist ein String Array mit allen Daten
d = np.zeros((len(lines)-1, 2))
i = 0
for l in lines[1:]:
    d[i,:] = np.array([np.float(l), np.float(l)])
    i = i+1
```

SPEICHERN



By [moeregrind8645](#)

Not published yet.

Last updated 10th August, 2023.

Page 4 of 13.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>

zeilenweise Lesen und Speichern (cont)

```
> with open('DataOut.txt', 'w') as f:
    f.writelines('x, y, z\n') #schreibe header
    for i in range(0, len(x)):
        f.writelines('{:0.5f}, {:0.5f}, {:0.5f}\n'.format(x[i], y[i], z[i]))
```

Lineare und logistische Regression

```
IMPORTS
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
# Daten in Trainingsset und Testset aufteilen
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Regressionsmodell trainieren, d.h. Koeffizienten optimieren.
linreg = LinearRegression()
linreg.fit(X_train, y_train)
#analog für logistische Regression
# Logistische Regression
lr = LogisticRegression()
lr.fit(X_train, y_train)
# Measure the quality of the regression in the test data.
y_pred = linreg.predict(X_test)
r2 = r2_score(y_test, y_pred)
print("R2 on test data: {:.3f}".format(r2))
Auf neuen Datensatz anwenden
X_final = pd.read_csv("Daten_Aufgabe2/X_final.csv")
```



By moeregrind8645

Not published yet.

Last updated 10th August, 2023.

Page 5 of 13.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>

Lineare und logistische Regression (cont)

```
> y_final = linreg.predict(X_final)
print(y_final)
from sklearn.neural_network import MLPClassifier
# Neuronales Netzwerk
mlp = MLPClassifier(hidden_layer_sizes=(100,), max_iter=1000, random_state=42)
mlp.fit(X_train, y_train)
y_pred_mlp = mlp.predict(X_test)
accuracy_mlp = accuracy_score(y_test, y_pred_mlp)
```

Entscheidungsbäume

```
IMPORTS
from sklearn.metrics import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.tree import plot_tree
# 1. Daten in Testdaten und Trainingsdaten aufteilen
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# 2. Entscheidungsbäum erstellen und trainieren
tree = DecisionTreeClassifier(random_state=42, max_depth=4)
tree.fit(X_train, y_train)
# 3. Vorher sagen des Baumes überprüfen
y_pred_tree = tree.predict(X_test)
accuracy_tree = accuracy_score(y_test, y_pred_tree)
print("Accuracy of classification tree on test data: {:.3f}".format(accuracy_tree))
# Entscheidungsbäum plotten
```



By **moeregrind8645**

Not published yet.

Last updated 10th August, 2023.

Page 6 of 13.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

Entscheidungsbäume (cont)

```
> plot_tree(tree, feature_names=X.columns)
plt.savefig('decision_tree.pdf')
```

pandas

```
IMPORTS
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from evaluate import *
# Lade Daten in ein pandas Dataframe
df = pd.read_csv('peptide_classification.csv')
# Zeige die ersten fünf Zeilen des Dataframes
df.head()
#Bereich von Spalten auswählen und anzeigen
df[5:10]
#Spalte auswählen (Namen eingeben)
df['AtomicWeight']
df[['Symbol', 'MeltingPoint', 'BoilingPoint']]
# Spalten ausschließen:
df.drop(['Name', 'AtomicNumber', 'AtomicRadius'], axis=1)
```



By [moeregrind8645](#)

Not published yet.

Last updated 10th August, 2023.

Page 7 of 13.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>

Mode und Median

```
def mode(data, resolution):
    # Berechnen der PMF
    x, px = PMF(data, resolution)
    # Der Wahrscheinlichste Wert ist derjenige, bei dem die PMF ein Maximum hat.
    return x[np.argmax(px)]

-----

def median (data):
    # Wir sortieren die Daten, der Median entspricht dem Datenpunkt in der Mitte.
    data_sorted = np.sort(data)
    if len(data) % 2 != 0:
        median = data_sorted[ len (data)//2]
    # Bei ungerader Anzahl Datenpunkten mitteln wir über die beiden benachbarten Punkte.
    else:
        median = (data_sorted [len (data )//2] + data_sorted[ len (da ta)//2 - 1]) / 2
    return median
```

Probability Mass Function

```
def PMF(data, resolution):
    # Definieren der Bins und Berechnen des Histogramms
    # Wir addieren resolution / 1000 zur Obergrenze, um sicher zu stellen, dass der letzte Wert auch
    im Array ist.
    bin_centers = np.arange (np.min (data), np.max (data) + resolution / 1000, resolution)
    bin_edges = np.linspace( bin_centers[0] - resolution / 2,
                             bin_centers[-1] + resolution / 2,
                             len (bin_centers) + 1)

    hist, _ = np.histogram (data, bin_edges)
```



By moeregrind8645

Not published yet.

Last updated 10th August, 2023.

Page 8 of 13.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

Probability Mass Function (cont)

```
> # Normieren
px = hist / np.sum(hist)

return bin_centers, px
```

Bandpass Filter

```
def ffilter(t, A, fmin, fmax):
    N = len(t)
    dt = t[1] - t[0]
    f = np.fft.fftfreq(N, dt)

    spectrum = np.fft.fft(A)

    for i, freq in enumerate(f):
        if abs(freq) < fmin or abs(freq) > fmax:
            spectrum[i] = 0

    return np.fft.ifft(spectrum)
```

Binomialverteilung

```
def binomial(n, k, p):
    return scipy.special.comb(n, k) * (1-p)**k * p**(n-k)
```

Tiefpassfilter durch Mitteln

```
def LP_filter(x_sample, N_num):
    # N_num ist die Anzahl Werte vor und nach jedem originalen Wert, die gemittelt werden
    N_av = 2*N_num+1 # Filter länge
    n_x = len(x_sample) # Anzahl des Messwerte in x
    x_filtered = x_sample.copy() # wir erstellen eine Kopie von x, um die neuen Werte hinein zu schreiben
    for n in range(N_num, n_x - N_num): # wir ignorieren wir N_num ersten und letzte Werte
```



By **moeregrind8645**

Not published yet.

Last updated 10th August, 2023.

Page 9 of 13.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

Tiefpassfilter durch Mitteln (cont)

```
> x_filtered[n] = sum(x_sample[n-N_num:n+N_num+1])/N_av # Mittelwert
return x_filtered
```

Spektrale Leistungsdichte

```
from scipy import signal
#f_samling = 1/sampling time
f,PSDx = scipy.signal.periodogram(U_mes s,f _sa mpl ing ,re tur n_o nes ide d=T rue ,sc ali -
ng= 'de nsity')
altern ative Berechnung
from numpy.fft import fftfreq, fft, ifft
data = np.loadtxt(' aut oco rre lat ion _la ser.txt')
t = data[:, 0] * 1e-12 # in ps abgesp eic hert. Wir konver tieren in s
signal = data[:, 1]
# Zeitau flösung und Länge des Signals berechnen
dt = t[1] - t[0]
N = len(t)
# Fourie rtr ans for mation
frequency = fftfreq(N, d=dt)
spectrum = fft(si gnal)
# Spektrale Leistu ngs dichte
psd = dt / N * np.abs (sp ectrum)**2
Fourier Rücktr ans for mation:
singal = np.fft.if ft( spe ctrum).
np.rea l(s ignal)
```



By [moeregrind8645](#)

Not published yet.

Last updated 10th August, 2023.

Page 10 of 13.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>

Fourieranalyse

Sei t das Zeit Array und V das Signal Array	$N = \text{len}(t)$
<code>dt = t[1] - t[0]</code>	
<code>f_ny = 1 / (2 * (t[1] - t[0]))</code>	Nyquist-Frequenz
<code>df = 1 / t[-1]</code>	Frequenzauflösung
<code>f = np.fft.fftfreq(N, dt)</code>	Berechnen der geordneten Frequenzachse: welche Frequenzen können vorkommen
<code>spectrum = np.fft.fft(V)</code>	Fouriertransformation (Amplituden der entsprechenden Frequenzen)
<code>psd = dt / N * np.abs(spectrum)**2</code>	Spektrale Leistungsdichte

Fouriertransformation

Koeffizienten für eine Frequenz bestimmen

```
def CN(x, f):
    Xn = 0
    for n in range(N):
        Xn += x[n] * np.exp(-1j * 2 * np.pi * f * n)
    return Xn

fc = 0.022 #f_sig1
print(CN(x, fc))
#print(CN(x, -fc))
```



By [moeregrind8645](#)

cheatography.com/moeregrind8645/

Not published yet.

Last updated 10th August, 2023.

Page 11 of 13.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>

Kovarianz und Autokovarianz

```
C = np.stack((U1_mess, U3_mess), axis=0)
np.cov(C) #Kovarianzmatrix
np.corrcoef(C) #Koeffizienten
AUTOKOVARIANZFUNCTION
def Rxx(x, delta):
    xm = np.mean(x)
    dev_sum = 0
    for i in range(len(x) - delta):
        dev_sum += (x[i] - xm) * (x[i + delta] - xm)
    return dev_sum / (len(x) - delta)
```

numpy Befehle

<code>np.min (array)</code>	Minimum
<code>np.mean(x)</code>	Mittelwert von Array x
<code>np.std(x)</code>	Standardabweichung von Array x
<code>np.abs(x)</code>	Betrag
<code>np.sort(array)</code>	gibt sortiertes array zurück
<code>np.diff(array)</code>	gibt differenz-array zurück

Histogramm

```
V = Spannungs werte
#Wähle Bingröße und erstelle einzelne bins
binsize = 0.01
bins = np.arange(np.min(V), np.max(V)+1, binsize)
hist, b = np.histogram(V, bins) #hist: Häufigkeiten, b: array mit bin_edges
```



By [moeregrind8645](#)

Not published yet.

Last updated 10th August, 2023.

Page 12 of 13.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>

Histogramm (cont)

```
> #plotte normal balken (breite kleiner als binsize)
fig = plt.figure(figsize=(30cm, 25.5cm))
ax1 = fig.add_subplot(2,2,1)
ax1.bar(bins[:-1], hist, width=0.8*binsize)
ax1.set_title('Binsize: {:.2f}V'.format(binsize))
```

Messe diskrete Werte für Spannung, Histogramm zeigt, wie oft welche Spannung gemessen wurde.

Auflösung

```
def extract_resolution(measured_values):
    ''' Extract smallest difference between two unique values in an array.
    Values are treated as unique if their absolute difference is greater than the tolerance.'''

    sorted_values = np.sort(measured_values)
    differences = np.diff(sorted_values)
    sorted_differences = np.sort(np.abs(differences))
    resolution = -1

    tolerance = np.max(np.abs(measured_values)) * 1e-10 # Gleitkommavergleich
    for diff in sorted_differences:
        if diff > tolerance:
            resolution = diff
            break

    return resolution
```



By **moeregrind8645**

Not published yet.

Last updated 10th August, 2023.

Page 13 of 13.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>