Steffie, Jonah

25	26	27	
6/7	6,5/7	3,5/6	16/20

notebook_09

December 19, 2018

1 Aufgabe 25

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        from numpy.polynomial.polynomial import polyval
        import uncertainties.unumpy as unp
        from uncertainties.unumpy import nominal_values as noms
        from uncertainties.unumpy import std_devs as stds
        import pandas as pd
```

a) Bestimme die Parameter mit der Methode der kleinsten Quadrate:

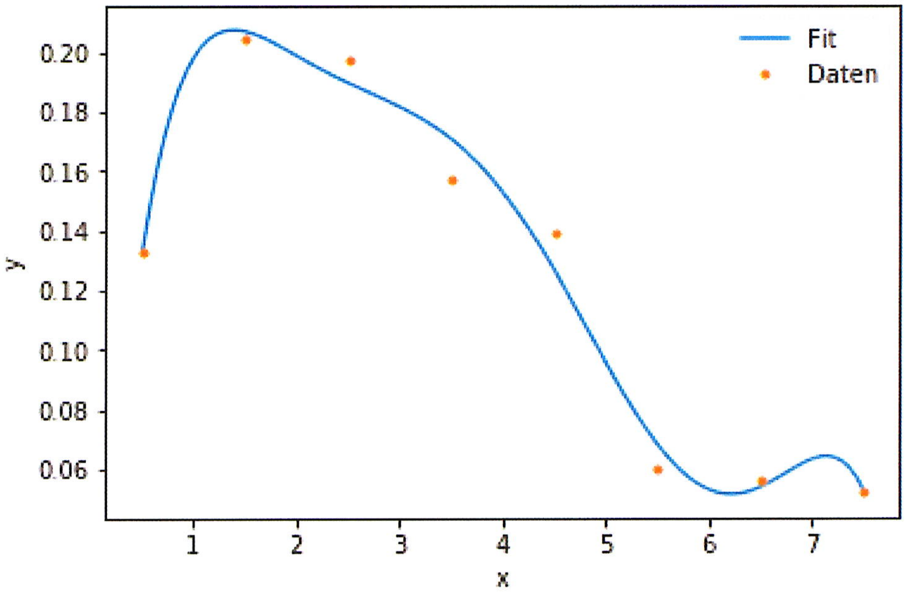
```
In [2]: #read data
        x, y = np.genfromtxt('aufg_a.csv', delimiter = ',', unpack = True)

        #design matrix
        A = np.array([x**i for i in range(7)]).T

        #parameters with least square
        best_a = np.linalg.inv(A.T @ A) @ A.T @ y

        for i in range(7):
            print(f'a_{i} ~ {best_a[i]:.4f}')
```

```
a_0 ~ -0.0674
a_1 ~ 0.6096
a_2 ~ -0.5137
a_3 ~ 0.2106
a_4 ~ -0.0452
a_5 ~ 0.0048
a_6 ~ -0.0002
```

Stelle das Ergebnis graphisch dar:

```
In [3]: xplot = np.linspace(x[0], x[-1], 100)
        plt.plot(xplot, polyval(xplot, best_a), label = 'Fit')
        plt.plot(x, y, '.', label = 'Daten')
```

```
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```



2P.

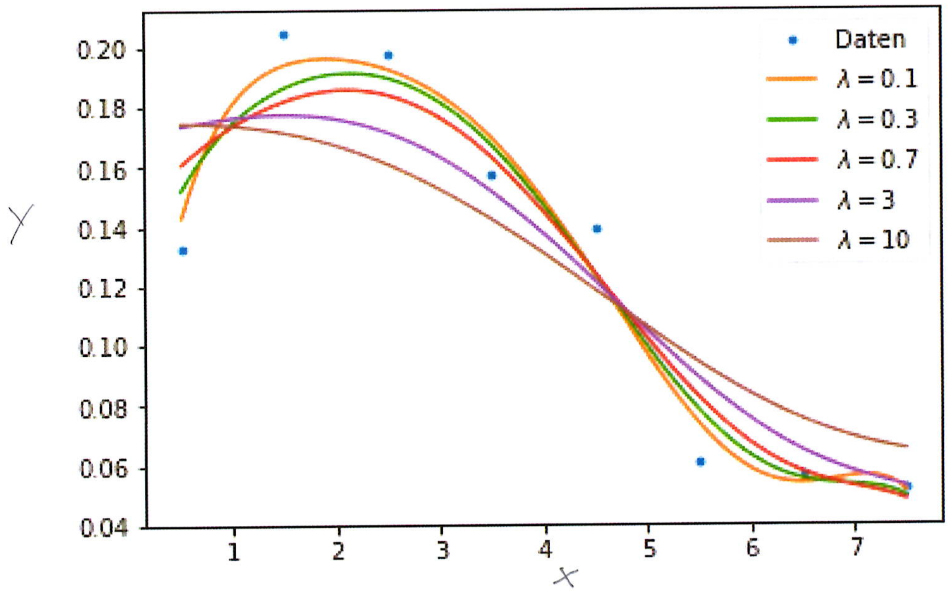
b) Erstelle zunächst die Matrix C, mit der die numerische zweite Ableitung bestimmt wird:

```
In [4]: C = np.zeros((np.shape(A)[0], np.shape(A)[0]))
        np.fill_diagonal(C, -2)
        np.fill_diagonal(C[1:], 1)
        np.fill_diagonal(C[:, 1:], 1)
        C[0, 0] = -1
        C[-1, -1] = -1
        C

Out[4]: array([[-1.,  1.,  0.,  0.,  0.,  0.,  0.,  0.],
               [ 1., -2.,  1.,  0.,  0.,  0.,  0.,  0.],
               [ 0.,  1., -2.,  1.,  0.,  0.,  0.,  0.],
               [ 0.,  0.,  1., -2.,  1.,  0.,  0.,  0.],
               [ 0.,  0.,  0.,  1., -2.,  1.,  0.,  0.],
               [ 0.,  0.,  0.,  0.,  1., -2.,  1.,  0.],
               [ 0.,  0.,  0.,  0.,  0.,  1., -2.,  1.],
               [ 0.,  0.,  0.,  0.,  0.,  0.,  1., -1.]])
```

ich kenne leider keine

Gibt es dafür eine fertige Methode?
Stelle die Ergebnisse der Regularisierung für verschiedene λ dar:

```
In [5]: plt.plot(x, y, '.', label = 'Daten')

        for lam in [0.1, 0.3, 0.7, 3, 10]:
            gamma = np.sqrt(lam) * C @ A
            best_a_reg = np.linalg.inv(A.T @ A + gamma.T @ gamma) @ A.T @ y
            plt.plot(xplot, polyval(xplot, best_a_reg),
                     label = f'$\lambda = {lam}$')

        plt.legend()
        plt.show()
```



✓

Koeffizienten nicht angegeben 2P.

c)

```
In [6]: #read data
        data = pd.read_csv('aufg_c.csv')
        x = data['x']

        #calculate mean and error for y
        y = unp.uarray(data.drop(columns = 'x').T.mean(),
                       data.drop(columns = 'x').T.std())

        #weight matrix
        W = np.zeros((np.shape(A)[0], np.shape(A)[0]))
        np.fill_diagonal(W, 1 / stds(y)**2)
```

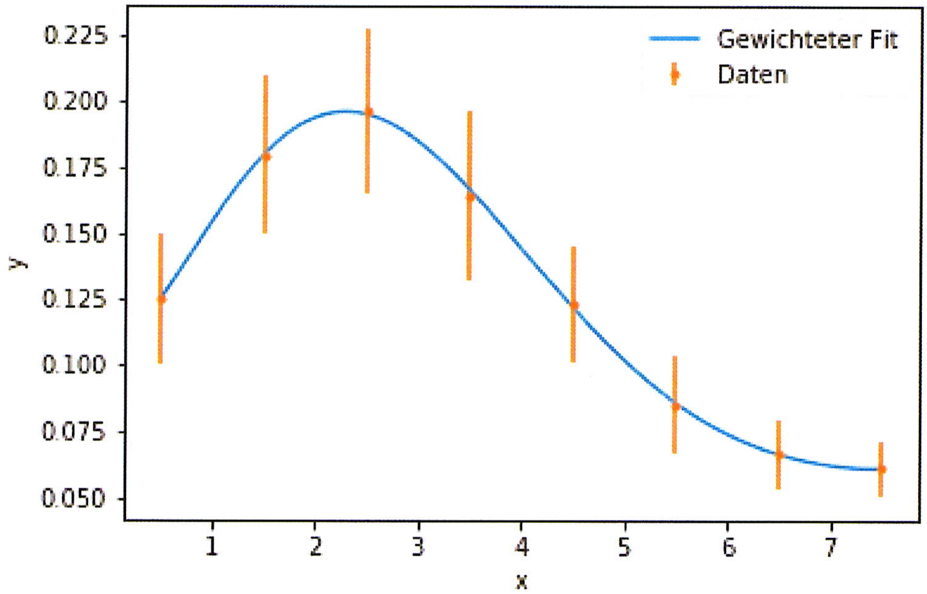
zeros_like(A)

← ist auf N-1 normalisiert was hier auch richtig ist denkt dran, dass das bei np.std() anders ist

```
In [7]: #calculate parameters
        best_a_weight = np.linalg.inv(A.T @ W @ A) @ A.T @ W @ noms(y)
```

Stelle Ergebnisse in einem Plot dar:

```
In [8]: plt.plot(xplot, polyval(xplot, best_a_weight), label = 'Gewichteter Fit')
        plt.errorbar(x = x, y = noms(y),
                     yerr = stds(y),
                     label = 'Daten', marker = '.', linestyle = '')
        plt.xlabel('x')
        plt.ylabel('y')
        plt.legend()
        plt.show()
```



sieht ok aus

✓

2P.

SMD A26

a) $\bar{x} = \frac{1}{n} \sum_{i=1}^{N} x_i$

$E(\bar{x}) = E\left(\frac{1}{n} \sum_{i=1}^{n} x_i\right) = \frac{1}{n} \sum_{i=1}^{n} E(x_i) = \frac{1}{n} \cdot n\mu = \mu$ ✓

$\rightarrow$ Erwartungsgetreu Schätzfunktion für μ 1P.

b) $V(\bar{x}) = V\left(\frac{1}{n} \sum_{i=1}^{n} x_i\right) = \frac{1}{n^2} V\left(\sum_{i=1}^{n} x_i\right) = \frac{1}{n^2} \sum_{i=1}^{n} V(x_i)$

$= \frac{1}{n^2} \cdot n\sigma^2 = \frac{\sigma^2}{n}$ $\square$ ✓ 1P.

c) $S_0^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2$

$E(S_0^2) = \frac{1}{n} E\left(\sum_{i=1}^{n} (x_i^2 - 2x_i\mu + \mu^2)\right)$

$= \frac{1}{n} E\left(\sum_{i=1}^{n} x_i^2 - 2\mu \sum_{i=1}^{n} x_i + n\mu^2\right)$

$= \frac{1}{n}\left(\sum_{i=1}^{n} E(x_i^2) - 2\mu E\left(\sum_{i=1}^{n} x_i\right) + n\mu^2\right)$

$= \frac{1}{n}\left(n(\sigma^2 + \mu^2) - 2n\mu^2 + n\mu^2\right)$ 2P.

$= \sigma^2$ ✓

$\rightarrow$ Erwartungsgetreu Schätzer für σ^2

d) $S_1'^2 = \frac{1}{n} E\left(\sum_{i=1}^{n} (x_i - \bar{x})^2\right) = \frac{1}{n} E\left(\sum_{i=1}^{n} [x_i^2 - 2x_i\bar{x} + \bar{x}^2]\right)$

$= \frac{1}{n} E\left(\sum_{i=1}^{n} x_i^2 - 2\bar{x} \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} \bar{x}^2\right)$

$= \frac{1}{n} E\left(\sum_{i=1}^{n} x_i^2 - 2n\bar{x}^2 + n\bar{x}^2\right) = \frac{1}{n} E\left(\sum_{i=1}^{n} x_i^2 - n\bar{x}^2\right)$

$= \frac{1}{n} \sum_{i=1}^{n} E(x_i^2) - nE(\bar{x}^2)$

$= \frac{1}{n}(\sigma^2 + \mu^2) - n\left(\frac{\sigma^2}{n} + \mu^2\right)$

bei mir kommt

$= \frac{1-n}{n} \sigma^2$ (✓) $\Rightarrow$ nicht Erwartungsgetreu ✓

$\frac{n-1}{n} \sigma^2$ raus

$\rightarrow$ mit Korrektur $S_1^2 = \frac{1}{1-n} \sum_{i=1}^{n} (x_i - \bar{x})^2$ (✓)

da $E\left(\frac{n}{1-n} S_1'^2\right) = \sigma^2$ (✓)

2,5P.

Nitschke
Grisard

Aufgabe 27:

a) Likelihood allgemein:

$$\mathcal{L}(\vec{a}) = f(\vec{x}_1|\vec{a}) \cdots f(\vec{x}_n,\vec{a}) = \prod_{i=1}^{n} f(\vec{x}_i|\vec{a})$$

hier 1D:

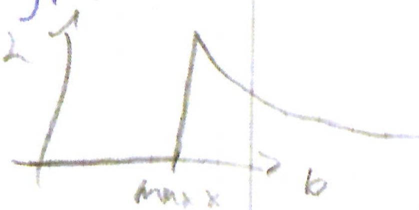
$$\mathcal{L}(b) = \prod_{i=1}^{n} f(x_i|b)$$

und $f(x) = \begin{cases} 1/b & , x \in [0,b] \\ 0 & , \text{sonst} \end{cases}$

für ein gegebenes Sample ist die Likelihood also 0, wenn b so gewählt wird, dass min. 1 x_i nicht in $[0,b]$ liegt.

Wenn $b \geq x_i \;\; \forall i \in \{1,\dots,n\}$, dann $f(x_i|b) = \frac{1}{b}$ und

$$\mathcal{L}(b) = \frac{1}{b^n}$$ ✓



$1/b^n$ fällt monoton mit b, daher ist $\mathcal{L}(\max_{x\in\{x_i\}} x)$ maximal und der beste Schätzer ist $b = \max_{x\in\{x_i\}}(x)$ ✓ 3P.

b)

$$E\left[\max_{x\in\{x_i\}} x\right] = \int_0^{\max_{x\in\{x_i\}}(x)} \frac{1}{\max_{x\in\{x_i\}}(x)}\, x\, dx = \frac{\max_{x\in\{x_i\}}(x)}{2}$$

Für Erwartungstreue muss, Erwartungswert vom Schätzer = Schätzer' sein, wähle daher 0,5P.

$$b^* = 2 \cdot \max_{x\in\{x_i\}}(x)$$ f

$$b^* = \frac{n+1}{n} \max_{x\in\{x_i\}}(x)$$