blatt14_guth_venker_jaekel

November 7, 2021

[1]:
```python
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

1 Aufgabe 31

1.1 a)

[2]:
```python
df=pd.read_csv('aufg_a.csv')
df
```

[2]:

	x	"y_0"
0	0.5	0.132939
1	1.5	0.204351
2	2.5	0.197394
3	3.5	0.157457
4	4.5	0.139232
5	5.5	0.060324
6	6.5	0.056360
7	7.5	0.051944

[3]:
```python
df.columns
```

[3]:
```
Index(['x', ' "y_0"'], dtype='object')
```

[4]:
```python
x=df.x.to_numpy()
y=df[' "y_0"'].to_numpy()
x,y
```

[4]:
```
(array([0.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5]),
 array([0.132939, 0.204351, 0.197394, 0.157457, 0.139232, 0.060324,
        0.05636 , 0.051944]))
```

[5]:
```python
A= np.column_stack([x**0,x,x**2,x**3,x**4,x**5,x**6])
print('Designmatrix A:')
print(f'{A.shape = }')
print(f'{A = }')
```

```
Designmatrix A:
A.shape = (8, 7)
A = array([[1.00000000e+00, 5.00000000e-01, 2.50000000e-01, 1.25000000e-01,
        6.25000000e-02, 3.12500000e-02, 1.56250000e-02],
       [1.00000000e+00, 1.50000000e+00, 2.25000000e+00, 3.37500000e+00,
        5.06250000e+00, 7.59375000e+00, 1.13906250e+01],
       [1.00000000e+00, 2.50000000e+00, 6.25000000e+00, 1.56250000e+01,
        3.90625000e+01, 9.76562500e+01, 2.44140625e+02],
       [1.00000000e+00, 3.50000000e+00, 1.22500000e+01, 4.28750000e+01,
        1.50062500e+02, 5.25218750e+02, 1.83826562e+03],
       [1.00000000e+00, 4.50000000e+00, 2.02500000e+01, 9.11250000e+01,
        4.10062500e+02, 1.84528125e+03, 8.30376562e+03],
       [1.00000000e+00, 5.50000000e+00, 3.02500000e+01, 1.66375000e+02,
        9.15062500e+02, 5.03284375e+03, 2.76806406e+04],
       [1.00000000e+00, 6.50000000e+00, 4.22500000e+01, 2.74625000e+02,
        1.78506250e+03, 1.16029062e+04, 7.54188906e+04],
       [1.00000000e+00, 7.50000000e+00, 5.62500000e+01, 4.21875000e+02,
        3.16406250e+03, 2.37304688e+04, 1.77978516e+05]])
```

[6]:

```
a = np.linalg.inv(A.T @ A) @ A.T @ y
print(f'{a.shape = }')
print(f'{a = }')
```

```
a.shape = (7,)
a = array([-6.74453270e-02,  6.09609038e-01, -5.13748213e-01,  2.10566521e-01,
       -4.52007751e-02,  4.78568049e-03, -1.96288196e-04])
```

[7]:

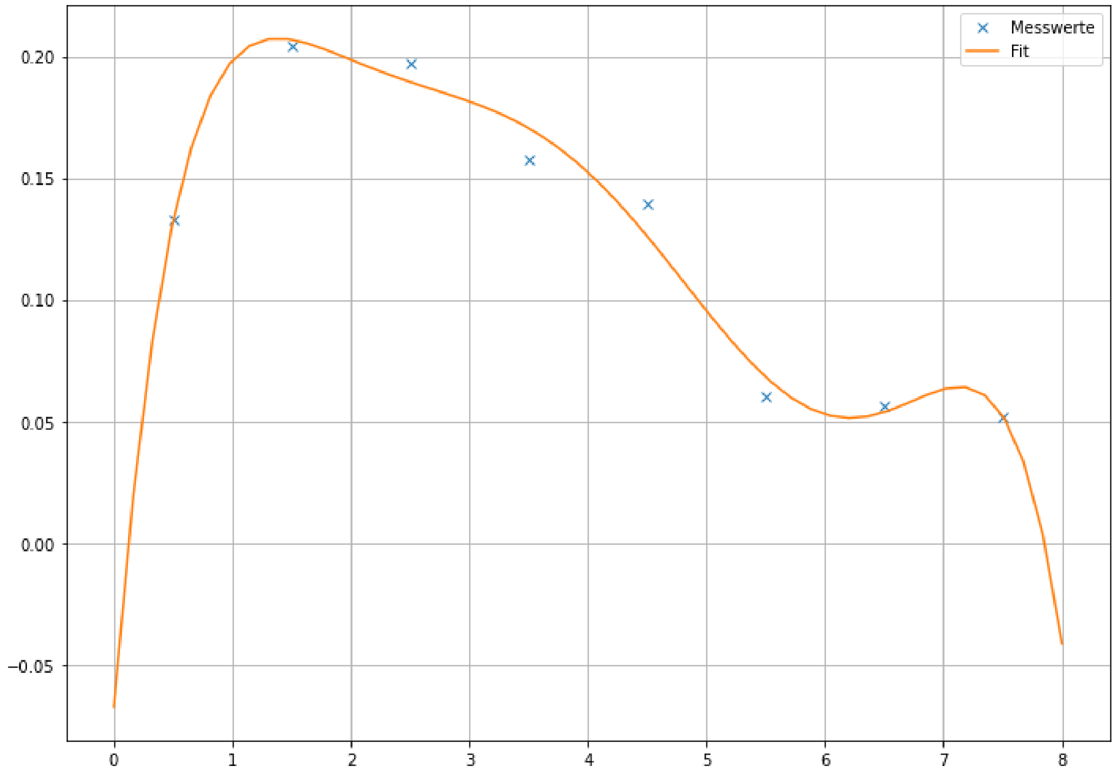
```
def polynom_6(x,a):
    return a[0]*x**0 + a[1]*x + a[2]*x**2 + a[3]*x**3 + a[4]*x**4 + a[5]*x**5 +␣
 ↪a[6]*x**6
```

[8]:

```
val= np.linspace(0,8)
```

[9]:

```
plt.figure(figsize=(10,7))
plt.plot(x,y,'x',label="Messwerte")
plt.plot(val,polynom_6(val,a),label="Fit")
plt.grid()
plt.legend(loc='best')
plt.tight_layout()
```



1.2 b)

[10]:
```
C = ([[-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]])
C = np.array(C)
lam = ([0.1,0.3,0.7,3,10])
print('Ableitungsmatrix C:')
print(f'{C.shape = }')
print(f'{C = }')
```

```
Ableitungsmatrix C:
C.shape = (8, 8)
C = 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],
```

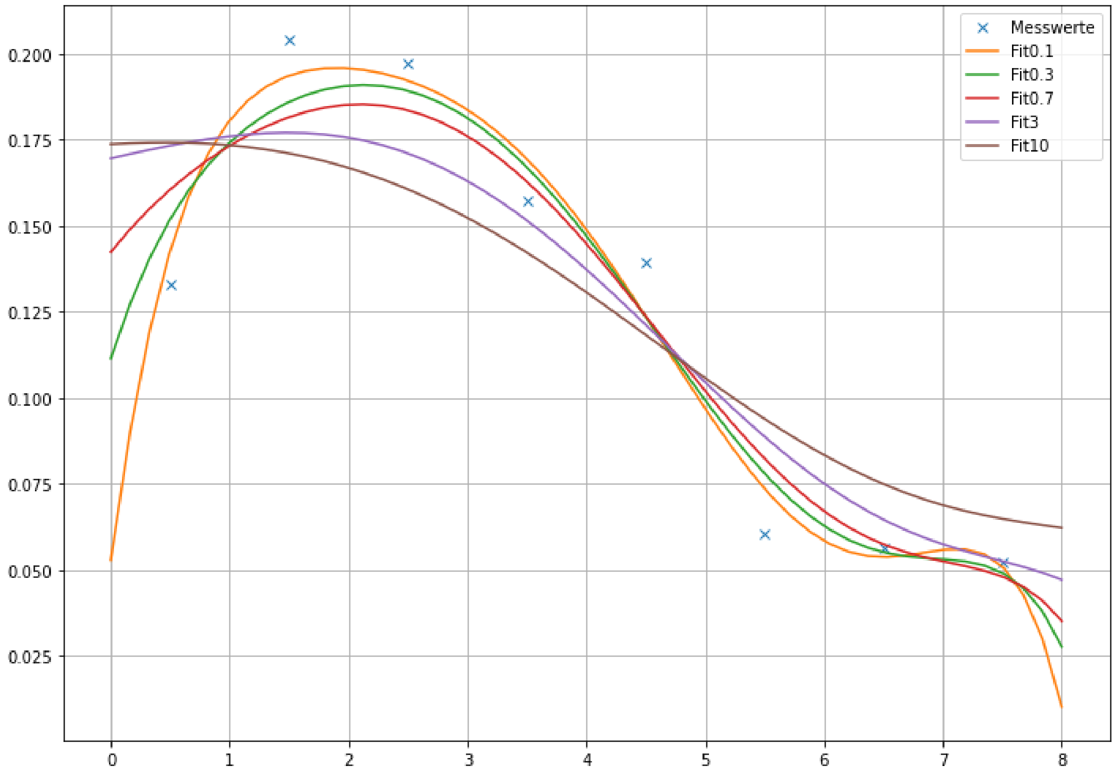
```
       [ 0,  0,  0,  0,  1, -2,  1,  0],
       [ 0,  0,  0,  0,  0,  1, -2,  1],
       [ 0,  0,  0,  0,  0,  0, -1,  1]])
```

[11]:
```
for i in range(len(lam)):
    a_reg = np.linalg.inv(A.T @ A + lam[i] * (C@A).T @ (C @ A)) @ A.T @ y
    with np.printoptions(precision=2):
        print(f'a_reg(lam={lam[i]}) = \n {a_reg}')
```

```
a_reg(lam=0.1) =
 [ 5.28e-02  2.60e-01 -1.93e-01  7.70e-02 -1.72e-02  1.90e-03 -8.10e-05]
a_reg(lam=0.3) =
 [ 1.11e-01  1.08e-01 -6.43e-02  2.49e-02 -6.34e-03  7.89e-04 -3.63e-05]
a_reg(lam=0.7) =
 [ 1.42e-01  4.37e-02 -1.72e-02  6.46e-03 -2.36e-03  3.59e-04 -1.81e-05]
a_reg(lam=3) =
 [ 1.70e-01  7.97e-03 -1.06e-03 -1.07e-04 -4.92e-04  1.08e-04 -6.00e-06]
a_reg(lam=10) =
 [ 1.74e-01  2.10e-03 -2.10e-03 -1.88e-04 -1.44e-04  3.94e-05 -2.29e-06]
```

[12]:
```
plt.figure(figsize=(10,7))
plt.plot(x,y,'x',label="Messwerte")
for i in range(len(lam)):
    a_reg = np.linalg.inv(A.T @ A + lam[i] * (C@A).T @ (C @ A)) @ A.T @ y
    plt.plot(val,polynom_6(val,a_reg),label="Fit"+ str(lam[i]))

plt.grid()
plt.legend(loc='best')
plt.tight_layout()
```



1.3 c)

[13]:
```
df_2=pd.read_csv('aufg_c.csv')
```

1.3.1 Fehler des Mittelwertes ist: $\frac{\textbf{Standardabweichung}}{\sqrt{\textbf{Anzahl der Werte}}}$

[14]:
```
df_2['mean'] = df_2.iloc[:, 1:51].mean(axis=1)
df_2['err_mean'] = df_2.iloc[:, 1:51].std(axis=1,ddof=1) / np.sqrt(50)
```

[15]:
```
df_2.head()
```

[15]:
```
     x       "y_0       "y_1       "y_2       "y_3       "y_4       "y_5       "y_6  \
0  0.5  0.103939  0.149753  0.112739  0.184439  0.145831  0.113267  0.155455
1  1.5  0.219092  0.193544  0.160651  0.170565  0.184050  0.163337  0.212139
2  2.5  0.177677  0.166459  0.236241  0.205570  0.197066  0.230060  0.170907
3  3.5  0.139580  0.163861  0.167301  0.141728  0.143816  0.161717  0.153893
4  4.5  0.115663  0.112004  0.111168  0.102507  0.108234  0.085693  0.113646

       "y_7       "y_8  …      "y_42      "y_43      "y_44      "y_45      "y_46  \
0  0.155187  0.105173  …  0.122147  0.146955  0.076564  0.114331  0.132773
1  0.166457  0.186264  …  0.147765  0.164665  0.238038  0.203325  0.203236
```

```
2  0.181053  0.186724  …  0.144067  0.239231  0.103315  0.198792  0.201374
3  0.131928  0.110090  …  0.213696  0.157921  0.229623  0.207216  0.125185
4  0.148677  0.199207  …  0.161769  0.111842  0.142766  0.097612  0.132306

      "y_47     "y_48     "y_49      mean  err_mean
0  0.167562  0.135968  0.131620  0.124865  0.003522
1  0.213183  0.223799  0.148379  0.179360  0.004211
2  0.197383  0.126766  0.193543  0.196042  0.004397
3  0.119271  0.212534  0.193191  0.164007  0.004500
4  0.101199  0.085060  0.094930  0.122767  0.003173

[5 rows x 53 columns]
```

[17]:
```python
y_mean = df_2['mean'].to_numpy()
y_err = df_2['err_mean'].to_numpy()
```

1.3.2 Berechne W nach dem Script mit $\frac{1}{\text{Fehler}^2}$auf der Diagonalen

[18]:
```python
W = np.eye(8,8) * (1/y_err**2)
print('Gewichtungsmatrix W:')
print(f'{W.shape = }')
print(f'W =\n{W}')
```

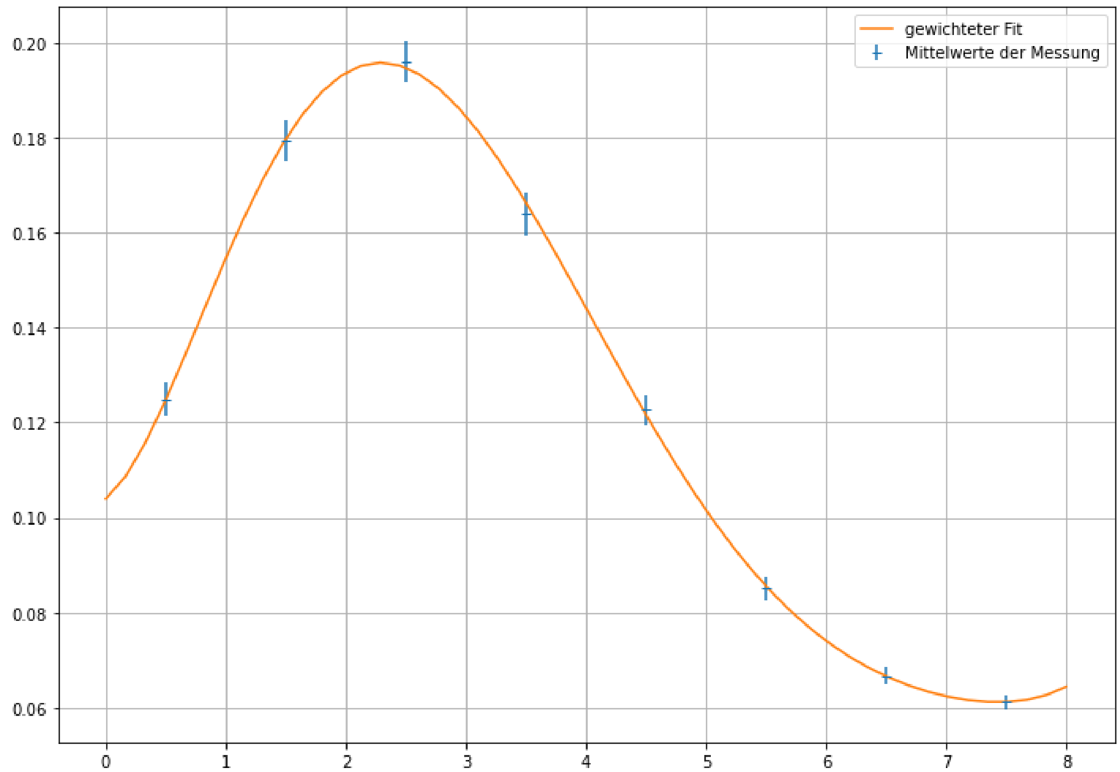
```
Gewichtungsmatrix W:
W.shape = (8, 8)
W =
[[ 80629.34129337      0.               0.               0.
       0.               0.               0.               0.        ]
 [     0.           56403.43595568      0.               0.
       0.               0.               0.               0.        ]
 [     0.               0.           51717.53013034      0.
       0.               0.               0.               0.        ]
 [     0.               0.               0.           49385.61212761
       0.               0.               0.               0.        ]
 [     0.               0.               0.               0.
   99332.59310101      0.               0.               0.        ]
 [     0.               0.               0.               0.
       0.          152282.27863975      0.               0.        ]
 [     0.               0.               0.               0.
       0.               0.          296112.26907967      0.        ]
 [     0.               0.               0.               0.
       0.               0.               0.          472574.91772621]]
```

1.3.3 $\vec{a} = (A^T W A)^-1 A^T W \vec{y}$

[19]:
```
a_w = np.linalg.inv(A.T @ W @ A) @ A.T @ W @ y_mean
print(f'{a_w.shape = }')
print(f'{a_w = }')
```

```
a_w.shape = (7,)
a_w = array([ 1.03975570e-01,  1.92955342e-02,  6.17000739e-02, -3.75650688e-02,
        7.91884317e-03, -7.34894171e-04,  2.56958525e-05])
```

[20]:
```
plt.figure(figsize=(10,7))
#plt.plot(x,y_mean,'x',label="Mittelwerte")
plt.errorbar(x,y_mean, yerr= y_err,fmt="_", label = "Mittelwerte der Messung")
#plt.plot(x,y, "o", label = "Messwerte aus a)")
plt.plot(val,polynom_6(val,a_w),label="gewichteter Fit")
#plt.plot(val,polynom_6(val,a),label="ungewichteter Fit")
plt.grid()
plt.legend(loc='best')
plt.tight_layout()
```



[ ]:

Gamma-Astronomie

Samstag, 6. November 2021 13:07

Aufgabe 32 *γ-Astronomie* **5 P.**

Bei einer typischen Messung in der γ-Astronomie wird das Teleskop auf eine Position (*on*-Position) gerichtet, an der eine γ-Strahlungs-Quelle vermutet wird. In der anschließenden Messung werden N_{on} Ereignisse über einen Zeitraum t_{on} aufgezeichnet. In den gemessenen Ereignissen N_{on} befinden sich sowohl Untergrund- als auch Signalphotonen. Um zu ermitteln, wie viel Untergrund vorhanden ist, wird ebenfalls an einer anderen Position ohne Quelle (*off*-Position) gemessen. Bei dieser Messung werden N_{off} Photonen-Ereignisse in einer Zeit t_{off} gemessen.

Um zu entscheiden, ob sich an der *on*-Position eine Quelle befindet, soll mit einem Likelihood-Quotienten-Test getestet werden, ob ein signifikanter Überschuss an Photonen über der Untergrunderwartung für die *on*-Position gemessen wurde (hier noch nicht, erst in Kapitel *Testen*).

Ziel dieser Aufgabe ist es, vorbereitend die richtige Likelihood-Funktion für den späteren Likelihood-Quotienten-Test aufzustellen.

Nutzen Sie für die Bearbeitung der Aufgabe die Ausdrücke:

- $\alpha = \frac{t_{\text{on}}}{t_{\text{off}}}$: Quotient der unterschiedlichen Messzeiten
- $b = \langle N_{\text{off}} \rangle$: Unbekannter Erwartungswert für die Zahl der Untergrundphotonen während der Messzeit t_{off}
- s: Unbekannter Erwartungswert für die Zahl der Signal Photonen während der Messzeit t_{on} aus der γ-Strahlungs-Quelle. Nicht zu verwechseln mit der gesamten Erwartung für die *on*-Position.

(a) Wie groß ist der Erwartungswert $\langle N_{\text{on}} \rangle$, ausgedrückt durch s, b und α?

(b) Welchen Wahrscheinlichkeitsverteilungen folgen N_{on} und N_{off}?
Tipp: Die gezählten Ereignisse kommen unabhängig voneinander im Detektor an.

(c) Wie sieht die Likelihoodfunktion $\mathcal{L}(b, s)$ für die Parameter b und s aus?

(d) Welche Werte $\hat{b}$ und $\hat{s}$ maximieren $\mathcal{L}$?
Tipp: Nutzen Sie die negative Likelihood-Funktion, dann wird die Rechnung einfacher.

(e) Berechnen Sie die Kovarianzmatrix von $\hat{b}$ und $\hat{s}$. Wie hängt die Kovarianzmatrix mit der Likelihood zusammen? Ist diese Art der Fehlerberechnung exakt?

(a) $\alpha = \frac{t_{on}}{t_{off}}$, $b = \langle N_{off} \rangle$

b: Erwartungswert der off Ereignisse (Untergrund)

s: Erwartungswert der Signal Ereignisse

$$\frac{\langle N_{on} \rangle - s}{t_{on}} = \frac{b}{t_{off}}$$

$$\Leftrightarrow \langle N_{on} \rangle = b \cdot \alpha + s$$

b) Welche W. Verteilungen haben N_{on} , N_{off} ?

N_{on} ist Poissonverteilt: $P_\lambda(k) = \frac{\lambda^k}{k!} \cdot \exp(-\lambda)$

$\lambda = np$, n Versuche

$= \langle N_{on} \rangle$ p Treffwahrscheinlichkeit

N_{off} ist Poissonverteilt: $P_\lambda(k) = \frac{\lambda^k}{k!} \exp(-\lambda)$

mit $\lambda = \langle N_{off} \rangle$

$$P_{on}(N_{on}) = \frac{(b\alpha + s)^{N_{on}}}{N_{on}!} \exp(-(b\alpha + s))$$

$$P_{off}(N_{off}) = \frac{b^{N_{off}}}{N_{off}!} \exp(-b)$$

c) Likelihood-Funktion $\mathcal{L}(b, s)$

$$\mathcal{L}(b,s) = P_{on}(N_{on})\; P_{off}(N_{off})$$

$$= \frac{(b\alpha+s)^{N_{on}}}{N_{on}!} \cdot \exp(-(b\alpha+s)) \cdot \frac{b^{N_{off}}}{N_{off}!} \exp(-b)$$

$$= \frac{(b\alpha+s)^{N_{on}}}{N_{on}!} \frac{b^{N_{off}}}{N_{off}!} \exp\left(-b\cdot(\alpha+1)-s\right)$$

d) $$-\ln(\mathcal{L}(b,s)) = -\ln\left(\frac{(b\alpha+s)^{N_{on}}}{N_{on}!}\frac{b^{N_{off}}}{N_{off}!}\right) + b(\alpha+1) + s = F(b,s)$$

$$= -\ln\left((b\alpha+s)^{N_{on}}\right) + \ln(N_{on}!) - \ln(b^{N_{off}}) + \ln(N_{off}!) + b(\alpha+1) + s$$

$$= -N_{on}\cdot\ln(b\alpha+s) + \ln(N_{on}!) - N_{off}\ln(b) + \ln(N_{off}!) + b(\alpha+1) + s$$

$$\frac{\partial F}{\partial b} = -N_{on}\cdot\frac{1}{b\alpha+s}\cdot\alpha - N_{off}\cdot\frac{1}{b} + \alpha + 1$$

$$\frac{\partial F}{\partial s} = -N_{on}\frac{1}{b\alpha+s} + 1$$

$$\frac{\partial F}{\partial s} = \frac{\partial F}{\partial b} \overset{!}{=} 0$$

$$\frac{-N_{on}\alpha}{b\alpha+s} - \frac{N_{off}}{b} + \alpha + 1 = 0 \qquad \mathrm{I}$$

$$\frac{-N_{on}}{b\alpha+s} + 1 = 0 \;\; \mathrm{II} \iff N_{on} = b\alpha + s$$

II in I: $$\cancel{-\alpha} - \frac{N_{off}}{b} \cancel{+\alpha} + 1 = 0$$

$$-\frac{N_{off}}{b} + 1 = 0 \Rightarrow \boxed{N_{off} = \hat{b} = b}$$

in II: $$N_{on} = N_{off}\,\alpha + s$$

$$\iff \boxed{s = N_{on} - N_{off}\,\alpha \,\hat{=}\, \hat{s}}$$

e) $$\frac{\partial F}{\partial b} = -N_{on}\cdot\frac{1}{b\alpha+s}\cdot\alpha - N_{off}\cdot\frac{1}{b} + \alpha + 1$$

$$\frac{\partial F}{\partial s} = -N_{on}\frac{1}{b\alpha+s} + 1$$

$\mathrm{cov}(b,s)$?

$$\frac{\partial^2 F}{\partial b^2} = N_{on}\,\alpha^2\cdot(b\alpha+s)^{-2} + N_{off}\,(b)^{-2}$$

$$\frac{\partial^2 F}{\partial s^2} = N_{on}\,(b\alpha+s)^{-2}$$

$$\frac{\partial^2 F}{\partial s^2} = N_{on}\,(bd+s)^{-2}$$

$$\frac{\partial^2 F}{\partial b \partial s} = N_{on}\, d \cdot (bd+s)^{-2}$$

$$\frac{\partial^2 F}{\partial s \partial b} = N_{on}\,(bd+s)^{-2} \cdot d$$

Ableitungsmatrix: $$G = \begin{pmatrix} N_{on}\, d^2 \cdot (bd+s)^{-2} + N_{off}\,(b)^{-2} & N_{on}\, d \cdot (bd+s)^{-2} \\ N_{on}\, d \cdot (bd+s)^{-2} & N_{on}\,(bd+s)^{-2} \end{pmatrix}$$

$$G^{-1} = Cov(b, s)$$

$$G^{-1} = \begin{pmatrix} N_{on}\,(bd+s)^{-2} & -N_{on}\, d \cdot (bd+s)^{-2} \\ -N_{on}\, d \cdot (bd+s)^{-2} & N_{on}\, d^2 \cdot (bd+s)^{-2} + N_{off}\,(b)^{-2} \end{pmatrix} \cdot \left(\cancel{N_{on}^2\, d^2\, (bd+s)^{-4}} + N_{off} N_{on}\,(b)^{-2} (bd+s)^{-2} - \cancel{N_{on}^2\, d^2\, (bd+s)^{-4}} \right)^{-1}$$

$$= \begin{pmatrix} N_{on}\,\cancel{(bd+s)^{-2}} & -N_{on}\, d \cdot \cancel{(bd+s)^{-2}} \\ -N_{on}\, d \cdot \cancel{(bd+s)^{-2}} & N_{on}\, d^2 \cdot \cancel{(bd+s)^{-2}} + N_{off}\,(b)^{-2} \end{pmatrix} \cdot \frac{b^2 (bd+s)^2}{N_{off} N_{on}}$$

$$= \begin{pmatrix} \frac{b^2}{N_{off}} & -\frac{d \cdot b^2}{N_{off}} \\ -\frac{d b^2}{N_{off}} & \frac{b^2 d^2}{N_{off}} + \frac{(bd+s)^2}{N_{on}} \end{pmatrix}$$

Wie hängt es zusammen?

$$Cov(b,s) = \frac{1}{\frac{\partial^2 F}{\partial b^2}\frac{\partial^2 F}{\partial s^2} - \frac{\partial^2 F}{\partial b \partial s}\frac{\partial^2 F}{\partial s \partial b}} \cdot \begin{pmatrix} \frac{\partial^2 F}{\partial b^2} & \frac{\partial^2 F}{\partial b \partial s} \\ \frac{\partial^2 F}{\partial s \partial b} & \frac{\partial^2 F}{\partial s^2} \end{pmatrix}$$