Jonaaah, Stefan

32	33	34	36	35	
4/5	5/5	4,25/5	4/5	4,75/5	22/25

notebook_11

January 24, 2019

0.1 Aufgabe 32: χ^2-Test

Für den χ^2-Test wird jeweils folgende GröSSe berechnet:

$$\chi^2 = \sum_{i=1}^{N} \frac{(E_i - E_H)^2}{\sigma_i^2}$$

E_H bezeichnet den Energiewert der Hypothese. Die Standardabweichung σ muss aus den Daten geschätzt werden. Hierfür wird $\sigma = 0.5$ angenommen.

das ist hier nicht richtig die Standardabweichung ist gegeben ihr habt die ja nicht aus den Daten errechnet

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
```

```
In [2]: E = np.array([31.6, 31.3, 32.2, 30.8, 31.2, 31.3, 31.9])
```

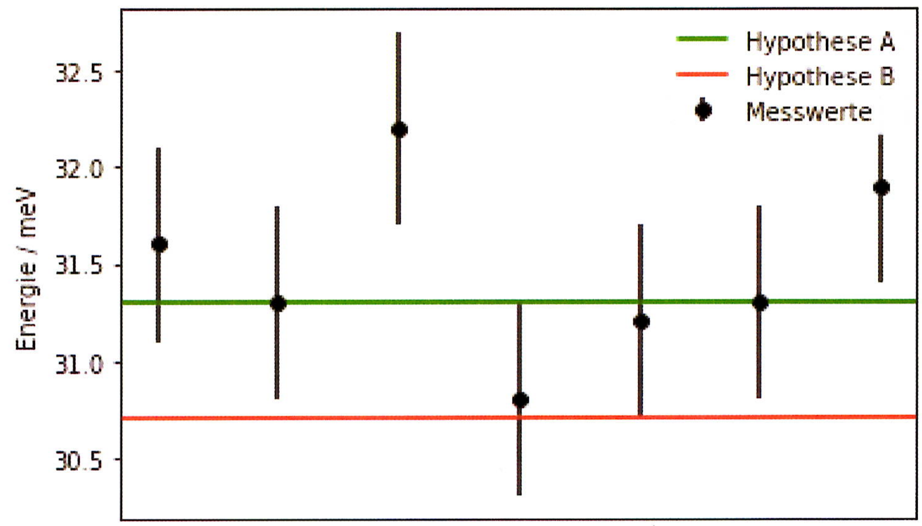
a)

```
In [3]: E_A = 31.3
        sig = 0.5

        chi2_A = np.sum( ( E - E_A )**2 / sig**2 )
        chi2_A

Out[3]: 6.080000000000008
```

b)

```
In [4]: E_B = 30.7
        chi2_B = np.sum( ( E - E_B )**2 / sig**2 )
        chi2_B

Out[4]: 21.920000000000066
```

```
In [5]: plt.axhline(y = E_A, label = 'Hypothese A', color = 'g')
        plt.axhline(y = E_B, label = 'Hypothese B', color = 'r')
        plt.errorbar(x = np.arange(7), y = E,yerr = 0.5,
                     label = 'Messwerte',
                     linestyle = '', marker = 'o', color = 'k')
        plt.ylabel('Energie / meV')
        plt.legend()
        plt.xticks([])
        None
```



hier sind 7 richtig

In χ^2-Tabelle findet man bei $7 - 1 = 6$ Freiheitsgraden für 5% Signifikanz den Wert 12,59. Eine Hypothese wird daher abgelehnt, wenn $\chi^2 > 12{,}59$ ist. (✓) Daher ~~kann~~ muss Hypothese B verworfen werden. (✓) Hypothese A kann auf Grundlage der gefundenen Daten nicht verworfen werden.

die wurde euch einfach gegeben → Es wurde die Varianz der theoretischen Verteilung geschätzt, daher muss die Zahl der Freiheitsgrade um 1 verringert werden. Hier läge der χ^2-Wert bei 11,07. Es ändert sich somit nichts am Ergebnis.

4/5

0.2 Aufgabe 33

a) Wähle $\mu = \lambda$ und $\sigma = \sqrt{\lambda}$ ✓ 0,5P.

b) + c) + d) Da die Bestimmung des Wertes für λ durch einen Zufallsprozess erfolgt, wird der Mittelwert aus 100 Durchläufen verwendet. ✓

```
In [6]: for alpha in [0.05, 0.025, 0.001]:
            print(f'Für die Signifikanz alpha = {alpha}')
            K_alpha = np.sqrt(np.log(2 / alpha) / 2 )

            lambda_accepted_list = []
            for i in range(100):
                for lam in np.arange(1, 51):
                    #sample gauss and poisson distribution
                    gaus_sample = np.round(np.random.normal(lam, np.sqrt(lam),
                                                            size=10000), 0)

                    poisson_sample = np.random.poisson(lam, size = 10000)


                    #binning
```

evtl. step = 0,1

sehr schön

```
gaus_hist, gaus_edges = np.histogram(gaus_sample, bins = 100,
                                     range= (lam - 5 * np.sqrt(lam),
                                             lam + 5 * np.sqrt(lam)))

poisson_hist, poisson_edges = np.histogram(poisson_sample, bins = 100,
                                     range= (lam - 5 * np.sqrt(lam),
                                             lam + 5 * np.sqrt(lam)))


#create the cdfs
gaus_cdf = np.array([np.sum(gaus_hist[:i + 1]) / len(gaus_sample)
                    for i in range(100)])
poisson_cdf = np.array([np.sum(poisson_hist[:i + 1]) / len(poisson_sample)
                    for i in range(100)])


#calculate the max distance between cdfs
d_max = np.max(np.abs(gaus_cdf - poisson_cdf))


#test condition
lambda_accepted = np.sqrt(len(gaus_sample)/2) * d_max < K_alpha
if lambda_accepted:
    lambda_accepted_list.append(lam)
    break
```

2P.

```
    print(f'kann ab lambda = {round(np.mean(lambda_accepted_list), 0)} +/- {round(np.s
print('...die Hypothese, dass beide Verteilungen gleich sind, nicht mehr verworfen wer
```

```
Für die Signifikanz alpha = 0.05
kann ab lambda = 10.0 +/- 3.0...

Für die Signifikanz alpha = 0.025
kann ab lambda = 9.0 +/- 2.0...

Für die Signifikanz alpha = 0.001
kann ab lambda = 6.0 +/- 2.0...

...die Hypothese, dass beide Verteilungen gleich sind, nicht mehr verworfen werden.
```

- Ihr habt alles richtig implementiert, die Werte sind bei euch höher als in der Musterlösung weil ihr λ nur in 1ner Schritten variiert, das könnte man noch feiner machen und beim Ausgeben muss man nicht unbedingt auf ganze Zahlen runden

2,5P.

0.3 Aufgabe 34: Ballon-Experiment

a) Berechnungen/Herleitungen handschriftlich.

```
In [7]: counts = np.array([4135, 4202, 4203, 4218, 4227, 4231, 4310])
        days = np.arange(1, 8)
```

Wahrscheinlichste Zählrate entspricht dem Mittelwert

```
In [8]: c_mean = round(np.mean(counts), 0)
        print(c_mean)
```

```
4218.0
```

Likelihood-Wert fehlt   0,75P.

b) Bestimme Parameter numerisch

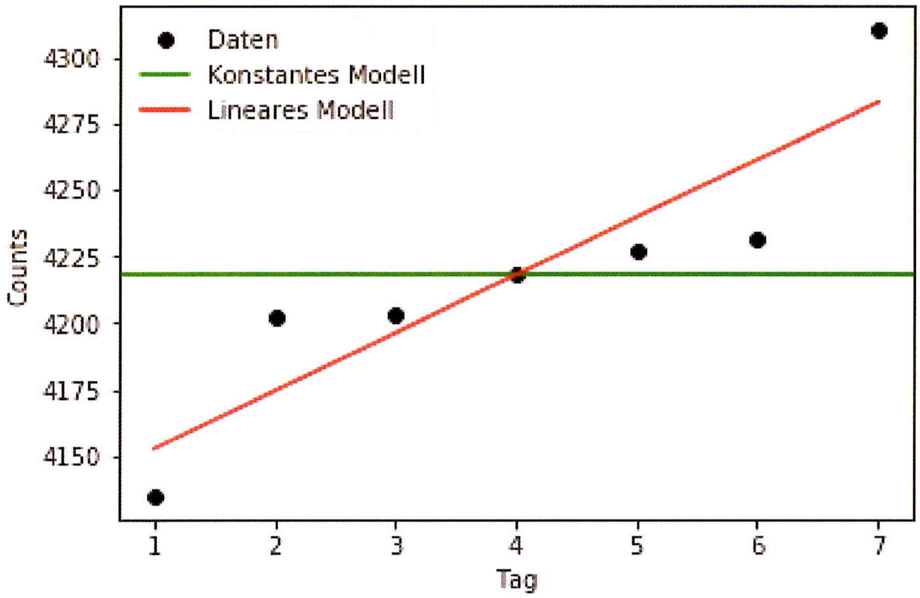
```
In [9]: from scipy.optimize import fmin
```

```
In [10]: def llh(params):
             A = params[0]
             B = params[1]
             return + len(counts) * B + A * np.sum(days) - np.sum(counts * np.log(A * days + B
```

```
In [11]: lin_model = fmin(llh, [0, 4000])
         print(f'Parameter für lineares Modell: {lin_model}')
```

```
Optimization terminated successfully.
         Current function value: -216932.516775
         Iterations: 85
         Function evaluations: 166
Parameter für lineares Modell: [  21.66767923 4131.32923889]
```

```
In [12]: plt.plot(days, counts, 'ko', label = 'Daten')
         plt.axhline(y = np.mean(counts), label = 'Konstantes Modell', color = 'g')
         plt.plot(days, lin_model[0] * days + lin_model[1],
                  color = 'r', label = 'Lineares Modell')
         plt.xlabel('Tag')
         plt.ylabel('Counts')
         plt.legend()
         None
```



gern auch mit Unsicherheiten

1P.

c) Test ist der Quotient aus den einzelnen Likelihoods.

```
In [13]: gamma = np.prod((c_mean / (lin_model[0] * days + lin_model[1]))**counts
                     * np.exp(-c_mean + lin_model[0] * days + lin_model[1]))
```

Aus Γ kann die χ^2 verteilte GröSSe $-2ln(\Gamma)$ berechnet werden.

das Wilk Theorem erlaubt hier $\sqrt{\Lambda'} = X\,\sigma$ so hättet ihr einen genauen Ausdruck für die Signifikanz bekommen

```
In [14]: - 2 * np.log(gamma)
```

$= \Lambda'$

```
Out[14]: 3.118198413376407
```

(✓)

kann benutzt werden, da Nullhypothese durch Beschränkung der Alternativhypothese formuliert werden kann 0,5P.

Aus Vergleich mit einer Tabelle ergibt dies eine Signifikanz von etwa 5%.

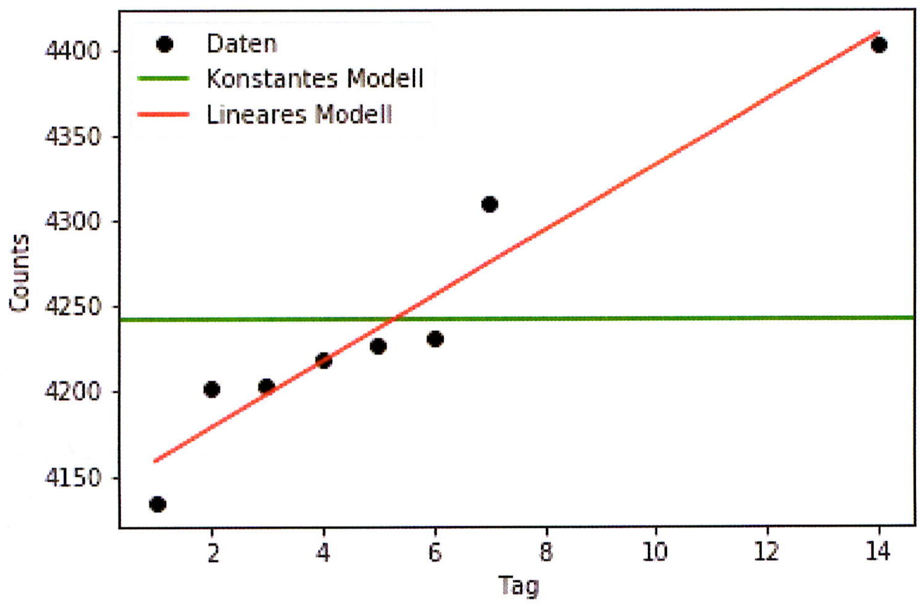
d) Analoges Vorgehen:

Vergleich mit Tabelle ist ok aber eure Schlussfolgerung ist falsch

```
In [15]: counts_d = np.array([4135, 4202, 4203, 4218, 4227, 4231, 4310, 4402])
         days_d = np.append(days, 14)

In [16]: c_mean_d = round(np.mean(counts_d), 0)

In [17]: def llh(params):
             A = params[0]
             B = params[1]
             return + len(counts_d) * B + A * np.sum(days_d) - np.sum(counts_d * np.log(A * da

In [18]: lin_model_d = fmin(llh, [0, counts_d[0]])
         print(f'Parameter für lineares Modell: {lin_model_d}')
```

```
Optimization terminated successfully.
         Current function value: -249462.452622
         Iterations: 146
         Function evaluations: 286
Parameter für lineares Modell: [  19.2184387   4140.10308047]
```

```
In [19]: plt.plot(days_d, counts_d, 'ko', label = 'Daten')
         plt.axhline(y = c_mean_d, label = 'Konstantes Modell', color = 'g')
         plt.plot(days_d, lin_model_d[0] * days_d + lin_model_d[1],
                  color = 'r', label = 'Lineares Modell')
         plt.xlabel('Tag')
         plt.ylabel('Counts')
         plt.legend()
         None
```




```
In [20]: gamma_d = np.prod((c_mean_d / (lin_model_d[0] * days_d + lin_model_d[1]))**counts_d
                              * np.exp(-c_mean_d + lin_model_d[0] * days_d + lin_model_d[1]))

In [21]: - 2 * np.log(gamma_d)

Out[21]: 9.975948394093018
```

≈ 3,1586 (✓)

Aus Vergleich mit einer Tabelle ergibt dies eine Signifikanz von etwa 0.1%.

2P.

0.4 Aufgabe 36: Zwei Histogramme

Berechnungen handschriftlich.

e)

In [22]:
```
n = np.array([111, 188, 333])
m = np.array([15, 36, 30])
N = np.sum(n)
M = np.sum(m)
```

In [23]:
```
chi2 = 1 / N / M * np.sum((N * m - M * n)**2 / (n + m))
```

In [24]:
```
chi2
```

Out[24]:
```
8.429160409693589
```

Nitschke
Grisard

SMD A34

a) $P(c_i|\lambda) = \frac{\lambda^{c_i}}{c_i!}\exp(-\lambda)$ ✓

$\rightarrow \mathcal{L}(\lambda) = \prod_{c_i=1}^{N} p(c_i|\lambda) = \prod_i^N \frac{\lambda^{c_i}}{c_i!}\exp(-\lambda)$ ✓

$-\ln(\mathcal{L}(\lambda)) = \sum_{i=1}^{N} -c_i \ln(\lambda) + \ln(c_i!) + \lambda$

$\frac{\partial \ln(\mathcal{L}(\lambda))}{\partial\lambda} = \sum_{i=1}^{N} \frac{c_i}{\lambda} - 1 \overset{!}{=} 0$

$N = \frac{1}{\lambda}\sum_{i=1}^{N} c_i \iff \lambda = \frac{1}{N}\sum_{i=1}^{N} c_i = \bar{c}_i$ ✓

b) $\lambda' = i\cdot\lambda + b$

$\mathcal{L}(\lambda) = \prod_{i=1}^{N} \frac{(i\lambda+b)^{c_i}}{c_i!}\exp(-i\lambda-b)$

$\ln(\mathcal{L}(\lambda)) = \sum_{i=1}^{N} c_i\ln(i\lambda+b) \mp \ln(c_i!) - i\lambda - b$

$\frac{\partial\ln(\mathcal{L}(\lambda))}{\partial\lambda} = \sum_{i=1}^{N} \frac{c_i}{i\lambda+b}\cdot i - i \overset{!}{=} 0$

$\sum_{i=1}^{N} \frac{ic_i}{i\lambda+b} = \sum_{i=1}^{N} i$

wird beim maximiere weggelassen, da es sich um eine konstante Verschiebung handelt.

→ wird numerisch bestimmt ✓

A35

a) $p(\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$

H_0: $\mu = \mu_0$ σ unbek.

H_1: $\mu =$ unbekannt, σ unbekannt

Testbed.: $\frac{\max \mathcal{L}_{H_0}(\mu_0,\sigma_0)}{\max \mathcal{L}_{H}(\mu_{best},\sigma_{best})} \leq k_\alpha$ mit Signifikanz α ✓ 1P.

b) H_0: $\mathcal{L}(\sigma_0) = \prod_i^N \frac{1}{\sqrt{2\pi}\sigma_0}\exp\left(-\frac{1}{2\sigma_0^2}(x_i-\mu_0)^2\right)$

$\ln(\mathcal{L}(\sigma_0)) = -\sum_{i=1}^{N} \ln(\sqrt{2\pi}\sigma_0) - \frac{1}{2\sigma_0^2}(x_i-\mu_0)^2$

$\frac{\partial\ln(\mathcal{L})}{\partial\sigma_0} = -\sum_{i=1}^{N} \frac{1}{\sigma_0} + \frac{1}{\sigma_0^3}(x_i-\mu_0)^2 \overset{!}{=} 0$

$N\frac{1}{\sigma_0} = \frac{1}{\sigma_0^3}\sum_{i=1}^{N}(x_i-\mu_0)^2 \iff \sigma_0^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_0)^2$ ✓

H_1: max $\ln(L)$ bzgl. μ und σ

bzgl μ: $\frac{\partial}{\partial\mu}\ln(L) = \sum_{N_i}(x_i-\mu)\cdot\frac{1}{\sigma^2} = 0$

$\rightarrow \sum_{i=1}(x_i-\mu) = 0 \quad \Leftrightarrow \mu = \frac{1}{N}\sum_{i=1}^{N} x_i = \bar{x}$ ✓

mit σ aus H_0 folgt $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2$ ✓ 1P.

c) $\frac{L_{H_0}}{L_{H_1}} = \prod_{i=1}^{N}\left(\frac{\sigma}{\sigma_0}\right)\exp\left(-\frac{1}{2\sigma^2}(x_i-\bar{x})^2 + \frac{1}{2\sigma_0^2}(x_i-\mu_0)^2\right)$

$= \prod_{i=1}^{N}\left(\frac{\sigma}{\sigma_0}\right)\exp\left(-\frac{N}{2}+\frac{N}{2}\right) = \prod_{i=1}^{N}\left(\frac{\sigma}{\sigma_0}\right) = \left(\frac{\sigma}{\sigma_0}\right)^N$

$\sigma_0^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_0)^2$

$= \frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x}+\bar{x}-\mu_0)^2$

$= \frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2 + \underbrace{2(x_i-\bar{x})(\bar{x}-\mu_0)} + (\bar{x}-\mu_0)^2$

$= \sum_{i=1}^{N} 2(x_i-\bar{x})(\bar{x}-\mu_0) = \sum_{i=1}^{N} 2(x_i\bar{x}-\bar{x}^2-\mu_0 x_i+\bar{x}\mu_0)$

$= 2N(\bar{x}^2-\bar{x}^2-\mu_0\bar{x}+\bar{x}\mu_0)$

$= 0$

$\rightarrow \sigma_0^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2 + (\bar{x}-\mu_0)^2 = \sigma^2 + \frac{1}{N}\sum_{i=1}^{N}(\bar{x}-\mu_0)^2$

~~$\rightarrow\left(\frac{\sigma^2}{\sigma_0^2}\right)^{N/2} = ($~~ $\quad \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2 = \frac{N-1}{N}\cdot s^2$

$\rightarrow\left(\frac{\sigma^2}{\sigma_0^2}\right)^{N/2} = \left(\frac{\frac{N-1}{N}\cdot s^2}{\frac{N-1}{N}s^2 + \cancel{\frac{N}{s^2(N-1)}}\frac{1}{N}\sum_{i=1}^{N}(\bar{x}-\mu_0)^2}\right)^{N/2}$

$= \left(\frac{1}{1+\frac{N}{s^2(N-1)}(\bar{x}-\mu_0)^2}\right)^{N/2} < k_\alpha$

$\rightarrow \frac{N}{N-1}\frac{1}{s^2}(\bar{x}-\mu_0)^2 > \left(\frac{1}{k_\alpha}\right)^{2/N} - 1$

$\rightarrow \underbrace{\frac{\sqrt{N}(\bar{x}-\mu_0)}{s}} > \sqrt{\left(\left(\frac{1}{k_\alpha}\right)^{2/N}-1\right)(N-1)}$ ✓ 2P.

folgt der t-Statistik

d) fehlt

für Werte einsetzen wieder zu faul ?

Aufgabe 36

a) jeweils Poisson-Verteilung ✓

Wahrscheinlichkeit, dass Beobachtung in Bin i fällt: ✓

$p_i = \frac{n_i}{N}$ bzw. $\frac{m_i}{M}$, Nullhypothese

$\rightarrow \frac{n_i}{N} = \frac{m_i}{M}$ ✓

PDF: $\cancel{P_i} = \frac{e^{-Np_i}(Np_i)^{n_i}}{n_i!}$ ✓ bzw. $\frac{e^{-Mp_i}(Mp_i)^{m_i}}{m_i!}$ ✓

1P.

b) $\mathcal{L} = \frac{e^{-Np_i}(Np_i)^{n_i}}{n_i!} \cdot \frac{e^{-Mp_i}(Mp_i)^{m_i}}{m_i!}$ ✓

$\sim p_i^{n_i+m_i} e^{-p_i(M+N)}$

$\frac{\partial \mathcal{L}}{\partial p_i} \sim \left[(n_i+m_i)p_i^{-1} - (N+M)\right] p_i^{n_i+m_i} e^{-p_i(N+M)} \stackrel{!}{=} 0$

$\Leftrightarrow n_i + m_i - (N+M)p_i = 0 \Leftrightarrow p_i = \frac{n_i+m_i}{N+M}$ ✓ 0,_P.

2. Ableitung zum test, ob es sich um ein globales Maximum handelt

−0,25P.

c) $\chi^2 = \sum_{i=1}^{r} \frac{(n_i - n_i^{Modell})^2}{n_i^{Modell}} + \sum_{i=1}^{r} \frac{(m_i - m_i^{Modell})^2}{m_i^{Modell}}$ ✓

wobei $n_i^{Modell} = N p_i$, $m_i^{Modell} = M p_i$

$\rightarrow = \frac{1}{NM} \sum_{i=1}^{r} \frac{(N m_i - M n_i)^2}{n_i + m_i}$

1P.

Tipp

d) Schätze N, M und die p_i, wobei ein p_i aus $\sum p_i = 1$ festgelegt wird

→ Freiheitsgrade $= 2r - 2 - (r-1)$
$= r - 1$ ✓

n_i und m_i
muss relativ hoch sein, damit Annahme einer Graußverteilung hinreichend gut. (✓)

1P.

e)

α	$\chi^2(24)$	
0,1	4,61 ✓	abgelehnt / verworfen ✓
0,05	5,99	abgelehnt ✓
0,01	9,21	nicht abgelehnt. ✓

1P.