

aufgabe2

January 24, 2019

1 Aufgabe 33

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

1.1 a)

Die Erwartungswerte von Gauß- und Poisson-Verteilung müssen gleich sein $\mu = \lambda$. Die Standardabweichung Gauß- und Poisson-Verteilung müssen auch gleich sein $\sigma = \sqrt{\lambda}$. Außerdem sollte $\lambda \gg 1$ sein.

1.2 b)

```
In [2]: def kstest(X, Y, alpha):
    # X einzelne Bineinträge
    # Y einzelne Bineinträge
    FX = np.array([0]) # empirische Verteilungsfunktion
    FY = np.array([0]) # empirische Verteilungsfunktion
    k1 = 0
    k2 = 0

    for i in X: # Summe der Bineinträge mit Binnummer < x
        FX = np.append(FX, (i+FX[k1]))
        k1 += 1

    for i in Y: # Summe der Bineinträge mit Binnummer < x
        FY = np.append(FY, (i+FY[k2]))
        k2 += 1

    FX = np.delete(FX, 0)
    FY = np.delete(FY, 0)

    FX = FX/np.sum(X) # geteilt durch Gesamtanzahl der Bineinträge
    FY = FY/np.sum(Y) # geteilt durch Gesamtanzahl der Bineinträge
```

```

d = np.max(np.abs(FX-FY)) #maximale Abstand

K_alpha = np.sqrt(np.log(2/alpha)/2)

bestanden = True

#Überprüfung der Hypothese
if np.sqrt(np.sum(X)*np.sum(Y)/(np.sum(X)+np.sum(Y)))*d > K_alpha:
    bestanden = False

return bestanden

```

1.3 c)

```

In [3]: prng = np.random.RandomState(10)
        abbruch = True

for lam in np.linspace(1,100,100):
    if abbruch:
        #Gau-Verteilung
        my = lam
        sigma = np.sqrt(lam)
        normal = prng.normal(my, sigma, 10000)
        normal = np.round(normal)

        # Entfernen aller Werte auerhalb der 5 Sigma Umgebung
        normal = np.delete(normal, np.where(normal < my-5*sigma))
        normal = np.delete(normal, np.where(normal > my+5*sigma))

        # Binning der Gau-Verteilung
        normal_bins = np.histogram(normal, bins=np.linspace(my-5*sigma,my+5*sigma,101))

        # Poisson-Verteilung
        poisson = prng.poisson(lam, 10000)

        # Entfernen aller Werte auerhalb der 5 Sigma Umgebung
        poisson = np.delete(poisson, np.where(poisson < my-5*sigma))
        poisson = np.delete(poisson, np.where(poisson > my+5*sigma))

        # Binning der Poisson-Verteilung
        poisson_bins = np.histogram(poisson, bins=np.linspace(my-5*sigma,my+5*sigma,101))

        # Durchführung des Kolmogorow-Smirnow-Tests
        bestanden = kstest(poisson_bins[0], normal_bins[0], 0.05) # alpha = 0.05

    if bestanden:
        print("Für dieses Lambda-Wert sind sich beide Verteilungen so aehnlich,")

```

```

print("dass der Test sie nicht unterscheiden kann:")
print(lam)
abbruch = False

```

Für dieses Lambda-Wert sind sich beide Verteilungen so aehnlich,
dass der Test sie nicht unterscheiden kann:
8.0

1.4 d)

In [4]: abbruch = True

```

for lam in np.linspace(1,100,100):
    if abbruch:
        #Gau-Verteilung
        my = lam
        sigma = np.sqrt(lam)
        normal = prng.normal(my, sigma, 10000)
        normal = np.round(normal)

        # Entfernen aller Werte auerhalb der 5 Sigma Umgebung
        normal = np.delete(normal, np.where(normal < my-5*sigma))
        normal = np.delete(normal, np.where(normal > my+5*sigma))

        # Binning der Gau-Verteilung
        normal_bins = np.histogram(normal, bins=np.linspace(my-5*sigma,my+5*sigma,101))

        # Poisson-Verteilung
        poisson = prng.poisson(lam, 10000)

        # Entfernen aller Werte auerhalb der 5 Sigma Umgebung
        poisson = np.delete(poisson, np.where(poisson < my-5*sigma))
        poisson = np.delete(poisson, np.where(poisson > my+5*sigma))

        # Binning der Poisson-Verteilung
        poisson_bins = np.histogram(poisson, bins=np.linspace(my-5*sigma,my+5*sigma,101))

        # Durchführung des Kolmogorow-Smirnow-Tests
        bestanden = kstest(poisson_bins[0], normal_bins[0], 0.025) # alpha = 0.025

        if bestanden:
            print("Für dieses Lambda-Wert sind sich beide Verteilungen so aehnlich,")
            print("dass der Test sie nicht unterscheiden kann:")
            print(lam)
            abbruch = False

```

Für dieses Lambda-Wert sind sich beide Verteilungen so aehnlich,
dass der Test sie nicht unterscheiden kann:
9.0

```
In [5]: abbruch = True
```

```
for lam in np.linspace(1,100,100):
    if abbruch:
        #Gau-Verteilung
        my = lam
        sigma = np.sqrt(lam)
        normal = prng.normal(my, sigma, 10000)
        normal = np.round(normal)

        # Entfernen aller Werte auerhalb der 5 Sigma Umgebung
        normal = np.delete(normal, np.where(normal < my-5*sigma))
        normal = np.delete(normal, np.where(normal > my+5*sigma))

        # Binning der Gau-Verteilung
        normal_bins = np.histogram(normal, bins=np.linspace(my-5*sigma,my+5*sigma,101))

        # Poisson-Verteilung
        poisson = prng.poisson(lam, 10000)

        # Entfernen aller Werte auerhalb der 5 Sigma Umgebung
        poisson = np.delete(poisson, np.where(poisson < my-5*sigma))
        poisson = np.delete(poisson, np.where(poisson > my+5*sigma))

        # Binning der Poisson-Verteilung
        poisson_bins = np.histogram(poisson, bins=np.linspace(my-5*sigma,my+5*sigma,101))

        # Durchführung des Kolmogorow-Smirnow-Tests
        bestanden = kstest(poisson_bins[0], normal_bins[0], 0.001) # alpha = 0.001

        if bestanden:
            print("Für dieses Lambda-Wert sind sich beide Verteilungen so aehnlich,")
            print("dass der Test sie nicht unterscheiden kann:")
            print(lam)
            abbruch = False
```

Für dieses Lambda-Wert sind sich beide Verteilungen so aehnlich,
dass der Test sie nicht unterscheiden kann:
3.0