In [33]:

```
import numpy as np
import matplotlib as plt
```

Aufgabe 29

In [48]:

```
L1 = np.array([0.13, 1.5, 0.5])
L2 = np.array([2.0, 0.5, 0.05])
L3 = np.array([0.07, 0.5, 1.3])

p1 = 0.8
p2 = 0.1
p3 = 0.1
```

Durch Bayes ergibt sich

$$P' = P_0 \cdot \frac{LR}{(1 - P_0 + P_0 \cdot LR)}.$$

Also müssen wir noch die Ratios bestimmen. Wir betrachten hier die positive ratios. Wir gehen davon aus, dass gilt:

Die Ratios müssen wir dann aber noch normieren. Also summieren wir alle zusammen und

In [47]:

```
def func(p, ratio):
    return p*ratio/(1-p+p*ratio)
```

In [56]:

```
ratio11 = L1[0]/L1.sum()
P11 = func(p1, ratio11)*100
ratio12 = L1[1]/L1.sum()
P12 = func(p2, ratio12)*100
ratio13 = L1[2]/L1.sum()
P13 = func(p3, ratio13)*100

ratio21 = L2[0]/L2.sum()
P21 = func(p1, ratio21)*100
ratio22 = L2[1]/L2.sum()
P22 = func(p2, ratio22)*100
ratio23 = L2[2]/L2.sum()
P23 = func(p3, ratio23)*100

ratio31 = L3[0]/L3.sum()
P31 = func(p1, ratio31)*100
ratio32 = L3[1]/L3.sum()
P32 = func(p2, ratio32)*100
ratio33 = L3[2]/L3.sum()
P33 = func(p3, ratio33)*100

print(f"a) Wahrscheinlichkeit, dass... \nein Pion erscheint: {P11:.2}%\nein Kaon erscheint: {P12:.2}%\nein Proton erscheint: {P13:.2}%.")
print(f"a) Wahrscheinlichkeit, dass... \nein Pion erscheint: {P21:.2}%\nein Kaon erscheint: {P22:.2}%\nein Proton erscheint: {P23:.2}%.")
print(f"a) Wahrscheinlichkeit, dass... \nein Pion erscheint: {P31:.2}%\nein Kaon erscheint: {P32:.2}%\nein Proton erscheint: {P33:.2}%.")
```

```
a) Wahrscheinlichkeit, dass...
ein Pion erscheint: 2e+01%
ein Kaon erscheint: 7.3%
ein Proton erscheint: 2.5%.
a) Wahrscheinlichkeit, dass...
ein Pion erscheint: 7.6e+01%
ein Kaon erscheint: 2.1%
ein Proton erscheint: 0.22%.
a) Wahrscheinlichkeit, dass...
ein Pion erscheint: 1.3e+01%
ein Kaon erscheint: 2.9%
ein Proton erscheint: 7.2%.
```

Also irgendwo ergibt das gar keinen Sinn, glaub ich. Aber ich weiß auch nicht, wie ich es sonst machen soll.

Aufgabe 30

a)

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

Wir setzen s = 0.

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

Ableiten und gleich null setzen:

$$\frac{\partial ln(L)}{\partial b} = 0 = \frac{N_{off}}{b} + \frac{N_{on}}{b} - (1 + \alpha)$$

Daraus folgt:

$$b = \frac{(N_{off} + N_{on})}{(1 - \alpha)}$$

b)

$$\lambda = \frac{\hat{b_0}}{\hat{b}} = \frac{(N_{off} + N_{on})}{(1 - \alpha) \cdot N_{off}}$$

Aufgabe 32

In [34]:

```
en_diff= np.array([31.6, 32.2, 31.2, 31.9, 31.3, 30.8, 31.3])
err = 0.5
hypo1 = 31.3
hypo2 = 30.7
```

In [35]:

```
chisquare1 = np.sum((hypo1 - en_diff)**2/err**2)
chisquare2 = np.sum((hypo2 - en_diff)**2/err**2)
```

In [36]:

```
print("a): ", chisquare1)
print("b): ", chisquare2)
```

```
a):  6.080000000000007
b):  21.920000000000073
```

Alpha liegt bei 14,067. Insofern wird der Wert bei b) abgelehnt.