Gruppe 7

A	10	11	12	Σ
MP	5	5	10	20
EP	4	5	8 ~~6~~	17 ~~.5~~

sehr gut!

blatt04_nitschke_grisard

November 15, 2018

1 Blatt 4

1.1 Aufgabe 10: Zwei Populationen

a) 2/2 P

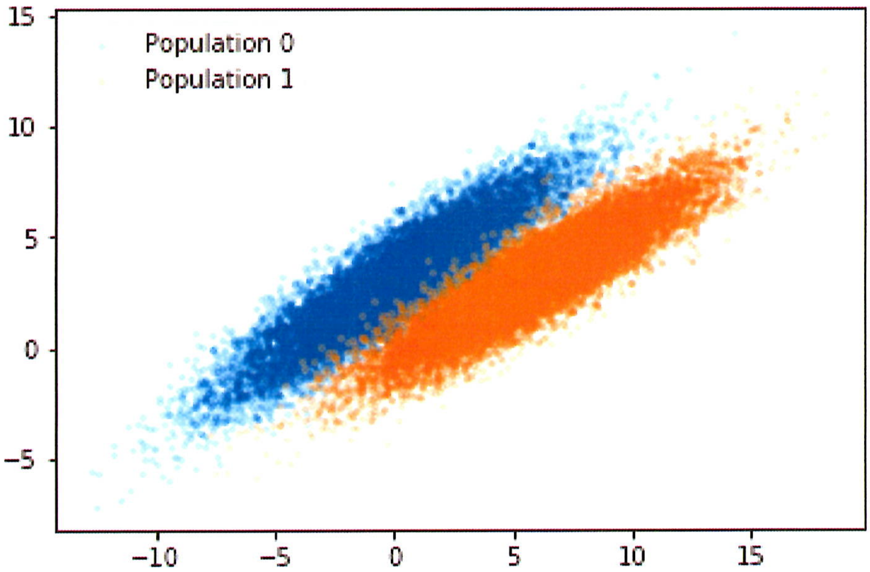
```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd

In [2]: mux0 = 0
        muy0 = 3
        sigx0 = 3.5
        sigy0 = 2.6
        cor0 = 0.9
        cov0 = cor0 * sigx0 * sigy0

        cov_mat0 = np.array([[sigx0**2, cov0], [cov0, sigy0**2]])

In [3]: population0_10000 = np.random.multivariate_normal([mux0, muy0], cov_mat0, 10000)

In [4]: mux1 = 6
        sigx1 = 3.5
        a = -0.5
        b = 0.6
        var_yx = 1

        muy1 = a + b * mux1
        sigy1 = np.sqrt(b**2 * sigx1**2 + var_yx)
        cor1 = np.sqrt(b**2 * sigx1**2 / sigy1**2)

        cov1 = cor1 * sigx1 * sigy1
        cov_mat1 = np.array([[sigx1**2, cov1], [cov1, sigy1**2]])

        print(muy1, sigy1, cor1)

3.0999999999999996 2.3259406699226015 0.9028605188239304

In [5]: population1 = np.random.multivariate_normal([mux1, muy1], cov_mat1, 10000)
```

Wie kann man 2D gauß mit 1D Funktion erzeugen?
auf 1D x sample mit Erwartungswert → auf y Wertesample

b) Zeichne Scatter-Plots: 7/7 P

In [6]:
```
plt.scatter(population0_10000[:, 0], population0_10000[:, 1], s=5, alpha=0.2, label =
plt.scatter(population1[:, 0], population1[:, 1], s=5, alpha=0.2, label = 'Population :
plt.legend()
```

Out[6]:
```
<matplotlib.legend.Legend at 0x2291ab5f940>
```



✓

c) fehlt 0/1 P $\quad Corr_{ij} = \frac{Cov_{ij}}{\sigma_i \sigma_j} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$

d) 7/7 P

In [7]:
```
population0_1000 = np.random.multivariate_normal([mux0, muy0], cov_mat0, 1000)

population0_10000_df = pd.DataFrame({
    'x': population0_10000[:, 0],
    'y': population0_10000[:, 1]
})

population0_1000_df = pd.DataFrame({
    'x': population0_1000[:, 0],
    'y': population0_1000[:, 1]
})

population1_df = pd.DataFrame({
    'x': population1[:, 0],
    'y': population1[:, 1]
})

population0_10000_df.to_hdf('sample.hdf5', key = 'population0_10000')
```
✓ ✓ ✓

Aufgabe 10 a)

$$\gamma = \frac{1}{2(1-\rho^2)}$$

$$\lambda = \frac{1}{\sqrt{2\pi}\,\sigma_y \sqrt{1-\rho^2}}$$

$$Var(Y|X) = E(Y^2|X) - E(Y|X)^2$$

$$E(Y|X) = \left(\rho \frac{\sigma_y}{\sigma_x}(x-\mu_x) - \mu_y\right)$$ bekannt

vom letzten Blatt?

$$E(Y^2|X) = \int_{-\infty}^{\infty} y^2 f(y|x)\, dy \qquad u_y = \frac{y-\bar{y}}{\sigma_y} \qquad dy = du_y \sigma_y$$

$$= \lambda \int_{-\infty}^{\infty} (u_y\sigma_y + \mu_y)^2 \exp(-\gamma[u_y - \rho u_x]^2)\,\sigma_y\, du_y$$

$$= \lambda\sigma_y \int_{-\infty}^{\infty} (u_y^2\sigma_y^2 + 2u_y\sigma_y\mu_y + \mu_y^2) \exp(-\gamma[u_y-\rho u_x]^2)\, du_y$$

$$= \lambda\sigma_y \Big\{\sigma_y^2 \int_{-\infty}^{\infty} u_y^2 \exp(-\gamma[u_y-\rho u_x]^2)\, du_y$$

$$- 2\sigma_y\mu_y \int_{-\infty}^{\infty} u_y \exp(-\gamma[u_y-\rho u_x]^2)\, du_y\Big\}$$

$$+ \mu_y^2$$

$$u_y - \rho u_x = k$$

$$= \lambda\sigma_y^3 \int_{-\infty}^{\infty} (k^2 + 2\rho u_x k + \rho^2 u_x^2) \exp(-\gamma k^2)\, dk$$

$$- 2\lambda\sigma_y^2\mu_y\rho u_x \int_{-\infty}^{\infty} \exp(-\gamma k^2)\, dk + \mu_y^2$$

$$= \lambda\sigma_y^3 \left\{\int_{-\infty}^{\infty} k^2 \exp(-\gamma k^2)\, dk + \rho^2 u_x^2 \int_{-\infty}^{\infty} \exp(-\gamma k^2)\, dk\right\}$$

$$- 2\lambda\sigma_y^2\mu_y\rho u_x \sqrt{\frac{\pi}{\gamma}} + \mu_y^2$$

$$= \lambda\sigma_y^3 \left\{\sqrt{\frac{\pi}{\gamma}} \cdot \frac{1}{2\gamma} + \rho^2 u_x^2 \sqrt{\frac{\pi}{\gamma}}\right\} - 2\lambda\sigma_y^2\mu_y\rho u_x \sqrt{\frac{\pi}{\gamma}} + \mu_y^2$$

$$= \frac{\sigma_y^2}{\cancel{\sqrt{2\pi}\sqrt{1-\rho^2}}}\left(\cancel{\sqrt{\pi}\sqrt{2(1-\rho^2)}}(1-\rho^2) + \rho^2 u_x^2 \cancel{\sqrt{\pi}\sqrt{2(1-\rho^2)}}\right)$$

$$- 2\,\frac{\cancel{\sqrt{2}}\,\sigma_y\mu_y\rho u_x \cancel{\sqrt{\pi}}\,\cancel{\sqrt{1-\rho^2}}}{\cancel{\sqrt{2\pi}\sqrt{1-\rho^2}}} + \mu_y^2$$

$$= \sigma_y^2\left((1-\rho^2) + \rho^2 u_x^2\right) - 2\sigma_y\mu_y\rho u_x + \mu_y^2$$

$\rightarrow \mathrm{Var}(Y|X)$

$$= \sigma_y^2\left((1-\rho^2) + \rho^2\left(\frac{x-\mu_x}{\sigma_x}\right)^2\right) - 2\frac{\sigma_y}{\sigma_x}\rho(x-\mu_y)\mu_y + \mu_y^2$$

$$- \left(\rho\frac{\sigma_y}{\sigma_x}(x-\mu_x)-\mu_y\right)^2$$

$$= \sigma_y^2(1-\rho^2) + \sigma_y^2\rho^2\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - 2\frac{\sigma_y}{\sigma_x}\rho(x-\mu_y)\mu_y + \mu_y^2$$

$$-\rho^2\frac{\sigma_y^2}{\sigma_x^2}(x-\mu_y)^2 + 2\rho\frac{\sigma_y}{\sigma_x}(x-\mu_y)\mu_y - \mu_y^2$$

$$= \cancel{\sigma_y^2(1-\sigma_x^2)}\ \sigma_y^2(1-\rho^2) \quad ✓$$

$$\rightarrow \quad 1 = (1-\rho^2)\sigma_y^2 \quad (= \sigma_{y|x})$$

mit $E(y|x) = \rho\frac{\sigma_y}{\sigma_x}(x-\mu_x) + \mu_y \overset{!}{=} a + bx$ ✓

folgt $a = -0{,}5 = -\cancel{+}\rho\frac{\sigma_y}{\sigma_x}\mu_x + \mu_y$

$$b = 0.6 = \rho\frac{\sigma_y}{\sigma_x} \quad \Leftrightarrow \quad \rho = b\frac{\sigma_x}{\sigma_y} \quad ✓$$

$$\sigma_{y|x}\ \cancel{1} = \left(1 - b^2\frac{\sigma_x^2}{\sigma_y^2}\right)\sigma_y^2$$

$$= \sigma_y^2 - b^2\sigma_x^2 \quad \Leftrightarrow \quad \sigma_y = \sqrt{\underset{\sigma_{y|x}}{\cancel{1}} + b^2\sigma_x^2} \quad (✓)$$

$(\sigma_{y|x} = 1)$

$$\rho = b\frac{\sigma_x}{\sigma_y} = b\sigma_x \cdot \frac{1}{\sqrt{\underset{\sigma_{y|x}}{\cancel{1}} + b^2\sigma_x^2}} \quad (✓)$$

$$a = \underline{\cancel{+}}\, b\frac{\sigma_x}{\sigma_y}\cdot\frac{\sigma_y}{\sigma_x}\mu_x + \mu_y = -\cancel{+}b\mu_x + \mu_y$$

Vorzeichen plötzlich wieder

$$\Leftrightarrow \quad \mu_y = (+b\mu_x + a) \quad (✓) \text{ richtig}$$

2/2 P

Schöne Lösung und richtig (bis auf Vorzeichen), geht aber erheblich einfacher (siehe Übung).

SMD Blatt 4

A11:

Pop 0 $\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}2\\1\end{pmatrix}\begin{pmatrix}1{,}5\\2\end{pmatrix}\begin{pmatrix}2\\2\end{pmatrix}\begin{pmatrix}2\\3\end{pmatrix}\begin{pmatrix}3\\3\end{pmatrix}$

$$\vec{\mu}_0 = \begin{pmatrix}23/12\\2\end{pmatrix} \checkmark \quad S_W = \sum_j S_j \quad \text{mit} \quad S_j = \sum_i (\vec{x}_i - \vec{\mu}_j)(\vec{x}_i - \vec{\mu}_j)^T$$

$$S_0 = \sum_i (\vec{x}_i - \vec{\mu}_0)(\vec{x}_i - \vec{\mu}_0)^T$$
$$= \begin{pmatrix}-11/12\\-1\end{pmatrix}(-11/12, -1) + \begin{pmatrix}1/12\\-1\end{pmatrix}(1/12, -1) + \begin{pmatrix}-5/12\\0\end{pmatrix}(-5/12, 0)$$
$$+ \begin{pmatrix}1/12\\0\end{pmatrix}(1/12, 0) + \begin{pmatrix}1/12\\1\end{pmatrix}(1/12, 1) + \begin{pmatrix}13/12\\1\end{pmatrix}(13/12, 1)$$
$$= \begin{pmatrix}121/144 & 11/12\\11/12 & 1\end{pmatrix} + \begin{pmatrix}1/144 & -1/12\\-1/12 & 1\end{pmatrix} + \begin{pmatrix}25/144 & 0\\0 & 0\end{pmatrix}$$
$$+ \begin{pmatrix}1/144 & 0\\0 & 0\end{pmatrix} + \begin{pmatrix}1/144 & 1/12\\1/12 & 1\end{pmatrix} + \begin{pmatrix}169/144 & 13/12\\13/12 & 1\end{pmatrix}$$
$$= \begin{pmatrix}53/24 & 2\\2 & 4\end{pmatrix} \checkmark$$

Pop 1: $\begin{pmatrix}1{,}5\\1\end{pmatrix}\begin{pmatrix}2{,}5\\1\end{pmatrix}\begin{pmatrix}3{,}5\\1\end{pmatrix}\begin{pmatrix}2{,}5\\2\end{pmatrix}\begin{pmatrix}3{,}5\\2\end{pmatrix}\begin{pmatrix}4{,}5\\2\end{pmatrix}$

$$\vec{\mu}_1 = \begin{pmatrix}3\\3/2\end{pmatrix} \checkmark$$

$$S_1 = \begin{pmatrix}-3/2\\-1/2\end{pmatrix}(-3/2, -1/2) + \begin{pmatrix}-1/2\\-1/2\end{pmatrix}(-1/2, -1/2) + \begin{pmatrix}1/2\\-1/2\end{pmatrix}(1/2, -1/2)$$
$$+ \begin{pmatrix}-1/2\\1/2\end{pmatrix}(-1/2, 1/2) + \begin{pmatrix}1/2\\1/2\end{pmatrix}(1/2, 1/2) + \begin{pmatrix}3/2\\1/2\end{pmatrix}(3/2, 1/2)$$
$$= \begin{pmatrix}9/4 & 3/4\\3/4 & 1/4\end{pmatrix} + \begin{pmatrix}1/4 & 1/4\\1/4 & 1/4\end{pmatrix} + \begin{pmatrix}1/4 & -1/4\\-1/4 & 1/4\end{pmatrix}$$
$$+ \begin{pmatrix}1/4 & -1/4\\-1/4 & 1/4\end{pmatrix} + \begin{pmatrix}1/4 & 1/4\\1/4 & 1/4\end{pmatrix} + \begin{pmatrix}9/4 & 3/4\\3/4 & 1/4\end{pmatrix}$$
$$= \begin{pmatrix}22/4 & 6/4\\6/4 & 6/4\end{pmatrix} = \begin{pmatrix}11/2 & 3/2\\3/2 & 3/2\end{pmatrix} \checkmark$$

$$\Rightarrow S_W = S_0 + S_1 = \begin{pmatrix}185/24 & 7/2\\7/2 & 11/2\end{pmatrix} \checkmark$$

1/1 P a)

nicht gefragt

$$S_B = (\vec{\mu}_1 - \vec{\mu}_2)(\vec{\mu}_1 - \vec{\mu}_2)^T$$

$$= \begin{pmatrix} -13/12 \\ 1/2 \end{pmatrix} \left(-13/12,\ 1/2\right) = \begin{pmatrix} 169/144 & -13/24 \\ -13/24 & 1/4 \end{pmatrix}$$

b) $\vec{\lambda}^* = S_w^{-1} (\vec{\mu}_1 - \vec{\mu}_2)$

1/1 P

$$S_w^{-1} = \frac{1}{\det(S_w)} \begin{pmatrix} 11/2 & -7/2 \\ -7/2 & 185/24 \end{pmatrix}$$

$$\det(S_w) = 1447/48$$

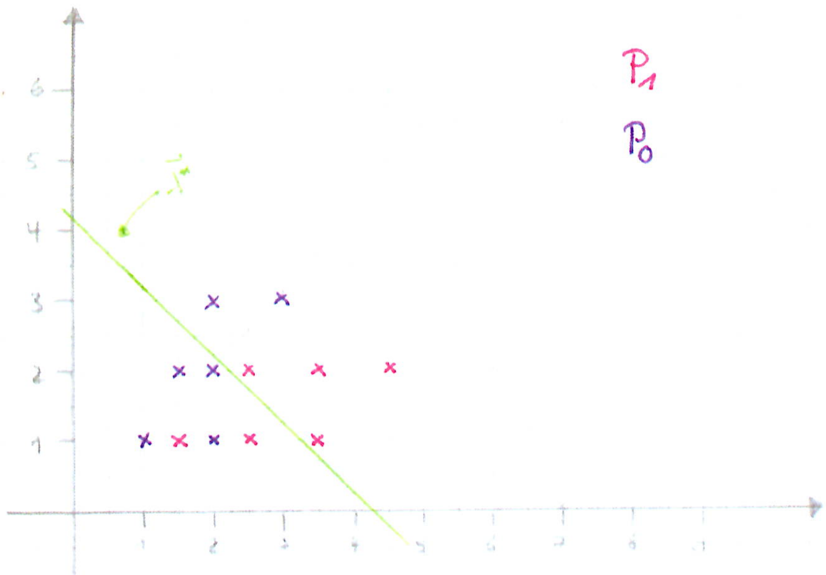
$$S_w^{-1} = \begin{pmatrix} 264/1447 & \cancel{84/1241} \; -168/1447 \\ -168/1447 & 370/1447 \end{pmatrix}$$ ✓

$$\vec{\lambda}^* = S_w^{-1} (\vec{\mu}_1 - \vec{\mu}_2)$$

$$= S_w^{-1} \begin{pmatrix} -13/12 \\ 1/2 \end{pmatrix} = \begin{pmatrix} -370/1447 \\ 367/1447 \end{pmatrix}$$ (✓) Bis auf Normierung

c)

1/1 P



$$|S_w^{-1} S_B - D| = 0 \Rightarrow EW = 0;\ 0{,}403824$$

$$\vec{v}_1 = \begin{pmatrix} 0{,}70557 \\ 0{,}415058 \end{pmatrix} \quad \vec{v}_2 = \begin{pmatrix} -0{,}7042263 \\ 0{,}907558 \end{pmatrix}$$

$(\vec{\lambda}^*)^2 \approx 0{,}064853 \approx 0{,}065$

d) Pop 0 $\quad \vec{\lambda}^* \vec{x}_1 = (-370/1447 \cdot 1 \quad + 367/1447 \cdot 1)$

7/7P

$= -3/1447 \approx -0{,}002 \rightarrow p_1 \approx -0{,}03$

$\vec{\lambda}^* \vec{x}_2 = -373/1447 \approx -0{,}258 \rightarrow p_2 \approx -3{,}97$

Graphisch hätte gereicht aber so ist's besser!

$\vec{\lambda}^* x_3 = 179/1447 \approx 0{,}124 \rightarrow p_3 \approx +1{,}91$

$\vec{\lambda}^* x_4 = -6/1447 \approx -0{,}004 \rightarrow p_4 \approx -0{,}06$

$\vec{\lambda}^* x_5 = 361/1447 \approx 0{,}249 \rightarrow p_5 \approx +3{,}83$

$\vec{\lambda}^* x_6 = -9/1447 \approx -0{,}006 \rightarrow p_6 \approx -0{,}09$

Pop 1:

$\vec{\lambda}^* \vec{x}_1 = -183/1447 \approx -0{,}13 \rightarrow p_1 \approx \cancel{18{,}84}\ -2$

$\vec{\lambda}^* \vec{x}_2 = -558/1447 \approx -0{,}386 \rightarrow p_2 \approx -5{,}94$

$\vec{\lambda}^* \vec{x}_3 = -928/1447 \approx -0{,}641 \rightarrow p_3 \approx -9{,}861$

$\vec{\lambda}^* \vec{x}_4 = -191/1447 \approx -0{,}132 \rightarrow p_4 \approx -2{,}03$

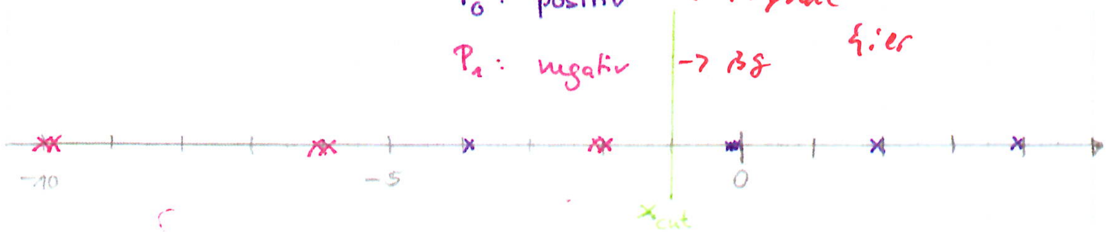
$\vec{\lambda}^* \vec{x}_5 = -561/1447 \approx -0{,}388 \rightarrow p_5 \approx -5{,}97$

$\vec{\lambda}^* \vec{x}_6 = -931/1447 \approx -0{,}643 \rightarrow p_6 \approx -9{,}89$

P_0: positiv $\rightarrow$ Signal

P_1: negativ $\rightarrow$ BG

hier



e) Reinheit: $\frac{tp}{tp+fp}$ $\qquad$ Effizienz $= \frac{tp}{tp+fn}$

7/7P

$\rightarrow$ minimiere fp und fn $\rightarrow$ mit der x_{cut} $fp=0$, $fn=1$

bzw. maximale Reinheit

$\rightarrow$ Reinheit $= \frac{5}{5+0} = 1 \qquad$ Effizienz $= \frac{5}{5+1} = \frac{5}{6} \approx 0{,}8\overline{3}$

```
population0_1000_df.to_hdf('sample.hdf5', key = 'population0_1000')
population1_df.to_hdf('sample.hdf5', key = 'population1')
```

1.2 Aufgabe 11: Fisher-Diskriminante: Per Hand

siehe Abgabe

1.3 Aufgabe 12: Fisher-Diskriminante: Implementierung

Lade Daten:

```
In [8]: import pandas as pd
        P_0 = pd.read_hdf('sample.hdf5', key='population0_10000')
        P_1 = pd.read_hdf('sample.hdf5', key='population1')
        P_0.head()
```

```
Out[8]:            x         y
        0  -0.470846  2.009796
        1   1.547380  3.960312
        2   1.259468  3.819498
        3   4.766957  6.299630
        4   0.340275  2.609898
```

a) Berechne Mittelwerte: 1/1 P

```
In [9]: mu0 = np.matrix([P_0.x.mean(), P_0.y.mean()]).T
        mu1 = np.matrix([P_1.x.mean(), P_1.y.mean()]).T
```

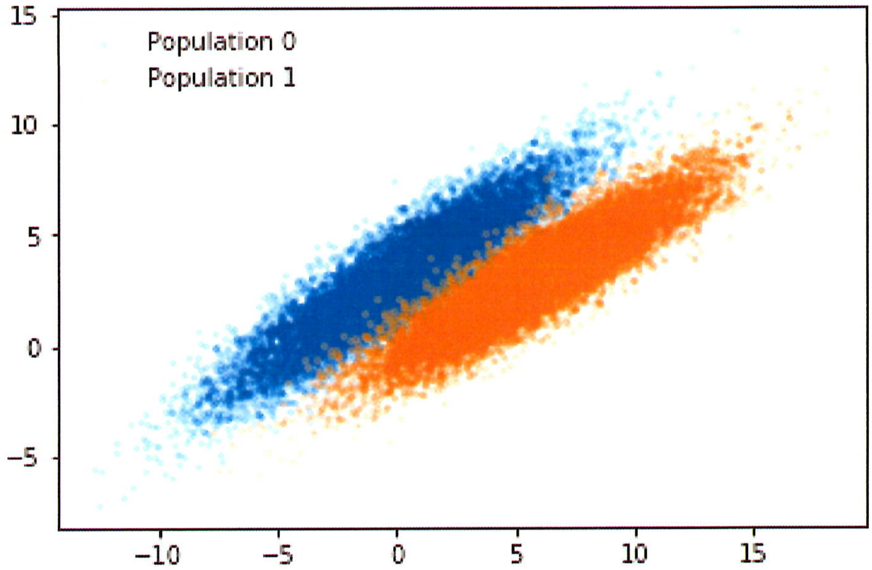
✓ aber np.mean(P_0/1) geht auch!

1/2 P **b)** Berechne Kovarianzmatrizen: Kombinierte Kovarianzmatrix fehlt

```
In [10]: V_0 = P_0.cov()
         V_1 = P_0.cov()
```

✓ (✓) 1

```
In [11]: plt.scatter(P_0.x, P_0.y, s = 5, alpha = 0.2, label = 'Population 0')
         plt.scatter(P_1.x, P_1.y, s = 5, alpha = 0.2, label = 'Population 1')
         plt.legend()

Out[11]: <matplotlib.legend.Legend at 0x2291b7f3d68>
```



c) Konstruiere $\vec{\lambda}$: 1/1 P

nicht machen

```
In [12]: S_0 = np.sum([(xi.T - mu0) * (xi.T - mu0).T for xi in np.matrix(P_0)], axis = 0)
         S_1 = np.sum([(xi.T - mu1) * (xi.T - mu1).T for xi in np.matrix(P_1)], axis = 0)
         S_W = np.matrix(S_0 + S_1)
```

✓

Alternativ: S_0 = P_0.cov() • (len(P_0)-1)

```
In [13]: lam = S_W.I * (mu1 - mu0)
         normed_lam = lam /  np.sqrt(lam[0]**2 + lam[1]**2)
         lam_array = np.array([lam[0], lam[1]])[:, 0]
         print(lam_array[1] / lam_array[0])
```

Nicht über numpy arrays iterieren!

```
[-1.26732449]
```

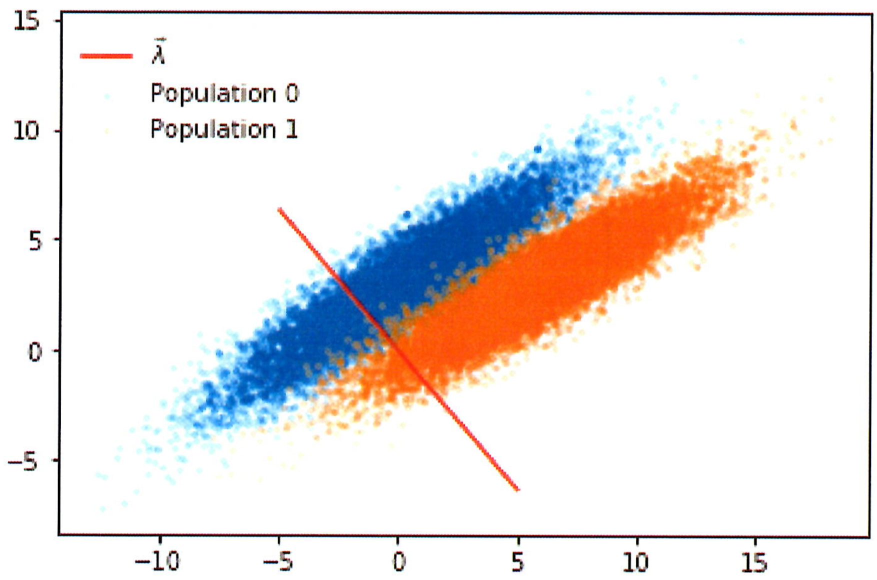
Die Geradengleichung lautet:

7?

$$y(x) \approx -1.269 \cdot x$$ ✓

kann man noch vergleichen, braucht man hier aber nicht

```
In [14]: xplot = np.linspace(-5, 5, 100)
         plt.plot(xplot, lam_array[1] / lam_array[0] * xplot,
                  color = 'red', label = r'$\vec{\lambda}$')

         plt.scatter(P_0.x, P_0.y, s = 5, alpha = 0.2, label = 'Population 0')
         plt.scatter(P_1.x, P_1.y, s = 5, alpha = 0.2, label = 'Population 1')
         plt.legend()

Out[14]: <matplotlib.legend.Legend at 0x2291b838978>
```



✓

d) Stelle die Projektionen in einem Histogramm dar: 1/1 P

```
In [15]: projection_0 = np.array([(xi * normed_lam)[0, 0] for xi in np.matrix(P_0)])
         projection_1 = np.array([(xi * normed_lam)[0, 0] for xi in np.matrix(P_1)])
```

nie machen!

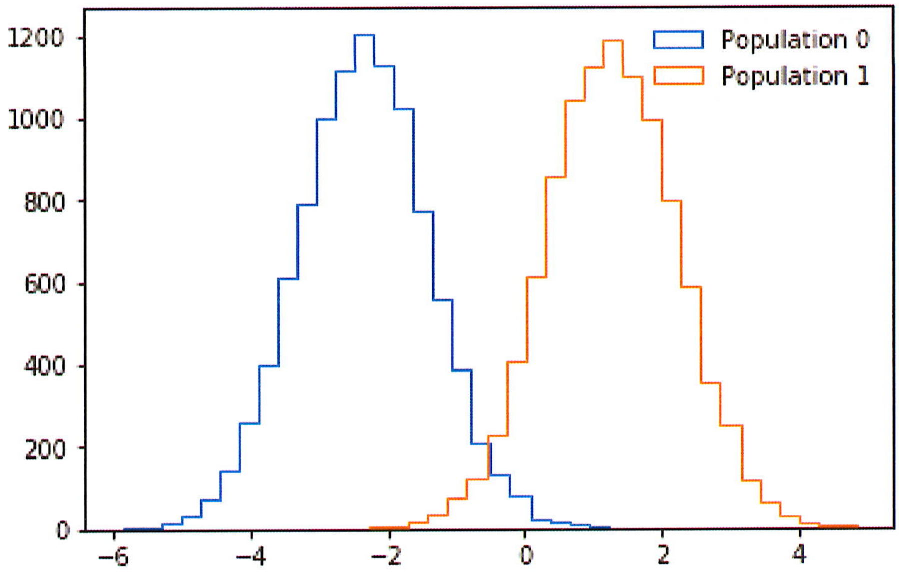
```
In [16]: plt.hist(projection_0, histtype = 'step', label = 'Population 0', bins = 25)
         plt.hist(projection_1, histtype = 'step', label = 'Population 1', bins = 25)
         plt.legend()

Out[16]: <matplotlib.legend.Legend at 0x2291c858e10>
```

Nie über numpy arrays iterieren, wenn nicht absolut nötig!

Hier funktioniert bspw.:

projection_0 = np.squeeze(np.array(np.dot(normed_lam.T, P_0.T)))

Da kommt das selbe raus, ist aber viel schneller. (meistens)



e) 2/2P

```
In [17]: def precision(signal, noise, cut):
             true_pos  = np.array([len(signal[signal < cut]) for cut in cut])
             false_pos = np.array([len(noise[noise < cut]) for cut in cut])
             return true_pos / (true_pos + false_pos)
```

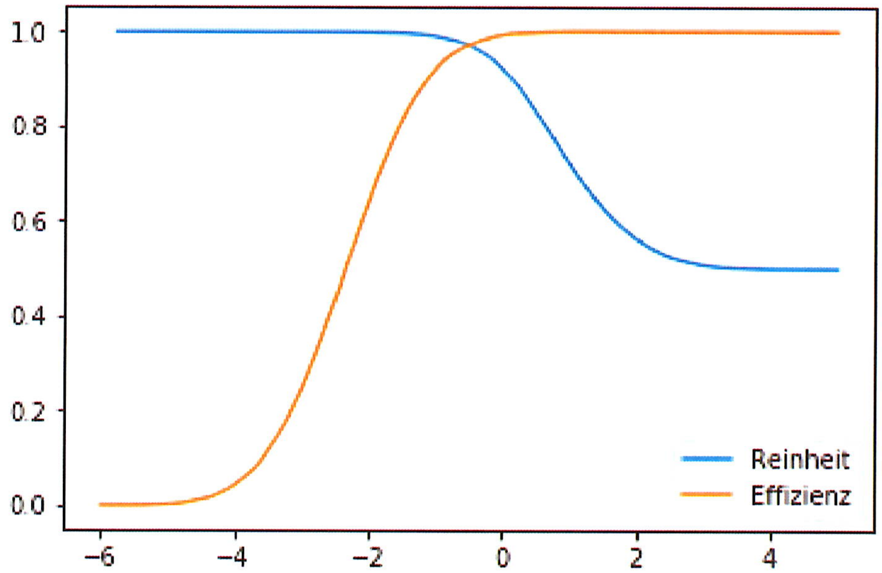
✓

```
In [18]: def recall(signal, noise, cut):
             true_pos  = np.array([len(signal[signal < cut]) for cut in cut])
             false_neg = np.array([len(signal[signal > cut]) for cut in cut])
             return true_pos / (true_pos + false_neg)
```

✓

Sehr schön

```
In [19]: signal = projection_0
         noise = projection_1

         lam_cut = np.linspace(-6, 5, 100)
         plt.plot(lam_cut, precision(signal, noise, lam_cut),
                  label = 'Reinheit')
         plt.plot(lam_cut, recall(signal, noise, lam_cut),
                  label = 'Effizienz')
         plt.legend()

C:\Users\lejon\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: RuntimeWarning: invalid va
  after removing the cwd from sys.path.


Out[19]: <matplotlib.legend.Legend at 0x2291c8ba4a8>
```



✓ Stellt man normalerweise aber andersrum dar. Immer Signal rechts vom BG. (Konvention)

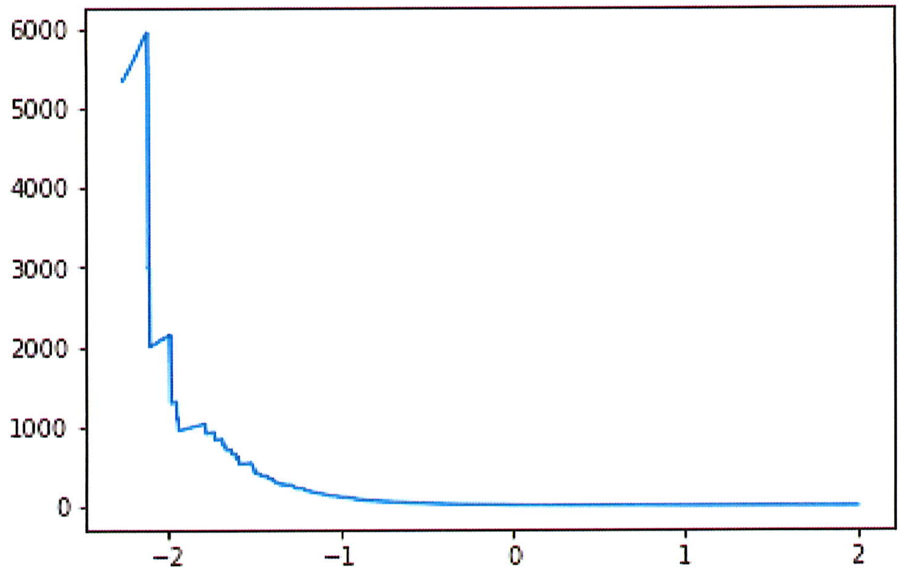
f) Untersuche Signal-Untergrundverhältnis: 0.5/1 P

In [20]:
```
def signal_noise_ratio(signal, noise, cut):
    return np.array([len(signal[signal <= cut]) / len(noise[noise <= cut]) for cut in
```
✓

In [21]:
```
lam_cut = np.linspace(min(noise), 2, 1000)
plt.plot(lam_cut, signal_noise_ratio(signal, noise, lam_cut),
         label = 'Signal-Untergrundverhältnis')

lam_cut[np.argmax(signal_noise_ratio(signal, noise, lam_cut))]
```

Out[21]: -2.1166576445868293

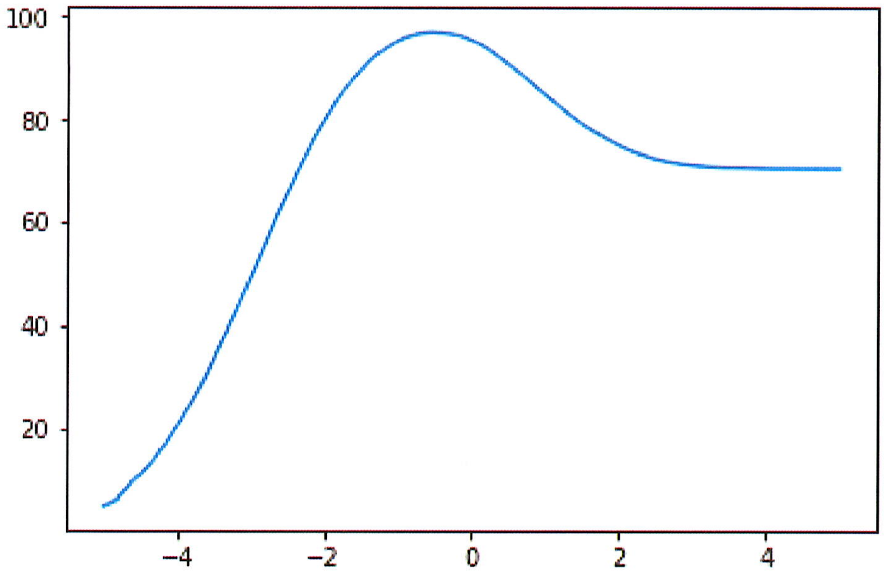
Für kleinere cut werte wird $\frac{S}{N} \to \infty$, weil $N=0$.



g) Untersuche Signifikanz: 7/7 P

```
In [22]: def sig(signal, noise, cut):
             return np.array([len(signal[signal <= cut]) /
                              np.sqrt(len(noise[noise <= cut]) + len(signal[signal <= cut])) f
```

```
In [23]: lam_cut = np.linspace(-5, 5, 1000)
         plt.plot(lam_cut, sig(signal, noise, lam_cut),
                  label = 'Signifikanz')

         lam_cut[np.argmax(sig(signal, noise, lam_cut))]

Out[23]: -0.5255255255255253
```



✓

b) 1.3.1 Nun alles nochmal mit der kleineren Population

Selbe Anmerkungen.
0.5/4 P Interpretation fehlt!

Lade Daten:

```
In [24]: import pandas as pd
         P_0 = pd.read_hdf('sample.hdf5', key='population0_1000')
         P_1 = pd.read_hdf('sample.hdf5', key='population1')
         P_0.head()
```

```
Out[24]:           x         y
         0 -1.718368  0.857438
         1 -0.811169  2.077295
         2  2.470046  5.312717
         3  1.116708  1.730386
         4 -0.639027  3.421546
```

a) Berechne Mittelwerte:

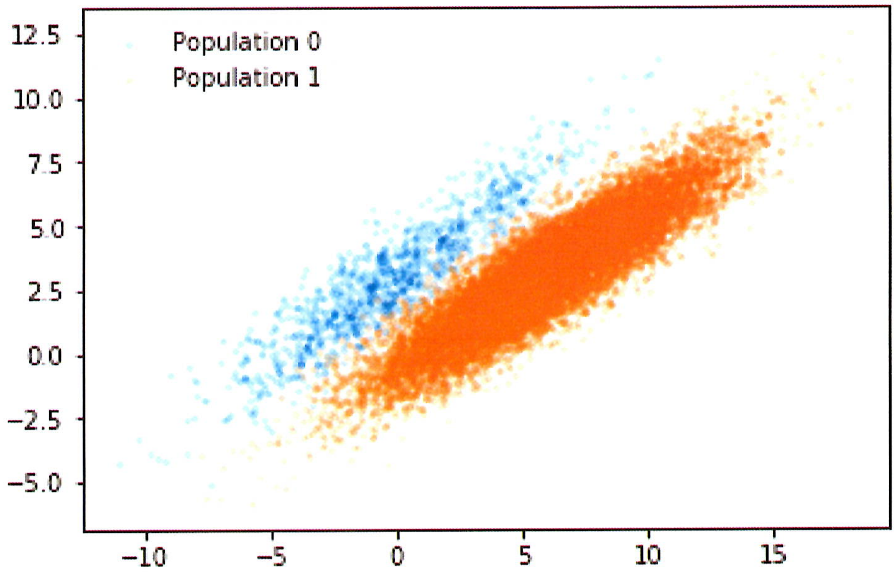
```
In [25]: mu0 = np.matrix([P_0.x.mean(), P_0.y.mean()]).T
         mu1 = np.matrix([P_1.x.mean(), P_1.y.mean()]).T
```

b) Berechne Kovarianzmatrizen:

```
In [26]: V_0 = P_0.cov()
         V_1 = P_0.cov()
```

```
In [27]: plt.scatter(P_0.x, P_0.y, s = 5, alpha = 0.2, label = 'Population 0')
         plt.scatter(P_1.x, P_1.y, s = 5, alpha = 0.2, label = 'Population 1')
         plt.legend()
```

```
Out[27]: <matplotlib.legend.Legend at 0x2291ca4bc50>
```



c) Konstruiere $\vec{\lambda}$:

```
In [28]: S_0 = np.sum([(xi.T - mu0) * (xi.T - mu0).T for xi in np.matrix(P_0)], axis = 0)
         S_1 = np.sum([(xi.T - mu1) * (xi.T - mu1).T for xi in np.matrix(P_1)], axis = 0)
         S_W = np.matrix(S_0 + S_1)
```

```
In [29]: lam = S_W.I * (mu1 - mu0)
         normed_lam = lam /  np.sqrt(lam[0]**2 + lam[1]**2)
         lam_array = np.array([lam[0], lam[1]])[:, 0]
         print(lam_array[1] / lam_array[0])
```

```
[-1.32435211]
```

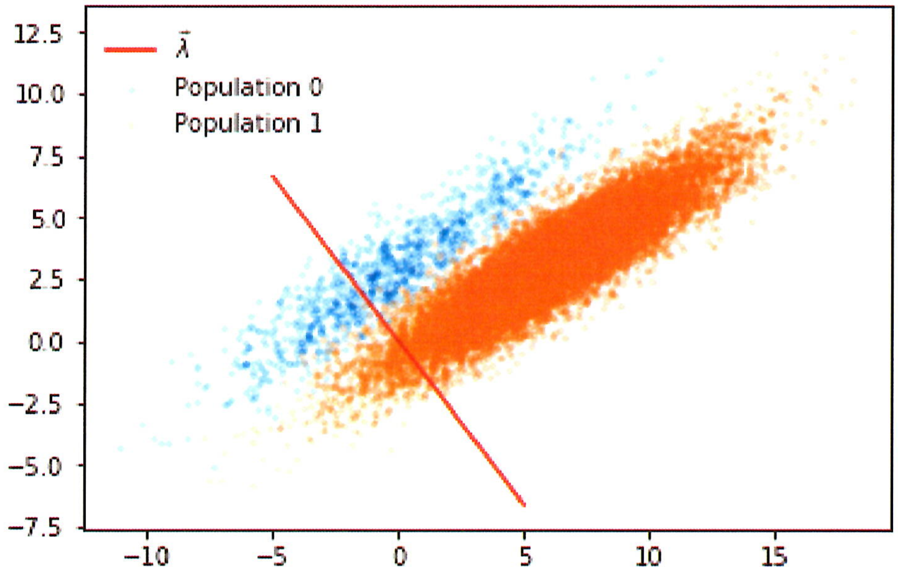
Die Geradengleichung lautet:

$$y(x) \approx -1.329 \cdot x$$

```
In [30]: xplot = np.linspace(-5, 5, 100)
         plt.plot(xplot, lam_array[1] / lam_array[0] * xplot,
                  color = 'red', label = r'$\vec{\lambda}$')
```

```
plt.scatter(P_0.x, P_0.y, s = 5, alpha = 0.2, label = 'Population 0')
plt.scatter(P_1.x, P_1.y, s = 5, alpha = 0.2, label = 'Population 1')
plt.legend()
```

Out[30]: <matplotlib.legend.Legend at 0x2291c93b0f0>

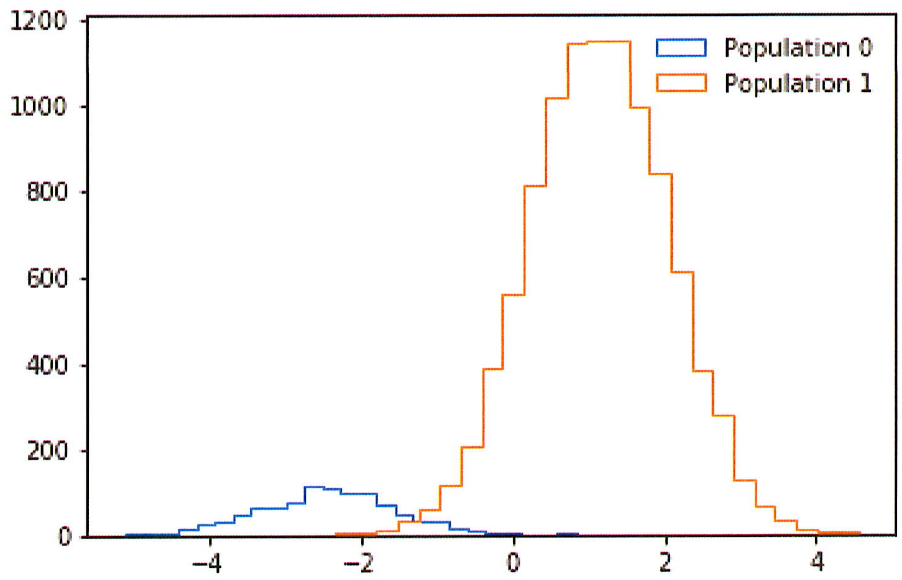


d) Stelle die Projektionen in einem Histogramm dar:

In [31]:
```
projection_0 = np.array([(xi * normed_lam)[0, 0] for xi in np.matrix(P_0)])
projection_1 = np.array([(xi * normed_lam)[0, 0] for xi in np.matrix(P_1)])
```

In [32]:
```
plt.hist(projection_0, histtype = 'step', label = 'Population 0', bins = 25)
plt.hist(projection_1, histtype = 'step', label = 'Population 1', bins = 25)
plt.legend()
```

Out[32]: <matplotlib.legend.Legend at 0x2291ca86b00>



e)

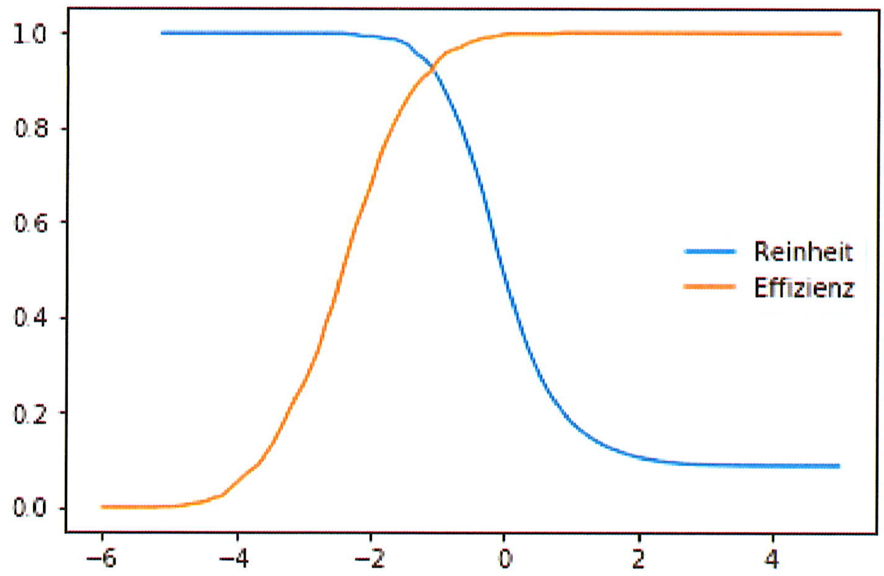
```
In [33]: def precision(signal, noise, cut):
             true_pos  = np.array([len(signal[signal < cut]) for cut in cut])
             false_pos = np.array([len(noise[noise < cut]) for cut in cut])
             return true_pos / (true_pos + false_pos)
```

```
In [34]: def recall(signal, noise, cut):
             true_pos  = np.array([len(signal[signal < cut]) for cut in cut])
             false_neg = np.array([len(signal[signal > cut]) for cut in cut])
             return true_pos / (true_pos + false_neg)
```

```
In [35]: signal = projection_0
         noise = projection_1

         lam_cut = np.linspace(-6, 5, 100)
         plt.plot(lam_cut, precision(signal, noise, lam_cut),
                  label = 'Reinheit')
         plt.plot(lam_cut, recall(signal, noise, lam_cut),
                  label = 'Effizienz')
         plt.legend()
```

```
C:\Users\lejon\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: RuntimeWarning: invalid va
  after removing the cwd from sys.path.
```

```
Out[35]: <matplotlib.legend.Legend at 0x2291cc4d940>
```

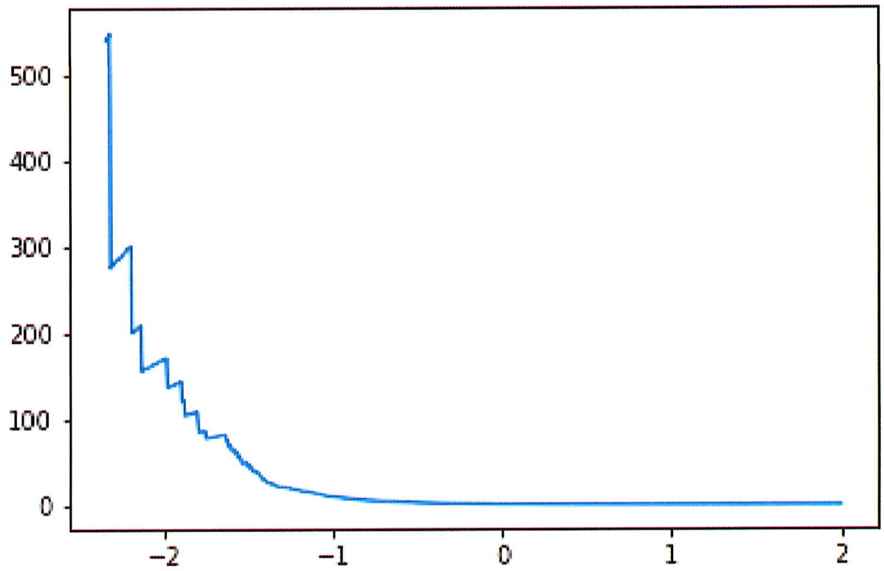


f) Untersuche Signal-Untergrundverhältnis:

```
In [36]: def signal_noise_ratio(signal, noise, cut):
             return np.array([len(signal[signal <= cut]) / len(noise[noise <= cut]) for cut in
```

```
In [37]: lam_cut = np.linspace(min(noise), 2, 1000)
         plt.plot(lam_cut, signal_noise_ratio(signal, noise, lam_cut),
                  label = 'Signal-Untergrundverhältnis')

         lam_cut[np.argmax(signal_noise_ratio(signal, noise, lam_cut))]
```

```
Out[37]: -2.3147673098893384
```

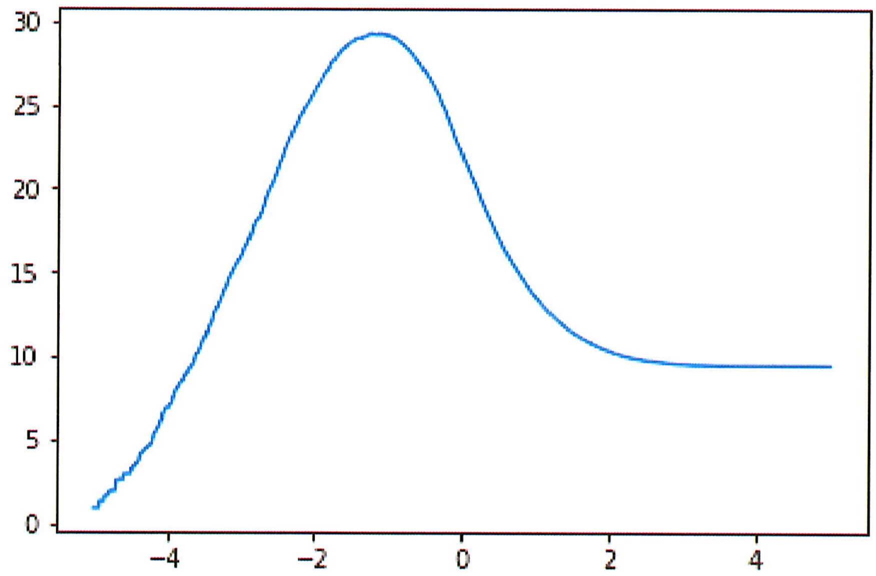


g) Untersuche Signifikanz:

```
In [38]: def sig(signal, noise, cut):
             return np.array([len(signal[signal <= cut]) /
                                  np.sqrt(len(noise[noise <= cut]) + len(signal[signal <= cut])) fo
```

```
In [39]: lam_cut = np.linspace(-5, 5, 1000)
         plt.plot(lam_cut, sig(signal, noise, lam_cut),
                  label = 'Signifikanz')

         lam_cut[np.argmax(sig(signal, noise, lam_cut))]
```

```
Out[39]: -1.2562562562562563
```


30
25
20
15
10
5
0
−4
−2
0
2
4

Aufgabe 10 a)

$$\gamma = \frac{1}{2(1-\rho^2)}$$
$$\lambda = \frac{1}{\sqrt{2\pi}\,\sigma_Y\sqrt{1-\rho^2}}$$

$$Var(Y|X) = E(Y^2|X) - E(Y|X)^2$$

$$E(Y|X) = (\rho \frac{\sigma_Y}{\sigma_X}(x-\mu_x) - \mu_y) \quad \text{bekannt}$$

$$E(Y^2|X) = \int_{-\infty}^{\infty} y^2 f(y|x)\,dy \qquad u_y = \frac{y-\bar{y}}{\sigma_y} \qquad dy = du_y\,\sigma_y$$

$$= \lambda \int_{-\infty}^{\infty} (u_y\sigma_y + \mu_y)^2 \exp(-\gamma[u_y - \rho u_x]^2)\,\sigma_y\,du_y$$

$$= \lambda\sigma_y \int_{-\infty}^{\infty} (u_y^2\sigma_y^2 + 2u_y\sigma_y\mu_y + \mu_y^2) \exp(-\gamma[u_y-\rho u_x]^2)\,du_y$$

$$= \lambda\sigma_y \Big\{\sigma_y^2\int_{-\infty}^{\infty} u_y^2 \exp(-\gamma[u_y-\rho u_x]^2)\,du_y$$
$$- 2\sigma_y\mu_y \int_{-\infty}^{\infty} u_y \exp(-\gamma[u_y-\rho u_x]^2)\,du_y\Big\}$$
$$+ \mu_y^2$$

$$u_y - \rho u_x = k$$

$$= \lambda\sigma_y^3 \int_{-\infty}^{\infty} (k^2 + 2\rho u_x k + \rho^2 u_x^2)\exp(-\gamma k^2)\,dk$$
$$- 2\lambda\sigma_y^2\mu_y\rho u_x \int_{-\infty}^{\infty} \exp(-\gamma k^2)\,dk + \mu_y^2$$

$$= \lambda\sigma_y^3 \Big\{\int_{-\infty}^{\infty} k^2\exp(-\gamma k^2)\,dk + \rho^2u_x^2\int_{-\infty}^{\infty}\exp(-\gamma k^2)\,dk\Big\}$$
$$- 2\lambda\sigma_y^2\mu_y\rho u_x\sqrt{\frac{\pi}{\gamma}} + \mu_y^2$$

$$= \lambda\sigma_y^3\Big\{\sqrt{\frac{\pi}{\gamma}}\cdot\frac{1}{2\gamma} + \rho^2u_x^2\sqrt{\frac{\pi}{\gamma}}\Big\} - 2\lambda\sigma_y^2\mu_y\rho u_x\sqrt{\frac{\pi}{\gamma}} + \mu_y^2$$

$$= \frac{\sigma_y^2}{\cancel{\sqrt{2\pi}\sqrt{1-\rho^2}}}\left(\cancel{\sqrt{\pi}\sqrt{2(1-\rho^2)}}(1-\rho^2) + \rho^2u_x^2\cancel{\sqrt{\pi}\sqrt{2(1-\rho^2)}}\right)$$
$$- 2\frac{\cancel{\sqrt{2}}\sigma_y\mu_y\rho u_x\cancel{\sqrt{\pi}}\cancel{\sqrt{1-\rho^2}}}{\cancel{\sqrt{2\pi}\sqrt{1-\rho^2}}} + \mu_y^2$$

$$= \sigma_y^2\left((1-\rho^2) + \rho^2u_x^2\right) - 2\sigma_y\mu_y\rho u_x + \mu_y^2$$

$\to \mathrm{Var}(Y|X)$

$$= \sigma_y^2\left((1-\rho^2) + \rho^2\left(\frac{x-\mu_x}{\sigma_x}\right)^2\right) - 2\frac{\sigma_y}{\sigma_x}\rho(x-\mu_y)\mu_y + \mu_y^2$$

$$- \left(\rho\frac{\sigma_y}{\sigma_x}(x-\mu_x) - \mu_y\right)^2$$

$$= \sigma_y^2(1-\rho^2) + \sigma_y^2\rho^2\left(\frac{x-\mu_x}{\sigma_x}\right)^2 - 2\frac{\sigma_y}{\sigma_x}\rho(x-\mu_y)\mu_y + \mu_y^2$$

$$-\rho^2\frac{\sigma_y^2}{\sigma_x^2}(x-\mu_y)^2 + 2\rho\frac{\sigma_y}{\sigma_x}(x-\mu_y)\mu_y - \mu_y^2$$

$$= \cancel{\sigma_y^2(1-\sigma_x^2)} \quad \sigma_y^2(1-\rho^2)$$

$$\to \quad 1 = (1-\rho^2)\sigma_y^2$$

mit $E(y|x) = \rho\frac{\sigma_y}{\sigma_x}(x-\mu_x) + \mu_y \overset{!}{=} a + bx$

folgt $a = -0{,}5 = +\rho\frac{\sigma_y}{\sigma_x}\mu_x + \mu_y$

$$b = 0.6 = \rho\frac{\sigma_y}{\sigma_x} \quad \Leftrightarrow \quad \rho = b\frac{\sigma_x}{\sigma_y}$$

$$1 = \left(1 - b^2\frac{\sigma_x^2}{\sigma_y^2}\right)\sigma_y^2$$

$$= \sigma_y^2 - b^2\sigma_x^2 \quad \Leftrightarrow \quad \sigma_y = \sqrt{1+b^2\sigma_x^2}$$

$$\rho = b\frac{\sigma_x}{\sigma_y} = b\sigma_x \cdot \frac{1}{\sqrt{1+b^2\sigma_x^2}}$$

$$a = +b\frac{\sigma_x}{\sigma_y}\cdot\frac{\sigma_y}{\sigma_x}\mu_x + \mu_y = +b\mu_x + \mu_y$$

$$\Leftrightarrow \quad \mu_y = (+b\mu_x + a)$$

SMD Blatt 4

A11:

$$\text{Pop 0: } \begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}2\\1\end{pmatrix}\begin{pmatrix}1{,}5\\2\end{pmatrix}\begin{pmatrix}2\\2\end{pmatrix}\begin{pmatrix}2\\3\end{pmatrix}\begin{pmatrix}3\\3\end{pmatrix}$$

$$\vec{\mu}_0 = \begin{pmatrix}23/12\\2\end{pmatrix} \qquad S_W = \sum_j S_j \quad \text{mit} \quad S_j = \sum_i (\vec{x}_i - \vec{\mu}_j)(\vec{x}_i - \vec{\mu}_j)^T$$

$$S_0 = \sum_i (\vec{x}_i - \vec{\mu}_0)(\vec{x}_i - \vec{\mu}_0)^T$$

$$= \begin{pmatrix}-11/12\\-1\end{pmatrix}(-11/12, -1) + \begin{pmatrix}1/12\\-1\end{pmatrix}(1/12, -1) + \begin{pmatrix}-5/12\\0\end{pmatrix}(-5/12, 0)$$

$$+ \begin{pmatrix}1/12\\0\end{pmatrix}(1/12, 0) + \begin{pmatrix}1/12\\1\end{pmatrix}(1/12, 1) + \begin{pmatrix}13/12\\1\end{pmatrix}(13/12, 1)$$

$$= \begin{pmatrix}121/144 & 11/12\\11/12 & 1\end{pmatrix} + \begin{pmatrix}1/144 & -1/12\\-1/12 & 1\end{pmatrix} + \begin{pmatrix}25/144 & 0\\0 & 0\end{pmatrix}$$

$$+ \begin{pmatrix}1/144 & 0\\0 & 0\end{pmatrix} + \begin{pmatrix}1/144 & 1/12\\1/12 & 1\end{pmatrix} + \begin{pmatrix}169/144 & 13/12\\13/12 & 1\end{pmatrix}$$

$$= \begin{pmatrix}53/24 & 2\\2 & 4\end{pmatrix}$$

$$\text{Pop 1: } \begin{pmatrix}1{,}5\\1\end{pmatrix}\begin{pmatrix}2{,}5\\1\end{pmatrix}\begin{pmatrix}3{,}5\\1\end{pmatrix}\begin{pmatrix}2{,}5\\2\end{pmatrix}\begin{pmatrix}3{,}5\\2\end{pmatrix}\begin{pmatrix}4{,}5\\2\end{pmatrix}$$

$$\vec{\mu}_1 = \begin{pmatrix}3\\3/2\end{pmatrix}$$

$$S_1 = \begin{pmatrix}-3/2\\-1/2\end{pmatrix}(-3/2, -1/2) + \begin{pmatrix}-1/2\\-1/2\end{pmatrix}(-1/2, -1/2) + \begin{pmatrix}1/2\\-1/2\end{pmatrix}(1/2, -1/2)$$

$$+ \begin{pmatrix}-1/2\\1/2\end{pmatrix}(-1/2, 1/2) + \begin{pmatrix}1/2\\1/2\end{pmatrix}(1/2, 1/2) + \begin{pmatrix}3/2\\1/2\end{pmatrix}(3/2, 1/2)$$

$$= \begin{pmatrix}9/4 & 3/4\\3/4 & 1/4\end{pmatrix} + \begin{pmatrix}1/4 & 1/4\\1/4 & 1/4\end{pmatrix} + \begin{pmatrix}1/4 & -1/4\\-1/4 & 1/4\end{pmatrix}$$

$$+ \begin{pmatrix}1/4 & -1/4\\-1/4 & 1/4\end{pmatrix} + \begin{pmatrix}1/4 & 1/4\\1/4 & 1/4\end{pmatrix} + \begin{pmatrix}9/4 & 3/4\\3/4 & 1/4\end{pmatrix}$$

$$= \begin{pmatrix}22/4 & 6/4\\6/4 & 6/4\end{pmatrix} = \begin{pmatrix}11/2 & 3/2\\3/2 & 3/2\end{pmatrix}$$

$$\Rightarrow \quad S_W = S_0 + S_1 = \begin{pmatrix}185/24 & 7/2\\7/2 & 11/2\end{pmatrix}$$

$$S_B = (\vec{\mu}_1 - \vec{\mu}_2)(\mu_1 - \mu_2)^T$$

$$= \begin{pmatrix} -13/12 \\ 1/2 \end{pmatrix} \left(-13/12, \; 1/2\right) = \begin{pmatrix} 169/144 & -13/24 \\ -13/24 & 1/4 \end{pmatrix}$$

b) $\vec{\lambda}^* = S_W^{-1} (\vec{\mu}_1 - \vec{\mu}_2)$

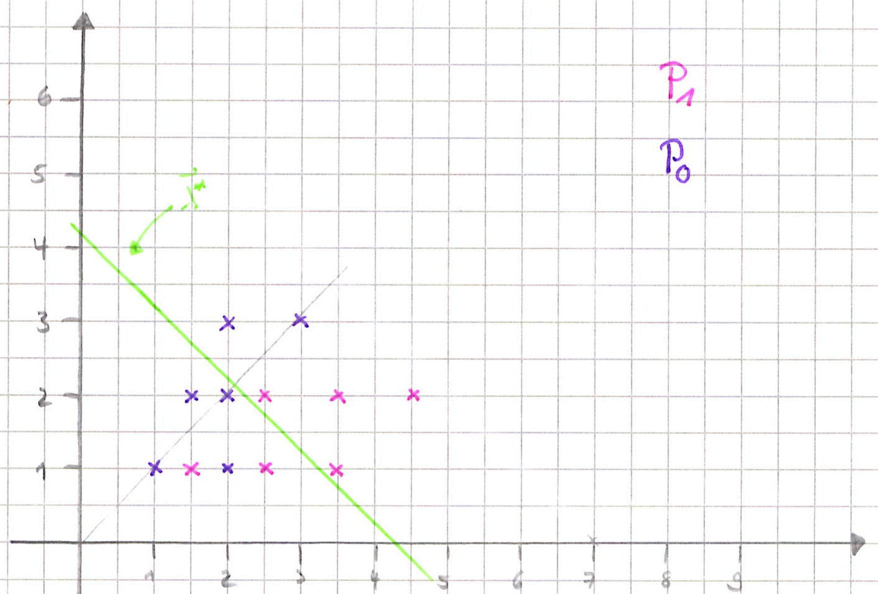
$$S_W^{-1} = \frac{1}{\det(S_W)} \begin{pmatrix} 11/2 & -7/2 \\ -7/2 & 185/24 \end{pmatrix}$$

$$\det(S_W) = 1447/48$$

$$S_W^{-1} = \begin{pmatrix} 264/1447 & \cancel{84/1741} \; -168/1447 \\ -168/1447 & 370/1447 \end{pmatrix}$$

$$\vec{\lambda}^* = S_W^{-1} (\vec{\mu}_1 - \vec{\mu}_2)$$

$$= S_W^{-1} \begin{pmatrix} -13/12 \\ 1/2 \end{pmatrix} = \begin{pmatrix} -370/1447 \\ 367/1447 \end{pmatrix}$$



$$|S_W^{-1} S_B - D| = 0 \Rightarrow EW = 0; \; 0{,}403824$$

$$\vec{v}_1 = \begin{pmatrix} 0{,}70997 \\ 0{,}419058 \end{pmatrix} \quad \vec{v}_2 = \begin{pmatrix} -0{,}7042263 \\ 0{,}907959 \end{pmatrix}$$

$(\vec{\lambda}^*)^2 \approx 0{,}064853 \approx 0{,}065$

Pop 0: $\vec{\lambda}^* \vec{x}_1 = \left(-370/1447 \cdot 1 + 367/1447 \cdot 1\right)$

$= -3/1447 \approx -0{,}002 \to p_1 \approx -0{,}03$

$\vec{\lambda}^* \vec{x}_2 = -373/1447 \approx -0{,}258 \to p_2 \approx -3{,}97$

$\vec{\lambda}^* x_3 = 179/1447 \approx 0{,}124 \to p_3 \approx +1{,}91$

$\vec{\lambda}^* x_4 = -6/1447 \approx -0{,}004 \to p_4 \approx -0{,}06$

$\vec{\lambda}^* x_5 = 361/1447 \approx 0{,}249 \to p_5 \approx +3{,}83$

$\vec{\lambda}^* x_6 = -9/1447 \approx -0{,}006 \to p_6 \approx -0{,}09$

Pop 1:

$\vec{\lambda}^* \vec{x}_1 = -188/1447 \approx -0{,}13 \to p_1 \approx \cancel{18{,}84}\ -2$

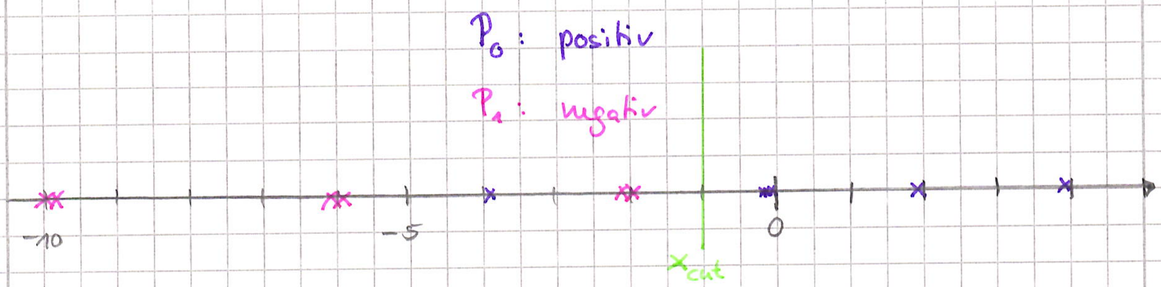
$\vec{\lambda}^* \vec{x}_2 = -558/1447 \approx -0{,}386 \to p_2 \approx -5{,}94$

$\vec{\lambda}^* \vec{x}_3 = -928/1447 \approx -0{,}641 \to p_3 \approx -9{,}861$

$\vec{\lambda}^* \vec{x}_4 = -191/1447 \approx -0{,}132 \to p_4 \approx -2{,}03$

$\vec{\lambda}^* \vec{x}_5 = -561/1447 \approx -0{,}388 \to p_5 \approx -5{,}97$

$\vec{\lambda}^* \vec{x}_6 = -931/1447 \approx -0{,}643 \to p_6 \approx -9{,}89$



Reinheit: $\dfrac{tp}{tp+fp}$ Effizienz $= \dfrac{tp}{tp+fn}$

$\to$ minimiere fp und fn $\to$ mit dem x_{cut} $fp = 0$, $fn = 1$

$\to$ Reinheit $= \dfrac{5}{5+0} = 1$ Effizienz $= \dfrac{5}{5+1} = \dfrac{5}{6} \approx 0{,}8\overline{3}$