1. $L(a(x), y) = \lambda_y [a(x) \neq y]$.

$Y = \{-1; +1\}$

$p(x|y=-1) \sim Exp(1)$

$p(x|y=+1) \sim N(0,1)$

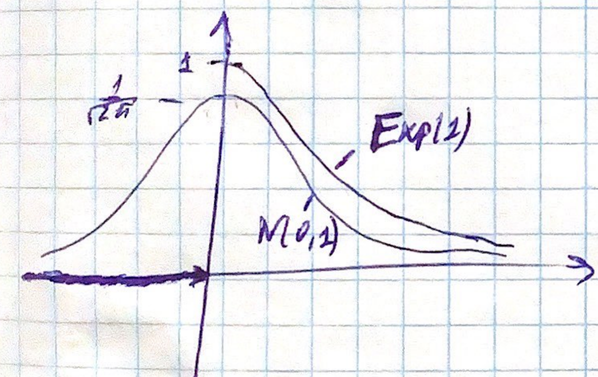
$p(y=-1) = 0.25$ / $p(y=+1) = 1 - p(y=-1) = 0.75$

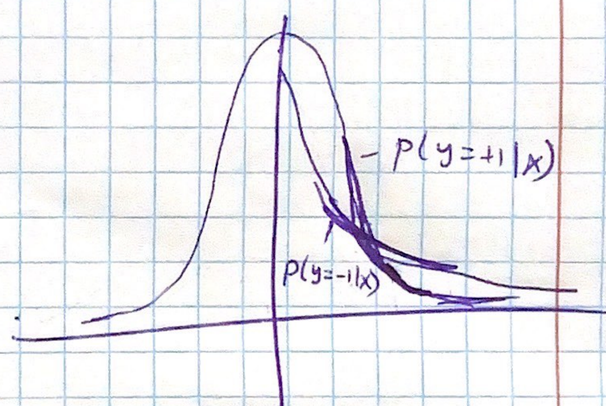
$\lambda_+ = 2$

$\lambda_- = 2$

$a(x), R(a) - ?$

$$a(x) = \begin{cases} -1, & \text{если } \frac{p(y=-1)}{p(y=+1)}\sqrt{2\pi}\exp\{\frac{x^2}{2} - x\} \geq \frac{\lambda_+}{\lambda_-},\ x \geq 0 \\ 1, & \text{иначе} \end{cases} \Leftrightarrow$$





$$\Leftrightarrow \begin{cases} -1, & x \in [1 + \sqrt{1 - 2\ln(\frac{\sqrt{2\pi}}{3})}, +\infty) \\ +1, & \text{иначе} \end{cases}$$

T1

$$\ln\left(\frac{\sqrt{2\pi}}{3}\cdot\exp\left\{\frac{x^2}{2}-x\right\}\right)\geq \ln(1).$$

$$\ln\left(\frac{\sqrt{2\pi}}{3}\right)+\frac{x^2}{2}-x\geq \ln(1)$$

$$\frac{1}{2}x^2-x+\ln\left(\frac{\sqrt{2\pi}}{3}\right)\geq 0.$$

$$x=\frac{1\pm\sqrt{1-2\ln\left(\frac{\sqrt{2\pi}}{3}\right)}}{1.} \rightarrow$$

$$x_0 = 2,165911$$

$$x_1 = -0,16591.$$

T1

$$R(a) = \iint L(a(x),y)\, p(x,y)\, dx\, dy =$$

$$= \int_X \sum_Y \lambda_y [a(x) \neq y]\, p(x|y)\, P(y)\, dx =$$

$$= \int_X \sum_Y (1 - \lambda_y [a(x) = y])\, p(x|y)\, P(y)\, dx =$$

$$= \underbrace{\int_X \sum_Y p(x|y) P(y)\, dx}_{= 1} - \int_X \sum_Y \lambda_y [a(x) = y]\, p(x|y)\, P(y)\, dx \boxed{=}$$

$\Rightarrow$ ~~т.к.~~: $a(x) = \arg\min_a R(a) =$

$$= \arg\max_a \int_X \sum_Y \lambda_y [a(x) = y]\, p(x|y)\, P(y)\, dx =$$

$$= \arg\max_y \lambda_y\, p(x|y)\, P(y).$$

$$a(x) = \mathrm{sign}\left(\frac{P(y=+1|x)}{P(y=-1|x)} - \frac{\lambda_-}{\lambda_+} \right) =$$

$$= -\mathrm{sign}\left(\frac{P(y=-1|x)}{P(y=+1|x)} - \frac{\lambda_+}{\lambda_-} \right)$$

$$\frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{x^2}{2} \right\} \sim N(0,1).$$

$$\begin{cases} 0, & x < 0 \\ e^{-x}, & x \geq 0 \end{cases} \sim Exp(1).$$

T1

R(a):

$\boxed{1}$

$$1 - \int_X \max_y \lambda_y p(x|y) p(y) dx =$$

$$= \int_X \min_y \lambda_y p(x|y) p(y) dx =$$

$x_0 := 1 + \sqrt{1 - \ln\left(\frac{\sqrt{2\pi}}{3}\right)}$

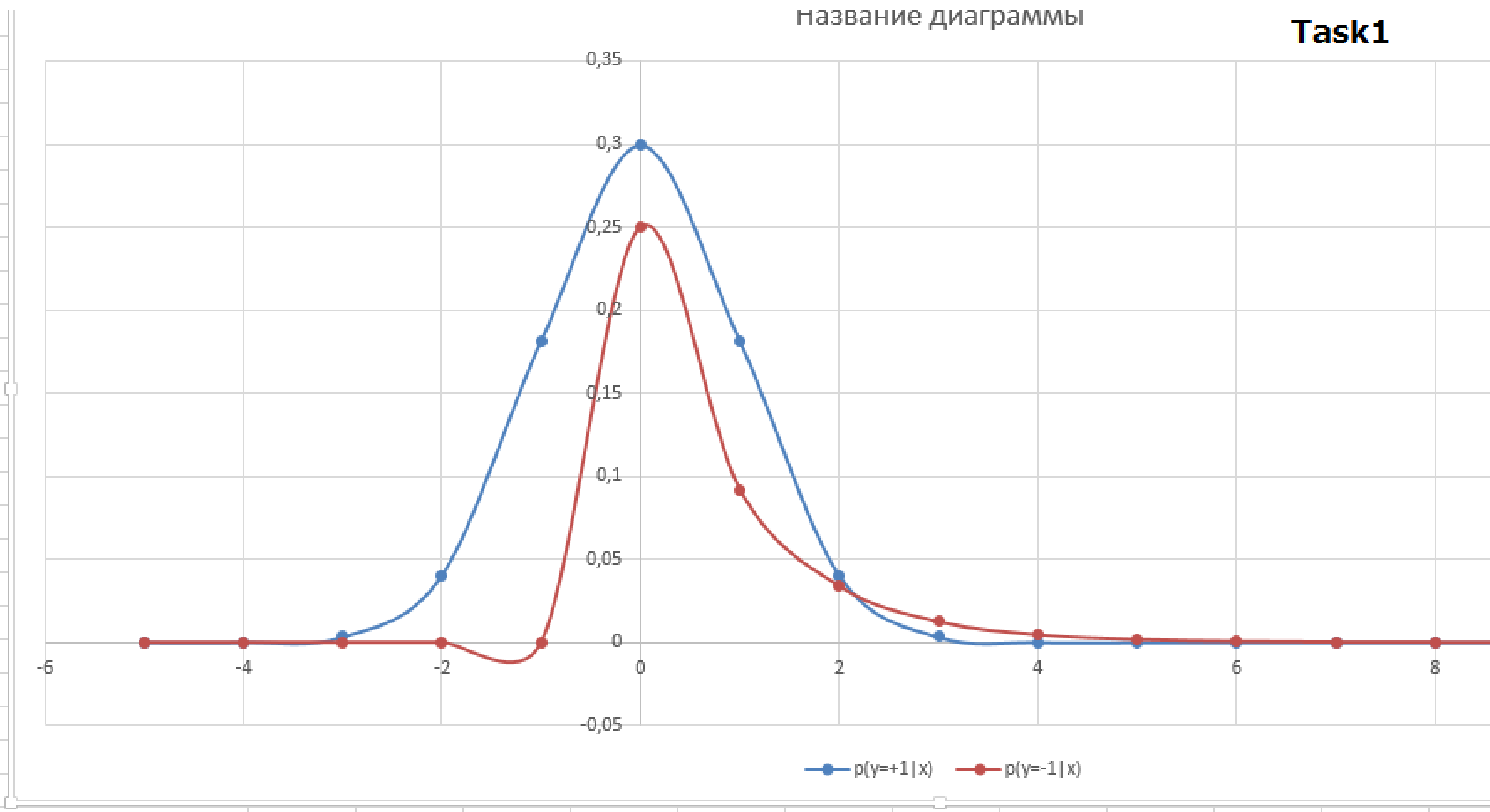
$$= \int_{0}^{x_0} e^{-x} \cdot 2 \cdot \frac{1}{4} dx + \int_{1+\sqrt{1-\ln\left(\frac{\sqrt{2\pi}}{3}\right)}}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \cdot 2 \cdot \frac{3}{4} dx$$

$$+ \int_{-\infty}^{0} 0 \, dx =$$

$$= \frac{1}{2} \left(\int_0^{x_0} e^{-x} dx + \frac{3}{\sqrt{2\pi}} \int_{x_0}^{+\infty} e^{-\frac{x^2}{2}} dx \right)$$

-8	3,7892E-15	0
-7	6,85104E-12	0
-6	4,55691E-09	0
-5	1,11504E-06	0
-4	0,000100373	0
-3	0,003323886	0
-2	0,040493225	0
-1	0,181478043	0
0	0,29920671	0,25
1	0,181478043	0,09196986
2	0,040493225	0,033833821
3	0,003323886	0,012446767
4	0,000100373	0,00457891
5	1,11504E-06	0,001684487
6	4,55691E-09	0,000619688
7	6,85104E-12	0,00022797
8	3,7892E-15	8,38657E-05
9	7,70983E-19	3,08525E-05
10	5,77095E-23	1,135E-05
11	1,58911E-27	4,17543E-06
12	1,60979E-32	1,53605E-06
13	5,99912E-38	5,65082E-07
14	8,22455E-44	2,07882E-07
15	4,14803E-50	7,64756E-08
16	7,69622E-57	2,81338E-08

Task1



Task1: `p(x| y=-1) ~ Exp(1), p(x| y=+1) ~ N(0,1) \lambda_{-} = 2, \lambda_{+} = 2 p(y=-1) = 0.25`

```
x0=1+math.sqrt(1-2*math.log(math.sqrt(2*math.pi)/3))
x0
```

```
2.165910593024557
```

```
from math import cos, exp, pi
from scipy.integrate import quad

def f1(x):
    return exp(-0.5 * x * x)

def f2(x):
    return exp(-x)

res1, err1 = quad(f1, x0, math.inf)
res2, err2 = quad(f2, 0, x0)

print("The numerical result is {:f} (+-{:g})"
    .format(res1, err1))
print("The numerical result is {:f} (+-{:g})"
    .format(res2, err2))

const1 = 3/math.sqrt(2*math.pi)
result = (res1 + const1*res2)/2
print('Result: ',result)
```

```
The numerical result is 0.037998 (+-2.2012e-09)
The numerical result is 0.885355 (+-9.82941e-15)
Result:  0.54880702111819
```

*Вычисляем интеграл для R(a)

Task2:

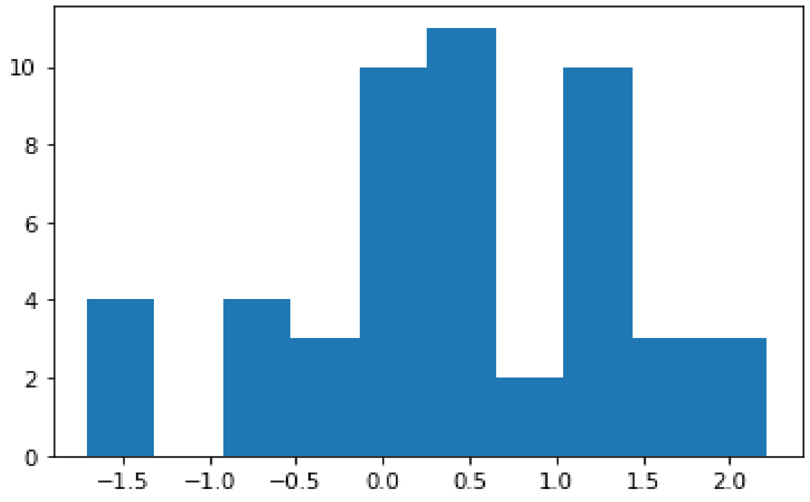
```
from numpy import mean
from numpy import std
import pandas as pd
import numpy as np

data = pd.read_csv('task2.csv')
print(data.shape)
data.describe().transpose()
```

```
(50, 3)
```

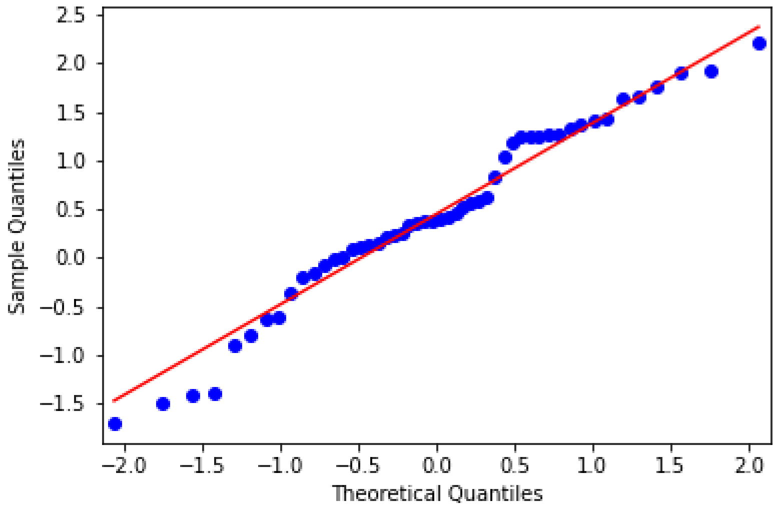
	count	mean	std	min	25%	50%	75%	max
x_1	50.0	0.448679	0.943112	-1.704860	-0.013628	0.395468	1.250642	2.220427
x_2	50.0	0.885331	1.836185	-2.783404	-0.258457	0.879242	2.028090	5.439624
target	50.0	0.000000	1.010153	-1.000000	-1.000000	0.000000	1.000000	1.000000

T2

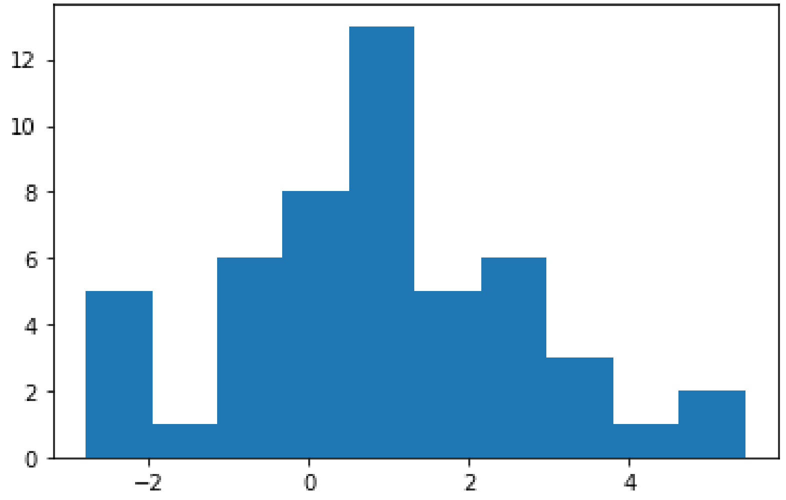
```python
from matplotlib import pyplot
pyplot.hist(data['x_1'])
pyplot.show()
```



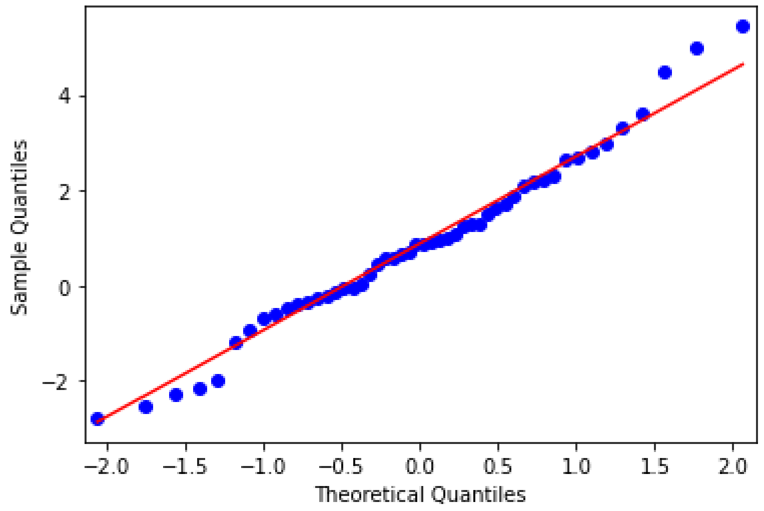

```python
from statsmodels.graphics.gofplots import qqplot
qqplot(data['x_1'], line='s')
pyplot.show()
```




```python
from matplotlib import pyplot
pyplot.hist(data['x_2'])
pyplot.show()
```




```python
from statsmodels.graphics.gofplots import qqplot
qqplot(data['x_2'], line='s')
pyplot.show()
```



T2

```
[ ] from scipy.stats import shapiro

    # normality test
    stat1, p1 = shapiro(data['x_1'])
    stat2, p2 = shapiro(data['x_2'])
    print('Statistics for x_1=%.3f, p=%.3f' % (stat1, p1))
    print('Statistics for x_2=%.3f, p=%.3f' % (stat2, p2))
    # interpret
    alpha = 0.05
    if p1 > alpha:
      print('x_1: Sample looks Gaussian (fail to reject H0)')
    else:
      print('x_1: Sample does not look Gaussian (reject H0)')

    if p2 > alpha:
      print('x_2: Sample looks Gaussian (fail to reject H0)')
    else:
      print('x_2: Sample does not look Gaussian (reject H0)')
```

```
Statistics for x_1=0.968, p=0.196
Statistics for x_2=0.983, p=0.695
x_1: Sample looks Gaussian (fail to reject H0)
x_2: Sample looks Gaussian (fail to reject H0)
```

```
[ ] from scipy.stats import normaltest
    # normality test
    stat1, p1 = normaltest(data['x_1'])
    stat2, p2 = normaltest(data['x_2'])
    print('Statistics for x_1=%.3f, p=%.3f' % (stat1, p1))
    print('Statistics for x_2=%.3f, p=%.3f' % (stat2, p2))
    # interpret
    alpha = 0.05
    if p1 > alpha:
      print('x_1: Sample looks Gaussian (fail to reject H0)')
    else:
      print('x_1: Sample does not look Gaussian (reject H0)')

    if p2 > alpha:
      print('x_2: Sample looks Gaussian (fail to reject H0)')
    else:
      print('x_2: Sample does not look Gaussian (reject H0)')
```

```
Statistics for x_1=1.295, p=0.523
Statistics for x_2=0.888, p=0.641
x_1: Sample looks Gaussian (fail to reject H0)
x_2: Sample looks Gaussian (fail to reject H0)
```

T2

```
from scipy.stats import anderson

# normality test
result1 = anderson(data['x_1'])
result2 = anderson(data['x_2'])
print('Statistic for x_1: %.3f' % result1.statistic)
print('Statistic for x_2: %.3f' % result2.statistic)
p = 0
for i in range(len(result1.critical_values)):
  sl, cv = result1.significance_level[i], result1.critical_values[i]
  if result1.statistic < result1.critical_values[i]:
    print('%.3f: %.3f, data(x_1) looks normal (fail to reject H0)' % (sl, cv))
  else:
    print('%.3f: %.3f, data(x_1) does not look normal (reject H0)' % (sl, cv))

for i in range(len(result2.critical_values)):
  sl, cv = result2.significance_level[i], result2.critical_values[i]
  if result2.statistic < result2.critical_values[i]:
    print('%.3f: %.3f, data(x_2) looks normal (fail to reject H0)' % (sl, cv))
  else:
    print('%.3f: %.3f, data(x_2) does not look normal (reject H0)' % (sl, cv))
```

```
Statistic for x_1: 0.522
Statistic for x_2: 0.243
15.000: 0.538, data(x_1) looks normal (fail to reject H0)
10.000: 0.613, data(x_1) looks normal (fail to reject H0)
5.000: 0.736, data(x_1) looks normal (fail to reject H0)
2.500: 0.858, data(x_1) looks normal (fail to reject H0)
1.000: 1.021, data(x_1) looks normal (fail to reject H0)
15.000: 0.538, data(x_2) looks normal (fail to reject H0)
10.000: 0.613, data(x_2) looks normal (fail to reject H0)
5.000: 0.736, data(x_2) looks normal (fail to reject H0)
2.500: 0.858, data(x_2) looks normal (fail to reject H0)
1.000: 1.021, data(x_2) looks normal (fail to reject H0)
```

T2

```
[ ] def get_p_of_y_equal_to_minus_one_and_plus_one(target):
      return [len(list(filter(lambda x: (x < 0), target)))/len(target),len(list(filter(lambda x: (x >= 0), target)))
                                                                                                        /len(target)]

    def get_all_mean_and_std_square(target_minus, target_plus):
      #calc mean
      mean_minus = np.sum(target_minus, axis=0)/len(target_minus) #result: mean of x_1, x_2, target
      mean_plus = np.sum(target_plus, axis=0)/len(target_plus)

      #calc std
      target_minus_transpose = target_minus.T
      for i in range(target_minus_transpose.shape[0]-1):
        target_minus_transpose[i]=(target_minus_transpose[i]-mean_minus[i])**2
      target_minus=target_minus_transpose.T
      std_square_minus = np.sum(target_minus, axis=0)/(len(target_minus)-1)

      target_plus_transpose = target_plus.T
      for i in range(target_plus_transpose.shape[0]-1):
        target_plus_transpose[i]=(target_plus_transpose[i]-mean_plus[i])**2
      target_plus=target_plus_transpose.T
      std_square_plus = np.sum(target_plus, axis=0)/(len(target_plus)-1)

      return [mean_minus[:-1],mean_plus[:-1]],[std_square_minus[:-1],std_square_plus[:-1]]
```

T2

```
[ ] import math
```

```
[ ] p1, p2 = get_p_of_y_equal_to_minus_one_and_plus_one(data['target'].values)
    print(p1, p2)
```

```
0.5 0.5
```

```
[ ] y_minus = data.values[:25]
    y_plus = data.values[25:]
    m, sigma_square = get_all_mean_and_std_square(y_minus, y_plus)
    print('means: ',m[0][0],m[0][1],' and ',m[1][0],m[1][1])
    print('std square: ',sigma_square[0][0],sigma_square[0][1],' and ',sigma_square[1][0],sigma_square[1][1])
    print('std square root ln : ',np.log(math.sqrt(sigma_square[0][0])),np.log(math.sqrt(sigma_square[0][1])),' and ',np.log(r
```

```
means:  0.17993739697070765 0.9654650474585901  and  0.7174210335926431 0.8051975503768627
std square:  1.1661549489331444 2.385011087123411  and  0.4993618181610397 4.48524294354885
std square root ln :  0.07685598420365 64 0.4346018865092804  and  -0.3472121797418795 0.7503963310295383
```

T2

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

$$0 = \ln\left(\frac{p(x|y=-1)\,p(t|y=-1)\cdot p(y=-1)}{p(x|y=+1)\,p(t|y=+1)\cdot p(y=+1)}\right)$$

$$= \ln\left(p(x|y=-1)\,p(t|y=-1)\,p(y=-1)\right) -$$

$$- \ln\left(p(x|y=+1)\,p(t|y=+1)\cdot p(y=+1)\right) =$$

$$\ln\left(p(x|y=-1)\right) + \ln\left(p(t|y=-1)\right) + \cancel{\ln\left(p(y=-1)\right)}^{\,1/2}$$

$$-\ln\left(p(x|y=+1)\right) - \ln\left(p(t|y=+1)\right) - \cancel{\ln\left(p(y=+1)\right)} \;\; {}^{"}\,1/2 \;\; (!!)$$

$$p(x|y=-1) \sim N(\mu_1, \sigma_1)$$

$$\ln\left(\frac{1}{\sqrt{2\pi}\,\sigma}\right) + \ln\left(e^{-\frac{(x-\mu)^2}{2\sigma^2}}\right) =$$

a

$$= \ln\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{(x-\mu)^2}{2\sigma^2}$$

$$\ln\left(\frac{x}{y}\right) \cancel{= \ln(x) - \ln}$$

$$\ln(y^{-1}) = -\ln(y)$$

$$-\ln(\sqrt{2\pi}) - \ln(\sigma)$$

T2

$$\Leftrightarrow \ln\left(\frac{1}{\sqrt{2\pi}\,\sigma_{x-}}\right) - \frac{(x-\mu_{x-})^2}{2\sigma_{x-}^2} +$$

$$\ln\left(\frac{1}{\sqrt{2\pi}\,\sigma_{t-}}\right) - \frac{(t-\mu_{t-})^2}{2\sigma_{t-}^2} -$$

$$\ln\left(\frac{1}{\sqrt{2\pi}\,\sigma_{x+}}\right) + \frac{(x-\mu_{x+})^2}{2\sigma_{x+}^2} -$$

$$\ln\left(\frac{1}{\sqrt{2\pi}\,\sigma_{t+}}\right) + \frac{(t-\mu_{t+})^2}{2\sigma_{t+}^2} \Leftrightarrow$$

$$+\ln(p(y=-1)) - \ln(p(y=+1)).$$

$$\Rightarrow -\ln(\sigma_{x-}) - \ln(\sigma_{t-}) + \ln(\sigma_{x+}) + \ln(\sigma_{t+})$$

$$\Leftrightarrow \frac{(x-\mu_{x+})^2}{2\sigma_{x+}^2} + \frac{(t-\mu_{t+})^2}{2\sigma_{t+}^2} - \frac{(x-\mu_{x-})^2}{2\sigma_{x-}^2} - \frac{(t-\mu_{t-})^2}{2\sigma_{t-}^2} +$$

$$+ \ln(\sigma_{x+}) + \ln(\sigma_{t+}) - \ln(\sigma_{x-}) - \ln(\sigma_{t-})$$

$$+ \ln(p(y=-1)) - \ln(p(y=+1)).$$