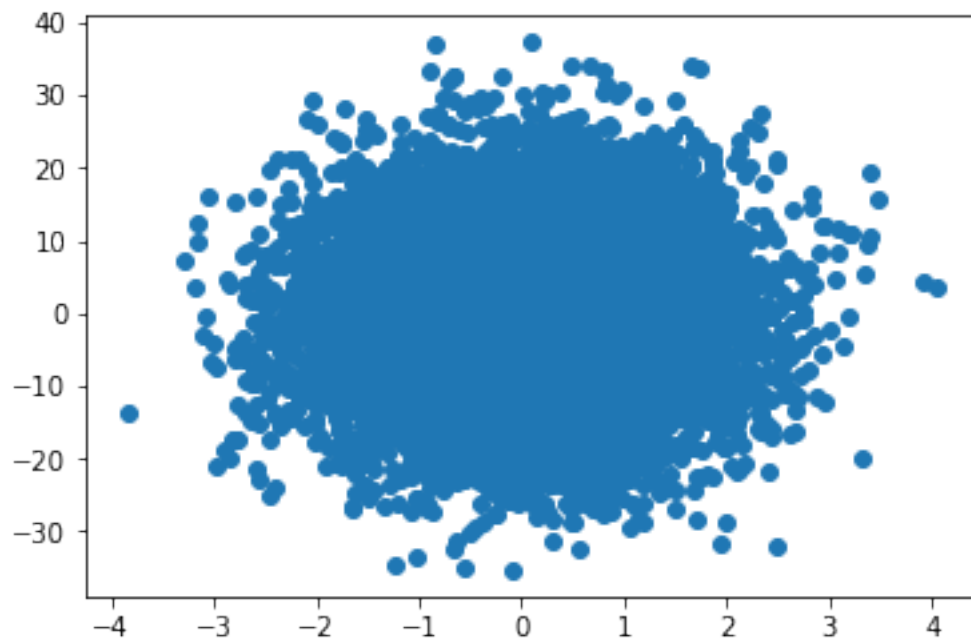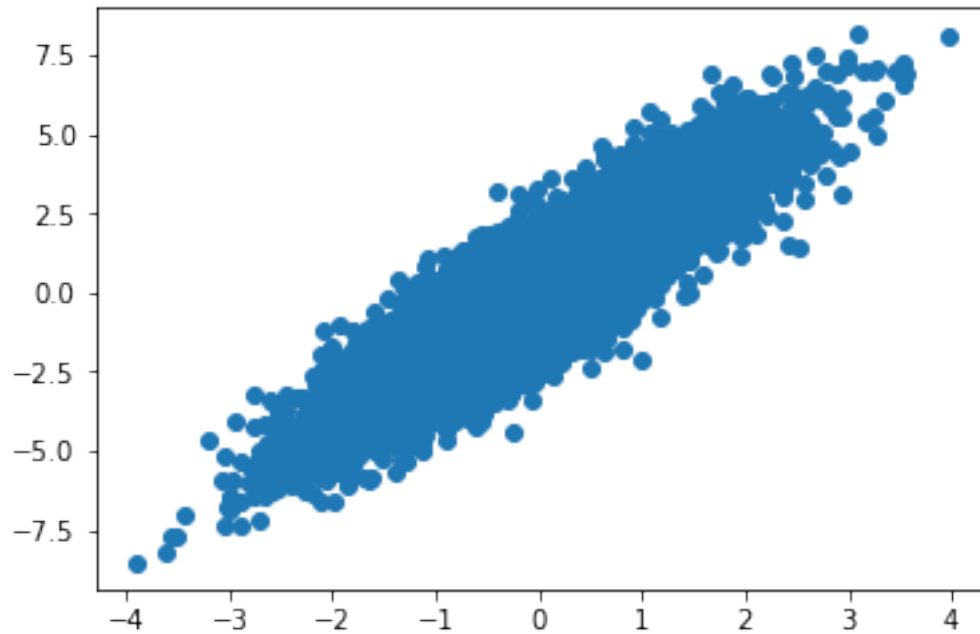# Rozwiazania

June 14, 2018

```
In [69]: import matplotlib.pyplot as plt
         import numpy as np
         mean = [0, 0]
         cov = [[1, 0], [0, 100]]
         x,y = np.random.multivariate_normal(mean, cov, 10000).T
         plt.scatter(x,y)
         plt.show()
```
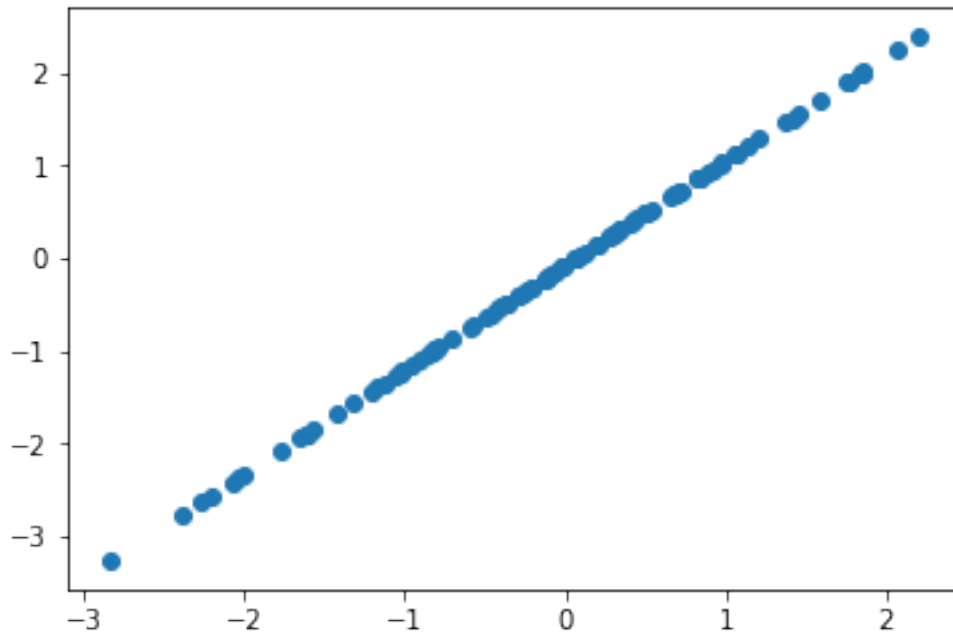


```
In [70]: mean = [0, 0]
         cov = [[1, 2], [2, 5]]
         x,y= np.random.multivariate_normal(mean, cov, 10000).T
         plt.scatter(x,y)
         plt.show()
```

```
In [71]: from sklearn.decomposition import PCA
         mean, cov, n = [0, 0], [[1,1],[1,1.5]], 100
         x = np.random.multivariate_normal(mean, cov, n)
         pca = PCA(n_components=1)
         pca.fit(x)
         X_pca = pca.transform(x)
         X_new = pca.inverse_transform(X_pca)
         plot1 = plt.scatter(X_new[:, 0], X_new[:, 1])
         plt.show()
```

In [75]:
```python
mean = [0,0]
cov = [[1, 0], [0, 100]]
x,y= np.random.multivariate_normal(mean, cov, 100000).T
bins = np.linspace(-4, 6, 100)
plt.hist(x, bins)
plt.show()
```
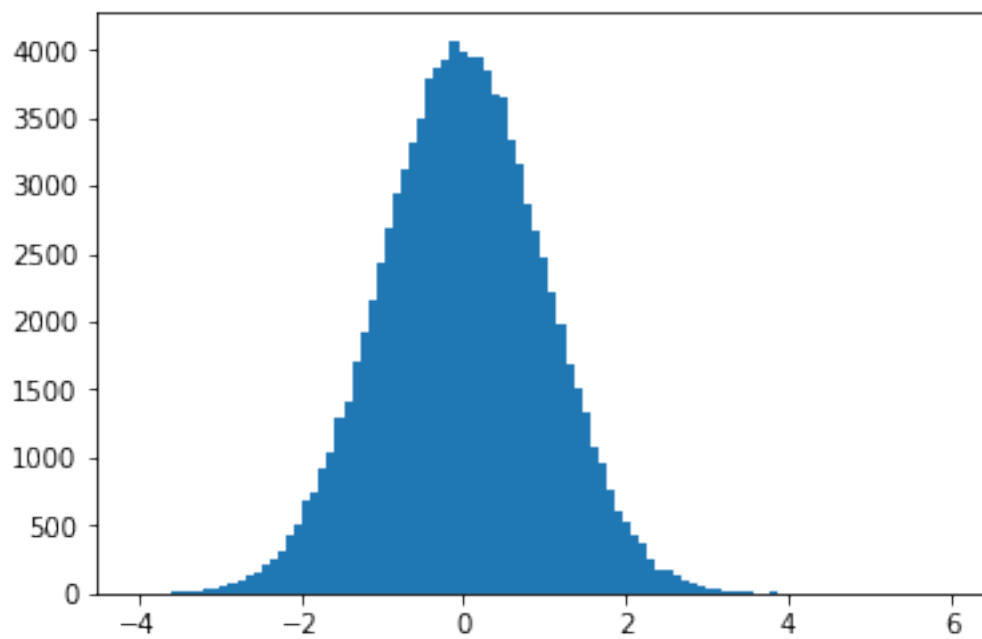
```
In [93]: mean, sigma = 0, 0.2
         x = np.random.normal(mean, sigma, 10000)
         count, bins, ignored = plt.hist(x, 100, density=True)
         plt.plot(bins, 1/(sigma * np.sqrt(2 * np.pi)) *np.exp( - (bins - mean)**2 / (2 * sigma
         plt.show()
```
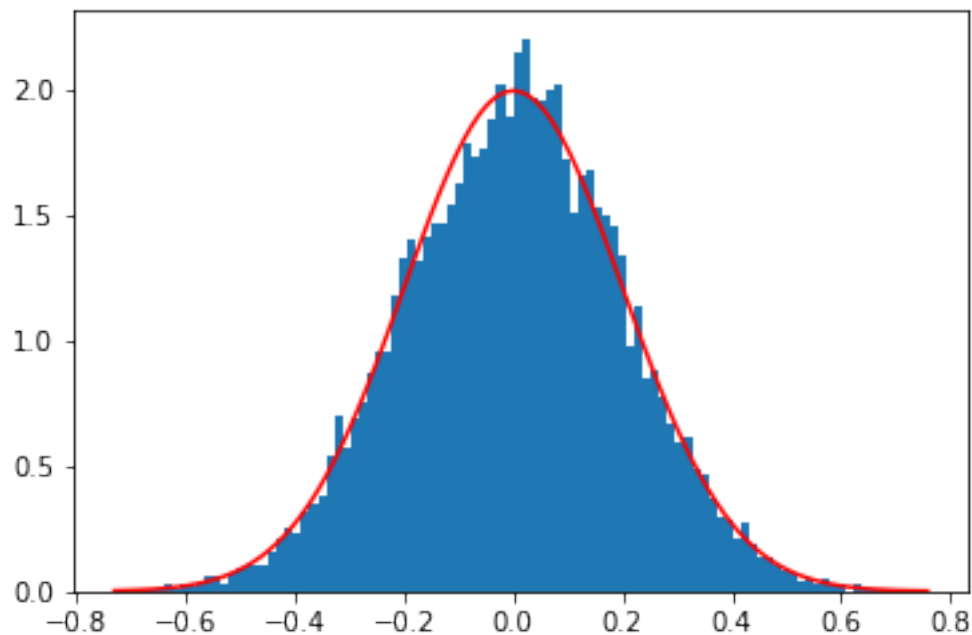


```
In [12]: from math import pi

         def multivariate_pdf(mean, cov, x, n):
             P,D = cov.diagonalize()
             B = P*(D**1/2)
             m1 = np.absolute(B*Transpose(B))**1/2
             m2 = (Transpose(B)**-1)*(B**-1)
             p = (1/(((2*pi)**n/2)*m1))*exp(-1/2*Transpose(x-mean)*m2*(x-mean))[0]
             return p


         mean = Matrix([1, 1])
         cov = Matrix([[3, 0.3], [0.3, 2]])
         n=2
         x=Matrix([1,2])


         multivariate_pdf(mean,cov,x,n)
```

```
Out[12]: array([[0.00203868063889270, 0.00760250335548637],
                 [0.00760250335548637, 0.0192081458071090]], dtype=object)
```