Aufgabe 2b

In [4]:
```python
#Notwendige Bibliotheken
import numpy as np
from scipy.stats import multivariate_normal
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

Mittelwertsvektor und Autokovarianzmatrix

In [5]:
```python
def mittelwertsvektor(x):
    return (1/len(x))*sum(x)

def autokovarianzmatrix(x):
    temp=0
    for x_i in x:
        temp2=x_i - mittelwertsvektor(x)
        temp+=(temp2)*np.transpose(temp2)
    return (1/len(x))*temp
```

In [6]:
```python
X = [np.array([[1,2]]), np.array([[-1,-1]]), np.array([[-5,1]]), np.array([[1,2]])]
```

In [7]:
```python
mittelwertsvektor(X)
```

Out[7]:
```
array([[-1.,  1.]])
```

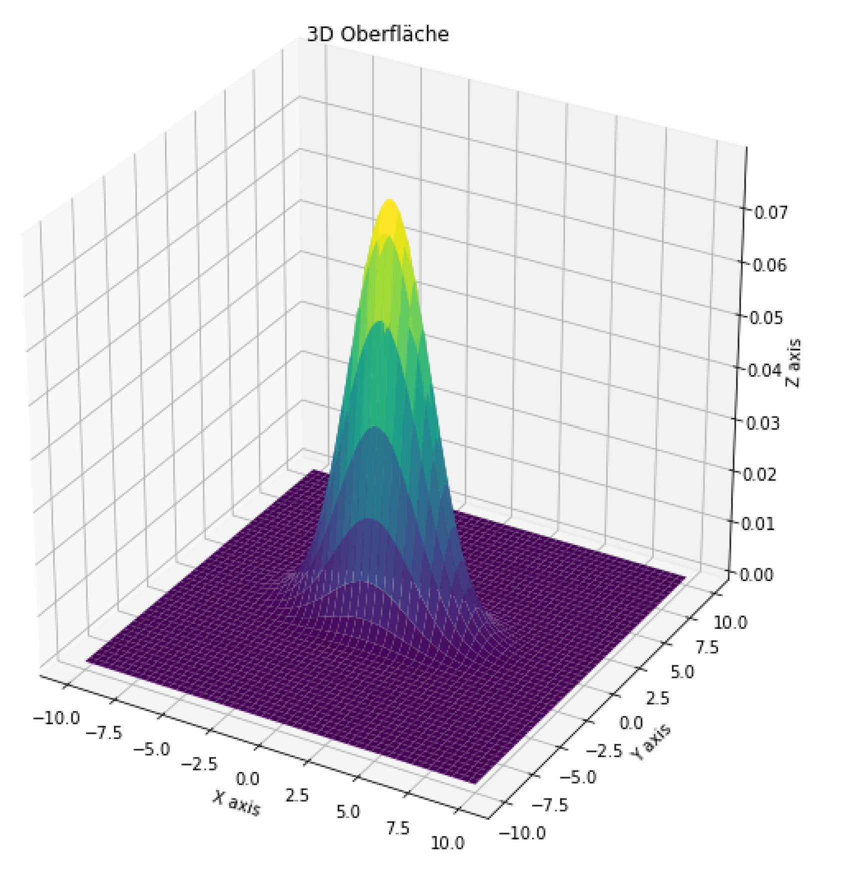
In [8]:
```python
autokovarianzmatrix(X)
```

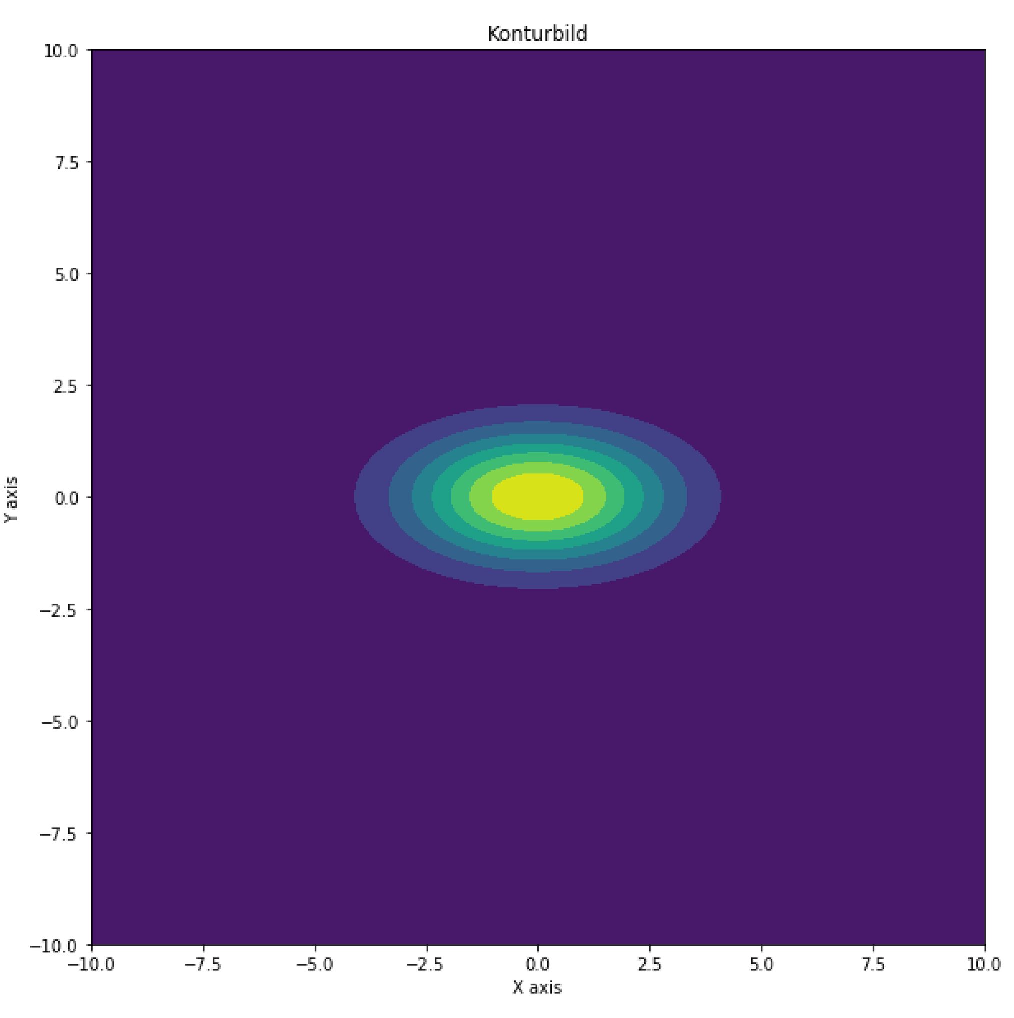
Out[8]:
```
array([[6. , 1. ],
       [1. , 1.5]])
```

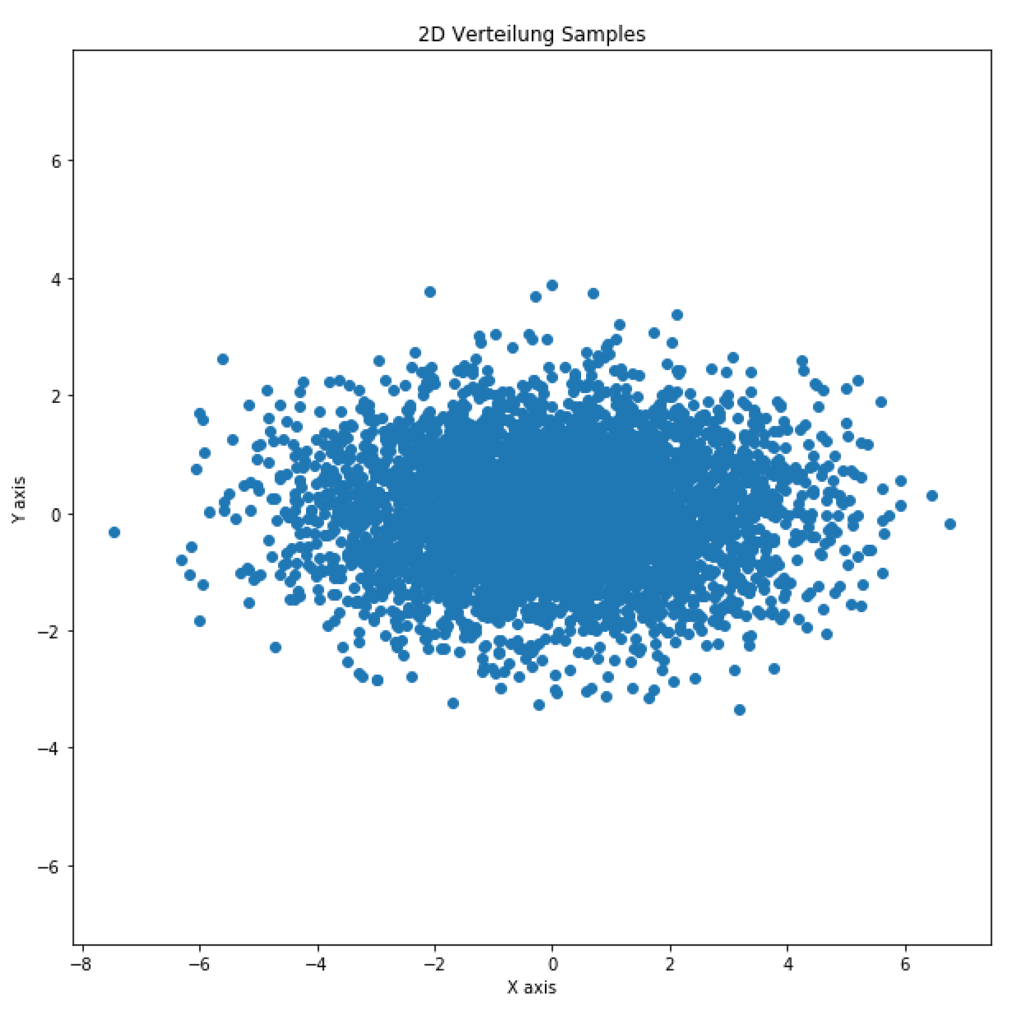
Visualisierung von Normalverteilung

In [9]:
```python
def generate_data(mittelwerts_vektor, kovarianz_matrix):
    #Erstellung von 500 Samples fuer x und y zwischen -8 und 8
    x = np.linspace(-10,10,500)
    y = np.linspace(-10,10,500)
    X, Y = np.meshgrid(x,y)
    pos = np.empty(X.shape + (2,))
    pos[:, :, 0] = X
    pos[:, :, 1] = Y
    random_variable = multivariate_normal(mittelwerts_vektor, kovarianz_matrix)
    return X, Y, random_variable, pos

def plot_3d(X, Y, random_variable, pos):
    #Erstellung des 3D Plot
    fig = plt.figure(figsize=(10, 10))
    ax = fig.gca(projection='3d')
    ax.plot_surface(X, Y, random_variable.pdf(pos),cmap='viridis',linewidth=0)
    ax.set_title("3D Oberfläche")
    ax.set_xlabel('X axis')
    ax.set_ylabel('Y axis')
    ax.set_zlabel('Z axis')
    plt.show()

def plot_contures(X, Y, random_variable, pos):
    #Erstellung Konturbild
    fig = plt.figure(figsize=(10, 10))
    ax = fig.add_subplot(111,aspect='equal')
    ax.contourf(X, Y, random_variable.pdf(pos))
    ax.set_title("Konturbild")
    ax.set_xlabel('X axis')
    ax.set_ylabel('Y axis')
    plt.show()

def plot_samples(sample_size, mittelwerts_vektor, kovarianz_matrix):
    x_random, y_random = np.random.multivariate_normal(mittelwerts_vektor, kovarianz_matrix, sample_siz
e).T
    fig = plt.figure(figsize=(10, 10))
    ax = fig.add_subplot(111,aspect='equal')
    ax.plot(x_random, y_random, 'o')
    ax.set_title("2D Verteilung Samples")
    ax.set_xlabel('X axis')
    ax.set_ylabel('Y axis')
    ax.axis('equal')
    plt.show()
```

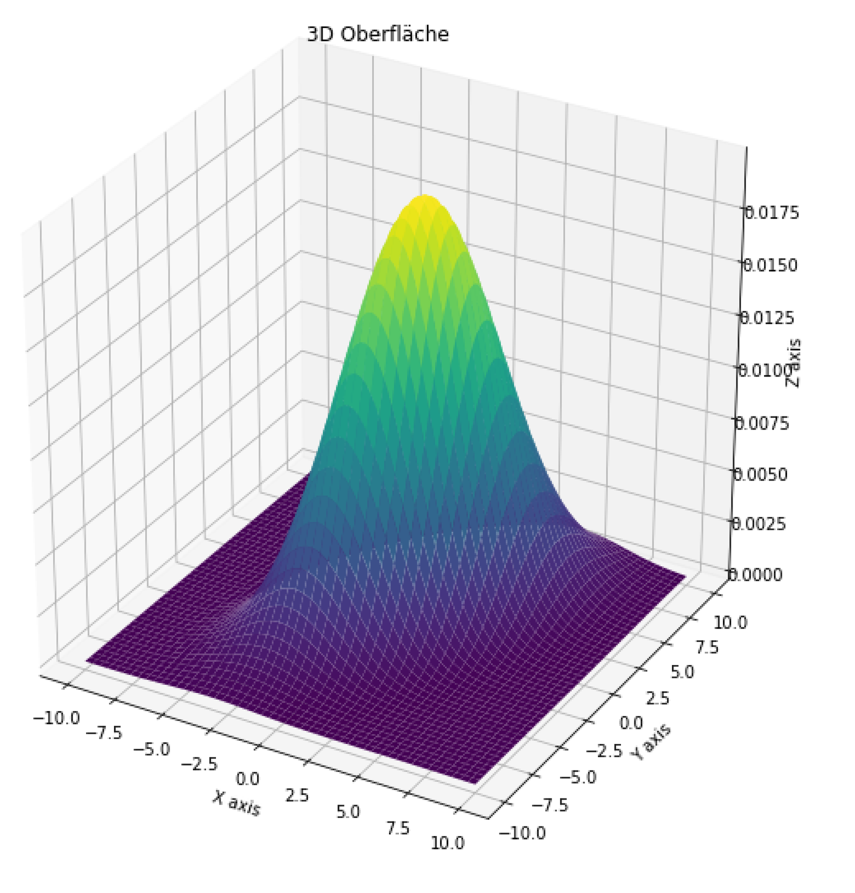
In [10]:
```python
mittelwerts_vektor = np.array([0,0])
kovarianz_matrix = np.matrix([[4,0], [0,1]])
sample_size= 5000

X, Y, rv, pos = generate_data(mittelwerts_vektor, kovarianz_matrix)
plot_3d(X, Y, rv, pos)
plot_contures(X, Y, rv, pos)
plot_samples(sample_size, mittelwerts_vektor, kovarianz_matrix)
```

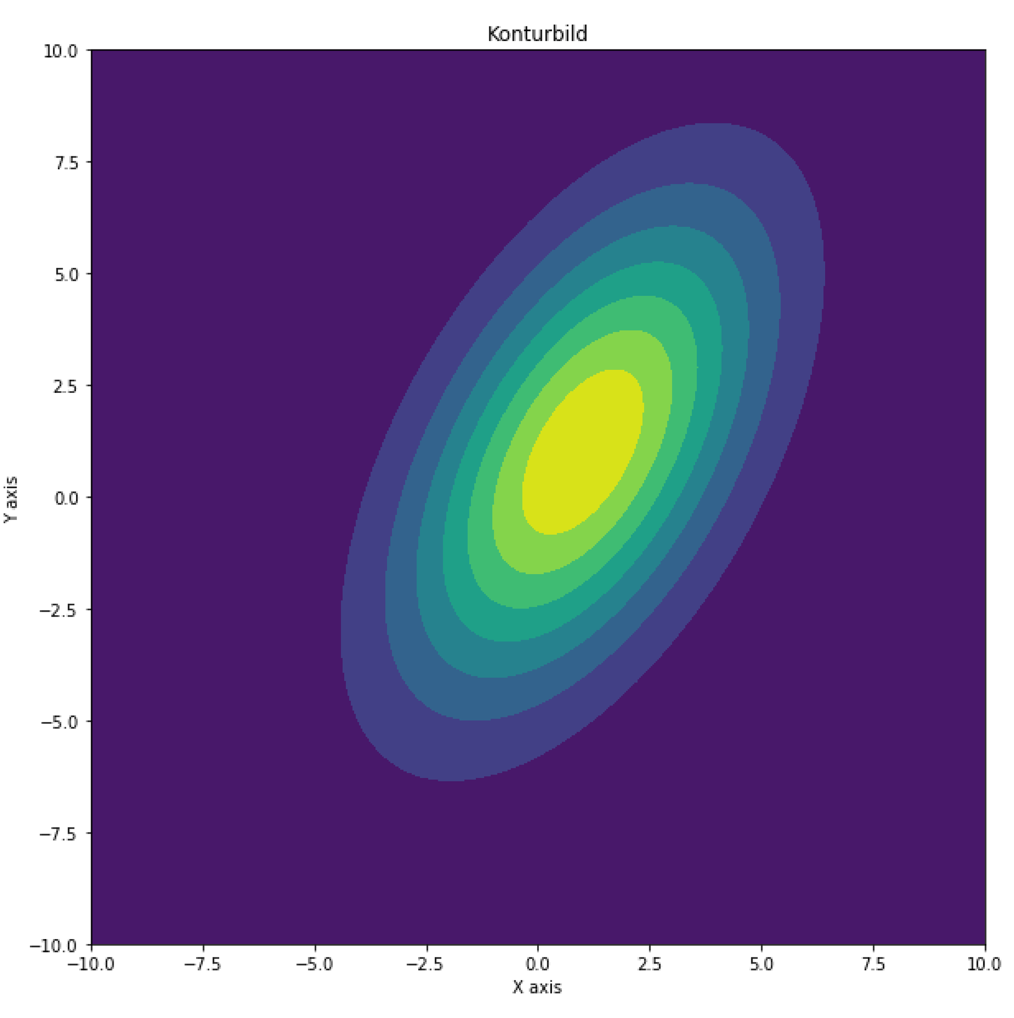


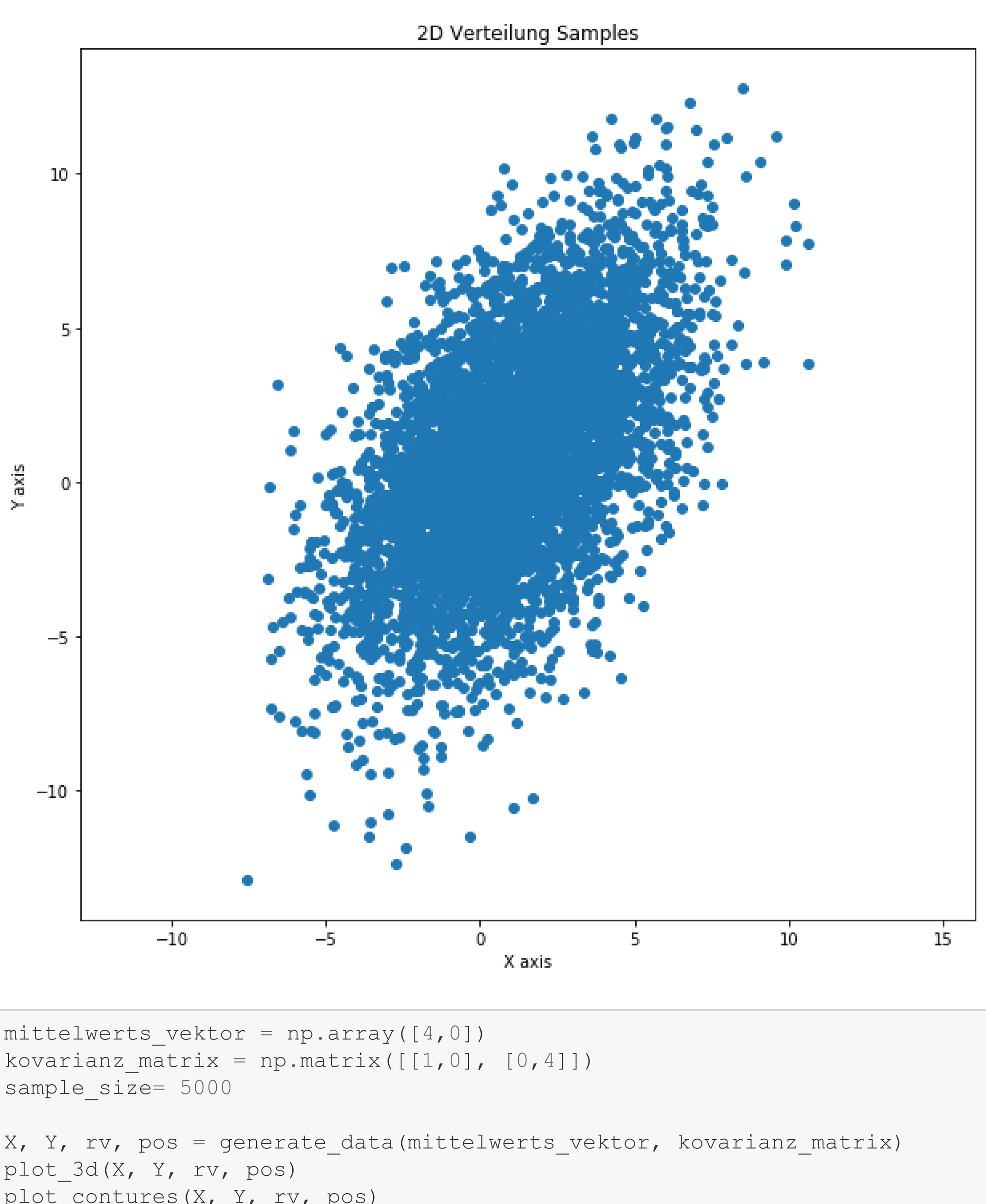




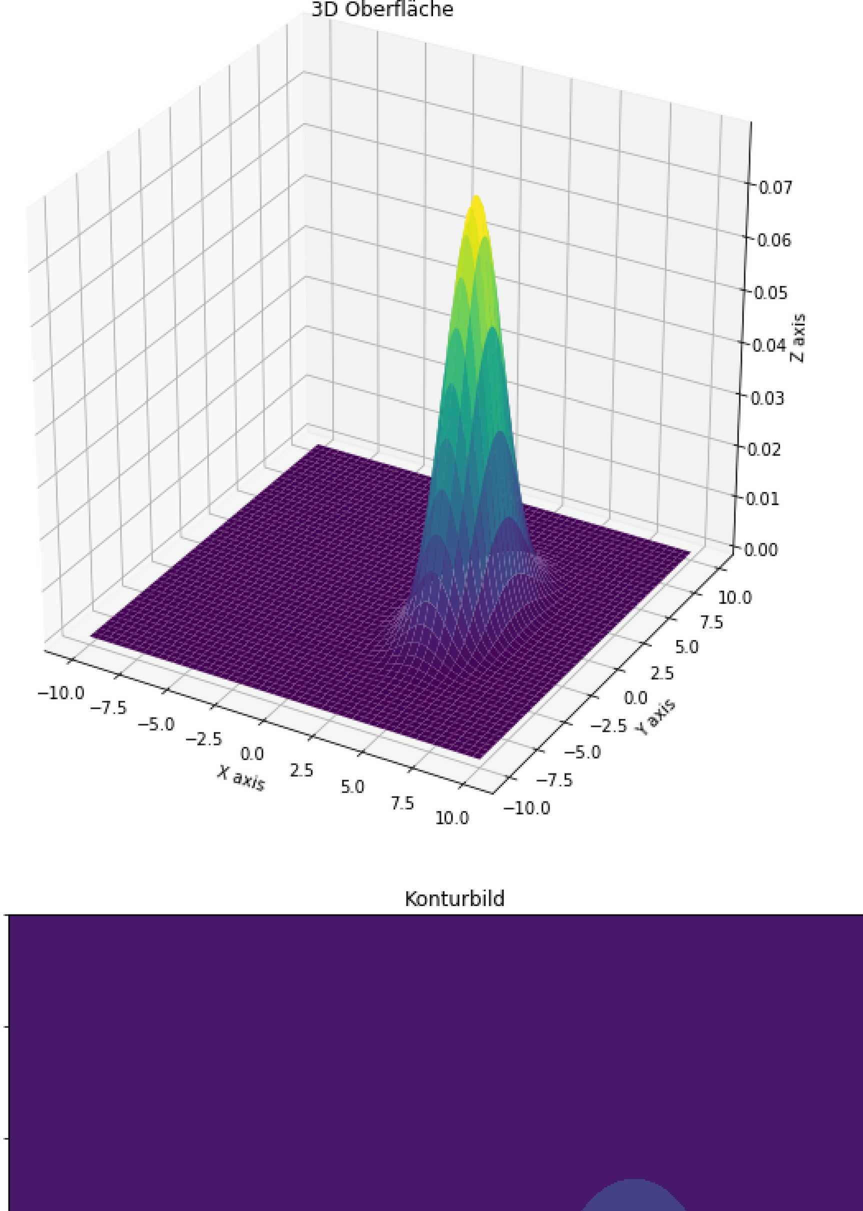
In [11]:
```python
mittelwerts_vektor = np.array([1,1])
kovarianz_matrix = np.matrix([[7,3*np.sqrt(3)], [3*np.sqrt(3),13]])
sample_size= 5000

X, Y, rv, pos = generate_data(mittelwerts_vektor, kovarianz_matrix)
plot_3d(X, Y, rv, pos)
plot_contures(X, Y, rv, pos)
plot_samples(sample_size, mittelwerts_vektor, kovarianz_matrix)
```

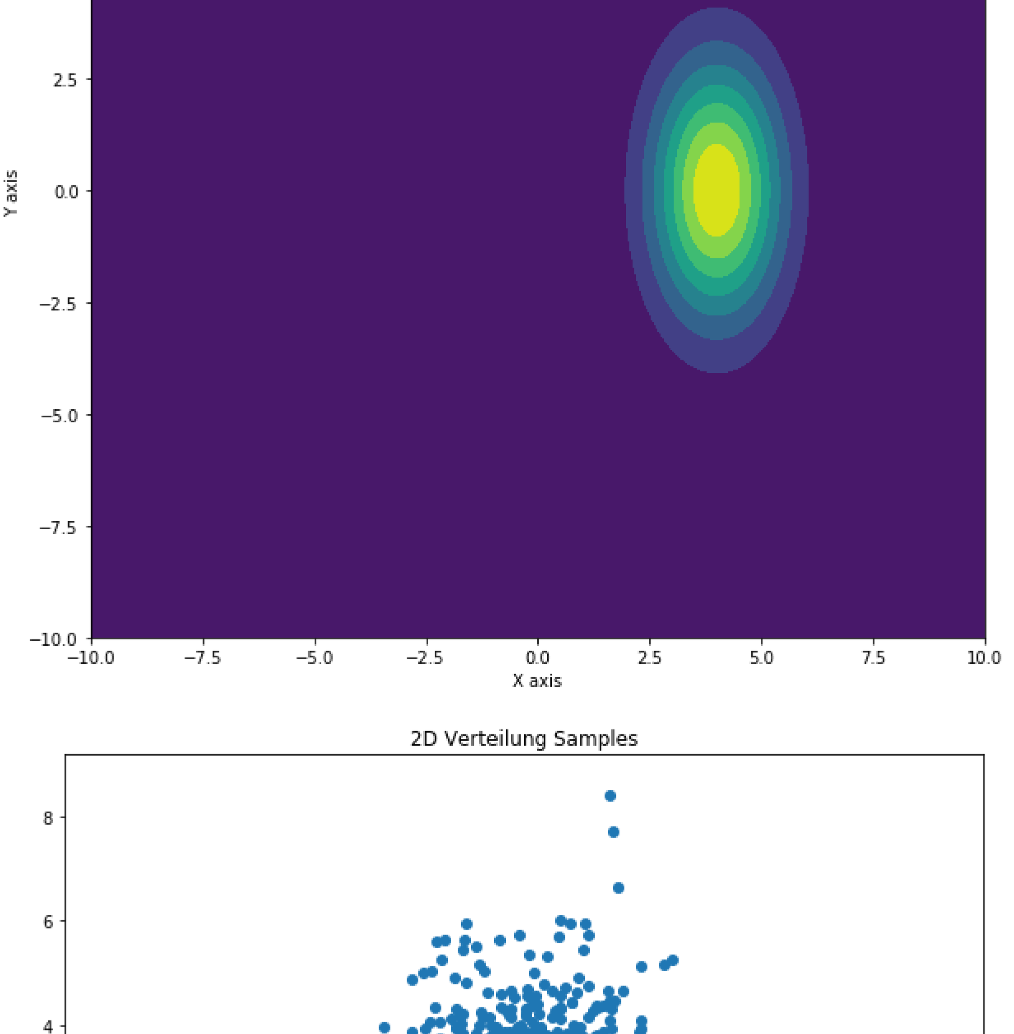


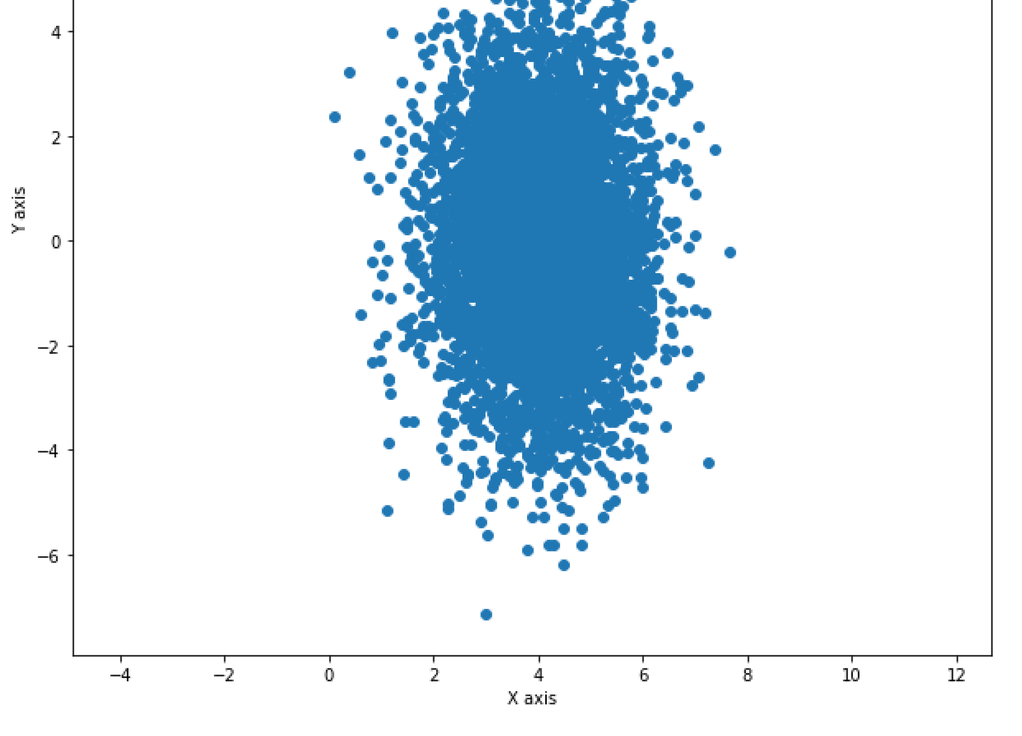




In [31]:
```python
mittelwerts_vektor = np.array([4,0])
kovarianz_matrix = np.matrix([[1,0], [0,4]])
sample_size= 5000

X, Y, rv, pos = generate_data(mittelwerts_vektor, kovarianz_matrix)
plot_3d(X, Y, rv, pos)
plot_contures(X, Y, rv, pos)
plot_samples(sample_size, mittelwerts_vektor, kovarianz_matrix)
```







In [ ]: