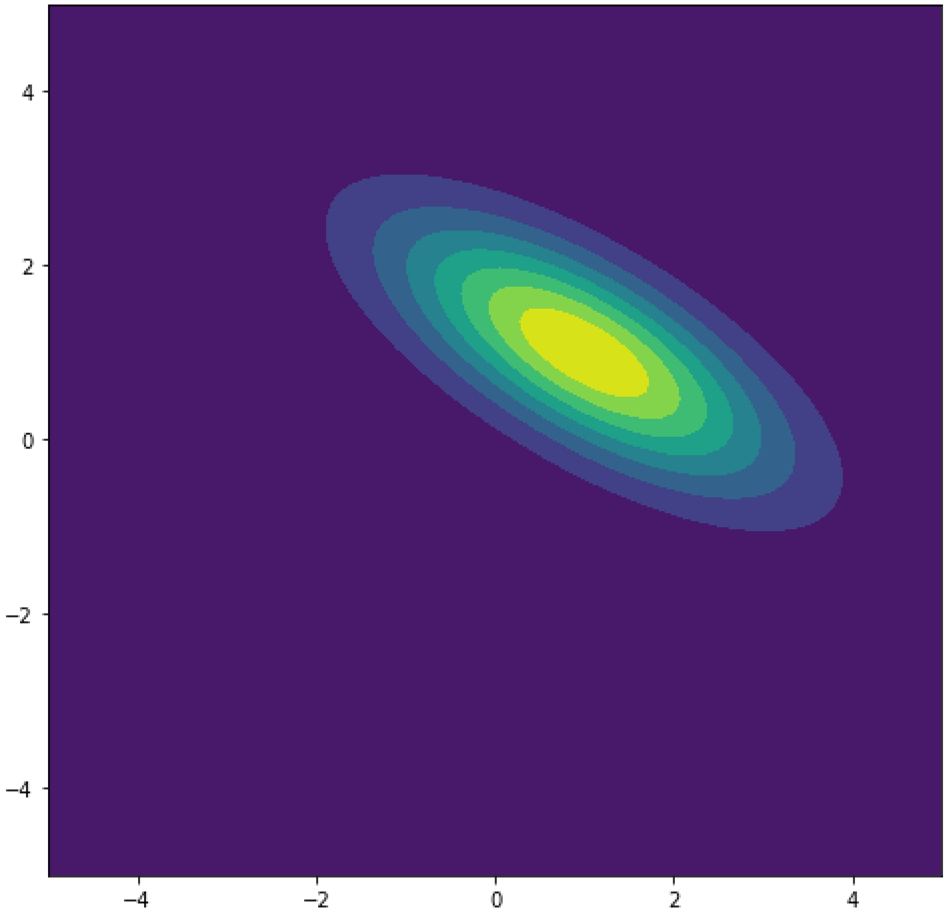
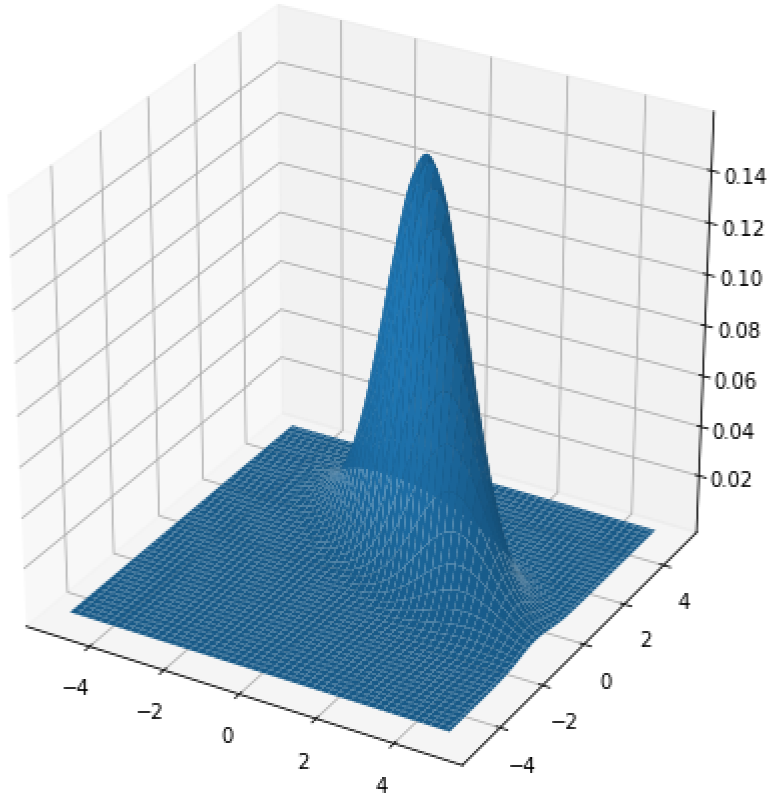
1 - Visualizing the Multivariate Gaussian (Normal) Probability Density Function

In [172]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal
from mpl_toolkits.mplot3d import Axes3D

x          = np.linspace(-5,5,100)
y          = np.linspace(-5,5,100)
X,Y        = np.meshgrid(x,y)
S          = np.meshgrid(x,y)

pos        = np.zeros(X.shape + (2,))
pos[:,:,0] = X
pos[:,:,1] = Y

mean       = [1,1]
cov        = [[2,1],[-1,1]]
var        = multivariate_normal(mean, cov)
fig0       = plt.figure(figsize=(8,8))
c_fig      = fig0.add_subplot(111)
c_fig.contourf(X,Y,var.pdf(pos))

fig1       = plt.figure(figsize=(8,8))
surf_fig   = fig1.add_subplot(111,projection='3d')
surf_fig.plot_surface(X,Y,var.pdf(pos))
```

```
Out[172]: <mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x269a86db6a0>
```





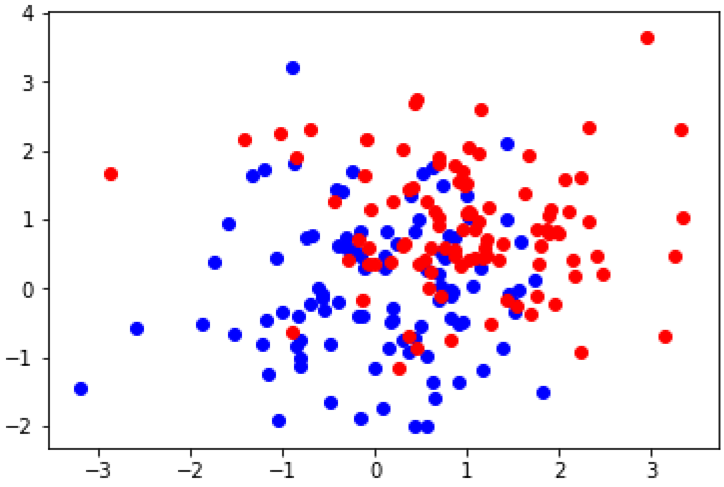
2 - Generate Synthetic Data for Classification

In [171]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal

# mean and cov for the first dataset
mean1 = [0,0]
cov1  = [[1,0.1],[0.1,1]]

# mean and cov for the second dataset
mean2 = [1,1]
cov2  = [[1,-0.1],[-0.1,1]]

# initializing sample points
data1            = np.random.multivariate_normal(mean1,cov1,100)
data2            = np.random.multivariate_normal(mean2,cov2,100)
# Plotting
fig_gen_syn_data = plt.figure()
plot_gsd         = fig_gen_syn_data.add_subplot(111)
plot_gsd.scatter(data1[:,0],data1[:,1],c='blue')
plot_gsd.scatter(data2[:,0],data2[:,1],c='red')
```

Out[171]:

```
<matplotlib.collections.PathCollection at 0x269a8766898>
```



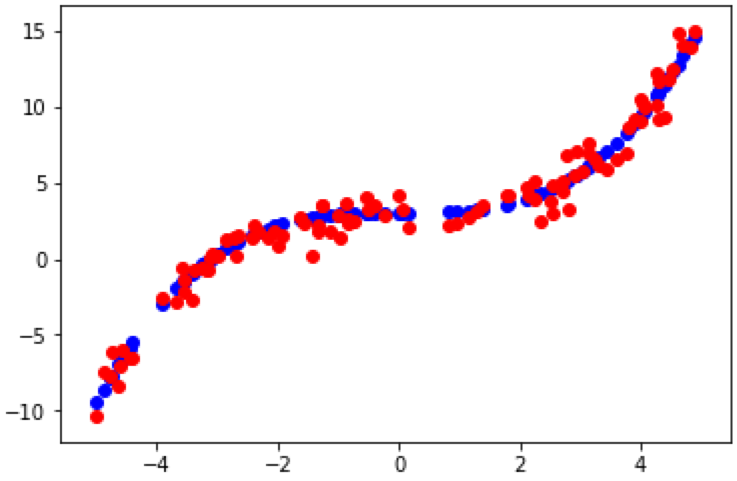
3 - Generate Synthetic Data for Regression

In [170]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal

uniform_var = np.random.uniform(-5,5,100)
target_val = 0.1 * (uniform_var**3) + 3
noise = np.random.normal(size=100)
noisy_obs = target_val + noise
# plotting
fig_noisy_data = plt.figure()
plot = fig_noisy_data.add_subplot(111)
plot.scatter(uniform_var,target_val,c='blue')
plot.scatter(uniform_var,noisy_obs,c='red')
```

Out[170]:

```
<matplotlib.collections.PathCollection at 0x269a9ff3780>
```



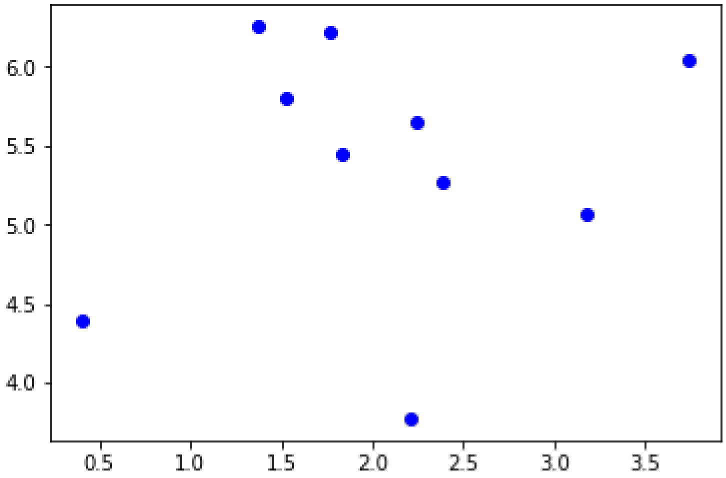
4 - Standardizing Data

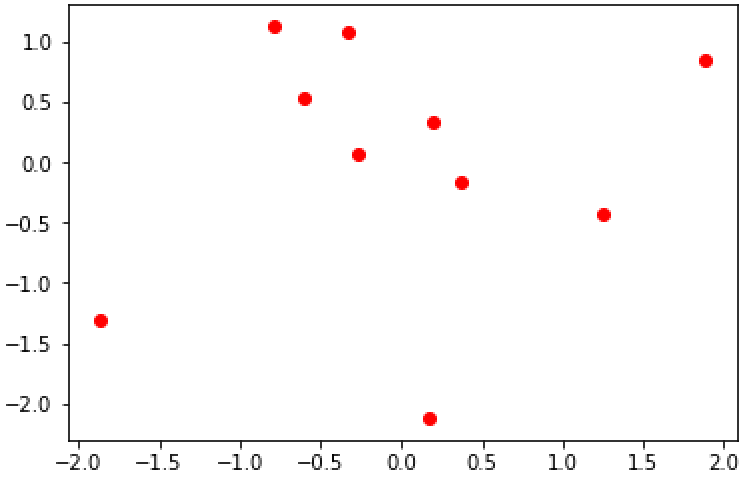
In [169]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal

mean        = [2,5]
cov         = [[1,-0.1],[-0.1,1]]
X           = np.random.multivariate_normal(mean,cov,size=10)
X_hat       = np.empty(X.shape)
X_hat[:,0] = (X[:,0]-np.mean(X[:,0]))/np.std(X[:,0])
X_hat[:,1] = (X[:,1]-np.mean(X[:,1]))/np.std(X[:,1])

fig1        = plt.figure()
plot1       = fig1.add_subplot(111)
plot1.scatter(X[:,0],X[:,1], c='blue')

fig2        = plt.figure()
plot2       = fig2.add_subplot(111)
plot2.scatter(X_hat[:,0],X_hat[:,1], c='red')
```

Out[169]:

```
<matplotlib.collections.PathCollection at 0x269aa084668>
```





In [ ]: