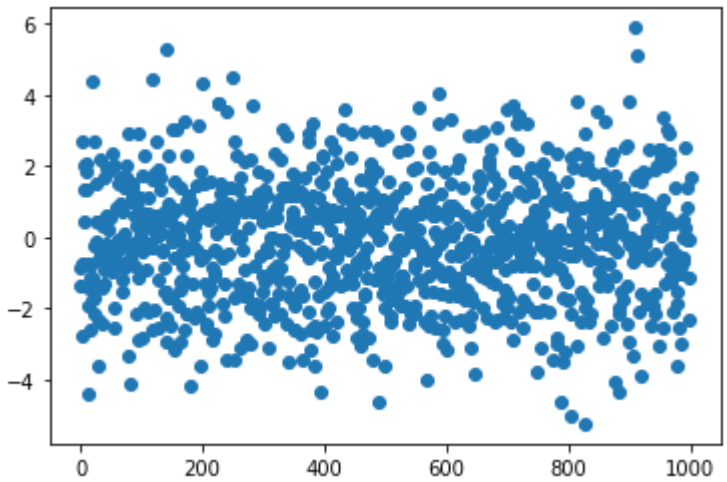
In [81]:
```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import numpy as np
from scipy.stats import norm, invgamma
```

In [82]:
```
theta_1_true, theta_2_true = 0.0, 3.0
```

In [97]:
```
n = 1000
data_distr = norm(theta_1_true, np.sqrt(theta_2_true))
x = data_distr.rvs(n)
plt.plot(x, 'o')
```

Out[97]:
```
[<matplotlib.lines.Line2D at 0x7fa67a779550>]
```

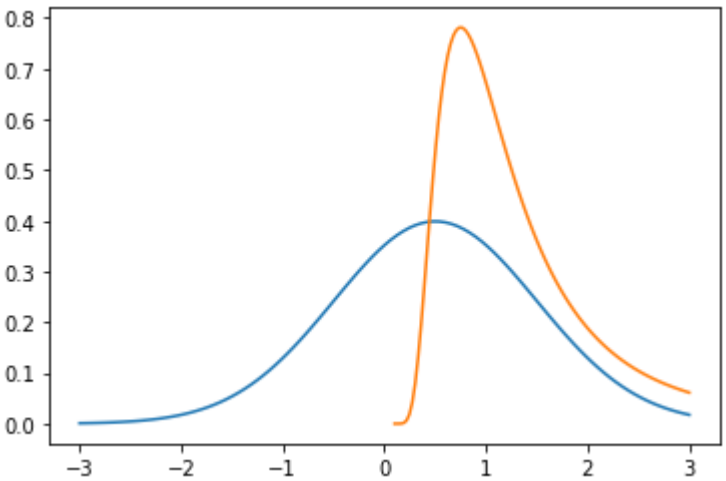


In [98]:
```
# prior of theta_1
mu_0, sigma2_0 = 0.5, 1.0
prior_theta_1 = norm(mu_0, np.sqrt(sigma2_0))
```

In [99]:
```
# prior of theta_2
a, b = 3, 3
prior_theta_2 = invgamma(a, scale=b)
```

In [100…
```
theta_1_vals = np.linspace(-3, 3, 101)
theta_2_vals = np.linspace(0.1, 3, 101)
plt.plot(theta_1_vals, prior_theta_1.pdf(theta_1_vals))
plt.plot(theta_2_vals, prior_theta_2.pdf(theta_2_vals))
```

Out[100…
```
[<matplotlib.lines.Line2D at 0x7fa67a822730>]
```



In [101…
```
def sample_from_posterior_1(mu_0, sigma2_0, theta_2, x):
    n = len(x)
    mu_0_prime = (mu_0/sigma2_0 + np.sum(x)/theta_2)/(1/sigma2_0 + n/theta
    sigma2_0_prime = 1/(1/sigma2_0 + n/theta_2)
    distr = norm(mu_0_prime, np.sqrt(sigma2_0_prime))
    return distr.rvs(1)
```

In [102…
```
def sample_from_posterior_2(a, b, theta_1, x):
    n = len(x)
    a_prime = a + n/2
    b_prime = b + 0.5 * np.sum((x - theta_1)**2)
    distr = invgamma(a_prime, scale=b_prime)
    return distr.rvs(1)
```

In [103…
```
k = 100
Q = 10000
samples = []
theta_1, theta_2 = 0.5, 1.0
for j in range(k + Q):
    theta_1 = sample_from_posterior_1(mu_0, sigma2_0, theta_2, x)
    theta_2 = sample_from_posterior_2(a, b, theta_1, x)
    if j >= k:
        samples.append((theta_1, theta_2))
samples = np.array(samples)
samples.shape
```

Out[103…
```
(10000, 2, 1)
```

In [104…
```
np.mean(samples[:,1])
```

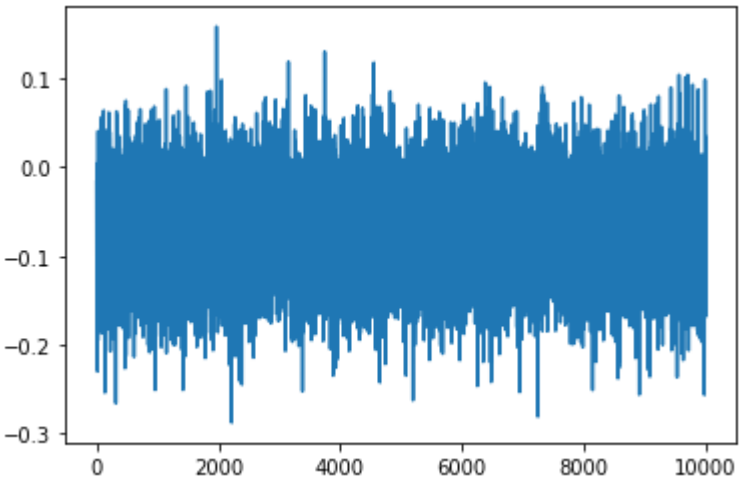
Out[104…
```
2.899165908717658
```

In [105…
```
np.mean(samples[:,0])
```

Out[105…
```
-0.07063969896688872
```

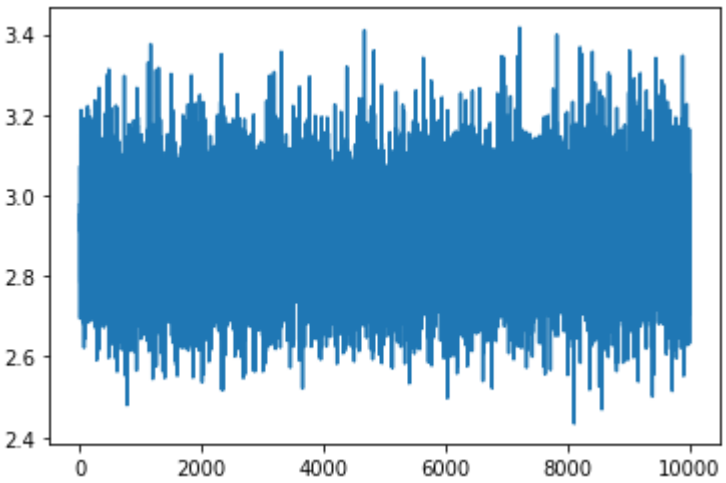
In [106…
```
plt.plot(samples[:,0])
```

Out[106… [<matplotlib.lines.Line2D at 0x7fa67a019640>]



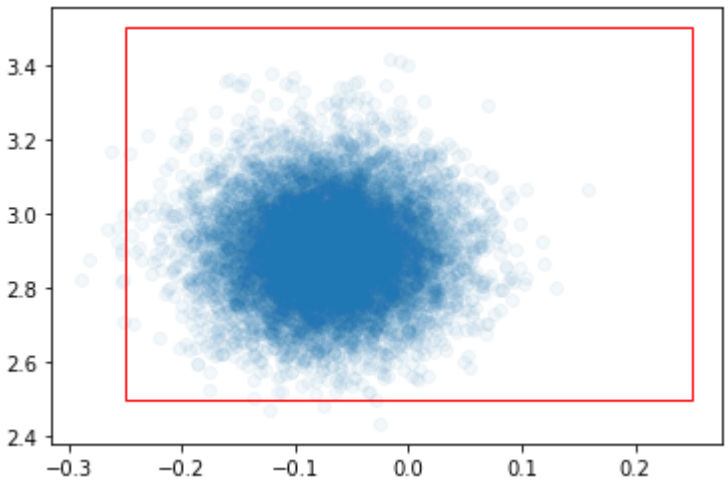
In [107…
```
plt.plot(samples[:,1])
```

Out[107… [<matplotlib.lines.Line2D at 0x7fa681049e80>]



In [118…
```
fig, ax = plt.subplots()
ax.scatter(samples[:,0], samples[:,1], alpha=0.05)
rect = patches.Rectangle((-0.25, 2.5), 0.5, 1.0, edgecolor='r', facecolor=
ax.add_patch(rect)
```

Out[118… <matplotlib.patches.Rectangle at 0x7fa67a0426d0>



In [119…
```
i = 0
for row in samples:
    if -0.25 < row[0] < 0.25 and 2.5 < row[1] < 3.5:
        i += 1
```

In [120…
```
i/Q
```

Out[120… 0.9984