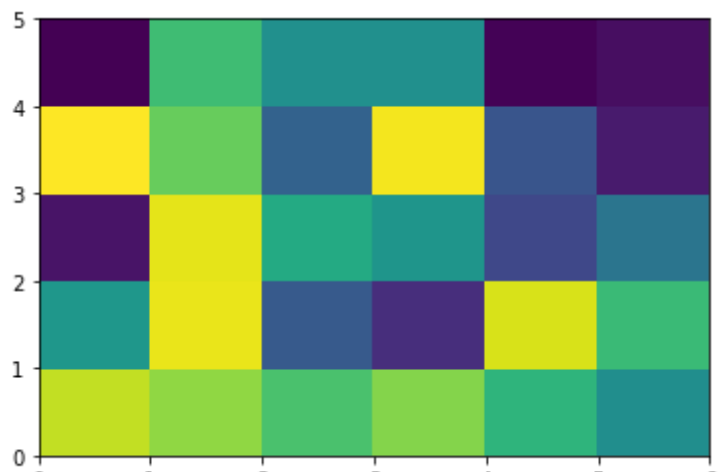
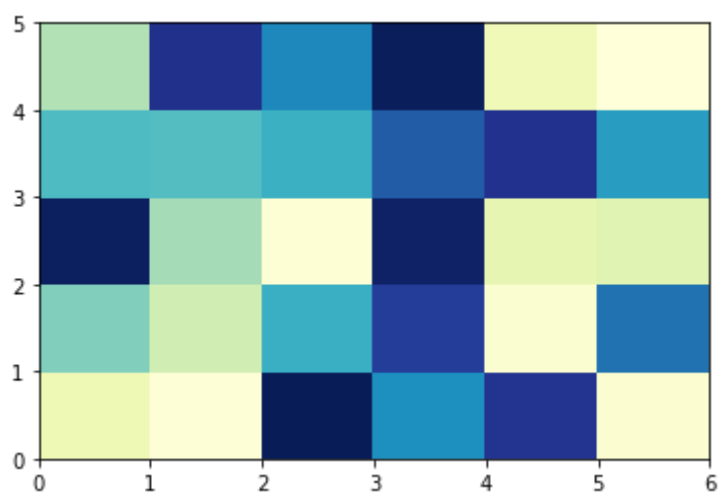
In [1]:
```
# pcolor()
# The routine pcolor() creates a pseudocolor plot with a rectangular
# (nonsquare) grid. Pseudocolor means the object or image is rendered
# in colors different than those in which it was recorded.

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
# Let's create a nonsquare matrix and use the routine pcolor()
# to visualize it, as shown here:
data = np.random.rand(5, 6)
plt.pcolor(data)
plt.show()
```



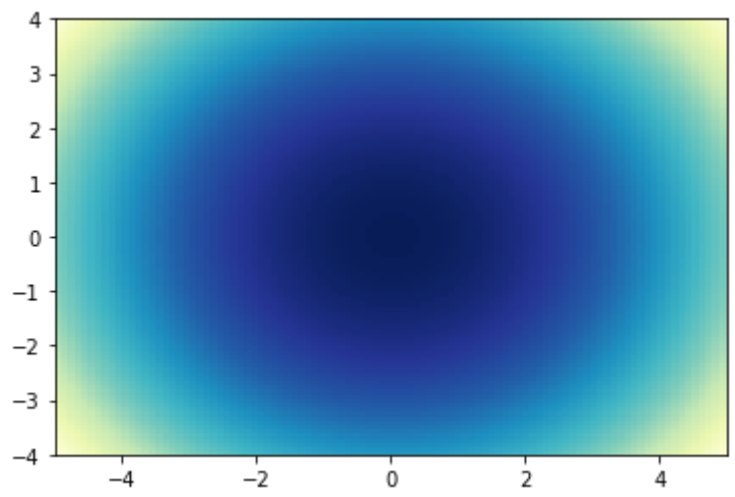
In [2]:
```
# You can also use custom color maps as follows:

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
data = np.random.rand(5, 6)
plt.pcolor(data, cmap='YlGnBu_r')
plt.show()
```



In [6]:
```
# Let's now try adding shading. Let's create a new dataset, as follows:

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
N = 100
X, Y = np.meshgrid(np.linspace(-5, 5, N),
np.linspace(-4, 4, N))
Z = (X**2 + Y**2)
# You can visualize it as follows:
plt.pcolor(X, Y, Z,cmap='YlGnBu_r')
plt.show()
```



In [9]:
```
# colorbar()
# You can also add a color bar that corelates with the
# magnitude of data points in the visualization.
# The routine colorbar() does the trick. The following is the code:

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
N = 100
X, Y = np.meshgrid(np.linspace(-5, 5, N),
np.linspace(-5, 5, N))
Z = (X**2 + Y**2)
img = plt.imshow(Z, cmap='YlGnBu_r')
plt.colorbar(img)
plt.show()
```


In [11]:
```
# You can shrink the color bar and change its position as follows:

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
N = 100
X, Y = np.meshgrid(np.linspace(-5, 5, N),
np.linspace(-5, 5, N))
Z = (X**2 + Y**2)
img = plt.imshow(Z, cmap='coolwarm')
plt.colorbar(img, shrink=0.6)
plt.show()
```

In [12]:
```
# You can also extend the color bar as follows:

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
N = 100
X, Y = np.meshgrid(np.linspace(-5, 5, N),
np.linspace(-5, 5, N))
Z = (X**2 + Y**2)
img = plt.imshow(Z, cmap='coolwarm')
plt.colorbar(img, extend='both')
plt.show()
```