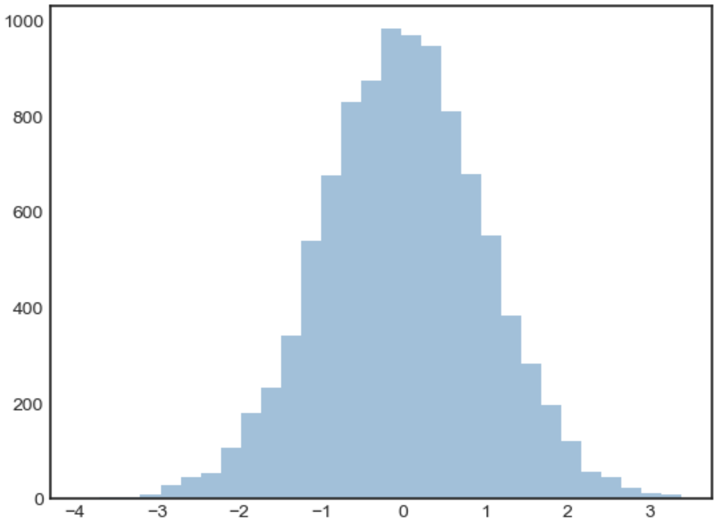
In [1]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('seaborn-white')
```

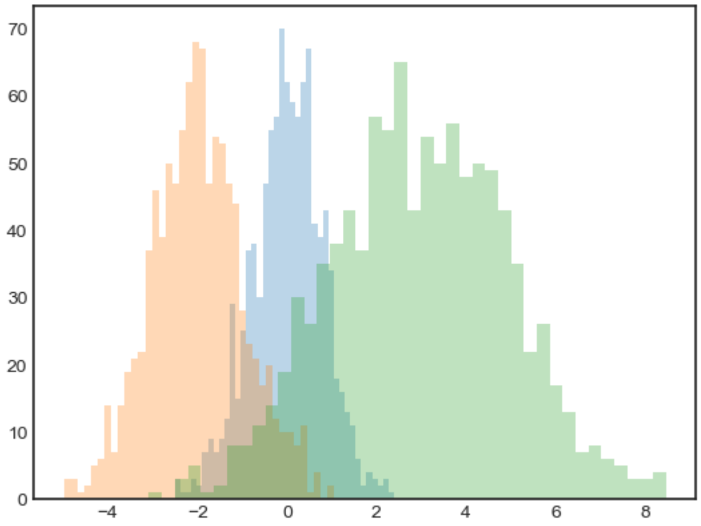
In [24]:
```
data=np.random.randn(10000)
plt.hist(data,bins=30,alpha=0.5,histtype="stepfilled",color="steelblue",edgecolor="none")
```

Out[24]:
```
(array([  1.,   2.,   3.,   9.,  27.,  44.,  54., 108., 180., 233., 341.,
        540., 677., 832., 875., 984., 971., 948., 810., 680., 552., 384.,
        284., 196., 120.,  57.,  44.,  24.,  12.,   8.]),
 array([-3.93423338, -3.69020093, -3.44616849, -3.20213604, -2.95810359,
        -2.71407115, -2.4700387 , -2.22600625, -1.98197381, -1.73794136,
        -1.49390891, -1.24987647, -1.00584402, -0.76181157, -0.51777913,
        -0.27374668, -0.02971423,  0.21431822,  0.45835066,  0.70238311,
         0.94641556,  1.190448  ,  1.43448045,  1.6785129 ,  1.92254534,
         2.16657779,  2.41061024,  2.65464268,  2.89867513,  3.14270758,
         3.38674002]),
 [<matplotlib.patches.Polygon at 0x2a90872c1c0>])
```



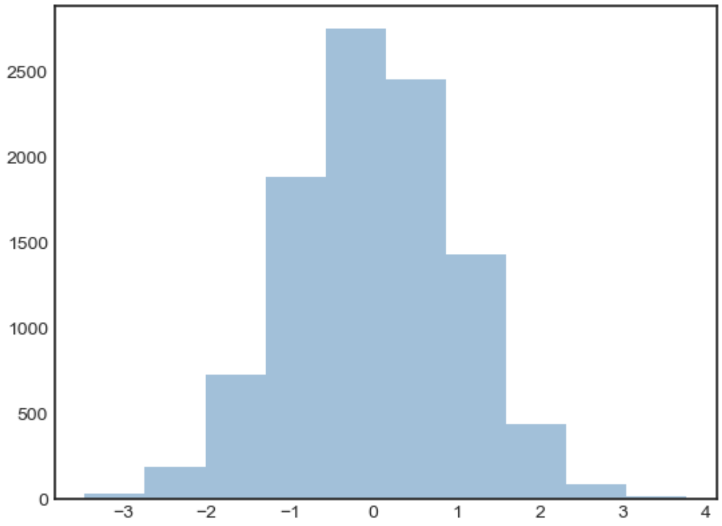
```
In [26]: #plotting mutliple histograms
         x1=np.random.normal(0,0.8,1000)
         x2=np.random.normal(-2,1,1000)
         x3=np.random.normal(3,2,1000)
         kwargs=dict(histtype="stepfilled",alpha=0.3,bins=40)
         plt.hist(x1,**kwargs)
         plt.hist(x2,**kwargs)
         plt.hist(x3,**kwargs)
```

```
Out[26]: (array([ 1.,  0.,  3.,  5.,  1.,  2.,  8.,  8., 11., 14., 19., 30., 26.,
                 35., 38., 43., 37., 57., 55., 65., 43., 54., 50., 56., 48., 50.,
                 49., 43., 35., 22., 26., 17., 13.,  7.,  8.,  6.,  5.,  3.,  3.,
                  4.]),
          array([-3.07997788, -2.79119923, -2.50242057, -2.21364191, -1.92486326,
                 -1.6360846 , -1.34730594, -1.05852728, -0.76974863, -0.48096997,
                 -0.19219131,  0.09658735,  0.385366  ,  0.67414466,  0.96292332,
                  1.25170198,  1.54048063,  1.82925929,  2.11803795,  2.40681661,
                  2.69559526,  2.98437392,  3.27315258,  3.56193124,  3.85070989,
                  4.13948855,  4.42826721,  4.71704586,  5.00582452,  5.29460318,
                  5.58338184,  5.87216049,  6.16093915,  6.44971781,  6.73849647,
                  7.02727512,  7.31605378,  7.60483244,  7.8936111 ,  8.18238975,
                  8.47116841]),
          [<matplotlib.patches.Polygon at 0x2a90721b550>])
```

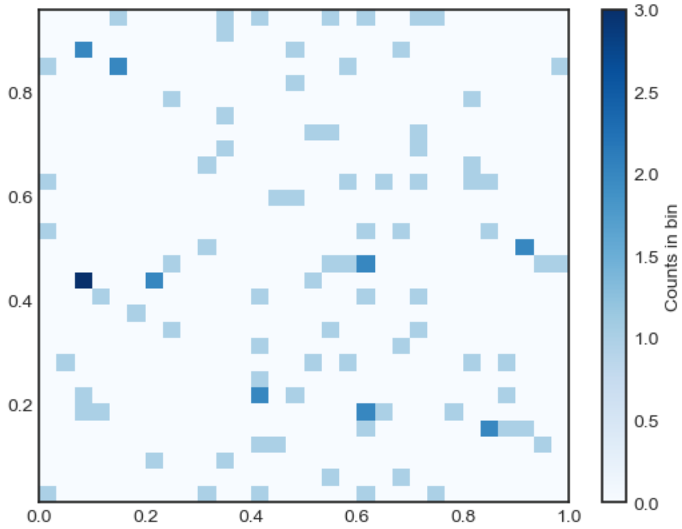


```
In [28]: #np.histrogram() is used to count number of points in given number of bins
         data=np.random.randn(10000)
         plt.hist(data,bins=10,alpha=0.5,histtype="stepfilled",color="steelblue",edgecolor="none")
         count,bin_edges=np.histogram(data,bins=10)#tuple with count of points in each bin and bin edges
         print(count)#prints the number of points in each bin
```

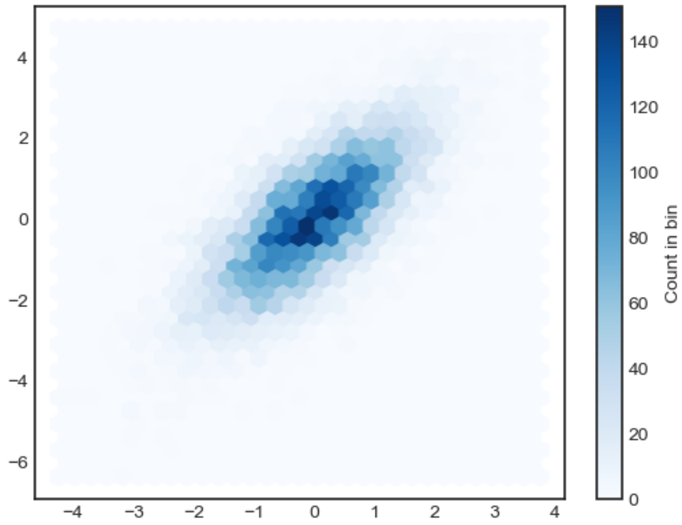
```
[  31  192  727 1881 2747 2451 1429  440   88   14]
```



In [51]:
```
x=np.random.rand(100).round(2)
y=np.random.rand(100).round(2)
plt.hist2d(x,y,bins=30,cmap="Blues")
plt.colorbar().set_label("Counts in bin")#plotting colorbar and setting label
count,x_edges,y_edges=np.histogram2d(x,y,bins=30)
#for histogram with multidimensions use histogramdd
```

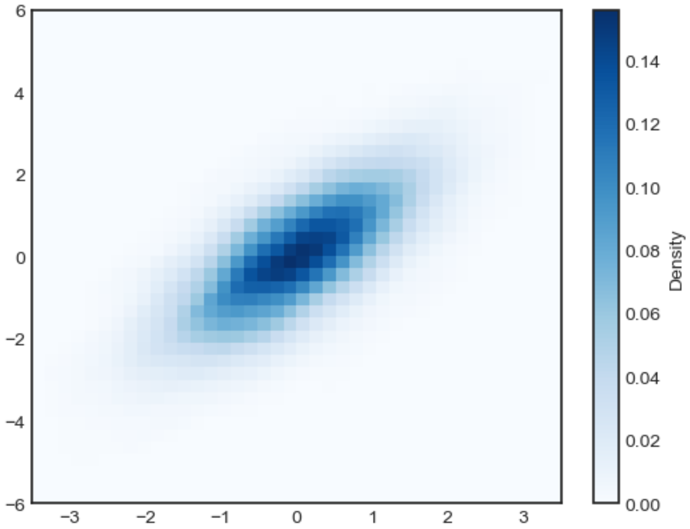


In [53]:
```
#hexagonal plots
mean=[0, 0]
cov=[[1, 1],[1, 2]]
x,y=np.random.multivariate_normal(mean,cov,10000).T#sample data
plt.hexbin(x,y,gridsize=30,cmap="Blues")#instead of bins we use gridsize
plt.colorbar().set_label("Count in bin")#plotting colorbar
```



In [58]:

```
#kernel density estimation
from scipy.stats import gaussian_kde
data=np.vstack([x, y])
kde=gaussian_kde(data)
xgrid=np.linspace(-3.5,3.5,40)
ygrid=np.linspace(-6,6,40)
Xgrid,Ygrid=np.meshgrid(xgrid,ygrid)#meshgrid to plot image
Z=kde.evaluate(np.vstack([Xgrid.ravel(),Ygrid.ravel()]))#kde evaluation
plt.imshow(Z.reshape(Xgrid.shape),origin="lower",aspect="auto",extent=[-3.5,3.5,-6,6],cmap="Blues")#plotting image
plt.colorbar().set_label("Density")
```


Density