

```
In [10]: # Let's begin by importing what we'll need (You'll probably be copying and pastin

# The normal imports
import numpy as np
from numpy.random import randn
import pandas as pd

# Import the stats library from numpy
from scipy import stats

# These are the plotting modules and libraries we'll use:
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

# Command so that plots appear in the iPython Notebook
%matplotlib inline
```

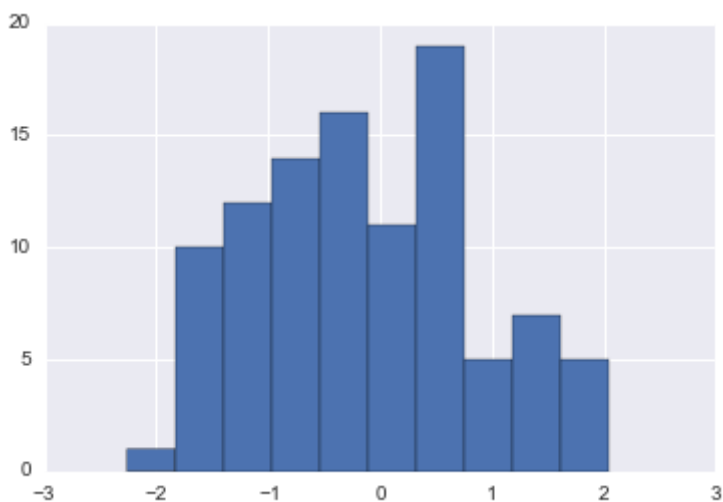
First of all, source of information for what a histogram actually is:

<http://en.wikipedia.org/wiki/Histogram> (<http://en.wikipedia.org/wiki/Histogram>)

```
In [11]: #Create a random normal-dist dataset
dataset1 = randn(100)

#Plot a histogram of the dataset, note bins=10 by default
plt.hist(dataset1)
```

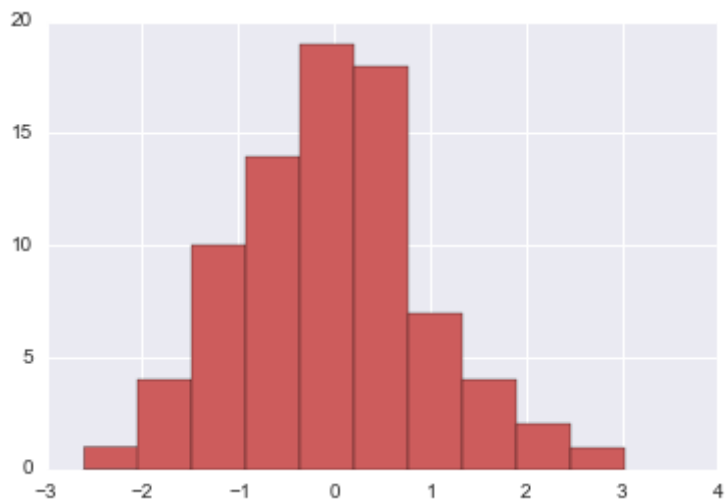
```
Out[11]: (array([ 1., 10., 12., 14., 16., 11., 19., 5., 7., 5.]),
array([-2.259995, -1.82911927, -1.39824353, -0.9673678, -0.53649206,
-0.10561633, 0.3252594, 0.75613514, 1.18701087, 1.61788661,
2.04876234]),
<a list of 10 Patch objects>)
```



```
In [12]: # Lets make another dataset
dataset2 = randn(80)

#Plot
plt.hist(dataset2,color='indianred')
```

```
Out[12]: (array([ 1.,  4., 10., 14., 19., 18.,  7.,  4.,  2.,  1.]),
array([-2.60686288, -2.04309819, -1.47933349, -0.9155688 , -0.3518041 ,
        0.21196059,  0.77572529,  1.33948999,  1.90325468,  2.46701938,
        3.03078407]),
<a list of 10 Patch objects>)
```

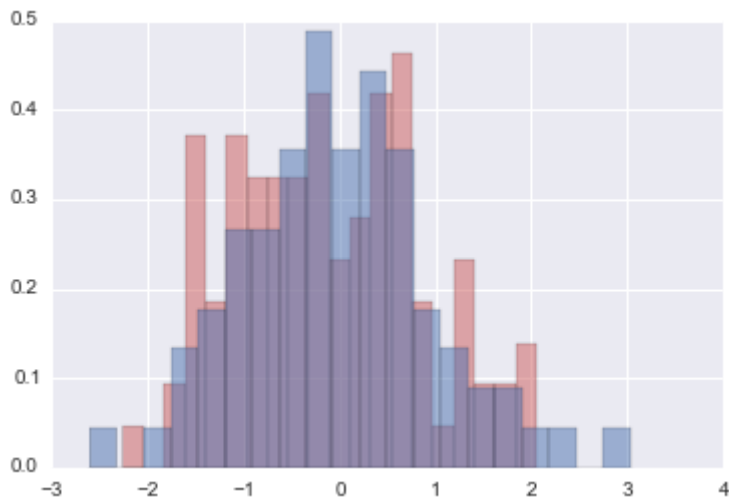


```
In [14]: # We can use normed to plot on same plot

# Set normed=True for the plots to be normalized in order to compare data sets wi
# Set alpha=0.5 for transparency

plt.hist(dataset1, normed=True, color='indianred', alpha=0.5, bins=20)
plt.hist(dataset2, normed=True, alpha=0.5, bins=20)
```

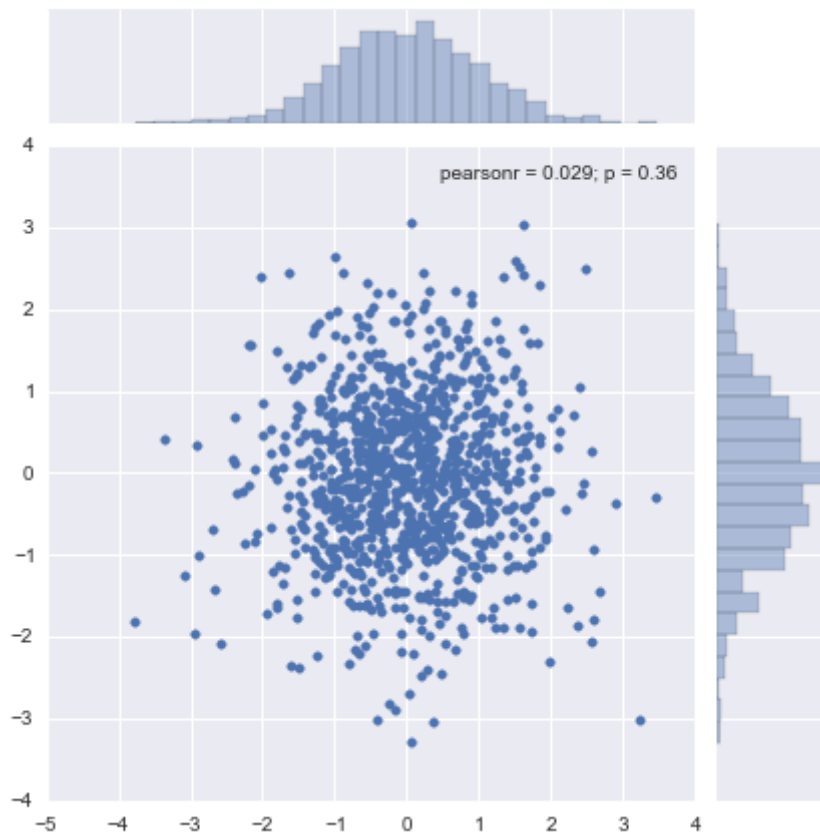
```
Out[14]: (array([ 0.04434474,  0.          ,  0.04434474,  0.13303423,  0.17737897,
                  0.26606845,  0.26606845,  0.35475794,  0.48779216,  0.35475794,
                  0.44344742,  0.35475794,  0.17737897,  0.13303423,  0.08868948,
                  0.08868948,  0.04434474,  0.04434474,  0.          ,  0.04434474]),
array([-2.60686288, -2.32498053, -2.04309819, -1.76121584, -1.47933349,
        -1.19745114, -0.9155688 , -0.63368645, -0.3518041 , -0.06992175,
         0.21196059,  0.49384294,  0.77572529,  1.05760764,  1.33948999,
         1.62137233,  1.90325468,  2.18513703,  2.46701938,  2.74890172,
         3.03078407]),
<a list of 20 Patch objects>)
```



```
In [31]: # Make two more random normal dist data sets
data1 = randn(1000)
data2 = randn(1000)

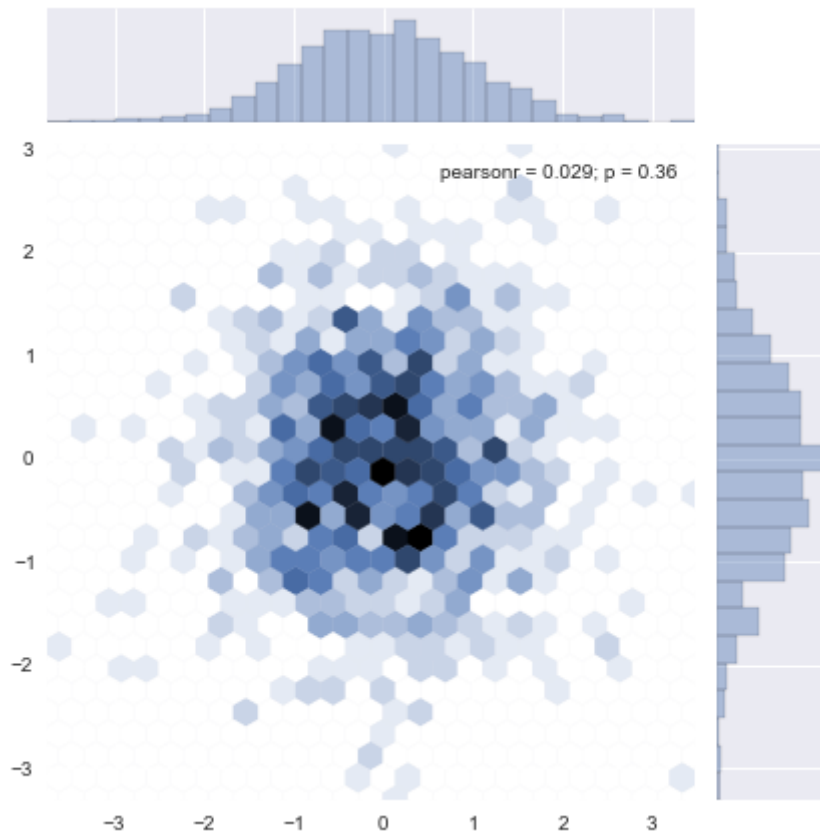
#Can represent joint distributions using joint plots
sns.jointplot(data1,data2)
```

Out[31]: <seaborn.axisgrid.JointGrid at 0x1f5a5d68>



```
In [32]: # Can also use hex bins for a more concise picture
sns.jointplot(data1,data2,kind='hex')
```

```
Out[32]: <seaborn.axisgrid.JointGrid at 0x2103ef98>
```



```
In [33]: # Next we'll learn how to use Kernel Estimation Plots
```

```
In [ ]:
```