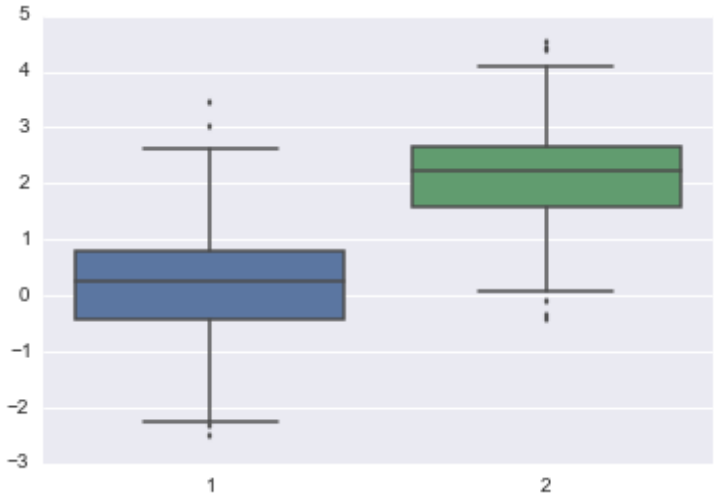
In [1]:
```
# The normal imports
import numpy as np
from numpy.random import randn
import pandas as pd

# Import the stats library from numpy
from scipy import stats

# These are the plotting modules adn 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
```

In [2]:
```
# Now we'll learn about box and violin plots
url = 'http://en.wikipedia.org/wiki/Box_plot#mediaviewer/File:Boxplot_vs_PDF.svg'

# Let's create two distributions
data1 = randn(100)
data2 = randn(100) + 2 # Off set the mean
```

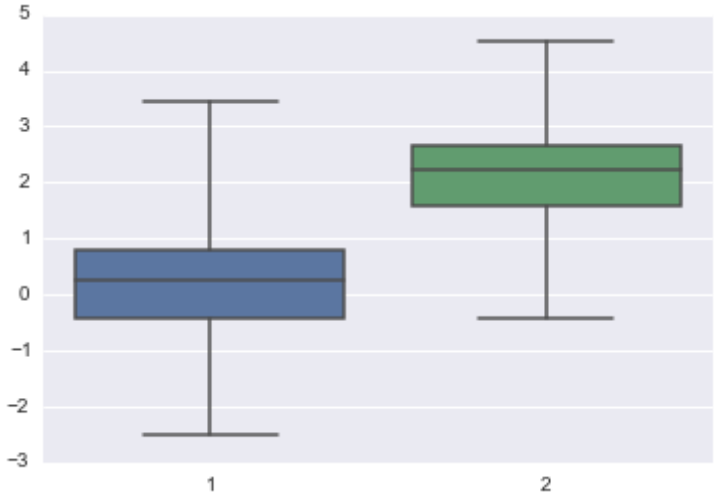
In [4]:
```
# Now we can create a box plot
sns.boxplot([data1,data2])
```

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x191d3b38>



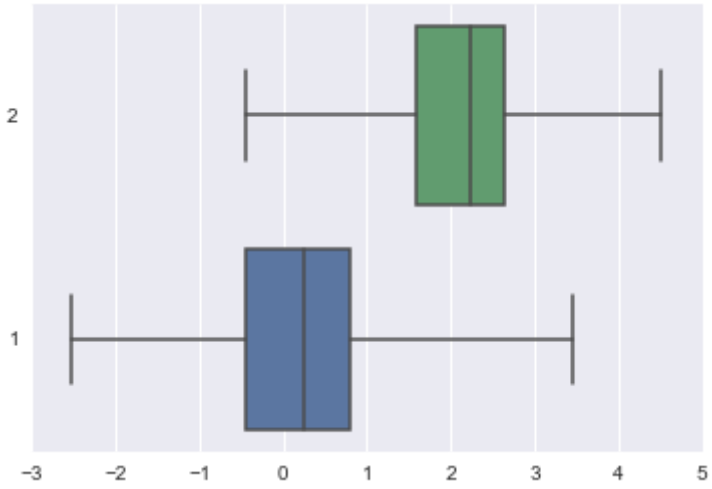
In [5]:
```
# Notice how the previous plot had outlier points, we can include those with the
sns.boxplot([data1,data2],whis=np.inf)
```

Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x19549240>



In [6]:
```
# WE can also set horizontal by setting vertical to false
sns.boxplot([data1,data2],whis=np.inf, vert = False)
```

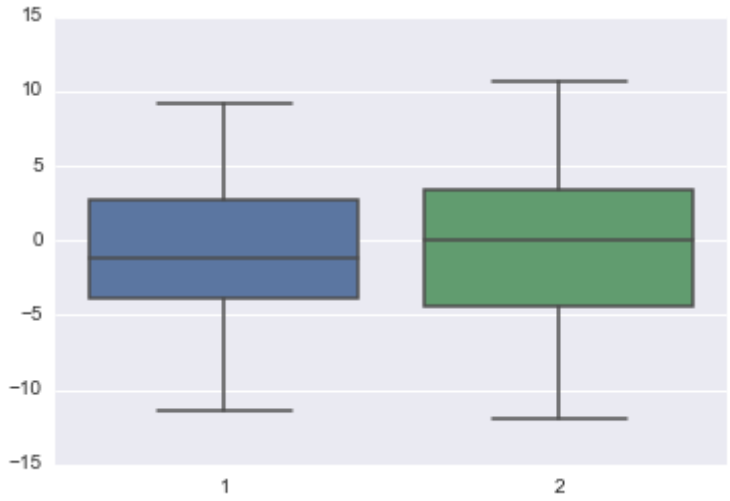
Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x195ae128>



In [7]:
```
# While box plots are great, they can sometimes not give the full picture

# Violin/Viola plots can combine the simplicity of a box plot with the informatio
```

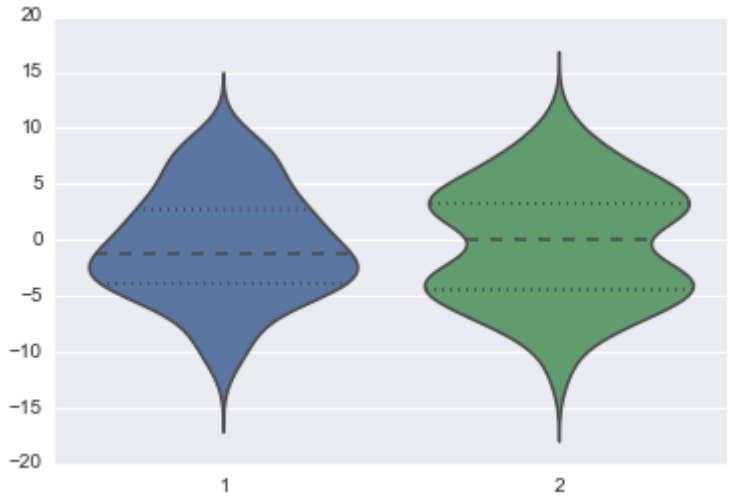
In [39]:
```
# Let's create an example where a box plot doesn't give the whole picture

# Normal Distribution
data1 = stats.norm(0,5).rvs(100)

# Two gamma distributions concatenated together (Second one is inverted)
data2 = np.concatenate([stats.gamma(5).rvs(50)-1,
                        -1*stats.gamma(5).rvs(50)])

# Box plot them
sns.boxplot([data1,data2],whis=np.inf)
```

Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x1fe246a0>



In [40]:
```
# From the above plots, you may think that the distributions are fairly similar
# But lets check out what a violin plot reveals
sns.violinplot([data1,data2])
```

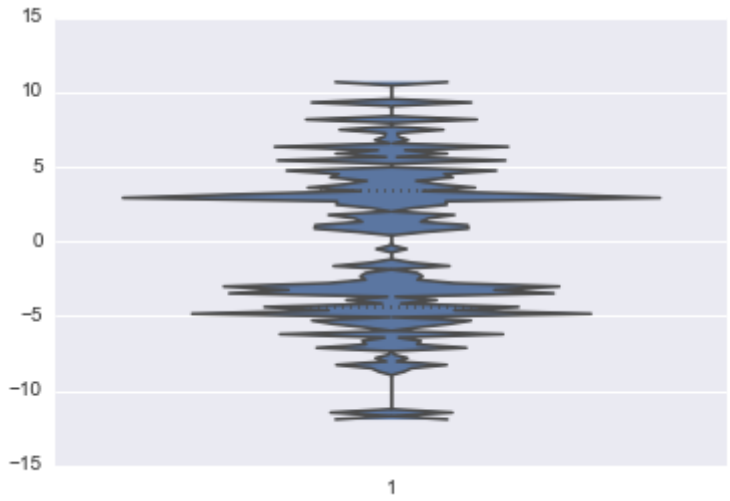
Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x203835c0>



In [41]:
```
# Wow, quite revealing!
```

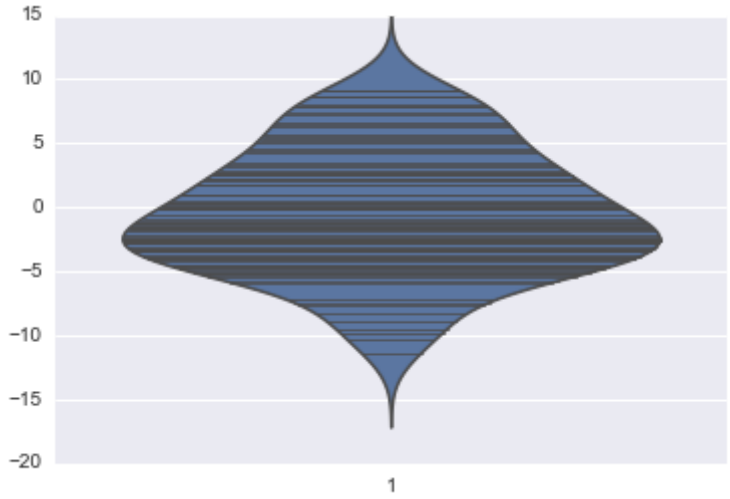
In [83]:
```
# We can also change the bandwidth of the kernel used for the density fit of the
sns.violinplot(data2,bw=0.01)
```

Out[83]: <matplotlib.axes._subplots.AxesSubplot at 0x24fdee80>



In [92]:
```
# Much like a rug plot, we can also include the individual points, or sticks
sns.violinplot(data1,inner="stick")
```

Out[92]: <matplotlib.axes._subplots.AxesSubplot at 0x25a33668>



In [ ]:
```
# Next up: Multiple Regression Plots!
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: