

# seaborn-tut-control-fig-aesthetics

December 25, 2015

## 1 Seaborn Tutorial

## 2 Style management

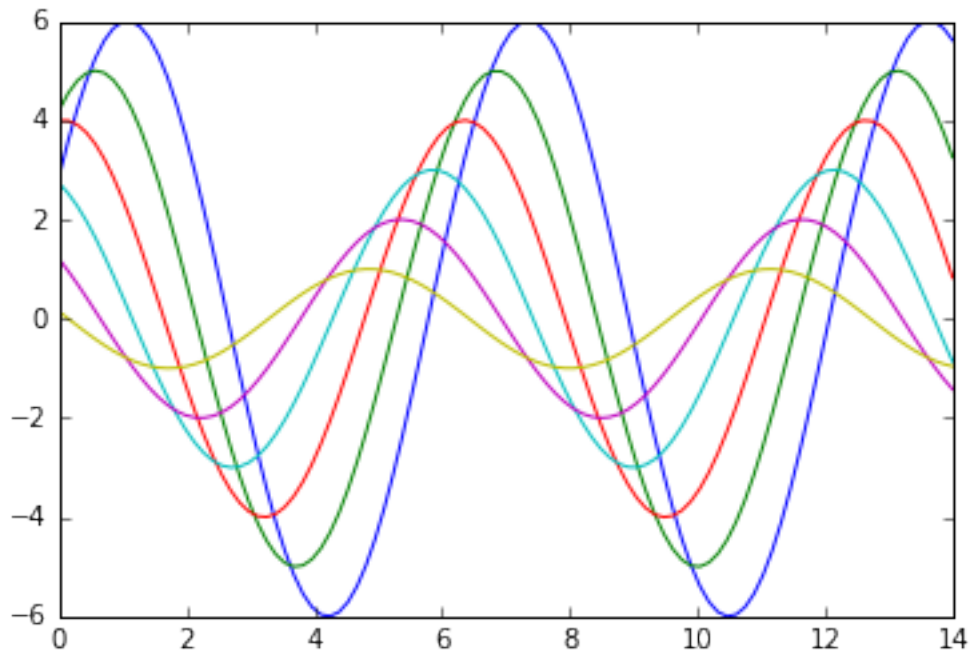
### 2.1 Controlling figure aesthetics

<http://stanford.edu/~mwaskom/software/seaborn/tutorial/aesthetics.html>

```
In [1]: import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
np.random.seed(sum(map(ord, 'aesthetics'))))

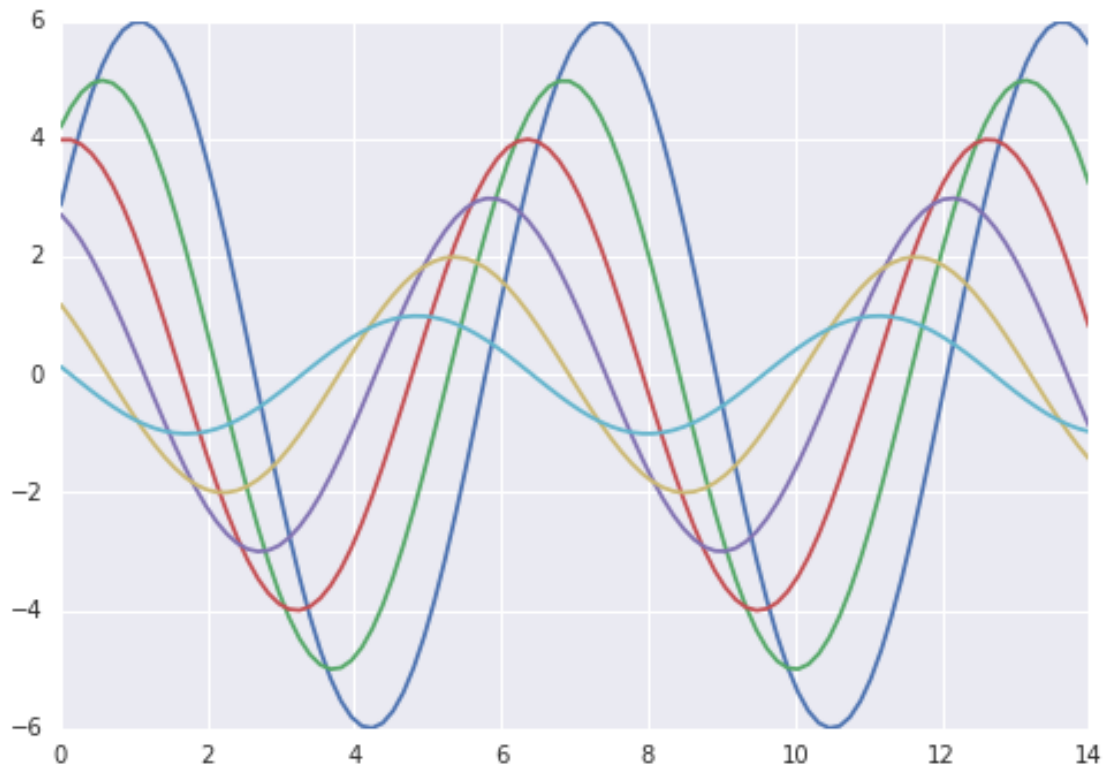
In [2]: def sinplot(flip=1):
x = np.linspace(0, 14, 100)
for i in range(1, 7):
    plt.plot(x, np.sin(x + i * .5) * (7 - i) * flip)

In [3]: sinplot()
```



```
In [4]: import seaborn as sns
        sinplot()
```

```
/Users/bartev/.virtualenvs/sbrn/lib/python2.7/site-packages/matplotlib/_init_.py:872: UserWarning: axes
warnings.warn(self.msg_depr % (key, alt_key))
```



### 2.1.1 Styling figures with `axes_style()` and `set_style()`

```
In [5]: sns.plotting_context()
```

```
Out[5]: {'axes.labelsize': 11.0,
         'axes.titlesize': 12.0,
         'figure.figsize': [8.0, 5.5],
         'grid.linewidth': 1.0,
         'legend.fontsize': 10.0,
         'lines.linewidth': 1.75,
         'lines.markeredgewidth': 0.0,
         'lines.markersize': 7.0,
         'patch.linewidth': 0.3,
         'xtick.labelsize': 10.0,
         'xtick.major.pad': 7.0,
         'xtick.major.width': 1.0,
         'xtick.minor.width': 0.5,
         'ytick.labelsize': 10.0,
         'ytick.major.pad': 7.0,
         'ytick.major.width': 1.0,
         'ytick.minor.width': 0.5}
```

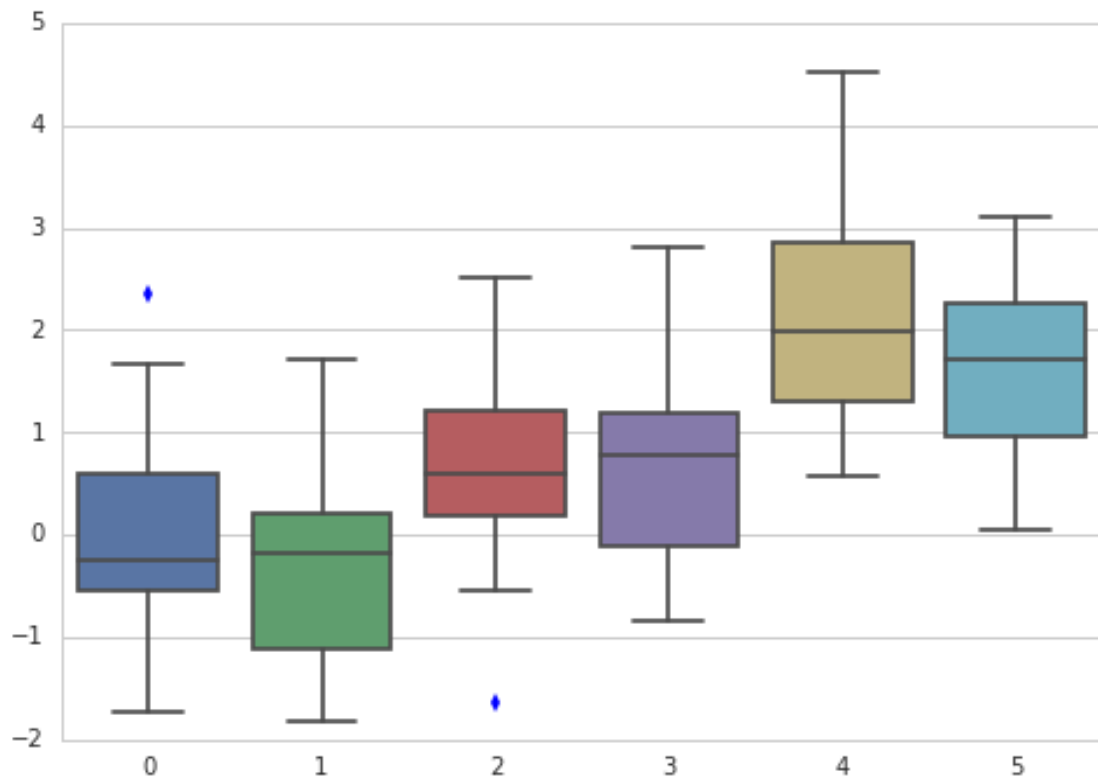
```
In [6]: sns.axes_style()
```

```
Out[6]: {'axes.axisbelow': True,
'axes.edgecolor': 'white',
'axes.facecolor': '#EAEAF2',
'axes.grid': True,
'axes.labelcolor': '.15',
'axes.linewidth': 0.0,
'figure.facecolor': 'white',
'font.family': [u'sans-serif'],
'font.sans-serif': [u'Arial',
u'Liberation Sans',
u'Bitstream Vera Sans',
u'sans-serif'],
'grid.color': 'white',
'grid.linestyle': u'-',
'image.cmap': u'Greys',
'legend.frameon': False,
'legend.numpoints': 1,
'legend.scatterpoints': 1,
'lines.solid_capstyle': u'round',
'text.color': '.15',
'xtick.color': '.15',
'xtick.direction': u'out',
'xtick.major.size': 0.0,
'xtick.minor.size': 0.0,
'ytick.color': '.15',
'ytick.direction': u'out',
'ytick.major.size': 0.0,
'ytick.minor.size': 0.0}
```

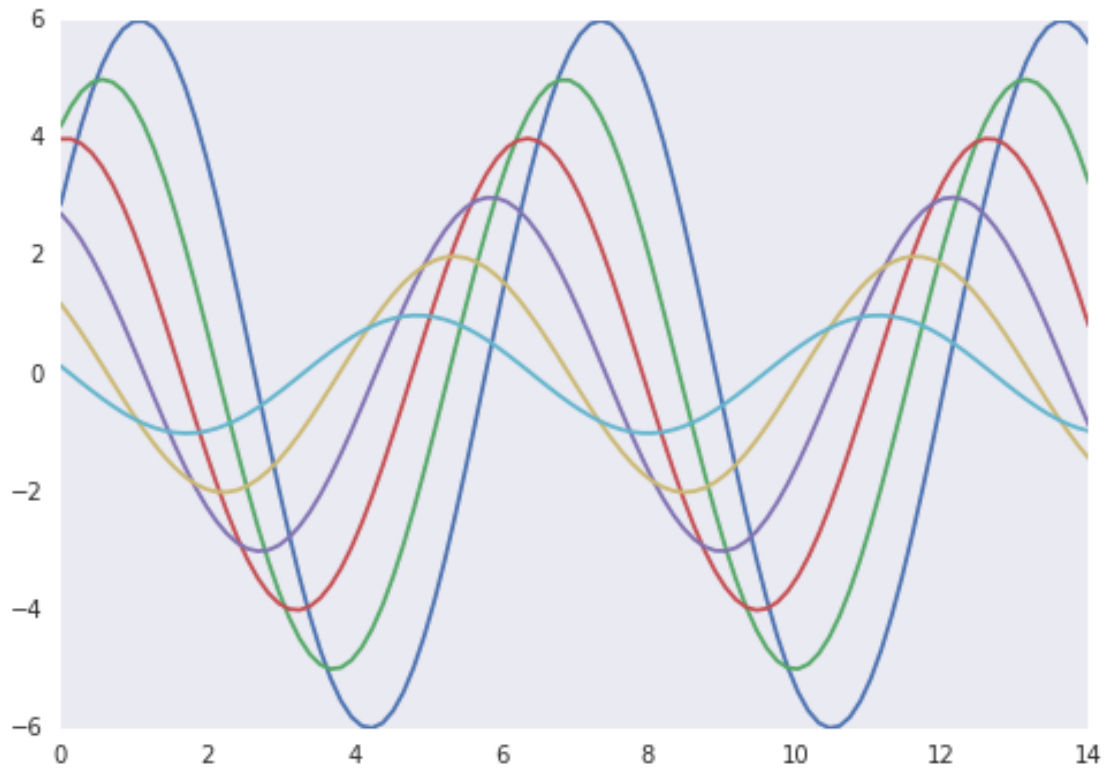
```
In [7]: sns.set_style('whitegrid')
data = np.random.normal(size = (20, 6)) + np.arange(6) / 2
sns.boxplot(data = data)
```

```
/Users/bartev/.virtualenvs/sbrn/lib/python2.7/site-packages/matplotlib/__init__.py:892: UserWarning: axes
warnings.warn(self.msg_depr % (key, alt_key))
```

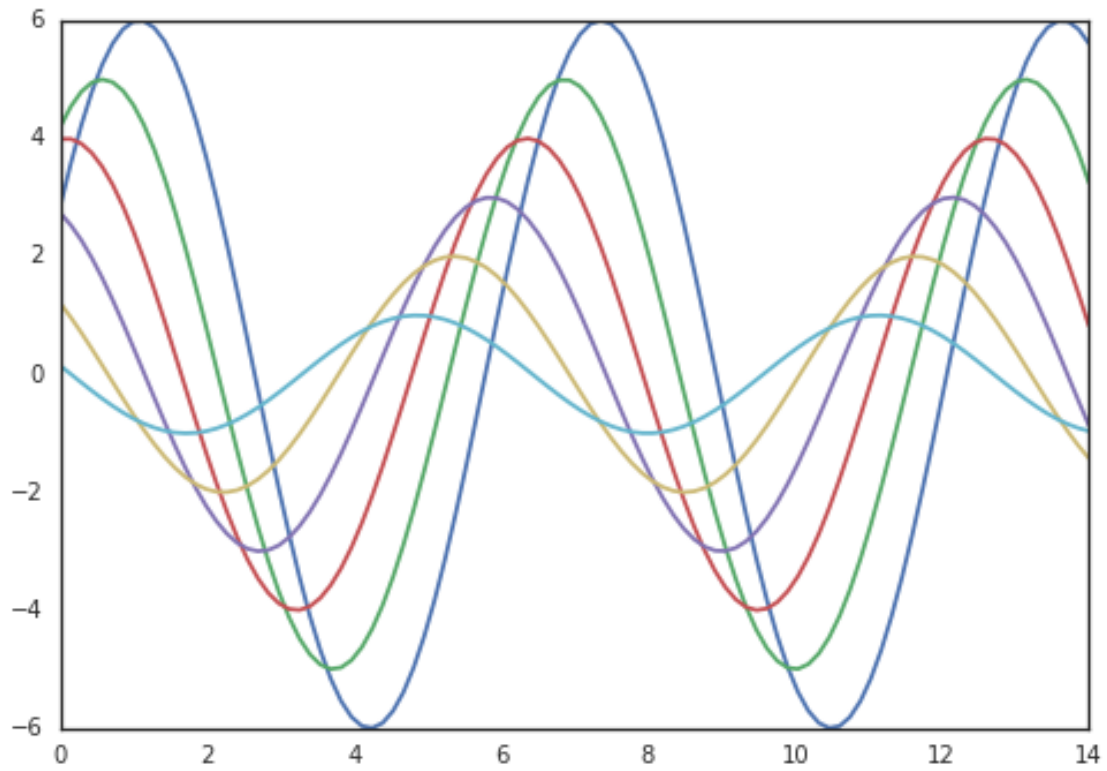
```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x10c33eb50>
```



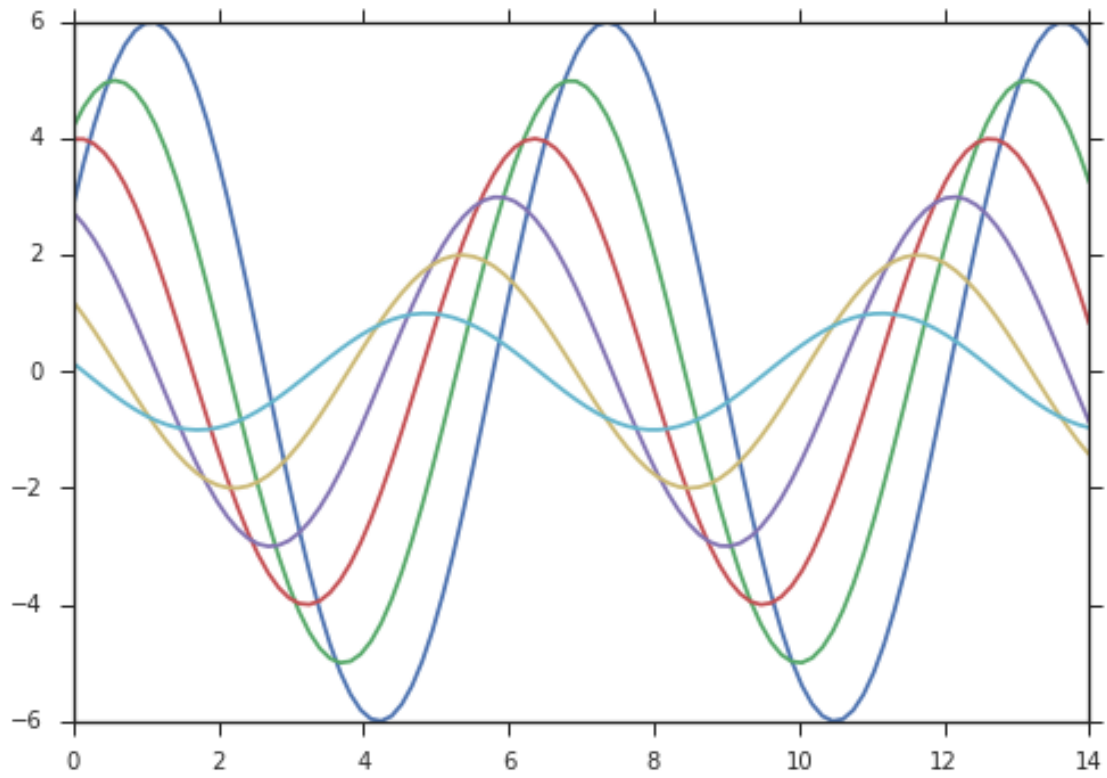
```
In [8]: sns.set_style('dark')  
        sinplot()
```



```
In [9]: sns.set_style('white')  
        sinplot()
```

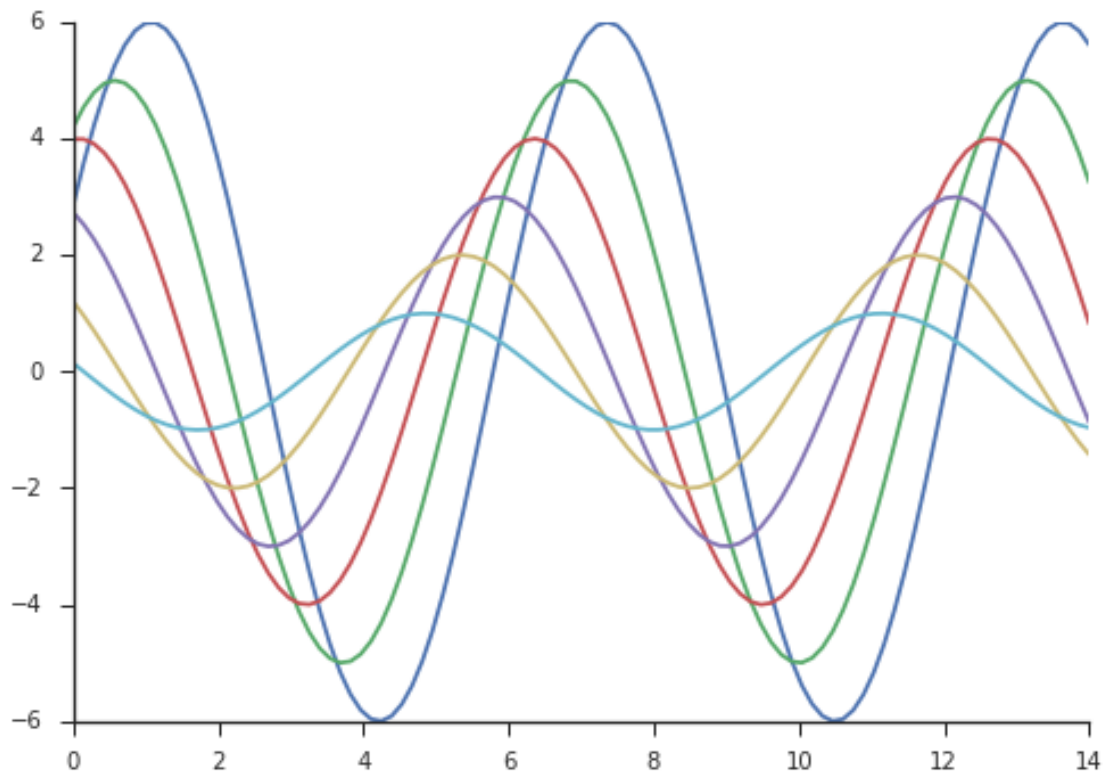


```
In [10]: sns.set_style('ticks')  
         sinplot()
```



### 2.1.2 Remove spines with `despine()`

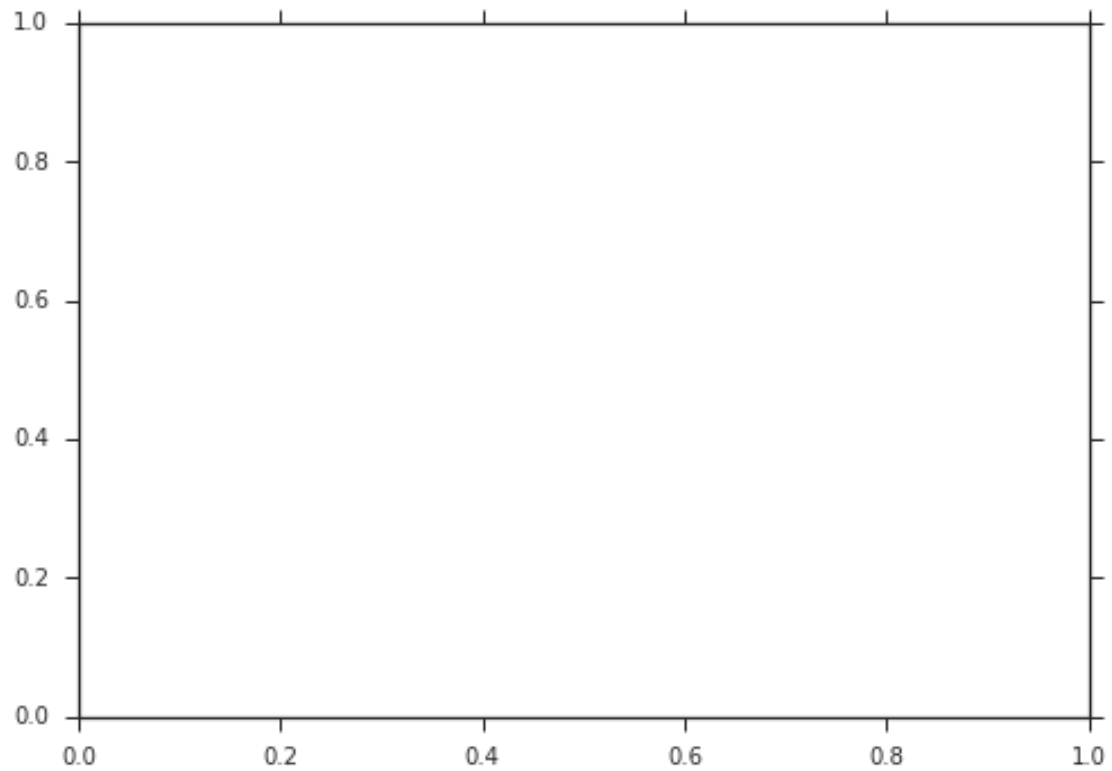
```
In [11]: sinplot()  
sns.despine()
```



Offset spines away from data

```
In [12]: f, ax = plt.subplots()
```





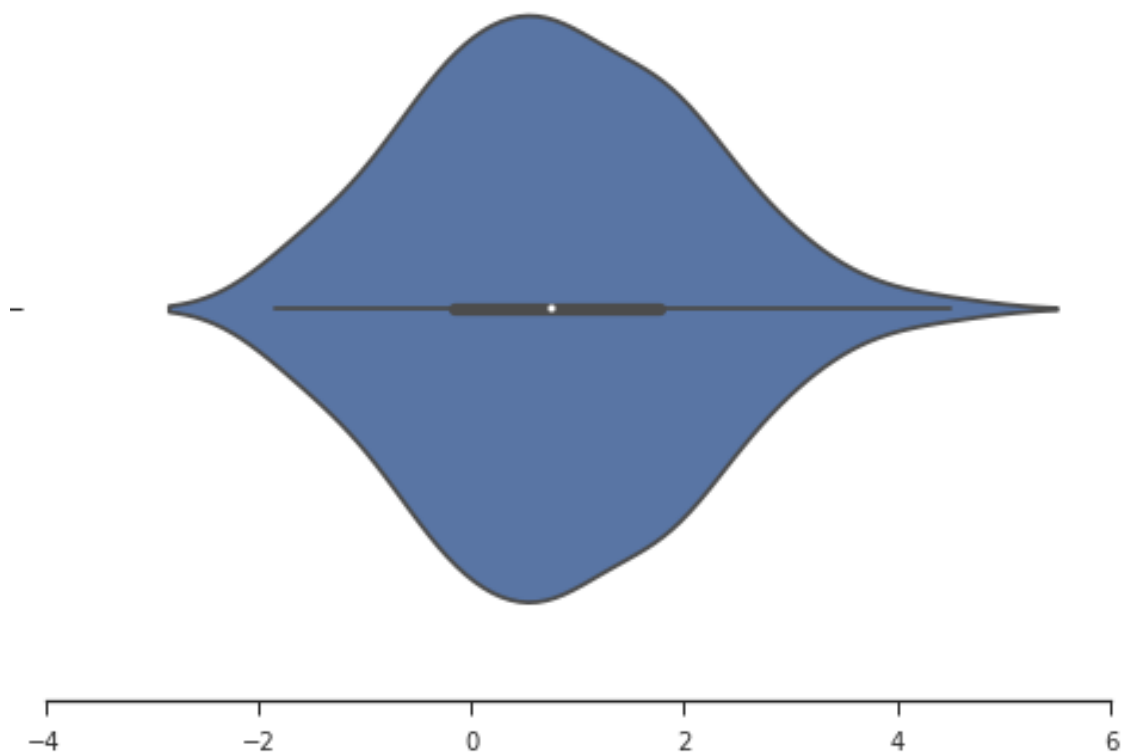
```
In [13]: type(f)
```

```
Out[13]: matplotlib.figure.Figure
```

```
In [14]: type(ax)
```

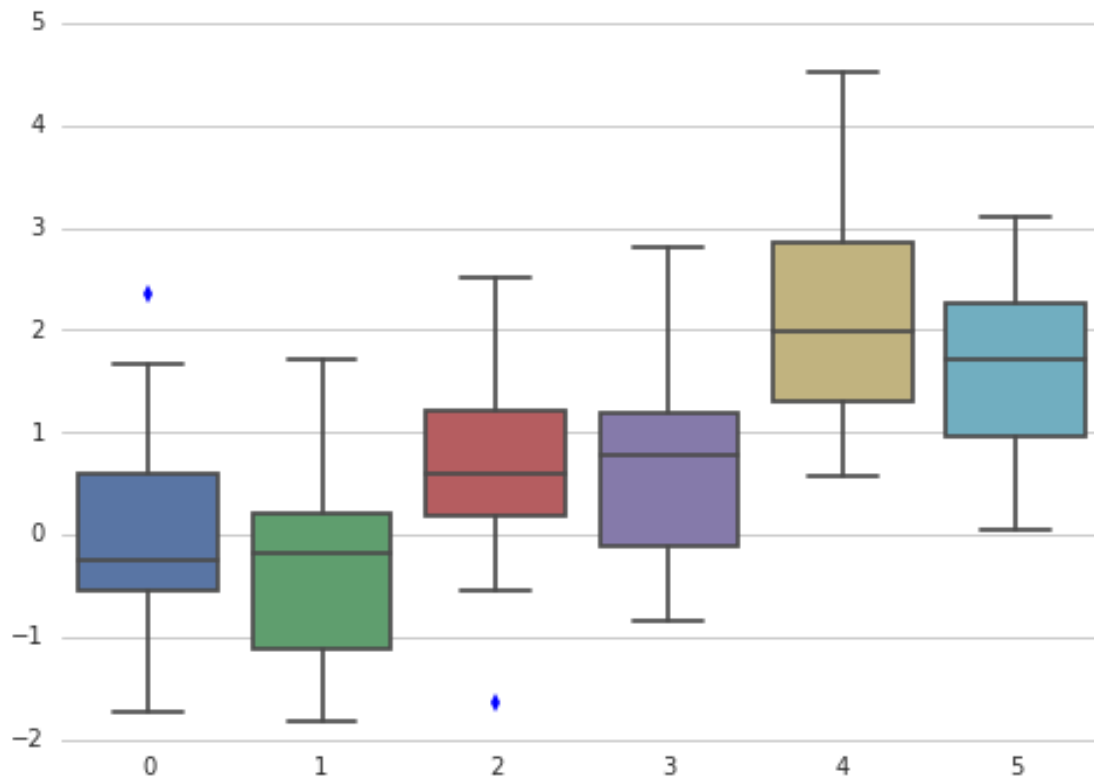
```
Out[14]: matplotlib.axes._subplots.AxesSubplot
```

```
In [15]: sns.violinplot(data)  
sns.despine(offset = 10, trim = True)
```



Control which spines are removed

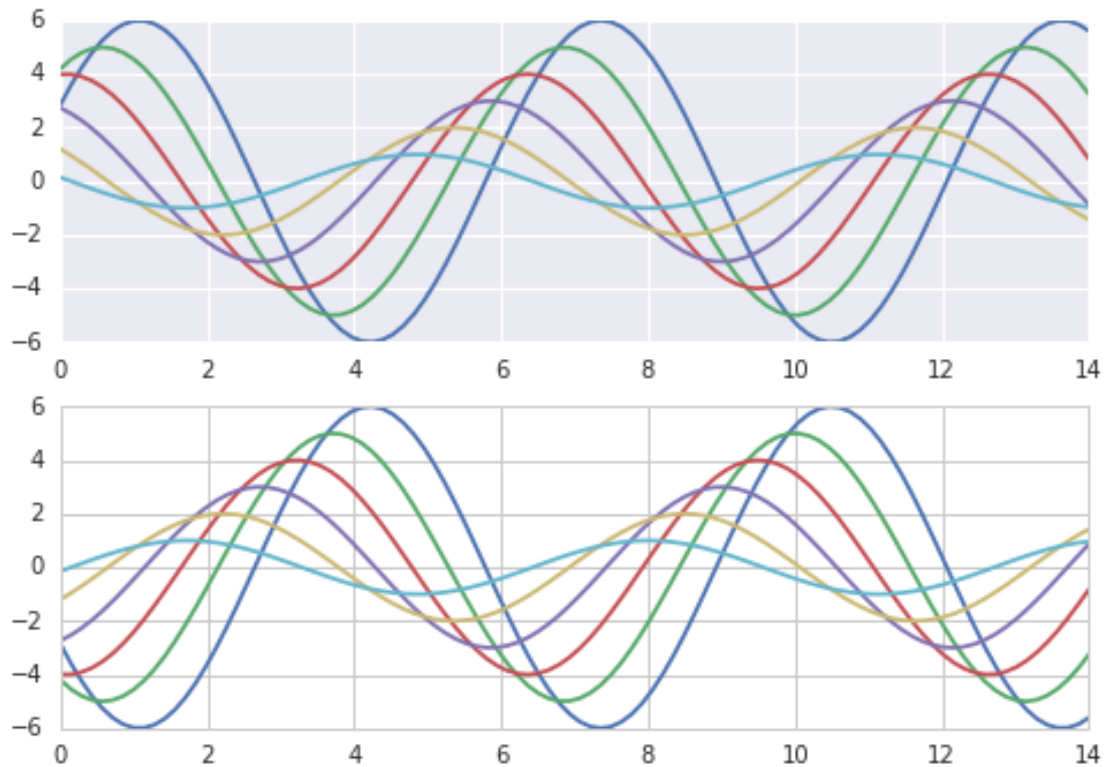
```
In [16]: sns.set_style('whitegrid')
          sns.boxplot(data = data, palette = 'deep')
          sns.despine(left = True)
```



### 2.1.3 Temporarily set figure style

use `axes_style()` in `awith` statement

```
In [17]: with sns.axes_style('darkgrid'):
          plt.subplot(211)
          sinplot()
          plt.subplot(212)
          sinplot(-1)
```



#### 2.1.4 Override elements of seaborn styles

Pass a dict of params ot the rc argument of `axes_style()` and `set_style()`

In [18]: `sns.axes_style()`

```
Out[18]: {'axes.axisbelow': True,
          'axes.edgecolor': '.8',
          'axes.facecolor': 'white',
          'axes.grid': True,
          'axes.labelcolor': '.15',
          'axes.linewidth': 1.0,
          'figure.facecolor': 'white',
          'font.family': [u'sans-serif'],
          'font.sans-serif': [u'Arial',
                              u'Liberation Sans',
                              u'Bitstream Vera Sans',
                              u'sans-serif'],
          'grid.color': '.8',
          'grid.linestyle': u'-',
          'image.cmap': u'Greys',
          'legend.frameon': False,
          'legend.numpoints': 1,
          'legend.scatterpoints': 1,
          'lines.solid_capstyle': u'round',
          'text.color': '.15',
```

```

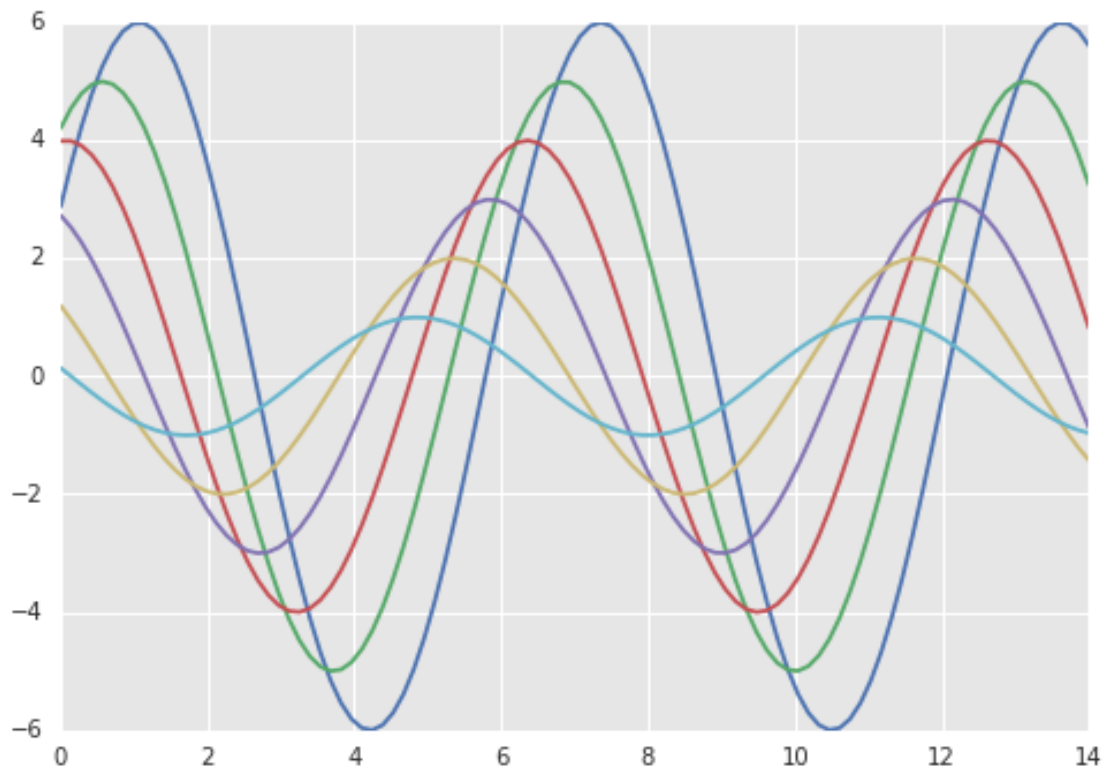
'xtick.color': '.15',
'xtick.direction': u'out',
'xtick.major.size': 0.0,
'xtick.minor.size': 0.0,
'ytick.color': '.15',
'ytick.direction': u'out',
'ytick.major.size': 0.0,
'ytick.minor.size': 0.0}

```

```

In [19]: sns.set_style(
        'darkgrid',
        {'axes.facecolor' : '0.9'})
sinplot()

```



### 2.1.5 Scale plot elements with `plotting_context()` and `set_context()`

Reset the default parameters

```

In [20]: sns.set()

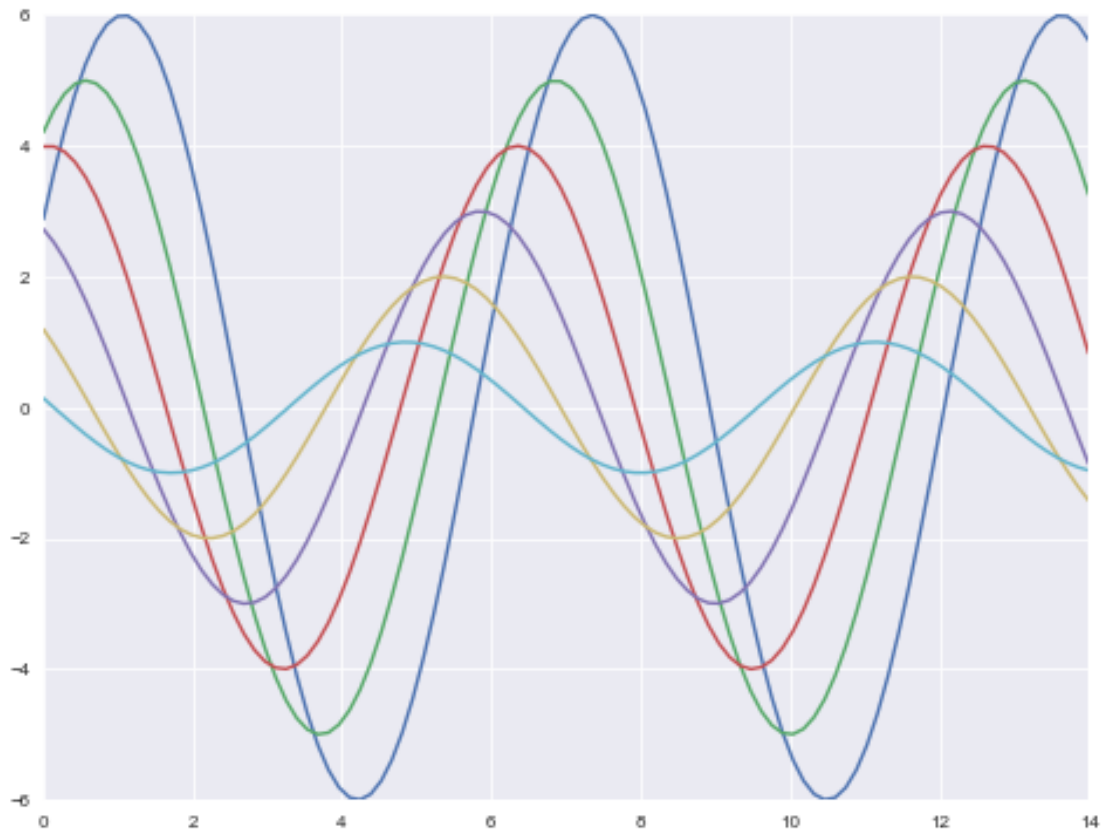
```

4 preset contexts: \* paper \* notebook \* talk \* poster

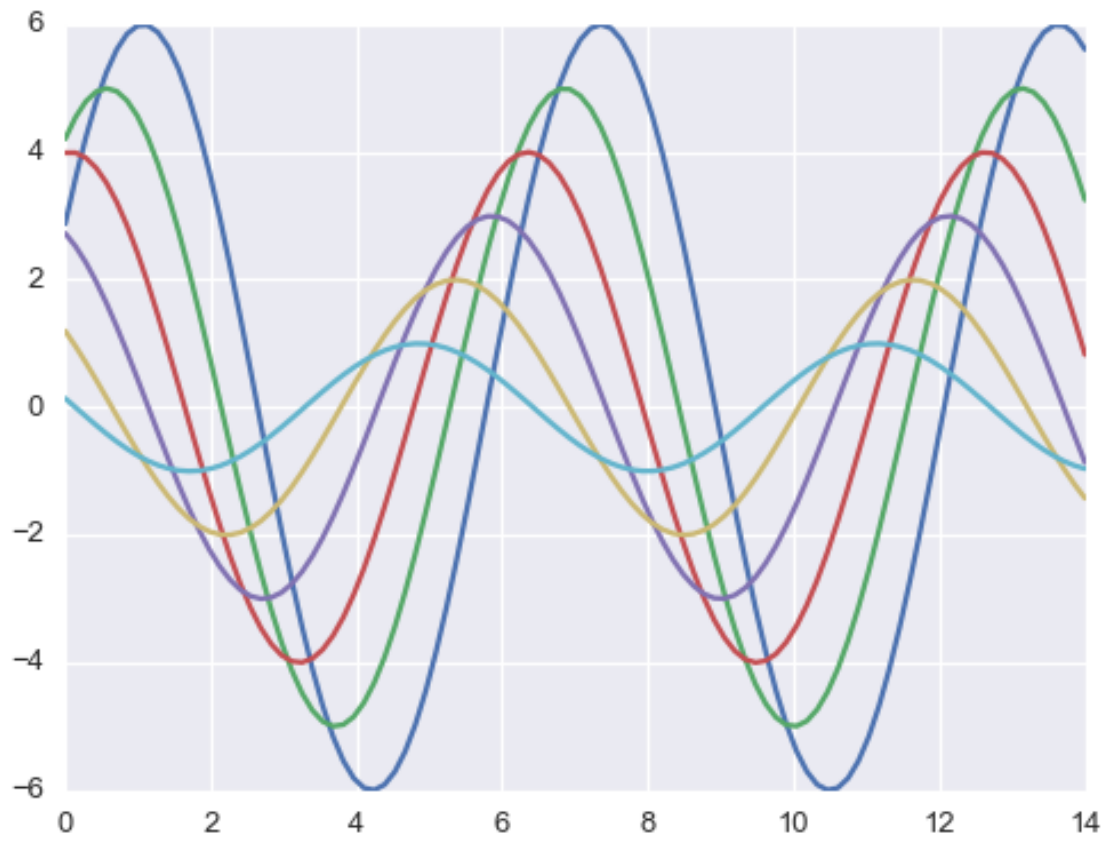
```

In [21]: sns.set_context('paper')
plt.figure(figsize = (8, 6))
sinplot()

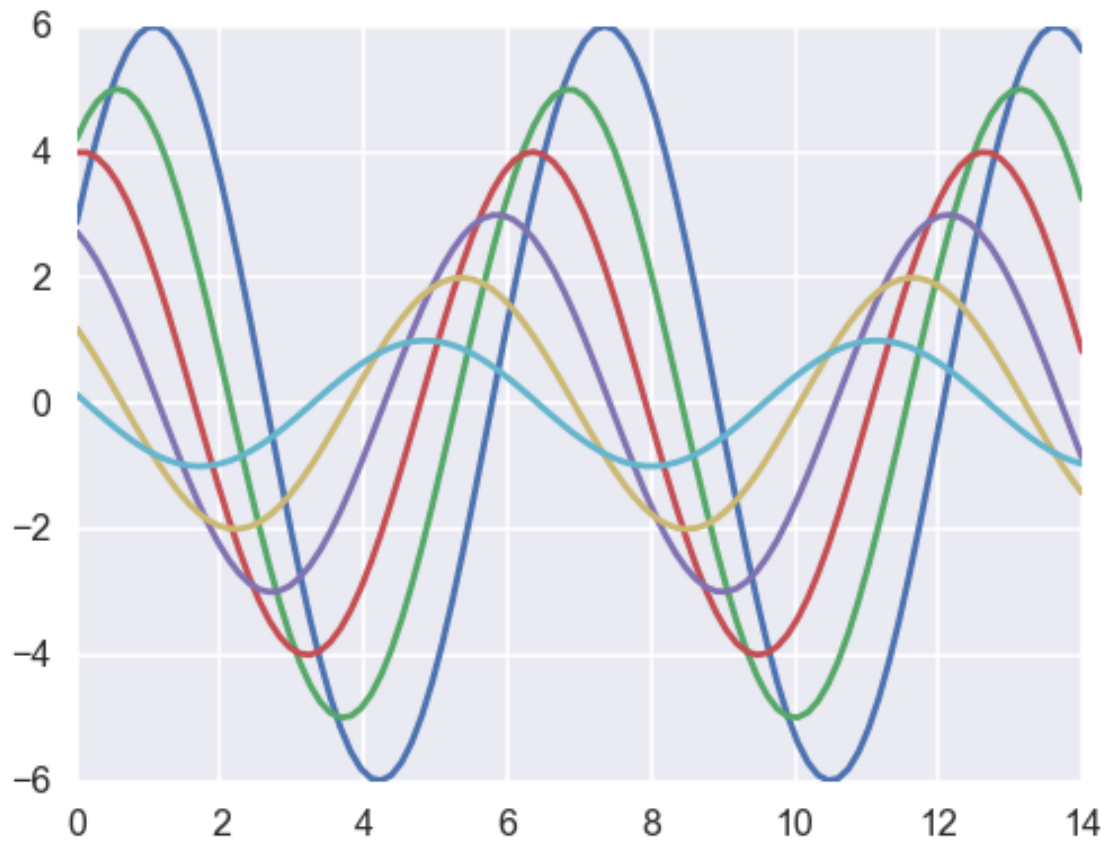
```



```
In [22]: sns.set_context('talk')  
plt.figure(figsize = (8, 6))  
sinplot()
```



```
In [23]: sns.set_context('poster')  
plt.figure(figsize = (8, 6))  
sinplot()
```



```
In [24]: sns.set_context(  
        'notebook',  
        font_scale = 1.5,  
        rc = {'lines.linewidth' : 2.5})  
sinplot()
```



