In [21]:

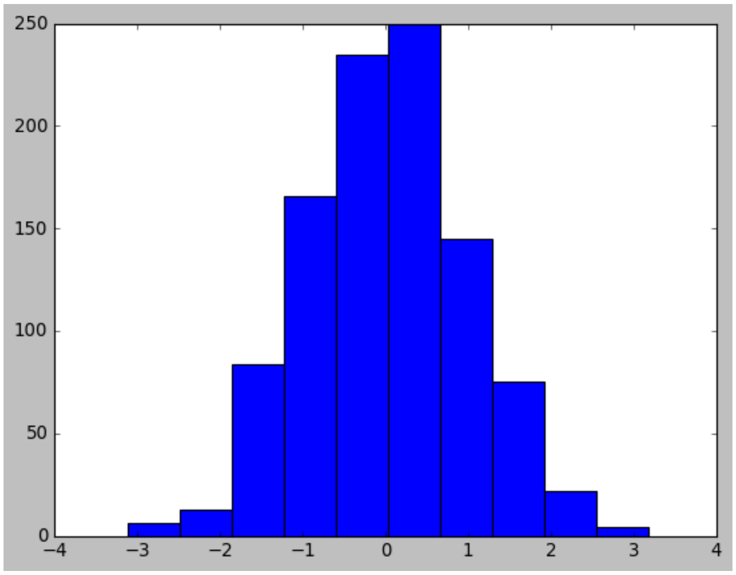
```
#plot customisation by hand
import matplotlib.pyplot as plt
plt.style.use('classic')
import numpy as np
%matplotlib inline
```

In [22]:

```
x=np.random.randn(1000)
plt.hist(x)
```

Out[22]:

```
(array([  6.,  13.,  84., 166., 235., 250., 145.,  75.,  22.,   4.]),
 array([-3.11685659, -2.48807346, -1.85929032, -1.23050718, -0.60172405,
         0.02705909,  0.65584223,  1.28462536,  1.9134085 ,  2.54219164,
         3.17097477]),
 <BarContainer object of 10 artists>)
```

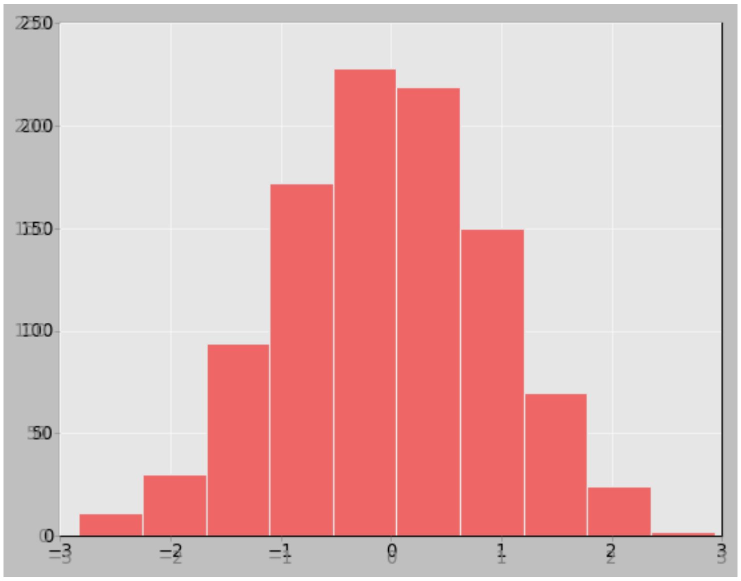


In [23]:

```
x=np.random.randn(1000)
plt.hist(x)
#setting background
ax=plt.axes()
ax.set_facecolor("#E6E6E6")
ax.set_axisbelow(True)
#drawing and changing grid lines
plt.grid(color="w",linestyle="solid")
#hiding axis spines
for spines in ax.spines.values():
    spines.set_visible(False)
# hide top and right ticks
ax.xaxis.tick_bottom()
ax.yaxis.tick_left()
#modifying ticks
ax.tick_params(colors="gray",direction="out")
for tick in ax.get_xticklabels():
    tick.set_color("gray")
for tick in ax.get_yticklabels():
    tick.set_color("gray")
#manipulating face and edge color of histogram
ax.hist(x,edgecolor="#E6E6E6",color="#EE6666")
```

Out[23]:

```
(array([ 11.,  30.,  94., 172., 228., 219., 150.,  70.,  24.,   2.]),
 array([-2.83201187, -2.25590106, -1.67979025, -1.10367944, -0.52756863,
         0.04854219,  0.624653  ,  1.20076381,  1.77687462,  2.35298543,
         2.92909624]),
 <BarContainer object of 10 artists>)
```



In [24]:

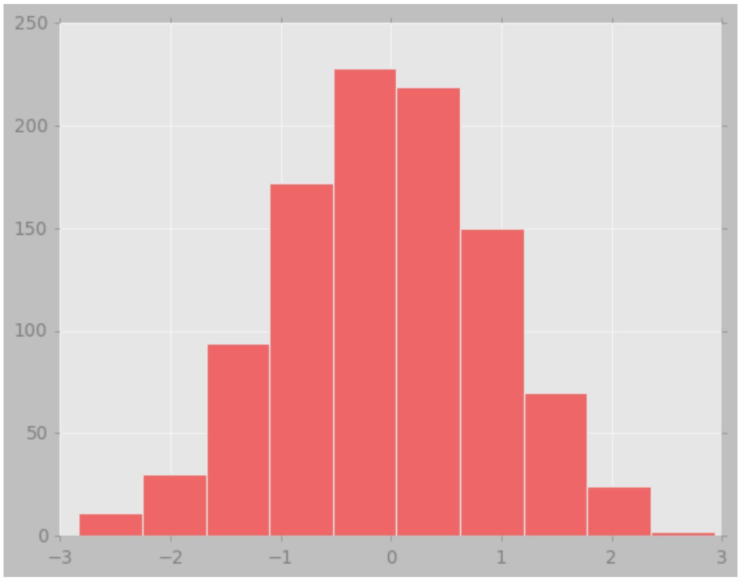
```
#changing defaults using rcParams
IPython_default=plt.rcParams.copy()#saving default settings
from matplotlib import cycler
colors=cycler("color",["#EE6666","#3388BB","#9988DD","#EECC55","#88BB44","#FFBBBB"])
#we use cycler to cycle through a set of colors when plotting graph
plt.rc("axes",facecolor="#E6E6E6",axisbelow=True,edgecolor="none",grid=True,prop_cycle=colors)
plt.rc("grid",color="w",linestyle="solid")
plt.rc("xtick",direction="out",color="gray")
plt.rc("ytick",direction="out",color="gray")
plt.rc("patch",edgecolor="#E6E6E6")
plt.rc("lines",linewidth=2)
```

In [25]:

```
plt.hist(x)
```

Out[25]:

```
(array([ 11.,  30.,  94., 172., 228., 219., 150.,  70.,  24.,   2.]),
 array([-2.83201187, -2.25590106, -1.67979025, -1.10367944, -0.52756863,
         0.04854219,  0.624653  ,  1.20076381,  1.77687462,  2.35298543,
         2.92909624]),
 <BarContainer object of 10 artists>)
```



In [26]:

```
#stylesheets
plt.style.available
```

Out[26]:

```
['Solarize_Light2',
 '_classic_test_patch',
 '_mpl-gallery',
 '_mpl-gallery-nogrid',
 'bmh',
 'classic',
 'dark_background',
 'fast',
 'fivethirtyeight',
 'ggplot',
 'grayscale',
 'seaborn',
 'seaborn-bright',
 'seaborn-colorblind',
 'seaborn-dark',
 'seaborn-dark-palette',
 'seaborn-darkgrid',
 'seaborn-deep',
 'seaborn-muted',
 'seaborn-notebook',
 'seaborn-paper',
 'seaborn-pastel',
 'seaborn-poster',
 'seaborn-talk',
 'seaborn-ticks',
 'seaborn-white',
 'seaborn-whitegrid',
 'tableau-colorblind10']
```

In [27]:

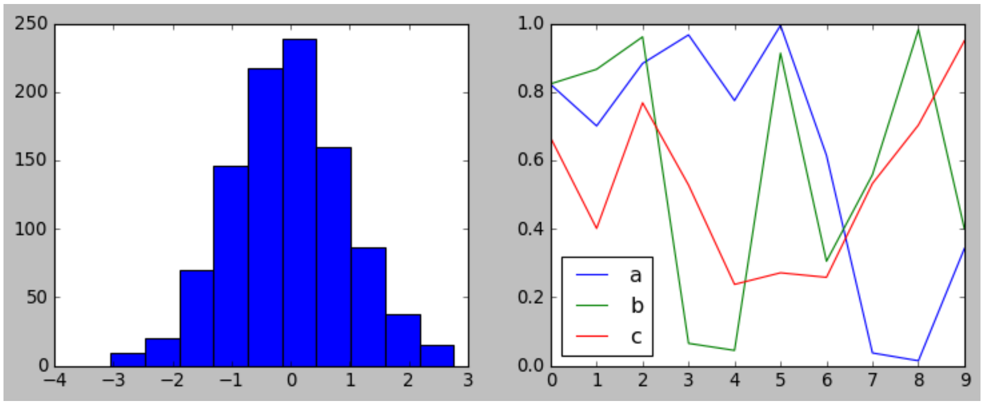
```
def hist_and_lines():
    np.random.seed(0)
    fig, ax = plt.subplots(1, 2, figsize=(11, 4))
    ax[0].hist(np.random.randn(1000))
    for i in range(3):
        ax[1].plot(np.random.rand(10))
    ax[1].legend(['a', 'b', 'c'], loc='lower left')
```

In [28]:

```
#default style
#resetting params
plt.rcParams.update(IPython_default)
```

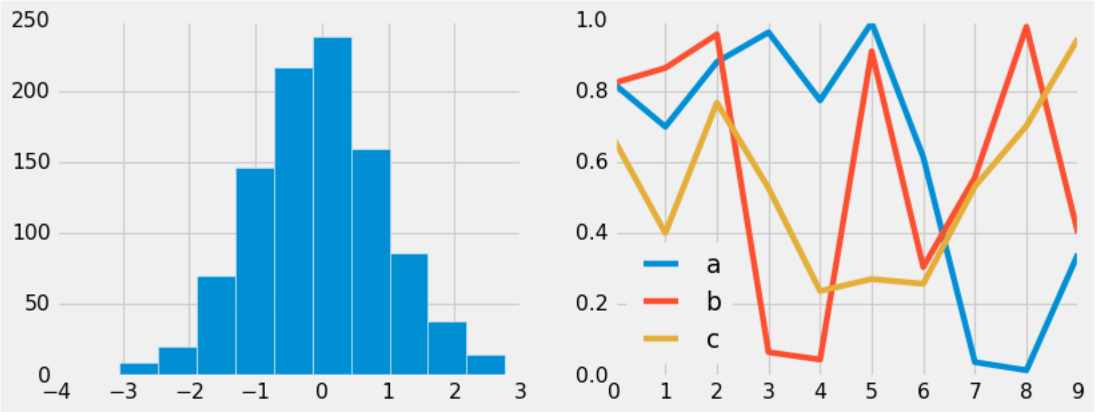
In [29]:

```
hist_and_lines()
```



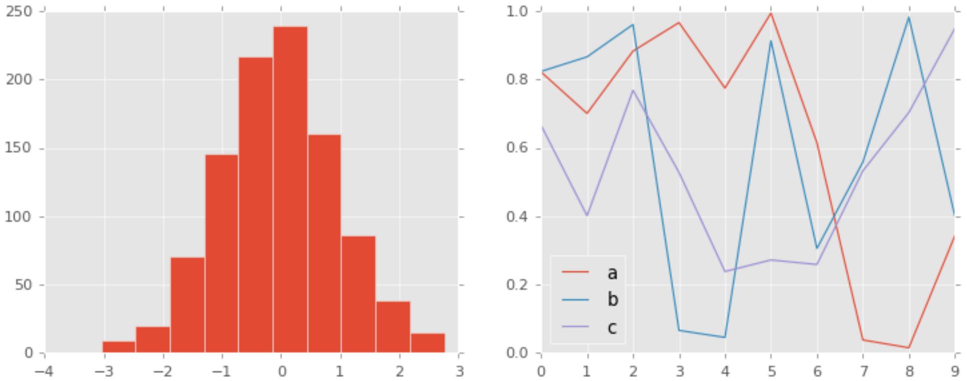
In [30]:

```
with plt.style.context("fivethirtyeight"):
    hist_and_lines()
```



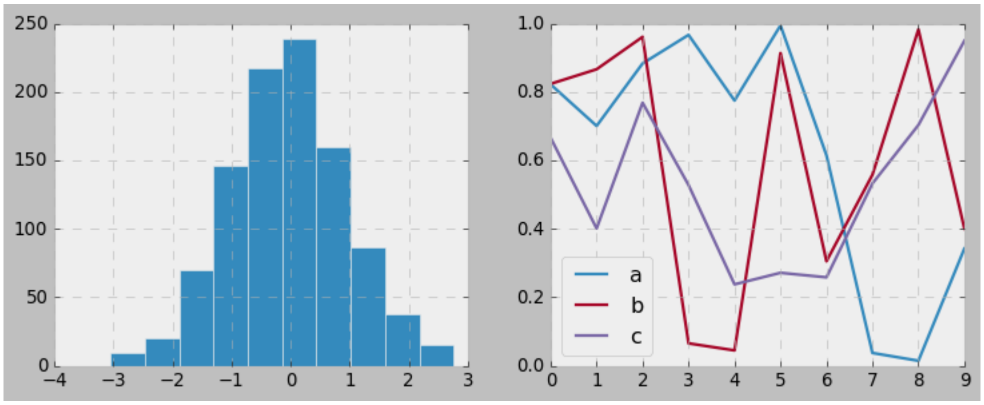
In [31]:

```
#ggplot
with plt.style.context("ggplot"):
    hist_and_lines()
```



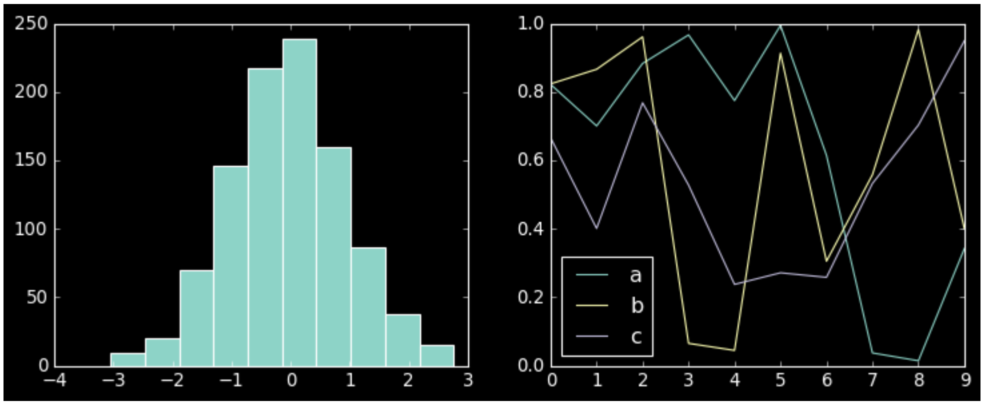
In [33]:

```
with plt.style.context("bmh"):#Baseyian methods for hackers
    hist_and_lines()
```



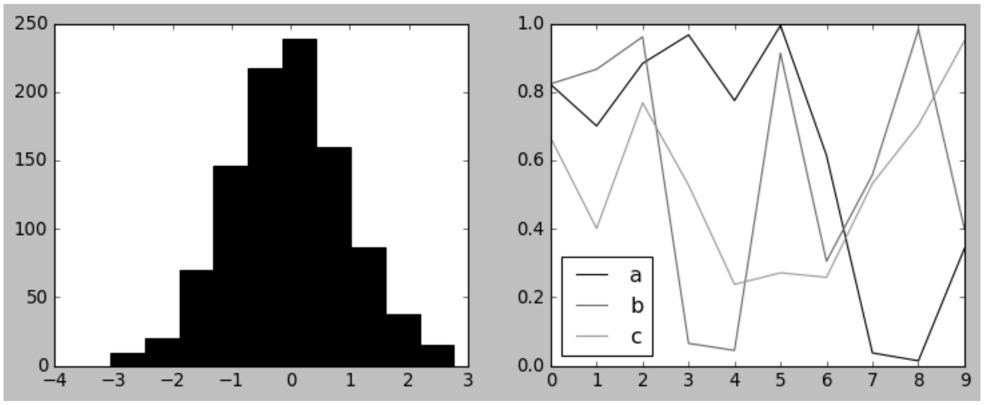
In [34]:

```
#dark_background
with plt.style.context("dark_background"):
    hist_and_lines()
```



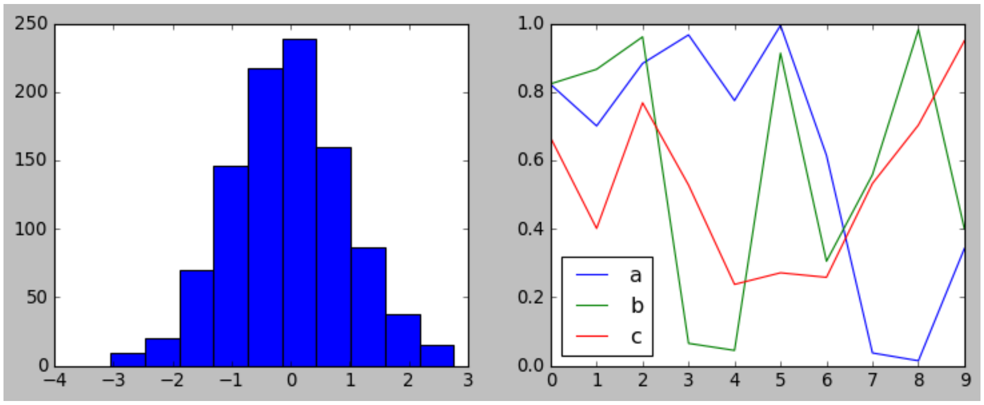
In [36]:

```
#grayscale
with plt.style.context("grayscale"):
    hist_and_lines()
```



In [37]:

```
#using seaborn
import seaborn
hist_and_lines()
```


250
200
150
100
50
0
−4
−3
−2
−1
0
1
2
3
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
9
a
b
c