MATPLOTLIB绘图

In [4]:

```
import pandas as pd
import numpy as np
```

In [7]:

```
abs = 'E:\\git\\jupyter\\text.csv'
df = pd.read_csv(abs, encoding = 'gbk')

def func(score):
    if score >= 85:
        return "优秀"
    elif score >= 75:
        return "良好"
    elif score >= 60:
        return "及格"
    else:
        return "不及格"

df['数学分类'] = df.数学.map(func)

def func1(x):
    return x.语文 + x.数学

df['new_score'] = df.apply(func1, axis = 1)
df.head()
```

Out[7]:

	序号	姓名	性别	班级	语文	数学	英语	总分	数学分类	new_score
0	1	wong	男	高一一班	89	100	99	288	优秀	189
1	2	li	女	高一一班	72	98	75	245	优秀	170
2	3	xi	女	高一一班	73	89	53	215	优秀	162
3	4	xiong	男	高一一班	84	66	75	225	及格	150
4	5	li	男	高一一班	99	77	87	263	良好	176

In [36]:

```
df = df.drop(['new_score'], axis = 1)
```

In [16]:

```
df.head()
```

Out[16]:

	序号	姓名	性别	班级	语文	数学	英语	总分	数学分类
0	1	wong	男	高一一班	89	100	99	288	优秀
1	2	li	女	高一一班	72	98	75	245	优秀
2	3	xi	女	高一一班	73	89	53	215	优秀
3	4	xiong	男	高一一班	84	66	75	225	及格
4	5	li	男	高一一班	99	77	87	263	良好

1 绘图

In [8]:

```
import numpy as np
import matplotlib.pyplot as plt
# 将图画在页面上，必不可少
%matplotlib inline
```

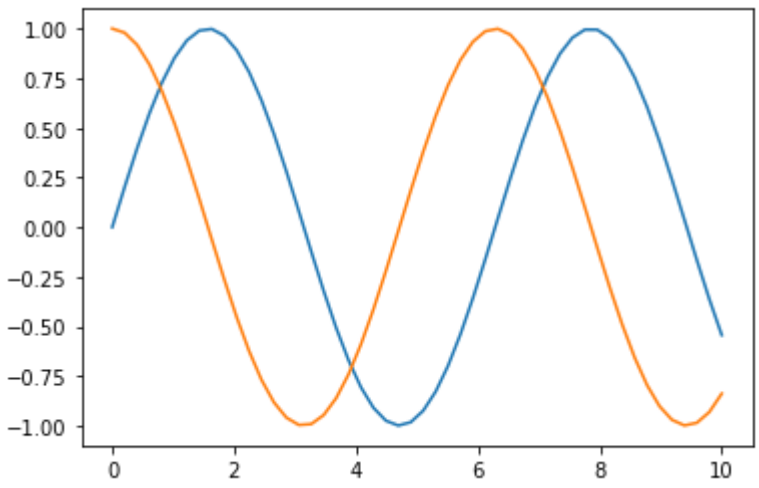
In [28]:

```
# 0 - 10 中等长生成20个数据
x = np.linspace(0, 10, 50)
y = np.sin(x)

plt.plot(x, y)
plt.plot(x, np.cos(x))
```

Out[28]:

```
[<matplotlib.lines.Line2D at 0x1f055f83fc8>]
```

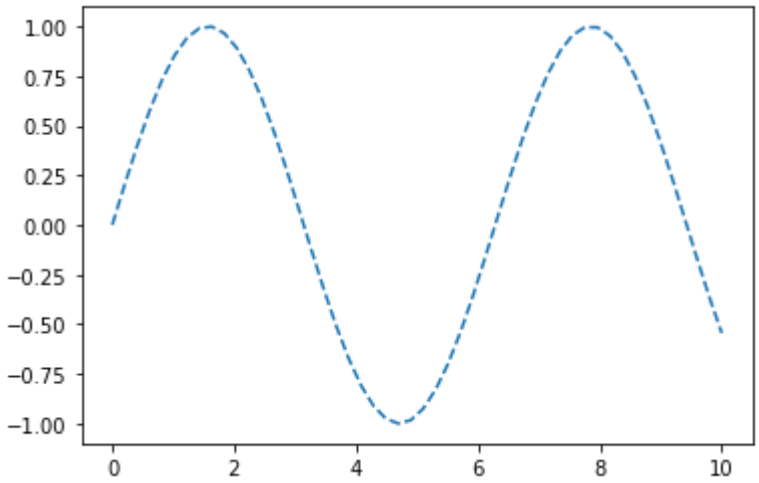


In [29]:

```
plt.plot(x, y, '--')
```

Out[29]:

```
[<matplotlib.lines.Line2D at 0x1f055d38f08>]
```

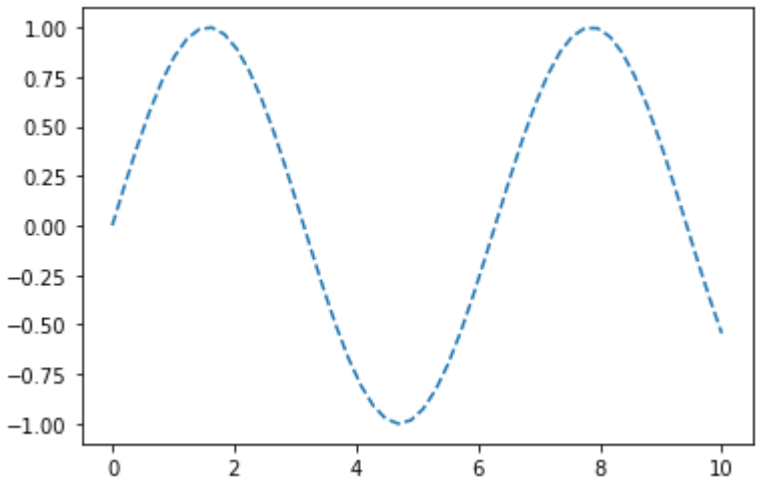


In [30]:

```
fig = plt.figure()
plt.plot(x, y, '--')
```

Out[30]:

```
[<matplotlib.lines.Line2D at 0x1f056808288>]
```



In [34]:

```
fig.savefig('E:\\git\\jupyter\\sin.png')
```

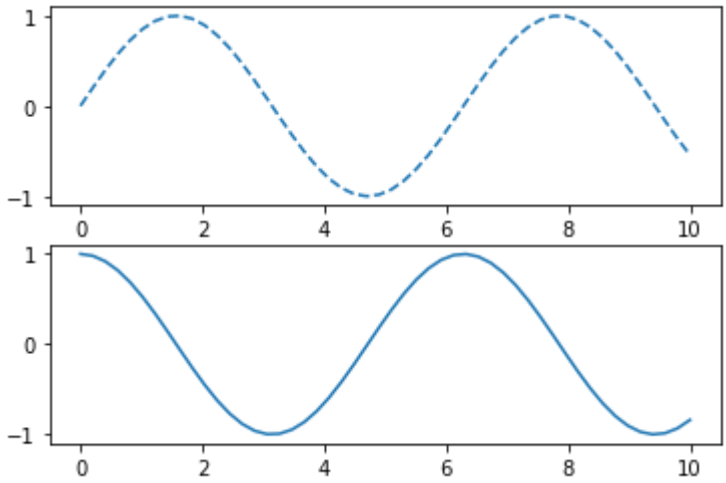
In [43]:

```
# 两行一列的第一幅图
# 虚线样式
plt.subplot(2, 1, 1)
plt.plot(x, np.sin(x), '--')

# 两行一列的第二幅图
plt.subplot(2, 1, 2)
plt.plot(x, np.cos(x))
```

Out[43]:

```
[<matplotlib.lines.Line2D at 0x1f058b43d08>]
```

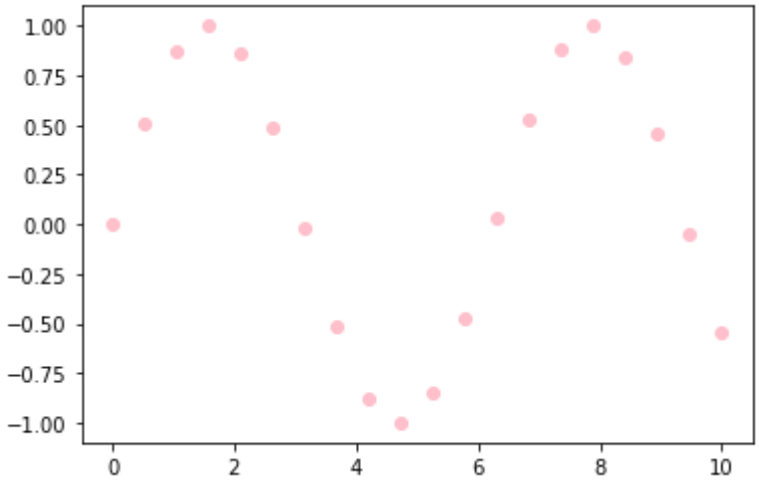


In [54]:

```
# 点状样式、color控制颜色
x = np.linspace(0, 10, 20)
plt.plot(x, np.sin(x), 'o', color = 'pink')
```

Out[54]:

```
[<matplotlib.lines.Line2D at 0x1f0574aad88>]
```



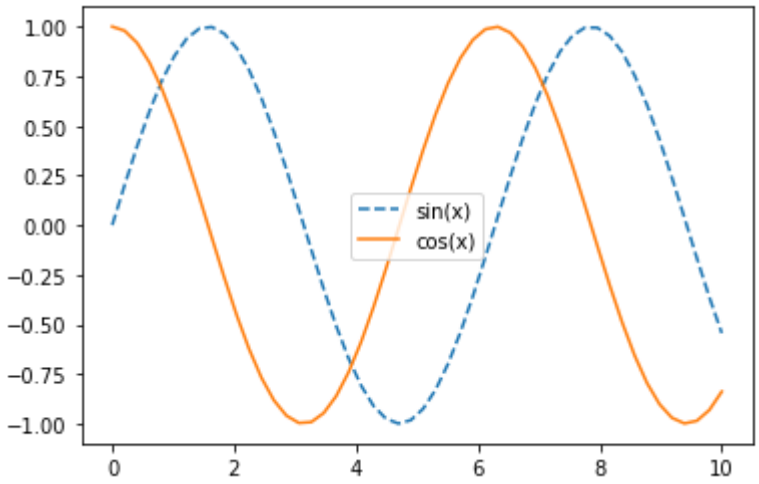
In [63]:

```
# label标签
x = np.linspace(0, 10, 50)
y = np.sin(x)
z = np.cos(x)

plt.plot(x, y, '--', label = 'sin(x)')
plt.plot(x, z, label = 'cos(x)')
# legend控制label位置，loc是控制label位置的显示 center= 10
# plt.legend? 查看函数使用
plt.legend(loc = 'center')
```

Out[63]:

```
<matplotlib.legend.Legend at 0x1f059346e08>
```



注意： plt.legend(loc = '**center**') 表格如下：

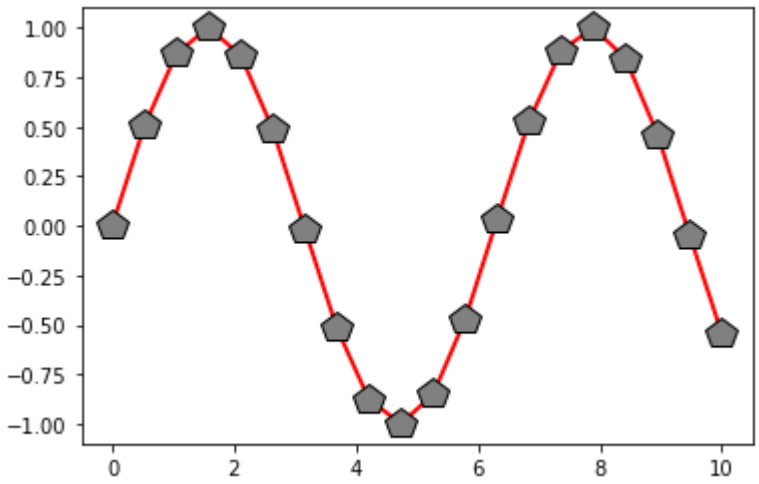
Location String	Location Code
'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10

In [75]:

```
# 可定制函数图像
# 具体参数查看 plt.plot?
x = np.linspace(0, 10, 20)
y = np.sin(x)
plt.plot(x, y, '-p', color = 'red', markersize = 16, linewidth = 2, markerfacecolor = 'grey', marke
```

Out[75]:

```
[<matplotlib.lines.Line2D at 0x1f055c14d88>]
```

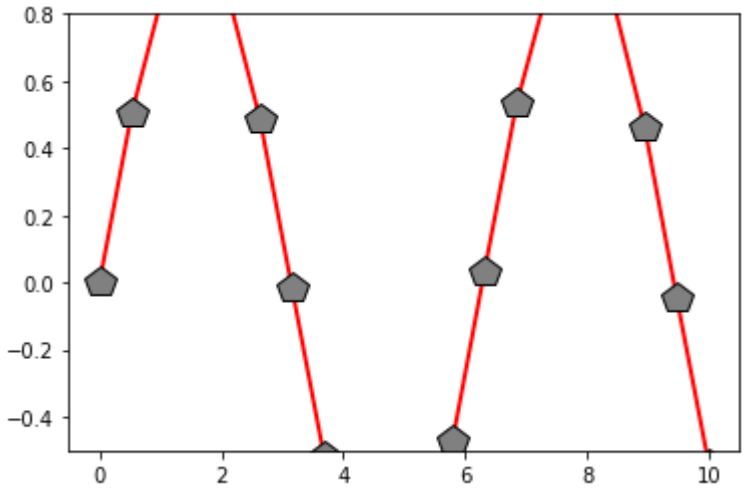


In [80]:

```
x = np.linspace(0, 10, 20)
y = np.sin(x)
plt.plot(x, y, '-p', color = 'red', markersize = 16, linewidth = 2, markerfacecolor = 'grey', marke
plt.ylim(-0.5, 0.8)
```

Out[80]:

(-0.5, 0.8)

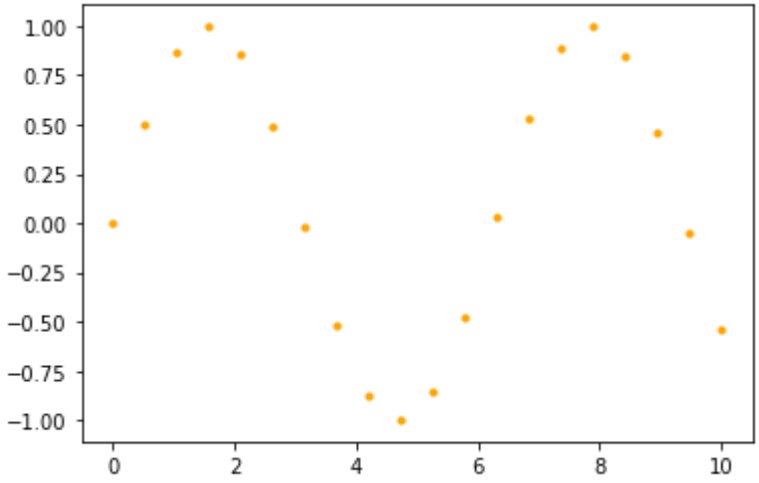


In [83]:

```
# 散点图制图
plt.scatter(x, y, s = 10, c = 'orange')
```

Out[83]:

<matplotlib.collections.PathCollection at 0x1f05a9ffc08>

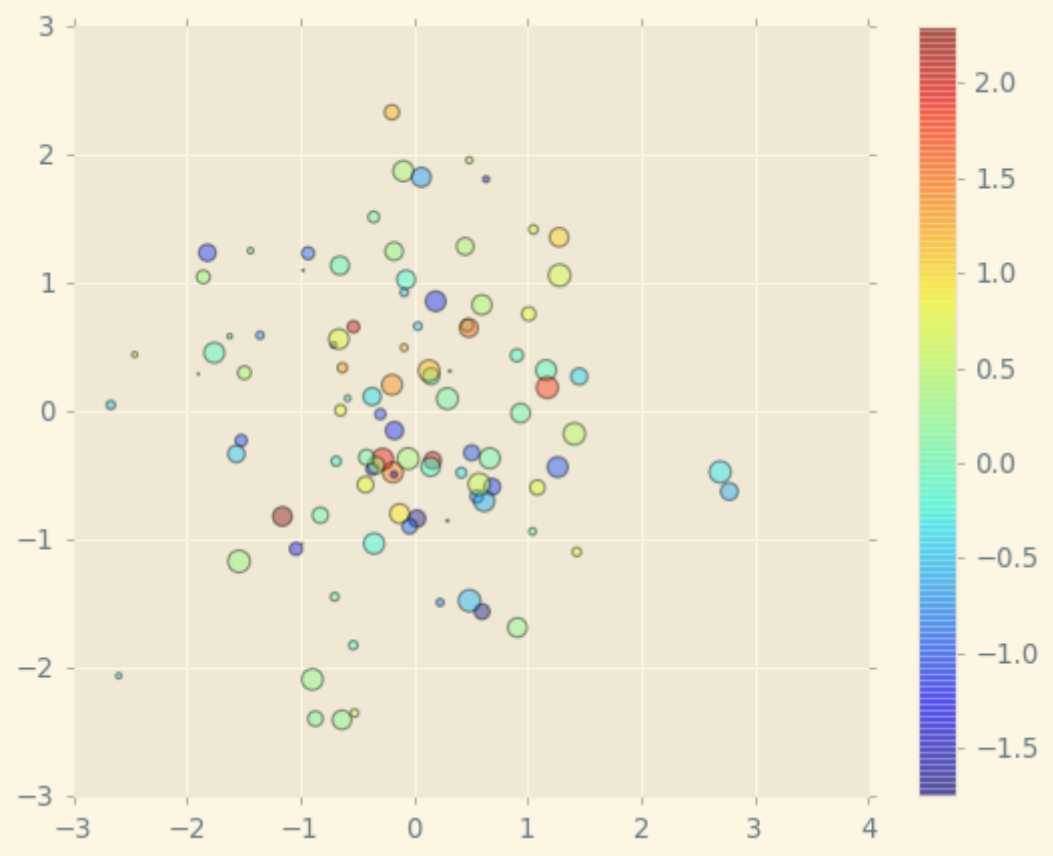


In [134]:

```
plt.style.use('Solarize_Light2')
x = np.random.randn(100)
y = np.random.randn(100)
colors = np.random.randn(100)
sizes = 100 * np.random.rand(100)
plt.scatter(x, y, c = colors, s = sizes, alpha = 0.4)
plt.colorbar()
```

Out[134]:

```
<matplotlib.colorbar.Colorbar at 0x1f05e5331c8>
```



In [136]:

```
print(plt.style.available)
```

```
['bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeight', 'ggplot', 'grayscal
e', 'seaborn-bright', 'seaborn-colorblind', 'seaborn-dark-palette', 'seaborn-dark',
'seaborn-darkgrid', 'seaborn-deep', 'seaborn-muted', 'seaborn-notebook', 'seaborn-pa
per', 'seaborn-pastel', 'seaborn-poster', 'seaborn-talk', 'seaborn-ticks', 'seaborn-
white', 'seaborn-whitegrid', 'seaborn', 'Solarize_Light2', 'tableau-colorblind10',
'_classic_test']
```

2 pandas 本身自带绘图

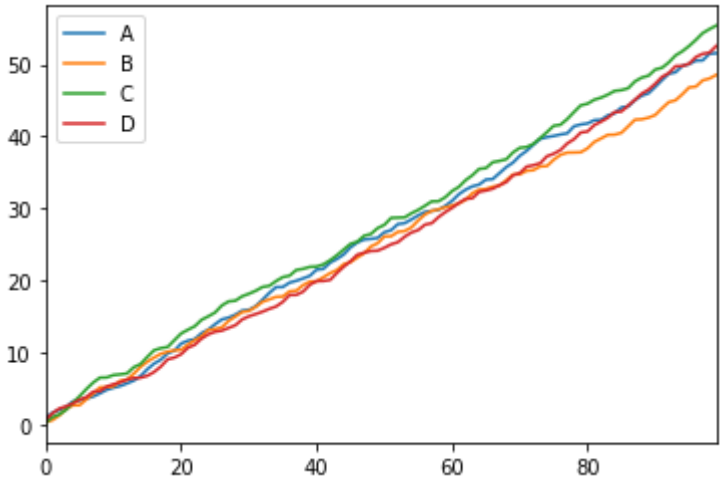
2.1 线性图

In [10]:

```
df = pd.DataFrame(np.random.rand(100, 4).cumsum(0), columns = ['A', 'B', 'C', 'D'])
df.plot()
```

Out[10]:

<matplotlib.axes._subplots.AxesSubplot at 0x26777544148>

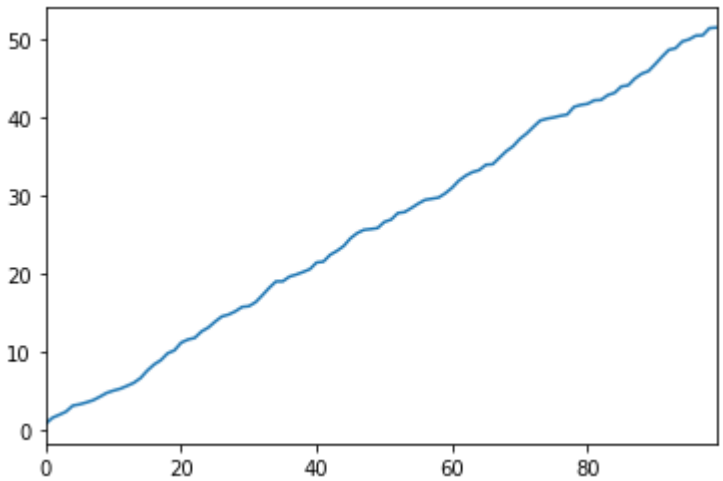


In [11]:

```
df.A.plot()
```

Out[11]:

<matplotlib.axes._subplots.AxesSubplot at 0x2677d8fda08>



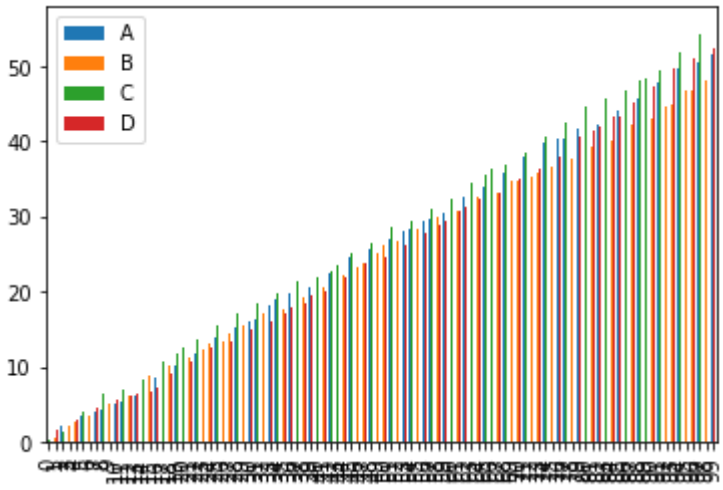
2.2 柱状图

In [12]:

```
df.plot.bar()
```

Out[12]:

<matplotlib.axes._subplots.AxesSubplot at 0x2677d4dd788>

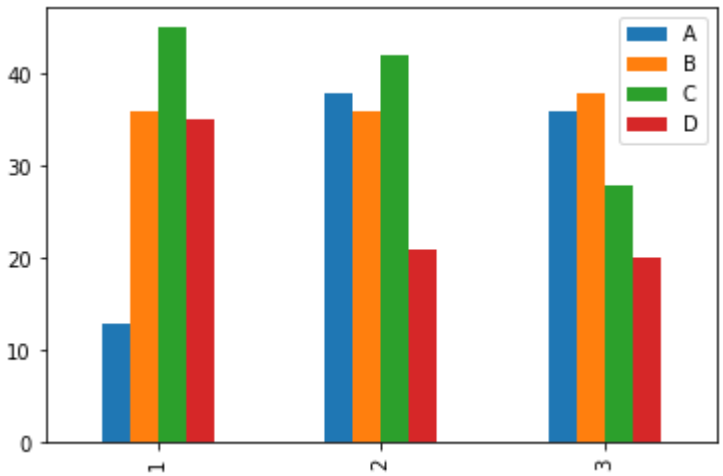


In [16]:

```
df = pd.DataFrame(np.random.randint(10, 50, (3, 4)), columns = ['A', 'B', 'C', 'D'], index = ['1',
df.plot.bar()
```

Out[16]:

<matplotlib.axes._subplots.AxesSubplot at 0x2670250d5c8>

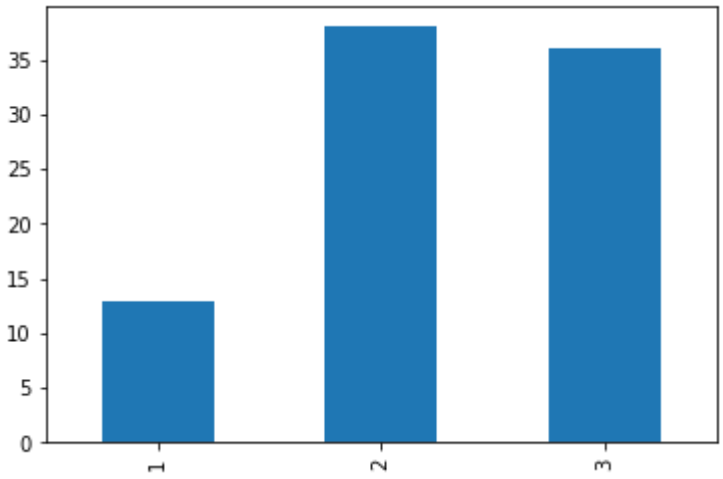


In [17]:

```
df.A.plot.bar()
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x26771e6c548>
```

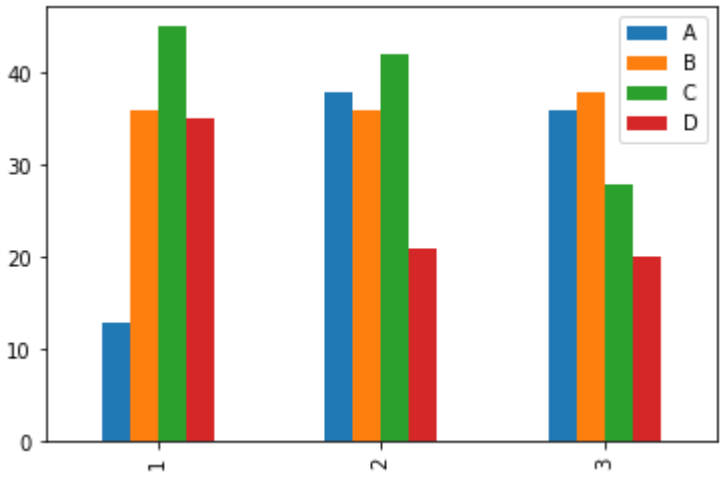


In [18]:

```
#等价于上面操作
df.plot(kind = 'bar')
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2670487f148>
```

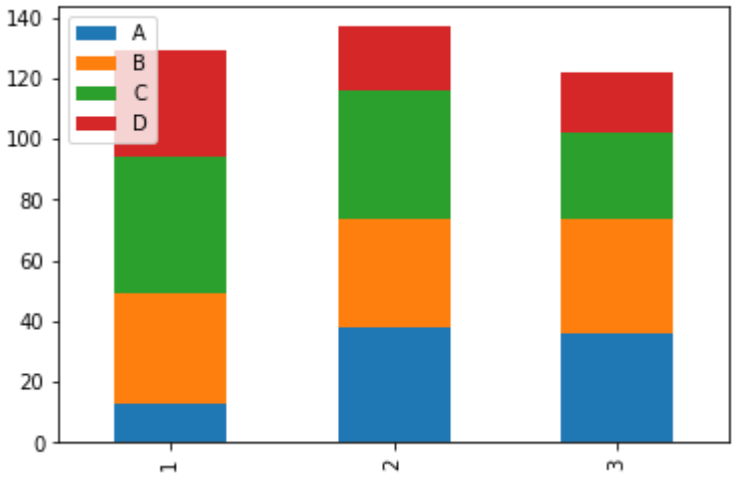


In [19]:

```
df.plot(kind = 'bar', stacked = True)
```

Out[19]:

<matplotlib.axes._subplots.AxesSubplot at 0x26704c3f288>



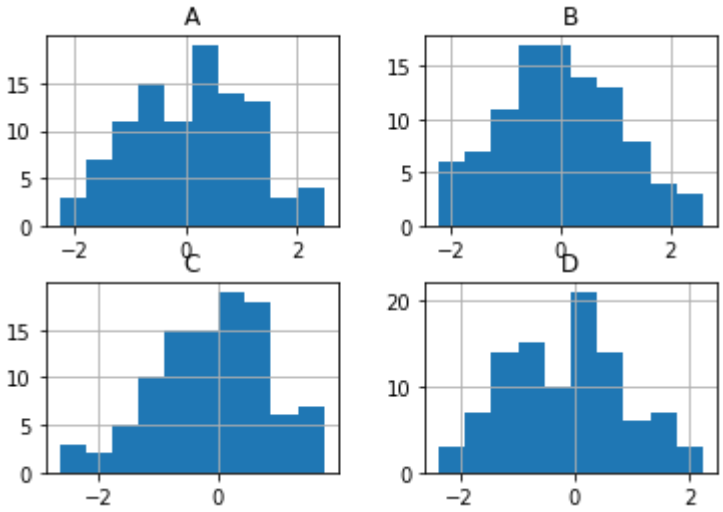
2.3 直方图

In [21]:

```
df = pd.DataFrame(np.random.randn(100, 4), columns = ['A', 'B', 'C', 'D'])
df.hist()
```

Out[21]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000026702BB8288>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002670558AF08>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x00000267055B8548>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x00000267055E9D48>]],
      dtype=object)
```

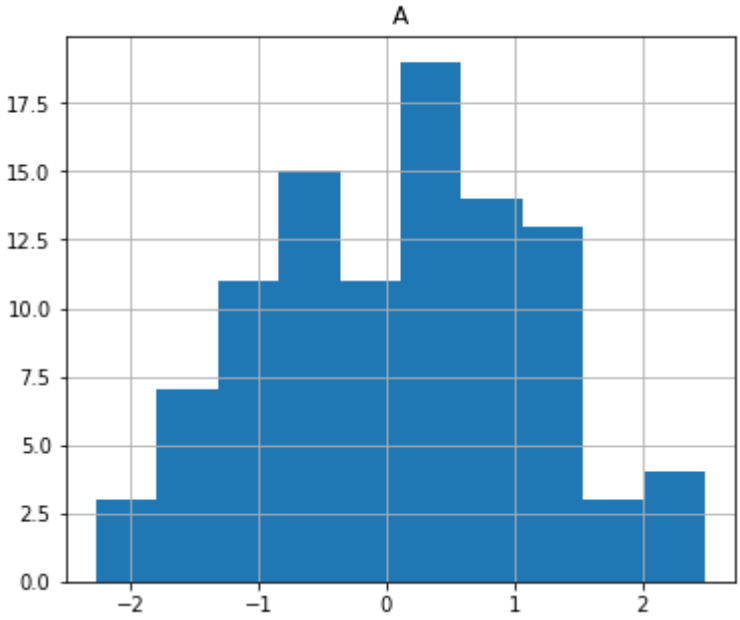


In [24]:

```
df.hist(column = 'A', figsize = (6, 5))
```

Out[24]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x00000267050A3D08>]],
      dtype=object)
```



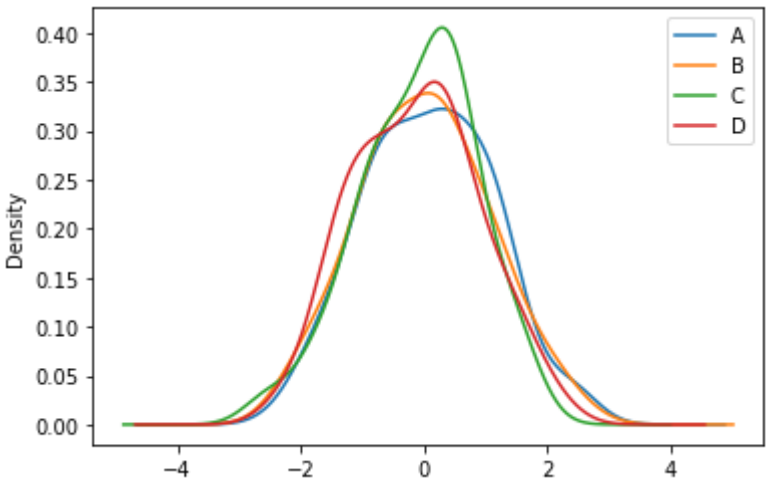
2.4 密度图

In [26]:

```
#df.plot(kind = 'kde')
df.plot.kde()
```

Out[26]:

<matplotlib.axes._subplots.AxesSubplot at 0x2670500b748>



In [37]:

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib.ticker import LinearLocator, FormatStrFormatter
import numpy as np

fig = plt.figure()
ax = fig.gca(projection = '3d')

# Make data
X = np.arange(-5, 5, 0.25)
Y = np.arange(-5, 5, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X ** 2 + Y ** 2)
Z = np.sin(R)

# plot the surface
surf = ax.plot_surface(X, Y, Z, cmap = cm.coolwarm, linewidth = 0, antialiased = False)

# Customize the Z axis
ax.set_zlim(-3.01, 3.01)
ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))

# Add a color bar which maps values to colors
fig.colorbar(surf, shrink = 0.5, aspect = 5)

plt.show()
```