正規分布

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
# 以上3つのライブラリが「3種の神器」
from scipy import stats # 統計ライブラリを使用
```

In [2]:
```
# 正規分布 N(m,s^2)
m=12              # 平均値の設定
s=3               # 標準偏差の設定
X=stats.norm(m,s) # 通常の記号 N(m,s^2) とパラメータの使い方が異なる
```

In [3]:
```
# 統計量の確認
X.mean()   # 平均値
```

Out[3]: 12.0

In [4]:
```
X.var()    # 分散
```

Out[4]: 9.0

In [5]:
```
X.std()    # 標準偏差
```

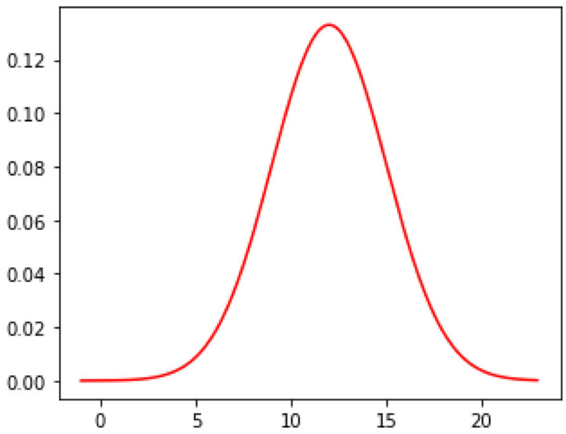
Out[5]: 3.0

In [6]:
```
# 密度関数の値（確率そのものではない）
X.pdf(12)
```

Out[6]: 0.1329807601338109

In [7]:
```
#  密度関数のグラフ
fig = plt.figure(figsize=(5,4))
x=np.arange(-1,23,0.1)  # x の範囲の指定、始点、終点、刻み幅
plt.plot(x,X.pdf(x),color='red')
```

Out[7]: [<matplotlib.lines.Line2D at 0x1cfefbb8520>]



分布関数と確率計算

In [8]:
```
# 分布関数 F(x)=P(X≦x)
# P(X ≦ 16.5)
X.cdf(16.5)        # Python の数値は先の方はあっていない。適当な桁で答えること
```

Out[8]:
```
0.9331927987311419
```

In [9]:
```
# Python は偶数丸めを行う
np.round(X.cdf(16.5),4) # 小数第4位まで表示
```

Out[9]:
```
0.9332
```

In [10]:
```
# P(a<X<b) = F(b)-F(a) で計算する
# P(13.2<X<16.5)
X.cdf(16.5)-X.cdf(13.2)  # Python の数値は先の方はあっていない。適当な桁で答えること
```

Out[10]:
```
0.27777105712081784
```

In [11]:
```
# P(X>a) = 1-F(a) で計算する
# P(X>16.5)
1-X.cdf(16.5)  # Python の数値は先の方はあっていない。適当な桁で答えること
```

Out[11]:
```
0.06680720126885809
```

上側α点 ¶

In [12]:
```
# 上側α点
# 上側 2.5% 点 (=両側 5% 点)
X.isf(0.025)
```

Out[12]:
```
17.879891953620163
```

2つの正規分布の比較

In [13]:
```
# 2つの正規分布を比較してみよう
Y=stats.norm(50,5)              # N(50,5^2)
Z=stats.norm(40,np.sqrt(24))    # N(40,24)
fig = plt.figure(figsize=(5,4))
xs=np.arange(20,70,0.1)
plt.plot(xs,Y.pdf(xs),color='blue',label='N(50,25)')
plt.plot(xs,Z.pdf(xs),color='red',label='N(40,24)')
plt.legend()        # 凡例の表示
```

Out[13]:
```
<matplotlib.legend.Legend at 0x1cff0ebda00>
```