PG11: 連続型確率分布

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
```

いつもの3つのライブラリに加えて、SciPy ライブラリに含まれる統計関連のパッケージを読み込んで始める。

特定の分布に従う確率変数は、

X = stats.xxxx()

のような書式で導入できる。 xxxx には分布の名称, () はパラメータの設定などのオプションを記入する。

1. 一様分布 U(a,b)

書式 uniform(a, s) によって, 区間 $[a, a+s]$ 上の一様分布を与える。 s は区間の端点ではなく、区間の幅である。

In [2]:

```
a, b = 2, 6
X = stats.uniform(a, b-a)
```

密度関数の図示

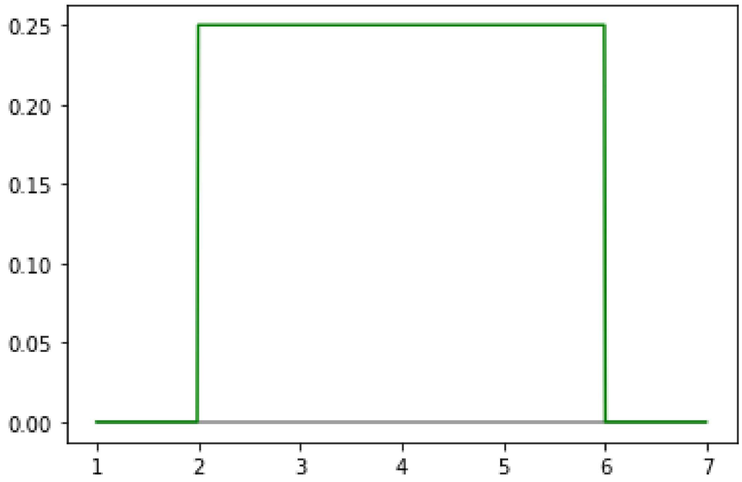
表示する x軸の範囲を指定する必要がある。ここでは区間 $[a, b]$ を前後に 1 伸ばした範囲で描画する。

In [3]:

```
x_range = np.arange(a-1, b+1, 0.01)
plt.plot(x_range, X.pdf(x_range), color='green')
plt.hlines(0, a-1, b+1, color='gray')
```

Out[3]:

<matplotlib.collections.LineCollection at 0x13c89bbef10>



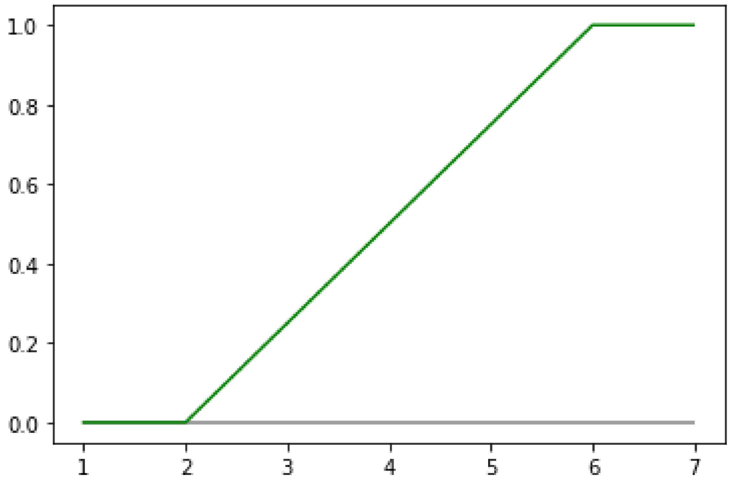
分布関数の図示

In [4]:

```
x_range = np.arange(a-1, b+1, 0.01)
plt.plot(x_range, X.cdf(x_range), color='green')
plt.hlines(0, a-1, b+1, color='gray')
```

Out[4]:

<matplotlib.collections.LineCollection at 0x13c8aecee50>



【便利】グラフを並べて描画する

まず、描画領域の大きさを指定して名前を付ける。デフォルトの大きさは (6, 4) である。 たとえば、(12, 4) の大きさの描画領域を確保して fig と言う名前を付けよう。

fig = plt.figure(figsize = (12, 4))

次に、描画領域を 1×2 に分割して、1番目の小領域に ax1、 2番目の小領域に ax2 という名前を付ける。

ax1 = fig.add_subplot(1, 2, 1)

ax2 = fig.add_subplot(1, 2, 2)

それぞれの領域にグラフを描画するのは plt.plot() 関数の代わりに ax1.plot() 関数を使う。

なお、hlines(c, a, b) によって水平線 $y = c$ を $a \leq x \leq b$ の範囲に描く。

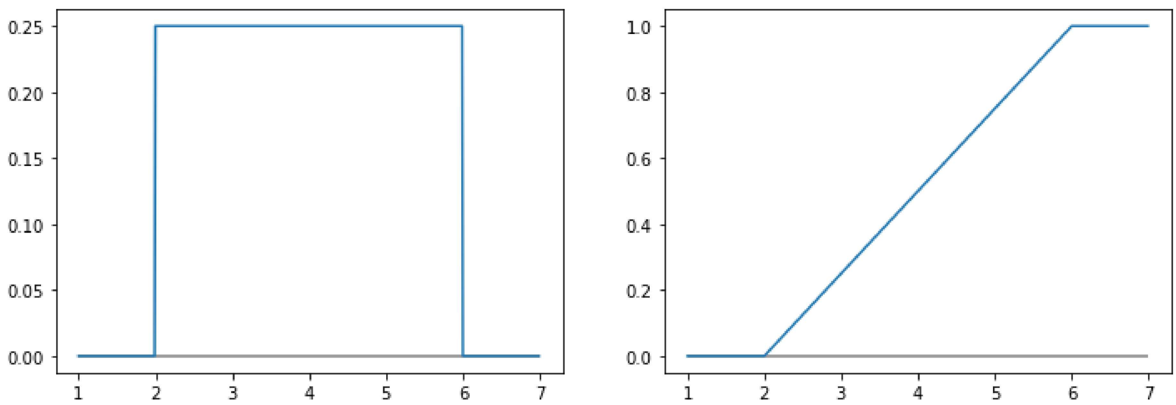
垂直線も同様で vlines(c, a, b) によって垂直線 $x = c$ を $a \leq y \leq b$ の範囲に描く。

In [5]:

```
fig = plt.figure(figsize = (12, 4))
ax1 = fig.add_subplot(1, 2, 1)
ax2 = fig.add_subplot(1, 2, 2)

x_range = np.arange(a-1, b+1, 0.01)

ax1.plot(x_range, X.pdf(x_range))
ax1.hlines(0, a-1, b+1, color='gray')

ax2.plot(x_range, X.cdf(x_range))
ax2.hlines(0, a-1, b+1, color='gray')
```

Out[5]:

```
<matplotlib.collections.LineCollection at 0x13c8af76df0>
```



In [6]:

```
# 確率計算 $P(3 ≦ X ≦ 4)$ は分布関数を使う。
X.cdf(4) - X.cdf(3)
```

Out[6]:

```
0.25
```

In [7]:

```
# 統計量の計算
X.mean(), X.var(), X.std()
```

Out[7]:

```
(4.0, 1.3333333333333333, 1.1547005383792515)
```

In [8]:

```
# 上側 α 点
X.isf(0.05)
```

Out[8]:

```
5.8
```

In [9]:

```
# 信頼区間
X.interval(0.9)
```

Out[9]:

```
(2.2, 5.8)
```

2. 指数分布 $Exp(\lambda)$

書式 expon(scale=s) によって、パラメータ $1/s$ の指数分布を与える。言い換えると、s は指数分布の平均値である。

In [10]:

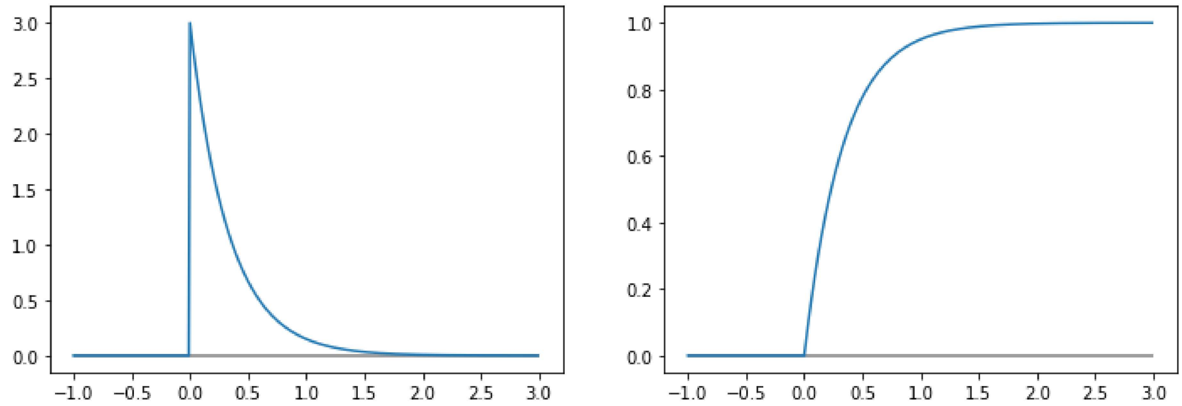
```
lam = 3
Y = stats.expon(scale=1/lam)
```

In [11]:

```
fig = plt.figure(figsize = (12, 4))
ax1 = fig.add_subplot(1, 2, 1)
ax2 = fig.add_subplot(1, 2, 2)

x_range = np.arange(-1, 3, 0.01)

ax1.plot(x_range, Y.pdf(x_range))
ax1.hlines(0, -1, 3, color='gray')

ax2.plot(x_range, Y.cdf(x_range))
ax2.hlines(0, -1, 3, color='gray')
```

Out[11]:

<matplotlib.collections.LineCollection at 0x13c8b0378b0>



In [12]:

```
# 確率計算 $P(0.5 ≦ Y ≦ 1.2)$ は分布関数を使う。
Y.cdf(1.2) - Y.cdf(0.5)
```

Out[12]:

0.19580643770113726

In [13]:

```
# 統計量の計算
Y.mean(), Y.var(), Y.std()
```

Out[13]:

(0.3333333333333333, 0.1111111111111111, 0.3333333333333333)

In [14]:

```
# 上側 α 点
Y.isf(0.05)
```

Out[14]:

0.9985774245179969

3. 正規分布 $N(\mu, \sigma^2)$

書式 norm(m, s) によって、平均値 m, 標準偏差 s の正規分布 $N(m, s^2)$ を与える。

In [15]:

```
m, s = 3, 2
Z = stats.norm(m, s)
```

密度関数と分布関数の描画

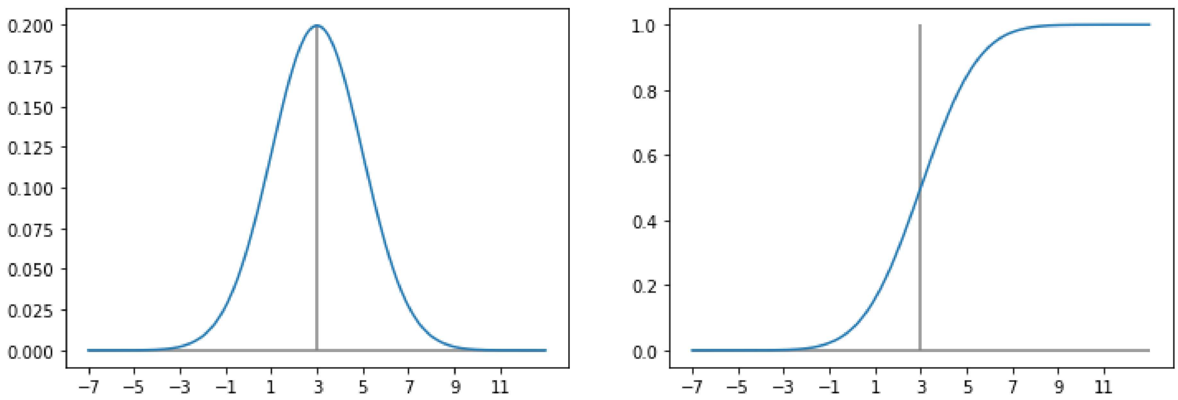
x軸の範囲を指定する必要がある。ここでは $m \pm 5s$ の範囲を指定した。

In [16]:

```
fig = plt.figure(figsize = (12, 4))
ax1 = fig.add_subplot(1, 2, 1)
ax2 = fig.add_subplot(1, 2, 2)

x_range = np.arange(m - 5*s, m + 5*s, 0.01)

ax1.plot(x_range, Z.pdf(x_range))
ax1.hlines(0, m - 5*s, m + 5*s, color='gray')
ax1.vlines(m, 0, 0.2, color='gray')
ax1.set_xticks(np.arange(m-5*s, m+5*s, s))

ax2.plot(x_range, Z.cdf(x_range))
ax2.hlines(0, m - 5*s, m + 5*s, color='gray')
ax2.vlines(m, 0, 1, color='gray')
ax2.set_xticks(np.arange(m-5*s, m+5*s, s))
```

Out[16]:

```
[<matplotlib.axis.XTick at 0x13c8b0d5430>,
 <matplotlib.axis.XTick at 0x13c8b0d5400>,
 <matplotlib.axis.XTick at 0x13c8b0c9f40>,
 <matplotlib.axis.XTick at 0x13c8b10da30>,
 <matplotlib.axis.XTick at 0x13c8b119310>,
 <matplotlib.axis.XTick at 0x13c8b119880>,
 <matplotlib.axis.XTick at 0x13c8b119d90>,
 <matplotlib.axis.XTick at 0x13c8b1232e0>,
 <matplotlib.axis.XTick at 0x13c8b1237f0>,
 <matplotlib.axis.XTick at 0x13c8b123d00>]
```



In [17]:

```
# 確率計算 $P(0.5 ≦ Z ≦ 2.5)$ は分布関数を使う。
Z.cdf(2.5) - Z.cdf(0.5)
```

Out[17]:

0.29564390065022095

In [18]:

```
# 統計量の計算
Z.mean(), Z.var(), Z.std()
```

Out[18]:

(3.0, 4.0, 2.0)

In [19]:

```
# 上側 α 点
Z.isf(0.05)
```

Out[19]:

6.289707253902946

In [20]:

```
# 信頼区間
CI = Z.interval(0.9)
CI
```

Out[20]:

(-0.2897072539029457, 6.289707253902945)

In [21]:

```
# 検算
Z.cdf(CI[1]) - Z.cdf(CI[0])
```

Out[21]:

0.9

例題

$X \sim N(-1, 2^2)$ のとき, (1) $P(X \leq 2.29)$, (2) $P(X > x) = 0.01$

In [22]:

```
X1 = stats.norm(-1,2)
X1.cdf(2.29)
```

Out[22]:

0.9500150944608786

In [23]:

```
X1.isf(0.01)
```

Out[23]:

```
3.6526957480816815
```

例題

$X \sim N(156, 5^2)$ のとき, $P(153 \leq X \leq 160)$

In [24]:

```
X2 = stats.norm(156,5)
X2.cdf(160) - X2.cdf(153)
```

Out[24]:

```
0.5138914836665298
```

In [ ]:

```
```