mwp-076

functions - 関数群　【 評価版 】

Stata には数多くの関数が用意されており、`generate` コマンドや数式記述の中で利用されます。詳細は `[FN]` マニュアルを参照いただくとして、ここでは代表的な関数に限定した形でその用法を紹介します。なお、`[D]` **egen** (*mwp-077*) についても併せてご参照ください。

1. 乱数の発生

統計解析を行う上で擬似乱数を発生させたいといったニーズが生ずる場合があります。Stata には種々の分布に従って乱数を発生させる関数が用意されていますが、本セクションでは一様分布と正規分布のケースを例にとってその用例を紹介します。その他の機能・用例については [FN] *Random-number functions* (*mwp-382*) をご参照ください。

▷ 用例 1: 一様乱数

関数 runiform() は $[0,1)$ 上で一様に分布する擬似乱数を発生させます。例えば観測データ (observations) 数を 10 に設定した上で、変数 x_1 中に一様乱数を発生させてみます。なお、repeatability を確保する意味で最初に擬似乱数発生用の seed を設定しておきます。

```
. set obs 10
number of observations (_N) was 0, now 10
. set seed 2
. generate x1 = runiform() *1
. list *2
```

	x1
1.	.903604
2.	.8502361
3.	.7838205
4.	.9253171
5.	.2529037
6.	.1358858
7.	.2245407
8.	.0996503
9.	.0220877
10.	.6858429

*1 メニュー操作： Data ▷ Create or change data ▷ Create new variable
*2 メニュー操作： Data ▷ Describe data ▷ List data

区間 $[a, a+b)$ 上での一様乱数を発生させたい場合には

```
a + b*runiform()
```

という数式を用いれば良いわけです。例えば区間 $[10, 20)$ 上での一様乱数を変数 x_2 中に発生させるには次のように操作します。

```
. generate x2 = 10 + 10*runiform()
. list x2
```

	x2
1.	16.54085
2.	19.68395
3.	18.03331
4.	11.32778
5.	12.00773
6.	10.67624
7.	19.9828
8.	15.65972
9.	13.56483
10.	13.83774

整数値が欲しい場合には数学関数 `round()`（四捨五入）、または `floor()`（切捨て）を使用します。

```
. generate x3 = round(x2)
. list x2 x3
```

	x2	x3
1.	16.54085	17
2.	19.68395	20
3.	18.03331	18
4.	11.32778	11
5.	12.00773	12
6.	10.67624	11
7.	19.9828	20
8.	15.65972	16
9.	13.56483	14
10.	13.83774	14

この round() や floor() を runiform() と組み合わせて使用することもできます。

```
. generate x4 = round(10 + 10*runiform())
. list x4
```

	x4
1.	11
2.	11
3.	15
4.	14
5.	18
6.	13
7.	15
8.	11
9.	14
10.	12

◁

▷ **用例** 2: **正規乱数**

正規分布 $N(\mu,\sigma)$ に従う乱数を発生させたい場合には rnormal(μ,σ) という関数を使用します。例えば $\mu = 10, \sigma = 3$ の正規分布に従う乱数を 100 個生成してみます。

```
. clear
. set obs 100
number of observations (_N) was 0, now 100
. set seed 2
. generate x1 = rnormal(10, 3)
```

この結果、変数 x_1 には次のような実数値が生成されます。

```
. list in 1/5
```

	x1
1.	8.957761
2.	4.316012
3.	11.90286
4.	8.515435
5.	7.285913

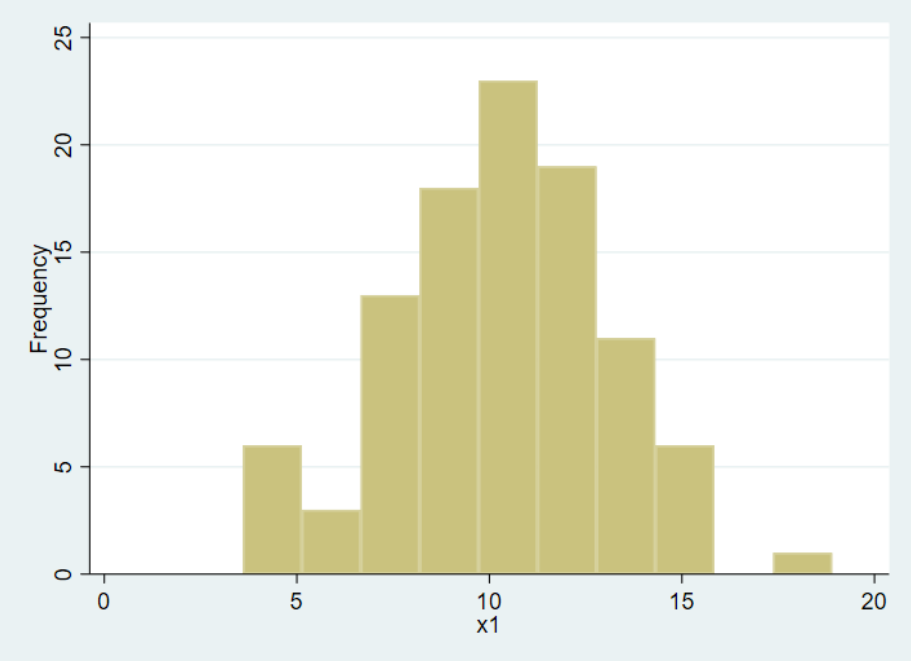
小数点以下 1 桁の実数が欲しい場合には用例 1 と同様、関数 round() を用いて次のように操作します。

```
. replace x1 = round(10*x1)/10 *3
(100 real changes made)
. list in 1/5
```

	x1
1.	9
2.	4.3
3.	11.9
4.	8.5
5.	7.3

生成された 100 個のデータからヒストグラムを作成してみると次のようになります。

```
. histogram x1, frequency *4
(bin=10, start=3.5999999, width=1.53)
```



rnormal(μ, σ) という関数は汎用性の高いものですが、これ以外に rnormal() と rnormal(μ) という関数形も用意されています。rnormal() の場合には $\mu = 0, \sigma = 1$ が、rnormal(μ) の場合には $\sigma = 1$ が仮定されます。

◁

*3 メニュー操作： Data ▷ Create or change data ▷ Change contents of variable
*4 メニュー操作： Graphics ▷ Histogram

2. 数学関数

2.1 整数値への変換

評価版では割愛しています。

2.2 Running sum

評価版では割愛しています。

3. 統計分布関数

3.1 t 検定の実行

評価版では割愛しています。

3.2 分布関数の用例

▷ 用例 3: t 分布のグラフ化

評価版では割愛しています。

▷ 用例 4: 面積の算出

評価版では割愛しています。

▷ 用例 5: 限界値の算出

評価版では割愛しています。

4. プログラミング関数

4.1 コード化関数

評価版では割愛しています。

▷ 用例 6: `recode()`

評価版では割愛しています。

▷ 用例 7: `autocode()`

評価版では割愛しています。

▷ **用例** 8: `irecode()`

評価版では割愛しています。

5. 文字列関数

▷ **用例** 9: **和暦 → 西暦**

評価版では割愛しています。

6. 日付/時間関数

評価版では割愛しています。

7. 行列操作関数

▷ **用例** 10: **行列の操作**

評価版では割愛しています。

補足１ – グラフ作成コマンド操作

評価版では割愛しています。

■