

データ分析の前に - vol.2 文字列の扱い

最初に、前回行った操作をひとまとめにしたコマンドを示しておきます。サンプルデータは前回と同じ samplecsv.csv を利用します。初めて EViews プログラミングに取り組む方は File/New/Program としてプログラムウィンドウを開き、次のコマンドを入力してください。コメント文を入力する際は必ず行頭に半角のアポストロフィを入力します。コメント文は緑色で表示されます。コマンドの間に敢えて1行空白行を入れていますが、これは任意です。また、ダブルクォーテーションで囲んだ情報は文字列として認識され、赤色で画面表示します。ここではcドライブの evIEWS/evnews201311 というフォルダにサンプルデータ samplecsv.csv が保存してあるものとします。

'デフォルトのディレクトの設定

```
cd c:¥evIEWS¥evnews201311
```

'テキストファイルのインポート

```
wfopen(page=text) "samplecsv.csv"
```

'元データの複製price2を作成する

```
series price2=price
```

'コードx100の欠損値だけを標本(sample)とする

```
smpl @all if code="x100" and price=na
```

'1期前の値を当期の値とする

```
price2=price2(-1)
```

'標本を全範囲(Range)に戻す

```
smpl @all
```

'code date time price price2からなるグループオブジェクトgroup01を作成する

```
group group01 code date time price price2
```

'画面にgroup01の内容を表示する

```
show group01
```

ここまで入力したら **Run** ボタンをクリックしてプログラムを実行します。ここにはウィンドウの一部だけを表示しますが、**x100** の欠損値には 1 期前の値がコピーされます。

| | CODE | DATE | TIME | PRICE | PRICE2 |
|----|------|----------|------|-------|----------|
| 1 | x100 | 20091001 | 904 | 221 | 221.0000 |
| 2 | x100 | 20091001 | 904 | 226 | 226.0000 |
| 3 | x100 | 20091001 | 905 | 280 | 280.0000 |
| 4 | x100 | 20091001 | 905 | 295 | 295.0000 |
| 5 | x100 | 20091001 | 908 | 259 | 259.0000 |
| 6 | x100 | 20091001 | 913 | 270 | 270.0000 |
| 7 | x100 | 20091001 | 913 | 244 | 244.0000 |
| 8 | x100 | 20091001 | 914 | 265 | 265.0000 |
| 9 | x100 | 20091001 | 914 | 280 | 280.0000 |
| 10 | x100 | 20091001 | 914 | 208 | 208.0000 |
| 11 | x100 | 20091001 | 914 | 248 | 248.0000 |
| 12 | | | | | |

プログラムが正常に終了したら、**step01** という名前でもプログラムファイルを保存します。

■他の銘柄でも繰り返す

サンプルデータには **x100** から **x800** までの 8 種類の銘柄が含まれています。ここではループレコメンドを利用して同じ処理を繰り返しましょう。注意点は銘柄が文字列 **x** を含んでいるところにあります。そこで、**x** という文字列と、数値の 100 から 800 までの数値を文字列属性に変更し、結合します。試しに、新しいプログラムウィンドウを開き次のコマンドを入力し、実行してください。

```
%sname="x"+@str(100)
```

```
show %sname
```

このコマンドは 1 行目で文字 **x** と、**@str()** コマンドで数値属性の 100 を文字属性に変換した情報を結合し、文字変数 **%sname** に確保します。2 行目のコマンドはその中身を仮の **alpha** オブジェクトとして画面表示します。

| | Object | Properties | Print | Name | Freeze | Default | Sort |
|--------------------------------|--------|------------|-------|------|--------|---------|------|
| "x100" | | | | | | | |
| Last updated: 11/26/13 - 17:57 | | | | | | | |
| Formula: "x100" | | | | | | | |
| 1 | x100 | | | | | | |
| 2 | x100 | | | | | | |
| 3 | x100 | | | | | | |
| 4 | x100 | | | | | | |
| 5 | x100 | | | | | | |
| 6 | x100 | | | | | | |
| 7 | x100 | | | | | | |
| 8 | x100 | | | | | | |
| 9 | | | | | | | |

次は**@str(100)**の中の数値を 100 から 800 まで変化させれば目的を果たすことができそうです。そのために **FOR** ループを使ってみましょう。

```
for !i=1 to 8
```

```
  !j=!i*100
```

```
  %sname="x"+@str(!j)
```

```
next
```

文字変数は記号%に任意の文字列を続けて、%snameのように作ります。数値変数は記号!
にこちらも任意の文字列を続けます。ここでは、2つの数値変数!iと!jを利用します。もち
ろん!countなど複数の文字を利用することも可能です。

まず、

```
for !i=1 to 8
    ----
next
```

は---部分のコマンドを8回繰り返し実行します。ただし、!iの値は1から8まで1単位づ
つ増加します。したがって、!jの値は100から800まで変化します。このとき、%sname
にはx100からx800までの値が順番に入ることになります。我々の目的は次のコマンドを
自動作成することです。

```
smpl @all if code="x100" and price=na
```

したがって、プログラムウィンドウに次のように入力し、実行してみましょう。

```
for !i=1 to 8
    !j=!i*100
    %sname="x"+@str(!j)
    smpl @all if code=%sname and price=na
next
smpl @all
```

このコマンドを実行しても画面上は何の変化もありません。標本(sample)を変更する作業を
繰り返しているだけです。今回の目的は各銘柄について欠損値に前期の値を代入すること
ですので、最初に示したプログラムと上記のルーチンを組み合わせます。

練習用に開いていたプログラムウィンドウを閉じ、step01のプログラムを次のように変更
します。

```

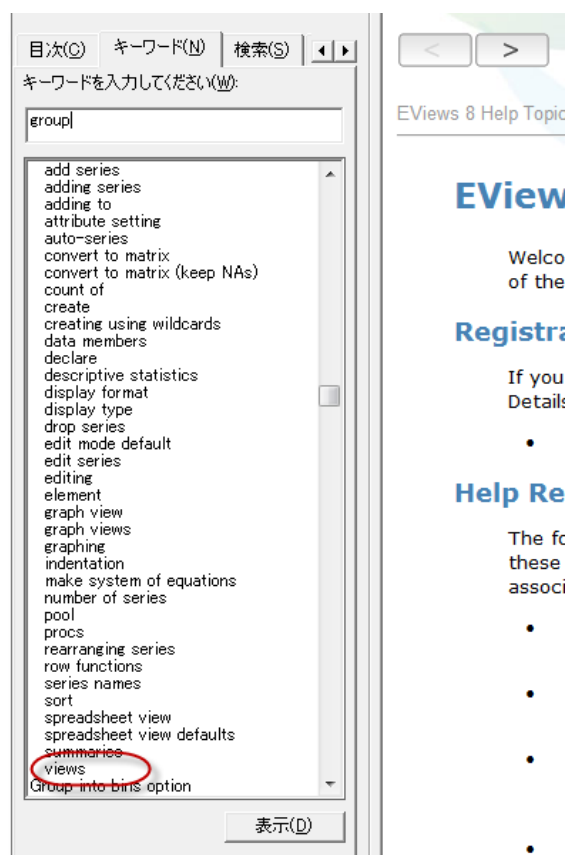
cd c:\%evIEWS%\evnews201311
wfoopen(page=text) "samplecsv.csv"
series price2=price
for !i=1 to 8
    !j=!i*100
    %sname="x"+@str(!j)
    smpl @all if code=%sname and price=na
    price2=price2(-1)
next
smpl @all
group group01 price price2

```

'priceとprice2の個別の統計量を表示する
show group01.stats(i)

コマンドが正常に実行できたら、プログラムファイルに **step02** という名前を付けて保存します。

最後のコマンドは **group01** のオブジェクトで **View/Descriptive Stats/Individual Samples** の操作に相当するコマンドです。EViewsには数値データを格納する **series** オブジェクト、それらをグループ化した **group** オブジェクトなど、たくさんのオブジェクトが存在します。これらのオブジェクトに対するコマンドは「オブジェクト名.コマンド名」という形で用いられます。試しに、EViewsのメインメニューから、**Help/EViews Help Topics** と操作し、キーワードに「**group**」と入力します。画面下の方に **views** という項目がありますので、ここをクリックしますと、**group** オブジェクトの **views** ボタンをクリックしたときに表示されるコマンドのプログラム用コマンドに関する解説を表示します。



最後に文字列の操作に関して便利なコマンドがありますので、それを紹介しておきます。まずはコマンドウィンドウに次のコマンドを入力し、**alpha** オブジェクトを作成します。

```
alpha wrd="123abc7890"
```

数値を格納する **series** オブジェクトに対し、**alpha** オブジェクトは文字列を格納する入れ物です。それでは、**wrd** から **123** と **7890** を取り出し、結合して文字列の **1237890** を作成し、最後にこれを数値属性に変換し、**series** オブジェクト **num** を作成します。コマンドウィンドウに次のように入力します。

```
alpha wrd2=@left(wrd,3)+@right(wrd,4)
```

```
series num=@val(wrd2)
```

1行目のコマンド**@left()**は **wrd** の左から3文字を取得、**@right()**は右から4文字を取得します。文字列の場合の**+**は結合を行うコマンドです。2行目の**@val()**は文字列属性を数値属性に変換するコマンドです。これ以外にも**@mid(整数 1[, 整数 2])**は文字列の左から整数1文字目から、(整数2)文字分取り出します。[整数2]を省略した場合、整数1文字目から右端までの文字を取り出します。この3つのコマンドの存在だけ、覚えておきましょう。これ以外にも文字列の操作に便利なコマンドが多数あります。調べてみたいときは、前出のオンラインヘルプのキーワードに **string** と入力し、サブ項目の **extract portion** をクリックしてください。

今回は、**for loop** と **if** 条件文を利用して、右のような表を作成するコマンドについて解説します。データ分析の過程で計算上のチェックポイントを設け、その時点での計算結果に誤りがないか確認する場合など、表を活用すると非常に助かります。

| CODE | Mean | Std. Dev. | Obs. |
|------|----------|-----------|------|
| x100 | 241.3023 | 29.59315 | 1118 |
| x200 | 373.4469 | 29.99888 | 866 |
| x300 | 181.8786 | 29.18443 | 997 |
| x400 | 517.8479 | 42.86735 | 1052 |
| x500 | 285782.9 | 3462.826 | 956 |
| x600 | 351.5190 | 32.03596 | 948 |
| x700 | 217.6525 | 29.06680 | 777 |
| x800 | 403.4112 | 29.80806 | 1138 |
| All | 35085.22 | 93356.57 | 7852 |

