

コマンドキャプチャ機能について

株式会社ライトストーン

2016年2月12日

今回は、EViews9で追加されたコマンドキャプチャ機能の実力を、多方面から検証していきます。コマンドキャプチャ機能は非常に便利な機能であることを確認していきますが、現時点での限界についても、忌憚なく明らかにしていきます。

EViews8以前をお使いの方も是非一度本資料にお目通しいただき、可能であれば体験版で機能をお試ください。

1 コマンドキャプチャ機能の概要

EViewsは使いやすいグラフィカルインターフェースに一つの強みがあるソフトウェアです。統一的なレイアウトがなされているため、初めて行う操作でもメニューのどの辺りを調べれば良いのかの見当が付く作りになっています。しかしながら、時にはメニューではなく、コマンドで操作したいような場合もございます。たとえば、以下のようなケースです。

- 同様の操作を日常的に繰り返す場合
- 同様の操作を複数のシリーズに行いたい場合

こういった場合、毎回メニューで操作するよりも、コマンド操作を行った方が遥かに時間の短縮になります。あるいは、以下のようなケースも考えられます。

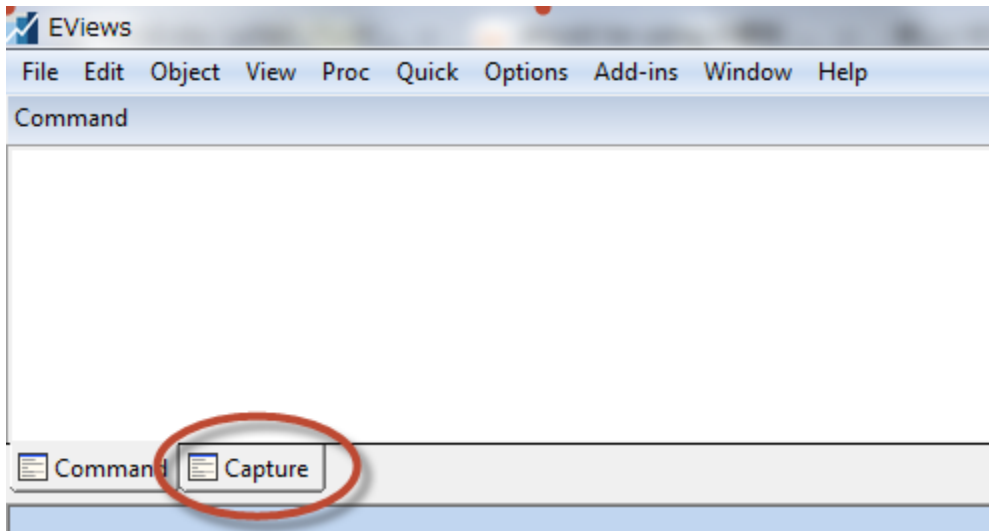
- 関連情報を調べるためのキーワードが知りたい
- よく行う処理をプログラム化したい

このような場合、メニュー操作に対応するコマンド名を知る必要がございます。

EViews9では、新しく搭載されたコマンドキャプチャ機能により、メニューから行った大抵の処理はコマンドとして記録されるようになりました。コマンドキャプチャ機能は、まさに上記のような場合に役に立つ機能です。コマンドキャプチャ機能により、メニューでの操作方法さえ知っていれば、いちいちヘルプやマニュアルから対応するコマンドを探す手間が省けるためです。

2 コマンドキャプチャ機能の使い方

まず、コマンドキャプチャウィンドウ（以下ではキャプチャウィンドウ）の位置を確認します。デフォルトでは、以下の位置にドッキングされています。



コマンドウィンドウとキャプチャウィンドウは、左クリックにより瞬時に切り替えることが可能です。また、ウィンドウとして独立させることも可能です。キャプチャウィンドウを独立させたい場合、マウスの左ボタンを押したまま、お好みの位置までドラッグ&ドロップしてください*1。逆にデフォルトの状態に戻したい場合は、Window>Reset Window Layout と操作してください。メニューから行った操作がコマンドキャプチャ機能に対応していれば（多くの機能は対応しております）このキャプチャウィンドウに対応するコマンドが逐次記録されていきます。以下ではいくつかの実例に基づき、コマンドキャプチャ機能の利便性について解説してきたいと思います。

3 操作例

3.1 外部ワークファイルを取り込む際の例

非構造データとして取り込む場合 data1.xlsx をダウンロードします。キャプチャウィンドウを表示した状態で、メニューから以下のように操作し、data1 を取り込んでください。なお、以下の私の説明は C ドライブ直下に作成した data\CC フォルダにデータを保存して行っています。ここにデータを保存しておけば、本 PDF でキャプチャされたコマンドがお客様の環境でもそのまま有効となります。

File>Open>Foreign Data as Workfile

ダイアログが表示されたら完了と押し、次の Link ~ では No としておきます。すると、キャプチャウィンドウに次のようなコマンドが表示されます（エクセルファイルの保存位置によってファイルパスの部分は変わります）。

```
wfopen C:\data\CC\data1.xlsx range=Sheet1 colhead=1 na="#"#N/A" @smpl @all
```

*1 なお、9 からは、コマンドウィンドウについてもキャプチャウィンドウと同様の位置変更が可能となりました。この点については弊社のツイッター (<https://twitter.com/lseviews>) でも簡単に解説しておりますので、もしよろしければそちらもご参照ください

このコマンドが正しいのかを検証してみます。せっかくワークファイルを作ったのですが、一旦閉じてしまいます（なお、閉じる際も wfclose という対応するコマンドがキャプチャされます）。先ほどキャプチャされたコマンドをコマンドウィンドウ側にコピーアンドペーストし、Enter キーを押します。ダイアログが表示されますので、そのまま Enter キーを押すか、あるいはマウスで Finish を押します。すると、先ほどと同じ形式でデータが読み込まれます。

これがコマンドキャプチャ機能の基本です。

パネルデータとして取り込む場合 上記では data1.xlsx を非構造データとして取り込みましたが、data1 はパネル構造を持っているので、パネルデータとして取り込むことも可能です。パネルデータとして取り込んだ場合に、キャプチャされるコマンドにもそのオプションが反映されるのかを確認してみましょう。先程取り込んだ非構造ワークファイルはもう使いませんので、閉じてしまいます。File>Open>Foreign Data as Workfile と操作し、ダイアログで「次へ」を 2 回押した後、以下のように設定して Finish を押しください（なお、左列の数字が?になっている場合、読み込み中ですので表示されるまでお待ちください）。Link ~ と表示されたら、No を選択します*2。

The dialog box 'Excel Read - Step 3 of 3' is used for configuring the import of data. It includes the following sections:

- Import method:** Create new workfile (dropdown)
- Import options:** Rename Series, Frequency Conversion (buttons)
- Structure of the Data to be Imported:**
 - Basic structure: Dated Panel (dropdown)
 - Frequency: Annual (dropdown)
 - Panel identifier series:
 - Cross section: prefecture (text input)
 - ID series: (empty text input)
 - Date series: year (text input)

Below the settings is a preview table showing the data structure:

	PREFECTURE	A	B	C01	D01	YEAR
1	1	10	100	10	10	2009
2	1	20	80	4	4	2010
3	1	15	60	3	2	2011
4	2	22	45	2	1	2009
5	2	43	41	5	2	2010
6	2	21	89	2	13	2011
7	3	5	70	14	14	2009
8	3	10	40	4	4	2010
9	3	4	38	2	4	2011

Buttons at the bottom: Cancel, <Back, Next>, Finish.

キャプチャウィンドウを確認すると、以下のようなコマンドが表示されているはずです。

*2 なお、説明の便宜上 No にしているだけで、Yes にすると Link Import に関するオプションが追加されたコマンドが出力されます

```
wfopen C:\data\CC\data1.xlsx range=Sheet1 colhead=1 na="#"N/A" @freq A @id prefecture @date(year) @smpl @all
```

先ほどと同じようにして、キャプチャされたコマンドを実行してみると、問題なく機能することが確認できます。比較してみると@freq A @id prefecture @date(year) という記述が追加されていることが分かります。このように、コマンドの根幹部分だけでなく、設定したオプション部分についても、キャプチャされることが分かります*3。

もちろん、時系列データに関してもキャプチャすることが可能です。興味がある方は、data2 を同じようにして読み込んでみてください。また、data2 には、1 行目に日本語の変数名が入っています。これらを避ける形でインポートを行い、その部分がキャプチャされることを確認してください。(wfopen C:\data\CC\data2.xlsx range=Sheet1!\$A\$2:\$E\$9 colhead=1 na="#"N/A" @freq A @id @date(year) @smpl @all となります)

3.2 統計表を作る

上で読み込んだパネルデータを使って、いくつかの統計表を作ってみます。まずは d01 を開き、View > Descriptive Statistics & Tests> Histogram and Stats と操作し、ヒストグラムを作ってみます。次のコマンドがキャプチャされます。

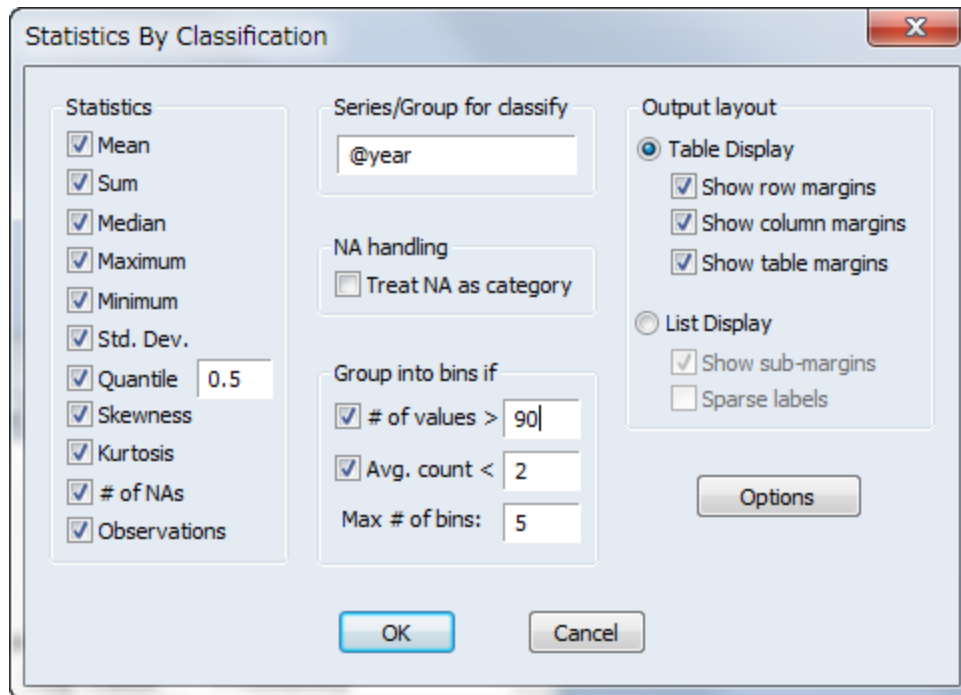
```
d01.hist
```

このコマンドをコマンドウィンドウで実行すると、ヒストグラムを作ることができます。

これではいささか簡単すぎるので、次のような例も試してみます。d01 で View > Descriptive Statistics & Tests> Stats by Classification と操作します*4。

*3 一部キャプチャされないコマンドもございますので、意図した出力が出ているかを確認する作業は忘れないようにしてください

*4 少し脱線となりますが、この機能は、特定の変数のカテゴリ別に統計表を作るために使うことができます。例えば月次データにおいて、@year を@month と入力して実行すれば、月別の統計表を作成することができます。非常に便利な機能ですので、是非とも使いこなしてみてください。

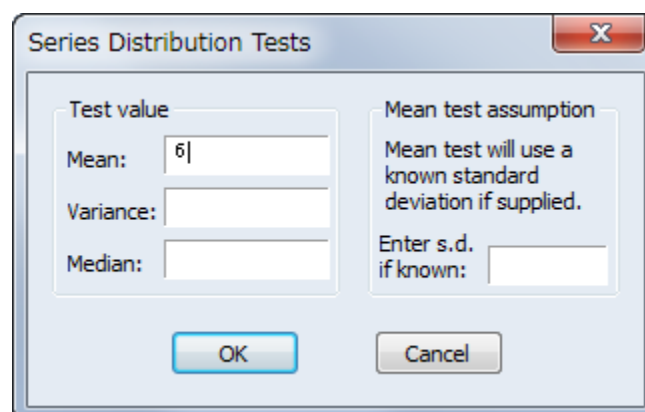


それぞれの年毎（これは、Series/Group for classify で、年を表すことができる@year を選択しているためです）に、左側で選択した統計量が出力されます。キャプチャされるコマンドは以下の通りです。

```
d01.statby(sum, med, max, min, na, skew, kurt, quandt=0.5, v=90) @year
```

statby がコマンド本体であること、() の中にオプションを書くこと、グループ変数を@year の位置に記述すればよいことなどが読み取れます。

簡単な検定も行ってみます。サンプルの平均が6であると統計的に言えるかどうかを確認してみます。View> Descriptive Statistics & Tests> Simple Hypothesis Tests と操作し、次のように入力します。



次のようなコマンドがキャプチャされます。

```
d01.teststat(mean=6)
```

これはそのまま動作します。このように、自身で入力した内容 (この場合数字の 6) に関してもキャプチャされます*⁵。試しに、Variance や Median にも数字を入れて、検定を実行してみてください。それらについても設定した数値までキャプチャがなされることが確認できるはずです。

3.3 グラフの作成について

開発元も認めている点ではございますが、コマンドキャプチャ機能はグラフの取り扱いが苦手です。そこで、グラフの作成に関してどの程度のことのできるのか、検証してみます。c01 と d01 でグループオブジェクトを作ります。c01 と d01 を ctrl キーを押しながら選択し、右クリックから open as group を選択してください。次のコマンドがキャプチャされます。

```
{%group}.sheet
```

このコマンドは、スプレッドシート形式で、%group という変数名で定義 (これはプログラムファイル上でないといけません) したオブジェクト (グループオブジェクトとは限りません) を開くという意味になります。この場合期待されるであろう、c01 と d01 で新しいグループオブジェクトを作るというコマンドとは異なりません。そのようなコマンド例は以下の通りです (c01 と d01 の両方を要素に含む grouptest という名前のグループオブジェクトを作成します)。

```
group grouptest c01 d01
```

コマンドキャプチャ機能には、キャプチャされるコマンドについてこのような癖がある場合がございます*⁶。キャプチャされたコマンドについては、お手数ですが一度動作をお確かめください。

grouptest と名前を付けたグループオブジェクトでグラフを作っていきます。散布図を regression line 付で作ってみます。View>Graph と操作し、Specific として Scatter を選び、Fit lines として Regression Line を選びます。以下のコマンドがキャプチャされます。

```
{%graph}.scat linefit()
```

{%graph} も上記の {%group} と意味はまったく同様で、このままでは動作いたしません。grouptest と置き換えることで、コマンドが動作するようになります。

他のグラフタイプもお試しく下さい。それぞれ、上記の .scat の部分が変わります。最も基本的な線グラフのコマンド .line くらいはご存じの方も多いかと存じますが、あまり使わないグラフタイプとなると、ご存じではない方もいらっしゃるのではないのでしょうか。そのような場合、EViews9 では、一度該当のグラフを作ってしまうと、対応するコマンドが分かります。また、複雑なカテゴリカルグラフに関して、例えば以下のようなコマンドをキャプチャすることができます。

*⁵ 上記の statby の例でも、# of values の値だけは変更したので、変更した後の値である 90 が記録されています。

*⁶ メニュー操作を行う場合、EViews は名前のついていないオブジェクトを扱うことができますが、多くのコマンド操作では名前のついていないオブジェクトを扱うことができません。そこで、こうした一般形の名前がついているとみなし、コマンドキャプチャが行われます。したがって、新規オブジェクトの作成を伴う操作が行われる場合は、キャプチャされたコマンドの実行前に、特に気を配ってください。

```
{%graph}.bar(m) across(@SERIES, iscale, iscalex, nodispname, label=auto, bincount=5) across(YEAR, cscale, cscalex, nodispname, label=auto, bintype=none, bincount=5)
```

(長すぎて PDF 上では 2 行になってしまいますが、EViews 上で実行する際は 1 行にまとめてください)

このように、グラフに関してもコマンドキャプチャが有用な部分が多々存在します。一方で、コマンドキャプチャには、グラフに関して、Graph Options>Graph Type 以外のオプション内容をキャプチャできないという特徴がございます。例えば、もう一度最初の散布図の操作を行い、Graph Element>Lines & Symbols の内容を適当に変更した上で OK を押してください。キャプチャされるコマンドは、

```
{%graph}.scat linefit()
```

のままです。つまり、線種や色のカスタマイズを伴うコマンドをキャプチャしたいと思っても、それは残念ながらできないということです。これがコマンドキャプチャ機能の、(現時点での) 主要な限界の一つです。

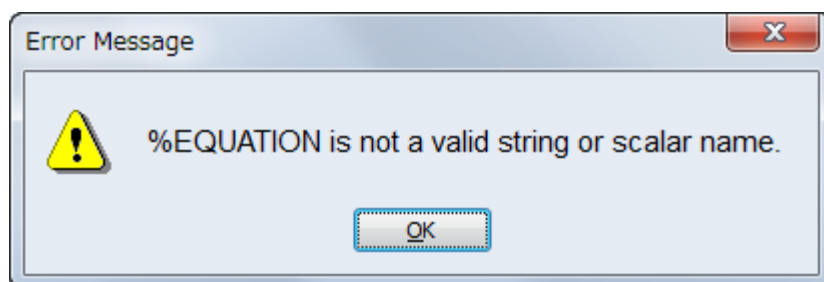
一方で、linefit() やカテゴリカルグラフの例でも明らかのように、Graph type 内部の設定に関しては、きちんと記録されます。また、Graph type 以外の設定に関しては、“グラフテンプレート”の機能(この機能も、将来的に技術資料で紹介したいと考えています)を用いることによって、保存することも可能です。コマンドキャプチャだけを考えると機能面の限界もございますが、便利に使える部分だけを抽出して使い分けることで、作業を最大限効率化することが可能です。

3.4 推定を行う

最後に、簡単な推定を行ってみましょう。Quick>Estimate Equation と操作し、a b c01 d01 と入力します。すると、以下のコマンドがキャプチャされます。

```
{%equation}.ls a b c01 d01
```

このコマンドをコマンドウィンドウに直接貼り付けても、以下のようなメッセージが出て機能しません。



グラフの場合と同様、{%equation} はプログラムファイルの内部で使われる表記ですので、上記のようなエラーが生じています。先頭部分を eqtest.ls 等に変更することで、このエラーを避けることができます。また、equation オブジェクトを新規に作成する場合は、equation がコマンドの最初に必要です。以下のように修正して実行してみてください。

```
equation eq01.ls a b c01 d01
```

今度は正しく推定されたはずですよ。

あるいは、以下のように ls 以下をはりつけても、推定が可能です（こちらの場合、名前がついていない推定式オブジェクトが作成されます）。

```
ls a b c01 d01
```

別の推定の例として、ステップワイズ回帰を行ってみます。Quick > Estimate Equation と操作し、STEPSLS を選びます。上の欄に a c b と入力し（この c は定数項の c です）、下の欄に c01 d01 と入力します。そうすると、以下のようなキャプチャ結果が得られます。

```
{%equation}.steps a c b @ c01 d01
```

ステップワイズ回帰を行う場合のコマンドが stepsls であること、ステップワイズ法で利用を検討する変数は@の後に記述すること、などが分かります。これらは上記の OLS 推定のものとは異なります。このように、推定手法ごとにどのようなコマンドを使うべきなのかが、一度手作業で推定を行っただけで分かります。また、推定の際 EViews では Options タブで多様なオプションを選択することができますが（ステップワイズ回帰の場合は特に）、オプション部分についてもほとんどはキャプチャされます。例えば以下のようにです。

```
{%equation}.steps(method=uni, ftol=0.2, btol=02, bmaxstep=800, nvars=2, back, tstat) a c b @ c01 d01
```

推定に関しては最初に挙げたような癖があることは事実ですが、実際に推定のオプションを色々と変えて試してみれば、有益な機能であることが実感できると考えます。まずは色々とお試し下さい。

4 Make Program

キャプチャしたコマンドをプログラム化することもできます。キャプチャウィンドウでプログラム化したい範囲を選択し、右クリックから Make Program をクリックします。上記の {%equation} 等はそのままでは利用することはできませんが、プログラムファイル内であれば、事前に以下のような記述を行えば実行することが可能になります。

```
%equation = "test"  
equation test
```

% が付く場合、それは文字列変数と呼ばれ、プログラムの内部でのみ機能します。1 行目で %equation の名称として test を指定しています（文字列なので、ダブルクォーテーションで囲います）。2 行目で test という名前の空の推定式オブジェクトを作成しています。これらのコマンドを事前に記述しておけば、

```
{%equation}.ls a b c01 d01
```

の {%equation} は test という名前のオブジェクトとして解釈され、test は既に空の推定式オブジェクトとして存在しますので、推定を実行することが可能です。

5 その他の情報

キャプチャウィンドウは、自由にフローティングさせたりドッキングさせたりすることができます。また、キャプチャウィンドウが表示されていない場合は、メインメニューの Window より Reset Windows Layout か、Display Command Capture Windows を選択してください。

キャプチャウィンドウ内でコマンドを実行することはできません。コマンドが動作しない場合、コマンドウィンドウではなく、キャプチャウィンドウになっていないかを一度お確かめください。しかし、General Options>Command Windows と操作し、Capture to Command Window にチェックを入れると、キャプチャウィンドウだけでなく、コマンドウィンドウにも該当操作がコマンドとして記録されます。コマンドウィンドウに多くの情報が集まることとなりますが、コマンドウィンドウ内であれば行にカーソルを合わせてエンターするだけで操作を再実行できますので、場合によっては便利です。

6 まとめ

今回の検証で分かった内容を簡潔にまとめると、以下の通りとなります。

- ・コマンド部分（選択したオプションや、入力した値を含む）のキャプチャに関しては、コマンドキャプチャ機能はかなり良く動作します
- ・コマンドの実行対象となるオブジェクト部分（. の前）に関しては、一般形の表記となってしまう、そのままでは動作しないコマンドになる場合がございます。その場合はオブジェクト部分を適宜書き換える必要がございます。

コマンドキャプチャ機能には限界もあるものの、それを差し引いても非常に便利な機能です。是非ご活用いただければと思います。

次回の技術資料では、日付データテーブル機能と、EViews9 の新機能である、日付データテーブルテンプレートについて解説します。