EViewsプログラミング入門

- 講座の目的
  - EViewsのプログラミングに必要な基礎知識を習得します。
    - コマンドとプログラム
    - 文字列変数と数値変数
    - よく利用する3つの構文
    - EViewsオブジェクトのデータメンバ
    - モンテカルロシミュレーション
    - テキストファイルの連結

eviews09フォルダを利用。

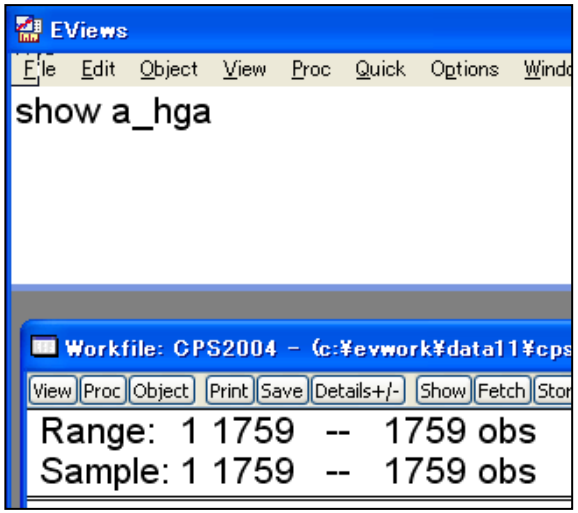
LightStone Corp
2020年9月

コマンドウィンドウの利用

操作：eviews09にあるcps2004.wf1を開きます。コマンドウィンドウに次のコマンドを入力してEnterキーを押します。

show a_hga

シリーズオブジェクトa_hgaのウィンドウを開きます。コマンドウィンドウでは独立した1行のコマンドを実行します。



*show コマンドは「アクションコマンド」の一種です。

show
do
freeze
print

プログラムファイルを利用する

操作1：File/New/Programと操作して新規プログラムファイル画面を表示します。
操作2:Program画面に次のコマンドを入力します。そして、Runボタンをクリックして実行します。Run Programダイアログでは、そのまま、OKボタンをクリックします。

```
!x=@mean(age)
series test1=!x*10
scalar test2=!x*0.1
```

!xは後述する数値変数。
@mean(シリーズオブジェクト)は平均値を求めるEViewsの関数です。

EViewの関数コマンド

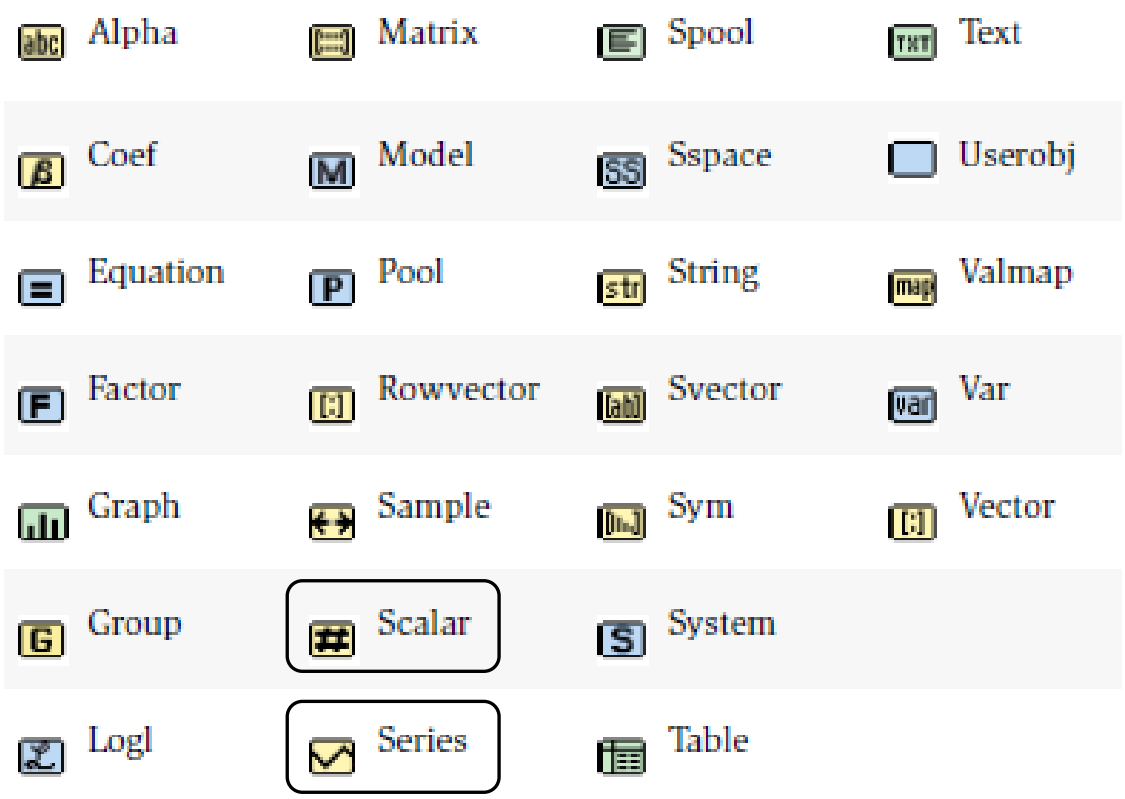
@で始まるコマンドはEViewsの関数コマンドです。関数は次のようにカテゴリ分けされています。

演算子	特別な関数
基本的な数学関数	三角関数
金融関数	統計分布関数
記述統計量	文字列関数
累積統計量関数	日付関数
移動統計量関数	指標関数
グループの行関数	ワークファイルの情報関数
グループ毎の統計量	バリューマップ関数

関数の意味を確認する場合は、Help/EViews Help Topicsとして、キーワードタブで検索します。

EViewsオブジェクト

EViewsオブジェクトを作成する場合は、最初に次に示すオブジェクトの種類を、次に自分で決めたオブジェクト名を入力します。



バッチ処理

- 学歴を示すコードa_hgaの値を教育年数に一括変換する!

a_hga=32	ed=2.5
a_hga=33	ed=5.5
a_hga=34	ed=7.5
a_hga>34 and a_hga<39	ed =a_hga – 26
a_hga = 39	ed =12
a_hga>39 and a_hga<43	ed =14
a_hga=43	ed =16
a_hga =44	ed =18
a_hga>44	ed =20

バッチ処理

- 例えばa_hgaが32という値を持つ人にはed=2.5という値を入力します。

操作1：プログラムファイルの内容を全て削除し、改めて次のコマンドを入力します。そしてtransformという名前でプログラムファイルを保存します。

```
series ed=0
smpl @all if a_hga=32
ed=2.5
```

操作2：正しく動作したら、a_hga=33以下のプログラム文も入力します。2つの条件もそのまま入力します。最後の行に次のコマンドを追加します。Runボタンをクリックしてプログラムを実行してください。

```
smpl @all
```

Transform.prg

```
'教育年数edの作成プログラムです。
series ed=0  '最初にシリーズオブジェクトを作成します。
smpl @all if a_hga=32
ed=2.5

'a_hga=32という条件に合致するデータだけを対象にし、
'シリーズedに2.5という値を入力します。

smpl @all if a_hga=33
ed=5.5
smpl @all if a_hga=34
ed=7.5
smpl @all if a_hga>34 and a_hga<39
ed=a_hga-26
smpl @all if a_hga=39
ed=12
```

Transform.prg

*前のページから続けて書きます。

```
smpl @all if a_hga>39 and a_hga<43
ed=14
smpl @all if a_hga=43
ed=16
smpl @all if a_hga=44
ed=18
smpl @all if a_hga>44
ed=20
smpl @all
```

確認と保存

操作1:edのウィンドウでView/One-Way tabulationと操作してそのままOKボタンをクリックします。

edを正しく作成できると、度数を示すCountの合計がサンプルの個数と一致しています。

Tabulation of ED
Date: 01/30/09 Time: 09:18
Sample: 1 1759
Included observations: 1759
Number of categories: 12

Value	Count	Percent	Cumulative Count	Cumulative Percent
0.000000	3	0.17	3	0.17
2.500000	7	0.40	10	0.57
5.500000	26	1.48	36	2.05
7.500000	7	0.40	43	2.44
9.000000	21	1.19	64	3.64
10.00000	32	1.82	96	5.46
11.00000	58	3.30	154	8.75
12.00000	462	26.26	616	35.02
14.00000	619	35.19	1235	70.21
16.00000	357	20.30	1592	90.51
18.00000	115	6.54	1707	97.04
20.00000	52	2.96	1759	100.00
Total	1759	100.00	1759	100.00

操作2:プログラムを上書き保存します。

重要:Runボタンをクリックすると、EViewsは入力されたコマンドをすべて解釈して実行します。つまり、実行前に必ずしも保存する必要はありません。

プログラムのポイント

- 標本の選択にはsmplコマンドを利用する

smpl @all…全標本
smpl @all if 条件文… 条件に合致するデータ(行)のみを標本とする

コマンド例)
smpl @all if a_hga>39 and a_hga<43
ed=14

注意!
条件をandやorで接続する場合は、条件をすべて書く
誤り)
smpl @all if a_hga>39 and <43

オンラインヘルプ

- コマンドの利用法はオンラインヘルプで調べます

操作1：メインメニューからHelp/EViews Help Topics…と操作します。
操作2：キーワードのテキストボックスにsmplと入力します。

キーワードを入力するだけですぐに候補となる情報を表示します。

ヘルプの表示方法

- 情報の表示方法に慣れておきましょう。

コマンドの文法(Syntax)とオプションやキーワード、そして例題(Example)を見れば、利用方法は把握できます。

smpl	Global Commands

Set sample range.

The `smpl` command sets the workfile sample to use for statistical operations and series assignment expressions.

Syntax

smpl *smpl_spec*

smpl *sample_name*

List the date or number of the first observation and the date or number of the last observation for the sample. Rules for specifying dates are given in "Dates". The sample spec may contain more than one pair of beginning and ending observations.

コメントを付ける

- コメントを付ける習慣を身に付けましょう。

操作1：コメントを付けたい所に半角入力モードでアポストロフィー「’」を入力します。正しく入力した文字列は、画面上、緑色で表示されます。
操作2：確認したら、プログラムファイルを閉じます。

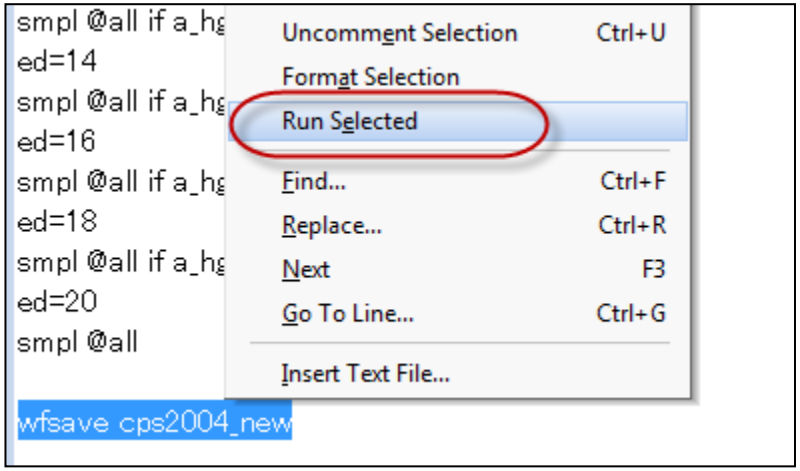
印刷時の注意点：

プログラム画面を印刷する場合、General OptionsのFonts/Editor – Text Objectsで日本語フォントを選択しておきます。

プログラムの一部を実行する

プログラムを修正、追加した場合、目的のコマンド行だけを実行することも可能です。

操作:次のコマンドをプログラムの最後に追加し、図のように、コマンドを選択後、右クリックしてRun Selectedコマンドを実行します。

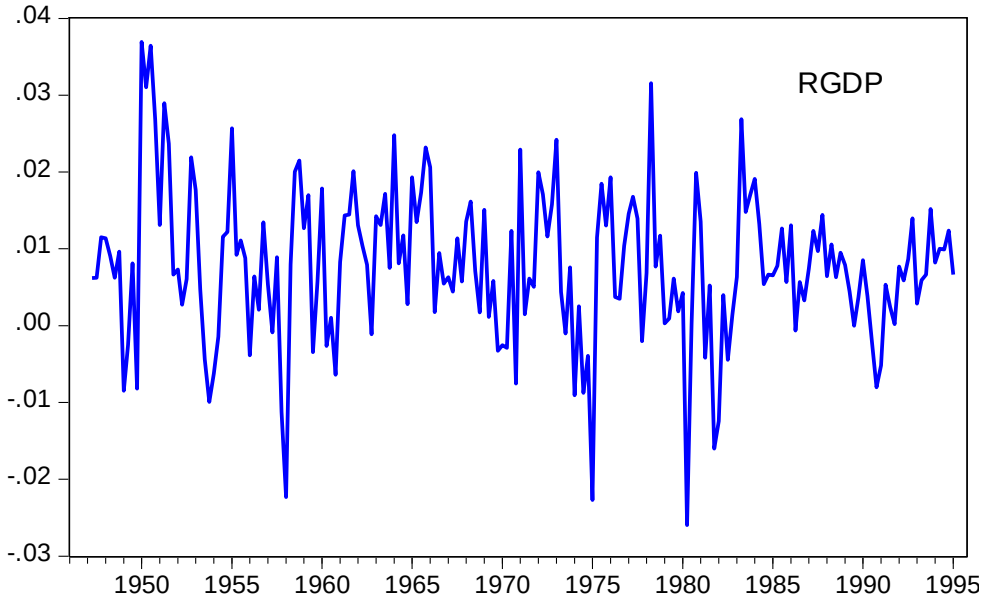


時系列データ

- 時系列データにおける実証分析で一番重要なEViewsコマンドはsmplです

操作:EViewsファイルchow_var.wf1を開きます。このワークファイルは四半期データです。次のコマンドでgdpの変化率のシリーズを作成します。

```
series rgdp=dlog(gdp)
show rgdp.line
```



smplコマンド

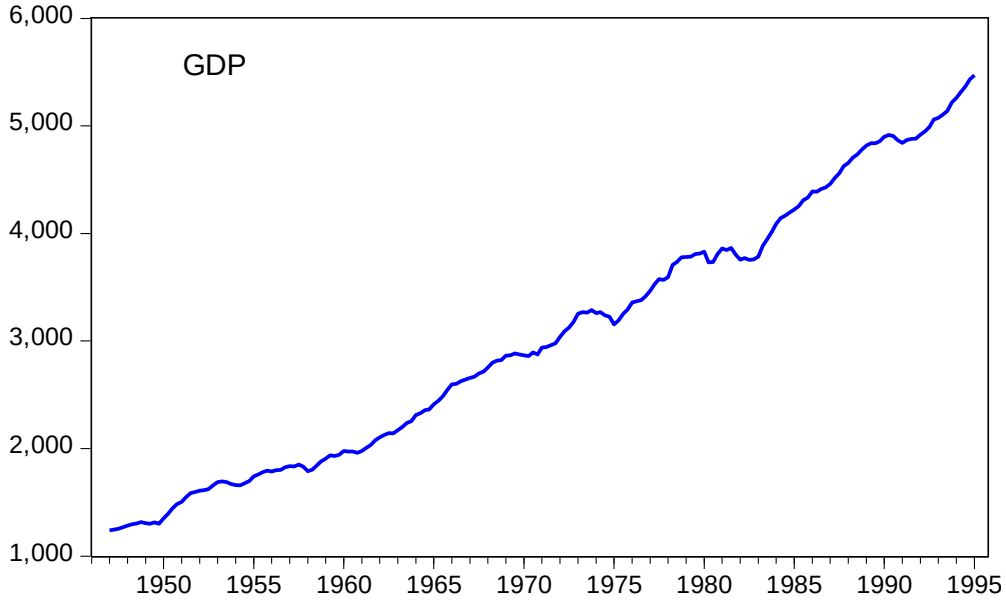
操作1:rgdpのウィンドウを閉じます。そして、gdpのグラフを表示します。

show gdp.line

操作2:次のコマンドでgdpの変化率が正のデータだけグラフ化してみましょう。

smpl @all if rgdp>0

*操作1のコマンドで showは必ずしも必要ではありません。



smplコマンド

操作3:1960Q1から1985Q4までの間で、gdpの変化率が正のデータだけグラフ化する場合は次のようにします。

```
smpl  1960q1  1985q4 if rgdp>0
```

注意)始期が1で、終期が4の場合、敢えて明記しなくてもかまいません。

```
smpl  1960  1985 if rgdp>0
```

時間情報

操作1:次のコマンドで全期間でのgdpの分布を確認しましょう。

```
smpl @all
gdp.hist
```

操作2:第一四半期に限定して調べてみましょう。

```
smpl @all if @quarter=1
```

ワークファイルに時間情報が設定されている場合、時間情報を取得する@コマンドが用意されています。

@year,@month,@day,@weekday,@hour,@minute,@second,
@hourf

時間情報

もちろん、次のように組み合わせて利用することもできます。

```
smpl @all if @quarter=1 and @weekday=1
```

第一四半期で、かつ、月曜日のデータに限定する

オフセット

- 以下は入門コースでの内容ですが、確認です。

smpl 1950q1 1950q1+7	(1950年-1951年末まで)
smpl @first+1 @last	(1946Q2-1995Q4まで)
smpl @first @first	(1946Q1のみ)
smpl @first @last-1	(1946Q1-1995Q3まで)

Time Series Function

- 時系列データにおいて階差や対数階差を取るためのコマンドはオンラインヘルプのキーワード検索で「time series functions」として用意されています。

d(x)
dlog(x)
@pc(x)
@pch(x)

まとめ

- バッチ処理 transform.prg
- smpl コマンドの用法
- オンラインヘルプの参照
- コメント文
- プログラムの一部を実行
- 時系列データでのsmplコマンドの用法

参考)プログラムファイルの取り扱い

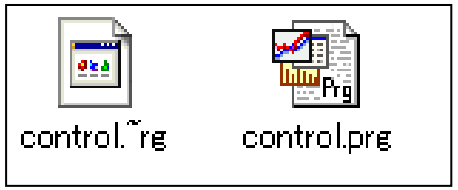
- プログラムファイルの基礎知識

プログラムファイルの注意事項!

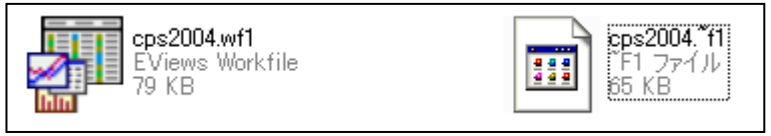
- ダブルクリックはしない！
- メール添付の際は圧縮するか、拡張子を変更する！

参考)バックアップファイル

EViewsのプログラムを実行すると、そのフォルダの中に拡張子の異なるアイコンが作成されます。これはプログラムのバックアップファイルです。何らかの理由で元のプログラムが壊れたり、削除してしまった場合は、バックアップファイルの拡張子をprgに変更します。



同じようにワークファイルについても図のようなバックアップファイルを自動作成します。こちらも拡張子をwf1にすれば、通常のEViewsワークファイルとして利用できます。



MEMO

文字列変数と数値変数

- 文字列変数と数値変数について学びます。これはEViewsプログラミングを行う上で最も重要な基礎知識です。

文字列変数

1.回帰式の説明変数を文字列変数に設定して、文字列変数の特徴と置換機能を理解します。
2.文字列の合成方法を学びます。

推定

操作1：cps2004.wf1に戻ります。最初にメインメニューでQuick/Estimate Equationと操作して次のモデルEQ01を推定します。

lnwage c ed

操作2:新しいプログラムファイルreg1で同じモデルをeq02として推定します。

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.004161	0.100311	10.01046	0.0000
ED	0.116773	0.007107	16.43133	0.0000

R-squared	0.133197	Mean dependent var	2.623455
Adjusted R-squared	0.132704	S.D. dependent var	0.843017
S.E. of regression	0.785091	Akaike info criterion	2.355102
Sum squared resid	1082.959	Schwarz criterion	2.361325
Log likelihood	-2069.312	Hannan-Quinn criter.	2.357402
F-statistic	269.9886	Durbin-Watson stat	1.857576
Prob(F-statistic)	0.000000		

equation eq02.ls lnwage c ed

推定

次に示すように空のequationオブジェクトを最初に作成し、2行目で、その内容を設定する、という方法もあります。

```
‘単回帰モデルの推定
equation eq02
eq02.ls lnwage  c  ed
```

次に示すように空のequationオブジェクトを最初に作成し、2行目で、その内容を設定する、という方法もあります。

オブジェクト名eq02の次にくる.lsで推定手法を設定します。

操作:オンラインヘルプで、equationオブジェクトのmethodについて調べてみましょう。

文字列変数と置換機能

- 文字列変数を使って、説明変数を選択できるようにしてみましょう。

操作：プログラムファイルreg1を次のように書き換えて実行します。

```
‘単回帰モデルの推定
%x1=“ed”
equation eq03.ls  lnwage  c {%x1}
```

■文字列変数は記号%を先頭につけて利用します。
■ダブルクォーテーションで囲んだ文字列は赤で表示されます。
■文字列変数の値を利用する場合は大カッコで囲みます。これを置換機能と呼びます。

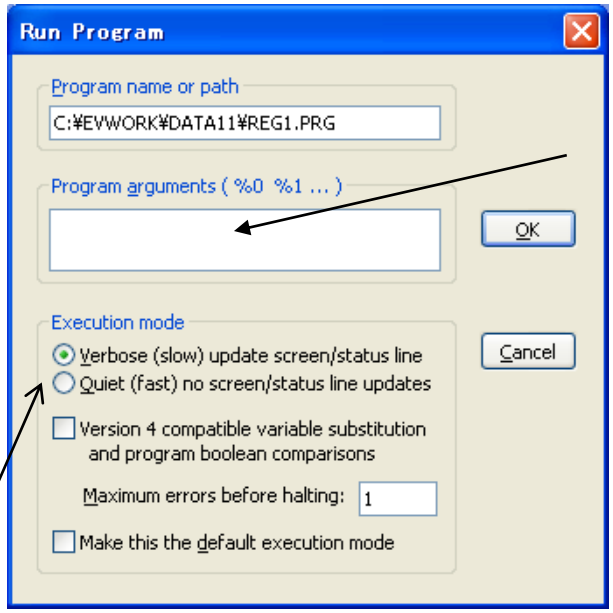
プログラムの実行ダイアログ

- ダイアログの利用方法

操作1:プログラムを次のように変更します。他の行はコメントアウトします。

equation eq04.ls lnwage c {%0}

操作2:プログラムを実行し、引数のウィンドウにedと入力してOKボタンをクリックします。



繰り返し計算回数が多い時はQuietを選択します。

プログラムの実行ダイアログで引数を設定する場合は、図にしめすテキストボックスを利用します。

文字列変数

問題：引数をedとageとする重回帰モデルeq05を作成してください。
文字列変数%0と%1を利用します。

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.556956	0.102688	5.423766	0.0000
ED	0.093610	0.007065	13.25065	0.0000
AGE	0.019585	0.001574	12.44048	0.0000

操作：図に示すようなeq05がプログラムで作成できたら、プログラムファイル「reg1」を上書き保存します。

文字列変数の操作

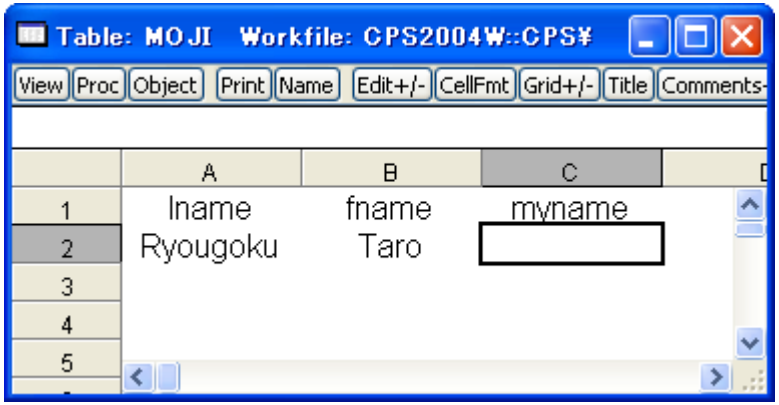
- 文字列を表形式で表示する場合はテーブルオブジェクトを利用します。

操作1:コマンドウィンドウで次のコマンドを実行し、6行3列のテーブルオブジェクトmojiを作成します。

table(6,3) moji

操作2:テーブルオブジェクトをダブルクリックして開きます。そして、次のコマンドで図のように5つの文字列を入力します。

moji(1,1)="lname"



*文字列の先頭はエルです。

文字列の結合

操作：次の要領でプログラムファイル「names」を作成します。

コマンド1行目：文字列変数%x1にテーブル2行1列目の情報を格納します。「%x1=moji(2,1)」

コマンド2行目：%x2には2行2列目を格納します。

コマンド3行目：%x3で%x1と%x2を結合します。ただし、両者の間には半角スペースをダブルクォーテーションで囲んで入れます。「%x3=%x1+“ ”+%x2」

コマンド4行目：最後に%x3を2行3列目に入れます。

Point:テーブル操作の場合、置換機能は利用しない。

names.prg

```
'文字列変数%x1と%x2にRyogokuとTaroを
'格納します。
%x1=moji(2,1)
%x2=moji(2,2)

'文字列を合成します。ただし、途中に半角
'スペースを入れます。
%x3=%x1+" "+%x2
moji(2,3)=%x3
```

アルファシリーズオブジェクト

- アルファシリーズオブジェクトは数値を代入するシリーズオブジェクトと同じように利用できます。

操作1：コマンドウィンドウでアルファオブジェクトnamelistを作成します。

alpha namelist

操作2：同じくコマンドウィンドウを利用して、次のコマンドを実行します。

namelist(1)="Text1"

アルファシリーズの1行目にText1という文字列が入ります。テーブルオブジェクトとアルファシリーズを目的に応じて使い分けてください。

練習問題1

- 次に示すコマンドを実行すると、mojiの4行1列目にはどのような文字列が入るでしょうか?プログラムファイル名はstrings1とします。

```
‘記号!で始まる変数は数値変数です。
‘@str()、@leftの意味はオンラインヘルプで確認します。
!repeat=500
%st1=“ draws from the normal”
%st2=“ t”
%st3=@str(!repeat)+@left(%st1,16)+%st2+ ” distribution”
moji(4,1)=%st3
```

文字列変数のまとめ

- 変数名Xの前に記号%を付けます。
- 置換機能を利用する場合は{%X}のようにします。
- 文字列変数は+記号で結合できます。
- 文字列を格納する場合はアルファシリーズオブジェクトか、またはテーブルオブジェクトを利用します。

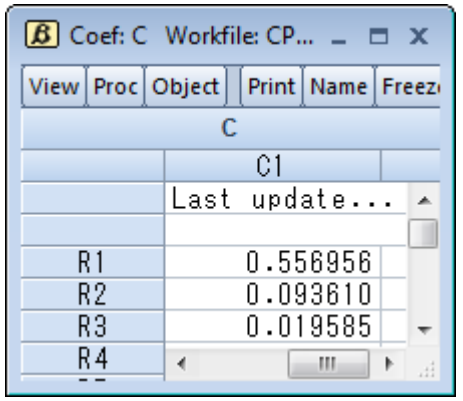
係数オブジェクト C

操作1：EQ05でView/Representations と操作します。

Estimation Command:
==========
LS LNWAGE C ED AGE

Estimation Equation:
==========
LNWAGE = C(1) + C(2)*ED + C(3)*AGE

推定値を示すC()は係数オブジェクトに格納されていることを示します。数字は行番号です。アイコンをダブルクリックし、推定値を確認しましょう。



Equationオブジェクトのデータメンバ

操作:Statsボタンをクリックして推定結果の画面に戻ります。

Dependent Variable: LNWAGE
Method: Least Squares
Date: 06/24/14 Time: 10:01
Sample: 1 1759
Included observations: 1759

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.556956	0.102688	5.423766	0.0000
ED	0.093610	0.007065	13.25065	0.0000
AGE	0.019585	0.001574	12.44048	0.0000

R-squared	0.203405	Mean dependent var	2.623455
Adjusted R-squared	0.202498	S.D. dependent var	0.843017
S.E. of regression	0.752839	Akaike info crite...	2.271774
Sum squared resid	995.2426	Schwarz criterion	2.281107
Log likelihood	-1995.025	Hannan-Quinn criter.	2.275223
F-statistic	224.1912	Durbin-Watson stat	1.844415
Prob(F-statistic)	0.000000		

推定値以外だけでなく、すべての情報はeq05に含まれています。それらは「データメンバ」と呼びます。

Equation:データメンバ

Equationオブジェクトのデータメンバの一部

@aic 赤池情報量基準	@df 推定式の自由度
@dw Durbin-Watson統計値	@f　F統計値.
@logl 対数尤度関数の値	@ncoef 推定した係数の個数
@pval(i) i番目の係数のp値	@r2 決定係数
@rbar2 自由度修正済み決定係数	
@regobs 回帰式で利用したデータの個数(行数)	
@schwarz シュワルツの情報量基準	
@sddep 被説明変数の標準偏差	
@se 回帰の標準誤差	@ssr 残差の二乗和
@stderrs(i) i番目の係数の標準誤差	
@tstats(i) i番目の係数のt値またはz値	

*データメンバ(data members)を調べる場合はオンラインヘルプを活用。

Equation:データメンバ

操作1：推定値はオンラインヘルプに@coefs(i)とあります。これを使って次のプログラムdatambs.prgを作成してみましょう。

```
table(1,3) myb

myb(1,1)="c(1)"
myb(1,2)="c(2)"
myb(1,3)="c(3)"

myb(2,1)=eq05.@coefs(1)
myb(2,2)=eq05.@coefs(2)
myb(2,3)=eq05.@coefs(3)

show myb
```

数値変数

- 数値変数は先頭に「!」を付けます。文字列変数同様、プログラムの中だけで利用します。

操作：新しいプログラムウィンドウを開き、次のプログラムを実行します。名前は「control」とします。

```
!x=7
!12345=0
!counter=12
!pi=3.14159

!y=!x+!12345
!counter=!counter+1
!s=2*!pi*!counter
```

*数値変数のことをコントロール変数とも呼びます。

シリーズとスカラーオブジェクト

操作1：プログラムに次の3行を追加して、実行してください。

```
scalar v1=!y
scalar v2=!counter
series v3=!s/!pi
```

数値はスカラーオブジェクトとシリーズオブジェクトの、どちらにも直接代入できます。スカラーオブジェクトとは数値が1つだけ入る、数値の入れ物です。内容を確認する場合はスカラーオブジェクトをダブルクリックしてスカラーオブジェクトのウィンドウを開きます。

数値変数のまとめ

- 変数名Xの前に記号!を付けます。
- 数値変数のことをコントロール変数ともいいます。
- 数値を格納する場合はシリーズオブジェクトやスカラーオブジェクト、さらには後述する行列オブジェクトなどを利用します。
- 数値の比較には等号や不等号を組み合わせて利用します。

X>Y	XはYより大
X>=Y	XはY以上
X<Y	YはXより大
X<=Y	YはX以上
X=Y	XとYは等しい
X<>Y	XとYは等しくない

よく利用する3つの構文

- IF条件文
- FOR LOOP
- WHILE LOOP

IF条件文

- 変数の値によって処理を変更する場合はIF条件文を利用します。

```
if  条件文  then
        (条件文が真の場合)
else
        (条件文が偽の場合)
endif
```

練習：スカラーv1の値を5とします。いま、v1が5より大きければLarger than 5、5以下ならばLess or equal to 5と、moji(6,1)に表示するプログラム「ifstat」を作成してください。

IF条件文

IF条件文のサンプルプログラムifstat.prg

```
'スカラーオブジェクトv1に関するif条件文の作成
scalar v1=5
if v1>5 then
        moji(6,1)="Larger than 5"
else
        moji(6,1)="Less or equal to 5"
endif
```

FOR ループ

- 同じ処理をn回繰り返し実行する場合はFORループを利用します。

```
for  !i=1 to n
            (繰り返しの処理)
next
```

練習：forループを利用して、1から始まり1759で終わるシリーズオブジェクトnumsを作成してください。コントロール変数には!iを利用します。シリーズオブジェクトの要素はnums(!i)として指定します。プログラムファイル名は「forstat」とします。

FORループ

```
'シリーズオブジェクトnumsを作成します。
series nums

'ループの開始
for !i=1 to 1759

        'numsの!i行目に整数!iを入力します。
        nums(!i)=!i
next
```

EViewsの機能をうまく使う

- プログラムforstatで作成したプログラムと同じ処理をコマンドウィンドウで、次の3行で実行することもできます。3行目コマンドはどのように書けばいいでしょうか？

```
series nums2=1
smpl @first+1  @last
nums2=??????
```

練習問題2

問題1.シリーズオブジェクトageの値が35歳以上ならA course、35歳未満ならB courseという文字列の入るアルファシリーズcourseを作成するプログラム「health_check」を作成してください。

問題2.問題1が正常に処理できたら、ageとcourseをグループオブジェクト「checklist」を作成し、画面に表示するコードを追加します。

ヒント)グループオブジェクトを作成するコマンドはgroup、画面に表示するコマンドはshowです。

問題が完了したらEViewsワークファイルを上書き保存します。

練習問題2

```
'文字列を格納するアルファシリーズを作成します。
alpha course

'forループの開始
for !i=1 to 1759

        if age(!i)>=35 then
        course(!i)="A course"
        else
        course(!i)="B course"
        endif

next
```

練習問題2

プログラムの続きを入力します。

```
'グループオブジェクトの作成
group checklist age course

'グループオブジェクトの表示
show checklist
```

自習用課題

標本期間でループコマンドを利用する練習用課題です。

操作1：InwageでView/Descriptive Statistics&Tests/Histogram and Statsと操作して、ヒストグラムを表示し、平均値が約2.62である事を確認します。
操作2:コマンドウィンドウで次のようにサンプル期間を変更します。平均値がサンプル期間によって変わることを確認します。

smpl 1 10 ⟹ 2.62

smpl 11 20 ⟹ 3.00

*小数点以下3桁で四捨五入。

自習用課題

シリーズオブジェクトの記述統計量は関数コマンドで取得できます。例えば次に示すプログラム「loops」でシリーズオブジェクトmyavgの1行目にlnwageの1番から10番目までの平均値を入力してください。

```
series myavg
smpl 1 10
myavg(1)=@mean(lnwage)
smpl @all
```

プログラムでサンプル期間を操作したら、作法として最後に全範囲に戻します。

@mean(シリーズ名)はシリーズオブジェクトの平均値を計算するコマンドです。

自習用課題

問題：プログラムファイルloopsを次のように変更してください。
myavgの1番目から10番目まで、最初の10個分(1から10番目まで)、2行目には11番目から20番目まで平均値、最後は91番目から100番目までの平均値をプログラムで自動入力します。

ヒント1：forループでsmplコマンドを変更して、その都度、データメンバから平均値を求めてください。
ヒント2:smplコマンドの引数でカッコを伴う演算はできません。中間変数を利用してください。

loops

```
series myavg
'smpl 1 10
'myavg(1)=@mean(lnwage)
'smpl @all

'ここからが練習問題のプログラム
for !i=1 to 10

        !k=(!i-1)*10+1
        !m=!i*10
'smplの引数の中で演算はできませんので、
'あらかじめ、!k、!mとして計算しておきます。
```

loops

プログラムの続きを入力します。

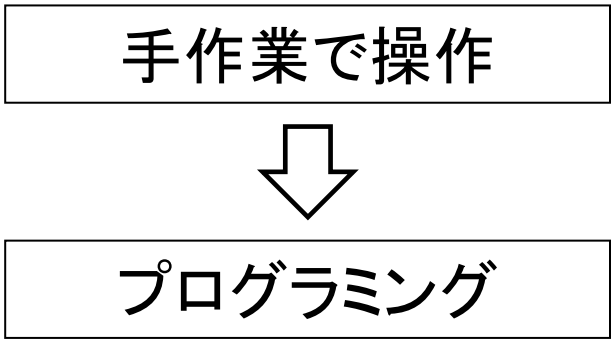
```
            smpl  !k !m
            myavg(!i)=@mean(lnwage)
next

smpl @all
'作法として最後にsmpl期間を全期間に戻します。
```

知識の整理

- ここまで学習したEViewsのプログラミングの知識を整理しましょう。

ローリング回帰

推定期間を1期づつ移動しながら、推定値の変化を観察することが目的です。



Forループによるローリング回帰

操作：サンプルファイルrollingdemo.wf1を開きます。そして、次のモデルをEQ01としてOLSで推定します。

$$\log(m1) = \beta_0 + \beta_1 \log(gdp) + \beta_2 rs + \beta_3 \,\mathrm{dlog}(pr)$$

推定期間:1952Q1-1981Q4(30年)

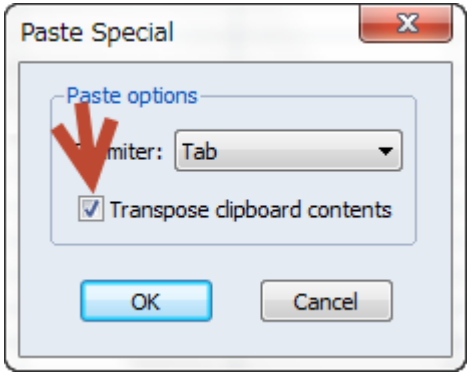
Dependent Variable: LOG(M1)
Method: Least Squares
Date: 05/20/13 Time: 17:42
Sample (adjusted): 1952Q2 1981Q4
Included observations: 119 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	2.033564	0.054619	37.23205	0.0000
LOG(GDP)	0.609100	0.012138	50.18284	0.0000
RS	-0.002846	0.002288	-1.243819	0.2161
DLOG(PR)	3.841357	0.885553	4.337804	0.0000

メニュー操作によるR回帰

操作1：2行4列のmatrixオブジェクト「test」を作成します。

操作2:次の図に示すように推定値をtestの画面に貼り付けます。ただし、転置して貼り付けますので、貼り付けの際に、右クリックして「Paste Special」コマンドを利用します。そして、オプションの



TEST

	C1	C2	C3	C4
	Last updated: 01/13/15 - 08:27			
R1	2.033564	0.609100	-0.002846	3.841357
R2	0.000000	0.000000	0.000000	0.000000

Dependent Variable: LOG(M1)
Method: Least Squares
Date: 05/20/13 Time: 17:42
Sample (adjusted): 1952Q2 1981Q4
Included observations: 119 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	2.033564	0.054619	37.23205	0.0000
LOG(GDP)	0.609100	0.012138	50.18284	0.0000
RS	-0.002846	0.002288	-1.243819	0.2161
DLOG(PR)	3.841357	0.885553	4.337804	0.0000

メニュー操作によるR回帰

操作3:次に推定期間を1952Q2-1982Q1に変更して再度、推定を行います。ただし、推定期間は次のように入力します。

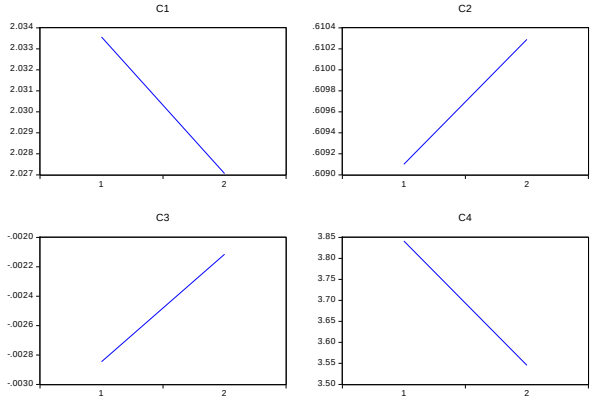
1952q1+1 1981q4+1

操作4:推定結果を同じ要領で、2行目に貼り付けます。

Last updated: 01/13/15 - 08:25

2.033564	0.609100	-0.002846	3.841357
2.027063	0.610290	-0.002115	3.545380

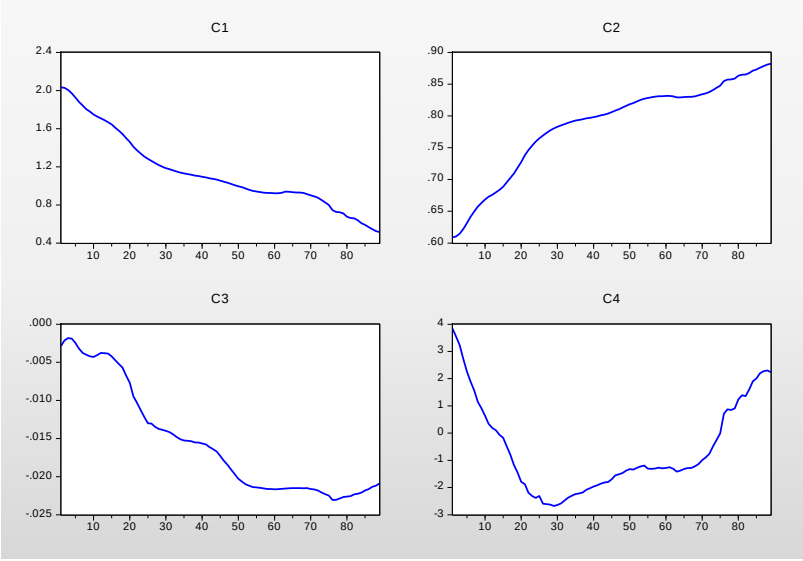
操作5:行列オブジェクトtestで View/Graphと操作し、Detailsにある Multiple Seriesのオプションで「Multiple graphs」を選択し、OKボタンをクリックします。



ローリング回帰プログラム

操作：File/Open/Programと操作し、rollingprog.prgを開きます。Runボタンをクリックして実行すると、ローリングの最初の始期を1952Q1、終期は1981年Q4。最後の始期を1974Q1、終期を2003年Q4として、22年分(88四半期)のローリング回帰を実行し、推定値の変化をグラフとして表示します。

$$\log(m1) = \beta_0 + \beta_1 \log(gdp) + \beta_2 rs + \beta_3 \,\mathrm{dlog}(pr)$$



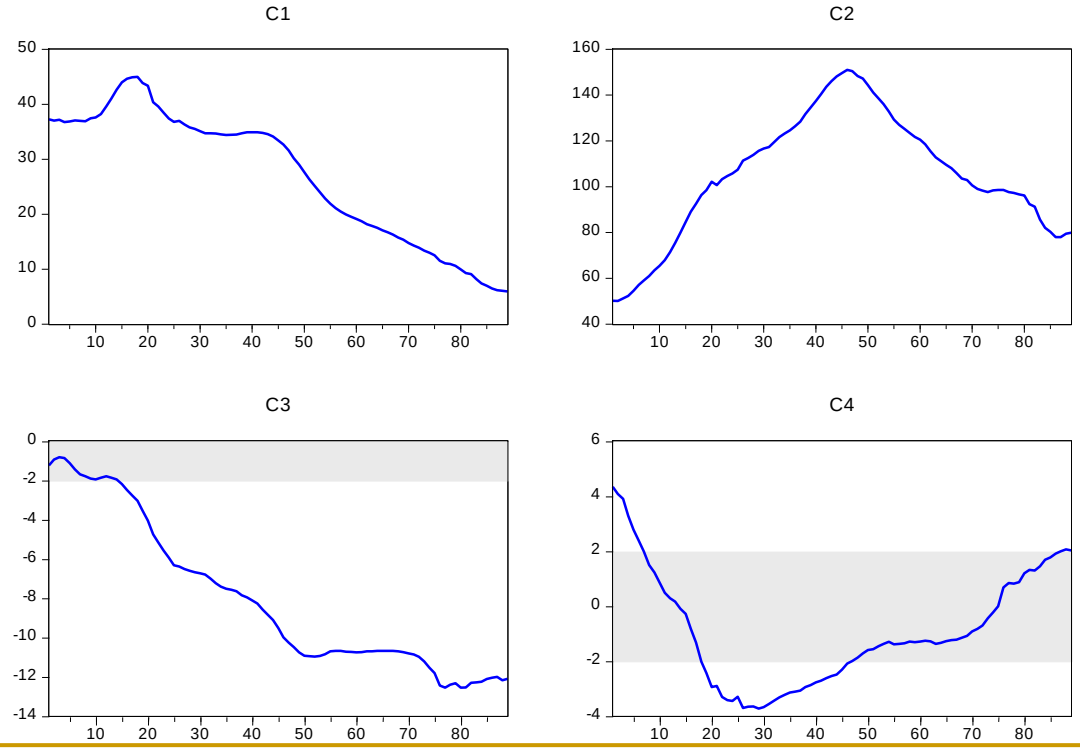
ローリング回帰プログラム

```
smpl @all

matrix(89,4) betas

for !i=0 to 88
        smpl 1952q1+!i 1952q1+!i+119
        equation model1.ls log(m1) c log(gdp) rs dlog(pr)

        !j=!i+1
        betas(!j,1)=c(1)
        betas(!j,2)=c(2)
        betas(!j,3)=c(3)
        betas(!j,4)=c(4)
next

show betas.line(m)
```

プログラムの編集

練習：rollingprog.prgでは推定値を行列オブジェクトbetasに格納しました。さらに、推定値のt値を行列オブジェクトtstatに格納し、グラフを作成するプログラムコードを追加してください。

ヒント:t値はEquationオブジェクトのデータメンバです。



回答はextstat.prg

WHILEループ

- FORループは繰り返し回数がある整数で決まっている
- ある条件を満たしている(条件が真の)場合に、繰り返しを行う場合はWhileループを利用する

WHILEループ

操作:basics.wf1を開きます。次にサンプルプログラムwhileloop.prgを開きます。

```
'Whileループの簡単な例

'wfopen basics
'1960年1月から1970年2月までの期間を最初の計算期間として、
!val<10かつ!a<10を満たす限り、シリーズinc**を作成するプログラム

!val=1
!a=1
```

WHILEループ

```
while !val<10000 and !a<10

        smpl 1960m1 1970m1+!a
        series inc{!val}=m1/!val
        !val=!val*10
        !a=!a+1

wend
```

*inc1,inc10,inc100,inc1000が作成されます

WHILEループ

tb3の累積和stb3のシリーズを求めるプログラムwhileloop2.prg

操作:次にサンプルプログラムwhileloop2.prgを開きます。

```
smpl @all

!a=1
series stb3

smpl @first 1969m12
!sum=@sum(tb3)
```

WHILEループ

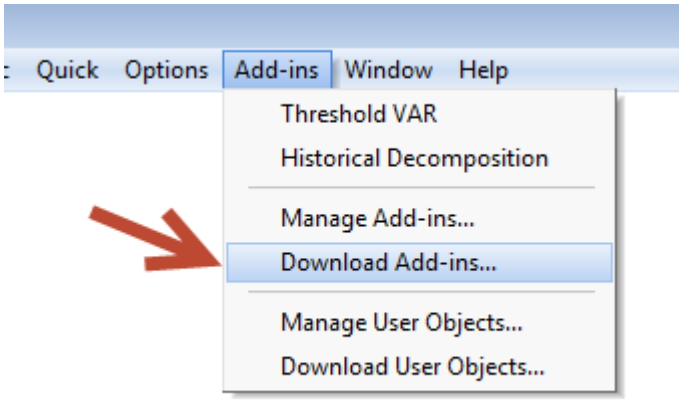
```
while  !sum<650
            smpl @first 1970m1+!a
            !sum=@sum(tb3)
            smpl 1970m1+!a 1970m1+!a
            stb3=!sum
            !a=!a+1
wend

group group01 tb3 stb3
smpl @all if stb3<>na

show group01
```

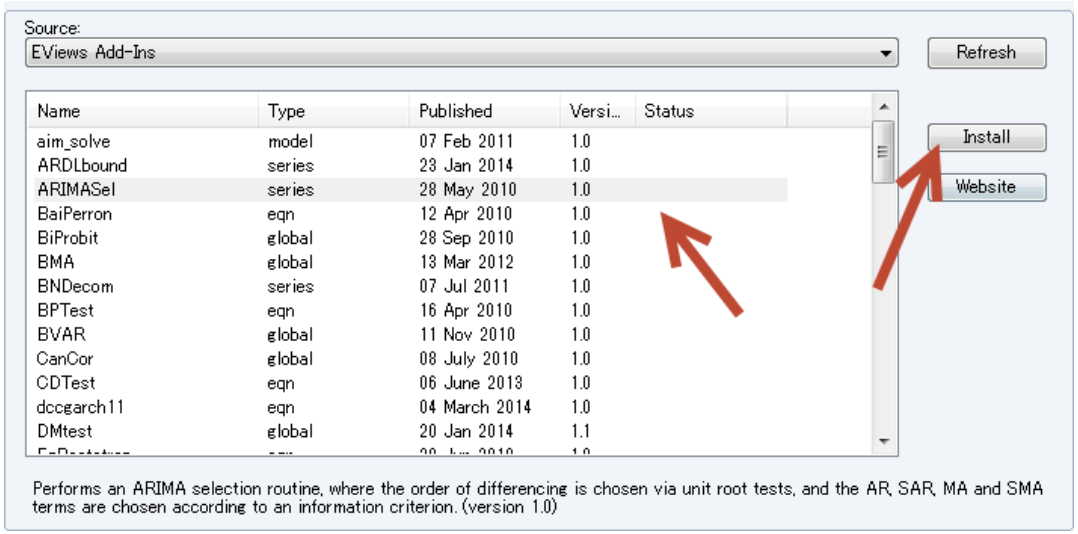
ループのまとめ

- Forループ
  - 繰り返し回数を数値変数で指定
- Whileループ
  - 条件が満たされている間は繰り返しを継続

EViews Add-ins

EViewsユーザの作成したプログラムで機能(コマンド)を拡張する。





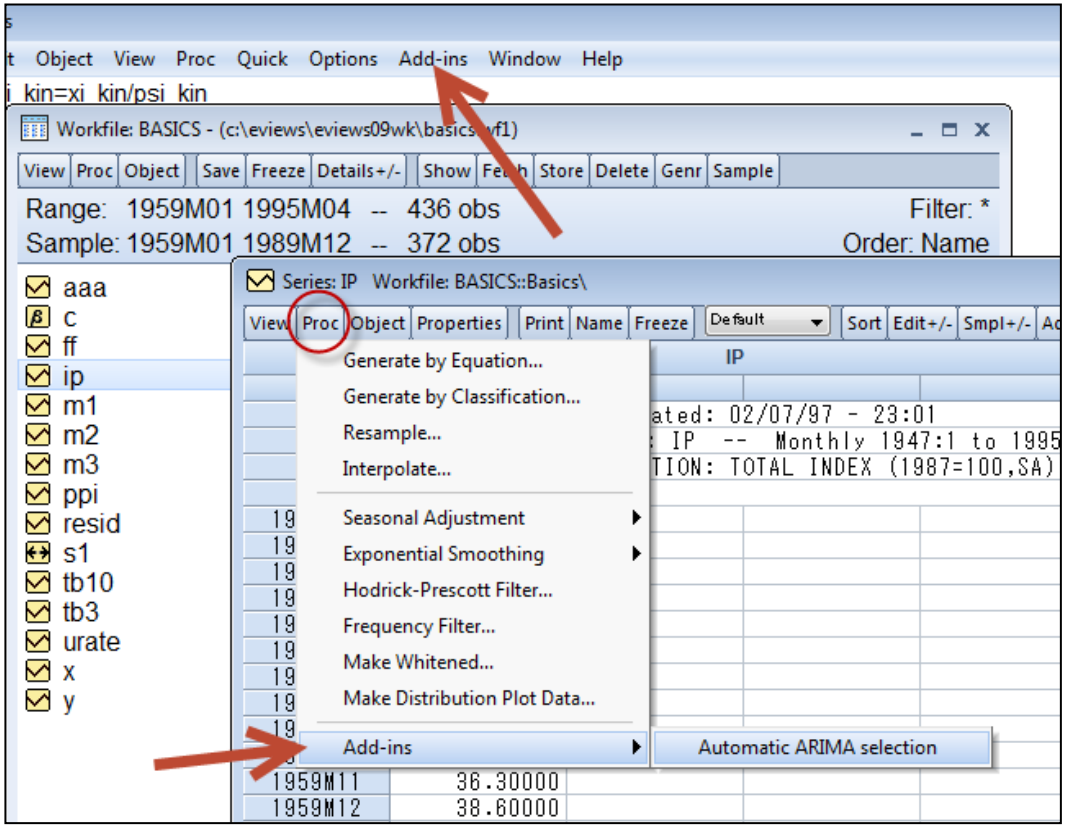
ARIMASelは時系列データ(シリーズオブジェクト)のARIMAモデルの仕様を自動決定するプログラム。

目的のadd-inを選択してInstallをクリック。

EViews Add-ins

Add-insは、一般的に、そのプログラムが利用できるオブジェクトのAdd-insメニューに追加されるが、メインメニューのAdd-insメニューにも追加される。

注意:Add-insはテクニカルサポートの対象外。



乱数の作成

- プログラムでよく利用する乱数発生コマンド。

標準正規分布：nrnd
平均0、分散1の分布から乱数を作成します。分散を4とするような場合は2×nrndとします。

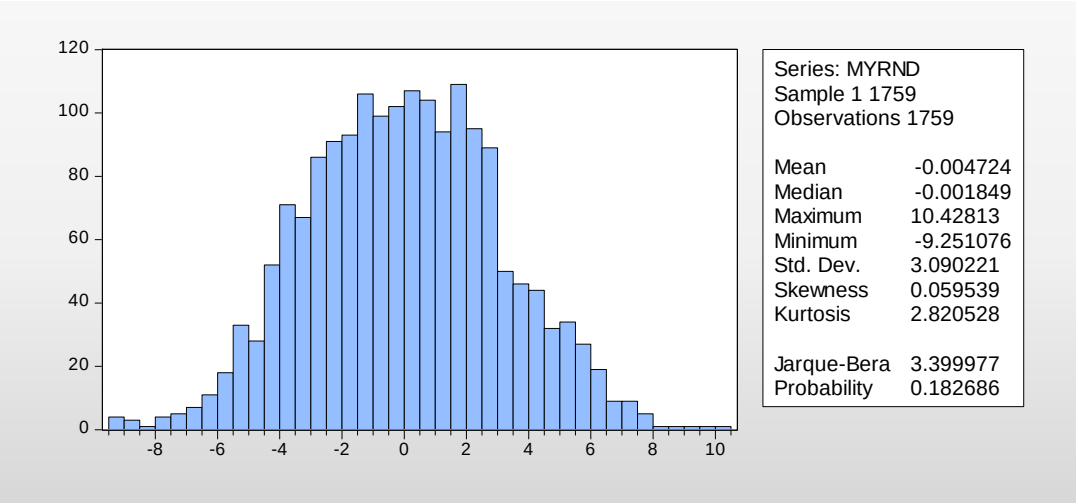
一様分布：rnd
区間[0, 1]で 乱数を作成します。生起確率は等しいものとします。
区間を[0,6]とする場合は6×rndとします。

重要：乱数の発生を時間に関係なく、統一する場合は乱数発生のシード(種)を、次のコマンドで予め設定します。引数の整数は何桁でもかまいません。どんな整数でも利用できます。

rndseed 123456

乱数

操作：ワークファイルcps2004.wf1において、次のスライドに示すプログラム「rndgen」を入力してください。
このプログラムは乱数キーを123として、分散を9とする乱数(正規分布)のシリーズmyrndを作成し、次のようなヒストグラムを表示する



平均がゼロ、標準偏差が3になっているでしょうか。

rndgen.prg

```
'乱数の作成プログラム
'分散が9なので、標準偏差は3

rndseed 123
series myrnd=3*nrnd

'ヒストグラムの表示
show myrnd.hist
```

モンテカルロシミュレーション1

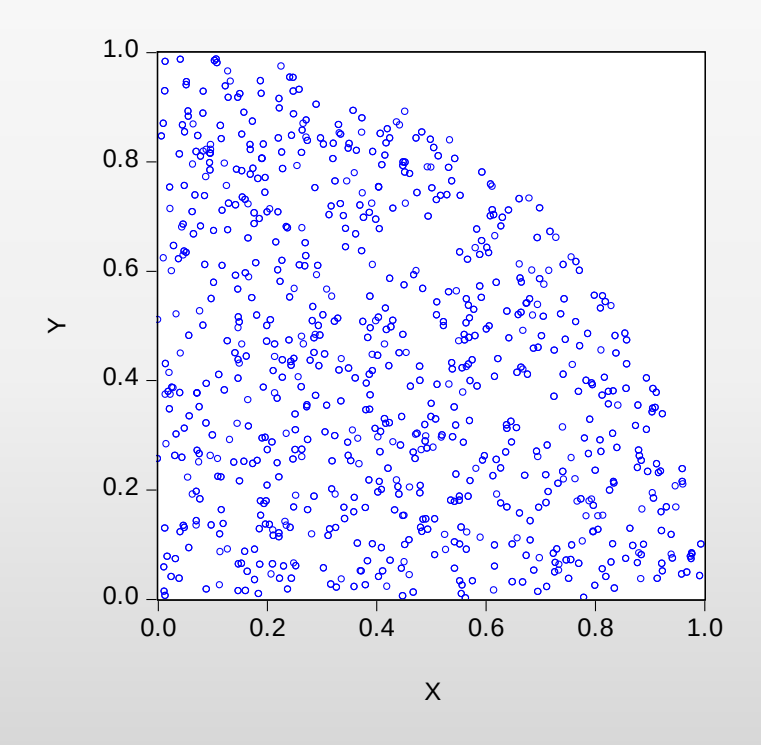
- 円周率Piの近似値をモンテカルロシミュレーションで求めます。

$$\frac{S_c}{S_{sq}} \simeq \frac{obs_c}{obs_{sq}}$$

この近似式を利用すると、半径1の単位円では、

$$\frac{\pi/4}{1} = \frac{obs_c}{obs_{sq}}$$

$$\pi = \frac{obs_c}{obs_{sq}} \times 4$$



getpi.prg

```
'ワークファイルの作成
wfcreate(wf=getpi) u 1000

'2次元平面の第一象限において座標xyのペアを1000個
'作成する
series x=rnd
series y=rnd
series r=@sqrt(x^2+y^2)

'単位円の内部に含まれる点に限定
smpl @all if r<=1

'グループcircleを作成
group circle x y
```

getpi.prg

```
'第一象限の点の個数は1000個
'四分円の中の点の個数を!nとする
'四分円の面積!sqの近似値は

!n=@obssmpl
'次式は近似式
!sq=1*(!n/1000)

'よって!piは

!pi=4*!sq

show circle.scat
show !pi
```

ワークファイルの操作

目的：eviews09¥stocksに10個のテキストファイル(day1.txtからday10.txtまで)があります。このファイルを取り込む自動的に取り込んで、最終的にeviews09にあるmydata.wf1を作成するプログラムを作成しましょう。

最初に、メニュー操作で「day1.txt」と「day2.txt」を一つのEViewsワークファイルにしてみましょう。

操作1:day1.txtをForeign data as Workfileとして開きます。

操作2:New Pageタブをクリックして別ページとしてday2.txtを開きます。

操作3:day1のページに戻ってday2をproc/appendします。

テキストファイルの連続読み込み

imptwks.prgにコメントをつけてみましょう。

```
cd c:¥eviews¥eviews09¥stocks
wfopen(type=txt) day1.txt
tic
for !i=2 to 10
        %txtnm="day"+@str(!i)+".txt"
        pageload {%txtnm}
        pageselect day1
        %wfnm="day"+@str(!i)
        pageappend {%wfnm}
        pagedelete {%wfnm}
next
wfsave mydata
toc
```

自習用課題

- パラメータの分布を作成する

シミュレーションの目的:

$$y = \beta_1 + \beta_2 * x + e$$

上式で係数とxの推定値を予め用意します。誤差項eを乱数機能で作成し、yを求めます。このxとyに対して上式の回帰を実行することで、新たな係数を得ることができます。
この作業を何回も繰り返し、数多くの係数を算出します。係数の「分布」を調べることを目的とします。

基本方針

- 次のような条件でプログラムファイルmotec.prgを作成します。

□ワークファイルはUnstructured/Undatedで、データの個数は100個とします。wfcreateコマンドを利用します。
□X値は80,100,120,140,160,180,200,220,240,260で固定とします(10個)。シリーズオブジェクト名をxとし、x.fillコマンドを利用します。
□定数項beta1は2.5、傾きbeta2は0.5とします。残差の平均は0、分散1の正規分布とします。
□推定を100回行います。beta1とbeta2の値を格納するシリーズオブジェクト名はb1,b2とします。繰り返しの回数は!repsに代入して利用します。
□最後にb1とb2のカーネル分布のグラフを次のコマンドで作成し、それぞれshowコマンドで表示します。

```
freeze(gra1) b1.distplot kernel
```

montec.prg

```
'Unstructured/Undatedのワークファイルを作成します。
wfcreate(wf=mcarlo) u 100

'値の固定されたシリーズxを用意します。
series x
x.fill 80,100,120,140,160,180,200,220,240,260

'係数の推定値を設定します。
!beta1=2.5
!beta2=0.5

'次の課題のことも考えて繰り返し回数は数値変数
'として設定します。
!reps=100
```

montec.prg

```
for !i=1 to  !reps
        smpl 1 10
        series y=!beta1+!beta2*x+3*nrnd
        equation eq1.ls y c x
        smpl !i !i
'シリーズオブジェクトb1,b2の!i行目に
'係数値を入力するためにサンプル期間
'を変更しました。

        series b1=eq1.@coefs(1)
        series b2=eq1.@coefs(2)
'ここは、
'series b1=c(1)
'などとしてもOKです。

next
```

montec.prg

```
'推定した係数値のカーネルプロットを
'作図します。
smpl 1 !reps
freeze(gra1) b1.distplot kernel
show gra1

freeze(gra2) b2.distplot kernel
show gra2
```

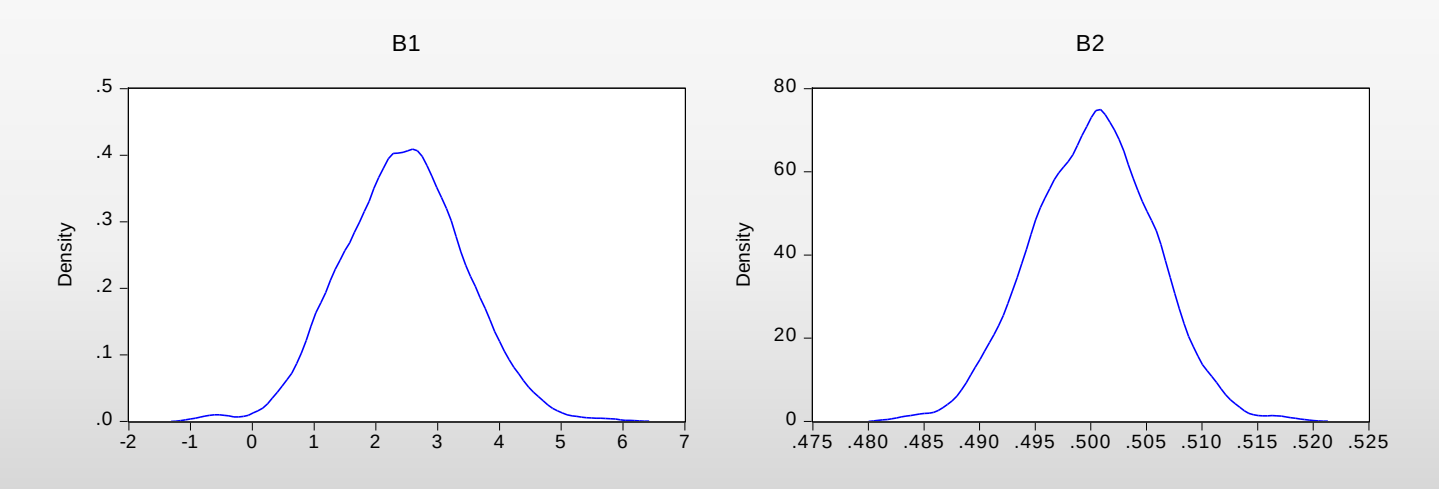
係数推定量のグラフ

- モンテカルロシミュレーションによって、点推定として与えた係数の区間推定量を視覚的に得ることができました。
- 100個の係数を単純にヒストグラムで表示しても良いのですが、それをスムーズ化した密度で表示することで、分布がはっきりと分かります。
- プログラムは乱数機能を使っていますので、実行するごとにグラフの形状は異なります。

操作：シミュレーションの回数を1000回とするプログラムmotec2を作成してください。乱数の発生には「123456」というシードを設定してください。また、ファイル名はmcarlo2とします。

シミュレーションの回数

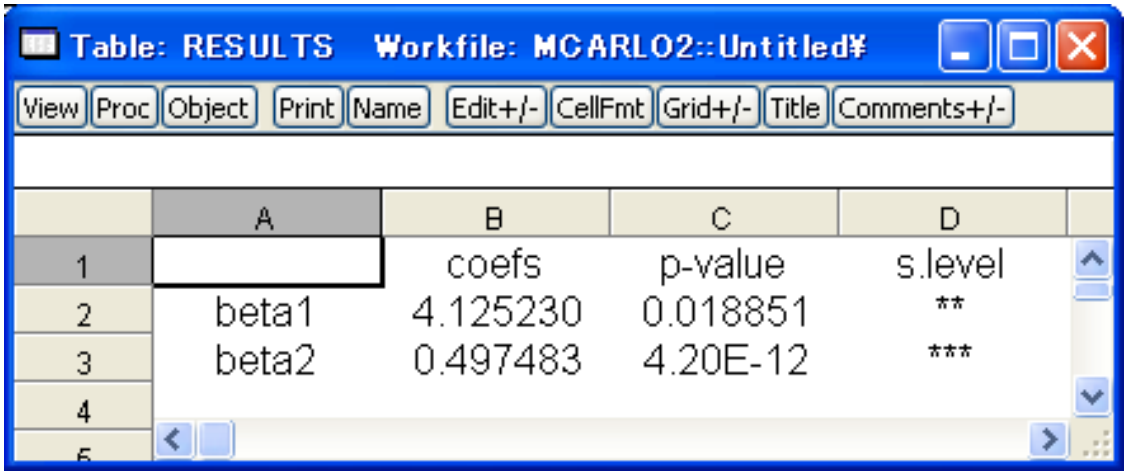
練習：シミュレーションの回数を1000回とするプログラムmotec2を作成してください。乱数の発生には「123456」というシードを設定してください。また、ファイル名はmcarlo2とします。



シミュレーションの回数を1000回にした時のグラフです。回数をさらに増やしていくと、グラフはどのような形になっていくでしょうか。

練習問題4

motec2のプログラムを実行した結果のワークファイルmcarlo2に対して、次のような表を作成するプログラムmtable.prgを作成して下さい。



	A	B	C	D
1		coefs	p-value	s.level
2	beta1	4.125230	0.018851	**
3	beta2	0.497483	4.20E-12	***
4				

1%有意水準は***、5%なら**、10%では*と、有意水準10%で有意でない場合はnsとします。1%有意水準とはp値が0.01未満とします。p値の計算と、それに応じたアスタリクの作成は次のスライドのヒントを参考にしてください。

ヒント

ヒント1：p値はデータメンバ@pval(i)です。

ヒント2：p値の判定は例えば、次のような形で行います。

```
!pv=@pval(i)
if !pv<value1 then ----
        else
        if !pv<value2 then ---
                else
                if !pv<value2 then---
                else ----
                endif
        endif
endif
```

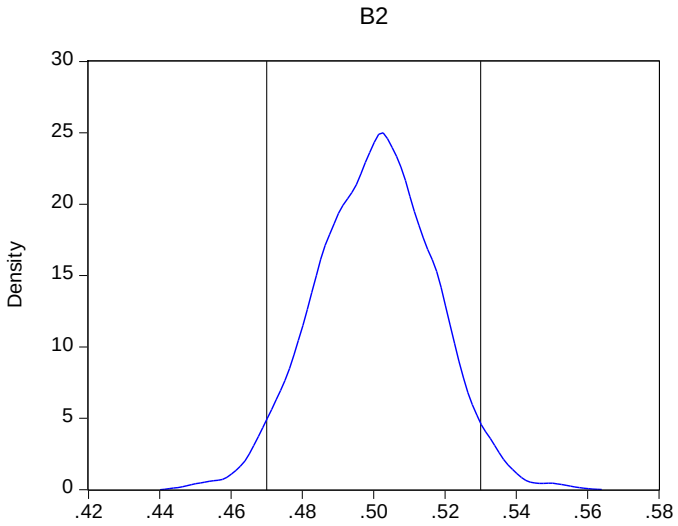
Quantile(分位点)

最後に知識として分位に関する次のコマンドを操作して、先ほど計算した係数b2の95%境界値を探してみましょう。プログラミングは行いません。

series u95=@quantile(b2,0.975)

series b95=@quantile(b2,0.025)

このようにして求めた値で右のような図を作成してください。線を加えるコマンドは、図を右クリックして表示するAdd lines & shadingです。



mtable.prg

```
'motec2の結果として作成されたmcarlo2を
'画面に表示しておきます。

'表の作成
table(3,4) results

results(1,2)="coefs"
results(1,3)="p-value"
results(1,4)="s.level"
results(2,1)="beta1"
results(3,1)="beta2"

results(2,2)=c(1)
results(3,2)=c(2)
```

```
'p値の計算
!p1=@tdist(@tstat(1),@regobs-@ncoef)
!p2=@tdist(@tstat(2),@regobs-@ncoef)

'有意水準の表示判定
for !j=1 to 2
if !p{!j}<0.01 then %ast="***"
        else
        if !p{!j}<0.05 then %ast="**"
                else
                if !p{!j}<0.1 then %ast="*"
                        else
                        %ast="ns"
                endif
        endif
endif
```

```
'p値とアスタリクの代入
results(1+!j,3)=!p{!j}
results(1+!j,4)=%ast

next
```

EViewsデータベースファイル

EViewsのデータベースファイルには観測度数の異なるデータを一緒に保存できます。

⇩

必要に応じて作業中のワークファイルにインポートする。

⇩

Frequencyは自動変換（Options/General Options/Series and Alpha/Frequency conversion）

メニュー操作

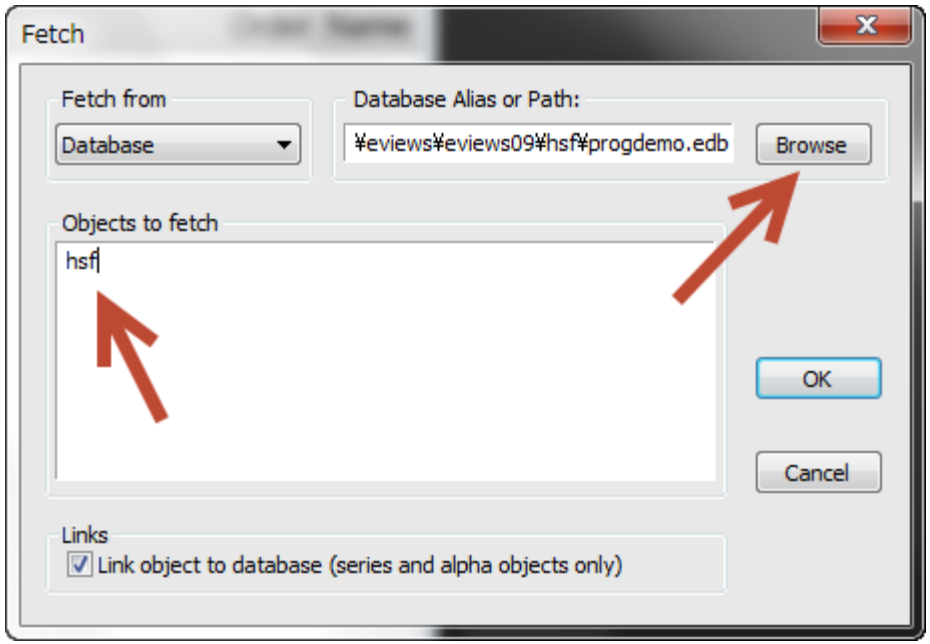
自己回帰モデルで短期予測を行うプログラムを作成します。最初に目的の処理をメニューコマンドで行い、内容を理解します。

操作1:データを分析するための月次ワークファイルmyhouse(1968m3-1997m6)を新規作成します。

操作2:住宅着工数の時系列データが下記のフォルダの中にEViewsデータベースファイルprogdemoとして入っています。

ワークファイウィンドウでマウスを右クリックして、fetch from DB…を選択して次に示すダイアログを表示します。

メニュー操作



操作3:Browseボタンをクリックして目的のデータベースファイルを選択します。

c:¥eviews¥eviews09¥hsf¥progdemo.edb

操作4:Object to fetchの項目にhsfと入力し、OKボタンをクリックします。

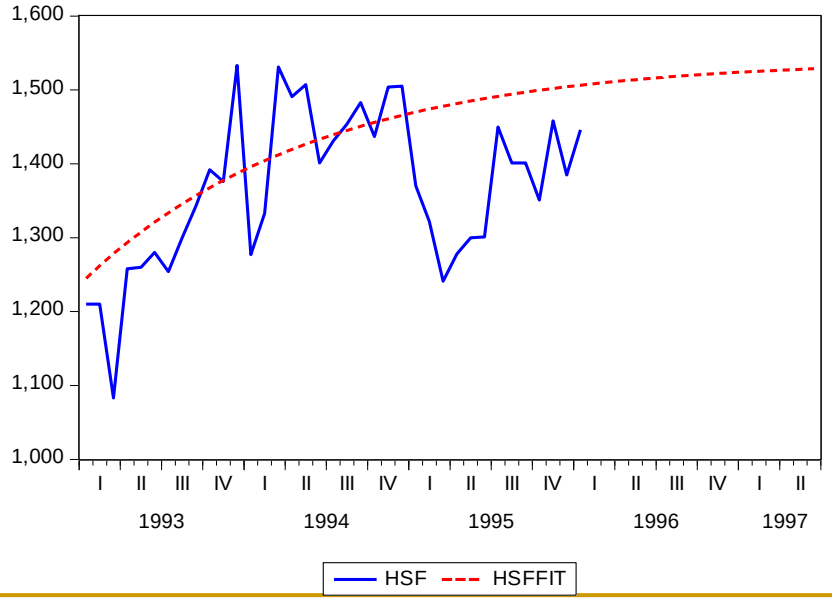
データベース

操作4:サンプル期間を1968m5-1992m12に変更して、1次の自己回帰モデルreg1を推定します。

hsf c hsf(-1)

操作5:予測のためにサンプル期間を1993m1-1997m6に変更します。forecastボタンをクリックし、ダイナミック予測を実行します。予測シリーズ名はhsffitとします。

操作6:hsfとhsffitのグループgroup01を作成し、グラフを表示します。



プログラミング

操作:プログラムファイルmyhsfにコメントを付けてみましょう。

```
cd c:¥eviews¥eviews09¥hsf
wfcreate(wf=myhsf) m 1968m3 1997m6
fetch progdemo::hsf
smpl 1968m5 1992m12
equation reg1.ls hsf c hsf(-1)
smpl 1993m1 1997m6
reg1.forecast hsffit
smpl @all
group group01 hsf hsffit
show group01.line
```

基礎と実用-まとめ

- EViewsプログラムの作成に必要な基礎知識を次の点について学びました。
  - EViewsプログラムファイル
  - 文字列変数と数値変数
  - IF条件文とFORループ
  - モンテカルロシミュレーション

参考文献

- EViews User's Guide I
- EViews User's Guide II
- EViews Illustrated
- EViews Command Reference