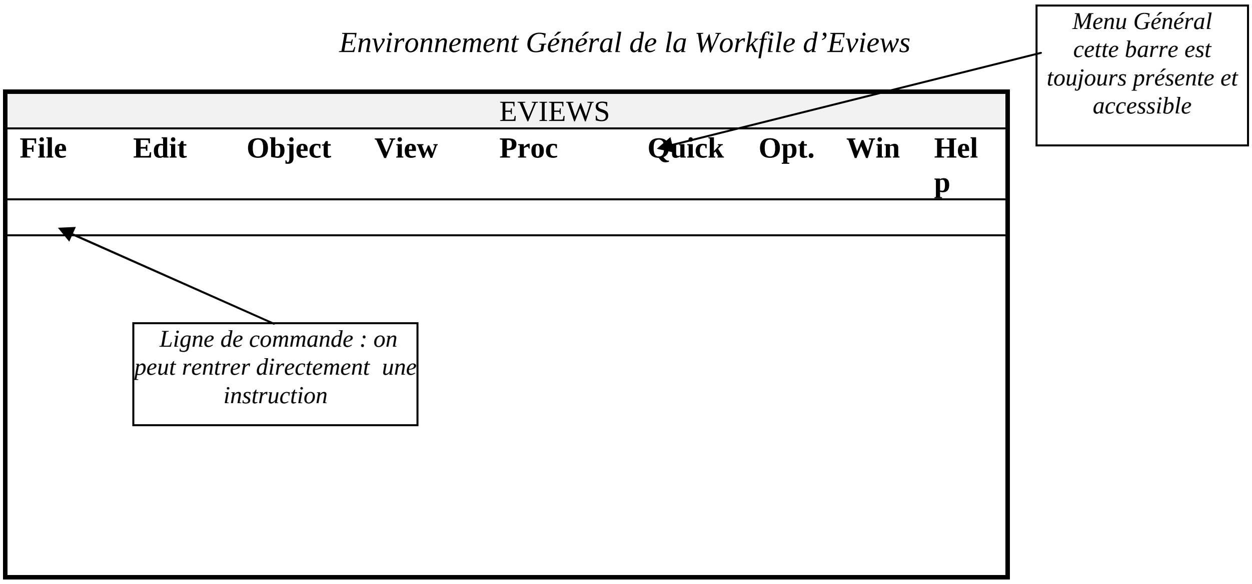
Régis Bourbonnais
Université de Paris–Dauphine
(Mise à jour Janvier 2006)

LOGICIEL EVIEWS

Ce mode d'emploi du logiciel EVIEWS (**http:\\www.eviews.com**) est un guide d'apprentissage à partir du manuel de Régis Bourbonnais, **« Économétrie : manuel et exercices corrigés », Dunod, 6ème éd., 2006**. Les données des exercices et les programmes peuvent être téléchargés sur le site ***http:\\www.cip.dauphine.fr/bourbonnais*** puis cliquer sur le livre Econométrie. Les noms des fichiers correspondent au numéro de chapitre et numéro d'exercice : par exemple C3EX2 = Exercice 2 du chapitre 3.
Merci de me signaler (sur mon E–mail : Regis.bourbonnais@dauphine.fr) toutes erreurs, imprécisions ou anomalies.

Environnement Général de la Workfile d'Eviews



Chapitre 2 – Exercices 1, 2 et 5

La première étape consiste à créer un espace de travail (*WORKFILE*) par *<file> <new–workfile>*, c'est–à–dire un espace d'accueil pour les séries statistiques (mensuelles, semestrielles, annuelles, sans dates ...) et les résultats de calcul. Dans la fenêtre proposée, il convient, soit de choisir la périodicité des données (la date de début et de fin pour les séries temporelles, sous le forme par exemple : 1997:08 ⇒ Août 1997, 1997 si les données sont annuelles), soit d'indiquer le nombre d'observations (séries UNDATED – pour les séries en coupe instantanée). ***Dans un WORKFILE ne peut figurer que des séries de même périodicité***.

Créer des séries : *<Objects> <new–objects> series* puis donner un nom pour la série (fenêtre à droite). Ensuite sélectionner, dans la *workfile*, le ou les noms des séries à rentrer, par exemple REV, (éventuellement faire une sélection de plusieurs noms de séries non adjacentes en maintenant appuyer sur Ctrl) et double cliquer et cliquer sur *Open Group*.

Rentrer les données
Cliquer sur *edit +/–* (dans la *Spreadsheat*) et après la saisie, quitter par fermeture de la fenêtre et revenir à la *workfile.* Pour vérifier la saisie, dans la *workfile* sélectionner les noms des séries, par exemple REV et CONS, (éventuellement faire une sélection en maintenant appuyer sur Ctrl) et double cliquer

Sauvegarder et donner un nom à la Workfile
Menu général : <*File*> <*SaveAs*> Donner un nom au fichier, les données sont maintenant sauvegardées.

Pour modifier/éditer les donnés d'une ou plusieurs séries
Sélectionner la ou les séries puis double cliquer puis <*edit +/–*>.
Si la série est en représentation graphique ou sur d'autres fonctions (corrélation, statistiques, etc.) sélectionner <*View*> puis <*Spreadsheat*> puis <*edit +/–*>.

Faire un graphique
Dans la barre d'outil de la base de données cliquer sur <*View*> puis <*Graph*>.
Pour modifier les options du graphique, double cliquer sur la zone du graphique et faire *Graph option.*

Créer un nouvelle série à partir de séries existantes
Dans le menu sélectionner <*Quick*> <*Generate series*> puis taper dans la fenêtre :
CONS = 0.8*REV + 1000 ou bien rentrer dans la ligne de commande (en haut à droite) :
GENR CONS = 0.8*REV + 1000

Coefficient de corrélation
Pour calculer le coefficient de corrélation entre REV et CONS (évidemment égal à 1) sélectionner dans la workfile REV et CONS puis double cliquer et choisir <*View*> <*Correlation*>.

Créer un terme aléatoire
Par exemple, on veut créer un terme aléatoire qui suit une loi normale centrée et de variance 20 000. Dans <*Quick*> <*Generate series*> taper : *EPS = NRND * SQR(20000)* ou bien rentrer dans la ligne de commande : GENR *EPS = NRND * SQR(20000)*

Créer la série à expliquer
<*Quick*> <*Generate series*> on tape : CONSA = CONS + EPS

Estimation par MCO
<*Quick*> <*Estimate equation*>
Rentrer dans l'ordre : CONSA C REV
(Série à expliquer, Constante, Variable explicative). On obtient alors l'estimation du modèle linéaire simple ou bien rentrer dans la ligne de commande LS CONSA C REV

```
LS // Dependent Variable is CONSA
SMPL   1992  -  2001
Included observations: 10
============================================================
     Variable     Coefficient Std. ErrorT-Statistic  Prob.
============================================================
        C           1176.090   207.3921   5.670852   0.0005
        REV         0.780983   0.017939   43.53518   0.0000
============================================================
R-squared           0.995797    Mean dependent var 9985.575
Adjusted R-squared  0.995271    S.D. dependent var 2089.553
S.E. of regression  143.6878    Akaike info criter 10.11214
Sum squared resid   165169.4    Schwartz criterion 10.17266
Log likelihood     -62.75009    F-statistic        1895.312
Durbin-Watson stat  1.881665    Prob(F-statistic)  0.000000
============================================================
```

View	Procs	Objects	Name	Freeze	Estimate	Forecast	Stats	Resids
Variable		Coefficient	St.Error	t–stat	Probabilty			
C		.	.	.	.			
REV		.	...	...	...			

R^2	Mean Dependent Variable
R^2 adjusted	Std Deviation Dependent Variable
Std. Error	AIC (Akaike)
SSE	SBC (Schwartz)
Log Likelihood	F-stat
DW	Prob of F-stat

(1) t-stat:
Ho: coeff =0
H_1: coeff ≠ 0
- si Prob < 0.05
Rejet Ho
Coeff significatif
- si Prob > 0.05
accept Ho
coeff non signif

(2) F-stat:
Ho: coeff = 0
H_1: coeff ≠ 0
Si prob < 0.05
Rejet Ho, donc
modèle O.K

Les graphiques de y observé et estimé ainsi que les résidus sont obtenus : *<Resid>* dans la fenêtre de « Estimation Output » en haut à droite, ou bien par *<View> <Resid>*. Pour revenir aux résultats : *<View> <Estimation Output>*.
Graphe des résidus : *<View> <Actual_fitted_residual>*

ESTIMATION D'UN MODELE SOUS EVIEWS PAR LES MCO

- **Estimation d'un modèle sous Eviews**

QUICK\ESTIMATE EQUATION
Taper Y C X (nom de la variable expliquée, constante, noms des variables explicatives)
Ou dans la ligne de commande : LS Y C X1 X2 X3

Faire des prévisions

– Pour calculer la prévision, il faut d'abord modifier l'horizon et le nombre d'observations de l'échantillon, c'est–à–dire respectivement ***Modifier*** les ***Range*** et ***Sample*** dans le ***Workfile.***
<Procs> <Expand range> permet d'augmenter l'espace de travail en rajoutant des années supplémentaires ou bien double cliquer directement dans *<Range>*.
<Procs> <Sample> permet de modifier la taille de l'échantillon ou bien double cliquer dans *<Sample>*.
Aller dans la série REV et rajouter les valeurs prévues pour la variable explicative. Revenir dans l'equation.
<Forecast> permet de réaliser la prévision de consommation à partir du modèle estimé par les MCO.

Remarque : Il faut nommer la nouvelle série (par défaut elle porte le nom de la série à expliquer avec un « F » à la fin) ainsi que la série des écarts types des erreurs de prévision, par exemple ET.

Forecast	**Method**
Forecast Equation: Untitled Series Name: CONSAF Std. Error: ET	♦ Dynamic ♦ Static ♦ Structural (ARMA)
Sample Range for forecast	**Output**
1 16	♦ Do Graph ♦ Forecast Evaluation
OK	**No**

Nom de la nouvelle variable prévue et de l'écart type

Créer un intervalle de confiance
Générer les séries bornes inférieures et bornes supérieurs :
GENR IC1 = CONSAF + 2.306*ET
GENR IC2 = CONSAF – 2.306*ET
(2.306 = valeur du t de Student à (n – 2) degrés de liberté et alpha/2)

Remarque : Chaque type d'objets (équation, graphique, matrice, ...) peut être repéré dans la *workfile* par un nom, ce qui permet de les conserver.

Chapitre 3 – Exercice 1

On considère que les données sont sur tableur Excel et on commence par les importer d'Excel vers EVIEWS.

Importer des données Excel
Tout d'abord vérifier que les données sur Excel sont bien adjacentes c'est–à–dire les unes à coté des autres par exemple comme ceci :

Y	X1	X2	X3
12	2	45	121
14	1	43	132
10	3	43	154
16	6	47	145
14	7	42	129
19	8	41	156
21	8	32	132
19	5	33	147
21	5	41	128
16	8	38	163
19	4	32	161
21	9	31	172
25	12	35	174
21	7	29	180

Ensuite il convient de bien repérer trois paramètres :
– le nombre d'observations ou la date de début et la date de fin,
– le nombre de séries à lire,
– la cellule ou se situe la première observations (ici A2).
Le fichier Excel ayant été **impérativement fermé**, nous pouvons ouvrir EVIEWS.

Tout d'abord créer l'espace de travail (ici undated et 14 observations) puis,
<File> <Import> <Read text–Lotus–Excel>
Important : Ne pas oublier de renseigner la fenêtre *List files of type* en sélectionnant l'option *Excel*.
Upper left data cell renvoie à la première cellule à partir de laquelle EVIEWS effectue l'importation des données (ici A2) puis le nombre de variables à lire (ici 4).

Une autre méthode, plus simple, consiste à procéder par <copier> <coller>.

Utilisation de EVIEWS pour le calcul matriciel

On veut, par exemple, calculer les coefficients de régression linéaire, c'est–à–dire la matrice :
$\hat{a} = (X'X)^{-1} X'Y$

Création d'un vecteur unité (pour le terme constant)
<Quick> <Generate series> et taper ONE =1 ou, sur la ligne de commande (sous <File>) :
Genr ONE = 1

Créer la matrice X'X
On crée un groupe de séries en sélectionnant, dans l'ordre les variables explicatives exemple : one x1 x2 x3. x4, puis on double clique et on nomme le groupe (par exemple GX).

Revenir dans le workfile et taper sur la ligne de commande :
Matrix xpx = @transpose(@convert(gx))*@convert(gx)
Matrix xpy = @transpose(@convert(gx))*@convert(y)
Matrix a = @inverse(xpx)*xpy

Remarque : @ s'obtient par Alt Gr

On vérifie que l'on trouve le même résultat en tapant dans la zone de commande :
LS Y C X1 X2 X3

Dependent Variable: Y
Method: Least Squares
Sample: 1 14
Included observations: 14

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	32.89132	11.66331	2.820068	0.0182
X1	0.801901	0.298436	2.687012	0.0228
X2	-0.381362	0.156581	-2.435564	0.0351
X3	-0.037132	0.052023	-0.713768	0.4917

R-squared	0.702687	Mean dependent var	17.71429
Adjusted R-squared	0.613493	S.D. dependent var	4.177385
S.E. of regression	2.597069	Akaike info criterion	4.981600
Sum squared resid	67.44767	Schwarz criterion	5.164188
Log likelihood	-30.87120	F-statistic	7.878181
Durbin-Watson stat	3.186886	Prob(F-statistic)	0.005452

View	Procs	Objects	Name	Freeze	Estimate	Forecast	Stats	Resids
Variable		Coefficient	St.Error	t–stat	Probabilty			
C		.	.	.	.			
X1		.	.	.	.			
X2		.	.	.	.			
X3		.	.	.	.			

(1) t-stat:
Ho: coeff =0
H_1: coeff ≠ 0
- si prob < 0.05
Rejet Ho
Coeff significatif
- si prob > 0,05
accept Ho
coeff non signif

R^2	Mean Dependent Variable
R^2 adjusted	Std Deviation Dependent Variable
Std. Error	AIC (Akaike)
SSE	SBC (Schwartz)
Log Likelihood	F-stat
DW	Prob of F-stat

(2) F-stat:
Ho: les coeffs = 0
H_1: 1 coeff ≠ 0
Si prob < 0.05
Rejet Ho , donc
modèle O.K

- **Analyse des résultats**

- colonne *Coefficient* : les paramètres estimés $\hat{a}_i$
- colonne *Std. Error* : les écart-types estimés des paramètres estimés $\hat{\sigma}_{\hat{a}_i}$
- colonne *t-Statistics* : les t-statistiques $t_{\hat{a}_i,obs} = \hat{a}_i / \hat{\sigma}_{\hat{a}_i}$
- colonne Prob : les p-values du test de significativité des coefficients $P(|t_{\hat{a}_i}| > t_{obs} | H_0)$ (si elles sont supérieures à α, accepter $H_0 : a_i = 0$)
- R-squared : coefficient de détermination R^2
- Adjusted R-squared : $\overline{R}^2$
- S.E. of regression (Standard Error of regression) : $\hat{\sigma}_\varepsilon$
- Sum squared resid : $SCR = e'e = \sum_i e_i^2$
- Log-likelihood : log-vraisemblance pour les paramètres estimés

- Durbin Watson : statistique de Durbin Watson du test d'autocorrélation des résidus
- Mean dependent variable : moyenne empirique de la variable endogène $\bar{y} = \sum y_i / n$
- S.D. dependent variable (standard deviation of the dependent variable) : $\sqrt{\frac{\sum (y_i - \bar{y})^2}{n-1}} = \sqrt{\frac{SCT}{n-1}}$
- Akaike Info Criterion : critère AIC (utilisé notamment en séries temporelles pour le choix des retards d'un modèle)
- Schwarz Criterion : critère de Schwarz (même utilisation que le critère AIC)
- F-statistic : statistique de Fisher du test de significativité globale de la régression ($H_0 : a_1 = \ldots = a_k = 0$), Prob (F-statistic) : p-value (si supérieure à α, accepter H_0)

EVIEWS fait la correspondance suivante :
C(1)....a0
C(2)....a1
C(3)....a2
C(4)....a3

Il s'agit de l'ordre dans lequel les variables ont été citées lors de l'instruction de régression. Vérifier par<*Views*> <*Representation*> et en double cliquant dans la *workfile* sur le vecteur C

Remarque : chaque type d'objets – série, équation, matrice, scalaire, ... – est repéré dans la *workfile* par un symbole différent.

Chapitre 3 – Exercice 2 : Matrice des variances/covariances
Elle est donnée par *: <View> <Covariance matrix>.*
La matrice des variances–covariances des coefficients permet d'effectuer, par le calcul, le test de la question 3. Ou bien le test peut être effectué directement par : <*Views CoefficientsTests*> <*Wald Coefficient Restriction*>

Chapitre 3 – Exercice 3 : Calcul de la statistique de Fisher quand on veut tester l'amélioration du modèle par ajout d'une ou plusieurs variables

– Test par les outils de EVIEWS
Faire une régression de y sur la ou les variables figurant initialement dans le modèle puis cliquer sur : <*View*> <*Coefficients tests*> <*omitted variables*>.
Taper en séparant avec un espace les variables à rajouter. On obtient alors:

Omitted Variables: X2 X3

F–statistic 3.095036 Probability **0.089899 > 0.05** ⇒ on accepte H0

(L'ajout de x2 et x3 n'améliore pas de manière significative le pouvoir explicatif du modèle, car la probabilité critique est supérieure à 0.05).

– Création d'un programme batch (programme de traitement répétitif)

Un programme batch est une suite d'instructions (comme dans un autre langage) qui permet d'effectuer des traitements répétitifs : boucles, tests, etc. Le programme est sous forme d'un fichier ASCII c'est–à–dire complètement lisible par n'importe quel éditeur de texte.

<File> <New–Programme>

Programmation du calcul du Fisher empirique du test précédent
H0: SCE – SCE1 = 0
H1: SCE – SCE1 ≠ 0

Programme :
```
ls y c x1
scalar sce1 = @ssr/(1–@r2)*@r2
scalar n1 = @ncoef
‘ On donne le nom eq1 à l'équation de régression à trois variables explicatives
equation eq1.ls y c x1 x2 x3
scalar sce = @ssr/(1–@r2)*@r2
scalar n2 = @ncoef
scalar fe = ((sce–sce1)/(n2–n1))/(@ssr/(@regobs–@ncoef))
```

fe = Fisher empirique
Remarques : une ligne commençant par le symbole ‘ est une ligne de commentaire. Pour faire apparaître la valeur d'un scalaire il faut « double cliquer » sur le nom de l'objet et sa valeur est indiquée en bas à droite de la fenêtre.

Chapitre 3 – Exercice 3 : Test de stabilité de Chow
Soit le test suivant :
H_0: SCR – (SCR1+SCR2) = 0
H_1: SCR – (SCR1+SCR2) ≠ 0

Programme EVIEWS :

```
‘ Si la variable test = 0 en fin de programme alors acceptation de H1: au seuil de 5%
‘Régression sur la totalité de la période
‘ On rappelle les valeurs calculées au programme précédent
scalar test = 0
scalar scr = eq1.@ssr
scalar n = eq1.@regobs – eq1.@ncoef
‘
smpl 1 7 ‘Régression sur la première sous–période
ls y c x1 x2 x3
scalar scr1 = @ssr
scalar n1 = @regobs – @ncoef
'
smpl 8 14 'Régression sur la deuxième sous–période
ls y c x1 x2 x3
scalar scr2 = @ssr
scalar n2 = @regobs – @ncoef
'
'Calcul du Fisher empirique
scalar ddln = n–(n1+n2)
scalar ddld = n1+n2
scalar fe = ((scr – (scr1+scr2))/ddln)/((scr1+scr2)/ddld)
smpl 1 14
if @fdist(fe,ddln,ddld) < 0.05 then test = 1
endif
```

Directement par utilisation des commandes EVIEWS :
<View> <Stability–test> <Chow–Breakpoint–test>
(Vérifier que l’on trouve bien le même résultat, le point de rupture est en période 8)

Chapitre 3 – Exercice 3 : Autre test
On veut tester si a1 = 1 et a2 = a3 par:
H0: scr1 – scr = 0
H1: scr1 – scr # 0

Programme :

```
' Test de contrainte linéaire
SCALAR SCR = EQ1.@SSR
SCALAR N = EQ1.@REGOBS – EQ1.@NCOEF
GENR Z = Y – X1
GENR V = X2 + X3
LS Z C V
SCALAR SCR4 = @SSR
SCALAR N4 = @REGOBS – @NCOEF
SCALAR FE = (((SCR4 – SCR)/(N4 – N))/(SCR/N))
```

Par utilisation des commandes :Estimer le modèle complet, puis *<Views CoefficientsTests>* *<Wald Coefficient Restriction>* et taper dans la fenêtre, C(2) = 1, C(3) = C(4)

Chapitre 3 – Exercice 6 : Variables indicatrices
Il faut générer autant de variables indicatrices que de périodes dans l’année (soit 4 puisque les données sont trimestrielles) :

```
GENR D1 = @SEAS(1)
GENR D2 = @SEAS(2)
GENR D3 = @SEAS(3)
GENR D4 = @SEAS(4)
```

Ce qui donne, si l’on considère des séries trimestrielles, sur deux ans :

$$d1=\begin{pmatrix}1\\0\\0\\0\\1\\0\\0\\0\end{pmatrix} d2=\begin{pmatrix}0\\1\\0\\0\\0\\1\\0\\0\end{pmatrix} d3=\begin{pmatrix}0\\0\\1\\0\\0\\0\\1\\0\end{pmatrix} d4=\begin{pmatrix}0\\0\\0\\1\\0\\0\\0\\1\end{pmatrix}$$

Remarque : Si on essaie d’estimer le modèle avec les 4 variables indicatrices, le message « Near singular matrix » apparaît. Cela signifie que la matrice est singulière. En effet, il y a colinéarité entre la constante et le groupe des 4 variables indicatrices (d1 + d2 + d3 + d4 = 1). Il convient alors d’en supprimer une ou bien d’effectuer une régression sans terme constant.

Chapitre 3 – Exercice 6 : Coefficient d’élasticité
Mettre le modèle sous forme Log Log en générant les séries :
GENR LV = LOG(VENTES)
GENR LPUB = LOG(PUB)
PUIS:
LS LV LPU D1 D2 D3 C
NB : le coefficient estimé de la variable LPUB donne alors directement le coefficient d’élasticité.
Nous pouvons aussi directement taper l’instruction :
LS LOG(VENTES) LOG(PUB) D1 D2 D3 C

Chapitre 4 – Exercice 1 : Calcul d’un coefficient de corrélation partielle
Exemple de calcul de r²yx3.x1x2
LS Y C X1 X2
GENR E1 = RESID
LS X3 C X1 X2
GENR E2 = RESID
scalar rau = @cor(E1,E2)

Chapitre 4 – Exercice 4 : Test de Klein

Programme

```
scalar test1 = 0     'test est un indicateur de colinéarité égal à 1 en cas de risque
equation eqt.ls y c x1 x2 x3 x4
scalar cr = @r2
for !i = 1 to 4
   for !j = 1 to 4
      if !i <> !j then scalar cp!i!j = @cor(x!i, x!j)^2
         if cp!i!j > cr then test1 = 1
         endif
      endif
   next
next
```

L'indicateur test = 0, il n'y a donc pas de risque grave de colinéarité.

Chapitre 4 – Exercice 4 : Test de Farrar et Glauber

Programme

```
ls y c x1 x2 x3 x4
matrix(4, 4) mar  'création de la matrice mar
for !i = 1 to 4
   for !j = 1 to 4
      mar(!i, !j) = @cor(x!i, x!j)
   next
next
scalar dt = @det(mar)
scalar chie = –(@regobs–1–(2*@ncoef+5)/6)*log(dt)  '    → χ2* empirique
scalar ndf = 0.5*@ncoef*(@ncoef–1)          'ddl
if @chisq(chie,ndf) < 0.05 then scalar test2 = 0 '' ''''‘Prob critique comparée au seuil de 0.05
else scalar test2 = 1
endif
```

Si le test = 1, on rejette H0, il y a présomption de multicolinéarité.

Chapitre 4 – Exercice 5 : Sélection de variables explicatives

Quatre méthodes :

– 1) ***Toutes les régressions possibles*** (2^k – 1)

```
' Le numéro de la meilleure équation se trouve dans la variable NEQ
'
 !a = 1
FOR !I =1 TO 4
equation  eq!a.ls Y C X!i
!a = !a + 1
next

FOR !I = 1 TO 3
FOR !J = !I+1 TO 4
equation eq!a.ls Y C X!I X!J  'equation à deux variables
!a = !a + 1
next
next

FOR !I = 1 TO 2
```

```
FOR !J = !I+1 TO 3
FOR !K = !J+1 TO 4
equation  eq!a.ls Y C X!I X!J  X!K  'equation à trois variables
!a = !a + 1
next
next
next

equation  eq!a.ls Y C X1 X2 X3 X4

' Sélection the BEST
Scalar  BEST = 0
FOR !I = 1 TO 15
scalar IND = 0
scalar NV = eq!I.@ncoef
for !J = 2 TO NV
scalar te =@abs( eq!I.C(!J)/sqr(eq!I.@covariance(!J,!J)))
scalar ddl = eq!I.@regobs– eq!I.@ncoef
IF @tdist(te,ddl)> 0.05 then  ind = 1
endif
NEXT !J
IF  IND = 0 then
        IF eq!I.@R2 > BEST then scalar neq= !I
        BEST =  eq!I.@R2

ENDIF
ENDIF
NEXT
```

<u>Critères de sélection de la meilleure régression :</u>
– Tous les coefficients sont significatifs (Prob. critique < = 0.05)
– R2 max

– 2) ***Backward elimination*** : régression sur le modèle complet puis élimination une par une des variables explicatives non significatives.

Variable	Coefficient	Std. Error	T–Statistic	Prob.
C	134.2903	25.36038	5.295282	0.0018
X1	0.206231	0.156079	1.321325	**0.2345** > **0.05** ⇒ **NS**
X2	0.330841	0.087753	3.770140	0.0093
X3	–0.557321	0.096206	–5.793019	0.0012

Modèle final : y c x2 x3 (tous les coefficients sont significatifs)

– 3) ***Forward selection***

Première étape : Coefficient de corrélation simple le plus élevé :

```
scalar cmax = 0
for !i = 1 to 3
 scalar c!i = @cor(y, x!i)^2
  if c!i > cmax then
```

```
    scalar vm = !i
    cmax = c!i
  endif
next
```

vm = 3 $\Rightarrow$ on retient x3

‘Recherche du coefficient. de corrélation partielle le plus élevé, ayant retiré l’influence de x3 :

```
' calcul der²y x1.x3
ls y c x3
genr e3 = resid
ls x1 c x3
genr e4 = resid
scalar cp1 = @cor(e3, e4)^2
scalar t1 = sqr(cp1/((1–cp1)/(@regobs–2)))
'
' calcul de r²yx2.x3
ls y c x3
genr e5 = resid
ls x2 c x3
genr e6 = resid
scalar cp2 = @cor(e5,e6)^2
scalar t2 = sqr(cp2/((1–cp2)/(@regobs–2)))
```

t2 > t1 $\Rightarrow$ on retient x2

```
'calcul de r²yx1.x2x3
ls y c x2 x3
genr e7 = resid
ls x1 c x2 x3
genr e8 = resid
scalar cp4 = @cor(e7,e8)^2
scalar t4 = sqr(cp4/((1–cp4)/(@regobs–2)))
```

t4 = 1.525 $\Rightarrow$ NS, la procédure est arrêtée.

– 4) ***Stagewise regression***

Première étape : Coefficient de corrélation simple le plus élevé (sélection x3), puis

```
ls y c x3
genr e9 = resid
for !i=1 to 3
scalar cs!i = @cor(x!i,e9)^2
scalar te!i = sqr(cs!i/((1–cs!i)/(@regobs–2)))
next
```

cs2 > cs1 > sc3, et cs2 significativement $\neq 0$ (te2 = 2.49 > 2.447)
donc on retient x2.

```
ls y c x2 x3
genr e9 = resid
for !i = 1 to 3
scalar cs!i = @cor(x!i,e9)^2
```

```
scalar te!i = sqr(cs!i/((1–cs!i)/(@regobs–2)))
next
```

Plus aucun coefficient de corrélation n’est significatif, la procédure est arrêtée.

Chapitre 5 – Exercice 2 : Procédures d’estimation en cas d’autocorrélation des erreurs

```
'Estimation directe de rau partir de la statistique de DW : première méthode'
ls y c x1 x2 x3
scalar rau1=1–@dw/2
genr dy = y–rau1*y(–1)    'On génére les quasi–diff.'
genr dx1 = x1–rau1*x1(–1)
genr dx2 = x2–rau1*x2(–1)
genr dx3 = x3–rau1*x3(–1)
equation eqm1.ls dy c dx1 dx2 dx3 'Régression nommée eqm1'
scalar am1 = c(1)/(1–rau1)            'Détermination du terme constant du modèle initial'

'estimation directe de rau partir d’une régression sur les résidus : deuxième méthode'
ls y c x1 x2 x3
genr res = resid
ls res c res(–1)
scalar rau2 = C(2)
genr dy = y–rau2*y(–1)    'On génère les quasi–diff.'
genr dx1 = x1–rau2*x1(–1)
genr dx2 = x2–rau2*x2(–1)
genr dx3 = x3–rau2*x3(–1)
equation eqm2. ls dy c dx1 dx2 dx3
scalar am2 = c(1)/(1–rau2)

'Méthode de Cochran Orchut'
ls y c x1 x2 x3
genr res = resid
for !i = 1 to 10     ‘On procède arbitrairement à 10 itérations afin d’obtenir la stabilité des coefficients
ls res c res(–1)
scalar rau3=c(2)
genr dy = y–rau3*y(–1)
genr dx1 = x1–rau3*x1(–1)
genr dx2 = x2–rau3*x2(–1)
genr dx3 = x3–rau3*x3(–1)
equation eqm3.ls dy c dx1 dx2 dx3
scalar am3 = c(1)*(1–rau3)
genr res = (y–(am3+c(2)*x1+c(3)*x2+C(4)*x3))
next

'Méthode dite de balayage'
scalar scrf = 999999         'somme des carrés des résidus
for !i = 1 to 99
scalar rau = !i/100
genr dy = y–rau*y(–1)
genr dx1 = x1–rau*x1(–1)
genr dx2 = x2–rau*x2(–1)
genr dx3 = x3–rau*x3(–1)
ls dy c dx1 dx2 dx3
if  scrf > @ssr then
scalar rau4 = rau
scalar scrf = @ssr
endif
next
genr dy = y–rau4*y(–1)
```

```
genr dx1 = x1–rau4*x1(–1)
genr dx2 = x2–rau4*x2(–1)
genr dx3 = x3–rau4*x3(–1)
equation eqm4.ls y c dx1 dx2 dx3
scalar am4 = c(1)/(1–rau4)
```

Chapitre 5 – Exercice 4 : Tests d'hétéroscédasticité

– Test de Goldfeld–Quant:

⇒ Une variable doit être la cause de l'hétéroscédasticité.
⇒ Nombre d'observations doit être important
⇒ Omettre C observations centrales (C = partie entière de N/4).
Ici n = 30, alors C = 7 ou 8.

Programme

```
'calcul du nombre d'observations à retirer (nm)'
smpl 1 30
scalar !nobs = @obs(y)
scalar nm = @ceiling(!nobs/4)
scalar !n1 = @ceiling((!nobs–nm)/2)
smpl 1 !n1
ls y c x
scalar scr1 = @ssr
scalar ddl1 = @regobs–@ncoef
scalar !n2 = !n1 + nm + 1
smpl !n2 !nobs
ls y c x
scalar scr2 = @ssr
scalar ddl2 = @regobs–@ncoef
if scr1>scr2 then scalar fe = (scr1/ddl1)/(scr2/ddl2)
             else   scalar  fe = (scr2/ddl2)/(scr1/ddl1)
endif
scalar test = 0
if @fdist(fe, ddl1, ddl2) < 0.05 then  test = 1
endif
```

Conclusion: si test = 1, alors on rejette H0, hypothèse selon laquelle le modèle est homoscédastique.

•*Test de Gleisjer:*

Programme

```
smpl 1 30
equation eq.ls y c x  'Régression sur le modèle de base.
genr resa = abs(resid)
equation eq1.ls resa c x  ' Régression de la valeur absolue des résidus sur la variable
genr xra = sqr(x)   'explicative supposée être la cause de l'hétéroscédasticité.
equation eq2.ls resa c xra  ' Idem mais avec la racine carrée de la variable explicative.
genr xin = 1 / x
equation eq3.ls resa c xin ' Idem mais avec l'inverse de la variable explicative
scalar proba = 1
for !I = 1 to 3
scalar te =@abs( eq!I.C(2)/sqr(eq!I.@covariance(2,2)))
scalar ddl = eq!I.@regobs– eq!I.@ncoef
' On retient la probabilité critique la plus faible et le numéro de l'équation significative
IF @tdist(te, ddl) < proba then
```

```
proba = @tdist(te, ddl)
scalar ind = !I
endif
next
' correction de l'héteroscedasticité
genr pon = 1 / sqr(x)  'ne pas oublier que la correction affecte aussi la constante.
genr yp=y*pon
genr xp=x*pon
equation pond1.ls yp xp pon
' Correction de l’hétéroscédasticité par une commande EVIEWS : Régression pondérée
genr pon = 1/sqr(x)
equation pond.ls(w = pon) y c x
' On détruit de la workfile les objets dont on n’a plus besoin
delete eq xra xin resa ddl1 ddl2 ddl scr1 scr2 nm
```

Conclusion : L’hypothèse d’homoscédasticité est rejetée si les coefficients a1 des spécifications sont non significatifs. Dans le cas contraire (hétéroscédasticité), on retient la forme dont le t de Student est le plus élevé, ici la forme 2. Vérifier que les équations de régression pond1 et pond donnent les mêmes résultats.

Remarque : En mode interactif la correction de l’hétéroscédasticité est effectuée par une commande EVIEWS : <*Quick*> <*Estimate Equation*> puis faire *Option* –, cliquer *Weighted LS/TSLS* et taper dans *Weight* le nom de la variable de pondération (ici PON).

Chapitre 6 – Exercice 2 : Modèles non linéaires

• *Le modèle logistique* $y_t = \dfrac{y_{max}}{1 + br^t}$

```
'
' Question 3
'
tend  = @trend(1978)
scalar somcr=9999999
for !i=680 to 900 step 10
  scalar ymax=!i
  genr Y=log(ymax/taux–1)
  equation EQ.ls Y c tend
  if @ssr<somcr then scalar somcr=@ssr
                              scalar ymaxmax=ymax
 endif
next
genr Y=log(ymaxmax/taux–1)
equation balay.ls Y c tend
scalar b3=exp(C(1))
scalar r3=exp(C(2))
```

Chapitre 6 – Exercice 3 : Modèle de Gompertz

'

Modèle de Gompertz : $y_t = e^{a+br^t}$

‘on affecte des valeurs initiales aux coefficients C(1) = 5, C(2) = –3, C(3) = 0.5

```
param 1 5. 2 –3. 3 0.5
```

ou en mode interactif : il faut double cliquer dans le vecteur C et rentrer les valeurs.
Puis <*Quick*> <*Estimate equation*> et taper l’équation : Taux = c(1) / (1+c(2)*c(3)^tend)

Puis :
ls taux = exp(c(1)+c(2)*c(3)^tend)

Chapitre 6 – Exercice 4 : Fonction de production de type CES

La fonction CES:

‘on affecte des valeurs initiales aux coefficients C(1) = 12 ; C(2) = –1 ; C(3) = 0.5 ; C(4) = –0.5

```
param 1 12. 2 –1. 3 0.5 4 –0.5
LS q = c(1)*(c(3)*k^c(4)+(1–c(3))*l^c(4))^c(2)
```

Chapitre 7 – Exercice 1 : Modèle autorégressif

Programme

```
'test d’autocorrélation dans modèle autorégressif'
equation eq1.ls po c po(–1) pe
scalar rau = (1–@dw/2)
scalar h = rau*sqr(@regobs/(1–@regobs*sqr(@covariance(2,2)))
if h > 1.96 then    'test bilatéral
   scalar test = 1
   else test = 0
endif
```

$\boxed{\hat{\sigma}_{\hat{a}_1}}$

Si le test est égal à 1, on accepte H1 (rau ≠ 0).

Procédons tout de même à la correction de l’autocorrélation des erreurs :

```
genr dpo = po–po(–1)
genr dpe = pe–pe(–1)                    '1ère itération
equation eq2.ls dpo c dpo(–1) dpe
equation eq3.ls po c po(–1) po(–2) pe pe(–1)            'estimation de rau
scalar rau = –c(5)/c(4)
genr dpo = po–rau*po(–1)
genr dpe = pe–rau*pe(–1)                '2ème itération
equation eq4.ls dpo c dpo(–1) dpe
```

Après stabilité des estimations des coefficients, on a l’estimation suivante :

Variable	Coefficient	Std. Error	T–Statistic	Prob.
C	–1.670427	0.734294	–2.274875	0.0439
DPO(–1)	0.233995	0.014200	16.47898	0.0000
DPE	0.613710	0.009885	62.08361	0.0000

rau = 0.4105

D’où â0 = –1.67/(1–0.41) ; â1 = 0.23 ; â2 = 0.63 ; DPO = â0 + â1 DPO(–1) + â2 DPE

NB : Si la convergence n’avait pas été rapide, on aurait pu utiliser le programme :

```
genr dpo = po–po(–1)
genr dpe = pe–pe(–1)                    ‘Initialisation
equation n0.ls dpo c dpo(–1) dpe
for !i=1 to 5
    equation int.ls po c po(–1) po(–2) pe pe(–1)
```

```
    scalar ro = –c(5)/c(4)
     genr dpo = po–ro*po(–1)
    genr dpe = pe–ro*pe(–1)
   equation nf.ls dpo c dpo(–1) dpe
next
```

Chapitre 7 – Exercice 4 : Modèle de Koyck

```
equation eq.ls y c y(–1) x
scalar b = c(1)/(1–c(2))
```

$b = a_0/(1–\lambda)$
$\lambda = \hat{a}_1$
$a = \hat{a}_2$

Chapitre 8 – Exercice 3 : Estimation par les Doubles Moindres Carrés

Programme

```
'
' Estimation du modèle à équations simultanées
'
genr tend = @trend(1919)
genr sw = w + wp
' Double Moindres Carrés par programmation
equation eq1.ls p c tend wp t g p(–1) k x(–1)
forcst pa
equation eq2.ls i c pa p(–1) k
'
' D M C par utilisation de la fonction EVIEWS
'
equation eq3.tsls i c p(–1) k p @ tend wp t g p(–1) k x(–1)
```

Chapitre 9 – Exercice 1 : Exemple d'application des tests DF et DFA au CAC40

Le programme permettant d'effectuer les trois régressions est le suivant :

```
'TESTS DE DF:'
genr tend=@trend(1)
genr dcac=cac–cac(–1)
equation eq3.ls dcac c cac(–1) tend
equation eq2.ls dcac c cac(–1)
equation eq1.ls dcac cac(–1)
```

En mode interactif les instructions sont – après avoir séléctionné la série – dans le menu *View* <*Unit Root Test*>. Dans la fenètre affichée, Eviews propose le choix entre les trois modèles [1], [2], et [3] et le nombre de décalages dans le cas d'un test DFA (si 0, alors on effectue le test DF). L'interprétation 4u test est la suivante :

```
       Augmented Dickey–Fuller Unit Root Test on CAC
============================================================
ADF Test Statistic –1.805839     1%   Critical Value*–3.4388
                                 5%   Critical Value –2.8645
                                 10% Critical Value  –2.5683
============================================================
*MacKinnon critical values for rejection of hypothesis of a unit root.


Augmented Dickey–Fuller Test Equation
Dependent Variable: D(CAC)
Method: Least Squares
Date: 08/22/98   Time: 10:58
Sample(adjusted): 2 1160
Included observations: 1159 after adjusting endpoints
============================================================
     Variable      CoefficientStd. Errort–Statistic  Prob.
============================================================
      CAC(–1)       –0.007093   0.003928  –1.805839   0.0712
         C           13.63563   7.344864   1.856485   0.0636
============================================================
R–squared            0.002811    Mean dependent var 0.417204
Adjusted R–squared   0.001949    S.D. dependent var 20.64858
S.E. of regression   20.62845    Akaike info criteri8.892943
Sum squared resid    492341.5    Schwarz criterion  8.901667
Log likelihood      –5151.461    F–statistic        3.261054
Durbin–Watson stat   1.867860    Prob(F–statistic)  0.071203
============================================================
```

Puisque la valeur empirique (ADF Test Statistic –1.8058) est supérieure aux trois valeurs critiques à 10, 5 et 1%, alors on accepte l’hypothèse H0 d’une série non stationnaire.

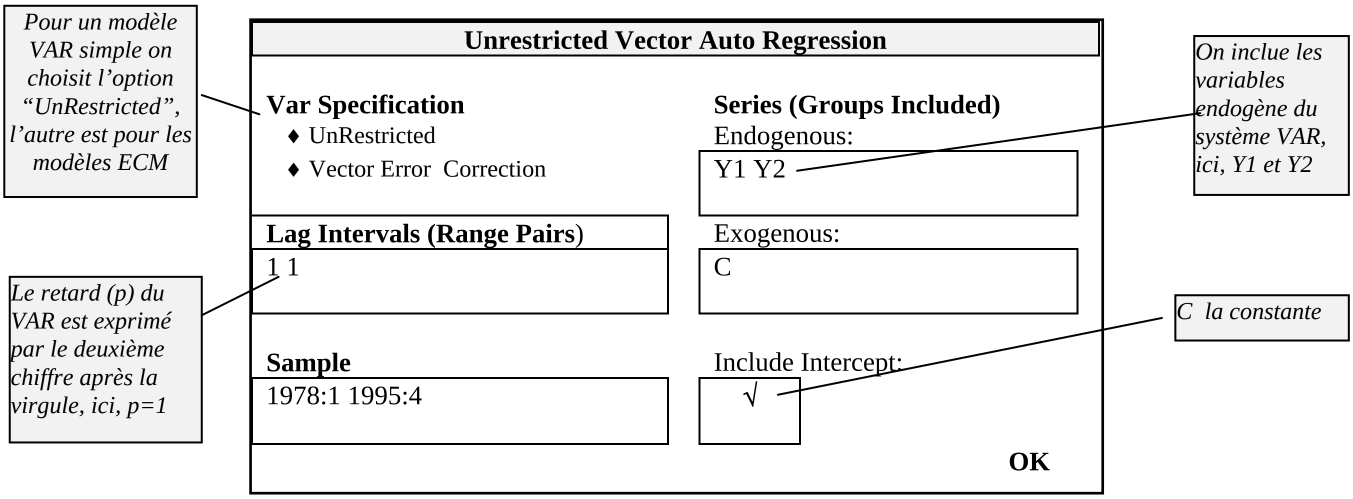
Chapitre 9 – Exercice 3 : Analyse des immatriculations en France

Le programme permettant de calculer la prévision à l’horizon de 12 mois est le suivant :

```
' Prévision du nombre d'immatriculations en France
SMPL 86:01 95:12
' Première étape Désaisonnalisation de la série
' série CVS = IMCVS et les coefficients saisonniers = CS
Immat.seas(M) IMCVS CS
' Stationnarisation de la série par différences premières
Genr DIMCVS = IMCVS – IMCVS(–1)
' Estimation du modèle ARMA
Equation eq1.LS DIMCVS MA(1)
' Prévision à l'horizon de 12 mois
SMPL 95:01 95:01
GENR IMCVSF = IMCVS(–1) + DIMCVSF
GENR IMMATF = IMCVSF*CS
FOR !I = 2 TO 12
SMPL 95:!I 95:!I
GENR IMCVSF = IMCVSF(–1) + DIMCVSF
GENR IMMATF = IMCVSF*CS
NEXT
```

Chapitre 10 – Exercice 2 : Spécification, estimation et prévision d'un modèle VAR

Sélectionner les deux séries y_1 et y_2, puis choisir <*Open Var*> :



Les résultats complets (estimations des coefficients, écarts types, critères AIC, SC, etc.) sont fournis. L'option <*Views*> <*Representation*> permet d'obtenir la représentation du modèle VAR. :

$y_{1,t} = 0{,}00676 * y_{1,t-1} - 0{,}6125 * y_{2,t-1} + 17{,}129$

$y_{2,t} = -0{,}1752 * y_{1,t-1} + 0{,}2992 * y_{2,t-1} - 12{,}862$

Le programme permettant de calculer la prévision assortie de son écart type est les suivant :

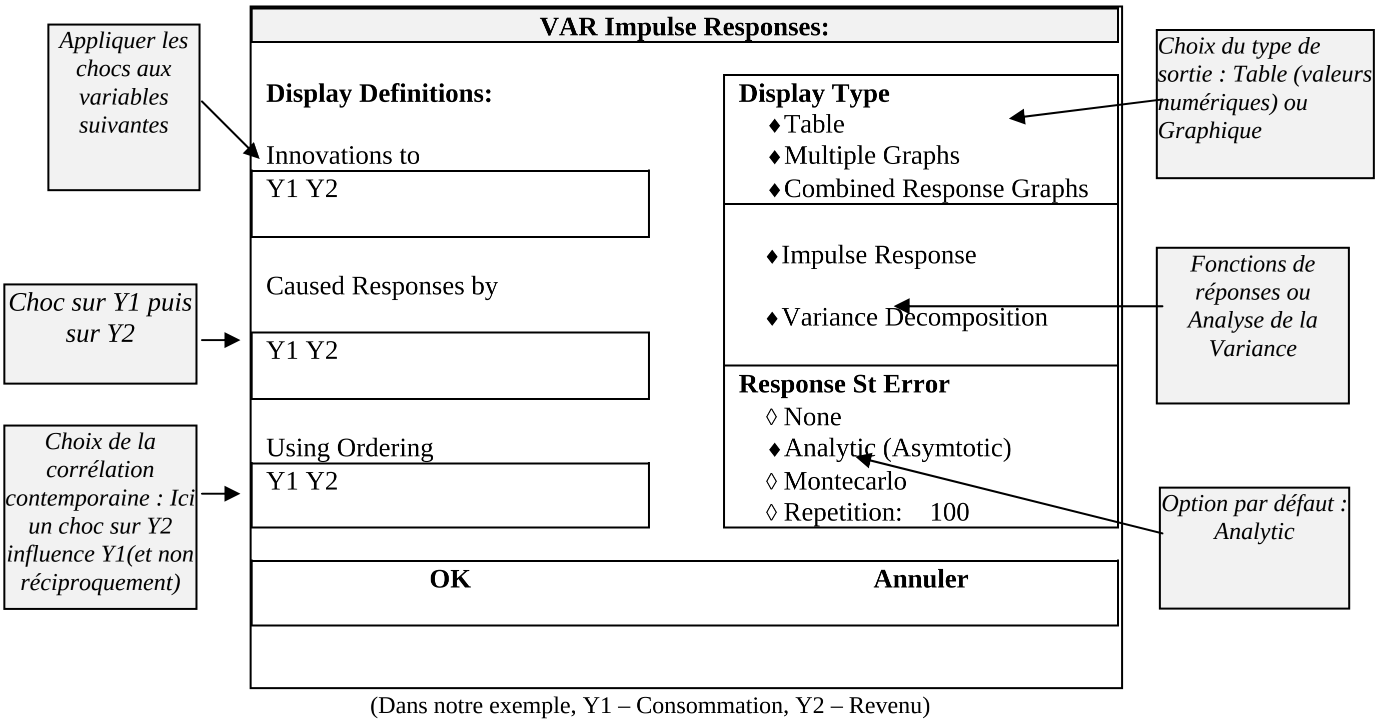
```
Expand 78:1 96:4
Var Var1.Ls 1 1 Y1 Y2
var1.makemodel(model1)  @ALL F
'Calcul de la prévision
'
smpl 96:1 96:4
Model1.SOLVE

smpl 78:1 95:4
'Calcul des écart–types de l'erreur de prévision pour les
' périodes 1, 2, 3 et 4
VAR1.MAKERESID
MATRIX(2, 2) VC
MATRIX(2, 2) A
MATRIX SIG1
MATRIX SIG2
MATRIX SIG3
MATRIX SIG4
A(1,1) = Var1.C(1,1)
A(1,2) = Var1.C(1,2)
A(2,1) = Var1.C(2,1)
A(2,2) = Var1.C(2,2)
FOR !I = 1 TO 2
    FOR !J = 1 TO 2
```

```
      VC(!I, !J) = @cov(RESID0!I, RESID0!J)
     NEXT !J
NEXT !I
 SIG1 = VC
SIG2 = VC + A * VC * @TRANSPOSE(A)
SIG3 = VC + A * VC * @TRANSPOSE(A) + A*A*VC*@TRANSPOSE(A*A)
SIG4 = VC + A * VC * @TRANSPOSE(A)+  A*A*VC*@TRANSPOSE(A*A) +
A*A*A*VC*@TRANSPOSE(A*A*A)
DELETE RESID01 RESID02 A VC model1
'
' Calcul de l'intervalle de prévision
'
for !I=1 TO 4
smpl 96:1 96:!I
GENR ICPY1 = Y1F + 1.96*sqr(SIG!I(1,1))
GENR ICPY2 = Y2F + 1.96*sqr(SIG!I(2,2))
GENR ICMY1 = Y1F – 1.96*sqr(SIG!I(1,1))
GENR ICMY2 = Y2F – 1.96*sqr(SIG!I(2,2))
next
smpl 78:1 96:4
```

Chapitre 10 – Exercice 3 : Analyse d’une fonction de réponse impulsionnelle et décomposition de la variance

Dans la fenêtre des résultats d’estimation du modèle VAR, sélectionner <*Impulse*> :



(Dans notre exemple, Y1 – Consommation, Y2 – Revenu)

Chapitre 10 – Exercice 4 : Tests de causalité

Programme :

```
'
' Test de causalité de Granger
'
```

```
equation EQU1.LS Y1 C Y1(–1) Y2(–1)
scalar SCRU1 = @SSR
scalar NDLU1 = @REGOBS–@NCOEF
equation EQR1.LS Y1 C Y1(–1)
scalar SCRR1 = @SSR
scalar NDLR1 = @REGOBS–@NCOEF
SCALAR F1N = (SCRR1 – SCRU1) /(NDLR1 – NDLU1)
SCALAR F1D =SCRU1/NDLU1
SCALAR F1 = F1N / F1D
equation EQU2.LS Y2 C Y1(–1) Y2(–1)
scalar SCRU2 = @SSR
scalar NDLU2 = @REGOBS–@NCOEF
equation EQR2.LS Y2 C Y2(–1)
scalar SCRR2 = @SSR
scalar NDLR2 = @REGOBS–@NCOEF
SCALAR F2N = (SCRR2 – SCRU2) /(NDLR2 – NDLU2)
SCALAR F2D =SCRU2/NDLU2
SCALAR F2 = F2N/F2D
delete  SCRU1 SCRU2 SCRR1 SCRR2 F1N F2N F1D F2D NDLU1 NDLR1 NDLU2 NDLR2

'
' Test de causalité de GRANGER PAR CHI2
'
'
EQUATION EQ1U.LS Y1 C Y1(–1) Y2(–1)
GENR EU1 = RESID
EQUATION EQ2U.LS Y2 C Y1(–1) Y2(–1)
GENR EU2 = RESID
MATRIX(2,2) GU
GU(1,1) = @COV(EU1,EU1)
GU(1,2) = @COV(EU1,EU2)
GU(2,1) = @COV(EU2,EU1)
GU(2,2) = @COV(EU2,EU2)
scalar LU = log(@det(GU))
EQUATION EQ1R.LS Y1 C Y1(–1)
GENR ER1 = RESID
MATRIX(2,2) GR1
GR1(1,1) = @COV(ER1,ER1)
GR1(1,2) = @COV(ER1,EU2)
GR1(2,1) = @COV(EU2,ER1)
GR1(2,2) = @COV(EU2,EU2)
scalar LR1 = log(@det(GR1))
scalar LE1 = 70*(LR1–LU)
EQUATION EQ2R.LS Y2 C Y2(–1)
GENR ER2 = RESID
MATRIX(2,2) GR2
GR2(1,1) = @COV(EU1,EU1)
GR2(1,2) = @COV(EU1,ER2)
GR2(2,1) = @COV(ER2,EU1)
GR2(2,2) = @COV(ER2,ER2)
scalar LR2 = log(@det(GR2))
scalar LE2 = 70*(LR2–LU)

' Test de causalité de SIMS
'
SMPL 78:2 95:3
equation EQU1.LS Y1 C Y1(–1) Y2(–1) Y2(1)
scalar SCRU1 = @SSR
scalar NDLU1 = @REGOBS–@NCOEF
```

```
equation EQR1.LS Y1 C Y1(–1) Y2(–1)
scalar SCRR1 = @SSR
scalar NDLR1 = @REGOBS–@NCOEF
SCALAR F1N = (SCRR1 – SCRU1) /(NDLR1 – NDLU1)
SCALAR F1D =SCRU1/NDLU1
SCALAR FS1 = F1N / F1D
equation EQU2.LS Y2 C Y1(–1) Y2(–1) Y1(1)
scalar SCRU2 = @SSR
scalar NDLU2 = @REGOBS–@NCOEF
equation EQR2.LS Y2 C Y1(–1) Y2(–1)
scalar SCRR2 = @SSR
scalar NDLR2 = @REGOBS–@NCOEF
SCALAR F2N = (SCRR2 – SCRU2) /(NDLR2 – NDLU2)
SCALAR F2D =SCRU2/NDLU2
SCALAR FS2 = F2N/F2D
delete  SCRU1 SCRU2 SCRR1 SCRR2 F1N F2N F1D F2D NDLU1 NDLR1 NDLU2 NDLR2
```

Chapitre 11 – Exercice 2 : Tests de cointégration

Test de Johansen de cointégration : Sélection des trois variables : Y1 Y2 Y3 puis <*Open Group*> <*View*> <*Test de Cointegration*>, puis choix du modèle avec Constante et sans tendance (Option 3).

Sample: 1 30
Included observations: 28
Test assumption: Linear deterministic trend in the data
Series: Y1 Y2 Y3
Lags interval: 1 to 1

Eigenvalue	Likelihood Ratio	5 Percent Critical Value	1 Percent Critical Value	Hypothesized No. of CE(s)
0.603749	37.38888	29.68	35.65	None **
0.229311	11.46904	15.41	20.04	At most 1
0.138550	4.175870	3.76	6.65	At most 2 *

*(**) denotes rejection of the hypothesis at 5%(1%) significance level
L.R. test indicates 1 cointegrating equation(s) at 5% significance level

Statistiques issues d'une équation de régression

Par défaut, ces statistiques font références à la dernière équation de régression utilisée ; cependant, ces fonctions peuvent être précédées du nom de l'équation de régression : vente@R2, renvoie la valeur du coefficient de détermination de l'équation de régression portant le nom vente.

@R2	R2 statistic
@RBAR2	adjusted R2 statistic
@SE	standard error of the regression
@SSR	sum of squared residuals
@DW	Durbin–Watson statistic
@F	F–statistic
@LOGL	value of the log–likelihood function
@REGOBS	number of observations in regression
@AIC	Akaike Information Criterion
@SC	Schwartz Criterion
@MEANDEP	mean of the dependent variable
@SDDEP	standard deviation of the dependent variable
@NCOEF	total number of estimated coefficients
@COVARIANCE(i,j)	covariance of coefficients i and j.
@RESIDCOVA(i,j)	covariance of residuals from equation i with those in equation j in a VAR or system object. @RESIDCOVA must be preceded with a named object, i.e. VAR1.@RESIDCOVA(2,2)

Opérateurs et fonctions usuels

+	add
–	subtract
*	multiply
/	divide
^	raise to the power
>	greater than; X>Y has the value 1 if X exceeds Y and zero otherwise.
<	less than; X<Y has the value 1 if Y exceeds X and zero otherwise
=	equal; X=Y has the value 1 if X equals Y and zero otherwise
<>	not equal; X<>Y has the value 1 if X differs from Y and zero otherwise
<=	less than or equal; X<=Y has the value one if X does not exceed Y and zero otherwise
>=	greater than or equal; X>=Y has the value 1 if X is greater than or equal to Y
AND	logical AND; X AND Y has the value 1 if both X and Y are nonzero
OR	logical OR; X OR Y has the value one if X or Y is nonzero
D(X)	first difference of X, X – X(–1)
D(X,n)	n–th order difference of X: , where L is the lag operator
D(X,n,s)	n–th order ordinary differencing and a seasonal difference at lag s:
LOG(X)	natural logarithm
DLOG(X)	change in the natural logarithm, LOG(X)– LOG(X(–1))
DLOG(X,n)	n–th order log difference of X: LOG(X), where L is the lag operator
DLOG(X,n,s)	n–th order log differencing and a seasonal difference at lag s: LOG(X)
EXP(X)	exponential function
ABS(X)	absolute value
SQR(X)	square root
SIN(X)	sine
COS(X)	cosine
@ASIN(X)	arc sine
@ACOS(X)	arc cosine
RND	uniformly distributed random number between zero and one
NRND	normally distributed random number with zero mean and variance of one
@PCH(X)	percent change (decimal), (X–X(–1))/X(–1)
@INV(X)	inverse or reciprocal, 1/X

@DNORM(X)	standard normal density
@CNORM(X)	cumulative normal distribution
@LOGIT(X)	logit of X:
@FLOOR(X)	convert to integer by rounding down; returns the largest integer not greater than X
@CEILING(X)	convert to integer by rounding up; returns the smallest integer not less than X

Calcul d'un scalaire à partir d'une série

@SUM(X)	sum of X
@MEAN(X)	mean of X
@VAR(X)	variance of X
@SUMSQ(X)	sum of squared X
@OBS(X)	number of valid observations in X
@COV(X,Y)	covariance between X and Y
@COR(X,Y)	correlation between X and Y
@CROSS(X,Y)	cross product of X and Y

Fonctions diverses

@MOVAV(X,n)	n period moving average of X, where n is an integer
@MOVSUM(X,n)	n period moving sum of X, where n is an integer
@TREND(d)	time trend variable normalized to be zero in period d, where d is a date or observation number
@SEAS(d)	seasonal dummy equal to one when the quarter or month equals d and zero otherwise.

Lois de probabilité

@DNORM(X)	standard normal density function of X
@CNORM(X)	standard cumulative normal distribution function of X
@TDIST(X, d)	Probability that a t–statistic exceeds X with d degrees of freedom
@FDIST(X, n, d)	Probability that an F–statistic exceeds X with n numerator degrees of freedom and d denominator degrees of freedom
@CHISQ(X, d)	Probability that a Chi–squared statistic exceeds X with d degrees of freedom