

```

. * CORRIGE DE L'EXERCICE CHAPITRE 3 - EXERCICE 1
. *Dalila Chenaf-Nicet Université de Bordeaux.
.
. * Tout d'abord s'assurer d'avoir ouvert le fichier de données stata C3EX1 avant d'ouvrir
le do.fileC3EX1 qui est fichier programme,
. * Une fois le fichier de données ouvert ainsi que le do-file appuyer sur Run (Exécute en
haut à droite de la barre de menu du dofile)
. *pour démarrer. Le programme sera exécuté dans son intégralité
.
. * Toutefois il est possible en sélectionnant les parties du programme de l'exécuter pas à
pas afin de voir apparaître pas à pas les
. * différents résultats.
.
. * On trouvera ici les exercices 1, 2 et 3 du chapitre 3 qui utilise à chaque fois le même
fichier C3EX1
.
. *EXERCICE 1 DU CHAPITRE 3 (Fichier
de données C3EX1)
.
. *Définir les différentes matrices
.
. *Créer la colonne de la constante qui ne prendra que des 1 et sera intégrée dans la
matrice des variables explicatives X (1, x11, x12, X13)
.
. gen b=1
.
.
. *Créer les vecteurs colonnes de données comme étant des matrices X et Y
.
. mkmat b x1 x2 x3, mat(X)
.
. mkmat y, mat(Y)
.
. *Présenter la matrice de l'ensemble des données utilisées dans l'exercice.
.
. mkmat y b x1 x2 x3, mat(Tableau1)
.
. mat list Tableau1

Tableau1[14,5]
      y      b      x1      x2      x3
r1    12      1      2      45     121
r2    14      1      1      43     132
r3    10      1      3      43     154
r4    16      1      6      47     145
r5    14      1      7      42     129
r6    19      1      8      41     156
r7    21      1      8      32     132
r8    19      1      5      33     147
r9    21      1      5      41     128
r10   16      1      8      38     163
r11   19      1      4      32     161
r12   21      1      9      31     172
r13   25      1     12      35     174
r14   21      1      7      29     180
.
. * Créer la matrice X'X et (X'X)-1
.
. matrix XX=X'*X
.
. mat list XX

symmetric XX[4,4]
      b      x1      x2      x3
b      14
x1      85      631
x2     532     3126     20666
x3     2094     13132     78683     317950

```

```

. matrix invXX=syminv(XX)

. mat list invXX

symmetric invXX[4,4]
      b          x1          x2          x3
b    20.168645
x1   .01506566   .0132049
x2   -.23145021   .0011944   .00363504
x3   -.07617483  -.00094019   .00057542   .00040126

.
. * Créer la matrice X'Y
. matrix XY=X'*Y

. mat list XY

XY[4,1]
      y
b    248
x1   1622
x2   9202
x3   37592

.
. *Faire le calcul final de la matrice des a chapeau, noté ahat (coefficients estimés)
.
. matrix ahat=invXX*XY

. mat list ahat

ahat[4,1]
      y
b    32.891324
x1    .80190069
x2   -.38136236
x3   -.03713244

.
. *Créer un vecteur colonne des valeurs de ahat.
.
. svmat double ahat, name(achapeau)

. * Calculer l'écart-type estimé de ahat avec tout d'abord la matrice des résidus
.
. matrix e=Y-X*ahat

. mat list e

e[14,1]
      y
r1   -.84079453
r2    1.6068382
r3   -3.1800496
r4    1.6055059
r5   -3.6973256
r6    1.1219871
r7   -1.2014526
r8    .14259837
r9    4.487981
r10  -2.7621729
r11    1.0829908
r12   -.89941822
r13    2.294594
r14    .23871792

. svmat e

.
. * La variance estimée des résidus
.

```

```

. matrix variancee=(e'*e)/(14-3-1)

. mat list variancee

symmetric variancee[1,1]
      y
y  6.7447666

. svmat variancee

. gen ecarttyperésidus=sqrt(variancee)
(13 missing values generated)

. display ecarttyperésidus
2.5970688

.
. * La diagonale de la matrice des variances et covariances
.
. matrix kk=vecdiag(invXX)

. mat list kk

kk[1,4]
      b      x1      x2      x3
r1  20.168645  .0132049  .00363504  .00040126

.
. * La matrice des variances et co-variances
.
. matrix vacova=variancee*inv(XX)

. mat list vacova

symmetric vacova[4,4]
      b      x1      x2      x3
b    136.0328
x1   .10161439  .08906395
x2  -1.5610777  .00805596  .02451751
x3  -0.51378146 -0.00634136  .0038811  .00270641

.
. *Les variances estimées de chaque coefficient sont sur la diagonale de la matrice vacova
.
. matrix variancedescoef=vecdiag(vacova)

. mat list variancedescoef

variancedescoef[1,4]
      b      x1      x2      x3
r1   136.0328  .08906395  .02451751  .00270641

.
. *Les écarts-types estimés de chaque coefficients
.
. matrix variancedescoeft= variancedescoef'

. mat list variancedescoeft

variancedescoeft[4,1]
      r1
b    136.0328
x1   .08906395
x2   .02451751
x3   .00270641

.
. * La creation d'un vecteur colonne avec les valeurs des variances des coefficients
.
. svmat variancedescoeft

```

```
. display variancedescoef
136.03281
```

```
.
. *Créer une colonne pour les ecart-types estimés des coefficients
.
. gen ecarttypedescoef=sqrt(variancedescoef)
(10 missing values generated)

.
. *Faire de la colonne des écartype des coefficients une matrice pour les calculs futurs
.
. mkmat ecarttypedescoef, mat(mecartcoef)

. mat list mecartcoef
```

```
mecartcoef[14,1]
      ecarttyped~f
r1      11.66331
r2      .29843584
r3      .15658069
r4      .05202312
r5      .
r6      .
r7      .
r8      .
r9      .
r10     .
r11     .
r12     .
r13     .
r14     .
```

```
.
. *Le calcul du R2, la moyenne de y et la somme des carrés des résidus
.
. egen ybarre=mean(y)

. gen ecartyy2=(y-ybarre)^2

. egen sumecarty=sum(ecartyy2)

. display sumecarty
226.85715

. gen insunecarty=(1/sumecarty)

. display insunecarty
.00440806
```

```
. mkmat insunecarty
```

```
. matrix e22=e'*e
```

```
. mat list e22
```

```
symmetric e22[1,1]
```

```
      y
y 67.447666
```

```
. matrix r=[insunecarty]*e22
```

```
. matrix list r
```

```
r[14,1]
      y
r1 .2973134
r2 .2973134
r3 .2973134
r4 .2973134
r5 .2973134
```

```

r6 .2973134
r7 .2973134
r8 .2973134
r9 .2973134
r10 .2973134
r11 .2973134
r12 .2973134
r13 .2973134
r14 .2973134

```

```

.
. *Pour le calcul du R2 qui pour l'instant à la forme d'un vecteur colonne, il faut créer
une matrice unité.
.

```

```

. matrix I=(1\1\1\1\1\1\1\1\1\1\1\1\1\1\1\1)

```

```

. matrix R2=I-r

```

```

. mat list R2

```

```

R2[14,1]

```

```

      y
r1 .7026866
r2 .7026866
r3 .7026866
r4 .7026866
r5 .7026866
r6 .7026866
r7 .7026866
r8 .7026866
r9 .7026866
r10 .7026866
r11 .7026866
r12 .7026866
r13 .7026866
r14 .7026866

```

```

.
. *Faire de R2 un scalaire pour les calculs futurs
. svmat R2

```

```

. display R2
.70268661

```

```

. * Calculer le R2 ajusté
. matrix R2adj=I-[(14-1)/(14-3-1)]*(I-R2)

```

```

. svmat R2adj

```

```

. display R2adj
.61349261

```

```

.
.
données C3EX1)

```

```

. * Test sur les coefficients

```

```

.
. * Il faut calculer les trois ratios de student et les comparer à la valeur de a table au
seuil de 5%.

```

```

. * On va utiliser les coefficients (ahat) et leur ecart-type estimé (ecartcoef), stocké dans
les matrices ahat et mecartcoef

```

```

. mat list ahat

```

```

ahat[4,1]

```

```

      y
b 32.891324
x1 .80190069
x2 -.38136236
x3 -.03713244

```

```
. mat list mecartcoef
```

```
mecartcoef[14,1]
      ecarttyped~f
r1      11.66331
r2      .29843584
r3      .15658069
r4      .05202312
r5      .
r6      .
r7      .
r8      .
r9      .
r10     .
r11     .
r12     .
r13     .
r14     .
```

```
.
. *Le calcul des 4 t de student correspondant aux 4 coefficients.
```

```
.
. gen student =achapeaul/ecarttypedescoef
(10 missing values generated)
```

```
.
. *Il faut donner les valeurs des quatre t-student . Le premier étant celui de la constante.
```

```
. mkmat student, mat(student)
```

```
. mat list student
```

```
student[14,1]
      student
r1      2.8200676
r2      2.687012
r3      -2.4355645
r4      -.71376789
r5      .
r6      .
r7      .
r8      .
r9      .
r10     .
r11     .
r12     .
r13     .
r14     .
```

```
.
. *Le test d'hypothèse pour les 4 coefficients séparément
```

```
.
. gen H0=abs(achapeaul-1)/ecarttypedescoef
(10 missing values generated)
```

```
. mkmat H0, mat(h0)
```

```
. mat list h0
```

```
h0[14,1]
      H0
r1      2.7343287
r2      .66379195
r3      8.8220482
r4      19.935989
r5      .
r6      .
r7      .
r8      .
r9      .
```

```

r10      .
r11      .
r12      .
r13      .
r14      .

.
. * Le test d'hypothèse pour deux coefficients simultanément. Estimation de la relation 15 de
l'ouvrage avec q=2 paramètres à estimer
. * On extrait dans la matrice a chapeau les deux coefficients avec lesquels on fait les
tests (coef 1 de la variable x1 et coef 2 de
. * la variable x2)
.
. matrix extrac=ahat[2..3,1]

. mat list extrac

extrac[2,1]
           y
x1      .80190069
x2     -.38136236

.
. *On crée aq
.
. matrix aq=(1\ -0.5)

. mat list aq

aq[2,1]
      c1
r1      1
r2     -.5

.
. * Calculer la matrice et la transposée de (achapeau -aq) et multiplier par q=2
.
. matrix Q=(extrac-aq)

. mat list Q

Q[2,1]
      c1
r1    -.19809931
r2     .11863764

. matrix Qt=Q'

. mat list Qt

Qt[1,2]
      r1      r2
c1    -.19809931  .11863764

.
. *La matrice des variances et covariance a été calculée dans la partie 1 de l'exercice
(C3EX1)
.
. mat list vacova

symmetric vacova[4,4]
           b           x1           x2           x3
b      136.0328
x1     .10161439  .08906395
x2    -1.5610777  .00805596  .02451751
x3    -0.51378146 -0.00634136  .0038811  .00270641

.
. *Nous ne retenons que la sous-matrice de dimension 2*2 correspondant aux deux coefficients
de régression du test.
.

```

```

. matrix sigmaaq=vacova[2..3,2..3]

. mat list sigmaaq

symmetric sigmaaq[2,2]
      x1      x2
x1  .08906395
x2  .00805596  .02451751

.
. * Calculer l'inverse de la matrice
.
. matrix invsigmaaq=inv(sigmaaq)

. mat list invsigmaaq

symmetric invsigmaaq[2,2]
      x1      x2
x1  11.571807
x2 -3.8022623  42.036519

.
. *Calculer le F avec la formule
.
. matrix F=(1/2)*Qt*invsigmaaq*Q

. mat list F

symmetric F[1,1]
      c1
c1  .61224873

.
. *Comparer avec le Ft de la table  $F < F_t$ . On accepte  $H_0$ 
.
. *Calculer l'intervalle de confiance de la variance de l'erreur à un seuil de 95%. Il est
calculé à partir de la formule [16]
. * de l'ouvrage.
.
. matrix IC=[(14-3-1)/(20.48)*6.745,(14-3-1)/(3.25)*6.745]

. mat list IC

IC[1,2]
      c1      c2
r1  3.293457  20.753846

.
. *Le calcul de hi et des résidus standardisés passent par le calcul de la matrice Hat
.
. matrix HAT=X*inv(X'*X)*X'

. mat list HAT

symmetric HAT[14,14]
      r1      r2      r3      r4      r5      r6
      r7      r8      r9      r10     r11     r12
>      r13     r14
r1  .27900806
r2  .24956842  .29656946
r3  .13309827  .22069823  .30911977
r4  .12652511  .08994954  .19321433  .32482111
r5  .14501625  .02592779  -.05147985  .13434158  .26087536
r6  .02490035  -.00450257  .09634554  .21056056  .10914624  .18254374
r7  .07423364  -.0368327  -.23815384  -.1413147  .23865843  -.02944797  .5327074
r8  .07986319  .10751744  .00827768  -.10245388  .03094189  -.04416872  .21388054
.20252966
r9  .18156209  .11227683  .00142248  .07272155  .19112854  .04373446  .21095137
.09018302  .18040529
r10 -.01278565  -.00636165  .0979735  .14444332  .04992947  .14926847  -.02680998
-.00167603  .01064138  .14420255

```



```

r11 .04373917 .15830889 .13555818 -.09879498 -.10481209 -.04918614 .036247
.20054459 .00034406 .02550396 .30664868
r12 -.09279531 -.05953444 .0077156 -.02074448 -.00180653 .07635301 .10448682
.10104937 -.01343048 .11900567 .13192369 .21148445
r13 -.13906413 -.18374354 -.02522948 .14839028 .10893074 .21798637 .06584473
-.04159959 -.00727492 .20480188 -.06854039 .20742583
> .40856714
r14 -.09286945 .03015828 .11143959 -.08165936 -.13679783 .01646666 -.00445074
.15511085 -.07466567 .10186311 .28251538 .22886681
> .10350506 .36051731

```

```

.
. *Le levier est sur la première diagonale de la matrice
.
. matrix ht=vecdiag(HAT)

```

```

. mat list ht

```

```

ht[1,14]
      r1      r2      r3      r4      r5      r6      r7
      r8      r9     r10     r11     r12     r13
>      r14
r1 .27900806 .29656946 .30911977 .32482111 .26087536 .18254374 .5327074 .20252966
.18040529 .14420255 .30664868 .21148445 .40856714 .
> 36051731

```

```

.
. *Transposer la matrice ht pour la récupérer comme un vecteur colonne, puis en faire des
scalaires (svmat) pour en faciliter le calcul

```

```

. matrix h=ht'

```

```

. mat list h

```

```

h[14,1]
      r1
r1 .27900806
r2 .29656946
r3 .30911977
r4 .32482111
r5 .26087536
r6 .18254374
r7 .5327074
r8 .20252966
r9 .18040529
r10 .14420255
r11 .30664868
r12 .21148445
r13 .40856714
r14 .36051731

```

```

. svmat h

```

```

.
. *Calculer les résidus standardisés
.
. gen es=e1*1/[(2.597)*sqrt(1-h)]

```

```

. mkmat e1 h es, mat(Tableau2)

```

```

. mat list Tableau2

```

```

Tableau2[14,3]
      e1      h1      es
r1 -.8407945 .27900806 -.38128763
r2 1.6068382 .29656947 .73771667
r3 -3.1800497 .30911976 -1.4731961
r4 1.6055059 .32482111 .75236839
r5 -3.6973255 .26087537 -1.6559869
r6 1.1219871 .18254374 .47784126
r7 -1.2014526 .53270739 -.67676866

```

```

r8      .14259836      .20252967      .06148728
r9      4.4879808      .18040529      1.9088837
r10     -2.7621729      .14420256     -1.1497235
r11      1.0829908      .30664867      .5008136
r12     -.89941823      .21148445     -.39001796
r13      2.294594      .40856713      1.1488973
r14      .23871791      .36051729      .11494727

```

```

.
. *Calculer le seuil du levier
.

```

```

. gen seuil=2*(3+1)/14

```

```

. display seuil
.5714286

```

```

. *****EXERCICE 3 DU CHAPITRE (Fichier de données C3EX1)

```

```

. *Analyse de la variance

```

```

. *On applique le test F-Fisher avec trois variables. Attention le R2 est ici noté R21.
(Stata code automatiquement différemment une colonne de donn
> ées

```

```

. * qui a eut le statut de vecteur colonne (R2) puis de scalaires (R21).

```

```

. gen F=(R21/3)/(1-R21)*(14-3-1)

```

```

. display F
7.878181

```

```

.
. *On rejette donc l'hypothèse H0. Les coefficients de la relation sont globalement
significatifs.

```

```

.
. *Le test d'ajout de variable

```

```

. *Etape 1, calculer SCT ; SCR, SCE. Certains éléments sont déjà calculé dans l'exercice 1
pour le modèle global avec 3 variables.

```

```

. egen SCT=sum(ecarty2)

```

```

. display SCT
226.85715

```

```

. egen SCR=sum(e1*e1)

```

```

. display SCR
67.447662

```

```

. gen yestimé=y-e1

```

```

. egen SCE=sum((yestimé-ybarre)^2)

```

```

. display SCE
159.40947

```

```

.
. *Etape 2 : Il faut récupérer les grandeurs précédentes dans le cas d'une seule variable
explicative.

```

```

. * On récupère SCR nommée rss et SCT nommée mss sous stata que l'on indice par 1.

```

```

. regress y x1

```

Source	SS	df	MS	Number of obs	=	14
Model	117.658883	1	117.658883	F(1, 12)	=	12.93
Residual	109.19826	12	9.09985498	Prob > F	=	0.0037
				R-squared	=	0.5186
				Adj R-squared	=	0.4785
Total	226.857143	13	17.4505495	Root MSE	=	3.0166

y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
x1	1.011809	.2813865	3.60	0.004	.3987201	1.624897
_cons	11.57116	1.889095	6.13	0.000	7.455177	15.68715

```
. gen SCR1=e(rss)
```

```
. display SCR1
109.19826
```

```
. gen SCE1=e(mss)
```

```
. display SCE1
117.65888
```

```
.
. *Calcul de la somme des carrés total du modèle a une variable.
```

```
. gen SCT1=SCE1+SCR1
```

```
. display SCT1
226.85715
```

```
.
. *Construire le tableau d'analyse de la variance (SCR SCR1 SCT SCT1 SCE SCE1) du modèle avec
trois variables et avec une variable.
```

```
. mkmat SCT, mat(SCT)
```

```
. matrix SCTTotal=el(SCT,1,1)
```

```
. svmat SCTTotal
```

```
. mkmat SCT1, mat(SCT1)
```

```
. matrix SCTTotal1=el(SCT1,1,1)
```

```
. svmat SCTTotal1
```

```
. mkmat SCE, mat(SCE)
```

```
. matrix SCEExpliqué=el(SCE,1,1)
```

```
. svmat SCEExpliqué
```

```
. mkmat SCE1, mat(SCE1)
```

```
. matrix SCEExpliqué1=el(SCE1,1,1)
```

```
. svmat SCEExpliqué1
```

```
. mkmat SCR, mat(SCR)
```

```
. matrix SCRésidus=el(SCR,1,1)
```

```
. svmat SCRésidus
```

```
. mkmat SCR1, mat(SCR1)
```

```
. matrix SCRésidus1=el(SCR1,1,1)
```

```
. svmat SCRésidus1
```

```
.
. *Récupérer les 6 valeurs dans un tableau nommé sommedescarrées
```

```
. mkmat SCTTotal1 SCTTotal11 SCEExpliqué1 SCEExpliqué11 SCRésidus1 SCRésidus11,
mat(sommedescarrées)
```

```
. matrix sommedescarréest=sommedescarrées'
```

```
. mat list sommedescarréest
```

```
sommedescarréest[6,14]
```

```

          r1      r2      r3      r4      r5      r6
          r7      r8      r9     r10     r11     r12
>      r13      r14
SCTotal1 226.85715      .      .      .      .      .
.      .      .      .      .      .      .
>      SCTotal11 226.85715      .      .      .      .      .
.      .      .      .      .      .      .
>      SCExpliqué1 159.40947      .      .      .      .      .
.      .      .      .      .      .      .
>      SCExpliqué11 117.65888      .      .      .      .      .
.      .      .      .      .      .      .
>      SCRésidus1 67.447662      .      .      .      .      .
.      .      .      .      .      .      .
>      SCRésidus11 109.19826      .      .      .      .      .
.      .      .      .      .      .      .
>      .      .      .      .      .      .      .

```

```
. *Calcul du Fisher empirique avec k=3 le nombre de variables explicatives du modèle globale
et k'= 1, le nombre
```

```
. * de variables du modèle à 1 variable. _N=14 le nombre de variables explicatives.
```

```
. gen Fisher=[(SCR1-SCR)/(3-1)]/[SCR/(_N-3-1)]
```

```
. display Fisher
```

```
3.0950365
```

```
. * Etape 3 : Le modèle avec 3 variables est-il stable sous la période ? On fait d'abord les
estimations pour la période qui
```

```
. * va de la 1ere à la 7ème période et on récupère la SCT, SCR, SCE d'indics(m1) pour le
modèle 1
```

```
. regress y x1 x2 x3 in 1/7
```

Source	SS	df	MS	Number of obs	=	7
Model	61.5395684	3	20.5131895	F(3, 3)	=	2.25
Residual	27.3175744	3	9.10585814	Prob > F	=	0.2610
				R-squared	=	0.6926
				Adj R-squared	=	0.3851
Total	88.8571429	6	14.8095238	Root MSE	=	3.0176

y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
x1	.7739161	.5290333	1.46	0.240	-.9097041 2.457536
x2	-.2931763	.313677	-0.93	0.419	-1.291437 .7050839
x3	-.0125032	.1008087	-0.12	0.909	-.3333214 .308315
_cons	25.2756	16.66755	1.52	0.227	-27.76798 78.31919

```
. gen SCEm1=e(mss)
```

```
. display SCEm1
```

```
61.53957
```

```
. gen SCRm1=e(rss)
```

```
. display SCRm1
```

```
27.317574
```

```
. gen SCTm1=SCEm1+SCRm1
```

```
. display SCTm1
88.857147
```

```
.
. * On fait les estimations pour la période qui va de la 8ème à la 14ème période et on
récupère la SCT, SCR, SCE indicés(m2)
. * pour modèle 2
.
. regress y x1 x2 x3 in 8/14
```

Source	SS	df	MS	Number of obs	=	7
Model	24.7066831	3	8.23556103	F(3, 3)	=	1.19
Residual	20.7218883	3	6.90729611	Prob > F	=	0.4442
				R-squared	=	0.5439
				Adj R-squared	=	0.0877
Total	45.4285714	6	7.57142857	Root MSE	=	2.6282

y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
x1	1.228196	.6852332	1.79	0.171	-.9525222	3.408914
x2	-.6208326	.5223629	-1.19	0.320	-2.283224	1.041559
x3	-.1843387	.152831	-1.21	0.314	-.6707152	.3020379
_cons	62.33574	37.23454	1.67	0.193	-56.1612	180.8327

```
. gen SCEm2=e(mss)
```

```
. display SCEm2
24.706682
```

```
. gen SCRm2=e(rss)
```

```
. display SCRm2
20.721888
```

```
. gen SCTm2=SCEm2+SCRm2
```

```
. display SCTm2
45.42857
```

```
.
. * On peut a présent calculer le Fisher avec la formule de l'ouvrage en faisant attention
aux ddl.
```

```
. gen Fisherempirique=(SCR-(SCRm1+SCRm2))/(3+1)/[(SCRm1 +SCRm2)/(6)]
```

```
. display Fisherempirique
.60600811
```

```
. drop Fisherempirique
```

```
.
. * Les coefficients sont globalement stables sur l'ensemble de la période puisque le F de
la table = 4.53
```

```
. *Etape 4 : Le test pour a1=1 et a2=a3.
```

```
. *Si l'hypothèse est vraie le modèle s'écrit z= a0 + a2*v. Il faut donc créer z et t
```

```
. gen z=y-x1
```

```
. gen v=x2+x3
```

```
. mkmat z v, mat(variablestransformées)
```

```
. mat list variablestransformées
```

```
variablestransformées[14,2]
```

```
      z      v
r1    10    166
```

```

r2  13  175
r3   7  197
r4  10  192
r5   7  171
r6  11  197
r7  13  164
r8  14  180
r9  16  169
r10  8  201
r11 15  193
r12 12  203
r13 13  209
r14 14  209

```

```

.
. * On récupère SCTzv SCEsv et SCRsv
.
. regress z v

```

Source	SS	df	MS	Number of obs	=	14
Model	.425473883	1	.425473883	F(1, 12)	=	0.05
Residual	108.788812	12	9.06573432	Prob > F	=	0.8321
				R-squared	=	0.0039
				Adj R-squared	=	-0.0791
Total	109.214286	13	8.4010989	Root MSE	=	3.0109

z	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
v	-.0111547	.0514902	-0.22	0.832	-.1233423 .1010328
_cons	13.73517	9.691559	1.42	0.182	-7.380925 34.85126

```

. gen SCEzv=e(mss)

```

```

. display SCEzv
.42547387

```

```

. gen SCRzv=e(rss)

```

```

. display SCRzv
108.78881

```

```

. gen SCTsv=SCEzv+SCRzv

```

```

. display SCTsv
109.214286

```

```

. * Le Fisherempirique est donné par

```

```

. gen Femp=(SCRzv - SCTsv)/2/[(SCR/10)]

```

```

. display Femp
3.0646837

```

```

. * On peut conclure que l'hypothèse H0 est acceptée

```

```

end of do-file

```

```

. exit, clear

```