Commandes Stata et utilitaires
Au 06/10/17

Analyse
cpairs
ggof
tmt3
qlt
competout
gpct

margins_transform (J.Pitblado – StataCorp)

Utilitaires - astuces
- profile.do
- Affecter plusieurs répertoires en début de programme (cf libname avec Sas)
- dossiers stata_temp et stata_temp/log
- tuse – tsave – terase

Les programmes et les fichiers d'aide se trouvent dans l'archive zip « Stata ». Les .ado et les fichiers d'aide peuvent être installés (collés) dans le répertoire ado (de préférence dans un sous-répertoire « personal»).

Pour toute question ou problème rencontré, et pour toute suggestion: marc.thevenin@ined.fr

Analyse

cpairs (postestimation)

Calcule quelques statistiques de mesure de la qualité de l'ajustement après un modèle de type logit, probit, cloglog. Le temps d'exécution de la commande peut être un peu long.
Statistiques: proportions de paires concordantes et discordantes, D de Somer, Tau-a, Gamma et aire sous la courbe de ROC (c-AUC)

Exemple

```
webuse lbw, clear
(Hosmer & Lemeshow data)

logit low age lwt i.race smoke ptl ht ui

Iteration 0:   log likelihood =   -117.336
Iteration 1:   log likelihood = -101.28644
Iteration 2:   log likelihood = -100.72617
Iteration 3:   log likelihood =   -100.724
Iteration 4:   log likelihood =   -100.724

Logistic regression                             Number of obs     =        189
                                                LR chi2(8)        =      33.22
                                                Prob > chi2       =     0.0001
Log likelihood =   -100.724                     Pseudo R2         =     0.1416

------------------------------------------------------------------------------
         low |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  -.0271003   .0364504    -0.74   0.457    -.0985418    .0443412
         lwt |  -.0151508   .0069259    -2.19   0.029    -.0287253   -.0015763
             |
        race |
       black |   1.262647   .5264101     2.40   0.016     .2309024    2.294392
       other |   .8620792   .4391532     1.96   0.050     .0013548    1.722804
             |
       smoke |   .9233448   .4008266     2.30   0.021      .137739    1.708951
         ptl |   .5418366    .346249     1.56   0.118     -.136799    1.220472
          ht |   1.832518   .6916292     2.65   0.008     .4769494    3.188086
          ui |   .7585135   .4593768     1.65   0.099    -.1418484    1.658875
       _cons |   .4612239    1.20459     0.38   0.702    -1.899729    2.822176
------------------------------------------------------------------------------

cpairs low

Association of Predicted Probabilities and Observed Responses

      Number of pairs =    7670

Proportion Concordant =   0.746
Proportion Discordant =   0.254
      Proportion Tied =   0.000

            Somer's D =   0.492
                Gamma =   0.493
                Tau-a =   0.213
```

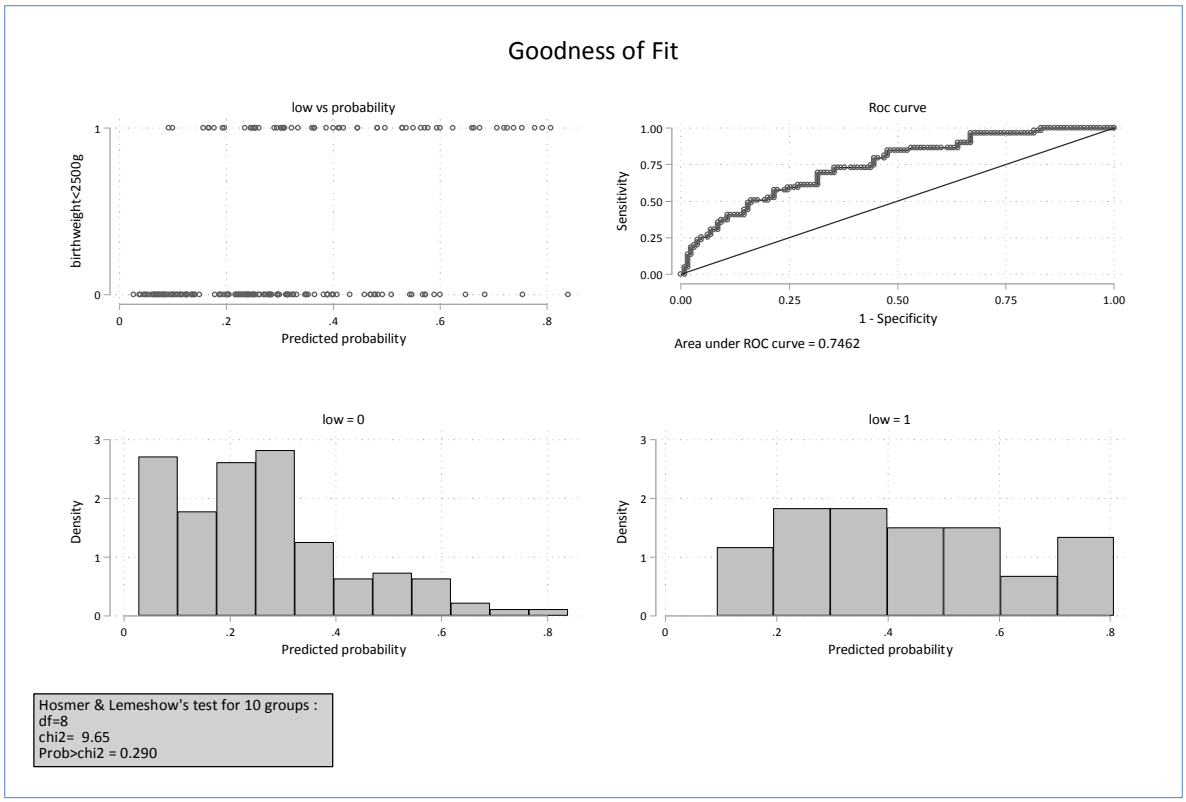
```
c-AUC =   0.746
```

ggof (postestimation)

Sous forme de graphiques combinés, la commande présente des éléments de diagnostique de la qualité de l'ajustement après un modèle de type logit, probit, cloglog.
Graphiques : probabilités prédites vs réponse, courbe de Roc, densités des probabilités prédites selon la réponse. Le graphique affiche également le résultat du test d'Hosmer et Lemeshow pour 10 groupes.

Exemple

`ggof low`



tmt3 (postestimation)

Affiche des tests multiples d'hypothèse nulle après un modèle (b1=b2=…bp=0=). Utile seulement pour les variables catégorielles à plus de 2 modalités.

A voir : affichage des seuils corrigés (type Bonferroni..) pour le nombre de df présent dans l'output.

Exemple avec le modèle logit précédent

```
tmt3 race

Type-III multiple test for categorical covariable(s) after logit

-------------+----------------------------------
   Variables |       df      Khi2      Prob>chi2
-------------+----------------------------------
         race|        2    3.11603      0.211
-------------+----------------------------------
```

Exemple avec mlogit

```
webuse sysdsn1
quietly mlogit insure age male nonwhite i.site
tmt3 site

Output:
Type-III multiple test for categorical covariable(s) after mlogit

For insure = 2
-------------+----------------------------------
   Variables |       df      Khi2      Prob>chi2
-------------+----------------------------------
         site|        2    10.7754      0.005
-------------+----------------------------------

For insure = 3
-------------+----------------------------------
   Variables |       df      Khi2      Prob>chi2
-------------+----------------------------------
         site|        2    6.80668      0.033
-------------+----------------------------------
```

Exemple avec regress

```
webuse auto
quietly gen gweight=weight
quietly recode gweight min/2239=1 2240/3189=2 3190/3599=3 3600/max=4
quietly regress mpg ib2.gweight ib2.rep78 i.foreign
tmt3 gweight rep78

Type-III multiple test for categorical covariable(s) after regress

-------------+--------------------------
   Variables |   df      F      Prob>F
-------------+--------------------------
      gweight| (3,60)    31.5   0.000
        rep78| (4,60)    2.65   0.042
-------------+--------------------------
```

qlt (analyse de survie)

Calcul des durées pour plusieurs quantiles de la fonction de survie estimée à partir de la méthode actuarielle (commande ltable).
La commande **`ltable`** ne permettant de récuper directement la fonction d survie estimées, il convient d'utiliser l'option **`saving(nom_base)`** .

Exemples

```
webuse rat
ltable t died, saving(lt, replace)
```

Interval		Beg. Total	Deaths	Lost	Survival	Std. Error	[95% Conf.	Int.]
142	143	40	1	0	0.9750	0.0247	0.8355	0.9964
143	144	39	1	0	0.9500	0.0345	0.8145	0.9873
156	157	38	1	0	0.9250	0.0416	0.7852	0.9752
163	164	37	1	0	0.9000	0.0474	0.7551	0.9612
164	165	36	1	0	0.8750	0.0523	0.7254	0.9460
188	189	35	2	0	0.8250	0.0601	0.6677	0.9125
190	191	33	1	0	0.8000	0.0632	0.6396	0.8946
192	193	32	1	0	0.7750	0.0660	0.6122	0.8760
198	199	31	1	0	0.7500	0.0685	0.5852	0.8569
204	205	30	0	1	0.7500	0.0685	0.5852	0.8569
205	206	29	1	0	0.7241	0.0708	0.5574	0.8367
206	207	28	1	0	0.6983	0.0729	0.5301	0.8161
209	210	27	1	0	0.6724	0.0746	0.5033	0.7950
213	214	26	1	0	0.6466	0.0761	0.4771	0.7734
216	217	25	1	1	0.6202	0.0774	0.4505	0.7511
220	221	23	1	0	0.5932	0.0786	0.4237	0.7279
227	228	22	1	0	0.5662	0.0795	0.3974	0.7043
230	231	21	1	0	0.5393	0.0802	0.3716	0.6803
232	233	20	2	0	0.4853	0.0807	0.3215	0.6310
233	234	18	4	0	0.3775	0.0788	0.2271	0.5272
234	235	14	1	0	0.3505	0.0776	0.2048	0.5001
239	240	13	1	0	0.3236	0.0762	0.1830	0.4726
240	241	12	1	0	0.2966	0.0745	0.1617	0.4445
244	245	11	0	1	0.2966	0.0745	0.1617	0.4445
246	247	10	1	0	0.2669	0.0727	0.1383	0.4141
261	262	9	1	0	0.2373	0.0704	0.1159	0.3828
265	266	8	1	0	0.2076	0.0676	0.0946	0.3507
280	281	7	2	0	0.1483	0.0599	0.0556	0.2834
296	297	5	2	0	0.0890	0.0484	0.0233	0.2109
304	305	3	1	0	0.0593	0.0404	0.0108	0.1717
323	324	2	1	0	0.0297	0.0291	0.0023	0.1305
344	345	1	0	1	0.0297	0.0291	0.0023	0.1305

```
use lt, clear
qlt
Durée pour différents quantiles de la fonction de survie
S(t)=0.90: t=  163.000
S(t)=0.75: t=  198.000
S(t)=0.50: t=  231.456
S(t)=0.25: t=  254.567
S(t)=0.10: t=  293.028
```

```
webuse rat
quietly ltable t died, saving(lt, replace) by(group)
use lt, clear
bysort group: qlt

-------------------------------------------------------------------------------
-> group = 1
Durée pour différents quantiles de la fonction de survie
S(t)=0.90: t=  161.900
S(t)=0.75: t=  189.500
S(t)=0.50: t=  214.425
S(t)=0.25: t=  233.012
S(t)=0.10: t=  259.792

-------------------------------------------------------------------------------
-> group = 2
Durée pour différents quantiles de la fonction de survie
S(t)=0.90: t=  156.700
S(t)=0.75: t=  207.382
S(t)=0.50: t=  232.779
S(t)=0.25: t=  271.059
S(t)=0.10: t=  296.635
```

competout (analyse de survie)

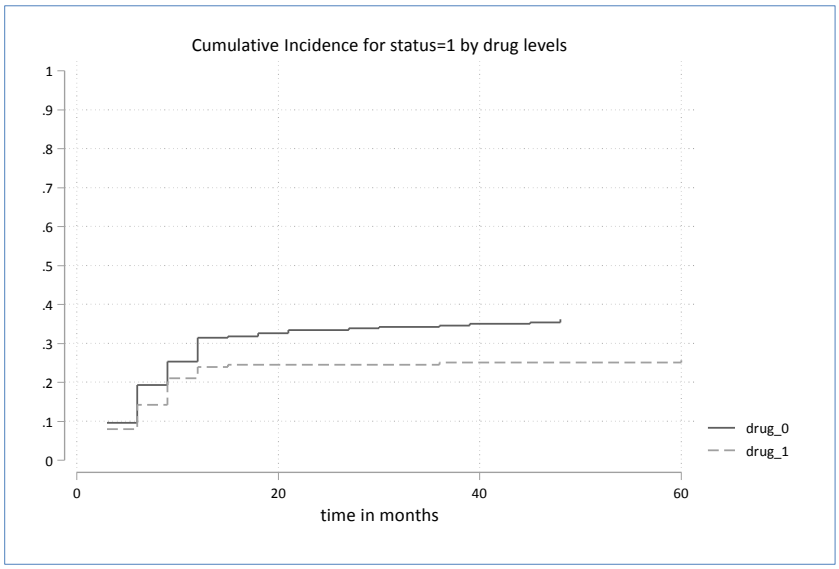
Maj 2017 : Voir également la commande **stcomlist** (janvier 2017 => ssc install stcomlist), très proche si on excepte la question du test de Gray.

Pour l'analyse des risques concurrents (analyses biographique) affichent les sous forme de tableaux les estimateurs des incidences cumulées, et le graphique associé. En option, plusieurs tests sont exécutés. competout utilise des commandes existantes qui sont automatiquement installées si besoin. Pour le test de Gray, il est exécuté par R. Cela nécessite donc une installation particulière décrite dans le fichier d'aide, mais aucune connaissance du langage R n'est requise (le fichier **competout_gray_test.do** doit se trouver dans le même répertoire que **competout.ado**).

Exemple

```
use http://www.stata-press.com/data/cggm3/bc_compete, clear
(Breast cancer with competing risks)
r; t=0.44 10:36:04


competout time status, event(1) group(drug) test(sr)

Cumulative incidence for drug=0

-------------------------------------------------------
analysis  |
time when |
record    |
ends      |         IC          SE      95% LB      95% UB
----------+--------------------------------------------
        3 |     0.0968      0.0188      0.0640      0.1375
        6 |     0.1935      0.0251      0.1471      0.2449
        9 |     0.2540      0.0276      0.2017      0.3095
       12 |     0.3145      0.0295      0.2577      0.3728
       15 |     0.3185      0.0296      0.2615      0.3769
       18 |     0.3266      0.0298      0.2691      0.3853
       21 |     0.3347      0.0300      0.2767      0.3936
       27 |     0.3387      0.0301      0.2805      0.3977
       30 |     0.3427      0.0301      0.2843      0.4019
       36 |     0.3468      0.0302      0.2881      0.4060
       39 |     0.3508      0.0303      0.2919      0.4102
       45 |     0.3548      0.0304      0.2957      0.4143
       48 |     0.3629      0.0305      0.3034      0.4226
-------------------------------------------------------

Cumulative incidence for drug=1

-------------------------------------------------------
analysis  |
time when |
record    |
ends      |         IC          SE      95% LB      95% UB
----------+--------------------------------------------
        3 |     0.0800      0.0205      0.0459      0.1262
        6 |     0.1429      0.0265      0.0959      0.1989
        9 |     0.2114      0.0309      0.1544      0.2746
       12 |     0.2400      0.0323      0.1797      0.3053
       15 |     0.2457      0.0325      0.1848      0.3114
       36 |     0.2514      0.0328      0.1899      0.3175
```

```
       60 |     0.2571      0.0330      0.1950      0.3236
----------------------------------------------------------
```



log-rank test (Cause specific hazards)
Main event failure: status == 1
Chi2(1) = 4.0664
Prob>Chi2 = 0.0437

Pepe and Mori test comparing the cumulative incidence of two groups of drug
Main event failure: status == 1
Chi2(1) = 4.6845 - p = 0.03044

Competing event failure: status == 2
Chi2(1) = 5.6476 - p = 0.01748

Gray's test
using Rsource (Newson) & cmprsk (Gray)

Line 1 - Test for main event failure: status == 1
Line 2 - Test for competing event(s) failure: status == 2

```
rsource, terminator(END_OF_R)
Assumed R program path: "C:\Program Files\R\R-3.3.1\bin\i386\R.exe"
Beginning of R output
      Chi2 df     Pr>Chi2
1 4.908811  1 0.026720033
2 8.615262  1 0.003333579
End of R output

end of do-file
r; t=2.86 10:36:07
```

gpct (descriptif – graphique)

Version très provisoire. Améliorations à venir :

- prise en charge des pondérations
- Reporter les intervalles de confiances, avec une option pour les versions de Stata <15 et une option pour la version 15 afin de bénéficier des effets transparence (enfin !).
- Dans le cas d'une variable (Y) binaire, ne tracer qu'une seule courbe avec choix de la modalité.

Permet de tracer sous forme de courbes des pourcentages issus d'un tableau croisé de type:
`tab X Y, nofreq r` avec X pour abscisse et pour ordonnée les % prises pour chaque valeurs de Y.
Les labels des courbes sont récupérées à partir des labels des modalités de la variables Y, si ces labels sont compatibles avec des noms de variable Stata.

Syntaxe

`gpct var_X var_Y [if/in] [, lab(0/1)]`

option lab(0) sans récupération des labels des modalités de Y
option lab(1) avec récupération des labels des modalités de Y

Exemple

```
use  http://www.stata-press.com/data/r15/nlswork.dta, clear
label define msp 0 "married" 1 "unmarried", modify
label value msp msp
```

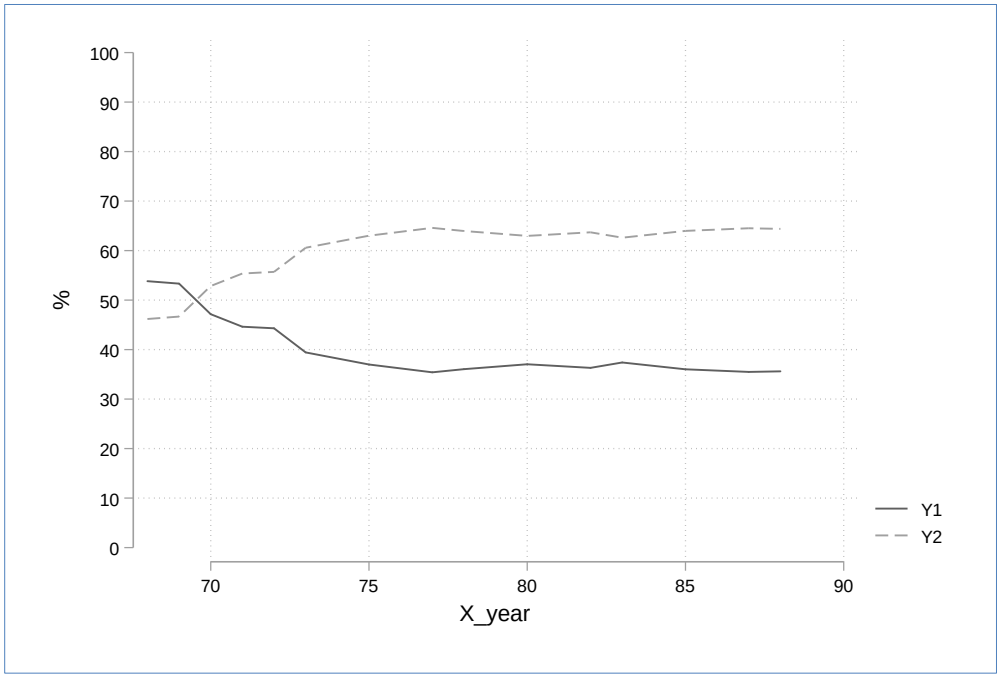
On va tracer les valeurs du tableau:

```
tab year msp, nofreq r
```

interview year	1 if married, spouse present: married	unmarried	Total
68	53.82	46.18	100.00
69	53.33	46.67	100.00
70	47.15	52.85	100.00
71	44.62	55.38	100.00
72	44.30	55.70	100.00
73	39.42	60.58	100.00
75	36.99	63.01	100.00
77	35.41	64.59	100.00
78	36.05	63.95	100.00
80	37.03	62.97	100.00
82	36.31	63.69	100.00
83	37.39	62.61	100.00
85	36.02	63.98	100.00
87	35.49	64.51	100.00
88	35.61	64.39	100.00
Total	39.71	60.29	100.00

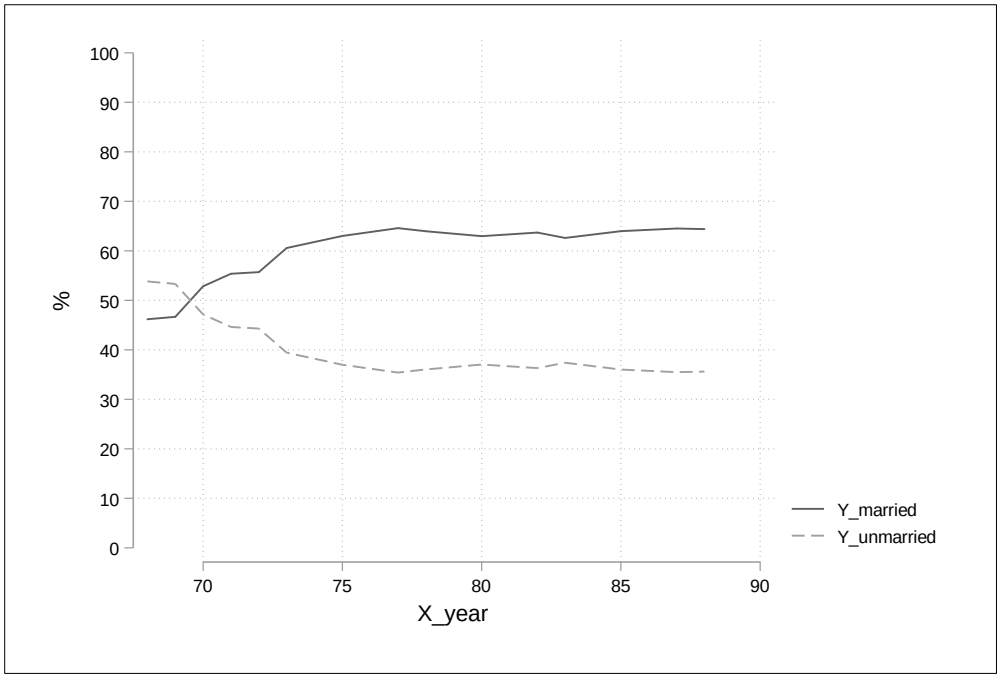
Sans récupération des labels de Y

```
gpct year msp, lab(0)
```



Avec récupération des labels de Y

```
gpct year msp, lab(1)
```



margins_transform [**J.Pitblado**- StataCorp LP]

Correction des bornes des intervalles de confiances des valeurs prédites moyennes générées par margins [avec option predict()]

Appliquées à un modèle non linéaire, la commande margins renvoie des valeurs erronées aux bornes des intervalles de confiance des valeurs prédites moyennes ajustées. Cela se traduit, par exemple, par des valeurs négatives ou supérieures à un lorsqu'il s'agit d'estimer des probabilités.

Dans le cas de la commande margins, le changement d'échelle est effectué immédiatement une fois la combinaison linéaire définie, le calcul de la variance et des bornes venant ensuite ; alors que la méthode correcte consiste à calculer la variance et les bornes sur la combinaison linéaire ajustée, le changement d'échelle s'effectuant en toute fin.

A noter que ce problème ne se pose pas pour le calcul des effets marginaux (option dydx).

J.Pitblado (StataCorp) a programmé une commande pour résoudre ce problème, pour l'instant non officielle, qui calcule correctement les bornes.

Après l'estimation du modèle, la commande margins doit être exécutée avec en option **predict(xb)**.

Si on souhaite récupérer les valeurs des bornes sous forme de variable, pour faire, par exemple, une représentation graphique, on peut utiliser l'option mat(*nom*). Les valeurs des bornes seront données pour les variables *nom2 nom3*.

Exemple (avec installation de la commande)

```
    ssc install transform_margins
qui sysuse auto
qui logit foreign mpg
    margins, at(mpg=(5(5)40))
```

```
------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         _at |
          1  |   .0271183   .0252542     1.07   0.283    -.0223792    .0766157
          2  |   .0583461   .0389888     1.50   0.135    -.0180704    .1347627
          3  |   .1210596   .0509373     2.38   0.017     .0212244    .2208948
          4  |   .2344013   .0547344     4.28   0.000      .127124    .3416787
          5  |   .4049667   .0743318     5.45   0.000      .259279    .5506543
          6  |   .6020462   .1162814     5.18   0.000     .3741387    .8299536
          7  |   .7707955   .1266899     6.08   0.000     .5224879    1.019103
          8  |   .8820117   .1004224     8.78   0.000     .6851874    1.078836
------------------------------------------------------------------------------
```

```
qui margins, at(mpg=(5(5)40)) predict(xb)
    transform_margins invlogit(@), mat(nom)
```

```
------------------------------------------
             |         b         ll         ul
-------------+----------------------------
         _at |
          1  |   .0271183   .0042517    .153952
          2  |   .0583461   .0151856   .1993467
          3  |   .1210596   .0511398   .2603462
          4  |   .2344013    .144129   .3575909
          5  |   .4049667   .2710298   .5547246
          6  |   .6020462   .3688264   .7966118
          7  |   .7707955   .4519781     .93203
          8  |   .8820117   .5300384   .9802168
------------------------------------------
```

Liste de quelques fonctions à appliquer à transform_margins

Modèle logit : **invlogistic(@)**

Modèle probit : **normal(@)**

Modèle cloglog **: invcloglog(@)**

Modèles Poisson, Binomial-Négatif, Gamma avec lien log…: **exp(@)**

Utilitaires

Profile.do

Le programme profile.do permet d'exécuter des commandes à l'ouverture d'une session de Stata.
Le fichier profile.do, s'il n'existe pas, est à coller dans `C:\Users\user_name` (C=>utilisateurs=> nom_utilisateur). Les fonctionnalités proposées peuvent être désactivées en insérant des zones de commentaires ou tout simplement en les supprimant, ou ajoutée à un profile.do existant.
Le contenu du profile.do est paramétré pour fonctionner sur Windows. Pour l'utiliser sous Linux, il suffit de modifier les chemins d'accès.

Fonctionnalités proposées

- Affectation d'un répertoire par défaut. Voir « **Répertoire de fichiers temporaires ou « trash box » »** plus bas.

- Lecture des .ado dans le lecteur D. Conseillé à l'Ined en raison du système de sauvegarde. Au préalable il convient de coller le répertoire ado qui se trouve normalement sur la racine du lecteur C, dans la racine du lecteur D.

- désactivation du blocage du défilement de l'output.

- Activation d'un répertoire « temporaire » (identique au répertoire par défaut) : Voir **Répertoire de fichiers temporaires ou « trash box »** plus bas.

- Création d'un fichier log à chaque ouverture de session : voir ci-dessous **Création automatique d'un fichier log à l'ouverture d'une session**

- Paramétrage de la commande **rsource** : pour les personnes désirant exécuter un script R dans un .do. Si l'exécutable de R se trouve dans un emplacement différent que celui paramétré dans le profile.do proposé ici, il suffit de modifier le chemin d'accès.

Affecter plusieurs répertoires de travail en début de programme (cf libname avec Sas)

On peut en début de programme affecter plusieurs répertoires où se trouvent, par exemple, plusieurs bases de données. On pourra alors faire les manipulations courantes de fichiers enregistrés dans plusieurs répertoires
Exemple : la base b1.dta se trouve dans le répertoire X et la base b2.dta se trouve dans le répertoire Y.
Pour affecter les répertoires, on utilise des macros soit locale ou globale. En terme de syntaxe, la macro globale présente un avantage ($nom au lieu de `nom'), il faut néanmoins vérifier qu'un nom de macro global n'est pas déjà utilisé et, de préférence, supprimer la nouvelle macro en fin de programme les noms de macro affectés aux répertoires.

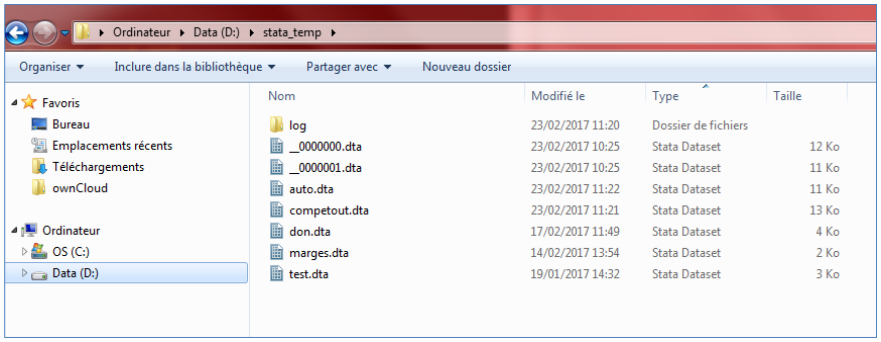
Exemple :

```
macro list

macro global X “path/X”
macro global Y “path/Y

use  $X/base1, clear
sort id
save $X/base1, replace
use  $Y/base1, clear
sort  id
merge id using $X/base1

macro drop X Y
```

Répertoire de fichiers « temporaires » ou « trash box »

Stata dispose d’un système d’enregistrement et d’ouverture des fichiers par défaut. A l’Ined il se situe dans la racine du lecteur D (pour vérifier l’emplacement du répertoire par défaut : exécuter **`pwd`**).

Dans un programme, une fois qu’un répertoire est affecté avec la commande **cd**, ce répertoire par défaut n’est plus identifié comme tel, celui affecté par cd prenant le relais. La solution proposée ici est de créer un répertoire nommé « stata_temp » dans D :/ et d’utiliser les commandes dédiées (**`tuse`**, **`tsave`**, **tdir** et **`terase`**) ou le la macro globale associée **$tmp**. Le répertoire sera reconnu grâce au profile.do (voir plus haut). Le nom de la macro associé est **tmp**.



Les commandes associées:

`tuse` nom_base : ouvre une base (identique à `use $tmp/nom_base`)

`tsave` nom_base : sauvegarde une base (identique à `save $tmp/nom_base`)

`tdir` nom_base : affiche le contenu du répertoire stata_temp. A l’ouverture d’une session Stata, le contenu du répertoire est affiché (propriété du profile, on peut supprimer cet affichage).

```
Librairie temporaire tmp chargee dans D:stata_temp
-----------------------------------
  Contenu du repertoire stata_temp
-----------------------------------
  <dir>   2/23/17 11:22  .
  <dir>   2/23/17 11:22  ..
  10.4k   2/23/17 11:22  auto.dta
  12.2k   2/23/17 11:21  competout.dta
   3.9k   2/17/17 11:49  don.dta
  <dir>   2/23/17 11:20  log
   1.4k   2/14/17 13:54  marges.dta
   2.8k   1/19/17 14:32  test.dta
  11.7k   2/23/17 10:25  __0000000.dta
  10.3k   2/23/17 10:25  __0000001.dta
r; t=0.02 11:23:36
```

terase [extension fichier]: efface le contenu du répertoire. **terase** seul supprime tous les fichiers Stata totalement ou par type de fichier (voir le fichier d'aide).

Exemples
terase
terase log

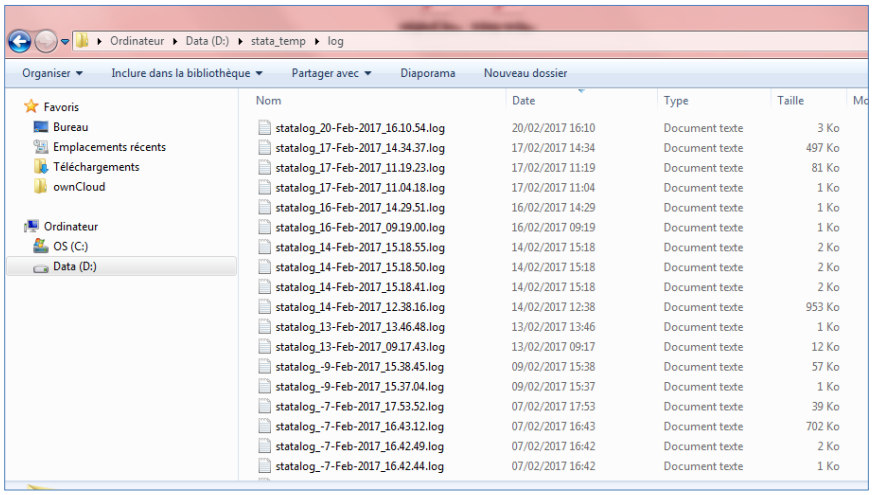
Pour apparier une ou plusieurs base présente dans ce répertoire, on utilisera le nom de la macro variable `$tmp/`, par exemple : **`merge id using $tmp/b1 $tmp/b2`**

Le nom de la macro peut être modifié dans le profile.do. Par exemple si on préfère que cela soit **t** au lieu de **tmp**, il suffit de modifier la ligne `global` **`tmp`** `"D:/stata_temp/"` par `global` **`t`** `"D:/stata_temp/"`.

Création automatique d'un fichier log à l'ouverture d'une session
Ici les fichiers log sont enregistrés dans le sous répertoire « log » du répertoire **stata_temp**. Il faut donc créer ce répertoire. Les log sont générés dans le profile.do fourni (voir ci-dessus).

```
running C:\Users\thevenin_m\profile.do ...
Creation d'un fichier log dans D:/stata_temp/log/
Nom du log: statalog_23-Feb-2017_11.23.36.log
```

Si on utilise la commande terase (ou terase log), le log d'une session active n'est pas supprimé.



Si l'on souhaite modifier l'emplacement des fichiers log, par exemple de `:D/Stata_temp/log` vers `:D/stata_log`, il suffit de créer un répertoire stata_log dans :D et de modifier la ligne :
`log using "D:/stata_temp/log/`statalogname'" , text name(statalog)`
par
`log using "D:/stata_log/`statalogname'" , text name(statalog)`