Les commandes Stata pour LaTeX

Florent Bresson *

16 juin 2005.

Version 1.2

Résumé

Avec ce document, j'ai tenté de recenser les commandes Stata permettant de simplifier le travail ultérieur de traitement de texte sous LaTeX. J'ai non seulement repris les fichiers d'aide accolés aux fichiers ado propres à chaque commande, mais aussi développé des exemples et apporté un certain nombre de conseil afin de vous faciliter la prise en main. Une fois ces commandes adoptés, vous n'aurez plus que marginalement besoin de recopier des statistiques manuellement. Autant de temps gagné pour se concentrer sur son travail de recherche.

*CERDI, Université d'Auvergne. Ce document n'est pas figé, n'hésitez donc pas à me contacter en cas d'erreur ou pour me faire bénéficier de vos lumières quant'à l'existence d'autres commandes que celles abordées dans ce document afin que je puisse l'enrichir. Contact: florent.bresson@u-clermont1.fr

Table des matières

Avant d'entamer la lecture de ce document, je tiens à préciser que celui-ci est l'oeuvre d'un néophyte en matière de LaTeX. Certaines de mes explications peuvent donc être entachées d'erreurs, notamment pour ce qui est des packages requis pour LaTeX.

Pour clarifier le document, les commandes Stata seront écrites écrites en sans serif et celles LaTeX en `type writer`. A chaque fois que celà nous a paru intéressant, nous avons inclus des exemples afin que vous puissiez juger la pertinence de chacune de ces commandes. La mise en forme des différents n'a en général pas fait l'objet de retouches, ce qui explique le manque d'homogénéités des différents tableaux présentés.

1 Se les procurer

Ces merveilleuses commandes sont disponibles à partir de Stata, pour peu que l'on dispose d'une connexion internet. Sous Stata, il suffit de lancer un *search, search net ressources* en précisant le nom de la commande désirée. On peut aussi retrouver une partie de ces commandes sur le site de Stata[1] ou au travers d'une recherche via Econpapers[2] ou le RePEc d'IDEAS[3].

2 Les commandes

2.1 Graph2tex

Cette commande permet de récupérer au format eps le dernier graphique réalisé sous Stata ainsi que le code destiné à son inclusion dans un document LaTeX. Le code est affiché dans la fenêtre de résultat de Stata et éventuellement dans un fichier log. Le fichier eps est quant à lui généré directement dans le répertoire Data (ou son équivalent). Après utilisation, il ne reste donc qu'à copier-coller le code dans son éditeur de texte

[1] `http://www.stata.com`
[2] `http://econpapers.hhs.se`
[3] `http://ideas.repec.org/s/boc/bocode.html`

préféré et de déplacer le fichier eps vers le répertoire contenant votre document LaTeX.

Afin que puissent être prises en compte les informations de taille générées par graph2tex, cette commande nécessite que soit activé le package `graphicx` (à ne pas confondre avec `graphics`) dans votre document tex. Précisons enfin que Ghostscript n'aime pas toujours les fichiers eps de Stata[4].

La syntaxe de la commande est la suivante :

graph2tex ,[epsfile(*fichier*) reset number caption(*nom*) label(*nom*)ht(*nombre*)]

L'option epsfile(*fichier*) permet de nommer à votre convenance le fichier eps. Sans celle-ci, le fichier porte par défaut le nom *defautgraphname.eps*.

L'option number est utile lorsque l'on souhaite convertir plusieurs graphiques, notamment au sein d'un fichier do. Elle permet en effet d'obtenir, à partir d'un nom générique défini via l'option epsfile(*fichier*), des fichiers qui porteront les noms *fichier1*.eps, *fichier2*.eps, *fichier3*.eps par exemple. Pour que la numérotation débute bien par 1, on ajoute l'option reset lors de la première utilisation de la commande graph2tex. Ensuite, à chaque utilisation de celle-ci, on se contente de rajouter l'option number. Pour ceux qui préfèrent raisonner directement à partir de globales, l'aide associée à la commande précise la marche à suivre.

Les trois dernières options caption(*nom*) label(*nom*) ht(*nombre*) interviennent dans le code LaTeX généré par Stata et permettent d'obtenir respectivement le titre, la référence et la taille du graphique dans le document. Précisons que la hauteur spécifiée dans l'option ht est un réel supérieur à 3 et qu'il s'agit d'une mesure en pouces (qui comme tout le monde le sait correspond à 2,54 cm).

Pour en finir avec cette commande, il convient de préciser que l'on peut parfaitement s'en passer. En effet, tout graphique généré sous Stata peut être exporté au format eps via la commande graph export *nom du fichier.eps* (diverses options sont disponibles, voir eps options sous Stata) ou par le biais d'un *Save Graph...* (disponible par un click droit sur la fenêtre graphique). Ensuite, il ne reste plus qu'à écrire les commandes LaTeXpour inclure le graphique dans le document[5].

[4] Personnellement, j'ai résolu ce problème en réalisant un copier-coller du graphique sous OpenOffice Draw, puis en exportant le document au format eps. Tout autre logiciel graphique doit à priori convenir pour réaliser l'opération.

[5] Au cas où Ghostscript refuserait de traiter correctement le fichier eps, voir la note 2.1.

2.2 Outtex

Autant il est relativement facile de se passer de graph2tex, autant outtex peut réellement passer pour indispensable. Cette commande[6] permet de transformer en code LaTeX les résultats d'une estimation de manière à obtenir un tableau déjà mis en forme (et sans avoir à réaliser une recopie fastidieuse des coefficients). Comme les exemples parlent souvent bien mieux que les mots, voici de quoi vous convertir rapidement[7] :

TABLEAU 1 – Ma belle régression

Variable	**Coefficient** (Ec. Type)
Variable de gauche n°1	-0.003**
	(0.001)
Variable de gauche n°2	-0.715*
	(0.310)
Constante	4.070**
	(0.102)
N	696
R^2	0.029
$F_{(2,693)}$	10.175

Niveau de significativité : † : 10% * : 5% ** : 1%

Avant d'envisager la syntaxe de la commande, il peut être utile de préciser que le tableau sera centré sur le document (utlisation de la commande `\centering`). La syntaxe de la commande est la suivante :

outtex [,below plain digits(*nombre*) level labels details legend nopar title(*titre*) key(*label*) longtable placement nocheck file(*nom du fichier*) append replace]

Commençons par plain et below. Ces options permettent d'agencer les différentes informations (coefficient, écart type et symboles de significativité) dans le tableau. En leur absence, les informations sont disposées de la manière suivante :

Variable	Coefficient	Symbole de significativité	Ecart-type
aligné à gauche	aligné à droite	aligné à gauche	centré

[6] La version testée date de Septembre 2001.

[7] Réalisé sans autre trucage que la traduction d'*intercept*, de *signifiance levels* et d'*écart type* en bon français.

L'option below place les écarts types en dessous des coefficients et les centre par rapport à ces derniers. On obtient donc :

Variable	Coefficient Ecart-type	Symbole de significativité
aligné à gauche	aligné à droite	aligné à gauche

L'option plain place les symboles de significativité dans la même colonne que les coefficients et centre celle-ci ainsi que celle des écarts types, soit :

Variable	Coefficient+Symbole de significativité	Ecart-type
aligné à gauche	centré	centré

Quand on a la bonne idée de combiner les deux, on obtient :

Variable	Coefficient+Symbole de significativité Ecart-type
aligné à gauche	centré

Avec digits, il est possible de préciser le nombre de chiffres affichés après la virgule pour les coefficients. Par défaut, la valeur est de trois.

L'option level permet d'activer l'inclusion des symboles de significativité qui ne sont pas générés par défaut. En lui associant legend, on affiche en bas du tableau la légende correspondant à ces symboles[8].

En ajoutant l'option labels, on demande à Stata de remplacer les codes des variables par leurs éventuelles étiquettes.

L'option details ajoute des lignes au tableau afin d'inclure un certain nombre de statistiques[9] comme le R^2.

[8] Si jamais vous n'aimez pas les symboles utilisés par outtex, il est toujours possible de modifier le code généré. Chaque symbole est précisé deux fois, une première pour le tableau, une seconde pour la légende. Si jamais vous désirez systématiser le changement, il suffit de modifier le fichier ado lignes 96 à 98.

[9] A nouveau, on peut modifier le fichier ado pour franciser le nom du log de vraissemblence ligne 86.

Avec nopar, on supprime les parenthèses autour des écarts-types. Précisons qu'il n'est malheureusement pas possible de remplacer les écarts types par des p-values ou des statistiques de Student[10].

L'option title(*nom*) permet, comme son nom l'indique, de nommer le tableau en insérant une commande `\caption` en tête du code, donc du tableau. En ajoutant key(*label*), on inclue dans la commande `\caption` une commande `\label` afin de référencer le tableau comme on le souhaite. Par défaut, l'étiquette attribuée est tabresult*commande de régression*[11]. Ainsi, après regress, l'étiquette par défaut sera tabresult regress. Attention donc si vous utilisez plusieurs fois outtex dans un même document, vous risquez de vous retrouver plusieurs fois avec la même étiquette pour des tableaux différents.

L'ajout de longtable permet d'avoir des tableaux dont la longueur dépasse celle de la feuille. Le tableau est ainsi coupé en autant de parties que nécessaires et des messages sont disposés de part et d'autre de la césure afin d'assurer la transition entre les parties[12]. Pour que l'option soit opérationnelle, le package `longtable` doit être ajouté dans le préambule du document LaTeX.

placement doit en principe permettre de spécifier le positionnement du tableau généré dans le document LaTeX mais aucun argument n'est associé à la commande. On en est donc réduit à la valeur par défaut `[htbp]` que l'on peut de toute manière modifier à loisir dans le code.

Si l'on inclue nocheck, on demande à Stata de ne pas traiter les caractères spéciaux réservés au codage LaTeX comme le « \ ». Il n'y a donc à priori aucune raison de l'utiliser.

Enfin, si l'on souhaite que le code généré par outtex apparaisse directement dans un fichier de type ASCII, il suffit d'ajouter file(*nom du fichier*). Si aucune extension n'est précisée pour le fichier de destination, celui-ci est créé par défaut avec l'extension tex. Si l'on ajoute encore l'option append, le code est mis à la suite du document, alors qu'avec replace, on remplacera un éventuel ancien document.

[10] Évidemment, on peut toujours modifier le fichier ado en conséquence, voire imaginer une option permettant de choisir la statistique désirée.

[11] Attention, l'aide de la commande contient une erreur puisqu'elle affirme que l'étiquette par défaut est simplement tabresult.

[12] Pour franciser les messages disposés, modifier les lignes 90 et 91 du fichier ado.

2.3 Sutex

Vous avez aimé réaliser de beaux tableaux en trois secondes avec outtex, alors vous allez adorer sutex qui réalise les mêmes opérations de base que summarize tout en produisant parallèlement le code LaTeX nécessaire à la production du tableau désiré[13]. Rappelons que summarize délivre pour les variables précisées un certain nombre de statistiques de bases comme la moyenne, l'écart type, le minimum... Avant d'appréhender la syntaxe de sutex, précisons que les options disponibles avec summarize ne le sont plus avec sutex. C'est ainsi le cas de details qui permet d'obtenir des statistiques supplémentaires comme le skewness et le kurtosis, tout comme de meanonly, format et separator(*nombre de lignes*). On retrouve par contre évidemment les préfixes et suffixes by, if, in et *pondération*. sutex s'écrit donc :

[by *variables* :] sutex *variables* [if] [in] [*pondération*] [, digits(*nombre*) labels
par nobs minmax na(*texte*) title(*titre*) key(*label*)
longtable placement nocheck file(*nom*) append replace]

Comme vous pouvez le constater, un certain nombre d'option sont similaires à celles de la commande outtex (cf. section 2.2), à savoir : digits, labels, title,key, longtable, placement, nocheck, file, append et replace. Nous ne reviendront donc pas dessus. Précisons tout de même que l'étiquette par défaut du tableau est *sumstat*. Lorsque le préfixe by est utilisé, sutex génère autant de tableaux que nécessaires. Un numéro est ajouté à chaque titre et chaque étiquette de tableau.

Il existe toutefois d'autres options comme nobs qui ajoute au tableau une colonne pour préciser le nombre d'observations pour chaque variables. Cette option est activée par défaut lorsque le nombre d'observations diffère d'une variable à l'autre.

L'option par permet de préciser si l'on souhaite que les écarts type figurent entre parenthèses.

En ajoutant minmax, le tableau généré comprendra deux nouvelles colonnes pour le minimum et le maximum.

Enfin, na(*texte*) permet d'indiquer à Stata quel texte afficher lorsque le calcul de la statistique n'est pas possible (comme le calcul d'un écart type sur une observation).

[13] La dernière version de la commande date de Septembre 2001.

Pour vous donner une idée du travail réalisé via sutex, voici un petit exemple[14].

TABLEAU 2 – Mes belles statistiques

Variable	Mean	Std. Dev.
Variable 1	3.828	2.253
Variable 2	59.591	89.836
Variable 3	0.239	0.426
N	696	

2.4 Latabstat

A l'instar de sutex, cette commande[15] nous permet de calculer et mettre rapidement en forme les résultats obtenus via tabstat. Rappelons que tabstat est un outil relativement substituable à summarize puisqu'il est aussi destiné au calcul d'un certain nombre de statistiques (en plus grand nombre tout de même). Notre objectif n'étant pas la description de tabstat, se reporter à [1] pour l'utilisation de celle-ci. Nous rappellerons toutefois celles qui peuvent avoir un intérêt du point de vue de la mise en forme.

Attention, cette commande génére parfois un code erroné. L'accolade fermante de la commande `\multicolumn` dans les dernières lignes du code n'est pas toujours présente après la note de fin de tableau. Pensez donc à effectuer la correction nécessaire.

La syntaxe de la commande est la suivante :

latabstat *variables* [*pondération* if in, statistics(*statistique*) by(*variable*) nototal missing nosep columns(variables|statistics) longstub labelwidth(*nombre*) varwidth(*nombre*) format[(*%fmt*)] casewise save tf(*fichier*) replace append tx(*taille*) caption(*titre*) clabel(*label*) hwidth(*nombre*)]

L'option tf(*fichier* permet d'enregistrer le code dans un fichier tex. Si l'on peut spécifier le chemin du fichier, il n'est pas nécessaire de préciser son extension. En ajoutant append, le code généré sera ajouté au fichier précisé alors qu'avec replace, on remplacera le code existant.

[14] A nouveau, nous avons francisé le résultat. La manipulation du fichier ado dans ce sens n'est guère difficile.

[15] La version présentée est la version 1.1 de Janvier 2003.

L'utilisation de l'option tx(*largeur*) nécessite l'inclusion préalable du package `tabularx` dans le document LaTeX pour être opérationnelle. Celui-ci permet de spécifier la largeur souhaitée du tableau. Celle-ci doit être précisée entre les parenthèses, l'unité de mesure utilisée étant le cm. En précisant 0, on indique à Stata que l'on souhaite que la largeur du tableau s'adapte à la longueur des lignes, ce qui est déjà le cas lorsque l'on n'active pas cette option (mais dans un environnement `tabular` classique).

De manière classique, caption(*titre*) (et non title) permet de donner un titre au tableau. Si cette option n'est pas spécifiée, aucun titre n'est attribué par défaut. Les commandes caption et label sont toutefois insérées dans le code.

L'option columns(variables/statistics) permet de modifier la présentation des données en définissant la dimension du tableau que l'on souhaite voir disposée en colonne. Par défaut, latabstat considère l'option columns(statistics). Il convient alors de préciser que la sortie du code dans la fenêtre de résultat de Stata peut introduire de petites coquilles dans le code généré lorsque la dimension colonne est trop longue. Dans ce cas de figure Stata réalise alors des renvois à la lignes signalés par des symboles > qu'il faut de supprimer un à un dans votre document tex.

Pour jouer sur le nombre de chiffres affichés, leur nombre après la virgule, le remplacement du point par une vigule... il est nécessaire d'utiliser l'option format(*% fmt*). Sans rentrer dans les détails (pour cela, consulter [2]), pour obtenir des points à la place des virgules et changer le format de l'ensemble des statistiques, (*% fmt*) doit prendre la forme suivante *%x,yz*. Le premier élément, x, permet de préciser le nombre maximal de chiffres à afficher. La virgule peut être substituée par un point si l'on a pour une quelconque raison décidé d'utiliser le format américain. Avec y, on indique le nombre de chiffres après la virgule. Le dernier élément, z, est une lettre qui précise véritablement le format de la statistique. Avec e, on aura une écriture scientifique ; avec f, on obtiendra des nombres comprenant exactement x chiffres ; enfin g réalise des arrondis après y chiffres, mais sans 0 pour les résulats inférieur à 1 en valeur absolue. Pour des valeurs importantes, on peut accoler un "c" pour avoir des séparateurs de milliers.

Enfin, un petit exemple pour admirer le travail réalisé[16] :

[16] Comme toujours le code à été francisé. Il est toujours possible de modifier le fichier ado, mais le travail nécessaire pour traduire en français les statistiques utilisées sera plus important. Une dernière correction a aussi été faite dans la note de bas de tableau. Celle-ci renvoie au fichier de données à partir duquel a été

TABLEAU 3 – Encore de belles statistiques

Statistique	variable 1	variable 2	variable 3
moyenne	.2385057	59.59052	3.827586
écart type	.4264764	89.83574	2.252835
skewness	1.227183	4.936407	.6012083
kurtosis	2.505979	40.29011	2.414294
médiane	0	31	4
N	696	696	696

*Source :*C :\DATA\dossier\fichier.dta

2.5 Latab

La commande latab se substitue à tabulate. Elle permet donc de réaliser des tableaux de fréquence pour une ou deux variables. Autant latabstat offrait toutes les options disponibles pour tabstat, autant latab se révèle relativement pauvre par rapport à son parent[17].

Pour pouvoir l'utiliser, il est nécessaire d'appliquer des étiquettes aux variables utilisées ainsi qu'à leurs valeurs. La syntaxe générale de la commande est la suivante :

latab *variable* [*pondération* if in, tf(*fichier*) replace append row col ncom dec(*nombre*) tx(*longueur*)]

Contrairement à tabout, cette commande n'accepte au maximum que deux variables. Elle produit dans la fenêtre de résultat de Stata un code LATEXque l'on peut directement insérer dans un document par copier-coller. Si l'on souhaite exporter l'output dans un document texte, il faut utiliser l'option tf qui permet de préciser le chemin et le nom du fichier accueillant le code. Si ce fichier existe déjà, il convient de ne pas oublier d'utiliser append ou replace.

Par défaut, latab effectue des dénombrements. Avec les options row et col, il est possible d'obtenir à la place des pourcentages en ligne ou en colonne. Tout comme avec tabout, il n'est pas possible d'obtenir des pourcentages croisés.

L'option dec permet de préciser le nombre de décimales que l'on souhaite voir affichées.

généré le tableau. Or le chemin d'accès est précisé avec des \ ; LATEX a donc tendance à le considérer comme une série de commandes, ce qui est souvent du plus mauvais effet. Pensez donc à corriger le code après l'avoir collé dans le document

[17] La version présentée date de Janvier 2002.

Enfin, on peut ajouter l'option tx permet de remplacer l'environnement tabular par un environnement tabularx. Entre parenthèses, on spécifie alors la largeur en cm du futur tableau.

Le tableau 4 permet d'apprécier les résultats obtenus via latab. On notera qu'un titre de la forme « *Variable 1* by *variable 2* » est automatiquement produit.

TABLEAU 4 – Variable 1 by variable 2

	Variable 2		
Variable 1	Catégorie 1	Catégorie 2	Total
Catégorie 1	7	11	18
Catégorie 2	10	13	23
Catégorie 3	6	3	9
Total	23	27	50

Source :

2.6 Tabout

Cette commande[18] se substitue à la fois à tabulate et table. Elle permet donc à la fois d'obtenir des fréquences pour une ou plusieurs variables et de faire apparaître un certain nombre de statistiques pour chacune d'elles. Avant de nous engager plus loin sur son utilisation, précisons que tabout est programmé pour la version 8.2 de Stata[19]. La syntaxe de la commande est donc :

tabout *variables* [if] [in] [weight] using *fichier* [, replace append cells(*contenu*) oneway format(*format*) stats(*statistiques*) n(*position* | *forme* | *pondération*) style(*style*) lines(*lstyle*) font(*fstyle*) bt noborder total(*total* | *total*) header1(*nom*) header2(*nom*) header3(*nom*) noheader lefthead body topf(*fichier*) botf(*fichier*) topstr(*nom*) botstr(*nom*) psymbol(*texte*) delimit(*délimiteur*) show(display) labelwidth(*largeur*) colwidth(*largeur*) tablewidth(*largeur*)]

Comme on peut s'en apercevoir, les options de cette commande sont particulièrement nombreuses. On peut toutefois se faire plaisir avec peu d'options. Ajoutons tout de même

[18] Pour ce document c'est la version 1.2.0 qui a été testée.

[19] Pensez donc à mettre à jour votre version de Stata. Si vous ne pouvez le faire, il est toujours possible de modifier légèrement le fichier ado en modifiant le numéro de version de la commande.

que pour employer tabout, il faut avoir auparavant étiqueté ses variables et les valeurs prises. De fait, elle ne présente d'intérêt que pour le traitement de variables catégorielles.

Les résultats des calculs effectués via tabout sont présentés dans la fenêtre de résultats et enregistrés en parallèle dans un fichier texte dont le contenu dépend des options retenues. Il faut donc avant tout donner un nom au futur fichier après using. Ensuite, pour définir le format de l'output, il est nécessaire d'utiliser style(*style*). Entre parenthèses, on indique tab si l'on désire un fichier texte avec une mise en page par tabulation, tex si l'on désire un tableau au format LaTeX ou html si l'on souhaite obtenir un tableau en html.

2.6.1 Le contenu

Mettons entre parenthèses les questions de style et passons les traditionnelles options replace et append arriver à des questions plus essentielles de contenu. La première option à préciser est oneway. Si l'on inclue cette option à tabout, les informations demandées pour les différentes variables retenues ne sont pas croisées. Chaque variable est donc traitée indépendamment et apparaît en colonne dans le tableau. C'est l'option par défaut lorsque notre liste de variables n'en contient qu'une. Pour faire un tableau à double entrée et obtenir des informations sur le croisement de plusieurs variables, il ne faut pas préciser oneway, mais il convient de faire attention lorsque l'on souhaite croiser plus de deux variables. En effet, tabout ne réalise de croisement que par rapport à la dernière variable incluse dans la ligne de commande. Pour croiser plusieurs variables simultanément et puisque tabout n'accepte pas le préfixe by, il faut ruser et utiliser plusieurs fois la commande en jouant sur le suffixe if. Pour schématiser le fonctionnement de la commande, nous avons schématisé les tableaux obtenus avec et sans oneway dans les tableaux 5 et 6.

L'option cells(*contenu*) peut apparaître comme relativement exotique car permettant d'effectuer plusieurs types de réglages suivant le type de tableau utilisé. Dans un premier temps, on peut utiliser fcount, fper et/ou fcum afin de récupérer respectivement des nombres, des pourcentages ou des fréquences cumulées lorsque l'on cherche à obtenir un tableau de fréquence. Ces options ne sont valides que lorsque oneway est précisé. Pour des tableaux à double entrée, on doit choisir entre row, col, count et double. Les trois

TABLEAU 5 – Utilisation de tabout sans oneway pour N variables

	Variable N			
	occurence 1	$\cdots$	occurence k	Total
	Info	$\cdots$	Info	Info
Variable 1				
occurence 1				
$\vdots$				
occurence l				
Total				
$\vdots$				
Variable $N-1$				
occurence 1				
$\vdots$				
occurence m				
Total				

TABLEAU 6 – Utilisation de tabout avec oneway pour N variables

	Info 1	$\cdots$	Info i
Variable 1			
occurence 1			
$\vdots$			
occurence l			
Total			
$\vdots$			
Variable N			
occurence 1			
$\vdots$			
occurence k			
Total			

premiers permettent d'obtenir des fréquences en ligne, en colonne ou un dénombrement. Avec double, on obtient à la fois les pourcentages en colonne et le nombre d'observations. On peut regretter qu'il ne soit pas possible d'afficher simultanément différents types de fréquences ni d'obtenir des pourcentages par rapport au total. Enfin, il est possible de demander l'affichage d'un certain nombre de statistiques (une seule lorsque l'option oneway n'est pas activée). On peut alors préciser : N, count, mean, median, var, sd, skewness, kurtosis, sum, min, max, p1, p5, p10, p25, p50, p75, p90, p95, p99, iqr, r9010, r9050, r7525 ou encore r1050 après lequel on précise le nom de la variable dont on souhaite obtenir l'information demandée. La ou les variables en question peuvent très bien ne pas être celles que l'on a précisées au début de notre commande. On écrira par exemple :

tabout *variable1 variable2* ..., cells(N *variable3* mean *variable4*)...

Attention, les résultats sont par défauts arrondis à l'unité. Il est donc conseillé d'inclure certaines options de format.

On peut ajouter une colonne pour afficher le nombre d'observations via l'option n. Celle ci nécessite que soient précisées trois options séparées par des « | » et correspondant respectivement à la position de l'information, sa forme et la pondération utilisée. Pour la position, il faut choisir entre col si l'on souhaite ajouter une colonne supplémentaire pour afficher les nombres d'observation, ou lab si l'on souhaite que l'information soit contenue dans la première colonne. L'effet de l'option de forme dépend du choix effectué juste auparavant. Avec col, cet espace permet de remplacer l'entête de la colonne (un « n » par défaut) par le texte de son choix. Avec lab, on précise la texte qui accompagnera les observations dans chaque cellule (par défaut c'est (n = #) où # désigne le nombre à indiquer). On peut ainsi écrire « # observations ». Pour obtenir la forme par défaut, il faut préciser d. Enfin, la pondération permet de présenter une information différente du nombre d'observations, comme par exemple le nombre d'individus dans la population mère. Pour celà il suffit de préciser le nom de la variable porteuse de l'information. Si l'on souhaite simplement le nombre d'observations de nos variables, on écrit d.

Si l'on ne souhaite pas faire apparaître les totaux pour certaines variables, il faut ajouter l'option nototals après laquelle on précise les noms des variables concernées. Attention, cette option produit quelques fois des codes erronés (lignes manquantes).

Enfin, lorsque l'on réalise un tableau croisé simple (deux variables seulement et sans l'option oneway), on peut obtenir les résultats d'un test du χ^2 en ajoutant[20] stats(chi2).

2.6.2 La forme et l'affichage

Du point de vue de la forme de l'output, tabout se révèle très riche et permet de limiter grandement les retouches du code (au prix certes d'une certaine complexité des lignes de commande). Pour l'option format, je vous renvoie aux explications données en p. 10

Avec lines, on précise le type de filets horizontaux que l'on souhaite obtenir. Par défaut, il s'agit de filet simple (option single. On obtient des filets doubles avec double et

[20] L'auteur indique dans le fichier d'aide qu'il souhaite permettre ultérieurement l'affichage d'autres tests.

on les retire avec none. Si l'on ajoute l'option bt dans la ligne de commande, on ne voit apparaître que les filets supérieurs et inférieurs du tableaux. Les filets intérieurs sont alors remplacés par des espacements (utilisation de la commande \midrule). L'option noborder n'est utile que pour un output en html.

L'option font permet de passer en gras (bold) ou en italique[21] (italic) le nom des différentes variables présentées dans le tableau.

Si l'on souhaite modifier le texte correspondant aux entêtes des lignes et colonnes correspondant aux totaux, on utilise total. Le premier champs correspond aux totaux en colonne et le second aux totaux en ligne. On sépare les deux par un « | ». Les autres entêtes de colonne peuvent être modifiés à loisir via header1, header2 et header3. Chacune correspond à un niveau différent d'entête. Lorsque l'option oneway n'est pas activée, le 1erentête reprend par défaut le nom de la variable de colonne et le centre via l'utilisation de \multicolumn. Si l'on utilise header1, les cellules du tableau ne sont plus fusionnées[22]. Dans certains cas, principalement pour header2[23] on souhaitera écrire des entêtes pour plusieurs colonnes. On utilise alors le « | » pour effectuer le changement de colonne de la même manière qu'on utilise le & sous LaTeXdans un environnement tabular. Lorsqu'une colonne doit rester vierge, on utilise un underscore. A l'inverse, si l'on ne souhaite pas obtenir d'entête, il suffit d'utiliser l'option noheader.

Par défaut, tabout produit un code minimaliste. Pour un output LaTeX, on ne disposera ainsi que du contenu des cellules. Afin d'enchâsser[24] celui-ci dans un environnement tabular et d'obtenir un document prêt à compiler, il suffit d'ajouter l'option body. Pour modifier le début et la fin du code, deux méthodes sont offertes. La première consiste à faire appel à des fichiers contenant le code de son choix. On utilise alors topf pour le début et botf pour le bas du tableau, en précisant à chaque fois le chemin des fichiers utilisés. La seconde méthode consiste à utiliser topstr et botstr pour écrire directement le code que l'on souhaite voir apparaître.

Avec show, il est possible de choisir ce que l'on souhaite voir apparaître dans la

[21] Du point de vue de LaTeX, l'utilisation de italic se traduira par l'utilisation de la commande \emph et non \textsl.

[22] A moins d'écrire l'option de cette manière : header1(\multicolumn{*x*}{*y*}{*entête*}).

[23] Par défaut, on retrouve les différentes étiquettes de valeur lorsque oneway n'est pas précisé. Avec oneway et lorsque l'on demande des statistiques relatives à certaines variables, ce sont les noms de celles-ci qui figurent en entête.

[24] ☺.

fenêtre de résultat de Stata. Par défaut, il s'agit d'un tableau classique de résultat. On dispose alors des sous-options all, code et none pour voir s'afficher à la place le tableau et le code, le code seulement ou rien du tout.

Les options labwidth, colwidth et tabwidth permettent de modifier l'apparence du tableau affiché dans la fenêtre de résultat de Stata. Tout aussi mineure est sans doute l'option delimit qui permet de remplacer le « | » utilisé pour certaines options par le caractère de son choix[25].

Pour finir, les figures 7 et 8 donnent un aperçu de ce que l'on peut obtenir[26] via tabout.

Tableau 7 – Encore un jolie tableau

	Variable 3							
	Catégorie 1		Catégorie 2		Catégorie 3		Total	
	Num	%	Num	%	Num	%	Num	%
Variable 1								
Catégorie 1	5	41.666667	6	33.333333	7	35	18	36
Catégorie 2	5	41.666667	8	44.444444	10	50	23	46
Catégorie 3	2	16.666667	4	22.222222	3	15	9	18
Total	12	100	18	100	20	100	50	100
Variable 2								
Catégorie 1	7	58.333333	7	38.888889	9	45	23	46
Catégorie 2	5	41.666667	11	61.111111	11	55	27	54
Total	12	100	18	100	20	100	50	100

2.7 Est2vec, est2tex et est2one

Dans la sous-section 2.2, nous avons vu comment récupérer une régression pour la mettre en forme sous LaTeX. Il est toutefois rare en économie de présenter une seule estimation. On préfère de loin présenter un ensemble de régressions dans un même tableau. Traditionnellement, on utilise outreg pour récupérer ce genre de tableau, mais cette commande ne permet pas une mise en forme directe pour LaTeX. Nous présentons donc est2tex qui permet de concilier toutes ces exigences[27].

[25] Je n'ai pas traité les options lefhead et psymbol car n'ayant pu comprendre ou voir leur effet et intérêt.
[26] Les légères modifications opérées sur les tableaux 7 et 8 concernent le « é » que Stata ne gère pas correctement et les colonnes de données qui ont été centrées.
[27] Nos commentaires concernent la version de Mars 2005 de ces commandes.

TABLEAU 8 – Un autre jolie tableau

	Variable 4		
	Observations	Moyenne	Médiane
Variable 1			
Catégorie 1	18	153	55
Catégorie 2	23	143	56
Catégorie 3	9	126	64
Total	50	144	56
Variable 3			
Cat.gorie 1	12	67	56
Cat.gorie 2	18	157	49
Cat.gorie 3	20	178	66
Total	50	144	56

Avant de se plonger dans la syntaxe et les différentes options disponibles, précisons rapidement la philosophie de la commande. Tout commence bien évidemment avec les régressions[28] que l'on souhaite incorporer au futur tableau. Après chacune d'elles on utilise la commande est2vec afin de récupérer les différentes informations nécessaires pour les insérer dans des matrices propres à chaque type d'information (coefficients, écarts type ...)[29]. Ceci fait, on utilise est2tex pour générer un fichier tex dans lequel on trouve le code LaTeX du tableau et que l'on peut insérer directement dans un article via la commande `\input(`*`fichier`*`)`. La commande est2one est sans doute moins intéressante puisqu'elle permet juste de rassembler dans une même matrice les différentes matrices générées par est2vec.

2.7.1 Le contenu : Est2vec

Pour est2vec, la syntaxe est la suivante :

est2vec *tableau* [, replace vars(*variables*) shvars(*variables*) e(*macros*) r(*macros*) addto(*ancienne table*) raddto(ancienne table) name(*nom*) force nokeep]

[28] A priori, est2vec et est2tex fonctionnent avec toutes les commandes d'estimation disponibles sur Stata. Lors des quelques essais réalisés pour la rédaction de cet article, nous n'avons pas connu de refus. N'hésitez pas à me signaler d'éventuelles commandes récalcitrantes

[29] Pour être précis on génère les matrices suivantes : *tableau*_b qui comprend les coefficients, *tableau*_se pour les écarts-type, *tableau*_e pour les statistiques d'estimation et éventuellement *tableau*_r pour les résultats des macros r()

L'option vars(*variables*) ne s'utilise que pour la récupération des premiers résultats. Elle permet de préciser quelles variables on souhaite faire apparaître dans notre tableau. Ceci peut permettre de se concentrer uniquement sur nos variables d'intérêt par exemple. Mais cette option se révèle bien plus importante car la première utilisation que l'on fait d'est2vec pose les limites du tableau généré par est2tex. Ainsi, si l'on souhaite insérer par la suite les résultats d'une régression comprenant une variable qui n'était pas précisée lors de la première utilisation d'est2vec, les informations relatives à cette variable ne seront pas incluse dans le futur tableau. Or est2vec retient par défaut les seules variables incluses dans la dernière régression, constante comprise. Lors de la première utilisation d'est2vec, il faut donc bien penser à utiliser l'option vars(*variables*) avec les noms de toutes les variables qui devront par la suite figurer dans le tableau (attention à ne pas oublier la constante). D'un point de vue purement technique, les informations relatives à ces variables non utilisées dans les régressions sont représentées dans les différentes matrices utilisées par est2vec par "-999". Lorsque l'on génère par la suite le tableau, ces informations sont remplacées par des blancs (ou des points en recourant à l'option dots). Ajoutons enfin que cette option ne doit être utilisée que lors de la récupération des premiers résultats et que l'ordre dans lequel sont précisées les variable compte puisqu'il définit l'ordre d'apparition des variables dans le tableau[30]. L'ordre des variables dans les commandes de régression ne pose par contre pas problème, Stata réalisant l'appariement des données.

Une alternative à vars est l'option shvars qui permet d'utiliser les raccourcis usuels pour les noms des variables. Problème, cette option ne fonctionne pas avec les commandes de régressions multiples comme reg3 ou mlogit. De plus, elle n'accepte pas les variables non existentes dans la base. Pour le reste, il ne semble pas qu'il y ait de différence avec vars.

L'option e(*macro*) permet de préciser les statistiques de régression que l'on souhaite voir apparaître dans le tableau. Par défaut, on ne recupère que le nombre d'observations (la macro e(N))). Pour compléter, on peut donc ajouter par exemple, pour peu que la

[30] L'ordre n'est de toute manière pas un problème puiqu'il est très facile de le modifier par la suite sous LaTeX.

commande d'estimation utilisée le permette[31], dans e() :

- r2 pour le R^2 ;
- r2_p pour le *pseudo* R^2 de McFadden ;
- F pour la statistique de Fisher ;
- chi2 pour la statistique du χ^2 ;
- ll pour le log de vraissemblance ;
- ll_r pour le ratio du log de vraissemblance ;
- pbar pour le pourcentage de prédictions correctes.

Si l'information demandée n'est pas disponible, un -999 apparaît dans la matrice *tableau_e* à la place[32].

L'option r(*macros*) permet d'inclure dans le tableau final les résultats de macros de type r(). Son fonctionnement est identique à l'option e().

En ajoutant replace, on demande à Stata de remplacer le contenu des matrices générées par est2vec par les résultats de la dernière estimation. Si l'on utilise par contre addto(*tableau*), les résultats de la dernière régression seront accolés aux autres résultats dans chaque matrice gérée par est2vec. Il convient donc de l'utiliser dès la deuxième estimation que l'on souhaite inclure dans le tableau. N'oubliez pas que seuls les coefficients des variables déjà incluses dans la liste de variable de la première utilisation d'est2vec sont repris.

Avec raddto(*macros*), on ajoute une colonne à la matrice *tableau_r* pour ajouter les résultats de macros de type r(). A nouveau, seuls les résultats des macros déjà présentes sont ajoutés.

L'option name(*nom*) est particulièrement importante car elle permet d'identifier chaque estimation dans les différentes matrices utilisées par est2vec. Par défaut chaque colonne de matrice prend pour nom celui de la variable de gauche. Si l'on récupère toujours les résultats de régressions d'une même variable, nos colonnes porteront toutes le même nom, ce que n'aime pas est2vec qui nous sanctionne d'un message d'erreur. Prenez donc le réflexe de nommer dès le début chacune de vos régressions (un chiffre

[31] Pour avoir la liste complète des macros disponibles pour chaque commande, se référer aux manuels de références de Stata ou utiliser la commande ereturn list après votre commande d'estimation.

[32] Il semble que certaines macros ne soient pas acceptées. Ainsi, avec e(cmd), je ne suis pas parvenu à récupérer la commande de régression utilisée

suffit)[33].

J'ai précisé que l'on ne pouvait pas inclure dans les matrices de résultats les résultats de macros ou de variables n'étant pas déjà présentes auparavant. Ce n'est pas tout à fait vrai grace à l'option force qui contraint Stata à ajouter dans les matrices les informations demandées ligne par ligne, dans l'ordre où les variables et/ou macros apparaissent dans la commande. On demande donc à Stata d'introduire les nouvelles informations sans respecter leur appariement avec les entêtes de ligne. Le coefficient de la variable x peut donc se retrouver en face de la variable z.

Enfin l'option nokeep permet d'effacer les vecteur de résultats relatifs à la dernière estimation après leur insertion dans *tableau*.

2.7.2 La forme : Est2tex

Cette commande est celle qui permet de générer le code LaTeX du tableau de régression tant désiré. En l'utilisant, on créé à la fois un fichier tex et un fichier dta[34]. Comme vous allez vous en apercevoir par vous même son utilisation est moins délicate qu'est2vec, l'essentiel des options correspondant à des options de mise en forme du tableau. La syntaxe de cette commande est la suivante :

est2tex *tableau* [, replace droppall preserve path(*chemin*) multse(*nombres*)
mark(*option*) levels(*niveaux de significativité*) marktbl
precision(*nombre réel*) digits(*nombre*) flexible(*nombre*) fancy
horizontal cut suppress label collabels(*noms*) extracols(*numéros*)
dots plain(*séparateur*) ready leadzero thousep]

Avec replace, on remplace les éventuels fichiers *tableau*.tex et *tableau*_tbl.dta existant auparavant.

Les options droppall et preserve doivent l'une ou l'autre être précisées. La première efface l'ensemble des données en mémoire, contrairement à la seconde. Avant d'opter pour dropall, réfléchissez si vous souhaitez ou non continuer à travailler sur vos données. Par contre cela permet de libérer de la mémoire vive.

[33] Pour le reste, si ce nom ne vous convient pas dans le tableau, vous pouvez de toute manière modifier le code généré par est2tex pour obtenir les entêtes de colonne qui vous conviennent.

[34] Ce dernier n'est pas sans intérêt car il permet de récupérer toutes les informations du tableau pour la création d'un tableau de résultat sous un tableur ou un autre logiciel de traitement de texte par simple copier-coller .

L'option path(*chemin*) donne, comme son nom l'indique, la possibilité de spécifier le dossier dans lequel Stata doit créer ses fichiers d'output. Par défaut, il utilise le dossier en cours (en général C :Data).

L'utilisation de multse(*nombres*) sera sans doute rare. Elle permet d'inclure plusieurs séries d'écarts type pour chaque estimation. On peut ainsi imaginer faire figurer des écart type non corrigés puis avec une correction de White en dessous de chaque coefficient. Pour utiliser cette option, il faut avoir généré plusieurs séries d'écart type[35] dans les matrices correspondantes *tableau*_se1, *tableau*_se2... Lorsqu'on utilise multse(*nombres*), on précise entre les parenthèses les numéros des séries que l'on souhaite faire figurer dans le tableau. L'ordre dans lesquelles elles sont indiquée conditionne l'ordre d'apparition dans le tableau.

Avec mark(*option*), on commence avec les options de formatage du tableau. Celle-ci permet de spécifier comment l'on souhaite mettre en évidence la significativité de nos variables. Trois variantes sont alors disponibles suivant que l'on utilise les options starse, starb ou it. La première permet s'obtenir des étoiles de significativité à côté des écarts type alors qu'avec la seconde, celles-ci seront accolées aux coefficients. Si l'on opte pour la troisième, les coefficients non significatifs passent en italique. Si on utilise simultanément l'option multse, il faut préciser dans mark, après l'option, le numéro de la série d'écart type à partir de laquelle établir la significativité

Pour ce qui est de la significativité, on dispose aussi de levels(*niveaux*) qui permet de préciser pour quels degrés de confiance on choisit d'établir la significativité. A l'intérieur des parenthèses, trois réels doivent être précisés afin de désigner les différents intervalles de confiance retenus. De manière classique on peut par exemple indiquer levels(90 95 99). Par défaut, les valeurs suivantes sont retenues : 95, 99 et 99,9. Visuellement, le résultat dépendra de l'option retenue pour mark. Avec mark(*it*), Stata passera en italique les coefficients non significatifs par rapport au seuil le plus faible. Avec mark(*starse*) et mark(*starb*), on aura des étoiles[36] dont le nombre augmente avec le degré de significativité. Si le même intervalle de confiance est spécifié deux fois entre les parenthèses, le nombre d'étoiles le moins important sera utilisé.

[35] J'avoue à priori ne pas trop savoir comment faire.

[36] La commande est2tex ne vous propose que des étoiles comme symbole, mais il est relativement facile de modifier les symboles retenus sous LaTeX si les étoiles ne vous conviennent pas

Enfin, l'option marktbl permet de conserver dans une matrice t_stars les informations relatives à la significativité des variables.

Trois options sont disponibles pour la gestion des chiffres. Les options precision(*réel*) et digits(*nombre*) permettent de fixer le nombre de chiffres après la virgule auquel Stata doit réaliser les arrondis dans le tableau. Ainsi avec precision(*.01*) et digits(*2*), nos informations apparaissent arrondies au centième dans le tableau. Par défaut, les valeurs sont respectivement 0,001 et 3. L'option flexible(*nombre*) est plus souple car permet de ne pas contraindre les statistiques dont les coefficients sont très proches de zéro, ce qui évite de faire apparaître un coefficient nul dans le tableau. A nouveau, on précise entre parenthèses le nombre de chiffre après la virgule pour lequel on souhaite faire l'arrondi. Par défaut, la valeur est 2.

Il faut savoir que les tableaux générés par est2tex sont dépourvus de lignes de bordure. Pour obtenir une sortie plus conforme à l'usage et surtout plus agréable à regarder, on ajoute l'option fancy. Cette option est nécessaire pour utiliser les options leadzero et thousep. La première ajoute des zéros avant la virgule aux coefficients inférieurs à l'unité en valeur absolue tandis que la seconde insère des virgules comme séparateur de milliers.

L'option horizontal permet de transposer le tableau de manière à afficher les régressions en ligne.

Si l'on décide de ne pas inclure dans le tableau les résultats des macros e() et r(), il suffit d'ajouter l'option cut. Avec suppress, ce seront les écarts types qui ne figureront pas dans le tableau.

Avec l'option label, les noms des variables retenues pour le tableau sont remplacés par leur étiquette[37]. De la même manière, on peut remplacer les noms attribués lors de la récupération des données par le nom que l'on souhaite avec collabels(*noms*). Cette commande ne sert pas à numéroter les régressions puisqu'est2tex le fait déjà automatiquement. Elle peut par contre permettre de préciser la variable estimée en tête de chaque colonne. Entre parenthèses, on précise donc les noms à donner à chaque entête de colonne séparés par des espaces. Lorsque l'entête doit comporter des espaces, on remplace ceux-ci par des ~. Attention, si on ne précise par un exemple qu'un nom alors que

[37] Précisons tout de même l'existence de la commande est2rowlbl qui permet aussi de gérer les étiquettes. Son intérêt semblant relativement limité, je renvoie au fichier d'aide de la commande les intéressés.

notre tableau comporte deux équations par exemple, seul le premier entête de colonne sera remplacé. L'entête de la seconde colonne restera donc celui attribué sous est2vec.

L'option extracols(*numéros*) permet d'ajouter des colonnes vides dans notre tableau. Entre parenthèses, il suffit de préciser le numéro de la ou des colonnes après lesquelles on souhaite que ces espaces soient insérés. Il faut préciser que la colonne relative au nom des variables ne compte pas. Le numéro 1 désigne donc la colonne de la première régression.

Si l'on souhaite remplace les blancs du tableau par des points, on peut utiliser l'option dots.

La commande est2tex sert normalement à obtenir un tableau au format LATEX. On peut toutefois avoir envie d'obtenir ce même tableau au format ascii, chose possible grâce à l'option plain. Avec plain(*tab*), un fichier txt est créé avec des tabulations comme séparateur, alors qu'avec plain(*csv*) ce sera un fichier csv avec des virgules comme séparateur.

Enfin, l'option ready n'est utilisable que si l'on a utilisé au préalable est2one, commande qui ne sera pas décrite dans les lignes suivantes, son utilité ne nous paraissant pas flagrante [38].

Pour finir un exemple de ce qu'on peut obtenir avec est2tex[39] :

TABLEAU 9 – Un tableau avec est2tex

	Variable 1	Variable 1
	(1)	(2)
Variable 2	.004***	.006***
	(.001)	(.002)
Variable 3	-.228***	-.223***
	(.031)	(.034)
Variable 4		-.062***
		(.009)
Constante	-.217*	.506***
	(.128)	(.171)
Observations	696	696
Pseudo R^2	.168	.263
Log de vraisemblance	-318.219	-281.756
χ^2 statistique	67.225	76.81

[38] Pour ceux qu'elle intéresse tout de même, se reporter au fichier d'aide d'est2tex.

[39] A nouveau, nous avons francisé certaines expression. De plus, il convient de préciser à ceux qui souhaiteront consulter le code source de ce fichier que nous avons inséré le code du tableau dans un environnement `table`.

Comme on peut le voir dans le tableau 9, le tableau généré est non centré et étendu de telle manière qu'il lui arrive parfois de dépasser le cadre de la feuille. Ceci est lié à l'utilisation de l'environnement `tabular*` avec l'option `\textwidth` au lieu du classique `tabular` lorsque l'on utilise l'option `fancy`. Avec quelques menues corrections,on peut obtenir quelque chose de mieux senti comme le tableau 10.

TABLEAU 10 – Mon beau tableau de régressions

	Variable 1	
Variables	(1)	(2)
Variable 2	0,004***	0,006***
	(0,001)	(0,002)
Variable 3	-0,228***	-0,223***
	(0,031)	(0,034)
Variable 4		-0,062***
		(0,009)
Constante	-0,217*	0,506***
	(0,128)	(0,171)
Observations	696	696
Pseudo R^2	0,168	0,263
Log de vraisemblance	-318,219	-281,756
χ^2 statistique	67,225	76,81

Note :Niveaux de significativité : * :10% ** :5% *** :1%

2.8 Maketex

Cette commande[40] permet de convertir au format LaTeX un fichier log. Il récupère donc l'ensemble de ce fichier et l'insère dans un fichier tex comportant un préambule aussi complet qu'on le souhaite, et notre texte inséré dans l'environnement document. Si la commande peut paraître séduisante, son grand problème est de ne pas reprendre la mise en forme du texte tel qu'il est présenté dans le log. On se retrouve donc avec du texte au kilomètre qu'il faut réorganiser soit même. On peut toutefois envisager de l'utiliser après un outreg [41]. La syntaxe est la suivante :

maketex using *fichier* [, class(*classe)* classoptions(*options*) packages(*packages*)
title(*titre*) author(*auteur*) date(*date*)

[40] La version présentée date d'Août 2001.
[41] Enfin, si on a quand même décidé de ne pas utiliser est2tex (voir section 2.7).

Pour utiliser maketex, précisons qu'il est nécesaire de spécifier le nom complet du fichier, chemin et extension comprise. Comme maketex aime rappeler dans le fichier tex le chemin d'accès du fichier, LaTeX risque de vous signaler un paquet de messages d'erreurs liés à la présence des "\" dans le chemin. Pensez donc à effectuer les corrections nécessaires si vous souhaitez pouvoir compiler tranquillement votre document. Pour le reste, la syntaxe parle d'elle même, les différentes options permettant de spécifier dans l'ordre la classe du document, les options de classe, les packages et les références à utiliser pour le titre du document. Par défaut, la classe retenue est article avec ses options par défaut. Aucun package n'est prévu par défaut. Pour activer plusieurs packages, il suffit de les séparer par un espace. Pour préciser des options de package, on insère celles-ci devant le nom du package en ajoutant « : ». Si l'on doit préciser plusieurs options, il suffit de séparer celles-ci par des virgules. On écrira ainsi :

packages(latin1 :inputenc francais,english :babel)

pour obtenir :

```
\usepackage[latin1]{inputenc}
\usepackage[francais,english]{babel}
```

Pour finir, ajoutons que les concepteurs ont eu la bonne idée de faire afficher par Stata un lien vers le fichier généré après l'utilisation de la commande maketex. Il ne suffit plus alors que de cliquer sur le nom dans la fenêtre de résultat pour faire apparaître rapidement le code LaTeX.

2.9 Sjlatex et sjlog

2.9.1 Sjlatex

Si jamais vous veniez à ne plus apprécier le style courant de LaTeX et préfériez un style plus brut, la commande[42] sjlatex vous permettra de récupérer à partir de Stata[43] les fichiers nécessaires à l'écriture au format SJ. Avant de s'aventurer plus loin, un petit exemple pour apprécier les résultats du style SJ semble tout indiqué :

[42] Nous présentons ici la version 1.2.

[43] Pour récupérer la commande comme le package, suivre la procédure décrite à l'adresse suivante www.ats.ucla.edu .

```
              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
id              float  %9.0g
female          float  %9.0g       fl
race            float  %12.0g      rl
ses             float  %9.0g       sl
schtyp          float  %9.0g       scl        type of school
prog            float  %9.0g       sel        type of program
read            float  %9.0g                  reading score
write           float  %9.0g                  writing score
math            float  %9.0g                  math score
science         float  %9.0g                  science score
socst           float  %9.0g                  social studies score
------------------------------------------------------------------------------
```

Le style est tout à fait grossier, mais il est vrai qu'il peut être utile pour réaliser une présentation de Stata ou indiquer les lignes de commandes Stata nécessaires à la réalisation d'une estimation quelconque par exemple. Précisons que pour les fans, le package téléchargé comprend normalement une classe de document SJ basé sur la classe book.

2.9.2 Sjlog

Avec sjlog[44], on ne fait rien de moins que gérer un fichier log destiné à être inséré directement dans un document TEXau format SJ. Tout comme pour log, il existe différentes sous-commandes à sjlog, dont la syntaxe est la suivante :

sjlog using *fichier*	[, append replace]
sjlog close	[, noclean nologfile replace book]
sjlog do *fichier*	[, clear replace book nostop saving(*nom*)]
sjlog clean *fichier*	[, log logclose sjlog sjlogdo]
sjlog type *fichier*	[, replace find path(*chemin*) logfile smclfile]
sjlog basename *fichier*	[, display]

Avec sjlog using/close, on ouvre et ferme le fichier log. On remarquera l'absence de sous-commandes on/off, ce qui signifie qu'il n'est possible d'interrompre momentanément l'écriture du fichier log. Lorsque l'on utilise sjlog, trois fichiers sont créés. Le premier

[44] Il s'agit de la version 1.2.7

est un fichier text simple (extension .txt, le second un fichier log traditionnel (extension .smcl) et le dernier un fichier tex (extension .log.tex) avec des commandes LaTeX pouvant être inséré dans un document TeX utilisant le style `stata`. L'option append permet d'ajouter du texte à un fichier log déjà existant alors qu'avec replace, on réécrit l'ancien fichier. Si l'on ajoute noclean à sjlog close, cela permet d'éviter que Stata supprime lors de la cloture du fichier log les instructions relatives à sa manipulation. L'option book avait semble-t'il des raisons historiques d'exister, ce qui n'est pas le cas actuellement puisqu'elle se contente de parsemer votre document tex d'apostrophes. A éviter donc. L'ajout de nologfile permet, comme son nom l'indique, de ne générer que les fichiers tex et txt.

La commande sjlog do permet simultanément d'ouvrir un log et de lancer un fichier do. Cela revient donc au même que d'utiliser sjlog using, puis de lancer un fichier do. Les fichiers créés portent automatiquement le même nom que le fichier do, à moins d'utiliser l'option saving(*nom*) qui permet de préciser quel nom donner aux fichiers d'output. En recourant à l'option clear, la commande program drop _all est lancée avant que ne débute la séquence du fichier do. Enfin, lorsque l'option nostop est ajoutée, le fichier do s'exécute jusqu'à sa fin, quand bien même une erreur serait détectée.

Afin de supprimer ou remplacer dans les différents fichiers certaines lignes liées à la gestion du fichier log, on peut utiliser sjlog clean. On dispose des options suivantes :

- log afin de retirer les lignes d'ouverture et de clôture du log ainsi que la dernière commande si celle-ci est log close ;
- logclose afin de retirer les lignes d'ouverture et de clôture du log ainsi que la dernière commande si celle-ci est logclose ;
- sjlog afin d'effacer la dernière ligne de commande si celle-ci est sjlog close ;
- sjlogdo pour supprimer la dernière ligne si celle-ci est end of do-file.

La commande sjlog type est destinée à la transformation d'un fichier smcl, do ou ado en fichier tex. Avec l'ajout des options smclfile et logfile, on peut obtenir la création supplémentaire de fichiers smcl et txt. L'option path(*chemin*) permet de préciser le chemin du fichier alors que find utilise le chemin que Stata a en mémoire si l'on a utilisé auparavant la commande findfile.

Enfin, la commande sjlog basename ne présente visiblement aucun intérêt. Elle est destinée à découper le nom du fichier entre chemin, nom et extension.

2.10 Dotex

Comme son nom l'indique dotex grossièrement permet de lancer l'exécution d'un fichier do et d'en récupérer l'output sous forme d'un fichier tex au format SJ LATEX. Cette commande[45] est donc similaire à sjlog do. Elle peut être utile lorsque l'on souhaite retranscrire dans le style Stata des lignes de commande et en montrer les résultats bruts. Pour fonctionner, cette commande nécessite donc que soit installé le package permettant l'utilisation de l'environnement `stlog` dans lequel dotex inclue les infos que l'on souhaite récupérer.

2.11 Outtable

Cette commande[46] permet de générer un tableau LATEXpour des matrices définies sous Stata. Le code obtenu est placé dans un fichier tex portant le nom de votre choix. Pour le reste, jetons un oeil sur la syntaxe.

outtable using *fichier* , mat(*matrice*) [replace append nobox center asis caption(*titre*) format(*format*) norowlab]

Comme on peut le remarquer, deux champs sont obligatoires, à savoir le nom du fichier de sortie (pas besoin de préciser l'extension tex) et le nom de la matrice que l'on souhaite récupérer (option mat()).

Les options replace et append n'ayant en principe plus de secret pour vous à ce point du document, je ne m'attarderai pas dessus.

Par défaut, outtable créé des tableaux avec une grille. Si vous souhaitez retirer ce quadrillage, il suffit d'ajouter l'option nobox. Avec center, nos résultats apparaissent centrés.

Il est fréquent de retrouver dans les noms des colonnes ou des lignes des caractères qui ont une signification particulière pour LATEX. Afin d'éviter les erreurs, on peut utiliser l'option asis qui en supprime certains comme les « _ ».

[45] Il s'agit de la version d'Août 2001.
[46] Nous présentons ici la version 1.0.5 de la commande.

Comme toujours, caption permet de définir le titre du futur tableau LaTeX.

L'option format(*format*) permet de définir le format des nombres dans le tableau. On peut ainsi changer les points en virgule, préciser le nombre de chiffres après la virgule... Pour savoir quoi insérer entre les parenthèses, lire le commentaire relatif à cette option pour la commande latabstat dans la section 2.4.

Enfin, on dipose de l'option norowlab pour supprimer la colonne d'étiquette des lignes[47].

Pour finir, un petit exemple autour d'une matrice identité :

TABLEAU 11 – Et une belle matrice

	c1	c2	c3
r1	1		
r2	0	1	
r3	0	0	1

On notera que la partie située au dessus de la diagonale est vide. Lorsque Stata repère des matrices symétriques, il se permet de les présenter sous cette forme. La commande outtable en fait de même.

2.12 Listtex

Avec listtex[48], il est possible récupérer sous une forme compatible avec LaTeX les observations des variables souhaitées[49]. Les séries de données récupérées sont présentées en colonne, comme sous Stata. Ajoutons que le code récupéré peut être très dépouillé, sans environnement LaTeX. Enfin, n'oubliez pas d'utiliser la commande sort avant de lancer listtex afin de s'éviter un tri manuel fastidieux.

La syntaxe est la suivante :

listtex [*variables*] [using *fichier*] [if] [in] [, begin(*texte*) delimiter(*texte*) end(*texte*) missnum(*texte*) rstyle(*style*) headlines(*texte*) footlines(*texte*) nolabel type replace appendto(*fichier*) handle(*fichier*)]

[47] Pour supprimer la ligne d'étiquette des colonnes, pas besoin de colonne ; il vous suffit de supprimer la ligne correspondante dans votre document tex.

[48] Il s'agit de la version de Mai 2004

[49] Cette commande ne se restreint pas à LaTeX, puisque les données peuvent aussi être récupérées de manière à intégrer des pages html ou bien différents types de logiciels de traitement de texte.

Les premières options de cette commandes sont relatives aux éléments que l'on souhaite insérer au niveau de chaque ligne. Ainsi begin(*texte*) et end(*texte*) permettent d'insérer du texte au début et à la fin de chaque ligne d'observation. Ce peut être très pratique pour ajouter des commandes. Pour ceux qui désirent intégrer le code sous LaTeX, il convient de finir le texte à insérer dans l'option end par \\ou \cr[50]. Si l'on souhaite aussi avoir un quadrillage, on peut ajouter \hline de manière à disposer immédiatement des lignes horizontales.

L'option delimiter(*texte*) permet de choisir le séparateur que l'on souhaite utiliser pour identifier nos colonnes. Pour Stata, ce sera évidemment &, ce qui est justement la valeur par défaut de l'option.

Avec missnum(*texte*), on peut choisir le texte que l'on désire faire apparaître pour les observations manquantes. Par défaut, il s'agit d'un blanc.

Plutôt que de définir un par un tous ces éléments de ligne, on peut choisir des styles prédéfinis au travers de l'option rstyle(*style*). Dans le tableau 12 sont présentés les différents styles disponibles. Le style retenu pas défaut est tabular. Précisons enfin que les options précédentes ont la priorité sur rstyle.

Tableau 12 – Les différents styles prédéfinis de listtex

Style	Début	Délimiteur	Fin	Observations manquantes	Description
html	<tr><td>	</td><td>	</td></tr>	rien	HTML table rows
htmlhead	<tr><th>	</th><th>	</th></tr>	rien	HTML table header rows
tabular	rien	&	\\	rien	LaTeX \tabular environment table rows
halign	rien	&	\cr	rien	Plain TeX \halign table rows
settabs	\+	&	\cr	rien	Plain TeX \settabs table rows
tabdelim	rien	char(9)	rien	rien	Tab-delimited text file rows

[50] Avec cr comme *change row*.

Comme nous l'avons déjà indiqué précédemment, listtex se contente par défaut de générer un code très dépouillé. Pour obtenir directement un tableau dans les règles de l'art, il est nécessaire d'utiliser les options headlines(*texte*) et footlines(*texte*) qui permettent d'insérer du texte respectivement en tête et à la fin du code généré. Ceci permet par exemple à l'utilisateur de LaTeX d'ouvrir et fermer ses environnements `table` et `tabular`.

Avec nolabel, on indique à Stata que l'on ne souhaite pas récupérer les séries numériques auxquelles sont associées des étiquettes de valeur sous la forme de ces étiquettes, mais sous format numérique.

Les dernières options sont relatives à la manière dont on souhaite récupérer les données. Par défaut, celles-ci apparaissent dans la fenêtre de résultat de Stata. Avec using on précise que le code doit être sorti dans le fichier log désigné après using. En ajoutant type, on précise que l'on souhaite faire apparaître simultanément le résultat dans la fenêtre de résultat et le log. Les options appendto(*fichier*) et replace permettent de préciser de manière classique si l'on souhaite ajouter le code à un fichier log existant ou, au contraire, à remplacer l'existant. Visiblement, le fichier désigné dans appendto peut être différent de celui indiqué après using. Enfin, l'option handle(*fichier)* donne la possibilité d'ajouter le code à un fichier déjà ouvert.

Références

[1] **Stata (2003) :** "Stata 8 : Reference S-Z". *Stata Press*, 391 p.

[2] **Stata (2003) :** "Stata 8 : Reference A-F". *Stata Press*, 464 p.

A Code des différents tableaux

Ci dessous, vous trouverez les codes LaTeX générés par Stata pour les différents tableaux présentés dans cet article. Ceci vous permettra d'apprécier par vous même la qualité du code utilisé et les modifications que j'ai opéré. Si le code généré par outtex peut sembler relativement lourd, les autres commandes génèrent un code plus simple.

Tableau 1

```
\def\sep{0.5em}
\def\fns{\footnotesize}
\def\onepc{$^{\ast\ast}$}
\def\fivepc{$^{\ast}$}
\def\tenpc{$^{\dag}$}
\def\legend{\multicolumn{2}{l}{\footnotesize{Niveau de significativité:
\hspace{1em} $\dag$ : 10\% \hspace{1em}
$\ast$ : 5\% \hspace{1em} $\ast\ast$ : 1\% \normalsize}}}
\begin{table}[htbp]\centering
\caption{Ma belle régression
\label{tabresult regress}}
\begin{tabular}{l c  }
\hline\hline
\multicolumn{1}{c}{\textbf{Variable}}
  & \textbf{Coefficient} \\
  & \fns{(Ec. Type)} \\
  \hline
Variable de gauche n°1  & -0.003\onepc \\
& \fns{(0.001)}\\[\sep]
Variable de gauche n°2  & -0.715\fivepc \\
& \fns{(0.310)}\\[\sep]
Constante   & 4.070\onepc \\
& \fns{(0.102)}\\[\sep]
\hline
\multicolumn{2}{c}{}\\
\hline
N & \multicolumn{1}{c}{696}\\
R$^{2}$  & \multicolumn{1}{c}{0.029}\\
F $ _{(2,693)}$  & \multicolumn{1}{c}{10.175}\\
\hline
\legend
\end{tabular}
\end{table}
```

Tableau 2

```
\begin{table}[htbp]\centering \caption{Mes belles statistiques
\label{table:sutex}}
\begin{tabular}{l c c  }\hline\hline
\multicolumn{1}{c}{\textbf{Variable}} & \textbf{Mean}
  & \textbf{Std. Dev.} \\ \hline
Variable 1 & 3.828 & 2.253  \\
Variable 2 & 59.591 & 89.836  \\
Variable 3 & 0.239 & 0.426  \\
\multicolumn{1}{c}{N} & \multicolumn{2}{c}{696}\\ \hline
\end{tabular}
\end{table}
```

Tableau 3

```
\begin{table}[htbp]\centering
\caption{\textbf{Encore de belles statistiques}\label{table:latabstat}}
\begin{tabular} {@{} l r r r @{}} \\ \hline
\textbf{Statistique} & \textbf{variable 1} & \textbf{variable 2} &
\textbf{variable 3}\\
\hline
moyenne    &   .2385057   &   59.59052   &   3.827586 \\
écart type    &   .4264764   &   89.83574   &   2.252835 \\
skewness    &   1.227183   &   4.936407   &   .6012083 \\
kurtosis    &   2.505979   &   40.29011   &   2.414294 \\
médiane     &          0   &         31   &          4 \\
N    &        696   &        696   &        696 \\
\hline
\multicolumn{4}{@{}l}{\footnotesize{\emph{Source:}
C:\textbackslash DATA\textbackslash dossier\textbackslash fichier.dta}}
\end{tabular}
\end{table}
```

Tableau 4

```
\begin{table}[htbp]\centering
\caption{\label{var1_by_var2}
\textbf{Variable 1 by variable 2}}
\begin{tabular} {@{} l r r r @{}} \\
\hline
& \multicolumn{3}{@{} c @{}}{\textbf{Variable 2}} \\
\textbf{Variable 1} & Catégorie 1 & Catégorie 2 & Total \\  \hline
Catégorie 1 & 7 & 11 & 18\\
Catégorie 2 & 10 & 13 & 23\\
Catégorie 3 & 6 & 3 & 9\\
Total & 23 & 27 & 50\\\hline
```

```
\multicolumn{4}{@{}l}{\footnotesize{\emph{Source : }}}
\end{tabular}
\end{table}
```

Tableau 7

```
\begin{table}[htbp]\centering
\caption{Encore un jolie tableau\label{tab:tabout}}
\bigskip
\begin{tabular}{lcccccccc}
\hline
& \multicolumn{8}{c}{\textbf{Variable 3}} \\
& \multicolumn{2}{c}{Catégorie 1} & \multicolumn{2}{c}{Catégorie 2}
& \multicolumn{2}{c}{Catégorie 3} & \multicolumn{2}{c}{Total} \\
   & Num &  \%  & Num &  \%  & Num &  \%  & Num &  \% \\
\hline
\hline
\textbf{Variable 1} \\
Catégorie 1 & 5  & 41.666667  & 6  & 33.333333  & 7  & 35  & 18 & 36 \\
Catégorie 2 & 5  & 41.666667  & 8  & 44.444444  & 10  & 50  & 23 & 46 \\
Catégorie 3 & 2  & 16.666667  & 4  & 22.222222  & 3  & 15  & 9 & 18 \\
Total  & 12  & 100  & 18  & 100  & 20  & 100  & 50 & 100 \\
\hline
\hline
\textbf{Variable 2} \\
Catégorie 1 & 7  & 58.333333  & 7  & 38.888889  & 9  & 45  & 23 & 46 \\
Catégorie 2 & 5  & 41.666667  & 11  & 61.111111  & 11  & 55  & 27 & 54 \\
Total  & 12  & 100  & 18  & 100  & 20  & 100  & 50 & 100 \\
\hline
\end{tabular}
\end{table}
```

Tableau 8

```
\begin{table}[htbp]\centering
\caption{Un autre jolie tableau\label{tab:tabout2}}
\bigskip
\begin{tabular}{lccc}
\hline
   & \multicolumn{3}{c}{\textbf{Variable 4}} \\
   & Observations  & Moyenne & Médiane \\
\hline
\hline
\textbf{Variable 1} \\
Catégorie 1 & 18  & 153  & 55 \\
Catégorie 2 & 23  & 143  & 56 \\
Catégorie 3 & 9 & 126  & 64 \\
```

```
Total  & 50  & 144  & 56 \\
\hline
\hline
\textbf{Variable 3} \\
Catégorie 1 & 12  & 67  & 56 \\
Catégorie 2 & 18  & 157  & 49 \\
Catégorie 3 & 20  & 178  & 66 \\
Total  & 50  & 144  & 56 \\
\hline
\end{tabular}
\end{table}
```

Tableau 9

```
\begin{table}[htbp] \caption{Un tableau avec \textsf{est2tex}
\label{table:est2tex}}
\begin{tabular*}{\textwidth}{@{\extracolsep{\fill}}lcc}
& \multicolumn{1}{c}{Variable 1}  & \multicolumn{1}{c}{Variable 1} \\
\cline{2-3}
& \multicolumn{1}{c}{(1)}  & \multicolumn{1}{c}{(2)} \\
\hline
Variable 2  & .004$^{***}$  & .006$^{***}$ \\
& \raisebox{.7ex}[0pt]{\scriptsize (.001)} & \raisebox{.7ex}[0pt]
{\scriptsize (.002)} \\
Variable 3  & -.228$^{***}$ & -.223$^{***}$ \\
& \raisebox{.7ex}[0pt]{\scriptsize (.031)} & \raisebox{.7ex}[0pt]
{\scriptsize (.034)} \\
Variable 4  & & -.062$^{***}$ \\
& & \raisebox{.7ex}[0pt]
{\scriptsize (.009)} \\
Constante  & -.217$^{*}$  & .506$^{***}$ \\
& \raisebox{.7ex}[0pt]{\scriptsize (.128)} & \raisebox{.7ex}[0pt]
{\scriptsize (.171)} \\
Observations  & 696  & 696 \\
Pseudo $R^2$ & .168  & .263 \\
Log de vraisemblance & -318.219  & -281.756 \\
$ \chi^2$ statistique  & 67.225  & 76.81 \\
\hline\hline
\end{tabular*}
\end{table}
```

Tableau 10

```
\begin{table}[htbp]\centering
\caption{Mon beau tableau de régressions\label{table:est2tex-bis}}
\begin{tabular}{lcc}\\
& \multicolumn{2}{c}{Variable 1}\\
\cline{2-3}
Variables & \multicolumn{1}{c}{(1)}  & \multicolumn{1}{c}{(2)} \\
\hline\hline
Variable 2  & 0,004$^{***}$  & 0,006$^{***}$ \\
& \raisebox{,7ex}[0pt]{\scriptsize (0,001)} & \raisebox{,7ex}[0pt]
{\scriptsize (0,002)} \\
Variable 3  & -0,228$^{***}$ & -0,223$^{***}$ \\
& \raisebox{,7ex}[0pt]{\scriptsize (0,031)} & \raisebox{,7ex}[0pt]
{\scriptsize (0,034)} \\
Variable 4  & & -0,062$^{***}$ \\
& & \raisebox{,7ex}[0pt]
{\scriptsize (0,009)} \\
Constante  & -0,217$^{*}$  & 0,506$^{***}$ \\
& \raisebox{,7ex}[0pt]{\scriptsize (0,128)} & \raisebox{,7ex}[0pt]
{\scriptsize (0,171)} \\
Observations  & 696  & 696 \\
Pseudo $R^2$ & 0,168  & 0,263 \\
Log de vraisemblance & -318,219  & -281,756 \\
$ \chi^2$ statistique  & 67,225  & 76,81 \\
\hline\hline
\multicolumn{3}{c}{\footnotesize{Note:Niveaux de significativité:
\hspace{1em}*:10\%\hspace{1em}**:5\%\hspace{1em}***:1\%}}
\end{tabular}
\end{table}
```

Tableau 11

```
\begin{table}[htbp]
\caption{Et une belle matrice}\centering\medskip
\begin{tabular}{lccc}
 & c1  & c2  & c3  \\
r1 & 1 \\
r2 & 0 & 1 \\
r3 & 0 & 0 & 1 \\
\end{tabular}
\end{table}
```