Observatoire Economique et Statistique d'Afrique Subsaharienne

MANUEL PRATIQUE D'INITIATION AU LOGICIEL STATA

Septembre 2012

Table des matières

Pages

Avant-propos

Introduction

Stata est un logiciel de plus en plus utilisé par les statisticiens et économètres. Il a été développé par Stata Corporation et au moment de la rédaction de ce manuel, le logiciel en est à la version 12. Cependant ce manuel est basé sur la version 11. Toutefois les nouvelles versions sont facilement utilisables si on a une bonne maitrise des versions antérieures. L'objectif poursuivi par ce manuel est double :

- Aider à la prise en main du logiciel ;
- Fournir un bagage minimum pour une utilisation pratique susceptible d'assurer la réalisation d'analyse allant de la simple description à l'usage de techniques économétriques approfondies.

Le logiciel Stata allie à la fois une simplicité d'accès et une complétude quant aux techniques avancées (économétrie, analyse des données, traitement de séries temporelles, etc.…). Il présente plusieurs aspects attractifs :

- Il permet une utilisation interactive qui est un élément important de convivialité ;
- Il présente d'immenses possibilités de programmation ;
- Par son réseau d'utilisateurs, on peut disposer des développements très récents en matière statistique et économétrique sur des thématiques économiques et statistiques majeurs ;
- Le logiciel est très rapide d'exécution ;
- La mise au point de programme y est grandement facilitée

Ce présent manuel outre la présentation des principales fonctionnalités de base va s'articuler autour de cas pratiques pour une efficacité plus accrue dans son apprentissage.

I. L'environnement de Stata

1.1 Installation de Stata

A l'instar de la plus part des logiciels, Stata fonctionne sous différents systèmes d'exploitation (Windows, Mac,…). Son installation peut se faire de deux façons :

- ✓ A partir d'un CD et suivre les instructions
- ✓ A partir d'un autre ordinateur où le logiciel est déjà installé, dans ce cas il faut copier le répertoire dans programmes files par exemple sous windows.

Dans les 2 cas vous aurez besoin d'une licence d'utilisation. Une fois cette procédure terminée Stata au démarrage proposera de faire une mise à jour des sous programmes de base dont il a besoin, d'où la nécessité de disposer d'une connexion internet. Les différents sous programmes de base sont gérés directement par Stata à partir de sous répertoires qu'il a crée à l'installation :

- Sous windows : *C:\Programmes\Stata11\ado*. Ce répertoire contient plusieurs sous répertoires notamment base et updates qui vont avoir en leur seins les sous programmes rangés dans d'autres sous répertoire par ordre alphabétique (a, b,…) en considérant la première lettre du sous programme ;

- Stata crée aussi à l'installation un répertoire par défaut pour les données : *c:\data*

1.2 Instructions de gestion de l'environnement de travail

- ✓ Vider la mémoire : *Clear*

- ✓ Allouer de l'espace mémoire pour la base maître : *set mem <taille allouée>*
  Si la mémoire qu'on veut allouer aux données est fixée à 400m on utilise la commande suivante : *set mem 400m*

- ✓ Définition de la dimension maximale des matrices
  Pour allouer une taille maximale de matrice on utilise la commande :

  set matsize 600

- ✓ Nettoyage de la mémoire
  les programmes ou sous programme sont stockées en mémoire vive. Il est recommandé en début de session de vider entièrement la mémoire en tapant la commande suivante : *program drop _all*

- ✓ Gestion des répertoires
  - Changement du répertoire par défaut : *cd "<chemin du répertoire> »*

  - Assignation de répertoires : on désigne des variables globales représentant différents répertoires (voir aussi instruction local):

    global <nom assigné> "<chemin du répertoire> »

Exemples : global dirh "E:\Atelier\menages"
global diri "E:\Atelier\individuel"
global dirsp "E:\Atelier\ado"
global dirlog "E:\Atelier\resultats"

- ✓ Utilisation d'un fichier de sortie pour récupérer l'exécution et les résultats (***log***)
  - Désignation du fichier : ***log using <nom du fichier>,replace***
    Exemple : log using «$dirlog\resultats.log », replace
  - Fermeture du fichier : ***log close***

Remarque : si on ne spécifie pas l'extension du fichier (.txt, ;.log…), Stata génère un fichier avec l'extension smcl qui ne peut être lu que par Stata.

Cf. Exercice N°4

II. Manipulation des données et instructions de programmation

2.1 Lecture des données

2.1.1 Les commandes :

- ✓ Input….end : saisie directe des données dans le programme
- ✓ Use : pour lire directement une base stata qui correspond à un fichier avec une extension *dta*
- ✓ Infile : lecture d'un fichier ascii avec comme séparateur l'espace
- ✓ Insheet : lecture d'un fichier cvs c'est-à-dire un fichier dont les valeurs sont séparées par des virgules. Il faut dans ce fichier que la première ligne contienne les noms des variables.

Cf. Exercices N°1 et 4

2.1.2 Chargement d'une base Stata

2.1.2.1 Spécification du chemin complet

Exemple : use "C:\Manuel Stata\Sortie\ficdep.dta", clear

Cf. Exercices N° 4

2.1.2.2 Utilisation des répertoires assignés

Exemple : use "$sortie\ficdep.dta", clear

2.1.2.3 Utilisation des bases installées par Stata

- ✓ lister les bases de données : ***sysuse dir***
- ✓ Charger en mémoire : ***sysuse <nom de la base>, clear***

2.1.3 Utilisation des labels

- ✓ Labelliser une variable : ***label variable <var> "label"***
- ✓ Labelliser les modalités d'une variable : ***label define***
- ✓ Suppression de labels :
  - Tous les labels : ***label drop _all***
  -
  - Labels d'une variable : ***label drop <var>***

Voir l'aide pour d'autres utilisations des labels: commande stata : ***help label***

Cf. Exercice N°4

2.2 Appariement/Fusion de bases de données

On peut fusionner des bases en exécutant les instructions suivantes :

- ✓ Merge : cette commande sert à fusionner deux bases une maître qui en mémoire et une « secondaire» qui est sur le disque et qui est désignée à la suite de la commande « using ». A partir de la version 11 du logiciel, on dispose de quatre façons de faire la fusion :
  - *Si l'identifiant de fusion est unique dans les deux bases :*
    merge 1:1 <critères> using <base> [, options]
  - *Si l'identifiant est multiple dans la base maitre et unique dans la base secondaire :*
    merge m:1 <critères> using <base> [, options]
  - *Si l'identifiant est unique dans la base maitre et multiple dans la base secondaire :*
    merge 1:m <critères> using <base> [, options]
  - *Si l'identifiant est multiple dans les deux bases :*
    merge m:m <critères> using <base> [, options]

Cf. Exercice N°9

- ✓ Append : pour concaténer deux bases

Exemple : `use "<chemin>\<base maitre>", clear`
`append using "<chemin>\<base secondaire", <options>`

- ✓ Cross : fusionner le contenu de deux bases sans critère de fusion.
  Exemple : `use "<` *chemin*`>\<base maitre>", clear`
  `cross using <`*base secondaire*`>`

2.3 Les boucles

- ✓ Commande foreach:

Syntaxe: foreach var <liste des variables> {
instructions
}

Cf. Exercice N°8

- ✓ Commande forvalues

Syntaxe: forvalues <var=compteur> {
instructions
}

Cf. Exercice N°8

- ✓ Commandes while et local

  - La commande while permet d'exécuter un groupe d'instructions tant que la condition n'est pas vérifiée. Donc il faut bien s'assurer que dans la boucle la condition peut être atteinte sinon on a une boucle sans fin.

while <condition> {
instructions
}

  - L'instruction local permet de définir des macro variables qui peuvent s'avérer très utiles pour :

    - assigner un répertoire :

local <nom assigné> "<chemin du répertoire> »

    - définir un scalaire : *local <var=valeur>*

    - indicer d'autres variables :

```
local i=1
local j=0
while `i'<=9 {
local j=`i'-1
replace lower=d`j' if deptete>=d`j' & deptete<d`i'
replace upper=d`i' if deptete>=d`j' & deptete<d`i'
local i=`i'+1
}
```

Cf. Exercice N°9

2.4 Utilisations de sous programmes adhoc

- ✓ Chargement des sous-programmes en mémoire

do "<chemin>\<nom du sous programme>"

Remarque : on peut se passer du chargement des sous programmes, pour cela il faut les copier manuellement dans le répertoire qu'utilise stata au démarrage en prenant soin dans ce répertoire (qui s'appelle base) de le copier dans le sous répertoire indiqué par la première lettre alphabétique du sous programme :

- ✓ Exécution des sous-programmes

  Pour connaître la syntaxe d'un sous programme il faut utiliser la commande ***Help <nom du sous programme>*** dans la fenêtre command de Stata

Cf. Exercice N°7

2.5 Les variables systèmes, les <return list>/ et les résultats des procédures*

- ✓ List : pour lister à l'écran le contenu de variables : ***list <var>***

- ✓ Return list : cette commande permet d'*obtenir les scalaires calculés par une procédure* (on peut les récupérer pour une utilisation ultérieure)
  Exemple : sum <var>, det
  return list

- ✓ *Ereturn list :* cette commande permet d'*obtenir les scalaires calculés et stockés dans le vecteur e() et donne en même temps les noms des vecteurs et matrices où sont stockés les résultats ce qui très utile notamment après les régressions pour accéder aux différentes statistiques générés*
  Exemple : reg x y z
  ereturn list

- ✓ *Sreturn list :* cette commande permet d'*obtenir les scalaires calculés et stockés dans le vecteur s()*
  Exemple : mean x
  Sreturn list

- ✓ Display : pour l'affichage à l'écran de résultats
  Exemple : sum <var>, det
  d r(mean)

- ✓ _n : variable système correspondant au rang de l'observation en cours
  Exemple : gen rang=_n

2.6 Quelques instructions utiles

- ✓ preserve et restore
  Ces deux commandes permettent dans un programme de préserver les données en mémoire pour pouvoir exécuter d'autres instructions y compris de charger une nouvelle base et de les restaurer pour la suite des traitements.
  Syntaxe: *preserve <instructions> restore*

Cf. Exercice N°1

- ✓ Reshape: conversion de données de format long en format court et vice versa

Cf. Exercice N°1

- ✓ Collapse : <u>agrégation de variables</u>
  Cette instruction très importante permet de faire des agrégations de variables : calcul de statistiques simples (moyenne, médiane,…) avec ou sans critère de sélection
  Syntaxe : *collapse <statistiques> <condition> <champs> <pondération> ,> options>*

  Quelques exemples de statistiques :
  Moyenne (par défaut), médiane, p1…p99, écart-type, somme max, min, etc.…

Cf. Exercice N°2

- ✓ Compress : compression des données en mémoire ce qui permet un gain d'espace

- ✓ Sort : trie les données

- ✓ Bysort : trie les données sur un critère (variable)
  Exemple : *bys sexe : sum depense* donne les dépenses moyennes des hommes et des femmes
- ✓ Recode : instruction pour recoder une variable avec regroupement de certaines modalités.

  Exemple : la variable statut matrimonial (matri*)* a 9 modalités :
  1 : Marié(e)monogame
  2 : Marié(e)polygame avec 2 épouses
  3 : Marié(e)polygame avec 3 épouses
  4 : Marié(e)polygame avec 4 épouses
  5 : Marié(e)polygame avec plus de 4 épouses
  6 : Célibataire
  7 : Veuf(ve)
  8 : Divorcé(e)
  9 : Concubinage ou union libre
  On regroupe les modalités 2 à 5 en leur affectant la modalité 2 pour créer une seule catégorie polygame. On décale les modalités 6 à 7 en leur affectant respectivement les valeurs de 3 à 5, puis on regroupe les modalités 1 et 9. On génère une nouvelle variable *nmatri* contenant le résultat de ce recodage.
  recode matri (2/5=2) (6=3) (7=4) (8=5) (9=1), gen(nmatri)

Cf. Exercice N°1

- ✓ Egenerate : ajoute des statistiques globales à chaque observation
  Syntaxe : *egen <variable globale à générer>=<fonction><variable>*

  Exemple : pour centrer et réduire la variable dépense

  egen sigma=sd(depense)
  egen moyenne=mean(depense)
  gen depcr=(depense-moyenne)/sigma

Cf. Exercice N°2

III. Traitements statistiques et économétriques

Pour les traitements statistiques et/ou économétriques, on a deux possibilités :

- ✓ Soit utiliser les menus déroulants une fois qu'on a chargé sa base de données, ce qui permet dans le cas où on ne connait pas la syntaxe, de faire son traitement et d'apprendre en même temps les instructions qui sont

générés dans la fenêtre résultat. De manière globale on peut aussi utiliser ces menus pour les manipulations des données chargées précédemment.

- ✓ Soit écrire un programme dans un ***do file***, ce qui est privilégié dans ce manuel. Cette solution à l'avantage de laisser une trace de tout le traitement effectué.

3.1Statistiques simples

3.1.1 Indices de tendances centrales (mean, ameans, sum….)

3.1.2 Tableaux de fréquences

3.1.2.1 Tableaux simples

tab <variable>

- *avec génération de variables dichotomiques :*

  tab <variable>, gen(préfixe)

- *tableaux de fréquence de plusieurs variables :*

  tab1 <variable1 <variable2> …<variablen>

Cf. Exercice N°9

3.1.2.2 Tableaux croisés

- *tab <variable1> <variable2>,col row*
- *tabstat <variable1 <variable2> …<varn>*

Cf. Exercice N°8

3.1.3 Distributions

A partir d'exemples simples on va introduire quelques instructions permettant de connaître la distribution de variable :

- ✓ centrer et réduire une variable :
  - *Calcul de la distribution : sum <variable>, det*
  - *Utilisation de la moyenne et de l'écart type pour centrer et réduire la variable à partir des résultats de la commande précédente :*

    gen <variablec>=(<variable>-r(mean))/r(sd)

- ✓ calcul des déciles d'une variable : *centile <variable>, centile(10(10)90)*
- ✓ corrélation de deux variables *: corr <variable1><variable2>*
  ou corr <variable1><variable2>,covariance

- ✓ corrélation avec test de nullité : ***pwcorr <variable1> <variable2>***

Cf. Exercice N°8

3.1.4 Estimation de la densité d'une variable continue

Les principales méthodes d'estimations de la densité d'une variable sous Stata sont:

- ✓ Epanechnikov qui est la méthode par défaut
- ✓ epan2 : methode alternative
- ✓ biweight
- ✓ cosine
- ✓ gaussian
- ✓ parzen
- ✓ rectangle
- ✓ triangle

Syntaxe :
kdensity <variable>, kernel(<method>) gen(<abcisse et ordonnée>) <options: nograph>

Cf. Exercice N°5

3.2 Analyse des données

3.2.1 Analyse en composantes principales (ACP)

- ✓ Syntaxe: *pca <liste des variables> <critère de sélection> <pondération>, <options>*

Cf. Exercice N°13

3.2.2 Analyse factorielle

- ✓ Syntaxe:
  factor <liste des variables> <critère de sélection> <pondération>, < méthode>

Cf. Exercice N°14

3.2.3 Analyse en correspondances multiples

- ✓ Syntaxe:
  mca <liste des variables> <critère de sélection> <pondération>, < méthode>

3.3 Econométrie

Stata étant à la base un logiciel d'économétrie, les méthodes offertes sont nombreuses. On va se contenter de quelques exemples.

- ✓ Régression avec contraintes
  - o Générer aléatoirement des variables à partir de fonctions de densités

*gen e=invnorm(uniform())*2*
*gen x=uniform()*6*
*gen y=3+5*x+e*
*gen z=uniform()*6*

- Définition de la contrainte sur le coefficient de x

  constraint define 1 x=3.5

- Instruction de la régression sous contrainte

  cnsreg y x z, constraint(1)
  reg y x z

- Calcul de la prédiction de y (yp)

  predict yp

- Calcul de la prédiction des résidus de la régression

  predict residu, resid

- Test pour l'égalité des coefficients

  test x=z

- Graphique de y et yp

  tw scatter yp y

- Sauvegarde des coefficients dans la variable coeff

  estimates store coeff

- Restauration des coefficients et prédiction de y (py)

  estimates restore coeff
  predict py

✓ Estimation avec variables instrumentals

ivreg y x sexcm region (x=sexcm tage)

✓ Quelques instructions pour effectuer des régressions:

Remarque : utiliser la commande ***help <nom instruction>*** pour obtenir la syntaxe exacte.

- *areg: une manière facile d'estimer des régressions avec beaucoup de variables dichotomiques.*
- *arch: estimation de modèles arima: estimation de modèle ARIMA (Box & Jenkins) boxcox: estimation de modèles Box-Cox*
- *cnreg: regression avec variable censurée*
- *eivreg: errors-in-variables regression*
- *frontier:stochastic frontier models*
- *heckman: estimation modèle de selection Heckman*
- *intreg: estimation modèle par intervalle*
- *ivregress: single-equation instrumental-variables regression*
- *ivtobit: tobit regression with endogenous variables*

- *newey: regression with Newey-West standard errors*
- *qreg: quantile (including median) regression*
- *reg3: three-stage least-squares (3SLS) regression*
- *rreg: a type of robust regression*
- *sureg: seemingly unrelated regression*
- *tobit: tobit regression*
- *treatreg: treatment-effects model*
- *truncreg: truncated regression*
- *xtabond: Arellano-Bond linear dynamic panel-data estimation*
- *xtdpd:linear dynamic panel-data estimation*
- *xtfrontier: panel-data stochastic frontier model*
- *xtgls: panel-data GLS models*
- *xthtaylor: Hausman-Taylor estimator for error-components models*
- *xtintreg: panel-data interval regression models*
- *xtivreg: panel-data instrumental variables (2SLS) regression*
- *xtpcse: linear regression with panel-corrected standard errors*
- *xtreg: fixed- and random-effects linear models*
- *xtregar: fixed- and random-effects linear models with an AR(1) disturbance*
- *xttobit: panel-data tobit models*
- *etc.*

Stata permet d'estimer aussi des séries temporelles en vue d'effectuer des prévisions

Cf. Exercices N^{os}11 et 12

IV. Graphiques

Stata offre une grande diversité de graphiques:

- ✓ histogramme :
  - *hist <var>*
  - *graph twoway histogramm <var1>*
  - etc.
- ✓ Nuage de points :
  - *tw scatter <var1> <var2>*
  - *graph <var1> <var2>*
  - *graph line ou tw line ou line*
  - etc.

Cf. Exercices N^{o}10

V. Exercices

5.1 Manipulations de données

Exercice N°1

1. Entrer les données suivantes :

Menage	individu	depense
1	1	20000
1	2	15000
1	3	40000
2	1	60000
2	2	10000
3	1	25000
4	1	30000
4	2	20000
4	3	15000
5	1	70000

2. Créer des variables « *depense* » indicées par le rang de l'individu : aller d'un tableau long à un tableau court
   résultat :

menage	depense1	depense2	depense3
1	20000	15000	40000
2	60000	10000	.
3	25000	.	.
4	30000	20000	15000
5	70000	.	.

3. Créer une seule variable « *depense* » pour tous les individus : aller du tableau court au tableau long
4. Revenir au tableau court
5. Revenir au tableau long
6. Calculer la dépense moyenne du ménage en effectuant un tri
7. En utilisant *preserve* et *restor*e, calculer la distribution de la dépense par tête contenue dans la base *ficdep.dta* en restaurant la base de départ à la fin du traitement

Correction – Exercice N°1

Question 1: Entrer les données suivantes

```
clear
input    menage        individu      depense
         1             1             20000
         1             2             15000
         1             3             40000
         2             1             60000
         2             2             10000
         3             1             25000
         4             1             30000
         4             2             20000
         4             3             15000
         5             1             70000
end
```

Question 2: Créer des variables « *depense* » indicées par le rang de l'individu : aller d'un tableau long à un tableau court *
reshape wide depense, i(menage) j(individu)

Question 3: Créer une seule variable « *depense* » pour tous les individus : aller du tableau court au tableau long *
reshape long depense, i(menage) j(individu)

Question 4: Retour au tableau court*
reshape wide

Question 5: Retour au tableau long*
reshape long
drop if depense==.

Question 6: Calculer la dépense moyenne du ménage en effectuant un tri
bys menage: sum depense

/* Remarque : en utilisant la commande *means* à la place de la commande sum on obtient les valeurs des moyennes arithmétique, géométrique et harmonique*/

Question 7: Calcul la distribution de la dépense par tête contenue dans la base *ficdep.dta* avec restauration la base de départ à la fin du traitement*
preserve
l'environnement de travail
global entree = "C:\Manuel Stata\Entrée"
global sortie = "C:\Manuel Stata\Sortie"

clear all
set memory 400m
set more off

Calcul la distribution la dépense par tête
use "$sortie\ficdep.dta", clear
sum deptete [w=pond3], det
restore

Exercice N°2

1. Entrer les données suivantes et calculer la dépense moyenne par sexe:

menage	individu	sexe	depense
1	1	Homme	20000
1	2	Femme	15000
1	3	Homme	40000
2	1	Femme	60000
2	2	Femme	10000
3	1	Homme	25000
4	1	Homme	30000
4	2	Femme	20000
4	3	Homme	15000
5	1	Femme	70000

2. A l'aide de la commande egenerate
   a. on va centrer et réduire la variable dépense
   b. générer une variable x contenant un numéro différent pour chaque modalité de la variable sexe
3. créer une base ménage contenant pour chaque observation le numéro du ménage et la dépense totale

Correction Exercice N°2

Question 1: Entrer les données suivantes et calculer la dépense moyenne par sexe

```
clear
input menage individu   str6 sexe       depense
        1       1       Homme           20000
        1       2       Femme           15000
        1       3       Homme           40000
        2       1       Femme           60000
        2       2       Femme           10000
        3       1       Homme           25000
        4       1       Homme           30000
        4       2       Femme           20000
        4       3       Homme           15000
        5       1       Femme           70000
end

bys sexe: sum depense
```

Question 2.a: Centrer et réduire la variable dépense : utilisation de egenerate

```
egen sigma=sd(depense)
egen moyenne=mean(depense)
gen depcr=(depense-moyenne)/sigma
```

Question 2.a : générer une variable x contenant un numéro différent pour chaque modalité de la variable sexe: utilisation de egenerate

```
egen x=group(sexe)
```

Question 3: créer une base ménage contenant pour chaque observation le numéro du ménage et la dépense totale : utilisation de collapse*

```
collapse (sum) depense, by(menage)
```

Exercice N°3

Question 1: lire uniquement les variables *sex male* et *female* dans la base sortie (qui contient d'autres variables) et faire la sauvegarde dans une nouvelle base sex.dta

```
clear
global sortie = "C:\Manuel Stata\Sortie"
input str6 sex
male
female
end
save "$sortie\sex.dta",replace
```

Question 2: enlever toutes les variables de la mémoire, entrer les valeurs de la nouvelle variable agecat, compléter les observations sur agecat par la variable sex de la base sauvegardée sex.dta fusion et afficher le résultat.

```
drop _all
input agecat
20
30
40
end
cross using "$sortie\sex.dta"
list
```

5.2 Manipulation des labels

Exercice N°4

1. Définir l'environnement de travail
2. Utiliser la base ficdep.dta, vérifier les labels de la variable a01 qui représente la région et supprimer ces labels en vérifiant qu'ils sont bien supprimés
3. Donner des labels aux modalités de la variable région (a01) et les vérifiés
4. Supprimer tous les labels de la base

Correction – Exercice N°4

Question 1 : l'environnement de travail*
```
version 10
capture clear
capture log close
global entree = "C:\Manuel Stata\Entrée"
global sortie = "C:\Manuel Stata\Sortie"

clear all
set memory 400m
set more off
```

Question 2: Enlever les labels des modalités de la variable région (a01)*
```
use "$sortie\ficdep.dta", clear
tab a01 /*vérification des labels avant de les supprimer*/
label drop a01
tab a01  /*vérification des labels après suppression*/
```

Question 3: Donner des labels aux modalités de la variable région (a01)*
```
lab def a01 1"Dakar"  2"Ziguinchor" 3"Diourbel" 4"Saint-Louis" 5"Tambacounda" 6"Kaolack"  7"Thiès" ///
        8"Louga"  9"Fatick" 10"Kolda"  11"Matam" 12"Kaffrine" 13"Kédougou" 14"Sédhiou"
tab a01 /*vérification des labels après ajout */
```

Question 4 : Suppression de tous les labels de la base
```
label drop _all
```

5.3 Estimation des fonctions de densité

Exercice N°5

1. Définir l'environnement de travail
2. Utiliser la commande kdensity pour estimer les courbes de densité de la dépense par tête suivantes avec la méthode du noyau
    a. de Parzen sans pondération avec graphique
    b. d'Epanechnikov avec pondération sans graphique

Correction – Exercice N°5

Question 1 : l'environnement de travail*
```
version 11.2
capture clear
capture log close
global entree = "C:\Manuel Stata\Entrée"
global sortie = "C:\Manuel Stata\Sortie"

clear all
set memory 400m
```

```
set more off
```

Question 2: utilisation de la commande kdensity*

```
use "$sortie\ficdep.dta", clear
gen pond=abs(int(poids3))
```

Question 2.a: méthode de noyau de Parzen sans pondération avec graphique

```
kdensity deptete, kernel(parzen)gen(x1 parzen)
```

Question 2.b: méthode de noyau d'epanechnikov sans pondération sans graphique

```
kdensity deptete [w=pond], kernel(epanechnikov)gen(x2 epan) nograph
```

5.4 Aperçu de quelques modèles de simulations avec Stata

Exercice N°6

On veut simuler par la méthode de Monte Carlo une loi normale

1. Définir le sous programme de la loi normale
    a. Il faut toujours quand on veut définir un sous programme s'assurer qu'aucun autre sous programme du même nom ne figure déjà en mémoire donc l'effacer
    b. Définir le sous programme loinorm de la loi normale centrée et réduite
    c. Simulation sur 100 observations et 10 réplications
    d. Calculer les distributions des paramètres de la loi normale simulés (moyenne et variance)

Correction – Exercice N°6.1

```
************************************************************
* sous programme: simulations Monte Carlo                  *
************************************************************
```

Question 1.a : effacer le sous programme loinorm de la mémoire

```
program drop loinorm
```

Question 1.b : sous programme loinorm de la loi normale centrée et réduite

```
program define loinorm, rclass
syntax [, obs(integer 1) mu(real 0) sigma(real 1)]
drop _all
set obs `obs'
tempvar z
gen z=exp(`mu'+`sigma'*invnorm(uniform()))
summarize `z'
return scalar mean=r(mean)
return scalar Var=r(Var)
end
```

Question 1.c : Simulation sur 100 observations et 10 réplications

```
simulate "loinorm,obs(100)" mean=r(mean) var=r(Var), reps(10)
```

Question 1.d : Distributions des paramètres de la loi normale simulés (moyenne et variance)

```
sum b0, det
sum b1, det
```

2. Simulation de l'estimateur des moindres carrés ordinaires (mco)

   a. Définir le sous programme de simulation de l'estimateur des moindres carrés ordinaires (mco)
   b. Exécution du sous programme avec 1000 réplications
   c. Calculer les distributions des coefficients simulés

Correction – Exercice N°6.2

Question 1.a : Sous programme de simulation de l'estimateur des moindres carrés ordinaires

```
program define smco, rclass
drop _all
set obs 100
gen e=invnorm(uniform())*2
gen x=uniform()*6
gen y=3+5*x+e
regress y x
return scalar b0=_coef[_cons]
return scalar b1=_coef[x]
end
```

Question 1.b : Exécution du sous programme avec 1000 réplications

```
simulate "smco" b0=r(b0) b1=r(b1), reps(1000)
```

Question 1.c : Calculer les distributions des coefficients simulés

```
sum b0, det
sum b1, det
```

5.5 Utilisation de sous-programmes

Exercice N°7

1. Définir l'environnement de travail
2. Charger en mémoire les sous programmes pour calculer les indicateurs d'inégalité
3. Appel des sous programmes et calcul des indicateurs d'inégalité sur les dépenses par tête des ménages en utilisant les pondérations

Question 1 : l'environnement de travail*

```
version 10
capture clear
capture log close
global entree = "C:\Manuel Stata\Entrée"
global sortie = "C:\Manuel Stata\Sortie"
global dirsp = "C:\Manuel Stata\ado"


clear all
set memory 400m
set more off
```

Question 2: Chargement des sous programmes pour calculer les indicateurs d'inégalité*

```
do "$dirsp\inequal.ado"
do "$dirsp\relsgini.ado"
do "$dirsp\atkinson.ado"
do "$dirsp\lorenz.ado"
do "$dirsp\ineqerr.ado"
do "$dirsp\descogini.ado"
```

Question 3: appel des sous programmes et calcul des indicateurs d'inégalité*

```
use "$sortie\ficdep.dta", clear
gen pond=abs(int(poids3))
inequal deptete [w=pond]
atkinson deptete [w=pond], e(0,0.5,1)
lorenz deptete [w=pond]
relsgini deptete [w=pond]
```

5.6 Evaluation de la pauvreté relative

Exercice N°8

On dispose d'une base de données *(ficdep.dta)* issue de l'étude de cas 1. On veut calculer trois taux de pauvreté relative de la manière suivante : est pauvre tout ménage dont la dépense par tête est inférieure à i) 50%, ii) 60% et iii) 70% de la médiane des dépenses par tête.

1. Définir l'environnement de travail
2. Calculer la distribution la dépense par tête, des trois seuils de pauvreté relative et les taux de pauvreté correspondants
3. Calculer directement les taux, les profondeurs et les sévérités de la pauvreté pour les trois seuils: Indices FGT
4. Donner quelques caractéristiques sociodémographiques des pauvres et des non pauvres : statistiques descriptives
5. Estimer économétriquement des déterminants de la pauvreté à l'aide des caractéristiques sociodémographiques : modèle logit

Correction – Exercice N°8

Question 1: l'environnement de travail

```
version 10
capture clear
capture log close
global entree = "C:\Manuel Stata\Entrée"
global sortie = "C:\Manuel Stata\Sortie"

clear all
set memory 400m
set more off
```

Question 2: Calculer la distribution la dépense par tête, des trois seuils de pauvreté relative et les taux de pauvreté correspondants

```
use "$sortie\ficdep.dta", clear
pctile decile=deptete [w=pond3], nq(10)
drop decile
forvalues f =5/7{
   gen seuil`f' = r(r5)*(`f'/10)
    gen poor`f' =(deptete<seuil`f')
  }
tabstat poor5 poor6 poor7 [w=pond3]
```

Question 3: Calcul direct des taux, des profondeurs et des sévérités de la pauvreté pour les trois seuils: Indices FGT

```
forvalues f =5/7{
gen P0_`f'=100*(deptete<seuil`f') /* Taux de pauvreté*/
gen P1_`f'=((1-deptete/seuil`f')^1)*P0_`f' /* Gap/profondeur de pauvreté  */
gen P2_`f'=((1-deptete/seuil`f')^2)*P0_`f' /* Sévérité de pauvreté  */
}
```

Question 4: Donner quelques caractéristiques sociodémographiques des pauvres et des non pauvres : statistiques descriptives

```
foreach x in poor5 poor6 poor7{
label  def `x' 0"non pauvre" 1"pauvre"
}
foreach var in poor5 poor6 poor7{
foreach car in milieu b2 b4 b5{
tab `var' `car' [aw=poids3], row col
}
}
```

Question 5: Estimer économétrique des déterminants de la pauvreté à l'aide des caractéristiques sociodémographiques : modèle logit

```
foreach var in poor5 poor6 poor7{
logit `var' sexe2 mat2-mat5 strate1 [pw=pond3] /*, noconstant*/
}

forvalues f =5/7{
  foreach var in P0_`f' P1_`f' P2_`f'{
        di in green "ratio de la médiane ="  in yellow "`f'" "0%"
        mean `var' [aw=poids3]
                              }
           }
```

/*Remarque : pour remplacer les coefficients par les odd-ratios il faut remplacer la commande logit par logistic *<logistic `var' sexe1 sexe2 mat1-mat5 strate1 strate2 [pw=pond3]>*/*

5.7 Estimation des résidus simulés

Exercice N°9

On dispose de 4 bases de données :

- ✓ Une **base individuelle** *(findividu.dta)* de 11 variables et 176296 observations avec les variables suivantes :
  - identifiant : a07b(DR), a01(Région), a02(Département) et a08(Ménage)
  - Caractéristiques de l'individu : b1(Lien de parenté avec le chef du ménage), b2(Sexe de l'individu), b4(Situation matrimoniale), b5(Situation de résidence), c2b(Formation professionnelle ou technique suivie) , c3(Diplôme professionnelle ou technique le plus élevé obtenu), milieu(milieu de résidence)

- ✓ Une base *(fpoids.dta)* de 7 variables et 17891 observations contenant les **pondérations** avec les variables suivantes:
  - identifiant : a07b(DR), a01(Région), a02(Département) et a08 (Ménage)
  - Pondération : poids3 (Pondération tout échantillon), poids3_a (Pondération sous-échantillon), taille (Taille du ménage)

- ✓ Une base *(fdepense.dta)* de 3 variables et 17849 observations des **dépenses du ménage** avec les variables suivantes :
  - identifiant : a07b(DR)et a08(Ménage)
  - deptot (dépense totale du ménage)

Pour tester la « méthode des résidus simulés » qui consiste à rendre continue une variable en tranches pour en étudier plus finement la distribution, on a à programmer les étapes ci-dessous :

1. Définir l'environnement de travail :
   a. Indiquer la version de stata pour la compilation des instructions et la sauvegarde des bases

b. Fermer par précaution le fichier de résultats (log)
c. désigner par *entrée* le répertoire des fichiers de base ci-dessus
d. désigner par *sortie* le répertoire des fichiers à sauvegarder
e. allouer une mémoire de 400m pour les données
f. désactiver l'interrupteur de défilement des résultats

2. Sélectionner les variables caractéristiques du chef de ménage. Pour cela on part de la base du niveau individu pour récupérer à l'aide de la variable lien avec le chef de ménage.

3. Regroupements de certaines modalités des variables

4. Dichotomiser les variables caractéristiques du chef de ménage

5. Sauvegarde de la base du chef de ménage *(fchef.dta)* en ne gardant que les variables utiles

6. Normaliser les pondérations du fichier *fpoids*.dta en le sauvegardant dans le répertoire de sortie

7. Fusionner les 3 bases (*fchef.dta, fdepense.dta, fpoids.dta*)
   a. Générer une variable dépense par tête (*deptete*)
   b. Sauvegarder le résultat en appelant le fichier *ficdep.dta*

8. Pour chaque ménage générer un intervalle pour la dépense par tête
   a. Calculer la distribution de la variable *deptete*
   b. A partir des résultats créer les bornes qui respondent au décile de dépense : lower=di et upper=dj si di<deptete<=dj

9. Estimation économétrique des bornes en fonction des caractéristiques du ménage et prédiction de la variable en continue

10. Ajouter des aléas à la variable en continue

Correction – Exercice N°9

Question 1 : l'environnement de travail

```
version 11
capture clear
capture log close
global entree = "C:\Manuel Stata\Entrée"
global sortie = "C:\Manuel Stata\Sortie"

clear all
set memory 400m
set more off
```

Question 2 : Sélection des variables

```
*Variables socio demo*/
use "$entree\findividu", clear
keep if b1==1
```

Question 3 : Regroupements de certaines modalités des variables

```
replace b4=2 if b4==3 | b4==4 | b4==5
replace b4=3 if b4==6 | b4==9
replace b4=4 if b4==7
replace b4=5 if b4==8
label drop b4
lab def b4 1 "Marié(e)monogame" 2 "Polygame" 3 "Célibataire" 4 "Veuf(ve)" 5 "Divorcé(e)"
replace b5=1 if b5==3
gen profes=c2b
replace profes=round(c2b/10) if c2b>9
replace c3=12 if c3==99
```

```
replace c3=c3+1
replace c3=4 if c3==5 | c3==6
replace c3=5  if c3==7
replace c3=6 if c3==8 | c3==9 | c3==10
replace c3=7 if c3==11
replace c3=8 if c3==12 | c3==13
label drop c3
lab def c3 1 "Aucun" 2 "Formation certifiante" 3 "CAP" 4 "BEP-BP-BT" 5 "BAC (T1,T2,G,S3,S4,S5)" 6 "DTS-BTS- DUT" 7 "Ingénieur" 8"Autres à préciser-Ne sait pas"
```

Question 4 : Dichotomisation des variables

```
tab b2, gen(sexe)
tab b4, gen(mat)
tab b5,gen(resid)
tab profes,gen(prof)
tab c3,gen(dip)
tab milieu,gen(strate)
```

Question 5 : Sauvegarde de la base du chef de ménage en ne gardant que les variables utiles

```
keep a01 a02 a07b a08 mat1-mat5 prof1-prof9 resid1-resid2 dip1-dip8 resid1-resid2 strate1-strate2
sort a07b a08
saveold "$sortie\fchef.dta", replace
```

Question 6 : Normaliser les pondérations du fichier fpoids.dta en le sauvegardant dans le répertoire de sortie

```
use "$entree\fpoids", clear
keep a01 a02 a07b a08 taille poids3 poids3_a
sum poids3
gen pond3=poids3/r(mean)
sum poids3_a
gen pond3_a=poids3_a/r(mean)
sort a07b a08
save "$sortie\fpoids.dta", replace
```

Question 7 : Fusion des 3 bases

```
use "$entree\fdepense.dta", clear
sort a07b a08
merge 1:1 a07b a08 using "$sortie\fchef.dta"
/*tab _merge :instruction exécuter automatiquement à partir de la version 11*/
drop _merge
sort a07b a08
merge 1:1 a07b a08 using "$sortie\fpoids.dta"
gen deptete=deptot/taille
drop _merge
save "$sortie\ficdep.dta", replace
```

Question 8: Définition des bornes la dépense par tête

```
use "$sortie\ficdep.dta", clear
pctile decile=deptete [w=pond3], nq(10)
drop decile
```

Bornes qui correspondent aux déciles de dépense : lower=di et upper=dj si di<deptete<=dj

```
sum deptete, det
gen lower=r(min)
gen upper=r(max)
pctile decile=deptete [w=pond3], nq(10)
drop decile
local i=1
local j=0
while `i'<=9 {
```

```
local j=`i'-1
replace lower=r(r`j') if deptete>=r(r`j') & deptete<r(r`i')
replace upper=r(r`i') if deptete>=r(r`j') & deptete<r(r`i')
local i=`i'+1
}
replace lower=r(r9) if deptete>r(r9)
```

Question 9: Estimation économétrique des bornes en fonction des caractéristiques du ménage et prédiction de la variable en continue

```
intreg lower upper sexe2 mat2-mat5 strate1 [w=pond3] /*, noconstant*/
predict hatdep
```

Question 10: Ajouter des aléas à partir d'une loi uniforme à la variable prédite

```
gen alea=0
gen prevu=0
local i=1
while `i'<=100 {
replace u=uniform()
replace alea=log(u/(1-u))
replace prevu=hatdep+exp(alea) /*if prevu>=lower & prevu<upper*/
local i=`i'+1
}
```

5.8 Les graphiques sous Stata

Exercice N°10

1. Définir l'environnement de travail en changeant de répertoire par défaut
2. Tracer des histogrammes et des courbes Box-Plot
3. Tracer des courbes avec sauvegarde de graphiques dans un répertoire et tracer un graphique combiné des graphiques sauvegardés
4. Supprimer les graphiques sauvegardés du répertoire par défaut
5. Tracer des graphiques après une régression
6. Exemples de tests de significativités après une régression
7. Tracer le graphique de la vitesse médiane sur le poids
8. Lister les bases de données installées dans la bibliothèque de Stata et utilisation de la base auto pour des graphiques

Correction – Exercice N°10

Question 1 : l'environnement de travail*

```
version 11.2
capture clear
capture log close
global entree = "C:\Manuel Stata\Entrée"
global sortie = "C:\Manuel Stata\Sortie"

clear all
set memory 400m
set more off
*Changement de repertoire par défaut*
cd "C:\Manuel Stata\Resultats"
```

Question 2: Histogramme , Box-Plot*

```
use "$sortie\ficdep.dta", clear
gen pond=abs(int(poids3))
sort a01
histogram b4
histogram a01 [fweight = pond], discrete percent
```

```
graph box deptete [fweight = pond], by(a01)
graph hbox deptete [fweight = pond], by(a01)
graph bar (median) deptete [fweight = pond], by(a01)
```

Question 3: Courbes et sauvegarde de graphiques dans un répertoire - combinaison de graphiques*

```
collapse (mean) (sexe1 sexe2 mat1-mat5 resid1-resid2) [w=poids3], by(a01)
 twoway (line sexe1 a01)

line sexe1 a01, title("hommes") saving(sexe1)
line sexe2 a01,title("femmes") saving(sexe2)
gr combine sexe1.gph sexe2.gph, title("Sex")
*Question 4: suppression de graphiques*
erase sexe1.gph
erase sexe2.gph
*Question 5: Graphiques après une régression*
preserve
clear

input str20 modele cylindre puissance vitesse poids longueur largeur
 "Honda Civic"          1396   90   174   850   369   166
 "Renault 19"           1721   92   180   965   415   169
 "Fiat Tipo"            1580   83   170   970   395   170
 "Peugeot 405"          1769   90   180  1080   440   169
 "Renault 21"           2068   88   180  1135   446   170
 "Citroen BX"           1769   90   182  1060   424   168
 "BMW 530i"             2986  188   226  1510   472   175
 "Rover 827i"           2675  177   222  1365   469   175
 "Renault 25"           2548  182   226  1350   471   180
 "Opel Omega"           1998  122   190  1255   473   177
 "Peugeot 405 Break"    1905  125   194  1120   439   171
 "Ford Sierra"          1993  115   185  1190   451   172
 "BMW 325iX"            2494  171   208  1300   432   164
 "Audi 90 Quattro"      1994  160   214  1220   439   169
 "Ford Scorpio"         2933  150   200  1345   466   176
 "Renault Espace"       1995  120   177  1265   436   177
 "Nissan Vanette"       1952   87   144  1430   436   169
 "VW Caravelle"         2109  112   149  1320   457   184
 "Ford Fiesta"          1117   50   135   810   371   162
 "Fiat Uno"             1116   58   145   780   364   155
 "Peugeot 205"          1580   80   159   880   370   156
 "Peugeot 205 Rallye"   1294  103   189   805   370   157
 "Seat Ibiza SX I"      1461  100   181   925   363   161
 "Citroen AX Sport"     1294   95   184   730   350   160
end
quietly regress vitesse cylindre puissance poids longueur largeur
predict hatvit

predict stf, stdf
gen binf = hatvit - 1.96*stf
gen bsup = hatvit + 1.96*stf
scatter vitesse hatvit || line hatvit vitesse binf bsup, pstyle(p2 p3 p3) sort
```

Question 6: exemples de tests de significativités*

```
test  cylindre puissance poids longueur largeur
test poids
```

test longueur largeur

Question 7: graphique de la vitesse médiane sur le poids*
egen vsigma=sd(vitesse)
egen vmoyenne=mean(vitesse)
gen vitcr=(vitesse-vmoyenne)/vsigma

egen csigma=sd(cylindre)
egen cmoyenne=mean(cylindre)
gen cylcr=(cylindre-cmoyenne)/csigma

egen psigma=sd(puissance)
egen pmoyenne=mean(puissance)
gen puiscr=(puissance-pmoyenne)/psigma

twoway (mband vitcr poids) (mband cylcr poids) (mband puiscr poids)
tw mband vitcr poids
restore

Question 8: lister les bases de données installées dans la bibliothèque de Stata et utilisation de la base auto pour des graphiques*
sysuse dir
sysuse auto, clear
egen msigma=sd(mpg)
egen mmoyenne=mean(mpg)
gen mpgcr=(mpg-mmoyenne)/msigma

egen psigma=sd(price)
egen pmoyenne=mean(price)
gen pricecr=(price-pmoyenne)/psigma
scatter mpgcr weight || scatter pricecr weight

5.9 Traitement des series temporelles : méthode de Box et Jenkins

Exercice N°11

1. Définir l'environnement de travail en changeant de répertoire par défaut
2. Lire la base ip.dta
3. Générer variable temps trimestrielle à partir du premier trimestre de 1963
4. Calculer les autocorrélations totales
5. Calculer les autocorrélations partielles
6. Estimer le modèle arima (0,1,1)
7. Calculer les autocorrélations avec retards
8. Estimer un modèle ARIMA(Ar(p=3),I(d=2) et MA(q=4))

Correction – Exercice N°11

Maaj, proposer pour cet exercice, une correction conforme plan d'agencement des questions posées ???

version 11.2
capture clear
capture log close
global entree = "C:\Manuel Stata\Entrée"
global sortie = "C:\Manuel Stata\Sortie"

clear all

```
set memory 400m
set more off

use "$entree\ip.dta",clear
gen qdate=q(1963q1)+_n-1
tsset qdate, quarterly
/*autocorrelation : détermination de MA(q)*/
ac ip,lags(20)
/*autocorrelation partielle : détermination de AR(p)*/
pac ip,lags(20)
arima ip,arima(0,1,1) det
        predict bj1

/*autocorrelation*/
ac D.ip,lags(20)

/*autocorrelation partielle : détermination de AR(p) de ip-1*/
pac D.ip,lags(20)
arima D2.ip, ar(1/3) ma(1/4)          /*p=3,d=2 et q=4*/
        predict bj2
```

5.10 Traitement des séries temporelles – Les lissages

Exercice N°12

1. Définir l'environnement de travail en changeant de répertoire par défaut
2. Lire la base *sales1.dta* contenant une série de ventes (*sales*)
3. Tracer le graphique des ventes
4. Faire un lissage exponentiel simple de la série avec comme paramètre de lissage *0.4* et faire une prévision de 3 valeurs
5. Illustrer les résultats par un graphique
6. Faire la même chose avec la série action contenue dans la base action.dta avec comme paramètre de lissage *0.419* et comme horizon de prévision 4
7. Lire la base *sales2.dta* et faire un lissage exponentiel double sur la série des ventes (*sales*). Faire une prévision sur 4 périodes. Tracer le graphique des résultats
8. Faire un lissage de Holt et Winter sur les ventes mensuelles de champagnes (série *champ*) contenues dans la base *champ.dta*

Correction – Exercice N°12

Question 1 : l'environnement de travail*

```
version 11.2
capture clear
capture log close
global entree = "C:\Manuel Stata\Entrée"
global sortie = "C:\Manuel Stata\Sortie"

clear all
set memory 400m
set more off
/*lissage exponentiel simple sur les ventes */
```

Question 2: lecture de la base*

```
use "$entree\sales1.dta", clear
```

Question 3: graphique des ventes*
scatter sales t, m(o) c(l)

Question 4: Lissage*
tssmooth exponential sm1=sales, parms(.4) forecast(3)

Question 5: graphique des résultats*
twoway connected sm1 sales t,title("Lissage exponentiel simple et prévision") ytitle(Sales) xtitle(Time)

Question 6: lissage exponentiel simple sur le cours des actions */
use "$entree\action.dta", clear
gen t=_n
tsset t
tssmooth exponential faction=action, parms(.419) forecast(4)
twoway connected faction action t,title("Single Exponential Forecast") ytitle(Action) xtitle(Jour)

Question7: lissage exponentiel double*
use "$entree\sales2.dta", clear
scatter sales t, m(o) c(l)
tssmooth dexponential f2=sales, forecast(4)
twoway connected f2 sales t,title("Lissage exponentiel double et prévision") ytitle(Sales) xtitle(Time)

Question 8 : Lissage de Holt et Winter*
use "$entree\champ.dta",clear
gen mdate=m(1962m1)+_n-1
tsset mdate, monthly

/*autocorrelation : détermination de MA(q)*/
ac champ,lag(25)

/*autocorrelation partielle : détermination de AR(p)*/
pac champ,lag(25)
arima champ,arima(0,1,1) det
predict hw

/*autocorrelation*/
ac D.champ,lag(25)

/*autocorrelation partielle : détermination de AR(p) de champ-1*/
pac D.champ,lag(25)

5.11 Analyses en composantes principales sous Stata

Exercice N°13: Analyse en composantes principales

1. Définir l'environnement de travail et lecture du fichier texte (*acp.txt)* contenant les 4 variables *(nom, a, b, c)*
2. Faire une analyse en composantes principales
3. Récupérer les valeurs propres « *lamda* » à partir de la matrice de résultats
4. Tracer les graphiques suivants :
    a. Le cercle des corrélations
    b. Tracer le graphique des deux premières composantes principales
5. Récupérer les troix premières composantes principales dans les variables *f1, f2 et f3* et tracer le graphique de *f1 et f2*
6. Faire une ACP sur les notes au Bac en utilisant la matrice de corrélation à la place de la matrice de variance-covariance

Correction – Exercice N°13

Question 1 : l'environnement de travail*
version 11.2
capture clear
capture log close
global entree = "C:\Manuel Stata\Entrée"
infile str3 nom a b c using "$entree\acp.txt"

Question 2 : Analyse en composantes principales*
pca a b c, comp(3)covariance
*Question 3 : valeurs propres *
matrix list e(L)
matrix VP= e(L)
svmat VP, names(lamda)
list a b c lamda* in 1/3
gen n=e(N)

Question 4.a: Cercle des corrélations*
loadingplot, component(2)

Question 4.b: graphique des composantes principales*
scoreplot, component(2) mlabel(nom)

Question 5 : Récupération des composantes principales et graphique*
predict f1 f2 f3
scatter f2 f1, xline(0) yline(0) mlabel(nom)

Question 6 : ACP sur les notes au Bac en utilisant la matrice de corrélation*

clear

```
input str3   nom    math     physique     philo   anglais     français
             E1      10          7          11       16          17
             E2       8         16           6        9           7
             E3      10         14          11       17          15
             E4       6         10           6       10          10
             E5       2          9           6       13           9
             E6      17         16           5       14          15
             E7       9         12           9       14          10
             E8       6         13          13       16          18
             E9      11         12           5       10          14
             E10      5          6          11       14          11
             E11      7          6           8       14           9
             E12     10         13           6        4          10
             E13      8         11          10       12           6
             E14     10         13           3       15          11
             E15     12          9           7       12           5
             E16     10         11          10       14          12
             E17     11         15           7       14          10
             E18     13          6           7       10           6
             E19      7         10          10       10          11
             E20     15         11           7       17           9
             E21     11         13          14       13          11
             E22      8          9           9        8           5
             E23     10         12          12        9           6
             E24      9          8           5       12          13
             E25     13         16          12        6           8
             E26     17         13          13       13          15
end
```

pca math physique philo anglais français, comp(5)

Question 3 : valeurs propres *
matrix list e(L)
matrix VP= e(L)
svmat VP, names(lamda)
list math physique philo anglais français lamda* in 1/5

gen n=e(N)

Question 4.a: Cercle des corrélations*
loadingplot, component(2)

Question 4.b : graphique des composantes principales*
scoreplot, component(2) mlabel(nom)

Question 5 : Récupération des composantes principales et graphique*
predict f1 f2 f3 f4 f5
scatter f2 f1, xline(0) yline(0) mlabel(nom)

5.12 Analyse factorielle des correspondances (AFC)

Exercice N°14

1. Lecture des données représentant les résultats de l'élection présidentielle au premier tour en 1981 par département (tableau de contingence)
2. Faire une analyse factorielle des correspondances
3. Récupérer les valeurs propres « *lamda* » à partir de la matrice de résultats
4. Tracer les graphiques suivants :
    a. Le cercle des corrélations
    b. Tracer le graphique des deux premiers axes factoriels
5. Récupérer les trois premiers axes factoriels dans les variables *f1, f2 et f3* et tracer le graphique de *f1 et f2*

Correction – Exercice N°14

Question 1 : lecture des données
clear

input str20 dep	mit	vge	chirac	marchais	lalonde	crepeau	arlette	debre	garaud	bouchardeau	total
Ain	51	64	36	23	9	5	4	4	3	3	202
Hautes-Alpes	14	17	9	9	3	1	2	1	1	1	58
Ariège	27	18	13	17	2	2	2	1	1	1	84
Bouches-du-Rhône	191	204	119	205	29	13	13	10	10	6	800
Charente-Maritime	71	76	47	37	8	34	5	4	4	2	288
Côtes-du-Nord	93	90	57	54	13	5	9	4	3	5	333
Drôme	57	55	31	30	10	4	5	4	3	3	202
Finistère	132	149	95	49	21	9	11	6	5	10	487
Gironde	195	137	98	83	20	16	13	13	8	5	588
Indre	34	39	28	28	4	3	4	3	2	1	146
Landes	62	47	31	26	5	3	3	3	2	1	183
Loire-Atlantique	149	156	94	49	23	15	13	10	7	8	524
Lozère	10	18	9	4	2	0	1	1	1	1	47
Haute-Marne	32	33	20	15	4	2	3	2	2	1	114
Morbihan	86	117	65	33	14	6	8	5	4	4	342
Oise	87	88	59	62	13	7	10	6	5	3	340
Pyrénées-Atlantiques	90	91	66	33	12	6	6	5	4	3	316
Haut-Rhin	75	125	58	19	17	6	8	6	6	4	324
Sarthe	72	87	49	40	10	6	8	4	3	3	282
Seine-Maritime	171	181	91	123	24	13	18	10	7	6	644
Deux-Sèvres	54	66	34	16	8	7	5	3	4	2	199
Val-d'Oise	111	100	74	81	22	12	10	7	7	6	430
Vendée	61	105	59	19	10	11	5	5	4	3	282
Yonne	44	52	31	24	7	4	4	3	3	2	174

end

Question 2 : AFC*factor mit vge chirac marchais lalonde crepeau arlette debre garaud bouchardeau, comp(5)

Question 3: valeurs propres
matrix list e(L)
matrix VP= e(L)
svmat VP, names(lamda)
l mit vge chirac marchais lalonde crepeau arlette debre garaud bouchardeau lamda* in 1/5
gen n=e(N)

Question 4.a: graphique du Cercle des corrélations
loadingplot, component(2)

Questions 4.b : graphique des axes factoriels 1 et 2
scoreplot, component(2) mlabel(dep)

Question 5 : Récupération des axes factoriels et graphique
predict f1 f2 f3 f4 f5
scatter f2 f1, xline(0) yline(0) mlabel(dep)

VI. Eléments bibliographiques

- Manuel Stata 12
- A. Bozio (2005), « Introduction au logiciel Stata »
- O. Cadot (2008) « Stata pour les nuls » : http://www.hec.unil.ch/ocadot/SECODEV_2008/Tools/Stata_nuls.pdf
- N. Couderc « Econométrie appliquée avec Stata»
- C. Normand, A. Robillard (2006) « Introduction au logiciel Stata »
- Michel Tenenhaus (1994) Méthodes statistiques en gestion, Dunod Entreprise