Introduction aux sections 4 et 5

Dans la **section 3**, nous nous sommes intéressés à l'estimation de la moyenne, ainsi que de l'erreur type, dont les formules sont exactes. Il s'agissait d'estimations ponctuelles dans le cas univarié. Dans ce cas, il est assez intuitif d'étudier la précision de la statistique estimant le paramètre de la loi d'une variable. D'autant plus que nous connaissions la (super)population !

Quand les fonctions des variables sont plus compliquées, que les lois suivies par les statistiques ne sont connues, ni exactement, ni asymptotiquement, ou que la supposition d'indépendance ne tient pas, on peut avoir recours à d'autres méthodes d'estimation avec Stata.

L'estimation dépend largement de la nature des variables (dichotomique, discrète, continue), et de la spécification du modèle, c'est-à-dire de la forme des relations entre ces variables (log-linéaire, autorégressive, système d'équations, etc.)

C'est la raison pour laquelle nous verrons la spécification des modèles en même temps que les méthodes d'estimation dans la **section 4**. Nous verrons dans la **section 5** comment obtenir quelques statistiques de test.

On se place dans le cadre du modèle d'échantillonnage.

4. Méthodes d'estimation

L'estimation peut porter sur le paramètre d'une loi. La méthode du maximum de vraissemblance (MV) est appropriée dans ce cas. On peut aussi estimer un paramètre sans être obligé de supposer une loi, avec, par exemple, la méthode des moindres carrés, inventée par Gauss.[1] Nous verrons des problèmes d'estimation ponctuelle dans le cas multivarié, et d'estimation par intervalle de confiance, notamment l'estimation bootstrap d'un paramètre, qui fonctionne dans les cas univarié et multivarié. Il y a aussi la méthode des moments, qui est disponible grâce à la commande `gmm`, où 'gmm' est l'abréviation de *general method of moments*. Nous ferons un petit exercice, afin de donner une intuition de la méthode, que vous verrez plus amplement dans d'autres cours, notamment l'an prochain.

4.1 Maximum de vraisemblance, `ml`

La commande `ml` est puissante lorsqu'il s'agit d'estimer un paramètre difficilement calculable à la main. La méthode consiste à trouver la forme de la statistique qui maximise une fonction des observations, la $\boxed{\text{fonction de vraisemblance}}$. Un exemple emblématique est celui de la $\boxed{\text{loi logistique}}$. La méthode est très efficace lorsque le modèle n'est pas linéaire. Considérons par exemple le cas d'une variable Y dichotomique, qui suit non pas une loi de Bernoulli, mais une loi logistique à valeur dans $\{0, 1\}$. Nous rentrons doucement dans le cadre multivarié, en supposant que $\Pr(Y = 1|X = x)$ est la fonction de répartition cumulée logistique :

$$\Pr(Y = 1|X = x) = \frac{e^{\beta x}}{1 + e^{\beta x}}.$$

$F(x, \beta)$ ne comporte qu'un paramètre, mais étant non-linéaire, la vraisemblance sera aussi compliquée que si nous avions considéré $Y \sim N(\cdot, \cdot)$. Comme on peut le voir, la probabilité que Y se réalise est une fonction croissante de x et tend vers 0 quand x tend vers $-\infty$. Et, $\ln(P/(1 - P)) = \beta x$ est une fonction linéaire de β. Le ratio $P/(1-P)$ s'appelle $\boxed{\text{ratio de chance}}$, et le logarithme (népérien) de ce ratio s'appelle $\boxed{\text{fonction logit}}$. Cette fonction intervient lors de la recherche de l'estimateur du MV.

Supposons un échantillon aléatoire. La loi jointe de $\boldsymbol{y}$ se note généralement $L(\boldsymbol{y}|\beta)$. L'écriture de la fonction de vraissemblance de l'échantillon "renverse" le conditionnement, $L(\beta|\boldsymbol{y})$ pour indiquer que c'est β qui varie, $\boldsymbol{y}$ est donné. Le problème est

[1] Nous ne verrons pas l'approche bayésienne de l'estimation d'un paramètre. Avec cette approche, on s'enfonce un peu plus dans l'estimation paramétrique, avec des hypothèses *a priori* sur ces paramètres. Par exemple, μ, le moment centré d'ordre un d'une variable normale de paramètres μ et σ^2, est lui-même borné entre a et b selon la loi $U[a, b]$. On peut facilement produire des valeurs pour ce type de variable avec Stata, en suivant la méthode Monte Carlo, ce qui est plus facile que de faire le calcul analytique. Vous pouvez consulter le `Stata Bayesian Analysis Reference Manual`.

alors le suivant :

$$\hat{\beta} = \text{argmax}_\beta\{L(\beta|\boldsymbol{y})\}.$$

Notons $P(Y_i = 1|X_i = x_i) \equiv p_i(x_i)$. La vraissemblance s'écrit :

$$L(\beta|\boldsymbol{y}) = \Pi_{i=1}^n p_i(x_i)^{y_i}(1 - p_i(x_i))^{1-y_i}.$$

Faisons un petit détour par la loi de Bernoulli. Pour cela, supposons que p_i ne varie pas avec i ($x_i \equiv 1\ \forall i$ par exemple). Dans ce cas, on peut écrire $e^{\beta x_i}/(1+e^{\beta x_i}) = e^\beta/(1 + e^\beta) \equiv p$. Autrement dit, si je connais β, je connais p, est *vice-versa.* Que devient la vraissemblance dans ce cas ?

Depuis Fisher, inventeur de la méthode, on s'intéresse à la log-vraissemblance, $\ln(L(p|\boldsymbol{y}))$, que l'on note $l(p|\boldsymbol{y}))$. Nous avons :

$$\begin{aligned} l(p|\boldsymbol{y}) &= \textstyle\sum_{i=1}^n (y_i \ln(p) + (1 - y_i)\ln(1 - p) \\ &= \ln(p)n\bar{y} + \ln(1 - p)n(1 - \bar{y}). \end{aligned}$$

L'estimateur du MV est simple dans ce cas, $\hat{p} = \bar{y}$.

On peut utiliser la commande **ml**, que l'on peut appliquer à l'estimation du paramètre d'une variable aléatoire qui suit une loi logistique, de Poisson. On peut très bien utiliser la commande `logit`, mais aussi `ml` , ou encore utiliser `mata` pour aller encore plus dans les aspects techniques (nous avons fait un petit programme qui résout les équations non-linéaires induites par la maximisation de la vraisemblance).

```
.  cls

. clear   all

. use            "http://www.stata-press.com/data/r13/repair", clear
(1978 Automobile Data)

.  set more off

.  keep           foreign repair

.  tabulate       foreign repair, cell

+-----------------+
| Key             |
|-----------------|
|    frequency    |
| cell percentage |
+-----------------+

           |              repair
  Car type |         1          2          3 |     Total
-----------+---------------------------------+----------
  Domestic |        10         27          9 |        46
           |     17.24      46.55      15.52 |     79.31
-----------+---------------------------------+----------
   Foreign |         0          3          9 |        12
           |      0.00       5.17      15.52 |     20.69
-----------+---------------------------------+----------
     Total |        10         30         18 |        58
           |     17.24      51.72      31.03 |    100.00
```

```
. logit           foreign repair

Iteration 0:   log likelihood = -29.569311
Iteration 1:   log likelihood = -23.253386
Iteration 2:   log likelihood = -22.346332
Iteration 3:   log likelihood = -22.341067
Iteration 4:   log likelihood = -22.341067

Logistic regression                               Number of obs   =         58
                                                  LR chi2(1)      =      14.46
                                                  Prob > chi2     =     0.0001
Log likelihood = -22.341067                       Pseudo R2       =     0.2445

------------------------------------------------------------------------------
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      repair |   2.298023   .7274809     3.16   0.002     .8721868     3.72386
       _cons |  -6.871362   1.939831    -3.54   0.000    -10.67336   -3.069364
------------------------------------------------------------------------------

. logit           foreign repair, or

Iteration 0:   log likelihood = -29.569311
Iteration 1:   log likelihood = -23.253386
Iteration 2:   log likelihood = -22.346332
Iteration 3:   log likelihood = -22.341067
Iteration 4:   log likelihood = -22.341067

Logistic regression                               Number of obs   =         58
                                                  LR chi2(1)      =      14.46
                                                  Prob > chi2     =     0.0001
Log likelihood = -22.341067                       Pseudo R2       =     0.2445

------------------------------------------------------------------------------
     foreign | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      repair |   9.954486   7.241699     3.16   0.002     2.392136    41.42397
       _cons |   .0010371   .0020117    -3.54   0.000     .0000232    .0464507
------------------------------------------------------------------------------

. margins         , dydx(repair) at(repair==2)

Conditional marginal effects                      Number of obs   =         58
Model VCE    : OIM

Expression   : Pr(foreign), predict()
dy/dx w.r.t. : repair
at           : repair          =           2

------------------------------------------------------------------------------
             |            Delta-method
             |      dy/dx   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      repair |   .1941922   .0598857     3.24   0.001     .0768184     .311566
------------------------------------------------------------------------------


.  predict         FOREIGNP, pr

. sort           repair FOREIGNP
```

On peut reestimer le logit avec la troisieme valeur de la variable **repair** en groupe de base

```
.  logit foreign ib3.repair, or baselevels

note: 1.repair != 0 predicts failure perfectly
      1.repair dropped and 10 obs not used

Iteration 0:   log likelihood = -26.992087
Iteration 1:   log likelihood = -22.483187
Iteration 2:   log likelihood = -22.230498
Iteration 3:   log likelihood = -22.229139
Iteration 4:   log likelihood = -22.229138

Logistic regression                               Number of obs   =         48
                                                  LR chi2(1)      =       9.53
                                                  Prob > chi2     =     0.0020
Log likelihood = -22.229138                       Pseudo R2       =     0.1765

------------------------------------------------------------------------------
     foreign | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      repair |
          1  |          1  (empty)
          2  |   .1111111   .0855334    -2.85   0.004     .0245755    .5023573
          3  |          1  (base)
             |
       _cons |          1   .4714045    -0.00   1.000     .3969536    2.519186
------------------------------------------------------------------------------

.  tabulate repair, generate(repair_d)

     repair |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |         10       17.24       17.24
          2 |         30       51.72       68.97
          3 |         18       31.03      100.00
------------+-----------------------------------
      Total |         58      100.00

.  logit foreign repair_d1 repair_d2, or

note: repair_d1 != 0 predicts failure perfectly
      repair_d1 dropped and 10 obs not used

Iteration 0:   log likelihood = -26.992087
Iteration 1:   log likelihood = -22.483187
Iteration 2:   log likelihood = -22.230498
Iteration 3:   log likelihood = -22.229139
Iteration 4:   log likelihood = -22.229138

Logistic regression                               Number of obs   =         48
                                                  LR chi2(1)      =       9.53
                                                  Prob > chi2     =     0.0020
Log likelihood = -22.229138                       Pseudo R2       =     0.1765

------------------------------------------------------------------------------
     foreign | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   repair_d1 |          1  (omitted)
   repair_d2 |   .1111111   .0855334    -2.85   0.004     .0245755    .5023573
       _cons |          1   .4714045    -0.00   1.000     .3969536    2.519186
------------------------------------------------------------------------------

.  program MYLOGIT
.  version 13
.  args lnf THETA
.  quietly: replace `lnf' = -ln(1+exp(-`THETA')) if $ML_y1==1
.  quietly: replace `lnf' = -`THETA' - ln(1+exp(-`THETA')) if $ML_y1==0
. end

. ml model lf MYLOGIT (foreign = repair)
```

```
. ml maximize

initial:       log likelihood = -40.202536
alternative:   log likelihood = -33.496465
rescale:       log likelihood = -30.169178
Iteration 0:   log likelihood = -30.169178
Iteration 1:   log likelihood = -22.986181
Iteration 2:   log likelihood = -22.349011
Iteration 3:   log likelihood = -22.341073
Iteration 4:   log likelihood = -22.341067
Iteration 5:   log likelihood = -22.341067

                                                  Number of obs   =         58
                                                  Wald chi2(1)    =       9.98
Log likelihood = -22.341067                       Prob > chi2     =     0.0016

------------------------------------------------------------------------------
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      repair |   2.298023    .727481     3.16   0.002     .8721868     3.72386
       _cons |  -6.871362   1.939831    -3.54   0.000    -10.67336   -3.069364
------------------------------------------------------------------------------
```

Nous allons maintenant résoudre le systeme de deux equations non-lineaires en β_1 et β_2 dans `MATA`.

```
.  table foreign FOREIGNP if repair==1, cellwidth(10)

----------------------
          |Pr(foreign)
 Car type |   .0102179
----------+-----------
 Domestic |         10
----------------------

. table foreign FOREIGNP if repair==2, cellwidth(10)

----------------------
          |Pr(foreign)
 Car type |    .093188
----------+-----------
 Domestic |         27
  Foreign |          3
----------------------

. table foreign FOREIGNP if repair==3, cellwidth(10)

----------------------
          |Pr(foreign)
 Car type |   .5056766
----------+-----------
 Domestic |          9
  Foreign |          9
----------------------

.  mata
------------------------------------------------- mata (type end to exit) ----------------------------------------------------------
:  void function myfun2(real colvector x, real colvector values)  {  values[1] =   -10*exp(x[1]+x[2])/(1+exp(x[1]+x[2])) - ///

:  S = solvenl_init()
```

```
:  solvenl_init_evaluator(S, &myfun2())


:  solvenl_init_type(S, "zero")


:  solvenl_init_technique(S, "newton")


:  solvenl_init_numeq(S, 2)


:  solvenl_init_startingvals(S, J(2,1,0))


:  solvenl_init_iter_log(S, "on")


:  x = solvenl_solve(S)
Iteration 0:  function =      1130
Iteration 1:  function = 28.394091  delta X = 3.5076214
Iteration 2:  function = 2.8026749  delta X = .47183187
Iteration 3:  function = .09205889  delta X =  .1565965
Iteration 4:  function = .00009591  delta X = .02503346
Iteration 5:  function = 8.894e-11  delta X = .00074643

invalid expression
r(3000);

:  x
                  1
    +----------------+
  1 |  -6.871356767  |
  2 |   2.298021399  |
    +----------------+

:  end
------------------------------------------------------------------------------------------------------------------------------------
```

Notons que nous pourrions estimer le modèle Probit de la même manière, ainsi que d'autres modèles à variable dépendante qualitative, tels que le modèle de Poisson, ou une généralisation de la fonction logistique, avec le modèle multinomial.

4.2 Moindres carrés, `regress`

Dans le cas univarié, la méthode des moindres carrés (MC) permet de trouver une estimation du moment centré d'ordre un d'une variable qui minimise sa variance :

$$\mu^{MC} \equiv argmin_{\mu} n^{-1} \sum_i (y_i - \mu)^2.$$

On la retrouve en économétrie où la moyenne est en fait l'espérance conditionnelle à une ou plusieurs autre variables, $\theta \equiv E(Y_i|X_1, X_2, \ldots, X_K) = \beta_1 X_1 + \cdots \beta_K X_K$.

La commande `regress` de Stata gère ces deux situations sans problème. Dans le premier cas, il suffit de faire une régression des de Y sur une constante, en tapant tout simplement `regress Y.` Dans le second cas, une régression sur les différentes variables, par exemple X_1 et X_2 :

`regress Y X1 X2` $\cdots$ `X3,`

Les justifications de la méthode sont dans tous les livres d'économétrie, dont les deux plus fameux (Wooldridge, 2010, et Greene, 2014). On peut sans difficulté, utiliser Stata pour calculer les coefficients d'un modèle en manipulant les matrices.

```
.  use     "http://www.evens-salies.com/1977_Rubin.dta", clear

. rename (GROUP POSY PREX)(D Y X)


. regress Y D

      Source |       SS       df       MS              Number of obs =      72
-------------+------------------------------           F(  1,    70) =   18.08
       Model |  170.057317     1  170.057317           Prob > F      =  0.0001
    Residual |  658.262128    70  9.40374468           R-squared     =  0.2053
-------------+------------------------------           Adj R-squared =  0.1940
       Total |  828.319444    71   11.666471           Root MSE      =  3.0666

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           D |   3.228085   .7590978     4.25   0.000     1.714112    4.742058
       _cons |   6.531915   .4473027    14.60   0.000     5.639798    7.424032
------------------------------------------------------------------------------


. generate ONE=1

. mkmat ONE D, matrix(X)


. mkmat Y, matrix(Y)


. matrix B=invsym(X'*X)*X'*Y

. matrix list B

B[2,1]
             Y
```

```
ONE  6.5319149
  D  3.2280851
```

Dans le cas du modèle linéaire simple, il y a un lien fort avec le coefficient de corrélation de pearson dans le cas de variables continues. Les résultats ne renvoient pas les coefficients de correlation de Spearman, tetrachorique (dans le cas de deux variables dichotomiques) car, par définition, la regression est pour des variables continues. La statistique R^2 affichée, après avoir exécuté la commande, est le coefficient de Pearson au carré.

4.3 Estimation bootstrap

Le bootstrap, que nous avions vu dans la **section 3** est au départ une méthode d'inférence. Revenons à l'exemple des capitalisations de cette section. Chacune des 23 capitalisations est la réalisation d'une variable aléatoire dont on ne connaît pas la loi sous-jacente. Dans ce cas, le $\boxed{\text{bootstrap non-paramétrique}}$ peut être utilisé, en appliquant un re-échantillonnage basé sur toutes les observations. Si $n = 23$ est la taille de l'échantillon, on crée un certains nombre d'échantillon bootstrap de 23 observations chacun. Rappelons que le nombre d'échantillons bootstrap <u>distincts</u> augmente très vite avec n.

Avant de continuer avec ces données concrètes, examinons le cas simpliste $N = n = 3$. Nous allons introduire la commande `bootstrap.` dans ce cas. Nous allons produire 1000 échantillons bootstrap, sachant très bien qu'il n'y en a que 10 de distincts. En effet, à partir de $\mathcal{S} = \{1, 2, 3\}$, on a les échantillons suivants : {1,1,1}, {2,2,2}, {3,3,3}, {1,1,2}, {2,2,1}, {1,1,3}, {3,3,1}, {2,2,3}, {3,3,2}, {1,2,3}. Supposons que les valeurs de la variable aléatoire soient $y_1 = 2$, $y_2 = 1$ et $y_3 = 3$. Il y a sept sommes, et donc sept moyennes différents. Ces 10 échantillons donnent les moyennes suivantes : $\frac{2+2+2}{3} = 2$, $\frac{1+1+1}{3} = 1$, ..., $\frac{3+3+1}{3} = 7/3$, $\frac{2+1+3}{3} = 2$. Vous pourrez vérifier qu'il y a sept moyennes différentes. Pour connaître la proportion de chaque moyenne, il faut revenir au nombre d'échantillons possibles : $3^3 = 27$. Les différentes moyennes avec leur fréquence relative théorique entre parenthèses, sont les suivantes : $3/3 = 1(1/27)$ (l'échantillon {1,1,1}), $4/3 = 1,33\ldots(3/27)$ (les échantillons {1,2,2}, {2,1,2}, {2,2,1}), $5/3 = 1,66\ldots(6/27)$ (les échantillons {1,1,2}, {1,2,1}, {2,1,1}, {2,2,3}, {2,3,2}, {3,2,2}), ..., $9/3 = 3(1/27)$ (l'échantillon {3,3,3}).

```
.  set seed        21041971

. clear all

. set     obs     3
obs was 0, now 3

. input int Y

             Y
  1. 2
. 1
. 3

. cls

. list

     +---+
     | Y |
     |---|
  1. | 2 |
  2. | 1 |
  3. | 3 |
     +---+

. generate        MEAN=.
```

```
(3 missing values generated)

. bootstrap       MEAN=r(mean), ///
  size(3) reps(1000) saving(bootstrap123, replace) nowarn: summarize Y

(running summarize on estimation sample)

Bootstrap replications (1000)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
..................................................   250
..................................................   300
..................................................   350
..................................................   400
..................................................   450
..................................................   500
..................................................   550
..................................................   600
..................................................   650
..................................................   700
..................................................   750
..................................................   800
..................................................   850
..................................................   900
..................................................   950
..................................................  1000

Bootstrap results                               Number of obs      =         3
                                                Replications       =      1000

      command:  summarize Y
         MEAN:  r(mean)

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        MEAN |          2   .4705071     4.25   0.000     1.077823    2.922177
------------------------------------------------------------------------------

.  use                    "C:\Users\evens\Documents\bootstrap123.dta", clear
(bootstrap: summarize)

. generate               ONE=1

. collapse (sum)  ONE, by(MEAN)

. summarize              ONE

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         ONE |         7    142.8571    93.38171         30        248

. generate               ONEP=100*ONE/r(sum)

. browse
```

La commande `bootstrap` a pour argument le nom de la variable dans laquelle on va mettre la statistique qui nous intéresse. Ici, c'est la variable `MEAN`. Cette variable contiendra une statistique post-commande, `r(mean),` calculée par `summarize`.

Il y a 1000 TAR, et les 1000 moyennes sont placées dans le fichier de données `bootstrap123.dta` dans votre dossier de travail (taper `pwd` dans la fenêtre de commande pour savoir quel est ce dossier). Après la commande `bootstrap,` le programme calcule la fréquence relative de chaque moyenne. La commande calcule un intervalle de confiance bootstrap de la moyenne, à partir de l'erreur standard bootstrap.

Le programme suivant calcule cet intervalle de confiance sur les données de capitalisation en ré-échantillonnant les 23 observations. Il y a trop d'échantillons possibles (non-distincts) : 23^{23}. Nous allons nous contenter d'en tirer 100. Vous pouvez ensuite essayer de voir quel est le résultat obtenu avec 500 réplications. Plutôt que de placer les fréquences relatives des différentes moyennes dans la feuille de données, on va faire un histogramme de ces moyennes, avec la commande `hist MEAN`.

```
.  use     "http://www.evens-salies.com/statainitiation_3_capitalisation.dta", clear

. rename         (var*)(CAP STR TEMP)


. encode         TEMP, generate(NOM)

. drop           TEMP

. set seed       21041971

. bootstrap      MEAN=r(mean), size(23) reps(1000) ///
  saving(bootstrap_capitalisation, replace) nowarn mse: summarize CAP

(running summarize on estimation sample)

Bootstrap replications (1000)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
..................................................   250
..................................................   300
..................................................   350
..................................................   400
..................................................   450
..................................................   500
..................................................   550
..................................................   600
..................................................   650
..................................................   700
..................................................   750
..................................................   800
..................................................   850
..................................................   900
..................................................   950
..................................................  1000

Bootstrap results                               Number of obs      =        23
                                                Replications       =      1000

      command:  summarize CAP
         MEAN:  r(mean)

------------------------------------------------------------------------------
             |   Observed   Bstrap *
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
```

```
-------------+----------------------------------------------------------------
        MEAN |   254.9565     59.677     4.27   0.000     137.9918    371.9213
------------------------------------------------------------------------------

.  preserve

.  use            "bootstrap_capitalisation.dta", clear
(bootstrap: summarize)

.  hist           MEAN
(bin=29, start=98.826088, width=11.776612)


. restore
```

Nous n'avons pas utilisé l'information capturée par la variable de stratification. Si l'on pense que la moyenne varie d'une strate à l'autre, de sorte que $Y_{is} = \mu_s + U_i$ est le modèle pertinant, avec par exemple $U_i \sim i.i.d.$ (la valeur de Y_{is} est déterminée par un PGD inconnu).

On peut combiner `regress` avec l'option `bootstrap` pour l'estimation de l'érreur type du coefficient d'un modèle de régression. Dans l'exemple ci-dessous, le coefficient est unique, c'est la constante dans une modèle de régression (il n'y a pas de variable explicative autre que la constante). On sait que dans ce cas, l'estimateur de la constante est $\hat{Y}$ de la **section 3**.

```
.  use     "http://www.evens-salies.com/statainitiation_3_capitalisation.dta", clear

. rename                 (var*)(CAP STR TEMP)


. encode                 TEMP, generate(NOM)

. drop                   TEMP

. order                  NOM

. sort                   NOM

. save                   "statainitiation_3_capitalisation_final.dta", replace
file statainitiation_3_capitalisation_final.dta saved

. set seed               21041971

.  regress                CAP, vce(bootstrap)     // Dans ce cas, s.e. sert a construire IC
(running regress on estimation sample)

Bootstrap replications (50)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
. ................................................    50

Linear regression                               Number of obs      =        23
                                                Replications       =        50
                                                Wald chi2(0)       =         .
                                                Prob > chi2        =         .
                                                R-squared          =    0.0000
                                                Adj R-squared      =    0.0000
                                                Root MSE           =  299.8770

------------------------------------------------------------------------------
```

```
             |   Observed   Bootstrap                         Normal-based
         CAP |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |   254.9565   57.97154     4.40   0.000     141.3344    368.5787
------------------------------------------------------------------------------

.  matrix list     e(b)

symmetric e(b)[1,1]
        _cons
y1  254.95652

. matrix list     e(V)                    // S^2/23

symmetric e(V)[1,1]
           _cons
_cons  3360.6998

. matrix define   B=J(1,1,0)

. matrix define   V=J(1,1,0)

. matrix define   B[1,1]=e(b)

. matrix define   V[1,1]=e(V)

. local                   B=B[1,1]

. local                   SE=V[1,1]^.5

. local                   ICL=`B'-invnormal(0.975)*`SE'

. di                              `ICL'
141.33439
```

On peut vérifier avec `summarize CAP` que l'unique coefficient estimé est bien la moyenne de la variable. L'avantage de `regress` est que nous avons au passage une estimation d'un intervalle de confiance pour la moyenne. Dans le cas d'un re-échantillonnage, l'estimateur bootstrap de l'erreur standard, qui vaut 57.97 dans mon cas, sert à calculer l'intervalle de confiance au seuil de 5%, comme on peut le vérifier :

$$254.95 \pm 1,96 \times 57,97 \Leftrightarrow 254.95 \pm 113,62 \Leftrightarrow [141,33; 368,57].$$

Vous verrez des méthodes plus compliquées cett année ou l'an prochain. Je pense par exemple à la méthode des moments généralisés, avec la commande `gmm`. Je vous conseille de revoir cette méthode dans le cas univarié, le cas "simple" par opposition à "généralisé" qui est le cadre dans lequel cette commande a été créée). Il y a un estimateur connu en économétrie qui s'appuie dessus. C'est celui d'Arellano et Bond, pour l'estimation des coefficients d'un modèle dynamique sur données de panel. Des informations sur l'application de la méthode dans le cas de petits échantillons sont disponibles dans Drukker (2010).[2]

[2] Drukker, D.M. 2010. An introduction to GMM estimation using Stata (slides). In German Stata Users' Group.

5. Test

La méthode du Maximum de Vraisemblance débouche naturellement sur le test du ratio de vraisemblance.[3] Nous nous intéressons principalement aux tests de spécification. Les tests de diagnostic (normalité, autocorrélation, ...) seront simplement discutés. L'application suivante vise à tester la nullité du paramètre de centralité d'une variable normalement distribuée.

5.1 Test du ratio de vraissemblance `lrtest`

Après avoir estimé un modèle en maximisant la vraisemblance, on peut utiliser le test du ratio de vraisemblance. La statistique de test suit un χ^2. La démonstration est relativement facile.

[Exposer la théorie]

```
.  cls

. clear all

. set obs 1000
obs was 0, now 1000

. set seed 3102019

.  generate Y=rnormal(0,1)

.  summarize Y

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
           Y |      1000    -.054438    1.000331  -2.880103   2.623836

. display r(N)*(r(mean)/r(sd))^2
2.9615405

.  mlexp (-ln({sigma})-(0.5/{sigma}^2)*Y^2)
initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -1312.1025
rescale:       log likelihood = -501.31242
Iteration 0:   log likelihood = -501.31242
Iteration 1:   log likelihood =  -501.3107
Iteration 2:   log likelihood =  -501.3107

Maximum likelihood estimation

Log likelihood =  -501.3107                     Number of obs   =       1000

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      /sigma |   1.001312     .02239    44.72   0.000      .957428    1.045195
------------------------------------------------------------------------------
```

[3]Il existe également un test du multiplicateur de Lagrange, que nous ne verrons pas dans cette section.

```
. estimate store M0

. mlexp (-ln({sigma})-(0.5/{sigma}^2)*(Y-{mu})^2)
initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -1920.9786
rescale:       log likelihood = -1055.7505
rescale eq:    log likelihood = -499.86316
Iteration 0:   log likelihood = -499.86316
Iteration 1:   log likelihood = -499.83064
Iteration 2:   log likelihood = -499.83064

Maximum likelihood estimation

Log likelihood = -499.83064                     Number of obs   =      1000

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      /sigma |   .9998307   .0223569    44.72   0.000      .956012    1.043649
         /mu |  -.054438   .0316174    -1.72   0.085    -.1164071     .007531
------------------------------------------------------------------------------

. estimate store M1

. lrtest M0 M1

Likelihood-ratio test                                 LR chi2(1)  =      2.96
(Assumption: M0 nested in M1)                         Prob > chi2 =    0.0853
```

L'interprétation des résultats est la suivante. Nous trouvons 2,96 pour la valeur de la statistique (Stata calcule aussi la p-valeur du test, et trouve 0,085, qui est donc la probabilité qu'une variable suivant une loi du χ^2 à 1 degré de liberté soit plus grande que 2,96). En conclusion, au seuil de 5%, 0,085 est trop grand, aussi Stata dit que M0 est contenu dans M1 (M0 est un sous-modèle de M1). Par conséquent, je ne rejette pas H0 au seuil de 5%. La vraisemblance avec M1 (l_1) ne sera pas plus grande que celle avec M0 (l_0) puisque M1 est le modèle 'faux', étant donné les observations disponibles, bien qu'il ne soit pas si mauvais (en effet, une p-valeur de 8,5% n'est pas si mauvais). Si nous avions opté pour un seuil de 10%, nous aurions rejeté H0 en faveur de H1 (l modèle non-contraint).

5.2 Test de Wald `ttest`

La commande `ttest` est très facile à utiliser sur une variable, pour tester une hypothèse sur sa moyenne. Mais aussi sur l'égalité des moyennes de deux variables. Ces tests, qui figurent dans tous les livres de statistiques, sont simplement décrits dans le petit dictionnaire de Nelson (2004, p. 95), qui s'avère très utile lorsque l'on veut s'informer vite et bien sur des méthodes statistiques.[4]

La famille des tests de Wald revient à imposer une contrainte linéaire dans un modèle de régression. Lorsqu'il s'agit d'une hypothèse simple, la statistique de test est celle de Student, et lorsqu'il s'agit d'une hypothèse composite (au moins deux coefficients), c'est la statistique de Fisher. Nous allons nous pencher sur le test d'égalité de moyennes, problème que l'on appelle de Bierens-Fisher. En général, on ne teste pas l'égalité des moyennes de deux sous-populations sans se poser la question de légalité des variances.

```
.  use "http://www.evens-salies.com/urgence.dta", clear

. table phstat phospyr

----------------------------------------------------------
FHS.500_00 |
. 000:      |
Reported   |     FAU.060_00.000: Was - - in a hospital
health     |                OVERNIGHT, 12m
status     |        Yes          No     Refused  Don't know
-----------+----------------------------------------------
 Excellent |      1,512      32,175          53          14
 Very good |      1,764      28,008         133          33
      Good |      2,195      22,568         192          54
      Fair |      1,476       5,822          26          12
      Poor |        827       1,476          11           8
   Refused |          8         118          76           8
Don't know |         11          59           2           8
----------------------------------------------------------

. keep if phstat<=5 & phospy<=2
(826 observations deleted)

. table phstat phospyr

--------------------------
FHS.500_0 |FAU.060_00.000:
0.000:    |  Was - - in a
Reported  |    hospital
health    | OVERNIGHT, 12m
status    |    Yes      No
----------+---------------
Excellent |  1,512  32,175
Very good |  1,764  28,008
     Good |  2,195  22,568
     Fair |  1,476   5,822
     Poor |    827   1,476
--------------------------

. generate HEALTH=6-phstat

. generate GROUP=phospy
```

[4] Nelson, D. 2004. *The Penguin Dictionary of Statistics.* Penguin Books.

```
. replace GROUP=0 if GROUP==2
(90049 real changes made)

. by GROUP, sort: summarize HEALTH

-----------------------------------------------------------------------------------------------------------------------------------
-> GROUP = 0

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      HEALTH |     90049    3.928206    1.004448          1          5

-----------------------------------------------------------------------------------------------------------------------------------
-> GROUP = 1

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      HEALTH |      7774    3.213275    1.254986          1          5


. ttest HEALTH, by(GROUP) unequal

Two-sample t test with unequal variances
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       0 |   90049    3.928206    .0033472    1.004448    3.921645    3.934766
       1 |    7774    3.213275    .0142337    1.254986    3.185373    3.241177
---------+--------------------------------------------------------------------
combined |   97823     3.87139      .00334    1.044642    3.864844    3.877937
---------+--------------------------------------------------------------------
    diff |            .7149307    .0146219                .6862683    .7435932
------------------------------------------------------------------------------
    diff = mean(0) - mean(1)                                      t =  48.8944
Ho: diff = 0                     Satterthwaite's degrees of freedom =  8654.22

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 1.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 0.0000

. ttest HEALTH, by(GROUP)

Two-sample t test with equal variances
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       0 |   90049    3.928206    .0033472    1.004448    3.921645    3.934766
       1 |    7774    3.213275    .0142337    1.254986    3.185373    3.241177
---------+--------------------------------------------------------------------
combined |   97823     3.87139      .00334    1.044642    3.864844    3.877937
---------+--------------------------------------------------------------------
    diff |            .7149307    .0121355                .6911453    .7387162
------------------------------------------------------------------------------
    diff = mean(0) - mean(1)                                      t =  58.9123
Ho: diff = 0                                     degrees of freedom =    97821

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 1.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 0.0000

.  summarize       HEALTH if GROUP==1

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      HEALTH |      7774    3.213275    1.254986          1          5

. scalar          N1=r(N)

. scalar          M1=r(mean)
```

```
. scalar          V1=r(Var)

. summarize       HEALTH if GROUP==0

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
      HEALTH |     90049    3.928206    1.004448          1          5

. scalar          N0=r(N)

. scalar          M0=r(mean)

. scalar          V0=r(Var)

. scalar          STNUM=M1-M0

. scalar          STDEN1=(1/N1+1/N0)^0.5

. scalar          STDEN2=(((N1-1)*V1+(N0-1)*V0)/(N1+N0-2))^0.5

. scalar          ST=STNUM/(STDEN1*STDEN2)

. quietly {
La statistique de test vaut -58.912323
```

* anova (ANOVA and treatment effects), Chi square test of independence

5.3 `oneway` (test d'égalité de deux moyennes ou plus)

En fait, à partir de trois moyennes (supposons que nous ayons trois groupes), on peut utiliser la commande `regress`. Nous reverrons cela dans le cours de méthodes statistiques d'évaluation de M2.

Dans l'exemple précédent, nous n'avons que deux groupes : soit la personne s'est rendue aux urgences, soit pas (rappelons qu'il s'agit de déclarations). On retrouve la statistique F du test de Wald en haut à droite de la régression. À vous de voir quelle approche vous souhaitez utiliser !

Nous verrons ensuite un deuxième exemple d'application du tests de Wald dans le cas où nous avons 3 groupes.

```
.  oneway HEALTH GROUP, tabulate

            |         Summary of HEALTH
      GROUP |        Mean   Std. Dev.       Freq.
------------+------------------------------------
          0 |   3.9282058    1.004448       90049
          1 |    3.213275   1.2549857        7774
------------+------------------------------------
      Total |   3.8713902   1.0446423       97823

                        Analysis of Variance
    Source              SS         df      MS            F     Prob > F
------------------------------------------------------------------------
Between groups      3657.71943      1   3657.71943   3470.66     0.0000
 Within groups       103093.24  97821   1.05389681
------------------------------------------------------------------------
    Total            106750.96  97822   1.09127762

Bartlett's test for equal variances:  chi2(1) = 804.8696  Prob>chi2 = 0.000

. regress HEALTH GROUP

      Source |       SS       df       MS              Number of obs =   97823
-------------+------------------------------           F(  1, 97821) = 3470.66
       Model |  3657.71943     1  3657.71943           Prob > F      =  0.0000
    Residual |   103093.24 97821  1.05389681           R-squared     =  0.0343
-------------+------------------------------           Adj R-squared =  0.0343
       Total |   106750.96 97822  1.09127762           Root MSE      =  1.0266

------------------------------------------------------------------------------
      HEALTH |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       GROUP |  -.7149307   .0121355   -58.91   0.000    -.7387162   -.6911453
       _cons |   3.928206   .0034211  1148.25   0.000     3.921501    3.934911
------------------------------------------------------------------------------


.  use "http://www.evens-salies.com/webstar.dta", clear

.  keep sesk cltypek

. drop if sesk==.
(5296 observations deleted)

. rename sesk FL


. rename cltypek GROUP
```

```
. replace FL=2-FL
(3250 real changes made)

. oneway FL GROUP, tabulate

  classroom |
       type |
  (1s2r3ra) |
         in | Summary of socio-economic status in
kindergarte |           kindergarten
          n |        Mean   Std. Dev.       Freq.
------------+------------------------------------
  small cla |   .47093023    .4992862        1892
  regular c |   .47736626   .49960168        2187
  regular + |   .50270027   .50010526        2222
------------+------------------------------------
      Total |   .48436756   .49979523        6301

                        Analysis of Variance
    Source              SS         df      MS            F     Prob > F
------------------------------------------------------------------------
Between groups      1.19561393      2   .597806965      2.39     0.0913
 Within groups      1572.51459   6298   .249684756
------------------------------------------------------------------------
    Total            1573.7102   6300   .249795271

Bartlett's test for equal variances:  chi2(2) =   0.0057  Prob>chi2 = 0.997

.  tabulate GROUP, generate(GROUP_)

      classroom type |
        (1s2r3ra) in |
        kindergarten |      Freq.     Percent        Cum.
---------------------+-----------------------------------
         small class |      1,892       30.03       30.03
       regular class |      2,187       34.71       64.74
regular + aide class |      2,222       35.26      100.00
---------------------+-----------------------------------
               Total |      6,301      100.00

. regress FL GROUP_*
note: GROUP_1 omitted because of collinearity

      Source |       SS       df       MS              Number of obs =    6301
-------------+------------------------------           F(  2,  6298) =    2.39
       Model |  1.19561393     2  .597806965           Prob > F      =  0.0913
    Residual |  1572.51459  6298  .249684756           R-squared     =  0.0008
-------------+------------------------------           Adj R-squared =  0.0004
       Total |   1573.7102  6300  .249795271           Root MSE      =  .49968

------------------------------------------------------------------------------
          FL |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     GROUP_1 |          0  (omitted)
     GROUP_2 |    .006436   .0156887     0.41   0.682    -.0243192    .0371913
     GROUP_3 |     .03177   .0156313     2.03   0.042     .0011273    .0624127
       _cons |   .4709302   .0114878    40.99   0.000     .4484103    .4934502
------------------------------------------------------------------------------


.  quietly log close
```