4. Méthodes d'estimation et tests

Nous nous sommes intéressés à l'estimation de la moyenne, ainsi que de l'erreur type, dont les formules sont exactes. Quand les fonctions des variables sont plus compliquées, que les lois suivies par les statistiques ne sont certainement pas normales, ou la supposition d'indépendance ne tient pas, on peut avoir recours au méthodes d'estimation précédentes, mais aussi à des méthodes classiques d'estimation avec Stata.

On place le problème d'estimation (**section 4.1**) dans le modèle d'échantillonnage. L'estimation peut porter sur le paramètre d'une loi. La méthode du maximum de vraissemblance (MV) est appropriée dans ce cas. Mais, on peut estimer un paramètre sans être obligé de supposer une loi, avec la méthode des moindres carrés par exemple. [1] Nous verrons des problèmes d'estimation ponctuelle et d'estimation par intervalle de confiance. Nous verrons ensuite comment obtenir quelques statistiques de test (**section 4.2**).

[1] L'approche bayésienne de l'estimation d'un paramètre s'enfonce un peu plus dans le cadre paramétrique, avec des hypothèses *a priori* supplémentaires. Par exemple, μ, le moment centré d'ordre un d'une variable d'espérance μ et de variance σ^2, est tel que $\mu \sim U[a, b]$. On peut facilement produire des valeurs pour ce type de variable avec Stata, en suivant la méthode Monte Carlo plutôt que de faire le calcul analytique.

4.1 Estimation

4.1.1 Maximum de vraisemblance, `ml`

La commande `ml` est puissante lorsqu'il s'agit d'estimer un paramètre difficilement calculable à la main. Un exemple emblématique est celui de la loi logistique. On considère le cas d'une variable Y dichotomique, qui suit non pas une loi de Bernouilli, mais logistique à valeur dans $\{0, 1\}$. Vous verrez le cadre multivarié (mdèles de plus de deux variables dans d'autres cours), en supposant que $P(Y = 1|X = x)$ est la fonction de répartition cumulée logistique :

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

$F(x, \beta)$ ne comporte qu'un paramètre, mais t́ant non-linéaire, la vraissemblance 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.

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

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

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

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

[Maximiser la vraissemblance]

Faisons un petit détour par la loi de Bernoulli. Pour cela, supposons que p_i ne dépende ni de i, ni de x ($x_i \equiv 1\ \forall i$ par exemple). Dans ce cas, on peut écrire $e^{\beta x}/(1 + e^{\beta x}) \equiv p = e^{\beta}/(1 + e^{\beta}) \equiv p$. 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 $l(p|\boldsymbol{y})$).

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

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

4.1.2 Moindres carrés, `regress`

La méthode des moindres carrés permet de trouver une estimation du moment centré d'ordre un d'une variable qui minimise sa variance. Par exemple,

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

[Minimiser la somme des carrés]

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

```
regress Y.
```

Dans le second cas, une régression sur les différentes variables,

```
regress Y X1 X2 ··· X3,
```

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 $\bar{Y}$.

```
.  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 rééchantillonnage, 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. J pense par exemple à la méthode des moments généralisés, la commande `gmm`. Je vous conseille de revoir cette méthode dans le cas univarié, le cas "simple" par opposition à "généralisée" qui est le cadre dans lequel cette commande a été créée).

4.2 Test

4.2.1 Test du ratio de vraissemblance lrtest

[Exposer la théorie]

```
.  cls

. clear all

. set obs 1000
obs was 0, now 1000

. set seed 3102019

.  generate Y=rnormal()

.  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
------------------------------------------------------------------------------

. 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
```

4.2.1 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 Neslon (2004, p. 95), qui s'avère très utile lorsque l'on veut s'informer vite et bien sur des méthodes statistiques.

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 de 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

. 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

.  quietly log close
```