Introduction au logiciel Stata

Mars 2015

Table des matières

Cours 8. Éléments du langage et statistiques descriptives

Sommaire

Dans ce premier chapitre, on s'intéressera principalement au mode de représentation des données sous Stata et à leur manipulation, en opérant sur des sous-ensembles de variables ou en ne sélectionnant que certaines observations. Les principales commandes permettant de résumer de manière quantitative ou graphique la distribution d'une variable numérique ou catégorielle sont discutées dans le cadre des données sur les poids à la naissance de Hosmer & Lemeshow.

8.1 Représentation des données sous Stata

Les données sous Stata sont généralement de type nombre ou chaîne de caractères. Le plus simple est souvent de travailler avec des nombres, et dans le cas des variables qualitatives d'associer à leurs modalités des étiquettes.

8.1.1 Le langage Stata

Il existe des commandes qui permettent de générer facilement des séries de nombres tirés au hasard. Par exemple, la série d'instructions suivantes stocke dans une variable appelée `x` 10 observations tirées d'une loi normale de moyenne 12 et d'écart-type 2.

```
. set obs 10
. generate x = rnormal(12, 2)
. format x %6.3f
. summarize x, format
```

```
obs was 0, now 10

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
           x |        10      11.112       2.246      7.956     14.224
```

On remarquera plusieurs caractéristiques remarquables du langage : il est nécessaire d'indiquer à Stata quelle est la taille de l'échantillon sur lequel on travaille. On verra dans les paragraphes suivants comment cette donnée peut être obtenue lors d'une saisie manuelle ou de l'importation d'un fichier de données externes. La commande `generate` sert à associer à une variable, ici `x`, une séquence de valeurs numériques (assimilée ici à nos 10 observations) fournies par la commande `rnormal()`. Cette dernière dispose d'options permettant de spécifier les paramètres de la loi (moyenne et écart-type, dans l'ordre). La commande `format x %6.3f` permet de limiter l'affichage à 3 décimales : il s'agit d'une propriété de représentation des valeurs de `x` directement associée à la variable que la commande `summarize` peut exploiter.

Les données individuelles peuvent être examinées à l'aide de la commande `list`. Par exemple, la commande `list x` affichera l'ensemble des valeurs de `x`. Comme il n'y a qu'une seule variable présente dans l'espace

de travail de Stata, il est d'ailleurs équivalent de taper `list` tout court. On peut utiliser l'option `in` pour restreindre l'affichage des valeurs de `x` à la 5e observation ou aux 5 premières observations. Dans ce dernier cas, on indique les rangs des observations sous la forme 1re valeur/dernière valeur : l'expression `1/5` désigne donc les observations no 1 à 5.

```
. list x in 5

     +-------+
     |     x |
     |-------|
  5. | 7.956 |
     +-------+

. list x in 1/5

     +--------+
     |      x |
     |--------|
  1. | 11.118 |
  2. | 13.889 |
  3. | 14.224 |
  4. |  8.726 |
  5. |  7.956 |
     +--------+
```

8.1.2 Créer et manipuler des variables

Dans le cas des petits jeux de données, il est possible de saisir soi-même les observations, bien que la plupart du temps on préférera travailler à partir de fichier externe. Pour cela, on dispose de la commande `input` qui s'utilise de la manière suivante : on indique après la commande le nom de la ou des variables, séparées par un espace, puis on tape sur la touche Entrée avant de saisir les données, toujours séparées par des espaces. Pour indiquer à Stata que la saisie est terminée, on tape le mot `end`. Cette saisie manuelle peut également être réalisée à partir de l'éditeur de données (`Data > Data Editor > Data Editor (Edit)`). Voici un exemple d'utilisation avec une série de 10 mesures de poids recueillies chez des nouveaux-nés (`x`, en grammes) et leur mère (`y`, en kilogrammes).

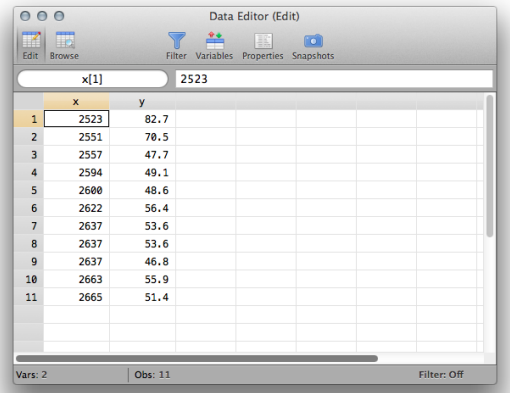
x	2523	2551	2557	2594	2600	2622	2637	2637	2663	2665
y	82.7	70.5	47.7	49.1	48.6	56.4	53.6	46.8	55.9	51.4

La saisie sous Stata se ferait ainsi : saisir `input x y` dans la console Stata, puis pour la ligne numérotée `1.` indiquer `2523 82.7` et taper sur Entrée, pour la ligne numérotée `2.` indiquer `2551 82.7` et taper sur Entrée et ainsi de suite jusqu'à la 10e ligne. Pour la 11e ligne, on écrira simplement `end` et on tapera sur Entrée. Le résultat devrait ressembler à ceci :

```
. input x y

            x          y
  1. 2523 82.7
  2. 2551 70.5
  3. 2557 47.7
  4. 2594 49.1
  5. 2600 48.6
  6. 2622 56.4
  7. 2637 53.6
  8. 2637 53.6
  9. 2637 46.8
 10. 2663 55.9
 11. 2665 51.4
 12. end
```

Sous l'éditeur de données, cela donnerait :



Il est possible de transformer les valeurs prises par une variable ou de créer de nouvelles variables à partir des valeurs prises par une variable à l'aide des commandes `generate` et `replace`. Cette dernière commande travaille exclusivement sur une variable existante. Voici un exemple d'utilisation de `generate` (comamnde que l'on peut abbréger `gen`) où l'on convertit les poids des bébés (`x`) en kilogrammes.

```
. generate x2 = x / 1000
. list x x2 in 1/3
```

```
     +--------------+
     |    x      x2 |
     |--------------|
  1. | 2523   2.523 |
  2. | 2551   2.551 |
  3. | 2557   2.557 |
     +--------------+
```

On peut remplacer l'ensemble des valeurs à partir d'une transformation quelconque, par exemple le logarithme

```
. replace x2 = log(x2)

(10 real changes made)
```

ou bien remplacer spécifiquement certaines valeurs en indiquant un n° d'observation, comme illustré ci-après.

```
. replace x2 = 2600 in 3
. list x x2 in 1/3

(1 real change made)

     +-----------------+
     |    x         x2 |
     |-----------------|
  1. | 2523   .9254487 |
  2. | 2551   .9364855 |
  3. | 2557       2600 |
     +-----------------+
```

La commande `drop` peut être utilisée pour supprimer n'importe quel variable de l'espace de travail.

```
. drop x2
```

8.1.3 Sélection indexée ou critériée d'observations

On a déjà présenté l'option de sélection des observations par indices (ou rangs) avec l'option `in`.

```
. list x in 1/3
```

```
     +------+
     |    x |
     |------|
  1. | 2523 |
  2. | 2551 |
  3. | 2557 |
     +------+
```

Il est également possible de sélectionner des observations sur la base d'un critère externe, par exemple les valeurs prises par une deuxième variable numérique. Dans l'exemple suivant, on ne retient que les poids des bébés pour lesquels le poids des mères est inférieur ou égal à 50.

```
. list x if y <= 50
```

```
     +------+
     |    x |
     |------|
  3. | 2557 |
  4. | 2594 |
  5. | 2600 |
  8. | 2637 |
     +------+
```

Supposons que l'on dispose également d'une information concernant le fait que la mère fumait pendant le premier trimestre de grossesse et appelons cette variable `z`. Lorsque la mère ne fumait pas pendant cette période, la variable vaut 1 ; lorsque la mère fumait, la variable vaut 2. Voici ci-dessous le tableau de données précédent modifié pour prendre en compte cette information.

x	2523	2551	2557	2594	2600	2622	2637	2637	2663	2665
y	82.7	70.5	47.7	49.1	48.6	56.4	53.6	46.8	55.9	51.4
z	1	1	2	2	2	1	1	1	2	2

La saisie des nouvelles données ne pose aucune difficulté, et on utilisera encore une fois `input` en indiquant `z` comme nouvelle variable. On n'oubliera de signaler la fin de la saisie à l'aide de l'instruction `end` (suivi de Entrée). Voici un aperçu des 5 premières observations pour les 3 variables :

```
. list in 1/5
```

```
     +-----------------+
     |    x      y   z |
     |-----------------|
  1. | 2523   82.7   1 |
  2. | 2551   70.5   1 |
  3. | 2557   47.7   2 |
  4. | 2594   49.1   2 |
  5. | 2600   48.6   2 |
     +-----------------+
```

À partir de là, il est possible de raffiner nos critères de recherche en restreignant la sélection des observations `x` selon les valeurs prises par `y` et `z`. L'instruction suivante affiche le poids des bébés dont la mère pèse moins (strictement) de 55 kilogrammes et qui ne fumait pas durant sa grossesse.

```
. list x if y < 55 & z == 1
```

```
     +------+
     |    x |
     |------|
```

```
  7. | 2637 |
  8. | 2637 |
     +------+
```

On voit qu'il y a deux unités statistiques qui vérifient les conditions précédentes ($y < 55$ et $z = 1$). On peut d'ailleurs les dénombrer avec la commande `count` :

```
. count if y < 55 & z == 1

  2
```

8.1.4 Traitement des valeurs manquantes

Les valeurs manquantes sont représentées par un point sous Stata. On peut par exemple remplacer la 3e observation de `x` par une valeur manquante, en utilisant la commande `replace` présentée plus haut.

```
. replace x = . in 3
. summarize x

(1 real change made, 1 to missing)

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
           x |         9    2610.222    48.53035       2523       2665
```

On vérifiera que le nombre d'observations reporté par `summarize` tient bien compte de la donnée manquante ($n = 9$ au lieu de 10). On peut d'ailleurs vérifier le nombre de valeurs manquantes identifiées pour une variable à l'aide de la commande `misstable` :

```
. misstable summarize x

                                                              Obs<.
                                                +------------------------------
             |                                  | Unique
    Variable |     Obs=.     Obs>.     Obs<.   | values        Min         Max
-------------+----------------------------------+------------------------------
           x |         1                    9   |      8       2523        2665
-------------------------------------------------------------------------------
```

En pratique, les données manquantes sont représentées sous la forme de très grand nombre donc il faut faire attention dans les tests de comparaisons numériques, et dans le doute toujours utiliser un test du type `if !missing(x)` (plus élégant que `if x < .`) pour être sûr de travailler sur les données observées.

```
. summarize y if !missing(x)

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
           y |         9    57.22222    11.85219       46.8       82.7
```

8.2 Gestion de données

8.2.1 Importation de données externes

Il existe plusieurs commandes permettant d'importer des données contenues dans un fichier texte. Pour les fichiers dans lesquels les champs sont séparés par un ou plusieurs espaces, on utilisera la commande `infile`. Dans le cas où il existe un séparateur de champ de type virgule (typique des fichiers CSV exportés depuis un tableau comme Excel) ou tabulation, la commande `insheet` sera utilisée, éventuellement en précisant le type de délimiteur de champ avec l'option `delimiter()`. La commande `insheet` fonctionne bien dans le cas où la première ligne du fichier contient le nom des variables. Mais dans les deux cas, il est possible de fournir une liste pour les noms des variables, et on indiquera toujours le nom du fichier après l'instruction `using`. Le

format de représentation des données lues peut également être adapté en spécifiant avant chaque variable son type : par exemple, la commande

```
. infile str5 nom age byte rep using "fichier.txt", clear
```

indique à Stata de construire à partir du fichier appelé `fichier.txt` un tableau contenant trois variables : `nom`, `age` et `rep`. La variable `nom` doit explicitement être traitée comme une chaîne de caractères (5 maximum) et le format de stockage de la variable `rep` doit être limité au minimum (1 byte = valeurs variant de -127 à 100), par exemple pour ne pas occuper inutilement la mémoire. Enfin, dans certains cas, on peut s'affranchir des options en ligne de commande (nom des variables et format de stockage) et stocker toutes ces informations dans ce que l'on appelle un dictionnaire, voir `help infile2`.

Considérons les données sur les poids à la naissance, disponibles dans un fichier appelé `birthwt.dat` dans lequel les champs sont séparés par un espace comme dans l'aperçu ci-dessous :

```
0 19 182 2 0 0 0 1 0 2523
0 33 155 3 0 0 0 0 3 2551
0 20 105 1 1 0 0 0 1 2557
0 21 108 1 1 0 0 1 2 2594
0 18 107 1 1 0 0 1 0 2600
```

On remarquera que le nom des variables n'apparaît pas sur la 1re ligne du fichier. Chaque colonne correspond, dans l'ordre, aux variables suivantes : statut pondéral du bébé à la naissance `low` (= 1 si poids < 2.5 kg, 0 sinon), `age` de la mère (années), `lwt` poids de la mère (en livres), `race` ethnicité de la mère (codée en trois classes, 1 = white, 2 = black, 3 = other), `smoke` (= 1 si consommation de tabac durant la grossesse, 0 sinon), `ptl` (nombre d'accouchements pré-terme antérieurs), `ht` (= 1 si antécédent d'hypertension, 0 sinon), `ui` (= 1 si manifestation d'irritabilité utérine, 0 sinon), `ftv` (nombre de consultations chez le gynécologue durant le premier trimestre de grossesse), `bwt` pour le poids des bébés à la naissance

On peut utiliser la commande `infile` pour importer ces données, en indiquant la liste des variables que Stata associera à chaque colonne du fichier de données. L'option `clear` indique à Stata de supprimer les données existantes dans l'espace de travail avant de réaliser l'importation.

```
. infile low age lwt race smoke ptl ht ui ftv bwt using "birthwt.dat", clear
. list in 1/5

(189 observations read)

     +------------------------------------------------------------------+
     | low   age   lwt   race   smoke   ptl   ht   ui   ftv    bwt |
     |------------------------------------------------------------------|
  1. |   0    19   182      2       0     0    0    1     0   2523 |
  2. |   0    33   155      3       0     0    0    0     3   2551 |
  3. |   0    20   105      1       1     0    0    0     1   2557 |
  4. |   0    21   108      1       1     0    0    1     2   2594 |
  5. |   0    18   107      1       1     0    0    1     0   2600 |
     +------------------------------------------------------------------+
```

Pour afficher les observations lues, à présent contenues dans l'espace de travail, on utilisera la commande `list`. Comme on l'a vu, il est possible de limiter le nombre d'observations (« lignes ») affichées en utilisant un filtre sur les numéros d'observations : l'option `in 1/5` indique à Stata de ne sélectionner que les observations allant de 1 à 5.

8.2.2 Gestion de variables

On peut également afficher les données importées dans l'espace de travail à l'aide de la commande `describe`. Avec l'option `simple`, Stata renvoie uniquement le nom des variables, alors qu'avec l'option `short` on a un résumé indiquant le nombre d'observations et de variables.

```
. describe, short

Contains data
  obs:           189
```

```
 vars:            10
 size:         7,560
Sorted by:
     Note:  dataset has changed since last saved
```

Il est également possible de fournir une liste de variables, par exemple les variables `low`, `age` et `lwt`. Il s'agit des trois premières variables, et l'expression `describe low age lwt` peut se simplifier en `describe low-lwt`.

```
. describe low-lwt

              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
low             float  %9.0g
age             float  %9.0g
lwt             float  %9.0g
```

On constate que ces trois variables (indicateur de poids $<$ 2500 g, âge de la mère et poids de la mère, en livres) sont traitées comme des nombres. Par souci d'économie d'espace mémoire, on pourrait préférer la commande

```
. infile byte low age lwt race smoke ptl ht ui ftv bwt using "birthwt.dat", clear

(189 observations read)
```

pour indiquer à Stata de réserver moins d'espace mémoire pour la variable `low` qui est une variable binaire (le nom de la variabe a été préfixé par l'instruction `byte`).

Pour associer des étiquettes aux variables, on utilisera la commande `label variable`.

```
. label variable low "Poids inférieur à 2,5 kg"
. describe low-lwt race

              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
low             byte   %8.0g                  Poids inférieur à 2,5 kg
age             float  %9.0g
lwt             float  %9.0g
race            float  %9.0g
```

La variable `race`, bien que qualitative, est représentée comme un nombre par Stata (`float`), et l'on peut vérifier ses modalités à l'aide de la commande `tabulate` qui fournit un tableau d'effectif :

```
. tabulate race

       race |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |         96       50.79       50.79
          2 |         26       13.76       64.55
          3 |         67       35.45      100.00
------------+-----------------------------------
      Total |        189      100.00
```

Les modalités ou niveaux des variables qualitatives traités comme des nombres peuvent se voir associer des étiquettes, ce qui facilite la lecture des tableaux de résumé ou graphiques descriptifs. Pour cela, il est nécessaire de définir, dans un premier temps, la correspondance entre les valeurs prises par la variable et les étiquettes (`label define`), puis, dans un second temps, d'associer ces étiquettes à la variables (`label values`). Voici comment procéder pour les variables `race`, `ht` et `ui` :

```
. label define yesno 0 "No" 1 "Yes"
. label define ethn 1 "White" 2 "Black" 3 "Other"
. label values ht ui yesno
. label values race ethn
```

L'usage de `label define` est assez simple : on donne le nom de l'étiquette qui servira de référence et on associe à chaque valeur une description sous forme de caractères (0 est associé à `"No"`). De même, pour `label values`, on indique la ou les variables suivies de la référence créée avec `label define`.

```
. tabulate race
```

```
       race |      Freq.     Percent        Cum.
------------+-----------------------------------
      White |         96       50.79       50.79
      Black |         26       13.76       64.55
      Other |         67       35.45      100.00
------------+-----------------------------------
      Total |        189      100.00
```

8.2.3 Convertir une variable numérique en variable catégorielle

Lorsque l'on connaît les bornes des intervalles de classe que l'on souhaite considérer, on peut utiliser la commande `egen cut`, en indiquant les bornes inférieures des intervalles. Voici un exemple d'utilisation.

```
. egen lwt3 = cut(lwt), at(70,120,170,220,270)
. tabulate lwt3
```

```
       lwt3 |      Freq.     Percent        Cum.
------------+-----------------------------------
         70 |         75       39.68       39.68
        120 |         93       49.21       88.89
        170 |         17        8.99       97.88
        220 |          4        2.12      100.00
------------+-----------------------------------
      Total |        189      100.00
```

Ici, on utilise un ensemble d'extensions de commandes appelé `egen more` : il s'agit basiquement d'un équivalent de la commande `generate` permettant de créer de nouvelles variables, mais acceptant un certain nombre d'options (agissant la plupart du temps comme des fonctions qui permettent d'effectuer des calculs sur une variable donnée). Voir l'aide en ligne, `help egen`, pour plus d'informations et en particulier la liste des fonctions disponibles (`count`, `iqr`, `max`, etc.).

Bien que l'on aît explicitement spécifié l'ensemble des bornes des intervalles, il serait également possible d'écrire `at(70(50)270)` pour indiquer à Stata de construire une séquence de valeurs variant de 70 à 270 par pas de 50. Stata peut également construire automatiquement des groupes plus ou moins équilibrés avec l'option `group(4)`.

Pour construire des classes basées sur les quartiles ou les déciles, on utilisera plutôt la commande `xtile` en précisant dans l'option `nq()` le nombre de groupes désiré.

```
. drop lwt3
. xtile lwt3 = lwt, nq(4)
. tabulate lwt3
```

```
4 quantiles |
     of lwt |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |         53       28.04       28.04
          2 |         43       22.75       50.79
          3 |         46       24.34       75.13
          4 |         47       24.87      100.00
------------+-----------------------------------
      Total |        189      100.00
```

8.3 Statistiques descriptives univariée et estimation

8.3.1 Résumer une variable numérique

La commande `summarize` fournit un résumé numérique en 4 points (moyenne, écart-type, minimum et maximum) pour une ou plusieurs variables numériques. L'option `detail` fournit un résumé plus exhaustif, incluant notamment les 5 valeurs les plus extrêmes, différents quantiles et les indicateurs de symétrie et d'aplatissement de la distribution de la variable.

```
. summarize bwt
```

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         bwt |       189    2944.587    729.2143        709       4990
```

Pour obtenir des intervalles de confiance à 95 % reposant sur l'approximation par la loi normale, on utilisera la commande `ci` suivie du nom de la variable d'intérêt.

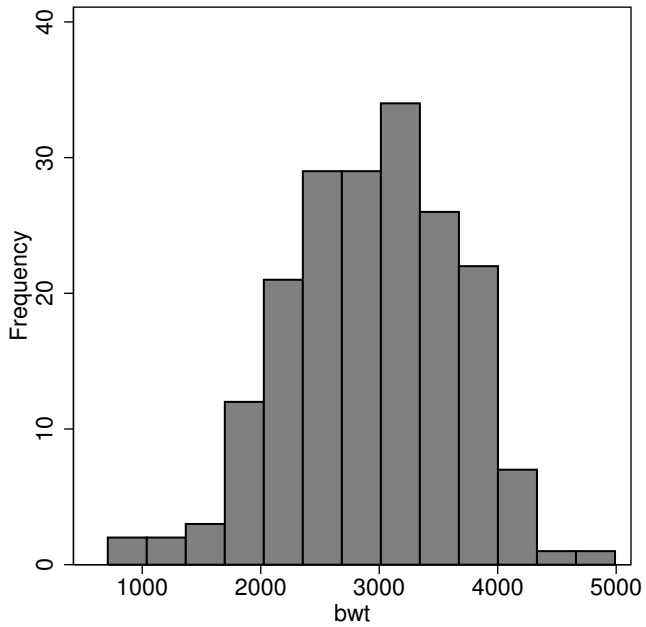
```
. ci bwt
```

```
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
         bwt |        189    2944.587    53.04254        2839.952    3049.222
```

Pour construire un histogramme de fréquences ou d'effectifs, la commande à utiliser est `histogram`. On précisera l'option `frequency` si l'on souhaite travailler avec les effectifs, ou `percent` pour les proportions.

```
. histogram bwt, frequency
```

```
(bin=13, start=709, width=329.30769)
```



L'option `bin()` peremt de modifier le nombre d'intervalles de classe utilisé par Stata. Si l'on souhaite afficher une courbe de densité non-paramétrique, on ajoutera l'option `kdensity`. Il existe également une option `discrete`, qui assimile l'histogramme à une représentation sous forme de diagramme en barres dans laquelle on ne représente sur l'axe des abscisses que les valeurs uniques de la variable numérique, sans considération d'intervalles de classe.

8.3.2 Résumer une variable catégorielle

La commande `tabulate` (que l'on peut abbréger en `tab`) fournit un tableau d'effectifs pour une variable.

```
. tabulate race, plot
```

```
       race |      Freq.
------------+------------+-----------------------------------------------------
      White |         96 |**************************************************
      Black |         26 |**************
      Other |         67 |***********************************
------------+------------+-----------------------------------------------------
      Total |        189
```

On lui préférera `tab1` si l'on souhaite construire des tableaux d'effectifs (univariés) pour plusieurs variables, avec la syntaxe suivante :

```
. tab1 race ht ui
```

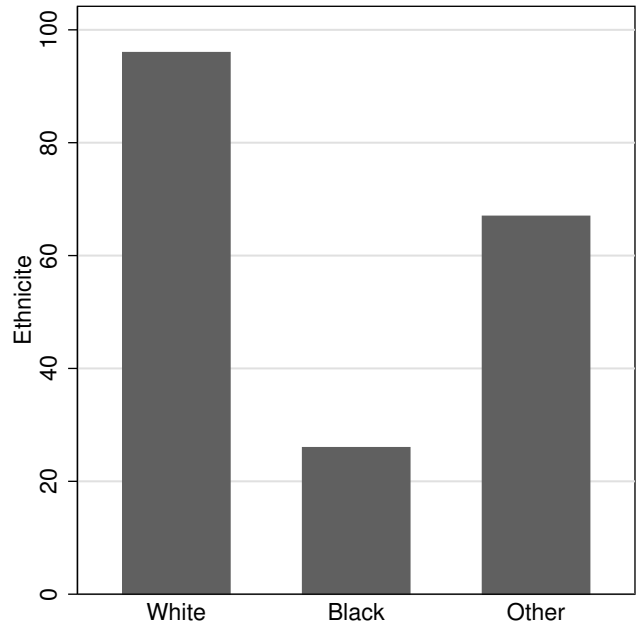
Comme dans le cas des variables numériques, la commande `ci` fournit des intervalles de confiance pour une proportion. On ajoutera l'option `binomial` si l'on souhaite utiliser la distribution binomiale pour construire ces intervalles de confiance.

```
. ci low, binomial
```

```
                                                    -- Binomial Exact --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
         low |        189    .3121693    .0337058        .2468886    .3834546
```

L'option `level()` permet de changer la taille des intervalles de confiance. Par exemple, `level(90)` indique à Stata de renvoyer des intervalles de confiance à 90 %, au lieu de 95 % qui est la valeur par défaut.

```
. gen freq = 1
. graph bar (sum) freq, over(race) ytitle("Ethnicite")
```



On remplacera l'instruction `bar` par `hbar` pour représenter les barres horizontalement plutôt que verticalement.

8.4 Statistiques descriptives bivariées

8.4.1 Décrire une variable numérique par d'une variable qualitative

La commande `summarize` n'opère que de manière univariée, c'est-à-dire pour chaque variable listée. Si l'on souhaite résumer une variable numérique pour chaque niveau d'une variable catégorielle, on peut utiliser une option de sélection `by`, à placer en début de commande.

```
. by low, sort: summarize lwt
```

```
-------------------------------------------------------------------------------
-> low = 0

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         lwt |       130       133.3    31.72402         85        250

-------------------------------------------------------------------------------
-> low = 1

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         lwt |        59    122.1356    26.55928         80        200
```

On notera que les données doivent être triées au préalable, d'où l'ajout de l'option `sort`. Une alternative consiste à utiliser `bysort` directement.

```
. bysort low: summarize
```

Attention à ne pas confondre : avec , lorsque `by` est spécifié avant une commande.

Il peut arriver que l'on ne souhaite calculer que certaines statistiques, par exemple la moyenne et l'écart-type. Dans ce cas, la commande `tabstat` est plus simple à utiliser. Voici un exemple de son utilisation :

```
. tabstat lwt, by(low) stats(mean sd) format(%6.2f)
```

```
Summary for variables: lwt
     by categories of: low (Poids inférieur à 2,5 kg)

     low |      mean        sd
---------+--------------------
       0 |    133.30     31.72
       1 |    122.14     26.56
---------+--------------------
   Total |    129.81     30.58
------------------------------
```

L'option `format(%6.2f)` permet de limiter l'affichage à 2 décimales.

Une formulation alternative, et qui se généralise à plusieurs facteurs de classification, consiste à utiliser la commande `table`. L'équivalent de l'option `stats()` de `tabstat` est ici `contents()` et l'on y spécifie ce que l'on souhaite calculer : `freq` correspond à l'effectif par modalité, `mean lwt` correspond à la moyenne de la variable `lwt` pour chaque niveau du facteur de classification, etc.

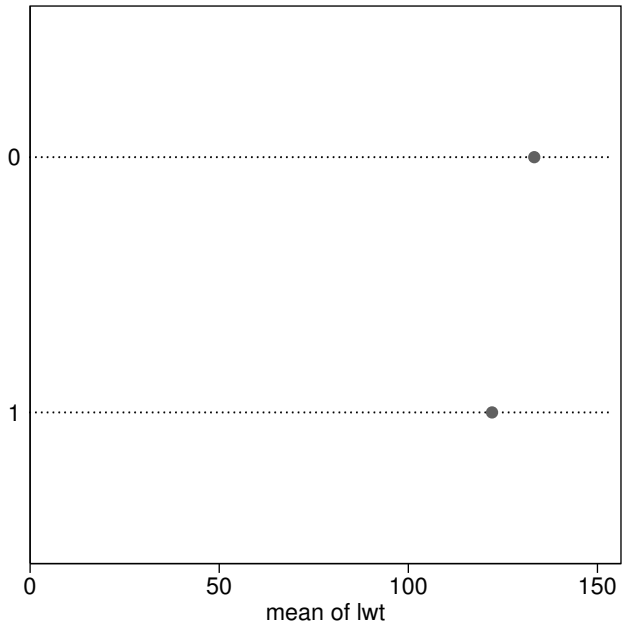
```
. table low, contents(freq mean lwt sd lwt) format(%6.2f)
```

```
----------------------------------------------
Poids     |
inférieu |
r à 2,5  |
kg        |      Freq.   mean(lwt)     sd(lwt)
----------+-----------------------------------
        0 |     130.00      133.30       31.72
```

```
       1 |      59.00      122.14       26.56
-------------------------------------------
```

Pour représenter le poids moyen des mères pour les enfants en sous-poids ou dans les normes, un diagramme en points (voire en barres) est approprié : le facteur de classification est indiqué dans une option `over()`, qui permet de superposer des éléments graphiques dans la même figure.

```
. graph dot lwt, over(low)
```



Par défaut, c'est la moyenne de la variable numérique (`lwt`) qui est considére, mais on peut utiliser une autre statistique en utilisant l'options `(stats)`. Par exemple, si l'on souhaite afficher le poids médian, on remplacera la commande précédente par

```
. graph dot (median) lwt, over(low)
```

Dans certains cas, il peut arriver que l'on ne s'intéresse qu'à une statistique de groupe particulière. Par exemple, supposons que l'on souhaite calculer le poids maximal des bébés (en grammes) dans les deux groupes d'individus définis par la variable `ui` (0, pas de douleur intra-utérine durant la grossesse ; 1, douleur intra-utérine). Il est possible de reposer sur la commande `egen` (et c'est ce que l'on fera la plupart du temps même si ce n'est vraiment pas toujours élégant). Voici comment on procéderait dans ce cas :

```
. bysort ui: egen maxbwt = max(bwt)
. tabulate(maxbwt)
```

```
     maxbwt |      Freq.     Percent        Cum.
------------+-----------------------------------
       3912 |         28       14.81       14.81
       4990 |        161       85.19      100.00
------------+-----------------------------------
      Total |        189      100.00
```

Cependant, on constate que Stata a créé une nouvelle variable, `maxbwt`, qui contient essentiellement deux valeurs uniques. Dans le cas de gros tableaux de données, ce n'est pas nécessairement le plus judicieux, et l'on peut reposer sur une approche différente : stocker les deux valeurs constantes dans des variables dites locales, en utilisant l'instruction `scalar` car il s'agit dans ce cas de valeurs numériques. Pour calculer et afficher le poids maximum observé chez les bébés dans le groupe des mères n'ayant présenté aucune douleur intra-utérine, on utiliserait les commandes suivantes :

```
. drop maxbwt
. quietly: summarize bwt if ui == 0
```

```
. scalar maxbwt0 = r(max)
. display maxbwt0
```

```
4990
```

On utilise en fait les résultats générés par la commande `summarize` mais stockés de manière invisible (on peut y accéder en tapant `return list`, juste après avoir taper `summarize bwt ...`), que l'on peut stocker à l'aide de la commande `r()` et afficher ensuite à l'aide de `display`. Le fait de préfixer la commande `summarize` par `quietly:` permet de supprimer l'affichage des résultats (mais ceux-ci sont toujours accessibles).

8.4.2 Décrire deux variables qualitatives

La commande `tabulate` permet de construire des tableaux de contingence lorsqu'on l'utilise avec une liste de deux variables. Par exemple, la commande suivante permet de croiser les modalités des variables `low` (en lignes) et `smoke` (en colonnes) et reporte les fréquences relatives calculées par lignes.

```
. tabulate low smoke, row
```

```
+----------------+
| Key            |
|----------------|
|   frequency    |
| row percentage |
+----------------+

     Poids |
inférieur |         smoke
 à 2,5 kg |         0          1 |     Total
----------+----------------------+----------
         0 |        86         44 |       130
           |     66.15      33.85 |    100.00
----------+----------------------+----------
         1 |        29         30 |        59
           |     49.15      50.85 |    100.00
----------+----------------------+----------
     Total |       115         74 |       189
           |     60.85      39.15 |    100.00
```

Les autres options, `col` et `cell`, permettent de calculer les fréquences relatives par colonnes ou par rapport à l'ensemble des observations (fréquences conditionnelles).

Ce qu'il faut retenir

- Stata représente une liste de variables, assimilables à un tableau de données, ayant toutes le même nombre d'observations et les valeurs des variables sont généralement des nombres auxquels on peut associer des étiquettes lorsqu'il désignent les modalités d'une variable qualitative.
- Les commandes `summarize` et `tabulate` fournissent un résumé descriptif univarié dans le cas des variables numériques et catégorielles, alors que `tabstat` et `tabulate` permettent de travailler de manière bivariée.
- Les principales commandes graphiques pour représenter la distribution d'une variable numérique ou catégorielle sont `histogram` et `graph bar` (diagramme en barres) ou `graph dot` (diagramme en points).

Cours 9. Mesures d'association, comparaisons de moyennes ou de proportions pour deux échantillons ou plus

Sommaire

Ce chapitre porte sur les mesures d'association entre deux variables catégorielles (test du χ^2 ou de Fisher pour l'analyse d'un tableau de contingence, et calcul de l'odds-ratio) ou entre une variable numérique et un facteur de classification. Dans ce dernier cas, on considèrera le cas de deux échantillons indépendants (test de Student) ou non (test de Wilcoxon), et les modèles paramétriques et non-paramétriques pour les situations à plus de deux échantillons (ANOVA, ANOVA de Kruskal-Wallis). La méthode de correction de Bonferroni pour les comparaisons multiples de traitement et le test de tendance linéaire pour l'ANOVA seront également discutés. Le cas de l'ANOVA à deux facteurs est présenté de manière succincte, en se limitant aux principales commandes permettant de construire le tableau d'ANOVA et tracer un graphique d'interaction.

9.1 Comparaisons de deux moyennes de groupe

9.1.1 Échantillons indépendants

La commande `ttest` permet de réaliser un test de Student pour la comparaison de deux moyennes de groupe, en considérant les variances égales ou non dans la population (option `unequal`). Si l'on considère le poids des mères en fonction de l'indicateur de sous-poids à la naissance pour les bébés, un bref résumé descriptif (moyenne et écart-type) peut être obtenu à l'aide de la commande `tabstat`, ou de la manière suivante :

```
. format lwt %4.1f
. tabulate low, summarize(lwt)
```

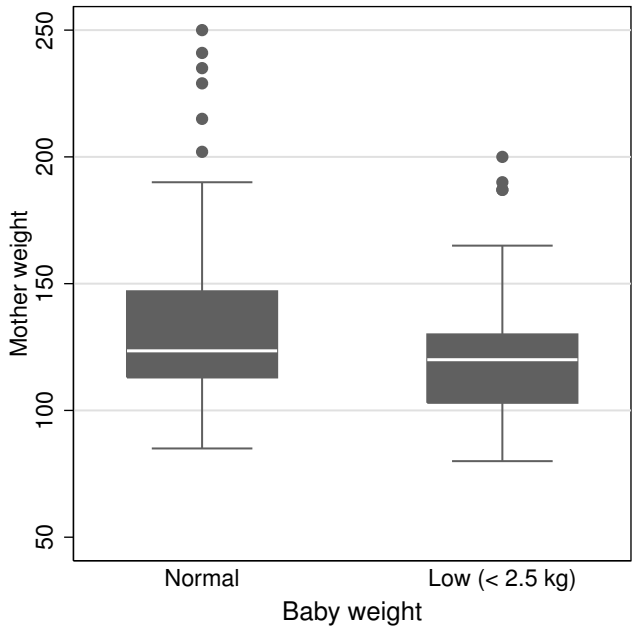
```
      Poids |
 inférieur |           Summary of lwt
  à 2,5 kg |        Mean   Std. Dev.       Freq.
-----------+------------------------------------
         0 |       133.3        31.7         130
         1 |       122.1        26.6          59
-----------+------------------------------------
     Total |       129.8        30.6         189
```

En termes de représentation graphique, les distributions de ces deux séries de mesure peuvent être visualisées à l'aide d'histogrammes,

```
. histogram lwt, by(low)
```

ou de boîtes à moustaches :

```
. graph box lwt, over(low, relabel(1 "Normal" 2 "Low (< 2.5 kg)")) ///
.   b1title("Baby weight") ytitle("Mother weight")
```



On notera que dans le cas des boîtes à moustaches (commande `graph box`), on dispose à la fois d'une option `by()` (affichage des distributions conditionnelles dans des graphiques juxtaposés) et d'une option `over()` (affichage des distributions conditionnelles dans le même graphique). Cette dernière n'est pas disponible pour `histogram`.

Le test de Student « classique » est réalisé en indiquant la variable réponse et le facteur de classification dans une option `by()`.

```
. ttest lwt, by(low)

Two-sample t test with equal variances
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       0 |     130       133.3     2.78238    31.72402     127.795     138.805
       1 |      59    122.1356    3.457723    26.55928    115.2142     129.057
---------+--------------------------------------------------------------------
combined |     189    129.8148    2.224323    30.57938     125.427    134.2027
---------+--------------------------------------------------------------------
    diff |            11.16441    4.743297                1.807157    20.52166
------------------------------------------------------------------------------
    diff = mean(0) - mean(1)                                      t =   2.3537
Ho: diff = 0                                     degrees of freedom =      187

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.9902         Pr(|T| > |t|) = 0.0196          Pr(T > t) = 0.0098
```

Dans l'approche ci-dessus, on dispose de deux variables bien identifiées, l'une servant de variable réponse et l'autre de facteur de classification. Il est également possible de travailler avec deux séries de mesures (non nécessairement de la même taille). Voici une approche possible, qui sert essentiellement à démontrer comment l'on peut gérer un second tableau de données sans effacer les données présentes dans l'espace de travail en utilisant les commande `preserve` et `restore`. Dans un premier temps, on créera deux nouvelles variables, `lwt1` et `lwt2`, dans lesquelles on stocke les poids des deux groupes de mères.

```
. preserve
. gen lwt1 = lwt if low == 0
. gen lwt2 = lwt if low == 1

(59 missing values generated)
```

```
(130 missing values generated)
```

On peut vérifier les caractéristiques de ces deux variables, voire les comparer avec l'ensemble des données de l'échantillon (`lwt`), à l'aide de `summarize` : la notation `lwt*` indique à Stata de considérer toutes les variables dont le nom commence par `lwt` (donc, `lwt`, `lwt1` et `lwt2` dans le cas présent).

```
. summarize lwt*
```

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         lwt |       189    129.8148    30.57938         80        250
        lwt1 |       130       133.3    31.72402         85        250
        lwt2 |        59    122.1356    26.55928         80        200
```

La formulation alternative pour le test t est alors :

```
. ttest lwt1 == lwt2, unpaired
```

Il est important dans ce cas de préciser l'option `unpaired`. On n'oubliera pas de taper la commande `restore` pour revenir au tableau de données initial (et supprimer les variables générées entre temps).

L'option `welch` fournit un test t utilisant l'approche de Welch (alors que `unequal` repose sur l'approximation de Satterthwaite).

Si l'on souhaite tester formellement l'hypothèse d'égalité des variances à l'aide d'un test F, on peut utiliser la commande `sdtest`.

```
. sdtest lwt, by(low)
```

```
Variance ratio test
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       0 |     130       133.3     2.78238    31.72402     127.795     138.805
       1 |      59    122.1356    3.457723    26.55928    115.2142     129.057
---------+--------------------------------------------------------------------
combined |     189    129.8148    2.224323    30.57938     125.427    134.2027
------------------------------------------------------------------------------
    ratio = sd(0) / sd(1)                                         f =   1.4267
Ho: ratio = 1                                 degrees of freedom =  129, 58

    Ha: ratio < 1               Ha: ratio != 1                 Ha: ratio > 1
  Pr(F < f) = 0.9356         2*Pr(F > f) = 0.1289           Pr(F > f) = 0.0644
```

Stata dispose également de commandes dites « immédiates » (voir § 9.2.1), pour lesquelles on se contente de fournir les données utiles pour construire la statistique de test. Dans le cas du test t de student, il s'agit de la commande `ttesti` dont la signature est reproduite ci-dessous :

```
ttesti #obs1 #mean1 #sd1 #obs2 #mean2 #sd2 [, options2]
```

Cette commande attend donc l'effectif, la moyenne et l'écart-type pour le 1er échantillon, et les mêmes informations pour le 2e échantillon. Les options (`options2`) correspondent aux options discutées précédemment pour le cas des variances inégales (`unequal` ou `welch`), ainsi qu'au niveau de risque $1 - \alpha$ (`level`).

9.1.2 Échantillons non indépendants

Dans le cas de deux échantillons appariés, on utilisera le même principe que celui évoqué pour deux séries de mesures représentées sous la forme de deux variables (cette fois-ci, les deux variables ont le même nombre d'observations), soit une formulation du type

```
. ttest x1 == x2
```

où `x1` et `x2` représentent les deux séries appariées. On supposera évidemment que les observations sont rangées dans le même ordre pour les deux variables. L'option `paired` est facultative dans ce cas.

9.1.3 Approche non-paramétrique

Le test de Wilcoxon pour échantillons indépendants, basé sur les rangs des observations, est obtenu avec la commande `ranksum`

```
. ranksum lwt, by(low)

Two-sample Wilcoxon rank-sum (Mann-Whitney) test

         low |      obs    rank sum    expected
-------------+---------------------------------
           0 |      130     13217.5       12350
           1 |       59      4737.5        5605
-------------+---------------------------------
    combined |      189       17955       17955

unadjusted variance   121441.67
adjustment for ties     -181.97
                     ----------
adjusted variance     121259.70

Ho: lwt(low==0) = lwt(low==1)
             z =   2.491
    Prob > |z| =   0.0127
```

Dans le cas de deux échantillons appariés, le test signé des rangs s'obtient avec la commande `signrank` et une syntaxe à peu près identique à celle du test t pour échantillons appariés :

```
. signrank x1 = x2
```

9.2 Comparaisons de deux proportions

9.2.1 Échantillons indépendants

Le test du χ^2 peut s'obtenir de plusieurs manières, généralement à partir de commandes permettant de construire un tableau de contingence. En utilisant `tabulate`, par exemple, on ajoutera l'option `chi`.

```
. tabulate low smoke, chi expected
```

```
+--------------------+
| Key                |
|--------------------|
|     frequency      |
| expected frequency |
+--------------------+

     Poids |
inférieur |         smoke
 à 2,5 kg |         0          1 |     Total
----------+----------------------+----------
        0 |        86         44 |       130
          |      79.1       50.9 |     130.0
----------+----------------------+----------
        1 |        29         30 |        59
          |      35.9       23.1 |      59.0
----------+----------------------+----------
    Total |       115         74 |       189
          |     115.0       74.0 |     189.0
```

```
          Pearson chi2(1) =   4.9237   Pr = 0.026
```

L'option `expected` permet d'afficher les effectifs théoriques (attendus sous l'hypothèse d'indépendance entre les deux variables) dans chaque cellule du tableau. Avec une option `exact`, Stata reporte également le résultat d'un test de Fisher.

On peut obtenir le même résultat en utilisant la commande immédiate `tabi`. Comme dans le cas du test de Student, il faut donner les informations essentielles pour la construction de la statistique de test. Dans le cas présent, il s'agit des effectifs pour les 4 cellules du tableau de contingence, soit

```
. tabi 86 44\ 29 30, chi2 exact nofreq

          Pearson chi2(1) =   4.9237   Pr = 0.026
           Fisher's exact =                 0.036
   1-sided Fisher's exact =                 0.020
```

On remarquera que la saisie des effectifs s'effectue par ligne, en séparant les lignes du tableau par le symbole `\`.

En ce qui concerne les tests de proportion pour un échantillon (généralement associés à l'hypothèse H_0 : $p = 0.5$), le test binomial est accessible via la commande `bitest` (la commande immédiate correspondante est `bitesti`). Si l'on souhaite utiliser une approximation par la loi normale, le test correspondant est obtenu avec `pretest` (`prtesti`). Cette commande fonctionne également dans le cas de deux échantillons.

Voici un exemple d'application en considérant la répartition des mères selon le statut fumeur ou non :

```
. bitest smoke == 0.5, detail
. prtest smoke == 0.5
    Variable |        N   Observed k   Expected k   Assumed p   Observed p
-------------+------------------------------------------------------------
       smoke |      189          74         94.5       0.50000      0.39153

  Pr(k >= 74)             = 0.998917  (one-sided test)
  Pr(k <= 74)             = 0.001754  (one-sided test)
  Pr(k <= 74 or k >= 115) = 0.003508  (two-sided test)

  Pr(k == 74)             = 0.000671  (observed)
  Pr(k == 114)            = 0.001029
  Pr(k == 115)            = 0.000671  (opposite extreme)

One-sample test of proportion                 smoke: Number of obs =      189
------------------------------------------------------------------------------
    Variable |       Mean   Std. Err.                     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       smoke |   .3915344   .0355036                      .3219487    .4611201
------------------------------------------------------------------------------
    p = proportion(smoke)                                         z =  -2.9823
Ho: p = 0.5

     Ha: p < 0.5                 Ha: p != 0.5                  Ha: p > 0.5
 Pr(Z < z) = 0.0014         Pr(|Z| > |z|) = 0.0029          Pr(Z > z) = 0.9986
```

Dans le premier cas, la quantité qui nous intéresse est celle intitulée `Pr(k <= 74 or k >= 115)`, et l'on retrouve son équivalent dans le test de proportion basé sur la loi normale sous `Pr(|Z| > |z|)`. Dans le cas de deux échantillons, la commande suivante permet de tester l'hypothèse que la répartition des mères fumeuses est la même quel que soit le statut du poids du bébé à la naissance (on pourra comparer le résultat du test bilatéral avec celui d'un test du χ^2).

```
. prtest smoke, by(low)

Two-sample test of proportions                     0: Number of obs =      130
                                                   1: Number of obs =       59
```

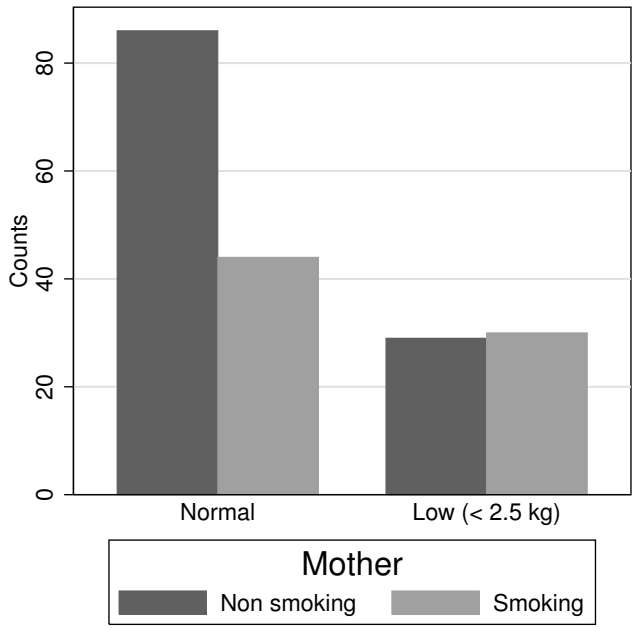
```
------------------------------------------------------------------------------
    Variable |       Mean   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           0 |   .3384615   .0415012                      .2571207    .4198024
           1 |   .5084746   .0650851                      .3809101     .636039
-------------+----------------------------------------------------------------
        diff |   -.170013   .0771908                     -.3213042   -.0187219
             |  under Ho:   .0766189    -2.22   0.026
------------------------------------------------------------------------------
        diff = prop(0) - prop(1)                                  z =  -2.2189
    Ho: diff = 0

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(Z < z) = 0.0132         Pr(|Z| < |z|) = 0.0265          Pr(Z > z) = 0.9868
```

On pourra afficher résumer n'importe quel tableau d'effectifs à partir des commandes `graph bar` (diagramme en barres) ou `graph dot` (diagramme en points). Voici un exemple de diagramme en barres, où l'on combine les approches utilisées p. 10 (création d'une variable auxiliaire pour le comptage des observations) et 14 (ajout d'étiquettes aux variables) :

```
. replace freq = 1 // la variable existe déjà
. graph bar (sum) freq, over(smoke, relabel(1 "Non smoking" 2 "Smoking")) ///
.   asyvars over(low, relabel(1 "Normal" 2 "Low (< 2.5 kg)")) ///
.   legend(title("Mother")) ytitle("Counts")

(0 real changes made)
```



Comme on le voit, les commandes de base de Stata ne sont pas toujours très intuitives pour travailler « graphiquement » avec des données catégorielles. On peut en revanche installer un package supplémentaire, en tapant

```
. ssc install catplot
```

à l'invite Stata. Cela suppose une connexion internet fonctionnelle. Moyennant quelques options de personnalisation du graphique (légende, étiquettes pour les modalités des variables, etc.), la commande précédente se résume alors à :

```
. catplot low smoke, recast(bar)
```

Voir l'aide en ligne pour plus de détails : `help catplot`.

9.2.2 Échantillons non indépendants

Dans le cas de deux variables binaires observées pour un même échantillons ou deux échantillons appariés (par exemple, étude cas-témoin), si l'on s'intéresse aux fréquences marginales du tableau de contingence résultant du croisement de ces deux variables, on peut utiliser le test de McNemar grâce à la commande `mcc`. La syntaxe suit les conventions de notation des études épidémiologiques (Cas/Contrôle, Exposé/Non exposé) et Stata fournit le degré de significativité exact et celui basé sur une loi $\chi^2(1)$. La commande immédiate est `mcci`.

9.3 Mesures de risque et odds-ratio

La plupart des mesures de risque ou d'association utilisées en épidémiologie se retrouve dans un sous-ensemble de commandes Stata spécifiques, appelées `epitab`. Elles sont également accessibles via le menu `Statistics > Epidemiology and related > Tables for epidemiologists`.

Par exemple, la commande `tabodds` s'utilise dans le cas des études cas-témoins ou des études transversales. Elle permet de calculer l'odds-ratio et son intervalle de confiance asymptotique, ainsi que tester l'homogénéité des OR entre strates (test de Mantel-Haenszel). Voici un exemple d'usage où l'on considère le statut de poids à la naissance (`low`) comme variable réponse et la variable `smoke` comme facteur de risque ou d'exposition :

```
. tabodds low smoke, or

-------------------------------------------------------------------------------
      smoke |  Odds Ratio       chi2       P>chi2     [95% Conf. Interval]
------------+------------------------------------------------------------------
          0 |    1.000000          .           .               .           .
          1 |    2.021944       4.90       0.0269      1.069897    3.821169
-------------------------------------------------------------------------------
Test of homogeneity (equal odds): chi2(1)  =     4.90
                                  Pr>chi2  =   0.0269

Score test for trend of odds:     chi2(1)  =     4.90
                                  Pr>chi2  =   0.0269
```

Il est possible d'indiquer à Stata quelle modalité de la 2e variable sert de catégorie de référence à l'aide de l'option `base()`, et également de changer le mode de calcul des intervalles de confiance (`cornfield` ou `woolf`).

Dans l'exemple précédent, on dispose des données individuelles, mais il arrive souvent que les données soient disponibles en format aggrégé, c'est-à-dire sous la forme d'un tableau de contingence que l'on peut également reformuler sous la forme d'un tableau à 3 entrées : les deux premières colonnes indiquent les croisements de chaque modalité des deux variables binaires, et la 3e colonne indique les effectifs associés. La syntaxe reste la même dans ce cas : on indique le nom des variables servant de réponse et de facteur explicatif mais on indique à Stata comment pondérer les traitements (croisement des deux modalités de chaque variable) par les effectifs renseignés dans une option `fweight=`. En supposant que les données soient présentées comme dans le tableau ci-dessous,

low	smoke	N
0	0	86
1	0	29
0	1	44
1	1	30

la syntaxe serait alors

```
. tabodds low smoke [fweight=N], or
```

Notons que dans ce type de configuration (tableau avec effectifs dans une colonne spécifique), les commandes graphiques pour réaliser des diagrammes en barres ou en points se simplifie légèrement : il n'est plus nécessaire de définir une variable de comptage et l'option `(sum)` pour cumuler les effectifs par niveaux des

variables, et on peut se contenter de la colonne d'effectif comme variable principale en spécificiant une option `(asis)`.

De même, la pondération par les effectifs avec une option de type `[fweight=N]` (que l'on peut abbréger en `[fw=N]`), où `N` désigne la variable contenant les effectifs est utilisable avec la commande `tabulate`. Un tel tableau peut être stocké dans un simple fichier texte incluant le nom des trois variables sur la 1re ligne du ficher (on utilisera alors `insheet` pour l'imorter), ou on peut saisir directement les données à l'aide de la commande `input`. Dans ce cas, les opérations peuvent être réalisées entre deux commandes `preserve/restore` pour ne pas perdre les données de la session en cours. On prendra garde au fait que les variables devront tout de même être renommées pour ne pas entrer en conflit avec celles présentes dans l'espace de travail. Dans l'illustration ci-dessous, après avoir taper la commande `preserve`, on a créé le tableau des données aggrégées en suffixant le nom des variables `low` et `smoke` par `b`.

```
. input lowb smokeb N

          lowb     smokeb          N
  1. 0 0 86
  2. 1 0 29
  3. 0 1 44
  4. 1 1 30
  5. end

. list lowb-N in 1/4

     +--------------------+
     | lowb   smokeb    N |
     |--------------------|
  1. |    0        0   86 |
  2. |    1        0   29 |
  3. |    0        1   44 |
  4. |    1        1   30 |
     +--------------------+
```

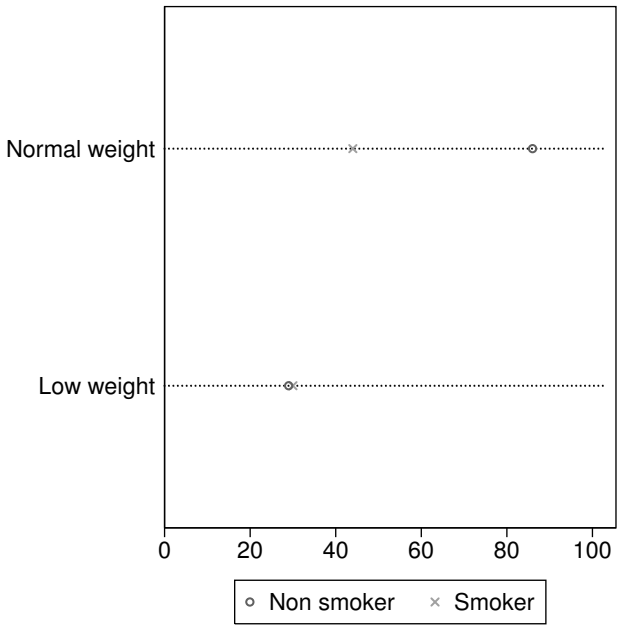
Avec l'exemple précédent, on écrirait donc

```
. tabulate lowb smokeb [fw=N]

           |        smokeb
      lowb |         0          1 |     Total
-----------+----------------------+----------
         0 |        86         44 |       130
         1 |        29         30 |        59
-----------+----------------------+----------
     Total |       115         74 |       189
```

On profitera de cet exemple pour illustrer quelques options de personnalisation sur les diagrammes en points, notamment la spécification d'un axe avec des unités choisies par l'utilisateur et des symboles différents pour souligner les niveaux de la variable de classification.

```
. label define status 0 "Normal weight" 1 "Low weight"
. label define smoker 0 "Non smoker" 1 "Smoker"
. label values lowb status
. label values smokeb smoker
. graph dot (asis) N, over(smokeb) asyvars over(lowb) ///
.   yscale(range(0 100)) ylabel(0(20)100) ///
.   marker(1, msymbol(oh)) marker(2, msymbol(X))
```



On n'oubliera pas de taper `restore` pour revenir au jeu de données initial.

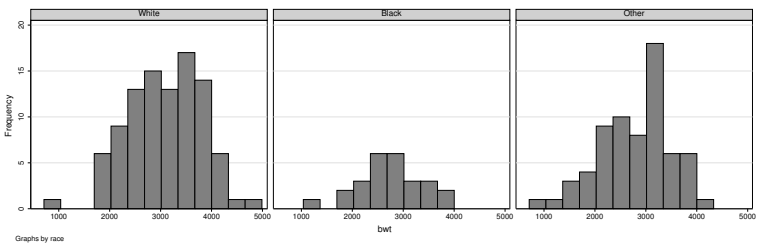
9.4 Analyse de variance

9.4.1 ANOVA à un facteur

La commande `summarize` (en combinaison avec `tabulate` ou `by:`) ou `tabstat` permet naturellement de résumer la distribution de la variable réponse en fonction des niveaux de la variable explicative. On verra toutefois que celle-ci peut être directement couplée à la commande produisant le tableau d'ANOVA. Concernant les méthodes graphiques, on peut toujours utiliser `graph bar` ou `graph dot` pour présenter visuellement la distribution des moyennes de groupe ; il suffit de remplacer la statistique de résumé `(sum)` par `(mean)`. Quant à la distribution des données individuelles, on peut construire des histogrammes d'effectifs (ou de fréquences) à l'aide de la commande `histogram`, en indiquant le facteur de classification dans une option `by()`.

Considérons les données portant sur les poids des bébés à la naissance (`bwt`) et l'origine ethnique des mères (`race`). La distribution des poids, en termes d'effectif (option `freq`), peut être obtenue de la manière suivante :

```
. histogram bwt, by(race, col(3)) freq
```



La commande principale pour réaliser une ANOVA *à un facteur* est `oneway`. Pour les modèles plus complexes, il faudra utiliser `anova` ou `regress`. Son utilisation est relativement simple : on fournit une liste de variables, en l'occurence la variable réponse puis la variable explicative. L'option `tabulate` ajoute automatiquement un tableau de résumé des moyennes et écart-types de groupe au tableau de décomposition des sources de variance.

```
. oneway bwt race, tabulate
```

```
           |          Summary of bwt
      race |        Mean   Std. Dev.       Freq.
-----------+------------------------------------
     White |   3102.7188   727.88615          96
     Black |   2719.6923   638.68388          26
     Other |   2805.2836   722.19436          67
-----------+------------------------------------
     Total |   2944.5873    729.2143         189

                       Analysis of Variance
    Source              SS         df      MS            F     Prob > F
------------------------------------------------------------------------
Between groups      5015725.25      2   2507862.63      4.91     0.0083
 Within groups      94953930.6    186   510505.003
------------------------------------------------------------------------
    Total           99969655.8    188   531753.488

Bartlett's test for equal variances:  chi2(2) =   0.6595  Prob>chi2 = 0.719
```

On remarquera que Stata indique également le résultat du test de Bartlett pour l'égalité des variances. Si l'on souhaite utiliser le test de Levene, on utilisera la commande `robvar` qui renvoie le résultat sous la statistique de test nommée `W0`.

```
. robvar bwt, by(race)
```

```
           |          Summary of bwt
      race |        Mean   Std. Dev.       Freq.
-----------+------------------------------------
     White |   3102.7188   727.88615          96
     Black |   2719.6923   638.68388          26
     Other |   2805.2836   722.19436          67
-----------+------------------------------------
     Total |   2944.5873    729.2143         189

W0  =  0.44717123   df(2, 186)     Pr > F = 0.64012002

W50 =  0.46842949   df(2, 186)     Pr > F = 0.62672105

W10 =  0.45725627   df(2, 186)     Pr > F = 0.63372775
```

On notera que la syntaxe est légèrement différente de celle utilisée avec `oneway` : il s'agit ici d'une commande à part entière pour les tests d'égalité de variance (comme `sdtest`, p. 16), elle n'est pas reliée directement au modèle d'ANOVA construit à l'aide de `oneway`.

9.4.2 Comparaisons de paires de moyennes

En ce qui concerne les comparaisons de l'ensemble des paires de moyennes (3 dans l'exemple précédent), le plus simple consiste à ajouter l'une des options de correction pour les tests multiples (`bonferroni`, `scheffe` ou `sidak`) lors de l'utilisation de `oneway`. Par souci de clarté, on a supprimé l'affichage du tableau d'ANOVA dans l'expression suivante.

```
. oneway bwt race, bonferroni noanova
```

```
                        Comparison of bwt by race
                              (Bonferroni)
Row Mean-|
Col Mean |      White      Black
---------+----------------------
   Black |   -383.026
         |      0.049
```

```
        |
  Other |   -297.435    85.5913
        |      0.029      1.000
```

On retrouvera le résultat pour la comparaison `White` vs. `Black` à partir d'un simple test t dont le degré de significativité est ajusté pour l'ensemble des comparaisons. La commande `ttest` renvoie en effet la statistique de test (`r(t)`) et la p-valeur (`r(p)`), comme on peut le vérifier en tapant `return list` après la première commande.

```
. quietly: ttest bwt if race != 3, by(race)
. display r(p)*3
```

```
.04853058
```

9.4.3 Test de tendance linéaire

Pour réaliser un test de tendance linéaire, l'approche par régression linéaire revient à remplacer la commande `oneway` par `regress`. Considérons la variable `ftv`, qui représente le nombre de visites chez le gynécologue durant le premier trimestre de grossesse. Cette variable prend des valeurs entre 0 et 6, les valeurs supérieures à 2 étant assez rarement observées. On peut recoder cette variable en une variable à 3 classes à l'aide de la commande `recode` dont la syntaxe est assez simple : les nouvelles modalités sont indiquées à côté des anciennes modalités (l'association se fait avec le symbole =) et le symbole / sert, comme dans le cas de l'opérateur `in`, à indiquer une gamme de valeurs (valeur de départ / valeur d'arrivée).

```
. recode ftv (0=0) (1=1) (2/6=2), gen(ftv2)
```

```
(12 differences between ftv and ftv2)
```

L'option `gen()` permet de générer une nouvelle variable. On pourra vérifier que le recodage s'est bien déroulé à l'aide d'un simple tri croisé des deux variables.

```
. tabulate ftv2 ftv
```

Sans autre indication, la variable `race` sera traitée comme une variable numérique, et en considérant que les distances entre niveaux sont égales (c'est le cas ici puisque les niveaux sont codés $\{1,2,3\}$), le test associé à la pente de la droite de régression fournit le résultat demandé.

```
. regress bwt ftv2
```

```
      Source |       SS       df       MS              Number of obs =     189
-------------+------------------------------           F(  1,   187) =    1.02
       Model |  542577.691     1  542577.691           Prob > F      =  0.3137
    Residual |  99427078.1   187  531695.605           R-squared     =  0.0054
-------------+------------------------------           Adj R-squared =  0.0001
       Total |  99969655.8   188  531753.488           Root MSE      =  729.17

------------------------------------------------------------------------------
         bwt |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        ftv2 |   66.09496   65.42879     1.01   0.314    -62.97845    195.1684
       _cons |   2898.775   69.78422    41.54   0.000      2761.11    3036.441
------------------------------------------------------------------------------
```

L'approche par la méthode des contrastes pour le test de tendance linéaire repose toujours sur l'utilisation de la commande `regress`, mais cette fois-ci on impose à Stata de traiter la variable `ftv2` comme une variable catégorielle en la préfixant par l'opérateur `i.` (voir l'aide en ligne, `help fvvarlist`). Ceci a pour effet de convertir la variable à $k = 3$ modalités en $k - 1 = 2$ variables indicatrices, codant pour les niveaux j ($j = 2, \ldots, k$) de la variable de classification. Comme les résultats de la régression en eux-mêmes ne nous intéresse pas vraiment, on supprime leur affichage en préfixant la commande de régression de l'instruction `quietly:` et on demande à Stata d'afficher les contrastes polynomiaux (orthogonaux) associés à la variable explicative. Cette dernière opération se réalise en préfixant le nom de la variable de groupement par l'opérateur `p.`.

```
. quietly: regress bwt i.ftv2
. contrast p.ftv2, noeffects

Contrasts of marginal linear predictions

Margins      : asbalanced

------------------------------------------------
             |         df           F        P>F
-------------+----------------------------------
        ftv2 |
   (linear)  |          1        0.41     0.5216
(quadratic)  |          1        2.55     0.1119
      Joint  |          2        1.79     0.1698
             |
    Residual |        186
------------------------------------------------
```

Le contraste d'intérêt est ici mentionné sous le nom `(linear)`.

On pourra vérifier que le coefficient de détermination, qui est retourné par la commande précédente non pas comme un résultat mais une post-estimation (voir `ereturn list`)

```
. display e(r2)

.01888498
```

correspond bien à la part de variance expliquée par le modèle d'ANOVA et que l'on peut obtenir de la manière suivante (plutôt que la calculer à partir des sommes de carré affichées par `oneway`) :

```
. anova bwt ftv2
                          Number of obs =     189     R-squared     =  0.0189
                          Root MSE      = 726.169     Adj R-squared =  0.0083

                  Source |  Partial SS    df       MS           F     Prob > F
              -----------+----------------------------------------------------
                   Model |  1887925.36     2  943962.682       1.79     0.1698
                         |
                    ftv2 |  1887925.36     2  943962.682       1.79     0.1698
                         |
                Residual |  98081730.4   186  527321.131
              -----------+----------------------------------------------------
                   Total |  99969655.8   188  531753.488
```

9.4.4 Utilisation de contrastes

De manière plus générale, il est également possible d'estimer, voire tester, n'importe quel contraste à partir d'une commande `regress`. Pour cela, on utilisera la commande `lincom`. Voici un exemple avec le modèle initial (poids des bébés à la naissance et ethnicité des mères) :

```
. regress bwt i.race

      Source |       SS       df       MS              Number of obs =     189
-------------+------------------------------           F(  2,   186) =    4.91
       Model |  5015725.25     2  2507862.63           Prob > F      =  0.0083
    Residual |  94953930.6   186  510505.003           R-squared     =  0.0502
-------------+------------------------------           Adj R-squared =  0.0400
       Total |  99969655.8   188  531753.488           Root MSE      =   714.5

------------------------------------------------------------------------------
         bwt |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
```

```
-------------+----------------------------------------------------------------
        race |
          2  |  -383.0264   157.9638    -2.42   0.016    -694.6575    -71.3954
          3  |  -297.4352    113.742    -2.61   0.010    -521.8254   -73.04498
             |
       _cons |   3102.719   72.92298    42.55   0.000     2958.856    3246.581
------------------------------------------------------------------------------
```

On voit que Stata fournit un coefficient de régression par niveau de la variable `race`, à l'exception du 1er qui sert de catégorie de référence. Le terme d'ordonnée à l'origine représente donc le poids moyen des bébés dont les mères sont de type `White`, et chacun des deux coefficients de régression représente la déviation entre les groupes `Black` et `Other` par rapport au groupe `White`. La différence de moyenne entre les deux groupes `Black` et `Other` peut être estimée de la manière suivante (Stata « numérote » les coefficients de régression à partir des codes numériques des niveaux des facteurs, en commençant à 1) :

```
. lincom 3.race - 2.race

 ( 1)  - 2.race + 3.race = 0

------------------------------------------------------------------------------
         bwt |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   85.59127   165.0887     0.52   0.605    -240.0958    411.2783
------------------------------------------------------------------------------
```

De même, on sait que l'on peut utiliser la commande `ci` pour former un intervalle de confiance pour une moyenne en utilisant la loi normale (§ 8.3.1, p. 9), par exemple

```
. ci bwt if race == 1

    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
         bwt |         96    3102.719    74.28957        2955.235    3250.202
```

Une manière de construire un intervalle de confiance en utilisant la loi de Student est alors :

```
. lincom _cons + 1.race

 ( 1)  1b.race + _cons = 0

------------------------------------------------------------------------------
         bwt |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   3102.719   72.92298    42.55   0.000     2958.856    3246.581
------------------------------------------------------------------------------
```

Attention à ne pas oublier le terme d'ordonnée à l'origine dans l'expression ci-dessus !

9.4.5 Approche non-paramétrique

L'alternative non-paramétrique à l'ANOVA discutée ci-dessus, ou ANOVA de Kruskal-Wallis, s'obtient à l'aide de la commande `kwallis`. La syntaxe est légèrement différente de celle de `oneway`, `anova` ou `regress`, et le facteur de classification figure cette fois dans une option `by()`. Voici le résultat de l'ANOVA basée sur les rangs avec les mêmes données de poids à la naissance (`bwt`) et d'ethnicité des mères (`race`).

```
. kwallis bwt, by(race)

Kruskal-Wallis equality-of-populations rank test

  +------------------------+
  |  race | Obs | Rank Sum |
  |-------+-----+----------|
  | White |  96 | 10189.00 |
```

```
  | Black |  26 |  2015.00 |
  | Other |  67 |  5751.00 |
  +------------------------+

chi-squared =     8.519 with 2 d.f.
probability =     0.0141

chi-squared with ties =     8.520 with 2 d.f.
probability =     0.0141
```

Il est toujours possible de compléter cette analyse par des tests multiples sur les paires de niveaux de la variable explicative, via la commande `ranksum` discutée en § 9.1.3 (p. 17). Pour isoler deux groupes parmi l'ensemble des groupes d'unités statistiques définis par la variable `race`, il suffit par exemple d'exclure le 3e à l'aide d'un filtre de type `if race != 3` (pour la comparaison entre les groupes `White` et `Black`, par exemple). Une alternative plus économique consiste à installer la commande externe `kwallis2` (taper `findit kwallis2` et suivre les instructions pour l'installation). La syntaxe est identique à celle de `kwallis`, mais cette commande fournit automatiquement l'ensemble des tests de Wilcoxon associés au modèle.

9.4.6 ANOVA à deux facteurs

Les ANOVA à plusieurs critères de classification sont réalisées à partir de la commande `anova`, plus complexe d'utilisation que `oneway` mais permettant de définir des termes d'interaction ou tester des contrastes spécifiques. Un exemple d'utilisation avec les variables `ht` (antécédents d'hypertension chez la mère) et `race` (ethnicité de la mère), en considérant toujours le poids des bébés (`bwt`) comme variable réponse, est fourni ci-après. On considèrera un modèle incluant un terme d'interaction, celle-ci étant symbolisé par `##` sous Stata. Avec la notation `race##ht`, on demande à Stata de considérer deux facteurs et leur interaction (avec `oneway` et `anova`, il n'est pas nécessaire d'indiquer à Stata que les variables doivent explicitement être représentées sous la forme de variables qualitatives).

Les résultats du modèle d'ANOVA sont indiqués ci-dessous.

```
. anova bwt race##ht
```

```
                          Number of obs =     189     R-squared     =  0.0768
                          Root MSE      = 710.143     Adj R-squared =  0.0516

                   Source |  Partial SS    df       MS           F     Prob > F
                ----------+----------------------------------------------------
                    Model |  7682087.67     5  1536417.53       3.05     0.0115
                          |
                     race |  2992590.25     2  1496295.12       2.97     0.0539
                       ht |  1757257.26     1  1757257.26       3.48     0.0635
                  race#ht |  889649.132     2  444824.566       0.88     0.4157
                          |
                 Residual |  92287568.1   183  504303.651
                ----------+----------------------------------------------------
                    Total |  99969655.8   188  531753.488
```

Pour calculer des sommes de carré de manière séquentielle (comme le fait R par défaut), il est impératif d'ajouter l'option `sequential`.

Une formulation équivalente du modèle ci-dessus et faisant apparaître explicitement les deux effets principaux et l'effet d'interaction serait

```
. anova bwt race ht race#ht
```

Les statistiques de résumé numérique peuvent se construire à partir des mêmes indicateurs (moyenne, écart-type, etc.) que dans le cas où un seul facteur de classification est étudié. Par contre, la commande `tabstat` ne fonctionne qu'avec une variable de classification. On peut en revanche utiliser la commande `table` de la manière suivante :

```
. table race, by(ht) contents(mean bwt sd bwt count bwt)
```

```
-----------------------------------------
ht and    |
race      |  mean(bwt)     sd(bwt)      N(bwt)
----------+------------------------------------
No        |
    White |    3110.89    726.7152          91
    Black |    2751.83    542.1708          23
    Other |    2852.41    704.9773          63
----------+------------------------------------
Yes       |
    White |       2954    819.4086           5
    Black |    2473.33    1327.633           3
    Other |       2063    649.5717           4
-----------------------------------------
```

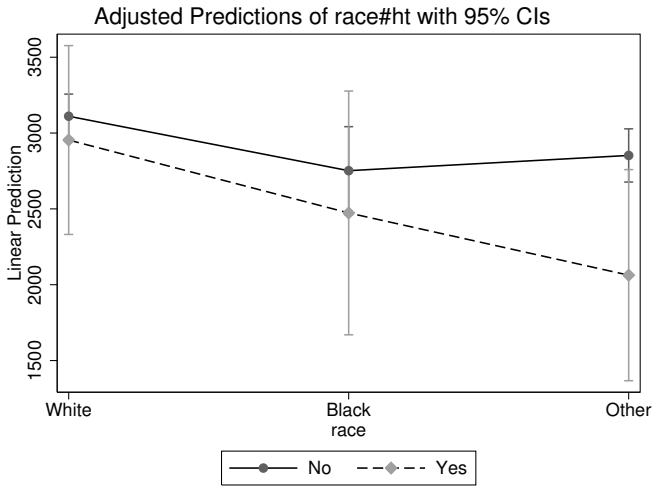
ou

```
. table race ht, contents(mean bwt sd bwt count bwt)
```

Un graphique d'interaction peut être construit à partir de la commande externe `anovaplot` (à télécharger et installer, `findit anovaplot`), ou à partir des commandes graphiques basées sur la commande `margins` introduites dans les versions récentes de Stata. Voici une solution possible :

```
. quietly: margins race#ht
. marginsplot

  Variables that uniquely identify margins: race ht
```



Ce qu'il faut retenir

- Les commandes `ttest` et `ranksum` (ou `signrank` dans le cas apparié) fournissent les tests de comparaison pour deux échantillons, indépendants ou non, en considérant une variable réponse numérique.
- Les commandes `bitest` et `prtest` permettent de tester des hypothèses portant des proportions calculées sur un ou deux échantillons, en utilisant la loi binomiale ou son approximation par la loi normale.
- La plupart des commandes de test pour un ou deux échantillons peuvent être utilisées à partir des statistiques de résumé (moyenne, écart-type, proportion), sans recourir aux données complètes : il s'agit des commandes dites immédiates.
- La commande `tabulate` dispose de deux options (`chi2` et `exact`) fournissant les statistiques du χ^2 ou de Fisher dans le cas d'un tableau de contingence, alors que les commandes `epitab` telle que `tabodds` permettent de calculer l'odds-ratio, éventuellement en tenant compte d'un facteur de stratification.
- Les commandes relatives au modèle d'ANOVA sont `oneway` (cas à un seul facteur de classification) et `anova` (ANOVA à plusieurs facteurs), et elles incluent des options (`bonferroni`, dans le cas de `oneway`) ou disposent de commandes associées (`contrast`, dans le cas de `anova`) pour réaliser des comparaisons multiples de moyennes, travailler sur des contrastes spécifiques ou résumer des effets marginaux (utiles pour la construction de graphique d'interaction, par exemple).

Cours 10. Régression linéaire et logistique

Sommaire

Dans ce chapitre on aborde les mesures d'association entre deux variables numériques (corrélation linéaire et corrélation de rangs) et le modèle de régression linéaire : estimation des coefficients de la droite de régression, ajustement et prédiction sur de nouvelles données avec intervalles de confiance. Ensuite, après avoir passé en revue les principales mesures de risque dans les études épidémiologiques et les études diagnostiques, on s'intéresse à la modélisation d'une variable binaire en fonction de variables explicatives numériques ou binaires à l'aide du modèle de régression logistique.

10.1 Mesures d'association entre deux variables numériques

10.1.1 Statistiques descriptives bivariées

La commande `summarize` fournit des informations sur la tendance centrale, la dispersion et l'étendue des valeurs observées pour une liste de variables. Il est donc tout à fait possible de l'utiliser pour résumer la distribution de deux variables numériques. Dans le cas du poids des bébés (`bwt`) et des mères (`lwt`), on a les résultats suivants (les poids des mères sont exprimés en livres, et on les convertit dans un premier temps en kilogrammes).

```
. replace lwt = lwt/2.2
. summarize bwt lwt
```

```
lwt was int now float
(189 real changes made)
```

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         bwt |       189    2944.587    729.2143        709       4990
         lwt |       189    59.00673    13.89972   36.36364   113.6364
```

On se rappelera que l'option `detail` permet de fournir des informations plus détaillées (quartiles, symétrie et aplatissement, etc.).

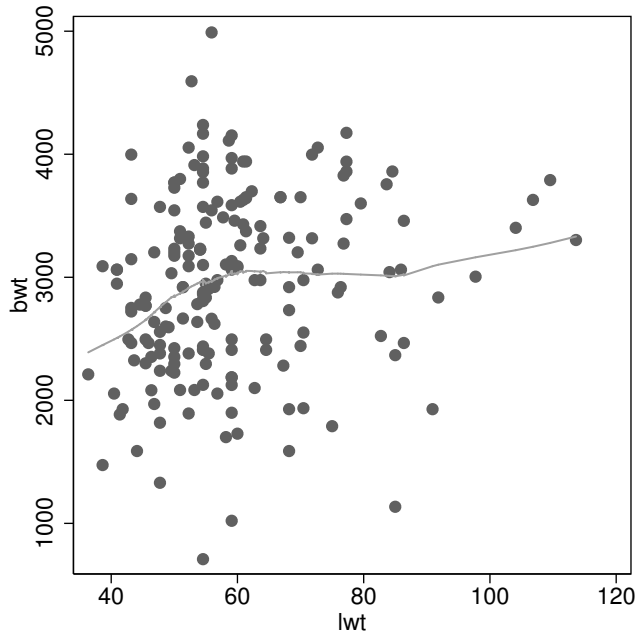
Avant de calculer n'importe quelle mesure d'association linéaire ou monotonique, il est conseillé d'inspecter visuellement le diagramme de dispersion qui décrit la co-variation entre les deux séries de mesures. Pour cela, on utilise l'une des commandes `twoway`, en l'occurence `scatter` en indiquant les deux variables d'intérêt : la première sera reportée sur l'axe des ordonnées et la seconde sur l'axe des abscisses.

```
. twoway scatter bwt lwt
```

Il est intéressant de superposer sur le graphique précédent une courbe loess qui donne une idée de la linéarité de la relation entre les deux variables, et de possibles écarts locaux. Pour cela, il suffit de combiner deux commandes `twoway`, en les délimitant par des parenthèses. Dans le cas suivant, la première commande,

`(scatter bwt lwt)`, affiche le diagramme de dispersion, et la seconde, `(lowess bwt lwt)`, une courbe loess. On pourrait remplacer `lowess` par `lfit` pour tracer la droite de régression.

```
. twoway (scatter bwt lwt) (lowess bwt lwt), legend(off) ytitle("bwt")
```



Notons qu'avec les versions récentes de Stata l'expression ci-dessus peut se formuler

```
. twoway scatter bwt lwt || lowess bwt lwt, legend(off) ytitle("bwt")
```

c'est-à-dire que l'opérateur `||` est utilisé pour séparer les instructions graphiques `twoway` multiples.

Dans le cas où on s'intéresse à plus de deux variables, la commande `graph matrix`, suivie de la liste des variables d'intérêt, permet d'afficher l'ensemble des diagrammes de dispersion croisant les variables deux par deux.

10.1.2 Corrélation de Bravais-Pearson

La commande fournissant le coefficient de corrélation de Bravais-Pearson entre deux variables numériques est `correlate`.

```
. correlate lwt bwt
(obs=189)

             |      lwt      bwt
-------------+------------------
         lwt |   1.0000
         bwt |   0.1857   1.0000
```

Notons que l'on pourrait ajouter une option `means` à la commande précédente afin d'afficher simultanément les résultats produits par `summarize`. Il est également possible d'utiliser la commande `pwcorr`, normalement réservé au cas de corrélations entre plus de deux variables, mais qui a l'avantage de fournir le résultat du test d'hypothèse concernant la nullité du paramètre d'intérêt.

```
. pwcorr lwt bwt, obs sig
             |      lwt      bwt
-------------+------------------
         lwt |   1.0000
             |
             |      189
```

```
           |
       bwt |   0.1857   1.0000
           |   0.0105
           |      189      189
           |
```

La commande `corrci` fournit une estimation de l'intervalle de confiance pour le coefficient de Bravais-Pearson (par défaut, en utilisant la transformation inverse de Fisher). L'option `level()` permet de modifier le niveau de confiance désiré.

```
. corrci lwt bwt

(obs=189)

          correlation and 95% limits
lwt bwt      0.186    0.044    0.320
```

10.1.3 Corrélation non-paramétrique

Si l'on préfère travailler avec une mesure de corrélation basée sur les rangs des observations, on utilisera le coefficient de corrélation de Spearman grâce à la commande `spearman`.

```
. spearman lwt bwt

 Number of obs =     189
Spearman's rho =       0.2489

Test of Ho: lwt and bwt are independent
    Prob > |t| =       0.0006
```

La commande `spearman` fonctionne également avec plus de variables et renvoie, comme `pwcorr`, une matrice de coefficients de corrélation avec éventuellement le degré de significativité (ajouter l'option `stats(rho p)`).

10.2 Régression linéaire

10.2.1 Estimation des paramètres du modèle

On a vu au chapitre précédent que la commande `regress` (p. 24) est utilisée lorsque l'on s'intéresse à modéliser la relation entre deux variables numériques, l'une étant considérée comme une variable réponse. Dans le cas présent, pour modéliser la relation entre le poids des bébés (variable réponse) et le poids des mères (variable explicative), on utiliserait donc la syntaxe suivante :

```
. regress bwt lwt
```

```
      Source |       SS       df       MS              Number of obs =     189
-------------+------------------------------           F(  1,   187) =    6.68
       Model |   3448639.3     1   3448639.3           Prob > F      =  0.0105
    Residual |  96521016.5   187  516155.168           R-squared     =  0.0345
-------------+------------------------------           Adj R-squared =  0.0293
       Total |  99969655.8   188  531753.488           Root MSE      =  718.44

------------------------------------------------------------------------------
         bwt |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         lwt |   9.744038   3.769686     2.58   0.011     2.307461    17.18061
       _cons |   2369.623   228.4932    10.37   0.000     1918.868    2820.379
------------------------------------------------------------------------------
```

Pour tout modèle de régression sous Stata, la variable réponse vient en premier, suivie de la ou des variables explicatives. Cette commande fournit le tableau d'analyse de variance pour la régression ainsi que le tableau des

coefficients de régression (avec intervalles de confiance), dont on peut supprimer l'affichage à l'aide de l'option `notable`. Il est toujours possible de réafficher les résultats du modèle de régression en tapant simplement le nom de la commande, `regress` (ceci est valable pour les autres modèles de régression sous Stata).

Il est également possible de stocker ou d'afficher uniquement le coefficient de régression pour la variable `lwt` (pente de la droite de régression) en exploitant les valeurs retournées par Stata. Par exemple, l'instruction suivante affiche le résultat demandé :

```
. display _b[lwt]

9.7440378
```

Il s'agit d'un résultat dit de *post-estimation*. La commande `ereturn list` fournit la liste des valeurs de post-estimation stockées par Stata. Dans ce cas précis, les coefficients de régression (ordonnée à l'origine et pente) sont enregistrés dans un objet appelé `e(b)`. Si l'on modifie légèrement l'instruction `regress` précédente, on peut vérifier que ces valeurs sont accessibles individuellement :

```
. regress bwt lwt, noheader coeflegend

------------------------------------------------------------------------------
         bwt |      Coef.  Legend
-------------+----------------------------------------------------------------
         lwt |   9.744038  _b[lwt]
       _cons |   2369.623  _b[_cons]
------------------------------------------------------------------------------
```

On peut également afficher la valeur du coefficient de détermination en utilisant `e(r2)`. Dans l'illustration suivante, on utilise la commande `display` avec une combinaison de texte et de valeur numérique (arrondie à 2 décimales).

```
. display "Coefficient de détermination = " %3.2f e(r2)*100 " %"

Coefficient de détermination = 3.45 %
```

On remarquera que lorsqu'on utilise `display`, le formatage des résultats numériques affichés à l'écran peut être réalisé en faisant précéder le nom de la variable dont on souhaite afficher le contenu par une instruction de formatage de type `%x.yf` (`x` positions, dont `y` décimales).

La droite de régression peut être représentée graphiquement à l'aide de la commande `twoway lfit bwt lwt`, mais comme on l'a indiqué à la section précédente, on peut combiner cette commande avec un simple diagramme de dispersion, par exemple

```
. twoway (scatter bwt lwt) (lfit bwt lwt)
```

10.2.2 Prédiction ponctuelle et par intervalle

Sous Stata, le principe général consiste à estimer les paramètres d'un modèle de régression et à travailler ensuite à partir de commandes de post-estimation. Cela est valable pour le calcul des valeurs prédites ou des résidus du modèle de régression. Si l'on souhaite calculer les valeurs ajustées pour le modèle (valeurs prédites de `bwt` pour les valeurs de `lwt` observées), on utilisera la commande `predict` après une commande d'estimation telle que `regress`. Les valeurs prédites correspondront toujours au dernier modèle de régression estimé.

```
. predict double p, xb
```

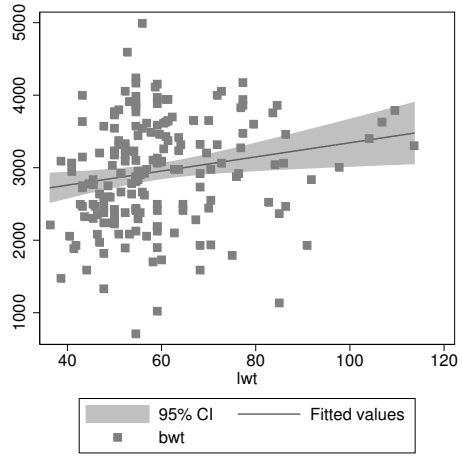
L'option `xb` (qui est celle par défaut) fournit les valeurs ajustées pour le modèle précédent. Il est important de ne pas oublier de préciser un nom de variable pour stocker les prédictions ! L'option `double` permet de limiter la taille de stockage en mémoire des valeurs prédites.

Les commandes précédentes ne fournissent pas d'intervalles de confiance. Cependant, il n'est pas difficile d'obtenir l'erreur standard pour les valeurs ajustées, et de calculer à partir des valeurs prédites les intervalles de confiance associés. Prenons le cas des valeurs ajustées :

```
. predict sep, stdp
. generate lci = p - 1.96*sep
. generate uci = p + 1.96*sep
```

Les variables `sep`, `lci` et `uci` correspondent à l'erreur standard, et aux bornes inférieure et supérieure de l'IC à 95 %, respectivement. Ces valeurs pourraient être utilisées pour afficher manuellement la droite de régression et son intervalle de confiance à 95 % (la commande `line` permet de tracer des lignes sous Stata), mais il est plus simple et plus rapide d'utiliser la commande `lfitci` comme indiqué ci-dessous.

```
. twoway (lfitci bwt lwt) (scatter bwt lwt)
```



On remplacera l'option `stdp` par `stdf` pour obtenir l'erreur standard de prédiction pour de nouvelles observations.

10.2.3 Diagnostic du modèle

La commande `estat` fournit un certain nombre d'informations concernant la qualité d'ajustement du modèle, et permet de diagnostiquer les éventuels problèmes de colinéarité (`estat vif`) dans le cas où le modèle inclut plusieurs variables explicatives.

```
. estat ic

-----------------------------------------------------------------------------
       Model |    Obs    ll(null)   ll(model)     df          AIC         BIC
-------------+---------------------------------------------------------------
           . |    189    -1513.56   -1510.242      2     3024.485    3030.968
-----------------------------------------------------------------------------
               Note:  N=Obs used in calculating BIC; see [R] BIC note
```

Il est également possible d'utiliser la commande externe `fitstat` (à installer depuis internet, `findit fitstat`) pour un résumé plus détaillé de la qualité d'ajustement du modèle.

```
. fitstat

Measures of Fit for regress of bwt

Log-Lik Intercept Only:      -1513.560   Log-Lik Full Model:          -1510.242
D(187):                       3020.485   LR(1):                           6.635
                                         Prob > LR:                       0.010
R2:                              0.034   Adjusted R2:                     0.029
AIC:                            16.003   AIC*n:                        3024.485
BIC:                          2040.278   BIC':                           -1.393
BIC used by Stata:            3030.968   AIC used by Stata:            3024.485
```

Pour obtenir les résidus du modèle de régression (écart entre valeurs observées et valeurs prédites), on utilise toujours la commande `predict`, en précisant cette fois une option parmi : `residuals` (résidus bruts), `rstandard`

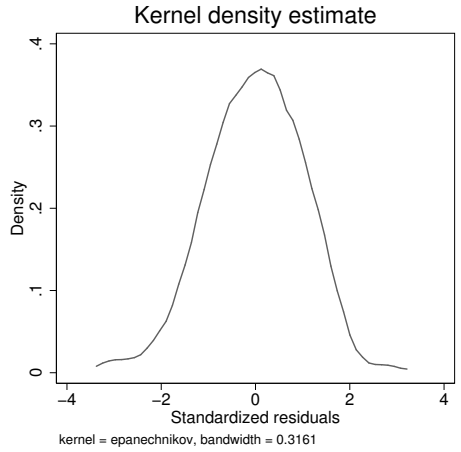
(résidus standardisés) ou `rstudent` (résidus studentisés). Dans la série d'instructions qui suit, on calcule les trois types de résidus, et on affiche leur résumé numérique à l'aide de `summarize` après avoir limité l'affichage numérique à 5 décimales.

```
. predict double r, rstandard
. predict double rr, residuals
. predict double rrr, rstudent
. format r-rrr %9.5f
. summarize r-rrr, format
```

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
           r |       189    -0.00044     1.00210   -3.06017    2.89709
          rr |       189    -0.00000   716.52611  -2.19e+03   2.08e+03
         rrr |       189    -0.00107     1.00806   -3.13139    2.95644
```

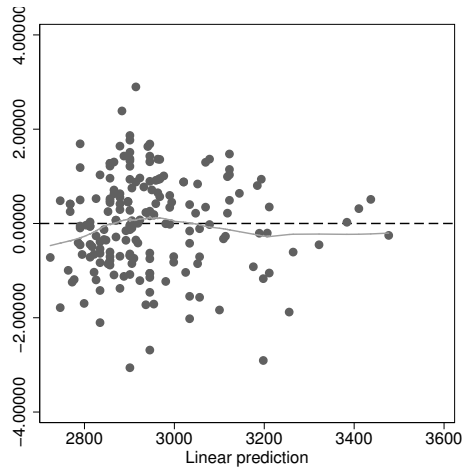
Pour afficher la distribution des résidus, on peut utiliser un histogramme d'effectifs via la commande `histogram`, ou une représentation sous forme de courbe de densité. En voici une illustration :

```
. kdensity r
```



Il est également intéressant de regarder la distribution des résidus en fonction des valeurs prédites pour vérifier la constance de la variance et l'absence de patterns spécifiques de variation des résidus. Pour cela, il suffit de combiner deux commandes `twoway`, sur le même principe qu'en § 10.1.1.

```
. twoway (scatter r p) (lowess r p), yline(0, lcolor(black) lpattern(dash)) legend(off)
```



En fait, le même graphique peut être obtenu en utilisant directement la commande `rvfplot` qui fournit une représentation des résidus en fonction des valeurs prédites.

```
. rvfplot, mlabel(smoke)
```

10.2.4 Régression linéaire multiple

L'extension au modèle de régression multiple ne pose pas vraiment de problème du point de vue des instructions : on indique à la commande `regress` le nom de la variable réponse suivi de l'ensemble des variables explicatives. Comme il est souvent utile de transformer certaines variables, ou de les centrer sur leur moyenne, on profite de cette section pour indiquer comment centrer une variable explicative. Sachant que lorsqu'on utilise `summarize` cette commande génère un certain nombre d'informations que l'on peut utiliser par la suite (voir `return list`), il n'est pas difficile de centrer la variable relative au poids des mères sur sa moyenne en procédant de la manière indiquée ci-dessous.

```
. quietly: summarize lwt
. generate lwts = lwt - r(mean)
```

Notons que si l'on souhaitait standardiser la variable `lwt` (c'est-à-dire, non seulement soustraire la moyenne à chaque observation mais également normaliser par l'écart-type), on remplacerait l'expression ci-dessus par

```
. generate lwts = (lwt - r(mean)) / r(sd)
```

En utilisant les commandes `egen`, le même résultat pourrait être obtenu avec une commande de type

```
. egen lwts = std(lwt)
```

Finalement, le modèle de régression incluant le poids des mères centrés sur leur moyenne et la fréquence des visites chez le gynécologue durant le 1er trimestre de grossesse s'écrirait :

```
. regress bwt lwts ftv i.race, noheader
```

```
------------------------------------------------------------------------------
         bwt |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        lwts |   10.14028   3.891895     2.61   0.010     2.461807    17.81876
         ftv |   11.82137   49.17071     0.24   0.810    -85.18953    108.8323
             |
        race |
          2  |  -450.1195   158.1307    -2.85   0.005    -762.1019   -138.1371
          3  |  -239.2497   114.4963    -2.09   0.038    -465.1442   -13.35526
             |
       _cons |    3081.94   83.91919    36.73   0.000     2916.372    3247.507
------------------------------------------------------------------------------
```

On remarquera l'usage de l'opérateur `i.` pour indiquer à Stata de traiter la variable `race` comme une variable qualitative, comme indiqué dans le chapitre précédent (p. 24).

10.3 Mesures d'association en épidémiologie

10.3.1 Études pronostiques et mesures de risque

En plus de `tabodds` vu au chapitre précédent (§ 9.3), Stata fournit la commande `mhodds`, pour les études cas-témoins et transversales.

```
. mhodds low smoke
```

```
Maximum likelihood estimate of the odds ratio
Comparing smoke==1 vs. smoke==0
```

```
------------------------------------------------------------------
 Odds Ratio    chi2(1)        P>chi2        [95% Conf. Interval]
------------------------------------------------------------------
   2.021944       4.90        0.0269         1.069897   3.821169
------------------------------------------------------------------
```

Considérons 4 groupes d'âge pour les mères et réalisons un test de Maentel-Haenszel pour obtenir une estimation de l'odds-ratio en contrôlant pour le facteur âge.

```
. xtile age4 = age, nq(4)
. table low smoke age4
```

```
----------------------------------------------------------
          |             4 quantiles of age and smoke
          | ---- 1 ---    ---- 2 ---    ---- 3 ---    ---- 4 ---
      low |    0     1       0     1       0     1       0     1
----------+-----------------------------------------------------
        0 |   20    16      26    10      15     6      25    12
        1 |    8     7       8    12      10     5       3     6
----------------------------------------------------------
```

On peut utiliser `mhodds low smoke age4` pour obtenir directement l'odds-ratio commun, mais en spécifiant le facteur de stratification dans une option `by()` Stata fournit les estimations par strate en plus de l'estimation commune.

```
. mhodds low smoke, by(age4)
```

```
Maximum likelihood estimate of the odds ratio
Comparing smoke==1 vs. smoke==0
by age4

------------------------------------------------------------------------------
     age4 | Odds Ratio        chi2(1)         P>chi2       [95% Conf. Interval]
----------+-------------------------------------------------------------------
        1 |   1.093750           0.02         0.8856         0.32257    3.70863
        2 |   3.900000           5.50         0.0191         1.14267   13.31098
        3 |   1.250000           0.09         0.7630         0.29217    5.34783
        4 |   4.166667           3.48         0.0619         0.81731   21.24180
------------------------------------------------------------------------------

    Mantel-Haenszel estimate controlling for age4
    ------------------------------------------------------------
     Odds Ratio    chi2(1)        P>chi2        [95% Conf. Interval]
    ------------------------------------------------------------
       2.138616       5.59        0.0181         1.121338   4.078767
    ------------------------------------------------------------

Test of homogeneity of ORs (approx): chi2(3)   =    3.36
                                     Pr>chi2   =  0.3399
```

Dans le cas présent, on travaille à partir de données individuelles, mais cette commande fonctionne également à partir d'un tableau d'effectif. Pour cela, on spécifiera la pondération par les effectifs dans une option `fweight` (voir `help mhodds` pour des exemples d'utilisation).

En termes de visualisation des données du tableau stratifié, on peut construire assez facilement une série de diagrammes en barres à l'aide de la commande `catplot` (p. 19). Pour faciliter la lecture du graphique, il est nécessaire d'ajouter des étiquettes aux trois variables manipulées : `low`, `smoke` et `age4`. Pour cette dernière, on a besoin de connaître les bornes des intervalles de classe que la commande `xtile` a utilisées. On peut obtenir cette information à partir de la commande `_pctile` de la manière suivante. Notons que les bornes extrêmes ne sont pas incluses dans l'affichage, mais on peut vérifier à partir de `summarize age` quelles sont les valeurs minimale et maximale pour cette variable. On retiendra également que les bornes affichées ci-dessous sont inclusives.

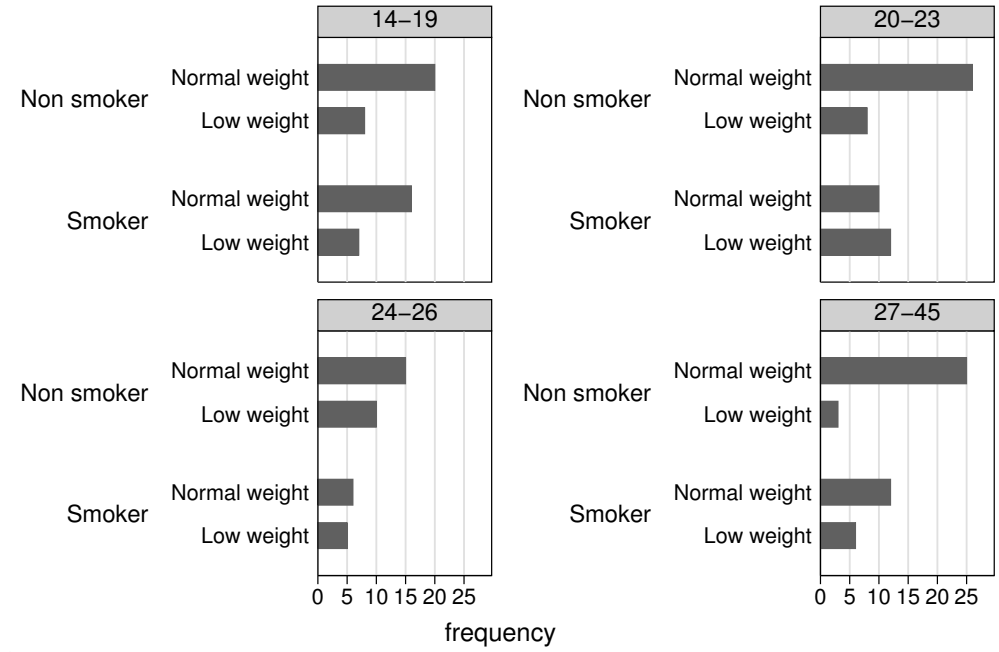
```
. _pctile age, n(4)
. return list
```

```
scalars:
                 r(r1) =  19
```

```
r(r2) =  23
r(r3) =  26
```

À partir de là, on peut créer un jeu d'étiquettes pour les trois variables, et afficher la distribution des effectifs correspondant au tableau à 3 dimensions. On ajoutera une option `percent` à la commande `catplot` si l'on préfère afficher des proportions plutôt que des effectifs.

```
. label define agec 1 "14-19" 2 "20-23" 3 "24-26" 4 "27-45"
. label values age4 agec
. label define wght 0 "Normal weight" 1 "Low weight"
. label values low wght
. label define smoking 0 "Non smoker" 1 "Smoker"
. label values smoke smoking
. catplot low smoke, by(age4)
```



Graphs by 4 quantiles of age

Les commandes `epitab` fourniront le même résultat concernant le calcul de l'odds-ratio. Par exemple, avec la commande `cc` pour les études cas-témoins on obtiendrait :

```
. cc low smoke, by(age4)

4 quantiles of a |       OR       [95% Conf. Interval]   M-H Weight
-----------------+-------------------------------------------------
           14-19 |    1.09375      .2719158   4.315057     2.509804 (exact)
           20-23 |        3.9       1.06682   14.50878     1.428571 (exact)
           24-26 |       1.25        .23063   6.531024     1.666667 (exact)
           27-45 |   4.166667       .713997   29.26378     .7826087 (exact)
-----------------+-------------------------------------------------
           Crude |   2.021944      1.029092   3.965864               (exact)
    M-H combined |   2.138616      1.130227    4.04669
-----------------------------------------------------------------------
Test of homogeneity (M-H)      chi2(3) =     3.48  Pr>chi2 = 0.3237

                   Test that combined OR = 1:
                               Mantel-Haenszel chi2(1) =      5.59
                                                Pr>chi2 =    0.0181
```

ou, si l'on ne considère pas la variable `age4` :

```
. cc low smoke , woolf
```

```
                 | smoke                   |             Proportion
                 |   Exposed   Unexposed   |      Total     Exposed
-----------------+-------------------------+-------------------------
           Cases |        30          29   |         59       0.5085
        Controls |        44          86   |        130       0.3385
-----------------+-------------------------+-------------------------
           Total |        74         115   |        189       0.3915
                 |                         |
                 |      Point estimate     |    [95% Conf. Interval]
                 |-------------------------+-------------------------
      Odds ratio |         2.021944        |     1.08066    3.783112 (Woolf)
 Attr. frac. ex. |         .5054264        |    .0746392    .7356673 (Woolf)
 Attr. frac. pop |         .2569965        |
                 +-----------------------------------------------------
                              chi2(1) =     4.92  Pr>chi2 = 0.0265
```

La variable réponse est toujours placée en première position, suivi du facteur d'exposition et des variables d'ajustement. Pour obtenir une mesure de risque relatif, on remplacera `cc` par `cs` lorsque cela s'applique (étude de cohorte, voire études transversales).

```
. tabulate low smoke, col nofreq
. display 40.54/25.22
```

```
             |         smoke
         low | Non smoke     Smoker |     Total
-------------+----------------------+----------
Normal weight |     74.78      59.46 |     68.78
   Low weight |     25.22      40.54 |     31.22
-------------+----------------------+----------
       Total |    100.00     100.00 |    100.00

1.6074544
```

```
. cs low smoke
```

```
                 | smoke                   |
                 |   Exposed   Unexposed   |      Total
-----------------+-------------------------+------------
           Cases |        30          29   |         59
        Noncases |        44          86   |        130
-----------------+-------------------------+------------
           Total |        74         115   |        189
                 |                         |
            Risk |   .4054054    .2521739   |   .3121693
                 |                         |
                 |      Point estimate     |    [95% Conf. Interval]
                 |-------------------------+------------------------
 Risk difference |         .1532315        |    .0160718    .2903912
      Risk ratio |         1.607642        |    1.057812    2.443262
 Attr. frac. ex. |          .377971        |    .0546528    .5907112
 Attr. frac. pop |         .1921887        |
                 +-----------------------------------------------------
                              chi2(1) =     4.92  Pr>chi2 = 0.0265
```

10.3.2 Études diagnostiques

En ce qui concerne l'évaluation des tests diagnostiques, on partira dans la plupart des cas d'un tableau de contingence décrivant les effectifs associés au croisement de chaque modalité de deux variables binaires. Consi-

dérons les données issues d'une étude de validation d'un nouveau test diagnostique chez 1586 patients. Parmi les 744 patients malades, 670 ont été identifiés comme tels par ce nouveau test.

Les données sont reportées ci-dessous. Elles sont disponibles dans un fichier Stata appelé `diagnos.dta` et sont importable directement avec la commande `use`. On remarquera que pour éviter de « perdre » la session en cours, on utilise `preserve` pour sauvegarder les données en mémoire, puis on nettoie l'espace de travail en supprimant l'ensemble (`*`) des variables. Il ne serait toutefois pas très difficile de saisir soi-même les données à l'aide de l'éditeur intégré ou de la commande `input`.

```
. preserve
. drop *
. use "diagnos.dta"
. list
```

```
     +------------------+
     | Test   Dis     N |
     |------------------|
  1. |    1     1   670 |
  2. |    0     1    74 |
  3. |    1     0   202 |
  4. |    0     0   640 |
     +------------------+
```

On peut reconstituer le tableau d'effectif en pondérant la commande `tabulate` :

```
. tabulate Test Dis [fweight=N]
```

```
           |          Dis
      Test |         0          1 |     Total
-----------+----------------------+----------
         0 |       640         74 |       714
         1 |       202        670 |       872
-----------+----------------------+----------
     Total |       842        744 |     1,586
```

À partir de là, on dispose de toutes les informations nécessaires pour calculer les valeurs telles que la sensibilité ou la spécificité, de même que les valeurs prédictives positive et négative. On verra par la suite qu'il est également possible de vérifier les qualités diagnostiques d'un test à partir d'un modèle de régression logistique. Toutefois, il existe un package Stata qui calcule automatiquement toutes ces quantités (taper `findit diagt` et suivre les proécdures d'installation). Voici les résultats que l'on obtiendrait avec ces données.

```
. diagt Dis Test [fw=N], chi
```

```
           |         Test
       Dis |      Pos.       Neg. |     Total
-----------+----------------------+----------
  Abnormal |       670         74 |       744
    Normal |       202        640 |       842
-----------+----------------------+----------
     Total |       872        714 |     1,586

          Pearson chi2(1) = 696.4558   Pr = 0.000
True abnormal diagnosis defined as Dis = 1

                                                  [95% Confidence Interval]
-------------------------------------------------------------------------
Prevalence                          Pr(A)       47%       44%      49.4%
-------------------------------------------------------------------------
Sensitivity                        Pr(+|A)     90.1%     87.7%     92.1%
Specificity                        Pr(-|N)       76%       73%     78.9%
ROC area                (Sens. + Spec.)/2       .83      .812      .848
-------------------------------------------------------------------------
Likelihood ratio (+)     Pr(+|A)/Pr(+|N)      3.75      3.32      4.24
```

```
Likelihood ratio (-)     Pr(-|A)/Pr(-|N)      .131      .105      .163
Odds ratio                   LR(+)/LR(-)      28.7      21.5      38.2
Positive predictive value        Pr(A|+)     76.8%     73.9%     79.6%
Negative predictive value        Pr(N|-)     89.6%     87.2%     91.8%
------------------------------------------------------------------------------
```

On n'oubliera pas de restaurer l'environnement de départ à l'aide de la commande

```
. restore
```

10.4 Régression logistique

10.4.1 Estimation des paramètres du modèle

Lorsque les données sont disponibles en format individuel (tableau long où chaque ligne dénote une unité statistique pour laquelle on dispose d'une variable réponse binaire et de valeurs pour la ou les variables explicatives), on dispose de deux commandes pour réaliser une régression logistique (simple ou multiple), `logit` et `logistic`. Elles diffèrent essentiellement dans le format des résultats qu'elles affichent : `logistic` retourne par défaut des odds-ratio, alors que `logit` affiche les coefficients de régression sur l'échelle du log-odds. La commande `probit` permet d'estimer les paramètres d'un modèle de régression logistique en considérant une fonction de lien de type probit plutôt que logit.

Voici le résultat d'une régression logistique considérant l'indicateur de poids des bébés à la naissance (`low`) et le poids des mères (`lwt`).

```
. logistic low lwt

Logistic regression                               Number of obs   =        189
                                                  LR chi2(1)      =       5.98
                                                  Prob > chi2     =     0.0145
Log likelihood = -114.34533                       Pseudo R2       =     0.0255

------------------------------------------------------------------------------
         low | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         lwt |   .9695452   .0131597    -2.28   0.023     .9440927     .995684
       _cons |   2.713702   2.131045     1.27   0.204     .5822659    12.64745
------------------------------------------------------------------------------
```

Comme dans le cas de la régression linéaire, Stata fournit les valeurs des paramètres du modèle (ici, en termes d'odds et d'odds-ratio associé à la variation d'une unité de la variable explicative numérique) et leurs intervalles de confiance, accompagnés des tests de significativité usuels (pour les coefficients et pour le modèle, `LR chi2(1)`). La valeur de pseudo R^2 reportée par Stata correspond au coefficient de McFadden. On obtiendra des indices supplémentaires de qualité d'ajustement du modèle en utilisant la commande `fitstat` :

```
. fitstat

Measures of Fit for logistic of low

Log-Lik Intercept Only:       -117.336   Log-Lik Full Model:          -114.345
D(187):                        228.691   LR(1):                          5.981
                                         Prob > LR:                      0.014
McFadden's R2:                   0.025   McFadden's Adj R2:              0.008
ML (Cox-Snell) R2:               0.031   Cragg-Uhler(Nagelkerke) R2:     0.044
McKelvey & Zavoina's R2:         0.053   Efron's R2:                     0.032
Variance of y*:                  3.475   Variance of error:              3.290
Count R2:                        0.688   Adj Count R2:                   0.000
AIC:                             1.231   AIC*n:                        232.691
BIC:                          -751.516   BIC':                          -0.740
BIC used by Stata:             239.174   AIC used by Stata:            232.691
```

Dans le cas où la variable explicative est binaire, le principe est le même. Voici une illustration avec le poids des bébés et l'existence de douleurs intra-utérines chez la mère, en utilisant cette fois la commande `logit` :

```
. logit low ui, nolog

Logistic regression                               Number of obs   =        189
                                                  LR chi2(1)      =       5.08
                                                  Prob > chi2     =     0.0243
Log likelihood = -114.79795                       Pseudo R2       =     0.0216

------------------------------------------------------------------------------
         low |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          ui |   .9469277   .4167734     2.27   0.023     .1300669    1.763789
       _cons |  -.9469277   .1756215    -5.39   0.000     -1.29114   -.6027159
------------------------------------------------------------------------------
```

Notons que l'on peut ajouter l'option `or` pour obtenir l'odds-ratio, que l'on peut d'ailleurs retrouver en prenant l'exponentielle du coefficient de régression stocké dans `_b[ui]`.

```
. display exp(_b[ui])

2.5777778
```

Dans le cas d'une étude diagnostique, ou plus généralement lorsque l'on est plutôt intéressé par une approche de classification que de régression, la commande suivante fournit un tableau de contingence résumant les individus correctement ou incorrectement classifiés comme positifs et négatifs par rapport à la variable réponse, en considérant un seuil de 0.5 pour la probabilité de détection ou d'allocation des événements.

```
. estat classification

Logistic model for low

              -------- True --------
Classified |         D             ~D  |      Total
-----------+--------------------------+-----------
     +     |        14             14  |         28
     -     |        45            116  |        161
-----------+--------------------------+-----------
   Total   |        59            130  |        189

Classified + if predicted Pr(D) >= .5
True D defined as low != 0
--------------------------------------------------
Sensitivity                     Pr( +| D)   23.73%
Specificity                     Pr( -|~D)   89.23%
Positive predictive value       Pr( D| +)   50.00%
Negative predictive value       Pr(~D| -)   72.05%
--------------------------------------------------
False + rate for true ~D        Pr( +|~D)   10.77%
False - rate for true D         Pr( -| D)   76.27%
False + rate for classified +   Pr(~D| +)   50.00%
False - rate for classified -   Pr( D| -)   27.95%
--------------------------------------------------
Correctly classified                        68.78%
--------------------------------------------------
```

10.4.2 Prédiction ponctuelle et par intervalle

La commande `predict` s'utilise pour calculer les valeurs ajustées pour un modèle donné ou pour estimer les valeurs de probabilité ou de log-odds pour de nouvelles observations : il s'agit d'une commande de post-

estimation, et elle s'utilisera donc après avoir construit un modèle de régression avec `logit` ou `logistic`. Les options permettent de définir le type de prédiction que l'on souhaite réaliser : si l'on s'intéresse à prédire des probabilités, on utilisera l'option `p` ; le cas échéant (option `xb`, par défaut), Stata fournit des prédictions individuelles sur l'échelle de lien.

```
. logit low lwt
. predict pr, p
```

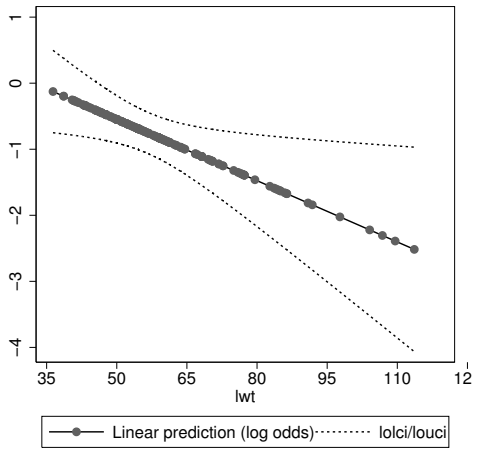
```
Iteration 0:   log likelihood =   -117.336
Iteration 1:   log likelihood = -114.37209
Iteration 2:   log likelihood = -114.34534
Iteration 3:   log likelihood = -114.34533

Logistic regression                                     Number of obs   =        189
                                                        LR chi2(1)      =       5.98
                                                        Prob > chi2     =     0.0145
Log likelihood = -114.34533                             Pseudo R2       =     0.0255

------------------------------------------------------------------------------
         low |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         lwt |  -.0309282   .0135731    -2.28   0.023    -.0575309   -.0043254
       _cons |   .9983136   .7852908     1.27   0.204    -.5408281    2.537455
------------------------------------------------------------------------------
```

Le calcul des intervalles de confiance à 95 % pour les valeurs ajustées ne pose pas de réelle difficulté. Manuellement, on peut procéder comme suit (sur l'échelle du log-odds) : on génère dans un premier temps les prédictions linéaires (`lo`), l'erreur de prédiction (`lose`) et les bornes des intervalles de confiance associés (`lolci` et `louci`). L'affichage consiste ensuite en un diagramme de dispersion des valeurs prédites (log odds) en fonction du poids de la mère, sur lequel on superpose des droites délimitant les intervalles de confiance (on remarquera qu'il est nécessaire d'ordonner les coordonnées des points dans ce dernier cas).

```
. predict lo, xb
. predict lose, stdp
. gen lolci = lo - 1.96*lose
. gen louci = lo + 1.96*lose
. twoway (scatter lo lwt, sort connect(l)) (line lolci louci lwt, sort pstyle(p3 p3)), ///
.   xlabel(35(15)120)
```



Considérons un modèle incluant en plus de la variable `lwt` l'ethnicité des mères (`race`). La commande `margins` fournit des outils très puissants pour calculer des prédictions avec intervalles de confiance ou des effets marginaux. Voici un exemple d'utilisation relativement simplifié pour un modèle incluant les variables `lwt` et `race` :

```
. logit low lwt i.race
. quietly: margins, at(lwt=(40(10)110)) over(race)
```

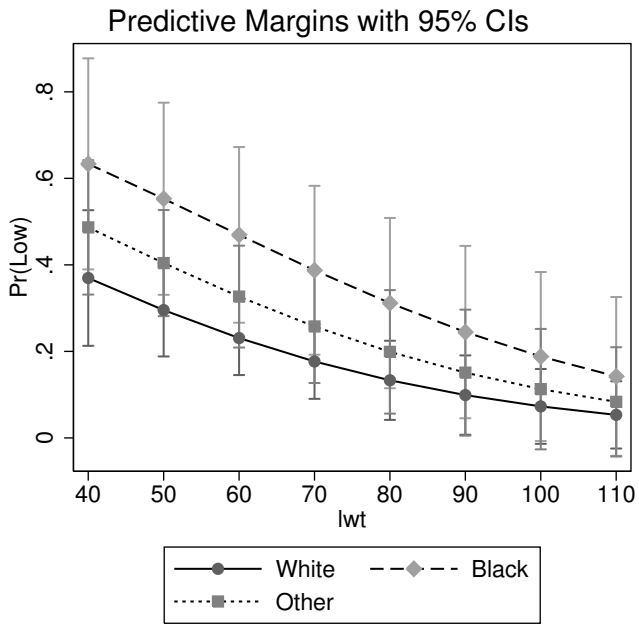
```
. marginsplot

Iteration 0:   log likelihood =   -117.336
Iteration 1:   log likelihood = -111.73378
Iteration 2:   log likelihood = -111.62959
Iteration 3:   log likelihood = -111.62954
Iteration 4:   log likelihood = -111.62954

Logistic regression                                     Number of obs   =        189
                                                        LR chi2(3)      =      11.41
                                                        Prob > chi2     =     0.0097
Log likelihood = -111.62954                             Pseudo R2       =     0.0486

------------------------------------------------------------------------------
         low |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         lwt |  -.0334908   .0141666    -2.36   0.018    -.0612568   -.0057248
             |
        race |
           2 |   1.081066   .4880522     2.22   0.027     .1245015    2.037631
           3 |   .4806032   .3566737     1.35   0.178    -.2184644    1.179671
             |
       _cons |   .8057537   .8451667     0.95   0.340    -.8507426     2.46225
------------------------------------------------------------------------------

  Variables that uniquely identify margins: lwt race
```



10.4.3 Cas des données groupées

Dans le cas où les données sont groupées, c'est-à-dire lorsqu'on ne dispose que d'un tableau résumant les effectifs observés pour chaque croisement des modalités de deux variables catégorielles, on utilisera la commande `blogit` en lieu et place de `logit`. Il est alors indispensable de préciser, dans l'ordre, les événements positifs, le nombre total d'événements et la ou les variables explicatives associées. Voici un exemple d'application avec les mêmes données (`low` et `ui`). Cette fois-ci, plutôt que d'utiliser un fichier externe ou de construire le tableau de données avec `input`, on va exploiter directement les données individuelles disponibles dans l'espace de travail. L'ensemble de ces manipulations sera encadré par une paire d'instructions `preserve/restore` pour

pouvoir revenir aux données initiales à la fin de l'exemple.

Dans un premier temps, on construit un tableau résumant la fréquence de co-occurence de chaque paire de modalité des variables `low` et `ui` à l'aide de la commande `contract`. L'option `freq(n)` permet simplement de renommer la variable servant de statistique de comptage. On a ensuite besoin de l'effectif total associé à chaque modalité de la variable `ui`, ce qui nous permettra d'obtenir la somme des événements positifs pour la variable `low` (contenu dans `n`) et la somme des événements positifs et négatifs associés (appelons la `tot`) lorsque `low=1` (`"Low weight"`) : cela peut se réaliser avec une instruction `egen`, groupée sur les modalités de `ui`.

```
. preserve
. contract low ui, freq(n)
. egen tot = sum(n), by(ui)
. list
```

```
     +--------------------------------+
     |           low    ui     n   tot |
     |--------------------------------|
  1. | Normal weight    No   116   161 |
  2. | Normal weight   Yes    14    28 |
  3. |    Low weight    No    45   161 |
  4. |    Low weight   Yes    14    28 |
     +--------------------------------+
```

On peut vérifier que les effectifs totaux sont bien identiques pour chacune des modalités de `ui`, et les données qui nous intéressent sont les deux dernières lignes de ce tableau. Cela permet en effet d'obtenir, pour chaque niveau de la variable `ui` le nombre d'événements positifs et le nombre d'événements total. Le modèle de régression est alors formulé comme ci-dessous.

```
. blogit n tot ui if low == 1, or
. restore
```

```
Logistic regression for grouped data              Number of obs   =        189
                                                  LR chi2(1)      =       5.08
                                                  Prob > chi2     =     0.0243
Log likelihood = -114.79795                       Pseudo R2       =     0.0216

------------------------------------------------------------------------------
    _outcome | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          ui |   2.577778   1.074349     2.27   0.023     1.138905      5.8345
       _cons |    .387931    .068129    -5.39   0.000     .2749573    .5473232
------------------------------------------------------------------------------
```

Ce qu'il faut retenir

- Les commandes `correlate` et `spearman` permettent de calculer les coefficients de corrélation de Bravais-Pearson et Spearman. Un diagramme de dispersion peut être construit à l'aide de `twoway scatter` afin de visualiser la forme du nuage de points.
- La commande `regress` est utilisée dans le cadre de la régression linéaire, alors que pour la régression logistique on utilisera `logit` (ou `logistic`). Dans les deux cas, il existe un même ensemble de commandes de post-estimation permettant d'obtenir des informations sur la qualité d'ajustement du modèle (`fitstat`), ou de calculer les valeurs ajustées ou de prédire de nouvelles observations (`predict`).
- La plupart des mesures de risque rencontrées en épidémiologie sont accessibles à partir des commandes `epitab` (`cc`, `cs`), ou de certaines commandes spécifiques (`tabodds`, `mhodds`).

Cours 11. Analyse de données de survie

Sommaire

Ce dernier chapitre constitue une introduction à la modélisation des données de survie avec Stata. Après avoir décrit le principe d'organisation et de représentation des données de survie sous Stata, on s'intéressera à l'estimation de la fonction de survie par la méthode de Kaplan-Meier et au modèle de régression de Cox sont ensuite discutés plus en détails.

Remarque : Dans ce chapitre, les illustrations sont réalisées à partir d'un jeu de données différent de celui utilisé jusqu'à présent. Les données sont disponibles dans le package R `survival` et ont été exportées au format CSV. Les données manquantes ont été codées sous la forme d'un point (au lieu de la valeur `NA`, utilisée par défaut sous R). Elles peuvent être chargées sous Stata de la manière suivante :

```
. insheet using "lung.csv", clear
. label define gender 1 "Male" 2 "Female"
. label values sex gender
```

```
(10 vars, 228 obs)
```

Il s'agit de données sur la survie de patients atteints d'un cancer du poumon. Les variables d'intérêt sont les suivantes : `time` temps de survie en jours, `status` état à la date de point (1=donnée censurée, 2=patient décédé), `age` âge du patient et `sex` sexe du patient (1=homme, 2=femme). Au total, il y a 228 patients.

11.1 Représentation des données et statistiques descriptives

11.1.1 Format de représentation des données de survie

Du fait de la nature des données de survie, qui sont représentées à l'aide d'une variable codant une durée et d'une autre variable codant un événement d'intérêt (décès, rechute, échec, etc.), la représentation et la modélisation de telles données font appel à une classe particulière de commandes Stata. Il s'agit de la commande `sts` qui possède plusieurs sous-commandes (`list`, `test`, `graph`, essentiellement) et de quelques commandes dont le préfixe est `st`.

La commande `stset` se charge de créer plusieurs variables auxiliaires (dont le nom débute généralement par `_`) que l'on ne manipule pas directement mais que Stata utilise de manière transparente. La commande `stset` exige que l'on indique la variable définissant les durées (indépendemment de l'unité de mesure) et via une option `failure()` quelle variable code pour les événements. Par défaut, Stata considère que toutes les valeurs non nulles et non manquantes signalent l'événement d'intérêt (par exemple, le décès). Avec des valeurs en 0/1, cela ne pose aucun problème, en supposant qu'une valeur de 1 indique l'événement.

```
failure event:  status != 0 & status < .
```

Il peut arriver, comme c'est le cas ici que le décès ou l'événement soit représenté par une autre valeur et que la censure ne soit pas codée 0. Dans le cas, il est nécessaire de préciser la valeur du décès dans l'option `failure()` comme indiqué ci-dessous :

```
. stset time, failure(status=2)
```

```
     failure event:  status == 2
obs. time interval:  (0, time]
 exit on or before:  failure

--------------------------------------------------------------------------
      228  total obs.
        0  exclusions
--------------------------------------------------------------------------
      228  obs. remaining, representing
      165  failures in single record/single failure data
    69593  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =      1022
```

11.1.2 Statistiques descriptives

Voici un aperçu des données brutes :

```
. list time status age sex in 1/5
```

```
     +----------------------------+
     | time   status   age    sex |
     |----------------------------|
  1. |  306        2    74   Male |
  2. |  455        2    68   Male |
  3. | 1010        1    56   Male |
  4. |  210        2    57   Male |
  5. |  883        2    60   Male |
     +----------------------------+
```

On peut naturellement utiliser l'ensemble des commandes Stata présentées dans les chapitres précédents pour procéder au résumé numérique des variables. On prendra toutefois garde au fait que dans ce cas les données sont traitées indépendemment de l'existence de censures.

```
. tabulate status, summarize(time)
```

```
            |           Summary of time
     status |        Mean   Std. Dev.       Freq.
------------+------------------------------------
          1 |   363.46032   221.13635          63
          2 |         283   202.80508         165
------------+------------------------------------
      Total |   305.23246   210.64554         228
```

11.2 Fonction de survie et courbe de Kaplan-Meier

11.2.1 Table de mortalité

Pour construire la table de mortalité et afficher les probabilités de survie au cours du temps, on utilisera la commande `sts list`. Sans autre option, Stata affiche tous les événements temporels.

```
. sts list
```

Il est toutefois possible de limiter l'affichage à certaines valeurs précises. Pour afficher, par exemple, la probabilité de survie associée aux temps 200 et 300, on écrirait :

```
. sts list, at(200 300) enter
```

```
         failure _d:  status == 2
   analysis time _t:  time
```

```
            Beg.                    Survivor      Std.
    Time     Total     Fail         Function     Error     [95% Conf. Int.]
-------------------------------------------------------------------------------
     200       145       72           0.6803    0.0311     0.6149    0.7369
     300        92       29           0.5306    0.0346     0.4605    0.5958
-------------------------------------------------------------------------------
Note:  survivor function is calculated over full data and evaluated at
       indicated times; it is not calculated from aggregates shown at left.
```

La médiane de survie et son intervalle de confiance à 95 % peut être obtenue grâce à la commande `stci`.

```
. stci, dd(2) noshow
```

```
             |    no. of
             |  subjects         50%     Std. Err.     [95% Conf. Interval]
-------------+-------------------------------------------------------------
       total |       228      310.00         21.77          284         361
```

L'option `noshow` permet quant à elle de ne pas afficher de rappel concernant les variables utilisées (ici, `time` et `status`). Quant à l'option `dd(2)`, elle permet de limiter l'affichage à deux décimales. Si l'on souhaite obtenir le fractile d'ordre 10 plutôt que la médiane, on indiquera `p(10)` en option.

```
. stci, p(10) dd(2)
```

```
        failure _d:  status == 2
  analysis time _t:  time

             |    no. of
             |  subjects         10%     Std. Err.     [95% Conf. Interval]
-------------+-------------------------------------------------------------
       total |       228       79.00         14.94           54         105
```

Cette commande peut être utilisée lorsqu'il y a plusieurs groupes et que l'on souhaite comparer leur médiane de survie. La variable de classification sera dans ce cas indiquée à l'aide d'un option `by()`.

```
. stci, by(sex) noshow
```

```
             |    no. of
sex          |  subjects         50%     Std. Err.     [95% Conf. Interval]
-------------+-------------------------------------------------------------
        Male |       138         270      26.78831          210         306
      Female |        90         426      44.20601          345         524
-------------+-------------------------------------------------------------
       total |       228         310      21.77251          284         361
```

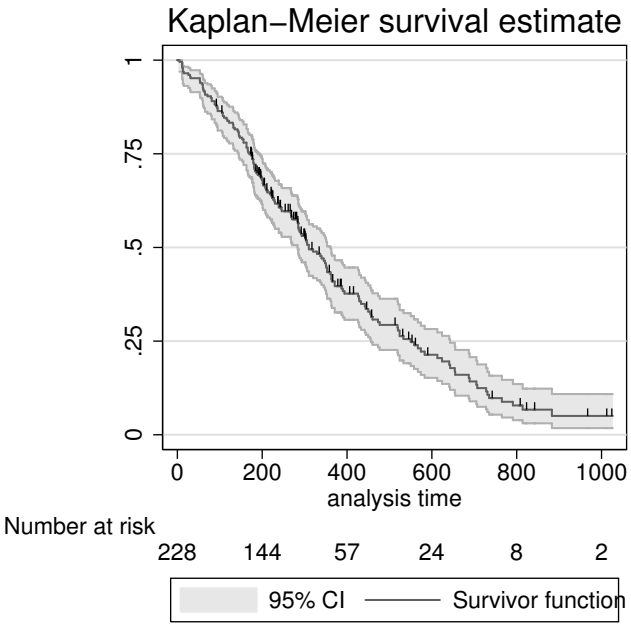
11.2.2 Courbe de Kaplan-Meier

La commande `sts graph` permet de représenter graphique la courbe de survie d'un ou plusieurs échantillons. Dans le cas de plusieurs échantillons, on indique le critère de classification à l'aide d'une obtention `by()`. La syntaxe de base est donc :

```
. sts graph
```

mais on on peut également taper simplement `sts`. On rajoutera l'option `ci` pour afficher des intervalles de confiance (pour chaque valeur de temps). Voici un exemple d'usage utilisant d'autres options telles que l'affichage simultané du nombre d'individus à risque au cours du temps (par défaut, Stata utilise les mêmes coordonnées temporelles que celles affichées pour l'axe des abscisses mais cela peut être modifié). Une autre option intéressante est `censored()` (que l'on peut abbréger en `cen()`) qui superpose sur la courbe de survie les censures observées.

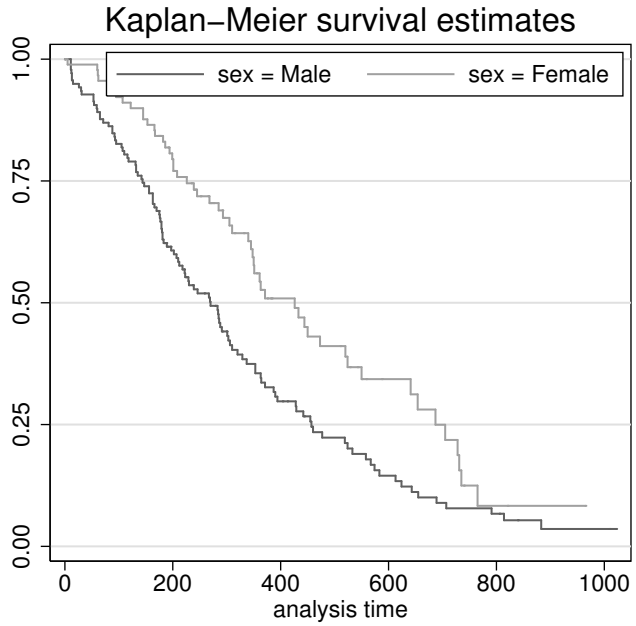
```
. sts graph, noshow ci risktable censored(single)
```



Dans le cas de deux échantillons, on incluera le facteur via l'option `by()`. Le reste des options est applicable. Dans l'illustration ci-dessous, on a modifié le positionnement de la légende afin que celle-ci apparaisse à l'intérieur du graphique et non dans la marge basse.

```
. sts graph, by(sex) legend(ring(0) position(2))

        failure _d:  status == 2
  analysis time _t:  time
```



11.2.3 Fonction de risque cumulé

Si l'on souhaite travailler avec la fonction de risque cumulé (notée le plus souvent $H(t)$), il suffit d'ajouter l'option `cumhaz` lorsque l'on utilise la commande `sts graph`.

```
. sts graph, noshow cumhaz ci
```

11.2.4 Test d'égalité de fonctions de survie

La commande `sts list` fournit le tableau de mortalité et les valeurs estimées de la survie pour chaque durée. L'option `by()` permet de calculer la fonction de survie pour 2 ou plusieurs groupes d'individus. Mais il est également possible de coupler cette option `by()` à `compare` pour afficher directement la survie estimée dans chacun des groupes côte à côte.

```
. sts list, by(sex) compare noshow
```

```
                Survivor Function
sex               Male     Female
--------------------------------
time       5     1.0000     0.9889
         132     0.7681     0.8993
         259     0.5192     0.7186
         386     0.3265     0.5089
         513     0.2232     0.4110
         640     0.1228     0.3433
         767     0.0781     0.0832
         894     0.0357     0.0832
        1021     0.0357          .
        1148          .          .
--------------------------------
```

Pour réaliser un test du log-rank (égalité des fonctions de survie), on utilisera la commande `sts test` en indiquant simplement la variable définissant les groupes à comparer.

```
. sts test sex, noshow
```

```
Log-rank test for equality of survivor functions
------------------------------------------------

       |   Events         Events
sex    |  observed       expected
-------+-------------------------
Male   |       112          91.58
Female |        53          73.42
-------+-------------------------
Total  |       165         165.00

             chi2(1) =      10.33
             Pr>chi2 =     0.0013
```

Si l'on souhaite réaliser un test de Wilcoxon à la place, on ajoutera l'option `wilcoxon` comme indiqué ci-après.

```
. sts test sex, wilcoxon noshow
```

```
Wilcoxon (Breslow) test for equality of survivor functions
----------------------------------------------------------

       |   Events         Events        Sum of
sex    |  observed       expected        ranks
-------+--------------------------------------
Male   |       112          91.58         3148
Female |        53          73.42        -3148
-------+--------------------------------------
Total  |       165         165.00            0

             chi2(1) =      12.47
             Pr>chi2 =     0.0004
```

11.3 Régression de Cox

La commande permettant de réaliser une régression de Cox est `stcox`. Son usage est sensiblement identique aux commandes de régression vues dans les chapîtres précédents, à ceci près qu'il n'est pas nécessaire d'indiquer de variable réponse : comme pour les autres commandes `sts`, Stata gère lui-même la représentation temps/événement. On se contentera donc d'indiquer la liste des variables explicatives après le nom de la commande. Voici un exemple d'utilisation en ne considérant que la variable `sex` :

```
. stcox sex, noshow

Iteration 0:   log likelihood = -750.12202
Iteration 1:   log likelihood = -744.83027
Iteration 2:   log likelihood = -744.81818
Iteration 3:   log likelihood = -744.81818
Refining estimates:
Iteration 0:   log likelihood = -744.81818

Cox regression -- Breslow method for ties

No. of subjects =          228                     Number of obs   =       228
No. of failures =          165
Time at risk    =        69593
                                                   LR chi2(1)      =     10.61
Log likelihood  =   -744.81818                     Prob > chi2     =    0.0011

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         sex |   .5883716   .0983645    -3.17   0.002     .4239817    .8165002
------------------------------------------------------------------------------
```

Pour indiquer la présence de variables de stratification, on utilisera l'option `strata()` de la manière suivante :

```
. stcox age, strata(sex) noshow

Iteration 0:   log likelihood = -643.61669
Iteration 1:   log likelihood = -642.03076
Iteration 2:   log likelihood = -642.02946
Refining estimates:
Iteration 0:   log likelihood = -642.02946

Stratified Cox regr. -- Breslow method for ties

No. of subjects =          228                     Number of obs   =       228
No. of failures =          165
Time at risk    =        69593
                                                   LR chi2(1)      =      3.17
Log likelihood  =   -642.02946                     Prob > chi2     =    0.0748

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   1.016324   .0093351     1.76   0.078      .998191    1.034786
------------------------------------------------------------------------------
                                                        Stratified by sex
```

Il est possible de modifier la manière dont Stata traite les ex-æquos, en indiquant l'option `efron` par exemple. Si l'on souhaite afficher les coefficients de régression plutôt que le hazard ratio, on spécifiera l'option `nohr` :

```
. stcox sex, noshow nolog nohr
```

```
Cox regression -- Breslow method for ties

No. of subjects =          228                     Number of obs   =       228
No. of failures =          165
Time at risk    =        69593
                                                   LR chi2(1)      =     10.61
Log likelihood  =   -744.81818                     Prob > chi2     =    0.0011

------------------------------------------------------------------------------
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         sex |  -.5303966   .1671808    -3.17   0.002     -.858065   -.2027282
------------------------------------------------------------------------------
```

Les options `noshow` et `nolog` permettent de supprimer l'affichage des variables de survie et des itérations pour la convergence du modèle.

Index