Etude statistique1D.

Sans donner un cours de statistique, nous allons programmer les méthodes de base qui permettent l'étude statistique d'un problème.

Quelques rappels semblent nécessaires.

On définit :
Une **population** : il s'agit d'un ensemble d'individus (ou objets) ayant un point commun.

Un **échantillon** : c'est un sous ensemble de la population étudiée.

Exemple : la population étant l'ensemble des habitants de Liège. Un échantillon étant les étudiants masculins de l'inpres.

Une **variable** est la caractéristique sur laquelle s'effectue l'étude.

Cette variable peut être **qualitative** si elle exprime une qualité, c'est-à-dire qu'elle ne se mesure pas.

Elle peut être **quantitative** si elle peut être mesurée.

Exemple : une personne est mariée ou non.

Elle est âgée de moins de 20 ans.

Une variable quantitative peut être **discrète** lorsqu'elle ne peut prendre qu'un nombre fini de valeurs, elle est **continue** lorsqu'elle peut prendre un nombre infini de valeurs dans un intervalle.

Exemple :

Le nombre d'enfants par famille. Le poids d'une personne.

Soit l'étude suivante : Recensement du nombre d'enfants par ménage dans un échantillon de 133 familles.

Enfants/ménage x_i	Répétition n_i
0	2
1	8
2	10
3	52
4	25
5	14
6	17
7	2
8	0
9	2
10	1
Total	n = 133

$n = \sum x_i \, . \, n_i$

On définit :

L'étendue :

L'étendue est la différence entre la plus grande et la plus petite valeur de l'échantillon.

$$\text{Etendue} = E = x_{max} - x_{min}$$

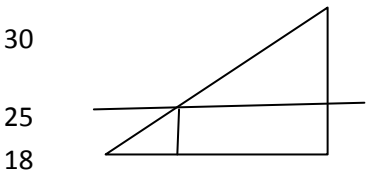
La médiane :

La médiane est la valeur centrale par excellence car elle divise la distribution en 2 parties égales.
Lorsque l'effectif total de la série est impaire, (EffTotal = 2*n + 1), c'est simple, la médiane est la $(n+1)^{ème}$ valeur.
Si par contre, l'effectif total est paire, il s'agit alors de ($n^{ème}$ valeur + $(n+1)^{ème}$ valeur)/2
Dans le cas d'une série discrète, pas de problème.
Dans le cas d'une série continue, il faut faire une interpolation linéaire.
Par exemple : soit la série suivante :

[0 ; 5[	10
[5 ; 8 [	8
[8 ;12 [	12
[12 ;15[	11
[15; 20[	9
	50

la médiane est donc la moyenne entre la 25 et la $26^{ème}$ valeurs.

La $25^{ème}$ valeur est dans l'intervalle [8 ;12[(celles-ci sont supposées répartie uniformément dans l'intervalle) et est la $7^{ème}$ valeur de l'intervalle qui en contient 12.



Etendue = 4
$n_i = 12$

$\text{Mediane}_1 = 8 + \frac{4}{12} * 7$ = 10.33

Début de l'intervalle

Position dans l'intervalle.

Idem pour la $26^{ème}$ valeur et faire la moyenne.

Le mode :
Le mode est la valeur la plus fréquente dans un échantillon.

Remarque : Il peut y avoir plusieurs modes.

Exemple : dans notre exemple, le mode est 3.

La moyenne arithmétique :

La **moyenne arithmétique** d'un échantillon d'effectif n est notée $\bar{x}$ et est définie par l'expression

$$Moy = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Cette valeur situe le centre de la distribution statistique.

Remarque : elle est sensible aux valeurs aberrantes (car elle fait intervenir toutes les valeurs de l'échantillon).

L'écart-type :

L'écart-type d'un échantillon de données d'effectif n est noté s et est défini par l'expression

$$EcartType = s = \sqrt{\frac{\sum n_i.(x_i - \bar{x})^2}{n}}$$

Où pour des raisons de précision, il et préférable d'utiliser la formule suivante

$$EcartType = s = \sqrt{\frac{\sum n_i.{x_i}^2 - (\sum x_i)^2/n}{n}}$$

L'écart-type est un indicateur de dispersion.

Pour 2 échantillons ayant la même moyenne, si **s1** > **s2**, le premier échantillon est plus dispersé que le second.

Exemple :

La valeur de l'écart-type pour nos famille est s = 1.68

Dans le cas d'une distribution suivant une symétrie de Gauss, on peut dire que 95% de l'effectif est situé dans l'intervalle [x – 2s, x + 2s] .

Coefficient de variation :

Le calcul de l'écart-type et de la moyenne est très utile pour déterminer la précision.

Le coefficient de variation, exprimé en % est donné par la formule

$$CV = \frac{s}{\bar{x}}.100\%$$

Plus ce coefficient est faible, plus les mesures sont précises.

Remarque :

Une autre application du calcul de l'écart-type et de la moyenne concerne le domaine de contrôle de qualité.

Si la valeur de xi est comprise dans l'intervalle [x – 2s, x + 2s], on dit que le processus est sous contrôle. (c'est-à-dire. Le cas de 95% des valeurs observées)

Si xi est hors des limites xi $\pm$ 2s mais endéans des limites xi $\pm$ 3s, on suspecte un problème.

Si xi est hors des limites xi $\pm$ 3s, le processus est hors contrôle.

On demande de créer une classe **CSerieStatistiques1D** dont le rôle est de calculer les paramètres statistiques d'une série statistique à une dimension :
A savoir :
– tendance centrale : moyenne (getMoyenne()), mode (getMode(…) ; attention au cas multimodal), médiane (getMediane()).

– dispersion : écart-type (getEcartType()), étendue (getEtendue()), coefficient de variation (getCV()).

– une méthode AfficheRapport().

Pas de commentaire sur les premières méthodes.

La méthode AfficheRapport() [1]

```
Nom :
Sujet de l'étude :
type :



Effectif total :
Nom :
Sujet de l'étude :
type :


Donnees :
---------
        …

Effectif total :

   Moyenne :
   Médiane :
   Mode :          0 : 0 : 0
   Ecart type :
   Coefficient de variation : %

Contrôle de qualité:
```

[1] Voir la description des fichiers de données plus bas.

```
valeur min     :
valeur Maximum :

Valeurs de la statistique sous contrôle.
```

Etude statistique2D.

Un problème intéressant est d'étudier les observations simultanées faites sur n individus.
Par exemple, y a-t-il une relation entre le poids et la taille des individus ?
Dans ce cas, lorsque les 2 variables X et Y sont calculées simultanément, on dit que l'on a affaire à un problème de corrélation.
Ces 2 variables sont quantitatives et le plus souvent continues.
Exemple :
Soit l'étude suivante : Age d'un médecin (années) et expérience professionnelle (années)

Age du médecin x_i	Expérience y_i
40	5
46	20
38	13
34	9
33	8
47	22
44	20
55	27
41	16
31	6
45	20
42	17
60	34
42	15
32	8

On obtient donc, pour chaque mesure, un couple (X,Y).

Calcul de la corrélation :
Ce calcul est important, car il permet de connaitre la dépendance qui existe entre les 2 variables. Plus le coefficient de corrélation est proche des valeurs extrêmes -1 et 1, plus la corrélation entre les variables est forte. Une corrélation égale à 0 signifie que les variables sont indépendantes.
Le coefficient de corrélation n'est pas sensible aux unités de chacune des variables. Ainsi, par exemple, le coefficient de corrélation linéaire entre l'âge et le poids d'un individu sera identique que l'âge soit mesuré en semaines, en mois ou en années. En revanche, ce coefficient de corrélation est extrêmement sensible à la présence de valeurs aberrantes.[2]

[2] Ceci sera étudier plus tard.

La formule est donnée par :

$$Corr = r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{[\sum(x_i - \bar{x})^2][\sum(y_i - \bar{y})^2]}}$$

Mais, toujours pour des raisons d'erreurs d'arrondis, on préfère utiliser la formule suivante ;

$$Corr = r = \frac{\sum x_i y_i - (\sum x_i)(\sum y_i)/n}{\sqrt{[\sum x_i^2 - (\sum x_i)^2/n][\sum y_i^2 - (\sum y_i)^2/n]}}$$

Pour l'exemple donné, r = 0.946

Il est aussi possible de prévoir la relation entre ces 2 variables. Ainsi, connaissant une variable, on peut estimer la valeur de l'autre.
Pour cela, il faut calculer la droite de régression.
Une droite a pour équation

$$y = a\,x + b$$

Ou

$$A = a = \frac{\sum x_i y_i - (\sum x_i)(\sum y_i)/n}{\sum x_i^2 - (\sum x_i)^2/n}$$

$$B = b = \bar{y} - a\,\bar{x}$$

On demande de créer une classe **CSerieStatistiques2D** dont le rôle est de calculer les paramètres statistiques d'une série statistique à 2 dimensions :
A savoir :
- A,B,Corr

Et dans le cas d'une étude 2D , AfficheRapport() sera :

```
Etude statistique:
------------------

Titre :           Temps de réaction complète (min) en fonction de la
température (°)
Sujet de l'etude  Temperature (°)  --  Temps (min)
Effectif Total :  10
Type :            C      C

Valeurs:
        25 - 0.64
        45 - 1.27
        55 - 0.95
        85 - 1.85
        115 - 2.81
        125 - 2.8
```

```
        150 - 3.42
        165 - 4.3
        175 - 4.54
        200 - 4.7

   Moyenne Val1 : 114
   Moyenne Val2 : 2.728


Corrélation :

   Coefficient a : 0.0249807
   Coefficient b : -0.1198
        1 : Prévision pour : Température (°)
        2 : Prévision pour : Temps (min)
        3 : Sortie         :
1
        Entrer la valeur pour  Température (°) : 70
la valeur prévue  : 1.62885

        1 : Prévision pour : Température (°)
        2 : Prévision pour : Temps (min)
        3 : Sortie         :
3
fin
```

Réalisation :

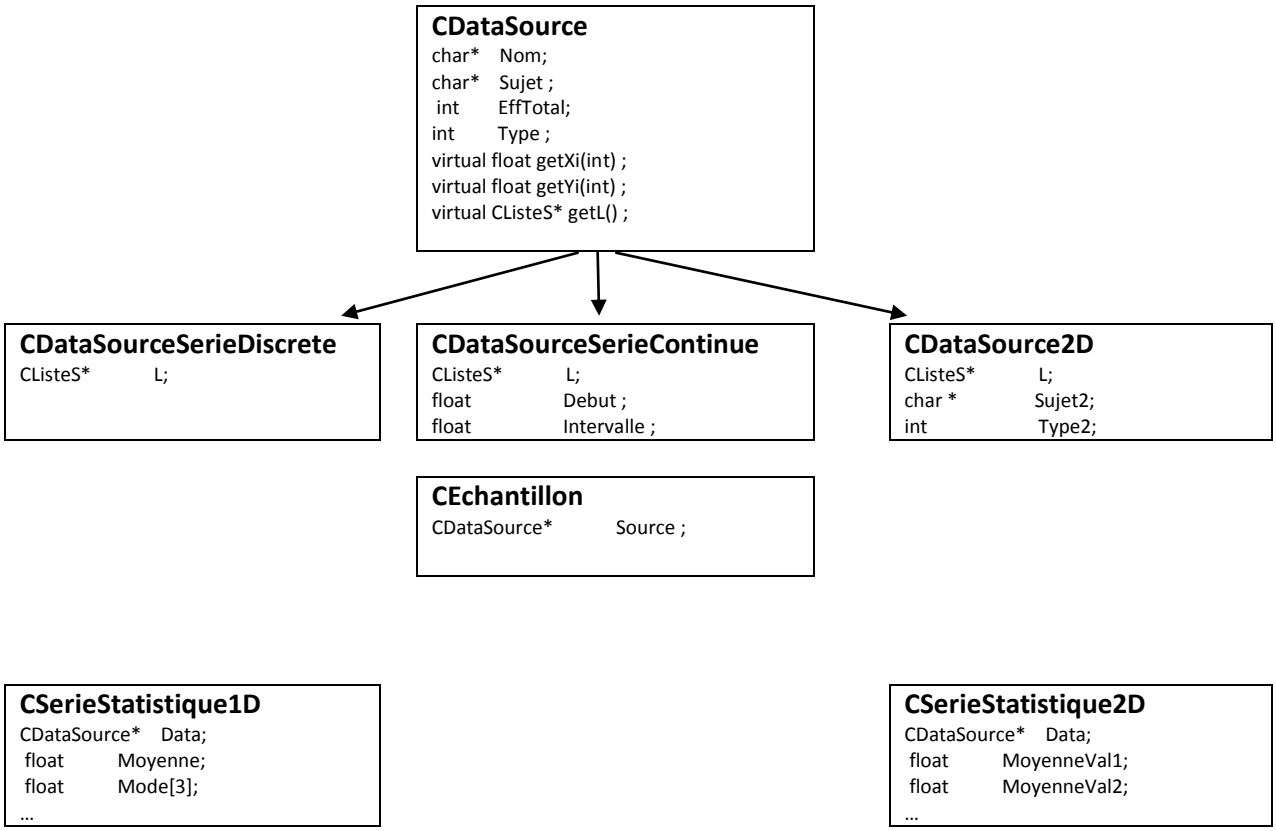
Il vous sera livré plusieurs fichiers de données pour tester votre programme.
Ils seront tous sous le format suivant :
1ère ligne : Nom de la statistique
2ème ligne : sujet de l'étude
3èmè ligne : Type de données (Continue /Discrète)
Lignes suivantes : Données

Si le fichier possède plusieurs sujets d'études, les différents champs seront séparés par le caractère ' : ' .

Exemple :

```
Résultats examens des Info (2 dernières années) Examens présentés.
ORG.ENT(/20):POO(/20):SYST.EXPL(/20):RESEAUX(/20):RES.TECH.INTERNET(/20):AN
GLAIS(/20):APOO(/20)
C:C:C:C:C:C:C
1.5:13.5:13.5:11.4:12.2:11.4:12
8.5:12.5:12.5:12.5:10.6:11:15.4
6:12.5:12.5:12.5:10.6:11:15.4
…
```

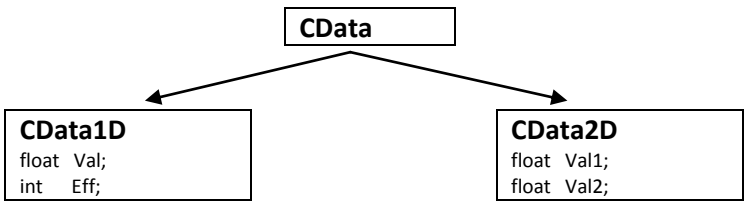
Pour cela, il faut les différentes classes suivantes :



En fait, on doit disposer d'une liste de données de 2 valeurs ((Valeur[3], Effectif[4]) pour une statistique 1D ou (Valeur, Valeur) pour une 2D). Il s'agit de la CListeS.

Quelque soit l'étude faite, on doit disposer de cette CListeS, et les éléments de cette CListeS seront interprétés comme (Valeur,Effectif) s'il s'agit d'une statistique 1D ou de (Valeur, Valeur) s'il s'agit d'une 2D.

Pour cela, il faut définir les classes suivantes :



Ainsi, la CListeS aura pour éléments de CData, qui seront interprété comme étant des CData1D ou CData2D. Ce qui est parfaitement possible grâce au **down-casting**.

[3] float
[4] int

Maintenant, on peut créer notre échantillon, c'est-à-dire nos données pour effectuer ensuite notre étude statistique.

Notre classe CEchantillon() aura donc comme variable membre une liste CDataSource* (Il s'agit d'un pointeur, car il faudra la transmettre à CSerieSattique ?D()).

Suivant le type de statistique (1D ou 2D), il faut lire les valeurs des éléments de la liste comme étant des valeurs (float) ou des effectifs (int) ,de même, la CListeS devra être une série discrète, continue ou 2D suivant l'étude.

Les méthodes getXi(), getYi(),getL() devront donc être **virtuelles.**

Une remarque :

Pour créer CListeS, je vous conseille de lire une liste intermédiaire qui sera triée si vous faites une statistique 1D, ensuite de créer la CListeS. (car il faut calculer l'effectif de chaque valeur)

Si vous travailler sur une 2D, la CListeS sera crée directement.

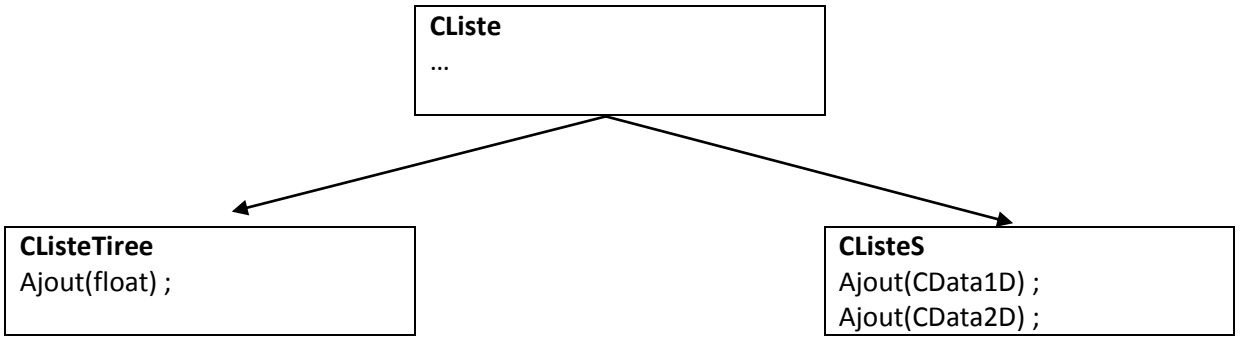
REMARQUE :

Dans le cas d'une série statistique discrète, pas de problème.

Dans le cas d'une série statistique continue, il faut évidemment connaitre la valeur minimale et la maximale ainsi que la longueur de l'intervalle. Par exemple, si on calcule le poids des hommes adultes en Belgique, on ne commence pas à partir de 0 avec un intervalle de 20kg. Cela n'aura pas de sens. Il faut donc afficher ces 2 valeurs (Min et Max), et demander à l'opérateur d'introduire la valeur de l'intervalle ainsi que son point de départ.

Même remarque pour ce qui concerne une étude 2D. Lors du graphique, il faut connaître ces 4 valeurs dans ce cas (2 pour l'axe des X et 2 pour celui des Y) afin de remplir au mieux l'écran.

Bien évidement, on a les classes suivantes :



Une remarque :

En ce qui concerne la CListe, il faut utiliser

```
template <class Type> struct Noeud
   { Type               *T;
     Noeud<Type>        *pSuivant;
   };
```


En effet, si T n'est pas un pointeur, il ne sera pas possible d'utiliser la technique du down-casting.
T sera soit un float, soit un CData.

L'application aura donc l'allure suivante :

```
…
int main(int argc, char* argv[])
{
if (argc == 2)
   { cout << "Etude 1D" << endl;
     E1 = new CEchantillon(argv[1],1);
     E1->Affiche();
     CSerieStatistique1D        C1D(E1);
     C1D.AfficheRapport();
     exit(0);
   }
if (argc == 3)
   { cout << "Etude 1D" << endl;
     E1 = new CEchantillon(argv[1],atoi(argv[2]));
     E1->Affiche();
     CSerieStatistique1D        C1D(E1);
     C1D.AfficheRapport();
     exit(0);
   }
if (argc == 4)
   { cout << "Etude 2D" << endl;
     E1 = new CEchantillon(argv[1],atoi(argv[2]),atoi(argv[3]));
     CSerieStatistique2D        C2D(E1);
     C2D.Affiche();
     C2D.Prevision();
   }
…
```

Suite :

Comme signaler précédemment, le calcul de la droite de régression est très sensible à une valeur aberrante dans la liste. (données hors étude, erreur d'encodage)

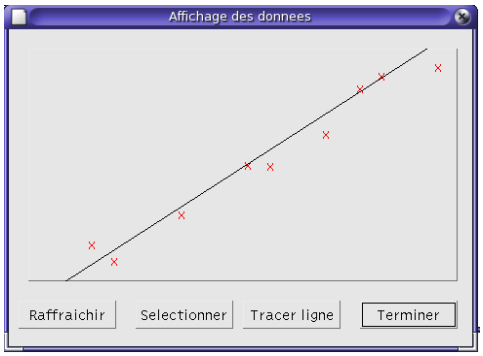
Il faut donc éliminer ces éventuelles données.

Il n'est pas possible de les déterminer par une simple lecture.

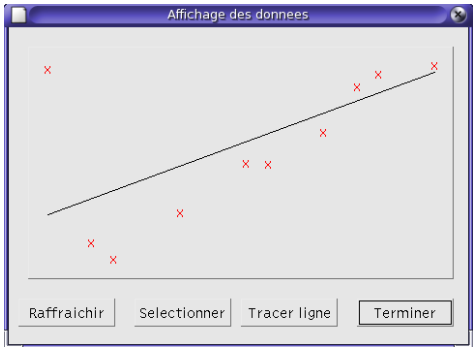
Pour cela, il faut en tracer le graphique, et exclure les données.

Exemple :

Dans notre étude sur le temps de réaction chimique en fonction de la température, on obtient le graphique suivant.



Mais, si une donnée est « fausse », (le couple (25,4.64) qui est manifestement une erreur de données) on obtient le graphique suivant :



Réalisation :

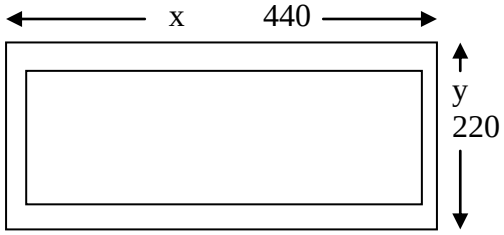
A l'aide de l'interface Qt .

On vous donne la fenêtre d'affichage des données.

Il ne vous reste plus qu'a tracer les points.

Remarque :

La taille de la frame est définie par 440 * 220 points.

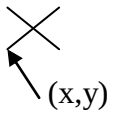


On travaillera dans le rectangle interne de 400 * 200 points.

Pour écrire un 'x', il faut utiliser la fonction

```
paint.drawText(x,y ,"x");
```

Attention:
Le caractère sera dessiné de la façon suivante :



Le caractère 'x' a une taille de 8 * 8.
Donc, pour le centré, il faudra faire y ⟶ y + 4 et x ⟶ x - 4

Remarque :

Pour remplir au maximum l'espace, il faut déterminer les valeurs minimum et maximum des 2 données, et par une règle de trois, tracer les points.
Donc, si Min1 et Max1 sont les minimum et maximum des valeurs de x, Min2 et Max2 celles des y. E1 = Max1 – Min1, on a donc la formule suivante pour une valeur X_i

$(int)((X_i - Min1)*400 / E1) + 20$

```
// +20 car on travaille dans
//le petit rectangle
```

La Source est transmise par variable globale.[5]

Il suffit maintenant de déterminer le point à éliminer. Pour cela, 2 méthodes

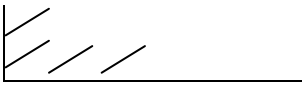
```
void FAffichage::mousePressEvent(QMouseEvent* e) ;
void FAffichage::mouseReleaseEvent(QMouseEvent* e) ;
```

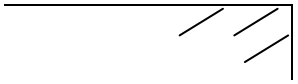
Qui déterminent les coordonnées du point lorsque l'on presse le bouton de la souris, et celles lorsque l'on le relâche.

Pour obtenir les coordonnées d'un point, il faut utiliser la fonction

```
Point = e->pos();
Point.setX(Point.x()- 20);          // -20 car on travaille dans
                                    //le petit rectangle
Point.setY(Point.y()- 20);
```

Il faut donc 2 opérateurs de surcharge pour la classe CData2D.

L'opérateur > : 

L'opérateur < : 

Et la méthode de suppression dans la CListe.
Reste un dernier petit problème. Si on lance l'affichage de la fenêtre, on ne sait plus utiliser la première partie de l'application.
FAUX, on connait les threads.
Il suffit de lancer le thread d'affichage de la fenêtre, et ensuite de synchroniser le calcul des coefficients. (rien de plus simple maintenant…)

Une fenêtre FAffichage est fournie.
(faffichage.cpp, faffichage.h, moc_faffichage.cpp)

```
FAffichage* F1;

void *Graph2D(void* D)
{
CDataSource2D* DD = (CDataSource2D*)D;
QApplication a( Argc, Argv );
    F1 = new FAffichage(DD);
    F1->show();
    a.exec();
return NULL;
}
```

Lors de la compilation, ne pas oublier les librairies (-lqt, -lpthread)

[5] Il est sera de même pour les coefficients a, b de la droite.