

## 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 connaître 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.

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 ;

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

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

$$y = a x + b$$

Où

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

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

On demande de créer une classe **EtudeStat2D** dont le rôle est de calculer les paramètres statistiques d'une série statistique à 2 dimensions, à savoir : A,B,Corr .

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

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 : Prévission pour : Température (°)  
2 : Prévission pour : Temps (min)  
3 : Sortie

1

Entrer la valeur pour Température (°) : 70  
la valeur prévue : 1.62885

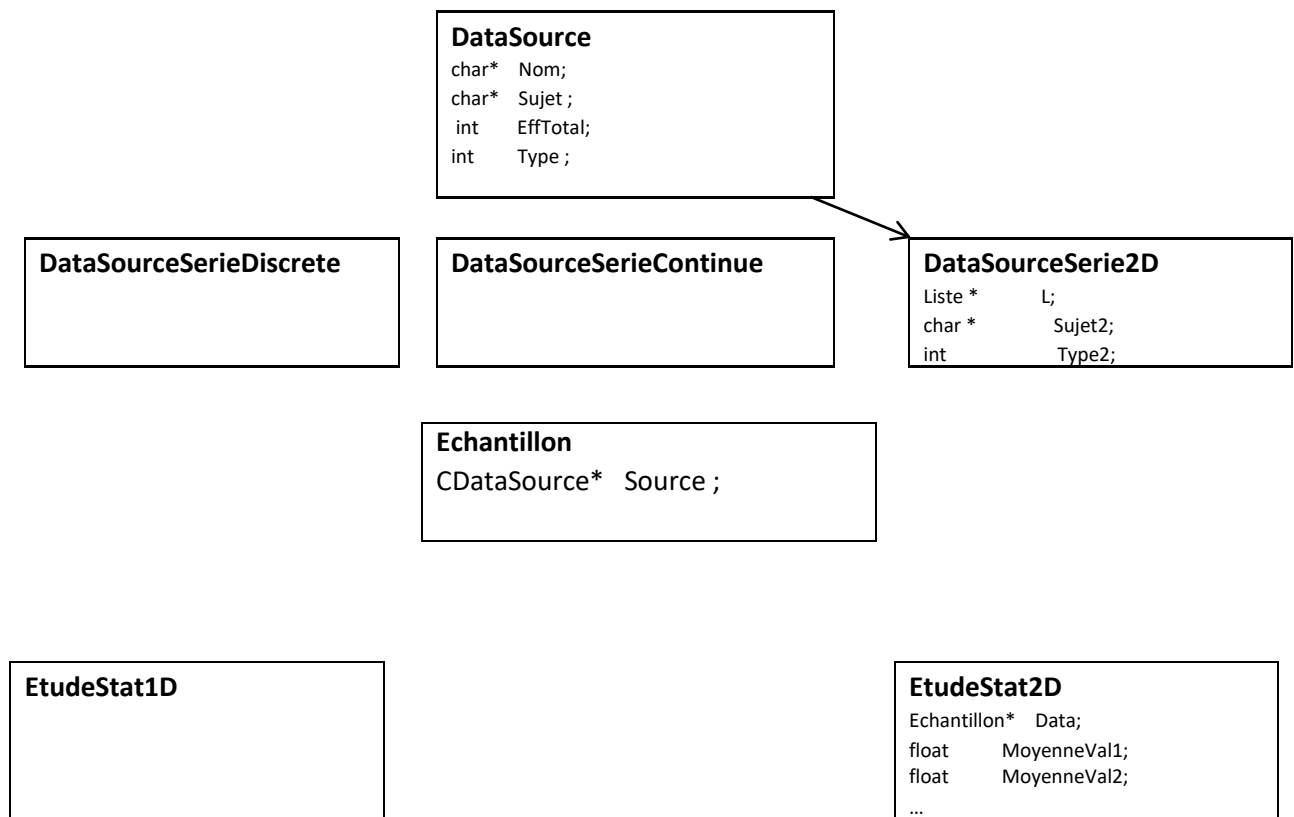
1 : Prévission pour : Température (°)  
2 : Prévission pour : Temps (min)  
3 : Sortie :

3

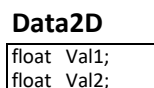
fin

---

## Réalisation :



On doit disposer d'une liste de données de 2 valeurs (Valeur, Valeur) pour une 2D. Il s'agit de la `ListeBase<Data2D>`.



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

On a les classes suivantes :

```

class EtudeStat2D
{
    private :
        Echantillon* E ;
        Float      CoeffA,
                  CoeffB,
                  CoeffCorrelation ;
    ...} ;
  
```

La classe Echantillon possède en plus le constructeur

```
Echantillon(const char*,int,int) ;
```

La classe EtudeStatDescriptive aura donc en plus :

```
...  
if (argc == 4)  
    EtudeStat2D(argv[1],atoi(argv[2]), atoi(argv[3])) ;  
...
```

Reste maintenant la class GraphStat2D :

```
private :  
  
    QPoint                PointDepart,  
                        PointArrivee ;  
  
    bool                  Pt ,  
                        TracerLigne ;  
  
    float                 MinX,MinY,MaxX,MaxY ;  
    ListeBase<Data2D>*    L;  
    ListeBase<Data2D>     LUndo;  
    EtudeStat2D*          EtudeLocal ;
```

ui->lineEquation pour pouvoir afficher l'équation de la droite de régression.

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

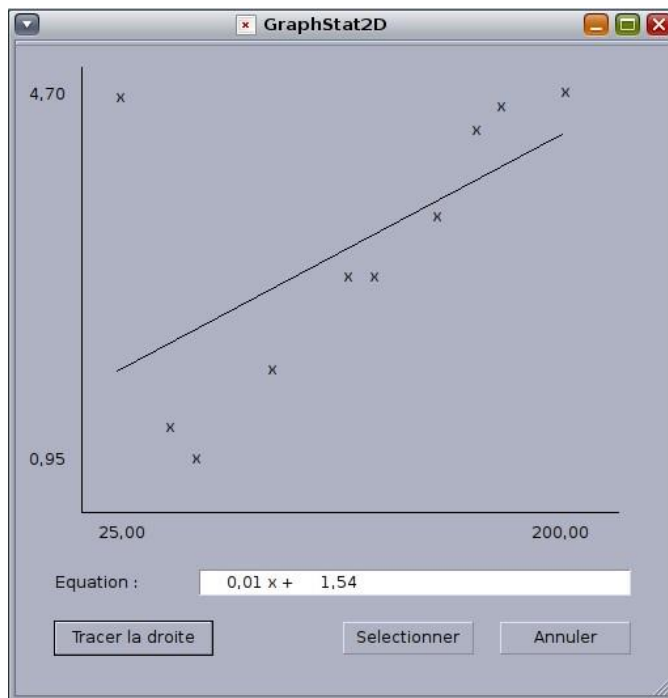
Il n'est pas possible de les déterminer par une simple lecture.

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

---

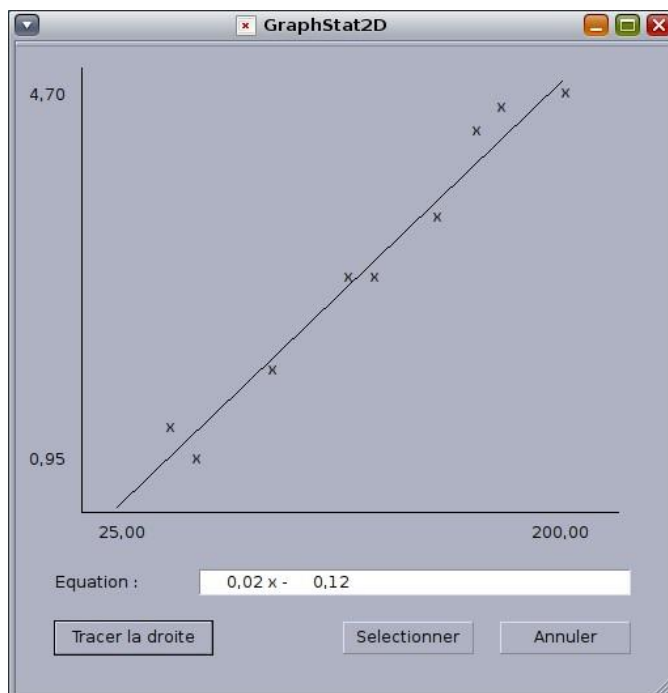
Exemple :

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



Manifestement, le couple (25,4.64) est une erreur de données). Il faut donc l'éliminer.

On obtient alors le graphique suivant :



**Réalisation :**

À l'aide de l'interface Qt.

On vous donne la fenêtre d'affichage des données avec ses méthodes et variables.

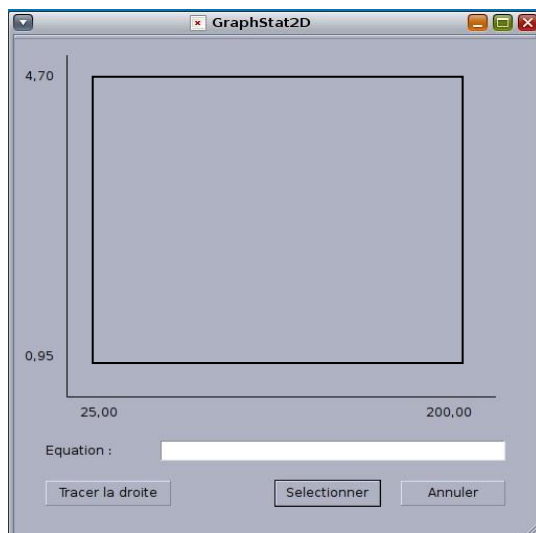
```
void paintEvent(QPainterEvent*) ;  
    qui affiche les différents couples de points, ainsi que la droite  
void mousePressEvent(QMouseEvent*) ;  
    qui détermine les coordonnées du point sélectionné  
    (PointDepart)  
void mouseReleaseEvent(QMouseEvent*) ;  
    qui détermine les coordonnées du second point sélectionné  
    (PointArrivee)
```

Des méthodes associées aux différents boutons :

```
void on_ButtonTracer_clicked() ; void  
on_Button_Selectionner_clicked() ;  
void on_ButtonAnnuler_clicked() ;
```

On peut deviner à quoi elles servent.

Elles doivent exécuter l'instruction `this->update()` ; qui permet de rafraîchir la fenêtre.

**Remarque :**

On travaillera dans le rectangle interne de 340 \* 280 points dont le coin supérieur gauche est à la position (80,40)

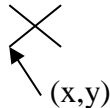
**Première étape : afficher les points**

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 6 \* 6.

Donc, pour le centrer, il faudra faire  $y \rightarrow y + 3$  et  $x \rightarrow x - 3$

**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 MinX et MaxX sont les minimum et maximum des valeurs de x, MinY et MaxY celles des y.  $E1 = \text{MaxX} - \text{MinX}$ , on a donc la formule suivante pour une valeur  $X_i$

```
80 + (Xi - MinX)*340 / E1) -3 // +80 car on travaille dans
                                // le petit rectangle
                                // -3 pour centrer le caractère X
```

**Deuxième étape : sélectionner d'éventuels points hors-norme.**

Il suffit maintenant de déterminer le point à éliminer. Pour cela, 2 méthodes déterminent les coordonnées du point lorsque l'on presse le bouton de la souris, et celles lorsque l'on le relâche :

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

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

```
PointArrivee = e->pos();
```

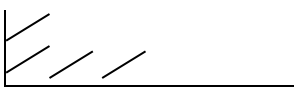
Les coordonnées du point sont

```
( PointArrivee.x(), PointArrivee.y() )
```

Il s'agit des coordonnées du point sur la fenêtre et non des valeurs Val1 et Val2 d'un éventuel point de la série. Pour cela, il faut le reconvertir (inverse de ce qui est fait plus haut)



Pour déterminer les points de la série situés dans la sélection, Il faut donc 2 opérateurs de surcharge pour la classe Data2D.

opérateur > : 

opérateur < : 

et la méthode de suppression dans la ListeBase.

Il reste à supprimer les points et les ajouter dans une autre liste :

ListeBase<Data2D> LUndo.

### Troisième partie : Annuler.

Retirer le dernier élément placé ajouté dans la liste LUndo et l'ajouter dans la Liste L de l'échantillon.

Ne pas oublier de rafraichir la fenêtre à chaque action.