

Chronique 12

Ajustement affine

Petite chronique présentant un problème d'ajustement affine tel qu'on peut en rencontrer dans certaines séries de lycée; elle permet de faire le point sur les repères cartésiens et les unités à utiliser dans ce genre d'exercice.

12.1 Le problème

Une étude sur une chaîne de magasins a été menée pour mesurer l'impact d'une campagne publicitaire sur les ventes.

Le tableau ci-dessous présente les montants des dépenses publicitaires, en milliers d'euros, et des ventes, en millions d'euros.

Dépenses publicitaires x_i en milliers d'euros	26	27	29	31	32	35
Ventes y_i en millions d'euros	4,5	4,8	4,95	5,1	5,25	5,4

On veut représenter cette série double par un nuage de points, placer le point moyen et tracer la droite d'ajustement de y en x obtenue par la méthode des moindres carrés; les calculs seront effectués à la calculatrice.

Enfin on voudrait déterminer graphiquement sur quel montant de ventes on pourrait compter avec des dépenses publicitaires de 28 et de 37 milliers d'euros.

Les calculs donnent $G(30, 5)$ comme point moyen et $y = 0,09375x + 2,1875$ pour équation de la droite d'ajustement.

En remplaçant x par 28 dans $y = 0,09375x + 2,1875$, on trouve $y = 4,8125$ et si on remplace x par 37, on trouve $y = 6,65625$.

Voir la représentation graphique attendue page suivante.

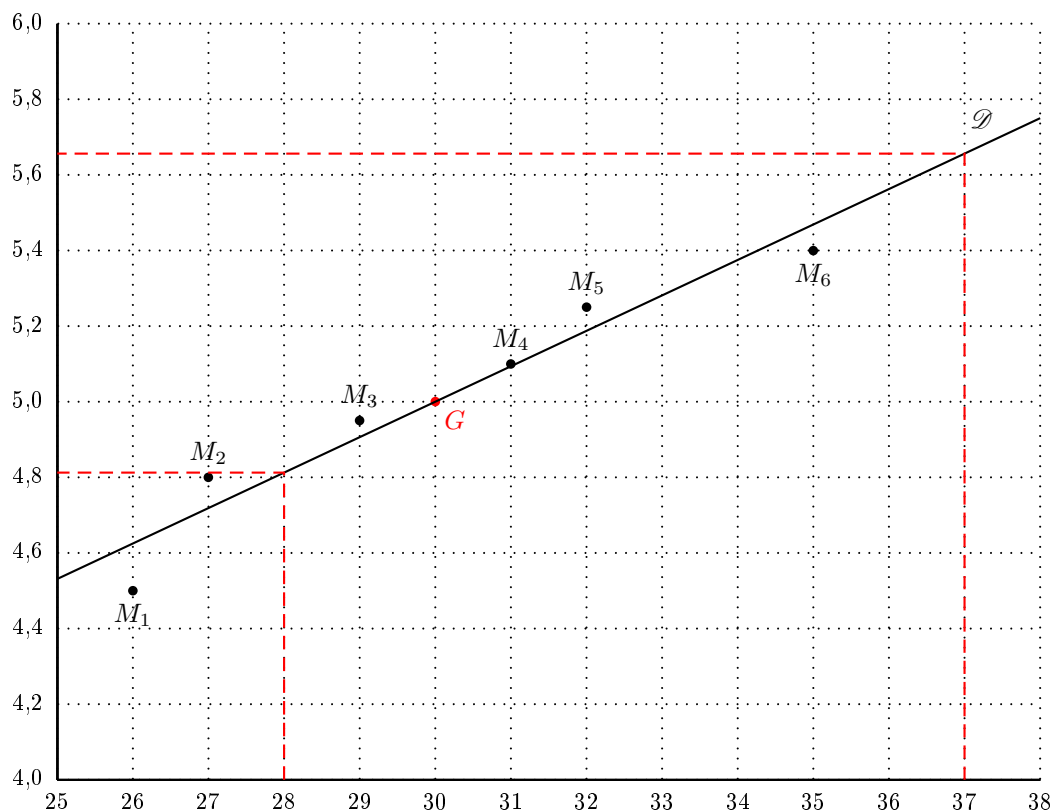
12.2 Le repère et les axes

Compte tenu des données du problème, on prendra des valeurs de x entre 25 et 38 et des valeurs de y entre 4 et 6. On prend 1 cm comme unité en abscisse et 5 cm en ordonnée.

Les axes seront tracés de `\xmin` à `\xmax` horizontalement, et de `\ymin` à `\ymax` verticalement.

On placera l'origine en $(25, 4)$, c'est-à-dire en $(\texttt{\textbackslash xmin}, \texttt{\textbackslash ymin})$.

Enfin on retirera les marques d'unités (`ticks=none`) et les légendes automatiques (`labels=none`) sur les axes.



Ce qui donne comme squelette de code :

```
\psset{xunit=1cm,yunit=5cm}
\def\xmin{25} \def\xmax{38}
\def\ymin{4} \def\ymax{6}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\psaxes[labels=none,ticks=none](\xmin,\ymin)(\xmax,\ymax)
...
\end{pspicture}
```

12.3 Le quadrillage et les légendes

On ne pourra pas utiliser l'instruction `\psgrid` pour dessiner le quadrillage de la figure ; en effet cette instruction utilise `xunit` et `yunit` pour déterminer l'espacement des divisions principales. Il faudrait alors diviser par 5 l'espacement en ordonnée, sans diviser l'espacement en abscisse, ce qui ne semble pas possible.

On va donc tracer « à la main » le quadrillage en utilisant l'instruction `\multido` deux fois : une fois pour les traits verticaux, une fois pour les traits horizontaux puisqu'il n'y a pas le même nombre de traits dans les deux directions.

Verticalement

On va tracer 14 segments en pointillés entre les points $(25,4)$ et $(25,6)$, puis entre $(26,4)$ et $(26,6)$, ..., puis entre $(38,4)$ et $(38,6)$. Et comme on a paramétré le graphique, on utilisera `\ymin` à la place de 4, `\ymax` à la place de 6, et `\xmin` à la place de 25.

Dans la même boucle, on écrira les légendes en utilisant `\uput`.

On aura donc besoin d'une variable de type `integer` qui partira de 25 (`\xmin`) et augmentera de 1 en prenant en tout 14 valeurs.

On aurait pu se contenter de 13 valeurs en partant de 26 pour tracer les droites verticales, mais il faut quand même écrire 25 en dessous de l'origine donc j'ai choisi de démarrer à 25. Voici le code de la boucle `\multido` :

```
\multido{\i=\xmin+1}{14}
{
  \uput[d](\i,\ymin){\small \i}
  \psline[linestyle=dotted](\i,\ymin)(\i,\ymax)
}
```

Horizontalement

On va tracer des segments en pointillés de (4;25) à (4;38), puis de (4,2;25) à (4,2;38), ..., puis de (6;25) à (6;38).

Au passage on écrira les légendes en utilisant `\uput`.

Comme compteur de `\multido`, il faudra une variable de type `number` qui partira de 4 en allant jusqu'à 6 par pas de 0,2; il faudra donc 11 valeurs.

On va également faire en sorte d'écrire 4,2, 4,4, etc. avec une virgule à la place du point; il faut pour cela charger l'extension `numprint` qui modifie l'écriture décimale en reconnaissant que l'on écrit en français.

Ce package chargé, on utilisera la fonction `\nombre` : ainsi `\nombre{5.4}` donnera 5,4.

On peut également charger l'extension avec l'option `np` pour utiliser le raccourci `\np` à la place de `\nombre`; on charge donc l'extension en tapant :

```
\usepackage[np]{numprint}
```

Voici le code de la boucle :

```
\multido{\n=\ymin+0.2}{11}
{
  \uput[1](\xmin,\n){\small \np{\n}}
  \psline[linestyle=dotted](\xmin,\n)(\xmax,\n)
}
```

L'extension `\numprint` permet aussi de regrouper 3 par 3 les chiffres des nombres qui dépassent trois chiffres; ainsi le nombre 1234567 peut être écrit 1 234 567 en utilisant `\np{1234567}`.

C'est aussi valable pour les chiffres à droite de la virgule; on obtient 3,141 592 653 5 en entrant `\np{3,1415926535}`. On retient les premières décimales de π (π) en comptant les lettres des mots de la strophe bien connue :

```

| Que j'aime à faire connaître ce nombre utile aux sages
| Immortel Archimède, artiste ingénieur
| ...
```

12.4 Les points

Il s'agit maintenant de placer les points du nuage dans le graphique; comme les unités sont adaptées au problème, il n'y a aucune précaution particulière à prendre.

J'ai déjà parlé d'une petite commande personnelle appelée `\point` qui place le point et marque son nom en même temps :

```
\newcommand*\point[4]{%
  \psdots(#1,#2)
  \uput[#3](#1,#2){#4}}
% permet de placer un point et de marquer son nom en même temps
% 4 paramètres : abscisse, ordonnée, emplacement et nom
```

On traite de façon spéciale le point G que l'on veut tracer en rouge.

12.5 La droite

Pas de problème pour tracer la droite d'équation $y = 0,09375x + 2,1875$ avec `\psplot`; on entrera comme définition de la fonction `{x 0.09375 mul 2.1875 add}`.

Enfin on tracera des segments en mode tirets (`dashed`) pour faire l'interpolation en $x = 28$ et l'extrapolation en $x = 37$.

12.6 Le code complet

Voici le code complet du graphique :

```
\psset{xunit=1cm, yunit=5cm}
\def\xmin{25} \def\xmax{38}
\def\ymin{4.0} \def\ymax{6.0}
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
%\psgrid[subgriddivx=5]
\psaxes[labels=none,ticks=none](\xmin,\ymin)(\xmax,\ymax)

\multido{\i=\xmin+1}
{14}%           nombre de traits
{
\uput[d](\i,\ymin){\small \i}%           écrit 25 26 27 etc.
\psline[linestyle=dotted](\i,\ymin)(\i,\ymax)% pointillés verticaux
}

\multido{\n=\ymin+0.2}
{11}%           nombre de traits
{
\uput[l](\xmin,\n){\small \np{\n}}%           écrit 4,0 4,2 etc.
\psline[linestyle=dotted](\xmin,\n)(\xmax,\n)% pointillés horizontaux
}

\point{26}{4.5}{d}{\$M_1$}
\point{27}{4.8}{u}{\$M_2$}
\point{29}{4.95}{u}{\$M_3$}%           points du nuage
\point{31}{5.1}{u}{\$M_4$}
\point{32}{5.25}{u}{\$M_5$}
\point{35}{5.4}{d}{\$M_6$}

\psdots[linecolor=red](30,5)%           point G
\uput[dr](30,5){\red $G$}

\uput[ul](37.5,5.7){$\mathscr{D}$}%           nom de la droite

\psplot{\xmin}{\xmax}{x 0.09375 mul 2.1875 add}% tracé de la droite

\psset{linestyle=dashed, linecolor=red}%           en rouge et en pointillés
\psline(28,\ymin)(28,4.8125)(\xmin,4.8125)$           pour x=28
\psline(37,\ymin)(37,5.65625)(\xmin,5.65625)$           pour x=37

\end{pspicture}
```