TP 5 – SY02
Régression linéaire

En R, pour réaliser une régression linéaire, on appelle la fonction `lm` (*linear model*). Le premier argument de `lm` est un nouvel objet R qu'on appelle une formule et qui spécifie une « sortie » et des « entrées » séparées par le signe `~`. Les entrées et sortie sont des noms de colonnes d'un `data.frame` qu'il faut spécifier en deuxième argument. Par exemple, si on veut réaliser la régression des données `vary` en fonction des données `varx`, on écrira

```
donnees <- data.frame(varx = c(0, 0.2, 0.3, 0.6),
                      vary = c(1.01, 1.44, 1.55, 2.1))
lm(vary~varx, data = donnees)
```

On fera attention à l'ordre des éléments dans une formule. La variable à régresser se situe à gauche, le ou les régresseurs à droite.

(1) Quelles sont les estimations de l'ordonnée à l'origine (*intercept*) $\widehat{a}$ et de la pente $\widehat{b}$?

(2) À l'aide des fonctions `plot` et `abline`, tracer les points de coordonnées `x` et `y` ainsi que la droite des moindres carrés.

Pour avoir plus d'informations sur la régression effectuée, il faut stocker l'objet renvoyé par la fonction `lm` dans une variable et appeler la fonction `summary` avec cette variable en argument.

Toutes les données affichées par `summary` sont accessibles programmatiquement (voir la table 1 pour quelques exemples)

(3) À l'aide des correspondances indiquées dans la table 1, vérifier que la somme des résidus vaut 0 et que l'image de $\overline{x}$ par la droite des moindres carrés est $\overline{y}$.

1 Qualité de l'ajustement

1.1 Équation d'analyse de la variance

(4) À l'aide des correspondances indiquées dans la table 1, calculer successivement

1. La variance totale

$$S_Y^2 = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2,$$

Notations	Code `R`
x_i	`x`
y_i	`y`
$\overline{x}$	`mean(x)`
$\overline{y}$	`mean(y)`
$\widehat{y}_i$	`m$fitted.values`
$y_i - \widehat{y}_i$	`m$residuals`
$\widehat{a}$	`m$coefficients[1]`
$\widehat{b}$	`m$coefficients[2]`

TABLE 1 – Correspondances notations/code `R`, où `m` est l'objet renvoyé par la fonction `lm`

2. La variance expliquée par le régression

$$S_{\text{reg}} = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{Y}_i - \overline{Y}\right)^2,$$

3. La variance résiduelle

$$S_{\text{res}} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2.$$

Vérifier que la variance totale est la somme de la variance expliquée par la régression et de la variance résiduelle.

(5) On appelle **coefficient de détermination**, noté R^2 la proportion de la variance expliquée dans la variance totale soit :

$$R^2 = \frac{S_{\text{reg}}}{S_Y^2}.$$

Calculer cette quantité et vérifier que c'est en fait le carré du coefficient de corrélation de Pearson entre les x_i et les y_i. Montrer que c'est également le carré du coefficient de corrélation de Pearson entre les observations y_i et les prédictions $\widehat{y}_i$.

1.2 Indépendance et normalité des résidus

Le coefficient de détermination est insuffisant pour rendre compte de la qualité de l'ajustement. À titre d'exemple, on utilise le jeu de données d'Anscombe qui consiste en 4 ensembles de 11 points du plan décrits à la figure 1. Pour rendre directement disponibles les colonnes en tapant leur nom, on pourra « attacher » ce jeu de données avec l'instruction

```
attach(anscombe)
```

Dès lors, au lieu de spécifier le jeu de données puis le nom de colonne

```
anscombe$x1
```

on peut se contenter de spécifier `x1`.

De même, pour éviter de définir systématiquement un `data.frame` avant de l'utiliser dans `lm`, on peut utiliser directement des vecteurs dans des formules. On peut alors simplement écrire

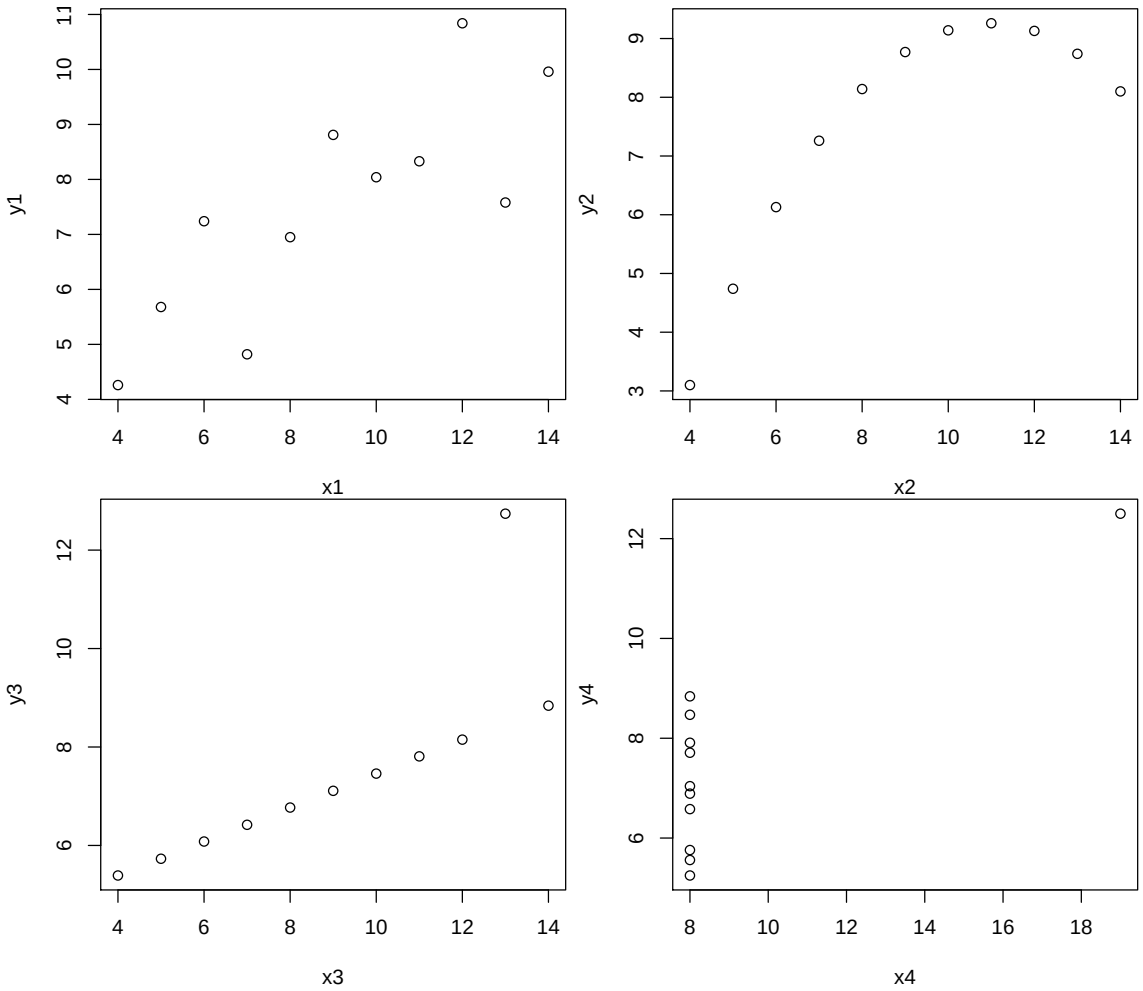

FIGURE 1 – Diagrammes de dispersion des 4 ensembles de 11 points du jeu de données d'Anscombe

```
lm(y1 ~ x1)
```

Attention, cette écriture rend impossible la prédiction en de nouveaux points. On préférera donc utiliser la syntaxe détaillée en début de TP lorsqu'il sera nécessaire de faire de la prédiction.

(6) Effectuer les régressions linéaires sur les 4 ensembles de points. Que remarquez-vous ?

Pour estimer la qualité de l'ajustement, on utilise le fait que le modèle de régression linéaire suppose que les résidus suivent une loi normale et sont indépendants.

(7) Évaluer visuellement la normalité des résidus à l'aide d'un diagramme quantile–quantile avec des fonctions `qqnorm` et `qqline`.

(8) Évaluer visuellement l'indépendance des résidus en représentant les résidus en fonction des prédictions. (`fitted.values`)

2 Prédiction

Le fichier hooker-data.data contient un jeu de données recueillies par le botaniste anglais Joseph Dalton Hooker. Il s'agit de températures d'ébullition de l'eau relevées pour différentes altitudes.

(9) Faire une étude de régression linéaire qui explique la pression atmosphérique.

(10) À l'aide de la function `confint`, donner un intervalle de confiance sur les coefficients de la droite des moindres carrés au niveau de confiance $1 - \alpha = 0.99$.

(11) À l'aide de la fonction `predict`, calculer un intervalle de confiance sur la pression pour une température d'ébullition mesurée de 97 °C.

Pour plus d'informations sur les arguments à fournir à la fonction `predict`, on pourra utiliser l'instruction suivante

```
?predict.lm
```

3 Étude de cas

La loi de Moore est une loi empirique qui dit que le nombre de transistors croit de manière exponentielle avec le temps. Autrement dit, on suppose que le nombre de transistors N_t au temps t est égal à

$$N_t = \alpha \exp(\beta t).$$

(12) À l'aide du fichier moore-data.data et en utilisant une régression linéaire, estimer les paramètres α et β et donner un intervalle de confiance et de prédiction sur N_{2018}. Retrouver le fait que le nombre de transistors double tous les 2 ans.