

S04: Statistiques multivariées

Analyse de données quantitatives avec R

Samuel Coavoux

1 Modèles de regression

Modèles de regression

Régression linéaire: lm

Classe formule

La fonction `lm()` (linear models) permet d'ajuster des modèles de régression linéaire.

Elle prend comme premier argument un objet de classe **formule** ; la variable dépendante est précisée en premier, suivi d'un tilde (~), puis de l'interaction entre variables indépendantes.

- L'interaction est habituellement précisée par + (dans le modèle linéaire classique: $Y = \alpha + \beta_1 X_1 + \beta_2 X_2$).
- On peut également ajouter, plutôt qu'une variable, l'interaction entre deux variables, en employant :. $x \sim a:b$ revient à chercher $Y = \alpha + \beta_1 X_1 X_2$ (surtout utile pour les régressions logistiques, lorsque l'on cherche l'interaction entre deux facteurs corrélés).
- Enfin, * cherche à la fois l'addition et l'interaction. $x \sim a*b$ est équivalent à $x \sim a + b + a:b$.

Classe formule

On peut enfin transformer les variables directement dans une formule. Par exemple $x \sim a + \log(b)$. Si l'on souhaite utiliser un terme réservé pour la classe formule comme `+`, il faut l'enclore dans `I()`. Ainsi, $x \sim a + b$ prend `a` et `b` comme variable indépendante, alors que $x \sim I(a + b)$ prend **la somme de a et b** comme variable indépendante.

lm()

Pour éviter d'avoir à répéter le nom du data.frame pour chaque variable de la formule, on peut employer l'argument data. Ainsi, les deux notations ci-dessous sont équivalentes

```
lm(imueclt ~ happy + income_dec, data=d)  
lm(d$imueclt ~ d$happy + d$income_dec)
```

L'argument weights permet de préciser un vecteur de pondération.

Explorer un modèle

Par défaut, la méthode print des objets lm (`stats:::print.lm()`) donne assez peu d'informations : seulement les coefficients, et la commande employée pour produire le résultat. `summary.lm()` est beaucoup plus disert. On y obtient:

- un summary des residu;
- les coefficients avec l'erreur standard, la valeur t et la p-value associée au test de nullité;
- quelques indicateurs de l'ajustement du modèle: R-squared, F, p-value;
- le nombre de valeurs manquantes

Variables

La variable dépendante doit être une variable numérique. `lm()` accepte des `factor`, mais c'est particulièrement déconseillée (en gros, la variable `factor` devrait être transformée en numérique, de sorte que votre variable dépendante sera discrète et prendra comme valeur 1 à k où k est le nombre de modalités).

Les variables indépendantes peuvent être des `factors`. Dans ce cas, la première modalité (le premier `level`) sera considéré comme la modalité de référence, et les coefficients des autres modalités sera calculé. Pour changer de modalité de référence rapidement (c'est à dire pour passer un `level` en premier `level` d'un `factor` sans avoir à réécrire `factor(x, levels=c(liste des levels))`), on peut employer `relevel()`

```
d$gndr <- relevel(d$gndr, ref = "Female")
```

Explorer un modèle: print()

```
ll <- lm(imueclt ~ happy + income_dec, data=d)
print(ll)

##
## Call:
## lm(formula = imueclt ~ happy + income_dec, data = d)
##
## Coefficients:
## (Intercept)      happy    income_dec
##           3.2059       0.2471       0.1210
```

Explorer un modèle: summary()

```
summary(l1)
```

```
##  
## Call:  
## lm(formula = imueclt ~ happy + income_dec, data = d)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -6.8865 -1.5236  0.1907  1.7286  6.6731  
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 3.205880  0.058305 54.98 <2e-16  
## happy       0.247081  0.007542 32.76 <2e-16  
## income_dec  0.120984  0.005067 23.88 <2e-16  
##
```

Valeurs manquantes

Attention! Par défaut, lm supprime les lignes de la base de données contenant une valeur manquante. On peut facilement se retrouver, dans une enquête par questionnaire, à faire des régressions sur quelques pourcents de l'échantillon si l'on ajoute trop de variables sans y prendre garde. Il convient donc de:

- limiter le nombre de variables;
- recoder en amont les NA autant que possible.

Explorer un modèle: plot()

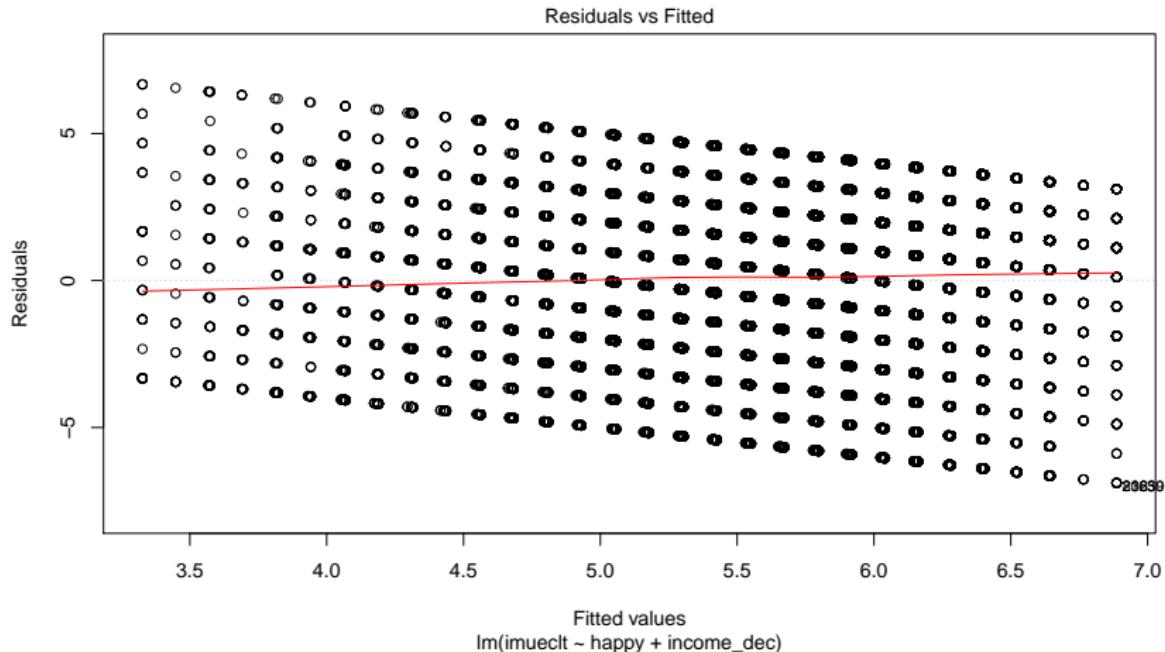
La fonction de base pour représenter graphiquement des modèles est la méthode `plot.lm()`. Par défaut, elle produit 4 graphiques (on peut en choisir un seul avec `which`):

- un scatterplot des résidu par valeur prédictive de Y (1);
- un diagramme Quantile-Quantile des résidu studentisé (2);
- un scatterplot de la racine des résidu studentisé par valeur prédictive de Y (3);
- un scatterplot des résidu studentisé pour les outliers (5).

Ces graphes devraient permettre de faire un premier diagnostic sur l'ajustement du modèle : vérifier la normalité des résidus et l'homoscédasticité du modèle.

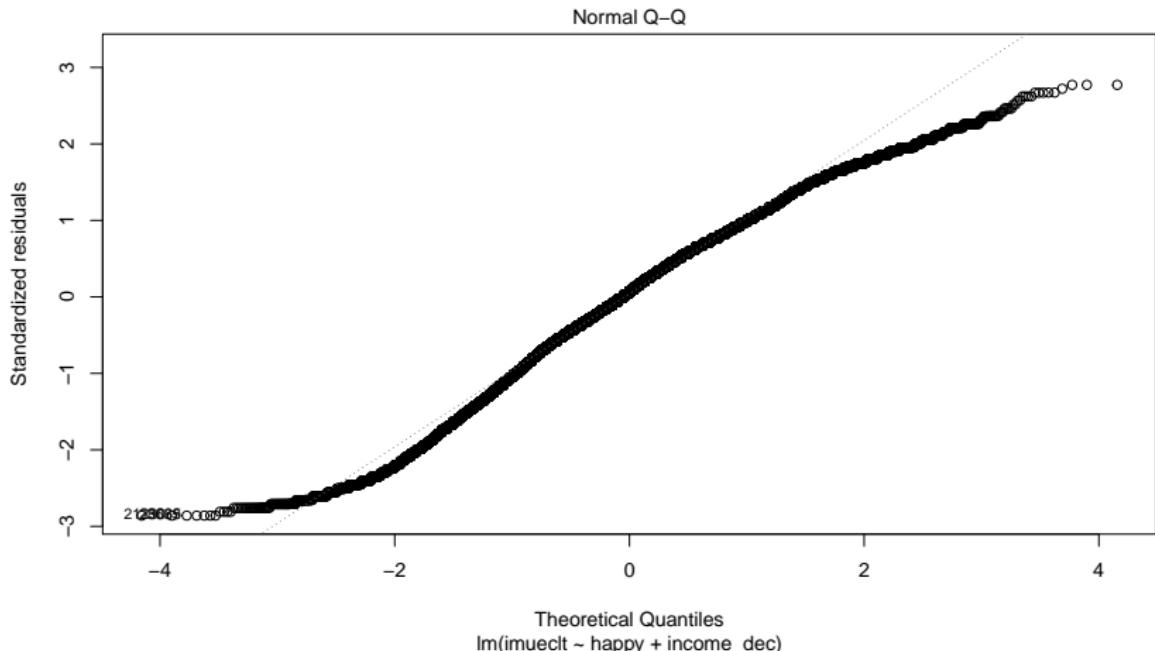
Explorer un modèle: plot()

```
plot(l1, which = 1)
```



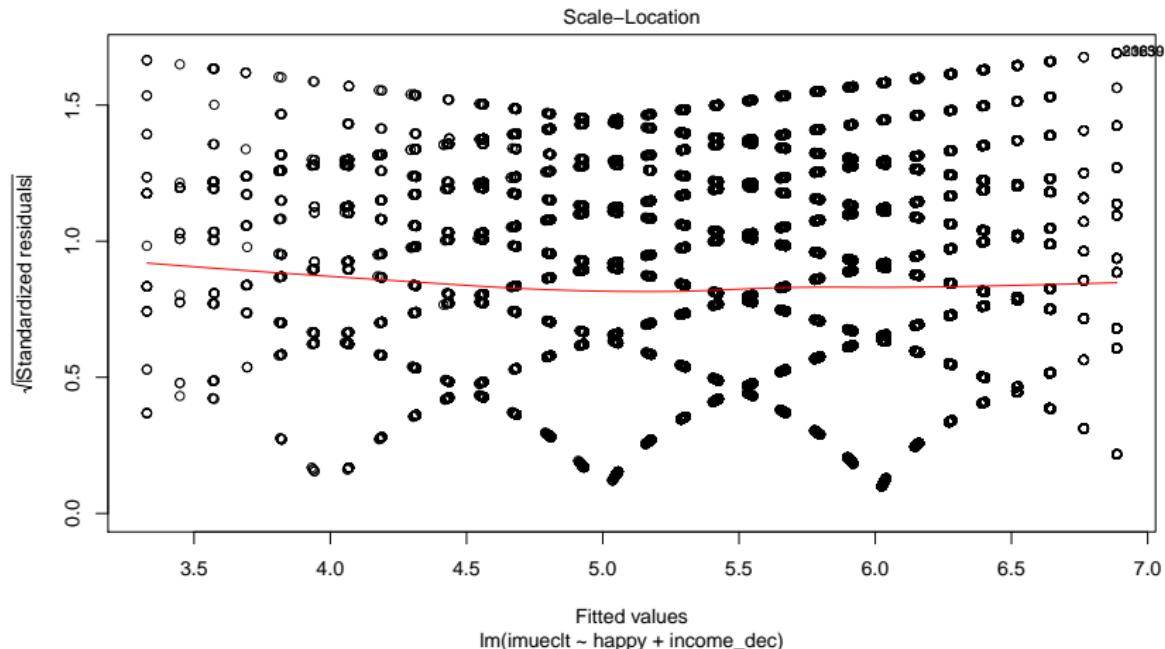
Explorer un modèle: plot()

```
plot(l1, which = 2)
```



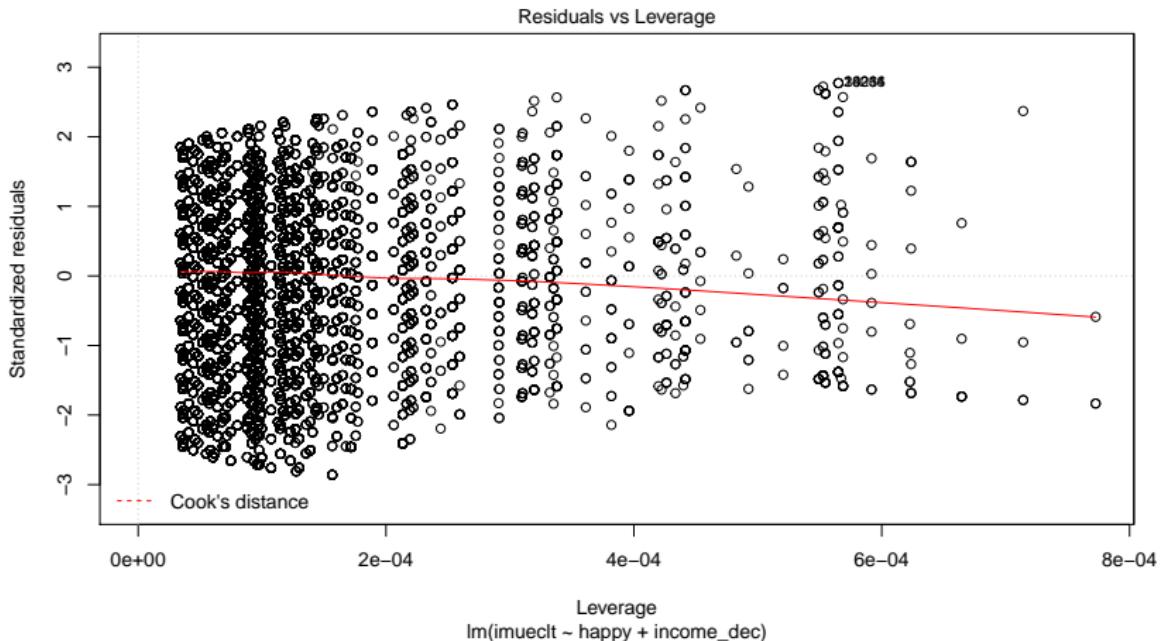
Explorer un modèle: plot()

```
plot(l1, which = 3)
```



Explorer un modèle: plot()

```
plot(l1, which = 5)
```



Explorer un modèle: résultats de lm()

```
names(l1)
```

```
## [1] "coefficients"   "residuals"  
## [3] "effects"        "rank"  
## [5] "fitted.values"  "assign"  
## [7] "qr"              "df.residual"  
## [9] "na.action"       "xlevels"  
## [11] "call"            "terms"  
## [13] "model"
```

Accéder aux coefficients

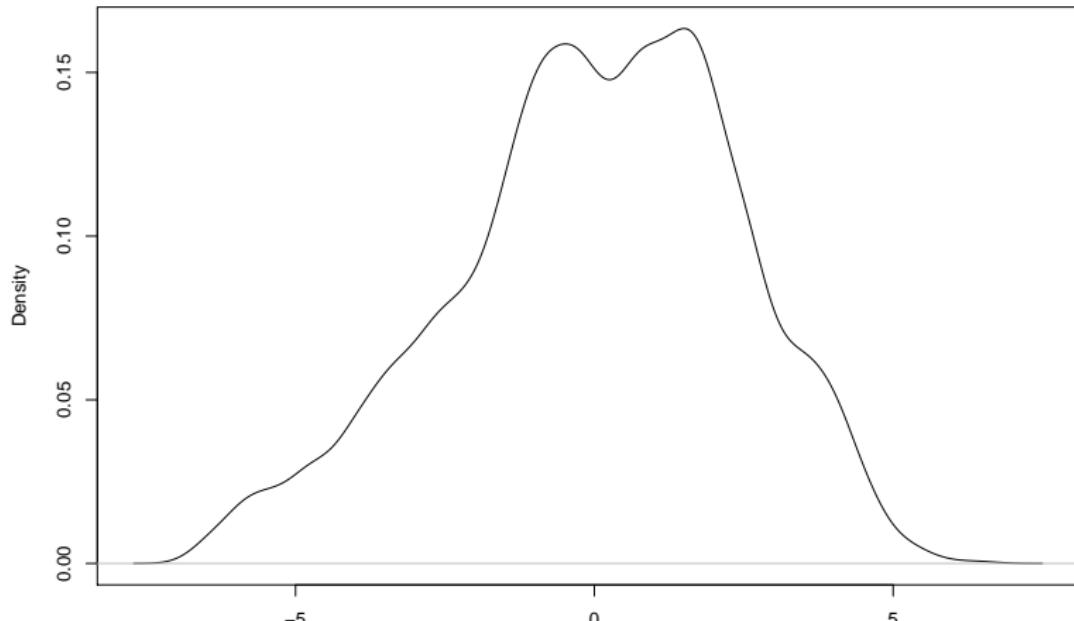
```
ll$coefficients
```

```
## (Intercept)      happy   income_dec
## 3.2058797    0.2470806   0.1209840
```

Accéder aux résidus

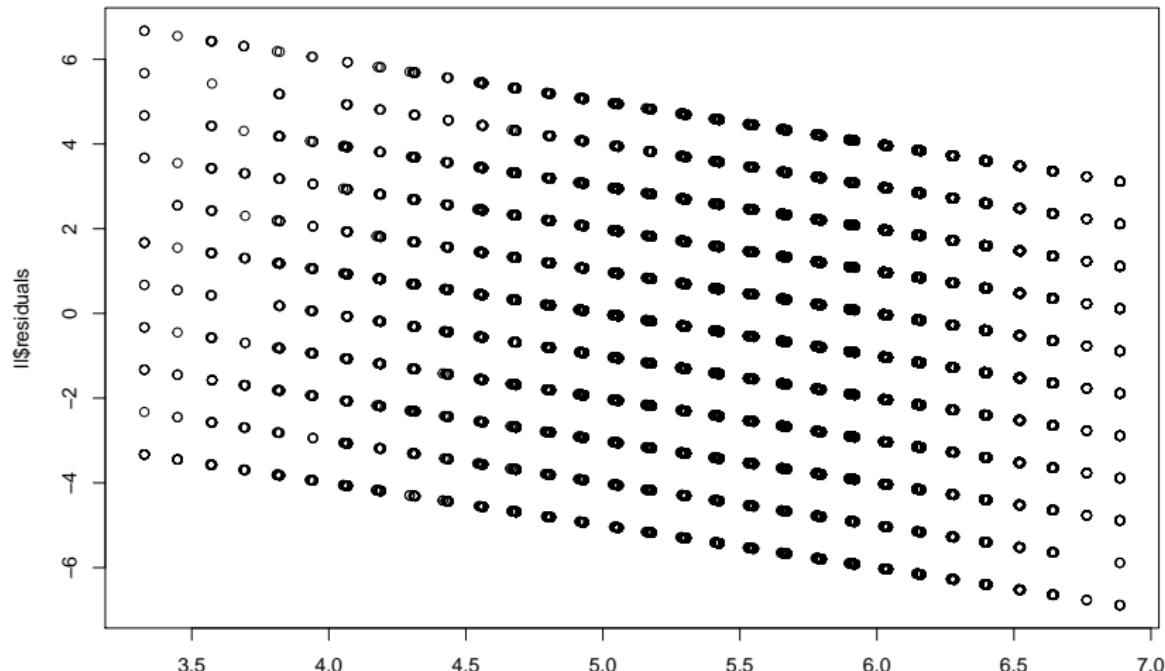
```
plot(density(l1$residuals))
```

```
density.default(x = l1$residuals)
```



Accéder aux résidus

```
plot(l1$fitted.values, l1$residuals)
```



ANOVA

Anova

Pour réaliser une ANOVA, on commence par ajuster un modèle linéaire.

```
ma <- anova(lm(imueclt ~ cntry, data = d))
```

Anova

Contrairement à lm, summary() ne donne pas d'information intéressante, et il faut employer print() pour afficher les informations sur le test.

ma

```
## Analysis of Variance Table
##
## Response: imueclt
##              Df Sum Sq Mean Sq F value    Pr(>F)
## cntry        20 20372 1018.62 179.94 < 2.2e-16
## Residuals 38812 219714      5.66
##
## cntry      ***
## Residuals
## ---
## Signif. codes:
```