Regression Lineaire en R (5)

October 17, 2018

Greta Laage,Luc Adjengue

Contents

1 Régression linéaire simple

1.1 Introduction

Les données utilisées pour cet exemple sont celles de l'exercice 3 du TD11.

Pour faire une analyse de régression linéaire, il faut suivre les étapes suivantes:
1 - Charger les données
2 - Définir un objet qui contiendra la régression linéaire
3 - Analyser les résultats de la régression, obtenus à partir de l'objet défini dans l'étape 2.

```
In [2]: # Charger les données
        X=c(14,18,40,43,45,112)
        Y=c(280,350,470,500,560,1200)
```

Le modèle de régression linéaire simple s'écrit:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

1.2 Ajuster un modèle de régression

Pour ajuster un modèle de régression linéaire, on utilise la fonction **lm()**. La syntaxe standard est $lm(y \sim x, data)$ où x est la variable indépendante et y la variable dépendante. Il n'est pas nécéssaire d'ajouter le paramètre *data* si les données sont bien définies. Si par exemple on avait une table de données *Table* avec deux colonnes Y et X, on pourrait écrire une des deux possibilités ci-dessous:

- $lm(Table\$Y \sim Table\$X)$
- $lm(Y \sim X, data = Table)$

```
In [4]: # On définit un objet qui contient les résultats de la régression linéaire
        linModel <- lm(Y~ X)
```

1.3 Afficher les résultats de la régression

On utilise la fonction **summary()** qui renvoie le résultat de la régression linéaire et notamment les coefficients de régression de chaque variable indépendante.

```
In [20]: # Afficher les résultats de la régression linéaire
         summary(linModel)
```

```
Call:
lm(formula = Y ~ X)

Residuals:
      1       2       3       4       5       6
 11.762  44.516 -40.338 -38.273   3.104  19.229

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 137.8756    26.3776   5.227   0.0064 **
X             9.3116     0.4745  19.622 3.98e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 37.39 on 4 degrees of freedom
Multiple R-squared:  0.9897,Adjusted R-squared:  0.9871
F-statistic:   385 on 1 and 4 DF,  p-value: 3.978e-05
```

La fonction **summary()** ci-dessus donne les informations dans l'ordre suivant:

- Rappel du modèle
- Résidus pour chaque observation (ils ne sont pas donnés lorsque l'ensemble de données est trop grand)
- Valeur des coefficients de régression avec leur estimation **Estimate** ainsi que les résultats du test d'hypothèse de Student testant $\beta_i = 0$. On peut voir ainsi très rapidement si un coefficient est statistiquement significatif avec sa p-value.
- RSE, R^2 et R^2_{adj}, la statistique F du test de signification de la régression ainsi que la valeur-p de ce test.

Pour connaître toutes les informations contenues dans l'objet **linModel** défini ci-dessus, on peut utiliser la fonction **names()**. Pour accéder à chacune de ces informations, on utilise le signe $

```
In [21]: names(linModel)
```

1. 'coefficients' 2. 'residuals' 3. 'effects' 4. 'rank' 5. 'fitted.values' 6. 'assign' 7. 'qr' 8. 'df.residual' 9. 'xlevels' 10. 'call' 11. 'terms' 12. 'model'

```
In [22]: # accéder aux coefficients
         linModel$coefficients
```

(Intercept) 137.875630974116 **X** 9.31156696380625

```
In [24]: # accéder au R^2
         summary(linModel)$r.sq
```

0.989717856763518

1.4 Intervalles de confiance et de prédiction

Pour obtenir un intervalle de confiance pour les estimations des coefficients, on utilise la fonction **confint()**. Par défaut le niveau de confiance est à 5%, on peut le modifier avec le paramètre level.

```
In [9]: confint(linModel)
        confint(linModel, level = 0.90)
```

	2.5 %	97.5 %
(Intercept)	64.639768	211.11149
X	7.994014	10.62912
	5 %	95 %
(Intercept)	81.642703	194.10856
X	8.299906	10.32323

On peut obtenir des intervalles de prédiction ou de confiance pour des nouvelles valeurs de X avec la fonction **predict()**. Il faut préciser :

- le modèle
- la nouvelle donnée indiquée sous la forme d'un dataframe
- le niveau de confiance (il est à 95% par défaut)
- le type d'intervalle recherché

```
In [11]: predict(linModel, data.frame(X = 80), level = 0.95, interval = "prediction")
         predict(linModel, data.frame(X = 80), level = 0.90, interval = "confidence")
```

	fit	lwr	upr
1	882.801	761.7318	1003.87
	fit	lwr	upr
1	882.801	834.9597	930.6423

Si on veut ces intervalles pour plusieurs valeurs à la fois, on les met sous forme de vecteurs.

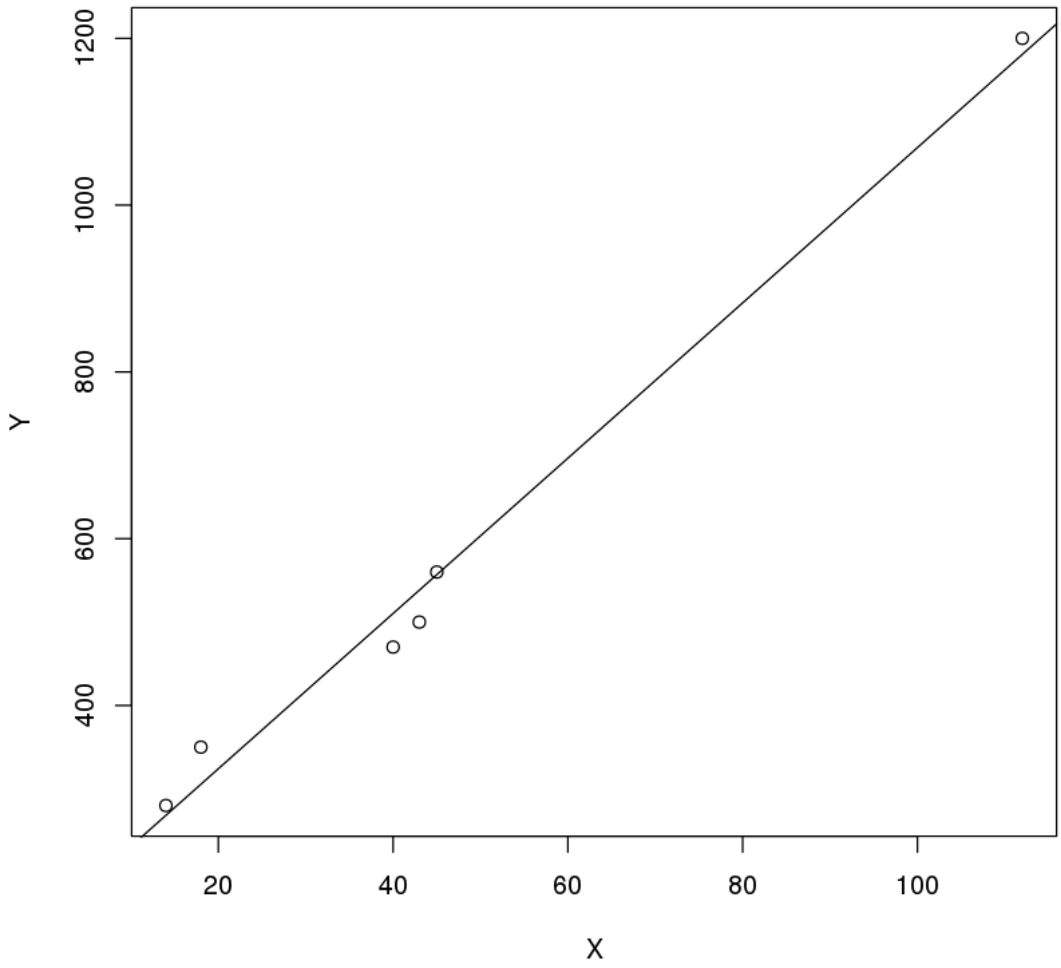
```
In [12]: predict(linModel, data.frame(X = c(80,70,60,68)),
                  level = 0.95, interval = "prediction")
```

	fit	lwr	upr
1	882.8010	761.7318	1003.8701
2	789.6853	672.9474	906.4233
3	696.5696	582.7938	810.3455
4	771.0622	655.0302	887.0941

1.5 Tracés

On peut tracer le nuage de points avec la fonction **plot()** et ajouter la droite de régression calculée avec la fonction **abline()**.

```
In [16]: plot(X,Y)
         abline(linModel)
```



1.6 Analyse de la variance

On utilise la fonction **anova()** qui fournit le tableau d'analyse de la variance

```
In [17]: anova(linModel)
```

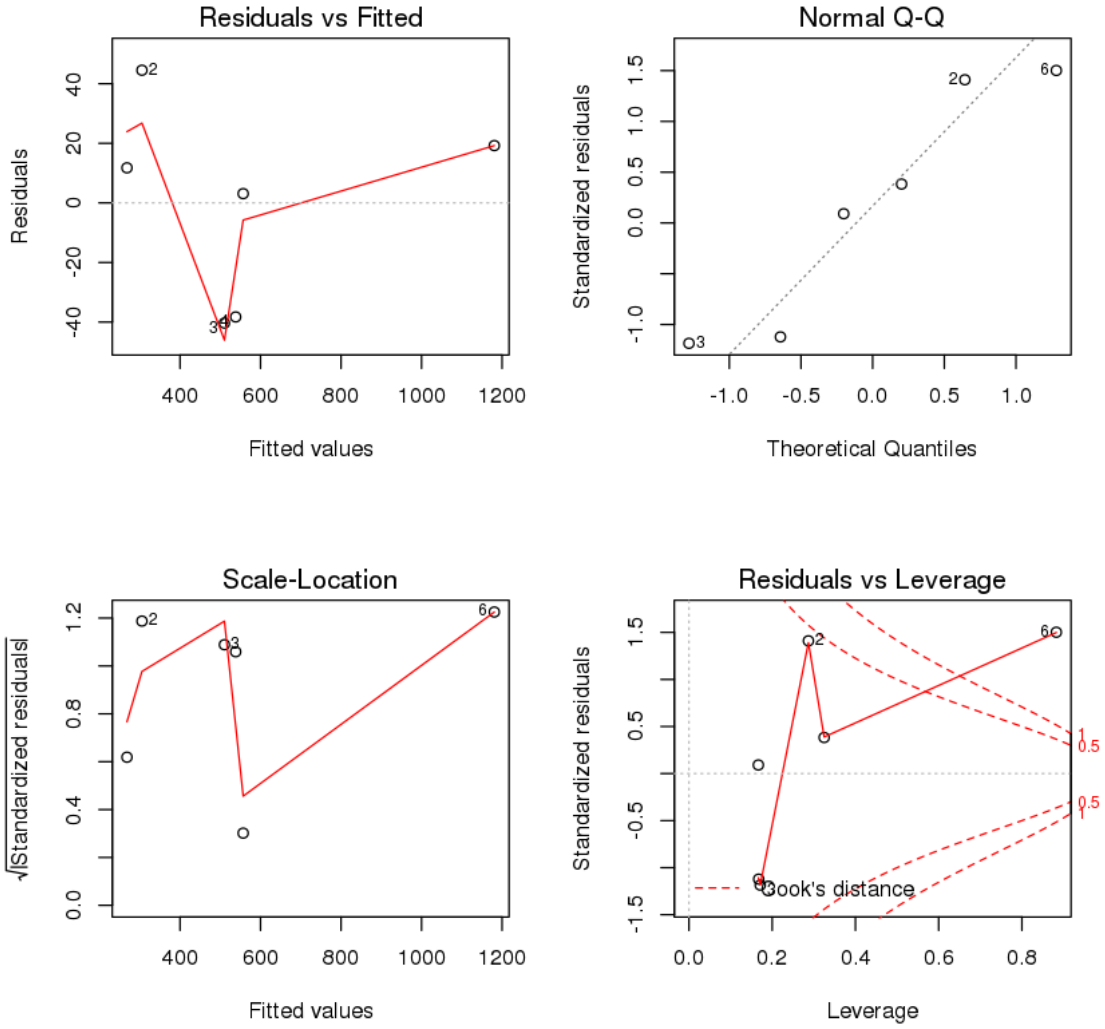
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
X	1	538208.571	538208.571	385.024	3.97826e-05
Residuals	4	5591.429	1397.857	NA	NA

1.7 Analyse des résidus

L'analyse des résidus consiste en partie à observer le graphe quantile-quantile de la régression ainsi que le tracé des résidus en fonction des valeurs prédites. Pour cela, on utilise la fonction

plot(). Comme on doit observer plusieurs graphes, on peut choisir de les afficher côte à côte. On utilise pour cela la fonction **par()**.

```
In [18]: par(mfrow = c(2,2)) # divise la figure en 2 colonnes et 2 lignes
         plot(linModel) # trace les graphes utiles pour l'analyse des résidus
```



Dans le modèle de régression linéaire simple ou multiple, on fait plusieurs hypothèses:

- les résidus ont une moyenne nulle et une variance fixe σ^2. La dispersion des résidus autour de la droite horizontale est donc homogène pour toutes les valeurs de X.
- les résidus sont des variables aléatoires indépendantes
- les résidus sont issus d'une loi normale.

Pour vérifier *l'homogénéité* de la variance des résidus, on regarde le graphe des résidus en fonction des valeurs ajustées.

Les points qui possèdent un numéro sont des valeurs extrêmes qui peuvent affecter considérablement l'hypothèse de normalité et l'homogénéité de la variance. Il est souvent utile de les enlever pour satisfaire les hypothèses initiales.

Pour vérifier l'hypothèse de *normalité*, on regarde le tracé de normalité des résidus ie le Q-Q plot (tracé des quantiles des résidus en fonction des quantiles de la distribution normale). Si l'hypothèse de normalité est respectée, alors les points devraient suivre une ligne droite.

2 Régression linéaire multiple

La plupart des commandes pour faire une analyse de régression multiple sont identiques pour la régression linéaire simple et multiple. C'est notamment le cas pour l'analyse de la variance et l'analyse des résidus.

On suppose que l'on a p prédicteurs (variables dépendantes) distincts. Le modèle de régresion linéaire multiple s'écrit:

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon$

Les données utilisées pour cet exemple sont celles de l'exercice 3 du TD12.

```
In [2]: # Télécharger les données
        d <- read.csv("supp_no3.csv", header = TRUE, sep = ";",dec = ",")
```

De même que pour la régression linéaire simple, on utilise la fonction **lm()** pour faire ajuster un modèle de régression linéaire multiple. Avec une variable dépendante y et des variables indépendantes x_1, x_2, x_3, la syntaxe pour définir le modèle de régression linéaire est :

$$lm(y \sim x_1 + x_2 + x_3, data)$$

.

De même que pour la régression linéaire simple, il n'est pas toujours nécessaire de préciser le paramètre data, ça dépend de la façon dont les données sont définies.

Si il y a beaucoup de prédicateurs, les écrire tous dans le modèle de régression peut ne pas être passionant. Si par exemple on veut prendre toutes les colonnes d'un tableau sauf celle que l'on veut prédire en variable X, alors on peut utiliser la notation ci-dessous.

$$lm(Y \sim ., data = tableau - de - donnees)$$

2.1 Ajuster le modèle de régression

```
In [27]: # définir un objet qui contient le modèle de régression
         lin = lm(Y ~ D+T+P,data = d)
```

2.2 Analyse de la variance

```
In [32]: # analyse de la variance
         anova(lin)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
D	1	1332.377528	1332.377528	263.714755	2.079433e-07
T	1	1124.501228	1124.501228	222.570225	4.019049e-07
P	1	6.831664	6.831664	1.352177	2.784085e-01
Residuals	8	40.418748	5.052343	NA	NA

Le tableau d'analyse de la variance montre la partition de la somme des carrés SSR en fonction des différents prédicateurs considérés dans le modèle de régression multiple.

On peut noter $SSR(T|D) = 1124.501228\$$

2.3 Intervalles de confiance des coefficients

```
In [29]: # Intervalles de confiance pour les estimations des coefficients
         confint(lin,level = 0.95)
```

	2.5 %	97.5 %
(Intercept)	39.2626890	56.8181475
D	1.2061876	2.1294543
T	0.5715403	0.7861355
P	-0.2083007	0.6320663

2.4 Intervalle de prédiction d'une nouvelle donnée

```
In [31]: # intervalle de prédiction pour une nouvelle donnée
         predict(lin,data.frame(D = 10,T = 50,P=12),interval = "prediction",level=0.90)
```

	fit	lwr	upr
1	101.2031	96.74449	105.6617