Modèles Linéaires

TD 3 : Régression Multiple : Généralités et prédictions
Licence 3 Informatique

Guillaume Metzler
Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France
guillaume.metzler@univ-lyon2.fr

Résumé

Le but de la présente fiche est de poursuivre notre travail et étude sur la régression linéaire multiple et d"étudier les prédictions effectuées par le modèle. Plus précisément :

- on cherchera à reproduire les outputs de la fonction *summary* d'un modèle de régression multiple,
- on construira un intervalle de confiance sur les prédictions du modèle.

1 Modèle linéaire multiple

Dans cette section, on considère le modèle multiple suivant

$$\mathbf{y} = \mathbf{X}\beta + \boldsymbol{\varepsilon}$$

où $\mathbf{y} \in \mathbb{R}^n$, $\beta \in \mathbb{R}^{p+1}$, $\mathbf{X} \in^{n\times(p+1)}$ et $\varepsilon \in \mathbb{R}^n$ représentent respectivement, la variable dépendante, le vecteur des paramètres du modèle, matrice de design et le vecteur des erreurs du modèles.

On considère un jeu de données où le but est de prédire le nombre de spectateurs (*attendance*)assistant à un match de baseball en fonctions de diverses variables come la *température*, le nombre de places assises, la taille du stade,... . Le jeu de données utilisé est *attendance.csv* qui est attaché au sujet.

```r
    # Pour charger un jeu de données
    data = read.csv("../data/attendance.csv", header = TRUE)
    n = nrow(data)
    # Régression linéaire
    mymodel = lm(attendance~.,data)
    summary(mymodel)
```

```
##
## Call:
## lm(formula = attendance ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -974.31 -280.59  -36.55  315.90 1067.98
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1239.40512  327.07553   3.789 0.000202 ***
## temperature     -4.03087    7.76298  -0.519 0.604189
## promotion       47.92031   66.49461   0.721 0.471992
## weekend         32.96119   63.49093   0.519 0.604255
## seats            0.34933    0.04671   7.479  2.6e-12 ***
## size             0.34210    0.03394  10.079  < 2e-16 ***
## rateofwins      -1.80828    2.85917  -0.632 0.527844
## rateofoppwins    4.01839    2.97794   1.349 0.178802
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 434.2 on 192 degrees of freedom
## Multiple R-squared:  0.5545,Adjusted R-squared:  0.5383
## F-statistic: 34.14 on 7 and 192 DF,  p-value: < 2.2e-16
```

1. Commenter les sorties du modèles : dire si ce dernier est globalement significatif ou non. Identifier les variables significatives.
2. A l'aide de vos connaissances, reproduire les sorties du logiciel : coefficients, erreurs standards (il s'agit de l'écart-type de l'estimateur), t_{test} et la p-value. On évaluera également l'écart type des valeurs résiduelles, le coefficient de détermination (ajusté) et on effectue le test de Fisher associé à la significativité global du modèle en précisant les paramètres de degrés de liberté qui sont utilisés.
3. Evaluer le BIC du modèle.
4. Exclure toutes les variables significatives du modèle et réapprendre le modèle et évaluer à nouveau son BIC. Est-ce un meilleur modèle ?

Nous verrons plus tard qu'il ne s'agit pas de la façon optimale de rechercher le meilleur modèle que l'on peut construire à partir d'un ensemble de variables (prochain TD).

2 Intervalles de confiance sur la prédiction

Pour cet exemple, on se placera dans le cas du modèle linéaire simple :

$$\mathbf{y} = \beta_0 + \beta_1 \mathbf{x} + \boldsymbol{\varepsilon}.$$

Cela nous permettra d'avoir une représentation graphique des résultats, mais la méthodologie à appliquer reste exactement la même dans le cas de la régression multiple.

Pour cela on va considérer le même jeu de données, mais on cherchera uniquement à prédire le nombre de spectateurs (*attendance*) en fonction de la taille du stade (*size*).

```
    # Pour charger un jeu de données
    data_bis = data[,c("attendance","size")]
    # Régression linéaire
    mymodel = lm(attendance~size,data_bis)
    summary(mymodel)
```

```
##
## Call:
## lm(formula = attendance ~ size, data = data_bis)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1307.25  -291.32     3.05   340.36  1120.57
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.390e+03  1.397e+02   9.947   <2e-16 ***
## size        4.264e-01  3.555e-02  11.992   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 487.6 on 198 degrees of freedom
## Multiple R-squared:  0.4207,Adjusted R-squared:  0.4178
## F-statistic: 143.8 on 1 and 198 DF,  p-value: < 2.2e-16
```

On peut vérifier que le modèle appris est bien globalement significatif, l'étude effectuée est donc pertinente.

Intervalle de confiance sur l'espérance de la prédiction

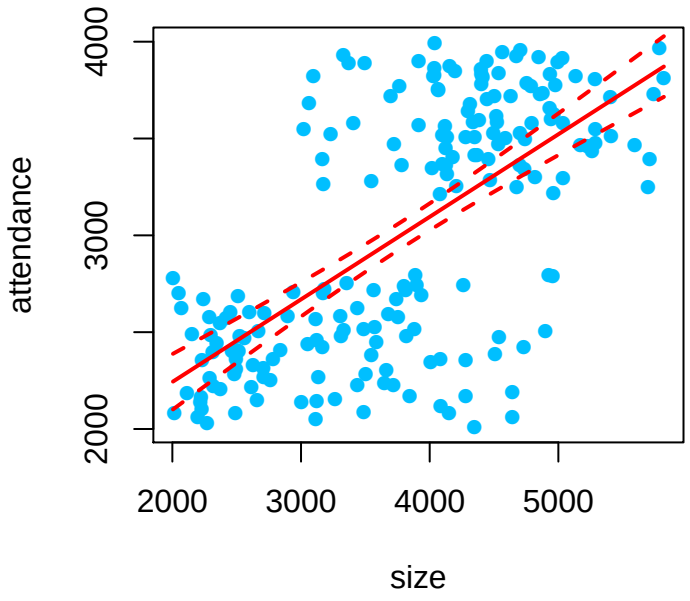
Dans un premier temps, on souhaite construire un intervalle de confiance sur l'espérance de $\mathbb{E}[Y]$ que l'on pourra représenter comme suit.

```
plot(attendance~size,data=data_bis, pch = 16, col = "deepskyblue")

# On génère échantillon qui nous servira à construire notre IC.

size=seq(min(data_bis$size),max(data_bis$size),length=100)
grille<-data.frame(size)

# Prédiction et intervalle de confiance
ICdte<-predict(mymodel,new=grille,interval="confidence",level=0.95)
matlines(grille$size,cbind(ICdte),lty=c(1,2,2), col = "red", lwd =2)
```



On peut montrer que l'intervalle de confiance de niveau $1-\alpha$ sur $\mathbb{E}[y_{\text{new}}]$

$$\hat{y}_{\text{new}} \pm t_{1-\alpha/2,n-p-1}\hat{\sigma}\sqrt{\mathbf{x}_{\text{new}}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_{\text{new}}}.$$

La démonstration est incluse dans la preuve de la section suivante.

1. Expliquer pourquoi, cette intervalle de confiance n'est pas symétrique. On pourra regarder le terme $\mathbf{x}_{\text{new}}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_{\text{new}}$ dans le cas d'un modèle simple pour comprendre le comportement de l'intervalle de confiance.

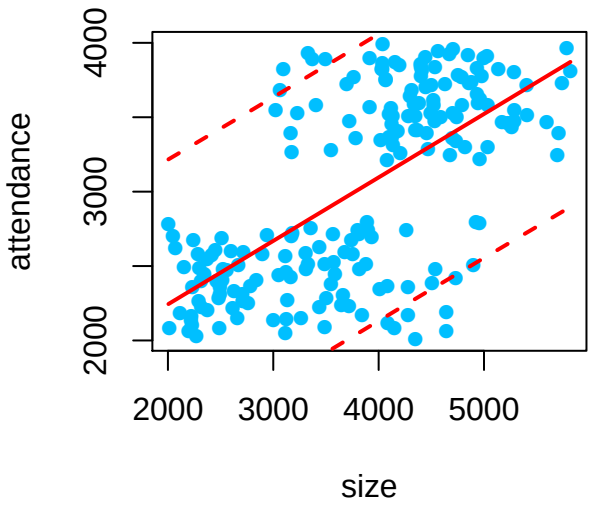
Intervalle de confiance sur la prédiction

On souhaite maintenant construire un intervalle de confiance pour une réponse individuelle Y_i.

```
plot(attendance~size,data=data_bis, pch = 16, col = "deepskyblue")

# On génère échantillon qui nous servira à construire notre IC.
size=seq(min(data_bis$size),max(data_bis$size),length=100)
grille<-data.frame(size)

# Calcul des prédictions ainsi que l'intervalle de confiance.
ICprev<-predict(mymodel,new=grille,interval="pred",level=0.95)
matlines(grille$size,cbind(ICprev),lty=c(1,2,2),col="red",lwd =2)
```



On considère une nouvelle donnée $\mathbf{x}_{\text{new}}$ et on note $\hat{y}_{\text{new}}$ sa prédiction. On sait alors que

$$\hat{y}_{\text{new}} = \hat{\beta}\mathbf{x}_{\text{new}},$$

1. Déterminer l'espérance de $\hat{y}_{\text{new}}$
2. Déterminer la variance $\text{Var}[\hat{y}_{\text{new}}]$. La démonstration est similaire à celle consistant à calculer la variance de l'estimateur $\hat{\boldsymbol{\beta}}$. On utilisera le fait que

$$\text{Var}[\hat{\boldsymbol{\beta}}] = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}.$$

3. En déduire la distribution de $y_{\text{new}} - \hat{y}_{\text{new}}$. On utilisera le fait que les données sont *i.i.d.*.
4. On admettra que la variable aléatoire $T = \dfrac{y_{\text{new}} - \hat{y}_{\text{new}}}{\sqrt{\text{Var}[y_{\text{new}} - \hat{y}_{\text{new}}]}}$ suit une loi de Student à $n - p - 1$ degrés de libertés où p représente le nombre de variables dans le modèle.
   En déduire un intervalle de confiance de niveau $1 - \alpha$ sur la prédiction d'une nouvelle observation.
5. Expliquer pourquoi cet intervalle de confiance est plus grand que le précédent.