Acquisition et analyse de données (BING-F4002): travaux pratiques

Marius Gilbert & Marc Dufrêne

Année académique 2015-2016

Séance VIII. Régression linéaire (suite)

Lors de cette séance, nous allons chercher à modéliser la distribution des captures d'Ips typographus (scolyte de l'épicéa) effectuées dans 459 pièges distribués dans une zone d'étude en région Wallonne. L'étude visait à identifier les facteurs environnementaux qui influencent les captures. Les données à traiter sont téléchargeables sur le site de l'université virtuelle, ainsi qu'à l'adresse http://www.ulb.ac.be/sciences/lubies/downloads/ips.txt Il s'agit du fichier ips.txt.

Les variables de ce jeu de données sont les suivantes : i) X_COORD : coordonnée X de l'emplacement du piège, ii) Y_COORD : coordonnée Y de l'emplacement du piège ; iii) AVIT: nombre moyen d'ips capturés par piège, iv) SUP : variable qualitative décrivant le support sur lequel était fixé le piège (1 : arbre ; 2 : pylône électrique ; 3 support en bois planté dans le sol), v) ACCQT : indicateur d'accessibilité du piège (allant de très peu accessible à très accessible) ; vi) CLCC : % de conifères dans un rayon de 500 m ; vii) CLCM : % de forêt mixte dans un rayon de 500 m ; viii) CLCF : % de forêt feuillue dans un rayon de 500 m ; ix) OUVQL : indicateur de l'ouverture du peuplement ; x) EPIQLP : indicateur du nombre d'épicéas dans un environnement proche (< 50m) ; xi) EPIQLD : indicateur du nombre d'épicéas dans un environnement éloigné (> 50 m) ; xii) MARR04 : nombre de martelages effectués en 2004 ; xiii) nombre de martelages effectués en 2005.

Instructions :

- Chargez le jeu de données: la fonction suivante va lire le contenu du fichier texte « ips.txt » et construire un dataframe avec celui-ci. Ici nous lui donnons le nom myD. La seconde ligne permet d'effacer toutes les données manquantes dans le jeu de données

```
myD = read.delim("http://lubies.ulb.ac.be/downloads/ips.txt")
myD = na.omit(myD)
```

- Construire un nouvelle variable appelée LGIT qui comprends le logarithme en base 10 (fonction `log10()`) de la variable `myD$AVIT`. A l'aide de la fonction `as.factor()`, convertissez ensuite la variable `myD$SUP` en facteur.

```
myD$LOGIT = log10(myD$AVIT + 1)
myD$SUP = as.factor(myD$SUP)
```

- Nous allons tout d'abord cartographier les sites de capture à l'aide des coordonnées. La seconde ligne de commande représente des points avec une taille proportionnelle à la variable LOGIT :

```
plot(Y_COORD ~ X_COORD, myD, pch = 16, col = rgb(0.5,0.5,1,0.3))
plot(Y_COORD ~ X_COORD, myD, pch = 16, col = rgb(0.5,0.5,1,0.3)
     , cex = myD$LOGIT)
```

- A l'aide de la fonction lm(), construisez un modèle de régression multiple qui prédit le nombre d'ips (variable AVIT) en fonction de toutes les variable prédictives, à l'exception de SUP et des coordonnées X_COORD et Y_COORD. Appelez ce modèle myFullReg et affichez le résumé de ce modèle. Ce modèle est-il satisfaisant ? Pourquoi ?
- Les fonctions suivantes vous permettent d'afficher le graphique quantile-quantile des résidus, et le graphique des résidus. Les résidus se distribuent-il de manière normale ? Sont-ils homogènes ?

plot(myReg, which = 1) plot(myReg, which = 2)

- Construisez à présent le même modèle, mais en modélisant cette fois le logarithme des captures (LOGIT) et vérifiez à l'aide des deux graphiques illustrés au point précédent si les conditions d'application ont été améliorées.
- Simplifiez le modèle en enlevant successivement, pas à pas, les variables les moins significatives (variables qui ont la valeur de t la plus faible en valeur absolue, et la valeur de p la plus grande), jusqu'à aboutir à un modèle dans lequel toutes les variables sont significatives. A quel modèle aboutissez-vous ? Pensez-vous qu'il s'agisse d'un bon modèle ?
- La fonction step() permet d'automatiser la procédure de sélection des variables. Reprenez le modèle de régression myFullReg du point 4, et utilisez ensuite la fonction suivante. L'algorithme a-t-il abouti au même modèle que vous ?

```
myStepReg = step(myFullReg, direction = "both")
summary(myStepReg)
```

- Ajoutez à présent la variable support (SUP) à votre modèle trouvé au point 7, et appelez ce modèle myFinalReg. Comme vous l'avez vu au point 1, il s'agit d'une variable qualitative à trois niveaux. En examinant le summary() de votre modèle, comment cette variable a-t-elle été incorporée ?
- La ligne de commande suivante vous permet d'afficher une analyse de variance sur votre régression. Quelle différence pouvez-vous observer par rapport au summary() de votre modèle.

```
summary(aov(myFinalReg))
```

- Nous allons à présent installer une librairie supplémentaire, appellée `ncf`, la charger, puis calculer le corrélogramme des résidus de cette régression. Selon-vous, qu'est-ce qui est représenté ici ?

```
install.packages("ncf")
library("ncf")
myD$res = residuals(myFinalReg)
myCorr <- correlog(myD$X_COORD, myD$Y_COORD, myD$res,na.rm=T, increment=500,
                   resamp=0, latlon = F)
plot(myCorr$mean.of.class[1:30],myCorr$correlation[1:30], ylim = c(-0.1,1)
     , type = "b", xlab = "distance (m)", ylab = "correlation")
```