G. Hunault

Angers, janvier 2020

CMI M1 BV

Modélisation statistique

Modélisation par régressions

1. Préparation et gestion des données

On veut ici utiliser des données HSP qui sont similaires aux données LEA mais pour des protéines de nature différente : ce sont des protéines liées aux chocs thermiques alors que les protéines LEA sont liées à l'embryogenèse tardive.

Les données sont à l'adresse

`http://forge.info.univ-angers.fr/~gh/Cmi/hsp.dar`

Vous fournirez, pour les réponses aux questions suivantes, le code R que vous avez utilisé avant de détailler vos réponses. Le terme "commande" signifie ici une ou plusieurs instructions R.

1. quelle commande R permet de lire le fichier de données pour en faire un *data.frame* nommé hsp ?

2. combien y a-t-il de lignes et de colonnes dans ce *data.frame* ?

3. comment y a-t-il de lignes dans ce *data.frame* avec des protéines de moins de 315 acides aminés ?

4. comment créer, dans ce *data.frame*, à l'aide la fonction ifelse() une variable gt (pour "Grande taille") qui vaut 0 si la longueur de la protéine est strictement inférieure à 150 acides aminés et 1 sinon ?

5. quelle est la distribution de cette variable gt ?

2. Régression linéaire simple

Effectuer, pour l'ensemble des données HSP une régression linéaire simple de la variable Foldindex en fonction de la variable Netcharge.

Discuter la validité et la qualité de la régression.

3. Régressions linéaires simples

Y aurait-il une meilleure régression linéaire simple pour modéliser Foldindex sur l'ensemble des variables ?

4. Régression linéaire multiple

Réaliser la régression linéaire multiple de la variable Netcharge en fonction de toutes les autres variables.

Discuter la validité et la qualité de la régression.

Y aurait-il un meilleur sous-ensemble de variables pour modéliser Netcharge ?

5. Discussion

Vous essaierez de construire une réponse structurée et bien rédigée à la question suivante, si possible à l'aide d'exemples concrets.

> *Est-il difficile d'effectuer des analyses par régression avec le logiciel R quand on ne maitrise pas les formules mathématiques de régression ?*

Il est conseillé d'utiliser au moins trois mots de trois syllabes ou plus pour « transmettre un contenu rédactionnel fort ».

Une dizaine de lignes parait être une rédaction minimale.

ESQUISSE DE SOLUTION

1. Préparation et gestion des données

```
## préparation et gestion des données

# lecture des fonctions gh

if (!exists("cats")) {
   void <- capture.output( source("http://forge.info.univ-angers.fr/~gh/statgh.r",
                                  encoding="latin1") )
} # fin si

# 1. lecture des données

# hsp <- lit.dar("http://forge.info.univ-angers.fr/~gh/Cmi/hsp.dar")

# autre solution

hsp <- read.table("http://forge.info.univ-angers.fr/~gh/Cmi/hsp.dar",
                  head=TRUE,row.names=1)

# 2. nombre de lignes et de colonnes (1952 x 7)

print(dim(hsp))

# 3. nombre de lignes avec moins de 315 acides aminés (1929)

flt <- hsp$Length<315
cat("il y a",sum(flt),"protéines avec moins de 315 acides aminés\n")

# 4. variable gt (0 si moins de 150 aa, 1 si 150 aa ou plus)

hsp$gt <- ifelse(hsp$Length<150,0,1)

# 5. distribution de la variable gt, solution minimale
#    on trouve 887 petites protéines, 1065 grandes (soit 55 %)

print(table(hsp$gt))

# solution attendue

decritQL("variable gt, données HSP",hsp$gt,"petite_taille grande_taille")
```

2. Régression linéaire simple

```
# régression linéaire simple de Foldindex en fonction de Netcharge
# dans les données hsp de l'examen

# lecture des données

hsp <- read.table("http://forge.info.univ-angers.fr/~gh/Cmi/hsp.dar",head=TRUE,row.names=1)

# affichage pour vérification

print(summary(cbind(hsp$Netcharge,hsp$Foldindex)))

# calculs et affichages pour la régression

mls <- lm( hsp$Foldindex ~ hsp$Netcharge)
print( mls )
print( anova(mls) )
print( summary(mls) )

# Fisher p-value 3.436e-15, régression valide
# R2a 0.03, régression de piètre qualité
```

3. Régressions linéaires simples

```
# meilleure régression linéaire simple pour modéliser Foldindex
# dans les données hsp de l'examen

# lecture des données

hsp <- read.table("http://forge.info.univ-angers.fr/~gh/Cmi/hsp.dar",head=TRUE,row.names=1)

# on charge la fonction vue en cours

source("http://forge.info.univ-angers.fr/~gh/Cmi/mrls.r",encoding="latin1")

# lecture des fonctions gh

if (!exists("cats")) {
   void <- capture.output( source("http://forge.info.univ-angers.fr/~gh/statgh.r",
                                  encoding="latin1") )
} # fin si

# on met Foldindex en premier (c'est la colonne 4 et il y a 7 colonnes QT)

hspF <- hsp[ , c( 4, 1:3, 5:7 ) ]
```

```
# on applique la fonction mrls()

mrls( hspF )

# lm( Foldindex ~ Gravy) est sans doute la meilleure régression linéaire simple
```

4. Régression linéaire multiple

```
# 1. régression linéaire multiple de Netcharge
#    en fonction de toutes les autres variables
#    dans les données hsp de l'examen

# lecture des données

hsp <- read.table("http://forge.info.univ-angers.fr/~gh/Cmi/hsp.dar",head=TRUE,row.names=1)

# calculs et affichages pour la régression

mlm <- lm( Netcharge ~ . , data=hsp)
print( mlm )
print( anova(mlm) )
print( summary(mlm) )

# Fisher p-value < 2.2e-16, régression valide
# R2a 0.8203, régression de bonne qualité


# 2. recherche du meilleur sous-ensemble pour la régression linéaire multiple de Netcharge

# on charge le package olsrr vu en cours

library(olsrr)

# on applique la fonction ols_step_best_subset()

print(ols_step_best_subset(mlm))

# le meilleur sous-ensemble est Length pI Foldindex Gravy
```