TP Modèle linéaire - Feuille 1

STID2 - IUT Vannes - 2021/2022
A. Cuzol / A. Poterie

Remarques :
— Les données sont disponibles sur la plateforme pédagogique "Modèle linéaire AC" ;
— Les TP seront réalisés avec R (privilégiez l'utilisation de Rstudio) ;
— Vous devez :
  • Indiquer à R quel est votre répertoire de travail en utilisant la commande `"setwd"` ou avec le menu de Rstudio : *Session -> Set Working Directory*.
  • Créer un script "TP1.R" dans ce répertoire, dans lequel vous écrirez vos commandes.
  • Sauvegarder régulièrement ce script.
— Pensez à commenter vos codes pour pouvoir les réutiliser par la suite (symbole "#" en début de chaque ligne à commenter).
— Gardez une trace des réponses aux questions et interprétations : sur une feuille, en commentaire dans votre script, sur un document Word, dans un fichier R markdown, etc.

Exercice 1

Copier le fichier "Ski.rda" dans votre répertoire personnel.

On a observé, pour 97 stations de ski françaises :
— le prix du forfait semaine ("Forfait") ;
— l'altitude de la station village ("AltVillage") ;
— le nombre de remontées ("NbRemontees") ;
— l'altitude du sommet des pistes ("AltPistes") ;
— le nombre de pistes ("NbPistes").

1. Importer les données :

```
load("Ski.rda")
```

2. Regarder brièvement le contenu du jeu de données :

```
dim(Ski)
head(Ski)
summary(Ski)
```

Combien de variables contient le jeu de données ? Combien d'individus ? Que représente un individu dans ce jeu de données ?

3. Les données contiennent-elles des valeurs manquantes ? Des valeurs aberrantes ?

4. Etudier graphiquement le lien entre le prix du forfait et le nombre de pistes :

```
plot(Ski$NbPistes,Ski$Forfait,xlab="Nombre de pistes",ylab="Prix du forfait")
```

5. Calculer le coefficient de corrélation entre ces deux variables :
```
coefcor = cor(Ski$Forfait,Ski$NbPistes)
```

6. Mettre en place un test de corrélation pour tester la significativité de la corrélation entre les deux variables (voir cours "Estimation et tests").

7. Rappeler l'écriture du modèle théorique de régression simple expliquant le prix du forfait par le nombre de pistes. Quels sont les paramètres à estimer dans ce modèle ?

8. Réaliser l'estimation grâce à la fonction `lm` (*linear model*) :
```
reg_ski = lm (Forfait ~ NbPistes, data=Ski)
summary(reg_ski)
```
   — Que contient `reg_ski$coefficients` ?
   — Que contient `reg_ski$fitted.values` ?
   — Que contient `reg_ski$residuals` ?

9. Donner l'écriture du modèle ajusté.

10. Quelle est la valeur du coefficient de détermination R^2 du modèle ? Vérifier le lien avec le coefficient de corrélation calculé précédemment.

11. Superposer le modèle ajusté aux données :
```
plot(Ski$NbPistes,Ski$Forfait,xlab="Nombre de pistes",ylab="Prix du forfait",
main="Modèle ajusté")
abline(reg_ski,col="red")
```

12. On souhaite tester la significativité des coefficients du modèle.
   (a) Quelles sont les hypothèses de test ?
   (b) Quelle est la statistique de test utilisée ?
   (c) Quelle est sa loi sous H_0 ?
   (d) Quelle valeur prend la statistique de test pour le test sur β_0 (ordonnée à l'origine) ? Pour le test sur β_1 (pente) ?
   (e) Calculer le quantile de niveau $1 - \alpha/2$ de la loi de la statistique de test sous H_0, pour $\alpha = 0.05$, en utilisant :
   `qt(0.975,n-2)`
   (où `n` est le nombre d'individus).
   (f) Conclure le test pour chaque coefficient en comparant la statistique de test au quantile calculé.

(g) Retrouver la conclusion des tests directement avec les p-values données dans la sortie de la fonction `lm`.

(h) Finalement, est-ce que le nombre de pistes a une influence sur le prix du forfait ? Quel test nous permet de répondre à cette question ?

13. On veut prédire le prix du forfait pour une station ayant 30 pistes, et pour une station ayant 200 pistes.

(a) Calculer les valeurs prédites à partir de la formule de prévision.

(b) Faire le même calcul avec la fonction `predict` de R :

```
newdata = data.frame(NbPistes=c(30,200))
predict(reg,newdata,se.fit=TRUE)
```

(c) Commenter les résultats.

Exercice 2

On reprend les données sur les stations de ski.

1. Afficher la matrice des nuages de points entre toutes les variables :

```
plot(Ski[,-1])
```

2. Afficher la matrice des corrélations entre toutes les variables. Commenter.

3. Combien de modèles de régressions (simples ou multiples) peut-on proposer pour expliquer le forfait semaine à partir des variables disponibles dans le jeu de données ? Lister tous ces modèles.

4. Ajuster le modèle de régression multiple expliquant le prix du forfait par les variables "NbPistes" et "AltVillage". Stocker ce modèle dans `reg_ski_mult_1`

5. Représenter le nuage de points en 3D des observations et le modèle ajusté :

```
library("scatterplot3d")
s3d = scatterplot3d(Ski$NbPistes,Ski$AltVillage, Ski$Forfait, pch=16,angle=55,
                    xlab="NbPistes",ylab="AltVillage",zlab="Forfait")
s3d$plane3d(reg_ski_mult_1)
```

6. Ajuster le modèle de régression multiple expliquant le prix du forfait par les variables "NbPistes" et "AltPistes". Superposer le modèle ajusté sur les observations.

7. Comparer ces deux modèles. Quel modèle préférez-vous et pourquoi ?

8. Serait-il judicieux de proposer un modèle de régression multiple intégrant à la fois "NbPistes" et "NbRemontees" dans le groupe des variables explicatives ?

TP Modèle linéaire - Feuille 2

STID2 - IUT Vannes - 2021/2022

A. Cuzol / A. Poterie

Remarques :
— Les données sont disponibles sur la plateforme pédagogique "Modèle linéaire AC" ;
— Les TP seront réalisés avec R (privilégiez l'utilisation de Rstudio) ;
— Vous devez :
  • Indiquer à R quel est votre répertoire de travail en utilisant la commande `"setwd"` ou avec le menu de Rstudio : *Session -> Set Working Directory*.
  • Créer un script "TP2.R" dans ce répertoire, dans lequel vous écrirez vos commandes.
  • Sauvegarder régulièrement ce script.
— Pensez à commenter vos codes pour pouvoir les réutiliser par la suite (symbole "#" en début de chaque ligne à commenter).
— Gardez une trace des réponses aux questions et interprétations : sur une feuille, en commentaire dans votre script, sur un document Word, dans un fichier R markdown, etc.

Exercice 1

On reprend les données du fichier "Ski.rda" dans lequel on observe, pour 97 stations de ski françaises :
— le prix du forfait semaine ("Forfait") ;
— l'altitude de la station village ("AltVillage") ;
— le nombre de remontées ("NbRemontees") ;
— l'altitude du sommet des pistes ("AltPistes") ;
— le nombre de pistes ("NbPistes").

1. Importer les données :

```
load("Ski.rda")
```

2. Ajuster le modèle de régression simple expliquant le prix du forfait semaine par le nombre de pistes dans la station. On appellera ce modèle "reg".

3. Vérification de l'hypothèse de lien linéaire : tracer le graphique des résidus en fonction des valeurs ajustées et commenter le résultat obtenu.

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

4. On souhaite vérifier l'hypothèse gaussienne sur les erreurs du modèle. A quoi sert cette hypothèse ? Afficher l'histogramme des résidus. Afficher le "qq-plot" ou "droite de Henry" des résidus (utiliser la fonction `qqnorm`). Commenter les graphiques. Les résidus semblent-ils suivre une loi gaussienne ?

5. Que fait le code R ci-dessous ?

```
hist(reg$residuals,freq=FALSE)
u<-seq(min(reg$residuals),max(reg$residuals),0.1)
lines(u,dnorm(u,mean(reg$residuals),sd(reg$residuals)),col="blue")
```

6. Vérifier l'hypothèse d'homoscédasticité des erreurs en traçant le nuage de points des résidus studentisés en fonction des valeurs ajustées :

```
plot(reg$fitted.values, rstudent(reg))
```

7. Repérer les individus atypiques ou influents (ou les deux) à l'aide des distances de Cook :

```
plot(cooks.distance(reg),type="h",ylab="Distances de Cook",ylim=c(0,0.8))
seuil2=qf(0.1,p,n-p) #Seuil raisonnable
seuil3=qf(0.5,p,n-p) #Seuil préoccupant
abline(h=seuil2,col="blue")
abline(h=seuil3,col="red")
```

Exercice 2

On considère des données qui viennent du Groupe d'Etude et de Réflexion Inter-régional (GERI) : on dispose d'observations sur 95 départements français pour l'année 1990. Ces observations concernent quatre grands thèmes : la démographie, l'emploi, la fiscalité directe locale et la criminalité. Voici la liste des variables disponibles :

- CRIM : taux de criminalité (en $^0/_{00}$)
- TXCR : taux de croissance de la population sur la période 1982-1990
- ETRA : part des étrangers dans la population totale
- URBR : taux d'urbanisation
- JEUN : part des 0-19 ans dans la population totale
- AGE : part des plus de 65 ans dans la population totale
- CHOM : taux de chômage
- Parts de chaque profession et catégorie socio-professionnelle dans la population active du département :
  - AGRI : agriculteurs
  - ARTI : artisans
  - CADR : cadres supérieurs
  - EMPL : employés
  - OUVR : ouvriers
  - PROF : professions intermédiaires
- FISC : produit, en francs constants 1990 et par habitant, des quatre taxes directes locales (professionnelle, habitation, foncier bâti, foncier non bâti)
- FE90 : taux de fécondité (en $^0/_{00}$)

L'objectif est le suivant : étudier le taux de criminalité en fonction des autres variables.

1. Importer les données dans "dep". Enlever les variables qui décrivent les départements :

```
dep=dep[,3:17]
```

2. On souhaite sélectionner le meilleur modèle par une recherche de type "backward". Le principe est le suivant :
   — On part du modèle complet.
   — On vérifie si toutes les variables sont significatives (la "significativité" d'une variable est mesurée par la p-value du test de Student associé). Si c'est le cas on conserve le modèle complet.
   — Sinon, on enlève la variable la moins significative et on recommence avec le modèle réduit jusqu'à ce que tous les tests soient significatifs.

   Le début du code R est le suivant :

```
fit=lm(CRIM~.,data=dep)  # modèle complet
summary(fit)
dep2=dep[,colnames(dep) !="AGE"]   # on enlève la variable "AGE"
fit=lm(CRIM~.,data=dep2)
summary(fit)
```

   Compléter la procédure de sélection. Quel modèle obtient-on finalement ? Donner l'écriture du modèle ajusté.

3. On souhaite maintenant sélectionner le meilleur modèle par une recherche exhaustive. Le principe est le suivant :
   — On se fixe un critère de choix (R^2 ajusté, AIC, BIC,...).
   — On cherche le meilleur modèle selon ce critère, parmi tous les sous-modèles possibles.

   Utiliser le code suivant pour mettre en place la recherche exhaustive pour les critères du R^2 ajusté et du BIC :

```
library(leaps)
choix =regsubsets(CRIM~., nvmax=14, method="exhaustive",data=dep)
par(mfrow=c(1,2))
plot(choix,scale="adjr2")
plot(choix, scale="bic")
```

   Quel modèle conserve-t-on finalement pour chaque critère ?

TP Modèle linéaire - Feuille 3

STID2 - IUT Vannes - 2021/2022
A. Cuzol / A. Poterie

Remarques :
— Les données sont disponibles sur la plateforme pédagogique "Modèle linéaire AC" ;
— Les TP seront réalisés avec R (privilégiez l'utilisation de Rstudio) ;
— Vous devez :
  • Indiquer à R quel est votre répertoire de travail en utilisant la commande `"setwd"` ou avec le menu de Rstudio : *Session -> Set Working Directory.*
  • Créer un script "TP3.R" dans ce répertoire, dans lequel vous écrirez vos commandes.
  • Sauvegarder régulièrement ce script.
— Pensez à commenter vos codes pour pouvoir les réutiliser par la suite (symbole "#" en début de chaque ligne à commenter).
— Gardez une trace des réponses aux questions et interprétations : sur une feuille, en commentaire dans votre script, sur un document Word, dans un fichier R markdown, etc.

Exercice 1

On a observé les hauteurs d'arbres pris au hasard dans deux types de forêts. Les mesures sont données dans le fichier foret.txt. On se pose la question : y a-t-il une différence significative entre les moyennes des hauteurs d'arbres dans les deux forêts ?

1. Importer les données dans l'objet foret :

```
foret <- read.table("foret.txt",header=T,sep="")
```

2. Faire une représentation graphique de la hauteur des arbres par rapport au type de forêt :

```
boxplot(hauteur~typeForet, data=foret)
```

Remarque : on peut modifier les options du graphique :

```
boxplot(hauteur~typeForet, data=foret, boxwex=0.3, col="lightblue",
pch=20, xlab="type de foret", ylab="hauteur des arbres (m)")
```

3. Calculer les moyennes des hauteurs des arbres dans chacune des deux forêts :

```
moy = by(foret$hauteur, INDICES=foret$typeForet, FUN=mean)
```

4. Afficher ces moyennes sur le graphique précédent :

```
points(x=1:2, y=moy, pch=20, col="red", cex=2)
```

5. Pour vérifier si la hauteur moyenne des arbres dans les deux forêts est différente, on peut utiliser un test de comparaison de moyennes (test de Student) :

```
t.test(formula=hauteur~typeForet,  data=foret, var.equal=TRUE)
```

   (a) Rappeler les hypothèses du test, la statistique de test, sa loi sous H_0.

   (b) Au seuil 5%, peut-on conclure à une différence des moyennes de hauteur d'arbres entre les deux forêts ?

6. Pour répondre à la même question, on peut faire une analyse de la variance à un facteur (on étudie ici le lien entre une variable quantitative et une variable qualitative à deux modalités).

   (a) Rappeler les hypothèses du test permettant de tester l'influence de la variable explicative.

   (b) Quelle est la statistique de test ?

   (c) Quelle est sa loi sous H_0 ?

   (d) Mettre en place le test :

```
mod=lm(hauteur ~ typeForet, data=foret)
anova(mod)
```

   (e) Conclure le test. Vérifier que l'on retrouve bien la même conclusion que le test de Student.

Exercice 2

Nous souhaitons comparer trois traitements, notés A, B et C contre l'asthme : le traitement B est un nouveau traitement, que nous souhaitons mettre en compétition avec les traitements classiques A et C. Nous répartissons par tirage au sort les patients venant consulter dans un centre de soin, et nous leur affectons l'un des trois traitements. Nous mesurons sur chaque patient la durée, en jours, séparant de la prochaine crise d'asthme. Les mesures sont reportées dans le tableau ci-dessous :

Traitement A	Traitement B	Traitement C
26 ; 27 ; 35 ; 36	29 ; 42 ; 44 ; 44	26 ; 26 ; 30 ; 30
38 ; 38 ; 41 ; 42	45 ; 48 ; 48 ; 52	33 ; 36 ; 38 ; 38
45 ; 50 ; 65	56 ; 56 ; 58 ; 58	39 ; 46 ; 47 ; 51
	60 ; 61 ; 63 ; 63	51 ; 56 ; 75
	69	

1. Saisir les données sous forme d'un fichier asthme.txt. On notera "Duree" la variable indiquant la durée jusqu'à la prochaine crise d'asthme, et "Traitement" le type de traitement.

2. Importer les données dans l'objet "asthme".

3. Représenter graphiquement les données à l'aide de boîtes de dispersion. Ajouter sur le graphique les points correspondant aux durées moyennes pour chaque traitement.

4. Les traitements ont-ils tous la même efficacité ? Répondre par le test d'influence de la variable "Traitement" après avoir écrit les hypothèses de test.

5. A l'aide du test de Tukey, complétez votre conclusion :

```
TukeyHSD(x=aov(asthme$Duree ~asthme$Traitement))
```

TP Modèle linéaire - Feuille 4

STID2 - IUT Vannes - 2021/2022

A. Cuzol / A. Poterie

Exercice 1

Le fichier "jellyfish.txt" contient la largeur (Breadth) et la longueur (Length) en millimètres de 46 méduses réparties sur 2 sites en Australie (Dangar Island et Salamander Bay).

1. Importer les données dans l'objet meduse :

   ```
   meduse=read.table("jellyfish.txt", header=T, sep=" ")
   ```

2. Vérifier que la variable "Site" est bien définie comme un facteur (utiliser `is.factor()` et `as.factor()`).
3. Représenter à l'aide de boxplots la largeur des méduses en fonction de leur site. Semble-t-il y avoir un effet site sur la largeur ? Confirmer l'analyse en effectuant une analyse de la variance.
4. Etudier de même l'effet du site sur la longueur des méduses.
5. Représenter le nuage de points entre la largeur et la longueur en utilisant des couleurs différentes selon le site :

   ```
   plot(meduse[,"Length"],meduse[,"Breadth"],col=as.numeric(meduse[,"Site"],
   xlab="Largeur",ylab="Longueur")
   ```

6. Faire une régression simple pour expliquer la largeur des méduses en fonction de leur longueur.
7. On désire modéliser la largeur des méduses en fonction de leur longueur et de leur site d'appartenance.

   (a) Commencer par estimer le modèle complet (un modèle de régression pour chaque site) :

   ```
   mod1=lm(Breadth ~ -1+Site+Site:Length,data=meduse)
   summary(mod1)
   ```

   (b) Extraire les coefficients du modèle et superposer le modèle ajusté au nuage de points :

   ```
   coefs=mod1$coefficients
   plot(meduse[,"Length"],meduse[,"Breadth"],col=as.numeric(meduse[,"Site"],
   xlab="Largeur",ylab="Longueur")
   abline(coefs[1], coefs[3],col=1)
   abline(coefs[2], coefs[4],col=2)
   ```

   (c) Estimer le modèle avec pentes égales :

   ```
   mod2=lm(Breadth ~ -1+Site+Length,data=meduse)
   summary(mod2)
   ```

   (d) Superposer ce dernier modèle ajusté au nuage de points.

   (e) On souhaite savoir si on peut considérer que les pentes sont égales ou s'il vaut mieux conserver le modèle complet. On utilise :

```
anova(mod2,mod1)
```

Rappeler les hypothèses de test et conclure au niveau $\alpha = 5\%$.

(f) Estimer le modèle avec ordonnées à l'origine égales :

```
mod3=lm(Breadth ~ Site:Length,data=meduse)
summary(mod3)
```

(g) Superposer ce dernier modèle ajusté au nuage de points.

(h) On souhaite savoir si on peut considérer que les ordonnées à l'origine sont égales ou s'il vaut mieux conserver le modèle complet. On utilise :

```
anova(mod3,mod1)
```

Rappeler les hypothèses de test et conclure au niveau $\alpha = 5\%$.

(i) Quel modèle conserve-t-on finalement ?