TP d'Econométrie II avec le logiciel R

Problème structurel, hérétoscedasticité et non-indépendance de l'erreur

L'objectif de ce TP est de mettre en application les méthodes permettant de dépasser certains problèmes rencontrés lors d'une estimation par moindres carrés ordinaires: problème structurel, hérétoscedasticité et non-indépendance de l'erreur. On propose d'abord de diagnostiquer les problèmes, puis de trouver des améliorations.

Changement de modèle

Soit les commandes suivantes:

```
i=1:100;
Z1=i;Z2=i^2;
epsilon=5*rnorm(100,0);
Y=50-0.5*Z1+0.01*Z2+epsilon;
plot(i,Y)
reg1=lm(Y~Z1)
summary(reg1); plot(reg1)
```

Commenter les différentes commandes executées. Le text de Fisher est-il accepté? Où détecter vous qu'il y a un problème? Comment pourriez-vous le régler?

```
library(car)
reg2=lm(Y~Z1+Z2)
summary(reg2); plot(reg2)
scatterplotMatrix(~Y+Z1+Z2)
```

Sur ce dernier graphe, on peut voir les résultats des régressions linéaires entre chacune des variables, ce qui peut donner des idées pour transformer les variables.
Une méthode pour améliorer le modèle peut être de chercher une transformation de Box-Cox de la variable:

```
library(MASS)
BX=boxcox(Z1 ~Y,plotit = TRUE,lambda = seq(-3,3))
ind=which(BX$y==max(BX$y))
lambda=BX$x[ind]
lambda
```

On détermine ainsi la transformation de Box-Cox à appliquer sur la variable Z_1 pour améliorer le modèle. Notez qu'il est aussi possible de trouver la transformation de Box-Cox de Y également (on écrit alors $Y \sim Z1$ dans la commande). Faites différents essais sur le jeu de données.

Premier exercice

Sur le jeu de données Fractales2.txt, effectuer une régression des $\log N$ en fonction des $\log r$. Qu'en pensez-vous? Appliquez ensuite une transformation de Box-Cox sur $\log r$ puis sur $\log N$ sur le modèle. Est-ce mieux?

Second exercice

Sur le jeu de données PIBChomage.txt, effectuer une régression du Taux de Chomage en fonction du temps et de la croissance. Qu'en pensez-vous? Appliquez ensuite une transformation de Box-Cox pour améliorer le modèle. Est-ce satisfaisant?

Troisième exercice

Dans cet exercice on va utiliser d'autres représentations possibles en exploration des données sur un jeu de données très particulier (interne à R)... Soit les commandes:

```
library(ggplot2); library(hexbin); library("gridExtra")
attach(anscombe); names(anscombe); head(anscombe); anscombe
par(mfrow = c(2,2))
gg1 <- ggplot(anscombe, aes(x=x1, y=y1)) + geom_point() +
        geom_smooth(method="lm")
gg2 <- ggplot(anscombe, aes(x=x2, y=y2)) + geom_point() +
        geom_smooth(method="lm")
gg3 <- ggplot(anscombe, aes(x=x3, y=y3)) + geom_point() +
        geom_smooth(method="lm")
gg4 <- ggplot(anscombe, aes(x=x4, y=y4)) + geom_point() +
        geom_smooth(method="lm")
grid.arrange(gg1, gg2, gg3, gg4, ncol = 2, nrow = 2)

mod1=lm(y1~x1); mod2=lm(y2~x2); mod3=lm(y3~x3); mod4=lm(y4~x4)
summary(mod1); summary(mod2); summary(mod3); summary(mod4)
plot(mod1);plot(mod2); plot(mod3); plot(mod4)

MASS::studres(mod1); hatvalues(mod4)
```

A chaque étape décrire ce qui a été fait. Qu'y a-t-il de remarquable avec ce jeu de données? Que pourriez-vous faire pour améliorer le traitement?

Détection de rupture

Soit la nouvelle simulation:

```
epsilon=5*rnorm(100,0);
i=1:100; Z1=i;Z2=i^2; Y=0;
for (j in c(1:60))
  Y[j]=50-0.5*j+0.01*j^2+epsilon[j];
for (j in c(61:100))
  Y[j]=60-0.2*j+epsilon[j];
ts.plot(Y); plot(Y)

p=0; Q=0;
for (k in c(3:97))
{p=p+1;
reg1=lm(Y[1:k]~Z1[1:k]+Z2[1:k])
reg2=lm(Y[(k+1):100]~Z1[(k+1):100]+Z2[(k+1):100])
Q[p]=sum(reg1$res^2)+sum(reg2$res^2)}
i0=(Q==min(Q))
k=c(3:97); k[i0]
```

Ceci permet de trouver l'instant de rupture le plus probable. On peut ensuite effectuer un test de Chow:

```
reg1=lm(Y[1:k[i0]]~Z1[1:k[i0]]+Z2[1:k[i0]])
reg2=lm(Y[(k[i0]+1):100]~Z1[(k[i0]+1):100]+Z2[(k[i0]+1):100])
reg=lm(Y~Z1+Z2)
```

```
Den=sum(reg1$res^2)+sum(reg2$res^2)
Num=sum(reg$res^2)-Den
Chow=(100-6)/3*Num/Den
pval=1-pchisq(3*Chow,df=3)
```

Quelle est votre conclusion? Répéter l'expérience et donner un histogramme de l'instant de rupture estimé.

Troisième exercice

Sur le jeu de données PIB1950.txt, détecter la rupture et appliquer un test de Chow. Conclusion?

Présence d'hétéroscedasticité

Soit les commandes suivantes

```
Y=10-0.2*Z1+0.01*Z2+Z2*epsilon;
plot(i,Y)
reg1=lm(Y~Z1+Z2)
summary(reg1);
plot(reg1)
```

Sur quel graphique détectez-vous un problème? On propose de modifier la régression de la manière suivante:

```
YY=1/Y
reg2=lm(YY~Z1+Z2)
summary(reg2); plot(reg2)
```

Expliquer ces commandes. Est-ce satisfaisant?

```
library(MASS)
boxcox(Y~Z1+Z2,lambda = seq(-2, 2,0.05))
```

Qu'en pensez-vous?

On suppose pour l'instant que l'on connait la matrice de covariance des erreurs (ici $\Sigma = (\sqrt{i}\,\delta_{ij})_{1\leq i,j\leq n}$).
On peut maintenant utiliser un estimateur par moindres carrés pondérés.
En utilisant la formule de l'estimateur par moindres carrés pondérés, estimer les paramètres du modèle en connaissant donc la matrice de pondération. Résultat? A-t-on amélioré le R^2?

L'hypothèse d'une connaissance a priori de l'hétéroscédasticité et donc de la matrice de covariance est en général impossible en pratique. On va utiliser une méthode qui va automatiquement chercher les poids comme une loi de puissance de $\widehat{Y}_i$:

```
library(nlme)
reg3= gls(Y ~Z1+Z2, weights = varPower())
summary(reg3); plot(reg3)
```

Commenter les résultats obtenus.

Non-indépendance des erreurs

On va simuler cette fois des erreurs non indépendantes:

```
epsilon=rnorm(102,0);
u=epsilon[3:102]-2*epsilon[2:101]-4*epsilon[1:100]
acf(u)
```

Quel type d'erreur a-t-on? Quelle est son autocovariance théorique? La commande `acf` trace les autocorrélations empiriques. Conclusion?
On peut alors appliquer les 2 types de tests vu en cours, test de runs et test du portemanteau, sur cette erreur:

```
library(lawstat)
runs.test(u)
Box.test(u,lag = 5)
Box.test(u, lag = 5, type="Ljung")
```

On peut maintenant utiliser ces tests dans les problèmes de régression précédents:

```
Y=5+3*Z1-0.02*Z2+20*u;
reg4=lm(Y~Z1+Z2)
summary(reg4); plot(reg4)
runs.test(reg4$res)
Box.test(reg4$res, lag = 5, type="Ljung")
```

Conclusion?
Pour tenir compte de cette corrélation entre les erreurs, on peut à nouveau utiliser un estimateur par moindres carrés généralisés adapté à la situation:

```
reg5= gls(Y ~Z1+Z2, correlation = corARMA(p=0,q=2))
summary(reg5); plot(reg5)
```

Conclusion?
On peut aussi essayer l'estimation des coefficients par une méthode en 2 temps. On écrira ainsi:

```
a0=reg4$coeff[1],a1=reg4$coeff[2],a2=reg4$coeff[3]
res4=reg4$res
rho=acf(res4)
correl=0*c(1:100)
correl[1]=1
correl[2]=rho$acf[2]
correl[3]=rho$acf[3]
Sigma=toeplitz(correl)
theta=0
Z=cbind(rep(1,100),Z1,Z2)
theta=solve(t(Z)%*%solve(Sigma)%*%Z)%*%t(Z)%*%Y
b0=theta[1], b1=theta[2],b2=theta[3]
```

Expliquer tout ce qui est fait dans les démarches du programme (en particulier comment est estimée la matrice de corrélation de u). On cherchera donc à comparer les risques quadratiques de l'estimateur par MCO et de l'estimateur par MCG. Ecrire un programme permettant d'effectuer une telle comparaison (on répétera 100 fois les commandes précédentes et on calculera les risques quadratiques estimés de chacun des estimateurs). Quel résultat obtenez vous pour $n = 100$? Essayer également avec des trajectoires de longueurs 1000.

On va enfin travailler sur les MCG appliqués au cas de l'hétéroscédasticité. On simule ainsi un autre bruit:

```
epsilon=rnorm(100,0);
u[1:50]=10*epsilon[1:50]
u[51:100]=30*epsilon[51:100]
acf(u)
```

Expliquer ce qui a été ainsi fait.
On re-simule l'échantillon:

```
Y=5+3*Z1-0.02*Z2+u;
reg5=lm(Y~Z1+Z2)
a0=reg5$coeff[1]
a1=reg5$coeff[2]
a2=reg5$coeff[3]
Sigma=diag(reg5$res^2)
theta=0
theta=solve(t(Z)%*%solve(Sigma)%*%Z)%*%t(Z)%*%Y
b0=theta[1]
b1=theta[2]
b2=theta[3]
```

Qu'en pensez-vous? Augmenter la taille de l'échantillon pour étudier l'effet de l'asymptotique.