Contents

Pràctica 2

Model lineal: Estimar i verificar hipòtesis

Seguirem amb les dades de preu i consum.

```
library(car);library(lmtest);library(lawstat)
library(nortest);library(doBy);library(foreign)
#
require(foreign)
data<-read.spss("preuconsum.sav",to.data.frame=T)
data<-na.omit(data[,1:3])      # eliminem els 3 casos NA, només volem 3 columnes
dim(data); head(data)
names(data)<-c("id","y","x")  # canviats els noms a minúscules, per comoditat
```

En primer lloc obtindrem les estimacions dels paràmetres del model a partir de les fórmules. Seguidament, amb la funció `lm()` obtindrem les estimacions i la resta d'indicadors del model lineal (simple, en aquest cas) i analitzarem gràficament les hipòtesis del model lineal: *centrament, homocedasticitat, incorrelació* i *normalitat.* [*Nota:* La incorrelació només cal comprovar-la si hi ha alguna sèrie temporal associada o un altre motiu que faci sospitar de dependència seriada de les observacions, en general no cal analitzar-la.]

2.1 El model: Estimacions

2.1.1 Usant funcions bàsiques

Les estimacions dels coeficients β_i, de la variància σ^2, així com les prediccions $\widehat{y}_i$ i els residus e_i es poden obtenir usant R com calculadora a partir de les funcions covariància `cov()`, mitjana `mean()` i variància `var()`. **Exercici 2.1:** Aplica-ho per obtenir els resultats en forma de llista amb els noms com a la sortida següent:

```
## $intercept
## [1] 4.458081
##
## $pendent
## [1] -1.268756
##
## $mse
## [1] 0.003359123
```

2.1.2 Usant formulació matricial

La manera més senzilla d'obtenir les estimacions del model lineal és amb càlcul matricial. Definim les matrius

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \qquad Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

on la primera columna de X és tota de "1" i és per a l'intercept. Els estimadors dels coeficients (pendent i ordenada a l'origen en regressió simple) es poden obtenir a partir de les fórmules matricials següents:

$$\hat{\beta} = (\mathbf{X^t X})^{-1}\mathbf{X^t Y}$$

on $\hat{\beta}$ és el vector que conté els dos coeficients.

Les prediccions i els residus tambe es poden obtenir matricialment (vectors d'una columna):

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} \qquad \mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}}$$

Exercici 2.2. Fes una funció MLIN() aplicable a qualsevol parella de x i y que calculi matricialment les estimacions dels coeficients del model així com el vector amb totes les prediccions i el vector de residus i, a partir dels residus, l'estimació de la variància `MSE`. Aplica el codi a les variables preu i consum i comprova els resultats. **Atenció:** Amb R poden ser útils les funcions `rep()`, `matrix()` i `cbind()` per crear matrius, i recorda també les funcions: `\%*\%`, `t(x)` i `solve()`.

2.1.3 Usant la funció `lm()` i el sumari

Els mateixos resultats es poden obtenir aplicant la funció bàsica de la modelització lineal `lm()` : utilitzant aquesta funció i guardant en un objecte el resultat `mod<-lm()` i en un altre objecte el resum del model `smod<-summary(lm())`. *Recorda:* `names(mod)` i `names(smod)` permeten explorar els resultats que contenen ambdós objectes.

Exercici 2.3 Què són els objectes que conté `mod` ? Fixa'-vos't que es criden fent: `mod$nomobjecte`:

```
## ?lm
mod<-lm(y~x,data=data,x=TRUE,y=TRUE) ## x,y "true" per retornar les dades
names(mod)
mod$coeff          ## ??   # no cal posar el nom complet, si no es confon amb altres
mod$residuals      ## ??
mod$rank           ## rang de la matriu X (num de v. explicatives + 1 al model amb intercept)
mod$fitted.values  ## ??
mod$df.residual    ## ??
mod$xlevels        ## només si x fos un factor
mod$call           ## ??
mod$model          ## ??
mod$x              ## ??
mod$y              ## ??
## no interpretem la resta d'objectes
```

Apliquem `summary(mod)` i el guardem en un nou objecte `smod`. Conté nombrosos objectes, alguns ja estaven dins de `mod`:

```
smod<-summary(mod)
names(smod)
smod$sigma          ## és l'arrel quadrada de MSE
smod$r.squa         ## r^2, en reg. simple
smod$cov.unscaled   ## matriu de variàncies-covar dels coeficients
## els següents els veurem més endavant:
smod$adj.r.squa
smod$fstatistic
```

Exercici 2.4 Utilitzant els objectes mod i smod, dóna la mateixa llista que obteníem a l'exercici 1:

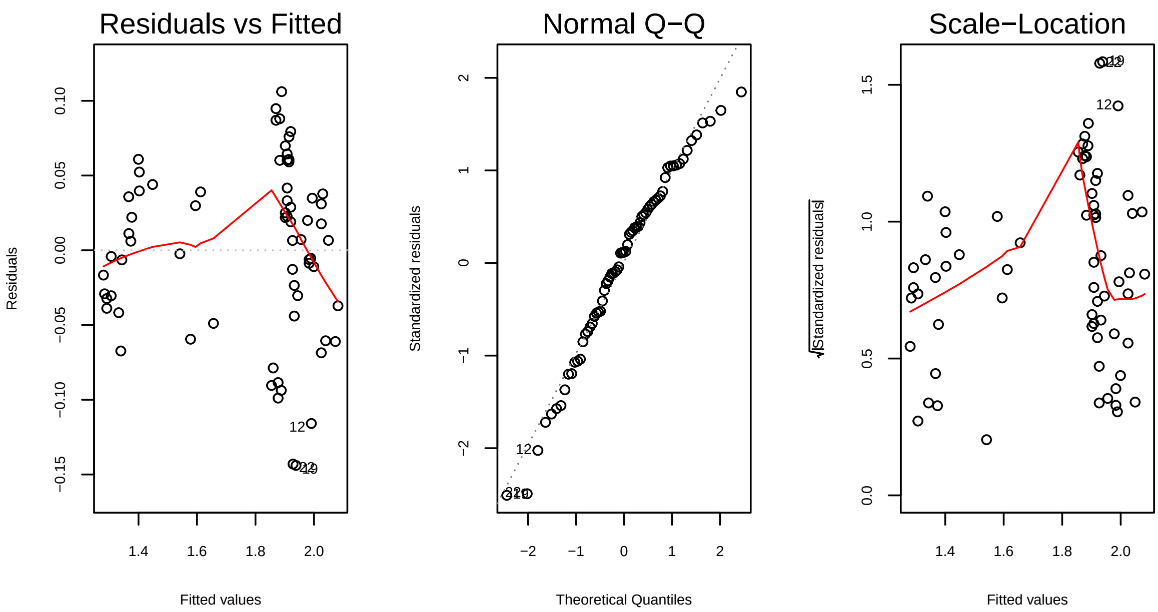
```
## $intercept
## [1] 4.458081
##
## $pendent
## [1] -1.268756
##
## $mse
## [1] 0.003359123
```

2.2 Gràfiques per analitzar el model: `plot(lm())`

Una variable resposta normal y_i, implica residus residus també normals, centrats i amb pautes de variància aprox. constant. Ara ens fixem en què podem fer quan les dades presenten alguna alteració evident de les hipòtesis.

Apliquem `plot(mod)` i vegem com fer una anàlisi bàsica dels residus. Tornarem a fer una revisió més exahustiva dels residus en pràctiques posteriors,

```
par(mfrow=c(1,3),cex=.7,cex.main=.6,cex.lab=.7,cex.axis=.7)
plot(mod,1:3)  # hi ha la possibilitat d'obtenir fins 6 gràfiques (en altres pràctiques)
```



Ordenades d'esquerra a dreta, la primera representa els residus e_i respecte de les prediccions $\hat{y}_i$ i indica que el centrament dels residus té alguna mancança (la línia vermella hauria de ser quasi-horitzontal), la segona és un *qq-plot* dels residus i indica una petita manca de normalitat a la cua dreta (asimetria) i la tercera, que representa l'arrel quadrada dels residus estandarditats respecte de les prediccions, apunta certa manca d'homocedasticitat, amb variàncies creixents primer i decreixents al final. Les dades marcades amb un número (la fila), són susceptibles de ser analitzades, ho veurem més endavant.

Són preocupants aquestes anomalies?

D'entrada, les anomalies sovint són presents i els mètodes són robustos enfront de *violacions lleus* de les hipòtesis, com en aquest cas. El més rellevant d'aquest exemple és que a la part central de les gràfiques (valors intermedis de $\hat{y}_i$) no hi pràcticament dades, per contra, hi ha un gran nombre de dades per a valors grans de $\hat{y}_i$. La mancança de dades només es pot arreglar obtenint-ne més.

2.2.1 El *qq-plot* o quantile-quantile-plot

Si tenim unes dades Gaussianes, al fer un qq-plot es veu una pauta d'ajust a la recta. Si les dades no són Gaussianes, el desajust a la recta és molt clar. En models lineals el qq-plot s'aplica als residus. Ara veurem la idea de la gràfica.

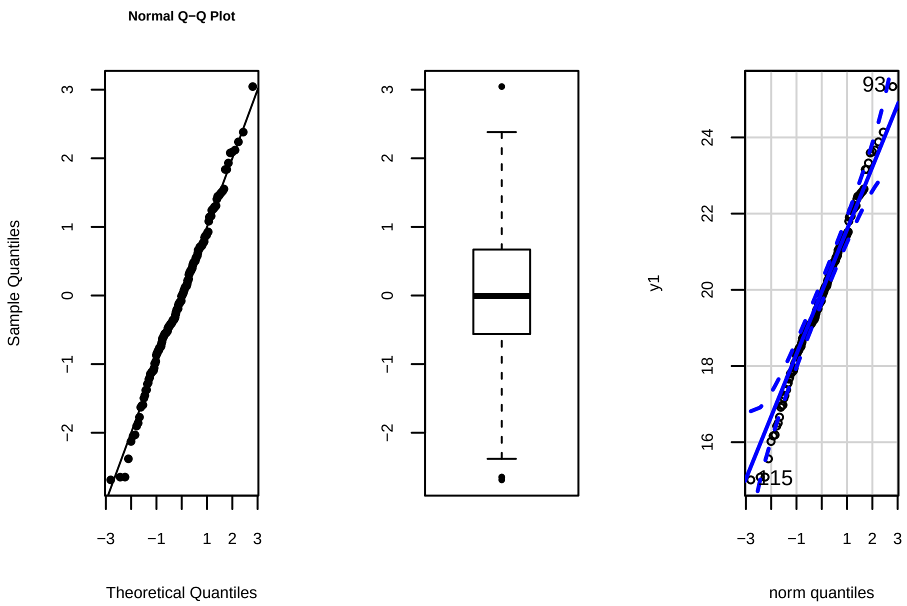
Què és un *pp-plot* d'una variable y? En un qq-plot es representen les dades tipificades $z_i = \frac{y_i - \overline{y}}{s_y}$ en un eix (el vertical, per exemple) i, a l'altre eix (horitzontal), les quantiles de la distribució $N(0,1)$, z_i^g, que correspondrien a la funció de distribució empírica de les dades. *Nota:* la funció de distribució empírica s'estima fent: $\frac{1}{n+1}, \frac{2}{n+1}, \ldots, \frac{n}{n+1}$ (modificada dividint per $n+1$ en lloc de n per tal que no acabi en 1, perquè en una normal l'1 no s'assoleix fins a l'infinit). Concretament:

$$z_i^g = \Phi^{-1}\left(\frac{i}{n+1}\right)$$

Si les dades són normals, els punts (z_i^g, z_i) estarien propers a la diagonal de la gràfica (bisectriu).

Vegem un exemple de qq-plot aplicat a dades Gaussianes.

```r
y1<-rnorm(200,20,2)      # gaussiana no-tipificada
par(mfrow=c(1,3),cex=.7,cex.main=.6,cex.lab=.7,cex.axis=.7,pch=20)
qqnorm(scale(y1))        # scale(y) tipifica ## qqnorm fa el qqplot
abline(a=0,b=1)          # tot i ser una normal, hi ha una certa fluctuació entorn de la diagonal
boxplot(scale(y1))
require(car)
qqPlot(y1)
```



Exercici 2.5: Repeteix la gràfica en dades no-Gaussianes (genera primer 200 valors d'una llei exponencial de paràmetre 1 i 200 valors d'una llei t-student amb 3 graus de llibertat) i observa el comportament. Comenta-ho.