Contents

Pràctica 4

Regressió simple: Intervals pels paràmetres. Bandes de confiança i de predicció.

• Intervals per als coeficients. • Interval per a la variància.
• Banda de confiança per a la resposta mitjana. • Banda de predicció. • Gràfiques de les bandes.

Utilitzarem les funcions de R: `lm()`, `summary(lm())`, `confint()`, `predict()`, `matplot()`, i noves funcions creades per l'usuari: `CI.var()`, `bandes()` i `wh.bandes()`

4.1 Intervals de confiança per als parametres

Torna a carregar les dades de y =preu i x =consum.

```
require(foreign)
data<-read.spss("preuconsum.sav",to.data.frame=T)
data<-data[!is.na(data$ID),1:3] ## eliminem els NA, casos que tots els valors son NA
names(data)<-c("id","y","x")
attach(data)# opcional
```

Guarda el resultat del model lineal de y sobre x en l'objecte `mod` i el sumari en l'objecte `smod`. *Recorda:* l'opció `x=TRUE`, `y=TRUE` dins de `lm()` fa que les dades x i y apareguin a l'output de la funció.

```
mod<-lm(y~x,data=data,x=TRUE,y=TRUE)   ## no cal attach si posem el nom del fitxer dins de lm()
smod<-summary(mod)
names(mod)
mod$x; mod$y                    ## per recuperar les dades inicials de l'output del model
```

1. A partir de `mod` i `smod`, guarda cadascun dels valors següents de manera individual en un objecte que es pugui fer servir després (noms suggerits entre parèntesis).

    - l'estimació de la intersecció (o intercept) $\hat{\beta}_0$: com interpreteu el valor? (`beta0`)
    - l'estimació del pendent $\hat{\beta}_1$: com interpreteu el valor? (`beta1`)
    - l'estimació de l'error típic de $\hat{\beta}_0$ i de $\hat{\beta}_1$ (`s.beta0` i `s.beta1`)
    - els graus de llibertat de la distribució $t-$ de Student (`gl`)
    - el valor de l'estadístic de contrast de $H_0 : \beta_0 = 0$ (`t.beta0`)
    - el valor de l'estadístic de contrast de $H_0 : \beta_1 = 0$ (`t.beta1`)
    - el p-valor del contrast $H_0 : \beta_0 = 0$ bilateral: com interpreteu el resultat? (`p.beta0`)

- el p-valor del contrast $H_0 : \beta_1 = 0$ bilateral: com interpreteu el resultat? (`p.beta1`)
- l'estimació de la variància del model $\hat{\sigma^2}$ (MSE)
- l'estimació de R-quadat $R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$, on $SSE = \sum e_i^2$ i $SST = \sum(y_i - \overline{y})^2$ (`R2`). Com interpretes el resultat obtingut?
- l'estimació de R-quadrat ajustat: $R_a^2 = 1 - \frac{MSE}{MST} = 1 - \frac{\frac{SSE}{n-2}}{\frac{SST}{n-1}}$. (`R2.a`) *Observació:* El coeficient de determinació ajustat permet comparar diversos models de regressió.

2. Amb els errors típics i els graus de llibertat guardats al problema anterior, calcula les quantiles de la distribució que pertoca i, seguidament, l'interval de confiança (lsup i linf), amb un NC del 95%, per a cada coeficient, β_0 i β_1.

```
## $linf.beta0
## [1] 4.315113
##
## $lsup.beta0
## [1] 4.601048
##
## $linf.beta1
## [1] -1.335924
##
## $lsup.beta1
## [1] -1.201588
```

3. Utilitza la funció `confint()`, aplicada al model, per obtenir els intervals de confiança per als coeficients i comprova que són (aproximadament) els mateixos que has calculat a l'apartat anterior. Interpreta els intervals.

   Canvia el nivell de confiança al 99%, calcula de nou els intervals i comenta els canvis.

4. Fes el codi d'una funció que anomenaràs `CI.var(x,y, nc=0.95)` (recorda que el valor numèric que s'assigna a un argument a la definició d'una funció és només el valor per defecte, el qual es pot canviar quan s'aplica la funció), aquesta funció ha de retornar l'estimació puntual de σ^2 i l'interval de confiança per al nivell de confiança desitjat. *Indicació:* dins de la funció li faràs calcular `mod` i `smod` per extraure informació d'aquests objectes. Recorda: no fer servir dins d'una funció objectes que hi ha fora de l'espai. Tots els objectes han de provenir dels arguments de la funció.
   Aplica la funció per calcular l'interval amb un nivell de confiança del 99%.

```
CI.var<-function(x,y,nc=0.95){
.
.
.
}
```

```
## $MSE
## [1] 0.003359123
##
## $linf.sigma2
## [1] 0.002459166
##
## $lsup.sigma2
## [1] 0.004865029
```

4.2 Bandes de confiança i de predicció de noves observacions

1. Considerem un vector de dades noves que posarem en forma de dataframe (`newdata`) -li direm `xh` , per abreujar-.

Observació important: Dins del data frame de dades noves, la variable independent ha de tenir el mateix nom que a la base de dades d'on s'ha extret el model. Tingues en compte que la funció `predict()` dóna les prediccions de `newdata`.

```
xh<-data.frame(x=seq(1.85,2.51,by=.03)) ## ATENCIÓ !!
## a la variable del data.frame li posem el MATEIX NOM  que a les dades!! ("x", pel cas del preu consum)
## si volguésssim predir uns pocs valors, faríem: xh<- data.frame(x=c(1.7,2.1))
predict(mod,newdata=xh)
```

Fes una funció (`bandes(x,y,xh,level)`) que calculi la **banda de confiança per a la resposta mitjana** i la **banda de predicció** d'aquestes obervacions. Per assajar, pots considerar que el vector de dades noves és el vector `xh` que apareix al chunk. *Nota:* Dins del codi de la funció *bandes()*, pots utilitzar la funció `predict(mod,newdata)` per obtenir les prediccions de "noves observacions".

```
bandes<-function(x,y,xh,level=.95){
  .
  .
  .
}
```

L'output de la funció aplicada a les dades de preu i consum i a `xh` anterior amb nivell de confiança del 95%, ha de ser el data.frame del qual es mostra la capçalera:

```
head(bandes(data$x,data$y,xh))

##    new     pred    lmean    umean    lpred    upred
## 1 1.85 2.110883 2.088103 2.133662 1.992977 2.228788
## 2 1.88 2.072820 2.051599 2.094040 1.955205 2.190435
## 3 1.91 2.034757 2.015013 2.054502 1.917400 2.152115
## 4 1.94 1.996695 1.978323 2.015066 1.879560 2.113829
## 5 1.97 1.958632 1.941506 1.975757 1.841687 2.075577
## 6 2.00 1.920569 1.904532 1.936606 1.803778 2.037360
```

2. Plot de les bandes. Utilitzem la funció `matplot()` per fer diverses corbes alhora.

```
xbandes<-xh
ybandes<-bandes(data$x,data$y,xh)[,-1]  ## omitim la 1a columna "new" de l'output de bandes !!

matplot(xbandes, ybandes, ylim=c(min(ybandes),max(ybandes)),
        lty = c(1,1,1,2,2), col=c(1,2,2,4,4), type = "l",
        ylab = "prediccions del consum",xlab="preu",
        main="bandes de confiança (vermell) i de predicció (blau)", cex.main=.8)
points(data$x,data$y,pch=20,cex=.8)
```

3. Comprova que *no caldria haver creat al funció `bandes()`* perquè el codi següent dóna les bandes utilitzanat la funció `predict()` de R:

```
conf.band<-predict(mod,newdata=xh,interval="confidence")
                              ## la primera columna són les estimacions de la resposta mitjana
pred.band<-predict(mod,newdata=xh,interval="prediction")
                              ## la primera columna, prediccions puntuals, és repetida de l'anterior
ybandes2<-data.frame(conf.band,pred.band[,-1])  ## traiem 1a columna repetida
```

Fés el matplot per veure que és el mateix resultat.

4. Comprova que també s'obté el mateix amb la funció `plotFit()` de la llibreria `investr`:

```
require(investr)
plotFit(mod,"both")
plotFit(mod,"both",pch=19,cex=.8,shade=T,col.conf = "skyblue4", col.pred = "lightskyblue2")
```

5. Bandes de confiança i de predicció de *Working-Hotelling* **amb correcció per múltiples intervals o bandes de confiança experimentals**. Cal fer la funció `wh.bandes(x,y,level=.95)` i la grafica. *Nota:* les fórmules són idèntiques al codi anterior, només canvia la quantila t-Student per w:

$$w = \sqrt{2 * F_{1-\alpha,2,n-2}}$$

Observació: Les bandes de Working-Hotelling són molt més segures (més àmplies) per a totes les prediccions simultànies, però són més ajustades que les de Bonferroni. A més. les de Bonferroni són cada cop més grans, quantes més prediccions fem, i les de W-Hotelling no,2 perquè només depenen del nombre de dades n amb les quals s'ha estimat el model.

```
wh.bandes<-function(x,y,xh,level=.95){
  .
  .
  .
}
head(wh.bandes(data$x,data$y,xh))
```

Mostra la capçalera del data.frame resultant.

```
head(wh.bandes(data$x,data$y,xh),3)


##    new     pred wh.lmean wh.umean wh.lpred wh.upred
## 1 1.85 2.110883 2.088103 2.133662 1.962998 2.258767
## 2 1.88 2.072820 2.051599 2.094040 1.925301 2.220339
## 3 1.91 2.034757 2.015013 2.054502 1.887561 2.181953
```

Fes un matplot de les bandes de Working-Hotelling.

Comprova que s'obté el mateix resultat fent ...

```
require(investr)
plotFit(mod,"both",adjust="Scheffe",k=2)
```