Práctica Calificada No3

Lourdes Galarza, Jaime Gómez, Julio Gutiérrez

28 December, 2018

## Pregunta 1 (20 puntos)

Se han recolectado datos sobre la velocidad del viento X y la corriente producida Y por un molino de viento a esta velocidad, los cuales se encuentran en el archivo molinos.csv. Para explicar la corriente producida en función de la velocidad del viento se consideró un modelo lineal normal con los siguientes enlaces:

* Modelo 1:
* Modelo 2:
* Modelo 3:

### (a) Usando WinBUGS a través de R estime los parámetros de los 3 modelos usando inferencia bayesiana. Realice el análisis de convergencia de las simulaciones. (12 p.)

##### Cargando librerias:

library(coda)  
library(boot)  
library(R2WinBUGS)  
library(mcmcplots)

##### Conectando con winBUGS:

#Directorio donde esta instalado WinBUGS  
WINBUGS.DIR <- "D:/bin/WinBUGS14/"  
#Nombre de los códigos BUGS de los 3 modelos  
NAME.FILE.MOD1.BUG <- "Practica03-modelo01.bug"  
NAME.FILE.MOD2.BUG <- "Practica03-modelo02.bug"  
NAME.FILE.MOD3.BUG <- "Practica03-modelo03.bug"

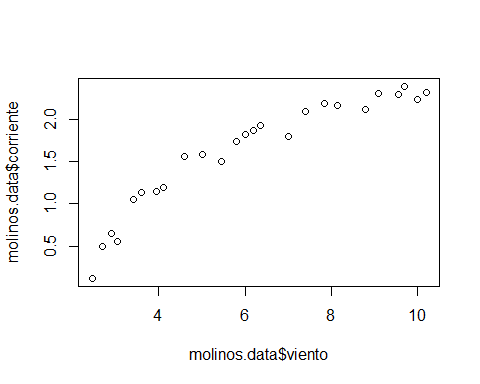
##### Cargando los datos:

PATH.FILE <- "https://raw.githubusercontent.com/jgomezz/MscEstadisticaAplicada-UNALM-2018-2/master/MLG/Practica-03/molinos.csv"  
molinos.data <- read.csv(PATH.FILE, header = TRUE)  
head(molinos.data)

## viento corriente  
## 1 5.0 1.582  
## 2 6.0 1.822  
## 3 3.4 1.057  
## 4 2.7 0.500  
## 5 10.0 2.236  
## 6 9.7 2.386

##### Análisis descriptivo:

plot(molinos.data$viento, molinos.data$corriente)



Se observa que entre las variables corriente y viento hay, aproximadamente, una relación lineal directa.

##### Definición de datos, parámetros de monitoreo e inicializaciones para el WinBUGS:

# Datos para el análisis  
molinos.data.bugs <- list(viento = molinos.data$viento ,  
 corriente = molinos.data$corriente ,  
 N = nrow(molinos.data))  
  
# Creación de parámetros  
molinos.param.bugs <- c("alpha","beta1","tau","sigma2")  
  
# Inicialización : asigna valores aleatorios para inicializar la simulación  
molinos.inits.bugs <- function() {  
 list( alpha = rnorm(1),   
 beta1 = rnorm(1),   
 tau = rgamma(1,1,1) )}

#### Enlace 1 del modelo lineal normal :

##### Inferencia Clásica

molinos.model.1.clasic<-lm(molinos.data$corriente ~ molinos.data$viento)  
summary(molinos.model.1.clasic)

##   
## Call:  
## lm(formula = molinos.data$corriente ~ molinos.data$viento)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.59869 -0.14099 0.06059 0.17262 0.32184   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.13088 0.12599 1.039 0.31   
## molinos.data$viento 0.24115 0.01905 12.659 7.55e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2361 on 23 degrees of freedom  
## Multiple R-squared: 0.8745, Adjusted R-squared: 0.869   
## F-statistic: 160.3 on 1 and 23 DF, p-value: 7.546e-12

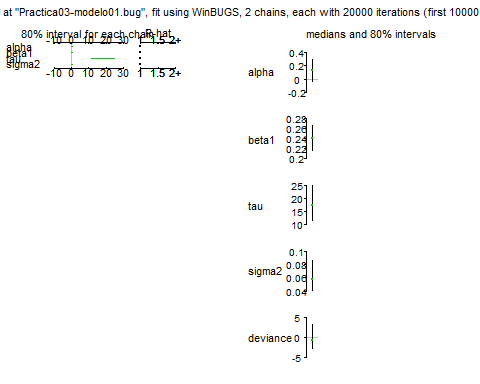
##### Inferencia Bayesiana

molinos.model.1.bugs <- function(){  
 # verosimilitud  
 for (i in 1:N) {  
 mu[i] <- alpha + beta1\*viento[i];  
 corriente[i] ~ dnorm(mu[i],tau);  
 }  
 # Las priori : estoy usando priori no informativos   
 # porque le estoy dando un rango amplio ( 0.0 en casi todos los modelos)  
 alpha ~ dnorm(0.0,1.0E-4); # 1er parametro   
 beta1 ~ dnorm(0.0,1.0E-4); # 2do parametro  
 tau ~ dgamma(1.0E-3,1.0E-3); # 3er parametro , la precisi?n , uso gamma   
  
 # con valores pequeños para tener una varianza grande  
 sigma2 <- 1/tau;  
}  
  
# Grabar archivo  
write.model(molinos.model.1.bugs, NAME.FILE.MOD1.BUG)  
  
# Inferencia Bayesiana  
molinos.fit.model.1.bugs <- bugs(data = molinos.data.bugs,  
 inits = molinos.inits.bugs,  
 parameters.to.save = molinos.param.bugs,  
 model.file= NAME.FILE.MOD1.BUG,  
 n.chains=2,   
 n.iter=20000,  
 n.burnin=10000,  
 n.thin=1,  
 bugs.directory=WINBUGS.DIR,  
 clearWD=TRUE,   
 debug=FALSE)  
  
# Resultados de simulación  
print(molinos.fit.model.1.bugs,4)

## Inference for Bugs model at "Practica03-modelo01.bug", fit using WinBUGS,  
## 2 chains, each with 20000 iterations (first 10000 discarded)  
## n.sims = 20000 iterations saved  
## mean sd 2.5% 25% 50% 75% 97.5% Rhat  
## alpha 0.1328 0.1314 -0.1290 0.0462 0.1330 0.2189 0.3925 1.0010  
## beta1 0.2409 0.0198 0.2024 0.2278 0.2409 0.2540 0.2803 1.0010  
## tau 17.8848 5.2740 9.1159 14.0900 17.3850 21.0900 29.6200 1.0010  
## sigma2 0.0612 0.0197 0.0338 0.0474 0.0575 0.0710 0.1097 1.0010  
## deviance -0.1616 2.5452 -3.0860 -2.0360 -0.7978 1.0192 6.4210 1.0011  
## n.eff  
## alpha 20000  
## beta1 20000  
## tau 20000  
## sigma2 20000  
## deviance 14000  
##   
## For each parameter, n.eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).  
##   
## DIC info (using the rule, pD = Dbar-Dhat)  
## pD = 3.1 and DIC = 2.9  
## DIC is an estimate of expected predictive error (lower deviance is better).

##### Diagnóstico de Convergencia

plot(molinos.fit.model.1.bugs)



#### Enlace 2 del modelo lineal normal :

##### Inferencia Clásica

molinos.model.2.clasic<-lm(molinos.data$corriente ~ I(1/molinos.data$viento))  
summary(molinos.model.2.clasic)

##   
## Call:  
## lm(formula = molinos.data$corriente ~ I(1/molinos.data$viento))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.20547 -0.04940 0.01100 0.08352 0.12204   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.9789 0.0449 66.34 <2e-16 \*\*\*  
## I(1/molinos.data$viento) -6.9345 0.2064 -33.59 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09417 on 23 degrees of freedom  
## Multiple R-squared: 0.98, Adjusted R-squared: 0.9792   
## F-statistic: 1128 on 1 and 23 DF, p-value: < 2.2e-16

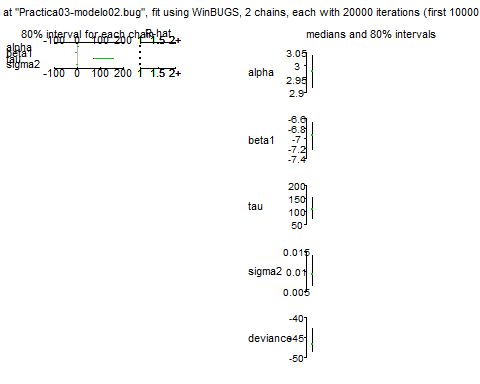
##### Aplicando Bayesianos

molinos.model.2.bugs <- function(){  
 # verosimilitud  
 for (i in 1:N) {  
 mu[i] <- alpha + beta1\*(1/viento[i]);  
 corriente[i] ~ dnorm(mu[i],tau);  
 }  
 # Las priori : estoy usando priori no informativos   
 # porque le estoy dando un rango amplio ( 0.0 en casi todos los modelos)  
 alpha ~ dnorm(0.0,1.0E-4); # 1er parametro   
 beta1 ~ dnorm(0.0,1.0E-4); # 2do parametro  
 tau ~ dgamma(1.0E-3,1.0E-3); # 3er parametro , la precisi?n , uso gamma   
   
 # con valores pequeños para tener una varianza grande  
 sigma2 <- 1/tau;  
}  
  
# Grabando archivo  
write.model(molinos.model.2.bugs, NAME.FILE.MOD2.BUG)  
  
# Inferencia Bayesiana  
molinos.fit.model.2.bugs <- bugs(data = molinos.data.bugs,  
 inits = molinos.inits.bugs,  
 parameters.to.save = molinos.param.bugs,  
 model.file= NAME.FILE.MOD2.BUG,  
 n.chains=2,   
 n.iter=20000,  
 n.burnin=10000,  
 n.thin=1,  
 bugs.directory=WINBUGS.DIR,  
 clearWD=TRUE,   
 debug=FALSE)  
  
#Mostrar resultados de la simulación  
print(molinos.fit.model.2.bugs,4)

## Inference for Bugs model at "Practica03-modelo02.bug", fit using WinBUGS,  
## 2 chains, each with 20000 iterations (first 10000 discarded)  
## n.sims = 20000 iterations saved  
## mean sd 2.5% 25% 50% 75% 97.5%  
## alpha 2.9796 0.0470 2.8850 2.9490 2.9800 3.0100 3.0720  
## beta1 -6.9376 0.2158 -7.3560 -7.0800 -6.9380 -6.7950 -6.5080  
## tau 111.4552 32.8664 56.8095 87.7800 108.3000 131.4000 184.6000  
## sigma2 0.0098 0.0032 0.0054 0.0076 0.0092 0.0114 0.0176  
## deviance -46.0906 2.5581 -49.0300 -47.9725 -46.7300 -44.9100 -39.4700  
## Rhat n.eff  
## alpha 1.0010 20000  
## beta1 1.0010 20000  
## tau 1.0010 20000  
## sigma2 1.0010 20000  
## deviance 1.0011 14000  
##   
## For each parameter, n.eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).  
##   
## DIC info (using the rule, pD = Dbar-Dhat)  
## pD = 3.1 and DIC = -43.0  
## DIC is an estimate of expected predictive error (lower deviance is better).

##### Diagnóstico de Convergencia

plot(molinos.fit.model.2.bugs)



#### Enlace 3 del modelo lineal normal :

##### Inferencia Clásica

molinos.model.3.clasic<-lm(molinos.data$corriente ~ I(log(molinos.data$viento)))  
summary(molinos.model.3.clasic)

##   
## Call:  
## lm(formula = molinos.data$corriente ~ I(log(molinos.data$viento)))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.31619 -0.07685 0.02395 0.11139 0.23029   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.83036 0.11083 -7.493 1.3e-07 \*\*\*  
## I(log(molinos.data$viento)) 1.41677 0.06234 22.728 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1376 on 23 degrees of freedom  
## Multiple R-squared: 0.9574, Adjusted R-squared: 0.9555   
## F-statistic: 516.6 on 1 and 23 DF, p-value: < 2.2e-16

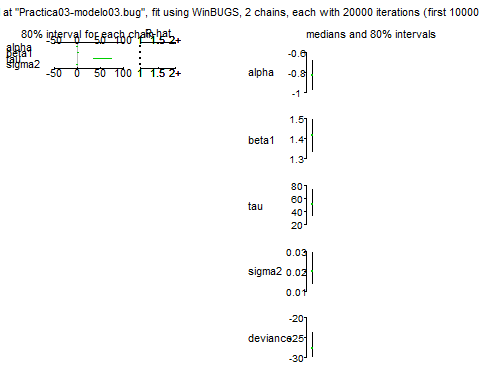
##### Aplicando Bayesianos

molinos.model.3.bugs <- function(){  
 # verosimilitud  
 for (i in 1:N) {  
 mu[i] <- alpha + beta1\*log(viento[i]);  
 corriente[i] ~ dnorm(mu[i],tau);  
 }  
 # Las priori : estoy usando priori no informativos   
 # porque le estoy dando un rango amplio ( 0.0 en casi todos los modelos)  
 alpha ~ dnorm(0.0,1.0E-4); # 1er parametro   
 beta1 ~ dnorm(0.0,1.0E-4); # 2do parametro  
 tau ~ dgamma(1.0E-3,1.0E-3); # 3er parametro , la precisi?n , uso gamma   
   
 # con valores pequeños para tener una varianza grande  
 sigma2 <- 1/tau;  
}  
  
# Grabar archivo  
write.model(molinos.model.3.bugs, NAME.FILE.MOD3.BUG)  
  
# Inferencia Bayesiana  
molinos.fit.model.3.bugs <- bugs(data = molinos.data.bugs,  
 inits = molinos.inits.bugs,  
 parameters.to.save = molinos.param.bugs,  
 model.file= NAME.FILE.MOD3.BUG,  
 n.chains=2,   
 n.iter=20000,  
 n.burnin=10000,  
 n.thin=1,  
 bugs.directory=WINBUGS.DIR,  
 clearWD=TRUE,   
 debug=FALSE)  
#Mostrar resultados de la simulación  
print(molinos.fit.model.3.bugs,4)

## Inference for Bugs model at "Practica03-modelo03.bug", fit using WinBUGS,  
## 2 chains, each with 20000 iterations (first 10000 discarded)  
## n.sims = 20000 iterations saved  
## mean sd 2.5% 25% 50% 75% 97.5%  
## alpha -0.8286 0.1157 -1.0590 -0.9046 -0.8284 -0.7523 -0.6008  
## beta1 1.4158 0.0650 1.2900 1.3730 1.4160 1.4590 1.5450  
## tau 52.4982 15.4810 26.7597 41.3500 51.0300 61.9000 86.9502  
## sigma2 0.0208 0.0067 0.0115 0.0162 0.0196 0.0242 0.0374  
## deviance -27.1516 2.5498 -30.0800 -29.0300 -27.7900 -25.9700 -20.5500  
## Rhat n.eff  
## alpha 1.0010 20000  
## beta1 1.0010 20000  
## tau 1.0010 20000  
## sigma2 1.0010 20000  
## deviance 1.0011 14000  
##   
## For each parameter, n.eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).  
##   
## DIC info (using the rule, pD = Dbar-Dhat)  
## pD = 3.1 and DIC = -24.1  
## DIC is an estimate of expected predictive error (lower deviance is better).

##### Diagnóstico de Convergencia

plot(molinos.fit.model.3.bugs)



### (b) En base a su respuesta en (a) escoja el mejor modelo (usando el DIC) para explicar la corriente producida en función de la velocidad del viento. Para el modelo escogido verifique si existen observaciones influyentes. (4 p.)

#### Elección del mejor modelo:

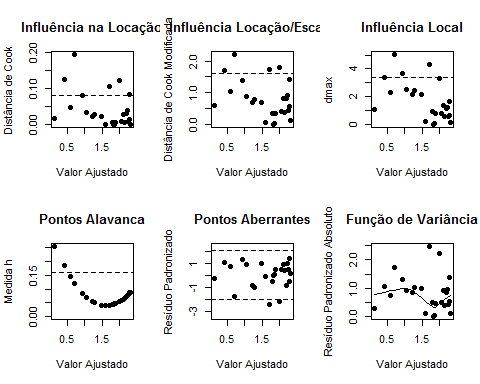
#Comparación de los modelos basados en el DIC : Deviance Information Criterion  
df <- data.frame(  
 modelo = c(" Modelo 1",   
 " Modelo 2",   
 " Modelo 3"),  
 DIC = c(molinos.fit.model.1.bugs$DIC,  
 molinos.fit.model.2.bugs$DIC,  
 molinos.fit.model.3.bugs$DIC)  
 )  
  
df[order(df$DIC),]

## modelo DIC  
## 2 Modelo 2 -43.021  
## 3 Modelo 3 -24.082  
## 1 Modelo 1 2.908

#### El modelo con el DIC más pequeño es el modelo 2, por lo que será el modelo de elección.

#### Interpretación de valores influyentes para el modelo 2 con ajuste clásico

#source.with.encoding('diag.pois.R', encoding='ISO-8859-1')  
source('http://www.poleto.com/funcoes/diag.norm.txt')  
diag.norm(molinos.model.2.clasic)



## $ResPearsonStd  
## 1 2 3 4 5 6   
## -0.10550271 -0.01171169 1.32821007 1.05433756 -0.54013842 1.37948240   
## 7 8 9 10 11 12   
## 0.44923248 -1.74037950 0.41015052 0.05972004 0.74301684 0.46191428   
## 13 14 15 16 17 18   
## 0.98270313 -0.49370374 0.49775225 0.92704397 0.91321733 -0.86516194   
## 19 20 21 22 23 24   
## -2.22465157 -2.46056219 0.95044236 0.11960300 -1.01982540 -0.86108599   
## 25   
## -0.30632719   
##   
## $Di  
## 1 2 3 4 5   
## 2.425065e-04 3.343512e-06 7.920370e-02 1.244580e-01 1.409659e-02   
## 6 7 8 9 10   
## 8.263683e-02 9.254502e-03 1.928880e-01 6.252177e-03 9.046160e-05   
## 11 12 13 14 15   
## 4.749893e-02 5.530286e-03 2.115307e-02 5.663941e-03 7.980953e-03   
## 16 17 18 19 20   
## 3.304417e-02 2.852286e-02 3.013265e-02 1.213192e-01 1.059356e-01   
## 21 22 23 24 25   
## 3.775662e-02 7.164994e-04 2.753313e-02 2.178619e-02 1.657276e-02   
##   
## $Ci  
## 1 2 3 4 5 6   
## 0.07306054 0.00857658 1.37193588 1.69600301 0.56056909 1.40544153   
## 7 8 9 10 11 12   
## 0.45328433 2.19722085 0.37228976 0.04461477 1.03498764 0.35049431   
## 13 14 15 16 17 18   
## 0.69699003 0.35494681 0.42137667 0.86912044 0.80702536 0.82793280   
## 19 20 21 22 23 24   
## 1.80815928 1.72393617 0.92992087 0.12559154 0.79646999 0.70388206   
## 25   
## 0.60510805   
##   
## $Dmax  
## [1] 0.22492325 0.02160714 3.65835893 3.39572927 0.67186852 1.69364169  
## [7] 0.57875122 5.02345873 0.59450816 0.10735192 2.32986651 0.81097706  
## [13] 2.19099395 0.93050748 0.77512953 2.49895202 1.34180282 1.16901076  
## [19] 3.28582243 4.33877646 1.24868921 0.14761293 2.48327216 2.17152317  
## [25] 1.05159510  
##   
## $h  
## 1 2 3 4 5 6   
## 0.04003113 0.04455503 0.08489949 0.18367814 0.08563836 0.08278760   
## 7 8 9 10 11 12   
## 0.08133173 0.12172805 0.06685400 0.04628476 0.14436551 0.04767874   
## 13 14 15 16 17 18   
## 0.04190993 0.04301321 0.05866268 0.07100285 0.06359075 0.07376002   
## 19 20 21 22 23 24   
## 0.05432426 0.04093763 0.07684526 0.08749331 0.05036690 0.05491366   
## 25   
## 0.25334699

Se obtuvieron los gráficos de valores influenciables, en estos podemos verificar según el grafico 1 de valores ajustados vs. distancia de cook que se evidencia, de manera general, y con respecto a las distancia no existe un valor que pueda ser considerado tomado con precaución; sin embargo, se visualiza un dato que se aleja más de la concentración de los otro puntos. Asimismo, y con respecto al grafico de puntos aberrantes, se puede evidenciar que casi la mayoría de los datos tienen un patrón aleatorio, y solo dos puntos se alejan de los limites -2 y 2, aunque este alejamiento no es sustancial por lo que aparentemente no existe ningún valor considerable. El mismo patrón aleatorio se visualiza en el gráfico de influencia localización escala, esto nos mostraría que no existe problemas algún tipo de violación del supuesto de homocedasticidad.

### (c) En base a su respuesta en (b) prediga la corriente producida por un molino cuando la velocidad del viento es de 5,8 (4 p.)

#### Según lo observado en el DIC, se elige el modelo 2

# Se obtiene la matriz de simulación  
sims.matrix <- molinos.fit.model.2.bugs$sims.matrix  
  
# Se obtiene los coeficientes de las variables predictoras  
p.alpha <- sims.matrix[,"alpha"]  
p.beta1 <- sims.matrix[,"beta1"]  
p.sigma <- 1/sqrt(sims.matrix[,"tau"])  
M <- nrow(sims.matrix)  
  
# Velocidad de viento es 5.8  
viento <- 5.8   
  
# Se realiza la predicción  
mu.new <- p.alpha + p.beta1\*(1/viento)  
y.new <- rnorm(M,mu.new,p.sigma)  
  
summary(mu.new)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.700 1.770 1.783 1.783 1.797 1.907

quantile(mu.new,probs=c(0.025,0.975))

## 2.5% 97.5%   
## 1.742999 1.823690

summary(y.new)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.300 1.718 1.784 1.784 1.850 2.234

quantile(y.new,probs=c(0.025,0.975))

## 2.5% 97.5%   
## 1.582900 1.984396

# Resultado  
sprintf("Para un viento de %.2f se tiene una corriente de %.2f",viento, mean(y.new))

## [1] "Para un viento de 5.80 se tiene una corriente de 1.78"

#### Tener en consideración que no se tiene las unidades de viento y corriente para poder incluirlo en el resultado