Script6

Yosune Miquelajauregui

31/12/2017

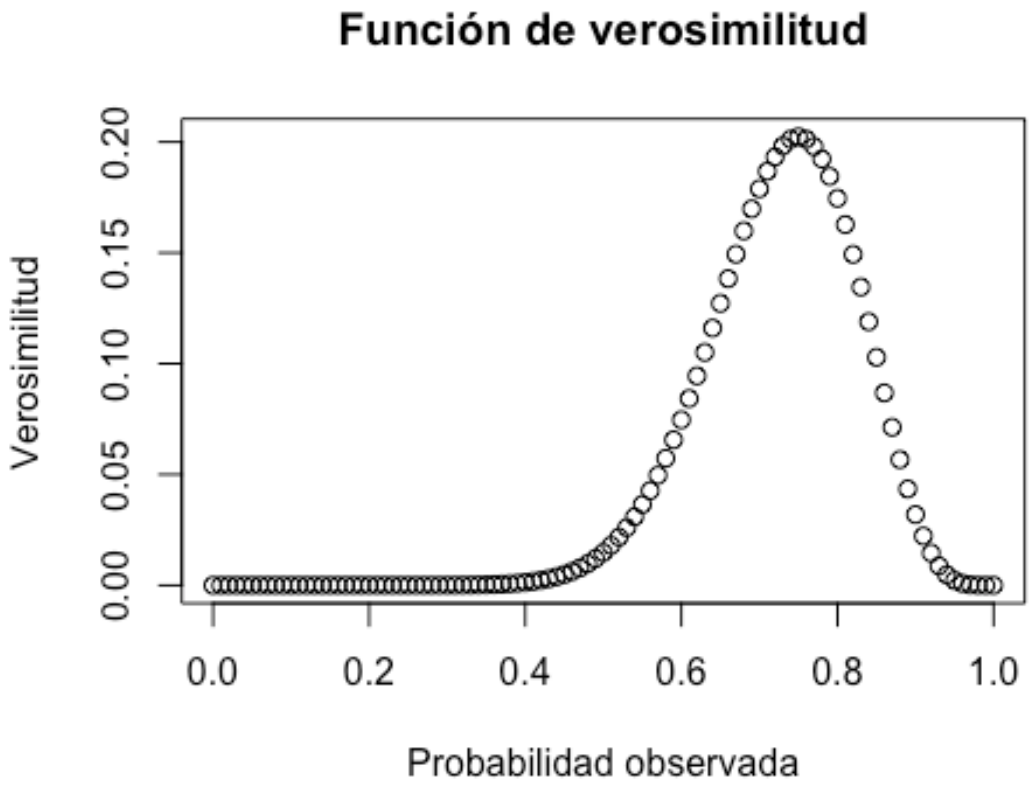
Criterio de información de Akaike, inferencia multimodelo y máxima verosimilitud

Máxima verosimilitud

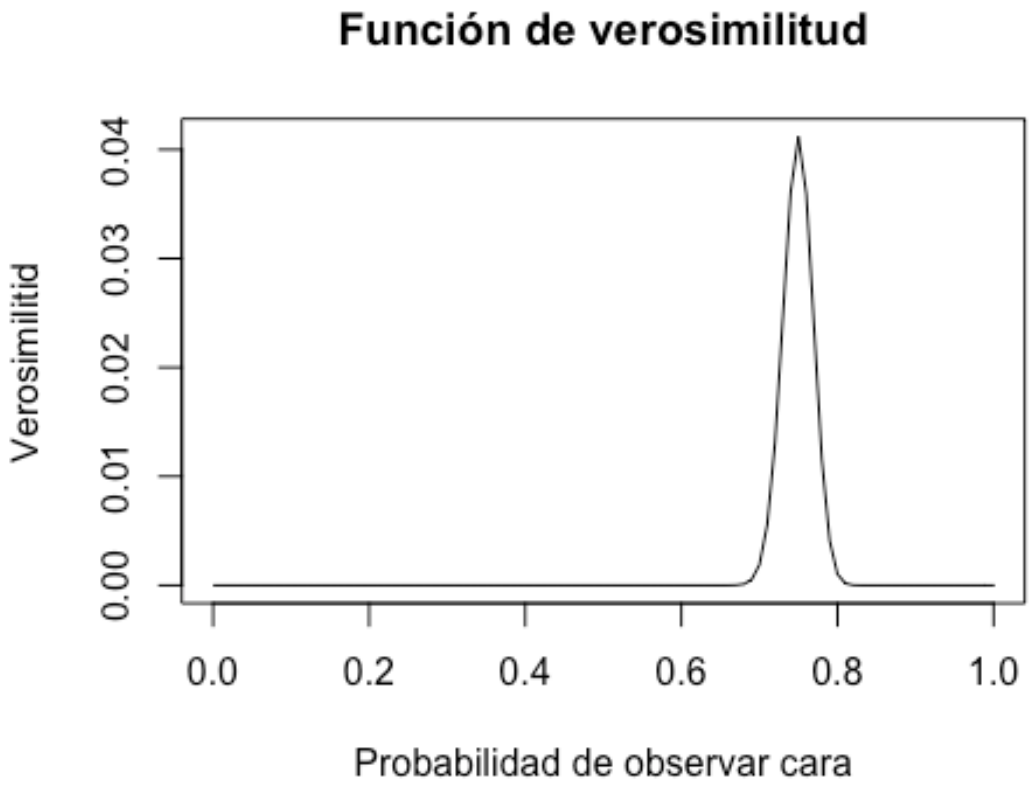
```
lanzamiento <-
c("cruz","cara","cara","cara","cara","cara","cara","cruz","cruz","cara","
cara","cara","cara","cara","cara","cruz","cara","cruz","cara")
table(lanzamiento)
```

```
## lanzamiento
## cara cruz
##   15    5
```

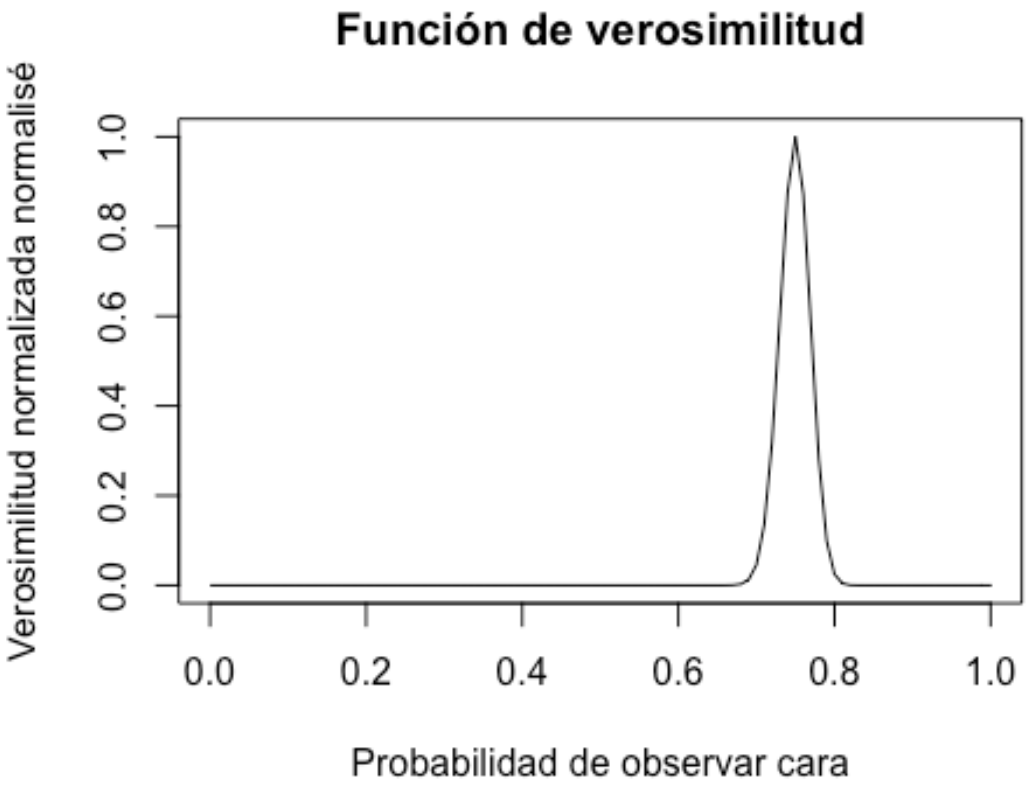
```
theta <- seq (from=0, to =1, by=0.01)
Vero <- dbinom(x=15, size=20, prob=theta)
plot(Vero~theta, ylab="Verosimilitud", xlab="Probabilidad observada",
main="Función de verosimilitud")
```



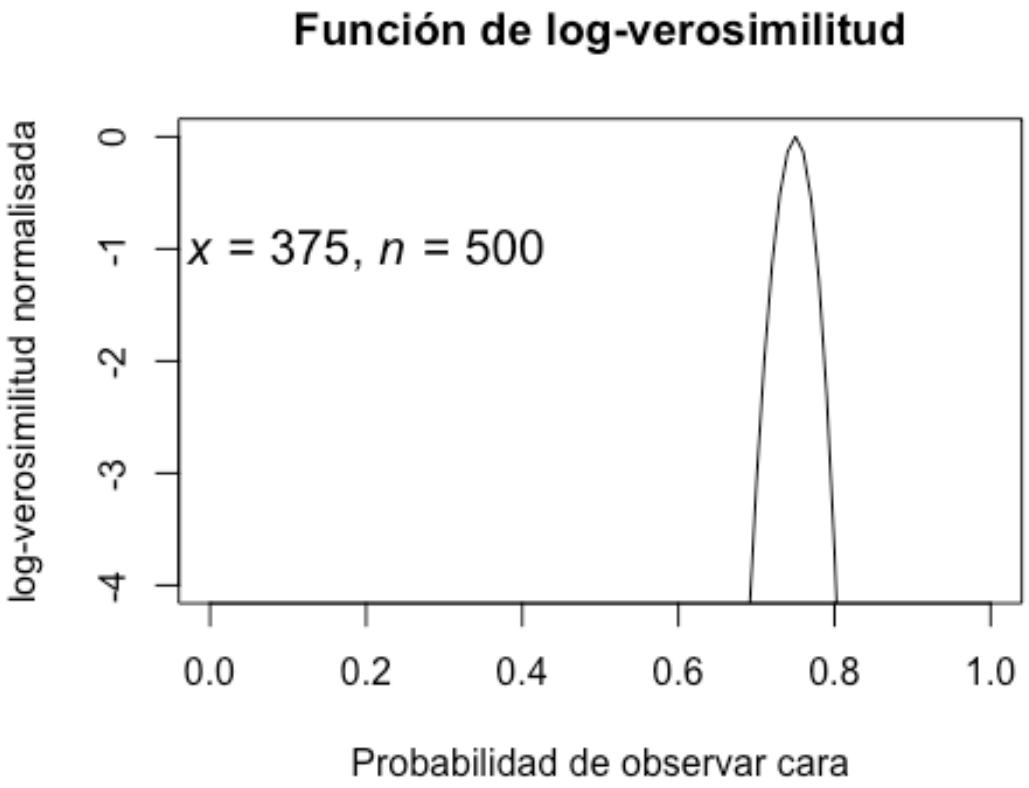

```r
which(Vero==max(Vero))

## [1] 76

theta[76]

## [1] 0.75

Vero_norm<-Vero/max(Vero)  #valores de la función de máxima verosimilitud
normalizada para tener un máximo de 1

x<-375
n<-500
theta<-seq(from=0, to=1, by=0.01)
Vero375<-dbinom(x=x, size=n, prob=theta)
Vero375_norm<-Vero375/max(Vero375)
ve375 <- plot(Vero375~theta, main="Función de verosimilitud",
xlab="Probabilidad de observar cara", ylab="Verosimilitid", type="l")
```



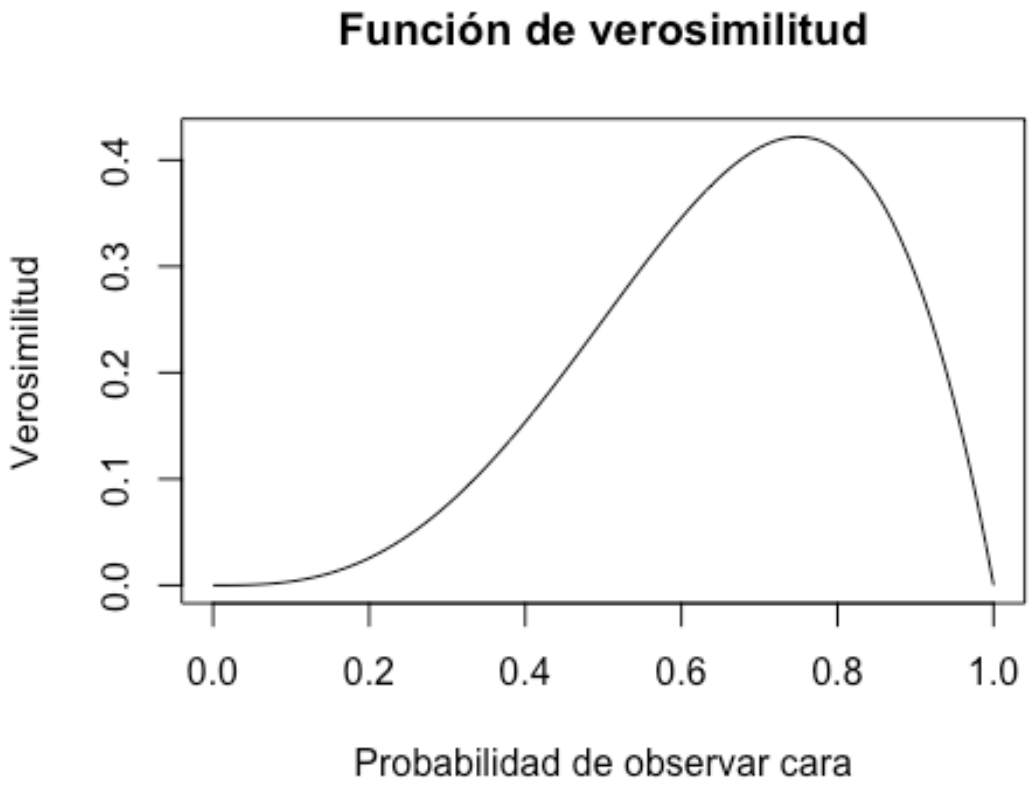

```
ve375nor <- plot(Vero375_norm~theta, main="Función de verosimilitud",
xlab="Probabilidad de observar cara", ylab="Verosimilitud normalizada
normalisé", type="l")
```



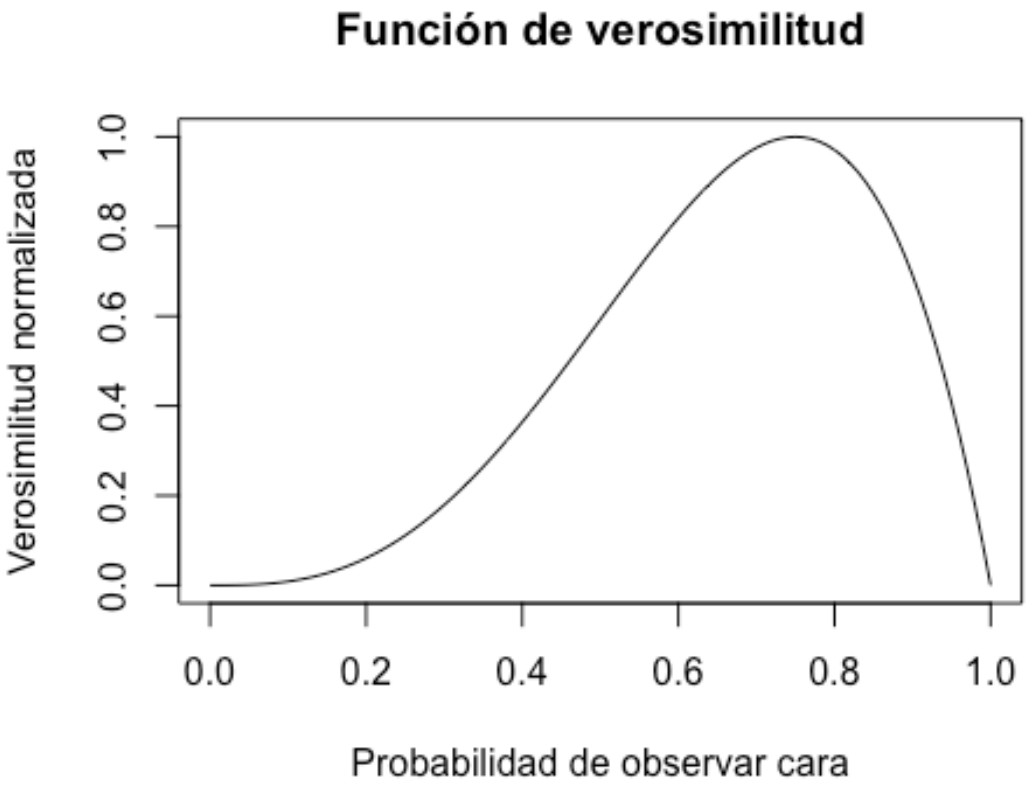

```
verologno <- plot(log(Vero375_norm)~theta, main="Función de log-
verosimilitud", xlab="Probabilidad de observar cara",
ylab="log-verosimilitud normalisada", type="l", ylim=c(-4,0))
text(x=0.2, y=-1, labels=expression(paste(italic(x), " = 375, ",
italic(n), " = 500")), cex=1.3)
```



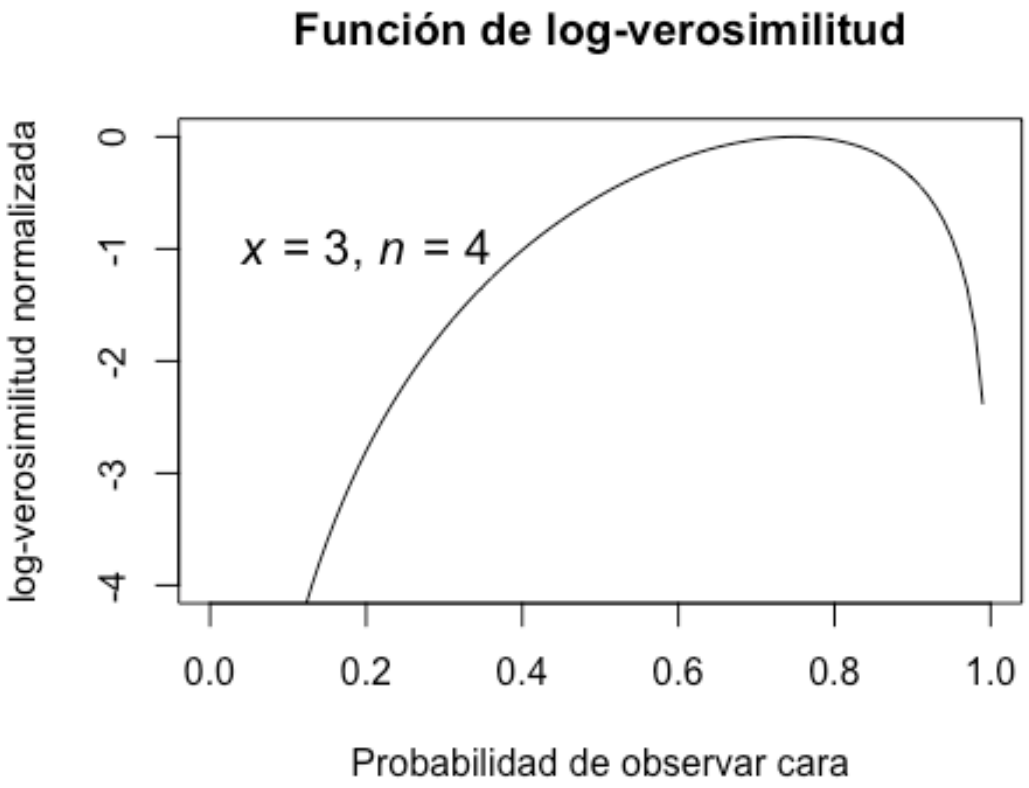

```r
x<-3
n<-4
theta<-seq(from=0, to=1, by=0.01)
Vero3<-dbinom(x=x, size=n, prob=theta)
Vero3_norm<-Vero3/max(Vero3)
v3<-plot(Vero3~theta, main="Función de verosimilitud", xlab="Probabilidad
de observar cara", ylab="Verosimilitud", type="l")
```




```
v3n <- plot(Vero3_norm~theta, main="Función de verosimilitud",
xlab="Probabilidad de observar cara", ylab="Verosimilitud normalizada",
type="l")
```




```
ve3lon <- plot(log(Vero3_norm)~theta, main="Función de log-
verosimilitud", xlab="Probabilidad de observar cara",
ylab="log-verosimilitud normalizada", type="l", ylim=c(-4,0))
text(x=0.2, y=-1, labels=expression(paste(italic(x), " = 3, ", italic(n),
" = 4")), cex=1.3)
```

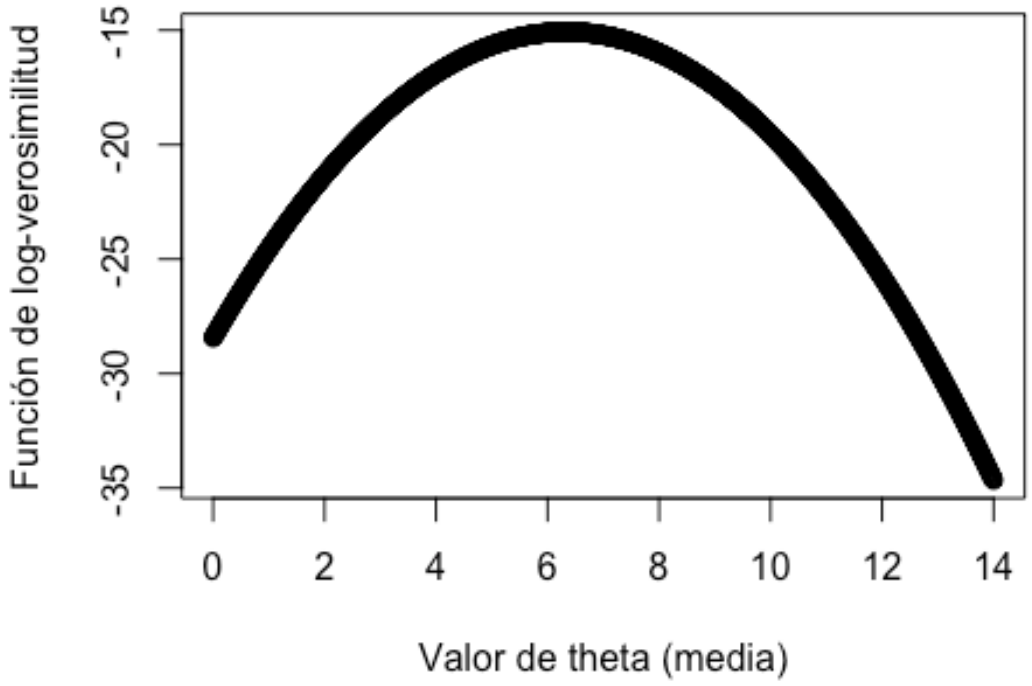


Ejemplo con distribución normal

```
x <- c(12,3,5,8,6,4)
mean(x)

## [1] 6.333333

theta2 <- seq(from=0, to=14, by=0.01)
LL1<-log(dnorm(x=12,mean=theta2,sd=3))
LL2<-log(dnorm(x=3,mean=theta2,sd=3))
LL3<-log(dnorm(x=5,mean=theta2,sd=3))
LL4<-log(dnorm(x=8,mean=theta2,sd=3))
LL5<-log(dnorm(x=6,mean=theta2,sd=3))
LL6<-log(dnorm(x=4,mean=theta2,sd=3))
log_LL <- LL1+LL2+LL3+LL4+LL5+LL6
max(log_LL)

## [1] -15.06827

theta2[which(log_LL==max(log_LL))]

## [1] 6.33
```

Graficar

```
plot(log_LL~theta2, xlab="Valor de theta (media)", ylab="Función de log-
verosimilitud")
```



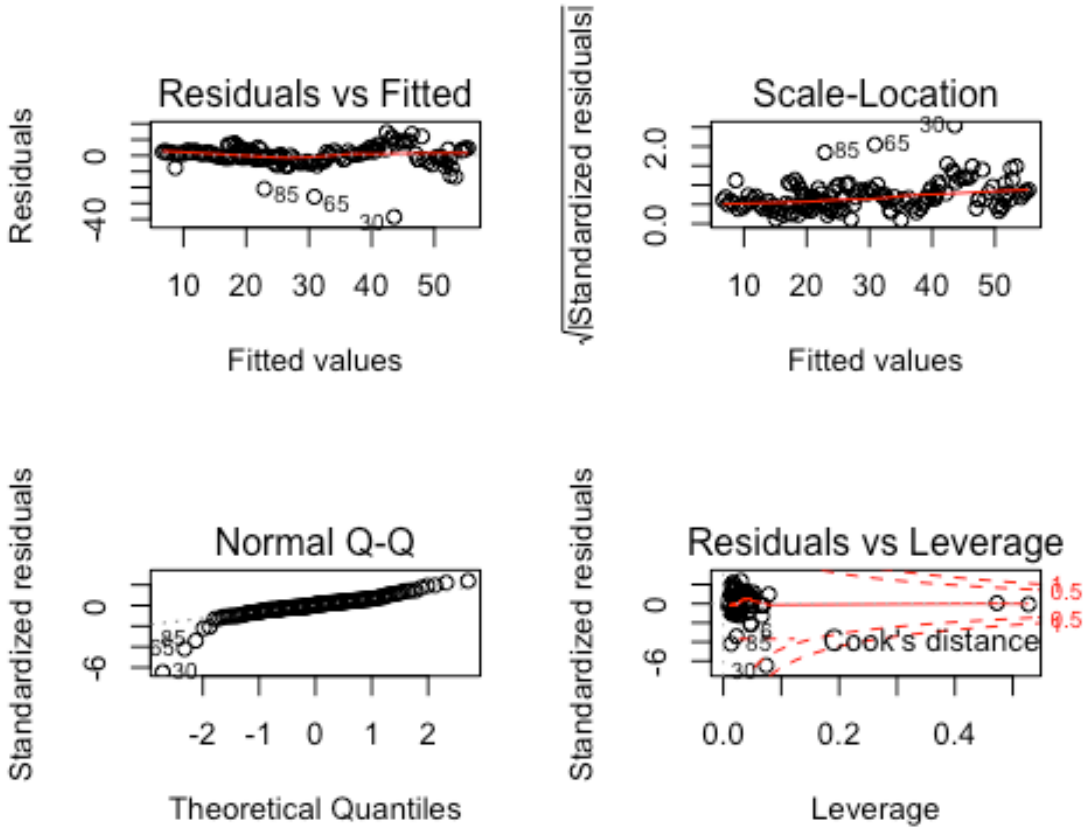
AIC: Selección de modelos e inferencia multimodelo

1. Importar datos

```
saltiempo<-read.csv("Salamandra.csv", header=TRUE)
saltiempo$Cobertura<-ifelse(saltiempo$Cob<50, 0,1)
```

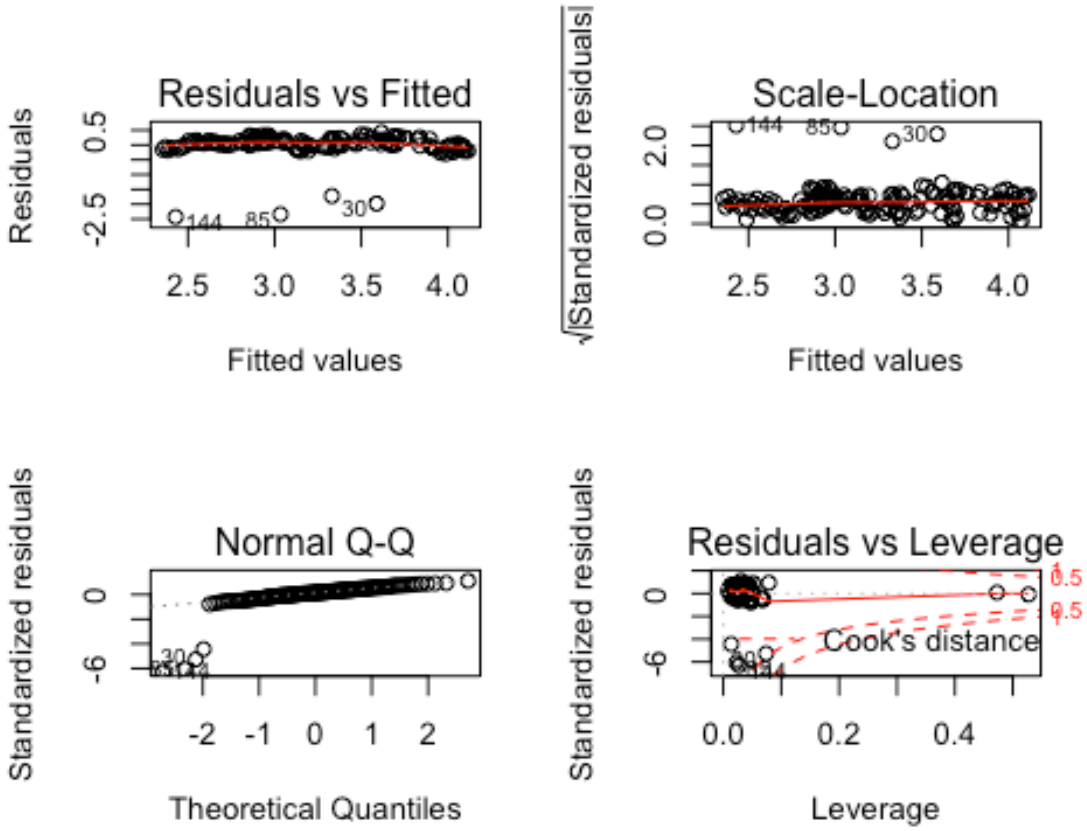
2. Verificar el ajuste del modelo global

```
mod1<-lm(Tiempo~Aire+Largo+Cobertura+Cobertura:Largo, data=saltiempo)
layout(mat=matrix(1:4, nrow=2, ncol=2))
plot(mod1)
```



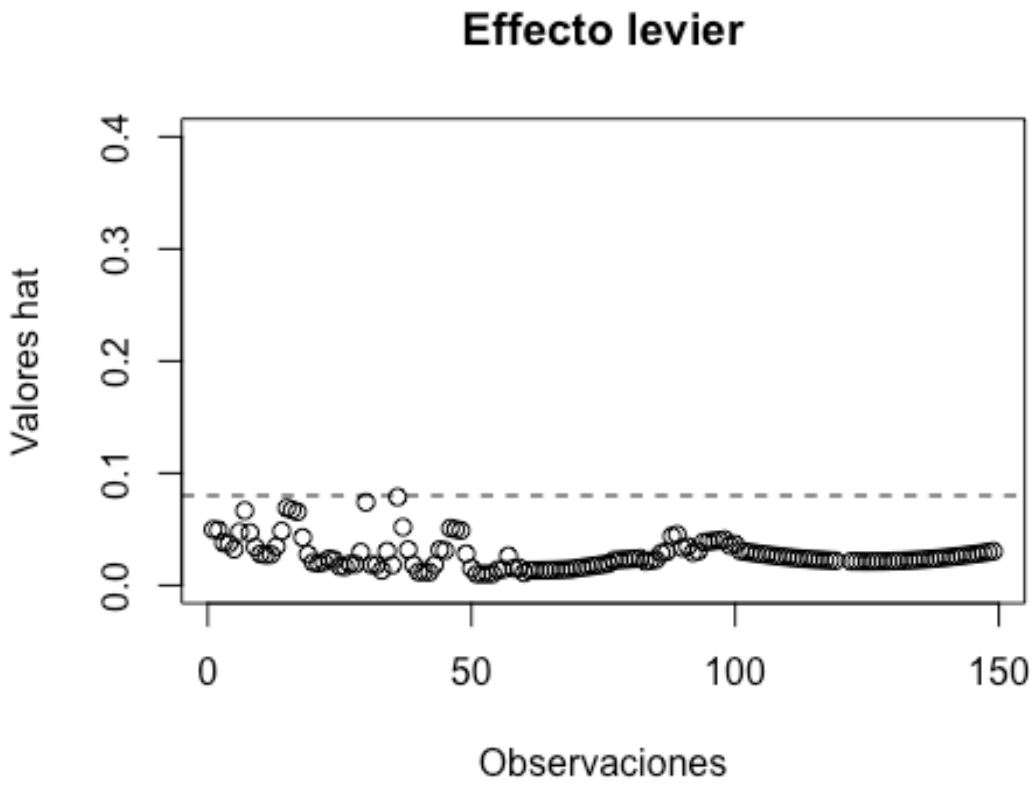
3. Ya que los residuales muestran ligeras ciertas desviaciones, intentar transformar la variable tiempo

```
mod1log<-lm(log(Tiempo)~Aire+Largo+Cobertura+Cobertura:Largo,
data=saltiempo)
layout(mat=matrix(1:4, nrow=2, ncol=2))
plot(mod1log)
```



4. Verificar influencia de observaciones

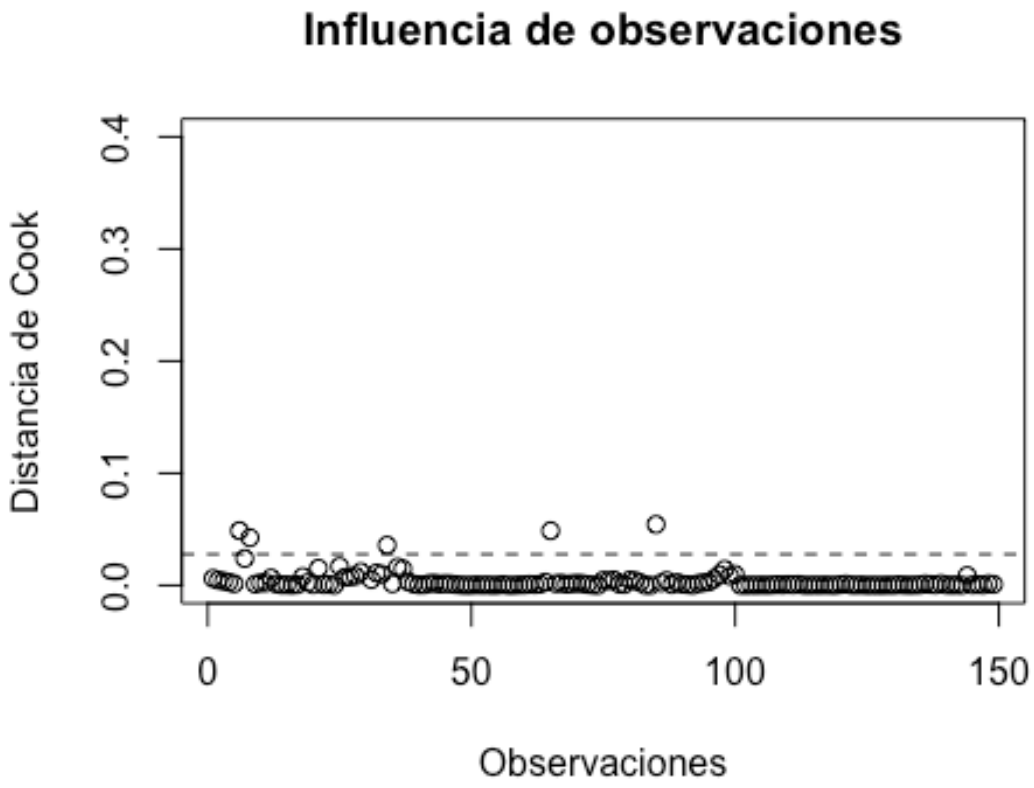
```
##checar efecto levier
#hatvalues(mod1)
plot(hatvalues(mod1), ylim=c(0,0.4), ylab="Valores hat",
xlab="Observaciones", main="Effecto levier")
abline(h=2*6/150, lty=2)

##Influencia
library(car)
```




```
plot(cookd(mod1), ylab="Distancia de Cook", xlab="Observaciones",
ylim=c(0,0.4), main="Influencia de observaciones")
```

```
## Warning: 'cookd' is deprecated.
## Use 'cooks.distance' instead.
## See help("Deprecated") and help("stats-deprecated").
```

```
abline(h=4/(150-6), lty=2) # 4/(n-p), donde n is el total de la muestra,
p es el número de parámetros
```




```
#influence.measures(mod1)
```

5. Obtener el valor de logL()

```
##La función de versoimilitud tiene su máximo con estos estimados
coefficients(mod1)

##     (Intercept)            Aire           Largo       Cobertura
##     132.4396899      -5.3205522       0.2104066      13.1442879
## Largo:Cobertura
##      -0.2107167

##El valor máximo de la función de verosimilitud
logLik(mod1)

## 'log Lik.' -480.9214 (df=6)
```

6. Correr modelos candidatos y obtener AIC

```
mod2<-lm(Tiempo~Aire, data=saltiempo)
mod3<-lm(Tiempo~Cobertura, data=saltiempo)
mod4<-lm(Tiempo~Largo, data=saltiempo)
mod5<-lm(Tiempo~Largo+Cobertura, data=saltiempo)
mod6<-lm(Tiempo~Largo+Cobertura+Cobertura:Largo, data=saltiempo)
mod7<-lm(Tiempo~Aire+Largo+Cobertura, data=saltiempo)
```

```
AICc_1<--2*logLik(mod1)[1]+2*(length(coefficients(mod1))+1)+
(2*(length(coefficients(mod1))))*(length(coefficients(mod1)+1))/(150-
length(coefficients(mod1))-1)

AICc_2<--2*logLik(mod2)[1]+2*(length(coefficients(mod2))+1)*(150/(150-
(length(coefficients(mod2))+1)-1))

AICc_3<--2*logLik(mod3)[1]+2*(length(coefficients(mod3))+1)*(150/(150-
(length(coefficients(mod3))+1)-1))

AICc_4<--2*logLik(mod4)[1]+2*(length(coefficients(mod4))+1)*(150/(150-
(length(coefficients(mod4))+1)-1))

AICc_5<--2*logLik(mod5)[1]+2*(length(coefficients(mod5))+1)*(150/(150-
(length(coefficients(mod5))+1)-1))

AICc_6<--2*logLik(mod6)[1]+2*(length(coefficients(mod6))+1)*(150/(150-
(length(coefficients(mod6))+1)-1))

AICc_7<--2*logLik(mod7)[1]+2*(length(coefficients(mod7))+1)*(150/(150-
(length(coefficients(mod7))+1)-1))
```

Generar tabla AICc

```
Resultados<-data.frame(Model=c("Largo+Aire+Cobertura+Largo:Cobertura",
"Aire", "Cobertura", "Largo", "Largo+Cobertura",
"Largo+Cobertura+Largo:Cobertura", "Aire+Largo+Cobertura"))
Resultados$AICc<-c(AICc_1, AICc_2, AICc_3, AICc_4, AICc_5, AICc_6,
AICc_7)
Resultados$Delta_AICc<-Resultados$AICc-min(Resultados$AICc)
Resultados$ModelLik<-exp(-0.5*Resultados$Delta_AICc)
Resultados$AICcPeso<-Resultados$ModelLik/sum(Resultados$ModelLik)
exp(-Resultados$Delta_AICc/2)/sum(exp(-Resultados$Delta_AICc/2))

## [1] 3.096158e-01 8.695523e-05 4.893606e-41 2.184876e-57 1.711783e-41
## [6] 6.747099e-34 6.902973e-01

AICctable<-Resultados[rev(order(Resultados$AICcPeso)),]
AICctable

##                                   Model      AICc Delta_AICc
ModelLik
## 7                 Aire+Largo+Cobertura  972.5865   0.000000
1.000000e+00
## 1 Largo+Aire+Cobertura+Largo:Cobertura  974.1901   1.603581 4.485252e-
01
## 2                                 Aire  990.5454  17.958969 1.259678e-
04
## 6      Largo+Cobertura+Largo:Cobertura 1124.6028 152.016295 9.774193e-
34
## 3                           Cobertura 1157.4813 184.894853 7.089128e-
```

```
41
## 5                       Largo+Cobertura 1159.5821 186.995641 2.479777e-
41
## 4                                 Largo 1232.7768 260.190317 3.165123e-
57
##       AICcPeso
## 7 6.902973e-01
## 1 3.096158e-01
## 2 8.695523e-05
## 6 6.747099e-34
## 3 4.893606e-41
## 5 1.711783e-41
## 4 2.184876e-57

AICctable

##                                   Model      AICc Delta_AICc
ModelLik
## 7                 Aire+Largo+Cobertura  972.5865   0.000000
1.000000e+00
## 1 Largo+Aire+Cobertura+Largo:Cobertura  974.1901   1.603581 4.485252e-
01
## 2                                  Aire  990.5454  17.958969 1.259678e-
04
## 6      Largo+Cobertura+Largo:Cobertura 1124.6028 152.016295 9.774193e-
34
## 3                           Cobertura 1157.4813 184.894853 7.089128e-
41
## 5                       Largo+Cobertura 1159.5821 186.995641 2.479777e-
41
## 4                                 Largo 1232.7768 260.190317 3.165123e-
57
##       AICcPeso
## 7 6.902973e-01
## 1 3.096158e-01
## 2 8.695523e-05
## 6 6.747099e-34
## 3 4.893606e-41
## 5 1.711783e-41
## 4 2.184876e-57
```

7. Inferencia multimodelo

```
ResultadosA<-data.frame(Model=c("Largo+Aire+Cobertura+Largo:Cobertura",
"Aire", "Aire+Largo+Cobertura"))
ResultadosA$AICc<-c(AICc_1, AICc_2, AICc_7)
ResultadosA$Delta_AICc<-ResultadosA$AICc-min(ResultadosA$AICc)
ResultadosA$AICcPeso<-exp(-ResultadosA$Delta_AICc/2)/sum(exp(-
ResultadosA$Delta_AICc/2))
AICcAtable<-ResultadosA[rev(order(ResultadosA$AICcPeso)),]
AICcAtable
```

```
##                                    Model     AICc Delta_AICc
AICcPeso
## 3                 Aire+Largo+Cobertura 972.5865   0.000000 6.902973e-
01
## 1 Largo+Aire+Cobertura+Largo:Cobertura 974.1901   1.603581 3.096158e-
01
## 2                                  Aire 990.5454  17.958969 8.695523e-
05
```

8.Calcular estimación promedio para la variable aire

```
aire.ests <- c(coef(mod1)[2], coef(mod2)[2], coef(mod7)[2])
aire.ests
```

```
##      Aire      Aire      Aire
## -5.320552 -4.149320 -5.212568
```

```
mod.avg.est <- sum(ResultadosA$AICcPeso*aire.ests)
mod.avg.est
```

```
## [1] -5.245909
```

9.Calcular SE incondicional para la estimación promedio

```
aire.se <- c(summary(mod1)$coef[2, 2], summary(mod2)$coef[2, 2],
             summary(mod7)$coef[2, 2])
aire.se
```

```
## [1] 0.3323302 0.1682901 0.2706232
```

```
incond.se <- sum(ResultadosA$AICcPeso*sqrt((aire.se^2) + ((aire.ests -
mod.avg.est)^2)))
incond.se
```

```
## [1] 0.2937775
```

10.Calcular intervalos de confianza 95%

```
incond.95CI <- mod.avg.est+1.96*c(-1*incond.se, incond.se)
incond.95CI
```

```
## [1] -5.821713 -4.670106
```