Solución Ejercicios Tema 1. Regresión lineal simple

Máster en Ciencia de Datos. Módulo: Análisis exploratorio de datos

Ana Navarro Quiles

Curso 2022/2023

Ejercicio 1

Utilizando el banco de datos **deportistas**, considerad la variable respuesta *Peso* relacionandola con el predictor *PrctGrasa*.

```
load('datosTema1.Rdata')
```

a) ¿Cuánto vale la pendiente de la recta? ¿Podemos afirmar que es positiva?

```
reg <- lm(Peso ~ PrctGrasa, data=deportistas)
coef(reg)
```

```
##   (Intercept)     PrctGrasa
## 75.0137910874 -0.0004345976
```

```
summary(reg)
```

```
##
## Call:
## lm(formula = Peso ~ PrctGrasa, data = deportistas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.210  -8.482  -0.608   9.116  48.192
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 75.0137911  2.3625769  31.751   <2e-16 ***
## PrctGrasa   -0.0004346  0.1590772  -0.003    0.998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.96 on 200 degrees of freedom
## Multiple R-squared:  3.732e-08,  Adjusted R-squared:  -0.005
## F-statistic: 7.464e-06 on 1 and 200 DF,  p-value: 0.9978
```

b) Compara la varianza de la variable respuesta con la varianza de los residuos: ¿Qué porcentaje de la variabilidad inicial está explicado por la recta de mínimos cuadrados? ¿Qué porcentaje de la variabilidad inicial falta todavía por explicar?

```
y<-deportistas$Peso
var(y)#varianza de la variable respuesta, es lo mismo que var(y)
```

```
## [1] 193.9112
```

```r
residuos <- residuals(reg)
var(residuos) #varianza de los residuos
```

```
## [1] 193.9112
```

```r
# Es lo mismo que (summary(reg)$sigma)^2


R2<-1-var(residuos)/var(y)      # bondad del ajuste. Es lo mismo que summary(reg)$r.squared
R2*100 # Está explicado.
```

```
## [1] 3.731889e-06
```

```r
(1-R2)*100 # Falta por explicar
```

```
## [1] 100
```

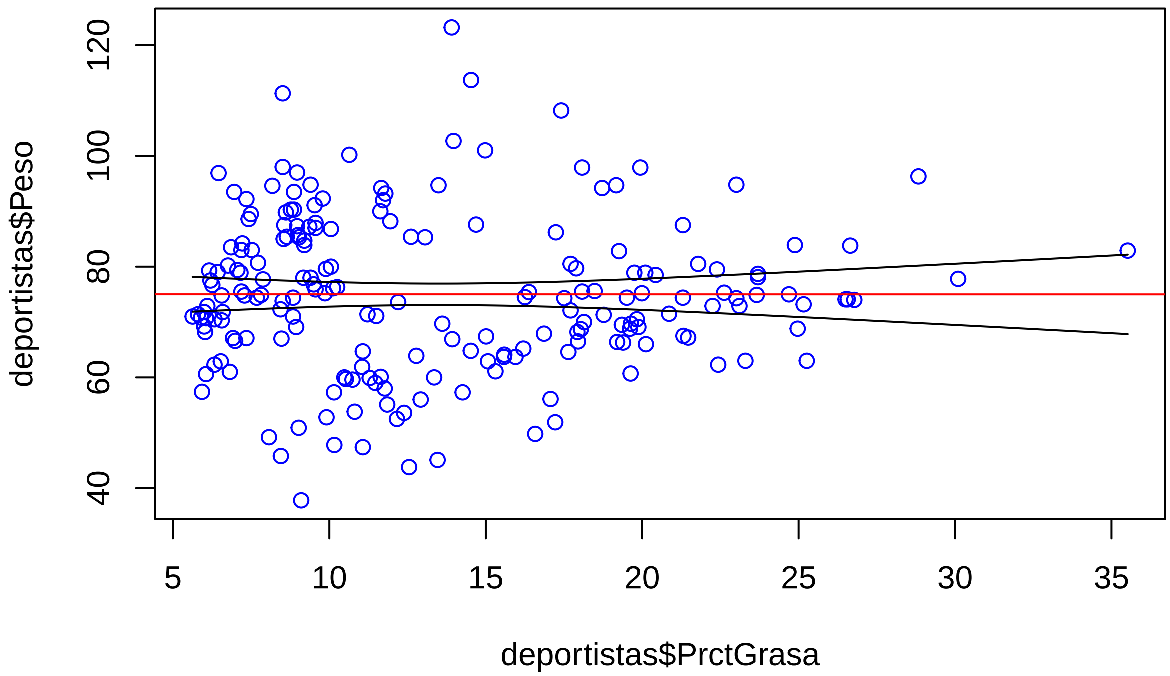
c) Obtén los intervalos de confianza al 95% sobre los parámetros de la recta.

```r
confint(reg)
```

```
##                  2.5 %     97.5 %
## (Intercept) 70.3550347 79.6725475
## PrctGrasa   -0.3141184  0.3132492
```

d) Dibuja el diagrama de dispersión, la recta de regresión y las bandas de confianza para la estimación al 95%.

```r
#Obtención de bandas de estimación:
minx<-range(deportistas$PrctGrasa)[1]; maxx<-range(deportistas$PrctGrasa)[2]
nuevos <- data.frame(list(PrctGrasa = seq(minx,maxx,length=100)))
bandas_est<-predict(reg, newdata = nuevos, interval = "confidence")

#Representación gráfica:
plot(deportistas$PrctGrasa,deportistas$Peso, col='BLUE')
abline(coef=coef(reg), col='RED')
lines(nuevos$PrctGrasa,bandas_est[,2],col='BLACK')
lines(nuevos$PrctGrasa,bandas_est[,3],col='BLACK')
```



e) Si te parece adecuado estima el peso correspondiente a nuevos individuos con los siguientes porcentajes de grasa: $25, 50, 75\%$. Calcula sus respectivos intervalos de confianza al 95%.

Ejercicio 2

Repite el ejercicio anterior considerando IMC en lugar de peso y compara los resultados con los del ejercicio anterior.

```
reg <- lm(IMC ~ PrctGrasa, data=deportistas)
coef(reg)
```

```
## (Intercept)   PrctGrasa
## 21.78371732  0.08677995
```

```
summary(reg)
```

```
##
## Call:
## lm(formula = IMC ~ PrctGrasa, data = deportistas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7340 -2.0182 -0.1511  1.4287 11.4292
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.78372    0.47728   45.64  < 2e-16 ***
## PrctGrasa    0.08678    0.03214    2.70  0.00752 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.82 on 200 degrees of freedom
## Multiple R-squared:  0.03518,    Adjusted R-squared:  0.03035
## F-statistic: 7.292 on 1 and 200 DF,  p-value: 0.007519
```

```r
y<-deportistas$IMC
var(y)
```

```
## [1] 8.202111
```

```r
residuos <- residuals(reg)
var(residuos)
```

```
## [1] 7.913578
```

```r
R2<-1-var(residuos)/var(y)      # bondad del ajuste
R2*100 # Está explicado
```

```
## [1] 3.517791
```

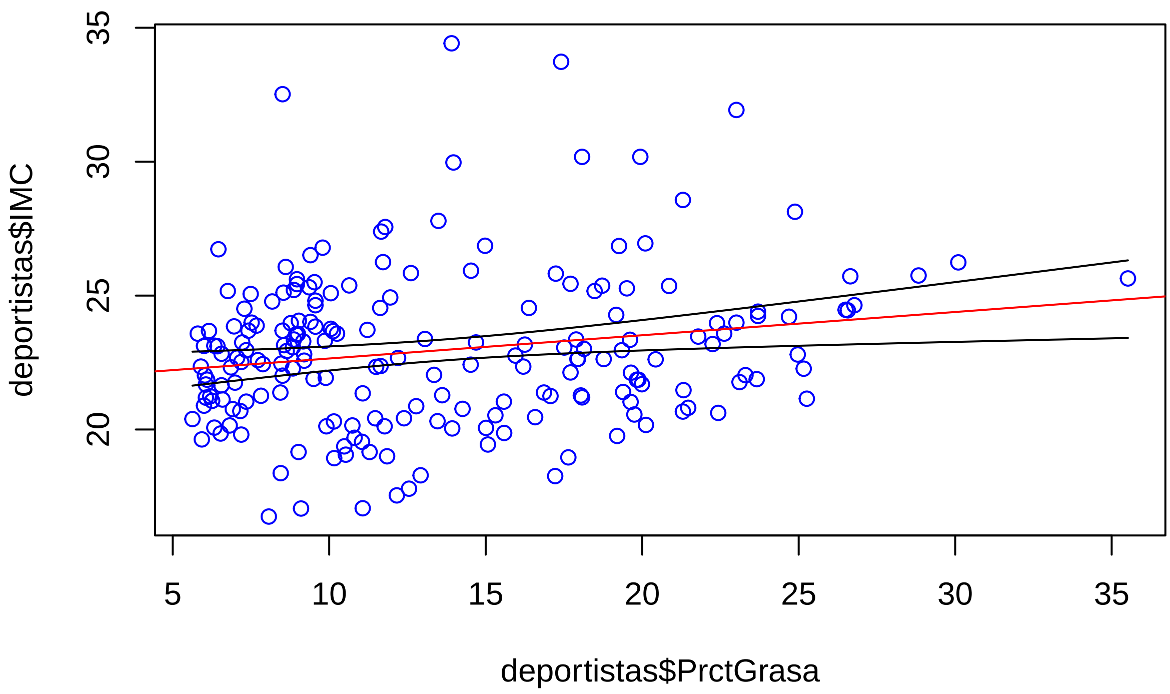
```r
(1-R2)*100 # Falta por explicar
```

```
## [1] 96.48221
```

```r
confint(reg)
```

```
##                    2.5 %    97.5 %
## (Intercept) 20.84257560 22.724859
## PrctGrasa    0.02341092  0.150149
```

```r
#Obtención de bandas de estimación:
minx<-range(deportistas$PrctGrasa)[1]; maxx<-range(deportistas$PrctGrasa)[2]
nuevos <- data.frame(list(PrctGrasa = seq(minx,maxx,length=100)))
bandas_est<-predict(reg, newdata = nuevos, interval = "confidence")

#Representación gráfica:
plot(deportistas$PrctGrasa,deportistas$IMC, col='BLUE')
abline(coef=coef(reg), col='RED')
lines(nuevos$PrctGrasa,bandas_est[,2],col='BLACK')
lines(nuevos$PrctGrasa,bandas_est[,3],col='BLACK')
```



```r
# Aunque la regresión no es muy buena, vamos a obtener las predicciones que se indican
valores <- data.frame(list(PrctGrasa = c(25,50,75)))
bandas_est<-predict(reg, newdata = valores, interval = "confidence")
bandas_est
```

```
##        fit      lwr      upr
## 1 23.95322 23.12649 24.77994
## 2 26.12271 23.77735 28.46808
## 3 28.29221 24.37589 32.20853
```

Ejercicio 3

Utilizando el banco de datos **deportistas.csv**, considerad la variable respuesta *PrctGrasa* relacionándola con el predictor *MCMagra*.

a) Obtén la recta mínimos cuadrados utilizando todos los datos, sin tener en cuenta el *sexo*

```r
reg <- lm(PrctGrasa ~ MCMagra, data=deportistas)
summary(reg)
```

```
##
## Call:
## lm(formula = PrctGrasa ~ MCMagra, data = deportistas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.636 -4.459 -1.439  4.434 20.057
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.62459    2.06573  11.921  < 2e-16 ***
## MCMagra     -0.17137    0.03122  -5.489 1.21e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.785 on 200 degrees of freedom
## Multiple R-squared:  0.1309, Adjusted R-squared:  0.1266
## F-statistic: 30.13 on 1 and 200 DF,  p-value: 1.214e-07
```

b) Evalua el efecto del *Genero* sobre *PrctGrasa*

```r
hombres <- subset(deportistas, Genero=='male')
mujeres <- subset(deportistas, Genero=='female')
cor(subset(hombres, select=c(PrctGrasa,MCMagra)))
```

```
##           PrctGrasa   MCMagra
## PrctGrasa 1.0000000 0.3704125
## MCMagra   0.3704125 1.0000000
```

```r
cor(subset(mujeres, select=c(PrctGrasa,MCMagra)))
```

```
##           PrctGrasa   MCMagra
## PrctGrasa 1.0000000 0.4061823
## MCMagra   0.4061823 1.0000000
```

```r
regsexo<-lm(PrctGrasa ~ Genero, data=deportistas)
summary(regsexo)
```

```
##
## Call:
## lm(formula = PrctGrasa ~ Genero, data = deportistas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.779 -2.713 -0.360  2.251 17.671
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.8491     0.4454   40.07   <2e-16 ***
## Generomale   -8.5982     0.6268  -13.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.454 on 200 degrees of freedom
## Multiple R-squared:  0.4847, Adjusted R-squared:  0.4822
## F-statistic: 188.2 on 1 and 200 DF,  p-value: < 2.2e-16
```

c) Obtén ahora una recta para los *hombres* y otra para las *mujeres*

```r
regh <- lm(PrctGrasa ~ MCMagra, data=hombres)
summary(regh)
```

```
##
## Call:
## lm(formula = PrctGrasa ~ MCMagra, data = hombres)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7408 -1.9952 -0.7775  0.9771 10.2902
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.34331    2.25303   0.152 0.879196
## MCMagra      0.11931    0.02992   3.988 0.000127 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.973 on 100 degrees of freedom
## Multiple R-squared:  0.1372, Adjusted R-squared:  0.1286
## F-statistic:  15.9 on 1 and 100 DF,  p-value: 0.000127
```

```r
regm <- lm(PrctGrasa ~ MCMagra, data=mujeres)
summary(regm)
```

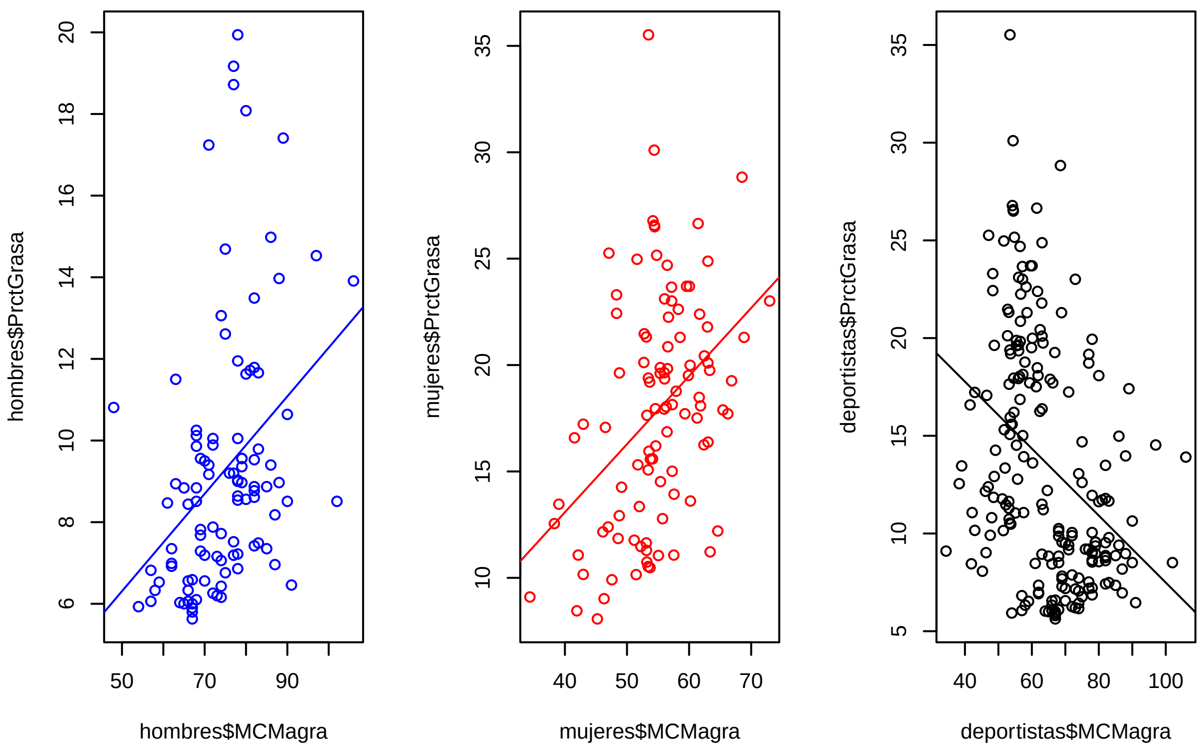
```
##
## Call:
## lm(formula = PrctGrasa ~ MCMagra, data = mujeres)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.347 -3.578 -0.417  3.050 18.130
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.28459    4.02292   0.071    0.944
## MCMagra      0.31997    0.07271   4.400 2.75e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.008 on 98 degrees of freedom
## Multiple R-squared:  0.165,  Adjusted R-squared:  0.1565
## F-statistic: 19.36 on 1 and 98 DF,  p-value: 2.753e-05
```

```
# En ambos casos podriamos haber calculado la recta forzando el intercepto a 0.
```

d) Dibuja en la misma gráfica las tres rectas y comenta los resultados

```
par(mfcol=c(1,3))
plot(hombres$MCMagra,hombres$PrctGrasa, col='BLUE')
abline(coef=coef(regh), col='BLUE')
plot(mujeres$MCMagra,mujeres$PrctGrasa, col='RED')
abline(coef=coef(regm), col='RED')
plot(deportistas$MCMagra,deportistas$PrctGrasa, col='BLACK')
abline(coef=coef(reg), col='BLACK')
```



Ejercicio 4

En la base **deportistas**,

a) Evalúa mediante regresión lineal si el *PrctGrasa* explica los resultados análiticos: *Hematocrito*, *Ferritina* y *Hemoglobina*.

```
reg1 <- lm(Hematocrito ~ PrctGrasa, data=deportistas)
summary(reg1)
```

```
##
## Call:
```

```
## lm(formula = Hematocrito ~ PrctGrasa, data = deportistas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4803 -2.1450  0.1412  1.9195 15.3646
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 47.34765    0.52605  90.006  < 2e-16 ***
## PrctGrasa   -0.31509    0.03542  -8.896 3.47e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.108 on 200 degrees of freedom
## Multiple R-squared:  0.2835, Adjusted R-squared:  0.2799
## F-statistic: 79.14 on 1 and 200 DF,  p-value: 3.474e-16
```

```
reg2 <- lm(Ferritina ~ PrctGrasa, data=deportistas)
summary(reg2)
```

```
##
## Call:
## lm(formula = Ferritina ~ PrctGrasa, data = deportistas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -77.77 -33.04  -9.98  19.16 153.30
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  95.8855     7.9225  12.103  < 2e-16 ***
## PrctGrasa    -1.4073     0.5334  -2.638  0.00899 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.81 on 200 degrees of freedom
## Multiple R-squared:  0.03363,    Adjusted R-squared:  0.0288
## F-statistic:  6.96 on 1 and 200 DF,  p-value: 0.00899
```

```
reg3 <- lm(Hemoglobina ~ PrctGrasa, data=deportistas)
summary(reg3)
```

```
##
## Call:
## lm(formula = Hemoglobina ~ PrctGrasa, data = deportistas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8025 -0.7513 -0.0446  0.7636  4.1718
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.14663    0.19580  82.465  < 2e-16 ***
## PrctGrasa   -0.11699    0.01318  -8.874 3.99e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.157 on 200 degrees of freedom
## Multiple R-squared:  0.2825, Adjusted R-squared:  0.2789
## F-statistic: 78.75 on 1 and 200 DF,  p-value: 3.993e-16
```

b) Evalúa mediante regresión lineal si *peso* y *altura* explican el *PrctGrasa*.

```r
reg4 <- lm(PrctGrasa ~ Peso, data=deportistas)
summary(reg4)
```

```
##
## Call:
## lm(formula = PrctGrasa ~ Peso, data = deportistas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.878 -4.962 -1.857  4.574 22.013
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.351e+01  2.398e+00   5.636 5.86e-08 ***
## Peso        -8.587e-05  3.143e-02  -0.003    0.998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.205 on 200 degrees of freedom
## Multiple R-squared:  3.732e-08,  Adjusted R-squared:  -0.005
## F-statistic: 7.464e-06 on 1 and 200 DF,  p-value: 0.9978
```

```r
reg5 <- lm(PrctGrasa ~ Altura, data=deportistas)
summary(reg5)
```

```
##
## Call:
## lm(formula = PrctGrasa ~ Altura, data = deportistas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.763 -4.560 -2.450  4.646 21.976
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.04001    7.96504   4.399 1.77e-05 ***
## Altura      -0.11956    0.04416  -2.707  0.00737 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.095 on 200 degrees of freedom
## Multiple R-squared:  0.03535,    Adjusted R-squared:  0.03053
## F-statistic:  7.33 on 1 and 200 DF,  p-value: 0.00737
```

c) Evalúa la relación entre *IMC* y las variables *SumPliegues* y *PrctGrasa*

```r
cor(subset(deportistas,select=c(IMC,SumPliegues,PrctGrasa)))
```

```
##                   IMC SumPliegues PrctGrasa
```

```
## IMC         1.0000000   0.3211164 0.1875578
## SumPliegues 0.3211164   1.0000000 0.9630168
## PrctGrasa   0.1875578   0.9630168 1.0000000
```

Ejercicio 5

Utilizando el modelo $Y = 25 + 2X + \epsilon$, siendo ϵ Normal con media 0 varianza $\sigma^2 = 4$, simula $N = 10000$ muestras de tamaño $n = 50$. Para ello, utiliza valores de X simulados de una Uniforme definida en el intervalo $(0, 5)$. A continuación, para cada una de las muestras simuladas, obtén el intervalo de confianza al 95% sobre la pendiente de la recta. ¿Qué porcentaje de intervalos no contienen al verdadero valor de la pendiente?

Vuelve a calcular ese porcentaje, pero ahora simulando ϵ de forma que $\epsilon/8$ sea t-Student con 4 grados de libertad. ¿Cómo afecta la falta de normalidad a la fiabilidad de ese intervalo?

El objetivo del problema es ver que el modelo de regresión es bastante robusto ante no normalidad de residuos. Podéis probar con otras distribuciones, como la exponencial.

```
ICbeta1<-function(){
n<-50
x<-runif(n,0,5)
epsilon<-rnorm(n,0,2)
y<-25+2*x+epsilon
mod<-lm(y~x)
return(confint(mod)[2,])
}
```

```
N<-10000
ICsim<-t(replicate(N, ICbeta1()))
(1-sum(ICsim[,1]<=2 & ICsim[,2]>=2)/N)*100 # 1 menos la suma de los que si que contienen al 2.
```

```
## [1] 5.13
```

```
ICbeta1_2<-function(){
n<-50
x<-runif(n,0,5)
epsilon<-8*rt(n,4)
y<-25+2*x+epsilon
mod<-lm(y~x)
return(confint(mod)[2,])
}
```

```
N<-10000
ICsim<-t(replicate(N, ICbeta1_2()))
(1-sum(ICsim[,1]<=2 & ICsim[,2]>=2)/N)*100 # 1 menos la suma de los que si que contienen al 2.
```

```
## [1] 5.2
```

Ejercicio 6

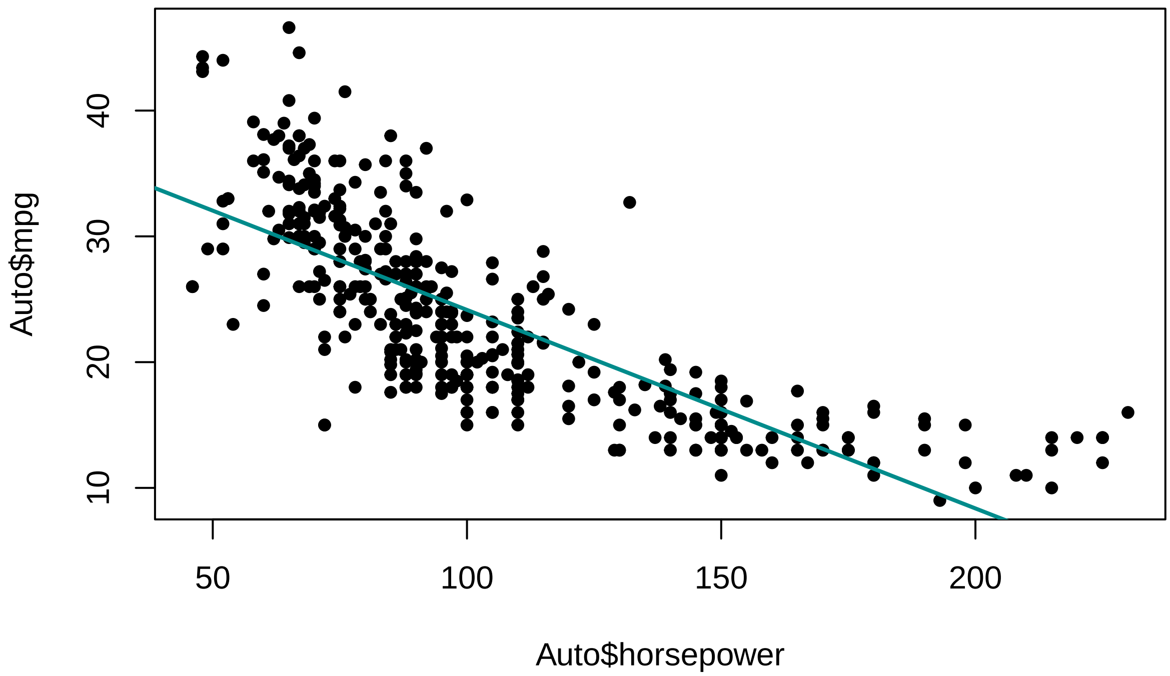
Utilizando el banco de datos **Auto**, en el paquete de R **ISLR**, se desea explicar el consumo de carburante, variable *mpg*, a partir de la potencia del motor, variable *horsepower*.

a) Dibuja el diagrama de dispersión y la recta de mínimos cuadrados.

```
library(ISLR)
```

```
mod1<-lm(mpg~horsepower,data=Auto)
```

```r
plot(Auto$horsepower, Auto$mpg, type="p", pch=19,cex=0.75)
abline(coef=coef(mod1),col="darkcyan",lwd=2)
```



b) ¿Hay relación entre esas dos variables? ¿Cómo de fuerte es esa relación? ¿Podemos afirmar si es positiva o negativa?

```r
summary(mod1)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
cor(Auto$horsepower,Auto$mpg)
```

```
## [1] -0.7784268
```

c) ¿Qué consumo se espera si potencia del motor es 75? Proporciona el intervalo de confianza y el de predicción para esa potencia de motor.

```
predict.lm(mod1,newdata=data.frame(horsepower=75), se=T)
```

```
## $fit
##        1
## 28.09751
##
## $se.fit
## [1] 0.3122068
##
## $df
## [1] 390
##
## $residual.scale
## [1] 4.905757
```

```
predict(mod1, newdata = data.frame(horsepower=75), interval = "prediction")  #banda de error pred de un
```

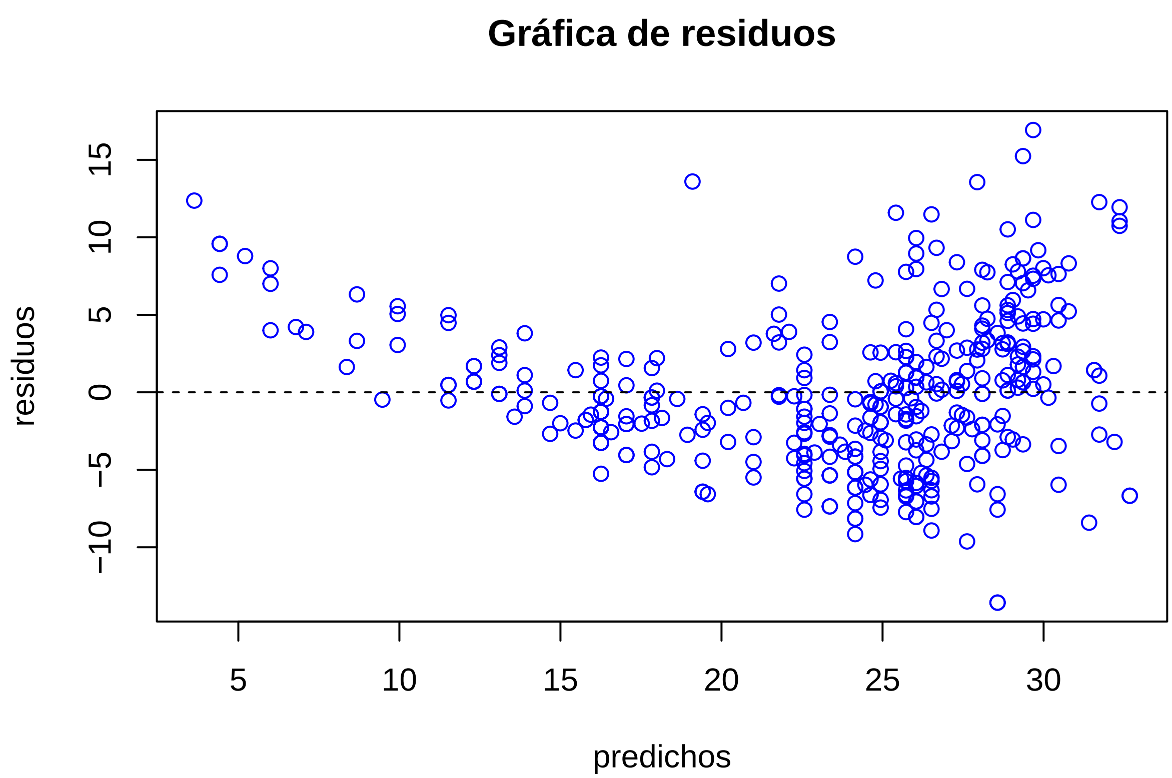
```
##        fit      lwr      upr
## 1 28.09751 18.43296 37.76206
```

```
predict(mod1, newdata = data.frame(horsepower=75), interval = "confidence")  #banda de error de estimac
```

```
##        fit      lwr      upr
## 1 28.09751 27.48369 28.71133
```

d) Analiza gráficamente los residuos y comenta los resultados.

```
residuos <- residuals(mod1)
predichos <- fitted.values(mod1)
plot(predichos,residuos, col='BLUE',main = 'Gráfica de residuos')
abline(h=0,lty=2)
```



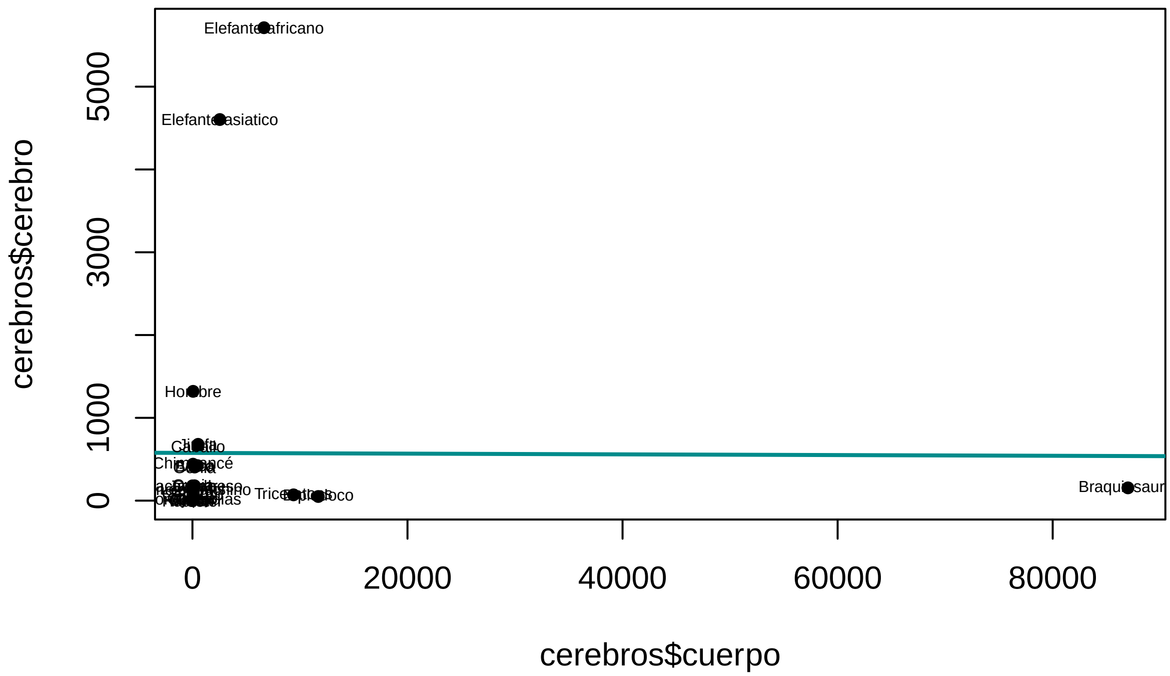
Vemos un ejemplo de no linealidad (al principio tampoco homocedasticidad)

Ejercicio 7

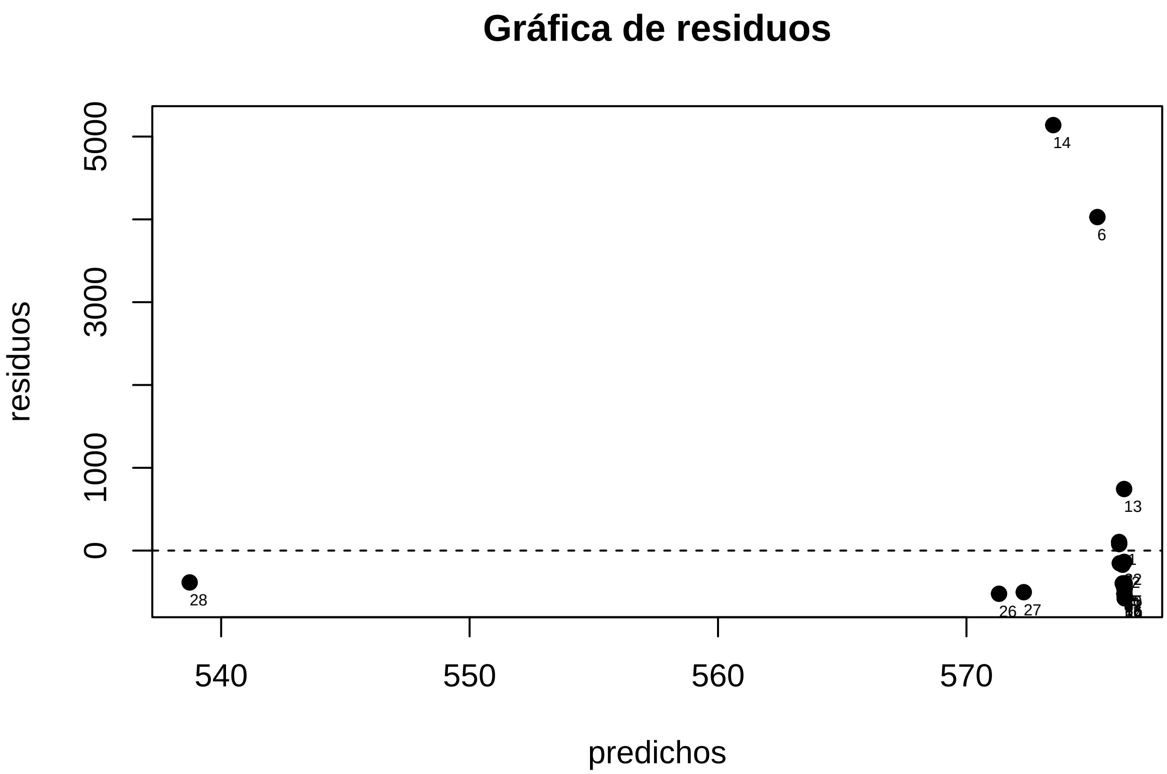
El banco de datos **cerebros** es un banco de datos famoso. En él se recogen los pesos del cuerpo y del cerebro de diversos animales. Vamos a explicar el peso del cerebro (en g) *cerebro* a partir del peso del cuerpo (en Kg) *cuerpo*.

a) Ajusta el modelo y realiza el diagnóstico del modelo.

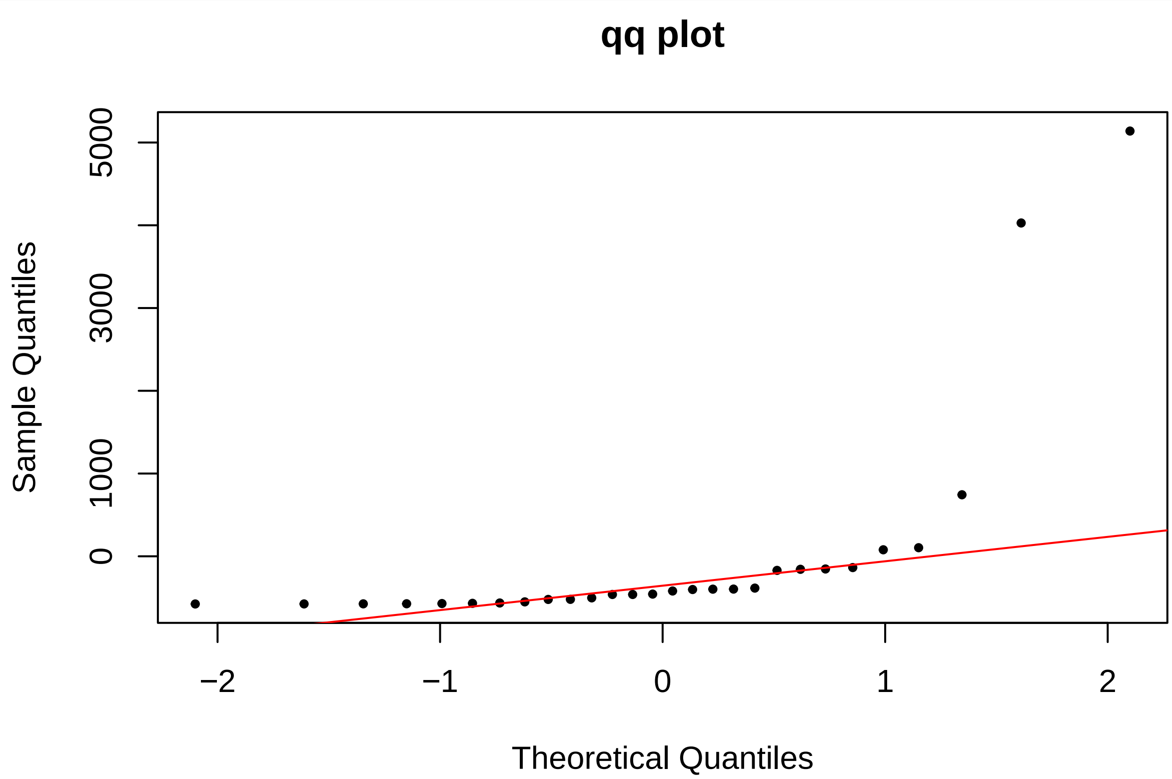
```
mod1<-lm(cerebro~cuerpo, data=cerebros, na.action=na.exclude)
plot(cerebros$cuerpo, cerebros$cerebro, type="p", pch=19,cex=0.75)
abline(coef=coef(mod1),col="darkcyan",lwd=2)
text(cerebros$cuerpo, cerebros$cerebro, labels=cerebros$Nombre,cex=0.5)
```




```
# Dibujamos residuos vs predichos para realizar el diagnóstico del modelo
residuos <- residuals(mod1)
predichos <- fitted.values(mod1)
plot(predichos,residuos, pch=19, main = 'Gráfica de residuos')
text(predichos,residuos, labels=rownames(cerebros),cex=0.5,adj=c(0,2))

# Normalidad
abline(h=0,lty=2)
```



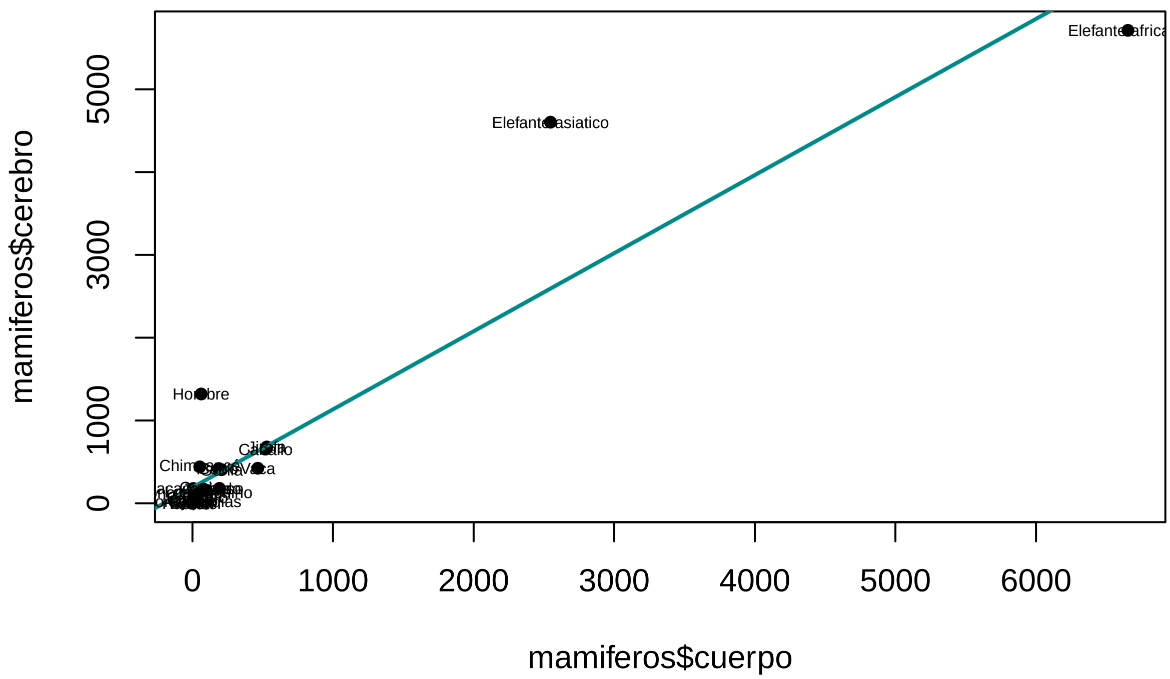

```r
qqnorm(residuos, pch=19, cex=0.5,main = 'qq plot')
qqline(residuos,col="red")
```



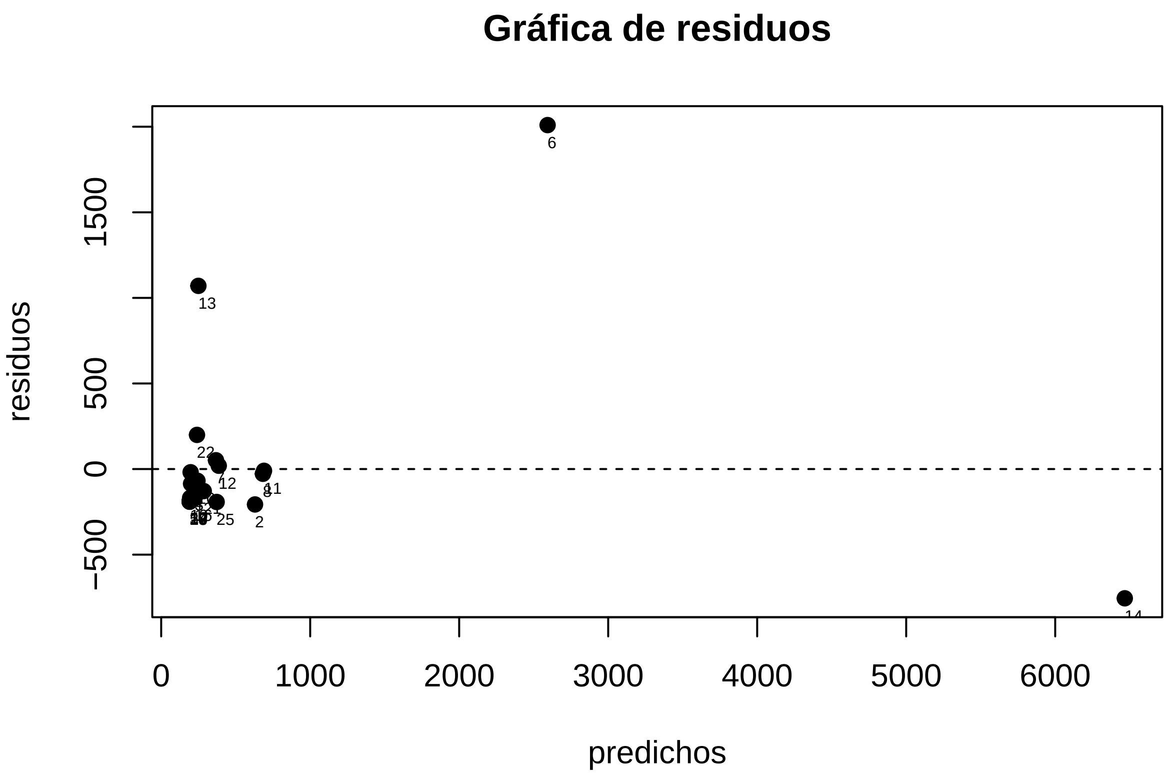
```r
# Observamos la falta de linealidad y homocedasticidad
```

b) Retira los datos que no pertenecen a la misma población que el resto y re-analiza.

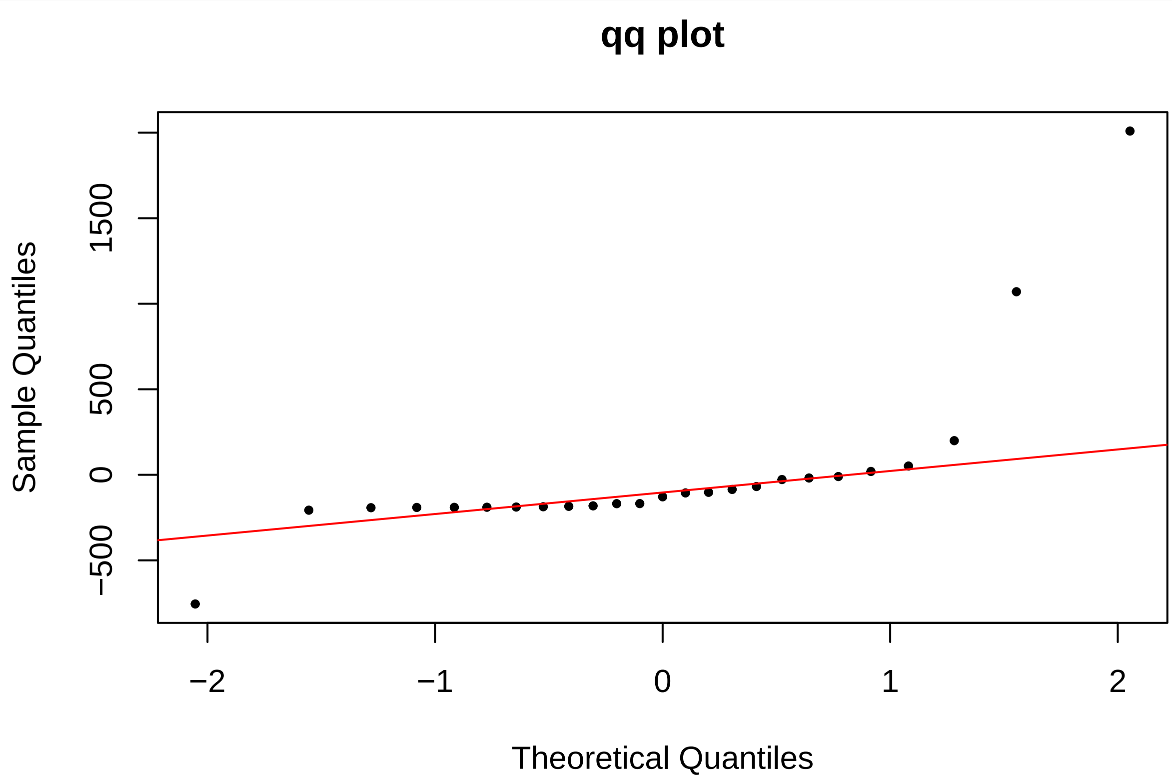
```r
# Indica los que no pertenecen a la misma población. En este caso el 26, 27 y 28
# son dinosaurios. Si nos preguntaran retirar aquellos datos influyentes habríamos
# quitado el 6, 14 y 28
mamiferos<-cerebros[-c(26,27,28),]
mod2<-lm(cerebro~cuerpo, data=mamiferos, na.action=na.exclude)
plot(mamiferos$cuerpo, mamiferos$cerebro, type="p", pch=19,cex=0.75)
abline(coef=coef(mod2),col="darkcyan",lwd=2)
text(mamiferos$cuerpo, mamiferos$cerebro, labels=mamiferos$Nombre,cex=0.5)
```




```r
# Dibujamos residuos vs predichos para realizar el diagnóstico del modelo
residuos <- residuals(mod2)
predichos <- fitted.values(mod2)
plot(predichos,residuos, pch=19, main = 'Gráfica de residuos')
text(predichos,residuos, labels=rownames(mamiferos),cex=0.5,adj=c(0,2))

# Normalidad
abline(h=0,lty=2)
```




```r
qqnorm(residuos, pch=19, cex=0.5,main = 'qq plot')
qqline(residuos,col="red")
```

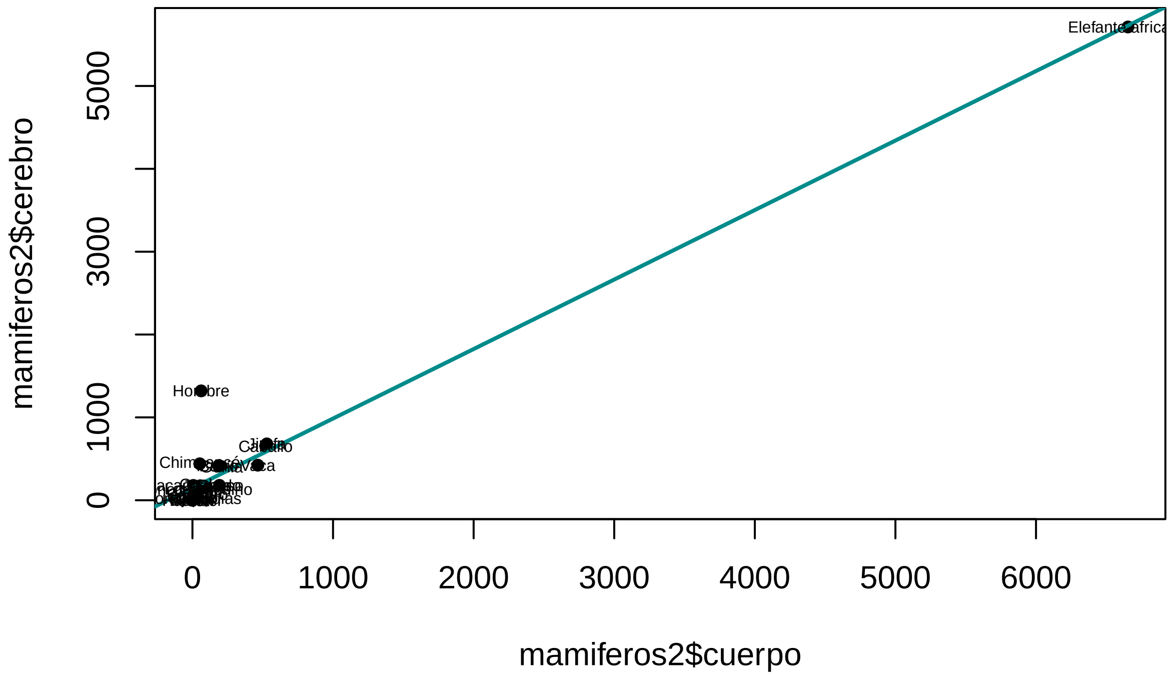



```r
# Aunque no lo pide, vamos a mejorar el modelo de mamíferos. Vemos que el 6
# es influtente
```

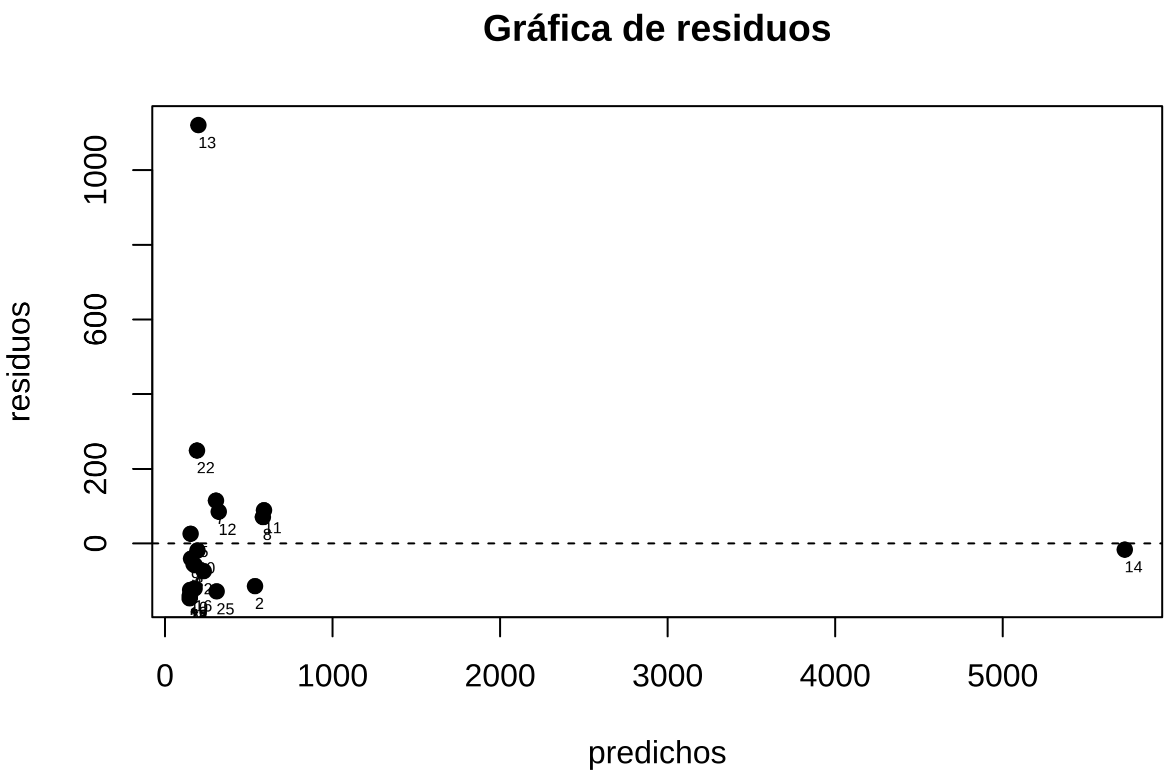
```
mamiferos[c(6),]
```

```
##              Nombre cerebro cuerpo
## 6 Elefante asiatico    4603   2547
```

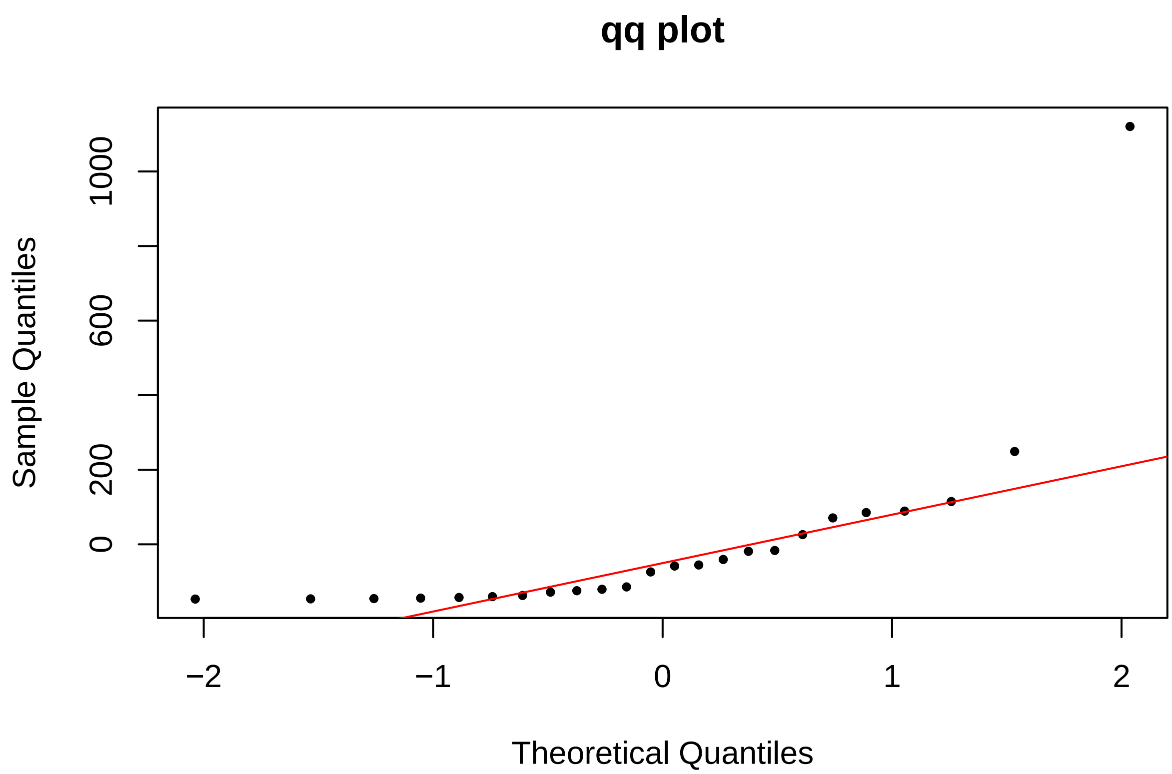
```
mamiferos2<-mamiferos[-c(6),]
mod3<-lm(cerebro~cuerpo, data=mamiferos2, na.action=na.exclude)
plot(mamiferos2$cuerpo, mamiferos2$cerebro, type="p", pch=19,cex=0.75)
abline(coef=coef(mod3),col="darkcyan",lwd=2)
text(mamiferos2$cuerpo, mamiferos2$cerebro, labels=mamiferos2$Nombre,cex=0.5)
```




```
# Dibujamos residuos vs predichos para realizar el diagnóstico del modelo
residuos <- residuals(mod3)
predichos <- fitted.values(mod3)
plot(predichos,residuos, pch=19, main = 'Gráfica de residuos')
text(predichos,residuos, labels=rownames(mamiferos2),cex=0.5,adj=c(0,2))

# Normalidad
abline(h=0,lty=2)
```



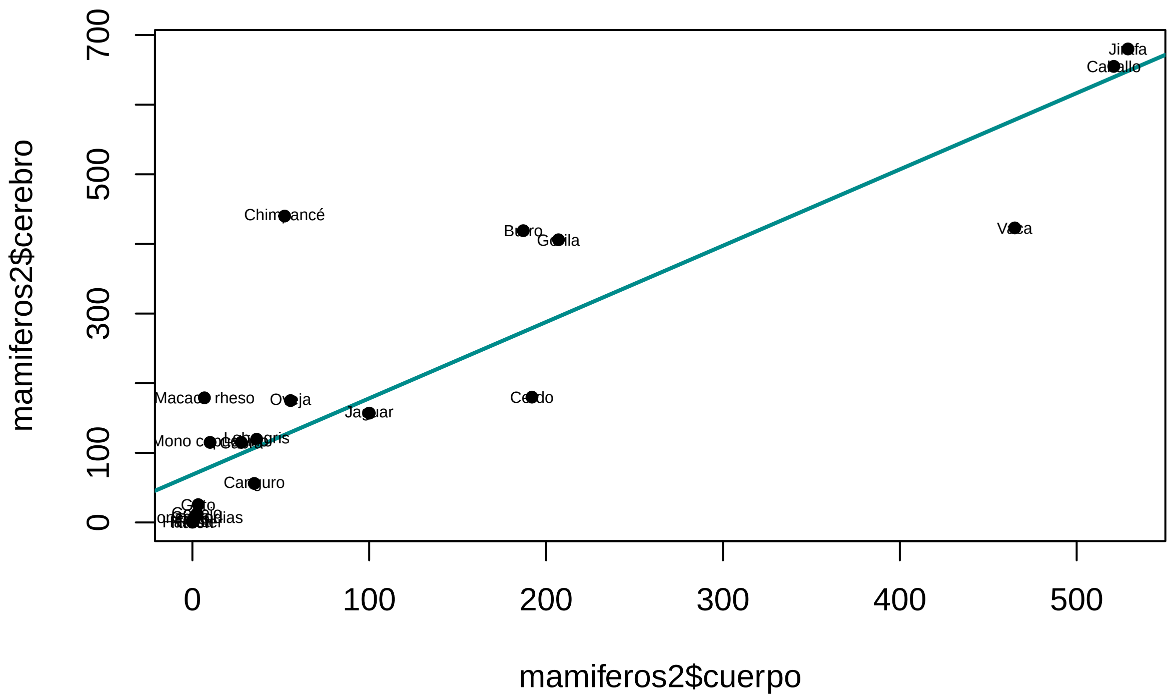

```
qqnorm(residuos, pch=19, cex=0.5,main = 'qq plot')
qqline(residuos,col="red")
```



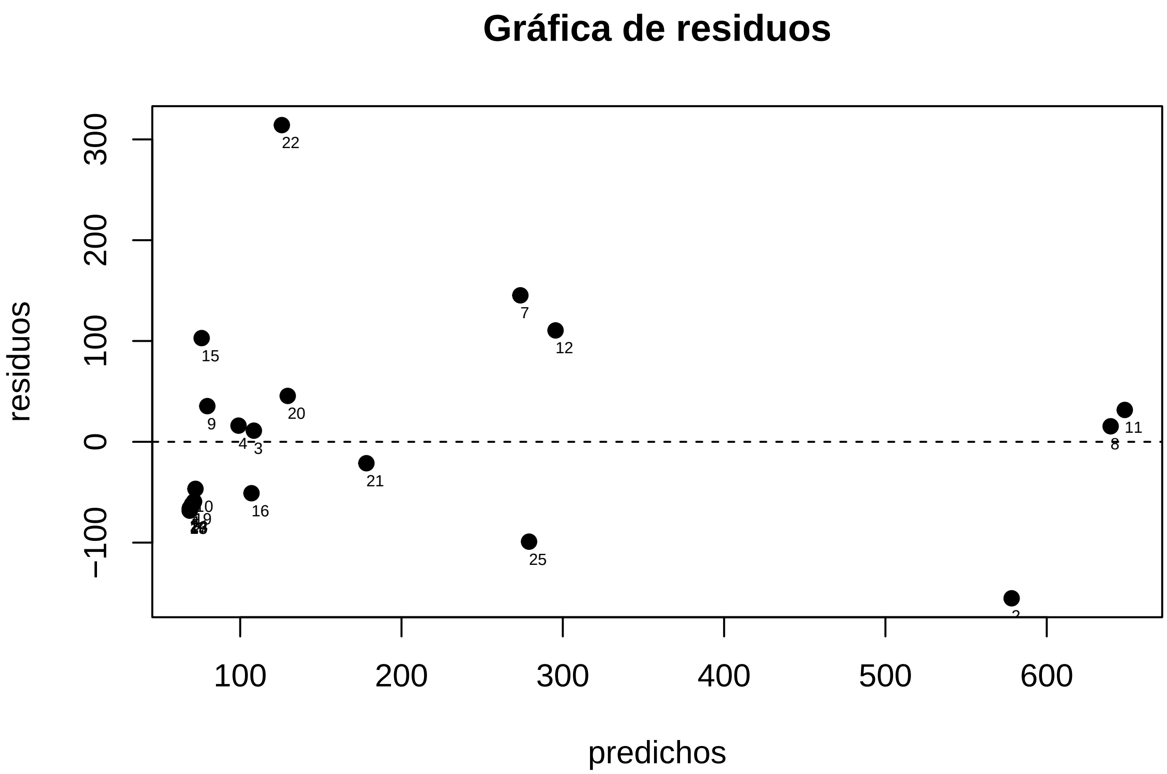

```
# Podríamos seguir quitando el 13 y/o 14 y viendo que sucede
mamiferos[c(6,13,14),]
```

```
##                Nombre cerebro cuerpo
## 6  Elefante asiatico    4603   2547
## 13            Hombre    1320     62
## 14 Elefante africano    5712   6654
```

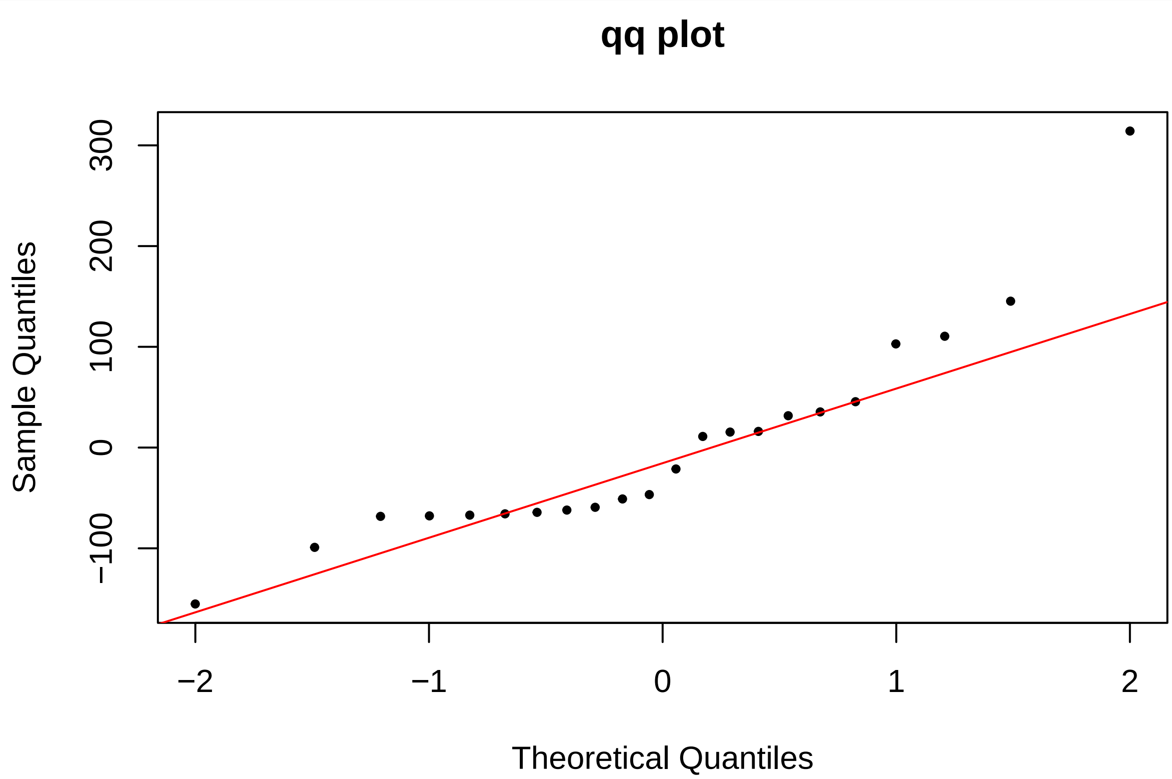
```r
mamiferos2<-mamiferos[-c(6,13,14),]
mod3<-lm(cerebro~cuerpo, data=mamiferos2, na.action=na.exclude)
plot(mamiferos2$cuerpo, mamiferos2$cerebro, type="p", pch=19,cex=0.75)
abline(coef=coef(mod3),col="darkcyan",lwd=2)
text(mamiferos2$cuerpo, mamiferos2$cerebro, labels=mamiferos2$Nombre,cex=0.5)
```




```r
# Dibujamos residuos vs predichos para realizar el diagnóstico del modelo
residuos <- residuals(mod3)
predichos <- fitted.values(mod3)
plot(predichos,residuos, pch=19, main = 'Gráfica de residuos')
text(predichos,residuos, labels=rownames(mamiferos2),cex=0.5,adj=c(0,2))

# Normalidad
abline(h=0,lty=2)
```




```
qqnorm(residuos, pch=19, cex=0.5,main = 'qq plot')
qqline(residuos,col="red")
```



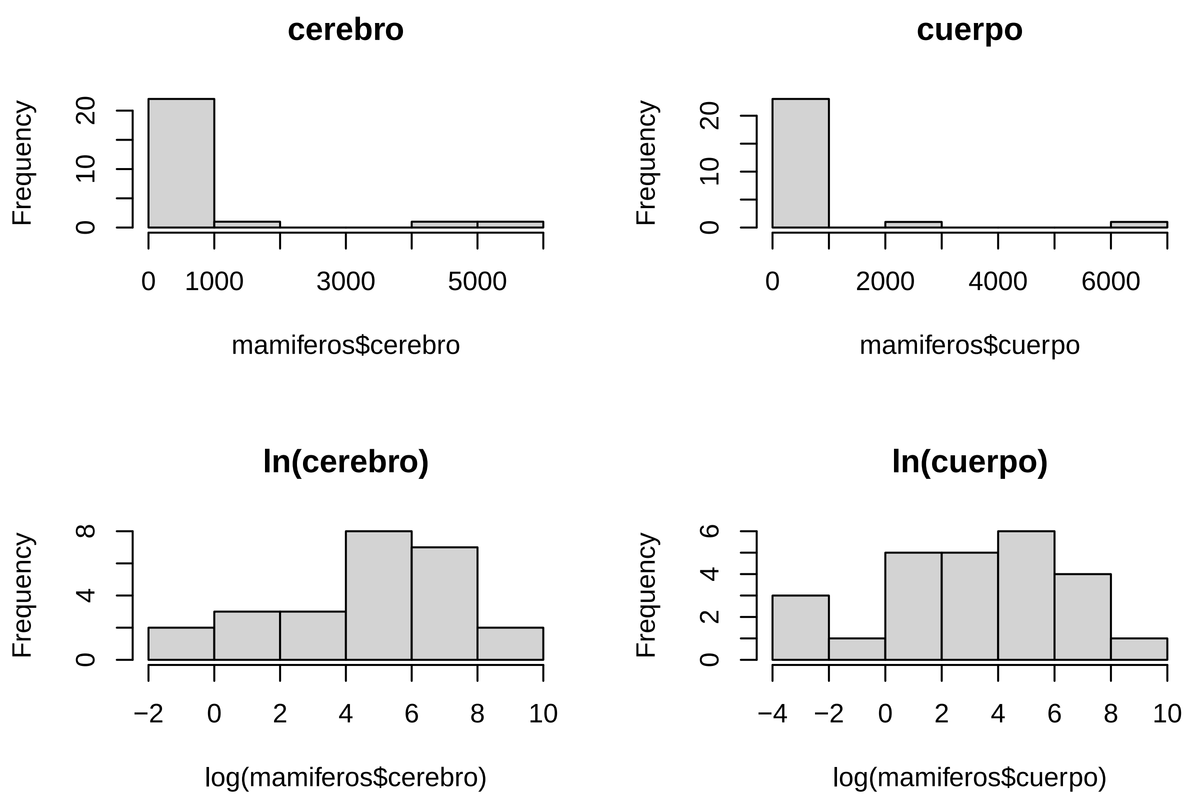

```
# Seguimos sin normalidad ni homocedasticidad. El tratamiento de estos
# datos directamente es muy cumplicado por las diferencias de escala.
```

```
# Una solución es la que se indica en el Ejercicio 8.
```

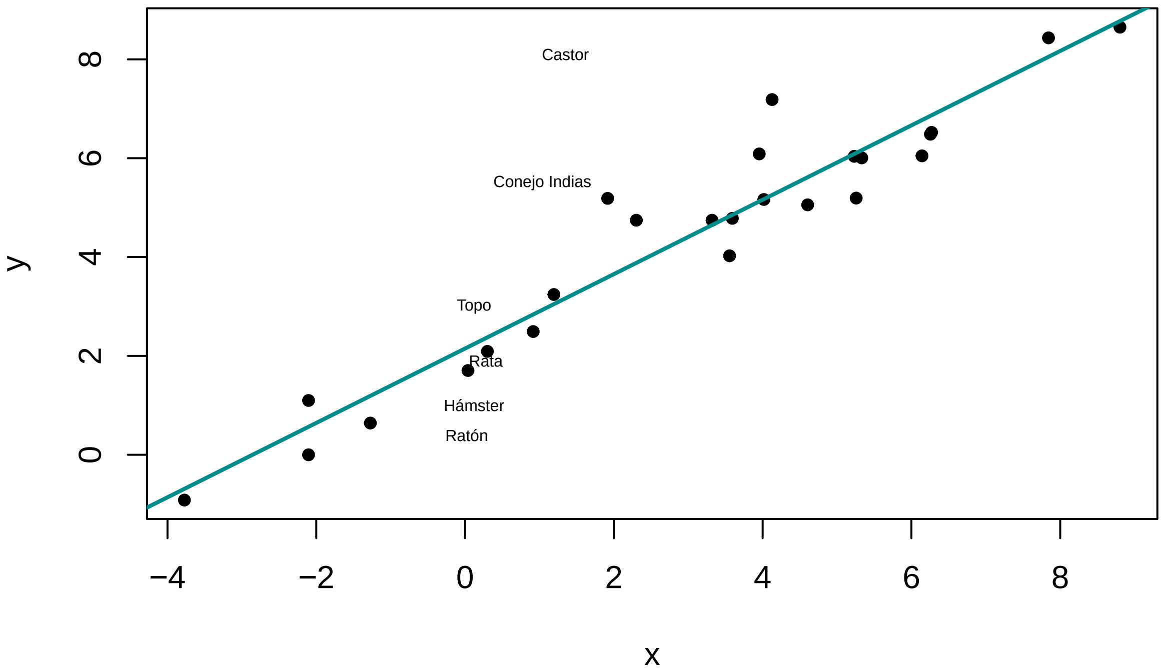
Ejercicio 8

Utilizando los datos de mamíferos, del banco **cerebros**, y las variables en escala logarítmica, dibuja el diagrama de puntos con la recta de mínimos cuadrados. A continuación, analiza gráficamente los residuos ¿crees que el modelo lineal sería adecuado?

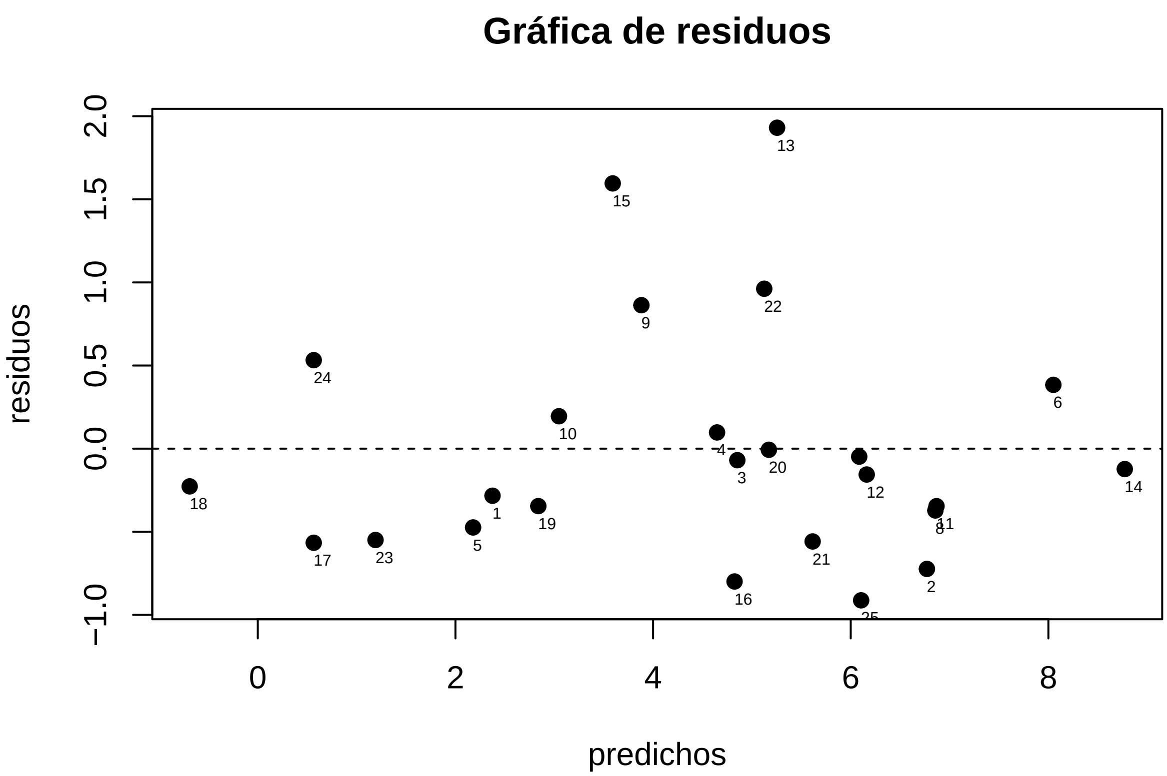
```
par(mfrow=c(2,2))
hist(mamiferos$cerebro, main="cerebro")
hist(mamiferos$cuerpo, main="cuerpo")
hist(log(mamiferos$cerebro),main="ln(cerebro)")
hist(log(mamiferos$cuerpo),main="ln(cuerpo)")
```



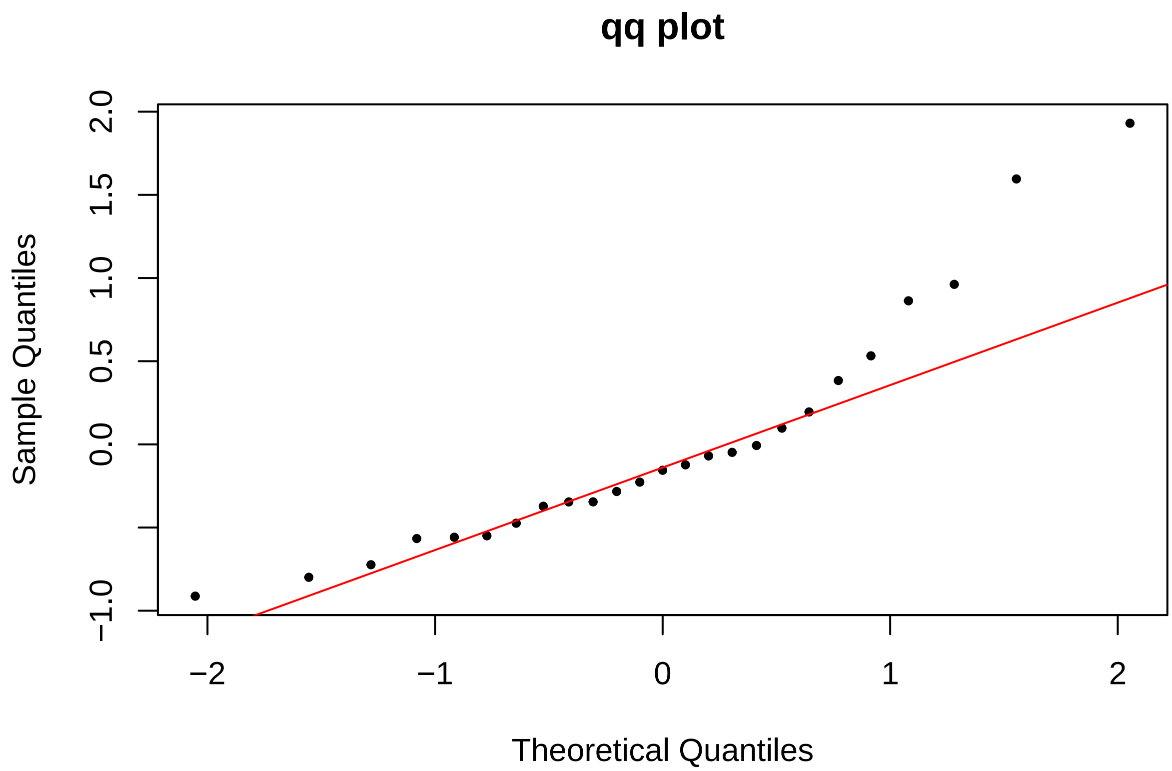

```
mamiferos<-cerebros[-c(26,27,28),]
y<-log(mamiferos$cerebro)
x<-log(mamiferos$cuerpo)


mod1<-lm(y~x, na.action=na.exclude)

plot(x,y, type="p", pch=19,cex=0.75)
abline(coef=coef(mod1),col="darkcyan",lwd=2)
text(mamiferos$cuerpo, mamiferos$cerebro, labels=mamiferos$Nombre,cex=0.5)
```




```r
# Dibujamos residuos vs predichos para realizar el diagnóstico del modelo
residuos <- residuals(mod1)
predichos <- fitted.values(mod1)
plot(predichos,residuos, pch=19, main = 'Gráfica de residuos')
text(predichos,residuos, labels=rownames(mamiferos),cex=0.5,adj=c(0,2))

# Normalidad
abline(h=0,lty=2)
```



```r
qqnorm(residuos, pch=19, cex=0.5,main = 'qq plot')
qqline(residuos,col="red")
```



Suponiendo adecuado el modelo lineal, contesta a las siguientes preguntas:

a) ¿Cuánto vale la pendiente de la recta? ¿Podemos afirmar que es positiva?

```r
coef(mod1)
```

```
## (Intercept)           x
##   2.1492577   0.7524776
```

b) Compara la varianza de la variable respuesta con la varianza de los residuos: ¿Qué porcentaje de la variabilidad inicial está explicado por la recta de mínimos cuadrados? ¿Qué porcentaje de la variabilidad inicial falta todavía por explicar?

```r
var(y)
```

```
## [1] 6.448081
```

```r
residuos <- residuals(mod1)
var(residuos)
```

```
## [1] 0.5054716
```

```r
R2<-1-var(residuos)/var(y)       # bondad del ajuste
R2*100 # Está explicado
```

```
## [1] 92.1609
```

```r
(1-R2)*100 # Falta por explicar
```

```
## [1] 7.8391
```

```r
# Lo anterior es lo mismo que:
n<-length(mamiferos$cerebro)
(summary(mod1)$sigma)^2*(n-2)/(n-1)
```

```
## [1] 0.5054716
```

```r
summary(mod1)$r.squared*100
```

```
## [1] 92.1609
```

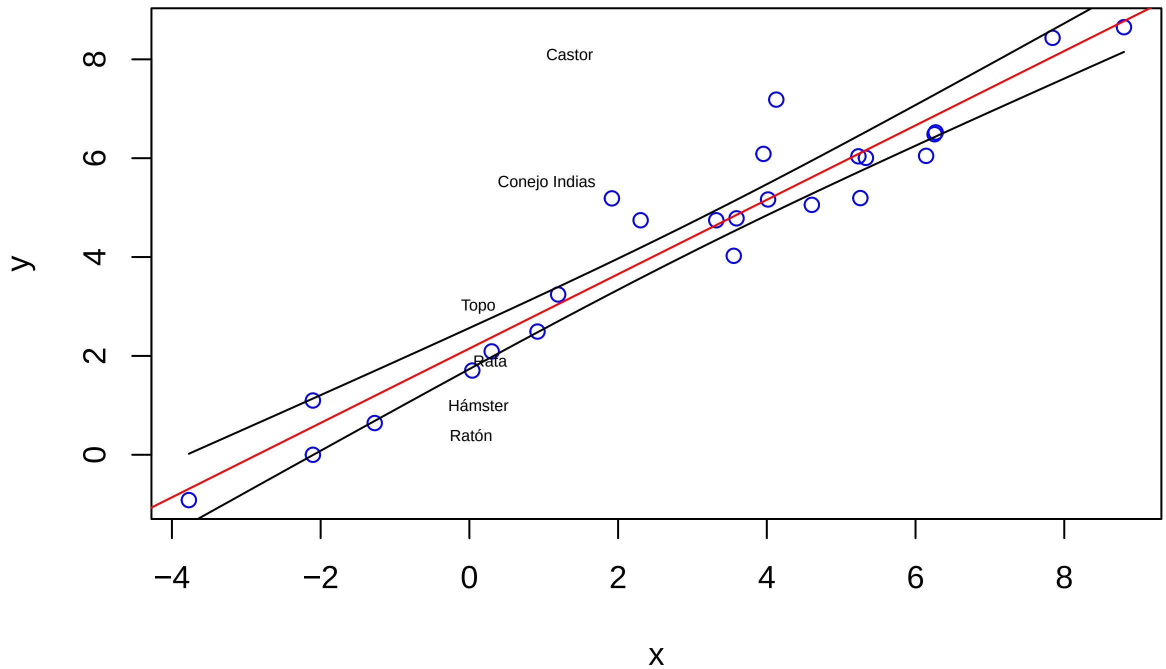
c) Obtén los intervalos de confianza al 90% sobre los parámetros de la recta.

```r
confint(mod1,level=.9)
```

```
##                   5 %     95 %
## (Intercept) 1.8051600 2.493355
## x           0.6740503 0.830905
```

d) Estima el valor de la recta de regresión en el punto *lcuerpo = 3* y calcula su intervalo de confianza al 95%. Dibuja el diagrama de dispersión, la recta de regresión y las bandas de confianza al 95% sobre la estimación de la recta.

```r
#Predicción
nuevos <- data.frame(list(x = 3))
bandas_est<-predict(mod1, newdata =nuevos, interval = "confidence")

#Obtención de bandas de estimación:
minx<-range(x)[1]; maxx<-range(x)[2]
nuevos <- data.frame(list(x = seq(minx,maxx,length=100)))
bandas_est<-predict(mod1, newdata = nuevos, interval = "confidence")

#Representación gráfica:
plot(x,y, col='BLUE')
abline(coef=coef(mod1), col='RED')
lines(nuevos$x,bandas_est[,2],col='BLACK')
lines(nuevos$x,bandas_est[,3],col='BLACK')
text(mamiferos$cuerpo, mamiferos$cerebro, labels=mamiferos$Nombre,cex=0.5)
```

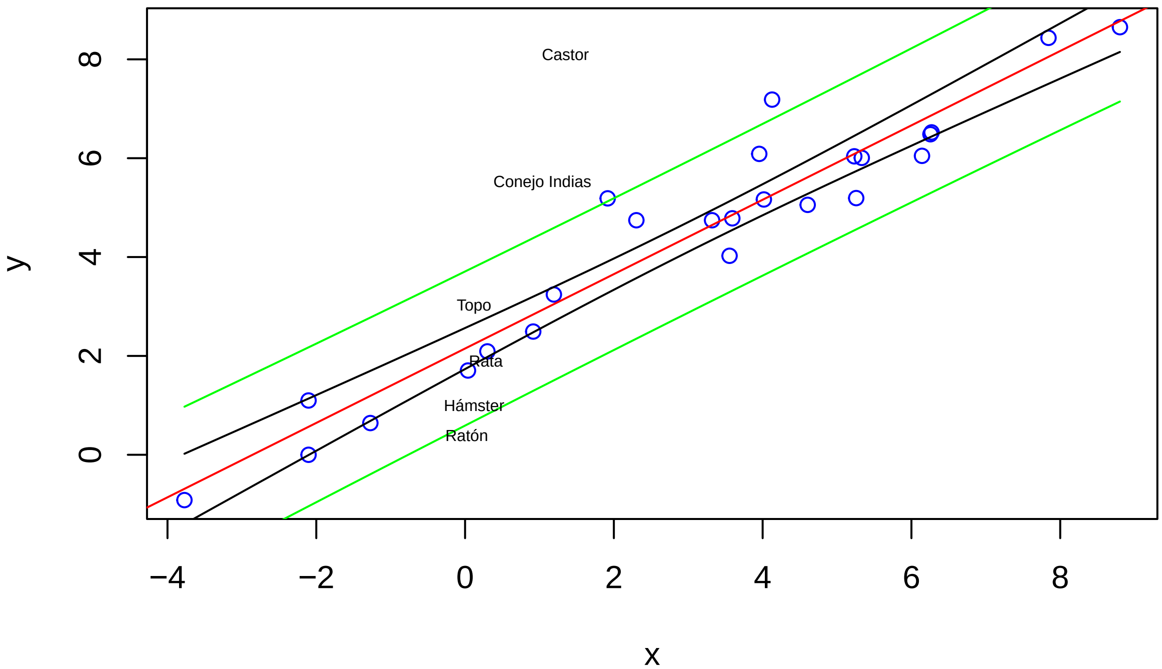


e) Obtén la predicción puntual y por intervalos (al 95%) de un nuevo mamífero con *lcuerpo = 6*. Añade a la gráfica anterior las bandas de predicción.

```
#Predicción
nuevos <- data.frame(list(x = 6))
predict(mod1, newdata =nuevos, interval = "prediction")
```

```
##        fit      lwr      upr
## 1 6.664123 5.106394 8.221853
```

```
#Obtención de bandas de estimación:
minx<-range(x)[1]; maxx<-range(x)[2]
nuevos <- data.frame(list(x = seq(minx,maxx,length=100)))
bandas_pred<-predict(mod1, newdata = nuevos, interval = "prediction")

#Representación gráfica:
plot(x,y, col='BLUE')
abline(coef=coef(mod1), col='RED')
lines(nuevos$x,bandas_est[,2],col='BLACK')
lines(nuevos$x,bandas_est[,3],col='BLACK')
lines(nuevos$x,bandas_pred[,2],col='GREEN')
lines(nuevos$x,bandas_pred[,3],col='GREEN')
text(mamiferos$cuerpo, mamiferos$cerebro, labels=mamiferos$Nombre,cex=0.5)
```



Ejercicio 9

El banco de datos **Advertising.csv** relaciona las ventas de ciertos productos con la inversión en publicidad, considerando diversos medios: televisión, radio y periódicos. Aquí vamos a estudiar la variable respuesta *sales* relacionándola con el predictor *TV*.

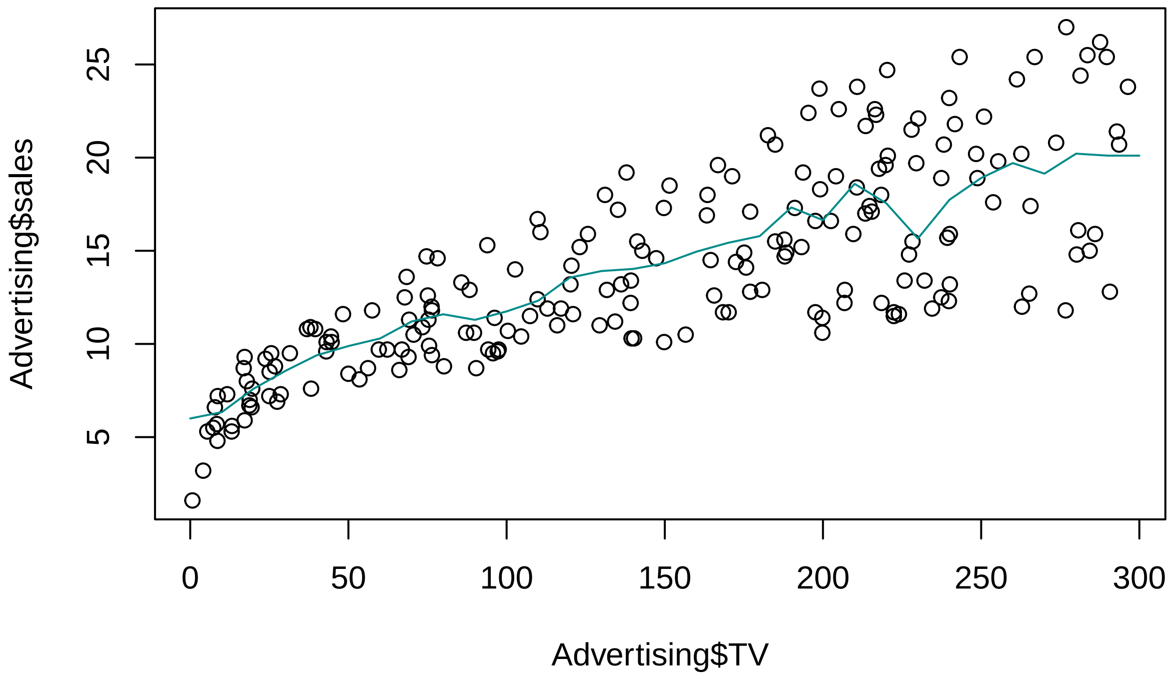
a) Obtén un ajuste mediante el método *KNN*, decidiendo el valor de *k* que consideres adecuado.

b) Obtén un ajuste mediante el método *loess*, decidiendo el valor de *span* que consideres adecuado.

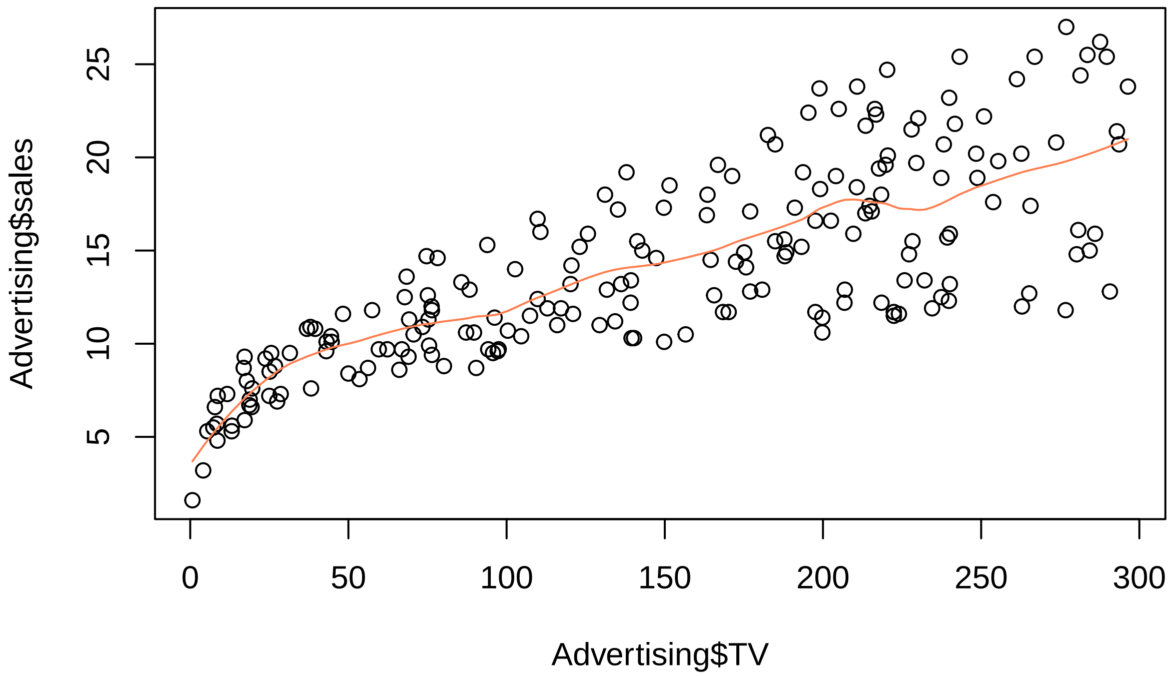
c) Dibuja, en la misma gráfica, los dos ajustes anteriores junto con la recta de mínimos cuadrados

```
Advertising <- read.csv('Advertising.csv')
```

```
plot(Advertising$TV,Advertising$sales)

xx <- seq(0,300,10)
TVred <- data.frame(list(TV = seq(0,300,10)))
modknn<- FNN::knn.reg(Advertising$TV,test=TVred, y=Advertising$sales, k=15)

lines(xx,modknn$pred,col="darkcyan")
```




```
plot(Advertising$TV,Advertising$sales)
modloess<- loess(sales~ TV,data=Advertising, span=0.3)
lines(sort(Advertising$TV),modloess$fitted[order(Advertising$TV)],col="coral")
```




```
plot(Advertising$TV,Advertising$sales)

xx <- seq(0,300,10)
TVred <- data.frame(list(TV = xx))
modknn<- FNN::knn.reg(Advertising$TV,test=TVred, y=Advertising$sales, k=15)
lines(xx,modknn$pred,col="darkcyan")
```

```
modloess<- loess(sales~ TV,data=Advertising, span=0.3)
lines(sort(Advertising$TV),modloess$fitted[order(Advertising$TV)],col="coral")
```