Ejercicio 8 LAB2

Laura Sudupe

12/10/2020

Utilizad los datos Altpes.xls (adjuntos) relativos a la altura(cm) y el peso (kg) de 30 individuos para saber si existe relación e intentad estimar el peso de una persona a partir de su altura.

P1: cargamos los datos, observamos el dataset y resumimos nuestras variables

```
library(readr)
altpes <- read_delim('/Users/lsudu/code/rlab/LAB2/LAB2_Data/Altpes.csv',
                     ';', escape_double =FALSE, trim_ws=TRUE)
```

```
##
## -- Column specification --------------------------------------------------------------------------------
## cols(
##   Indiv = col_double(),
##   Altura = col_double(),
##   Peso = col_double()
## )
```

```
head(altpes)
```

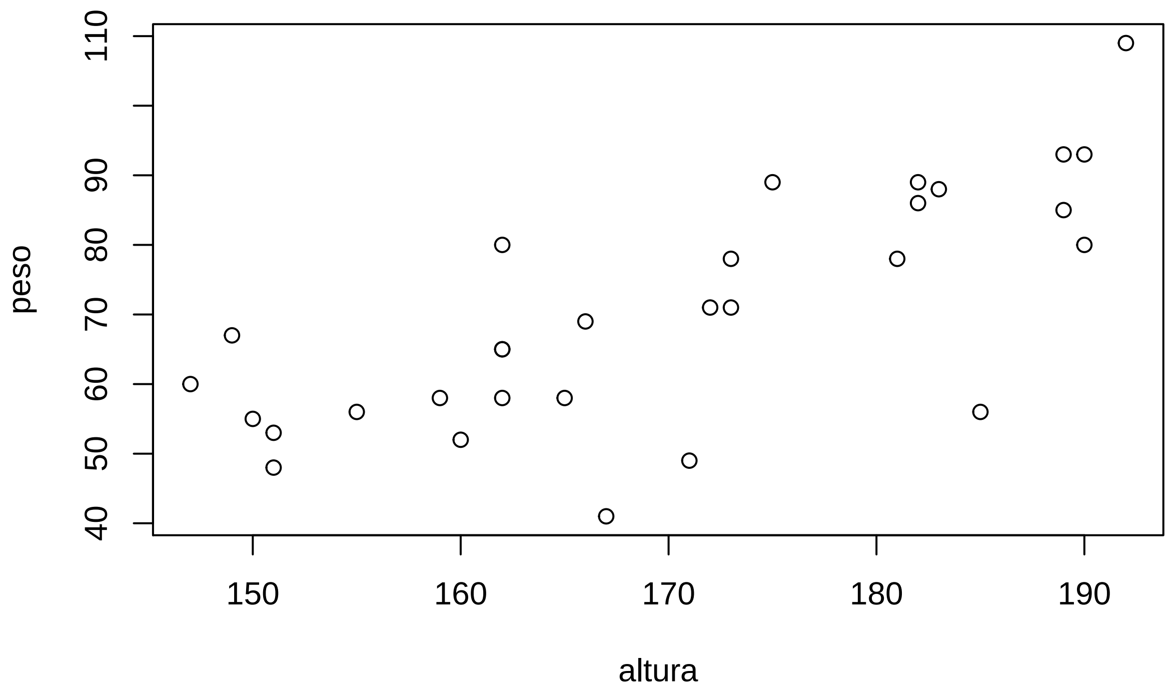
```
## # A tibble: 6 x 3
##   Indiv Altura  Peso
##   <dbl>  <dbl> <dbl>
## 1     1    190    80
## 2     2    155    56
## 3     3    167    41
## 4     4    171    49
## 5     5    182    89
## 6     6    173    71
```

```
summary(altpes)
```

```
##      Indiv           Altura           Peso
##  Min.   : 1.00   Min.   :147.0   Min.   : 41.00
##  1st Qu.: 8.25   1st Qu.:160.5   1st Qu.: 56.50
##  Median :15.50   Median :169.0   Median : 68.00
##  Mean   :15.50   Mean   :169.8   Mean   : 70.00
##  3rd Qu.:22.75   3rd Qu.:182.0   3rd Qu.: 83.75
##  Max.   :30.00   Max.   :192.0   Max.   :109.00
```

P2: Representamos las variables con un grafico de dispersión usando la funcion plot(). En nuestro caso, la variable "x" sera la altura, y la variable peso la "y"

```
plot(altpes$Altura, altpes$Peso, xlab='altura', ylab='peso')
```



P3: Como vemos en este gráfico, existe una relación lineal entre la altura y el peso aunque hay algunos puntos que se alejan. En este punto generamos nuestro modelo lineal y estimamos los parámetros que tendrá. .

```
modelo <- lm(formula= altpes$Peso~altpes$Altura, data=altpes)
summary(modelo)
```

```
##
## Call:
## lm(formula = altpes$Peso ~ altpes$Altura, data = altpes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.2452  -4.5041   0.6441   6.0433  19.6417
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -78.3168    25.8615  -3.028  0.00524 **
## altpes$Altura   0.8733     0.1518   5.754 3.56e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 11.46 on 28 degrees of freedom
## Multiple R-squared:  0.5418, Adjusted R-squared:  0.5254
## F-statistic: 33.11 on 1 and 28 DF,  p-value: 3.556e-06
```

```
modelo2 <- lm(formula= altpes$Altura~altpes$Peso, data=altpes)
summary(modelo2)
```

```
##
## Call:
## lm(formula = altpes$Altura ~ altpes$Peso, data = altpes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.9722  -5.0714  -0.0047   4.3261  23.8521
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 126.4061     7.7507  16.309 7.91e-16 ***
## altpes$Peso   0.6204     0.1078   5.754 3.56e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.658 on 28 degrees of freedom
## Multiple R-squared:  0.5418, Adjusted R-squared:  0.5254
## F-statistic: 33.11 on 1 and 28 DF,  p-value: 3.556e-06
```

P4: Para ajustar la recta de regresión debemos mirar el valor estimado de los parámetros en el resumen anterior. Nos fijamos en el coeficiente de determinación (R2) o "R-squared" = 0.5418 para comprobar cómo de bueno es el ajuste. Bastante malo

P5: Miramos la correlación de Pearson con la matriz de coeficientes de correlación y con el test de Pearson. Para medir una tendencia lineal entre dos variables numericas

```
cor(altpes)
```

```
##              Indiv     Altura       Peso
## Indiv   1.00000000 -0.1370394 0.01601266
## Altura -0.13703938  1.0000000 0.73606413
## Peso    0.01601266  0.7360641 1.00000000
```

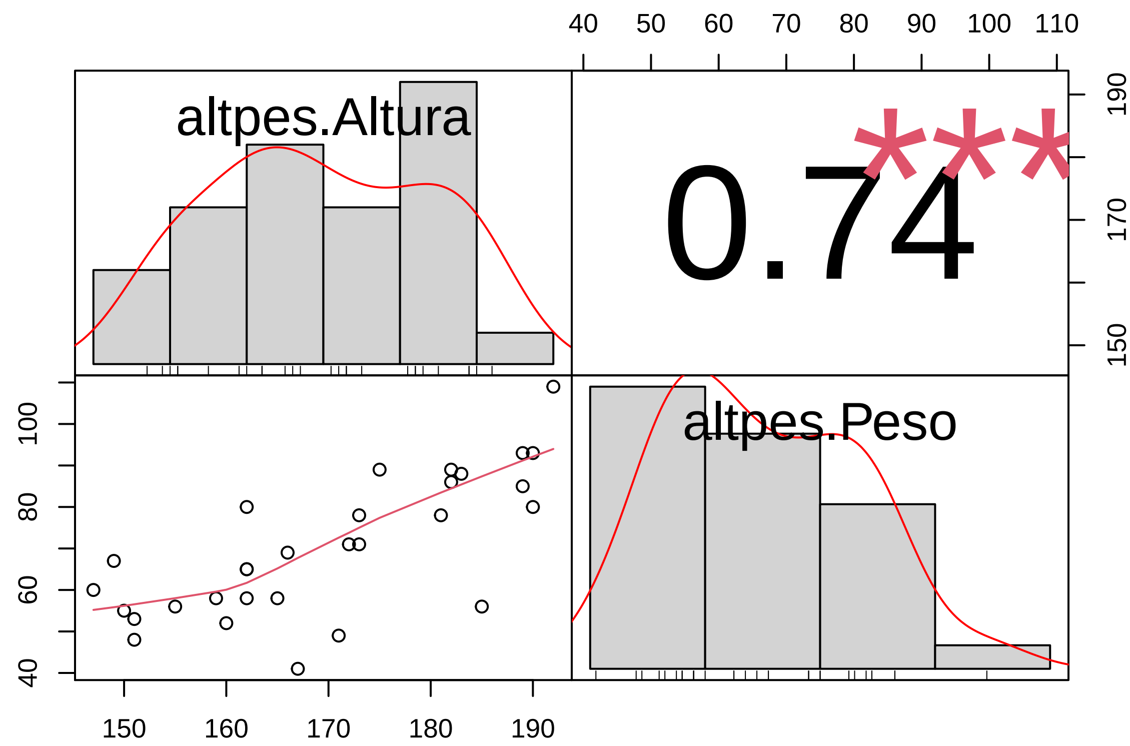
```
cor.test(altpes$Altura, altpes$Peso, method='pearson')
```

```
##
##  Pearson's product-moment correlation
##
## data:  altpes$Altura and altpes$Peso
## t = 5.7539, df = 28, p-value = 3.556e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5114117 0.8665425
## sample estimates:
##       cor
## 0.7360641
```

```r
#Otra manera de representarlo
#install.packages('PerformanceAnalytics')
library(PerformanceAnalytics)
```

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend
```

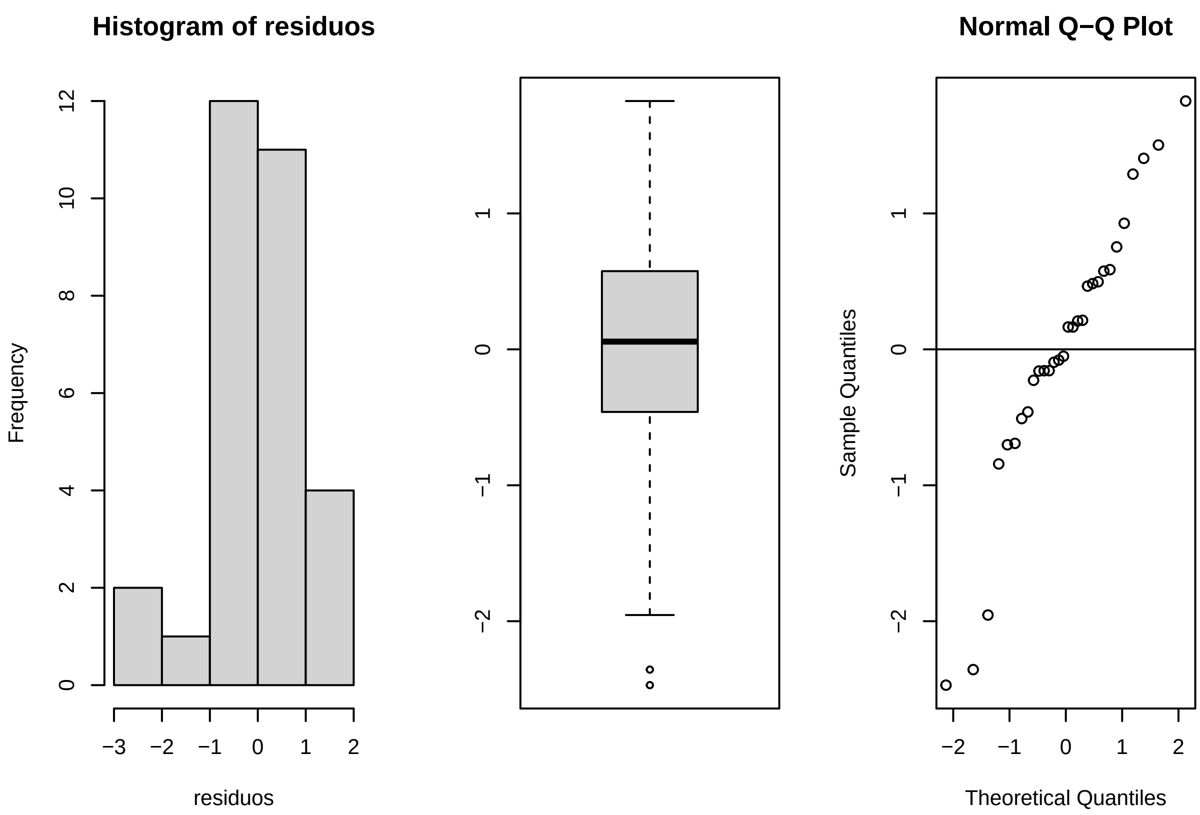
```r
data <- data.frame(altpes$Altura, altpes$Peso)
chart.Correlation(data)
```



Vemos que ambas variables son dependientes de la otra.

P6: Miramos la normalidad, es decir, si los errores estandarizados siguen una distribución normal con 3 gráficos.

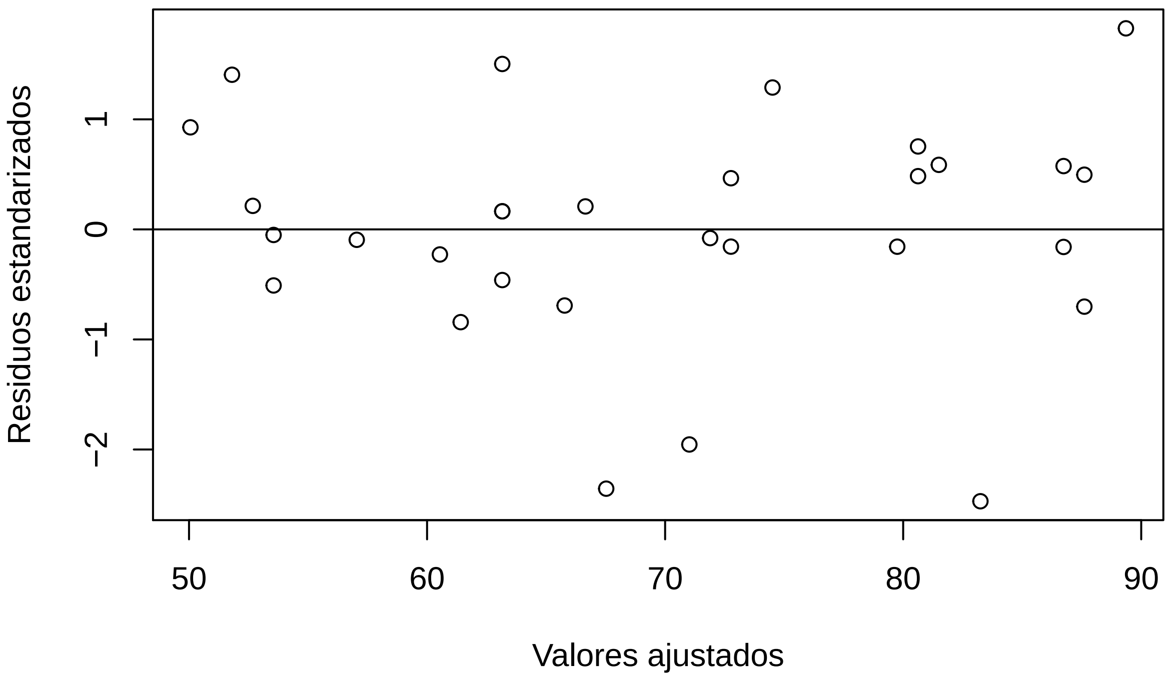
```
#install.packages('AER')
#library(AER)
residuos<-rstandard(modelo) # residuos estándares del modelo ajustado (completo)
par(mfrow=c(1,3)) # divide la ventana en una fila y tres columnas
hist(residuos) # histograma de los residuos estandarizados
boxplot(residuos) # diagrama de cajas de los residuos estandarizados
qqnorm(residuos) # gráfico de cuantiles de los residuos estandarizados
abline(h=0)
```




```
par(mfrow=c(1,1)) # devuelve la pantalla a su estado original sin divisiones
```

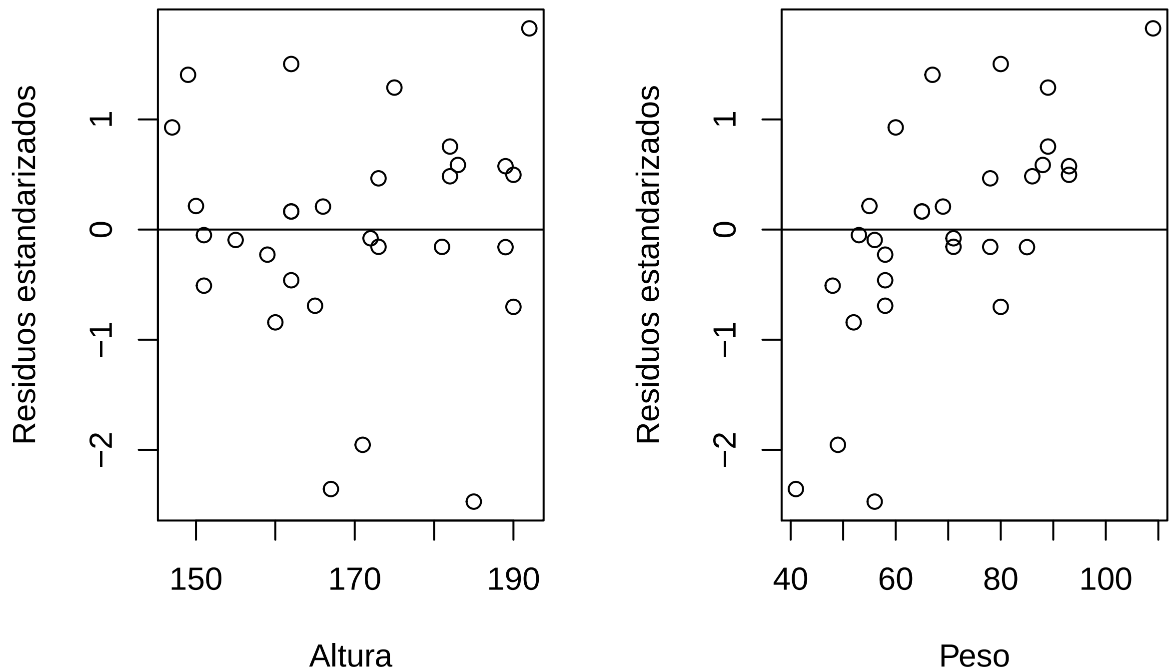
P7: Miramos si la varianza de los errores es constante con el gráfico siguiente

```
plot(fitted.values(modelo), rstandard(modelo), xlab='Valores ajustados',
     ylab='Residuos estandarizados')
abline(h=0) #dibujamos la recta en el 0
```



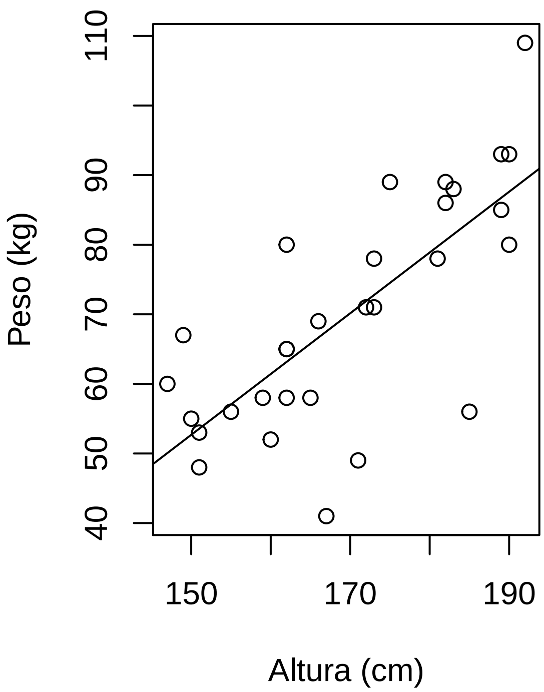
P8: Buscamos valores atípicos

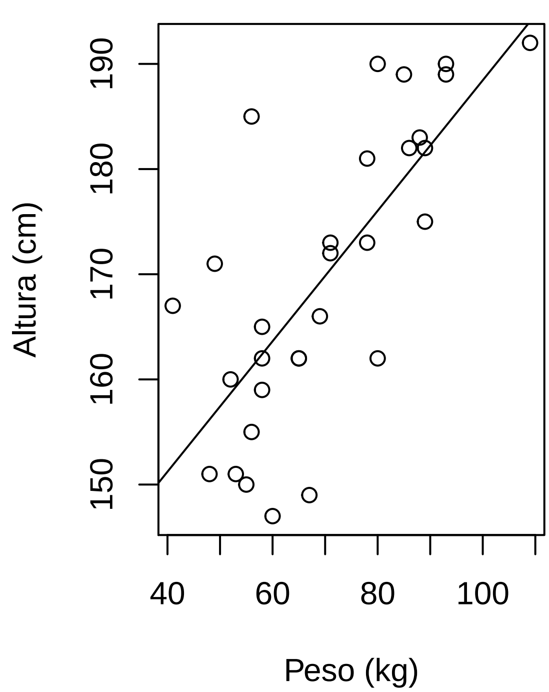
```
par(mfrow=c(1,2))
plot(altpes$Altura,rstandard(modelo),xlab="Altura",
     ylab="Residuos estandarizados")
abline(h=0)
plot(altpes$Peso,rstandard(modelo),xlab="Peso",
     ylab="Residuos estandarizados")
abline(h=0)
```



P9: Visualizamos la regresión lineal

```
par(mfrow=c(1,2))
plot(altpes$Altura, altpes$Peso, xlab='Altura (cm)', ylab='Peso (kg)')
abline(modelo)
plot(altpes$Peso, altpes$Altura, xlab='Peso (kg)', ylab='Altura (cm)')
abline(modelo2)
```





P10: Utilizando regresión lineal, obtenemos el rendimiento que cabe esperar • (C) Estimad el peso que tendrá una mujer que mida 168 cm. Podemos usar la función predict() o sustituir directamente en el modelo con parámetros estimados.

```
#(Intercept)    -78.3168
#altpes$Altura    0.8733

peso_m168 = -78.3168 + 0.8733*(168)
peso_m168
```

```
## [1] 68.3976
```