Práctica 3 - Parte I

Aprendizaje Estadístico - (84:04)

A. Verri Kozlowski y F. Patitucci

Ejercicio 1

Implementar una función que dado un vector `y` de valores de respuesta y una matriz `X` de valores observados, mediante las ecuaciones normales, calcule el estimador de cuadrados mínimos

Ejercicio 2

Se tiene en el archivo `girasol.txt` el rinde de diversas parcelas de girasol (en toneladas) según la cantidad de dinero invertida en fertilizantes (en miles de pesos).

a Graficar en un diagrama de dispersión inversión vs rinde. b. Plantear un modelo de regresión lineal simple obtener el estimador de mínimos cuadrados. c. Graficar la recta de regresión obtenida, ¿detecta algo sospechoso?

Ejercicio 3

Considerar el archivo abalone.txt que contiene información sobre distintas muestras de abalones. Los atributos están separados por coma, con los siguientes campos:

- Sexo (categórica): M (masculino), F (femenino) o I (infante).
- Longitud (continua), en milimetros.
- Diametro (continua), en milimetros.
- Altura (continua), en milimetros.
- Peso completo del abalone (continua), en gramos.
- Peso de la carne (continua), en gramos.
- Peso de las vísceras (continua), en gramos.
- Peso del caparazón (continua), en gramos.
- Anillos (discreta).

a. Plantear un modelo de regresión lineal simple para predecir el diámetro en función de la longitud.

b. Observar que el conjunto de datos tiene información del peso total de cada espécimen junto con un desagregado por partes. Ajustar un modelo de regresión múltiple que explique el peso total en función del peso del caparazón, las vísceras y la carne.

c. Se trata ahora de establecer una relación entre el peso total y el diámetro del espécimen. Empezar dibujando en un scatter plot ambas variables. Si definimos como `P` al peso total y `D` al diámetro, se consideran los siguientes modelos:

- Modelo lineal simple, $P = b + aD + \varepsilon$

- Modelo cuadrático, $P = c + bD + aD^2 + \varepsilon$
- Modelo cubico sin términos de orden inferior, \$P = a.D^3 + ε

Efectuar en cada caso una regresión y graficar las curvas superpuestas sobre el scatter plot.

Ejercicio 4

En este ejercicio se crearán datos simulados y se ajustará un modelo de regresión lineal simple.

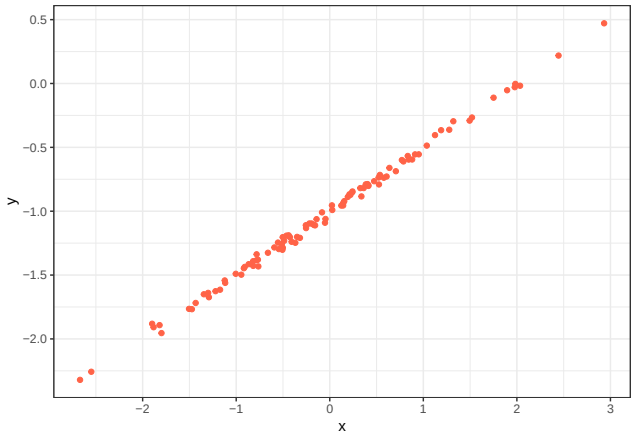
a) Utilizando la función rnorm, crear un vector x que contenga 100 observaciones provenientes de una distribución $X \sim \mathcal{N}(\mu = 0,\ \sigma = 1)$
b) Utilizando la función rnorm, crear un vector epsilon que contenga 100 observaciones provenientes de una distribución $\Sigma \sim \mathcal{N}(\mu = 0, \sigma = 0.025)$
c) Usando x y ε, generar un vector acorde al modelo: $y = -1 + 0.5x + \varepsilon$. ¿Cuál es la longitud del vector y? ¿Cuáles son los valores de β_0 y β_1 en el modelo?

```
X <- rnorm(n=100,mean=0,sd=1)
e <- rnorm(n=100,mean=0,sd=0.025)
Y <- -1 + 0.5*X +e
```

La longitud del vector `Y=-1+0.5*X+e` *es igual a 100 ya que fue construido como la suma de dos vectores de 100 elementos, y además R tiene sobrecargado el operador suma y permite agregar la constante* `1` *a cada elemento del vector. Los parámetros* β_0 *y* β_1 *del modelo son los coeficientes de regresión del modelo, y para este caso resultan* $\beta_0 = -1$ *y* $\beta_1 = +0.5$

d) Realizar un scatterplot y observar la relación entre x e y.

```
DT <- data.table(x=X,y=Y)
ggplot(data=DT, aes(x=x, y=y)) + geom_point(color="tomato") + theme_bw()
```



e) Ajustar un modelo lineal para predecir y en función de x utilizando el método de cuadrados mínimos.

```
DT <- data.table(x=X,y=Y)
MDL <- lm(data=DT,y~x)
b0 <- MDL$coefficients["(Intercept)"]
b1 <- MDL$coefficients["x"]
```

Comparar los valores exactos de β_0 Y β_1 con sus estimaciones.

```
data.table("\\beta_0 / \\hat{\\beta_0}"=-1/b0,"\\beta_1 / \\hat{\\beta_1}"=0.5/b1) |>
  flextable() |>  colformat_double(digits = 4) |>
  mk_par(part = "header", value = as_paragraph(as_equation(.)),use_dot = TRUE) |>    autofit()
```

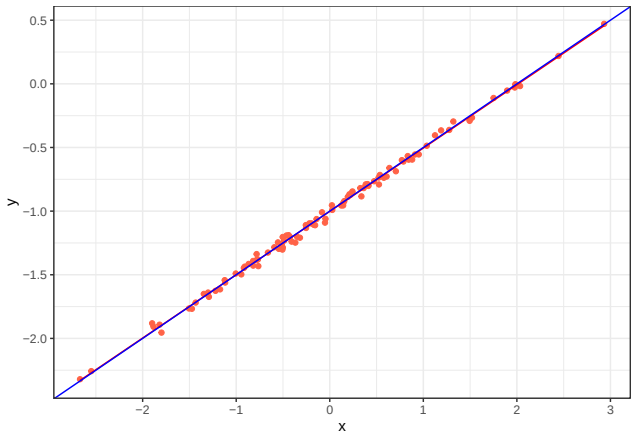
$\beta_0/\hat{\beta}_0$	$\beta_1/\hat{\beta}_1$
0.9979	1.0052

Los estimadores de β_0 y β_1 resultaron $\hat{\beta}_0 \approx$ -1.0021334 y $\hat{\beta}_1 \approx$ 0.4974194, respectivamente

f) Graficar la recta de cuadrados mínimos sobre el gráfico realizado en (d). En otro color graficar la recta $Y = -1 + 0,5X$

```
DT <- data.table(x=X,y=Y)
DTP <- data.table(x=X,y=b0+b1*X) #Predictor
B0 <- -1
B1 <- 0.5

ggplot() + geom_point(data=DT, aes(x=x, y=y),color="tomato") + geom_line(data=DTP,aes(x=x, y=y),color="red")+ geom_
```



Las lineas son prácticamente idénticas y no hay manera de diferenciarlas, tal como se vio en la comparación de los coeficientes

g) Ajustar un modelo polinomial que prediga y usando x y x^2 .

```
DT <- data.table(y=Y,x1=X,x2=X^2)
MDL <- lm(data=DT,y~x1+x2)
b0 <- MDL$coefficients["(Intercept)"]
b1 <- MDL$coefficients["x1"]
b2 <- MDL$coefficients["x2"]
```

Los estimadores de β_0 y β_1 ahora resultan en $\hat{\beta}_0 \approx$ -1.0038418 y $\hat{\beta}_1 \approx$ 0.4972626 respectivamente y se agrega un tercer estimador para el término cuadrático igual a $\hat{\beta}_2 \approx$ 0.0015385

```
data.table("\\beta_0/\\hat{\\beta_0}"=B0/b0,"\\beta_1/\\hat{\\beta_1}"=B1/b1) |>
  flextable() |>  colformat_double(digits = 4) |>
  mk_par(part = "header", value = as_paragraph(as_equation(.)),use_dot = TRUE) |>    autofit()
```

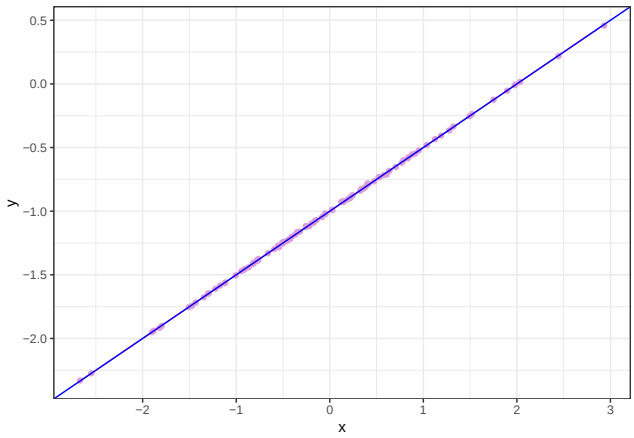
$\beta_0/\hat{\beta}_0$	$\beta_1/\hat{\beta}_1$
0.9962	1.0055

¿Encuentra alguna evidencia de que el término cuadrático mejora el ajuste del modelo?

No, los estimadores $\hat{\beta}_0 \approx$ -1.0038418 y $\hat{\beta}_1 \approx$ 0.4972626 prácticamente no difieren de los anteriores.

h) Repetir los ítems (a) a (f) modificando los datos generados de manera que haya menos ruido en los datos. Una forma de hacerlo es disminuyendo el valor de la varianza de la distribución normal usada para general el término del error epsilon.

```
e <- rnorm(n=100,mean=0,sd=0.025/5)
Y <- -1 + 0.5*X + e
DT <- data.table(x=X,y=Y)
MDL <- lm(data=DT,y~1+x)
b0 <- MDL$coefficients["(Intercept)"]
b1 <- MDL$coefficients["x"]
DTP <- data.table(x=X,y=b0+b1*X)
ggplot() + geom_point(data=DT, aes(x=x, y=y),color="plum") + geom_line(data=DTP,aes(x=x, y=y),color="salmon")+ geom
```



Reduciendo la varianza cinco veces ($\varepsilon \sim \mathcal{N}(0, 0.005)$,los estimadores de β_0 y β_1 ahora resultan en $\hat{\beta}_0 \approx$ -0.9999276 y $\hat{\beta}_1 \approx$ 0.5000883 respectivamente

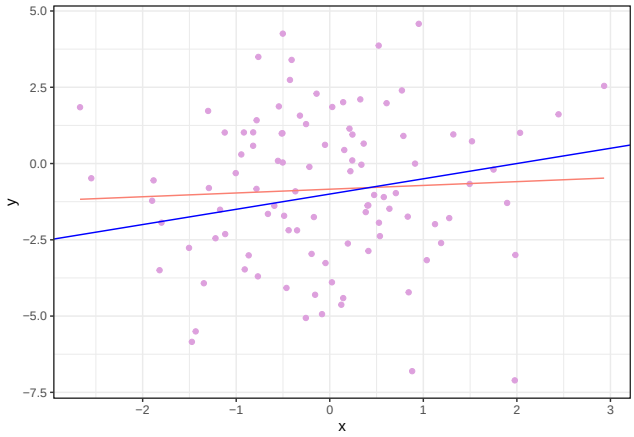
```
data.table("\\beta_0 / \\hat{\\beta_0}"=B0/b0,"\\beta_1 / \\hat{\\beta_1}"=B1/b1) |>
  flextable() |>  colformat_double(digits = 4) |>
  mk_par(part = "header", value = as_paragraph(as_equation(.)),use_dot = TRUE) |>    autofit()
```

$\beta_0/\hat{\beta}_0$	$\beta_1/\hat{\beta}_1$
1.0001	0.9998

i) Repetir los ítems (a) a (f) modificando los datos generados de manera que haya más ruido en los datos. Una forma de hacerlo es aumentando el valor de la varianza de la distribución normal usada para general el término del error epsilon.

```
e <- rnorm(n=100,mean=0,sd=0.25)
Y <- -1 + 0.5*X + 10*e
DT <- data.table(x=X,y=Y)
MDL <- lm(data=DT,y~1+x)
b0 <- MDL$coefficients["(Intercept)"]
b1 <- MDL$coefficients["x"]
DTP <- data.table(x=X,y=b0+b1*X)


ggplot() + geom_point(data=DT, aes(x=x, y=y),color="plum") + geom_line(data=DTP,aes(x=x, y=y),color="salmon")+ geom
```



Aumentando la varianza cinco veces($\varepsilon \sim \mathcal{N}(0, 0.125)$,*los estimadores de* β_0 *y* β_1 *ahora resultan en* $\hat{\beta}_0 \approx$ *-0.8412213 y* $\hat{\beta}_1 \approx$ *0.1238318 respectivamente*

```
data.table("\\beta_0 / \\hat{\\beta_0}"=B0/b0,"\\beta_1 / \\hat{\\beta_1}"=B1/b1) |>
  flextable() |>  colformat_double(digits = 4) |>
  mk_par(part = "header", value = as_paragraph(as_equation(.)),use_dot = TRUE) |>    autofit()
```

$\beta_0/\hat{\beta}_0$	$\beta_1/\hat{\beta}_1$
1.1887	4.0377

j) En ambos escenarios, hallar una estimación de la varianza.

```
if(knitr::is_html_output()){
   plotNoise()
}
```

Ejercicio 5

a) Generar el siguiente modelo: Crear dos vectores de datos x_1 y x_2 de tamaño 100 con una distribución $X \sim \mathcal{U}(0, 1)$ y crear un vector $y = 2 + 2 * x_1 + 0,3 * x_2 + \varepsilon$, con ε que contenga 100 observaciones con una distribución $\varepsilon \sim \mathcal{N}(0, 1)$. ¿Cuales son los coeficientes de regresión?. Estimar la correlación

entre x_1 y x_2. Realizar un scatterplot en el que pueda observarse la relación entre x_1 y x_2. Utilizando los datos generados, ajustar a un modelo lineal para predecir y en función de x_1 y x_2 , utilizando el método de cuadrados mínimos y comparar los valores exactos de β con sus valores estimados.

```
X1 <- runif(n=100,min=0,max=1)
X2 <- runif(n=100,min=0,max=1)
e <- rnorm(n=100,mean=0,sd=1)
B0 <- 2
B1 <- 2
B2 <- 0.3
Y <- B0+B1*X1+B2*X2+e
```

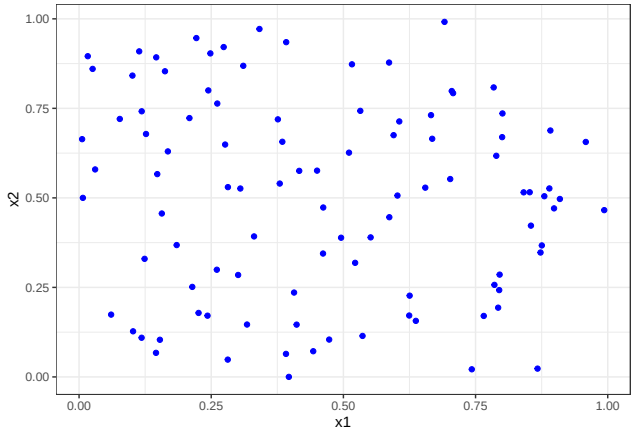
Para el modelo propuesto, los coeficientes de regresión son $\beta_0 = 2$, $\beta_1 = 2$ y $\beta_2 = 0.3$ La función `cor()` permite estimar la correlación entre dos vectores que para este caso resulta `cor(X1,X2)`= -0.105048

```
cor(X1,X2)
```

```
## [1] -0.105048
```

Los vectores X_1 y X_2 son realizaciones de una muestra aleatoria con distribución uniforme y no deberían tener ninguna correlación como se puede ver en el siguiente gráfico de dispersión.

```
DT <- data.table(x1=X1,x2=X2)
ggplot() + geom_point(data=DT, aes(x=x1, y=x2),color="blue")  + theme_bw()
```




```
DT <- data.table(x1=X1,x2=X2,y=Y)
MDL <- lm(data=DT,y~x1+x2)
b0 <- MDL$coefficients["(Intercept)"]
b1 <- MDL$coefficients["x1"]
b2 <- MDL$coefficients["x2"]
B0 <- 2
B1 <- 2
B2 <- 0.3
SUM <- data.table("\\beta_0 / \\hat{\\beta_0}"=B0/b0,"\\beta_1 / \\hat{\\beta_1}"=B1/b1,"\\beta_2 / \\hat{\\beta_2}
```

La regresión mediante cuadrados mínimos se efectúa mediante el paquete 'lm()' y los estimadores de los oeficientes de regresión del modelo, resultan en $\hat{\beta}_0 = 2.3161401$,$\hat{\beta}_0 = 1.8964135$ y $\hat{\beta}_0 = 0.2563734$

Comparar los valores exactos de β_0 Y β_1 con sus estimaciones.

```
flextable(SUM) |>  colformat_double(digits = 4) |>
mk_par(part = "header", value = as_paragraph(as_equation(.)),use_dot = TRUE) |>    autofit()
```

$\beta_0/\hat{\beta}_0$	$\beta_1/\hat{\beta}_1$	$\beta_2/\hat{\beta}_2$
0.8635	1.0546	1.1702

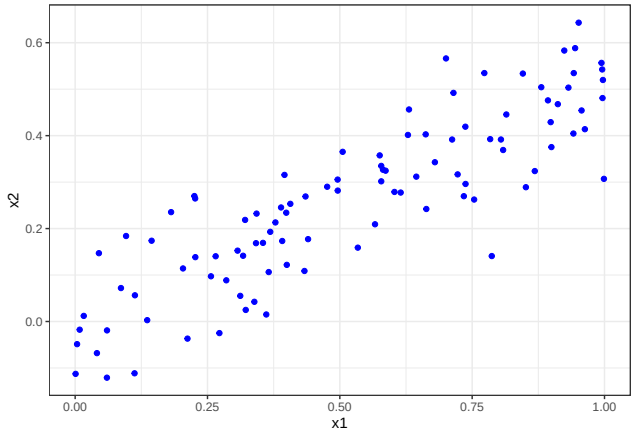
f) Repetir el inciso a) pero con el siguiente modelo:

- Crear dos vectores de datos de tamaño 100 x_1 y $x_2 = 0.5x_1 + rnorm(100)/10$ a partir de una distribución uniforme en el intervalo $(0,\ 1)$ $X \sim \mathcal{U}(0,1)$
- Crear el vector $y = 2 + 2x_1 + 0,3x_2 + rnorm(100)$
- Comparar los resultados obtenidos con los del ítem a)

```
X1 <- runif(n=100,min=0,max=1)
X2 <- 0.5*X1+rnorm(n=100,mean=0,sd=1)/10 # == rnorm(n=100,mean=0,sd=0.1) ?
e <- rnorm(n=100,mean=0,sd=1)
B0 <- 2
B1 <- 2
B2 <- 0.3
Y <- B0+B1*X1+B2*X2+e
```

En este ejemplo sólo el vector X_1 es una realización "pura" de una muestra aleatoria con distribución normal y el vector X_2 se genera como una combinación lineal de X_1 pero incluye un ruido aleatorio del 10% de la varianza de X_1, con lo cual la correlación entre ambos resulta ahora más alta `cor(X1,X2)=` *0.8666937*

```
DT <- data.table(x1=X1,x2=X2)
ggplot() + geom_point(data=DT, aes(x=x1, y=x2),color="blue")  + theme_bw()
```




```
DT <- data.table(x1=X1,x2=X2,y=Y)
MDL <- lm(data=DT,y~x1+x2)
b0 <- MDL$coefficients["(Intercept)"]
b1 <- MDL$coefficients["x1"]
b2 <- MDL$coefficients["x2"]
SUM <- data.table("\\beta_0 / \\hat{\\beta_0}"=B0/b0,"\\beta_1 / \\hat{\\beta_1}"=B1/b1,"\\beta_2 / \\hat{\\beta_2}
```

Los estimadores de los oeficientes de regresión del modelo, resultan ahora en en $\hat{\beta}_0 = 2.234869$,$\hat{\beta}_0 = 0.9685999$ *y* $\hat{\beta}_0 = 1.5500093$

```
flextable(SUM) |>  colformat_double(digits = 4) |>
mk_par(part = "header", value = as_paragraph(as_equation(.)),use_dot = TRUE) |>    autofit()
```

$\beta_0/\hat{\beta}_0$	$\beta_1/\hat{\beta}_1$	$\beta_2/\hat{\beta}_2$
0.8949	2.0648	0.1935