Ejercicios 1 a 3 hogg fitting

Preparación previa

Primero vamos a preparar los datos para usarlos facilmente

```
library(MASS)
x <- c(201, 244, 47, 287, 203, 58, 210, 202, 198, 158,165,201,157,131,166,160,186,125,218,146)
y <- c(592,401,583,402,495,173,479,504,510,416,393,442,317,311,400,337,423,334,533,344)
stdy <- c(61,25,38,15,21,15,27,14,30,16,14,25,52,16,34,31,42,26,16,22)
stdx <- c(9,4,11,7,5,9,4,4,11,7,5,5,5,6,6,5,9,8,6,5)
roxy <- c(-0.84, 0.31,0.64,-0.27,-0.33,0.67,-0.02,-0.05,-0.84,-0.69,0.3,-0.46,-0.03,0.5,0.73,-0.52,0.9,

df <- data.frame(x,y,stdy,stdx,roxy)
df["varx"] <- stdx^2
df["vary"] <- stdy^2
df["rosd"] <- roxy*stdx*stdy
df["uncertxn"] <- x-stdx
df["uncertxp"] <- x+stdx
df["uncertyn"] <- y-stdy
df["uncertyp"] <- y+stdy
mstd1 <- diag(df[,"vary"])
mstd2 <- diag(df[5:20,"vary"])
```

Ejercicio 1

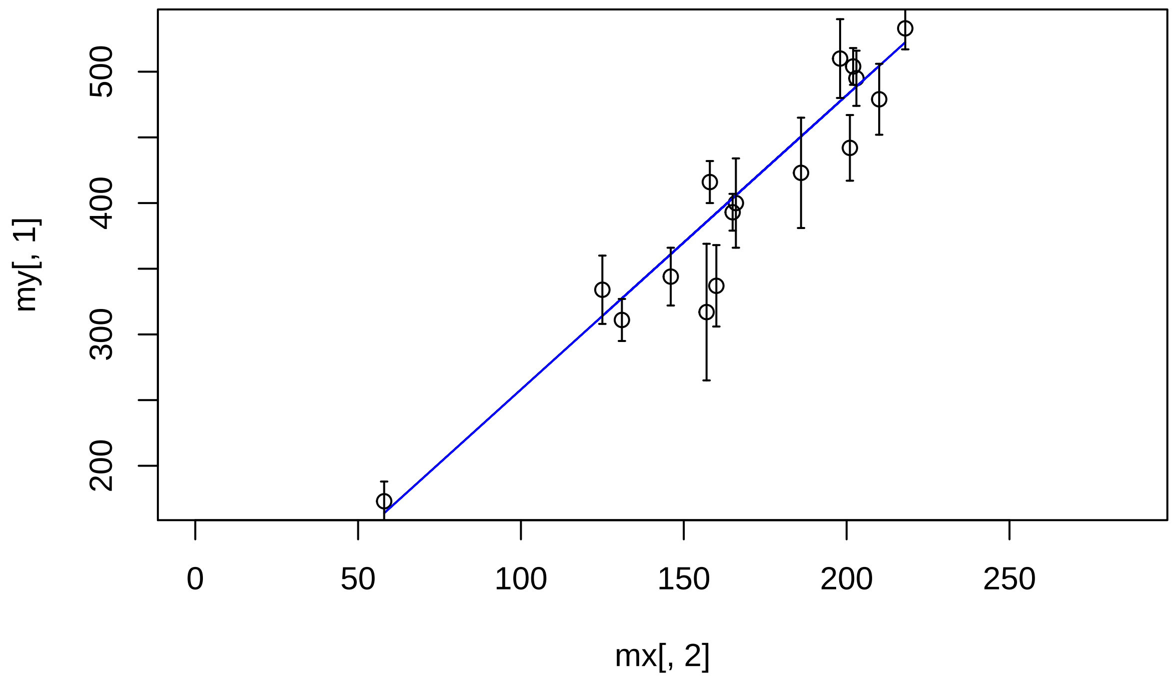
En este ejercicio lo que se nos pide es que usemos regresion lineal simple para hacer algo similar a la figura 1 usando un modelo lineal tal que

$$y = mx + b$$

Para hallar los parámetros W_{ML} vamos a hacer lo siguiente

$$\begin{bmatrix} b \\ m \end{bmatrix} = \left[A^T C^{-1} A\right]^{-1} A^T C^{-1} Y$$

```
mx <- cbind(rep(1,16),df[5:20,"x"]) # matriz de diseño
my <- matrix(df[5:20,"y"], ncol=1,nrow=16) # variable objetivo
model <- solve(t(mx) %*% solve(mstd2) %*% mx) %*% t(mx) %*% solve(mstd2) %*% my
predicty1 <- mx %*% model
plot(mx[,2], my[,1], xlim = c(0,max(x)))
lines(mx[,2], predicty1, col="blue")
segments(mx[,2], df[5:20,"uncertyn"], mx[,2], df[5:20, "uncertyp"])
segments(mx[,2]-1, df[5:20,"uncertyn"], mx[,2]+1, df[5:20, "uncertyn"])
segments(mx[,2]-1, df[5:20,"uncertyp"], mx[,2]+1, df[5:20, "uncertyp"])
```



Vemos que se ajusta correctamente a la figura 1. Los parámetros son

```
model
```

```
##           [,1]
## [1,] 34.047728
## [2,]  2.239921
```

Y la varianza o incertidumbre en m sería

```
solve(t(mx) %*% solve(mstd2) %*% mx)
```

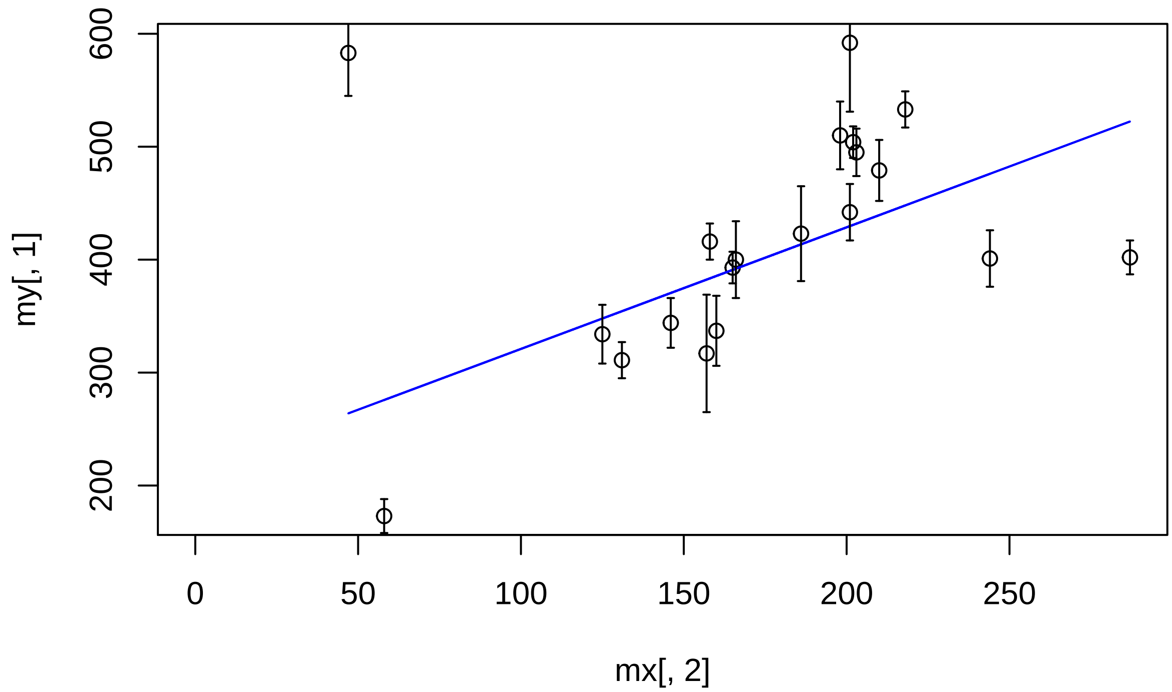
```
##            [,1]        [,2]
## [1,] 332.922601 -1.88954491
## [2,]  -1.889545  0.01161663
```

Ejercicio 2

En este ejercicio se trata de reproducir el ejercicio 1 pero tomando todos los puntos.

```
mx <- cbind(x^0, x) # matriz de diseño
my <- matrix(df[1:20,"y"], ncol=1,nrow=20) # variable objetivo
model <- model <- solve(t(mx) %*% solve(mstd1) %*% mx) %*% t(mx) %*% solve(mstd1) %*% my
predicty1 <- mx %*% model
```

```
plot(mx[,2], my[,1], xlim = c(0,max(x)))
lines(mx[,2], predicty1, col="blue")
segments(mx[,2], df[1:20,"uncertyn"], mx[,2], df[1:20, "uncertyp"])
segments(mx[,2]-1, df[1:20,"uncertyn"], mx[,2]+1, df[1:20, "uncertyn"])
segments(mx[,2]-1, df[1:20,"uncertyp"], mx[,2]+1, df[1:20, "uncertyp"])
```



Se ve que en este caso, al incluir los nuevos puntos el ajuste es mucho peor.

```
model
```

```
##          [,1]
##    213.273492
## x    1.076748
```

Y la varianza o incertidumbre en m sería

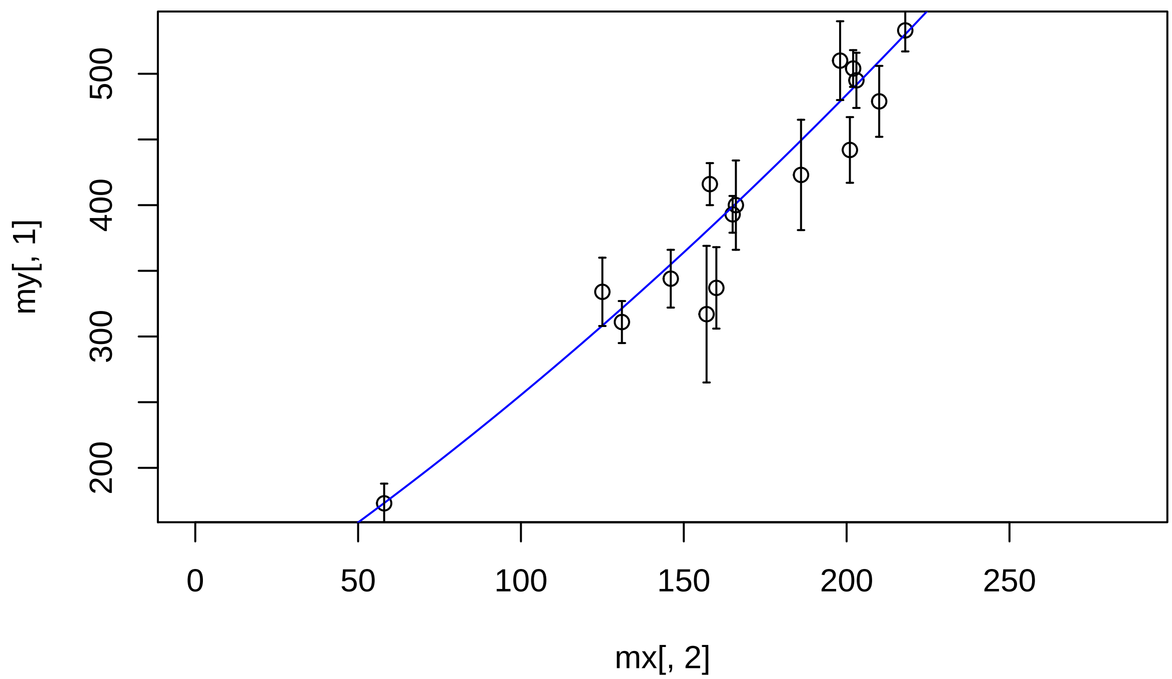
```
solve(t(mx) %*% solve(mstd1) %*% mx)
```

```
##                          x
##    207.188189 -1.05427206
## x   -1.054272  0.00599181
```

Ejercicio 3

En este caso se trata de usar una forma cuadratica de la expresión ignorando los primeros puntos al igual que en el ejercicio 1.

```r
nx <- seq(from=0,to=300, length.out=100)
mx <- cbind(rep(1,16),df[5:20,"x"], df[5:20,"x"]^2) # matriz de diseño
nmx <- cbind(nx^0, nx, nx^2) # nueva, sobre la que se usará el modelo
my <- matrix(df[5:20,"y"], ncol=1,nrow=16) # variable objetivo
model <- solve(t(mx) %*% solve(mstd2) %*% mx) %*% (t(mx) %*% solve(mstd2) %*%  my)
predicty1 <- nmx %*% model
plot(mx[,2], my[,1], xlim = c(0,max(x)))
lines(nx, predicty1, col="blue")
segments(mx[,2], df[5:20,"uncertyn"], mx[,2], df[5:20, "uncertyp"])
segments(mx[,2]-1, df[5:20,"uncertyn"], mx[,2]+1, df[5:20, "uncertyn"])
segments(mx[,2]-1, df[5:20,"uncertyp"], mx[,2]+1, df[5:20, "uncertyp"])
```



Y los parámetros W_{ML}

```r
model
```

```
##                [,1]
## [1,] 72.894626472
## [2,]  1.596050452
## [3,]  0.002298888
```

Ejercicio 4

Tomamos la expresión de partida para la log-verosimilitud de los datos dado el parámetro del modelo

$$\frac{1}{2T}\sum^{N}(y_i-\mu)^2$$

derivando e igualando a 0

$$0=\frac{1}{T}\left(\sum y_i-\sum\mu\right)$$

$$\frac{1}{T}\sum y_i=\frac{1}{T}N\mu$$
$$\frac{1}{N}\sum y_i=\mu_{ML}$$

Ejercicio 5

Tomar $(Y-AX)^T(Y-AX)$ (7 en Hogg fitting) y demostrar, derivando, que $X=\left[A^TC^{-1}A\right]^{-1}\left[A^TC^{-1}Y\right]$

Solución

Este ejercicio es igual al 3.3 de PRML de Bishop.

$$\frac{1}{2}\sum^{N}r_i\left(t_i-w^T\phi(x_i)\right)^2$$

derivando para obtener el gradiente, tal como en 3.13 de Bishop

$$\sum r_i\left(t_i-w^T\phi(x_i)\right)\phi(x_i)^T$$

igualando a 0 y reordenando

$$0=\sum r_it_i\phi(x_i)^T-w^T\left(\sum r_i\phi(x_i)\phi(x_i)^T\right)$$

que podemos expresarlo (resolviendo para w), si definimos $r_i=$ cada elemento de la diagonal de C^{-1} como

$$\left(\Phi^TC^{-1}\Phi\right)^{-1}\left(\Phi C^{-1}t\right)$$

Para demostrarlo basta con reordenar

$$\sum r_it_i\phi(x_i)^T-w^T\left(\sum r_i\phi(x_i)\phi(x_i)^T\right)$$
$$=\underbrace{\sum_i r_it_i\phi(x_i)^T}_{\Phi^TC^{-1}t}\underbrace{\left(\sum_i r_i\phi(x_i)\phi(x_i)^T\right)^{-1}}_{(\Phi^TC^{-1}\Phi)^{-1}}=w^T$$

haciendo la traspuesta ya tenemos el resultado final.