Tarea 3

Diego Alejandro Estrada Rivera 165352

Parte teórica

Supongamos que quieren explicar una variable estadística Y (por ejemplo altura) utilizando la información de p variables $X^1, \ldots, X^p$ (peso, ancho de huesos, etc). Si se toma una muestra de N individuos, cada variable está representada por un vector de tamaño N. La información de las variables explicativas se pueden juntar en una matriz

$$X = [X^1 \mid \ldots \mid X^p]$$

de tamaño n x p donde cada columna es una variable y cada fila uno de los individuos de la muestra. Tienen que contestar lo siguiente:

- **Plantear el problema de regresión lineal como un problema de mínimos cuadrados, encontrar el vector $\hat{\beta} = [\hat{\beta}_1, \ldots, \hat{\beta}_p]^T$ que resuelva**

$$\hat{\beta} = argmin_{\beta \in R^p} \mid\mid Y - X\beta \mid\mid^2$$

  y encontrar la solución teórica.

Realizamos el gradiente con respecto a Beta e igualamos a cero:

$$Gradiente: \mid\mid Y - X\beta \mid\mid^2 = Gradiente: (Y^T Y + \beta^T X^T X\beta - 2\beta^T X^T Y)$$

$$0 = 2X^T X\beta - 2X^T Y$$

Dividiendo ambos lados entre dos y resolviendo para Beta tenemos:

$$\beta_{OLS} = (X^T X)^{-1} X^T Y$$

¿Por qué este planteamiento nos da un ajuste lineal a nuestros datos?

Los estimadores β son lineales, es decir, estos coeficientes se utilizan para explicar la variable dependiente Y mediante los datos o variables independientes X.

¿Podríamos usarlo para ajustar polinomios (ej $y = x^2$)?

Podemos ajustar un polinomio $y = x^2$ usando el resultado de regresión lineal para las β's. Aunque la regresión de polinomio ajusta un modelo no-lineal, el problema de estimación estadística continúa siendo lineal, es decir, es lineal en las β's.

- **Argumentar la relación entre la solución encontrada y un problema de proyección en subespacios vectoriales de álgebra lineal ¿Cuál es la relación particular con el teorema de Pitágoras?**

El ajuste al que se llegó por regresión lineal es exactamente la proyección de Y en el subespacio generado por las columnas de X. Siendo esta proyección el punto en el subespacio de X más cercano a Y y el error siendo ortogonal al subespacio, se forma un triángulo rectángulo, siendo la proyección y el error los catetos y la Y la hipotenusa.

- **¿Qué logramos al agregar una columna de unos en la matriz X? es decir, definir mejor**

$$X = [1_n \mid X^1 \mid \ldots \mid X^p]$$

con $1_n = [1,1,\ldots,1]^T$

El estimador β_0 que se captura con la columna de 1´s nos es útil para considerar la información no contenida en las variables independientes que usamos para describir a la variable dependiente. De esta manera, el valor estimado $\hat{Y}$ no necesariamente inicia desde el origen, permitiendonos un mejor ajuste.

- **Plantear el problema de regresión ahora como un problema de estadística**

$$Y_i = \beta_0 + \beta_1 X_i^1 + \beta_2 X_i^2 + \ldots + \beta_p X_i^p + \epsilon_i$$

donde los errores son no correlacionados con distribución

- **¿Cuál es la función de verosimilitud del problema anteriror? Hint: empiecen por escribir el problema como**

$$\epsilon_i \sim N(0, \sigma^2)$$

$$Y = X\beta + \epsilon$$

con

$$\epsilon \sim N(0, \sigma^2 I_n)$$

con I_n la matriz identidad. Y concluyan entonces que

$$Y \sim N(X\beta, \sigma^2 I_n)$$

Escriban entonces la verosimilitud como

$$L(\beta, \sigma^2; Y, X) = \prod_{i=1}^{p} f_y\,(y_i|X; \beta, \sigma^2)$$

$$= \prod_{i=1}^{p} (2\pi\sigma^2)^{-1/2} exp(-\frac{1}{2}\frac{(y_i - x_i\beta)^2}{\sigma^2})$$

$$= (2\pi\sigma^2)^{-N/2} exp(-\frac{1}{2\sigma^2}\sum_{i=1}^{p}(y_i - x_i\beta)^2)$$

En términos matriciales:

$$L(\beta, \sigma^2; Y, X) = (2\pi\sigma^2)^{-N/2} exp(-\frac{1}{2\sigma^2}(Y - X\beta)^2)$$

$X\beta$ corresponde a la media de la distribución normal de Y, mientras que eñ término σ^2 en la formula se refiere a la varianza de la distribución.

- **Mostrar que la solución de máxima verosimilitud es la misma que la del problema de mínimos cuadrados.**

La función log de máxima verosimilitud es:

$$l(\beta, \sigma^2; Y, X) = -\frac{N}{2}ln(2\pi) - \frac{N}{2}ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - x_i\beta)^2$$

El siguiente paso es derivar respecto a cada una de las β:

$$(Gradiente\ de\ \beta): l(\beta, \sigma^2; Y, X)$$

$$(Gradiente\ de\ \beta): (-\frac{N}{2}ln(2\pi) - \frac{N}{2}ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - x_i\beta)^2)$$

$$= \frac{1}{\sigma^2}\sum_{i=1}^{N} x_i^T\,(y_i - x_i\beta)$$

$$= \frac{1}{\sigma^2}(\sum_{i=1}^{N} x_i^T\,y_i - \sum_{i=1}^{N} x_i^T\,x_i\beta)$$

Que es igual a cero solo si

$$\sum_{i=1}^{N} x_i^T \, y_i - \sum_{i=1}^{N} x_i^T \, x_i \beta = 0$$

Esto se satisface si:

$$\beta = (\sum_{i=1}^{N} x_i^T \, x_i)^{-1} \sum_{i=1}^{N} x_i^T \, y_i = (X^T X)^{-1} X^T y$$

- **Investiga el contenido del Teorema de Gauss-Markov sobre mínimos cuadrados.**

El Teorema de Gauss-Márkov establece que en un modelo lineal general (MLG) en el que se cumplan los siguientes supuestos:

- Correcta especificación: el MLG ha de ser una combinación lineal de los parámetros β y no necesariamente de las variables: $Y = X\beta + u$

- Muestreo aleatorio simple: la muestra de observaciones del vector $(y_i, x_{2i}, x_{3i}, \ldots, x_{ki})$ es una muestra aleatoria simple y, por lo tanto, el vector (y_i, X'_i) es independiente del vector (y_i, X'_j)
- Esperanza condicionada de los errores nula: $E(u_i|X'_i) = 0$

- Correcta identificación: la matriz de regresoras (X) ha de tener rango completo: rg(X)=K<=N

- Homocedasticidad: la varianza del error condicional a las variables explicativas es constante a lo largo de las observaciones:$Var(U|X) = \sigma^2 I$

el estimador mínimo cuadrático ordinario (MCO) de β es el estimador lineal e insesgado óptimo, es decir, el estimador MCO es el estimador eficiente dentro de la clase de estimadores lineales e insesgados.

Parte aplicada

Para esta parte pueden usar la base de datos *diamonds* que sugirieron, aunque hay puntos adicionales si usan alguna base original interesante.

Cargar la base que se encuentra en el paquete ggplot2. Los comandos que pueden usar para cargar la base diamonds a su ambiente de trabajo en R son:

```
#install.packages("ggplot2")
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.1
```

```
data(diamonds)
head(diamonds)
```

```
## # A tibble: 6 x 10
##   carat       cut color clarity depth table price     x     y     z
##   <dbl>     <ord> <ord>   <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23     Ideal     E     SI2  61.5    55   326  3.95  3.98  2.43
## 2  0.21   Premium     E     SI1  59.8    61   326  3.89  3.84  2.31
## 3  0.23      Good     E     VS1  56.9    65   327  4.05  4.07  2.31
## 4  0.29   Premium     I     VS2  62.4    58   334  4.20  4.23  2.63
## 5  0.31      Good     J     SI2  63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good     J    VVS2  62.8    57   336  3.94  3.96  2.48
```

```
#?diamonds
```

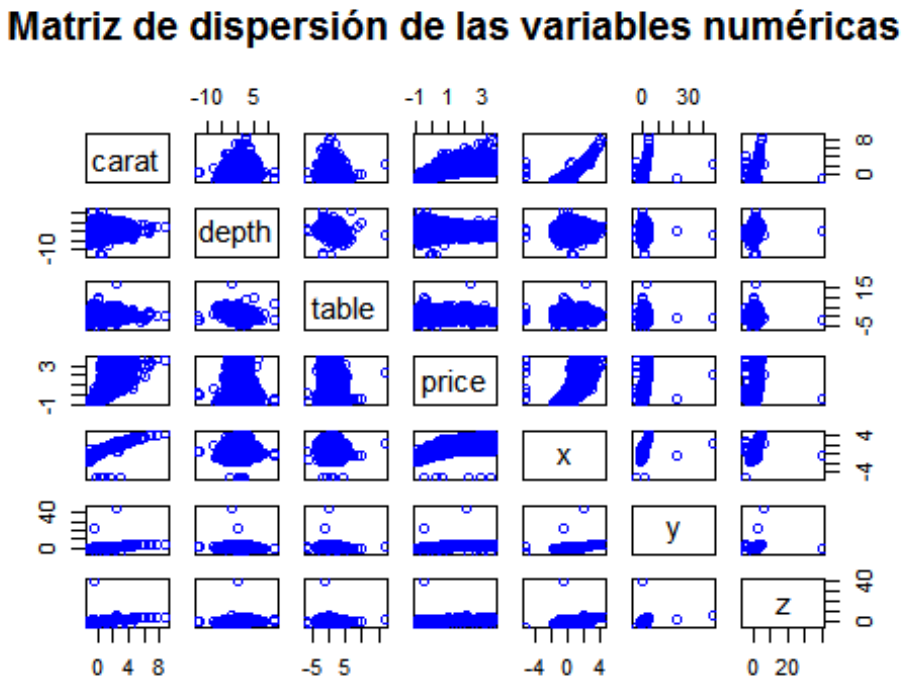
Posteriormente deben hacer una regresión lineal. Su objetivo es explicar la variable price usando las demás variables. Noten que algunas variables no son numéricas, por lo que no pueden incluirse en un análisis crudo de regresión lineal. Para este proyecto NO es necesario saber transformar las variables no numéricas para poder usarlas en la regresión; hacerlo es optativo, de hecho, las paqueterías lo hacen por ustedes pero deben ser cuidadosos. Pueden usar la función lm de R para su análisis de regresión.

Creamos un nuevo data_frame que solo contiene los datos que vamos a utilizar, es decir, las variables numéricas centralizadas

```
diamantes = diamonds[ ,c(1,5,6,7,8,9,10)]
d = scale(diamantes)
d <- as.data.frame(d)
```

Obtenemos una matriz de dispersión para formarnos una apreciación inicial de las relaciones entre las variables.

```
plot(d, col= "blue", main="Matriz de dispersión de las variables numérica
s")
```

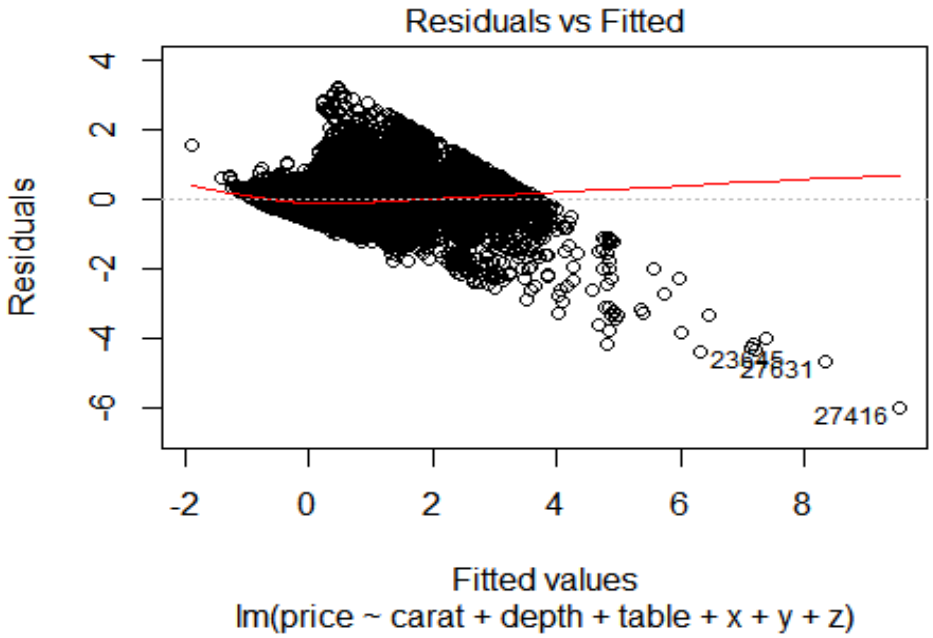


Gracias a la matriz de las variables en pares podemos discernir de forma rápida una relación positiva entre el precio y la variable "carat", aunque la varianza aumenta conforme aumenta esta variable. También podemos observar que no hay mucha relación entre el precio y las variables "y"(ancho en mm) y "z"(profundidad en mm).

```r
#Realizamos la regresión lineal usando todas las variables numéricas
mod = lm(price ~ carat + depth + table + x + y + z, data = d)
summary(mod)
```

```
##
## Call:
## lm(formula = price ~ carat + depth + table + x + y + z, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9853 -0.1542 -0.0127  0.0872  3.1982
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.359e-14  1.616e-03   0.000  1.00000
## carat        1.270e+00  7.509e-03 169.085  < 2e-16 ***
## depth       -7.295e-02  1.976e-03 -36.910  < 2e-16 ***
## table       -5.738e-02  1.727e-03 -33.216  < 2e-16 ***
## x           -3.699e-01  1.211e-02 -30.547  < 2e-16 ***
## y            1.899e-02  7.307e-03   2.599  0.00937 **
## z            7.364e-03  7.837e-03   0.940  0.34744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3752 on 53933 degrees of freedom
## Multiple R-squared:  0.8592, Adjusted R-squared:  0.8592
## F-statistic: 5.486e+04 on 6 and 53933 DF,  p-value: < 2.2e-16
```

```r
plot(mod, which =1)
```



- **¿Qué tan bueno fue el ajuste?**

Del "summary" de la regresión podemos observar inicialmente la información de los residuales, que son todas las diferencias entre el precio estimado y el precio real; una media lo más cercana a cero es lo que querríamos ver aquí, pues indicaría que el modelo es muy certero. En la gráfica vemos los valores predichos contra los residuales; en un modelo perfecto, la media de los residuales sería 0 por lo que la línea roja iría perfectamente a lo largo de la línea que representa el 0. En seguida podemos apreciar los coeficientes de la intercepción y de cada una de las variables independientes, usadas para calcular las predicciones. Así también confirmamos que las variables "z" y "y" son las menos importantes para el modelo, esto lo sabemos por el valor p en la columna "Pr(>|t|)", que se resta a 1 para conocer la probabilidad de que NO sean relevantes al modelo, es decir, para todos las variables queremos que este sea lo más pequeño posible. En general, podemos decir que el ajuste es bueno, sobre todo por el valor de R cuadrada que se explica en siguiente punto.

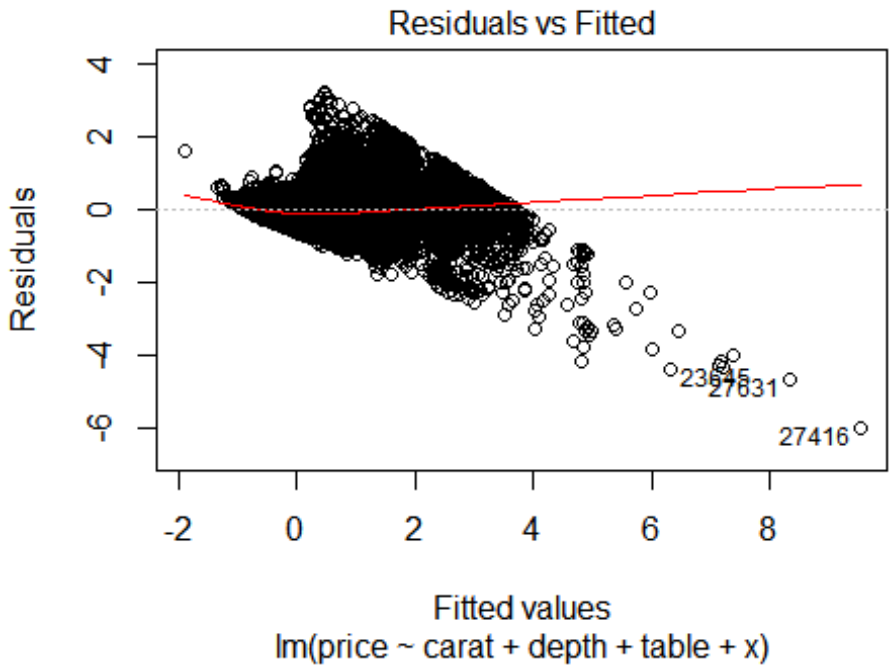
- **¿Qué medida puede ayudarnos a saber la calidad del ajuste? ¿Cuál fue el valor de $ σ^2 $ que ajustó su modelo y qué relación tienen con la calidad del ajuste?**

Al final del summary podemos apreciar la R cuadrada, que es la cantidad de variabilidad en lo que estas prediciendo que es explicado por el modelo, en este caso podríamos decir que un 85.92 % (.8592) del precio de los diamantes es explicado por las variables en nuestro modelo. Es una buena medida de la calidad del ajuste.

```r
# Por curiosidad, realizamos otra regresión lineal, dejando fuera a "y" y
"z", que no aportaban mucho en el modelo anterior
mod1 = lm(price ~ carat + depth + table + x, data = d)
summary(mod1)

##
## Call:
## lm(formula = price ~ carat + depth + table + x, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9894 -0.1541 -0.0127  0.0869  3.1986
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.374e-14  1.616e-03    0.00        1
## carat        1.270e+00  7.505e-03  169.27   <2e-16 ***
## depth       -7.226e-02  1.742e-03  -41.48   <2e-16 ***
## table       -5.759e-02  1.726e-03  -33.37   <2e-16 ***
## x           -3.449e-01  7.501e-03  -45.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3753 on 53935 degrees of freedom
## Multiple R-squared:  0.8592, Adjusted R-squared:  0.8592
## F-statistic: 8.228e+04 on 4 and 53935 DF,  p-value: < 2.2e-16

plot(mod1, which=1)
```

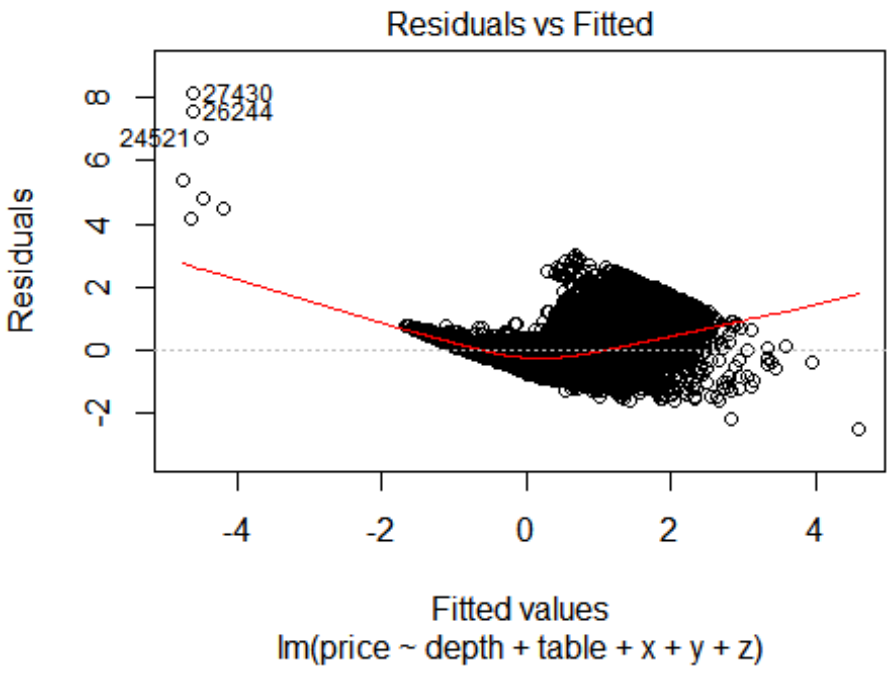


Aquí vemos, como era esperado, que deshacernos de las dos variables que no eran relevantes (y,z) prácticamente no modifica valores como la media de los residuales o la R cuadrada.

```r
# Por curiosidad, ahora realizamos otra regresión lineal, pero ahora deja
mos fuera la variable "carat", que es una variable relevante para el mode
lo
mod2 = lm(price ~ depth + table + x + y + z, data = d)
summary(mod2)
```

```
##
## Call:
## lm(formula = price ~ depth + table + x + y + z, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5052 -0.3148 -0.0494  0.2370  8.1391
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.179e-14  1.998e-03   0.000   1.0000
## depth       -3.771e-03  2.392e-03  -1.576   0.1149
## table       -4.753e-02  2.136e-03 -22.255  < 2e-16 ***
## x            8.206e-01  1.219e-02  67.330  < 2e-16 ***
## y            5.871e-02  9.034e-03   6.499 8.13e-11 ***
## z            1.624e-02  9.694e-03   1.675   0.0939 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4641 on 53934 degrees of freedom
## Multiple R-squared:  0.7846, Adjusted R-squared:  0.7846
## F-statistic: 3.929e+04 on 5 and 53934 DF,  p-value: < 2.2e-16
```

```r
plot(mod2, which=1)
```



Como era esperado, remover una variable importante (carat), disminuyó el valor de R cuadrada y aumentó la media de los residuales (cuyo efecto puede verse en la recta de la gráfica de residuales).

- **¿Cuál es el ángulo entre Y y $\hat{Y}$? Hint: usen la R^2 cuadrada y el arcocoseno**

```
angulo <- acos(sqrt(.8592))
angulo * 180/pi
```

```
## [1] 22.03873
```

- **Definan una función que calcule la logverosimilitud de unos parámetros β y σ^2.**

```
diamonds_data = data(diamonds)
diamonds_short <-  diamonds[ ,c(1,5,6,7,8,9,10)]
diamonds_x <- diamonds[ ,c(5,6,7,8,9,10)]
head(diamonds)
```

```
## # A tibble: 6 x 10
##   carat       cut color clarity depth table price     x     y     z
##   <dbl>     <ord> <ord>   <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23     Ideal     E     SI2  61.5    55   326  3.95  3.98  2.43
## 2  0.21   Premium     E     SI1  59.8    61   326  3.89  3.84  2.31
## 3  0.23      Good     E     VS1  56.9    65   327  4.05  4.07  2.31
## 4  0.29   Premium     I     VS2  62.4    58   334  4.20  4.23  2.63
## 5  0.31      Good     J     SI2  63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good     J    VVS2  62.8    57   336  3.94  3.96  2.48
```

```
diamonds_m <- data.matrix(diamonds_x)

n <- length(diamonds_x )
sigma_sq <- 0.8563
mod <- lm(formula = diamonds$price ~ diamonds$carat + diamonds$x + diamon
ds$y + diamonds$z + diamonds$depth)
summary(mod)$coefficients[,1]
```

```
##    (Intercept) diamonds$carat     diamonds$x     diamonds$y     diamon
ds$z
##    12196.68697    10615.49551    -1369.67016       97.59636       64.1
9545
## diamonds$depth
##     -156.62430
```

```
Beta_1 <- c(12196.7, 10615.5, -1369.7, 97.6, 64.2, -156.6)

funcionVerosimilitud <- function(bet, sig){
  -(n/2)*(log(2*pi))-((n/2)*log(sig))-((1/(2*sig))*((diamonds$price-(diam
onds_m*bet))*(diamonds$price-(diamonds_m*bet))))
}

funcionVerosimilitud(Beta_1,sigma_sq)
```

- **Utilicen la función optim de R para numéricamente el máximo de la función de verosimilitud. Si lo hacen correctamente, su solución debe coincidir con la del método lm.**