Curso Propedéutico en Ciencias de Datos ITAM

Tarea 3: Regresión Lineal

Ariel Vallarino

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

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

de tamaño n × 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 como un problema de mínimos cuadrados, encontrar el vector

$$\beta = [\beta 1, \ldots, \beta p]$$

que resuelva

$$\widehat{\beta} = argmin_{\beta \in R^p} \ |Y - X\beta|^2$$

y encontrar la solución teórica.
¿Por qué este planteamiento nos da un ajuste lineal a nuestros datos?
¿Podríamos usarlo para ajustar polinomios (ej y = x2)?

El estimador por mínimos cuadrados de β se obtiene minimizando la suma de los residuos al cuadrado.

$$X^T \vec{Y} = X^T X \widehat{\beta}$$

Derivando e igualando a cero se obtienen las escuaciones de regresión

$$\widehat{\beta} = (X^T X)^{-1} X^T \vec{Y}$$

Este planteamiento da un ajuste lineal porque cumple con el principio de Linealidad. Es decir, los valores de la variable dependiente están generados por un modelo lineal del tipo: $Y = X * B + U$
Cuando los datos no son representados por una linea recta, pueden ajustarse mediante una curva. En tales casos se utiliza la regresión polinomial.

• 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?

La solución es la proyección de Y en el en subespacios de columnas de X. Este punto minimiza el error formando un triangulo rectangulo.

• ¿Qué logramos al agregar una columna de unos en la matriz X? Es decir, definir mejor X=[1n ∣X1 ∣...∣Xp] con 1n = [1,1,...,1]⊤.

Agregar una columna de 1's evita que la proyección pase por el origen y mejora el ajuste considerando el valor del β_0

• Plantear el problema de regresión ahora como un problema de estadística

$$Y_i = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon_i$$

donde los errores son no correlacionados con distribución

εi ∼N(0,σ2)

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

$$Y = X\beta + \varepsilon$$

con

$$\varepsilon \sim N(0, \sigma 2 In)$$

con In la matriz identidad. Y concluyan entonces que Y ∼N(Xβ,σ2In)
Escriban entonces la verosimilitud como

$$L(\beta, \sigma^2) = f(Y \mid \beta, \sigma^2, X)$$

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

> Por MLE tambien se llega a
>
> $$\widehat{\beta} = (X^T X)^{-1} X^T \vec{Y}$$

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

> El teorema de Gauss-Markov establece que el método de estimación de mínimos cuadrados va a producir estimadores óptimos, en el sentido que los parámetros estimados van a estar centrados y van a ser de mínima varianza.

Parte aplicada:

Cargar la base diamonds que se encuentra en el paquete ggplot2

```
#install.packages("ggplot2") # solo si necesario...
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

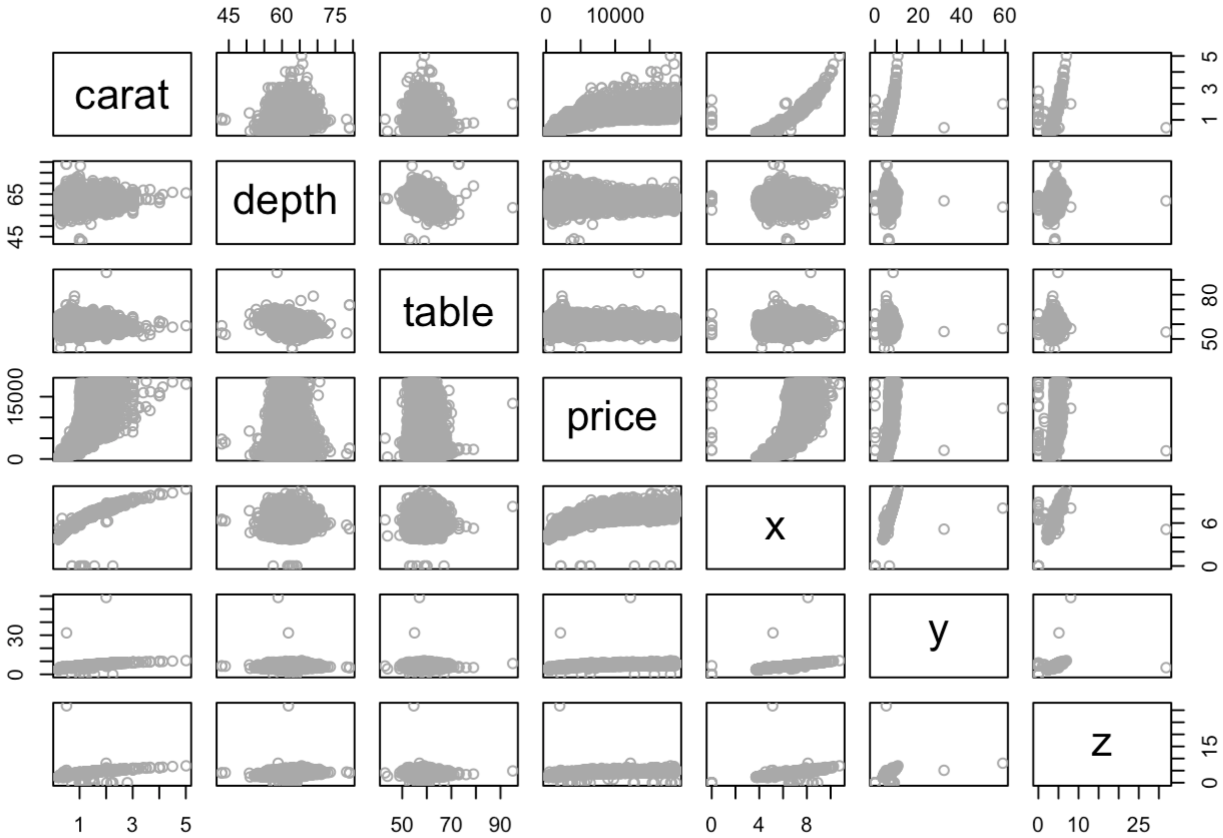
```
data(diamonds)
head(diamonds)
```

```
## # 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
```

Regresión lineal.

Explicar la variable price usando los demás datos.

```
# Grafico relacion entre las variables
diamonds_n <- diamonds[c(1,5:10)]
plot(diamonds_n, col = 'darkgray')
```




```
# Construir un modelo lineal del Precio en funcion de las demas variables numericas
modelo <- lm(price ~ carat + depth + table + x + y + z, data = diamonds_n)
modelo
```

```
##
## Call:
## lm(formula = price ~ carat + depth + table + x + y + z, data = diamonds_n)
##
## Coefficients:
## (Intercept)        carat        depth        table            x
##    20849.32     10686.31      -203.15      -102.45     -1315.67
##           y            z
##       66.32        41.63
```

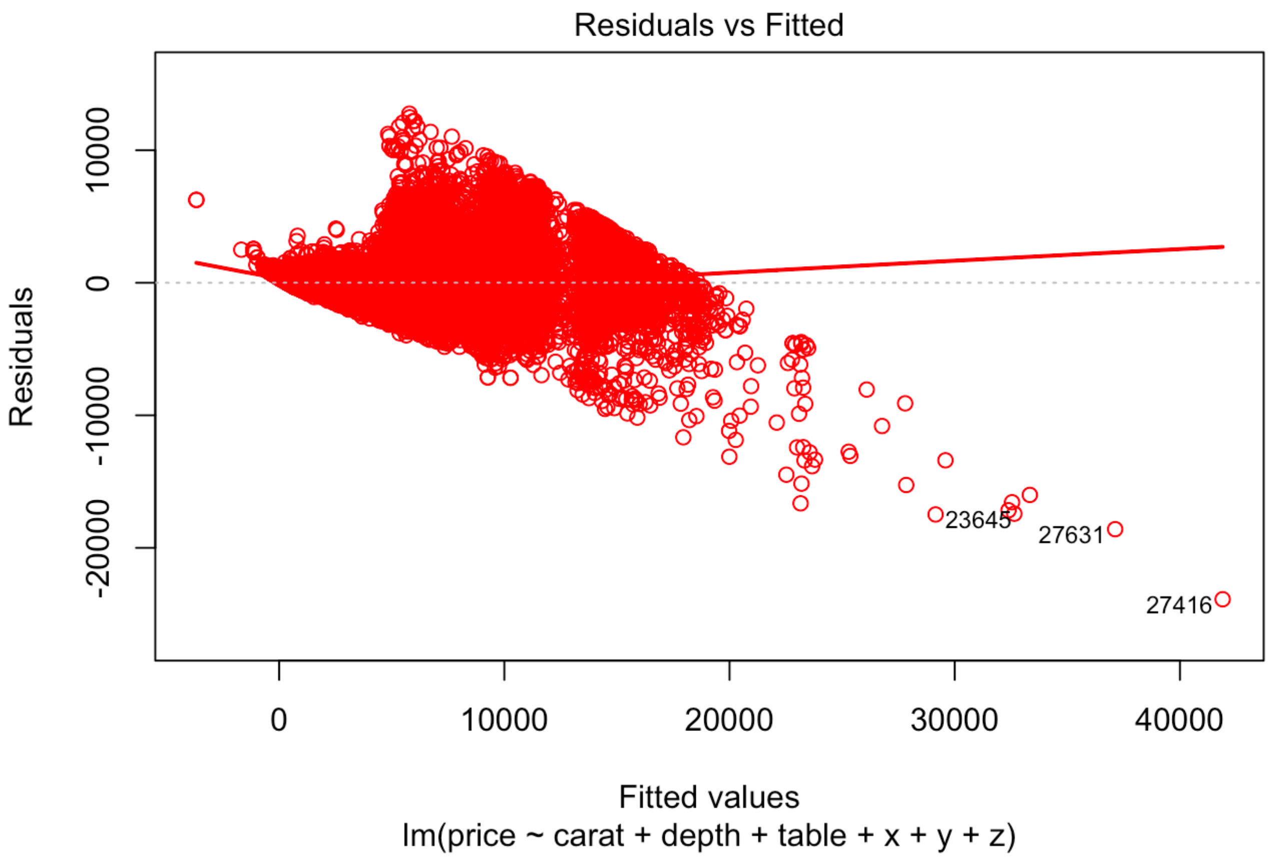
```
summary(modelo)
```

```
##
## Call:
## lm(formula = price ~ carat + depth + table + x + y + z, data = diamonds_n)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23878.2   -615.0    -50.7    347.9  12759.2
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20849.316    447.562  46.584  < 2e-16 ***
## carat       10686.309     63.201 169.085  < 2e-16 ***
## depth        -203.154      5.504 -36.910  < 2e-16 ***
## table        -102.446      3.084 -33.216  < 2e-16 ***
## x           -1315.668     43.070 -30.547  < 2e-16 ***
## y              66.322     25.523   2.599  0.00937 **
## z              41.628     44.305   0.940  0.34744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1497 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
```

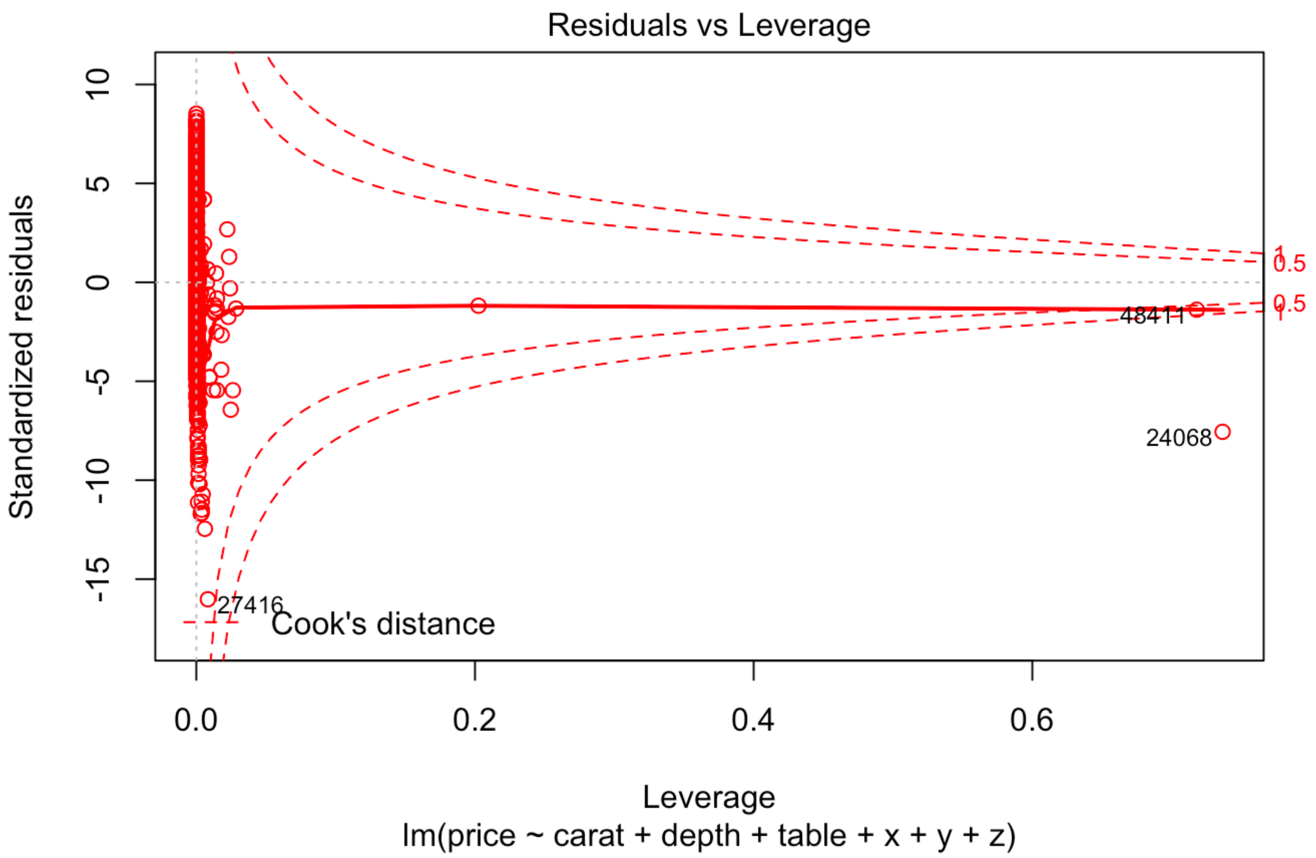
```
cor(diamonds_n)
```

```
##            carat       depth      table      price           x           y
## carat 1.00000000  0.02822431  0.1816175  0.9215913  0.97509423  0.95172220
## depth 0.02822431  1.00000000 -0.2957785 -0.0106474 -0.02528925 -0.02934067
## table 0.18161755 -0.29577852  1.0000000  0.1271339  0.19534428  0.18376015
## price 0.92159130 -0.01064740  0.1271339  1.0000000  0.88443516  0.86542090
## x     0.97509423 -0.02528925  0.1953443  0.8844352  1.00000000  0.97470148
## y     0.95172220 -0.02934067  0.1837601  0.8654209  0.97470148  1.00000000
## z     0.95338738  0.09492388  0.1509287  0.8612494  0.97077180  0.95200572
##                z
## carat 0.95338738
## depth 0.09492388
## table 0.15092869
## price 0.86124944
## x     0.97077180
## y     0.95200572
## z     1.00000000
```

```
# Grafico recta de lm junto con el grafico anterior
plot(lm(price ~ carat + depth + table + x + y + z, data = diamonds_n), lwd = 2, col =
 'red', which = 1)
```




```
# Grafico recta de lm junto con el grafico anterior
plot(lm(price ~ carat + depth + table + x + y + z, data = diamonds_n), lwd = 2, col =
 'red', which = 5)
```



- ¿Qué tan bueno fue el ajuste? Una buena respuesta incluye argumentaciones teóricas y visualizaciones. Puntos adicionales si investigan como usar alguna de las librerias ggplot2 o plotly para sus gráficas.

- ¿Qué medida puede ayudarnos a saber la calidad del ajuste? ¿Cuál fue el valor de σ2 que ajustó su modelo y que relación tiene con la calidad del ajuste? σ^2

> El coeficiente de determinación R-squared que determina la calida del modelo indica un %85.92 (0.8592), considerando que las variables NO numericas no han sido utilizadas.

- ¿Cuál es el ángulo entre Y y $ widehat{Y} $?. Hint: usen la R y el arcocoseno.

```
angulo <- acos(sqrt(0.8592))
angulo
```

```
## [1] 0.3846484
```

- Defininan una funcion que calcule la logverosimilitud de unos parámetros β y σ2 .

- 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 .