Tarea 3

1. Sección teórica

1.1. Plantear el problema de regresión como uno de mínimos cuadrados

Al buscar una $\hat{\beta}$ que minimice la norma del vector ε que es la diferencia entre Y y $X\beta$, el problema, como se plantea es el siguiente:

$$\hat{\beta} = argmax_{\beta \epsilon \Re^p} \|Y - X\beta\|^2$$

Partiendo de este punto, desarrollemos estas ecuaciones:

$$\|Y - X\beta\|^2$$

$$\left(\sqrt{(Y - X\beta)(Y - X\beta)}\right)^2 = (Y - X\beta)(Y - X\beta)$$

$$Y'Y - \beta'X'Y - Y'X\beta + \beta'X'X\beta$$

Donde $(-\beta'X'Y)_{1\times1}$ dado que $\beta'_{1\times k}$, $X'_{k\times n}$ y $Y_{n\times1}$ y de la misma forma $(-Y'X\beta)_{1\times1}$. Por lo tanto, definamos la función $F_{(\beta)}$

$$F_{(\beta)} = Y'Y - 2Y'X\beta + \beta'X'X\beta$$

Si derivamos con respecto a β, obtenemos:

$$\frac{dF_{(\beta)}}{d\beta} = -2Y'X\beta + \beta'X'X\beta$$

Y donde su condición de primer orden son las ecuaciones normales de mínimos cuadrados ordinarios:

$$(X'X)\beta = X'Y$$

Y si X'X es invertible, la solución teórica a este problema es:

$$\hat{\beta} = (X'X)^{-1}X'Y$$

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

Este planteamiento nos da un ajuste lineal a los datos porque el vector $\hat{\beta}$ óptimo obtenido a través del problema anterior en combinación lineal con las columnas X permiten construir Y al mismo tiempo que se tiene una distancia (norma) entre Y y $X\hat{\beta}$ mínima.

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

Sí, porque el planteamiento de mínimo cuadrados es lineal en el sentido de que se pueda expresar Y como una combinación lineal de la información en X.

1.2. Argumentar la relación entre la solución encontrada y un problema de proyección en subespacios vectoriales de álgebra lineal.

La solución encontrada es una proyección de la variable Y sobre el espacio vectorial de las X. El problema en sí es encontrar la proyección de ese vector Y de tal forma que la distancia entre la proyección y el vector sea mínima.

El error, o la diferencia entre esa proyección y el vector es lo que estamos minimizando, ya sea interpretándolo como una norma mínima o como una varianza mínima.

¿Cuál es la relación particular con el teorema de Pitágoras?

Otra forma de escribir las ecuaciones normales de mínimos cuadrados es:

$$(X'X)\beta = X'Y$$

$$0 = X'Y - X'X\beta = X'(Y - X\beta)$$

Que es igual a:

$$X'\varepsilon = 0$$

Es decir, las información de las variables explicativas y los errores son ortogonales, pues su producto punto es 0.

Visto de otra forma, cuando se tiene una β que minimiza la distancia entre el vector Y y su proyección $X\beta$ en el plano de las X, ε es ortogonal a esa proyección.

Su relación con el teorema de pitágoras es que dada la fórmula para calcular el ángulo entre dos vectores:

$$cos\theta = \frac{< a, b >}{\|a\|\|b\|}$$

En el caso de que θ es $\frac{\pi}{2}$ (ángulo recto), se requiere que $< a, b >$ sea igual a cero. En este caso, $X'\varepsilon$ son ortogonales y representan dos catetos con Y de hipotenusa.

Cuando se cumple esta condición, significa que $X\beta$ es una proyección de Y sobre el plano de las X.

1.3. ¿Qué logramos al agregar una columna de unos en la matriz X?

Dado que X solo está compuesto de columnas con distintas variables, toda combinación lineal de ellas que está forzada a pasar por el vector 0 cuando los coeficientes β_i sean iguales a cero

1.4. Plantear el problema de regresión ahora como un problema de estadística

Desde una perspectiva estadística, el problema consiste en poder explicar una variable a través de una combinación lineal de otras variables, minimizando el error que existe entre el resultado de esa combinación lineal y la variable.

$$Y_i = X_i\beta_i + \varepsilon_i$$

Dado que ε_i puede tomar valores positivos o negativos, para encontrar un conjunto de parámetros β que junto con las observaciones X y minimicen el error entre la predicción y lo observado, debe usarse el término de error al cuadrado, de tal forma que:

$$\sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (Y_i - Xi\beta_i)^2$$

1.5. ¿Cuál es la función de verosimilitud del problema anterior?

Asumiendo que cada una de las desviaciones $\epsilon_i \sim (N, \sigma^2)$, entonces la función de verosimilitud es:

$$\prod_{i=1}^{n} L_{i(\beta,\sigma^2)} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(Y_i - X_i\beta)^2}{2\sigma^2}}$$

O bien,

$$L_{(\beta,\sigma^2)} = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n e^{-\frac{\sum_{i=1}^{n}(Y_i - X_i\beta)^2}{2\sigma^2}}$$

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

Obteniendo logaritmo (transformación lineal que no afecta el orden en la función) de la ecuación anterior:

$$lnL_{(\beta,\sigma^2)} = -\frac{n}{2}ln(2\pi) - \frac{n}{2}ln(\sigma^2) - \frac{\sum_{i=1}^{n}(Y_i - X_i\beta)^2}{2\sigma^2}$$

Si se deriva L con respecto a los parámetros para obtener parámetros óptimos en los que se maximiza la probabilidad de que las ε provienen de esa distribución:

$$\frac{\partial lnL_{(\beta,\sigma^2)}}{\partial\beta} = \frac{\partial}{\partial\beta}\left(-\frac{\sum_{i=1}^{n}(Y_i - X_i\beta)^2}{2\sigma^2}\right) = 0$$

Dado que $-2\sigma^2$ no depende de β, esta derivada parcial tiene la misma forma que otras formas de ver el mismo problema:

$$\frac{\partial}{\partial\beta}\left(\sum_{i=1}^{n}(Y_i - X_i\beta)^2\right) = \frac{\partial}{\partial\beta}\left((Y - X\beta)'(Y - X\beta)\right) = \frac{\partial}{\partial\beta}\left(Y'Y - \beta'X'Y - Y'X\beta + \beta'X'X\beta\right) = 0$$

$$\frac{\partial}{\partial\beta}\left(-2Y'X\beta + \beta'X'X\beta\right) = 0$$

$$X'Y = (X'X)\beta$$

$$\hat{\beta}_{MV} = (X'X)^{-1}X'Y = \hat{\beta}_{MC}$$

Nota: no olvidar que el parámetro de varianza que se obtiene a través de Máxima Verosimilitud es distinta de la que se obtiene por implicación a través de mínimos cuadrados ordinarios.

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

El teorema de Gauss Markov establece que el estimador de mínimos cuadrados ordinarios es el estimador lineal insesgado de β de mínima varianza.

2. Regresión lineal

```r
library(ggplot2)
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
```

Primero veamos qué hay en la base:

```r
str(diamonds)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    53940 obs. of  10 variables:
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
##  $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
##  $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
##  $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

```r
summary(diamonds)
```

```
##      carat               cut        color        clarity
##  Min.   :0.2000   Fair     : 1610   D: 6775   SI1    :13065
##  1st Qu.:0.4000   Good     : 4906   E: 9797   VS2    :12258
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194
##  Mean   :0.7979   Premium  :13791   G:11292   VS1    : 8171
##  3rd Qu.:1.0400   Ideal    :21551   H: 8304   VVS2   : 5066
##  Max.   :5.0100                     I: 5422   VVS1   : 3655
##                                     J: 2808   (Other): 2531
##      depth           table           price             x
##  Min.   :43.00   Min.   :43.00   Min.   :  326   Min.   : 0.000
##  1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710
##  Median :61.80   Median :57.00   Median : 2401   Median : 5.700
##  Mean   :61.75   Mean   :57.46   Mean   : 3933   Mean   : 5.731
##  3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540
##  Max.   :79.00   Max.   :95.00   Max.   :18823   Max.   :10.740
##
##        y                z
##  Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 4.720   1st Qu.: 2.910
##  Median : 5.710   Median : 3.530
##  Mean   : 5.735   Mean   : 3.539
```

```
##  3rd Qu.: 6.540   3rd Qu.: 4.040
##  Max.   :58.900   Max.   :31.800
##
```

Parece que las variables numéricas son `price` (que es int), `carat`,`depth`,`table`,`x`,`y`, y `z`; y las variables categóricas son `cut`, `color` y `clarity`.

Pasemos `price` a numérica

```
diamonds$price <- as.numeric(diamonds$price)
```

Ahora construyamos un modelo lineal usando solo las variables numéricas.

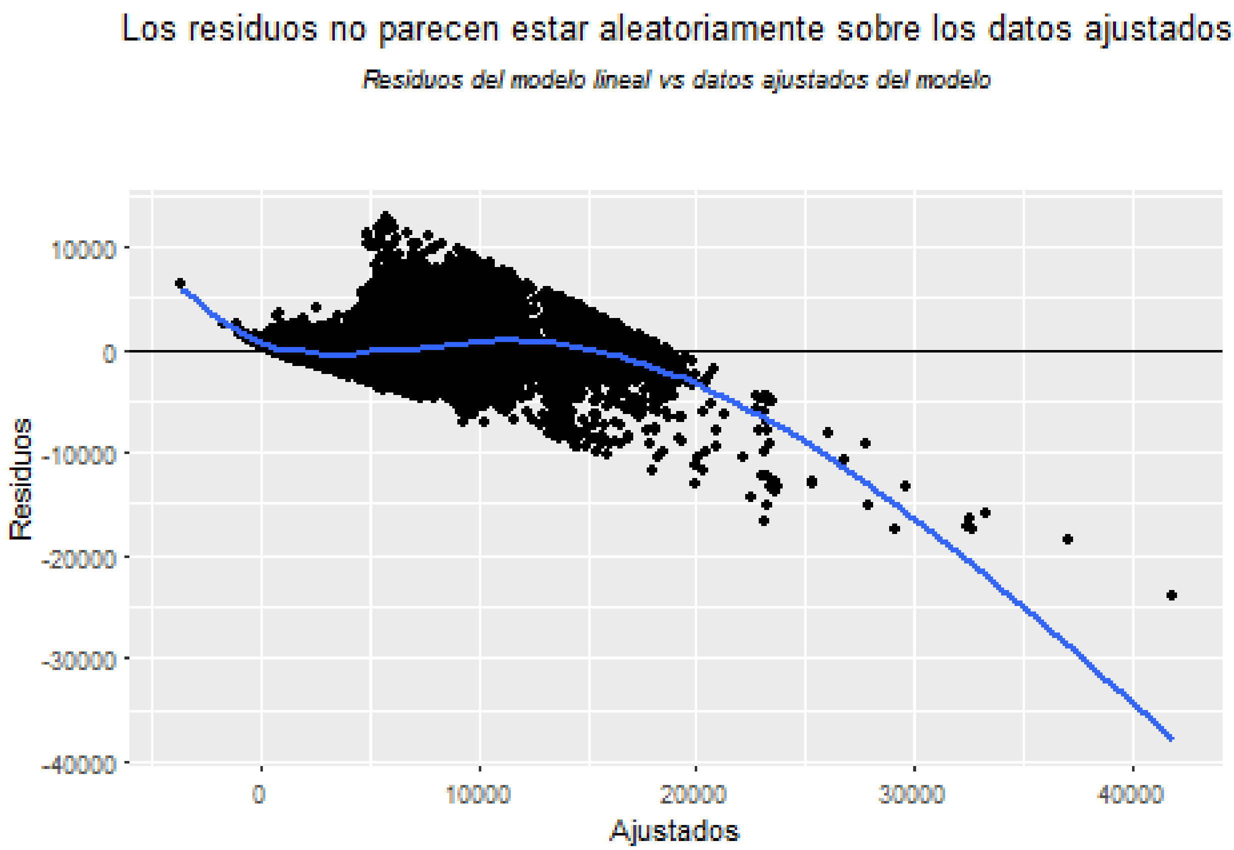
```
modelonum <- lm(diamonds$price ~ diamonds$carat + diamonds$depth + diamonds$table + diamonds$x + diamond
summary(modelonum)
```

```
##
## Call:
## lm(formula = diamonds$price ~ diamonds$carat + diamonds$depth +
##     diamonds$table + diamonds$x + diamonds$y + diamonds$z)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23878.2   -615.0    -50.7    347.9  12759.2
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    20849.316    447.562  46.584  < 2e-16 ***
## diamonds$carat 10686.309     63.201 169.085  < 2e-16 ***
## diamonds$depth  -203.154      5.504 -36.910  < 2e-16 ***
## diamonds$table  -102.446      3.084 -33.216  < 2e-16 ***
## diamonds$x     -1315.668     43.070 -30.547  < 2e-16 ***
## diamonds$y        66.322     25.523   2.599  0.00937 **
## diamonds$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
```

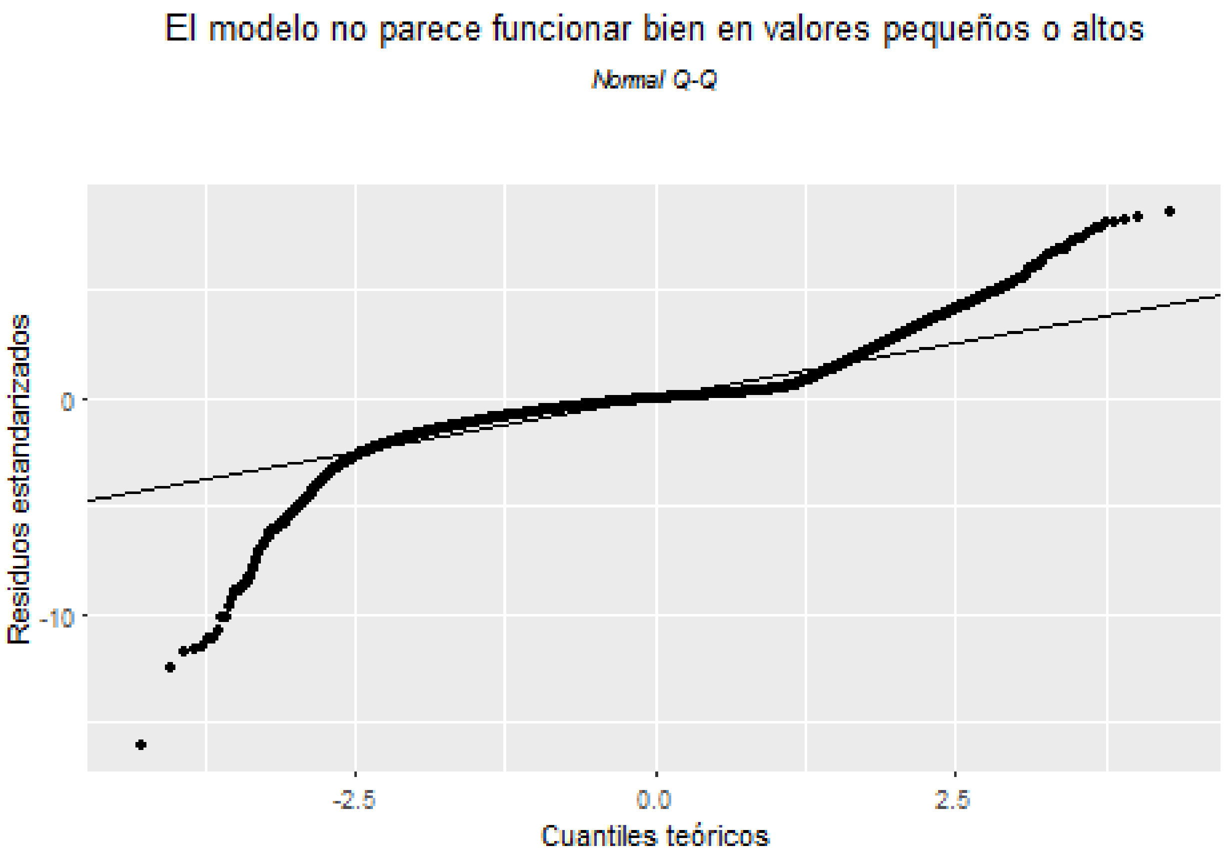
Ahora hagamos una visualización con `ggplot` que se vea *bien*.

```
library(ggplot2)
ggplot(modelonum, aes(.fitted, .resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_smooth(se = FALSE) +
  labs(x ="Ajustados", y = "Residuos") +
  ggtitle(expression(atop("Los residuos no parecen estar aleatoriamente sobre los datos ajustados", atop
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using method = 'gam'
```

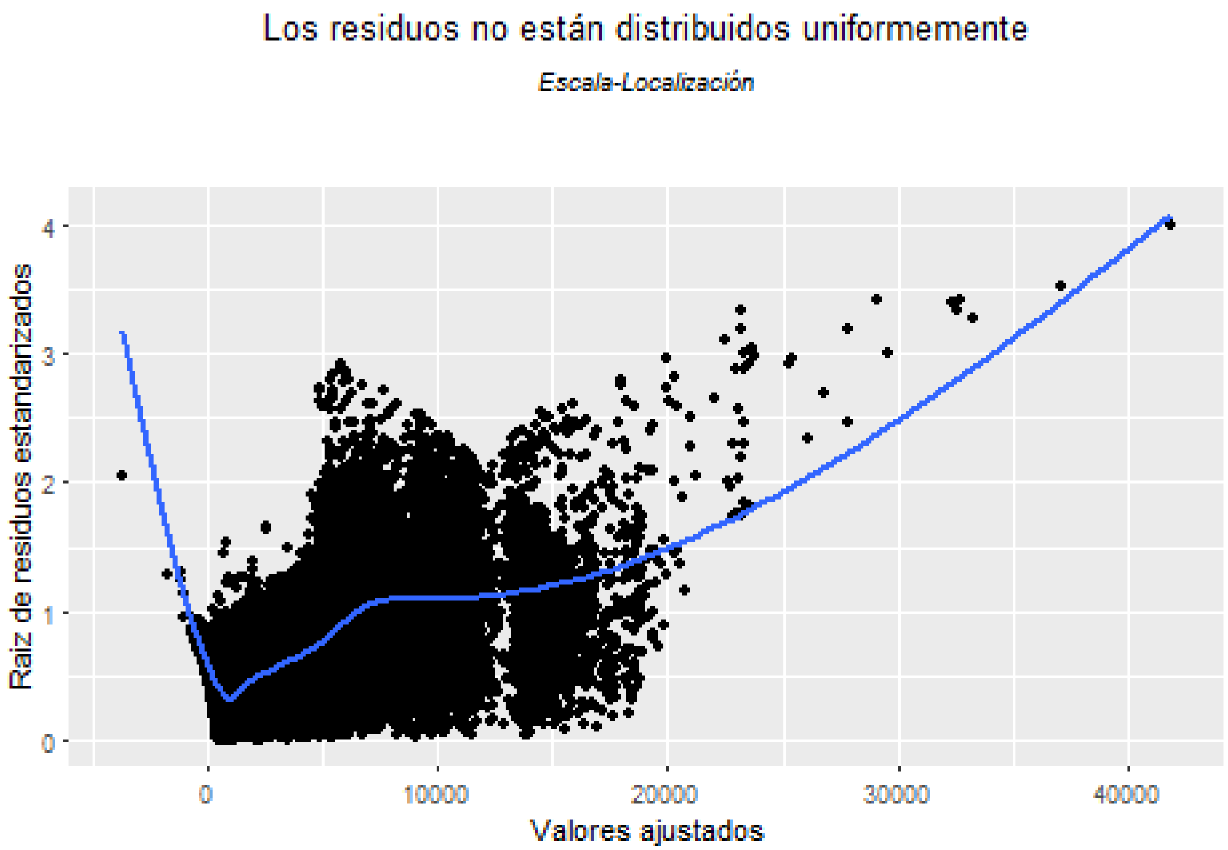



```
ggplot(modelonum) +
  stat_qq(aes(sample = .stdresid)) +
  geom_abline() +
  labs(x ="Cuantiles teóricos", y = "Residuos estandarizados") +
  ggtitle(expression(atop("El modelo no parece funcionar bien en valores pequeños o altos", atop(italic
  theme(plot.title = element_text(hjust = 0.5))
```



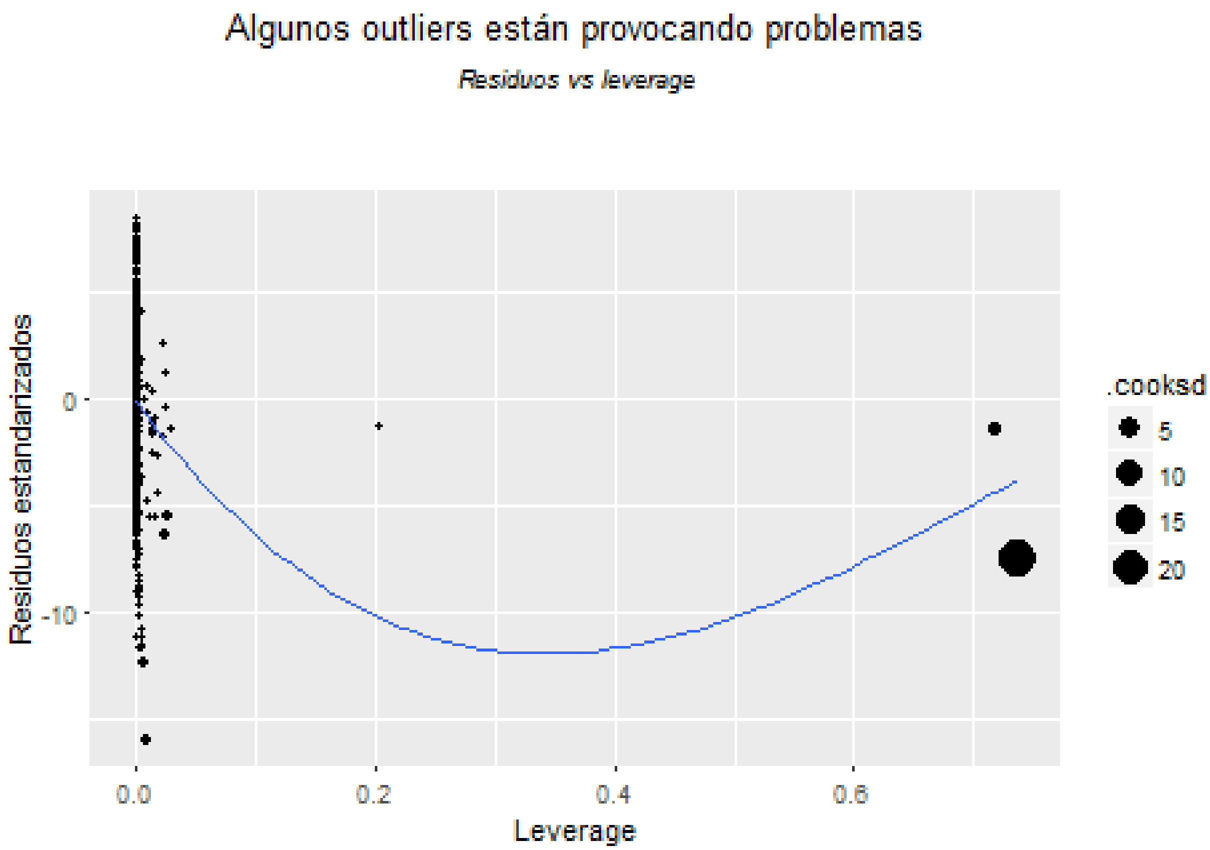

```
ggplot(modelonum, aes(.fitted, sqrt(abs(.stdresid)))) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(x ="Valores ajustados", y = "Raiz de residuos estandarizados") +
  ggtitle(expression(atop("Los residuos no están distribuidos uniformemente", atop(italic("Escala-Locali
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using method = 'gam'
```




```
ggplot(modelonum, aes(.hat, .stdresid)) +
  geom_point(aes(size = .cooksd)) +
  geom_smooth(se = FALSE, size = 0.5) +
  labs(x ="Leverage", y = "Residuos estandarizados") +
  ggtitle(expression(atop("Algunos outliers están provocando problemas", atop(italic("Residuos vs levera
  theme(plot.title = element_text(hjust = 0.5))
```

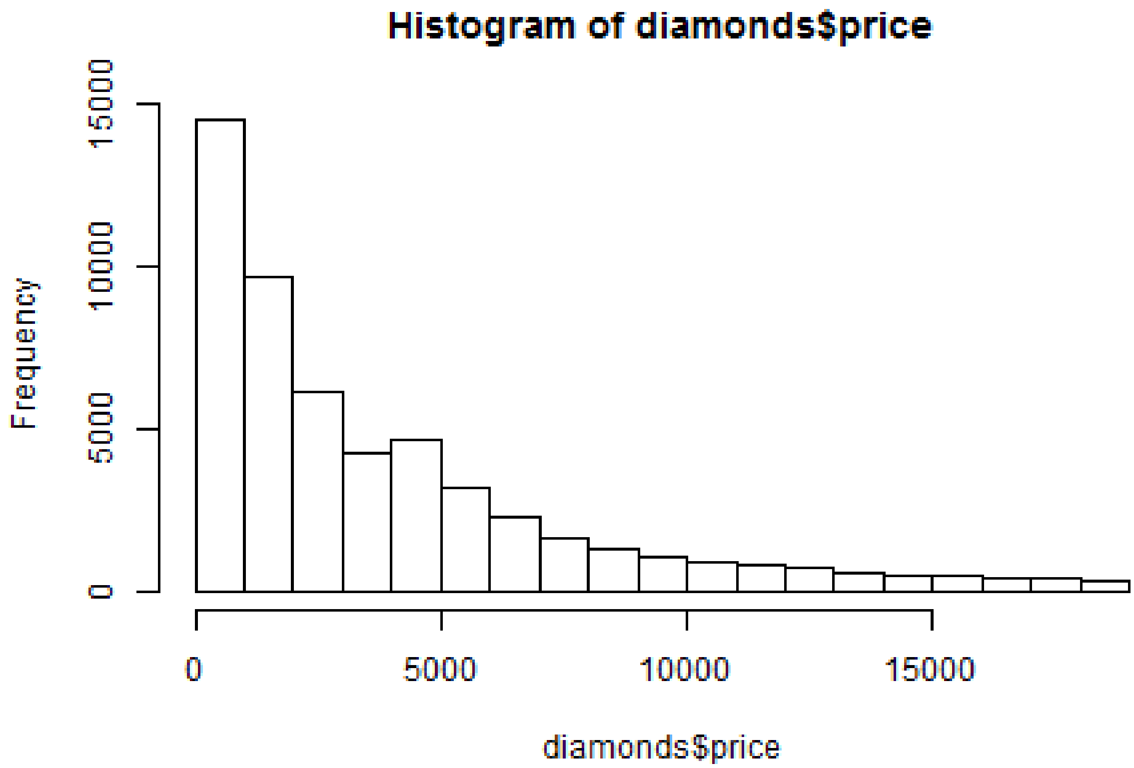
```
## `geom_smooth()` using method = 'gam'
```



Se ven algo extrañas las gráficas. Si bien la R^2 está indicando una bondad de ajuste adecuada, el modelo lineal parece estar teniendo problemas para ajustarse a los datos.

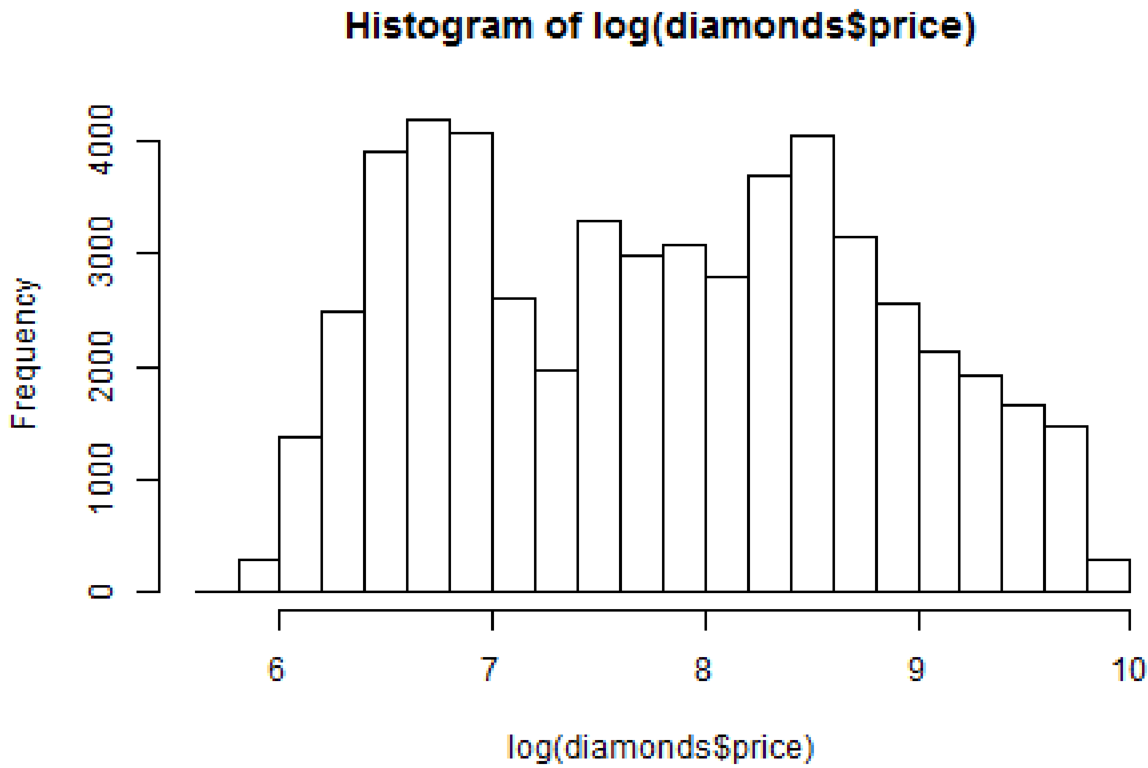
Veamos la distribución de la variable dependiente:

```r
hist(diamonds$price)
```



Está muy cargada hacia valores bajos.

```
hist(log(diamonds$price))
```



Mejor.

Hagamos algunos cambios: + La variable dependiente parece no estar bien balanceada, usemos mejor el logaritmo de esa variable + También a las variables independientes se les aplica una transformación logarítmica + Se ven dos grupos, vamos a agregar una variable que las separe + La variable `carat` parece más una variable categórica que una numérica, con unos saltos. Modelemos esos saltos.

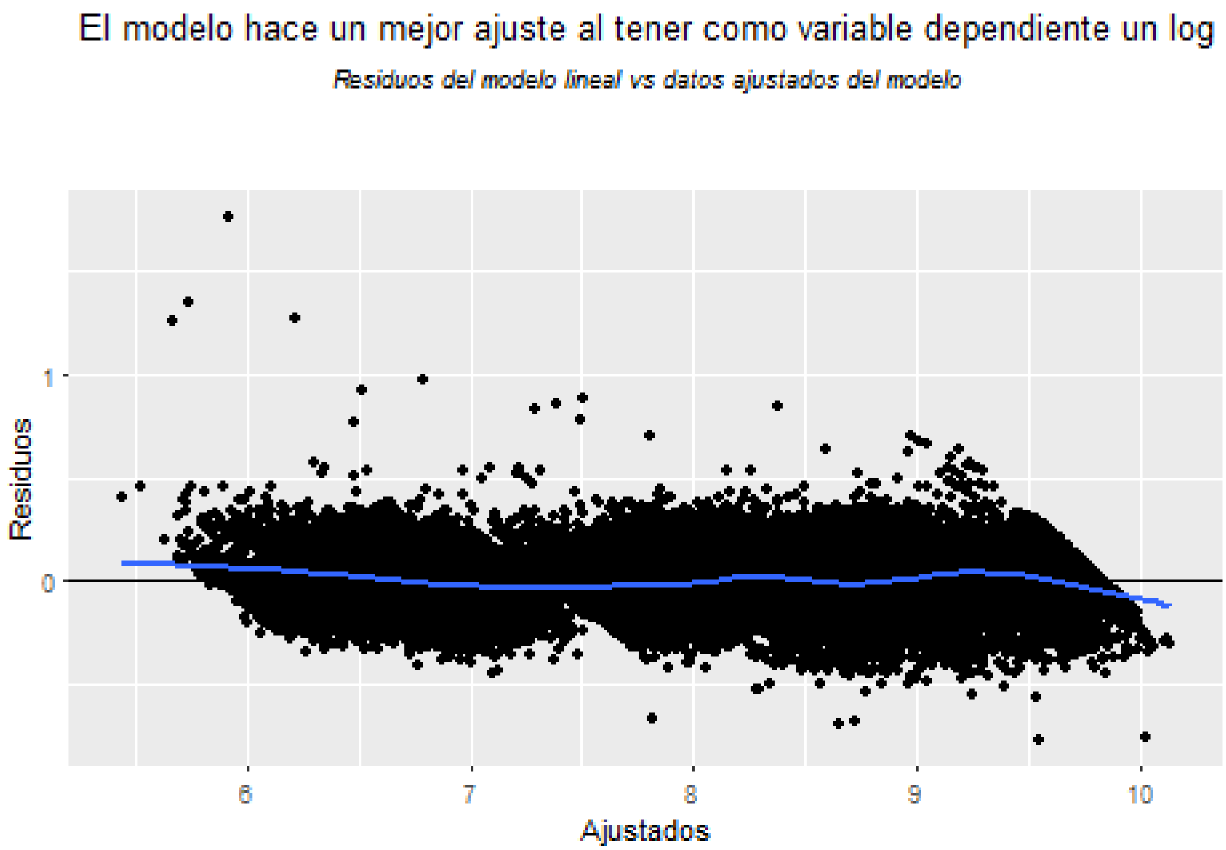
```
diamonds2= diamonds
y <- log(diamonds2$price)
bimodal <- ifelse(y > 7.4, c("1"), c("2"))
e <-c(0, .5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5)
bimodal2 <- cut(diamonds2$carat, e)
modelonum2 <- lm( y ~ bimodal + bimodal2 + log(diamonds2$carat,) + log(diamonds2$depth,) + log(diamonds2
summary(modelonum2)
```

```
##
## Call:
## lm(formula = y ~ bimodal + bimodal2 + log(diamonds2$carat, ) +
##     log(diamonds2$depth, ) + log(diamonds2$table, ) + log1p(diamonds2$x) +
##     log1p(diamonds2$y) + log1p(diamonds2$z) + diamonds2$cut +
##     diamonds2$color + diamonds2$clarity)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76888 -0.08369 -0.00262  0.08031  1.75707
##
## Coefficients:
##                          Estimate Std. Error  t value Pr(>|t|)
```

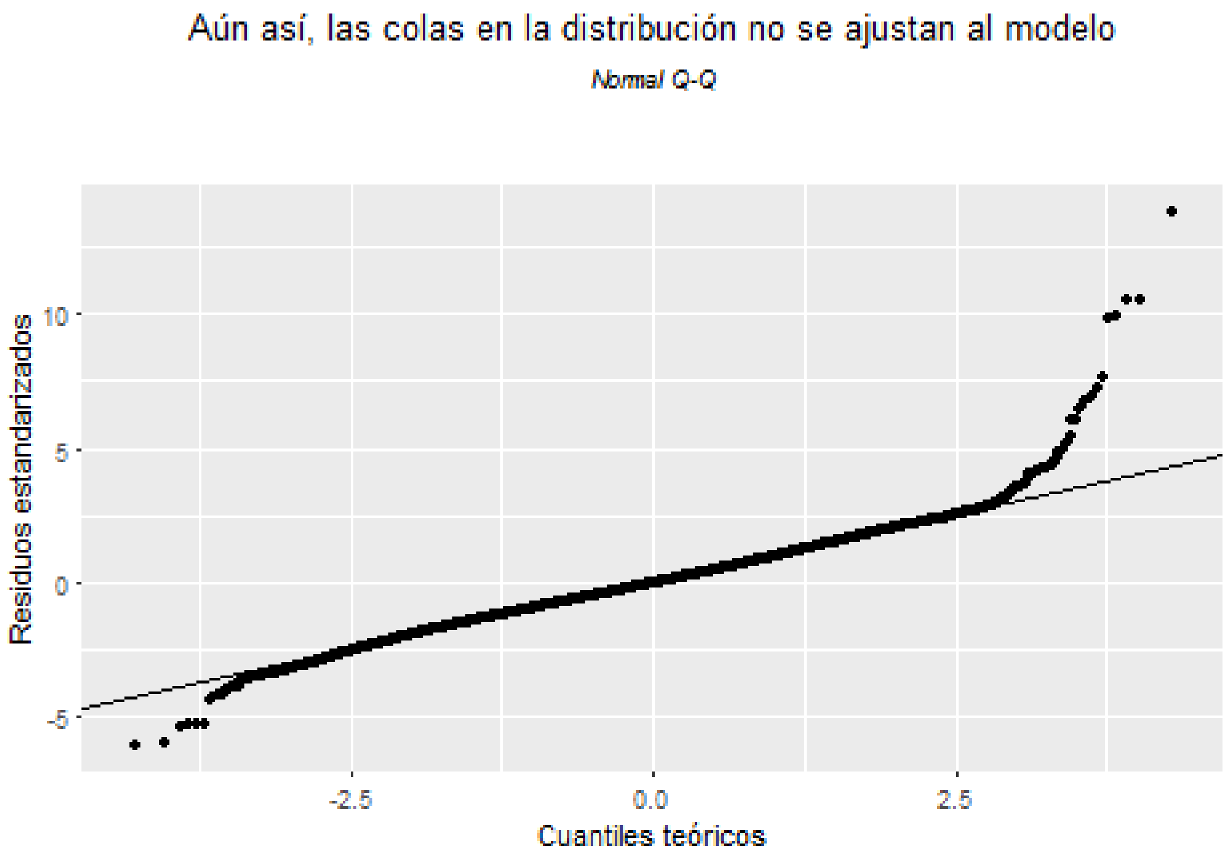
```
## (Intercept)             8.570180   0.193357   44.323  < 2e-16 ***
## bimodal2               -0.134072   0.002773  -48.349  < 2e-16 ***
## bimodal2(0.5,1]        -0.009916   0.002914   -3.403 0.000667 ***
## bimodal2(1,1.5]         0.102180   0.004187   24.403  < 2e-16 ***
## bimodal2(1.5,2]         0.128633   0.005469   23.522  < 2e-16 ***
## bimodal2(2,2.5]         0.149918   0.006586   22.764  < 2e-16 ***
## bimodal2(2.5,3]        -0.039832   0.014719   -2.706 0.006810 **
## bimodal2(3,3.5]        -0.178812   0.027634   -6.471 9.83e-11 ***
## bimodal2(3.5,4]        -0.344984   0.064436   -5.354 8.64e-08 ***
## bimodal2(4,4.5]        -0.236972   0.064546   -3.671 0.000241 ***
## log(diamonds2$carat, )  1.690926   0.007155  236.333  < 2e-16 ***
## log(diamonds2$depth, ) -0.055459   0.032337   -1.715 0.086347 .
## log(diamonds2$table, ) -0.013368   0.019245   -0.695 0.487288
## log1p(diamonds2$x)      0.311766   0.039018    7.990 1.37e-15 ***
## log1p(diamonds2$y)     -0.293715   0.036053   -8.147 3.82e-16 ***
## log1p(diamonds2$z)      0.055773   0.019622    2.842 0.004479 **
## diamonds2$cut.L         0.107649   0.002524   42.655  < 2e-16 ***
## diamonds2$cut.Q        -0.031027   0.002038  -15.221  < 2e-16 ***
## diamonds2$cut.C         0.014140   0.001763    8.018 1.09e-15 ***
## diamonds2$cut^4        -0.001475   0.001405   -1.050 0.293657
## diamonds2$color.L      -0.426192   0.001974 -215.929  < 2e-16 ***
## diamonds2$color.Q      -0.092101   0.001785  -51.599  < 2e-16 ***
## diamonds2$color.C      -0.011622   0.001666   -6.977 3.04e-12 ***
## diamonds2$color^4       0.010363   0.001530    6.772 1.28e-11 ***
## diamonds2$color^5      -0.003492   0.001445   -2.417 0.015669 *
## diamonds2$color^6       0.001527   0.001313    1.163 0.244972
## diamonds2$clarity.L     0.879144   0.003487  252.125  < 2e-16 ***
## diamonds2$clarity.Q    -0.228893   0.003209  -71.320  < 2e-16 ***
## diamonds2$clarity.C     0.130859   0.002742   47.732  < 2e-16 ***
## diamonds2$clarity^4    -0.062055   0.002188  -28.359  < 2e-16 ***
## diamonds2$clarity^5     0.022427   0.001785   12.566  < 2e-16 ***
## diamonds2$clarity^6    -0.001740   0.001551   -1.122 0.261940
## diamonds2$clarity^7     0.029765   0.001369   21.735  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1278 on 53906 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9842, Adjusted R-squared:  0.9841
## F-statistic: 1.046e+05 on 32 and 53906 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
ggplot(modelonum2, aes(.fitted, .resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_smooth(se = FALSE) +
  labs(x ="Ajustados", y = "Residuos") +
  ggtitle(expression(atop("El modelo hace un mejor ajuste al tener como variable dependiente un log", a
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using method = 'gam'
```

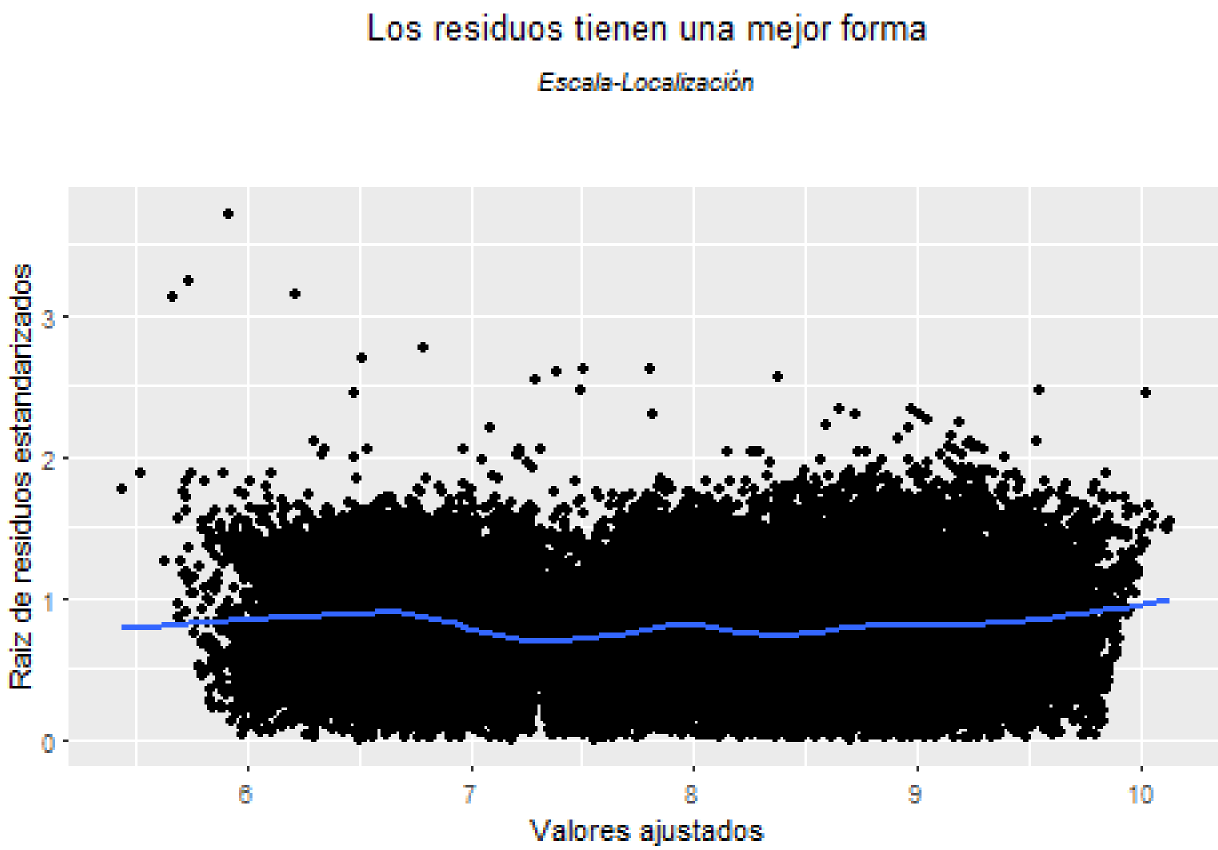



```
ggplot(modelonum2) +
  stat_qq(aes(sample = .stdresid)) +
  geom_abline() +
  labs(x ="Cuantiles teóricos", y = "Residuos estandarizados") +
  ggtitle(expression(atop("Aún así, las colas en la distribución no se ajustan al modelo", atop(italic(
  theme(plot.title = element_text(hjust = 0.5))
```



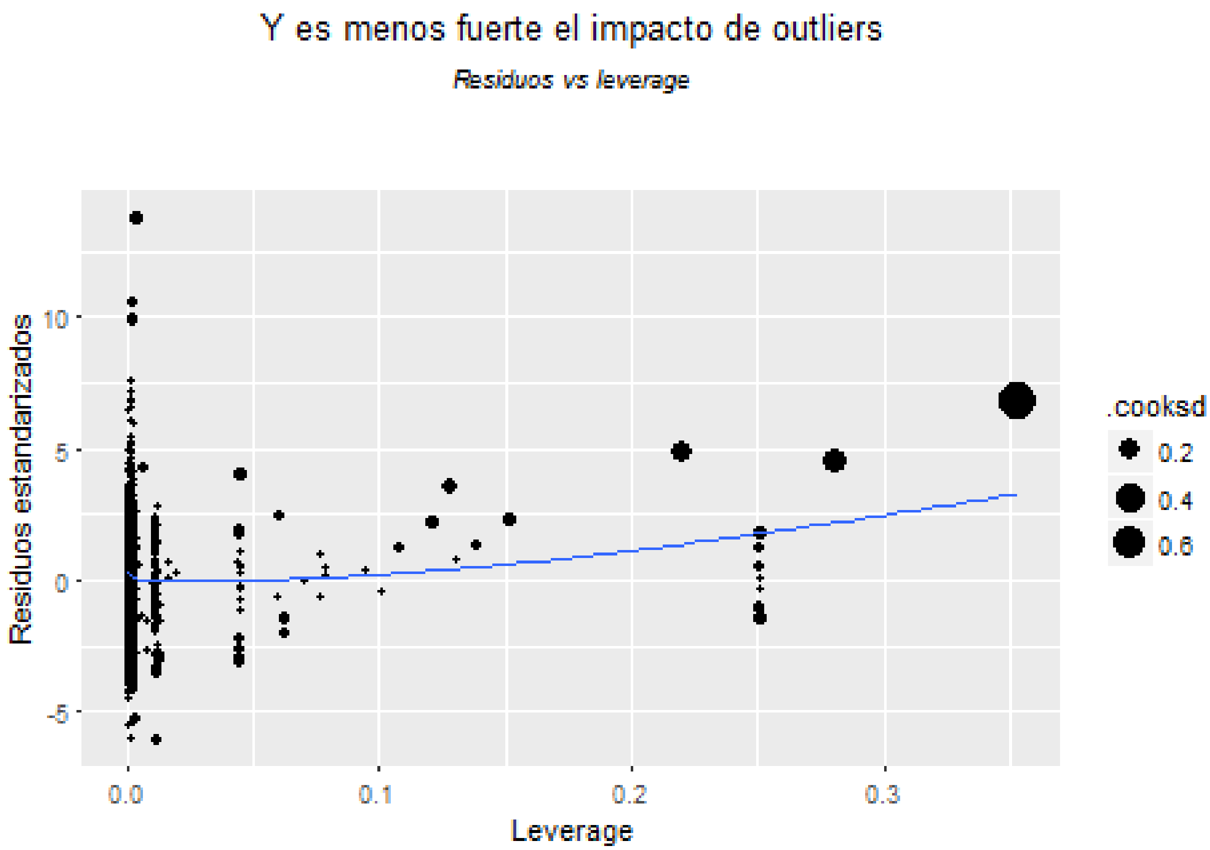

```
ggplot(modelonum2, aes(.fitted, sqrt(abs(.stdresid)))) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(x ="Valores ajustados", y = "Raiz de residuos estandarizados") +
  ggtitle(expression(atop("Los residuos tienen una mejor forma", atop(italic("Escala-Localización"), ""
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using method = 'gam'
```




```
ggplot(modelonum2, aes(.hat, .stdresid)) +
  geom_point(aes(size = .cooksd)) +
  geom_smooth(se = FALSE, size = 0.5) +
  labs(x ="Leverage", y = "Residuos estandarizados") +
  ggtitle(expression(atop("Y es menos fuerte el impacto de outliers", atop(italic("Residuos vs leverage
  theme(plot.title = element_text(hjust = 0.5))
```

```
## `geom_smooth()` using method = 'gam'
```



2.1 ¿Qué tan bueno fue el ajuste?

La muestra es bastante grande (~53 mil observaciones), y el modelo lineal inicial tiene una R^2 (de .8592). No obstante, las visualizaciones relacionadas con el modelo lineal mostraron que tenía problemas que se resolvieron al usar logaritmos para reescalar las variables en el modelo. El segundo modelo tiene una R^2 de .9841.

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

La R^2 en general representa una aproximación sencilla para entender la calidad del ajuste del modelo. También llamada coeficiente de determinación, es:

$$R^2 = \frac{\textit{Suma de los cuadrados de la regresión}}{\textit{Suma de los cuadrados de la regresión } + \textit{Suma de los cuadrados del error}}$$

Es decir, qué tan bien el modelo está explicando la variabilidad total de la información.

Su relación con la σ^2 es que la suma de los cuadrados del error son el estimador de la varianza del error real.

El valor de sigma del modelo es

```r
sigma(modelonum2)^2
```

```
## [1] 0.0163238
```

2.3 ¿Cuál es el ángulo entre Y y $\hat{Y}$?

```
a = modelonum$model$`diamonds$price`
b = predict(modelonum, type = "response")
theta1 <- acos( sum(a*b) / ( sqrt(sum(a * a)) * sqrt(sum(b * b)) ) )
paste("Ángulo modelo 1: ", theta1)
```

```
## [1] "Ángulo modelo 1:  0.270487433209261"
```

```
a = exp(modelonum2$model$y)
b = exp(predict(modelonum2, type = "response") )
theta2 <- acos( sum(a*b) / ( sqrt(sum(a * a)) * sqrt(sum(b * b)) ) )
paste("Ángulo modelo 2: ", theta2)
```

```
## [1] "Ángulo modelo 2:  0.131274999264681"
```

2.4 Defininan una funcion que calcule la logverosimilitud de unos parámetros β y σ^2.

Programemos aparte una columna de unos

```
ones <- array(1,c(length(diamonds$price)))
```

Y hagamos una función de log verosimilitud basada en la normal.

```
normal.lik1<-function(beta){
mu<-    diamonds$price -
        ones          * beta[1]+
        diamonds$carat* beta[2]+
        diamonds$depth* beta[3]+
        diamonds$table* beta[4]+
        diamonds$x    * beta[5]+
        diamonds$y    * beta[6]+
        diamonds$z    * beta[7]

sigma2<-beta[8]
n<-nrow(diamonds$price)
logl<- -.5*n*log(2*pi) -.5*n*log(sigma2) - (1/(2*sigma2))*sum(mu)**2
return(-logl) #minimiza
}
```

2.5 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`.

```
#optim(beta <- c(0,0,0,0,0,0,0,1), normal.lik1(beta))
```

No he logrado hacer que los estimadores sean iguales que los de Mínimos Cuadrados :/...