Parte Teórica

- Plantear el Problema de regresión como un problema de mínimos cuadrados.

En forma matricial se puede expresar el problema como

$$\mathbf{y} = \mathbf{X}\hat{\beta} + \hat{\mathbf{u}}$$

donde el vector u representa los errores (residuos), la idea es minimizar los errores al cuadrado, éstos son u transpuesta por u cuya expresión es

$$\begin{aligned}\hat{\mathbf{u}}'\hat{\mathbf{u}} =& (\mathbf{y} - \mathbf{X}\hat{\beta})'(\mathbf{y} - \mathbf{X}\hat{\beta}) = (\mathbf{y}' - \hat{\beta}'\mathbf{X}')(\mathbf{y} - \mathbf{X}\hat{\beta}) \\ =& \mathbf{y}'\mathbf{y} - \hat{\beta}'\mathbf{X}'\mathbf{y} - \mathbf{y}'\mathbf{X}\hat{\beta} + \hat{\beta}'\mathbf{X}'\mathbf{X}\hat{\beta} \\ =& \mathbf{y}'\mathbf{y} - 2\hat{\beta}'\mathbf{X}'\mathbf{y} + \hat{\beta}'\mathbf{X}'\mathbf{X}\hat{\beta}\end{aligned}$$

Para hallar el mínimo de la expresión anterior respecto a beta, derivamos e igualamos a cero

$$\frac{\partial(\hat{\mathbf{u}}'\hat{\mathbf{u}})}{\partial\hat{\beta}} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\hat{\beta} = \mathbf{0}_k$$

Se llega al siguiente sistema de ecuaciones

$$\mathbf{X}'\mathbf{X}\hat{\beta} = \mathbf{X}'\mathbf{y}$$

Así la solución para beta es

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

- ¿Por qué el planteamiento nos da un ajuste lineal?

Porque se encuentra una solución para las betas que por planteamiento son lineales

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

- ¿Qué logramos al agregar una columna de unos a la matriz X?

Se logra incluir un coeficiente de beta constante (no depende de las variables de la matriz X), este coeficiente no necesariamente tiene una interpretación sino que sirve para ajustar la estimación

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

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

Para el modelo lineal, en forma vectorial con n observaciones y p coeficientes, en forma vectorial tenemos tenemos

$$\mathrm{Y} = \mathrm{X}\beta + \mathrm{e}$$

con

$$\mathrm{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \mathrm{X} = \begin{pmatrix} 1 & x_{11} & x_{21} & \cdots & x_{p\text{-}11} \\ 1 & x_{12} & x_{22} & \cdots & x_{p\text{-}12} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{p\text{-}1n} \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p\text{-}1} \end{pmatrix}, \ \mathrm{e} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix},$$

Donde X, Y son conocidas y el vector Beta desconocido, los errores se distribuyen normal, $e \sim N(0, \sigma^2 I)$

- ¿Cuál es la función de verosimilitud del problema anterior?

La función de densidad conjunta es

$$\mathrm{f}\,(\mathrm{e};\,0\,,\,\sigma^2\mathrm{I}) = \frac{1}{(2\pi\sigma^2)^{p/2}} \exp^{-\frac{1}{2}(\mathrm{e})'\left(\frac{1}{\sigma^2}\mathrm{I}\right)(\mathrm{e})}$$

$$= \frac{1}{(2\pi\sigma^2)^{p/2}} \exp^{-\frac{1}{2}(\mathrm{Y}-\mathrm{X}\beta)'\left(\frac{1}{\sigma^2}\mathrm{I}\right)(\mathrm{Y}-\mathrm{X}\beta)}$$

$$= \frac{1}{(2\pi\sigma^2)^{p/2}} \exp^{-\frac{1}{2\sigma^2}(\mathrm{Y}-\mathrm{X}\beta)'(\mathrm{Y}-\mathrm{X}\beta)}$$

La función de máxima verosimilitud es

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

$$Ln(L(\sigma^2,\beta;e)) = -\frac{p}{2}\left(\ln 2\pi + \ln \sigma^2\right) - \frac{1}{2\sigma^2}(Y-X\beta)^t(Y-X\beta)$$

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

Derivando respecto a beta y sigma cuadrada se tiene

$$\frac{\partial Ln(L(\cdot))}{\partial \beta} = -\frac{1}{2\sigma^2}\left(-2X^tY + 2X^tX\beta\right)$$

$$\begin{aligned}\frac{\partial Ln(L(\cdot))}{\partial \sigma^2} &= -\frac{p}{2\sigma^2} + \frac{1}{2\sigma^4}\left(Y^tY - 2Y^tX\beta + \beta^tX^tX\beta\right)\\ &= -\frac{p}{2\sigma^2} + \frac{1}{2\sigma^4}(Y-X\beta)^t(Y-X\beta)\end{aligned}$$

Igualando a cero y resolviendo tenemos

$$-\frac{1}{2\sigma^2}\left(-2X^tY + 2X^tX\beta\right) = 0$$

$$\Rightarrow -X^tY + X^tX\beta = 0$$

$$\boxed{\Rightarrow \hat{\beta} = \left(X^tX\right)^{-1}X^tY}$$

Misma solución que mínimos cuadrados

$$-\frac{p}{2\sigma^2} + \frac{1}{2\sigma^4}(Y-X\beta)^t(Y-X\beta) = 0$$

$$\Rightarrow \widehat{\sigma^2} = \frac{\left(Y-X\hat{\beta}\right)^t\left(Y-X\hat{\beta}\right)}{p}$$

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

El Teorema de Gauss-Markov establece que dada una función estimable, su **estimador lineal de mínima varianza** es el estimador de mínimos cuadrados.

Parte Aplicada

Utilizarlos datos diamonds del paquete ggplot2, explicarla variable Price usando las variables númericas

```
> library(ggplot2)
> data("diamonds")
>
> ## Estructura de los datos
> 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 ...
> head(diamonds)
  carat       cut color clarity depth table price    x    y    z
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
```

```
> ##############################################################################################################################################
> ## Utilizar unicamente variables númericas
>
> variables.numericas <- which(sapply(diamonds,is.numeric))   #Un indice (vector) para saber que columnas son numericas (int o decimal)
>
> datos <- diamonds[,variables.numericas]
> str(datos)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':	53940 obs. of  7 variables:
 $ carat: num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
 $ 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 ...
>
> p <- dim(datos)[2] # número de coeficientes
> n <- dim(datos)[1] # número de observaciones
```

Generar Modelo

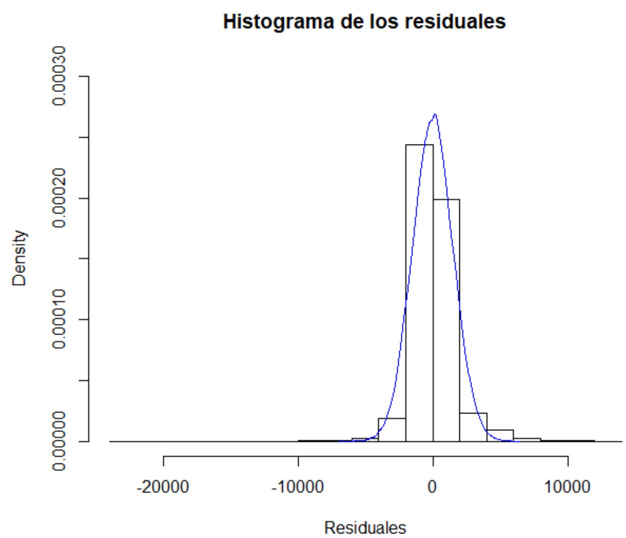
```
> ####################################################################
> ## Modelo de regresión
> diamonds.lm <- lm(price~.,data=datos)
>
> #diamonds.lm
> summary(diamonds.lm)

Call:
lm(formula = price ~ ., data = datos)

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
```

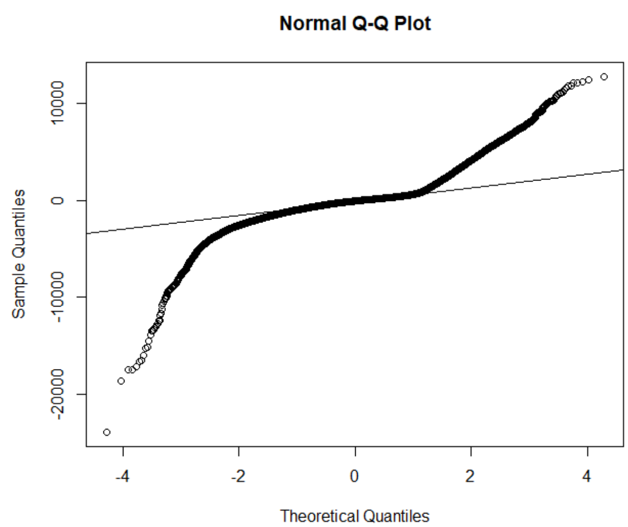
- ¿Qué tan bueno es el ajuste?

Los p-values de cada variable a excepción de la variable z, nos indican que cada variable es significativa, es decir, su coeficiente realmente debe ser distinto de cero, de la misma forma, el p-value del estadístico F nos indica que el modelo completo (la combinación lineal de las variables con los coeficientes obtenidos) es significativo

Validación de los supuestos del modelo

```
> ## 1) Normalidad de los errores
> hist(diamonds.lm$residuals,main="Histograma de los residuales",prob=TRUE,ylim = c(0,0.00030),xlab = "Residuales")
> lines(density(sort(rnorm(100000,0,1500))),col="blue")
```





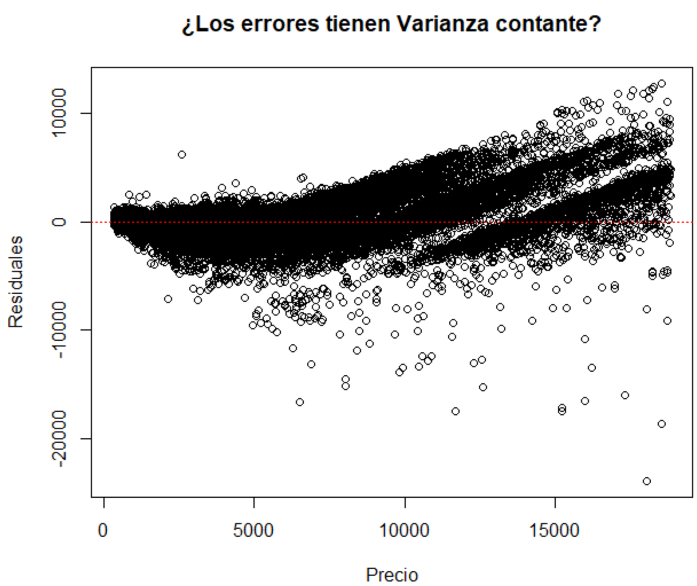
Visualmente el histograma de los errores parece normal, sin embargo el QQ plot muestra separación entre los cuartiles teóricos y los observados

Veamos ahora un prueba de bondad de ajuste

```
> shapiro.test(sample(diamonds.lm$residuals,5000))

        Shapiro-Wilk normality test

data:  sample(diamonds.lm$residuals, 5000)
W = 0.8267, p-value < 2.2e-16
```

Como el p-value de la prueba < 0.05 rechazamos la hipótesis de que los residuales se distribuyan normal.

```
> ## 2)  ¿Varianza constante?
> ## Los errores deben verse de forma aleatoria al rededor del cero en función de la variable objetivo (price)
> plot(diamonds.lm$residuals ~ datos$price, xlab="Precio",ylab = "Residuales",main="¿Los errores tienen Varianza contante?")
> abline(h = 0, lty = 3, col="red")
>
```



La varianza muestra cierta tendencia conforme el precio aumenta, lo que suguiere que no se cumpla con la condición de varianza constante

- ¿Qué medida ayuda a saber la calidad del modelo?

La R cuadrada ajustada, pues además de medir la variabilidad del modelo respecto a la variabilidad total, penaliza la inclusión de variables no significativas

- ¿Cuál es el ángulo entre los precios reales y los estimados?

```
> ## Ángulo entre los precios reales y las estimaciones
> model <- summary(diamonds.lm)
>
> angulo <- (180/pi)*(acos(sqrt(model$adj.r.squared)))
> angulo
[1] 22.03848
```

- Crear una función que calcule la Log verosimilitud

```
> ##############################################################################################################################
> ## Crear función de Log-verosiilitud
>
> #help(optim)
>
> ## Nota: Asignar de forma predeterminada las matrices X y Y, para que la función sólo tenga que recibir un parametro
> ## que sean los coeficientes a optimizar
> datos.Y <- datos$price
> datos.X <- datos
> datos.X$price <- NULL ## Drop la variable price
>
```

```
> ##############################################################################################################################
>
> log.mle <- function(parametros,X = datos.X,Y = datos.Y)
+
+ {
+
+ # Los parametros de la función son:
+   #- parmetros: un vector con los coeficientes de beta + la varianza en la última posición del vector
+   #- X: la matrix de variables independientes, no incluye la columna de unos (puede ser data frame o matriz)
+   #- Y: la variable independiente
+
+ # Garantizar el tipo de objeto correcto
+   parametros <- as.vector(parametros)
+   matriz.x <- cbind(1,X)  ## Agregar la columna de unos (1s)
+   matriz.x <- as.matrix(matriz.x)
+   Y <- as.vector(Y)
+
+  # En el objeto prametros, incluir los coeficientes de beta a estimar más la varianza al en la última entrada del vector
+   k <- length(parametros)
+   beta <- parametros[c(1:(k-1))]    # Extrae los elementos expecto el último
+   varianza <- parametros[k]         # Extare el último elemento
+
+ # P es en número de parametros a  estimar
+ p <- length(beta)
+
+ ## Notar que en R ya existe por default el valor para pi = 3.14
+
+
+ error <- sum((Y - matriz.x%*%beta)^2)
+ # Nota: La función "optim" minimiz por default, definir nuestra función con un menos
+ # Nota: se definio de forma positiva "mle" pues la versión original tiene signo negativo, esto para poder minimizar
+ mle <-  (p/2)*(log(2*pi)+log(varianza)) + (1/(2*varianza))*error
+
+ return(mle)
+ }
>
```

Validar el funcionamiento de la función

```
> ## Utilizar la función
>
> ## iniciar los parametros
> # Deben ser 7 betas + 1 varianza
> valores.iniciales <- c(100,100,100,100,100,100,100,100)
> length(valores.iniciales)
[1] 8
>
>
> ## Checar que la función definida previamente se ejecute e forma correcta
> log.mle(valores.iniciales)
[1] 28845659992
```

- Utilizar la función optim para hallar el máximo de la función de Logverosimilitud

```
> ##########################################################################
>
> ### Utilizar algunos valores iniciales cercanos a los optimos
>
> beta <- as.vector(diamonds.lm$coefficients)
> matriz.x <- as.matrix(cbind(1,datos.X))  ## Agregar la columna de 1s
>
> ## Sabemos que el estimador por MLE (sesgado) de la varianz es:
> varianza <-  sum((datos.Y - matriz.x%*%beta)^2)/(n - p)
>
```

Resultados de la función optim

```
> ##############
> casi.optimos <- c(diamonds.lm$coefficients,varianza) + 1000
> casi.optimos
 (Intercept)       carat       depth       table           x           y           z
  21849.3164  11686.3091    796.8459    897.5543   -315.6678   1066.3216   1041.6277 2241872.5361
>
> optim(par=casi.optimos, log.mle, method = "L-BFGS-B")
$par
  (Intercept)         carat         depth         table             x             y             z
  21850.34796   10764.25874    -213.17787    -107.06417   -1340.78307      58.60291      44.32766 2241932.96717

$value
[1] 27014.21

$counts
function gradient
     122      122

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

Comprar resultados con los obtenidos previamente

```
>
> ##############
> ## Comparar con los resultados de la función lm
> diamonds.lm$coefficients
(Intercept)       carat       depth       table           x           y           z
 20849.3164  10686.3091   -203.1541   -102.4457  -1315.6678     66.3216     41.6277
> varianza
[1] 2240873
>
>
> ## Notar que para la variables  depth, table, x, y , z se tomó un valor inicial
> ## alejado del optimo y la función los aproximó a los ya conocidos y correctos
>
```