Tipología y ciclo de vida de los datos: PRA2

Autora: Jesica Piñón Rodríguez

Enero 2020

PRÁCTICA 2

---

1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

El conjunto de datos escogido se ha descargado del enlace "https://www.kaggle.com/shivam2503/diamonds" y se trata de un juego de datos sobre los tipos de diamantes. Los atributos que forman parte de este conjunto de datos son los siguientes:

- Index –> Contador
- Carat –> Peso en quilates del diamante
- cut –> Cualidad del corte del diamante
- color –> Calidad del color del diamante
- clarity –> Calidad de las inclusiones hechas en el diamante.I1 (peor), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (mejor).
- depth –> Profundidad del diamante
- table –> El ancho de la tabla del diamante expresado como un porcentaje de su diámetro promedio
- price –> Precio en dólares
- x –> Longitud
- y –> Anchura
- z –> Altura

El objetivo es conocer **qué características de los diamantes son las que más influyen a la hora de fijar el precio de estos**. Es un estudio relevante para las joyerías y comercios que pretendan fijar los precios de una manera más objetiva.

Empezaremos, por tanto, leyendo los datos y visualizando sus características principales en cuanto a tipo de dato y formato. Se realizará, por tanto, un primer análisis estadístico descriptivo previo a realizar cualquier modificación del conjunto de datos.

```
##############################################
# Tipología y ciclo de vida de los datos    #
# PRA2                                      #
# Jesica Piñón                              #
##############################################

#=========================================
# Fijamos el directorio de trabajo
#=========================================
setwd("E:/UOC_MASTER/3_SEMESTRE/TIPOLOGIA_CICLO_VIDA_DATOS/PRACS/PRA2")

#=========================================
# Cargamos librerías necesarias -  Se instalan los paquetes automáticamen
te si la librería no es encontrada!
#=========================================
is.installed <- function(mypkg) is.element(mypkg, installed.packages()[,1
])
if (!is.installed("ggplot2")){install.packages("ggplot2",repos="http://cr
an.r-project.org")}

if (!is.installed("data.table")){install.packages("data.table",repos="htt
p://cran.r-project.org")}

if (!is.installed("rmarkdown")){install.packages("rmarkdown",repos="http:
//cran.r-project.org")}


if (!is.installed("corrplot")){install.packages("rmarkdown",repos="http:/
/cran.r-project.org")}

suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(data.table))
suppressPackageStartupMessages(library(rmarkdown))
suppressPackageStartupMessages(library(corrplot))

print ("Librerías cargadas\n")

## [1] "Librerías cargadas\n"

#=========================================
# Cargamos los datos
#=========================================
data.dt <- as.data.table(read.csv("diamonds.csv", na.strings=""))

#=========================================
# Analizamos la estructura de los datos
#=========================================
# Comprobamos atributos y clases
str(data.dt)
```

```
## Classes 'data.table' and 'data.frame':   53940 obs. of  11 variables:
##  $ X      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : Factor w/ 5 levels "Fair","Good",..: 3 4 2 4 2 5 5 5 1 5 .
..
##  $ color  : Factor w/ 7 levels "D","E","F","G",..: 2 2 2 6 7 7 6 5 2 5
...
##  $ clarity: Factor w/ 8 levels "I1","IF","SI1",..: 4 3 5 6 4 8 7 3 6 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 ...
##  - attr(*, ".internal.selfref")=<externalptr>

summary(data.dt)

##        X             carat               cut        color
##  Min.   :    1   Min.   :0.2000   Fair     : 1610   D: 6775
##  1st Qu.:13486   1st Qu.:0.4000   Good     : 4906   E: 9797
##  Median :26971   Median :0.7000   Ideal    :21551   F: 9542
##  Mean   :26971   Mean   :0.7979   Premium  :13791   G:11292
##  3rd Qu.:40455   3rd Qu.:1.0400   Very Good:12082   H: 8304
##  Max.   :53940   Max.   :5.0100                     I: 5422
##                                                     J: 2808
##     clarity          depth           table           price
##  SI1    :13065   Min.   :43.00   Min.   :43.00   Min.   :  326
##  VS2    :12258   1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  950
##  SI2    : 9194   Median :61.80   Median :57.00   Median : 2401
##  VS1    : 8171   Mean   :61.75   Mean   :57.46   Mean   : 3933
##  VVS2   : 5066   3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5324
##  VVS1   : 3655   Max.   :79.00   Max.   :95.00   Max.   :18823
##  (Other): 2531
##        x                y               z
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
##  Median : 5.700   Median : 5.710   Median : 3.530
##  Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
##  3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
##  Max.   :10.740   Max.   :58.900   Max.   :31.800
##
```

Se observa que el conjunto de datos consta de **11 atributos**; 3 categóricos y el resto numéricos y enteros. Tenemos un total de **53940 registros**.

2. Integración y selección de los datos de interés a analizar.

En este caso, no es necesario un proceso de integración ya que disponemos de una única fuente de datos. De todas formas, debemos revisar si existen valores duplicados. En cuanto a la selección de datos, eliminaremos la columna 'X' ya que no nos aporta información relevante para el análisis por tratarse de un mero contador. Asimismo, crearemos una **nueva variable** a partir de las dimensiones x,y,z que se corresponderá con el volumen del diamante. Por tanto, reduciremos los 3 atributos a uno sin pérdida de información relevante para el proyecto.

```
#=========================================
# Eliminamos variables
#=========================================
# Hay variables que no nos aportan información como la 'X' que es un cont
ador.
data.dt$X <- NULL

#=========================================
# Eliminamos las filas duplicadas
#=========================================
print(paste0("El número de filas duplicadas es: ",nrow(data.dt[duplicated
(data.dt),])))
```

```
## [1] "El número de filas duplicadas es: 146"
```

```
data.dt <- unique(data.dt)

# Analizamos las variables categóricas un poco más.
unique(data.dt$clarity)
```

```
## [1] SI2  SI1  VS1  VS2  VVS2 VVS1 I1   IF
## Levels: I1 IF SI1 SI2 VS1 VS2 VVS1 VVS2
```

```
unique(data.dt$cut)
```

```
## [1] Ideal     Premium   Good      Very Good Fair
## Levels: Fair Good Ideal Premium Very Good
```

```
unique(data.dt$color)
```

```
## [1] E I J H F G D
## Levels: D E F G H I J
```

```
#=========================================
# Creación de nuevos atributos
#=========================================
# Creación de la variable 'Vol' - volumen del diamante.
data.dt[,vol:= round(x*y*z,1),]

# Comprobamos límites:
min(data.dt$vol)
```

```
## [1] 0

# El volumen no puede ser 0; eliminamos esos casos:
data.dt <- data.dt[vol != 0]
min(data.dt$vol)

## [1] 31.7

max(data.dt$vol)

## [1] 3840.6

summary(data.dt$vol)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    31.7    65.2   114.9   129.9   170.8  3840.6

#=========================================
# Eliminamos los atributos innecesarios
#=========================================
data.dt$x <- data.dt$y <- data.dt$z <- NULL
```

Aunque en el apartado siguiente se habla de limpieza de datos, entendemos que estos puntos anteriores también corresponden a dicho proceso ETL necesario para la fiabilidad del análisis. Por ello, a continuación se muestran algunas tareas más que ayudan a la selección de variables mediante la reducción de la dimensionalidad, esto es, del número de atributos del dataset mediante un análisis de componentes principales, ACP. Dicho análisis debe realizarse sólo con los atributos numéricos por lo que las variables 'clarity', 'color' y 'cut' quedan fuera de este análisis. Del mismo modo, el atributo del precio queremos conservarlo como objetivo del análisis por lo que no se incluirá en el método de reducción aplicado.

```
#=========================================
# ACP
#=========================================
acp <- prcomp(data.dt[,c(1,5:6,8)],center=T,scale=T)
print(acp)

## Standard deviations (1, .., p=4):
## [1] 1.4278277 1.1253072 0.8207337 0.1462460
##
## Rotation (n x k) = (4 x 4):
##               PC1        PC2         PC3         PC4
## carat  0.68580996 -0.1408632  0.08843121  0.70852111
## depth -0.04552512 -0.7391263 -0.67175688 -0.01903945
## table  0.24280958  0.6443566 -0.72496752 -0.01643610
## vol    0.68456969 -0.1365812  0.12387379 -0.70524123

summary(acp)

## Importance of components:
##                           PC1    PC2    PC3     PC4
```

```
## Standard deviation     1.4278 1.1253 0.8207 0.14625
## Proportion of Variance 0.5097 0.3166 0.1684 0.00535
## Cumulative Proportion  0.5097 0.8263 0.9947 1.00000

data.dt$vol <- NULL
```

Se observa, por tanto, que de las 4 variables, podemos quedarnos con 3, las cuales explican más del **95% de la varianza del conjunto de datos**. Aunque la última era la variable previamente creada, parece que no es relevante para el estudio.

3. Limpieza de los datos.

3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Comprobamos la existencia de ceros o vacíos así como elementos que puedan designar datos perdidos como NA.

```
#=========================================
# Datos perdidos
#=========================================
summary(data.dt)

##      carat               cut        color        clarity
##  Min.   :0.2000   Fair      : 1597   D: 6754   SI1    :13030
##  1st Qu.:0.4000   Good      : 4888   E: 9776   VS2    :12225
##  Median :0.7000   Ideal     :21485   F: 9517   SI2    : 9142
##  Mean   :0.7975   Premium   :13737   G:11254   VS1    : 8155
##  3rd Qu.:1.0400   Very Good:12068   H: 8266   VVS2   : 5056
##  Max.   :5.0100                     I: 5406   VVS1   : 3646
##                                     J: 2802   (Other): 2521
##      depth           table           price
##  Min.   :43.00   Min.   :43.00   Min.   :  326
##  1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  951
##  Median :61.80   Median :57.00   Median : 2401
##  Mean   :61.75   Mean   :57.46   Mean   : 3931
##  3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5324
##  Max.   :79.00   Max.   :95.00   Max.   :18823
##

colSums(is.na(data.dt)) # No hay valores nulos.

##   carat     cut   color clarity   depth   table   price
##       0       0       0       0       0       0       0
```

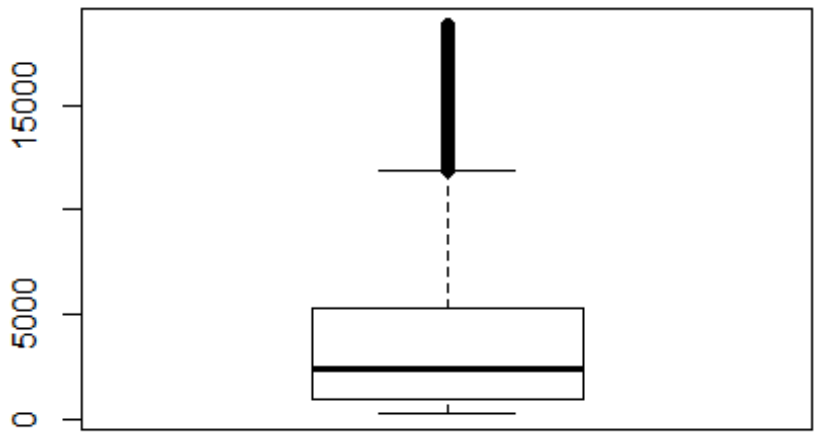
Se observa que no existen ceros (los valores mínimos son superiores a cero) ni NA's. Además, al cargar los datos lo hemos hecho usando 'na.strings=""' de modo que los registros vacíos se leerían como NA, por lo que tampoco hay registros vacíos.

En todo caso, si hubiese valores nulos o vacíos que correspondiesen a errores en la medida o perdida de información (es decir, asegurándonos que dichos valores no son vacíos legítimos), habría varias opciones para resolver este problema. Por un lado, si el número de estos valores perdidos es insignificante se pueden directamente eliminar dichos valores e ignorarlos para el análisis posterior. Se pueden también sustituir estos valores por la media o mediana, por ejemplo.

3.2. Identificación y tratamiento de valores extremos.

Al mostrar el resumen estadístico de los datos, llama la atención el salto grande que existe entre el tercer quantil y el máximo para atributos como "carat", esto es, el peso en quilates del diamante. También el precio muestra valores muy alejados de la distribución normal de dicha variable. Basándonos en el método del diagrama de caja o boxplot que indica que los valores que superen 3 veces el valor de la desviación estándar con respecto a la media se consideran outliers. Para ello, representaremos ambas variables, precio y peso, con un diagrama de caja para visualizar estos valores y la existencia de valores extremos.

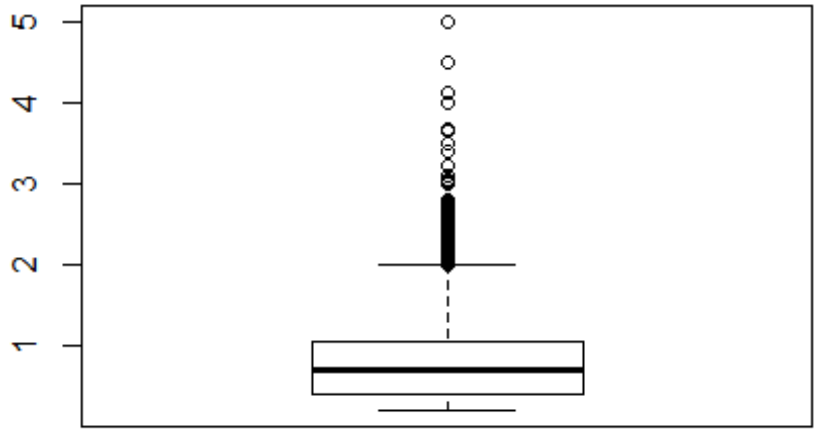
```
#=========================================
# Outliers
#=========================================
#PRICE
boxplot(data.dt$price)
```




```
print (paste0("Representan un porcentaje de ",round((nrow(data.dt[price >
3*sd(data.dt$price)])/nrow(data.dt))*100,2),"% del total de los datos"))
```

```
## [1] "Representan un porcentaje de 6.46% del total de los datos"
```

```
##CARAT
boxplot(data.dt$carat)
```




```
print (paste0("Representan un porcentaje de ",round((nrow(data.dt[carat >
3*sd(data.dt$carat)])/nrow(data.dt))*100,2),"% del total de los datos"))
```

```
## [1] "Representan un porcentaje de 11.77% del total de los datos"
```

Se observa que hay muchos valores en el rango considerado dentro de los valores extremos, sin embargo, son datos lógicos que no corresponden por tanto a datos erróneos y que son necesarios para el análisis ya que ayudan a entender cómo se fija un precio alto de un diamante y ver también si el peso en quilates es directamente proporcional al precio o influyen otras características como el corte o color.

4. Análisis de los datos.

4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

Tras la limpieza de datos, nos quedamos con 6 atributos, 3 numéricos y 3 categóricos que definen las características de un diamante, además del atributo numérico que contiene la información sobre el precio. Consideramos que estos datos ya son los adecuados para realizar el análisis.

4.2. Comprobación de la normalidad y homogeneidad de la varianza.

Con el fin de comprobar la **normalidad** del conjunto de datos, realizaremos el **test de Kolmogorov-Smirnov** ya que el de Shapiro-Wilk sólo permite muestras de máximo 5000 registros (tenemos 53633). El test de normalidad parte de la hipótesis nula de que la población está distribuida normalmente y, por tanto, si p-value es menor que el nivel de significancia que se quiera considerar, en este caso, 0.05, entonces la hipótesis nula es rechazada y los datos no siguen una distribución normal. Por el contrario, si p-value es mayor que 0.05, los datos sí son normales.

Este análisis se realiza sólo sobre las variables numéricas.

```
#========================================
# Test de normalidad de Kolmogorov-Simnorv
#========================================
cols <- names(data.dt)[c(1,5:7)]
file <- data.dt[,cols,with=F]
for (i in c(1:length(cols))){
  print(apply(file[,i,with=FALSE],2,function(x) ks.test(x,pnorm,mean(x),s
d(x),exact=F)))
}
```

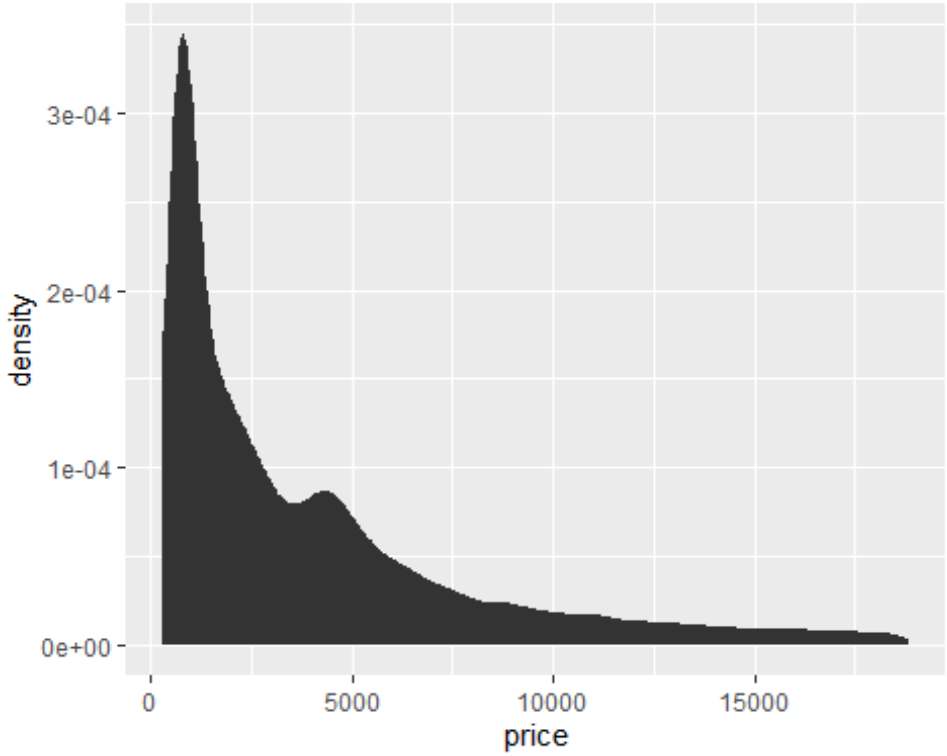
```
## $carat
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.1226, p-value < 2.2e-16
## alternative hypothesis: two-sided
##
##
## $depth
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.075556, p-value < 2.2e-16
## alternative hypothesis: two-sided
##
##
## $table
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.13219, p-value < 2.2e-16
## alternative hypothesis: two-sided
##
```

```
##
## $price
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.18457, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

p-value es inferior a 0.05 por lo que **la distribución no es normal**.

Representamos a continuación, a modo de ejemplo, la función de distribución para el atributo precio.

```
ggplot(data.dt,aes(price))+stat_density()
```



En cuanto a la comprobación de la **homogeneidad**, dado que los datos no siguen una distribución normal, usaremos el **test de Fligner-Killeen**. En este test la hipótesis nula asume igualdad de varianzas en los diferentes grupos de datos, por lo que p-value inferiores al nivel de significancia (0.05) indicarán heterogeneidad.

Comprobaremos la homogeneidad del precio con el resto de atributos numéricos.

```
#=========================================
# Test de homogeneidad de Fligner-Killen
#=========================================
fligner.test(price ~ carat, data=data.dt)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  price by carat
## Fligner-Killeen:med chi-squared = 27079, df = 272, p-value <
## 2.2e-16
```

```
fligner.test(price ~ table, data=data.dt)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  price by table
## Fligner-Killeen:med chi-squared = 2484, df = 126, p-value <
## 2.2e-16
```

```
fligner.test(price ~ depth, data=data.dt)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  price by depth
## Fligner-Killeen:med chi-squared = 1066.2, df = 183, p-value <
## 2.2e-16
```

Dado que los p-value son menores que 0.05, se rechaza la hipótesis nula y se concluye que las variables carat, table y depth presentan **varianzas estadísticamente diferentes** para los grupos de price.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

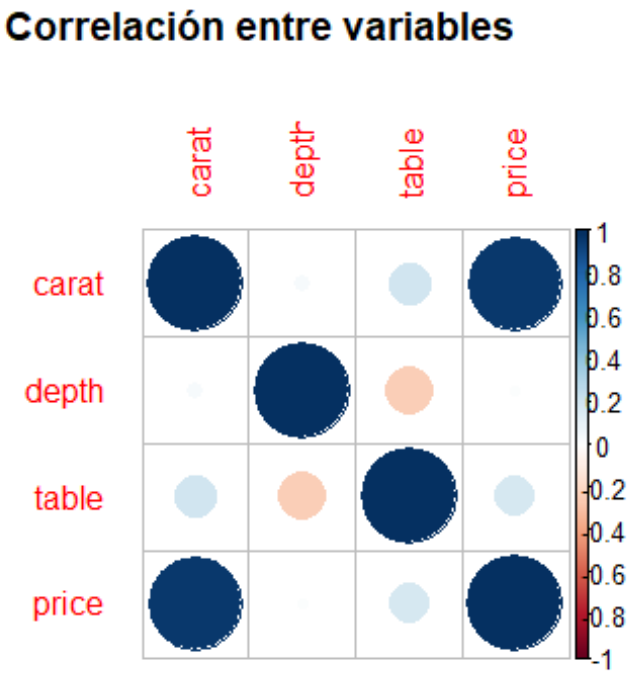
Correlación entre las variables

El primer análisis que realizaremos será el análisis de correlación entre las variables numéricas para determinar cuáles de ellas ejercen una mayor influencia sobre el precio. Como tenemos datos que no siguen una distribución normal ni cumplen el criterio de homogeneidad, no podremos usar el coeficiente de correlación de Pearson y debemos utilizar **el coeficiente de correlación de Spearman** que no supone ningún tipo de distribución de los datos.

```
#=========================================
# Correlación
#=========================================
data.dt <- as.data.table(data.dt)
table_cor <- cor(data.dt[,cols,with=F],method="spearman")
table_cor
```

```
##            carat       depth       table     price
## carat  1.0000000  0.0303339  0.1941256 0.9629298
## depth  0.0303339  1.0000000 -0.2451838 0.0101722
## table  0.1941256 -0.2451838  1.0000000 0.1710321
## price  0.9629298  0.0101722  0.1710321 1.0000000
```

```
corrplot(table_cor,title="Correlación entre variables",mar=c(0,0,5,0),tl.
offset = 1)
```



Tanto en la tabla como en el gráfico parece verse claramente que price y carat, están fuertemente correlacionados, esto es, el peso en quilates influye en el precio del diamante. Sin embargo, para poder hacer la afirmación anterior se precisa comprobar que, efectivamente, está correlación es significativa. Para ello, aplicaremos la función cor.test y comprobaremos el valor de p-value.

```
#=========================================
# Correlación
#=========================================
cor.test(data.dt$price,data.dt$carat,method="spearman",exact=F)
```

```
##
##  Spearman's rank correlation rho
##
## data:  data.dt$price and data.dt$carat
## S = 9.6076e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
```

```
##       rho
## 0.9629298
```

Como p-value es menor que 0.05, consideramos que es un **resultado significativo**.

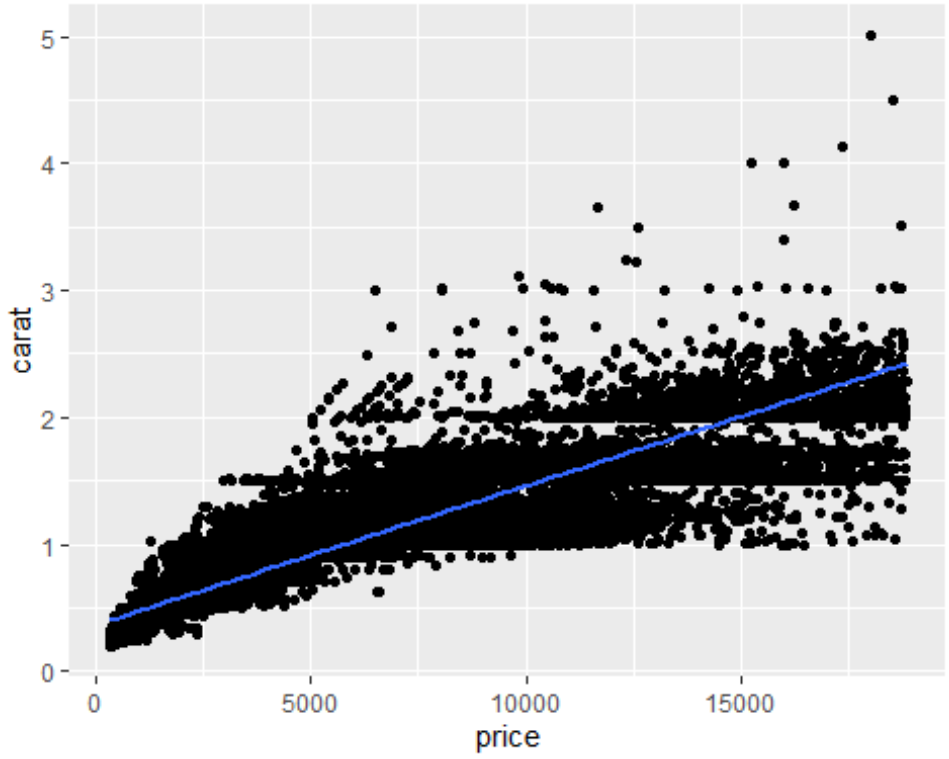
Regresión

Otro análisis para ver la dependencia lineal entre las variables es la regresión lineal. Previamente ya hemos visto que price y carat están fuertemente correlacionadas. En este análisis realizaremos modelos de regresión lineal considerando la variable carat pero también analizaremos la regresión del resto de variables categóricas (cut,color y clarity), las cuales todavía no se han analizado.

```
#=========================================
# Regresión lineal
#=========================================
a <- lm(price ~ carat, data=data.dt)
print(paste0("Regresión lineal entre price y carat. r2: ",summary(a)$r.sq
uared))

## [1] "Regresión lineal entre price y carat. r2: 0.849251590553195"

b <- lm(price ~ cut, data=data.dt)
print(paste0("Regresión lineal entre price y cut. r2: ",summary(b)$r.squa
red))

## [1] "Regresión lineal entre price y cut. r2: 0.0126281873491449"

c <- lm(price ~ clarity, data=data.dt)
print(paste0("Regresión lineal entre price y clarity. r2: ",summary(c)$r.
squared))

## [1] "Regresión lineal entre price y clarity. r2: 0.0269126344933559"

d <- lm(price ~ color, data=data.dt)
print(paste0("Regresión lineal entre price y color. r2: ",summary(d)$r.sq
uared))

## [1] "Regresión lineal entre price y color. r2: 0.030990646792154"
```

Tal y como se observa, sólo el peso en quilates, carat, consigue un coeficiente de correlación, r2, de 0.85 mientras que los demás apenas alcanzan el 0.1

```
#=========================================
# Regresión lineal
#=========================================
ggplot(data.dt,aes(x=price,y=carat))+geom_point()+geom_smooth(method = "l
m")
```



Este análisis nos permitiría utilizar la ecuación de la regresión lineal entre precio y peso del diamante para predecir los precios de los diamantes en función de lo que pesen.

```
#=========================================
# Predicción del precio del diamante en función de su peso en quilates.
#=========================================
# Creamos un dataset con valores de peso en quilates.
peso <- data.table(carat=c(0.2,0.3,4.7,3.8))
predict(a,newdata=peso)
```

```
##           1           2           3           4
##  -707.46281    68.83877 34226.10842 27239.39417
```

Evidentemente, no pueden existir precios negativos con lo que esta ecuación no sería la ideal para realizar este tipo de análisis.

Comparación entre grupos de datos

Al tratarse de datos que no siguen una ditribución normal ni son homogéneos, realizaremos el **test de Mann-Whitney** para comparar grupos de datos. Nos interesa ver si el precio varía, esto es, un diamante es más caro o más barato según determinadas características. Para ello, para ciertos atributos, realizaremos el test para determinar si existe dependencia o no entre los grupos de datos y si existe, por tanto, un precio dependiendo de un grupo o es algo totalmente independiente del valor que alcance esa característica.

Usaremos la opción “alternative=‘less’” para determinar qué grupo tiene más peso sobre el precio del diamante.

```
#=========================================
# Test de Mann-Whitney
#=========================================
# Carat
# Creamos dos grupos con los precios según tengan un peso u otro en quila
tes.
data.carat.low <- data.dt[carat < 3]$price
data.carat.high <- data.dt[carat >= 3]$price
wilcox.test(data.carat.low,data.carat.high,alternative="less")

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data.carat.low and data.carat.high
## W = 89725, p-value < 2.2e-16
## alternative hypothesis: true location shift is less than 0

# Cut
# Creamos dos grupos con los precios según la calidad del tipo de corte.
data.cut.low <- data.dt[cut != 'Premium']$price
data.cut.high <- data.dt[cut == 'Premium']$price
wilcox.test(data.cut.low,data.cut.high,alternative="less")

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data.cut.low and data.cut.high
## W = 240730727, p-value < 2.2e-16
## alternative hypothesis: true location shift is less than 0

# Clarity
# Creamos dos grupos con los precios según tengan una categoría de clarid
ad.
data.clarity.low <- data.dt[clarity %in% c("I1","SI1","SI2","VS2")]$price
data.clarity.high <- data.dt[clarity %in% c("VS1","V","SI2","VS2")]$price
wilcox.test(data.clarity.low,data.clarity.high,alternative="less")

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data.clarity.low and data.clarity.high
## W = 526212656, p-value = 0.9993
## alternative hypothesis: true location shift is less than 0

# Depth
# Creamos dos grupos con los precios según la profundidad del diamante.
data.depth.low <- data.dt[depth <= 65]$price
```

```
data.depth.high <- data.dt[depth > 65]$price
wilcox.test(data.depth.low,data.depth.high,alternative='less')

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data.depth.low and data.depth.high
## W = 17934282, p-value < 2.2e-16
## alternative hypothesis: true location shift is less than 0

# Table
# Creamos dos grupos con los precios según tengan un ancho de tabla de di
amante.
data.table.low <- data.dt[table <= 75]$price
data.table.high <- data.dt[table > 75]$price
wilcox.test(data.table.low,data.table.high,alternative='less')

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data.table.low and data.table.high
## W = 57485, p-value = 0.1944
## alternative hypothesis: true location shift is less than 0
```

Observamos que para el corte, peso en quilates y profundidad del diamante se verifica que p-value es menor que 0.05 por lo que se rechaza la hipótesis nula y se deduce que estas características tienen significativamente más peso en el precio si el corte es de más calidad y el peso en quilates y la profundidad del diamante es mayor. Por otro lado, tanto la calidad de las inclusiones hechas en el diamante como el ancho de la tabla del diamante no tienen significativamente más peso en el precio si son de más calidad o mayores, respectivamente.

5. Representación de los resultados a partir de tablas y gráficas.

A lo largo de los diferentes apartados se han ido mostrando tablas y gráficas de los resultados.

6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Considero que los resultados muestran de manera clara la **dependencia significativa del precio con el peso en quilates del diamante**. Se han realizado diferentes análisis y en todos ellos se ha podido verificar este comportamiento. Por otro lado, se ha comprobado que el conjunto de datos era un conjunto que **no** seguía una **distribución normal** ni era **homogéneo**. Se ha intentado, sin éxito, usar la regresión

lineal para predecir el precio de un diamante en base a su peso. Esta predicción carece de fiabilidad y se precisaría de otros métodos de machine learning, más precisos, para mejorar la predicción.

Asimismo, en el proceso previo al análisis, se ha verificado la no existencia de valores nulos o vacíos y se han comprobado los valores extremos.

7. Código: Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

El código, elaborado en R, se ha ido enseñando a lo largo de la práctica pero también se puede encontrar en la carpeta del proyecto con el nombre de "main.R".