Universidad del Valle de Guatemala
Ejercicio en clase - Grupo 5

Estudiante: Esteban Armas, Carla Ayala, Jackelin Billingslea, Rudik Rompich
Correos: arm19371@uvg.edu.gt,aya19048@uvg.edu.gt,bil19161@uvg.edu.gt, rom19857@uvg.edu.gt
Carnés: 19371, 19048,19161, 19857

IA3028 - Data Mining - Catedrático: Luis Pedro Flores
6 de septiembre de 2021

Tarea precios - Joyas Exclusivas

Instrucciones: La tienda Joyas Exclusivas S.A. ,ubicada en Guatemala, compra diamantes en el extranjero para abastecer sus distintas sucursales. Normalmente lo hacen en unas ferias en donde distintos proveedores ofrecen sus tiendas. Han tenido el inconveniente que no saben con exactitud el mejor precio al cual deberían de comprar los diamantes. Para ello, lo han contratado a usted para poder desarrollar un modelo que pueda predecir los precios de los diamantes y usarlo para conocer mejor el mercado. La libreria de datos que deben cargar es GGPLOT2, y el set de datos está dentro de esa librería y se llama "diamonds".

```
library(ggplot2)
library(kableExtra)
library(corrplot)
```

```
## corrplot 0.90 loaded
```

1. Haga un análisis inicial de los datos usando las funciones apropiadas para el mismo. Haga un screenshot del output en su informe.

```
resumen <- summary(diamonds)
cabecera <- head(diamonds)
```

```
kbl(cabecera, booktabs = T) %>% kable_styling(latex_options =
                                              c("striped", "hold_position"))
```

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48

```
kbl(resumen, booktabs = T) %>% kable_styling(latex_options =
                                             c("striped","scale_down", "hold_position"))
```

```
str(diamonds)
```

```
## tibble [53,940 x 10] (S3: tbl_df/tbl/data.frame)
```

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

```
##  $ carat  : num [1:53940] 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 [1:53940] 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num [1:53940] 55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int [1:53940] 326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num [1:53940] 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num [1:53940] 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num [1:53940] 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

```
any(is.na(diamonds))
```

```
## [1] FALSE
```

2. Haga un diagrama de caja utilizando la siguiente fórmula e interprete los resultados:

2.1. Precio vrs. corte

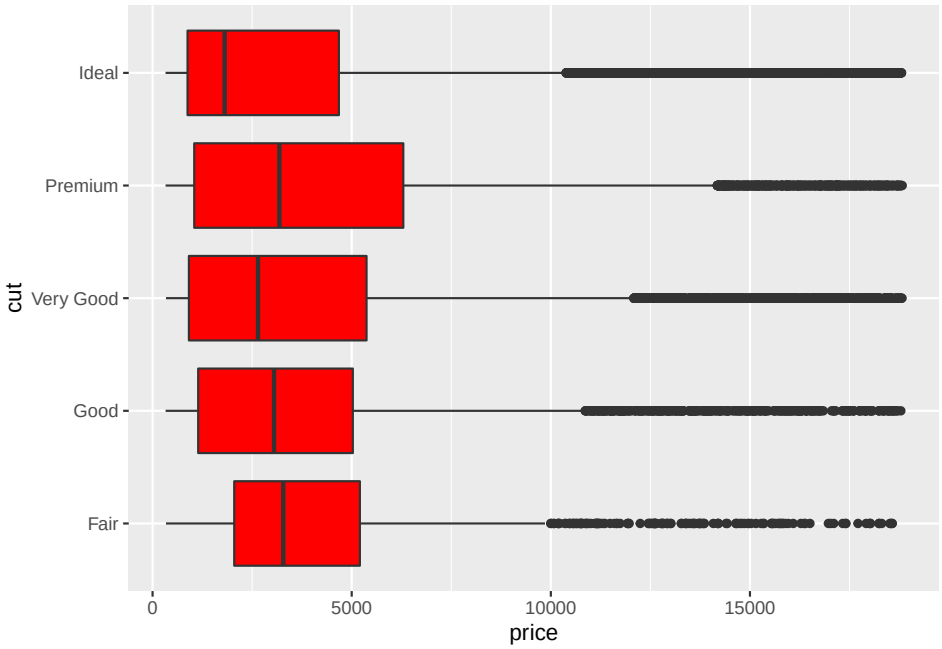

Figura 1: Precio vrs. corte.

2.2. Precio vrs. peso

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

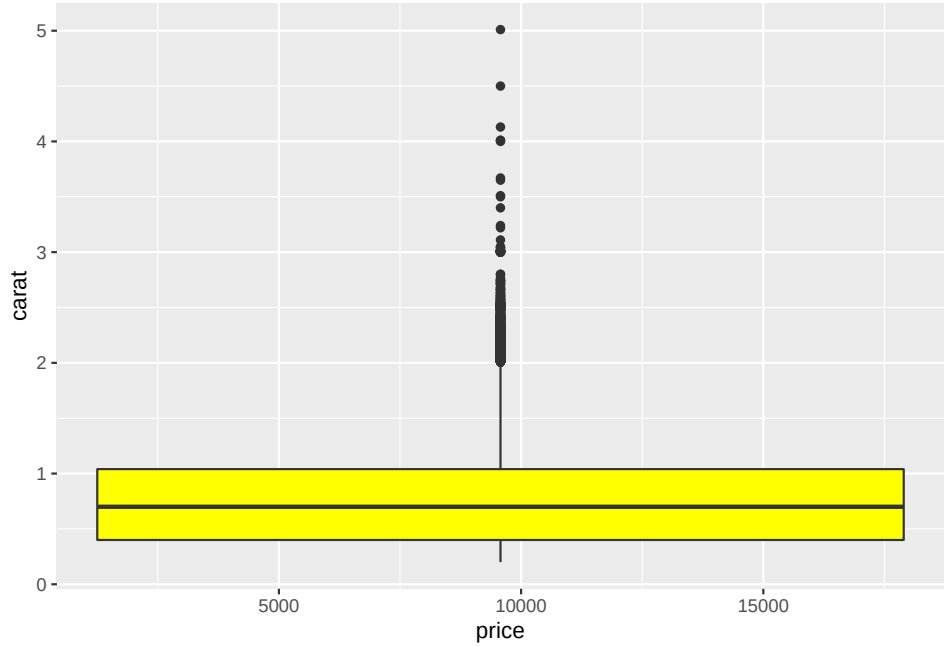

Figura 2: Precio vrs. peso.

2.3. Precio vrs. claridad

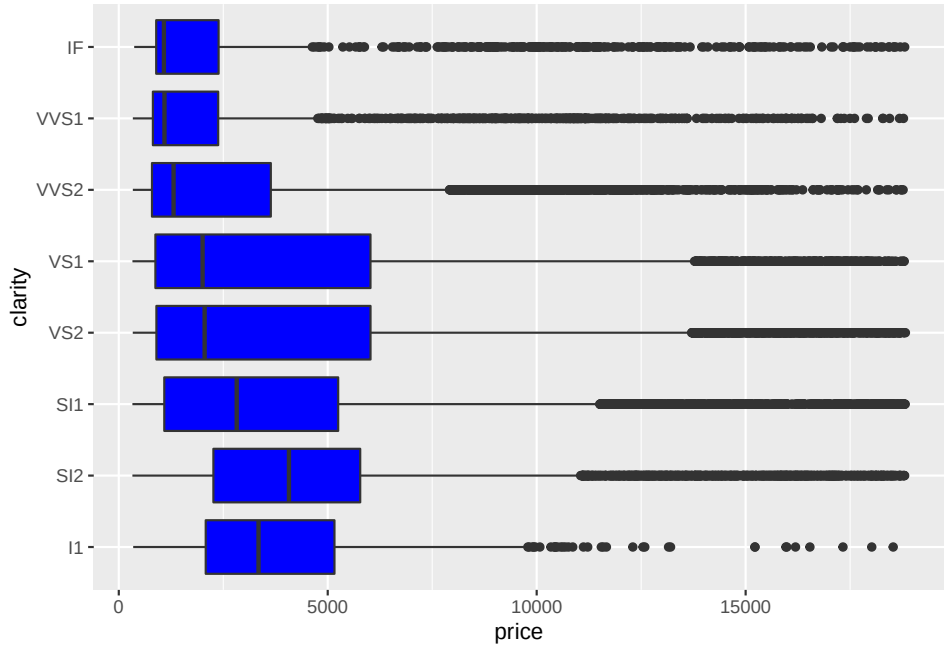

Figura 3: Precio vrs. claridad.

La interpretación de las cajas de bigote parecieran indicar que existen una cantidad excesiva de puntos atípicos, por lo tanto, en una regresión lineal mucho más detallada; sería necesario eliminarlos para obtener mejores resultados.

3. Haga una matriz de correlación y analice los resultados. ¿Hay alguna variable que esté altamente relacionada?

```
#Evaluar que columnas son numéricas
vect_num <- sapply(diamonds, is.numeric)
#Vector de valores lógicos para filtrar numéricos
cor_diamonds <- cor(diamonds[,vect_num])
#Graficar
corrplot(cor_diamonds,method = "number", sig.level=0.05)
```

Figura 4: Matriz de correlación

```
help(corrplot)
```

Las relaciones significativas son las siguientes:

1. carat con price,x,y,z.
2. price con carat,x,y,z.

4. Haga la partición de datos en entrenamiento y prueba.

```
library(caTools)
```

```r
# Fijamos la aleatoriedad
set.seed(69)
#Partición de los datos
muestreo <- sample.split(diamonds$price, SplitRatio = 0.7)
# Los subconjuntos de datos entrenamiento y prueba
entrenamiento <- subset(diamonds, muestreo == T)
prueba <- subset(diamonds, muestreo == F)
```

5. Entrene el modelo de regresión lineal usando todas las variables como predictoras.

```r
modelo <- lm(price ~ .,entrenamiento)
summary(modelo)
```

```
##
## Call:
## lm(formula = price ~ ., data = entrenamiento)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21396.7   -621.3   -192.9    396.2  10537.9
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5481.993    543.114  10.094  < 2e-16 ***
## carat       11280.437     58.662 192.294  < 2e-16 ***
## cut.L         592.607     27.741  21.362  < 2e-16 ***
## cut.Q        -305.994     22.246 -13.755  < 2e-16 ***
## cut.C         150.618     18.962   7.943 2.03e-15 ***
## cut^4         -20.724     15.193  -1.364  0.17255
## color.L     -2043.221     21.202 -96.370  < 2e-16 ***
## color.Q      -692.693     19.276 -35.935  < 2e-16 ***
## color.C      -177.222     18.012  -9.839  < 2e-16 ***
## color^4        35.826     16.582   2.161  0.03074 *
## color^5      -107.147     15.669  -6.838 8.15e-12 ***
## color^6       -40.671     14.263  -2.851  0.00435 **
## clarity.L    4234.218     37.148 113.982  < 2e-16 ***
## clarity.Q   -1965.189     34.687 -56.655  < 2e-16 ***
## clarity.C    1010.730     29.720  34.009  < 2e-16 ***
## clarity^4    -373.315     23.733 -15.730  < 2e-16 ***
## clarity^5     257.064     19.394  13.255  < 2e-16 ***
## clarity^6       4.514     16.878   0.267  0.78914
## clarity^7     103.048     14.869   6.930 4.27e-12 ***
## depth         -60.021      6.770  -8.866  < 2e-16 ***
## table         -28.555      3.584  -7.967 1.66e-15 ***
## x            -907.544     51.372 -17.666  < 2e-16 ***
## y               8.096     20.744   0.390  0.69633
## z            -168.213     75.499  -2.228  0.02588 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1172 on 38453 degrees of freedom
```

```
## Multiple R-squared:  0.9196, Adjusted R-squared:  0.9195
## F-statistic: 1.912e+04 on 23 and 38453 DF,  p-value: < 2.2e-16
```

```
predicciones <- predict(modelo, prueba)
resultados <- cbind(prueba, predicciones)
kbl(head(resultados), booktabs = T) %>% kable_styling(latex_options =
                                                      c("striped","scale_down", "hold_position"))
```

carat	cut	color	clarity	depth	table	price	x	y	z	predicciones
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31	-715.8744
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48	-1455.3399
0.26	Very Good	H	SI1	61.9	55	337	4.07	4.11	2.53	-1143.5258
0.32	Premium	E	I1	60.9	58	345	4.38	4.42	2.68	-3713.8419
0.30	Very Good	J	SI1	62.7	59	351	4.21	4.27	2.66	-2452.5050
0.23	Very Good	H	VS1	61.0	57	353	3.94	3.96	2.41	-393.1858

6. Haga una función que diga si el resultado es negativo, debe colocar el valor mínimo de precio dentro de la base de datos.

```
convertir_cero <- function(x){
  if(x<0){
    return (min(diamonds$price))
  } else{
    return (x)
  }
}
```

7. Corra la función con las predicciones.

```
resultados$predicciones <- sapply(resultados$predicciones, convertir_cero)

kbl(head(resultados), booktabs = T) %>% kable_styling(latex_options =
                                                      c("striped","scale_down", "hold_position"))
```

carat	cut	color	clarity	depth	table	price	x	y	z	predicciones
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31	326
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48	326
0.26	Very Good	H	SI1	61.9	55	337	4.07	4.11	2.53	326
0.32	Premium	E	I1	60.9	58	345	4.38	4.42	2.68	326
0.30	Very Good	J	SI1	62.7	59	351	4.21	4.27	2.66	326
0.23	Very Good	H	VS1	61.0	57	353	3.94	3.96	2.41	326

8. ¿Cuál es el MAPE del modelo? ¿Cuál es el RMSE del modelo? ¿Está haciendo un buen trabajo el modelo?

```
MAPE <- 100/nrow(resultados)* sum((resultados$price-resultados$predicciones)/resultados$price)
MAPE
```

```
## [1] -6.234558
```

```
MSE <- mean((resultados$price - resultados$predicciones)^2)
RMSE <- sqrt(MSE)
RMSE
```

```
## [1] 949.8472
```

Los resultados encontrados en el MAPE y indican un porcentaje negativo de aproximadamente -6 % comparado entre los precios reales y las prediciones; por otra parte el RMSE indica una pérdida muy pequeña. Por lo tanto, el modelo está haciendo un buen trabajo.

9. Utilice el modelo para hacer las proyecciones utilizando el archivo de "diamantes - uso.csv".

```
getwd()
```

```
## [1] "/Users/rudiks/Git/UVG-DataMining-Notas-6-Semestre/Laboratorio4"
```

```
uso <- read.csv("diamantes - uso.csv", sep = ";")
predicciones_uso <- predict(modelo, uso)
resultados_uso <- cbind(uso, predicciones_uso)
resultados_uso$predicciones_uso <- sapply(resultados_uso$predicciones_uso, convertir_cero)
kbl(head(resultados_uso, 30), booktabs = T) %>% kable_styling(latex_options =
                                                c("striped","scale_down", "hold_position"))
```

carat	cut	color	clarity	depth	table	x	y	z	predicciones_uso
0.32	Ideal	I	VVS1	62.0	55.3	4.39	4.42	2.73	200.84299
0.31	Very Good	G	SI1	63.3	57.0	4.33	4.30	2.73	326.00000
0.31	Premium	G	SI1	61.8	58.0	4.35	4.32	2.68	326.00000
0.24	Premium	E	VVS1	60.7	58.0	4.01	4.03	2.44	923.56890
0.24	Very Good	D	VVS1	61.5	60.0	3.97	4.00	2.45	1048.58370
0.30	Very Good	H	SI1	63.1	56.0	4.29	4.27	2.70	326.00000
0.30	Premium	H	SI1	62.9	59.0	4.28	4.24	2.68	326.00000
0.30	Premium	H	SI1	62.5	57.0	4.29	4.25	2.67	326.00000
0.30	Good	H	SI1	63.7	57.0	4.28	4.26	2.72	326.00000
0.26	Very Good	F	VVS2	59.2	60.0	4.19	4.22	2.49	869.01722
0.26	Very Good	E	VVS2	59.9	58.0	4.15	4.23	2.51	976.31288
0.26	Very Good	D	VVS2	62.4	54.0	4.08	4.13	2.56	1225.68413
0.26	Very Good	D	VVS2	62.8	60.0	4.01	4.05	2.53	1098.26967
0.26	Very Good	E	VVS1	62.6	59.0	4.06	4.09	2.55	908.05723
0.26	Very Good	E	VVS1	63.4	59.0	4.00	4.04	2.55	914.08788
0.26	Very Good	D	VVS1	62.1	60.0	4.03	4.12	2.53	1171.24140
0.26	Ideal	E	VVS2	62.9	58.0	4.02	4.06	2.54	1015.49743
0.38	Ideal	I	SI2	61.6	56.0	4.65	4.67	2.87	326.00000
0.26	Good	E	VVS1	57.9	60.0	4.22	4.25	2.45	885.36310
0.24	Premium	G	VVS1	62.3	59.0	3.95	3.92	2.45	556.28487
0.24	Premium	H	VVS1	61.2	58.0	4.01	3.96	2.44	89.33879
0.24	Premium	H	VVS1	60.8	59.0	4.02	4.00	2.44	76.04035
0.24	Premium	H	VVS2	60.7	58.0	4.07	4.04	2.46	13.63949
0.32	Premium	I	SI1	62.9	58.0	4.35	4.33	2.73	326.00000
0.70	Ideal	E	SI1	62.5	57.0	5.70	5.72	3.57	2996.88262
0.86	Fair	E	SI2	55.1	69.0	6.45	6.33	3.52	2422.64877
0.70	Ideal	G	VS2	61.6	56.0	5.70	5.67	3.50	3429.13881
0.71	Very Good	E	VS2	62.4	57.0	5.68	5.73	3.56	3660.79754
0.78	Very Good	G	SI2	63.8	56.0	5.81	5.85	3.72	2355.13596
0.70	Good	E	VS2	57.5	58.0	5.85	5.90	3.38	3541.76590