Trabajo N°1 Punto 1: Ajuste de un modelo de R.L.M

Universidad Nacional de Colombia
Analisís de Regresión 2022-1S
Medellín, Colombia
2022

Daniel Villa 1005087556 Juan Pablo Vanegas 1000640165

Contents

1 Introducción

Este conjunto de datos contiene información sobre coches usados. La descripción de cada característica se explica a continuación:

- `nombre`: Nombre de los coches
- `ano`: Año del coche cuando se compró
- `precio`: Precio al que se vende el coche
- `kilometros`: Número de kilómetros recorridos por el coche
- `combustible`: Tipo de combustible del coche (gasolina / diesel / GNC / GLP)
- `tipo_vendedor`: Indica si el coche es vendido por un particular o por un concesionario
- `transmisión`: Transmisión del coche (automática/manual)
- `Propietario`: Número de propietarios anteriores
- `kilometraje`: Kilometraje del coche (kmpl)
- `Motor`: Capacidad del motor del coche (CC)
- `max_power`: Potencia máxima del motor (CV)
- `asientos`: número de asientos del coche

Este conjunto de datos se utilizará para predecir el precio de venta de los coches usados, por lo que estableceremos el `precio` como variable objetivo.

2 Preparación de datos

2.1 Cargar conjunto de datos

Nombre	Año	Precio	Kilometros	Combustible	Tipo_vendedor	Transmicio
Maruti Swift Dzire VDI	2014	450000	145500	Diesel	particular	Manual
Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	particular	Manual
Honda City 2017-2020 EXi	2006	158000	140000	Petroleo	particular	Manual
Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	particular	Manual
Maruti Swift VXI BSIII	2007	130000	120000	Petroleo	particular	Manual
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
Hyundai i20 Asta Optional with Sunroof 1.2	2013	525000	61500	Petroleo	concesionario	Manual
Maruti Swift Dzire LDI	2016	6e+05	150000	Diesel	particular	Manual
Hyundai Xcent 1.2 Kappa SX Option AT	2016	565000	72000	Petroleo	concesionario	Automatic
Maruti Alto LX BSIII	2008	120000	68000	Petroleo	concesionario	Manual
Hyundai i20 1.4 Asta Option	2017	725000	110000	Diesel	particular	Manual

2.2 Estructuración de la base de datos

```
## Rows: 100
## Columns: 12
## $ Nombre        <chr> "Maruti Swift Dzire VDI", "Skoda Rapid 1.5 TDI Ambition"~
## $ Año           <dbl> 2014, 2014, 2006, 2010, 2007, 2017, 2007, 2001, 2011, 20~
## $ Precio        <dbl> 450000, 370000, 158000, 225000, 130000, 440000, 96000, 4~
## $ Kilometros    <dbl> 145500, 120000, 140000, 127000, 120000, 45000, 175000, 5~
## $ Combustible   <chr> "Diesel", "Diesel", "Petroleo", "Diesel", "Petroleo", "P~
```

```
## $ Tipo_vendedor <chr> "particular", "particular", "particular", "particular", ~
## $ Transmicion   <chr> "Manual", "Manual", "Manual", "Manual", "Manual", "Manua~
## $ Dueños        <dbl> 1, 2, 3, 1, 1, 1, 1, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2,~
## $ kilometraje   <dbl> 23.40, 21.14, 17.40, 23.00, 16.10, 20.14, 17.30, 16.10, ~
## $ Motor         <dbl> 1248, 1498, 1497, 1396, 1298, 1197, 1061, 796, 1364, 139~
## $ Potencia      <dbl> 74.00, 103.52, 78.00, 90.00, 88.20, 81.86, 57.50, 37.00,~
## $ Asientos      <dbl> 5, 5, 5, 5, 5, 5, 5, 4, 5, 5, 5, 5, 5, 5, 5, 5, 7, 5, 5,~
```

Los datos tienen 100 filas y 12 columnas. La Nombre son identificadores únicos para cada coche, así que podemos eliminarlos porque no necesitamos esa información.

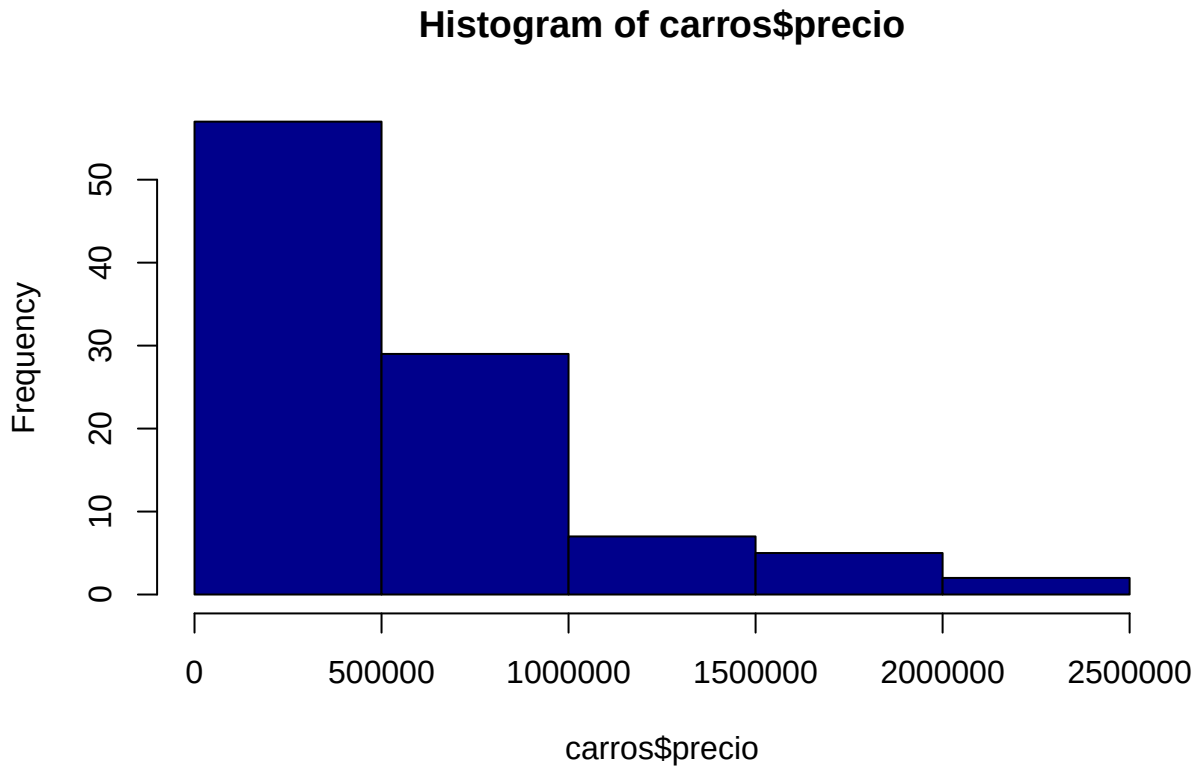
Antes de seguir adelante, primero tenemos que asegurarnos de que nuestros datos son del tipo correcto. Hay algunas características que tenemos que limpiar y poner en el tipo correcto. Lo que hacemos para el siguiente paso es:

- Cambiar los tipos de datos de caracter a factores:
  - `combustible`
  - `tipo_vendedor`
  - `transmisión`
  - `Dueño`
  - `Asientos`

3 Análisis exploratorio de datos

El análisis exploratorio de datos es una fase en la que exploramos las variables de los datos, para ver si hay algún patrón que pueda indicar algún tipo de correlación entre las variables.

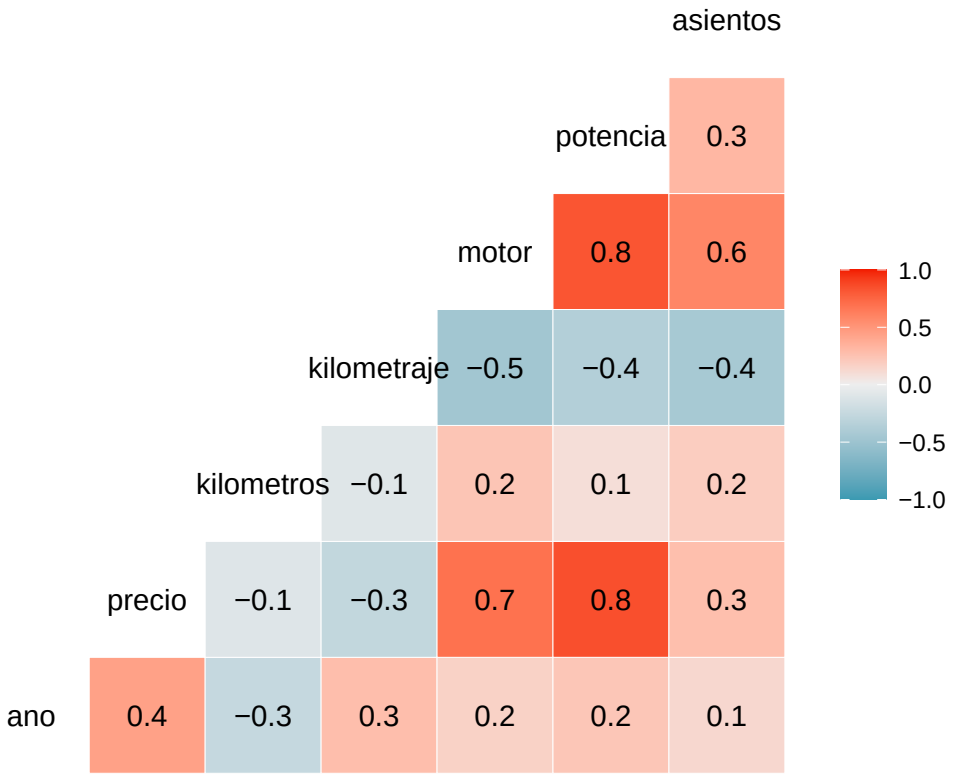
En primer lugar, queremos conocer la distribución de nuestra variable objetivo, que es el `precio`.



Del histograma se desprende que la mayoría de los precios de venta de los coches usados son inferiores a 1.000.000

```
ggcorr(carros, label = T)
```

```
## Warning in ggcorr(carros, label = T): data in column(s) 'combustible',
## 'tipo_vendedor', 'transmicion', 'duenos' are not numeric and were ignored
```



El gráfico muestra que `kilometros` y el `kilometraje` tienen una correlación negativa con el `precio`, en otro caso, el precio de venta tiene una fuerte correlación con la `potencia` (0,8)

4 Modelo

El primer modelo que podemos hacer es utilizar todas las variables (excepto `precio`) como variables predictoras.

```
##
## Call:
## lm(formula = precio ~ ., data = carros)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -653581 -119915   20000  119277  744812
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -6.759e+07  1.550e+07  -4.359 3.60e-05 ***
## ano                     3.353e+04  7.760e+03   4.320 4.16e-05 ***
## kilometros             -8.053e-01  5.137e-01  -1.567  0.12068
## combustiblediesel      -3.237e+04  2.413e+05  -0.134  0.89359
## combustibleGLP          1.666e+05  2.952e+05   0.564  0.57392
## combustiblegasolina     5.410e+03  2.484e+05   0.022  0.98267
```

```
## tipo_vendedorparticular  3.980e+04  5.208e+04   0.764  0.44688
## transmicionManual       -2.203e+05  8.284e+04  -2.659  0.00934 **
## duenos2                  9.965e+04  5.902e+04   1.688  0.09494 .
## duenos3                  1.250e+04  1.176e+05   0.106  0.91565
## kilometraje             -3.264e+03  8.938e+03  -0.365  0.71585
## motor                    1.323e+02  1.254e+02   1.054  0.29467
## potencia                 8.944e+03  1.294e+03   6.911 7.93e-10 ***
## asientos                -1.044e+04  4.153e+04  -0.251  0.80212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 214000 on 86 degrees of freedom
## Multiple R-squared:  0.8203, Adjusted R-squared:  0.7932
## F-statistic: 30.21 on 13 and 86 DF,  p-value: < 2.2e-16
```

El resumen de `model_all` muestra mucha información. Pero por ahora, es mejor centrarse en la $Pr(>|t|)$. Esta columna muestra el nivel de significación de la variable para el modelo. Si el valor es inferior a 0.05, **podemos asumir con seguridad que la variable tiene un efecto significativo en el modelo**.

5 Selección de características

La selección de características es la etapa en la que se seleccionan las variables que se van a utilizar, y se trabaja evaluando y reduciendo las variables no significativas y prestando atención al valor del AIC. El AIC (Criterio de Información de Akaike) es un valor que representa una gran cantidad de información perdida en el modelo, **cuanto menor sea el valor del AIC, mejor será el modelo**.

Hay tres pasos de selección de características que podemos aplicar:

1. **eliminación de los predictores**: De todos los predictores utilizados, se evalúa el modelo reduciendo las variables predictoras de forma que se obtenga el modelo con el menor AIC (Criterio de Información de Akaike)..
2. **selección hacia delante**: Del modelo sin predictor, luego se evalúa el modelo añadiendo variables predictoras de forma que se obtenga el modelo con el menor AIC.
3. **ambos**: A partir del modelo realizado, se puede evaluar el modelo añadiendo o restando variables predictoras de forma que se obtenga el modelo con el menor AIC.

5.1 Backward Elimination

```
##
## Call:
## lm(formula = precio ~ ano + kilometros + transmicion + duenos +
##     potencia, data = carros)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -626234 -134298   14719  126124  755467
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.229e+07  1.265e+07  -4.923 3.68e-06 ***
## ano                3.087e+04  6.283e+03   4.913 3.83e-06 ***
## kilometros        -6.746e-01  4.760e-01  -1.417  0.15976
## transmicionManual -2.069e+05  7.539e+04  -2.745  0.00726 **
## duenos2            1.047e+05  5.365e+04   1.951  0.05407 .
```

```
## duenos3            2.012e+04  1.113e+05   0.181  0.85690
## potencia           1.018e+04  7.029e+02  14.477  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 210300 on 93 degrees of freedom
## Multiple R-squared:  0.8125, Adjusted R-squared:  0.8004
## F-statistic: 67.15 on 6 and 93 DF,  p-value: < 2.2e-16
```

5.2 Forward Elimiination

```
##
## Call:
## lm(formula = precio ~ potencia + ano + transmicion + kilometros +
##     duenos, data = carros)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -626234 -134298   14719  126124  755467
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.229e+07  1.265e+07  -4.923 3.68e-06 ***
## potencia           1.018e+04  7.029e+02  14.477  < 2e-16 ***
## ano                3.087e+04  6.283e+03   4.913 3.83e-06 ***
## transmicionManual -2.069e+05  7.539e+04  -2.745  0.00726 **
## kilometros        -6.746e-01  4.760e-01  -1.417  0.15976
## duenos2            1.047e+05  5.365e+04   1.951  0.05407 .
## duenos3            2.012e+04  1.113e+05   0.181  0.85690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 210300 on 93 degrees of freedom
## Multiple R-squared:  0.8125, Adjusted R-squared:  0.8004
## F-statistic: 67.15 on 6 and 93 DF,  p-value: < 2.2e-16
```

5.3 Both

```
##
## Call:
## lm(formula = precio ~ potencia + ano + transmicion + kilometros +
##     duenos, data = carros)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -626234 -134298   14719  126124  755467
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.229e+07  1.265e+07  -4.923 3.68e-06 ***
## potencia           1.018e+04  7.029e+02  14.477  < 2e-16 ***
## ano                3.087e+04  6.283e+03   4.913 3.83e-06 ***
## transmicionManual -2.069e+05  7.539e+04  -2.745  0.00726 **
## kilometros        -6.746e-01  4.760e-01  -1.417  0.15976
```

```
## duenos2            1.047e+05  5.365e+04   1.951  0.05407 .
## duenos3            2.012e+04  1.113e+05   0.181  0.85690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 210300 on 93 degrees of freedom
## Multiple R-squared:  0.8125, Adjusted R-squared:  0.8004
## F-statistic: 67.15 on 6 and 93 DF,  p-value: < 2.2e-16
```

5.4 Comparación de Modelos

Name	Model	AIC	AIC_wt	BIC	BIC_wt	R2	R2_adjusted	RMSE	Sigma
model_all	lm	2753.493	0.002602	2792.571	0.0000003	0.8203447	0.7931875	198496.8	214044.6
model_back	lm	2743.793	0.332466	2764.634	0.3333332	0.8124519	0.8003521	202810.2	210304.3
model_forward	lm	2743.793	0.332466	2764.634	0.3333332	0.8124519	0.8003521	202810.2	210304.3
model_both	lm	2743.793	0.332466	2764.634	0.3333332	0.8124519	0.8003521	202810.2	210304.3

Después de realizar la selección de características de tres maneras, resulta que no hay ningún cambio significativo en el `model_all` vs los otros 3 modelos tanto por los valores de AIC, R2 (adj.) y RMSE, entonces el modelo que utilizaremos es `model_forward`.

6 Supuesto de regresión lineal

Como modelo estadístico, la regresión lineal tiene varios supuestos que deben cumplirse para que la interpretación obtenida no esté sesgada. Este supuesto sólo debe cumplirse si el propósito de hacer un modelo de regresión lineal es querer una interpretación o ver el efecto de cada predictor sobre el valor de la variable objetivo. Si sólo se quiere utilizar la regresión lineal para hacer predicciones, no es necesario que se cumplan los supuestos del modelo.

6.1 Linealidad

La linealidad significa que la variable objetivo con su predictor tiene una relación lineal o la relación es una línea recta. Además, el efecto o valor del coeficiente entre las variables es aditivo. Si no se cumple esta linealidad, automáticamente todos los valores de los coeficientes que obtengamos no son válidos porque el modelo supone que el patrón que vamos a realizar es lineal.

6.2 Normalidad del residuo (normalidad del residuo)

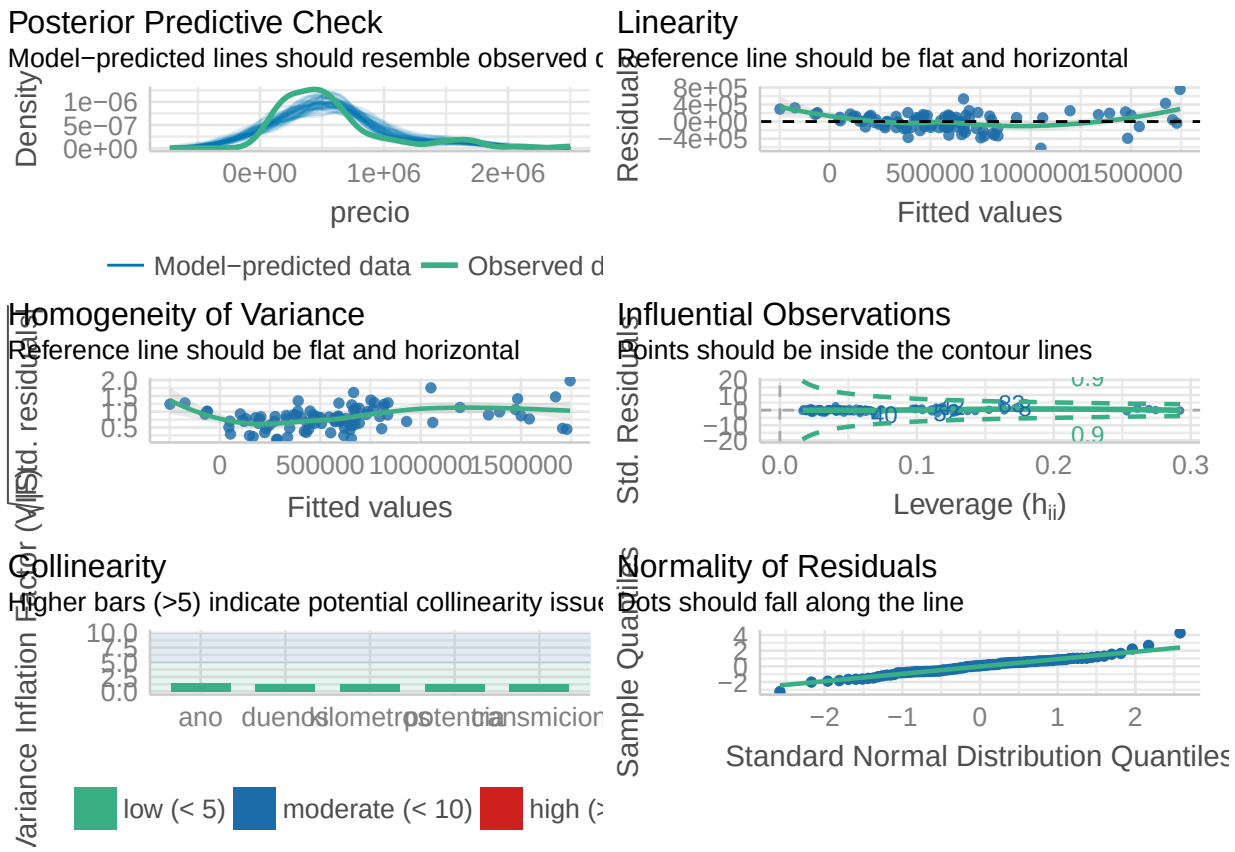
El supuesto de normalidad significa que los residuos del modelo de regresión lineal deben estar distribuidos normalmente porque esperamos obtener residuos cercanos al valor cero

6.3 Homocedasticidad del residuo

La homocedasticidad indica que el residuo o error es constante o no forma un determinado patrón. Si el error forma un determinado patrón, como una línea lineal o cónica, lo llamamos heterocedasticidad y afectará al valor del error estándar en una estimación/coeficiente de predictor sesgado (demasiado estrecho o demasiado ancho). La homocedasticidad puede comprobarse visualmente viendo si existe un patrón entre los resultados predichos de los datos y el valor residual.

6.4 No hay multicolinealidad

La multicolinealidad se produce cuando las variables predictoras utilizadas en el modelo tienen una fuerte relación. no se espera que un buen modelo tenga multicolinealidad. La presencia o ausencia de multicolinealidad puede verse a partir del valor del VIF (Factor de Inflación de la Varianza). Cuando el valor del VIF es superior a 10, significa que hay multicolinealidad



7 Modelo con datos escalados

La imagen anterior muestra que los supuestos que se cumplen son sólo la multicolinealidad, mientras que los otros no son apropiados. Ahora intentaré hacer un escalado de datos en las variables predictoras y en la variable objetivo para superar la normalidad del residuo y la heterocedasticidad

```
##
## Call:
## lm(formula = precio ~ ., data = car_scale)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.38862 -0.25478  0.04249  0.25342  1.58245
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.32651    0.54229   0.602  0.54870
## ano                       0.28649    0.06631   4.320 4.16e-05 ***
## kilometros               -0.08372    0.05341  -1.567  0.12068
## kilometraje              -0.02958    0.08099  -0.365  0.71585
## motor                     0.13102    0.12426   1.054  0.29467
## potencia                  0.64898    0.09390   6.911 7.93e-10 ***
## asientos                 -0.01664    0.06620  -0.251  0.80212
## combustiblediesel        -0.06878    0.51267  -0.134  0.89359
## combustibleGLP            0.35403    0.62721   0.564  0.57392
## combustiblegasolina       0.01149    0.52773   0.022  0.98267
```

```
## tipo_vendedorparticular  0.08456    0.11066   0.764  0.44688
## transmicionManual       -0.46800    0.17600  -2.659  0.00934 **
## duenos2                  0.21172    0.12539   1.688  0.09494 .
## duenos3                  0.02655    0.24991   0.106  0.91565
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4548 on 86 degrees of freedom
## Multiple R-squared:  0.8203, Adjusted R-squared:  0.7932
## F-statistic: 30.21 on 13 and 86 DF,  p-value: < 2.2e-16
```

Después de crear un modelo con datos que han sido escalados, resulta que los resultados dados no son muy diferentes del modelo_todo y producen un mismo valor R^2 de ≈ 0.80, y los supuestos que se cumplen son sólo la multicolinealidad. Por esta razón, se recomienda utilizar otros modelos de regresión en estos datos como la regresión polinómica, la regresión de bosque aleatorio, etc.

8 Predicciones

```
##        fit        lwr       upr
## 1 329882.6  -96906.03  756671.3
## 2 752175.7  322501.03 1181850.3
## 3 147450.3 -323998.17  618898.7
## 4 381703.0  -44494.60  807900.7
## 5 275493.0 -156039.01  707025.0
## 6 570286.6  147996.56  992576.7
```

9 Evaluación

Después de hacer predicciones a partir de los datos, debemos averiguar si el modelo de aprendizaje automático que se ha creado puede producir predicciones con el menor error. Hay varias formas de realizar la evaluación del modelo de regresión. Para realizar la evaluación del modelo de regresión, hay varias métricas que se pueden utilizar:

- R-cuadrado y R-cuadrado adjunto: para determinar lo bien que el modelo explica la varianza de la variable objetivo
- Valor de error : para ver si la predicción realizada produce el menor valor de error

Los valores de error que utilizaremos para ver el rendimiento del modelo son MAE (error absoluto medio) y MAPE. El MAE muestra la media de los valores de error absoluto, mientras que el MAPE muestra la magnitud de la desviación en términos porcentuales.

$$MAE = \frac{\sum |\hat{y} - y|}{n}$$

$$MAPE = \frac{1}{n} \sum \frac{|\hat{y} - y|}{y}$$

	value
min	1.500000e+04
max	2.500000e+06
mape	1.544055e+00
mae	5.690309e+05

Si se observa el MAPE (0.69), significa que el error en la predicción de este modelo es de alrededor del 69%, por lo que se puede decir que la regresión lineal no es adecuada para predecir el precio de venta de los coches usados en estos datos.

10 Conclusión

Los datos son un historial de ventas de coches usados procedentes de kaggle.com. El objetivo de este análisis es crear un modelo que pueda predecir el precio de venta de un coche usado basándose en varias características existentes. se han utilizado modelos de regresión lineal y modelos de sintonía mediante la selección de características, pero siguen dando los mismos resultados. sólo se cumple un supuesto de la regresión lineal, la multicolinealidad, mientras que no se cumplen los supuestos de normalidad del residuo, linealidad y homocedasticidad.

De los resultados del análisis realizado concluyo que la regresión lineal no es adecuada para predecir el precio de venta de los coches usados en estos datos, se recomienda utilizar otros modelos de regresión en estos datos como la regresión polinómica, la regresión de bosque aleatorio, etc.

11 Codigo

```
## ----message=FALSE, warning=FALSE, include=FALSE-------------------------------------------------
## Importar paquetes requeridos
library(readr)
library(tidyverse)
library(kableExtra)
library(magrittr)
library(ggExtra)
library(GGally)
library(janitor)
library(tidystats)
library(car)
library(faraway)
library(lmtest)
library(caret)
library(data.table)
library(MLmetrics)
library(performance)
library(mctest)
library(see)


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
carros <- read_csv("Trabajo1/carros.csv")

kable(rbind(head(carros, n = 5),rep(".", ncol(carros)),
            rep(".", ncol(carros)),rep(".", ncol(carros)),
            tail(carros, n = 5)),digits = 30, align = "c")


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
glimpse(carros)


## ----message=FALSE, warning=FALSE, include=FALSE-------------------------------------------------
carros <- carros[-1]

carros %<>% clean_names()

carros$combustible %<>% as.factor()
```

```
levels(carros$combustible) <- c("GNC","diesel","GLP","gasolina")

carros$tipo_vendedor %<>%  as.factor()

carros$transmicion %<>%  as.factor()

carros$duenos %<>%  as.factor()

# carros$asientos %<>%  as.factor()


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
hist(carros$precio, col="darkblue")


## ------------------------------------------------------------------------------------------------
ggcorr(carros, label = T)


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
model0 <- lm(precio~1, carros)
model_all <- lm(precio~., carros)
summary(model_all)


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
model_back <- step(model_all,direction = "backward", trace = 0)
summary(model_back)


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
model_forward <- step(model0,
                      direction = "forward",
                      scope = list(lower = model0,upper = model_all),
                      trace = 0)
summary(model_forward)


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
model_both <- step(model0,
                   direction = "both",
                   scope =list(lower = model0,upper = model_all),
                   trace = 0)
summary(model_both)


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
compare_performance(model_all,model_back,model_forward,model_both)


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
performance::check_model(model_forward)
```

```
## ----message=FALSE, warning=FALSE, include=FALSE-------------------------------------------------
num_data <- carros %>% select(is.numeric) %>% sapply(scale)


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
fac_data <- carros %>% select(is.factor)
car_scale <- data.frame(num_data,fac_data)

model_scale <- lm(precio~., car_scale)
summary(model_scale)


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
pred <- predict(model_forward, newdata = carros,
                interval = "prediction", level = 0.95)
head(pred)


## ----echo=FALSE, message=FALSE, warning=FALSE-----------------------------------------------------
RNGkind(sample.kind = "Rounding")
set.seed(123)
intrain <- sample(x=nrow(carros), size = nrow(carros)*0.8)
car_train <- carros[intrain,]
car_test <- carros[-intrain,]

min <- min(carros$precio)
max <- max(carros$precio)
mape <- MAPE(y_pred = pred, y_true = car_test$precio)
mae <- MAE(y_pred = pred, y_true = car_test$precio)

value <- c(min,max,mape,mae)
eval <- as.data.frame(value)
row.names(eval) <- c("min","max","mape","mae")
eval
```