Actividad Integradora 2

Luis Maximiliano López Ramírez

2024-09-06

Una empresa automovilística china aspira a entrar en el mercado estadounidense. Desea establecer allí una unidad de fabricación y producir automóviles localmente para competir con sus contrapartes estadounidenses y europeas. Contrataron una empresa de consultoría de automóviles para identificar los principales factores de los que depende el precio de los automóviles, específicamente, en el mercado estadounidense, ya que pueden ser muy diferentes del mercado chino. Esencialmente, la empresa quiere saber:

- Qué variables son significativas para predecir el precio de un automóvil
- Qué tan bien describen esas variables el precio de un automóvil

Con base en varias encuestas de mercado, la consultora ha recopilado un gran conjunto de datos de diferentes tipos de automóviles en el mercado estadounidense que presenta en el siguiente archivo. Las variables recopiladas vienen descritas en el diccionario de términos. Por un análisis de correlación, la empresa automovilistica tiene interés en analizar las variables agrupadas de la siguiente forma para hacer el análisis de variables significativas:

- **Primer grupo. Distancia entre los ejes (wheelbase), tipo de gasolina que usa y caballos de fuerza** *Segundo grupo. Altura del auto, ancho del auto y si es convertible o no.* Tercer grupo. Tamaño del motor (ensinesize), carrera o lanzamiento del pistón (stroke) y localización del motor en el carro

Selecciona uno de los tres grupos analizados (te será asignado por tu profesora) y analiza la significancia de las variables para predecir o influir en la variable precio. ¿propondrías una nueva agrupación a la empresa automovilísitica?

```r
datos <- read.csv("precios_autos.csv")
```

 # 1. Exploración de base

1. Calcula medidas estadísticas apropiadas para las variables

1. Cuantitativas (media, desviación estándar, ciartiles, etc)

```r
M1 <- data.frame(datos$wheelbase, datos$horsepower, datos$price)

# Número de variables
n <- 3

# Crear una matriz vacía para almacenar las estadísticas
d <- matrix(NA, ncol = 7, nrow = n)
```

```r
# Calcular las estadísticas para cada columna
for(i in 1:n) {
  d[i,] <- c(as.numeric(summary(M1[, i])), sd(M1[, i], na.rm = TRUE))
}
m <- as.data.frame(d)

# Asignar nombres a las filas y columnas
row.names(m) <- c("Wheelbase", "Horsepower", "Price")
names(m) <- c("Minimo", "Q1", "Mediana", "Media", "Q3", "Máximo", "Desv
Est")
m
```

```
##             Minimo     Q1 Mediana       Media      Q3  Máximo    Desv
Est
## Wheelbase    86.6   94.5      97    98.75659   102.4   120.9
6.021776
## Horsepower   48.0   70.0      95   104.11707   116.0   288.0
39.544167
## Price      5118.0 7788.0   10295 13276.71057 16503.0 45400.0
7988.852332
```

2. Cualitativas: cuantiles, frecuencias (puedes usar el comando table o prop.table)

```r
# Calcular las frecuencias absolutas de la variable cualitativa
'fueltype'
frecuencias <- table(datos$fueltype)

# Calcular las frecuencias relativas (proporción)
frecuencia_relativa <- prop.table(frecuencias)

# Calcular las frecuencias acumuladas
frecuencia_acumulada <- cumsum(frecuencias)

# Calcular la frecuencia relativa acumulada (percentiles)
frecuencia_relativa_acumulada <- cumsum(frecuencia_relativa)

# Crear un DataFrame con los resultados
resultados_cualitativos <- data.frame(
  Categoria = names(frecuencias),
  Frecuencia = as.numeric(frecuencias),
  Frecuencia_Relativa = round(as.numeric(frecuencia_relativa) * 100, 2),
# En porcentaje
  Frecuencia_Acumulada = frecuencia_acumulada,
  Frecuencia_Relativa_Acumulada =
round(as.numeric(frecuencia_relativa_acumulada) * 100, 2) # En porcentaje
)

# Mostrar los resultados
resultados_cualitativos
```

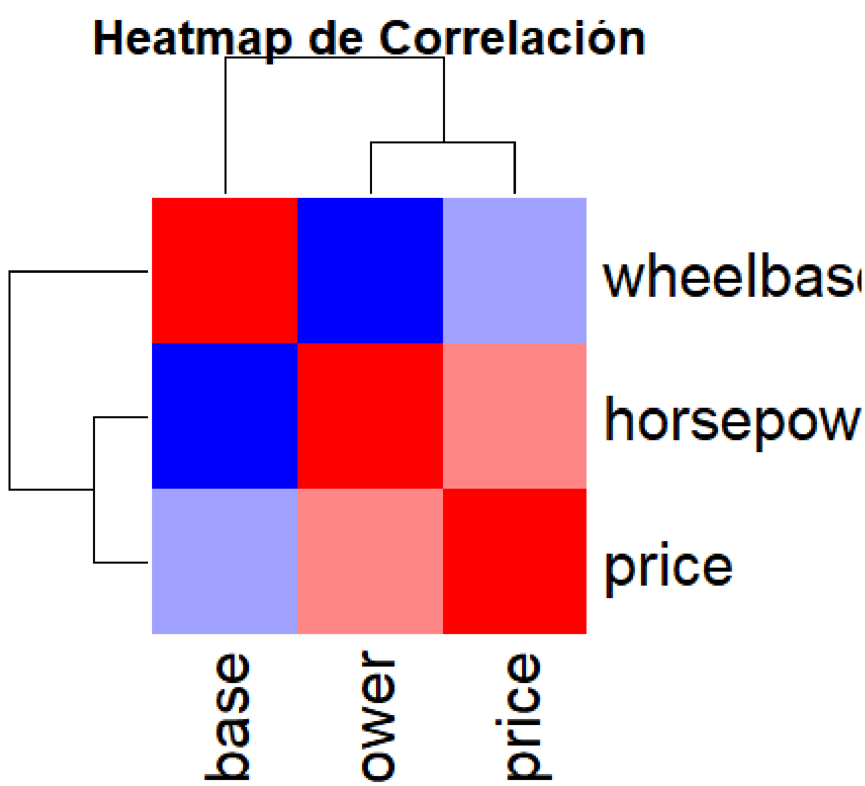
```
##         Categoria Frecuencia Frecuencia_Relativa Frecuencia_Acumulada
## diesel    diesel         20                9.76                   20
## gas          gas        185               90.24                  205
##         Frecuencia_Relativa_Acumulada
## diesel                           9.76
## gas                            100.00
```

2. Analiza la correlación entre las variables (analiza posible colinealidad entre las variables)

```r
# Seleccionar las columnas numéricas de interés
variables_numericas <- datos[, c("wheelbase", "horsepower", "price")]

# Calcular la matriz de correlación
matriz_correlacion <- cor(variables_numericas, use = "complete.obs")

# Mostrar la matriz de correlación
print(matriz_correlacion)
```

```
##            wheelbase horsepower     price
## wheelbase  1.0000000  0.3532945 0.5778156
## horsepower 0.3532945  1.0000000 0.8081388
## price      0.5778156  0.8081388 1.0000000
```

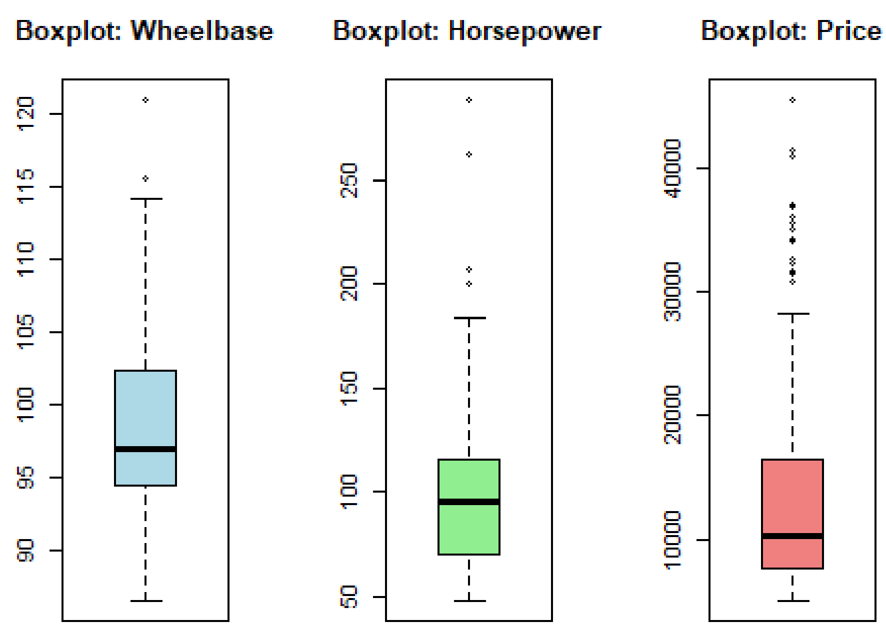
```r
# Visualizar la matriz de correlación con un heatmap
heatmap(matriz_correlacion, main = "Heatmap de Correlación", symm = TRUE,
col = colorRampPalette(c("blue", "white", "red"))(20))
```



3. Explora los datos usando herramientas de visualización (si lo consideras necesario):

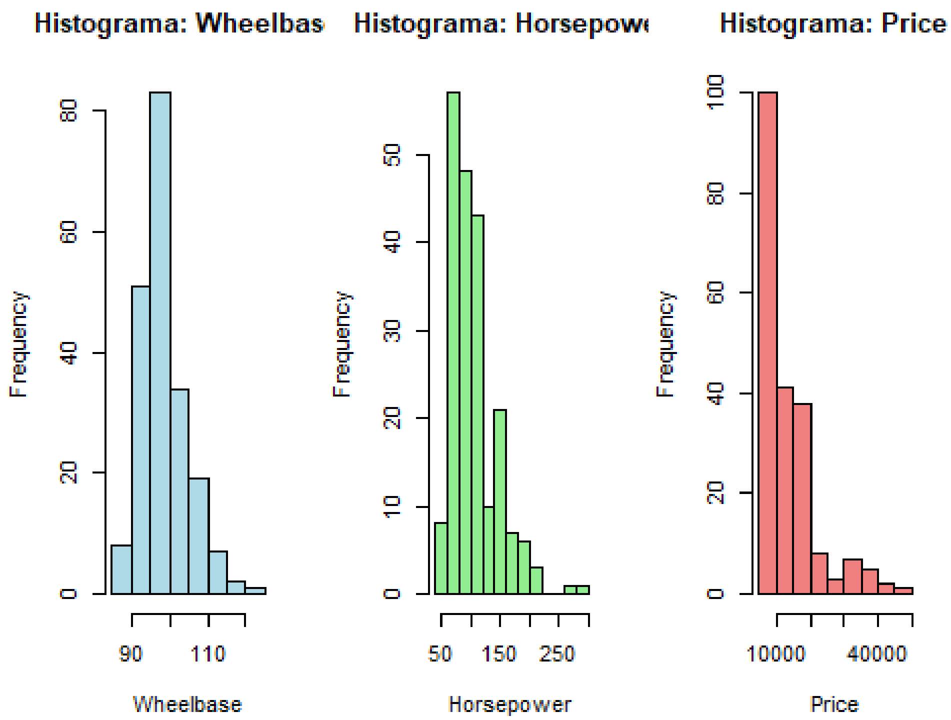
1. Variables cuantitativas (Boxplot, Histogramas, Diagramas de dispersión y correlación por pares)

```
# Seleccionar las variables cuantitativas de interés
variables_numericas <- datos[, c("wheelbase", "horsepower", "price")]

# Boxplots para cada variable
par(mfrow = c(1, 3)) # Configurar la gráfica para mostrar tres boxplots
en una fila
boxplot(variables_numericas$wheelbase, main = "Boxplot: Wheelbase", col =
"lightblue")
boxplot(variables_numericas$horsepower, main = "Boxplot: Horsepower", col
= "lightgreen")
boxplot(variables_numericas$price, main = "Boxplot: Price", col =
"lightcoral")
```

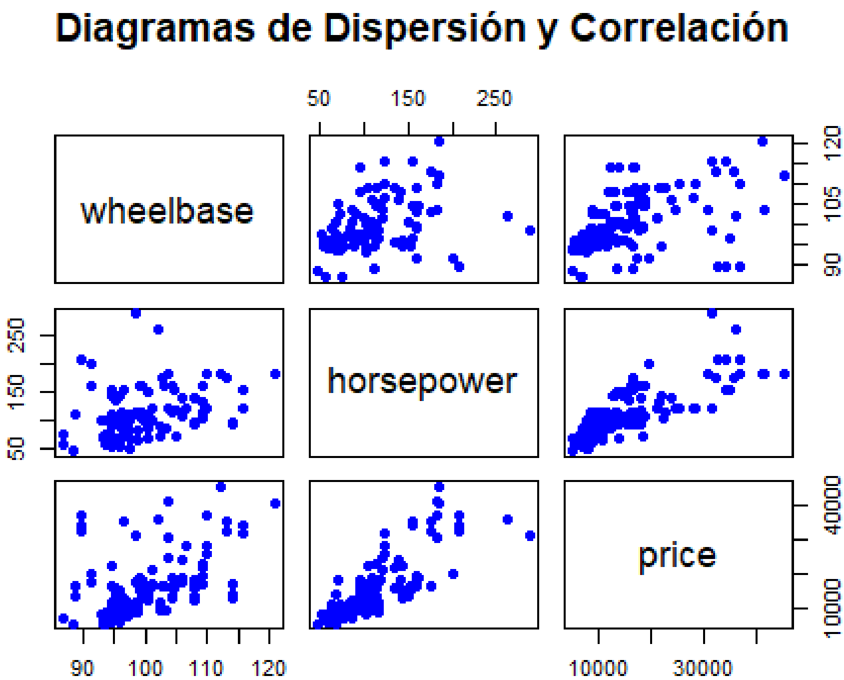



```
# Histogramas para cada variable
par(mfrow = c(1, 3)) # Configurar la gráfica para mostrar tres
histogramas en una fila
hist(variables_numericas$wheelbase, main = "Histograma: Wheelbase", col =
"lightblue", xlab = "Wheelbase")
hist(variables_numericas$horsepower, main = "Histograma: Horsepower", col
= "lightgreen", xlab = "Horsepower")
```

```r
hist(variables_numericas$price, main = "Histograma: Price", col =
"lightcoral", xlab = "Price")
```




```r
# Diagramas de dispersión y correlación por pares
pairs(variables_numericas, main = "Diagramas de Dispersión y
Correlación", col = "blue", pch = 16)
```

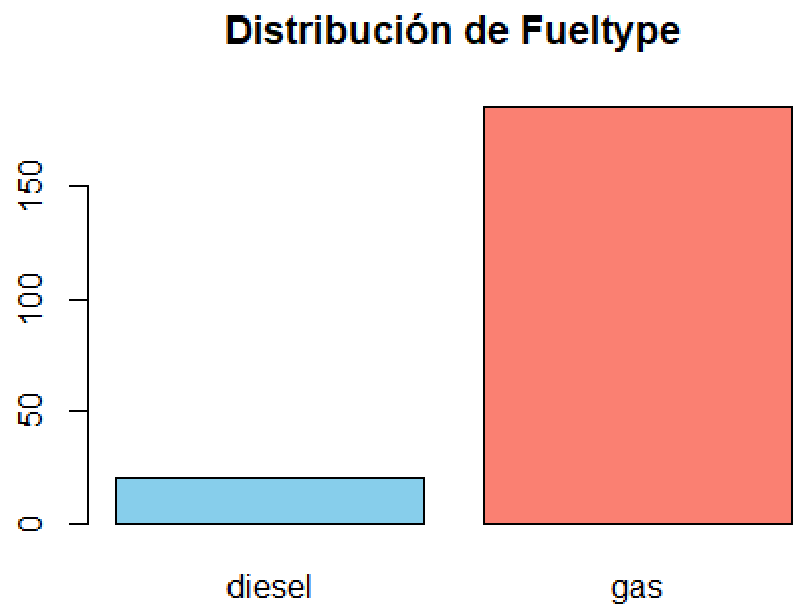



```r
# Mostrar la matriz de correlación numérica
correlacion_pares <- cor(variables_numericas, use = "complete.obs")
print(correlacion_pares)
```

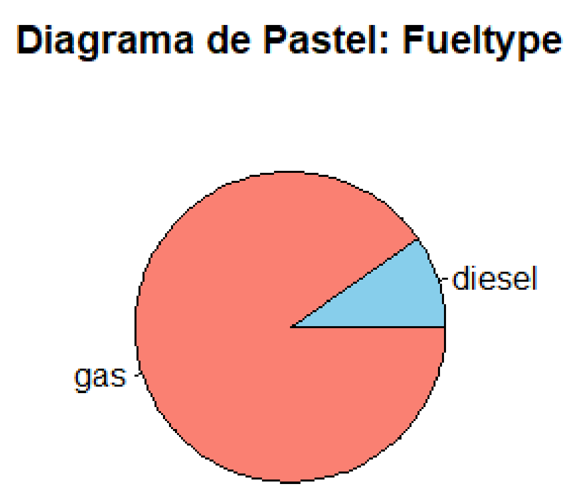
```
##             wheelbase horsepower     price
## wheelbase  1.0000000  0.3532945 0.5778156
## horsepower 0.3532945  1.0000000 0.8081388
## price      0.5778156  0.8081388 1.0000000
```

2. Variables categóricas (Distribución de los datos (diagramas de barras, diagramas de pastel), Boxplot por categoría de las variables cuantitativas)

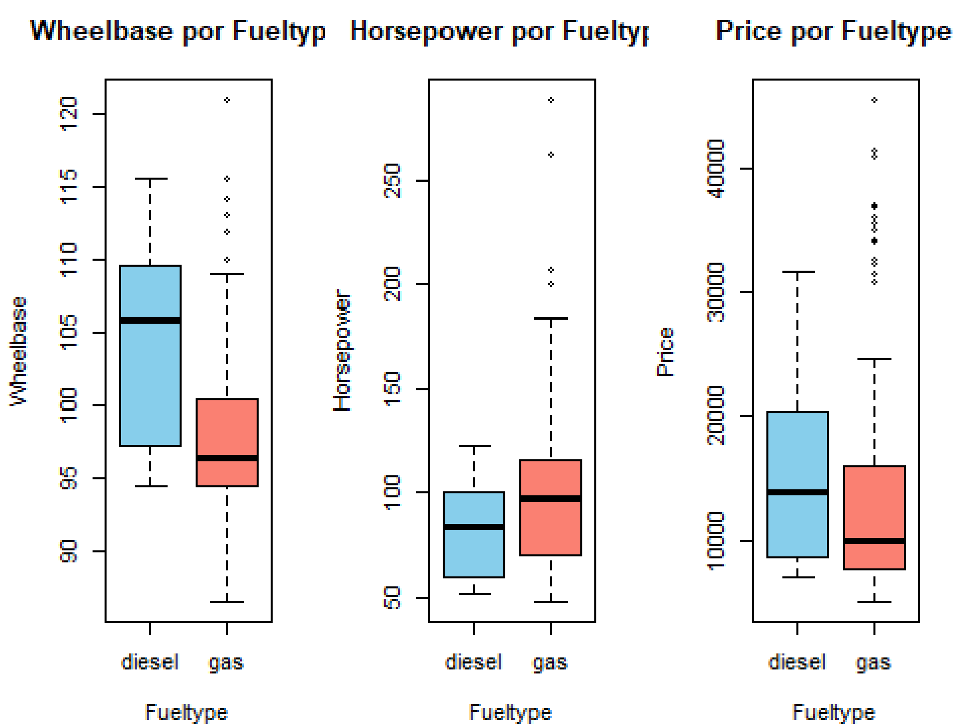
```r
# Distribución de la variable categórica 'fueltype' (puedes cambiar por
otras variables categóricas)
# Diagrama de barras
barplot(table(datos$fueltype), main = "Distribución de Fueltype", col =
c("skyblue", "salmon"))
```




```r
# Diagrama de pastel
pie(table(datos$fueltype), main = "Diagrama de Pastel: Fueltype", col =
c("skyblue", "salmon"))
```



```r
# Boxplots de variables cuantitativas por categoría de 'fueltype'
par(mfrow = c(1, 3)) # Configurar para mostrar tres gráficos en una fila

# Boxplot de 'wheelbase' por 'fueltype'
boxplot(datos$wheelbase ~ datos$fueltype, main = "Wheelbase por
Fueltype",
        col = c("skyblue", "salmon"), xlab = "Fueltype", ylab =
"Wheelbase")

# Boxplot de 'horsepower' por 'fueltype'
boxplot(datos$horsepower ~ datos$fueltype, main = "Horsepower por
Fueltype",
        col = c("skyblue", "salmon"), xlab = "Fueltype", ylab =
"Horsepower")

# Boxplot de 'price' por 'fueltype'
boxplot(datos$price ~ datos$fueltype, main = "Price por Fueltype",
        col = c("skyblue", "salmon"), xlab = "Fueltype", ylab = "Price")
```



2. Modelación y verificación del modelo

1. Encuentra la ecuación de regresión de mejor ajuste. Propón al menos 2 modelos de ajuste para encontrar la mejor forma de ajustar la variable precio.

```r
datos$Gas_numeric <- ifelse(datos$fueltype == 'diesel', 0, 1)

# Sin interacción
```

```r
modelo_1 <- lm(price ~ wheelbase + horsepower + Gas_numeric, data =
datos)

# Con interacción
modelo_2 <- lm(price ~ wheelbase * horsepower, data = datos)
```

2. Para cada uno de los modelos propuestos:

 # 1. Realiza la regresión entre las variables involucradas

```r
# Sin interacción
modelo_1 <- lm(price ~ wheelbase + horsepower + Gas_numeric, data =
datos)

# Con interacción
modelo_2 <- lm(price ~ wheelbase * horsepower, data = datos)
```

 # 2. Analiza la significancia del modelo:

1. Valida la significancia del modelo con un alfa de 0.04 (incluye las hipótesis que pruebas y el valor frontera)

Para ambos modelos (y), las hipótesis son:

$$H_0: \beta_1 = \beta_2 = \cdots = \beta_k = 0$$

Esto implica que todas las variables predictoras no tienen un efecto significativo sobre la variable respuesta ().

$$H_a: \beta_i \neq 0 \quad \text{para al menos una } \beta_i$$

Esto indica que al menos una de las variables predictoras tiene un efecto significativo sobre la variable respuesta ().

Comparamos el valor p con el nivel de significancia $\alpha = 0.04$:

$$\begin{cases} \text{Si } p \leq \alpha, & \text{rechazamos } H_0 \text{ y concluimos que el modelo es significativo.} \\ \text{Si } p > \alpha, & \text{no rechazamos } H_0 \text{ y concluimos que el modelo no es significativo.} \end{cases}$$

```r
# Resumen de los modelos para obtener los valores p globales
resumen_modelo_1 <- summary(modelo_1)
resumen_modelo_2 <- summary(modelo_2)

# Valores p del test F de ambos modelos
p_valor_modelo_1 <- pf(resumen_modelo_1$fstatistic[1],
                       resumen_modelo_1$fstatistic[2],
                       resumen_modelo_1$fstatistic[3],
                       lower.tail = FALSE)

p_valor_modelo_2 <- pf(resumen_modelo_2$fstatistic[1],
                       resumen_modelo_2$fstatistic[2],
                       resumen_modelo_2$fstatistic[3],
```

```
                         lower.tail = FALSE)

# Comparar con alfa = 0.04
cat("Modelo 1 (Sin Interacción):\n")

## Modelo 1 (Sin Interacción):

cat("Valor p =", p_valor_modelo_1, "\n")

## Valor p = 2.526837e-63

if (p_valor_modelo_1 <= 0.04) {
  cat("El modelo 1 es significativo a un nivel de alfa = 0.04.\n\n")
} else {
  cat("El modelo 1 NO es significativo a un nivel de alfa = 0.04.\n\n")
}

## El modelo 1 es significativo a un nivel de alfa = 0.04.

cat("Modelo 2 (Con Interacción):\n")

## Modelo 2 (Con Interacción):

cat("Valor p =", p_valor_modelo_2, "\n")

## Valor p = 2.855049e-61

if (p_valor_modelo_2 <= 0.04) {
  cat("El modelo 2 es significativo a un nivel de alfa = 0.04.\n")
} else {
  cat("El modelo 2 NO es significativo a un nivel de alfa = 0.04.\n")
}

## El modelo 2 es significativo a un nivel de alfa = 0.04.
```

2. Valida la significancia de Beta_i con un alfa de 0.04 (incluye las hipótesis que pruebas y el valor frontera de cada una de ellas)

Para cada coeficiente $\hat{\beta}_i$ en el modelo, las hipótesis son:

$$H_0: \beta_i = 0$$

Esto implica que el coeficiente $\hat{\beta}_i$ no tiene un efecto significativo sobre la variable de respuesta ().

$$H_a: \beta_i \neq 0$$

Esto indica que el coeficiente $\hat{\beta}_i$ tiene un efecto significativo sobre la variable de respuesta ().

Comparar el valor p asociado a cada $\hat{\beta}_i$ con el nivel de significancia $\alpha = 0.04$:

$$\begin{cases} \text{Si } p \le \alpha, & \text{rechazamos } H_0 \text{ y concluimos que } \hat{\beta}_i \text{ es significativo.} \\ \text{Si } p > \alpha, & \text{no rechazamos } H_0 \text{ y concluimos que } \hat{\beta}_i \text{ no es significativo.} \end{cases}$$

```r
# Resumen de los modelos para obtener los valores p de los coeficientes
resumen_modelo_1 <- summary(modelo_1)
resumen_modelo_2 <- summary(modelo_2)

# Extraer los valores p de los coeficientes
p_valores_1 <- resumen_modelo_1$coefficients[, 4]
p_valores_2 <- resumen_modelo_2$coefficients[, 4]

# Nivel de significancia
alfa <- 0.04

# Función para evaluar la significancia de cada coeficiente
validar_significancia <- function(p_valores, alfa, modelo) {
  for (i in seq_along(p_valores)) {
    cat(paste("Coeficiente:", names(p_valores)[i], "\n"))
    cat(paste("Valor p =", round(p_valores[i], 4), "\n"))
    if (p_valores[i] <= alfa) {
      cat(paste("El coeficiente es significativo a un nivel de alfa =",
alfa, ".\n\n"))
    } else {
      cat(paste("El coeficiente NO es significativo a un nivel de alfa
=", alfa, ".\n\n"))
    }
  }
}

# Validar significancia de los coeficientes del modelo 1
cat("Modelo 1 (Sin Interacción):\n")

## Modelo 1 (Sin Interacción):

validar_significancia(p_valores_1, alfa, "Modelo 1")

## Coeficiente: (Intercept)
## Valor p = 0
## El coeficiente es significativo a un nivel de alfa = 0.04 .
##
## Coeficiente: wheelbase
## Valor p = 0
## El coeficiente es significativo a un nivel de alfa = 0.04 .
##
## Coeficiente: horsepower
## Valor p = 0
## El coeficiente es significativo a un nivel de alfa = 0.04 .
##
## Coeficiente: Gas_numeric
```

```
## Valor p = 2e-04
## El coeficiente es significativo a un nivel de alfa = 0.04 .
```

```r
# Validar significancia de los coeficientes del modelo 2
cat("Modelo 2 (Con Interacción):\n")
```

```
## Modelo 2 (Con Interacción):
```

```r
validar_significancia(p_valores_2, alfa, "Modelo 2")
```

```
## Coeficiente: (Intercept)
## Valor p = 0.2368
## El coeficiente NO es significativo a un nivel de alfa = 0.04 .
##
## Coeficiente: wheelbase
## Valor p = 0.2943
## El coeficiente NO es significativo a un nivel de alfa = 0.04 .
##
## Coeficiente: horsepower
## Valor p = 0.4231
## El coeficiente NO es significativo a un nivel de alfa = 0.04 .
##
## Coeficiente: wheelbase:horsepower
## Valor p = 0.0412
## El coeficiente NO es significativo a un nivel de alfa = 0.04 .
```

3. Indica cuál es el porcentaje de variación explicada por el modelo.

```r
# Resumen de los modelos para extraer el R^2
resumen_modelo_1 <- summary(modelo_1)
resumen_modelo_2 <- summary(modelo_2)

# Extraer R^2 de cada modelo
r2_modelo_1 <- resumen_modelo_1$r.squared
r2_modelo_2 <- resumen_modelo_2$r.squared

# Calcular el porcentaje de variación explicada
porcentaje_variacion_modelo_1 <- r2_modelo_1 * 100
porcentaje_variacion_modelo_2 <- r2_modelo_2 * 100

# Mostrar los resultados
cat("Modelo 1 (Sin Interacción):\n")
```

```
## Modelo 1 (Sin Interacción):
```

```r
cat("Porcentaje de variación explicada:",
round(porcentaje_variacion_modelo_1, 2), "%\n\n")
```

```
## Porcentaje de variación explicada: 76.71 %
```

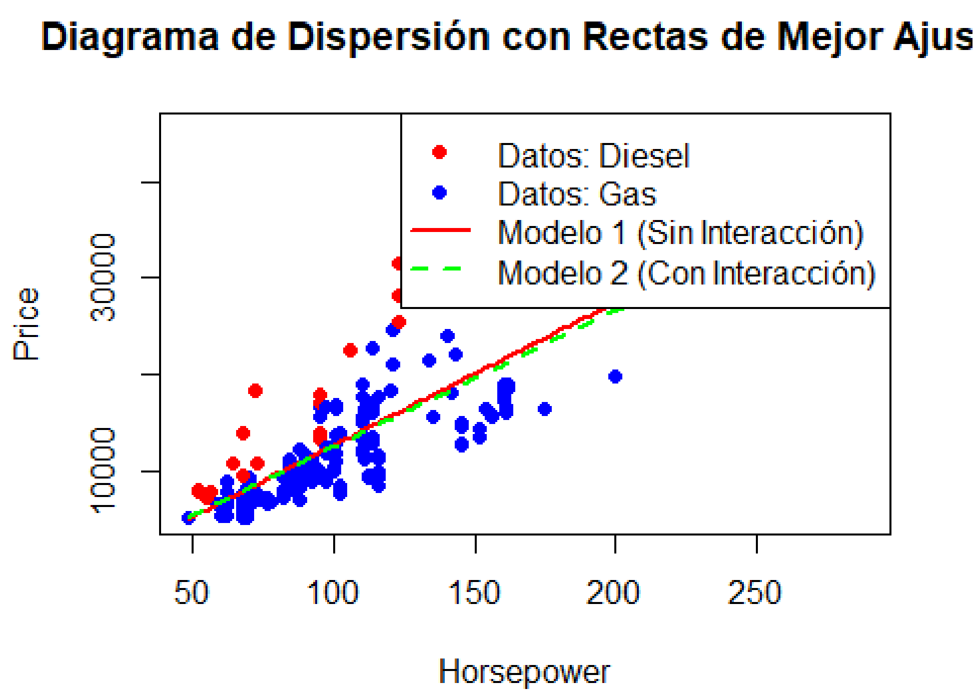
```r
cat("Modelo 2 (Con Interacción):\n")
```

```
## Modelo 2 (Con Interacción):

cat("Porcentaje de variación explicada:",
round(porcentaje_variacion_modelo_2, 2), "%\n")

## Porcentaje de variación explicada: 75.58 %
```

4. Dibuja el diagrama de dispersión de los datos por pares y la recta de mejor ajuste.

```
# Leer los datos desde el archivo CSV
datos <- read.csv("precios_autos.csv")

# Crear la variable Gas_numeric según tu definición
datos$Gas_numeric <- ifelse(datos$fueltype == 'diesel', 0, 1)

# Ajustar los modelos de regresión
modelo_1 <- lm(price ~ wheelbase + horsepower + Gas_numeric, data =
datos)
modelo_2 <- lm(price ~ wheelbase * horsepower, data = datos)

# Dibujar el diagrama de dispersión de horsepower y price con diferentes
colores
plot(datos$horsepower, datos$price,
     main = "Diagrama de Dispersión con Rectas de Mejor Ajuste",
     xlab = "Horsepower",
     ylab = "Price",
     pch = 19,               # Tipo de punto
     col = ifelse(datos$Gas_numeric == 0, "red", "blue"))  # Color rojo
para diesel, azul para gas

# Crear un rango de valores de horsepower para la predicción
horsepower_vals <- seq(min(datos$horsepower), max(datos$horsepower),
length.out = 100)

# Predecir valores del modelo 1 (sin interacción) usando los valores de
horsepower
pred_1 <- predict(modelo_1, newdata = data.frame(
  wheelbase = mean(datos$wheelbase),  # Usamos el promedio de wheelbase
  horsepower = horsepower_vals,
  Gas_numeric = mean(datos$Gas_numeric)  # Usamos el promedio de
Gas_numeric para la predicción
))

# Dibujar la recta de ajuste del modelo 1
lines(horsepower_vals, pred_1, col = "red", lwd = 2, lty = 1)

# Predecir valores del modelo 2 (con interacción) usando los valores de
horsepower
pred_2 <- predict(modelo_2, newdata = data.frame(
  wheelbase = mean(datos$wheelbase),  # Usamos el promedio de wheelbase
```

```
  horsepower = horsepower_vals
))

# Dibujar la recta de ajuste del modelo 2
lines(horsepower_vals, pred_2, col = "green", lwd = 2, lty = 2)

# Añadir leyenda para identificar cada modelo y tipos de combustible
legend("topright",
       legend = c("Datos: Diesel", "Datos: Gas", "Modelo 1 (Sin
Interacción)", "Modelo 2 (Con Interacción)"),
       col = c("red", "blue", "red", "green"),
       pch = c(19, 19, NA, NA),
       lwd = c(NA, NA, 2, 2),
       lty = c(NA, NA, 1, 2))
```



5. Interpreta en el contexto del problema cada uno de los análisis que hiciste.

- **Modelo 1** tiene un mayor porcentaje de variación explicada y todos los coeficientes son significativos, lo que lo hace más robusto y fiable en la predicción de price.

- **Modelo 2** no logra explicar mejor la variación y sus coeficientes individuales no son significativos, lo que sugiere que la interacción no mejora el ajuste.

 # 3. Analiza la validez de los modelos propuestos:

1. Normalidad de los residuos

% Hipótesis de Normalidad (Anderson-Darling)

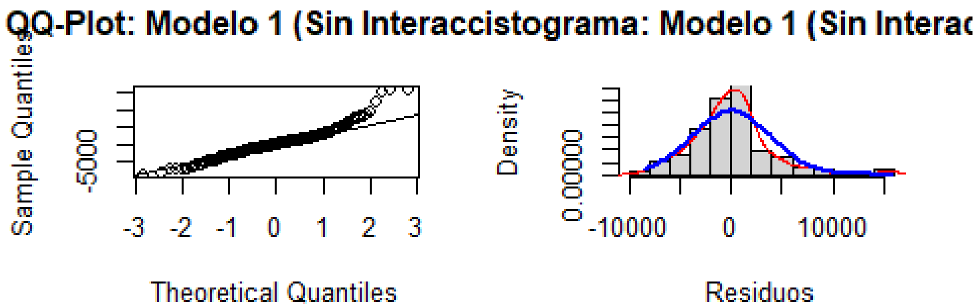
$$\begin{cases} H_0\text{:Los residuos siguen una distribución normal.} \\ H_1\text{:Los residuos no siguen una distribución normal.} \end{cases}$$

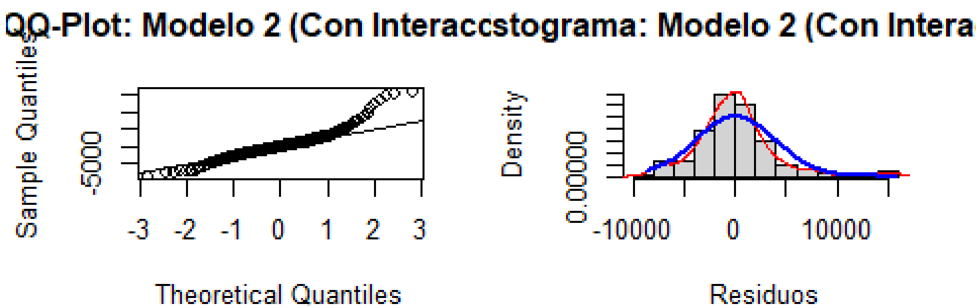
```r
# Cargar la librería para la prueba Anderson-Darling
library(nortest)

# Realizar la prueba de Anderson-Darling sobre los residuos de los
modelos
resultado_modelo_1 <- ad.test(modelo_1$residuals)
resultado_modelo_2 <- ad.test(modelo_2$residuals)

# Mostrar los resultados de las pruebas
cat("Resultados de la prueba de normalidad Anderson-Darling:\n")

## Resultados de la prueba de normalidad Anderson-Darling:

cat("\nModelo 1 (Sin Interacción):\n")

##
## Modelo 1 (Sin Interacción):

print(resultado_modelo_1)

##
##  Anderson-Darling normality test
##
## data:  modelo_1$residuals
## A = 2.7561, p-value = 5.82e-07

cat("\nModelo 2 (Con Interacción):\n")

##
## Modelo 2 (Con Interacción):

print(resultado_modelo_2)

##
##  Anderson-Darling normality test
##
## data:  modelo_2$residuals
## A = 3.6742, p-value = 3.374e-09

# Verificar normalidad en base al valor p
if (resultado_modelo_1$p.value > 0.03) {
  cat("\nSe tiene normalidad en el Modelo 1 (Sin Interacción).\n")
} else {
  cat("\nNo se tiene normalidad en el Modelo 1 (Sin Interacción).\n")
}

##
## No se tiene normalidad en el Modelo 1 (Sin Interacción).
```

```r
if (resultado_modelo_2$p.value > 0.03) {
  cat("\nSe tiene normalidad en el Modelo 2 (Con Interacción).\n")
} else {
  cat("\nNo se tiene normalidad en el Modelo 2 (Con Interacción).\n")
}
```

```
##
## No se tiene normalidad en el Modelo 2 (Con Interacción).
```

```r
# Gráficos de normalidad y distribuciones para Modelo 1
par(mfrow = c(2, 2))  # Configurar para mostrar 4 gráficos en una
cuadrícula 2x2

# Calcular los límites del eje Y para los histogramas
ylim_1 <- range(density(modelo_1$residuals)$y)
ylim_2 <- range(density(modelo_2$residuals)$y)

# QQ-Plot del Modelo 1
qqnorm(modelo_1$residuals, main = "QQ-Plot: Modelo 1 (Sin Interacción)")
qqline(modelo_1$residuals)

# Histograma del Modelo 1
hist(modelo_1$residuals, freq = FALSE, ylim = ylim_1,
     main = "Histograma: Modelo 1 (Sin Interacción)", xlab = "Residuos")
lines(density(modelo_1$residuals), col = "red")
curve(dnorm(x, mean = mean(modelo_1$residuals), sd =
sd(modelo_1$residuals)),
      from = min(modelo_1$residuals), to = max(modelo_1$residuals), add =
TRUE, col = "blue", lwd = 2)

# QQ-Plot del Modelo 2
qqnorm(modelo_2$residuals, main = "QQ-Plot: Modelo 2 (Con Interacción)")
qqline(modelo_2$residuals)

# Histograma del Modelo 2
hist(modelo_2$residuals, freq = FALSE, ylim = ylim_2,
     main = "Histograma: Modelo 2 (Con Interacción)", xlab = "Residuos")
lines(density(modelo_2$residuals), col = "red")
curve(dnorm(x, mean = mean(modelo_2$residuals), sd =
sd(modelo_2$residuals)),
      from = min(modelo_2$residuals), to = max(modelo_2$residuals), add =
TRUE, col = "blue", lwd = 2)
```





2. Verificación de media cero

% Hipótesis de Media Diferente de Cero (t de Student)

$$\begin{cases} H_0: \mu = 0 & \text{(El promedio de los residuos es igual a cero)} \\ H_1: \mu \neq 0 & \text{(El promedio de los residuos es diferente de cero)} \end{cases}$$

```
# Realiza la prueba t para los residuos de los modelos sin y con
interacción
resultado_t_modelo_1 <- t.test(modelo_1$residuals)
resultado_t_modelo_2 <- t.test(modelo_2$residuals)

# Mostrar los resultados de las pruebas t
cat("Resultados de la prueba t de Student para los residuos:\n")

## Resultados de la prueba t de Student para los residuos:

cat("\nModelo 1 (Sin Interacción):\n")

##
## Modelo 1 (Sin Interacción):

print(resultado_t_modelo_1)

##
##  One Sample t-test
##
## data:  modelo_1$residuals
```

```
## t = 5.6678e-16, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -530.9376  530.9376
## sample estimates:
##    mean of x
## 1.526259e-13
```

```
cat("\nModelo 2 (Con Interacción):\n")
```

```
##
## Modelo 2 (Con Interacción):
```

```
print(resultado_t_modelo_2)
```

```
##
##  One Sample t-test
##
## data:  modelo_2$residuals
## t = -5.7484e-17, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -543.5923  543.5923
## sample estimates:
##     mean of x
## -1.584857e-14
```

```
# Verificar si el promedio de los residuos es significativamente
diferente de cero
if (resultado_t_modelo_1$p.value > 0.04) {
  cat("\nEl promedio de los residuos del Modelo 1 (Sin Interacción) no es
significativamente diferente de cero.\n")
} else {
  cat("\nEl promedio de los residuos del Modelo 1 (Sin Interacción) es
significativamente diferente de cero.\n")
}
```

```
##
## El promedio de los residuos del Modelo 1 (Sin Interacción) no es
significativamente diferente de cero.
```

```
if (resultado_t_modelo_2$p.value > 0.04) {
  cat("\nEl promedio de los residuos del Modelo 2 (Con Interacción) no es
significativamente diferente de cero.\n")
} else {
  cat("\nEl promedio de los residuos del Modelo 2 (Con Interacción) es
significativamente diferente de cero.\n")
}
```

```
##
## El promedio de los residuos del Modelo 2 (Con Interacción) no es
significativamente diferente de cero.
```

3. Homocedasticidad, linealidad e independencia

% Hipótesis de Autocorrelación (Durbin-Watson y Breusch-Godfrey)

$$\begin{cases} H_0\text{:Los errores no están autocorrelacionados (independencia).} \\ H_1\text{:Los errores están autocorrelacionados.} \end{cases}$$

% Hipótesis de Homocedasticidad (Breusch-Pagan y Goldfeld-Quandt)

$$\begin{cases} H_0\text{:La varianza de los errores es constante (homocedasticidad).} \\ H_1\text{:La varianza de los errores no es constante (heterocedasticidad).} \end{cases}$$

```
# Cargar la librería necesaria
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

# Gráficos de residuos contra valores ajustados para ambos modelos
par(mfrow = c(2, 2))  # Configurar para mostrar 4 gráficos en una
cuadrícula 2x2

# Modelo 1: Sin Interacción
plot(modelo_1$fitted.values, modelo_1$residuals,
     main = "Residuos vs. Ajustados: Modelo 1 (Sin Interacción)",
     xlab = "Valores Ajustados", ylab = "Residuos", pch = 19, col =
"blue")
abline(h = 0, col = "red", lwd = 2)

# Modelo 2: Con Interacción
plot(modelo_2$fitted.values, modelo_2$residuals,
     main = "Residuos vs. Ajustados: Modelo 2 (Con Interacción)",
     xlab = "Valores Ajustados", ylab = "Residuos", pch = 19, col =
"blue")
abline(h = 0, col = "red", lwd = 2)

# Pruebas de autocorrelación de errores
cat("Pruebas de autocorrelación de errores:\n")

## Pruebas de autocorrelación de errores:

# Modelo 1: Sin Interacción
dw_modelo_1 <- dwtest(modelo_1)
bg_modelo_1 <- bgtest(modelo_1)

cat("\nModelo 1 (Sin Interacción):\n")
```

```
##
## Modelo 1 (Sin Interacción):
```

```r
# Verificar autocorrelación en base al valor p
if (dw_modelo_1$p.value > 0.04) {
  cat("\nNo se rechaza H₀: Los errores no están autocorrelacionados (Durbin-Watson).\nTiene independencia.\n")
} else {
  cat("\nSe rechaza H₀: Los errores están autocorrelacionados (Durbin-Watson).\nNo tiene independencia.\n")
}
```

```
##
## Se rechaza H₀: Los errores están autocorrelacionados (Durbin-Watson).
## No tiene independencia.
```

```r
if (bg_modelo_1$p.value > 0.04) {
  cat("No se rechaza H₀: Los errores no están autocorrelacionados (Breusch-Godfrey).\nTiene independencia.\n")
} else {
  cat("Se rechaza H₀: Los errores están autocorrelacionados (Breusch-Godfrey).\nNo tiene independencia.\n")
}
```

```
## Se rechaza H₀: Los errores están autocorrelacionados (Breusch-Godfrey).
## No tiene independencia.
```

```r
# Modelo 2: Con Interacción
dw_modelo_2 <- dwtest(modelo_2)
bg_modelo_2 <- bgtest(modelo_2)

cat("\nModelo 2 (Con Interacción):\n")
```

```
##
## Modelo 2 (Con Interacción):
```

```r
# Verificar autocorrelación en base al valor p
if (dw_modelo_2$p.value > 0.04) {
  cat("\nNo se rechaza H₀: Los errores no están autocorrelacionados (Durbin-Watson).\nTiene independencia.\n")
} else {
  cat("\nSe rechaza H₀: Los errores están autocorrelacionados (Durbin-Watson).\nNo tiene independencia.\n")
}
```

```
##
## Se rechaza H₀: Los errores están autocorrelacionados (Durbin-Watson).
## No tiene independencia.
```

```r
if (bg_modelo_2$p.value > 0.04) {
  cat("No se rechaza H₀: Los errores no están autocorrelacionados
```

```r
(Breusch-Godfrey).\nTiene independencia.\n")
} else {
  cat("Se rechaza H₀: Los errores están autocorrelacionados (Breusch-
Godfrey).\nNo tiene independencia.\n")
}
```

```
## Se rechaza H₀: Los errores están autocorrelacionados (Breusch-
Godfrey).
## No tiene independencia.
```

```r
# Pruebas de homocedasticidad
cat("\nPruebas de homocedasticidad:\n")
```

```
##
## Pruebas de homocedasticidad:
```

```r
# Modelo 1: Sin Interacción
bp_modelo_1 <- bptest(modelo_1)
gq_modelo_1 <- gqtest(modelo_1)

cat("\nModelo 1 (Sin Interacción):\n")
```

```
##
## Modelo 1 (Sin Interacción):
```

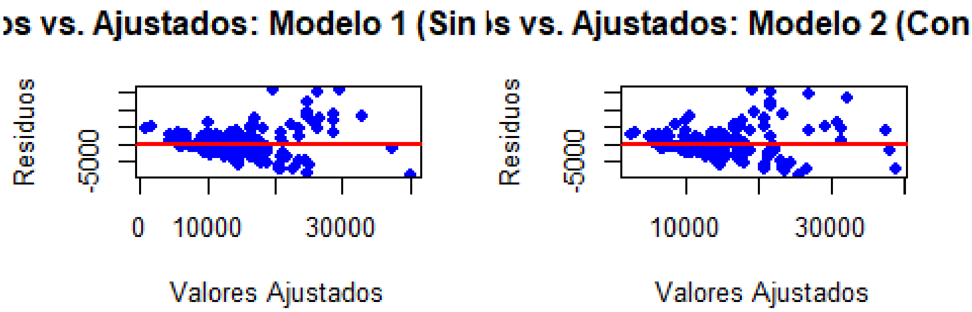
```r
# Verificar homocedasticidad en base al valor p
if (bp_modelo_1$p.value > 0.04) {
  cat("\nNo se rechaza H₀: La varianza de los errores es constante
(Breusch-Pagan).\nTiene homocedasticidad.\n")
} else {
  cat("\nSe rechaza H₀: La varianza de los errores no es constante
(Breusch-Pagan).\nNo tiene homocedasticidad.\n")
}
```

```
##
## Se rechaza H₀: La varianza de los errores no es constante (Breusch-
Pagan).
## No tiene homocedasticidad.
```

```r
if (gq_modelo_1$p.value > 0.04) {
  cat("No se rechaza H₀: La varianza de los errores es constante
(Goldfeld-Quandt).\nTiene homocedasticidad.\n")
} else {
  cat("Se rechaza H₀: La varianza de los errores no es constante
(Goldfeld-Quandt).\nNo tiene homocedasticidad.\n")
}
```

```
## No se rechaza H₀: La varianza de los errores es constante (Goldfeld-
Quandt).
## Tiene homocedasticidad.
```

```r
# Modelo 2: Con Interacción
bp_modelo_2 <- bptest(modelo_2)
gq_modelo_2 <- gqtest(modelo_2)

cat("\nModelo 2 (Con Interacción):\n")

##
## Modelo 2 (Con Interacción):

# Verificar homocedasticidad en base al valor p
if (bp_modelo_2$p.value > 0.04) {
  cat("\nNo se rechaza H₀: La varianza de los errores es constante
(Breusch-Pagan).\nTiene homocedasticidad.\n")
} else {
  cat("\nSe rechaza H₀: La varianza de los errores no es constante
(Breusch-Pagan).\nNo tiene homocedasticidad.\n")
}

##
## Se rechaza H₀: La varianza de los errores no es constante (Breusch-
Pagan).
## No tiene homocedasticidad.

if (gq_modelo_2$p.value > 0.04) {
  cat("No se rechaza H₀: La varianza de los errores es constante
(Goldfeld-Quandt).\nTiene homocedasticidad.\n")
} else {
  cat("Se rechaza H₀: La varianza de los errores no es constante
(Goldfeld-Quandt).\nNo tiene homocedasticidad.\n")
}

## No se rechaza H₀: La varianza de los errores es constante (Goldfeld-
Quandt).
## Tiene homocedasticidad.
```



4. Interpreta cada uno de los analisis que realizaste

Los modelos cumplen con el supuesto de media de residuos igual a cero y en general presentan homocedasticidad, lo cual es positivo. Sin embargo, la falta de normalidad de los residuos y la presencia de autocorrelación son áreas de preocupación. Esto indica que aunque los modelos son útiles, no cumplen con todos los supuestos por lo que los modelos son inválidos a la hora de predecir la variable de precio.

 # Emite una conclusión final sobre el mejor modelo de regresión lineal y contesta la pregunta central:

Concluye sobre el mejor modelo que encontraste y argumenta por qué es el mejor

- **El Mejor Modelo:** El Modelo 1 (Sin Interacción) es el más adecuado para predecir el precio del auto debido a su simplicidad, alta significancia de los coeficientes y un porcentaje mayor de variación explicada sin la complejidad adicional y la falta de significancia observada en el Modelo 2.

¿Cuáles de las variables asignadas influyen en el precio del auto? ¿de qué manera lo hacen?

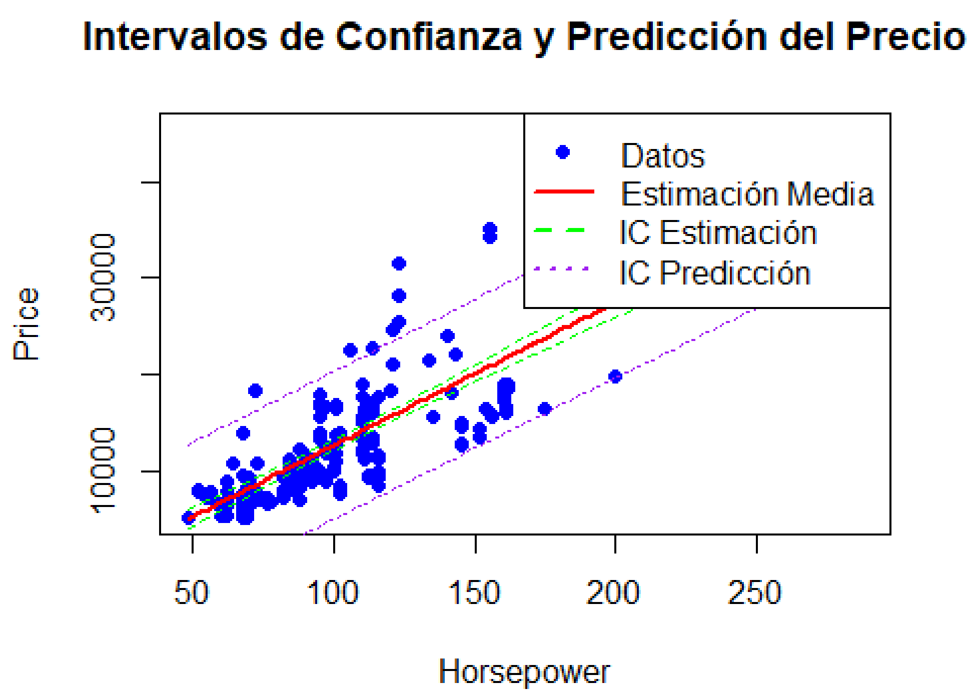
- **Variables que Influyen:** wheelbase, horsepower, y Gas_numeric son variables que influyen significativamente en el precio del auto. wheelbase y horsepower aumentan el precio, mientras que el tipo de combustible tiene un impacto diferencial según la elección entre gasolina y diésel.

 # Intervalos de predicción y confianza

1. Con los datos de las variables asignadas construye la gráfica de los intervalos de confianza y predicción para la estimación y predicción del precio para el mejor modelo seleccionado:

```
# Leer los datos desde el archivo CSV
datos <- read.csv("precios_autos.csv")

# Crear la variable Gas_numeric según tu definición
datos$Gas_numeric <- ifelse(datos$fueltype == 'diesel', 0, 1)

# Ajustar el mejor modelo de regresión (sin interacción)
modelo_1 <- lm(price ~ wheelbase + horsepower + Gas_numeric, data =
datos)

# Crear un rango de valores para horsepower para visualizar los
intervalos
horsepower_vals <- seq(min(datos$horsepower), max(datos$horsepower),
length.out = 100)

# Crear un nuevo DataFrame para predecir usando valores promedio para las
demás variables
new_data <- data.frame(
  wheelbase = mean(datos$wheelbase),  # Usar el promedio de wheelbase
  horsepower = horsepower_vals,       # Secuencia de valores para
horsepower
  Gas_numeric = mean(datos$Gas_numeric)  # Usar el promedio de
Gas_numeric
)

# Obtener las predicciones con intervalos de confianza para la media y
predicción
pred_conf <- predict(modelo_1, newdata = new_data, interval =
"confidence", level = 0.95)
pred_pred <- predict(modelo_1, newdata = new_data, interval =
"prediction", level = 0.95)

# Graficar los puntos observados de horsepower contra price
plot(datos$horsepower, datos$price,
     main = "Intervalos de Confianza y Predicción del Precio",
     xlab = "Horsepower", ylab = "Price",
     pch = 19, col = "blue")

# Añadir la línea de ajuste del modelo
lines(horsepower_vals, pred_conf[, "fit"], col = "red", lwd = 2)

# Añadir los intervalos de confianza (líneas alrededor de la estimación
media)
lines(horsepower_vals, pred_conf[, "lwr"], col = "green", lty = 2)
lines(horsepower_vals, pred_conf[, "upr"], col = "green", lty = 2)
```

```r
# Añadir los intervalos de predicción (líneas más amplias alrededor de
los valores de predicción)
lines(horsepower_vals, pred_pred[, "lwr"], col = "purple", lty = 3)
lines(horsepower_vals, pred_pred[, "upr"], col = "purple", lty = 3)

# Añadir leyenda para identificar las líneas
legend("topright",
       legend = c("Datos", "Estimación Media", "IC Estimación", "IC
Predicción"),
       col = c("blue", "red", "green", "purple"),
       pch = c(19, NA, NA, NA),
       lwd = c(NA, 2, 2, 2),
       lty = c(NA, 1, 2, 3))
```



 # 1. Calcula los intervalos para la variable Y

```r
# Leer los datos desde el archivo CSV
datos <- read.csv("precios_autos.csv")

# Crear la variable Gas_numeric según tu definición
datos$Gas_numeric <- ifelse(datos$fueltype == 'diesel', 0, 1)

# Ajustar el mejor modelo de regresión (sin interacción)
modelo_1 <- lm(price ~ wheelbase + horsepower + Gas_numeric, data =
datos)
```

```r
# Calcular los intervalos de confianza y de predicción para la variable Y
(precio)
intervalos_conf <- predict(modelo_1, newdata = datos, interval =
"confidence", level = 0.95)
intervalos_pred <- predict(modelo_1, newdata = datos, interval =
"prediction", level = 0.95)

# Mostrar los primeros resultados como ejemplo
head(intervalos_conf)

##        fit       lwr      upr
## 1 10223.75  9055.649 11391.84
## 2 10223.75  9055.649 11391.84
## 3 18753.13 17645.921 19860.33
## 4 12973.00 12372.863 13573.13
## 5 14755.34 14174.964 15335.71
## 6 14159.58 13575.384 14743.78

head(intervalos_pred)

##        fit       lwr      upr
## 1 10223.75  2476.113 17971.38
## 2 10223.75  2476.113 17971.38
## 3 18753.13 11014.440 26491.81
## 4 12973.00  5290.450 20655.55
## 5 14755.34  7074.309 22436.37
## 6 14159.58  6478.265 21840.90
```

 # 2. Selecciona la categoría de la variable cualitativa que, de acuerdo a tu análisis resulte la más importante, y separa la base de datos por esa variable categórica.

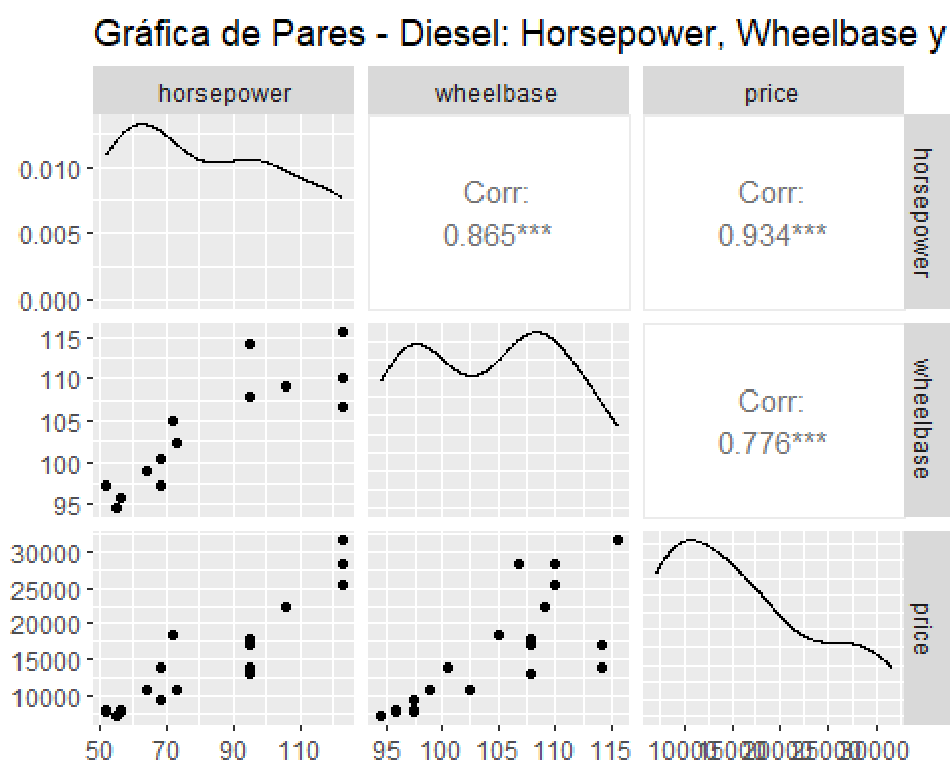
```r
# Leer los datos desde el archivo CSV
datos <- read.csv("precios_autos.csv")

# Crear la variable Gas_numeric según tu definición
datos$Gas_numeric <- ifelse(datos$fueltype == 'diesel', 0, 1)

# Separar la base de datos por la categoría de 'Gas_numeric'
datos_diesel <- subset(datos, Gas_numeric == 0)
datos_gasolina <- subset(datos, Gas_numeric == 1)
```

 # 3. Grafica por pares de variables numéricas

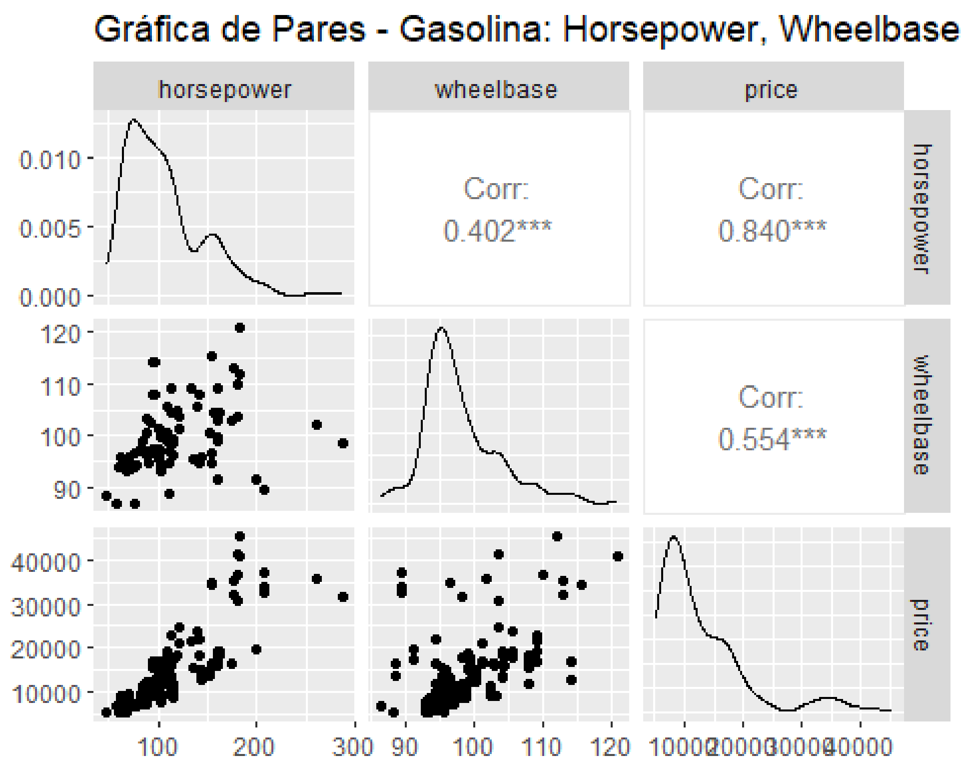
```r
# Cargar la librería necesaria para graficar
library(GGally)

## Warning: package 'GGally' was built under R version 4.3.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
# Filtrar las variables numéricas deseadas (horsepower, wheelbase y
price) en cada subconjunto
variables_diesel <- datos_diesel[, c("horsepower", "wheelbase", "price")]
variables_gasolina <- datos_gasolina[, c("horsepower", "wheelbase",
"price")]

# Gráfica de pares de variables numéricas para los datos de diesel
ggpairs(variables_diesel,
        title = "Gráfica de Pares - Diesel: Horsepower, Wheelbase y
Price")
```




```r
# Gráfica de pares de variables numéricas para los datos de gasolina
ggpairs(variables_gasolina,
        title = "Gráfica de Pares - Gasolina: Horsepower, Wheelbase y
Price")
```



 # 2. Puedes hacer el mismo análisis para otra categoría de la variable cualitativa, pero no es necesario, bastará con que justiques la categoría seleccionada anteriormente.

Solamente hay una variable categórica en el Primer Grupo y ya se realizaron las gráficas para loa valores que puede tomar siendo diesel y gas.

 # 3. Interpreta en el contexto del problema

En autos diesel, tanto la potencia del motor como la distancia entre ejes influyen significativamente en el precio. Esto sugiere que los consumidores asocian mayores valores en estas variables con vehículos de mayor valor y características premium.

En autos de gasolina, la potencia del motor sigue siendo un factor importante para el precio, pero la influencia de la distancia entre ejes es menor en comparación con los autos diesel. Esto podría deberse a que los autos de gasolina suelen tener más variabilidad en el diseño y características.

Los intervalos muestran que el modelo tiene un buen ajuste y permite predecir precios con un rango de variabilidad aceptable. Sin embargo, la amplitud de los intervalos de predicción sugiere que existe una considerable incertidumbre al estimar precios de autos con características similares.

4. Más Allá

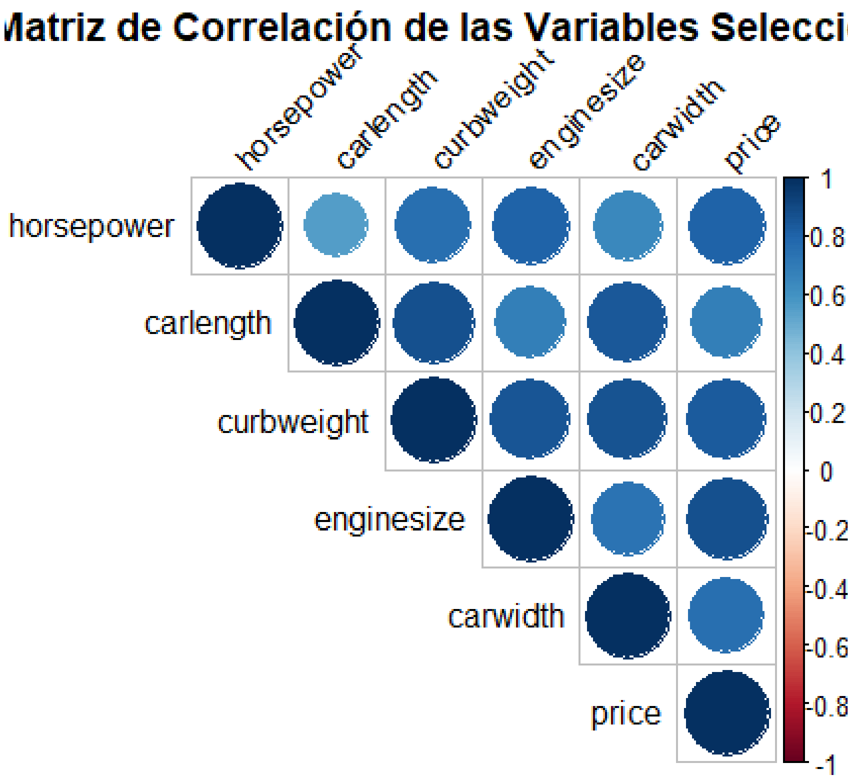
Contesta la pregunta referida a la agrupación de variables que propuso la empresa para el análisis: ¿propondrías una nueva agrupación de las variables a la empresa automovilísitica?

Las cinco variables numéricas que más influyen en el precio del auto son horsepower (potencia del motor), carlength (Longitud del auto), curbweight (peso del auto sin carga), enginesize (tamaño del motor) y carwidth (ancho del auto). Estas variables reflejan aspectos clave del rendimiento, la estabilidad, el tamaño y la percepción de lujo y seguridad del vehículo, factores que los consumidores valoran al considerar el precio. Autos con mayor potencia, dimensiones amplias y motores grandes suelen tener precios más altos debido a sus mejores prestaciones, mayor confort y la calidad percibida de sus materiales y construcción.

Retoma todas las variables y haz un análisis estadístico muy leve (medias y correlación) de cómo crees que se deberían agrupar para analizarlas.

```
# Seleccionar las variables de interés
variables_seleccionadas <- datos[, c("horsepower", "carlength",
"curbweight", "enginesize", "carwidth", "price")]

# Calcular las medias de las variables seleccionadas
medias <- colMeans(variables_seleccionadas, na.rm = TRUE)
print("Medias de las Variables Seleccionadas:")

## [1] "Medias de las Variables Seleccionadas:"

print(medias)

## horsepower  carlength curbweight enginesize   carwidth      price
##   104.1171   174.0493  2555.5659   126.9073    65.9078 13276.7106

# Calcular la matriz de correlación entre las variables seleccionadas
correlacion <- cor(variables_seleccionadas, use = "complete.obs")
print("Matriz de Correlación:")

## [1] "Matriz de Correlación:"

print(correlacion)

##            horsepower carlength curbweight enginesize  carwidth
price
## horsepower  1.0000000 0.5526230  0.7507393  0.8097687 0.6407321
0.8081388
## carlength   0.5526230 1.0000000  0.8777285  0.6833599 0.8411183
0.6829200
## curbweight  0.7507393 0.8777285  1.0000000  0.8505941 0.8670325
0.8353049
## enginesize  0.8097687 0.6833599  0.8505941  1.0000000 0.7354334
0.8741448
## carwidth    0.6407321 0.8411183  0.8670325  0.7354334 1.0000000
0.7593253
```

```
## price      0.8081388 0.6829200  0.8353049  0.8741448 0.7593253
1.0000000

# Visualización de la matriz de correlación
library(corrplot)

## Warning: package 'corrplot' was built under R version 4.3.2

## corrplot 0.92 loaded

corrplot(correlacion, method = "circle", type = "upper", tl.col =
"black", tl.srt = 45,
         title = "Matriz de Correlación de las Variables Seleccionadas",
mar = c(0, 0, 1, 0))
```



Las variables enginesize, curbweight, y horsepower son las más influyentes sobre el precio del auto, reflejando el desempeño y la robustez del vehículo. Estas características, junto con las dimensiones (carwidth y carlength), capturan los aspectos que los consumidores valoran al establecer el precio de un auto, como potencia, confort y estabilidad. Estos hallazgos sugieren que los autos más grandes, potentes y pesados tienden a tener un precio más alto debido a las prestaciones superiores y la percepción de calidad y lujo que ofrecen.