

Actividad Integradora 2

Erika Martínez Meneses

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. Por un análisis de correlación, la empresa automovilística 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 (engine size), carrera o lanzamiento del pistón (stroke) y localización del motor en el carro

Usando el segundo grupo, analiza la significancia de las variables para predecir o influir en la variable precio. ¿propondrías una nueva agrupación a la empresa automovilística?

Lectura de Datos

```
file.choose()
```

```
## [1] "C:\\Users\\erika\\Documents\\Agos-  
Dic2024\\Estadística\\precios_autos.csv"
```

```
library(readr)
```

```
data <- read_csv("C:\\Users\\erika\\Documents\\Agos-  
Dic2024\\Estadística\\precios_autos.csv")
```

```
## Rows: 205 Columns: 21
```

```
## — Column specification
```

```
## Delimiter: ","
```

```
## chr (7): CarName, fueltype, carbody, drivewheel, enginelocation,
enginetype...
## dbl (14): symboling, wheelbase, carlength, carwidth, carheight,
curbweight, ...
##
## i Use `spec()` to retrieve the full column specification for this
data.
## i Specify the column types or set `show_col_types = FALSE` to quiet
this message.
```

1. Exploración de la base de datos

Calcula medidas estadísticas apropiadas para las variables

```
head(data)
```

```
## # A tibble: 6 × 21
##   symboling CarName      fueltype carbody drivewheel enginelocation
wheelbase
##   <dbl> <chr>          <chr>    <chr>    <chr>    <chr>
<dbl>
## 1      3 alfa-romero gi... gas      conver... rwd      front
88.6
## 2      3 alfa-romero st... gas      conver... rwd      front
88.6
## 3      1 alfa-romero Qu... gas      hatchb... rwd      front
94.5
## 4      2 audi 100 ls      gas      sedan    fwd      front
99.8
## 5      2 audi 100ls      gas      sedan    4wd      front
99.4
## 6      2 audi fox        gas      sedan    fwd      front
99.8
## # i 14 more variables: carlength <dbl>, carwidth <dbl>, carheight
<dbl>,
## #   curbweight <dbl>, enginetype <chr>, cylindernumber <chr>,
engine size <dbl>,
## #   stroke <dbl>, compressionratio <dbl>, horsepower <dbl>, peakrpm
<dbl>,
## #   citympg <dbl>, highwaympg <dbl>, price <dbl>
```

Cuantitativas (media, desviación estándar, cuantiles, etc)

```
summary(data$carwidth)
```

```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   60.30  64.10   65.50   65.91  66.90   72.30
```

```
summary(data$carheight)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    47.80  52.00   54.10   53.72  55.50   59.80
```

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

```
table(data$carbody)
```

```
##
## convertible      hardtop   hatchback      sedan      wagon
##           6           8          70          96          25
```

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

Transformación de la columna a variable binaria

```
data$convertible <- ifelse(data$carbody == "convertible", 1, 0)
data$carbody <- as.factor(data$carbody)
```

Correlación

```
cor(data[, c("carheight", "carwidth", "price", "convertible")], use =
"complete.obs")

##           carheight   carwidth    price convertible
## carheight    1.0000000  0.27921032 0.1193362  -0.16323866
## carwidth      0.2792103  1.00000000 0.7593253  -0.02632807
## price         0.1193362  0.75932530 1.0000000   0.18768121
## convertible  -0.1632387 -0.02632807 0.1876812   1.00000000
```

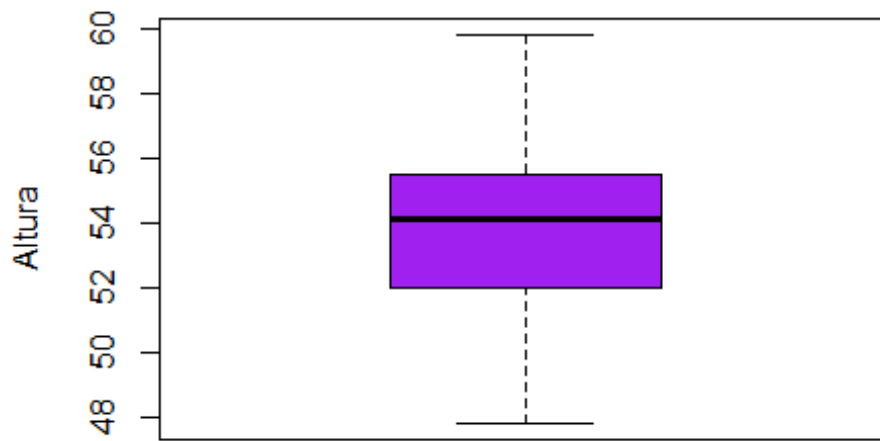
Explora los datos usando herramientas de visualización (si lo consideras necesario):

Variables cuantitativas:

Boxplot (visualización de datos atípicos)

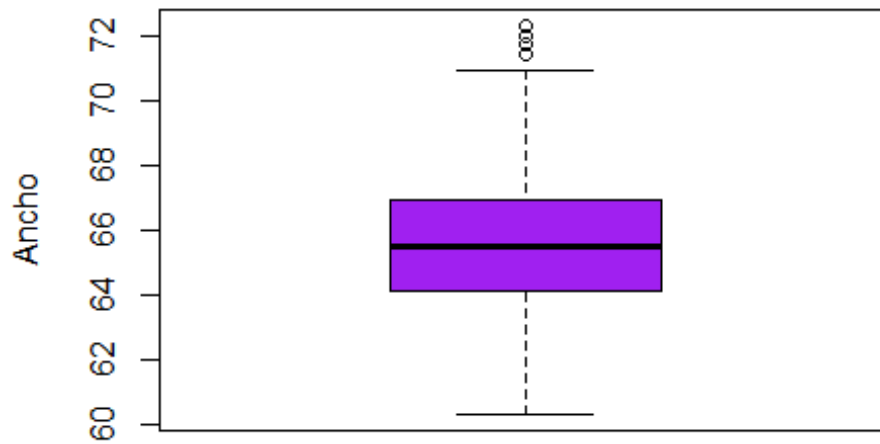
```
boxplot(data$carheight, main = "Altura del Auto", ylab = "Altura", col =
"purple")
```

Altura del Auto



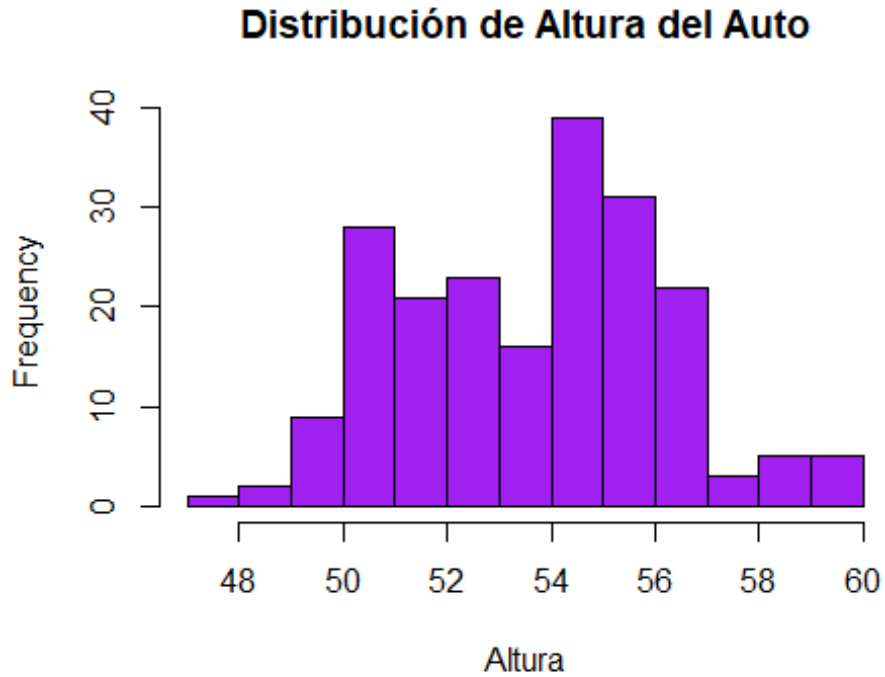
```
boxplot(data$carwidth, main = "Ancho del Auto", ylab = "Ancho", col = "purple")
```

Ancho del Auto



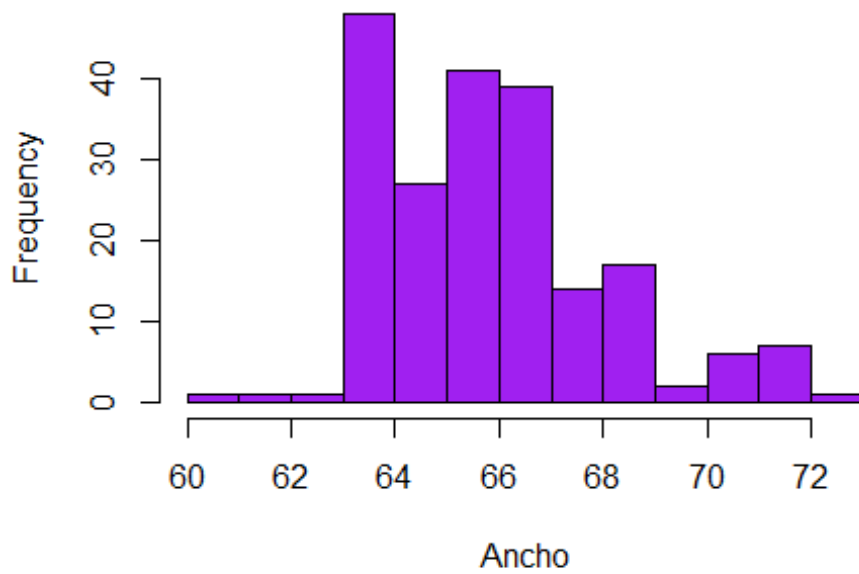
Histogramas

```
hist(data$carheight, main = "Distribución de Altura del Auto", xlab =  
"Altura", col = "purple")
```



```
hist(data$carwidth, main = "Distribución de Ancho del Auto", xlab =  
"Ancho", col = "purple")
```

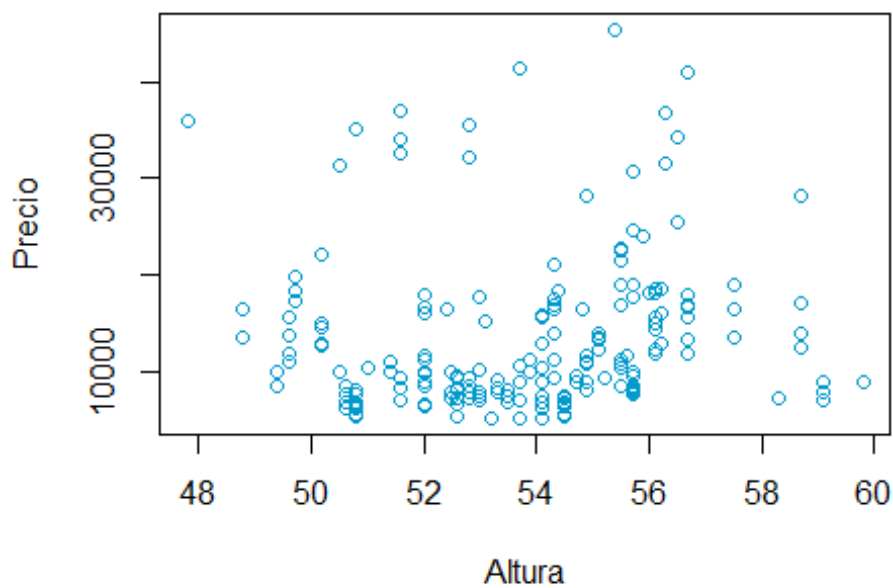
Distribución de Ancho del Auto



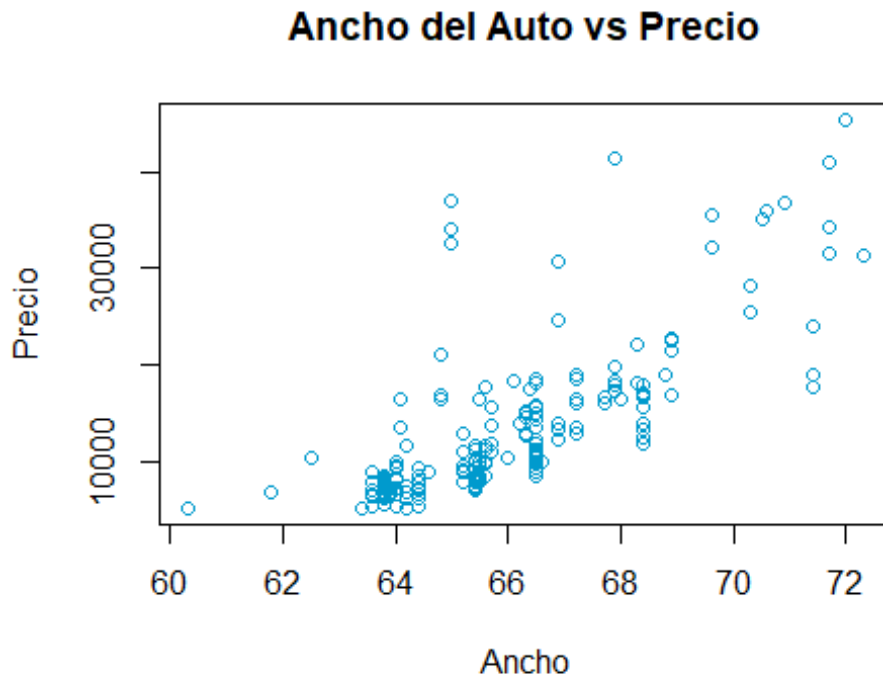
Diagramas de dispersión y correlación por pares

```
plot(data$carheight, data$price, main = "Altura del Auto vs Precio", xlab = "Altura", ylab = "Precio", col="deepskyblue3")
```

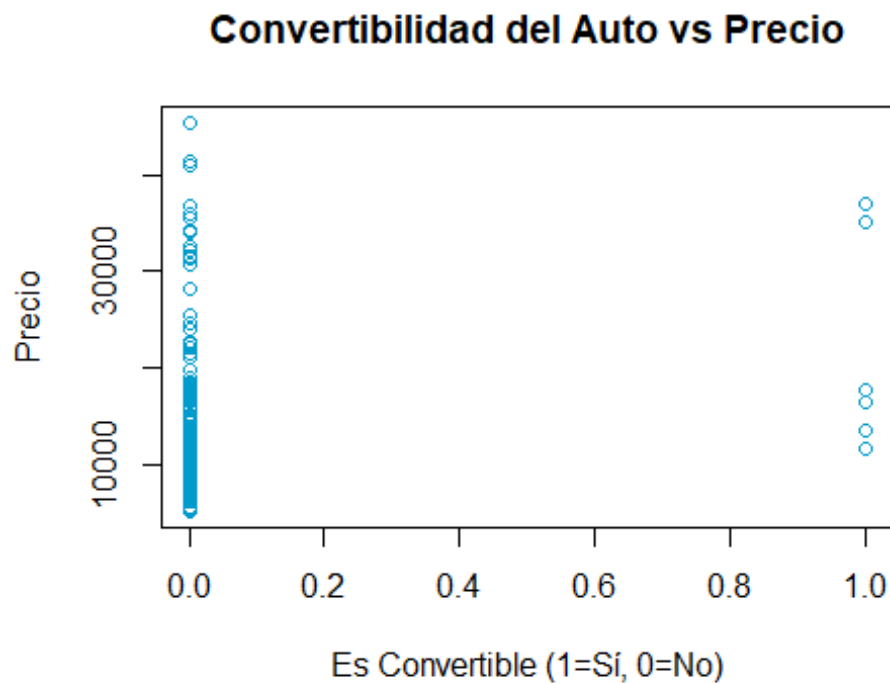
Altura del Auto vs Precio



```
plot(data$carwidth, data$price, main = "Ancho del Auto vs Precio", xlab =  
"Ancho", ylab = "Precio", col="deepskyblue3")
```



```
plot(data$convertible, data$price, main = "Convertibilidad del Auto vs  
Precio", xlab = "Es Convertible (1=Sí, 0=No)", ylab =  
"Precio", col="deepskyblue3")
```



Variables categóricas

Distribución de los datos (diagramas de barras, diagramas de pastel)

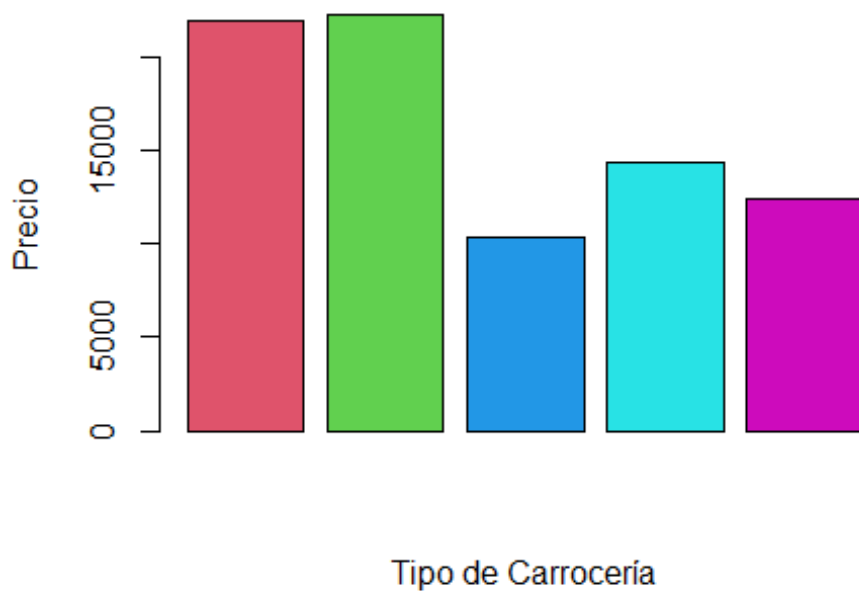
```
avg_price <- aggregate(price ~ carbody, data = data, mean)
pie(avg_price$price, labels = avg_price$carbody, col =
rainbow(length(avg_price$carbody)), main = "Precio Promedio por Tipo de
Carrocería")
```


Precio Promedio por Tipo de Carrocería



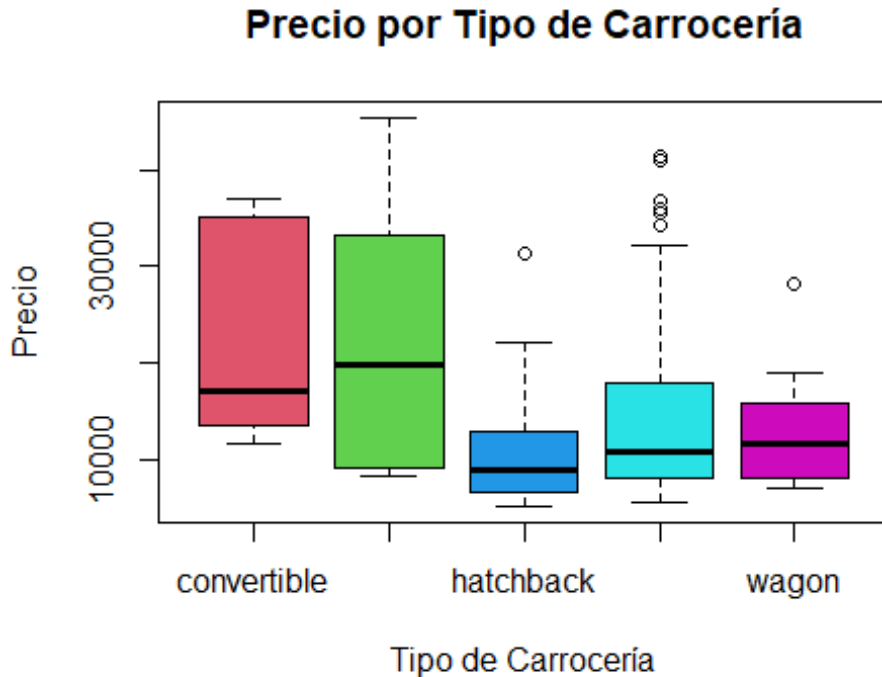
```
barplot(avg_price$price, col = 2:6, main = "Precio Promedio por Tipo de Carrocería", xlab = "Tipo de Carrocería", ylab = "Precio")
```

Precio Promedio por Tipo de Carrocería



Boxplot por categoría de las variables cuantitativas

```
boxplot(price ~ carbody, data = data, col = 2:6, main = "Precio por Tipo de Carrocería", xlab = "Tipo de Carrocería", ylab = "Precio")
```



2. Modelación y verificación del modelo

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.

Para cada uno de los modelos propuestos:

Realiza la regresión entre las variables involucradas

```
# Regresión Lineal Simple
modelo1 <- lm(price ~ carheight + carwidth + convertible, data = data)
summary(modelo1)

##
## Call:
## lm(formula = price ~ carheight + carwidth + convertible, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10880.4  -2612.0   -956.7   1065.8  23205.9
##
## Coefficients:
```

```

##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -167451.0    11724.6 -14.282  < 2e-16 ***
## carheight   -219.5      149.3   -1.471    0.143
## carwidth     2916.9      167.8   17.381  < 2e-16 ***
## convertible  9330.3      2073.7    4.499 1.15e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4937 on 201 degrees of freedom
## Multiple R-squared:  0.6238, Adjusted R-squared:  0.6182
## F-statistic: 111.1 on 3 and 201 DF,  p-value: < 2.2e-16

# Regresión Lineal Multiplicativa con Interacciones
modelo2 <- lm(price ~ carheight * carwidth * convertible, data = data)
summary(modelo2)

##
## Call:
## lm(formula = price ~ carheight * carwidth * convertible, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10501.5  -2659.6   -909.6    857.1   23325.8
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -404340.68   243496.69  -1.661    0.0984
##
## carheight                      4149.84     4484.67    0.925    0.3559
## carwidth                       6490.59     3679.54    1.764    0.0793
##
## convertible                   -529478.70  6284532.55  -0.084    0.9329
## carheight:carwidth              -65.88        67.71  -0.973    0.3317
## carheight:convertible           10911.09   123780.55    0.088    0.9298
## carwidth:convertible             8682.68    97730.66    0.089    0.9293
## carheight:carwidth:convertible   -175.46     1924.54  -0.091    0.9275
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4973 on 197 degrees of freedom
## Multiple R-squared:  0.6258, Adjusted R-squared:  0.6125
## F-statistic: 47.06 on 7 and 197 DF,  p-value: < 2.2e-16

# Regresión Lineal Multiplicativa con Interacciones
modelo3 <- lm(price ~ carwidth + convertible, data = data)
summary(modelo3)

##
## Call:
## lm(formula = price ~ carwidth + convertible, data = data)
##

```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10921.7  -2505.8   -756.8    971.2   23624.4
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -174725.7    10660.4  -16.390  < 2e-16 ***
## carwidth     2848.1      161.6   17.621  < 2e-16 ***
## convertible  9825.5      2052.1    4.788 3.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4951 on 202 degrees of freedom
## Multiple R-squared:  0.6197, Adjusted R-squared:  0.616
## F-statistic: 164.6 on 2 and 202 DF,  p-value: < 2.2e-16

# Regresión Lineal Multiplicativa con Interacciones
modelo4 <- lm(price ~ convertible, data = data)
summary(modelo4)

##
## Call:
## lm(formula = price ~ convertible, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10295  -5279  -3028   3483   32383
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13017.0      557.6   23.344  < 2e-16 ***
## convertible   8873.5      3259.4    2.722  0.00704 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7866 on 203 degrees of freedom
## Multiple R-squared:  0.03522, Adjusted R-squared:  0.03047
## F-statistic: 7.412 on 1 and 203 DF,  p-value: 0.007044

# Regresión Lineal Multiplicativa con Interacciones
modelo5 <- lm(price ~ carwidth, data = data)
summary(modelo5)

##
## Call:
## lm(formula = price ~ carwidth, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11097.4  -2690.0   -857.3    798.7   26318.4
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -173095.2    11215.6  -15.43  <2e-16 ***
## carwidth      2827.8      170.1   16.63  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5211 on 203 degrees of freedom
## Multiple R-squared:  0.5766, Adjusted R-squared:  0.5745
## F-statistic: 276.4 on 1 and 203 DF,  p-value: < 2.2e-16

# Regresión Lineal Multiplicativa con Interacciones
modelo6 <- lm(price ~ carheight, data = data)
summary(modelo6)

##
## Call:
## lm(formula = price ~ carheight, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8456  -5466  -2778   2876  31470
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7684.5    12252.6  -0.627   0.5313
## carheight      390.2      227.8   1.713   0.0883 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7951 on 203 degrees of freedom
## Multiple R-squared:  0.01424, Adjusted R-squared:  0.009385
## F-statistic: 2.933 on 1 and 203 DF,  p-value: 0.08833

# Regresión Lineal Multiplicativa con Interacciones
modelo7 <- lm(price ~ carheight + carwidth, data = data)
summary(modelo7)

##
## Call:
## lm(formula = price ~ carheight + carwidth, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11022  -2951  -1196   1156  25715
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -162328.7    12212.4 -13.292  <2e-16 ***
## carheight     -328.6      154.2  -2.132   0.0342 *
## carwidth      2932.3      175.6  16.699  <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5166 on 202 degrees of freedom
## Multiple R-squared:  0.5859, Adjusted R-squared:  0.5818
## F-statistic: 142.9 on 2 and 202 DF,  p-value: < 2.2e-16

# Regresión Lineal Multiplicativa con Interacciones
modelo8 <- lm(price ~ carheight + convertible, data = data)
summary(modelo8)

##
## Call:
## lm(formula = price ~ carheight + convertible, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12394  -5132  -2633   2969  31574
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -14081.5    12183.9  -1.156  0.24915
## carheight      503.7      226.3   2.226  0.02709 *
## convertible  10062.7     3272.0   3.075  0.00239 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7791 on 202 degrees of freedom
## Multiple R-squared:  0.05833, Adjusted R-squared:  0.04901
## F-statistic: 6.256 on 2 and 202 DF,  p-value: 0.00231

# Regresión Lineal Multiplicativa con Interacciones
modelo9 <- lm(price ~ carheight * carwidth, data = data)
summary(modelo9)

##
## Call:
## lm(formula = price ~ carheight * carwidth, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10642.6  -2971.3  -1138.7    852.8  25801.8
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -376230.07  241981.51  -1.555  0.1216
## carheight      3621.09   4465.16   0.811  0.4183
## carwidth      6166.03   3657.83   1.686  0.0934 .
## carheight:carwidth    -59.69     67.43  -0.885  0.3772
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 5169 on 201 degrees of freedom
## Multiple R-squared:  0.5875, Adjusted R-squared:  0.5813
## F-statistic: 95.42 on 3 and 201 DF,  p-value: < 2.2e-16

# Regresión Lineal Multiplicativa con Interacciones
modelo10 <- lm(price ~ carheight * convertible, data = data)
summary(modelo10)

##
## Call:
## lm(formula = price ~ carheight * convertible, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9864  -5079  -2672   2915  31524
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -15773.0    12373.4  -1.275   0.2039
## carheight       535.2      229.8    2.329   0.0208 *
## convertible   66345.9    69842.8    0.950   0.3433
## carheight:convertible -1092.9    1354.6  -0.807   0.4208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7797 on 201 degrees of freedom
## Multiple R-squared:  0.06137,    Adjusted R-squared:  0.04736
## F-statistic: 4.381 on 3 and 201 DF,  p-value: 0.005181

# Regresión Lineal Multiplicativa con Interacciones
modelo11 <- lm(price ~ carwidth * convertible, data = data)
summary(modelo11)

##
## Call:
## lm(formula = price ~ carwidth * convertible, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10905.3  -2510.2   -763.1    964.9   23621.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -174528.30   10866.42  -16.061  <2e-16 ***
## carwidth       2845.15     164.76   17.268  <2e-16 ***
## convertible   3846.64    59654.08    0.064    0.949
## carwidth:convertible    91.15     908.90    0.100    0.920
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4963 on 201 degrees of freedom
## Multiple R-squared:  0.6198, Adjusted R-squared:  0.6141
## F-statistic: 109.2 on 3 and 201 DF,  p-value: < 2.2e-16
```

Significancia del modelo

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

- $H_0: \beta_1 = 0$
- $H_1: \beta_1 \neq 0$

```
if (summary(modelo1)$fstatistic[1] > qf(1-0.04, df1 =
summary(modelo1)$fstatistic[2], df2 = summary(modelo1)$fstatistic[3])) {
  print("El modelo 1 es significativo con alfa de 0.04.")
} else {
  print("El modelo 1 no es significativo con alfa de 0.04.")
}

## [1] "El modelo 1 es significativo con alfa de 0.04."

if (summary(modelo2)$fstatistic[1] > qf(1-0.04, df1 =
summary(modelo2)$fstatistic[2], df2 = summary(modelo2)$fstatistic[3])) {
  print("El modelo 2 es significativo con alfa de 0.04.")
} else {
  print("El modelo 2 no es significativo con alfa de 0.04.")
}

## [1] "El modelo 2 es significativo con alfa de 0.04."

if (summary(modelo3)$fstatistic[1] > qf(1-0.04, df1 =
summary(modelo3)$fstatistic[2], df2 = summary(modelo3)$fstatistic[3])) {
  print("El modelo 3 es significativo con alfa de 0.04.")
} else {
  print("El modelo 3 no es significativo con alfa de 0.04.")
}

## [1] "El modelo 3 es significativo con alfa de 0.04."

if (summary(modelo4)$fstatistic[1] > qf(1-0.04, df1 =
summary(modelo4)$fstatistic[2], df2 = summary(modelo4)$fstatistic[3])) {
  print("El modelo 4 es significativo con alfa de 0.04.")
} else {
  print("El modelo 4 no es significativo con alfa de 0.04.")
}

## [1] "El modelo 4 es significativo con alfa de 0.04."

if (summary(modelo5)$fstatistic[1] > qf(1-0.04, df1 =
summary(modelo5)$fstatistic[2], df2 = summary(modelo5)$fstatistic[3])) {
  print("El modelo 5 es significativo con alfa de 0.04.")
} else {
```



```

    print("El modelo 5 no es significativo con alfa de 0.04.")
}

## [1] "El modelo 5 es significativo con alfa de 0.04."

if (summary(modelo6)$fstatistic[1] > qf(1-0.04, df1 =
summary(modelo6)$fstatistic[2], df2 = summary(modelo6)$fstatistic[3])) {
    print("El modelo 6 es significativo con alfa de 0.04.")
} else {
    print("El modelo 6 no es significativo con alfa de 0.04.")
}

## [1] "El modelo 6 no es significativo con alfa de 0.04."

if (summary(modelo7)$fstatistic[1] > qf(1-0.04, df1 =
summary(modelo7)$fstatistic[2], df2 = summary(modelo7)$fstatistic[3])) {
    print("El modelo 7 es significativo con alfa de 0.04.")
} else {
    print("El modelo 7 no es significativo con alfa de 0.04.")
}

## [1] "El modelo 7 es significativo con alfa de 0.04."

if (summary(modelo8)$fstatistic[1] > qf(1-0.04, df1 =
summary(modelo8)$fstatistic[2], df2 = summary(modelo8)$fstatistic[3])) {
    print("El modelo 8 es significativo con alfa de 0.04.")
} else {
    print("El modelo 8 no es significativo con alfa de 0.04.")
}

## [1] "El modelo 8 es significativo con alfa de 0.04."

if (summary(modelo9)$fstatistic[1] > qf(1-0.04, df1 =
summary(modelo9)$fstatistic[2], df2 = summary(modelo9)$fstatistic[3])) {
    print("El modelo 9 es significativo con alfa de 0.04.")
} else {
    print("El modelo 9 no es significativo con alfa de 0.04.")
}

## [1] "El modelo 9 es significativo con alfa de 0.04."

if (summary(modelo10)$fstatistic[1] > qf(1-0.04, df1 =
summary(modelo10)$fstatistic[2], df2 = summary(modelo10)$fstatistic[3]))
{
    print("El modelo 10 es significativo con alfa de 0.04.")
} else {
    print("El modelo 10 no es significativo con alfa de 0.04.")
}

## [1] "El modelo 10 es significativo con alfa de 0.04."

if (summary(modelo11)$fstatistic[1] > qf(1-0.04, df1 =
summary(modelo11)$fstatistic[2], df2 = summary(modelo11)$fstatistic[3]))

```

```
{
  print("El modelo 11 es significativo con alfa de 0.04.")
} else {
  print("El modelo 11 no es significativo con alfa de 0.04.")
}
```

```
## [1] "El modelo 11 es significativo con alfa de 0.04."
```

Valida la significancia \hat{B}_i con un alfa de 0.04 (incluye las hipótesis que pruebas y el valor frontera de cada una de ellas)

Hipótesis * $H_0: \beta_i = 0$ * $H_1: \exists \beta_i \neq 0$

```
coef(summary(modelo1))[, 4] < 0.04 # Devuelve TRUE si Los coeficientes
son significativos
```

```
## (Intercept)    carheight    carwidth convertible
##           TRUE          FALSE          TRUE          TRUE
```

```
coef(summary(modelo2))[, 4] < 0.04 # Devuelve TRUE si Los coeficientes
son significativos
```

```
##                (Intercept)                carheight
##                FALSE                FALSE
##                carwidth                convertible
##                FALSE                FALSE
##                carheight:carwidth    carheight:convertible
##                FALSE                FALSE
##                carwidth:convertible  carheight:carwidth:convertible
##                FALSE                FALSE
```

```
coef(summary(modelo3))[, 4] < 0.04 # Devuelve TRUE si Los coeficientes
son significativos
```

```
## (Intercept)    carwidth convertible
##           TRUE          TRUE          TRUE
```

```
coef(summary(modelo4))[, 4] < 0.04 # Devuelve TRUE si Los coeficientes
son significativos
```

```
## (Intercept) convertible
##           TRUE          TRUE
```

```
coef(summary(modelo5))[, 4] < 0.04 # Devuelve TRUE si Los coeficientes
son significativos
```

```
## (Intercept)    carwidth
##           TRUE          TRUE
```

```
coef(summary(modelo6))[, 4] < 0.04 # Devuelve TRUE si Los coeficientes
son significativos
```

```
## (Intercept)    carheight
##             FALSE      FALSE

coef(summary(modelo7))[, 4] < 0.04 # Devuelve TRUE si Los coeficientes
son significativos

## (Intercept)    carheight    carwidth
##             TRUE      TRUE      TRUE

coef(summary(modelo8))[, 4] < 0.04 # Devuelve TRUE si Los coeficientes
son significativos

## (Intercept)    carheight convertible
##             FALSE      TRUE      TRUE

coef(summary(modelo9))[, 4] < 0.04 # Devuelve TRUE si Los coeficientes
son significativos

##             (Intercept)            carheight            carwidth
carheight:carwidth
##             FALSE            FALSE            FALSE
FALSE

coef(summary(modelo10))[, 4] < 0.04 # Devuelve TRUE si Los coeficientes
son significativos

##             (Intercept)            carheight            convertible
##             FALSE            TRUE            FALSE
## carheight:convertible
##             FALSE

coef(summary(modelo11))[, 4] < 0.04 # Devuelve TRUE si Los coeficientes
son significativos

##             (Intercept)            carwidth            convertible
##             TRUE            TRUE            FALSE
## carwidth:convertible
##             FALSE
```

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

```
paste("El modelo 1 explica el", round(summary(modelo1)$r.squared * 100,
2), "% de la variabilidad del precio")

## [1] "El modelo 1 explica el 62.38 % de la variabilidad del precio"

paste("El modelo 2 explica el", round(summary(modelo2)$r.squared * 100,
2), "% de la variabilidad del precio")

## [1] "El modelo 2 explica el 62.58 % de la variabilidad del precio"

paste("El modelo 3 explica el", round(summary(modelo3)$r.squared * 100,
2), "% de la variabilidad del precio")
```

```
## [1] "El modelo 3 explica el 61.97 % de la variabilidad del precio"
paste("El modelo 4 explica el", round(summary(modelo4)$r.squared * 100,
2), "% de la variabilidad del precio")

## [1] "El modelo 4 explica el 3.52 % de la variabilidad del precio"
paste("El modelo 5 explica el", round(summary(modelo5)$r.squared * 100,
2), "% de la variabilidad del precio")

## [1] "El modelo 5 explica el 57.66 % de la variabilidad del precio"
paste("El modelo 6 explica el", round(summary(modelo6)$r.squared * 100,
2), "% de la variabilidad del precio")

## [1] "El modelo 6 explica el 1.42 % de la variabilidad del precio"
paste("El modelo 7 explica el", round(summary(modelo7)$r.squared * 100,
2), "% de la variabilidad del precio")

## [1] "El modelo 7 explica el 58.59 % de la variabilidad del precio"
paste("El modelo 8 explica el", round(summary(modelo8)$r.squared * 100,
2), "% de la variabilidad del precio")

## [1] "El modelo 8 explica el 5.83 % de la variabilidad del precio"
paste("El modelo 9 explica el", round(summary(modelo9)$r.squared * 100,
2), "% de la variabilidad del precio")

## [1] "El modelo 9 explica el 58.75 % de la variabilidad del precio"
paste("El modelo 10 explica el", round(summary(modelo10)$r.squared * 100,
2), "% de la variabilidad del precio")

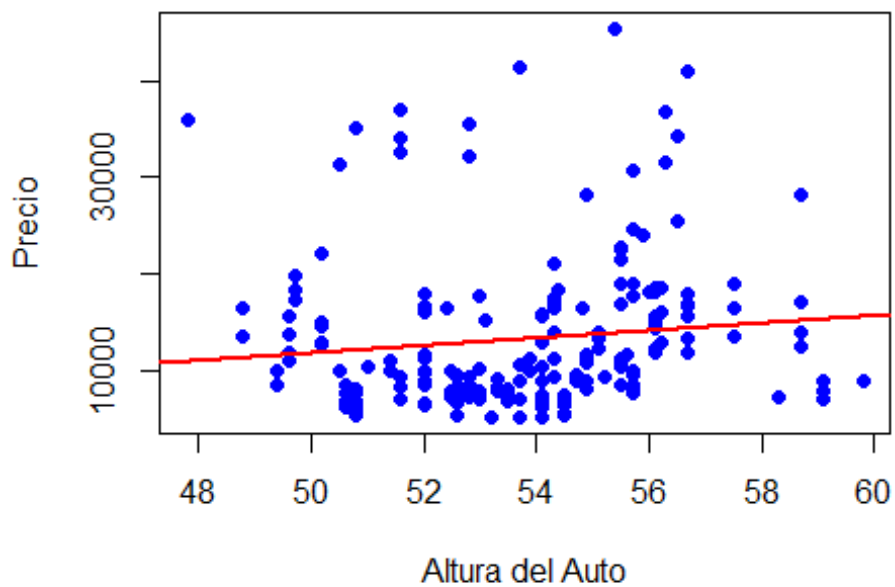
## [1] "El modelo 10 explica el 6.14 % de la variabilidad del precio"
paste("El modelo 11 explica el", round(summary(modelo11)$r.squared * 100,
2), "% de la variabilidad del precio")

## [1] "El modelo 11 explica el 61.98 % de la variabilidad del precio"
```

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

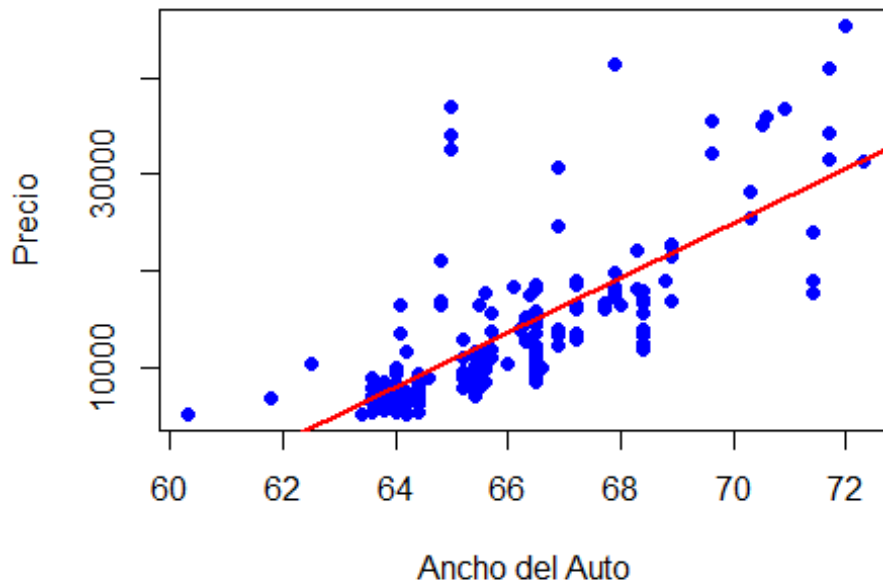
```
plot(data$carheight, data$price,
     main = "Diagrama de Dispersión: Altura del Auto vs Precio",
     xlab = "Altura del Auto", ylab = "Precio",
     pch = 19, col = "blue")
abline(lm(price ~ carheight, data = data), col = "red", lwd = 2)
```

Diagrama de Dispersión: Altura del Auto vs Precio



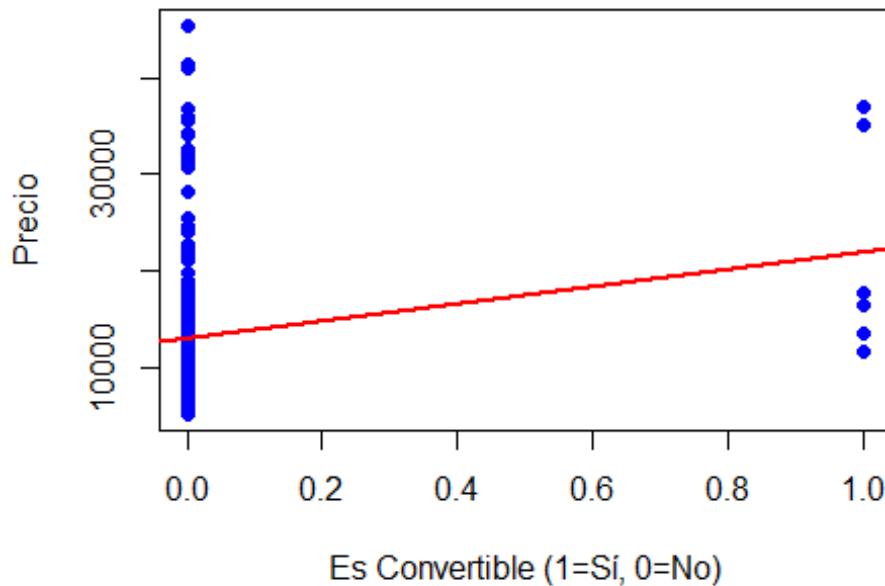
```
plot(data$carwidth, data$price,  
     main = "Diagrama de Dispersión: Ancho del Auto vs Precio",  
     xlab = "Ancho del Auto", ylab = "Precio",  
     pch = 19, col = "blue")  
abline(lm(price ~ carwidth, data = data), col = "red", lwd = 2)
```

Diagrama de Dispersión: Ancho del Auto vs Preci



```
plot(data$convertible, data$price,  
     main = "Diagrama de Dispersión: Convertibilidad del Auto vs Precio",  
     xlab = "Es Convertible (1=Sí, 0=No)", ylab = "Precio",  
     pch = 19, col = "blue")  
abline(lm(price ~ convertible, data = data), col = "red", lwd = 2)
```

Diagrama de Dispersión: Convertibilidad del Auto vs Precio



Analiza la validez de los modelos propuestos:

Normalidad de los residuos

Prueba de Hipótesis

- H_0 = La muestra proviene de una distribución normal
- H_1 = La muestra no proviene de una distribución normal

Regla de decisión: Se rechaza H_0 si valor $p < \alpha$

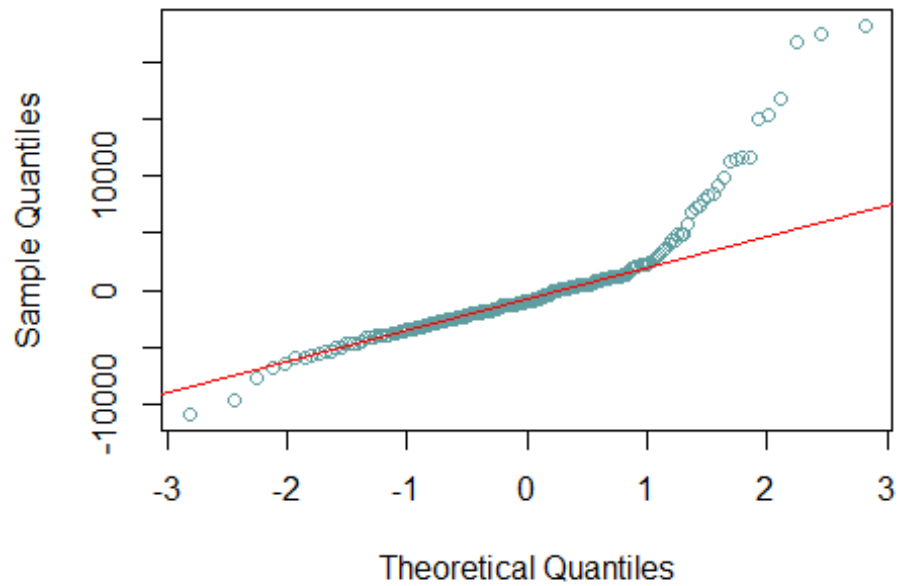
Modelo 1

```
library(nortest)
ad.test(modelo1$residuals)

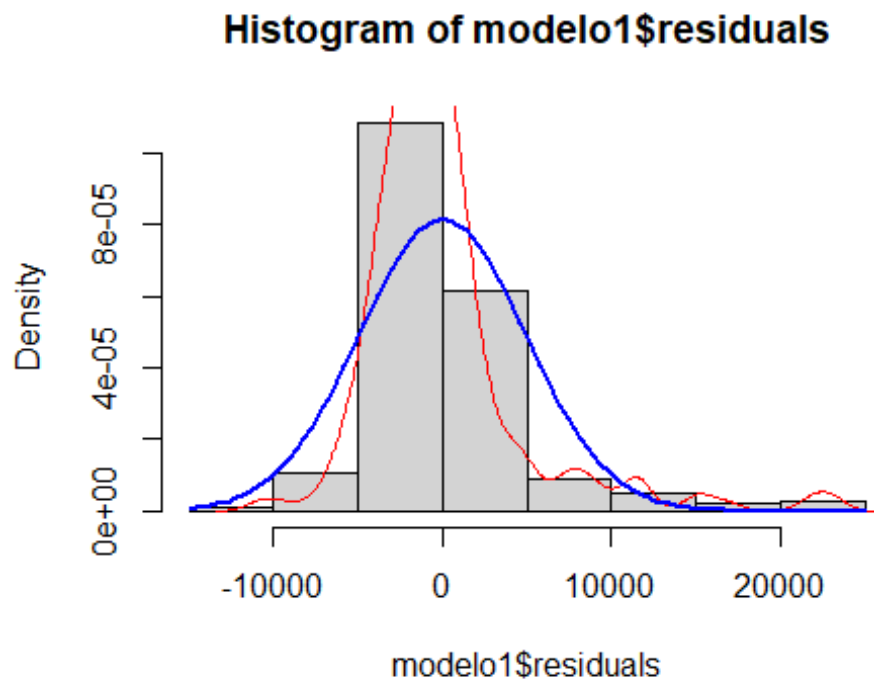
##
## Anderson-Darling normality test
##
## data: modelo1$residuals
## A = 10.657, p-value < 2.2e-16

qqnorm(modelo1$residuals, col = "#5F9EA0")
qqline(modelo1$residuals, col = "red")
```

Normal Q-Q Plot



```
hist(modelo1$residuals,freq=FALSE)
lines(density(modelo1$residual),col="red")
curve(dnorm(x,mean=mean(modelo1$residuals),sd=sd(modelo1$residuals)),
add=TRUE, col="blue",lwd=2)
```

Rechazamos

H_0 ya que el valor $p < \alpha = 0.04$ por lo que podemos decir que nuestros datos no provienen de una distribución normal

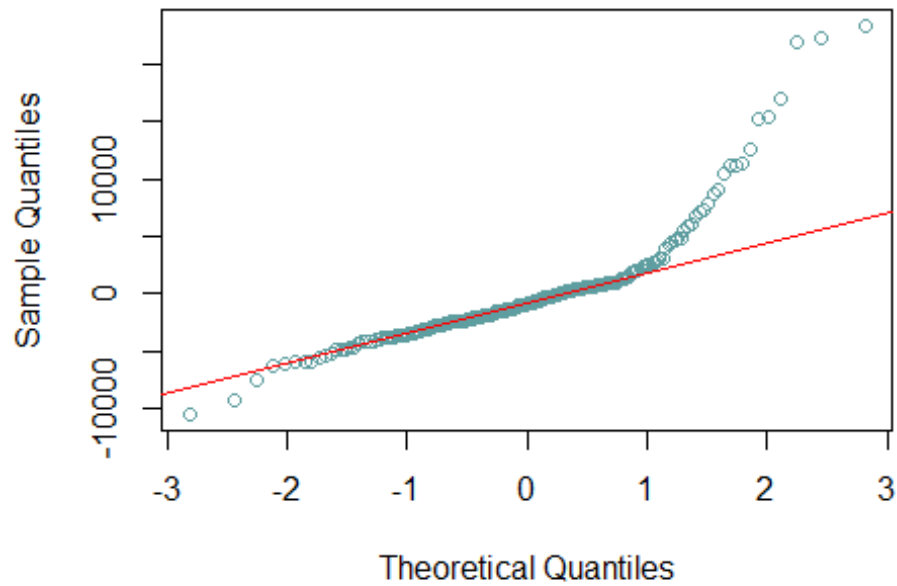
Modelo 2

```
library(nortest)
ad.test(modelo2$residuals)

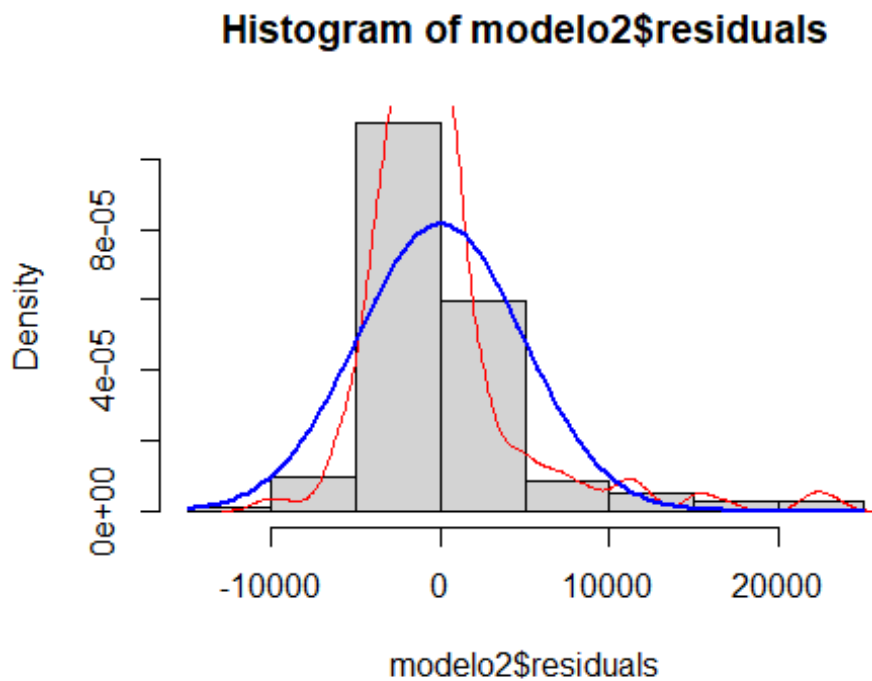
##
## Anderson-Darling normality test
##
## data: modelo2$residuals
## A = 10.77, p-value < 2.2e-16

qqnorm(modelo2$residuals, col = "#5F9EA0")
qqline(modelo2$residuals, col = "red")
```

Normal Q-Q Plot



```
hist(modelo2$residuals,freq=FALSE)
lines(density(modelo2$residual),col="red")
curve(dnorm(x,mean=mean(modelo2$residuals),sd=sd(modelo2$residuals)),
add=TRUE, col="blue",lwd=2)
```



Rechazamos

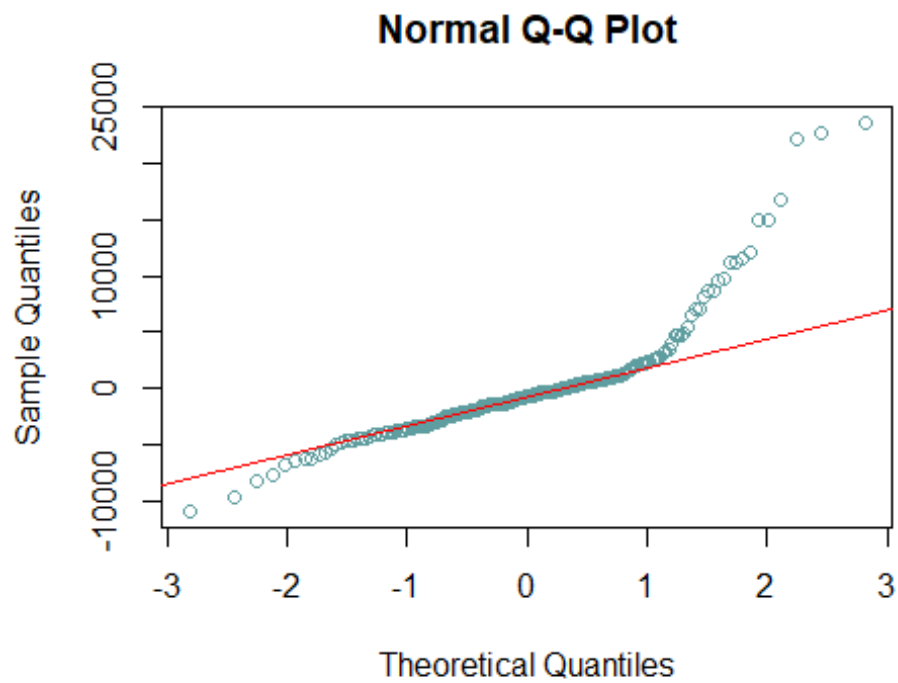
H_0 ya que el valor $p < \alpha = 0.04$ por lo que podemos decir que nuestros datos no provienen de una distribución normal

Modelo 3

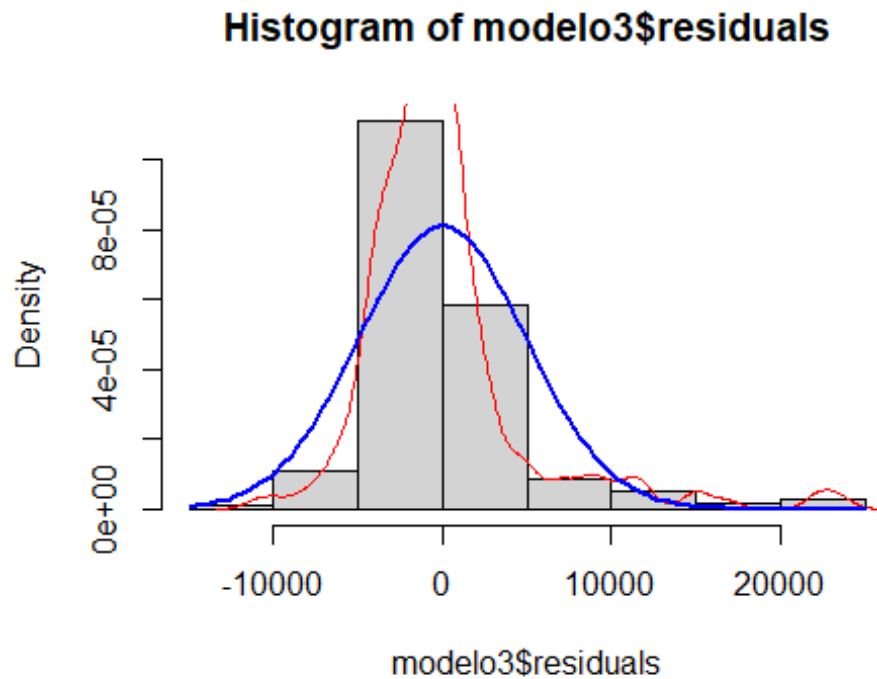
```
library(nortest)
ad.test(modelo3$residuals)

##
## Anderson-Darling normality test
##
## data: modelo3$residuals
## A = 10.721, p-value < 2.2e-16

qqnorm(modelo3$residuals, col = "#5F9EA0")
qqline(modelo3$residuals, col = "red")
```



```
hist(modelo3$residuals,freq=FALSE)  
lines(density(modelo3$residual),col="red")  
curve(dnorm(x,mean=mean(modelo3$residuals),sd=sd(modelo3$residuals)),  
add=TRUE, col="blue",lwd=2)
```



Rechazamos

H_0 ya que el valor $p < \alpha = 0.04$ por lo que podemos decir que nuestros datos no provienen de una distribución normal

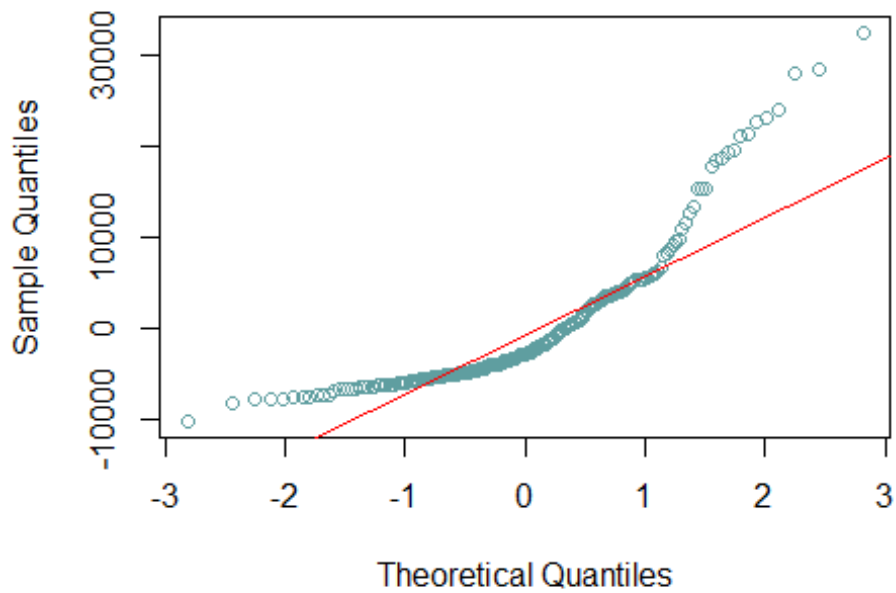
Modelo 4

```
library(nortest)
ad.test(modelo4$residuals)

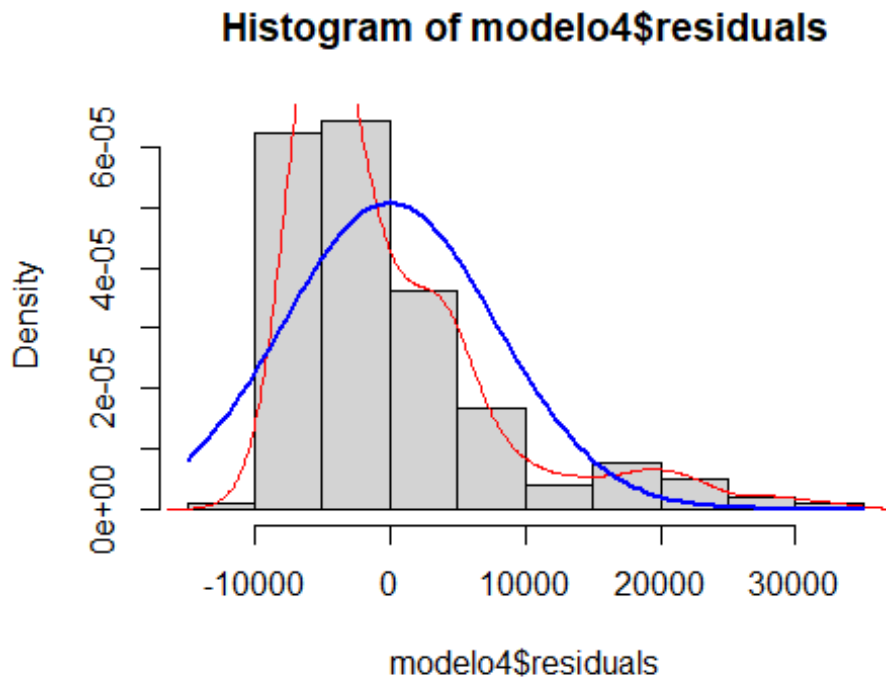
##
##  Anderson-Darling normality test
##
## data:  modelo4$residuals
## A = 11.901, p-value < 2.2e-16

qqnorm(modelo4$residuals, col = "#5F9EA0")
qqline(modelo4$residuals, col = "red")
```

Normal Q-Q Plot



```
hist(modelo4$residuals,freq=FALSE)
lines(density(modelo4$residual),col="red")
curve(dnorm(x,mean=mean(modelo4$residuals),sd=sd(modelo4$residuals)),
add=TRUE, col="blue",lwd=2)
```



Rechazamos H_0 ya que el valor $p < \alpha = 0.04$ por lo que podemos decir que nuestros datos no provienen de una distribución normal.

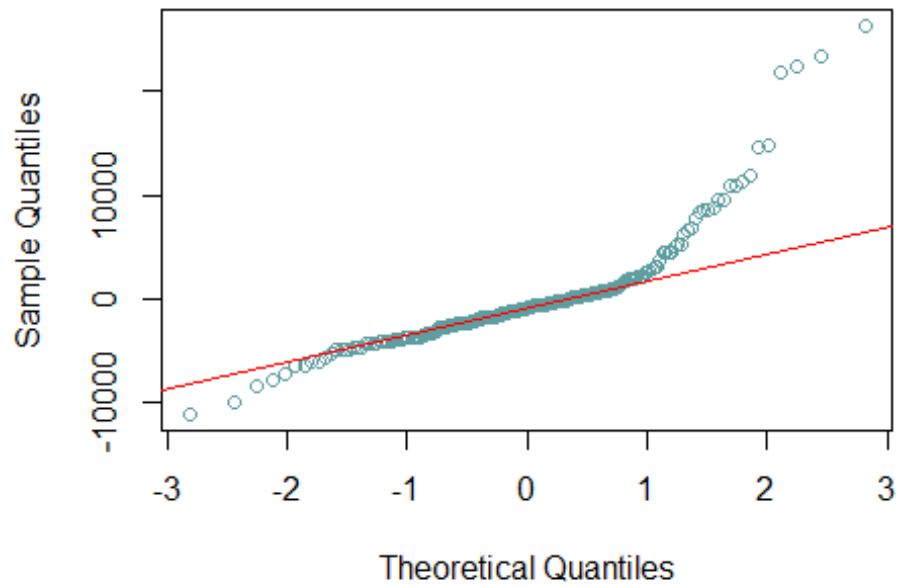
Modelo 5

```
library(nortest)
ad.test(modelo5$residuals)

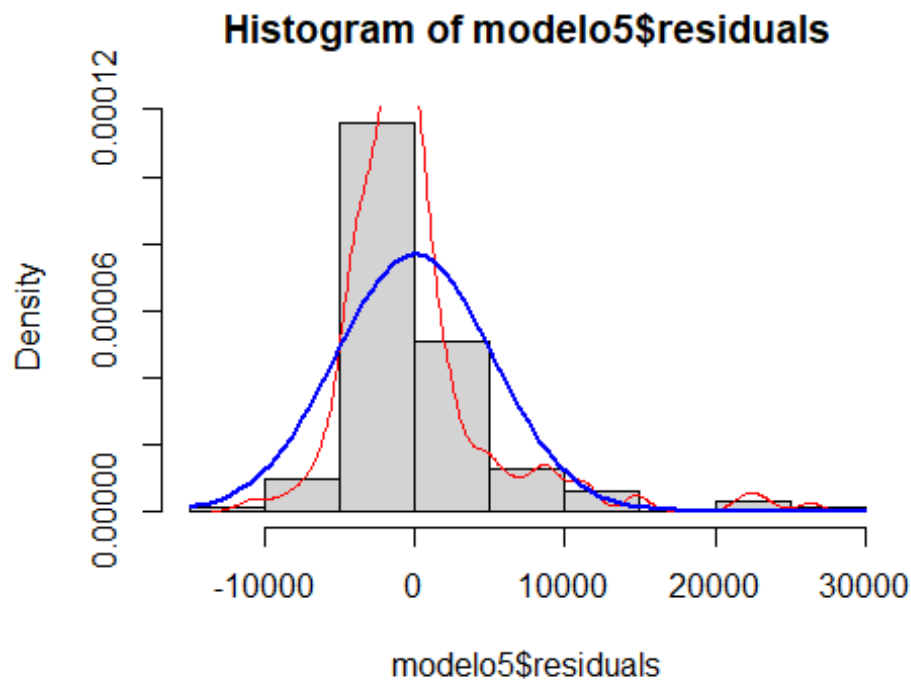
##
## Anderson-Darling normality test
##
## data: modelo5$residuals
## A = 10.961, p-value < 2.2e-16

qqnorm(modelo5$residuals, col = "#5F9EA0")
qqline(modelo5$residuals, col = "red")
```

Normal Q-Q Plot



```
hist(modelo5$residuals,freq=FALSE)
lines(density(modelo5$residual),col="red")
curve(dnorm(x,mean=mean(modelo5$residuals),sd=sd(modelo5$residuals)),
add=TRUE, col="blue",lwd=2)
```

Rechazamos

H_0 ya que el valor $p < \alpha = 0.04$ por lo que podemos decir que nuestros datos no provienen de una distribución normal.

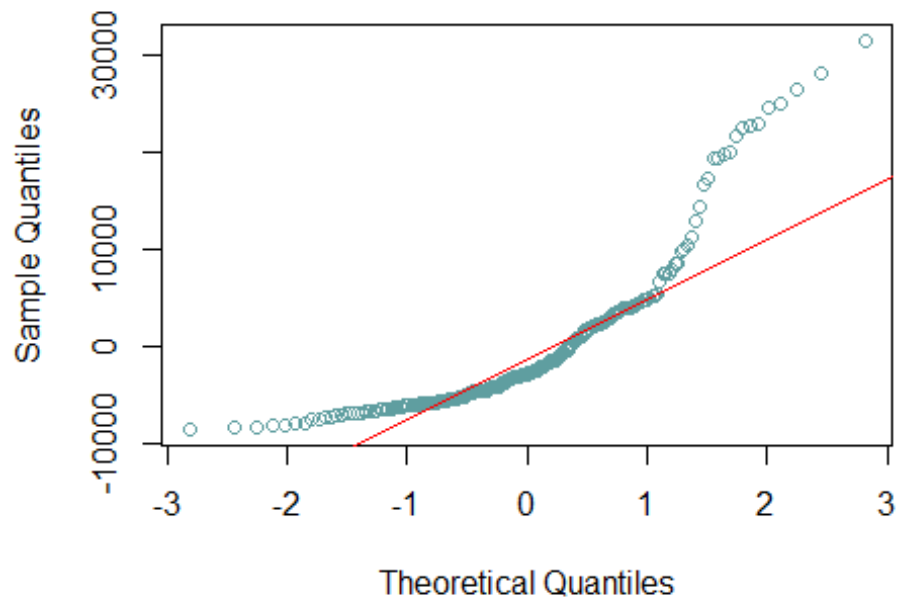
Modelo 6

```
library(nortest)
ad.test(modelo6$residuals)

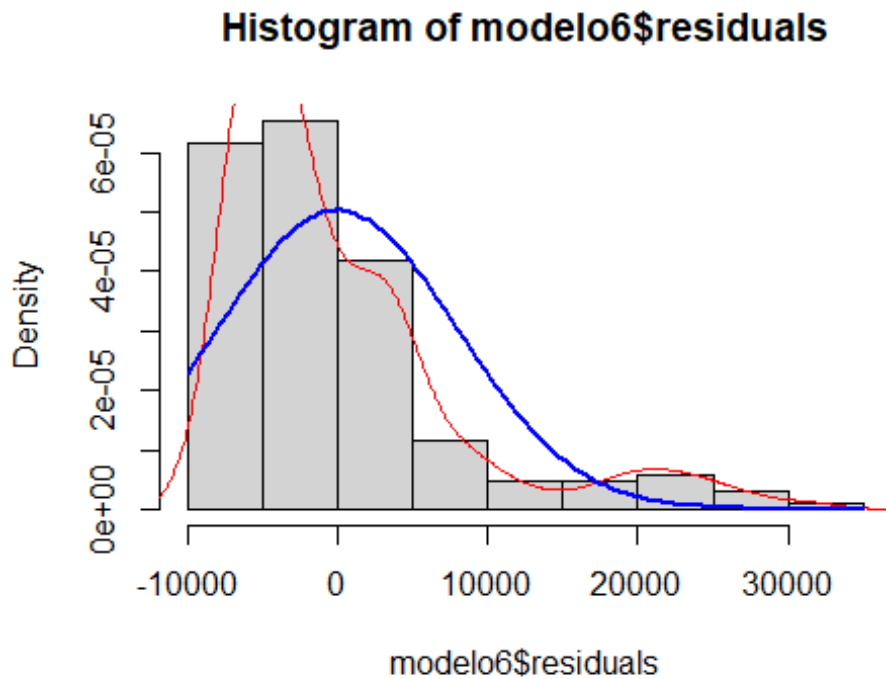
##
## Anderson-Darling normality test
##
## data: modelo6$residuals
## A = 12.504, p-value < 2.2e-16

qqnorm(modelo6$residuals, col = "#5F9EA0")
qqline(modelo6$residuals, col = "red")
```

Normal Q-Q Plot



```
hist(modelo6$residuals,freq=FALSE)
lines(density(modelo6$residual),col="red")
curve(dnorm(x,mean=mean(modelo6$residuals),sd=sd(modelo6$residuals)),
add=TRUE, col="blue",lwd=2)
```



Rechazamos

H_0 ya que el valor $p < \alpha = 0.04$ por lo que podemos decir que nuestros datos no provienen de una distribución normal.

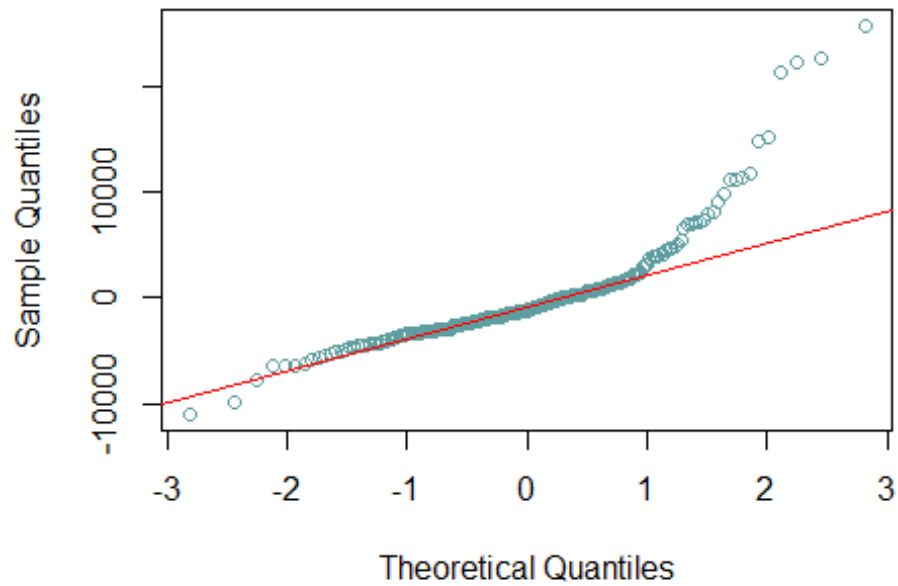
Modelo 7

```
library(nortest)
ad.test(modelo7$residuals)

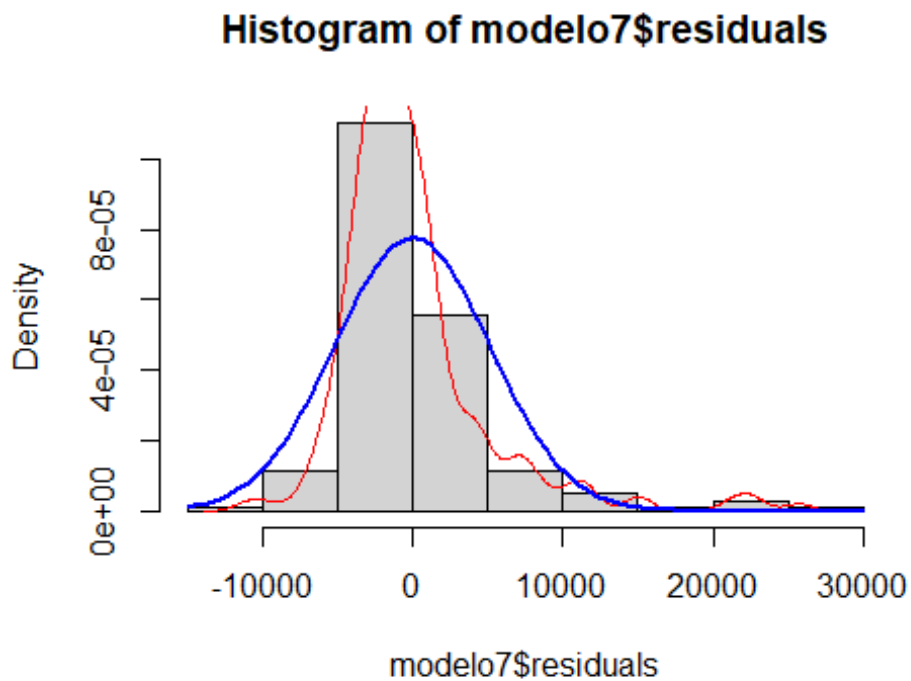
##
## Anderson-Darling normality test
##
## data: modelo7$residuals
## A = 10.319, p-value < 2.2e-16

qqnorm(modelo7$residuals, col = "#5F9EA0")
qqline(modelo7$residuals, col = "red")
```

Normal Q-Q Plot



```
hist(modelo7$residuals,freq=FALSE)
lines(density(modelo7$residual),col="red")
curve(dnorm(x,mean=mean(modelo7$residuals),sd=sd(modelo7$residuals)),
add=TRUE, col="blue",lwd=2)
```



Rechazamos

H_0 ya que el valor $p < \alpha = 0.04$ por lo que podemos decir que nuestros datos no provienen de una distribución normal.

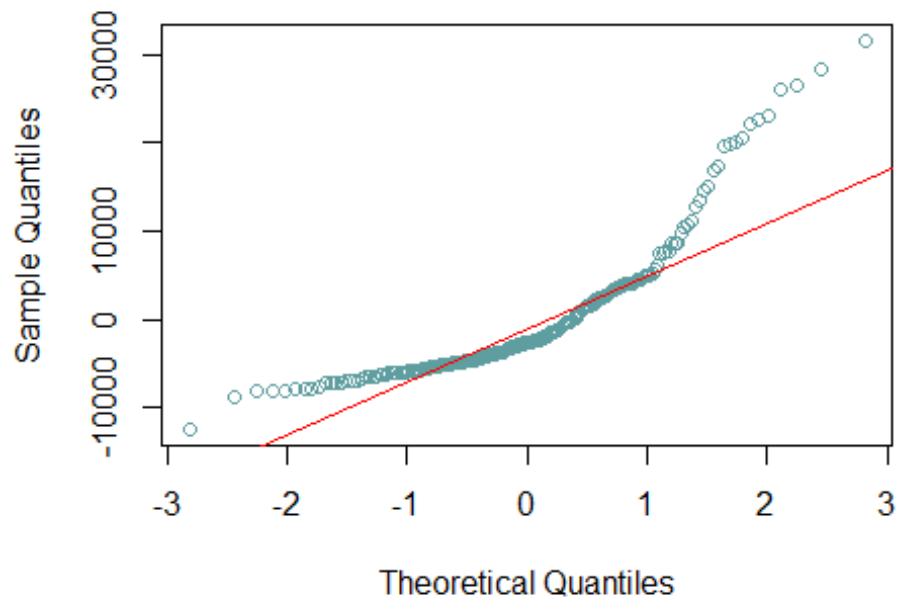
Modelo 8

```
library(nortest)
ad.test(modelo8$residuals)

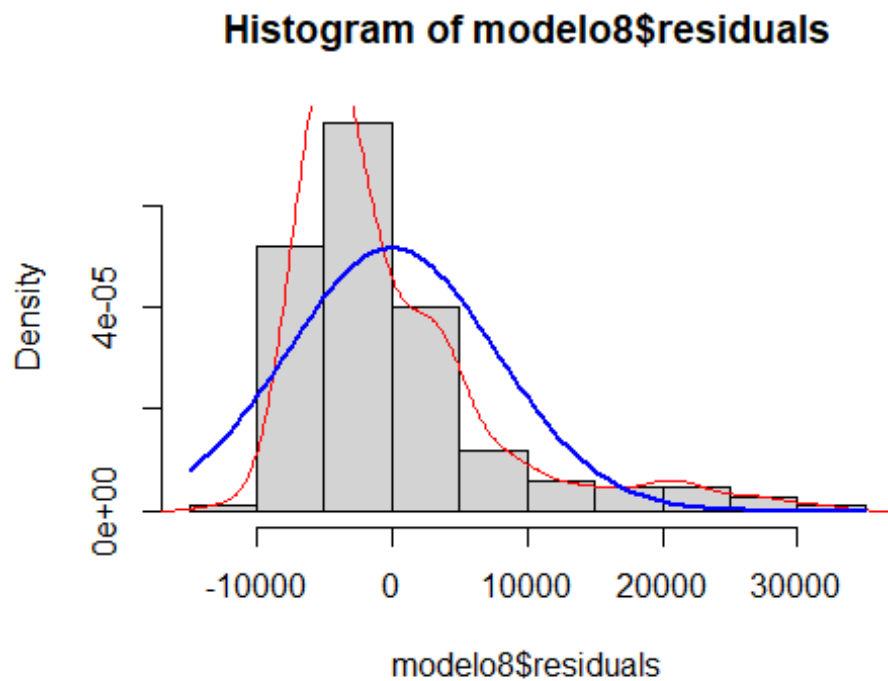
##
## Anderson-Darling normality test
##
## data:  modelo8$residuals
## A = 11.712, p-value < 2.2e-16

qqnorm(modelo8$residuals, col = "#5F9EA0")
qqline(modelo8$residuals, col = "red")
```

Normal Q-Q Plot



```
hist(modelo8$residuals,freq=FALSE)
lines(density(modelo8$residual),col="red")
curve(dnorm(x,mean=mean(modelo8$residuals),sd=sd(modelo8$residuals)),
add=TRUE, col="blue",lwd=2)
```



Rechazamos

H_0 ya que el valor $p < \alpha = 0.04$ por lo que podemos decir que nuestros datos no provienen de una distribución normal.

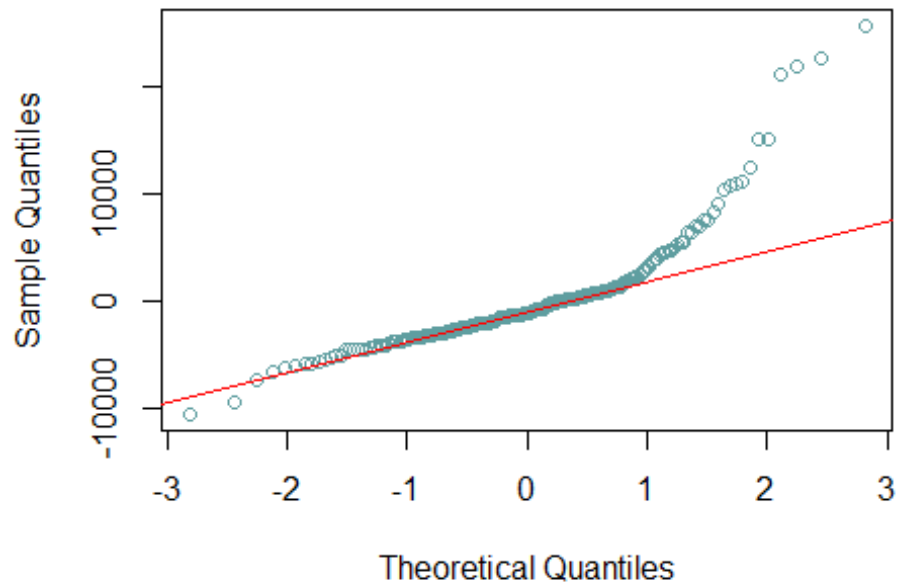
Modelo 9

```
library(nortest)
ad.test(modelo9$residuals)

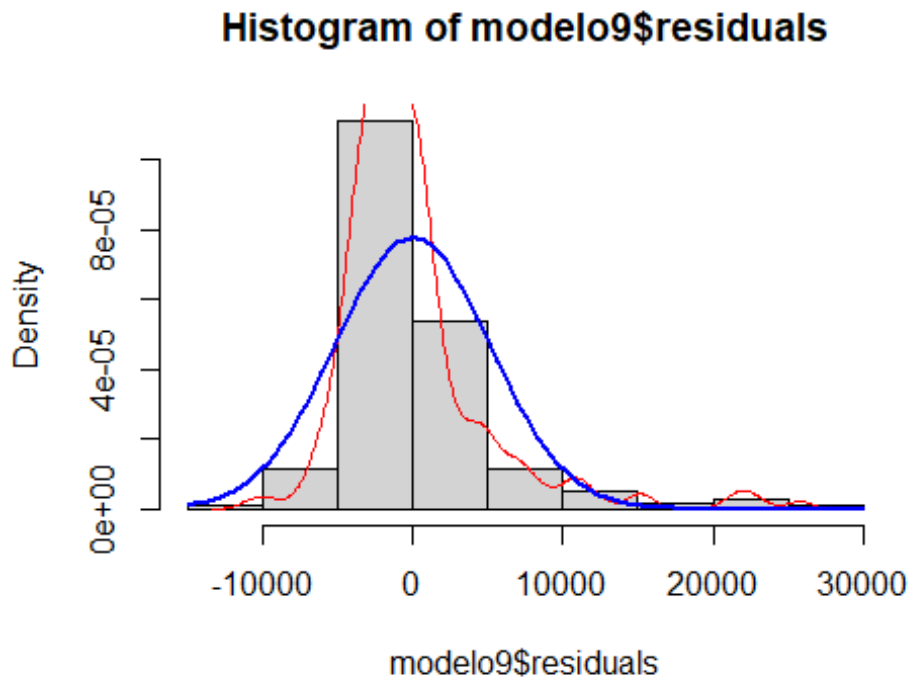
##
## Anderson-Darling normality test
##
## data:  modelo9$residuals
## A = 10.43, p-value < 2.2e-16

qqnorm(modelo9$residuals, col = "#5F9EA0")
qqline(modelo9$residuals, col = "red")
```

Normal Q-Q Plot



```
hist(modelo9$residuals,freq=FALSE)
lines(density(modelo9$residual),col="red")
curve(dnorm(x,mean=mean(modelo9$residuals),sd=sd(modelo9$residuals)),
add=TRUE, col="blue",lwd=2)
```

Rechazamos

H_0 ya que el valor $p < \alpha = 0.04$ por lo que podemos decir que nuestros datos no provienen de una distribución normal

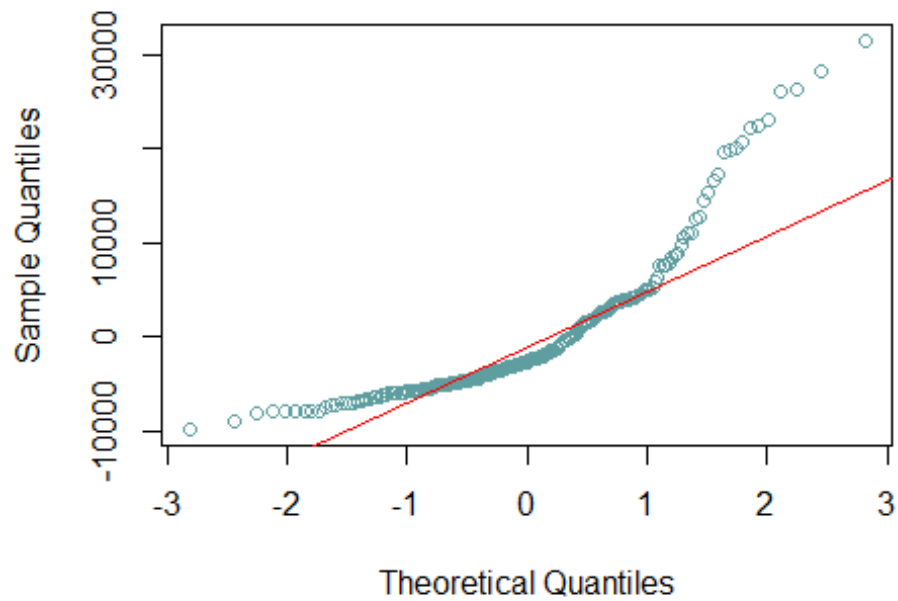
Modelo 10

```
library(nortest)
ad.test(modelo10$residuals)

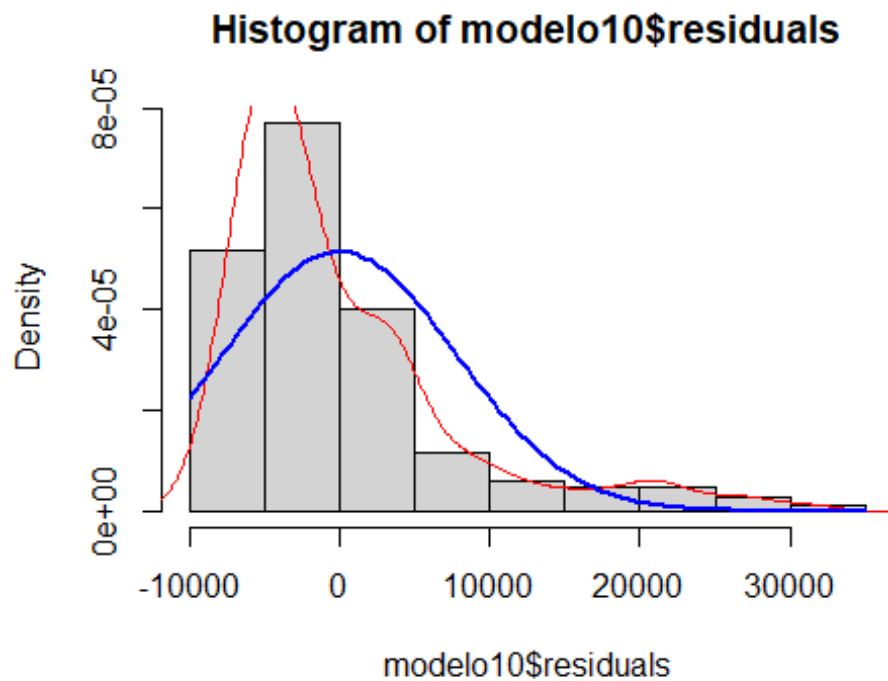
##
## Anderson-Darling normality test
##
## data:  modelo10$residuals
## A = 11.585, p-value < 2.2e-16

qqnorm(modelo10$residuals, col = "#5F9EA0")
qqline(modelo10$residuals, col = "red")
```

Normal Q-Q Plot



```
hist(modelo10$residuals,freq=FALSE)
lines(density(modelo10$residual),col="red")
curve(dnorm(x,mean=mean(modelo10$residuals),sd=sd(modelo10$residuals)),
add=TRUE, col="blue",lwd=2)
```



Rechazamos

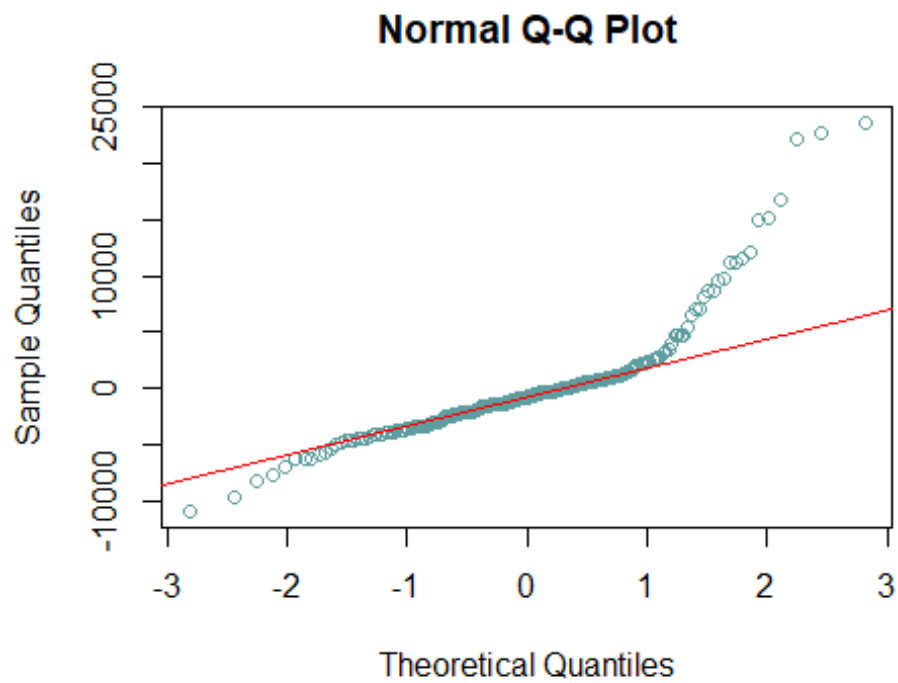
H_0 ya que el valor $p < \alpha = 0.04$ por lo que podemos decir que nuestros datos no provienen de una distribución normal.

Modelo 11

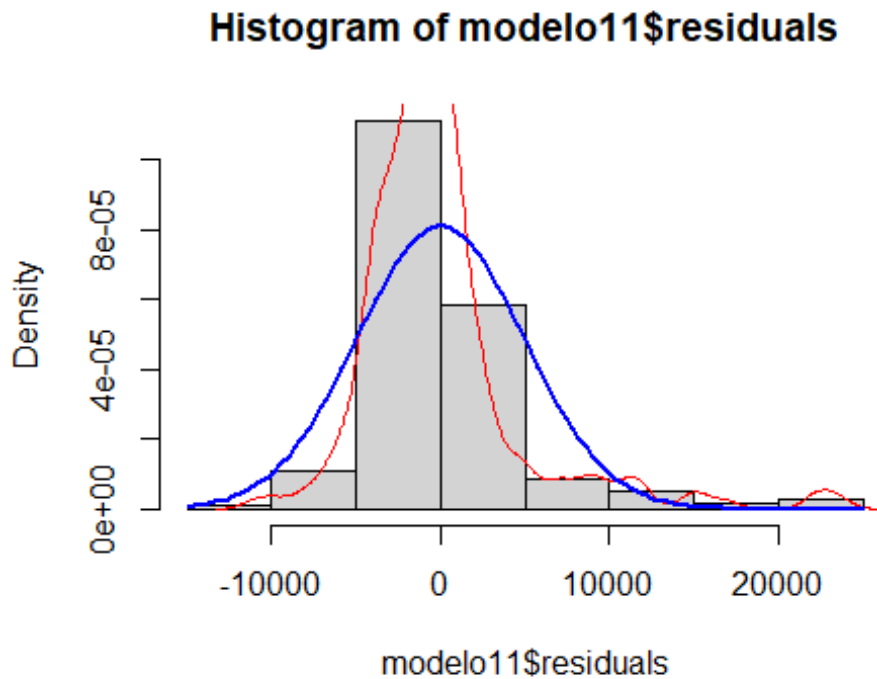
```
library(nortest)
ad.test(modelo11$residuals)

##
## Anderson-Darling normality test
##
## data: modelo11$residuals
## A = 10.765, p-value < 2.2e-16

qqnorm(modelo11$residuals, col = "#5F9EA0")
qqline(modelo11$residuals, col = "red")
```



```
hist(modelo11$residuals,freq=FALSE)  
lines(density(modelo11$residual),col="red")  
curve(dnorm(x,mean=mean(modelo11$residuals),sd=sd(modelo11$residuals)),  
add=TRUE, col="blue",lwd=2)
```



Rechazamos

H_0 ya que el valor $p < \alpha = 0.04$ por lo que podemos decir que nuestros datos no provienen de una distribución normal.

Verificación de media cero

Prueba de Hipótesis

- $H_0: \mu_e = 0$
- $H_1: \mu_e \neq 0$

Regla de decisión * Se rechaza si valor $p < \alpha$

Modelo 1

```
t.test(modelo1$residuals)

##
## One Sample t-test
##
## data:  modelo1$residuals
## t = 1.5987e-16, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -674.7772  674.7772
## sample estimates:
##  mean of x
##  5.471504e-14
```

Aceptamos H_0 ya que nuestro valor $p = 1 > \alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Modelo 2

```
t.test(modelo2$residuals)

##
## One Sample t-test
##
## data:  modelo2$residuals
## t = 3.6151e-16, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -672.9939  672.9939
## sample estimates:
##  mean of x
## 1.233951e-13
```

Aceptamos H_0 ya que nuestro valor $p = 1 > \alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Modelo 3

```
t.test(modelo3$residuals)

##
## One Sample t-test
##
## data:  modelo3$residuals
## t = -2.5793e-16, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -678.3973  678.3973
## sample estimates:
##  mean of x
## -8.874635e-14
```

Aceptamos H_0 ya que nuestro valor $p = 1 > \alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Modelo 4

```
t.test(modelo4$residuals)

##
## One Sample t-test
##
## data:  modelo4$residuals
## t = 2.0001e-17, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
```

```
## -1080.57 1080.57
## sample estimates:
## mean of x
## 1.096142e-14
```

Aceptamos H_0 ya que nuestro valor $p = 1 > \alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Modelo 5

```
t.test(modelo5$residuals)

##
## One Sample t-test
##
## data: modelo5$residuals
## t = -9.7813e-16, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -715.8598 715.8598
## sample estimates:
## mean of x
## -3.551338e-13
```

Aceptamos H_0 ya que nuestro valor $p = 1 > \alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Modelo 6

```
t.test(modelo6$residuals)

##
## One Sample t-test
##
## data: modelo6$residuals
## t = -2.4903e-16, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1092.257 1092.257
## sample estimates:
## mean of x
## -1.379579e-13
```

Aceptamos H_0 ya que nuestro valor $p = 1 > \alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Modelo 7

```
t.test(modelo7$residuals)

##
## One Sample t-test
```

```
##
## data: modelo7$residuals
## t = -4.443e-17, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -707.942 707.942
## sample estimates:
## mean of x
## -1.595309e-14
```

Aceptamos H_0 ya que nuestro valor $p = 1 > \alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Modelo 8

```
t.test(modelo8$residuals)
##
## One Sample t-test
##
## data: modelo8$residuals
## t = 7.6893e-16, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1067.551 1067.551
## sample estimates:
## mean of x
## 4.163347e-13
```

Aceptamos H_0 ya que nuestro valor $p = 1 > \alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Modelo 9

```
t.test(modelo9$residuals)
##
## One Sample t-test
##
## data: modelo9$residuals
## t = 6.0862e-16, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -706.5664 706.5664
## sample estimates:
## mean of x
## 2.181063e-13
```

Aceptamos H_0 ya que nuestro valor $p = 1 > \alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Modelo 10


```
t.test(modelo10$residuals)

##
## One Sample t-test
##
## data: modelo10$residuals
## t = 1.3296e-15, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -1065.827 1065.827
## sample estimates:
## mean of x
## 7.187573e-13
```

Aceptamos H_0 ya que nuestro valor $p = 1 > \alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Modelo 11

```
t.test(modelo11$residuals)

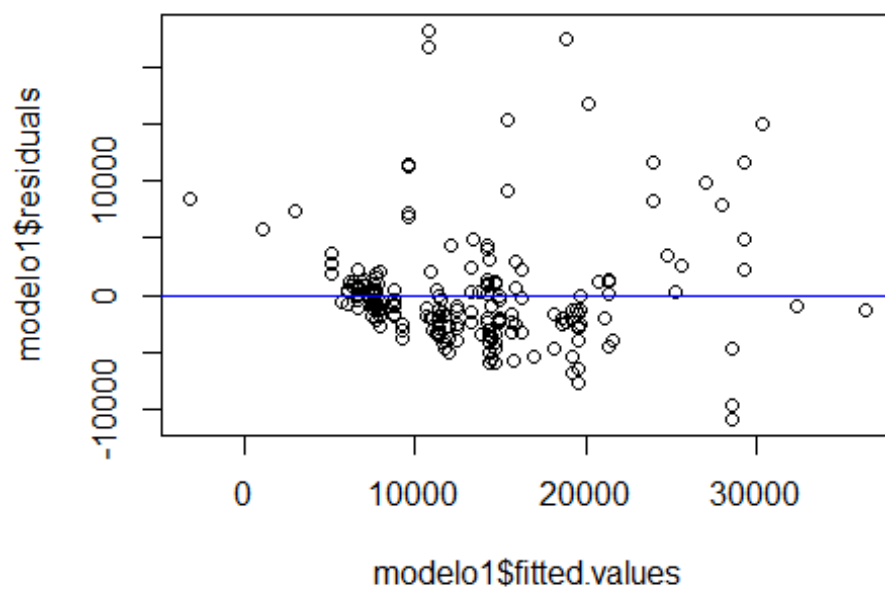
##
## One Sample t-test
##
## data: modelo11$residuals
## t = 2.0766e-15, df = 204, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## -678.3803 678.3803
## sample estimates:
## mean of x
## 7.144767e-13
```

Aceptamos H_0 ya que nuestro valor $p = 1 > \alpha = 0.04$ entonces podemos concluir que $\mu_e = 0$. Los residuos tienen media cero: el modelo es bueno.

Homocedasticidad, linealidad e independencia

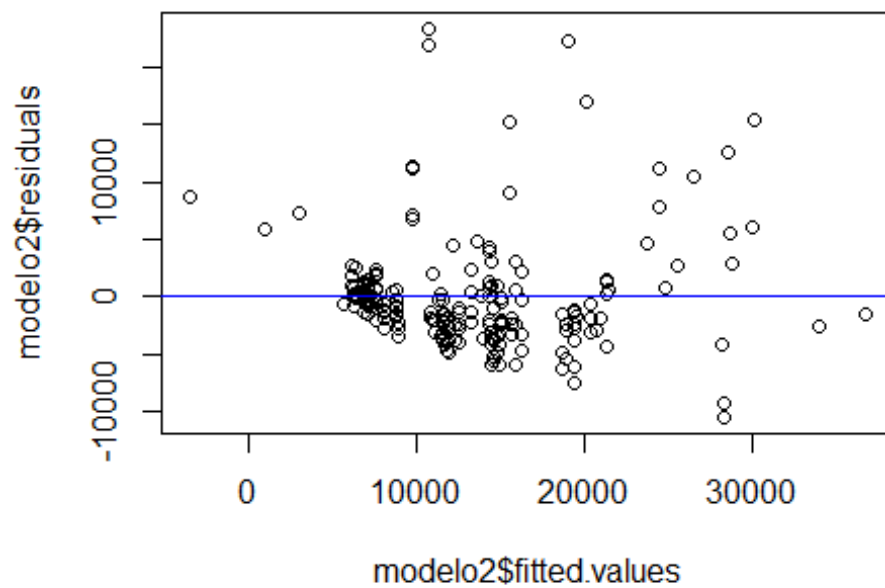
Modelo 1

```
plot(modelo1$fitted.values, modelo1$residuals)
abline(h=0, col="blue")
```



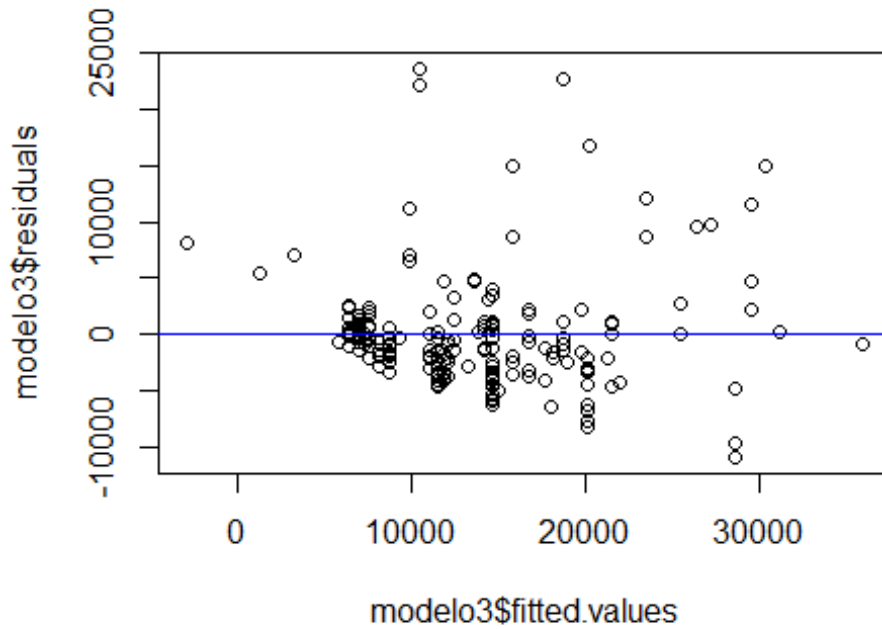
Modelo 2

```
plot(modelo2$fitted.values,modelo2$residuals)  
abline(h=0, col="blue")
```



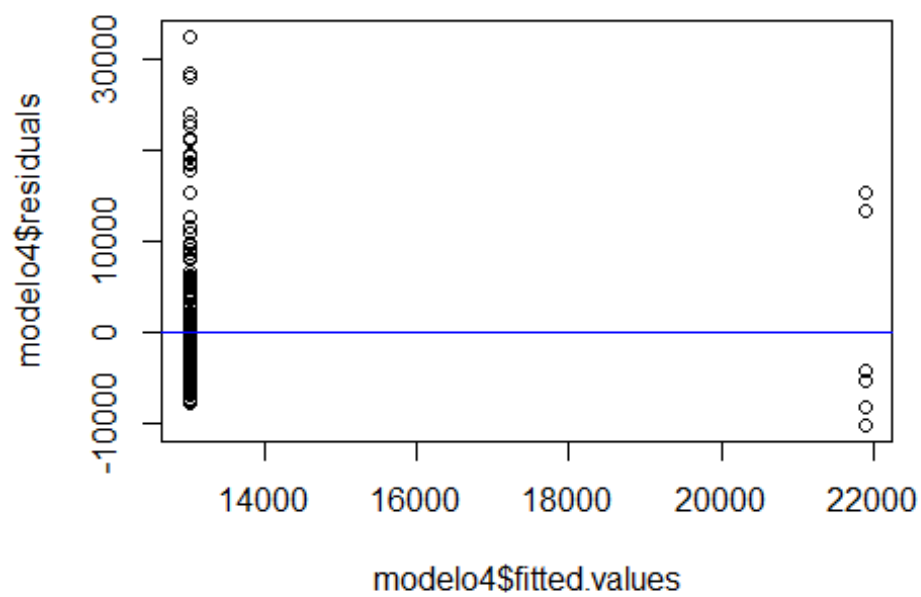
Modelo 3

```
plot(modelo3$fitted.values,modelo3$residuals)  
abline(h=0, col="blue")
```



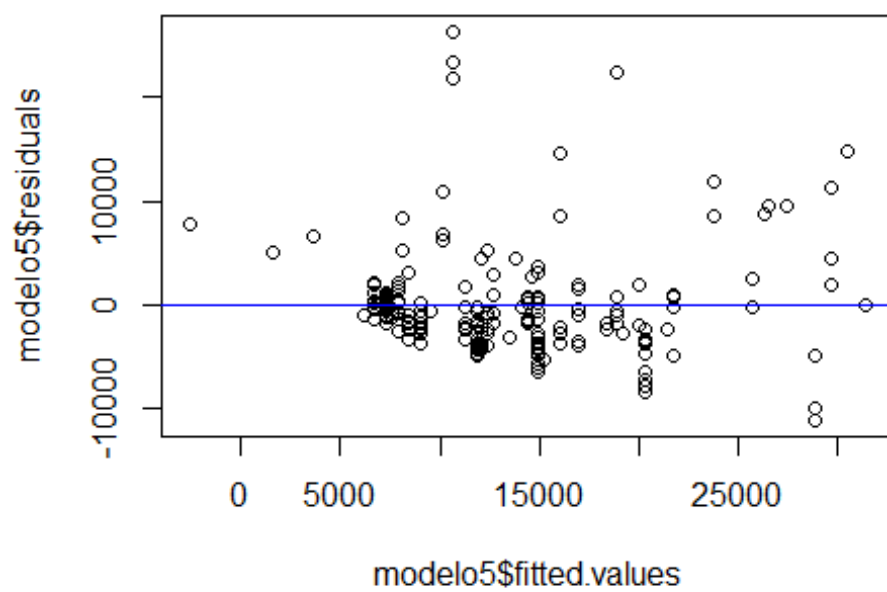
Modelo 4

```
plot(modelo4$fitted.values,modelo4$residuals)  
abline(h=0, col="blue")
```



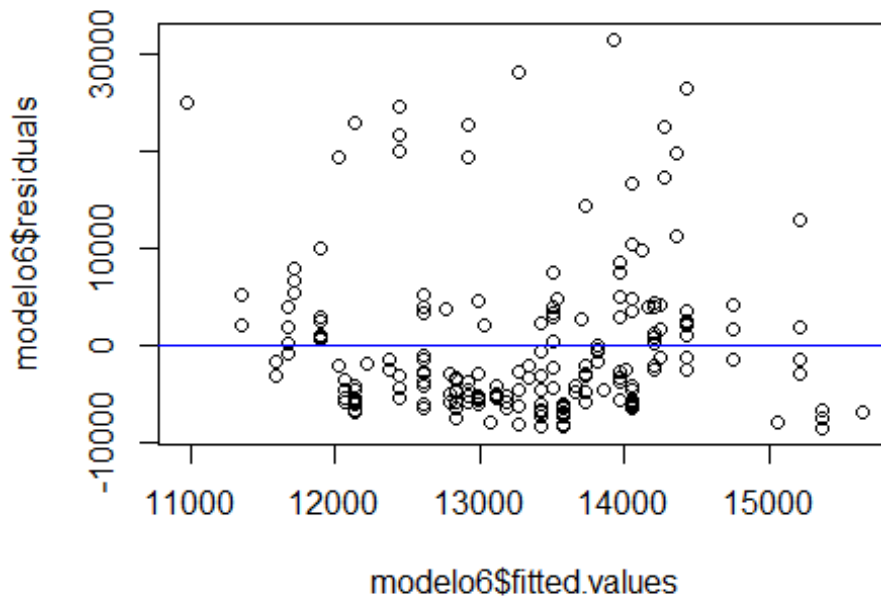
Modelo 5

```
plot(modelo5$fitted.values,modelo5$residuals)
abline(h=0, col="blue")
```



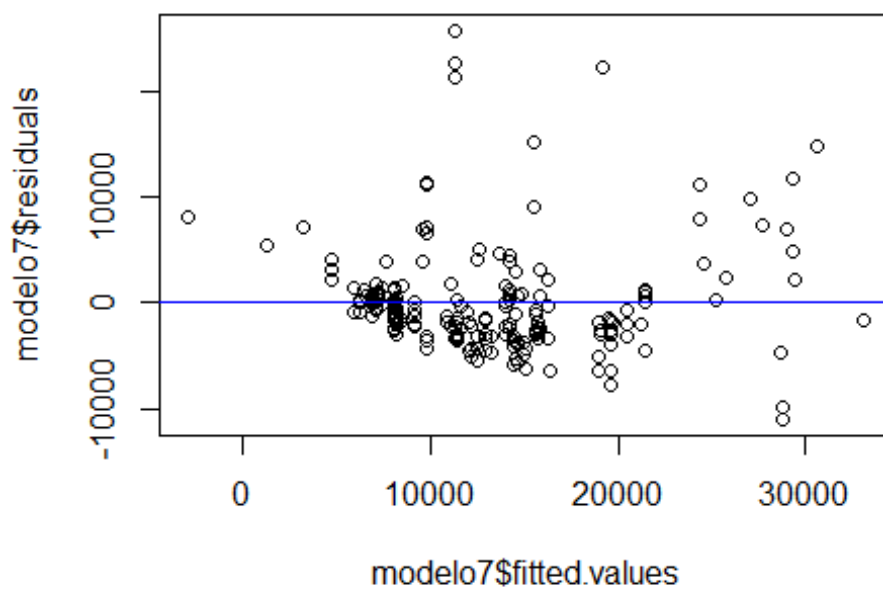
Modelo 6

```
plot(modelo6$fitted.values,modelo6$residuals)  
abline(h=0, col="blue")
```



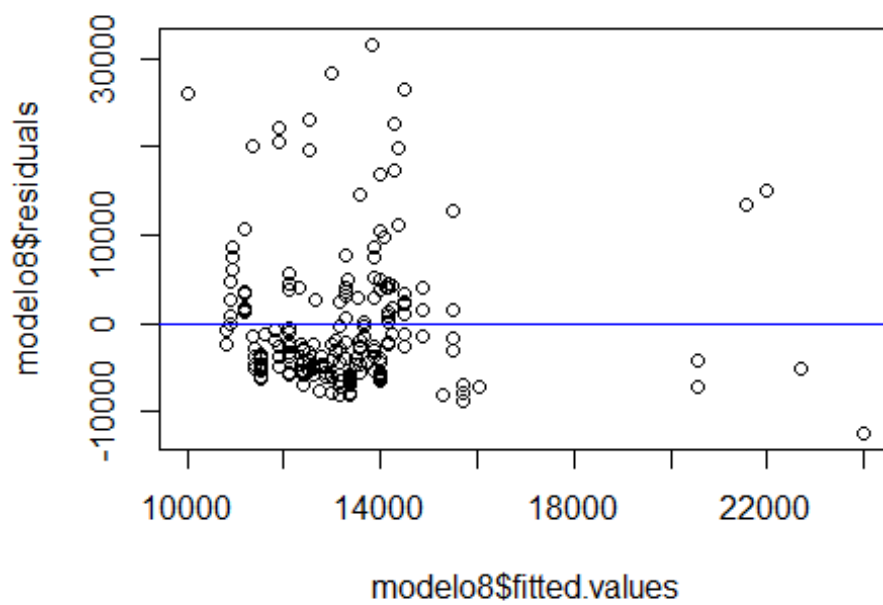
Modelo 7

```
plot(modelo7$fitted.values,modelo7$residuals)  
abline(h=0, col="blue")
```



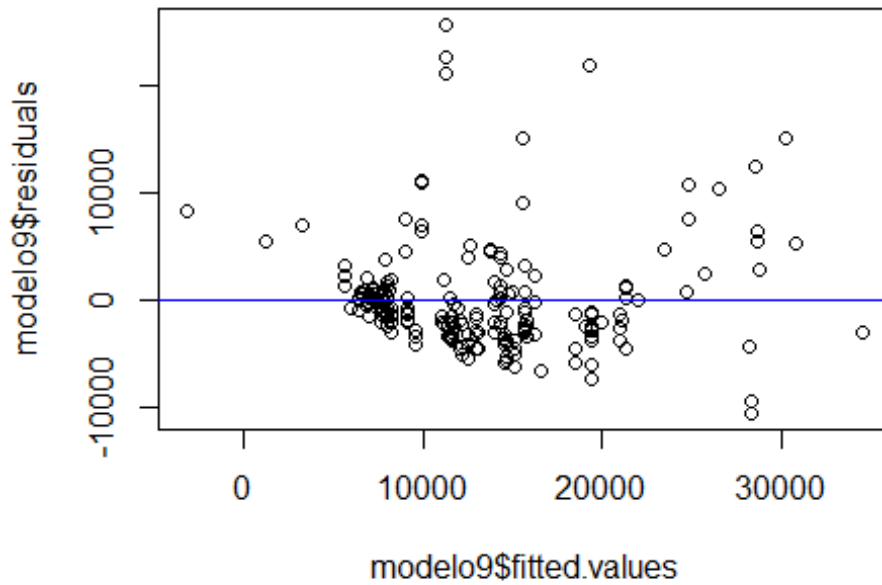
Modelo 8

```
plot(modelo8$fitted.values,modelo8$residuals)  
abline(h=0, col="blue")
```



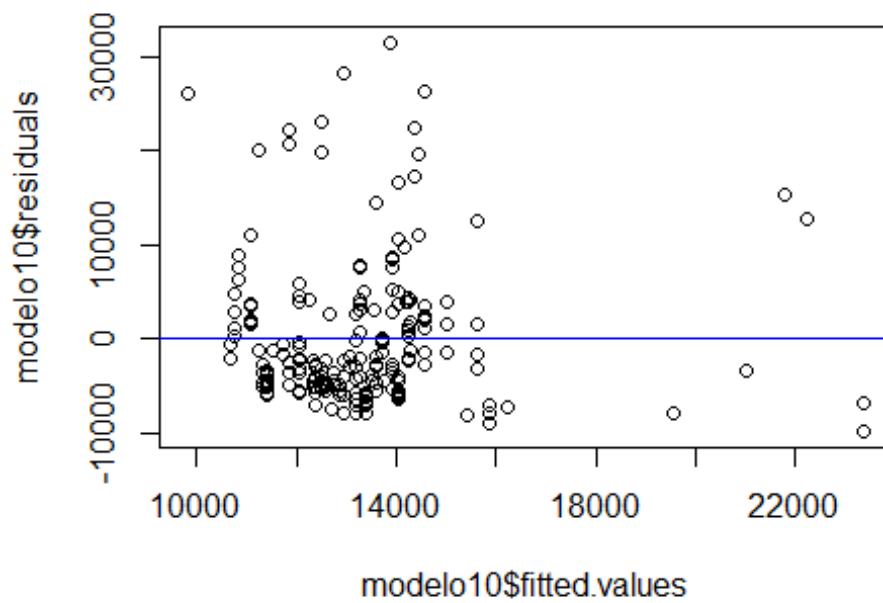
Modelo 9

```
plot(modelo9$fitted.values,modelo9$residuals)  
abline(h=0, col="blue")
```



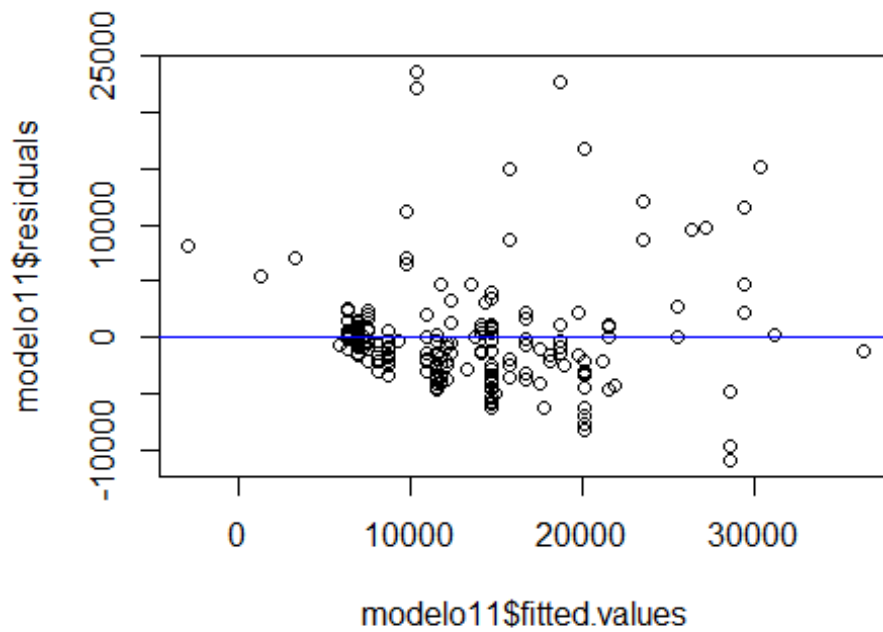
Modelo 10

```
plot(modelo10$fitted.values,modelo10$residuals)  
abline(h=0, col="blue")
```



Modelo 11

```
plot(modelo11$fitted.values, modelo11$residuals)  
abline(h=0, col="blue")
```



Pruebas de hipótesis para independencia

Test de Durbin-Watson y Prueba Breusch-Godfrey

- H_0 : Los errores no están autocorrelacionados.
- H_1 : Los errores están autocorrelacionados.

Regla de decisión: Se rechaza H_0 si valor $p < \alpha$

Modelo 1

```
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

dwtest(modelo1)

##
## Durbin-Watson test
##
## data:  modelo1
## DW = 0.6719, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0

bgtest(modelo1)

##
## Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  modelo1
## LM test = 92.308, df = 1, p-value < 2.2e-16
```

Rechazo H_0 ya que valor $p < \alpha = 0.04$ lo que significa que los errores están autocorrelacionados.

Modelo 2

```
library(lmtest)
dwtest(modelo2)

##
## Durbin-Watson test
##
## data:  modelo2
## DW = 0.68451, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

```
bgtest(modelo2)
```

```
##  
## Breusch-Godfrey test for serial correlation of order up to 1  
##  
## data: modelo2  
## LM test = 92.525, df = 1, p-value < 2.2e-16
```

Rechazo H_0 ya que valor $p < \alpha = 0.04$ lo que significa que los errores están autocorrelacionados.

Modelo 3

```
library(lmtest)  
dwtest(modelo3)
```

```
##  
## Durbin-Watson test  
##  
## data: modelo3  
## DW = 0.65511, p-value < 2.2e-16  
## alternative hypothesis: true autocorrelation is greater than 0
```

```
bgtest(modelo3)
```

```
##  
## Breusch-Godfrey test for serial correlation of order up to 1  
##  
## data: modelo3  
## LM test = 94.277, df = 1, p-value < 2.2e-16
```

Rechazo H_0 ya que valor $p < \alpha = 0.04$ lo que significa que los errores están autocorrelacionados.

Modelo 4

```
library(lmtest)  
dwtest(modelo4)
```

```
##  
## Durbin-Watson test  
##  
## data: modelo4  
## DW = 0.48185, p-value < 2.2e-16  
## alternative hypothesis: true autocorrelation is greater than 0
```

```
bgtest(modelo4)
```

```
##  
## Breusch-Godfrey test for serial correlation of order up to 1  
##  
## data: modelo4  
## LM test = 118.4, df = 1, p-value < 2.2e-16
```

Rechazo H_0 ya que valor $p < \alpha = 0.04$ lo que significa que los errores están autocorrelacionados.

Modelo 5

```
library(lmtest)
dwtest(modelo5)

##
## Durbin-Watson test
##
## data: modelo5
## DW = 0.63823, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0

bgtest(modelo5)

##
## Breusch-Godfrey test for serial correlation of order up to 1
##
## data: modelo5
## LM test = 95.479, df = 1, p-value < 2.2e-16
```

Rechazo H_0 ya que valor $p < \alpha = 0.04$ lo que significa que los errores están autocorrelacionados.

Modelo 6

```
library(lmtest)
dwtest(modelo6)

##
## Durbin-Watson test
##
## data: modelo6
## DW = 0.46823, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0

bgtest(modelo6)

##
## Breusch-Godfrey test for serial correlation of order up to 1
##
## data: modelo6
## LM test = 120.39, df = 1, p-value < 2.2e-16
```

Rechazo H_0 ya que valor $p < \alpha = 0.04$ lo que significa que los errores están autocorrelacionados.

Modelo 7

```
library(lmtest)
dwtest(modelo7)
```

```
##
## Durbin-Watson test
##
## data: modelo7
## DW = 0.67299, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0

bgtest(modelo7)

##
## Breusch-Godfrey test for serial correlation of order up to 1
##
## data: modelo7
## LM test = 92.131, df = 1, p-value < 2.2e-16
```

Rechazo H_0 ya que valor $p < \alpha = 0.04$ lo que significa que los errores están autocorrelacionados.

Modelo 8

```
library(lmtest)
dwtest(modelo8)

##
## Durbin-Watson test
##
## data: modelo8
## DW = 0.53167, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0

bgtest(modelo8)

##
## Breusch-Godfrey test for serial correlation of order up to 1
##
## data: modelo8
## LM test = 111.83, df = 1, p-value < 2.2e-16
```

Rechazo H_0 ya que valor $p < \alpha = 0.04$ lo que significa que los errores están autocorrelacionados.

Modelo 9

```
library(lmtest)
dwtest(modelo9)

##
## Durbin-Watson test
##
## data: modelo9
## DW = 0.68459, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

```
bgtest(modelo9)
```

```
##  
## Breusch-Godfrey test for serial correlation of order up to 1  
##  
## data: modelo9  
## LM test = 92.519, df = 1, p-value < 2.2e-16
```

Rechazo H_0 ya que valor $p < \alpha = 0.04$ lo que significa que los errores están autocorrelacionados.

Modelo 10

```
library(lmtest)  
dwtest(modelo10)
```

```
##  
## Durbin-Watson test  
##  
## data: modelo10  
## DW = 0.53183, p-value < 2.2e-16  
## alternative hypothesis: true autocorrelation is greater than 0
```

```
bgtest(modelo10)
```

```
##  
## Breusch-Godfrey test for serial correlation of order up to 1  
##  
## data: modelo10  
## LM test = 110.89, df = 1, p-value < 2.2e-16
```

Rechazo H_0 ya que valor $p < \alpha = 0.04$ lo que significa que los errores están autocorrelacionados.

Modelo 11

```
library(lmtest)  
dwtest(modelo11)
```

```
##  
## Durbin-Watson test  
##  
## data: modelo11  
## DW = 0.65738, p-value < 2.2e-16  
## alternative hypothesis: true autocorrelation is greater than 0
```

```
bgtest(modelo11)
```

```
##  
## Breusch-Godfrey test for serial correlation of order up to 1  
##  
## data: modelo11  
## LM test = 94.109, df = 1, p-value < 2.2e-16
```

Rechazo H_0 ya que valor $p < \alpha = 0.04$ lo que significa que los errores están autocorrelacionados.

Pruebas de hipótesis para homocedasticidad

Prueba de Breusch-Pagan y White

- H_0 : La varianza de los errores es constante (homocedasticidad)
- H_1 : La varianza de los errores no es constante (heterocedasticidad)

Regla de decisión: Se rechaza H_0 si valor $p < \alpha$

Modelo 1

```
library(lmtest)
bptest(modelo1)

##
## studentized Breusch-Pagan test
##
## data:  modelo1
## BP = 6.9477, df = 3, p-value = 0.07359

gqtest(modelo1)

##
## Goldfeld-Quandt test
##
## data:  modelo1
## GQ = 0.72021, df1 = 99, df2 = 98, p-value = 0.9476
## alternative hypothesis: variance increases from segment 1 to 2
```

Aceptamos H_0 ya que el valor $p = 0.07359$ y 0.9476 para Breusch-Pagan test y Goldfeld-Quandt test respectivamente siendo mayores que $\alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Modelo 2

```
bptest(modelo2)

##
## studentized Breusch-Pagan test
##
## data:  modelo2
## BP = 8.5439, df = 7, p-value = 0.2871

gqtest(modelo2)

##
## Goldfeld-Quandt test
##
## data:  modelo2
```

```
## GQ = 0.60897, df1 = 95, df2 = 94, p-value = 0.9917
## alternative hypothesis: variance increases from segment 1 to 2
```

Aceptamos H_0 ya que el valor $p = 0.2871$ y 0.9917 para Breusch-Pagan test y Goldfeld-Quandt test respectivamente siendo mayores que $\alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Modelo 3

```
bptest(modelo3)

##
## studentized Breusch-Pagan test
##
## data: modelo3
## BP = 7.2341, df = 2, p-value = 0.02686

gqtest(modelo3)

##
## Goldfeld-Quandt test
##
## data: modelo3
## GQ = 0.75487, df1 = 100, df2 = 99, p-value = 0.9188
## alternative hypothesis: variance increases from segment 1 to 2
```

Si solo consideraramos la prueba de Breusch-Pagan rechazaríamos H_0 ya que valor $p = 0.02686 < \alpha = 0.04$ lo que significa que la varianza de los errores no es constante y existe heterocedasticidad pero si consideramos Goldfeld-Quandt test aceptamos H_0 ya que valor $p = 0.9188 > \alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad)

Modelo 4

```
bptest(modelo4)

##
## studentized Breusch-Pagan test
##
## data: modelo4
## BP = 0.60542, df = 1, p-value = 0.4365

gqtest(modelo4)

##
## Goldfeld-Quandt test
##
## data: modelo4
## GQ = 0.38792, df1 = 101, df2 = 100, p-value = 1
## alternative hypothesis: variance increases from segment 1 to 2
```

Aceptamos H_0 ya que el valor $p = 0.4365$ y 1 para Breusch-Pagan test y Goldfeld-Quandt test respectivamente siendo mayores que $\alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Modelo 5

```
bptest(modelo5)

##
##  studentized Breusch-Pagan test
##
## data:  modelo5
## BP = 4.0726, df = 1, p-value = 0.04358

gqtest(modelo5)

##
##  Goldfeld-Quandt test
##
## data:  modelo5
## GQ = 0.8777, df1 = 101, df2 = 100, p-value = 0.7429
## alternative hypothesis: variance increases from segment 1 to 2
```

Aceptamos H_0 ya que el valor $p = 0.04358$ y 0.7429 para Breusch-Pagan test y Goldfeld-Quandt test respectivamente siendo mayores que $\alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Modelo 6

```
bptest(modelo6)

##
##  studentized Breusch-Pagan test
##
## data:  modelo6
## BP = 0.03553, df = 1, p-value = 0.8505

gqtest(modelo6)

##
##  Goldfeld-Quandt test
##
## data:  modelo6
## GQ = 0.43957, df1 = 101, df2 = 100, p-value = 1
## alternative hypothesis: variance increases from segment 1 to 2
```

Aceptamos H_0 ya que el valor $p = 0.8505$ y 1 para Breusch-Pagan test y Goldfeld-Quandt test respectivamente siendo mayores que $\alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Modelo 7


```
bptest(modelo7)

##
## studentized Breusch-Pagan test
##
## data: modelo7
## BP = 4.6072, df = 2, p-value = 0.0999
```

```
gqtest(modelo7)

##
## Goldfeld-Quandt test
##
## data: modelo7
## GQ = 0.8351, df1 = 100, df2 = 99, p-value = 0.815
## alternative hypothesis: variance increases from segment 1 to 2
```

Aceptamos H_0 ya que el valor $p = 0.0999$ y 0.815 para Breusch-Pagan test y Goldfeld-Quandt test respectivamente siendo mayores que $\alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Modelo 8

```
bptest(modelo8)

##
## studentized Breusch-Pagan test
##
## data: modelo8
## BP = 0.9035, df = 2, p-value = 0.6365
```

```
gqtest(modelo8)

##
## Goldfeld-Quandt test
##
## data: modelo8
## GQ = 0.43279, df1 = 100, df2 = 99, p-value = 1
## alternative hypothesis: variance increases from segment 1 to 2
```

Aceptamos H_0 ya que el valor $p = 0.6365$ y 1 para Breusch-Pagan test y Goldfeld-Quandt test respectivamente siendo mayores que $\alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Modelo 9

```
bptest(modelo9)

##
## studentized Breusch-Pagan test
##
```

```
## data: modelo9
## BP = 5.2068, df = 3, p-value = 0.1573
```

```
gqtest(modelo9)
```

```
##
## Goldfeld-Quandt test
##
## data: modelo9
## GQ = 0.84264, df1 = 99, df2 = 98, p-value = 0.8016
## alternative hypothesis: variance increases from segment 1 to 2
```

Aceptamos H_0 ya que el valor $p = 0.1573$ y 0.8016 para Breusch-Pagan test y Goldfeld-Quandt test respectivamente siendo mayores que $\alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Modelo 10

```
bptest(modelo10)
```

```
##
## studentized Breusch-Pagan test
##
## data: modelo10
## BP = 0.70212, df = 3, p-value = 0.8727
```

```
gqtest(modelo10)
```

```
##
## Goldfeld-Quandt test
##
## data: modelo10
## GQ = 0.40854, df1 = 99, df2 = 98, p-value = 1
## alternative hypothesis: variance increases from segment 1 to 2
```

Aceptamos H_0 ya que el valor $p = 0.8727$ y 1 para Breusch-Pagan test y Goldfeld-Quandt test respectivamente siendo mayores que $\alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad).

Modelo 11

```
bptest(modelo11)
```

```
##
## studentized Breusch-Pagan test
##
## data: modelo11
## BP = 8.3582, df = 3, p-value = 0.03916
```

```
gqtest(modelo11)
```

```
##
## Goldfeld-Quandt test
```

```
##
## data:  modelo11
## GQ = 0.75, df1 = 99, df2 = 98, p-value = 0.9225
## alternative hypothesis: variance increases from segment 1 to 2
```

Si solo consideráramos la prueba de Breusch-Pagan rechazaríamos H_0 ya que valor $p = 0.03916 < \alpha = 0.04$ lo que significa que la varianza de los errores no es constante y existe heterocedasticidad pero si consideramos Goldfeld-Quandt test aceptamos H_0 ya que valor $p = 0.9225 > \alpha = 0.04$ lo que significa que La varianza de los errores es constante (hay homocedasticidad)

Pruebas de hipótesis para linealidad

- H_0 : No hay términos omitidos que indican linealidad
- H_1 : Hay una especificación errónea en el modelo que indica no linealidad

Regla de decisión: Se rechaza H_0 si valor $p < \alpha$

Modelo 1

```
resettest(modelo1)

##
## RESET test
##
## data:  modelo1
## RESET = 6.4265, df1 = 2, df2 = 199, p-value = 0.001974
```

Rechazamos H_0 ya que $p\text{-value} = 0.001974 < \alpha = 0.04$ lo que indica que hay una especificación errónea en el modelo que indica no linealidad

Modelo 2

```
resettest(modelo2)

##
## RESET test
##
## data:  modelo2
## RESET = 6.7702, df1 = 2, df2 = 195, p-value = 0.001437
```

Rechazamos H_0 ya que $p\text{-value} = 0.001437 < \alpha = 0.04$ lo que indica que hay una especificación errónea en el modelo que indica no linealidad

Modelo 3

```
resettest(modelo3)

##
## RESET test
##
```

```
## data:  modelo3
## RESET = 5.4501, df1 = 2, df2 = 200, p-value = 0.004958
```

Rechazamos H_0 ya que $p\text{-value} = 0.004958 < \alpha = 0.04$ lo que indica que hay una especificación errónea en el modelo que indica no linealidad

Modelo 4

```
resettest(modelo4)

##
##  RESET test
##
## data:  modelo4
## RESET = 0, df1 = 2, df2 = 201, p-value = 1
```

Aceptamos H_0 ya que $p\text{-value} = 1 > \alpha = 0.04$ por lo que podemos interpretar que existe linealidad

Modelo 5

```
resettest(modelo5)

##
##  RESET test
##
## data:  modelo5
## RESET = 5.4702, df1 = 2, df2 = 201, p-value = 0.004861
```

Rechazamos H_0 ya que $p\text{-value} = 0.004861 < \alpha = 0.04$ lo que indica que hay una especificación errónea en el modelo que indica no linealidad

Modelo 6

```
resettest(modelo6)

##
##  RESET test
##
## data:  modelo6
## RESET = 11.228, df1 = 2, df2 = 201, p-value = 2.384e-05
```

Rechazamos H_0 ya que $p\text{-value} = 2.384e-05 < \alpha = 0.04$ lo que indica que hay una especificación errónea en el modelo que indica no linealidad

Modelo 7

```
resettest(modelo7)

##
##  RESET test
##
```

```
## data:  modelo7
## RESET = 6.2885, df1 = 2, df2 = 200, p-value = 0.002246
```

Rechazamos H_0 ya que $p\text{-value} = 0.002246 < \alpha = 0.04$ lo que indica que hay una especificación errónea en el modelo que indica no linealidad

Modelo 8

```
resettest(modelo8)

##
##  RESET test
##
## data:  modelo8
## RESET = 3.3581, df1 = 2, df2 = 200, p-value = 0.03677
```

Rechazamos H_0 ya que $p\text{-value} = 0.03677 < \alpha = 0.04$ lo que indica que hay una especificación errónea en el modelo que indica no linealidad

Modelo 9

```
resettest(modelo9)

##
##  RESET test
##
## data:  modelo9
## RESET = 6.0337, df1 = 2, df2 = 199, p-value = 0.002857
```

Rechazamos H_0 ya que $p\text{-value} = 0.002857 < \alpha = 0.04$ lo que indica que hay una especificación errónea en el modelo que indica no linealidad

Modelo 10

```
resettest(modelo10)

##
##  RESET test
##
## data:  modelo10
## RESET = 12.596, df1 = 2, df2 = 199, p-value = 7.065e-06
```

Rechazamos H_0 ya que $p\text{-value} = 7.065e-06 < \alpha = 0.04$ lo que indica que hay una especificación errónea en el modelo que indica no linealidad

Modelo 11

```
resettest(modelo11)

##
##  RESET test
##
```

```
## data:  modelo11
## RESET = 5.6813, df1 = 2, df2 = 199, p-value = 0.003986
```

Rechazamos H_0 ya que $p\text{-value} = 0.003986 < \alpha = 0.04$ lo que indica que hay una especificación errónea en el modelo que indica no linealidad

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

Para determinar los mejores modelos, se analizan cada uno en función de los siguientes criterios:

- Significancia del modelo y de los coeficientes: Un modelo significativo con coeficientes significativos es más confiable.
- Porcentaje de variabilidad explicada por el modelo (R^2): Cuanto mayor sea este porcentaje, mejor explicará el modelo la variabilidad de los datos.
- Cumplimiento de los supuestos del modelo lineal: Como la normalidad de los residuos, independencia, homocedasticidad, y linealidad.

Mejores modelos

El mejor modelo podríamos con Modelo 1: carheight + carwidth + convertible * Significancia del modelo: Sí. * Coeficientes significativos: Todos excepto carheight. * R^2 (variabilidad explicada): 62.38%. * Normalidad: No cumple. * Independencia: No cumple. * Homocedasticidad: Pruebas BP y GQ p-values = 0.07359 y 0.9476 (casi homocedástico). * Linealidad: No hay linealidad.

En segundo lugar está el modelo 3: carwidth + convertible * Significancia del modelo: Sí. * Coeficientes significativos: Todos. * R^2 (variabilidad explicada): 61.97%. * Normalidad: No cumple. * Independencia: No cumple. * Homocedasticidad: Pruebas BP y GQ p-values = 0.02686 y 0.9188 (homocedástico según GQ). * Linealidad: No hay linealidad.

Modelo 11: carwidth * convertible * Significancia del modelo: Sí. * Coeficientes significativos: Solo carwidth. * R^2 (variabilidad explicada): 61.98%. * Normalidad: No cumple. * Independencia: No cumple. * Homocedasticidad: Pruebas BP y GQ p-values = 0.03916 y 0.9225 (homocedástico según GQ). * Linealidad: No hay linealidad.

Modelo 7: carheight + carwidth * Significancia del modelo: Sí. * Coeficientes significativos: Todos. * R^2 (variabilidad explicada): 58.59%. * Normalidad: No cumple. * Independencia: No cumple. * Homocedasticidad: Pruebas BP y GQ p-values = 0.0999 y 0.815 (casi homocedástico). * Linealidad: No hay linealidad.

Modelos 2, 4, 5, 6, 8, 9, 10: No cumplen con la significancia de los coeficientes o tienen un R^2 demasiado bajo en comparación con los modelos seleccionados.

El modelo 1 es el mejor modelo ya que es el modelo con el mayor R^2 (62.38%), y a pesar de que el coeficiente de carheight no es significativo, es el modelo más robusto en términos de variabilidad explicada. Además, las pruebas de homocedasticidad

indican que es casi homocedástico. Este modelo es el mejor debido a su mayor R^2 y significancia del modelo.

En segundo lugar tenemos el modelo 3 ya que tiene un R^2 cercano al del Modelo 1 (61.97%) y todos sus coeficientes son significativos. Es más simple que el Modelo 1 pero sigue capturando una gran parte de la variabilidad del precio, con mejores resultados en homocedasticidad. Este modelo es el segundo mejor por su simplicidad y significancia de todos los coeficientes.

Y en tercer lugar el modelo 11, tiene un R^2 similar al Modelo 3 (61.98%) y solo un coeficiente significativo (carwidth). Aunque no todos los coeficientes son significativos, muestra un buen ajuste general y cumple con homocedasticidad. Este modelo es el tercero mejor, proporcionando un balance entre complejidad y variabilidad explicada.

3. Intervalos de predicción y confianza

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:

Calcula los intervalos para la variable Y

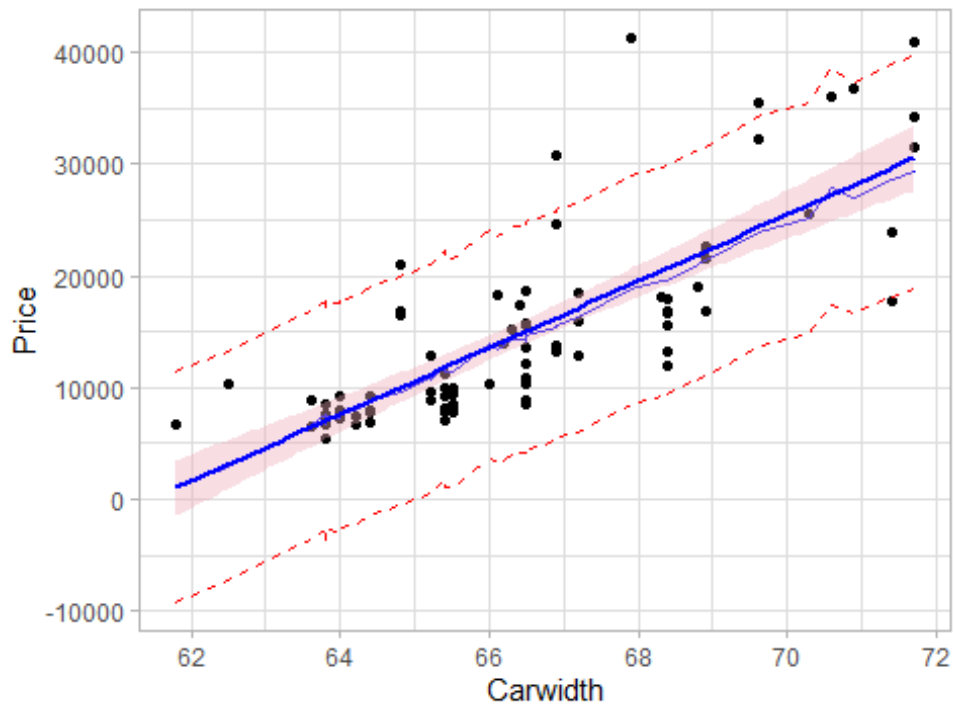
```
#Calcula los intervalos para la variable Y
Ip = predict(object = modelo1, interval = "prediction", level = 0.96)

## Warning in predict.lm(object = modelo1, interval = "prediction", level
= 0.96): predictions on current data refer to _future_ responses

datos1 = cbind(data, Ip)
autos_sedan = subset(datos1, datos1$carbody == "sedan")

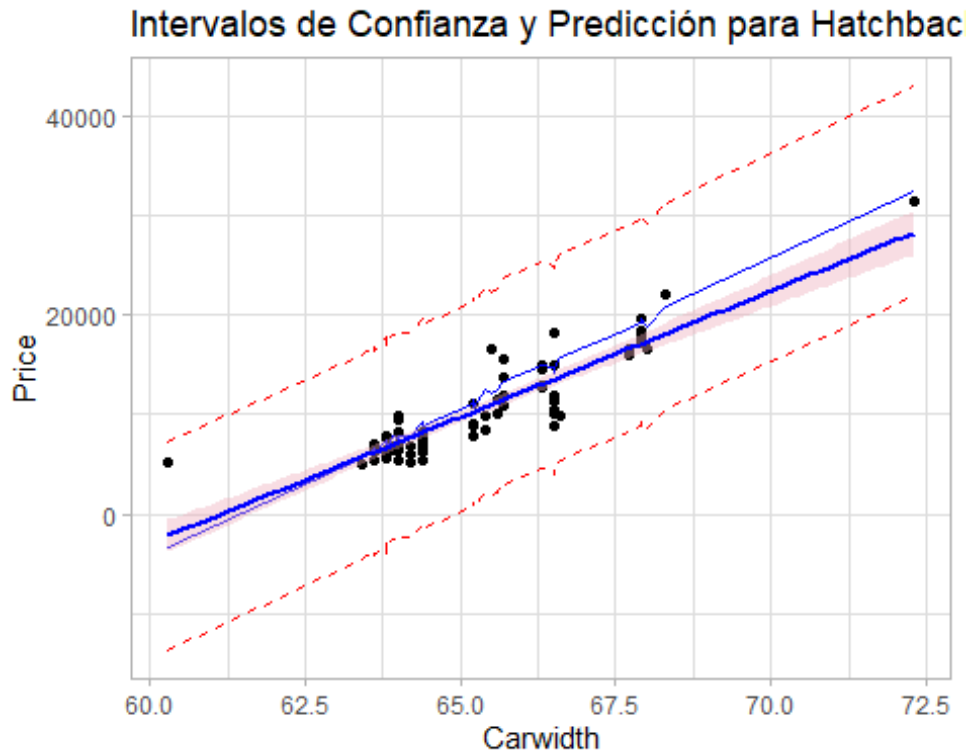
#Grafica por pares de variables numéricas
library(ggplot2)
ggplot(autos_sedan, aes(x = carwidth, y = price)) +
  geom_point() +
  geom_line(aes(y = fit), color = "blue") +
  geom_line(aes(y = lwr), color = "red", linetype = "dashed") +
  geom_line(aes(y = upr), color = "red", linetype = "dashed") +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, level = 0.96, col =
"blue", fill = "pink2") +
  theme_light() +
  labs(title = "Intervalos de Confianza y Predicción para Sedanes", x =
"Carwidth", y = "Price")
```

Intervalos de Confianza y Predicción para Sedanes



```
autos_hatch = subset(datos1, datos1$carbody == "hatchback")

ggplot(autos_hatch, aes(x = carwidth, y = price)) +
  geom_point() +
  geom_line(aes(y = fit), color = "blue") +
  geom_line(aes(y = lwr), color = "red", linetype = "dashed") +
  geom_line(aes(y = upr), color = "red", linetype = "dashed") +
  geom_smooth(method = lm, formula = y ~ x, se = TRUE, level = 0.96, col =
"blue", fill = "pink2") +
  theme_light() +
  labs(title = "Intervalos de Confianza y Predicción para Hatchbacks", x =
"Carwidth", y = "Price")
```

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 automovilística?

Multicolinealidad e Interacciones: Los modelos que incluyeron interacciones entre variables, como $\text{carheight} * \text{carwidth} * \text{convertible}$, no resultaron en modelos significativos ni mejoraron la explicación de la variabilidad del precio. Esto sugiere que las interacciones de segundo y tercer orden no aportan valor en la predicción del precio.

Significancia de las Variables: En modelos más simples como el Modelo 3 ($\text{carwidth} + \text{convertible}$), tanto carwidth como convertible resultaron ser significativos y explican el 61.97% de la variabilidad del precio. Esto indica que estas variables tienen un efecto importante y directo sobre el precio, y su agrupación es eficaz.

Incluir Variables Relevantes Únicamente: Algunas variables como carheight no son significativas en varios modelos. Por ejemplo, en el Modelo 1 ($\text{carheight} + \text{carwidth} + \text{convertible}$), carheight no resultó significativa. Esto sugiere que su inclusión podría no ser necesaria para todos los modelos, y que deberíamos enfocarnos en variables que muestran una influencia más fuerte en la predicción del precio.

Propuesta de Nueva Agrupación: Variables Relacionadas con el Tamaño del Auto: carwidth parece ser una variable muy influyente y debería mantenerse. carheight no

resultó significativa en algunos modelos, por lo que podríamos agregar otras dimensiones del auto (e.g., altura, ancho, longitud), tambien las variables podrían agruparse en función de características mecánicas (e.g., tamaño del motor, potencia)

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

```
summary(data)
```

```
##      symboling      CarName      fueltype
carbody
## Min.      :-2.0000   Length:205      Length:205      convertible:
6
## 1st Qu.: 0.0000   Class :character   Class :character   hardtop      :
8
## Median : 1.0000   Mode  :character   Mode  :character   hatchback
:70
## Mean    : 0.8341                                     sedan
:96
## 3rd Qu.: 2.0000                                     wagon
:25
## Max.     : 3.0000
## drivewheel      enginelocation      wheelbase      carlength
## Length:205      Length:205      Min.      : 86.60   Min.      :141.1
## Class :character   Class :character   1st Qu.: 94.50   1st Qu.:166.3
## Mode  :character   Mode  :character   Median : 97.00   Median :173.2
##                                     Mean    : 98.76   Mean     :174.0
##                                     3rd Qu.:102.40   3rd Qu.:183.1
##                                     Max.     :120.90   Max.      :208.1
##
##      carwidth      carheight      curbweight      enginetype
## Min.      :60.30   Min.      :47.80   Min.      :1488   Length:205
## 1st Qu.:64.10   1st Qu.:52.00   1st Qu.:2145   Class :character
## Median :65.50   Median :54.10   Median :2414   Mode  :character
## Mean    :65.91   Mean     :53.72   Mean     :2556
## 3rd Qu.:66.90   3rd Qu.:55.50   3rd Qu.:2935
## Max.     :72.30   Max.     :59.80   Max.     :4066
## cylindernumber      enginesize      stroke      compressionratio
## Length:205      Min.      : 61.0   Min.      :2.070   Min.      : 7.00
## Class :character   1st Qu.: 97.0   1st Qu.:3.110   1st Qu.: 8.60
## Mode  :character   Median :120.0   Median :3.290   Median : 9.00
##                                     Mean    :126.9   Mean     :3.255   Mean     :10.14
##                                     3rd Qu.:141.0   3rd Qu.:3.410   3rd Qu.: 9.40
##                                     Max.     :326.0   Max.      :4.170   Max.      :23.00
##
##      horsepower      peakrpm      citympg      highwaympg
price
## Min.      : 48.0   Min.      :4150   Min.      :13.00   Min.      :16.00   Min.
: 5118
## 1st Qu.: 70.0   1st Qu.:4800   1st Qu.:19.00   1st Qu.:25.00   1st
Qu.: 7788
## Median : 95.0   Median :5200   Median :24.00   Median :30.00   Median
```

```

:10295
## Mean :104.1 Mean :5125 Mean :25.22 Mean :30.75 Mean
:13277
## 3rd Qu.:116.0 3rd Qu.:5500 3rd Qu.:30.00 3rd Qu.:34.00 3rd
Qu.:16503
## Max. :288.0 Max. :6600 Max. :49.00 Max. :54.00 Max.
:45400
## convertible
## Min. :0.00000
## 1st Qu.:0.00000
## Median :0.00000
## Mean :0.02927
## 3rd Qu.:0.00000
## Max. :1.00000

cor(data[, c("carheight", "carwidth", "price", "carlength",
"curbweight")], use = "complete.obs")

## carheight carwidth price carlength curbweight
## carheight 1.0000000 0.2792103 0.1193362 0.4910295 0.2955717
## carwidth 0.2792103 1.0000000 0.7593253 0.8411183 0.8670325
## price 0.1193362 0.7593253 1.0000000 0.6829200 0.8353049
## carlength 0.4910295 0.8411183 0.6829200 1.0000000 0.8777285
## curbweight 0.2955717 0.8670325 0.8353049 0.8777285 1.0000000

```