Act Integradora 2

2024-09-06

```
library(car)
```

```
## Loading required package: carData
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:car':
##
##     recode
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data = read.csv("documents/precios_autos.csv")
head(data)
```

```
##   symboling                  CarName fueltype     carbody drivewheel
## 1         3       alfa-romero giulia      gas convertible        rwd
## 2         3      alfa-romero stelvio      gas convertible        rwd
## 3         1 alfa-romero Quadrifoglio      gas   hatchback        rwd
## 4         2              audi 100 ls      gas       sedan        fwd
## 5         2               audi 100ls      gas       sedan        4wd
## 6         2                 audi fox      gas       sedan        fwd
##   enginelocation wheelbase carlength carwidth carheight curbweight enginetype
## 1          front      88.6     168.8     64.1      48.8       2548       dohc
## 2          front      88.6     168.8     64.1      48.8       2548       dohc
## 3          front      94.5     171.2     65.5      52.4       2823       ohcv
## 4          front      99.8     176.6     66.2      54.3       2337        ohc
## 5          front      99.4     176.6     66.4      54.3       2824        ohc
## 6          front      99.8     177.3     66.3      53.1       2507        ohc
##   cylindernumber enginesize stroke compressionratio horsepower peakrpm citympg
## 1           four        130   2.68              9.0        111    5000      21
## 2           four        130   2.68              9.0        111    5000      21
## 3            six        152   3.47              9.0        154    5000      19
## 4           four        109   3.40             10.0        102    5500      24
## 5           five        136   3.40              8.0        115    5500      18
## 6           five        136   3.40              8.5        110    5500      19
##   highwaympg price
## 1         27 13495
## 2         27 16500
## 3         26 16500
## 4         30 13950
## 5         22 17450
## 6         25 15250
```

```
variables_cuantitativas = select(data, carheight, carwidth, price)


medidas = data.frame(
  Variable = names(variables_cuantitativas),
  Media = sapply(variables_cuantitativas, mean, na.rm = TRUE),
  Desviacion_Estandar = sapply(variables_cuantitativas, sd, na.rm = TRUE),
  Mediana = sapply(variables_cuantitativas, median, na.rm = TRUE),
  Q1 = sapply(variables_cuantitativas, function(x) quantile(x, 0.25, na.rm = TRUE)),
  Q3 = sapply(variables_cuantitativas, function(x) quantile(x, 0.75, na.rm = TRUE)),
  Min = sapply(variables_cuantitativas, min, na.rm = TRUE),
  Max = sapply(variables_cuantitativas, max, na.rm = TRUE)
)

print(medidas)
```

```
##           Variable       Media Desviacion_Estandar Mediana     Q1      Q3
## carheight carheight    53.72488           2.443522    54.1   52.0    55.5
## carwidth   carwidth    65.90780           2.145204    65.5   64.1    66.9
## price         price 13276.71057        7988.852332 10295.0 7788.0 16503.0
##              Min     Max
## carheight   47.8    59.8
## carwidth    60.3    72.3
## price     5118.0 45400.0
```

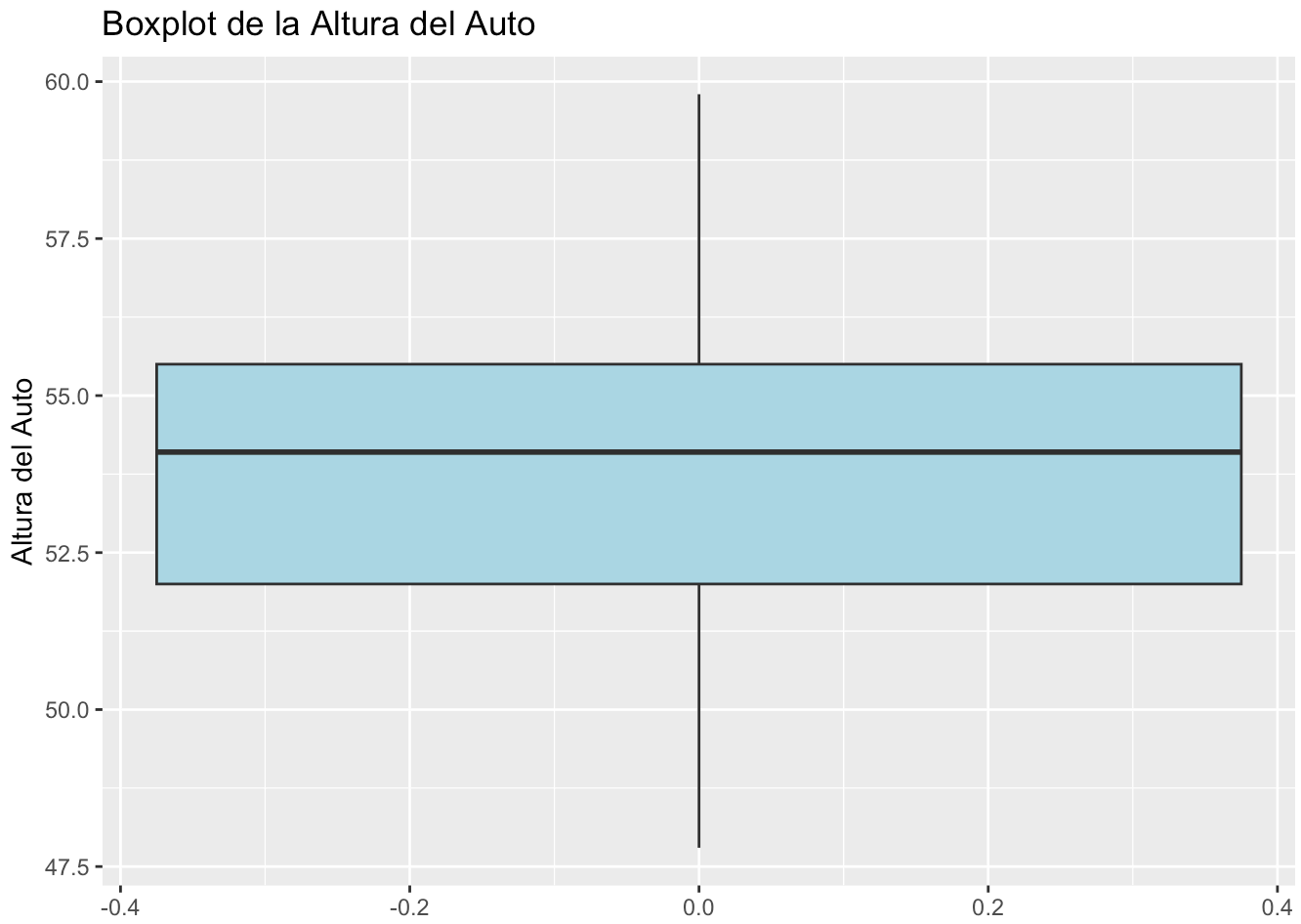
```
frecuencias = summarize(
  group_by(data, carbody),
  Frecuencia_Absoluta = n(),
  Frecuencia_Relativa = n() / nrow(data)
)

print(frecuencias)
```

```
## # A tibble: 5 × 3
##   carbody     Frecuencia_Absoluta Frecuencia_Relativa
##   <chr>                     <int>               <dbl>
## 1 convertible                   6              0.0293
## 2 hardtop                       8              0.0390
## 3 hatchback                    70              0.341
## 4 sedan                        96              0.468
## 5 wagon                        25              0.122
```

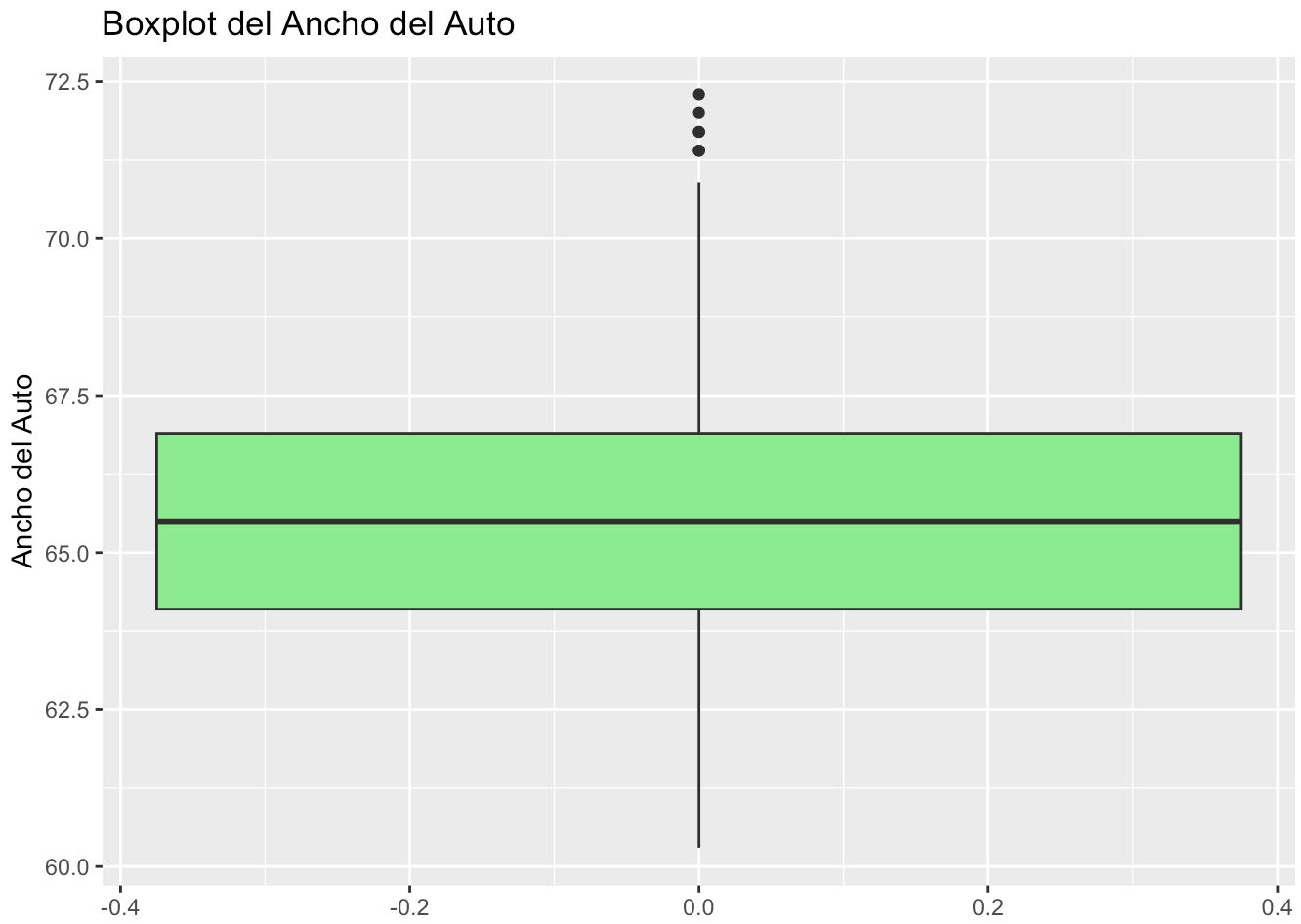
```
matriz_corr = cor(variables_cuantitativas)

print(matriz_corr)
```

```
##           carheight  carwidth     price
## carheight 1.0000000 0.2792103 0.1193362
## carwidth  0.2792103 1.0000000 0.7593253
## price     0.1193362 0.7593253 1.0000000
```

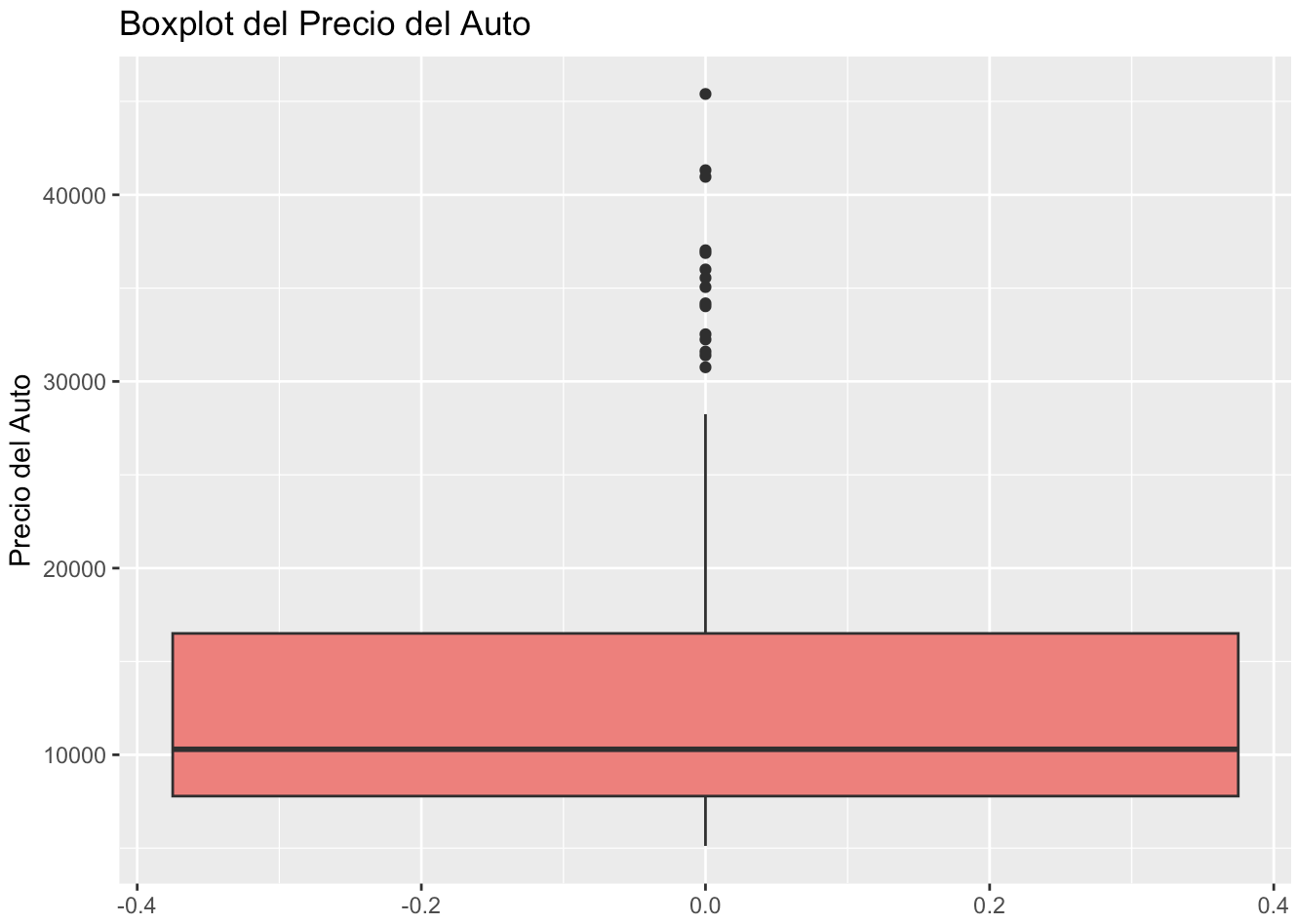
```
ggplot(data, aes(y = carheight)) +
  geom_boxplot(fill = "lightblue") +
  ggtitle("Boxplot de la Altura del Auto") +
  ylab("Altura del Auto")
```



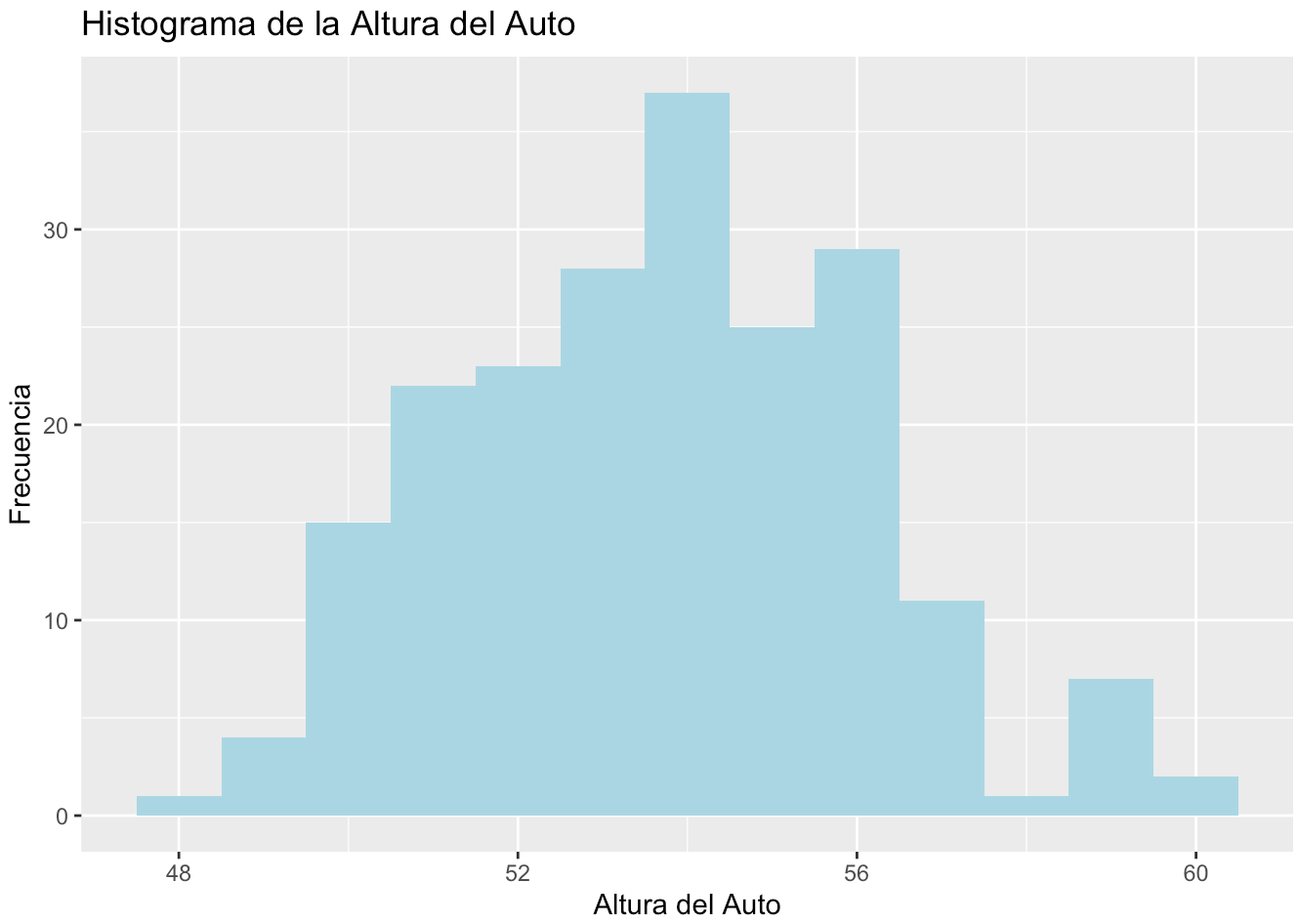

```
ggplot(data, aes(y = carwidth)) +
  geom_boxplot(fill = "lightgreen") +
  ggtitle("Boxplot del Ancho del Auto") +
  ylab("Ancho del Auto")
```



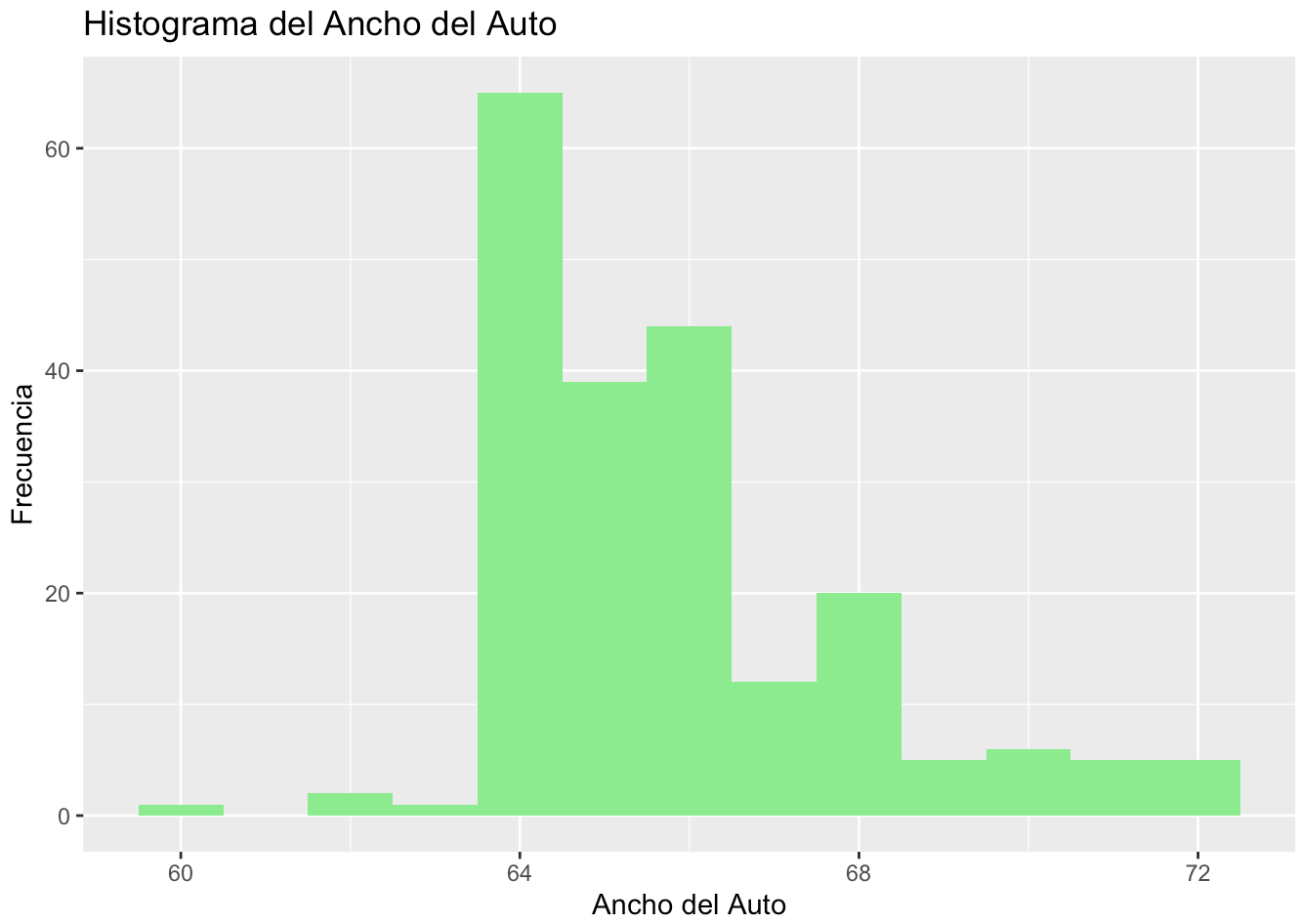

```
ggplot(data, aes(y = price)) +
  geom_boxplot(fill = "lightcoral") +
  ggtitle("Boxplot del Precio del Auto") +
  ylab("Precio del Auto")
```



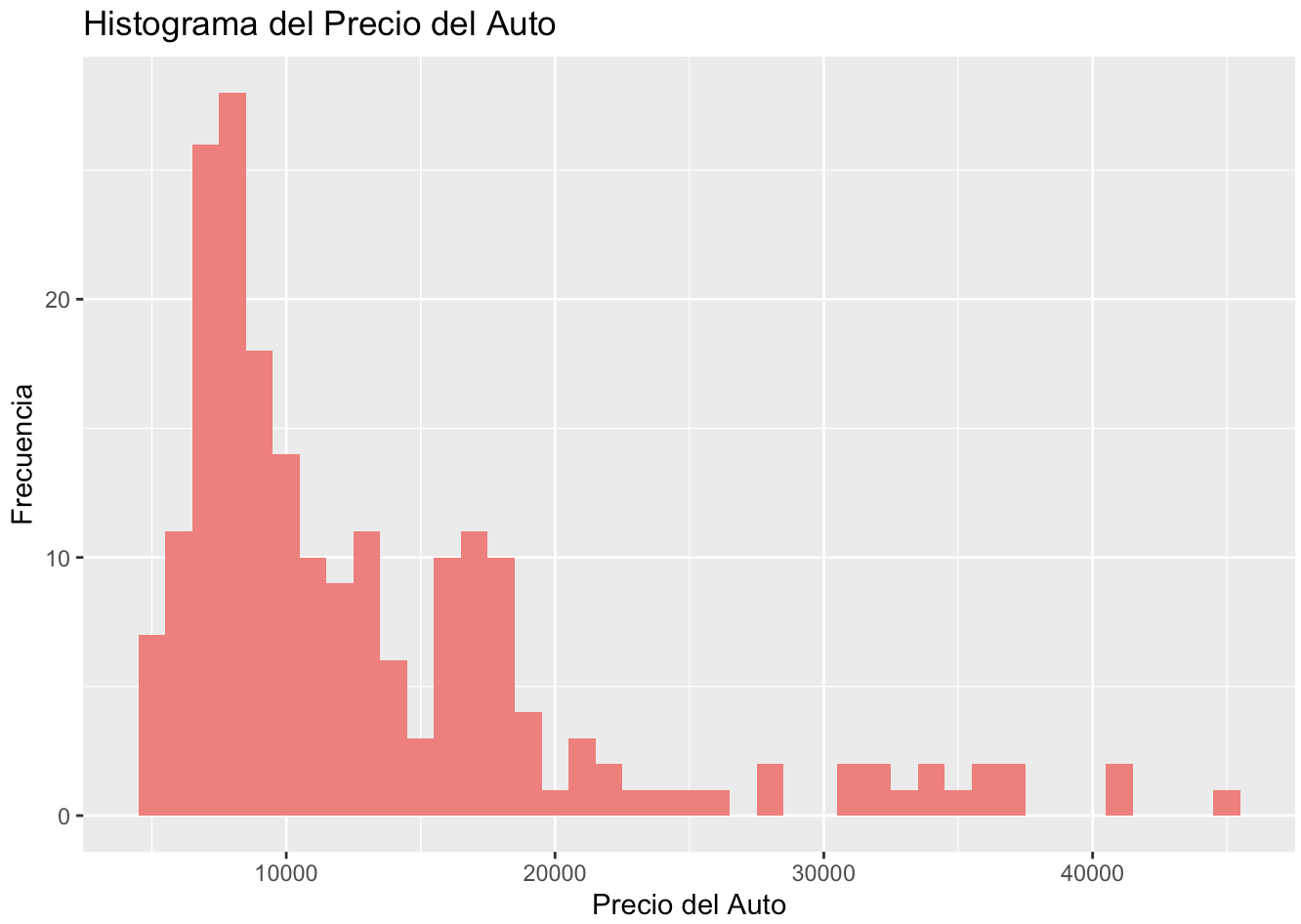

```
ggplot(data, aes(x = carheight)) +
  geom_histogram(binwidth = 1, fill = "lightblue") +
  ggtitle("Histograma de la Altura del Auto") +
  xlab("Altura del Auto") +
  ylab("Frecuencia")
```



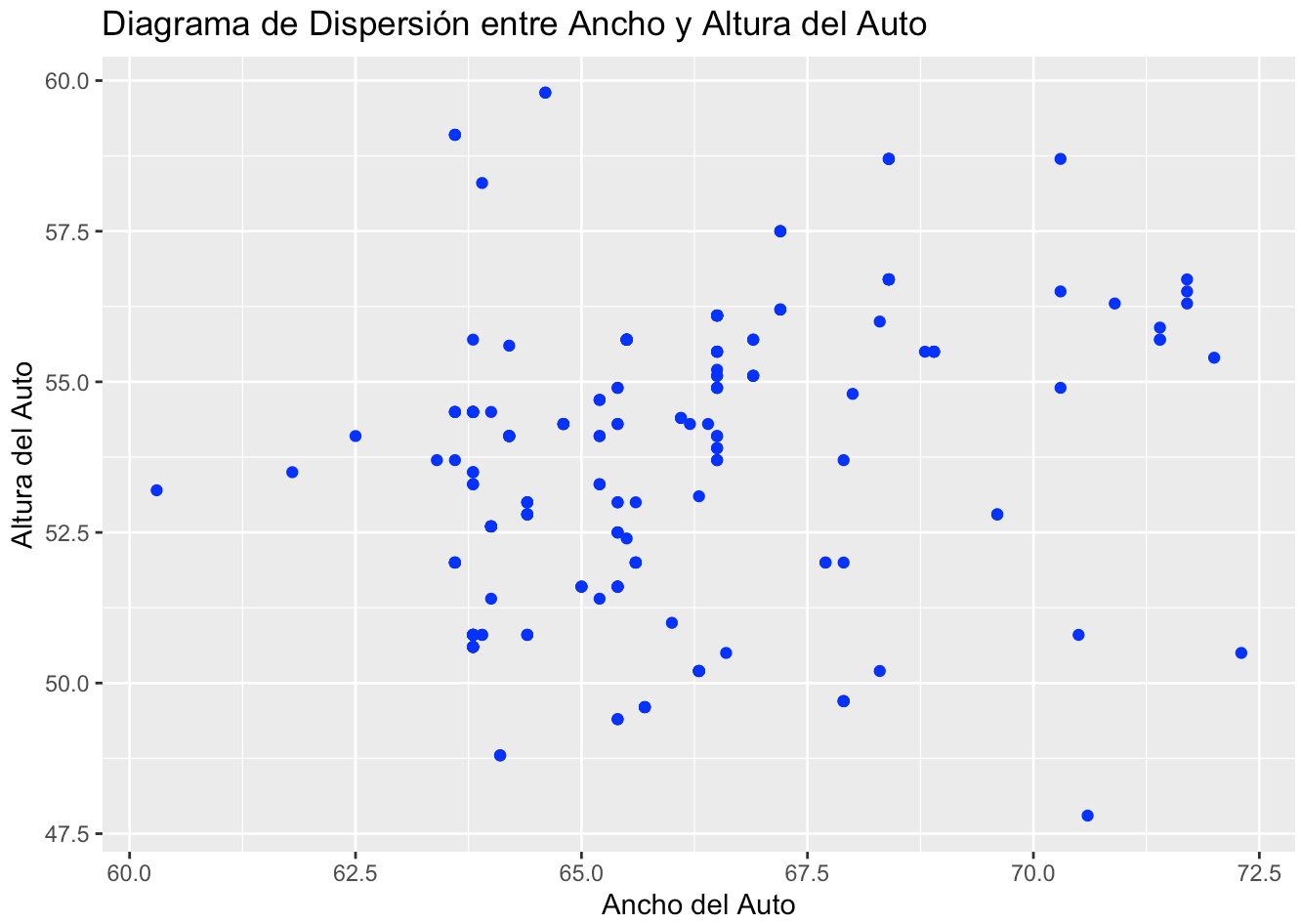

```
ggplot(data, aes(x = carwidth)) +
  geom_histogram(binwidth = 1, fill = "lightgreen") +
  ggtitle("Histograma del Ancho del Auto") +
  xlab("Ancho del Auto") +
  ylab("Frecuencia")
```



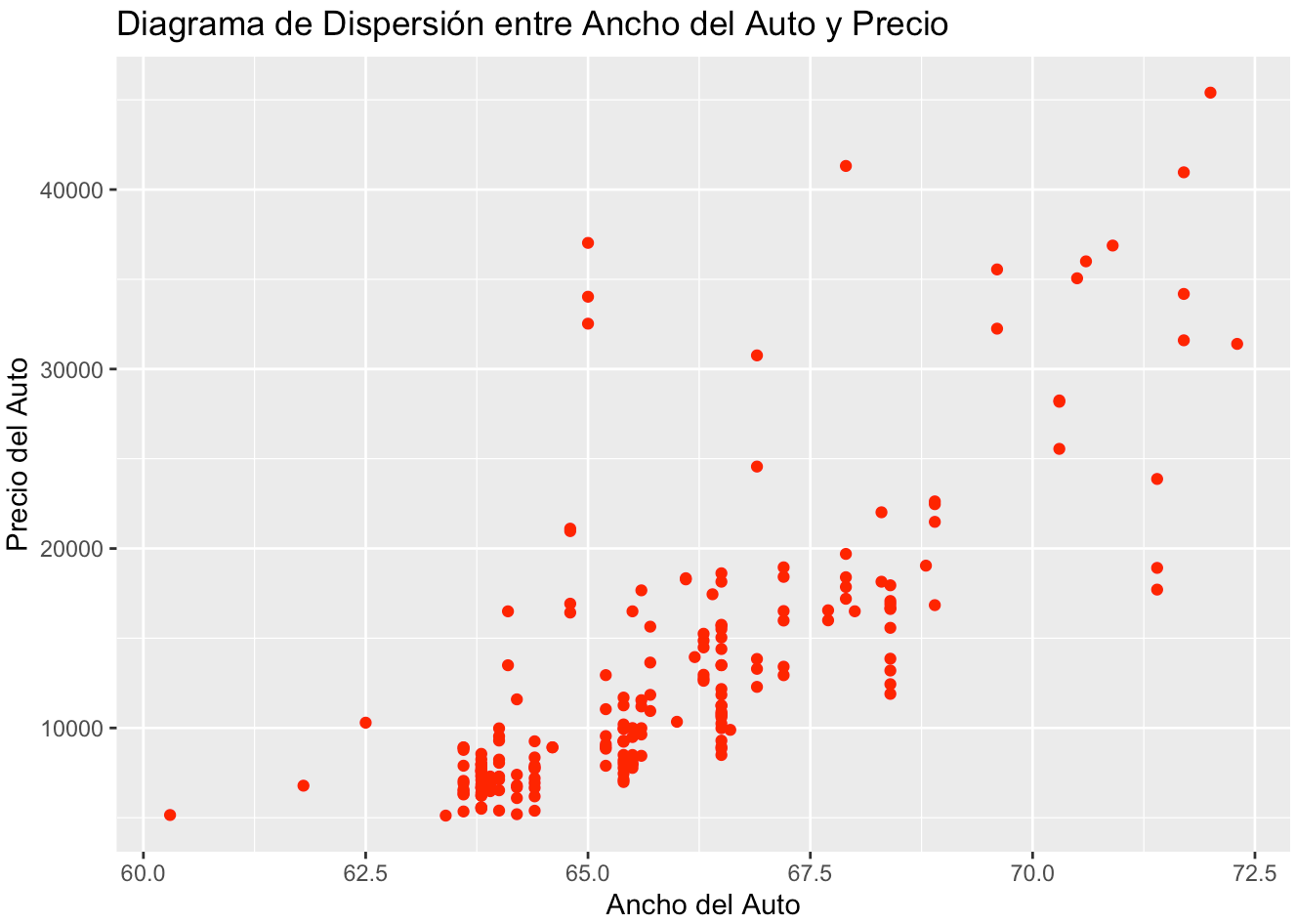

```
ggplot(data, aes(x = price)) +
  geom_histogram(binwidth = 1000, fill = "lightcoral") +
  ggtitle("Histograma del Precio del Auto") +
  xlab("Precio del Auto") +
  ylab("Frecuencia")
```



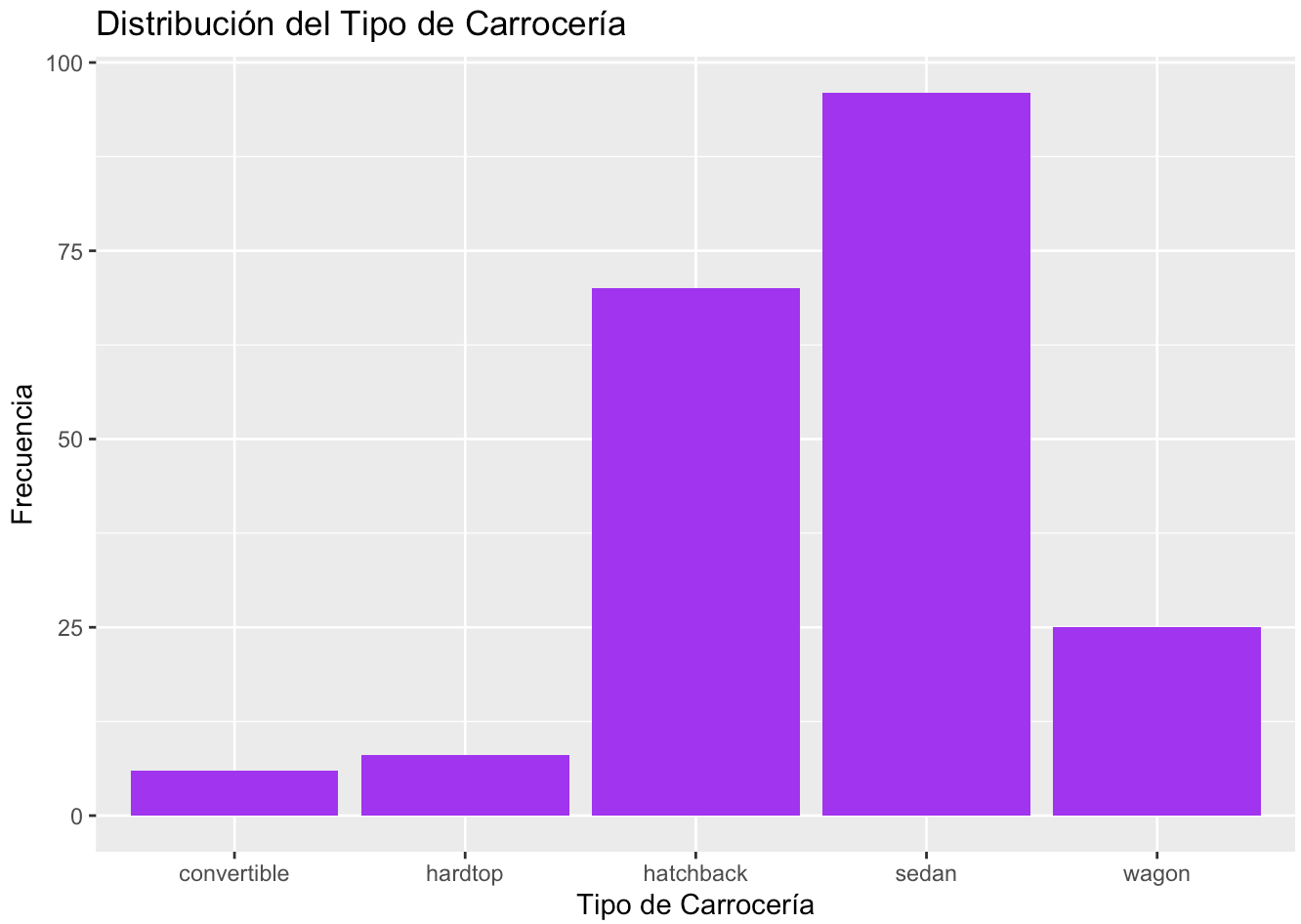

```
ggplot(data, aes(x = carwidth, y = carheight)) +
  geom_point(color = "blue") +
  ggtitle("Diagrama de Dispersión entre Ancho y Altura del Auto") +
  xlab("Ancho del Auto") +
  ylab("Altura del Auto")
```



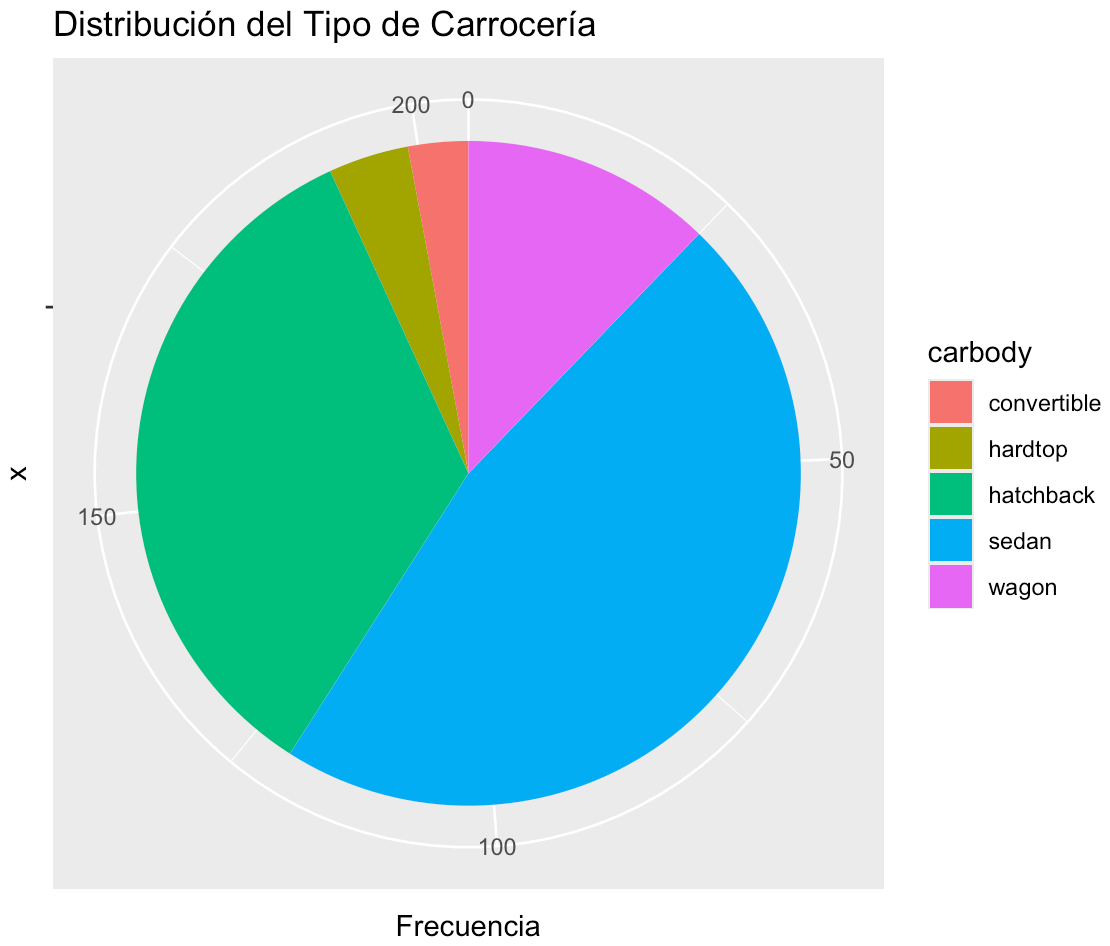

```
ggplot(data, aes(x = carwidth, y = price)) +
  geom_point(color = "red") +
  ggtitle("Diagrama de Dispersión entre Ancho del Auto y Precio") +
  xlab("Ancho del Auto") +
  ylab("Precio del Auto")
```



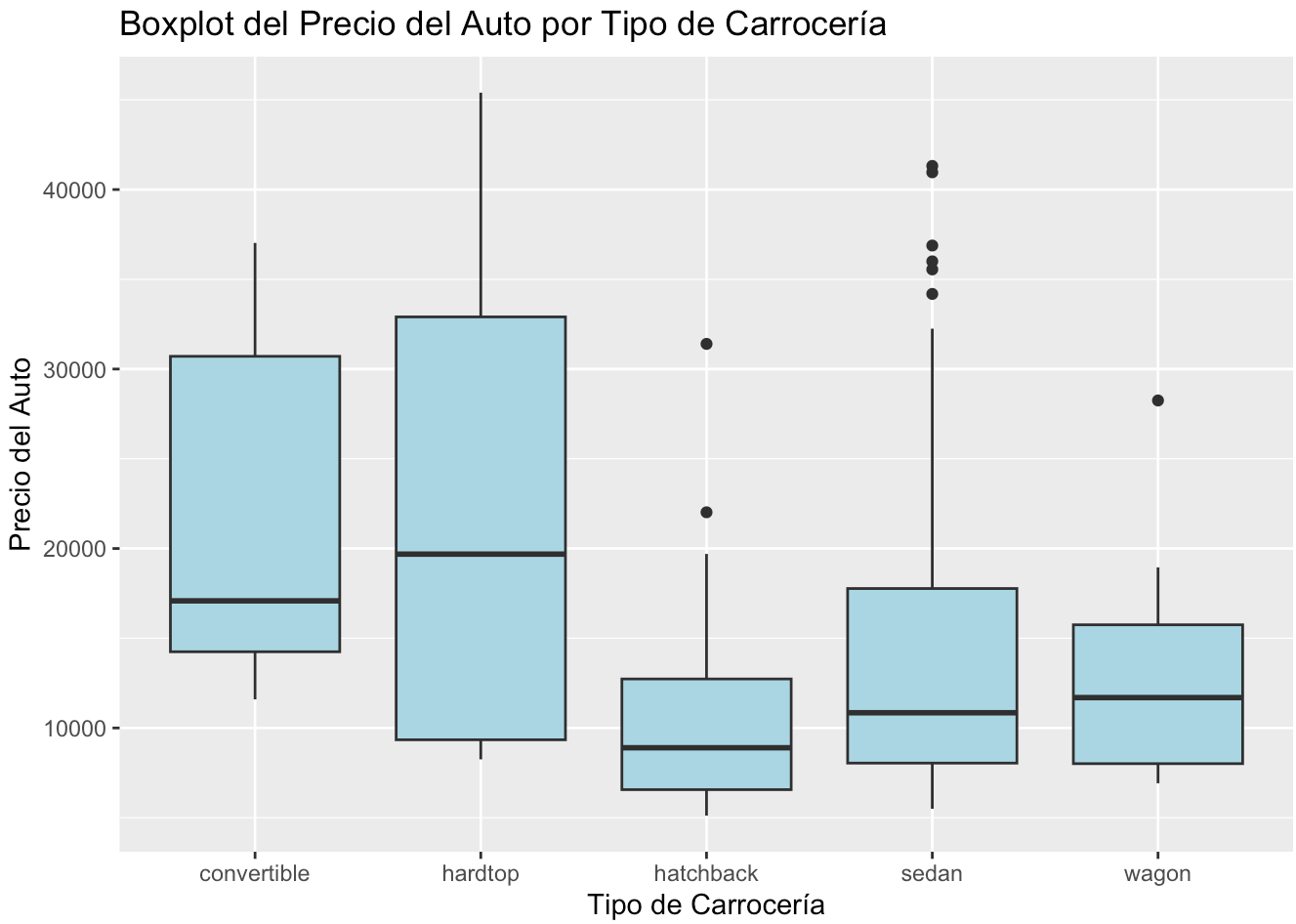

```
ggplot(data, aes(x = carbody)) +
  geom_bar(fill = "purple") +
  ggtitle("Distribución del Tipo de Carrocería") +
  xlab("Tipo de Carrocería") +
  ylab("Frecuencia")
```




```
frecuencias_carbody = count(data, carbody)

ggplot(frecuencias_carbody, aes(x = "", y = n, fill = carbody)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0) +
  ggtitle("Distribución del Tipo de Carrocería") +
  ylab("Frecuencia")
```




```
ggplot(data, aes(x = carbody, y = price)) +
  geom_boxplot(fill = "lightblue") +
  ggtitle("Boxplot del Precio del Auto por Tipo de Carrocería") +
  xlab("Tipo de Carrocería") +
  ylab("Precio del Auto")
```




```
modelo1 = lm(price ~ carheight + carwidth + carbody, data = data)

summary(modelo1)
```

```
##
## Call:
## lm(formula = price ~ carheight + carwidth + carbody, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11103.9  -2404.6   -657.1   1430.6  22217.3
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -150934.9    12535.9 -12.040  < 2e-16 ***
## carheight           -225.0      177.7  -1.266    0.207
## carwidth            2811.7      161.7  17.388  < 2e-16 ***
## carbodyhardtop     -2256.9     2554.2  -0.884    0.378
## carbodyhatchback  -10416.6     2005.7  -5.194 5.10e-07 ***
## carbodysedan       -8796.5     2042.1  -4.307 2.60e-05 ***
## carbodywagon      -10218.4     2328.8  -4.388 1.86e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4703 on 198 degrees of freedom
## Multiple R-squared:  0.6636, Adjusted R-squared:  0.6534
## F-statistic:  65.1 on 6 and 198 DF,  p-value: < 2.2e-16
```

```
modelo2 = lm(price ~ carheight * carwidth + carbody, data = data)

summary(modelo2)
```

```
##
## Call:
## lm(formula = price ~ carheight * carwidth + carbody, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11220.7  -2458.1   -563.4   1382.6  22008.4
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -436072.67  221003.36  -1.973   0.0499 *
## carheight             5043.33    4080.68   1.236   0.2180
## carwidth              7117.82    3336.14   2.134   0.0341 *
## carbodyhardtop       -2157.81    2551.05  -0.846   0.3987
## carbodyhatchback    -10424.67    2002.28  -5.206 4.82e-07 ***
## carbodysedan         -8774.32    2038.77  -4.304 2.64e-05 ***
## carbodywagon        -10319.42    2326.15  -4.436 1.52e-05 ***
## carheight:carwidth     -79.53      61.54  -1.292   0.1978
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4695 on 197 degrees of freedom
## Multiple R-squared:  0.6664, Adjusted R-squared:  0.6546
## F-statistic: 56.23 on 7 and 197 DF,  p-value: < 2.2e-16
```

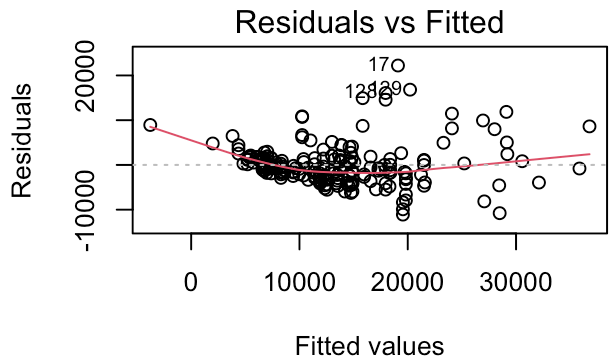
```
r2m1 = summary(modelo1)$r.squared
r2m1a = summary(modelo1)$adj.r.squared

r2m2 = summary(modelo2)$r.squared
r2m2a = summary(modelo2)$adj.r.squared

cat("Modelo 1 - R^2:", r2m1, "R^2 Ajustado:", r2m1a, "\n")
```

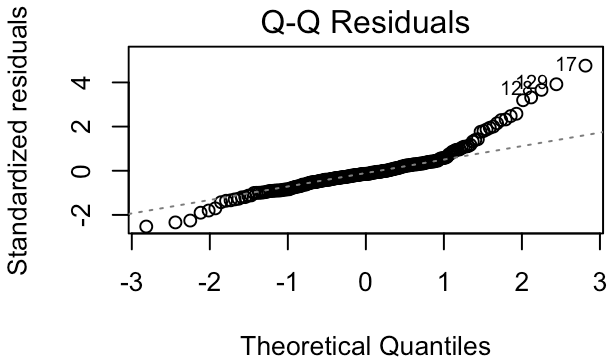
```
## Modelo 1 - R^2: 0.6636217 R^2 Ajustado: 0.6534284
```

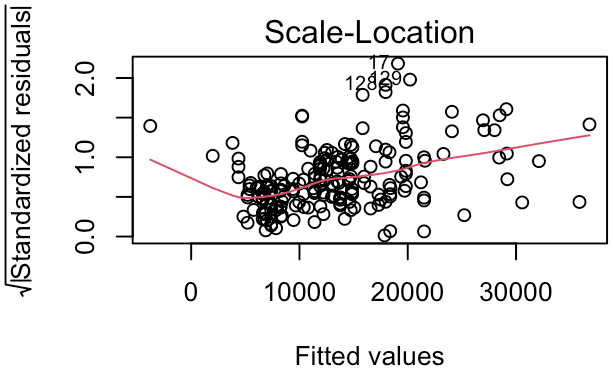
```
cat("Modelo 2 - R^2:", r2m2, "R^2 Ajustado:", r2m2a, "\n")
```

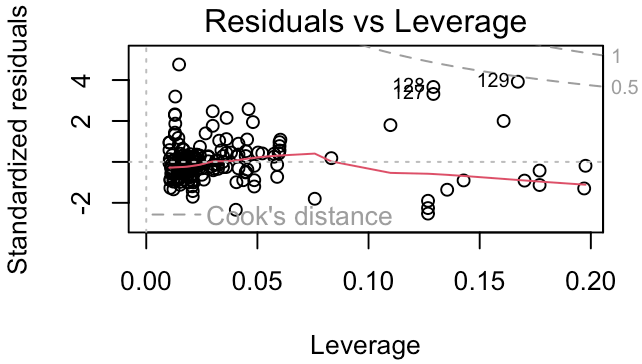
```
## Modelo 2 - R^2: 0.6664492 R^2 Ajustado: 0.6545972
```

```
par(mfrow = c(2, 2))
plot(modelo1)
```

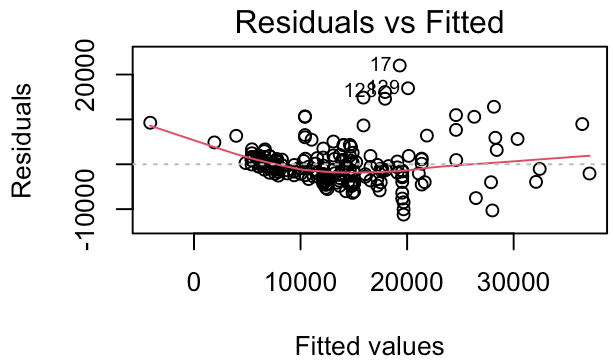


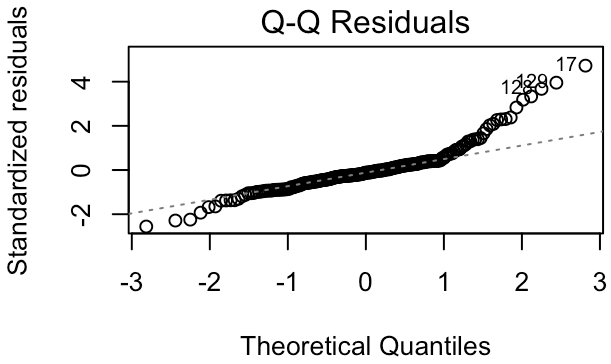


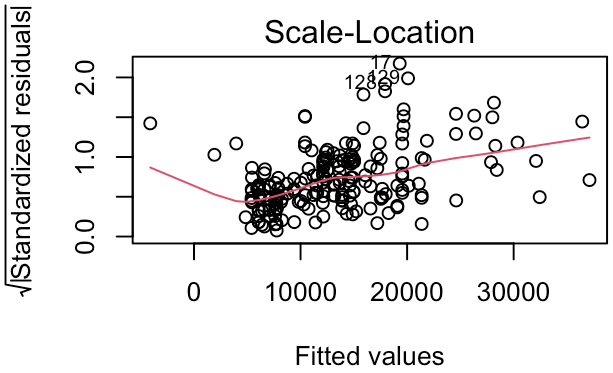


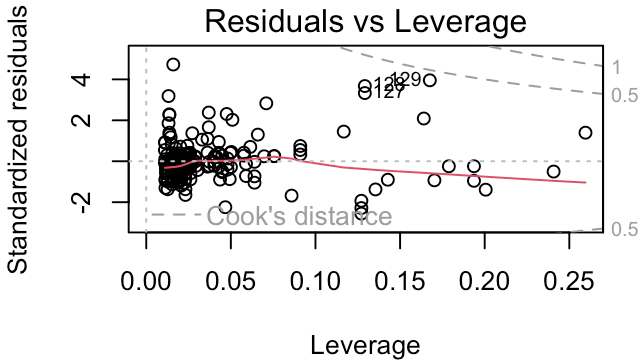



```
par(mfrow = c(2, 2))
plot(modelo2)
```









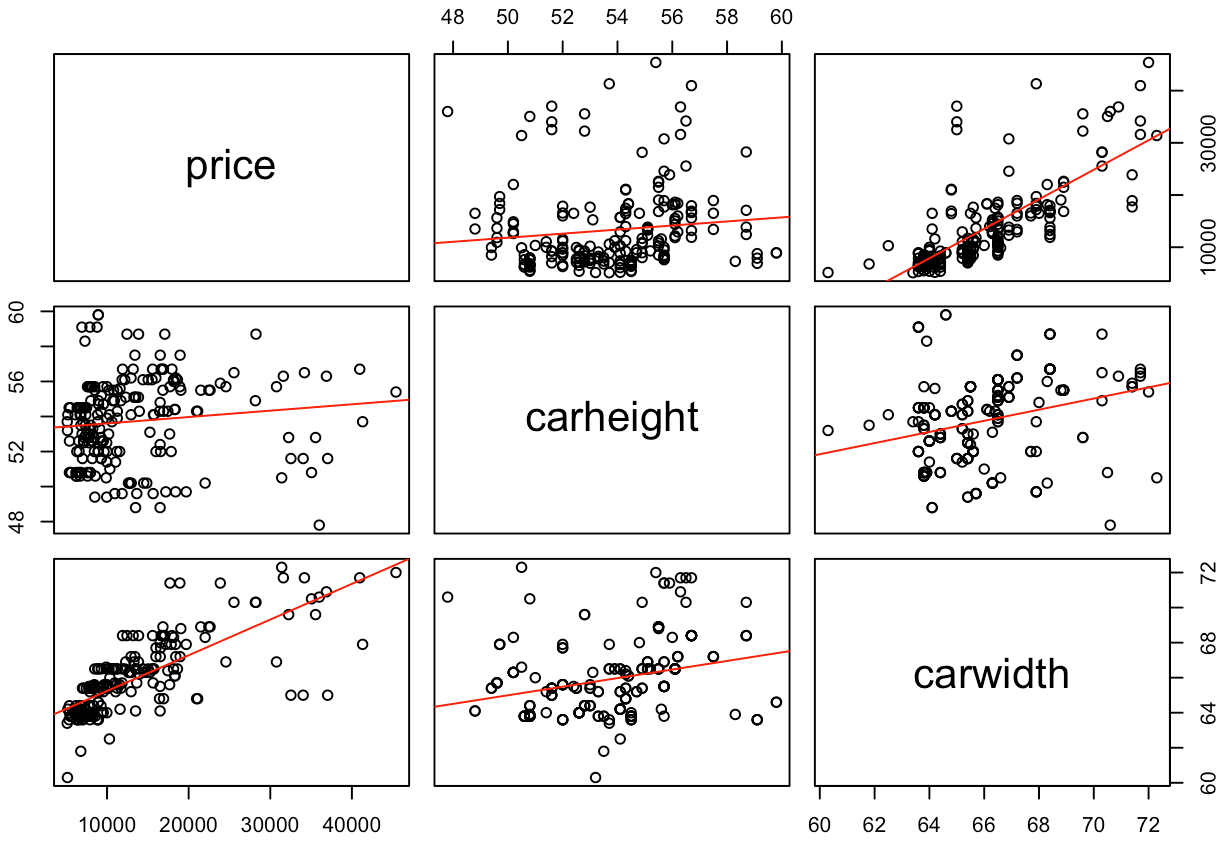

```
p_value_modelo1 = summary(modelo1)$fstatistic[1]
df1 = summary(modelo1)$fstatistic[2]
df2 = summary(modelo1)$fstatistic[3]

p_value_global = pf(p_value_modelo1, df1, df2, lower.tail = FALSE)

cat("Valor p global del modelo 1:", p_value_global, "\n")
```

```
## Valor p global del modelo 1: 3.217377e-44
```

```
coef_m1 = summary(modelo1)$coefficients

coef_m1
```

```
##                     Estimate Std. Error     t value     Pr(>|t|)
## (Intercept)      -150934.8809 12535.9431 -12.0401696 2.075736e-25
## carheight           -225.0242   177.6798  -1.2664588 2.068367e-01
## carwidth            2811.6766   161.7043  17.3877716 1.009475e-41
## carbodyhardtop     -2256.8996  2554.2114  -0.8835994 3.779841e-01
## carbodyhatchback  -10416.6438  2005.6575  -5.1936303 5.102059e-07
## carbodysedan       -8796.4608  2042.1413  -4.3074692 2.598115e-05
## carbodywagon      -10218.4330  2328.7718  -4.3879065 1.859349e-05
```

```
pairs(data[, c("price", "carheight", "carwidth")],
      panel = function(x, y) {
        points(x, y)
        abline(lm(y ~ x), col = "red")
      })
```



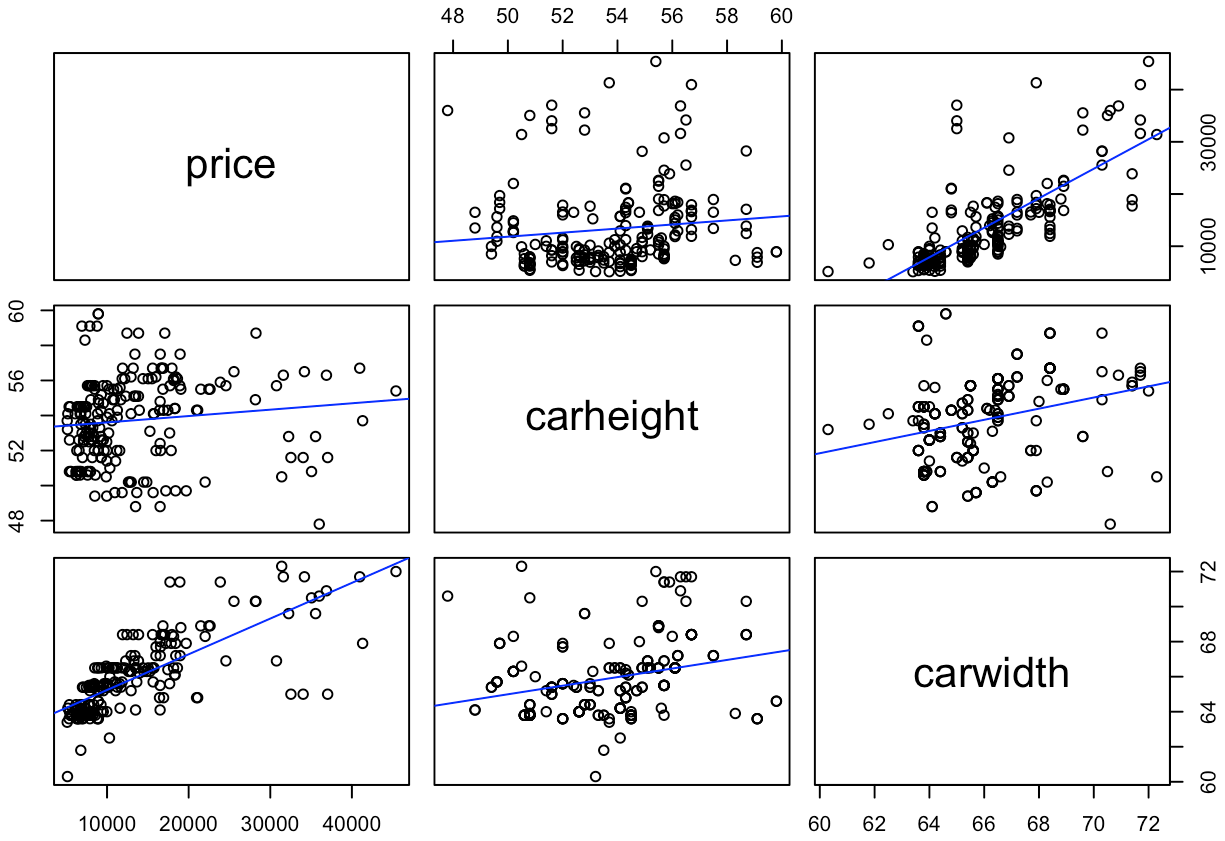

```
p_value_modelo2 = summary(modelo2)$fstatistic[1]
df1_2 = summary(modelo2)$fstatistic[2]
df2_2 = summary(modelo2)$fstatistic[3]

p_value_global_2 = pf(p_value_modelo2, df1_2, df2_2, lower.tail = FALSE)

cat("Valor p global del modelo 2:", p_value_global_2, "\n")
```

```
## Valor p global del modelo 2: 1.195226e-43
```

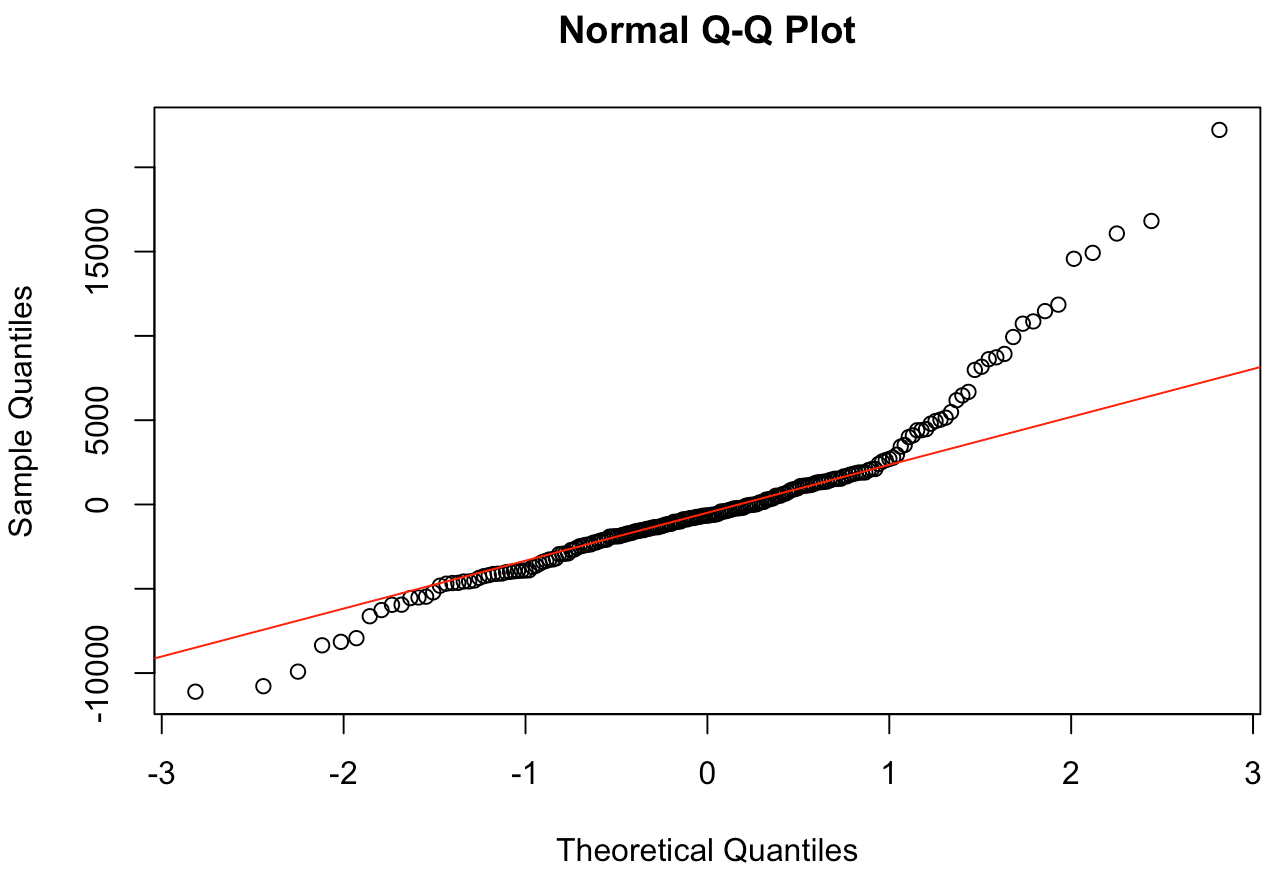
```
coef_m2 = summary(modelo2)$coefficients

coef_m2
```

```
##                          Estimate   Std. Error    t value     Pr(>|t|)
## (Intercept)        -436072.67110 221003.35964 -1.9731495 4.987698e-02
## carheight             5043.33048   4080.67828  1.2359049 2.179656e-01
## carwidth              7117.81533   3336.13580  2.1335508 3.411645e-02
## carbodyhardtop       -2157.81061   2551.05368 -0.8458507 3.986622e-01
## carbodyhatchback    -10424.66690   2002.28252 -5.2063916 4.823586e-07
## carbodysedan         -8774.32495   2038.76708 -4.3037407 2.644138e-05
## carbodywagon        -10319.41939   2326.15494 -4.4362562 1.520956e-05
## carheight:carwidth      -79.52676     61.54035 -1.2922702 1.977774e-01
```

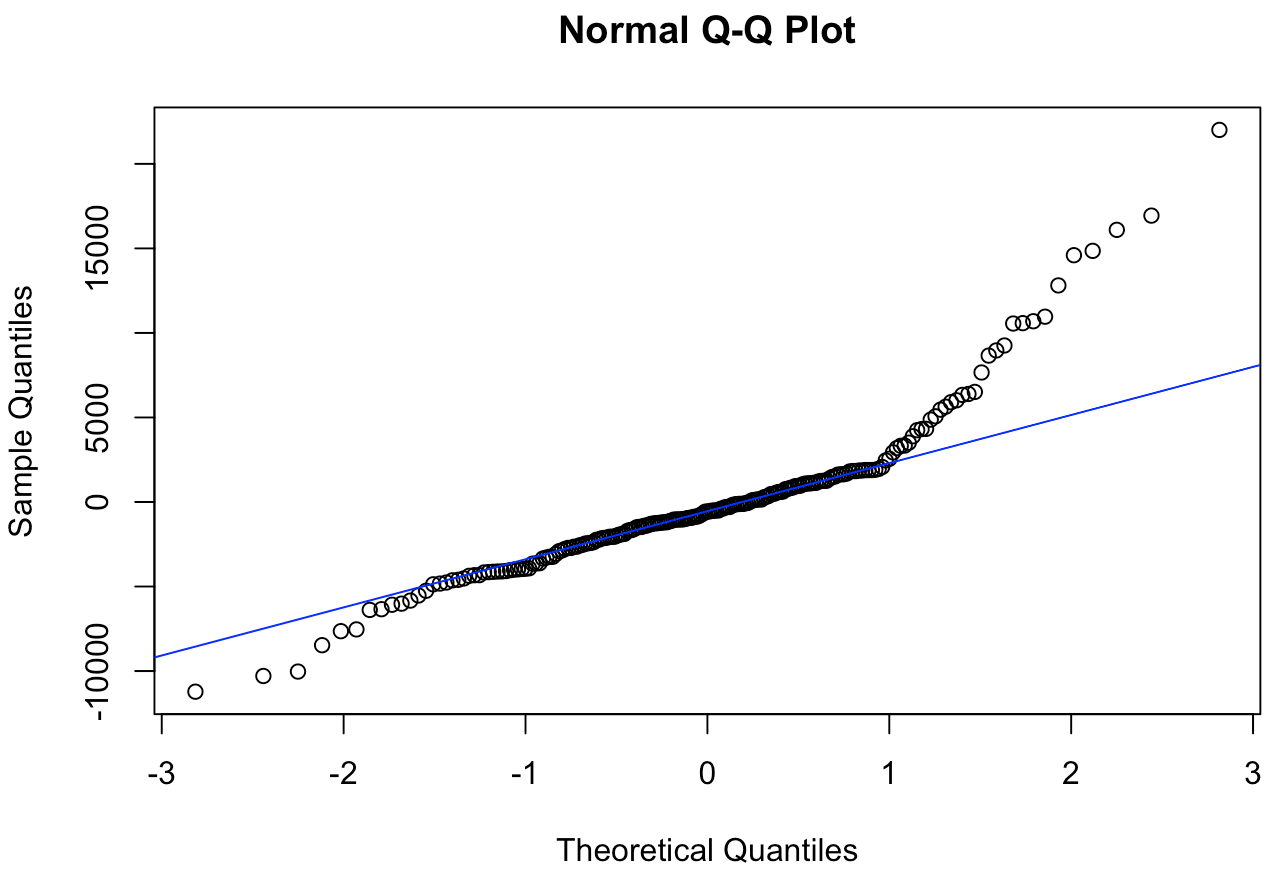
```
pairs(data[, c("price", "carheight", "carwidth")],
      panel = function(x, y) {
        points(x, y)
        abline(lm(y ~ x), col = "blue")
      })
```




```
qqnorm(resid(modelo1))
qqline(resid(modelo1), col = "red")
```




```
qqnorm(resid(modelo2))
qqline(resid(modelo2), col = "blue")
```




```
shapirom1 = shapiro.test(resid(modelo1))
shapirom1
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(modelo1)
## W = 0.89026, p-value = 4.299e-11
```

```
shapirom2 = shapiro.test(resid(modelo2))
shapirom2
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(modelo2)
## W = 0.88702, p-value = 2.748e-11
```

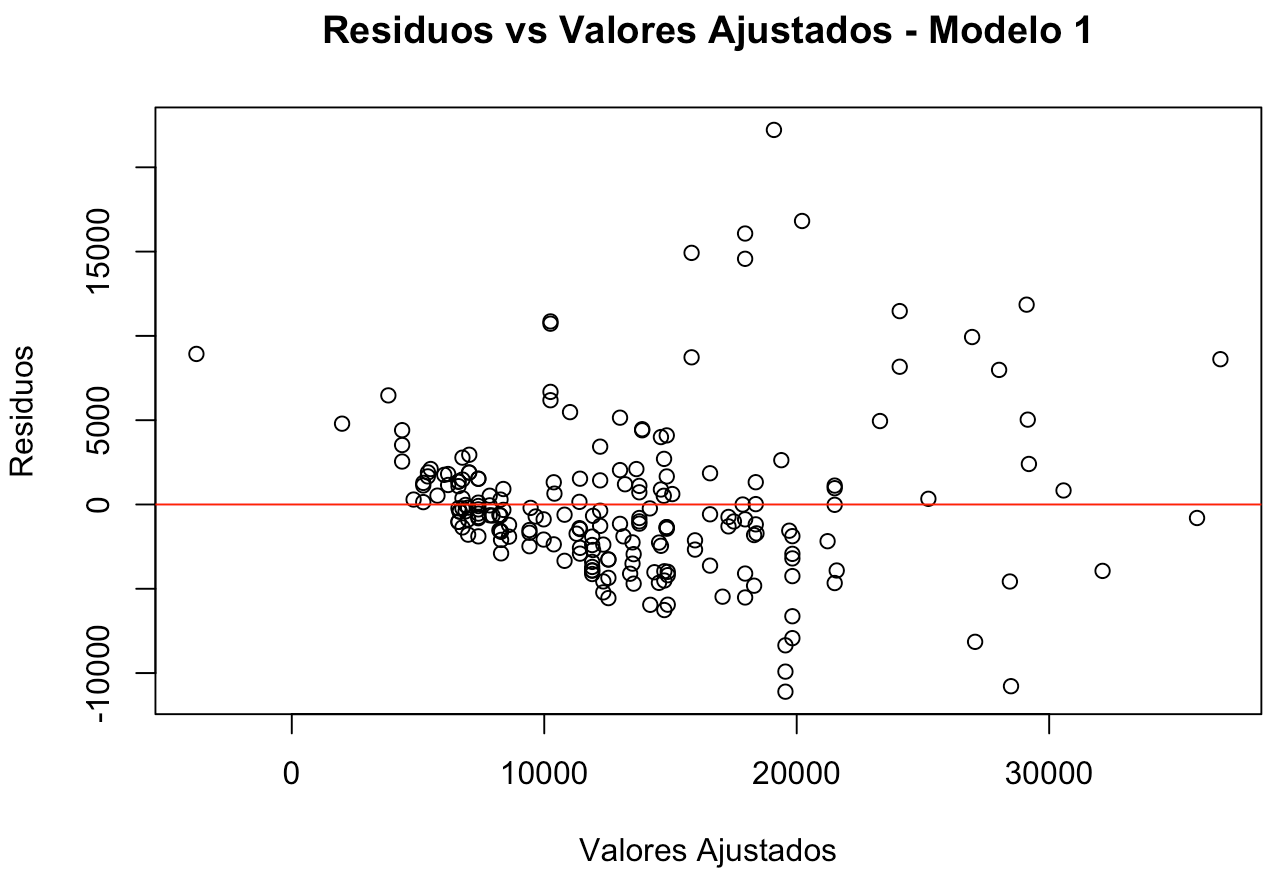
```
meanm1 = mean(resid(modelo1))
cat("Media de los residuos - Modelo 1:", meanm1, "\n")
```

```
## Media de los residuos - Modelo 1: 3.739394e-14
```

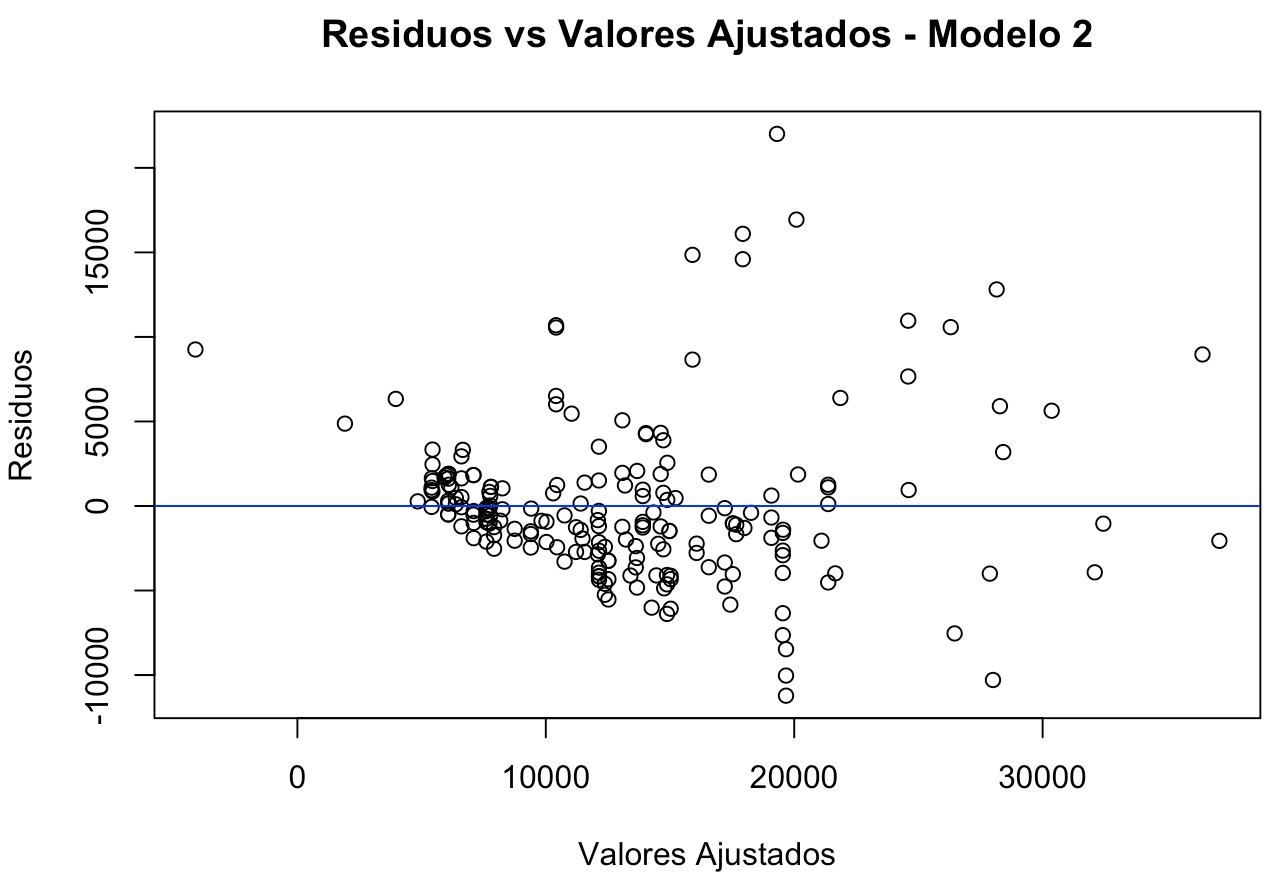
```
meanm2 = mean(resid(modelo2))
cat("Media de los residuos - Modelo 2:", meanm2, "\n")
```

```
## Media de los residuos - Modelo 2: 7.268668e-13
```

```
plot(fitted(modelo1), resid(modelo1), xlab = "Valores Ajustados", ylab = "Residuos", mai
n = "Residuos vs Valores Ajustados - Modelo 1")
abline(h = 0, col = "red")
```




```
plot(fitted(modelo2), resid(modelo2), xlab = "Valores Ajustados", ylab = "Residuos", mai
n = "Residuos vs Valores Ajustados - Modelo 2")
abline(h = 0, col = "blue")
```



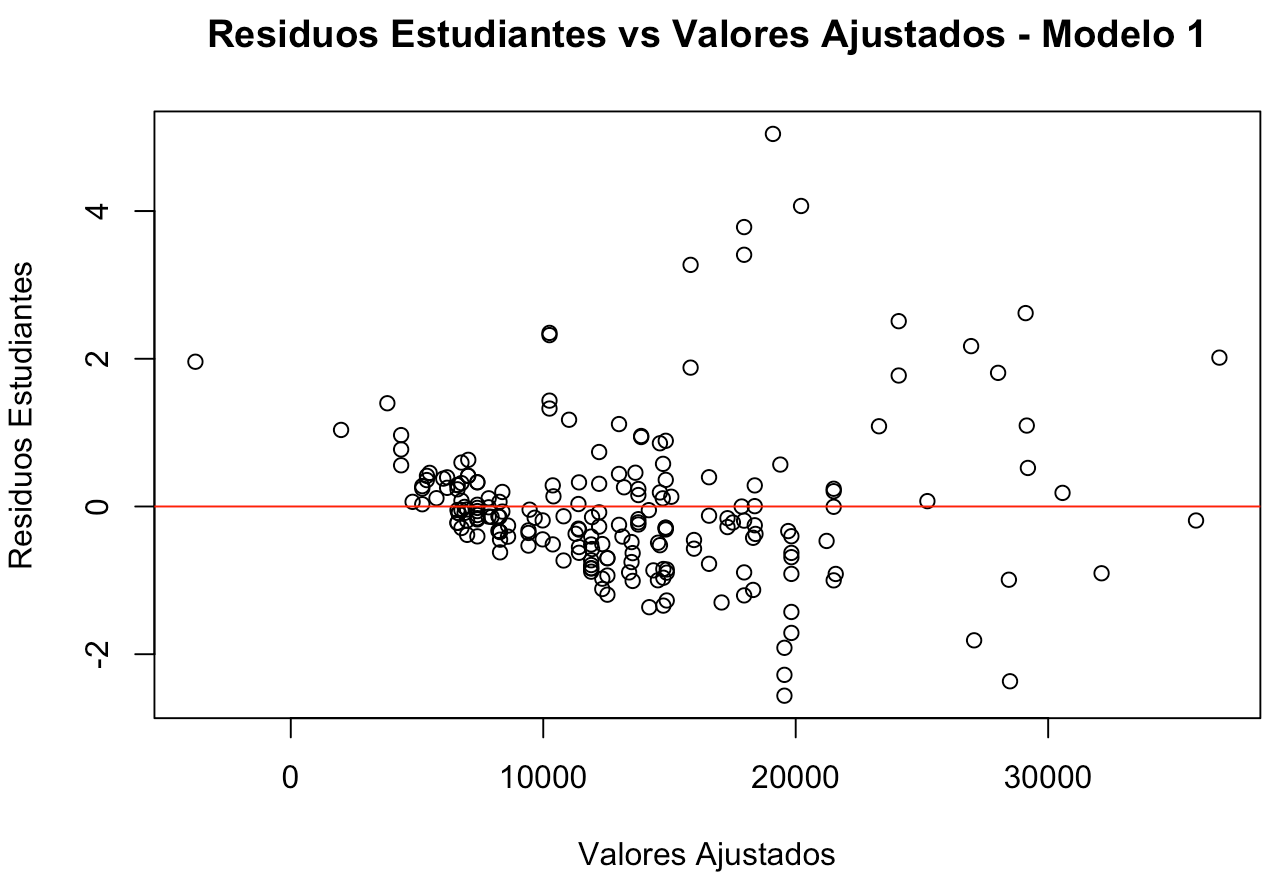

```
bptest_modelo1 = bptest(modelo1)
bptest_modelo1
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modelo1
## BP = 37.966, df = 6, p-value = 1.141e-06
```

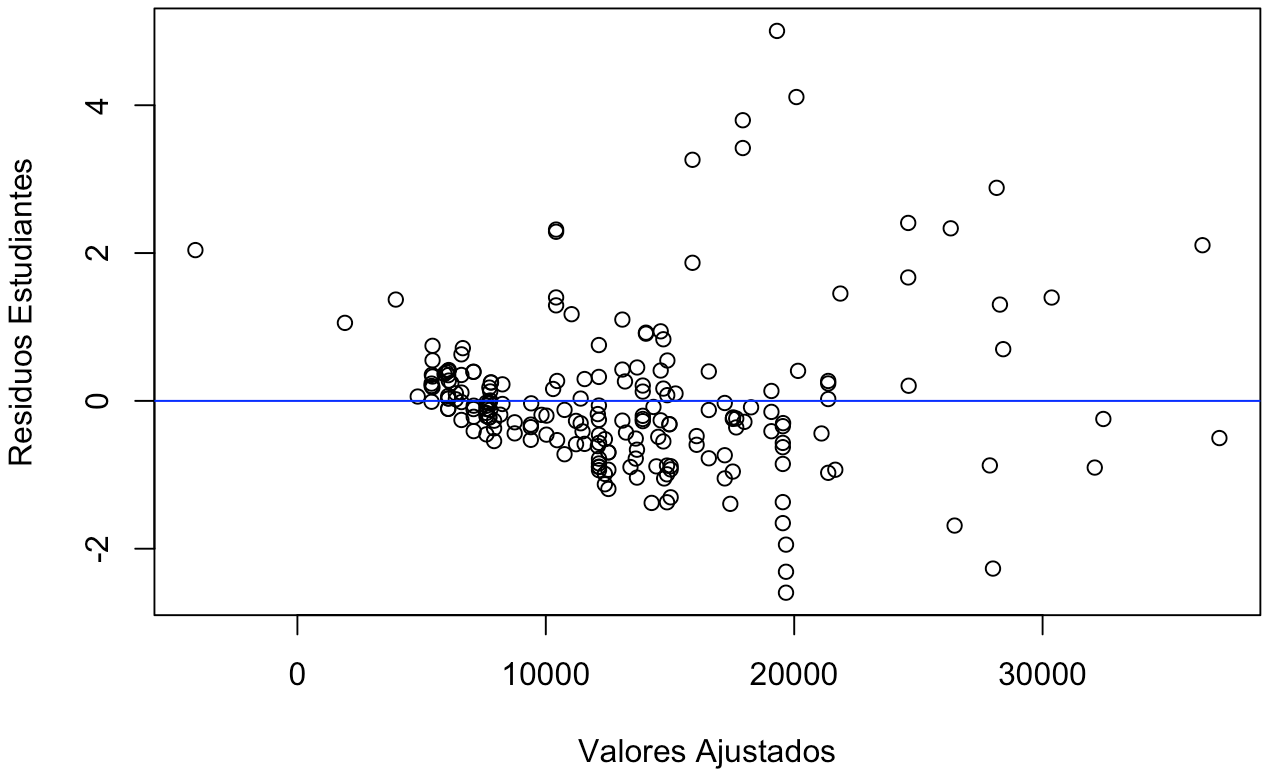
```
bptest_modelo2 = bptest(modelo2)
bptest_modelo2
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modelo2
## BP = 39.458, df = 7, p-value = 1.598e-06
```

```
plot(fitted(modelo1), rstudent(modelo1),
     xlab = "Valores Ajustados", ylab = "Residuos Estudiantes",
     main = "Residuos Estudiantes vs Valores Ajustados - Modelo 1")
abline(h = 0, col = "red")
```




```
plot(fitted(modelo2), rstudent(modelo2),
     xlab = "Valores Ajustados", ylab = "Residuos Estudiantes",
     main = "Residuos Estudiantes vs Valores Ajustados - Modelo 2")
abline(h = 0, col = "blue")
```




```
dwm1 = durbinWatsonTest(modelo1)
dwm1
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.6123376      0.769739       0
##  Alternative hypothesis: rho != 0
```

```
dwm2 = durbinWatsonTest(modelo2)
dwm2
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.5942951     0.8073016       0
##  Alternative hypothesis: rho != 0
```

```
modelo2 = lm(price ~ carheight * carwidth + carbody, data = data)

summary(modelo2)
```

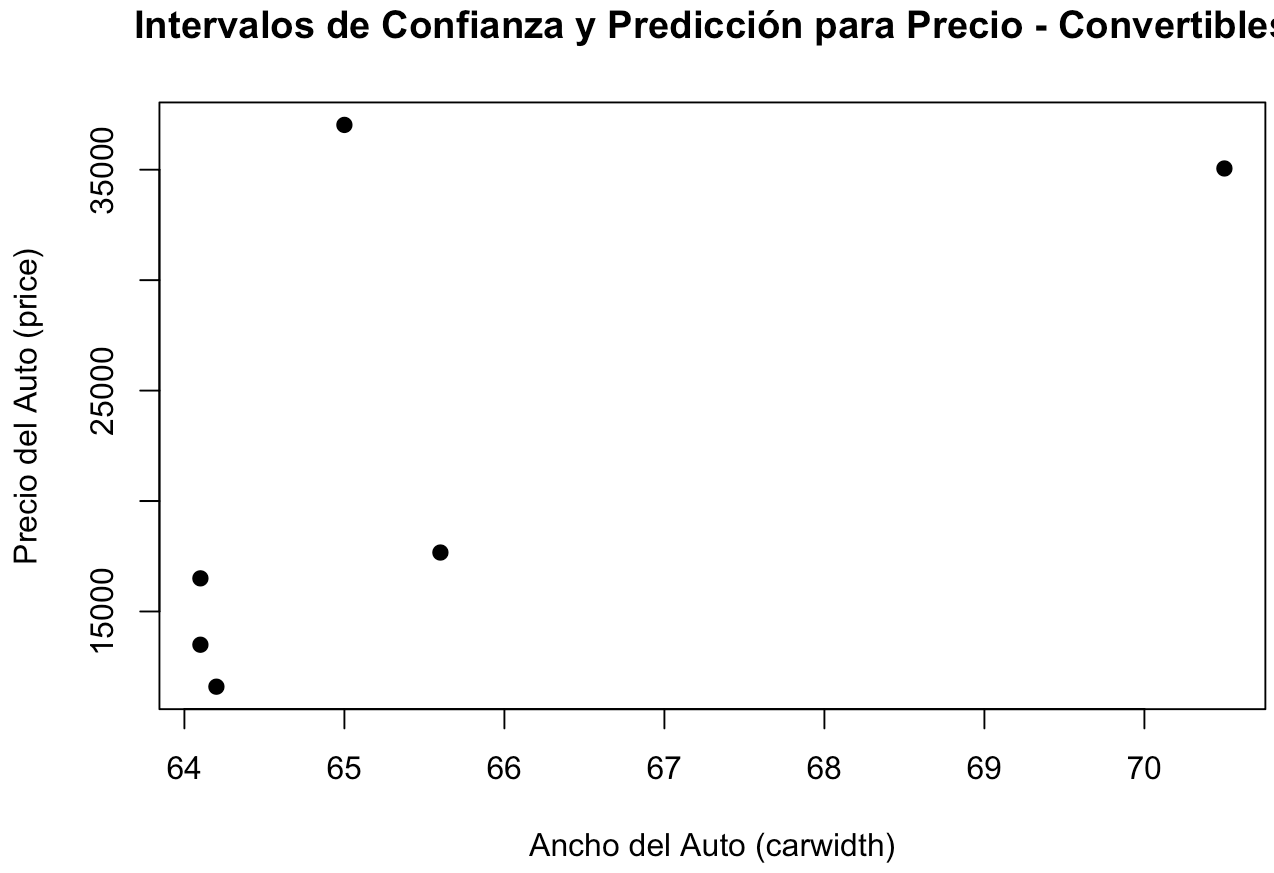
```
##
## Call:
## lm(formula = price ~ carheight * carwidth + carbody, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11220.7  -2458.1   -563.4   1382.6  22008.4
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -436072.67  221003.36  -1.973   0.0499 *
## carheight             5043.33    4080.68   1.236   0.2180
## carwidth              7117.82    3336.14   2.134   0.0341 *
## carbodyhardtop       -2157.81    2551.05  -0.846   0.3987
## carbodyhatchback    -10424.67    2002.28  -5.206 4.82e-07 ***
## carbodysedan         -8774.32    2038.77  -4.304 2.64e-05 ***
## carbodywagon        -10319.42    2326.15  -4.436 1.52e-05 ***
## carheight:carwidth     -79.53      61.54  -1.292   0.1978
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4695 on 197 degrees of freedom
## Multiple R-squared:  0.6664, Adjusted R-squared:  0.6546
## F-statistic: 56.23 on 7 and 197 DF,  p-value: < 2.2e-16
```

```
predicciones = predict(modelo2, interval = "confidence")
predicciones_pred = predict(modelo2, interval = "prediction")
```

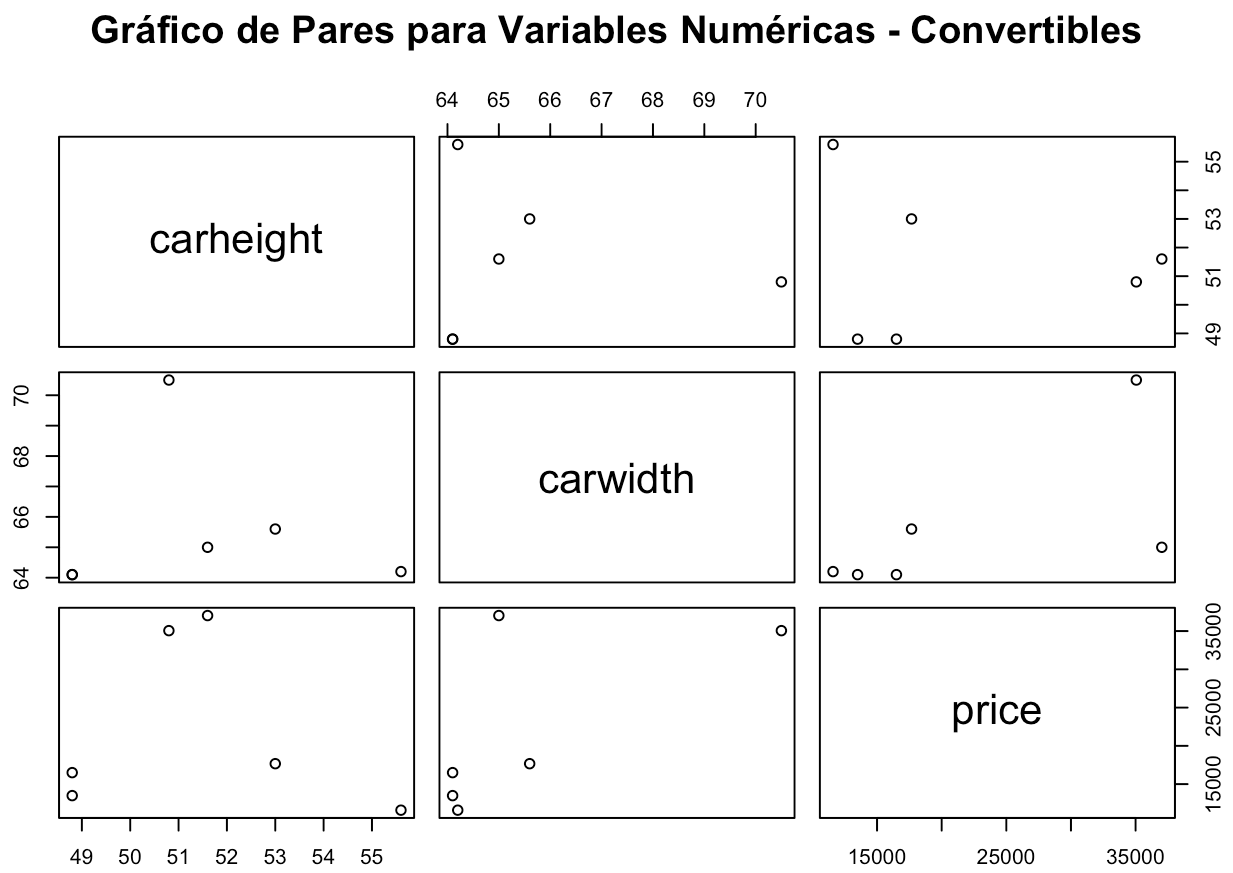
```
## Warning in predict.lm(modelo2, interval = "prediction"): predictions on current data
refer to _future_ responses
```

```
data$pred_conf = predicciones[, "fit"]
data$conf_lower = predicciones[, "lwr"]
data$conf_upper = predicciones[, "upr"]

data$pred_pred = predicciones_pred[, "fit"]
data$pred_lower = predicciones_pred[, "lwr"]
data$pred_upper = predicciones_pred[, "upr"]
```

```
convertible_data = subset(data, carbody == "convertible")

b0 = modelo2$coefficients[1]
b1 = modelo2$coefficients[2]

pred_conf_line = function(x){ b0 + b1*x }

x = seq(min(convertible_data$carwidth), max(convertible_data$carwidth), length.out = 10
0)

colores = c("blue", "red")

plot(convertible_data$carwidth, convertible_data$price, pch=19, col="black",
     main="Intervalos de Confianza y Predicción para Precio - Convertibles",
     xlab="Ancho del Auto (carwidth)", ylab="Precio del Auto (price)")

lines(x, pred_conf_line(x), col=colores[1], lwd=2, lty=2)

lines(x, pred_conf_line(x) + 1.96 * sd(convertible_data$price), col=colores[2], lwd=2)
lines(x, pred_conf_line(x) - 1.96 * sd(convertible_data$price), col=colores[2], lwd=2)

polygon(c(x, rev(x)), c(pred_conf_line(x) - 1.96 * sd(convertible_data$price),
                        rev(pred_conf_line(x) + 1.96 * sd(convertible_data$price))),
        col=adjustcolor("pink2", alpha.f = 0.2), border=NA)
```



```
pairs(~ carheight + carwidth + price, data = convertible_data,
      main = "Gráfico de Pares para Variables Numéricas - Convertibles")
```




```
dimensiones = select(data, carwidth, carheight, carlength, wheelbase)
med_dimensiones = colMeans(dimensiones, na.rm = TRUE)
corr_dimensiones = cor(dimensiones)

desempeno_motor = select(data, enginesize, horsepower, peakrpm)
med_desempeno = colMeans(desempeno_motor, na.rm = TRUE)
corr_desempeno = cor(desempeno_motor)

print(med_dimensiones)
```

```
##  carwidth carheight carlength wheelbase
##  65.90780  53.72488 174.04927  98.75659
```

```
print(corr_dimensiones)
```

```
##            carwidth carheight carlength wheelbase
## carwidth  1.0000000 0.2792103 0.8411183 0.7951436
## carheight 0.2792103 1.0000000 0.4910295 0.5894348
## carlength 0.8411183 0.4910295 1.0000000 0.8745875
## wheelbase 0.7951436 0.5894348 0.8745875 1.0000000
```

```
print(med_desempeno)
```

```
## enginesize horsepower    peakrpm
##   126.9073   104.1171  5125.1220
```

```
print(corr_desempeno)
```

```
##            enginesize horsepower    peakrpm
## enginesize  1.0000000  0.8097687 -0.2446598
## horsepower  0.8097687  1.0000000  0.1310725
## peakrpm    -0.2446598  0.1310725  1.0000000
```

Conclusiones

En las 3 variables que utilizamos vimos que en carwidth tiene una correlacion positiva con el precio, esto significa que los autos mas anchos son mas costosos por lo general. El carbody tambien influyen de manera significativa en el precio, lo mas probable porque estos carros tienden a tener un diseño mas exclusivo. Por ultimo el carheight no mostro una correlacion muy fuerte con el precio, pero si se noto como afectaba si lo juntabamos junto con el ancho.

El modelo 2 fue el mejor porque este incluye las interacciones entre las varibales, este modelo nos ayudaba de una mejor manera a predecir el precio de los automoviles, tambien este modelo tiene mas variabilidad que el modelo 1 y tambien vimos como con la combinacion de carwidht y carheight el precio era afectado.

Las agrupacion de variables que se nos dio tiene logica como vimos, pero creo que se pueden considerar agrupaciones alternas a esta. Hay variables que tienen mas sentido revisar juntas y se puede ver mejor como estas afectan el precio, en nuestro caso el carwidht y carheight combinadas si afectaban el precio pero se puede comparar mejor con variables que impactan el motor por ejemplo.

En conclusion agrupar variables por como afectan el carro de una manera similar es mejor porque daria un analisis mas centrado y especifico, esto ayudaria a identificar que afecta el precio de una mejor manera, lo que se nos dio ahorita si ayudo a ver como afectaba en cierta manera pero no fue lo mejor.