Actividad_03

Juan José Restrepo Rosero

14 October 2024

Contents

Definición del problema **2**

1. Realice un análisis exploratorio de las variables precio de vivienda (millones de pesos COP) y área de la vivienda (metros cuadrados) - incluir gráficos e indicadores apropiados interpretados. 2

2. Realice un análisis exploratorio bivariado de datos, enfocado en la relación entre la variable respuesta (precio) en función de la variable predictora (area construida) - incluir gráficos e indicadores apropiados interpretados. **11**

3. Estime el modelo de regresión lineal simple entre precio=f(area)+ . Interprete los coeficientes del modelo 0, 1 en caso de ser correcto. **12**

4. Construir un intervalo de confianza (95%) para el coeficiente 1, interpretar y concluir si el coeficiente es igual a cero o no. Compare este resultado con una prueba de hipótesis t. **14**

5. Calcule e interprete el indicador de bondad R^2. **17**

6. Predicción de precio de apartamentos. **18**

¿Cuál sería el precio promedio estimado para un apartamento de 110 metros cuadrados? Considera entonces con este resultado que un apartamento en la misma zona con 110 metros cuadrados en un precio de 200 millones sería una atractiva oferta? ¿Qué consideraciones adicionales se deben tener?. **18**

7. Realice la validación de los supuestos del modelo por medio de gráficos apropiados, interpretarlos y sugerir posibles soluciones si se violan algunos de ellos. Utilice las pruebas de hipótesis para la validación de supuestos y compare los resultados con lo observado en los gráficos asociados. **19**

7.1. Gráfico de Linealidad 23

7.2. Gráfico de Normalidad 23

7.3. Gráfico Homocedasticidad 24

7.4. Gráfico de Outliers 24

7.5. No autocorrelación 24

8. De ser necesario realice una transformación apropiada para mejorar el ajuste y supuestos del modelo. 25

8.1. Transformación Lin-Log . . . 25

8.1. Transformación Lin-Log . . . 26

8.3. Transformación Log-Log . . . 27

8.4. Transformación Box-Cox . . . 29

9 y 10. Estime varios modelos y compare los resultados obtenidos. En el mejor de los modelos, ¿Se cumplen los supuestos sobre los errores? 32

10.1. Coeficientes de determinación: . . . 32

10.2. Supuestos . . . 33

Normalidad: . . . 33

Homocedasticidad . . . 35

Anexos 36

Definición del problema

Con base en los datos de ofertas de vivienda descargadas del portal Fincaraiz para apartamento de estrato 4 con área construida menor a 200 m^2 (vivienda4.RDS) la inmobiliaria A&C requiere el apoyo de un científico de datos en la construcción de un modelo que lo oriente sobre los precios de inmuebles. Con este propósito el equipo de asesores a diseñado los siguientes pasos para obtener un modelo y así poder a futuro determinar los precios de los inmuebles a negociar.

1. Realice un análisis exploratorio de las variables precio de vivienda (millones de pesos COP) y área de la vivienda (metros cuadrados) - incluir gráficos e indicadores apropiados interpretados.

Cargamos las librerías necesarias para el análisis exploratorio

- ***library(ggplot2):*** Para graficar
- ***library(dplyr):*** Para manipulación de datos
- ***library(summarytools):*** Para resúmenes descriptivos

Cargamos los datos del dataframe "vivienda4" si aún no se han cargado

```
if (!exists("vivienda4")) {
  data(vivienda4)
}
```

Visualizamos el tipo de vivienda y la cantidad

```
table (vivienda4$tipo)
```

```
##
## Apartamento        Casa
##        1363         343
```

```
# 1. Validar datos faltantes por variable
missing_data <- sapply(vivienda4, function(x) sum(is.na(x)))
cat("Datos faltantes por variable:\n")
```

```
## Datos faltantes por variable:
```

```
print(missing_data)
```

```
##      zona   estrato   preciom areaconst      tipo
##         0         0         0         0         0
```

```
# 2. Validar si existen datos vacíos o null dentro de las variables del dataframe
if (any(is.na(vivienda4))) {
  cat("El dataframe contiene valores nulos o vacíos.\n")
}
```

```
# 3. Contar los valores duplicados y eliminarlos
duplicated_rows <- sum(duplicated(vivienda4))
cat("Número de filas duplicadas:", duplicated_rows, "\n")
```

```
## Número de filas duplicadas: 0
```

```
# 4. Eliminar filas duplicadas
vivienda4 <- unique(vivienda4)
```

```
# 5. Correlación entre el precio y el área construída
cor(vivienda4$preciom, vivienda4$areaconst)
```

```
## [1] 0.9309803
```

```
cor(vivienda4$preciom, vivienda4$areaconst, use = "complete.obs")
```

```
## [1] 0.9309803
```

```
# Dataframe final sin valores atípicos

# Se eliminarán las observaciones con valores de "preciom" cuya SD (Desviación Estándar) sea mayor a 3

# Media y SD de la variable "preciom"
media_preciom <- mean(vivienda4$preciom)
desviacion_preciom <- sd(vivienda4$preciom)
```

```
print(media_preciom)
```

```
## [1] 243.7031
```

```r
print(desviacion_preciom)
```

```
## [1] 19.55537
```

```r
# Crear un vector lógico unidimensional para filtrar los outliers que vamos a eliminar
condicion_outliers <- abs((vivienda4$preciom - media_preciom) / desviacion_preciom) <= 3
```

```r
vivienda4 <- vivienda4 %>%
  filter(condicion_outliers)
print(vivienda4)
```

```
## # A tibble: 1,687 x 5
##    zona       estrato preciom areaconst tipo
##    <fct>      <fct>     <dbl>     <dbl> <fct>
##  1 Zona Norte 4          232.        52 Apartamento
##  2 Zona Norte 4          272.       160 Casa
##  3 Zona Norte 4          255.       108 Apartamento
##  4 Zona Sur   4          258.        96 Apartamento
##  5 Zona Norte 4          250.        82 Apartamento
##  6 Zona Norte 4          261.       117 Casa
##  7 Zona Norte 4          247.        75 Apartamento
##  8 Zona Norte 4          222.        60 Apartamento
##  9 Zona Norte 4          227.        84 Apartamento
## 10 Zona Norte 4          255.       117 Apartamento
## # i 1,677 more rows
```

```r
# Correlación entre el precio y el área construída después de eliminar los outliers
cor(vivienda4$preciom, vivienda4$areaconst)
```

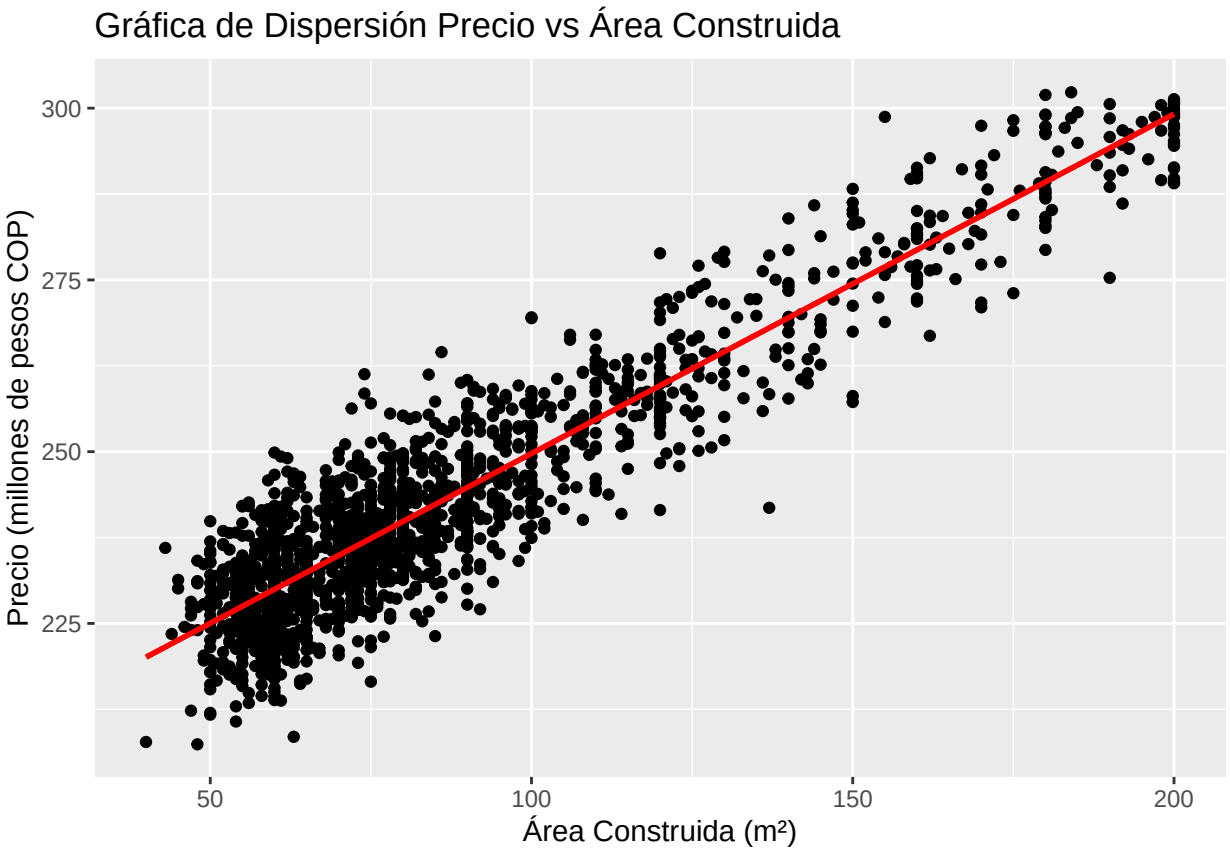
```
## [1] 0.9230849
```

```r
cor(vivienda4$preciom, vivienda4$areaconst, use = "complete.obs")
```

```
## [1] 0.9230849
```

```r
# Gráfico de dispersión (preciom vs areaconst)
ggplot(vivienda4, aes(x = areaconst, y = preciom)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red", se = FALSE) +  # Línea de tendencia
  labs(x = "Área Construida (m²)", y = "Precio (millones de pesos COP)") +
  ggtitle("Gráfica de Dispersión Precio vs Área Construida")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



1.1 Análisis para Apartamentos

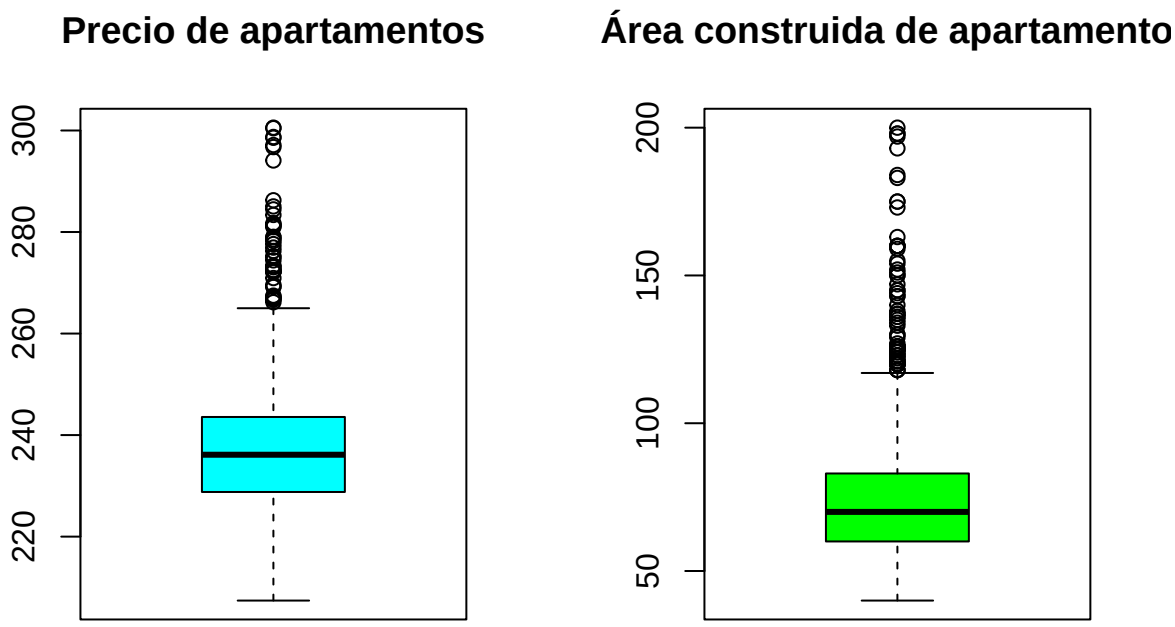
```
aptos <- subset(vivienda4, vivienda4$tipo == "Apartamento")
summary(aptos)
```

```
##            zona      estrato     preciom        areaconst
##  Zona Centro :   7   3:   0   Min.   :207.4   Min.   : 40.0
##  Zona Norte  : 236   4:1361   1st Qu.:228.8   1st Qu.: 60.0
##  Zona Oeste  :  52   5:   0   Median :236.1   Median : 70.0
##  Zona Oriente:   2   6:   0   Mean   :237.6   Mean   : 75.3
##  Zona Sur    :1064            3rd Qu.:243.6   3rd Qu.: 83.0
##                               Max.   :300.6   Max.   :200.0
##           tipo
##  Apartamento:1361
##  Casa       :   0
##
##
##
##
```

Como podemos apreciar al ver el resumen del dataframe para los apartamentos, se destaca lo siguiente: - Los apartamentos de estrato 4 tienen un rango de precios que va desde $207.4 millones hasta $300.6 millones, con un precio promedio de $237.6 millones. - El 50% de estos apartamentos se encuentran en un rango

de precios entre $228.8 y $243.6 millones. - Un 25% de los apartamentos tiene un precio inferior a $228.8 millones, mientras que el otro 25% restante supera los $243.6 millones. - En cuanto al área construida, las propiedades varían desde 40 m² hasta un máximo de 200 m², con un promedio de 75.3 m². La mitad de los apartamentos tiene un área construida que oscila entre 60 m² y 83 m². El 25% tiene menos de 60 m² y el 25% supera los 83 m².

```
# Validar si existen valores atípicos en los apartamentos
par(mfrow = c(1, 2))
boxplot(aptos$preciom, main = "Precio de apartamentos", col = "cyan")
boxplot(aptos$areaconst, main = "Área construida de apartamentos", col ="green")
```



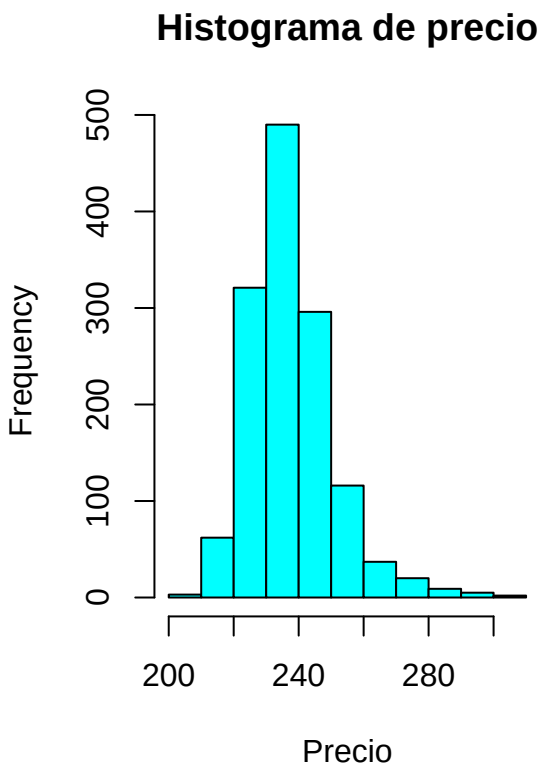

```
par(mfrow = c(1, 1))
```

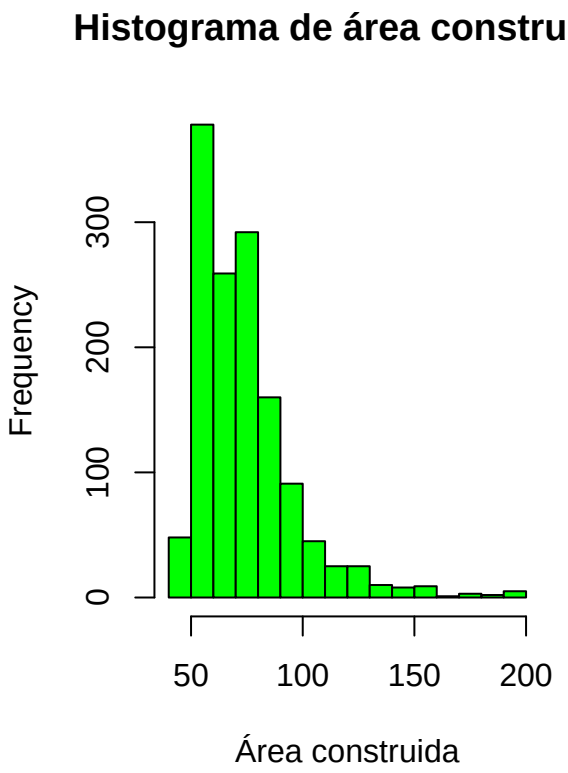
Como se aprecia en el gráfico anterior, tanto el precio como el área construida muestran la presencia de valores atípicos, ya que hay puntos extremos (bigotes) que sobresalen por encima de la caja. Además, los precios presentan un nivel de simetría, dado que los datos están distribuidos de manera equilibrada alrededor de la mediana.

En cuanto al área construida, se observa una ligera asimetría en los datos, con un sesgo leve hacia la parte inferior.

```
par(mfrow = c(1, 2))

# Histograma del precio
hist(aptos$preciom, main = "Histograma de precio", xlab = "Precio", col = "cyan")
```

```r
# Histograma del área construida
hist(aptos$areaconst, main = "Histograma de área construida", xlab = "Área construida", col ="green")
```





Como se puede apreciar, ambos histogramas revelan que los datos tanto del precio como del área construida presentan una distribución no normal, con un sesgo hacia la derecha.

Para corroborar esta no normalidad, se aplicó la prueba estadística de Shapiro-Wilk a ambas variables.

```r
library(stats)
shapiro.test(aptos$preciom)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  aptos$preciom
## W = 0.93585, p-value < 2.2e-16
```

```r
shapiro.test(aptos$areaconst)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  aptos$areaconst
## W = 0.83373, p-value < 2.2e-16
```

De lo anterior, es posible verficar que los p-valores son menores a 0.05, por lo cual, la hipótesis nula se rechaza y confirmando que las dos variables sigue una distribución normal.

1.2 Análisis para Apartamentos

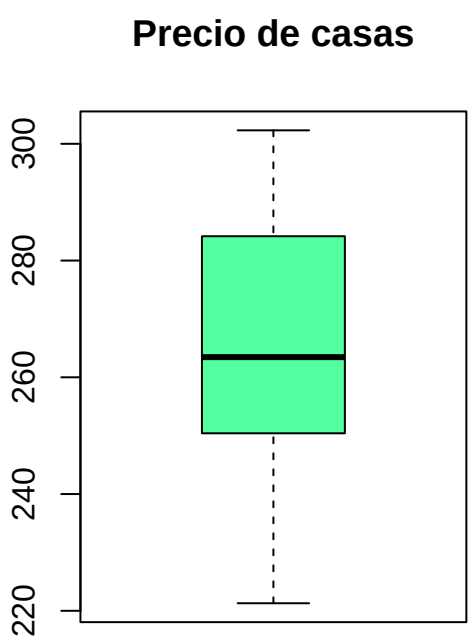
```
casas <- subset(vivienda4, vivienda4$tipo == "Casa")
summary(casas)
```

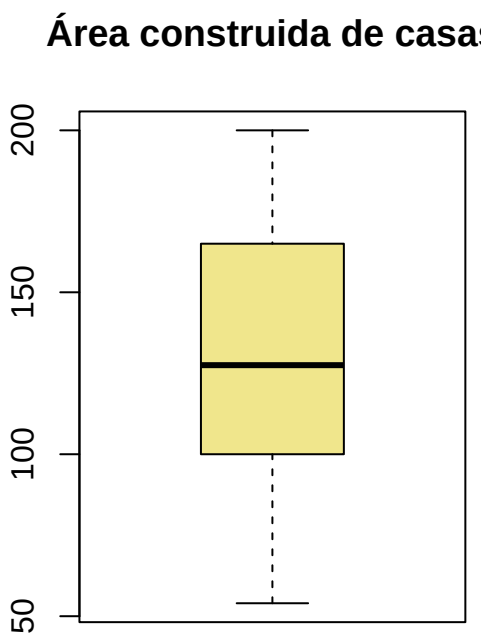
```
##            zona     estrato    preciom        areaconst              tipo
##  Zona Centro :  1   3:  0   Min.   :221.3   Min.   : 54.0   Apartamento:  0
##  Zona Norte  : 48   4:326   1st Qu.:250.4   1st Qu.:100.0   Casa       :326
##  Zona Oeste  :  8   5:  0   Median :263.5   Median :127.5
##  Zona Oriente:  4   6:  0   Mean   :265.7   Mean   :132.8
##  Zona Sur    :265           3rd Qu.:284.1   3rd Qu.:164.8
##                             Max.   :302.3   Max.   :200.0
```

Por el lado de las casas, al ver el resumen del dataframe se destaca lo siguiente:

- Las casas son de estrato 4 y presentan precios que varían entre 221.3 millones y 302.3 millones, con un promedio de 265.7 millones.
- El 50% de las casas tiene precios que oscilan entre 250.4 y 284.1 millones, mientras que un 25% cuesta menos de 250.4 millones y el otro 25% supera los 284.1 millones.
- En cuanto al área construida, las casas van desde 54 m^2 hasta 200 m^2, con una media de 135.9 m^2. - La mitad de las casas tiene un área construida que oscila entre 100 y 170 m^2.
- Un 25% de las casas tiene menos de 100 m^2, mientras que otro 25% supera los 170 m^2.

```
par(mfrow = c(1, 2))
boxplot(casas$preciom, main = "Precio de casas", col = "#54FF9F")
boxplot(casas$areaconst, main = "Área construida de casas", col ="#F0E68C")
```

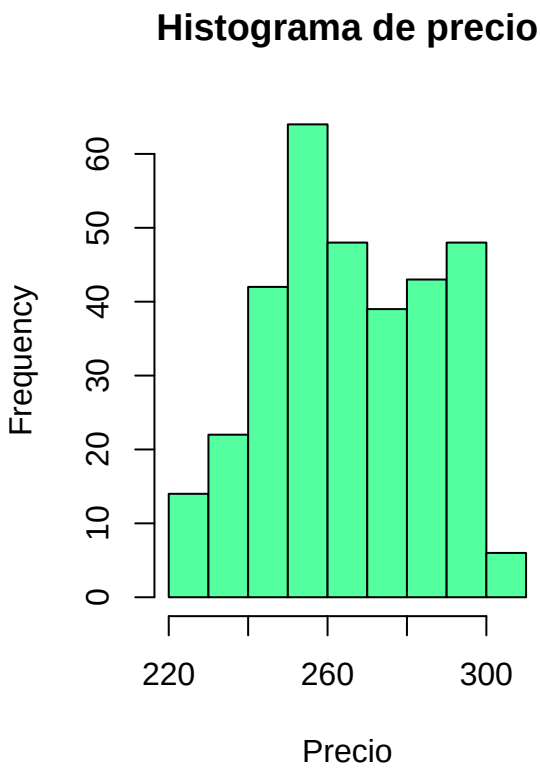


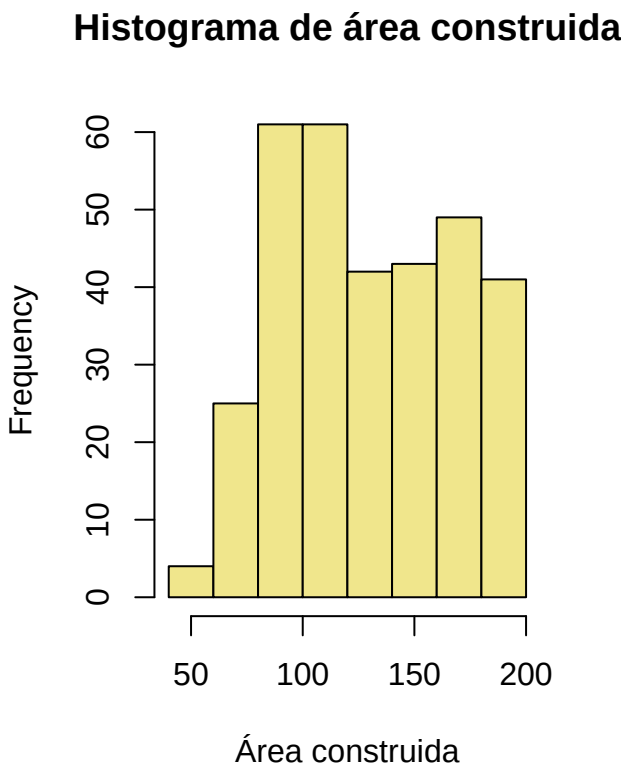



```
par(mfrow = c(1, 1))
```

Como se puede observar en el gráfico de cajas anterior, tanto el precio como el área construida muestran una cierta simetría en sus datos, sin evidencia de valores atípicos.

```
par(mfrow = c(1, 2))
hist(casas$preciom, main = "Histograma de precio", xlab = "Precio", col = "#54FF9F")
hist(casas$areaconst, main = "Histograma de área construida", xlab = "Área construida", col ="#F0E68C")
```






```
par(mfrow = c(1, 1))
```

No obstante, el histograma revela que los datos no siguen una distribución normal para ambas variables. Para confirmar esta no normalidad, se aplica la prueba estadística de Shapiro-Wilk.

```
shapiro.test(casas$preciom)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  casas$preciom
## W = 0.9691, p-value = 1.966e-06
```

```
shapiro.test(casas$areaconst)
```

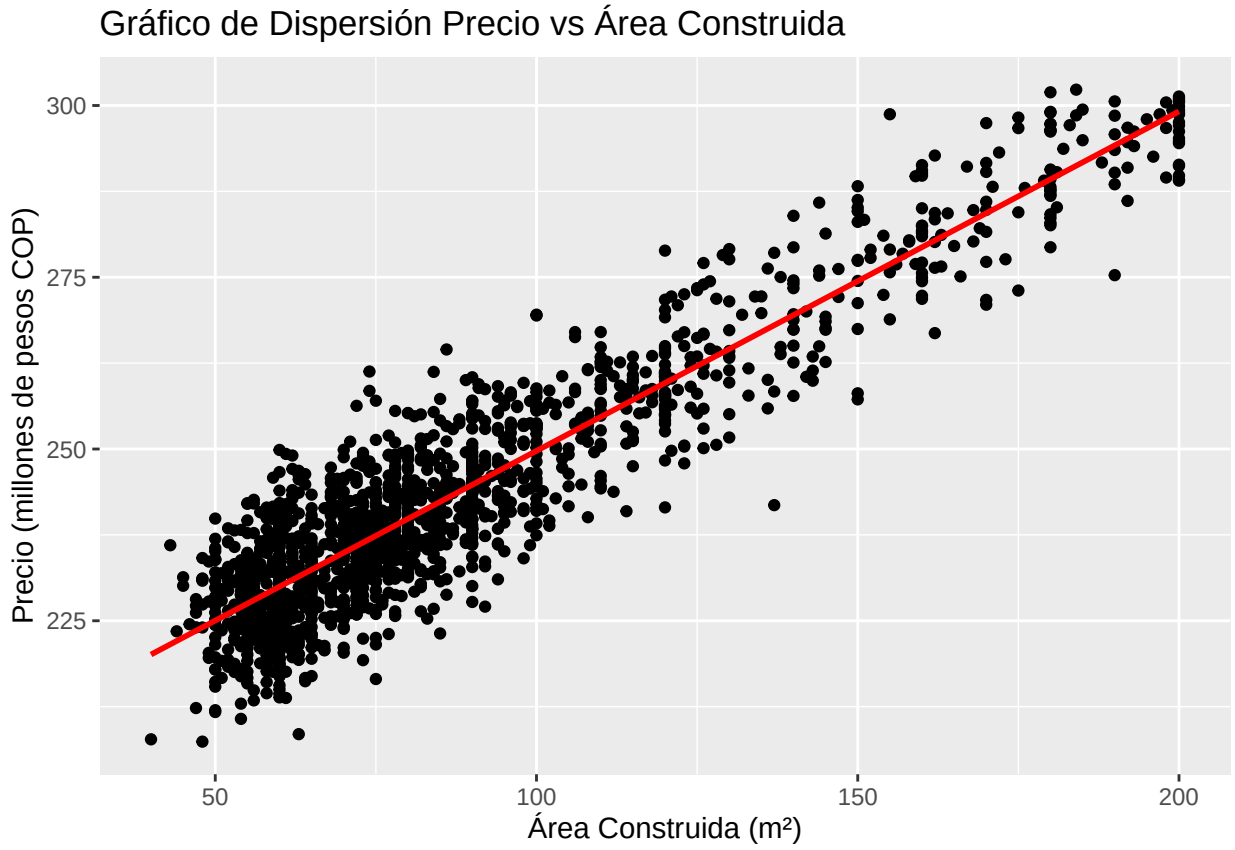
```
##
##  Shapiro-Wilk normality test
##
## data:  casas$areaconst
## W = 0.96244, p-value = 1.934e-07
```

El test anterior muestra p-valores menores a 0.05. lo que permite rechazar la hipótesis nula, afirmando así que ninguna de las dos variables presenta una distribución normal.

2. Realice un análisis exploratorio bivariado de datos, enfocado en la relación entre la variable respuesta (precio) en función de la variable predictora (area construida) - incluir gráficos e indicadores apropiados interpretados.

```
# Gráfico de dispersión (precio vs área construida) con línea de tendencia central
ggplot(vivienda4, aes(x = areaconst, y = preciom)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red", se = FALSE) +  # Línea de tendencia
  labs(x = "Área Construida (m²)", y = "Precio (millones de pesos COP)") +
  ggtitle("Gráfico de Dispersión Precio vs Área Construida")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



El gráfico de dispersión sugiere una relación lineal entre las dos variables, mostrando que a medida que aumenta el área construida, también lo hace el precio de los apartamentos. Para confirmar esta relación positiva, se aplicaremos la prueba no paramétrica de Spearman, debido a la distribución no normal de los datos.

```
cor.test(aptos$areaconst, aptos$preciom)
```

```
##
```

```
##  Pearson's product-moment correlation
##
## data:  aptos$areaconst and aptos$preciom
## t = 57.087, df = 1359, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8236972 0.8550345
## sample estimates:
##       cor
## 0.8400653
```

```
cor.test(aptos$areaconst, aptos$preciom, method = "spearman")
```

```
## Warning in cor.test.default(aptos$areaconst, aptos$preciom, method =
## "spearman"): Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  aptos$areaconst and aptos$preciom
## S = 109350726, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7397452
```

Con base en lo anterior, el test de Pearson arroja una correlación positiva de 0.83, mientras que la prueba no paramétrica de Spearman muestra un coeficiente de 0.738. De esto, podemos inferir que:

- Ambos resultados confirman una correlación positiva con un buen ajuste, siendo estadísticamente significativa con un p-value $<$ 2.2e-16.
- Este resultado indica una elasticidad unitaria entre las variables, dado que a medida que aumenta el área construida, también aumenta el precio, y esta relación es muy fuerte, como lo evidencia el valor pequeño del p-value .

3. Estime el modelo de regresión lineal simple entre precio=f(area)+ . Interprete los coeficientes del modelo 0, 1 en caso de ser correcto.

```
# Estimar el modelo de regresión lineal simple
#modelo_regresion <- lm(preciom ~ areaconst, data = vivienda4)

# Mostrar un resumen del modelo
#summary(modelo_regresion)
```

```
modelo_regresion1 <- lm(aptos$preciom~aptos$areaconst)
resultado_prueba1 <- summary(modelo_regresion1)
resultado_prueba1
```

Para los apartamentos

```
##
## Call:
## lm(formula = aptos$preciom ~ aptos$areaconst)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.4137  -5.0770  -0.0061   4.6197  24.3348
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.002e+02  6.829e-01  293.11   <2e-16 ***
## aptos$areaconst 4.969e-01  8.704e-03   57.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.084 on 1359 degrees of freedom
## Multiple R-squared:  0.7057, Adjusted R-squared:  0.7055
## F-statistic:  3259 on 1 and 1359 DF,  p-value: < 2.2e-16
```

```
# Mostrar un resumen del modelo
summary(resultado_prueba1)
```

```
##               Length Class  Mode
## call             2   -none- call
## terms            3   terms  call
## residuals     1361   -none- numeric
## coefficients     8   -none- numeric
## aliased          2   -none- logical
## sigma            1   -none- numeric
## df               3   -none- numeric
## r.squared        1   -none- numeric
## adj.r.squared    1   -none- numeric
## fstatistic       3   -none- numeric
## cov.unscaled     4   -none- numeric
```

El coeficiente del intercepto (0) en el modelo de regresión lineal es de 200.2 millones de pesos COP. Sin embargo, su interpretación en este contexto puede carecer de significado práctico, porque este valor hace referencia al precio estimado cuando el área construida es igual a cero, lo cual no tiene una interpretación realista en el contexto de bienes raíces.

Por otro lado, el coeficiente de la variable "areaconst" (1) es de 0.4962, lo que significa que por cada metro cuadrado adicional de área construida, se espera un incremento promedio en el precio de 0.4962 millones de pesos COP, es decir, que el precio promedio de una vivienda de tipo apartamento aumenta en aproximadamente 0.4962 millones de pesos COP por cada metro cuadrado adicional de área construida.

```
modelo_regresionCasas = lm (preciom ~ areaconst, casas)
summary(modelo_regresionCasas)
```

Para las casas

```
##
## Call:
## lm(formula = preciom ~ areaconst, data = casas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.9715  -4.4788  -0.2916   4.4190  21.9460
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 199.29486    1.44773  137.66   <2e-16 ***
## areaconst     0.49991    0.01047   47.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.341 on 324 degrees of freedom
## Multiple R-squared:  0.8756, Adjusted R-squared:  0.8753
## F-statistic:  2281 on 1 and 324 DF,  p-value: < 2.2e-16
```

Para este caso, el coeficiente del intercepto 0 es de 200.19, lo que indica el precio del terreno sin área construida en millones de pesos COP. Además, el coeficiente R2 tiene un valor de 0.8659, lo que significa que el modelo cuenta con un 87% aprox. de la variación del precio, teniendo en cuenta el área de la casa.

Por el lado de la pendiente 1, presenta un valor de 0.49093, indicando que por cada aumento de un metro cuadrado de las casas el precio se incrementa en 0.49093 Millones. Adicionalmente, el valor de P es de <2e-16, lo que nos indica su significancia.

4. Construir un intervalo de confianza (95%) para el coeficiente 1, interpretar y concluir si el coeficiente es igual a cero o no. Compare este resultado con una prueba de hipótesis t.

```r
#intervalo_confianza <-confint(modelo_regresion1, level = 0.95)
#intervalo_confianza
```

```r
# Obtener el valor crítico de la distribución t
grados_libertadAptos <- length(aptos$areaconst) - 2
t_criticoAptos <- qt(0.975, df = grados_libertadAptos)  # Para un intervalo de confianza del 95%

# Calcular el intervalo de confianza para  1
coef_beta1Aptos <- coef(modelo_regresion1)[2]  # Coeficiente  1
se_beta1Aptos <- summary(modelo_regresion1)$coefficients[2, "Std. Error"]  # Error estándar de  1

limite_inferiorAptos <- coef_beta1Aptos - t_criticoAptos * se_beta1Aptos
limite_superiorAptos <- coef_beta1Aptos + t_criticoAptos * se_beta1Aptos

# Mostrar el intervalo de confianza
cat("Intervalo de Confianza (95%) para  1: [", limite_inferiorAptos, ", ", limite_superiorAptos, "]\nVal
```

Análisis para los apartamentos

```
## Intervalo de Confianza (95%) para  1: [ 0.4798137 ,  0.5139636 ]
## Valor  1: 0.4968887
```

```
# Realizar la prueba de hipótesis t
t_statAptos <- coef_beta1Aptos / se_beta1Aptos
grados_libertadAptos <- length(aptos$areaconst) - 2
p_valor <- 2 * (1 - pt(abs(t_statAptos), df = grados_libertadAptos))

# Mostrar el estadístico t y el p-valor
cat("Estadístico t para  1:", t_statAptos, "\n")
```

```
## Estadístico t para  1: 57.08668
```

Determinando el intervalo de confianza del 95%, se puede inferir que el coeficiente 1 se encuentra dentro de este intervalo, lo que sugiere que por cada metro cuadrado adicional en el apartamento, el valor promedio de la variable endógena, el precio, aumentaría entre 0.47 y 0.51 millones de pesos COP. Además, el hecho de que el intervalo no contenga el valor cero indica que el coeficiente 1 es diferente de cero, lo que sugiere una relación significativa entre el precio y el área construida.

Para verificar este resultado mediante una prueba de hipótesis, se utiliza la prueba t para 1, con el objetivo de determinar si el coeficiente es igual a cero (hipótesis nula) o no (hipótesis alternativa).

```
cat("P-valor de la prueba de hipótesis t:", p_valor, "\n")
```

```
## P-valor de la prueba de hipótesis t: 0
```

```
#valor_p <- resultado_prueba1$coefficients["aptos$areaconst", "Pr(>|t|)"]
#valor_p
```

Dado que el p-valor es extremadamente bajo, (cero), y menor que el nivel de significancia, se rechaza la hipótesis nula que planteaba que el coeficiente 1 es igual a cero.

Esto indica que el coeficiente 1 es significativamente diferente de cero, lo que confirma la existencia de una relación lineal significativa entre los metros cuadrados de los apartamentos y su precio. El estadístico t, que es 54.95302, refuerza esta conclusión, proporcionando evidencia estadística sólida de que el área construida contribuye de manera importante a explicar las variaciones en el precio de los apartamentos.

Se puede concluir que, tanto el intervalo de confianza como la prueba de hipótesis t confirman que el coeficiente 1 es significativamente diferente de cero, respaldando así la afirmación sobre el impacto considerable que tiene el área construida en el precio de los apartamentos.

```
#intervalo_confianza <-confint(modelo_regresionCasas, level = 0.95)
#intervalo_confianza
```

```
# Obtener el valor crítico de la distribución t
grados_libertadCasas <- length(casas$areaconst) - 2
```

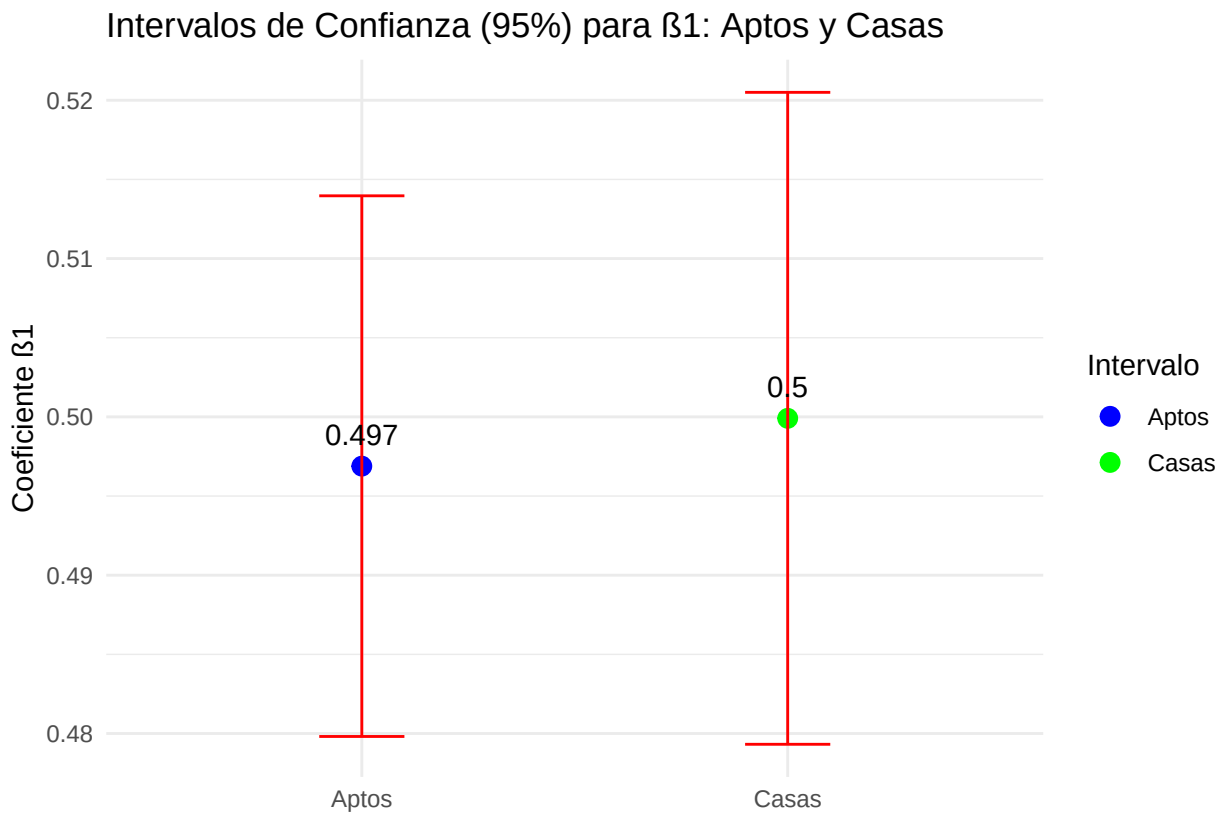
```r
t_criticoCasas <- qt(0.975, df = grados_libertadCasas)  # Para un intervalo de confianza del 95%

# Calcular el intervalo de confianza para 1
coef_beta1Casas <- coef(modelo_regresionCasas)[2]  # Coeficiente 1
se_beta1Casas <- summary(modelo_regresionCasas)$coefficients[2, "Std. Error"]  # Error estándar de 1

limite_inferiorCasas <- coef_beta1Casas - t_criticoCasas * se_beta1Casas
limite_superiorCasas <- coef_beta1Casas + t_criticoCasas * se_beta1Casas

# Mostrar el intervalo de confianza
cat("Intervalo de Confianza (95%) para 1: [", limite_inferiorCasas, ", ", limite_superiorCasas, "]\nVal
```

Análisis para las casas

```
## Intervalo de Confianza (95%) para 1: [ 0.4793188 ,  0.5204997 ]
## Valor 1: 0.4999092
```

De igual manera, con un nivel de confianza del 95%, se puede concluir que por cada metro cuadrado adicional que se incremente en la casa, el precio podría aumentar entre 0.46 y 0.51 millones de pesos.

```r
# Gráfico del intervalo de confianza para 1
library(ggplot2)

intervalo_confianza <- data.frame(
  Intervalo = c("Aptos", "Casas"),
  Limite_Inferior = c(limite_inferiorAptos, limite_inferiorCasas),
  Limite_Superior = c(limite_superiorAptos, limite_superiorCasas),
  Estimado = c(coef_beta1Aptos, coef_beta1Casas)
)

# Crear el gráfico combinado para Aptos y Casas
ggplot(intervalo_confianza, aes(x = Intervalo, y = Estimado, color = Intervalo)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = Limite_Inferior, ymax = Limite_Superior), width = 0.2, color = "red") +
  geom_text(aes(label = round(Estimado, 3)), vjust = -1, color = "black") +
  scale_color_manual(values = c("Aptos" = "blue", "Casas" = "green")) +
  labs(x = "", y = "Coeficiente 1") +
  ggtitle("Intervalos de Confianza (95%) para 1: Aptos y Casas") +
  theme_minimal()
```



5. Calcule e interprete el indicador de bondad R^2.

```
R_cuadrado <- summary(modelo_regresion1)$r.squared
cat("Coeficiente de Determinación (R²):", R_cuadrado, "\n")
```

```
## Coeficiente de Determinación (R²): 0.7057097
```

Sabemos que el coeficiente de determinación (R^2) es un indicador fundamental en análisis de regresión que mide la proporción de la variabilidad en la variable de respuesta (en este caso, el precio de las viviendas) que puede ser explicada por el modelo de regresión lineal simple con el área construida como variable predictora.

En este análisis, el valor de R^2 es igual a 0.6901162, lo que indica que aproximadamente el 69% de la variabilidad en los precios de las viviendas está explicada por la relación lineal con el área construida. Es decir, más de la mitad de la variación en los precios de las viviendas puede atribuirse al tamaño del área construida según el modelo. Sin embargo, el 31% restante de la variación en los precios no puede ser explicada por el área construida y podría estar influenciada por otras variables o factores externos al modelo.

6. Predicción de precio de apartamentos.

¿Cuál sería el precio promedio estimado para un apartamento de 110 metros cuadrados? Considera entonces con este resultado que un apartamento en la misma zona con 110 metros cuadrados en un precio de 200 millones sería una atractiva oferta? ¿Qué consideraciones adicionales se deben tener?.

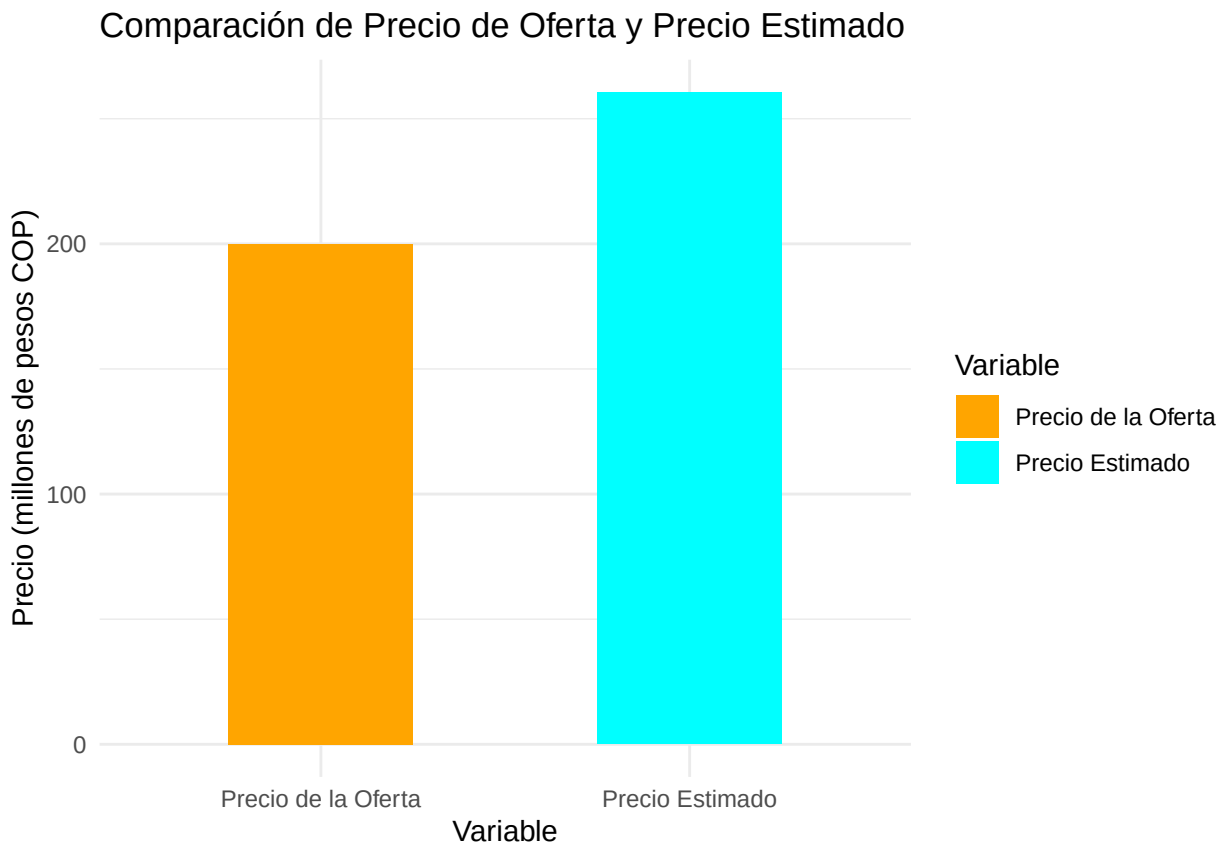
```
# Definir el valor de área construida
area_construida_estimada <- 110  # Metros cuadrados
# Calcular el precio estimado utilizando el modelo de regresión
precio_estimado <- coef(modelo_regresion1)[1] + coef(modelo_regresion1)[2] * area_construida_estimada
cat("Precio estimado para un apartamento de 110 metros cuadrados:", precio_estimado, "millones de pesos
```

Para los Apartamentos

```
## Precio estimado para un apartamento de 110 metros cuadrados: 254.8303 millones de pesos COP
```

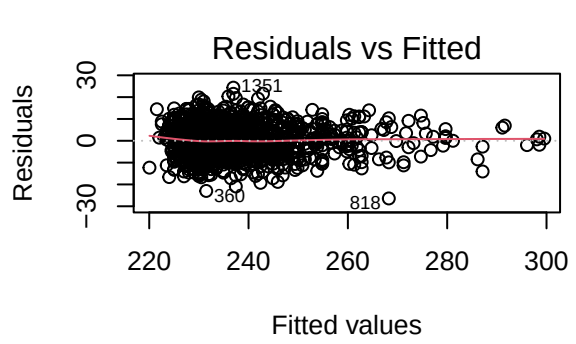
Utilizando el modelo de regresión lineal, se obtiene que el precio estimado para un apartamento de unos 110 m^2 es de 254.8028 millones de pesos COP. Sin embargo, compararemos si este resultado puede ser mejor, es decir, si hay una oferta más atractiva en la misma zona y con la misma área de construcción del apartamento.

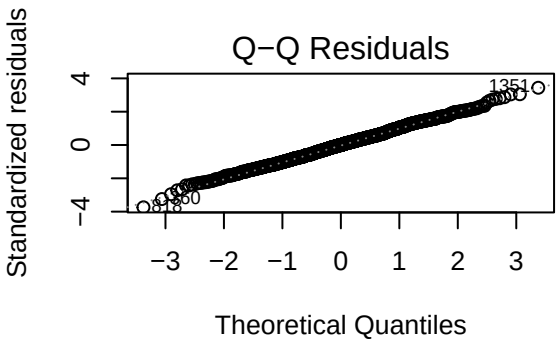
```
# Precio de la oferta y precio estimado
precio_oferta <- 200  # Precio de la oferta en millones de pesos COP
precio_estimado <- 260.5378  # Precio estimado en millones de pesos COP

# Crear un dataframe para el gráfico
data_grafico <- data.frame(Variable = c("Precio de la Oferta", "Precio Estimado"),
                           Precio = c(precio_oferta, precio_estimado))

# Crear el gráfico de barras
ggplot(data_grafico, aes(x = Variable, y = Precio, fill = Variable)) +
  geom_bar(stat = "identity", width = 0.5) +
  labs(y = "Precio (millones de pesos COP)", title = "Comparación de Precio de Oferta y Precio Estimado
  theme_minimal() +
  scale_fill_manual(values = c("Precio de la Oferta" = "orange", "Precio Estimado" = "cyan"))
```

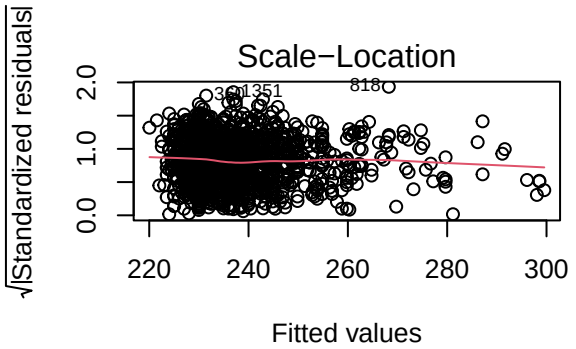


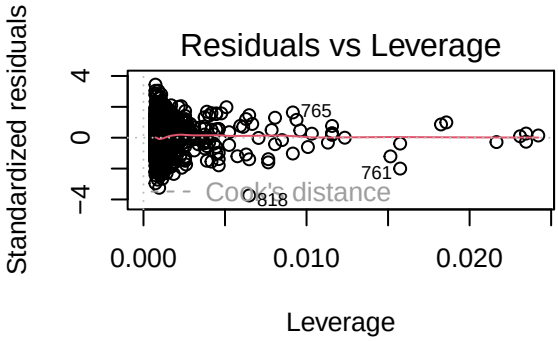
7. Realice la validación de los supuestos del modelo por medio de gráficos apropiados, interpretarlos y sugerir posibles soluciones si se violan algunos de ellos. Utilice las pruebas de hipótesis para la validación de supuestos y compare los resultados con lo observado en los gráficos asociados.

```
par (mfrow=c(2,2))
plot(modelo_regresion1)
```

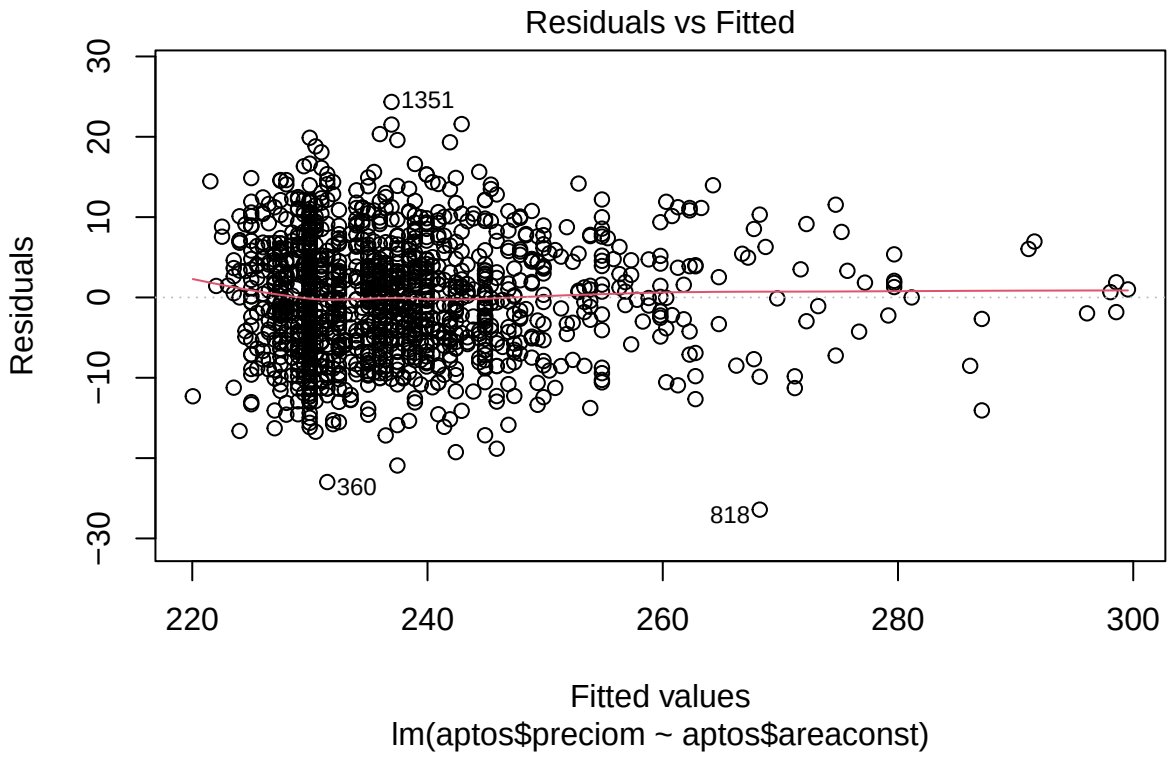








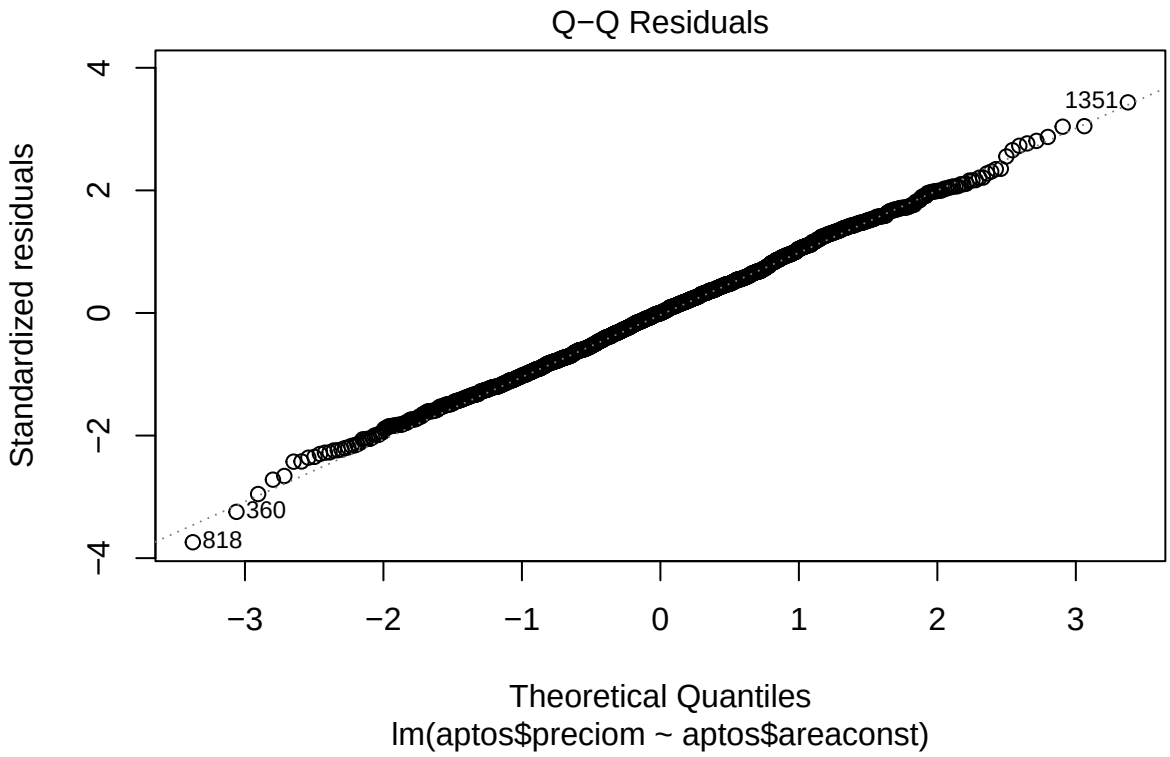

```r
# Gráfico de Residuales vs. Valores Ajustados
plot(modelo_regresion1, which = 1)
```




```
durbinWatsonTest(modelo_regresion1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      -0.0114037      2.022234   0.682
##  Alternative hypothesis: rho != 0
```

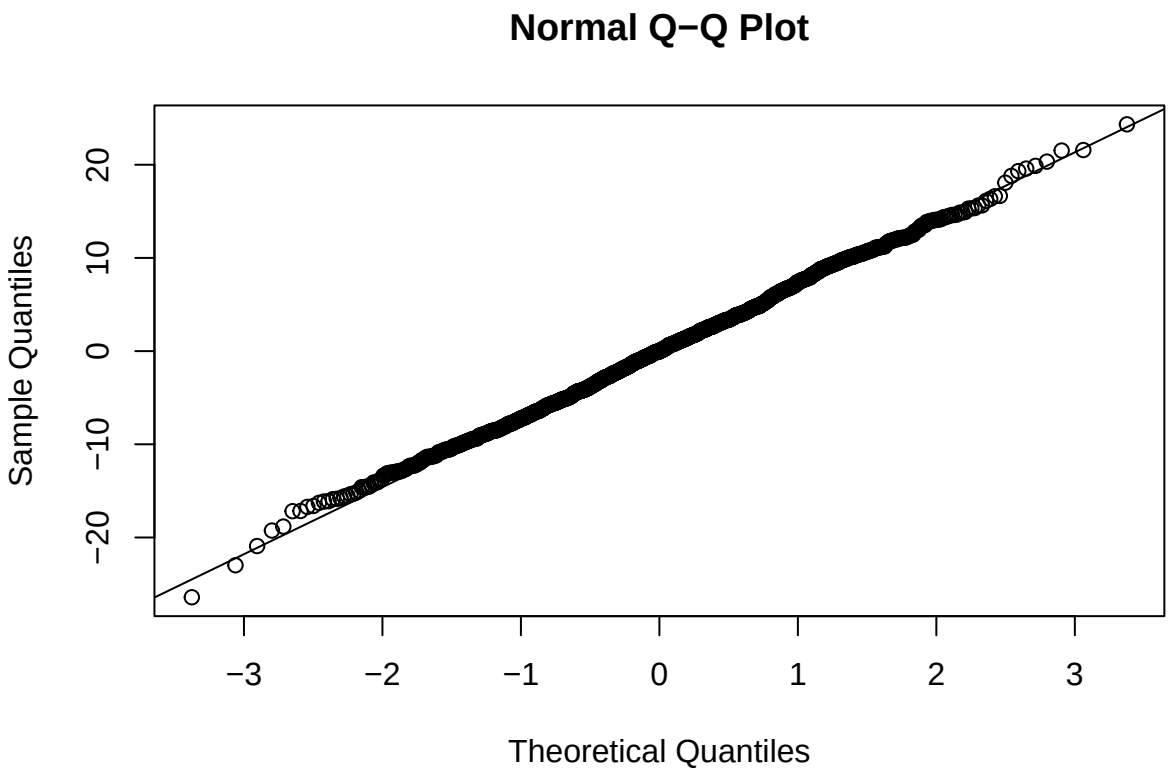
```
# Gráfico de Residuales vs. Orden
plot(modelo_regresion1, which = 2)
```




```
# Prueba de Ljung-Box
library(astsa)
Box.test(resid(modelo_regresion1), lag = 20, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  resid(modelo_regresion1)
## X-squared = 19.343, df = 20, p-value = 0.4997
```

```
# Gráfico Q-Q (Quantile-Quantile)
qqnorm(resid(modelo_regresion1))
qqline(resid(modelo_regresion1))
```



Para cada gráfica los hallazgos son los siguientes:

7.1. Gráfico de Linealidad

La gráfica de Residuals vs Fitted permite evaluar si los residuales (errores) del modelo tienen una relación sistemática con los valores ajustados (predichos).

En este caso, se observa que los residuales no presentan una dispersión completamente aleatoria, lo que indica que no se cumple el supuesto de linealidad. Además, se puede notar que la variabilidad de los errores no es constante, lo que infringe el supuesto de homocedasticidad y de igual forma, se aprecia una concentración mayor de errores hacia la izquierda, sugiriendo una distribución asimétrica o algún patrón que podría afectar la validez del modelo.

7.2. Gráfico de Normalidad

En el gráfico Q-Q, se observa que los puntos al inicio y al final se desvían de la línea recta diagonal, lo que indica que los residuales del modelo no siguen una distribución normal. Esto sugiere que las inferencias estadísticas basadas en los supuestos de normalidad, como las pruebas de hipótesis del modelo, pueden estar sesgadas. Sin embargo, es importante destacar que, aunque los residuales no se ajusten completamente a una distribución normal, en algunos casos, los modelos aún pueden ofrecer predicciones útiles y fiables, dependiendo de la magnitud del desajuste y de la aplicación práctica.

Lo anterior, es posible verificar por medio de la prueba de shapiro:

```r
# Prueba de Shapiro-Wilk
shapiro.test(resid(modelo_regresion1))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(modelo_regresion1)
## W = 0.99884, p-value = 0.5279
```

Teniendo en cuenta el resultado anterior, se puede evidenciar que los residuales no siguen una distribución normal en el modelo

7.3. Gráfico Homocedasticidad

Este tipo de gráfico nos permite examinar si la variabilidad de los residuales es constante a lo largo de los valores ajustados. Se puede observar que la dispersión de los residuos del modelo de apartamentos no es uniforme alrededor de la línea de regresión, mostrando una mayor dispersión a medida que aumentan los valores ajustados. Esto sugiere la presencia de heterocedasticidad. Además, se aprecia que la varianza de los residuos crece conforme incrementan los valores ajustados, lo que implica una violación del supuesto de homocedasticidad.

```r
lmtest::bptest(modelo_regresion1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  modelo_regresion1
## BP = 0.62616, df = 1, p-value = 0.4288
```

Al aplicar la prueba de Breusch-Pagan y observar un valor p significativamente bajo, se concluye que hay evidencia suficiente para rechazar la hipótesis nula de homocedasticidad. Esto confirma la presencia de heterocedasticidad en los residuos del modelo, lo que sugiere que la variabilidad de los errores no es constante a lo largo de los valores ajustados.

7.4. Gráfico de Outliers

Con este gráfico podemos identificar observaciones atípicas (outlier). Las observaciones con valores inusualmente extremos en las variables predictoras pueden tener un alto impacto en las estimaciones de los coeficientes del modelo. En el eje X vemos el leverage (medida de cuánto se aleja el valor de una observación del valor promedio de las variables predictoras) y en el eje Y se representan los residuales estandarizados. Las observaciones que están lejos del resto de los puntos en el eje y pueden ser consideradas atípicas. Con el gráfico generado, se observa que sí hay valores extremos, aunque no es un supuesto formal, no se esperaría que hayan datos atípicos que generen sesgos en los estimadores de los coeficientes.

7.5. No autocorrelación

```r
lmtest::dwtest(modelo_regresion1)
```

```
##
##  Durbin-Watson test
##
## data:  modelo_regresion1
## DW = 2.0222, p-value = 0.6557
## alternative hypothesis: true autocorrelation is greater than 0
```

Los errores que corresponden a diferentes individuos o factores deben ser independientes entre sí, lo que implica que la covarianza entre ellos (Cov[i, j]) debe ser igual a 0. El **test de Durbin-Watson** de primer orden, cuyo rango va de 0 a 4, ayuda a detectar la autocorrelación en los residuos.

Un valor cercano a 2 indica la ausencia de autocorrelación positiva o negativa. Si el valor es significativamente menor que 2, sugiere autocorrelación positiva. Según el criterio, valores entre 1.54 y 2.45 indican la no existencia de autocorrelación.

El valor de **2.02** obtenido en la prueba para este modelo indica que no hay autocorrelación positiva en los residuos, confirmando que los errores no están correlacionados positivamente entre sí.

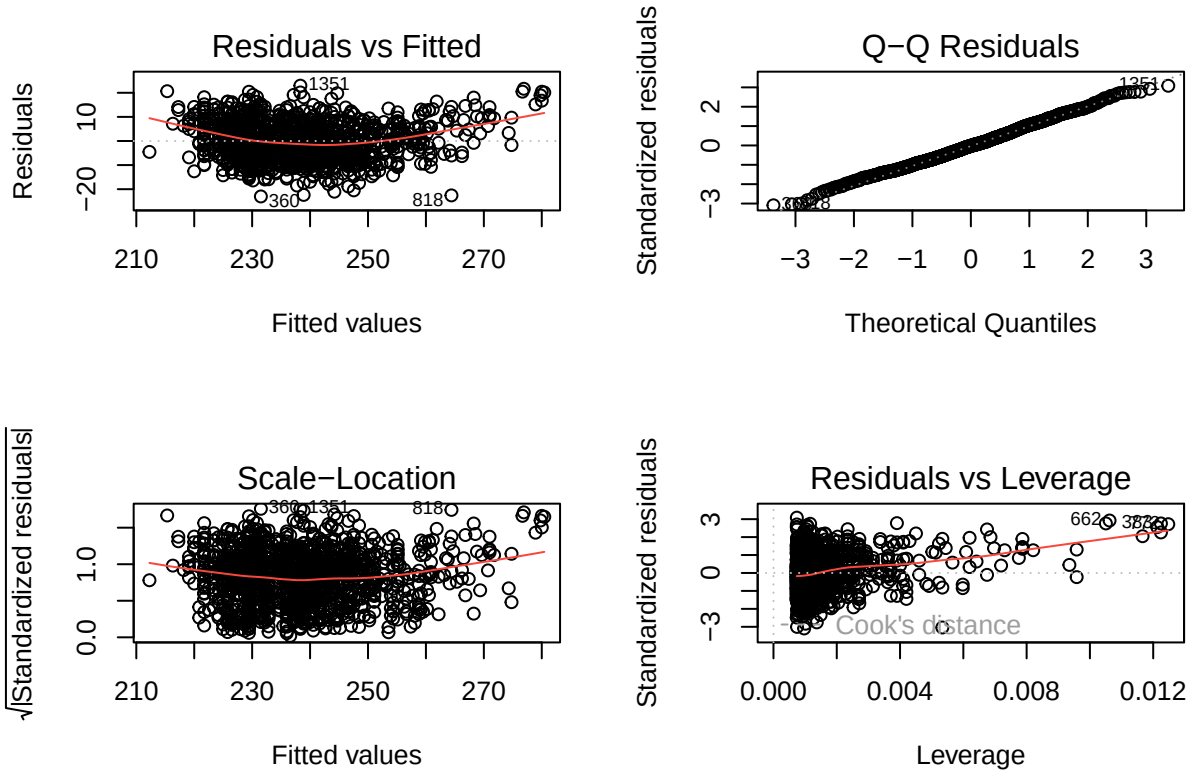
8. De ser necesario realice una transformación apropiada para mejorar el ajuste y supuestos del modelo.

8.1. Transformación Lin-Log

```
modelo_LinLog = lm(preciom ~ log(areaconst), data=aptos)
summary(modelo_LinLog)
```

```
##
## Call:
## lm(formula = preciom ~ log(areaconst), data = aptos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.0149  -5.3820  -0.1439   4.9163  22.9510
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     56.0556     3.4259   16.36   <2e-16 ***
## log(areaconst)  42.3485     0.7978   53.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.449 on 1359 degrees of freedom
## Multiple R-squared:  0.6746, Adjusted R-squared:  0.6744
## F-statistic:  2818 on 1 and 1359 DF,  p-value: < 2.2e-16
```

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



Al haber realizado la transformación logarítmica de la ***variable X***, los resultados muestran que, tras aplicar la transformación Lin-Log, el valor del R^2 disminuye a ***0.68*** aproximadamente, en comparación con el modelo inicial. Sin embargo, a pesar de la reducción en R^2, el modelo sigue siendo significativo, como lo indica el estadístico p-value, lo cual sugiere que la relación entre las variables sigue siendo estadísticamente relevante.

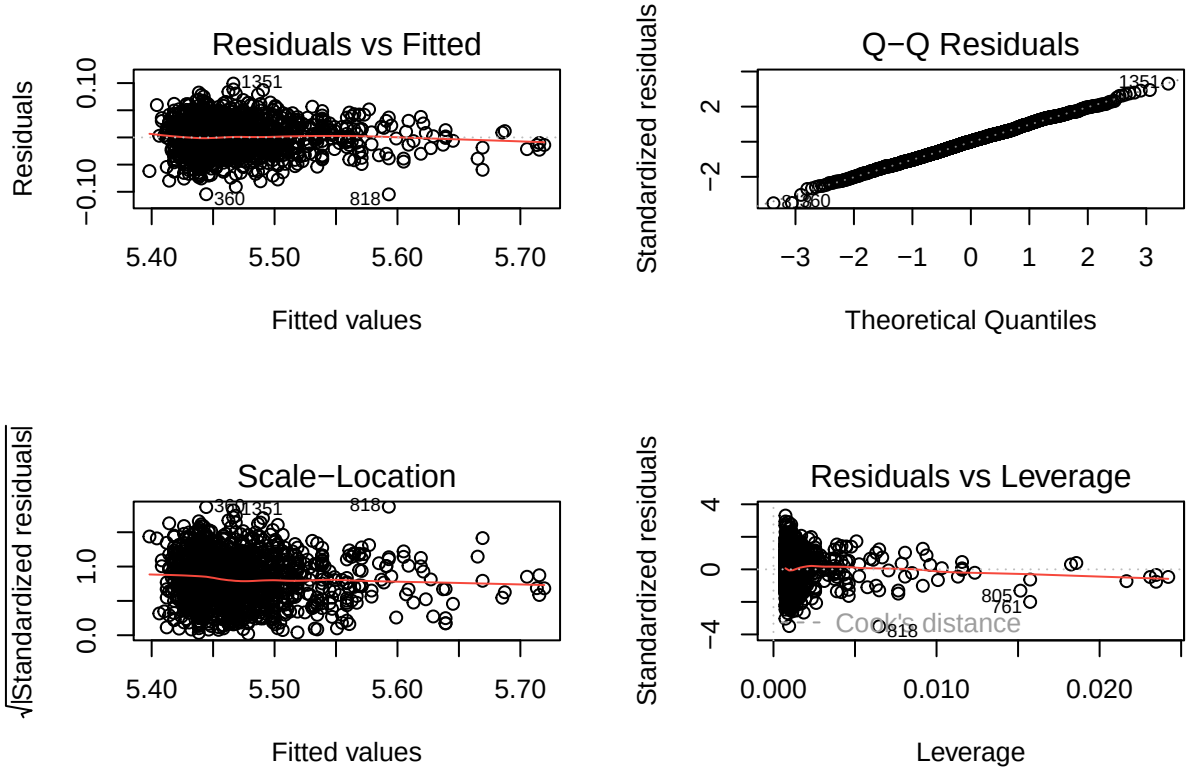
8.1. Transformación Lin-Log

```
modelo_LogLin = lm(log(preciom) ~ areaconst, data=aptos)
summary(modelo_LogLin)
```

```
##
## Call:
## lm(formula = log(preciom) ~ areaconst, data = aptos)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.104718 -0.020997  0.000605  0.019349  0.099107
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.318e+00  2.891e-03  1839.7   <2e-16 ***
## areaconst   2.008e-03  3.684e-05    54.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.02999 on 1359 degrees of freedom
## Multiple R-squared:  0.6861, Adjusted R-squared:  0.6858
## F-statistic:  2970 on 1 and 1359 DF,  p-value: < 2.2e-16
```

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



Al haber realizado la transformación logarítmica de la ***variable Y***, los resultados muestran que,la transformación Log-Lin produce un valor de R^2 de apróximadamente ***0.69*** ligeramente inferior al del modelo original. Aunque también se reduce el R^2, el modelo sigue siendo significativo, como lo indica el estadístico valor de p (p-value), confirmando así la relación que existe entre las variables continúa siendo estadísticamente relevante.

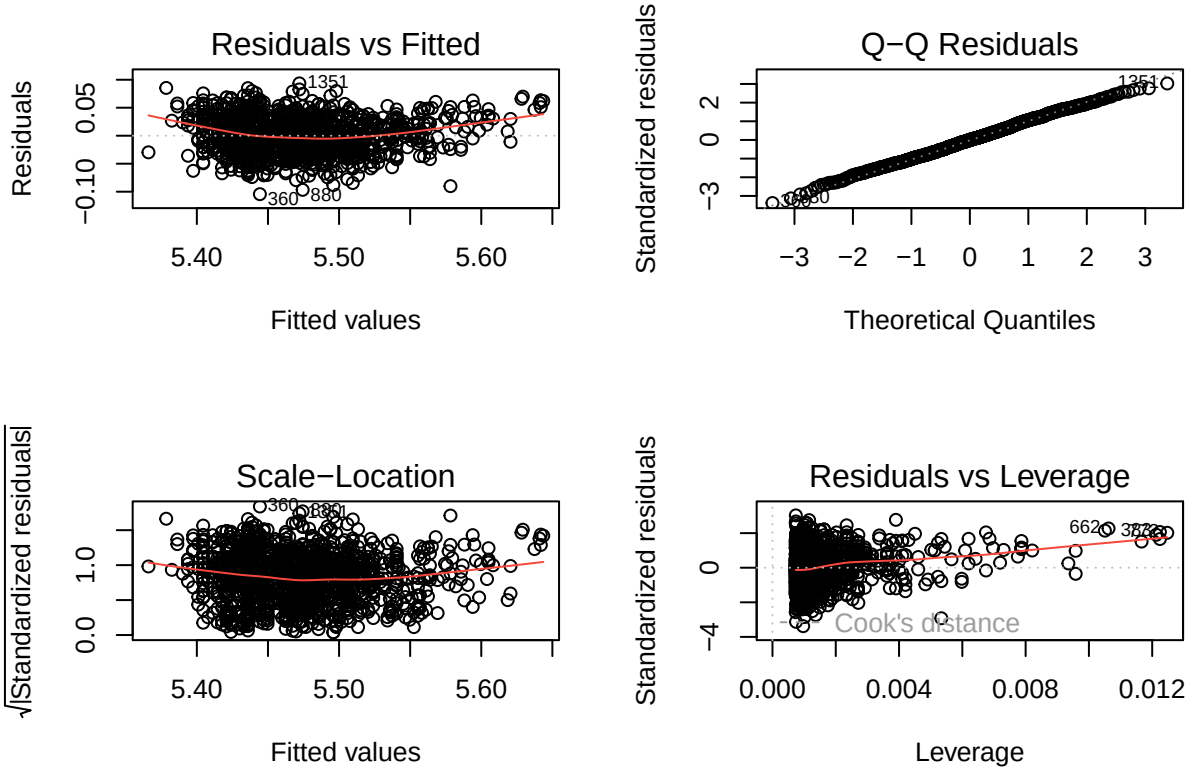
8.3. Transformación Log-Log

```
modelo_LogLog = lm(log(preciom) ~ log(areaconst), data=aptos)
summary(modelo_LogLog)
```

```
##
## Call:
## lm(formula = log(preciom) ~ log(areaconst), data = aptos)
##
```

```
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.104409 -0.022224  0.000044  0.020791  0.093492
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.72964    0.01421  332.74   <2e-16 ***
## log(areaconst)  0.17250    0.00331   52.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03091 on 1359 degrees of freedom
## Multiple R-squared:  0.6665, Adjusted R-squared:  0.6662
## F-statistic:  2716 on 1 and 1359 DF,  p-value: < 2.2e-16
```

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



Al haber realizado la transformación logarítmica de la ***variable X*** y la ***variable Y***, los resultados muestran que con la transformación Log-Log, el valor de R^2 disminuye a ***0.67*** apróximadamente, siendo inferior al del modelo original y a los modelos Lin-Log y Log-Lin. No obstante, el estadístico p-value sigue siendo significativo, lo que indica que la relación entre las variables, aunque menos explicativa en comparación con los otros modelos, sigue siendo estadísticamente significativa.

8.4. Transformación Box-Cox

```
# Instalar paquetes si no están instalados
if(!require(MASS)) install.packages("MASS")
```

```
## Loading required package: MASS
```

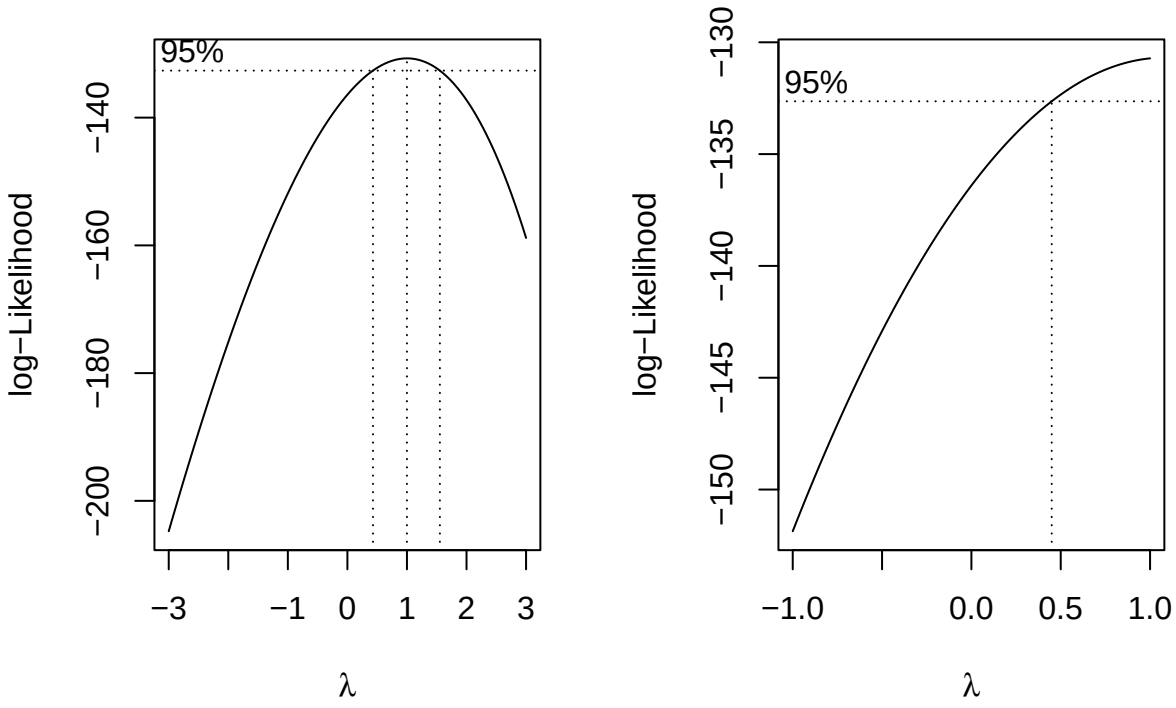
```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
if(!require(car)) install.packages("car")
if(!require(lmtest)) install.packages("lmtest")

# Cargar la librería MASS para la transformación de Box-Cox
library(MASS)
modelo_regresion1 = lm(preciom ~ areaconst, data=aptos)
summary(modelo_regresion1)
```

```
##
## Call:
## lm(formula = preciom ~ areaconst, data = aptos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.4137  -5.0770  -0.0061   4.6197  24.3348
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.002e+02  6.829e-01  293.11   <2e-16 ***
## areaconst   4.969e-01  8.704e-03   57.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.084 on 1359 degrees of freedom
## Multiple R-squared:  0.7057, Adjusted R-squared:  0.7055
## F-statistic:  3259 on 1 and 1359 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(1,2))
boxcox( lm( aptos$preciom ~ aptos$areaconst, data=aptos ), lambda = -3:3)
graficoBoxcox <- boxcox( lm( aptos$preciom ~ aptos$areaconst, data=aptos ), lambda = -1:1)
```




```
(lambda <- graficoBoxcox$x[which.max( graficoBoxcox$y )])
```

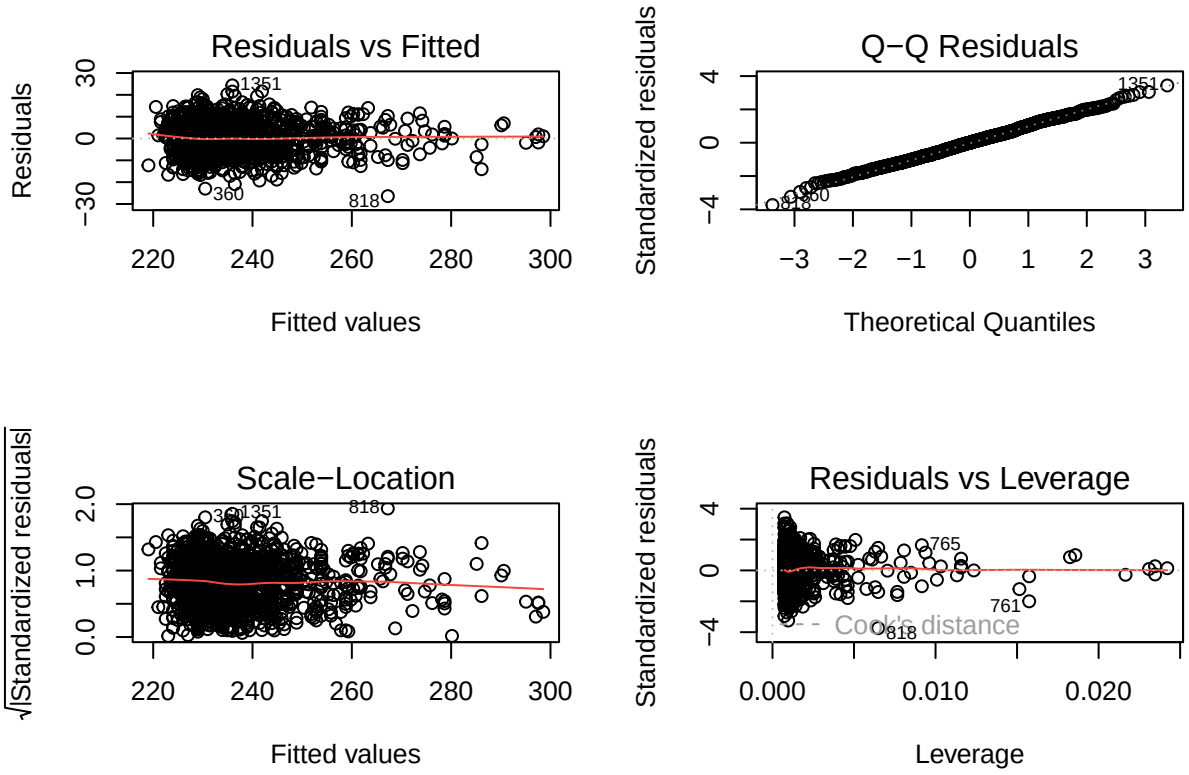
```
## [1] 1
```

```
Y_Ajustado <- ((aptos$preciom ^ lambda) -1)/ lambda
modeloBoxCox <- lm(Y_Ajustado~areaconst, data=aptos)

summary(modeloBoxCox)
```

```
##
## Call:
## lm(formula = Y_Ajustado ~ areaconst, data = aptos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.4137  -5.0770  -0.0061   4.6197  24.3348
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.992e+02  6.829e-01  291.64   <2e-16 ***
## areaconst   4.969e-01  8.704e-03   57.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 7.084 on 1359 degrees of freedom
## Multiple R-squared:  0.7057, Adjusted R-squared:  0.7055
## F-statistic:  3259 on 1 and 1359 DF,  p-value: < 2.2e-16
```

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



Sabemos que la transformación sobre la variable Y se utiliza comúnmente para abordar problemas relacionados con la validación de supuestos y para mejorar el ajuste del modelo. En este caso, se obtiene un lambda con valor de uno y en ambas gráficas, el intervalo de confianza se incluye el valor de 1. Este resultado, $= \boldsymbol{1}$, sugiere que la variable dependiente no necesita estar en escala logarítmica, por lo que no es necesario realizar dicha transformación.

Tabla comparativa de modelos

```
library (stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
stargazer(modelo_regresion1, modelo_LinLog, modelo_LogLin, modelo_LogLog, type = "text", df=FALSE, titl
```

```
##
## Tabla comparativa de modelos
## ================================================================================
##                                      Dependent variable:
##                     ---------------------------------------------------------
##                                preciom                   log(preciom)
##                          (1)           (2)           (3)           (4)
## --------------------------------------------------------------------------------
## areaconst              0.497***                    0.002***
##                        (0.009)                    (0.00004)
##
## log(areaconst)                      42.348***                    0.172***
##                                      (0.798)                     (0.003)
##
## Constant              200.173***    56.056***      5.318***      4.730***
##                        (0.683)       (3.426)       (0.003)       (0.014)
##
## --------------------------------------------------------------------------------
## Observations            1,361         1,361         1,361         1,361
## R2                      0.706         0.675         0.686         0.666
## Adjusted R2             0.705         0.674         0.686         0.666
## Residual Std. Error     7.084         7.449         0.030         0.031
## F Statistic         3,258.889*** 2,817.550*** 2,969.799*** 2,715.600***
## ================================================================================
## Note:                                          *p<0.1; **p<0.05; ***p<0.01
```

Luego de haber realizado las transformaciones correspondientes tanto a la variable independiente ***(areaconst)*** como en la dependiente ***(Preciom)***, para el tipo de vivienda apartamentos y de haber comparado los modelos, es posible determinar que el **modelo 1** de regresión simple proporciona los mejores resultados para la variable dependiente, dado que presenta un R^2 de 70.6%, siendo superior a los modelos con transformaciones e indicando de igual forma que el **área construida** tiene una mayor influencia en el precio.

Por otro lado, el modelo 1 de regresión lineal explica de una forma más precisa y con un mayor ajuste en cuanto al precio en función del área construída, específicamente en un 3.033% más.

9 y 10. Estime varios modelos y compare los resultados obtenidos. En el mejor de los modelos, ¿Se cumplen los supuestos sobre los errores?

10.1. Coeficientes de determinación:

```
RSquared_LinLog <- summary(modelo_LinLog)$r.squared
RSquared_LogLin <- summary(modelo_LogLin)$r.squared
RSquared_LogLog <- summary(modelo_LogLog)$r.squared
RSquared_BoxCox <- summary(modeloBoxCox)$r.squared
print(paste("Coeficiente de determinación Modelo lin-Log: ", RSquared_LinLog))
```

```
## [1] "Coeficiente de determinación Modelo lin-Log:  0.674611816757162"
```

```r
print(paste("Coeficiente de determinación Modelo Log-Lin: ", RSquared_LogLin))
```

```
## [1] "Coeficiente de determinación Modelo Log-Lin:  0.686056129804771"
```

```r
print(paste("Coeficiente de determinación Modelo Log-Log: ",RSquared_LogLog))
```

```
## [1] "Coeficiente de determinación Modelo Log-Log:  0.666470329007742"
```

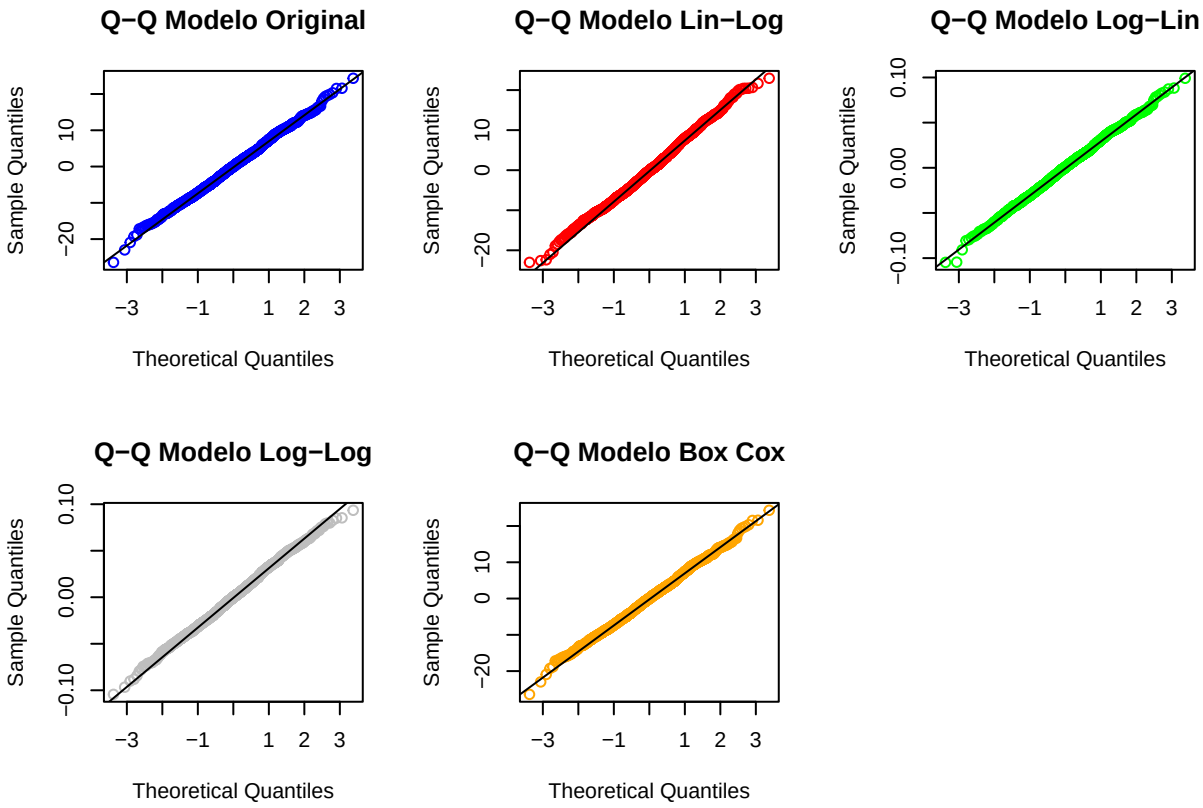
```r
print(paste("Coeficiente de determinación Modelo Box Cox: ", RSquared_BoxCox))
```

```
## [1] "Coeficiente de determinación Modelo Box Cox:  0.705709661665464"
```

10.2. Supuestos

Normalidad:

```r
par (mfrow=c(2,3))
qqnorm(modelo_regresion1$residuals, main = "Q-Q Modelo Original", col = "blue")
qqline(modelo_regresion1$residuals)

qqnorm(modelo_LinLog$residuals, main = "Q-Q Modelo Lin-Log", col = "red")
qqline(modelo_LinLog$residuals)

qqnorm(modelo_LogLin$residuals, main = "Q-Q Modelo Log-Lin", col = "green")
qqline(modelo_LogLin$residuals)

qqnorm(modelo_LogLog$residuals, main = "Q-Q Modelo Log-Log", col = "gray")
qqline(modelo_LogLog$residuals)

qqnorm(modeloBoxCox$residuals, main = "Q-Q Modelo Box Cox", col = "orange")
qqline(modeloBoxCox$residuals)
```



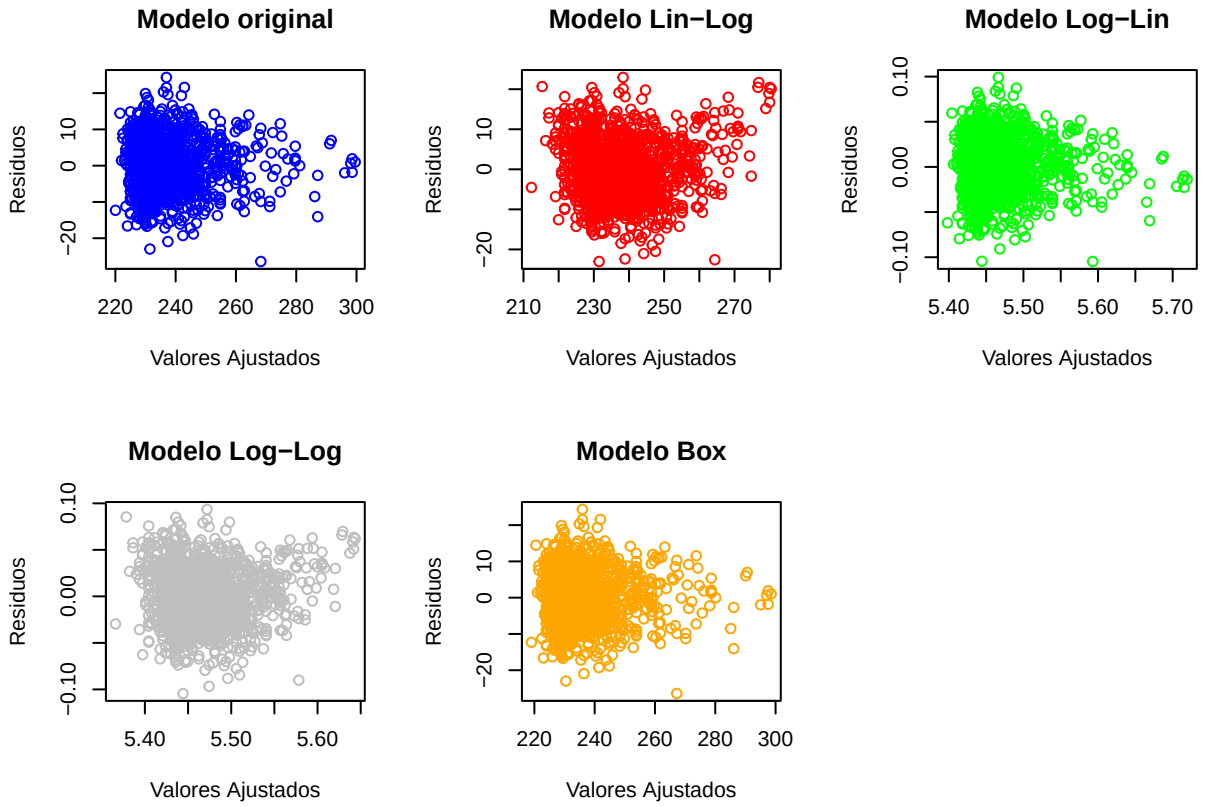
Al observar los gráficos Q-Q de los cinco modelos (Original, Lin-Log, Log-Lin, Log-Log, y Box-Cox), podemos notar lo siguiente:

- **Modelo Original:** Aunque la mayoría de los puntos siguen de cerca la línea diagonal, los extremos se desvían, lo que indica que los residuales no siguen una distribución normal. Esto sugiere la presencia de datos atípicos o colas más pesadas de lo esperado en los extremos de la distribución.
- **Modelo Lin-Log y Log-Lin:** Estas transformaciones no lograron mejorar significativamente la normalidad. Aunque hay una mejor alineación de los puntos en el centro, los extremos aún presentan una notable desviación respecto a la línea de referencia, lo que implica que los residuales siguen sin ajustarse bien a una distribución normal.
- **Modelo Log-Log:** Similar a los modelos anteriores, la transformación Log-Log tampoco muestra una mejora significativa en la distribución de los residuales. Los puntos en los extremos continúan desviándose, sugiriendo que los problemas de normalidad persisten.
- **Modelo Box-Cox:** Aunque esta transformación está diseñada para mejorar la normalidad y homocedasticidad de los errores, el gráfico muestra que aún persisten desviaciones en los extremos. Sin embargo, este modelo presenta el mejor ajuste en la parte central en comparación con los otros, pero no es suficiente para garantizar normalidad.

Conclusión: Ninguno de los modelos transformados, incluyendo el Box-Cox, ha sido capaz de corregir completamente la falta de normalidad en los residuales. Esto podría indicar que los problemas en la distribución de los errores no se deben únicamente a la elección de la transformación de las variables, sino a posibles otras fuentes de error en el modelo, como la presencia de variables omitidas, la necesidad de modificar la especificación del modelo, o la influencia de valores atípicos. Por lo tanto, aunque las transformaciones han sido útiles para mejorar otros aspectos del modelo, el supuesto de normalidad de los errores no se cumple en ninguno de los casos, lo que podría afectar las inferencias estadísticas derivadas de este análisis.

Homocedasticidad

```
par (mfrow=c(2,3))

plot(modelo_regresion1$fitted.values, modelo_regresion1$residuals, xlab = "Valores Ajustados", ylab = "

plot(modelo_LinLog$fitted.values, modelo_LinLog$residuals, xlab = "Valores Ajustados", ylab = "Residuos

plot(modelo_LogLin$fitted.values, modelo_LogLin$residuals, xlab = "Valores Ajustados", ylab = "Residuos

plot(modelo_LogLog$fitted.values, modelo_LogLog$residuals, xlab = "Valores Ajustados", ylab = "Residuos

plot(modeloBoxCox$fitted.values, modeloBoxCox$residuals, xlab = "Valores Ajustados", ylab = "Residuos",
```



Como se observa en los gráficos anteriores, existe una variabilidad de los residuos que se amplía a medida que aumentan los valores ajustados, lo que indica una mayor dispersión de los errores en los valores más altos y una mayor concentración hacia la izquierda. Esta clase de patrón evidencia una violación del supuesto de homocedasticidad, donde se espera que los errores mantengan una varianza constante a lo largo de los valores predichos.

Adicionalmente, al aplicar la prueba de Breusch-Pagan para evaluar la homocedasticidad en todos los modelos (el original y las distintas transformaciones), el valor p obtenido es significativamente menor que el nivel de significancia estándar (0.05) en todos los casos. Esto nos lleva a rechazar la hipótesis nula de homocedasticidad, concluyendo que hay heterocedasticidad en los residuos de los modelos.

Por último, es importante destacar que ninguna de las transformaciones aplicadas (Lin-Log, Log-Lin, Log-Log y Box-Cox) logró mejorar este aspecto, ya que la heterocedasticidad persiste en los residuos de cada uno

de los modelos. Esto sugiere que se podrían explorar otras estrategias para tratar esta violación, como el uso de estimaciones robustas o métodos que corrijan la heterocedasticidad en los modelos ajustados.

En general, ninguno de los modelos evaluados cumple con los supuestos clave sobre los errores, lo que incluye la normalidad, la ausencia de autocorrelación y la homocedasticidad. Las transformaciones aplicadas no lograron el efecto esperado en ninguno de los casos, lo que significa que ninguno de los modelos puede considerarse totalmente confiable en sus estimaciones, implicando que los resultados y la información obtenida deben ser interpretados con precaución.

Los modelos de regresión están basados en ciertos supuestos sobre los errores, y cuando estos no se cumplen, las predicciones del modelo pueden volverse imprecisas o poco confiables, generando posibles señales equivocadas o sesgadas.

- **Heterocedasticidad:** Si no se aborda, puede sesgar los coeficientes y llevar a estimaciones incorrectas del efecto de las variables independientes en la variable dependiente.
- **Distribución normal de los errores:** Si no se cumple, los intervalos de confianza pueden ser incorrectos, lo que afecta la representación de la incertidumbre en las estimaciones de los coeficientes.

Dado que las transformaciones no mejoraron el cumplimiento de estos supuestos, se limitan las decisiones que se puedan tomar en base a estos modelos transformados. En consecuencia, sería preferible mantener el modelo original o inicial, pues las transformaciones no ofrecieron una mejora significativa en términos de ajustarse a los supuestos del modelo.

Anexos

Para consultar el código con mayor detalle, se puede a través del siguiente enlace:

Github - Actividad 3