Examen

Fernando Coz

2023-07-15

Librerias

```
suppressMessages( library( readxl ) )
suppressMessages( library( performance ) )
suppressMessages( library( ggplot2 ) )
suppressMessages( library( MASS ) )
suppressMessages( library( olsrr ) )
suppressMessages( library( leaps ) )
suppressMessages( library( gamlss ) )
suppressMessages( library( readr ) )
suppressMessages( library( gridExtra ) )
suppressMessages( library( vcd ) )
suppressMessages( library( ROCR ) )
suppressMessages( library( ResourceSelection ) )
suppressMessages( library( MLmetrics ) )
suppressMessages( library( caret ) )
```

Funciones

```
test_supuestos <- function( my_model, nivel_significancia = 0.05 ) {

suppressMessages( require(lmtest) )
suppressMessages( require(car) )

  #Shapiro Test
  t_sha <- shapiro.test( my_model$residuals )
  t_sha_obs <- ifelse( t_sha$p.value > nivel_significancia,
                       "Los residuos son normales.",
                       "Los residuos NO son normales.")

  #BP Test
  t_bp <- bptest( my_model )
  t_bp_obs <- ifelse( unname(t_bp$p.value) > nivel_significancia,
                      "Homocedasticidad. La varianza de los residuos es constante.",
                      "Heterocedasticidad. La varianza de los residuos NO es constante.")

  #Durbin-Watson Test
  t_dwt <- durbinWatsonTest( my_model )
```

```
  t_dwt_obs <- ifelse( t_dwt$p > nivel_significancia,
                       "No hay autocorrelación. Los residuos son independientes.",
                       "Hay autocorrelación. Los residuos NO son independientes.")

  resultados <- data.frame(
    Prueba = c("Shapiro", "Breusch-Pagan", "Durbin-Watson"),
    P_Value = c(t_sha$p.value, unname(t_bp$p.value), t_dwt$p)
  )

  resultados$H0 <- ifelse( resultados$P_Value > nivel_significancia, "No rechazada", "Rechazada" )
  resultados$Observaciones <- ifelse( resultados$Prueba == "Shapiro", t_sha_obs,
                                      ifelse( resultados$Prueba == "Breusch-Pagan", t_bp_obs, t_dwt_obs )

  return(resultados)
}

test_supuestos_aov <- function( my_model, nivel_significancia = 0.05 ) {

suppressMessages( require(lmtest) )
suppressMessages( require(car) )

  #Shapiro Test
  t_sha <- shapiro.test( my_model$residuals )
  t_sha_obs <- ifelse( t_sha$p.value > nivel_significancia,
                       "Los residuos son normales.",
                       "Los residuos NO son normales.")

  #Levene Test
  t_l <- leveneTest( my_model )
  t_l_obs <- ifelse( unname(t_l$`Pr(>F)`[1]) > nivel_significancia,
                     "Homocedasticidad. La varianza de los residuos es constante.",
                     "Heterocedasticidad. La varianza de los residuos NO es constante.")

  resultados <- data.frame(
    Prueba = c("Shapiro", "Levene"),
    P_Value = c(t_sha$p.value, t_l$`Pr(>F)`[1])
  )

  resultados$H0 <- ifelse( resultados$P_Value > nivel_significancia, "No rechazada", "Rechazada" )
  resultados$Observaciones <- ifelse( resultados$Prueba == "Shapiro", t_sha_obs, t_l_obs )

  return(resultados)
}

outliers_influential_test <- function(model) {
  library(dplyr)

  fitted_values <- model$fitted.values
  residuals <- model$residuals
  hat_values <- lm.influence(model)$hat
  p <- length(model$coefficients)
  n <- length(residuals)
```

```
  # Bonferroni
  bonferroni <- list(which(abs(residuals) > qt(0.975, n-2) * sd(residuals)))

  # Cook's
  cook <- list(which(cooks.distance(model) > 4/(n - p - 1)))
  cook2 <- list(which(cooks.distance(model) > 4/n))

  # dfbetas
  dfbetas <- list(which(abs(dfbetas(model)) > 2/sqrt(n)))

  # dffits
  dffits <- list(unique(which(dffits(model) > 2 * sqrt(p/n))))
  dffits2 <- list(which(abs(dffits(model)) > 2 * sqrt(p * (n - p)/n)))

  # Leverage
  leverage_criteria_1 <- 0.2
  leverage_criteria_2 <- 2 * p/n
  leverage <- list(which(hat_values > leverage_criteria_1 | hat_values > leverage_criteria_2))

  bonferroni_str <- toString(unlist(bonferroni))
  cook_str <- toString(unlist(cook))
  cook_str2 <- toString(unlist(cook2))
  dfbetas_str <- toString(unlist(dfbetas))
  dffits_str <- toString(unlist(dffits))
  dffits_str2 <- toString(unlist(dffits2))
  leverage_str <- toString(unlist(leverage))

  results <- data.frame(
    Test_Criteria = c("Bonferroni", "Cook 1", "Cook 2", "DFBetas", "DFFits", "DFFits2", "Leverage"),
    Influential_Points = c(bonferroni_str, cook_str, cook_str2, dfbetas_str, dffits_str, dffits_str2, le
    stringsAsFactors = FALSE
  )

  return(results)
}
```

Ejercicio 1

En el archivo preciocasas.xlsx se han registrado respecto de 100 viviendas las siguientes variables:

- impuestos: valor de impuesto anual de la vivienda.
- dormitorios cantidad de ambientes de la vivienda.
- banios: cantidad de baños del inmueble.
- estrena: si es a estrenar.
- precio: valor del alquiler de la vivienda.
- tamanio: superficie total de la vivienda.

Dataset

```
casas <- read_excel("C:/Austral/mcd-reg-adv/datasets/preciocasas.xlsx")
```

1. Construir un modelo lineal simple para explicar el precio en función de la superficie y evaluar la bondad del ajuste.

```
modelo.casas <- lm(precio ~ tamanio, data=casas)
summary(modelo.casas)
```

```
##
## Call:
## lm(formula = precio ~ tamanio, data = casas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -236780  -29552   -2507   21639  151675
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -50926.255  14896.373  -3.419 0.000918 ***
## tamanio        126.594      8.468  14.951  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56190 on 98 degrees of freedom
## Multiple R-squared:  0.6952, Adjusted R-squared:  0.6921
## F-statistic: 223.5 on 1 and 98 DF,  p-value: < 2.2e-16
```

- El test de Wald para el coeficiente tamanio arroja un p value < 0.05.
- El test de la regresión tiene un p value < 0.05, lo que implica que el modelo es en su conjunto significativo.

2. Realizar un análisis diagnóstico y de puntos influyentes e indicar si el modelo es adecuado.

```
test_supuestos(modelo.casas)
```

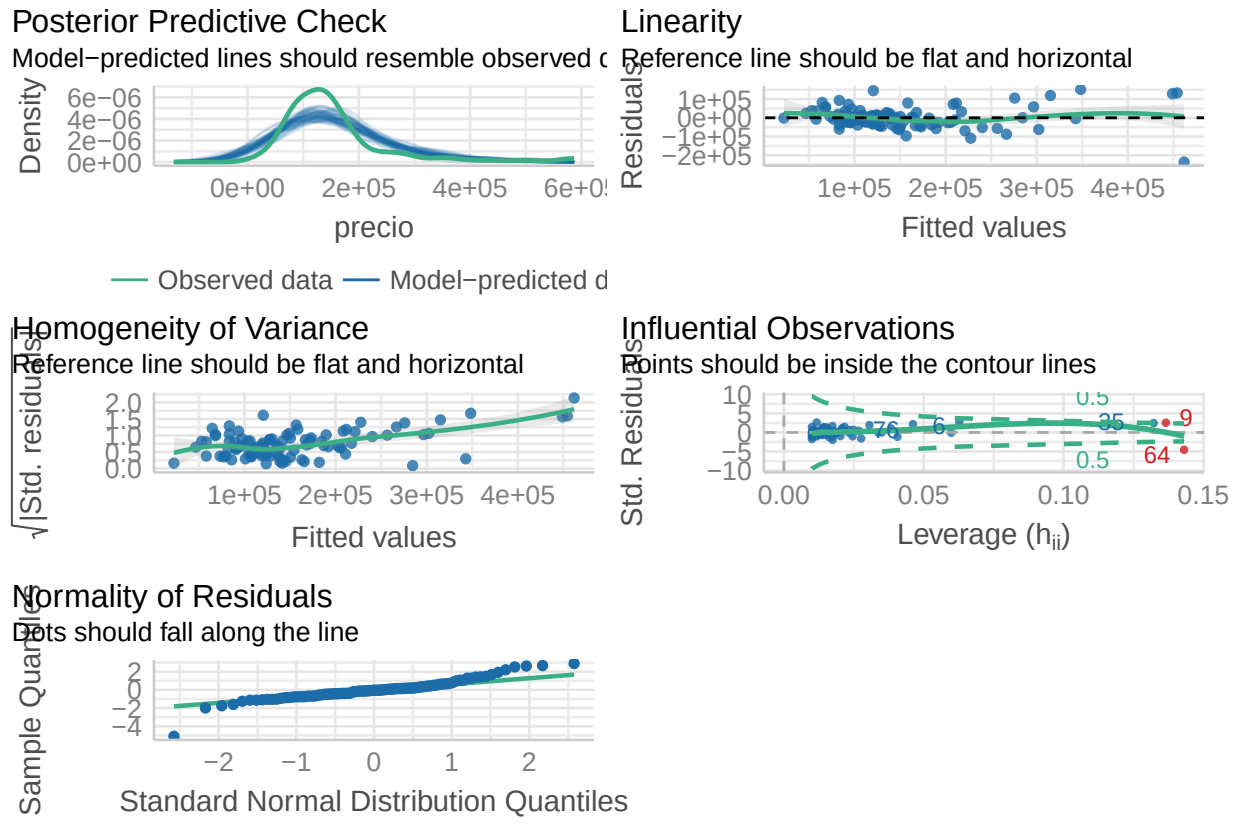
Analíticamente

```
##          Prueba      P_Value        H0
## 1       Shapiro 8.728976e-05 Rechazada
## 2 Breusch-Pagan 1.013482e-09 Rechazada
## 3 Durbin-Watson 8.000000e-03 Rechazada
##                                                    Observaciones
## 1                                 Los residuos NO son normales.
## 2 Heterocedasticidad. La varianza de los residuos NO es constante.
## 3         Hay autocorrelación. Los residuos NO son independientes.
```

```
check_model( modelo.casas )
```

Gráficamente

```
## Not enough model terms in the conditional part of the model to check for
##   multicollinearity.
```




```
outliers_influential_test(modelo.casas)
```

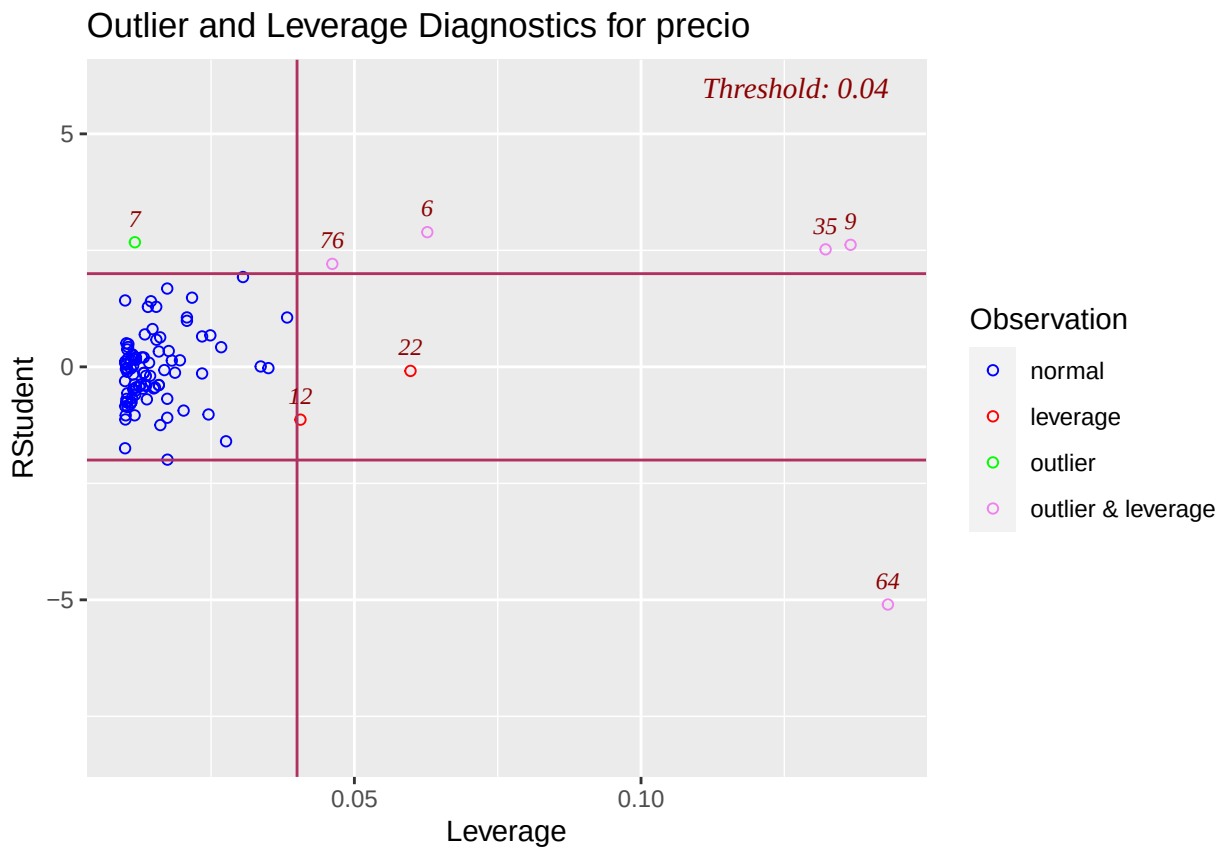
Analisis de outliers y puntos influyentes

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

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

## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following object is masked from 'package:nlme':
##
##     collapse

## The following object is masked from 'package:MASS':
##
##     select

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

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

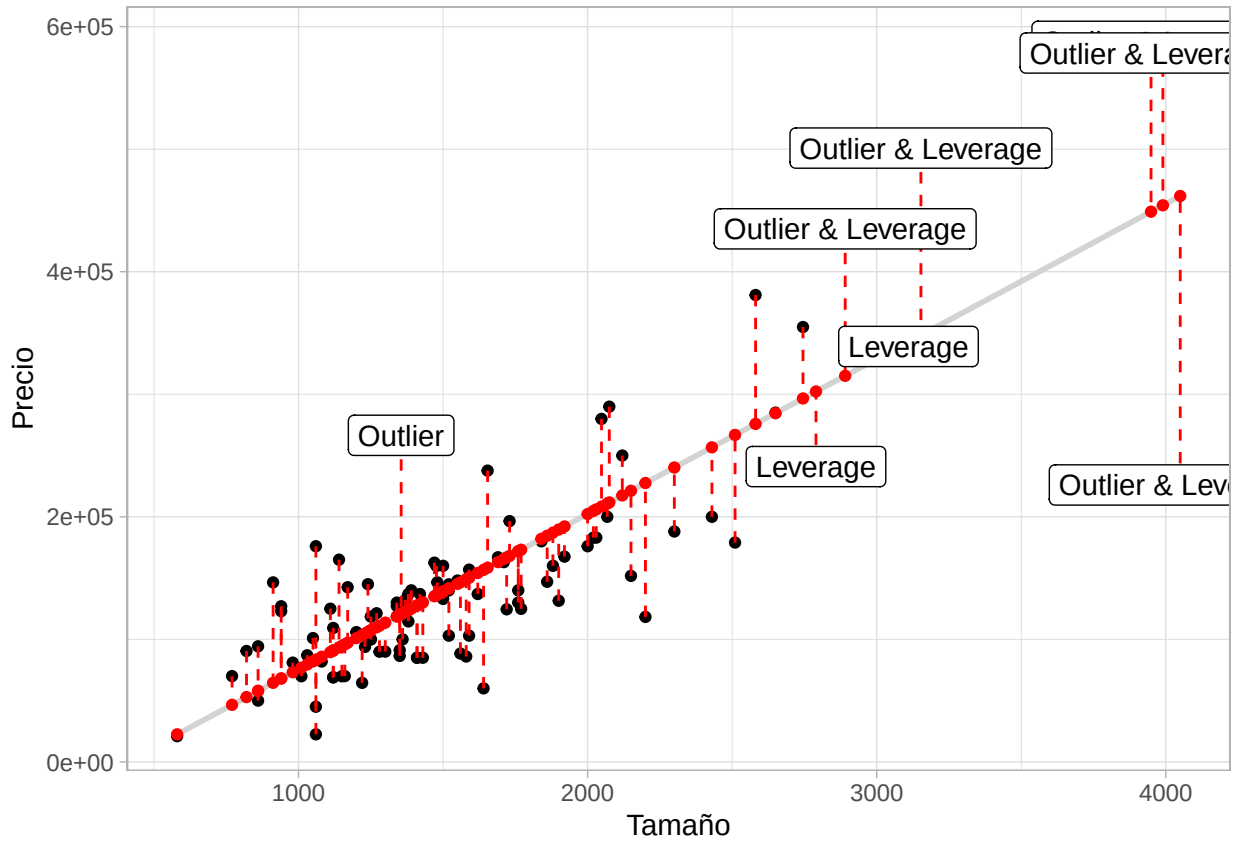
Test_Criteria	Influential_Points
1 Bonferroni	6, 7, 9, 35, 64, 76
2 Cook 1	6, 9, 35, 64, 66, 76
3 Cook 2	6, 9, 35, 64, 66, 76
4 DFBetas	2, 6, 7, 9, 35, 64, 76, 106, 109, 112, 121, 135, 164, 166, 176
5 DFFits	6, 7, 9, 35, 66, 76
6 DFFits2	
7 Leverage	6, 9, 12, 22, 35, 64, 76

```
ols_plot_resid_lev(modelo.casas, print_plot = TRUE)
```



```r
#Opcion 1 gráfico
datos <- casas
datos$predicciones <- predict(modelo.casas)

leverage <- hatvalues(modelo.casas)
std_residuals <- rstandard(modelo.casas)

datos$label <- NA

identify_label <- "Leverage"
datos$label[leverage > 2 * mean(leverage)] <- identify_label

identify_label <- "Outlier"
datos$label[abs(std_residuals) > 2] <- identify_label

identify_label <- "Outlier & Leverage"
datos$label[leverage > 2 * mean(leverage) & abs(std_residuals) > 2] <- identify_label

ggplot(datos, aes(x = tamanio, y = precio)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "lightgrey") +
  geom_segment(aes(xend = tamanio, yend = predicciones), col = "red", lty = "dashed") +
  geom_point(aes(y = predicciones), col = "red") +
  geom_label(aes(label = label), nudge_y = 1000) +
  labs(x = "Tamaño", y = "Precio") +
  theme_light()
```

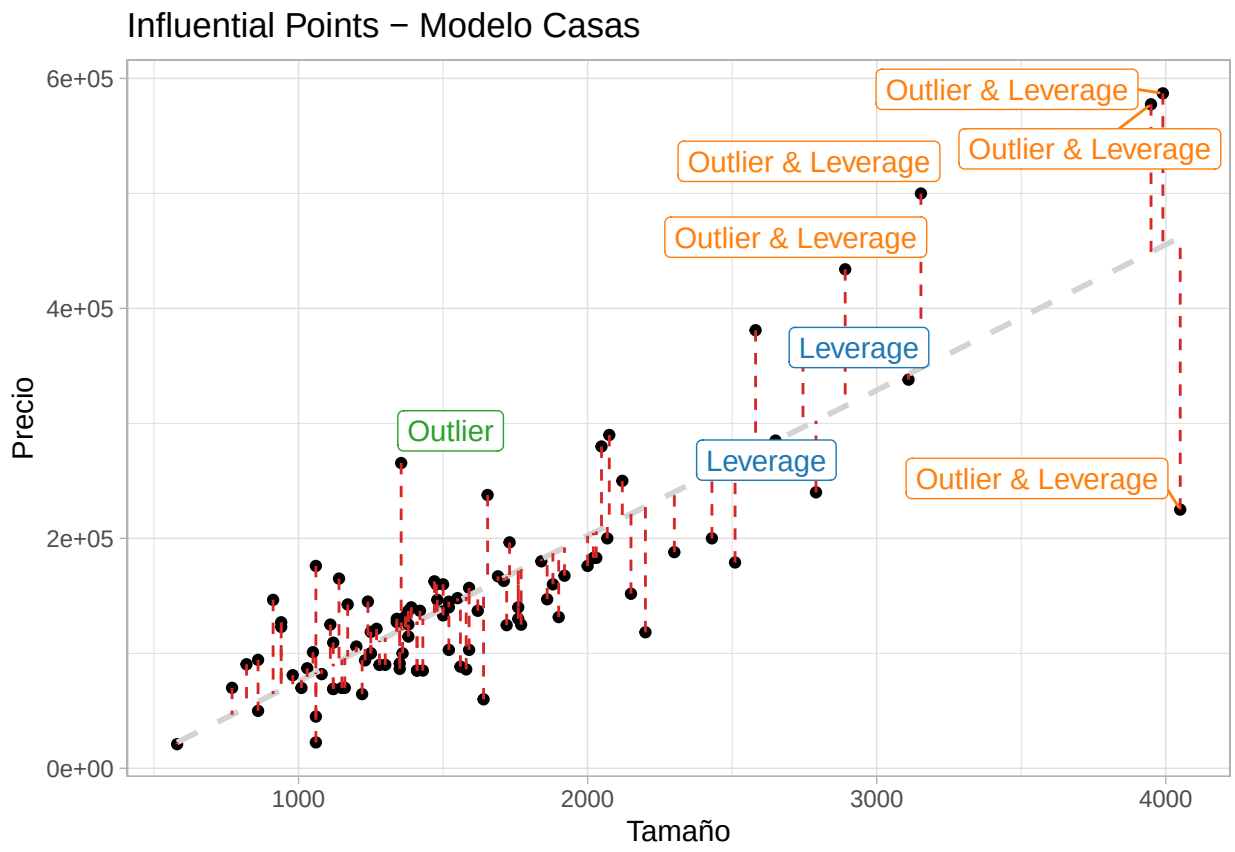
```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
#Opcion 2 gráfico
library(ggrepel)

datos <- casas
datos$predicciones <- predict(modelo.casas)

leverage <- hatvalues(modelo.casas)
std_residuals <- rstandard(modelo.casas)

datos$label <- NA

identify_label <- "Leverage"
datos$label[leverage > 2 * mean(leverage)] <- identify_label

identify_label <- "Outlier"
datos$label[abs(std_residuals) > 2] <- identify_label

identify_label <- "Outlier & Leverage"
datos$label[leverage > 2 * mean(leverage) & abs(std_residuals) > 2] <- identify_label

ggplot(datos, aes(x = tamanio, y = precio)) +
  ggtitle("Influential Points - Modelo Casas")+
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "lightgrey", linetype = "dashed") +
  geom_segment(aes(xend = tamanio, yend = predicciones), col = "#d62728", lty = "dashed") +
  #geom_point(aes(y = predicciones), col = "lightgrey") +
```

```
  geom_label_repel(aes(label = label, color = label), nudge_y = 1000, show.legend = FALSE) +
  labs(x = "Tamaño", y = "Precio") +
  scale_color_manual(values = c("Leverage" = "#1f77b4", "Outlier" = "#2ca02c", "Outlier & Leverage" = "#
  theme_light()
```

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



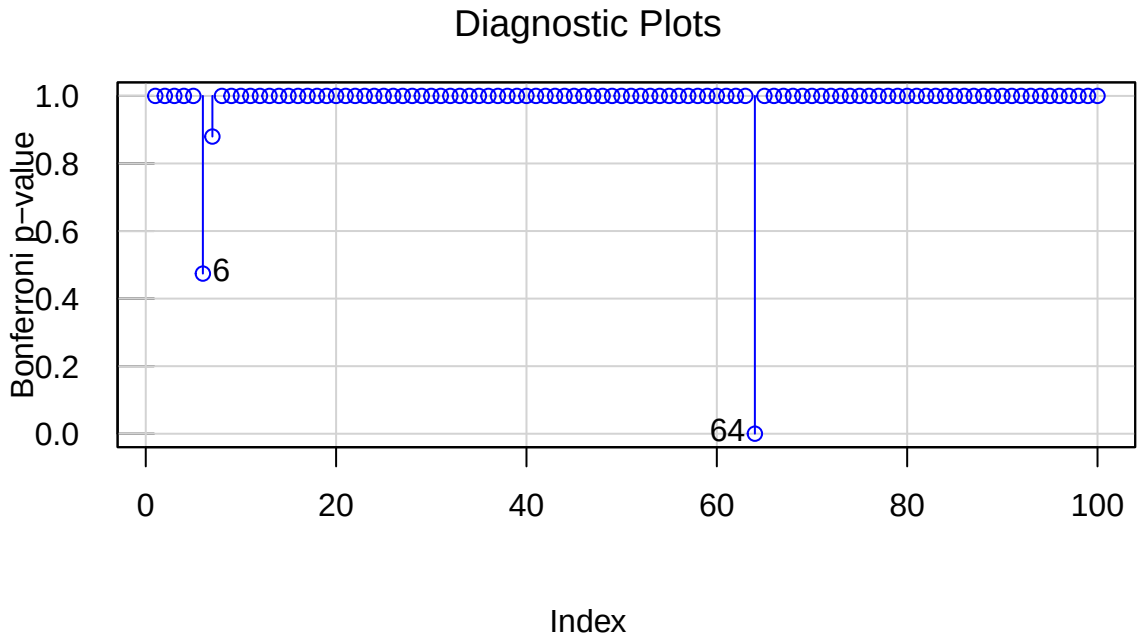

```
# Test de Bonferroni
outlierTest( modelo.casas )
```

OUTLIERS

```
##     rstudent unadjusted p-value Bonferroni p
## 64 -5.099962         1.6833e-06   0.00016833
```

La observación 64 quedó señalada como valor extremo u outlier. Viendolo gráficamente:

```
influenceIndexPlot( modelo.casas, vars="Bonf", las=1, col="blue")
```




```
summary( influence.measures(model = modelo.casas) )
```

PUNTOS INFLUYENTES

```
## Potentially influential observations of
##   lm(formula = precio ~ tamanio, data = casas) :
##
##    dfb.1_  dfb.tamn dffit   cov.r   cook.d  hat
## 6  -0.52    0.69     0.75_*  0.92_*  0.26    0.06_*
## 7   0.20   -0.11     0.29    0.90_*  0.04    0.01
## 9  -0.82    1.00_*   1.04_*  1.03    0.51    0.14_*
## 22  0.02   -0.02    -0.02    1.09_*  0.00    0.06
## 35 -0.77    0.95     0.98_*  1.04    0.46    0.13_*
## 64  1.65_* -2.01_*  -2.08_*  0.74_*  1.73_*  0.14_*
## 76 -0.31    0.43     0.49_*  0.97    0.11    0.05
```

Analizando la salida aparecen señalados distintos puntos bajo distintos criterios, como puntos influyentes: 6,7,9,22,35,64,76

Distancia de Cook

```r
n<-length(casas$caso)
p<-length(modelo.casas$coefficients)

dcook<-cooks.distance(modelo.casas)
influyentes <- unique(which(dcook>4/n))
influyentes
```

```
## [1]  6  9 35 64 66 76
```

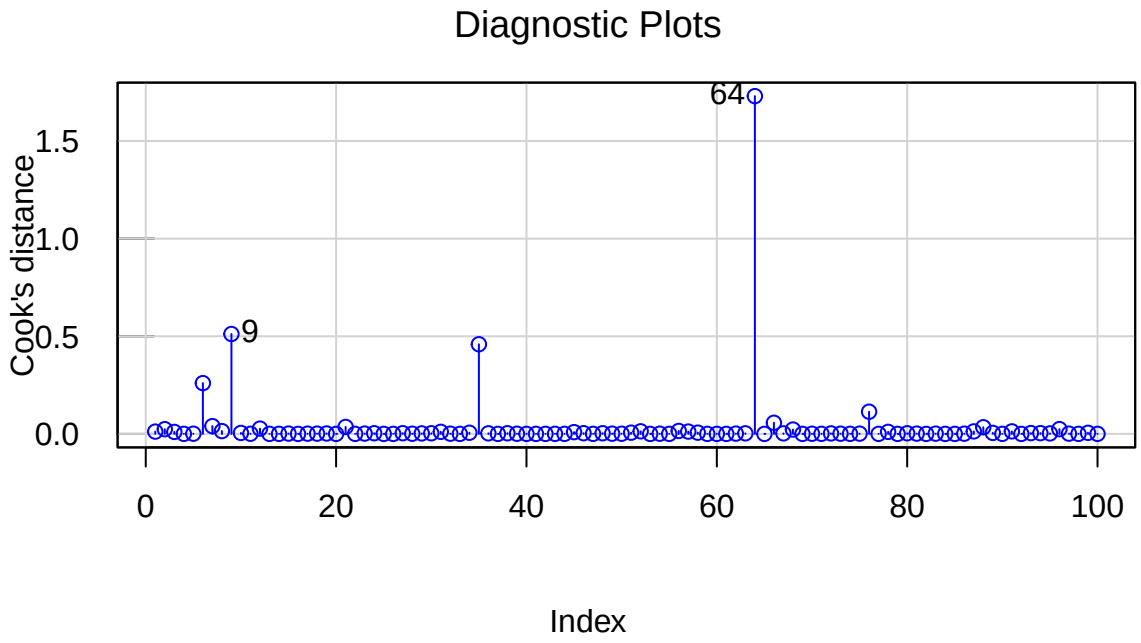
Usando un criterio mas exigente:

```r
corted<-qf(0.5,2,n-2)
unique(which(dcook>corted))
```

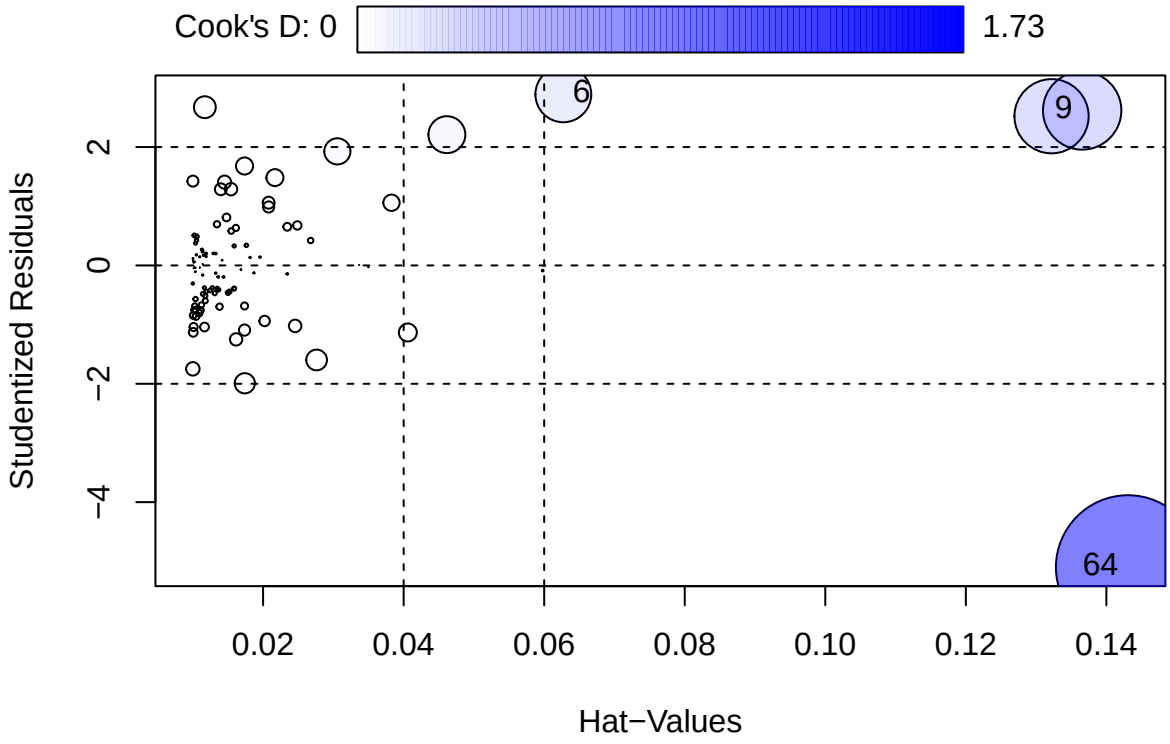
```
## [1] 64
```

Visualizandolo gráficamente

```r
influenceIndexPlot( modelo.casas, vars="Cook", las=1, col="blue")
```




```r
influencePlot( model = modelo.casas )
```




```
##       StudRes        Hat     CookD
## 6    2.890934 0.06272291 0.2601149
## 9    2.618641 0.13655462 0.5116615
## 64 -5.099962 0.14306939 1.7297845
```

DFFITS y DFBETA

El punto 9 se corresponde con un valor superior de corte (1) y queda señalado como punto influyente.

```
unique(which(dfbetas(modelo.casas)[,2]>1))
```

```
## [1] 9
```

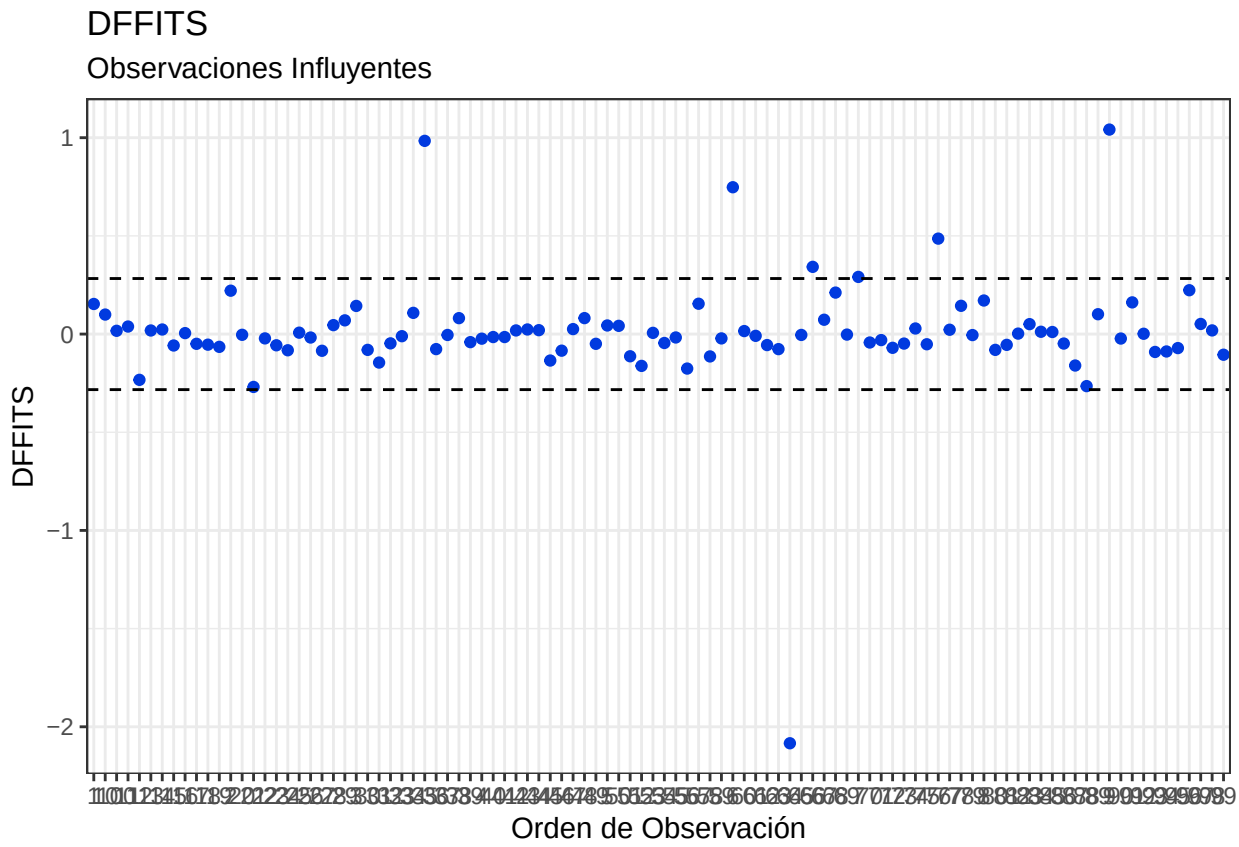
Usando el criterio de DFFITS, y el punto de corte 2 * sqrt(p/n) aparecen otras observaciones: 6,7,9,35,66,76

```
unique(which(dffits(modelo.casas)>2 * sqrt(p / n)))
```

```
## [1]  6  7  9 35 66 76
```

Si vemos el DFFITS gráficamente

```
df <- modelo.casas$df.residual

dffits_crit = 2 * sqrt( p/n )
dffits <- dffits( modelo.casas )

df <- data.frame( obs = names( dffits ), dffits = dffits )

ggplot( df, aes( y = dffits, x = obs ) ) +
    geom_point( color = '#013ADF' ) +
    geom_hline( yintercept = c( dffits_crit, -dffits_crit ),
        linetype = 'dashed' ) +
    labs( title = 'DFFITS',
        subtitle = 'Observaciones Influyentes',
        x = 'Orden de Observación',
        y = 'DFFITS') +
    theme_bw()
```



No aparecen valores influyentes con

```
lev <- hatvalues(modelo.casas)
unique(which(lev>0.2))
```

```
## integer(0)
```

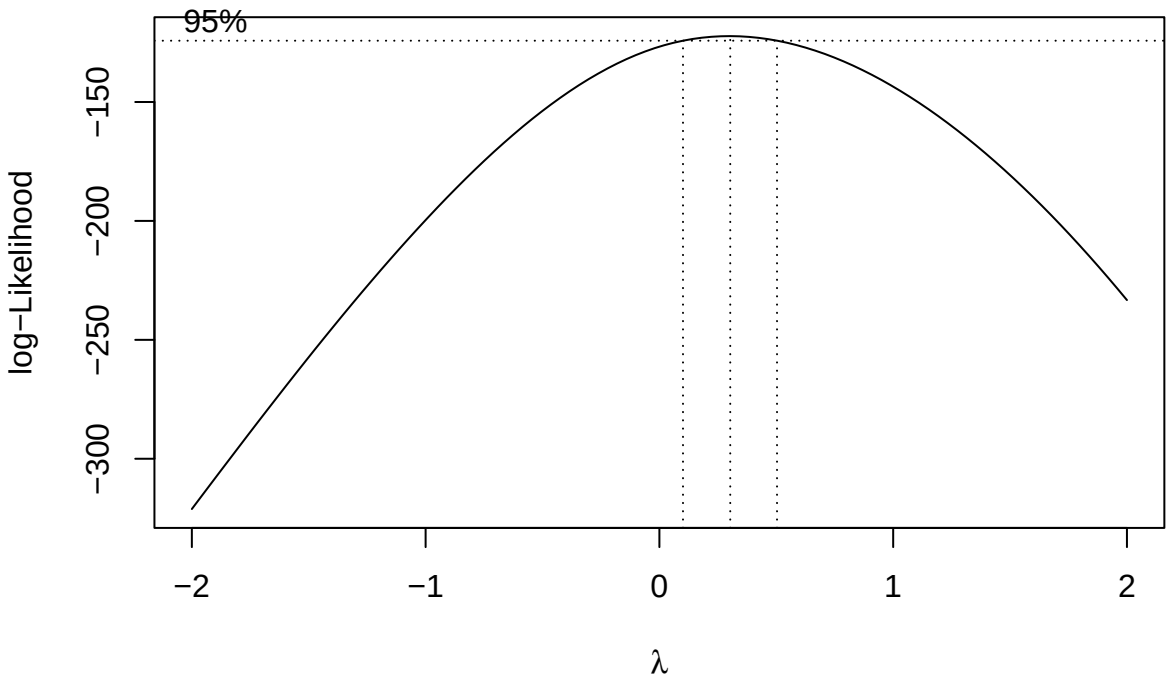
```r
lev <- hatvalues(modelo.casas)
unique(which(lev>2*p/n))
```

```
## [1]  6  9 12 22 35 64 76
```

Finalmente, las observaciones 6, 9 y 64 son identificadas como influyentes bajo distintos criterios y puntos de cortes.

3. Realizar una transformación de la variable respuesta para lograr normalidad en la distribución de los residuos. Indicar si el modelo con esta transformación resulta adecuado.

```r
bc <- boxcox( precio ~ tamanio, lambda = -2:2, data = casas )
```




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

```
## [1] 0.3030303
```

```r
#Arreglar la transformación
modelo.casas2 <- lm( ( ( precio^(lambda) - 1 ) / lambda ) ~ tamanio, data = casas )
summary( modelo.casas2 )
```

```
##
## Call:
## lm(formula = ((precio^(lambda) - 1)/lambda) ~ tamanio, data = casas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.941  -7.268   0.492   5.873  32.797
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 75.166760   3.200453   23.49   <2e-16 ***
## tamanio      0.025095   0.001819   13.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.07 on 98 degrees of freedom
## Multiple R-squared:  0.6601, Adjusted R-squared:  0.6566
## F-statistic: 190.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
test_supuestos(modelo.casas2)
```

```
##          Prueba    P_Value           H0
## 1       Shapiro 0.02278615    Rechazada
## 2 Breusch-Pagan 0.10929343 No rechazada
## 3 Durbin-Watson 0.00000000    Rechazada
##                                                 Observaciones
## 1                              Los residuos NO son normales.
## 2 Homocedasticidad. La varianza de los residuos es constante.
## 3    Hay autocorrelación. Los residuos NO son independientes.
```

Realizando nuevamente un análisis diagnóstico, vemos que pese a realizar la transformación por boxcox los residuos siguen sin ser normales.

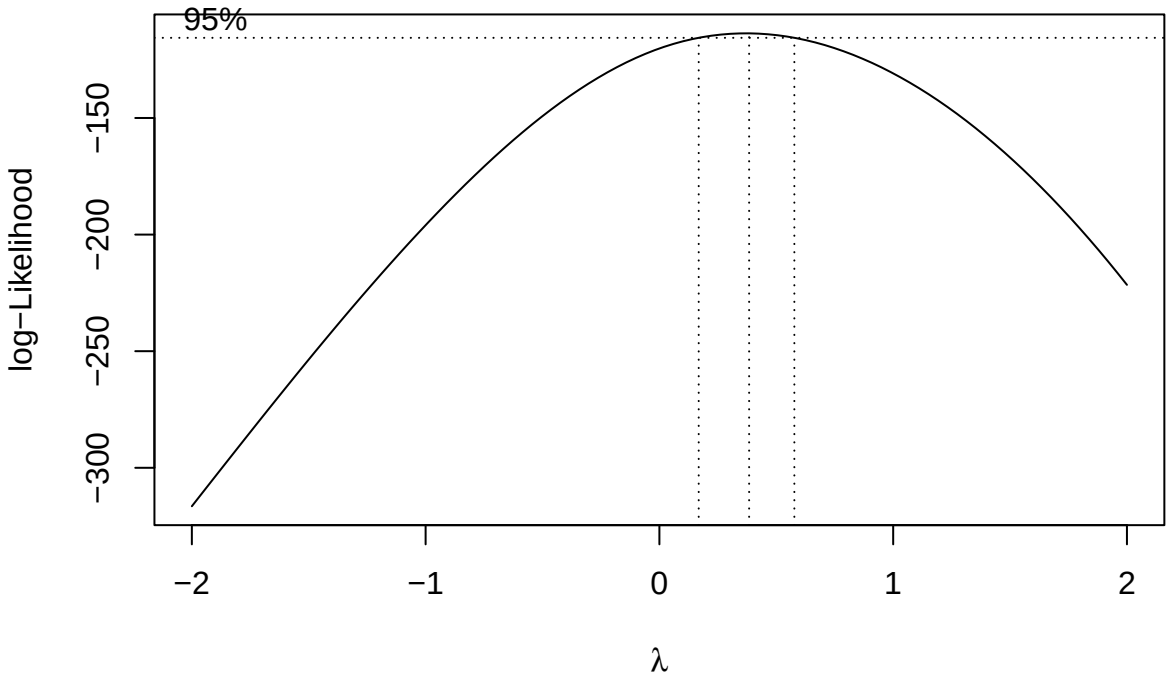
4. Eliminar la observación 64 y ajustar nuevamente el segundo modelo evaluando su validez.

Elimino la observación 64 del dataset

```
casa_clean <- casas[casas$caso != 64, ]
```

Volvemos a ajustar la transformación de boxcox con el nuevo dataset y obtener el nuevo valor de lambda

```
bc <- boxcox( precio ~ tamanio, lambda = -2:2, data = casa_clean )
```




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

```
## [1] 0.3838384
```

Ajustamos nuevamente un modelo lineal con el nuevo set de datos y valor de lambda:

```
modelo.casas3 <- lm( ( ( precio^(lambda) - 1 ) / lambda ) ~ tamanio, data = casa_clean )
summary( modelo.casas3 )
```

```
##
## Call:
## lm(formula = ((precio^(lambda) - 1)/lambda) ~ tamanio, data = casa_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -83.638 -21.460   1.613  14.213  87.133
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.245e+02  8.068e+00   15.43   <2e-16 ***
## tamanio     7.411e-02  4.689e-03   15.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 28.95 on 97 degrees of freedom
## Multiple R-squared:  0.7203, Adjusted R-squared:  0.7174
## F-statistic: 249.8 on 1 and 97 DF,  p-value: < 2.2e-16
```

Finalmente comprobamos que luego de eliminar la observación 64 no rechazamos la hipotesis de normalidad del test de Shapiro al igual que el test de homocedasticidad, con lo cual podemos concluir que dicha observación era influyente y aumentaba los residuos.

```
test_supuestos(modelo.casas3)
```

```
##          Prueba   P_Value           H0
## 1       Shapiro 0.7888461 No rechazada
## 2 Breusch-Pagan 0.4268116 No rechazada
## 3 Durbin-Watson 0.0060000    Rechazada
##                                                 Observaciones
## 1                                 Los residuos son normales.
## 2 Homocedasticidad. La varianza de los residuos es constante.
## 3    Hay autocorrelación. Los residuos NO son independientes.
```

5. Ajustar un modelo robusto y evaluar el promedio de los errores absolutos cometidos. Comparar con el mejor modelo lineal disponible.

```
modelo.casas.robusto <- rlm(precio ~ tamanio, data = casa_clean, psi=psi.huber)
summary(modelo.casas.robusto)
```

```
##
## Call: rlm(formula = precio ~ tamanio, data = casa_clean, psi = psi.huber)
## Residuals:
##       Min        1Q    Median        3Q       Max
## -107695.6  -25312.4     209.5   25273.5  151589.5
##
## Coefficients:
##             Value       Std. Error  t value
## (Intercept) -56368.4803  12704.8695     -4.4368
## tamanio        128.3473      7.3839     17.3820
##
## Residual standard error: 38100 on 97 degrees of freedom
```

```
indices_validacion <- sample(nrow(casa_clean), round(0.2 * nrow(casa_clean)))
datos_entrenamiento <- casa_clean[-indices_validacion, ]
datos_validacion <- casa_clean[indices_validacion, ]

modelo.lm <- lm(( ( precio^(lambda) - 1 ) / lambda ) ~ tamanio, data = datos_entrenamiento)
pred.lm <- predict(modelo.lm, newdata = datos_validacion)

#library(caret)
#RMSE(pred.lm, datos_validacion$precio)
lm.resid <- pred.lm - datos_validacion$precio
lm.rmse <- sqrt(mean(lm.resid^2))
```

```r
modelo.rlm <- rlm( precio ~ tamanio, data = datos_validacion, psi=psi.huber)
pred.rlm <- predict(modelo.rlm, newdata = datos_validacion)

#RMSE(pred.rlm, datos_validacion$precio)
rlm.resid <- pred.rlm - datos_validacion$precio
rlm.rmse <- sqrt(mean(rlm.resid^2))

print(paste("LM RSME:", round(lm.rmse,2)))
```

```
## [1] "LM RSME: 136951.87"
```

```r
print(paste("RLM RSME:", round(rlm.rmse,2)))
```

```
## [1] "RLM RSME: 32486.1"
```

Un valor menor de RSME sugiere que el modelo robusto es mejor que el modelo lineal prediciendo la variable dependiente.

6. Utilizar un método de selección de variables para proponer un modelo multivariado. Analizar el cumplimiento de los supuestos.

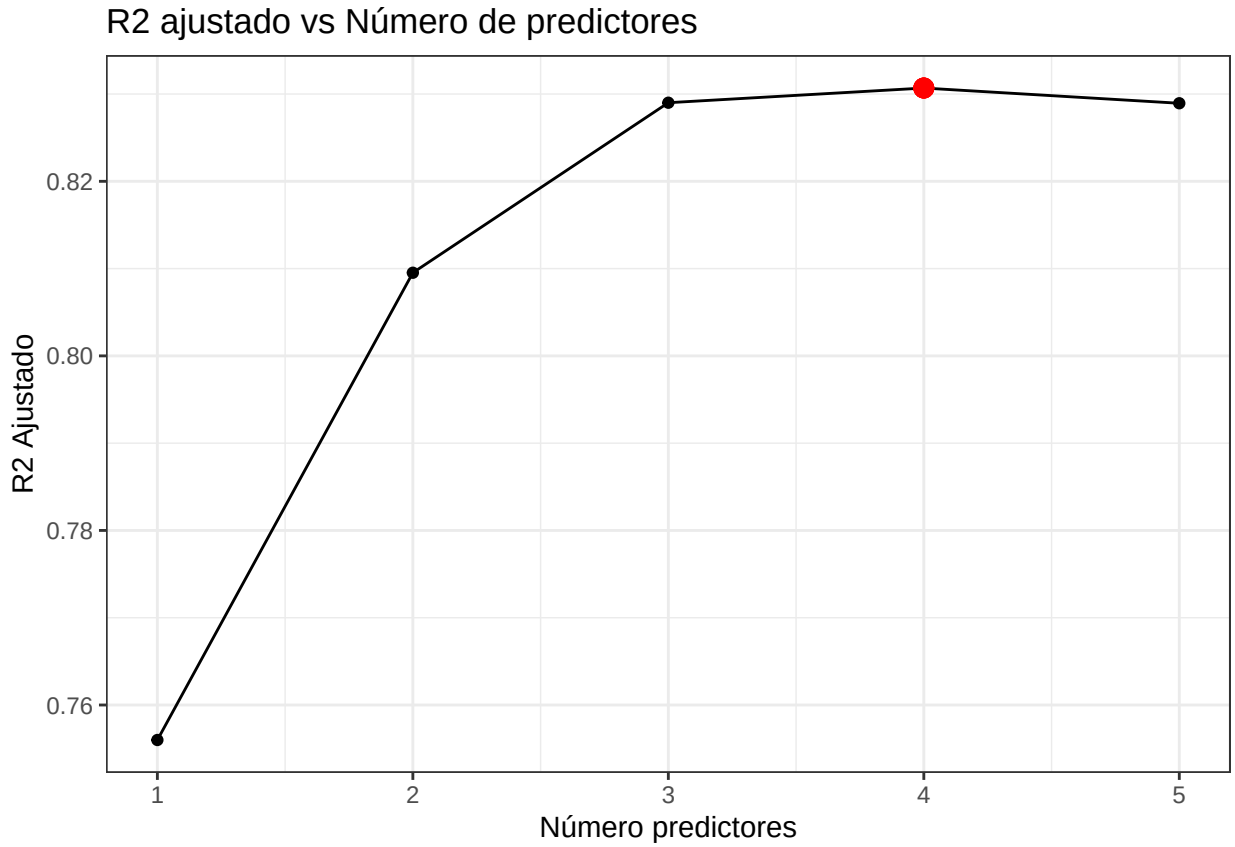
```r
modelo.casas.comb <- regsubsets(precio ~ ., data = casa_clean[,2:7], nvmax = 5)
summary(modelo.casas.comb)
```

```
## Subset selection object
## Call: regsubsets.formula(precio ~ ., data = casa_clean[, 2:7], nvmax = 5)
## 5 Variables  (and intercept)
##             Forced in Forced out
## impuestos       FALSE      FALSE
## dormitorios     FALSE      FALSE
## banios          FALSE      FALSE
## estrena         FALSE      FALSE
## tamanio         FALSE      FALSE
## 1 subsets of each size up to 5
## Selection Algorithm: exhaustive
##          impuestos dormitorios banios estrena tamanio
## 1  ( 1 ) " "       " "         " "    " "     "*"
## 2  ( 1 ) "*"       " "         " "    " "     "*"
## 3  ( 1 ) "*"       "*"         " "    " "     "*"
## 4  ( 1 ) "*"       "*"         " "    "*"     "*"
## 5  ( 1 ) "*"       "*"         "*"    "*"     "*"
```

```r
adjr2_values <- summary(modelo.casas.comb)$adjr2
best_model_adjr2 <- which.max(summary(modelo.casas.comb)$adjr2)
```

```r
p <- ggplot(data = data.frame(n_predictores = 1:5,
            R_ajustado = summary(modelo.casas.comb)$adjr2),
            aes( x = n_predictores, y = R_ajustado) ) +
            geom_line() +
```

```
            geom_point()

p <- p + geom_point( aes( x=n_predictores[which.max(summary(modelo.casas.comb)$adjr2)],
                          y=R_ajustado[which.max(summary(modelo.casas.comb)$adjr2)]),
                          colour = "red", size = 3 )

p <- p + scale_x_continuous(breaks = c(0:14)) + theme_bw() +
  labs(title = "R2 ajustado vs Número de predictores", x = "Número predictores", y = "R2 Ajustado")

p
```




```
cat( "R2 Ajustado modelo 2: ", adjr2_values[2], "\n")
```

```
## R2 Ajustado modelo 2:  0.8095246
```

```
cat( "R2 Ajustado modelo 3: ", adjr2_values[3], "\n")
```

```
## R2 Ajustado modelo 3:  0.8290052
```

```
cat( "R2 Ajustado modelo 4: ", adjr2_values[4], "\n")
```

```
## R2 Ajustado modelo 4:  0.8306874
```

Dado que la mejora entre el modelo con 3 y 4 variables no es significativa, y siguiendo con el principio de parsimonia elijo el modelo con la menora cantiidad de variables, es decir el modelo 3.

```
modelo.casas.final <- lm( precio ~ tamanio + impuestos + dormitorios, data=casa_clean )
summary(modelo.casas.final)
```

```
##
## Call:
## lm(formula = precio ~ tamanio + impuestos + dormitorios, data = casa_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -113125  -22396     911   21919  145922
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5868.483  20349.140   0.288 0.773677
## tamanio        104.178     12.762   8.163 1.36e-12 ***
## impuestos       34.193      6.056   5.646 1.70e-07 ***
## dormitorios -27612.823   7992.197  -3.455 0.000824 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41980 on 95 degrees of freedom
## Multiple R-squared:  0.8342, Adjusted R-squared:  0.829
## F-statistic: 159.4 on 3 and 95 DF,  p-value: < 2.2e-16
```

```
test_supuestos(modelo.casas.final)
```

```
##          Prueba      P_Value           H0
## 1       Shapiro 1.912590e-01 No rechazada
## 2 Breusch-Pagan 7.512097e-05    Rechazada
## 3 Durbin-Watson 8.000000e-03    Rechazada
##                                                    Observaciones
## 1                                    Los residuos son normales.
## 2 Heterocedasticidad. La varianza de los residuos NO es constante.
## 3         Hay autocorrelación. Los residuos NO son independientes.
```

Podemos observar que se cumple el supuesto de normalidad pero no así el de heterocedasticidad.

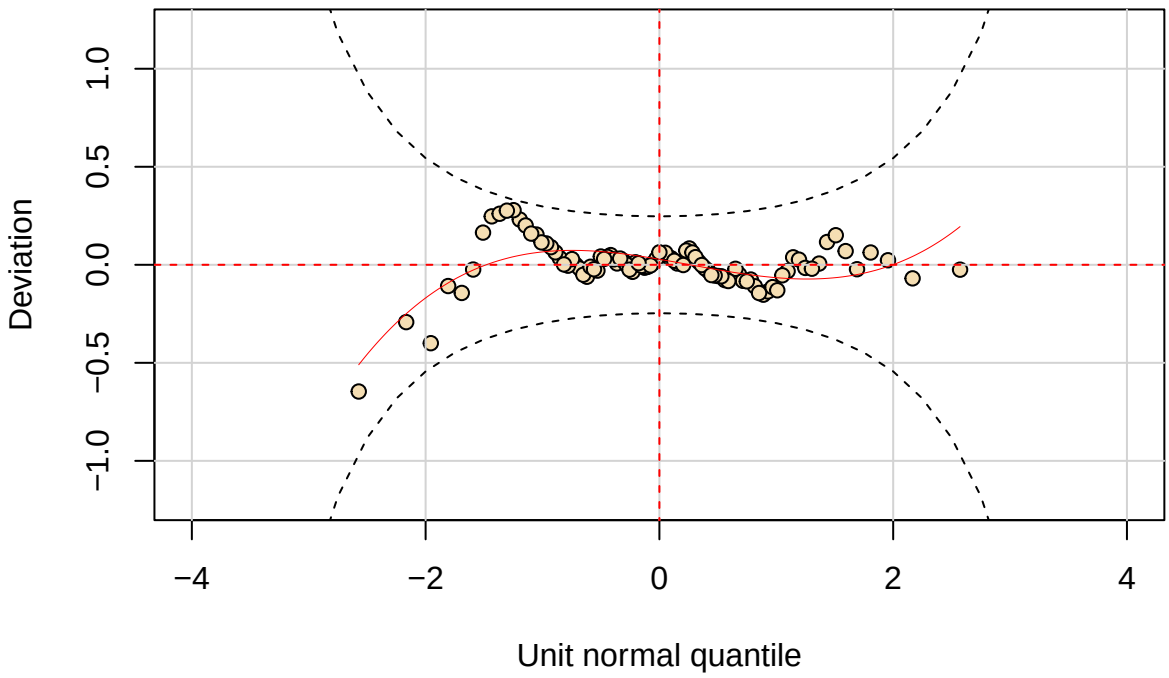
7. Le parece adecuado un modelo GAMLSS en este caso? Justifique.

Tal como se observa en el punto anterior, y quedando en evidencia que no se cumple el supuesto de homocedasticidad, resultaría útil modelar usando un modelo GAMLSS.

```
modelo.casas.gamlss <- gamlss( formula = precio ~ pb(tamanio) + pb(impuestos) + pb(dormitorios),
    #sigma.formula = ~ pb(tamanio) + pb(impuestos) + pb(dormitorios),
    family = GA,
    data = casa_clean[2:7],
    trace = FALSE )

summary(modelo.casas.gamlss)
```

```
## Warning in summary.gamlss(modelo.casas.gamlss): summary: vcov has failed, option qr is used instead

## *******************************************************************
## Family:  c("GA", "Gamma")
##
## Call:
## gamlss(formula = precio ~ pb(tamanio) + pb(impuestos) + pb(dormitorios),
##     family = GA, data = casa_clean[2:7], trace = FALSE)
##
## Fitting method: RS()
##
## ------------------------------------------------------------------
## Mu link function:  log
## Mu Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.103e+01  1.209e-01  91.206  < 2e-16 ***
## pb(tamanio)      4.662e-04  7.582e-05   6.149 2.01e-08 ***
## pb(impuestos)    2.061e-04  3.598e-05   5.728 1.28e-07 ***
## pb(dormitorios) -1.144e-01  4.748e-02  -2.408    0.018 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------------------------------
## Sigma link function:  log
## Sigma Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.38857    0.07034  -19.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------------------------------
## NOTE: Additive smoothing terms exist in the formulas:
##  i) Std. Error for smoothers are for the linear effect only.
## ii) Std. Error for the linear terms may not be reliable.
## ------------------------------------------------------------------
## No. of observations in the fit:  99
## Degrees of Freedom for the fit:  8.436531
##       Residual Deg. of Freedom:  90.56347
##                       at cycle:  3
##
## Global Deviance:     2342.141
##             AIC:     2359.014
##             SBC:     2380.908
## *******************************************************************
```

```
wp( modelo.casas.gamlss )
```



Vemos que el modelo tiene los residuos dentro del rando de variación aceptable.

```
# modelo_OLS
#mod OLS <- gamlss( formula = valor metros + anio + calef +
#local,
#family = NO, data = datos, trace = FALSE)
#summary(mod OLS)
#GAIC(modelo.casas.final, modelo.casas.gamlss)
```

8. Resuma sus conclusiones

1. Normalidad: Comenzamos evaluando y trabajando sobre la normalidad. Dado que no se cumplia el supuesto, se trabajó con una transformación hasta llegar a un modelo donde se cumpliera dicho supuesto.
2. Residuos: En la segunda etapa se hizo énfasis en los residuos y dado que no se cumplia el supuesto trabajamos con modelos especificos para casos donde la varianza no se es constante. En resumen, fuimos trabajando en cada area y corrigiendo aplicando métodos y modelos para disminuir la influencia de estas alteraciones.

Ejercicio 2

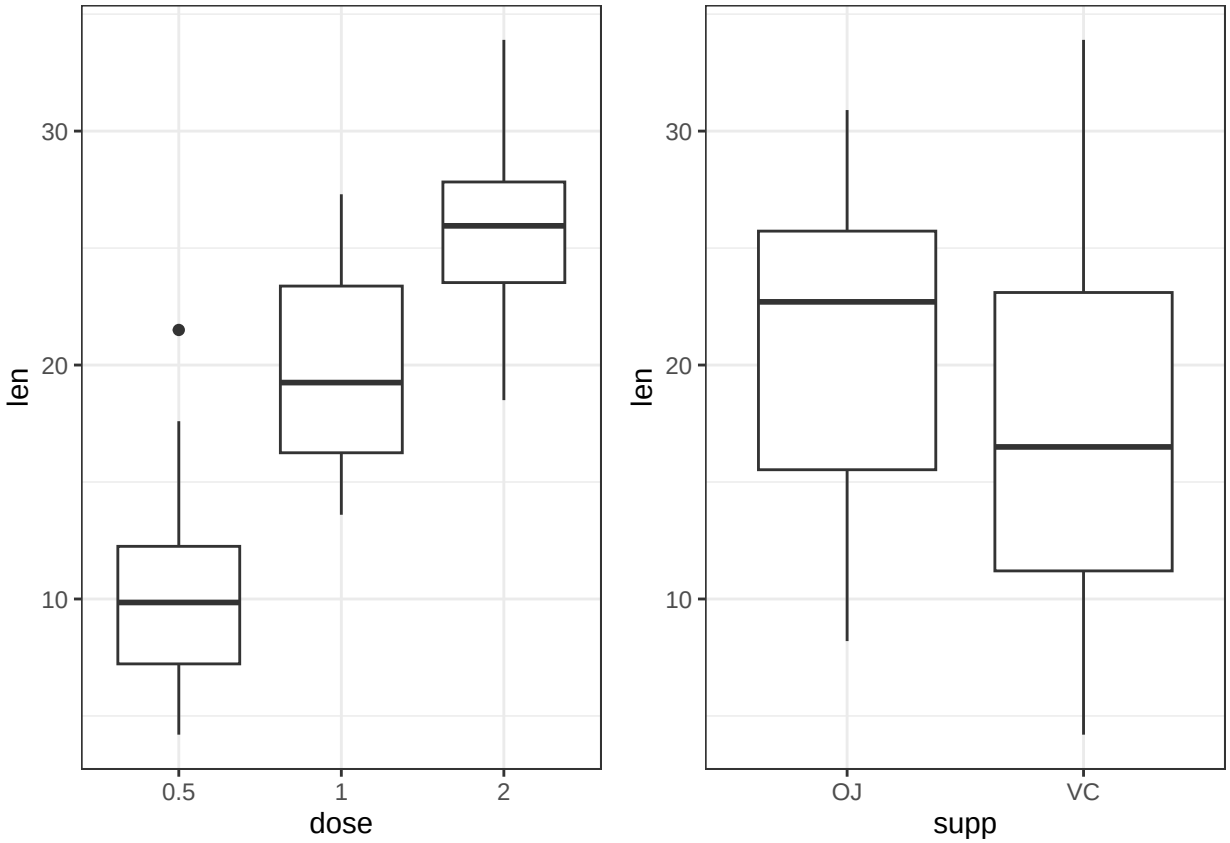
Se desea saber si la dosis de ácido ascórbico y el tipo de bebida en la cual se lo administró a ciertos animales de laboratorio logró mayor desarrollo de los dientes en los mismo. Se utilizaron 60 replicaciones del experimento y se tienen grupos balanceados. La variable respuesta de interés es la longitud de los dientes frontales(len). Los resultados están en el archivo odonto.csv Se pide analizar, analítica y gráficamente, si:

Dataset

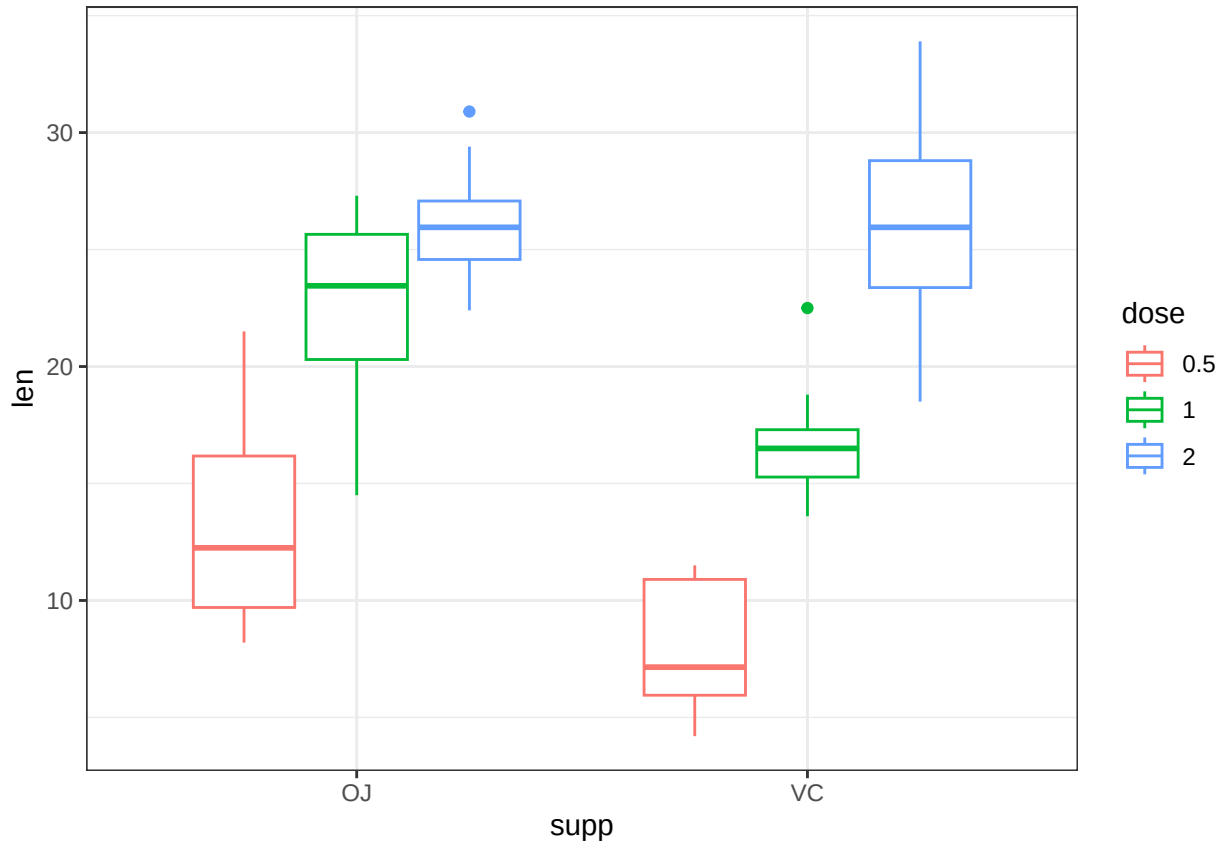
```
odonto <- read.csv("C:/Austral/mcd-reg-adv/datasets/odonto.csv")
```

1. ¿Existen diferencias estadísticamente significativas respecto de las dosis administradas?

```
odonto$dose <- factor(odonto$dose)

p1 <- ggplot(data = odonto, mapping = aes(x = dose, y = len)) + geom_boxplot() +
  theme_bw()
p2 <- ggplot(data = odonto, mapping = aes(x = supp, y = len)) + geom_boxplot() +
  theme_bw()
p3 <- ggplot(data = odonto, mapping = aes(x = supp, y = len, colour = dose)) +
  geom_boxplot() + theme_bw()

grid.arrange(p1, p2, ncol = 2)
```




```
p3
```

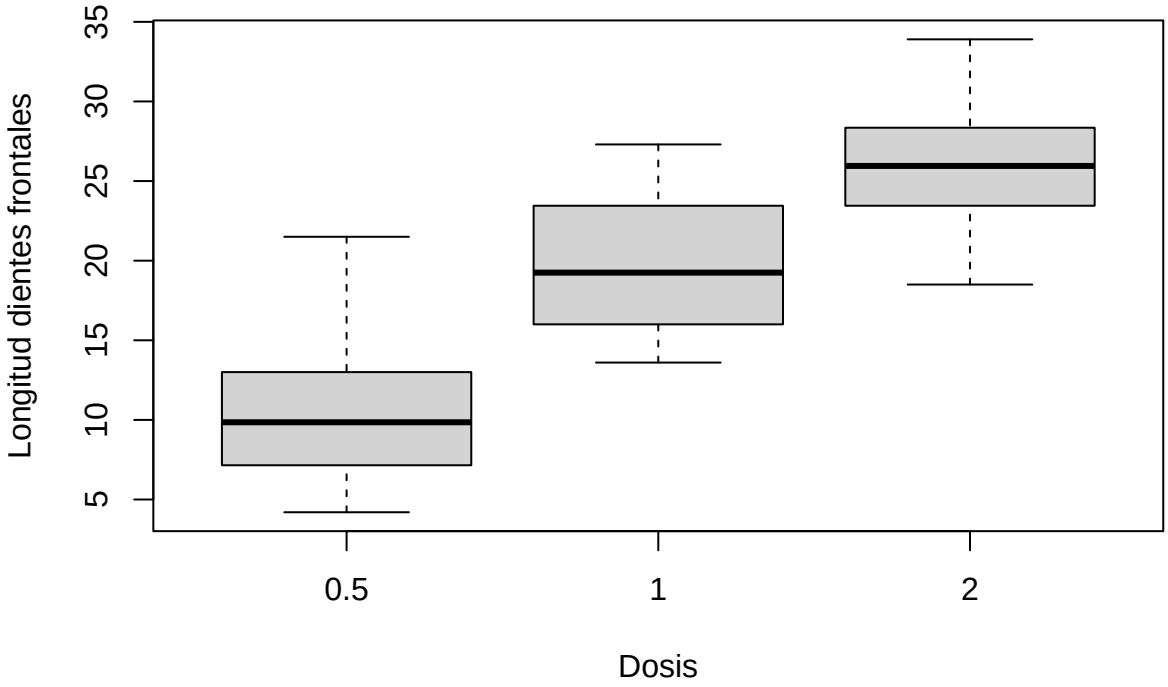



```
aov_odonto <- aov(formula = len ~ dose, data = odonto)
summary(aov_odonto)
```

Analíticamente

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## dose         2   2426    1213   67.42 9.53e-16 ***
## Residuals   57   1026      18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
boxplot(len ~ dose, data = odonto, xlab = "Dosis", ylab = "Longitud dientes frontales")
```



Gráficamente

Podemos corrobarar tanto analíticamente como gráficamente que existe diferencias estadísticamente significativas respecto de las dosis administradas para la longitud de crecimiento de los dientes.

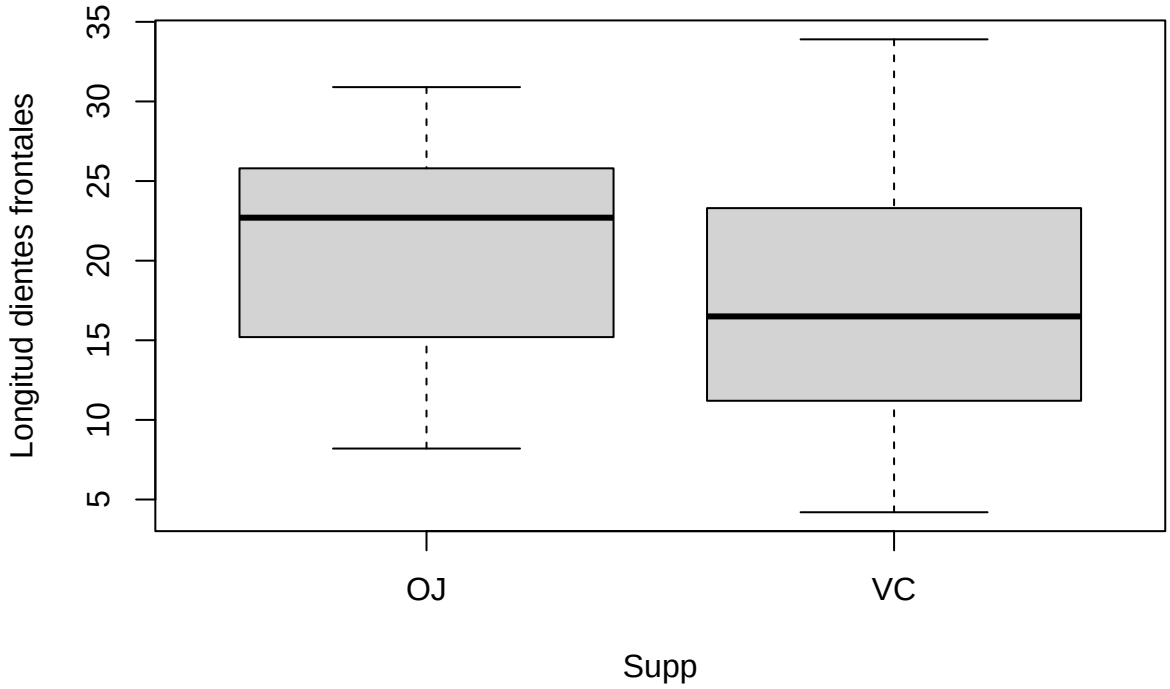
2. ¿Existen diferencias estadísticamente singificativas respeccto del tipo de vehículo de administración?

```
aov_odonto_supp <- aov(formula = len ~ supp, data = odonto)
summary(aov_odonto_supp)
```

Analíticamente

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## supp         1    205  205.35   3.668 0.0604 .
## Residuals   58   3247   55.98
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
boxplot(len ~ supp, data = odonto, xlab = "Supp", ylab = "Longitud dientes frontales")
```



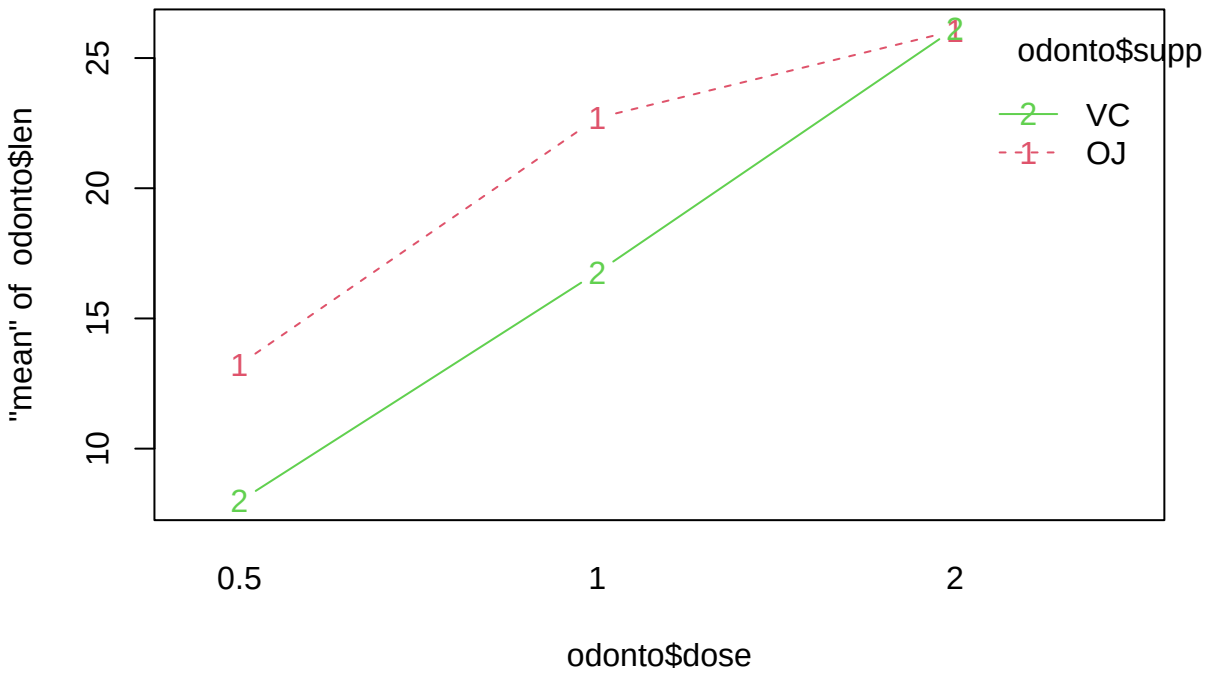
Graficamente
Podemos corrobarar tanto analíticamente como gráficamente que no existe diferencias estadísticamente significativas respecto de las vehiculos para la longitud de crecimiento de los dientes. El p-valor de la muestra ~0.06 con lo que

3. La interacción entre estas variables es significativa?

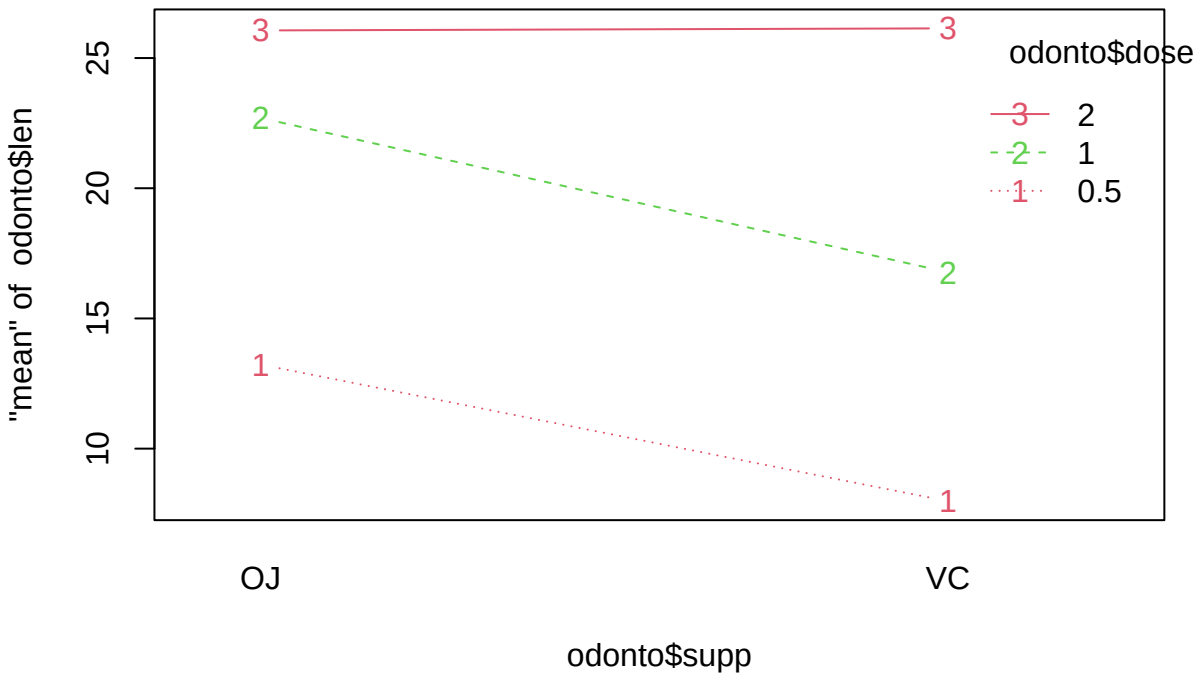
```
aov_odonto_inter <- aov(formula = len ~ supp*dose, data = odonto)
summary(aov_odonto_inter)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## supp         1  205.4   205.4  15.572 0.000231 ***
## dose         2 2426.4  1213.2  92.000  < 2e-16 ***
## supp:dose    2  108.3    54.2   4.107 0.021860 *
## Residuals   54  712.1    13.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
interaction.plot(trace.factor = odonto$supp,
                 x.factor = odonto$dose,
                 response = odonto$len,
                 fun = "mean",
                 legend = TRUE,
                 col = 2:3,
                 type = "b")
```




```
interaction.plot(trace.factor = odonto$dose,
                 x.factor = odonto$supp,
                 response = odonto$len,
                 fun = "mean",
                 legend = TRUE,
                 col = 2:3,
                 type = "b")
```



Existen diferencias estadísticamente significativas entre ambas dosis y tipos de bebidas, y que existe una interacción estadísticamente significativa entre estas dos variables en términos de su efecto sobre la longitud del diente La diferencia en los valores de p para Supp entre los dos modelos puede explicarse por la presencia de una interacción entre Supp y dosis. Cuando se tiene en cuenta esta interacción, queda claro que existe una diferencia estadísticamente significativa entre los tipos de bebidas.

4. ¿Se satisfacen los supuestos del modelo?

```
test_supuestos_aov(aov_odonto_inter)
```

```
##    Prueba   P_Value           H0
## 1 Shapiro 0.6694242 No rechazada
## 2  Levene 0.1483606 No rechazada
##                                              Observaciones
## 1                                 Los residuos son normales.
## 2 Homocedasticidad. La varianza de los residuos es constante.
```

```
shapiro.test( aov_odonto_inter$residuals )
```

Analíticamente

```
##
##  Shapiro-Wilk normality test
##
## data:  aov_odonto_inter$residuals
## W = 0.98499, p-value = 0.6694
```

No se rechaza la hipotesis nula de normalidad.

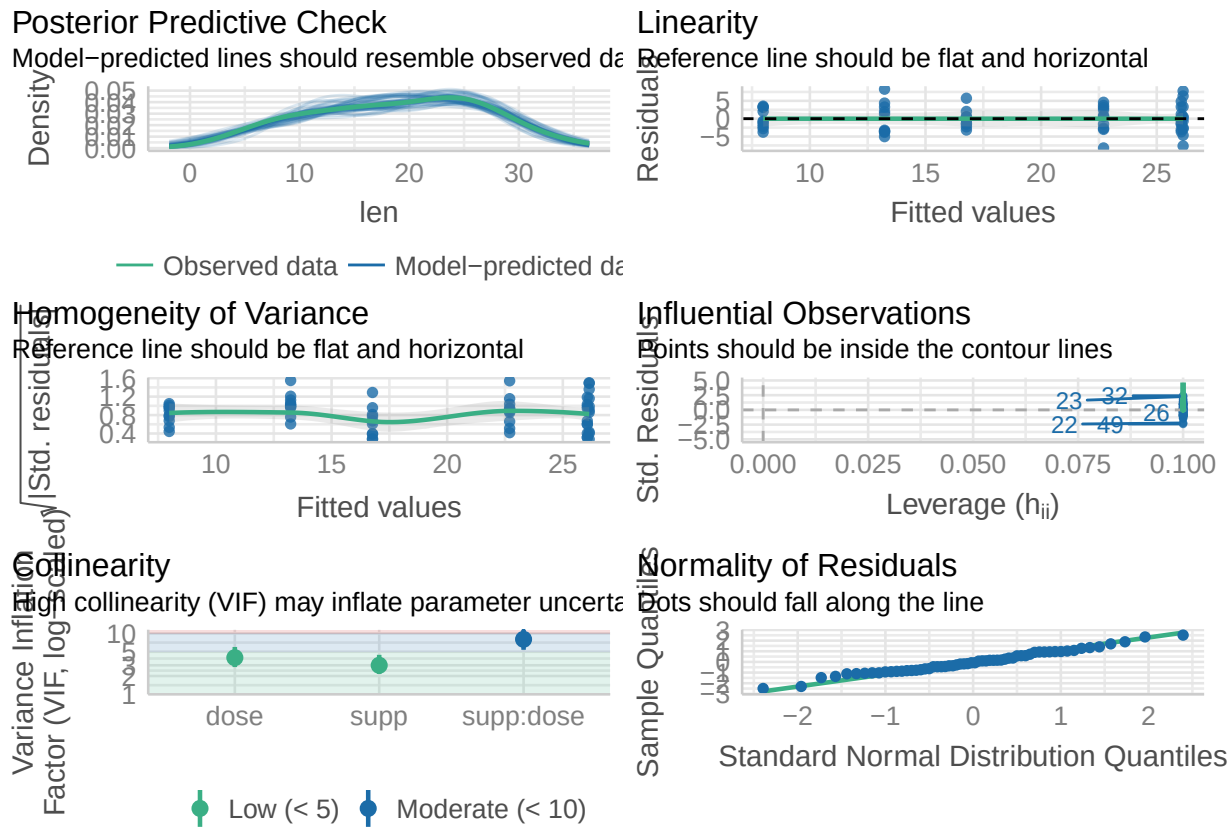
```
leveneTest(aov_odonto_inter)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  5  1.7086 0.1484
##       54
```

No se rechaza la hipotesis nula de homocedasticidad.

Graficamente podemos corroborar lo mencionado en la parte analítica ##### Gráficamente

```
check_model(aov_odonto_inter)
```



5. ¿Puede realizar una recomendación?

Si la dosis es de 2mg no importa el tipo de vehiculo que se utilice, la longitud promedio del largo de los dientes será la misma. Ahora para el caso de dosis de 0.5 como 1mg conviene la aplicación por medio del vehiculo VJ si se quiere tener una longitud promedio mayor de los dientes frontales.

Ejercicio 3

En el archivo morosos.xlsx se encuentran los registros de 10 mil clientes de un banco para los cuales se relevaron las siguientes variables:

- mora: si está en mora con el saldo de su tarjeta de crédito.
- estudiantes: si es estudiante o no.
- balance: el saldo al 31/12 próximo pasado.
- ingreso: ingreso mensual medio del cliente.

Dataset

```
morosos <-read_excel("C:/Austral/mcd-reg-adv/datasets/morosos.xlsx")
```

1. Ajustar un modelo logístico para predecir la probabilidad de incurrir en mora.

```
morosos$mora <- ifelse(morosos$mora == "Yes", 1, 0)

modelo.morosos <- glm(mora ~ estudiante + balance + ingreso, data = morosos, family = "binomial")
summary(modelo.morosos)
```

```
##
## Call:
## glm(formula = mora ~ estudiante + balance + ingreso, family = "binomial",
##     data = morosos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4691  -0.1418  -0.0557  -0.0203   3.7383
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
## estudianteYes -6.468e-01  2.363e-01  -2.738  0.00619 **
## balance        5.737e-03  2.319e-04  24.738  < 2e-16 ***
## ingreso        3.033e-06  8.203e-06   0.370  0.71152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.5  on 9996  degrees of freedom
## AIC: 1579.5
##
## Number of Fisher Scoring iterations: 8
```

Analizando las variables se observa que en el Test de Wald, la variable

```r
table(morosos$mora)
```

```
##
##    0    1
## 9667  333
```

Existe un desbalanceo significativo en la distribucion de los casos de mora.

2. Evaluar la calidad de ajuste del modelo con al menos dos criterios distintos.

```r
indices_validacion <- sample(nrow(morosos), round(0.2 * nrow(morosos)))
datos_entrenamiento <- morosos[-indices_validacion, ]
datos_validacion <- morosos[indices_validacion, ]

modelo.morosos2 <- glm(mora ~ estudiante + balance + ingreso, data = datos_entrenamiento, family = "bin
```

```r
probs_validacion <- predict(modelo.morosos2, newdata = datos_validacion, type = "response")

corte <- 0.5

predicciones_validacion <- ifelse(probs_validacion >= corte, 1, 0)

matriz_confusion <- table(observado = datos_validacion$mora, predicho = predicciones_validacion)

print(matriz_confusion)
```

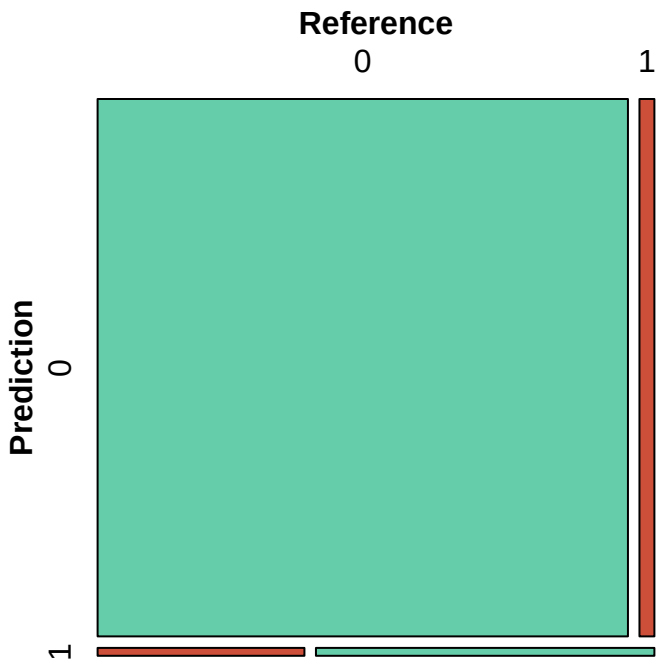
```
##          predicho
## observado    0    1
##         0 1917   11
##         1   54   18
```

```r
matriz_confusion <- confusionMatrix(factor(predicciones_validacion), factor(datos_validacion$mora))
```

```r
print(matriz_confusion$table)
```

```
##           Reference
## Prediction    0    1
##          0 1917   54
##          1   11   18
```

```r
mosaic(matriz_confusion$table,
       shade = TRUE,
       colorize = TRUE,
       xlab = "Predicciones",
       ylab = "Observado",
       main = "Matriz de Confusión",
       gp = gpar(fill = matrix(c("aquamarine3", "tomato3", "tomato3", "aquamarine3"),2,2)))
```




```
sensitivity(matriz_confusion$table)
```

```
## [1] 0.9942946
```

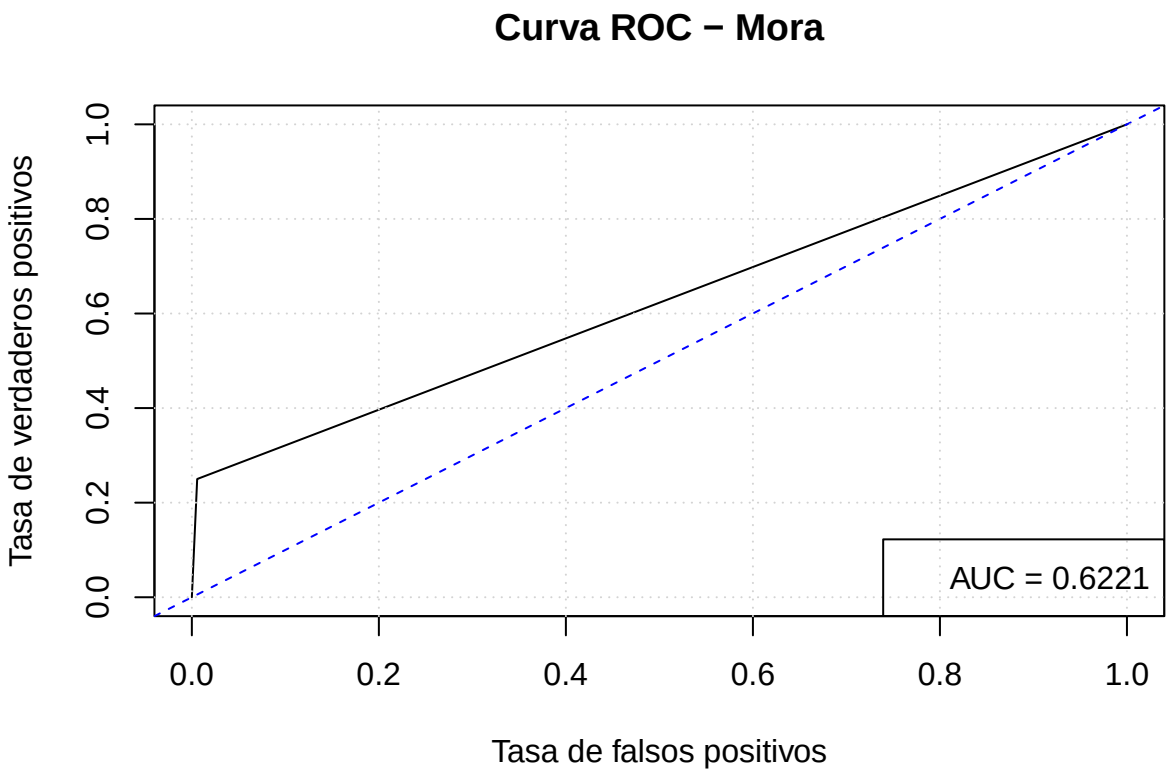
```
specificity(matriz_confusion$table)
```

```
## [1] 0.25
```

Se observa la existencia de una baja especificidad: el modelo predijo el 34% de los casos POSITIVOS. Sin embargo, existe un alta sensibilidad: el modelo predijo el 99% de los casos NEGATIVOS.

Teniendo en cuenta estos indicadores, es posible indicar que si bien el modelo es eficiente en encontrar casos NETIVOS, posee un desempeño regular en la predicción de casos POSITIVOS, es decir, en clientes con mora. Esto puede deberse al desbalanceo de clases que se observa en la variable a predecir.

```
predic <- prediction(predicciones_validacion, datos_validacion$mora)
perf <-  performance(predic, "tpr","fpr", alpha = seq(0, 1, by = 0.01))

plot(perf,
     main = "Curva ROC - Mora",
     xlab="Tasa de falsos positivos",
     ylab="Tasa de verdaderos positivos")
abline(a=0,b=1,col="blue",lty=2)
grid()
auc <- as.numeric(performance(predic,"auc")@y.values)
legend("bottomright",legend=paste(" AUC =",round(auc,4)))
```




```
hoslem.test(datos_entrenamiento$mora, fitted(modelo.morosos2))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  datos_entrenamiento$mora, fitted(modelo.morosos2)
## X-squared = 5.5299, df = 8, p-value = 0.6997
```

Si bien el test de Hosmer y Lemeshow no rechaza la hipotesis nula de un buen ajuste por parte del modelo logistico. Como vimos en el apartado anterior, solo lo hace para los casos negativos o sin probabilidad de mora.

3. Interpretar los coeficientes del modelo elegido.

```
exp(coef(modelo.morosos))
```

```
##   (Intercept) estudianteYes       balance       ingreso
##  1.903854e-05  5.237317e-01  1.005753e+00  1.000003e+00
```

estudianteYes: El coeficiente para la variable “estudianteYes” es -0.6468. Al aplicar la función exponencial, el odds ratio es aproximadamente 0.5247. Esto significa que, manteniendo todas las demás variables constantes, los estudiantes tienen aproximadamente un 47.53% menos de odds de estar en mora en comparación con los no estudiantes.

balance: El coeficiente para "balance" es 0.005737. Al aplicar la función exponencial, el odds ratio es aproximadamente 1.00576. Esto implica que, manteniendo todas las demás variables constantes, por cada unidad adicional de saldo en la tarjeta de crédito, las odds de estar en mora aumentan en aproximadamente un 0.576%.

ingreso: El coeficiente para "ingreso" es muy pequeño, 3.033e-06 (aproximadamente 0.000003033). Al aplicar la función exponencial, el odds ratio es cercano a 1, lo que significa que no hay un cambio significativo en las odds de estar en mora asociado con el ingreso mensual medio del cliente. Además, el valor p (Pr(>|z|)) es 0.71152, lo que indica que el ingreso no es estadísticamente significativo para predecir la mora.

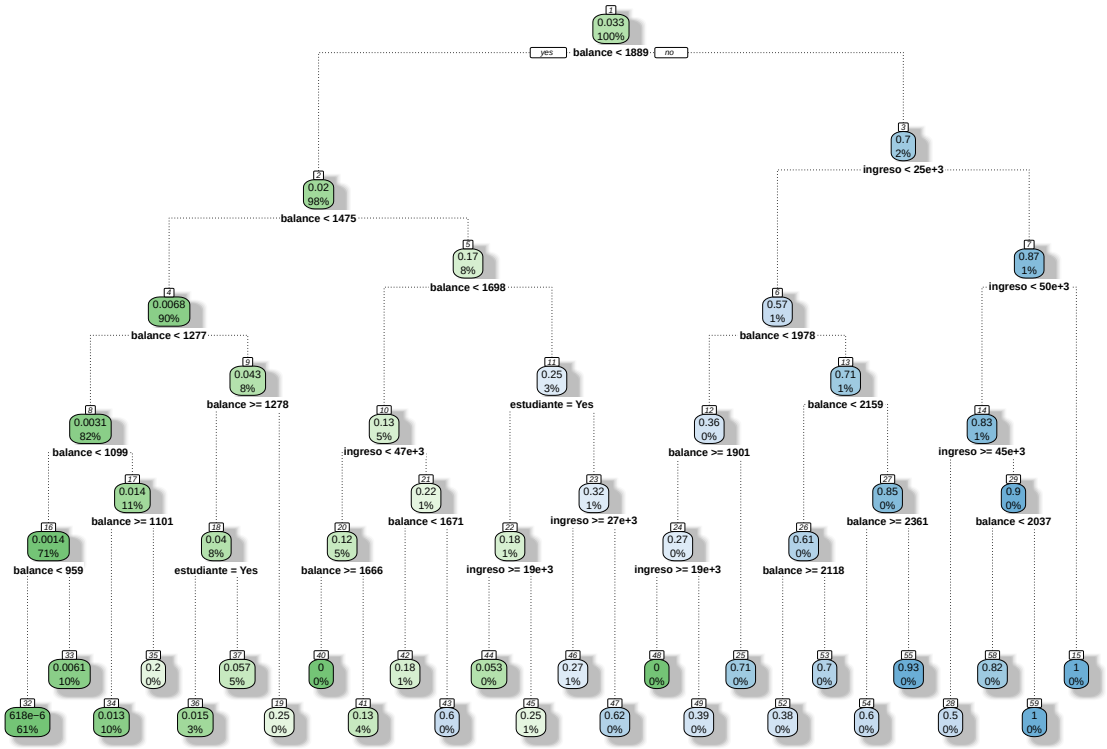
En resumen, en este modelo logístico, las variables "estudiante" y "saldo" parecen tener un impacto significativo en la predicción de si un cliente estará en mora con su tarjeta de crédito. Mientras que, el ingreso no parece ser un predictor relevante para la mora.

4. Evaluar la calidad de clasificación y compararlo con otro método de clasificación.

```
library(rpart)

arbol <- rpart( mora ~ estudiante + balance + ingreso,
            data = datos_entrenamiento,
            xval=      0,
            cp=        0,   # esto significa no limitar la complejidad de los splits
            minsplit=  5,   # minima cantidad de registros para que se haga el split
            minbucket= 5,   # tamaño minimo de una hoja
            maxdepth=  5 )  # profundidad maxima del arbol
```

```
predicted_classes <- factor( predict(arbol, datos_validacion, type = "vector") >= 0.5 )
```

```
library(rpart.plot)

rpart.plot(arbol,
                    # show fitted class, probs, percentages
           box.palette = "GnBu", # color scheme
           branch.lty = 3,       # dotted branch lines
           shadow.col = "grey",  # shadows under the node boxes
           nn = TRUE)
```




```
pred_arbol <- predict(arbol, datos_validacion, type = "vector")
pred_arbol_clases <- ifelse(pred_arbol > 0.5, 1, 0)
```

```
confusion_matrix_arbol <- confusionMatrix(factor(pred_arbol_clases), factor(datos_validacion$mora))
print(confusion_matrix_arbol)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1916   54
##          1   12   18
##
##                Accuracy : 0.967
##                  95% CI : (0.9582, 0.9744)
##     No Information Rate : 0.964
##     P-Value [Acc > NIR] : 0.258
##
##                   Kappa : 0.3389
##
##  Mcnemar's Test P-Value : 4.494e-07
##
##             Sensitivity : 0.9938
##             Specificity : 0.2500
##          Pos Pred Value : 0.9726
```

```
##          Neg Pred Value : 0.6000
##              Prevalence : 0.9640
##          Detection Rate : 0.9580
##    Detection Prevalence : 0.9850
##       Balanced Accuracy : 0.6219
##
##        'Positive' Class : 0
##
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
roc_arbol <- roc(datos_validacion$mora, pred_arbol)
```

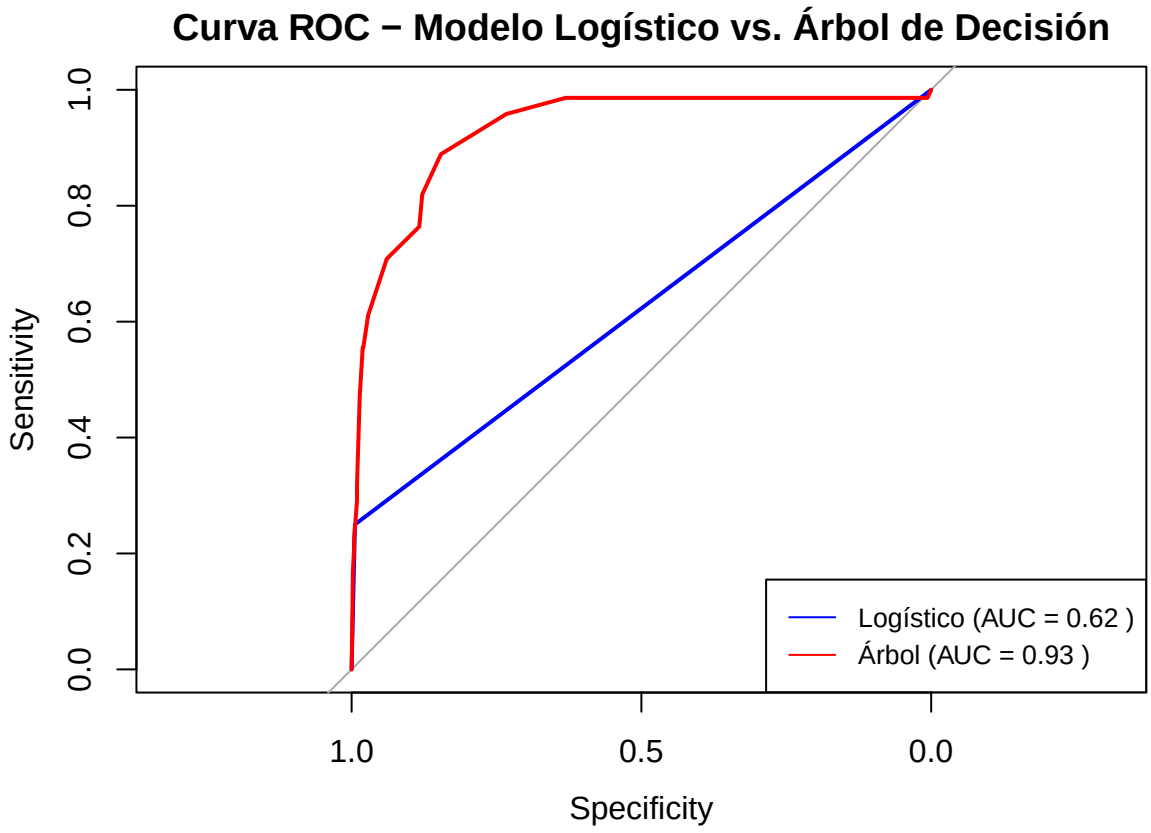
```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```
auc_arbol <- auc(roc_arbol)

roc_logistico <- roc(datos_validacion$mora, predicciones_validacion)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
auc_logistico <- auc(roc_logistico)
```

```
plot(roc_logistico, col = "blue", main = "Curva ROC - Modelo Logístico vs. Árbol de Decisión")
lines(roc_arbol, col = "red")
legend("bottomright", legend = c(paste("Logístico (AUC =", round(auc_logistico, 2), ")"),
                                 paste("Árbol (AUC =", round(auc_arbol, 2), ")")),
       col = c("blue", "red"), lty = 1, cex = 0.8)
```




```
data_roc_logistico <- data.frame(Specificity = 1 - roc_logistico$specificities,
                                 Sensitivity = roc_logistico$sensitivities, Modelo = "Modelo Logístico")
data_roc_arbol <- data.frame(Specificity = 1 - roc_arbol$specificities,
                             Sensitivity = roc_arbol$sensitivities, Modelo = "Árbol de Decisión")

data_roc <- rbind(data_roc_logistico, data_roc_arbol)

#Punto óptimo en la curva ROC para cada modelo (máximo valor de la suma de sensibilidad y especificidad
punto_optimo_logistico <- roc_logistico$thresholds[which.max(roc_logistico$sensitivities + roc_logistico
punto_optimo_arbol <- roc_arbol$thresholds[which.max(roc_arbol$sensitivities + roc_arbol$specificities)]
```

```
p <- ggplot(data_roc, aes(x = Specificity, y = Sensitivity)) +
  geom_line(aes(color=Modelo), size=1) +
  scale_color_manual(values=c("#ff7f0e","#1f77b4"))+
  geom_point(data = data.frame(Specificity = 1 - roc_logistico$specificities[roc_logistico$thresholds ==
                               Sensitivity = roc_logistico$sensitivities[roc_logistico$thresholds == pun
             size = 3, shape = 16, color = "#1f77b4") +
  geom_point(data = data.frame(Specificity = 1 - roc_arbol$specificities[roc_arbol$thresholds == punto_
                               Sensitivity = roc_arbol$sensitivities[roc_arbol$thresholds == punto_optim
             size = 3, shape = 16, color = "#ff7f0e") +
  geom_text(aes(label = paste("AUC =", round(auc_logistico, 2))), x = 0.75, y = 0.25, color = "#1f77b4"
  geom_text(aes(label = paste("AUC =", round(auc_arbol, 2))), x = 0.75, y = 0.2, color = "#ff7f0e", hjus
  annotate("text", x = 0.7, y = 0.15, label = paste("Punto Óptimo =", round(punto_optimo_logistico, 2))
           color = "#1f77b4", hjust = 0, vjust = 0) +
  annotate("text", x = 0.7, y = 0.1, label = paste("Punto Óptimo =", round(punto_optimo_arbol, 2)),
```

```
           color = "#ff7f0e", hjust = 0, vjust = 0) +
  labs(x = "1 - Specificity", y = "Sensitivity", title = "Curva ROC - Modelo Logístico vs. Árbol de Deci
       color = "Modelo") +
  theme_minimal() +
  theme(legend.position = "right")
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
# Agregar línea diagonal en 0.5
p <- p + geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "gray")

# Ajustar límites del eje x e y para que la línea diagonal se extienda solo desde (0,0) hasta (1,1)
p <- p + xlim(0, 1) + ylim(0, 1)

print(p)
```