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, leverage\_str),  
 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.

#### Analíticamente

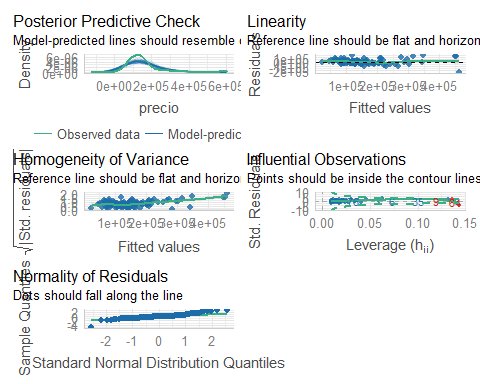
test\_supuestos(modelo.casas)

## Prueba P\_Value H0  
## 1 Shapiro 8.728976e-05 Rechazada  
## 2 Breusch-Pagan 1.013482e-09 Rechazada  
## 3 Durbin-Watson 4.000000e-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.

#### Gráficamente

check\_model( modelo.casas )

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



#### Analisis de outliers y puntos influyentes

outliers\_influential\_test(modelo.casas)

##   
## 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

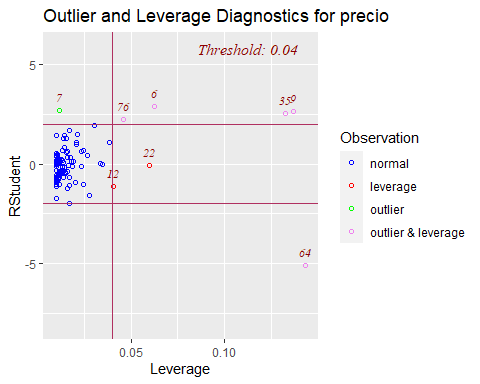
## The following object is masked from 'package:MASS':  
##   
## select

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

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

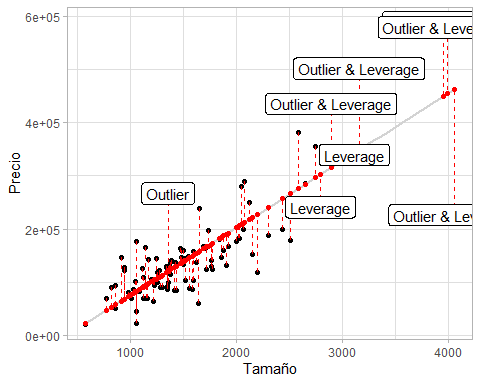
## Test\_Criteria Influential\_Points  
## 1 Bonferroni 6, 7, 9, 35, 64, 76  
## 2 Cook 1 6, 9, 35, 64, 66, 76  
## 3 Cook 2 6, 9, 35, 64, 66, 76  
## 4 DFBetas 2, 6, 7, 9, 35, 64, 76, 106, 109, 112, 121, 135, 164, 166, 176  
## 5 DFFits 6, 7, 9, 35, 66, 76  
## 6 DFFits2   
## 7 Leverage 6, 9, 12, 22, 35, 64, 76

ols\_plot\_resid\_lev(modelo.casas, print\_plot = TRUE)



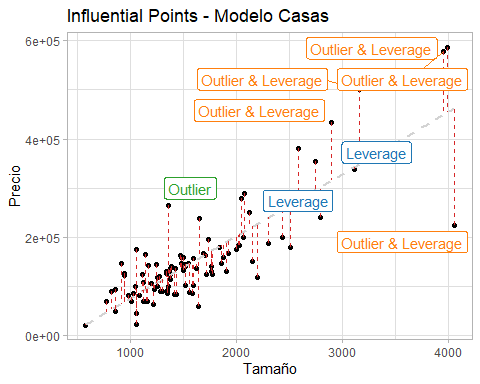
#Opcion 1 gráfico  
datos <- casas  
datos$predicciones <- predict(modelo.casas)  
  
leverage <- hatvalues(modelo.casas)  
std\_residuals <- rstandard(modelo.casas)  
  
datos$label <- NA  
  
identify\_label <- "Leverage"  
datos$label[leverage > 2 \* mean(leverage)] <- identify\_label  
  
identify\_label <- "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'



#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" = "#ff7f0e")) +  
 theme\_light()

## `geom\_smooth()` using formula = 'y ~ x'



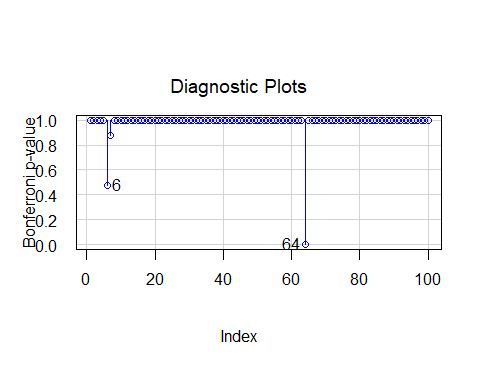
##### OUTLIERS

# Test de Bonferroni  
outlierTest( modelo.casas )

## 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")



##### PUNTOS INFLUYENTES

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

## 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

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

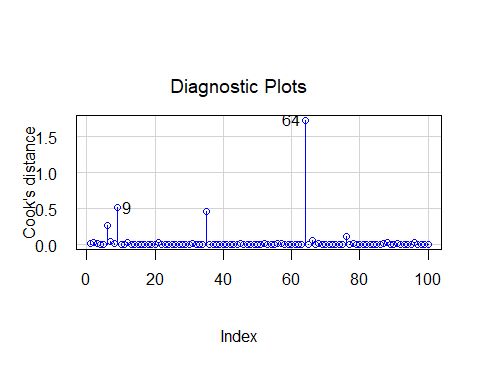
Usando un criterio mas exigente:

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

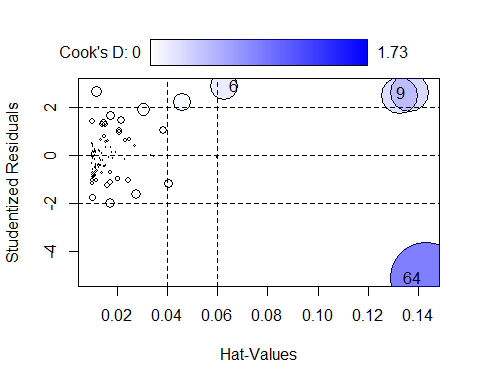
## [1] 64

Visualizandolo gráficamente

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



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

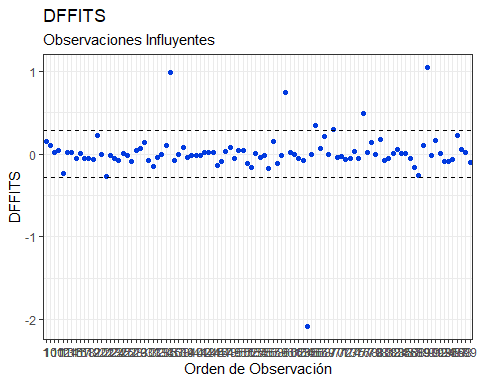
Usando el criterio de DFFITS, y el punto de corte 2 \* sqrt( p/n ) aparecen otras observaciones: 6,7,9,35,66,76

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)

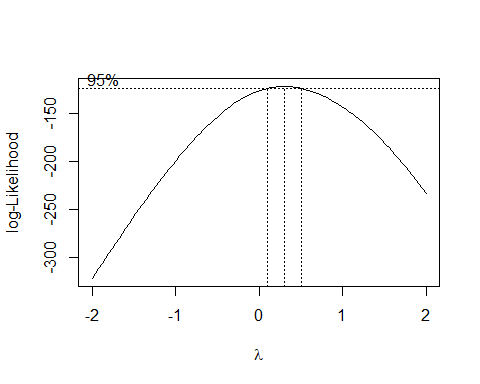
lev <- hatvalues(modelo.casas)  
unique(which(lev>2\*p/n))

## [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.

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



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

## [1] 0.3030303

#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 0.01200000 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.

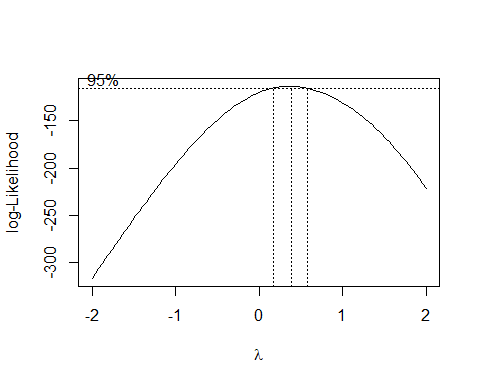
### 4. Eliminar la observación 64 y ajustar nuevamente el segundo modelo evaluando su validez.

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))  
  
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: 153919.31"

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

## [1] "RLM RSME: 49104.4"

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

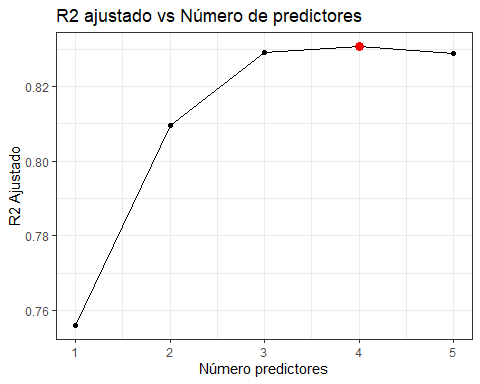
### 6. Utilizar un método de selección de variables para proponer un modelo multivariado. Analizar el cumplimiento de los supuestos.

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 ) "\*" "\*" "\*" "\*" "\*"

adjr2\_values <- summary(modelo.casas.comb)$adjr2  
best\_model\_adjr2 <- which.max(summary(modelo.casas.comb)$adjr2)

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 6.000000e-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.

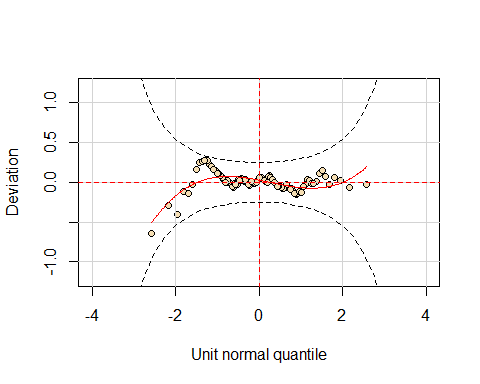
Tal como se observa en el punto anterior, y quedando en evidencia que no se cumple el supuesto de homocedasticidad, resultaría útil modelar usando un modelo GAMLSS.

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

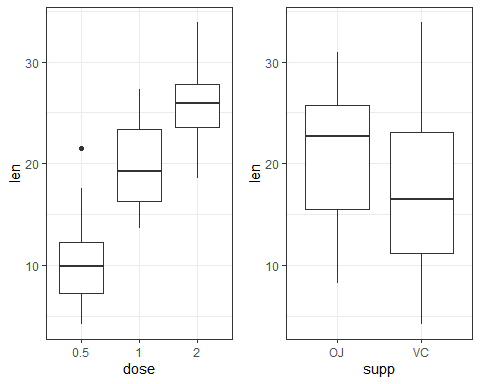
Se desea saber si la dosis de ácido ascórbico y el tipo de bebida en la cual se lo administró a ciertos animales de laboratorio logró mayor desarrollo de los dientes en los mismo. Se utilizaron 60 replicaciones del experimento y se tienen grupos balanceados. La variable respuesta de interés es la longitud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

### Dataset

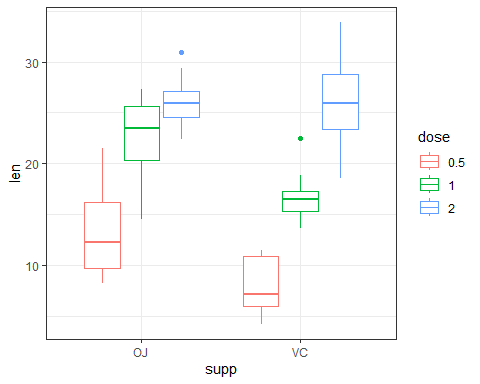
setwd("C:/Austral/mcd-reg-adv/datasets")  
odonto <- read.csv("odonto.csv")

### 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



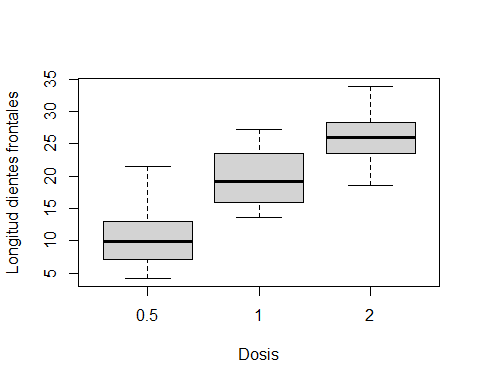
#### Analíticamente

aov\_odonto <- aov(formula = len ~ dose, data = odonto)  
summary(aov\_odonto)

## 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

#### Gráficamente

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



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?

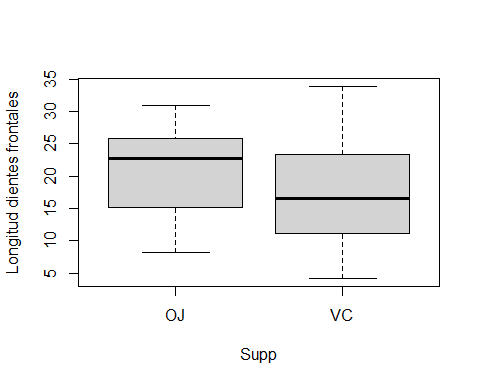
#### Analíticamente

aov\_odonto\_supp <- aov(formula = len ~ supp, data = odonto)  
summary(aov\_odonto\_supp)

## 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

#### Graficamente

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

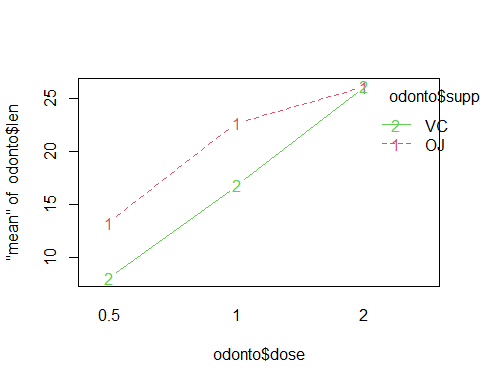
 Podemos corrobarar tanto analíticamente como gráficamente que no existe diferencias estadísticamente significativas respecto de las vehiculos para la longitud de crecimiento de los dientes. El p-valor de la muestra ~0.06 con lo que

### 3. La interacción entre estas variables es significativa?

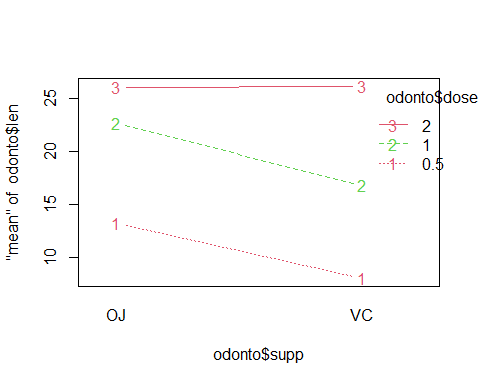
aov\_odonto\_inter <- aov(formula = len ~ supp\*dose, data = odonto)  
summary(aov\_odonto\_inter)

## 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.

#### Analíticamente

shapiro.test( aov\_odonto\_inter$residuals )

##   
## Shapiro-Wilk normality test  
##   
## data: aov\_odonto\_inter$residuals  
## W = 0.98499, p-value = 0.6694

No se rechaza la hipotesis nula de normalidad.

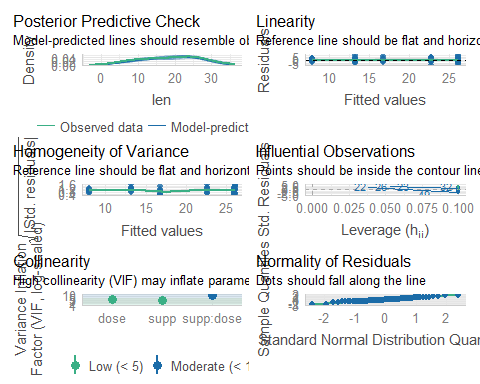
leveneTest(aov\_odonto\_inter)

## 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

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.

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 = "binomial")  
  
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)

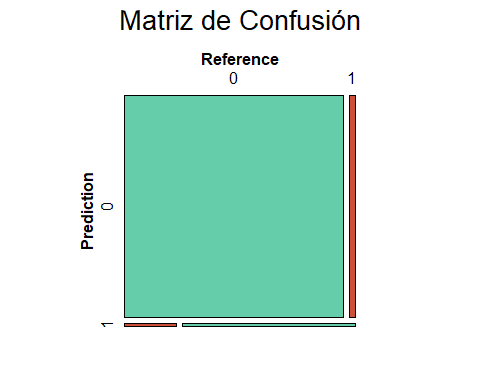
## predicho  
## observado 0 1  
## 0 1926 6  
## 1 48 20

matriz\_confusion <- confusionMatrix(factor(predicciones\_validacion), factor(datos\_validacion$mora))

print(matriz\_confusion$table)

## Reference  
## Prediction 0 1  
## 0 1926 48  
## 1 6 20

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] 0.9968944

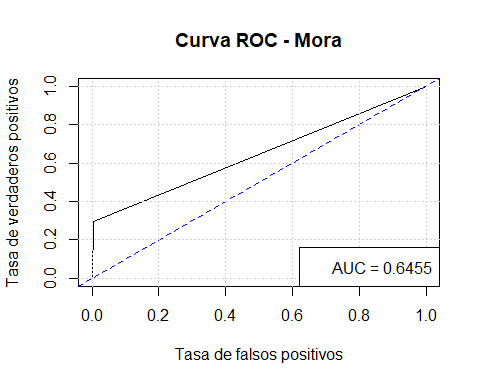
specificity(matriz\_confusion$table)

## [1] 0.2941176

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 = 3.9527, df = 8, p-value = 0.8614

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.

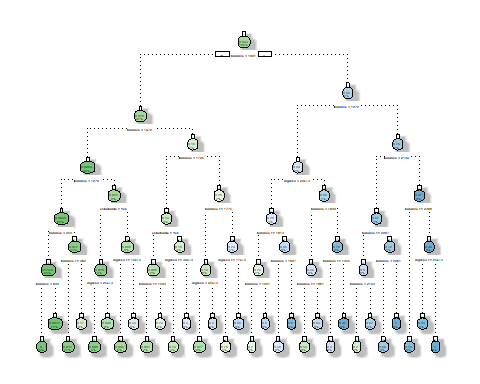
En resumen, en este modelo logístico, las variables “estudiante” y “saldo” parecen tener un impacto significativo en la predicción de si un cliente estará en mora con su tarjeta de crédito. Mientras que, el ingreso no parece ser un predictor relevante para la mora.

### 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 1919 43  
## 1 13 25  
##   
## Accuracy : 0.972   
## 95% CI : (0.9638, 0.9788)  
## No Information Rate : 0.966   
## P-Value [Acc > NIR] : 0.0748361   
##   
## Kappa : 0.4585   
##   
## Mcnemar's Test P-Value : 0.0001065   
##   
## Sensitivity : 0.9933   
## Specificity : 0.3676   
## Pos Pred Value : 0.9781   
## Neg Pred Value : 0.6579   
## Prevalence : 0.9660   
## Detection Rate : 0.9595   
## Detection Prevalence : 0.9810   
## Balanced Accuracy : 0.6805   
##   
## '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

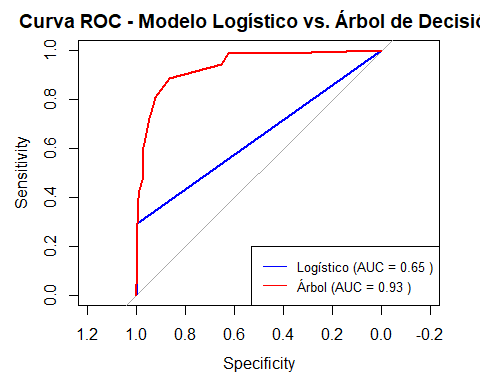
## Setting direction: controls < cases

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$specificities)]  
punto\_optimo\_arbol <- roc\_arbol$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 == punto\_optimo\_logistico],  
 Sensitivity = roc\_logistico$sensitivities[roc\_logistico$thresholds == punto\_optimo\_logistico], Modelo = "Modelo Logístico"),  
 size = 3, shape = 16, color = "#1f77b4") +  
 geom\_point(data = data.frame(Specificity = 1 - roc\_arbol$specificities[roc\_arbol$thresholds == punto\_optimo\_arbol],  
 Sensitivity = roc\_arbol$sensitivities[roc\_arbol$thresholds == punto\_optimo\_arbol], Modelo = "Árbol de Decisión"),  
 size = 3, shape = 16, color = "#ff7f0e") +  
 geom\_text(aes(label = paste("AUC =", round(auc\_logistico, 2))), x = 0.75, y = 0.25, color = "#1f77b4", hjust = 0, vjust = 0) +  
 geom\_text(aes(label = paste("AUC =", round(auc\_arbol, 2))), x = 0.75, y = 0.2, color = "#ff7f0e", hjust = 0, vjust = 0) +  
 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 Decisión",  
 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)

