Activitat 3: Models predictius

Marc Cervera Rosell

2024-05-24

```
setRepositories(ind=2) # Per descarregar paquets de CRAN
```

1. Regressió lineal

1.1 Preparació de les dades

```
tryCatch({
  data <- read.csv("Casas.csv", header = TRUE)
  print("Fitxer llegit correctament")
}, error = function(e){
  cat("ERROR en el moment de llegir el fitxer:",conditionMessage(e), "\n")
})
```

```
## [1] "Fitxer llegit correctament"
```

```
any(is.na(data))
```

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

El fitxer no conté valors NA Canvi de peus quadrats a metres quadrats i de dòlars a euros:

```
for (i in seq_along(data$sqft_living)) {
  data$sqft_living[i] <- data$sqft_living[i] * 0.0929
  data$sqft_lot[i] <- data$sqft_lot[i] * 0.0929
  data$sqft_living15[i] <- data$sqft_living15[i] * 0.0929
  data$sqft_lot15[i] <- data$sqft_lot15[i] * 0.0929
  data$sqft_basement[i] <- data$sqft_basement[i] * 0.0929
  data$price[i] <- data$price[i] * 0.93
}
```

La funció *seq_along()* ens permet iterar sobre la columna indicada i, atès que, totes les columnes tenen la mateixa longitud no hi ha problema a posar com a argument una columna o una altra.

IMPORTANT! ELS NOMS DE LES COLUMNES NO CANVIEN, PERÒ SÍ ELS SEUS VALORS, PER TANT, ENCARA QUE LES COLUMNES ES DIGUIN "sqft_[...]", ELS VALORS NO SON PEUS SINÓ METRES! EN EL CAS DE LA COLUMNA DE PREUS, ELS VALORS CORRESPONEN A EUROS I NO A DÒLARS!

```
columns <- names(data)
type <- sapply(data, class)
for (i in seq_along(columns)) {
  cat("La columna", columns[i], "és de tipus", type[i], "\n")
}
```

```
## La columna date és de tipus character
```

```
## La columna price és de tipus numeric
## La columna bedrooms és de tipus integer
## La columna bathrooms és de tipus numeric
## La columna sqft_living és de tipus numeric
## La columna sqft_lot és de tipus numeric
## La columna floors és de tipus numeric
## La columna waterfront és de tipus integer
## La columna view és de tipus integer
## La columna condition és de tipus integer
## La columna sqft_basement és de tipus numeric
## La columna yr_built és de tipus integer
## La columna yr_renovated és de tipus integer
## La columna zipcode és de tipus integer
## La columna sqft_living15 és de tipus numeric
## La columna sqft_lot15 és de tipus numeric
```

Després d'analitzar els tipus de les variables del fitxer es determina realitzar un canvi de tipus de les següents variables:

- *bathrooms* -> Actualment és de tipus numèric, i no es canviarà atès que si s'observa **l'apartat de discusions del dataset**, es podrà veure el significat dels decimals. Per veure la web del dataset, clicar sobre el text en negreta.
- *floors* -> Actualment és de tipus numèric. En aquest cas, no s'ha trobat cap explicació per als decimals d'aquesta variable, per tant es decideix fer el canvi a *integer*.

```
data <- transform(data,
                  floors = as.integer(floors))
```

```
columns <- names(data)
type <- sapply(data, class)
for (i in seq_along(columns)) {
  cat("La columna", columns[i], "és de tipus", type[i], "\n")
}
```

```
## La columna date és de tipus character
## La columna price és de tipus numeric
## La columna bedrooms és de tipus integer
## La columna bathrooms és de tipus numeric
## La columna sqft_living és de tipus numeric
## La columna sqft_lot és de tipus numeric
## La columna floors és de tipus integer
## La columna waterfront és de tipus integer
## La columna view és de tipus integer
## La columna condition és de tipus integer
## La columna sqft_basement és de tipus numeric
## La columna yr_built és de tipus integer
## La columna yr_renovated és de tipus integer
## La columna zipcode és de tipus integer
## La columna sqft_living15 és de tipus numeric
## La columna sqft_lot15 és de tipus numeric
```

Com s'observa, després d'aplicar la funció *transform()* s'han modificat els tipus.

1.2 Estudi de correlació lineal

Considerant que s'han d'excloure dues de les variables del fitxer en el moment del càlcul de la correlació lineal, cal seleccionar, primer, aquelles columnes que sí que s'usaran en el càlcul.

```
subset_estudi_correlacio <- data[, c("price", "bedrooms", "bathrooms", "sqft_living",
                                     "sqft_lot", "floors", "waterfront", "view",
                                     "condition", "sqft_basement", "yr_built",
                                     "yr_renovated", "sqft_living15", "sqft_lot15")]
```

```
matriu_correlacio <- cor(subset_estudi_correlacio)
indexs <- which(matriu_correlacio > 0.2, arr.ind = TRUE)
indexs_ordenats <- indexs[order(matriu_correlacio[indexs], decreasing = TRUE), ]
files <- rownames(matriu_correlacio)[indexs_ordenats[,1]]
columnes <- colnames(matriu_correlacio)[indexs_ordenats[,2]]
matriu_noms <- cbind(files, columnes, matriu_correlacio[indexs_ordenats])
matriu_final <- matrix(matriu_noms, ncol = 3, byrow = FALSE)
colnames(matriu_final) <- c("Nom variable", "Nom variable", "Coef. Correlació")
print(matriu_final)
```

```
##       Nom variable    Nom variable    Coef. Correlació
##  [1,] "price"         "price"         "1"
##  [2,] "bedrooms"      "bedrooms"      "1"
##  [3,] "bathrooms"     "bathrooms"     "1"
##  [4,] "sqft_living"   "sqft_living"   "1"
##  [5,] "sqft_lot"      "sqft_lot"      "1"
##  [6,] "floors"        "floors"        "1"
##  [7,] "waterfront"    "waterfront"    "1"
##  [8,] "view"          "view"          "1"
##  [9,] "condition"     "condition"     "1"
## [10,] "sqft_basement" "sqft_basement" "1"
## [11,] "yr_built"      "yr_built"      "1"
## [12,] "yr_renovated"  "yr_renovated"  "1"
## [13,] "sqft_living15" "sqft_living15" "1"
## [14,] "sqft_lot15"    "sqft_lot15"    "1"
## [15,] "sqft_living15" "sqft_living"   "0.756420259017221"
## [16,] "sqft_living"   "sqft_living15" "0.756420259017221"
## [17,] "sqft_living"   "bathrooms"     "0.754665278967373"
## [18,] "bathrooms"     "sqft_living"   "0.754665278967373"
## [19,] "sqft_lot15"    "sqft_lot"      "0.718556752433035"
## [20,] "sqft_lot"      "sqft_lot15"    "0.718556752433035"
## [21,] "sqft_living"   "price"         "0.702043721232527"
## [22,] "price"         "sqft_living"   "0.702043721232527"
## [23,] "sqft_living15" "price"         "0.585374006317152"
## [24,] "price"         "sqft_living15" "0.585374006317152"
## [25,] "yr_built"      "floors"        "0.578619375159292"
## [26,] "floors"        "yr_built"      "0.578619375159292"
## [27,] "sqft_living"   "bedrooms"      "0.576670692502244"
## [28,] "bedrooms"      "sqft_living"   "0.576670692502244"
## [29,] "sqft_living15" "bathrooms"     "0.568634289578224"
## [30,] "bathrooms"     "sqft_living15" "0.568634289578224"
## [31,] "bathrooms"     "price"         "0.525134072745601"
## [32,] "price"         "bathrooms"     "0.525134072745601"
## [33,] "floors"        "bathrooms"     "0.519018991536239"
## [34,] "bathrooms"     "floors"        "0.519018991536239"
```

```
## [35,] "bathrooms"     "bedrooms"      "0.51588363761583"
## [36,] "bedrooms"      "bathrooms"     "0.51588363761583"
## [37,] "yr_built"      "bathrooms"     "0.506019438285253"
## [38,] "bathrooms"     "yr_built"      "0.506019438285253"
## [39,] "sqft_basement" "sqft_living"   "0.435042973669821"
## [40,] "sqft_living"   "sqft_basement" "0.435042973669821"
## [41,] "view"          "waterfront"    "0.401857350697571"
## [42,] "waterfront"    "view"          "0.401857350697571"
## [43,] "view"          "price"         "0.397346474378939"
## [44,] "price"         "view"          "0.397346474378939"
## [45,] "sqft_living15" "bedrooms"      "0.391637523968824"
## [46,] "bedrooms"      "sqft_living15" "0.391637523968824"
## [47,] "floors"        "sqft_living"   "0.35332060339984"
## [48,] "sqft_living"   "floors"        "0.35332060339984"
## [49,] "sqft_living15" "yr_built"      "0.326228899595712"
## [50,] "yr_built"      "sqft_living15" "0.326228899595712"
## [51,] "sqft_basement" "price"         "0.32383735813766"
## [52,] "price"         "sqft_basement" "0.32383735813766"
## [53,] "yr_built"      "sqft_living"   "0.318048768996441"
## [54,] "sqft_living"   "yr_built"      "0.318048768996441"
## [55,] "bedrooms"      "price"         "0.308338368688097"
## [56,] "price"         "bedrooms"      "0.308338368688097"
## [57,] "sqft_basement" "bedrooms"      "0.303093375320663"
## [58,] "bedrooms"      "sqft_basement" "0.303093375320663"
## [59,] "sqft_living15" "floors"        "0.296560578164614"
## [60,] "floors"        "sqft_living15" "0.296560578164614"
## [61,] "view"          "sqft_living"   "0.284611186216901"
## [62,] "sqft_living"   "view"          "0.284611186216901"
## [63,] "sqft_basement" "bathrooms"     "0.283770034004669"
## [64,] "bathrooms"     "sqft_basement" "0.283770034004669"
## [65,] "sqft_living15" "view"          "0.280439081995455"
## [66,] "view"          "sqft_living15" "0.280439081995455"
## [67,] "sqft_basement" "view"          "0.276946578767584"
## [68,] "view"          "sqft_basement" "0.276946578767584"
## [69,] "waterfront"    "price"         "0.266330510522256"
## [70,] "price"         "waterfront"    "0.266330510522256"
## [71,] "floors"        "price"         "0.237207363532409"
## [72,] "price"         "floors"        "0.237207363532409"
## [73,] "sqft_living15" "sqft_basement" "0.200354983394243"
## [74,] "sqft_basement" "sqft_living15" "0.200354983394243"
```

Tenint en compte que solament s'han mostrat aquells coeficients de correlació lineal majors a 0.2, es pot assegurar que la correlació lineal de les variables és positiva, és a dir, quan una de les dues variables augmenta el seu valor, la segona variable també augmenta el seu valor de manera proporcional.

En aquest cas d'estudi, el llindar s'ha establert en 0.2, per tant, aquelles parelles de variables amb un coeficient de correlació lineal proper a 0.2 tindran una correlació dèbil i aquelles parelles amb un coeficient de correlació lineal proper a 1 (o 1 en el cas del càlcul de la correlació lineal amb elles mateixes) tindran una forta correlació.

1.3 Generació dels conjunts d'entrenament i de test

```
set.seed(123)
indexs_training <- sample(nrow(subset_estudi_correlacio), 0.8 *
                            nrow(subset_estudi_correlacio))
```

```
set_training <- subset_estudi_correlacio[indexs_training, ]
set_test <- subset_estudi_correlacio[-indexs_training, ]
```

1.4 Estimació del model de regressió lineal

L'ajust d'un model de regressió lineal utilitzant el mètode de mínims quadrats ordinaris es du a terme, popularment, amb la funció *lm()*.

```
model <- lm(price ~ ., data = set_training)
```

La variable *price*, es la variable anomenada "de resposta" atès que és la variable que està a l'esquerra de la titlla (*virgulilla* en castellà).

1.4.1

```
summary(model)
```

```
##
## Call:
## lm(formula = price ~ ., data = set_training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1287434  -114103   -11637    90398  3792888
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.555e+06  1.559e+05  35.631  < 2e-16 ***
## bedrooms      -4.826e+04  2.222e+03 -21.718  < 2e-16 ***
## bathrooms      6.084e+04  3.847e+03  15.815  < 2e-16 ***
## sqft_living    2.362e+03  4.280e+01  55.189  < 2e-16 ***
## sqft_lot      -1.733e-01  6.281e-01  -0.276  0.78262
## floors         4.766e+04  4.271e+03  11.159  < 2e-16 ***
## waterfront     4.928e+05  2.021e+04  24.389  < 2e-16 ***
## view           4.740e+04  2.480e+03  19.113  < 2e-16 ***
## condition      2.162e+04  2.755e+03   7.846 4.55e-15 ***
## sqft_basement -2.262e+02  5.287e+01  -4.277 1.90e-05 ***
## yr_built      -2.926e+03  7.952e+01 -36.791  < 2e-16 ***
## yr_renovated   1.161e+01  4.332e+00   2.680  0.00736 **
## sqft_living15  7.985e+02  4.094e+01  19.504  < 2e-16 ***
## sqft_lot15    -7.460e+00  9.258e-01  -8.058 8.23e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 212700 on 17276 degrees of freedom
## Multiple R-squared:  0.6093, Adjusted R-squared:  0.609
## F-statistic:  2072 on 13 and 17276 DF,  p-value: < 2.2e-16
```

```
prediccions <- predict(model, newdata = set_training)
```

```
coeficient_r <- 1 - (sum((set_training$price - prediccions)^2) /
                       sum((set_training$price - mean(set_training$price))^2))
cat("Coeficient R quadrat:",coeficient_r,"\n")
```

```
## Coeficient R quadrat: 0.609292
```

```r
fiv_model_ajustat <- 1 / (1 - coeficient_r)
cat("FIV del model ajustat:",fiv_model_ajustat)
```

```
## FIV del model ajustat: 2.559456
```

Per calcular els valors dels FIV per cada una de les variables predictores del model, cal ajustar un model de regressió lineal incloent totes les variables predictores menys una. És a dir s'han de calcular els valors FIV de excloent a cada model una de les variables predictores del model original.

```r
valors_fiv <- data.frame(variable_exclosa = character(ncol(set_training) - 1),
                         fiv = numeric(ncol(set_training) - 1))
for (i in 2:ncol(set_training)) {
  training_aux <- set_training
  columna <- colnames(set_training)[i]
  training_aux <- training_aux[, -i]
  model_sense_variable_i <- lm(price ~ ., data = training_aux)
  prediccions_aux <- predict(model_sense_variable_i, newdata = training_aux)
  coeficient_r_aux <- 1 - (sum((training_aux$price - prediccions_aux)^2) /
                             sum((training_aux$price - mean(training_aux$price))^2))
  fiv <- 1 / (1 - coeficient_r_aux)
  valors_fiv[i, "variable_exclosa"] <- columna
  valors_fiv[i, "fiv"] <- fiv
}
print(valors_fiv[-1, ])
```

```
##    variable_exclosa      fiv
## 2          bedrooms 2.491435
## 3         bathrooms 2.522930
## 4       sqft_living 2.175844
## 5          sqft_lot 2.559445
## 6            floors 2.541138
## 7        waterfront 2.474266
## 8              view 2.506456
## 9         condition 2.550368
## 10    sqft_basement 2.556748
## 11         yr_built 2.373490
## 12     yr_renovated 2.558392
## 13    sqft_living15 2.504313
## 14       sqft_lot15 2.549871
```

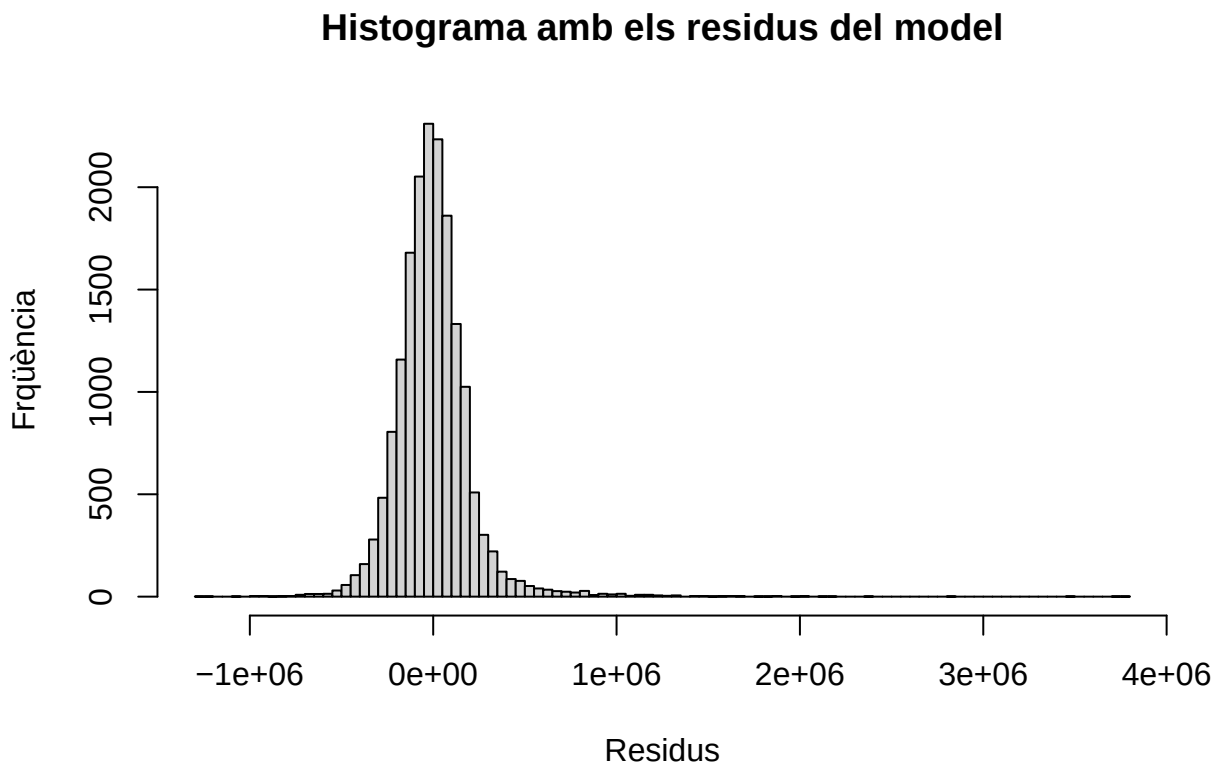
Considerant el FIV del model ajustat i els FIVs dels models individuals, es determina que existeix colinealitat entre les variables. És a dir, l'existència de colinealitat suggereix que les variables predictores estan correlacionades. Per tant, sota la premissa de la seva rellevància teòrica, és a dir, totes les variables incloses en el model són necessàries per a obtenir tots els aspectes importants del fenomen d'estudi (explicar el preu de l'habitatge en funció de les variables seleccionades), i tot i la colinealitat, no es considera excloure cap variable del model.

1.5 Diagnosi del model

Per calcular els residus cal restar els valors observats i els valors reals. Per obtenir els valors predits hi ha dues opcions: la primera utilitzar una crida a la funció *predict()* (utilitzada més amunt per al càlcul del valor de R quadrat) i posteriorment realitzar la resta, o utilitzar la funció *residuals()* que ja retorna directament el càlcul fet.

```r
valors_observats <- set_training$price
residus <- valors_observats - prediccions
```

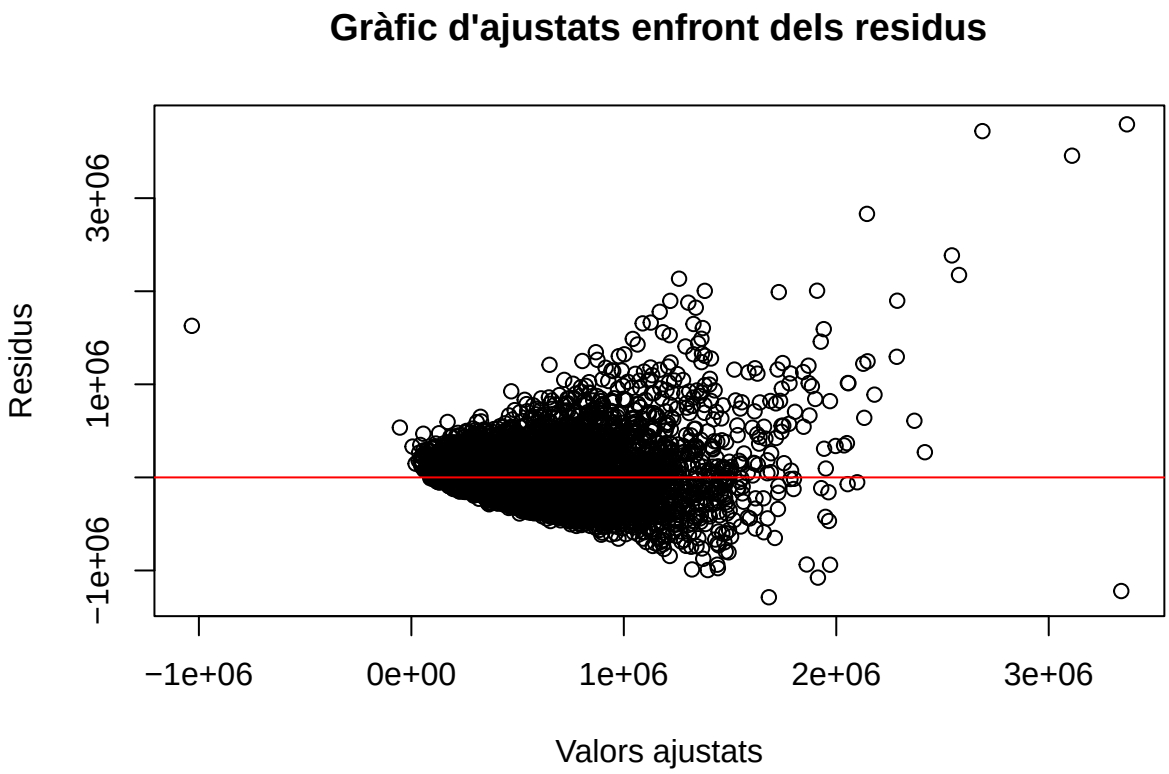
```r
hist(residus, breaks = 100, main = "Histograma amb els residus del model",
     xlab = "Residus", ylab = "Frqüència")
```



S'observa que l'histograma s'assembla a una campana al voltant del valor 0. Aquest fet indica que els residus del model segueixen una distribució normal.

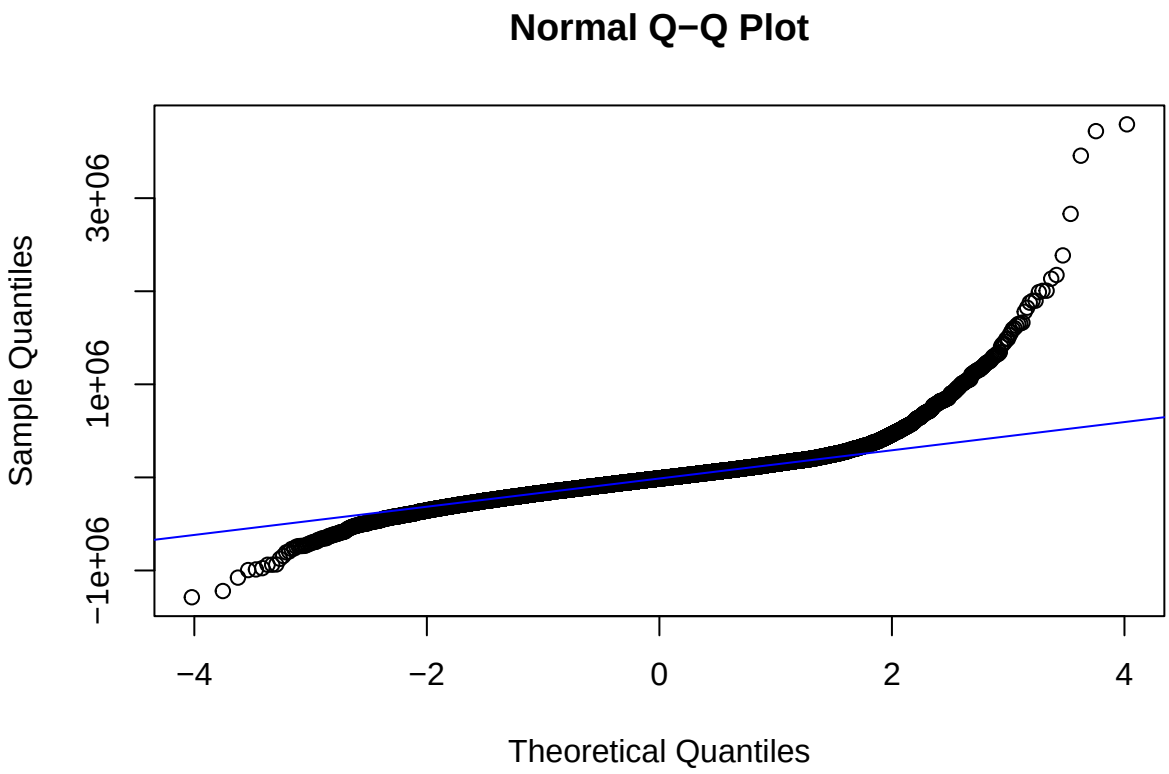
L'esmentada forma de campana al voltant del 0, és un indicador de què el model fa bones prediccions.

```r
plot(fitted(model), residus, main = "Gràfic d'ajustats enfront dels residus",
     xlab = "Valors ajustats", ylab = "Residus")
abline(h = 0, col = "red")
```



En el gràfic de residus enfront dels valors ajustats, es pot observar, que el model presenta certs problemes de dispersió irregular, és a dir, la variància dels residus augmenta a mesura que ho fan els valors ajustats (heteroscedasticitat).

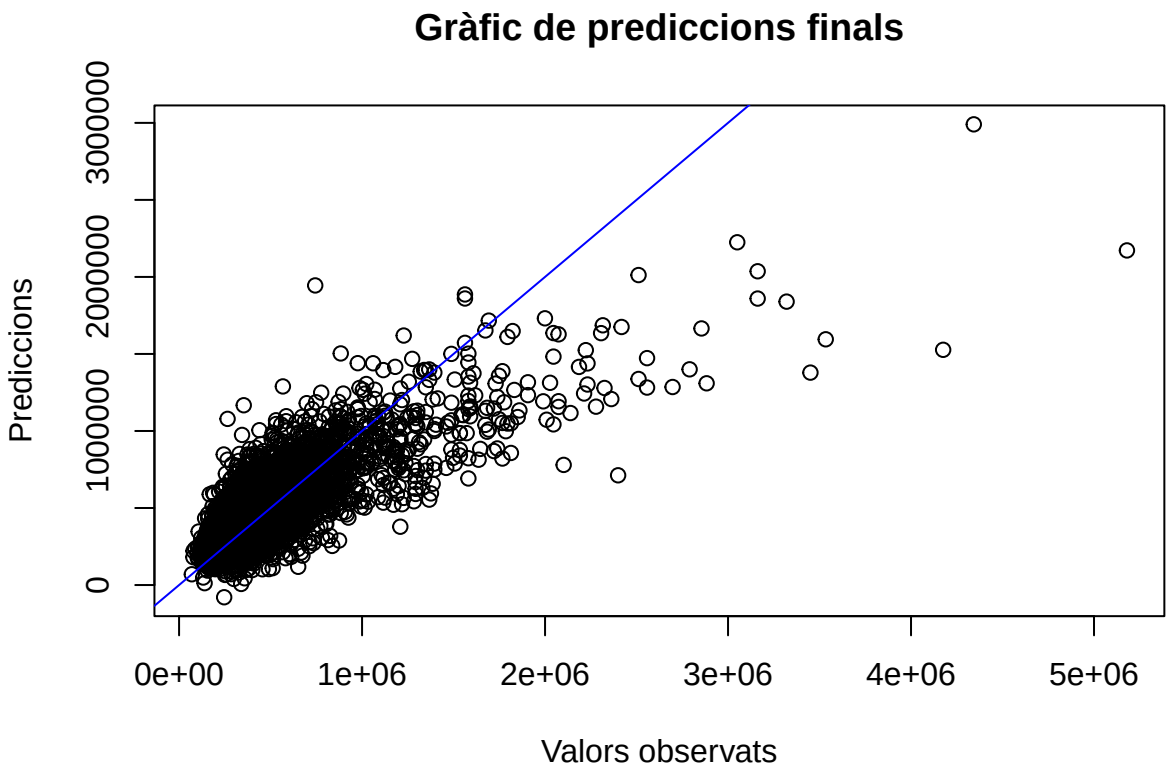
```
qqnorm(residus)
qqline(residus, col = "blue")
```



S'observa que el gràfic QQ presenta dues corbes a les cues. Aquestes curvatures són indicadores d'asimetries, és a dir, llocs on la distribució de les dades no és normal. Atès que la cua dreta és més llarga que l'esquerra es pot assegurar que hi ha una asimetria positiva (quantitat major de valors atípics en l'extrem superior).

1.6 Predicció del model

```
prediccions_finals <- predict(model, newdata = set_test)
plot(set_test$price, prediccions_finals, main = "Gràfic de prediccions finals",
     xlab = "Valors observats", ylab = "Prediccions")
abline(0, 1, col = "blue")
```




```
sumatori <- 0
for (i in 1:length(prediccions_finals)) {
  sumatori <- sumatori + ((set_test$price[i] - prediccions_finals[[i]])^2)
}
rmse <- sqrt(sumatori / nrow(set_test))
cat("Valor RMSE:",rmse,"\n")
```

```
## Valor RMSE: 219223.3
```

```
mitjana_preus <- mean(set_test$price)
cat("Mitjana del preu dels habitatges:",mitjana_preus)
```

```
## Mitjana del preu dels habitatges: 502769
```

Com el RMSE està calculat en el preu dels habitatges (variable depenent), es pot interpretar el resultat del RMSE comparant amb el valor mitjà del preu dels habitatges. S'observa que el RMSE és significativament menor que el preu mitjà dels habitatges, per tant, es conclou que el model té bona precisió.