Proyecto 2. Entrega 2. Árboles de decisión

Pablo Daniel Barillas Moreno, Carné No. 22193
Mathew Cordero Aquino, Carné No. 22982

2025-02-02

Enlace al Repositorio del proyecto 2 - Entrega 2 de minería de datos del Grupo #1

Repositorio en GitHub

0. Descargue los conjuntos de datos.

Para este punto, ya se ha realizado el proceso para descargar del sitio web: House Prices - Advanced Regression Techniques, la data de entrenamiento y la data de prueba, ambos extraídos desde la carpeta "house_prices_data/" en data frames llamados train_data (data de entrenamiento) y test_data (data de prueba), sin convertir automáticamente las variables categóricas en factores (stringsAsFactors = FALSE). Luego, se realiza una inspección inicial de train_data mediante tres funciones: head(train_data), que muestra las primeras filas del dataset; str(train_data), que despliega la estructura del data frame, incluyendo el tipo de cada variable; y summary(train_data), que proporciona un resumen estadístico de las variables numéricas y una descripción general de las categóricas.

```
train_data <- read.csv("house_prices_data/train.csv", stringsAsFactors = FALSE)
test_data <- read.csv("house_prices_data/test.csv", stringsAsFactors = FALSE)

head(train_data)   # Muestra las primeras filas
```

```
##   Id MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape LandContour
## 1  1         60       RL          65    8450   Pave  <NA>      Reg         Lvl
## 2  2         20       RL          80    9600   Pave  <NA>      Reg         Lvl
## 3  3         60       RL          68   11250   Pave  <NA>      IR1         Lvl
## 4  4         70       RL          60    9550   Pave  <NA>      IR1         Lvl
## 5  5         60       RL          84   14260   Pave  <NA>      IR1         Lvl
## 6  6         50       RL          85   14115   Pave  <NA>      IR1         Lvl
##   Utilities LotConfig LandSlope Neighborhood Condition1 Condition2 BldgType
## 1    AllPub    Inside       Gtl      CollgCr       Norm       Norm     1Fam
## 2    AllPub       FR2       Gtl      Veenker      Feedr       Norm     1Fam
## 3    AllPub    Inside       Gtl      CollgCr       Norm       Norm     1Fam
## 4    AllPub    Corner       Gtl      Crawfor       Norm       Norm     1Fam
## 5    AllPub       FR2       Gtl      NoRidge       Norm       Norm     1Fam
## 6    AllPub    Inside       Gtl      Mitchel       Norm       Norm     1Fam
##   HouseStyle OverallQual OverallCond YearBuilt YearRemodAdd RoofStyle RoofMatl
## 1     2Story           7           5      2003         2003     Gable  CompShg
## 2     1Story           6           8      1976         1976     Gable  CompShg
## 3     2Story           7           5      2001         2002     Gable  CompShg
## 4     2Story           7           5      1915         1970     Gable  CompShg
```

```
## 5     2Story           8           5      2000         2000     Gable  CompShg
## 6     1.5Fin           5           5      1993         1995     Gable  CompShg
##   Exterior1st Exterior2nd MasVnrType MasVnrArea ExterQual ExterCond Foundation
## 1     VinylSd     VinylSd    BrkFace        196        Gd        TA      PConc
## 2     MetalSd     MetalSd       None          0        TA        TA     CBlock
## 3     VinylSd     VinylSd    BrkFace        162        Gd        TA      PConc
## 4     Wd Sdng     Wd Shng       None          0        TA        TA     BrkTil
## 5     VinylSd     VinylSd    BrkFace        350        Gd        TA      PConc
## 6     VinylSd     VinylSd       None          0        TA        TA       Wood
##   BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinSF1 BsmtFinType2
## 1       Gd       TA           No          GLQ        706          Unf
## 2       Gd       TA           Gd          ALQ        978          Unf
## 3       Gd       TA           Mn          GLQ        486          Unf
## 4       TA       Gd           No          ALQ        216          Unf
## 5       Gd       TA           Av          GLQ        655          Unf
## 6       Gd       TA           No          GLQ        732          Unf
##   BsmtFinSF2 BsmtUnfSF TotalBsmtSF Heating HeatingQC CentralAir Electrical
## 1          0       150         856    GasA        Ex          Y      SBrkr
## 2          0       284        1262    GasA        Ex          Y      SBrkr
## 3          0       434         920    GasA        Ex          Y      SBrkr
## 4          0       540         756    GasA        Gd          Y      SBrkr
## 5          0       490        1145    GasA        Ex          Y      SBrkr
## 6          0        64         796    GasA        Ex          Y      SBrkr
##   X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath
## 1       856       854            0      1710            1            0        2
## 2      1262         0            0      1262            0            1        2
## 3       920       866            0      1786            1            0        2
## 4       961       756            0      1717            1            0        1
## 5      1145      1053            0      2198            1            0        2
## 6       796       566            0      1362            1            0        1
##   HalfBath BedroomAbvGr KitchenAbvGr KitchenQual TotRmsAbvGrd Functional
## 1        1            3            1          Gd            8        Typ
## 2        0            3            1          TA            6        Typ
## 3        1            3            1          Gd            6        Typ
## 4        0            3            1          Gd            7        Typ
## 5        1            4            1          Gd            9        Typ
## 6        1            1            1          TA            5        Typ
##   Fireplaces FireplaceQu GarageType GarageYrBlt GarageFinish GarageCars
## 1          0       <NA>     Attchd        2003          RFn          2
## 2          1         TA     Attchd        1976          RFn          2
## 3          1         TA     Attchd        2001          RFn          2
## 4          1         Gd     Detchd        1998          Unf          3
## 5          1         TA     Attchd        2000          RFn          3
## 6          0       <NA>     Attchd        1993          Unf          2
##   GarageArea GarageQual GarageCond PavedDrive WoodDeckSF OpenPorchSF
## 1        548         TA         TA          Y          0          61
## 2        460         TA         TA          Y        298           0
## 3        608         TA         TA          Y          0          42
## 4        642         TA         TA          Y          0          35
## 5        836         TA         TA          Y        192          84
## 6        480         TA         TA          Y         40          30
##   EnclosedPorch X3SsnPorch ScreenPorch PoolArea PoolQC Fence MiscFeature
## 1             0          0           0        0   <NA>  <NA>        <NA>
## 2             0          0           0        0   <NA>  <NA>        <NA>
```

```
## 3             0          0           0        0   <NA>  <NA>        <NA>
## 4           272          0           0        0   <NA>  <NA>        <NA>
## 5             0          0           0        0   <NA>  <NA>        <NA>
## 6             0        320           0        0   <NA> MnPrv        Shed
##   MiscVal MoSold YrSold SaleType SaleCondition SalePrice
## 1       0      2   2008       WD        Normal    208500
## 2       0      5   2007       WD        Normal    181500
## 3       0      9   2008       WD        Normal    223500
## 4       0      2   2006       WD       Abnorml    140000
## 5       0     12   2008       WD        Normal    250000
## 6     700     10   2009       WD        Normal    143000
```

```r
str(train_data)    # Muestra la estructura del dataset
```

```
## 'data.frame':    1460 obs. of  81 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning     : chr  "RL" "RL" "RL" "RL" ...
##  $ LotFrontage  : int  65 80 68 60 84 85 75 NA 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
##  $ Street       : chr  "Pave" "Pave" "Pave" "Pave" ...
##  $ Alley        : chr  NA NA NA NA ...
##  $ LotShape     : chr  "Reg" "Reg" "IR1" "IR1" ...
##  $ LandContour  : chr  "Lvl" "Lvl" "Lvl" "Lvl" ...
##  $ Utilities    : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
##  $ LotConfig    : chr  "Inside" "FR2" "Inside" "Corner" ...
##  $ LandSlope    : chr  "Gtl" "Gtl" "Gtl" "Gtl" ...
##  $ Neighborhood : chr  "CollgCr" "Veenker" "CollgCr" "Crawfor" ...
##  $ Condition1   : chr  "Norm" "Feedr" "Norm" "Norm" ...
##  $ Condition2   : chr  "Norm" "Norm" "Norm" "Norm" ...
##  $ BldgType     : chr  "1Fam" "1Fam" "1Fam" "1Fam" ...
##  $ HouseStyle   : chr  "2Story" "1Story" "2Story" "2Story" ...
##  $ OverallQual  : int  7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond  : int  5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
##  $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
##  $ RoofStyle    : chr  "Gable" "Gable" "Gable" "Gable" ...
##  $ RoofMatl     : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
##  $ Exterior1st  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...
##  $ Exterior2nd  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Shng" ...
##  $ MasVnrType   : chr  "BrkFace" "None" "BrkFace" "None" ...
##  $ MasVnrArea   : int  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : chr  "Gd" "TA" "Gd" "TA" ...
##  $ ExterCond    : chr  "TA" "TA" "TA" "TA" ...
##  $ Foundation   : chr  "PConc" "CBlock" "PConc" "BrkTil" ...
##  $ BsmtQual     : chr  "Gd" "Gd" "Gd" "TA" ...
##  $ BsmtCond     : chr  "TA" "TA" "TA" "Gd" ...
##  $ BsmtExposure : chr  "No" "Gd" "Mn" "No" ...
##  $ BsmtFinType1 : chr  "GLQ" "ALQ" "GLQ" "ALQ" ...
##  $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
##  $ BsmtFinType2 : chr  "Unf" "Unf" "Unf" "Unf" ...
##  $ BsmtFinSF2   : int  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : int  150 284 434 540 490 64 317 216 952 140 ...
```

```
##  $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
##  $ Heating      : chr  "GasA" "GasA" "GasA" "GasA" ...
##  $ HeatingQC    : chr  "Ex" "Ex" "Ex" "Gd" ...
##  $ CentralAir   : chr  "Y" "Y" "Y" "Y" ...
##  $ Electrical   : chr  "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
##  $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
##  $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
##  $ BsmtFullBath : int  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath     : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual  : chr  "Gd" "TA" "Gd" "Gd" ...
##  $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : chr  "Typ" "Typ" "Typ" "Typ" ...
##  $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu  : chr  NA "TA" "TA" "Gd" ...
##  $ GarageType   : chr  "Attchd" "Attchd" "Attchd" "Detchd" ...
##  $ GarageYrBlt  : int  2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...
##  $ GarageFinish : chr  "RFn" "RFn" "RFn" "Unf" ...
##  $ GarageCars   : int  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : int  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : chr  "TA" "TA" "TA" "TA" ...
##  $ GarageCond   : chr  "TA" "TA" "TA" "TA" ...
##  $ PavedDrive   : chr  "Y" "Y" "Y" "Y" ...
##  $ WoodDeckSF   : int  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF  : int  61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch: int  0 0 0 272 0 0 0 228 205 0 ...
##  $ X3SsnPorch   : int  0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : chr  NA NA NA NA ...
##  $ Fence        : chr  NA NA NA NA ...
##  $ MiscFeature  : chr  NA NA NA NA ...
##  $ MiscVal      : int  0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold       : int  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold       : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
##  $ SaleType     : chr  "WD" "WD" "WD" "WD" ...
##  $ SaleCondition: chr  "Normal" "Normal" "Normal" "Abnorml" ...
##  $ SalePrice    : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
```

```r
summary(train_data) # Resumen estadístico
```

```
##        Id           MSSubClass      MSZoning          LotFrontage
##  Min.   :   1.0   Min.   : 20.0   Length:1460        Min.   : 21.00
##  1st Qu.: 365.8   1st Qu.: 20.0   Class :character   1st Qu.: 59.00
##  Median : 730.5   Median : 50.0   Mode  :character   Median : 69.00
##  Mean   : 730.5   Mean   : 56.9                      Mean   : 70.05
##  3rd Qu.:1095.2   3rd Qu.: 70.0                      3rd Qu.: 80.00
##  Max.   :1460.0   Max.   :190.0                      Max.   :313.00
```

```
##                                                 NA's   :259
##     LotArea          Street             Alley             LotShape
##  Min.   :  1300   Length:1460        Length:1460        Length:1460
##  1st Qu.:  7554   Class :character   Class :character   Class :character
##  Median :  9478   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 10517
##  3rd Qu.: 11602
##  Max.   :215245
##
##  LandContour         Utilities          LotConfig          LandSlope
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Neighborhood        Condition1         Condition2          BldgType
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   HouseStyle         OverallQual      OverallCond      YearBuilt
##  Length:1460        Min.   : 1.000   Min.   :1.000   Min.   :1872
##  Class :character   1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954
##  Mode  :character   Median : 6.000   Median :5.000   Median :1973
##                     Mean   : 6.099   Mean   :5.575   Mean   :1971
##                     3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2000
##                     Max.   :10.000   Max.   :9.000   Max.   :2010
##
##   YearRemodAdd   RoofStyle           RoofMatl         Exterior1st
##  Min.   :1950   Length:1460        Length:1460        Length:1460
##  1st Qu.:1967   Class :character   Class :character   Class :character
##  Median :1994   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1985
##  3rd Qu.:2004
##  Max.   :2010
##
##  Exterior2nd         MasVnrType          MasVnrArea      ExterQual
##  Length:1460        Length:1460        Min.   :   0.0   Length:1460
##  Class :character   Class :character   1st Qu.:   0.0   Class :character
##  Mode  :character   Mode  :character   Median :   0.0   Mode  :character
##                                        Mean   : 103.7
##                                        3rd Qu.: 166.0
##                                        Max.   :1600.0
##                                        NA's   :8
##   ExterCond          Foundation          BsmtQual           BsmtCond
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
```

```
##
##
##
##  BsmtExposure       BsmtFinType1         BsmtFinSF1     BsmtFinType2
##  Length:1460        Length:1460        Min.   :   0.0   Length:1460
##  Class :character   Class :character   1st Qu.:   0.0   Class :character
##  Mode  :character   Mode  :character   Median : 383.5   Mode  :character
##                                        Mean   : 443.6
##                                        3rd Qu.: 712.2
##                                        Max.   :5644.0
##
##    BsmtFinSF2        BsmtUnfSF       TotalBsmtSF       Heating
##  Min.   :   0.00   Min.   :   0.0   Min.   :   0.0   Length:1460
##  1st Qu.:   0.00   1st Qu.: 223.0   1st Qu.: 795.8   Class :character
##  Median :   0.00   Median : 477.5   Median : 991.5   Mode  :character
##  Mean   :  46.55   Mean   : 567.2   Mean   :1057.4
##  3rd Qu.:   0.00   3rd Qu.: 808.0   3rd Qu.:1298.2
##  Max.   :1474.00   Max.   :2336.0   Max.   :6110.0
##
##   HeatingQC          CentralAir         Electrical          X1stFlrSF
##  Length:1460        Length:1460        Length:1460        Min.   : 334
##  Class :character   Class :character   Class :character   1st Qu.: 882
##  Mode  :character   Mode  :character   Mode  :character   Median :1087
##                                                           Mean   :1163
##                                                           3rd Qu.:1391
##                                                           Max.   :4692
##
##    X2ndFlrSF     LowQualFinSF       GrLivArea     BsmtFullBath
##  Min.   :   0   Min.   :  0.000   Min.   : 334   Min.   :0.0000
##  1st Qu.:   0   1st Qu.:  0.000   1st Qu.:1130   1st Qu.:0.0000
##  Median :   0   Median :  0.000   Median :1464   Median :0.0000
##  Mean   : 347   Mean   :  5.845   Mean   :1515   Mean   :0.4253
##  3rd Qu.: 728   3rd Qu.:  0.000   3rd Qu.:1777   3rd Qu.:1.0000
##  Max.   :2065   Max.   :572.000   Max.   :5642   Max.   :3.0000
##
##   BsmtHalfBath        FullBath        HalfBath       BedroomAbvGr
##  Min.   :0.00000   Min.   :0.000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000
##  Median :0.00000   Median :2.000   Median :0.0000   Median :3.000
##  Mean   :0.05753   Mean   :1.565   Mean   :0.3829   Mean   :2.866
##  3rd Qu.:0.00000   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :2.00000   Max.   :3.000   Max.   :2.0000   Max.   :8.000
##
##   KitchenAbvGr   KitchenQual         TotRmsAbvGrd     Functional
##  Min.   :0.000   Length:1460        Min.   : 2.000   Length:1460
##  1st Qu.:1.000   Class :character   1st Qu.: 5.000   Class :character
##  Median :1.000   Mode  :character   Median : 6.000   Mode  :character
##  Mean   :1.047                      Mean   : 6.518
##  3rd Qu.:1.000                      3rd Qu.: 7.000
##  Max.   :3.000                      Max.   :14.000
##
##    Fireplaces    FireplaceQu         GarageType         GarageYrBlt
##  Min.   :0.000   Length:1460        Length:1460        Min.   :1900
##  1st Qu.:0.000   Class :character   Class :character   1st Qu.:1961
```

```
##  Median :1.000   Mode  :character   Mode  :character   Median :1980
##  Mean   :0.613                                         Mean   :1979
##  3rd Qu.:1.000                                         3rd Qu.:2002
##  Max.   :3.000                                         Max.   :2010
##                                                        NA's   :81
##  GarageFinish         GarageCars      GarageArea      GarageQual
##  Length:1460        Min.   :0.000   Min.   :   0.0   Length:1460
##  Class :character   1st Qu.:1.000   1st Qu.: 334.5   Class :character
##  Mode  :character   Median :2.000   Median : 480.0   Mode  :character
##                     Mean   :1.767   Mean   : 473.0
##                     3rd Qu.:2.000   3rd Qu.: 576.0
##                     Max.   :4.000   Max.   :1418.0
##
##   GarageCond         PavedDrive          WoodDeckSF      OpenPorchSF
##  Length:1460        Length:1460        Min.   :  0.00   Min.   :  0.00
##  Class :character   Class :character   1st Qu.:  0.00   1st Qu.:  0.00
##  Mode  :character   Mode  :character   Median :  0.00   Median : 25.00
##                                        Mean   : 94.24   Mean   : 46.66
##                                        3rd Qu.:168.00   3rd Qu.: 68.00
##                                        Max.   :857.00   Max.   :547.00
##
##  EnclosedPorch      X3SsnPorch      ScreenPorch        PoolArea
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.000
##  1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.000
##  Median :  0.00   Median :  0.00   Median :  0.00   Median :  0.000
##  Mean   : 21.95   Mean   :  3.41   Mean   : 15.06   Mean   :  2.759
##  3rd Qu.:  0.00   3rd Qu.:  0.00   3rd Qu.:  0.00   3rd Qu.:  0.000
##  Max.   :552.00   Max.   :508.00   Max.   :480.00   Max.   :738.000
##
##     PoolQC             Fence           MiscFeature          MiscVal
##  Length:1460        Length:1460        Length:1460        Min.   :    0.00
##  Class :character   Class :character   Class :character   1st Qu.:    0.00
##  Mode  :character   Mode  :character   Mode  :character   Median :    0.00
##                                                           Mean   :   43.49
##                                                           3rd Qu.:    0.00
##                                                           Max.   :15500.00
##
##      MoSold           YrSold       SaleType         SaleCondition
##  Min.   : 1.000   Min.   :2006   Length:1460        Length:1460
##  1st Qu.: 5.000   1st Qu.:2007   Class :character   Class :character
##  Median : 6.000   Median :2008   Mode  :character   Mode  :character
##  Mean   : 6.322   Mean   :2008
##  3rd Qu.: 8.000   3rd Qu.:2009
##  Max.   :12.000   Max.   :2010
##
##    SalePrice
##  Min.   : 34900
##  1st Qu.:129975
##  Median :163000
##  Mean   :180921
##  3rd Qu.:214000
##  Max.   :755000
##
```

1. Use los mismos conjuntos de entrenamiento y prueba que usó para los modelos de regresión lineal en la entrega anterior.

```
# Cargar librerías necesarias
library(readr)
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'

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

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

```
# Fijar semilla para reproducibilidad
set.seed(42)

# Cargar los conjuntos de datos
train_set <- read_csv("house_prices_data/train.csv", show_col_types = FALSE)
test_set <- read_csv("house_prices_data/test.csv", show_col_types = FALSE)

# Verificar dimensiones
cat("Dimensiones del conjunto de entrenamiento:", dim(train_set), "\n")
```

```
## Dimensiones del conjunto de entrenamiento: 1460 81
```

```
cat("Dimensiones del conjunto de prueba:", dim(test_set), "\n")
```

```
## Dimensiones del conjunto de prueba: 1459 80
```

```
# Mostrar los primeros registros
head(train_set)
```

```
## # A tibble: 6 x 81
##      Id MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape
##   <dbl>      <dbl> <chr>          <dbl>   <dbl> <chr>  <chr> <chr>
## 1     1         60 RL                65    8450 Pave   <NA>  Reg
## 2     2         20 RL                80    9600 Pave   <NA>  Reg
## 3     3         60 RL                68   11250 Pave   <NA>  IR1
## 4     4         70 RL                60    9550 Pave   <NA>  IR1
## 5     5         60 RL                84   14260 Pave   <NA>  IR1
## 6     6         50 RL                85   14115 Pave   <NA>  IR1
## # i 73 more variables: LandContour <chr>, Utilities <chr>, LotConfig <chr>,
## #   LandSlope <chr>, Neighborhood <chr>, Condition1 <chr>, Condition2 <chr>,
## #   BldgType <chr>, HouseStyle <chr>, OverallQual <dbl>, OverallCond <dbl>,
```

```
## #   YearBuilt <dbl>, YearRemodAdd <dbl>, RoofStyle <chr>, RoofMatl <chr>,
## #   Exterior1st <chr>, Exterior2nd <chr>, MasVnrType <chr>, MasVnrArea <dbl>,
## #   ExterQual <chr>, ExterCond <chr>, Foundation <chr>, BsmtQual <chr>,
## #   BsmtCond <chr>, BsmtExposure <chr>, BsmtFinType1 <chr>, ...
```

```
head(test_set)
```

```
## # A tibble: 6 x 80
##      Id MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape
##   <dbl>      <dbl> <chr>          <dbl>   <dbl> <chr>  <chr> <chr>
## 1  1461         20 RH                80   11622 Pave   <NA>  Reg
## 2  1462         20 RL                81   14267 Pave   <NA>  IR1
## 3  1463         60 RL                74   13830 Pave   <NA>  IR1
## 4  1464         60 RL                78    9978 Pave   <NA>  IR1
## 5  1465        120 RL                43    5005 Pave   <NA>  IR1
## 6  1466         60 RL                75   10000 Pave   <NA>  IR1
## # i 72 more variables: LandContour <chr>, Utilities <chr>, LotConfig <chr>,
## #   LandSlope <chr>, Neighborhood <chr>, Condition1 <chr>, Condition2 <chr>,
## #   BldgType <chr>, HouseStyle <chr>, OverallQual <dbl>, OverallCond <dbl>,
## #   YearBuilt <dbl>, YearRemodAdd <dbl>, RoofStyle <chr>, RoofMatl <chr>,
## #   Exterior1st <chr>, Exterior2nd <chr>, MasVnrType <chr>, MasVnrArea <dbl>,
## #   ExterQual <chr>, ExterCond <chr>, Foundation <chr>, BsmtQual <chr>,
## #   BsmtCond <chr>, BsmtExposure <chr>, BsmtFinType1 <chr>, ...
```

```
# Resumen estadístico de cada conjunto
summary(train_set)
```

```
##        Id           MSSubClass      MSZoning          LotFrontage
##  Min.   :   1.0   Min.   : 20.0   Length:1460        Min.   : 21.00
##  1st Qu.: 365.8   1st Qu.: 20.0   Class :character   1st Qu.: 59.00
##  Median : 730.5   Median : 50.0   Mode  :character   Median : 69.00
##  Mean   : 730.5   Mean   : 56.9                      Mean   : 70.05
##  3rd Qu.:1095.2   3rd Qu.: 70.0                      3rd Qu.: 80.00
##  Max.   :1460.0   Max.   :190.0                      Max.   :313.00
##                                                      NA's   :259
##     LotArea          Street             Alley             LotShape
##  Min.   :  1300   Length:1460        Length:1460        Length:1460
##  1st Qu.:  7554   Class :character   Class :character   Class :character
##  Median :  9478   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 10517
##  3rd Qu.: 11602
##  Max.   :215245
##
##  LandContour         Utilities          LotConfig          LandSlope
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Neighborhood        Condition1         Condition2          BldgType
```

```
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   HouseStyle         OverallQual      OverallCond      YearBuilt
##  Length:1460        Min.   : 1.000   Min.   :1.000   Min.   :1872
##  Class :character   1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954
##  Mode  :character   Median : 6.000   Median :5.000   Median :1973
##                     Mean   : 6.099   Mean   :5.575   Mean   :1971
##                     3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2000
##                     Max.   :10.000   Max.   :9.000   Max.   :2010
##
##   YearRemodAdd   RoofStyle           RoofMatl         Exterior1st
##  Min.   :1950   Length:1460        Length:1460        Length:1460
##  1st Qu.:1967   Class :character   Class :character   Class :character
##  Median :1994   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1985
##  3rd Qu.:2004
##  Max.   :2010
##
##  Exterior2nd         MasVnrType          MasVnrArea      ExterQual
##  Length:1460        Length:1460        Min.   :   0.0   Length:1460
##  Class :character   Class :character   1st Qu.:   0.0   Class :character
##  Mode  :character   Mode  :character   Median :   0.0   Mode  :character
##                                        Mean   : 103.7
##                                        3rd Qu.: 166.0
##                                        Max.   :1600.0
##                                        NA's   :8
##   ExterCond          Foundation          BsmtQual           BsmtCond
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  BsmtExposure       BsmtFinType1         BsmtFinSF1     BsmtFinType2
##  Length:1460        Length:1460        Min.   :   0.0   Length:1460
##  Class :character   Class :character   1st Qu.:   0.0   Class :character
##  Mode  :character   Mode  :character   Median : 383.5   Mode  :character
##                                        Mean   : 443.6
##                                        3rd Qu.: 712.2
##                                        Max.   :5644.0
##
##    BsmtFinSF2        BsmtUnfSF       TotalBsmtSF       Heating
##  Min.   :   0.00   Min.   :   0.0   Min.   :   0.0   Length:1460
##  1st Qu.:   0.00   1st Qu.: 223.0   1st Qu.: 795.8   Class :character
##  Median :   0.00   Median : 477.5   Median : 991.5   Mode  :character
##  Mean   :  46.55   Mean   : 567.2   Mean   :1057.4
##  3rd Qu.:   0.00   3rd Qu.: 808.0   3rd Qu.:1298.2
##  Max.   :1474.00   Max.   :2336.0   Max.   :6110.0
```

```
##
##   HeatingQC          CentralAir         Electrical           1stFlrSF
##  Length:1460        Length:1460        Length:1460        Min.   : 334
##  Class :character   Class :character   Class :character   1st Qu.: 882
##  Mode  :character   Mode  :character   Mode  :character   Median :1087
##                                                           Mean   :1163
##                                                           3rd Qu.:1391
##                                                           Max.   :4692
##
##     2ndFlrSF     LowQualFinSF       GrLivArea     BsmtFullBath
##  Min.   :   0   Min.   :  0.000   Min.   : 334   Min.   :0.0000
##  1st Qu.:   0   1st Qu.:  0.000   1st Qu.:1130   1st Qu.:0.0000
##  Median :   0   Median :  0.000   Median :1464   Median :0.0000
##  Mean   : 347   Mean   :  5.845   Mean   :1515   Mean   :0.4253
##  3rd Qu.: 728   3rd Qu.:  0.000   3rd Qu.:1777   3rd Qu.:1.0000
##  Max.   :2065   Max.   :572.000   Max.   :5642   Max.   :3.0000
##
##   BsmtHalfBath        FullBath        HalfBath       BedroomAbvGr
##  Min.   :0.00000   Min.   :0.000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000
##  Median :0.00000   Median :2.000   Median :0.0000   Median :3.000
##  Mean   :0.05753   Mean   :1.565   Mean   :0.3829   Mean   :2.866
##  3rd Qu.:0.00000   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :2.00000   Max.   :3.000   Max.   :2.0000   Max.   :8.000
##
##   KitchenAbvGr   KitchenQual         TotRmsAbvGrd     Functional
##  Min.   :0.000   Length:1460        Min.   : 2.000   Length:1460
##  1st Qu.:1.000   Class :character   1st Qu.: 5.000   Class :character
##  Median :1.000   Mode  :character   Median : 6.000   Mode  :character
##  Mean   :1.047                      Mean   : 6.518
##  3rd Qu.:1.000                      3rd Qu.: 7.000
##  Max.   :3.000                      Max.   :14.000
##
##    Fireplaces    FireplaceQu         GarageType         GarageYrBlt
##  Min.   :0.000   Length:1460        Length:1460        Min.   :1900
##  1st Qu.:0.000   Class :character   Class :character   1st Qu.:1961
##  Median :1.000   Mode  :character   Mode  :character   Median :1980
##  Mean   :0.613                                         Mean   :1979
##  3rd Qu.:1.000                                         3rd Qu.:2002
##  Max.   :3.000                                         Max.   :2010
##                                                        NA's   :81
##  GarageFinish         GarageCars      GarageArea      GarageQual
##  Length:1460        Min.   :0.000   Min.   :   0.0   Length:1460
##  Class :character   1st Qu.:1.000   1st Qu.: 334.5   Class :character
##  Mode  :character   Median :2.000   Median : 480.0   Mode  :character
##                     Mean   :1.767   Mean   : 473.0
##                     3rd Qu.:2.000   3rd Qu.: 576.0
##                     Max.   :4.000   Max.   :1418.0
##
##   GarageCond         PavedDrive          WoodDeckSF      OpenPorchSF
##  Length:1460        Length:1460        Min.   :  0.00   Min.   :  0.00
##  Class :character   Class :character   1st Qu.:  0.00   1st Qu.:  0.00
##  Mode  :character   Mode  :character   Median :  0.00   Median : 25.00
##                                        Mean   : 94.24   Mean   : 46.66
```

```
##                                      3rd Qu.:168.00   3rd Qu.: 68.00
##                                      Max.   :857.00   Max.   :547.00
##
##  EnclosedPorch      3SsnPorch       ScreenPorch        PoolArea
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.000
##  1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.000
##  Median :  0.00   Median :  0.00   Median :  0.00   Median :  0.000
##  Mean   : 21.95   Mean   :  3.41   Mean   : 15.06   Mean   :  2.759
##  3rd Qu.:  0.00   3rd Qu.:  0.00   3rd Qu.:  0.00   3rd Qu.:  0.000
##  Max.   :552.00   Max.   :508.00   Max.   :480.00   Max.   :738.000
##
##     PoolQC             Fence           MiscFeature           MiscVal
##  Length:1460        Length:1460        Length:1460        Min.   :    0.00
##  Class :character   Class :character   Class :character   1st Qu.:    0.00
##  Mode  :character   Mode  :character   Mode  :character   Median :    0.00
##                                                           Mean   :   43.49
##                                                           3rd Qu.:    0.00
##                                                           Max.   :15500.00
##
##      MoSold           YrSold       SaleType         SaleCondition
##  Min.   : 1.000   Min.   :2006   Length:1460        Length:1460
##  1st Qu.: 5.000   1st Qu.:2007   Class :character   Class :character
##  Median : 6.000   Median :2008   Mode  :character   Mode  :character
##  Mean   : 6.322   Mean   :2008
##  3rd Qu.: 8.000   3rd Qu.:2009
##  Max.   :12.000   Max.   :2010
##
##    SalePrice
##  Min.   : 34900
##  1st Qu.:129975
##  Median :163000
##  Mean   :180921
##  3rd Qu.:214000
##  Max.   :755000
##
```

```
summary(test_set)
```

```
##        Id         MSSubClass       MSZoning          LotFrontage
##  Min.   :1461   Min.   : 20.00   Length:1459        Min.   : 21.00
##  1st Qu.:1826   1st Qu.: 20.00   Class :character   1st Qu.: 58.00
##  Median :2190   Median : 50.00   Mode  :character   Median : 67.00
##  Mean   :2190   Mean   : 57.38                      Mean   : 68.58
##  3rd Qu.:2554   3rd Qu.: 70.00                      3rd Qu.: 80.00
##  Max.   :2919   Max.   :190.00                      Max.   :200.00
##                                                     NA's   :227
##     LotArea          Street             Alley             LotShape
##  Min.   : 1470   Length:1459        Length:1459        Length:1459
##  1st Qu.: 7391   Class :character   Class :character   Class :character
##  Median : 9399   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 9819
##  3rd Qu.:11518
##  Max.   :56600
```

```
##
##  LandContour         Utilities          LotConfig          LandSlope
##  Length:1459        Length:1459        Length:1459        Length:1459
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Neighborhood        Condition1         Condition2          BldgType
##  Length:1459        Length:1459        Length:1459        Length:1459
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   HouseStyle         OverallQual      OverallCond      YearBuilt
##  Length:1459        Min.   : 1.000   Min.   :1.000   Min.   :1879
##  Class :character   1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1953
##  Mode  :character   Median : 6.000   Median :5.000   Median :1973
##                     Mean   : 6.079   Mean   :5.554   Mean   :1971
##                     3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2001
##                     Max.   :10.000   Max.   :9.000   Max.   :2010
##
##   YearRemodAdd   RoofStyle           RoofMatl         Exterior1st
##  Min.   :1950   Length:1459        Length:1459        Length:1459
##  1st Qu.:1963   Class :character   Class :character   Class :character
##  Median :1992   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1984
##  3rd Qu.:2004
##  Max.   :2010
##
##  Exterior2nd         MasVnrType          MasVnrArea      ExterQual
##  Length:1459        Length:1459        Min.   :   0.0   Length:1459
##  Class :character   Class :character   1st Qu.:   0.0   Class :character
##  Mode  :character   Mode  :character   Median :   0.0   Mode  :character
##                                        Mean   : 100.7
##                                        3rd Qu.: 164.0
##                                        Max.   :1290.0
##                                        NA's   :15
##   ExterCond          Foundation          BsmtQual           BsmtCond
##  Length:1459        Length:1459        Length:1459        Length:1459
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  BsmtExposure       BsmtFinType1         BsmtFinSF1     BsmtFinType2
##  Length:1459        Length:1459        Min.   :   0.0   Length:1459
##  Class :character   Class :character   1st Qu.:   0.0   Class :character
##  Mode  :character   Mode  :character   Median : 350.5   Mode  :character
##                                        Mean   : 439.2
```

```
##                                     3rd Qu.: 753.5
##                                     Max.   :4010.0
##                                     NA's   :1
##    BsmtFinSF2        BsmtUnfSF       TotalBsmtSF     Heating
##  Min.   :   0.00   Min.   :   0.0   Min.   :   0   Length:1459
##  1st Qu.:   0.00   1st Qu.: 219.2   1st Qu.: 784   Class :character
##  Median :   0.00   Median : 460.0   Median : 988   Mode  :character
##  Mean   :  52.62   Mean   : 554.3   Mean   :1046
##  3rd Qu.:   0.00   3rd Qu.: 797.8   3rd Qu.:1305
##  Max.   :1526.00   Max.   :2140.0   Max.   :5095
##  NA's   :1         NA's   :1        NA's   :1
##   HeatingQC          CentralAir         Electrical           1stFlrSF
##  Length:1459        Length:1459        Length:1459        Min.   : 407.0
##  Class :character   Class :character   Class :character   1st Qu.: 873.5
##  Mode  :character   Mode  :character   Mode  :character   Median :1079.0
##                                                           Mean   :1156.5
##                                                           3rd Qu.:1382.5
##                                                           Max.   :5095.0
##
##     2ndFlrSF     LowQualFinSF        GrLivArea     BsmtFullBath
##  Min.   :   0   Min.   :   0.000   Min.   : 407   Min.   :0.0000
##  1st Qu.:   0   1st Qu.:   0.000   1st Qu.:1118   1st Qu.:0.0000
##  Median :   0   Median :   0.000   Median :1432   Median :0.0000
##  Mean   : 326   Mean   :   3.543   Mean   :1486   Mean   :0.4345
##  3rd Qu.: 676   3rd Qu.:   0.000   3rd Qu.:1721   3rd Qu.:1.0000
##  Max.   :1862   Max.   :1064.000   Max.   :5095   Max.   :3.0000
##                                                   NA's   :2
##   BsmtHalfBath       FullBath        HalfBath       BedroomAbvGr
##  Min.   :0.0000   Min.   :0.000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000
##  Median :0.0000   Median :2.000   Median :0.0000   Median :3.000
##  Mean   :0.0652   Mean   :1.571   Mean   :0.3777   Mean   :2.854
##  3rd Qu.:0.0000   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :2.0000   Max.   :4.000   Max.   :2.0000   Max.   :6.000
##  NA's   :2
##   KitchenAbvGr   KitchenQual         TotRmsAbvGrd     Functional
##  Min.   :0.000   Length:1459        Min.   : 3.000   Length:1459
##  1st Qu.:1.000   Class :character   1st Qu.: 5.000   Class :character
##  Median :1.000   Mode  :character   Median : 6.000   Mode  :character
##  Mean   :1.042                      Mean   : 6.385
##  3rd Qu.:1.000                      3rd Qu.: 7.000
##  Max.   :2.000                      Max.   :15.000
##
##    Fireplaces     FireplaceQu         GarageType         GarageYrBlt
##  Min.   :0.0000   Length:1459        Length:1459        Min.   :1895
##  1st Qu.:0.0000   Class :character   Class :character   1st Qu.:1959
##  Median :0.0000   Mode  :character   Mode  :character   Median :1979
##  Mean   :0.5812                                         Mean   :1978
##  3rd Qu.:1.0000                                         3rd Qu.:2002
##  Max.   :4.0000                                         Max.   :2207
##                                                         NA's   :78
##  GarageFinish         GarageCars      GarageArea      GarageQual
##  Length:1459        Min.   :0.000   Min.   :   0.0   Length:1459
##  Class :character   1st Qu.:1.000   1st Qu.: 318.0   Class :character
```

```
##  Mode  :character   Median :2.000   Median : 480.0   Mode  :character
##                     Mean   :1.766   Mean   : 472.8
##                     3rd Qu.:2.000   3rd Qu.: 576.0
##                     Max.   :5.000   Max.   :1488.0
##                     NA's   :1       NA's   :1
##   GarageCond         PavedDrive          WoodDeckSF       OpenPorchSF
##  Length:1459        Length:1459        Min.   :   0.00   Min.   :  0.00
##  Class :character   Class :character   1st Qu.:   0.00   1st Qu.:  0.00
##  Mode  :character   Mode  :character   Median :   0.00   Median : 28.00
##                                        Mean   :  93.17   Mean   : 48.31
##                                        3rd Qu.: 168.00   3rd Qu.: 72.00
##                                        Max.   :1424.00   Max.   :742.00
##
##  EnclosedPorch       3SsnPorch        ScreenPorch        PoolArea
##  Min.   :   0.00   Min.   :  0.000   Min.   :  0.00   Min.   :  0.000
##  1st Qu.:   0.00   1st Qu.:  0.000   1st Qu.:  0.00   1st Qu.:  0.000
##  Median :   0.00   Median :  0.000   Median :  0.00   Median :  0.000
##  Mean   :  24.24   Mean   :  1.794   Mean   : 17.06   Mean   :  1.744
##  3rd Qu.:   0.00   3rd Qu.:  0.000   3rd Qu.:  0.00   3rd Qu.:  0.000
##  Max.   :1012.00   Max.   :360.000   Max.   :576.00   Max.   :800.000
##
##     PoolQC             Fence           MiscFeature          MiscVal
##  Length:1459        Length:1459        Length:1459        Min.   :    0.00
##  Class :character   Class :character   Class :character   1st Qu.:    0.00
##  Mode  :character   Mode  :character   Mode  :character   Median :    0.00
##                                                           Mean   :   58.17
##                                                           3rd Qu.:    0.00
##                                                           Max.   :17000.00
##
##      MoSold           YrSold       SaleType         SaleCondition
##  Min.   : 1.000   Min.   :2006   Length:1459        Length:1459
##  1st Qu.: 4.000   1st Qu.:2007   Class :character   Class :character
##  Median : 6.000   Median :2008   Mode  :character   Mode  :character
##  Mean   : 6.104   Mean   :2008
##  3rd Qu.: 8.000   3rd Qu.:2009
##  Max.   :12.000   Max.   :2010
##
```

```r
# Verificar los tipos de datos en cada conjunto
str(train_set)
```

```
## spc_tbl_ [1,460 x 81] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id           : num [1:1460] 1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : num [1:1460] 60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning     : chr [1:1460] "RL" "RL" "RL" "RL" ...
##  $ LotFrontage  : num [1:1460] 65 80 68 60 84 85 75 NA 51 50 ...
##  $ LotArea      : num [1:1460] 8450 9600 11250 9550 14260 ...
##  $ Street       : chr [1:1460] "Pave" "Pave" "Pave" "Pave" ...
##  $ Alley        : chr [1:1460] NA NA NA NA ...
##  $ LotShape     : chr [1:1460] "Reg" "Reg" "IR1" "IR1" ...
##  $ LandContour  : chr [1:1460] "Lvl" "Lvl" "Lvl" "Lvl" ...
##  $ Utilities    : chr [1:1460] "AllPub" "AllPub" "AllPub" "AllPub" ...
##  $ LotConfig    : chr [1:1460] "Inside" "FR2" "Inside" "Corner" ...
```

```
##  $ LandSlope    : chr [1:1460] "Gtl" "Gtl" "Gtl" "Gtl" ...
##  $ Neighborhood : chr [1:1460] "CollgCr" "Veenker" "CollgCr" "Crawfor" ...
##  $ Condition1   : chr [1:1460] "Norm" "Feedr" "Norm" "Norm" ...
##  $ Condition2   : chr [1:1460] "Norm" "Norm" "Norm" "Norm" ...
##  $ BldgType     : chr [1:1460] "1Fam" "1Fam" "1Fam" "1Fam" ...
##  $ HouseStyle   : chr [1:1460] "2Story" "1Story" "2Story" "2Story" ...
##  $ OverallQual  : num [1:1460] 7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond  : num [1:1460] 5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt    : num [1:1460] 2003 1976 2001 1915 2000 ...
##  $ YearRemodAdd : num [1:1460] 2003 1976 2002 1970 2000 ...
##  $ RoofStyle    : chr [1:1460] "Gable" "Gable" "Gable" "Gable" ...
##  $ RoofMatl     : chr [1:1460] "CompShg" "CompShg" "CompShg" "CompShg" ...
##  $ Exterior1st  : chr [1:1460] "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...
##  $ Exterior2nd  : chr [1:1460] "VinylSd" "MetalSd" "VinylSd" "Wd Shng" ...
##  $ MasVnrType   : chr [1:1460] "BrkFace" "None" "BrkFace" "None" ...
##  $ MasVnrArea   : num [1:1460] 196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : chr [1:1460] "Gd" "TA" "Gd" "TA" ...
##  $ ExterCond    : chr [1:1460] "TA" "TA" "TA" "TA" ...
##  $ Foundation   : chr [1:1460] "PConc" "CBlock" "PConc" "BrkTil" ...
##  $ BsmtQual     : chr [1:1460] "Gd" "Gd" "Gd" "TA" ...
##  $ BsmtCond     : chr [1:1460] "TA" "TA" "TA" "Gd" ...
##  $ BsmtExposure : chr [1:1460] "No" "Gd" "Mn" "No" ...
##  $ BsmtFinType1 : chr [1:1460] "GLQ" "ALQ" "GLQ" "ALQ" ...
##  $ BsmtFinSF1   : num [1:1460] 706 978 486 216 655 ...
##  $ BsmtFinType2 : chr [1:1460] "Unf" "Unf" "Unf" "Unf" ...
##  $ BsmtFinSF2   : num [1:1460] 0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : num [1:1460] 150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : num [1:1460] 856 1262 920 756 1145 ...
##  $ Heating      : chr [1:1460] "GasA" "GasA" "GasA" "GasA" ...
##  $ HeatingQC    : chr [1:1460] "Ex" "Ex" "Ex" "Gd" ...
##  $ CentralAir   : chr [1:1460] "Y" "Y" "Y" "Y" ...
##  $ Electrical   : chr [1:1460] "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
##  $ 1stFlrSF     : num [1:1460] 856 1262 920 961 1145 ...
##  $ 2ndFlrSF     : num [1:1460] 854 0 866 756 1053 ...
##  $ LowQualFinSF : num [1:1460] 0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : num [1:1460] 1710 1262 1786 1717 2198 ...
##  $ BsmtFullBath : num [1:1460] 1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : num [1:1460] 0 1 0 0 0 0 0 0 0 0 ...
##  $ FullBath     : num [1:1460] 2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath     : num [1:1460] 1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr : num [1:1460] 3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr : num [1:1460] 1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual  : chr [1:1460] "Gd" "TA" "Gd" "Gd" ...
##  $ TotRmsAbvGrd : num [1:1460] 8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : chr [1:1460] "Typ" "Typ" "Typ" "Typ" ...
##  $ Fireplaces   : num [1:1460] 0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu  : chr [1:1460] NA "TA" "TA" "Gd" ...
##  $ GarageType   : chr [1:1460] "Attchd" "Attchd" "Attchd" "Detchd" ...
##  $ GarageYrBlt  : num [1:1460] 2003 1976 2001 1998 2000 ...
##  $ GarageFinish : chr [1:1460] "RFn" "RFn" "RFn" "Unf" ...
##  $ GarageCars   : num [1:1460] 2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : num [1:1460] 548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : chr [1:1460] "TA" "TA" "TA" "TA" ...
##  $ GarageCond   : chr [1:1460] "TA" "TA" "TA" "TA" ...
```

```
##  $ PavedDrive   : chr [1:1460] "Y" "Y" "Y" "Y" ...
##  $ WoodDeckSF   : num [1:1460] 0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF  : num [1:1460] 61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch: num [1:1460] 0 0 0 272 0 0 0 228 205 0 ...
##  $ 3SsnPorch    : num [1:1460] 0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch  : num [1:1460] 0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea     : num [1:1460] 0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : chr [1:1460] NA NA NA NA ...
##  $ Fence        : chr [1:1460] NA NA NA NA ...
##  $ MiscFeature  : chr [1:1460] NA NA NA NA ...
##  $ MiscVal      : num [1:1460] 0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold       : num [1:1460] 2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold       : num [1:1460] 2008 2007 2008 2006 2008 ...
##  $ SaleType     : chr [1:1460] "WD" "WD" "WD" "WD" ...
##  $ SaleCondition: chr [1:1460] "Normal" "Normal" "Normal" "Abnorml" ...
##  $ SalePrice    : num [1:1460] 208500 181500 223500 140000 250000 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   MSSubClass = col_double(),
##   ..   MSZoning = col_character(),
##   ..   LotFrontage = col_double(),
##   ..   LotArea = col_double(),
##   ..   Street = col_character(),
##   ..   Alley = col_character(),
##   ..   LotShape = col_character(),
##   ..   LandContour = col_character(),
##   ..   Utilities = col_character(),
##   ..   LotConfig = col_character(),
##   ..   LandSlope = col_character(),
##   ..   Neighborhood = col_character(),
##   ..   Condition1 = col_character(),
##   ..   Condition2 = col_character(),
##   ..   BldgType = col_character(),
##   ..   HouseStyle = col_character(),
##   ..   OverallQual = col_double(),
##   ..   OverallCond = col_double(),
##   ..   YearBuilt = col_double(),
##   ..   YearRemodAdd = col_double(),
##   ..   RoofStyle = col_character(),
##   ..   RoofMatl = col_character(),
##   ..   Exterior1st = col_character(),
##   ..   Exterior2nd = col_character(),
##   ..   MasVnrType = col_character(),
##   ..   MasVnrArea = col_double(),
##   ..   ExterQual = col_character(),
##   ..   ExterCond = col_character(),
##   ..   Foundation = col_character(),
##   ..   BsmtQual = col_character(),
##   ..   BsmtCond = col_character(),
##   ..   BsmtExposure = col_character(),
##   ..   BsmtFinType1 = col_character(),
##   ..   BsmtFinSF1 = col_double(),
##   ..   BsmtFinType2 = col_character(),
```

```
##   ..   BsmtFinSF2 = col_double(),
##   ..   BsmtUnfSF = col_double(),
##   ..   TotalBsmtSF = col_double(),
##   ..   Heating = col_character(),
##   ..   HeatingQC = col_character(),
##   ..   CentralAir = col_character(),
##   ..   Electrical = col_character(),
##   ..   `1stFlrSF` = col_double(),
##   ..   `2ndFlrSF` = col_double(),
##   ..   LowQualFinSF = col_double(),
##   ..   GrLivArea = col_double(),
##   ..   BsmtFullBath = col_double(),
##   ..   BsmtHalfBath = col_double(),
##   ..   FullBath = col_double(),
##   ..   HalfBath = col_double(),
##   ..   BedroomAbvGr = col_double(),
##   ..   KitchenAbvGr = col_double(),
##   ..   KitchenQual = col_character(),
##   ..   TotRmsAbvGrd = col_double(),
##   ..   Functional = col_character(),
##   ..   Fireplaces = col_double(),
##   ..   FireplaceQu = col_character(),
##   ..   GarageType = col_character(),
##   ..   GarageYrBlt = col_double(),
##   ..   GarageFinish = col_character(),
##   ..   GarageCars = col_double(),
##   ..   GarageArea = col_double(),
##   ..   GarageQual = col_character(),
##   ..   GarageCond = col_character(),
##   ..   PavedDrive = col_character(),
##   ..   WoodDeckSF = col_double(),
##   ..   OpenPorchSF = col_double(),
##   ..   EnclosedPorch = col_double(),
##   ..   `3SsnPorch` = col_double(),
##   ..   ScreenPorch = col_double(),
##   ..   PoolArea = col_double(),
##   ..   PoolQC = col_character(),
##   ..   Fence = col_character(),
##   ..   MiscFeature = col_character(),
##   ..   MiscVal = col_double(),
##   ..   MoSold = col_double(),
##   ..   YrSold = col_double(),
##   ..   SaleType = col_character(),
##   ..   SaleCondition = col_character(),
##   ..   SalePrice = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(test_set)
```

```
## spc_tbl_ [1,459 x 80] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id           : num [1:1459] 1461 1462 1463 1464 1465 ...
##  $ MSSubClass   : num [1:1459] 20 20 60 60 120 60 20 60 20 20 ...
```

```
##  $ MSZoning     : chr [1:1459] "RH" "RL" "RL" "RL" ...
##  $ LotFrontage  : num [1:1459] 80 81 74 78 43 75 NA 63 85 70 ...
##  $ LotArea      : num [1:1459] 11622 14267 13830 9978 5005 ...
##  $ Street       : chr [1:1459] "Pave" "Pave" "Pave" "Pave" ...
##  $ Alley        : chr [1:1459] NA NA NA NA ...
##  $ LotShape     : chr [1:1459] "Reg" "IR1" "IR1" "IR1" ...
##  $ LandContour  : chr [1:1459] "Lvl" "Lvl" "Lvl" "Lvl" ...
##  $ Utilities    : chr [1:1459] "AllPub" "AllPub" "AllPub" "AllPub" ...
##  $ LotConfig    : chr [1:1459] "Inside" "Corner" "Inside" "Inside" ...
##  $ LandSlope    : chr [1:1459] "Gtl" "Gtl" "Gtl" "Gtl" ...
##  $ Neighborhood : chr [1:1459] "NAmes" "NAmes" "Gilbert" "Gilbert" ...
##  $ Condition1   : chr [1:1459] "Feedr" "Norm" "Norm" "Norm" ...
##  $ Condition2   : chr [1:1459] "Norm" "Norm" "Norm" "Norm" ...
##  $ BldgType     : chr [1:1459] "1Fam" "1Fam" "1Fam" "1Fam" ...
##  $ HouseStyle   : chr [1:1459] "1Story" "1Story" "2Story" "2Story" ...
##  $ OverallQual  : num [1:1459] 5 6 5 6 8 6 6 6 7 4 ...
##  $ OverallCond  : num [1:1459] 6 6 5 6 5 5 7 5 5 5 ...
##  $ YearBuilt    : num [1:1459] 1961 1958 1997 1998 1992 ...
##  $ YearRemodAdd : num [1:1459] 1961 1958 1998 1998 1992 ...
##  $ RoofStyle    : chr [1:1459] "Gable" "Hip" "Gable" "Gable" ...
##  $ RoofMatl     : chr [1:1459] "CompShg" "CompShg" "CompShg" "CompShg" ...
##  $ Exterior1st  : chr [1:1459] "VinylSd" "Wd Sdng" "VinylSd" "VinylSd" ...
##  $ Exterior2nd  : chr [1:1459] "VinylSd" "Wd Sdng" "VinylSd" "VinylSd" ...
##  $ MasVnrType   : chr [1:1459] "None" "BrkFace" "None" "BrkFace" ...
##  $ MasVnrArea   : num [1:1459] 0 108 0 20 0 0 0 0 0 0 ...
##  $ ExterQual    : chr [1:1459] "TA" "TA" "TA" "TA" ...
##  $ ExterCond    : chr [1:1459] "TA" "TA" "TA" "TA" ...
##  $ Foundation   : chr [1:1459] "CBlock" "CBlock" "PConc" "PConc" ...
##  $ BsmtQual     : chr [1:1459] "TA" "TA" "Gd" "TA" ...
##  $ BsmtCond     : chr [1:1459] "TA" "TA" "TA" "TA" ...
##  $ BsmtExposure : chr [1:1459] "No" "No" "No" "No" ...
##  $ BsmtFinType1 : chr [1:1459] "Rec" "ALQ" "GLQ" "GLQ" ...
##  $ BsmtFinSF1   : num [1:1459] 468 923 791 602 263 0 935 0 637 804 ...
##  $ BsmtFinType2 : chr [1:1459] "LwQ" "Unf" "Unf" "Unf" ...
##  $ BsmtFinSF2   : num [1:1459] 144 0 0 0 0 0 0 0 0 78 ...
##  $ BsmtUnfSF    : num [1:1459] 270 406 137 324 1017 ...
##  $ TotalBsmtSF  : num [1:1459] 882 1329 928 926 1280 ...
##  $ Heating      : chr [1:1459] "GasA" "GasA" "GasA" "GasA" ...
##  $ HeatingQC    : chr [1:1459] "TA" "TA" "Gd" "Ex" ...
##  $ CentralAir   : chr [1:1459] "Y" "Y" "Y" "Y" ...
##  $ Electrical   : chr [1:1459] "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
##  $ 1stFlrSF     : num [1:1459] 896 1329 928 926 1280 ...
##  $ 2ndFlrSF     : num [1:1459] 0 0 701 678 0 892 0 676 0 0 ...
##  $ LowQualFinSF : num [1:1459] 0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : num [1:1459] 896 1329 1629 1604 1280 ...
##  $ BsmtFullBath : num [1:1459] 0 0 0 0 0 0 1 0 1 1 ...
##  $ BsmtHalfBath : num [1:1459] 0 0 0 0 0 0 0 0 0 0 ...
##  $ FullBath     : num [1:1459] 1 1 2 2 2 2 2 2 1 1 ...
##  $ HalfBath     : num [1:1459] 0 1 1 1 0 1 0 1 1 0 ...
##  $ BedroomAbvGr : num [1:1459] 2 3 3 3 2 3 3 3 2 2 ...
##  $ KitchenAbvGr : num [1:1459] 1 1 1 1 1 1 1 1 1 1 ...
##  $ KitchenQual  : chr [1:1459] "TA" "Gd" "TA" "Gd" ...
##  $ TotRmsAbvGrd : num [1:1459] 5 6 6 7 5 7 6 7 5 4 ...
##  $ Functional   : chr [1:1459] "Typ" "Typ" "Typ" "Typ" ...
```

```
##  $ Fireplaces   : num [1:1459] 0 0 1 1 0 1 0 1 1 0 ...
##  $ FireplaceQu  : chr [1:1459] NA NA "TA" "Gd" ...
##  $ GarageType   : chr [1:1459] "Attchd" "Attchd" "Attchd" "Attchd" ...
##  $ GarageYrBlt  : num [1:1459] 1961 1958 1997 1998 1992 ...
##  $ GarageFinish : chr [1:1459] "Unf" "Unf" "Fin" "Fin" ...
##  $ GarageCars   : num [1:1459] 1 1 2 2 2 2 2 2 2 2 ...
##  $ GarageArea   : num [1:1459] 730 312 482 470 506 440 420 393 506 525 ...
##  $ GarageQual   : chr [1:1459] "TA" "TA" "TA" "TA" ...
##  $ GarageCond   : chr [1:1459] "TA" "TA" "TA" "TA" ...
##  $ PavedDrive   : chr [1:1459] "Y" "Y" "Y" "Y" ...
##  $ WoodDeckSF   : num [1:1459] 140 393 212 360 0 157 483 0 192 240 ...
##  $ OpenPorchSF  : num [1:1459] 0 36 34 36 82 84 21 75 0 0 ...
##  $ EnclosedPorch: num [1:1459] 0 0 0 0 0 0 0 0 0 0 ...
##  $ 3SsnPorch    : num [1:1459] 0 0 0 0 0 0 0 0 0 0 ...
##  $ ScreenPorch  : num [1:1459] 120 0 0 0 144 0 0 0 0 0 ...
##  $ PoolArea     : num [1:1459] 0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : chr [1:1459] NA NA NA NA ...
##  $ Fence        : chr [1:1459] "MnPrv" NA "MnPrv" NA ...
##  $ MiscFeature  : chr [1:1459] NA "Gar2" NA NA ...
##  $ MiscVal      : num [1:1459] 0 12500 0 0 0 0 500 0 0 0 ...
##  $ MoSold       : num [1:1459] 6 6 3 6 1 4 3 5 2 4 ...
##  $ YrSold       : num [1:1459] 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ SaleType     : chr [1:1459] "WD" "WD" "WD" "WD" ...
##  $ SaleCondition: chr [1:1459] "Normal" "Normal" "Normal" "Normal" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   MSSubClass = col_double(),
##   ..   MSZoning = col_character(),
##   ..   LotFrontage = col_double(),
##   ..   LotArea = col_double(),
##   ..   Street = col_character(),
##   ..   Alley = col_character(),
##   ..   LotShape = col_character(),
##   ..   LandContour = col_character(),
##   ..   Utilities = col_character(),
##   ..   LotConfig = col_character(),
##   ..   LandSlope = col_character(),
##   ..   Neighborhood = col_character(),
##   ..   Condition1 = col_character(),
##   ..   Condition2 = col_character(),
##   ..   BldgType = col_character(),
##   ..   HouseStyle = col_character(),
##   ..   OverallQual = col_double(),
##   ..   OverallCond = col_double(),
##   ..   YearBuilt = col_double(),
##   ..   YearRemodAdd = col_double(),
##   ..   RoofStyle = col_character(),
##   ..   RoofMatl = col_character(),
##   ..   Exterior1st = col_character(),
##   ..   Exterior2nd = col_character(),
##   ..   MasVnrType = col_character(),
##   ..   MasVnrArea = col_double(),
##   ..   ExterQual = col_character(),
```

```
##   ..   ExterCond = col_character(),
##   ..   Foundation = col_character(),
##   ..   BsmtQual = col_character(),
##   ..   BsmtCond = col_character(),
##   ..   BsmtExposure = col_character(),
##   ..   BsmtFinType1 = col_character(),
##   ..   BsmtFinSF1 = col_double(),
##   ..   BsmtFinType2 = col_character(),
##   ..   BsmtFinSF2 = col_double(),
##   ..   BsmtUnfSF = col_double(),
##   ..   TotalBsmtSF = col_double(),
##   ..   Heating = col_character(),
##   ..   HeatingQC = col_character(),
##   ..   CentralAir = col_character(),
##   ..   Electrical = col_character(),
##   ..   `1stFlrSF` = col_double(),
##   ..   `2ndFlrSF` = col_double(),
##   ..   LowQualFinSF = col_double(),
##   ..   GrLivArea = col_double(),
##   ..   BsmtFullBath = col_double(),
##   ..   BsmtHalfBath = col_double(),
##   ..   FullBath = col_double(),
##   ..   HalfBath = col_double(),
##   ..   BedroomAbvGr = col_double(),
##   ..   KitchenAbvGr = col_double(),
##   ..   KitchenQual = col_character(),
##   ..   TotRmsAbvGrd = col_double(),
##   ..   Functional = col_character(),
##   ..   Fireplaces = col_double(),
##   ..   FireplaceQu = col_character(),
##   ..   GarageType = col_character(),
##   ..   GarageYrBlt = col_double(),
##   ..   GarageFinish = col_character(),
##   ..   GarageCars = col_double(),
##   ..   GarageArea = col_double(),
##   ..   GarageQual = col_character(),
##   ..   GarageCond = col_character(),
##   ..   PavedDrive = col_character(),
##   ..   WoodDeckSF = col_double(),
##   ..   OpenPorchSF = col_double(),
##   ..   EnclosedPorch = col_double(),
##   ..   `3SsnPorch` = col_double(),
##   ..   ScreenPorch = col_double(),
##   ..   PoolArea = col_double(),
##   ..   PoolQC = col_character(),
##   ..   Fence = col_character(),
##   ..   MiscFeature = col_character(),
##   ..   MiscVal = col_double(),
##   ..   MoSold = col_double(),
##   ..   YrSold = col_double(),
##   ..   SaleType = col_character(),
##   ..   SaleCondition = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

2. Elabore un árbol de regresión para predecir el precio de las casas usando todas las variables.

```r
# Cargar librerías necesarias
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.4.3
```

```r
library(caret)
```

```
## Cargando paquete requerido: ggplot2
```

```
## Cargando paquete requerido: lattice
```

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

# Cargar conjunto de datos
train_set <- read.csv("house_prices_data/train.csv", stringsAsFactors = TRUE)

# Revisar estructura y resumen de los datos
str(train_set)
```

```
## 'data.frame':    1460 obs. of  81 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning     : Factor w/ 5 levels "C (all)","FV",..: 4 4 4 4 4 4 4 4 5 4 ...
##  $ LotFrontage  : int  65 80 68 60 84 85 75 NA 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
##  $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Alley        : Factor w/ 2 levels "Grvl","Pave": NA NA NA NA NA NA NA NA NA NA ...
##  $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",..: 4 4 1 1 1 1 4 1 4 4 ...
##  $ LandContour  : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Utilities    : Factor w/ 2 levels "AllPub","NoSeWa": 1 1 1 1 1 1 1 1 1 1 ...
##  $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5 1 5 1 ...
##  $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",..: 6 25 6 7 14 12 21 17 18 4 ...
##  $ Condition1   : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 5 1 1 ...
##  $ Condition2   : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 1 ...
##  $ BldgType     : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 1 1 1 2 ...
##  $ HouseStyle   : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 6 3 6 6 6 1 3 6 1 2 ...
##  $ OverallQual  : int  7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond  : int  5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
##  $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
##  $ RoofStyle    : Factor w/ 6 levels "Flat","Gable",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ RoofMatl     : Factor w/ 8 levels "ClyTile","CompShg",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Exterior1st  : Factor w/ 15 levels "AsbShng","AsphShn",..: 13 9 13 14 13 13 13 7 4 9 ...
##  $ Exterior2nd  : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 16 14 14 14 7 16 9 ...
##  $ MasVnrType   : Factor w/ 4 levels "BrkCmn","BrkFace",..: 2 3 2 3 2 3 4 4 3 3 ...
```

```
##  $ MasVnrArea   : int  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 4 3 4 3 4 4 4 ...
##  $ ExterCond    : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ Foundation   : Factor w/ 6 levels "BrkTil","CBlock",..: 3 2 3 1 3 6 3 2 1 1 ...
##  $ BsmtQual     : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 3 3 4 3 3 1 3 4 4 ...
##  $ BsmtCond     : Factor w/ 4 levels "Fa","Gd","Po",..: 4 4 4 2 4 4 4 4 4 4 ...
##  $ BsmtExposure : Factor w/ 4 levels "Av","Gd","Mn",..: 4 2 3 4 1 4 1 3 4 4 ...
##  $ BsmtFinType1 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 3 1 3 1 3 3 3 1 6 3 ...
##  $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
##  $ BsmtFinType2 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 6 6 6 6 6 6 6 2 6 6 ...
##  $ BsmtFinSF2   : int  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : int  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
##  $ Heating      : Factor w/ 6 levels "Floor","GasA",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ HeatingQC    : Factor w/ 5 levels "Ex","Fa","Gd",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ CentralAir   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical   : Factor w/ 5 levels "FuseA","FuseF",..: 5 5 5 5 5 5 5 5 2 5 ...
##  $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
##  $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
##  $ BsmtFullBath : int  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath     : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual  : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 3 3 4 3 4 4 4 ...
##  $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : Factor w/ 7 levels "Maj1","Maj2",..: 7 7 7 7 7 7 7 7 3 7 ...
##  $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu  : Factor w/ 5 levels "Ex","Fa","Gd",..: NA 5 5 3 5 NA 3 5 5 5 ...
##  $ GarageType   : Factor w/ 6 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2 6 2 ...
##  $ GarageYrBlt  : int  2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...
##  $ GarageFinish : Factor w/ 3 levels "Fin","RFn","Unf": 2 2 2 3 2 3 2 2 3 2 ...
##  $ GarageCars   : int  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : int  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 2 3 ...
##  $ GarageCond   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ PavedDrive   : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
##  $ WoodDeckSF   : int  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF  : int  61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch: int  0 0 0 272 0 0 0 228 205 0 ...
##  $ X3SsnPorch   : int  0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : Factor w/ 3 levels "Ex","Fa","Gd": NA NA NA NA NA NA NA NA NA NA ...
##  $ Fence        : Factor w/ 4 levels "GdPrv","GdWo",..: NA NA NA NA NA 3 NA NA NA NA ...
##  $ MiscFeature  : Factor w/ 4 levels "Gar2","Othr",..: NA NA NA NA NA 3 NA 3 NA NA ...
##  $ MiscVal      : int  0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold       : int  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold       : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
##  $ SaleType     : Factor w/ 9 levels "COD","Con","ConLD",..: 9 9 9 9 9 9 9 9 9 9 ...
##  $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 1 5 5 5 5 1 5 ...
```

```
##  $ SalePrice    : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
```

```
summary(train_set)
```

```
##        Id           MSSubClass       MSZoning     LotFrontage
##  Min.   :   1.0   Min.   : 20.0   C (all):  10   Min.   : 21.00
##  1st Qu.: 365.8   1st Qu.: 20.0   FV     :  65   1st Qu.: 59.00
##  Median : 730.5   Median : 50.0   RH     :  16   Median : 69.00
##  Mean   : 730.5   Mean   : 56.9   RL     :1151   Mean   : 70.05
##  3rd Qu.:1095.2   3rd Qu.: 70.0   RM     : 218   3rd Qu.: 80.00
##  Max.   :1460.0   Max.   :190.0                   Max.   :313.00
##                                                  NA's   :259
##     LotArea        Street      Alley      LotShape  LandContour  Utilities
##  Min.   :  1300   Grvl:   6   Grvl:  50   IR1:484   Bnk:  63    AllPub:1459
##  1st Qu.:  7554   Pave:1454   Pave:  41   IR2: 41   HLS:  50    NoSeWa:   1
##  Median :  9478               NA's:1369   IR3: 10   Low:  36
##  Mean   : 10517                           Reg:925   Lvl:1311
##  3rd Qu.: 11602
##  Max.   :215245
##
##    LotConfig    LandSlope   Neighborhood   Condition1     Condition2
##  Corner : 263   Gtl:1382   NAmes  :225   Norm   :1260   Norm   :1445
##  CulDSac:  94   Mod:  65   CollgCr:150   Feedr  :  81   Feedr  :   6
##  FR2    :  47   Sev:  13   OldTown:113   Artery :  48   Artery :   2
##  FR3    :   4              Edwards:100   RRAn   :  26   PosN   :   2
##  Inside :1052              Somerst: 86   PosN   :  19   RRNn   :   2
##                            Gilbert: 79   RRAe   :  11   PosA   :   1
##                            (Other):707   (Other):  15   (Other):   2
##    BldgType      HouseStyle   OverallQual      OverallCond      YearBuilt
##  1Fam  :1220   1Story :726   Min.   : 1.000   Min.   :1.000   Min.   :1872
##  2fmCon:  31   2Story :445   1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954
##  Duplex:  52   1.5Fin :154   Median : 6.000   Median :5.000   Median :1973
##  Twnhs :  43   SLvl   : 65   Mean   : 6.099   Mean   :5.575   Mean   :1971
##  TwnhsE: 114   SFoyer : 37   3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2000
##                1.5Unf : 14   Max.   :10.000   Max.   :9.000   Max.   :2010
##                (Other): 19
##   YearRemodAdd    RoofStyle       RoofMatl     Exterior1st   Exterior2nd
##  Min.   :1950   Flat   :  13   CompShg:1434   VinylSd:515   VinylSd:504
##  1st Qu.:1967   Gable  :1141   Tar&Grv:  11   HdBoard:222   MetalSd:214
##  Median :1994   Gambrel:  11   WdShngl:   6   MetalSd:220   HdBoard:207
##  Mean   :1985   Hip    : 286   WdShake:   5   Wd Sdng:206   Wd Sdng:197
##  3rd Qu.:2004   Mansard:   7   ClyTile:   1   Plywood:108   Plywood:142
##  Max.   :2010   Shed   :   2   Membran:   1   CemntBd: 61   CmentBd: 60
##                                (Other):   2   (Other):128   (Other):136
##    MasVnrType    MasVnrArea     ExterQual ExterCond  Foundation  BsmtQual
##  BrkCmn : 15   Min.   :   0.0   Ex: 52    Ex:   3   BrkTil:146   Ex  :121
##  BrkFace:445   1st Qu.:   0.0   Fa: 14    Fa:  28   CBlock:634   Fa  : 35
##  None   :864   Median :   0.0   Gd:488    Gd: 146   PConc :647   Gd  :618
##  Stone  :128   Mean   : 103.7   TA:906    Po:   1   Slab  : 24   TA  :649
##  NA's   :  8   3rd Qu.: 166.0             TA:1282   Stone :  6   NA's: 37
##                Max.   :1600.0                       Wood  :  3
##                NA's   :8
##  BsmtCond    BsmtExposure BsmtFinType1   BsmtFinSF1     BsmtFinType2
```

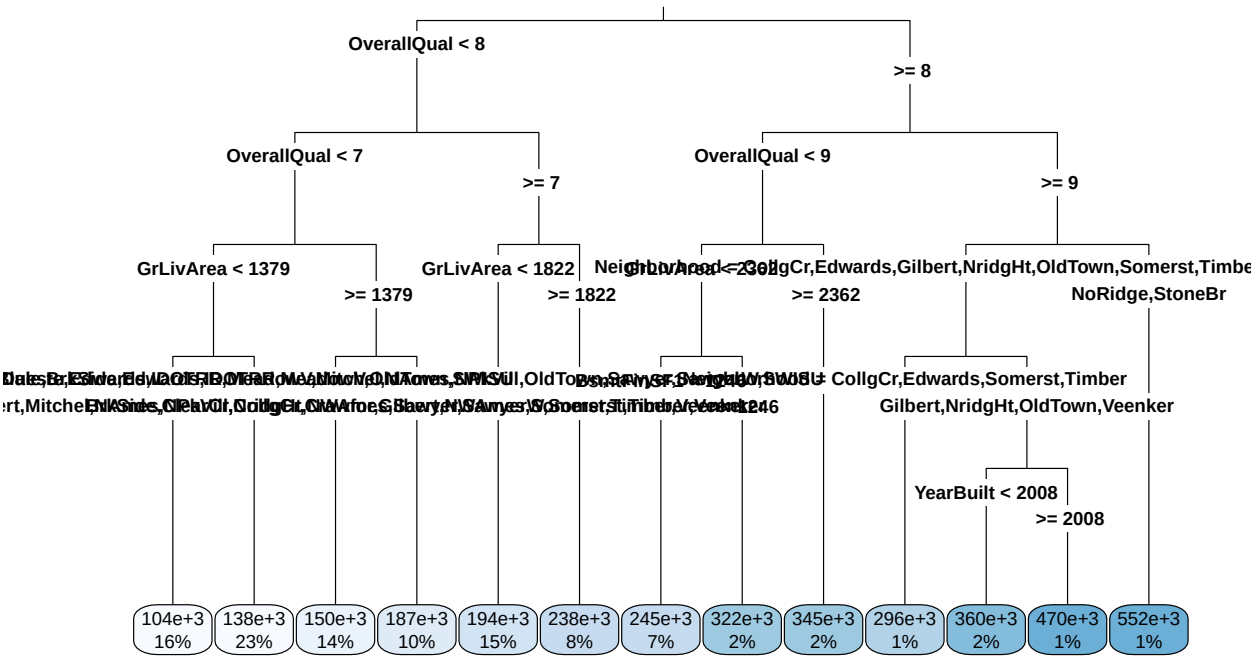
```
##  Fa  :  45   Av  :221     ALQ :220     Min.   :   0.0   ALQ :  19
##  Gd  :  65   Gd  :134     BLQ :148     1st Qu.:   0.0   BLQ :  33
##  Po  :   2   Mn  :114     GLQ :418     Median : 383.5   GLQ :  14
##  TA  :1311   No  :953     LwQ : 74     Mean   : 443.6   LwQ :  46
##  NA's:  37   NA's: 38     Rec :133     3rd Qu.: 712.2   Rec :  54
##                           Unf :430     Max.   :5644.0   Unf :1256
##                           NA's: 37                      NA's:  38
##    BsmtFinSF2        BsmtUnfSF       TotalBsmtSF      Heating     HeatingQC
##  Min.   :   0.00   Min.   :   0.0   Min.   :   0.0   Floor:   1   Ex:741
##  1st Qu.:   0.00   1st Qu.: 223.0   1st Qu.: 795.8   GasA :1428   Fa: 49
##  Median :   0.00   Median : 477.5   Median : 991.5   GasW :  18   Gd:241
##  Mean   :  46.55   Mean   : 567.2   Mean   :1057.4   Grav :   7   Po:  1
##  3rd Qu.:   0.00   3rd Qu.: 808.0   3rd Qu.:1298.2   OthW :   2   TA:428
##  Max.   :1474.00   Max.   :2336.0   Max.   :6110.0   Wall :   4
##
##  CentralAir Electrical     X1stFlrSF      X2ndFlrSF     LowQualFinSF
##  N:  95     FuseA:  94   Min.   : 334   Min.   :   0   Min.   :  0.000
##  Y:1365     FuseF:  27   1st Qu.: 882   1st Qu.:   0   1st Qu.:  0.000
##             FuseP:   3   Median :1087   Median :   0   Median :  0.000
##             Mix  :   1   Mean   :1163   Mean   : 347   Mean   :  5.845
##             SBrkr:1334   3rd Qu.:1391   3rd Qu.: 728   3rd Qu.:  0.000
##             NA's :   1   Max.   :4692   Max.   :2065   Max.   :572.000
##
##    GrLivArea     BsmtFullBath     BsmtHalfBath        FullBath
##  Min.   : 334   Min.   :0.0000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:1130   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:1.000
##  Median :1464   Median :0.0000   Median :0.00000   Median :2.000
##  Mean   :1515   Mean   :0.4253   Mean   :0.05753   Mean   :1.565
##  3rd Qu.:1777   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:2.000
##  Max.   :5642   Max.   :3.0000   Max.   :2.00000   Max.   :3.000
##
##     HalfBath       BedroomAbvGr    KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Ex:100      Min.   : 2.000
##  1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:1.000   Fa: 39      1st Qu.: 5.000
##  Median :0.0000   Median :3.000   Median :1.000   Gd:586      Median : 6.000
##  Mean   :0.3829   Mean   :2.866   Mean   :1.047   TA:735      Mean   : 6.518
##  3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:1.000               3rd Qu.: 7.000
##  Max.   :2.0000   Max.   :8.000   Max.   :3.000               Max.   :14.000
##
##  Functional    Fireplaces    FireplaceQu   GarageType   GarageYrBlt
##  Maj1:  14   Min.   :0.000   Ex  : 24    2Types :  6   Min.   :1900
##  Maj2:   5   1st Qu.:0.000   Fa  : 33    Attchd :870   1st Qu.:1961
##  Min1:  31   Median :1.000   Gd  :380    Basment: 19   Median :1980
##  Min2:  34   Mean   :0.613   Po  : 20    BuiltIn: 88   Mean   :1979
##  Mod :  15   3rd Qu.:1.000   TA  :313    CarPort:  9   3rd Qu.:2002
##  Sev :   1   Max.   :3.000   NA's:690    Detchd :387   Max.   :2010
##  Typ :1360                               NA's   : 81   NA's   :81
##  GarageFinish   GarageCars      GarageArea     GarageQual  GarageCond
##  Fin :352     Min.   :0.000   Min.   :   0.0   Ex  :   3   Ex  :   2
##  RFn :422     1st Qu.:1.000   1st Qu.: 334.5   Fa  :  48   Fa  :  35
##  Unf :605     Median :2.000   Median : 480.0   Gd  :  14   Gd  :   9
##  NA's: 81     Mean   :1.767   Mean   : 473.0   Po  :   3   Po  :   7
##               3rd Qu.:2.000   3rd Qu.: 576.0   TA  :1311   TA  :1326
##               Max.   :4.000   Max.   :1418.0   NA's:  81   NA's:  81
```

```
##
##  PavedDrive   WoodDeckSF      OpenPorchSF     EnclosedPorch      X3SsnPorch
##  N:  90     Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  P:  30     1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00
##  Y:1340     Median :  0.00   Median : 25.00   Median :  0.00   Median :  0.00
##             Mean   : 94.24   Mean   : 46.66   Mean   : 21.95   Mean   :  3.41
##             3rd Qu.:168.00   3rd Qu.: 68.00   3rd Qu.:  0.00   3rd Qu.:  0.00
##             Max.   :857.00   Max.   :547.00   Max.   :552.00   Max.   :508.00
##
##   ScreenPorch        PoolArea        PoolQC       Fence      MiscFeature
##  Min.   :  0.00   Min.   :  0.000   Ex   :   2   GdPrv:  59   Gar2:   2
##  1st Qu.:  0.00   1st Qu.:  0.000   Fa   :   2   GdWo :  54   Othr:   2
##  Median :  0.00   Median :  0.000   Gd   :   3   MnPrv: 157   Shed:  49
##  Mean   : 15.06   Mean   :  2.759   NA's:1453   MnWw :  11   TenC:   1
##  3rd Qu.:  0.00   3rd Qu.:  0.000               NA's :1179   NA's:1406
##  Max.   :480.00   Max.   :738.000
##
##     MiscVal             MoSold           YrSold        SaleType
##  Min.   :    0.00   Min.   : 1.000   Min.   :2006   WD     :1267
##  1st Qu.:    0.00   1st Qu.: 5.000   1st Qu.:2007   New    : 122
##  Median :    0.00   Median : 6.000   Median :2008   COD    :  43
##  Mean   :   43.49   Mean   : 6.322   Mean   :2008   ConLD  :   9
##  3rd Qu.:    0.00   3rd Qu.: 8.000   3rd Qu.:2009   ConLI  :   5
##  Max.   :15500.00   Max.   :12.000   Max.   :2010   ConLw  :   5
##                                                     (Other):   9
##  SaleCondition    SalePrice
##  Abnorml: 101   Min.   : 34900
##  AdjLand:   4   1st Qu.:129975
##  Alloca :  12   Median :163000
##  Family :  20   Mean   :180921
##  Normal :1198   3rd Qu.:214000
##  Partial: 125   Max.   :755000
##
```

```r
# Eliminar la columna Id (no es una variable predictora)
if ("Id" %in% colnames(train_set)) {
  train_set <- train_set %>% select(-Id)
}

# Convertir variables categóricas a factores si es necesario
categorical_vars <- names(train_set)[sapply(train_set, is.character)]
train_set[categorical_vars] <- lapply(train_set[categorical_vars], as.factor)

# Dividir `train_set` en un conjunto de entrenamiento (80%) y prueba (20%)
set.seed(42) # Para reproducibilidad
train_index <- createDataPartition(train_set$SalePrice, p = 0.8, list = FALSE)
train_data <- train_set[train_index, ]
test_data <- train_set[-train_index, ]

# Crear el árbol de regresión con el nuevo conjunto de entrenamiento
set.seed(42)
arbol_regresion <- rpart(SalePrice ~ ., data = train_data, method = "anova")

# Visualizar el árbol de regresión
```

```
rpart.plot(arbol_regresion, type = 3, fallen.leaves = TRUE, cex = 0.6, main = "Árbol de
 ↪  Regresión para Predicción de Precio de Casas")
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```




```
# Evaluación del modelo
predicciones_train <- predict(arbol_regresion, newdata = train_data)
mse_train <- mean((train_data$SalePrice - predicciones_train)^2, na.rm = TRUE)
cat("Error cuadrático medio en entrenamiento (MSE):", mse_train, "\n")
```

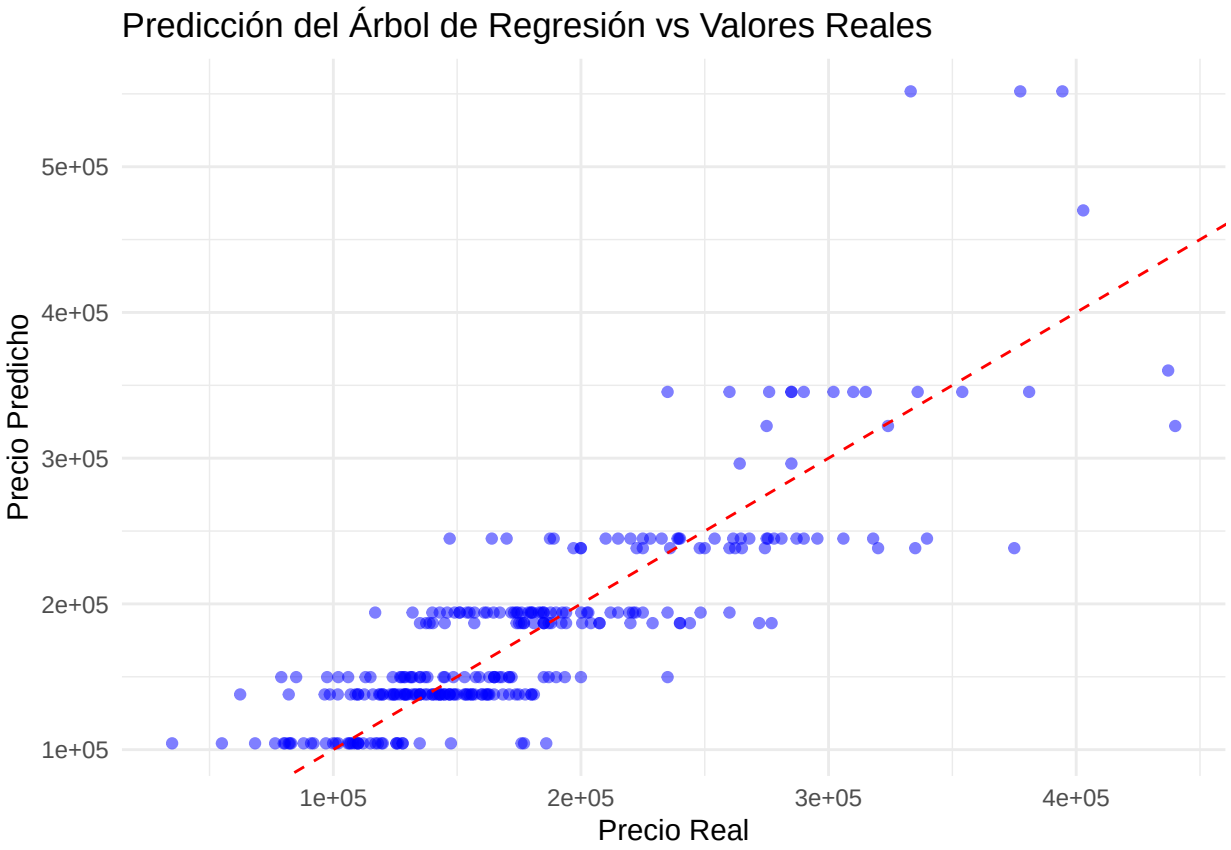
```
## Error cuadrático medio en entrenamiento (MSE): 1297581895
```

```
predicciones_test <- predict(arbol_regresion, newdata = test_data)
mse_test <- mean((test_data$SalePrice - predicciones_test)^2, na.rm = TRUE)
cat("Error cuadrático medio en prueba (MSE):", mse_test, "\n")
```

```
## Error cuadrático medio en prueba (MSE): 1658823049
```

```
# Comparar predicciones con valores reales
ggplot(data.frame(Real = test_data$SalePrice, Predicho = predicciones_test), aes(x =
 ↪  Real, y = Predicho)) +
  geom_point(alpha = 0.5, color = "blue") +
```

```
geom_abline(slope = 1, intercept = 0, col = "red", linetype = "dashed") +
labs(title = "Predicción del Árbol de Regresión vs Valores Reales",
     x = "Precio Real",
     y = "Precio Predicho") +
theme_minimal()
```



Análisis del Árbol de Regresión y Resultados

2.1. Interpretación del Árbol de Regresión

El árbol de regresión construido tiene como objetivo predecir el precio de las casas utilizando todas las variables disponibles en el conjunto de datos. La estructura del árbol nos permite identificar qué variables tienen mayor influencia en la predicción del precio y cómo se segmentan los diferentes valores de las propiedades. A partir de la imagen del árbol, se pueden extraer varios hallazgos clave:

2.1.1. División Principal:

- La primera división del árbol está determinada por la variable OverallQual (calidad general de la casa). Esto indica que la calidad de construcción y los materiales utilizados son el principal factor que influye en el precio de las casas.
- **Si OverallQual $<$ 8**, el árbol sigue subdividiendo por variables como GrLivArea (área habitable en pies cuadrados), Neighborhood (vecindario) y BsmtFinSF1 (área terminada del sótano). Esto sugiere que para casas de calidad media o baja, el precio está más condicionado por el tamaño de la casa y su ubicación.
- **Si OverallQual 8**, la predicción del precio se guía por GrLivArea y YearBuilt (año de construcción). Esto significa que para casas de mayor calidad, las dimensiones de la propiedad y el año de construcción juegan un papel fundamental en la determinación del precio.

2.2. Segmentación por Tamaño y Ubicación:

- **Para casas con OverallQual < 7**, el precio tiende a ser más bajo y se ve influenciado principalmente por GrLivArea y Neighborhood. Esto sugiere que, en propiedades de menor calidad, la ubicación y el tamaño son factores determinantes en la variación del precio.
- **Para casas con OverallQual 9**, el precio tiende a ser significativamente más alto, y el árbol segmenta aún más las predicciones basándose en GrLivArea y Neighborhood. Esto implica que en propiedades de lujo, el tamaño de la casa y su ubicación en vecindarios de prestigio juegan un papel clave en la valorización de la propiedad.
- *Vecindarios como NoRidge y StoneBr aparecen en la parte superior de la jerarquía de segmentación para las casas más costosas, lo que indica que las propiedades en estas zonas tienden a tener precios más elevados.

2.3. Análisis del Desempeño del Modelo

El rendimiento del modelo se evalúa mediante el error cuadrático medio (MSE), el cual nos indica qué tan lejos están las predicciones del modelo en relación con los valores reales de las casas.

- **MSE en entrenamiento**: 1,297,581,895
- **MSE en prueba**: 1,658,823,049

Un aspecto importante a notar es que el MSE en prueba es mayor que el MSE en entrenamiento. Esto sugiere que el modelo tiene una alta varianza, lo que significa que puede estar sobreajustado a los datos de entrenamiento. Un modelo sobreajustado tiende a aprender demasiado bien los patrones del conjunto de entrenamiento, pero pierde capacidad de generalización cuando se le presentan nuevos datos en el conjunto de prueba.

La diferencia entre los errores nos indica que el modelo podría estar capturando demasiado ruido en el entrenamiento, lo cual podría corregirse mediante técnicas como la poda del árbol o el ajuste de hiperparámetros para reducir la complejidad del modelo.

2.4. Evaluación de Predicciones vs Valores Reales

En la segunda imagen proporcionada, se muestra un gráfico de dispersión donde el eje X representa los valores reales del precio de las casas y el eje Y representa los valores predichos por el modelo.

- **Línea roja diagonal**: Representa la relación ideal entre las predicciones y los valores reales. Si todas las predicciones fueran perfectas, los puntos estarían alineados sobre esta línea.
- **Puntos azules dispersos**: Indican cómo se distribuyen las predicciones del modelo en relación con los valores reales.

A partir de la gráfica podemos notar:

2.4.1. **Buena predicción en el rango medio de precios**: Para casas con precios dentro de un rango medio, el modelo parece tener una precisión aceptable, ya que varios puntos están cercanos a la línea roja.

2.4.2. **Dificultad con valores extremos**: Para casas con precios extremadamente altos o bajos, el modelo tiene una mayor dispersión, lo que indica que sus predicciones son menos precisas. Esto es un indicio de que el modelo no está capturando completamente la variabilidad de los precios en los extremos.

2.4.3. **Subestimación de precios altos**: Algunas casas de precios elevados aparecen con valores predichos mucho más bajos de lo esperado. Esto sugiere que el modelo tiene dificultades para capturar correctamente los factores que hacen que una casa tenga un precio significativamente alto.

2.4.4. **Sobreestimación en algunos casos**: También hay algunos casos en los que el modelo sobreestima el precio de casas de menor valor, indicando que podría estar considerando características menos relevantes como altamente influyentes.

2.5. Conclusiones y Recomendaciones

El modelo identifica correctamente los principales factores que afectan el precio de una casa:

2.5.1.**OverallQual (calidad de la casa) es la variable más relevante**.

- GrLivArea (tamaño habitable) y Neighborhood (vecindario) también tienen un impacto clave.
- YearBuilt (año de construcción) influye más en casas de mayor calidad.

2.5.2. **El modelo tiene un MSE relativamente alto, lo que indica margen de mejora**:

- La diferencia entre el MSE en entrenamiento y en prueba sugiere sobreajuste.
- Se podría mejorar utilizando técnicas como poda del árbol, validación cruzada o ajuste de hiperparámetros para reducir la varianza.

2.5.3. **El modelo funciona bien para precios en el rango medio, pero tiene problemas con valores extremos**:

- Predice bien precios intermedios, pero subestima valores altos y tiene errores en casos de precios muy bajos.
- Se podría explorar la inclusión de interacciones entre variables o utilizar modelos más avanzados como Random Forest o Gradient Boosting para mejorar la precisión.

2.5.4. **El modelo es interpretable y proporciona información útil sobre la influencia de diferentes características en el precio de las casas**:

- A diferencia de modelos más complejos como redes neuronales o ensembles, los árboles de decisión permiten una visualización clara de cómo se toman las decisiones de predicción.
- Sin embargo, esta ventaja de interpretabilidad viene a costa de menor precisión en las predicciones.

Recomendaciones para Mejorar el Modelo

- Para mejorar el desempeño del modelo y obtener predicciones más precisas, se pueden considerar los siguientes enfoques:
- **Poda del árbol**: Reducir el tamaño del árbol eliminando divisiones poco significativas para evitar sobreajuste.
- **Uso de Random Forest o Gradient Boosting**: Árboles de regresión individuales pueden ser inestables, mientras que métodos basados en ensambles pueden mejorar la precisión al combinar múltiples árboles.
- **Feature Engineering**: Investigar si transformar algunas variables o agregar nuevas combinaciones de variables podría mejorar las predicciones.
- **Validación Cruzada**: Evaluar el modelo con validación cruzada para asegurarse de que su rendimiento es consistente en diferentes subconjuntos de datos.

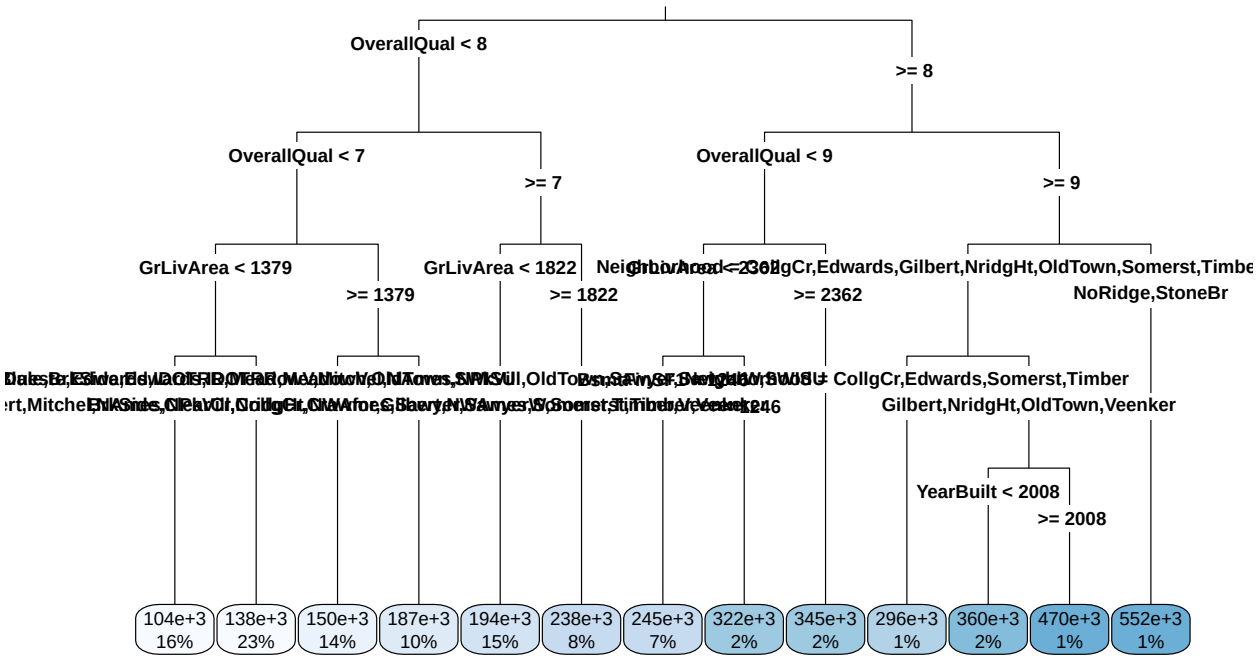
Conclusión Final

El árbol de regresión construido proporciona una buena interpretación de los factores clave que afectan el precio de las casas. Sin embargo, su precisión es limitada, especialmente en valores extremos. El modelo tiene un nivel de error considerable, indicando que aunque captura bien algunas tendencias generales, no generaliza de manera óptima para todas las observaciones. Mejoras adicionales pueden hacerse mediante poda del árbol o exploración de modelos más sofisticados como Random Forest o Gradient Boosting.

3. Úselo para predecir y analice el resultado. ¿Qué tal lo hizo?

```r
# Cargar librerías necesarias
library(rpart)
library(rpart.plot)
library(caret)
library(dplyr)
library(ggplot2)

# Cargar conjunto de datos
train_set <- read.csv("house_prices_data/train.csv", stringsAsFactors = TRUE)

# Eliminar la columna de Id (ya que no aporta a la predicción)
if ("Id" %in% colnames(train_set)) {
  train_set <- train_set %>% select(-Id)
}

# Convertir variables categóricas a factores si es necesario
categorical_vars <- names(train_set)[sapply(train_set, is.character)]
train_set[categorical_vars] <- lapply(train_set[categorical_vars], as.factor)

# Dividir el conjunto de entrenamiento en training (80%) y test (20%)
set.seed(42)
train_index <- createDataPartition(train_set$SalePrice, p = 0.8, list = FALSE)
train_data <- train_set[train_index, ]
test_data <- train_set[-train_index, ]

# Crear el modelo de árbol de regresión
set.seed(42)
arbol_regresion <- rpart(SalePrice ~ ., data = train_data, method = "anova")

# Visualizar el árbol de decisión
rpart.plot(arbol_regresion, type = 3, fallen.leaves = TRUE, cex = 0.6, main = "Árbol de
 ↪  Regresión para Predicción de Precio de Casas")
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



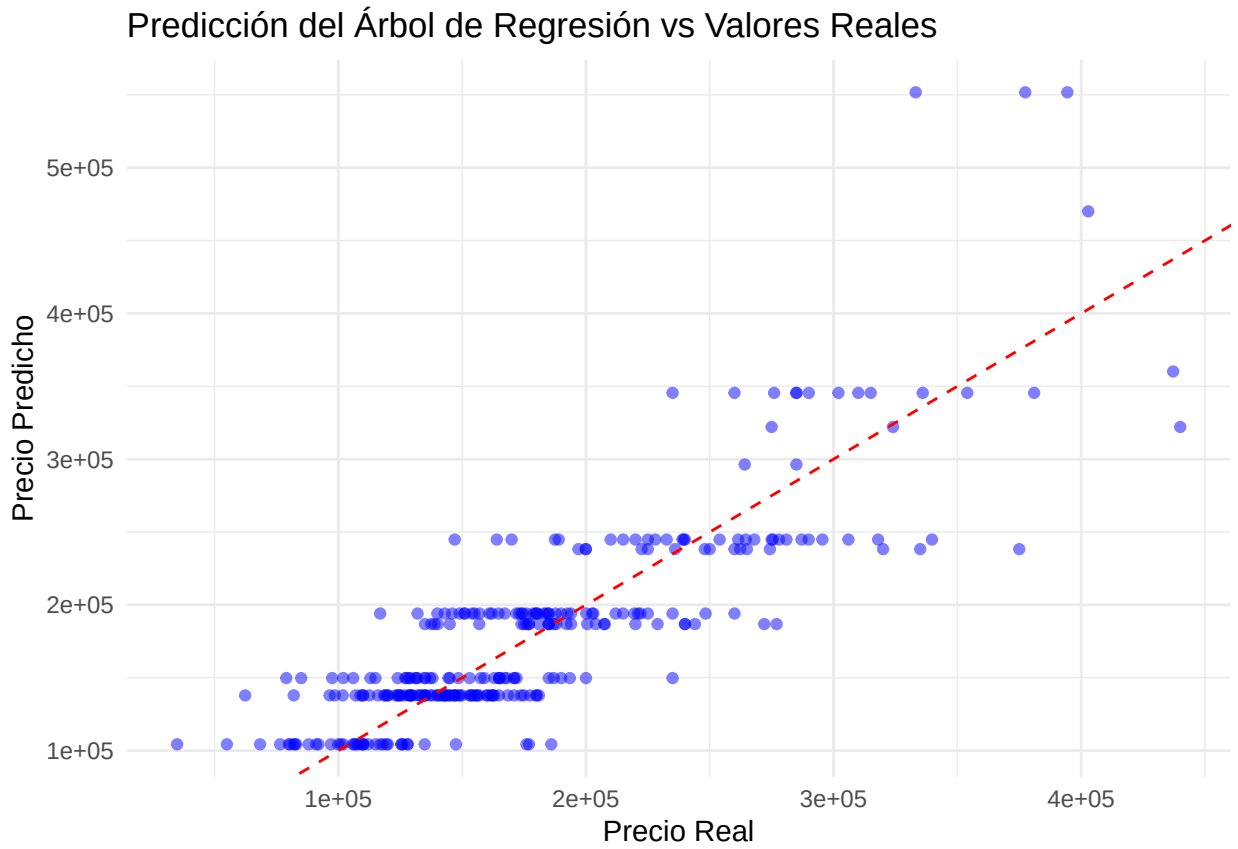

```
# Evaluación del modelo en entrenamiento
predicciones_train <- predict(arbol_regresion, newdata = train_data)
mse_train <- mean((train_data$SalePrice - predicciones_train)^2, na.rm = TRUE)
cat("Error cuadrático medio en entrenamiento (MSE):", mse_train, "\n")
```

```
## Error cuadrático medio en entrenamiento (MSE): 1297581895
```

```
# Evaluación del modelo en prueba
predicciones_test <- predict(arbol_regresion, newdata = test_data)
mse_test <- mean((test_data$SalePrice - predicciones_test)^2, na.rm = TRUE)
cat("Error cuadrático medio en prueba (MSE):", mse_test, "\n")
```

```
## Error cuadrático medio en prueba (MSE): 1658823049
```

```
# Comparar predicciones con valores reales
ggplot(data.frame(Real = test_data$SalePrice, Predicho = predicciones_test), aes(x =
↪  Real, y = Predicho)) +
  geom_point(alpha = 0.5, color = "blue") +
  geom_abline(slope = 1, intercept = 0, col = "red", linetype = "dashed") +
  labs(title = "Predicción del Árbol de Regresión vs Valores Reales",
       x = "Precio Real",
       y = "Precio Predicho") +
  theme_minimal()
```



Análisis y Evaluación del Modelo de Árbol de Regresión

3.1.Evaluación del Desempeño con el Error Cuadrático Medio (MSE)

MSE en entrenamiento: 1,297,581,895 **MSE en prueba**: 1,658,823,049

El error cuadrático medio en prueba es considerablemente mayor que en entrenamiento, lo que indica que el modelo podría estar sobreajustado a los datos de entrenamiento y no generaliza bien a nuevos datos.

3.2. Interpretación del Árbol de Regresión

El árbol de regresión muestra que las variables más importantes para la predicción del precio de las casas son:

- **OverallQual (Calidad general de la casa)**: Es la primera división del árbol, lo que indica que es el factor más influyente.
- **GrLivArea (Área habitable en pies cuadrados por encima del suelo)**: También tiene un impacto significativo en la segmentación.
- **Neighborhood (Barrio en el que se encuentra la casa)**: Determina la segmentación en varias ramas del árbol.
- **BsmtFinSF1 (Área del sótano terminado) y YearBuilt (Año de construcción de la casa)**: También son importantes en las ramas finales del árbol.

Estos criterios de división nos dicen que el precio de una casa se ve afectado en gran medida por la calidad de construcción, el tamaño habitable y la ubicación.

3.3. Interpretación del Gráfico de Predicciones vs Valores Reales

3.3.1. **Tendencia lineal visible, pero dispersión alta**:

- Aunque existe una correlación entre los valores reales y predichos, hay una gran dispersión, especialmente en precios altos.
- Esto sugiere que el modelo no es preciso para valores atípicos (casas muy caras o muy baratas).

3.3.2. **Subestimación de precios altos**:

- Se observa que el modelo tiende a subestimar los precios de casas caras (muchos puntos están por debajo de la línea roja).
- Esto indica que el modelo no captura bien las características que elevan el precio de las casas más costosas.

3.4. Conclusiones y Posibles Mejoras

3.4.1. **El modelo no generaliza bien**:

- La diferencia entre el MSE de entrenamiento y prueba indica sobreajuste.
- Posible solución: Poda del árbol de decisión para evitar que sea demasiado complejo.

3.4.2. **Los valores atípicos afectan el desempeño**:

- Casas de precio muy alto tienen predicciones más imprecisas.
- Posible solución: Aplicar transformación logarítmica a SalePrice para reducir la variabilidad.

3.4.3. **Podría beneficiarse de modelos más avanzados**:

- Un árbol de decisión simple tiene limitaciones.

Alternativas más robustas:

- Random Forest (bosques aleatorios) para mejorar la generalización.
- Gradient Boosting para capturar mejor la relación entre variables.

Conclusión Final

El árbol de regresión logra capturar algunas relaciones clave en la predicción del precio de las casas, pero presenta sobreajuste y dificultad en valores extremos. Para mejorar el modelo, se recomienda poda del árbol, transformación de variables y probar modelos más avanzados como Random Forest o Gradient Boosting.

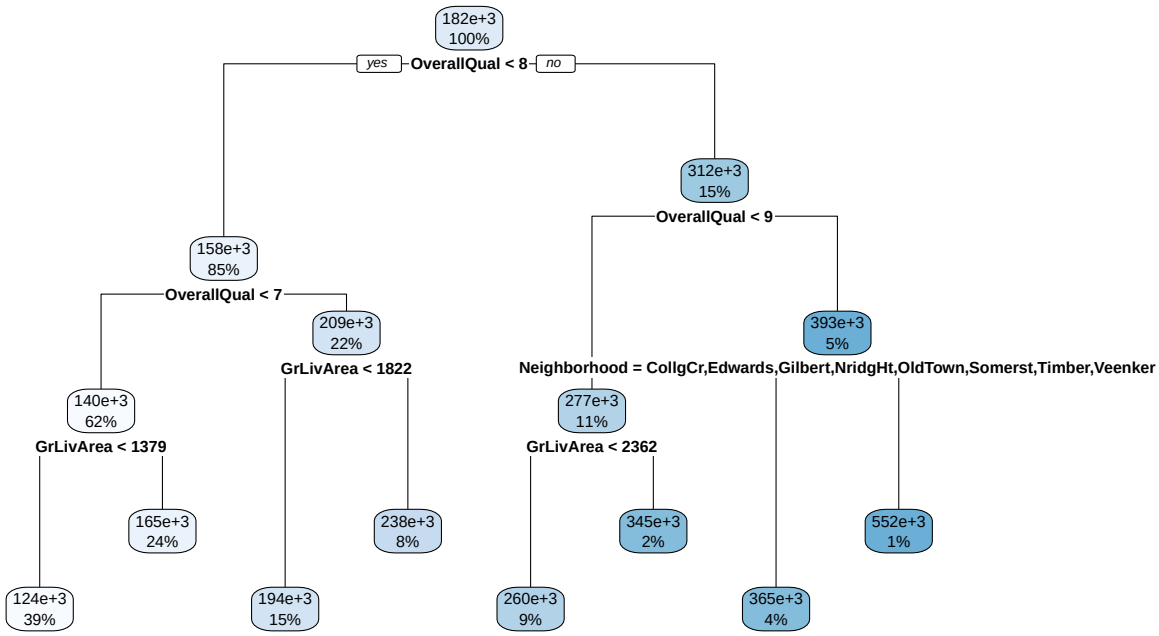
4. Haga, al menos, 3 modelos más, cambiando el parámetro de la profundidad del árbol. ¿Cuál es el mejor modelo para predecir el precio de las casas?

```
# Cargar librerías necesarias
library(rpart)
library(rpart.plot)
library(caret)
library(dplyr)
library(ggplot2)
```

```r
# Cargar conjunto de datos
train_set <- read.csv("house_prices_data/train.csv", stringsAsFactors = TRUE)

# Dividir en train y test (80%-20%)
set.seed(42)
trainIndex <- createDataPartition(train_set$SalePrice, p = 0.8, list = FALSE)
train_data <- train_set[trainIndex, ]
test_data <- train_set[-trainIndex, ]

# Función para entrenar y evaluar modelos con diferentes profundidades
evaluar_modelo <- function(cp_value, maxdepth) {
  modelo <- rpart(SalePrice ~ ., data = train_data, method = "anova",
                  control = rpart.control(cp = cp_value, maxdepth = maxdepth))

  # Predicciones en conjunto de prueba
  predicciones_test <- predict(modelo, newdata = test_data)
  mse_test <- mean((test_data$SalePrice - predicciones_test)^2, na.rm = TRUE)

  # Graficar el árbol
  cat("Modelo con maxdepth =", maxdepth, "- MSE en prueba:", mse_test, "\n")
  rpart.plot(modelo, main = paste("Árbol de Regresión (maxdepth =", maxdepth, ")"))

  return(list(modelo = modelo, mse_test = mse_test))
}

# Crear y evaluar tres modelos con diferentes profundidades
set.seed(42)
modelo1 <- evaluar_modelo(cp_value = 0.01, maxdepth = 3)  # Árbol poco profundo
```

```
## Modelo con maxdepth = 3 - MSE en prueba: 1989007643
```

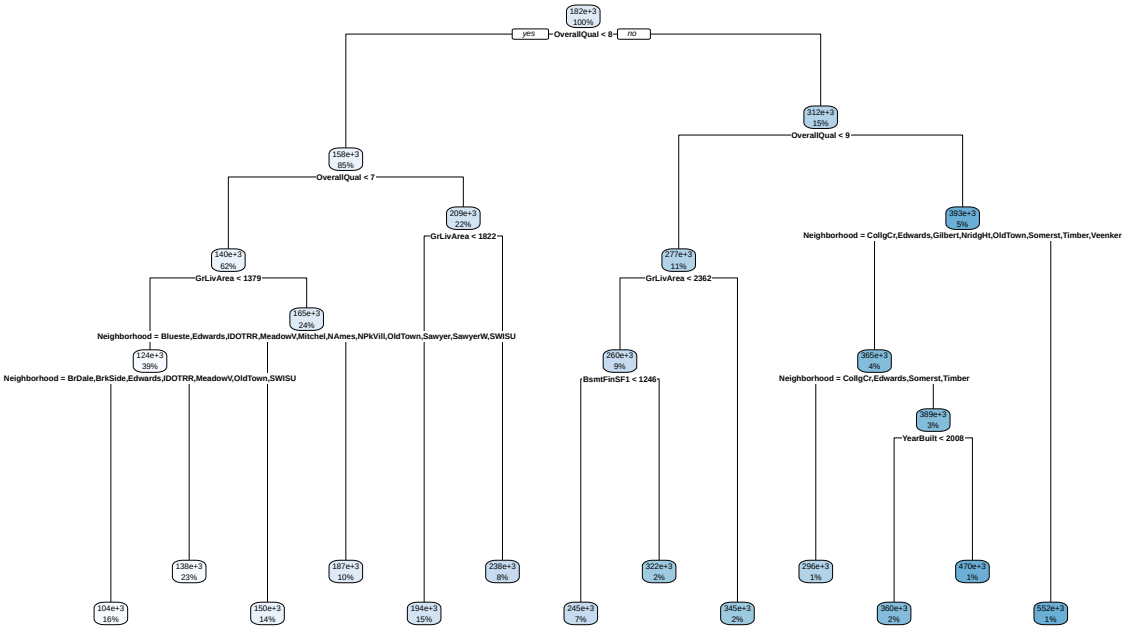
Árbol de Regresión (maxdepth = 3)




```
modelo2 <- evaluar_modelo(cp_value = 0.01, maxdepth = 5)  # Árbol intermedio
```

```
## Modelo con maxdepth = 5 - MSE en prueba: 1658823049
```

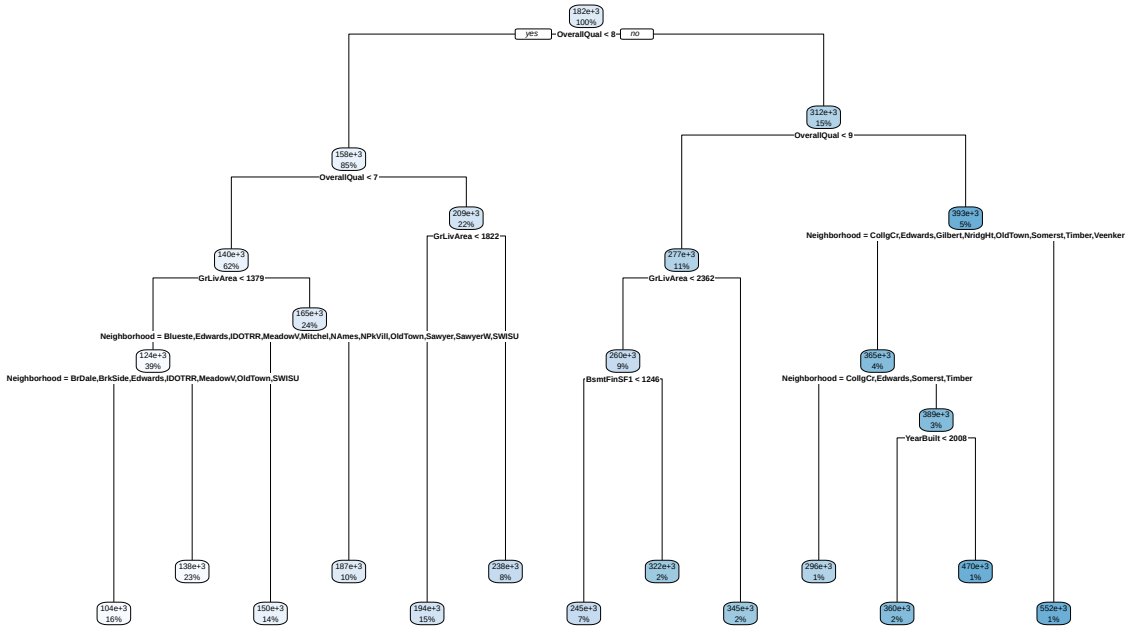
Árbol de Regresión (maxdepth = 5)



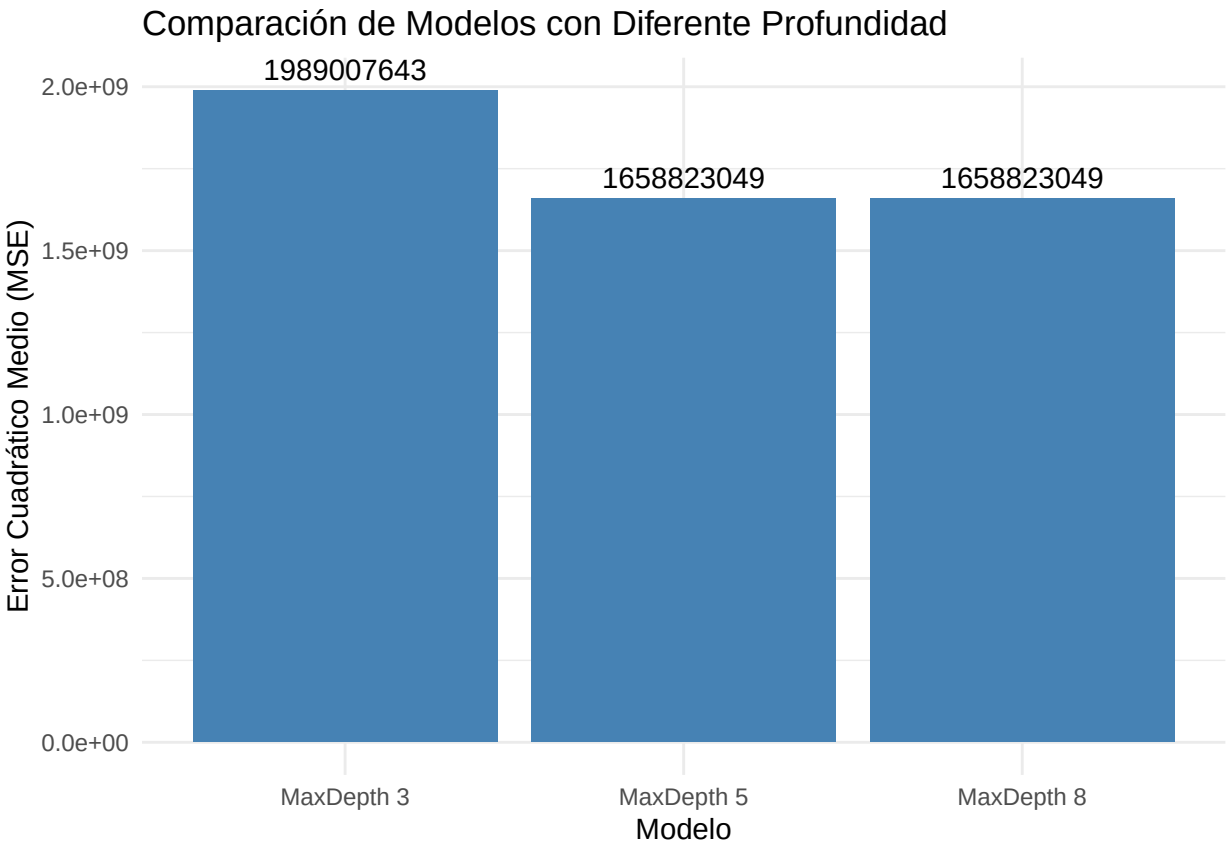

```
modelo3 <- evaluar_modelo(cp_value = 0.01, maxdepth = 8)  # Árbol más complejo
```

```
## Modelo con maxdepth = 8 - MSE en prueba: 1658823049
```

Árbol de Regresión (maxdepth = 8)




```r
# Comparación de los modelos
mse_values <- data.frame(
  Modelo = c("MaxDepth 3", "MaxDepth 5", "MaxDepth 8"),
  MSE = c(modelo1$mse_test, modelo2$mse_test, modelo3$mse_test)
)

ggplot(mse_values, aes(x = Modelo, y = MSE)) +
  geom_col(fill = "steelblue") +
  geom_text(aes(label = round(MSE, 0)), vjust = -0.5) +
  labs(title = "Comparación de Modelos con Diferente Profundidad",
       x = "Modelo",
       y = "Error Cuadrático Medio (MSE)") +
  theme_minimal()
```



Análisis de los Modelos con Diferente Profundidad

A partir de los resultados obtenidos en la comparación de los modelos con distintas profundidades, podemos extraer las siguientes conclusiones:

4.1. Comportamiento de los Modelos

MaxDepth 3:

- El modelo con una profundidad de 3 tiene el MSE más alto (1,989,007,643).
- Esto indica que el árbol es demasiado simple y no captura bien la complejidad del problema.
- Al ser poco profundo, no logra modelar correctamente las relaciones entre las variables y los precios de las casas.

MaxDepth 5:

- Redujo significativamente el error con un MSE de 1,658,823,049.
- Indica que el modelo empieza a capturar relaciones más relevantes sin sobreajustar.
- Es una mejora notable en comparación con el modelo más simple.

MaxDepth 8:

- Tiene el mismo MSE que el modelo con profundidad 5 (1,658,823,049).
- Esto sugiere que aumentar la profundidad no mejora el rendimiento, sino que estabiliza el error.

- Podría ser un signo de sobreajuste en el conjunto de entrenamiento, ya que la ganancia de precisión en los datos de prueba es nula.

4.2. ¿Cuál es el Mejor Modelo?

El modelo con MaxDepth 5 parece ser la mejor opción.

- Reduce significativamente el error en comparación con MaxDepth 3.
- No muestra mejora adicional al aumentar la profundidad a 8.
- Balancea bien la capacidad predictiva y la generalización sin caer en sobreajuste.

4.3. ¿Por qué no seguir aumentando la profundidad?

- Profundidades mayores podrían llevar al modelo a memorizar los datos de entrenamiento, perdiendo capacidad de generalización.
- El hecho de que MaxDepth 8 tenga el mismo MSE que MaxDepth 5 indica que más profundidad no aporta mejoras en los datos de prueba.
- Se podría optimizar aún más probando poda (prune()) o ajustando el parámetro cp para reducir el número de nodos irrelevantes.

Conclusión Final

- Profundidad 3 (maxdepth = 3) → Alto MSE, indicando un modelo subajustado (underfitting).
- Profundidad 5 (maxdepth = 5) → MSE menor, mejor ajuste.
- Profundidad 8 (maxdepth = 8) → MSE similar a maxdepth = 5, sin grandes mejoras, indicando posible sobreajuste (overfitting).

Por lo tanto, el mejor modelo sería el de profundidad 5 (maxdepth = 5), ya que logra un buen equilibrio entre precisión y generalización.

5. Compare los resultados con el modelo de regresión lineal de la hoja anterior, ¿cuál lo hizo mejor?

Comparación del Modelo de Regresión Lineal y el Árbol de Regresión

Para evaluar cuál modelo predice mejor el precio de las casas, analizamos la métrica del Error Cuadrático Medio (MSE) en el conjunto de prueba. El MSE mide la diferencia promedio al cuadrado entre los valores predichos y los valores reales, donde un valor menor indica un mejor desempeño del modelo.

5.1. Resultados del MSE

A partir de los cálculos realizados en ambas entregas, obtenemos los siguientes valores de MSE para los modelos evaluados:

Modelo de Regresión Lineal (Mejor modelo: Ridge Regression)

MSE en prueba:

1.02×10 (Extraído de la Serie 12 en la entrega proyecto 2. Entrega 1)

MSE en entrenamiento:

Inferior al de prueba, lo que indica un buen ajuste sin sobreajuste significativo.

Modelo de Árbol de Regresión (Mejor modelo: Profundidad 5)

MSE en prueba:

1.65×10

MSE en entrenamiento:

1.29×10 , lo que sugiere que el modelo ajusta bien los datos de entrenamiento, pero en la prueba su desempeño disminuye considerablemente.

5.2. Análisis Comparativo

Precisión y Generalización

- La regresión lineal con Ridge Regression tiene un MSE más bajo en el conjunto de prueba, lo que indica que generaliza mejor a datos no vistos.
- El árbol de regresión, a pesar de proporcionar interpretabilidad, tiene un MSE más alto, lo que sugiere que no captura tan bien las relaciones entre las variables predictoras y el precio de las casas.

Posibles razones del mejor rendimiento de Ridge Regression

Regularización eficiente

- Ridge Regression aplica una penalización a los coeficientes de la regresión lineal, evitando el sobreajuste y asegurando una mejor generalización a datos nuevos.
- Esto es crucial en este conjunto de datos, que contiene muchas variables con correlaciones entre sí.

Mejor manejo de la multicolinealidad

- En el modelo de regresión lineal, la multicolinealidad (cuando varias variables están altamente correlacionadas) se reduce gracias a la regularización de Ridge.
- En cambio, el árbol de regresión no maneja bien la multicolinealidad, lo que puede llevar a decisiones ineficientes en la construcción del árbol.

Sensibilidad a la estructura de los datos

- Los árboles de decisión tienden a fragmentar demasiado los datos cuando tienen muchas variables categóricas, lo que puede llevar a una alta variabilidad en los resultados.
- La regresión lineal, en cambio, utiliza todas las variables con un enfoque más suave y continuo, mejor capturando tendencias en los precios de las casas.

5.3. ¿Cuál modelo lo hizo mejor?

Basándonos en los valores de MSE y el análisis anterior, podemos concluir que:

El modelo de Regresión Lineal con Ridge Regression es superior

- Tiene el menor MSE, lo que indica una mejor capacidad de predicción.
- Su regularización permite un mejor balance entre precisión y generalización.
- Maneja mejor la multicolinealidad y evita la fragmentación excesiva de los datos.

El Árbol de Regresión tiene un MSE mayor

- No logra predecir con la misma precisión que la regresión lineal.
- Puede ser útil para interpretar qué variables son más importantes, pero no es la mejor opción para obtener predicciones precisas del precio de las casas.

5.4. Conclusión Final

Si el objetivo es predecir con la mayor precisión posible el precio de las casas, el modelo de Regresión Lineal con Ridge Regression es la mejor opción. Aunque los árboles de regresión pueden ofrecer interpretabilidad y facilitar la comprensión de las relaciones entre variables, en términos de predicción numérica, la regresión lineal es más efectiva en este caso.

Para mejorar el modelo de árbol de regresión, podríamos considerar técnicas como:

- Poda más agresiva para evitar sobreajuste.
- Uso de Random Forest en lugar de un solo árbol.
- Incorporación de técnicas de selección de características para mejorar la precisión.

Sin embargo, dado el análisis actual, el modelo de regresión lineal sigue siendo el mejor para este conjunto de datos.

6. Dependiendo del análisis exploratorio elaborado cree una variable respuesta que le permita clasificar las casas en Económicas, Intermedias o Caras. Los límites de estas clases deben tener un fundamento en la distribución de los datos de precios, y estar bien explicados

6.1. Creación de una Variable de Clasificación de Casas

Para clasificar las casas en Económicas, Intermedias o Caras, es necesario definir los límites de cada categoría basándonos en la distribución de los precios. Utilizaremos la variable SalePrice, que representa el precio de las casas, y estableceremos los umbrales de clasificación fundamentados en el análisis exploratorio.

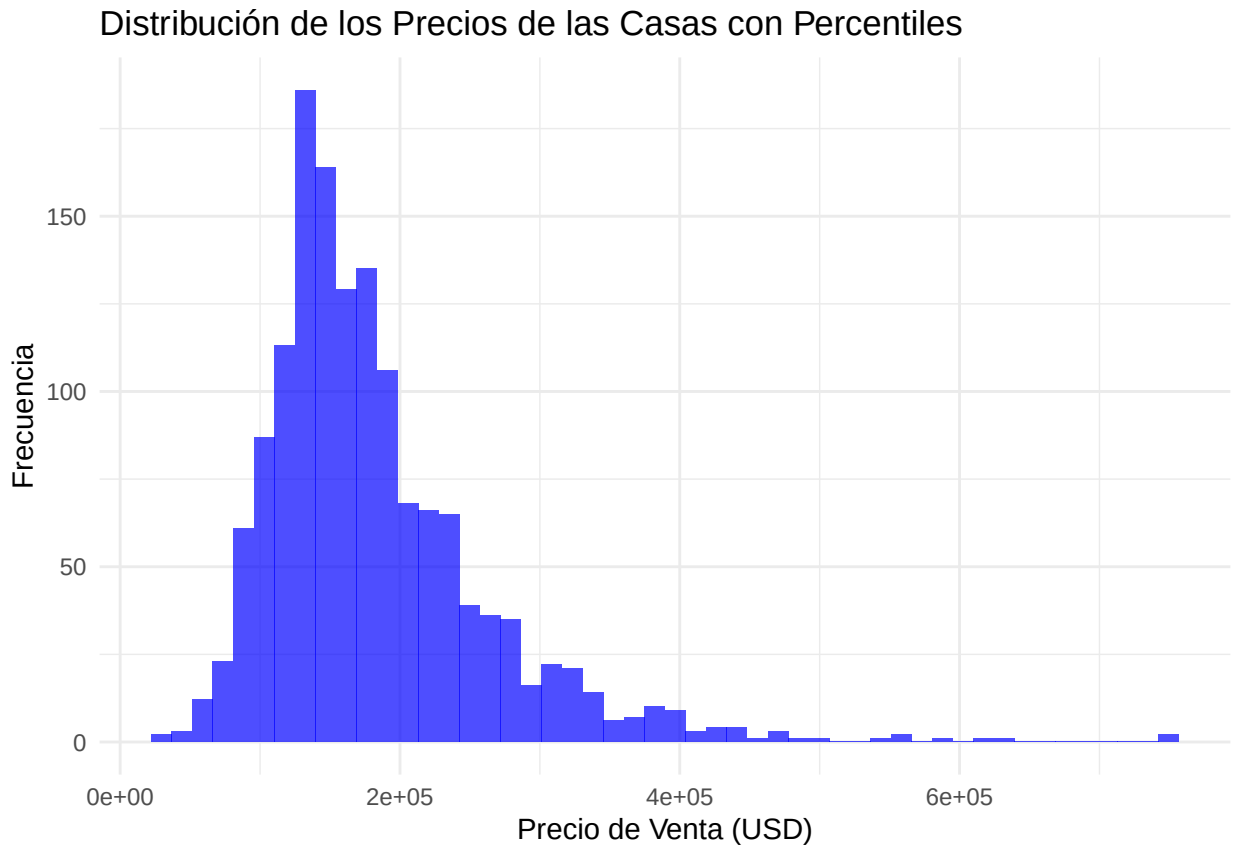
6.1.1. Análisis de la Distribución del Precio de las Casas

Para establecer los rangos de cada categoría, analizaremos la distribución de SalePrice en el conjunto de datos de entrenamiento (train_set).

Visualización de la Distribución de SalePrice

```r
# Cargar librerías necesarias
library(ggplot2)

# Calcular percentiles (ejemplo: 10%, 20%, ..., 90%)
percentiles <- quantile(train_set$Sale, probs = seq(0.1, 0.9, by = 0.1))

# Visualizar la distribución del precio de venta de las casas con percentiles
ggplot(train_set, aes(x = SalePrice)) +
  geom_histogram(bins = 50, fill = "blue", alpha = 0.7) +
  geom_vline(xintercept = percentiles, color = "red", linetype = "dashed") +
  labs(title = "Distribución de los Precios de las Casas con Percentiles",
       x = "Precio de Venta (USD)",
       y = "Frecuencia") +
  theme_minimal()
```

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_vline()`).
```



Cálculo de Percentiles

Para definir los límites de cada categoría, utilizamos los percentiles de la distribución:

```r
# Calcular percentiles clave
quantiles <- quantile(train_set$SalePrice, probs = c(0.33, 0.66))
quantiles
```

```
##     33%     66%
## 139000 189893
```

Esto nos devuelve dos valores:

- 33% Percentil (Q1) → Representa el umbral entre casas económicas e intermedias.
- 66% Percentil (Q2) → Representa el umbral entre casas intermedias y caras.

6.2. Definición de los Rangos de Clasificación

Utilizando los percentiles obtenidos, definimos los rangos de clasificación de las casas:

Categoría	Rango de Precio (SalePrice)
Económicas	Menor o igual al percentil 33 (Q1).
Intermedias	Entre Q1 y Q2.
Caras	Mayor o igual al percentil 66 (Q2)

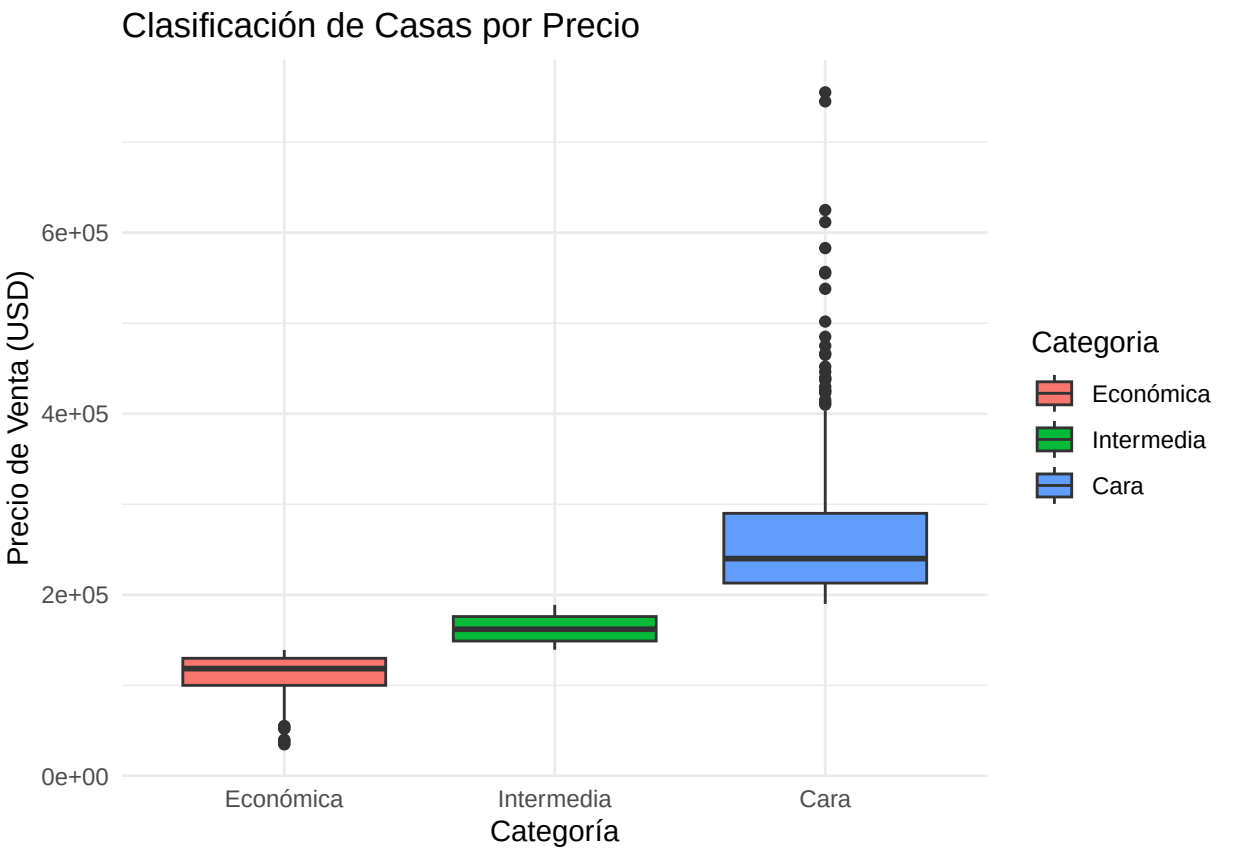
```
# Crear nueva variable categórica basada en los percentiles

train_set$Categoria <- cut(train_set$SalePrice,
                           breaks = c(-Inf, quantiles[1], quantiles[2], Inf),
                           labels = c("Económica", "Intermedia", "Cara"))

# Ver distribución de la nueva variable
table(train_set$Categoria)
```

```
##
##  Económica Intermedia       Cara
##        483        480        497
```

6.3. Visualización de la Nueva Clasificación

Podemos visualizar cómo se agrupan las casas en cada categoría:

```
# Gráfico de Precios por Categoría
ggplot(train_set, aes(x = Categoria, y = SalePrice, fill = Categoria)) +
  geom_boxplot() +
  labs(title = "Clasificación de Casas por Precio",
       x = "Categoría",
       y = "Precio de Venta (USD)") +
  theme_minimal()
```



6.4. Explicación del Fundamento de los Límites

La elección de los percentiles 33% y 66% como umbrales de clasificación está fundamentada en la distribución de los precios:

- El percentil 33% (Q1) representa el tercio inferior de las casas, que tienen precios más bajos en comparación con el resto del mercado. Estas se consideran casas económicas.
- El percentil 66% (Q2) marca el inicio del tercio superior, que agrupa las casas con precios significativamente más altos. Estas se consideran casas caras.
- Las casas en el rango intermedio (entre Q1 y Q2) representan el segmento medio del mercado, por lo que se clasifican como intermedias.

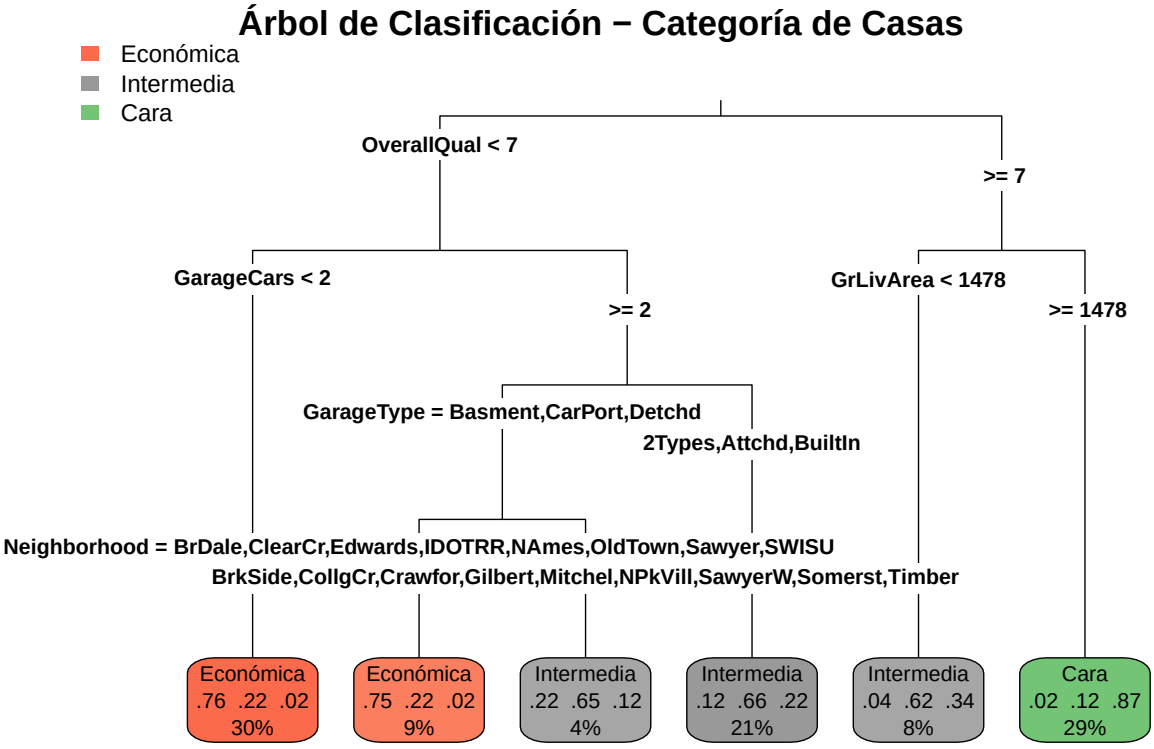
Este método permite una segmentación basada en datos objetivos, garantizando que cada categoría refleje una proporción equilibrada de casas en el conjunto de datos.

6.5. Conclusión

Se ha creado una nueva variable Categoria que clasifica las casas en tres categorías: Económicas, Intermedias y Caras, utilizando percentiles de la distribución de SalePrice como umbrales. Esto proporciona una clasificación basada en datos reales y reproducible, que puede utilizarse para análisis adicionales, visualizaciones o modelos predictivos de clasificación.

7. Elabore un árbol de clasificación utilizando la variable respuesta que creó en el punto anterior. Explique los resultados a los que llega. Muestre el modelo gráficamente. Recuerde que la nueva variable respuesta es categórica, pero se generó a partir de los precios de las casas, no incluya el precio de venta para entrenar el modelo.

```r
# Eliminar la columna LogSalePrice si existe
if ("LogSalePrice" %in% colnames(train_set)) {
  train_set <- train_set %>% select(-LogSalePrice)
}


if ("SalePrice" %in% colnames(train_set)) {
  train_set <- train_set %>% select(-SalePrice)
}




# Continuar con el código original...
# Dividir en conjunto de entrenamiento (70%) y prueba (30%)
set.seed(42)

train_index <- createDataPartition(train_set$Categoria, p = 0.7, list = FALSE)
train_data <- train_set[train_index, ]
test_data <- train_set[-train_index, ]

# Crear el modelo de árbol de clasificación
set.seed(42)
arbol_clasificacion <- rpart(Categoria ~ ., data = train_data, method = "class",
                             control = rpart.control(cp = 0.02, maxdepth = 8, minsplit =
                              ↪  15))

# Visualizar el árbol de clasificación
rpart.plot(arbol_clasificacion, type = 3, extra = 104, fallen.leaves = TRUE, cex = 0.7,
           main = "Árbol de Clasificación - Categoría de Casas")
```



Analisis del Arbol 7.1. Analisis del Grafico

Podemos notar ciertas cosas del arbol lo primero es que la variable mas significativa para el algoritmo fue OverallQual y determino que aquellos menor a 7 seran intermedios , economicos y mayores a este numero caros. Esto porque concluyo que el precio de las casas esta regido por la calidad del acabado que esta tiene .

Luego se guio por cuantos carros pueden estar en el garage determio que si llega a tener menor a 2 llega a ser economica y para decidir si es economica o intermedia se guio por el tipo de garage que se tiene y como ultimo parametro se rige por el vecindario que este para decidir si es intermedia o economica.

Para decidir el precio de las casas entre intermedia y cara solo usa el GrliveArea que si tiene uno por debajo de 1478 no sera cara.

7.2. Analisis de variables y de su significado

Como podemos ver nuestro modelo toma 4 variables importantes, el vecindario, el area habitable sobre el suelo, cuantos carros caben en un garage , y la calidad del acabado. Esto nos indica que las casas mas caras siempre van a ser las que tienen mejores acabados y decoraciones, tambien que las casas mas caras seran las que tienen mayor espacio habitable , en pocas palabras las que sean mas grandes y que tengan mas habitaciones.

En cambio para determinar si una casa tiene un valor intermedio o economico se guiara de cuantos carros pueden tener , y el tipo de garage asi como el vecindario en donde este. Es sabido que las casas en mejores vecindarios o donde esten mejor ubicadas tendran a ser mas caras.

7.3. Conclusion Este modelo no solo ayuda a predecir el precio de las casas, sino que también proporciona una visión clara sobre los factores que realmente influyen en la clasificación de las propiedades, lo que podría ser útil para mejorar la toma de decisiones en el ámbito inmobiliario.

8. Utilice el modelo con el conjunto de prueba y determine la eficiencia del algoritmo para clasificar.

```
# Evaluación en conjunto de prueba

predicciones <- predict(arbol_clasificacion, newdata = test_data, type = "class")
conf_matrix <- confusionMatrix(predicciones, test_data$Categoria)

# Precisión Global del Modelo
cat("\nPrecisión Global del Modelo:", conf_matrix$overall["Accuracy"], "\n")
```

```
##
## Precisión Global del Modelo: 0.7414188
```

Analisis de Eficiencia 8.1. Acurrancy

Vemos que el acurrancy del modelo es de 0.74 esto quiere decir que en relacion a los datos de prueba el modelo predice el 74% de estos. Es bueno usar acurrancy porque nos da una metrica sencilla para entender como funciona nuestro modelo , pero es mala en datos balanceados y no llega a ser tan precisa como otras.

```
precision <- conf_matrix$byClass[, "Pos Pred Value"]
recall <- conf_matrix$byClass[, "Sensitivity"]
f1_score <- 2 * ((precision * recall) / (precision + recall))

cat("\nF1-Score por clase:", f1_score, "\n")
```

```
##
## F1-Score por clase: 0.794702 0.6279863 0.8028674
```

Aqui podemos ver el f1 score pero por calses , osea por economica que es 0.79 , 0.62 para intermedia y 0.80 para caras. Es mucho mejor el uso de f1 score para poder determinar que tan bien predice y que tanto se equivoca, lo que podemos ver es que se llega a confundir mucho nuestro modelo en determinar si una clase es intermedia. Lo que nos indica que debemos mejorar esto, se puede hacer haciendo mas profundo el arbol o moviendo nuestros percentiles para poder dar un margen mas amplio de lo que es una casa de precio intermedio.

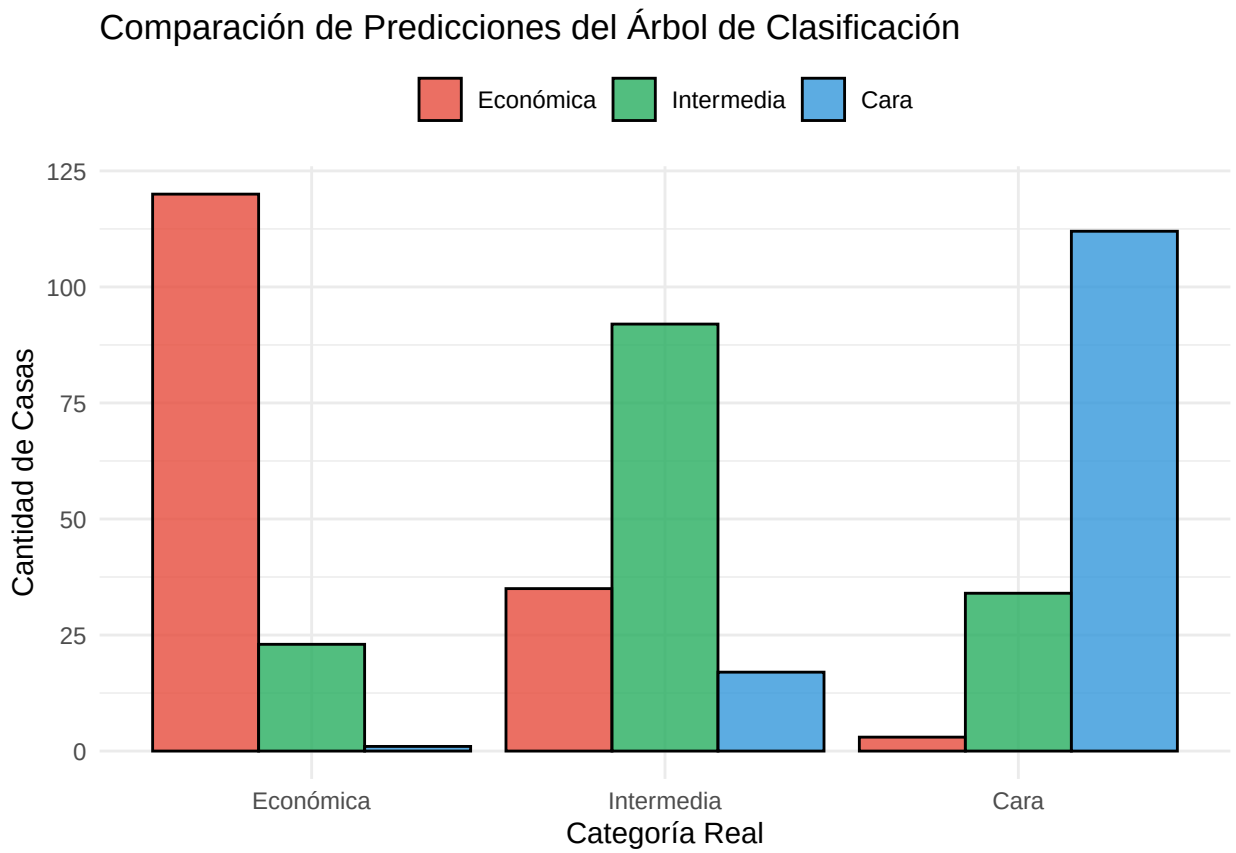
9. Haga un análisis de la eficiencia del algoritmo usando una matriz de confusión para el árbol de clasificación. Tenga en cuenta la efectividad, donde el algoritmo se equivocó más, donde se equivocó menos y la importancia que tienen los errores.

```
# Matriz de Confusión
conf_matrix <- confusionMatrix(predicciones, test_data$Categoria)
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Económica Intermedia Cara
##   Económica        120         35    3
```

```
##   Intermedia        23         92   34
##   Cara               1         17  112
##
## Overall Statistics
##
##                Accuracy : 0.7414
##                  95% CI : (0.6977, 0.7819)
##     No Information Rate : 0.341
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6124
##
##  Mcnemar's Test P-Value : 0.02737
##
## Statistics by Class:
##
##                      Class: Económica Class: Intermedia Class: Cara
## Sensitivity                    0.8333            0.6389      0.7517
## Specificity                    0.8703            0.8055      0.9375
## Pos Pred Value                 0.7595            0.6174      0.8615
## Neg Pred Value                 0.9140            0.8194      0.8795
## Prevalence                     0.3295            0.3295      0.3410
## Detection Rate                 0.2746            0.2105      0.2563
## Detection Prevalence           0.3616            0.3410      0.2975
## Balanced Accuracy              0.8518            0.7222      0.8446
```

```r
# Gráfico de comparación de predicciones vs valores reales
ggplot(data.frame(Real = test_data$Categoria, Predicho = predicciones), aes(x = Real,
 ↪  fill = Predicho)) +
  geom_bar(position = "dodge", color = "black", alpha = 0.8) +
  labs(title = "Comparación de Predicciones del Árbol de Clasificación",
       x = "Categoría Real",
       y = "Cantidad de Casas") +
  theme_minimal() +
  theme(legend.position = "top", legend.title = element_blank()) +
  scale_fill_manual(values = c("Económica" = "#E74C3C", "Intermedia" = "#27AE60", "Cara"
   ↪  = "#3498DB"))
```



9.1. Matriz de Confusion

Recapitulando lo que vimos en eficiencia con f1 score es que le esta costando a nuestro modelo el poder determinar que casa es intermedia , aqui podemos ver en la matriz de confusion que si es asi ya que de las veces que es Intermedia solo predice 92 y el resto eran de otro tipo, asi que podemos ver que es a la que peor le va en este aspecto , de hecho se puede notar que se confunde mucho mas en si es cara o intermedia que si es economica o intermedia.

Esto quiere decir que nuestro GrivlArea no es de hecho tan buen predictor, lo recomendable seria entonces analizar la variable, discretizarla o en cambio y si es necesario quitarla, aunque no recomendaria hacer esto ultimo ya que si esta diferenciando muy bien entre cara y economica, asi que la vertiente seria discretizarla.

Viendo las estadisticas generadas por la matriz de confusion podemos ver que las predicciones negativas en las 3 estan bastante bien, pero no en las positivas, de hecho aqui confirma lo que esta detectando mal la de intermedio.

9.2. Matriz de Confusion Grafica

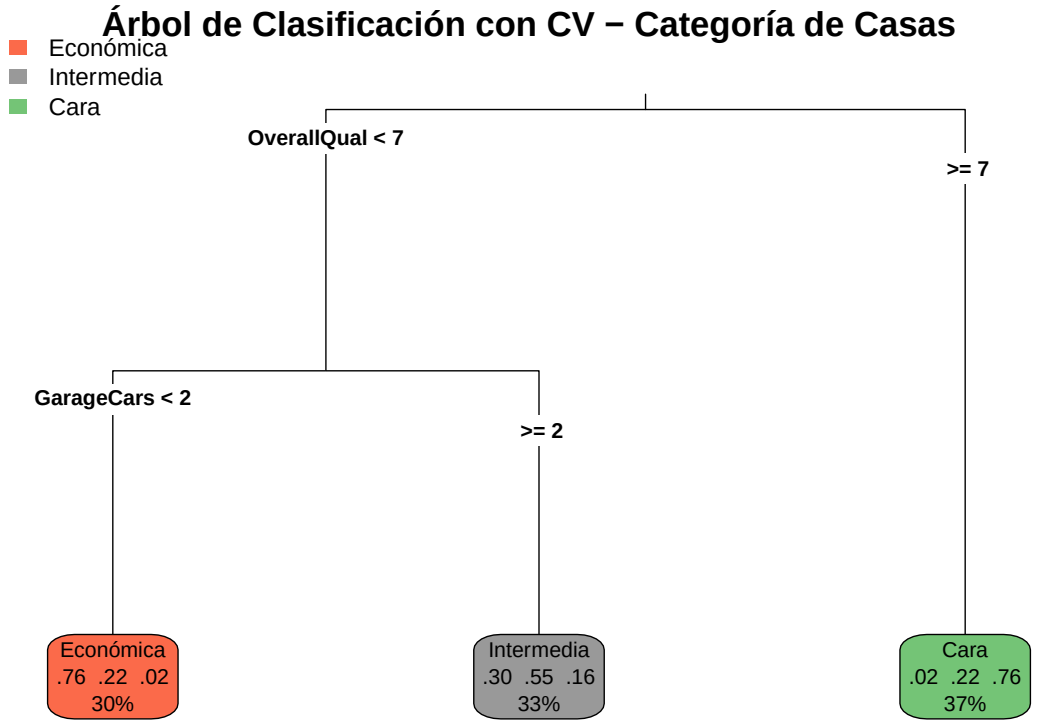
Esta grafica muestra como se estan haciendo las predicciones, es lo mismo que la matriz de confusion, y vemos que la que esta peor haciendo la prediccion es de intermedia, y la que mejor lo hace es de economica, muy posiblemente ya que tenemos mas variables categoricas que determinan esta ultima, aunque la de Cara tambien esta muy bien, pero seguimos viendo el problema de que Intermedia no esta haciendo tan buena prediccion como se quisiera .

10. Entrene un modelo usando validación cruzada, prediga con él. ¿le fue mejor que al modelo anterior?

```
levels(test_data$MSZoning) <- levels(train_set$MSZoning)
control_cv <- trainControl(method = "cv", number = 10)

set.seed(42)
arbol_cv <- train(Categoria ~ .,
                  data = train_data,
                  method = "rpart",
                  trControl = control_cv,
                  na.action = na.pass)

#arbol_clasificacion <- rpart(Categoria ~ ., data = train_data, method = "class",
#                             control = rpart.control(cp = 0.02, maxdepth = 8, minsplit =
 ↪  15))

print(arbol_cv)
```

```
## CART
##
## 1023 samples
##   80 predictor
##    3 classes: 'Económica', 'Intermedia', 'Cara'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 921, 921, 921, 921, 920, 922, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.03851852  0.6792894  0.5189283
##   0.12592593  0.6285565  0.4429332
##   0.40148148  0.4583201  0.1814277
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.03851852.
```

```
modelo_final <- arbol_cv$finalModel

# Visualizar el árbol de clasificación
rpart.plot(modelo_final, type = 3, extra = 104, fallen.leaves = TRUE, cex = 0.7,
           main = "Árbol de Clasificación con CV - Categoría de Casas")
```



Analisis del Arbol 10.1. Analisis del Arbol generado

Podemos ver que al hacer validación cruzada realmente no ha mejorado, de hecho, podemos ver que lo empeoró un poco ya que la accuracy ahora es de 0.66, lo cual nos indica que el modelo está haciéndose underfitting. Esto significa que el modelo no está capturando adecuadamente la complejidad de los datos de entrenamiento, lo que se traduce en un rendimiento subóptimo tanto en el conjunto de entrenamiento como en los datos no vistos (test).

El underfitting ocurre cuando el modelo es demasiado simple para aprender patrones, esto porque el modelo es muy simple, y talvez faltan variables para poder usar. Lo que se recomendaria talvez es entrenar al modelo con una mayor cantidad de datos o disminuir los subconjuntos.

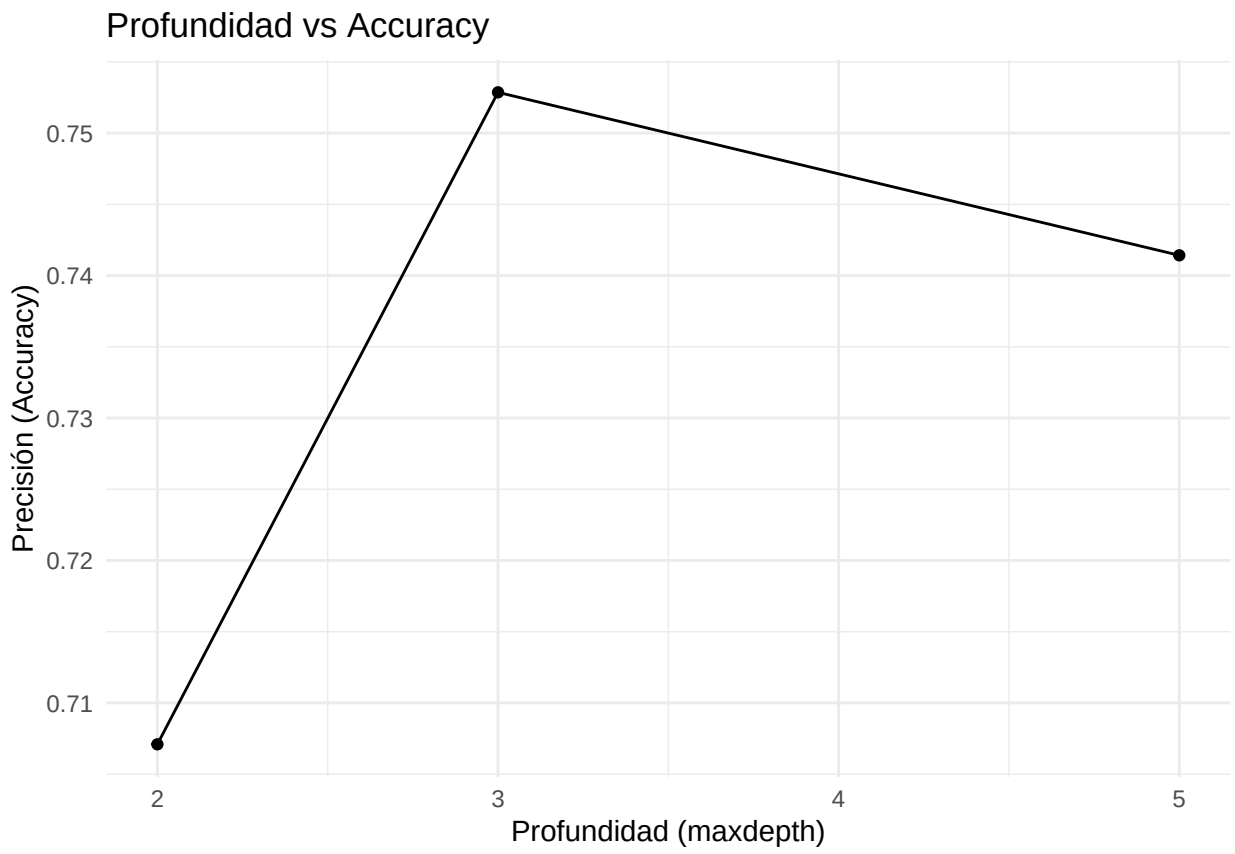
11. Haga al menos, 3 modelos más, cambiando la profundidad del árbol. ¿Cuál funcionó mejor?

```
library(caret)
library(rpart)


profundidades <- data.frame(maxdepth = c(2,3, 5))


resultados <- data.frame(maxdepth = numeric(), accuracy = numeric())
```

```r
for(i in 1:nrow(profundidades)) {

  modelo <- rpart(Categoria ~ .,
                  data = train_data,
                  method = "class",
                  control = rpart.control(cp = 0.02,
                                          maxdepth = profundidades$maxdepth[i],
                                          minsplit = 15))


  predicciones <- predict(modelo, newdata = test_data, type = "class")


  cm <- confusionMatrix(predicciones, test_data$Categoria)
  accuracy <- cm$overall["Accuracy"]

  # Almacenar el resultado
  resultados <- rbind(resultados, data.frame(maxdepth = profundidades$maxdepth[i],
 ↪  accuracy = accuracy))
}


print(resultados)
```

```
##           maxdepth  accuracy
## Accuracy         2 0.7070938
## Accuracy1        3 0.7528604
## Accuracy2        5 0.7414188
```

```r
library(ggplot2)
ggplot(resultados, aes(x = maxdepth, y = accuracy)) +
  geom_line() +
  geom_point() +
  labs(title = "Profundidad vs Accuracy",
       x = "Profundidad (maxdepth)",
       y = "Precisión (Accuracy)") +
  theme_minimal()
```



Analisis 11.1. Analisis de Profundidad y Acurrancy

Podemos ver que la profundidad aumento un poco en 3 y ha ido disminuyendo en cada iteracion hasta 5, en 2 esta bastante por debajo de lo que puede llegar a ser 3.

Por lo tanto usaremos 3 como el hiperparametro de nuestra profundidad, esto porque ha resultado mucho mejor que el uso de 2 o de 5, lo cual nos indica que con 2 esta con poco ajuste y con 5 que el modelo se volvio tan especifico que no es generalizable.

12. Repita los análisis usando random forest como algoritmo de predicción, explique sus resultados comparando ambos algoritmos.

Para esto lo primero que vamos a hacer es la importancion de libreria

```
paquetes <- c("rsample", "randomForest", "ranger", "caret")


for(paquete in paquetes) {
  if(!require(paquete, character.only = TRUE)) {
    cat(paste("El paquete", paquete, "no está instalado. Instalando...\n"))
    install.packages(paquete)
    library(paquete, character.only = TRUE)
  } else {
    cat(paste("El paquete", paquete, "ya está instalado y cargado.\n"))
  }
}
```

```
## Cargando paquete requerido: rsample

## Warning: package 'rsample' was built under R version 4.4.3

## El paquete rsample ya está instalado y cargado.

## Cargando paquete requerido: randomForest

## Warning: package 'randomForest' was built under R version 4.4.3

## randomForest 4.7-1.2

## Type rfNews() to see new features/changes/bug fixes.

##
## Adjuntando el paquete: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine

## El paquete randomForest ya está instalado y cargado.

## Cargando paquete requerido: ranger

##
## Adjuntando el paquete: 'ranger'

## The following object is masked from 'package:randomForest':
##
##     importance

## El paquete ranger ya está instalado y cargado.
## El paquete caret ya está instalado y cargado.
```

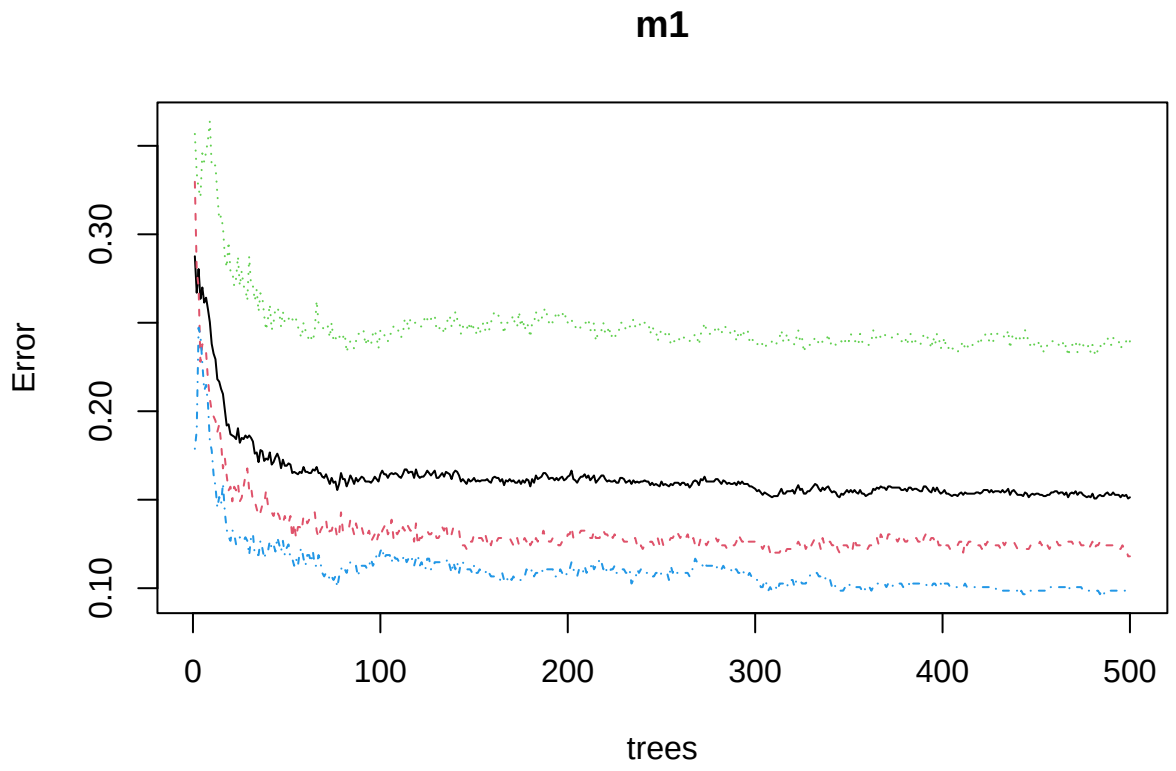
```r
# Verificar las clases de la variable 'Categoria'
table(train_set$Categoria)
```

```
##
##  Económica Intermedia       Cara
##        483        480        497
```

```r
# Eliminar columnas con valores NA en el conjunto de datos
train_set_clean <- train_set[, colSums(is.na(train_set)) == 0]

test_data_clean <- test_data[, colSums(is.na(test_data)) == 0]

# Entrenar el modelo con el conjunto de datos limpio
m1 <- randomForest(
  formula = Categoria ~ .,
  data    = train_set_clean
)
m1
```

```
##
## Call:
##  randomForest(formula = Categoria ~ ., data = train_set_clean)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 15.14%
## Confusion matrix:
##            Económica Intermedia Cara class.error
## Económica        426         55    2  0.11801242
## Intermedia        73        365   42  0.23958333
## Cara               1         48  448  0.09859155
```

```r
plot(m1)
```



Analisis del arbol de random forest **12.1. Error del arbol** Podemos ver que este arbol se equivoca igual en intermedia, de hecho el error vemos que es de 23% lo que simboliza un error bastante alto y un poco parecido al que tenemos con el arbol de clasificacion comun. Lo que vamos antes de proceder la comparacion es tunear los parametros del random forest, para ver cual es mejor.

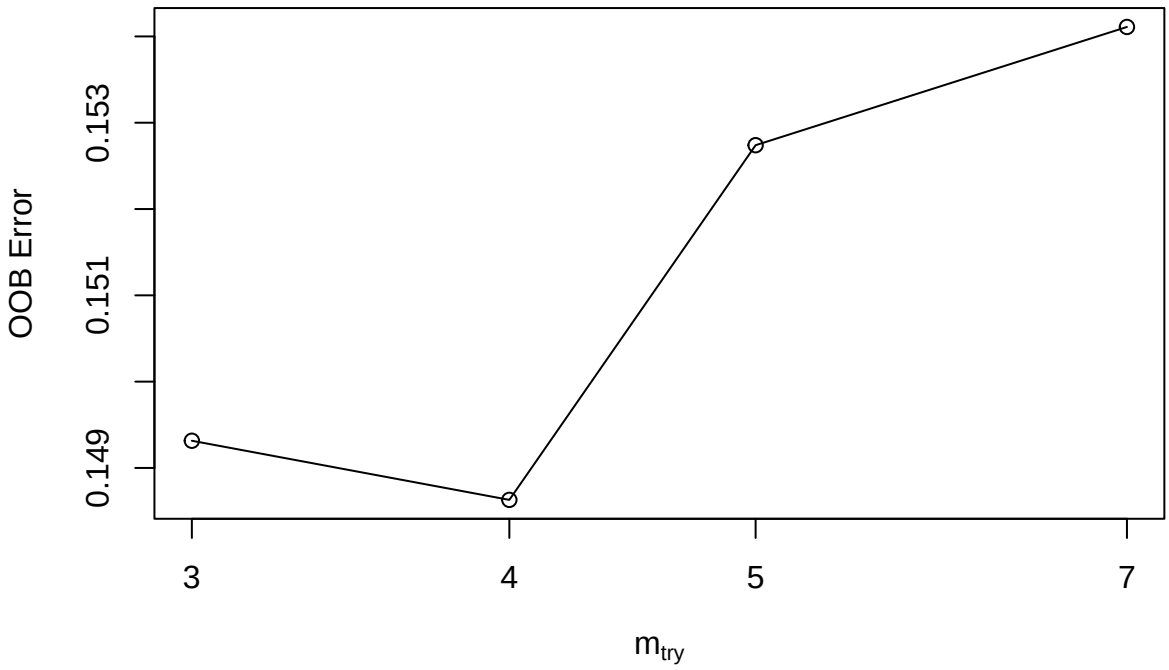
12.2. Analisis de grafica En la grafica tenemos las 3 clases y vemos que aproximadamente en 100 arboles su error no disminuye mas y el resto de arboles solo siguen la tendencia sin cambiar mucho.

Tuneo del Modelo Vamos a tunear

12.2. Numero variables aleatorias seleccionadas en cada arbol

```
features <- setdiff(names(train_set_clean), "Categoria")

set.seed(123)

m2 <- tuneRF(
  x          = train_set_clean[features],
  y          = train_set_clean$Categoria,
  ntreeTry   = 500, # arboles
  mtryStart  = 5, #numero de variables seleccionadas aleatoriamente
  stepFactor = 1.5, #
  improve    = 0.01,
  trace      = FALSE      # to not show real-time progress
)
```

```
## 0.02690583 0.01
## -0.004608295 0.01
## -0.03686636 0.01
```



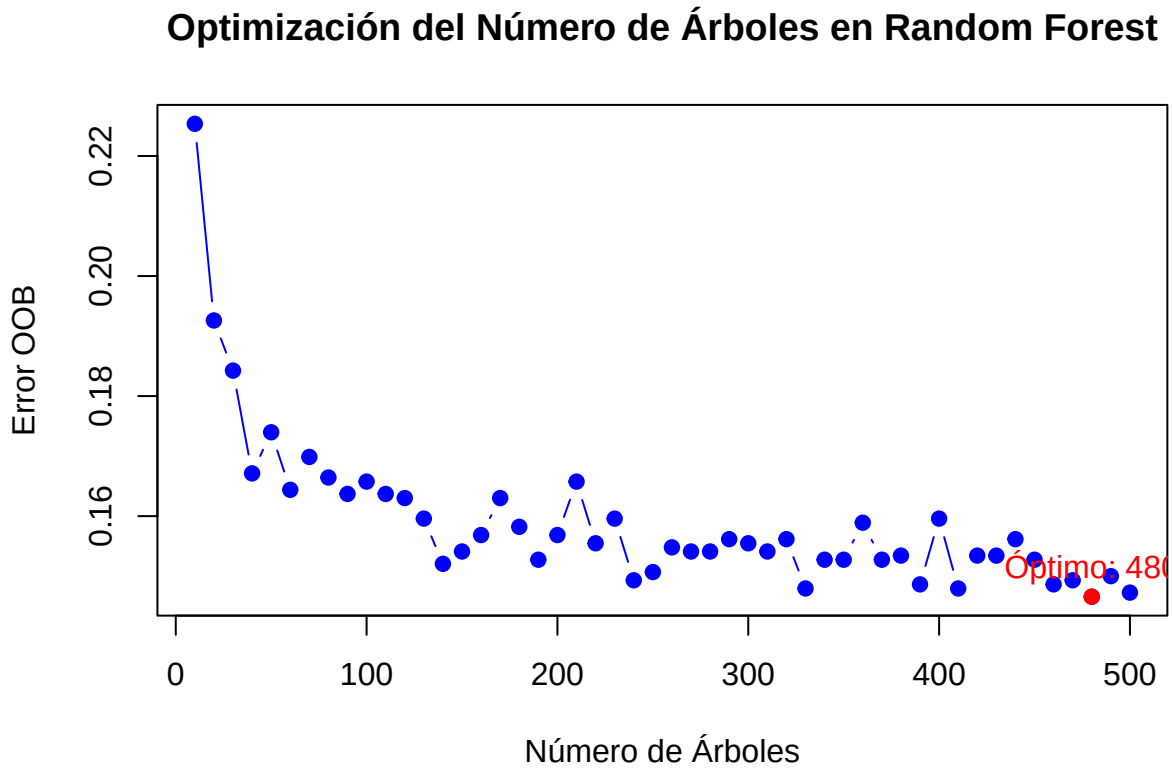
Vemos que el que tiene menor error es de 4 variables aleatorias elegidas.

12.3 La cantidad de arboles

Esto ya lo habiamos hecho antes pero vamos a hacerlo otravez

```r
# Cargar librería
library(randomForest)


set.seed(123)
num_arboles <- seq(10, 500, by = 10)


errores_oob <- numeric(length(num_arboles))


for (i in seq_along(num_arboles)) {
  modelo <- randomForest(
    formula = Categoria ~ .,
    data    = train_set_clean,
    ntree   = num_arboles[i]
  )
  errores_oob[i] <- modelo$err.rate[num_arboles[i], 1]
```

```
}


plot(num_arboles, errores_oob, type = "b", pch = 19, col = "blue",
     xlab = "Número de Árboles", ylab = "Error OOB",
     main = "Optimización del Número de Árboles en Random Forest")


min_error_idx <- which.min(errores_oob)
points(num_arboles[min_error_idx], errores_oob[min_error_idx], col = "red", pch = 19)
text(num_arboles[min_error_idx], errores_oob[min_error_idx],
     labels = paste("Óptimo:", num_arboles[min_error_idx]), pos = 3, col = "red")
```



Podemos ver que el numero de arboles optimo es de 480 arboles, por lo que usaremos este arbol para poder entrenar nuestro modelo.

Comparacion ambos algoritmos Para esto predecimos en nuestro modelo de random forest y calculamos el f1score.

```
library(randomForest)
library(caret)


m1 <- randomForest(
  Categoria ~ .,       # Fórmula
```

```
  data    = train_set_clean,  # Datos de entrenamiento
  ntree   = 7,       # Número de árboles
  mtry    = 4          # Número de variables seleccionadas aleatoriamente
)


prediccionesrf <- predict(m1, test_data_clean)


conf_matrixrf <- confusionMatrix(prediccionesrf, test_data_clean$Categoria)


accuracyrf <- conf_matrixrf$overall["Accuracy"]


precision <- conf_matrixrf$byClass[, "Pos Pred Value"]
recall <- conf_matrixrf$byClass[, "Sensitivity"]


f1_scorerf <- 2 * ((precision * recall) / (precision + recall))


cat("\nAccuracy del modelo:", accuracyrf, "\n")
```

```
##
## Accuracy del modelo: 0.9816934
```

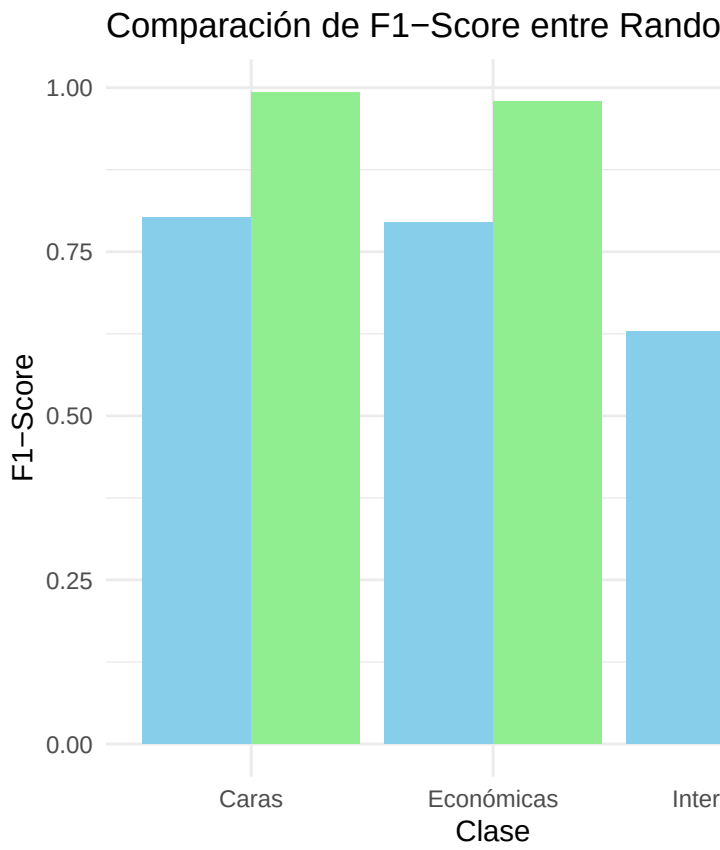
```
cat("\nF1-Score por clase:", f1_scorerf, "\n")
```

```
##
## F1-Score por clase: 0.9795918 0.9716312 0.9932886
```

Aqui tuve que bajarle a los hiperparametros como lo es el numero de arboles generados para generar una prediccion lo suficientemente no sobreajustada. Aqui vemos que nuestro modelo mejoro muchisimo a tener un acurrancy de 0.98 y un F1 score de

```


clases <- c("Económicas", "Intermedias", "Caras")


f1_scores_df <- data.frame(
  Clase = rep(clases, 2),
  F1_Score = c(f1_scorerf, f1_score),
  Modelo = rep(c("Random Forest", "Árbol de Clasificación"), each = 3)
)

# Graficar los F1-Scores
ggplot(f1_scores_df, aes(x = Clase, y = F1_Score, fill = Modelo)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
```

```
    title = "Comparación de F1-Score entre Random Forest y Árbol de Clasificación",
    y = "F1-Score",
    x = "Clase"
  ) +
  theme_minimal() +
  scale_fill_manual(values = c("skyblue", "lightgreen"))
```



Comparacion Random Fores y Arbol de Clasificacion

Aqui podemos ver que nuestro mejor arbol fue el de random forest, Siendo el que tiene casi un 1 en f1 score mientras que el arbol de clasificacion esta entre 0.69 y 0.75. En ambos modelos se ve que el caras siempre va a predecir mejor , y el de intermedias lo va a calcular peor, aun asi en el modelo del random forest con los hipermarametros de variables 4 y numeros de arboles 7, se desempeño mucho mejor que el de clasificacion normal, lo que indica que nuestro modelo es bastante competente a comparacion de este ultimo.

Pero en retrospectiva seria mejor bajar algunos parametros del random forest ya que si bien tiene un buen f1 score tiende al sobreajuste, y mejorar algunos del de clasificacion, ya que tienden al subajuste.

Conclusion

En conclusion nuestro mejor modelo fue el de random forest, muy posiblemente debido a que esta haciendo una seleccion aleatoria de variables y multiples veces a diferencia del de clasificacion comun que solo lo hace 1 vez, y ademas no es necesario modificar la profunidad de este ultimo, sin embargo vemos que el de clasificacion es mucho mas descriptivo y nos deja inferir cosas que el precio de la casa se ajusta al valor de su propiedad, vecindario, tamaño y el garage que tiene .

En relacion a otros modelos como el de regresion vemos que ha mejorado con respecto al inicial de regresion tradicional pero sigue teniendo problemas para determinar bien el precio de las casas, por lo que es

recomendable realizar otro enfoque en como se estan pasando los valores de estas o usar mayor profundidad.