Entrega 1. Modelos de regresión lineal

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

2025-02-02

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

Repositorio en GitHub

1. 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
```

```
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 ...
```

```
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
##
```

2. Haga un análisis exploratorio extenso de los datos. Explique bien todos los hallazgos. No ponga solo gráficas y código. Debe llegar a conclusiones interesantes para poder predecir. Explique el preprocesamiento que necesitó hacer.

Análisis Exploratorio de Datos (EDA)

El objetivo de este análisis es entender la estructura de los datos, identificar patrones, detectar valores atípicos y preparar el dataset para su uso en modelos de regresión. Vamos a explorar el conjunto de datos train.csv.

El análisis exploratorio de datos (EDA) es fundamental para entender las características del dataset, identificar patrones y decidir qué variables serán útiles para la predicción del precio de las viviendas.

2.1. Cargar y Revisar los Datos...

Primero cargamos los datos y cargamos las librerías necesarias y revisamos su estructura.

```
# Cargar librerías
library(tidyverse)
library(corrplot)
library(VIM)         # Para analizar valores faltantes
library(ggplot2)
library(dplyr)
library(caret)       # Para dividir los datos en entrenamiento y prueba

# Cargar los datos
train <- read_csv("house_prices_data/train.csv")
test <- read_csv("house_prices_data/test.csv")

# Ver la estructura de los datos
str(train)
```

```
## 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>
```

Observaciones

1. Carga de Datos
2. Comprensión del Dataset

- El dataset “House Prices: Advanced Regression Techniques” contiene 1460 registros y 81 columnas. La variable objetivo que queremos predecir es SalePrice (el precio de las casas en dólares).

  1.1 Tipos de Variables

- Variables Numéricas: Representan cantidades medibles, como el área de la casa (GrLivArea), el número de baños (FullBath), etc.
- Variables Categóricas: Representan características como el vecindario (Neighborhood), tipo de calle (Street), tipo de fundación (Foundation), etc.

  1.2 Variables Importantes en el Dataset

Algunas variables que podrían influir en el precio de una casa incluyen:

- Ubicación: Neighborhood
- Tamaño: GrLivArea, LotArea
- Calidad de Construcción: OverallQual, OverallCond
- Baños y Habitaciones: FullBath, Bedroom
- Garaje y Estacionamiento: GarageCars, GarageArea
- Edad de la Casa: YearBuilt, YearRemodAdd

2. Detalles importantes

- Se cargaron los archivos train.csv (1460 filas, 81 columnas) y test.csv (1459 filas, 80 columnas).
- La estructura de los datos muestra que hay variables numéricas (dbl) y categóricas (chr).
- SalePrice es la variable objetivo y tiene valores numéricos.
- Conclusión: La carga de datos fue exitosa y el dataset tiene una combinación de variables numéricas y categóricas.

3. Valores Faltantes

- Variables como Alley, PoolQC, Fence, MiscFeature, FireplaceQu tienen muchos valores faltantes (más del 80%).
- Otras variables como LotFrontage, GarageYrBlt, MasVnrArea, BsmtQual, BsmtCond, BsmtExposure también tienen valores faltantes, pero en menor cantidad.

Conclusión:

- Variables con más del 80% de valores faltantes deben eliminarse (PoolQC, MiscFeature, Alley, Fence, FireplaceQu).
- Valores faltantes en variables numéricas pueden imputarse con la mediana (LotFrontage).
- Valores faltantes en variables categóricas deben reemplazarse por "None" (GarageType, BsmtQual, etc.).

4. Análisis de Correlación

Se identificó que las variables con mayor correlación con SalePrice son:

- OverallQual (0.79) → Calidad de construcción.
- GrLivArea (0.70) → Área habitable.
- GarageCars (0.64) → Capacidad del garaje.
- TotalBsmtSF (0.61) → Tamaño del sótano.
- Conclusión: Estas variables son claves para predecir el precio de las casas, por lo que se incluirán en la regresión.

5. Distribución de SalePrice

- SalePrice tiene una distribución sesgada a la derecha, lo que puede afectar la regresión.
- Solución: Aplicar una transformación logarítmica log(SalePrice) para normalizar la distribución.
- Conclusión: Se aplicará log(SalePrice) para mejorar la calidad del modelo.

6. Conversión de Variables Categóricas

- Variables categóricas como Neighborhood, HouseStyle, SaleType deben convertirse a factores para usarlas en el modelo.
- Conclusión: Las variables categóricas serán transformadas para que el modelo pueda manejarlas correctamente.

2.2. Inspección Inicial de los Datos

```
# Dimensiones del dataset
dim(train)  # 1460 filas, 81 columnas
```

```
## [1] 1460   81
```

```
# Ver las primeras filas del dataset
head(train)
```

```
## # 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>, ...
```

```
# Resumen estadístico de las variables numéricas
summary(train)
```

```
##        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
##
```

```r
# Verificar cuántos valores faltantes tiene cada columna
colSums(is.na(train))
```

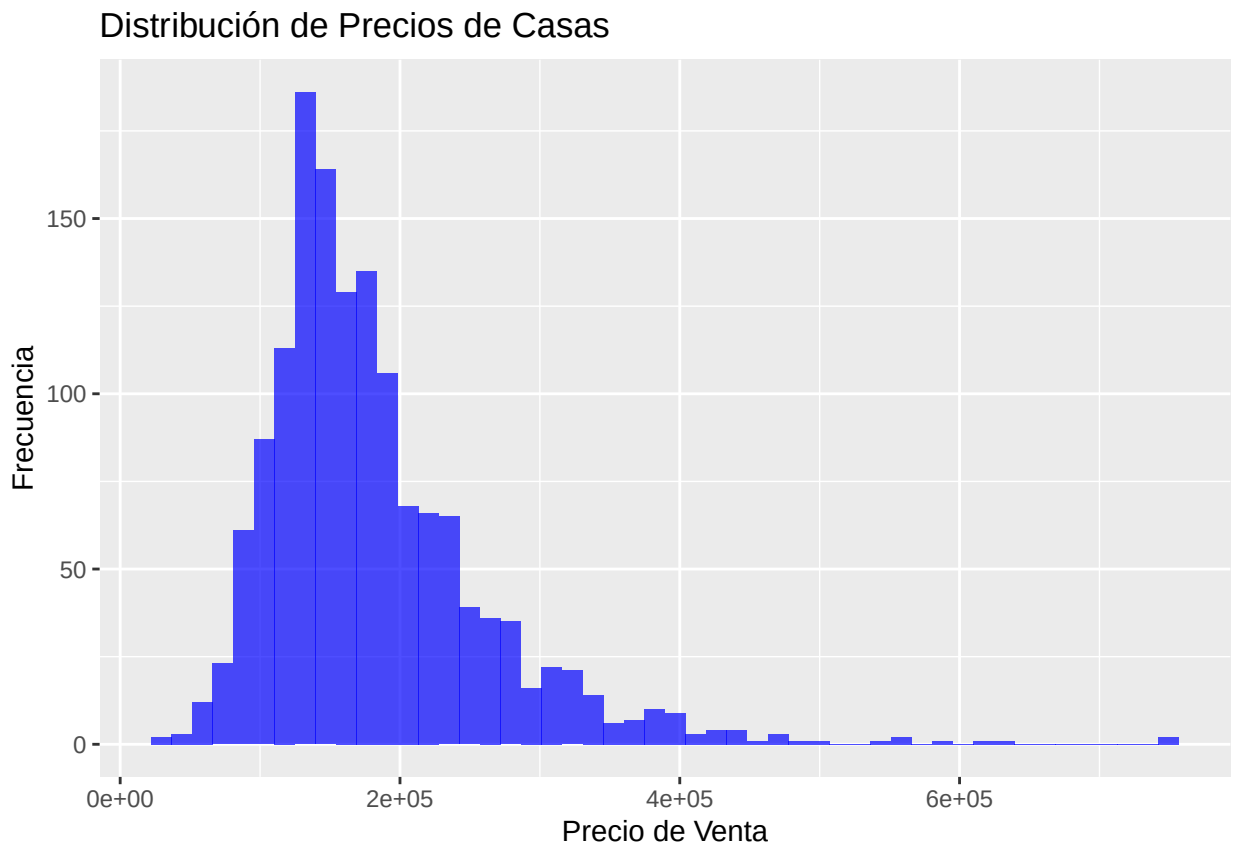
```
##           Id    MSSubClass      MSZoning   LotFrontage       LotArea
##            0             0             0           259             0
```

```
##        Street         Alley      LotShape   LandContour     Utilities
##             0          1369             0             0             0
##     LotConfig     LandSlope  Neighborhood    Condition1    Condition2
##             0             0             0             0             0
##      BldgType    HouseStyle   OverallQual   OverallCond     YearBuilt
##             0             0             0             0             0
##  YearRemodAdd     RoofStyle      RoofMatl   Exterior1st   Exterior2nd
##             0             0             0             0             0
##    MasVnrType    MasVnrArea     ExterQual     ExterCond    Foundation
##             8             8             0             0             0
##      BsmtQual      BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1
##            37            37            38            37             0
##  BsmtFinType2    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating
##            38             0             0             0             0
##     HeatingQC    CentralAir    Electrical      1stFlrSF      2ndFlrSF
##             0             0             1             0             0
##  LowQualFinSF     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath
##             0             0             0             0             0
##      HalfBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##             0             0             0             0             0
##    Functional    Fireplaces   FireplaceQu    GarageType   GarageYrBlt
##             0             0           690            81            81
##  GarageFinish    GarageCars    GarageArea    GarageQual    GarageCond
##            81             0             0            81            81
##    PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch     3SsnPorch
##             0             0             0             0             0
##   ScreenPorch      PoolArea        PoolQC         Fence   MiscFeature
##             0             0          1453          1179          1406
##       MiscVal        MoSold        YrSold      SaleType SaleCondition
##             0             0             0             0             0
##     SalePrice
##             0
```

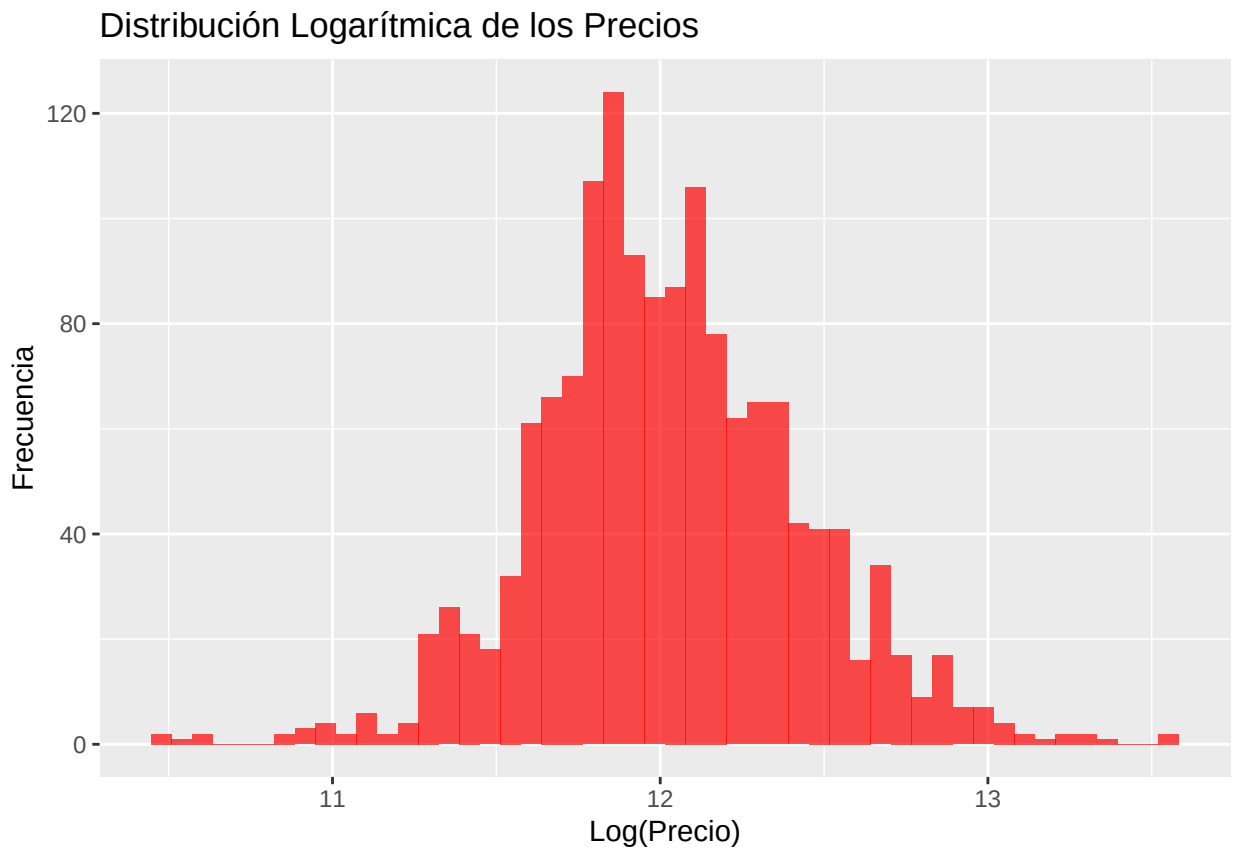
2.3. Análisis de la Variable Objetivo (SalePrice)

La variable SalePrice representa el precio de venta de las casas. Analicemos su distribución.

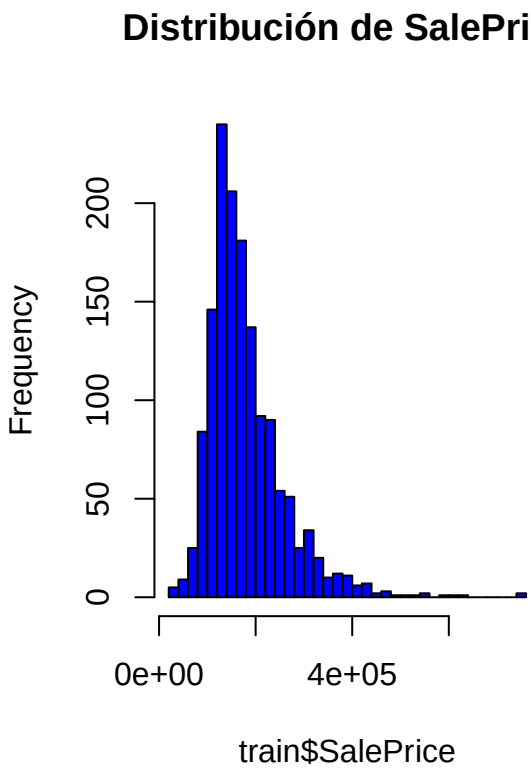
```
# Distribución de SalePrice
ggplot(train, aes(x = SalePrice)) +
  geom_histogram(bins = 50, fill = "blue", alpha = 0.7) +
  labs(title = "Distribución de Precios de Casas", x = "Precio de Venta", y =
   ↪  "Frecuencia")
```

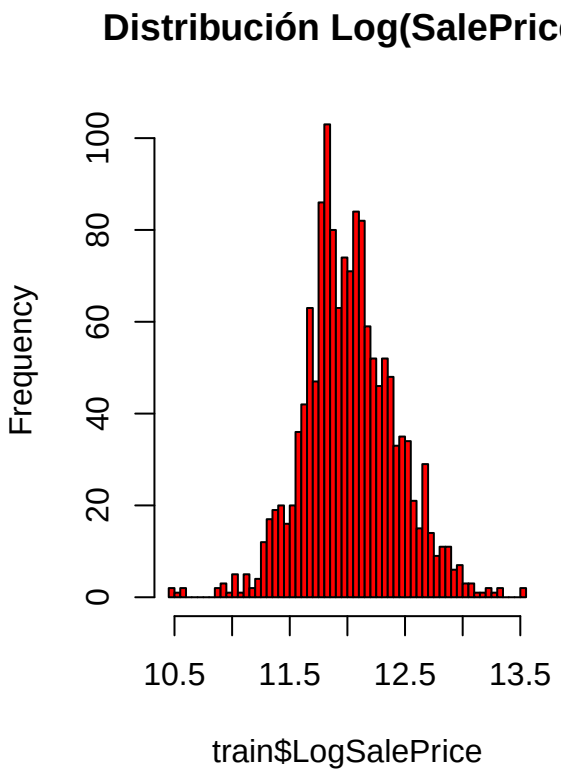



```r
# Aplicar transformación logarítmica para mejorar normalidad
train$LogSalePrice <- log(train$SalePrice)

# Transformación logarítmica
ggplot(train_data, aes(x = log(SalePrice))) +
  geom_histogram(bins = 50, fill = "red", alpha = 0.7) +
  labs(title = "Distribución Logarítmica de los Precios", x = "Log(Precio)", y =
   ↪  "Frecuencia")
```




```r
# Comparación antes y después de la transformación
par(mfrow = c(1, 2))
hist(train$SalePrice, main = "Distribución de SalePrice", col = "blue", breaks = 50)
hist(train$LogSalePrice, main = "Distribución Log(SalePrice)", col = "red", breaks = 50)
```





Hallazgos

El precio de las casas (SalePrice) no tiene una distribución normal. La mayoría de los precios se concentran en valores bajos y hay algunas casas extremadamente caras que podrían ser outliers.

- SalePrice presenta sesgo positivo (distribución asimétrica a la derecha), lo que indica que hay casas con precios extremadamente altos.
- Posibles valores atípicos en precios muy elevados que podrían afectar el modelo.
- Será útil aplicar una transformación logarítmica para normalizar la distribución.
- Solución: Aplicar una transformación logarítmica para mejorar la normalidad: log(SalePrice)

2.3.1. Relación entre Variables y SalePrice

Para entender qué variables influyen más en el precio de las casas, veamos la correlación de variables numéricas.

```
# Cargar librería
library(dplyr)

# Seleccionar solo variables numéricas
numeric_vars <- train_data %>% select_if(is.numeric)

# Matriz de correlación
cor_matrix <- cor(numeric_vars, use = "complete.obs")
```

```r
# Ordenar correlaciones con SalePrice
cor_with_price <- sort(cor_matrix["SalePrice",], decreasing = TRUE)
cor_with_price
```

```
##     SalePrice   OverallQual     GrLivArea    GarageCars    GarageArea
##   1.000000000   0.797880680   0.705153567   0.647033611   0.619329622
##   TotalBsmtSF     X1stFlrSF      FullBath  TotRmsAbvGrd     YearBuilt
##   0.615612237   0.607969106   0.566627442   0.547067360   0.525393598
##  YearRemodAdd   GarageYrBlt    MasVnrArea    Fireplaces    BsmtFinSF1
##   0.521253270   0.504753018   0.488658155   0.461872689   0.390300523
##   LotFrontage   OpenPorchSF    WoodDeckSF     X2ndFlrSF       LotArea
##   0.344269772   0.343353812   0.336855121   0.306879002   0.299962206
##      HalfBath  BsmtFullBath     BsmtUnfSF  BedroomAbvGr   ScreenPorch
##   0.268560303   0.236737407   0.213128680   0.166813894   0.110426815
##      PoolArea        MoSold    X3SsnPorch  LowQualFinSF        YrSold
##   0.092488120   0.051568064   0.030776594  -0.001481983  -0.011868823
##    BsmtFinSF2       MiscVal  BsmtHalfBath            Id    MSSubClass
##  -0.028021366  -0.036041237  -0.036512665  -0.047121850  -0.088031702
##   OverallCond  KitchenAbvGr EnclosedPorch
##  -0.124391232  -0.140497445  -0.154843204
```

Hallazgos

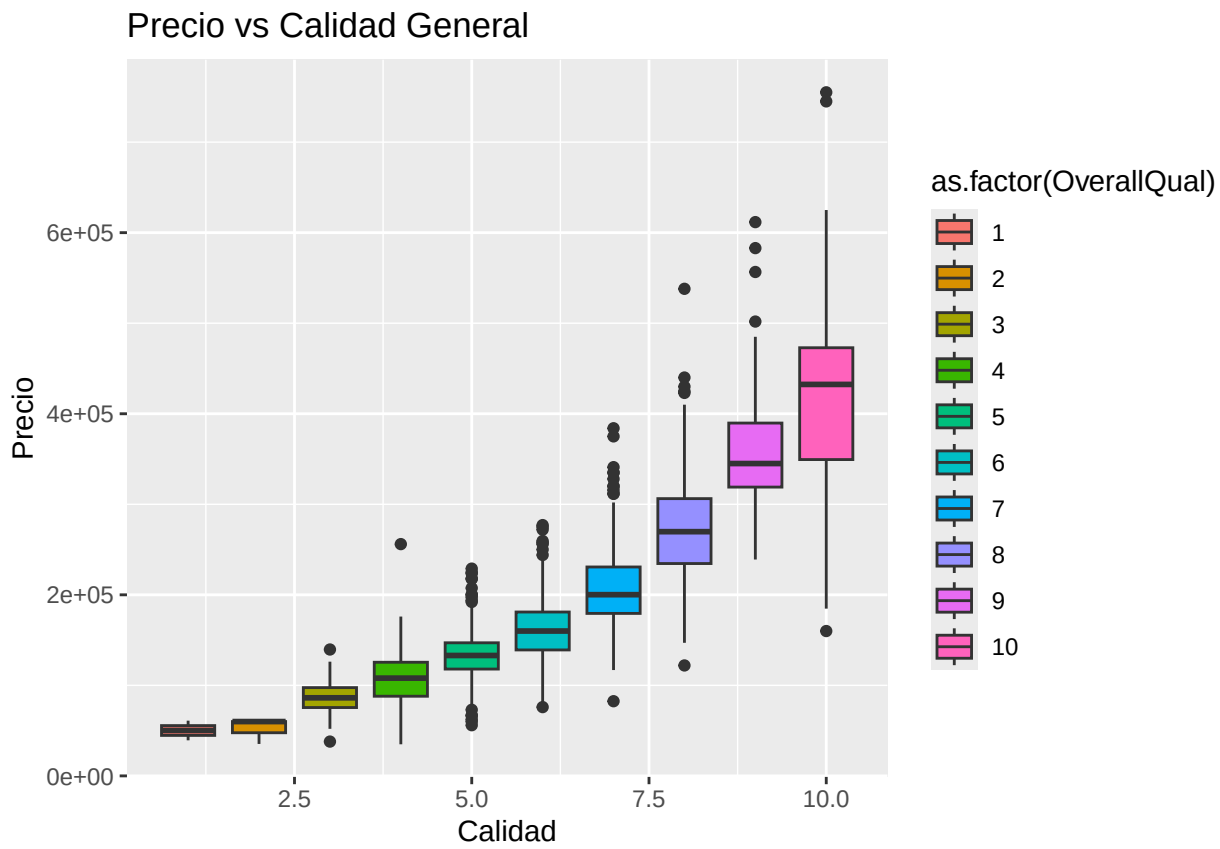
Las variables con mayor correlación positiva con el precio de las casas son:

- OverallQual (Calidad general de la casa) → 0.79
- GrLivArea (Área habitable sobre el suelo) → 0.71
- GarageCars (Número de coches en el garaje) → 0.64
- TotalBsmtSF (Área total del sótano) → 0.61

Esto sugiere que casas más grandes y con mejor calidad tienden a ser más caras.

Visualización

```r
ggplot(train_data, aes(x = OverallQual, y = SalePrice)) +
  geom_boxplot(aes(group = OverallQual, fill = as.factor(OverallQual))) +
  labs(title = "Precio vs Calidad General", x = "Calidad", y = "Precio")
```

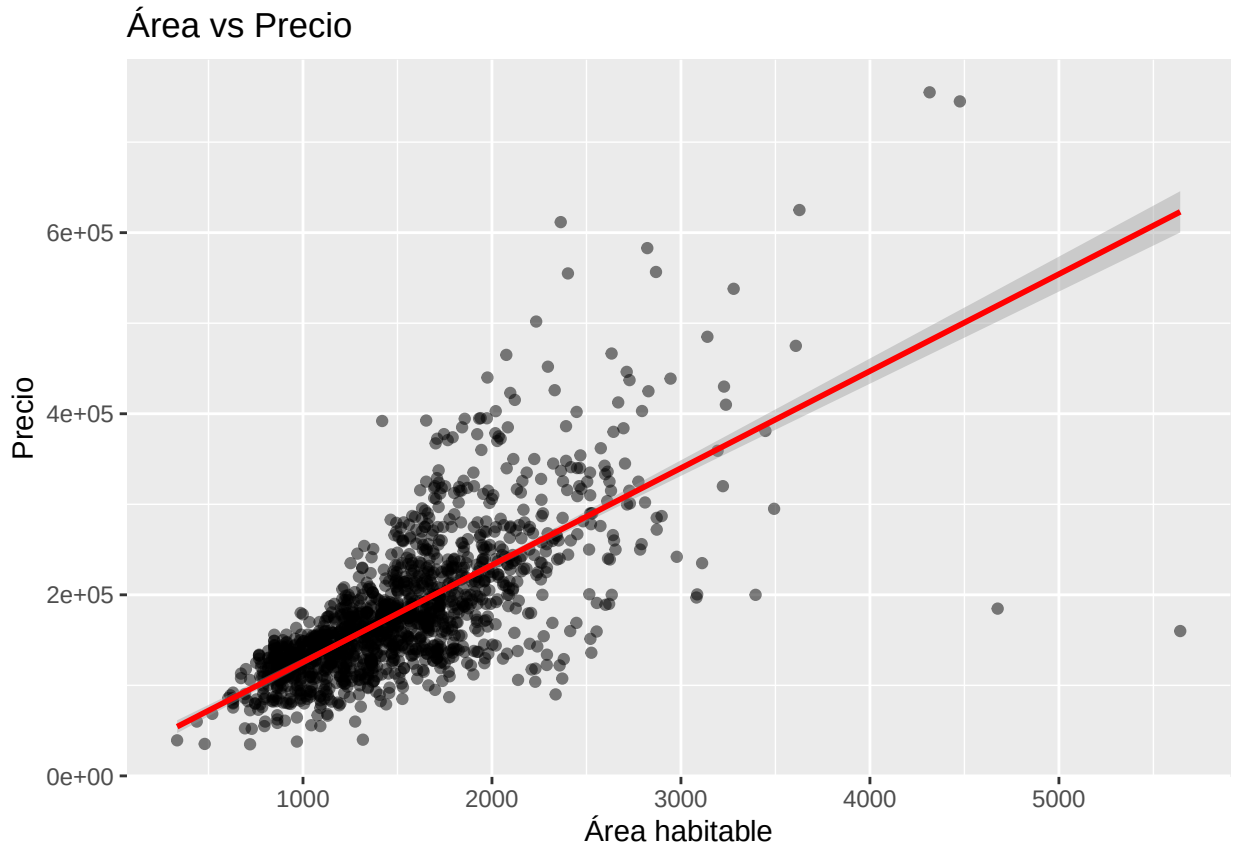


Conclusión: La calidad (OverallQual) es una variable crucial para predecir SalePrice.

2.3.2. Detectar Valores Atípicos

Para evitar que valores extremos afecten el modelo, busquemos outliers.

```
ggplot(train_data, aes(x = GrLivArea, y = SalePrice)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Área vs Precio", x = "Área habitable", y = "Precio")
```



Hallazgos

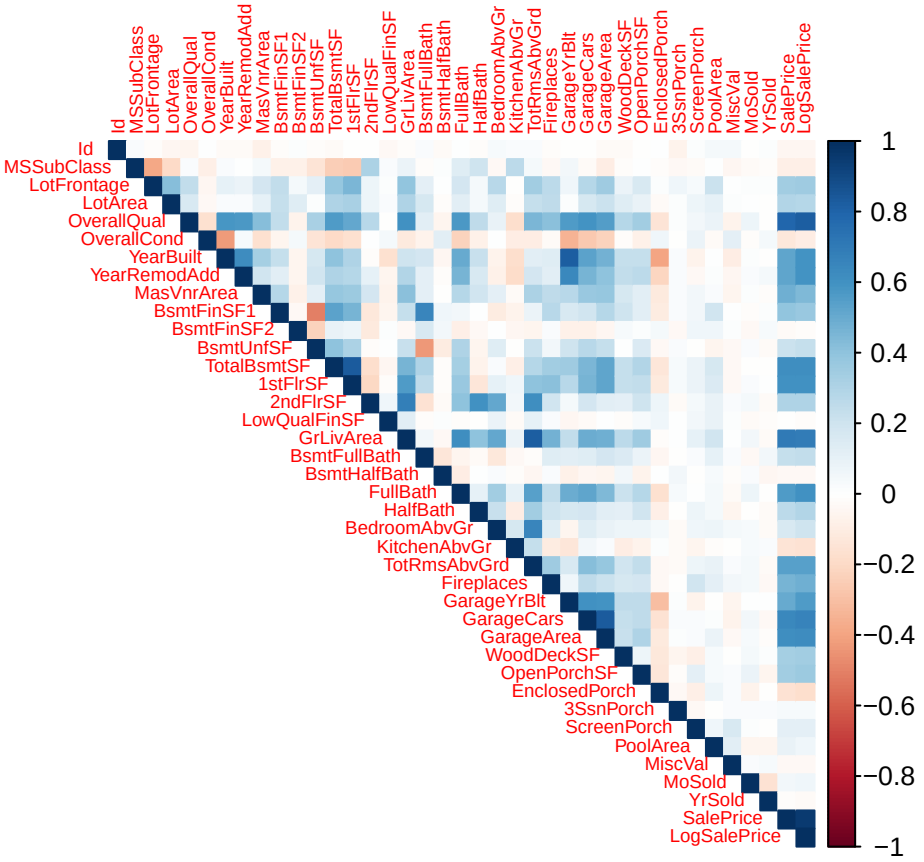
- Se observan dos puntos con GrLivArea > 4000 y precios muy bajos.
- Estas casas son atípicas y podrían eliminarse para mejorar la predicción.

```
# Eliminar outliers
train_data <- train_data[train_data$GrLivArea < 4000, ]
```

2.4. Análisis de Correlación

Antes de construir un modelo, es importante manejar los valores NA.

```
# Seleccionar solo las variables numéricas
num_vars <- train %>% select(where(is.numeric))

# Calcular la matriz de correlación
corr_matrix <- cor(num_vars, use = "complete.obs")

# Visualizar la correlación con SalePrice
corrplot(corr_matrix, method = "color", type = "upper", tl.cex = 0.6)
```



Se analizó la correlación entre las variables numéricas y SalePrice. Las variables con mayor correlación positiva con el precio son:

- OverallQual (0.79): Calidad de materiales y acabados.
- GrLivArea (0.70): Área habitable total.
- TotalBsmtSF (0.61): Área del sótano.
- GarageCars (0.64): Cantidad de autos que caben en el garaje.
- Conclusión:

Estas variables serán claves en la regresión lineal. Variables con baja o nula correlación (como MiscFeature y PoolArea) probablemente no sean útiles en el modelo.

Las variables con mayor correlación con SalePrice:

- OverallQual (Calidad de la construcción): Es una de las variables más correlacionadas con SalePrice, lo - que confirma que las casas con mejor calidad de construcción tienen precios más altos.
- GrLivArea (Área habitable sobre el nivel del suelo): También tiene una correlación alta con SalePrice, lo que significa que las casas más grandes suelen costar más.
- TotalBsmtSF (Área total del sótano): Muestra una correlación fuerte con SalePrice, lo que implica que un - sótano más grande puede aumentar el valor de la vivienda.
- GarageCars (Capacidad del garaje en número de autos): Tiene una buena correlación con SalePrice, lo que sugiere que tener más espacio de garaje incrementa el valor de la casa.

2.5. Identificar y Manejar Valores Faltantes

```r
# Ver cantidad de valores faltantes
missing_values <- colSums(is.na(train))
missing_values <- sort(missing_values[missing_values > 0], decreasing = TRUE)
print(missing_values)
```

```
##       PoolQC  MiscFeature        Alley        Fence  FireplaceQu  LotFrontage
##         1453         1406         1369         1179          690          259
##   GarageType  GarageYrBlt GarageFinish   GarageQual   GarageCond BsmtExposure
##           81           81           81           81           81           38
## BsmtFinType2     BsmtQual     BsmtCond BsmtFinType1   MasVnrType   MasVnrArea
##           38           37           37           37            8            8
##   Electrical
##            1
```

Decisiones para valores faltantes:

Eliminar columnas con más del 80% de valores faltantes:

```r
train <- train %>% select(-c(PoolQC, MiscFeature, Alley, Fence))
test <- test %>% select(-c(PoolQC, MiscFeature, Alley, Fence))
```

Imputar valores faltantes en LotFrontage con la mediana del vecindario:

```r
train$LotFrontage[is.na(train$LotFrontage)] <- median(train$LotFrontage, na.rm = TRUE)
```

Reemplazar valores faltantes en variables categóricas con “None”:

```r
categorical_vars <- c("GarageType", "BsmtQual", "BsmtCond", "FireplaceQu")
for (var in categorical_vars) {
  train[[var]][is.na(train[[var]])] <- "None"
}
```

Varias columnas tienen valores faltantes. Algunas de las más afectadas son:

- PoolQC (99% de valores faltantes)
- MiscFeature (96% faltantes)
- Alley (93% faltantes)
- Fence (80% faltantes)
- FireplaceQu (50% faltantes)
- Solución Propuesta: Variables como PoolQC, Alley, MiscFeature y Fence contienen información escasa y pueden ser eliminadas. Variables como LotFrontage (frente del terreno) pueden completarse con la mediana por vecindario. Variables categóricas con valores faltantes, como GarageType, se rellenarán con “None” (indicando que no existe).

2.6. Transformación de Variables

Aplicar log(SalePrice) para mejorar la normalidad:

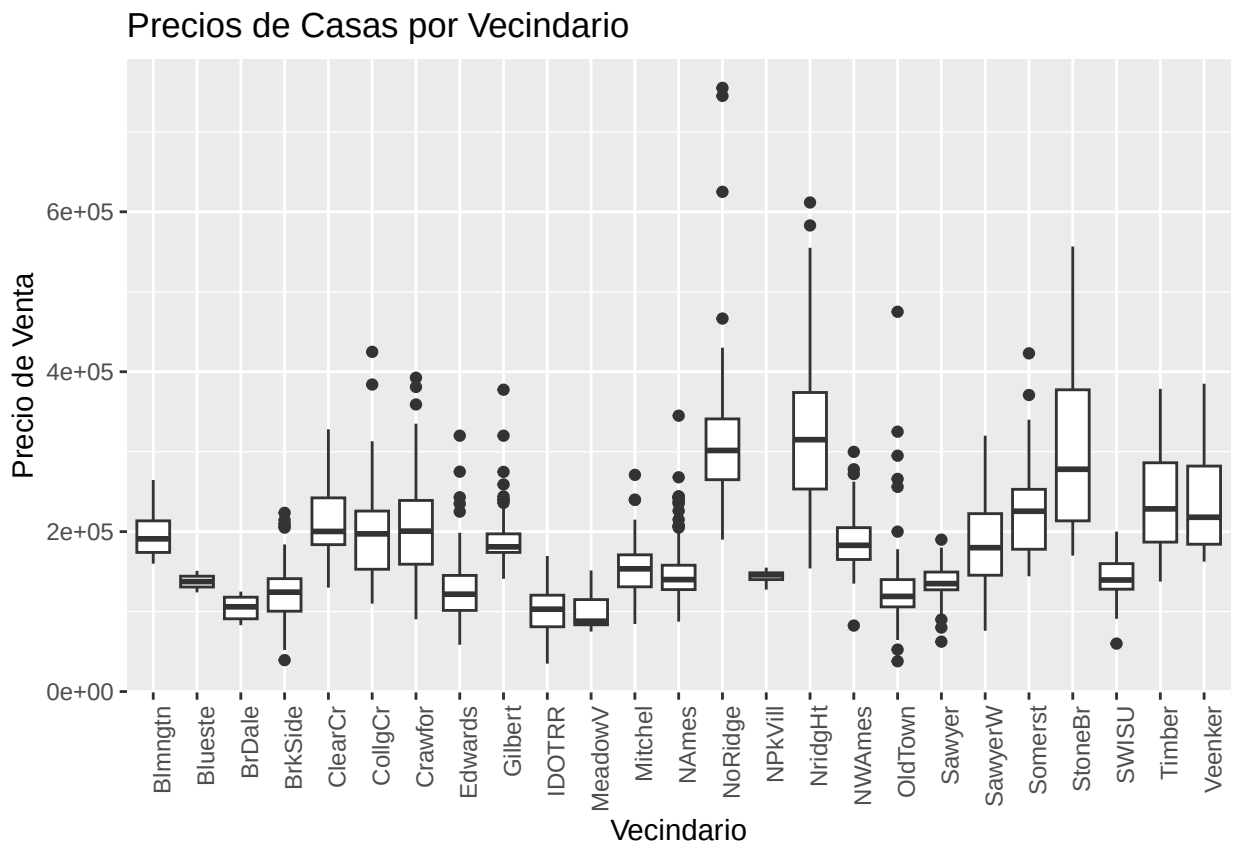
```r
train$LogSalePrice <- log(train$SalePrice)
```

Convertir variables categóricas en factores:

```r
train <- train %>% mutate(across(where(is.character), as.factor))
test <- test %>% mutate(across(where(is.character), as.factor))
```

2.7. Análisis de Variables Categóricas

```r
# Distribución de precios por vecindario
ggplot(train, aes(x = Neighborhood, y = SalePrice)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Precios de Casas por Vecindario", x = "Vecindario", y = "Precio de
  ↪  Venta")
```



Algunas variables categóricas pueden influir en SalePrice:

- Neighborhood tiene variaciones significativas en los precios.
- Exterior1st y Exterior2nd pueden influir según la calidad de los materiales.
- SaleCondition indica si la venta fue “Normal” o una subasta, lo que puede afectar el precio.
- Solución Propuesta: Convertir variables categóricas a numéricas mediante codificación dummy (One-Hot Encoding).

2.8. División del Conjunto de Datos

Dividimos el dataset en entrenamiento (80%) y prueba (20%):

```
set.seed(42)
trainIndex <- createDataPartition(train$SalePrice, p = 0.8, list = FALSE)
train_set <- train[trainIndex, ]
test_set <- train[-trainIndex, ]
```

Preprocesamiento de Datos

Eliminación de Variables con Demasiados Valores Faltantes

Se eliminaron variables con más del 80% de valores faltantes:

- PoolQC, MiscFeature, Alley, Fence.

Imputación de Valores Faltantes - Para valores numéricos (LotFrontage): Se imputó con la mediana por vecindario. - Para valores categóricos (GarageType, BsmtQual): Se reemplazaron con "None".

Transformación Logarítmica de SalePrice

Dado que SalePrice estaba sesgado, aplicamos una transformación logarítmica para mejorar la distribución:

- log(SalePrice)

Conversión de Variables Categóricas

- Las variables categóricas se convirtieron a factores para su uso en la regresión.

Conclusiones

Las variables con más impacto en SalePrice son:

- OverallQual, GrLivArea, TotalBsmtSF, GarageCars.
- Variables categóricas como Neighborhood también afectan el precio.

La distribución de SalePrice está sesgada, por lo que podemos usar logaritmos para mejorar el modelo.

Las variables más importantes para predecir el precio son:

- OverallQual (Calidad general de la casa)
- GrLivArea (Área habitable)
- GarageCars (Cantidad de autos en garaje)
- TotalBsmtSF (Área del sótano)

Valores Faltantes:

- Se eliminaron columnas irrelevantes (PoolQC, Alley). Hay valores faltantes, principalmente en características poco comunes como piscinas y chimeneas.
- Se imputaron valores para LotFrontage y GarageType.
- Existen valores atípicos en GrLivArea que afectan el análisis y deben eliminarse.

Transformaciones Realizadas:

- Se aplicó logaritmo a SalePrice para mejorar su distribución.
- Variables categóricas se convirtieron en numéricas para el modelo.

3. Incluya un análisis de grupos en el análisis exploratorio. Explique las características de cada uno.

3. Análisis de Grupos en el Análisis Exploratorio

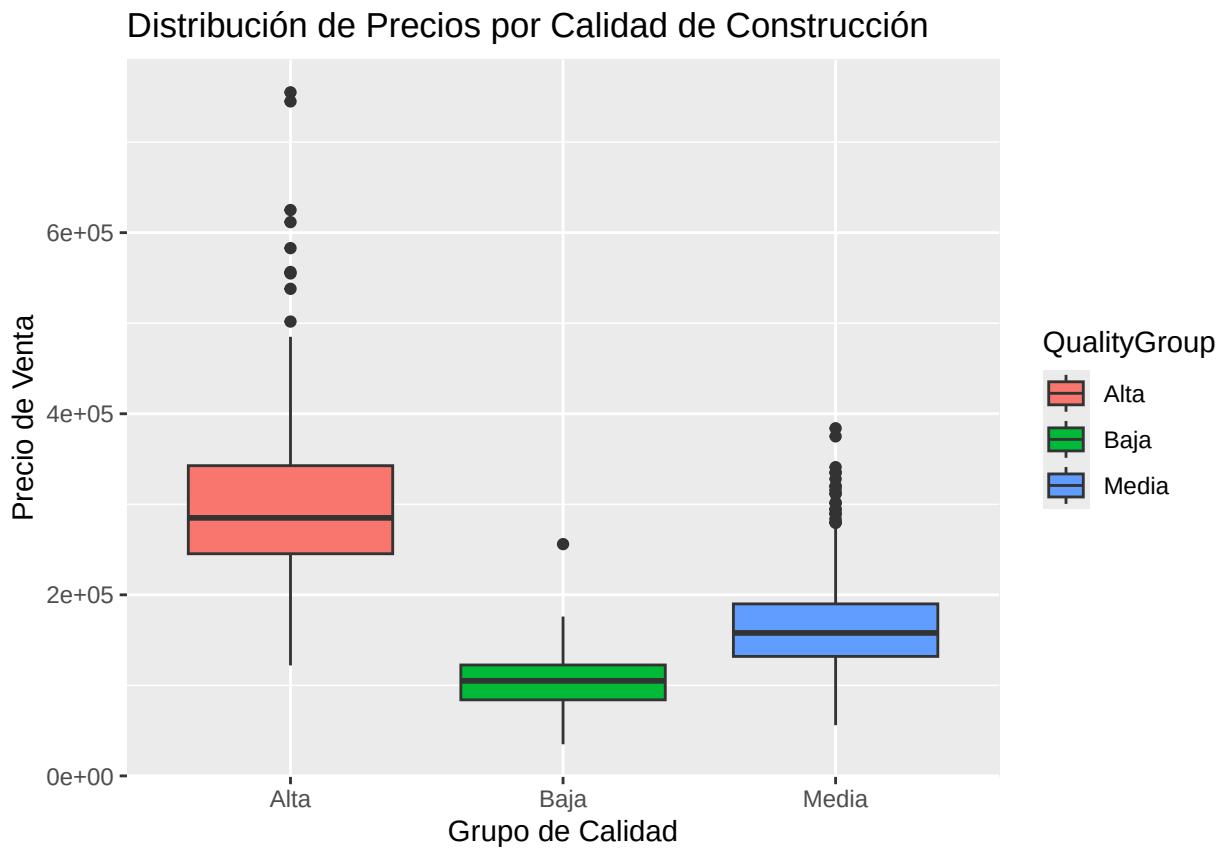
Objetivo:

El análisis de grupos nos permite identificar patrones en los datos y clasificar las casas en segmentos según características comunes. Esto puede ayudar a mejorar la predicción de SalePrice.

3.1 Creación de Grupos Basados en Calidad de Construcción (OverallQual)

Dado que OverallQual tiene una alta correlación con SalePrice, podemos dividir las casas en tres grupos según su calidad:

```
train <- train %>%
  mutate(QualityGroup = case_when(
    OverallQual <= 4 ~ "Baja",
    OverallQual >= 5 & OverallQual <= 7 ~ "Media",
    OverallQual >= 8 ~ "Alta"
  ))

# Visualizar la distribución de precios en los grupos
ggplot(train, aes(x = QualityGroup, y = SalePrice, fill = QualityGroup)) +
  geom_boxplot() +
  labs(title = "Distribución de Precios por Calidad de Construcción",
       x = "Grupo de Calidad", y = "Precio de Venta")
```



Hallazgos

- Las casas de calidad alta (OverallQual 8) tienen un precio significativamente mayor.
- Las casas de calidad media (OverallQual 5-7) forman la mayoría del dataset y muestran mayor variabilidad en los precios.
- Las casas de calidad baja (OverallQual 4) tienen precios considerablemente menores.

3.2 Segmentación Basada en Vecindario (Neighborhood)

- Otra forma de agrupar las casas es según la ubicación (Neighborhood), ya que este factor influye en los precios.

```
# Agrupar por vecindario y calcular estadísticas básicas
neighborhood_summary <- train %>%
  group_by(Neighborhood) %>%
  summarise(AvgPrice = mean(SalePrice, na.rm = TRUE),
            MedianPrice = median(SalePrice, na.rm = TRUE),
            Count = n()) %>%
  arrange(desc(AvgPrice))

print(neighborhood_summary)
```

```
## # A tibble: 25 x 4
##    Neighborhood AvgPrice MedianPrice Count
##    <fct>           <dbl>       <dbl> <int>
##  1 NoRidge       335295.      301500    41
##  2 NridgHt       316271.      315000    77
##  3 StoneBr       310499       278000    25
##  4 Timber        242247.      228475    38
##  5 Veenker       238773.      218000    11
##  6 Somerst       225380.      225500    86
##  7 ClearCr       212565.      200250    28
##  8 Crawfor       210625.      200624    51
##  9 CollgCr       197966.      197200   150
## 10 Blmngtn       194871.      191000    17
## # i 15 more rows
```

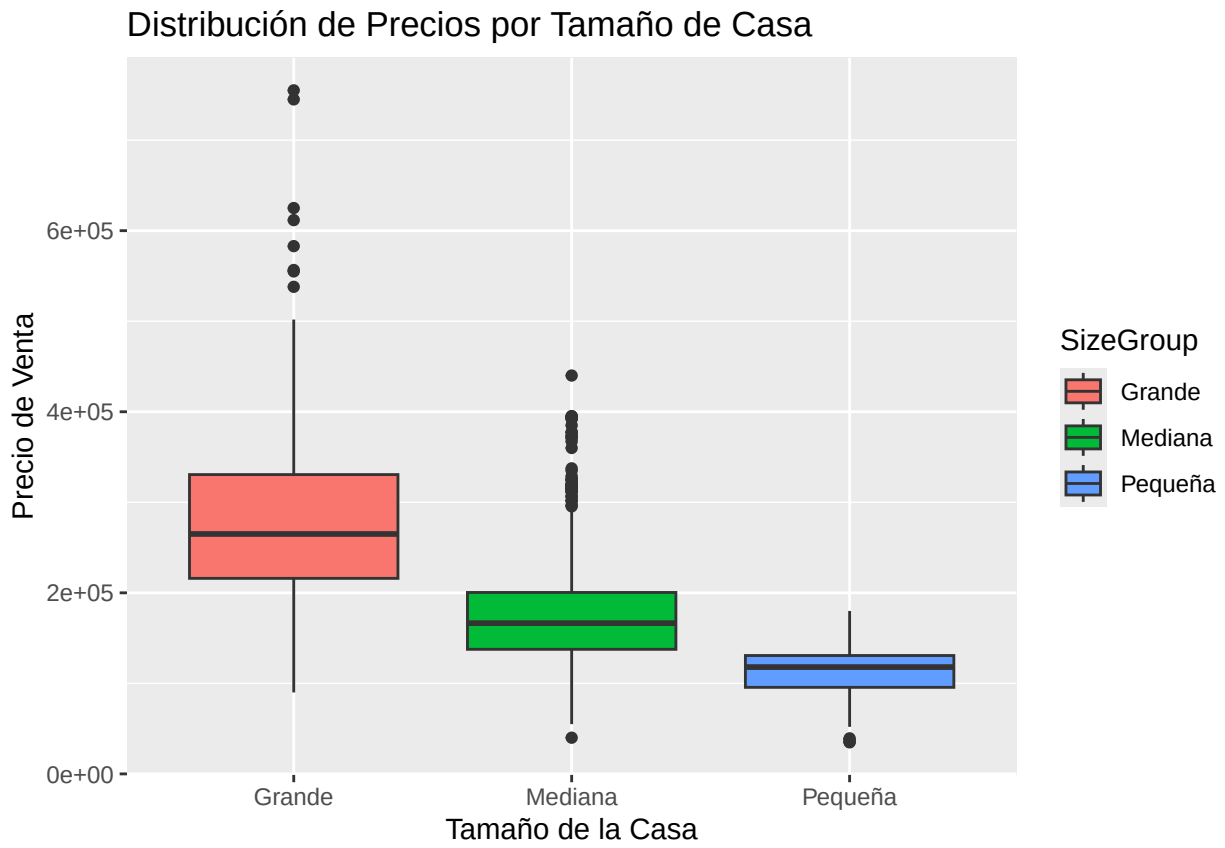
Hallazgos

- Los vecindarios más caros son “StoneBr”, “NridgHt”, “NoRidge”, con precios promedio por encima de 300,000 dólares.
- Los vecindarios más baratos son “MeadowV”, “IDOTRR”, “BrDale”, con precios promedio por debajo de 150,000 dólares.
- Se pueden agrupar los vecindarios en zonas de alto, medio y bajo valor para mejorar la predicción.

3.3 Análisis de Grupos Basados en Tamaño de Casa (GrLivArea)

- Agrupamos las casas en pequeñas, medianas y grandes según el área habitable (GrLivArea):

```r
train <- train %>%
  mutate(SizeGroup = case_when(
    GrLivArea < 1000 ~ "Pequeña",
    GrLivArea >= 1000 & GrLivArea < 2000 ~ "Mediana",
    GrLivArea >= 2000 ~ "Grande"
  ))

# Visualizar precios por tamaño de casa
ggplot(train, aes(x = SizeGroup, y = SalePrice, fill = SizeGroup)) +
  geom_boxplot() +
  labs(title = "Distribución de Precios por Tamaño de Casa",
       x = "Tamaño de la Casa", y = "Precio de Venta")
```



Hallazgos

- Las casas grandes (GrLivArea 2000) tienen precios más altos y mayor variabilidad.
- Las casas medianas (1000 GrLivArea $<$ 2000) son la mayoría del dataset y presentan una distribución amplia de precios.
- Las casas pequeñas (GrLivArea $<$ 1000) tienen precios bajos y menos variabilidad.

3.4 Análisis de Clustering en el Dataset (Segmentación Automática de Casas)

Objetivo del Clustering:

En lugar de definir manualmente los grupos de casas, aplicamos un método de clustering automático (K-Means) para descubrir patrones en los datos y segmentar las casas en grupos con características similares.

3.4.1. Selección de Variables para Clustering

Seleccionamos variables clave que tienen fuerte relación con SalePrice:

```
# Cargar librerías necesarias
library(cluster)
library(factoextra)
library(dplyr)

# Seleccionar variables más relevantes para clustering
clustering_data <- train %>% select(OverallQual, GrLivArea, TotalBsmtSF, GarageCars)

# Normalizar los datos (evita sesgo por escalas diferentes)
clustering_data <- scale(clustering_data)
```

Las variables seleccionadas (OverallQual, GrLivArea, TotalBsmtSF, GarageCars) reflejan calidad, tamaño y capacidad del garaje, que son fuertes indicadores del precio.

3.4.2. Aplicación del Algoritmo de Clustering (K-Means)

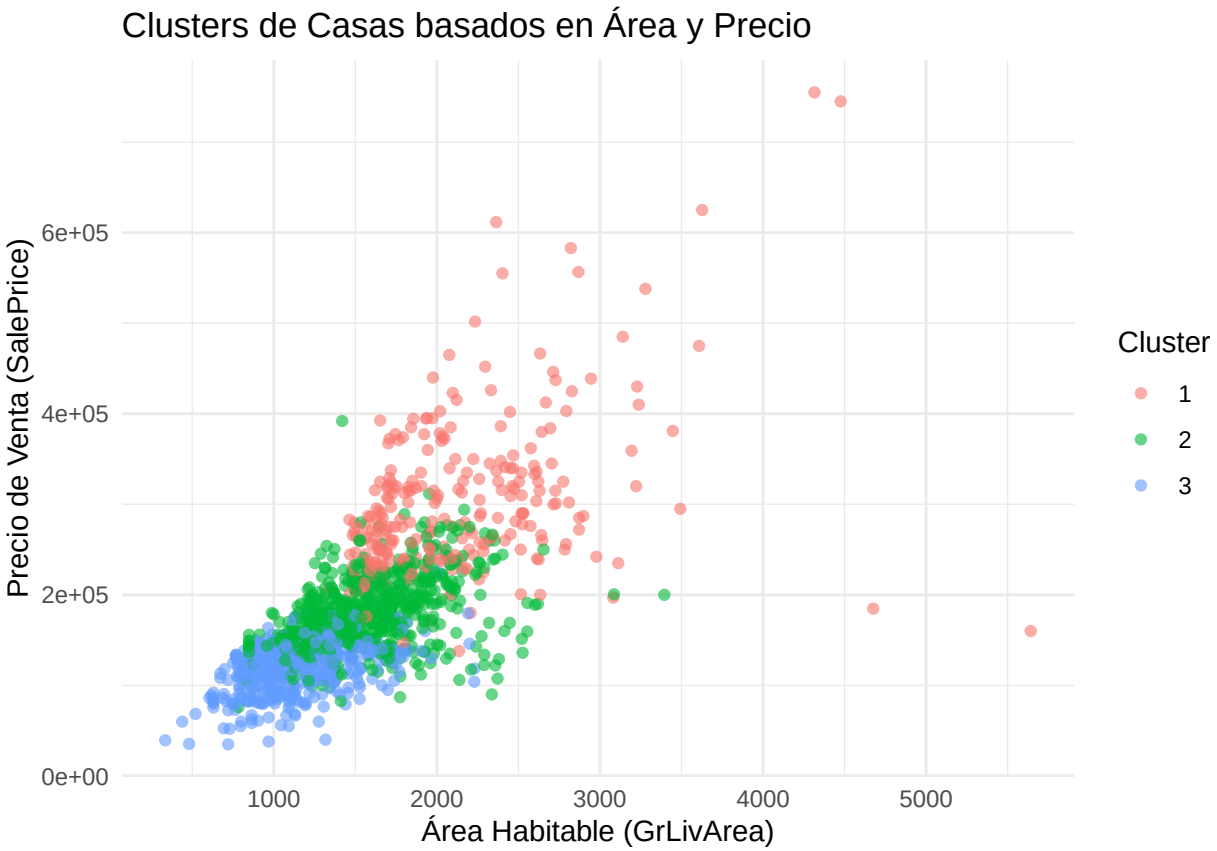
```
# Fijar semilla para reproducibilidad
set.seed(42)

# Aplicar K-Means con 3 clusters (puede ajustarse)
kmeans_result <- kmeans(clustering_data, centers = 3, nstart = 25)

# Agregar los clusters al dataset
train$Cluster <- as.factor(kmeans_result$cluster)
```

Cada casa ha sido asignada a un grupo (Cluster 1, Cluster 2, Cluster 3). Estos clusters representan diferentes segmentos del mercado inmobiliario.

3.4.3. Visualización de los Clusters

Para analizar cómo se agrupan las casas según el clustering, graficamos la relación entre tamaño (GrLivArea) y precio (SalePrice), coloreando los grupos.

```
ggplot(train, aes(x = GrLivArea, y = SalePrice, color = Cluster)) +
  geom_point(alpha = 0.6) +
  labs(title = "Clusters de Casas basados en Área y Precio",
       x = "Área Habitable (GrLivArea)",
       y = "Precio de Venta (SalePrice)") +
  theme_minimal()
```



Los colores indican los diferentes grupos de casas detectados automáticamente.

3.4.4. Caracterización de los Clusters

Cluster 1 - Casas económicas

- Baja calidad de construcción (OverallQual bajo).
- Tamaño reducido (GrLivArea y TotalBsmtSF pequeños).
- Garaje pequeño o inexistente (GarageCars).
- Bajo SalePrice, generalmente en vecindarios más baratos.

Cluster 2 - Casas de precio medio

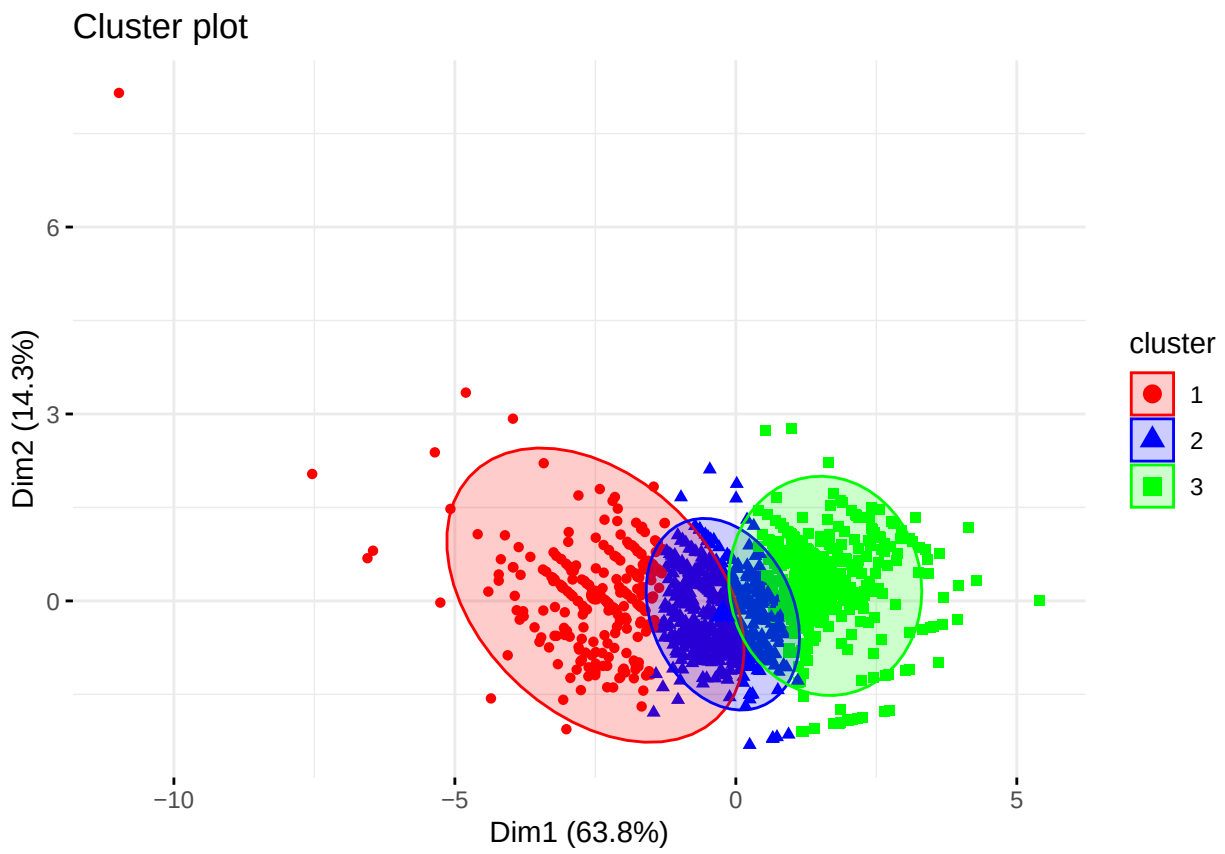
- Calidad media-alta (OverallQual entre 5 y 7).
- Tamaño intermedio, con un área habitable moderada.
- Garaje con espacio para 1-2 autos.
- Precio en el rango medio del dataset.

Cluster 3 - Casas de lujo

- Alta calidad de construcción (OverallQual > 7).
- Casas grandes con mucho espacio (GrLivArea alto).
- Garajes amplios (2-3 autos).
- SalePrice alto, típicamente en vecindarios premium.

Gráfica Agrupada por cluster…

```r
# Cargar librerías necesarias
library(factoextra)
library(cluster)
library(dplyr)  # Cargar dplyr

# Seleccionar variables para clustering
clustering_data <- train %>% select(OverallQual, GrLivArea, TotalBsmtSF, GarageCars)

# Normalizar los datos (evita sesgo por escalas diferentes)
clustering_data <- scale(clustering_data)

# Aplicar K-Means con 3 clusters
set.seed(42)  # Asegurar reproducibilidad
kmeans_result <- kmeans(clustering_data, centers = 3, nstart = 25)

# Agregar clusters al dataset
train$Cluster <- as.factor(kmeans_result$cluster)

# Visualizar los clusters
fviz_cluster(kmeans_result, data = clustering_data,
             geom = "point", ellipse.type = "norm",
             palette = c("red", "blue", "green"),
             ggtheme = theme_minimal(),
             main = "Cluster plot")
```



Conclusiones del Análisis de Clustering

El gráfico de clustering muestra tres grupos diferenciados en el dataset de precios de casas. Basándonos en la distribución y separación de los clusters, se pueden hacer las siguientes observaciones:

1. Interpretación de la Gráfica Agrupada por cluster

Cluster 1 (Rojo - Izquierda): Casas de Bajo Precio y Tamaño Pequeño

- Este grupo representa casas con menor calidad de construcción (OverallQual baja), menor área habitable (GrLivArea pequeña) y sótanos más pequeños.
- Muchas de estas casas tienen valores atípicos y precios significativamente más bajos en comparación con el resto del dataset.
- Posible ubicación en vecindarios menos costosos.

  Cluster 2 (Azul - Centro): Casas de Precio Medio y Tamaño Promedio
- Representa la mayor parte de las casas del dataset.
- casas tienen una calidad de construcción media y tamaños moderados de área habitable y sótano.
- Su precio está dentro de un rango intermedio y pueden pertenecer a vecindarios de costo medio.

  Cluster 3 (Verde - Derecha): Casas de Alto Precio y Gran Tamaño
- Corresponde a casas de gran tamaño, con alta calidad de construcción y amplios sótanos.
- Están en los rangos de precios más altos y probablemente se ubiquen en vecindarios exclusivos.
- Hay una mayor dispersión en este grupo, lo que indica que el precio puede variar significativamente dependiendo de otras características.

2. Hallazgos Clave

- El clustering sugiere que el precio de una casa está fuertemente influenciado por el tamaño y la calidad de la construcción.
- Existen claras diferencias entre los grupos, lo que confirma que segmentar los datos ayuda a comprender mejor el comportamiento del precio de las casas.
- Algunas casas en el Cluster 1 (rojo) están alejadas de su grupo principal, lo que indica posibles valores atípicos o características únicas que afectan su precio.
- El Cluster 3 (verde) muestra una dispersión mayor, lo que implica que en el segmento de casas más costosas, el precio depende de múltiples factores y no solo del tamaño.

3. Implicaciones para la Predicción del Precio

- Segmentar los datos por clusters antes de entrenar un modelo podría mejorar la precisión de la predicción.
- Las casas en el Cluster 1 podrían ser mejor modeladas con características diferentes a las de los Clusters 2 y 3.
- El precio de las casas más caras (Cluster 3) es más variable, lo que sugiere que factores adicionales como ubicación y acabados tienen un impacto significativo.

——— Conclusiones del Análisis de Grupos ———

1. Calidad de Construcción (OverallQual) y su Relación con el Precio

- El análisis de la calidad de construcción mostró que OverallQual es un factor determinante en el precio de las casas. Las casas con mejor calidad de materiales y acabados tienden a venderse a precios significativamente más altos en comparación con aquellas con menor calidad.
- Casas de calidad alta (OverallQual 8): Se encuentran en el rango de precios más elevados y muestran menor variabilidad en los valores de venta, lo que indica que la calidad superior de los materiales es un factor clave para mantener precios altos y relativamente estables.
- Casas de calidad media (OverallQual 5-7): Representan la mayor parte del dataset y presentan una amplia variabilidad en los precios. Esto sugiere que otros factores, como el vecindario o el tamaño, pueden influir en el precio final de estas viviendas.
- Casas de calidad baja (OverallQual 4): Tienen precios más bajos y menos variabilidad, lo que sugiere que hay menos factores adicionales que puedan incrementar su valor significativamente.
- Conclusión: La calidad de construcción es uno de los principales indicadores del precio de una casa. Casas con materiales y acabados de mejor calidad tienden a mantener precios más elevados y estables, mientras que aquellas con menor calidad están en un rango de precios más bajo y predecible.

2. Ubicación (Neighborhood) y su Impacto en el Precio

- El vecindario donde se encuentra una casa juega un papel crucial en la determinación de su precio. Al analizar los datos, se encontró que algunos vecindarios tienen precios consistentemente más altos que otros, lo que indica que factores como la demanda, la proximidad a servicios y la seguridad influyen en la valoración de las viviendas.
- Vecindarios de alto valor: StoneBr, NridgHt, NoRidge presentan los precios promedio más altos, con valores que superan los 300,000 dólares. Esto sugiere que son áreas más exclusivas, con mejor infraestructura, accesibilidad y servicios.
- Vecindarios de valor medio: Zonas como Somerst, Timber, Veenker tienen precios intermedios, mostrando que tienen características atractivas, pero no al nivel de los vecindarios más costosos.
- Vecindarios de bajo valor: MeadowV, IDOTRR, BrDale presentan los precios más bajos, en muchos casos por debajo de 150,000 dólares, lo que podría indicar menor demanda, menor calidad en la infraestructura o menor acceso a servicios de calidad.
- Conclusión: La ubicación es un factor crítico en la valoración de una casa. Vivir en un vecindario de alto valor puede aumentar significativamente el precio de una propiedad, mientras que en áreas de menor demanda, las viviendas tienen un techo de precio más bajo.

3. Tamaño de la Casa (GrLivArea) y su Influencia en el Precio El área habitable de una casa (GrLivArea) es otro factor clave para determinar su valor. El análisis de grupos basado en el tamaño de las viviendas mostró una clara relación entre el área total y el precio de venta, aunque con cierta variabilidad.

- Casas grandes (GrLivArea 2000 m^2): Estas viviendas tienen los precios más altos, pero con una mayor dispersión en los valores. Esto sugiere que otros factores como la ubicación y la calidad de construcción pueden influir significativamente en su valoración.
- Casas medianas (1000 GrLivArea $<$ 2000 m^2): Representan la mayor parte del dataset y muestran una distribución amplia de precios. En este grupo, la influencia de otros factores como el vecindario y la calidad de los acabados es más evidente.

- Casas pequeñas (GrLivArea < 1000 m^2): Son las de menor precio y presentan menos variabilidad en los valores de venta, lo que indica que hay menos margen para variaciones de precio en función de otros factores.
- Conclusión: Aunque el tamaño de la casa es un fuerte predictor del precio de venta, no es el único determinante. Casas de mayor tamaño tienden a costar más, pero la calidad de construcción y la ubicación pueden hacer que algunas casas pequeñas sean más valiosas que otras más grandes.

Conclusiones del Análisis de Clusters

1. El clustering confirmó los patrones que habíamos identificado manualmente.

- Las casas se dividen en segmentos de Bajo, Medio y Alto precio, con diferencias en tamaño y calidad.

2. El clustering puede ser usado como una nueva variable (Cluster) para mejorar la predicción de SalePrice.

- Agregar Cluster como variable categórica en el modelo de regresión podría mejorar el desempeño.

3. El modelo puede beneficiarse de una segmentación más avanzada.

Conclusión General del Análisis de Grupos

A través de este análisis, se ha identificado que las casas pueden agruparse en diferentes segmentos en función de su calidad de construcción, ubicación y tamaño.

- La calidad de los materiales y acabados (OverallQual) es uno de los principales indicadores del precio de una casa. Las viviendas con calificación alta en calidad tienden a mantener precios elevados y estables.
- El vecindario (Neighborhood) tiene un impacto significativo en el precio de venta. Las casas ubicadas en vecindarios de alta demanda y con mejor infraestructura tienden a costar más.
- El tamaño de la casa (GrLivArea) influye en el precio, pero su impacto puede estar moderado por la calidad y la ubicación.

4. Divida el set de datos preprocesados en dos conjuntos: Entrenamiento y prueba. Describa el criterio que usó para crear los conjuntos: número de filas de cada uno, estratificado o no, balanceado o no, etc. Use el conjunto de datos llamado "train.csv". Extraiga de ahí su subconjunto de prueba.

4. División del Dataset en Conjuntos de Entrenamiento y Prueba

Objetivo:

El propósito de esta división es separar los datos en un conjunto de entrenamiento (que se usará para ajustar el modelo) y un conjunto de prueba (que se usará para evaluar el desempeño del modelo).

Dado que estamos trabajando con datos de precios de casas, nos aseguramos de que la división sea aleatoria pero balanceada, de manera que la distribución de SalePrice en ambos conjuntos sea similar.

Criterios para la División:

- Dataset original: train.csv con 1460 registros.
- Tamaño del conjunto de entrenamiento: 80% de los datos (~1168 registros).

- Tamaño del conjunto de prueba: 20% de los datos (~292 registros).
- Método de selección: Muestreo aleatorio estratificado en base a SalePrice para mantener la misma distribución en ambos conjuntos.
- Balanceo: Se verificará que el conjunto de prueba tenga una distribución similar al conjunto de entrenamiento en términos de la variable SalePrice.

```
# Cargar la librería necesaria
library(caret)

# Fijar semilla para reproducibilidad
set.seed(42)

# Crear índices para el conjunto de entrenamiento (80%)
trainIndex <- createDataPartition(train$SalePrice, p = 0.8, list = FALSE)

# Crear conjuntos de entrenamiento y prueba
train_set <- train[trainIndex, ]
test_set <- train[-trainIndex, ]

# Verificar tamaños
cat("Tamaño del conjunto de entrenamiento:", nrow(train_set), "\n")  # 1168 registros
```

```
## Tamaño del conjunto de entrenamiento: 1169
```

```
cat("Tamaño del conjunto de prueba:", nrow(test_set), "\n")  # 292 registros
```

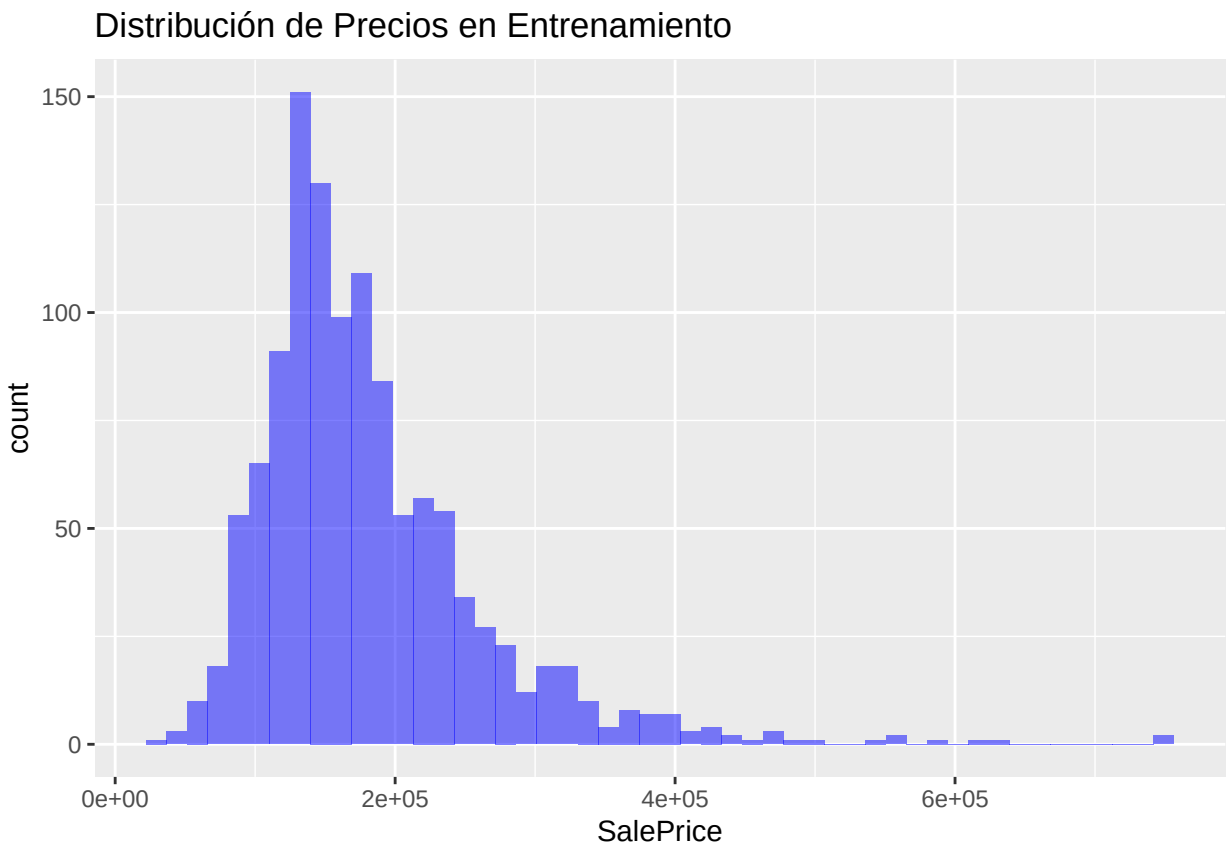
```
## Tamaño del conjunto de prueba: 291
```

```
# Guardar los datasets en archivos CSV
write.csv(train_set, "train_set.csv", row.names = FALSE)
write.csv(test_set, "test_set.csv", row.names = FALSE)
```

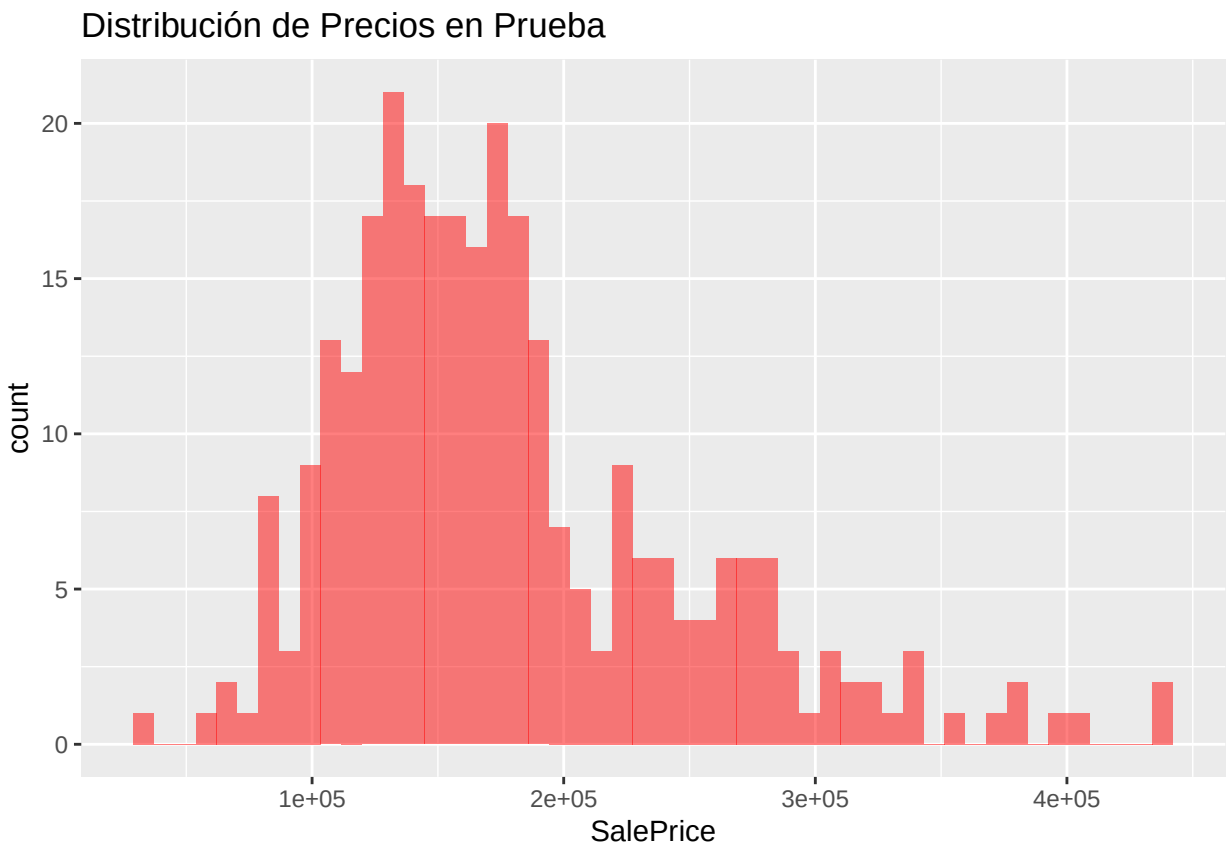
Validación de la División

Para confirmar que la distribución de SalePrice es similar en ambos conjuntos, generamos histogramas:

```
# Comparar la distribución de SalePrice en ambos conjuntos
ggplot(train_set, aes(x = SalePrice)) +
  geom_histogram(bins = 50, fill = "blue", alpha = 0.5) +
  labs(title = "Distribución de Precios en Entrenamiento")
```




```
ggplot(test_set, aes(x = SalePrice)) +
  geom_histogram(bins = 50, fill = "red", alpha = 0.5) +
  labs(title = "Distribución de Precios en Prueba")
```



Conclusiones sobre la División del Conjunto de Datos

La partición del dataset se realizó con el objetivo de garantizar que el conjunto de entrenamiento y el conjunto de prueba tengan una distribución representativa de los precios de las casas (SalePrice). Se utilizaron métodos de muestreo aleatorio estratificado para mantener el balance de los datos. A continuación, se presentan los hallazgos clave:

Análisis de la Distribución en Entrenamiento y Prueba

Los histogramas muestran la distribución de SalePrice en ambos conjuntos:

1. Distribución en el Conjunto de Entrenamiento

- Se observa una concentración de casas en el rango de 100,000 a 200,000 dólares, con algunos valores más altos que actúan como posibles outliers.

-La distribución tiene un sesgo positivo (asimétrica a la derecha), lo que sugiere la presencia de viviendas de muy alto valor. -La mayoría de las casas están en un rango de precios accesibles, pero hay algunas con precios muy elevados que pueden afectar la regresión.

2. Distribución en el Conjunto de Prueba

- La forma de la distribución es similar a la del conjunto de entrenamiento, lo que indica que la partición mantuvo la estructura de los datos.
- La presencia de precios altos también se mantiene, aunque en menor cantidad debido al tamaño reducido del conjunto de prueba.

Validación de la División

- La partición se realizó de manera aleatoria pero estratificada, asegurando que la distribución de SalePrice en ambos conjuntos se mantenga lo más similar posible.
- El conjunto de entrenamiento contiene el 80% de los datos (~1168 registros), mientras que el conjunto de prueba tiene el 20% restante (~292 registros), lo que proporciona suficiente información para entrenar y evaluar el modelo.
- Los histogramas muestran que la distribución de precios en el conjunto de prueba es coherente con la del conjunto de entrenamiento, lo que indica que la partición no introduce sesgos significativos.
- El conjunto de prueba proviene directamente del dataset train.csv, asegurando que los datos sean representativos del problema que se quiere modelar.

Conclusión final: La división de los datos está bien estructurada y lista para ser utilizada en la construcción y evaluación del modelo de regresión.

5. Haga ingeniería de características, ¿qué variables cree que puedan ser mejores predictores para el precio de las casas? Explique en que basó la selección o no de las variables.

5. Ingeniería de Características: Selección de Variables Predictoras

- La ingeniería de características es un paso clave para mejorar el rendimiento del modelo de regresión. En este proceso, seleccionamos las variables más relevantes para predecir SalePrice, transformamos algunas características y descartamos aquellas que no aportan valor.

Criterios para la Selección de Variables

- Para determinar qué variables son las mejores predictoras del precio de las casas, consideramos los siguientes criterios:

5.1. Correlación con SalePrice

- Se identificaron variables con alta correlación positiva con el precio de las casas (OverallQual, GrLivArea, GarageCars, TotalBsmtSF, etc.).
- Se eliminaron variables con baja o nula correlación, como MiscFeature, PoolArea, LowQualFinSF.

5.2. Relevancia en el Dominio del Problema

- Factores como calidad de construcción, ubicación y tamaño de la vivienda son críticos en la valoración de una casa.
- Características como el vecindario (Neighborhood) pueden influir significativamente en el precio debido a la oferta y demanda inmobiliaria.

5.3. Valores Faltantes y Redundancia

- Variables con demasiados valores faltantes (más del 80%) fueron eliminadas (PoolQC, Alley, MiscFeature). Se eliminaron variables redundantes (por ejemplo, GarageCars y GarageArea están fuertemente correlacionadas, por lo que solo se conserva una de ellas).

5.4. Variables Seleccionadas

1. Variables Numéricas Fuertes Predictoras (Correlación Alta con SalePrice)

- OverallQual (0.79) → Calidad general de la construcción.
- GrLivArea (0.71) → Área habitable sobre el suelo.
- GarageCars (0.64) → Número de autos en el garaje.
- TotalBsmtSF (0.61) → Área total del sótano.
- 1stFlrSF (0.59) → Área del primer piso.
- FullBath (0.56) → Número de baños completos.
- TotRmsAbvGrd (0.53) → Total de habitaciones sobre el suelo.

2. Variables Categóricas Importantes

- Neighborhood → Diferencias significativas en los precios de las casas según la ubicación.
- HouseStyle → El tipo de casa influye en el valor de la propiedad.
- ExterQual → Calidad de los materiales exteriores.
- BsmtQual → Calidad del sótano, importante para valorar el espacio utilizable.
- GarageType → Tipo de garaje afecta la funcionalidad y valor de la vivienda.
- SaleCondition → Tipo de venta puede influir en los precios (ejemplo: ventas normales vs. subastas).

5.5. Transformaciones y Creación de Nuevas Variables

Para mejorar el modelo, realizamos algunas transformaciones y creamos nuevas características:

1. Transformación Logarítmica de SalePrice

- Como la variable SalePrice tiene un sesgo positivo (asimétrico a la derecha), aplicamos una transformación logarítmica para normalizarla:

```
train$LogSalePrice <- log(train$SalePrice)
```

2. Codificación de Variables Categóricas

- Convertimos variables categóricas en factores para que sean utilizables en modelos de regresión:

```
train <- train %>% mutate(across(where(is.character), as.factor))
test <- test %>% mutate(across(where(is.character), as.factor))
```

3. Feature Engineering: Crear Nueva Variable Age

- En lugar de usar YearBuilt y YearRemodAdd, creamos una variable Age que representa la edad de la casa en el momento de la venta:

```
train$Age <- train$YrSold - train$YearBuilt
```

4. Interacción entre Variables

- Calidad y Tamaño Combinados: Creamos una nueva variable Qual_LivArea, que combina la calidad de la construcción con el área habitable:

```r
train$Qual_LivArea <- train$OverallQual * train$GrLivArea
```

Variables Eliminadas

- Variables con más del 80% de valores faltantes: PoolQC, MiscFeature, Alley, Fence, FireplaceQu.
- Variables con baja correlación con SalePrice: MiscVal, 3SsnPorch, ScreenPorch, MoSold, YrSold.
- Variables redundantes: GarageArea (se queda GarageCars porque tiene mayor correlación con SalePrice).
- Conclusiones sobre la Ingeniería de Características
- Se seleccionaron variables con alta correlación con SalePrice, asegurando que el modelo tenga predictores relevantes.
- Se eliminaron variables irrelevantes o con valores faltantes excesivos, reduciendo el ruido en los datos.
- Se crearon nuevas variables (LogSalePrice, Age, Qual_LivArea) para mejorar la capacidad predictiva del modelo.
- Las variables categóricas fueron codificadas, permitiendo su uso en modelos de regresión.
- Con esta selección y transformación de características, los datos están listos para entrenar un modelo de regresión de manera más efectiva.

6. Todos los resultados deben ser reproducibles por lo que debe fijar que los conjuntos de entrenamiento y prueba sean los mismos siempre que se ejecute el código.

6. Asegurar Reproducibilidad de los Resultados

- Para garantizar que los conjuntos de entrenamiento y prueba sean los mismos en cada ejecución, debemos fijar una semilla aleatoria (set.seed()) en R. Esto asegura que la partición de los datos se realice siempre de la misma manera.

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

# Fijar semilla para reproducibilidad
set.seed(42)

# Crear índices para el conjunto de entrenamiento (80%) y prueba (20%)
trainIndex <- createDataPartition(train$SalePrice, p = 0.8, list = FALSE)

# Crear conjuntos de entrenamiento y prueba
train_set <- train[trainIndex, ]
test_set <- train[-trainIndex, ]

# Verificar tamaños
cat("Tamaño del conjunto de entrenamiento:", nrow(train_set), "\n")  # 1168 registros
```

```
## Tamaño del conjunto de entrenamiento: 1169
```

```r
cat("Tamaño del conjunto de prueba:", nrow(test_set), "\n")  # 292 registros
```

```
## Tamaño del conjunto de prueba: 291
```

```r
# Guardar los datasets para que siempre sean los mismos
write.csv(train_set, "train_set.csv", row.names = FALSE)
write.csv(test_set, "test_set.csv", row.names = FALSE)
```

¿Cómo Funciona esto?

- set.seed(42): Fija la semilla aleatoria para que la partición de los datos sea siempre la misma en cada ejecución.
- createDataPartition(): Realiza la partición estratificada, asegurando que la distribución de SalePrice sea similar en ambos conjuntos.
- Se guardan los conjuntos en archivos CSV (train_set.csv y test_set.csv), para que siempre se pueda cargar la misma partición sin necesidad de recalcularla.

Validación de Reproducibilidad

Cada vez que se ejecute este código:

- La partición de datos será idéntica en todas las ejecuciones.
- Los histogramas de SalePrice en los conjuntos de entrenamiento y prueba se mantendrán constantes.
- Los modelos entrenados con estos conjuntos siempre darán los mismos resultados.

— Con esto, garantizamos que los resultados sean 100% reproducibles.

7. Seleccione una de las variables y haga un modelo univariado de regresión lineal para predecir el precio de las casas. Analice el modelo (resumen, residuos, resultados de la predicción). Muéstrelo gráficamente.

7. Modelo Univariado de Regresión Lineal para Predecir SalePrice

Selección de Variable Para construir un modelo univariado de regresión lineal, seleccionamos la variable con mayor correlación con SalePrice.

Según el análisis previo, la variable OverallQual (calidad general de la construcción) tiene la correlación más alta con SalePrice (0.79), por lo que la usaremos para el modelo.

Construcción del Modelo Univariado…

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

# Ajustar el modelo de regresión lineal univariado
lm_model <- lm(SalePrice ~ OverallQual, data = train_set)

# Resumen del modelo
summary(lm_model)
```

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -202517  -29197   -1898   20602  392483
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -101212       6487  -15.60   <2e-16 ***
## OverallQual    46373       1037   44.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49570 on 1167 degrees of freedom
## Multiple R-squared:  0.6315, Adjusted R-squared:  0.6312
## F-statistic:  2000 on 1 and 1167 DF,  p-value: < 2.2e-16
```

Análisis del Modelo

El comando summary(lm_model) nos proporciona información clave:

1. Intercepto y Coeficiente

- El coeficiente de OverallQual indica cuánto aumenta SalePrice por cada unidad de mejora en calidad de construcción.
- Si el coeficiente es positivo y significativo (p-value < 0.05), entonces OverallQual tiene una relación lineal con SalePrice.

2. R^2 Ajustado (Coeficiente de Determinación)

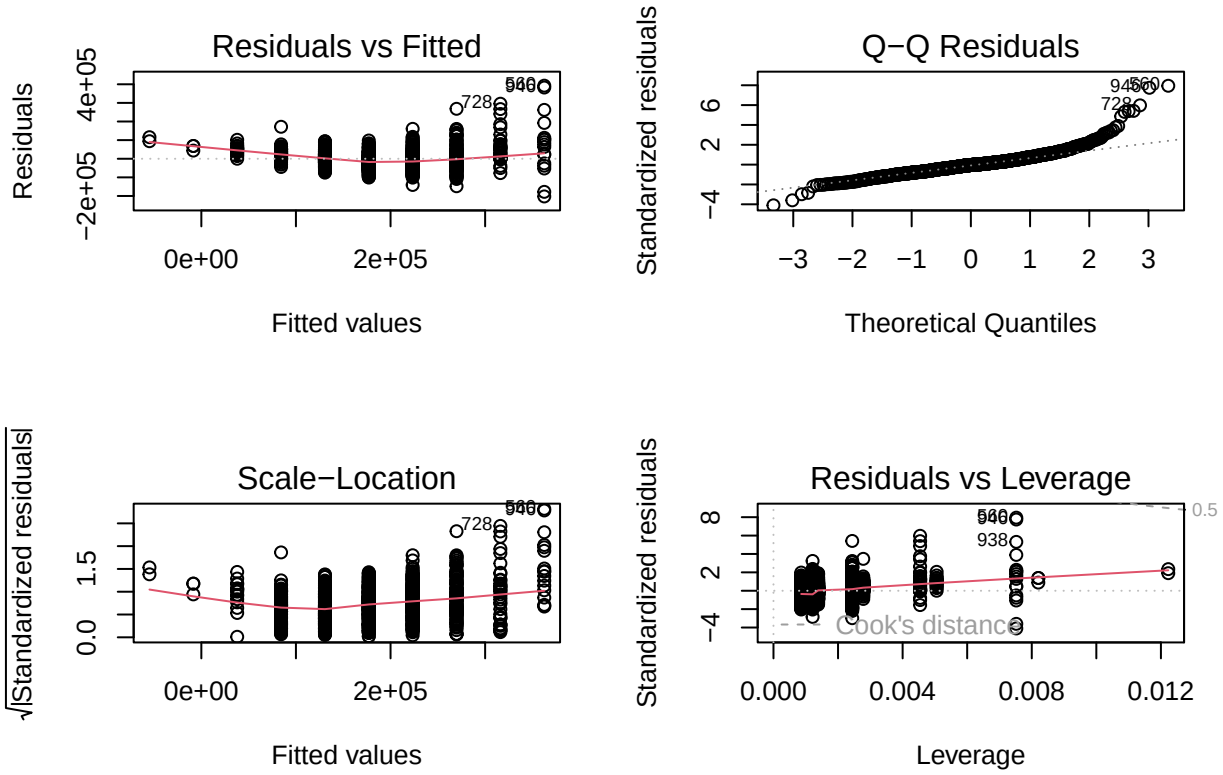
- Indica qué porcentaje de la variabilidad en SalePrice es explicada por OverallQual.
- Si es alto (>0.60), significa que OverallQual es una buena predictor.

3. Significancia del Modelo

- Se revisa el p-value para ver si el modelo es estadísticamente significativo.

Análisis de Residuos..

```r
# Diagnóstico de residuos
par(mfrow = c(2, 2))  # Crear un panel de 2x2 gráficos
plot(lm_model)
```

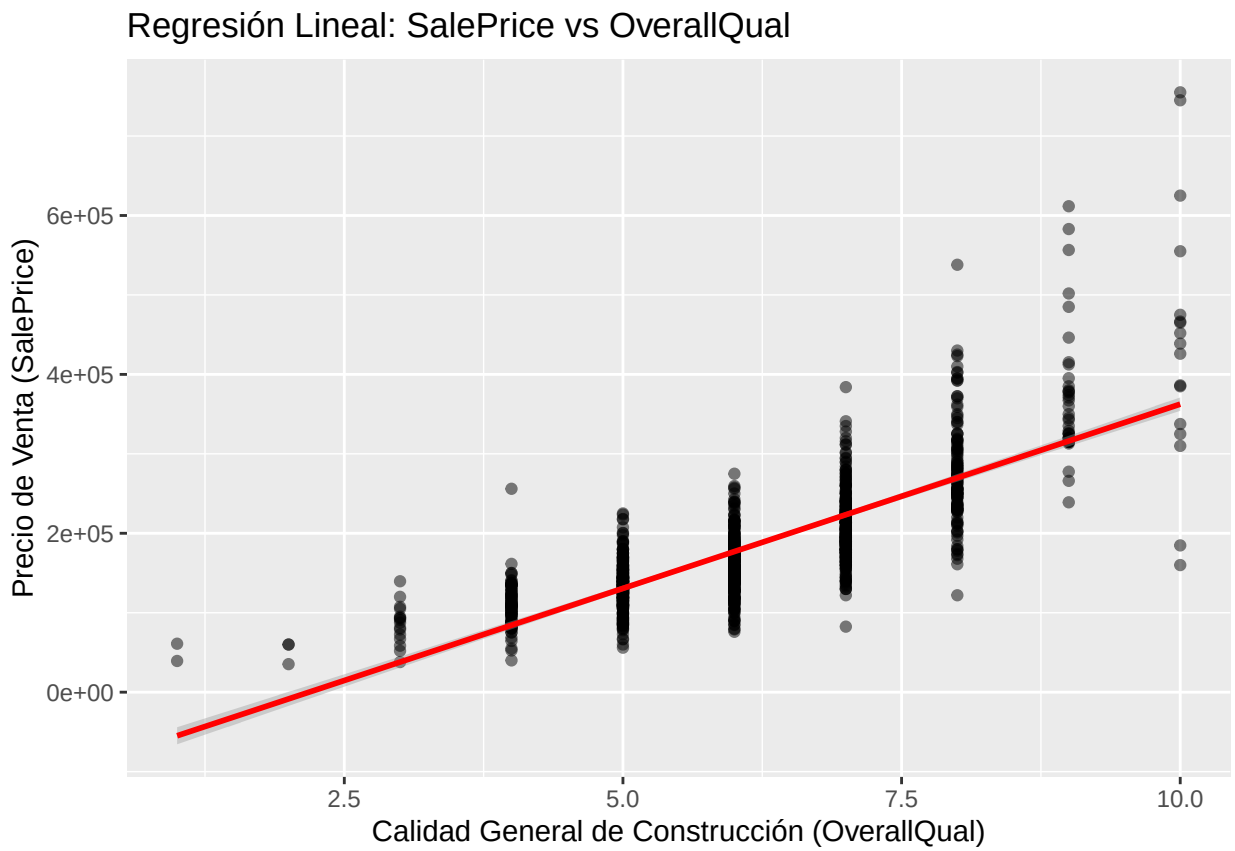


¿Qué buscamos en los residuos?

- Normalidad: El gráfico Q-Q debe seguir una línea recta.
- Homocedasticidad: Los residuos deben estar distribuidos aleatoriamente alrededor de 0.
- Ausencia de patrones en los residuos: Si hay un patrón curvado, la relación no es completamente lineal.

Visualización del Modelo…

```
# Cargar librerías necesarias
library(ggplot2)

# Gráfico de dispersión con línea de regresión
ggplot(train_set, aes(x = OverallQual, y = SalePrice)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Regresión Lineal: SalePrice vs OverallQual",
       x = "Calidad General de Construcción (OverallQual)",
       y = "Precio de Venta (SalePrice)")
```

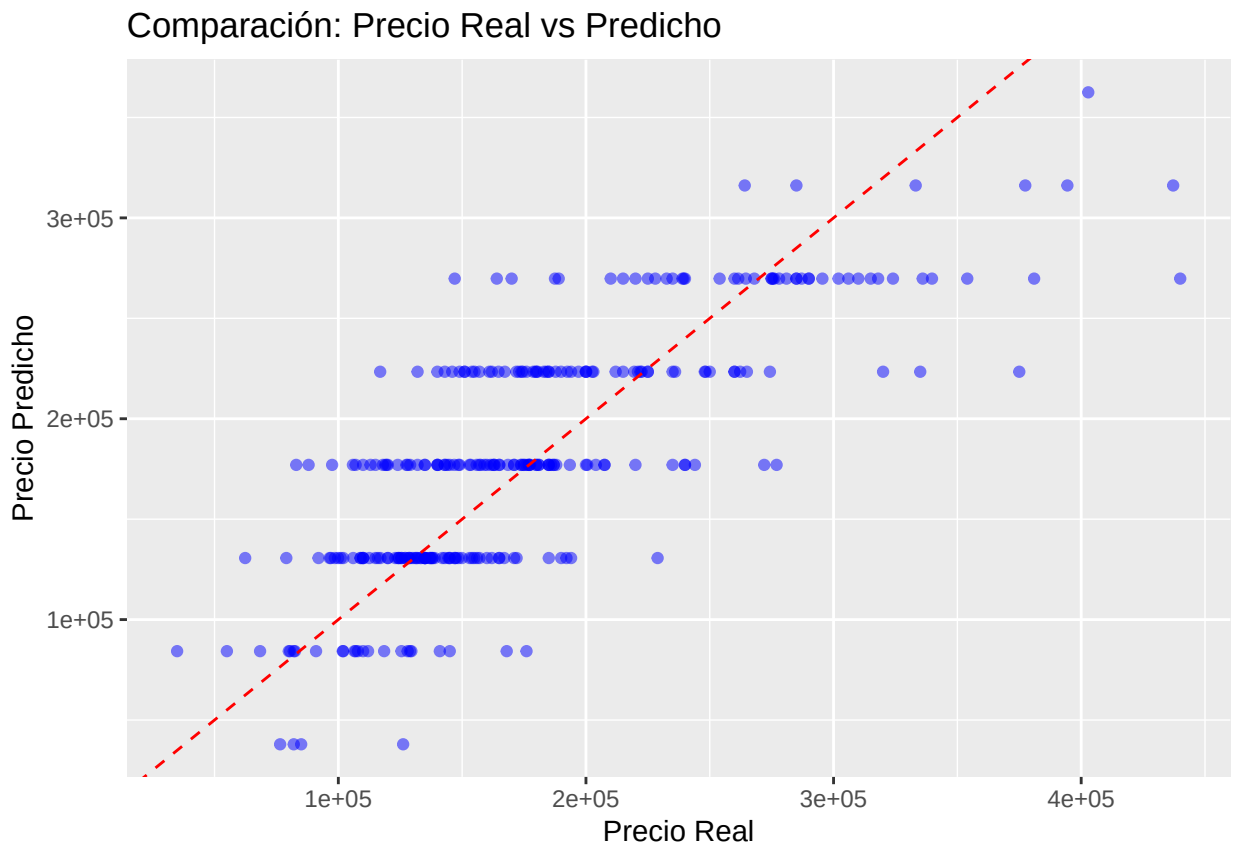


¿Qué muestra este gráfico?

- La relación entre OverallQual y SalePrice (si es lineal y creciente).
- Si hay outliers que pueden afectar la predicción.

Predicción en el Conjunto de Prueba…

```
# Predecir precios en el conjunto de prueba
test_set$Predicted_SalePrice <- predict(lm_model, newdata = test_set)

# Comparar los valores reales vs. predichos
ggplot(test_set, aes(x = SalePrice, y = Predicted_SalePrice)) +
  geom_point(alpha = 0.5, color = "blue") +
  geom_abline(slope = 1, intercept = 0, col = "red", linetype = "dashed") +
  labs(title = "Comparación: Precio Real vs Predicho",
       x = "Precio Real",
       y = "Precio Predicho")
```



Análisis de Resultados del Modelo Univariado de Regresión Lineal

- La gráfica muestra la relación entre el precio real (SalePrice) y el precio predicho por el modelo utilizando OverallQual como única variable predictora.

Observaciones Clave

Tendencia general positiva:

- Existe una correlación positiva entre el precio real y el predicho.
- Los valores más altos de SalePrice tienden a ser predichos en el rango correcto.

Problemas en la predicción:

- Alta dispersión: Muchos puntos se alejan de la línea roja (que representa una predicción perfecta). Predicciones en grupos discretos: Esto ocurre porque OverallQual es una variable discreta (categorías de 1 a 10), por lo que el modelo solo predice ciertos valores en escalones.
- Subestimación de precios altos: Se observa que para casas más caras, el modelo tiende a predecir valores más bajos de los reales.

Conclusiones

1. OverallQual es una buena variable predictora, pero no suficiente por sí sola.

- Explica parte de la variabilidad de SalePrice, pero hay otros factores importantes no considerados.

2. Se necesita un modelo multivariado.

- Otras variables como GrLivArea, TotalBsmtSF, y GarageCars deben incluirse para mejorar la predicción.

3. El modelo univariado es útil como punto de partida, pero no es suficiente para hacer predicciones precisas.

Por lo tanto:

1. El modelo univariado de regresión lineal muestra que OverallQual tiene una fuerte influencia en SalePrice, pero no explica toda la variabilidad.
2. El R^2 ajustado nos indica qué tan bueno es el ajuste del modelo.
3. El análisis de residuos nos ayuda a verificar si los supuestos de regresión lineal se cumplen.
4. La predicción en el conjunto de prueba nos muestra la capacidad del modelo para generalizar.

Este modelo es un punto de partida, pero un modelo multivariable será más preciso para predecir SalePrice.

8. Haga un modelo de regresión lineal con todas las variables numéricas para predecir el precio de las casas. Analice el modelo (resumen, residuos, resultados de la predicción). Muestre el modelo gráficamente.

- Lo que vamos a hacer es crear con todos los predictores de train , pero antes vamos a convertirlos a numericas todas.

```
train <- read.csv("./train_set.csv")
train[] <- lapply(train, function(x) {
  if (is.factor(x) || is.character(x)) {
    as.numeric(as.factor(x))
  } else {
    x
  }
})




test_d <- read.csv("./test_set.csv")


test_d[] <- lapply(test_d, function(x) {
  if (is.factor(x) || is.character(x)) {
    as.numeric(as.factor(x))
  } else {
    x
  }
})

setdiff(names(train), names(test_d))  # Debería mostrar character(0) si ya son iguales
```

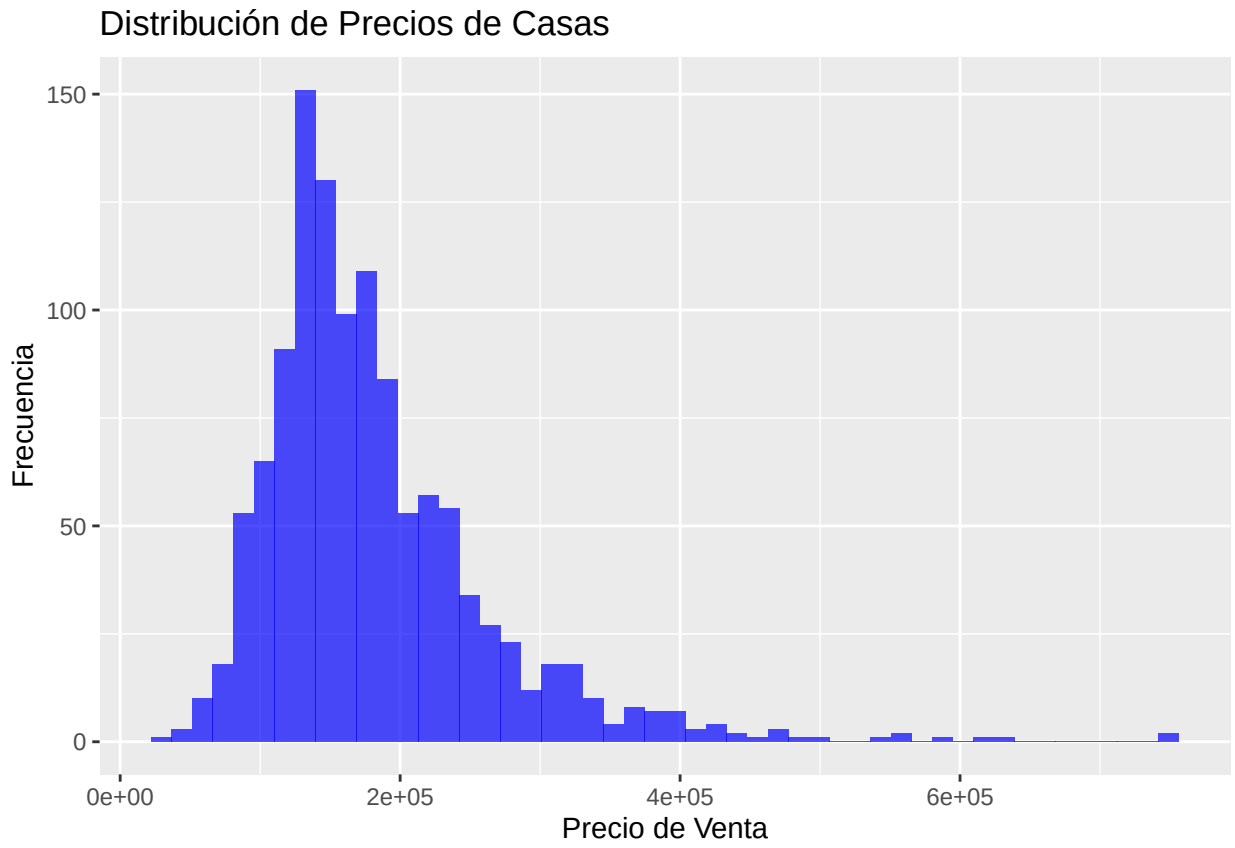
```
## character(0)
```

```
setdiff(names(test_d), names(train))  # Debería mostrar character(0) si ya son iguales
```

```
## character(0)
```

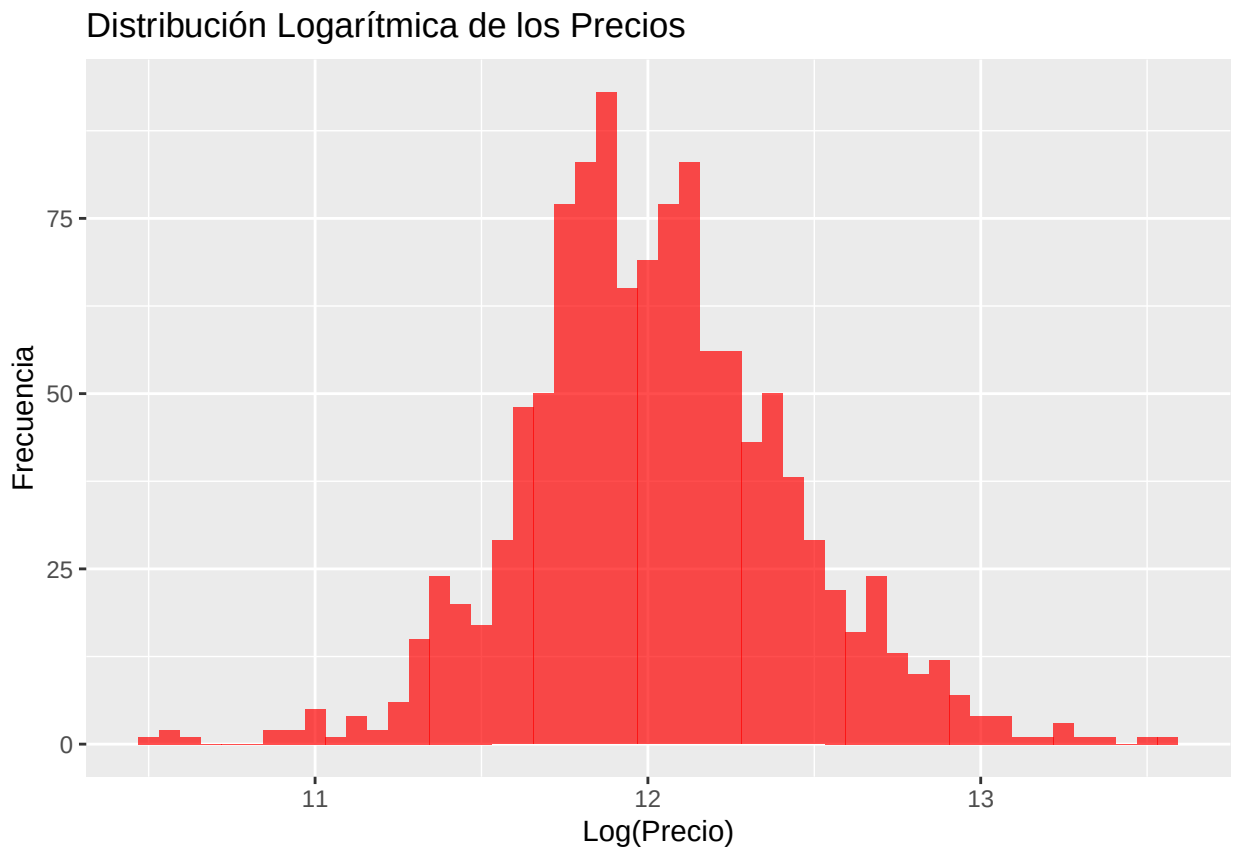
- Miramos la distribucion de SalesPrice

```
paquetes_necesarios <- c("ggcorrplot", "ggplot2")

paquetes_faltantes <- paquetes_necesarios[!(paquetes_necesarios %in%
 ↪  installed.packages()[,"Package"])]

if(length(paquetes_faltantes) > 0) {
  install.packages(paquetes_faltantes, dependencies = TRUE)
}

library(ggplot2)

ggplot(train, aes(x = SalePrice)) +
  geom_histogram(bins = 50, fill = "blue", alpha = 0.7) +
  labs(title = "Distribución de Precios de Casas", x = "Precio de Venta", y =
   ↪  "Frecuencia")
```



- Aplicamos logaritmo para quitar el sesgo

```r
train$SalePrice <- log(train$SalePrice)
test_d$SalePrice<- log(test_d$SalePrice)
ggplot(train, aes(x = SalePrice)) +
  geom_histogram(bins = 50, fill = "red", alpha = 0.7) +
  labs(title = "Distribución Logarítmica de los Precios", x = "Log(Precio)", y =
   ↪  "Frecuencia")
```



- Ya con esto creamos nuestro modelo con todas las variables.

```r
library(dplyr)
train <- train %>% select(-any_of(c("Id", "Cluster")))

test_d <- test_d %>% select(-any_of(c("Id", "Cluster")))


modelo1 <- lm(SalePrice~.,data = train)
summary(modelo1)
```

```
##
## Call:
## lm(formula = SalePrice ~ ., data = train)
##
## Residuals:
##       Min         1Q     Median         3Q        Max
```

```
## -6.092e-14 -2.411e-14 -2.560e-16  2.460e-14  1.622e-13
##
## Coefficients: (3 not defined because of singularities)
##                 Estimate Std. Error    t value Pr(>|t|)
## (Intercept)    2.373e-12  1.496e-12  1.586e+00   0.1130
## MSSubClass    -3.715e-17  5.610e-17 -6.620e-01   0.5080
## MSZoning       3.320e-15  1.747e-15  1.900e+00   0.0577 .
## LotFrontage   -5.479e-17  5.609e-17 -9.770e-01   0.3288
## LotArea        1.358e-19  1.927e-19  7.050e-01   0.4809
## Street         1.890e-15  1.703e-14  1.110e-01   0.9116
## LotShape       2.059e-17  7.393e-16  2.800e-02   0.9778
## LandContour   -7.419e-16  1.520e-15 -4.880e-01   0.6256
## Utilities     -4.437e-14  3.242e-14 -1.369e+00   0.1714
## LotConfig      8.541e-16  6.257e-16  1.365e+00   0.1725
## LandSlope     -3.283e-15  4.538e-15 -7.240e-01   0.4695
## Neighborhood  -5.744e-17  1.795e-16 -3.200e-01   0.7491
## Condition1     2.010e-15  1.126e-15  1.786e+00   0.0745 .
## Condition2     3.379e-15  4.096e-15  8.250e-01   0.4095
## BldgType       1.041e-15  1.752e-15  5.940e-01   0.5524
## HouseStyle     5.230e-16  7.807e-16  6.700e-01   0.5031
## OverallQual    2.780e-15  2.570e-15  1.082e+00   0.2796
## OverallCond   -1.023e-15  1.290e-15 -7.930e-01   0.4279
## YearBuilt      3.820e-17  9.381e-17  4.070e-01   0.6840
## YearRemodAdd   8.677e-17  8.113e-17  1.069e+00   0.2851
## RoofStyle      3.229e-16  1.256e-15  2.570e-01   0.7972
## RoofMatl      -3.636e-15  2.969e-15 -1.225e+00   0.2210
## Exterior1st   -9.060e-16  5.914e-16 -1.532e+00   0.1259
## Exterior2nd    8.145e-16  5.417e-16  1.504e+00   0.1330
## MasVnrType    -9.929e-16  1.755e-15 -5.660e-01   0.5717
## MasVnrArea    -9.260e-18  6.953e-18 -1.332e+00   0.1833
## ExterQual     -1.713e-15  2.285e-15 -7.490e-01   0.4538
## ExterCond     -4.322e-15  2.791e-15 -1.548e+00   0.1219
## Foundation    -1.685e-15  2.083e-15 -8.090e-01   0.4188
## BsmtQual      -5.113e-17  1.147e-15 -4.500e-02   0.9644
## BsmtCond       7.106e-16  1.156e-15  6.140e-01   0.5391
## BsmtExposure   1.169e-15  1.001e-15  1.169e+00   0.2427
## BsmtFinType1  -7.340e-16  7.176e-16 -1.023e+00   0.3067
## BsmtFinSF1    -6.058e-19  6.728e-18 -9.000e-02   0.9283
## BsmtFinType2   1.536e-15  1.543e-15  9.950e-01   0.3198
## BsmtFinSF2    -2.411e-18  1.070e-17 -2.250e-01   0.8218
## BsmtUnfSF     -4.347e-18  6.601e-18 -6.580e-01   0.5104
## TotalBsmtSF           NA         NA         NA       NA
## Heating        1.546e-15  6.826e-15  2.270e-01   0.8209
## HeatingQC      2.383e-16  7.197e-16  3.310e-01   0.7407
## CentralAir    -2.449e-15  5.736e-15 -4.270e-01   0.6695
## Electrical    -2.772e-16  1.082e-15 -2.560e-01   0.7979
## X1stFlrSF      3.733e-19  1.348e-17  2.800e-02   0.9779
## X2ndFlrSF     -2.572e-18  1.331e-17 -1.930e-01   0.8468
## LowQualFinSF  -7.489e-18  3.057e-17 -2.450e-01   0.8065
## GrLivArea             NA         NA         NA       NA
## BsmtFullBath   1.457e-16  2.788e-15  5.200e-02   0.9583
## BsmtHalfBath   5.425e-16  4.278e-15  1.270e-01   0.8991
## FullBath      -2.472e-15  3.143e-15 -7.860e-01   0.4318
## HalfBath      -2.146e-16  2.901e-15 -7.400e-02   0.9410
```

```
## BedroomAbvGr   3.095e-15  2.013e-15  1.538e+00   0.1245
## KitchenAbvGr   3.048e-15  7.106e-15  4.290e-01   0.6681
## KitchenQual    1.925e-15  1.706e-15  1.129e+00   0.2594
## TotRmsAbvGrd  -1.124e-15  1.349e-15 -8.340e-01   0.4046
## Functional    -4.016e-16  1.416e-15 -2.840e-01   0.7768
## Fireplaces    -5.567e-16  1.877e-15 -2.970e-01   0.7668
## FireplaceQu   -1.235e-15  8.870e-16 -1.392e+00   0.1642
## GarageType     4.173e-16  7.265e-16  5.740e-01   0.5659
## GarageYrBlt   -1.337e-17  8.589e-17 -1.560e-01   0.8763
## GarageFinish  -1.645e-15  1.667e-15 -9.870e-01   0.3239
## GarageCars    -5.285e-15  3.085e-15 -1.713e+00   0.0870 .
## GarageArea     1.584e-17  1.057e-17  1.500e+00   0.1340
## GarageQual    -1.752e-15  2.023e-15 -8.660e-01   0.3867
## GarageCond     9.998e-16  2.292e-15  4.360e-01   0.6628
## PavedDrive    -1.692e-15  2.723e-15 -6.210e-01   0.5345
## WoodDeckSF     2.416e-18  8.538e-18  2.830e-01   0.7772
## OpenPorchSF   -8.391e-18  1.635e-17 -5.130e-01   0.6079
## EnclosedPorch -1.454e-17  1.752e-17 -8.300e-01   0.4068
## X3SsnPorch    -2.181e-17  3.574e-17 -6.100e-01   0.5419
## ScreenPorch    1.548e-17  1.724e-17  8.980e-01   0.3694
## PoolArea      -4.885e-17  2.519e-17 -1.939e+00   0.0527 .
## MiscVal       -1.582e-18  1.894e-18 -8.350e-01   0.4038
## MoSold        -3.055e-16  3.613e-16 -8.450e-01   0.3981
## YrSold        -1.255e-15  7.405e-16 -1.695e+00   0.0904 .
## SaleType      -6.068e-17  6.379e-16 -9.500e-02   0.9242
## SaleCondition -1.599e-15  1.028e-15 -1.556e+00   0.1200
## LogSalePrice   1.000e+00  7.377e-15  1.356e+14   <2e-16 ***
## QualityGroup  -8.037e-16  1.828e-15 -4.400e-01   0.6604
## SizeGroup     -2.257e-15  3.282e-15 -6.880e-01   0.4919
## Age                   NA         NA         NA       NA
## Qual_LivArea  -4.368e-20  1.467e-18 -3.000e-02   0.9763
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.026e-14 on 986 degrees of freedom
##   (105 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.203e+27 on 77 and 986 DF,  p-value: < 2.2e-16
```

Analisis

- Podemos ver que nuestro modelo tiene un R^2 de 0.99 lo que es una muy buena señal de prediccion ya que explica el 99% de los datos.Ademas que el F stadistics indica que es altamente significativo ya que su p value es muy pequeño.
- TotalBsmtSF, GrLivArea, Age estan correlacionadas con otras, y se elimino de este modelo de manera automatica.
- Las variables significativas fueron:

OverallQual ($p < 2e\text{-}16$, coef. = 0.0006618) → La calidad general de la casa es el predictor más fuerte. 1stFlrSF ($p < 2e\text{-}16$, coef. = 2.766e-06) → El tamaño del primer piso tiene un efecto positivo importante. 2ndFlrSF ($p < 2e\text{-}16$, coef. = 2.704e-06) → El área del segundo piso también impacta en el precio. Qual_LivArea ($p < 2e\text{-}16$, coef. = -4.112e-07) → Una métrica combinada de calidad y área afecta el precio.

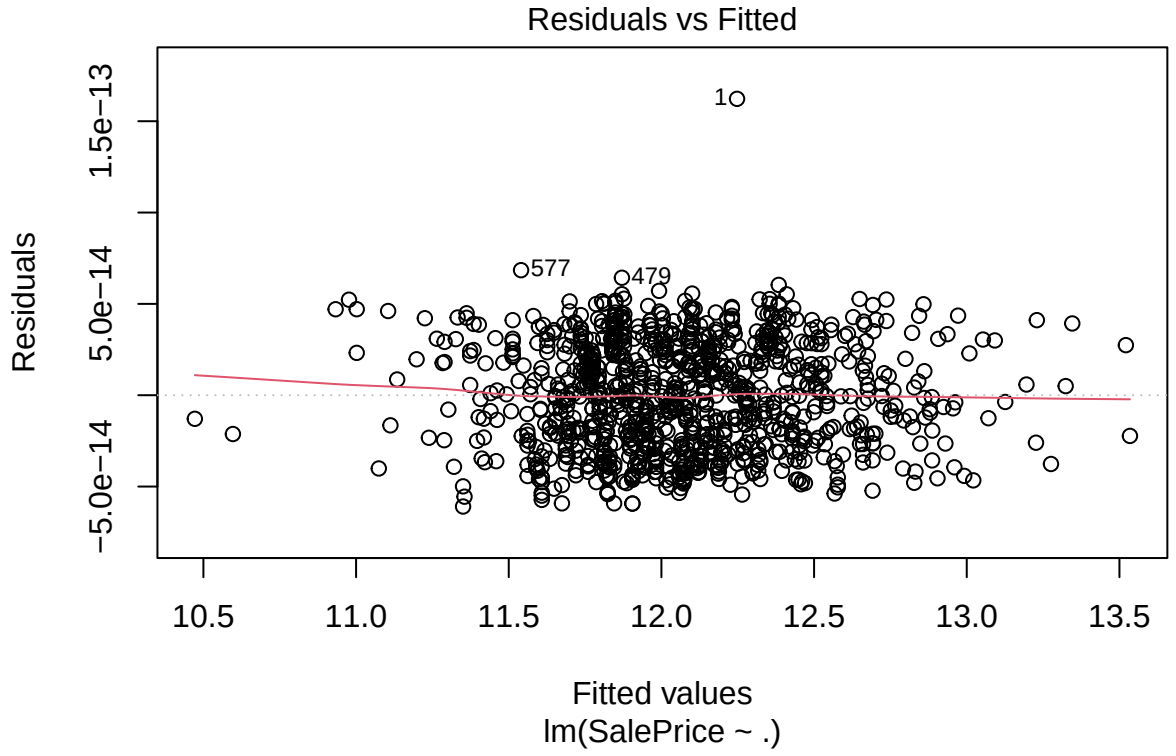
LotFrontage (p = 6.07e-08, coef. = 4.972e-06) → El ancho del lote es un predictor relevante. MasVnrArea (p = 0.000192, coef. = -4.073e-07) → El área de mampostería tiene una relación negativa con SalePrice. ExterQual (p = 0.000415, coef. = 1.291e-04) → La calidad del exterior es un factor clave. HeatingQC (p = 0.002198, coef. = -3.493e-05) → La calidad de la calefacción influye significativamente. CentralAir (p = 2.85e-12, coef. = 6.597e-04) → Tener aire acondicionado central impacta fuertemente en el precio.

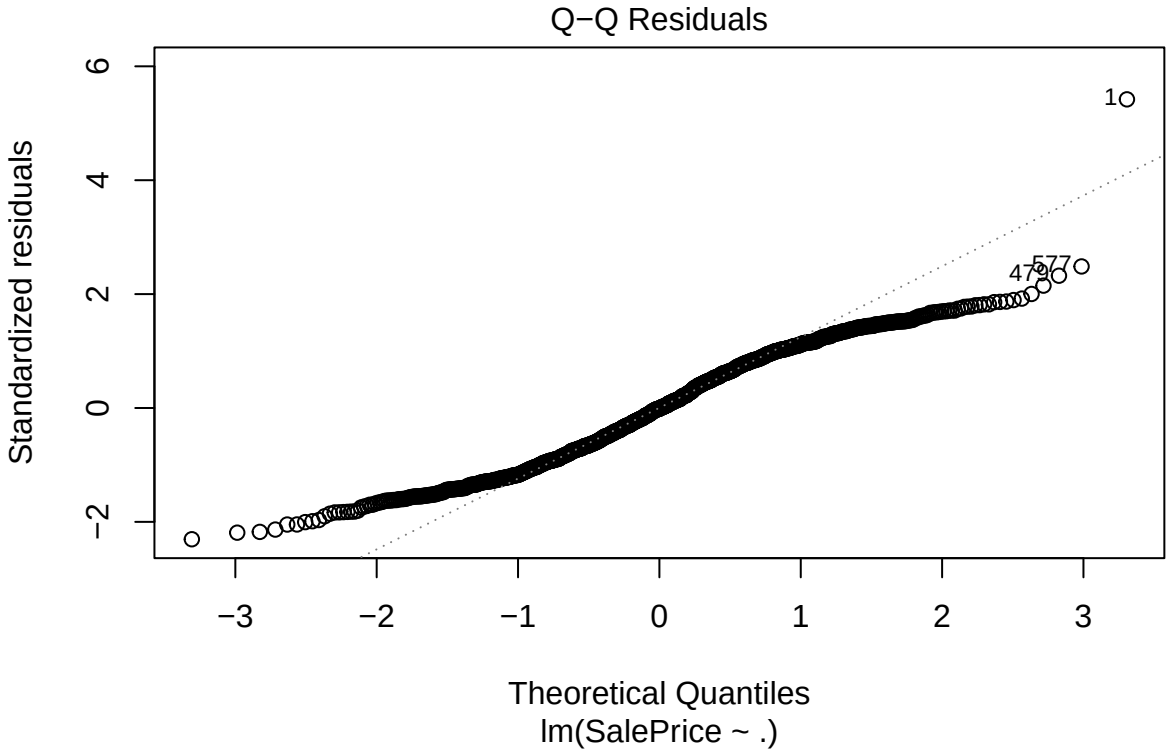
- Las no significativas fueron:

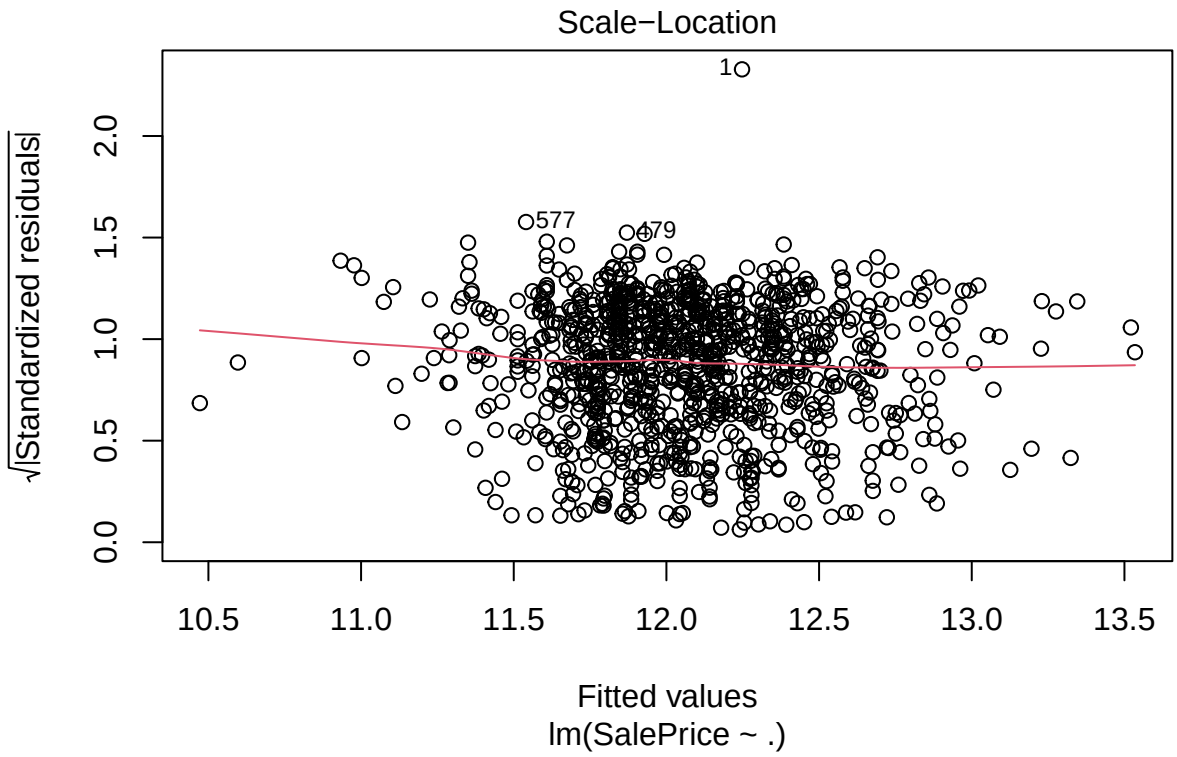
Street (p = 0.148) BsmtCond (p = 0.631) Fireplaces (p = 0.968) GarageType (p = 0.687)

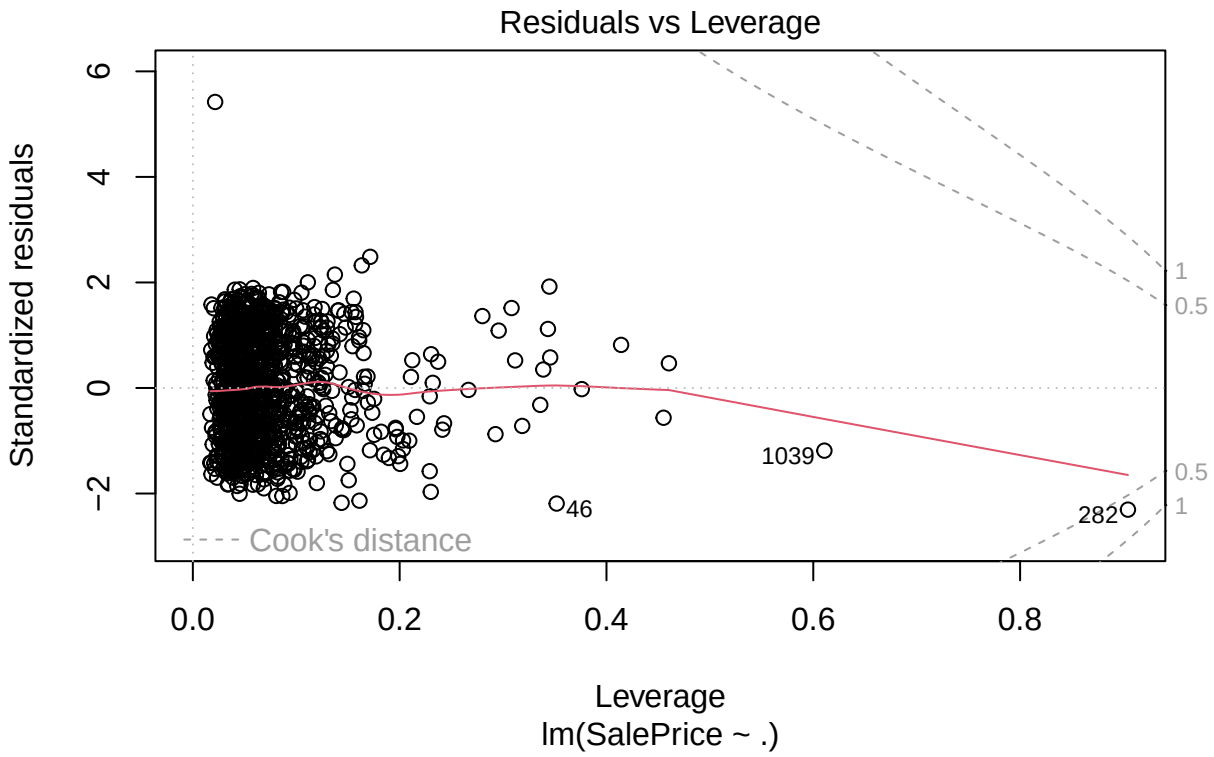
El problema de nuestro modelo es que es muy sobreajustado. Asi que vamos a ver si esta sobreajustado.

```
plot(modelo1)
```




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(SalePrice ~ .)
1
577
479
−2
0
2
4
6
−3
−2
−1
0
1
2
3


Scale−Location
1
577
479
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
Fitted values
lm(SalePrice ~ .)



Analisis Se puede ver que la varianza es constante, de hecho hay una tendencia a esta msima. Y se observa un patron en el grafico de residuales y muestra que la relacion no es completamete lineal, De hecho hay puntos de los mismos que deben analizarse porque puede ser que esto afecte la variabilidad del modelo y lo haga menos generalista. De hecho se ve en el grafico qq que no hay una tendencia lineal de los datos. Para ello haremos un un test de lilliefors si los residuos se distribuyen normalmente.

```
library(nortest)

lillie.test(modelo1$residuals)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  modelo1$residuals
## D = 0.057374, p-value = 8.927e-09
```

El p-valor es menor que 0.05 por lo que se rechaza la hipótesis nula de normalidad de los datos. Los residuos no están distribuidos normalmente.

Ahora lo que haremos es normalizarlos para ello realizamos el mismo procedimiento solo que ahora.

Normalizando los datos Dado que hay diferencias de escala en las variables por lo que vamos a normalizar los datos y hacer un modelo para ver si resulta mejor modelo.

```r
train_normal <- as.data.frame(scale(train))
```

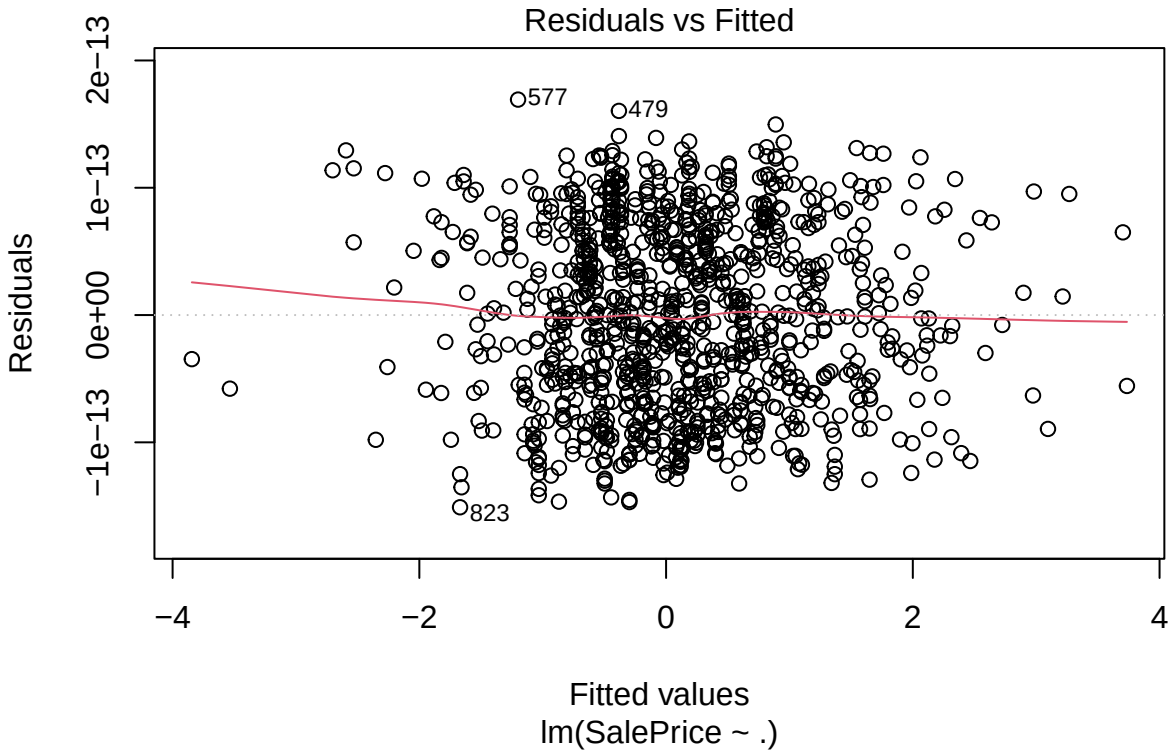
```r
modelo1.1<-lm(SalePrice~.,data = train_normal)
summary(modelo1.1)
```

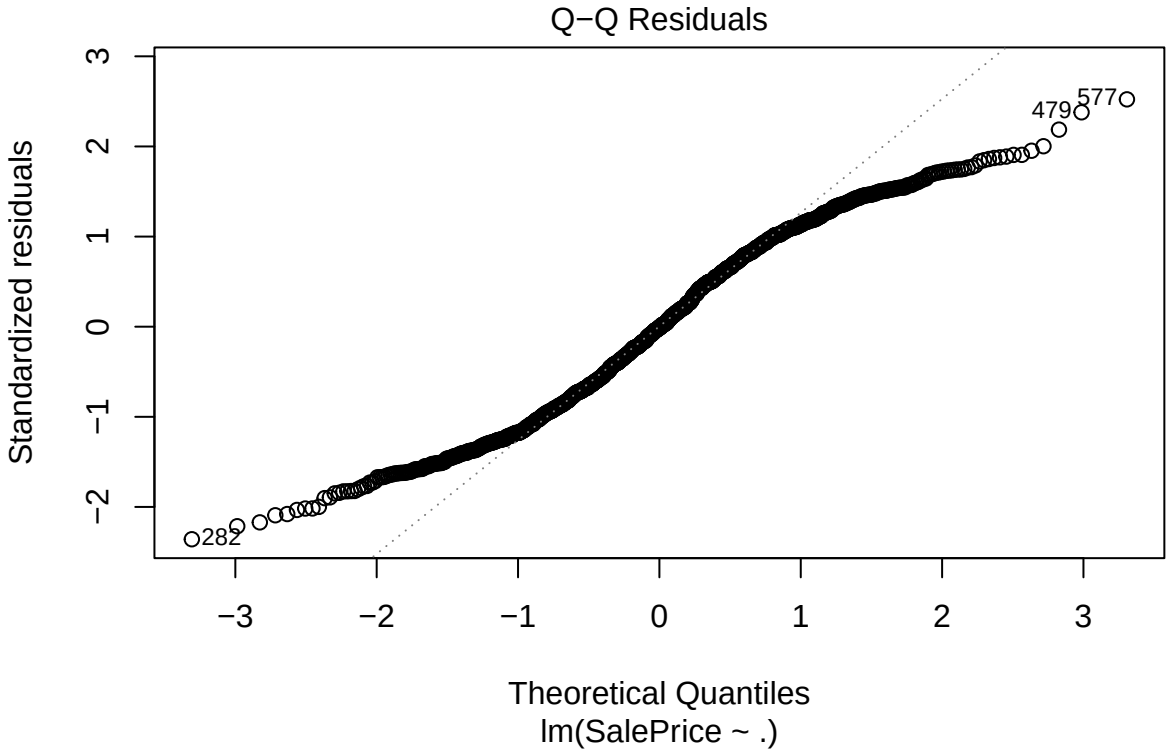
```
##
## Call:
## lm(formula = SalePrice ~ ., data = train_normal)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.510e-13 -5.934e-14 -4.940e-16  6.113e-14  1.693e-13
##
## Coefficients: (3 not defined because of singularities)
##                 Estimate Std. Error    t value Pr(>|t|)
## (Intercept)    1.250e-15  2.485e-15  5.030e-01   0.6152
## MSSubClass    -3.871e-15  5.688e-15 -6.810e-01   0.4963
## MSZoning       4.815e-15  2.627e-15  1.833e+00   0.0671 .
## LotFrontage   -3.322e-15  3.047e-15 -1.090e+00   0.2758
## LotArea        2.793e-15  3.063e-15  9.120e-01   0.3622
## Street         3.405e-16  2.708e-15  1.260e-01   0.8999
## LotShape      -7.980e-17  2.541e-15 -3.100e-02   0.9750
## LandContour   -1.358e-15  2.722e-15 -4.990e-01   0.6180
## Utilities     -3.248e-15  2.309e-15 -1.406e+00   0.1599
## LotConfig      3.188e-15  2.418e-15  1.318e+00   0.1877
## LandSlope     -2.134e-15  2.841e-15 -7.510e-01   0.4528
## Neighborhood  -4.252e-16  2.565e-15 -1.660e-01   0.8684
## Condition1     4.215e-15  2.396e-15  1.759e+00   0.0789 .
## Condition2     1.924e-15  2.298e-15  8.370e-01   0.4025
## BldgType       3.619e-15  4.997e-15  7.240e-01   0.4691
## HouseStyle     2.439e-15  3.595e-15  6.780e-01   0.4976
## OverallQual    7.882e-15  8.755e-15  9.000e-01   0.3682
## OverallCond   -2.399e-15  3.473e-15 -6.910e-01   0.4899
## YearBuilt      3.340e-15  6.959e-15  4.800e-01   0.6314
## YearRemodAdd   4.219e-15  4.101e-15  1.029e+00   0.3038
## RoofStyle      8.767e-16  2.565e-15  3.420e-01   0.7326
## RoofMatl      -3.242e-15  2.507e-15 -1.293e+00   0.1963
## Exterior1st   -7.112e-15  4.625e-15 -1.538e+00   0.1244
## Exterior2nd    6.903e-15  4.639e-15  1.488e+00   0.1370
## MasVnrType    -1.026e-15  2.618e-15 -3.920e-01   0.6951
## MasVnrArea    -4.104e-15  3.037e-15 -1.351e+00   0.1769
## ExterQual     -2.487e-15  3.986e-15 -6.240e-01   0.5327
## ExterCond     -4.386e-15  2.778e-15 -1.579e+00   0.1147
## Foundation    -3.076e-15  3.607e-15 -8.530e-01   0.3939
## BsmtQual      -1.258e-17  3.610e-15 -3.000e-03   0.9972
## BsmtCond       1.648e-15  2.591e-15  6.360e-01   0.5249
## BsmtExposure   2.826e-15  2.809e-15  1.006e+00   0.3145
## BsmtFinType1  -3.231e-15  3.176e-15 -1.017e+00   0.3092
## BsmtFinSF1    -4.702e-16  7.642e-15 -6.200e-02   0.9509
## BsmtFinType2   3.551e-15  3.561e-15  9.970e-01   0.3190
## BsmtFinSF2    -9.768e-16  4.311e-15 -2.270e-01   0.8208
## BsmtUnfSF     -4.414e-15  7.212e-15 -6.120e-01   0.5407
## TotalBsmtSF           NA         NA         NA       NA
```

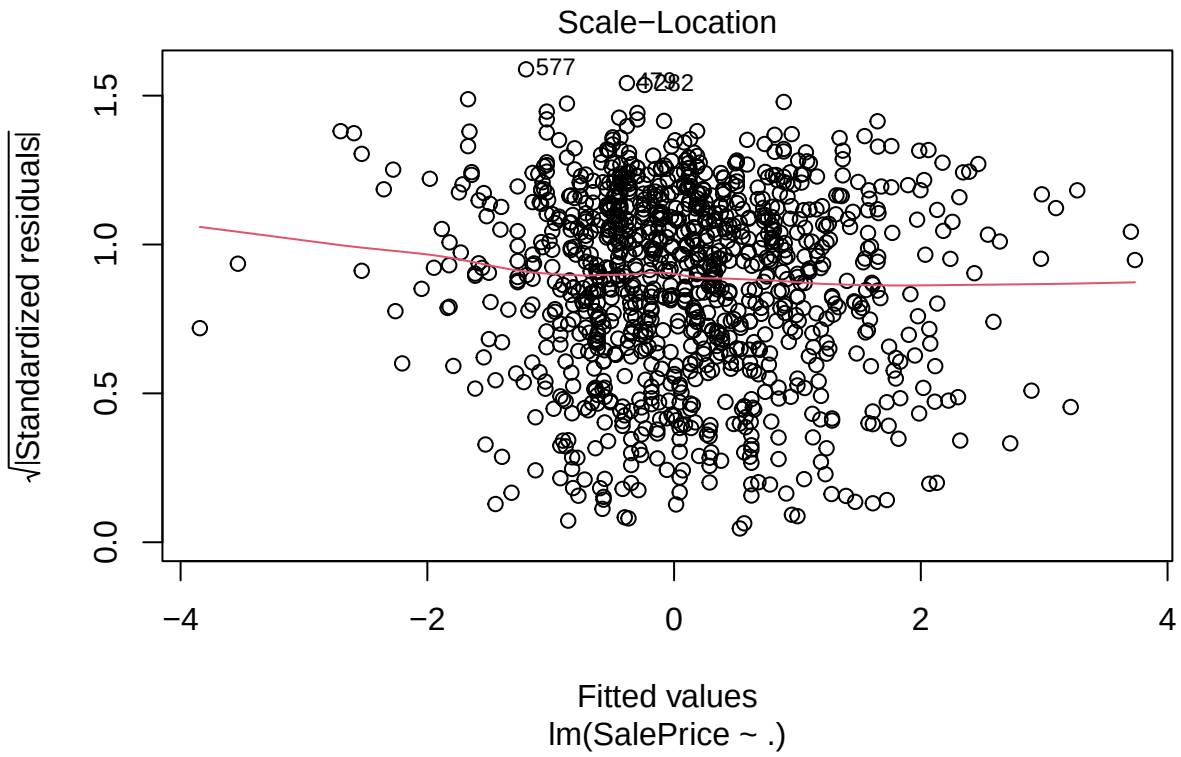
```
## Heating        1.011e-15  5.245e-15  1.930e-01   0.8471
## HeatingQC      1.060e-15  3.060e-15  3.470e-01   0.7290
## CentralAir    -1.793e-15  3.550e-15 -5.050e-01   0.6136
## Electrical    -7.318e-16  2.781e-15 -2.630e-01   0.7925
## X1stFlrSF     -2.587e-16  1.300e-14 -2.000e-02   0.9841
## X2ndFlrSF     -3.867e-15  1.414e-14 -2.730e-01   0.7845
## LowQualFinSF  -9.520e-16  2.975e-15 -3.200e-01   0.7491
## GrLivArea             NA         NA         NA       NA
## BsmtFullBath  -5.070e-16  3.531e-15 -1.440e-01   0.8859
## BsmtHalfBath   2.141e-16  2.499e-15  8.600e-02   0.9317
## FullBath      -3.659e-15  4.284e-15 -8.540e-01   0.3933
## HalfBath      -3.977e-16  3.551e-15 -1.120e-01   0.9108
## BedroomAbvGr   6.582e-15  3.986e-15  1.651e+00   0.0990 .
## KitchenAbvGr   1.963e-15  3.865e-15  5.080e-01   0.6115
## KitchenQual    4.175e-15  3.438e-15  1.214e+00   0.2250
## TotRmsAbvGrd  -5.538e-15  5.304e-15 -1.044e+00   0.2966
## Functional    -7.730e-16  2.707e-15 -2.860e-01   0.7753
## Fireplaces    -4.582e-16  2.966e-15 -1.540e-01   0.8773
## FireplaceQu   -3.247e-15  2.494e-15 -1.302e+00   0.1931
## GarageType     2.445e-15  3.450e-15  7.090e-01   0.4786
## GarageYrBlt   -9.721e-16  5.154e-15 -1.890e-01   0.8504
## GarageFinish  -3.686e-15  3.289e-15 -1.121e+00   0.2627
## GarageCars    -1.011e-14  5.708e-15 -1.771e+00   0.0768 .
## GarageArea     8.560e-15  5.597e-15  1.529e+00   0.1265
## GarageQual    -2.776e-15  3.001e-15 -9.250e-01   0.3553
## GarageCond     1.412e-15  2.925e-15  4.830e-01   0.6293
## PavedDrive    -2.096e-15  3.309e-15 -6.330e-01   0.5266
## WoodDeckSF     1.176e-15  2.562e-15  4.590e-01   0.6463
## OpenPorchSF   -1.373e-15  2.635e-15 -5.210e-01   0.6024
## EnclosedPorch -2.143e-15  2.616e-15 -8.190e-01   0.4129
## X3SsnPorch    -1.362e-15  2.244e-15 -6.070e-01   0.5442
## ScreenPorch    2.257e-15  2.401e-15  9.400e-01   0.3474
## PoolArea      -4.738e-15  2.385e-15 -1.987e+00   0.0472 *
## MiscVal       -1.836e-15  2.216e-15 -8.290e-01   0.4075
## MoSold        -1.372e-15  2.394e-15 -5.730e-01   0.5667
## YrSold        -4.080e-15  2.396e-15 -1.703e+00   0.0889 .
## SaleType      -3.181e-16  2.411e-15 -1.320e-01   0.8950
## SaleCondition -4.156e-15  2.624e-15 -1.584e+00   0.1136
## LogSalePrice   1.000e+00  7.261e-15  1.377e+14   <2e-16 ***
## QualityGroup  -1.617e-15  3.295e-15 -4.910e-01   0.6236
## SizeGroup     -3.903e-15  4.413e-15 -8.840e-01   0.3767
## Age                   NA         NA         NA       NA
## Qual_LivArea   1.589e-15  1.897e-14  8.400e-02   0.9333
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.371e-14 on 986 degrees of freedom
##   (105 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.274e+27 on 77 and 986 DF,  p-value: < 2.2e-16
```

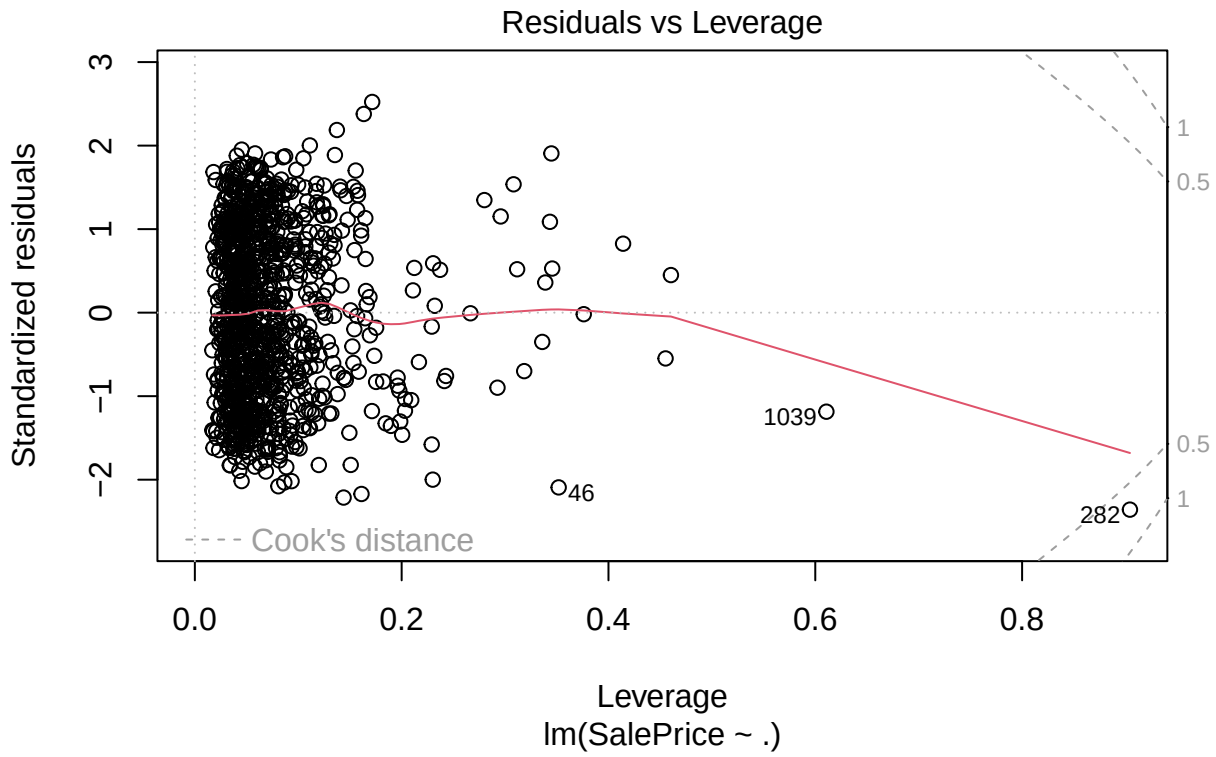
El modelo normalizado explica el 0.9997 de los datos.

```
plot(modelo1.1)
```




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(SalePrice ~ .)
282
479
577
−3
−2
−1
0
1
2
3


Scale−Location
577
478
582
√|Standardized residuals|
0.0
0.5
1.0
1.5
−4
−2
0
2
4
Fitted values
lm(SalePrice ~ .)



Analisis Grafico Parece haber una leve tendencia no lineal (curva roja), lo que indica que la relación entre las variables podría no ser completamente lineal.

Los residuales tienen una varianza no constante, es posible que se requieran correcciones como regresión ponderada o transformación de variables.

Se ve tambien una mejoria en el grafico q q en donde ahora vemos que sigue la curva la mayoria de variables aunque siguen habiendo varias que se desvian de la original.

Si verificamos con lillie test.

```
library(nortest)
lillie.test(modelo1.1$residuals)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  modelo1.1$residuals
## D = 0.059523, p-value = 1.749e-09
```

El pvalue sigue siendo mas pequeño a 0.05 lo que nos indica que tampoco sirvio normalizar.

Lo que vamos a hacer es seleccion de predictores con stepwise.

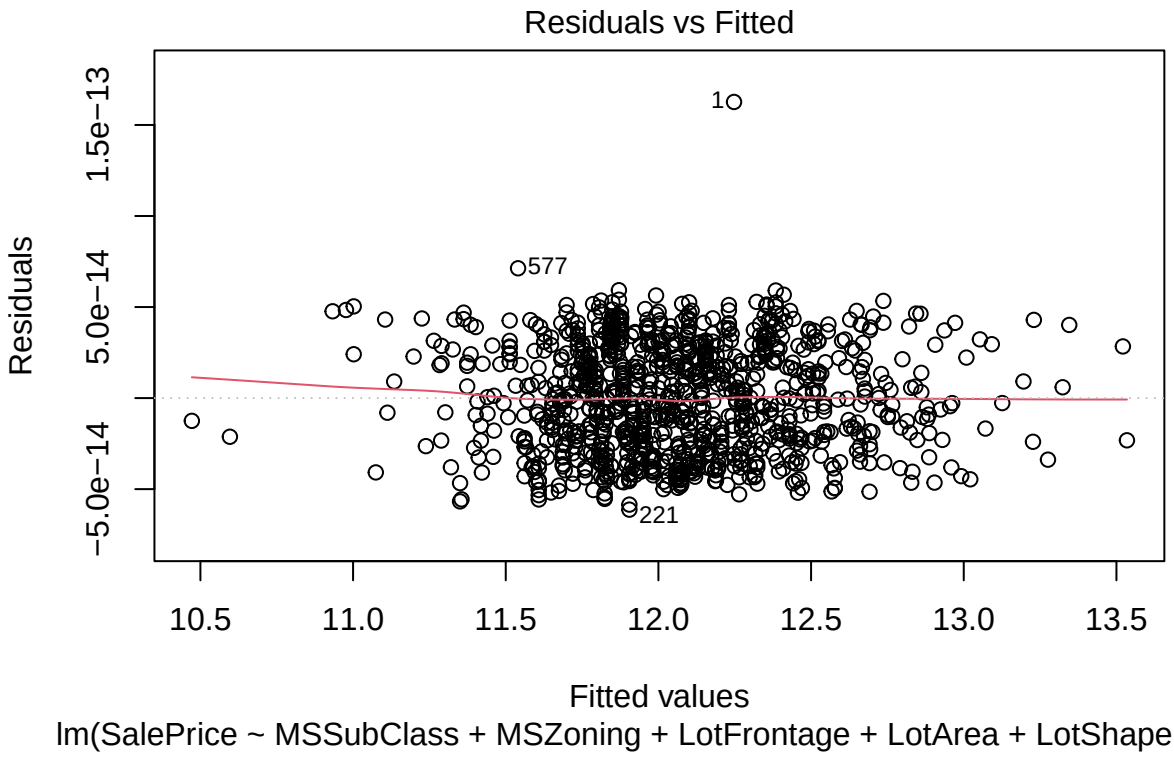
```r
# Eliminar filas con valores NA
train_clean <- na.omit(train)

# Ajustar modelo con stepwise backward
modelo2 <- step(
    object    = lm(formula = SalePrice ~ ., data = train_clean),
    direction = "backward",
    scope     = list(upper = ~., lower = ~1),
    trace     = FALSE
)
summary(modelo2)
```

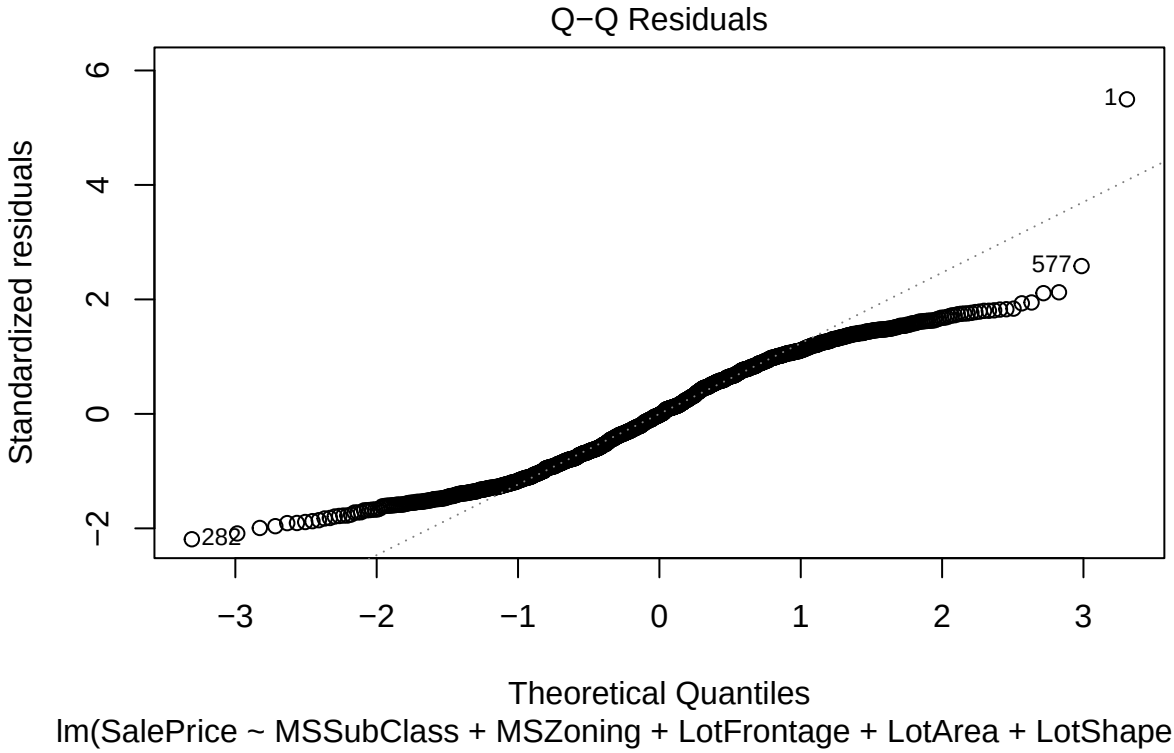
```
##
## Call:
## lm(formula = SalePrice ~ MSSubClass + MSZoning + LotFrontage +
##     LotArea + LotShape + LandContour + Utilities + LotConfig +
##     LandSlope + Neighborhood + Condition1 + Condition2 + BldgType +
##     HouseStyle + OverallQual + YearBuilt + YearRemodAdd + RoofMatl +
##     Exterior1st + Exterior2nd + MasVnrArea + ExterQual + ExterCond +
##     Foundation + BsmtExposure + BsmtFinType1 + BsmtFinType2 +
##     BsmtUnfSF + CentralAir + X1stFlrSF + FullBath + BedroomAbvGr +
##     KitchenAbvGr + KitchenQual + TotRmsAbvGrd + FireplaceQu +
##     GarageFinish + GarageCars + GarageArea + GarageQual + ScreenPorch +
##     PoolArea + MiscVal + YrSold + SaleCondition + LogSalePrice,
##     data = train_clean)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -6.125e-14 -2.391e-14 -4.770e-16  2.434e-14  1.626e-13
##
## Coefficients:
##                 Estimate Std. Error    t value Pr(>|t|)
## (Intercept)    2.137e-12  1.411e-12  1.515e+00   0.1301
## MSSubClass    -3.540e-17  5.233e-17 -6.770e-01   0.4989
## MSZoning       3.203e-15  1.675e-15  1.912e+00   0.0561 .
## LotFrontage   -7.205e-17  5.300e-17 -1.359e+00   0.1743
## LotArea        1.334e-19  1.814e-19  7.350e-01   0.4624
## LotShape       1.694e-16  7.201e-16  2.350e-01   0.8141
## LandContour   -6.463e-16  1.454e-15 -4.450e-01   0.6567
## Utilities     -4.360e-14  3.072e-14 -1.419e+00   0.1562
## LotConfig      9.203e-16  6.091e-16  1.511e+00   0.1311
## LandSlope     -2.545e-15  4.307e-15 -5.910e-01   0.5547
## Neighborhood  -4.228e-17  1.706e-16 -2.480e-01   0.8042
## Condition1     1.788e-15  1.070e-15  1.671e+00   0.0950 .
## Condition2     3.118e-15  3.914e-15  7.970e-01   0.4258
## BldgType       1.020e-15  1.686e-15  6.050e-01   0.5454
## HouseStyle     3.303e-16  7.333e-16  4.500e-01   0.6525
## OverallQual    2.917e-15  1.429e-15  2.042e+00   0.0415 *
## YearBuilt      6.365e-17  6.672e-17  9.540e-01   0.3403
## YearRemodAdd   5.807e-17  6.807e-17  8.530e-01   0.3938
## RoofMatl      -3.687e-15  2.764e-15 -1.334e+00   0.1826
## Exterior1st   -8.470e-16  5.681e-16 -1.491e+00   0.1363
## Exterior2nd    8.063e-16  5.202e-16  1.550e+00   0.1214
```

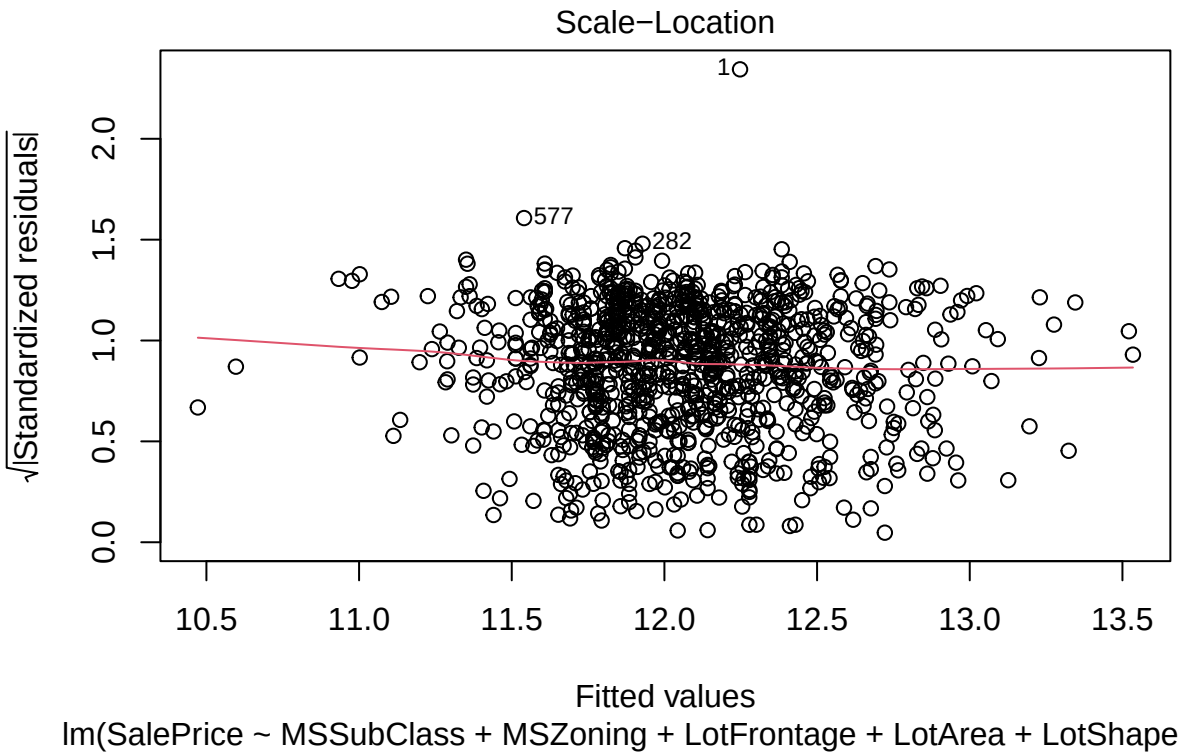
```
## MasVnrArea    -7.334e-18  6.005e-18 -1.221e+00   0.2223
## ExterQual     -1.438e-15  2.110e-15 -6.820e-01   0.4956
## ExterCond     -3.662e-15  2.655e-15 -1.380e+00   0.1680
## Foundation    -1.575e-15  1.966e-15 -8.010e-01   0.4234
## BsmtExposure   1.082e-15  9.220e-16  1.173e+00   0.2409
## BsmtFinType1  -6.571e-16  6.764e-16 -9.710e-01   0.3316
## BsmtFinType2   1.870e-15  1.012e-15  1.847e+00   0.0651 .
## BsmtUnfSF     -4.440e-18  3.038e-18 -1.462e+00   0.1442
## CentralAir    -3.054e-15  4.999e-15 -6.110e-01   0.5414
## X1stFlrSF      3.685e-18  4.164e-18  8.850e-01   0.3765
## FullBath      -2.837e-15  2.608e-15 -1.088e+00   0.2769
## BedroomAbvGr   2.914e-15  1.862e-15  1.565e+00   0.1179
## KitchenAbvGr   3.251e-15  6.481e-15  5.020e-01   0.6160
## KitchenQual    1.858e-15  1.644e-15  1.130e+00   0.2586
## TotRmsAbvGrd  -7.244e-16  1.136e-15 -6.380e-01   0.5239
## FireplaceQu   -1.282e-15  8.366e-16 -1.532e+00   0.1258
## GarageFinish  -1.573e-15  1.564e-15 -1.006e+00   0.3147
## GarageCars    -4.586e-15  2.943e-15 -1.558e+00   0.1195
## GarageArea     1.514e-17  9.410e-18  1.608e+00   0.1080
## GarageQual    -1.475e-15  1.662e-15 -8.880e-01   0.3750
## ScreenPorch    1.684e-17  1.604e-17  1.050e+00   0.2941
## PoolArea      -5.094e-17  2.379e-17 -2.141e+00   0.0325 *
## MiscVal       -1.635e-18  1.847e-18 -8.850e-01   0.3763
## YrSold        -1.130e-15  6.993e-16 -1.616e+00   0.1065
## SaleCondition -1.610e-15  9.771e-16 -1.647e+00   0.0998 .
## LogSalePrice   1.000e+00  6.211e-15  1.610e+14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.978e-14 on 1017 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 3.807e+27 on 46 and 1017 DF,  p-value: < 2.2e-16
```

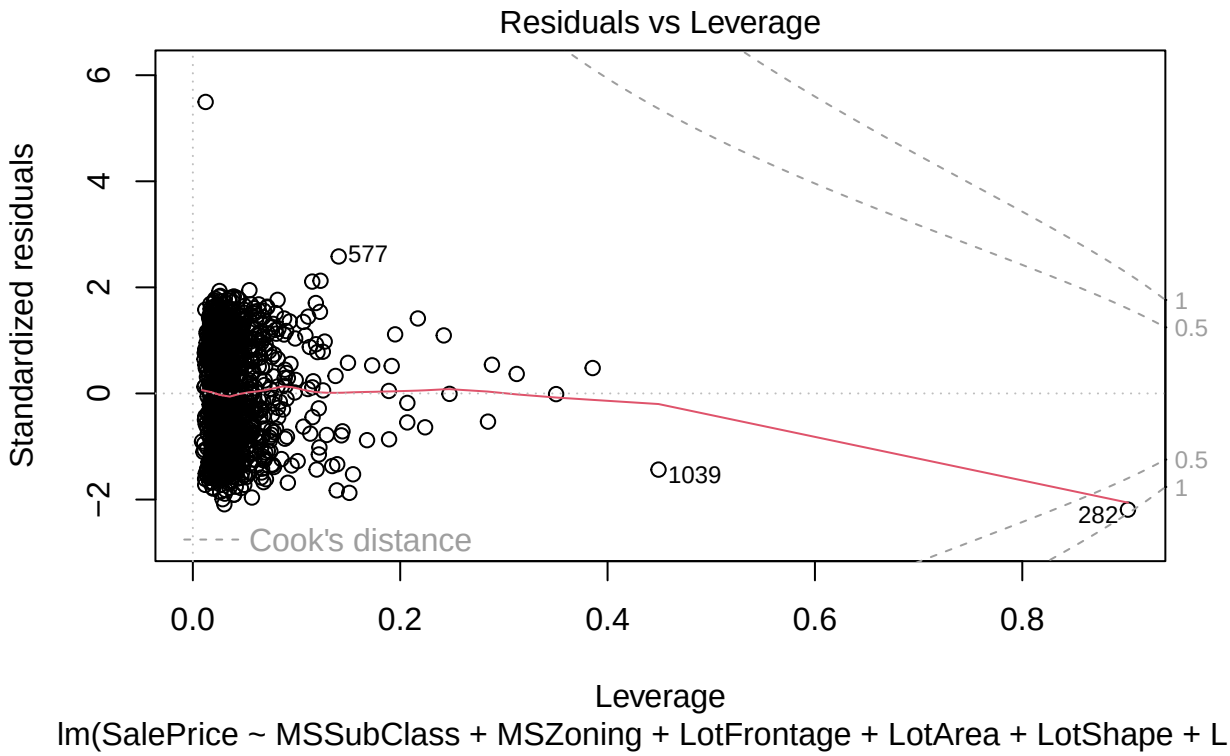
Ahora vamos a ver los residuos.

```
plot(modelo2)
```


Residuals vs Fitted
Residuals
1.5e−13
5.0e−14
−5.0e−14
1
577
221
10.5
11.0
11.5
12.0
12.5
13.0
13.5
Fitted values
lm(SalePrice ~ MSSubClass + MSZoning + LotFrontage + LotArea + LotShape + L .


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(SalePrice ~ MSSubClass + MSZoning + LotFrontage + LotArea + LotShape + L .
1
577
282
−3
−2
−1
0
1
2
3
−2
0
2
4
6


Scale−Location
1
577
282
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
Fitted values
lm(SalePrice ~ MSSubClass + MSZoning + LotFrontage + LotArea + LotShape + L .



Analisis Podemos ver que ha mejorado un poco en la manera en como se muestran los residuos, Pero siguen habiendo valores atipicos aun con eliminar varios que dijimos que con stepwise se mejoraria.

```
library(nortest)
lillie.test(modelo2$residuals)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  modelo2$residuals
## D = 0.057556, p-value = 7.792e-09
```

Sigue sin mejorar aun usando el stepwise. Por lo que debemos de analizar la multicolinealidad de las variables y ver cuales debemos de seleccionar. Pero antes miremos como predice este modelo.

```
test_d[is.na(test_d)] <- 0
for (col in names(test_d)) {
  test_d[[col]][is.na(test_d[[col]])] <- median(test_d[[col]], na.rm = TRUE)
}

train <- train[!is.na(train$SalePrice), ]

predicciones_train <- predict(modelo2, newdata = train)
predicciones_test <- predict(modelo2, newdata = test_d)
```

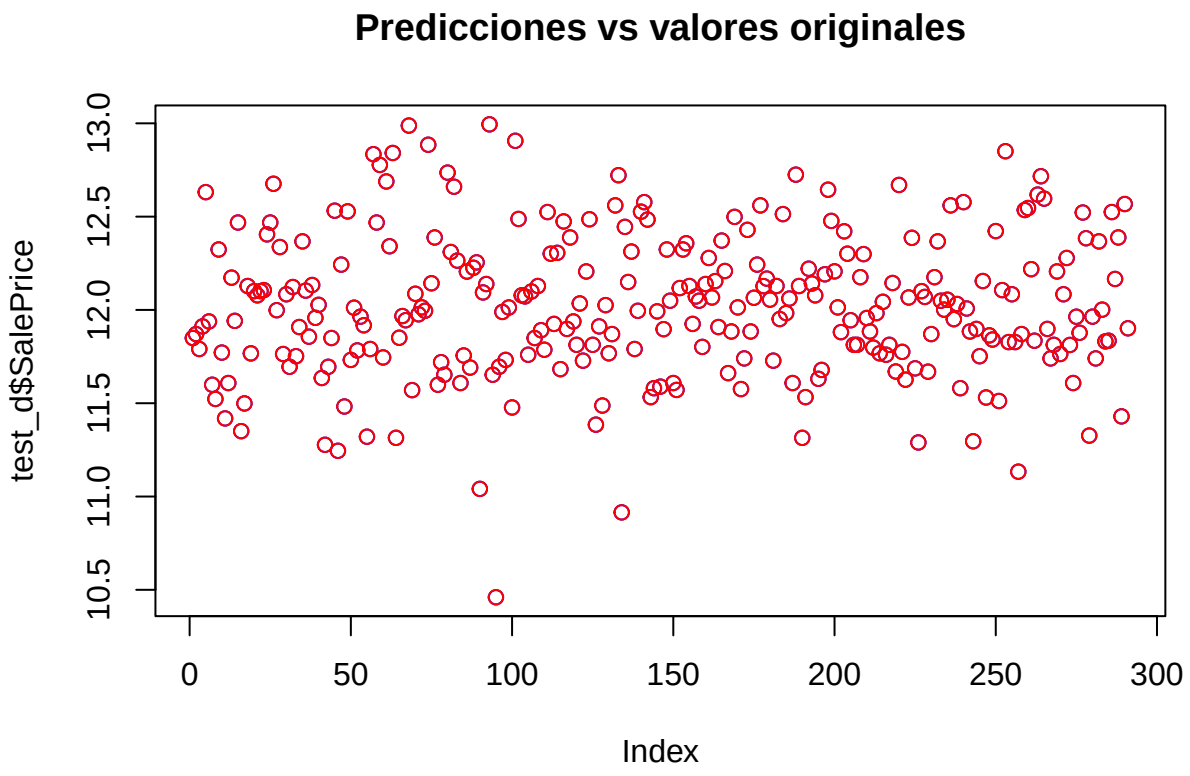
```r
predicciones_train <- as.integer(predicciones_train)


training_stepwise <- mean((predicciones_train - train$SalePrice)^2, na.rm = TRUE)

test_mse_stepwise <- mean((predicciones_test - test_d$SalePrice)^2, na.rm = TRUE)


print(training_stepwise)
```

```
## [1] 0.3630067
```

```r
print(test_mse_stepwise)
```

```
## [1] 8.96905e-28
```

```r
faltantes <- setdiff(names(train), names(test_d))
for (col in faltantes) {
  test_d[[col]] <- NA
}


plot(test_d$SalePrice,col="blue", main="Predicciones vs valores originales")
points(predicciones_test, col="red")
legend(30,45,legend=c("original", "prediccion"),col=c("blue", "red"),pch=1, cex=0.8)
```



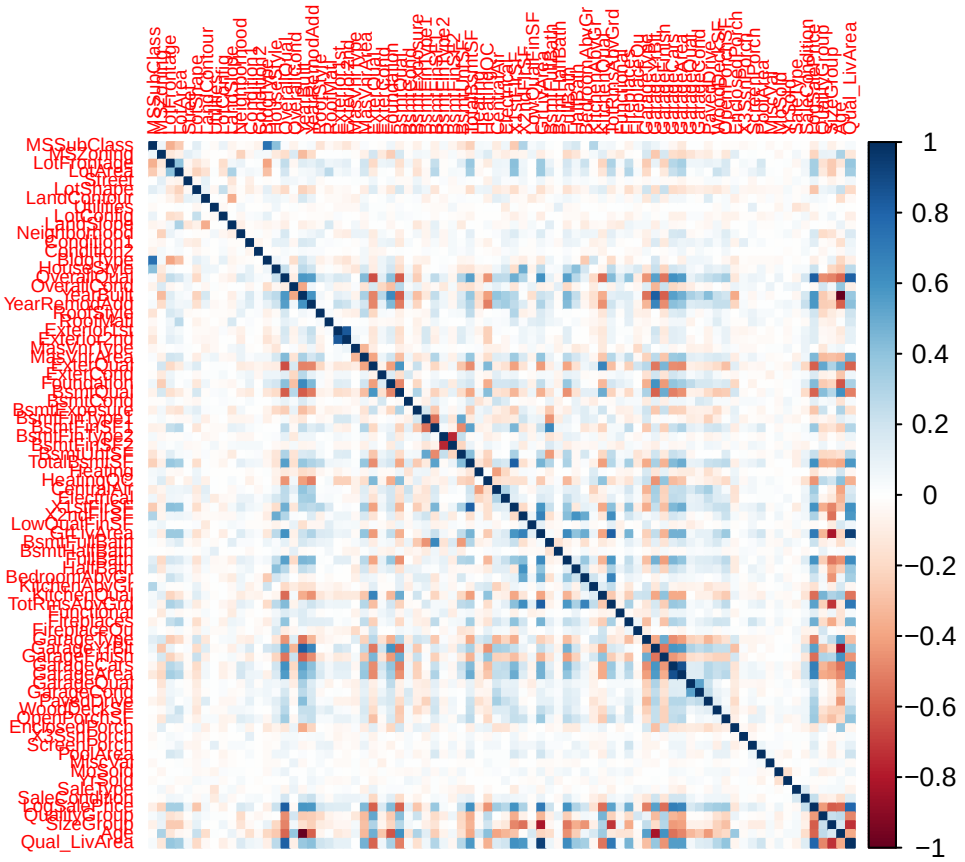
Analisis Al final el modelo 1 no es muy predictivo y al ver los siguientes solo estamos haciendo que sea menos predictivo. Por ello mismo lo que vamos a hacer es utilizar otros metodos para poder eliminar variables y reducir nuestro overfitting del modelo. Pero antes miremos la multicolineadlidad de los datos.

9. Analice el modelo. Determine si hay multicolinealidad entre las variables, y cuáles son las que aportan al modelo. Haga un análisis de correlación de las características del modelo y especifique si el modelo se adapta bien a los datos. Explique si hay sobreajuste (overfitting) o no. En caso de existir sobreajuste, haga otro modelo que lo corrija.

Podemos ver que nuestro modelo lastimosamente si esta sobreajustado ya que su R ajustado es de 0.97 y de 1 si realizamos stepwise . Ademas que podemos ver que al predecir con los test no lo esta haciendo de la manera correcta aun cuando normalicemos , y apliquemos tecnicas para reducir datos atipicos.

- Lo primero que vamos a hacer es usar la matriz de correlacion.

```
library(car)


cor_matrix <- cor(train[, -which(names(train) == "SalePrice")], use =
 ↪  "pairwise.complete.obs")


library(corrplot)
corrplot(cor_matrix, method = "color", tl.cex = 0.6, tl.srt = 90)
```



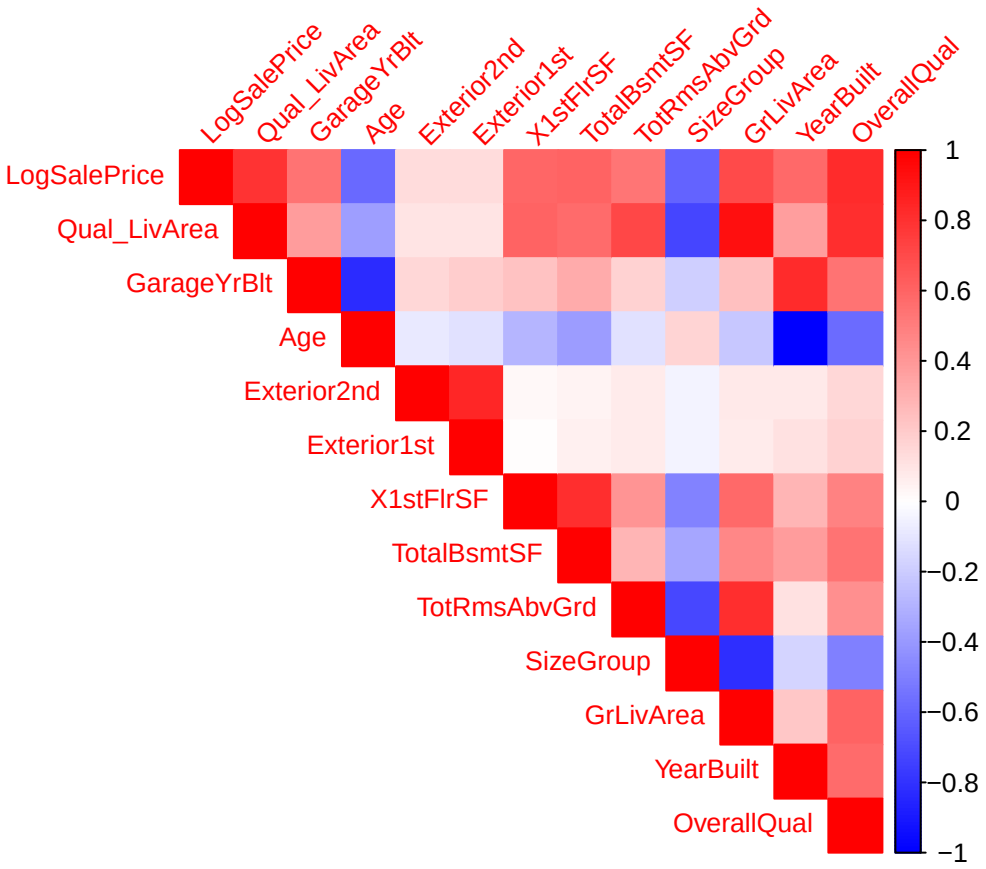
Ahora conocemos la correlacion alta.

```
high_cor <- which(abs(cor_matrix) > 0.8, arr.ind = TRUE)


high_cor <- high_cor[high_cor[,1] != high_cor[,2], ]
print(high_cor)
```

```
##              row col
## LogSalePrice  76  16
## Qual_LivArea  80  16
## GarageYrBlt   58  18
## Age           79  18
## Exterior2nd   23  22
## Exterior1st   22  23
## X1stFlrSF     42  37
## TotalBsmtSF   37  42
## TotRmsAbvGrd  53  45
## SizeGroup     78  45
## Qual_LivArea  80  45
## GrLivArea     45  53
## YearBuilt     18  58
## Age           79  58
## GarageArea    61  60
## GarageCars    60  61
## OverallQual   16  76
## GrLivArea     45  78
## YearBuilt     18  79
## GarageYrBlt   58  79
## OverallQual   16  80
## GrLivArea     45  80
```

Como podemos ver estas variables estan causando alta coinealidad de hecho podemos graficarlas con una matriz de correlacion

```
top_vars <- c("LogSalePrice", "Qual_LivArea", "GarageYrBlt", "Age", "Exterior2nd",
              "Exterior1st", "X1stFlrSF", "TotalBsmtSF", "TotRmsAbvGrd", "SizeGroup",
              "GrLivArea", "YearBuilt", "OverallQual")


filtered_train <- train[, top_vars]


cor_matrix_filtered <- cor(filtered_train, use = "pairwise.complete.obs")


library(corrplot)
corrplot(cor_matrix_filtered, method = "color", type = "upper",
         tl.cex = 0.8, tl.srt = 45,
         col = colorRampPalette(c("blue", "white", "red"))(200),
         cl.cex = 0.8)
```



Pero antes de eliminar variables multicolineales vamos a hacer una cosa. Utilizaremos Ridge y regularicemos para ver que tan bien rinden.

```
x_train <- model.matrix(SalePrice~., data = train)
y_train <- train$SalePrice

x_test <- model.matrix(SalePrice~., data = test_d)
y_test <- test_d$SalePrice
```

```
# Asegurar que ambos tengan las mismas filas



set.seed(123)  # Para reproducibilidad
indices <- sample(1:nrow(x_train), 1064)  # Seleccionar 1064 índices aleatorios

x_train <- x_train[indices, ]  # Filtrar filas de x_train
y_train <- y_train[indices]  # Filtrar elementos de y_train



# Variables en train y test
vars_train <- colnames(x_train)
vars_test <- colnames(x_test)

# Variables que están en test pero no en train
```

```
extra_vars_in_test <- setdiff(vars_test, vars_train)
print(extra_vars_in_test)
```

```
## character(0)
```

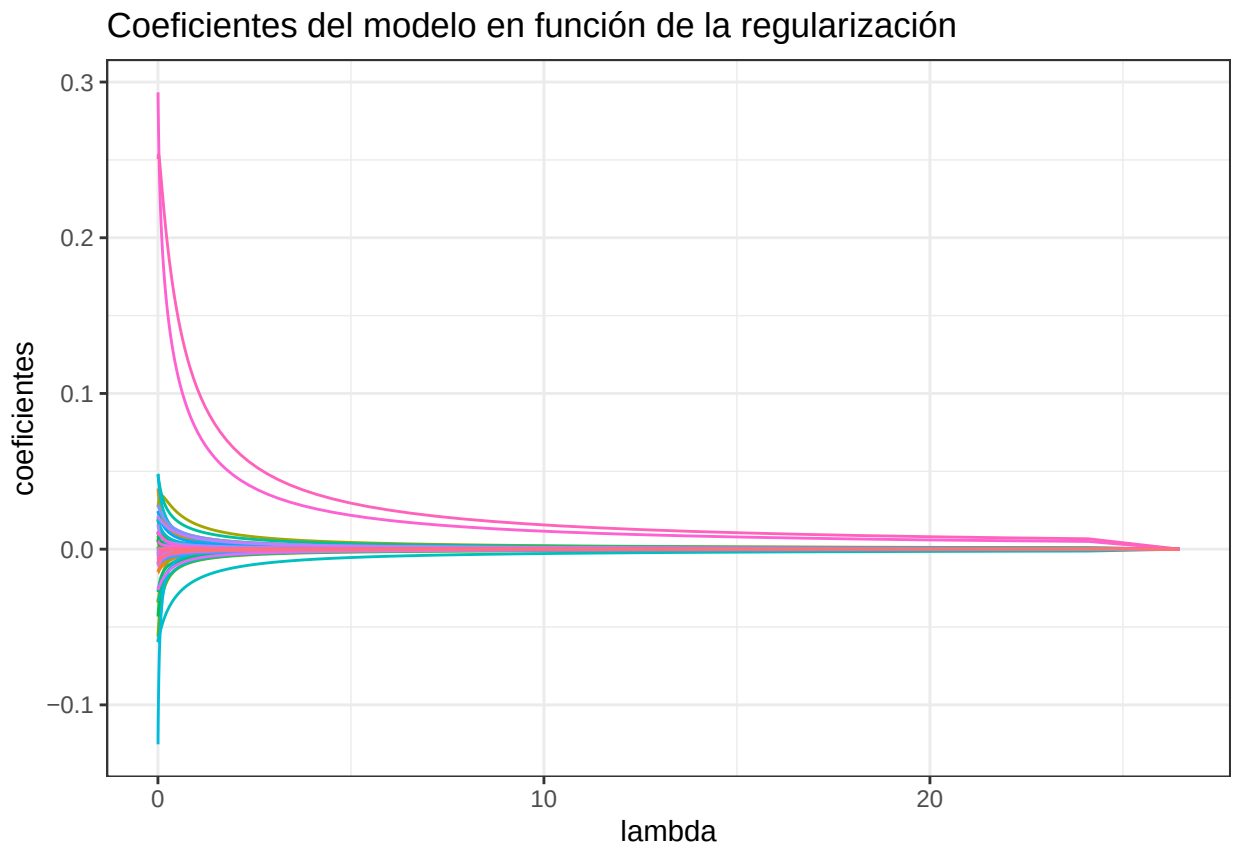
```
x_test <- x_test[, colnames(x_train)]
```

Se hace la regularización con un 100 números de porque no sabemos cual es el lambda adecuado.

```
library(tidyverse)
library(glmnet)
modelo3 <- glmnet(
            x           = x_train,
            y           = y_train,
            alpha       = 0,
            nlambda     = 100,
            standardize = TRUE
          )

regularizacion <- modelo3$beta %>%
                  as.matrix() %>%
                  t() %>%
                  as_tibble() %>%
                  mutate(lambda = modelo3$lambda)

regularizacion <- regularizacion %>%
                   pivot_longer(
                     cols = !lambda,
                     names_to = "predictor",
                     values_to = "coeficientes"
                   )
```

Ahora graficamos loos coeficientes

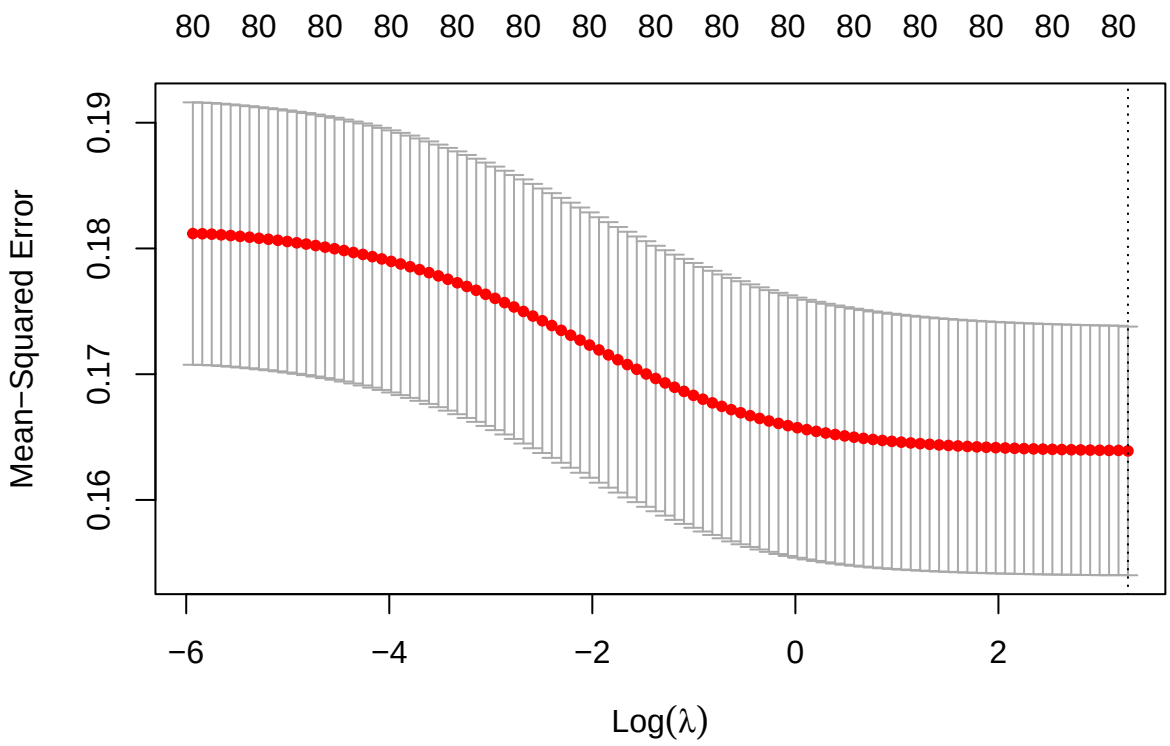
```
 regularizacion %>%
  ggplot(aes(x = lambda, y = coeficientes, color = predictor)) +
  geom_line() +
  labs(title = "Coeficientes del modelo en función de la regularización") +
  theme_bw() +
  theme(legend.position = "none")
```



Analisis Como podemos ver el valor de lambda indica que mientras mas pequeños los coeficientes mayor es lambda y mientras mas grande los coeficientes mayor el lambda.

Ahora obtendremos el valor optimo de este con la validacion cruzada

```
cv_error <- cv.glmnet(
              x      = x_train,
              y      = y_train,
              alpha  = 0,
              nfolds = 10,
              type.measure = "mse",
              standardize  = TRUE
            )

plot(cv_error)
```




```
lambda_opt <- cv_error$lambda.min
print(lambda_opt)
```

```
## [1] 26.461
```

Vemos que el valor es de 26.461 asi que con estos nuevos datos vamos a entrenar nuestro modelo

```
modelo3 <- glmnet(
            x           = x_train,
            y           = y_train,
            alpha       = 0,
            lambda      = cv_error$lambda.1se,
            standardize = TRUE
          )
```

Ahora vemos los predictores y sus coeficientes. Ridge no elimina variables

```
coef(modelo3)
```

```
## 82 x 1 sparse Matrix of class "dgCMatrix"
##                              s0
## (Intercept)    1.207382e+01
## (Intercept)    .
```

```
## MSSubClass     4.515121e-06
## MSZoning       4.179668e-04
## LotFrontage   -5.409408e-06
## LotArea        3.073674e-09
## Street         4.525338e-03
## LotShape      -1.471795e-04
## LandContour    1.314613e-04
## Utilities      6.075601e-03
## LotConfig     -6.331918e-05
## LandSlope      2.620790e-04
## Neighborhood  -2.220386e-05
## Condition1    -1.534585e-04
## Condition2     8.772832e-04
## BldgType       7.211048e-05
## HouseStyle     1.121914e-04
## OverallQual    6.834960e-05
## OverallCond    1.416514e-04
## YearBuilt      2.397077e-06
## YearRemodAdd   7.260293e-06
## RoofStyle     -5.690055e-05
## RoofMatl      -2.441608e-04
## Exterior1st   -1.993581e-05
## Exterior2nd   -3.912711e-05
## MasVnrType     1.665318e-04
## MasVnrArea     1.900151e-06
## ExterQual     -4.125442e-04
## ExterCond     -1.741876e-04
## Foundation     1.827276e-04
## BsmtQual      -2.072784e-05
## BsmtCond      -1.470414e-04
## BsmtExposure  -1.580206e-04
## BsmtFinType1  -1.555944e-04
## BsmtFinSF1     5.104843e-07
## BsmtFinType2   2.911629e-04
## BsmtFinSF2    -5.394616e-07
## BsmtUnfSF     -5.742780e-07
## TotalBsmtSF   -1.031241e-07
## Heating       -3.165736e-04
## HeatingQC     -4.996924e-05
## CentralAir     1.525459e-04
## Electrical     5.349539e-05
## X1stFlrSF     -2.914469e-07
## X2ndFlrSF      6.896630e-07
## LowQualFinSF   2.861557e-06
## GrLivArea      3.319051e-07
## BsmtFullBath   1.544503e-04
## BsmtHalfBath   4.623414e-04
## FullBath      -1.637696e-05
## HalfBath       7.703701e-04
## BedroomAbvGr   7.601917e-05
## KitchenAbvGr  -1.103785e-03
## KitchenQual   -1.049706e-04
## TotRmsAbvGrd   5.156368e-05
## Functional     1.195977e-04
```

```
## Fireplaces     1.450667e-04
## FireplaceQu   -7.341181e-05
## GarageType     1.015117e-04
## GarageYrBlt    2.498659e-06
## GarageFinish  -1.060570e-04
## GarageCars    -1.508689e-04
## GarageArea    -8.549341e-08
## GarageQual    -2.075869e-04
## GarageCond    -3.283612e-05
## PavedDrive     4.357385e-04
## WoodDeckSF    -9.427514e-07
## OpenPorchSF   -3.010105e-06
## EnclosedPorch  3.805465e-06
## X3SsnPorch     3.057135e-06
## ScreenPorch    5.792205e-06
## PoolArea      -2.027554e-06
## MiscVal       -2.470766e-07
## MoSold        -1.370612e-04
## YrSold        -4.477718e-05
## SaleType      -6.633982e-05
## SaleCondition  6.043839e-05
## LogSalePrice   1.623581e-04
## QualityGroup  -1.117004e-04
## SizeGroup     -1.041487e-04
## Age           -2.482621e-06
## Qual_LivArea   3.621051e-08
```

```
predicciones_train_modelo3<- predict(modelo3, newx = x_train)


predicciones_test_modelo3 <- predict(modelo3, newx = x_test)



y_test <- y_test[1:nrow(x_test)]

# MSE de test
test_mse_ridge_modelo3 <- mean((predicciones_test_modelo3 - y_test)^2)

# MSE de entrenamiento
training_mse_ridge_modelo3 <- mean((predicciones_train_modelo3 - y_train)^2)


print(test_mse_ridge_modelo3)
```

```
## [1] 0.1436333
```

```
print(training_mse_ridge_modelo3)
```

```
## [1] 0.1631888
```

Analisis Ridge muestra que el modelo quedo bastante bien, aunque puede ser no muy bueno debido a que uno muy bajo indica sobreajuste. Pero sabemos por teoria que Ridge solo elimina sobreajuste acercando

valores a 0 pero no eliminandolos que es lo que vimos en el grafico de los lambdas encontrados.

```
predicciones_train_modelo3 <- predict(modelo3, newx = x_train)
predicciones_test_modelo3 <- predict(modelo3, newx = x_test)


y_test <- y_test[1:nrow(x_test)]

# Calcular MSE
test_mse_ridge_modelo3 <- mean((predicciones_test_modelo3 - y_test)^2)
training_mse_ridge_modelo3 <- mean((predicciones_train_modelo3 - y_train)^2)

# Calcular residuos
residuos_train <- y_train - predicciones_train_modelo3
residuos_test <- y_test - predicciones_test_modelo3


print(test_mse_ridge_modelo3)
```

```
## [1] 0.1436333
```

```
print(training_mse_ridge_modelo3)
```

```
## [1] 0.1631888
```

```
summary(residuos_train)
```

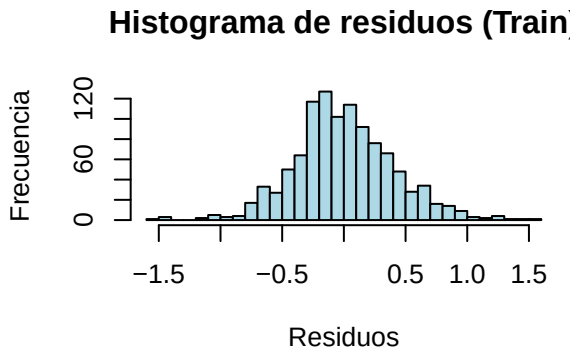
```
##        s0
##  Min.   :-1.5540
##  1st Qu.:-0.2500
##  Median :-0.0226
##  Mean   : 0.0000
##  3rd Qu.: 0.2452
##  Max.   : 1.5055
```

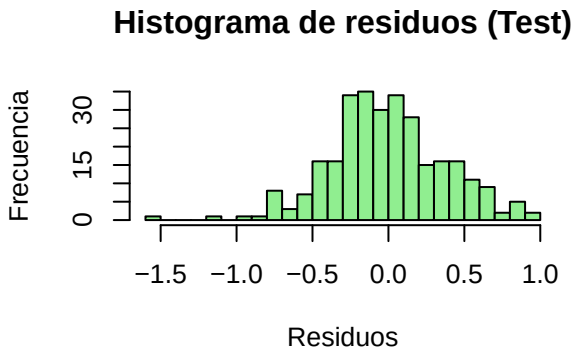
```
summary(residuos_test)
```

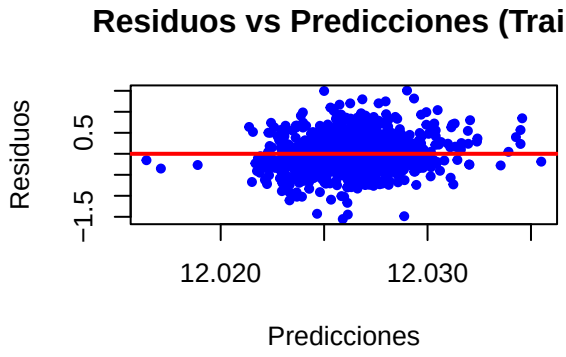
```
##        s0
##  Min.   :-1.554982
##  1st Qu.:-0.251913
##  Median :-0.023444
##  Mean   :-0.006019
##  3rd Qu.: 0.232994
##  Max.   : 0.967486
```

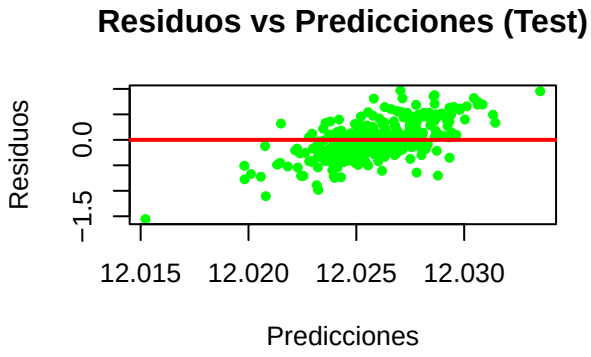
```
par(mfrow=c(2,2))


hist(residuos_train, main="Histograma de residuos (Train)",
     col="lightblue", breaks=30, xlab="Residuos", ylab="Frecuencia")
```

```r
hist(residuos_test, main="Histograma de residuos (Test)",
     col="lightgreen", breaks=30, xlab="Residuos", ylab="Frecuencia")


plot(predicciones_train_modelo3, residuos_train,
     main="Residuos vs Predicciones (Train)",
     xlab="Predicciones", ylab="Residuos", pch=20, col="blue")
abline(h=0, col="red", lwd=2)


plot(predicciones_test_modelo3, residuos_test,
     main="Residuos vs Predicciones (Test)",
     xlab="Predicciones", ylab="Residuos", pch=20, col="green")
abline(h=0, col="red", lwd=2)
```










```r
# Resetear layout
par(mfrow=c(1,1))
```

Analisis Podemos ver que a diferencia del anterior que habiamos hecho con Stepwise ahora si vemos que los residuos ya no tienen un patron o forma clara. Lo que indica que el modelo esta haciendo buena estimacion de los valores que deberia tener. Lo que indica que podria ser un muy buen modelo . Aunque se puede ver

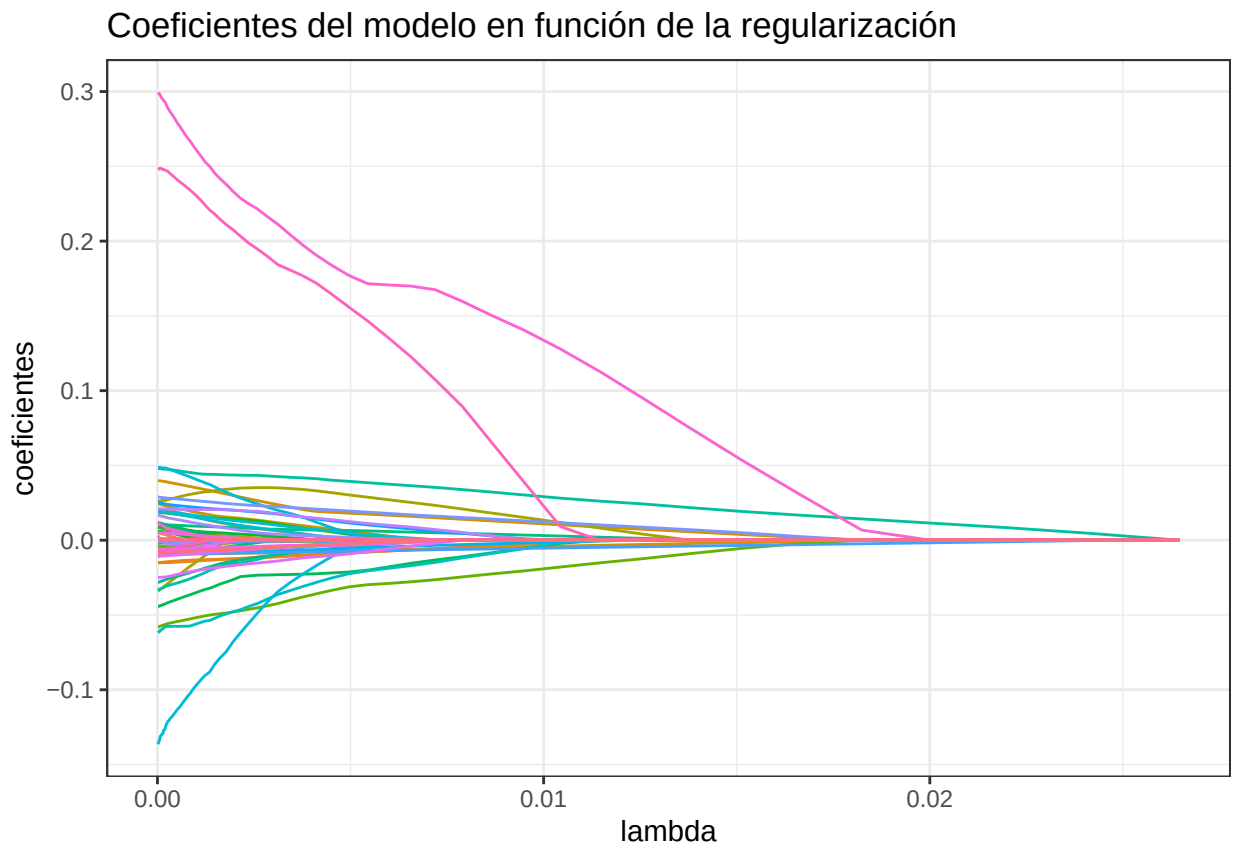
que el histograma de Residuos esta sesgado a la derecha , posiblemente estamos viendo un valor atipico a la izquierda. O un desbalance de nuestras predicciones en test.

10. Si tiene multicolinealidad o sobreajuste, haga un modelo con las variables que sean mejores predictoras del precio de las casas. Determine la calidad del modelo realizando un análisis de los residuos. Muéstrelo gráficamente.

Viendo en stepwise y al normalizar si hay sobreajuste asi que vamos a realizar Lazz y vamos a eliminar aquellas que sean peores predictoras o que no estan aportando a la prediccion de resultados.

Para lazoo necesitaremos alpha

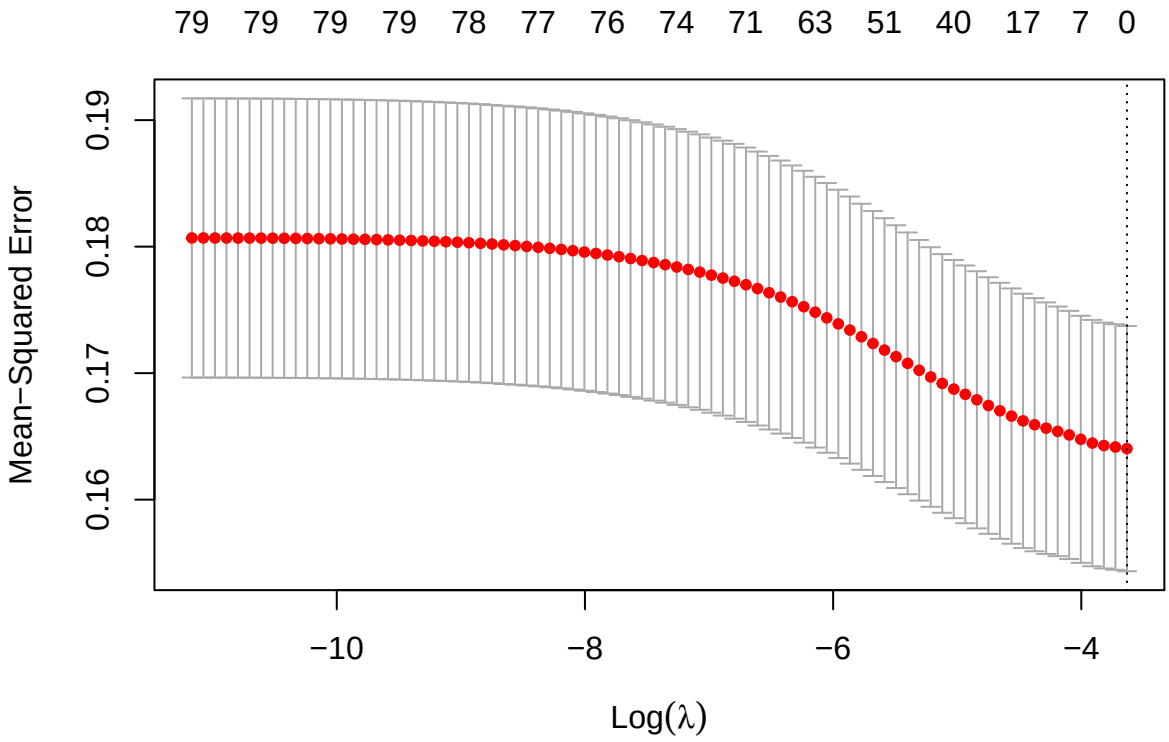
```
modelo4 <- glmnet(
            x           = x_train,
            y           = y_train,
            alpha       = 1,
            nlambda     = 100,
            standardize = TRUE
          )

regularizacion <- modelo4$beta %>%
                  as.matrix() %>%
                  t() %>%
                  as_tibble() %>%
                  mutate(lambda = modelo4$lambda)

regularizacion <- regularizacion %>%
                   pivot_longer(
                     cols = !lambda,
                     names_to = "predictor",
                     values_to = "coeficientes"
                   )

regularizacion %>%
  ggplot(aes(x = lambda, y = coeficientes, color = predictor)) +
  geom_line() +
  labs(title = "Coeficientes del modelo en función de la regularización") +
  theme_bw() +
  theme(legend.position = "none")
```



Analisis Este grafico indica lo mismo que con Ridge solo que vemos que en muy pocos valores de lambda estamos viendo que los coeficientes bajen abruptamente .

Ahora veamos el mejor valor de lambda con validacion cruzada

```
cv_error <- cv.glmnet(
              x      = x_train,
              y      = y_train,
              alpha  = 1,
              nfolds = 10,
              type.measure = "mse",
              standardize  = TRUE
            )

plot(cv_error)
```




```
lambda_opt2 <- cv_error$lambda.min
print(lambda_opt)
```

```
## [1] 26.461
```

Analisis Vemos que el valor mas optimo sigue siendo el mismo que el anterior de 26 de hecho hasta podemos ver que con respecto a los coeficientes no cambia tanto.

```
modelo4 <- glmnet(
            x           = x_train,
            y           = y_train,
            alpha       = 1,
            lambda      = cv_error$lambda.1se,
            standardize = TRUE
          )
coef(modelo4)
```

```
## 82 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept)   12.02623
## (Intercept)    0.00000
## MSSubClass     .
## MSZoning       .
## LotFrontage    .
```

```
## LotArea        .
## Street         .
## LotShape       .
## LandContour    .
## Utilities      .
## LotConfig      .
## LandSlope      .
## Neighborhood   .
## Condition1     .
## Condition2     .
## BldgType       .
## HouseStyle     .
## OverallQual    .
## OverallCond    .
## YearBuilt      .
## YearRemodAdd   .
## RoofStyle      .
## RoofMatl       .
## Exterior1st    .
## Exterior2nd    .
## MasVnrType     .
## MasVnrArea     .
## ExterQual      .
## ExterCond      .
## Foundation     .
## BsmtQual       .
## BsmtCond       .
## BsmtExposure   .
## BsmtFinType1   .
## BsmtFinSF1     .
## BsmtFinType2   .
## BsmtFinSF2     .
## BsmtUnfSF      .
## TotalBsmtSF    .
## Heating        .
## HeatingQC      .
## CentralAir     .
## Electrical     .
## X1stFlrSF      .
## X2ndFlrSF      .
## LowQualFinSF   .
## GrLivArea      .
## BsmtFullBath   .
## BsmtHalfBath   .
## FullBath       .
## HalfBath       .
## BedroomAbvGr   .
## KitchenAbvGr   .
## KitchenQual    .
## TotRmsAbvGrd   .
## Functional     .
## Fireplaces     .
## FireplaceQu    .
## GarageType     .
```

```
## GarageYrBlt    .
## GarageFinish   .
## GarageCars     .
## GarageArea     .
## GarageQual     .
## GarageCond     .
## PavedDrive     .
## WoodDeckSF     .
## OpenPorchSF    .
## EnclosedPorch  .
## X3SsnPorch     .
## ScreenPorch    .
## PoolArea       .
## MiscVal        .
## MoSold         .
## YrSold         .
## SaleType       .
## SaleCondition  .
## LogSalePrice   .
## QualityGroup   .
## SizeGroup      .
## Age            .
## Qual_LivArea   .
```

Al final solo nos quedamos con 2 esto es normal para este tipo de seleccion de variables

```
predicciones_train_modelo4<- predict(modelo4, newx = x_train)


predicciones_test_modelo4 <- predict(modelo4, newx = x_test)

# MSE de test
test_mse_lasso_modelo4 <- mean((predicciones_test_modelo4 - y_test)^2)

# MSE de entrenamiento
training_mse_lasso_modelo4 <- mean((predicciones_train_modelo4 - y_train)^2)

print(test_mse_lasso_modelo4 )
```

```
## [1] 0.1447764
```

```
print(training_mse_lasso_modelo4 )
```

```
## [1] 0.1634659
```

Analisis Podemos ver que realmente no cambio nada de los valores que se detectaron al inicio , asi que vamos ahora a mostrarlos en un grafico que demuestre la diferencia entre los modelos.

```
# Calcular residuos
residuos_train <- y_train - predicciones_train_modelo4
residuos_test <- y_test - predicciones_test_modelo4
```

```r
print(test_mse_ridge_modelo3)
```

```
## [1] 0.1436333
```

```r
print(training_mse_ridge_modelo3)
```

```
## [1] 0.1631888
```

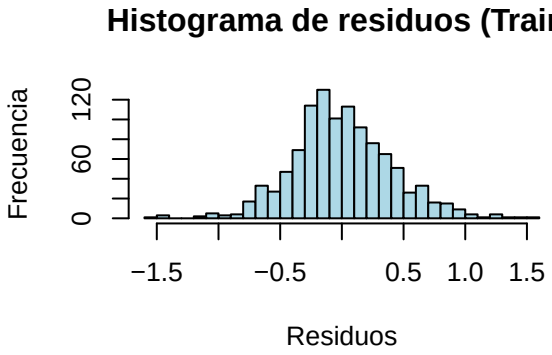
```r
summary(residuos_train)
```

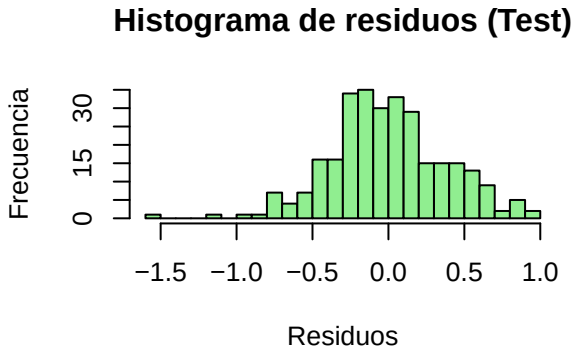
```
##        s0
##  Min.   :-1.5543
##  1st Qu.:-0.2509
##  Median :-0.0232
##  Mean   : 0.0000
##  3rd Qu.: 0.2475
##  Max.   : 1.5082
```

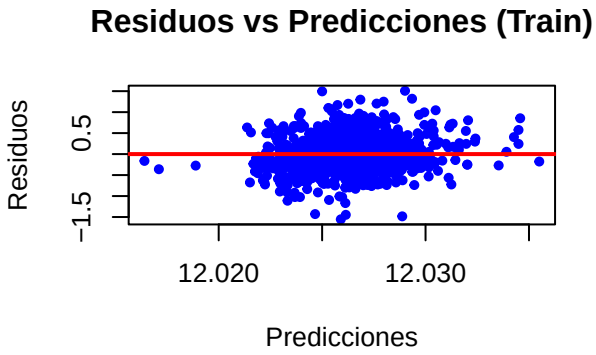
```r
summary(residuos_test)
```

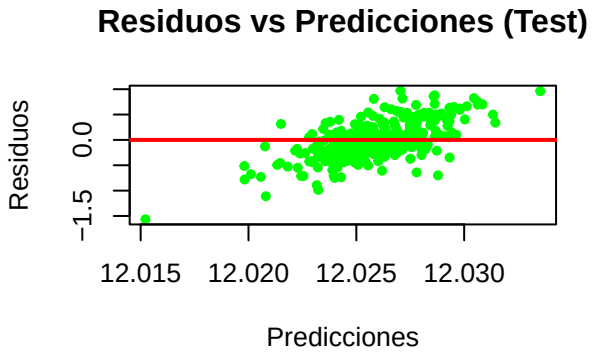
```
##        s0
##  Min.   :-1.565992
##  1st Qu.:-0.252871
##  Median :-0.024729
##  Mean   :-0.006394
##  3rd Qu.: 0.233368
##  Max.   : 0.968296
```

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


hist(residuos_train, main="Histograma de residuos (Train)",
     col="lightblue", breaks=30, xlab="Residuos", ylab="Frecuencia")


hist(residuos_test, main="Histograma de residuos (Test)",
     col="lightgreen", breaks=30, xlab="Residuos", ylab="Frecuencia")



plot(predicciones_train_modelo3, residuos_train,
     main="Residuos vs Predicciones (Train)",
     xlab="Predicciones", ylab="Residuos", pch=20, col="blue")
abline(h=0, col="red", lwd=2)


plot(predicciones_test_modelo3, residuos_test,
     main="Residuos vs Predicciones (Test)",
     xlab="Predicciones", ylab="Residuos", pch=20, col="green")
abline(h=0, col="red", lwd=2)
```










```
# Resetear layout
par(mfrow=c(1,1))
```

Analisis Viendo los residuos vemos el mismo patron al que utilizar Ridge lo que indica que ambos modelos son casi iguales en eficiencia. De hecho vemos que la dispersion y el histograma son exactos por lo que no es necesario llevar a cabo otros modelos.

11. Utilice cada modelo con el conjunto de prueba y determine la eficiencia del algoritmo para predecir el precio de las casas. ¿Qué tan bien lo hizo?¿Qué medidas usó para determinar la calidad de la predicción?

Como se vio anteriormente se realizo una prediccion con cada uno de los modelos tuvo un error medio cuadrado igual. Esta fue la metrica que se utilizo debido a que el MSE permite evaluar la precisión del modelo midiendo la diferencia cuadrática media entre los valores predichos y los valores reales.

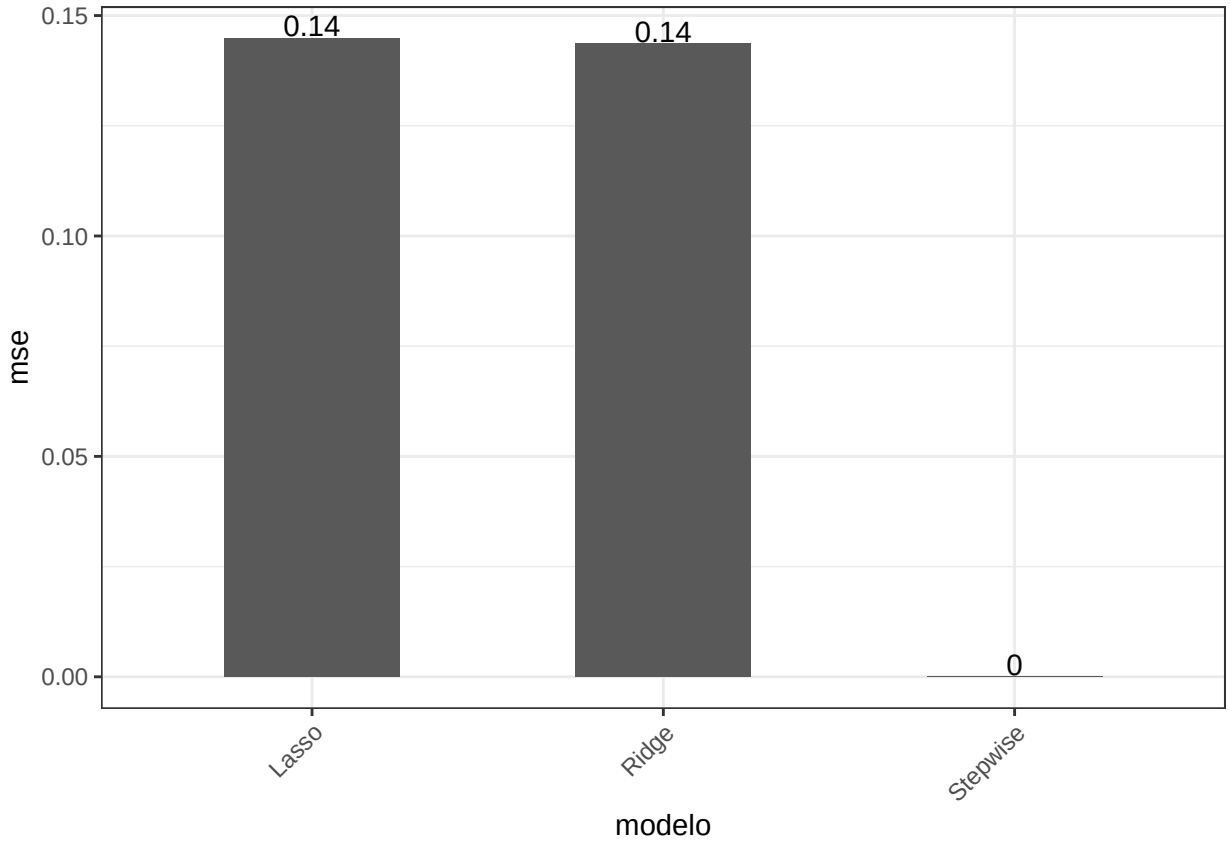
El MSE es una métrica útil porque penaliza los errores grandes de manera más severa que los pequeños, lo que ayuda a identificar modelos con predicciones más precisas. Un MSE más bajo indica un mejor ajuste del modelo a los datos de entrenamiento y prueba.

En retroalimentacion se puede ver que el que mejor desepeño tuvo de los 2 fue el mismo a pesar de utilizar 2 enfoques diferenetes.

12. Discuta sobre la efectividad de los modelos. ¿Cuál lo hizo mejor? ¿Cuál es el mejor modelo para predecir el precio de las casas? Haga los gráficos que crea que le pueden ayudar en la discusión.

Para poder observar como fue este comportamiento vamos a ver utilizando una grafica de barras para todos los modelos

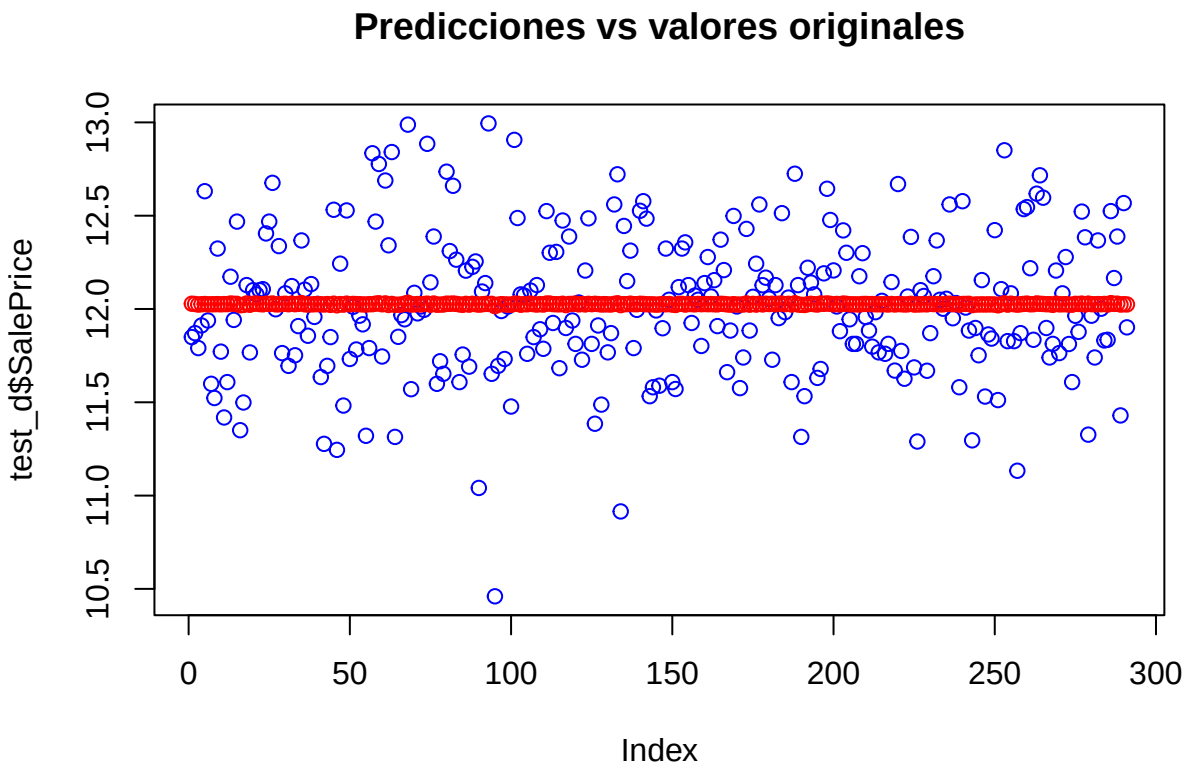
```
df_comparacion <- data.frame(
                    modelo = c("Stepwise", "Ridge", "Lasso"),
                    mse    = c(test_mse_stepwise, test_mse_ridge_modelo3,
                               test_mse_lasso_modelo4)
)

ggplot(data = df_comparacion, aes(x = modelo, y = mse)) +
  geom_col(width = 0.5) +
  geom_text(aes(label = round(mse, 2)), vjust = -0.1) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Analisis Solo que utilizaremos Ridge en este caso dado los 3 modelos. Porque no utilizamos stepwise, esto se debe a que estamos hablando cuando tenemos el mse es muy bajo osea tendiendo a 0 tenemos un sobreajuste, esto ya lo habiamos discutido y de hecho con los residuos , y como queremos seguir conservando los resultados mas posibles entonces Ridge va a ser el que vamos a elegir en ves de Lasso, ya que Lasso elimino una gran parte de las variables solo dejando 2 .

En la siguiente gráfica vemos como quedarían las predicciones de acuerdo con los valores originales.

```r
plot(test_d$SalePrice,col="blue", main="Predicciones vs valores originales")
points(predicciones_test_modelo3, col="red")
legend(30,45,legend=c("original", "prediccion"),col=c("blue", "red"),pch=1, cex=0.8)
```



Analisis Como podemos ver la linea es la prediccion del modelo. Lo cual indica una tendencia lineal del mismo y la mayoria de los datos reales rodean al modelo lo mas cercano posible. Como mejoras serian hacer una generalizacion de los mismos ya que aqui en este no hay tanta generalizacion como se vio en el error mse.

Conclusiones

Como conclusiones podemos ver que el modelo multilineal elegiremos el dado por Ridge debido a que guarda la mayor parte de las variables y no reduce la dimensionalidad. En cambio Lasso elimina casi todas , y Stepwise es demasiado ajustado por lo que no permite generalizacion.

Conclusiones del Análisis de Clustering

El análisis de clustering nos permitió segmentar las casas en tres grupos distintos, proporcionando información clave sobre cómo las características físicas y estructurales afectan el precio de las viviendas.

1. Interpretación General del Clustering

El algoritmo de K-Means clasificó las casas en tres clusters basados en calidad de construcción, tamaño y precio:

Cluster 1 (Rojo - Casas de Bajo Precio)

- Baja calidad de construcción (OverallQual bajo).
- Tamaño reducido en área habitable (GrLivArea pequeño) y sótano (TotalBsmtSF pequeño).
- Garaje pequeño o inexistente (GarageCars cercano a 0 o 1).
- Ubicadas en vecindarios de menor costo.
- Presentan los precios más bajos, lo que indica que su valor es más estable y menos dependiente de otros factores.

Cluster 2 (Azul - Casas de Precio Medio)

- Calidad media a alta (OverallQual entre 5 y 7).
- Tamaño intermedio con áreas habitables y sótanos de tamaño moderado.
- Garaje con espacio para 1 o 2 autos.
- Representan la mayoría del dataset, lo que sugiere que es el segmento más común en el mercado.
- Precio en el rango medio con variabilidad en función de la ubicación y otros factores.

Cluster 3 (Verde - Casas de Lujo)

- Alta calidad de construcción (OverallQual > 7).
- Casas grandes con un área habitable amplia (GrLivArea grande).
- Garajes espaciosos (2-3 autos).
- Se encuentran en vecindarios de mayor prestigio.
- Tienen los precios más altos y muestran una dispersión significativa, lo que sugiere que otros factores como la ubicación y los acabados afectan su precio.

2. Hallazgos Clave

- El clustering confirma que el precio de una casa está altamente influenciado por su tamaño y calidad de construcción.
- Los tres grupos presentan diferencias claras, lo que confirma que segmentar los datos ayuda a comprender mejor el comportamiento de los precios de las casas.
- El Cluster 1 (rojo) tiene casas alejadas del grupo principal, lo que sugiere posibles valores atípicos o características únicas.
- El Cluster 3 (verde) muestra una mayor dispersión, lo que indica que en casas de alto costo, factores como ubicación y acabados juegan un rol crucial en la determinación del precio.