

Tipología de datos PRA2 - Limpieza y análisis de datos

Autor: Begoña Martínez Arribas /Silvia Martín Albarrán

Mayo 2020

Contents

Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder	1
Integración y selección de los datos de interés a analizar	6
Limpieza de datos	12
Missing Values : ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?	12
Outliers (Extreme scores) - Identificación y tratamiento de valores extremos.	19
Analisis de datos	29
Selección de los grupos de datos que se quieren analizar/comparar (planificación los análisis a aplicar).	29
Comprobación de la normalidad y homogeneidad de la varianza.	56
Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc.	
Aplicar al menos tres métodos de análisis diferentes.	68
Representación de los resultados a partir de tablas y gráficas.	105
Conclusiones :	105
Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?	105
Contribuciones :	108
Entregable:	108
Referencias bibliográficas :	108

Enunciado PRACTICA Nº 2

El objetivo de esta actividad será el tratamiento de un dataset, que puede ser el creado en la práctica 1 o bien cualquier dataset libre disponible en Kaggle (<https://www.kaggle.com>).

Algunos ejemplos de dataset con los que podéis trabajar son: + [Red Wine Quality (<https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009>) + [Titanic: Machine Learning from Disaster (<https://www.kaggle.com/c/titanic>)

El último ejemplo corresponde a una competición activa de Kaggle de manera que, opcionalmente, podéis aprovechar el trabajo realizado durante la práctica para entrar en esta competición.

Siguiendo las principales etapas de un proyecto analítico, las diferentes tareas a realizar (y justificar) son las siguientes:

Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder

Se ha seleccionado el juego de datos seleccionado en la página web kaggle denominado: + [house-prices-advanced-regression-techniques] en el que se puede observar el precio de las casas o propiedades vendidas en Ames , ciudad de Iowa y una serie de características propias de las viviendas : año de construcción o remodelación, materiales, el tipo de vivienda , area de la parcela, el barrio donde está ubicado y el tipo de zona , nº de habitaciones en el sotano o en plantas , si tiene o no piscina etc.

Consideramos importante este dataset para poder responder como evoluciona el precio de venta en función de las características de la casa y para poder aplicar técnicas de regresión para poder predecir el precio que tendrá una casa similar en el futuro.

El juego de datos de Kaggle consta de dos ficheros csv : train y test. Para la práctica vamos a utilizar el de entrenamiento que tiene la variable objetivo del precio de casa , con el fin de poder hacer el análisis inicial y la limpieza y depuración . Alguna preguntas inicialmente planteadas que podremos contestar a lo largo de esta práctica serán: - ¿Poseer aire acondicionado puede influir en el precio de la vivienda? - ¿Puede el vecindario influir en el precio? - ¿Dependiendo de la zona (residencial, industrial, etc) como varía el precio? - ¿El tipo de vivienda puede producir variaciones en el precio final? - ¿Qué otras características pueden influir en el precio (nº habitaciones, metros cuadrados, año de construcción, etc)?

Este dataset consta de numerosas columnas (81) que pasamos a detallar:

- MSSubClass: Identifica el tipo de casa de la venta - Variable categórica . Puede tomar los siguientes valores.

20 1-STORY 1946 & NEWER ALL STYLES
30 1-STORY 1945 & OLDER
40 1-STORY W/FINISHED ATTIC ALL AGES
45 1-1/2 STORY - UNFINISHED ALL AGES
50 1-1/2 STORY FINISHED ALL AGES
60 2-STORY 1946 & NEWER
70 2-STORY 1945 & OLDER
75 2-1/2 STORY ALL AGES
80 SPLIT OR MULTI-LEVEL
85 SPLIT FOYER
90 DUPLEX - ALL STYLES AND AGES

120 1-STORY PUD (Planned Unit Development) - 1946 & NEWER 150 1-1/2 STORY PUD - ALL AGES 160 2-STORY PUD - 1946 & NEWER 180 PUD - MULTILEVEL - INCL SPLIT LEV/FOYER 190 2 FAMILY CONVERSION - ALL STYLES AND AGES

- MSZoning: Zona donde se ubica la vivienda . Categórica. Puede tomar los siguientes valores A Agriculture
C Commercial

FV Floating Village Residential I Industrial RH Residential High Density RL Residential Low Density RP Residential Low Density Park RM Residential Medium Density

- LotFrontage: Pies (medida de longitud) lineales de la calle conectados con la casa.
- LotArea: Tamaño de la parcela en metros cuadrados - Cuantitativa
- Street: Tipo de acceso a la calle - Categorica

Grvl Gravel
Pave Paved

- Alley: Tipo de callejón de acceso a la propiedad (*B) :
Grvl Gravel Pave Paved NA No alley access
- LotShape: Tipo de estructura : (*B)
Reg Regular IR1 Slightly irregular IR2 Moderately Irregular IR3 Irregular
- LandContour: Flatness of the property (*B)
Lvl Near Flat/Level Bnk Banked - Quick and significant rise from street grade to building HLS Hillside
- Significant slope from side to side Low Depression
- Utilities: Tipo de servicios disponibles/accesibles
AllPub All public Utilities (E,G,W,& S)
NoSewr Electricity, Gas, and Water (Septic Tank) NoSeWa Electricity and Gas Only ELO Electricity only
- LotConfig: Configuración de la propiedad.
Inside Inside lot Corner Corner lot CulDSac Cul-de-sac FR2 Frontage on 2 sides of property FR3
Frontage on 3 sides of property
- LandSlope: Slope of property (*B)
Gtl Gentle slope Mod Moderate Slope
Sev Severe Slope
- Neighborhood: Vecindario donde se ubica la propiedad dentro de Ames (Iowa)
Blmngtn Bloomington Heights Blueste Bluestem BrDale Briardale BrkSide Brookside ClearCr Clear Creek CollgCr College Creek Crawfor Crawford Edwards Edwards Gilbert Gilbert IDOTRR Iowa DOT and Rail Road MeadowV Meadow Village Mitchel Mitchell Names North Ames NoRidge Northridge NPkVill Northpark Villa NridgHt Northridge Heights NWAmes Northwest Ames OldTown Old Town SWISU South & West of Iowa State University Sawyer Sawyer SawyerW Sawyer West Somerst Somerset StoneBr Stone Brook Timberland Veenker Veenker
- Condition1: Proximidad a carreteras/ Normal / o Ferrocarril
Artery Adjacent to arterial street Feedr Adjacent to feeder street
Norm Normal
RRNn Within 200' of North-South Railroad RRAn Adjacent to North-South Railroad PosN Near positive off-site feature—park, greenbelt, etc. PosA Adjacent to postive off-site feature RRNe Within 200' of East-West Railroad RRAe Adjacent to East-West Railroad
- Condition2: Proximidad a carreteras/ Normal / o Ferrocarril (Si se da más de condicion) (*)
Artery Adjacent to arterial street Feedr Adjacent to feeder street
Norm Normal
RRNn Within 200' of North-South Railroad RRAn Adjacent to North-South Railroad PosN Near positive off-site feature—park, greenbelt, etc. PosA Adjacent to postive off-site feature RRNe Within 200' of East-West Railroad RRAe Adjacent to East-West Railroad
- BldgType: Tipo de Alojamiento
1Fam Single-family Detached
2FmCon Two-family Conversion; originally built as one-family dwelling Duplx Duplex TwnhsE Townhouse End Unit TwnhsI Townhouse Inside Unit
- HouseStyle: Estilo de vivienda (*B)

1Story One story 1.5Fin One and one-half story: 2nd level finished 1.5Unf One and one-half story: 2nd level unfinished 2Story Two story 2.5Fin Two and one-half story: 2nd level finished 2.5Unf Two and one-half story: 2nd level unfinished SFoyer Split Foyer SLvl Split Level

- OverallQual: Calidad de los materiales de la casa

10 Very Excellent 9 Excellent 8 Very Good 7 Good 6 Above Average 5 Average 4 Below Average 3 Fair 2 Poor 1 Very Poor

- OverallCond: Evaluación del estado general de la casa (*B)

10 Very Excellent 9 Excellent 8 Very Good 7 Good 6 Above Average

5 Average 4 Below Average

3 Fair 2 Poor 1 Very Poor

- YearBuilt: Año de construcción de la casa

- YearRemodAdd: Año de remodelación de la casa

- RoofStyle: Tipo de tejado /techo (*B)

Flat Flat Gable Gable Gambrel Gabrel (Barn) Hip Hip Mansard Mansard Shed Shed

- RoofMatl: Material del tejado (*B)

ClyTile Clay or Tile CompShg Standard (Composite) Shingle Membran Membrane Metal Metal Roll Roll Tar&Grv Gravel & Tar WdShake Wood Shakes WdShngl Wood Shingles

- Exterior1st: Cubierta exterior (*B)

AsbShng Asbestos Shingles AsphShn Asphalt Shingles BrkComm Brick Common BrkFace Brick Face CBlock Cinder Block CemntBd Cement Board HdBoard Hard Board ImStucc Imitation Stucco MetalSd Metal Siding Other Other Plywood Plywood PreCast PreCast Stone Stone Stucco Stucco VinylSd Vinyl Siding Wd Sdng Wood Siding WdShing Wood Shingles

- Exterior2nd: Segundo material de la cubierta exterior (si hay más de un material) (*B)

AsbShng Asbestos Shingles AsphShn Asphalt Shingles BrkComm Brick Common BrkFace Brick Face CBlock Cinder Block CemntBd Cement Board HdBoard Hard Board ImStucc Imitation Stucco MetalSd Metal Siding Other Other Plywood Plywood PreCast PreCast Stone Stone Stucco Stucco VinylSd Vinyl Siding Wd Sdng Wood Siding WdShing Wood Shingles

- MasVnrType: Tipo de revestimiento de manposteria (*B)

BrkCmn Brick Common BrkFace Brick Face CBlock Cinder Block None None Stone Stone

- MasVnrArea: Área de revestimiento de manposteria. Tiene relación con la anterior. (*B)

- ExterQual: Calidad de los materiales externos (*B)

Ex Excellent Gd Good TA Average/Typical Fa Fair Po Poor

- ExterCond: Condiciones de los materiales externos (*B)

Ex Excellent Gd Good TA Average/Typical Fa Fair Po Poor

- Foundation: Tipo de cimientos (*B)

BrkTil Brick & Tile CBlock Cinder Block PConc Poured Contrete Slab Slab Stone Stone Wood Wood

- BsmtQual: Altura del sotano

Ex Excellent (100+ inches) Gd Good (90-99 inches) TA Typical (80-89 inches) Fa Fair (70-79 inches) Po Poor (<70 inches) NA No Basement

- BsmtCond: Evalua el estado general del sotano (*B)

Ex Excellent Gd Good TA Typical - slight dampness allowed Fa Fair - dampness or some cracking or settling Po Poor - Severe cracking, settling, or wetness NA No Basement

- BsmtExposure: Exposición del sotano al jardín o a un paseo. (*B)

Gd Good Exposure Av Average Exposure (split levels or foyers typically score average or above)
Mn Minimum Exposure No No Exposure NA No Basement

- BsmtFinType1: Estado del Acabado del sotano (*B)

GLQ Good Living Quarters ALQ Average Living Quarters BLQ Below Average Living Quarters
Rec Average Rec Room LwQ Low Quality Unf Unfinished NA No Basement

- BsmtFinSF1: El número de metros cuadrados del sotano

- BsmtFinType2: Clasificación del acabado del sotano (si hay más de uno) (*B)

GLQ Good Living Quarters ALQ Average Living Quarters BLQ Below Average Living Quarters
Rec Average Rec Room LwQ Low Quality Unf Unfinished NA No Basement

- BsmtFinSF2: El número de metros cuadrados del sotano2 (*B)

- BsmtUnfSF: Unfinished square feet of basement area (*B)

- TotalBsmtSF: Área total del sotano

- Heating: Tipo de calefacción

Floor Floor Furnace GasA Gas forced warm air furnace GasW Gas hot water or steam heat Grav Gravity furnace OthW Hot water or steam heat other than gas Wall Wall furnace

- HeatingQC: Calidad de la calefacción /condiciones.

Ex Excellent Gd Good TA Average/Typical Fa Fair Po Poor

- CentralAir: Aire acondicionado

N No Y Yes

- Electrical: Sistema Eléctrico. (*B)

SBrkr Standard Circuit Breakers & Romex FuseA Fuse Box over 60 AMP and all Romex wiring (Average) FuseF 60 AMP Fuse Box and mostly Romex wiring (Fair) FuseP 60 AMP Fuse Box and mostly knob & tube wiring (poor) Mix Mixed

- 1stFlrSF: Metros cuadrados de la primera planta (*B)

- 2ndFlrSF: Metros cuadrados de las segunda planta (*B)

- LowQualFinSF: Área terminada de baja calidad (metros cuadrados) (*B)

- GrLivArea: Área habitable en metros cuadrados

- BsmtFullBath: N° de baños completos en el sotano (*B)

- BsmtHalfBath: N° de aseos en el sotano. (*B)

- FullBath: N° de baños en la vivienda (plantas)

- HalfBath: N° de aseos en la vivienda (plantas)

- BedroomAbvGrd: N° de habitaciones en la planta (no incluye los del sotano)

- Kitchen: N° de cocinas en la vivienda

- KitchenQual: Calidad de la cocina
Ex Excellent Gd Good TA Typical/Average Fa Fair Po Poor
- TotRmsAbvGrd: N° total de habitaciones de la vivienda (sin contar baños)
- Functional: Uso de la vivienda (por si tienen descuentos) (*B)
Typ Typical Functionality Min1 Minor Deductions 1 Min2 Minor Deductions 2 Mod Moderate Deductions Maj1 Major Deductions 1 Maj2 Major Deductions 2 Sev Severely Damaged Sal Salvage only
- Fireplaces: Numero de chimeneas
- FireplaceQu: Calidad de la chimeneas
Ex Excellent - Exceptional Masonry Fireplace Gd Good - Masonry Fireplace in main level TA Average - Prefabricated Fireplace in main living area or Masonry Fireplace in basement Fa Fair - Prefabricated Fireplace in basement Po Poor - Ben Franklin Stove NA No Fireplace
- GarageType: Ubicación del garage
2Types More than one type of garage Attchd Attached to home Basment Basement Garage BuiltIn Built-In (Garage part of house - typically has room above garage) CarPort Car Port Detchd Detached from home NA No Garage
- GarageFinish: Acabado del garage
Fin Finished RFn Rough Finished Unf Unfinished NA No Garage
- GarageYrBlt: Año de construcción del garage
- GarageCars: Capacidad del n° de coches (plazas)
- GarageArea: Tamaño del garaje (metros cuadrados)
- GarageQual: Calidad del garaje
Ex Excellent Gd Good TA Typical/Average Fa Fair Po Poor NA No Garage
- GarageCond: Condición del garaje
Ex Excellent Gd Good TA Typical/Average Fa Fair Po Poor NA No Garage
- PavedDrive: Camino de entrada (*B)
Y Paved P Partial Pavement N Dirt/Gravel
- WoodDeckSF: Area de cubierta de madera (metros cuadrados) (*B)
- OpenPorchSF: Area del porche abierto (metros cuadrados)
- EnclosedPorch: Area del porche cerrado (metros cuadrados) (*B)
- 3SsnPorch: Area del porche 3 estaciones (metros cuadrados)(*B)
- ScreenPorch: Area de la pantalla del porches (metros cuadrados) (*B)
- PoolArea: Tamaño de la piscina (en metros cuadrados) (*B)
- PoolQC: Calidad de la piscina (*B)
Ex Excelente Gd Buenda TA Media/Tipica Fa Justa/pequeña NA Sin piscina
- Fence: Calidad de la valla (en metros cuadrados)
GdPrv Good Privacy MnPrv Minimum Privacy GdWo Good Wood MnWw Minimum Wood/Wire NA No Fence

- MiscFeature: Otras características adicionales.

Elev Elevator Gar2 2nd Garage (if not described in garage section) Othr Other Shed Shed (over 100 SF) TenC Tennis Court NA None

- MiscVal: Importe (\$) de las características adicionales.
- MoSold: Month Sold (MM) (Mes de la venta)
- YrSold: Year Sold (YYYY) (Año de la venta)
- SaleType: Condiciones de la venta (*B)

WD Warranty Deed - Conventional CWD Warranty Deed - Cash VWD Warranty Deed - VA Loan New Home just constructed and sold COD Court Officer Deed/Estate Con Contract 15% Down payment regular terms ConLw Contract Low Down payment and low interest ConLI Contract Low Interest ConLD Contract Low Down Oth Other

- SaleCondition: Condition of sale (*B)

Normal Normal Sale Abnrmal Abnormal Sale - trade, foreclosure, short sale AdjLand Adjoining Land Purchase Allocat Allocation - two linked properties with separate deeds, typically condo with a garage unit

Family Sale between family members Partial Home was not completed when last assessed (associated with New Homes)

Integración y selección de los datos de interés a analizar

A continuación se lleva a cabo la carga el dataset que vamos a utilizar. Se muestran algunos datos y la estructura del dataset:

Para cargar el dataset lo hacemos desde el directorio del entorno por defecto de cada ordenador, por lo que se ha usado un setwd para indicar el path correspondiente

```
#houses<-read.csv("train_house.csv", stringsAsFactors = FALSE)

#Mostramos las primeras filas del juego de datos.
head(houses[,1:7],3)
```

```
##   Id MSSubClass MSZoning LotFrontage LotArea Street Alley
## 1  1          60      RL        65    8450  Pave <NA>
## 2  2          20      RL        80    9600  Pave <NA>
## 3  3          60      RL        68   11250  Pave <NA>
```

Guardamos un backup del dataset por si fuera necesario recuperarlo.

```
houses_bk <- houses
```

Con la siguiente instrucción vemos el nº observaciones (filas) y atributos (columnas) del juego de datos. (Dimension)

```
# Se muestran el nº de columnas y observaciones que tiene el juego de datos
dim.datos <- dim(houses)
dim.datos
```

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

```
# Guardamos el nº de filas y de variables para su utilización posterior
n.ind = dim.datos[1]
n.var = dim.datos[2]
```

Vemos el tipo de variables y valores de los datos leídos del fichero

```
str(houses)
```

```
## '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 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 ...

```

Se extraen los principales descriptivos de las variables con la función summary

```
summary(houses)
```

```

##           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      NA's   :259        Mean   : 70.05
## 3rd Qu.:1095.2     3rd Qu.: 70.0      NA's   :259        3rd Qu.: 80.00
## Max.   :1460.0     Max.   :190.0      NA's   :259        Max.   :313.00
## 
##           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
##   Length:1460      Length:1460      Length:1460
##   Class :character Class :character Class :character
##   Mode  :character Mode  :character Mode  :character
##
## 
## 
## 
##   LandSlope      Neighborhood      Condition1
##   Length:1460      Length:1460      Length:1460
##   Class :character Class :character Class :character
##   Mode  :character Mode  :character Mode  :character
##
## 
## 
## 
##   Condition2      BldgType      HouseStyle      OverallQual
##   Length:1460      Length:1460      Length:1460      Min.    : 1.000
##   Class :character Class :character Class :character 1st Qu.: 5.000
##   Mode  :character Mode  :character Mode  :character Median  : 6.000
##                                         Mean    : 6.099
##                                         3rd Qu.: 7.000
##                                         Max.   :10.000
##
## 
##   OverallCond      YearBuilt      YearRemodAdd     RoofStyle
##   Min.    :1.000    Min.    :1872    Min.    :1950    Length:1460
##   1st Qu.:5.000    1st Qu.:1954    1st Qu.:1967    Class :character
##   Median  :5.000    Median :1973    Median :1994    Mode  :character
##   Mean    :5.575    Mean    :1971    Mean    :1985
##   3rd Qu.:6.000    3rd Qu.:2000    3rd Qu.:2004
##   Max.    :9.000    Max.    :2010    Max.    :2010
##
##   RoofMatl      Exterior1st      Exterior2nd
##   Length:1460      Length:1460      Length:1460
##   Class :character Class :character Class :character
##   Mode  :character Mode  :character Mode  :character
##
## 
## 
## 
##   MasVnrType      MasVnrArea      ExterQual      ExterCond
##   Length:1460      Min.    : 0.0    Length:1460      Length:1460
##   Class :character 1st Qu.: 0.0    Class :character Class :character
##   Mode  :character Median : 0.0    Mode  :character Mode  :character
##                                         Mean    : 103.7
##                                         3rd Qu.: 166.0
##                                         Max.   :1600.0
##                                         NA's   :8
##   Foundation      BsmtQual      BsmtCond
##   Length:1460      Length:1460      Length:1460
##   Class :character Class :character Class :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
##   3rd Qu.:1.000
##   Max.   :3.000
##
##   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
## Length:1460       Length:1460    Length:1460
## Class :character  Class :character Class :character
## Mode  :character  Mode  :character Mode  :character
##
## MiscVal          MoSold      YrSold      SaleType
## Min.   : 0.00  Min.   : 1.000  Min.   :2006  Length:1460
## 1st Qu.: 0.00  1st Qu.: 5.000  1st Qu.:2007  Class :character
## Median : 0.00  Median : 6.000  Median :2008  Mode  :character
## Mean   : 43.49  Mean   : 6.322  Mean   :2008
## 3rd Qu.: 0.00  3rd Qu.: 8.000  3rd Qu.:2009
## Max.   :15500.00 Max.   :12.000  Max.   :2010
##
## SaleCondition    SalePrice
## Length:1460       Min.   :34900
## Class :character  1st Qu.:129975
## Mode  :character  Median :163000
##                           Mean   :180921
##                           3rd Qu.:214000
##                           Max.   :755000
##

```

Como se ha indicado previamente el dataset tratado es muy amplio en cuanto al número de atributos. Para el análisis a llevar acabo acorde a las preguntas planetadas en el apartado previo, seleccionaremos aquellos atributos que a lo largo del estudio realizado en la practica contribuyan a encontrar las respuestas. Eliminaremos aquellos atributos que no aporten información relevante (no muestren correlación) o puedan incluir ruido (variables con más de 90% de valores vacíos).

Limpieza de datos

Missing Values : ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Los valores missing o desconocidos pueden deberse a errores en la introducción de datos, o bien ser valores lógicos porque no aplica poner cualquier otra categoría o en el caso del 0 , porque se corresponde con un valor valido de la medida de dicha variable. Una vez revisado el dataset se ha visto que los posibles valores vacíos vienen cumplimentados como NA o 0, excluyendo otros valores como '?'. Para detectar los nulos ,se puede utilizar :

```
colSums(is.na(houses))
```

##	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	X1stFlrSF	X2ndFlrSF
##	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	X3SsnPorch
##	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

```

Se extrae el % de valores perdidos por cada atributo:

```

misscols<-sapply(houses, function(x)all(any(is.na(x))))
colswithmiss <-names(misscols[misscols>0]);
x = data.frame(houses[,colswithmiss])
PorcentajeDatosMissing<-apply(x, 2, function(col)sum(is.na(col))/length(col))
missing.char<-as.data.table(PorcentajeDatosMissing,keep.rownames = "Variable")
print(missing.char)

```

	Variable	PorcentajeDatosMissing
## 1:	LotFrontage	0.1773972603
## 2:	Alley	0.9376712329
## 3:	MasVnrType	0.0054794521
## 4:	MasVnrArea	0.0054794521
## 5:	BsmtQual	0.0253424658
## 6:	BsmtCond	0.0253424658
## 7:	BsmtExposure	0.0260273973
## 8:	BsmtFinType1	0.0253424658
## 9:	BsmtFinType2	0.0260273973
## 10:	Electrical	0.0006849315
## 11:	FireplaceQu	0.4726027397
## 12:	GarageType	0.0554794521
## 13:	GarageYrBlt	0.0554794521
## 14:	GarageFinish	0.0554794521
## 15:	GarageQual	0.0554794521
## 16:	GarageCond	0.0554794521
## 17:	PoolQC	0.9952054795
## 18:	Fence	0.8075342466
## 19:	MiscFeature	0.9630136986

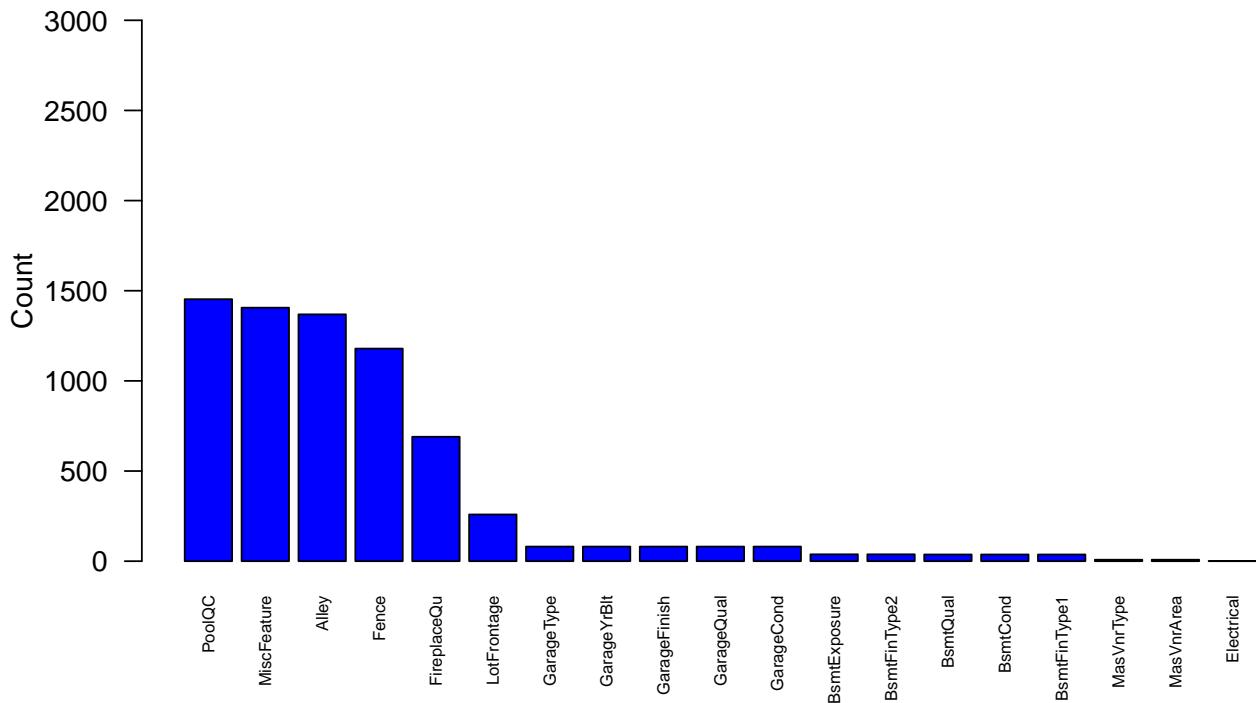
Visualizamos como se distribuyen los valores vacíos por atributo

```

options(repr.plot.width=6, repr.plot.height=5)
houses_mis = function(x){sum(is.na(x))}
mis <- sort(apply(houses,2,houses_mis),decreasing=T);
barplot(mis[mis!=0],
        las=2,
        cex.names=0.6,
        ylab="Count",
        ylim=c(0,3000),
        horiz=F,
        col="blue",
        main=paste(toString(sum(mis!=0)), "Variables con datos faltantes"))

```

19 Variables con datos faltantes



Para aquellas variables que tienen más de un 80% de valores vacíos se incluye la etiqueta "Not apply" en estos. Esas variables son: Alley, PoolQC, MiscFeature y Fence. Además para el resto de variables se gestionan los valores perdidos que están vacíos bien con la moda o mediana según se trate de un atributo categórico o numérico. El valor 0 en las variables numéricas que admiten este valor se ha considerado un valor legítimo. En los casos que se trata del area de una variable categórica que no aplica este valor ciertamente debe ser 0. A continuación se incluye esta implementación de los valores perdidos.

```

for (i in 1:nrow(houses))
{
  # Para Alley, PoolQC, MiscFeature y Fence con más del 80% de valores a NA se decide
  # incluir la etiqueta 'Not apply'
  houses$Alley[is.na(houses$Alley)] <- "Not Apply"
  houses$PoolQC[is.na(houses$PoolQC)]<- "Not apply"
  houses$MiscFeature[is.na(houses$MiscFeature)]<- "Not apply"
  houses$Fence[is.na(houses$Fence)]<- "Not apply"

  #Para la variable MasVnrType se decide completar los vacíos con "Not apply" ya que
  #todos tienen el siguiente atributo relacionado MasVnrArea vacío se distingue
  #del valor None para saber los que se han limpiado con este proceso
  houses$MasVnrType[is.na(houses$MasVnrType)]<- "Not apply"

  #Para la variable BsmtQual los vacíos se ponen como "Not apply" ya que se trata de
  #una variable de tipo categórico
  houses$BsmtQual[is.na(houses$BsmtQual)]<- "Not apply"

  #Para la variable BsmtCond los vacíos se ponen como "Not apply" ya que se trata de
  #una variable de tipo categórico
  houses$BsmtCond[is.na(houses$BsmtCond)]<- "Not apply"

  #Para la variable BsmtExposure los vacíos se ponen como "Not apply" ya que se trata
}

```

```

#de una variable de tipo categórico
houses$BsmtExposure[is.na(houses$BsmtExposure)]<- "Not apply"

#Para la variable BsmtFinType1 los vacíos se ponen como "Not apply" ya que se trata
#de una variable de tipo categórico
houses$BsmtFinType1[is.na(houses$BsmtFinType1)]<- "Not apply"

#Para la variable FireplaceQu los vacíos se ponen como "Not apply" ya que se trata
#de una variable de tipo categórico y puede darse este valor
houses$FireplaceQu[is.na(houses$FireplaceQu)]<- "Not apply"

#Para la variable GarageType los vacíos se ponen como "Not apply" ya que se trata de
#una variable de tipo categórico y puede darse este valor, es decir que no tenga garaje
houses$GarageType[is.na(houses$GarageType)]<- "Not apply"

#Para la variable GarageQual los vacíos se ponen como "Not apply" ya que se trata de una
#variable de tipo categórico y al no tener garaje no aplica
houses$GarageQual[is.na(houses$GarageQual)]<- "Not apply"

#Para la variable GarageCond los vacíos se ponen como "Not apply" ya que se trata de una
#variable de tipo categórico e igualmente no aplicaría si no #tiene garaje la vivienda
houses$GarageCond[is.na(houses$GarageCond)]<- "Not apply"

#Para la variable BsmtFinType2 los vacíos se ponen como "Not apply" ya que se trata de
#una variable de tipo categórico
houses$BsmtFinType2[is.na(houses$BsmtFinType2)]<- "Not apply"

#Para la variable BsmtFinSF1 los vacíos se ponen a 0 ya que se trata de una variable
#de tipo numérico
houses$BsmtFinSF1[is.na(houses$BsmtFinSF1)]<- 0

#Para la variable BsmtFinSF2 los vacíos se ponen a 0 ya que se trata de una variable
#de tipo numérico
houses$BsmtFinSF2[is.na(houses$BsmtFinSF2)]<- 0

#Para la variable TotalBsmtSF los vacíos se ponen a 0 ya que se trata de una variable
#de tipo numérico
houses$TotalBsmtSF[is.na(houses$TotalBsmtSF)]<- 0

#Para la variable BsmtUnfSF los vacíos se ponen a 0 ya que se trata de una variable
#de tipo numérico
houses$BsmtUnfSF[is.na(houses$BsmtUnfSF)]<- 0

#Para la variable BsmtFullBath los vacíos se ponen a 0 ya que se trata de una variable
#de tipo numérico y se entiende que no tienen baños
houses$BsmtFullBath[is.na(houses$BsmtFullBath)]<- 0

#Para la variable BsmtHalfBath los vacíos se ponen a 0 ya que se trata de una variable
#de tipo numérico, se entiende que no tienen aseos
houses$BsmtHalfBath[is.na(houses$BsmtHalfBath)]<- 0

#Para la variable GarageYrBlt se decide completar como 0 ya que corresponde a los
#GarageType con NA, que se ha supuesto no tienen garaje

```

```

houses$GarageYrBlt[is.na(houses$GarageYrBlt)]<- 0

#Para la variable GarageFinish se decide completar como 0 ya que corresponde a los
#GarageYrBlt con NA, que se ha supuesto no tienen garaje
houses$GarageFinish[is.na(houses$GarageFinish)]<- 0

#Para la variable MasVnrArea se decide completar los vacíos con 0 ya que todos tiene
#la variable MasVnrType a None o lo tenían vacío
houses$MasVnrArea[is.na(houses$MasVnrArea)]<- 0

#Para la variable MSZoning se decide completar con la moda (al ser categórica)
#acorde al vecindario
if(is.na(houses$MSZoning[i]))
{
  houses$MSZoning[i]<-mode(houses$MSZoning[houses$Neighborhood==houses$Neighborhood[i]])
}

#Para la variable LotFrontage se deciden completar los vacíos con la mediana
#(al ser numérica) por vecindario
if(is.na(houses$LotFrontage[i]))
{

  houses$LotFrontage[i]<-median(houses$LotFrontage[houses$Neighborhood==houses$Neighborhood[i]],
                                 na.rm=TRUE)
}

#Para la variable Utilities se decide completar con la moda (al ser categórica) acorde
#al vecindario
if(is.na(houses$Utilities[i]))
{
  houses$Utilities[i]<-mode(houses$Utilities[houses$Neighborhood==houses$Neighborhood[i]])
}

#Para la variable Exterior1st se decide completar con la moda (al ser categórica) acorde
#a la variable RoofMtl
if(is.na(houses$Exterior1st[i]))
{
  houses$Exterior1st[i]<-mode(houses$Exterior1st[houses$RoofMatl==houses$RoofMatl[i]])
}

#Para la variable Exterior2nd se decide completar con la moda (al ser categórica) acorde
#a la variable RoofMtl
if(is.na(houses$Exterior2nd[i]))
{
  houses$Exterior2nd[i]<- mode(houses$Exterior2nd[houses$RoofMatl==houses$RoofMatl[i]])
}

#Para la variable Functional se decide completar con la moda (al ser categórica)
if(is.na(houses$Functional[i]))
{
  houses$Functional[i]<-mode(houses$Functional)
}

```

```

    }
}
```

En ocasiones, se suele utilizar otros métodos de imputación de los vecinos más cercanos (Knn etc) #Para la variable Electrical se detecta solo 1 vacío, se decide completar con la moda (al ser categórica) acorde al vecindario partiendo de la #hipótesis de que la instalación de los servicios públicos en el vecindario es común

```
library(VIM)
```

```

## Loading required package: colorspace
## VIM is ready to use.
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
##
## Attaching package: 'VIM'
##
## The following object is masked from 'package:missForest':
## 
##     nrmse
##
## The following object is masked from 'package:mclust':
## 
##     diabetes
##
## The following object is masked from 'package:datasets':
## 
##     sleep
distancia=c("BldgType", "Neighborhood", "MSZoning", "GrLivArea", "Heating", "CentralAir")

houses <- kNN(houses, variable = c("Electrical"), k=3, dist_var = distancia, trace= FALSE)
```

Aplicando la moda a esta misma variables la gestión de valores nulos hubiera quedado de la siguiente forma:

```

#for (i in 1:nrow(houses))
# if(is.na(houses$Electrical[i]))
# {
#   houses$Electrical[i] <-
#   mode(houses$Electrical[houses$Neighborhood==houses$Neighborhood[i]])
# }
```

```
colSums(houses==0)
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea
##	0	0	0	0	0
##	Street	Alley	LotShape	LandContour	Utilities
##	0	0	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
##	0	869	0	0	0

```

##      BsmtQual      BsmtCond      BsmtExposure      BsmtFinType1      BsmtFinSF1
##      0             0             0                 0                  467
##      BsmtFinType2    BsmtFinSF2      BsmtUnfSF      TotalBsmtSF      Heating
##      0             1293            118                37                  0
##      HeatingQC    CentralAir      Electrical      X1stFlrSF      X2ndFlrSF
##      0             0               0                  0                  829
##      LowQualFinSF    GrLivArea    BsmtFullBath    BsmtHalfBath      FullBath
##      1434            0              856                1378                 9
##      HalfBath     BedroomAbvGr  KitchenAbvGr  KitchenQual  TotRmsAbvGrd
##      913             6              1                  0                  0
##      Functional    Fireplaces  FireplaceQu  GarageType  GarageYrBlt
##      0              690            0                  0                  81
##      GarageFinish   GarageCars  GarageArea  GarageQual  GarageCond
##      81              81              81                  0                  0
##      PavedDrive    WoodDeckSF  OpenPorchSF EnclosedPorch  X3SsnPorch
##      0              761            656                1252                1436
##      ScreenPorch    PoolArea   PoolQC       Fence      MiscFeature
##      1344            1453            0                  0                  0
##      MiscVal        MoSold    YrSold      SaleType  SaleCondition
##      1408            0              0                  0                  0
##      SalePrice Electrical_im
##      0              1459

```

Se observa que existen valores a 0 en las siguientes variables:

BsmtFinSF2 , BsmtUnfSF, TotalBsmtSF, LowQualFinSF, BsmtFullBath, BsmtHalfBath, HalfBath, BedroomAbvGr, KitchenAbvGr, Fireplaces, GarageCars, GarageArea, WoodDeckSF, OpenPorchSF, nclosedPorch, MiscVal, X3SsnPorch, ScreenPorch

Para todas ellas, pueden un valor perfectamente valido , ya que se refieren al nº de metros, o de habitacion/plazas de garange o chimeneas y consideramos que dichos valores no requieren de ningnºn tratamiento.

Las variables que son nmericas (años, o metros cuadrados etc) tienen como tipo de dato el que le correponde por lo que no hacemos transformación. Sin embargo, las variables cualitativas/categoricas que son cadenas de textos y deberíamos pasar a variables de tipo factor en R. Se realiza la conversión mediante el siguiente bucle.

```

for(i in 1:ncol(houses)) {
  if (is.character(houses[,i])){
    houses[,i] <- as.factor(houses[,i])
  }
}

#Se podria mostrar de nuevo la siguiente instrucción para ver los factors convertidos
#sapply(houses, function(x) class(x))

```

Se revisa el total de valores unicos por variable

```
apply(houses, 2, function(x) length(unique(x)))
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea
##	1460	15	5	115	1073
##	Street	Alley	LotShape	LandContour	Utilities
##	2	3	4	4	2
##	LotConfig	LandSlope	Neighborhood	Condition1	Condition2
##	5	3	25	9	8
##	BldgType	HouseStyle	OverallQual	OverallCond	YearBuilt

##	5	8	10	9	112
##	YearRemodAdd	RoofStyle	RoofMatl	Exterior1st	Exterior2nd
##	61	6	8	15	16
##	MasVnrType	MasVnrArea	ExterQual	ExterCond	Foundation
##	5	327	4	5	6
##	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	BsmtFinSF1
##	5	5	5	7	637
##	BsmtFinType2	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	Heating
##	7	144	780	721	6
##	HeatingQC	CentralAir	Electrical	X1stFlrSF	X2ndFlrSF
##	5	2	5	753	417
##	LowQualFinSF	GrLivArea	BsmtFullBath	BsmtHalfBath	FullBath
##	24	861	4	3	4
##	HalfBath	BedroomAbvGr	KitchenAbvGr	KitchenQual	TotRmsAbvGrd
##	3	8	4	4	12
##	Functional	Fireplaces	FireplaceQu	GarageType	GarageYrBlt
##	7	4	6	7	98
##	GarageFinish	GarageCars	GarageArea	GarageQual	GarageCond
##	4	5	441	6	6
##	PavedDrive	WoodDeckSF	OpenPorchSF	EnclosedPorch	X3SsnPorch
##	3	274	202	120	20
##	ScreenPorch	PoolArea	PoolQC	Fence	MiscFeature
##	76	8	4	5	5
##	MiscVal	MoSold	YrSold	SaleType	SaleCondition
##	21	12	5	9	6
##	SalePrice	Electrical_imp			
##	663	2			

Outliers (Extreme scores) - Identificación y tratamiento de valores extremos.

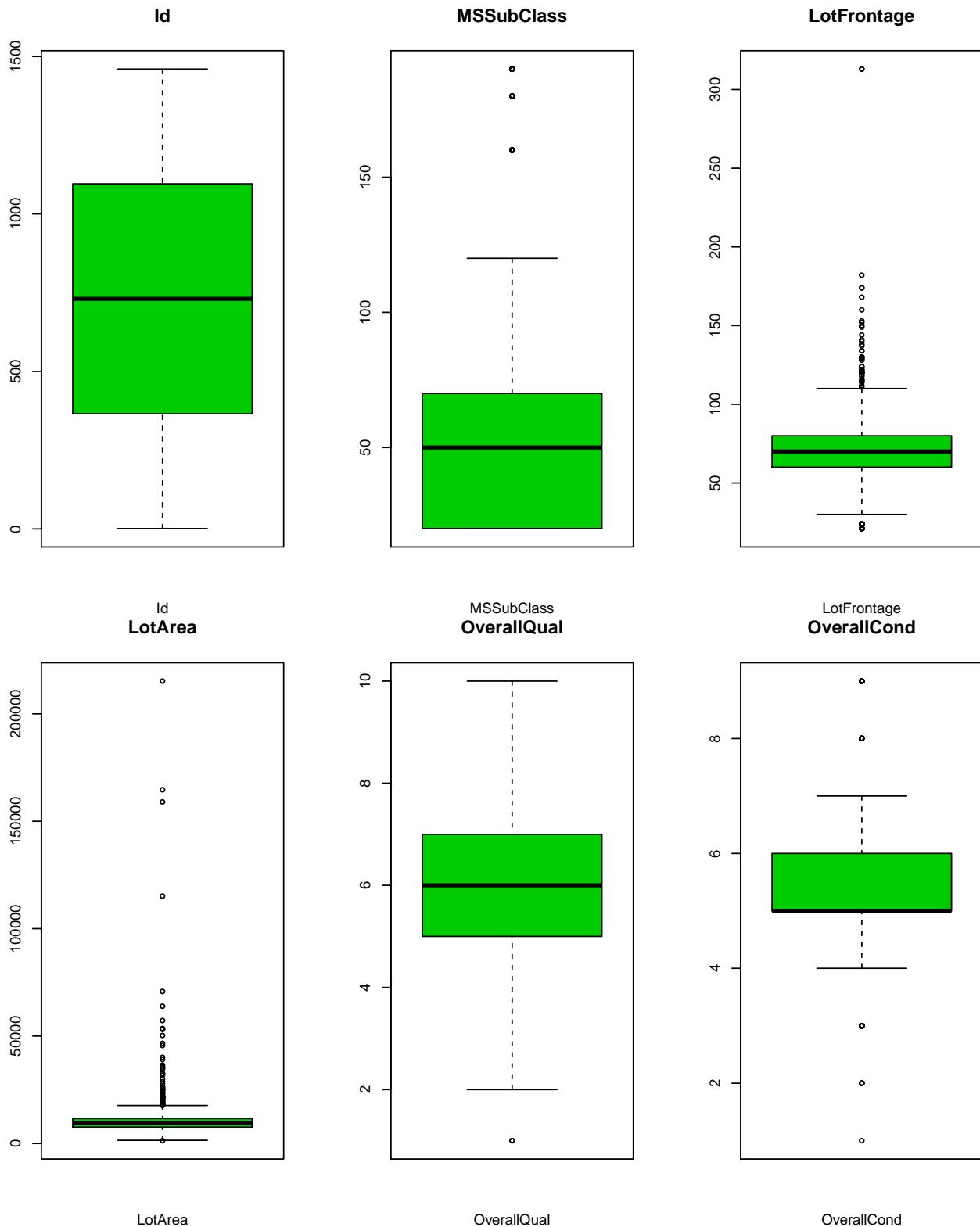
Las variables del dataset pueden tener valores anómalos o valores atípicos.

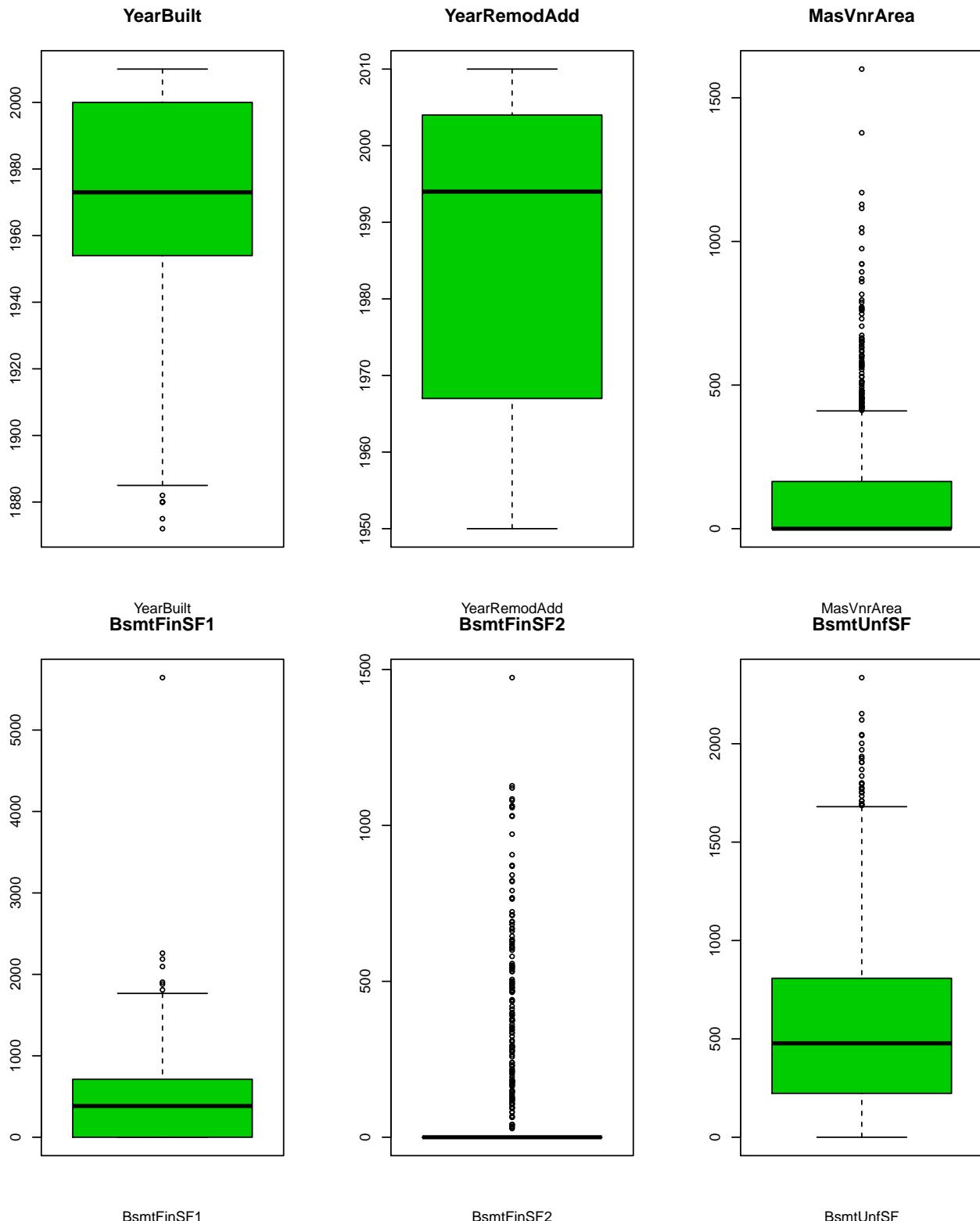
Un extreme score / outlier un valor que es bastante diferente (por encima o por debajo) de los valores restantes de la muestra y no sigue la distribución de los datos. Puede ser un error en la toma de los datos o bien puede ser algún caso excepcional que deba analizarse porque nos proporcione valor añadido al estudio. Para identificarlos, podemos hacer uso de dos opciones: representar un diagrama de caja por cada variable y ver qué valores distan mucho del rango intercuartílico (la caja) o una función en R denominada boxplot_stats

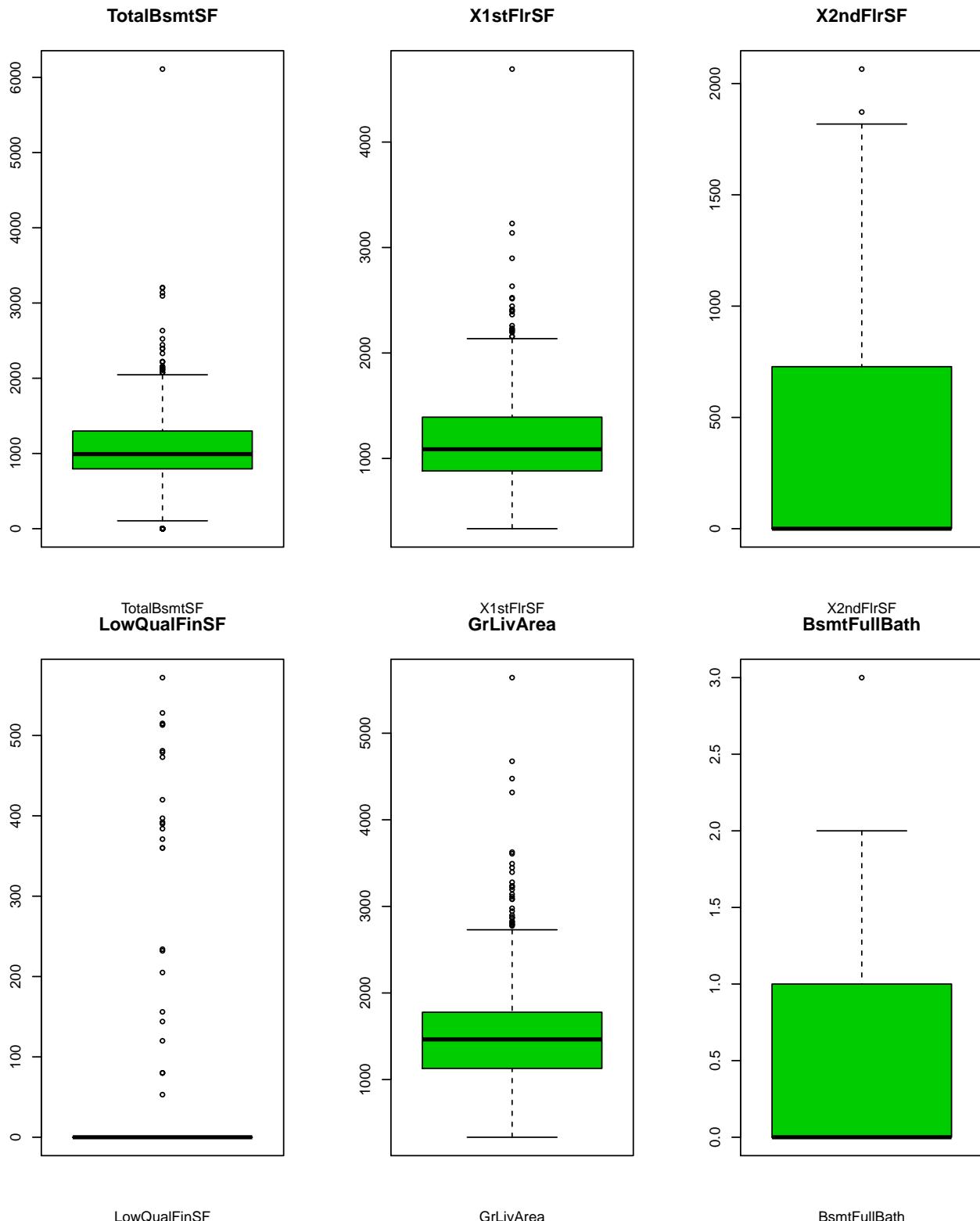
Si se consideran que son anómalos ó que distorsionan la muestra, se pueden eliminar. Si se considera que puede ser de un error de conversión entre medidas diferentes(metros/pies etc.) puede realizarse alguna operación matemática para arreglarlo.

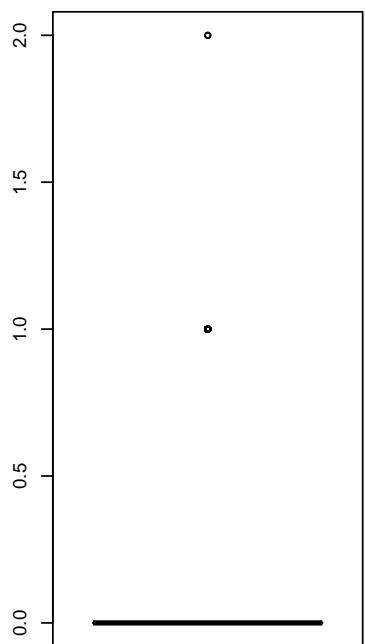
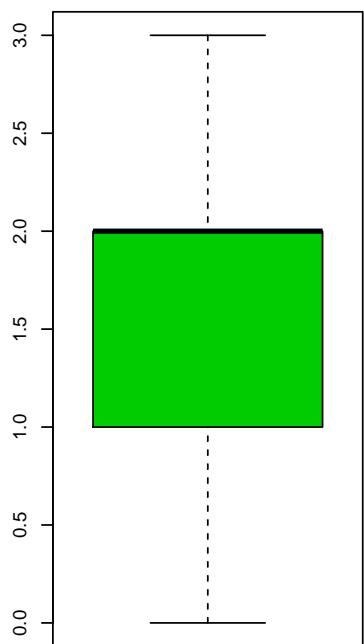
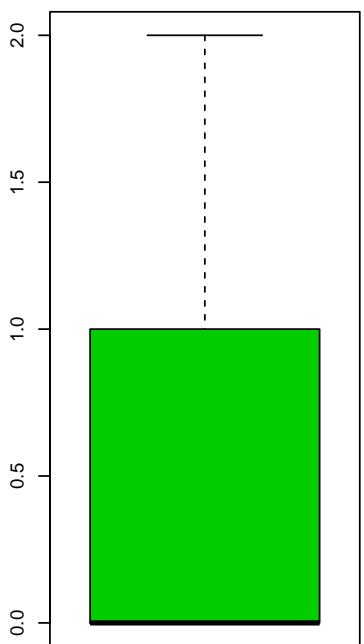
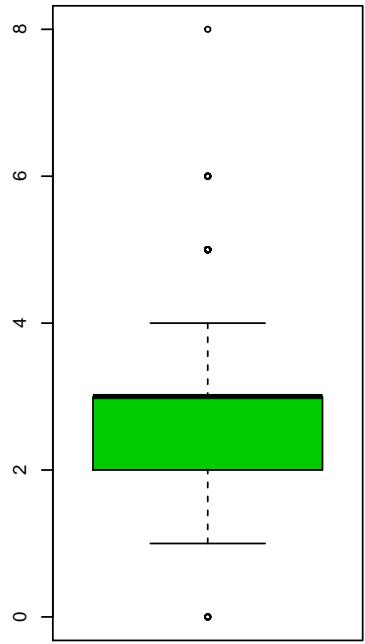
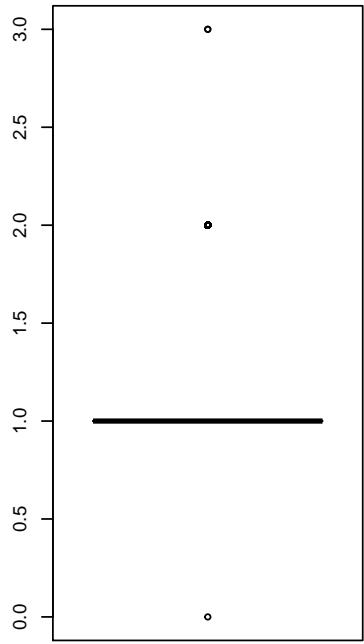
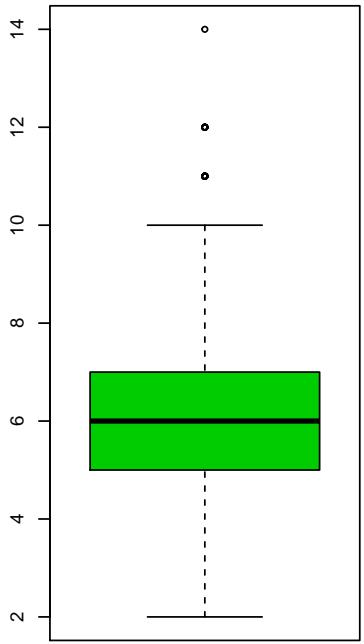
Así, se mostrarán sólo los valores atípicos para aquellas variables que los contienen:

```
par(mfrow=c(1,3))
for(i in 1:ncol(houses)) {
  if (is.numeric(houses[,i])){
    boxplot(houses[,i], main = colnames(houses)[i] ,width = 50 ,
            xlab=colnames(houses)[i],col= "green3")
  }
}
```





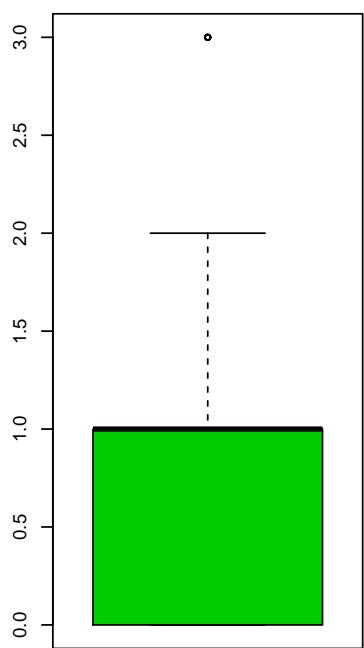
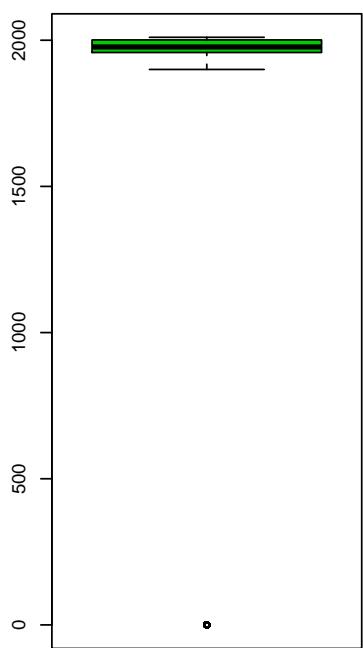
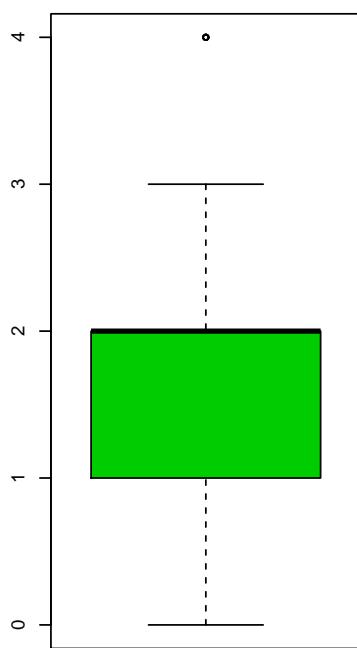
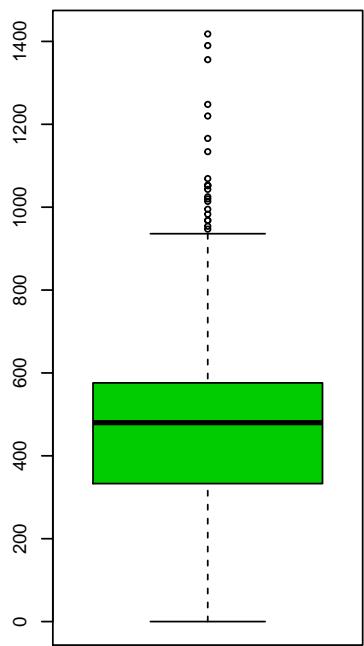
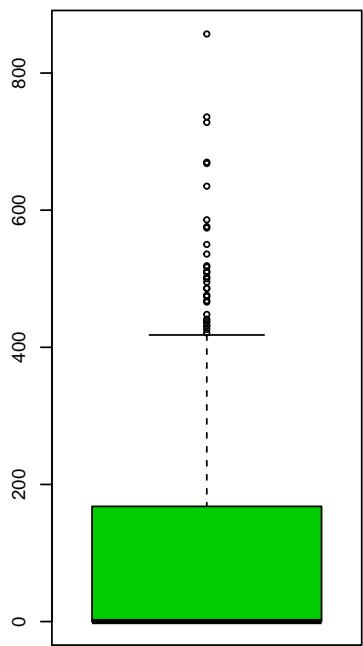
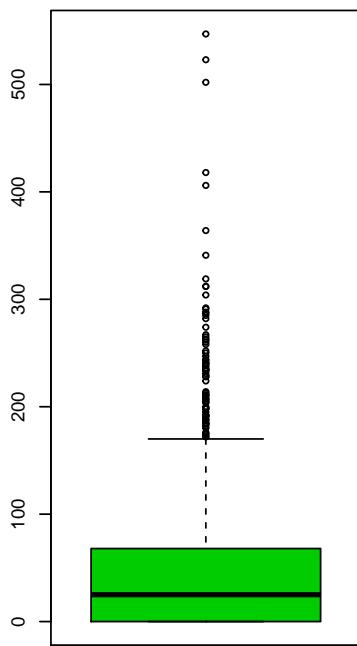


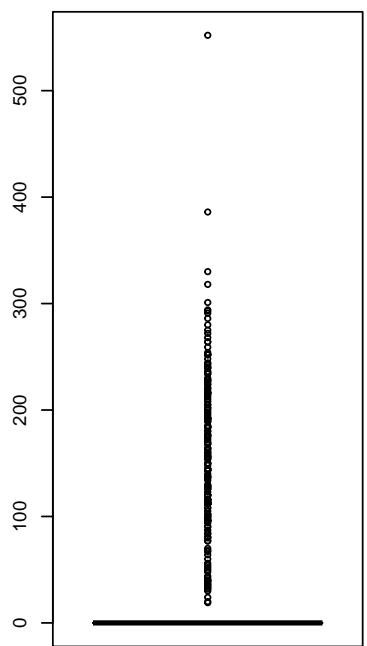
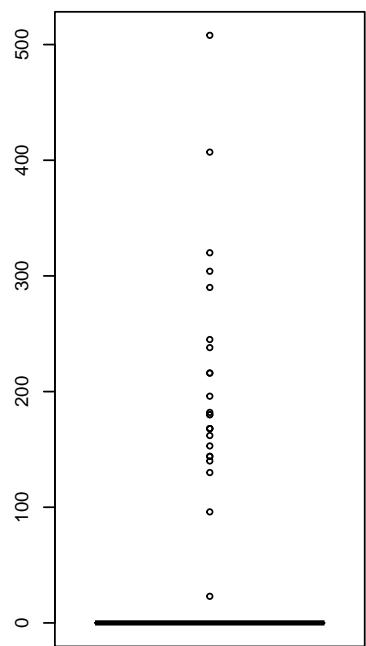
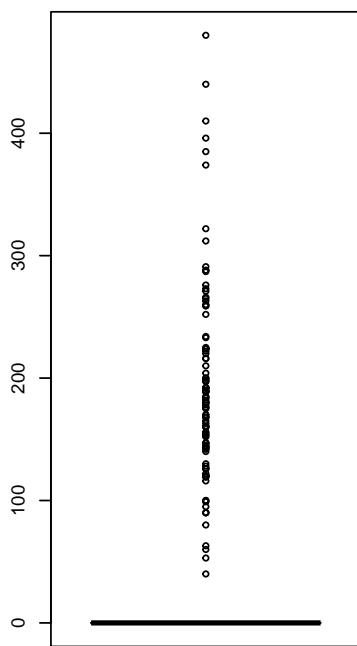
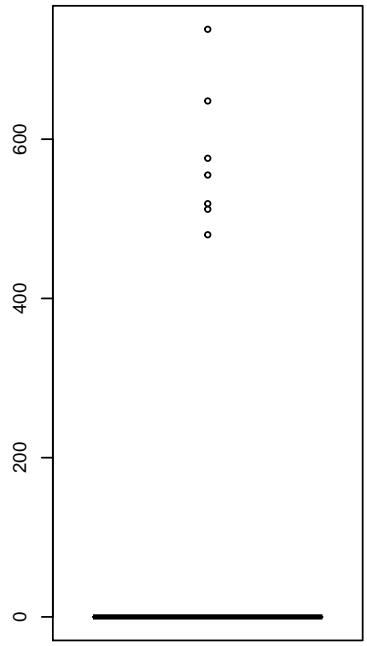
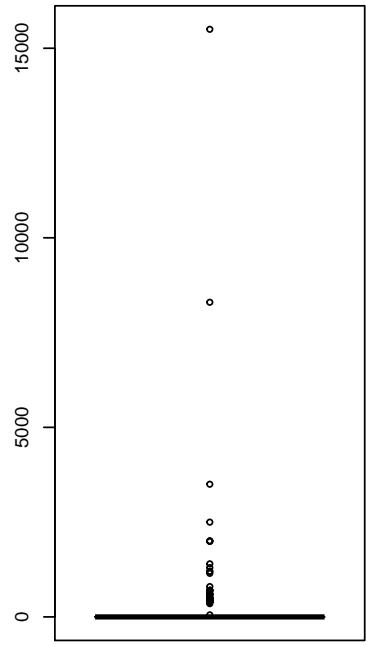
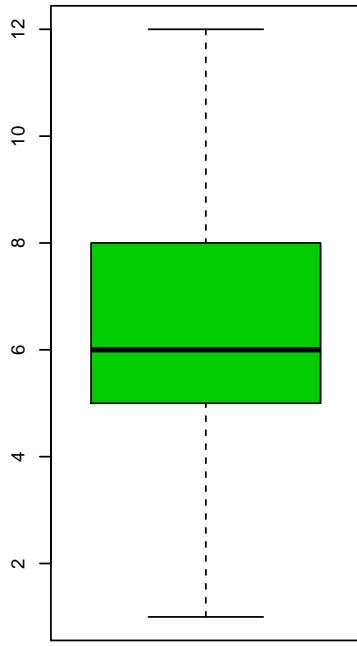
BsmtHalfBath**FullBath****HalfBath****BsmtHalfBath
BedroomAbvGr****FullBath
KitchenAbvGr****HalfBath
TotRmsAbvGrd**

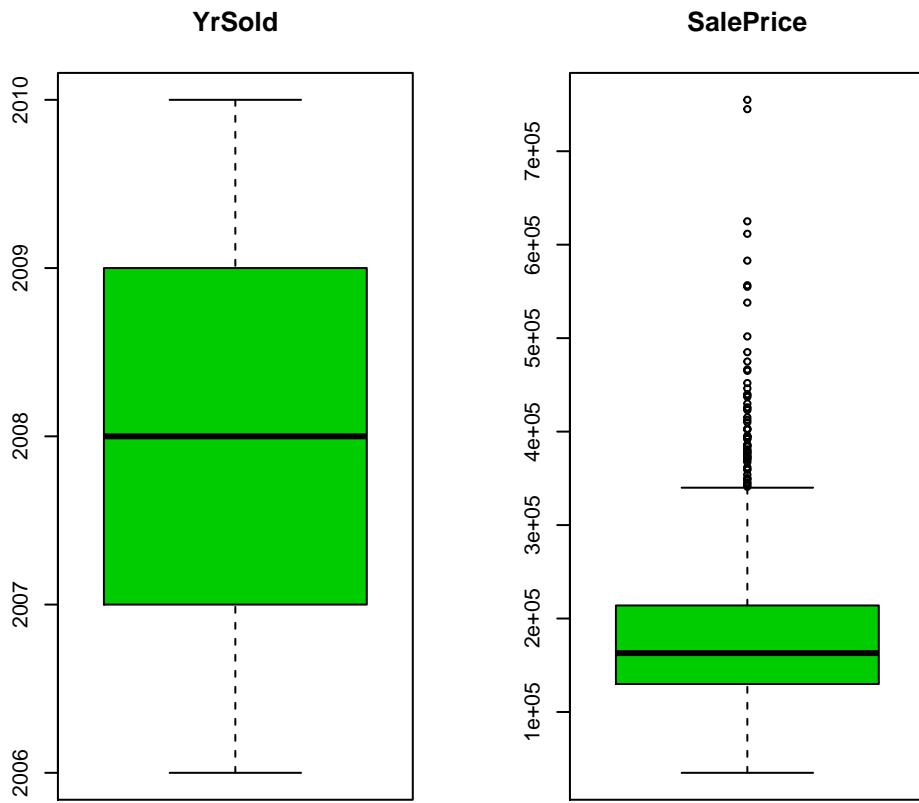
BedroomAbvGr

KitchenAbvGr

TotRmsAbvGrd

Fireplaces**GarageYrBlt****GarageCars****Fireplaces
GarageArea****GarageYrBlt
WoodDeckSF****GarageCars
OpenPorchSF****GarageArea****WoodDeckSF****OpenPorchSF**

EnclosedPorch**X3SsnPorch****ScreenPorch****EnclosedPorch
PoolArea****X3SsnPorch
MiscVal****ScreenPorch
MoSold****PoolArea****MiscVal****MoSold**



YrSold **SalePrice** Se observan outliers en las siguientes variables : * MSSubClass - Esta variable son códigos que identifican distintos tipos de vivienda. Por tanto no se puede considerar que sean valores extremos o anómalos, ya que son códigos asignados por la codificación pactada. No requiere tratamiento adicional

- LotFrontage - Esta variable indica el nº de pies (longitud) . También consideramos que puede haber propiedades en venta que tengan un mayor nº de pie. No requiere tratamiento.
 - LotArea - Esta variable indica el nº de metros cuadrados de la parcela. Aunque existen valores que están alejados de la media del resto de viviendas, vemos lógico que puedan ser un nº valido. No requiere tratamiento.
 - OverallQual : Los valores estan dentro de los permitidos , de 0-10 , aunque la mayoria se situan en la calidad media , pero este se correponde con alguna propiedad que no estará en buenas condiciones. No requiere tratamiento , ya que es nos da referencia de los precios con dicha calidad.
 - OverCond : Es similar , son todos valores permitidos , aunque la mayoria esten entre los valores intermedios.No requiere tratamiento.
 - YearBuilt : Año de construcción. En este caso , se opta por quitarlos porque distan mucho de la media situados entre los años 1950 - 2000 approximaamente, por lo que puede distorsionar la muestra.

```
outliers_yb <- boxplot.stats(houses$YearBuilt)$outliers  
outliers_yb
```

```

## [1] 1880 1880 1880 1882 1880 1875 1872

idx_out<- which(houses$YearBuilt %in% outliers_yb)
#Se eliminan las observaciones relativos a dichos outliers
houses <- houses[ -idx_out, ]

```

- MassVnrArea : N° de metros de mampostería . Se considera que aunque haya outliers , puede ser que correspondan con las casas más grandes . No se realiza tratamiento.
- BsmtFinSF1 - Se considera que puede ser normal con el resto de las variables . No se realiza tratamiento (Son el n° de metros del sótano acabados)
- BsmtFinSF2 - Se considera que puede ser normal con el resto de las variables . No se realiza tratamiento (Son el n° de metros del sótano2 acabados)
- BsmtUnSF - Se considera que puede ser normal con el resto de las variables . No se realiza tratamiento
- TotalBsmtSF - idem a las anteriores atributos. El n° de metros puede ser elevado en propiedades muy grandes . No se realiza tratamiento.
- 1stFlrSF: Metros cuadrados de la primera planta - Idem, puede ser normal esos valores en propiedades grandes y por tanto las que mayor precio tengan.
- 2ndFlrSF: Metros cuadrados de la segunda planta Idem, puede ser normal esos valores en propiedades grandes y por tanto las que mayor precio tengan.
- GrLiveArea - idem a las anteriores atributos. El n° de metros puede ser elevado en propiedades muy grandes . No se realiza tratamiento.
- BsmtFullBath - El n° de baños . Es un n° perfectamente valido (3 baños completos en una casa grande) . No se realiza tratamiento
- BsmtHalfBath - Aquí lo que se observa es que la mayoría no tiene. Esta columna será una de no las quitemos ya que no parece que vaya aportarnos.

```
houses2 <- houses[ , -houses$BsmtHalfBath ]
```

- LowQualFinSF - Son valores validos, aunque la mayoría son cero. Son los metros de baja calidad que han dejados .De momento no se realiza tratamiento (esta columna seguramente no la utilicemos y la quitarmos)
- BedroomAbvGr - Son valores validos, son el n° de habitaciones en las plantas . No se requiere tratamiento.
- TotRmsAbrGr - Son valores validos. Es el n° de habitaciones totales sin contar la casa que tiene la propiedad. Los valores extremos se corresponderán con las más grandes seguramente. 12 y 14 habitaciones no suele ser lo habitual.
- Fireplaces - Tambien dejaremos ya que no es un valor tan extremo (3 chimeneas).Seguramente de las casas que se salen fuera de lo normal.
- GarageCars - También tiene un outlier que es 4 plazas de garaje, pero consideramos que puede ser un valor valido.

```
outliers_gc <- boxplot.stats(houses$GarageCars)$out
outliers_gc
```

```
## [1] 4 4 4 4
```

- WoodDeckSF: Área de cubierta de madera (metros cuadrados) Consideramos que pueden ser valores validos.No requiere tratamiento . Además esta columna no la vamos a tener en cuenta para el estudio.
- OpenPorchSF: Área del porche abierto (metros cuadrados) Consideramos que pueden ser valores validos.No requiere tratamiento .

```
outliers_op <- boxplot.stats(houses$OpenPorchSF)$out
outliers_op
```

```
## [1] 204 213 258 199 234 184 205 228 238 260 198 172 208 228 184 250 175
## [18] 195 214 231 192 187 176 523 285 406 182 502 274 172 243 235 312 267
```

```

## [35] 265 288 341 204 174 247 291 312 418 240 364 188 207 234 192 191 252
## [52] 189 282 224 319 244 185 200 180 263 304 234 240 192 229 211 198 287
## [69] 292 207 241 547 211 184 262 210 236

```

- EnclosedPorch: Area del porche cerrado (metros cuadrados) Consideramos que pueden ser valores validos.No requiere tratamiento . Además esta columna no la vamos a tener en cuenta para el estudio.
- 3SsnPorch: Area del porche 3 estaciones (metros cuadrados) Consideramos que pueden ser valores validos.No requiere tratamiento . Además esta columna no la vamos a tener en cuenta para el estudio.
- ScreenPorch: Area de la pantalla del porches (metros cuadrados) Consideramos que pueden ser valores validos.No requiere tratamiento . Además esta columna no la vamos a tener en cuenta para el estudio.
- PoolArea - La mayoria no tiene piscina y por tanto el resto aparecen como outliers. Seguramente esta columna completa la quitaremos.
- MiscVal - La mayoria son cero , pero el resto se refiere al importe en caracaristicas adicionales por lo que creo que se debe mantener.

```

outliers_mv <- boxplot.stats(houses$MiscVal)$out
outliers_mv

```

```

## [1] 700 350 700 500 400 700 480 400 400 450 450
## [12] 500 450 700 400 15500 1200 800 480 400 2000 2000
## [23] 600 500 600 600 3500 500 400 450 500 1300 1200
## [34] 500 400 54 500 400 400 2000 620 400 560 500
## [45] 700 1400 400 8300 600 1150 2000 2500

```

- SalesPrices - Hay valores que superan la media habitual de las ventas, y los vamos a descartar puesto que para este estudio , hacen que la media se distorsiones

```

outliers_sp <- boxplot.stats(houses$SalePrice)$out
outliers_sp

```

```

## [1] 345000 385000 438780 383970 372402 412500 501837 475000 386250 403000
## [11] 415298 360000 375000 342643 354000 377426 437154 394432 426000 555000
## [21] 440000 380000 374000 430000 402861 446261 369900 451950 359100 345000
## [31] 370878 350000 402000 423000 372500 392000 755000 361919 341000 538000
## [41] 395000 485000 582933 385000 350000 611657 395192 348000 556581 424870
## [51] 625000 392500 745000 367294 465000 378500 381000 410000 466500 377500
## [61] 394617

```

```

idx_out_sp<- which(houses$SalePrice %in% outliers_sp)
#Se eliminan las observaciones relativos a dichos outliers
houses <- houses[ -idx_out_sp, ]

```

Analisis de datos

Selección de los grupos de datos que se quieren analizar/comparar (planificación los análisis a aplicar).

Con el fin de ver que variables nos interesan analizar para ver como explican la variación de precio de las casas de Ames , podremos hacer primero un analisis exploratorio y posteriormente métodos de analisis y

estadísticos para validar lo que se observa.

Primero se resumen las variables cuantitativas o cualitativas que tenemos.

```
names <- colnames(houses)
```

```
#Definimos las variables cualitativas y las cuantitativas
varCualitativas <- which(names == "MSZoning" | names == "Street" | names == "Alley"
| names == "LotShape" | names == "LandContour"
| names == "Utilities"
| names == "LotConfig" | names == "LandSlope"
| names == "Neighborhood" | names == "Condition1"
| names == "Condition2" | names == "BldgType"
| names == "HouseStyle" | names == "RoofStyle"
| names == "RoofMatl" | names == "Exterior1st"
| names == "Exterior2nd" | names == "Foundation"
| names == "MasVnrType" | names == "RoofStyle"
| names == "ExterQual" | names == "ExterCond"
| names == "BsmtQual" | names == "BsmtCond"
| names == "BsmtExposure" | names == "BsmtFinType1"
| names == "BsmtFinType2" | names == "Heating"
| names == "HeatingQC" | names == "CentralAir"
| names == "Electrical" | names == "KitchenQual"
| names == "Functional" | names == "FireplaceQu"
| names == "GarageType" | names == "GarageFinish"
| names == "GarageQual" | names == "GarageCond"
| names == "PavedDrive" | names == "PoolQC"
| names == "Fence" | names == "MiscFeature"
| names == "SaleType" | names == "SaleCondition")
```



```
varCuantitativas <- which(names == "MSSubClass" | names == "LotFrontage"
| names == "LotArea" | names == "OverallQual"
| names == "OverallCond" | names == "YearBuilt"
| names == "YearRemodAdd" | names == "MasVnrArea"
| names == "BsmtFinSF1" | names == "BsmtFinSF2"
| names == "BsmtUnfSF" | names == "TotalBsmtSF"
| names == "X1stFlrSF" | names == "X2ndFlrSF"
| names == "LowQualFinSF" | names == "GrLivArea"
| names == "BsmtFullBath" | names == "FullBath"
| names == "HalfBath" | names == "BedroomAbvGr"
| names == "KitchenAbvGr" | names == "BsmtHalfBath"
| names == "TotRmsAbvGrd" | names == "Fireplaces"
| names == "GarageYrBlt" | names == "GarageCars"
| names == "GarageArea" | names == "WoodDeckSF"
| names == "OpenPorchSF" | names == "EnclosedPorch"
| names == "X3SsnPorch" | names == "ScreenPorch"
| names == "MiscVal" | names == "MoSold"
| names == "PoolArea" | names == "YrSold"
| names == "SalePrice")
```

Se hace un resumen de los distintos tipos valores de cada tipo. (Cualitativas)

```
options(knitr.kable.NA = '')
```

```
#kable(summary(houses)[,varCualitativas], digits=2, align='l',
#caption="Resumen descriptivo de variables cualitativas")
#kable(summary(houses)[,varCuantitativas],
```

```

#digits=2, align='l', caption="Resumen descriptivo de variables cuantitativas")

# Ahora se puede mostrar una tabla con medidas robustas/ no robustas
# Calculamos la media para todas las variables.
mean.n <- as.vector(sapply( houses[,varCuantitativas],mean,na.rm=TRUE ) )
# Calculamos la desviacion estandar para todas las variables.
std.n <- as.vector(sapply( houses[,varCuantitativas],sd, na.rm=TRUE))
# Calculamos la mediana para todas las variables.
median.n <- as.vector(sapply( houses[,varCuantitativas],median, na.rm=TRUE))
# Calculamos la media recortadas y winsor
mean.trim.0.05 <- as.vector(sapply( houses[,varCuantitativas],mean, na.rm=TRUE, trim=0.05))
mean.winsor.0.05 <- as.vector(sapply( houses[,varCuantitativas],winsor.mean, na.rm=TRUE,
                                         trim=0.05))
# Los parametros IQR (rango intercuartil)
IQR.n <- as.vector(sapply(houses[,varCuantitativas],IQR, na.rm=TRUE))
# Y desviación media absoluta desde la mediana.
mad.n <- as.vector(sapply( houses[,varCuantitativas],mad, na.rm=TRUE))
kable(data.frame(variables= names(houses)[varCuantitativas],Media = mean.n,
                  Mediana = median.n,Media.recort.0.05= mean.trim.0.05,
                  Media.winsor.0.05= mean.winsor.0.05),
      digits=2, caption="Estimaciones media - Tendencia Central")

kable(data.frame(variables= names(houses)[varCuantitativas],
Desv.Standard = std.n,
IQR = IQR.n,
MAD = mad.n),
digits=2, caption="Estimaciones de Dispersion")

```

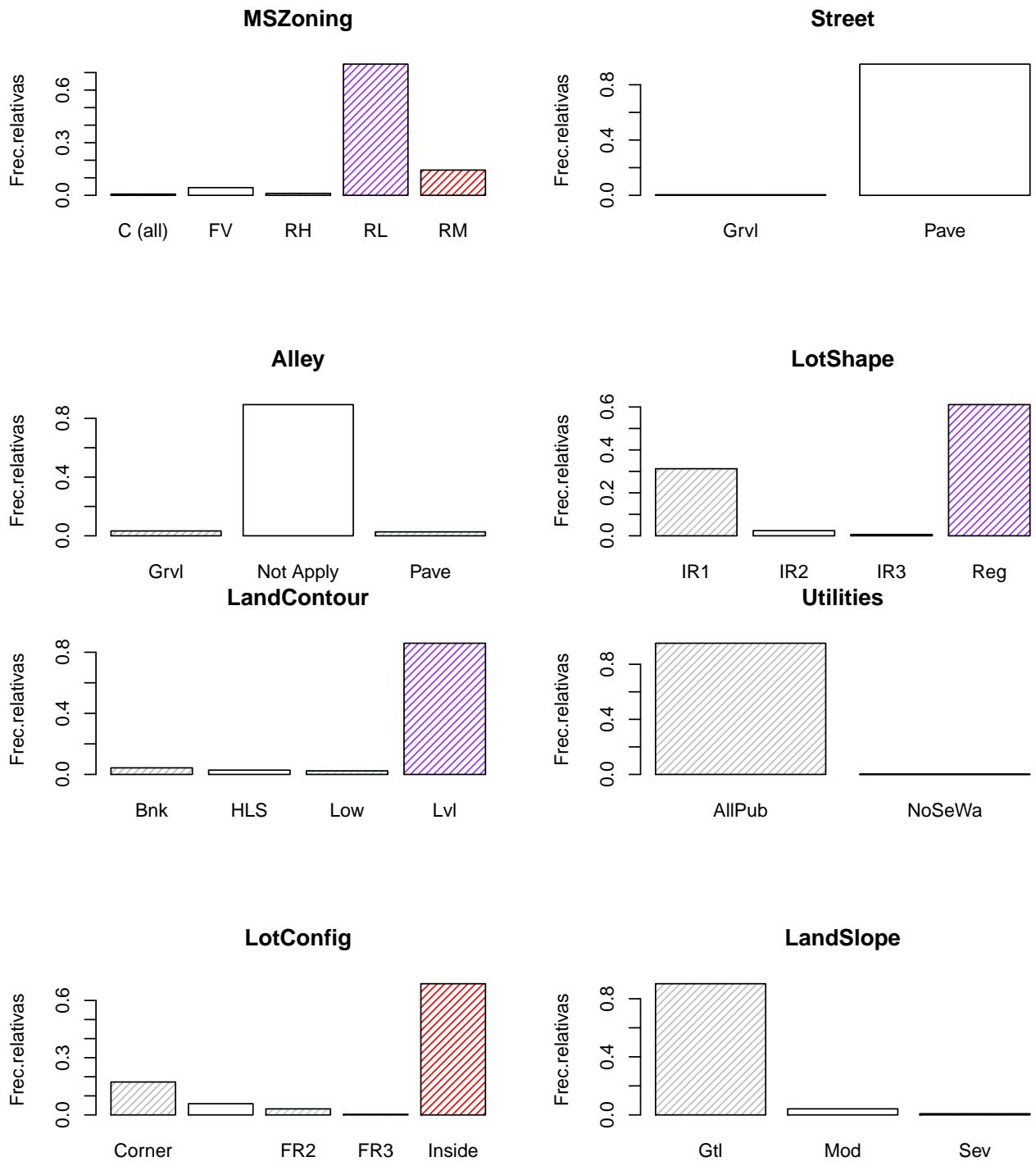
NOTA: En este caso las medidas de MAD = 0 ya nos da una idea que no al no haber desviacion , pueden no influir en nuestro estudio , por lo que la mayoría de esas columnas las eliminaremos.

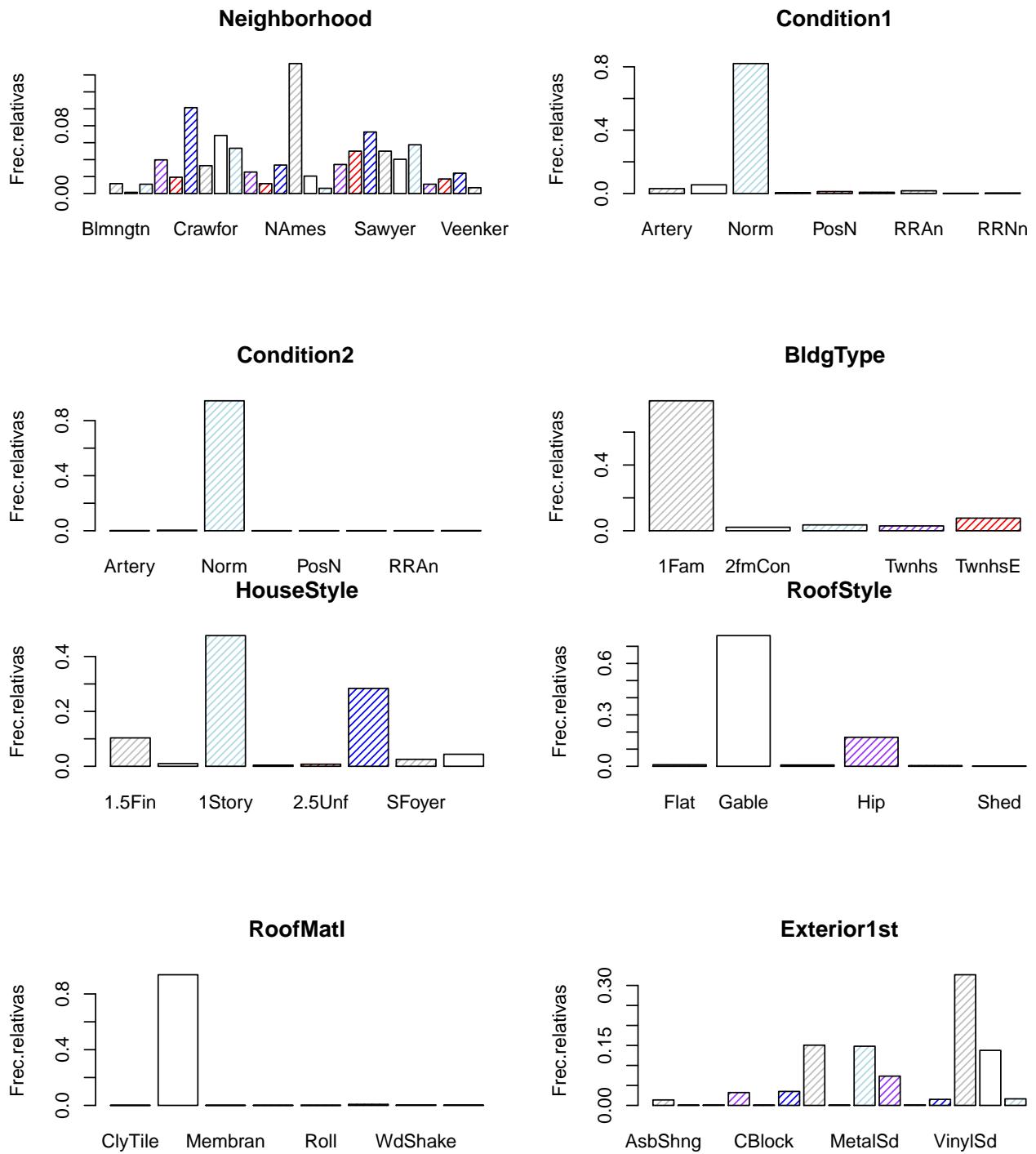
Para las variables cualitativas podemos ver como se distribuyen las categorias por frecuencias.(Absolutas o relativas)

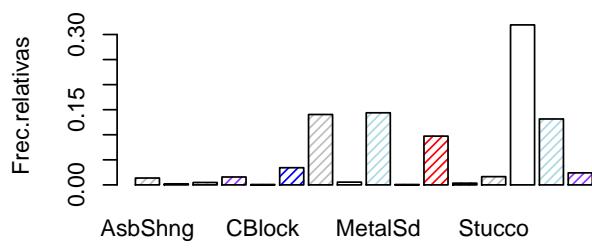
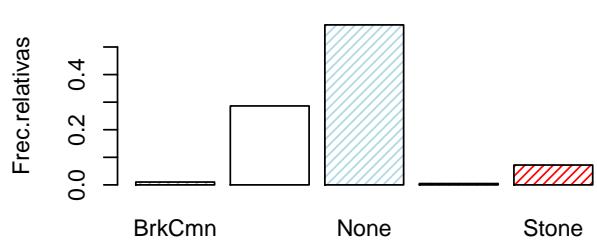
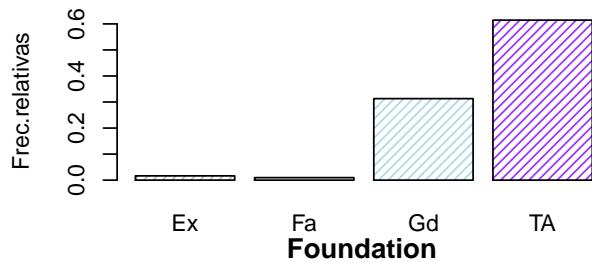
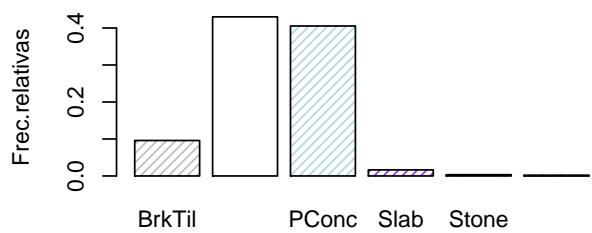
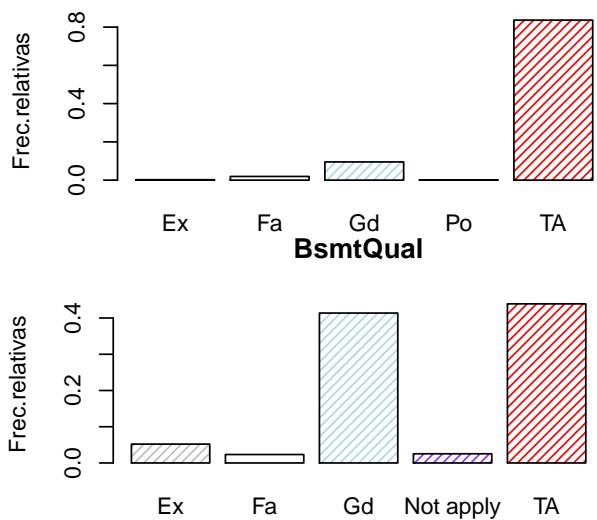
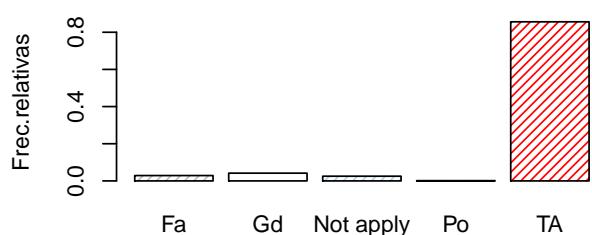
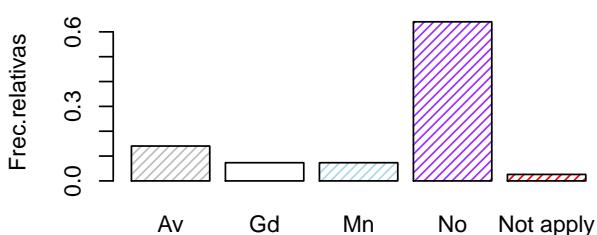
```

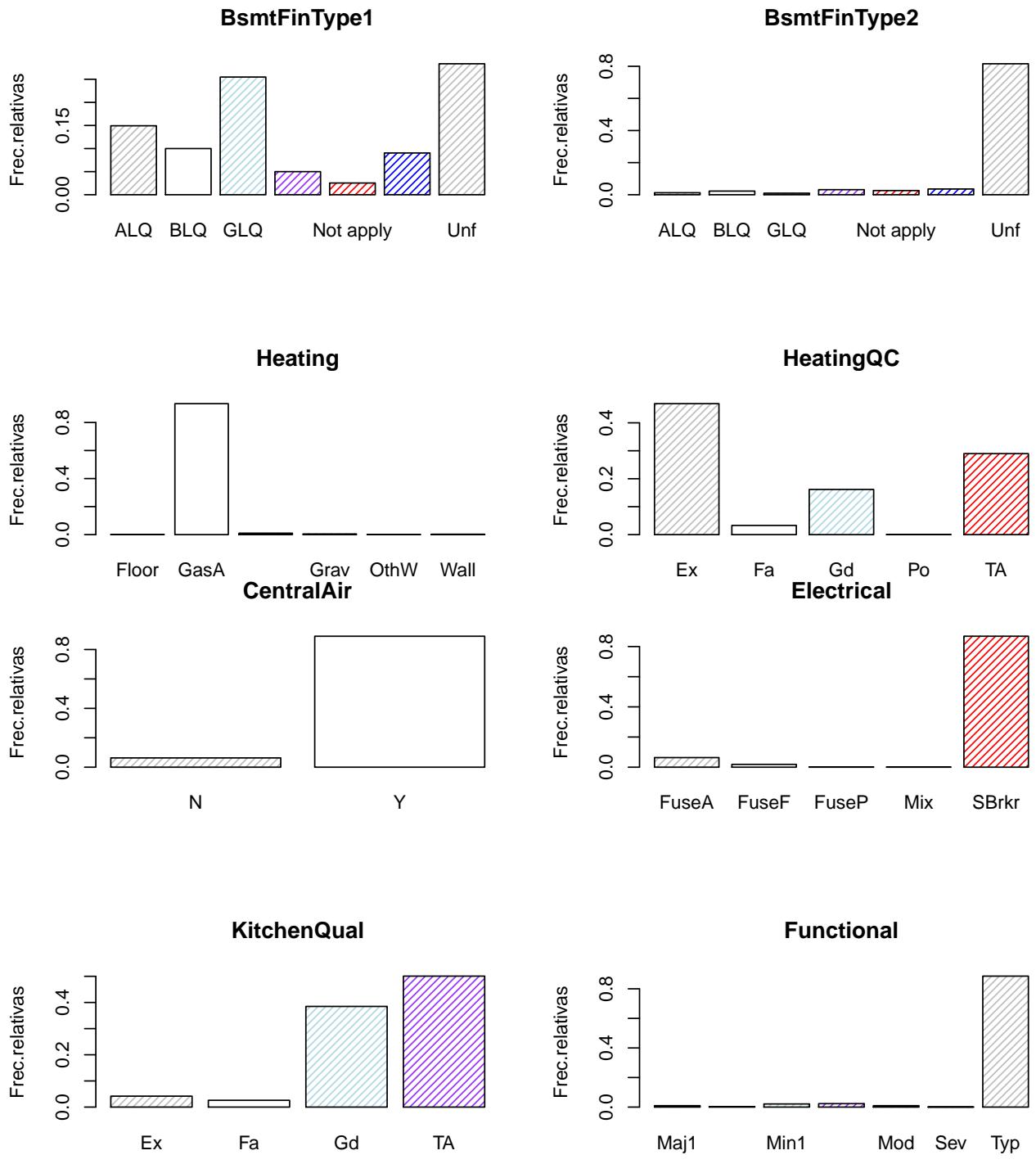
par(mfrow=c(2,2))
for(i in 1:ncol(houses)) {
  # Variable cualitativa - Grafico de barras
  if (is.factor(houses[,i])){
    fabs <- table(houses[i])
    frel <- fabs/n.ind
    # barplot(fabs,ylab="Frec.absol",main=colnames(houses)[i],
    # col=c('grey','white','lightblue','purple','mistyrose','blue'))
    barplot(frel,ylab="Frec.relativas", density = 25 ,main=colnames(houses)[i],
            col=c('grey','white','lightblue','purple1','red','blue'))
  }
}

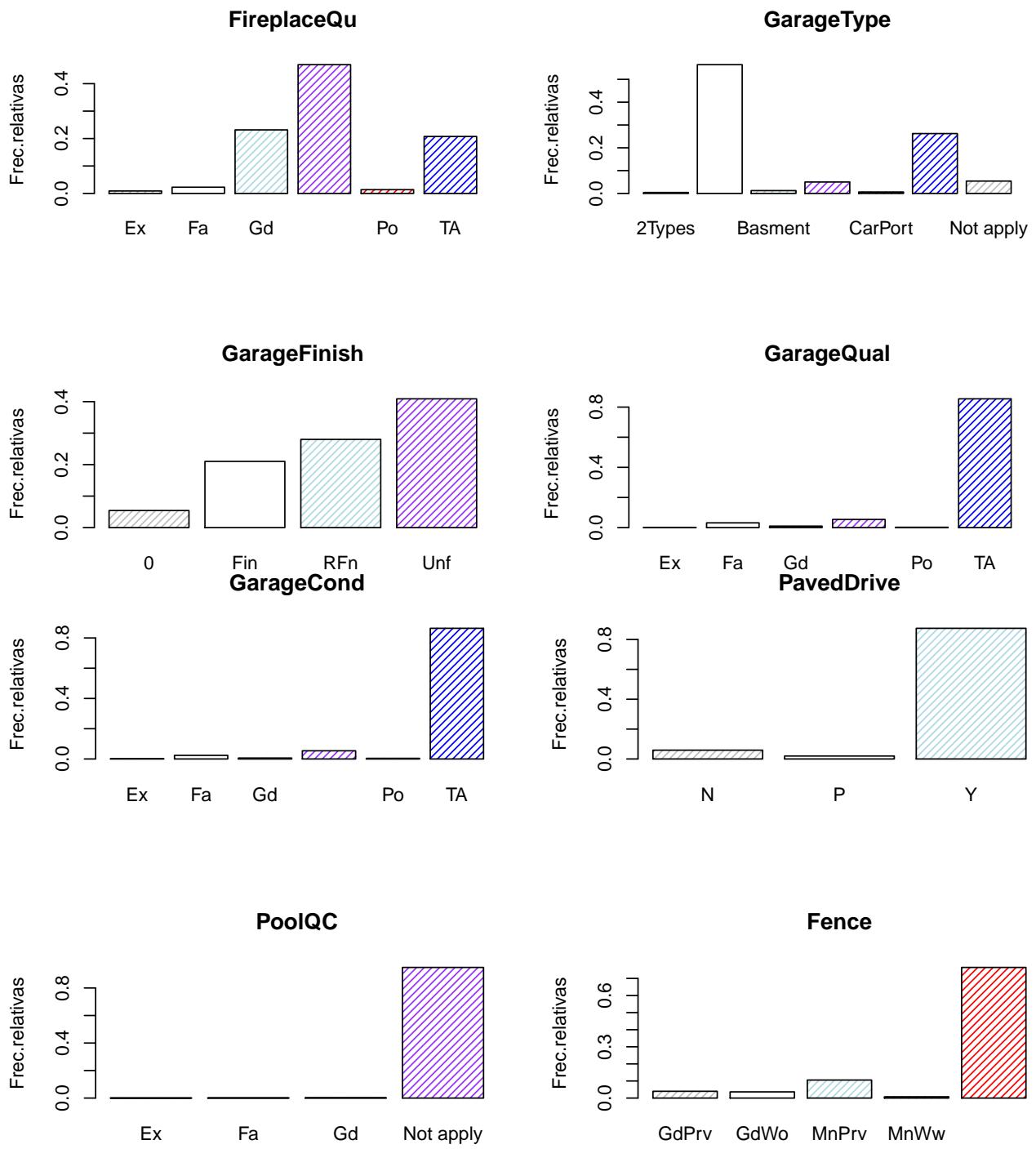
```

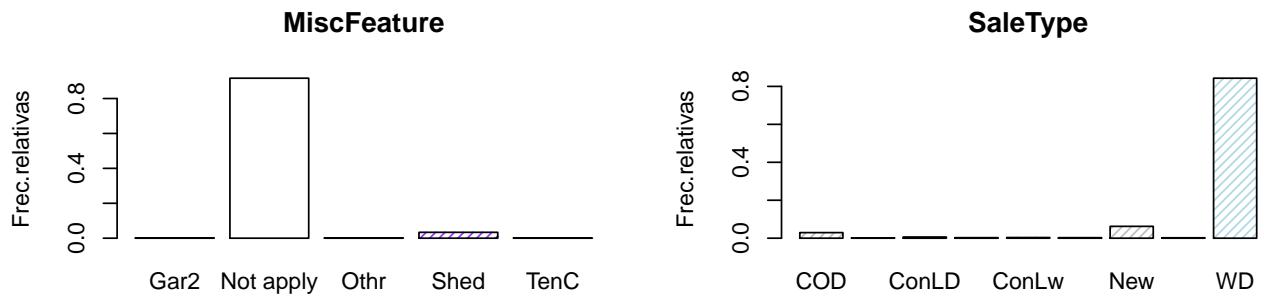




Exterior2nd**MasVnrType****ExterQual****Foundation****ExterCond****BsmtQual****BsmtCond**

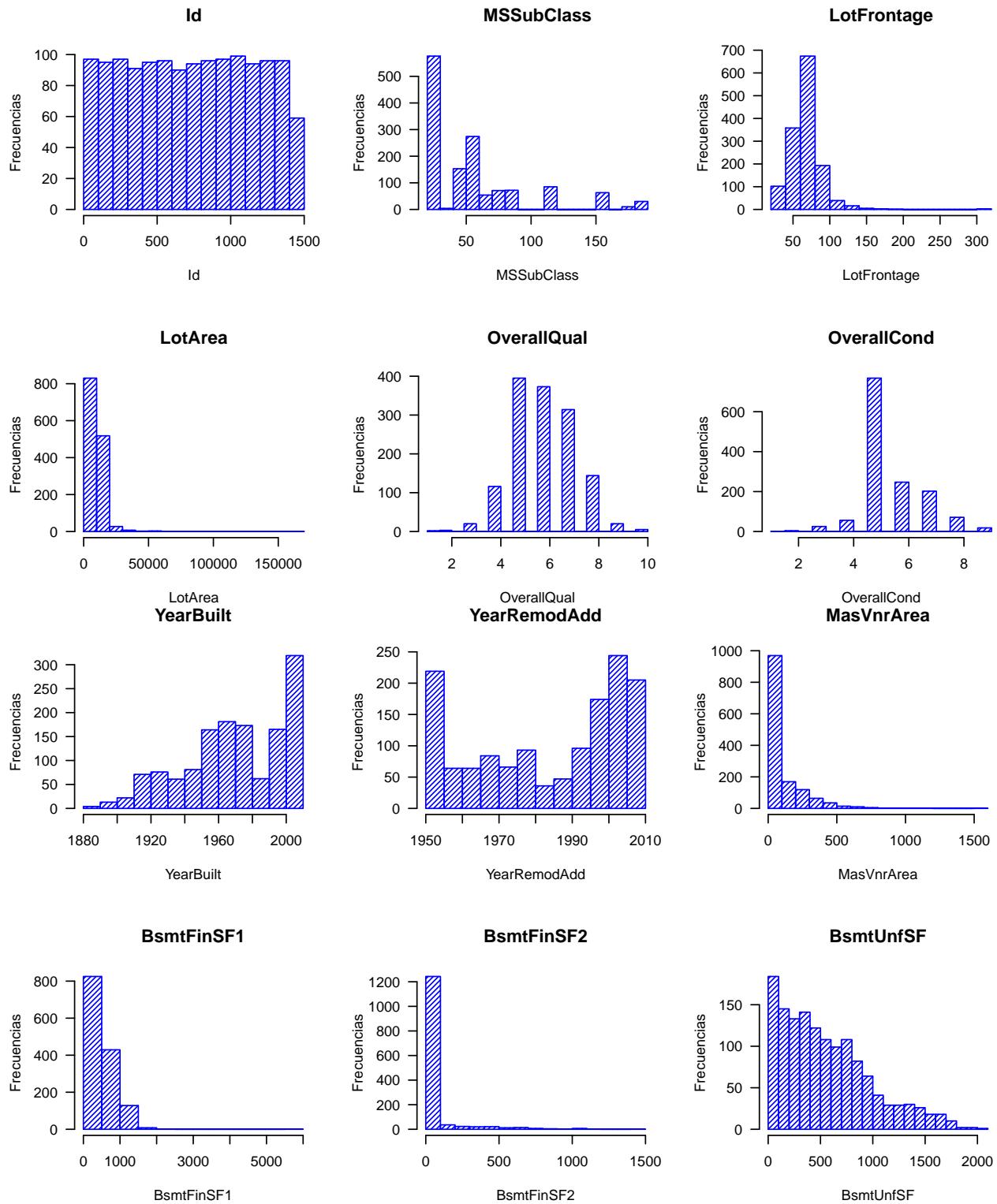


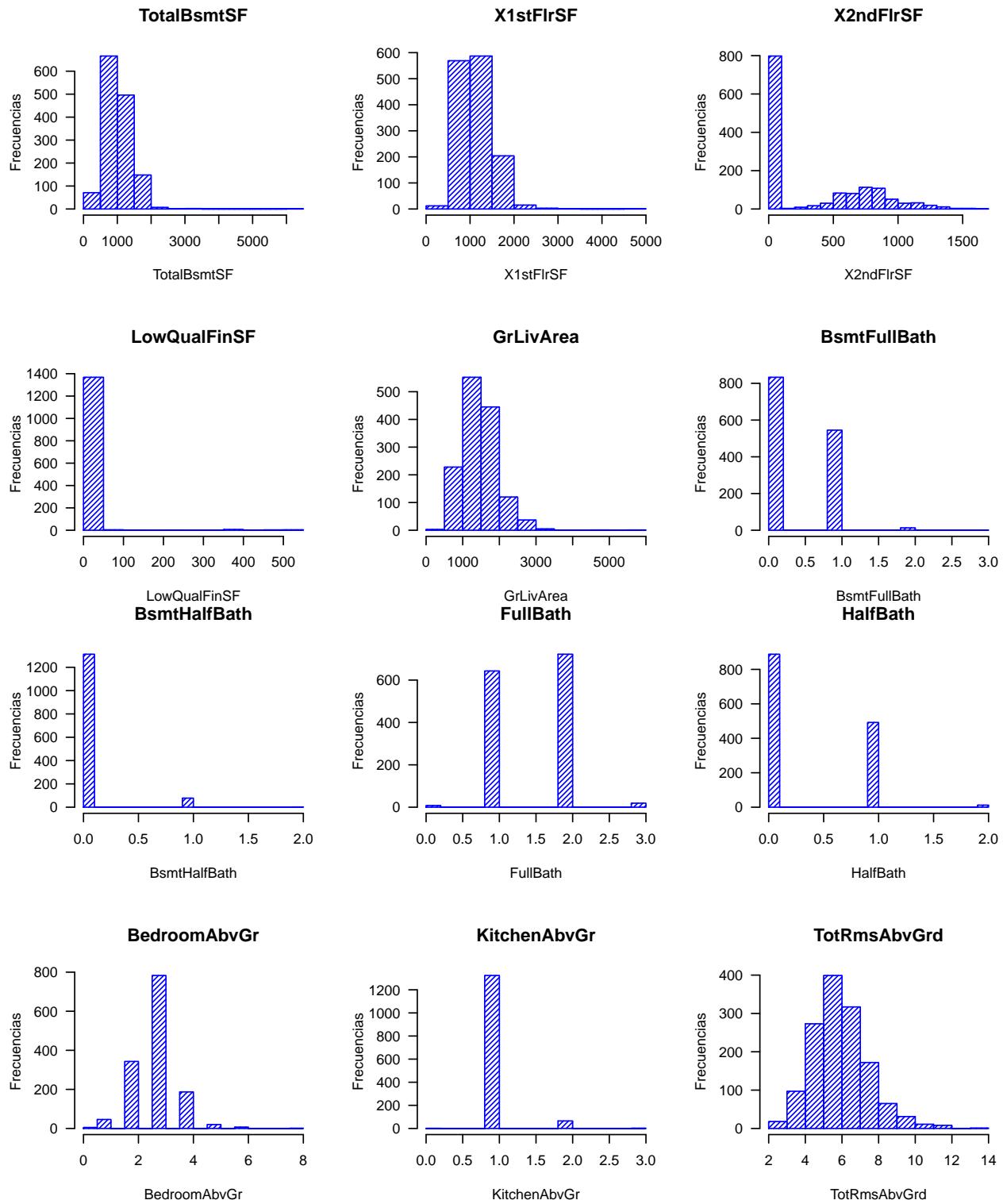


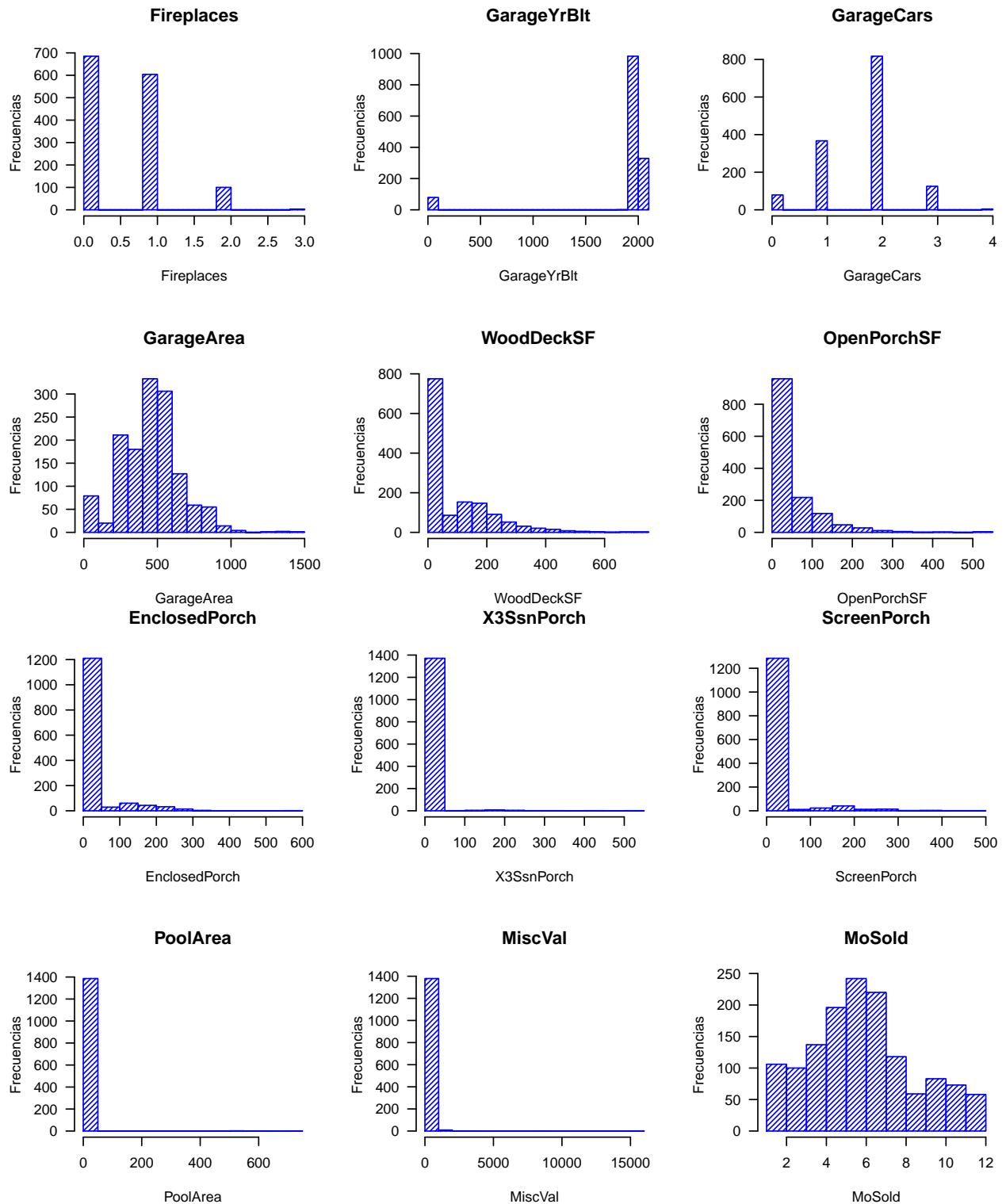


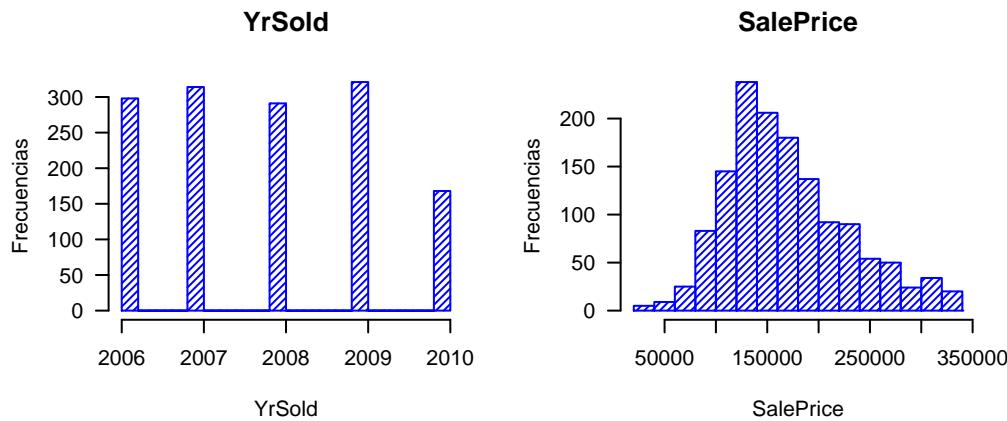
Y para las variables cuantitativas , podemos ver :

```
par(mfrow=c(2,3))
for(i in 1:ncol(houses)) {
  # Variable cualitativa - Grafico de barras
  if (is.numeric(houses[,i])){
    fabs <- table(houses[,i])
    frel <- fabs/n.ind
    hist(x=houses[,i], breaks=15, main=colnames(houses)[i], xlab = colnames(houses)[i],
          ylab="Frecuencias", las=1, col = "blue", density = 35)
  }
}
```









En relación a la dimensionalidad del dataset y tras el estudio de los atributos se decide crear para el total baños una variable nueva con el sumatorio de todos los baños:

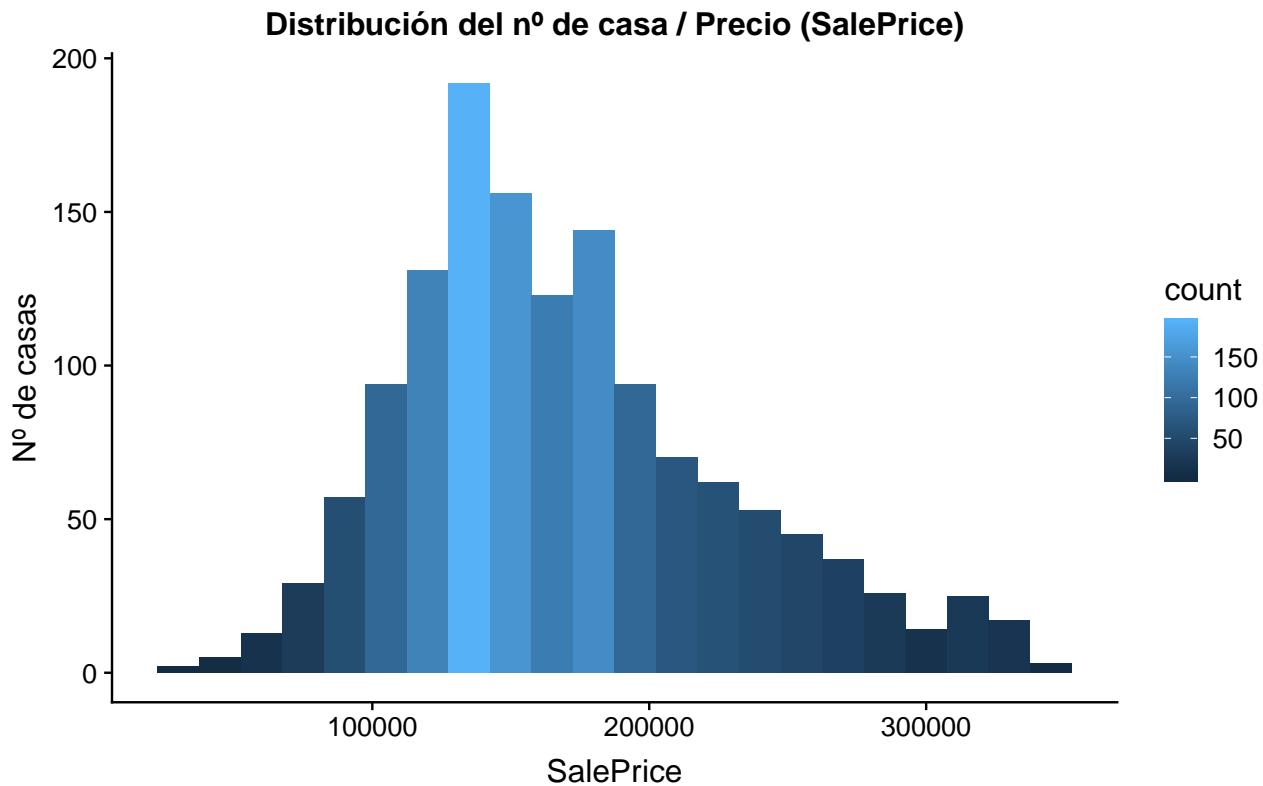
```
# Se calcula el total de baños en la propiedad en una nueva variable
houses$TotalBaths <- houses$BsmtFullBath + houses$BsmtHalfBath + houses$FullBath + houses$HalfBath
```

Para nuestro análisis, vamos a realizar un análisis previo exploratorio con determinadas columnas, para la relación con el precio de la casa y otras variables que consideramos que pueden ser interesantes :

Precio - Total viviendas.

Pasamos a ver una distribución de nº de casas por franja de precio:

```
options(scipen=1000)
ggplot(houses, aes(x = SalePrice, fill = ..count..)) +
  geom_histogram(binwidth = 15000) +
  ggtitle("Distribución del nº de casa / Precio (SalePrice)") +
  ylab("Nº de casas")
```



```
xlab("Precio de Venta") +
  theme(plot.title = element_text(hjust = 0.5))
```

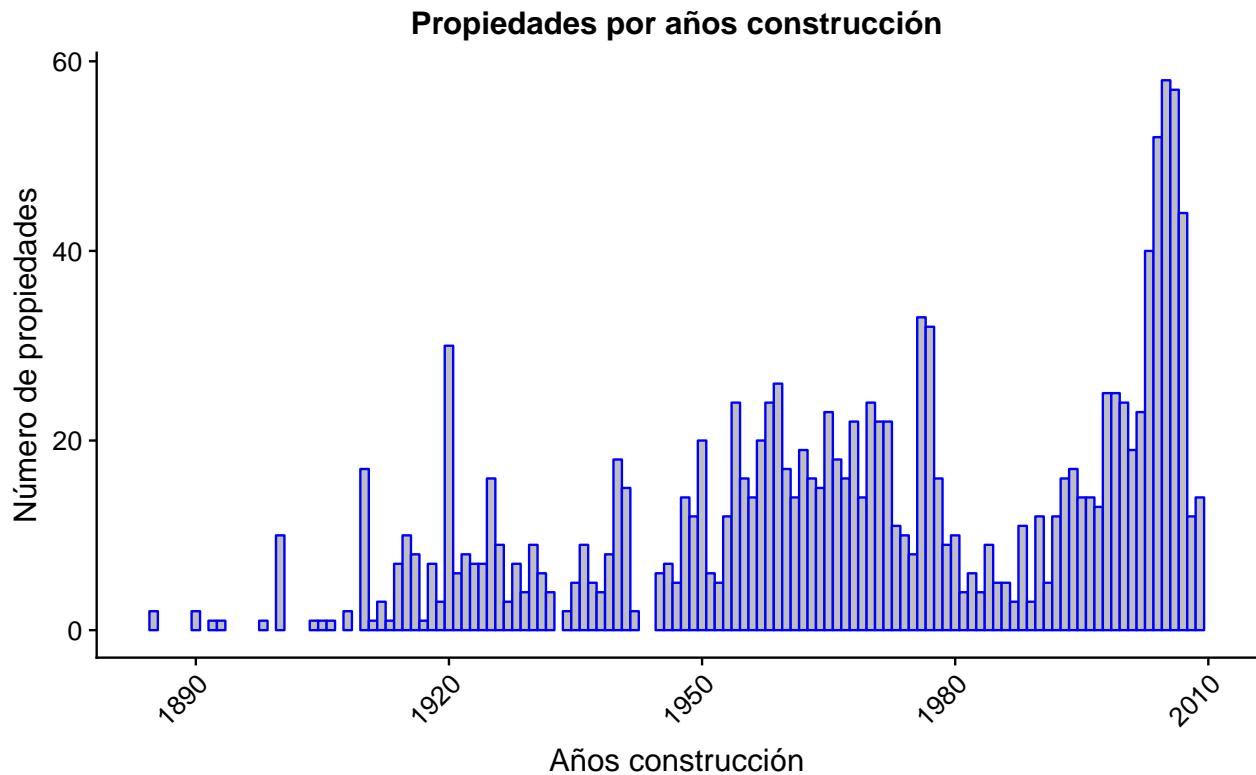
```
## NULL
```

El mayor volumen de propiedades se encuentra por debajo de los 200.000 USD. En el juego de datos original existían algunas propiedades que sobrepasaban los 600.000 USD que hemos eliminado para no distorsionar el precio medio.

Año construcción - Total de precio.

La mayoría de las casas están entre el año 1950 y 1970 o son posteriores al año 2000.

```
yr<-ggplot(houses, aes(x =YearBuilt, y = ..count..))+geom_bar(width = 1,
  colour="blue", fill="grey")
yr+xlab("Años")+ylab("Número de propiedades")+theme(axis.text.x = element_text(angle=45,
  hjust = 1))+ labs(title="Propiedades por años construcción",x="Años construcción")
```



Discretización

```

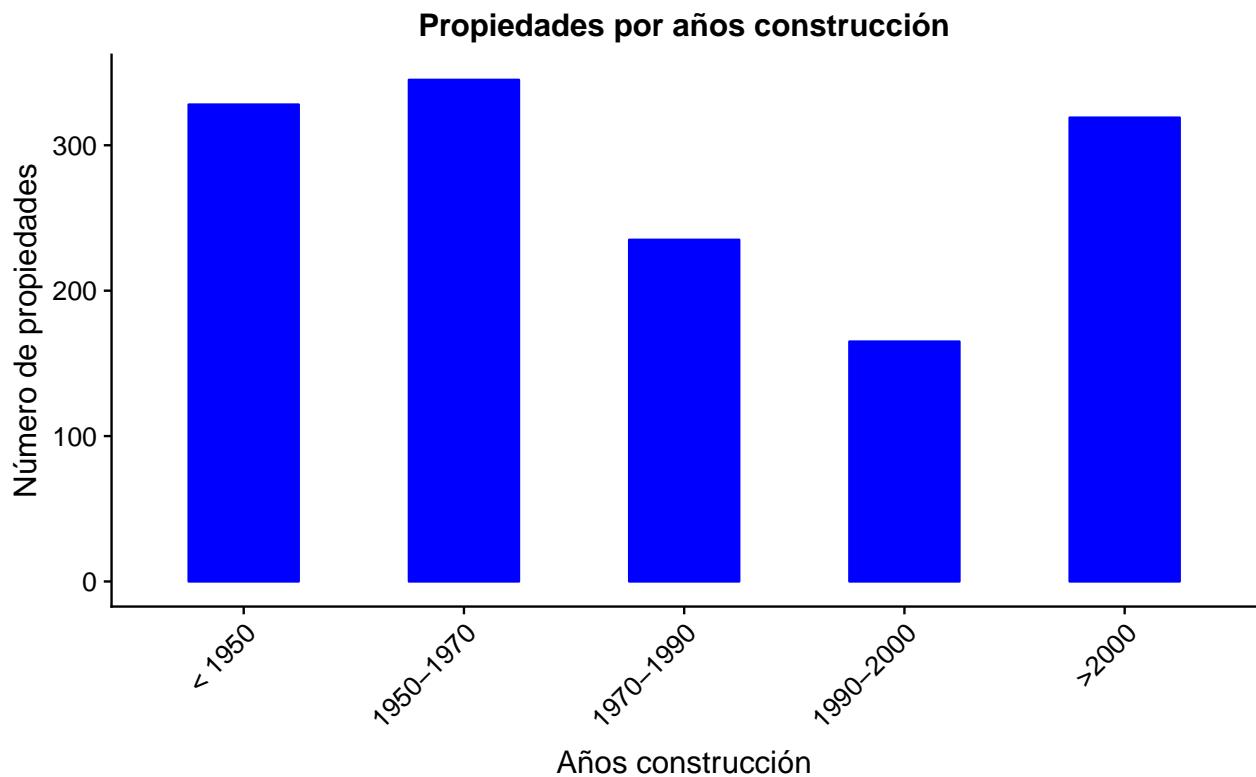
housesanio<-houses
housesanio$YearBuilt <- cut(housesanio$YearBuilt, breaks=c(1884,1950,1970,1990,2000,Inf),
                                labels=c('< 1950','1950-1970','1970-1990', '1990-2000', '>2000'))
table(housesanio$YearBuilt)

##
##      < 1950 1950-1970 1970-1990 1990-2000      >2000
##      328       345       235       165       319

granios <- ggplot(housesanio, aes(x =YearBuilt)) + geom_bar(width=0.5, colour="blue",
                                                               fill="blue", na.omit = "TRUE" )

## Warning: Ignoring unknown parameters: na.omit
granios+xlab("Años")+ylab("Número de propiedades") +theme(axis.text.x=element_text(angle=45,
                                         hjust = 1)) + labs(title="Propiedades por años construcción",x="Años construcción")

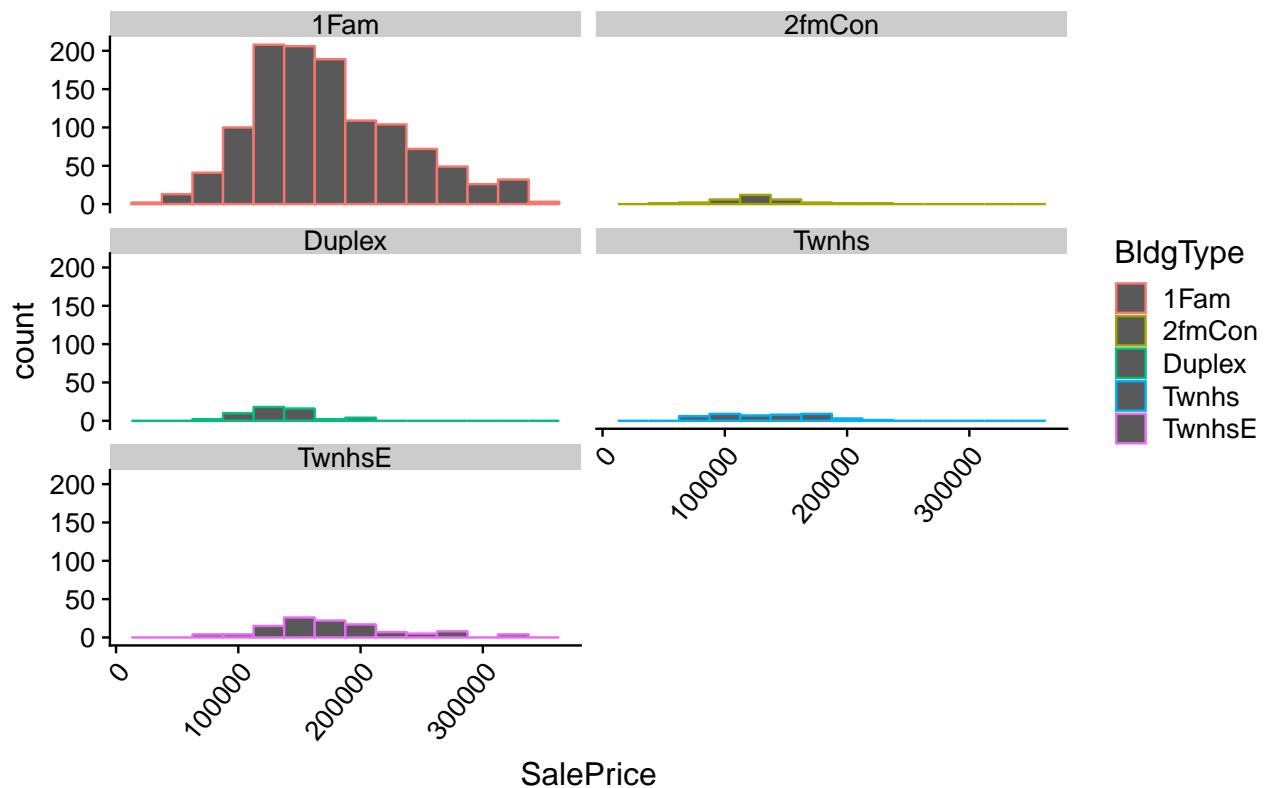
```



Precio - Tipo de Vivienda.

Grafico diferenciando el precio entre los distintos tipos de edificación de las casas (Familiares, Duplex, etc)

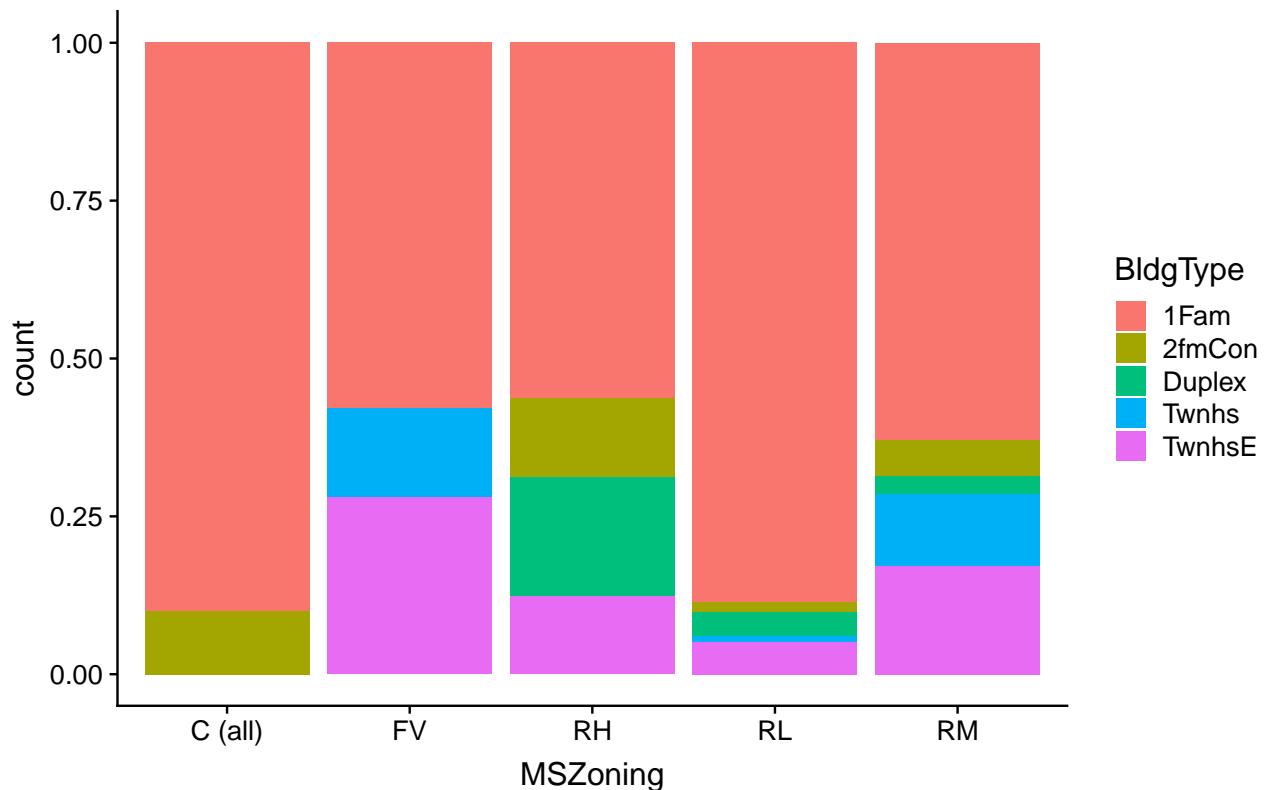
```
ggplot(houses, mapping = aes(x =SalePrice, colour = BldgType)) + geom_histogram(binwidth = 25000)+  
  facet_wrap(~BldgType,nrow=3, ncol=2)+theme(axis.text.x = element_text(angle=50, hjust = 1))
```



Se precisa que el tipo de vivienda con mayor volumen es la del tipo 1Fam cuya distribución de precios se encuentra en un rango inferior a los 200.000USD.

Tipo de Zona / Tipo de Vivienda /

```
ggplot(houses ,aes(MSZoning,fill=BldgType), heigth = 0.4)+geom_bar(position="fill")
```

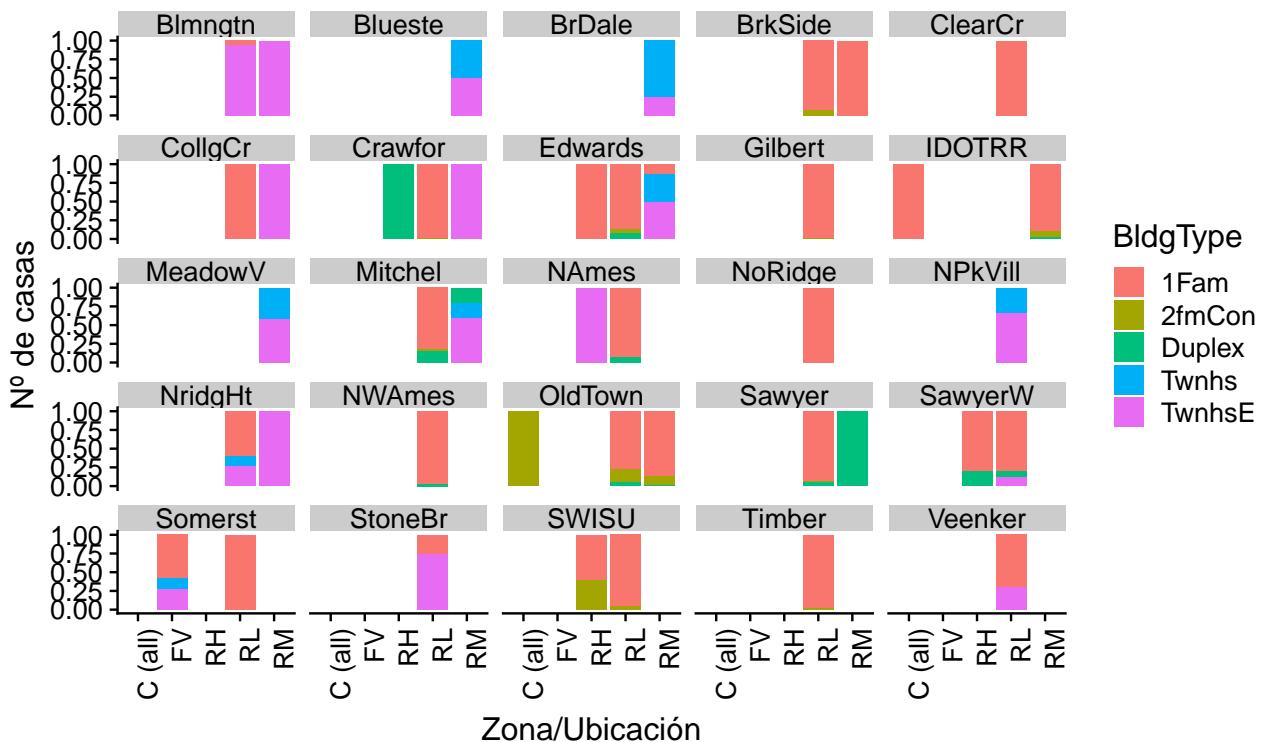


Se puede ver que existe diferencias en el tipo de vivienda dependiendo de la zona, aunque se aprecia que el volumen mayor es la de tipo 1Fam. Así en zona residencial de alta densidad la vivienda fuera de 1Fam (Unifamiliar) es de tipo duplex, sin embargo en zona residencial baja densidad es de tipo TwnhsE y 2fmCon. En el FV(Residencial en zonas rurales) es donde hay más volumen de TwnhsH

Tipo de Zona / Tipo de Vivienda y por los distintos barrios

```
ggplot(houses ,aes(x=MSZoning,fill=BldgType))+geom_bar(position="fill")+facet_wrap(~Neighborhood, nrow=1)
  theme(axis.text.x = element_text(angle=90, hjust = 1))+ 
  ggtitle("Grafico de n° de casas en los distintos barrios,por Tipo de Vivienda /Zona ") +
  ylab("Nº de casas") +
  xlab("Zona/Ubicación")
```

Grafico de nº de casas en los distintos barrios,por Tipo de Vivienda /Zona



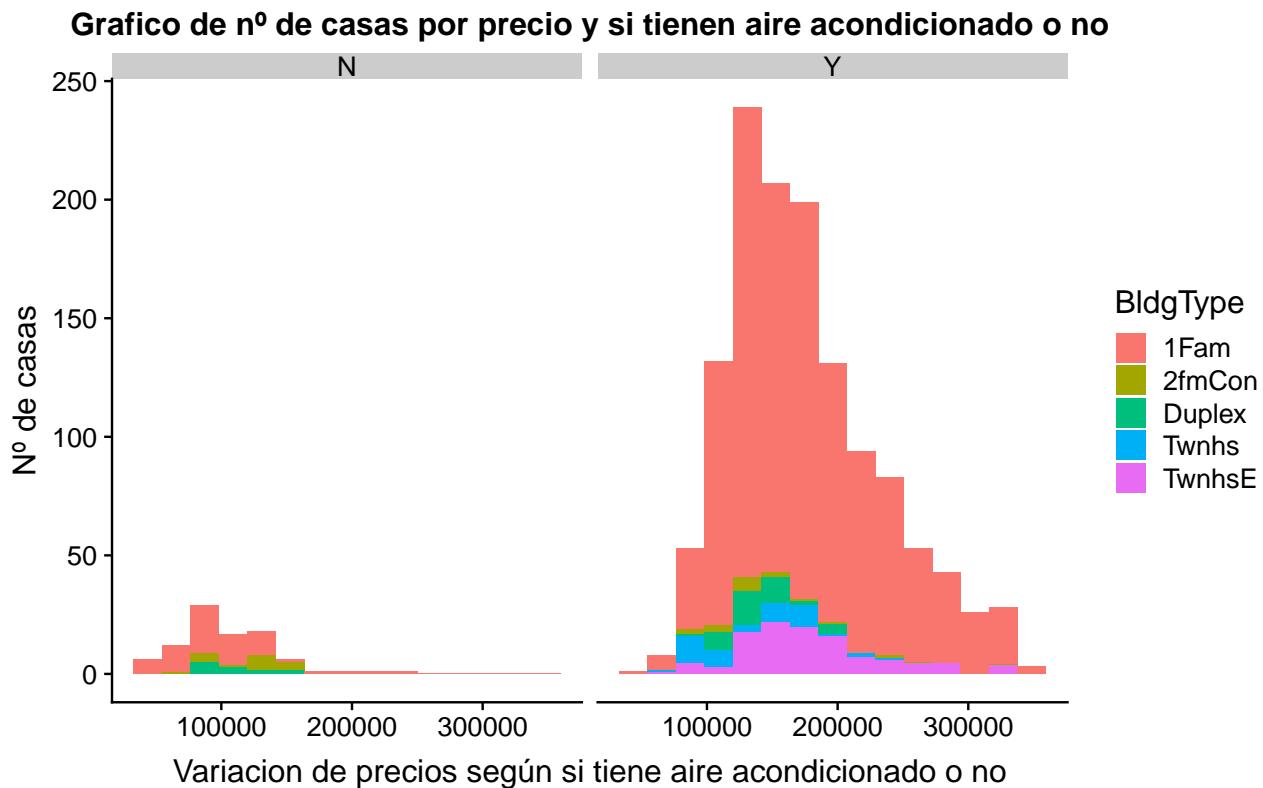
Si añadimos el tipo de barrio a la gráfica anterior se observa que la distribución del tipo de vivienda también varía por barrios, si bien domina en casi todos los barrios el tipo 1Fam, hay barrios que solo tienen tipo TwnhsE o Twnhs únicamente.

En las zonas residenciales RL /RM es donde existen más casas de tipo TwnhsE y en determinados barrios que apunta a que sean los que mayor precio tengan también.

La viviendas de tipo 2fmCon (que son las viviendas transformadas de una vivienda original para una familia en dos viviendas) de da en determinados barrios (OldTown, SWISU)

Estas variables, vemos que pueden ser indicativas de variaciones en el precio , y por tanto en los métodos de comparación estadísticos , utilizaremos estas variables para demostrar que son explicativas del precio.

```
ggplot(houses, aes(x = SalePrice, fill = BldgType), na.rm = TRUE) +
  geom_histogram(position="stack", bins = 15)+facet_wrap(~CentralAir)+
  ggtitle("Grafico de nº de casas por precio y si tienen aire acondicionado o no") +
  ylab("Nº de casas") +
  xlab("Variacion de precios según si tiene aire acondicionado o no")
```

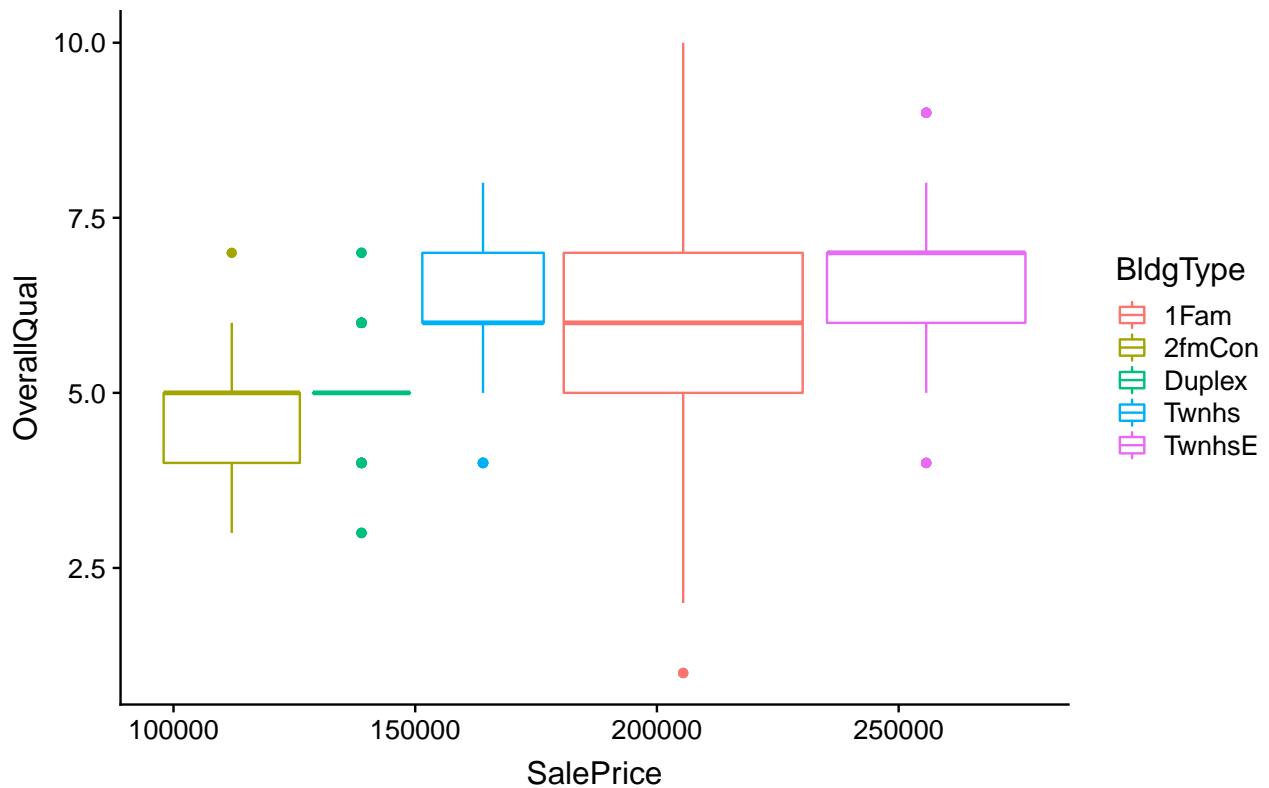


Se observa que la mayoría de las propiedades cuentan con aire acondicionado y que la distribución del precio , dependiendo de si tienen o no aire acondicindo es diferente. Ya se aprecia, que el precio de las que no tienen aire acondicinado están en un franja de precio menor. Más adelante , haremos un contraste de hipotesis para comprobar que la media del precio de las casas, es diferente si tiene aire o no .

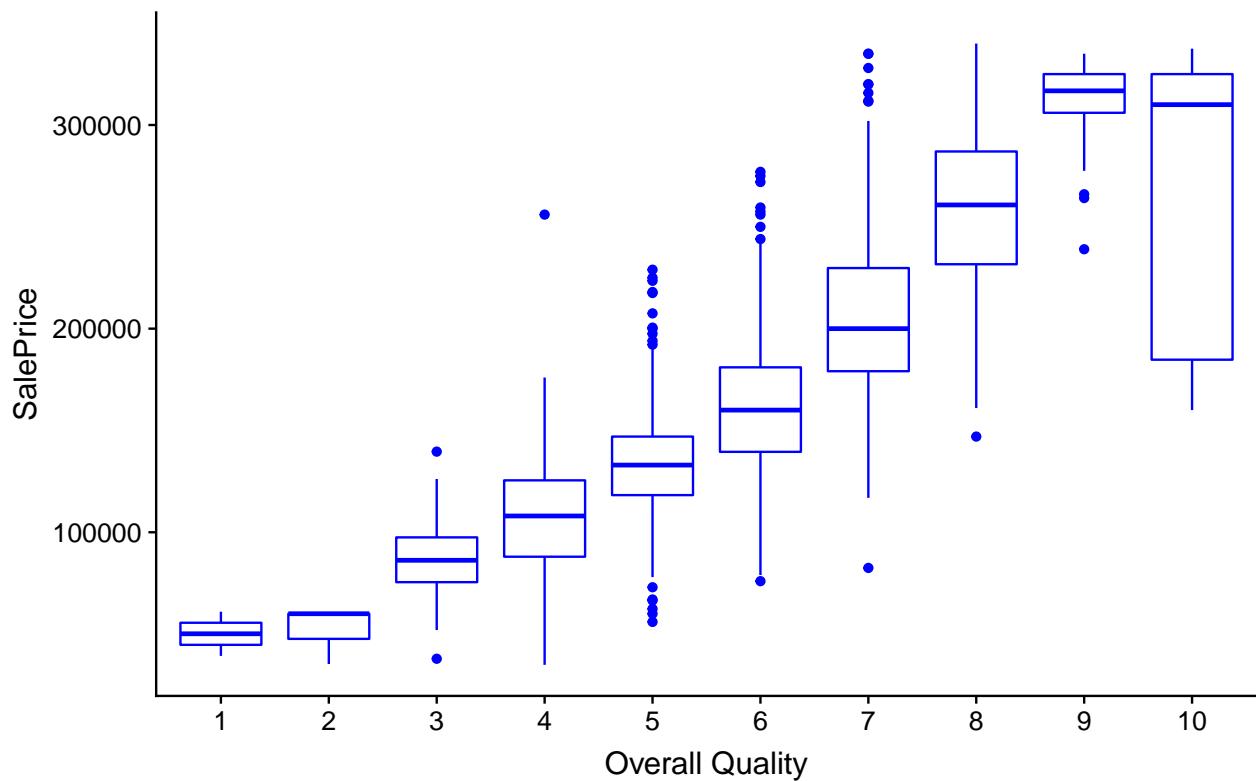
Precio /Calidad Vivienda

Se diferencian por colores los distintos tipos de vivienda.

```
ggplot(houses, aes(SalePrice, OverallQual , colour = BldgType )) + geom_boxplot()
```



```
ggplot(data=houses[!is.na(houses$SalePrice),], aes(x=factor(OverallQual), y=SalePrice)) +
  geom_boxplot(col='blue') + labs(x='Overall Quality') +
  scale_y_continuous(breaks= seq(0, 800000, by=100000))
```

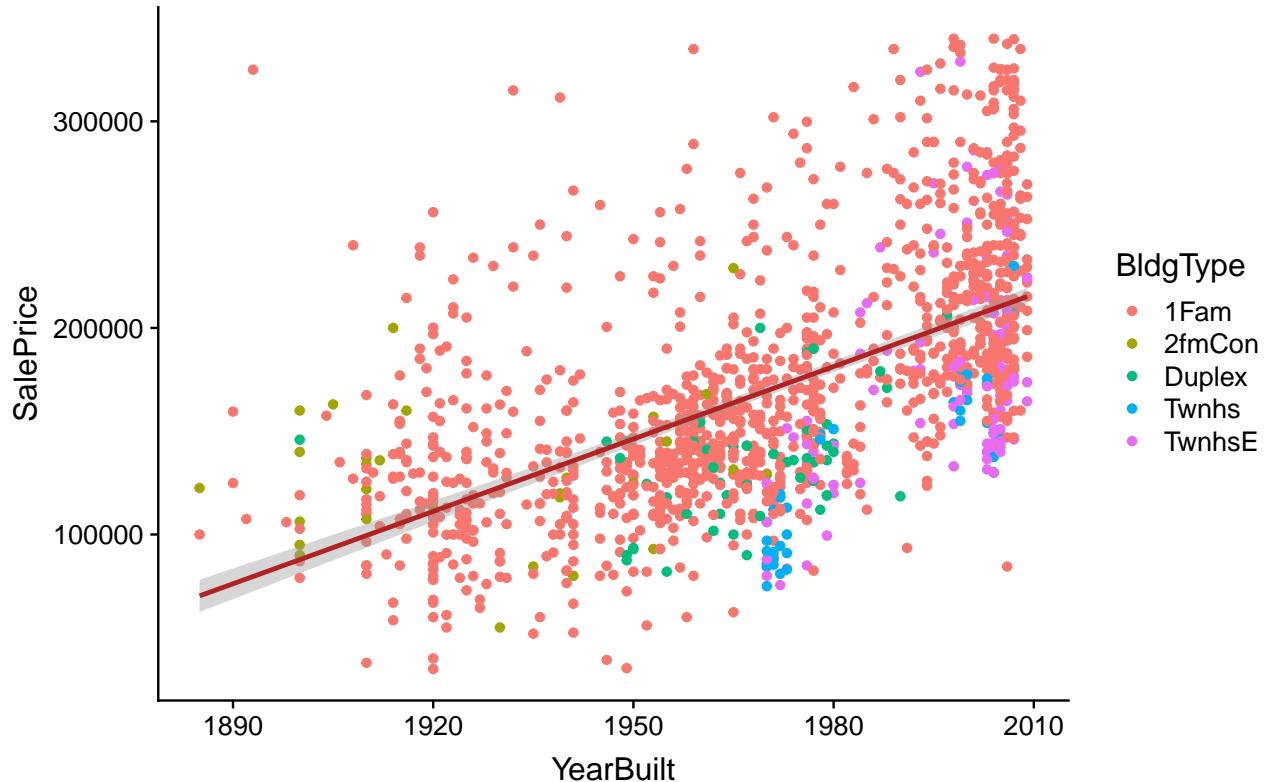


En cuanto a la relación entre calidad de la propiedad y el precio, este aumenta de forma directa con la calidad como se puede observar en la gráfica anterior.

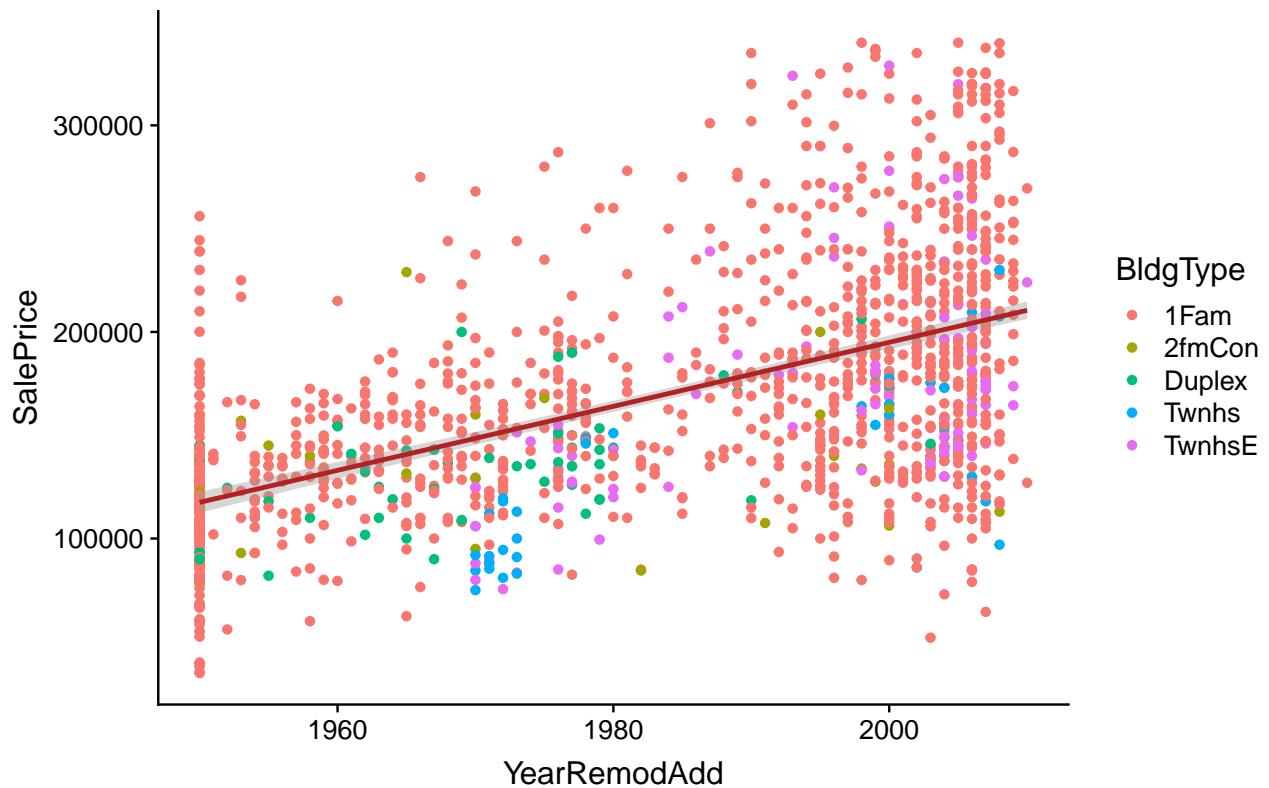
Precio /Año de Contrucción y Precio /Año de Remodelación

Se diferencian por colores los distintos tipos de vivienda.

```
ggplot(houses, aes(YearBuilt, SalePrice , colour = BldgType )) + geom_point() +  
  geom_smooth(color = "firebrick", method = 'lm')
```



```
ggplot(houses, aes(YearRemodAdd, SalePrice , colour = BldgType )) +  
  geom_point() + geom_smooth(color = "firebrick", method = 'lm')
```



Igualmente para la relación del año de construcción con el precio de la propiedad según el tipo de esta, se ve un ligero incremento con los años del precio, es decir para propiedades construidas en años recientes el precio es mayor para el mismo tipo de vivienda.

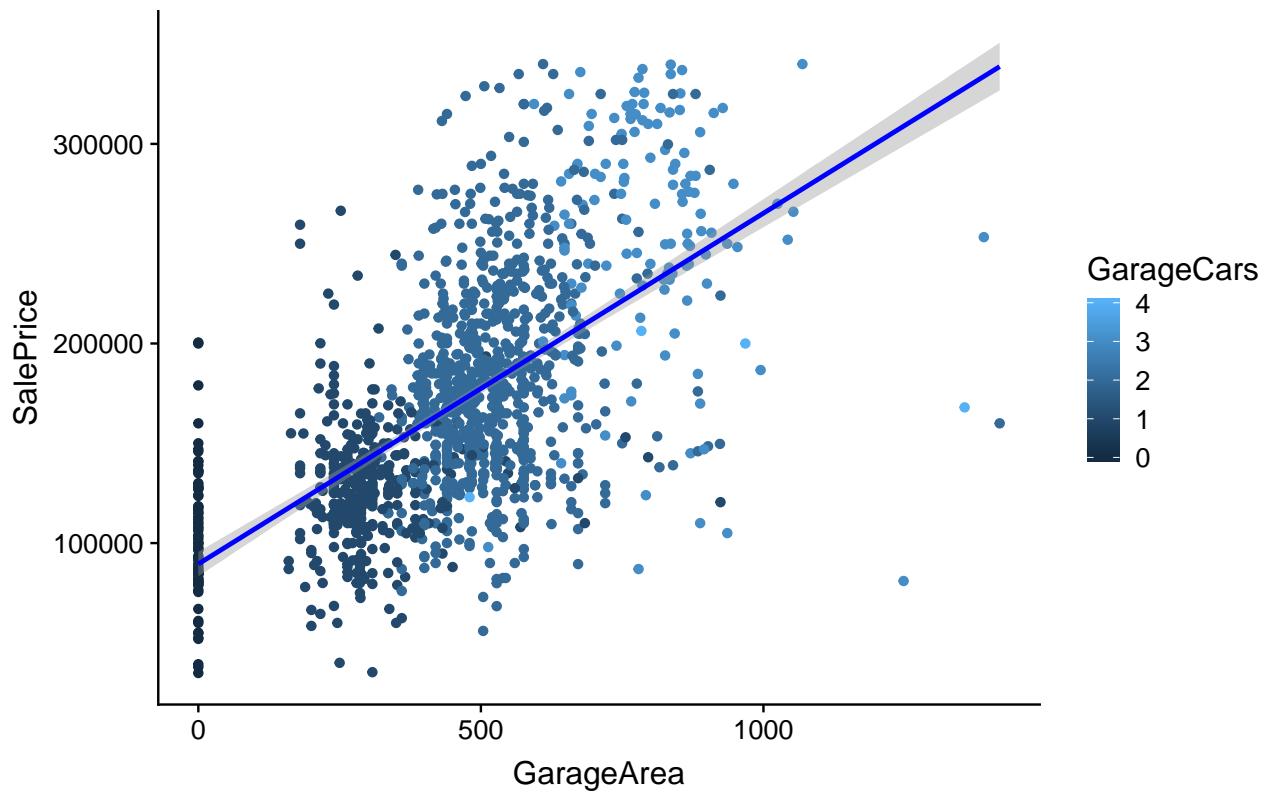
Y en la gráfica del año de remodelación , también se observa que la variación de precio aumenta si se ha remodelado recientemente.

Son variables explicativas del precio de la vivienda. Posteriormente , se comprobará si el precio de construcción y de remodelación tienen correlación (PCA) y se utilizarán en otros métodos (LM) etc.

Precio /Area del Garage

Se diferencian por colores el nº de plazas de garage

```
ggplot(houses, aes(GarageArea, SalePrice      , colour = GarageCars    )) + geom_point() +
  geom_smooth(color = "blue", method = 'lm')
```



La relación del precio con el área del garaje igualmente es incremental, aumentando este a medida que el número de plazas de aparcamiento de la vivienda es mayor.

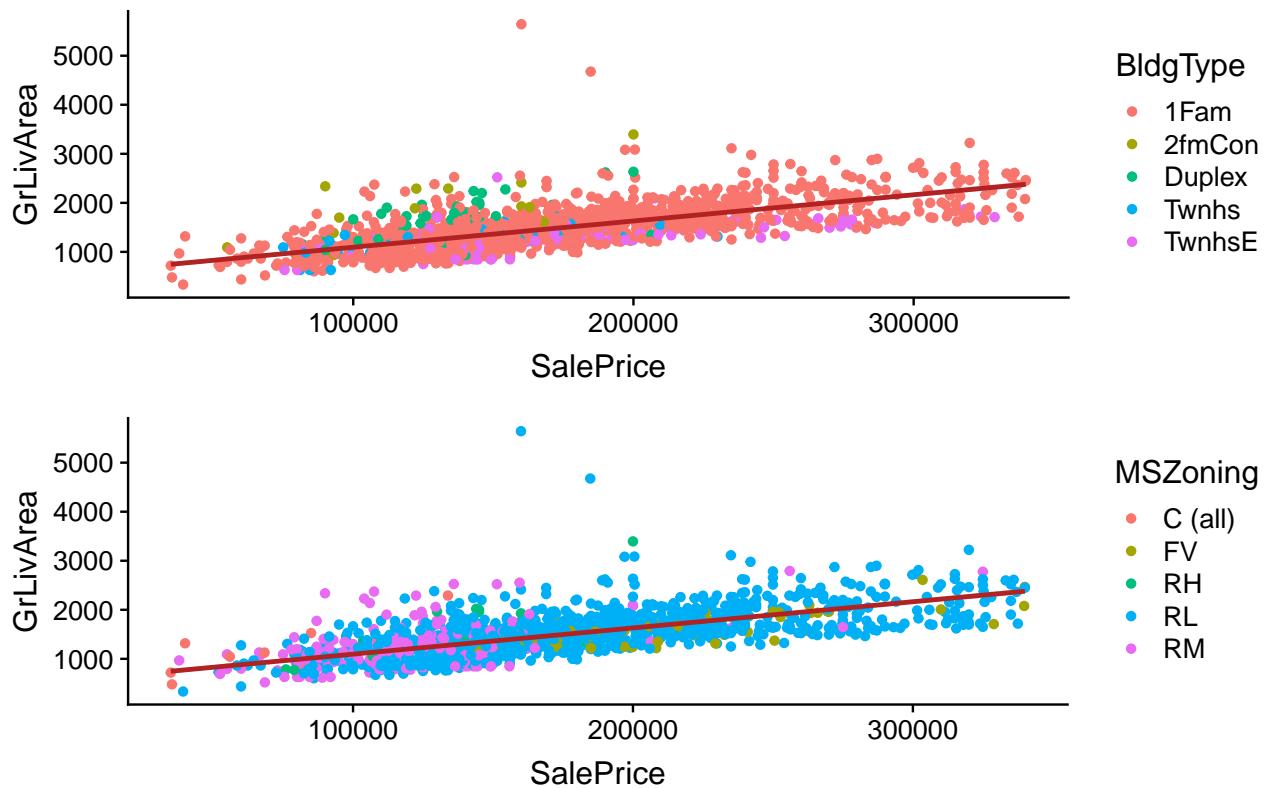
Precio /GrLiveArea (Área habitable)

```

g1<- ggplot(houses, aes(SalePrice, GrLivArea      , colour = BldgType )) + geom_point() +
  geom_smooth(color = "firebrick", method = 'lm')
g2<-ggplot(houses, aes(SalePrice, GrLivArea , colour = MSZoning )) + geom_point() +
  geom_smooth(color = "firebrick", method = 'lm')

grid.arrange(g1,g2)

```



Se observa claramente como según aumenta el n° de metros de area habitable, el precio se incrementa y que las casas de distintos tipos tambien se distribuyen en cluster diferenciados por lo que el precio medio variará entre ellas.

En el contraste de hipotesis planteado más adelante comprobaremos que efectivamente la diferencia entre medias se confirma.

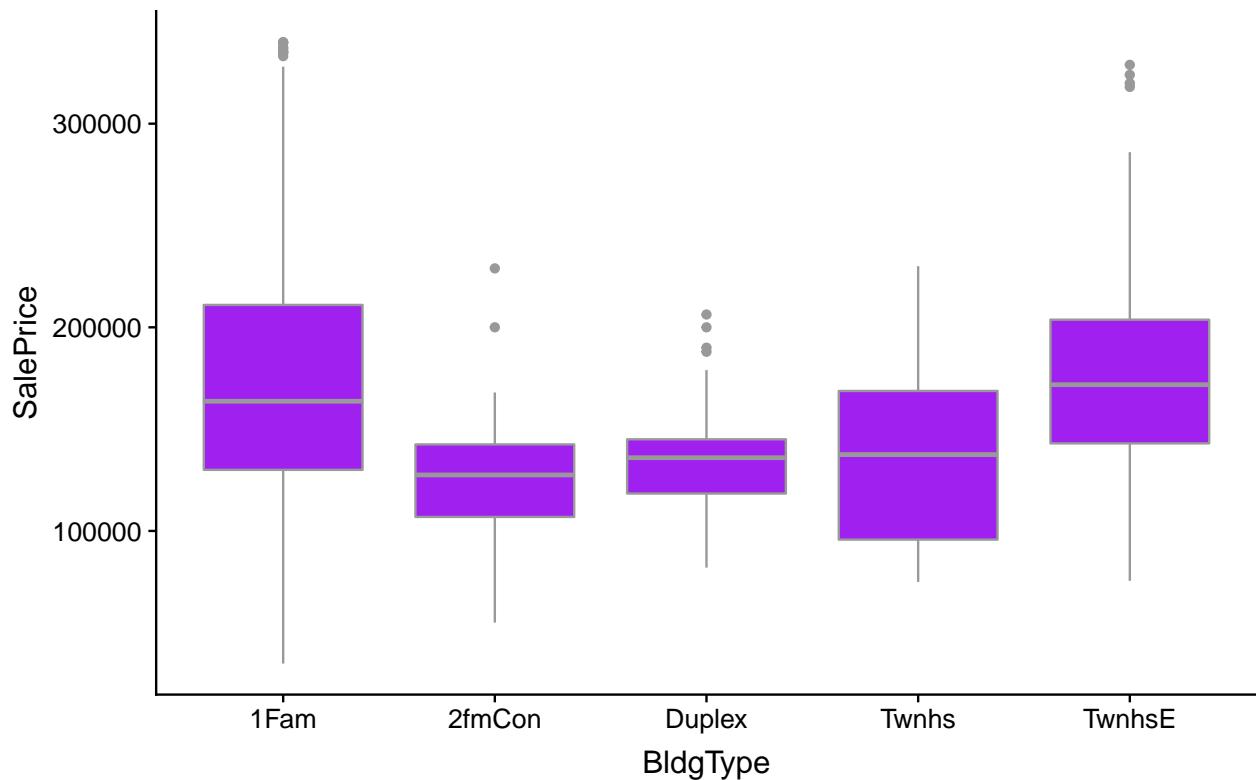
En la segunda gráfica, también se observa que el precio de la vivienda en función del n° de metros habitables, y diferenciando por zona, se ve que el precio medio entre zonas es diferente . Igualmente se demostrará en el contraste de hipotesis.

Otras distribuciones

```
# Distribucion de la variable SalePrice en función de Tipo de Edificio/propiedad

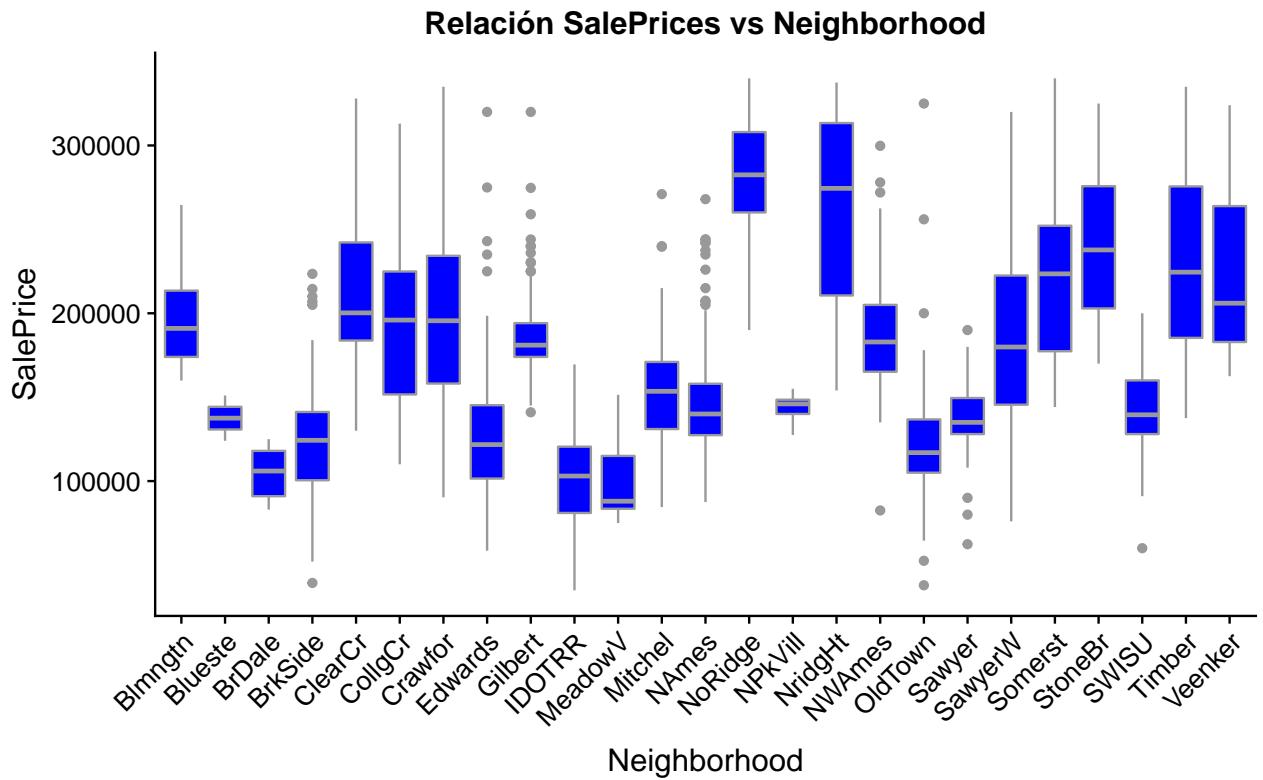
plotbyBldgType <- ggplot(houses,aes(BldgType,SalePrice))+
  geom_boxplot(fill="purple",color="gray60") +
  scale_fill_manual(values=c("green1","green3","gray60","lightblue","blue","yellow"))

grid.arrange(plotbyBldgType,ncol=1)
```



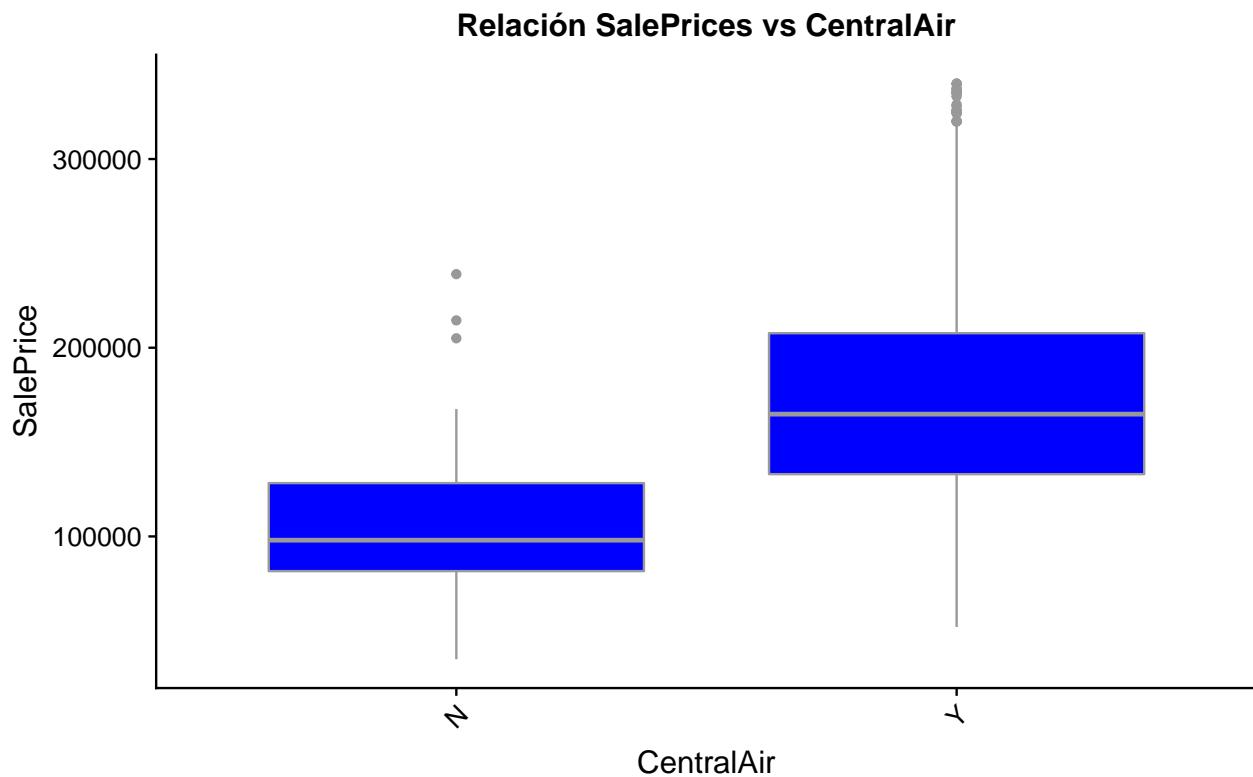
```
# Distribucion de la variable SalePrice vs Neighborhood
plotbyNeigh <- ggplot(houses,aes(Neighborhood, SalePrice))+
  geom_boxplot(fill="blue",color="gray60") +ggtitle("Relación SalePrices vs Neighborhood")+
  theme(axis.text.x = element_text(angle=45, hjust = 1))

grid.arrange(plotbyNeigh,ncol=1)
```



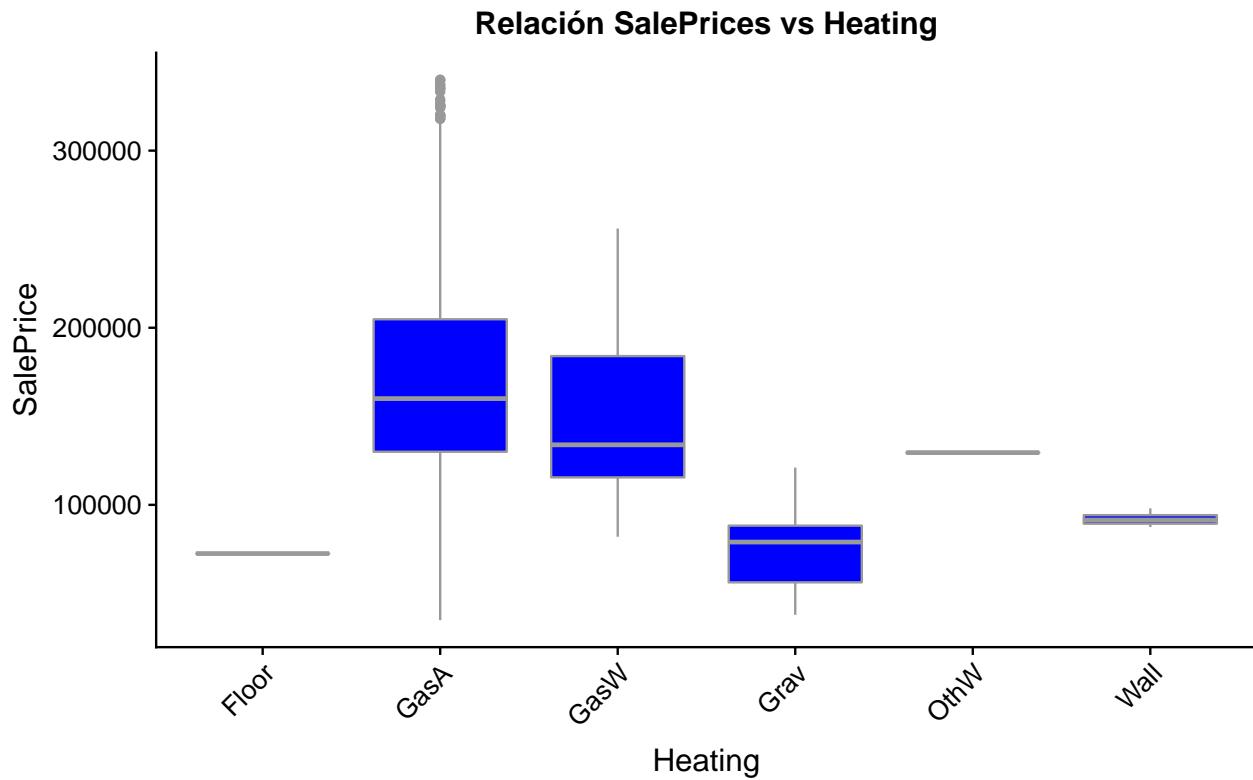
```
# Distribucion de la variable SalePrice vs CentralAir
plotbyCentralAir <- ggplot(houses,aes(CentralAir, SalePrice))+
  geom_boxplot(fill="blue",color="gray60") +ggtitle("Relación SalePrices vs CentralAir")+
  theme(axis.text.x = element_text(angle=45, hjust = 1))

grid.arrange(plotbyCentralAir,ncol=1)
```



```
# Distribucion de la variable SalePrice vs Heatingg
plotbyHeating <- ggplot(houses,aes(Heating, SalePrice))+
  geom_boxplot(fill="blue",color="gray60") +ggtitle("Relación SalePrices vs Heating")+
  theme(axis.text.x = element_text(angle=45, hjust = 1))

grid.arrange(plotbyHeating,ncol=1)
```



En la relación del tipo de vivienda con el precio, se aprecia que los mayores precios se encuentran en el tipo 1Fam si bien este es el más frecuente. En cuanto al vecindario se ve claramente que hay vecindarios que destacan por ser donde se encuentran las propiedades más caras. Y en cuanto al precio por Aire Acondicionado es distinto a los que no tienen, siendo mayor los que tienen aire acondicionado (suelen ser la mayoría) Y en cuanto al precio por tipo de calefacción también varía siendo mayor el precio los que tienen tipo GasA.

Comprobación de la normalidad y homogeneidad de la varianza.

```

alpha = 0.05
col.names = colnames(houses)
for (i in 1:ncol(houses)) {
  if (i == 1) cat("Variables no tienen distribución normal:\n")
  if (is.integer(houses[,i]) | is.numeric(houses[,i])) {
    p_val = ad.test(houses[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      # Format output
    if (i < ncol(houses) - 1) cat(", ")
    if (i %% 3 == 0) cat("\n")
  }
}
}

## Variables no tienen distribución normal:
## Id, MSSubClass, LotFrontage, LotArea, OverallQual,
## OverallCond, YearBuilt, YearRemodAdd,
## MasVnrArea,
## BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF,
```

```

## X1stFlrSF, X2ndFlrSF,
## LowQualFinSF, GrLivArea, BsmtFullBath,
## BsmtHalfBath, FullBath, HalfBath,
## BedroomAbvGr, KitchenAbvGr, TotRmsAbvGrd, Fireplaces,
## GarageYrBlt,
## GarageCars, GarageArea,
## WoodDeckSF, OpenPorchSF, EnclosedPorch,
## X3SsnPorch, ScreenPorch, PoolArea,
## MiscVal, MoSold, YrSold,
## SalePrice,
## TotalBaths

```

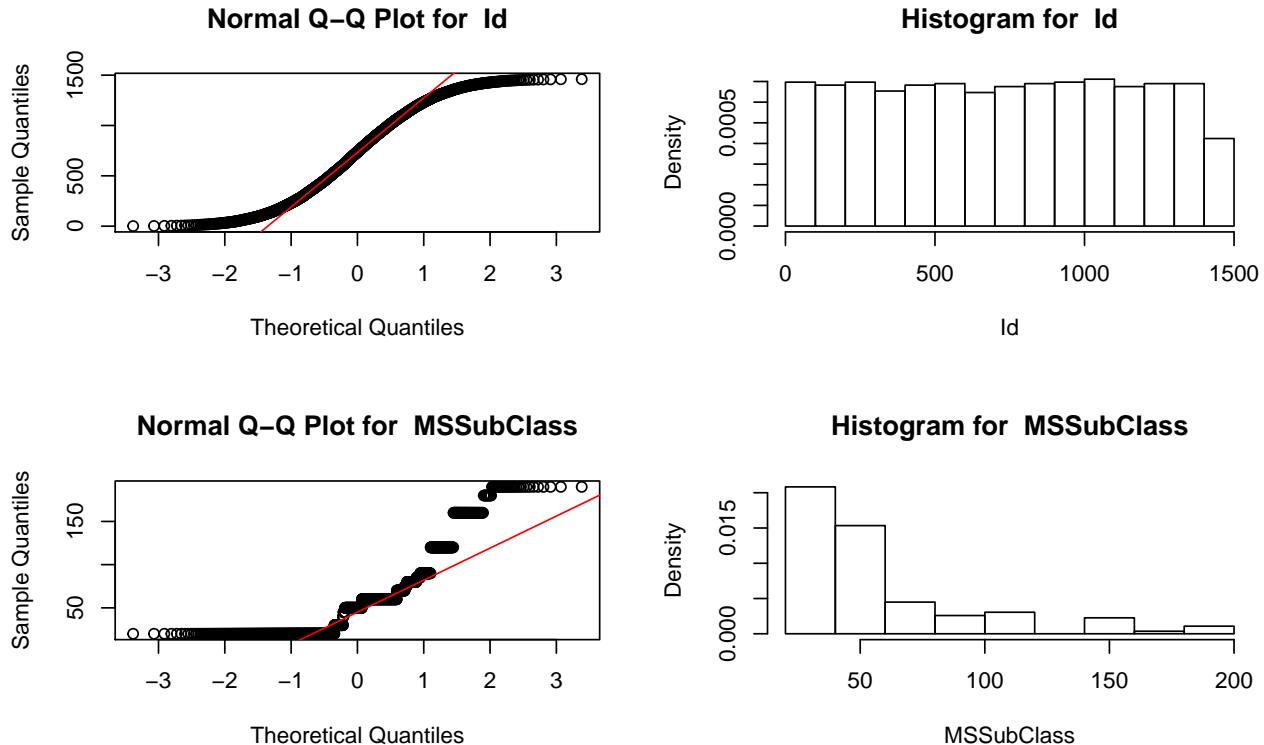
Se observan las variables que no tienen distribución normal.

Revisión normalización con graficas de quantile-quantile plot y el histograma.

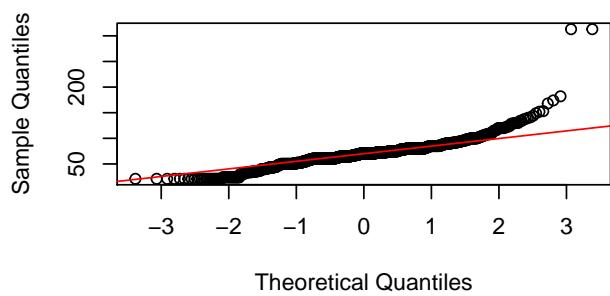
```

par(mfrow=c(2,2))
for(i in 1:ncol(houses)){
  if (is.numeric(houses[,i])){
    qqnorm(houses[,i],main = paste("Normal Q-Q Plot for ",colnames(houses)[i]))
    qqline(houses[,i],col="red")
    hist(houses[,i],
    main=paste("Histogram for ", colnames(houses)[i]),
    xlab=colnames(houses)[i], freq = FALSE)
  }
}

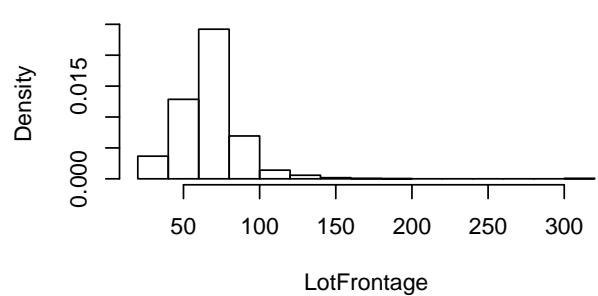
```



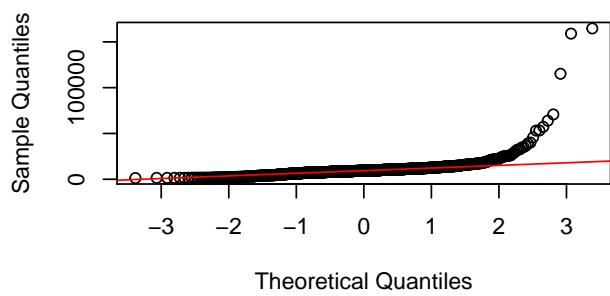
Normal Q-Q Plot for LotFrontage



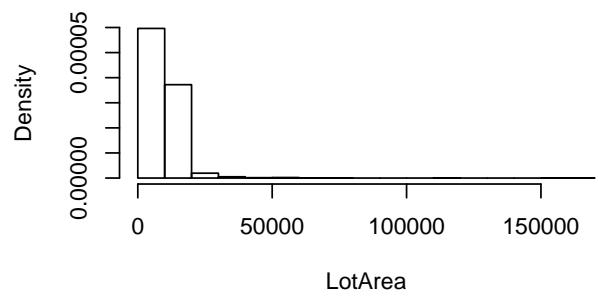
Histogram for LotFrontage



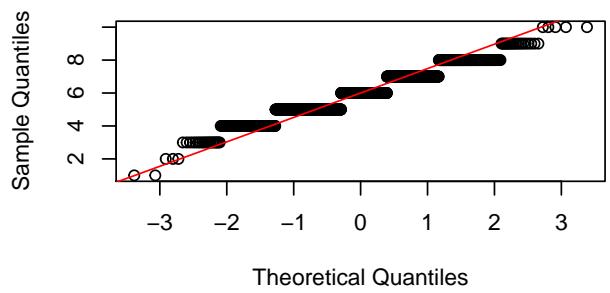
Normal Q-Q Plot for LotArea



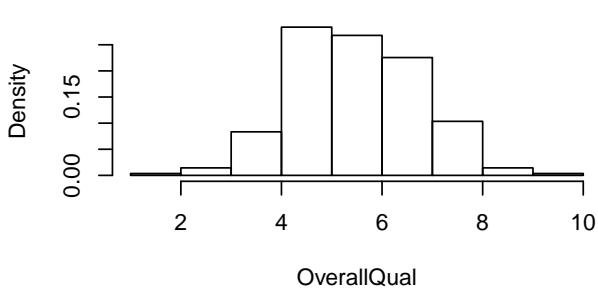
Histogram for LotArea



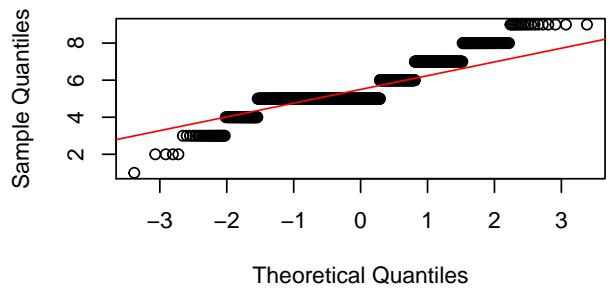
Normal Q-Q Plot for OverallQual



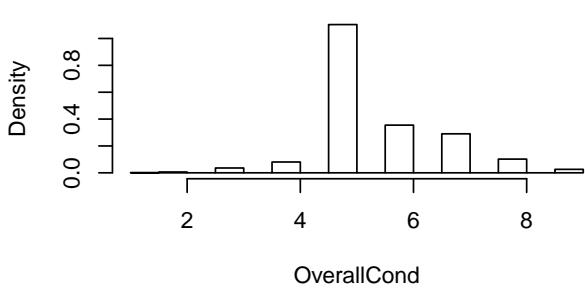
Histogram for OverallQual



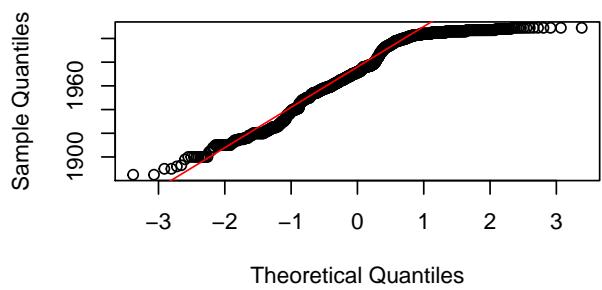
Normal Q-Q Plot for OverallCond



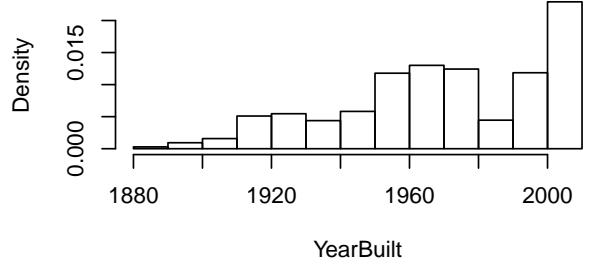
Histogram for OverallCond



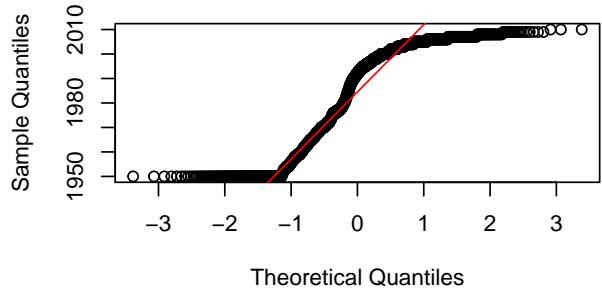
Normal Q-Q Plot for YearBuilt



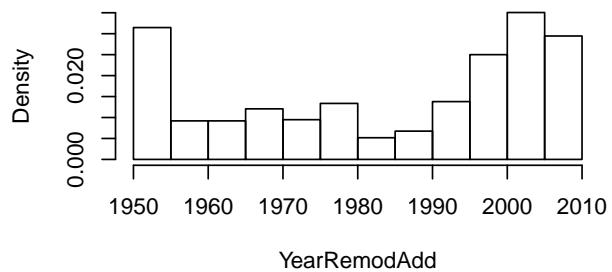
Histogram for YearBuilt



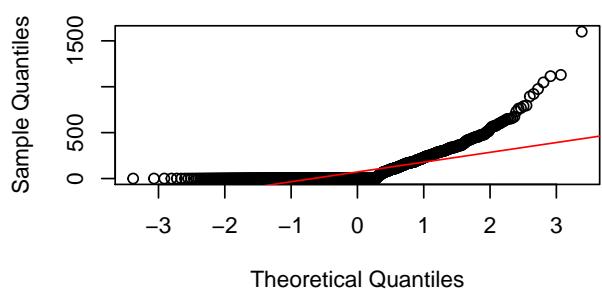
Normal Q-Q Plot for YearRemodAdd



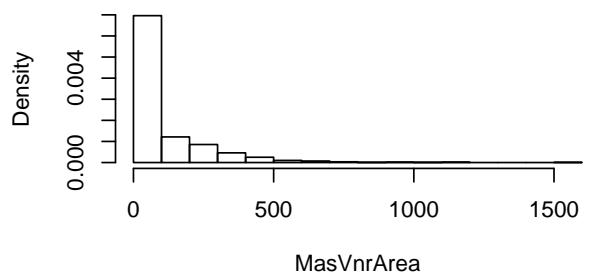
Histogram for YearRemodAdd



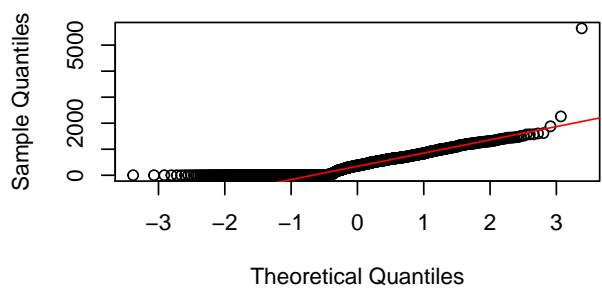
Normal Q-Q Plot for MasVnrArea



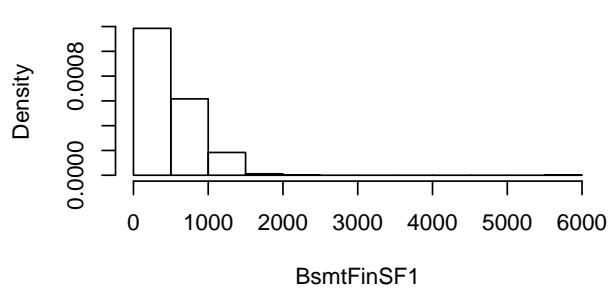
Histogram for MasVnrArea

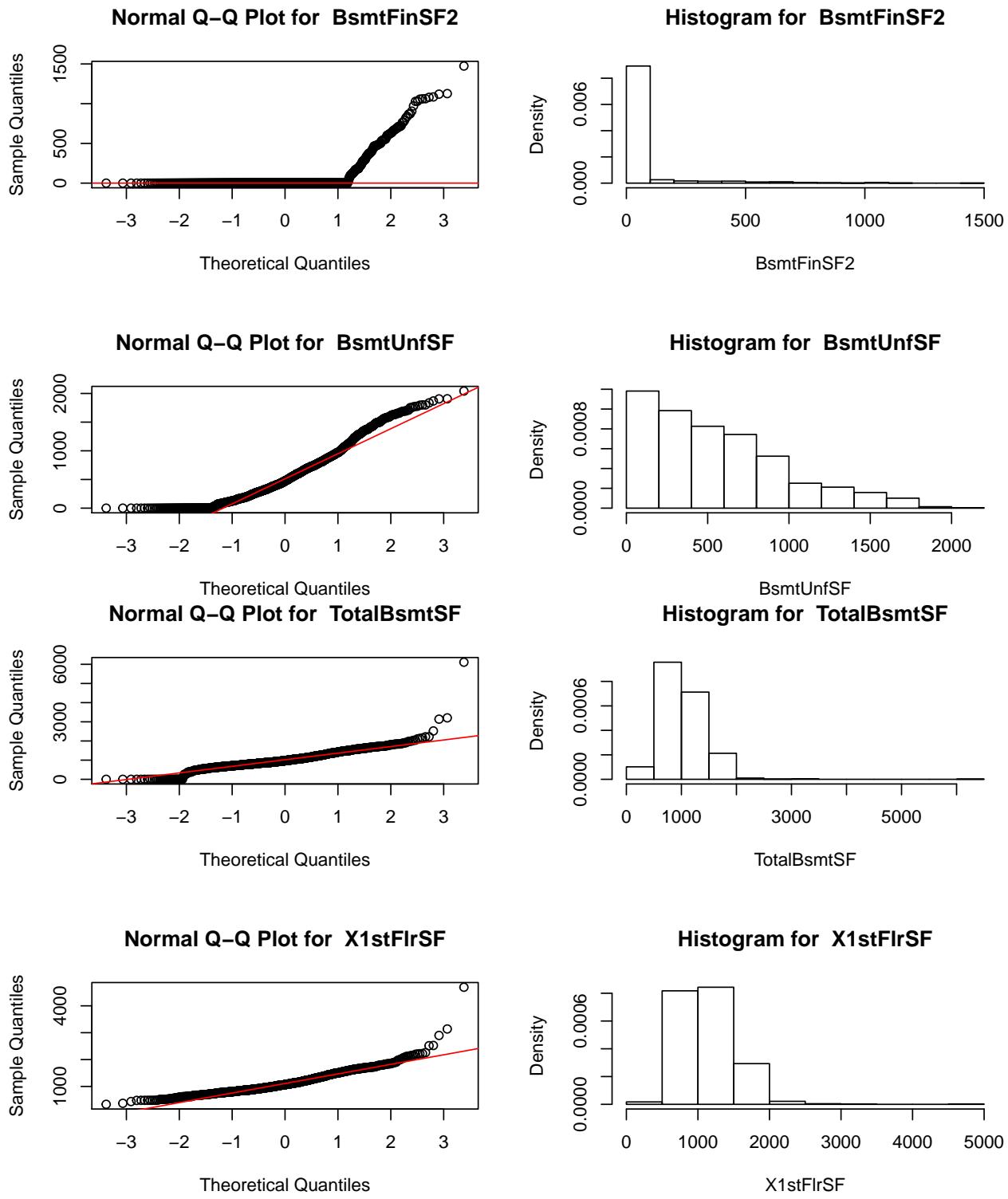


Normal Q-Q Plot for BsmtFinSF1

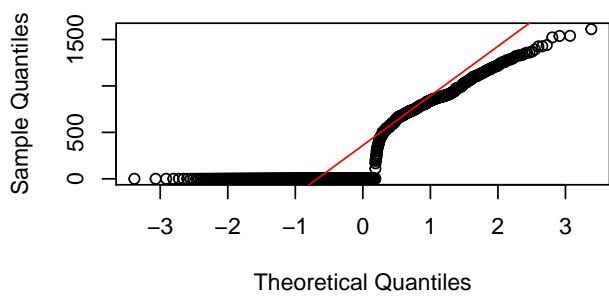


Histogram for BsmtFinSF1

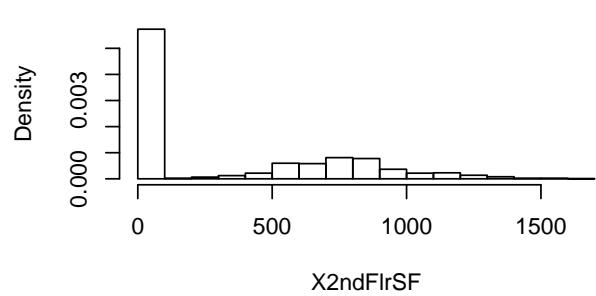




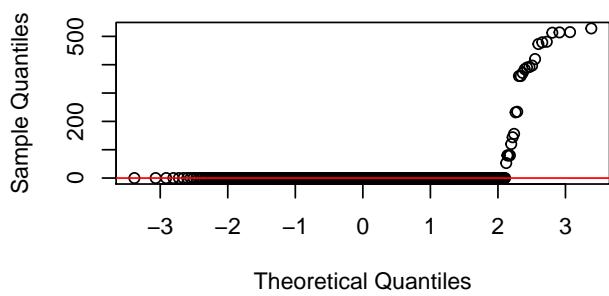
Normal Q-Q Plot for X2ndFlrSF



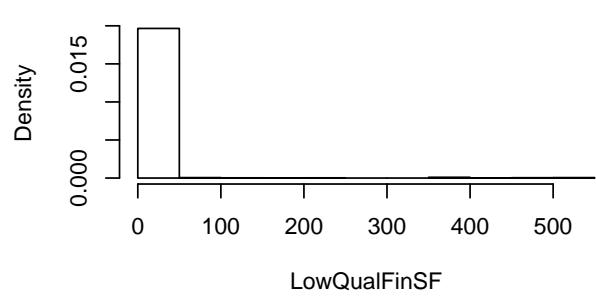
Histogram for X2ndFlrSF



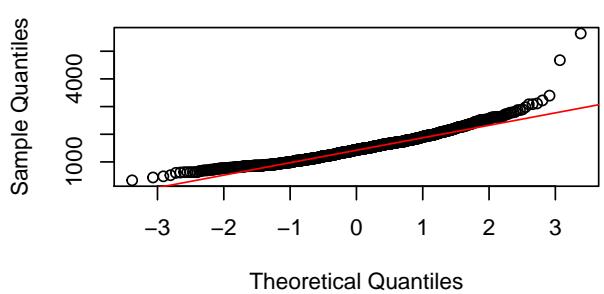
Normal Q-Q Plot for LowQualFinSF



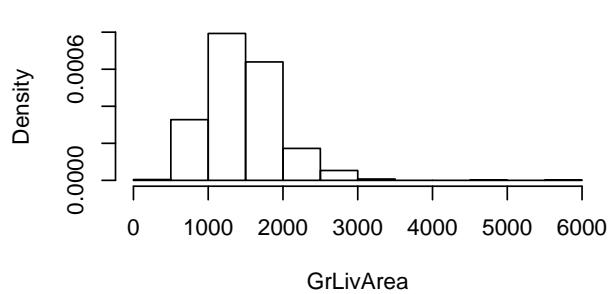
Histogram for LowQualFinSF



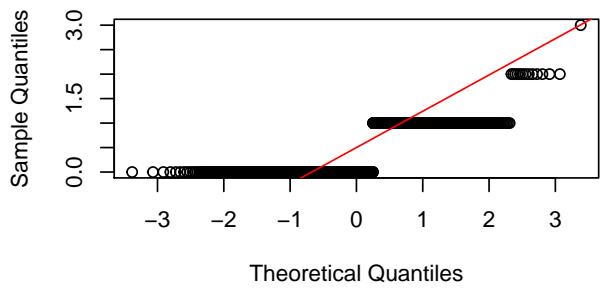
Normal Q-Q Plot for GrLivArea



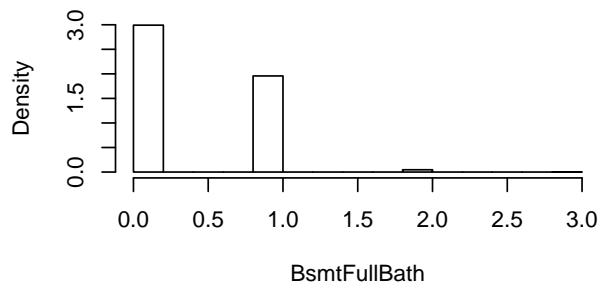
**LowQualFinSF
Histogram for GrLivArea**



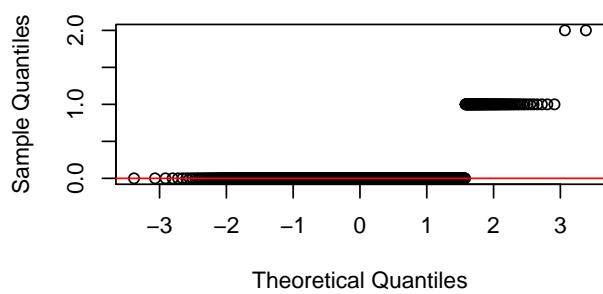
Normal Q-Q Plot for BsmtFullBath



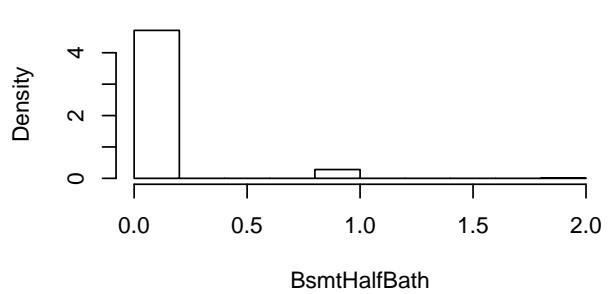
Histogram for BsmtFullBath



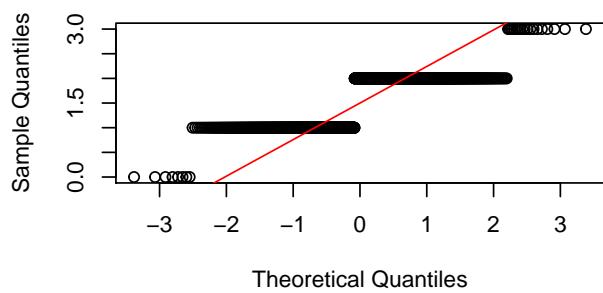
Normal Q-Q Plot for BsmtHalfBath



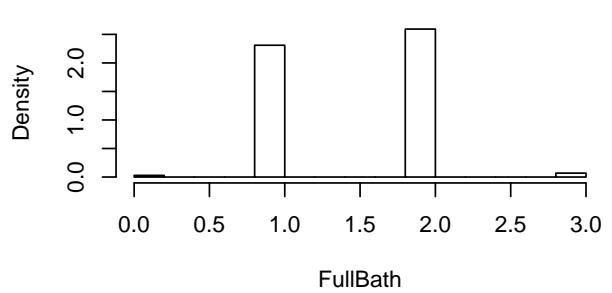
Histogram for BsmtHalfBath



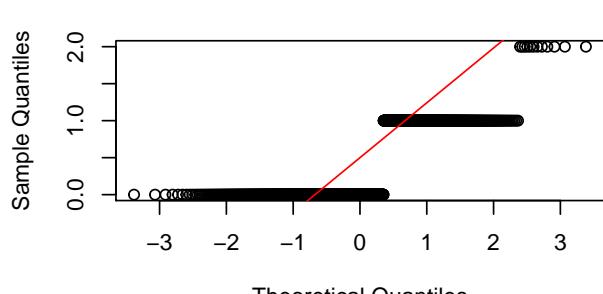
Normal Q-Q Plot for FullBath



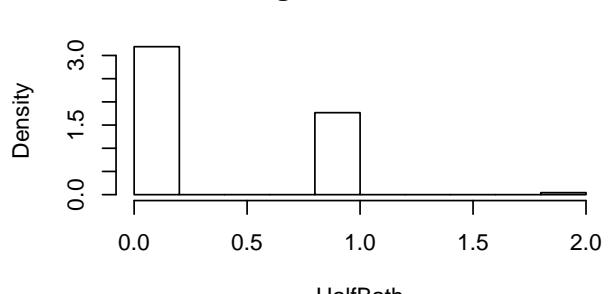
Histogram for FullBath



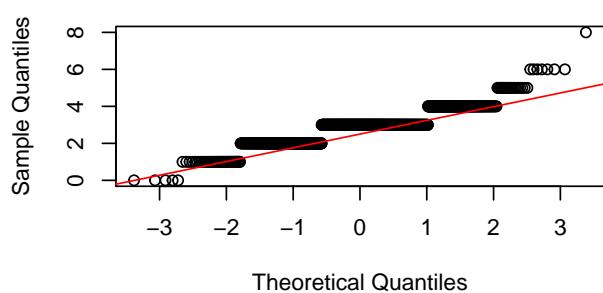
Normal Q-Q Plot for HalfBath



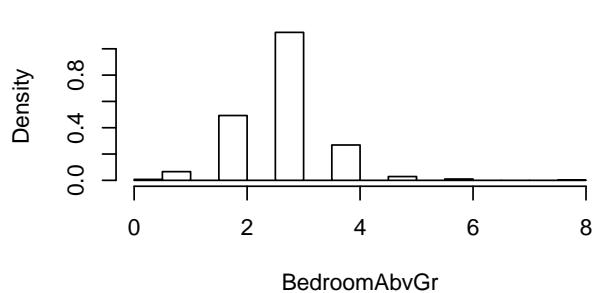
Histogram for HalfBath



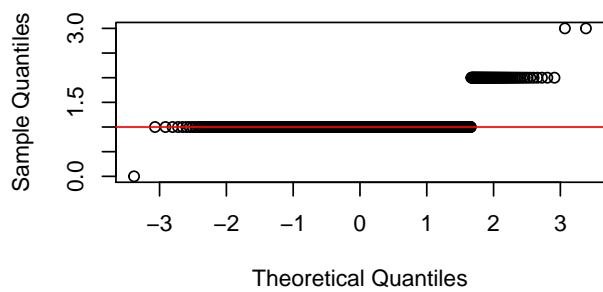
Normal Q-Q Plot for BedroomAbvGr



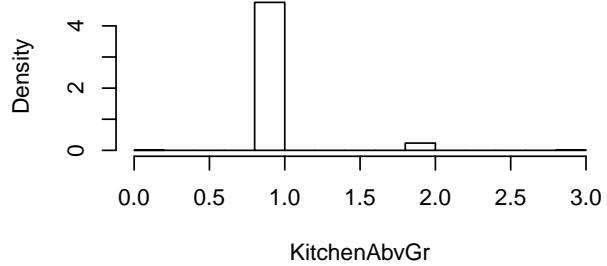
Histogram for BedroomAbvGr



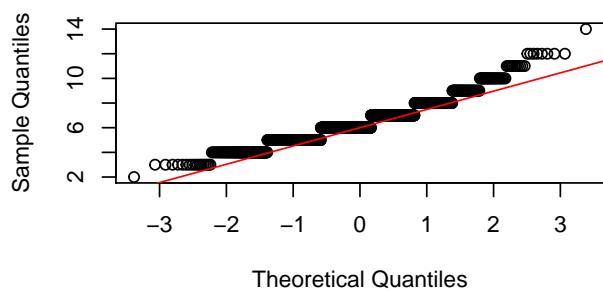
Normal Q-Q Plot for KitchenAbvGr



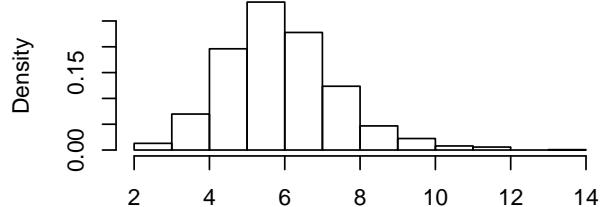
Histogram for KitchenAbvGr



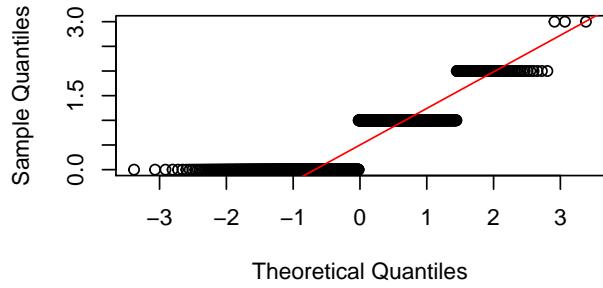
Normal Q-Q Plot for TotRmsAbvGrd



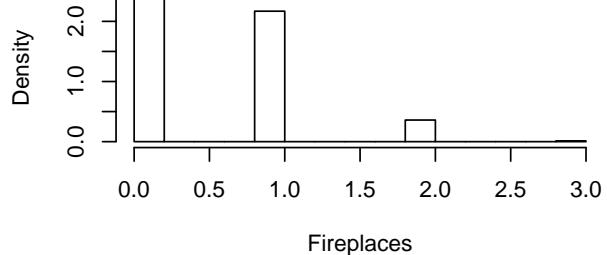
Histogram for TotRmsAbvGrd



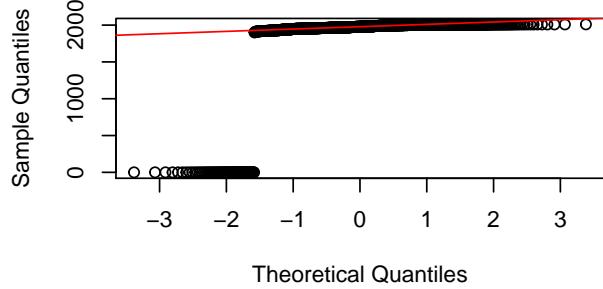
Normal Q-Q Plot for Fireplaces



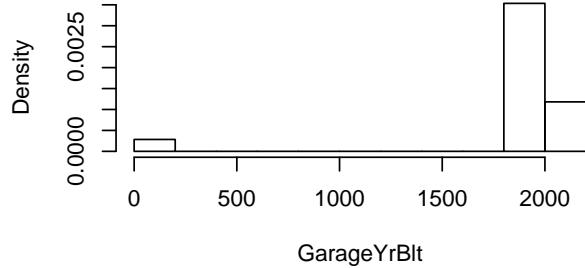
Histogram for Fireplaces



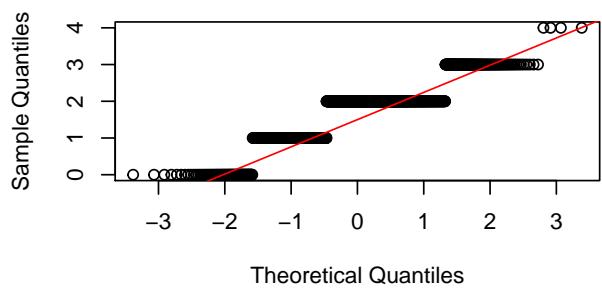
Normal Q-Q Plot for GarageYrBlt



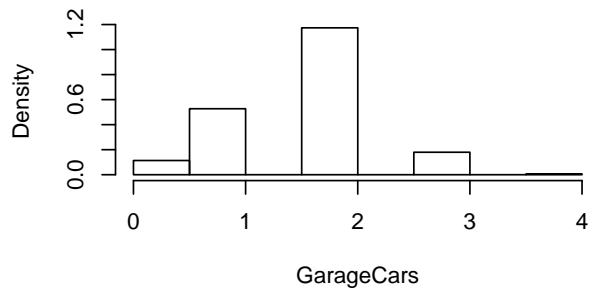
Histogram for GarageYrBlt



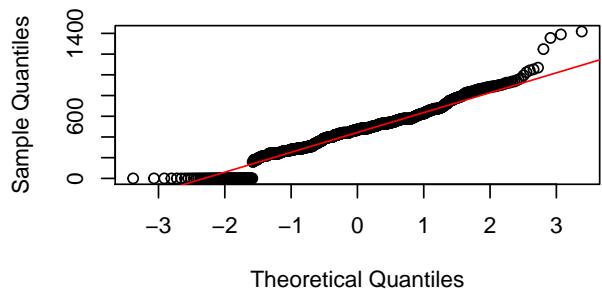
Normal Q-Q Plot for GarageCars



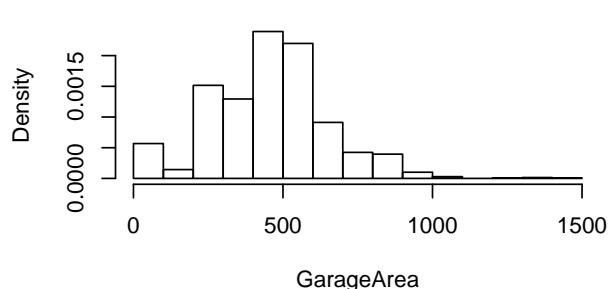
Histogram for GarageCars



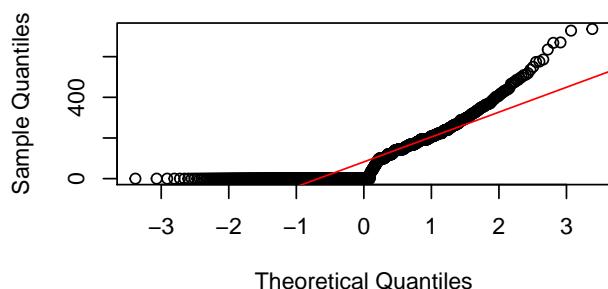
Normal Q-Q Plot for GarageArea



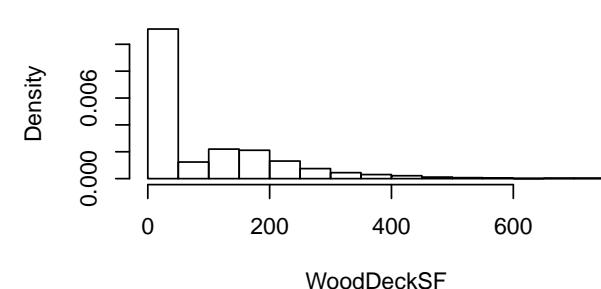
Histogram for GarageArea



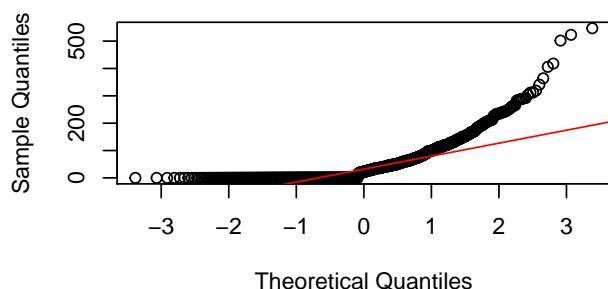
Normal Q-Q Plot for WoodDeckSF



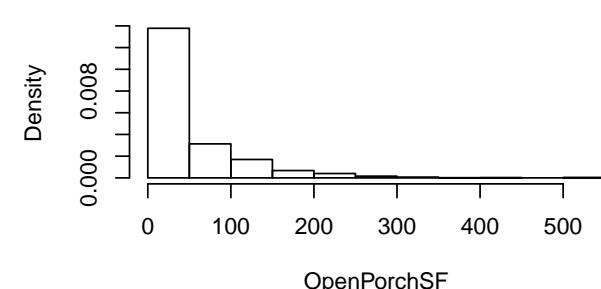
Histogram for WoodDeckSF



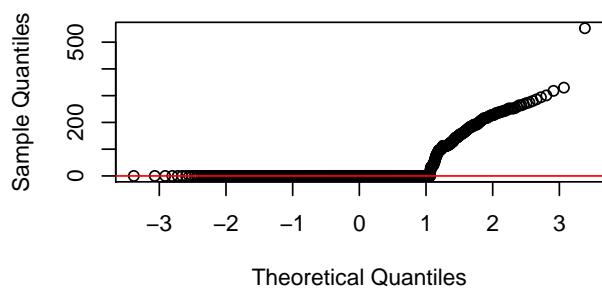
Normal Q-Q Plot for OpenPorchSF



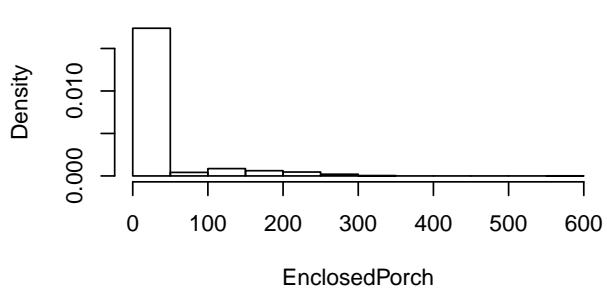
Histogram for OpenPorchSF



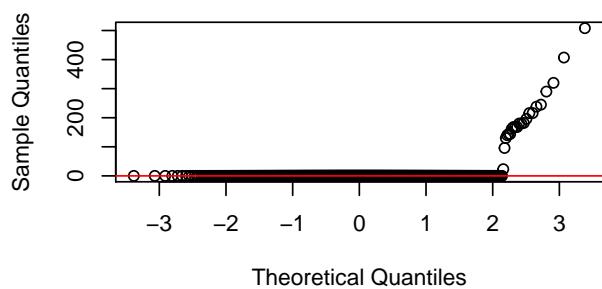
Normal Q-Q Plot for EnclosedPorch



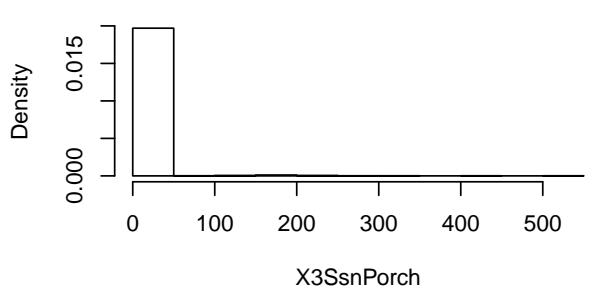
Histogram for EnclosedPorch



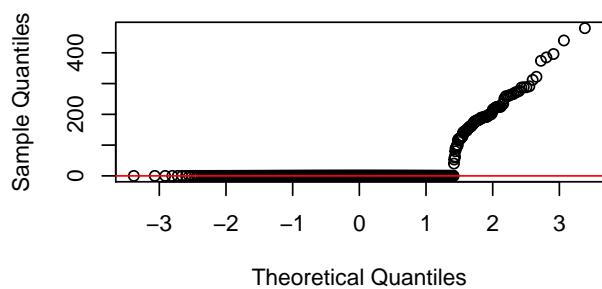
Normal Q-Q Plot for X3SsnPorch



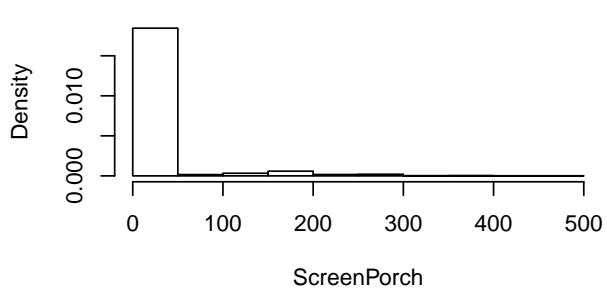
Histogram for X3SsnPorch



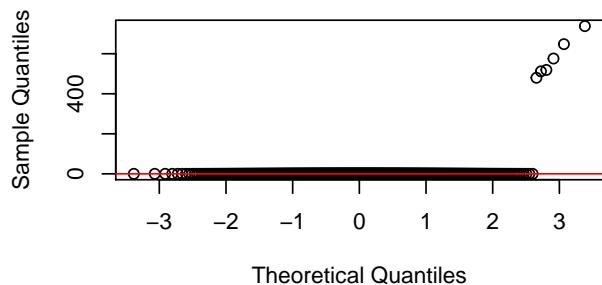
Normal Q-Q Plot for ScreenPorch



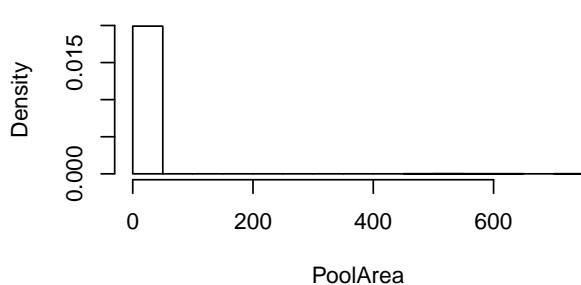
**X3SsnPorch
Histogram for ScreenPorch**

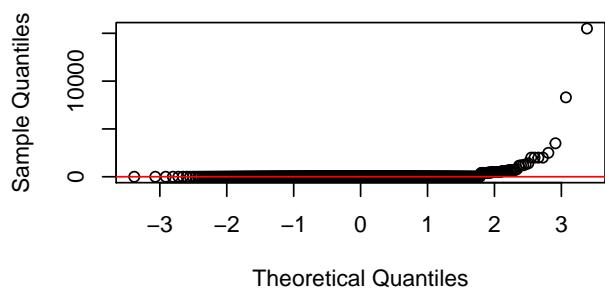
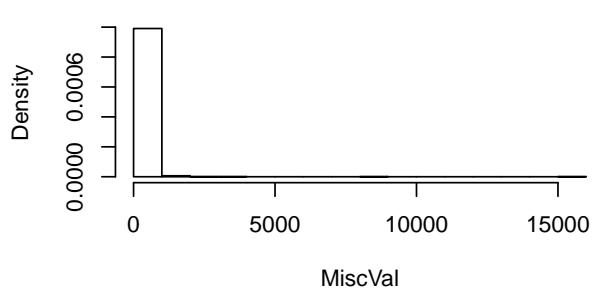
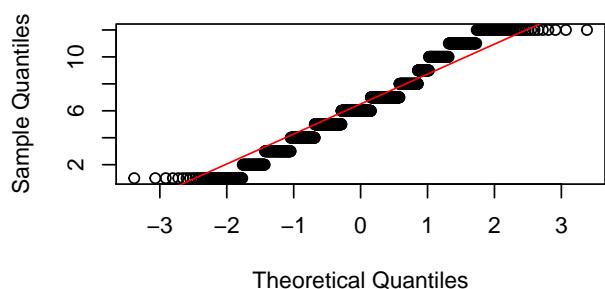
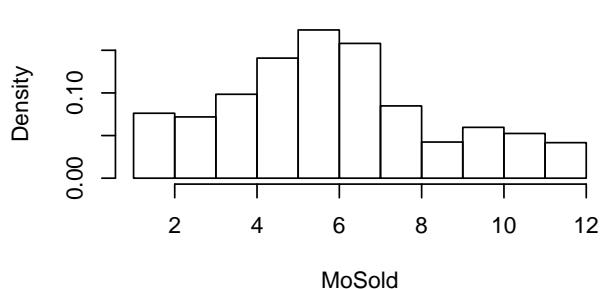
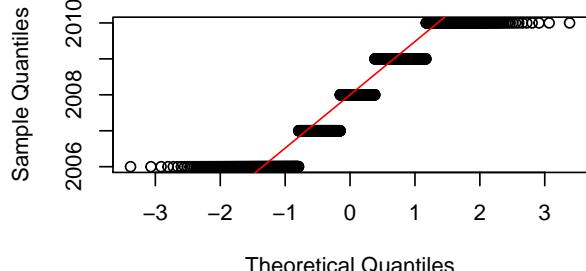
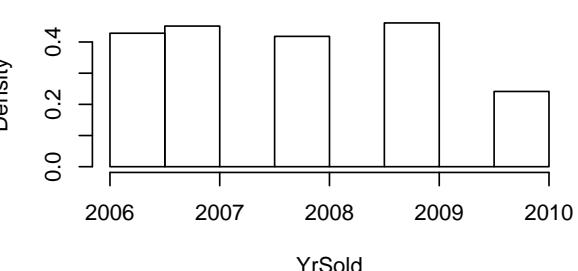
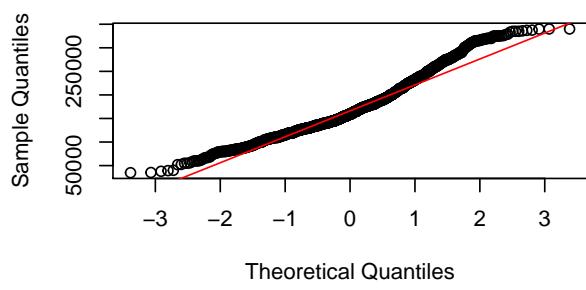
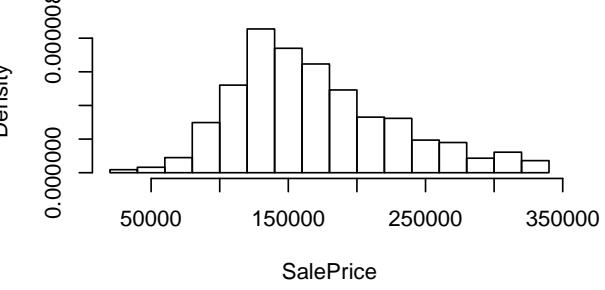


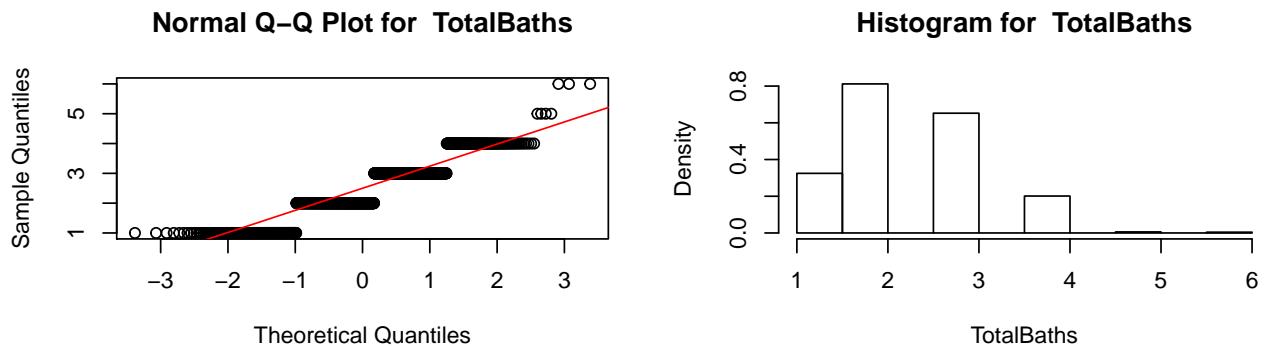
Normal Q-Q Plot for PoolArea



Histogram for PoolArea



Normal Q-Q Plot for MiscVal**Histogram for MiscVal****Normal Q-Q Plot for MoSold****Histogram for MoSold****Normal Q-Q Plot for YrSold****Histogram for YrSold****Normal Q-Q Plot for SalePrice****Histogram for SalePrice**



Revisando los resultados algunas variables podrían ser normalizadas, lo estudiamos a continuación con el test de Shapiro.

Test Shapiro

Sapiro-test - Shapiro-Wilk se usa para contrastar si un conjunto de datos siguen una distribución normal o no La hipótesis son : * Nula H_0 : los datos provienen de una distribución normal * Alternativa H_1 : los datos no provienen de una distribución normal

```
# Vamos a ver si son los datos del SalePrice normales o no.
shapiro.test(houses$SalePrice)
```

```
##
##  Shapiro-Wilk normality test
##
## data: houses$SalePrice
## W = 0.96282, p-value < 0.0000000000000022
```

En este caso el shapiro.test nos indica que el p-value bastante pequeño , menor al 0,05 por lo que tenemos que descartar la hipótesis nula a favor de la alternativa. Por tanto no es normal la distribución del precio de venta.

```
# Vamos a ver si son los datos del SalePrice normales o no.
shapiro.test(houses$OverallQual)
```

```
##
##  Shapiro-Wilk normality test
##
## data: houses$OverallQual
## W = 0.94363, p-value < 0.0000000000000022
# Vamos a ver si son los datos del SalePrice normales o no.
shapiro.test(houses$GrLivArea)
```

```
##
##  Shapiro-Wilk normality test
##
## data: houses$GrLivArea
## W = 0.94096, p-value < 0.0000000000000022
# Vamos a ver si son los datos del SalePrice normales o no.
shapiro.test(houses$YearBuilt)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data: houses$YearBuilt
## W = 0.93246, p-value < 0.00000000000000022
```

Podemos apreciar que todas ellas están por debajo del coeficiente 0.05, es decir se rechaza la hipótesis nula pudiendo confirmar que no están normalizadas.

Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Matriz de correlación entre variables

Vamos a realizar el análisis de correlación, para lo se va a utilizar la función pairs.panels(), de cara a visualizar los dispersiogramas con una línea de ajuste, histogramas para ver la distribución de las variables, y además los coeficientes de correlación.

```
# pasamos a numéricos algunos factor
```

```
houses$CentralAirnum <- as.numeric(factor(houses$CentralAir,
                                             levels = c("N", "Y"),
                                             labels = c(0,1) ,ordered = TRUE))
houses$GarageQualnum <- as.numeric(factor(houses$GarageQual,
                                             levels = c("Ex", "Gd", "TA","Fa", "Po", "Not apply"),
                                             labels = c(5,4,3,2,1,0) ,ordered = TRUE))
houses$GarageCondnum <- as.numeric(factor(houses$GarageCond,
                                             levels = c("Ex", "Gd", "TA", "Fa","Po","Not apply"),
                                             labels = c(5,4,3,2,1,0) ,ordered = TRUE))
houses$ExterCondnum <- as.numeric(factor(houses$ExterCond,
                                             levels = c("Ex", "Gd", "TA", "Fa", "Po"),
                                             labels = c(5,4,3,2,1) ,ordered = TRUE))
houses$HeatingQCnum <- as.numeric(factor(houses$HeatingQC,
                                             levels = c("Ex", "Gd", "TA", "Fa", "Po"),
                                             labels = c(5,4,3,2,1) ,ordered = TRUE))
houses$PoolQCnum <- as.numeric(factor(houses$PoolQC,
                                             levels = c("Ex", "Gd", "Fa", "Not apply"),
                                             labels = c(3,2,1,0) ,ordered = TRUE))
houses$BsmtQualnum <- as.numeric(factor(houses$BsmtQual,
                                             levels = c("Ex", "Gd", "Fa", "Not apply"),
                                             labels = c(3,2,1,0) ,ordered = TRUE))
```

Visualizamos algunos dispersiogramas con una línea de ajuste, histogramas para ver la distribución de las variables, y además los coeficientes de correlación

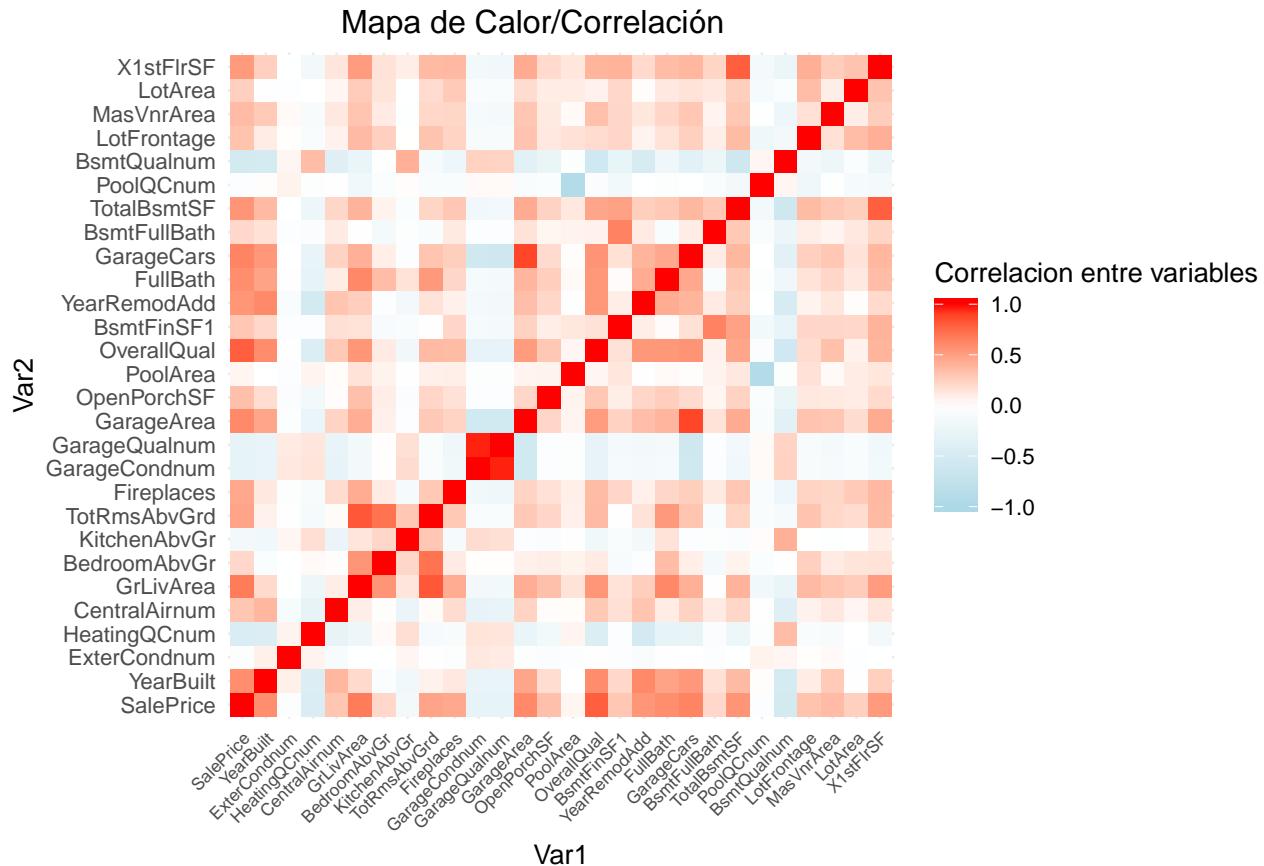
Seleccionamos algunas variables para la correlación

```
correl <- c('SalePrice', 'YearBuilt','ExterCondnum','HeatingQCnum',
           'CentralAirnum','GrLivArea','BedroomAbvGr','KitchenAbvGr',
           'TotRmsAbvGrd','Fireplaces','GarageCondnum', 'GarageQualnum',
           'GarageArea','OpenPorchSF','PoolArea','OverallQual',
           'BsmtFinSF1', 'YearRemodAdd','FullBath','GarageCars',
           'BsmtFullBath','TotalBsmtSF','PoolQCnum', 'BsmtQualnum',
           'LotFrontage', 'MasVnrArea','LotArea', 'X1stFlrSF')
heat <- houses[,correl]
```

heatmap con Saleprice

```
options(repr.plot.width=10, repr.plot.height=10)
qplot(x=Var1, y=Var2, data=melt(cor(heat, use="p")), fill=value, geom="tile") +
  scale_fill_gradient2(low = "light blue", high = "red", mid = "white",
  midpoint = 0, limit = c(-1,1), space = "Lab",
  name="Correlacion entre variables") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 7, hjust = 1))+ 
  coord_fixed()+
  ggtitle("Mapa de Calor/Correlación") +
  theme(plot.title = element_text(hjust = 0.4))
```

```
## Warning in melt(cor(heat, use = "p")): The melt generic in data.table has
## been passed a matrix and will attempt to redirect to the relevant reshape2
## method; please note that reshape2 is deprecated, and this redirection is
## now deprecated as well. To continue using melt methods from reshape2 while
## both libraries are attached, e.g. melt.list, you can prepend the namespace
## like reshape2::melt(cor(heat, use = "p")). In the next version, this
## warning will become an error.
```



Correlacionan en positivo con SalePrice -> X1stFlrSF, LotArea, MasVnrArea, LotFrontage, TotalBsmtSF, GarageCars, FullBath, YearRemodAdd, BsmtFinSF1, OverallQual, OpenPorchSF, GarageArea, Fireplaces, TotRmsAbvGrd, BedroomAbvGr, GrLivArea, CentralAir, HeatingQC, ExterCond, KitchenAbvGr, PoolQC, BsmtQual, BsmtFinSF, YearRemodAdd, BsmtFinSF1, FullBath, GarageCars, TotalBsmtSF, PoolQC, BsmtQual, BsmtFinSF, LotFrontage, LotArea, X1stFlrSF

Asimismo se aprecian variables que no correlacionan como ExterCond, KitchenAbvGr, PoolArea, PoolQC.

Creamos un dataset con las variables que correlacionan y estudiamos mediante componentes principales, aplicando PCA, el porcentaje con el cual contribuye cada variable.

```

correlpca <- c( 'YearBuilt', 'CentralAirnum', 'GrLivArea', 'BedroomAbvGr',
              'TotRmsAbvGrd', 'Fireplaces', 'GarageArea', 'OpenPorchSF',
              'OverallQual', 'BsmtFinSF1', 'YearRemodAdd', 'FullBath',
              'GarageCars', 'BsmtFullBath', 'TotalBsmtSF', 'X1stFlrSF')
heatpca <- houses[,correlpca]

```

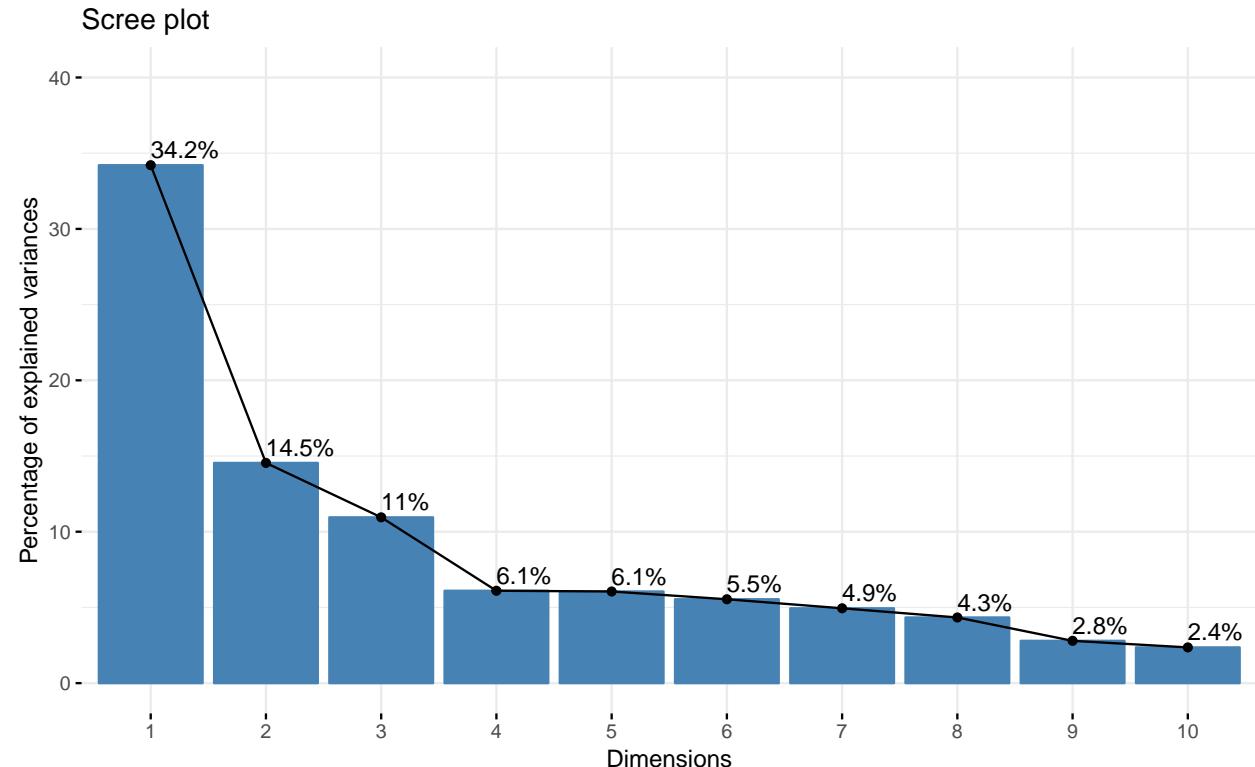
PCA

Con el fin de reducir la dimension, vamos a llevar a cabo un estudio de análisis de componentes principales o PCA, en sus siglas en inglés. PCA se trata de una transformación lineal k-dimensional del espacio paramétrico a un espacio n-dimensional.

```
res.pca <- PCA(heatpca, graph = FALSE)
```

Visualizamos el porcentaje de variación de la información que somos capaces de explicar con los diferentes componentes principales:

```
fviz_screepplot(res.pca, addlabels = TRUE, ylim = c(0, 40))
```



Con los primeros dos componentes PC1 y PC2 se explica el 50% de la varianza y con PC3 el 60%.

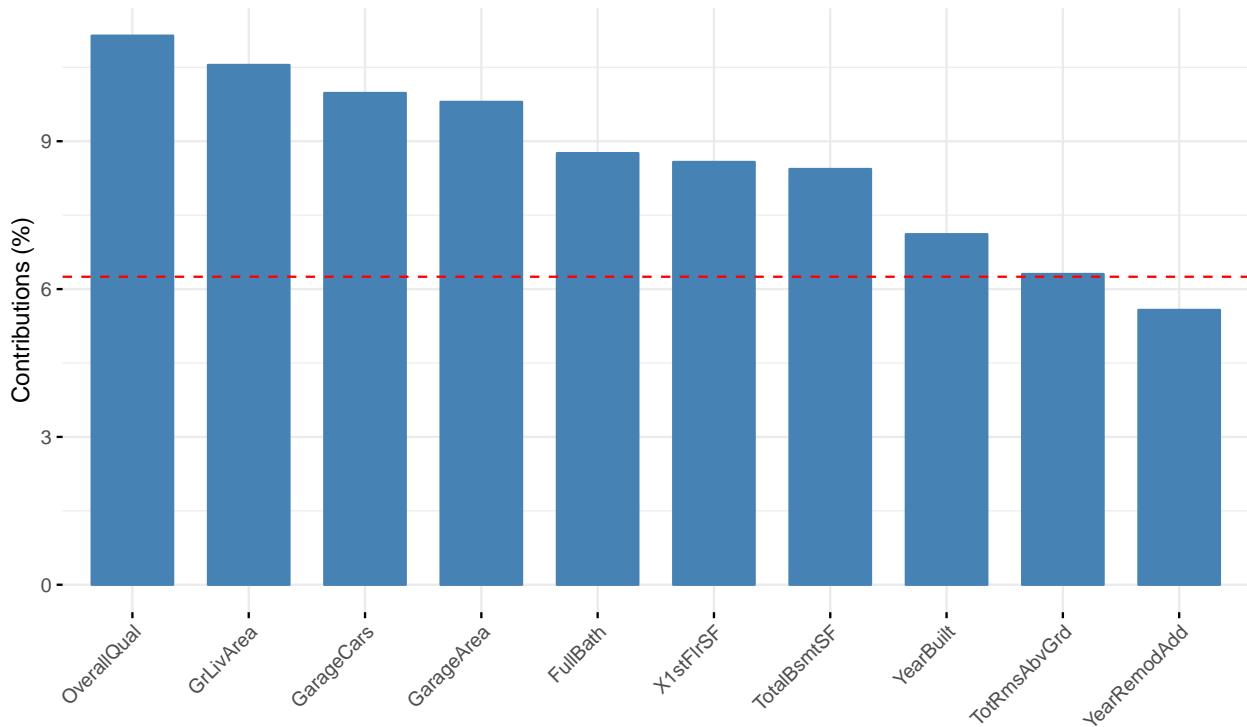
Se visualizan las variables de estos 3 primeros componentes.

```

var <- get_pca_var(res.pca)
# Contribucion PC1
fviz_contrib(res.pca, choice = "var", axes = 1, top = 10)

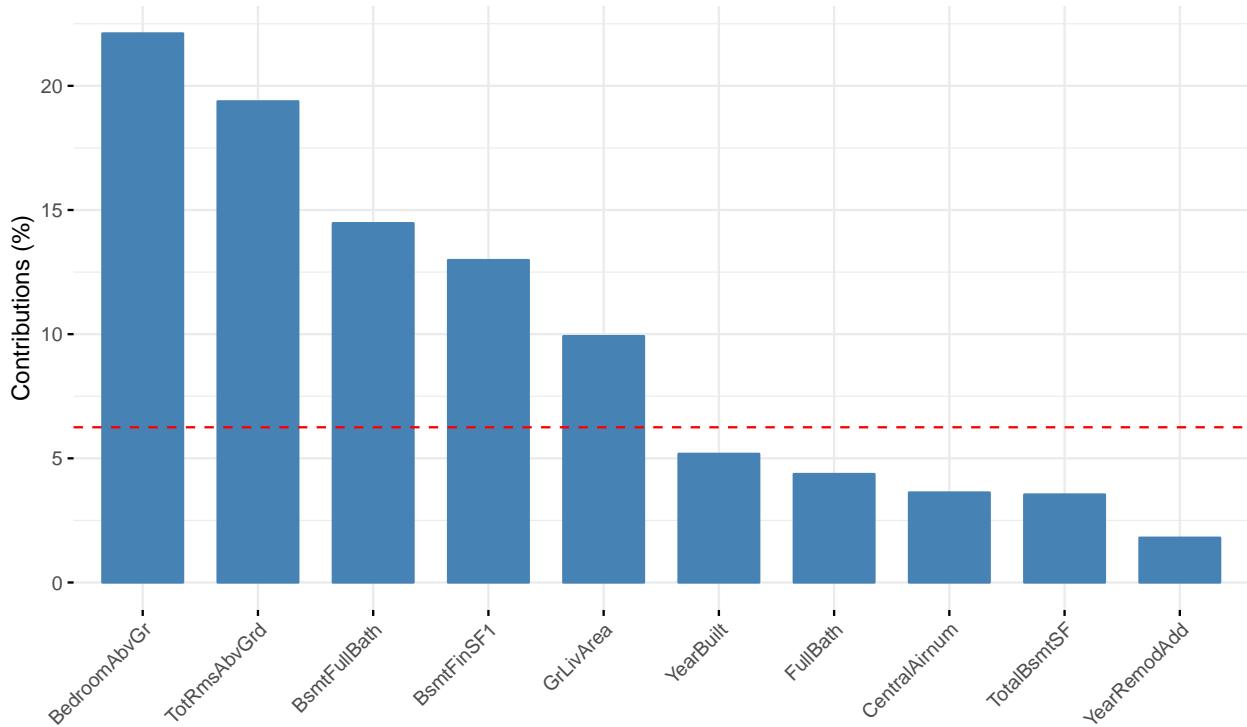
```

Contribution of variables to Dim–1



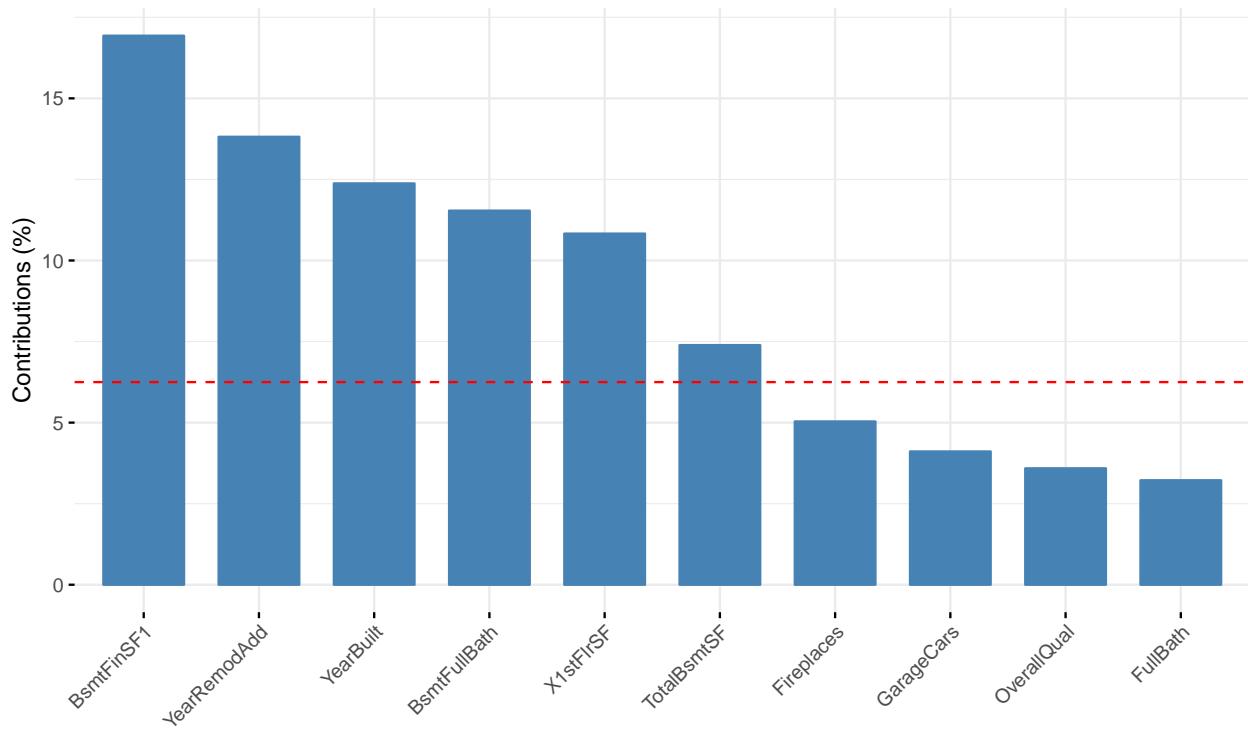
```
# Contribucion PC2
fviz_contrib(res.pca, choice = "var", axes = 2, top = 10)
```

Contribution of variables to Dim–2



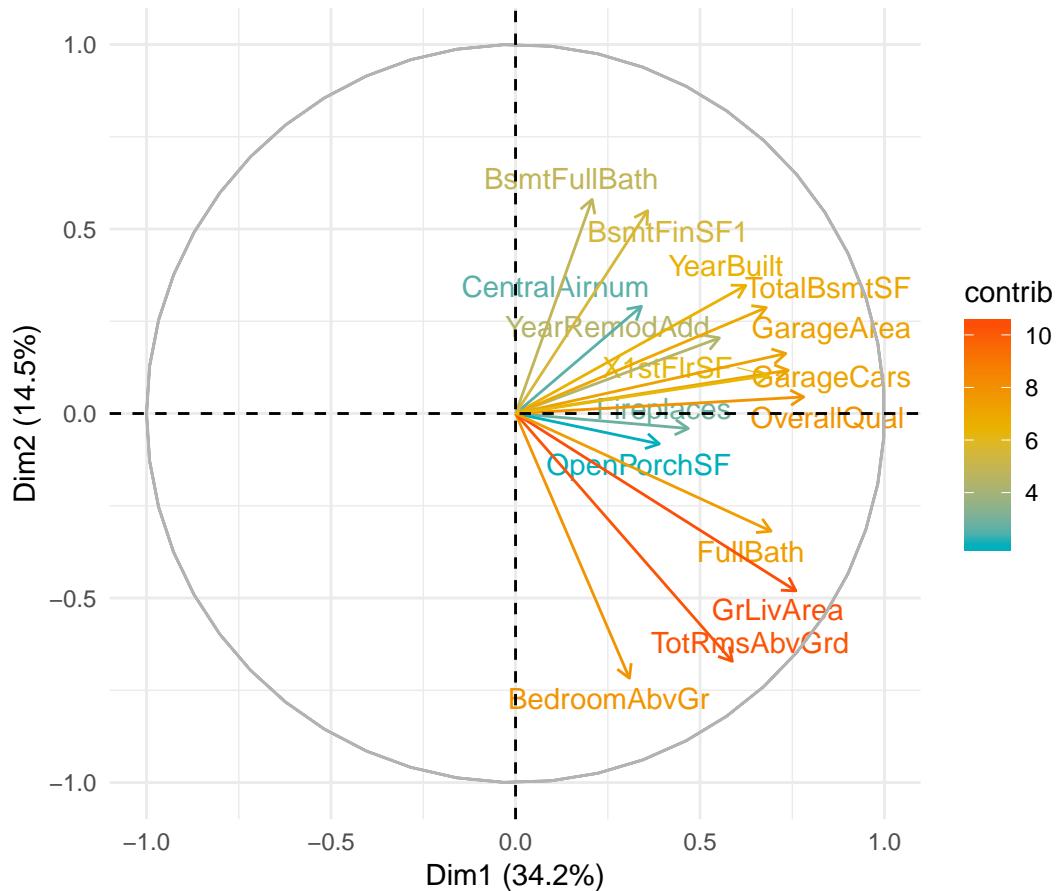
```
# Contribucion PC3
fviz_contrib(res.pca, choice = "var", axes = 3, top = 10)
```

Contribution of variables to Dim-3



```
fviz_pca_var(res.pca, col.var="contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE # Avoid text overlapping)
) + theme_minimal() + ggtitle("Variables - PCA")
```

Variables – PCA



En el biplot de los dos componentes principales más importantes podemos observar que variables contribuyen más a formar cada uno de ellos.

Para el PC1, se aprecia que las variables principales son OverQuall (Calidad Vivienda), GrLivearea (Nº de metros habitables), GarageCars (Nº de plazas de aparcamiento), GarageArea (Nº de metros del garaje), FullBath (Nº de baños), X1stFlrSF (Nº metros habitables en la primera planta) y TotalBsmtSF (Nº de metros del sotano totales).

Para el PC2, las variables BedroomAbvGr (Nº de habitaciones en primera planta), TotRmsAbvGrd (Nº totales de habitaciones), BsmtFullBath (Nº de baños completos), BsmtFinSF1 (Metros acabados del sotano9), y GrLivArea (Nº de metros de area habitable).

En PC3 se observa que destaca algo más el año de remodelación y de construcción junto a BsmtFinSF1 (Nº de metros del sotano acabados), BsmtFullBath (Nº de baños).

Continuando con el PCA se va a revisar mediante cluster la distribución por atributos y se va a verificar los outliers una vez que se han eliminado para algunas variables.

Clustering

En una primera aproximación vamos a realizar un análisis con clusters para a través de la visualización de estos poder hacernos una idea de cual sería la agrupación en base a las variables y confirmar si después de la eliminación de outliers sigue apareciendo algún elemento atípico o extremo. La función “fviz_cluster” representa las observaciones en un plano utilizando los dos primeros PCAs cuando el número de columnas es mayor a dos.

Reducimos la cantidad de observaciones por claridad para el estudio de los cluster y se escalan las variables:

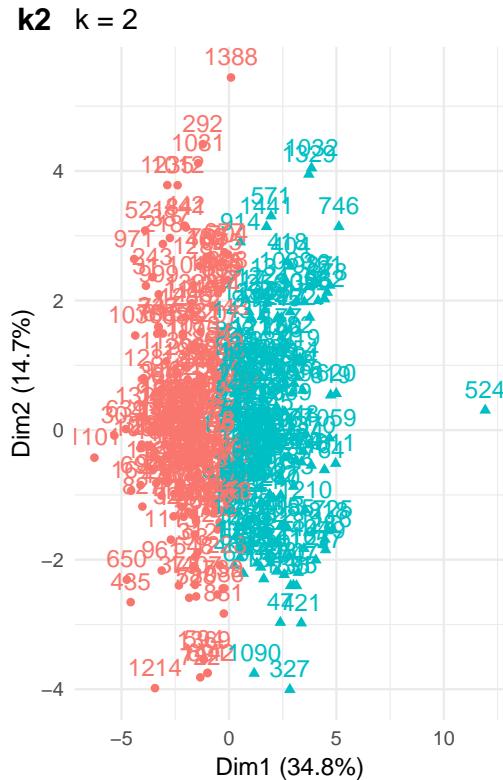
```
houses5<-houses[,c('YearBuilt', 'TotalBsmtSF',
                     'CentralAirnum', 'GrLivArea', 'BedroomAbvGr',
                     'TotRmsAbvGrd', 'Fireplaces',
                     'GarageArea', 'OpenPorchSF', 'OverallQual',
                     'BsmtFinSF1', 'YearRemodAdd', 'FullBath', 'GarageCars',
                     'BsmtFullBath', 'X1stFlrSF')]

set.seed(1234)
houses5 <- houses5[sample(nrow(houses), 400),]
houses5 <- scale(houses5)

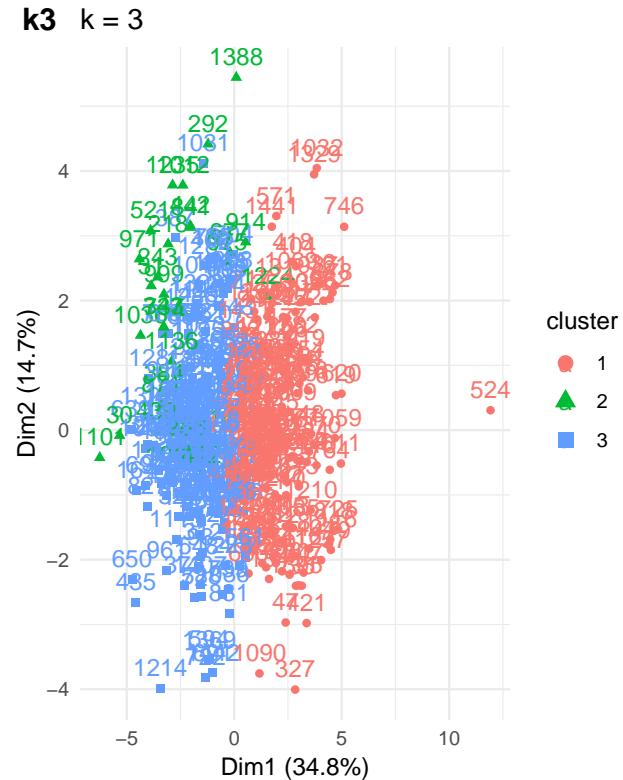
set.seed(1234)
kmean_calc <- function(df, ...){
  kmeans(df, scaled = ..., nstart = 30)
}

km2 <- kmean_calc(houses5, 2)
km3 <- kmean_calc(houses5, 3)
km4 <- kmeans(houses5, 4)
km5 <- kmeans(houses5, 5)

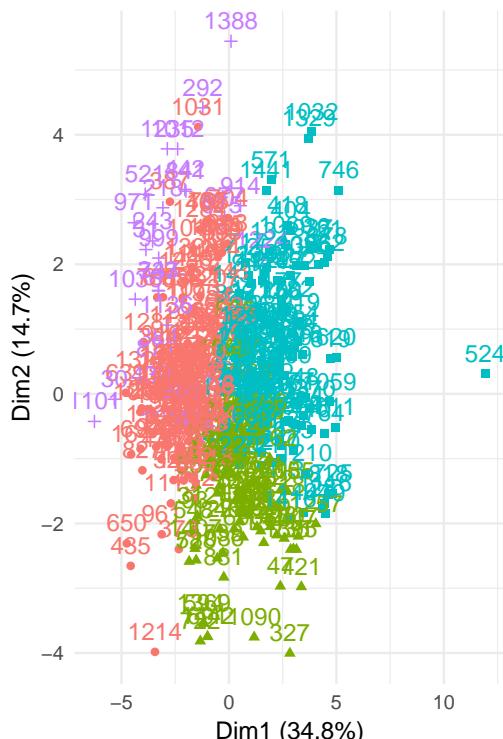
p1 <- fviz_cluster(km2, data = houses5, ellipse.type = "ellipse") +
  theme_minimal() + ggtitle("k = 2")
p2 <- fviz_cluster(km3, data = houses5, ellipse.type = "ellipse") +
  theme_minimal() + ggtitle("k = 3")
p3 <- fviz_cluster(km4, data = houses5, ellipse.type = "ellipse") +
  theme_minimal() + ggtitle("k = 4")
p4 <- fviz_cluster(km5, data = houses5, ellipse.type = "ellipse") +
  theme_minimal() + ggtitle("k = 5")
plot_grid(p1, p2, labels = c("k2", "k3"))
```



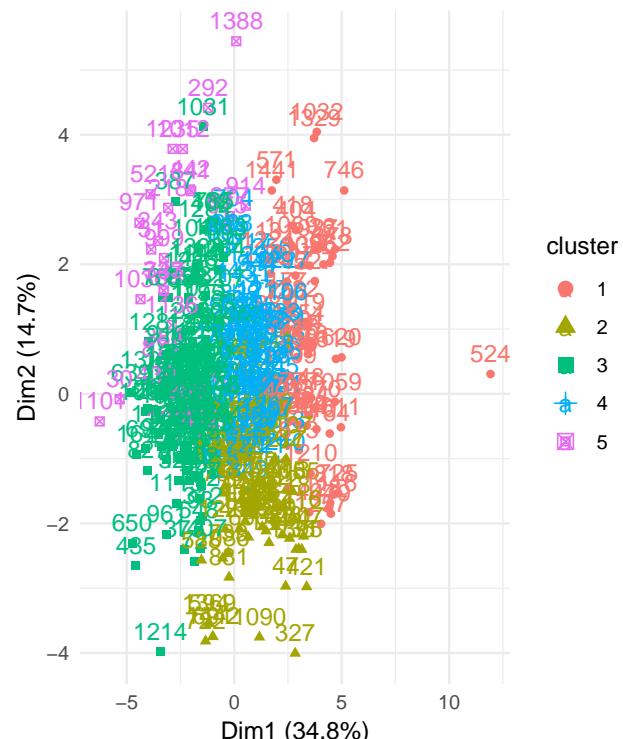
```
plot_grid(p3, p4, labels = c("k4", "k5"))
```



k4 k = 4



k5 k = 5



Tras una primera visualización se podrían indicar como número óptimo de clusters $k = 4$. Revismos el tamaño de estos cluster

```

set.seed(1234)
clus <- kmeans(houses5, 4)
housesclu <- data.frame(houses5, clus$cluster)
aggregate(housesclu, by=list(clus$cluster), FUN=mean)

##   Group.1  YearBuilt TotalBsmtSF CentralAirnum  GrLivArea BedroomAbvGr
## 1       1 -0.1736047  0.2865357   -0.2685917  1.5837966    1.4244021
## 2       2  0.9249690  0.3906409     0.2843912  0.3533149   -0.1248754
## 3       3 -0.8484795 -0.6353960   -0.3328921 -0.6054165   -0.1522843
## 4       4 -0.1038822  0.1708671    0.1962079 -0.5386936   -0.3676594
##   TotRmsAbvGrd Fireplaces GarageArea OpenPorchSF OverallQual BsmtFinSF1
## 1      1.5706540  0.4859959  0.6773453   0.5747570  0.2913963 -0.1464924
## 2      0.1825349  0.1748497  0.5466652   0.2771226  0.8534579  0.1341031
## 3     -0.4287725 -0.4434220 -0.9370283  -0.3995996 -0.7021046 -0.5344447
## 4     -0.5242675  0.1153399  0.1566390  -0.1628580 -0.4690612  0.6698009
##   YearRemodAdd FullBath GarageCars BsmtFullBath X1stFlrSF clus.cluster
## 1     -0.004618294  0.9802635  0.6821435   -0.23911428  0.7857036           1
## 2      0.772677285  0.8493129  0.6359262   0.08272302  0.2312484           2
## 3     -0.703241707 -0.7076833 -1.0161473   -0.55167587 -0.6125274           3
## 4     -0.173452671 -0.8385741  0.1304444    0.82919324  0.1118748           4

```

Para el cluster 1 destacan las variables TotalBsmtSF, X1stFlrSF, OverallQual, GrLivArea, GarageArea y GarageCars. Para el cluster 2 destacan BsmtFullBath, BsmtFinSF1.

Para el cluster 3 destacan FullBath, TotRmsAbvGrd. Para el cluster 4 destacan YearBuilt, OverallQual, FullBath, GarageCars.

Como conclusión podemos extraer en general que hay agrupaciones en base a los metros cuadrados o area de la propiedad como se esperaba. Además también en base al número de habitaciones, número de baños y plazas de aparcamiento. Y agrupaciones en base al año de construcción y calidades de la vivienda. Con ello se confirma lo que ya se había extraido en el estudio previo de relaciones entre variables.

Vemos el tamaño de cada cluster:

```

set.seed(1234)
clus$size

## [1] 48 137 129 86

```

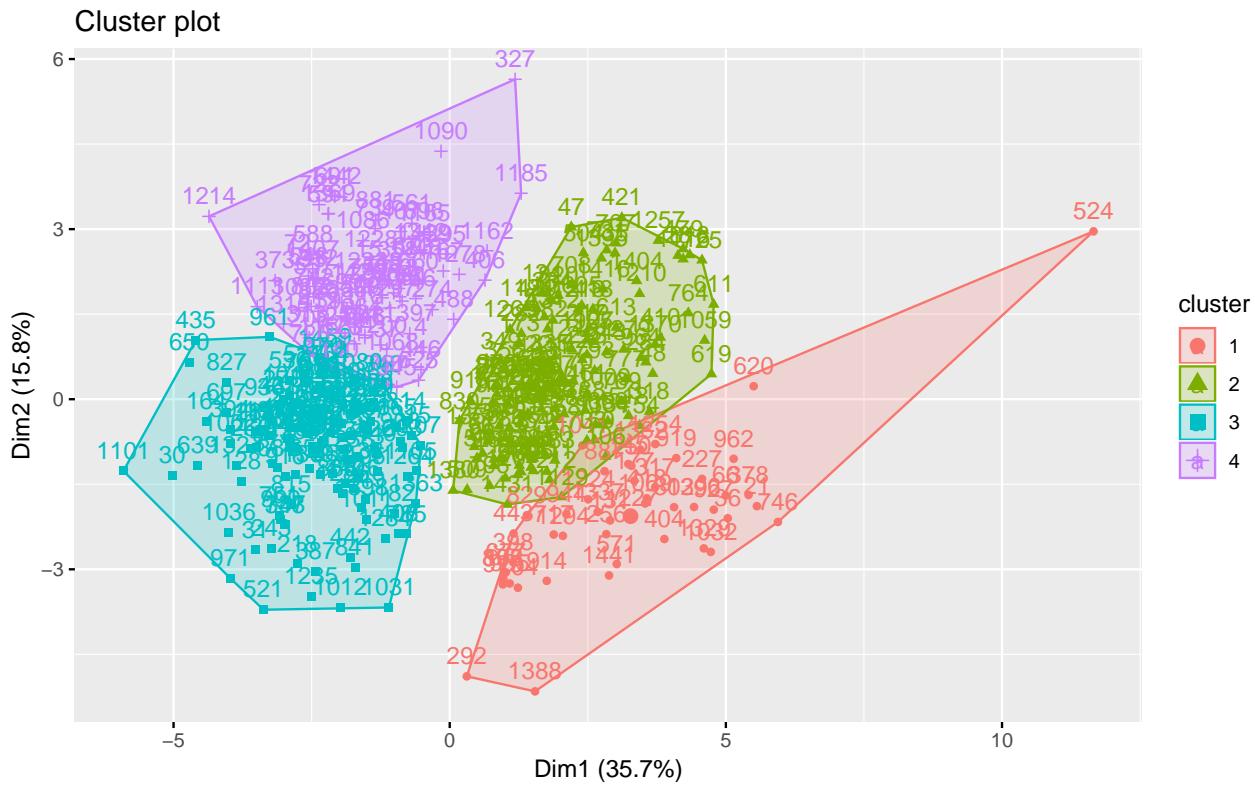
Están equilibrados, se muestran visualmente

```

set.seed(1234)
housesclu$cluster <- clus$cluster

fviz_cluster(clus, data=housesclu)

```



Se obtienen los valores de las variables para los centros de los clusters:

```
set.seed(1234)
clucen<- kmeans(housesclu, centers=4)
clucen$centers

##      YearBuilt TotalBsmtSF CentralAirnum GrLivArea BedroomAbvGr
## 1 -0.2838374   0.4304149  -0.3962032  1.6277582   1.5289383
## 2  0.8954491   1.3087091   0.2843912  0.5020002  -0.4061016
## 3  0.8470574  -0.1976424   0.2843912  0.3561497   0.1146932
## 4 -0.5694969  -0.3466868  -0.1309103 -0.5979746  -0.2250570
##      TotRmsAbvGrd Fireplaces GarageArea OpenPorchSF OverallQual BsmtFinSF1
## 1     1.6423477  0.59888384  0.7386401  0.69860878  0.3319666 -0.11794068
## 2     0.2476283  0.58119760  0.9775728  0.61650045  1.1935519  1.07331953
## 3     0.2230637 -0.02448569  0.3064154  0.06597272  0.5410763 -0.38504209
## 4    -0.4720684 -0.25801466 -0.5396746 -0.32660811 -0.6278111 -0.09707415
##      YearRemodAdd FullBath GarageCars BsmtFullBath X1stFlrSF clus.cluster
## 1    -0.03110917  0.9592175  0.6713200 -0.16813302  0.7781296  1.000000
## 2     0.78713412  0.7313508  0.9028301  0.82722899  1.2625574  2.157895
## 3     0.67766431  0.8630071  0.4863524 -0.35093768 -0.2804848  1.914894
## 4    -0.51120872 -0.7629483 -0.5874282 -0.03622249 -0.3616536  3.385714
##      cluster
## 1 1.000000
## 2 2.157895
## 3 1.914894
## 4 3.385714
```

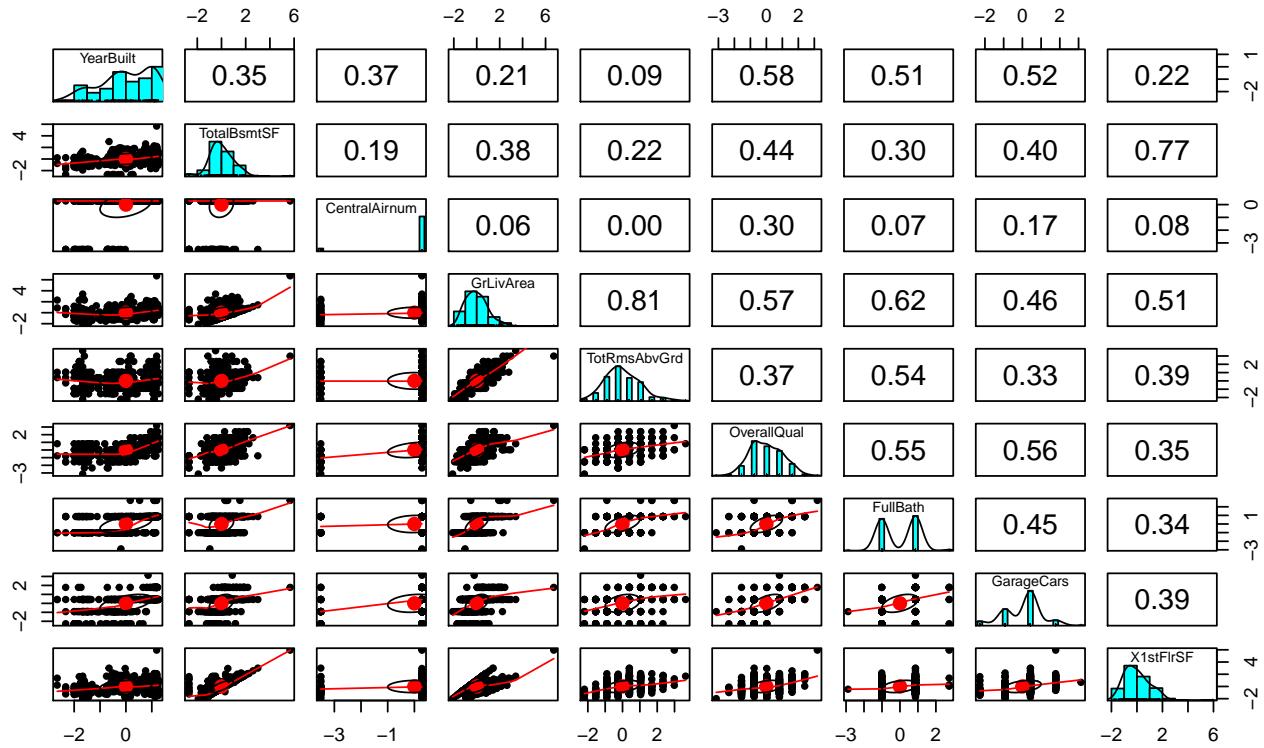
Con esto se confirma que la variables que más influyen son los metros cuadrados o area de la propiedad, el número de habitaciones, número de baños, plazas de aparcamiento, año de construcción y remodelación.

Si revisamos la correlación de algunas variables del cluster:

```

pairs.panels(houses5[, c('YearBuilt', 'TotalBsmtSF',
  'CentralAirnum', 'GrLivArea',
  'TotRmsAbvGrd', 'OverallQual', 'FullBath',
  'GarageCars', 'X1stFlrSF')])

```



Se aprecia que hay variables que correlacionan fuertemente como:

GrLiveArea con TotRmsAbvGrd, YaerBuilt con OverallQual, GrLiveArea con FullBath, TotRmsAbvGrd con X1stFlrSF.

Contraste de hipótesis

Diferencias en el nivel del precio de la casa, diferenciando por los que tienen AireAcondicionado y los que no.

Escribir la hipótesis nula y alternativa

H0: Hipótesis Nula - H0 : El precio de las casas con aire acondicionado = El precio de las casas con aire acondicionado (Hipótesis que queremos descartar) H1: Hipótesis Alternativa : H1 : El precio de las casas con aire acondicionado es diferente que las casas que no lo tienen (Se establece como la hipótesis a confirmar)

Método

a.Se trata de un contraste entre dos muestras que no estan relacionadas (independientes) porque son tipos diferentes de caracteristica de la casa. b.Consideramos normal la muestra puesto que según el teorema del limite central, el tamaño de las muestras es mayor de 30 y varianza desconocida. c.Metodo paramétrico ya que los métodos no parametricos, se utilizan cuando no se cumple el supuesto de normalidad y el tamaño de la muestra es pequeño. En este caso , el tamaño de la muestra no se puede considerar pequeño y los datos tienen una distribución normal. d.La hipótesis alternativa es bilateral ya que queremos ver si la media del precio de las casas es superior o inferior a la otro tipo.

Calculos

Paso 1) Las hipotesis han sido planteadas en el anterior apartado. $H_0 = \text{El precio de las casas con aire acondicionado} = \text{El precio de las casas con aire acondicionado}$. Por lo tanto la diferencia entre las medias es igual a 0.

$$H_0 : \mu = \mu_0$$

$H_1 = \text{El precio de las casas con aire acondicionado es diferente a la de las que no tiene aire acondicionado}$. Por tanto la diferencia entre las medias es distinto de cero

$$H_1 : \mu \neq \mu_0$$

Paso 2) Definimos las muestras de las dos poblaciones, que son independientes) y hacemos el test de sapiro para cada muestra

```
muestra.aire <- houses$SalePrice[which(houses$CentralAir == "Y")]
tamano.muestra.aire <- length(muestra.aire)
tamano.muestra.aire

## [1] 1300
shapiro.test(muestra.aire)

##
## Shapiro-Wilk normality test
##
## data: muestra.aire
## W = 0.95421, p-value < 0.0000000000000022

muestra.sinaire <- houses$SalePrice[which(houses$CentralAir == "N")]
tamano.muestra.sinaire<- length(muestra.sinaire)
tamano.muestra.sinaire

## [1] 92
shapiro.test(muestra.sinaire)

##
## Shapiro-Wilk normality test
##
## data: muestra.sinaire
## W = 0.95597, p-value = 0.003502

# TEst de igualdad de varianzas para verificar si son iguales o no.
#H0: F (ratio de varianzas) = 1 (Varianzas iguales) (Hipotesis que queremos descartar)
#H1 : F diferente de 1 (Varianzas distintas) (Hipotesis alternativa)
var.test(muestra.aire, muestra.sinaire,alternative = "two.sided")

##
## F test to compare two variances
##
## data: muestra.aire and muestra.sinaire
## F = 2.3351, num df = 1299, denom df = 91, p-value = 0.000001143
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 1.691985 3.096179
## sample estimates:
## ratio of variances
## 2.335079
```

El resultado del test de varianzas nos indica que son distintas .

Paso 3) Definimos el nivel de significación = 0,05

```
nivel.signif.2muestras = 0.05
```

Paso 4) Calculamos la media muestrales y las varianzas muestrales

```
media.muestra.aire <- mean(muestra.aire)  
media.muestra.aire
```

```
## [1] 174967
```

```
media.muestra.sinaire<- mean(muestra.sinaire)  
media.muestra.sinaire
```

```
## [1] 103457.7
```

```
desvi.std.aire <- sd(muestra.aire)  
desvi.std.aire
```

```
## [1] 57523.76
```

```
desvi.std.sinaire<- sd(muestra.sinaire)  
desvi.std.sinaire
```

```
## [1] 37644.07
```

```
varianza.muestra.aire <- desvi.std.aire*desvi.std.aire  
varianza.muestra.aire
```

```
## [1] 3308983179
```

```
varianza.muestra.sinaire <- desvi.std.sinaire*desvi.std.sinaire  
varianza.muestra.sinaire
```

```
## [1] 1417075639
```

Las varianzas se observan que son distintas tal y como nos habia adelantado el test anteriamente sobre las varianzas.

Paso 5) Calculamos el error estandar y el estadístico de contraste

```
error.estandar.2muestras <- sqrt(varianza.muestra.aire/tamano.muestra.aire+
                                    varianza.muestra.sinaire/tamano.muestra.sinaire)
```

error.estandar.2muestras

```
## [1] 4236.551
```

```
estadistico.contraste<-(media.muestra.aire-media.muestra.sinaire)/error.estandar.2muestras  
estadistico.contraste
```

```
## [1] 16.87914
```

Paso 6) Calculamos el p-valor como : $2P(Z>|z|)$ por la hipótesis planteada.

```
p.valor = 2*pt(abs(estadistico.contraste),df=pmin(tamano.muestra.aire,
                                                    tamano.muestra.sinaire)-1,lower.tail = F)
p.valor
```

Si El valor de p-value es menor que el nivel de significación $< 0,05$, por lo que rechazaremos la hipótesis nula a favor de la alternativa. Es decir que se demuestra que los precios medios son distintos en función de si

tienen o no aire acondicionado.

Interpretación de los resultados

CONCLUSiON : El valor de p-value es menor que el nivel de significación $< 0,05$, por lo que rechazaremos la hipótesis nula a favor de la alternativa. Es decir, que existe diferencia en la media entre los que tienen aire acondicionado y los que no.

En R :

```
# Aplica la función de Welch ya que las varianzas no son iguales:  
sol.ttest=t.test(muestra.aire,muestra.sinaire,alternative="two.side",  
                   var.equal=FALSE,conf.level=0.95)  
sol.ttest  
  
##  
##  Welch Two Sample t-test  
##  
## data: muestra.aire and muestra.sinaire  
## t = 16.879, df = 123.32, p-value < 0.0000000000000022  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 63123.57 79895.12  
## sample estimates:  
## mean of x mean of y  
## 174967.0 103457.7
```

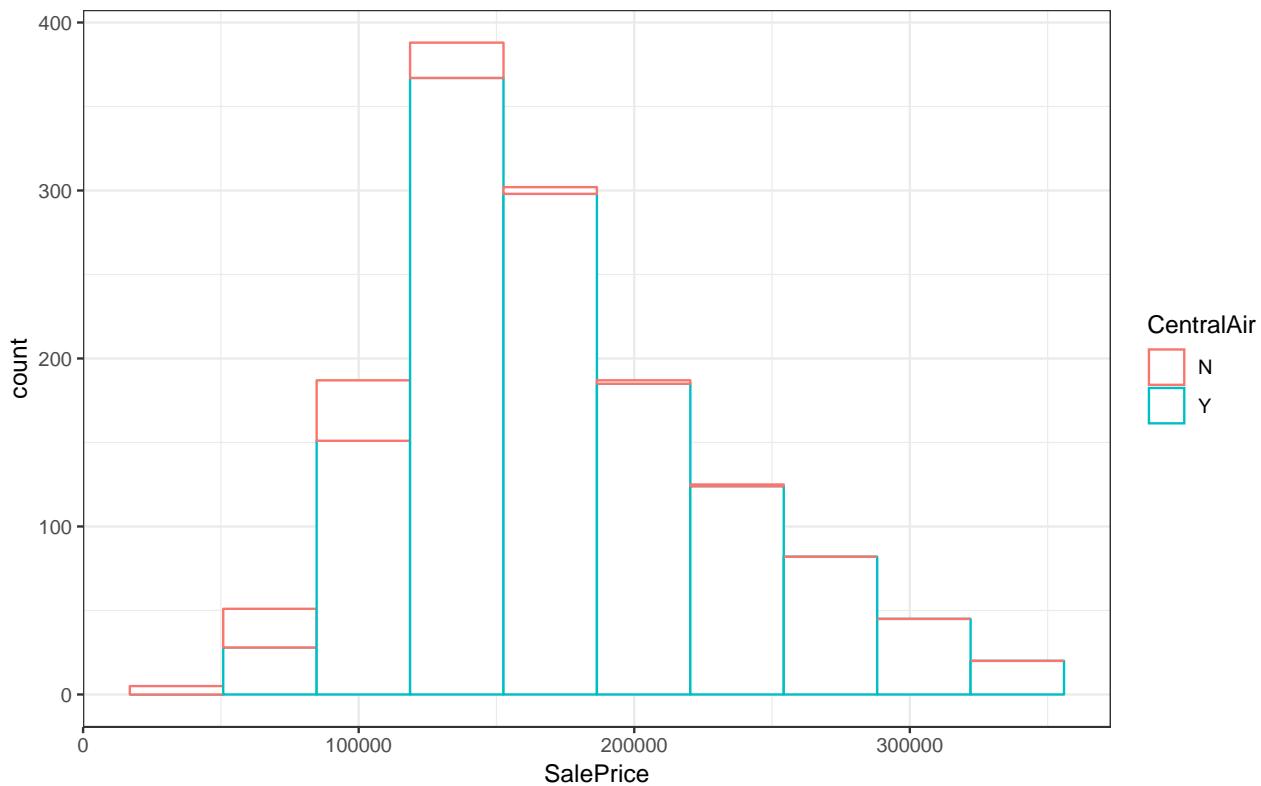
Contraste de Hipótesis (Método no paramétrico)

Condiciones para aplicar el test: * Los datos tienen que ser independientes

* Los datos tienen que se tienen que poder ordenar de menor a mayor. * La distribución de los datos no tiene porque asumirse como normal o que proceden de poblaciones normales. * Igualdad de varianza entre grupos (homocedasticidad) -> Esto en este caso no cumple

Las hipótesis son iguales que antes.

```
par(mfrow=c(1,2))  
ggplot(data =houses , mapping = aes(x = SalePrice, colour = CentralAir)) +  
  geom_histogram( fill = "white" , bins = 10) +  
  theme_bw()
```



```
#+ facet_grid(. ~ CentralAir) +
  theme(legend.position = "none")
```

```
## List of 1
## $ legend.position: chr "none"
## - attr(*, "class")= chr [1:2] "theme" "gg"
## - attr(*, "complete")= logi FALSE
## - attr(*, "validate")= logi TRUE
```

Un método no paramétrico es el siguiente :

```
wilcox.test(muestra.aire,muestra.sinaire,alternative = "two.sided",mu =0, paired = F,
            conf.level = 0.95, conf.int = TRUE)
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: muestra.aire and muestra.sinaire
## W = 103820, p-value < 0.000000000000022
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
## 55500 75500
## sample estimates:
## difference in location
## 65100
```

El metodo no paramétrico también nos indica que son distintas medias.

ANOVA de un factor

Por ejemplo podemos utilizar la función de ANOVA para ver como varian los precios de las casas según el vecindario. Vamos a filtrar por los que eran menores de 345000 ya que ese valor ya existia como outliers..

```
#Se seleccionan los datos  
datosAnova <- houses[which(houses$SalePrice < 345000),]
```

Hipótesis nula y alternativa

H_0 = El precio medio de las viviendas que cuestan menos de 345000 es igual independientemente del barrio en el que se ubique. (Todas las medias poblacionales son iguales) (Hipótesis que queremos descartar)

$$H_0 : \mu_1 = \mu_2 = \mu_3$$

H_1 = El precio medio de las viviendas que cuestan menos de 345000 es distinto dependiendo del barrio en la que se ubique. (No todas las medias son iguales) Hipótesis que queremos demostrar.

Modelo

El análisis de la varianza (ANOVA) de un conjunto de muestras consiste en contrastar la hipótesis nula “todas las medias poblacionales” de las que provienen las muestras son iguales”, contra la hipótesis alternativa “no todas las medias son iguales” con un nivel de significación prefijado.

Vemos el nº de muestras de cada tipo.

```
tam.muestras <- table(datosAnova$Neighborhood)  
tam.muestras  
  
##  
## Blmngtn Blueste BrDale BrkSide ClearCr CollgCr Crawfor Edwards Gilbert  
## 17 2 16 58 28 148 48 100 78  
## IDOTRR MeadowV Mitchel NAmes NoRidge NPkVill NridgHt NWAmes OldTown  
## 37 17 49 224 30 9 50 73 106  
## Sawyer SawyerW Somerst StoneBr SWISU Timber Veenker  
## 73 59 84 16 25 35 10  
  
#Vemos la media del salario en función del vecindario  
datosAgrupadosMedia <- aggregate(SalePrice ~ Neighborhood , data = datosAnova, FUN = mean)  
datosAgrupadosMedia
```

```
## Neighborhood SalePrice  
## 1 Blmngtn 194870.88  
## 2 Blueste 137500.00  
## 3 BrDale 104493.75  
## 4 BrkSide 124834.05  
## 5 ClearCr 212565.43  
## 6 CollgCr 195175.85  
## 7 Crawfor 200192.94  
## 8 Edwards 128219.70  
## 9 Gilbert 190487.26  
## 10 IDOTRR 100123.78  
## 11 MeadowV 98576.47  
## 12 Mitchel 156270.12  
## 13 NAmes 144958.00  
## 14 NoRidge 282386.93  
## 15 NPkVill 142694.44
```

```

## 16      NridgHt 260396.52
## 17      NWAmes 189050.07
## 18      OldTown 121905.47
## 19      Sawyer 137379.34
## 20      SawyerW 186555.80
## 21      Somerst 221295.10
## 22      StoneBr 239312.50
## 23      SWISU 142591.36
## 24      Timber 230914.37
## 25      Veenker 224150.00

#Vemos la media del salario en función del tipo de Vivienda
datosAgrupadosMediaBldg <- aggregate(SalePrice ~ BldgType , data = datosAnova, FUN = mean)
datosAgrupadosMediaBldg

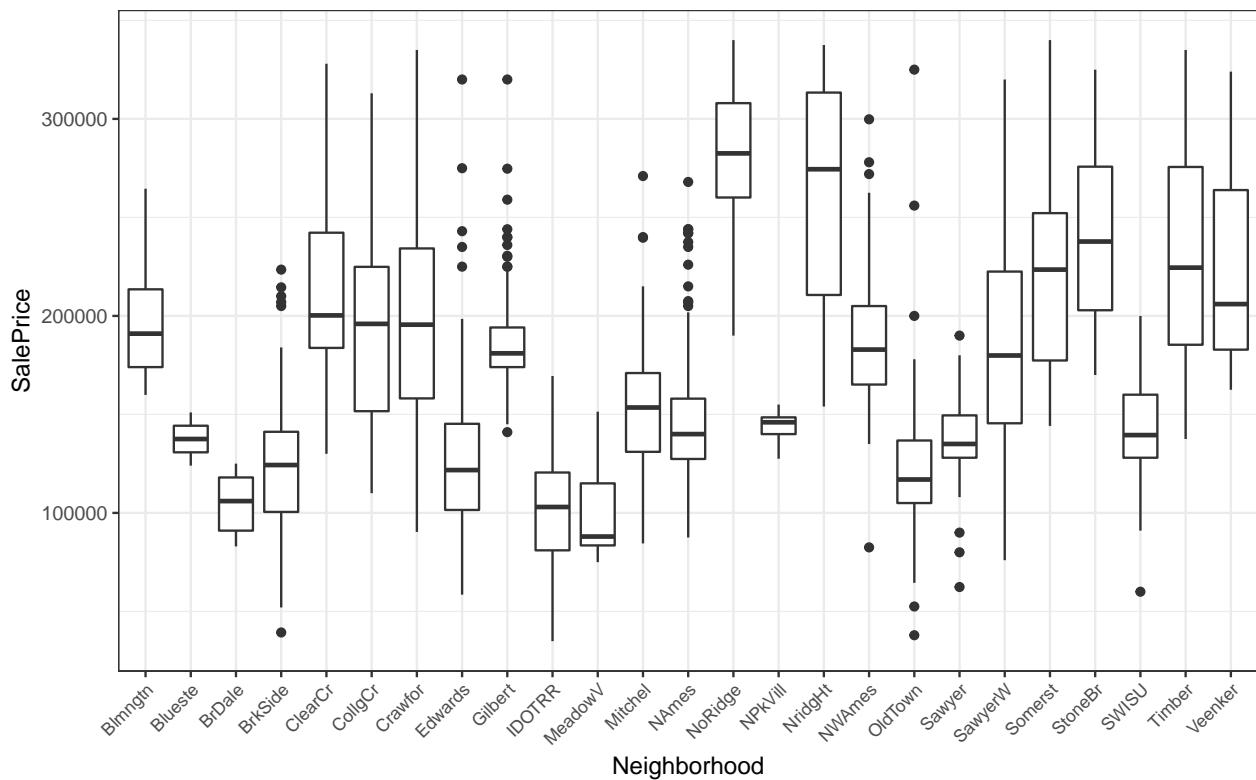
##   BldgType SalePrice
## 1     1Fam 173502.6
## 2    2fmCon 128432.3
## 3    Duplex 133541.1
## 4    Twnhs 135911.6
## 5  TwnhsE 178424.7

#Vemos la media del salario en función de la zona
datosAgrupadosMediaZon <- aggregate(SalePrice ~ MSZoning , data = datosAnova, FUN = mean)
datosAgrupadosMediaZon

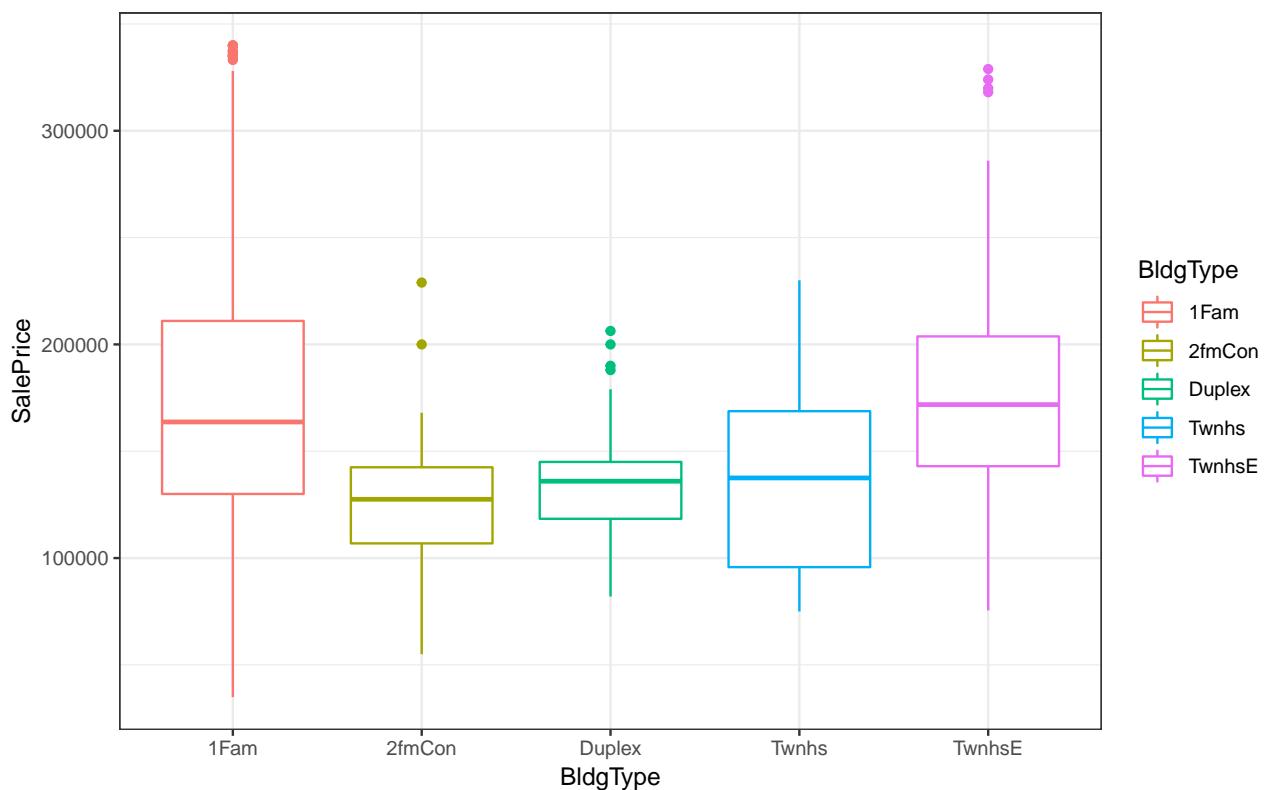
##   MSZoning SalePrice
## 1    C (all) 74528.0
## 2      FV 211563.1
## 3      RH 131558.4
## 4      RL 178578.7
## 5      RM 121795.7

# Pintamos un grafico para mostrar los datos..
ggplot(data = datosAnova, aes(x = Neighborhood , y = SalePrice )) +
  geom_boxplot() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 8, hjust = 1))

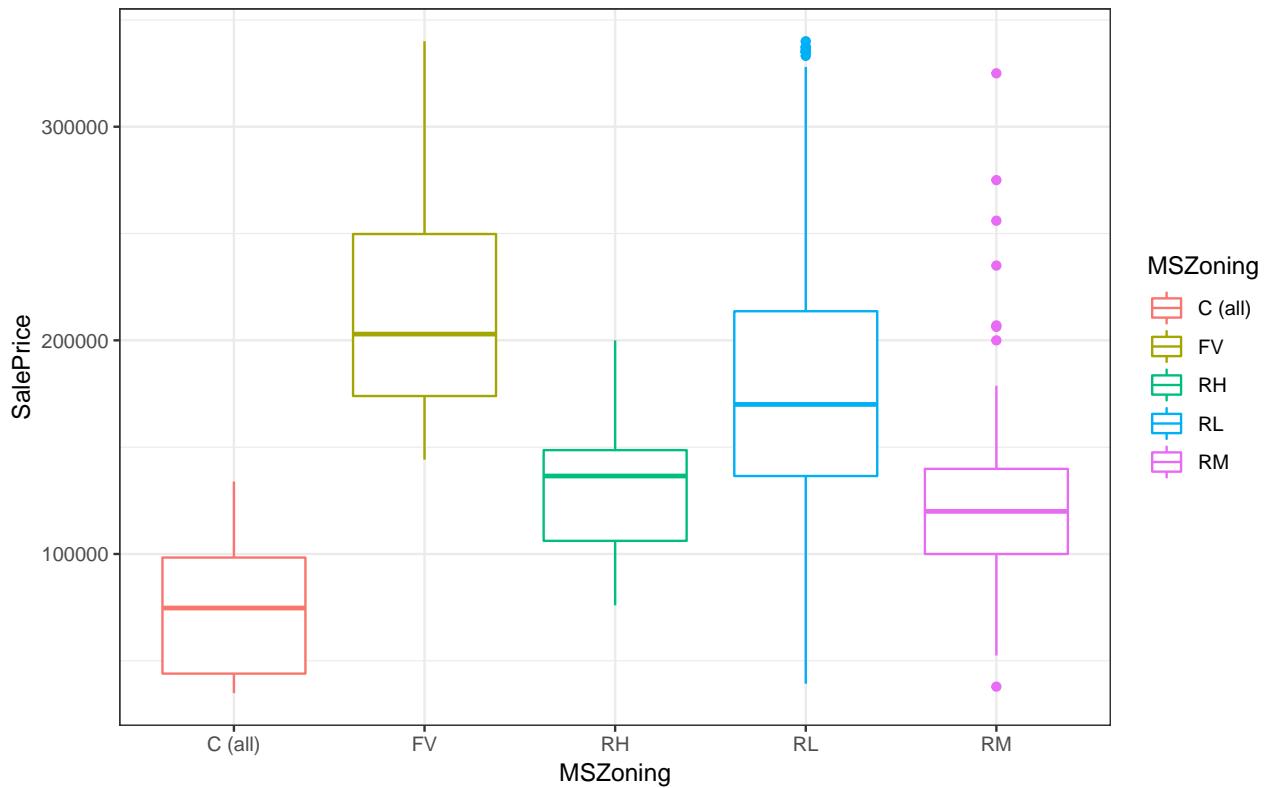
```



```
ggplot(data = datosAnova, aes(x = BldgType , y = SalePrice, color = BldgType)) +
  geom_boxplot() +
  theme_bw()
```



```
ggplot(data = datosAnova, aes(x = MSZoning , y = SalePrice, color = MSZoning )) +
  geom_boxplot() +
  theme_bw()
```



Vamos a ejecutar el modelo de anova :

```
# Utilizamos la función aov de R
modelo.aov.neig <- aov(SalePrice ~ Neighborhood, data = datosAnova, qr=T, projections=T)
summary(modelo.aov.neig)

##          Df      Sum Sq   Mean Sq F value    Pr(>F)
## Neighborhood  24 2647805878733 110325244947   67.97 <0.0000000000000002
## Residuals    1367 2218874158583   1623170562
##
## Neighborhood ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

modelo.aov.Bldg <- aov(SalePrice ~ BldgType, data = datosAnova, qr = T, projections =T)
summary(modelo.aov.Bldg)

##          Df      Sum Sq   Mean Sq F value    Pr(>F)
## BldgType      4 194677874173 48669468543   14.45 0.0000000000146 ***
## Residuals    1387 4672002163143  3368422612
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

modelo.aov.zone <- aov(SalePrice ~ MSZoning, data = datosAnova, qr = T, projections=T)
summary(modelo.aov.zone)
```

```

##          Df      Sum Sq   Mean Sq F value    Pr(>F)
## MSZoning     4  793604221876 198401055469   67.56 <0.0000000000000002
## Residuals 1387  4073075815441   2936608375
##
## MSZoning ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Interpretación :

En el gráfico y según el resultado del modelo de ANOVA se queda demostrado que el precio medio varia mucho entre los distintos barrios.

En el gráfico tb que el precio medio varia tb en función el tipo de vivienda y de tipo de zona en el que se ubique.

Los resultados del modelo : Sum Sq: Suma de los cuadrados Mean SQ: Media de los cuadrados F: El valor si es grande sugiere que hay bastante diferencia entre las medias según el el barrio o el tipo de zona. por tanto tambien nos ayuda a descartar la hipótesis nula a favor de la hipótesis alternativa. Pr (>0): Puesto que el valor p es menor que el nivel de significancia de 0.05 se puede rechazar la hipótesis nula a favor de la alternativa y determinar por tanto , que dependiendo del nivel de vecindario el precio medio puede ser diferente.

No necesariamente todas las medias deben ser distintas, pero si que hay diferencias entre las medias de algunos vecindarios.

Ocurre lo mismo por tipo y por zona, el test de Anova nos confirma que las medias son diferentes.

ANOVA no paramétrico :

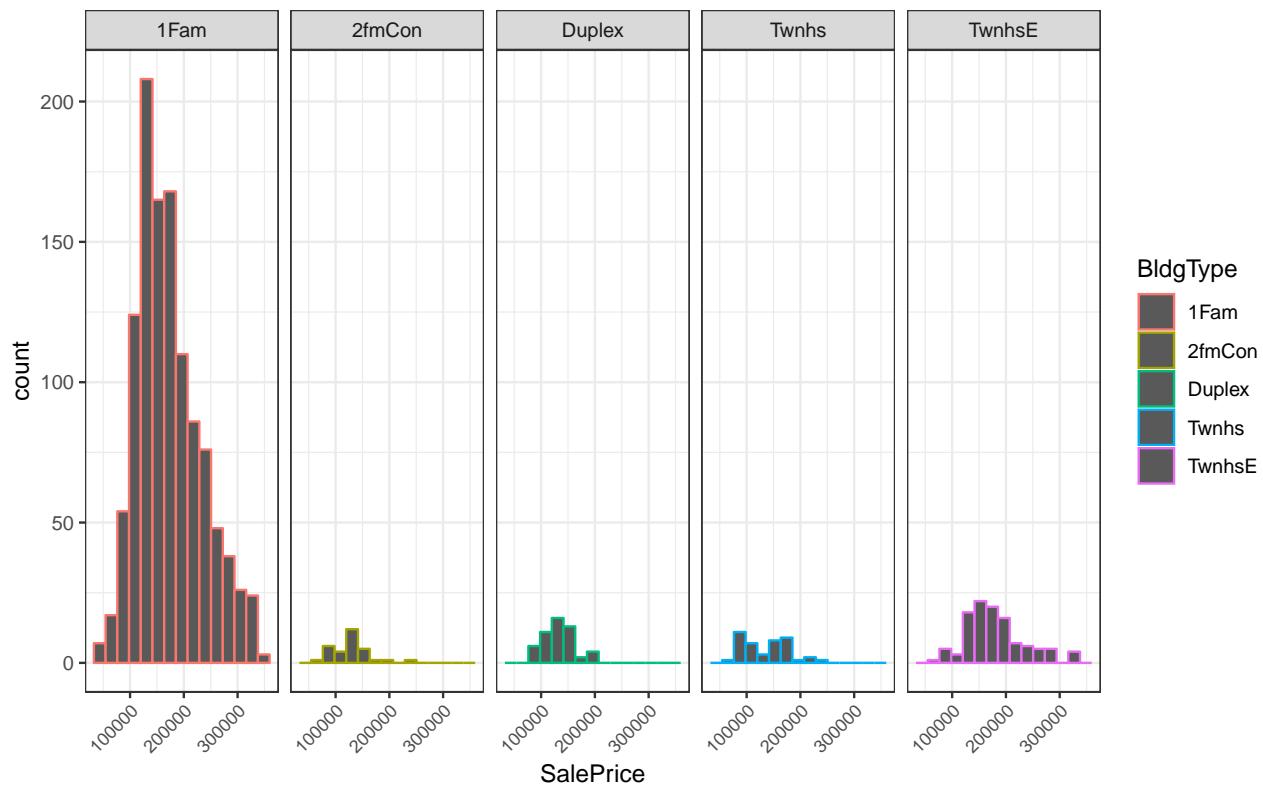
Con este test se pretende demostrar que las medianas son diferentes (hipótesis alternativa)

Con el siguiente gráfico se observa que los distintos tipos de vivienda siguen una distribución asimétrica aunque la dirección parece la misma

```

ggplot(data =houses , mapping = aes(x = SalePrice, colour = BldgType)) +
  geom_histogram(bins = 15) +
  theme_bw() + facet_grid(. ~ BldgType) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 8, hjust = 1))

```



Además las medianas no son iguales

```
datosAgrupadosMediana <- aggregate(SalePrice ~ BldgType , data = datosAnova, FUN = median)
datosAgrupadosMediana
```

```
##   BldgType SalePrice
## 1     1Fam 163700
## 2    2fmCon 127500
## 3    Duplex 135980
## 4    Twnhs 137500
## 5   TwnhsE 171825
```

Por lo que aplicamos un anova no paramétrico:

```
modelo.KT <- kruskal.test(SalePrice ~ BldgType, data = houses)
modelo.KT
```

```
##
## Kruskal-Wallis rank sum test
##
## data: SalePrice by BldgType
## Kruskal-Wallis chi-squared = 63.229, df = 4, p-value =
## 0.0000000000006074
```

Interpretación :

El p-value al ser tan bajo, menor a 0,05 , si consideramos un nivel de 95% de confianza.. indica que el test encuentra significancia en la diferencia entre grupos. Es decir que nos decantamos por la hipótesis alternativa de que las medianas son diferentes y por tanto las muestras no provienen de la misma distribución poblacional. El valor df , indica que en al menos 4 grupos ve diferencia en las medianas..

ANOVA Multifactorial

```
# Modelo aditivo
modelo.aov.multifa<- aov(SalePrice ~ BldgType + MSZoning, data = houses)
summary(modelo.aov.multifa)

##          Df     Sum Sq   Mean Sq F value    Pr(>F)
## BldgType      4 194677874173 48669468543 17.03 < 0.00000000000000122
## MSZoning      4 718820365120 179705091280 62.87 < 0.0000000000000002
## Residuals 1383 3953181798023 2858410555
##
## BldgType *** 
## MSZoning ***
## Residuals
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Modelo con interacción:
modelo.aov.multifi<- aov(SalePrice ~ BldgType * MSZoning, data = houses)
summary(modelo.aov.multifi)

##          Df     Sum Sq   Mean Sq F value    Pr(>F)
## BldgType      4 194677874173 48669468543 17.495
## MSZoning      4 718820365120 179705091280 64.599
## BldgType:MSZoning 10 133689014944 133689014944 4.806
## Residuals 1373 3819492783079 2781859274
##
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

modelo.aov.multif<- aov(SalePrice ~ BldgType + Neighborhood + MSZoning + Heating +
                           CentralAir, data = houses)
summary(modelo.aov.multif)

##          Df     Sum Sq   Mean Sq F value    Pr(>F)
## BldgType      4 194677874173 48669468543 33.553 < 0.0000000000000002
## Neighborhood 24 2586350697592 107764612400 74.293 < 0.0000000000000002
## MSZoning      4  31542093456  7885523364  5.436       0.000242
## Heating        5  39809154136  7961830827  5.489       0.000052151
## CentralAir     1  51725154453  51725154453 35.659       0.000000003
## Residuals 1353 1962575063507 1450535893
##
## BldgType *** 
## Neighborhood ***
## MSZoning ***
## Heating ***
## CentralAir ***
## Residuals
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

El resultado nos indica que la interacción BldgType:MSZoning tiene un p-valor menor que 0.05 con lo que

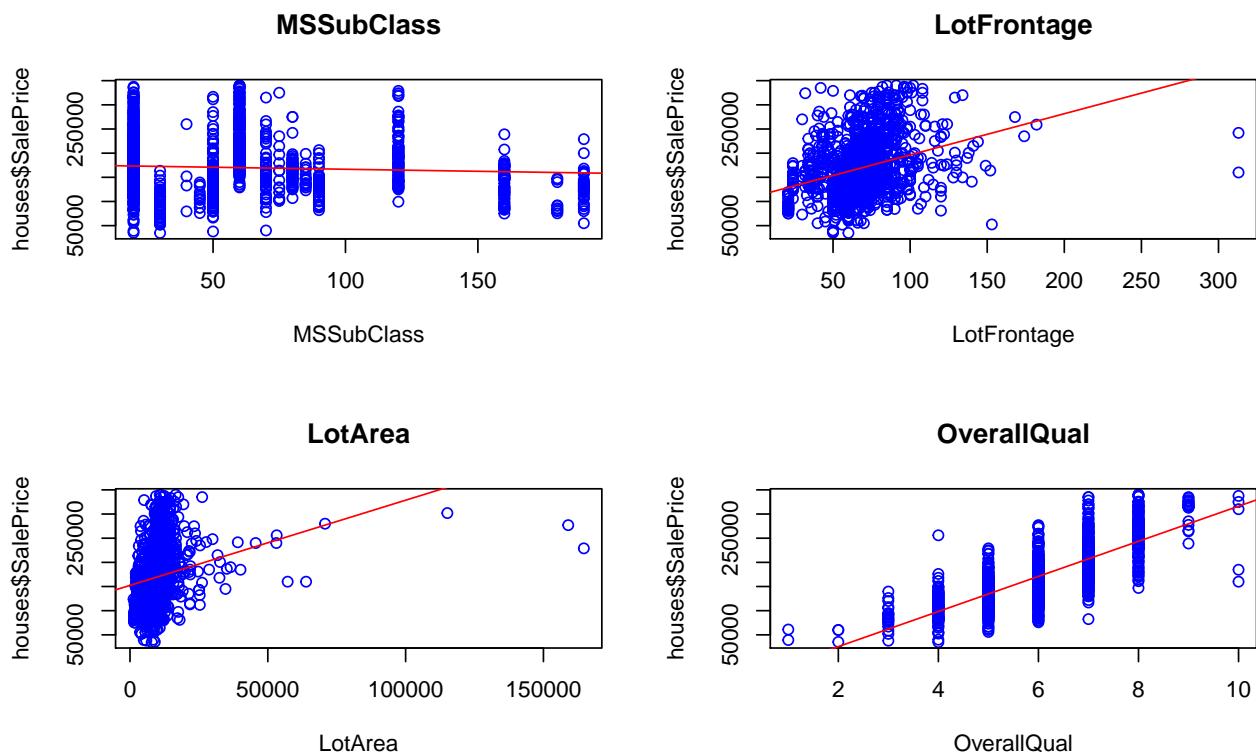
parece que si existe interacción entre ambas variables. Y ambas variables son explicativas del salario medio. Cuando se hace un anova multifactorial aditivo por distintas variables cualitativas , se observa que todas ellas , por su nivel de significancia, indica que hay diferencias (al menos en el vecindario hay más de 24 grupos con diferencias, en los etc.)

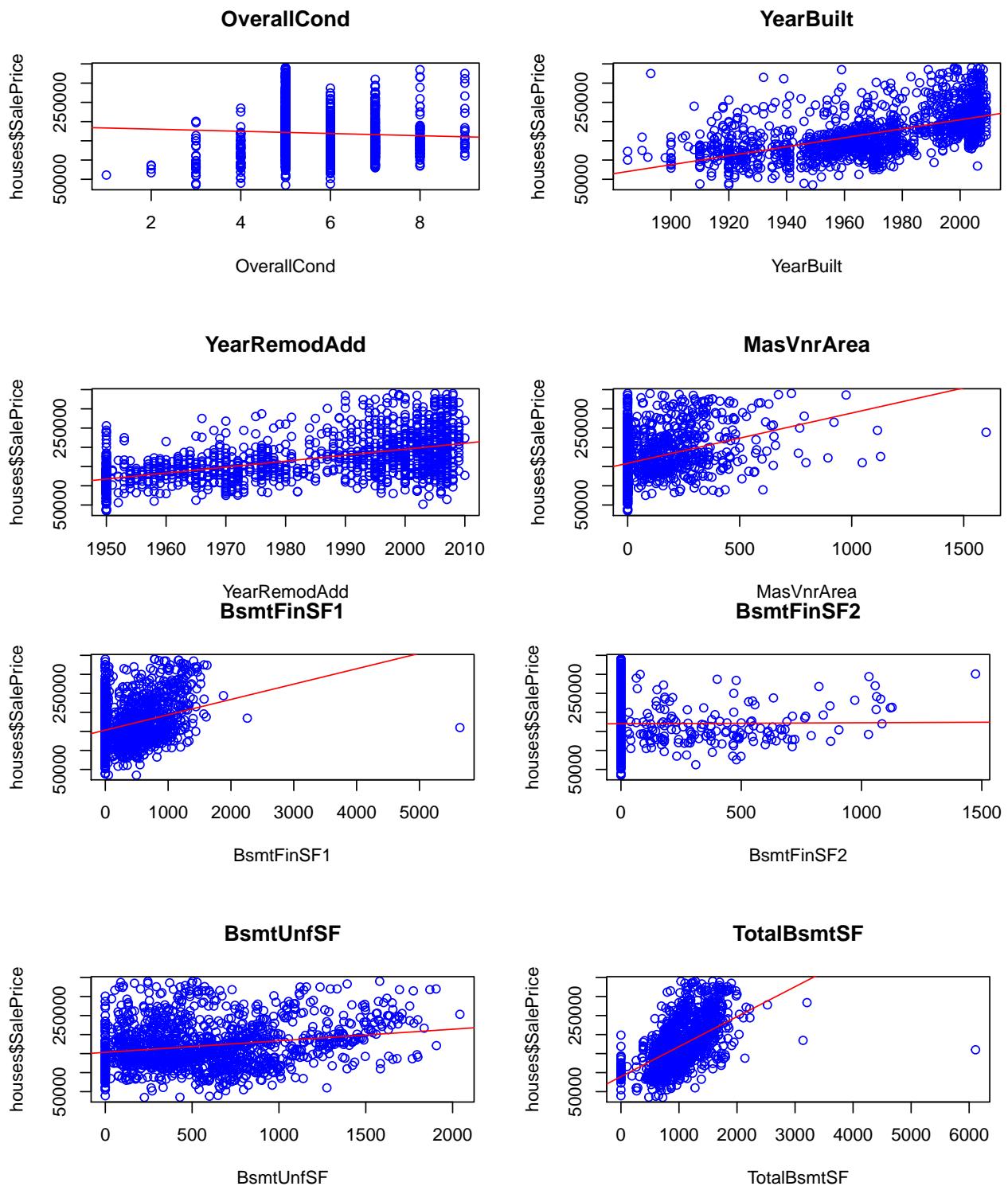
Algoritmo de regresión

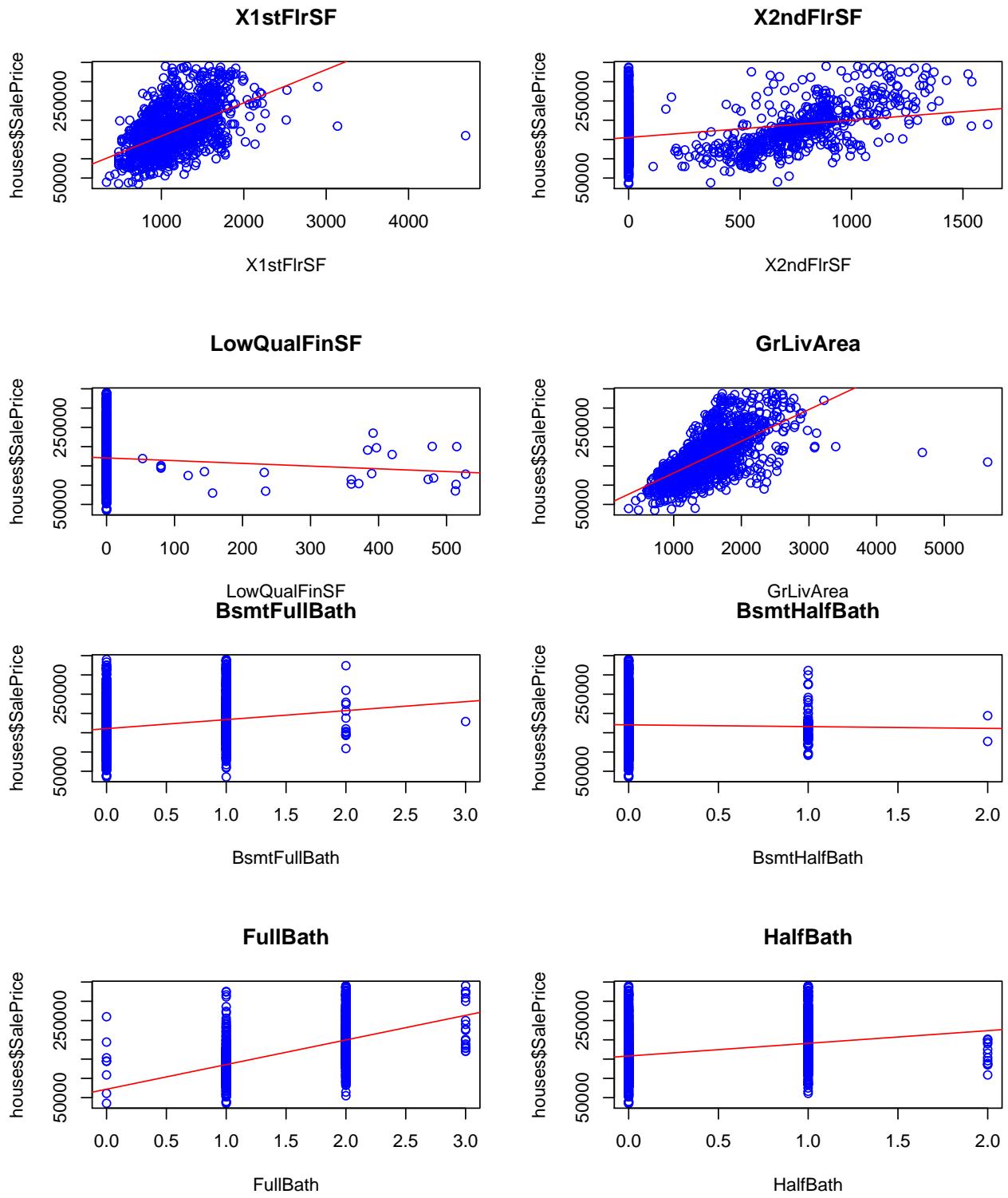
Vamos a llevar a cabo un modelo predictivo para diferentes variables. El objetivo es ver como impactan estas en el precio de la propiedad. Para ello dividimos el dataset en un conjunto train sobre el que entrenaremos el modelo de regresión lineal partiendo de variable numéricas y un conjunto de test sobre el que llevaremos la predicción del modelo. Se calculará el error cuadrático medio para medir la cantidad de error entre los dos conjuntos, comparando el valor predicho y el valor conocido.

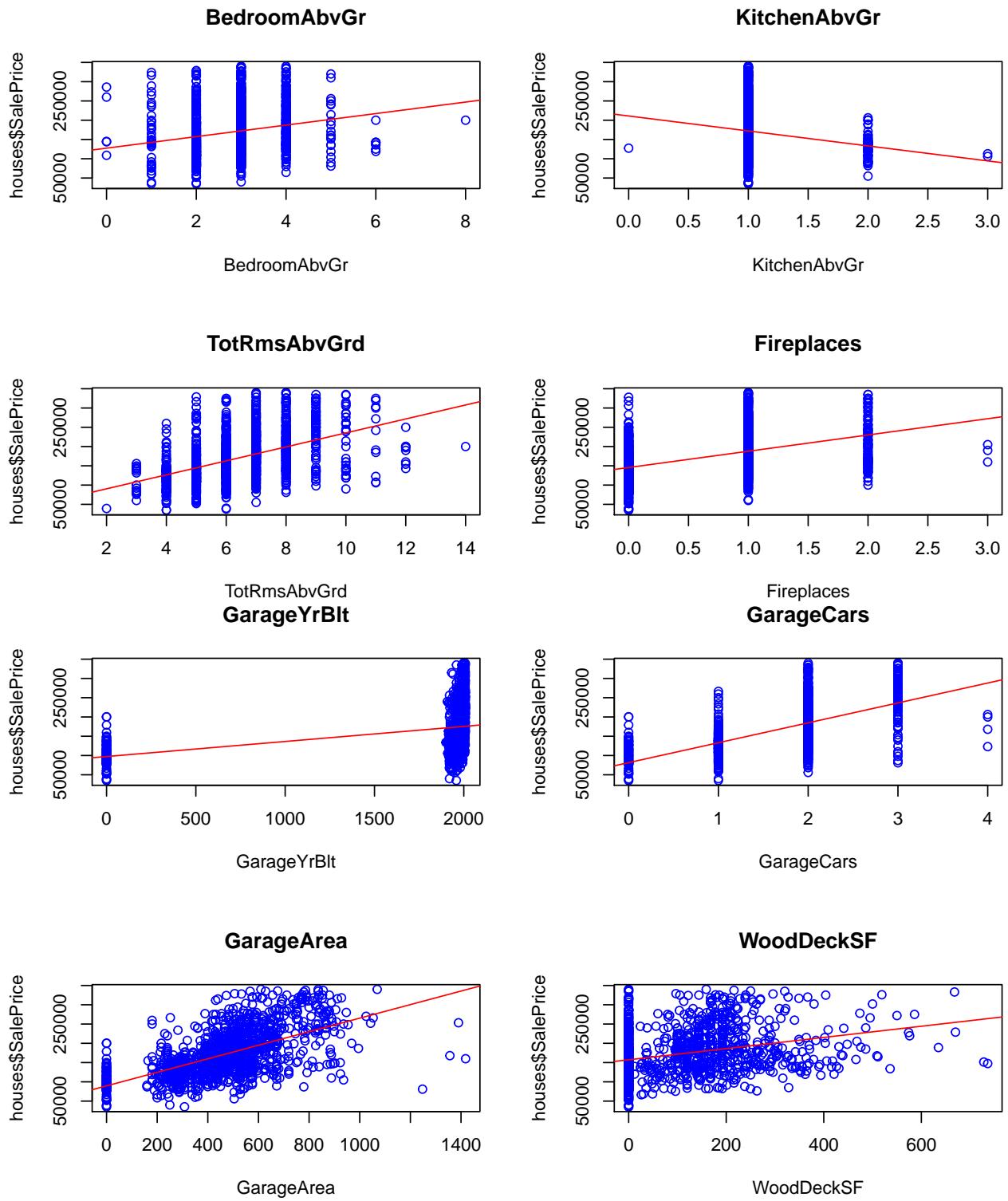
En primer lugar se muestran las gráficas del modelo de regresión lineal para todo el dataset.

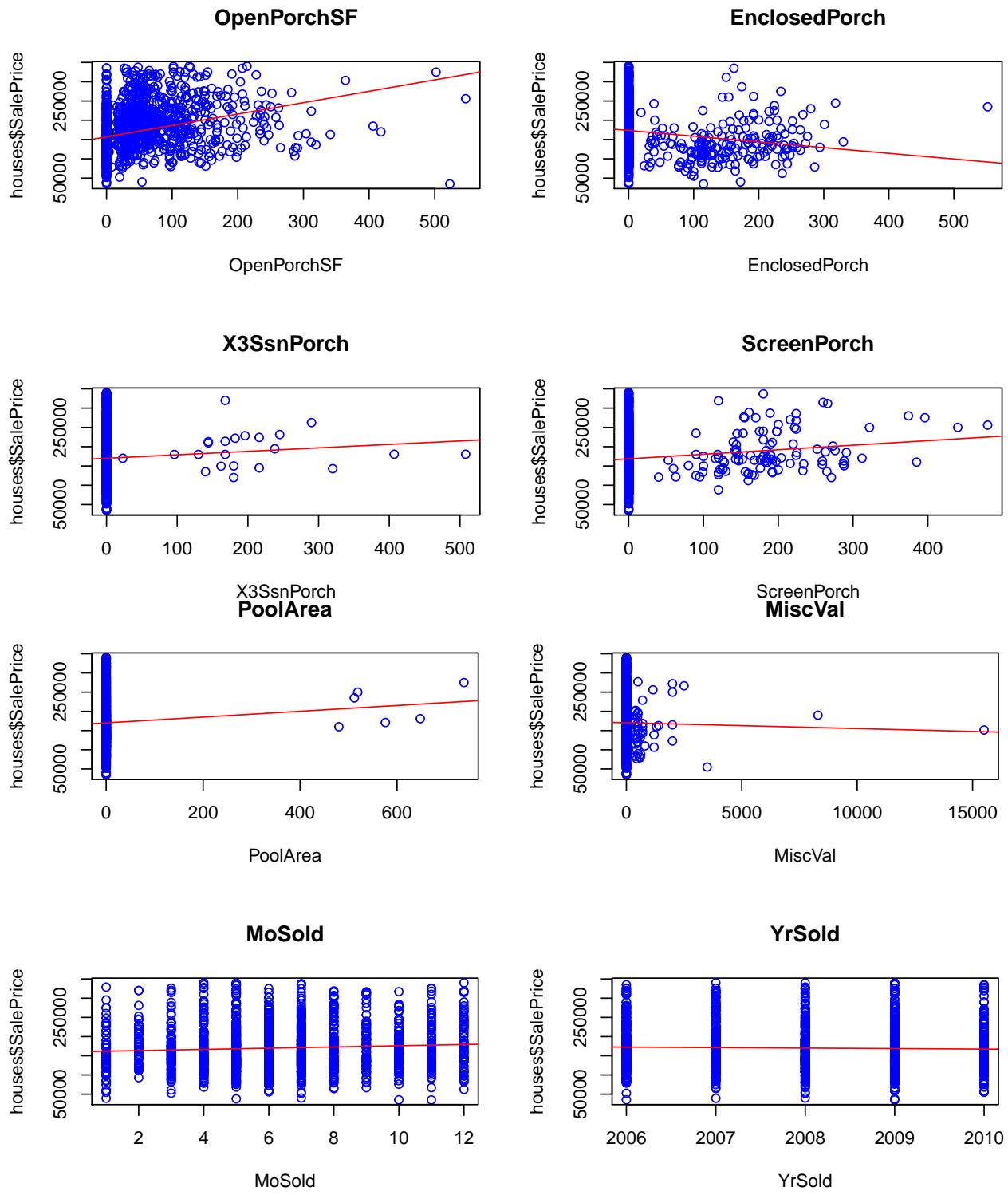
```
par(mfrow=c(2,2))
colnames <- dimnames(data)[[2]]
for (i in 2:(ncol(houses)-1)) {
  if (is.numeric(houses[,i]) == "TRUE"){
    plot(houses$SalePrice ~ houses[,i],main=names(houses)[i],
         xlab=names(houses)[i],col="blue")
    reg_line <- lm(houses$SalePrice ~ houses[,i])
    abline(reg_line,col="red")
  }
}
```

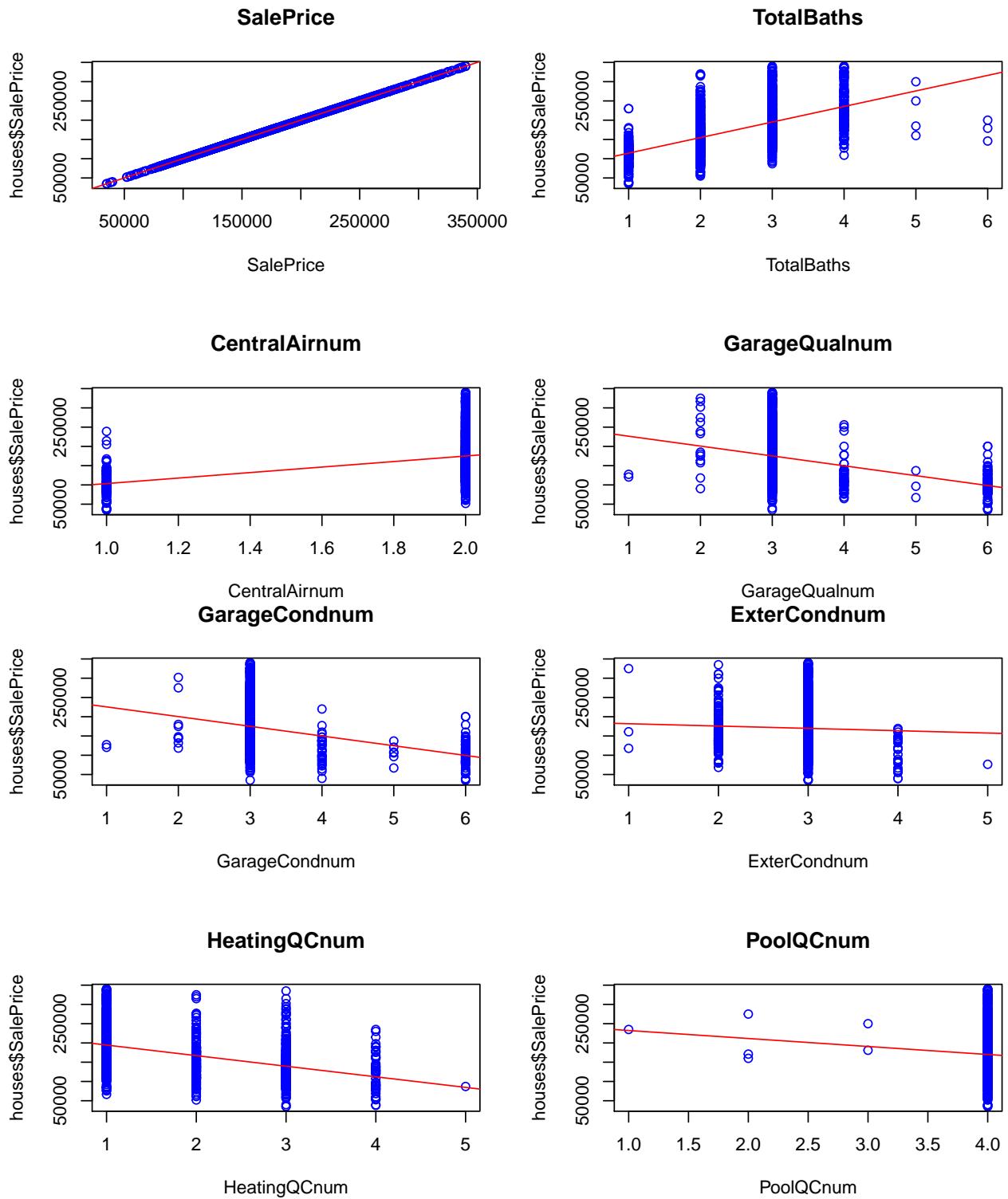












A continuación se divide el dataset en el conjunto de entrenamiento y el conjunto de test.

```
partrain <- createDataPartition(y = houses$SalePrice,
                                p = 0.80,
                                list = FALSE)

trainh <- houses[partrain, ]
```

```
testh <- houses[-partrain, ]
```

Guardamos las variables target

```
train_y <- houses$SalePrice  
test_y <- houses$SalePrice
```

Revisamos el modelo de regresión lineal sobre algunas de las variables que más correlacionan con la variable target.

```
modelh1 <- lm(SalePrice ~ GrLivArea, data = trainh)
```

```
kable(summary(modelh1)$coef, digits=4)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	51228.7881	4266.9984	12.0058	0
GrLivArea	80.9761	2.7642	29.2943	0

```
predh1 <- predict(modelh1, testh)
```

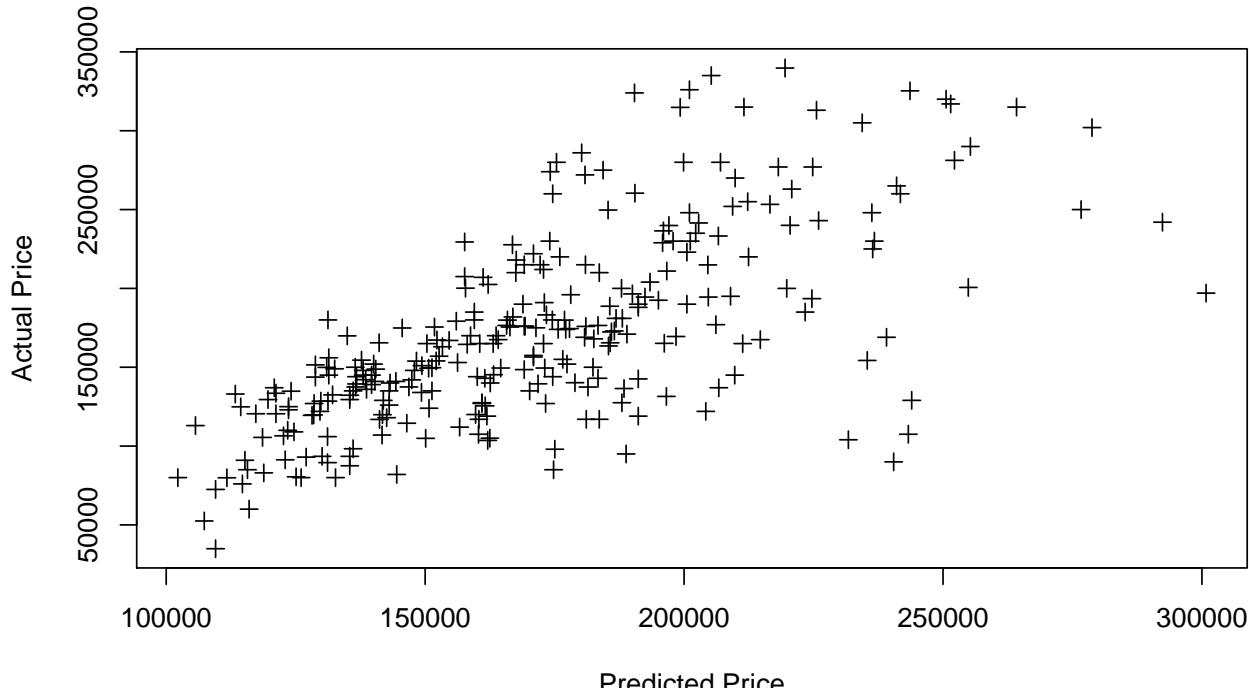
```
rmseh1<-sqrt(sum( (log(predh1)-log(testh$SalePrice))^2, na.rm=TRUE)/length(predh1))
```

```
rmseh1
```

```
## [1] 0.2647652
```

Con la predicción solo con el atributo GrLivArea se obtiene un error cuadrático medio elevado. Si lo visualizamos gráficamente

```
plot(predh1,testh$SalePrice,xlab="Predicted Price",ylab="Actual Price",pch = 3)
```



```
modelh2 <- lm(SalePrice ~ X1stFlrSF, data = trainh)
```

```
kable(summary(modelh2)$coef, digits=4)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	74619.802	4907.6611	15.2048	0
X1stFlrSF	84.675	4.1451	20.4276	0

```

predh2 <- predict(modelh2, testh)

rmseh2<-sqrt(sum((log(predh2)-log(testh$SalePrice))2,na.rm=TRUE)/length(predh2))

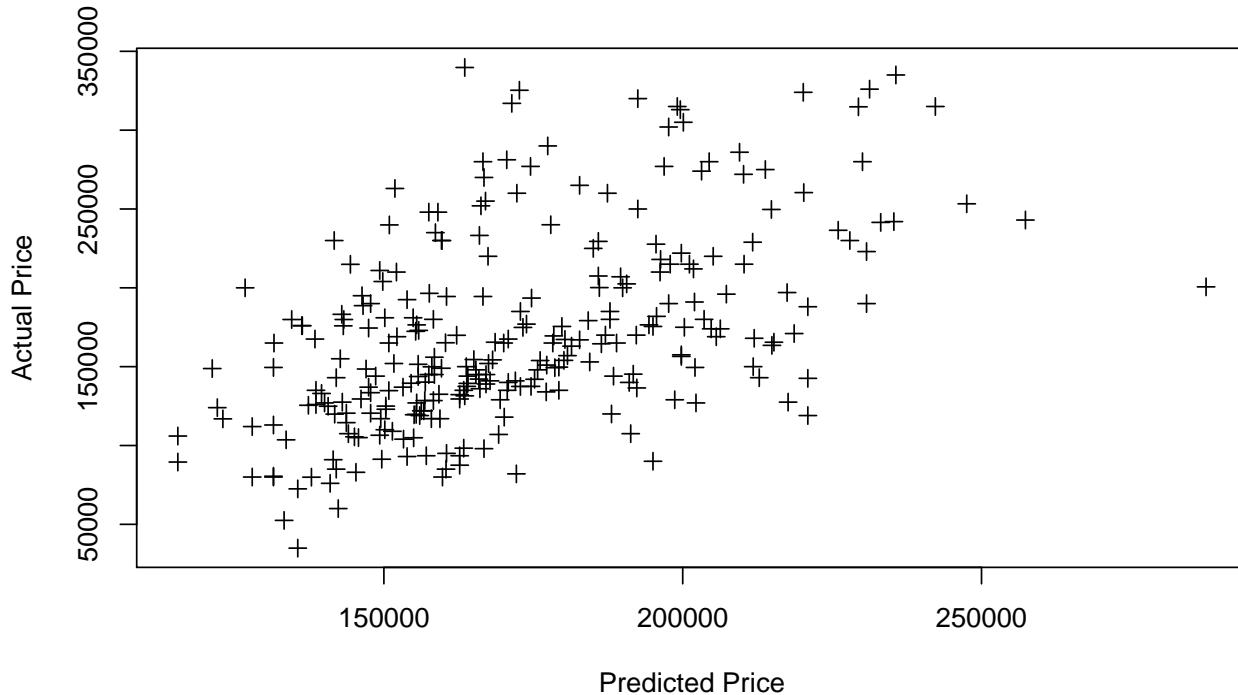
rmseh2

## [1] 0.3117367

```

Con la predicción solo con el atributo X1stFlrSF se obtiene un error cuadrático medio más elevado que el anterior. Si lo visualizamos gráficamente

```
plot(predh2,testh$SalePrice,xlab="Predicted Price",ylab="Actual Price",pch = 3)
```



```
modelh3 <- lm(SalePrice ~ YearBuilt, data = trainh)
```

```
kable(summary(modelh3)$coef, digits=4)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2059443.86	95743.0134	-21.5101	0
YearBuilt	1131.21	48.5729	23.2889	0

```
predh3 <- predict(modelh3, testh)
```

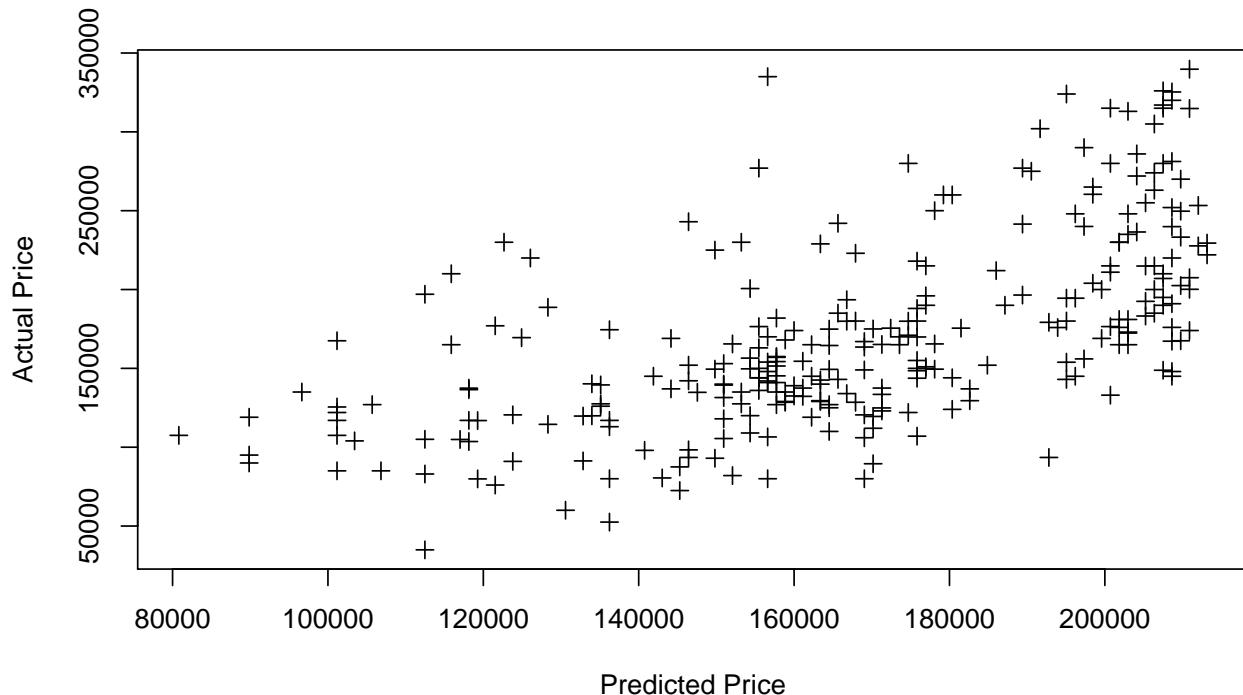
```
rmseh3<-sqrt(sum((log(predh3)-log(testh$SalePrice))2,na.rm=TRUE)/length(predh3))
```

```
rmseh3
```

```
## [1] 0.286921
```

Con la predicción con el atributo YearBuilt se obtiene un error cuadrático todavía mayor. Probamos modelo de predicción con las variables que correlacionan obtenidas en el estudio anterior. Si lo visualizamos gráficamente

```
plot(predh3,testh$SalePrice,xlab="Predicted Price",ylab="Actual Price",pch = 3)
```



```
model_lm <- lm(SalePrice ~ YearBuilt + Fireplaces + TotalBsmtSF + GrLivArea + FullBath +  
BedroomAbvGr + TotRmsAbvGrd + OpenPorchSF + GarageCars + GarageArea +  
YearRemodAdd + BsmtFullBath + BsmtFinSF1 + X1stFlrSF, data = trainh )
```

```
kable(summary(model_lm)$coef, digits=4)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1710529.6282	111946.7818	-15.2798	0.0000
YearBuilt	387.4875	44.4409	8.7192	0.0000
Fireplaces	14203.3807	1629.6613	8.7155	0.0000
TotalBsmtSF	18.7325	3.9424	4.7516	0.0000
GrLivArea	38.2769	4.1687	9.1821	0.0000
FullBath	8082.8348	2536.2698	3.1869	0.0015
BedroomAbvGr	-1112.7409	1645.9969	-0.6760	0.4992
TotRmsAbvGrd	1380.9884	1203.3278	1.1476	0.2514
OpenPorchSF	33.1609	14.8594	2.2316	0.0258
GarageCars	9713.1799	2760.8694	3.5182	0.0005
GarageArea	24.0906	9.6013	2.5091	0.0122
YearRemodAdd	496.9917	56.7601	8.7560	0.0000
BsmtFullBath	9913.2665	2339.0744	4.2381	0.0000
BsmtFinSF1	-1.3455	3.0515	-0.4409	0.6593
X1stFlrSF	-3.4987	4.5009	-0.7773	0.4371

```
summary(model_lm)
```

```
##  
## Call:  
## lm(formula = SalePrice ~ YearBuilt + Fireplaces + TotalBsmtSF +  
##     GrLivArea + FullBath + BedroomAbvGr + TotRmsAbvGrd + OpenPorchSF +  
##     GarageCars + GarageArea + YearRemodAdd + BsmtFullBath + BsmtFinSF1 +
```

```

##      X1stFlrSF, data = trainh)
##
## Residuals:
##   Min     1Q Median     3Q    Max
## -367014 -15583 -1466 15082 110467
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1710529.628 111946.782 -15.280 < 0.0000000000000002 ***
## YearBuilt      387.487   44.441   8.719 < 0.0000000000000002 ***
## Fireplaces     14203.381  1629.661   8.716 < 0.0000000000000002 ***
## TotalBsmtSF     18.732    3.942   4.752     0.00000229 ***
## GrLivArea      38.277    4.169   9.182 < 0.0000000000000002 ***
## FullBath       8082.835  2536.270   3.187    0.001479 **
## BedroomAbvGr   -1112.741 1645.997  -0.676    0.499165
## TotRmsAbvGrd   1380.988  1203.328   1.148    0.251366
## OpenPorchSF     33.161    14.859   2.232    0.025839 *
## GarageCars      9713.180  2760.869   3.518    0.000452 ***
## GarageArea      24.091    9.601   2.509    0.012247 *
## YearRemodAdd    496.992   56.760   8.756 < 0.0000000000000002 ***
## BsmtFullBath    9913.267  2339.074   4.238    0.00002443 ***
## BsmtFinSF1      -1.345    3.051  -0.441    0.659349
## X1stFlrSF       -3.499    4.501  -0.777    0.437127
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30000 on 1101 degrees of freedom
## Multiple R-squared:  0.7433, Adjusted R-squared:  0.7401
## F-statistic: 227.7 on 14 and 1101 DF,  p-value: < 0.0000000000000022

```

El modelo con las principales variables tratadas puede explicar casi el 75%, un 74,57% de la varianza observada. El p-value del modelo es significativo < 0.0000000000000022 por lo que se puede concluir que existe una relación entre estas variables tratadas como predictoras y la variable target, precio de la vivienda.

```

pred_lm <- predict(model_lm, testh)

rmse_lm<-sqrt(sum((log(pred_lm)-log(testh$SalePrice))^-2,na.rm=TRUE)/length(pred_lm))

c(RMSE = rmse_lm, R2 = summary(model_lm)$r.squared)

```

```

##      RMSE      R2
## 0.1601480 0.7433212

```

Mostramos las primeras observaciones y comparamos el precio real y el predicho con regresión lineal

```

table_lm <- data.frame(x = pred_lm*10, y = testh$SalePrice)
names(table_lm) <- c("Predicted_Price", ylab = "Actual_Price")
head(table_lm)

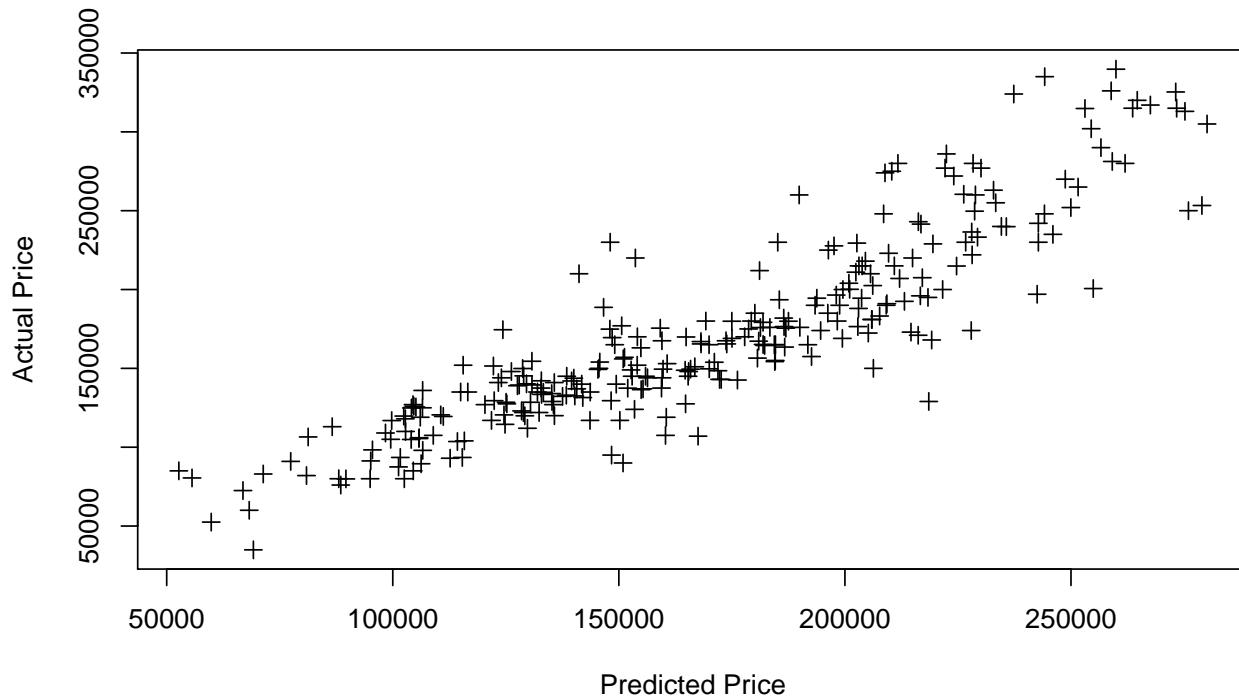
```

```

##      Predicted_Price Actual_Price
## 6          1722786     143000
## 11         1224308     129500
## 15         1512474     157000
## 17         1527101     149000
## 21         2731600     325300
## 27         1332998     134800

```

```
plot(pred_lm,testh$SalePrice,xlab="Predicted Price",ylab="Actual Price",pch = 3)
```



Regresión lineal multiple con variables cuantitativas y cualitativas

Probamos un lm con dos variables numericas y dos factors (las que hemos visto que tienen mayor relevancia para influir en el precio)

```
recta.regresion.multiple <- lm(SalePrice~GrLivArea+GarageCars+MSZoning+BldgType+
                               YearBuilt,houses, na.action = na.exclude,method="qr")
# Se resume los resultados del modelo.
summary(recta.regresion.multiple)
```

```
##
## Call:
## lm(formula = SalePrice ~ GrLivArea + GarageCars + MSZoning +
##     BldgType + YearBuilt, data = houses, na.action = na.exclude,
##     method = "qr")
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -300720 -18694   -2775   14729  136130
##
## Coefficients:
##             Estimate Std. Error t value            Pr(>|t|)
## (Intercept) -1314126.554  77419.186 -16.974 < 0.0000000000000002 ***
## GrLivArea       61.205     1.981  30.900 < 0.0000000000000002 ***
## GarageCars     17674.780   1510.670  11.700 < 0.0000000000000002 ***
## MSZoningFV     49560.334  11014.790   4.499          0.000007386 ***
## MSZoningRH     34822.589  12754.741   2.730          0.00641 **
## MSZoningRL     44123.279  10136.915   4.353          0.000014436 ***
## MSZoningRM     31608.630  10218.901   3.093          0.00202 **
```

```

## BldgType2fmCon    -8894.417    5878.559   -1.513           0.13050
## BldgTypeDuplex   -42580.165    4476.794   -9.511 < 0.0000000000000002 ***
## BldgTypeTwnhs   -27329.740    5246.444   -5.209           0.000000218 ***
## BldgTypeTwnhsE  -4116.696     3442.229   -1.196           0.23193
## YearBuilt        672.339      40.300    16.684 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31410 on 1380 degrees of freedom
## Multiple R-squared:  0.7202, Adjusted R-squared:  0.7179
## F-statistic: 322.8 on 11 and 1380 DF,  p-value: < 0.0000000000000002

```

El algoritmo de regresión lineal indica que la recta tiene una formula en la que las variables que más influyen es el Área Habitable , y también el n° de plazas de garage y el año de contrucción, mientras que si se trata de un duplex, contribuye pero para decrementar el precio (idem para Twnhs)

Precio de una casa en función del modelo LM multiple para una casa de las siguientes características

```
predict.lm(recta.regresion.multiple,data.frame(GrLivArea = 45000,GarageCars = 2,  
MSZoning = "RL" , BldgType = "Twnhs" , YearBuilt = 2000))
```

1
2836908

```
predict.lm(recta.regresion.multiple,data.frame(GrLivArea = 45000,GarageCars =2,  
MSZoning ="RL" , BldgType = "TwnhsE", YearBuilt = 2000))
```

1
2860121

```
predict.lm(recta.regresion.multiple,data.frame(GrLivArea = 15000, GarageCars = 2,  
                                              MSZoning = "RM" , BldgType = "Duplex", YearBuilt = 1999))
```

1
972328.3

```
predict.lm(recta.regresion.multiple,data.frame(GrLivArea = 15000,GarageCars = 4,  
MSZoning ="FV" , BldgType = "1Fam", YearBuilt = 2009))
```

1
1074933

Otro algoritmo.

Probamos con un algoritmo de random forest.

```
suppressMessages(library(randomForest))
set.seed(1234)
model_rf <- randomForest(SalePrice ~ YearBuilt + Fireplaces + TotalBsmtSF +
                           GrLivArea + FullBath + BedroomAbvGr + TotRmsAbvGrd +
                           OpenPorchSF + GarageCars + GarageArea + YearRemodAdd +
                           BsmtFullBath + BsmtFinSF1 + X1stFlrSF, data = trainh)
model_rf

## 
## Call:
## randomForest(formula = SalePrice ~ YearBuilt + Fireplaces + TotalBsmtSF +      GrLivArea + FullBath
##                 Type of random forest: regression
```

```

##                               Number of trees: 500
## No. of variables tried at each split: 4
##
##                               Mean of squared residuals: 588769842
##                                         % Var explained: 82.98
summary(model_rf)

##                                Length Class  Mode
## call                      3   -none- call
## type                      1   -none- character
## predicted                 1116  -none- numeric
## mse                       500   -none- numeric
## rsq                        500   -none- numeric
## oob.times                  1116  -none- numeric
## importance                14    -none- numeric
## importanceSD              0    -none- NULL
## localImportance            0    -none- NULL
## proximity                 0    -none- NULL
## ntree                      1    -none- numeric
## mtry                      1    -none- numeric
## forest                     11   -none- list
## coefs                      0    -none- NULL
## y                          1116  -none- numeric
## test                      0    -none- NULL
## inbag                      0    -none- NULL
## terms                      3    terms  call

set.seed(12345)
pred_rf <- predict(model_rf, testh)

rmse_rf<-sqrt(sum((log(pred_rf)-log(testh$SalePrice))2,na.rm=TRUE)/length(pred_rf))

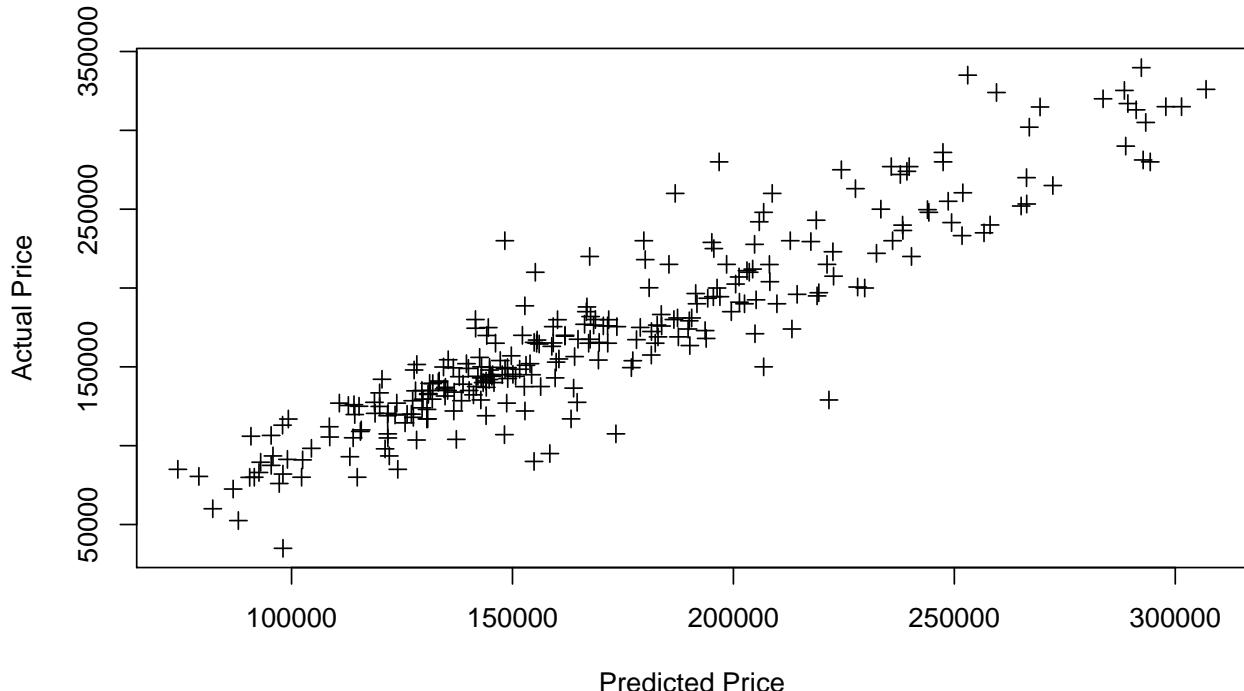
R2 = summary(model_rf$r.squared)
c(RMSE = rmse_rf)

##      RMSE
## 0.1531514

Se aprecia que el error cuadrático medio es inferior para random forest, además que la varianza explicada es algo superior con random forest.

plot(pred_rf,testh$SalePrice,xlab="Predicted Price",ylab="Actual Price",pch = 3)

```



Mostramos las primeras observaciones y comparamos el precio real y el predicho con random forest.

```
table_rf <- data.frame(x = pred_rf*10, y = testh$SalePrice)
names(table_rf) <- c("Predicted_Price", ylab = "Actual_Price")
head(table_rf)

##      Predicted_Price Actual_Price
## 6          1596603     143000
## 11         1297420     129500
## 15         1497376     157000
## 17         1400236     149000
## 21         2884863     325300
## 27         1280564     134800

Linear_Model <- c(RMSE = rmse_lm, R2 = summary(model_lm)$r.squared)
Random_Forest_Model <- c(RMSE = rmse_rf, pseudoR2 = mean(model_rf$rsq))
model_comparison <- rbind(Linear_Model, Random_Forest_Model)
model_comparison

##                               RMSE           R2
## Linear_Model       0.1601480 0.7433212
## Random_Forest_Model 0.1531514 0.8229264
```

Se puede ver que con ramdon forest se explica casi el 83% de la varianza, es decir algo más que con el modelo de regresión lineal, teniendo el modelo de random forest un menor error cuadrático medio, un mayor coeficiente R2 que explica la proporción de varianza de la variable SalePrice, de los precios de las viviendas, de acuerdo a las variables seleccionadas en base al estudio previo de las relaciones entre estas.

Grabación Fichero preprocessado

Tras el análisis llevado a cabo, nos guardamos un juego de datos con aquellos atributos relevantes con el objetivo de poder ser utilizado en futuros estudios.

```

varCualitativas_borrar <- which(names == "Street" | names == "Alley"
| names == "LotShape" | names == "LandContour"
| names == "Utilities" | names == "LotConfig"
| names == "LandSlope" | names == "Condition2"
| names == "HouseStyle" | names == "RoofStyle"
| names == "RoofMatl" | names == "Exterior1st"
| names == "Exterior2nd" | names == "Foundation"
| names == "MasVnrType" | names == "ExterQual"
| names == "ExterCond" | names == "BsmtFinType2"
| names == "BsmtQual" | names == "BsmtCond"
| names == "BsmtExposure" | names == "BsmtFinType1"
| names == "HeatingQC" | names == "KitchenQual"
| names == "Functional" | names == "FireplaceQu"
| names == "GarageType" | names == "GarageFinish"
| names == "GarageQual" | names == "GarageCond"
| names == "PavedDrive" | names == "PoolQC"
| names == "Fence" | names == "MiscFeature"
| names == "SaleType" | names == "SaleCondition")

housesprepro<-houses[, -varCualitativas_borrar]
names<-colnames(housesprepro)

varCuantitativas_borrar <- which(names == "MSSubClass" | names == "LotArea"
| names == "OverallCond" | names == "MasVnrArea"
| names == "BsmtFinSF2" | names == "BsmtUnfSF"
| names == "X2ndFlrSF" | names == "LowQualFinSF"
| names == "HalfBath" | names == "KitchenAbvGr"
| names == "BsmtHalfBath" | names == "GarageYrBlt"
| names == "EnclosedPorch" | names == "X3SsnPorch"
| names == "ScreenPorch" | names == "MiscVal"
| names == "PoolArea"
| names == "BsmtQualnum"
| names == "GarageQualnum" | names == "GarageCondnum"
| names == "ExterCondnum" | names == "PoolQcnum" )

```

```
housesprepro2<-housesprepro[, -varCuantitativas_borrar]
```

```
str(housesprepro2)
```

```

## 'data.frame': 1392 obs. of 32 variables:
## $ Id      : int 1 2 3 4 5 6 7 8 9 10 ...
## $ MSZoning: Factor w/ 5 levels "C (all)", "FV", ...: 4 4 4 4 4 4 4 4 5 4 ...
## $ LotFrontage: num 65 80 68 60 84 85 75 80 51 50 ...
## $ Neighborhood: Factor w/ 25 levels "Blmgtn", "Blueste", ...: 6 25 6 7 14 12 21 17 18 4 ...
## $ Condition1: Factor w/ 9 levels "Artery", "Feedr", ...: 3 2 3 3 3 3 3 5 1 1 ...
## $ BldgType: Factor w/ 5 levels "1Fam", "2fmCon", ...: 1 1 1 1 1 1 1 1 2 ...
## $ OverallQual: int 7 6 7 7 8 5 8 7 7 5 ...
## $ YearBuilt: int 2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
## $ YearRemodAdd: int 2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
## $ BsmtFinSF1: num 706 978 486 216 655 ...
## $ TotalBsmtSF: num 856 1262 920 756 1145 ...
## $ Heating: Factor w/ 6 levels "Floor", "GasA", ...: 2 2 2 2 2 2 2 2 2 2 ...

```

```

## $ CentralAir      : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 ...
## $ Electrical     : Factor w/ 5 levels "FuseA","FuseF",...: 5 5 5 5 5 5 5 5 2 5 ...
## $ X1stFlrSF       : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
## $ GrLivArea        : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
## $ BsmtFullBath    : num  1 0 1 1 1 1 1 1 0 1 ...
## $ FullBath         : int  2 2 2 1 2 1 2 2 2 1 ...
## $ BedroomAbvGr    : int  3 3 3 3 4 1 3 3 2 2 ...
## $ TotRmsAbvGrd    : int  8 6 6 7 9 5 7 7 8 5 ...
## $ Fireplaces        : int  0 1 1 1 1 0 1 2 2 2 ...
## $ GarageCars        : int  2 2 2 3 3 2 2 2 2 1 ...
## $ GarageArea         : int  548 460 608 642 836 480 636 484 468 205 ...
## $ WoodDeckSF        : int  0 298 0 0 192 40 255 235 90 0 ...
## $ OpenPorchSF       : int  61 0 42 35 84 30 57 204 0 4 ...
## $ MoSold             : int  2 5 9 2 12 10 8 11 4 1 ...
## $ YrSold              : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
## $ SalePrice            : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
## $ Electrical_imp: logi FALSE FALSE FALSE FALSE FALSE FALSE ...
## $ TotalBaths           : num  4 3 4 2 4 3 3 4 2 2 ...
## $ CentralAirnum        : num  2 2 2 2 2 2 2 2 2 2 ...
## $ HeatingQCnum        : num  1 1 1 2 1 1 1 1 2 1 ...
write.csv(housesprepro2, file="pra2_clean.csv", row.names = FALSE)
write.table(housesprepro2,file="pra2_td.csv" , sep=",", na = "NA", row.names=FALSE)

```

Representación de los resultados a partir de tablas y gráficas.

La representación de resultados a partir de tablas y gráficas se ha llevado a cabo a lo largo de toda la práctica y queda incluida en la misma.

Conclusiones :

Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

En el estudio abordado a lo largo de esta práctica utilizando el dataset house prices se han podido extraer varias conclusiones que van en línea al objetivo marcado cuando se decidió seleccionar el dataset indicado y que responden a las preguntas planteadas inicialmente. A continuación se exponen estas conclusiones de manera ordenada tal y como se han ido en los distintos puntos realizados en la práctica:

1. Estudio inicial para selección del dataset: en primer lugar se ha llevado a cabo una exploración del dataset seleccionado. Este consta de dos ficheros csv, train.csv y test.csv que para su tratamiento conjunto deben ser integrados. Durante el estudio del dataset se confirma que contiene 80 variables independientes y una variable dependiente, lo que cubre el objetivo de la práctica. Además entre estas variables existen de diferentes tipos, categóricas y numéricas.

2. Limpieza de datos: además en esta fase inicial del proceso, se lleva a cabo la búsqueda de valores vacíos para su limpieza. Entre estos se encuentran varias variables con valores vacíos, las cuales se estudian una a una para su gestión óptima. Si bien es cierto que inicialmente en primeras versiones se lleva a cabo la limpieza con ambos dataset combinados, finalmente se opta por utilizar el dataset train.csv para simplificar la práctica, aunque esta limpieza se deja completa. Para la gestión de los valores vacíos encontrados se tiene en cuenta el tipo de variable (categórica o numérica) y el dato que aporta, reemplazando de esta forma valores vacíos con una etiqueta de “Not apply” para las variables categóricas que pueden tomar este valor como (PoolQC, Fence, etc) y además para distinguir de la etiqueta None que se incluye en los datos originales. Por otro lado se han reemplazado con el valor 0 para las variables numéricas que pueden tomar este valor y además están asociadas a variables categóricas con valor None o “Not Apply”. A su vez los valores a 0 originales se han considerado valores legítimos por estar asociados a variables que no aplicaban (ej PoolArea). Por último también se ha incluido reemplazo utilizando estadísticos teniendo en cuenta agrupaciones por otras variables relacionadas, entre estos se han utilizado como la moda y la mediana.
3. Eliminación de outliers: tras la limpieza de valores nulos se han explorado los posibles outliers para su mejor gestión, en la cual se ha llevado a cabo la eliminación de estos sobre alguna de las variables como son:
 - MSSubClass - Esta variable son códigos que identifican distintos tipos de vivienda. Por tanto no se puede considerar que sean valores extremos o anómalos, ya que son códigos asignados por la codificación pactada. No requiere tratamiento adicional
 - LotFrontage - Esta variable indica el nº de pies (longitud) . También consideramos que puede haber propiedades en venta que tengan un mayor nº de pie. No requiere tratamiento.
 - LotArea - Esta variable indica el nº de metros cuadrados de la parcela. Aunque existen valores que están alejados de la media del resto de viviendas, vemos lógico que puedan ser un nº válido. No requiere tratamiento.
 - OverallQual : Los valores están dentro de los permitidos , de 0-10 , aunque la mayoría se sitúan en la calidad media , pero este se corresponde con alguna propiedad que no estará en buenas condiciones. No requiere tratamiento , ya que es nos da referencia de los precios con dicha calidad.
 - OverCond : Es similar , son todos valores permitidos , aunque la mayoría estén entre los valores intermedios.No requiere tratamiento.
 - YearBuilt : Año de construcción. En este caso , se opta por quitarlos porque distan mucho de la media situados entre los años 1950 - 2000 aproximadamente, por lo que puede distorsionar la muestra.
4. Análisis exploratorio: a continuación se ha estudiado la distribución y relación entre las diferentes variables independientes y entre estas y la variable dependiente. En este análisis se han podido extraer las primeras conclusiones
 - Se aprecia que el tipo de vivienda con mayor volumen es la del tipo 1Fam cuya distribución de precios se encuentra en un rango inferior a los 200.000USD.
 - El mayor volumen de propiedades se encuentra por debajo de los 200.000\$.con casos muy puntuales por encima de los 600.000USD.
 - Se puede ver que existe diferencias en el tipo de vivienda dependiendo de la zona, aunque se aprecia que el volumen mayor es la de tipo 1Fam. Así en zona residencial alta la vivienda fuera de 1Fam es de tipo duplex, sin embargo en zona residencial baja es de tipo Twnhse y 2fmCon.
 - Si añadimos el tipo de barrio a la relación anterior se observa que la distribución del tipo de vivienda también varía por barrios, si bien domina en casi todos los barrios el tipo 1Fam, hay barrios que solo tienen tipo Twnhse o Twnhs únicamente.
 - Se observa que la mayoría de las propiedades cuentan con aire acondicionado y que la distribución del precio , dependiendo de si tienen o no aire acondicionado es diferente. Ya se aprecia, que el precio de las que no tienen aire acondicionado están en un franja de precio menor. Más adelante , haremos un contraste de hipótesis para comprobar que la media del precio de las casas, es diferente si tiene aire o no .
 - En cuanto a la relación entre calidad de la propiedad y el precio, este aumenta de forma directa con la calidad.

- Igualmente para la relación del año de construcción con el precio de la propiedad según el tipo de esta, se ve un ligero incremento con los años del precio, es decir para propiedades construidas en años recientes el precio es mayor para el mismo tipo de vivienda. En cuanto al año de remodelación, también se observa que la variación de precio aumenta si se ha remodelado recientemente. Son variables explicativas del precio de la vivienda.
 - La relación del precio con el área del garaje igualmente es incremental, aumentando este a medida que el número de plazas de aparcamiento de la vivienda es mayor.
 - En cuanto al precio por el área habitable general de la vivienda, se observa claramente como según aumenta el nº de metros de área habitable, el precio se incrementa y que las casas de distintos tipos también se distribuyen en cluster diferenciados por lo que el precio medio variará entre ellas. También se observa que el precio de la vivienda en función del nº de metros habitables, y diferenciando por zona, se ve que el precio medio entre zonas es diferente.
 - En la relación del tipo de vivienda con el precio, se aprecia que los mayores precios se encuentran en el tipo 1Fam si bien este es el más frecuente. En cuanto al vecindario se ve claramente que hay vecindarios que destacan por ser donde se encuentran las propiedades más caras. Y en cuanto al precio por utilidades al tener la mayoría de propiedades el tipo AllPub, aquí se encuentra todo el rango de precios. Además se ha creado una variable nueva para el total baños con la suma de las 4 variables relacionadas con el nº de baños. Por otro lado se ha llevado una discretización con la variable año de construcción para ver su distribución.
5. Normalización y homogenización de la varianza: se han extraído en primer lugar las variables que no tienen distribución normal. Se ha revisado la normalización con gráficas de quantile-quantile plot e histograma. A continuación se ha procedido con el test de Shapiro para confirmar o refutar la hipótesis nula en algunas variables y se ha concluido que todas las variables testeadas están por debajo del coeficiente 0.05, es decir se rechaza la hipótesis nula pudiendo confirmar que no están normalizadas.
6. Reducción de la dimensionalidad y Matriz de correlación y estudio de PCA: tras el estudio de las relaciones entre variables se ha seleccionado un subconjunto de atributos para confirmar su correlación con la variable target, precio de la vivienda. Tras su visualización se ha confirmado que correlacionan en positivo con SalePrice las variables: X1stFlrSF, LotArea, MasVnrArea, LotFrontage, TotalBsmtSF, GarageCars, FullBath, YearRemodAdd, BsmtFinSF1, OverallQual, OpenPorchSF, GarageArea, Fireplaces, TotRmsAbvGrd, BedroomAbvGr, GrLivArea, CentralAir, YearBuilt.
Tras la correlación se ha llevado a cabo el estudio de PCA, con el objetivo de confirmar el porcentaje de contribución de los principales componentes y las variables que los constituyen.
7. Muestreo aleatorio simple y cluster: una vez realizado el estudio de PCA y a través de un muestreo aleatorio simple, se ha confirmado mediante clustering las variables que constituyen las agrupaciones, una vez realizada la limpieza y eliminación de outliers, obteniéndose 4 clusters cuyas variables confirman que las que más influyen son los metros cuadrados o área de la propiedad, el número de habitaciones, número de baños, plazas de aparcamiento, año de construcción y remodelación.
8. Contraste de hipótesis: a partir del contraste de medias realizado, se puede determinar que los precios medios difieren entre las propiedades que tienen aire acondicionado de aquellas que no lo incluyen. Asimismo se ha verificado que el precio puede diferir también por el tipo de vivienda, vecindario o zona donde se ubica la propiedad. Con el análisis ANOVA multifactorial se ha comprobado en qué medida influyen las variables explicativas del tipo de vivienda, vecindario, zona de ubicación, aire acondicionado y calefacción.
9. Algoritmo de regresión: se ha llevado a cabo un análisis de regresión para estudiar la proporción de varianza explicada, primero con algunas variables que correlacionan con la variable precio como regresión lineal simple y después con el conjunto de variables seleccionadas previamente, regresión lineal múltiple. De este estudio se ha extraído que el modelo con las principales variables tratadas puede explicar casi el 75%, un 74,57% de la varianza observada. El p-value del modelo es significativo < 0.00000000000000022 por lo que se puede concluir que existe una relación entre estas variables tratadas como predictoras y la variable target, precio de la vivienda, siendo estas variables predictoras las ya indicadas previamente.
10. Otro algoritmo: a su vez se ha realizado un modelo predictivo aplicando random forest, extrayéndose

que con ramdon forest se explica casi el 83% de la varianza, es decir algo más que con el modelo de regresión lineal, teniendo el modelo de random forest un menor error cuadrático medio, un mayor coeficiente R2 que explica la proporción de varianza de la variable SalePrice, de los precios de las viviendas, de acuerdo a las variables seleccionadas.

Con todo ello podemos dar respuesta a las preguntas plantadas al inicio de esta práctica, concluyendo que en el análisis queda demostrada las variables que son explicativas o influyen a la hora de determinar el precio, siendo las más relevantes: Entre las variables cualitativas: tipo de Zona, el tipo de vivienda , el vecindario, aire condicionado, calefacción. Entre las variables cuantitativas: calidad de la vivienda, área en metros cuadrados de la vivienda, nº de plazas de garaje, nº de habitaciones, nº de baños, año de construcción y remodelación. En este juego de datos no se ha podido confirmar la influencia de tener piscina en el precio de la vivienda, ya que la mayoría de las viviendas no han contado con esta característica.

Contribuciones :

A continuación se presentan los integrantes que han contribuido de forma conjunta en las distintas fases:

Contribuciones	Firmas
Investigación Previa	SMA/BMA
Redacción de las respuestas	SMA/BMA
Desarrollo Código	SMA/BMA

Entregable:

Los entregables asociados con esta PRA2 de la asignatura de Tipología de datos son: - Tipología de Datos_PRA2_Entrega.Rmd - Tipología de Datos_PRA2_Entrega.pdf

Ambos documentos se encuentran en: <https://github.com/smartialbar/PRA2/>

Además se ha completado la descripción de la práctica en el fichero README.

Referencias bibliográficas :

- Calvo M, Subirats L, Pérez D (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC.
- Squire, Megan (2015). Clean Data. Packt Publishing Ltd.
- Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.
- Transform en R, Carlos J. Gil Bellotta
- Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
- Tutorial de Github (<https://guides.github.com/activities/hello-world/>)

Table 1: Estimaciones media - Tendencia Central

variables	Media	Mediana	Media.recort.0.05	Media.winsor.0.05
MSSubClass	57.44	50.0	52.97	56.65
LotFrontage	69.34	70.0	68.66	68.54
LotArea	10150.14	9317.0	9422.15	9474.88
OverallQual	5.98	6.0	5.98	5.98
OverallCond	5.58	5.0	5.55	5.60
YearBuilt	1970.54	1972.0	1971.93	1970.95
YearRemodAdd	1984.04	1992.0	1984.59	1983.99
MasVnrArea	90.18	0.0	67.75	81.30
BsmtFinSF1	419.09	372.5	387.56	408.47
BsmtFinSF2	47.72	0.0	15.78	34.28
BsmtUnfSF	557.78	473.5	530.28	548.29
TotalBsmtSF	1024.59	973.5	1021.53	1027.53
X1stFlrSF	1132.27	1068.5	1115.71	1123.78
X2ndFlrSF	333.06	0.0	301.46	326.04
LowQualFinSF	5.57	0.0	0.00	0.00
GrLivArea	1470.90	1435.0	1445.15	1458.22
BsmtFullBath	0.41	0.0	0.39	0.40
BsmtHalfBath	0.06	0.0	0.01	0.06
FullBath	1.54	2.0	1.54	1.53
HalfBath	0.37	0.0	0.35	0.36
BedroomAbvGr	2.86	3.0	2.86	2.87
KitchenAbvGr	1.05	1.0	1.00	1.00
TotRmsAbvGrd	6.41	6.0	6.35	6.36
Fireplaces	0.58	1.0	0.54	0.58
GarageYrBlt	1865.34	1977.0	1960.14	1865.30
GarageCars	1.72	2.0	1.74	1.72
GarageArea	459.25	471.0	459.47	454.84
WoodDeckSF	89.72	0.0	75.85	84.48
OpenPorchSF	44.92	22.0	36.09	41.04
EnclosedPorch	21.86	0.0	11.48	19.18
X3SsnPorch	3.25	0.0	0.00	0.00
ScreenPorch	14.71	0.0	3.80	11.13
PoolArea	2.49	0.0	0.00	0.00
MiscVal	45.61	0.0	0.00	0.00
MoSold	6.31	6.0	6.28	6.31
YrSold	2007.82	2008.0	2007.80	2007.82
SalePrice	170240.85	159500.0	167700.08	169528.12

Table 2: Estimaciones de Dispersion

variables	Desv.Standard	IQR	MAD
MSSubClass	42.90	50.00	44.48
LotFrontage	21.93	20.00	14.83
LotArea	8338.57	3867.50	2865.87
OverallQual	1.28	2.00	1.48
OverallCond	1.11	1.00	0.00
YearBuilt	29.37	46.00	35.58
YearRemodAdd	20.68	37.00	20.76
MasVnrArea	159.49	144.00	0.00
BsmtFinSF1	428.69	686.00	552.27
BsmtFinSF2	163.02	0.00	0.00
BsmtUnfSF	429.13	589.25	426.25
TotalBsmtSF	407.54	464.00	326.17
X1stFlrSF	357.67	480.25	326.91
X2ndFlrSF	418.41	720.00	0.00
LowQualFinSF	47.11	0.00	0.00
GrLivArea	476.41	607.25	451.45
BsmtFullBath	0.52	1.00	0.00
BsmtHalfBath	0.24	0.00	0.00
FullBath	0.54	1.00	0.00
HalfBath	0.50	1.00	0.00
BedroomAbvGr	0.80	1.00	0.00
KitchenAbvGr	0.23	0.00	0.00
TotRmsAbvGrd	1.54	2.00	1.48
Fireplaces	0.63	1.00	1.48
GarageYrBlt	458.33	43.00	32.62
GarageCars	0.72	1.00	0.00
GarageArea	203.26	258.75	164.57
WoodDeckSF	121.83	165.00	0.00
OpenPorchSF	65.76	64.00	32.62
EnclosedPorch	60.25	0.00	0.00
X3SsnPorch	28.63	0.00	0.00
ScreenPorch	54.88	0.00	0.00
PoolArea	38.39	0.00	0.00
MiscVal	508.01	0.00	0.00
MoSold	2.70	3.00	2.97
YrSold	1.33	2.00	1.48
SalePrice	59149.74	74250.00	51223.83