TFM - Kaggle House Prices: Advanced Regression Techniques with caret

01 PREPROCESAMIENTO - Análisis y Limpieza inicial del conjunto de datos

Juan Carlos Santiago Culebras

2019-09-21

El objetivo de esta etapa es realizar un estudio de los datos y realizar una primera limpieza, ha sido una fase muy costosa ya que implica entender el conjunto de datos tanto en su estructura como en su contenido.

Primeros pasos

Librerías

Realizamos la carga de las librerías necesarias

```
if(!is.element("dplyr", installed.packages()[, 1]))
      install.packages("dplyr", repos = 'http://cran.us.r-project.org')
library(dplyr)

if(!is.element("tidyr", installed.packages()[, 1]))
      install.packages("tidyr", repos = 'http://cran.us.r-project.org')
library(tidyr)

if(!is.element("ggplot2", installed.packages()[, 1]))
      install.packages("ggplot2", repos = 'http://cran.us.r-project.org')
library(ggplot2)

if(!is.element("tibble", installed.packages()[, 1]))
      install.packages("tibble", repos = 'http://cran.us.r-project.org')
library(tibble)

# grid.arrange / marrangeGrob
if(!is.element("gridExtra", installed.packages()[, 1]))
      install.packages("gridExtra", repos = 'http://cran.us.r-project.org')
library(gridExtra)


if(!is.element("cowplot", installed.packages()[, 1]))
      install.packages("cowplot", repos = 'http://cran.us.r-project.org')
library(cowplot)

if(!is.element("psych", installed.packages()[, 1]))
      install.packages("psych", repos = 'http://cran.us.r-project.org')
library(psych)

# correlation matrixes - rcorr (niveles de significación)
```

```r
if(!is.element("Hmisc", installed.packages()[, 1]))
      install.packages("Hmisc", repos = 'http://cran.us.r-project.org')
library(Hmisc)

# correlation matrixes - ggcorr
if(!is.element("GGally", installed.packages()[, 1]))
      install.packages("GGally", repos = 'http://cran.us.r-project.org')
library(GGally)

# correlation matrixes - corrplot
if(!is.element("corrplot", installed.packages()[, 1]))
      install.packages("corrplot", repos = 'http://cran.us.r-project.org')
library(corrplot)

if(!is.element("ggpubr", installed.packages()[, 1]))
      install.packages("ggpubr", repos = 'http://cran.us.r-project.org')
library(ggpubr)
```

Funciones

Función que permiten presentar gráficamente las variables y sus estadísticas de resumen, tendencia central, dispersión y forma.

```r
ggplotHistogramaDensidad <- function (strCampo,ds) {
  require(psych)
  require(ggplot2)

  # Estudio mas detallado medidas de dispersión con curtosis y sesgo
  strDescribe = paste("d <- psych::describe(ds$",strCampo,")",sep = "")
  eval(parse(text = paste(strDescribe)))

  title <- strCampo

  t1 <- capture.output(summary(ds[,strCampo]))
  t1 <- paste("Summary:", paste(t1, collapse="\n"), " ", sep = "\n")
  t2 <- paste("Kurtosis:", signif(d$kurtosis,3)," / Skew:", signif(d$skew,3))

  subtitle <- paste(t1, t2, sep = "\n")

  p1 <- ggplot(ds, aes(x=get(strCampo))) +
    geom_histogram(aes(y=..density..), colour="black", fill = "white") +
    geom_vline(aes(xintercept=mean(get(strCampo))),
            color="blue", linetype="dashed", size=1) +
    geom_density(alpha=.2, fill="#FF6666") +
    labs(title=title, subtitle=subtitle, x = strCampo) +
    theme(plot.subtitle = element_text(size=10)) +
    scale_x_continuous(labels = scales::comma)

  return(p1)
}
```

Cargamos datos

Datos originales de la competición

```
# Leer datos
dsTrain <- read.csv("./input/train.csv")
dsTest <- read.csv("./input/test.csv")
```

Conjunto unificado

Juntamos los datos de entrenamiento con los de test para realizar el estudio y las transformaciones pertinentes sobre todos los datos.

- Añadimos SalePrice al conjunto de Test con valor NA
- Marcamos datos de entrenamiento y test

```
dsTest <- dsTest %>%
  mutate(SalePrice = as.integer(NA), indTrain = 0)

dsDataAll <- dsTrain %>%
  mutate(indTrain = 1) %>%
  union(dsTest) %>%
  select(SalePrice, indTrain, everything())

dsDataAll$indTrain <- as.factor(dsDataAll$indTrain)

# Elimino los conjuntos originales
rm(dsTrain)
rm(dsTest)
```

He generado un fichero de documentación de los campos, con su descripción, tipo, posibles valores y una clasificación según la información que suministra para el problema en cuestión. Ver fichero "campos.csv".

Para los campos Ordinales / Nominales he creado un fichero con los valores posibles, de tal forma que se pueda verificar que el contenido del fichero es correcto. Además permitira crear correctamente los factores, en caso de falta de información.

Cargo los datos de los csv

```
# Definición de campos
dsCampos <- read.csv("./input/campos.csv",sep=";",stringsAsFactors = FALSE)

dsCampos <- dsCampos %>%
  mutate_if(is.factor, as.character)

# Dejamos el Tipo y el segmento como factor
dsCampos$Tipo <- as.factor(dsCampos$Tipo)
dsCampos$Segmento <- as.factor(dsCampos$Segmento)


# Valores de campos
dsCamposValor <- read.csv("./input/Campos_Valor.csv",sep=";",stringsAsFactors = FALSE)
dsCamposValor$Valor <- stringr::str_trim(dsCamposValor$Valor)
```

Análisis descriptivo inicial del conjunto de datos

Estudio preliminar, vemos tipos de datos y una primera aproximación al contenido.

```
dim(dsDataAll)
```

```
## [1] 2919   82
```

```
names(dsDataAll)
```

```
##  [1] "SalePrice"     "indTrain"      "Id"            "MSSubClass"
##  [5] "MSZoning"      "LotFrontage"   "LotArea"       "Street"
##  [9] "Alley"         "LotShape"      "LandContour"   "Utilities"
## [13] "LotConfig"     "LandSlope"     "Neighborhood"  "Condition1"
## [17] "Condition2"    "BldgType"      "HouseStyle"    "OverallQual"
## [21] "OverallCond"   "YearBuilt"     "YearRemodAdd"  "RoofStyle"
## [25] "RoofMatl"      "Exterior1st"   "Exterior2nd"   "MasVnrType"
## [29] "MasVnrArea"    "ExterQual"     "ExterCond"     "Foundation"
## [33] "BsmtQual"      "BsmtCond"      "BsmtExposure"  "BsmtFinType1"
## [37] "BsmtFinSF1"    "BsmtFinType2"  "BsmtFinSF2"    "BsmtUnfSF"
## [41] "TotalBsmtSF"   "Heating"       "HeatingQC"     "CentralAir"
## [45] "Electrical"    "X1stFlrSF"     "X2ndFlrSF"     "LowQualFinSF"
## [49] "GrLivArea"     "BsmtFullBath"  "BsmtHalfBath"  "FullBath"
## [53] "HalfBath"      "BedroomAbvGr"  "KitchenAbvGr"  "KitchenQual"
## [57] "TotRmsAbvGrd"  "Functional"    "Fireplaces"    "FireplaceQu"
## [61] "GarageType"    "GarageYrBlt"   "GarageFinish"  "GarageCars"
## [65] "GarageArea"    "GarageQual"    "GarageCond"    "PavedDrive"
## [69] "WoodDeckSF"    "OpenPorchSF"   "EnclosedPorch" "X3SsnPorch"
## [73] "ScreenPorch"   "PoolArea"      "PoolQC"        "Fence"
## [77] "MiscFeature"   "MiscVal"       "MoSold"        "YrSold"
## [81] "SaleType"      "SaleCondition"
```

```
str(dsDataAll) #glimpse(dsDataAll)
```

```
## 'data.frame':    2919 obs. of  82 variables:
##  $ SalePrice    : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
##  $ indTrain     : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning     : Factor w/ 5 levels "C (all)","FV",..: 4 4 4 4 4 4 4 4 5 4 ...
##  $ LotFrontage  : int  65 80 68 60 84 85 75 NA 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
##  $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Alley        : Factor w/ 2 levels "Grvl","Pave": NA NA NA NA NA NA NA NA NA NA ...
##  $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",..: 4 4 1 1 1 1 4 1 4 4 ...
##  $ LandContour  : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Utilities    : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
##  $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5 1 5 1 ...
##  $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",..: 6 25 6 7 14 12 21 17 18 4 ...
##  $ Condition1   : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 5 1 1 ...
##  $ Condition2   : chr  "Norm" "Norm" "Norm" "Norm" ...
```

```
##  $ BldgType     : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 1 1 2 ...
##  $ 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    : Factor w/ 6 levels "Flat","Gable",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ RoofMatl     : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
##  $ Exterior1st  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...
##  $ Exterior2nd  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Shng" ...
##  $ MasVnrType   : Factor w/ 4 levels "BrkCmn","BrkFace",..: 2 3 2 3 2 3 4 4 3 3 ...
##  $ MasVnrArea   : int  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 4 3 4 3 4 4 4 ...
##  $ ExterCond    : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ Foundation   : Factor w/ 6 levels "BrkTil","CBlock",..: 3 2 3 1 3 6 3 2 1 1 ...
##  $ BsmtQual     : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 3 3 4 3 3 1 3 4 4 ...
##  $ BsmtCond     : Factor w/ 4 levels "Fa","Gd","Po",..: 4 4 4 2 4 4 4 4 4 4 ...
##  $ BsmtExposure : Factor w/ 4 levels "Av","Gd","Mn",..: 4 2 3 4 1 4 1 3 4 4 ...
##  $ BsmtFinType1 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 3 1 3 1 3 3 3 1 6 3 ...
##  $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
##  $ BsmtFinType2 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 6 6 6 6 6 6 6 2 6 6 ...
##  $ BsmtFinSF2   : int  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : int  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
##  $ Heating      : chr  "GasA" "GasA" "GasA" "GasA" ...
##  $ HeatingQC    : Factor w/ 5 levels "Ex","Fa","Gd",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ CentralAir   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical   : 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  : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 3 3 4 3 4 4 4 ...
##  $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : Factor w/ 7 levels "Maj1","Maj2",..: 7 7 7 7 7 7 7 7 3 7 ...
##  $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu  : Factor w/ 5 levels "Ex","Fa","Gd",..: NA 5 5 3 5 NA 3 5 5 5 ...
##  $ GarageType   : Factor w/ 6 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2 6 2 ...
##  $ GarageYrBlt  : int  2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...
##  $ GarageFinish : Factor w/ 3 levels "Fin","RFn","Unf": 2 2 2 3 2 3 2 2 3 2 ...
##  $ GarageCars   : int  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : int  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : chr  "TA" "TA" "TA" "TA" ...
##  $ GarageCond   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ PavedDrive   : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
##  $ WoodDeckSF   : int  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF  : int  61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch: int  0 0 0 272 0 0 0 228 205 0 ...
```

```
##  $ X3SsnPorch   : int  0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : chr  NA NA NA NA ...
##  $ Fence        : Factor w/ 4 levels "GdPrv","GdWo",..: NA NA NA NA NA 3 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     : Factor w/ 9 levels "COD","Con","ConLD",..: 9 9 9 9 9 9 9 9 9 9 ...
##  $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 1 5 5 5 5 1 5 ...
```

```
summary(dsDataAll)
```

```
##    SalePrice      indTrain       Id           MSSubClass
##  Min.   : 34900   0:1459   Min.   :   1.0   Min.   : 20.00
##  1st Qu.:129975   1:1460   1st Qu.: 730.5   1st Qu.: 20.00
##  Median :163000            Median :1460.0   Median : 50.00
##  Mean   :180921            Mean   :1460.0   Mean   : 57.14
##  3rd Qu.:214000            3rd Qu.:2189.5   3rd Qu.: 70.00
##  Max.   :755000            Max.   :2919.0   Max.   :190.00
##  NA's   :1459
##     MSZoning     LotFrontage        LotArea        Street      Alley
##  C (all):  25   Min.   : 21.00   Min.   :  1300   Grvl:  12   Grvl: 120
##  FV     : 139   1st Qu.: 59.00   1st Qu.:  7478   Pave:2907   Pave:  78
##  RH     :  26   Median : 68.00   Median :  9453               NA's:2721
##  RL     :2265   Mean   : 69.31   Mean   : 10168
##  RM     : 460   3rd Qu.: 80.00   3rd Qu.: 11570
##  NA's   :   4   Max.   :313.00   Max.   :215245
##                 NA's   :486
##  LotShape   LandContour  Utilities           LotConfig    LandSlope
##  IR1: 968   Bnk: 117    Length:2919        Corner : 511   Gtl:2778
##  IR2:  76   HLS: 120    Class :character   CulDSac: 176   Mod: 125
##  IR3:  16   Low:  60    Mode  :character   FR2    :  85   Sev:  16
##  Reg:1859   Lvl:2622                       FR3    :  14
##                                            Inside :2133
##
##
##   Neighborhood    Condition1    Condition2          BldgType
##  NAmes  : 443   Norm   :2511   Length:2919        1Fam  :2425
##  CollgCr: 267   Feedr  : 164   Class :character   2fmCon:  62
##  OldTown: 239   Artery :  92   Mode  :character   Duplex: 109
##  Edwards: 194   RRAn   :  50                      Twnhs :  96
##  Somerst: 182   PosN   :  39                      TwnhsE: 227
##  NridgHt: 166   RRAe   :  28
##  (Other):1428   (Other):  35
##   HouseStyle         OverallQual      OverallCond      YearBuilt
##  Length:2919        Min.   : 1.000   Min.   :1.000   Min.   :1872
##  Class :character   1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954
##  Mode  :character   Median : 6.000   Median :5.000   Median :1973
##                     Mean   : 6.089   Mean   :5.565   Mean   :1971
##                     3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2001
##                     Max.   :10.000   Max.   :9.000   Max.   :2010
##
```

```
##   YearRemodAdd    RoofStyle      RoofMatl         Exterior1st
##  Min.   :1950   Flat   :  20   Length:2919        Length:2919
##  1st Qu.:1965   Gable  :2310   Class :character   Class :character
##  Median :1993   Gambrel:  22   Mode  :character   Mode  :character
##  Mean   :1984   Hip    : 551
##  3rd Qu.:2004   Mansard:  11
##  Max.   :2010   Shed   :   5
##
##  Exterior2nd          MasVnrType     MasVnrArea     ExterQual ExterCond
##  Length:2919        BrkCmn :  25   Min.   :   0.0   Ex: 107   Ex:  12
##  Class :character   BrkFace: 879   1st Qu.:   0.0   Fa:  35   Fa:  67
##  Mode  :character   None   :1742   Median :   0.0   Gd: 979   Gd: 299
##                     Stone  : 249   Mean   : 102.2   TA:1798   Po:   3
##                     NA's   :  24   3rd Qu.: 164.0             TA:2538
##                                    Max.   :1600.0
##                                    NA's   :23
##   Foundation   BsmtQual    BsmtCond    BsmtExposure BsmtFinType1
##  BrkTil: 311   Ex  : 258   Fa  : 104   Av  : 418    ALQ :429
##  CBlock:1235   Fa  :  88   Gd  : 122   Gd  : 276    BLQ :269
##  PConc :1308   Gd  :1209   Po  :   5   Mn  : 239    GLQ :849
##  Slab  :  49   TA  :1283   TA  :2606   No  :1904    LwQ :154
##  Stone :  11   NA's:  81   NA's:  82   NA's:  82    Rec :288
##  Wood  :   5                                       Unf :851
##                                                    NA's: 79
##    BsmtFinSF1     BsmtFinType2   BsmtFinSF2        BsmtUnfSF
##  Min.   :   0.0   ALQ :  52    Min.   :   0.00   Min.   :   0.0
##  1st Qu.:   0.0   BLQ :  68    1st Qu.:   0.00   1st Qu.: 220.0
##  Median : 368.5   GLQ :  34    Median :   0.00   Median : 467.0
##  Mean   : 441.4   LwQ :  87    Mean   :  49.58   Mean   : 560.8
##  3rd Qu.: 733.0   Rec : 105    3rd Qu.:   0.00   3rd Qu.: 805.5
##  Max.   :5644.0   Unf :2493    Max.   :1526.00   Max.   :2336.0
##  NA's   :1        NA's:  80    NA's   :1         NA's   :1
##   TotalBsmtSF       Heating          HeatingQC CentralAir
##  Min.   :   0.0   Length:2919        Ex:1493   N: 196
##  1st Qu.: 793.0   Class :character   Fa:  92   Y:2723
##  Median : 989.5   Mode  :character   Gd: 474
##  Mean   :1051.8                      Po:   3
##  3rd Qu.:1302.0                      TA: 857
##  Max.   :6110.0
##  NA's   :1
##   Electrical          X1stFlrSF      X2ndFlrSF       LowQualFinSF
##  Length:2919        Min.   : 334   Min.   :   0.0   Min.   :   0.000
##  Class :character   1st Qu.: 876   1st Qu.:   0.0   1st Qu.:   0.000
##  Mode  :character   Median :1082   Median :   0.0   Median :   0.000
##                     Mean   :1160   Mean   : 336.5   Mean   :   4.694
##                     3rd Qu.:1388   3rd Qu.: 704.0   3rd Qu.:   0.000
##                     Max.   :5095   Max.   :2065.0   Max.   :1064.000
##
##    GrLivArea     BsmtFullBath     BsmtHalfBath        FullBath
##  Min.   : 334   Min.   :0.0000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:1126   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:1.000
##  Median :1444   Median :0.0000   Median :0.00000   Median :2.000
##  Mean   :1501   Mean   :0.4299   Mean   :0.06136   Mean   :1.568
##  3rd Qu.:1744   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:2.000
```

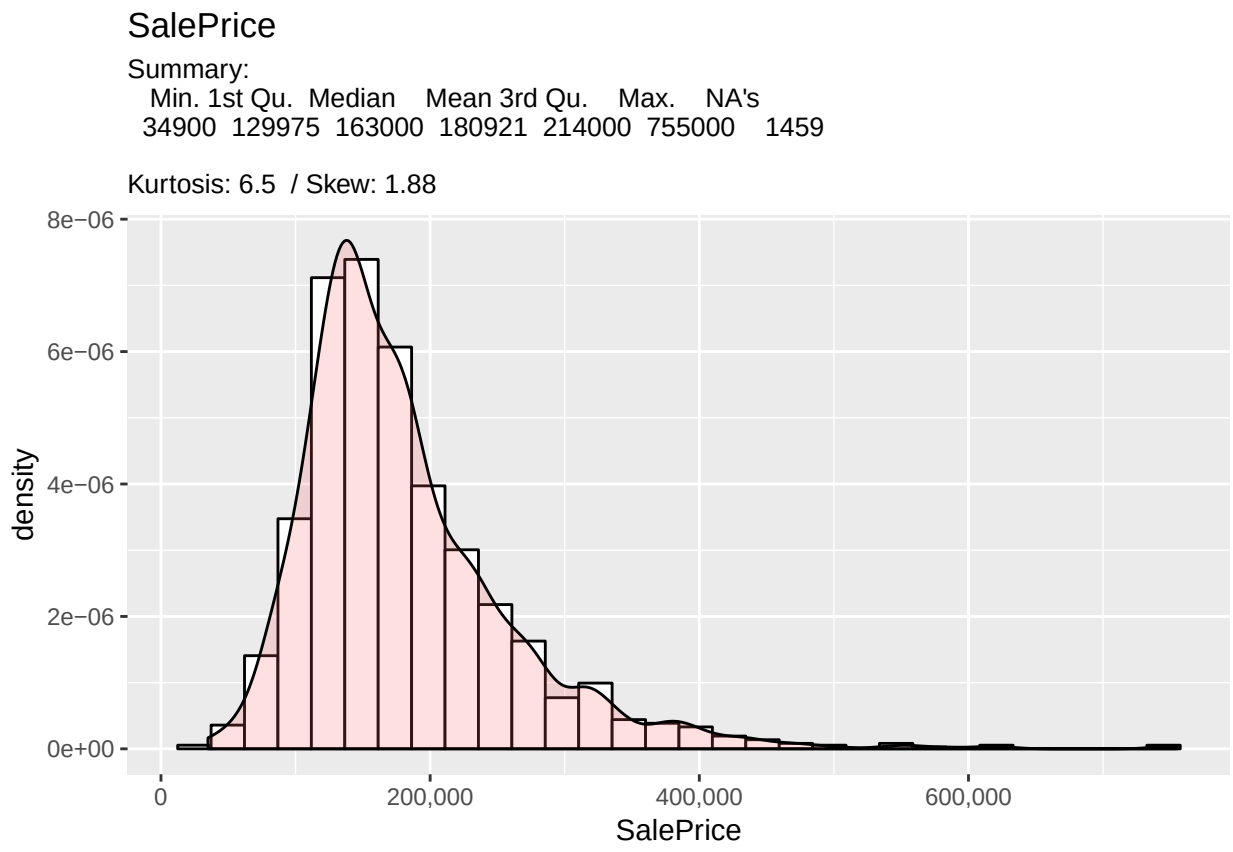
```
##  Max.   :5642   Max.   :3.0000   Max.   :2.00000   Max.   :4.000
##                 NA's   :2        NA's   :2
##     HalfBath       BedroomAbvGr   KitchenAbvGr   KitchenQual
##  Min.   :0.0000   Min.   :0.00   Min.   :0.000   Ex  : 205
##  1st Qu.:0.0000   1st Qu.:2.00   1st Qu.:1.000   Fa  :  70
##  Median :0.0000   Median :3.00   Median :1.000   Gd  :1151
##  Mean   :0.3803   Mean   :2.86   Mean   :1.045   TA  :1492
##  3rd Qu.:1.0000   3rd Qu.:3.00   3rd Qu.:1.000   NA's:   1
##  Max.   :2.0000   Max.   :8.00   Max.   :3.000
##
##   TotRmsAbvGrd      Functional     Fireplaces     FireplaceQu
##  Min.   : 2.000   Typ    :2717   Min.   :0.0000   Ex  :  43
##  1st Qu.: 5.000   Min2   :  70   1st Qu.:0.0000   Fa  :  74
##  Median : 6.000   Min1   :  65   Median :1.0000   Gd  : 744
##  Mean   : 6.452   Mod    :  35   Mean   :0.5971   Po  :  46
##  3rd Qu.: 7.000   Maj1   :  19   3rd Qu.:1.0000   TA  : 592
##  Max.   :15.000   (Other):  11   Max.   :4.0000   NA's:1420
##                   NA's   :   2
##    GarageType    GarageYrBlt   GarageFinish   GarageCars
##  2Types :  23   Min.   :1895   Fin : 719    Min.   :0.000
##  Attchd :1723   1st Qu.:1960   RFn : 811    1st Qu.:1.000
##  Basment:  36   Median :1979   Unf :1230    Median :2.000
##  BuiltIn: 186   Mean   :1978   NA's: 159    Mean   :1.767
##  CarPort:  15   3rd Qu.:2002                3rd Qu.:2.000
##  Detchd : 779   Max.   :2207                Max.   :5.000
##  NA's   : 157   NA's   :159                 NA's   :1
##    GarageArea      GarageQual        GarageCond  PavedDrive
##  Min.   :   0.0   Length:2919        Ex  :   3   N: 216
##  1st Qu.: 320.0   Class :character   Fa  :  74   P:  62
##  Median : 480.0   Mode  :character   Gd  :  15   Y:2641
##  Mean   : 472.9                      Po  :  14
##  3rd Qu.: 576.0                      TA  :2654
##  Max.   :1488.0                      NA's: 159
##  NA's   :1
##    WoodDeckSF       OpenPorchSF     EnclosedPorch      X3SsnPorch
##  Min.   :   0.00   Min.   :  0.00   Min.   :   0.0   Min.   :  0.000
##  1st Qu.:   0.00   1st Qu.:  0.00   1st Qu.:   0.0   1st Qu.:  0.000
##  Median :   0.00   Median : 26.00   Median :   0.0   Median :  0.000
##  Mean   :  93.71   Mean   : 47.49   Mean   :  23.1   Mean   :  2.602
##  3rd Qu.: 168.00   3rd Qu.: 70.00   3rd Qu.:   0.0   3rd Qu.:  0.000
##  Max.   :1424.00   Max.   :742.00   Max.   :1012.0   Max.   :508.000
##
##   ScreenPorch        PoolArea          PoolQC            Fence
##  Min.   :  0.00   Min.   :  0.000   Length:2919        GdPrv: 118
##  1st Qu.:  0.00   1st Qu.:  0.000   Class :character   GdWo : 112
##  Median :  0.00   Median :  0.000   Mode  :character   MnPrv: 329
##  Mean   : 16.06   Mean   :  2.252                      MnWw :  12
##  3rd Qu.:  0.00   3rd Qu.:  0.000                      NA's :2348
##  Max.   :576.00   Max.   :800.000
##
##  MiscFeature          MiscVal             MoSold           YrSold
##  Length:2919        Min.   :    0.00   Min.   : 1.000   Min.   :2006
##  Class :character   1st Qu.:    0.00   1st Qu.: 4.000   1st Qu.:2007
##  Mode  :character   Median :    0.00   Median : 6.000   Median :2008
```

```
##                     Mean   :   50.83   Mean   : 6.213   Mean   :2008
##                     3rd Qu.:    0.00   3rd Qu.: 8.000   3rd Qu.:2009
##                     Max.   :17000.00   Max.   :12.000   Max.   :2010
##
##     SaleType    SaleCondition
##  WD     :2525   Abnorml: 190
##  New    : 239   AdjLand:  12
##  COD    :  87   Alloca :  24
##  ConLD  :  26   Family :  46
##  CWD    :  12   Normal :2402
##  (Other):  29   Partial: 245
##  NA's   :   1
```

Variable objetivo

Estudiamos la variable objetivo mediante la función creada.

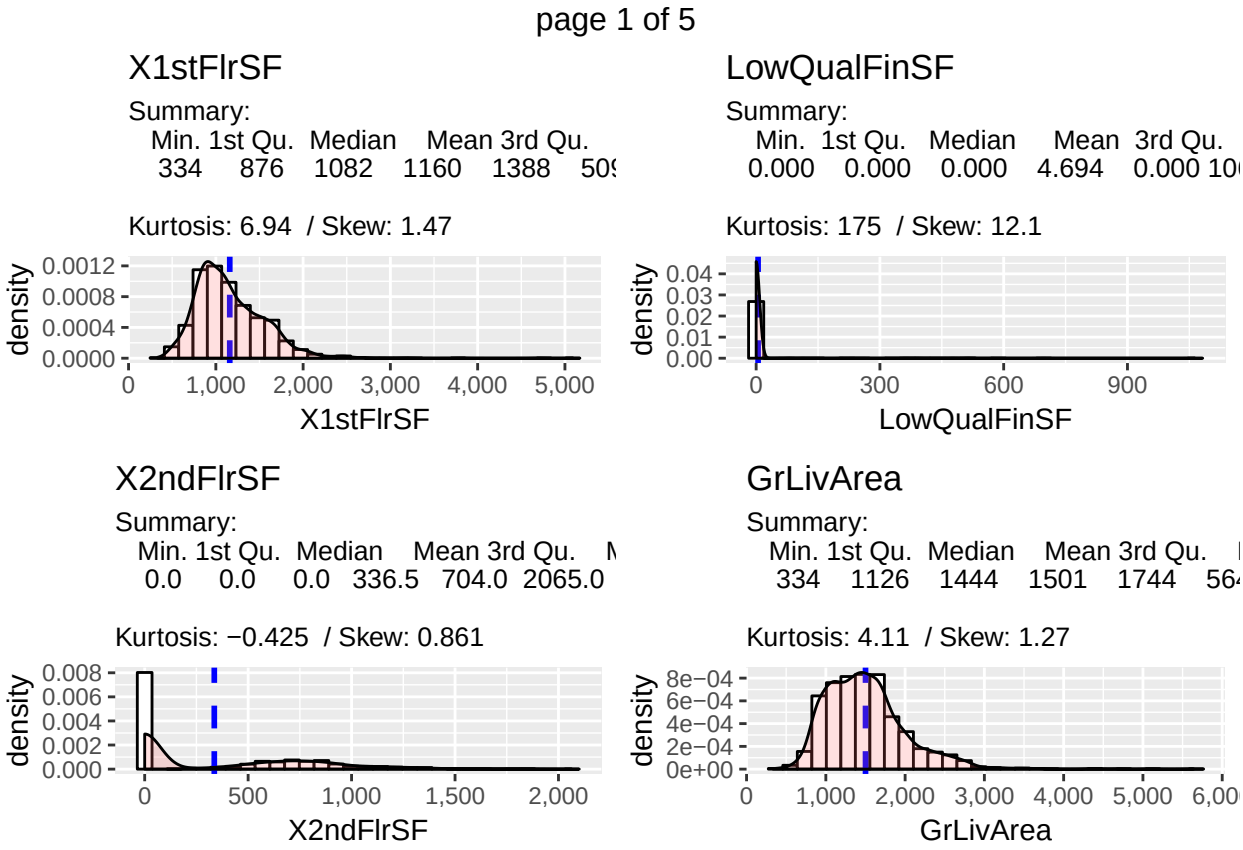
```
ggplotHistogramaDensidad("SalePrice",dsDataAll)
```



Resto de variables continuas

Realizamos la misma operación con el resto de variables continuas.

```r
# Selecciono variables continuas
dsCamposContinua <- dsCampos %>%
  filter(Tipo=="Continua" & Campo!="SalePrice")  %>%
  select(Campo)

#Genero histograma con densidad para cada variable
gs <- apply(dsCamposContinua, MARGIN=1, ggplotHistogramaDensidad, ds=dsDataAll)
marrangeGrob(grobs=gs, nrow=2, ncol=2)
```

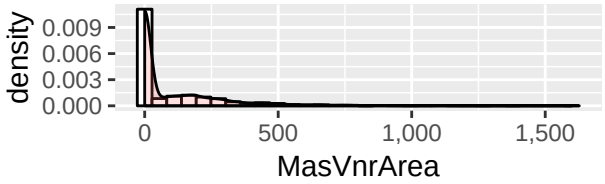


MasVnrArea

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
0.0	0.0	0.0	102.2	164.0	1600.0

Kurtosis: 9.23 / Skew: 2.6

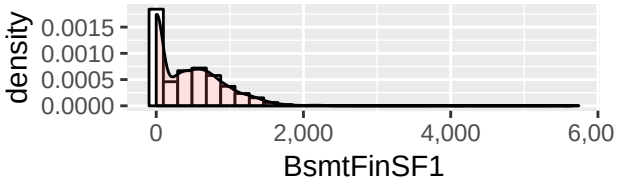


BsmtFinSF1

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	
0.0	0.0	368.5	441.4	733.0	5644.

Kurtosis: 6.88 / Skew: 1.42

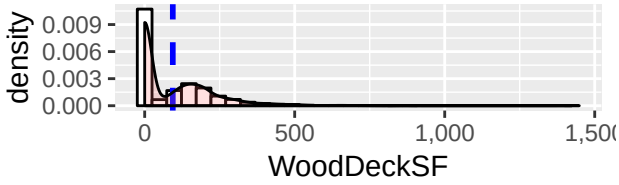


WoodDeckSF

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
0.00	0.00	0.00	93.71	168.00	1424.(

Kurtosis: 6.72 / Skew: 1.84

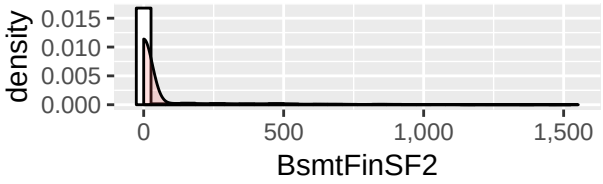


BsmtFinSF2

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
0.00	0.00	0.00	49.58	0.00	1526.0(

Kurtosis: 18.8 / Skew: 4.14

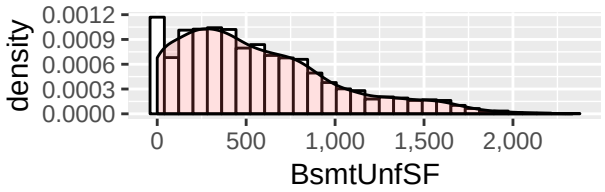


BsmtUnfSF

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	
0.0	220.0	467.0	560.8	805.5	233(

Kurtosis: 0.399 / Skew: 0.919

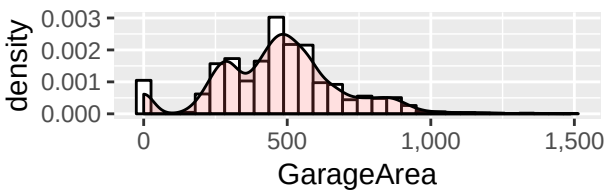


GarageArea

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
0.0	320.0	480.0	472.9	576.0	1488.

Kurtosis: 0.933 / Skew: 0.241

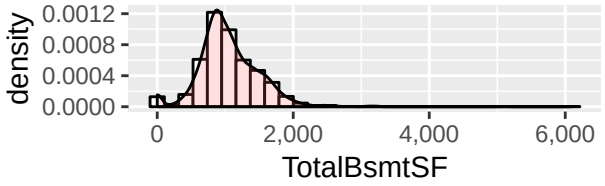


TotalBsmtSF

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	
0.0	793.0	989.5	1051.8	1302.0	61:

Kurtosis: 9.13 / Skew: 1.16

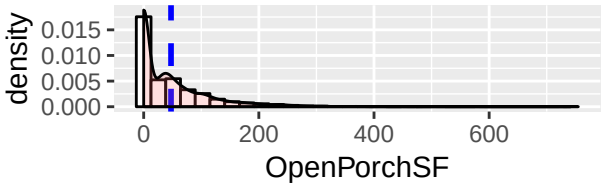


OpenPorchSF

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
0.00	0.00	26.00	47.49	70.00	742.0

Kurtosis: 10.9 / Skew: 2.53

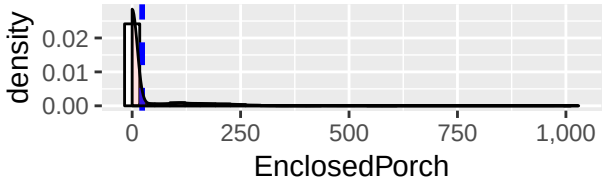


EnclosedPorch

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
0.0	0.0	0.0	23.1	0.0	1012.0

Kurtosis: 28.3 / Skew: 4

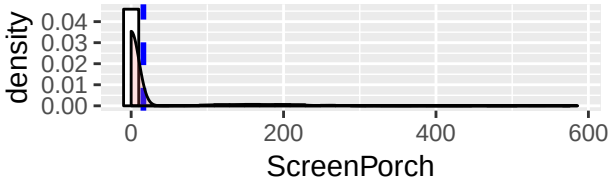


ScreenPorch

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
0.00	0.00	0.00	16.06	0.00	576.00

Kurtosis: 17.7 / Skew: 3.94

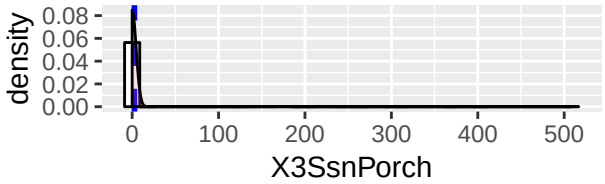


X3SsnPorch

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
0.000	0.000	0.000	2.602	0.000	508.0

Kurtosis: 149 / Skew: 11.4

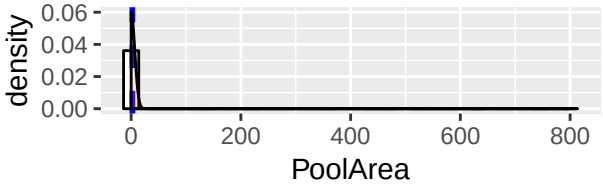


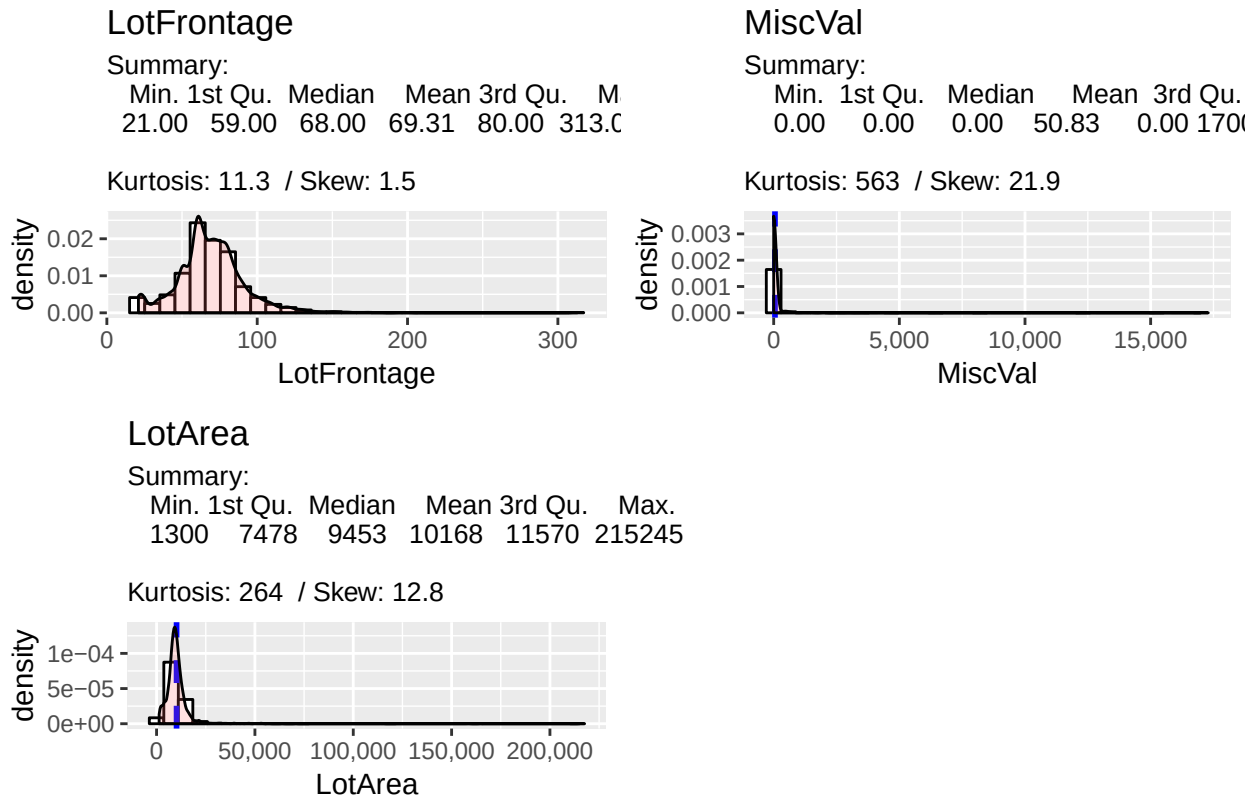
PoolArea

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
0.000	0.000	0.000	2.252	0.000	800.0

Kurtosis: 298 / Skew: 16.9






```
rm(gs)
```

Resto de variables discretas

Realizamos la misma operación con el resto de variables discretas.

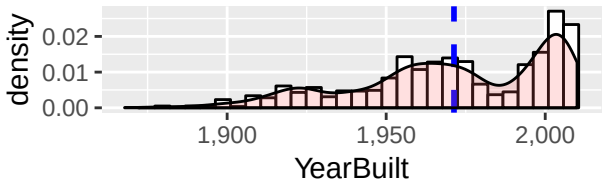
```
# Selecciono variables continuas
dsCamposDiscreta <- dsCampos %>%
  filter(Tipo=="Discreta")  %>%
  select(Campo)

#Genero histograma con densidad para cada variable
gs <- apply(dsCamposDiscreta, MARGIN=1, ggplotHistogramaDensidad, ds=dsDataAll)
marrangeGrob(grobs=gs, nrow=2, ncol=2)
```

YearBuilt

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
1872	1954	1973	1971	2001	201(

Kurtosis: −0.514 / Skew: −0.599

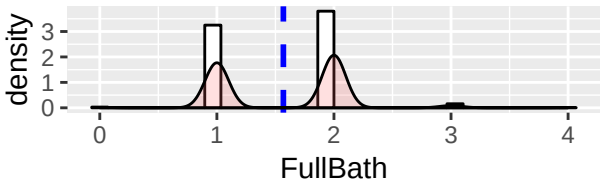


FullBath

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.000	1.000	2.000	1.568	2.000	4.000

Kurtosis: −0.541 / Skew: 0.168

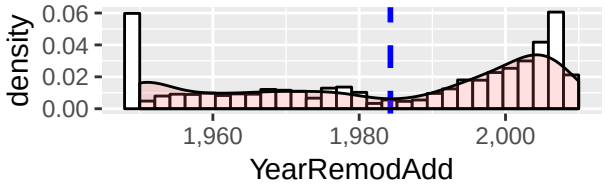


YearRemodAdd

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
1950	1965	1993	1984	2004	201(

Kurtosis: −1.35 / Skew: −0.451

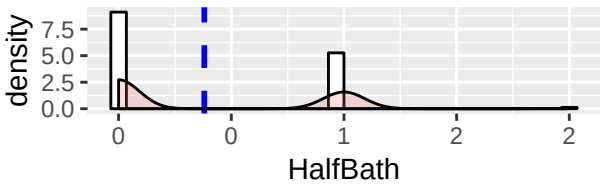


HalfBath

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	Ma
0.0000	0.0000	0.0000	0.3803	1.0000	2.0

Kurtosis: −1.04 / Skew: 0.694

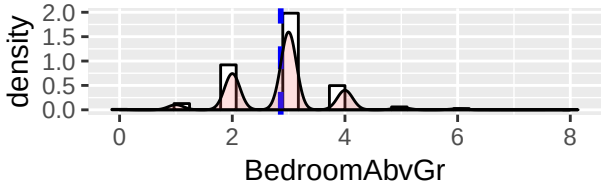


BedroomAbvGr

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	Ma
0.00	2.00	3.00	2.86	3.00	8.00

Kurtosis: 1.93 / Skew: 0.326

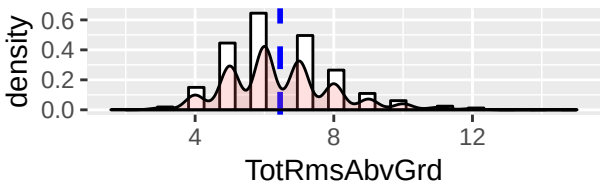


TotRmsAbvGrd

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	Ma
2.000	5.000	6.000	6.452	7.000	15.000

Kurtosis: 1.16 / Skew: 0.758

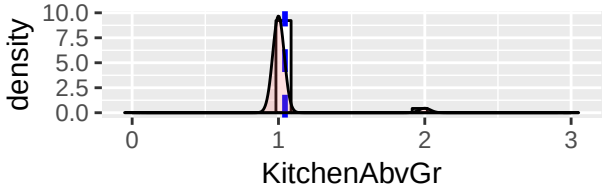


KitchenAbvGr

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
0.000	1.000	1.000	1.045	1.000	3.000

Kurtosis: 19.7 / Skew: 4.3

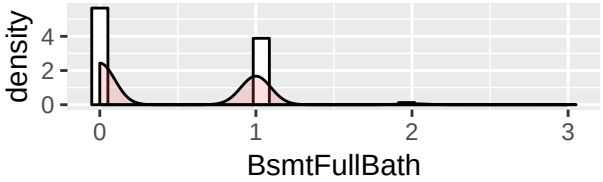


BsmtFullBath

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0000	0.0000	0.0000	0.4299	1.0000	3.000

Kurtosis: −0.738 / Skew: 0.623

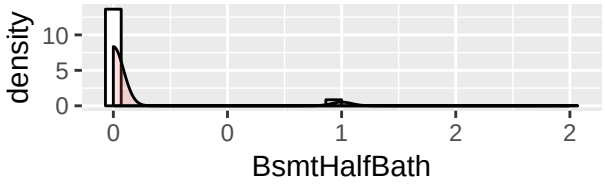


BsmtHalfBath

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max
0.00000	0.00000	0.00000	0.06136	0.00000	2

Kurtosis: 14.8 / Skew: 3.93

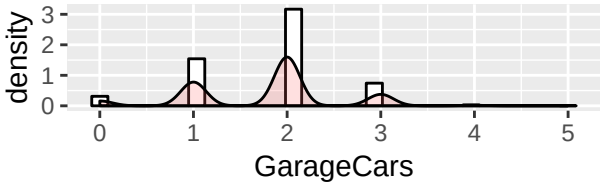


GarageCars

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.000	1.000	2.000	1.767	2.000	5.000

Kurtosis: 0.234 / Skew: −0.218

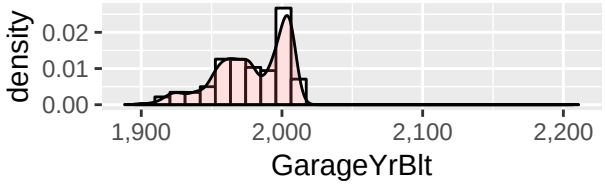


GarageYrBlt

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	M
1895	1960	1979	1978	2002	220

Kurtosis: 1.8 / Skew: −0.382

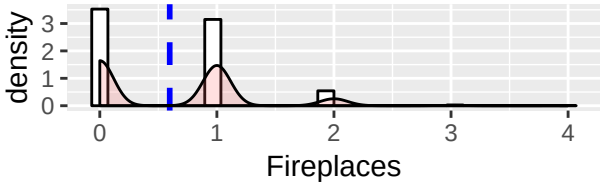


Fireplaces

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0000	0.0000	1.0000	0.5971	1.0000	4.000

Kurtosis: 0.0721 / Skew: 0.733

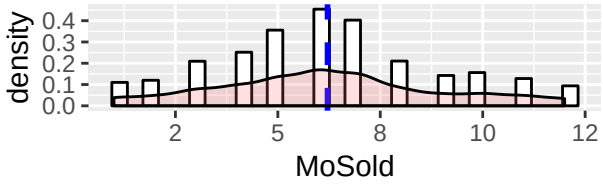


MoSold

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	4.000	6.000	6.213	8.000	12.000

Kurtosis: −0.457 / Skew: 0.196

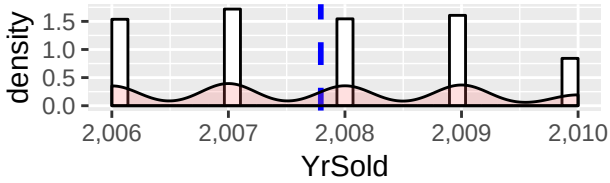


YrSold

Summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
2006	2007	2008	2008	2009	2010

Kurtosis: −1.16 / Skew: 0.132




```
rm(gs)
```

Limpieza y preparación de los datos

En general se ha intentado no eliminar observaciones, ya que el conjunto de datos es muy reducido. Se han realizado las siguientes tareas:

Verificación de contenido de datos campos Nominales y Ordinales

En primer lugar, convertimos variables factor a texto, para poder buscar y corregir valores.

```
dsDataAll <- dsDataAll %>%
  mutate_if(is.factor, as.character)

# Dejamos el indicador de entrenamiento como factor
dsDataAll$indTrain <- as.factor(dsDataAll$indTrain)
```

Cruzo los valores existentes en la documentación con los datos de los ficheros.

```
# Dataset con el nombre de los campos ordinales y nominales
dsCamposOrdinalesNominal <- dsCampos %>%
  filter(Tipo=="Ordinal" | Tipo=="Nominal")  %>%
```

```
  select(Campo)

dsCamposValorOriginales <- select(dsDataAll, c("Id",c(dsCamposOrdinalesNominal$Campo))) %>%
  gather("Campo","Valor",c(dsCamposOrdinalesNominal$Campo)) %>%
  na.omit() %>%
  arrange(Id)

# Busco valores que no concuerdan con especificaciones
dsCamposValorOriginales %>%
  anti_join(dsCamposValor,  by = c("Campo","Valor")) %>%
  group_by(Campo, Valor) %>%
  tally()
```

```
## # A tibble: 8 x 3
## # Groups:   Campo [4]
##   Campo       Valor       n
##   <chr>       <chr>   <int>
## 1 Exterior1st Wd Sdng   411
## 2 Exterior1st WdShing    56
## 3 Exterior2nd Brk Cmn    22
## 4 Exterior2nd CmentBd   126
## 5 Exterior2nd Wd Sdng   391
## 6 Exterior2nd Wd Shng    81
## 7 MSZoning    C (all)    25
## 8 RoofMatl    Tar&Grv    23
```

CORRECCIÓN DE ERRORES

```
# Normalizar valores para los campos Exterior1st / Exterior2nd
dsDataAll <- dsDataAll %>% mutate(Exterior1st = ifelse(Exterior1st=="WdShing","WdShng",Exterior1st))
dsDataAll <- dsDataAll %>% mutate(Exterior1st = ifelse(Exterior1st=="Wd Sdng","WdSdng",Exterior1st))
dsDataAll <- dsDataAll %>% mutate(Exterior1st = ifelse(Exterior1st=="Wd Shng","WdShng",Exterior1st))

dsDataAll <- dsDataAll %>% mutate(Exterior2nd = ifelse(Exterior2nd=="CmentBd","CemntBd",Exterior2nd))
dsDataAll <- dsDataAll %>% mutate(Exterior2nd = ifelse(Exterior2nd=="Wd Sdng","WdSdng",Exterior2nd))
dsDataAll <- dsDataAll %>% mutate(Exterior2nd = ifelse(Exterior2nd=="Wd Shng","WdShng",Exterior2nd))
dsDataAll <- dsDataAll %>% mutate(Exterior2nd = ifelse(Exterior2nd=="Brk Cmn","BrkComm",Exterior2nd))

# MSZoning  C (all) -> Cambio valor a C
dsDataAll %>%
  select(MSZoning) %>%
  na.omit() %>%
  count(MSZoning)

dsDataAll <- dsDataAll %>% mutate(MSZoning = ifelse(MSZoning=="C (all)","C",MSZoning))

# RoofMatl  Tar&Grv
dsDataAll %>%
  select(RoofMatl) %>%
  na.omit() %>%
  count(RoofMatl)

dsDataAll <- dsDataAll %>% mutate(RoofMatl = ifelse(RoofMatl=="Tar&Grv","Tar",RoofMatl))
```

```
# verificación Neighborhood=="NAmes"
# Neighborhood=="NAmes" -> cambio valor en excel a NAmes para Names: North Ames
# filter(dsDataAll,Neighborhood=="NAmes")

# BldgType -> cambio valor en excel a Twnhs para BldgType: TwnhsI - Townhouse Inside Unit
# verifico valores en datos
filter(dsDataAll,grepl('Twnhs', BldgType)) %>%
    select(BldgType) %>%
    count(BldgType)
```

Verificamos que no exista valores no contemplados.

```
# VERIFICACIÓN

dsCamposValorOriginales <- select(dsDataAll, c("Id",c(dsCamposOrdinalesNominal$Campo))) %>%
  gather("Campo","Valor",c(dsCamposOrdinalesNominal$Campo)) %>%
  na.omit() %>%
  arrange(Id)

# Busco valores que no concuerdan con especificaciones
dsCamposValorOriginales %>%
  anti_join(dsCamposValor,  by = c("Campo","Valor"))
```

```
## [1] Id    Campo Valor
## <0 rows> (or 0-length row.names)
```

```
rm(dsCamposValorOriginales)
rm(dsCamposOrdinalesNominal)
```

Valores faltantes - Missing Data

Muchos algoritmos no aceptan observaciones con valores no definidos (NA), por lo que, es necesario encontrarlos y darles una solución, se puede: * Eliminar observaciones que estén incompletas: dado que existen pocos datos esta opción no se ha usado. * Eliminar variables que contengan valores ausentes. * Estimar los valores ausentes empleando el resto de información disponible (imputación). Identifico valores faltantes, con el porcentaje que suponen frente al total de observaciones.

```
missingData <- dsDataAll %>%
  summarise_all(funs(sum(is.na(.)))) %>%
  gather("column") %>%
  rename(NumNAs = value) %>%
  mutate(PrcNAs = NumNAs/nrow(dsDataAll)) %>%
  filter(NumNAs!=0) %>%
  arrange(desc(PrcNAs))

missingData
```

```
##          column NumNAs       PrcNAs
## 1        PoolQC   2909 0.9965741692
## 2   MiscFeature   2814 0.9640287770
```

```
## 3         Alley   2721 0.9321685509
## 4         Fence   2348 0.8043850634
## 5     SalePrice   1459 0.4998287085
## 6   FireplaceQu   1420 0.4864679685
## 7   LotFrontage    486 0.1664953751
## 8   GarageYrBlt    159 0.0544707091
## 9  GarageFinish    159 0.0544707091
## 10   GarageQual    159 0.0544707091
## 11   GarageCond    159 0.0544707091
## 12   GarageType    157 0.0537855430
## 13     BsmtCond     82 0.0280918123
## 14 BsmtExposure     82 0.0280918123
## 15     BsmtQual     81 0.0277492292
## 16 BsmtFinType2     80 0.0274066461
## 17 BsmtFinType1     79 0.0270640630
## 18   MasVnrType     24 0.0082219938
## 19   MasVnrArea     23 0.0078794108
## 20     MSZoning      4 0.0013703323
## 21    Utilities      2 0.0006851662
## 22 BsmtFullBath      2 0.0006851662
## 23 BsmtHalfBath      2 0.0006851662
## 24   Functional      2 0.0006851662
## 25  Exterior1st      1 0.0003425831
## 26  Exterior2nd      1 0.0003425831
## 27   BsmtFinSF1      1 0.0003425831
## 28   BsmtFinSF2      1 0.0003425831
## 29    BsmtUnfSF      1 0.0003425831
## 30  TotalBsmtSF      1 0.0003425831
## 31   Electrical      1 0.0003425831
## 32  KitchenQual      1 0.0003425831
## 33   GarageCars      1 0.0003425831
## 34   GarageArea      1 0.0003425831
## 35     SaleType      1 0.0003425831
```

Las variables con un porcentaje de valores asuntes muy alto (>80%) las excluimos del modelo, ya que pueden dar errores al realizar subconjuntos de datos pare entrenar y validar los modelos.

```
# PoolQC - Calidad de la piscina
# MiscFeature -  características varias no cubiertas en otras categorías
# Alley - tipo de acceso al callejón
# Fence - calidad de la cerca

eliminar <- filter(missingData, PrcNAs > 0.80) %>% select(column)

dsDataAll <- dsDataAll %>%
    select(-c(eliminar$column))

rm(eliminar)
```

Del resto de valores pendientes realizo estudio y modifico valores faltantes.

```
#FireplaceQu
# a <- dsDataAll %>% filter(!is.na(FireplaceQu)) %>% select(FireplaceQu, Fireplaces)
# b <- dsDataAll %>% filter(Fireplaces!=0) %>% select(FireplaceQu, Fireplaces)
```

```r
# b %>% filter(is.na(FireplaceQu))

# Si no tiene chimenea asigno un valor None
dsDataAll <- mutate(dsDataAll, FireplaceQu = ifelse(is.na(FireplaceQu),"None",FireplaceQu))


#LotFrontage -  pies lineales de calle conectados a la propiedad
# a <- dsDataAll %>% filter(!is.na(LotFrontage)) %>% select(LotFrontage, MSSubClass)
# b <- train %>% filter(!is.na(LotFrontage)) %>% select(LotFrontage, MSSubClass, SalePrice)
# ggplot(b, aes(x=LotFrontage, y=SalePrice, color=MSSubClass)) + geom_point()

# Si no tiene valor asigno la media
dsDataAll <- mutate(dsDataAll, LotFrontage = ifelse(is.na(LotFrontage),mean(dsDataAll$LotFrontage,na.rm

# GarageYrBlt/GarageFinish/GarageQual/GarageCond/GarageType
# a <- dsDataAll %>% filter(is.na(GarageYrBlt)) %>%
#   select(GarageCars, GarageArea, GarageYrBlt, GarageFinish, GarageQual, GarageCond, GarageType)

#Ordinales asigno texto None
dsDataAll <- mutate(dsDataAll, GarageCond = ifelse(is.na(GarageCond),"None",GarageCond))
dsDataAll <- mutate(dsDataAll, GarageQual = ifelse(is.na(GarageQual),"None",GarageQual))
dsDataAll <- mutate(dsDataAll, GarageFinish = ifelse(is.na(GarageFinish),"None",GarageFinish))

#Nominales None
dsDataAll <- mutate(dsDataAll, GarageType = ifelse(is.na(GarageType),"None",GarageType))

#Discretas y continuas 0 no tienen garage
dsDataAll <- mutate(dsDataAll, GarageYrBlt = ifelse(is.na(GarageYrBlt),0,GarageYrBlt))
dsDataAll <- mutate(dsDataAll, GarageCars = ifelse(is.na(GarageCars),0,GarageCars))
dsDataAll <- mutate(dsDataAll, GarageArea = ifelse(is.na(GarageArea),0,GarageArea))

#TotalBsmtSF 0
dsDataAll <- mutate(dsDataAll, TotalBsmtSF = ifelse(is.na(TotalBsmtSF),0,TotalBsmtSF))

# BsmtCond / BsmtExposure / BsmtQual / BsmtFinType2 / BsmtFinType1
# a <- dsDataAll %>% filter(TotalBsmtSF==0) %>%
#  select(TotalBsmtSF
#    ,BsmtFinSF1
#    ,BsmtFinType2
#    ,BsmtFinSF2
#    ,BsmtUnfSF
#    ,BsmtQual
#    ,BsmtCond
#    ,BsmtExposure
#    ,BsmtFinType1
#    ,BsmtFullBath
#    ,BsmtHalfBath)

#Discretas y continuas 0 no tienen garage
dsDataAll <- mutate(dsDataAll, BsmtFinSF1 = ifelse(is.na(BsmtFinSF1),0,BsmtFinSF1))
dsDataAll <- mutate(dsDataAll, BsmtFinSF2 = ifelse(is.na(BsmtFinSF2),0,BsmtFinSF2))
dsDataAll <- mutate(dsDataAll, BsmtUnfSF = ifelse(is.na(BsmtUnfSF),0,BsmtUnfSF))
dsDataAll <- mutate(dsDataAll, BsmtFullBath = ifelse(is.na(BsmtFullBath),0,BsmtFullBath))
```

```
dsDataAll <- mutate(dsDataAll, BsmtHalfBath = ifelse(is.na(BsmtHalfBath),0,BsmtHalfBath))

#Ordinales asigno texto None
dsDataAll <- mutate(dsDataAll, BsmtFinType2 = ifelse(is.na(BsmtFinType2),"None",BsmtFinType2))
dsDataAll <- mutate(dsDataAll, BsmtQual = ifelse(is.na(BsmtQual),"None",BsmtQual))
dsDataAll <- mutate(dsDataAll, BsmtCond = ifelse(is.na(BsmtCond),"None",BsmtCond))
dsDataAll <- mutate(dsDataAll, BsmtExposure = ifelse(is.na(BsmtExposure),"None",BsmtExposure))
dsDataAll <- mutate(dsDataAll, BsmtFinType1 = ifelse(is.na(BsmtFinType1),"None",BsmtFinType1))

#MasVnrType  tipo de chapa de albañilería
# summary(as.factor(dsDataAll$MasVnrType))
dsDataAll <- mutate(dsDataAll, MasVnrType = ifelse(is.na(MasVnrType),"None",MasVnrType))

#MasVnrArea  área de revestimiento de mampostería en pies cuadrados
# summary(dsDataAll$MasVnrArea)
dsDataAll <- mutate(dsDataAll, MasVnrArea = ifelse(is.na(MasVnrArea),0,MasVnrArea))

#MSZoning    la clasificación general de zonificación (Nominal)
# summary(as.factor(dsDataAll$MSZoning))
dsDataAll <- mutate(dsDataAll, MSZoning = ifelse(is.na(MSZoning),"RL",MSZoning))

#Utilities  tipo de utilidades disponibles
# summary(dsDataAll$Utilities)
# Descartamos la variable Utilities ya que todas las filas tienen el mismo valor menos 3 y dos son NAs
dsDataAll <- select(dsDataAll, -Utilities)

#Functional  calificación de funcionalidad del hogar
# summary(as.factor(dsDataAll$Functional))
dsDataAll <- mutate(dsDataAll, Functional = ifelse(is.na(Functional),"Typ",Functional))

#Exterior1st  cubierta exterior en la casa
# summary(as.factor(dsDataAll$Exterior1st))
dsDataAll <- mutate(dsDataAll, Exterior1st = ifelse(is.na(Exterior1st),"VinylSd",Exterior1st))

#Exterior2nd  revestimiento exterior de la casa (si hay más de un material)
#summary(as.factor(dsDataAll$Exterior2nd))
dsDataAll <- mutate(dsDataAll, Exterior2nd = ifelse(is.na(Exterior2nd),"VinylSd",Exterior2nd))

#Electrical -  sistema eléctrico
#summary(as.factor(dsDataAll$Electrical))
dsDataAll <- mutate(dsDataAll, Electrical = ifelse(is.na(Electrical),"SBrkr",Electrical))

#KitchenQual
#summary(as.factor(dsDataAll$KitchenQual))
dsDataAll <- mutate(dsDataAll, KitchenQual = ifelse(is.na(KitchenQual),"TA",KitchenQual))

#SaleType
#summary(as.factor(dsDataAll$SaleType))
dsDataAll <- mutate(dsDataAll, SaleType = ifelse(is.na(SaleType),"WD",SaleType))
```

Verificamos que no quedan pendientes valores faltantes, solo debe de quedar SalePrice para el conjunto de test.

```
# VERIFICACIÓN

missingData <- dsDataAll %>%
  summarise_all(funs(sum(is.na(.)))) %>%
  gather("column") %>%
  rename(NumNAs = value) %>%
  mutate(PrcNAs = NumNAs/nrow(dsDataAll)) %>%
  filter(NumNAs!=0) %>%
  arrange(desc(PrcNAs))

missingData
```

```
##      column NumNAs    PrcNAs
## 1 SalePrice   1459 0.4998287
```

```
rm(missingData)
```

Verificación del tipo de datos

Determinamos los tipos de variables: *Continuas* Discretas *Ordinales* Nominales

Cruzo campos documentación con los campos originales para verificar nombres de campos y tipos

```
# Obtengo campos originales de los ficheros
dsCamposOriginales <- data.frame(unlist(sapply(dsDataAll, class))) %>%
  select(Tipo = 1) %>%
  rownames_to_column("Campo")

# Verifico nombres de campos y que existen todos
dsCamposOriginales %>%
  anti_join(dsCampos,  by = c("Campo"))
```

```
##      Campo    Tipo
## 1 indTrain  factor
## 2       Id integer
```

```
# Verificamos que estan todos los campos (salen las variables ya eliminadas)
dsCampos %>%
  anti_join(dsCamposOriginales,  by = c("Campo"))  %>%
  select(Campo)
```

```
##         Campo
## 1   Utilities
## 2      PoolQC
## 3       Fence
## 4       Alley
## 5 MiscFeature
```

Variables Continuas

```r
dsCamposOriginales %>%
  inner_join(dsCampos,  by = c("Campo")) %>%
  select(Campo, Tipo.x, Tipo.y) %>%
  filter(Tipo.y == "Continua" & Tipo.x != "integer" & Tipo.x != "numeric")
```

```
## [1] Campo  Tipo.x Tipo.y
## <0 rows> (or 0-length row.names)
```

Variables Discretas

```r
dsCamposOriginales %>%
  inner_join(dsCampos,  by = c("Campo")) %>%
  select(Campo, Tipo.x, Tipo.y) %>%
  filter(Tipo.y == "Discreta" & Tipo.x != "integer" & Tipo.x != "numeric")
```

```
## [1] Campo  Tipo.x Tipo.y
## <0 rows> (or 0-length row.names)
```

Variables Ordinales

Convertiremos las variables ordinales a numéricas basandonos en el orden establecido en la documentación.

Primero verifico si todas las variables ordinales son del tipo character

```r
dsCamposOriginales %>%
  inner_join(dsCampos,  by = c("Campo")) %>%
  select(Campo, Tipo.x, Tipo.y) %>%
  filter(Tipo.y == "Ordinal" & Tipo.x != "character")
```

```
##         Campo  Tipo.x  Tipo.y
## 1 OverallQual integer Ordinal
## 2 OverallCond integer Ordinal
```

```r
# OverallQual / OverallCond -> se mantienen con el orden indicado ya estan como númericas.
```

OverallQual / OverallCond son enteras y se mantienen con el orden indicado.

Al resto de variables les asigno los niveles de la documentación y posteriormente las convierto a numéricas, como las listas en la documentación están ordenadas de mayor a menor, utilizo la función rev para invertir el orden.

Si a alguna variable le he añadido el valor "None", este valor lo pongo el primero y al pasarlo a numérico resto 1, por lo que el valor None pasara a 0.

```r
# ExterQual -  calidad del material exterior
#  Ex: Excellent
#  Gd: Good
#  TA: Average/Typical
#  Fa: Fair
#  Po: Poor

dsDataAll$ExterQual <- factor(dsDataAll$ExterQual, levels = rev(c("Ex","Gd","TA","Fa","Po")))
```

```r
dsDataAll$ExterQual <- as.numeric(c(dsDataAll$ExterQual))

#ExterCond   Condición actual del material en el exterior.
#Ex: Excellent
#Gd: Good
#TA: Average/Typical
#Fa: Fair
#Po: Poor

dsDataAll$ExterCond <- factor(dsDataAll$ExterCond, levels = rev(c("Ex","Gd","TA","Fa","Po")))
dsDataAll$ExterCond <- as.numeric(c(dsDataAll$ExterCond))

#LotShape    forma general de propiedad
#Reg: Regular
#IR1: Slightly irregular
#IR2: Moderately Irregular
#IR3: Irregular

dsDataAll$LotShape <- factor(dsDataAll$LotShape, levels = rev(c("Reg","IR1","IR2","IR3")))
dsDataAll$LotShape <- as.numeric(c(dsDataAll$LotShape))

#LandSlope   pendiente de la propiedad
#Gtl: Gentle slope
#Mod: Moderate Slope
#Sev: Severe Slope

dsDataAll$LandSlope <- factor(dsDataAll$LandSlope, levels = rev(c("Gtl","Mod","Sev")))
dsDataAll$LandSlope <- as.numeric(c(dsDataAll$LandSlope))

#BsmtQual    altura del sótano
#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

dsDataAll$BsmtQual <- factor(dsDataAll$BsmtQual, levels = rev(c("Ex","Gd","TA","Fa","Po","None")))
dsDataAll$BsmtQual <- as.numeric(c(dsDataAll$BsmtQual))-1

#BsmtCond    estado general del sótano
#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

dsDataAll$BsmtCond <- factor(dsDataAll$BsmtCond, levels = rev(c("Ex","Gd","TA","Fa","Po","None")))
dsDataAll$BsmtCond <- as.numeric(c(dsDataAll$BsmtCond))-1

#BsmtExposure    paredes de sótano a nivel de jardín o de huelga
#Gd: Good Exposure
```

```
#Av: Average Exposure (split levels or foyers typically score average or above)
#Mn: Mimimum Exposure
#No: No Exposure
#NA: No Basement

dsDataAll$BsmtExposure <- factor(dsDataAll$BsmtExposure, levels = rev(c("Gd","Av","Mn","No","None")))
dsDataAll$BsmtExposure <- as.numeric(c(dsDataAll$BsmtExposure))-1

#BsmtFinType1    Calidad del área terminada del sótano
#GLQ: Good Living Quarters
#ALQ: Average Living Quarters
#BLQ: Below Average Living Quarters
#Rec: Average Rec Room
#LwQ: Low Quality
#Unf: Unfinshed
#NA: No Basement

dsDataAll$BsmtFinType1 <- factor(dsDataAll$BsmtFinType1, levels = rev(c("GLQ","ALQ","BLQ","Rec","LwQ","
dsDataAll$BsmtFinType1 <- as.numeric(c(dsDataAll$BsmtFinType1))-1

#BsmtFinType2    Calidad de la segunda área terminada (si está presente)
#GLQ: Good Living Quarters
#ALQ: Average Living Quarters
#BLQ: Below Average Living Quarters
#Rec: Average Rec Room
#LwQ: Low Quality
#Unf: Unfinshed
#NA: No Basement

dsDataAll$BsmtFinType2 <- factor(dsDataAll$BsmtFinType2, levels = rev(c("GLQ","ALQ","BLQ","Rec","LwQ","
dsDataAll$BsmtFinType2 <- as.numeric(c(dsDataAll$BsmtFinType2))-1

#HeatingQC   calidad y condición de calefacción
#Ex: Excellent
#Gd: Good
#TA: Average/Typical
#Fa: Fair
#Po: Poor

dsDataAll$HeatingQC <- factor(dsDataAll$HeatingQC, levels = rev(c("Ex","Gd","TA","Fa","Po")))
dsDataAll$HeatingQC <- as.numeric(c(dsDataAll$HeatingQC))

#Electrical  sistema eléctrico
#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

dsDataAll$Electrical <- factor(dsDataAll$Electrical, levels = rev(c("SBrkr","FuseA","FuseF","FuseP","Mi
dsDataAll$Electrical <- as.numeric(c(dsDataAll$Electrical))

#KitchenQual     calidad de cocina
```

```r
#Ex: Excellent
#Gd: Good
#TA: Typical/Average
#Fa: Fair
#Po: Poor

dsDataAll$KitchenQual <- factor(dsDataAll$KitchenQual, levels = rev(c("Ex","Gd","TA","Fa","Po")))
dsDataAll$KitchenQual <- as.numeric(c(dsDataAll$KitchenQual))

#Functional  calificación de funcionalidad del hogar
#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

dsDataAll$Functional <- factor(dsDataAll$Functional, levels = rev(c("Typ","Min1","Min2","Mod","Maj1","Ma
dsDataAll$Functional <- as.numeric(c(dsDataAll$Functional))

#FireplaceQu     calidad de chimenea
#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

dsDataAll$FireplaceQu <- factor(dsDataAll$FireplaceQu, levels = rev(c("Ex","Gd","TA","Fa","Po","None")))
dsDataAll$FireplaceQu <- as.numeric(c(dsDataAll$FireplaceQu))-1

#GarageFinish    acabado interior del garaje
#Fin: Finished
#RFn: Rough Finished
#Unf: Unfinished
#NA: No Garage

dsDataAll$GarageFinish <- factor(dsDataAll$GarageFinish, levels = rev(c("Fin","RFn","Unf","None")))
dsDataAll$GarageFinish <- as.numeric(c(dsDataAll$GarageFinish))-1

#GarageQual  calidad de garaje
#Ex: Excellent
#Gd: Good
#TA: Typical/Average
#Fa: Fair
#Po: Poor
#NA: No Garage

dsDataAll$GarageQual <- factor(dsDataAll$GarageQual, levels = rev(c("Ex","Gd","TA","Fa","Po","None")))
dsDataAll$GarageQual <- as.numeric(c(dsDataAll$GarageQual))-1
```

```r
#GarageCond  condición del garaje
#Ex: Excellent
#Gd: Good
#TA: Typical/Average
#Fa: Fair
#Po: Poor
#NA: No Garage

dsDataAll$GarageCond <- factor(dsDataAll$GarageCond, levels = rev(c("Ex","Gd","TA","Fa","Po","None")))
dsDataAll$GarageCond <- as.numeric(c(dsDataAll$GarageCond))-1

#PavedDrive  entrada pavimentada
#Y: Paved
#P: Partial Pavement
#N: Dirt/Gravel

dsDataAll$PavedDrive <- factor(dsDataAll$PavedDrive, levels = rev(c("Y","P","N")))
dsDataAll$PavedDrive <- as.numeric(c(dsDataAll$PavedDrive))

#PoolQC  calidad de la piscina (Eliminada)
```

Verificamos que todos los campos ordinales sean numéricos.

```r
## VERIFICACIÓN

# Obtengo campos originales una vez trasformados
dsCamposOriginales <- data.frame(unlist(sapply(dsDataAll, class))) %>%
  select(Tipo = 1) %>%
  rownames_to_column("Campo")

# Verifico que todos han quedado numéricos
dsCamposOriginales %>%
  inner_join(dsCampos,  by = c("Campo")) %>%
  select(Campo, Tipo.x, Tipo.y) %>%
  filter(Tipo.y == "Ordinal" & Tipo.x != "integer" & Tipo.x != "numeric")
```

```
## [1] Campo  Tipo.x Tipo.y
## <0 rows> (or 0-length row.names)
```

Variables Nominales convierto a factor

```r
dsCamposOriginales %>%
  inner_join(dsCampos,  by = c("Campo")) %>%
  select(Campo, Tipo.x, Tipo.y) %>%
  filter(Tipo.y == "Nominal" & Tipo.x != "character")
```

```
##        Campo  Tipo.x  Tipo.y
## 1 MSSubClass integer Nominal
```

```r
dsDataAll$MSSubClass <- as.character(dsDataAll$MSSubClass)
```

```
# Verifico que todos los campos caracter son Nominales u Ordinales
dsCamposOriginales %>%
  inner_join(dsCampos,  by = c("Campo")) %>%
  select(Campo, Tipo.x, Tipo.y) %>%
  filter(Tipo.y != "Ordinal" & Tipo.y != "Nominal" & Tipo.x == "character")
```

```
## [1] Campo  Tipo.x Tipo.y
## <0 rows> (or 0-length row.names)
```

```
# Paso a factor
dsDataAll <- dsDataAll %>%
    mutate_if(is.character, as.factor)


## VERIFICACIÓN

# Obtengo campos originales una vez trasformados
dsCamposOriginales <- data.frame(unlist(sapply(dsDataAll, class))) %>%
  select(Tipo = 1) %>%
  rownames_to_column("Campo")

# Verifico que todos han quedado factor
dsCamposOriginales %>%
  inner_join(dsCampos,  by = c("Campo")) %>%
  select(Campo, Tipo.x, Tipo.y) %>%
  filter(Tipo.y == "Nominal" & Tipo.x != "factor")
```

```
## [1] Campo  Tipo.x Tipo.y
## <0 rows> (or 0-length row.names)
```

```
rm(dsCamposOriginales)
rm(dsCamposValor)
```

Ls variables con dos valores se convierten directamente a numéricas indicando 0 ausencia y 1 presencia del valor, CentralAir y Street que pasa a llamarse StreetPave

$ Street : Factor w/ 2 levels “Grvl”,“Pave” $ CentralAir : Factor w/ 2 levels “N”,“Y”

```
#  Grvl:  12
#  Pave:2907

dsDataAll$StreetPave[dsDataAll$Street != "Pave"] <- "0"
dsDataAll$StreetPave[dsDataAll$Street == "Pave"] <- "1"
dsDataAll$StreetPave <- as.numeric(dsDataAll$StreetPave)
dsDataAll <- select(dsDataAll, -Street)

# CentralAir
# Y:2723
# N: 196

dsDataAll$CentralAir <- as.character(dsDataAll$CentralAir)
dsDataAll$CentralAir[dsDataAll$CentralAir != "Y"] <- "0"
dsDataAll$CentralAir[dsDataAll$CentralAir == "Y"] <- "1"
dsDataAll$CentralAir <- as.numeric(dsDataAll$CentralAir)
```

Salvar progreso

```
# save(dsDataAll, file = './FO1_Datos/FO1_dsDataAll.RData')
# load('./FO1_Datos/FO1_dsDataAll.RData')
```

Outliers - Busqueda de valores atípicos

Tratamientos posibles:

- Retirar la fila
- Asigne el siguiente valor más cercano a la mediana en lugar del valor atípico

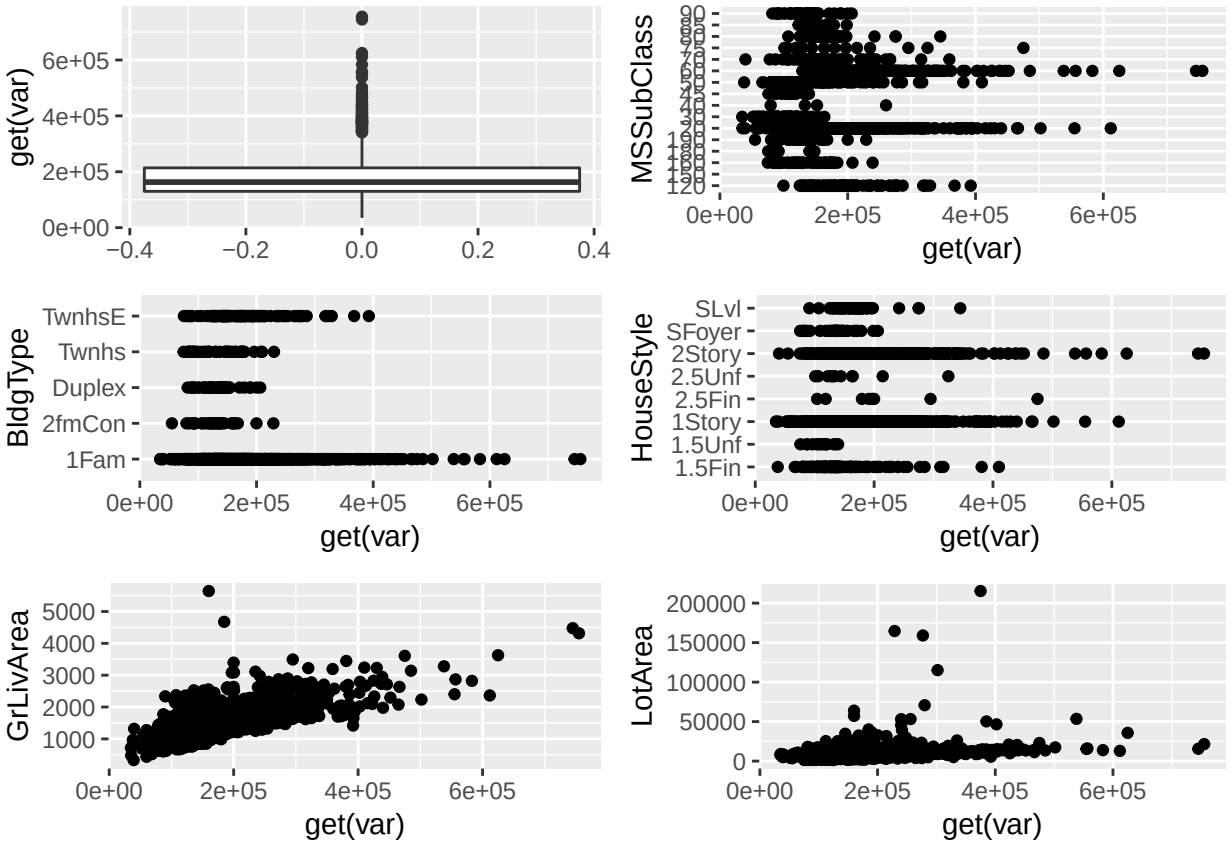
SalesPrice

No identifico valores a eliminar por precio, aunque si existen un par de valores que se pueden identificar como raros en función de su precio y su area, se estudiaran seguidamente.

```
# SalesPrice

summary(dsDataAll$SalePrice)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   34900  129975  163000  180921  214000  755000    1459
```

```
var <- "SalePrice"

a <- dsDataAll %>%
    select(var)

# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=MSSubClass)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=BldgType)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=HouseStyle)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=LotArea)) + geom_point()
  ,ncol=2)
```




```
# NO IDENTIFICO OUTLIERS

rm(var)
rm(a)
```

Resto de variables continuas

GrLivArea superficie habitable por encima del nivel del suelo (pies cuadrados)

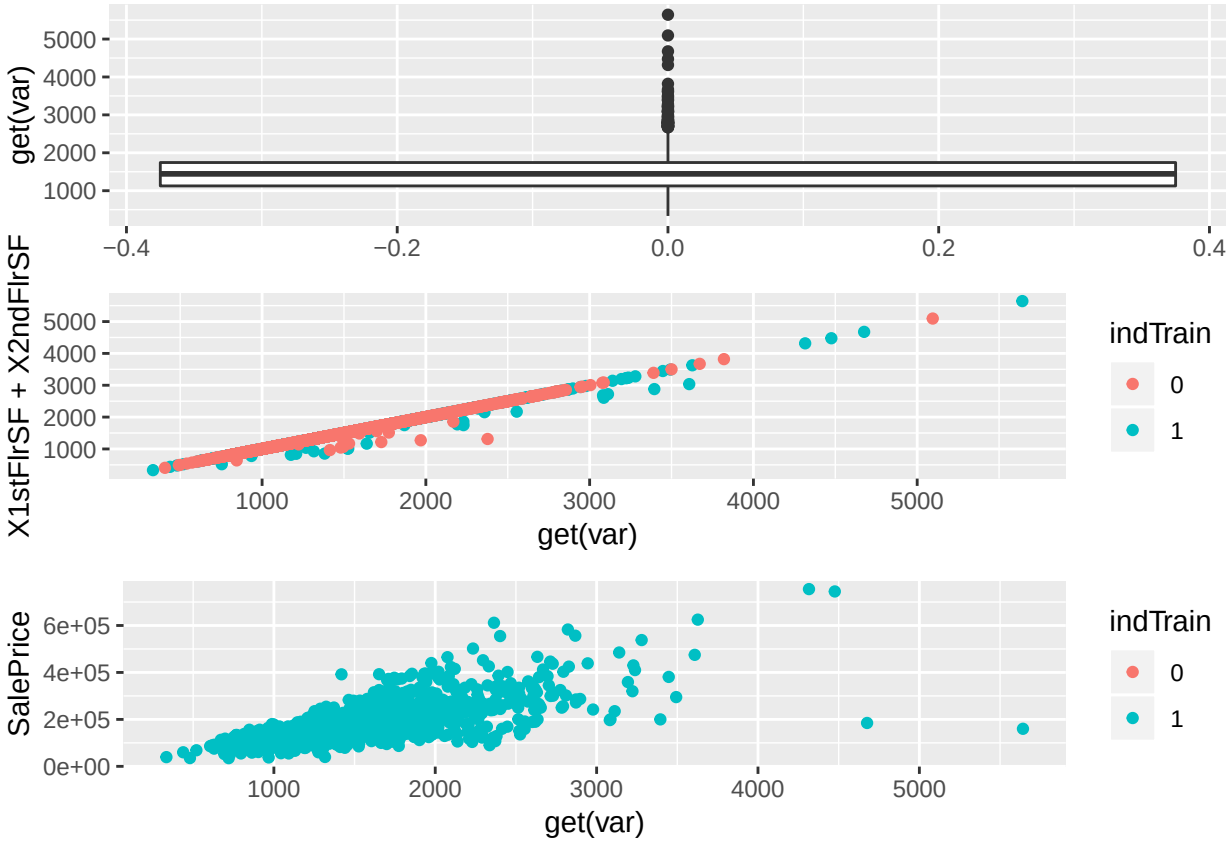
Existen 2 valores atípicos son muy altos para el precio que tienen en el conjunto de entrenamiento, estas filas se eliminarán al ser esta una variable principal para el proceso de predicción

```
summary(dsDataAll$GrLivArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334    1126    1444    1501    1744    5642
```

```
var <- "GrLivArea"

a <- dsDataAll %>%
    select(var)

# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
```

```
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=X1stFlrSF+X2ndFlrSF)) + geom_point(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point(aes(color = indTrain)) #solo conjunto T
  ,ncol=1)
```




```
# Existen 2 valores atípicos son muy altos para el precio que tienen en el conjunto de entrenamiento

# Selecciono las filas a eliminar
eliminar <- dsDataAll %>%
    filter(indTrain==1&GrLivArea>4500) %>%
    select(Id, GrLivArea, SalePrice, indTrain)

# dsDataAll %>%
#   inner_join(eliminar,by="Id")

dsDataAll <- dsDataAll %>%
  anti_join(eliminar,by="Id")

rm(eliminar)
rm(var)
rm(a)
```

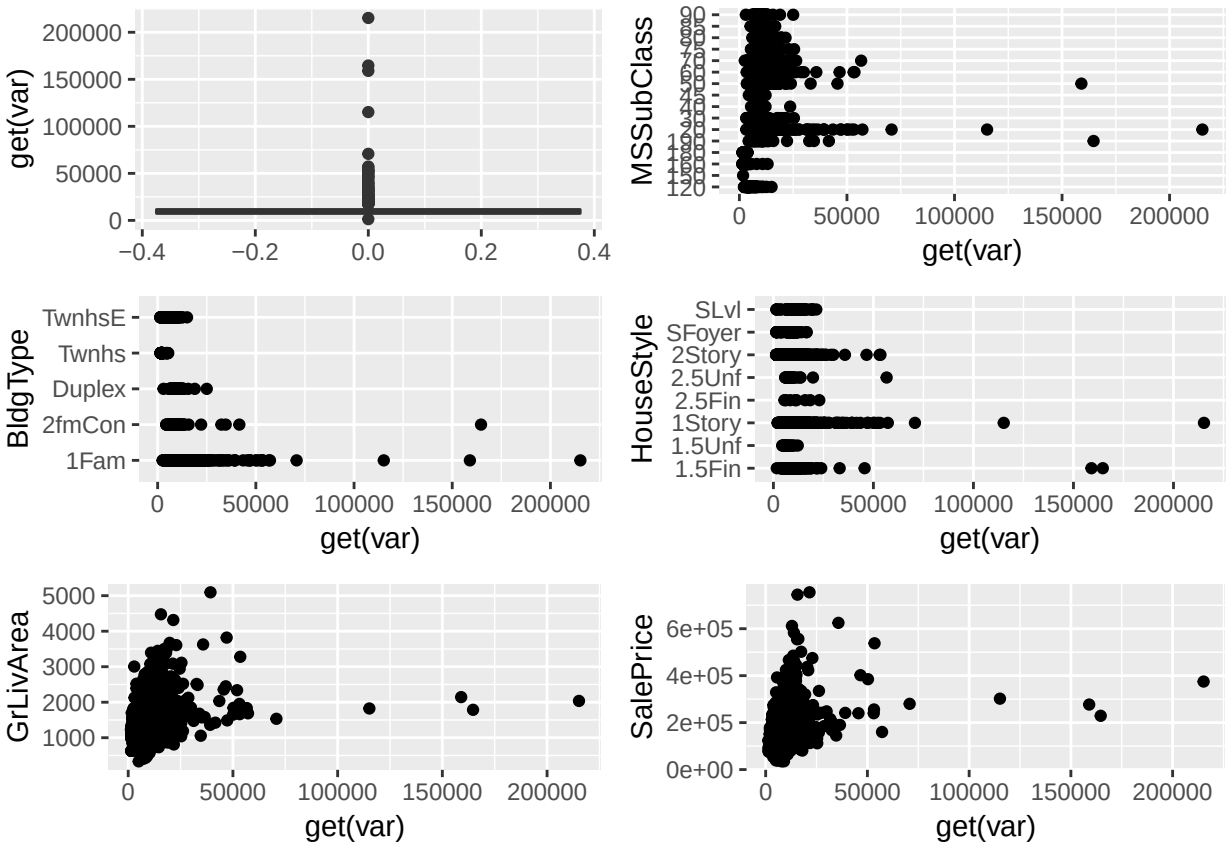
LotArea tamaño del lote en pies cuadrados

Existen 4 valores claramente fuera de rango, creo variable nueva actualizandolos con los valores con la mediana según el tipo de construcción

```r
summary(dsDataAll$LotArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1300    7476    9452   10139   11556  215245
```

```r
var <- "LotArea"

a <- dsDataAll %>%
    select(var)

# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=MSSubClass)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=BldgType)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=HouseStyle)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol = 2)
```




```r
# Existen 4 valores claramente fuera de rango,
# los actualizo con los valores con la mediana según el tipo de construcción
```

```r
# Calculo mediana por tipo de construcción
lotAreaMedian <- select(dsDataAll,BldgType,LotArea) %>%
    group_by(BldgType) %>%
    summarise(medianLotArea = median(LotArea))

f <- function(x){
  a <- as.numeric(lotAreaMedian[lotAreaMedian$BldgType==x,2])
  return(a)
}


# Seleccion Outliers
outlier_values <- as.data.frame(boxplot.stats(dsDataAll$LotArea)$out)
names(outlier_values) = "LotArea"
outlier_values$LotArea <- as.numeric(outlier_values$LotArea)
outlier_values <- outlier_values %>%
  arrange(desc(LotArea)) %>%
  top_n(4)

outlier_values
```

```
##   LotArea
## 1  215245
## 2  164660
## 3  159000
## 4  115149
```

```r
# Modificación directa
dsDataAll <- dsDataAll %>%
  rowwise() %>%
    mutate(LotArea = ifelse(LotArea>=115149,f(BldgType),LotArea))

# Modificación medianta uniones
# outlier_values <- outlier_values %>%
#   inner_join(data, by="LotArea") %>%
#   inner_join(lotAreaMedian, by="BldgType") %>%
#   select(Id, medianLotArea)
#
# data <- data %>%
#   left_join(outlier_values) %>%
#   mutate(LotArea = ifelse(is.na(medianLotArea),LotArea,medianLotArea)) %>%
#   select(-medianLotArea)

rm(outlier_values)
rm(lotAreaMedian)
rm(f)
rm(var)
rm(a)
```

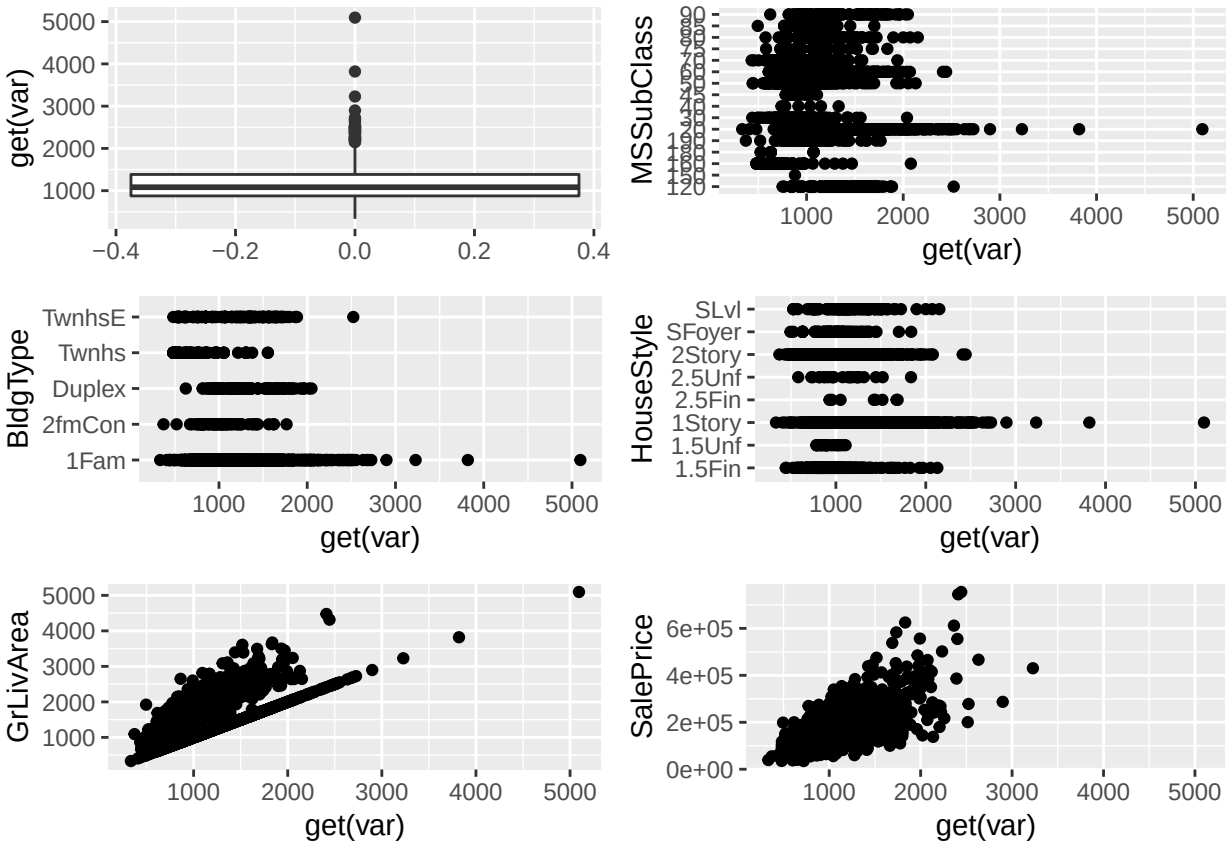
X1stFlrSF pies cuadrados del primer piso

NO IDENTIFICO OUTLIERS – Existian dos pero se han eliminado en el tratamiento GrLivArea

```r
summary(dsDataAll$X1stFlrSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334     876    1082    1158    1384    5095
```

```r
var <- "X1stFlrSF"

a <- dsDataAll %>%
    select(var)

# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=MSSubClass)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=BldgType)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=HouseStyle)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol = 2)
```




```r
# NO IDENTIFICO OUTLIERS -- Existian dos pero se han eliminado en el tratamiento GrLivArea

rm(var)
rm(a)
```

X2ndFlrSF pies cuadrados del segundo piso

NO IDENTIFICO OUTLIERS – Existian dos pero se han eliminado en el tratamiento GrLivArea

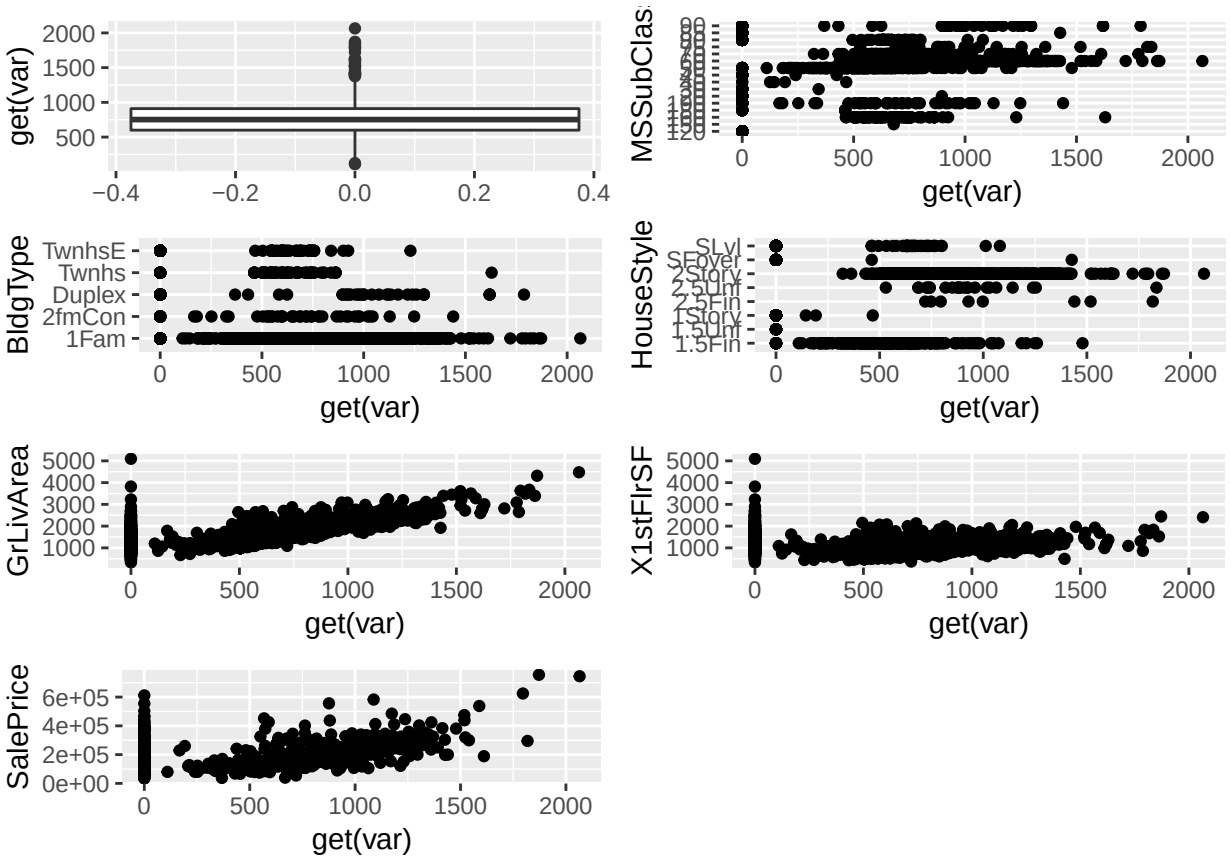
```
summary(dsDataAll$X2ndFlrSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0   335.9   704.0  2065.0
```

```
var <- "X2ndFlrSF"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$X2ndFlrSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   110.0   600.0   752.0   784.4   910.0  2065.0
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=MSSubClass)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=BldgType)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=HouseStyle)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=X1stFlrSF)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol = 2)
```




```
# NO IDENTIFICO OUTLIERS -- Existian dos pero se han eliminado en el tratamiento GrLivArea

rm(var)
rm(a)
```

LowQualFinSF pies cuadrados terminados de baja calidad (todos los pisos)

Parece que existe un par de valores extraños, creo variable nueva y actualizo a la mediana de todos los valores no cero

```
summary(dsDataAll$LowQualFinSF)
```

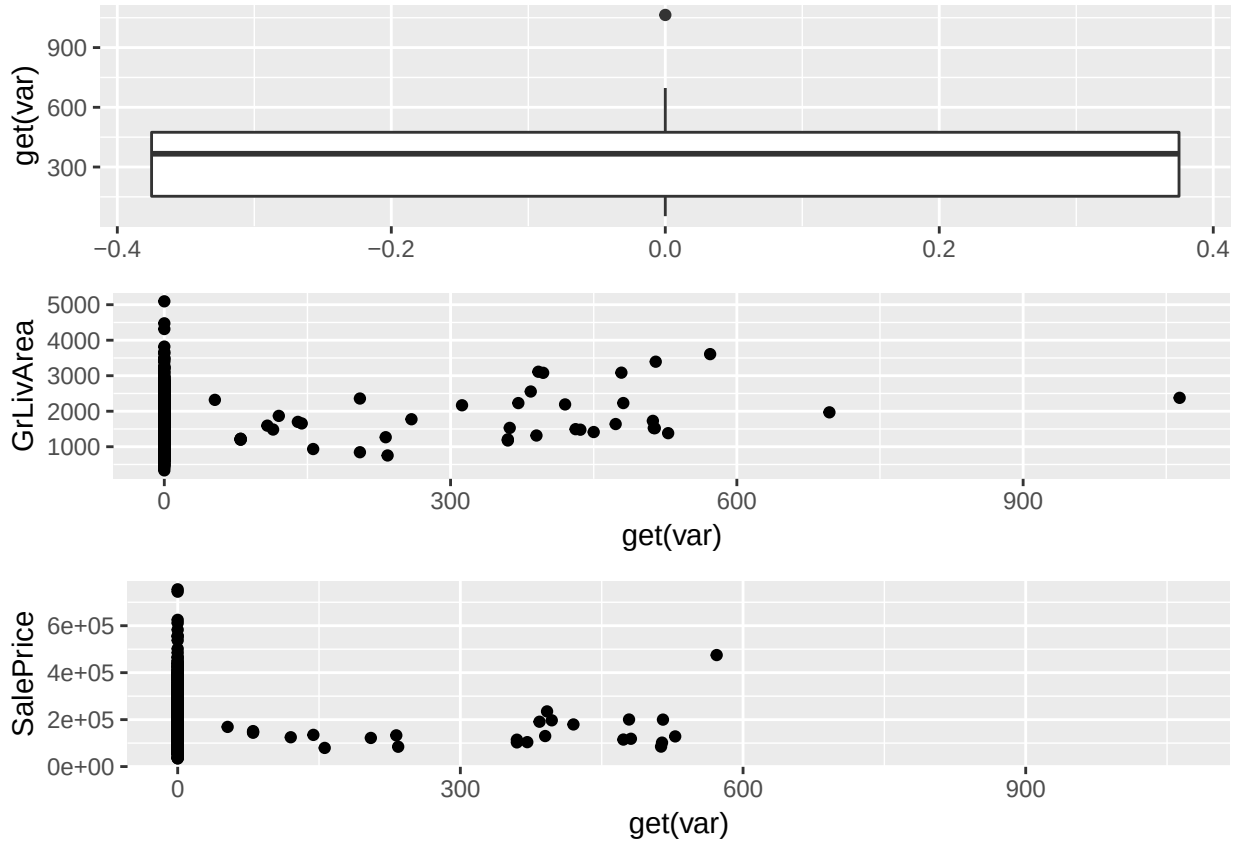
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.000    0.000    0.000    4.698    0.000 1064.000
```

```
var <- "LowQualFinSF"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a)
```

```
##   LowQualFinSF
##  Min.   :  53.0
```

```
##  1st Qu.: 153.0
##  Median : 366.5
##  Mean   : 342.6
##  3rd Qu.: 474.5
##  Max.   :1064.0
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=1)
```




```
# Parece que existe un par de valores extraños
# Actualizo a la mediana de todos los valores no cero
medianLowQualFinSF <- median(a$LowQualFinSF)

#select(data,Id,LowQualFinSF) %>% filter(LowQualFinSF>600)

# Modificación directa
dsDataAll <- dsDataAll %>%
  rowwise() %>%
    mutate(LowQualFinSF = ifelse(LowQualFinSF>600,medianLowQualFinSF,LowQualFinSF))
```

```
rm(medianLowQualFinSF)
rm(var)
rm(a)
```

MasVnrArea área de revestimiento de mampostería en pies cuadrados

Parece que existe un valor extraño actualizo a la mediana de todos los valores no cero

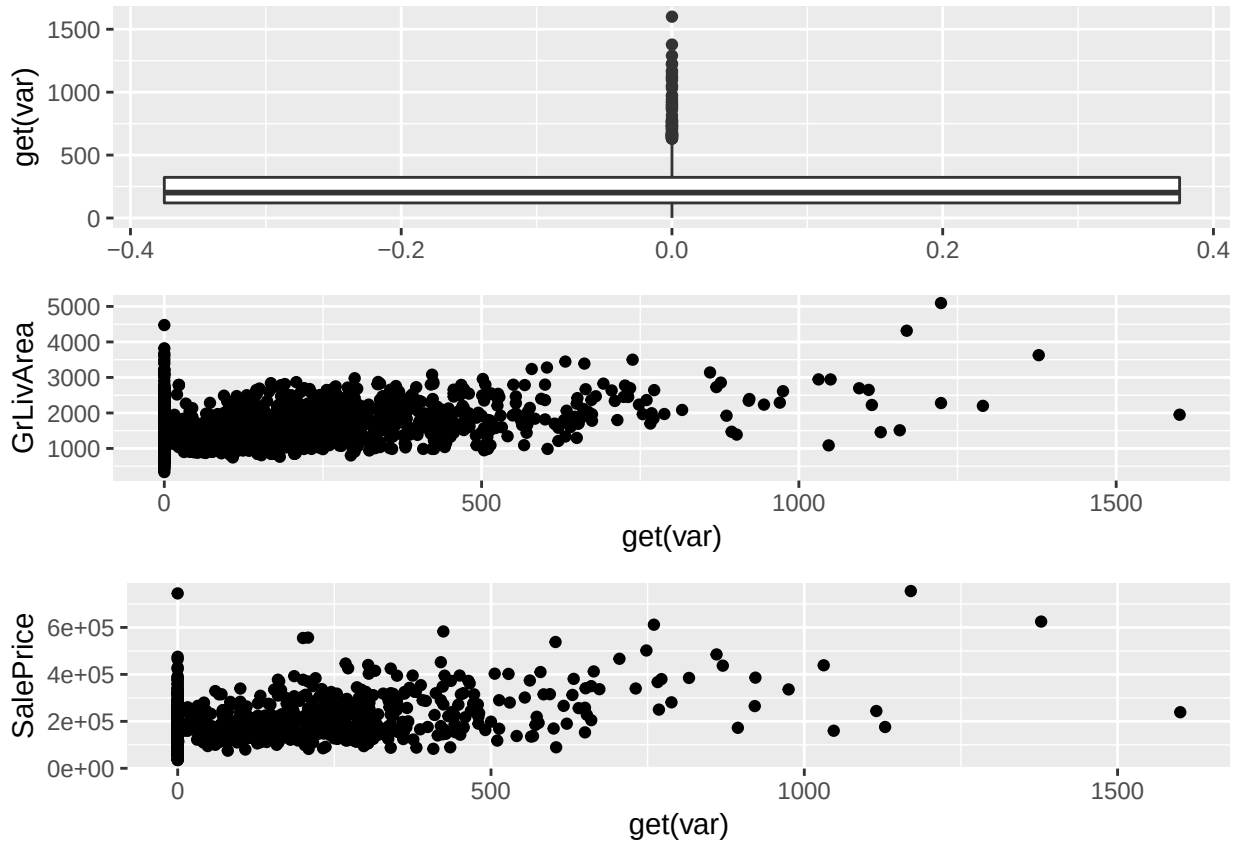
```
summary(dsDataAll$MasVnrArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0   100.9   163.0  1600.0
```

```
var <- "MasVnrArea"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$MasVnrArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0   120.0   202.0   254.7   323.2  1600.0
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=1)
```




```
# Parece que existe un valor extraño
# Actualizo a la mediana de todos los valores no cero
medianMasVnrArea <- median(a$MasVnrArea)

#select(data,Id,MasVnrArea) %>% filter(MasVnrArea>1500)

# Modificación directa
dsDataAll <- dsDataAll %>%
  rowwise() %>%
    mutate(MasVnrArea = ifelse(MasVnrArea>1500,medianMasVnrArea,MasVnrArea))


rm(medianMasVnrArea)
rm(var)
rm(a)
```

WoodDeckSF área de cubierta de madera en pies cuadrados

Parece que existe un valor extraño, sin embargo existe la posibilidad de que sea una casa completamente de madera, pero como el valor esta en el conjunto de test no se puede usar para entrenar y el modelo resultante no podrá calcular precios para casas solo de madera, por lo que actualizo el valor a la mediana según la superficie.

```
summary(dsDataAll$WoodDeckSF)
```

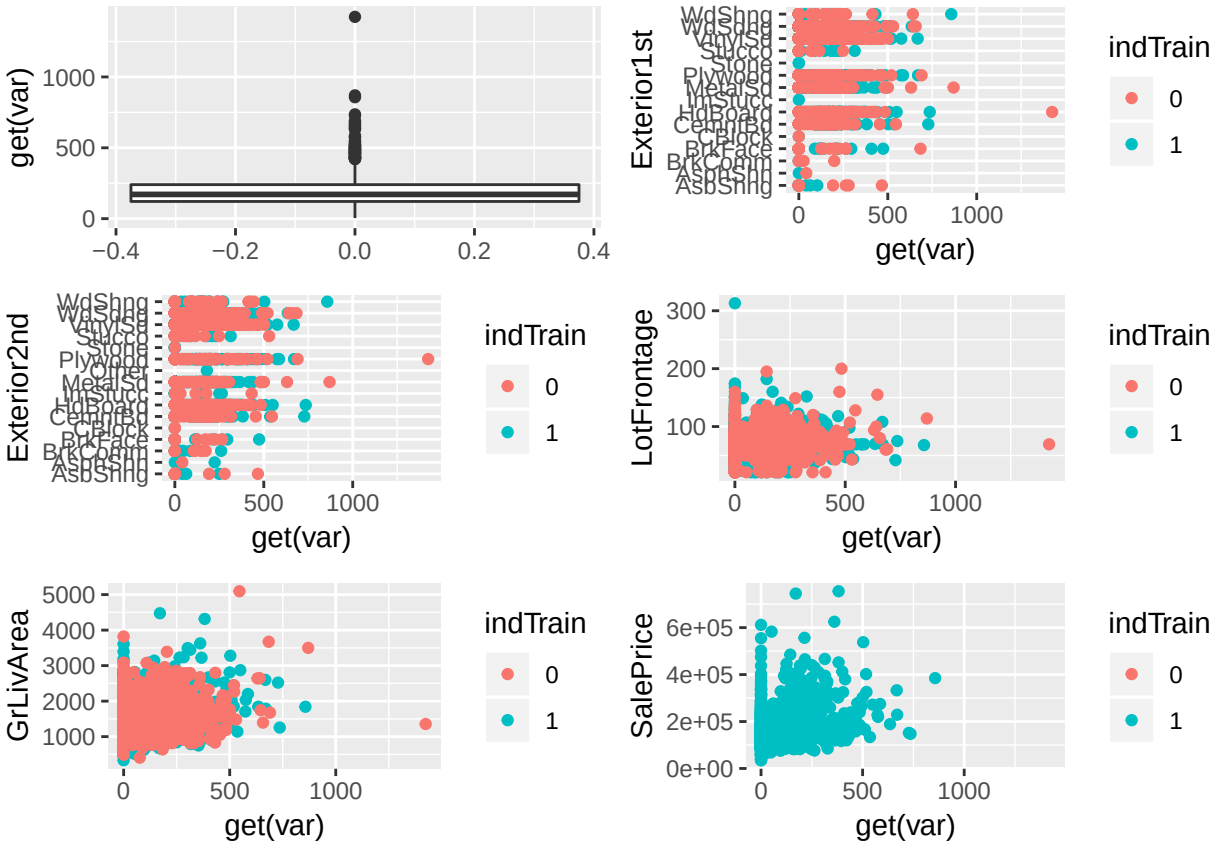
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##    0.00    0.00    0.00   93.63  168.00 1424.00
```

```
var <- "WoodDeckSF"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$WoodDeckSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     4.0   121.2   171.0   195.9   240.0  1424.0
```

```
# Comparo con otras variables
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=Exterior1st)) + geom_point(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=Exterior2nd)) + geom_point(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=LotFrontage)) + geom_point(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point(aes(color = indTrain)) #solo conjunto T
  ,ncol=2)
```



```
## Parece que existe un valor extraño
## Sin embargo existe la posibilidad de que sea una casa completamente de madera,
## pero como el valor esta en el conjunto de test no se puede usar para entrenar
## y el modelo resultante no podrá calcular precios para casas solo de madera,
## por lo que actualizo el valor a la mediana según la superficie

select(dsDataAll,Id,WoodDeckSF,GrLivArea) %>% filter(WoodDeckSF>1000)
```

```
## Source: local data frame [1 x 3]
## Groups: <by row>
##
## # A tibble: 1 x 3
##      Id WoodDeckSF GrLivArea
##   <int>      <int>     <int>
## 1  2607       1424      1356
```

```
a <- dsDataAll %>%
    filter(get(var)!=0 & GrLivArea > 1300 & GrLivArea < 1400) %>%
    select(var)

medianWoodDeckSF <- median(a$WoodDeckSF)

# Modificación directa
dsDataAll <- dsDataAll %>%
  rowwise() %>%
    mutate(WoodDeckSF = ifelse(WoodDeckSF>1500,medianWoodDeckSF,WoodDeckSF))

rm(medianWoodDeckSF)
rm(var)
rm(a)
```

BsmtFinSF1 SOTANO Tipo 1 terminado pies cuadrados

NO IDENTIFICO OUTLIERS

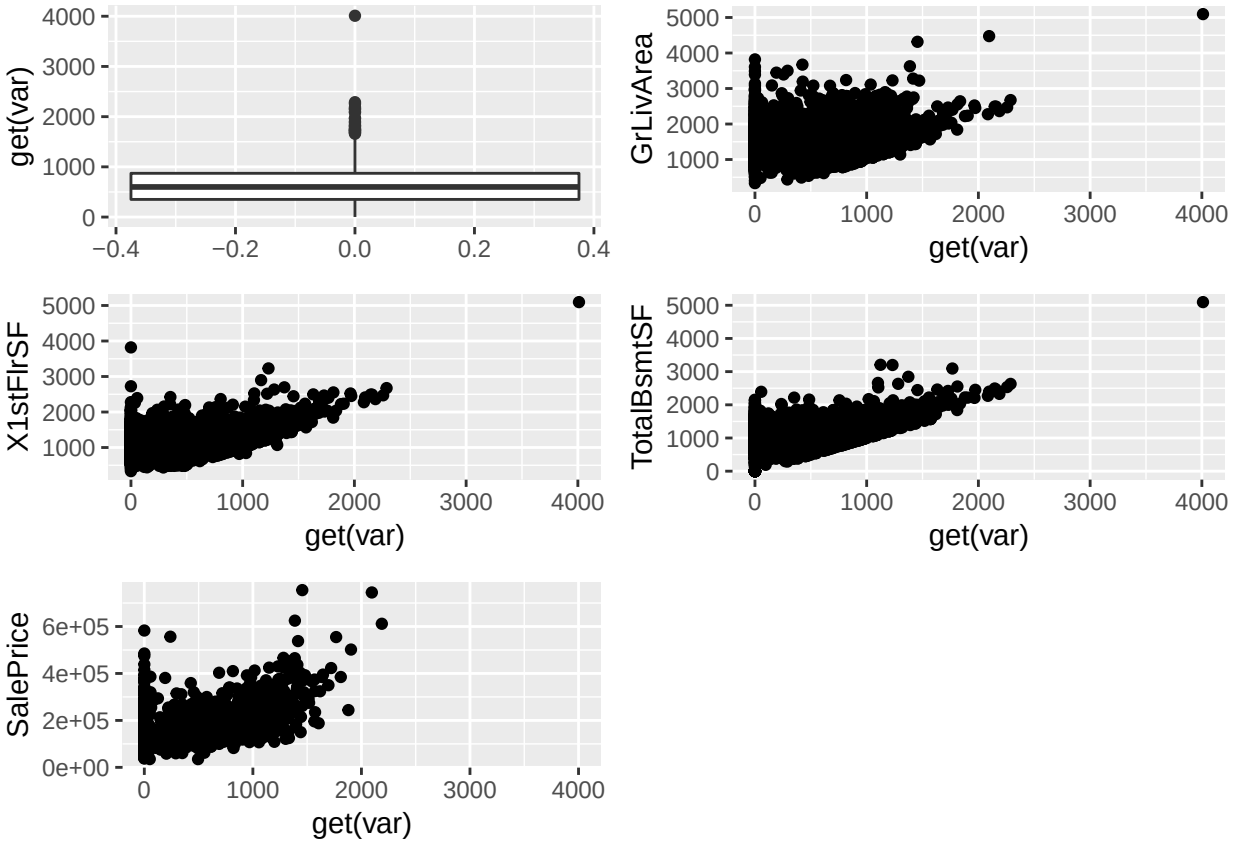
```
summary(dsDataAll$BsmtFinSF1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0   368.0   438.9   733.0  4010.0
```

```
var <- "BsmtFinSF1"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$BsmtFinSF1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     2.0   351.5   600.0   644.3   871.5  4010.0
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=X1stFlrSF)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=TotalBsmtSF)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=2)
```




```
# NO IDENTIFICO OUTLIERS

rm(var)
rm(a)
```

BsmtFinSF2 SOTANO pies cuadrados terminados tipo 2

NO IDENTIFICO OUTLIERS Aunque hay 3 valores muy altos, parecen estar dentro de la tendencia tanto para el precio como para la suma de superficies.

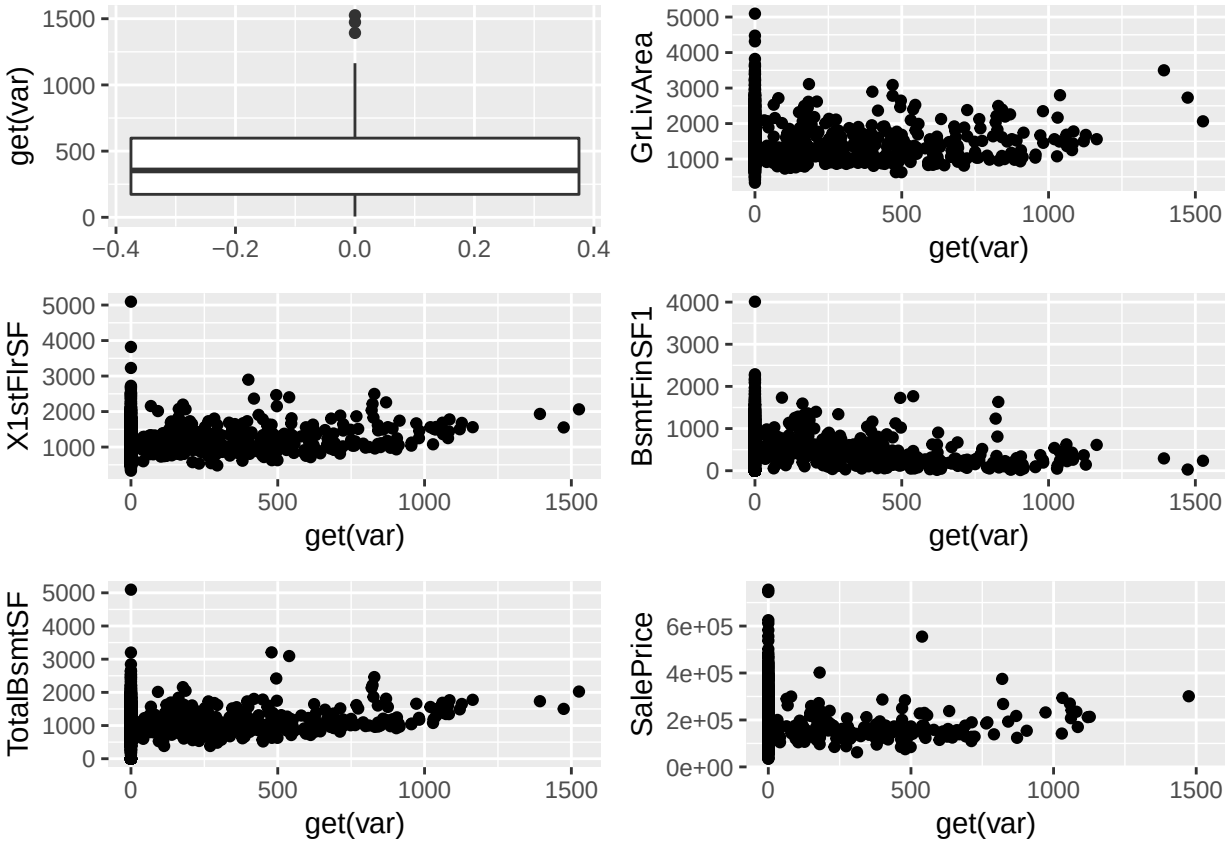
```
summary(dsDataAll$BsmtFinSF2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0    49.6     0.0  1526.0
```

```r
var <- "BsmtFinSF2"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$BsmtFinSF2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     6.0   173.5   354.0   416.9   598.0  1526.0
```

```r
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=X1stFlrSF)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=BsmtFinSF1)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=TotalBsmtSF)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=2)
```




```r
# NO IDENTIFICO OUTLIERS
# Aunque hay 3 valores muy altos, parecen estar dentro de la tendencia tanto para el precio como para l
```

```
rm(var)
rm(a)
```

BsmtUnfSF SOTANO pies cuadrados inacabados de área de sótano

NO IDENTIFICO OUTLIERS

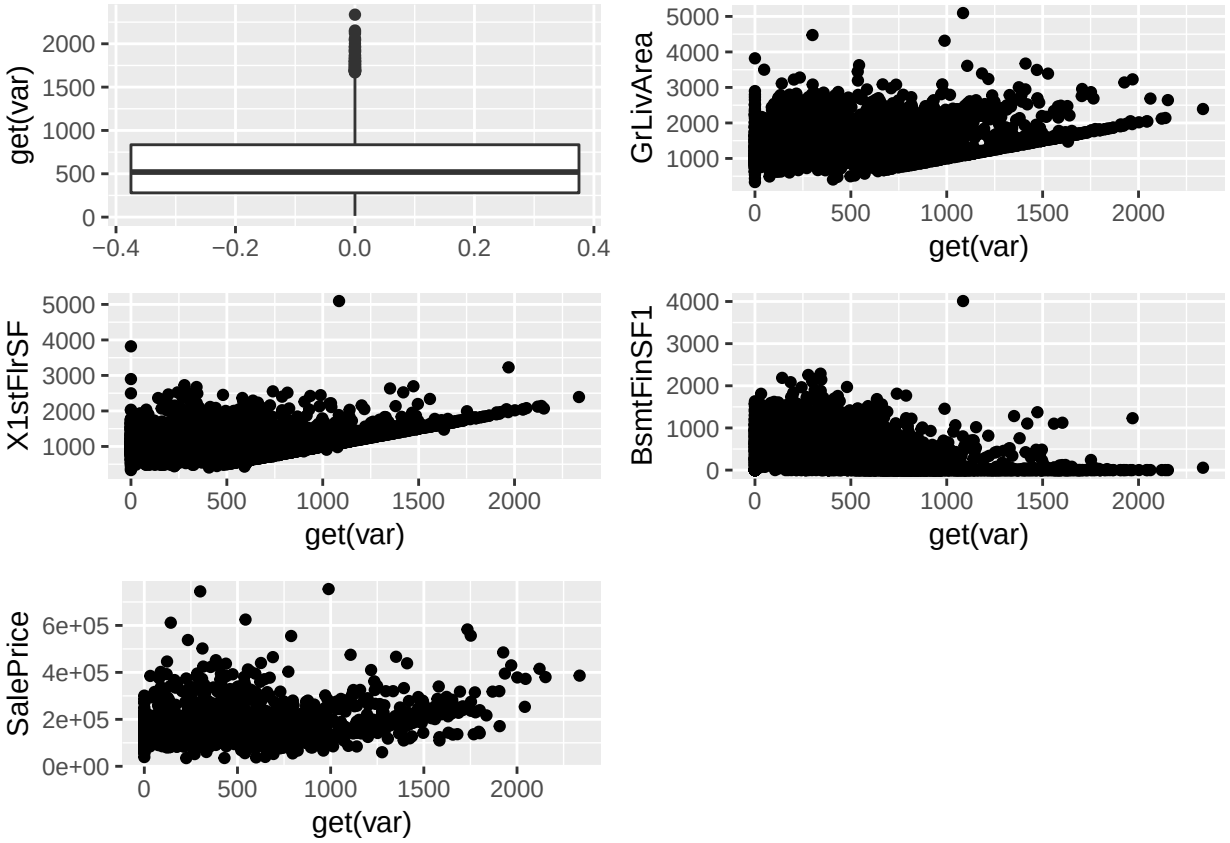
```
summary(dsDataAll$BsmtUnfSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   220.0   467.0   560.5   804.0  2336.0
```

```
var <- "BsmtUnfSF"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$BsmtUnfSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    14.0   280.0   520.0   611.2   835.0  2336.0
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=X1stFlrSF)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=BsmtFinSF1)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol = 2)
```




```
# NO IDENTIFICO OUTLIERS

rm(var)
rm(a)
```

TotalBsmtSF SOTANO pies cuadrados totales del área del sótano

NO IDENTIFICO OUTLIERS Existen 2 valores que pueden ser atípicos, pero son posibles y estan en el conjunto de entrenamiento, los mantengo

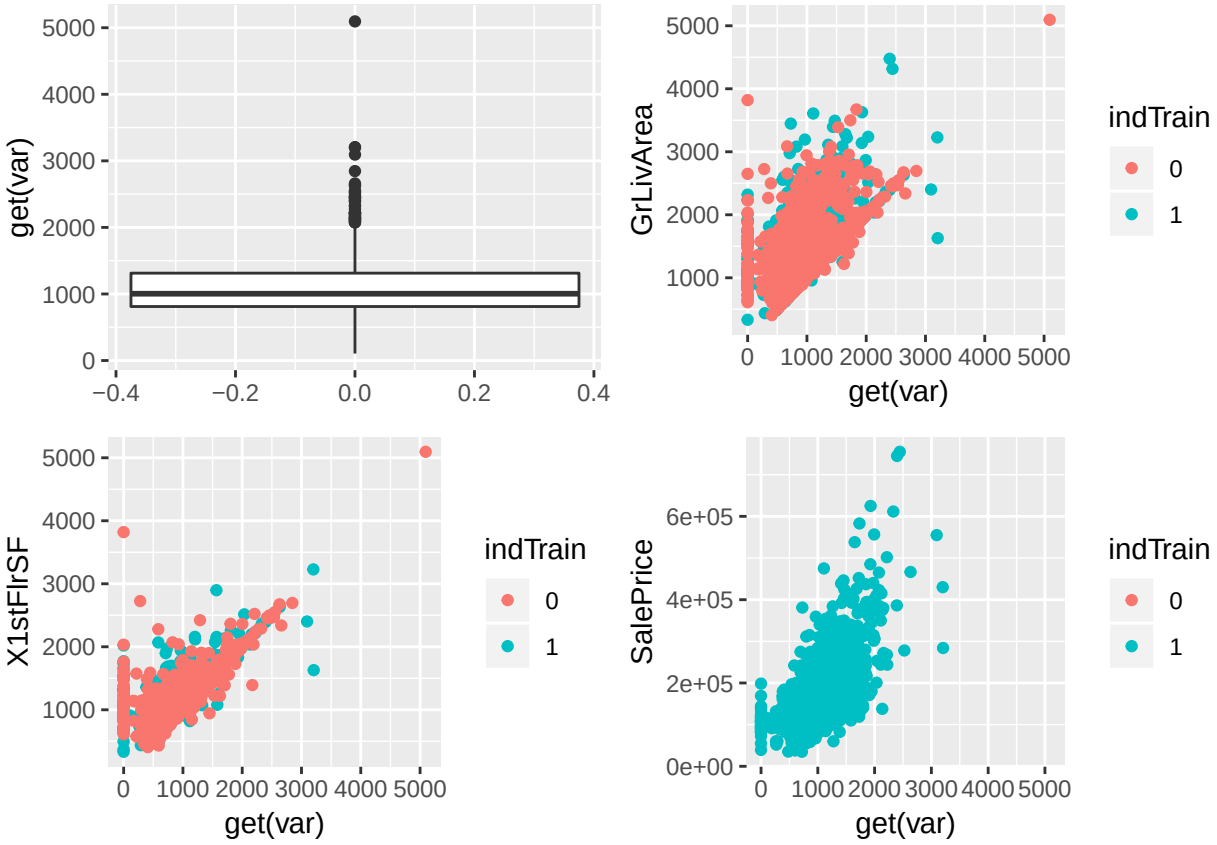
```
summary(dsDataAll$TotalBsmtSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     793     988    1049    1302    5095
```

```
var <- "TotalBsmtSF"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$TotalBsmtSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   105.0   811.2  1003.5  1078.2  1313.0  5095.0
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=X1stFlrSF)) + geom_point(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point(aes(color = indTrain)) #solo conjunto T
  ,ncol = 2)
```




```
# NO IDENTIFICO OUTLIERS
# Existen 2 valores que pueden ser atípicos, pero son posibles y estan en el conjunto de entrenamiento,

rm(var)
rm(a)
```

GarageArea tamaño del garaje en pies cuadrados

NO IDENTIFICO OUTLIERS

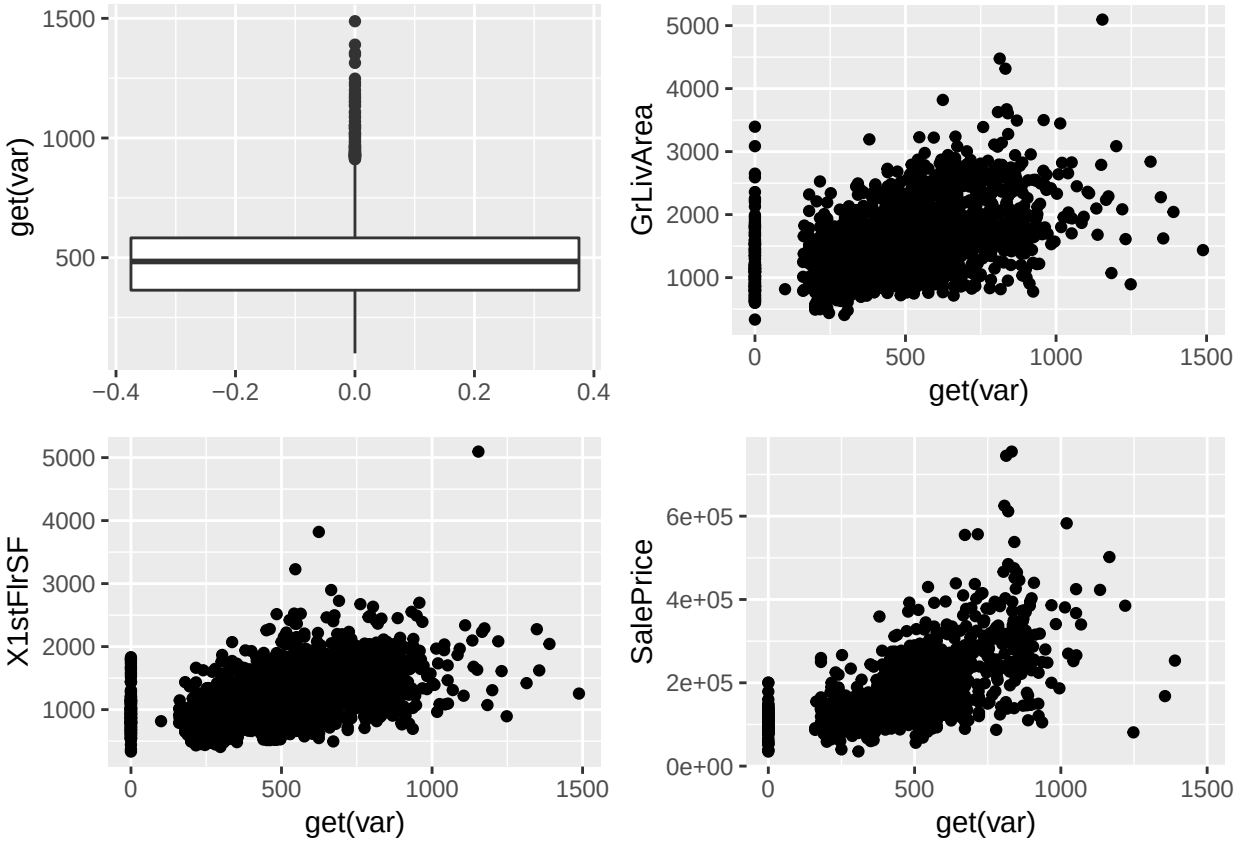
```
summary(dsDataAll$GarageArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   320.0   480.0   472.2   576.0  1488.0
```

```
var <- "GarageArea"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$GarageArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   100.0   364.0   484.0   499.3   582.5  1488.0
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=X1stFlrSF)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=2)
```




```
# NO IDENTIFICO OUTLIERS

rm(var)
rm(a)
```

OpenPorchSF área de porche abierto en pies cuadrados

Parece que existen un par de valores extraños: *Uno en el conjunto de entrenamiento, con un porche muy grande y un precio bajo* Otro en el conjunto de test, con una superficie muy grande Asigno mediana segun el area

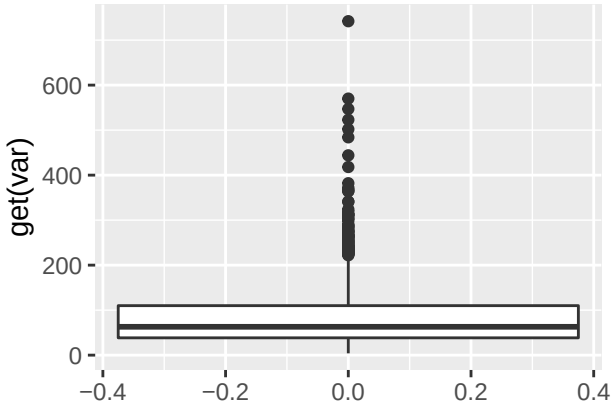
```
summary(dsDataAll$OpenPorchSF)
```

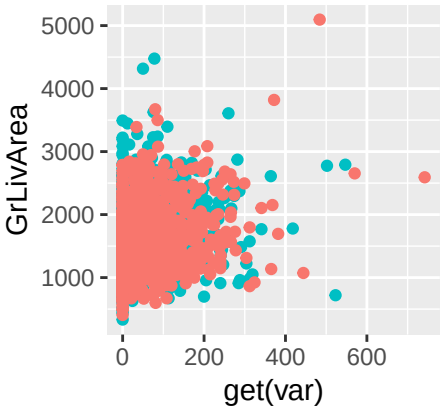
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00   26.00   47.28   70.00  742.00
```

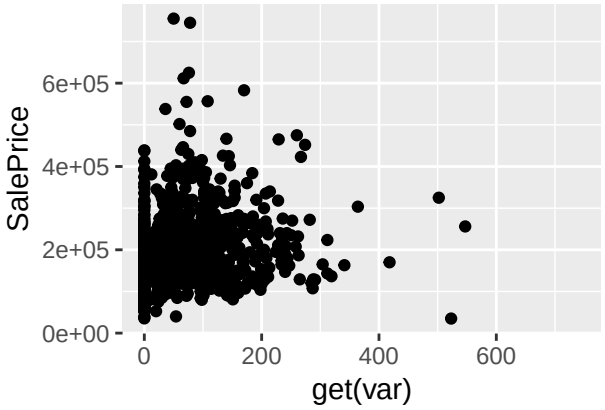
```
var <- "OpenPorchSF"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$OpenPorchSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4.00   38.50   63.00   85.19  110.00  742.00
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol = 2)
```








```
## Parece que existen un par de valores extraños
## Uno en el conjunto de entrenamiento, con un porche muy grande y un precio bajo
## Otro en el conjunto de test, con una superficie muy grande

## Asigno mediana segun el area
select(dsDataAll,Id,OpenPorchSF,GrLivArea) %>% filter(OpenPorchSF>600|(OpenPorchSF>500&GrLivArea<1000))
```

```
## Source: local data frame [2 x 3]
## Groups: <by row>
##
## # A tibble: 2 x 3
##      Id OpenPorchSF GrLivArea
##   <int>       <int>     <int>
## 1   496         523       720
## 2  2558         742      2592
```

```
a <- dsDataAll %>%
    filter(get(var)!=0 & GrLivArea > 700 & GrLivArea < 750) %>%
    select(var)

medianOpenPorchSF <- median(a$OpenPorchSF)

# Modificación directa
dsDataAll <- dsDataAll %>%
  rowwise() %>%
```

```
    mutate(OpenPorchSF = ifelse(OpenPorchSF>500&GrLivArea<1000,medianOpenPorchSF,OpenPorchSF))


a <- dsDataAll %>%
    filter(get(var)!=0 & GrLivArea > 2550 & GrLivArea < 2650) %>%
    select(var)

medianOpenPorchSF <- median(a$OpenPorchSF)

# Modificación directa
dsDataAll <- dsDataAll %>%
  rowwise() %>%
    mutate(OpenPorchSF = ifelse(OpenPorchSF>600,medianOpenPorchSF,OpenPorchSF))

## Verifico
dsDataAll %>%
  filter(OpenPorchSF!=OpenPorchSF) %>%
  select(OpenPorchSF, OpenPorchSF)
```

```
## Source: local data frame [0 x 1]
## Groups: <by row>
##
## # A tibble: 0 x 1
## # ... with 1 variable: OpenPorchSF <dbl>
```

```
rm(medianOpenPorchSF)
rm(var)
rm(a)
```

EnclosedPorch área de porche cerrado en pies cuadrados

Parece que existen un valor extraño en el conjunto de test, con una superficie muy grande

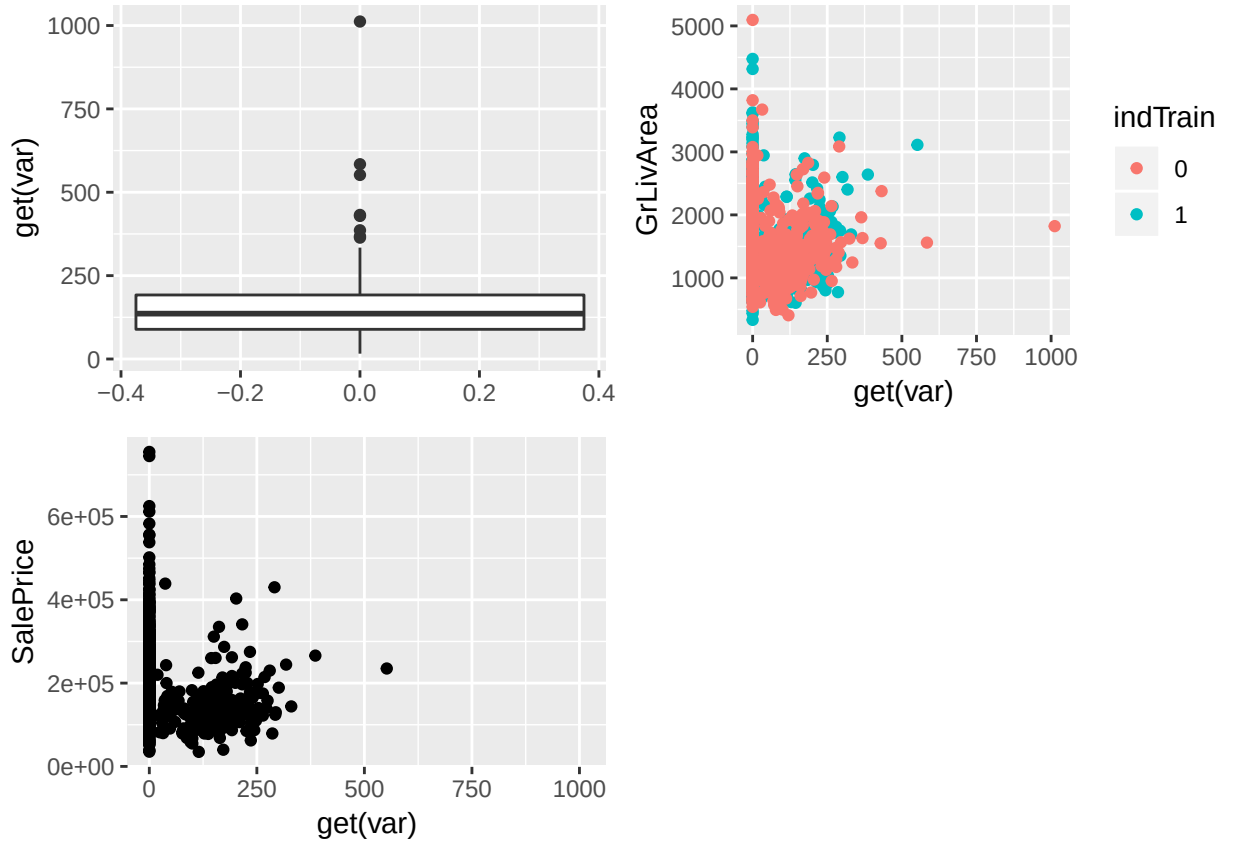
```
summary(dsDataAll$EnclosedPorch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   23.11    0.00 1012.00
```

```
var <- "EnclosedPorch"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$EnclosedPorch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    16.0    89.0   136.0   146.9   192.0  1012.0
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=2)
```




```
## Parece que existen un valor extraño en el conjunto de test, con una superficie muy grande

## Asigno mediana segun el area
select(dsDataAll,Id,EnclosedPorch,GrLivArea) %>% filter(EnclosedPorch>600)
```

```
## Source: local data frame [1 x 3]
## Groups: <by row>
##
## # A tibble: 1 x 3
##      Id EnclosedPorch GrLivArea
##   <int>         <int>     <int>
## 1  2504          1012      1822
```

```
a <- dsDataAll %>%
    filter(get(var)!=0 & GrLivArea > 1800 & GrLivArea < 1850) %>%
    select(var)
```

```
medianEnclosedPorch <- median(a$EnclosedPorch)

# Modificación directa
dsDataAll <- dsDataAll %>%
  rowwise() %>%
    mutate(EnclosedPorch = ifelse(OpenPorchSF>600,medianEnclosedPorch,EnclosedPorch))


rm(medianEnclosedPorch)
rm(var)
rm(a)
```

X3SsnPorch área de porche de tres estaciones en pies cuadrados

NO IDENTIFICO OUTLIERS

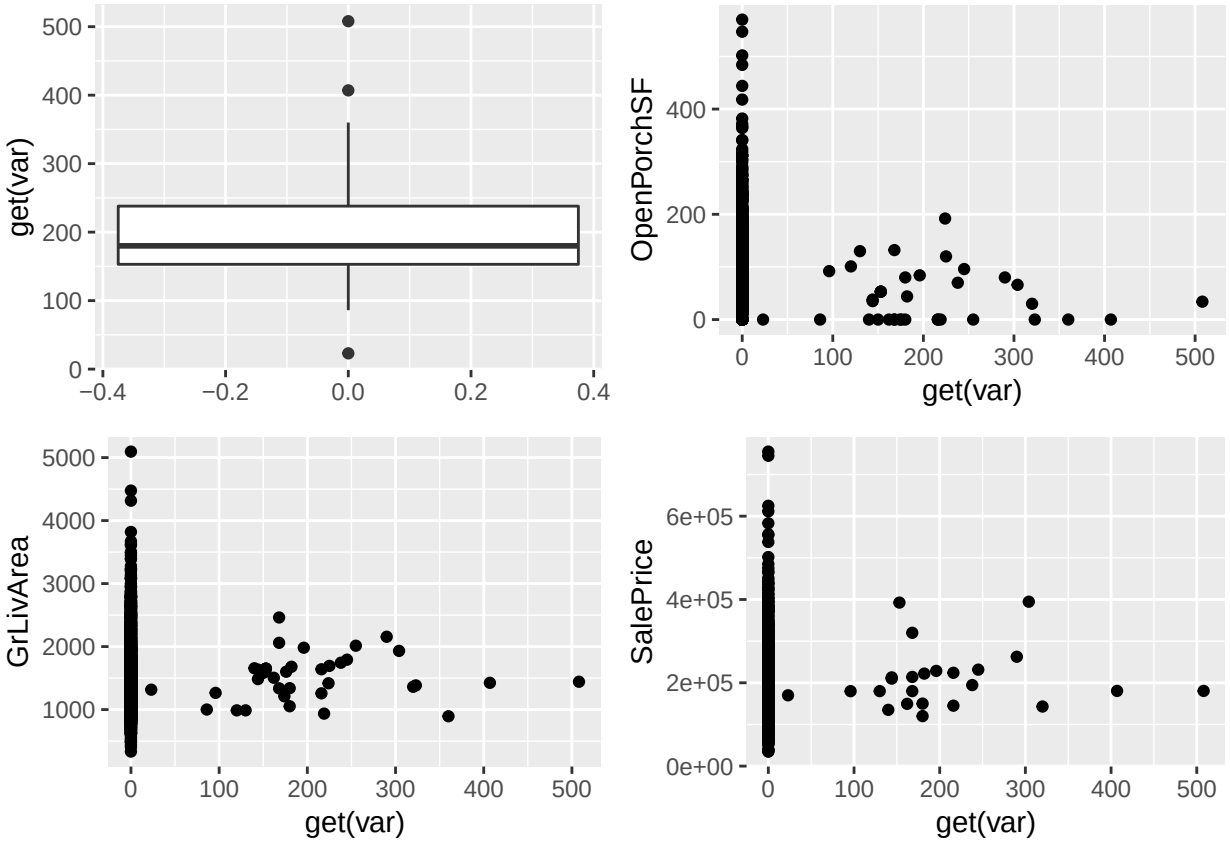
```
summary(dsDataAll$X3SsnPorch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   2.604   0.000 508.000
```

```
var <- "X3SsnPorch"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$X3SsnPorch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    23.0   153.0   180.0   205.3   238.0   508.0
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v

plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=OpenPorchSF)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=2)
```




```
# NO IDENTIFICO OUTLIERS

rm(var)
rm(a)
```

ScreenPorch área del porche de la pantalla en pies cuadrados

NO IDENTIFICO OUTLIERS

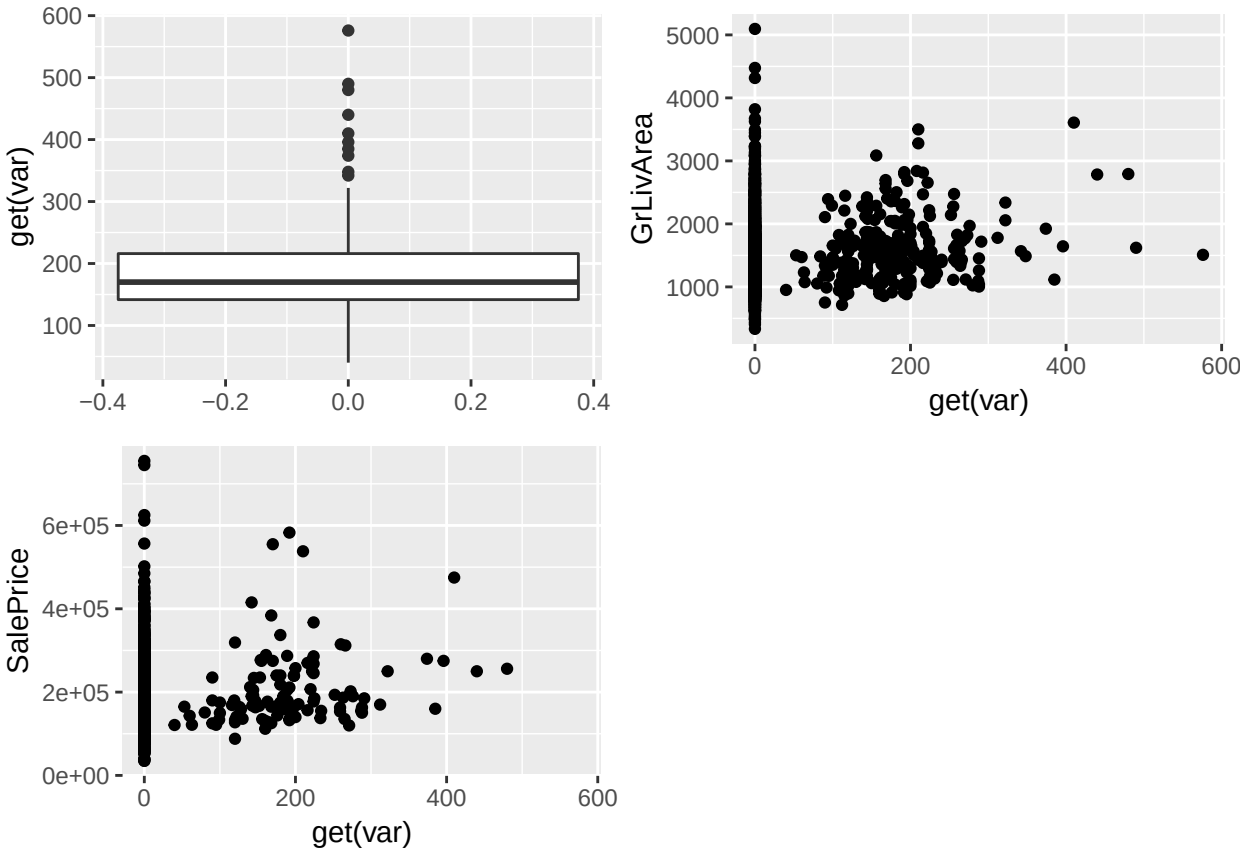
```
summary(dsDataAll$ScreenPorch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   16.07    0.00  576.00
```

```
var <- "ScreenPorch"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$ScreenPorch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    40.0   141.8   170.0   183.1   216.0   576.0
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=2)
```




```
# NO IDENTIFICO OUTLIERS

rm(var)
rm(a)
```

PoolArea área de la piscina en pies cuadrados

NO IDENTIFICO OUTLIERS

```
summary(dsDataAll$PoolArea)
```

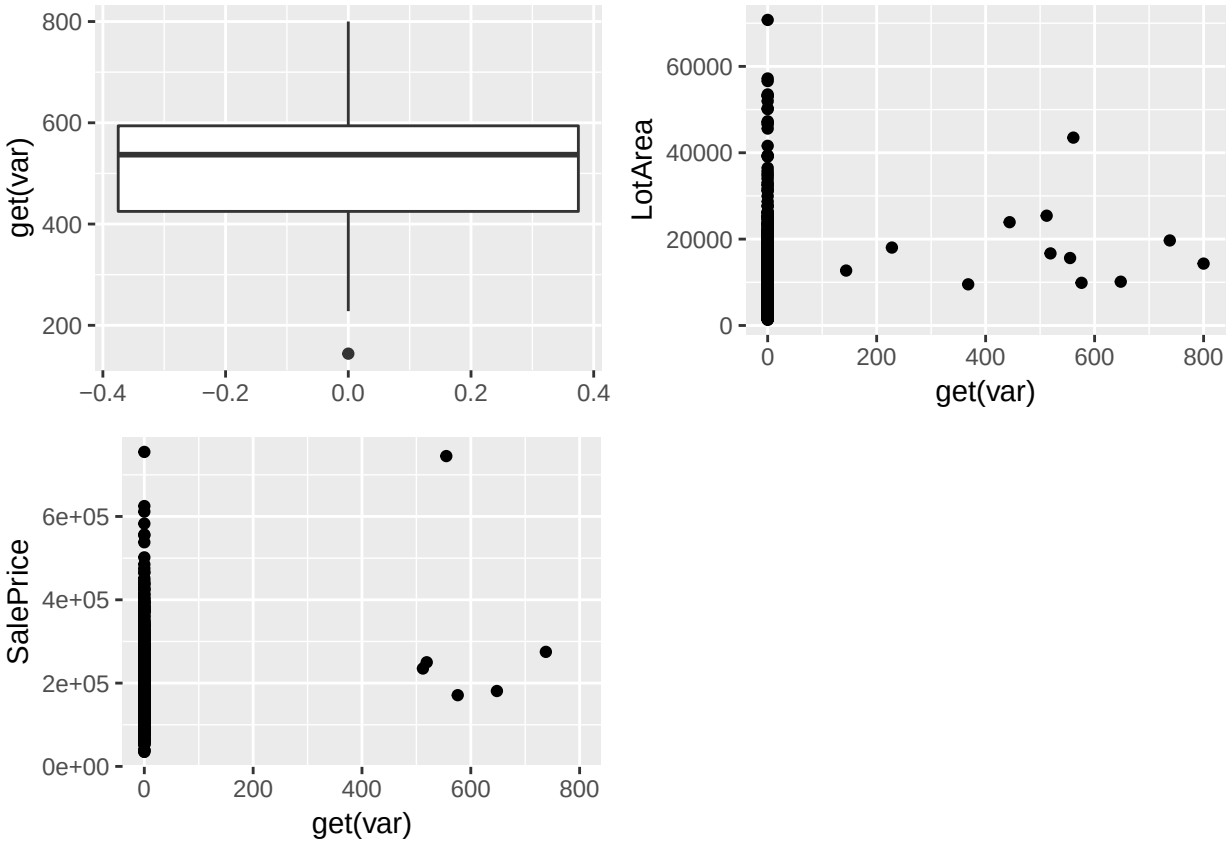
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   2.089   0.000 800.000
```

```
var <- "PoolArea"
```

```r
a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$PoolArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   144.0   425.0   537.0   507.8   594.0   800.0
```

```r
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=LotArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol = 2)
```




```r
# NO IDENTIFICO OUTLIERS

rm(var)
rm(a)
```

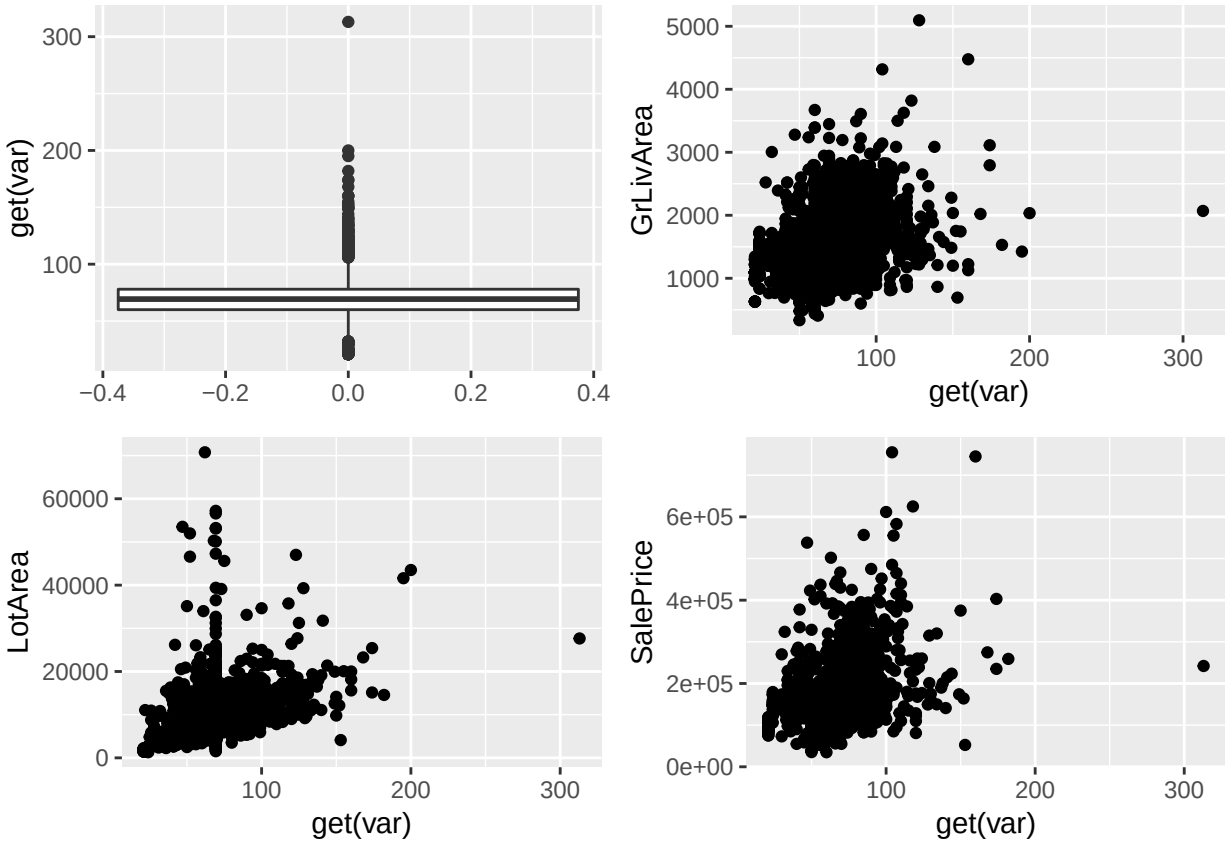
LotFrontage pies lineales de calle conectados a la propiedad

NO IDENTIFICO OUTLIERS

```r
summary(dsDataAll$LotFrontage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   21.00   60.00   69.31   69.20   78.00  313.00
```

```r
var <- "LotFrontage"

a <- dsDataAll %>%
    select(var)

# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=LotArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=2)
```




```r
# NO IDENTIFICO OUTLIERS

rm(var)
rm(a)
```

MiscVal Valor de la característica miscelánea

Aunque se identifican valores extremos, esta variable puede contener cualquier valor ya que como su nombre indica es un cajon desastre. Por lo que no realizo ninguna acción.

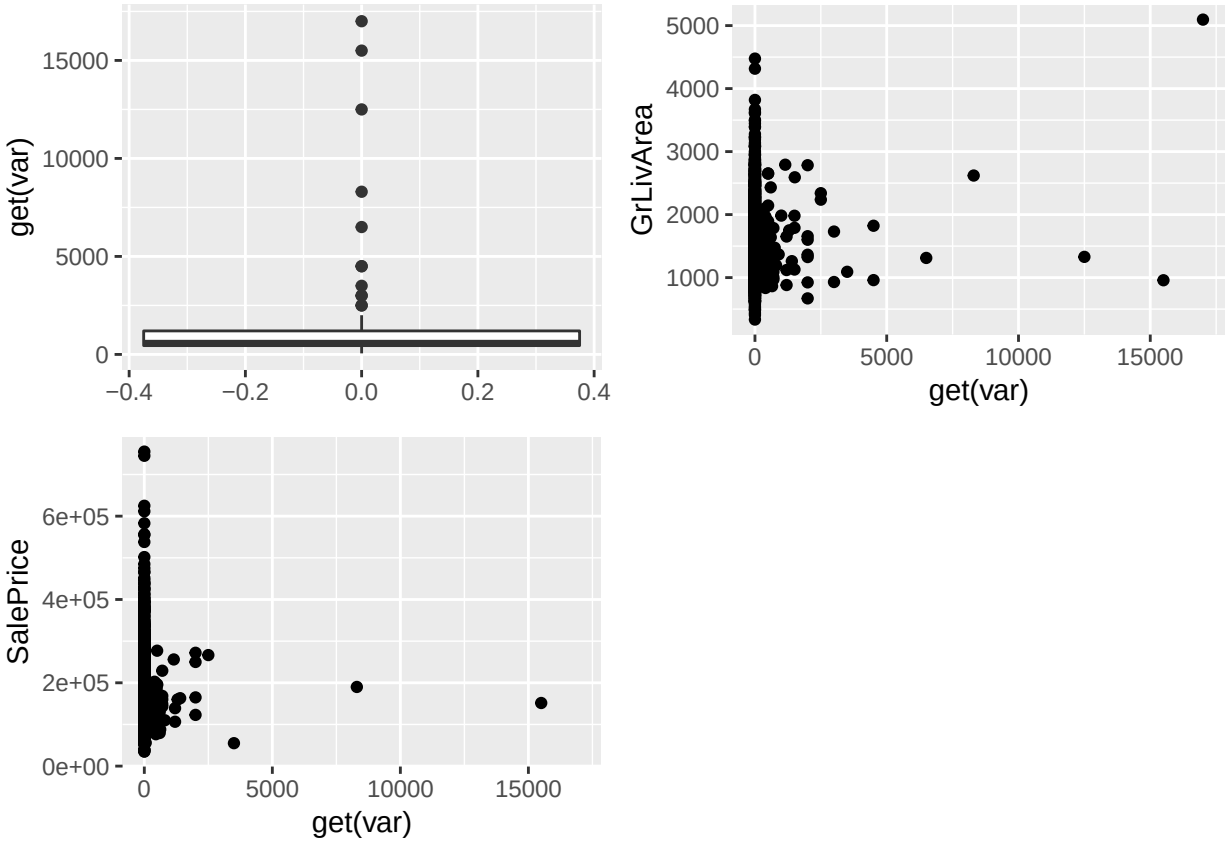
```
summary(dsDataAll$MiscVal)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      0.00      0.00      0.00     50.86      0.00 17000.00
```

```
var <- "MiscVal"

a <- dsDataAll %>%
    select(var) %>%
    filter(get(var)!=0)

summary(a$MiscVal)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      54     450     600    1440    1200   17000
```

```
# Comparo con otras variables
# la clase de construcción / tipo de vivienda / estilo de vivienda / superficie habitable / precio de v
plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(dsDataAll, aes(x=get(var), y=GrLivArea)) + geom_point()
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=2)
```



```
# Aunque se identifican valores extremos, esta variable puede contener cualquier valor ya que como su n

rm(var)
rm(a)
```

Resto de variables discretas

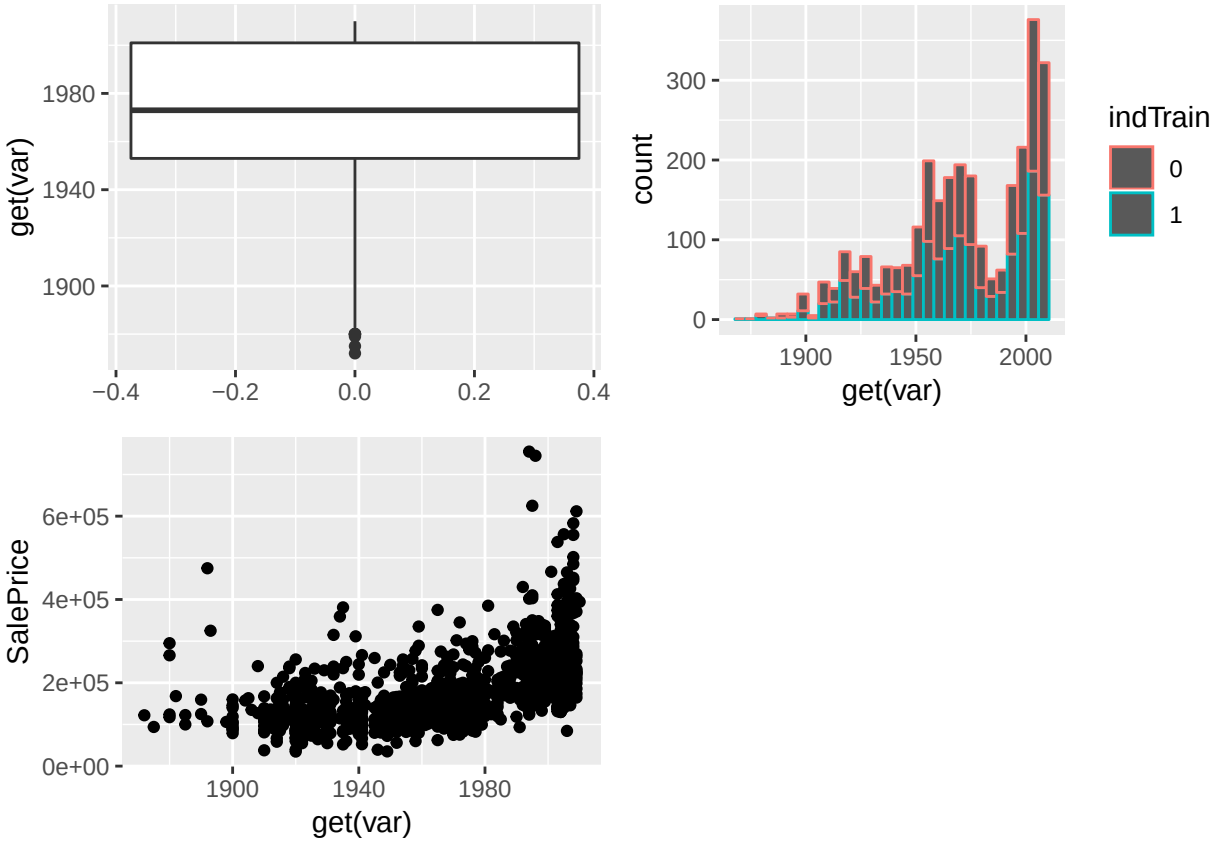
YearBuilt Año de construcción original

NO IDENTIFICO OUTLIERS, Aunque hay fechas antiguas, parece que los datos son consistentes.

```
summary(dsDataAll$YearBuilt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1872    1953    1973    1971    2001    2010
```

```
var <- "YearBuilt"

a <- dsDataAll %>%
    select(var)

plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(data=dsDataAll, aes(x=get(var))) + geom_histogram(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=2)
```



```
# NO IDENTIFICO OUTLIERS
# Aunque hay fechas antiguas, parece que los datos son consistentes.

rm(var)
rm(a)
```

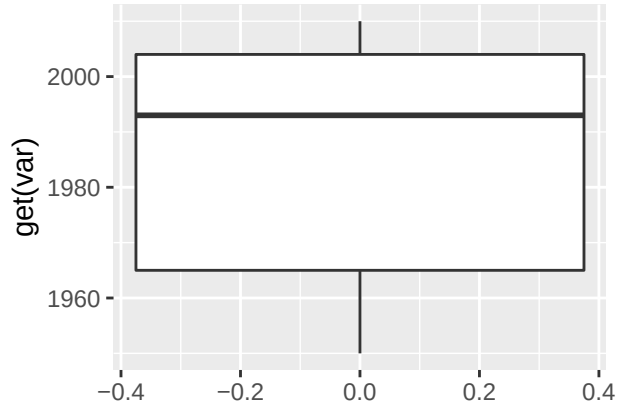
YearRemodAdd Año de remodelación

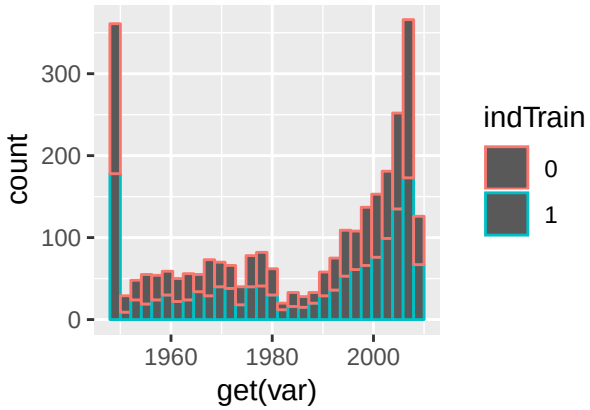
Las casas que se construyeron antes de 1950 se les puso una fecha de remodelación 1950, modifico la fecha de remodelación para casas anteriores a 1950 asignandoles la fecha de construcción

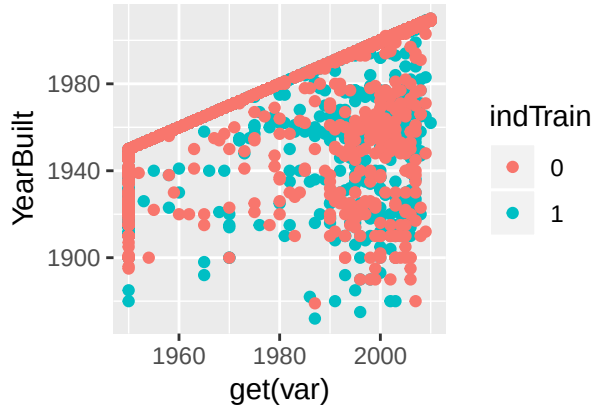
```
summary(dsDataAll$YearRemodAdd)
```

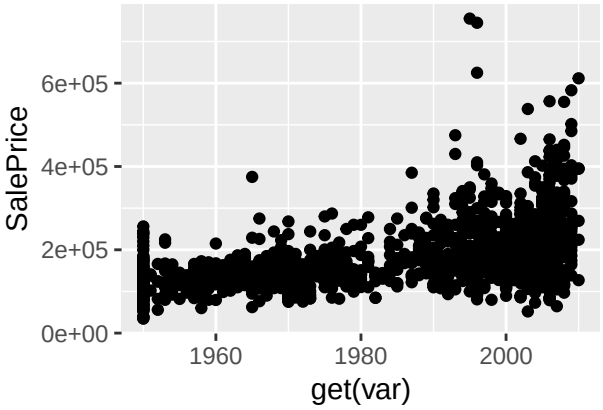
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1950    1965    1993    1984    2004    2010
```

```
var <- "YearRemodAdd"

a <- dsDataAll %>%
    select(var)

plot_grid(
  ggplot(a, aes(y = get(var))) + geom_boxplot()
  ,ggplot(data=dsDataAll, aes(x=get(var))) + geom_histogram(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=YearBuilt)) + geom_point(aes(color = indTrain))
  ,ggplot(dsDataAll, aes(x=get(var), y=SalePrice)) + geom_point() #solo conjunto TRAIN al estar SalePri
  ,ncol=2)
```










```
# Parece que a las casas que se construyeron antes de 1950 se les puso una fecha de remodelación 1950

# Verifico si muchas casas tienen la misma fecha de construcción que de remoelación
dsDataAll %>%
  filter(YearBuilt==YearRemodAdd) %>%
  group_by(indTrain) %>%
  summarise(n = n())
```

```
## # A tibble: 2 x 2
##   indTrain     n
##   <fct>    <int>
## 1 0          796
## 2 1          763
```

```
# Modifico la fecha de remodelación para casas anteriores a 1950 asignandoles la fecha de construcción
dsDataAll <- dsDataAll %>%
  mutate(YearRemodAdd = ifelse(YearBuilt<1950 & YearRemodAdd==1950,YearBuilt,YearRemodAdd))

# verifico modificación
dsDataAll %>% filter(YearRemodAdd != YearRemodAdd) %>%
    select(Id, YearBuilt, YearRemodAdd, YearRemodAdd)
```

```
## Source: local data frame [0 x 3]
## Groups: <by row>
##
```

```
## # A tibble: 0 x 3
## # ... with 3 variables: Id <int>, YearBuilt <int>, YearRemodAdd <int>
```

```
rm(var)
rm(a)
```

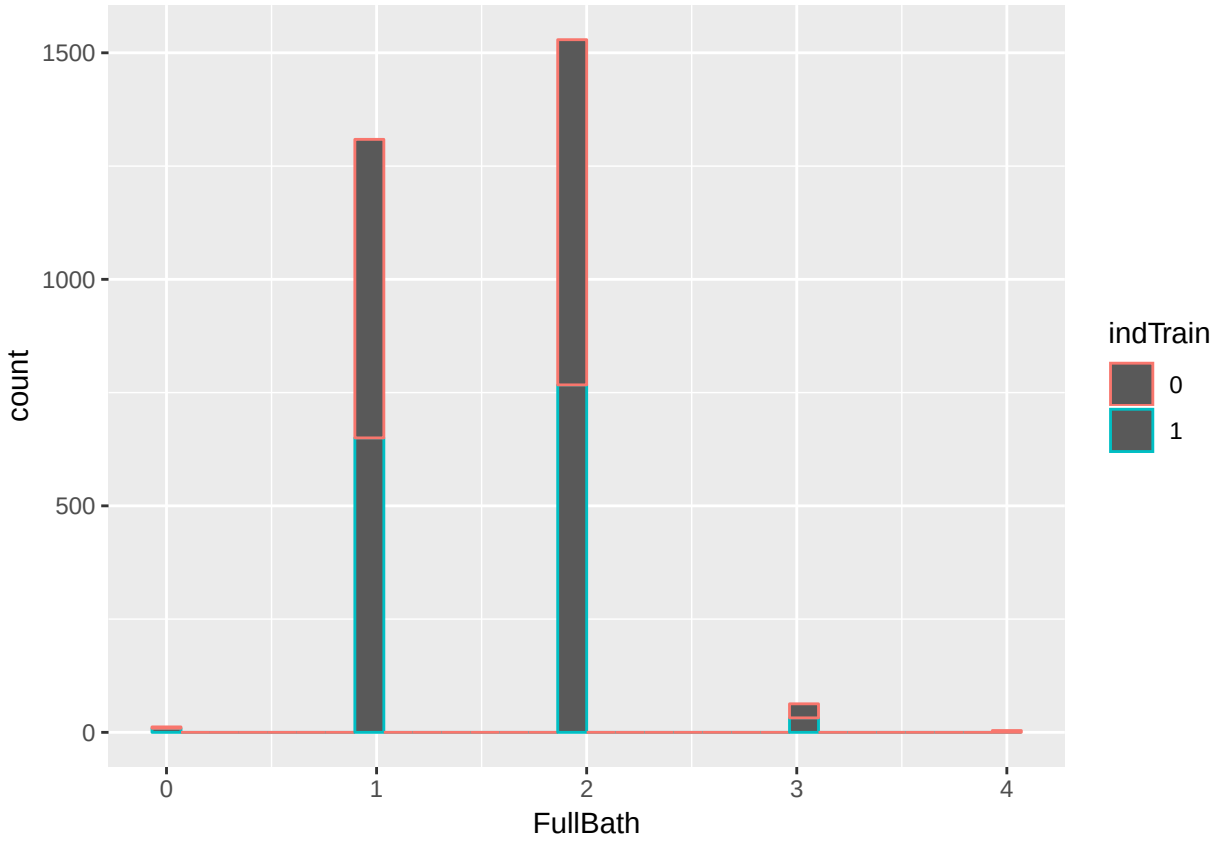
FullBath baños completos por encima del grado

NO IDENTIFICO OUTLIERS

```
summary(dsDataAll$FullBath)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   2.000   1.567   2.000   4.000
```

```
ggplot(data=dsDataAll, aes(x=FullBath)) + geom_histogram(aes(color = indTrain))
```




```
#NO IDENTIFICO OUTLIERS
```

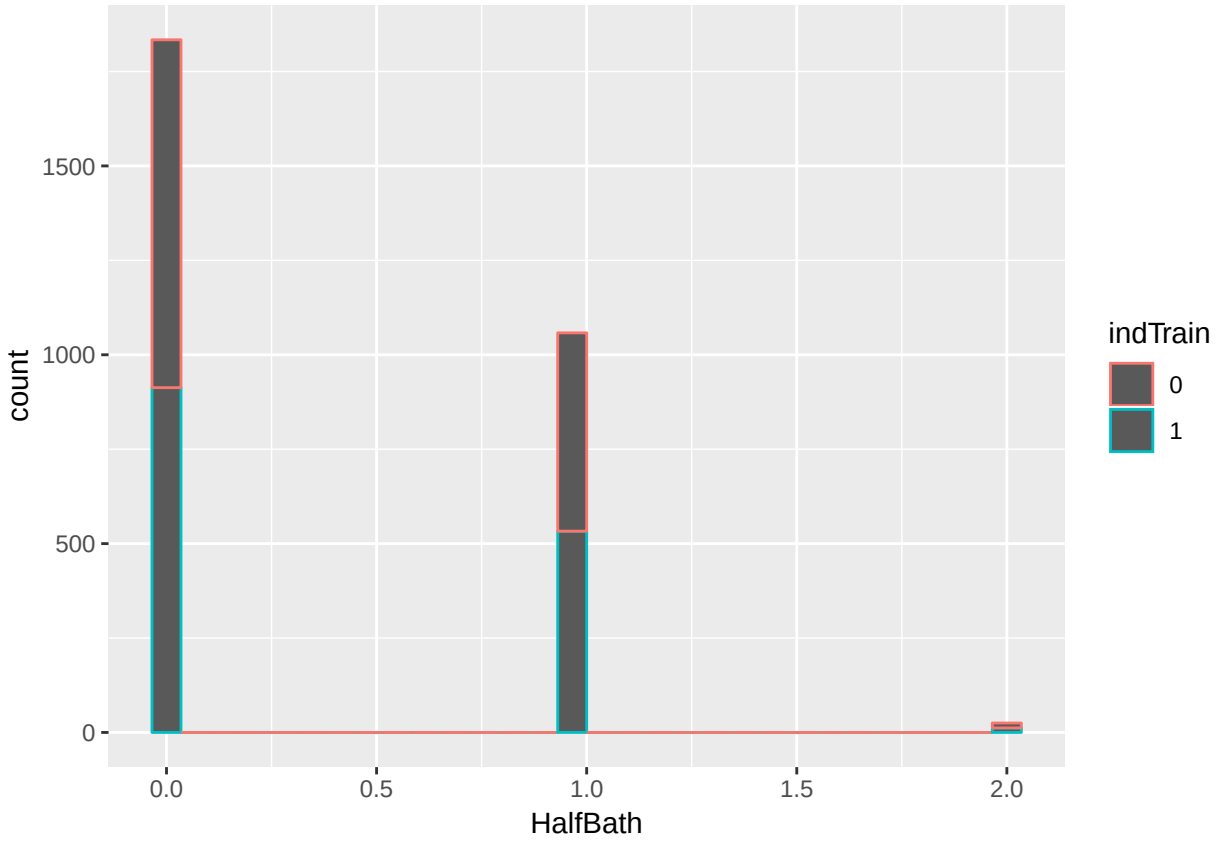
HalfBath La mitad de los baños por encima de grado

NO IDENTIFICO OUTLIERS

```r
summary(dsDataAll$HalfBath)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3798  1.0000  2.0000
```

```r
ggplot(data=dsDataAll, aes(x=HalfBath)) + geom_histogram(aes(color = indTrain))
```




```r
#NO IDENTIFICO OUTLIERS
```

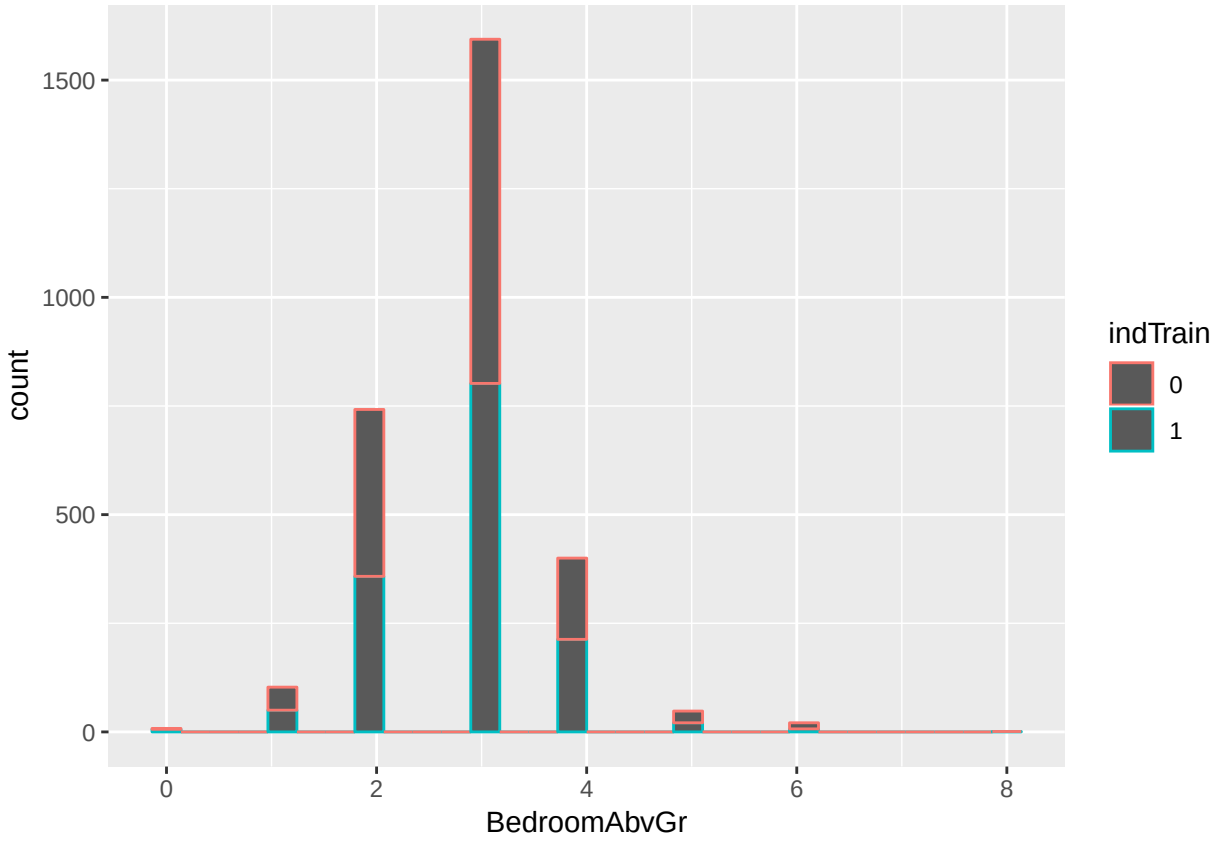
BedroomAbvGr número de dormitorios por encima del nivel del sótano

NO IDENTIFICO OUTLIERS

```r
summary(dsDataAll$BedroomAbvGr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    2.00    3.00    2.86    3.00    8.00
```

```r
ggplot(data=dsDataAll, aes(x=BedroomAbvGr)) + geom_histogram(aes(color = indTrain))
```




```
#NO IDENTIFICO OUTLIERS
```

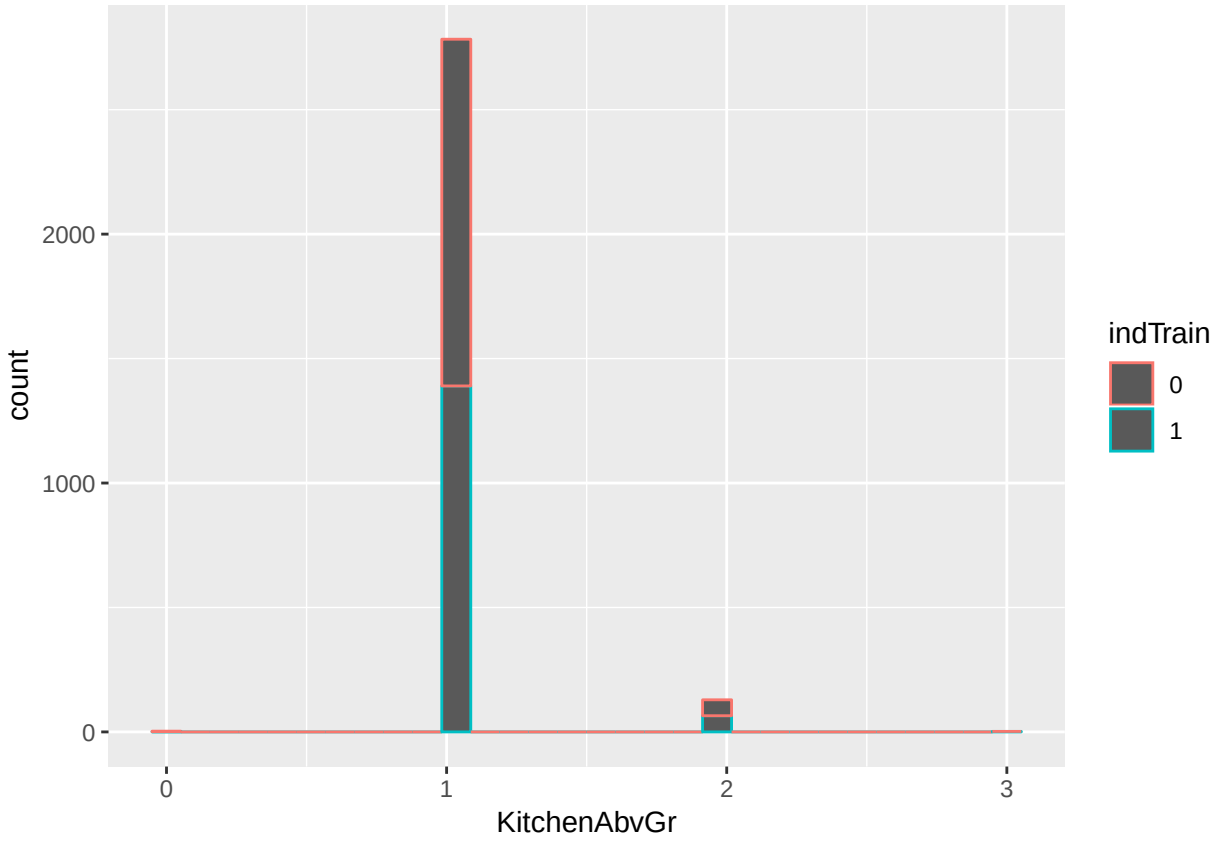
KitchenAbvGr número de cocinas

NO IDENTIFICO OUTLIERS

```
summary(dsDataAll$KitchenAbvGr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   1.000   1.045   1.000   3.000
```

```
ggplot(data=dsDataAll, aes(x=KitchenAbvGr)) + geom_histogram(aes(color = indTrain))
```




```
#NO IDENTIFICO OUTLIERS
```

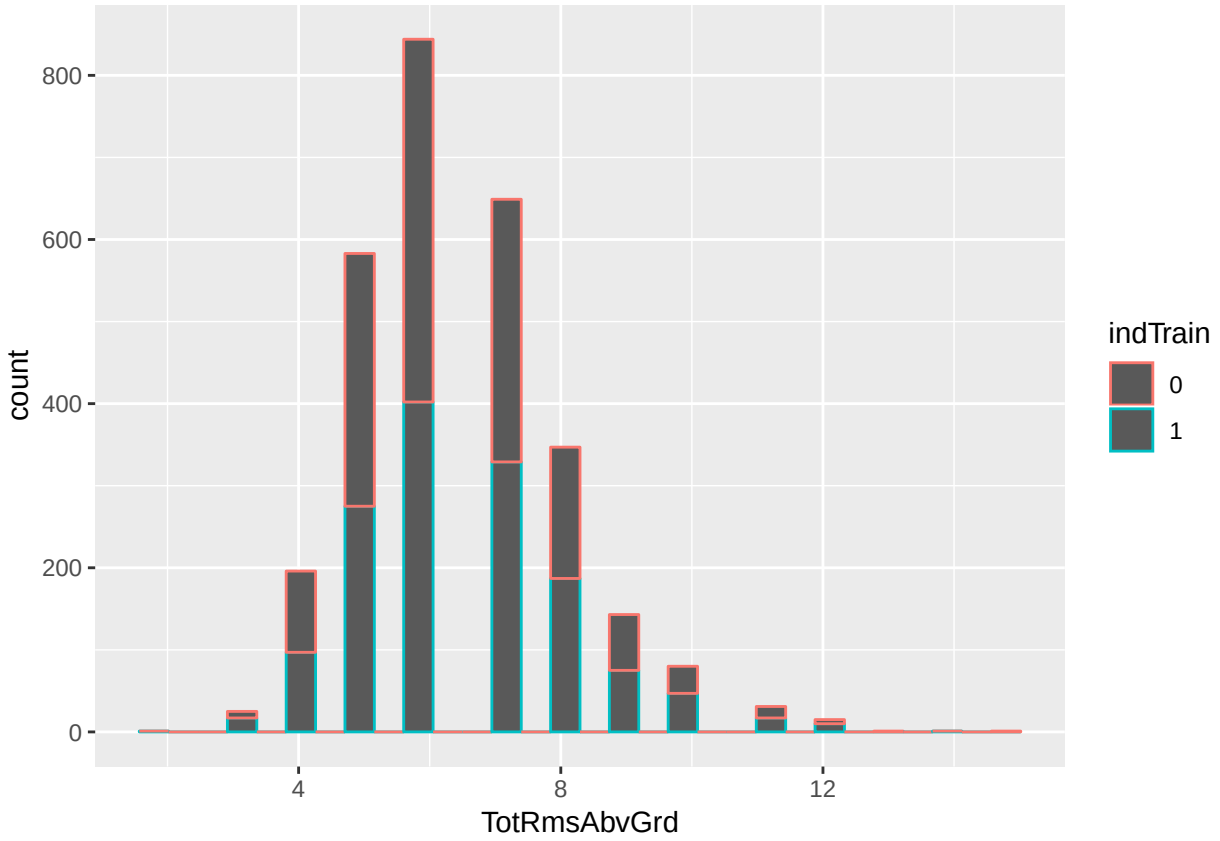
TotRmsAbvGrd total de habitaciones por encima del grado (no incluye baños)

NO IDENTIFICO OUTLIERS

```
summary(dsDataAll$TotRmsAbvGrd)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   5.000   6.000   6.448   7.000  15.000
```

```
ggplot(data=dsDataAll, aes(x=TotRmsAbvGrd)) + geom_histogram(aes(color = indTrain))
```




```
#NO IDENTIFICO OUTLIERS
```

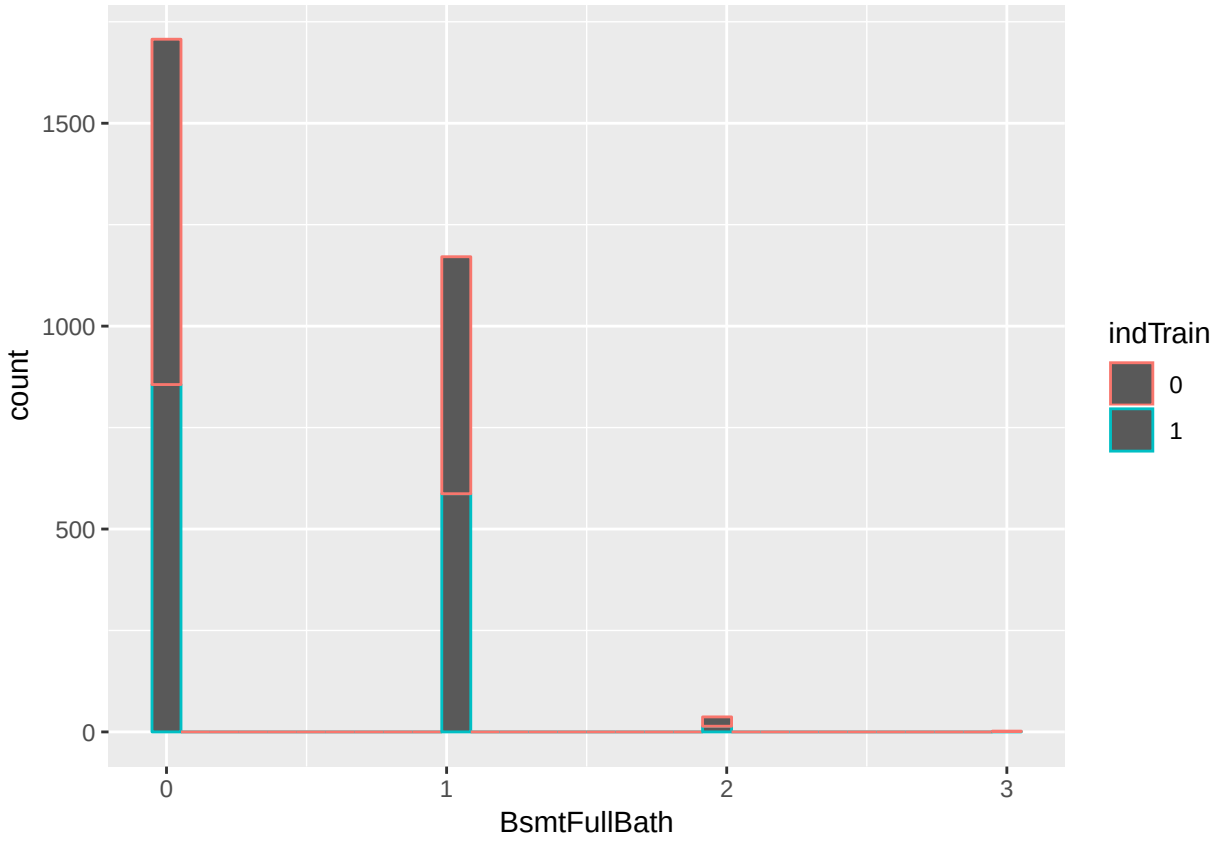
BsmtFullBath baños completos en el sótano

NO IDENTIFICO OUTLIERS

```
summary(dsDataAll$BsmtFullBath)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.4289  1.0000  3.0000
```

```
ggplot(data=dsDataAll, aes(x=BsmtFullBath)) + geom_histogram(aes(color = indTrain))
```




```
#NO IDENTIFICO OUTLIERS
```

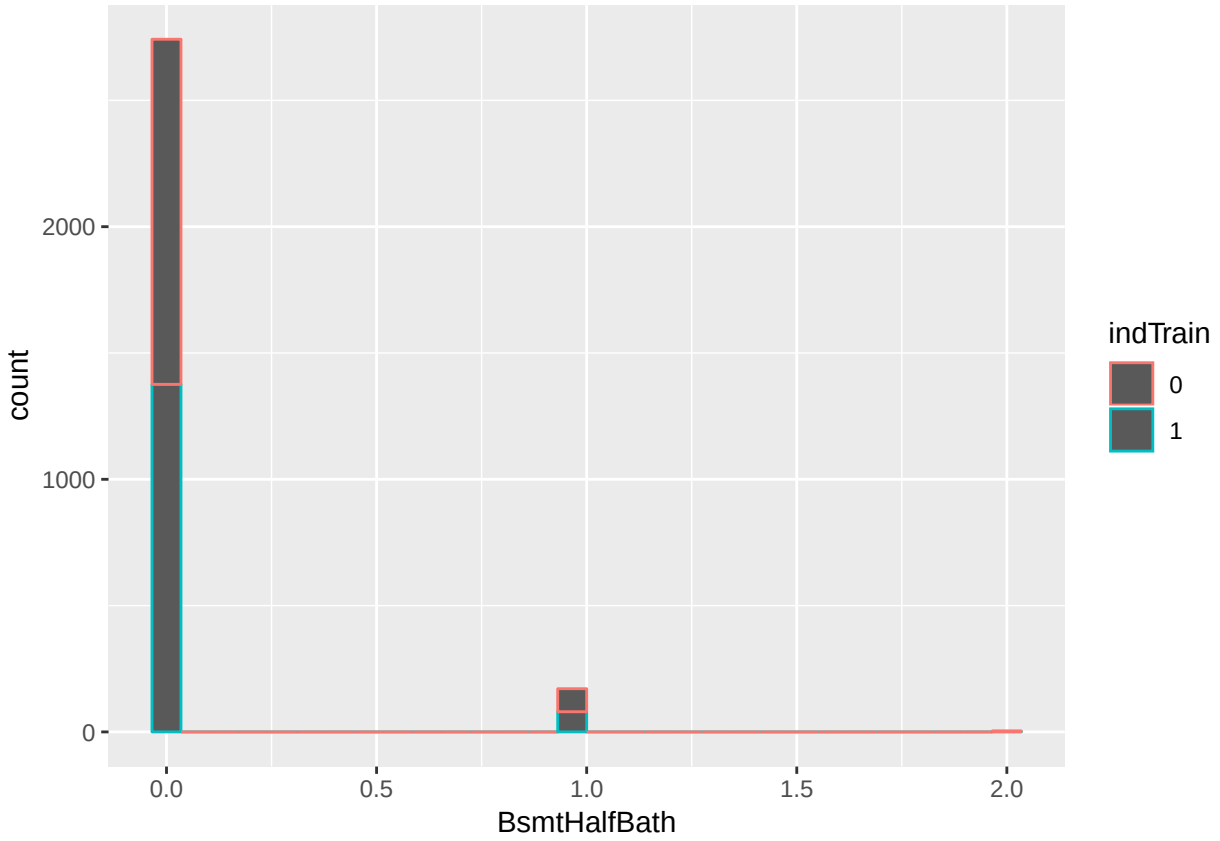
BsmtHalfBath medio baño en el sótano

NO IDENTIFICO OUTLIERS

```
summary(dsDataAll$BsmtHalfBath)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.06136 0.00000 2.00000
```

```
ggplot(data=dsDataAll, aes(x=BsmtHalfBath)) + geom_histogram(aes(color = indTrain))
```




```
#NO IDENTIFICO OUTLIERS
```

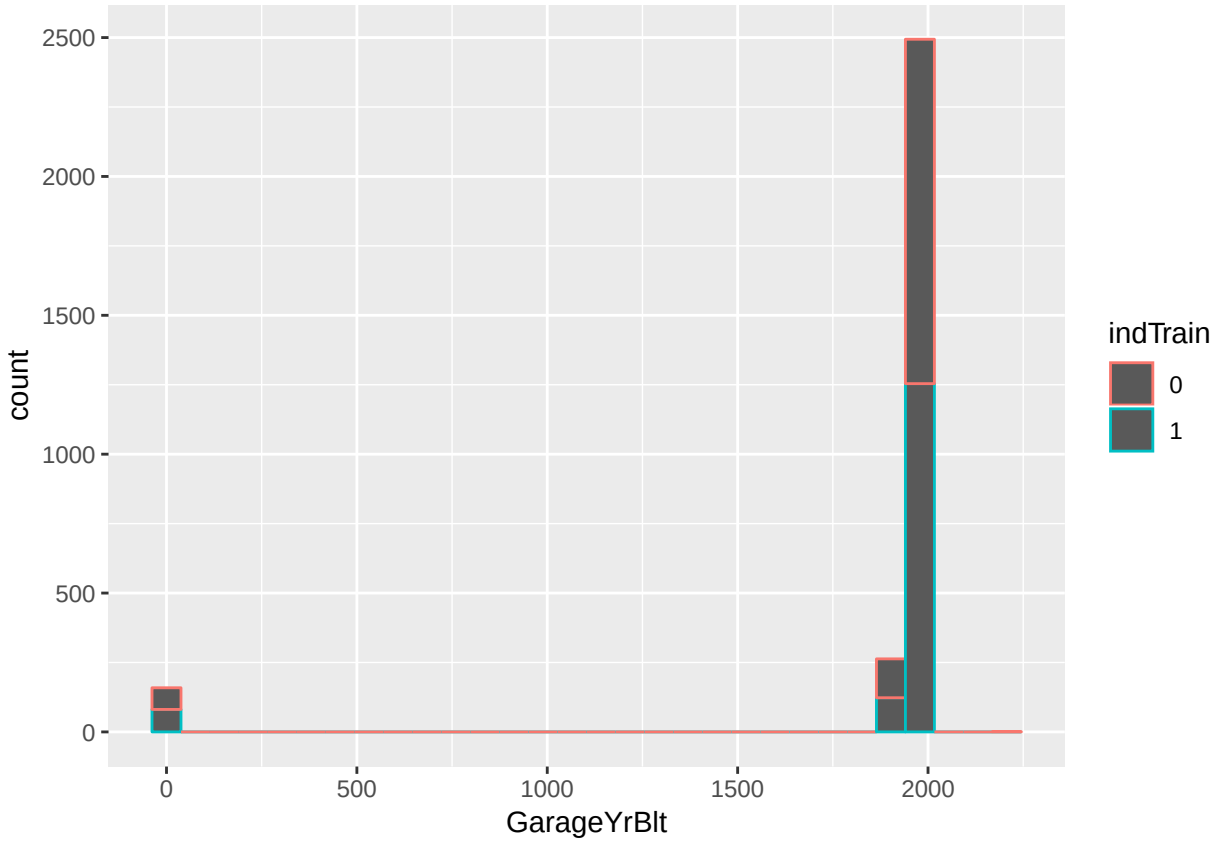
GarageYrBlt año en que se construyó el garaje

NO IDENTIFICO OUTLIERS

```
summary(dsDataAll$GarageYrBlt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    1957    1977    1870    2001    2207
```

```
ggplot(data=dsDataAll, aes(x=GarageYrBlt)) + geom_histogram(aes(color = indTrain))
```




```
# Esta variable parece incorrecta (Elimino)
dsDataAll <- select(dsDataAll, -GarageYrBlt)
```

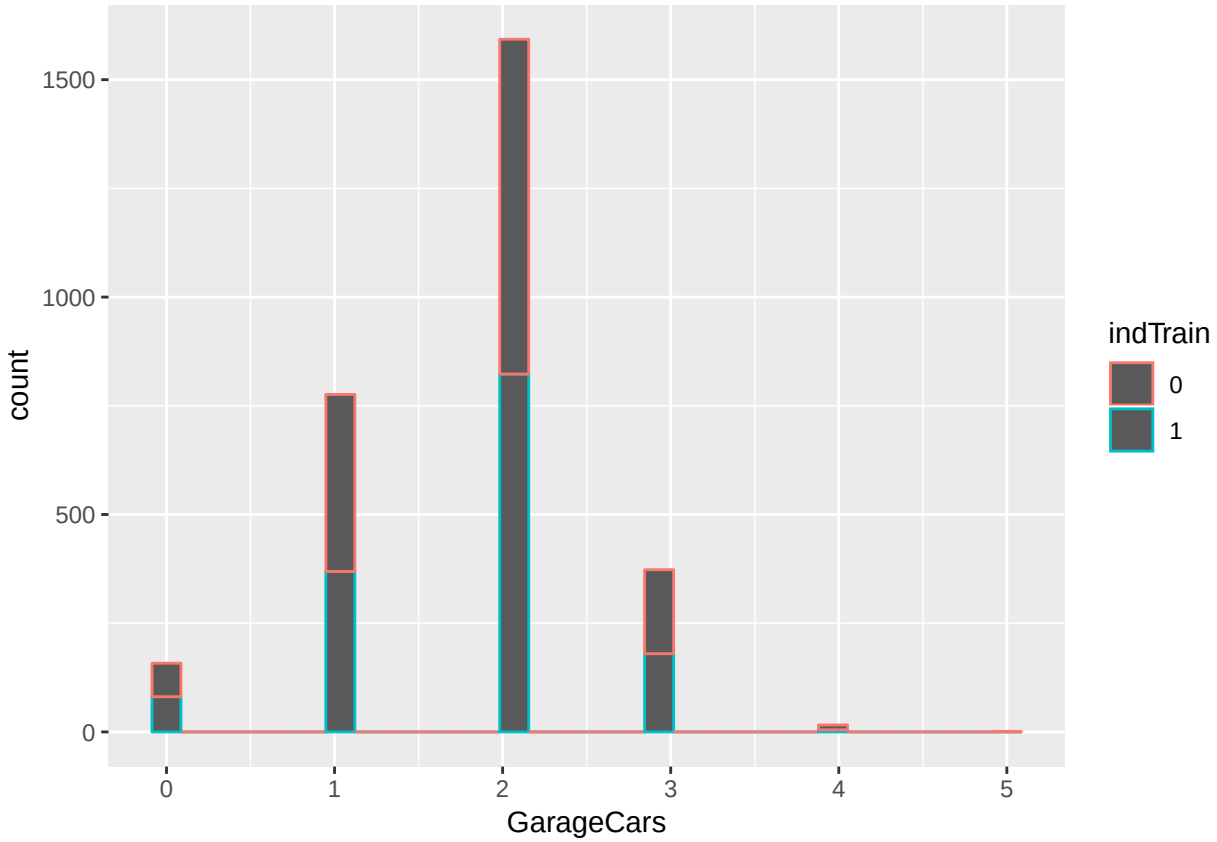
GarageCars tamaño del garaje en la capacidad del automóvil

NO IDENTIFICO OUTLIERS

```
summary(dsDataAll$GarageCars)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   2.000   1.766   2.000   5.000
```

```
ggplot(data=dsDataAll, aes(x=GarageCars)) + geom_histogram(aes(color = indTrain))
```




```
#NO IDENTIFICO OUTLIERS
```

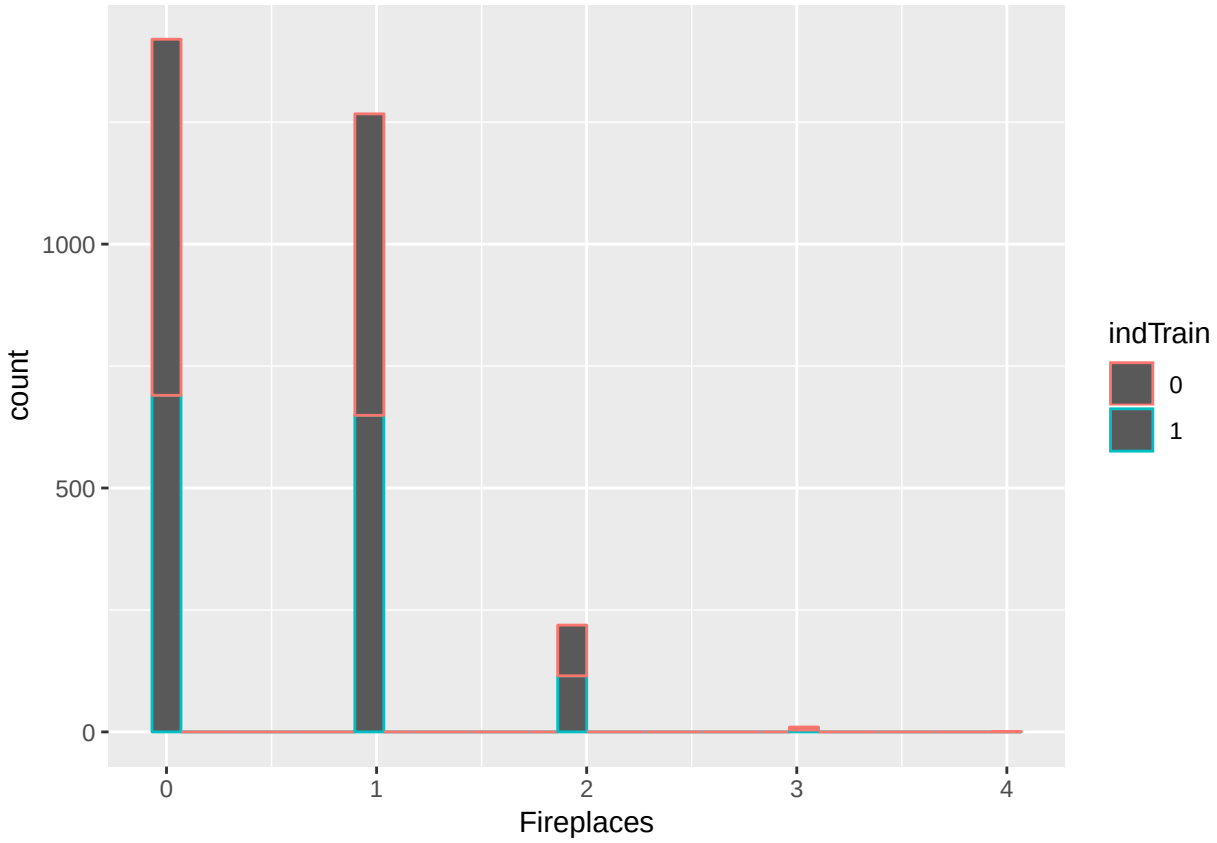
Fireplaces número de chimeneas

NO IDENTIFICO OUTLIERS

```
summary(dsDataAll$Fireplaces)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  1.0000  0.5962  1.0000  4.0000
```

```
ggplot(data=dsDataAll, aes(x=Fireplaces)) + geom_histogram(aes(color = indTrain))
```




```
#NO IDENTIFICO OUTLIERS
```

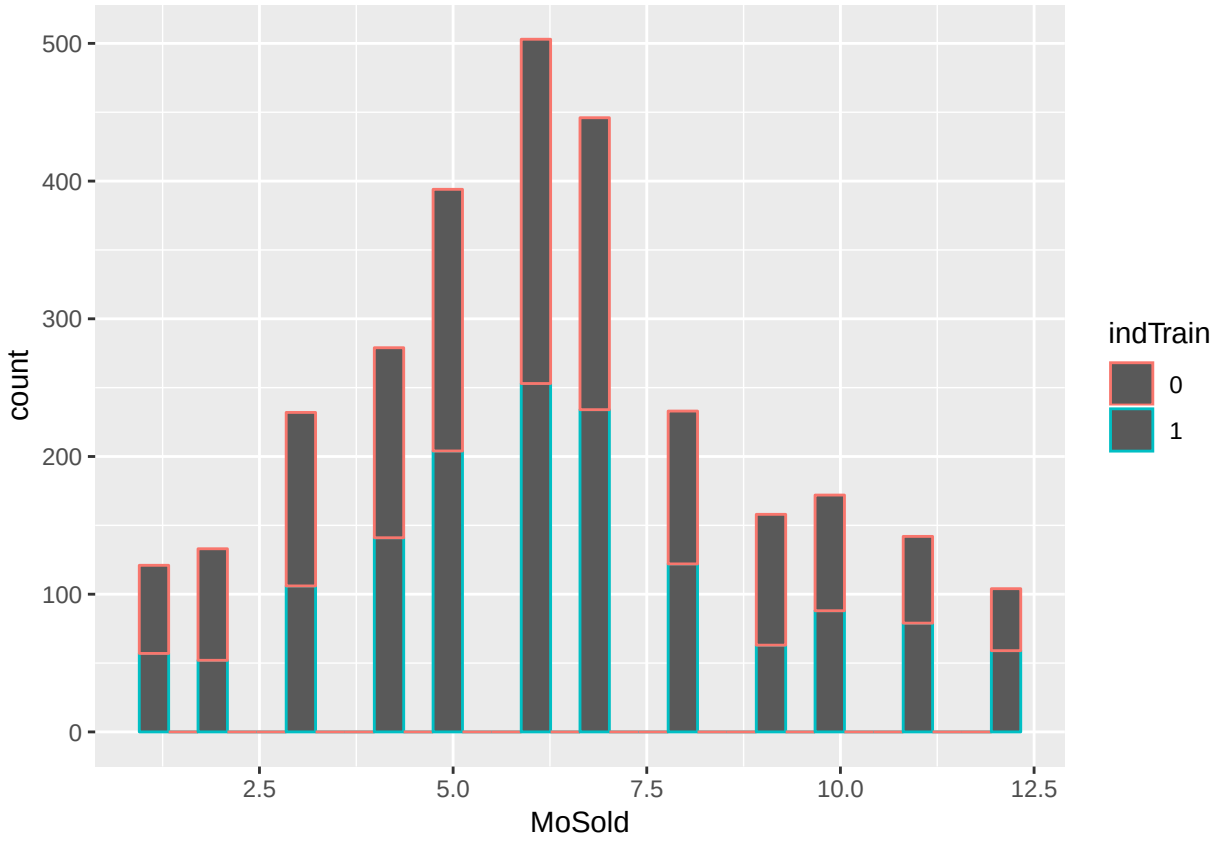
MoSold Mes vendido

NO IDENTIFICO OUTLIERS

```
summary(dsDataAll$MoSold)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   6.000   6.214   8.000  12.000
```

```
ggplot(data=dsDataAll, aes(x=MoSold)) + geom_histogram(aes(color = indTrain))
```




```
#NO IDENTIFICO OUTLIERS
```

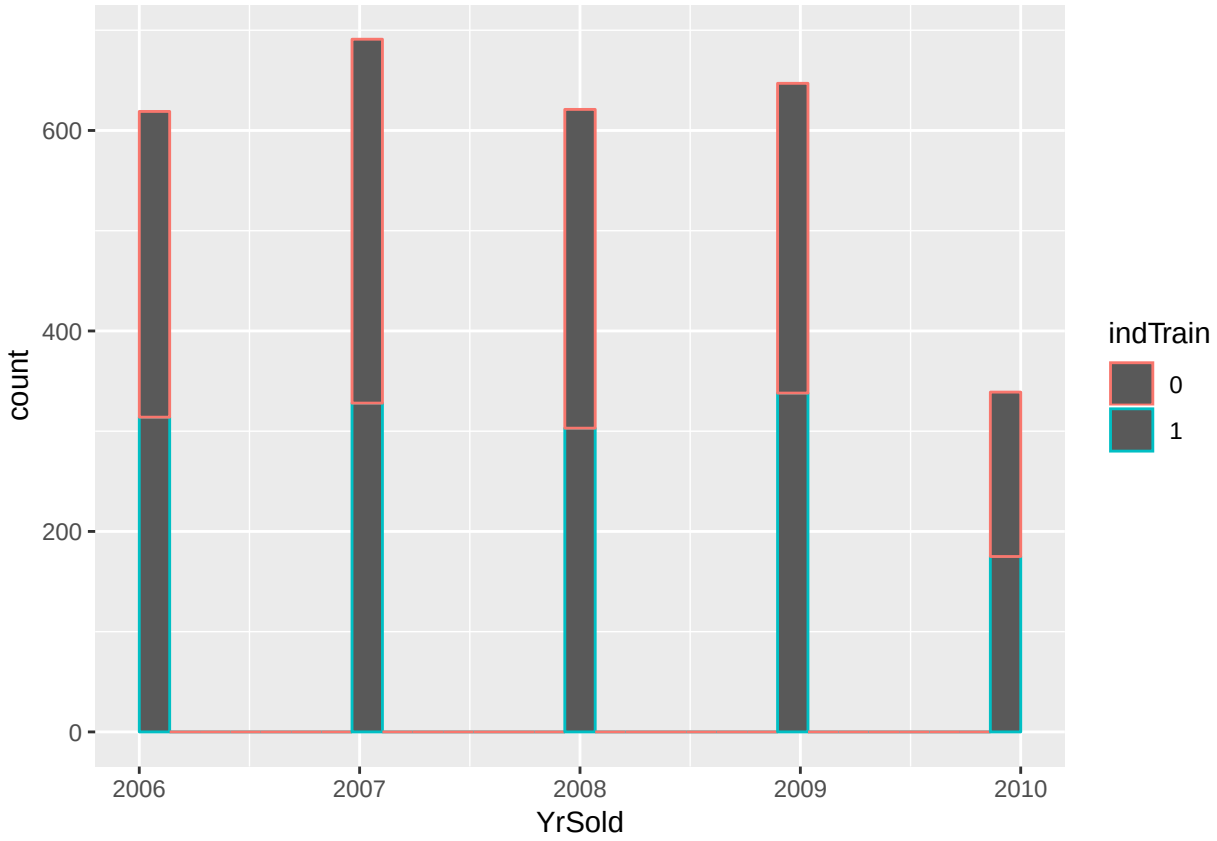
YrSold año vendido

NO IDENTIFICO OUTLIERS Ventas 2006 A 2010

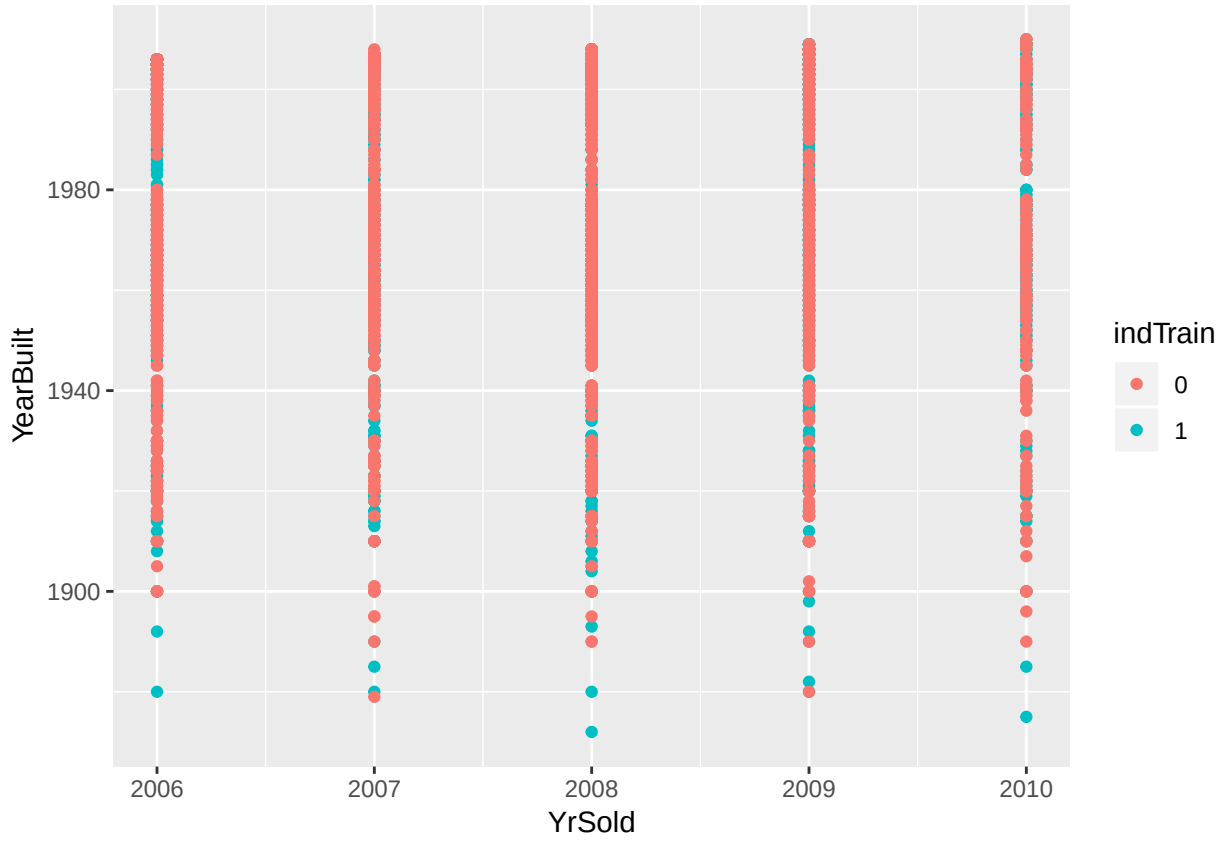
```
summary(dsDataAll$YrSold)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2006    2007    2008    2008    2009    2010
```

```
ggplot(data=dsDataAll, aes(x=YrSold)) + geom_histogram(aes(color = indTrain))
```




```
ggplot(dsDataAll, aes(x=YrSold, y=YearBuilt)) + geom_point(aes(color = indTrain))
```




```
dsDataAll %>%
  filter(YrSold<YearBuilt) %>%
  select(Id,YrSold,YearBuilt,SaleType)
```

```
## Source: local data frame [1 x 4]
## Groups: <by row>
##
## # A tibble: 1 x 4
##      Id YrSold YearBuilt SaleType
##   <int>  <int>     <int> <fct>
## 1  2550   2007      2008 New
```

```
# VENTAS DE 2006 A 2010
# NO IDENTIFICO OUTLIERS
```

Salvar progreso

```
dsDataAll <- as.data.frame(dsDataAll)
save(dsDataAll, file = './F01_Datos/F01_dsDataAll.RData')
# load('./F01_Datos/F01_dsDataAll.RData')
# str(dsDataAll$MSSubClass)

rm(dsCampos)
rm(dsCamposContinua)
rm(dsCamposDiscreta)
```

Estudio de correlaciones

El número tan elevado de variable, 73 después de la primera limpieza, impide realizar una sola matriz que nos muestre las correlaciones por lo que he optado por separar en varias matrices..

Además para poder incluir en el estudio todas las variables es necesario que sean númericas.

Conversión de variables nominales a Dummy (Esta modificación no se guarda, se realizará en la fase 2)

```
dsCamposActuales <- data.frame(unlist(sapply(dsDataAll, class))) %>%
  select(Tipo = 1) %>%
  rownames_to_column("Campo")

dsCamposFactor <- filter(dsCamposActuales, Tipo == "factor" & Campo != "indTrain") %>% select(Campo)

dsDummy <- dsDataAll %>% select(c("Id",c(dsCamposFactor$Campo)))
dsDummy <- fastDummies::dummy_cols(dsDummy)
dsDummy <- select(dsDummy,-c(dsCamposFactor$Campo))

dsDataAll <- select(dsDataAll,-c(dsCamposFactor$Campo))

dsDataAll <- dsDataAll %>%
    inner_join(dsDummy, by="Id")

rm(dsCamposFactor)
rm(dsCamposActuales)
rm(dsDummy)
```

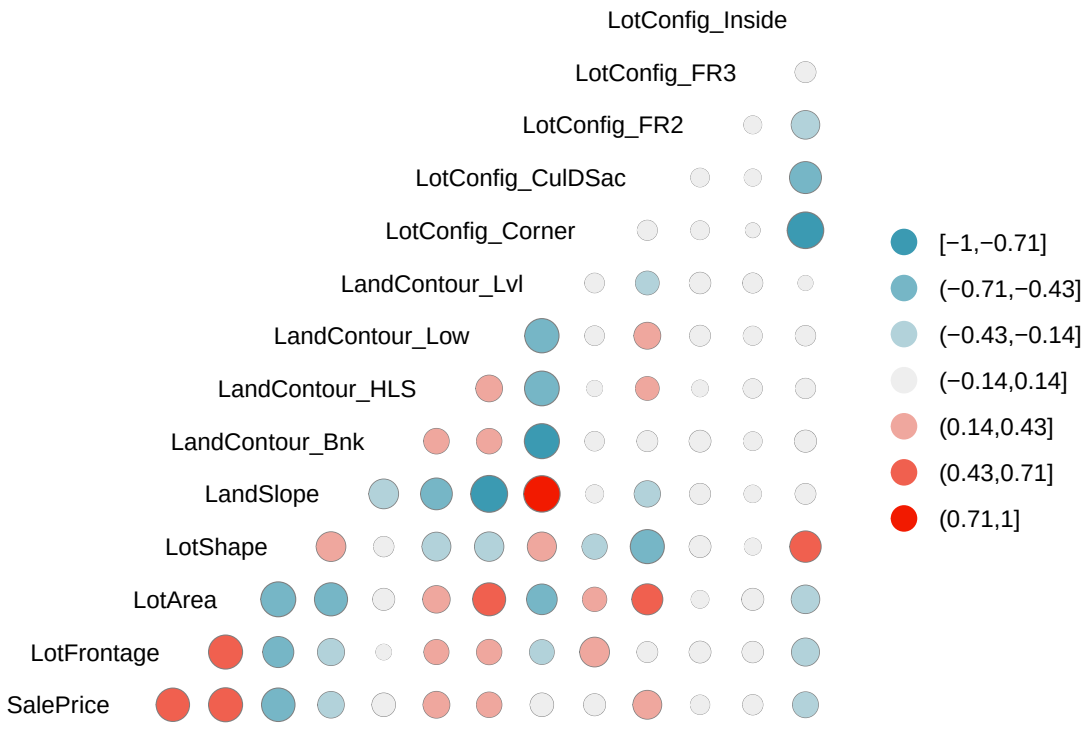
Selecciono variables según la clasificación realizada manualmente (segmento y subsegmento)

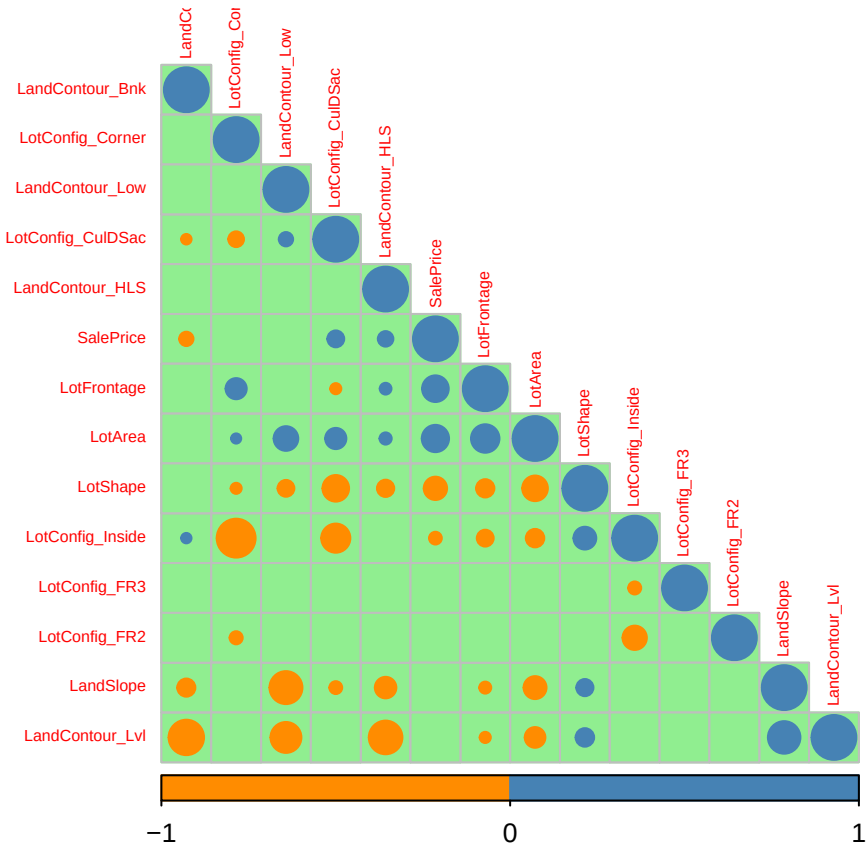
```
dsCamposOriginales <- read.csv("./input/campos.csv",sep=";",stringsAsFactors = FALSE)

dsCamposOriginales <- dsCamposOriginales %>%
  mutate_if(is.factor, as.character)

# Obtengo campos actuales en dsDataAll
dsCamposActuales <- data.frame(unlist(sapply(dsDataAll, class))) %>%
  select(Tipo = 1) %>%
  rownames_to_column("CampoNuevo") %>%
  filter(CampoNuevo != "Id" & CampoNuevo != "indTrain")

# Para los nuevos campos Dummies busco los campos origen
dsCamposNominalDummy <- dsCamposActuales %>%
  filter(grepl('[_]', CampoNuevo)) %>%
  mutate(CampoOrigen = sub("_.*", "", CampoNuevo)) %>%
  select(CampoNuevo, CampoOrigen)

dsCamposActuales <- dsCamposActuales %>%
  left_join(dsCamposNominalDummy,by = "CampoNuevo") %>%
  mutate(CampoOrigen = ifelse(is.na(CampoOrigen),CampoNuevo,CampoOrigen)) %>%
  mutate(CampoOrigen = ifelse(CampoNuevo=="StreetPave","Street",CampoOrigen))

# Busco se
dsCamposActuales <- dsCamposActuales %>%
```

```
  inner_join(dsCamposOriginales, by = c("CampoOrigen" = "Campo")) %>%
  select(CampoNuevo, CampoOrigen, Segmento,Subsegmento)%>%
  rename(Campo = CampoNuevo) %>%
  mutate(grupo = paste(Segmento,Subsegmento,sep="-"))

# Verificación de los grupos que se han formado
# dsCamposActuales %>%
#   group_by(grupo) %>%
#   tally()

grupos <- distinct(dsCamposActuales,  grupo) %>% filter(grupo != '-')
grupos <- grupos[,1]

rm(dsCamposOriginales)
rm(dsCamposNominalDummy)
```

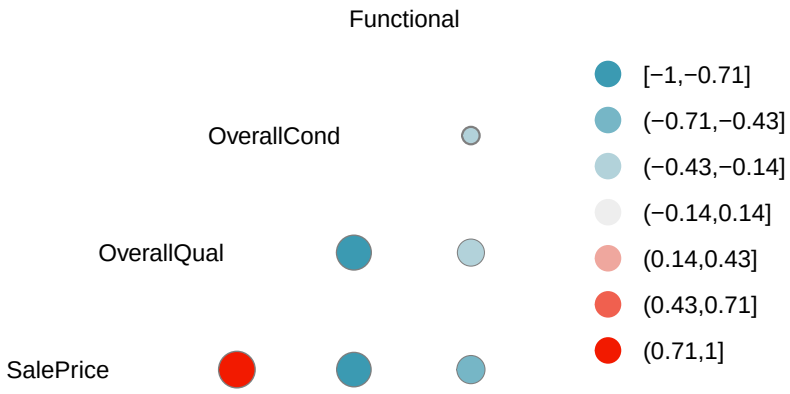
Matrices de correlaciones por grupos

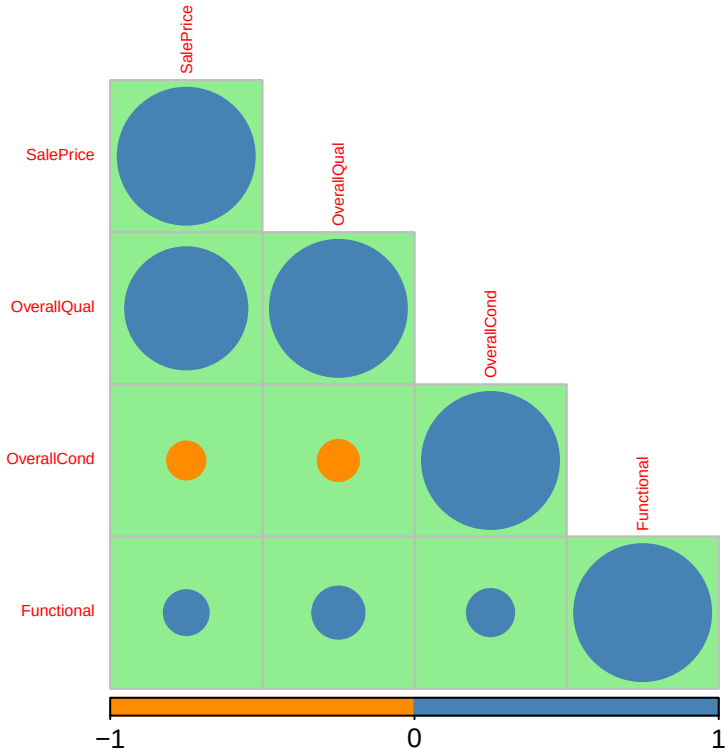
```
for(i in grupos){
  campos <- dsCamposActuales %>%
    filter(grupo==i) %>%
    select(Campo)

  datos <- dsDataAll %>%
    filter(indTrain == 1) %>%
    select(SalePrice,c(campos$Campo))

  datosMatriz <- as.matrix(datos)

  corGrupo <- rcorr(datosMatriz, type="pearson")

  # Presentamos la matríz de correlación en variables continuas
  p1 <- ggcorr(corGrupo$r, geom = "circle", nbreaks = 7, size = 3, hjust= 1.1, layout.exp = 4)

  grid.arrange(p1, nrow = 1, top=i)

  # Utilizamos corrplot para reordenar la matriz, esto permite ver relaciones de forma mas fácil
  # No he podido convertir corrplot a grob

  corGrupo$r[is.na(corGrupo$r)] <- 0

  corrplot(corGrupo$r, p.mat = corGrupo$P, sig.level = 0.05
                , insig = "blank", tl.cex = 0.5, type = "lower"
                , order = "hclust", col = c("darkorange", "steelblue")
                , bg = "lightgreen")
}
```

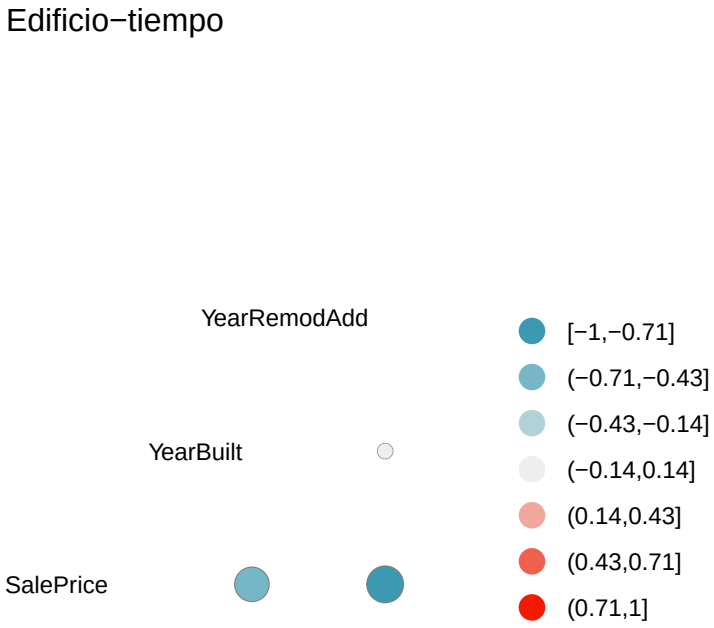

Proiedad−
LotConfig_Inside
LotConfig_FR3
LotConfig_FR2
LotConfig_CulDSac
LotConfig_Corner
LandContour_Lvl
LandContour_Low
LandContour_HLS
LandContour_Bnk
LandSlope
LotShape
LotArea
LotFrontage
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]

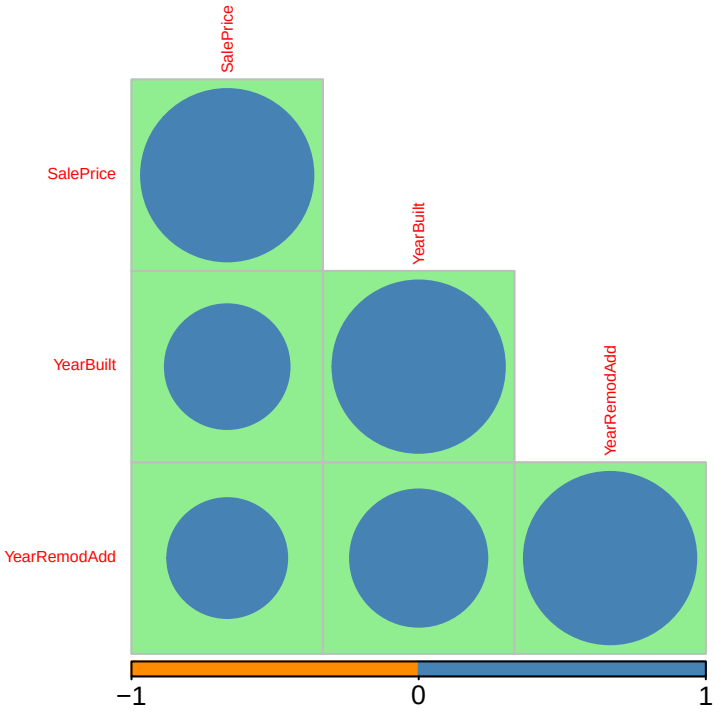

LandC
LotConfig_Cor
LandContour_Low
LotConfig_CulDSac
LandContour_HLS
SalePrice
LotFrontage
LotArea
LotShape
LotConfig_Inside
LotConfig_FR3
LotConfig_FR2
LandSlope
LandContour_Lvl
LandContour_Bnk
LotConfig_Corner
LandContour_Low
LotConfig_CulDSac
LandContour_HLS
SalePrice
LotFrontage
LotArea
LotShape
LotConfig_Inside
LotConfig_FR3
LotConfig_FR2
LandSlope
LandContour_Lvl
−1
0
1

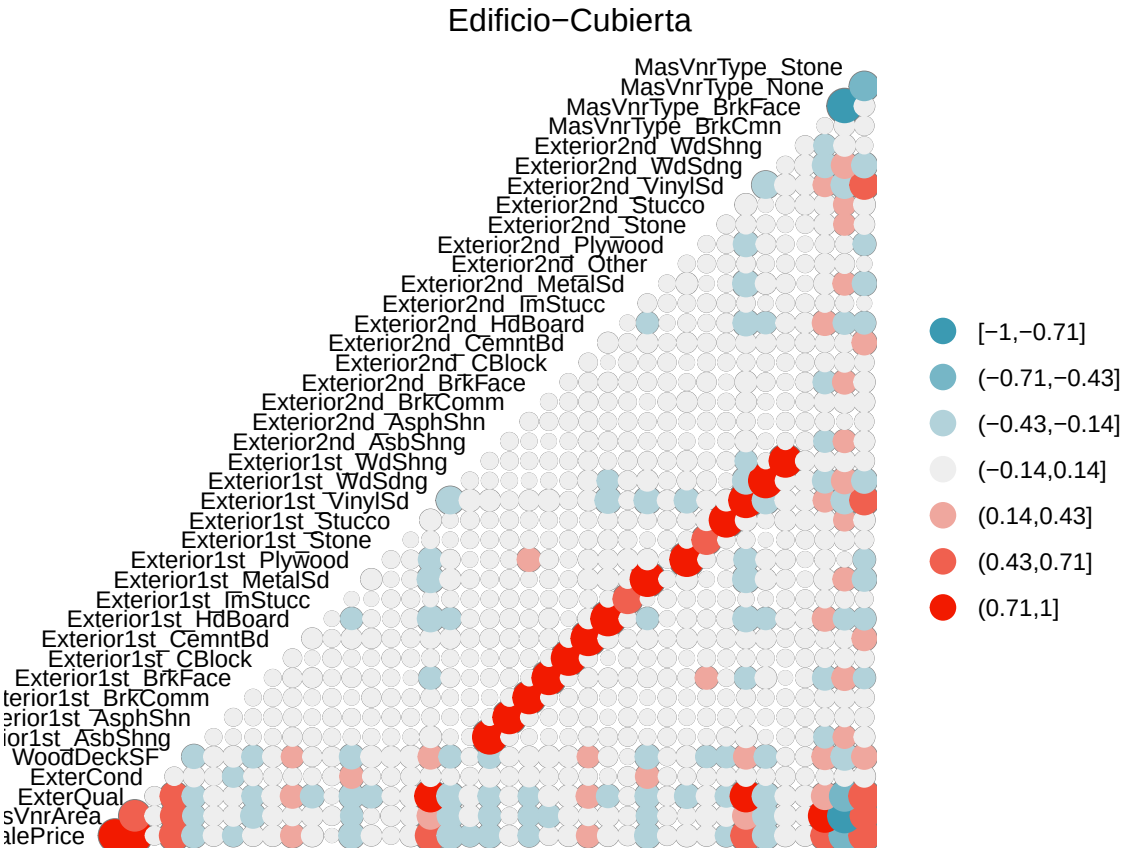
Edificio−calidad

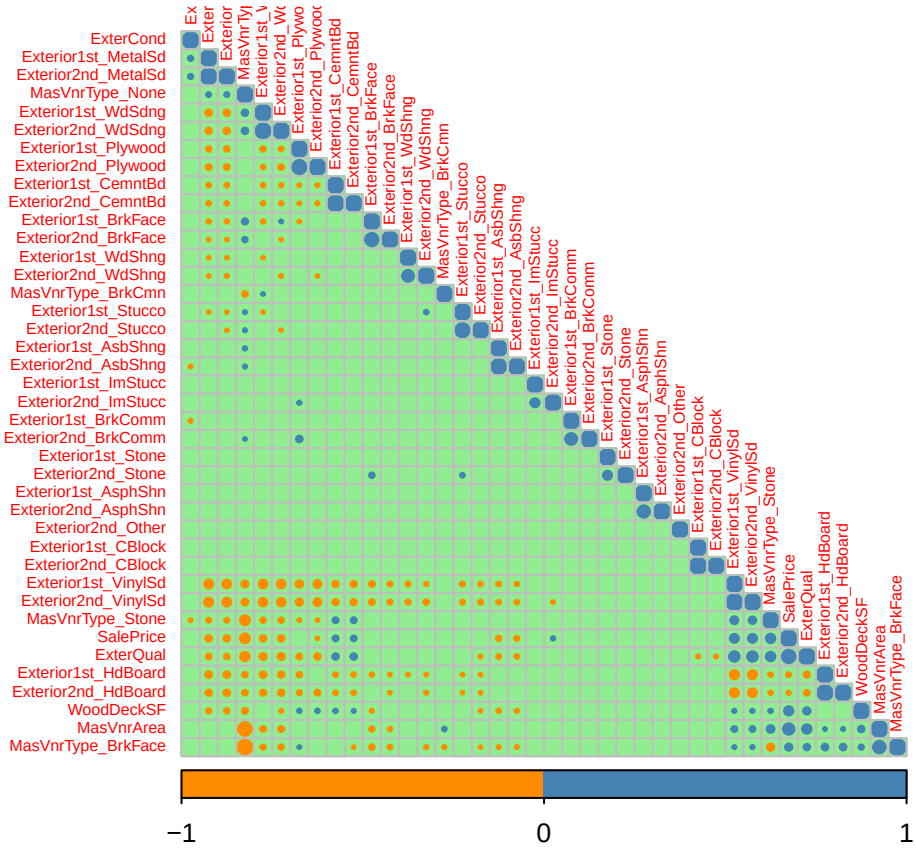


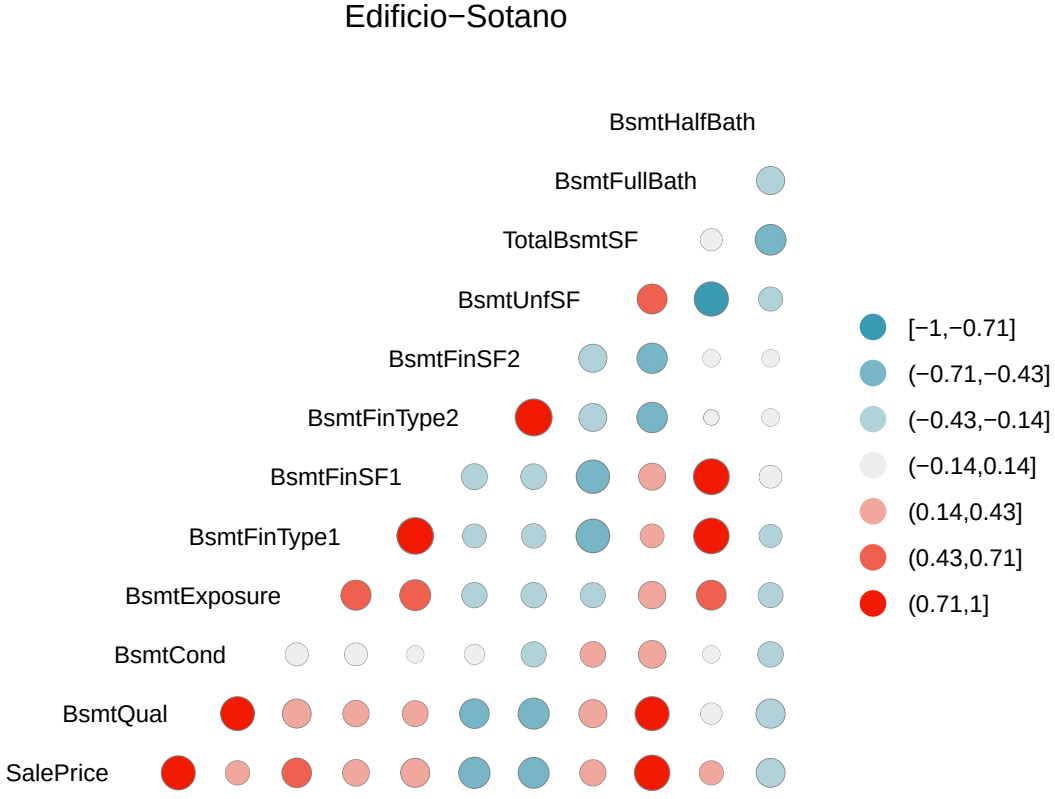

SalePrice
OverallQual
OverallCond
Functional
SalePrice
OverallQual
OverallCond
Functional
−1
0
1

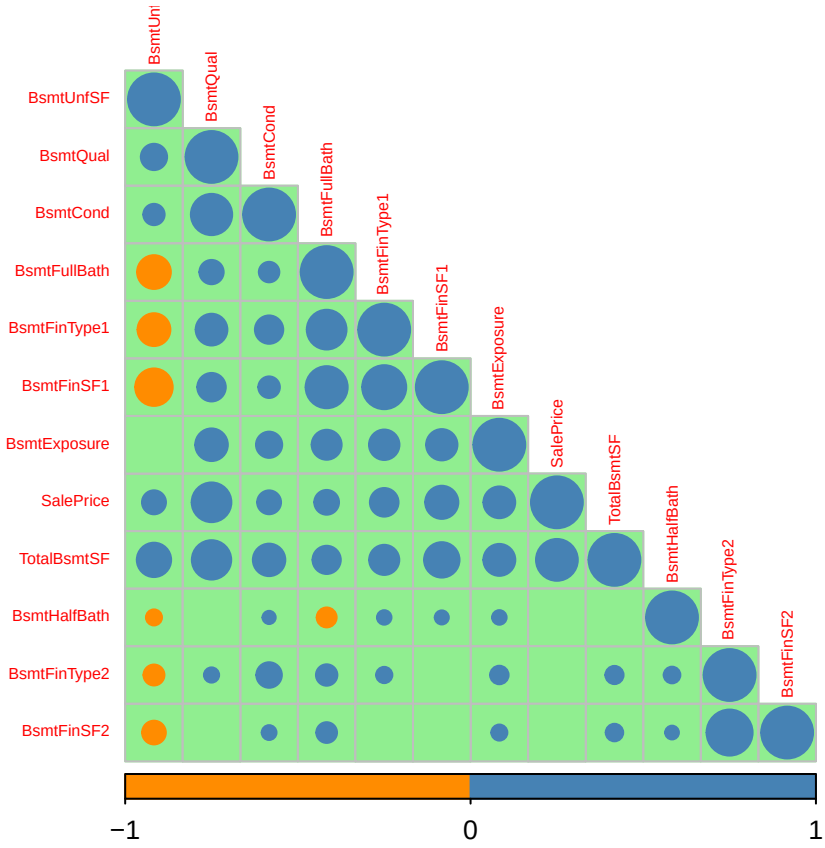

Edificio−tiempo
YearRemodAdd
YearBuilt
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]


SalePrice
YearBuilt
YearRemodAdd
SalePrice
YearBuilt
YearRemodAdd
−1
0
1

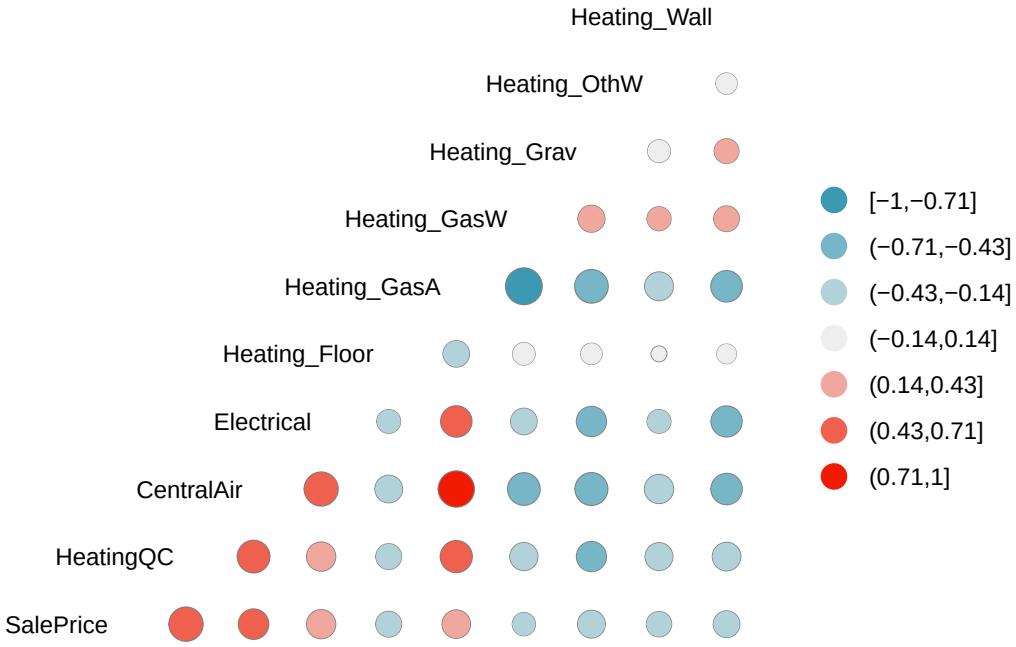

Edificio−Cubierta
MasVnrType_Stone
MasVnrType_None
MasVnrType_BrkFace
MasVnrType_BrkCmn
Exterior2nd_WdShng
Exterior2nd_WdSdng
Exterior2nd_VinylSd
Exterior2nd_Stucco
Exterior2nd_Stone
Exterior2nd_Plywood
Exterior2nd_Other
Exterior2nd_MetalSd
Exterior2nd_ImStucc
Exterior2nd_HdBoard
Exterior2nd_CemntBd
Exterior2nd_CBlock
Exterior2nd_BrkFace
Exterior2nd_BrkComm
Exterior2nd_AsphShn
Exterior2nd_AsbShng
Exterior1st_WdShng
Exterior1st_WdSdng
Exterior1st_VinylSd
Exterior1st_Stucco
Exterior1st_Stone
Exterior1st_Plywood
Exterior1st_MetalSd
Exterior1st_ImStucc
Exterior1st_HdBoard
Exterior1st_CemntBd
Exterior1st_CBlock
Exterior1st_BrkFace
terior1st_BrkComm
erior1st_AsphShn
ior1st_AsbShng
WoodDeckSF
ExterCond
ExterQual
sVnrArea
alePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]

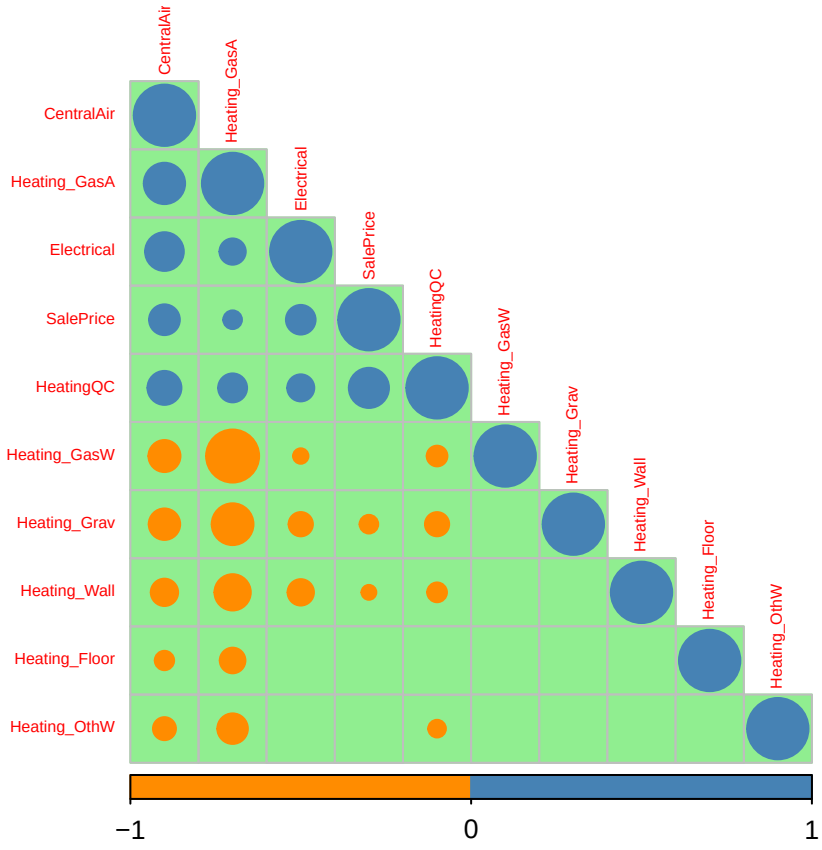

ExterCond
Exterior1st_MetalSd
Exterior2nd_MetalSd
MasVnrType_None
Exterior1st_WdSdng
Exterior2nd_WdSdng
Exterior1st_Plywood
Exterior2nd_Plywood
Exterior1st_CemntBd
Exterior2nd_CemntBd
Exterior1st_BrkFace
Exterior2nd_BrkFace
Exterior1st_WdShng
Exterior2nd_WdShng
MasVnrType_BrkCmn
Exterior1st_Stucco
Exterior2nd_Stucco
Exterior1st_AsbShng
Exterior2nd_AsbShng
Exterior1st_ImStucc
Exterior2nd_ImStucc
Exterior1st_BrkComm
Exterior2nd_BrkComm
Exterior1st_Stone
Exterior2nd_Stone
Exterior1st_AsphShn
Exterior2nd_AsphShn
Exterior2nd_Other
Exterior1st_CBlock
Exterior2nd_CBlock
Exterior1st_VinylSd
Exterior2nd_VinylSd
MasVnrType_Stone
SalePrice
ExterQual
Exterior1st_HdBoard
Exterior2nd_HdBoard
WoodDeckSF
MasVnrArea
MasVnrType_BrkFace
−1
0
1


Edificio−Sotano
BsmtHalfBath
BsmtFullBath
TotalBsmtSF
BsmtUnfSF
BsmtFinSF2
BsmtFinType2
BsmtFinSF1
BsmtFinType1
BsmtExposure
BsmtCond
BsmtQual
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]

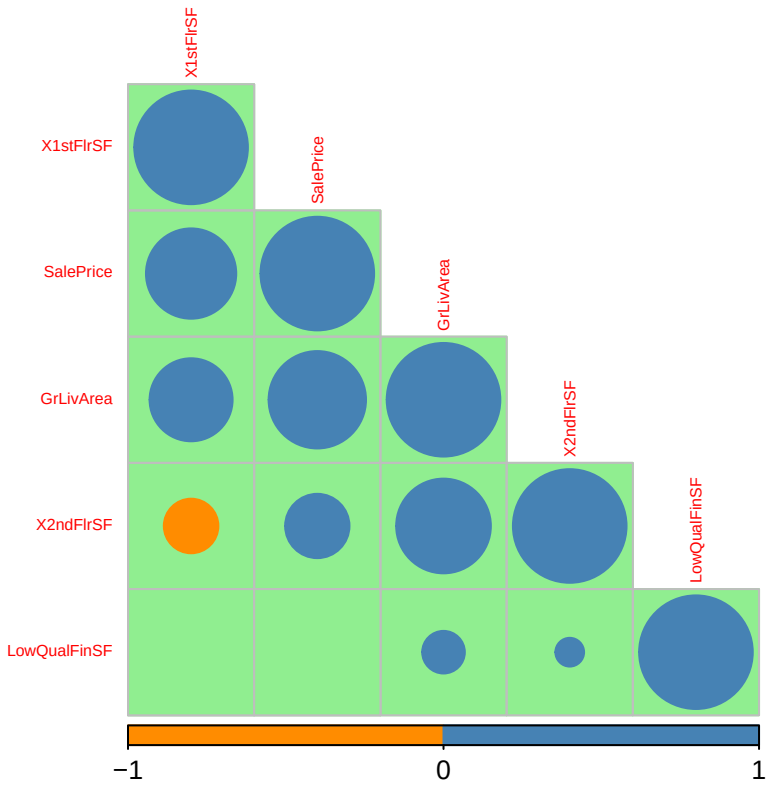

BsmtUnf
BsmtQual
BsmtCond
BsmtFullBath
BsmtFinType1
BsmtFinSF1
BsmtExposure
SalePrice
TotalBsmtSF
BsmtHalfBath
BsmtFinType2
BsmtFinSF2
BsmtUnfSF
BsmtQual
BsmtCond
BsmtFullBath
BsmtFinType1
BsmtFinSF1
BsmtExposure
SalePrice
TotalBsmtSF
BsmtHalfBath
BsmtFinType2
BsmtFinSF2
−1
0
1

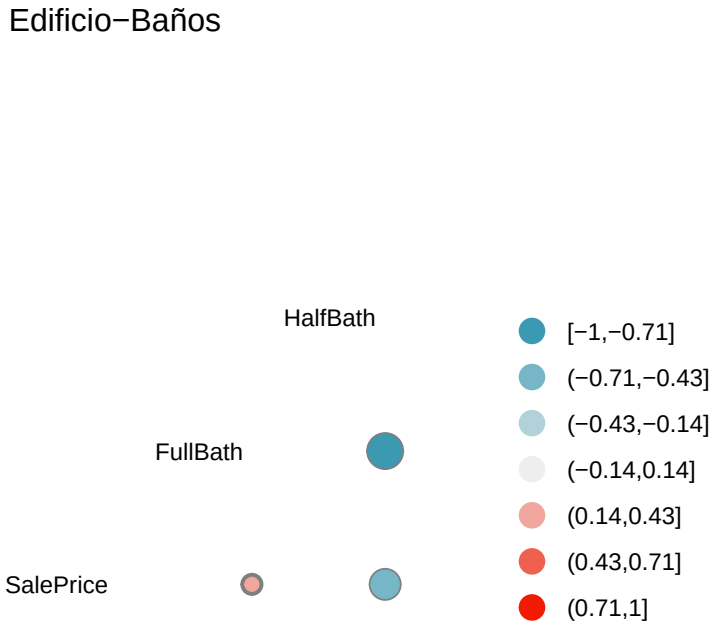
Edificio−Servicios

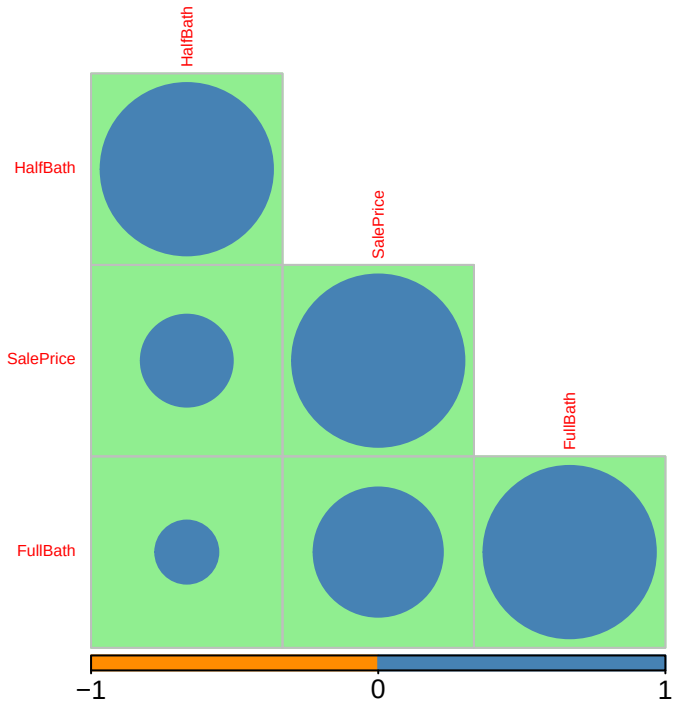



CentralAir
Heating_GasA
Electrical
SalePrice
HeatingQC
Heating_GasW
Heating_Grav
Heating_Wall
Heating_Floor
Heating_OthW
−1
0
1

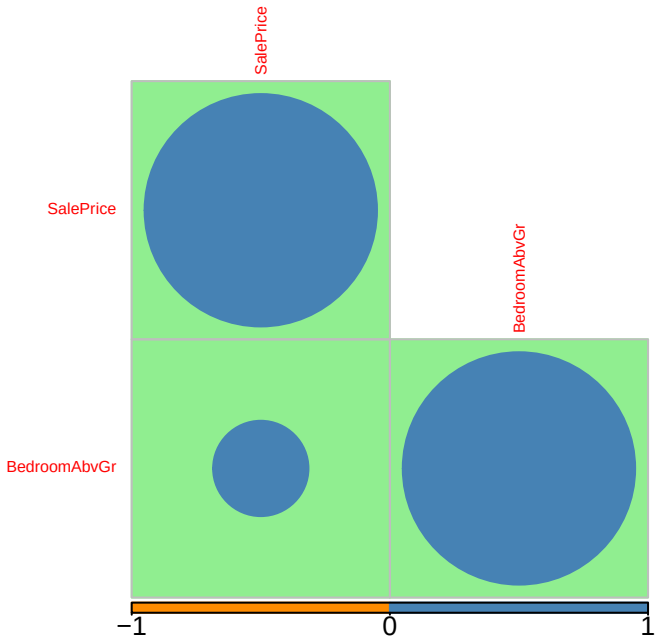

Edificio−superficie
GrLivArea
LowQualFinSF
X2ndFlrSF
X1stFlrSF
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]

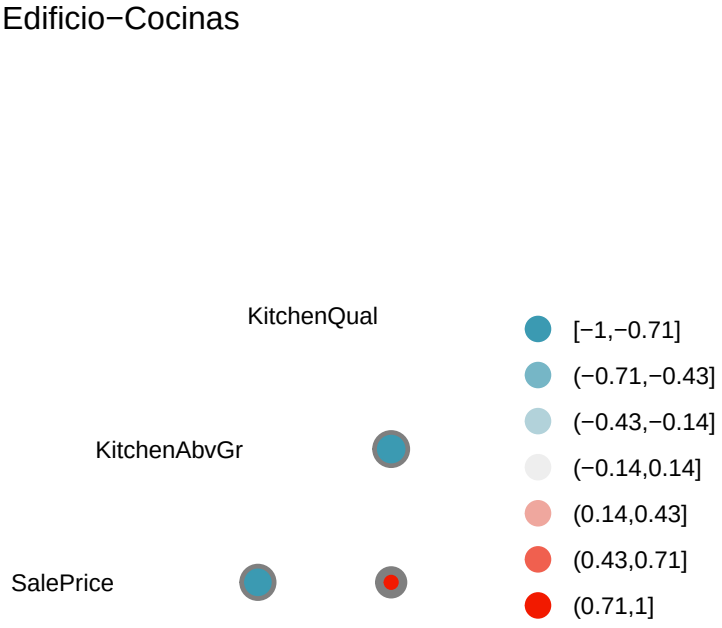

X1stFlrSF
SalePrice
GrLivArea
X2ndFlrSF
LowQualFinSF
−1
0
1

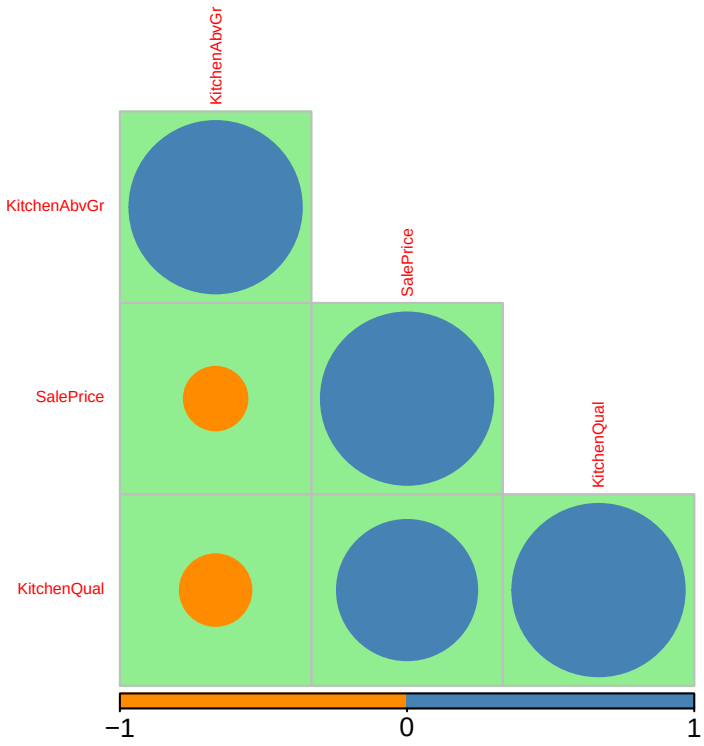

Edificio−Baños
HalfBath
FullBath
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]

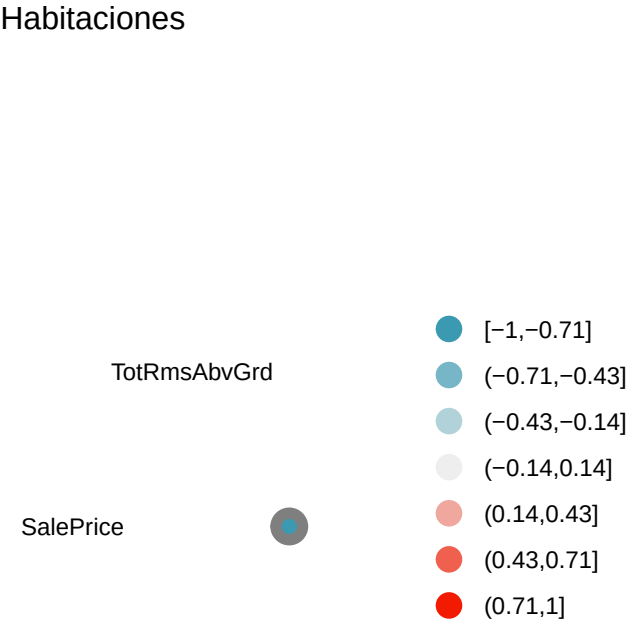

HalfBath
SalePrice
FullBath
HalfBath
SalePrice
FullBath
−1
0
1

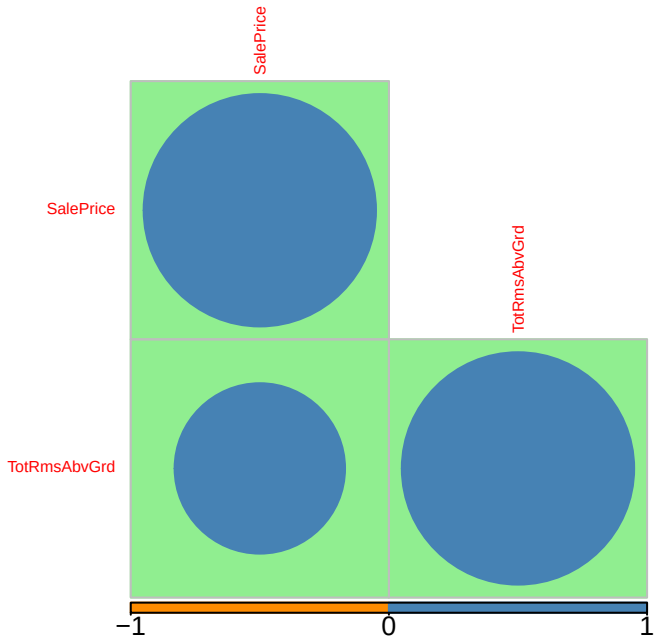

Edificio−Dormitorios
BedroomAbvGr
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]


SalePrice
SalePrice
BedroomAbvGr
BedroomAbvGr
−1
0
1

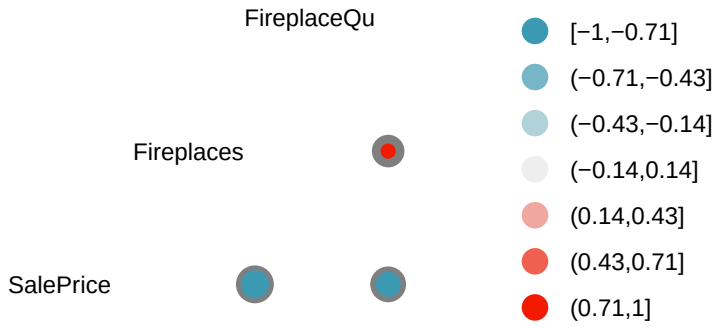

Edificio−Cocinas
KitchenQual
KitchenAbvGr
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]

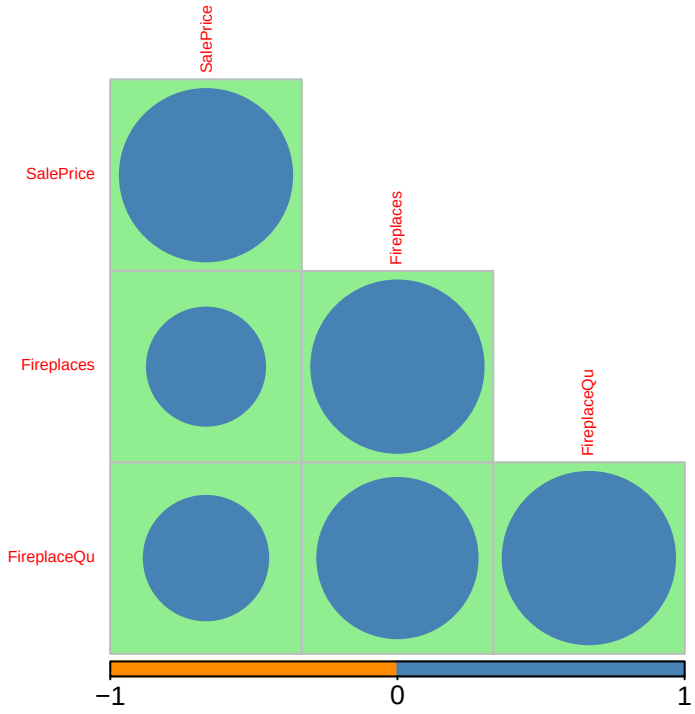

KitchenAbvGr
SalePrice
KitchenQual
KitchenAbvGr
SalePrice
KitchenQual
−1
0
1

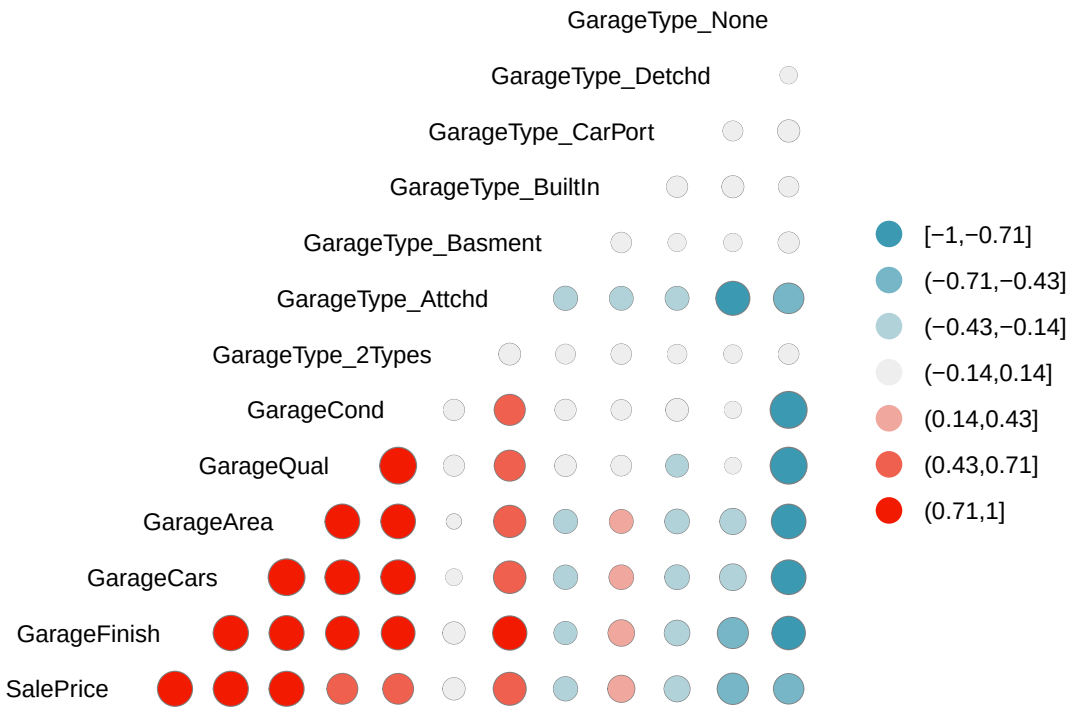

Edificio−Habitaciones
TotRmsAbvGrd
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]

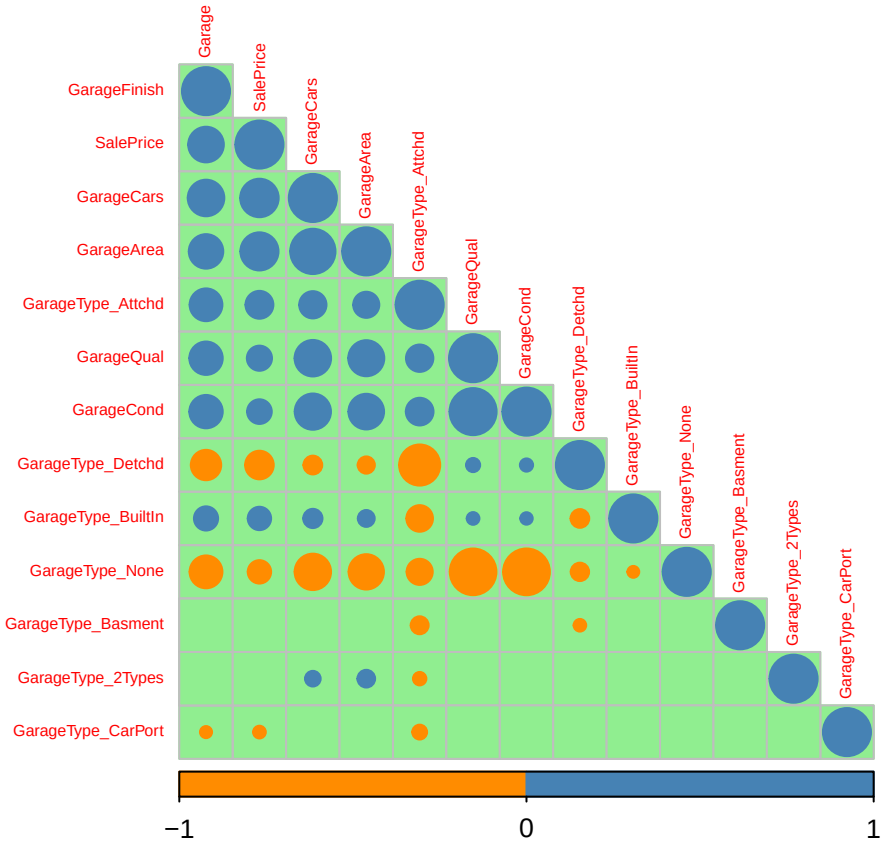

SalePrice
SalePrice
TotRmsAbvGrd
TotRmsAbvGrd
−1
0
1

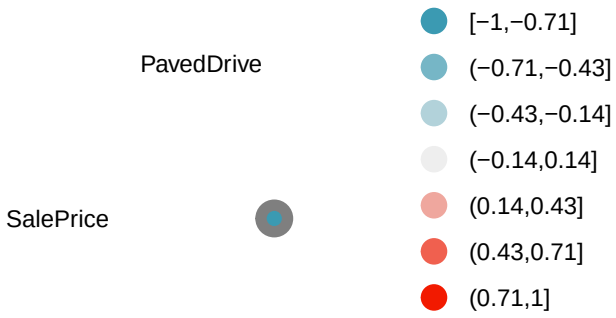
Edificio−Chimenea

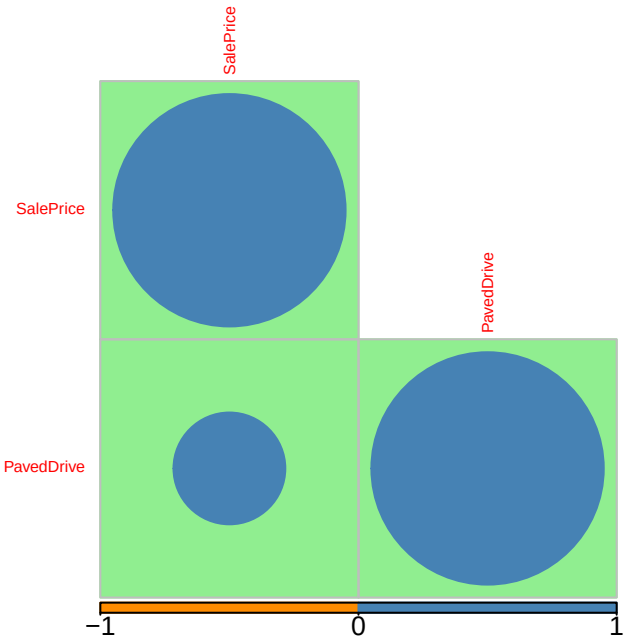


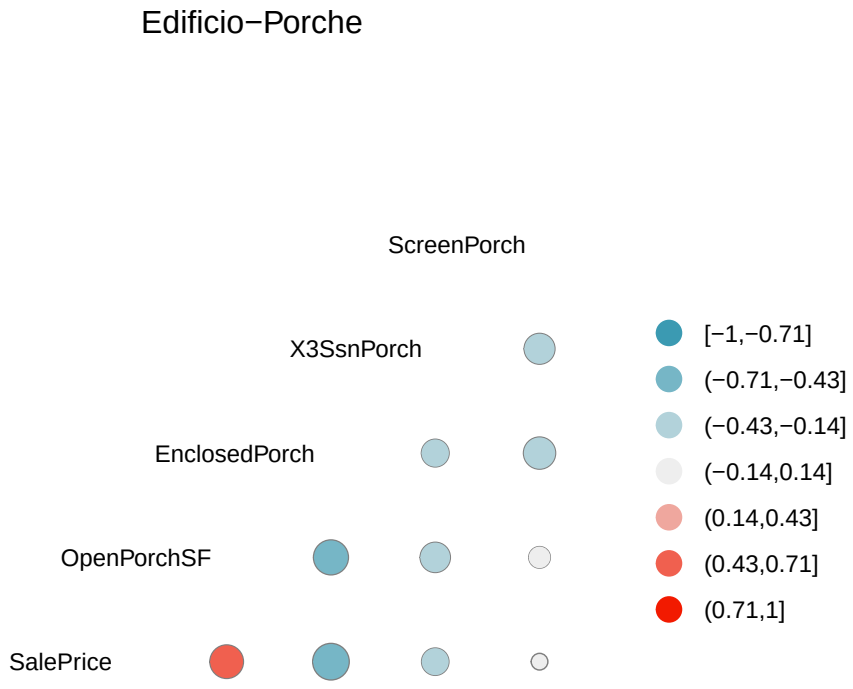

SalePrice
SalePrice
Fireplaces
Fireplaces
FireplaceQu
FireplaceQu
−1
0
1

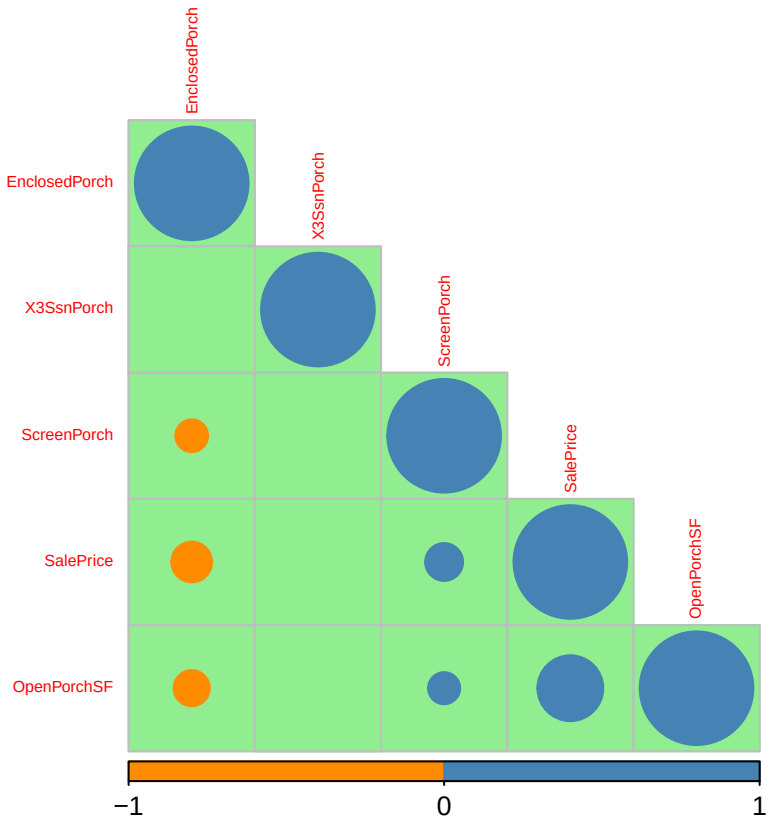

Edificio−Garage
GarageType_None
GarageType_Detchd
GarageType_CarPort
GarageType_BuiltIn
GarageType_Basment
GarageType_Attchd
GarageType_2Types
GarageCond
GarageQual
GarageArea
GarageCars
GarageFinish
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]


Garage
SalePrice
GarageCars
GarageArea
GarageType_Attchd
GarageQual
GarageCond
GarageType_Detchd
GarageType_BuiltIn
GarageType_None
GarageType_Basment
GarageType_2Types
GarageType_CarPort
GarageFinish
SalePrice
GarageCars
GarageArea
GarageType_Attchd
GarageQual
GarageCond
GarageType_Detchd
GarageType_BuiltIn
GarageType_None
GarageType_Basment
GarageType_2Types
GarageType_CarPort
−1
0
1

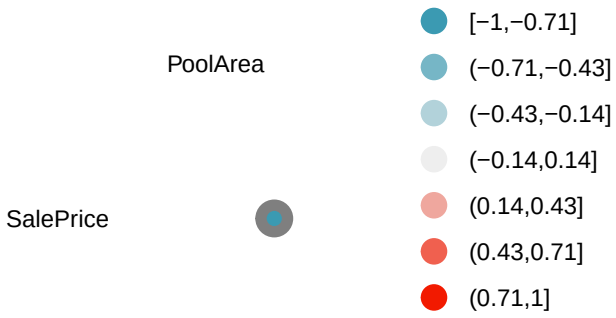

Edificio−Entrada
PavedDrive
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]

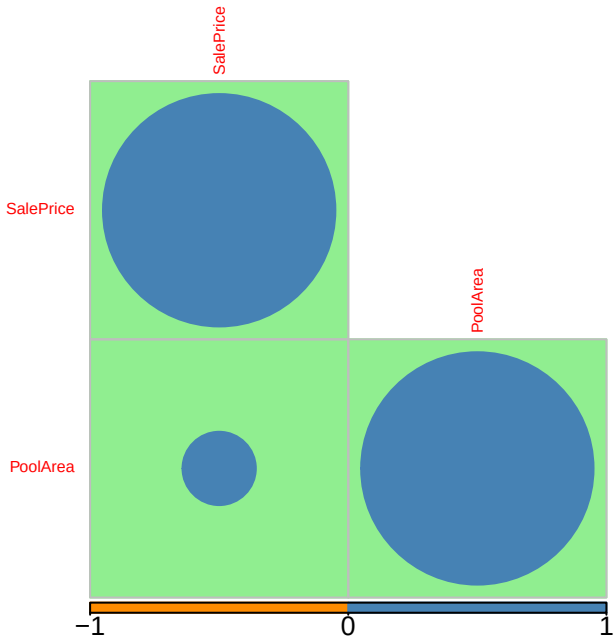

SalePrice
SalePrice
PavedDrive
PavedDrive
−1
0
1


Edificio−Porche
ScreenPorch
X3SsnPorch
EnclosedPorch
OpenPorchSF
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]

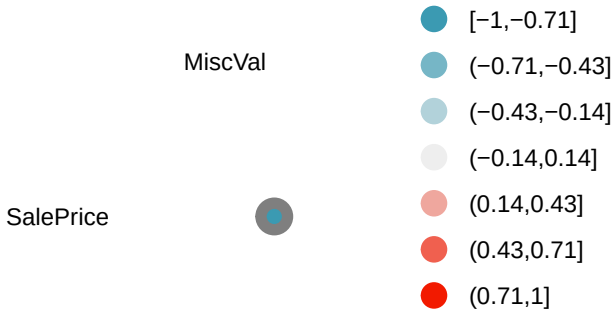

EnclosedPorch
X3SsnPorch
ScreenPorch
SalePrice
OpenPorchSF
−1
0
1

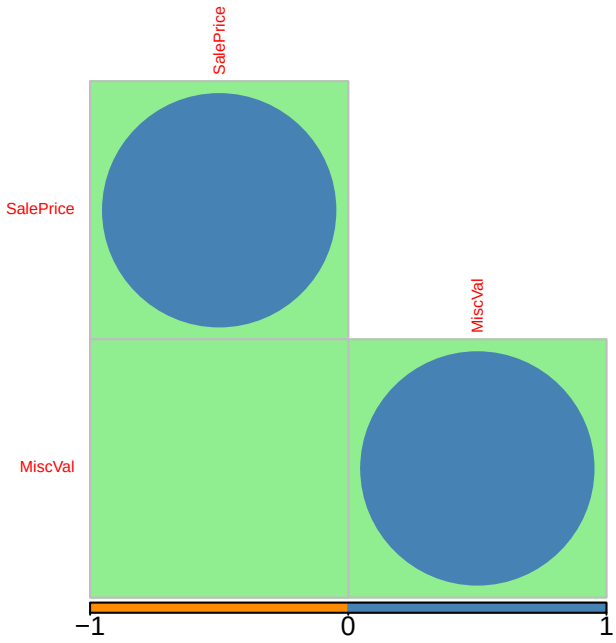
Edificio−Piscina

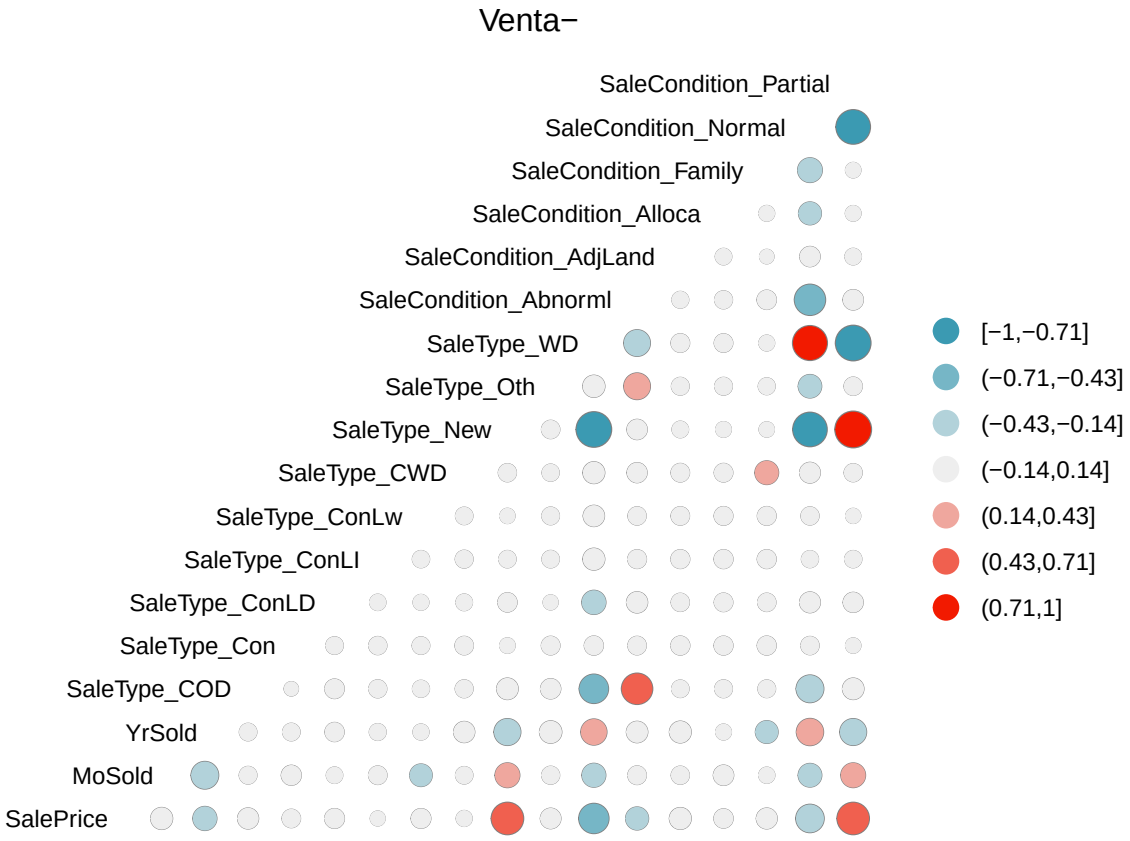


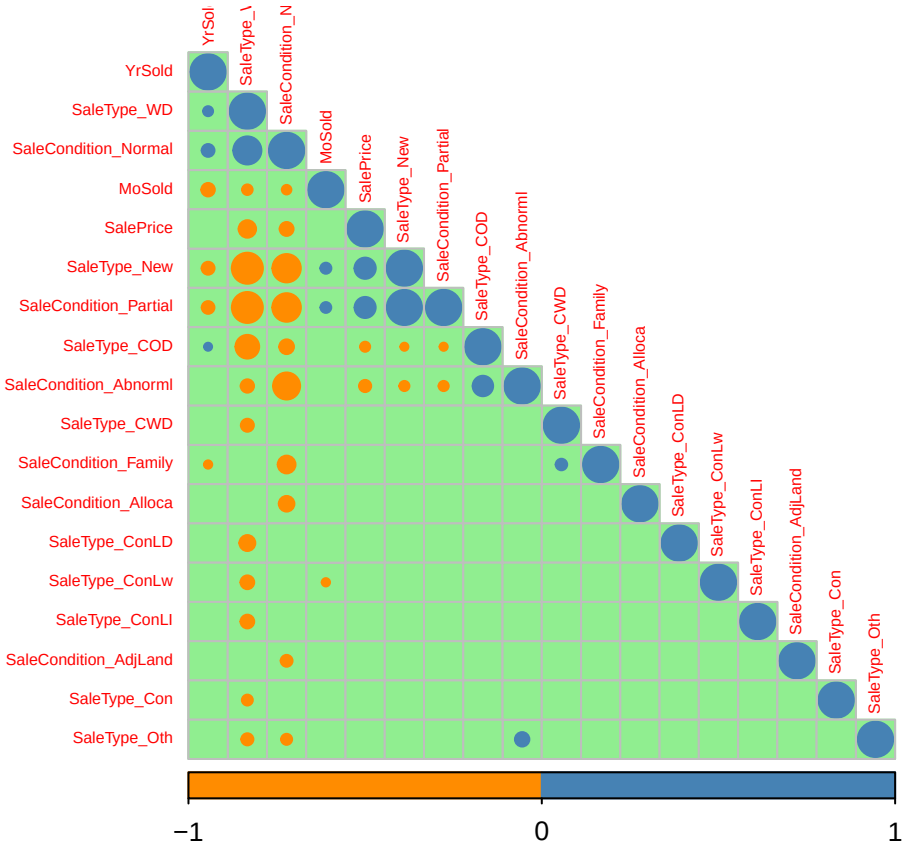

SalePrice
SalePrice
PoolArea
PoolArea
−1
0
1

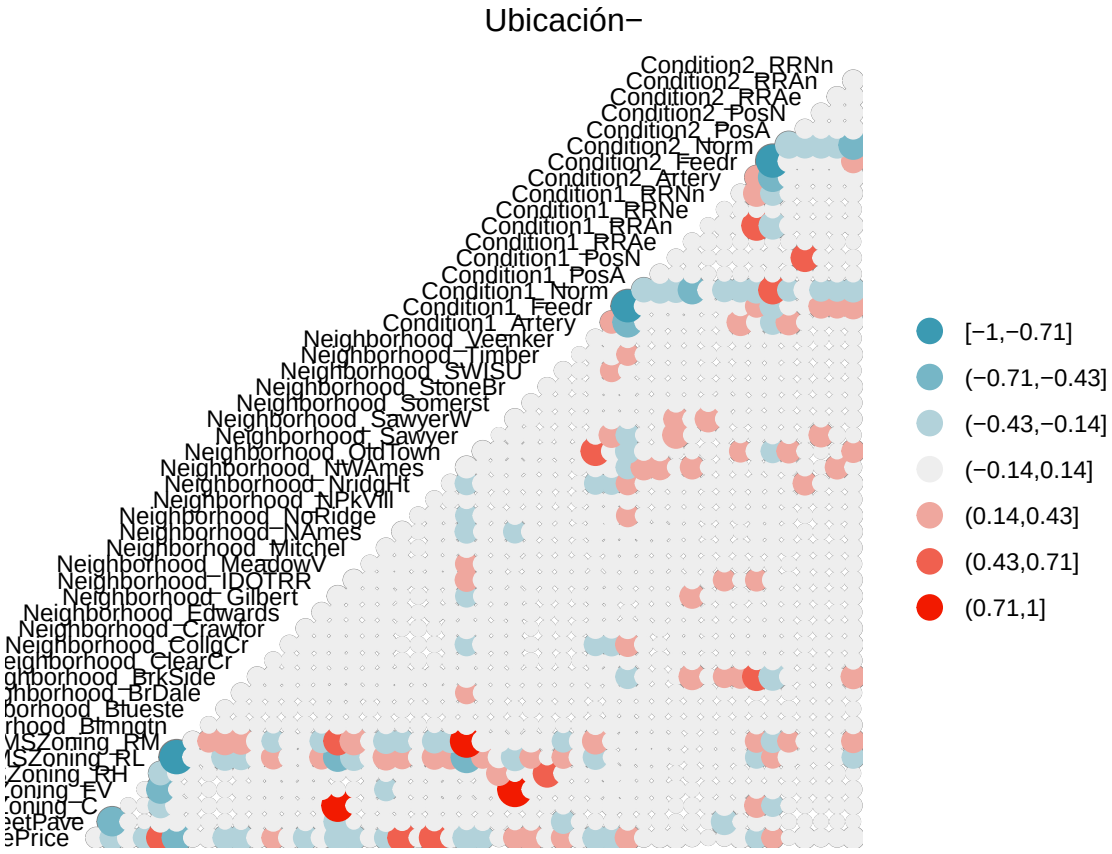
Otras caracteristicas−

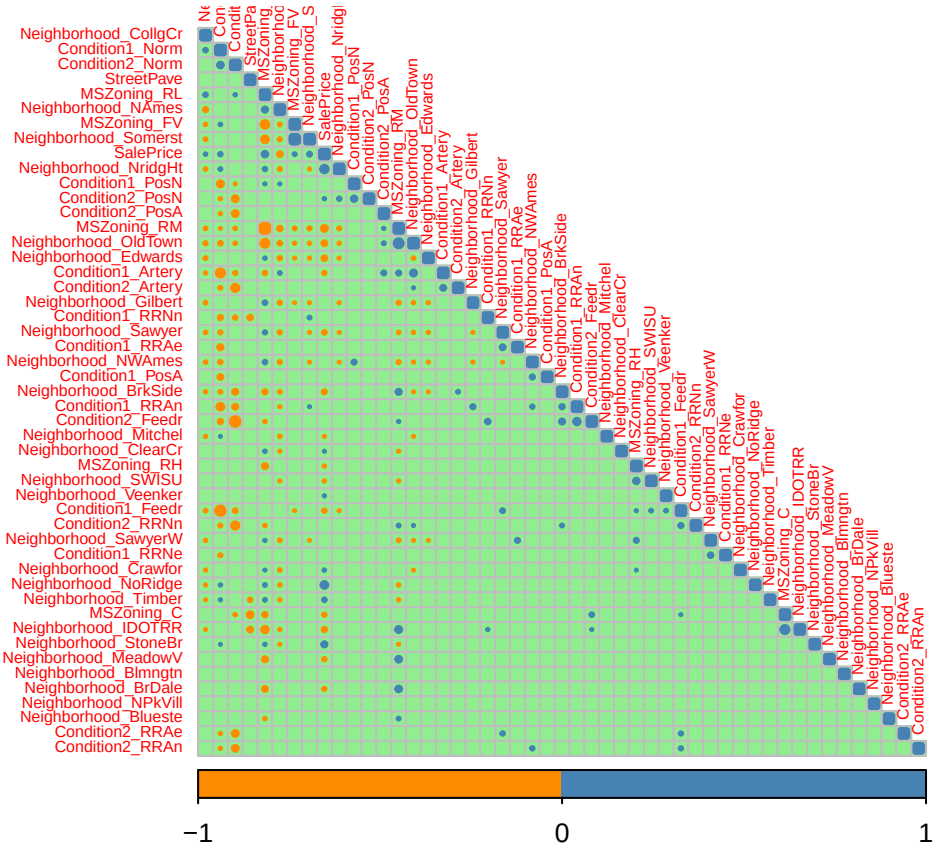


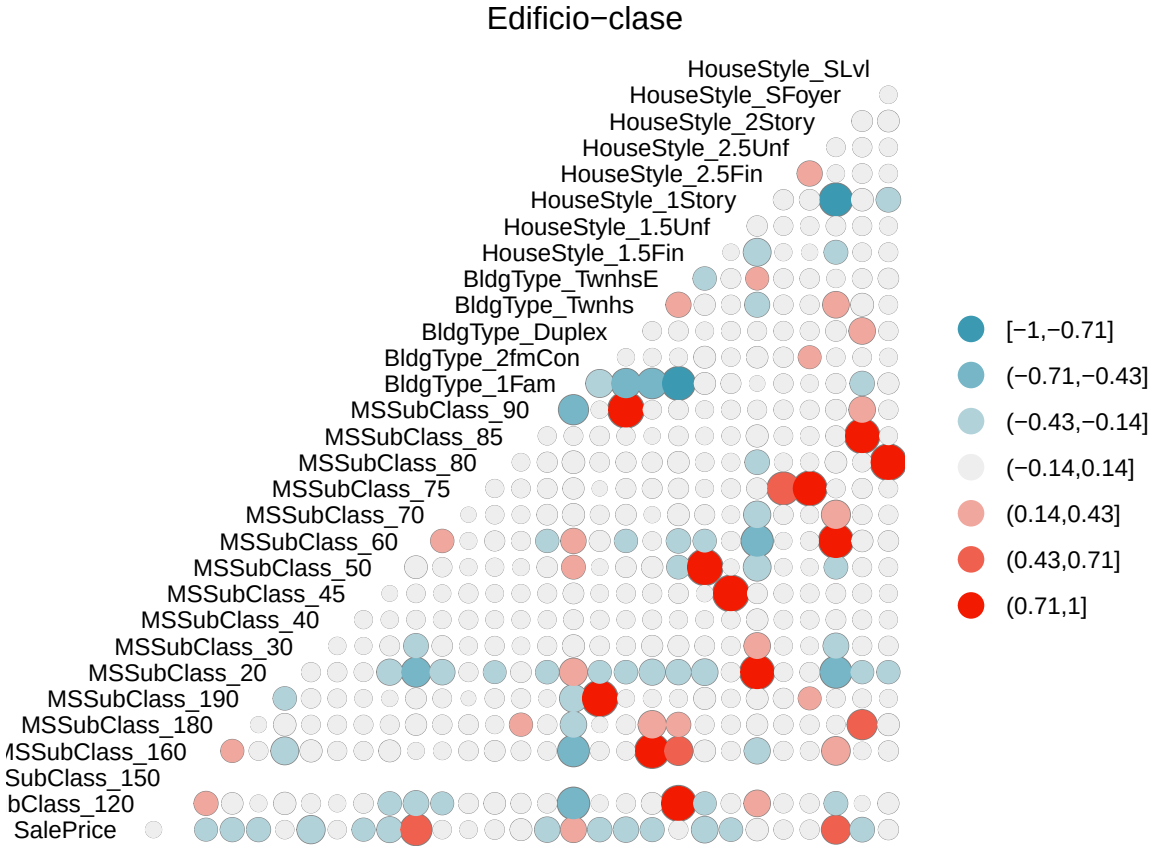

SalePrice
SalePrice
MiscVal
MiscVal
−1
0
1

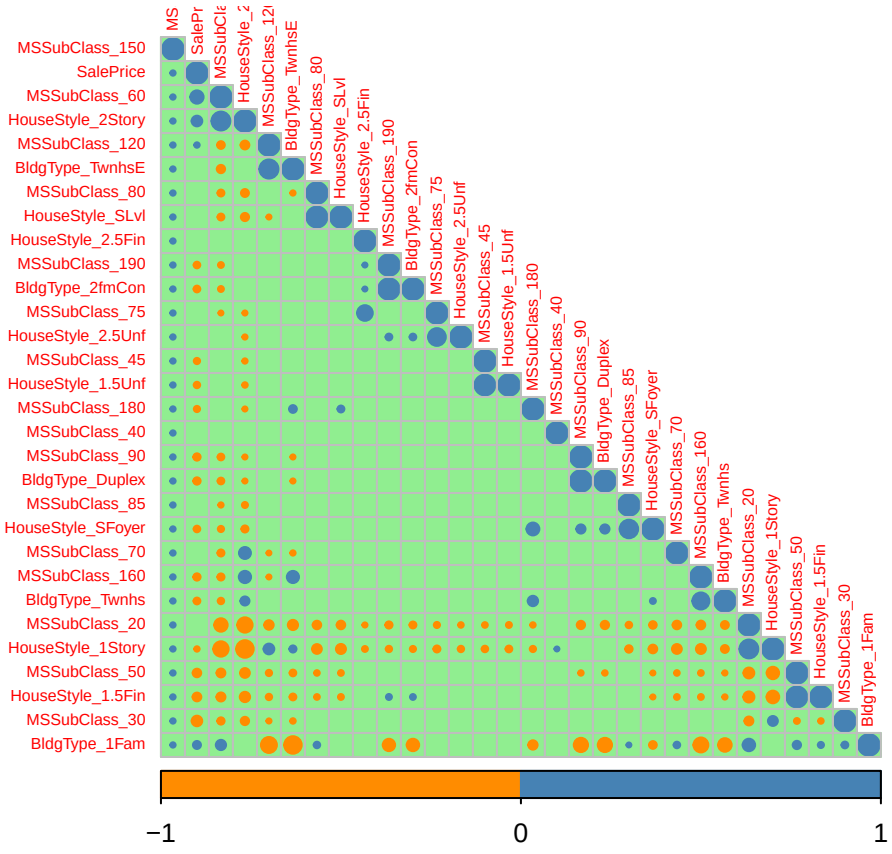

Venta−
SaleCondition_Partial
SaleCondition_Normal
SaleCondition_Family
SaleCondition_Alloca
SaleCondition_AdjLand
SaleCondition_Abnorml
SaleType_WD
SaleType_Oth
SaleType_New
SaleType_CWD
SaleType_ConLw
SaleType_ConLI
SaleType_ConLD
SaleType_Con
SaleType_COD
YrSold
MoSold
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]

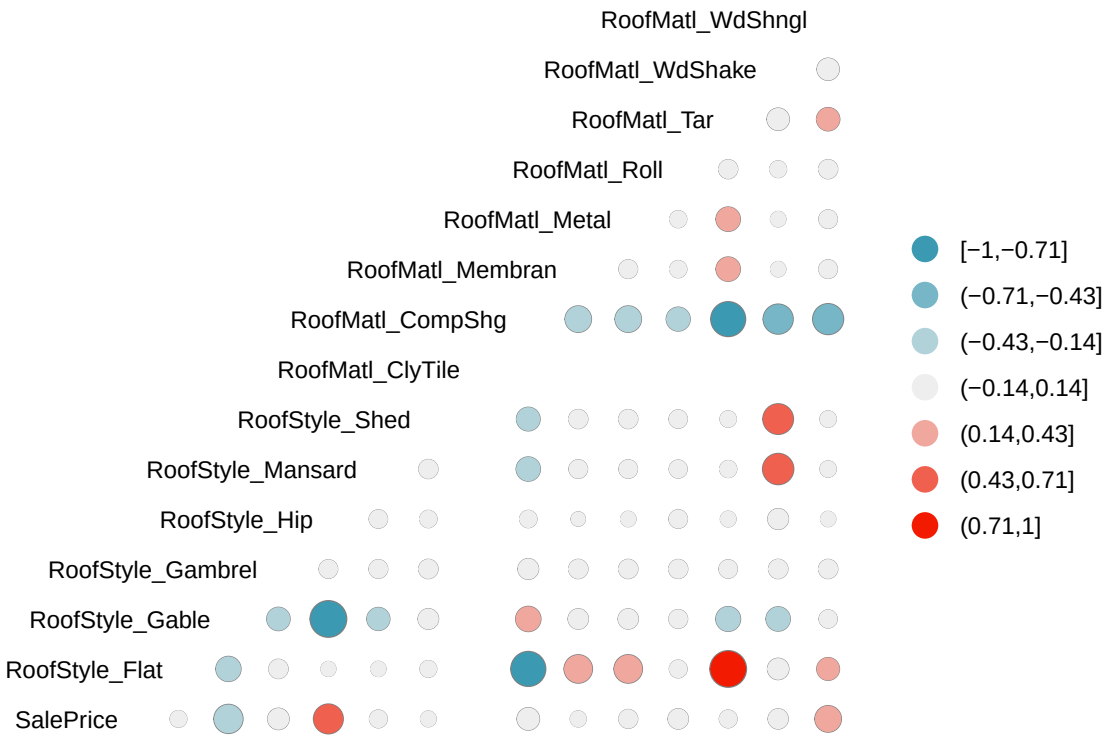

YrSold
SaleType_WD
SaleCondition_Normal
MoSold
SalePrice
SaleType_New
SaleCondition_Partial
SaleType_COD
SaleCondition_Abnorml
SaleType_CWD
SaleCondition_Family
SaleCondition_Alloca
SaleType_ConLD
SaleType_ConLw
SaleType_ConLI
SaleCondition_AdjLand
SaleType_Con
SaleType_Oth
−1
0
1

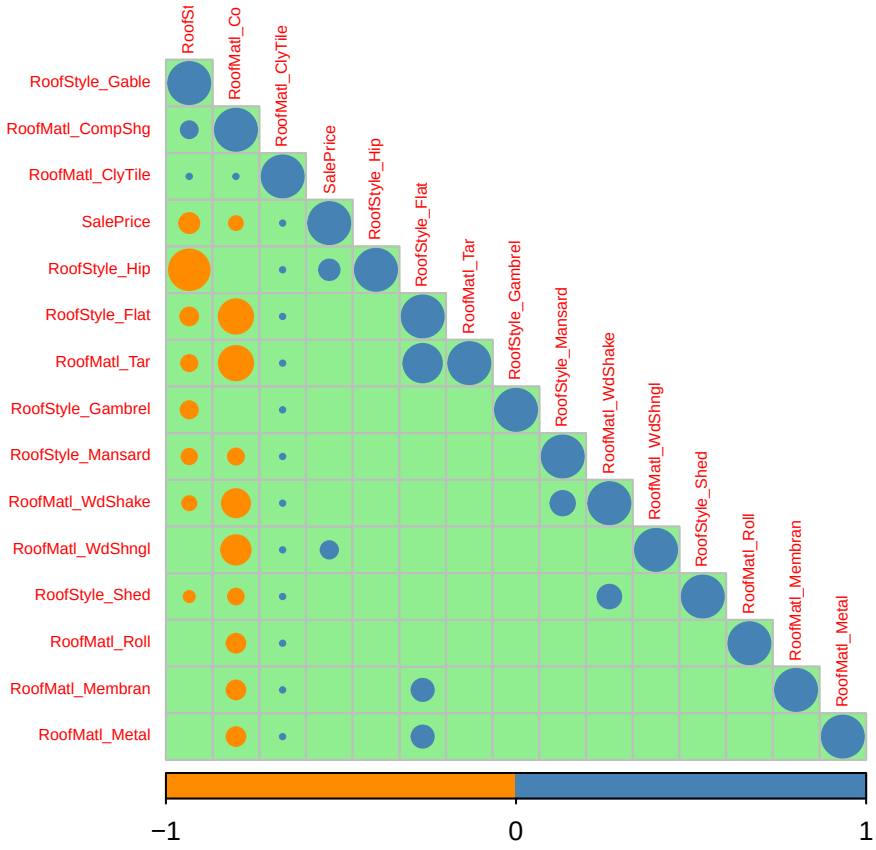

Ubicación−
Condition2_RRNn
Condition2_RRAn
Condition2_RRAe
Condition2_PosN
Condition2_PosA
Condition2_Norm
Condition2_Feedr
Condition2_Artery
Condition1_RRNn
Condition1_RRNe
Condition1_RRAn
Condition1_RRAe
Condition1_PosN
Condition1_PosA
Condition1_Norm
Condition1_Feedr
Condition1_Artery
Neighborhood_Veenker
Neighborhood_Timber
Neighborhood_SWISU
Neighborhood_StoneBr
Neighborhood_Somerst
Neighborhood_SawyerW
Neighborhood_Sawyer
Neighborhood_OldTown
Neighborhood_NWAmes
Neighborhood_NridgHt
Neighborhood_NPkVill
Neighborhood_NoRidge
Neighborhood_NAmes
Neighborhood_Mitchel
Neighborhood_MeadowV
Neighborhood_IDOTRR
Neighborhood_Gilbert
Neighborhood_Edwards
Neighborhood_Crawfor
Neighborhood_CollgCr
eighborhood_ClearCr
ghborhood_BrkSide
nborhood_BrDale
borhood_Blueste
rhood_Blmngtn
MSZoning_RM
tSZoning_RL
Zoning_RH
oning_FV
oning_C
etPave
ePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]


Neighborhood_CollgCr
Condition1_Norm
Condition2_Norm
StreetPave
MSZoning_RL
Neighborhood_NAmes
MSZoning_FV
Neighborhood_Somerst
SalePrice
Neighborhood_NridgHt
Condition1_PosN
Condition2_PosN
Condition2_PosA
MSZoning_RM
Neighborhood_OldTown
Neighborhood_Edwards
Condition1_Artery
Condition2_Artery
Neighborhood_Gilbert
Condition1_RRNn
Neighborhood_Sawyer
Condition1_RRAe
Neighborhood_NWAmes
Condition1_PosA
Neighborhood_BrkSide
Condition1_RRAn
Condition2_Feedr
Neighborhood_Mitchel
Neighborhood_ClearCr
MSZoning_RH
Neighborhood_SWISU
Neighborhood_Veenker
Condition1_Feedr
Condition2_RRNn
Neighborhood_SawyerW
Condition1_RRNe
Neighborhood_Crawfor
Neighborhood_NoRidge
Neighborhood_Timber
MSZoning_C
Neighborhood_IDOTRR
Neighborhood_StoneBr
Neighborhood_MeadowV
Neighborhood_Blmngtn
Neighborhood_BrDale
Neighborhood_NPkVill
Neighborhood_Blueste
Condition2_RRAe
Condition2_RRAn
−1
0
1

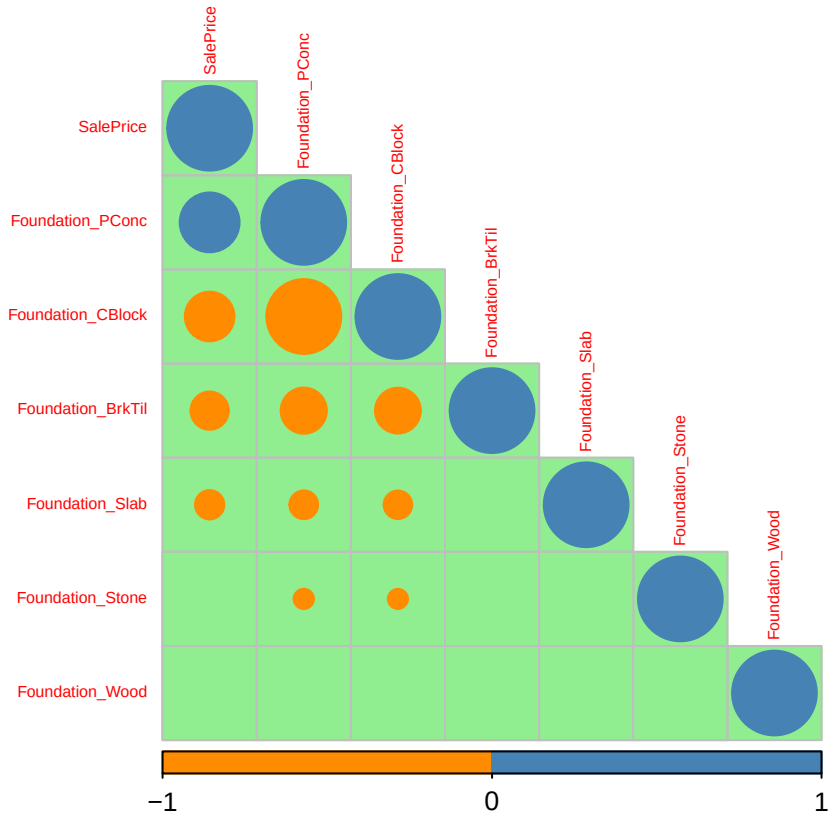

Edificio−clase
HouseStyle_SLvl
HouseStyle_SFoyer
HouseStyle_2Story
HouseStyle_2.5Unf
HouseStyle_2.5Fin
HouseStyle_1Story
HouseStyle_1.5Unf
HouseStyle_1.5Fin
BldgType_TwnhsE
BldgType_Twnhs
BldgType_Duplex
BldgType_2fmCon
BldgType_1Fam
MSSubClass_90
MSSubClass_85
MSSubClass_80
MSSubClass_75
MSSubClass_70
MSSubClass_60
MSSubClass_50
MSSubClass_45
MSSubClass_40
MSSubClass_30
MSSubClass_20
MSSubClass_190
MSSubClass_180
ISSubClass_160
SubClass_150
bClass_120
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]


MSSubClass_150
SalePrice
MSSubClass_60
HouseStyle_2Story
MSSubClass_120
BldgType_TwnhsE
MSSubClass_80
HouseStyle_SLvl
HouseStyle_2.5Fin
MSSubClass_190
BldgType_2fmCon
MSSubClass_75
HouseStyle_2.5Unf
MSSubClass_45
HouseStyle_1.5Unf
MSSubClass_180
MSSubClass_40
MSSubClass_90
BldgType_Duplex
MSSubClass_85
HouseStyle_SFoyer
MSSubClass_70
MSSubClass_160
BldgType_Twnhs
MSSubClass_20
HouseStyle_1Story
MSSubClass_50
HouseStyle_1.5Fin
MSSubClass_30
BldgType_1Fam
−1
0
1


Edificio−Techo
RoofMatl_WdShngl
RoofMatl_WdShake
RoofMatl_Tar
RoofMatl_Roll
RoofMatl_Metal
RoofMatl_Membran
RoofMatl_CompShg
RoofMatl_ClyTile
RoofStyle_Shed
RoofStyle_Mansard
RoofStyle_Hip
RoofStyle_Gambrel
RoofStyle_Gable
RoofStyle_Flat
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]


RoofSt
RoofMatl_Co
RoofMatl_ClyTile
SalePrice
RoofStyle_Hip
RoofStyle_Flat
RoofMatl_Tar
RoofStyle_Gambrel
RoofStyle_Mansard
RoofMatl_WdShake
RoofMatl_WdShngl
RoofStyle_Shed
RoofMatl_Roll
RoofMatl_Membran
RoofMatl_Metal
RoofStyle_Gable
RoofMatl_CompShg
RoofMatl_ClyTile
SalePrice
RoofStyle_Hip
RoofStyle_Flat
RoofMatl_Tar
RoofStyle_Gambrel
RoofStyle_Mansard
RoofMatl_WdShake
RoofMatl_WdShngl
RoofStyle_Shed
RoofMatl_Roll
RoofMatl_Membran
RoofMatl_Metal
−1
0
1


Edificio−material
Foundation_Wood
Foundation_Stone
Foundation_Slab
Foundation_PConc
Foundation_CBlock
Foundation_BrkTil
SalePrice
[−1,−0.71]
(−0.71,−0.43]
(−0.43,−0.14]
(−0.14,0.14]
(0.14,0.43]
(0.43,0.71]
(0.71,1]




```
rm(datos)
rm(datosMatriz)
rm(corGrupo)
rm(p1)
```

Relaciones entre Saleprice y el resto de variables

Calculo tabla con las variables mas correlacionadas con SalePrice

```
datosMatriz <- dsDataAll %>%
  filter(indTrain == 1) %>%
  select(-c(Id,indTrain))

#Matriz de correlación con niveles de significación para las correlaciones de Pearson y Spearman
dsPrecios.rcorr <- rcorr(as.matrix(datosMatriz), type="pearson")

# Busqueda de las variables mas correlacionadas con Precio de Venta
dsPreciosR <- data.frame(round(dsPrecios.rcorr$r, 3)) %>%
  select(corr = SalePrice) %>%
  rownames_to_column(var = "Campo") %>%
  filter(Campo!="SalePrice") %>%
  mutate(corrAbs = abs(corr)) %>%
  arrange(desc(corrAbs))
```

```
dsPPreciosP <- data.frame(round(dsPrecios.rcorr$P, 5)) %>%
  select(p = SalePrice) %>%
  rownames_to_column(var = "Campo") %>%
  arrange(p)

dsPreciosR <- dsPreciosR %>%
  inner_join(dsPPreciosP, by = "Campo")

head(dsPreciosR,20)
```

```
##               Campo  corr corrAbs p
## 1       OverallQual 0.796   0.796 0
## 2         GrLivArea 0.735   0.735 0
## 3         ExterQual 0.687   0.687 0
## 4       KitchenQual 0.662   0.662 0
## 5       TotalBsmtSF 0.651   0.651 0
## 6        GarageCars 0.641   0.641 0
## 7         X1stFlrSF 0.632   0.632 0
## 8        GarageArea 0.629   0.629 0
## 9          BsmtQual 0.587   0.587 0
## 10         FullBath 0.562   0.562 0
## 11     GarageFinish 0.550   0.550 0
## 12     TotRmsAbvGrd 0.538   0.538 0
## 13        YearBuilt 0.524   0.524 0
## 14      FireplaceQu 0.521   0.521 0
## 15 Foundation_PConc 0.498   0.498 0
## 16       MasVnrArea 0.486   0.486 0
## 17     YearRemodAdd 0.484   0.484 0
## 18       Fireplaces 0.470   0.470 0
## 19        HeatingQC 0.428   0.428 0
## 20       BsmtFinSF1 0.409   0.409 0
```

```
rm(datosMatriz)
rm(dsPrecios.rcorr)
rm(dsPPreciosP)
```

Relaciones entre otras variables

```
# Conjunto entero test y train, sin SalePrice
datosMatriz <- dsDataAll %>%
  select(-c(Id,indTrain,SalePrice))

#Matriz de correlación con niveles de significación para las correlaciones de Pearson y Spearman
dsOtras.rcorr <- rcorr(as.matrix(datosMatriz), type="pearson")

# Busqueda de las variables mas correlacionadas
dsOtrasR <- data.frame(round(dsOtras.rcorr$r, 3)) %>%
  rownames_to_column( var = "row") %>%
  gather(column, corr, -1) %>%
  filter(row!=column) %>%
  mutate(corrAbs = abs(corr)) %>%
```

```
  arrange(desc(corrAbs))

dsOtrasP  <- data.frame(round(dsOtras.rcorr$P, 3)) %>%
  rownames_to_column( var = "row") %>%
  gather(column, p, -1) %>%
  filter(row!=column)

dsOtrasR <- dsOtrasR %>%
  inner_join(dsOtrasP, by = c("row", "column"))

head(dsOtrasR,20)
```

```
##                      row                column   corr corrAbs p
## 1        BldgType_Duplex          MSSubClass_90  1.000   1.000 0
## 2          MSSubClass_90        BldgType_Duplex  1.000   1.000 0
## 3  SaleCondition_Partial           SaleType_New  0.986   0.986 0
## 4           SaleType_New  SaleCondition_Partial  0.986   0.986 0
## 5    Exterior2nd_CemntBd    Exterior1st_CemntBd  0.983   0.983 0
## 6    Exterior1st_CemntBd    Exterior2nd_CemntBd  0.983   0.983 0
## 7    Exterior2nd_VinylSd    Exterior1st_VinylSd  0.978   0.978 0
## 8    Exterior1st_VinylSd    Exterior2nd_VinylSd  0.978   0.978 0
## 9        BldgType_2fmCon         MSSubClass_190  0.975   0.975 0
## 10        MSSubClass_190        BldgType_2fmCon  0.975   0.975 0
## 11   Exterior2nd_MetalSd    Exterior1st_MetalSd  0.970   0.970 0
## 12   Exterior1st_MetalSd    Exterior2nd_MetalSd  0.970   0.970 0
## 13       HouseStyle_SLvl          MSSubClass_80  0.958   0.958 0
## 14         MSSubClass_80        HouseStyle_SLvl  0.958   0.958 0
## 15            GarageCond             GarageQual  0.947   0.947 0
## 16            GarageQual             GarageCond  0.947   0.947 0
## 17       GarageType_None             GarageCond -0.940   0.940 0
## 18            GarageCond        GarageType_None -0.940   0.940 0
## 19         RoofStyle_Hip        RoofStyle_Gable -0.939   0.939 0
## 20       RoofStyle_Gable          RoofStyle_Hip -0.939   0.939 0
```

```
rm(datosMatriz)
rm(dsOtras.rcorr)
rm(dsOtrasP)
```

```
rm(dsCamposActuales)
rm(dsPreciosR)
rm(dsOtrasR)
```