Kaggle House Prices

Javier Guzmán Figueira Domínguez

04/03/2018

Inspección de los datos

```
updatePartitions <- function() {
    train.size <- nrow(train)
    full.set.size <- nrow(full.set)

    train <<- full.set[c(1:train.size), ]
    test <<- full.set[c((train.size + 1):full.set.size), ]
}
```

Las dimensiones del conjunto de entrenamiento son las siguientes:

```
dim(train)
```

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

```
dim(test)
```

```
## [1] 1459   80
```

```
dim(full.set)
```

```
## [1] 2919   81
```

A continuacin, procedemos a examinar las variables del dataset:

```
str(full.set)
```

```
## 'data.frame':    2919 obs. of  81 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning     : Factor w/ 5 levels "C (all)","FV",..: 4 4 4 4 4 4 4 4 5 4 ...
##  $ LotFrontage  : int  65 80 68 60 84 85 75 NA 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
##  $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Alley        : Factor w/ 2 levels "Grvl","Pave": NA NA NA NA NA NA NA NA NA NA ...
##  $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",..: 4 4 1 1 1 1 4 1 4 4 ...
##  $ LandContour  : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Utilities    : Factor w/ 2 levels "AllPub","NoSeWa": 1 1 1 1 1 1 1 1 1 1 ...
##  $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5 1 5 1 ...
##  $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",..: 6 25 6 7 14 12 21 17 18 4 ...
##  $ Condition1   : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 5 1 1 ...
##  $ Condition2   : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 1 ...
##  $ BldgType     : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 1 1 1 2 ...
##  $ HouseStyle   : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 6 3 6 6 6 1 3 6 1 2 ...
##  $ OverallQual  : int  7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond  : int  5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
##  $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
```

```
##  $ RoofStyle    : Factor w/ 6 levels "Flat","Gable",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ RoofMatl     : Factor w/ 8 levels "ClyTile","CompShg",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Exterior1st  : Factor w/ 15 levels "AsbShng","AsphShn",..: 13 9 13 14 13 13 13 7 4 9 ...
##  $ Exterior2nd  : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 16 14 14 14 7 16 9 ...
##  $ MasVnrType   : Factor w/ 4 levels "BrkCmn","BrkFace",..: 2 3 2 3 2 3 4 4 3 3 ...
##  $ MasVnrArea   : int  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 4 3 4 3 4 4 4 ...
##  $ ExterCond    : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ Foundation   : Factor w/ 6 levels "BrkTil","CBlock",..: 3 2 3 1 3 6 3 2 1 1 ...
##  $ BsmtQual     : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 3 3 4 3 3 1 3 4 4 ...
##  $ BsmtCond     : Factor w/ 4 levels "Fa","Gd","Po",..: 4 4 4 2 4 4 4 4 4 4 ...
##  $ BsmtExposure : Factor w/ 4 levels "Av","Gd","Mn",..: 4 2 3 4 1 4 1 3 4 4 ...
##  $ BsmtFinType1 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 3 1 3 1 3 3 3 1 6 3 ...
##  $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
##  $ BsmtFinType2 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 6 6 6 6 6 6 6 2 6 6 ...
##  $ BsmtFinSF2   : int  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : int  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
##  $ Heating      : Factor w/ 6 levels "Floor","GasA",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ HeatingQC    : Factor w/ 5 levels "Ex","Fa","Gd",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ CentralAir   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical   : Factor w/ 5 levels "FuseA","FuseF",..: 5 5 5 5 5 5 5 5 2 5 ...
##  $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
##  $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
##  $ BsmtFullBath : int  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath     : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual  : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 3 3 4 3 4 4 4 ...
##  $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : Factor w/ 7 levels "Maj1","Maj2",..: 7 7 7 7 7 7 7 7 3 7 ...
##  $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu  : Factor w/ 5 levels "Ex","Fa","Gd",..: NA 5 5 3 5 NA 3 5 5 5 ...
##  $ GarageType   : Factor w/ 6 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2 6 2 ...
##  $ GarageYrBlt  : int  2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...
##  $ GarageFinish : Factor w/ 3 levels "Fin","RFn","Unf": 2 2 2 3 2 3 2 2 3 2 ...
##  $ GarageCars   : int  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : int  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 2 3 ...
##  $ GarageCond   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ PavedDrive   : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
##  $ WoodDeckSF   : int  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF  : int  61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch: int  0 0 0 272 0 0 0 228 205 0 ...
##  $ X3SsnPorch   : int  0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : Factor w/ 3 levels "Ex","Fa","Gd": NA NA NA NA NA NA NA NA NA NA ...
##  $ Fence        : Factor w/ 4 levels "GdPrv","GdWo",..: NA NA NA NA NA 3 NA NA NA NA ...
##  $ MiscFeature  : Factor w/ 4 levels "Gar2","Othr",..: NA NA NA NA NA 3 NA 3 NA NA ...
```

```
##  $ MiscVal      : int  0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold       : int  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold       : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
##  $ SaleType     : Factor w/ 9 levels "COD","Con","ConLD",..: 9 9 9 9 9 9 9 9 9 9 ...
##  $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 1 5 5 5 5 1 5 ...
##  $ SalePrice    : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
```

Y observamos el inicio:

```
head(full.set)
```

```
##   Id MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape
## 1  1         60       RL          65    8450   Pave  <NA>      Reg
## 2  2         20       RL          80    9600   Pave  <NA>      Reg
## 3  3         60       RL          68   11250   Pave  <NA>      IR1
## 4  4         70       RL          60    9550   Pave  <NA>      IR1
## 5  5         60       RL          84   14260   Pave  <NA>      IR1
## 6  6         50       RL          85   14115   Pave  <NA>      IR1
##   LandContour Utilities LotConfig LandSlope Neighborhood Condition1
## 1         Lvl    AllPub    Inside       Gtl      CollgCr       Norm
## 2         Lvl    AllPub       FR2       Gtl      Veenker      Feedr
## 3         Lvl    AllPub    Inside       Gtl      CollgCr       Norm
## 4         Lvl    AllPub    Corner       Gtl      Crawfor       Norm
## 5         Lvl    AllPub       FR2       Gtl      NoRidge       Norm
## 6         Lvl    AllPub    Inside       Gtl      Mitchel       Norm
##   Condition2 BldgType HouseStyle OverallQual OverallCond YearBuilt
## 1       Norm     1Fam     2Story           7           5      2003
## 2       Norm     1Fam     1Story           6           8      1976
## 3       Norm     1Fam     2Story           7           5      2001
## 4       Norm     1Fam     2Story           7           5      1915
## 5       Norm     1Fam     2Story           8           5      2000
## 6       Norm     1Fam     1.5Fin           5           5      1993
##   YearRemodAdd RoofStyle RoofMatl Exterior1st Exterior2nd MasVnrType
## 1         2003     Gable  CompShg     VinylSd     VinylSd    BrkFace
## 2         1976     Gable  CompShg     MetalSd     MetalSd       None
## 3         2002     Gable  CompShg     VinylSd     VinylSd    BrkFace
## 4         1970     Gable  CompShg     Wd Sdng     Wd Shng       None
## 5         2000     Gable  CompShg     VinylSd     VinylSd    BrkFace
## 6         1995     Gable  CompShg     VinylSd     VinylSd       None
##   MasVnrArea ExterQual ExterCond Foundation BsmtQual BsmtCond BsmtExposure
## 1        196        Gd        TA      PConc       Gd       TA           No
## 2          0        TA        TA     CBlock       Gd       TA           Gd
## 3        162        Gd        TA      PConc       Gd       TA           Mn
## 4          0        TA        TA     BrkTil       TA       Gd           No
## 5        350        Gd        TA      PConc       Gd       TA           Av
## 6          0        TA        TA       Wood       Gd       TA           No
##   BsmtFinType1 BsmtFinSF1 BsmtFinType2 BsmtFinSF2 BsmtUnfSF TotalBsmtSF
## 1          GLQ        706          Unf          0       150         856
## 2          ALQ        978          Unf          0       284        1262
## 3          GLQ        486          Unf          0       434         920
## 4          ALQ        216          Unf          0       540         756
## 5          GLQ        655          Unf          0       490        1145
## 6          GLQ        732          Unf          0        64         796
##   Heating HeatingQC CentralAir Electrical X1stFlrSF X2ndFlrSF LowQualFinSF
## 1    GasA        Ex          Y      SBrkr       856       854            0
```

```
## 2    GasA        Ex          Y      SBrkr      1262         0            0
## 3    GasA        Ex          Y      SBrkr       920       866            0
## 4    GasA        Gd          Y      SBrkr       961       756            0
## 5    GasA        Ex          Y      SBrkr      1145      1053            0
## 6    GasA        Ex          Y      SBrkr       796       566            0
##   GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr
## 1      1710            1            0        2        1            3
## 2      1262            0            1        2        0            3
## 3      1786            1            0        2        1            3
## 4      1717            1            0        1        0            3
## 5      2198            1            0        2        1            4
## 6      1362            1            0        1        1            1
##   KitchenAbvGr KitchenQual TotRmsAbvGrd Functional Fireplaces FireplaceQu
## 1            1          Gd            8        Typ          0        <NA>
## 2            1          TA            6        Typ          1          TA
## 3            1          Gd            6        Typ          1          TA
## 4            1          Gd            7        Typ          1          Gd
## 5            1          Gd            9        Typ          1          TA
## 6            1          TA            5        Typ          0        <NA>
##   GarageType GarageYrBlt GarageFinish GarageCars GarageArea GarageQual
## 1     Attchd        2003          RFn          2        548         TA
## 2     Attchd        1976          RFn          2        460         TA
## 3     Attchd        2001          RFn          2        608         TA
## 4     Detchd        1998          Unf          3        642         TA
## 5     Attchd        2000          RFn          3        836         TA
## 6     Attchd        1993          Unf          2        480         TA
##   GarageCond PavedDrive WoodDeckSF OpenPorchSF EnclosedPorch X3SsnPorch
## 1         TA          Y          0          61             0          0
## 2         TA          Y        298           0             0          0
## 3         TA          Y          0          42             0          0
## 4         TA          Y          0          35           272          0
## 5         TA          Y        192          84             0          0
## 6         TA          Y         40          30             0        320
##   ScreenPorch PoolArea PoolQC Fence MiscFeature MiscVal MoSold YrSold
## 1           0        0   <NA>  <NA>        <NA>       0      2   2008
## 2           0        0   <NA>  <NA>        <NA>       0      5   2007
## 3           0        0   <NA>  <NA>        <NA>       0      9   2008
## 4           0        0   <NA>  <NA>        <NA>       0      2   2006
## 5           0        0   <NA>  <NA>        <NA>       0     12   2008
## 6           0        0   <NA> MnPrv        Shed     700     10   2009
##   SaleType SaleCondition SalePrice
## 1       WD        Normal    208500
## 2       WD        Normal    181500
## 3       WD        Normal    223500
## 4       WD       Abnorml    140000
## 5       WD        Normal    250000
## 6       WD        Normal    143000
```

- Análisis de valores perdidos

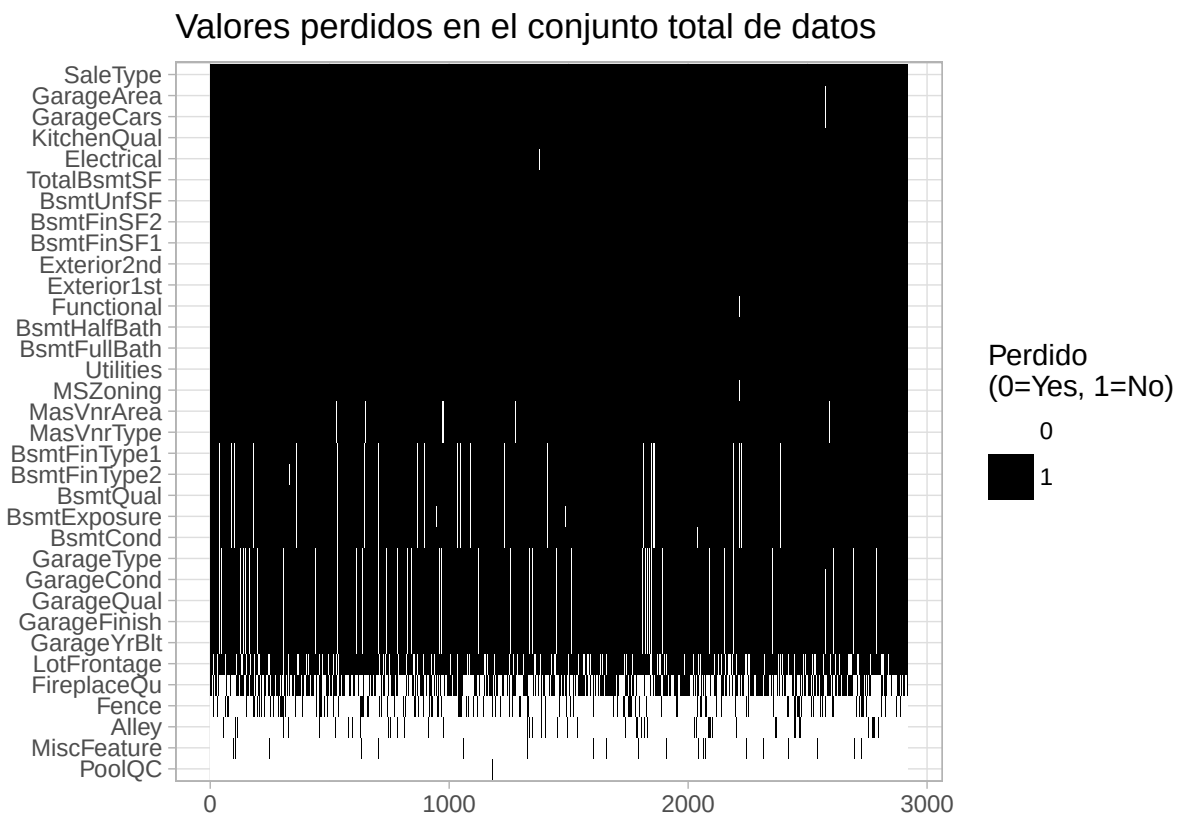
```
getLostValuesStats <- function() {
    lost.count <- colSums(sapply(select(full.set, -SalePrice),
        is.na))
    lost.count <- subset(lost.count, lost.count > 0)
    lost.percentage <- (lost.count/nrow(full.set)) * 100
```

```
    return(data.frame(lost.count, lost.percentage))
}

getLostValuesStats()
```

```
##              lost.count lost.percentage
## MSZoning              4      0.13703323
## LotFrontage         486     16.64953751
## Alley              2721     93.21685509
## Utilities             2      0.06851662
## Exterior1st           1      0.03425831
## Exterior2nd           1      0.03425831
## MasVnrType           24      0.82219938
## MasVnrArea           23      0.78794108
## BsmtQual             81      2.77492292
## BsmtCond             82      2.80918123
## BsmtExposure         82      2.80918123
## BsmtFinType1         79      2.70640630
## BsmtFinSF1            1      0.03425831
## BsmtFinType2         80      2.74066461
## BsmtFinSF2            1      0.03425831
## BsmtUnfSF             1      0.03425831
## TotalBsmtSF           1      0.03425831
## Electrical            1      0.03425831
## BsmtFullBath          2      0.06851662
## BsmtHalfBath          2      0.06851662
## KitchenQual           1      0.03425831
## Functional            2      0.06851662
## FireplaceQu        1420     48.64679685
## GarageType          157      5.37855430
## GarageYrBlt         159      5.44707091
## GarageFinish        159      5.44707091
## GarageCars            1      0.03425831
## GarageArea            1      0.03425831
## GarageQual          159      5.44707091
## GarageCond          159      5.44707091
## PoolQC             2909     99.65741692
## Fence              2348     80.43850634
## MiscFeature        2814     96.40287770
## SaleType              1      0.03425831
```

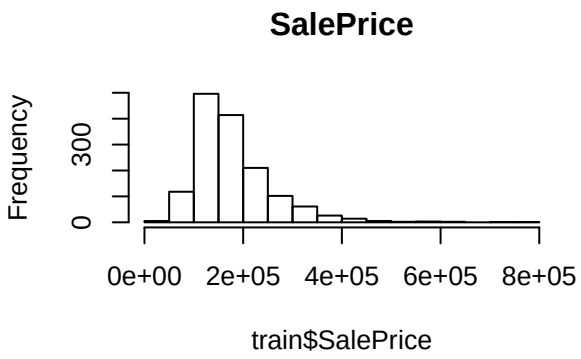
```
lost.values.count <- full.set[, colSums(is.na(select(full.set,
    -SalePrice))) > 0]

is.lost.value <- as.data.frame(ifelse(is.na(lost.values.count),
    0, 1))
is.lost.value <- is.lost.value[, order(colSums(is.lost.value))]

is.lost.value.grid <- expand.grid(list(x = 1:nrow(is.lost.value),
    y = colnames(is.lost.value)))
is.lost.value.grid$m <- as.vector(as.matrix(is.lost.value))
is.lost.value.grid <- data.frame(x = unlist(is.lost.value.grid$x),
    y = unlist(is.lost.value.grid$y), m = unlist(is.lost.value.grid$m))
```

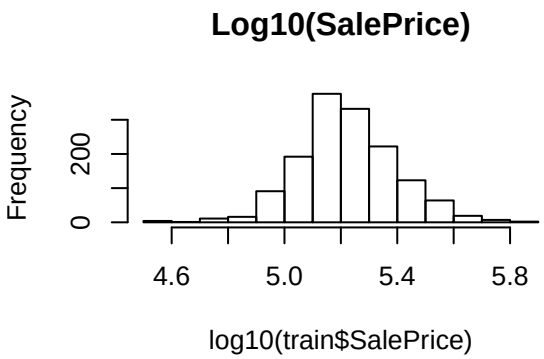
```
ggplot2::ggplot(is.lost.value.grid) + ggplot2::geom_tile(ggplot2::aes(x = x,
    y = y, fill = factor(m))) + ggplot2::scale_fill_manual(values = c("white",
    "black"), name = "Perdido\n(0=Yes, 1=No)") + ggplot2::theme_light() +
    ggplot2::ylab("") + ggplot2::xlab("") + ggplot2::ggtitle("Valores perdidos en el conjunto total de d
```

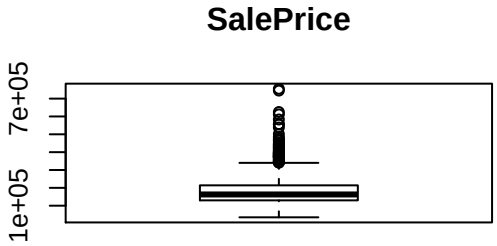


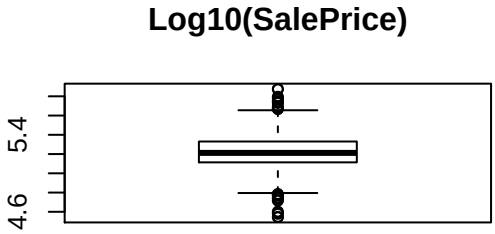
- Análisis de la distribución de la variable clase

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

hist(train$SalePrice, main = "SalePrice")
hist(log10(train$SalePrice), main = "Log10(SalePrice)")

boxplot(train$SalePrice, main = "SalePrice")
boxplot(log10(train$SalePrice), main = "Log10(SalePrice)")
```










```
par(mfrow = c(1, 1))
```

- Distribución de los valores perdidos en función de la variable Log(SalePrice)

“‘{r} # TODO: DELETE ¿? lost.values.features <- rownames(getLostValuesStats()) factor.features <- names(train)[which(sapply(train, is.factor))] lost.values.features.factor <- dplyr::intersect(factor.features, lost.values.features)

plots <- lapply(lost.values.features.factor, function(feature) { categories <- train[, feature] ggplot(data = train, aes(x = feature, y = log(SalePrice), fill = categories)) + geom_boxplot() })

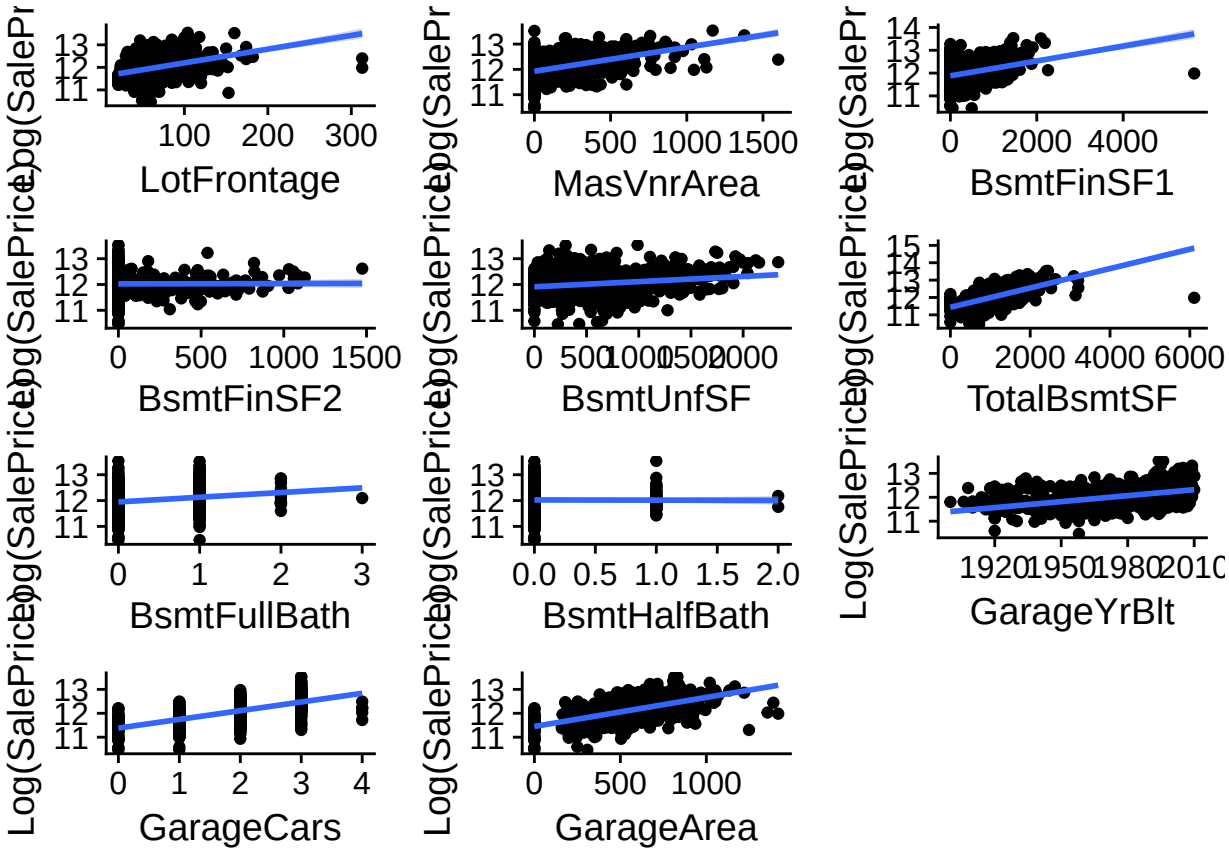
cowplot::plot_grid(plotlist = plots, ncol = 3) “‘

- Examinamos la distribución de las variables continuas con respecto a Log(SalePrice)

```
lost.values.features <- rownames(getLostValuesStats())
numeric.features <- names(train)[which(sapply(train, is.numeric))]
lost.values.features.numeric <- dplyr::intersect(numeric.features,
    lost.values.features)

plots <- lapply(lost.values.features.numeric, function(feature) {
    ggplot(data = train, aes(x = train[, feature], y = log(train$SalePrice))) +
        geom_point() + geom_smooth(method = "lm") + xlab(label = feature) +
        ylab(label = "Log(SalePrice)")
})

cowplot::plot_grid(plotlist = plots, ncol = 3)
```

```
## Warning: Removed 259 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 259 rows containing missing values (geom_point).
## Warning: Removed 8 rows containing non-finite values (stat_smooth).
## Warning: Removed 8 rows containing missing values (geom_point).
## Warning: Removed 81 rows containing non-finite values (stat_smooth).
## Warning: Removed 81 rows containing missing values (geom_point).
```



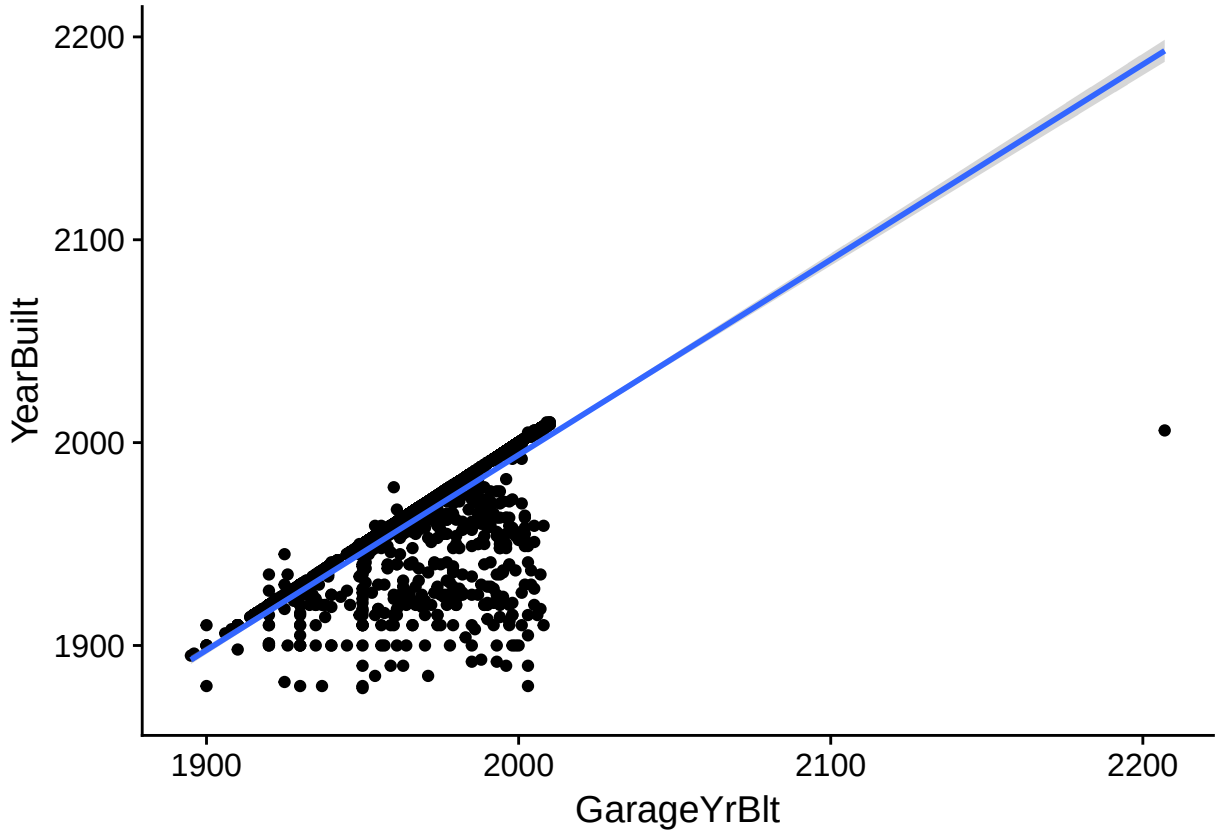
Tratamiento de los valores perdidos

- GarageYrBlt

Se aprecia que es una propidad que, lógicamente, está muy relacionada con YearBuilt (año de construcción). En general, se puede decir que GarageYrBlt tiende a ser igual a YearBuilt. Por consiguiente, en los valores perdidos de GarageYrBlt, se procede a asígnar el correspondiente valor de YearBuilt.

```
ggplot(data = full.set, aes(x = GarageYrBlt, y = YearBuilt)) +
    geom_point() + geom_smooth(method = "lm")
```

```
## Warning: Removed 159 rows containing non-finite values (stat_smooth).
## Warning: Removed 159 rows containing missing values (geom_point).
```




```
full.set$GarageYrBlt[full.set$GarageYrBlt == 2207] <- 2007
full.set$GarageYrBlt[is.na(full.set$GarageYrBlt)] <- full.set$YearBuilt[is.na(full.set$GarageYrBlt)]

updatePartitions()
```

- LotFrontage

Por lógica, se puede decir que el área de la propiedad con la longitud de la fachada. Para confirmarlo, comprobamos la correlación entre ellas:

```
cor(full.set$LotFrontage, full.set$LotArea, use = "complete.obs")
```

```
## [1] 0.4898956
```

```
cor(log(full.set$LotFrontage), log(full.set$LotArea), use = "complete.obs")
```

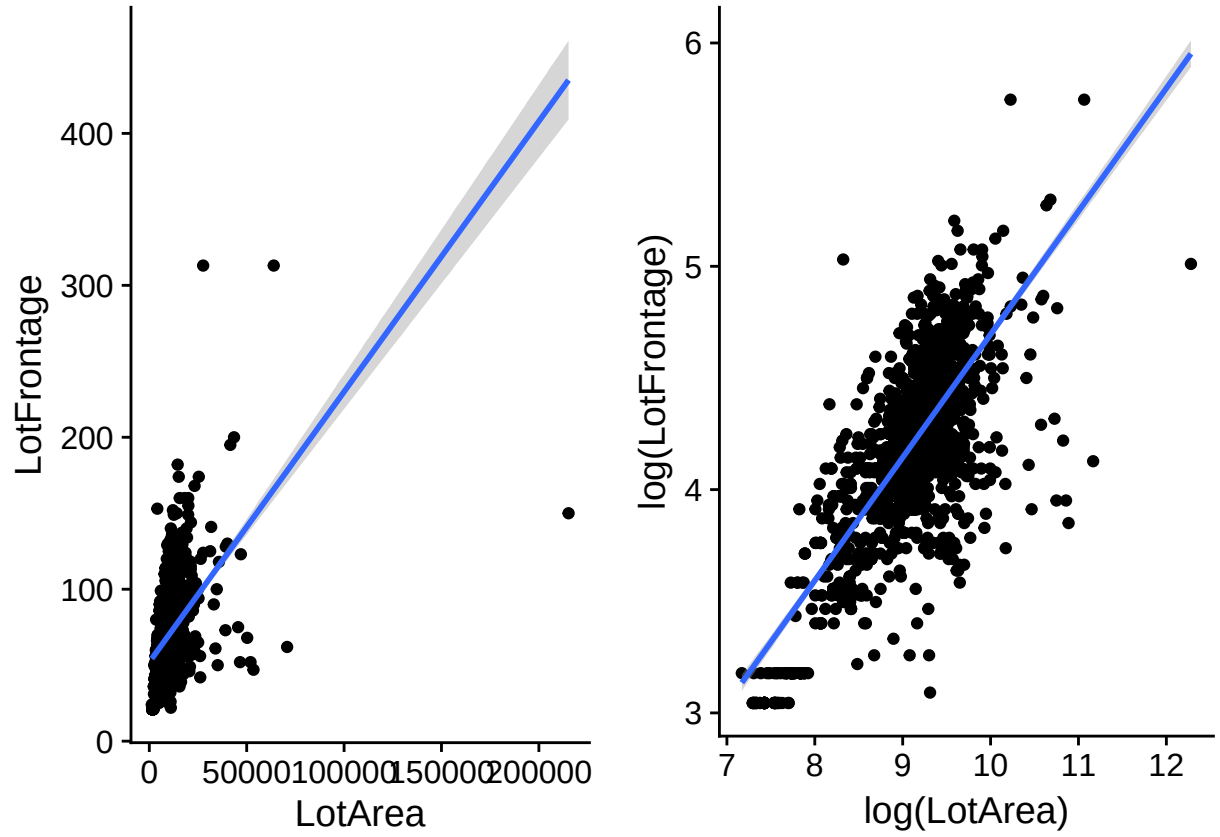
```
## [1] 0.7662858
```

Y visualizamos su relación:

```
plotLotRelation <- ggplot(data = full.set, aes(x = LotArea, y = LotFrontage)) +
    geom_point() + geom_smooth(method = "lm")

plotLogLotRelation <- ggplot(data = full.set, aes(x = log(LotArea),
    y = log(LotFrontage))) + geom_point() + geom_smooth(method = "lm")

cowplot::plot_grid(plotLotRelation, plotLogLotRelation, ncol = 2)
```

```
## Warning: Removed 486 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 486 rows containing missing values (geom_point).
## Warning: Removed 486 rows containing non-finite values (stat_smooth).
## Warning: Removed 486 rows containing missing values (geom_point).
```



Se puede confirmar que existe una alta correlación directa entre *LotFrontage* con *LotArea*. Dado, que estas dos propiedades están relacionadas, seguramente una de ellas sea desechada en el proceso de selección de variables. Idependientemente de ello, en este paso sustituiremos los valores de *LotFrontage*, por la mediana de los valores existentes.

```
full.set$LotFrontage[is.na(full.set$LotFrontage)] <- mean(full.set$LotFrontage[!is.na(full.set$LotFronta

updatePartitions()
```

```
cor(full.set$LotFrontage, full.set$LotArea, use = "complete.obs")
```

```
## [1] 0.364382
```

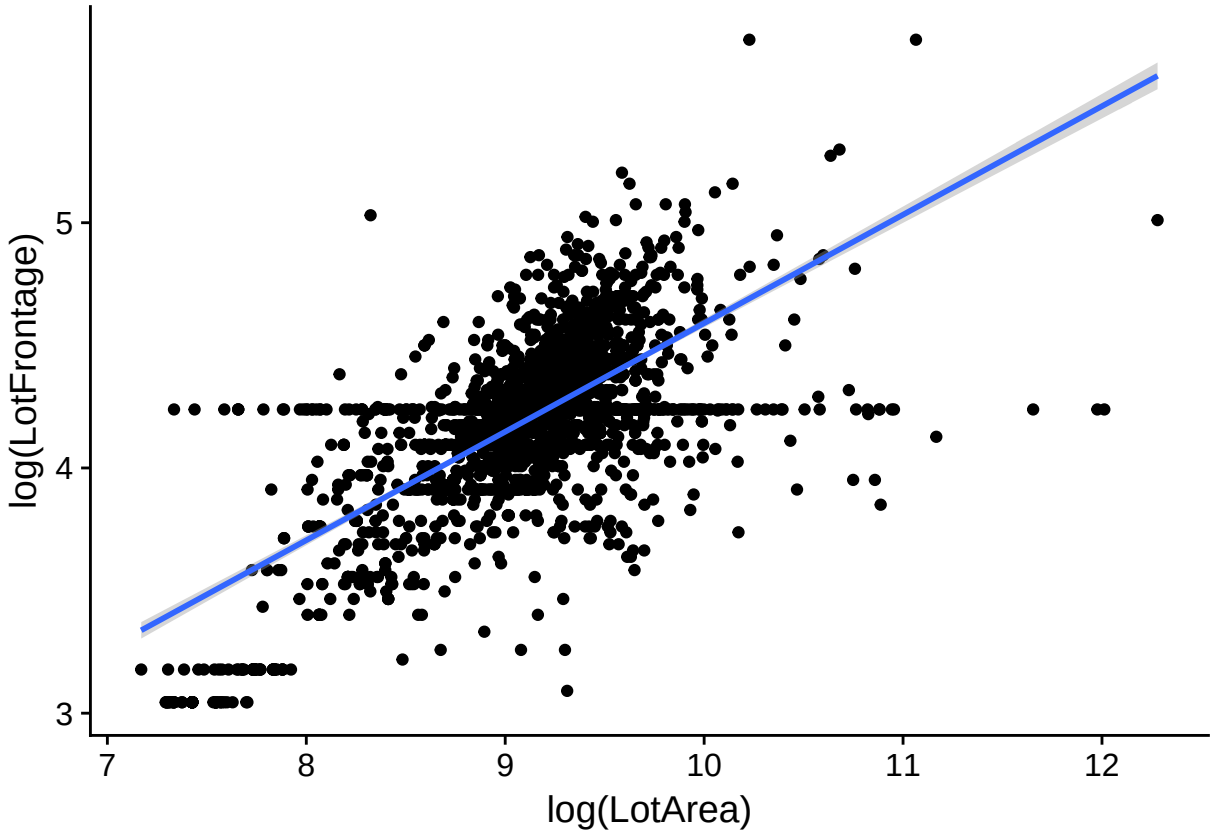
```
cor(log(full.set$LotFrontage), log(full.set$LotArea), use = "complete.obs")
```

```
## [1] 0.6894001
```

Observamos que la correlación continúa siendo similar después de tratar los valores perdidos en *LotFrontage*.

TODO: Quizás remplazar con la media no sea la mejor opción (cambia bastante la correlación). Si no se encuentra una solución mejor, quizás habría que cargarse directamente la variable.

```
ggplot(data = full.set, aes(x = log(LotArea), y = log(LotFrontage))) +
    geom_point() + geom_smooth(method = "lm")
```

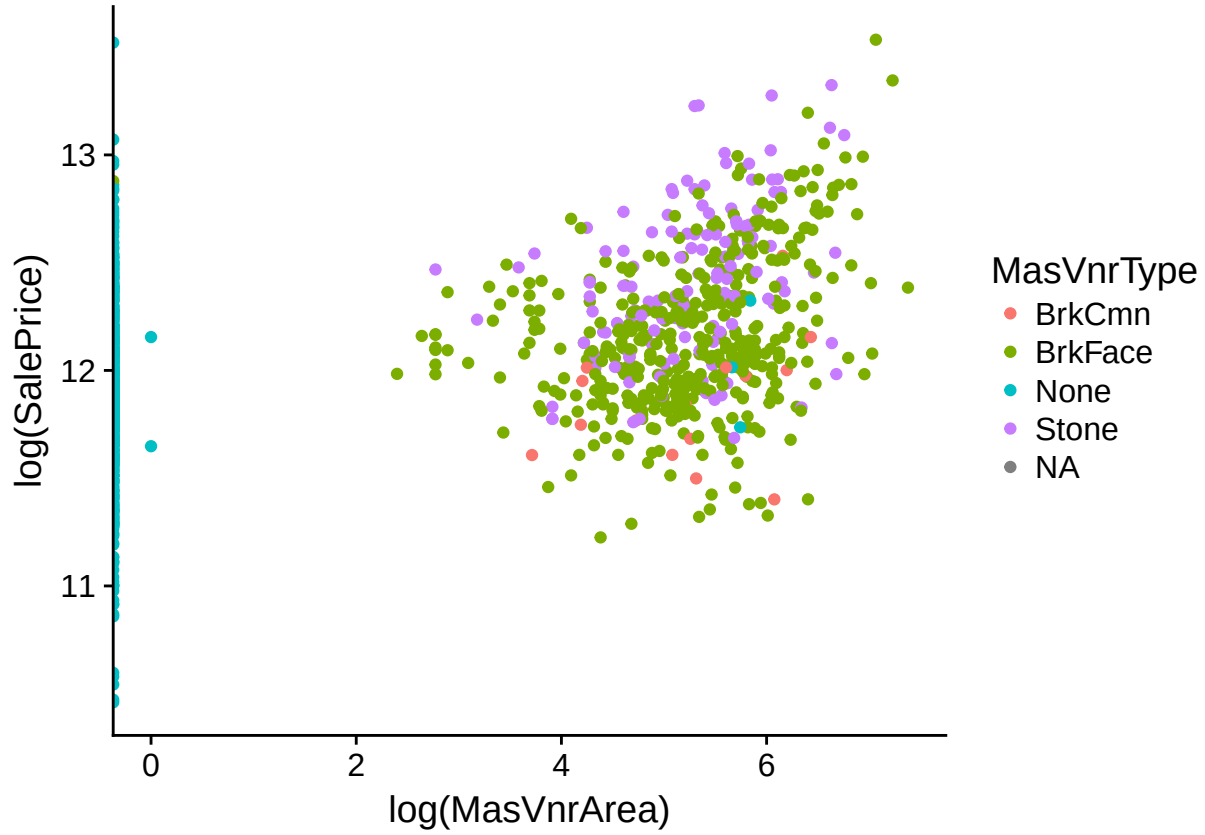


- MasVnrArea

Existe una gran cantidad de entradas con valor 0. Esto seguramente se deba a la carencia de "chapado":

```
ggplot2::qplot(data = train, x = log(MasVnrArea), y = log(SalePrice),
    col = MasVnrType)
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```



También observamos que en ambas variables los valores perdidos (8) forman parte de los mismos ejemplos:

```
full.set$Id[is.na(full.set$MasVnrArea)]
```

```
##  [1]  235  530  651  937  974  978 1244 1279 1692 1707 1883 1993 2005 2042
## [15] 2312 2326 2341 2350 2369 2593 2658 2687 2863
```

```
full.set$Id[is.na(full.set$MasVnrType)]
```

```
##  [1]  235  530  651  937  974  978 1244 1279 1692 1707 1883 1993 2005 2042
## [15] 2312 2326 2341 2350 2369 2593 2611 2658 2687 2863
```

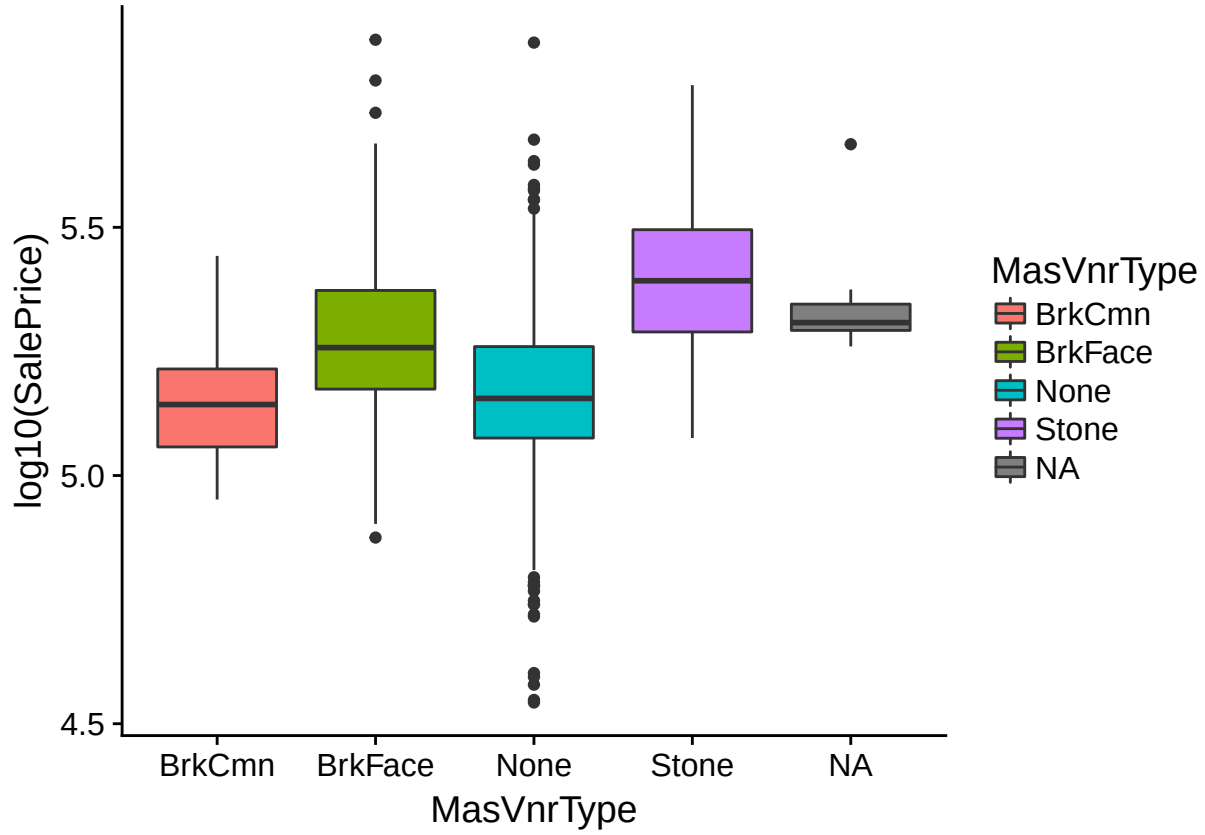
TODO: hay una entrada más en el conjunto de test

Por consiguiente, se elimina la caraterística *MasVnrArea*, ya que las entradas con valor 0, por ser del tipo "None", hacen que la información desprendida de la variable esté "deformada".

```
full.set <- dplyr::select(full.set, -MasVnrArea)

updatePartitions()
```

Ahora se deben tratar los valores perdidos de *MasVnrType*. Para ello, observamos *MasVnrType* en relación a *SalePrice* para entender su distribución.

```
qplot(data = train, x = MasVnrType, y = log10(SalePrice), geom = c("boxplot"),
    fill = MasVnrType)
```

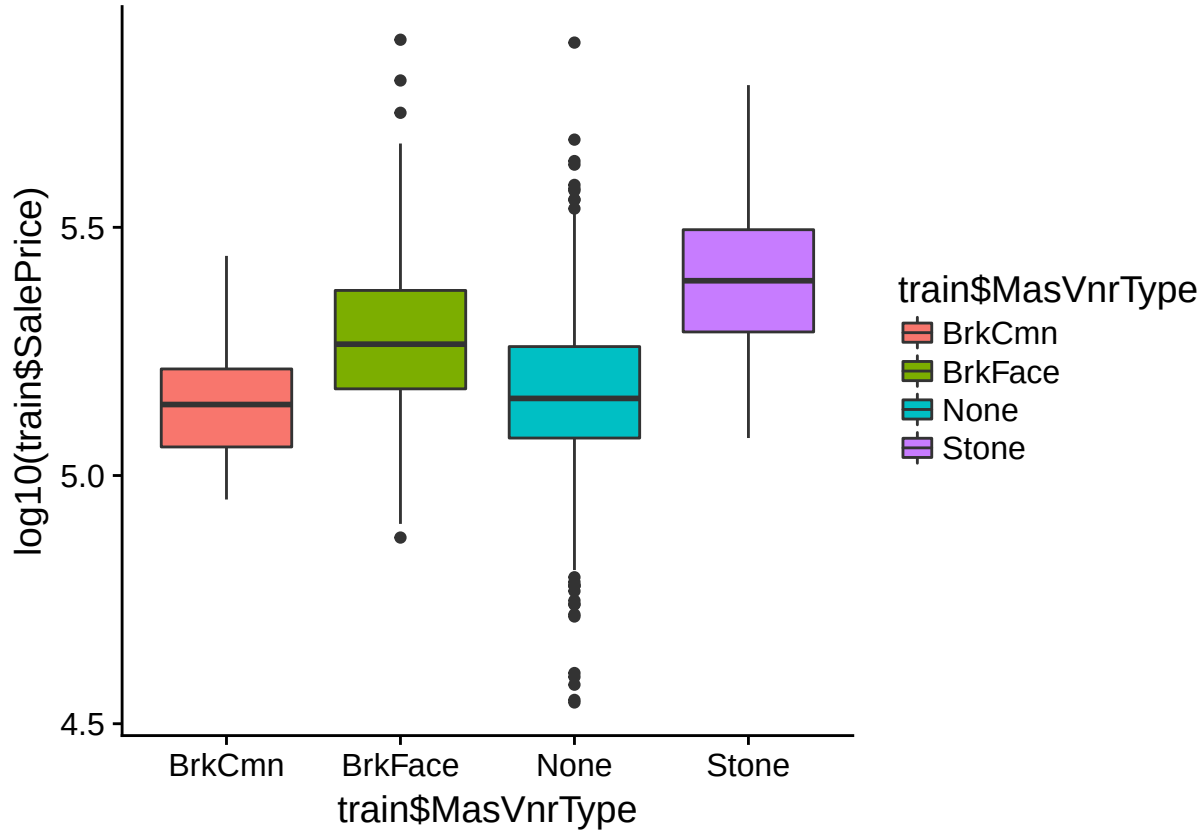


Asignamos a los valores perdidos el tipo "BrkFace", por mayor proximidad de sus medias. Aunque también se les podría asignar el tipo "Stone".

```
full.set$MasVnrType[is.na(full.set$MasVnrType)] <- "BrkFace"

updatePartitions()
```

Finalmente, observamos la distribución resultante en las categorías de *MasVnrType* en relación a *SalePrice*.

```
qplot(data = train, x = train$MasVnrType, y = log10(train$SalePrice),
    geom = c("boxplot"), fill = train$MasVnrType)
```



Variables categóricas

Las siguientes caraterísticas con valores perdidos se corresponden con aquellas que contienen entradas en las que hay una ausencia de la propiedad a la que representan. Por ejemplo, la propiedad *PoolQC* representa la calidad de la piscina, pero es obvio que en aquellas propiedades en las haya una ausencia de piscina, será inviable representar su calidad. Por consiguiente, se le asignarán el tipo "None" a aquellos valores ausentes (NA). Las caracteríasticas que contienen este tipo de valores perdidos son: *PoolQC*, *MiscFeature*, *Alley*, *Fence*, *FireplaceQu*, *GarageCond*, *GarageFinish*, *GarageQual*, *GarageType*, *BsmtCond*, *BsmtExposure*, *BsmtFinType1*, *BsmtFinType2* y *BsmtQual*.

```
categorical.features <- c("PoolQC", "MiscFeature", "Alley", "Fence",
    "FireplaceQu", "GarageCond", "GarageFinish", "GarageQual",
    "GarageType", "BsmtCond", "BsmtExposure", "BsmtFinType1",
    "BsmtFinType2", "BsmtQual")

for (feature in categorical.features) {
    if (is.factor(full.set[, feature])) {
        full.set[, feature] <- as.character(full.set[, feature])
        full.set[, feature][which(is.na(full.set[, feature]))] <- "None"
        full.set[, feature] <- factor(full.set[, feature])
    }
}

updatePartitions()
```

- Electrical

Esta característica presenta un valor perdido. Dada la mínima influencia que puede tener, se le asigna la categoría mayoritaria.

```
full.set$Electrical[is.na(full.set$Electrical)] <- as.character(sort(full.set$Electrical,
    decreasing = TRUE)[1])

updatePartitions()
```

Corrección de valores perdidos en el conjunto de test

En primer lugar, se examinan los valores perdidos que presenta el conjunto de test y se visualizan de la misma forma que se procedió con el conjunto de entrenamiento.

```
getLostValuesStats()
```

```
##              lost.count lost.percentage
## MSZoning              4      0.13703323
## Utilities             2      0.06851662
## Exterior1st           1      0.03425831
## Exterior2nd           1      0.03425831
## BsmtFinSF1            1      0.03425831
## BsmtFinSF2            1      0.03425831
## BsmtUnfSF             1      0.03425831
## TotalBsmtSF           1      0.03425831
## BsmtFullBath          2      0.06851662
## BsmtHalfBath          2      0.06851662
## KitchenQual           1      0.03425831
## Functional            2      0.06851662
## GarageCars            1      0.03425831
## GarageArea            1      0.03425831
## SaleType              1      0.03425831
```

TODO: ¿En las caraterísticas que presenten valores perdidos y ya se haya examinado previamente, se proceden a tratar de la misma forma para realizar un procedimiento consistente?

Así mismo, se procede a tratar las demás variables. Por una parte, se procesan las de tipo numérico, asignando la mediana a los valores faltantes.

```
lost.test.numeric.features <- c("BsmtFinSF1", "BsmtFinSF2", "BsmtUnfSF",
    "TotalBsmtSF", "BsmtFullBath", "BsmtHalfBath", "GarageCars",
    "GarageArea")

for (feature in lost.test.numeric.features) {
    full.set[, feature][is.na(full.set[, feature])] <- median(full.set[,
        feature][!is.na(full.set[, feature])])
}

updatePartitions()
```

En cuanto a las variables de tipo nominal, se les asigna la moda de sus valores.

```
lost.test.categorical.features <- c("Exterior1st", "Exterior2nd",
    "Functional", "KitchenQual", "MSZoning", "SaleType", "Utilities")

for (feature in lost.test.categorical.features) {
    full.set[, feature][is.na(full.set[, feature])] <- as.character(sort(full.set[,
        feature], decreasing = T)[1])
}

updatePartitions()
```

Antes de continuar, comprobamos que no hay valores perdidos en ninguno de los dos conjuntos.

```
getLostValuesStats()
```

```
## [1] lost.count      lost.percentage
## <0 rows> (or 0-length row.names)
```

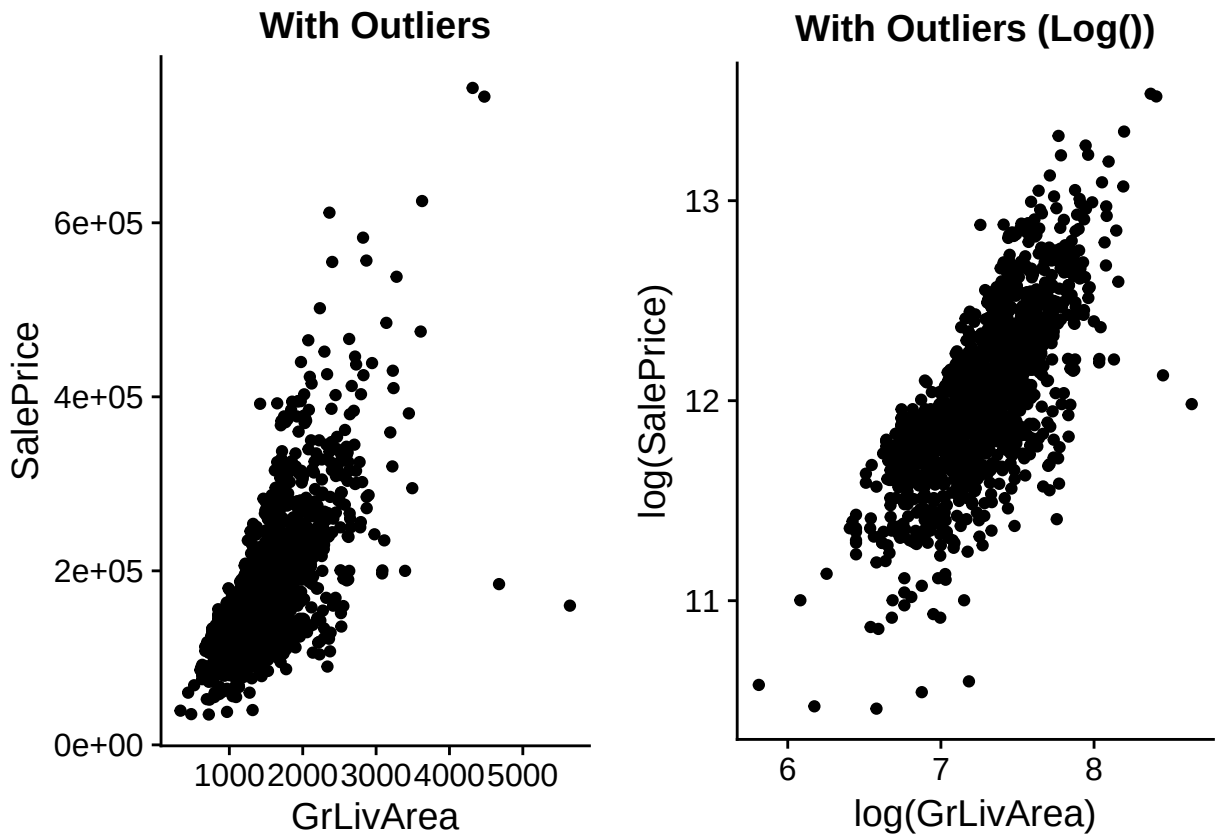
Transformación de datos

En primer lugar, se procede a eliminar la propiedad Id de los conjuntos de entrenamiento y test.

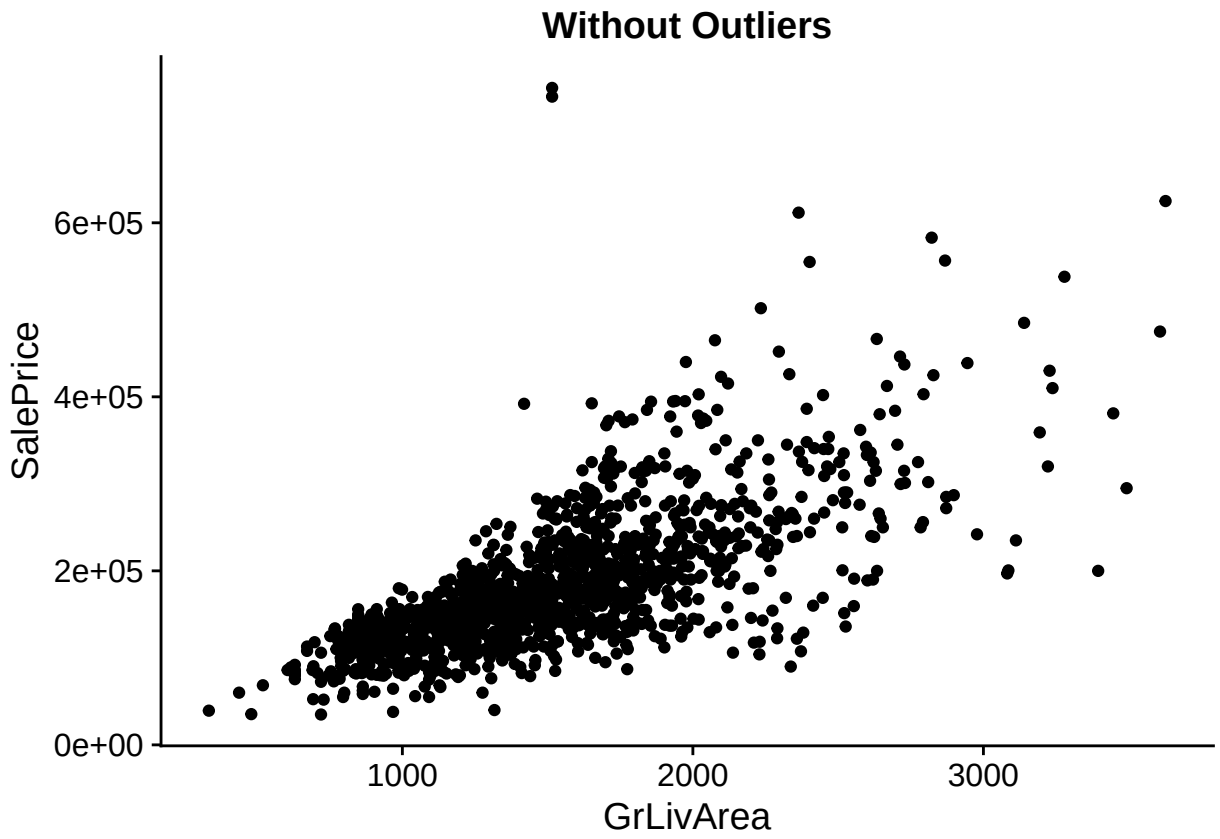
```
train.transformed <- dplyr::select(train, -Id)
test.transformed <- dplyr::select(test, -Id)
```

Tratamiento de outliers

```
plot.with.outliers <- ggplot(train.transformed, aes(y = SalePrice,
    x = GrLivArea)) + ggtitle("With Outliers") + geom_point()
plot.with.outliers.log <- ggplot(train.transformed, aes(y = log(SalePrice),
    x = log(GrLivArea))) + ggtitle("With Outliers (Log())") +
    geom_point()
cowplot::plot_grid(plot.with.outliers, plot.with.outliers.log,
    ncol = 2)
```




```
train.transformed[train.transformed$GrLivArea > 4000, ]$GrLivArea <- mean(train.transformed$GrLivArea) %
    as.numeric

ggplot(train.transformed, aes(y = SalePrice, x = GrLivArea)) +
    ggtitle("Without Outliers") + geom_point()
```




```
full.set.transformed <- data.frame(data.table::rbindlist(list(train.transformed,
    test.transformed), use.names = F, fill = F))
```

- Feature engeneriing

Creación de una nueva variable: Area total basement e

```
full.set.transformed <- dplyr::select(full.set.transformed, -LotFrontage)

full.set.transformed$TotalSF = full.set.transformed$TotalBsmtSF +
    full.set.transformed$X1stFlrSF + full.set.transformed$X2ndFlrSF

full.set.transformed$Age <- full.set.transformed$YrSold - full.set.transformed$YearRemodAdd

full.set.transformed$TotalProch <- full.set.transformed$EnclosedPorch +
    full.set.transformed$ScreenPorch + full.set.transformed$X3SsnPorch
```

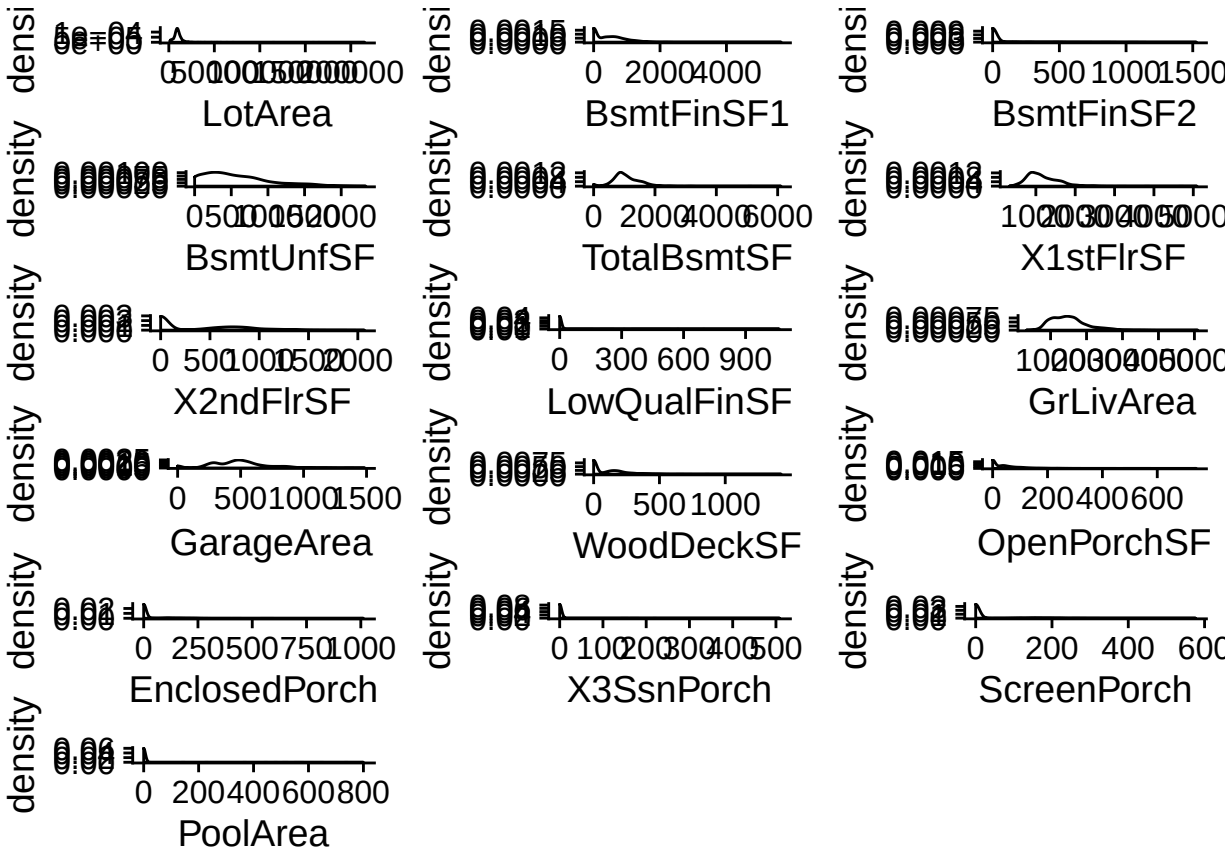
Regularización de las variables continuas

Tal y como se ha mostrado, la variable *SalePrice* contiene una distribución asimétrica. Por consiguiente, para evitar el efecto que los valores extremos puedan causar, se procede a aplicar logaritmos a los valores de la distribución.

Así mismo, se procede a mostrar diagramas de densidad de cada una de las características que contengan datos númericos. De esta forma se podrá observa que transformaciones pueden ser convenientes de hacer a cada variable.

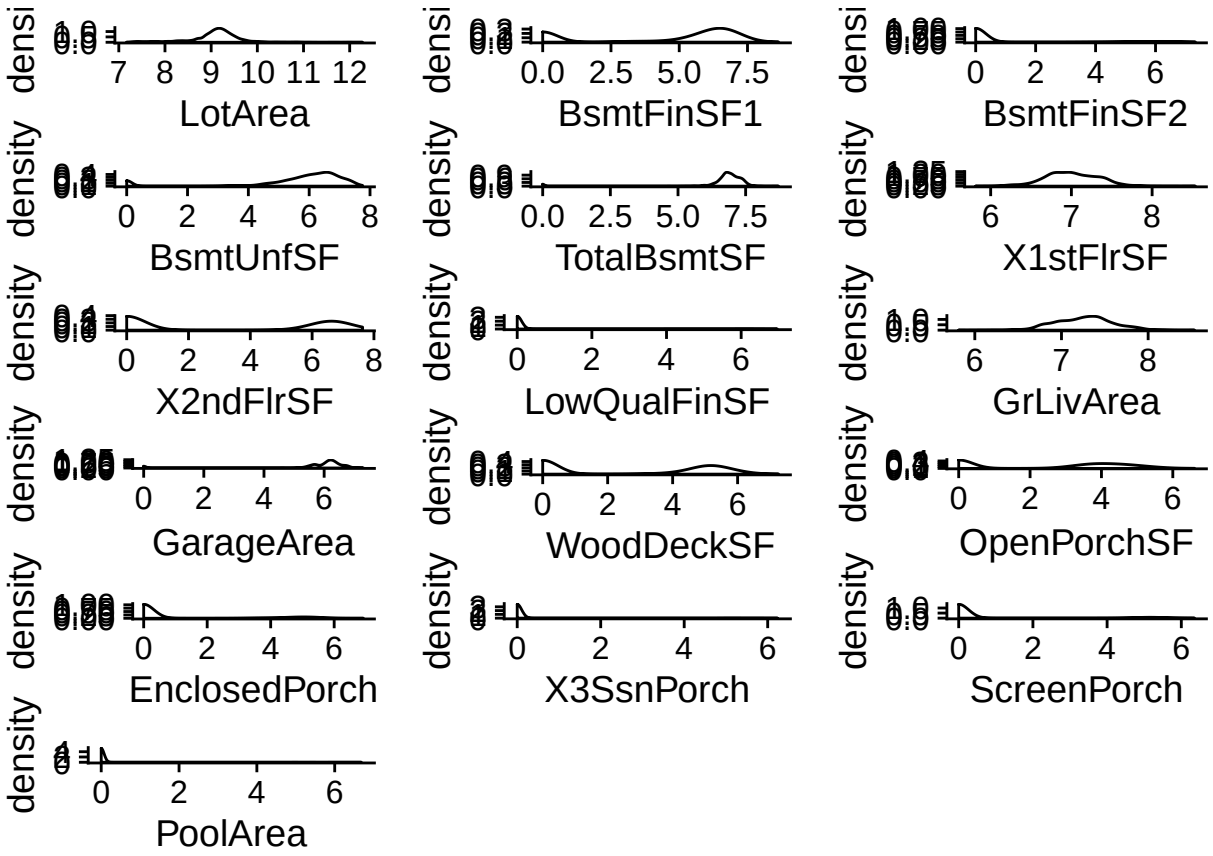
```r
full.set.transformed$SalePrice <- log(full.set.transformed$SalePrice)
```

```r
continuous.features <- c(
  "LotArea", ## Lot size in square feet
  "BsmtFinSF1", ## Type 1 finished square feet
  "BsmtFinSF2", ## Type 2 finished square feet
  "BsmtUnfSF", ## Unfinished square feet of basement area
  "TotalBsmtSF", ## Total square feet of basement area
  "X1stFlrSF", ## First Floor square feet
  "X2ndFlrSF", ## Second floor square feet
  "LowQualFinSF", ## Low quality finished square feet (all floors)
  "GrLivArea", ## Above grade (ground) living area square feet
  "GarageArea", ## Size of garage in square feet
  "WoodDeckSF", ## Wood deck area in square feet
  "OpenPorchSF", ## Open porch area in square feet
  "EnclosedPorch", ## Enclosed porch area in square feet
  "X3SsnPorch", ## Three season porch area in square feet
  "ScreenPorch", ## Screen porch area in square feet
  "PoolArea" ## Pool area in square feet
)

plots <- lapply(continuous.features, function(feature) {
    if (is.numeric(full.set.transformed[, feature])) {
        ggplot2::ggplot(data = full.set.transformed, aes(x = full.set.transformed[, feature])) +
                    geom_density() +
                    xlab(feature)
    }
})

cowplot::plot_grid(plotlist = plots, ncol = 3)
```



Let's normalize the continuous values

```
full.set.transformed[, continuous.features] <- log(1 + full.set.transformed[,
    continuous.features])

plots <- lapply(continuous.features, function(feature) {
    if (is.numeric(full.set.transformed[, feature])) {
        ggplot2::ggplot(data = full.set.transformed, aes(x = full.set.transformed[,
            feature])) + geom_density() + xlab(feature)
    }
})

cowplot::plot_grid(plotlist = plots, ncol = 3)
```



Center & scale....

```
full.set.preProcessed <- caret::preProcess(select(full.set.transformed,
    -SalePrice), method = c("center", "scale"))
full.set.transformed <- predict(full.set.preProcessed, full.set.transformed)
```

Cambiado las variable categóricas por numéricas # TODO: Buscar explicación

```
for (i in 1:ncol(full.set.transformed)) {
    if (is.factor(full.set.transformed[, i])) {
        levels(full.set.transformed[, i]) <- c(1:length(levels(full.set.transformed[,
            i])))
        full.set.transformed[, i] <- as.numeric(full.set.transformed[,
            i])
    }
}
```

- Spliting into train and test

```
train.processed <- full.set.transformed[1:nrow(train.transformed),
    ]
test.processed <- full.set.transformed[(nrow(train.transformed) +
    1):nrow(full.set.transformed), ]
test.processed <- dplyr::select(test.processed, -SalePrice)
```

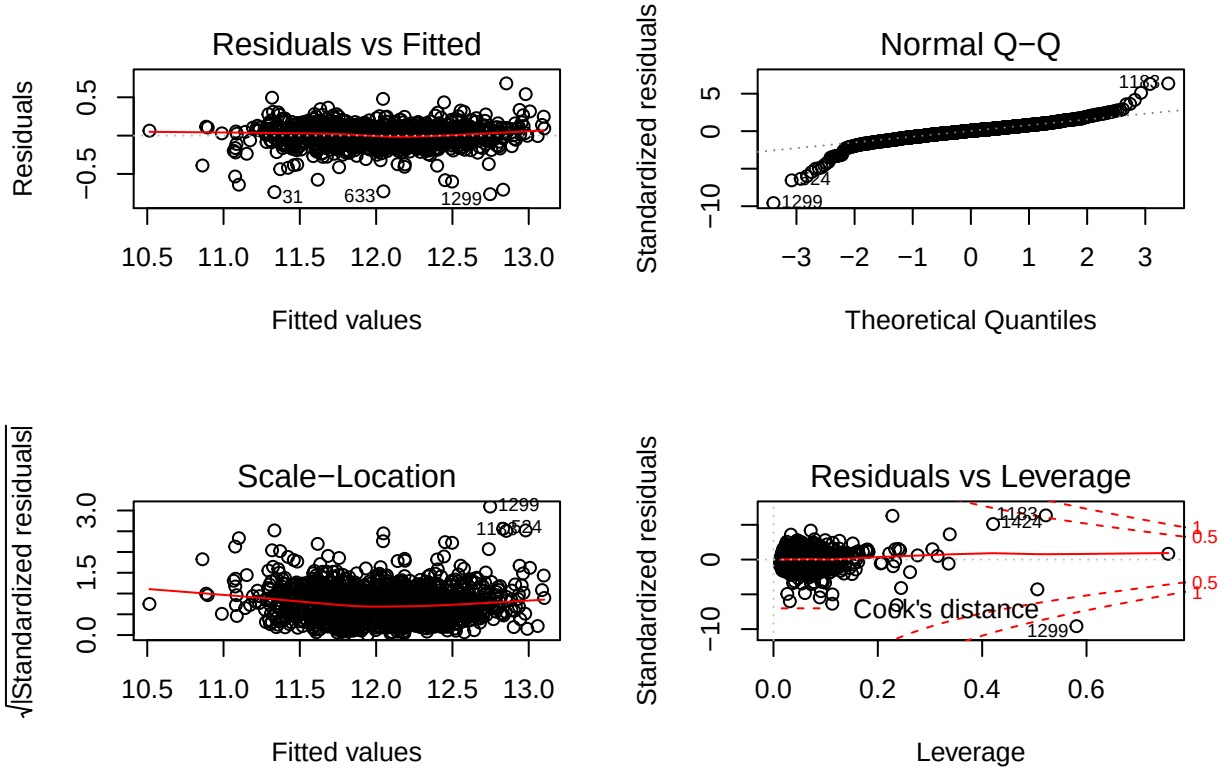
- lm

```
exploratory.lm = lm(SalePrice ~ ., data = train.processed)
```

```
par(mfrow = c(2, 2))
plot(exploratory.lm)
```

```
## Warning: not plotting observations with leverage one:
##   945
```

```
## Warning: not plotting observations with leverage one:
##   945
```




```
par(mfrow = c(1, 1))
```

```
importance <- caret::varImp(exploratory.lm)
importance.sort <- sort(importance$Overall, decreasing = TRUE,
    index.return = TRUE)

data.frame(Feature = rownames(importance)[importance.sort$ix],
    Overall = importance[importance.sort$ix, ])[1:15, ]
```

```
##          Feature   Overall
## 1      GrLivArea 12.731488
## 2    OverallQual 12.235935
## 3    OverallCond 11.489647
## 4        LotArea  8.350168
## 5  SaleCondition  6.885938
## 6     Functional  6.117128
## 7      YearBuilt  5.694831
## 8     GarageCars  5.470990
```

```
## 9         PoolQC  5.435886
## 10      PoolArea  4.572355
## 11  KitchenAbvGr  4.515456
## 12    BsmtFinSF1  4.364160
## 13   KitchenQual  3.949574
## 14    Fireplaces  3.809485
## 15     X2ndFlrSF  3.751977
```

Entrenamiento

- Entrenamiento "parcial"

```
train.processed.partition.index <- createDataPartition(train.processed$SalePrice,
    p = 0.7, list = FALSE)
train.processed.partition.train <- train.processed[train.processed.partition.index,
    ]
train.processed.partition.validation <- train.processed[-train.processed.partition.index,
    ]
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1501             nan     0.1000    0.0133
##      2        0.1370             nan     0.1000    0.0120
##      3        0.1272             nan     0.1000    0.0092
##      4        0.1176             nan     0.1000    0.0092
##      5        0.1097             nan     0.1000    0.0071
##      6        0.1033             nan     0.1000    0.0068
##      7        0.0973             nan     0.1000    0.0065
##      8        0.0920             nan     0.1000    0.0057
##      9        0.0869             nan     0.1000    0.0049
##     10        0.0817             nan     0.1000    0.0047
##     20        0.0520             nan     0.1000    0.0015
##     40        0.0306             nan     0.1000    0.0005
##     60        0.0233             nan     0.1000    0.0001
##     80        0.0200             nan     0.1000    0.0001
##    100        0.0181             nan     0.1000    0.0000
##    120        0.0169             nan     0.1000   -0.0000
##    140        0.0161             nan     0.1000   -0.0000
##    150        0.0158             nan     0.1000   -0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1457             nan     0.1000    0.0175
##      2        0.1307             nan     0.1000    0.0138
##      3        0.1177             nan     0.1000    0.0132
##      4        0.1066             nan     0.1000    0.0105
##      5        0.0970             nan     0.1000    0.0092
##      6        0.0878             nan     0.1000    0.0083
##      7        0.0806             nan     0.1000    0.0070
##      8        0.0738             nan     0.1000    0.0064
##      9        0.0682             nan     0.1000    0.0046
##     10        0.0631             nan     0.1000    0.0043
##     20        0.0351             nan     0.1000    0.0015
##     40        0.0208             nan     0.1000    0.0001
##     60        0.0167             nan     0.1000   -0.0000
##     80        0.0145             nan     0.1000    0.0000
```

```
##    100        0.0132             nan     0.1000   -0.0001
##    120        0.0123             nan     0.1000   -0.0000
##    140        0.0117             nan     0.1000   -0.0001
##    150        0.0115             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1431             nan     0.1000    0.0218
##      2        0.1265             nan     0.1000    0.0162
##      3        0.1120             nan     0.1000    0.0140
##      4        0.1003             nan     0.1000    0.0114
##      5        0.0896             nan     0.1000    0.0096
##      6        0.0810             nan     0.1000    0.0087
##      7        0.0731             nan     0.1000    0.0073
##      8        0.0668             nan     0.1000    0.0060
##      9        0.0613             nan     0.1000    0.0054
##     10        0.0568             nan     0.1000    0.0037
##     20        0.0297             nan     0.1000    0.0013
##     40        0.0172             nan     0.1000    0.0001
##     60        0.0136             nan     0.1000    0.0000
##     80        0.0118             nan     0.1000   -0.0000
##    100        0.0107             nan     0.1000   -0.0001
##    120        0.0100             nan     0.1000   -0.0001
##    140        0.0094             nan     0.1000   -0.0001
##    150        0.0091             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1492             nan     0.1000    0.0135
##      2        0.1365             nan     0.1000    0.0121
##      3        0.1255             nan     0.1000    0.0094
##      4        0.1161             nan     0.1000    0.0087
##      5        0.1087             nan     0.1000    0.0080
##      6        0.1024             nan     0.1000    0.0057
##      7        0.0958             nan     0.1000    0.0057
##      8        0.0901             nan     0.1000    0.0058
##      9        0.0847             nan     0.1000    0.0052
##     10        0.0800             nan     0.1000    0.0043
##     20        0.0507             nan     0.1000    0.0016
##     40        0.0299             nan     0.1000    0.0005
##     60        0.0224             nan     0.1000    0.0001
##     80        0.0192             nan     0.1000    0.0001
##    100        0.0172             nan     0.1000    0.0000
##    120        0.0160             nan     0.1000   -0.0000
##    140        0.0151             nan     0.1000   -0.0000
##    150        0.0148             nan     0.1000   -0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1455             nan     0.1000    0.0168
##      2        0.1302             nan     0.1000    0.0151
##      3        0.1173             nan     0.1000    0.0128
##      4        0.1066             nan     0.1000    0.0098
##      5        0.0970             nan     0.1000    0.0091
##      6        0.0879             nan     0.1000    0.0077
##      7        0.0799             nan     0.1000    0.0075
##      8        0.0733             nan     0.1000    0.0066
```

```
##      9        0.0676             nan     0.1000    0.0049
##     10        0.0633             nan     0.1000    0.0041
##     20        0.0358             nan     0.1000    0.0014
##     40        0.0208             nan     0.1000    0.0002
##     60        0.0165             nan     0.1000   -0.0000
##     80        0.0145             nan     0.1000    0.0000
##    100        0.0133             nan     0.1000   -0.0000
##    120        0.0124             nan     0.1000   -0.0001
##    140        0.0117             nan     0.1000   -0.0001
##    150        0.0114             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1421             nan     0.1000    0.0210
##      2        0.1251             nan     0.1000    0.0154
##      3        0.1110             nan     0.1000    0.0133
##      4        0.0992             nan     0.1000    0.0123
##      5        0.0891             nan     0.1000    0.0097
##      6        0.0802             nan     0.1000    0.0087
##      7        0.0731             nan     0.1000    0.0067
##      8        0.0660             nan     0.1000    0.0066
##      9        0.0601             nan     0.1000    0.0049
##     10        0.0552             nan     0.1000    0.0048
##     20        0.0292             nan     0.1000    0.0014
##     40        0.0169             nan     0.1000    0.0000
##     60        0.0132             nan     0.1000    0.0000
##     80        0.0117             nan     0.1000    0.0000
##    100        0.0106             nan     0.1000   -0.0000
##    120        0.0098             nan     0.1000   -0.0001
##    140        0.0092             nan     0.1000   -0.0000
##    150        0.0089             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1432             nan     0.1000    0.0133
##      2        0.1316             nan     0.1000    0.0118
##      3        0.1208             nan     0.1000    0.0101
##      4        0.1123             nan     0.1000    0.0087
##      5        0.1042             nan     0.1000    0.0080
##      6        0.0973             nan     0.1000    0.0061
##      7        0.0913             nan     0.1000    0.0054
##      8        0.0861             nan     0.1000    0.0053
##      9        0.0810             nan     0.1000    0.0045
##     10        0.0770             nan     0.1000    0.0037
##     20        0.0481             nan     0.1000    0.0017
##     40        0.0281             nan     0.1000    0.0004
##     60        0.0215             nan     0.1000    0.0001
##     80        0.0186             nan     0.1000    0.0001
##    100        0.0170             nan     0.1000    0.0000
##    120        0.0159             nan     0.1000   -0.0000
##    140        0.0151             nan     0.1000   -0.0000
##    150        0.0148             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1399             nan     0.1000    0.0167
##      2        0.1249             nan     0.1000    0.0150
```

```
##      3        0.1120             nan     0.1000    0.0117
##      4        0.1017             nan     0.1000    0.0103
##      5        0.0921             nan     0.1000    0.0091
##      6        0.0841             nan     0.1000    0.0076
##      7        0.0777             nan     0.1000    0.0064
##      8        0.0710             nan     0.1000    0.0068
##      9        0.0652             nan     0.1000    0.0058
##     10        0.0597             nan     0.1000    0.0049
##     20        0.0339             nan     0.1000    0.0014
##     40        0.0196             nan     0.1000    0.0003
##     60        0.0156             nan     0.1000    0.0000
##     80        0.0137             nan     0.1000    0.0000
##    100        0.0127             nan     0.1000   -0.0001
##    120        0.0121             nan     0.1000   -0.0000
##    140        0.0114             nan     0.1000   -0.0000
##    150        0.0111             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1377             nan     0.1000    0.0183
##      2        0.1219             nan     0.1000    0.0153
##      3        0.1077             nan     0.1000    0.0138
##      4        0.0952             nan     0.1000    0.0118
##      5        0.0852             nan     0.1000    0.0094
##      6        0.0768             nan     0.1000    0.0083
##      7        0.0696             nan     0.1000    0.0066
##      8        0.0634             nan     0.1000    0.0060
##      9        0.0578             nan     0.1000    0.0053
##     10        0.0530             nan     0.1000    0.0046
##     20        0.0279             nan     0.1000    0.0011
##     40        0.0160             nan     0.1000    0.0001
##     60        0.0127             nan     0.1000    0.0000
##     80        0.0112             nan     0.1000   -0.0000
##    100        0.0101             nan     0.1000   -0.0001
##    120        0.0093             nan     0.1000   -0.0000
##    140        0.0087             nan     0.1000   -0.0001
##    150        0.0084             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1478             nan     0.1000    0.0142
##      2        0.1351             nan     0.1000    0.0117
##      3        0.1250             nan     0.1000    0.0098
##      4        0.1162             nan     0.1000    0.0083
##      5        0.1081             nan     0.1000    0.0084
##      6        0.1004             nan     0.1000    0.0074
##      7        0.0940             nan     0.1000    0.0059
##      8        0.0880             nan     0.1000    0.0055
##      9        0.0830             nan     0.1000    0.0045
##     10        0.0784             nan     0.1000    0.0043
##     20        0.0486             nan     0.1000    0.0017
##     40        0.0289             nan     0.1000    0.0004
##     60        0.0222             nan     0.1000    0.0000
##     80        0.0191             nan     0.1000   -0.0001
##    100        0.0174             nan     0.1000    0.0000
##    120        0.0162             nan     0.1000   -0.0000
```

```
##    140        0.0153             nan     0.1000   -0.0000
##    150        0.0150             nan     0.1000   -0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1454             nan     0.1000    0.0179
##      2        0.1301             nan     0.1000    0.0136
##      3        0.1168             nan     0.1000    0.0135
##      4        0.1055             nan     0.1000    0.0112
##      5        0.0951             nan     0.1000    0.0107
##      6        0.0873             nan     0.1000    0.0068
##      7        0.0797             nan     0.1000    0.0074
##      8        0.0730             nan     0.1000    0.0056
##      9        0.0680             nan     0.1000    0.0043
##     10        0.0627             nan     0.1000    0.0049
##     20        0.0355             nan     0.1000    0.0014
##     40        0.0203             nan     0.1000    0.0002
##     60        0.0161             nan     0.1000   -0.0000
##     80        0.0141             nan     0.1000    0.0001
##    100        0.0129             nan     0.1000   -0.0000
##    120        0.0121             nan     0.1000   -0.0000
##    140        0.0114             nan     0.1000   -0.0000
##    150        0.0112             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1413             nan     0.1000    0.0194
##      2        0.1238             nan     0.1000    0.0163
##      3        0.1091             nan     0.1000    0.0138
##      4        0.0974             nan     0.1000    0.0117
##      5        0.0869             nan     0.1000    0.0086
##      6        0.0780             nan     0.1000    0.0084
##      7        0.0707             nan     0.1000    0.0062
##      8        0.0641             nan     0.1000    0.0063
##      9        0.0588             nan     0.1000    0.0045
##     10        0.0535             nan     0.1000    0.0050
##     20        0.0285             nan     0.1000    0.0008
##     40        0.0165             nan     0.1000    0.0002
##     60        0.0133             nan     0.1000   -0.0000
##     80        0.0116             nan     0.1000   -0.0001
##    100        0.0104             nan     0.1000   -0.0001
##    120        0.0097             nan     0.1000   -0.0001
##    140        0.0091             nan     0.1000   -0.0000
##    150        0.0088             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1528             nan     0.1000    0.0139
##      2        0.1399             nan     0.1000    0.0127
##      3        0.1293             nan     0.1000    0.0102
##      4        0.1200             nan     0.1000    0.0093
##      5        0.1122             nan     0.1000    0.0079
##      6        0.1048             nan     0.1000    0.0072
##      7        0.0980             nan     0.1000    0.0053
##      8        0.0923             nan     0.1000    0.0055
##      9        0.0866             nan     0.1000    0.0061
##     10        0.0822             nan     0.1000    0.0043
```

```
##     20        0.0521             nan     0.1000    0.0014
##     40        0.0300             nan     0.1000    0.0004
##     60        0.0223             nan     0.1000    0.0001
##     80        0.0188             nan     0.1000   -0.0000
##    100        0.0168             nan     0.1000    0.0000
##    120        0.0155             nan     0.1000    0.0000
##    140        0.0146             nan     0.1000   -0.0000
##    150        0.0142             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1483             nan     0.1000    0.0161
##      2        0.1331             nan     0.1000    0.0161
##      3        0.1202             nan     0.1000    0.0135
##      4        0.1082             nan     0.1000    0.0111
##      5        0.0973             nan     0.1000    0.0106
##      6        0.0893             nan     0.1000    0.0072
##      7        0.0815             nan     0.1000    0.0073
##      8        0.0755             nan     0.1000    0.0061
##      9        0.0691             nan     0.1000    0.0064
##     10        0.0636             nan     0.1000    0.0051
##     20        0.0351             nan     0.1000    0.0014
##     40        0.0203             nan     0.1000    0.0002
##     60        0.0158             nan     0.1000    0.0000
##     80        0.0137             nan     0.1000    0.0000
##    100        0.0125             nan     0.1000    0.0000
##    120        0.0116             nan     0.1000   -0.0001
##    140        0.0109             nan     0.1000    0.0000
##    150        0.0106             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1442             nan     0.1000    0.0199
##      2        0.1263             nan     0.1000    0.0166
##      3        0.1117             nan     0.1000    0.0145
##      4        0.1001             nan     0.1000    0.0115
##      5        0.0900             nan     0.1000    0.0105
##      6        0.0810             nan     0.1000    0.0081
##      7        0.0733             nan     0.1000    0.0066
##      8        0.0663             nan     0.1000    0.0064
##      9        0.0603             nan     0.1000    0.0052
##     10        0.0551             nan     0.1000    0.0049
##     20        0.0285             nan     0.1000    0.0011
##     40        0.0163             nan     0.1000    0.0003
##     60        0.0126             nan     0.1000   -0.0001
##     80        0.0110             nan     0.1000   -0.0001
##    100        0.0100             nan     0.1000   -0.0000
##    120        0.0091             nan     0.1000   -0.0000
##    140        0.0085             nan     0.1000   -0.0000
##    150        0.0083             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1497             nan     0.1000    0.0134
##      2        0.1372             nan     0.1000    0.0126
##      3        0.1269             nan     0.1000    0.0093
##      4        0.1176             nan     0.1000    0.0087
```

```
##      5        0.1101             nan     0.1000    0.0070
##      6        0.1027             nan     0.1000    0.0066
##      7        0.0959             nan     0.1000    0.0063
##      8        0.0900             nan     0.1000    0.0060
##      9        0.0852             nan     0.1000    0.0042
##     10        0.0811             nan     0.1000    0.0042
##     20        0.0512             nan     0.1000    0.0017
##     40        0.0297             nan     0.1000    0.0003
##     60        0.0224             nan     0.1000    0.0002
##     80        0.0188             nan     0.1000   -0.0001
##    100        0.0169             nan     0.1000   -0.0000
##    120        0.0158             nan     0.1000    0.0000
##    140        0.0149             nan     0.1000    0.0000
##    150        0.0146             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1459             nan     0.1000    0.0148
##      2        0.1308             nan     0.1000    0.0157
##      3        0.1170             nan     0.1000    0.0137
##      4        0.1052             nan     0.1000    0.0107
##      5        0.0957             nan     0.1000    0.0094
##      6        0.0880             nan     0.1000    0.0077
##      7        0.0806             nan     0.1000    0.0077
##      8        0.0741             nan     0.1000    0.0059
##      9        0.0684             nan     0.1000    0.0057
##     10        0.0635             nan     0.1000    0.0046
##     20        0.0359             nan     0.1000    0.0017
##     40        0.0200             nan     0.1000    0.0003
##     60        0.0155             nan     0.1000    0.0000
##     80        0.0133             nan     0.1000    0.0000
##    100        0.0121             nan     0.1000   -0.0001
##    120        0.0112             nan     0.1000    0.0000
##    140        0.0105             nan     0.1000   -0.0000
##    150        0.0102             nan     0.1000   -0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1440             nan     0.1000    0.0210
##      2        0.1266             nan     0.1000    0.0181
##      3        0.1119             nan     0.1000    0.0133
##      4        0.0996             nan     0.1000    0.0114
##      5        0.0891             nan     0.1000    0.0109
##      6        0.0802             nan     0.1000    0.0084
##      7        0.0723             nan     0.1000    0.0075
##      8        0.0663             nan     0.1000    0.0061
##      9        0.0604             nan     0.1000    0.0051
##     10        0.0551             nan     0.1000    0.0047
##     20        0.0289             nan     0.1000    0.0014
##     40        0.0161             nan     0.1000    0.0002
##     60        0.0127             nan     0.1000   -0.0001
##     80        0.0111             nan     0.1000   -0.0000
##    100        0.0101             nan     0.1000   -0.0001
##    120        0.0092             nan     0.1000   -0.0000
##    140        0.0087             nan     0.1000   -0.0001
##    150        0.0084             nan     0.1000   -0.0000
```

```
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1484             nan     0.1000    0.0137
##      2        0.1360             nan     0.1000    0.0113
##      3        0.1251             nan     0.1000    0.0101
##      4        0.1164             nan     0.1000    0.0087
##      5        0.1085             nan     0.1000    0.0078
##      6        0.1005             nan     0.1000    0.0072
##      7        0.0942             nan     0.1000    0.0063
##      8        0.0885             nan     0.1000    0.0055
##      9        0.0835             nan     0.1000    0.0047
##     10        0.0788             nan     0.1000    0.0044
##     20        0.0501             nan     0.1000    0.0017
##     40        0.0303             nan     0.1000    0.0002
##     60        0.0229             nan     0.1000    0.0000
##     80        0.0194             nan     0.1000    0.0000
##    100        0.0174             nan     0.1000   -0.0001
##    120        0.0159             nan     0.1000    0.0000
##    140        0.0149             nan     0.1000   -0.0000
##    150        0.0146             nan     0.1000   -0.0001
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1441             nan     0.1000    0.0160
##      2        0.1282             nan     0.1000    0.0155
##      3        0.1156             nan     0.1000    0.0128
##      4        0.1046             nan     0.1000    0.0109
##      5        0.0945             nan     0.1000    0.0101
##      6        0.0859             nan     0.1000    0.0088
##      7        0.0792             nan     0.1000    0.0058
##      8        0.0722             nan     0.1000    0.0066
##      9        0.0671             nan     0.1000    0.0041
##     10        0.0620             nan     0.1000    0.0046
##     20        0.0351             nan     0.1000    0.0015
##     40        0.0208             nan     0.1000    0.0002
##     60        0.0162             nan     0.1000    0.0001
##     80        0.0141             nan     0.1000    0.0001
##    100        0.0127             nan     0.1000   -0.0000
##    120        0.0118             nan     0.1000   -0.0001
##    140        0.0113             nan     0.1000    0.0000
##    150        0.0109             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1424             nan     0.1000    0.0192
##      2        0.1249             nan     0.1000    0.0162
##      3        0.1106             nan     0.1000    0.0145
##      4        0.0987             nan     0.1000    0.0118
##      5        0.0883             nan     0.1000    0.0095
##      6        0.0794             nan     0.1000    0.0088
##      7        0.0720             nan     0.1000    0.0069
##      8        0.0656             nan     0.1000    0.0066
##      9        0.0602             nan     0.1000    0.0055
##     10        0.0552             nan     0.1000    0.0047
##     20        0.0295             nan     0.1000    0.0014
##     40        0.0166             nan     0.1000    0.0002
```

```
##     60        0.0132             nan     0.1000    0.0000
##     80        0.0115             nan     0.1000   -0.0001
##    100        0.0105             nan     0.1000   -0.0001
##    120        0.0096             nan     0.1000   -0.0000
##    140        0.0091             nan     0.1000   -0.0000
##    150        0.0087             nan     0.1000   -0.0000
```

```
## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 8: Utilities has no variation.
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1449             nan     0.1000    0.0136
##      2        0.1331             nan     0.1000    0.0114
##      3        0.1235             nan     0.1000    0.0102
##      4        0.1151             nan     0.1000    0.0087
##      5        0.1071             nan     0.1000    0.0081
##      6        0.1003             nan     0.1000    0.0063
##      7        0.0937             nan     0.1000    0.0064
##      8        0.0881             nan     0.1000    0.0051
##      9        0.0833             nan     0.1000    0.0048
##     10        0.0783             nan     0.1000    0.0044
##     20        0.0503             nan     0.1000    0.0014
##     40        0.0295             nan     0.1000    0.0002
##     60        0.0228             nan     0.1000    0.0002
##     80        0.0195             nan     0.1000    0.0001
##    100        0.0177             nan     0.1000   -0.0000
##    120        0.0164             nan     0.1000    0.0000
##    140        0.0155             nan     0.1000   -0.0000
##    150        0.0152             nan     0.1000   -0.0000
```

```
## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 8: Utilities has no variation.
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1400             nan     0.1000    0.0185
##      2        0.1246             nan     0.1000    0.0141
##      3        0.1123             nan     0.1000    0.0122
##      4        0.1012             nan     0.1000    0.0106
##      5        0.0926             nan     0.1000    0.0079
##      6        0.0849             nan     0.1000    0.0077
##      7        0.0775             nan     0.1000    0.0063
##      8        0.0712             nan     0.1000    0.0058
##      9        0.0660             nan     0.1000    0.0048
##     10        0.0614             nan     0.1000    0.0047
##     20        0.0352             nan     0.1000    0.0016
##     40        0.0207             nan     0.1000    0.0001
##     60        0.0165             nan     0.1000    0.0001
##     80        0.0143             nan     0.1000   -0.0000
##    100        0.0131             nan     0.1000   -0.0000
##    120        0.0125             nan     0.1000   -0.0001
##    140        0.0119             nan     0.1000   -0.0000
##    150        0.0116             nan     0.1000   -0.0001
```

```
## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 8: Utilities has no variation.
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
```

```
##      1        0.1390             nan     0.1000    0.0174
##      2        0.1218             nan     0.1000    0.0176
##      3        0.1074             nan     0.1000    0.0130
##      4        0.0957             nan     0.1000    0.0113
##      5        0.0862             nan     0.1000    0.0095
##      6        0.0772             nan     0.1000    0.0084
##      7        0.0704             nan     0.1000    0.0069
##      8        0.0640             nan     0.1000    0.0056
##      9        0.0586             nan     0.1000    0.0051
##     10        0.0539             nan     0.1000    0.0043
##     20        0.0283             nan     0.1000    0.0010
##     40        0.0167             nan     0.1000    0.0001
##     60        0.0131             nan     0.1000    0.0001
##     80        0.0113             nan     0.1000    0.0000
##    100        0.0104             nan     0.1000   -0.0001
##    120        0.0096             nan     0.1000   -0.0000
##    140        0.0090             nan     0.1000   -0.0000
##    150        0.0087             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1488             nan     0.1000    0.0137
##      2        0.1368             nan     0.1000    0.0120
##      3        0.1270             nan     0.1000    0.0097
##      4        0.1182             nan     0.1000    0.0088
##      5        0.1104             nan     0.1000    0.0077
##      6        0.1032             nan     0.1000    0.0071
##      7        0.0971             nan     0.1000    0.0061
##      8        0.0913             nan     0.1000    0.0058
##      9        0.0859             nan     0.1000    0.0048
##     10        0.0817             nan     0.1000    0.0042
##     20        0.0521             nan     0.1000    0.0016
##     40        0.0303             nan     0.1000    0.0006
##     60        0.0234             nan     0.1000    0.0002
##     80        0.0200             nan     0.1000    0.0001
##    100        0.0181             nan     0.1000   -0.0000
##    120        0.0166             nan     0.1000   -0.0000
##    140        0.0157             nan     0.1000    0.0000
##    150        0.0153             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1447             nan     0.1000    0.0176
##      2        0.1288             nan     0.1000    0.0156
##      3        0.1164             nan     0.1000    0.0122
##      4        0.1057             nan     0.1000    0.0102
##      5        0.0965             nan     0.1000    0.0088
##      6        0.0873             nan     0.1000    0.0083
##      7        0.0790             nan     0.1000    0.0079
##      8        0.0721             nan     0.1000    0.0068
##      9        0.0666             nan     0.1000    0.0053
##     10        0.0620             nan     0.1000    0.0044
##     20        0.0353             nan     0.1000    0.0013
##     40        0.0206             nan     0.1000    0.0002
##     60        0.0164             nan     0.1000    0.0000
##     80        0.0144             nan     0.1000   -0.0000
```

```
##    100        0.0133             nan     0.1000   -0.0001
##    120        0.0124             nan     0.1000   -0.0001
##    140        0.0116             nan     0.1000   -0.0000
##    150        0.0113             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1422             nan     0.1000    0.0193
##      2        0.1259             nan     0.1000    0.0169
##      3        0.1109             nan     0.1000    0.0132
##      4        0.0984             nan     0.1000    0.0123
##      5        0.0879             nan     0.1000    0.0102
##      6        0.0802             nan     0.1000    0.0075
##      7        0.0729             nan     0.1000    0.0062
##      8        0.0661             nan     0.1000    0.0068
##      9        0.0606             nan     0.1000    0.0051
##     10        0.0555             nan     0.1000    0.0048
##     20        0.0305             nan     0.1000    0.0013
##     40        0.0177             nan     0.1000    0.0001
##     60        0.0139             nan     0.1000   -0.0000
##     80        0.0122             nan     0.1000   -0.0001
##    100        0.0108             nan     0.1000   -0.0000
##    120        0.0099             nan     0.1000   -0.0001
##    140        0.0091             nan     0.1000   -0.0001
##    150        0.0088             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1476             nan     0.1000    0.0133
##      2        0.1354             nan     0.1000    0.0125
##      3        0.1255             nan     0.1000    0.0099
##      4        0.1172             nan     0.1000    0.0084
##      5        0.1083             nan     0.1000    0.0079
##      6        0.1015             nan     0.1000    0.0069
##      7        0.0956             nan     0.1000    0.0056
##      8        0.0904             nan     0.1000    0.0050
##      9        0.0853             nan     0.1000    0.0051
##     10        0.0805             nan     0.1000    0.0044
##     20        0.0515             nan     0.1000    0.0015
##     40        0.0306             nan     0.1000    0.0002
##     60        0.0234             nan     0.1000    0.0002
##     80        0.0201             nan     0.1000    0.0001
##    100        0.0181             nan     0.1000    0.0000
##    120        0.0170             nan     0.1000   -0.0000
##    140        0.0160             nan     0.1000   -0.0000
##    150        0.0158             nan     0.1000    0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1438             nan     0.1000    0.0175
##      2        0.1280             nan     0.1000    0.0147
##      3        0.1158             nan     0.1000    0.0119
##      4        0.1054             nan     0.1000    0.0098
##      5        0.0965             nan     0.1000    0.0093
##      6        0.0879             nan     0.1000    0.0081
##      7        0.0813             nan     0.1000    0.0076
##      8        0.0756             nan     0.1000    0.0063
```

```
##      9        0.0697             nan     0.1000    0.0059
##     10        0.0643             nan     0.1000    0.0049
##     20        0.0364             nan     0.1000    0.0014
##     40        0.0213             nan     0.1000    0.0002
##     60        0.0168             nan     0.1000    0.0000
##     80        0.0146             nan     0.1000   -0.0000
##    100        0.0133             nan     0.1000   -0.0000
##    120        0.0124             nan     0.1000   -0.0000
##    140        0.0118             nan     0.1000   -0.0001
##    150        0.0114             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1417             nan     0.1000    0.0191
##      2        0.1253             nan     0.1000    0.0149
##      3        0.1101             nan     0.1000    0.0152
##      4        0.0984             nan     0.1000    0.0111
##      5        0.0888             nan     0.1000    0.0095
##      6        0.0800             nan     0.1000    0.0080
##      7        0.0727             nan     0.1000    0.0074
##      8        0.0659             nan     0.1000    0.0063
##      9        0.0603             nan     0.1000    0.0045
##     10        0.0556             nan     0.1000    0.0045
##     20        0.0298             nan     0.1000    0.0012
##     40        0.0172             nan     0.1000    0.0001
##     60        0.0139             nan     0.1000    0.0000
##     80        0.0124             nan     0.1000   -0.0001
##    100        0.0113             nan     0.1000   -0.0000
##    120        0.0104             nan     0.1000   -0.0001
##    140        0.0099             nan     0.1000   -0.0000
##    150        0.0096             nan     0.1000   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1413             nan     0.1000    0.0222
##      2        0.1244             nan     0.1000    0.0162
##      3        0.1095             nan     0.1000    0.0140
##      4        0.0972             nan     0.1000    0.0119
##      5        0.0873             nan     0.1000    0.0091
##      6        0.0787             nan     0.1000    0.0075
##      7        0.0713             nan     0.1000    0.0070
##      8        0.0650             nan     0.1000    0.0058
##      9        0.0595             nan     0.1000    0.0055
##     10        0.0549             nan     0.1000    0.0043
##     20        0.0293             nan     0.1000    0.0011
##     40        0.0171             nan     0.1000    0.0001
##     60        0.0135             nan     0.1000   -0.0000
##     80        0.0118             nan     0.1000   -0.0000
##    100        0.0108             nan     0.1000   -0.0001
##    120        0.0100             nan     0.1000   -0.0001
##    140        0.0094             nan     0.1000   -0.0000
##    150        0.0091             nan     0.1000   -0.0001
```

```
model.partial  # 0.1320744 # 0.1305035 # 1266631 # 1285614
```

```
## Stochastic Gradient Boosting
##
```

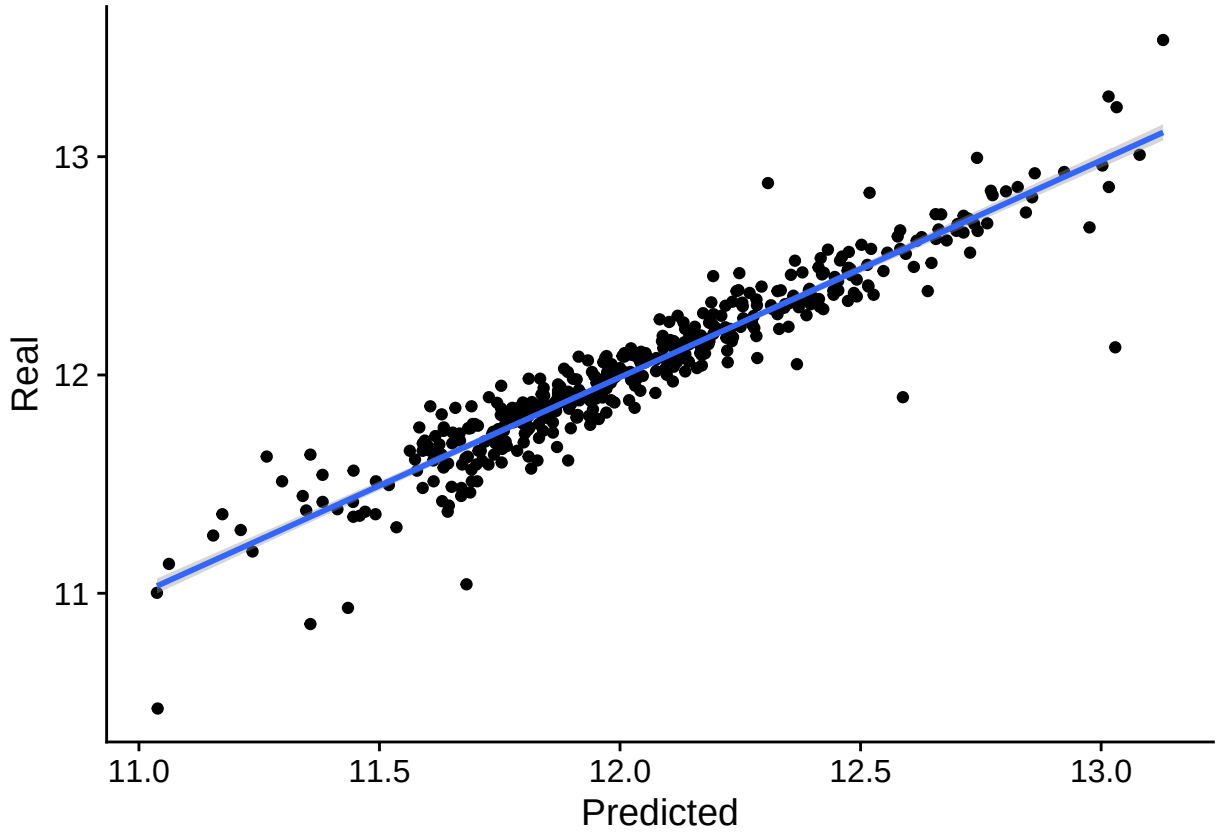
```
## 1024 samples
##   80 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 922, 923, 921, 920, 920, 922, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  RMSE       Rsquared   MAE
##   1                   50      0.1682590  0.8444157  0.11670376
##   1                  100      0.1485041  0.8692648  0.10203270
##   1                  150      0.1417860  0.8797562  0.09726639
##   2                   50      0.1505312  0.8667998  0.10385932
##   2                  100      0.1372135  0.8866656  0.09459342
##   2                  150      0.1352014  0.8898472  0.09223647
##   3                   50      0.1427230  0.8794867  0.09913112
##   3                  100      0.1352764  0.8909551  0.09232436
##   3                  150      0.1340584  0.8929384  0.09098677
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were n.trees = 150,
##  interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.
```

- Validación del modelo

```
predictAndEvaluate <- function(model, validation.set) {
    validation.prediction <- stats::predict(model, dplyr::select(validation.set,
        -SalePrice))

    print(ggplot2::qplot(x = validation.prediction, y = validation.set$SalePrice,
        geom = c("point", "smooth"), method = "lm", xlab = "Predicted",
        ylab = "Real"))

    rmse(validation.set$SalePrice, validation.prediction)
}
```

```
predictAndEvaluate(model.partial, train.processed.partition.validation)  # 0.1299482 # 0.1483049 # 0.14
```

```
## Warning: Ignoring unknown parameters: method
```




```
## [1] 0.1279009
```

Trying to improve model ...

```
ensembleTrain <- function(train.set) {
    set.seed(SEED)

    trControl <- trainControl(method = "cv", number = 7, savePredictions = "final",
        index = createResample(train.set$OverallQual, 7), allowParallel = TRUE)

    modelList <- caretEnsemble::caretList(SalePrice ~ ., data = train.set,
        trControl = trControl, metric = "RMSE", tuneList = list(gbm = caretModelSpec(method = "gbm",
            tuneGrid = expand.grid(n.trees = 700, interaction.depth = 5,
                shrinkage = 0.05, n.minobsinnode = 10)), xgbTree = caretModelSpec(method = "xgbTree",
            tuneGrid = expand.grid(nrounds = 2500, max_depth = 6,
                min_child_weight = 1.41, eta = 0.01, gamma = 0.0468,
                subsample = 0.769, colsample_bytree = 0.283))))

    greedy_ensemble <- caretEnsemble(modelList, metric = "RMSE",
        trControl = trainControl(number = 25))

    return(greedy_ensemble)
}
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1539             nan     0.0500    0.0122
##      2        0.1428             nan     0.0500    0.0104
```

```
##      3        0.1332             nan     0.0500    0.0090
##      4        0.1245             nan     0.0500    0.0084
##      5        0.1160             nan     0.0500    0.0073
##      6        0.1082             nan     0.0500    0.0072
##      7        0.1015             nan     0.0500    0.0070
##      8        0.0948             nan     0.0500    0.0064
##      9        0.0888             nan     0.0500    0.0056
##     10        0.0833             nan     0.0500    0.0054
##     20        0.0479             nan     0.0500    0.0022
##     40        0.0226             nan     0.0500    0.0006
##     60        0.0150             nan     0.0500    0.0002
##     80        0.0117             nan     0.0500    0.0000
##    100        0.0100             nan     0.0500    0.0001
##    120        0.0087             nan     0.0500   -0.0000
##    140        0.0077             nan     0.0500    0.0000
##    160        0.0070             nan     0.0500   -0.0000
##    180        0.0065             nan     0.0500   -0.0000
##    200        0.0059             nan     0.0500   -0.0000
##    220        0.0055             nan     0.0500   -0.0000
##    240        0.0052             nan     0.0500   -0.0000
##    260        0.0048             nan     0.0500   -0.0000
##    280        0.0045             nan     0.0500   -0.0000
##    300        0.0042             nan     0.0500   -0.0000
##    320        0.0040             nan     0.0500   -0.0000
##    340        0.0037             nan     0.0500   -0.0000
##    360        0.0035             nan     0.0500   -0.0000
##    380        0.0033             nan     0.0500   -0.0000
##    400        0.0032             nan     0.0500   -0.0000
##    420        0.0030             nan     0.0500   -0.0000
##    440        0.0028             nan     0.0500   -0.0000
##    460        0.0027             nan     0.0500   -0.0000
##    480        0.0025             nan     0.0500   -0.0000
##    500        0.0024             nan     0.0500   -0.0000
##    520        0.0023             nan     0.0500   -0.0000
##    540        0.0022             nan     0.0500   -0.0000
##    560        0.0021             nan     0.0500   -0.0000
##    580        0.0020             nan     0.0500   -0.0000
##    600        0.0019             nan     0.0500   -0.0000
##    620        0.0018             nan     0.0500   -0.0000
##    640        0.0017             nan     0.0500   -0.0000
##    660        0.0016             nan     0.0500   -0.0000
##    680        0.0016             nan     0.0500   -0.0000
##    700        0.0015             nan     0.0500   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1423             nan     0.0500    0.0109
##      2        0.1320             nan     0.0500    0.0100
##      3        0.1229             nan     0.0500    0.0082
##      4        0.1142             nan     0.0500    0.0088
##      5        0.1068             nan     0.0500    0.0075
##      6        0.1001             nan     0.0500    0.0067
##      7        0.0936             nan     0.0500    0.0062
##      8        0.0877             nan     0.0500    0.0055
##      9        0.0824             nan     0.0500    0.0053
```

```
##     10        0.0776             nan     0.0500    0.0047
##     20        0.0458             nan     0.0500    0.0018
##     40        0.0221             nan     0.0500    0.0005
##     60        0.0147             nan     0.0500    0.0001
##     80        0.0114             nan     0.0500    0.0001
##    100        0.0096             nan     0.0500    0.0000
##    120        0.0084             nan     0.0500    0.0000
##    140        0.0074             nan     0.0500   -0.0000
##    160        0.0068             nan     0.0500    0.0000
##    180        0.0062             nan     0.0500   -0.0000
##    200        0.0058             nan     0.0500   -0.0000
##    220        0.0053             nan     0.0500   -0.0000
##    240        0.0049             nan     0.0500   -0.0000
##    260        0.0046             nan     0.0500   -0.0000
##    280        0.0043             nan     0.0500   -0.0000
##    300        0.0040             nan     0.0500   -0.0000
##    320        0.0037             nan     0.0500   -0.0000
##    340        0.0035             nan     0.0500    0.0000
##    360        0.0033             nan     0.0500   -0.0000
##    380        0.0031             nan     0.0500    0.0000
##    400        0.0029             nan     0.0500   -0.0000
##    420        0.0027             nan     0.0500   -0.0000
##    440        0.0025             nan     0.0500   -0.0000
##    460        0.0024             nan     0.0500    0.0000
##    480        0.0023             nan     0.0500   -0.0000
##    500        0.0022             nan     0.0500   -0.0000
##    520        0.0020             nan     0.0500   -0.0000
##    540        0.0019             nan     0.0500   -0.0000
##    560        0.0018             nan     0.0500   -0.0000
##    580        0.0017             nan     0.0500   -0.0000
##    600        0.0017             nan     0.0500   -0.0000
##    620        0.0016             nan     0.0500   -0.0000
##    640        0.0015             nan     0.0500   -0.0000
##    660        0.0014             nan     0.0500   -0.0000
##    680        0.0013             nan     0.0500   -0.0000
##    700        0.0013             nan     0.0500   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1505             nan     0.0500    0.0128
##      2        0.1399             nan     0.0500    0.0100
##      3        0.1305             nan     0.0500    0.0087
##      4        0.1218             nan     0.0500    0.0092
##      5        0.1136             nan     0.0500    0.0078
##      6        0.1060             nan     0.0500    0.0075
##      7        0.0986             nan     0.0500    0.0070
##      8        0.0924             nan     0.0500    0.0059
##      9        0.0867             nan     0.0500    0.0059
##     10        0.0815             nan     0.0500    0.0053
##     20        0.0475             nan     0.0500    0.0024
##     40        0.0216             nan     0.0500    0.0006
##     60        0.0138             nan     0.0500    0.0002
##     80        0.0105             nan     0.0500    0.0000
##    100        0.0088             nan     0.0500    0.0000
##    120        0.0076             nan     0.0500    0.0000
```

```
##    140        0.0067             nan     0.0500    0.0000
##    160        0.0060             nan     0.0500   -0.0000
##    180        0.0055             nan     0.0500    0.0000
##    200        0.0050             nan     0.0500   -0.0000
##    220        0.0046             nan     0.0500   -0.0000
##    240        0.0043             nan     0.0500   -0.0000
##    260        0.0040             nan     0.0500    0.0000
##    280        0.0037             nan     0.0500    0.0000
##    300        0.0034             nan     0.0500   -0.0000
##    320        0.0032             nan     0.0500   -0.0000
##    340        0.0030             nan     0.0500   -0.0000
##    360        0.0028             nan     0.0500   -0.0000
##    380        0.0026             nan     0.0500   -0.0000
##    400        0.0025             nan     0.0500   -0.0000
##    420        0.0023             nan     0.0500   -0.0000
##    440        0.0022             nan     0.0500   -0.0000
##    460        0.0021             nan     0.0500   -0.0000
##    480        0.0020             nan     0.0500   -0.0000
##    500        0.0019             nan     0.0500   -0.0000
##    520        0.0018             nan     0.0500   -0.0000
##    540        0.0017             nan     0.0500   -0.0000
##    560        0.0016             nan     0.0500   -0.0000
##    580        0.0015             nan     0.0500   -0.0000
##    600        0.0015             nan     0.0500   -0.0000
##    620        0.0014             nan     0.0500   -0.0000
##    640        0.0013             nan     0.0500   -0.0000
##    660        0.0013             nan     0.0500   -0.0000
##    680        0.0012             nan     0.0500   -0.0000
##    700        0.0012             nan     0.0500   -0.0000

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 8: Utilities has no variation.

## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1487             nan     0.0500    0.0108
##      2        0.1379             nan     0.0500    0.0105
##      3        0.1280             nan     0.0500    0.0097
##      4        0.1192             nan     0.0500    0.0081
##      5        0.1113             nan     0.0500    0.0080
##      6        0.1039             nan     0.0500    0.0071
##      7        0.0972             nan     0.0500    0.0061
##      8        0.0908             nan     0.0500    0.0060
##      9        0.0852             nan     0.0500    0.0054
##     10        0.0800             nan     0.0500    0.0049
##     20        0.0462             nan     0.0500    0.0020
##     40        0.0210             nan     0.0500    0.0004
##     60        0.0135             nan     0.0500    0.0002
##     80        0.0103             nan     0.0500    0.0001
##    100        0.0086             nan     0.0500    0.0000
##    120        0.0075             nan     0.0500   -0.0000
##    140        0.0067             nan     0.0500   -0.0000
##    160        0.0060             nan     0.0500    0.0000
##    180        0.0055             nan     0.0500   -0.0000
##    200        0.0050             nan     0.0500   -0.0000
##    220        0.0046             nan     0.0500   -0.0000
```

```
##    240        0.0043             nan     0.0500   -0.0000
##    260        0.0040             nan     0.0500   -0.0000
##    280        0.0037             nan     0.0500   -0.0000
##    300        0.0035             nan     0.0500   -0.0000
##    320        0.0032             nan     0.0500   -0.0000
##    340        0.0031             nan     0.0500   -0.0000
##    360        0.0028             nan     0.0500   -0.0000
##    380        0.0027             nan     0.0500   -0.0000
##    400        0.0025             nan     0.0500    0.0000
##    420        0.0023             nan     0.0500   -0.0000
##    440        0.0022             nan     0.0500   -0.0000
##    460        0.0021             nan     0.0500   -0.0000
##    480        0.0020             nan     0.0500   -0.0000
##    500        0.0019             nan     0.0500   -0.0000
##    520        0.0018             nan     0.0500   -0.0000
##    540        0.0017             nan     0.0500   -0.0000
##    560        0.0016             nan     0.0500   -0.0000
##    580        0.0015             nan     0.0500   -0.0000
##    600        0.0014             nan     0.0500   -0.0000
##    620        0.0013             nan     0.0500   -0.0000
##    640        0.0013             nan     0.0500   -0.0000
##    660        0.0012             nan     0.0500   -0.0000
##    680        0.0012             nan     0.0500   -0.0000
##    700        0.0011             nan     0.0500   -0.0000

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 8: Utilities has no variation.

## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1507             nan     0.0500    0.0122
##      2        0.1406             nan     0.0500    0.0101
##      3        0.1311             nan     0.0500    0.0097
##      4        0.1227             nan     0.0500    0.0086
##      5        0.1136             nan     0.0500    0.0076
##      6        0.1062             nan     0.0500    0.0073
##      7        0.0992             nan     0.0500    0.0065
##      8        0.0926             nan     0.0500    0.0062
##      9        0.0868             nan     0.0500    0.0055
##     10        0.0811             nan     0.0500    0.0057
##     20        0.0453             nan     0.0500    0.0019
##     40        0.0207             nan     0.0500    0.0005
##     60        0.0133             nan     0.0500    0.0002
##     80        0.0102             nan     0.0500    0.0000
##    100        0.0084             nan     0.0500    0.0001
##    120        0.0074             nan     0.0500   -0.0000
##    140        0.0065             nan     0.0500   -0.0000
##    160        0.0060             nan     0.0500   -0.0000
##    180        0.0055             nan     0.0500   -0.0000
##    200        0.0051             nan     0.0500   -0.0000
##    220        0.0047             nan     0.0500   -0.0000
##    240        0.0044             nan     0.0500    0.0000
##    260        0.0042             nan     0.0500   -0.0000
##    280        0.0039             nan     0.0500   -0.0000
##    300        0.0037             nan     0.0500   -0.0000
##    320        0.0035             nan     0.0500   -0.0000
```

```
##    340        0.0033             nan     0.0500   -0.0000
##    360        0.0031             nan     0.0500   -0.0000
##    380        0.0029             nan     0.0500   -0.0000
##    400        0.0028             nan     0.0500   -0.0000
##    420        0.0026             nan     0.0500   -0.0000
##    440        0.0025             nan     0.0500   -0.0000
##    460        0.0023             nan     0.0500   -0.0000
##    480        0.0022             nan     0.0500   -0.0000
##    500        0.0021             nan     0.0500   -0.0000
##    520        0.0020             nan     0.0500   -0.0000
##    540        0.0019             nan     0.0500   -0.0000
##    560        0.0018             nan     0.0500   -0.0000
##    580        0.0018             nan     0.0500   -0.0000
##    600        0.0017             nan     0.0500   -0.0000
##    620        0.0016             nan     0.0500   -0.0000
##    640        0.0015             nan     0.0500   -0.0000
##    660        0.0015             nan     0.0500   -0.0000
##    680        0.0014             nan     0.0500   -0.0000
##    700        0.0013             nan     0.0500   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1501             nan     0.0500    0.0109
##      2        0.1396             nan     0.0500    0.0113
##      3        0.1299             nan     0.0500    0.0095
##      4        0.1214             nan     0.0500    0.0080
##      5        0.1132             nan     0.0500    0.0080
##      6        0.1057             nan     0.0500    0.0070
##      7        0.0987             nan     0.0500    0.0071
##      8        0.0922             nan     0.0500    0.0058
##      9        0.0862             nan     0.0500    0.0054
##     10        0.0812             nan     0.0500    0.0052
##     20        0.0471             nan     0.0500    0.0021
##     40        0.0226             nan     0.0500    0.0004
##     60        0.0150             nan     0.0500    0.0001
##     80        0.0117             nan     0.0500    0.0000
##    100        0.0098             nan     0.0500    0.0000
##    120        0.0087             nan     0.0500    0.0000
##    140        0.0078             nan     0.0500   -0.0000
##    160        0.0070             nan     0.0500    0.0000
##    180        0.0065             nan     0.0500   -0.0000
##    200        0.0060             nan     0.0500   -0.0000
##    220        0.0056             nan     0.0500   -0.0000
##    240        0.0052             nan     0.0500   -0.0000
##    260        0.0048             nan     0.0500   -0.0000
##    280        0.0045             nan     0.0500   -0.0000
##    300        0.0042             nan     0.0500   -0.0000
##    320        0.0040             nan     0.0500   -0.0000
##    340        0.0038             nan     0.0500   -0.0000
##    360        0.0036             nan     0.0500   -0.0000
##    380        0.0034             nan     0.0500   -0.0000
##    400        0.0032             nan     0.0500   -0.0000
##    420        0.0030             nan     0.0500   -0.0000
##    440        0.0028             nan     0.0500   -0.0000
##    460        0.0027             nan     0.0500   -0.0000
```

```
##    480        0.0025             nan     0.0500   -0.0000
##    500        0.0024             nan     0.0500    0.0000
##    520        0.0023             nan     0.0500   -0.0000
##    540        0.0022             nan     0.0500   -0.0000
##    560        0.0021             nan     0.0500   -0.0000
##    580        0.0020             nan     0.0500   -0.0000
##    600        0.0019             nan     0.0500   -0.0000
##    620        0.0018             nan     0.0500   -0.0000
##    640        0.0017             nan     0.0500   -0.0000
##    660        0.0016             nan     0.0500   -0.0000
##    680        0.0016             nan     0.0500   -0.0000
##    700        0.0015             nan     0.0500   -0.0000
```

```
## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 8: Utilities has no variation.
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1516             nan     0.0500    0.0110
##      2        0.1411             nan     0.0500    0.0103
##      3        0.1316             nan     0.0500    0.0087
##      4        0.1231             nan     0.0500    0.0087
##      5        0.1154             nan     0.0500    0.0078
##      6        0.1078             nan     0.0500    0.0074
##      7        0.1009             nan     0.0500    0.0064
##      8        0.0944             nan     0.0500    0.0061
##      9        0.0886             nan     0.0500    0.0054
##     10        0.0833             nan     0.0500    0.0052
##     20        0.0483             nan     0.0500    0.0023
##     40        0.0228             nan     0.0500    0.0005
##     60        0.0148             nan     0.0500    0.0001
##     80        0.0115             nan     0.0500    0.0000
##    100        0.0095             nan     0.0500    0.0000
##    120        0.0083             nan     0.0500    0.0000
##    140        0.0074             nan     0.0500    0.0000
##    160        0.0067             nan     0.0500    0.0000
##    180        0.0061             nan     0.0500   -0.0000
##    200        0.0056             nan     0.0500   -0.0000
##    220        0.0052             nan     0.0500   -0.0000
##    240        0.0048             nan     0.0500   -0.0000
##    260        0.0045             nan     0.0500   -0.0000
##    280        0.0042             nan     0.0500    0.0000
##    300        0.0039             nan     0.0500   -0.0000
##    320        0.0037             nan     0.0500   -0.0000
##    340        0.0035             nan     0.0500    0.0000
##    360        0.0033             nan     0.0500   -0.0000
##    380        0.0031             nan     0.0500   -0.0000
##    400        0.0029             nan     0.0500   -0.0000
##    420        0.0028             nan     0.0500   -0.0000
##    440        0.0026             nan     0.0500   -0.0000
##    460        0.0025             nan     0.0500   -0.0000
##    480        0.0023             nan     0.0500   -0.0000
##    500        0.0022             nan     0.0500   -0.0000
##    520        0.0021             nan     0.0500   -0.0000
##    540        0.0020             nan     0.0500   -0.0000
##    560        0.0019             nan     0.0500   -0.0000
```

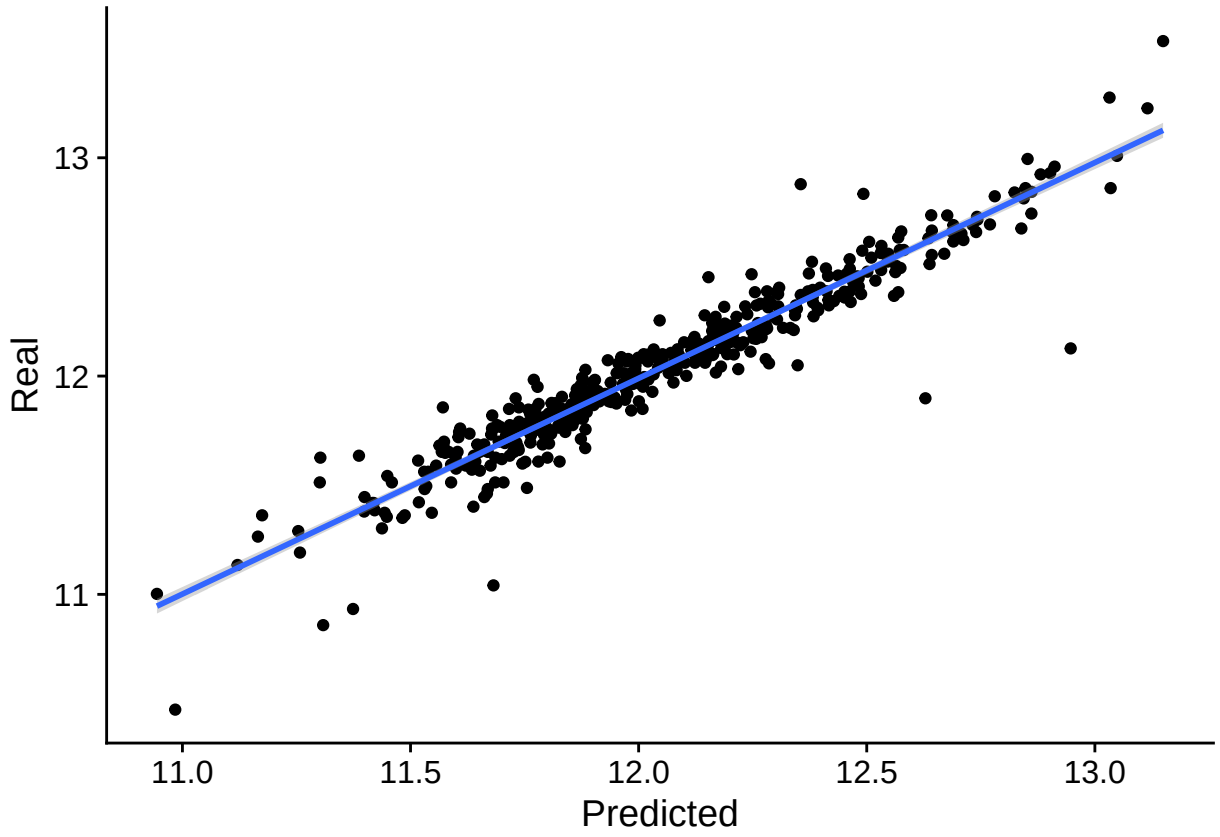
```
##    580        0.0018             nan     0.0500   -0.0000
##    600        0.0017             nan     0.0500   -0.0000
##    620        0.0016             nan     0.0500   -0.0000
##    640        0.0016             nan     0.0500   -0.0000
##    660        0.0015             nan     0.0500   -0.0000
##    680        0.0014             nan     0.0500   -0.0000
##    700        0.0013             nan     0.0500   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1506             nan     0.0500    0.0110
##      2        0.1407             nan     0.0500    0.0097
##      3        0.1308             nan     0.0500    0.0092
##      4        0.1222             nan     0.0500    0.0085
##      5        0.1144             nan     0.0500    0.0075
##      6        0.1071             nan     0.0500    0.0066
##      7        0.1002             nan     0.0500    0.0064
##      8        0.0939             nan     0.0500    0.0062
##      9        0.0885             nan     0.0500    0.0049
##     10        0.0831             nan     0.0500    0.0047
##     20        0.0482             nan     0.0500    0.0021
##     40        0.0239             nan     0.0500    0.0005
##     60        0.0168             nan     0.0500    0.0002
##     80        0.0136             nan     0.0500    0.0000
##    100        0.0119             nan     0.0500   -0.0000
##    120        0.0106             nan     0.0500    0.0000
##    140        0.0098             nan     0.0500   -0.0000
##    160        0.0091             nan     0.0500   -0.0000
##    180        0.0086             nan     0.0500   -0.0000
##    200        0.0081             nan     0.0500   -0.0000
##    220        0.0077             nan     0.0500   -0.0000
##    240        0.0073             nan     0.0500   -0.0000
##    260        0.0070             nan     0.0500   -0.0000
##    280        0.0066             nan     0.0500   -0.0000
##    300        0.0064             nan     0.0500   -0.0000
##    320        0.0061             nan     0.0500   -0.0000
##    340        0.0058             nan     0.0500   -0.0000
##    360        0.0056             nan     0.0500   -0.0000
##    380        0.0054             nan     0.0500   -0.0000
##    400        0.0052             nan     0.0500   -0.0000
##    420        0.0050             nan     0.0500   -0.0000
##    440        0.0048             nan     0.0500   -0.0000
##    460        0.0046             nan     0.0500   -0.0000
##    480        0.0044             nan     0.0500   -0.0000
##    500        0.0043             nan     0.0500   -0.0000
##    520        0.0041             nan     0.0500   -0.0000
##    540        0.0039             nan     0.0500   -0.0000
##    560        0.0038             nan     0.0500   -0.0000
##    580        0.0037             nan     0.0500   -0.0000
##    600        0.0035             nan     0.0500   -0.0000
##    620        0.0034             nan     0.0500   -0.0000
##    640        0.0033             nan     0.0500   -0.0000
##    660        0.0032             nan     0.0500   -0.0000
##    680        0.0031             nan     0.0500   -0.0000
##    700        0.0029             nan     0.0500   -0.0000
```

```
model.ensemble  # 0.130408 # 0.1307172
```

```
## A glm ensemble of 2 base models: gbm, xgbTree
##
## Ensemble results:
## Generalized Linear Model
##
## 2641 samples
##    2 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 2641, 2641, 2641, 2641, 2641, 2641, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.1267544  0.8998718  0.08639417
```

```
predictAndEvaluate(model.ensemble, train.processed.partition.validation)  # 0.1217003 # 0.1296369
```

```
## Warning: Ignoring unknown parameters: method
```




```
## [1] 0.1181355
```

Full train

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1541             nan     0.0500    0.0128
##      2        0.1435             nan     0.0500    0.0105
##      3        0.1337             nan     0.0500    0.0096
##      4        0.1248             nan     0.0500    0.0083
##      5        0.1167             nan     0.0500    0.0080
##      6        0.1093             nan     0.0500    0.0067
##      7        0.1023             nan     0.0500    0.0068
##      8        0.0958             nan     0.0500    0.0061
##      9        0.0899             nan     0.0500    0.0059
##     10        0.0843             nan     0.0500    0.0055
##     20        0.0477             nan     0.0500    0.0024
##     40        0.0225             nan     0.0500    0.0006
##     60        0.0149             nan     0.0500    0.0002
##     80        0.0117             nan     0.0500    0.0001
##    100        0.0099             nan     0.0500    0.0000
##    120        0.0087             nan     0.0500   -0.0000
##    140        0.0080             nan     0.0500   -0.0000
##    160        0.0074             nan     0.0500   -0.0000
##    180        0.0069             nan     0.0500   -0.0000
##    200        0.0065             nan     0.0500   -0.0000
##    220        0.0061             nan     0.0500    0.0000
##    240        0.0056             nan     0.0500    0.0000
##    260        0.0053             nan     0.0500   -0.0000
##    280        0.0050             nan     0.0500   -0.0000
##    300        0.0047             nan     0.0500   -0.0000
##    320        0.0045             nan     0.0500   -0.0000
##    340        0.0043             nan     0.0500   -0.0000
##    360        0.0041             nan     0.0500   -0.0000
##    380        0.0039             nan     0.0500   -0.0000
##    400        0.0037             nan     0.0500   -0.0000
##    420        0.0036             nan     0.0500   -0.0000
##    440        0.0034             nan     0.0500   -0.0000
##    460        0.0033             nan     0.0500   -0.0000
##    480        0.0031             nan     0.0500   -0.0000
##    500        0.0030             nan     0.0500   -0.0000
##    520        0.0029             nan     0.0500   -0.0000
##    540        0.0028             nan     0.0500   -0.0000
##    560        0.0026             nan     0.0500   -0.0000
##    580        0.0025             nan     0.0500   -0.0000
##    600        0.0024             nan     0.0500   -0.0000
##    620        0.0023             nan     0.0500   -0.0000
##    640        0.0022             nan     0.0500   -0.0000
##    660        0.0022             nan     0.0500   -0.0000
##    680        0.0021             nan     0.0500    0.0000
##    700        0.0020             nan     0.0500   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1487             nan     0.0500    0.0113
##      2        0.1382             nan     0.0500    0.0102
##      3        0.1287             nan     0.0500    0.0100
##      4        0.1201             nan     0.0500    0.0081
```

```
##      5        0.1124             nan     0.0500    0.0077
##      6        0.1052             nan     0.0500    0.0069
##      7        0.0984             nan     0.0500    0.0066
##      8        0.0924             nan     0.0500    0.0055
##      9        0.0868             nan     0.0500    0.0052
##     10        0.0815             nan     0.0500    0.0047
##     20        0.0478             nan     0.0500    0.0022
##     40        0.0229             nan     0.0500    0.0005
##     60        0.0151             nan     0.0500    0.0002
##     80        0.0119             nan     0.0500    0.0001
##    100        0.0101             nan     0.0500    0.0000
##    120        0.0091             nan     0.0500   -0.0000
##    140        0.0083             nan     0.0500    0.0000
##    160        0.0077             nan     0.0500    0.0000
##    180        0.0071             nan     0.0500   -0.0000
##    200        0.0066             nan     0.0500   -0.0000
##    220        0.0062             nan     0.0500   -0.0000
##    240        0.0058             nan     0.0500   -0.0000
##    260        0.0055             nan     0.0500   -0.0000
##    280        0.0052             nan     0.0500   -0.0000
##    300        0.0049             nan     0.0500   -0.0000
##    320        0.0047             nan     0.0500   -0.0000
##    340        0.0044             nan     0.0500   -0.0000
##    360        0.0042             nan     0.0500   -0.0000
##    380        0.0040             nan     0.0500   -0.0000
##    400        0.0038             nan     0.0500   -0.0000
##    420        0.0036             nan     0.0500    0.0000
##    440        0.0034             nan     0.0500   -0.0000
##    460        0.0033             nan     0.0500   -0.0000
##    480        0.0032             nan     0.0500   -0.0000
##    500        0.0030             nan     0.0500   -0.0000
##    520        0.0029             nan     0.0500   -0.0000
##    540        0.0028             nan     0.0500   -0.0000
##    560        0.0027             nan     0.0500   -0.0000
##    580        0.0026             nan     0.0500   -0.0000
##    600        0.0025             nan     0.0500   -0.0000
##    620        0.0024             nan     0.0500   -0.0000
##    640        0.0023             nan     0.0500   -0.0000
##    660        0.0022             nan     0.0500   -0.0000
##    680        0.0021             nan     0.0500   -0.0000
##    700        0.0020             nan     0.0500   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1574             nan     0.0500    0.0121
##      2        0.1463             nan     0.0500    0.0111
##      3        0.1359             nan     0.0500    0.0095
##      4        0.1268             nan     0.0500    0.0090
##      5        0.1184             nan     0.0500    0.0080
##      6        0.1102             nan     0.0500    0.0069
##      7        0.1029             nan     0.0500    0.0074
##      8        0.0966             nan     0.0500    0.0059
##      9        0.0903             nan     0.0500    0.0059
##     10        0.0852             nan     0.0500    0.0049
##     20        0.0492             nan     0.0500    0.0023
```

```
##     40        0.0241             nan     0.0500    0.0005
##     60        0.0161             nan     0.0500    0.0002
##     80        0.0128             nan     0.0500    0.0001
##    100        0.0109             nan     0.0500   -0.0000
##    120        0.0097             nan     0.0500    0.0000
##    140        0.0089             nan     0.0500    0.0000
##    160        0.0081             nan     0.0500    0.0000
##    180        0.0075             nan     0.0500    0.0000
##    200        0.0070             nan     0.0500   -0.0000
##    220        0.0066             nan     0.0500    0.0000
##    240        0.0062             nan     0.0500   -0.0000
##    260        0.0059             nan     0.0500   -0.0000
##    280        0.0056             nan     0.0500   -0.0000
##    300        0.0053             nan     0.0500   -0.0000
##    320        0.0050             nan     0.0500   -0.0000
##    340        0.0047             nan     0.0500   -0.0000
##    360        0.0045             nan     0.0500   -0.0000
##    380        0.0043             nan     0.0500   -0.0000
##    400        0.0041             nan     0.0500   -0.0000
##    420        0.0039             nan     0.0500   -0.0000
##    440        0.0038             nan     0.0500   -0.0000
##    460        0.0036             nan     0.0500   -0.0000
##    480        0.0035             nan     0.0500   -0.0000
##    500        0.0033             nan     0.0500   -0.0000
##    520        0.0032             nan     0.0500   -0.0000
##    540        0.0031             nan     0.0500   -0.0000
##    560        0.0030             nan     0.0500   -0.0000
##    580        0.0029             nan     0.0500   -0.0000
##    600        0.0027             nan     0.0500   -0.0000
##    620        0.0026             nan     0.0500   -0.0000
##    640        0.0025             nan     0.0500   -0.0000
##    660        0.0025             nan     0.0500   -0.0000
##    680        0.0024             nan     0.0500   -0.0000
##    700        0.0023             nan     0.0500   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1627             nan     0.0500    0.0119
##      2        0.1514             nan     0.0500    0.0112
##      3        0.1412             nan     0.0500    0.0101
##      4        0.1319             nan     0.0500    0.0092
##      5        0.1234             nan     0.0500    0.0081
##      6        0.1157             nan     0.0500    0.0071
##      7        0.1083             nan     0.0500    0.0071
##      8        0.1015             nan     0.0500    0.0069
##      9        0.0954             nan     0.0500    0.0060
##     10        0.0897             nan     0.0500    0.0055
##     20        0.0517             nan     0.0500    0.0023
##     40        0.0247             nan     0.0500    0.0007
##     60        0.0163             nan     0.0500    0.0002
##     80        0.0126             nan     0.0500    0.0001
##    100        0.0106             nan     0.0500   -0.0000
##    120        0.0093             nan     0.0500    0.0000
##    140        0.0084             nan     0.0500    0.0000
##    160        0.0076             nan     0.0500   -0.0000
```

```
##    180        0.0070             nan     0.0500   -0.0000
##    200        0.0064             nan     0.0500   -0.0000
##    220        0.0060             nan     0.0500   -0.0000
##    240        0.0056             nan     0.0500   -0.0000
##    260        0.0053             nan     0.0500   -0.0000
##    280        0.0050             nan     0.0500   -0.0000
##    300        0.0047             nan     0.0500   -0.0000
##    320        0.0045             nan     0.0500   -0.0000
##    340        0.0042             nan     0.0500   -0.0000
##    360        0.0040             nan     0.0500    0.0000
##    380        0.0038             nan     0.0500   -0.0000
##    400        0.0037             nan     0.0500    0.0000
##    420        0.0035             nan     0.0500   -0.0000
##    440        0.0033             nan     0.0500   -0.0000
##    460        0.0032             nan     0.0500   -0.0000
##    480        0.0031             nan     0.0500   -0.0000
##    500        0.0029             nan     0.0500   -0.0000
##    520        0.0028             nan     0.0500   -0.0000
##    540        0.0027             nan     0.0500   -0.0000
##    560        0.0026             nan     0.0500   -0.0000
##    580        0.0025             nan     0.0500   -0.0000
##    600        0.0024             nan     0.0500   -0.0000
##    620        0.0023             nan     0.0500   -0.0000
##    640        0.0022             nan     0.0500   -0.0000
##    660        0.0021             nan     0.0500   -0.0000
##    680        0.0021             nan     0.0500   -0.0000
##    700        0.0020             nan     0.0500   -0.0000

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =
## "bernoulli", : variable 8: Utilities has no variation.

## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1549             nan     0.0500    0.0106
##      2        0.1438             nan     0.0500    0.0103
##      3        0.1341             nan     0.0500    0.0094
##      4        0.1251             nan     0.0500    0.0091
##      5        0.1169             nan     0.0500    0.0081
##      6        0.1098             nan     0.0500    0.0073
##      7        0.1032             nan     0.0500    0.0067
##      8        0.0967             nan     0.0500    0.0066
##      9        0.0907             nan     0.0500    0.0060
##     10        0.0853             nan     0.0500    0.0053
##     20        0.0495             nan     0.0500    0.0023
##     40        0.0243             nan     0.0500    0.0006
##     60        0.0165             nan     0.0500    0.0002
##     80        0.0130             nan     0.0500    0.0001
##    100        0.0111             nan     0.0500    0.0000
##    120        0.0098             nan     0.0500    0.0000
##    140        0.0088             nan     0.0500    0.0000
##    160        0.0080             nan     0.0500    0.0000
##    180        0.0074             nan     0.0500   -0.0000
##    200        0.0068             nan     0.0500   -0.0000
##    220        0.0064             nan     0.0500   -0.0000
##    240        0.0060             nan     0.0500   -0.0000
##    260        0.0056             nan     0.0500   -0.0000
```

```
##    280        0.0053             nan     0.0500   -0.0000
##    300        0.0050             nan     0.0500   -0.0000
##    320        0.0047             nan     0.0500   -0.0000
##    340        0.0044             nan     0.0500   -0.0000
##    360        0.0042             nan     0.0500   -0.0000
##    380        0.0040             nan     0.0500   -0.0000
##    400        0.0038             nan     0.0500    0.0000
##    420        0.0037             nan     0.0500   -0.0000
##    440        0.0035             nan     0.0500   -0.0000
##    460        0.0033             nan     0.0500   -0.0000
##    480        0.0032             nan     0.0500   -0.0000
##    500        0.0031             nan     0.0500   -0.0000
##    520        0.0030             nan     0.0500   -0.0000
##    540        0.0029             nan     0.0500   -0.0000
##    560        0.0028             nan     0.0500    0.0000
##    580        0.0027             nan     0.0500   -0.0000
##    600        0.0026             nan     0.0500   -0.0000
##    620        0.0025             nan     0.0500   -0.0000
##    640        0.0024             nan     0.0500   -0.0000
##    660        0.0023             nan     0.0500   -0.0000
##    680        0.0022             nan     0.0500   -0.0000
##    700        0.0021             nan     0.0500   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1427             nan     0.0500    0.0109
##      2        0.1325             nan     0.0500    0.0098
##      3        0.1233             nan     0.0500    0.0094
##      4        0.1152             nan     0.0500    0.0077
##      5        0.1075             nan     0.0500    0.0073
##      6        0.1005             nan     0.0500    0.0074
##      7        0.0942             nan     0.0500    0.0062
##      8        0.0882             nan     0.0500    0.0051
##      9        0.0825             nan     0.0500    0.0053
##     10        0.0774             nan     0.0500    0.0049
##     20        0.0448             nan     0.0500    0.0021
##     40        0.0210             nan     0.0500    0.0005
##     60        0.0137             nan     0.0500    0.0002
##     80        0.0106             nan     0.0500    0.0001
##    100        0.0090             nan     0.0500    0.0000
##    120        0.0079             nan     0.0500    0.0000
##    140        0.0072             nan     0.0500    0.0000
##    160        0.0066             nan     0.0500    0.0000
##    180        0.0061             nan     0.0500   -0.0000
##    200        0.0057             nan     0.0500   -0.0000
##    220        0.0053             nan     0.0500    0.0000
##    240        0.0050             nan     0.0500    0.0000
##    260        0.0048             nan     0.0500   -0.0000
##    280        0.0045             nan     0.0500   -0.0000
##    300        0.0043             nan     0.0500   -0.0000
##    320        0.0041             nan     0.0500   -0.0000
##    340        0.0039             nan     0.0500    0.0000
##    360        0.0037             nan     0.0500   -0.0000
##    380        0.0036             nan     0.0500   -0.0000
##    400        0.0034             nan     0.0500   -0.0000
```

```
##    420        0.0033             nan     0.0500   -0.0000
##    440        0.0031             nan     0.0500   -0.0000
##    460        0.0030             nan     0.0500   -0.0000
##    480        0.0029             nan     0.0500   -0.0000
##    500        0.0028             nan     0.0500   -0.0000
##    520        0.0026             nan     0.0500   -0.0000
##    540        0.0025             nan     0.0500   -0.0000
##    560        0.0024             nan     0.0500   -0.0000
##    580        0.0023             nan     0.0500   -0.0000
##    600        0.0022             nan     0.0500   -0.0000
##    620        0.0022             nan     0.0500   -0.0000
##    640        0.0021             nan     0.0500   -0.0000
##    660        0.0020             nan     0.0500   -0.0000
##    680        0.0019             nan     0.0500    0.0000
##    700        0.0019             nan     0.0500   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1564             nan     0.0500    0.0119
##      2        0.1460             nan     0.0500    0.0107
##      3        0.1359             nan     0.0500    0.0107
##      4        0.1268             nan     0.0500    0.0091
##      5        0.1187             nan     0.0500    0.0077
##      6        0.1118             nan     0.0500    0.0069
##      7        0.1048             nan     0.0500    0.0063
##      8        0.0987             nan     0.0500    0.0061
##      9        0.0927             nan     0.0500    0.0061
##     10        0.0874             nan     0.0500    0.0051
##     20        0.0507             nan     0.0500    0.0027
##     40        0.0247             nan     0.0500    0.0005
##     60        0.0167             nan     0.0500    0.0002
##     80        0.0131             nan     0.0500    0.0001
##    100        0.0112             nan     0.0500    0.0000
##    120        0.0099             nan     0.0500   -0.0000
##    140        0.0089             nan     0.0500    0.0000
##    160        0.0081             nan     0.0500   -0.0000
##    180        0.0076             nan     0.0500   -0.0000
##    200        0.0070             nan     0.0500   -0.0000
##    220        0.0066             nan     0.0500    0.0000
##    240        0.0062             nan     0.0500   -0.0000
##    260        0.0058             nan     0.0500    0.0000
##    280        0.0055             nan     0.0500   -0.0000
##    300        0.0051             nan     0.0500   -0.0000
##    320        0.0049             nan     0.0500   -0.0000
##    340        0.0046             nan     0.0500   -0.0000
##    360        0.0044             nan     0.0500   -0.0000
##    380        0.0042             nan     0.0500   -0.0000
##    400        0.0039             nan     0.0500   -0.0000
##    420        0.0038             nan     0.0500   -0.0000
##    440        0.0036             nan     0.0500    0.0000
##    460        0.0034             nan     0.0500   -0.0000
##    480        0.0033             nan     0.0500   -0.0000
##    500        0.0031             nan     0.0500   -0.0000
##    520        0.0030             nan     0.0500    0.0000
##    540        0.0029             nan     0.0500   -0.0000
```

```
##    560        0.0028             nan     0.0500   -0.0000
##    580        0.0027             nan     0.0500   -0.0000
##    600        0.0026             nan     0.0500   -0.0000
##    620        0.0025             nan     0.0500   -0.0000
##    640        0.0024             nan     0.0500   -0.0000
##    660        0.0023             nan     0.0500   -0.0000
##    680        0.0022             nan     0.0500   -0.0000
##    700        0.0021             nan     0.0500   -0.0000
##
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        0.1480             nan     0.0500    0.0115
##      2        0.1377             nan     0.0500    0.0096
##      3        0.1286             nan     0.0500    0.0087
##      4        0.1205             nan     0.0500    0.0085
##      5        0.1134             nan     0.0500    0.0066
##      6        0.1059             nan     0.0500    0.0069
##      7        0.0991             nan     0.0500    0.0070
##      8        0.0933             nan     0.0500    0.0057
##      9        0.0880             nan     0.0500    0.0050
##     10        0.0832             nan     0.0500    0.0047
##     20        0.0498             nan     0.0500    0.0020
##     40        0.0258             nan     0.0500    0.0005
##     60        0.0183             nan     0.0500    0.0001
##     80        0.0149             nan     0.0500    0.0001
##    100        0.0130             nan     0.0500    0.0000
##    120        0.0118             nan     0.0500   -0.0000
##    140        0.0109             nan     0.0500   -0.0000
##    160        0.0102             nan     0.0500   -0.0000
##    180        0.0096             nan     0.0500   -0.0000
##    200        0.0092             nan     0.0500   -0.0000
##    220        0.0088             nan     0.0500   -0.0000
##    240        0.0083             nan     0.0500   -0.0000
##    260        0.0080             nan     0.0500   -0.0000
##    280        0.0077             nan     0.0500   -0.0000
##    300        0.0075             nan     0.0500   -0.0000
##    320        0.0072             nan     0.0500   -0.0000
##    340        0.0069             nan     0.0500   -0.0000
##    360        0.0067             nan     0.0500   -0.0000
##    380        0.0065             nan     0.0500   -0.0000
##    400        0.0062             nan     0.0500   -0.0000
##    420        0.0060             nan     0.0500   -0.0000
##    440        0.0058             nan     0.0500   -0.0000
##    460        0.0056             nan     0.0500   -0.0000
##    480        0.0054             nan     0.0500   -0.0000
##    500        0.0053             nan     0.0500   -0.0000
##    520        0.0051             nan     0.0500   -0.0000
##    540        0.0050             nan     0.0500   -0.0000
##    560        0.0048             nan     0.0500   -0.0000
##    580        0.0047             nan     0.0500   -0.0000
##    600        0.0045             nan     0.0500   -0.0000
##    620        0.0044             nan     0.0500   -0.0000
##    640        0.0043             nan     0.0500   -0.0000
##    660        0.0042             nan     0.0500   -0.0000
##    680        0.0040             nan     0.0500   -0.0000
```

```
##    700        0.0039             nan     0.0500   -0.0000
```

```
model.full  # 0.1261884 # 0.1345461 # 0.1254348
```

```
## A glm ensemble of 2 base models: gbm, xgbTree
##
## Ensemble results:
## Generalized Linear Model
##
## 3736 samples
##    2 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 3736, 3736, 3736, 3736, 3736, 3736, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.1254348  0.8993666  0.08295416
```

Prediction and submit

```
predictions <- predict(model.full, newdata = test.processed)
prediction.table <- data.frame(Id = test$Id, SalePrice = exp(predictions))
write.csv(prediction.table, "prediction.csv", row.names = FALSE)  # Kaggle: 0.12246
```