Ames House Price prediction using tidymodel tools

Ravi Hela

22/03/2020

Load libraries

```
library(tidymodels)
library(readr)
library(tidyverse)
library(lubridate)
library(skimr)
library(DataExplorer)
library(tidyquant)
library(corrr)
library(caret)
library(vip)
library(Hmisc)
```

Define some useful functions

Caliing few function to help get correlation dataframe

```
#function to help filter corelated variables
source("correlation_df.R")
#get model metrics in a dataframe
source("model_metric_compare.R")
```

Read data

```
train <- read_csv("train.csv")
test <- read_csv("test.csv")

names(train) <- make.names(names(train))
names(test) <- make.names(names(test))
```

The "test" file is the one to be used for Kaggle submission. This will not beused for model training but is a good for EDA use. For trainiing model we crate a validation set from train file later on. For model training we call the validation set as testing set

Combinie test and train for EDA

Lets look at the summary of this combined dataset

Table 1: Data summary

Name	house_comb
Number of rows	2919
Number of columns	82
Column type frequency:	
character	44
numeric	38
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
MSZoning	4	1.00	2	7	0	5	0
Street	0	1.00	4	4	0	2	0
Alley	2721	0.07	4	4	0	2	0
LotShape	0	1.00	3	3	0	4	0
LandContour	0	1.00	3	3	0	4	0
Utilities	2	1.00	6	6	0	2	0
LotConfig	0	1.00	3	7	0	5	0
LandSlope	0	1.00	3	3	0	3	0
Neighborhood	0	1.00	5	7	0	25	0
Condition1	0	1.00	4	6	0	9	0
Condition2	0	1.00	4	6	0	8	0
BldgType	0	1.00	4	6	0	5	0
HouseStyle	0	1.00	4	6	0	8	0
RoofStyle	0	1.00	3	7	0	6	0
RoofMatl	0	1.00	4	7	0	8	0
Exterior1st	1	1.00	5	7	0	15	0
Exterior2nd	1	1.00	5	7	0	16	0
MasVnrType	24	0.99	4	7	0	4	0
ExterQual	0	1.00	2	2	0	4	0
ExterCond	0	1.00	2	2	0	5	0
Foundation	0	1.00	4	6	0	6	0
BsmtQual	81	0.97	2	2	0	4	0
BsmtCond	82	0.97	2	2	0	4	0
BsmtExposure	82	0.97	2	2	0	4	0
BsmtFinType1	79	0.97	3	3	0	6	0
BsmtFinType2	80	0.97	3	3	0	6	0
Heating	0	1.00	4	5	0	6	0
HeatingQC	0	1.00	2	2	0	5	0
CentralAir	0	1.00	1	1	0	2	0
Electrical	1	1.00	3	5	0	5	0
KitchenQual	1	1.00	2	2	0	4	0
Functional	2	1.00	3	4	0	7	0
FireplaceQu	1420	0.51	2	2	0	5	0
GarageType	157	0.95	6	7	0	6	0
GarageFinish	159	0.95	3	3	0	3	0
GarageQual	159	0.95	2	2	0	5	0
GarageCond	159	0.95	2	2	0	5	0
PavedDrive	0	1.00	1	1	0	3	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
PoolQC	2909	0.00	2	2	0	3	0
Fence	2348	0.20	4	5	0	4	0
MiscFeature	2814	0.04	4	4	0	4	0
SaleType	1	1.00	2	5	0	9	0
SaleCondition	0	1.00	6	7	0	6	0
set	0	1.00	4	5	0	2	0

Variable type: numeric

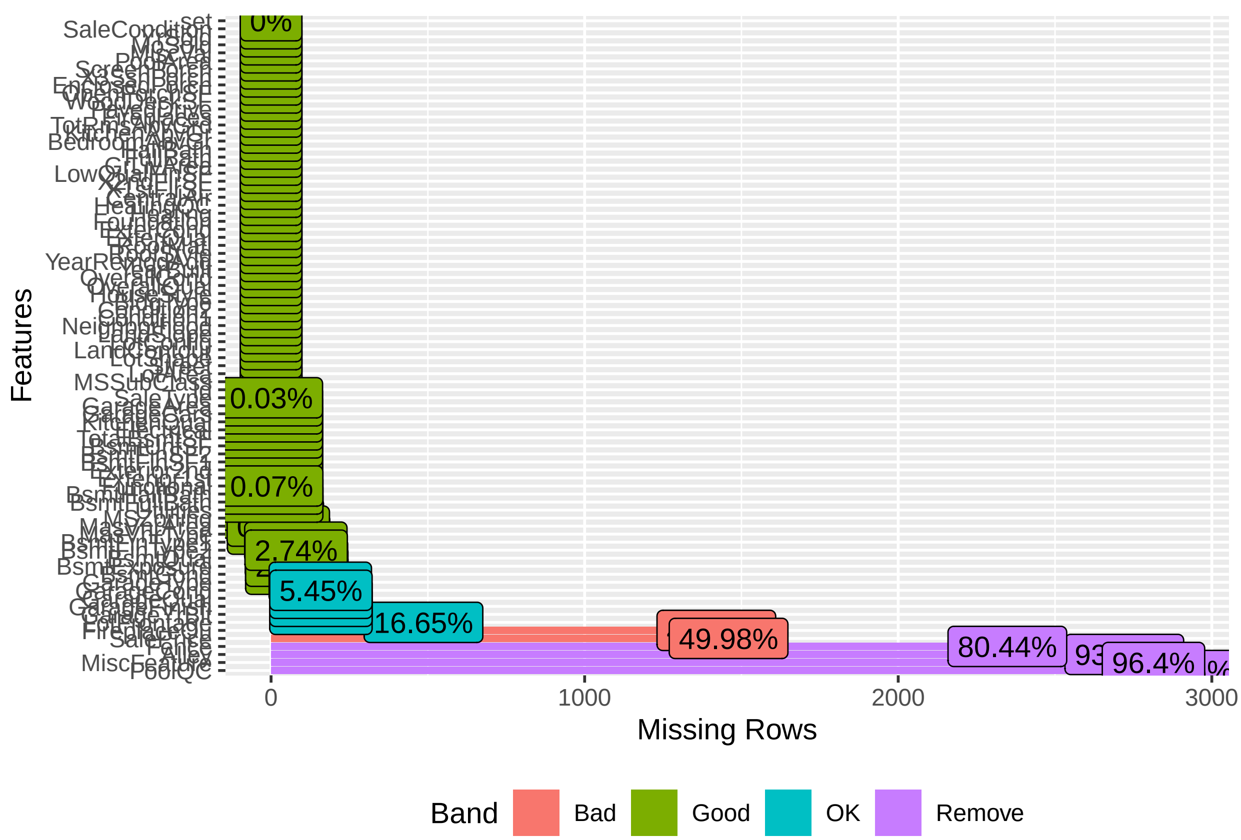
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1.00	1460.00	842.79	1	730.5	1460.0	2189.5	2919
MSSubClass	0	1.00	57.14	42.52	20	20.0	50.0	70.0	190
LotFrontage	486	0.83	69.31	23.34	21	59.0	68.0	80.0	313
LotArea	0	1.00	10168.11	7887.00	1300	7478.0	9453.0	11570.0	215245
OverallQual	0	1.00	6.09	1.41	1	5.0	6.0	7.0	10
OverallCond	0	1.00	5.56	1.11	1	5.0	5.0	6.0	9
YearBuilt	0	1.00	1971.31	30.29	1872	1953.5	1973.0	2001.0	2010
YearRemodAdd	0	1.00	1984.26	20.89	1950	1965.0	1993.0	2004.0	2010
MasVnrArea	23	0.99	102.20	179.33	0	0.0	0.0	164.0	1600
BsmtFinSF1	1	1.00	441.42	455.61	0	0.0	368.5	733.0	5644
BsmtFinSF2	1	1.00	49.58	169.21	0	0.0	0.0	0.0	1526
BsmtUnfSF	1	1.00	560.77	439.54	0	220.0	467.0	805.5	2336
TotalBsmtSF	1	1.00	1051.78	440.77	0	793.0	989.5	1302.0	6110
X1stFlrSF	0	1.00	1159.58	392.36	334	876.0	1082.0	1387.5	5095
X2ndFlrSF	0	1.00	336.48	428.70	0	0.0	0.0	704.0	2065
LowQualFinSF	0	1.00	4.69	46.40	0	0.0	0.0	0.0	1064
GrLivArea	0	1.00	1500.76	506.05	334	1126.0	1444.0	1743.5	5642
BsmtFullBath	2	1.00	0.43	0.52	0	0.0	0.0	1.0	3
BsmtHalfBath	2	1.00	0.06	0.25	0	0.0	0.0	0.0	2
FullBath	0	1.00	1.57	0.55	0	1.0	2.0	2.0	4
HalfBath	0	1.00	0.38	0.50	0	0.0	0.0	1.0	2
BedroomAbvGr	0	1.00	2.86	0.82	0	2.0	3.0	3.0	8
KitchenAbvGr	0	1.00	1.04	0.21	0	1.0	1.0	1.0	3
TotRmsAbvGrd	0	1.00	6.45	1.57	2	5.0	6.0	7.0	15
Fireplaces	0	1.00	0.60	0.65	0	0.0	1.0	1.0	4
GarageYrBlt	159	0.95	1978.11	25.57	1895	1960.0	1979.0	2002.0	2207
GarageCars	1	1.00	1.77	0.76	0	1.0	2.0	2.0	5
GarageArea	1	1.00	472.87	215.39	0	320.0	480.0	576.0	1488
WoodDeckSF	0	1.00	93.71	126.53	0	0.0	0.0	168.0	1424
OpenPorchSF	0	1.00	47.49	67.58	0	0.0	26.0	70.0	742
EnclosedPorch	0	1.00	23.10	64.24	0	0.0	0.0	0.0	1012
X3SsnPorch	0	1.00	2.60	25.19	0	0.0	0.0	0.0	508
ScreenPorch	0	1.00	16.06	56.18	0	0.0	0.0	0.0	576
PoolArea	0	1.00	2.25	35.66	0	0.0	0.0	0.0	800
MiscVal	0	1.00	50.83	567.40	0	0.0	0.0	0.0	17000
MoSold	0	1.00	6.21	2.71	1	4.0	6.0	8.0	12
YrSold	0	1.00	2007.79	1.31	2006	2007.0	2008.0	2009.0	2010
SalePrice	1459	0.50	180921.20	79442.50	34900	129975.0	163000.0	214000.0	755000

EDA

Lets do EDA for data quality.

We first check for missing values.

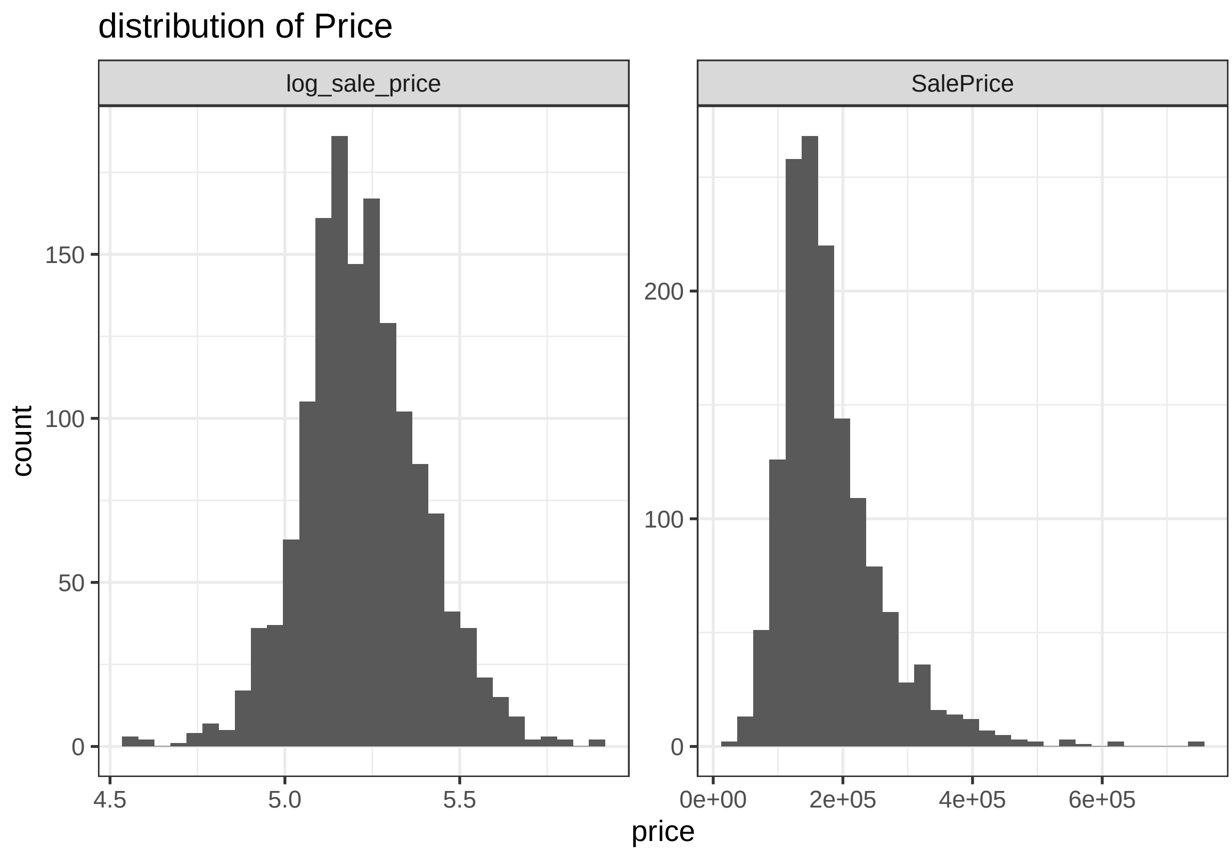
```
## # A tibble: 82 x 3
##    feature      num_missing pct_missing
##    <fct>              <int>       <dbl>
##  1 PoolQC              2909      0.997
##  2 MiscFeature         2814      0.964
##  3 Alley               2721      0.932
##  4 Fence               2348      0.804
##  5 SalePrice           1459      0.500
##  6 FireplaceQu         1420      0.486
##  7 LotFrontage          486      0.166
##  8 GarageYrBlt          159      0.0545
##  9 GarageFinish         159      0.0545
## 10 GarageQual           159      0.0545
## # ... with 72 more rows
```



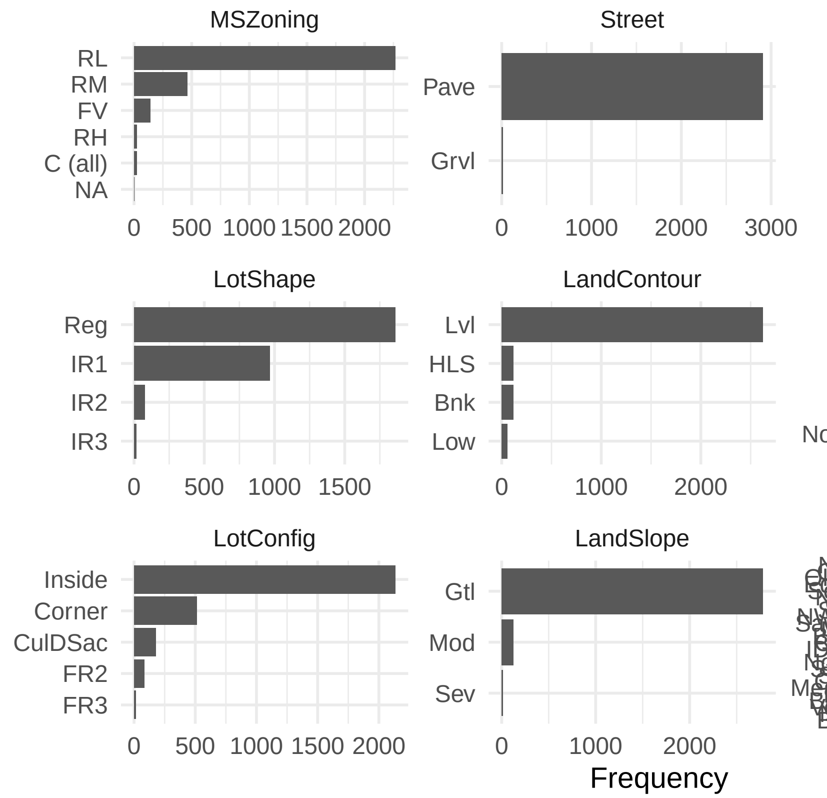
Most of the higher missing values is result of absence of a particular feature in the property as per Data description. We will code the missing values with appropriate categorical variable later on. Rest of the features have very low missing percentages which can be handled during model data preprocessing using median or knnimpte.

Further Data Exploration

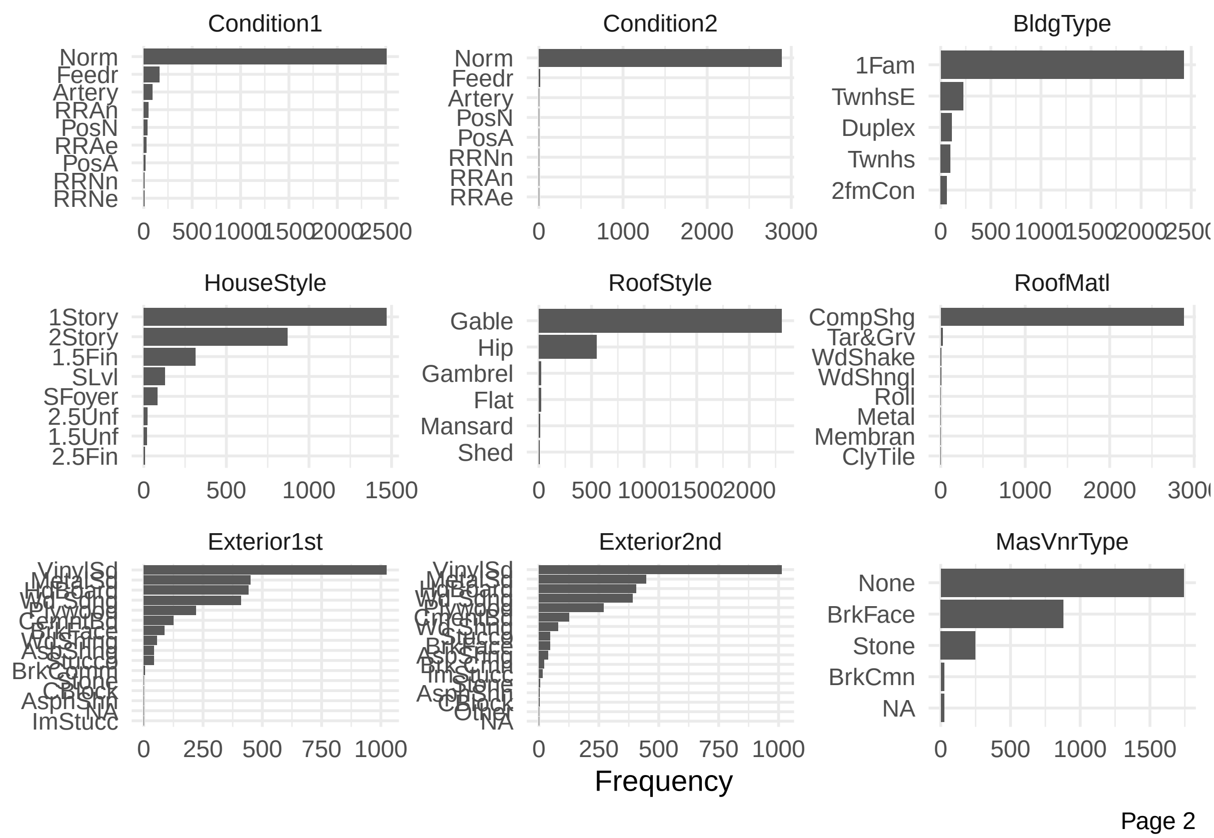
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

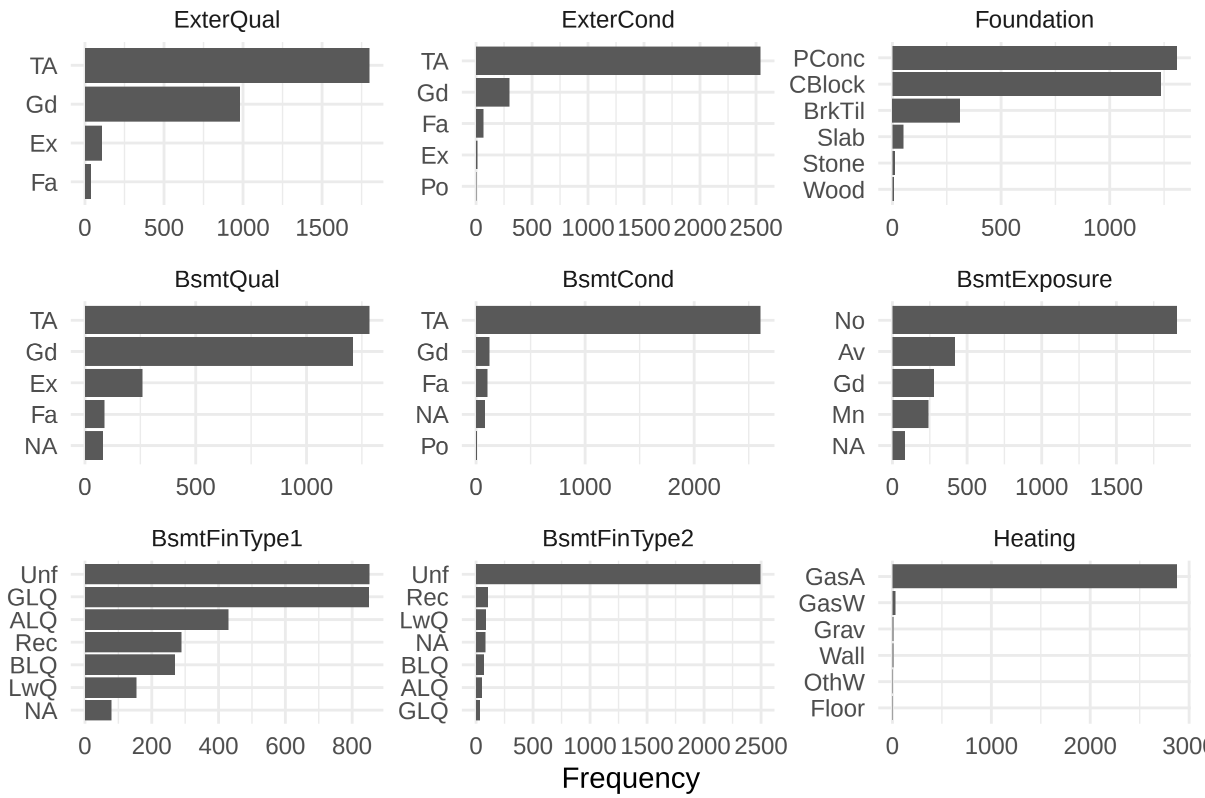


Sale Price distribtion is right skewed however log transformation is Normal and is a candidate for Linear Regression. Lets explore other features.

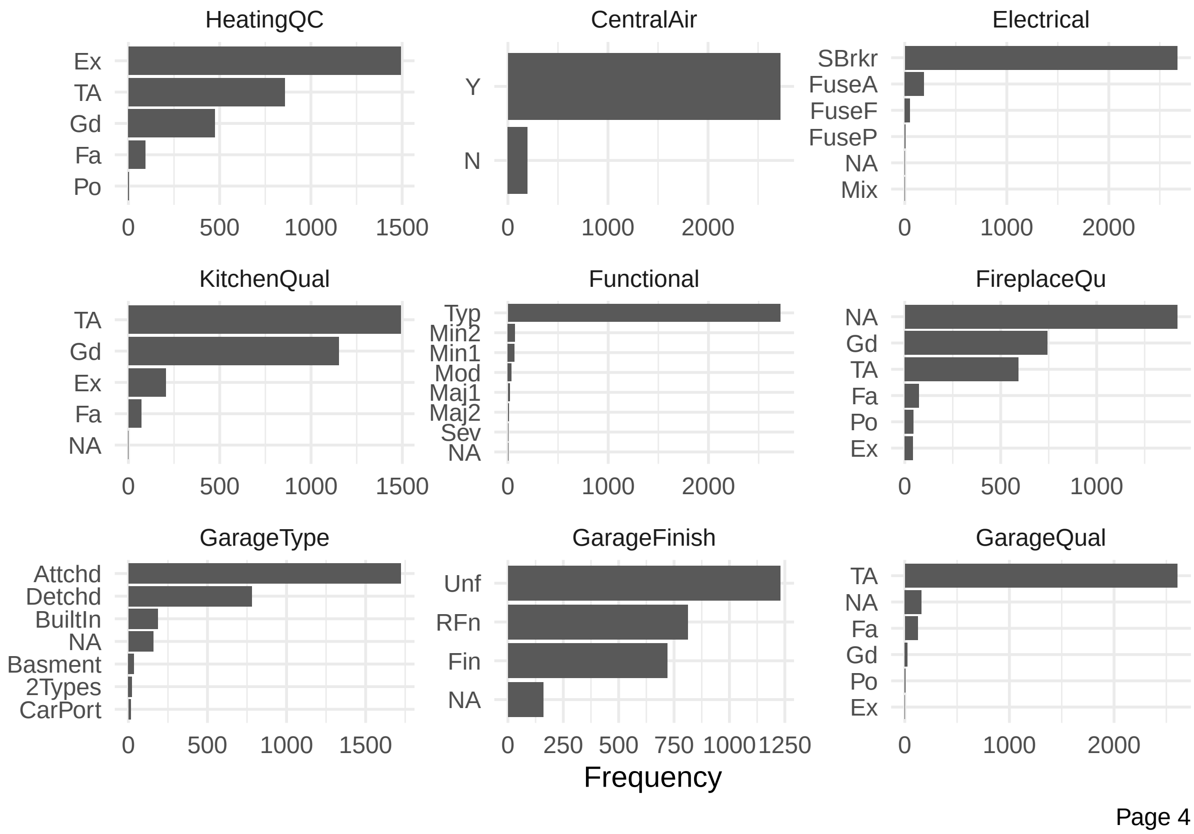


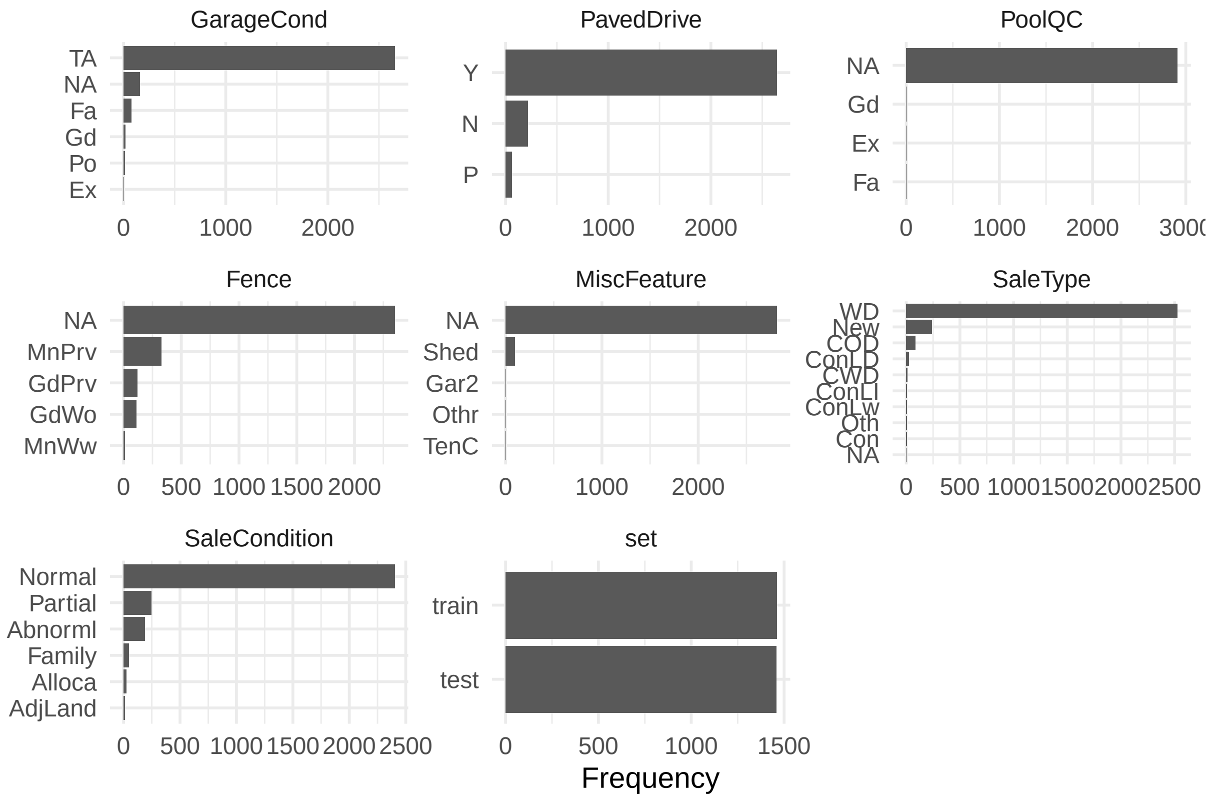
Check distribution of categorical variables



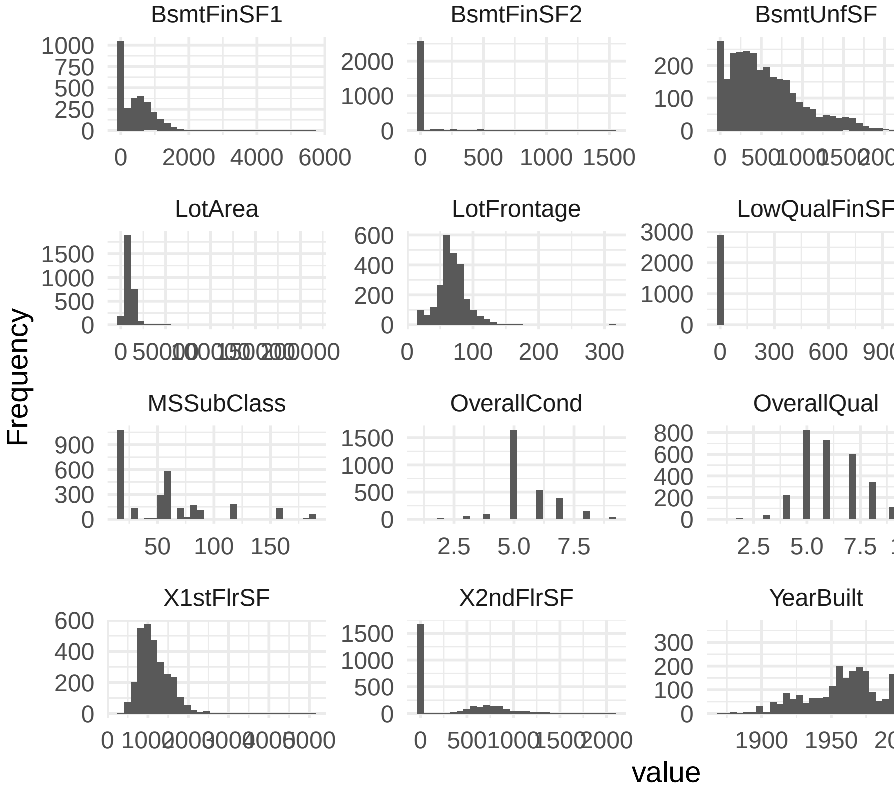

ExterQual
TA
Gd
Ex
Fa
0
500
1000
1500
ExterCond
TA
Gd
Fa
Ex
Po
0
500
1000
1500
2000
2500
Foundation
PConc
CBlock
BrkTil
Slab
Stone
Wood
0
500
1000
BsmtQual
TA
Gd
Ex
Fa
NA
0
500
1000
BsmtCond
TA
Gd
Fa
NA
Po
0
1000
2000
BsmtExposure
No
Av
Gd
Mn
NA
0
500
1000
1500
BsmtFinType1
Unf
GLQ
ALQ
Rec
BLQ
LwQ
NA
0
200
400
600
800
BsmtFinType2
Unf
Rec
LwQ
NA
BLQ
ALQ
GLQ
0
500
1000
1500
2000
2500
Heating
GasA
GasW
Grav
Wall
OthW
Floor
0
1000
2000
Frequency

Page 3

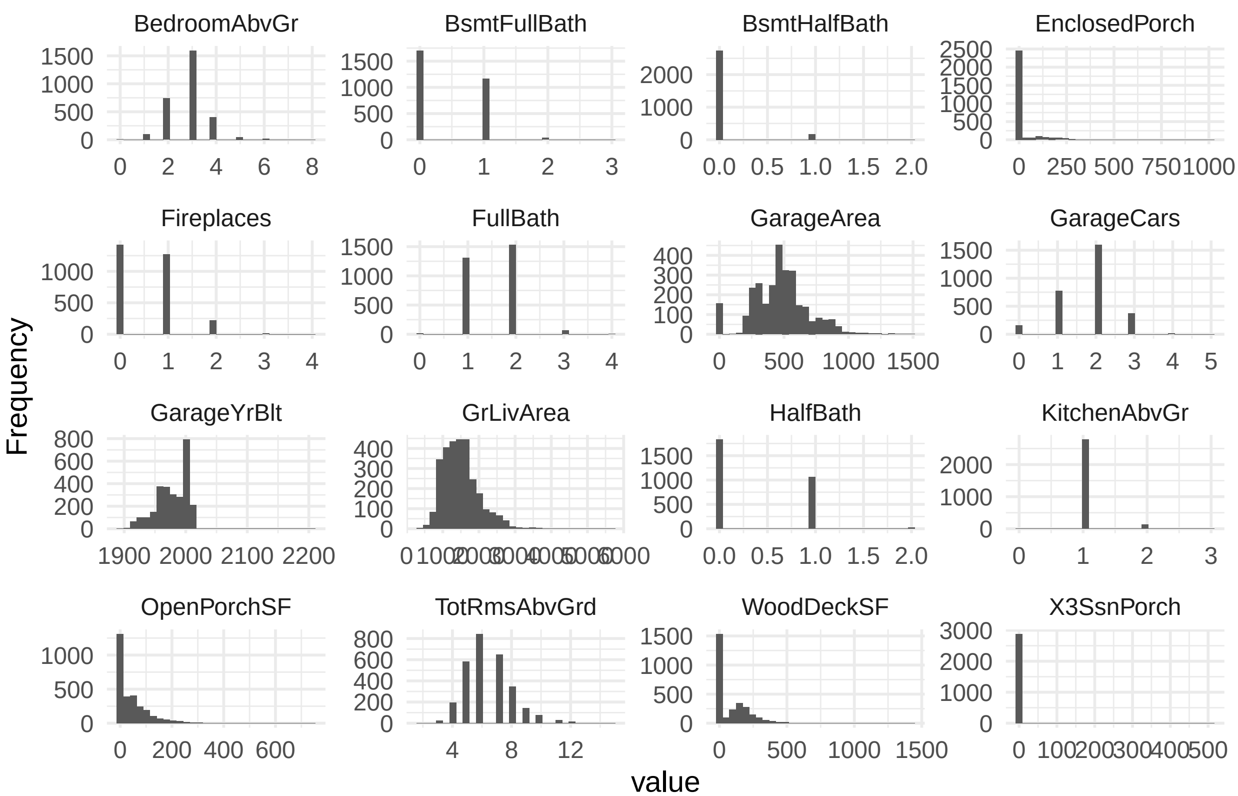

HeatingQC
Ex
TA
Gd
Fa
Po
0 500 1000 1500
CentralAir
Y
N
0 1000 2000
Electrical
SBrkr
FuseA
FuseF
FuseP
NA
Mix
0 1000 2000
KitchenQual
TA
Gd
Ex
Fa
NA
0 500 1000 1500
Functional
Typ
Min2
Min1
Mod
Maj1
Maj2
Sev
NA
0 1000 2000
FireplaceQu
NA
Gd
TA
Fa
Po
Ex
0 500 1000
GarageType
Attchd
Detchd
BuiltIn
NA
Basment
2Types
CarPort
0 500 1000 1500
GarageFinish
Unf
RFn
Fin
NA
0 250 500 750 1000 1250
GarageQual
TA
NA
Fa
Gd
Po
Ex
0 1000 2000
Frequency
Page 4

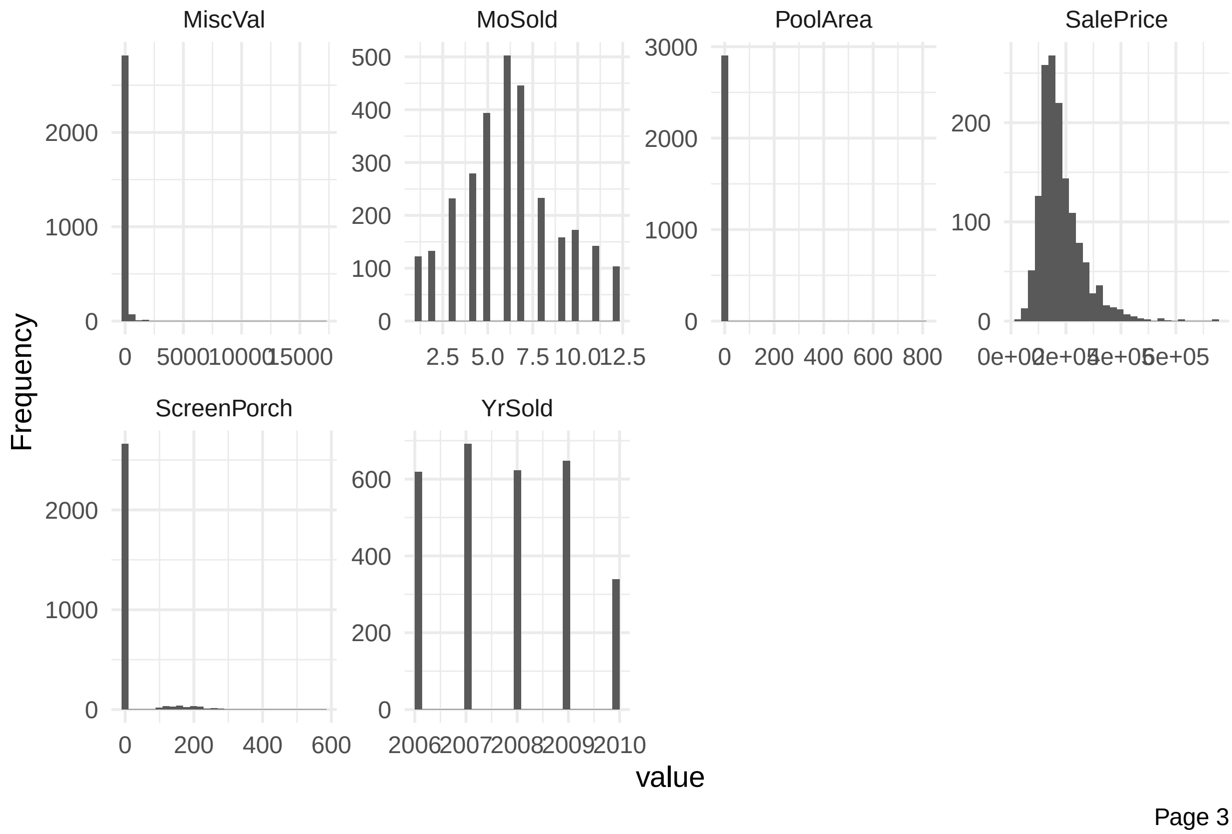

GarageCond
TA
NA
Fa
Gd
Po
Ex
0
1000
2000
PavedDrive
Y
N
P
0
1000
2000
PoolQC
NA
Gd
Ex
Fa
0
1000
2000
300(
Fence
NA
MnPrv
GdPrv
GdWo
MnWw
0
500
1000
1500
2000
MiscFeature
NA
Shed
Gar2
Othr
TenC
0
1000
2000
SaleType
WD
New
COD
ConLD
CWD
ConLI
ConLw
Oth
Con
NA
0
500
1000
1500
2000
2500
SaleCondition
Normal
Partial
Abnorml
Family
Alloca
AdjLand
0
500
1000
1500
2000
2500
set
train
test
0
500
1000
1500
Frequency

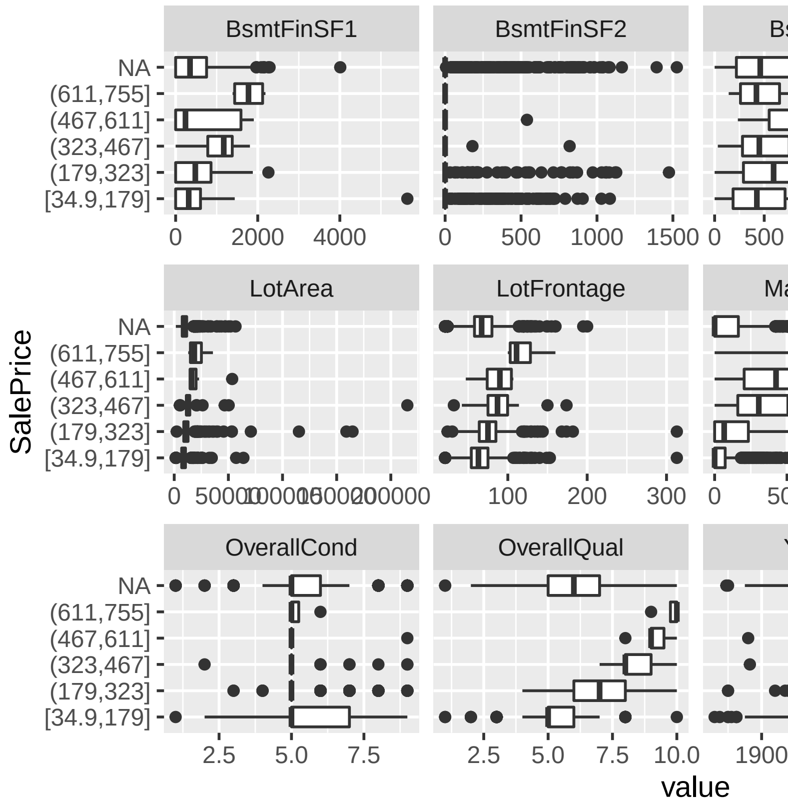
Page 5



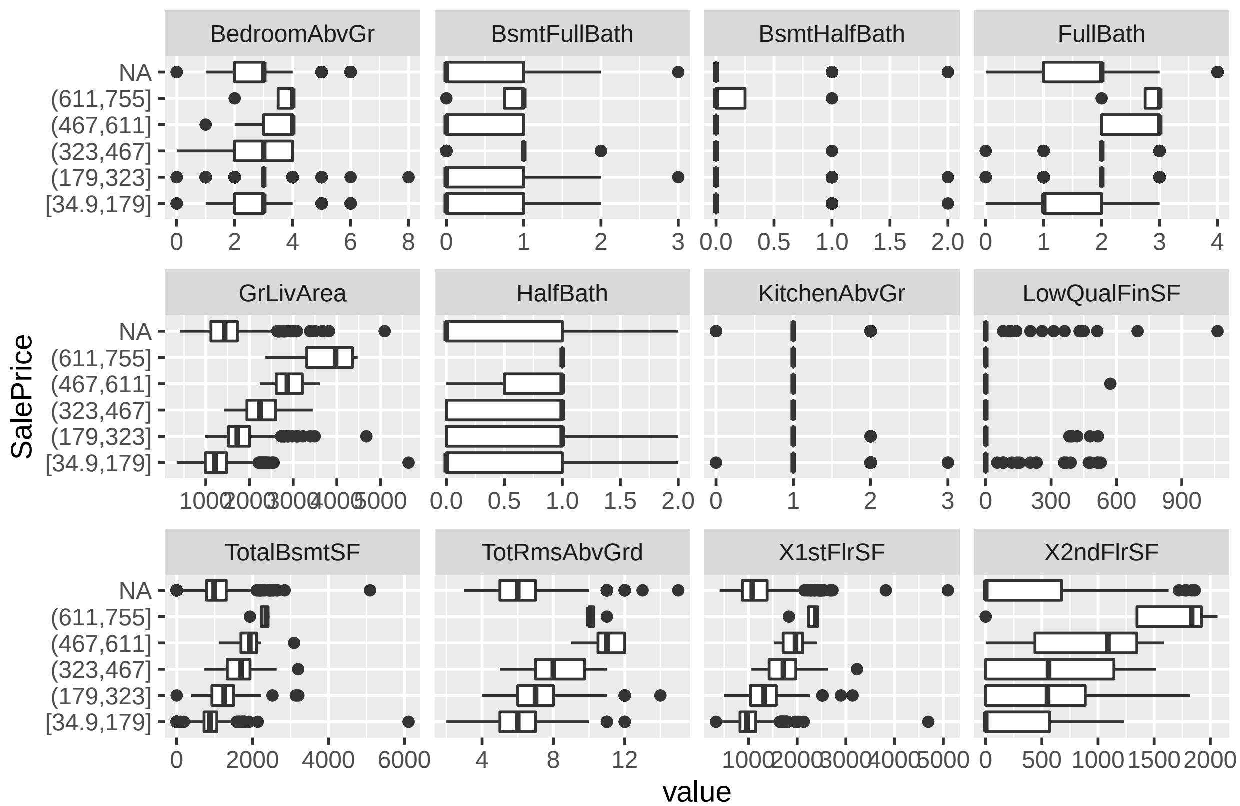
Check distribution of Numeric features

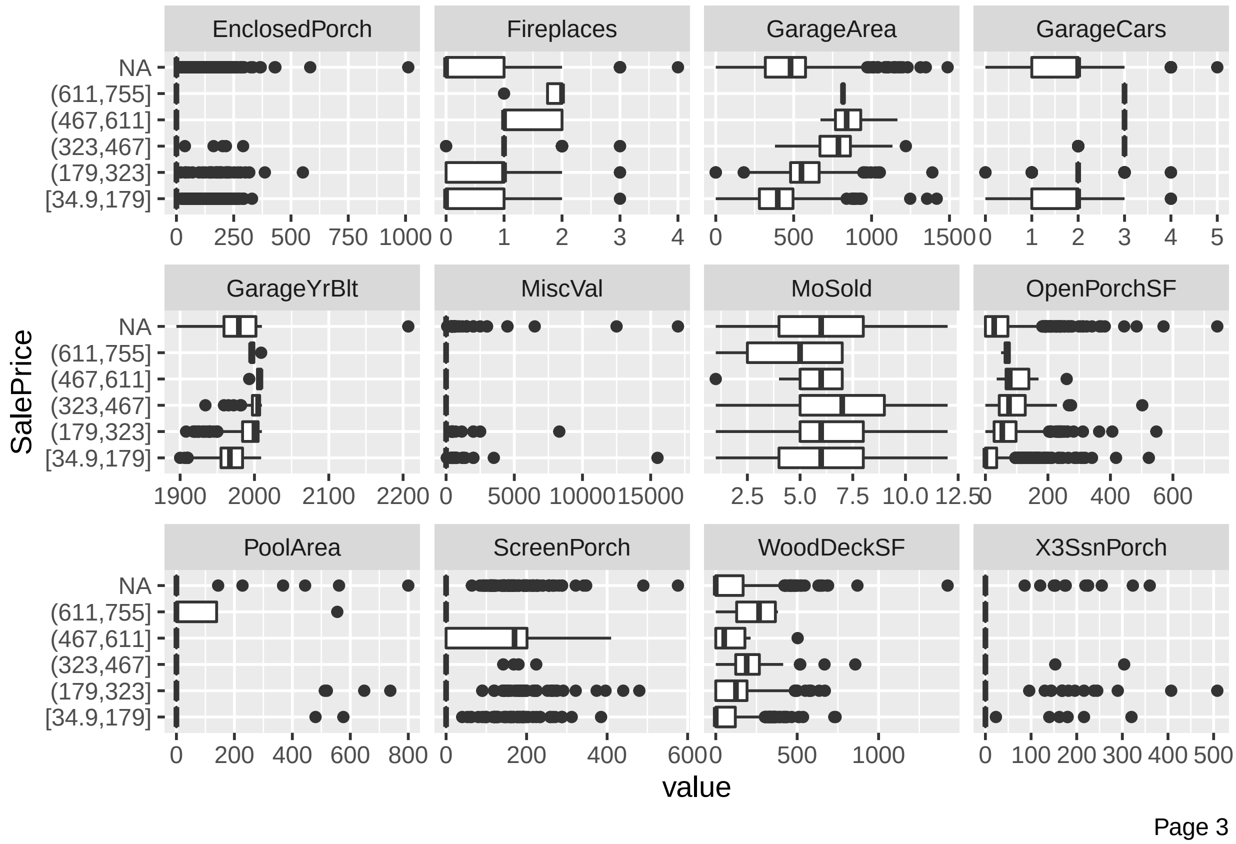


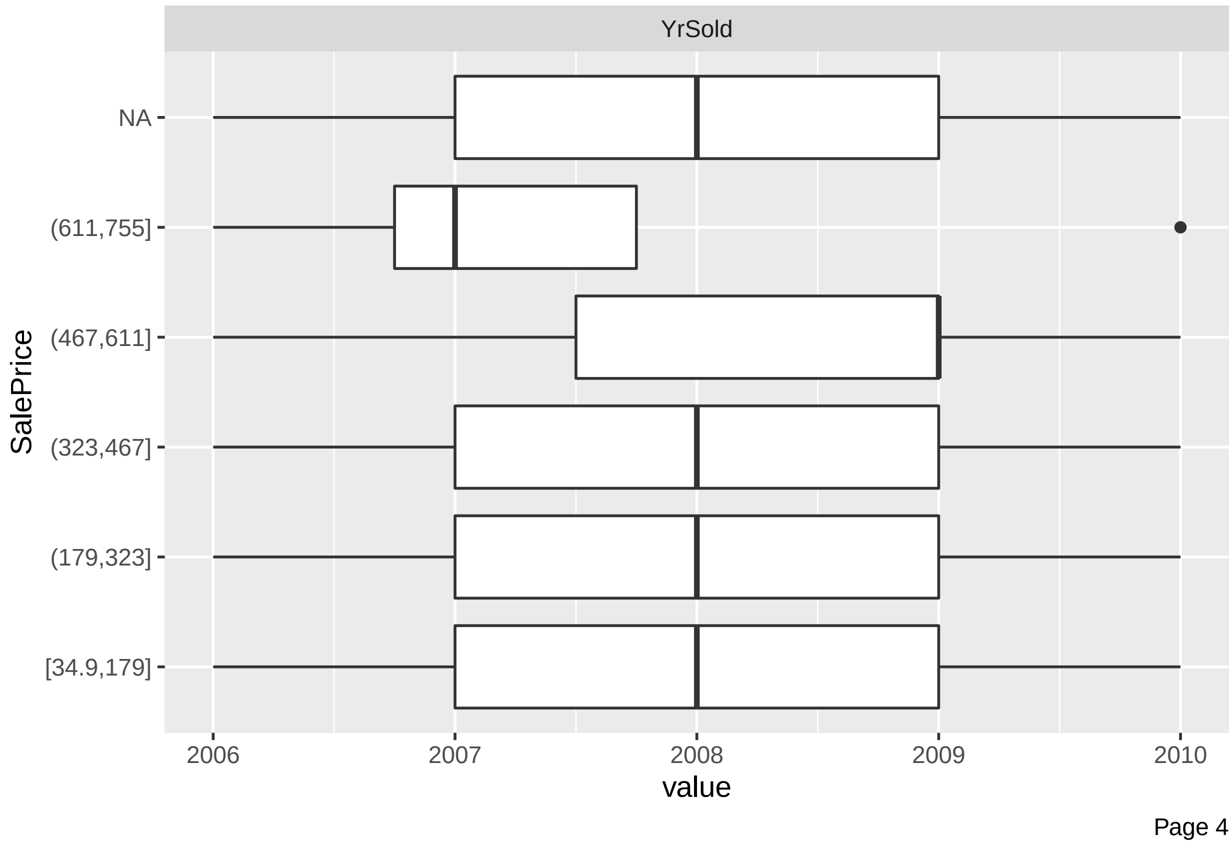

MiscVal
MoSold
PoolArea
SalePrice
ScreenPorch
YrSold
Frequency
value
Page 3



Explore categorical variables against Sale price.




EnclosedPorch
Fireplaces
GarageArea
GarageCars
GarageYrBlt
MiscVal
MoSold
OpenPorchSF
PoolArea
ScreenPorch
WoodDeckSF
X3SsnPorch
NA
(611,755]
(467,611]
(323,467]
(179,323]
[34.9,179]
SalePrice
value
Page 3

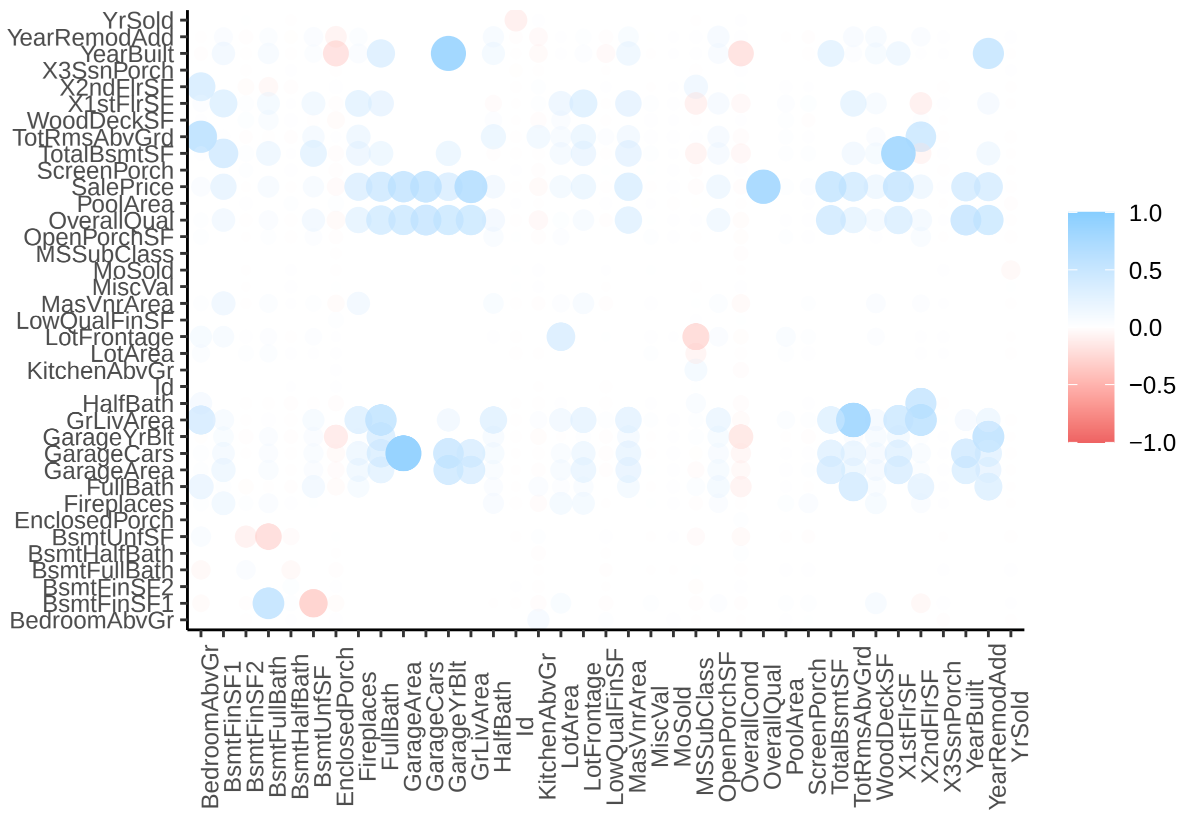


Checking correlation among variables

```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```

```
## Registered S3 method overwritten by 'seriation':
##   method         from
##   reorder.hclust gclus
```

```
## Don't know how to automatically pick scale for object of type noquote. Defaulting to continuous.
```




```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```

```
## # A tibble: 18 x 3
##    rowname      features     corr
##    <chr>        <chr>        <dbl>
##  1 GarageCars   GarageArea   0.890
##  2 YearBuilt    GarageYrBlt  0.835
##  3 GrLivArea    TotRmsAbvGrd 0.808
##  4 TotalBsmtSF  X1stFlrSF    0.802
##  5 OverallQual  SalePrice    0.791
##  6 GrLivArea    SalePrice    0.709
##  7 BedroomAbvGr TotRmsAbvGrd 0.670
##  8 X2ndFlrSF    GrLivArea    0.655
##  9 YearRemodAdd GarageYrBlt  0.652
## 10 GarageCars   SalePrice    0.640
## 11 BsmtFinSF1   BsmtFullBath 0.639
## 12 GrLivArea    FullBath     0.630
## 13 GarageArea   SalePrice    0.623
## 14 TotalBsmtSF  SalePrice    0.614
## 15 YearBuilt    YearRemodAdd 0.612
## 16 X2ndFlrSF    HalfBath     0.611
## 17 X1stFlrSF    SalePrice    0.606
## 18 OverallQual  GarageCars   0.601
```

Check for near zero variables

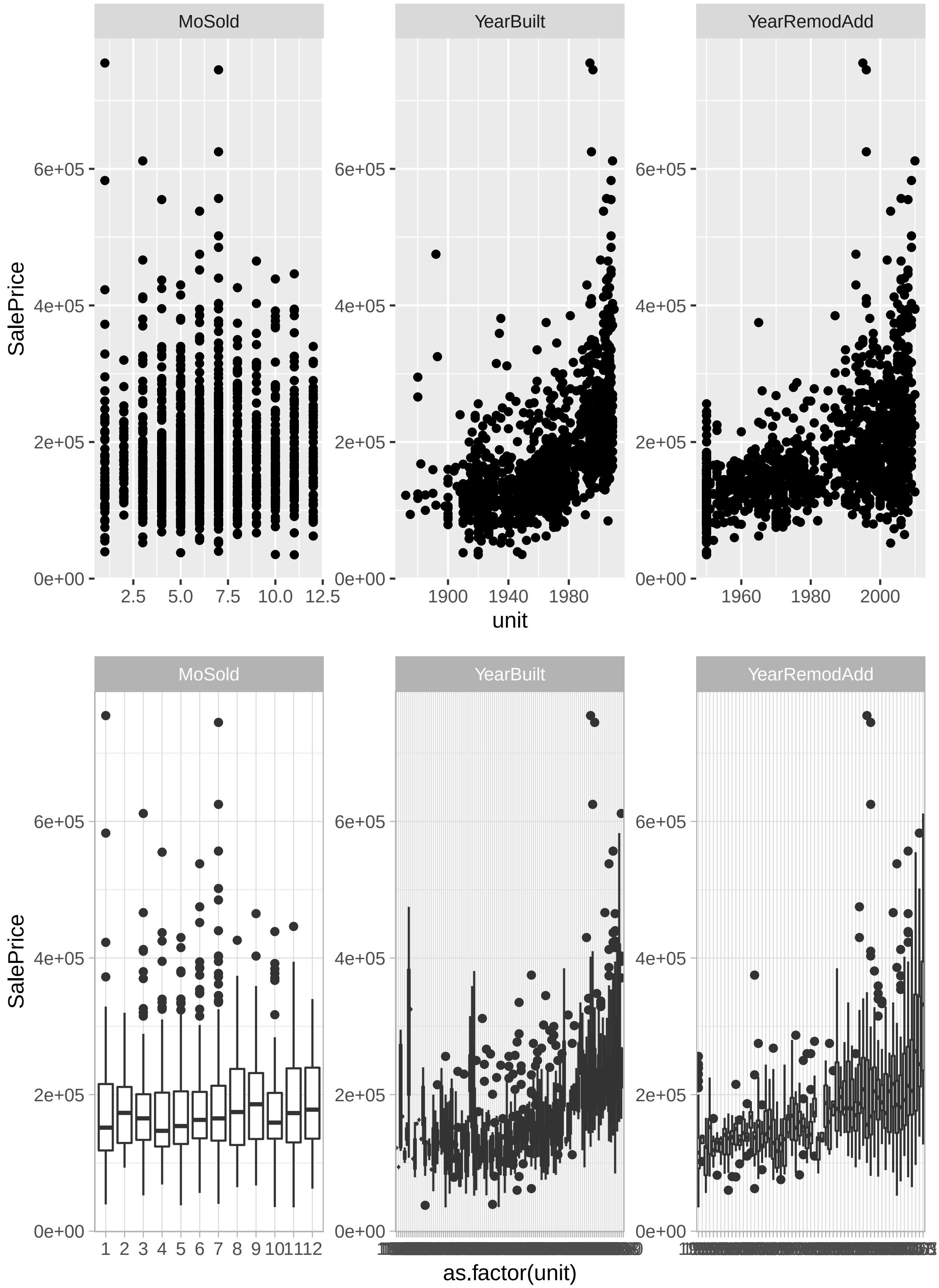
```
##   Street     LandContour  Utilities    LandSlope    Condition2      RoofMatl
##  Grvl:  12   Bnk: 117    AllPub:2916   Gtl:2778   Norm   :2889   CompShg:2876
##  Pave:2907   HLS: 120    NoSeWa:   1   Mod: 125   Feedr  :  13   Tar&Grv:  23
##              Low:  60    NA's  :   2   Sev:  16   Artery :   5   WdShake:   9
##              Lvl:2622                             PosA   :   4   WdShngl:   7
##                                                   PosN   :   4   ClyTile:   1
##                                                   RRNn   :   2   Membran:   1
##                                                   (Other):   2   (Other):   2
##  BsmtCond    BsmtFinType2   BsmtFinSF2       Heating      LowQualFinSF
##  Fa  : 104   ALQ :  52    Min.   :   0.00   Floor:   1   Min.   :   0.000
##  Gd  : 122   BLQ :  68    1st Qu.:   0.00   GasA :2874   1st Qu.:   0.000
##  Po  :   5   GLQ :  34    Median :   0.00   GasW :  27   Median :   0.000
##  TA  :2606   LwQ :  87    Mean   :  49.58   Grav :   9   Mean   :   4.694
##  NA's:  82   Rec : 105    3rd Qu.:   0.00   OthW :   2   3rd Qu.:   0.000
##              Unf :2493    Max.   :1526.00   Wall :   6   Max.   :1064.000
##              NA's:  80    NA's   :1
##   KitchenAbvGr     Functional   GarageQual  GarageCond   OpenPorchSF
##  Min.   :0.000   Typ    :2717   Ex  :   3   Ex  :   3   Min.   :  0.00
##  1st Qu.:1.000   Min2   :  70   Fa  : 124   Fa  :  74   1st Qu.:  0.00
##  Median :1.000   Min1   :  65   Gd  :  24   Gd  :  15   Median : 26.00
##  Mean   :1.045   Mod    :  35   Po  :   5   Po  :  14   Mean   : 47.49
##  3rd Qu.:1.000   Maj1   :  19   TA  :2604   TA  :2654   3rd Qu.: 70.00
##  Max.   :3.000   (Other):  11   NA's: 159   NA's: 159   Max.   :742.00
##                  NA's   :   2
##  EnclosedPorch      X3SsnPorch       ScreenPorch       PoolArea
##  Min.   :   0.0   Min.   :  0.000   Min.   :  0.00   Min.   :  0.000
##  1st Qu.:   0.0   1st Qu.:  0.000   1st Qu.:  0.00   1st Qu.:  0.000
##  Median :   0.0   Median :  0.000   Median :  0.00   Median :  0.000
##  Mean   :  23.1   Mean   :  2.602   Mean   : 16.06   Mean   :  2.252
##  3rd Qu.:   0.0   3rd Qu.:  0.000   3rd Qu.:  0.00   3rd Qu.:  0.000
##  Max.   :1012.0   Max.   :508.000   Max.   :576.00   Max.   :800.000
##
##     MiscVal
##  Min.   :    0.00
##  1st Qu.:    0.00
##  Median :    0.00
##  Mean   :   50.83
##  3rd Qu.:    0.00
##  Max.   :17000.00
##
```

Curious to know if street type makes a difference

```
## # A tibble: 6 x 4
##   Var1  Var2       Grvl    Pave
##   <fct> <fct>     <dbl>   <dbl>
## 1 A     Min.     55993   34900
## 2 A     1st Qu.  88250  130000
## 3 A     Median  114250  163000
## 4 A     Mean    130190. 181131.
## 5 A     3rd Qu. 169650  214000
## 6 A     Max.    228950  755000
```

In general we do see House in pavement have higher Sale price than the Gravel.

Checking affect of time over Sale Price



There is no particular effect of month sold on Sale price. However, In general we dprices does increase Year on Year which makes common sense.

Data Cleaning Based on our EDA

The code below will fix few missing values which are actually features not present in a property. These features are related to . Alley . Bsmt . Garage . FireplaceQu . PoolQC . Fence . MiscFeatures . MasVnr

We also create a categorical bucketing variable out of YearSold and Yr Modelled. We create new variabe 'log_sale_price' that is log of Sale Price. This new variable will be used as response variables.

Last we remove Id, Sale Price and set variable that will not contribute to the model.

```
#pp3
house <- train %>%

  #converting sales to log scale
  mutate(log_sale_price = log(SalePrice)) %>%


  #fill in missing values NA for factor variables as per data description

  #Alley
  mutate(Alley = if_else(is.na(Alley), "No Alley", Alley)) %>%

  #Bsmt
  mutate(
    BsmtCond = if_else(is.na(BsmtCond), "No Bsmnt", BsmtCond),
    BsmtExposure =  if_else(is.na(BsmtExposure), "No Bsmnt", BsmtExposure),
    BsmtQual = if_else(is.na(BsmtQual), "No Bsmnt", BsmtQual),
    BsmtFinType1 =  if_else(is.na(BsmtFinType1), "No Bsmnt", BsmtFinType1),
    BsmtFinType2 =  if_else(is.na(BsmtFinType2), "No Bsmnt", BsmtFinType2),
    BsmtFinSF1 =  if_else(is.na(BsmtFinSF1), 0, BsmtFinSF1),
    BsmtFinSF2 =  if_else(is.na(BsmtFinSF2), 0, BsmtFinSF2),
    BsmtUnfSF =  if_else(is.na(BsmtUnfSF), 0, BsmtUnfSF),
    TotalBsmtSF =  if_else(is.na(TotalBsmtSF), 0, TotalBsmtSF),
    BsmtFullBath =  if_else(is.na(BsmtFullBath), 0, BsmtFullBath),
    BsmtHalfBath =  if_else(is.na(BsmtHalfBath), 0, BsmtHalfBath)

  ) %>%

  #Garage
  #GarageType, GarageType, GarageFinish, GarageQual, GarageCond, 'GarageYrBlt', 'GarageArea', 'GarageCa
  mutate(
    GarageType = if_else(is.na(GarageType), "No Garage", GarageType),
    GarageFinish =  if_else(is.na(GarageFinish), "No Garage", GarageFinish),
    GarageQual = if_else(is.na(GarageQual), "No Garage", GarageQual),
    GarageCond =  if_else(is.na(GarageCond), "No Garage", GarageCond),
    GarageYrBlt =  if_else(is.na(GarageYrBlt), 0, GarageYrBlt),
    GarageArea =  if_else(is.na(GarageArea), 0, GarageArea),
    GarageCars =  if_else(is.na(GarageCond), 0, GarageCars)
  ) %>%
```

```
  mutate(MasVnrType = if_else(is.na(MasVnrType), "No MasVnrType", MasVnrType)) %>%
  mutate(MasVnrArea = if_else(is.na(MasVnrArea), 0, MasVnrArea)) %>%


  #FireplaceQu, PoolQC,Fence, MiscFeature
  mutate(
    FireplaceQu = if_else(is.na(FireplaceQu), "No Fireplc", FireplaceQu),
    PoolQC =  if_else(is.na(PoolQC), "No Pool", PoolQC),
    Fence = if_else(is.na(Fence), "No Fence", Fence),
    MiscFeature =  if_else(is.na(MiscFeature), "None", MiscFeature)
  ) %>%

  mutate(YrBuilt_cat = cut2(YearBuilt, cuts = seq(min(YearBuilt), max(YearBuilt), 5))) %>%

  mutate(YearRemodAdd = if_else(is.na(YearRemodAdd), YearBuilt, YearRemodAdd)) %>%


  #remove non contributing features and IDs
  select(-Id,-SalePrice, -set)
```

Creating a Training and Validation(test) set using rsample

```
set.seed(1234)
house_split <- initial_split(house)

train_split <- training(house_split)
```

Create recepies with series of preprocessing steps on training set

```
house_recipe <- train_split %>%
  recipe(log_sale_price ~ .) %>%
  step_string2factor(all_nominal(), -all_outcomes()) %>%
  step_num2factor(MoSold, levels = as.character(unique(train_split$MoSold))) %>%
  step_num2factor(YrSold, levels = as.character(unique(train_split$YrSold))) %>%
  step_num2factor(MSSubClass, levels = as.character(unique(train_split$MSSubClass))) %>%
  step_num2factor(OverallCond, levels = as.character(unique(train_split$OverallCond))) %>%
  step_dummy(all_nominal(), -all_outcomes()) %>%
  step_center(all_predictors(), -all_outcomes()) %>%
  step_scale(all_predictors(), -all_outcomes()) %>%
  step_corr(all_predictors()) %>%
  step_nzv(all_predictors()) %>%
  step_knnimpute(all_predictors(), -all_outcomes())


doParallel::registerDoParallel()
house_prep <- prep(house_recipe)
```

```r
house_recipe_treebased <- train_split %>%
  recipe(log_sale_price ~ .) %>%
  step_string2factor(all_nominal(), -all_outcomes()) %>%
  step_num2factor(MoSold, levels = as.character(unique(train_split$MoSold))) %>%
  step_num2factor(YrSold, levels = as.character(unique(train_split$YrSold))) %>%
  step_num2factor(OverallCond, levels = as.character(unique(train_split$OverallCond))) %>%
  step_num2factor(MSSubClass, levels = as.character(unique(train_split$MSSubClass))) %>%
  step_dummy(all_nominal(), -all_outcomes()) %>%
  step_corr(all_numeric()) %>%
  step_nzv(all_numeric()) %>%
  step_knnimpute(all_predictors(), -all_outcomes())

doParallel::registerDoParallel()
house_treebased_prep <- prep(house_recipe_treebased)
```

Applying preprocessing on training and validation(testing) set

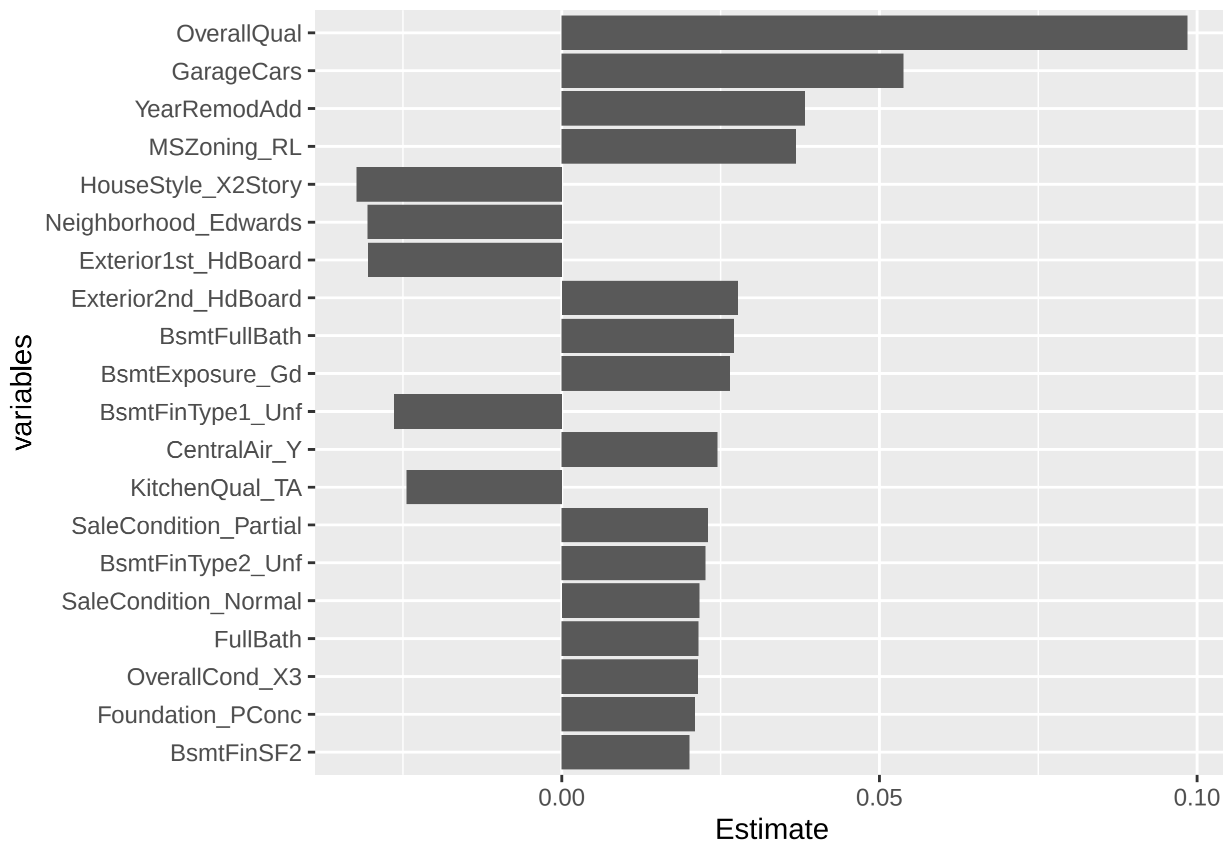
```r
house_train <- juice(house_prep)
house_test <- bake(house_prep, testing(house_split))

#splits for tree based models
house_train_treebased <- juice(house_treebased_prep)
house_test_treebased <- bake(house_treebased_prep, testing(house_split))
```

Train models

Train linear regression

```r
#linear model
lm_model <- #recipe(log_sale_price ~ . , data = house_train) %>%
  linear_reg() %>%
  set_mode("regression") %>%
  set_engine("lm") %>%
  fit(log_sale_price ~ . , data = house_train)
```

Important variables from linear model



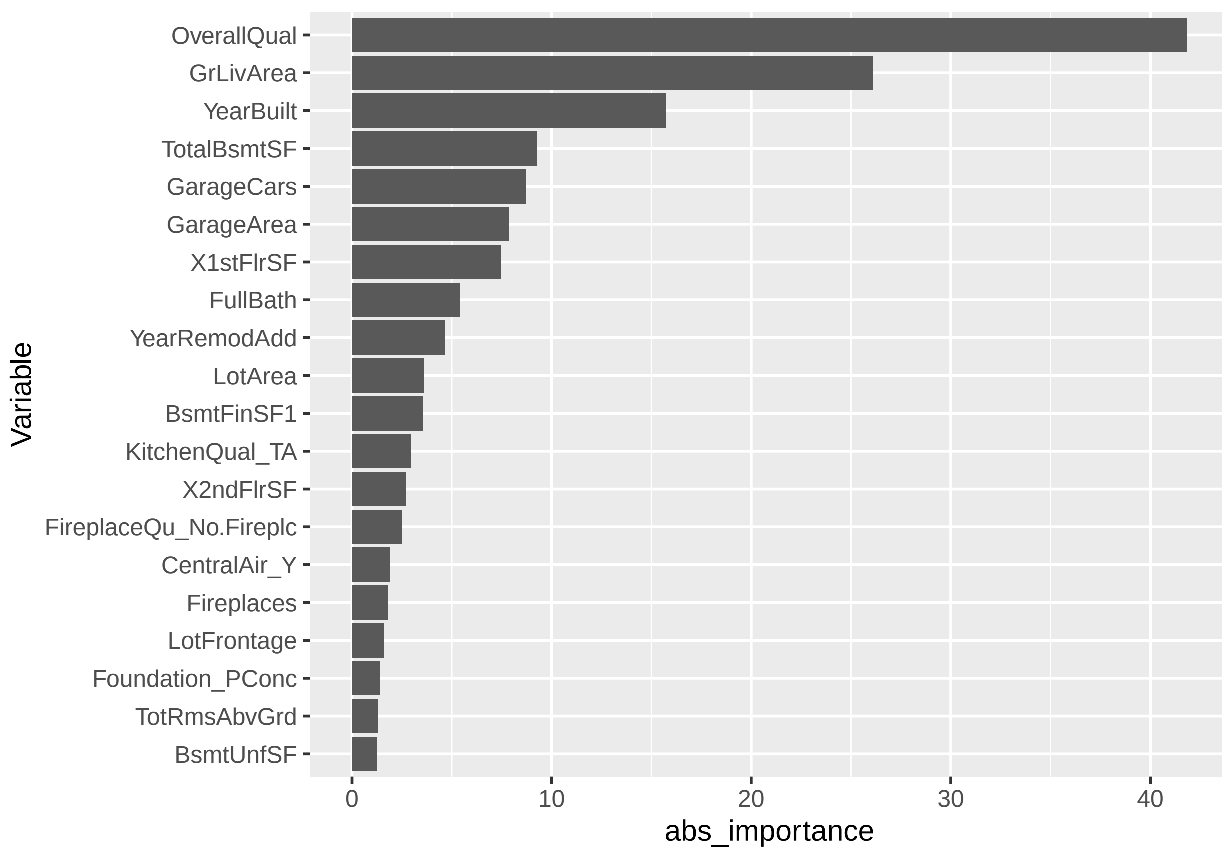
Train random forest

```
#create recepie on the preped house train data
rf_rec <-
  recipe(log_sale_price ~. , data = house_train_treebased)

#give model spec
rf_mod <-
  rand_forest(mtry = tune(), min_n = tune()) %>%
  set_engine("ranger") %>%
  set_mode("regression")

#create Search grid
rf_grid <-
  grid_regular(mtry(range = c(15,40)), min_n(range= c(10, 2)), levels = 5)

#create samples for cross validation
folds <- vfold_cv(house_train_treebased, v = 10)


doParallel::registerDoParallel()

#create models with grid search
rf_res <-
```

```r
  tune_grid(model = rf_mod, rf_rec,resamples = folds ,  grid = rf_grid)

final_mtry <- select_best(rf_res, "rmse", maximize = FALSE)$mtry
final_min_node <- select_best(rf_res, "rmse", maximize = FALSE)$min_n

#random forest
rf_model <- rand_forest(mtry = final_mtry, min_n = final_min_node) %>%
  set_mode("regression") %>%
  set_engine("ranger",importance = 'impurity') %>%
  fit(log_sale_price ~ . , data = house_train_treebased)
```

Important variables from randome forest model

```
## Selecting by abs_importance
```

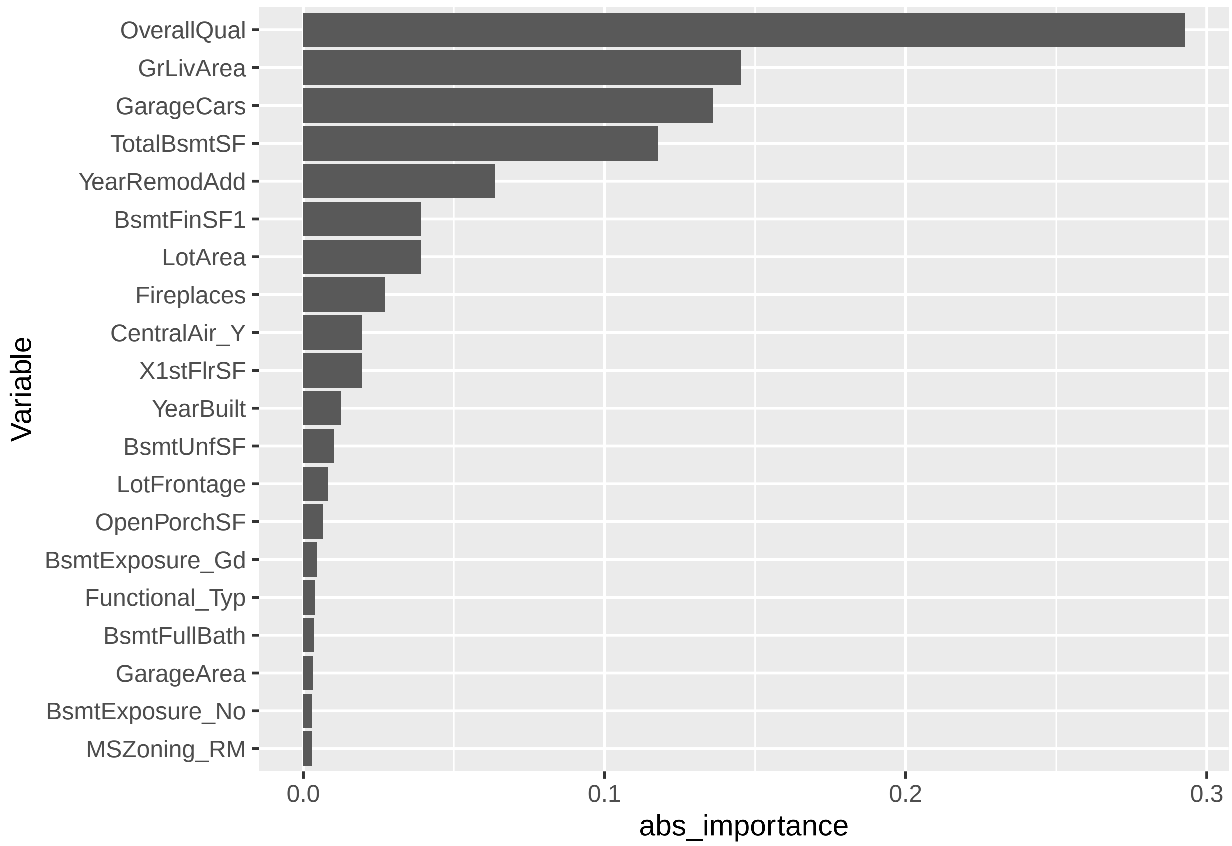


train xgboost

```r
doParallel::registerDoParallel()
#xgboost
xg_model <- boost_tree() %>%
  set_mode("regression") %>%
  set_engine("xgboost") %>%
  fit(log_sale_price ~ . , data = house_train_treebased)
```

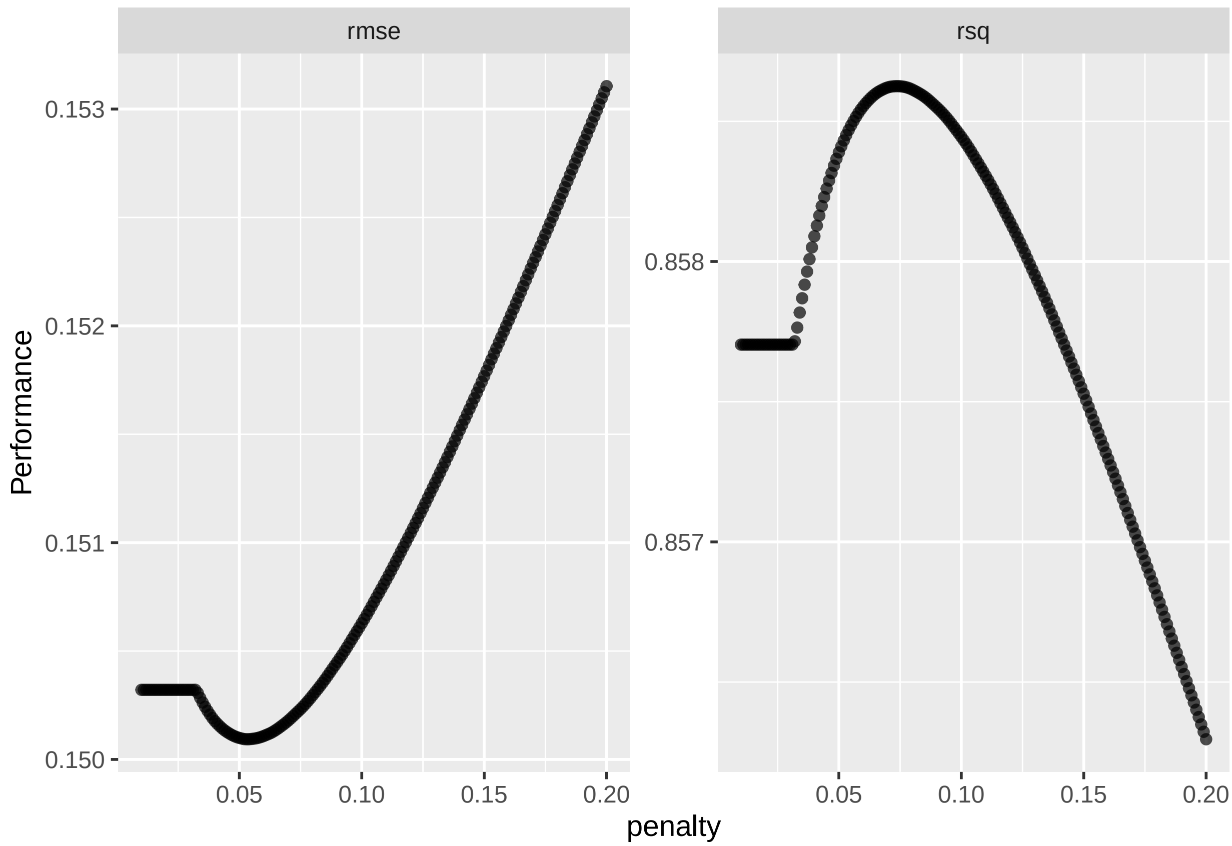
Important variables from xgoost model

```
## Selecting by abs_importance
```



train linear model ridge

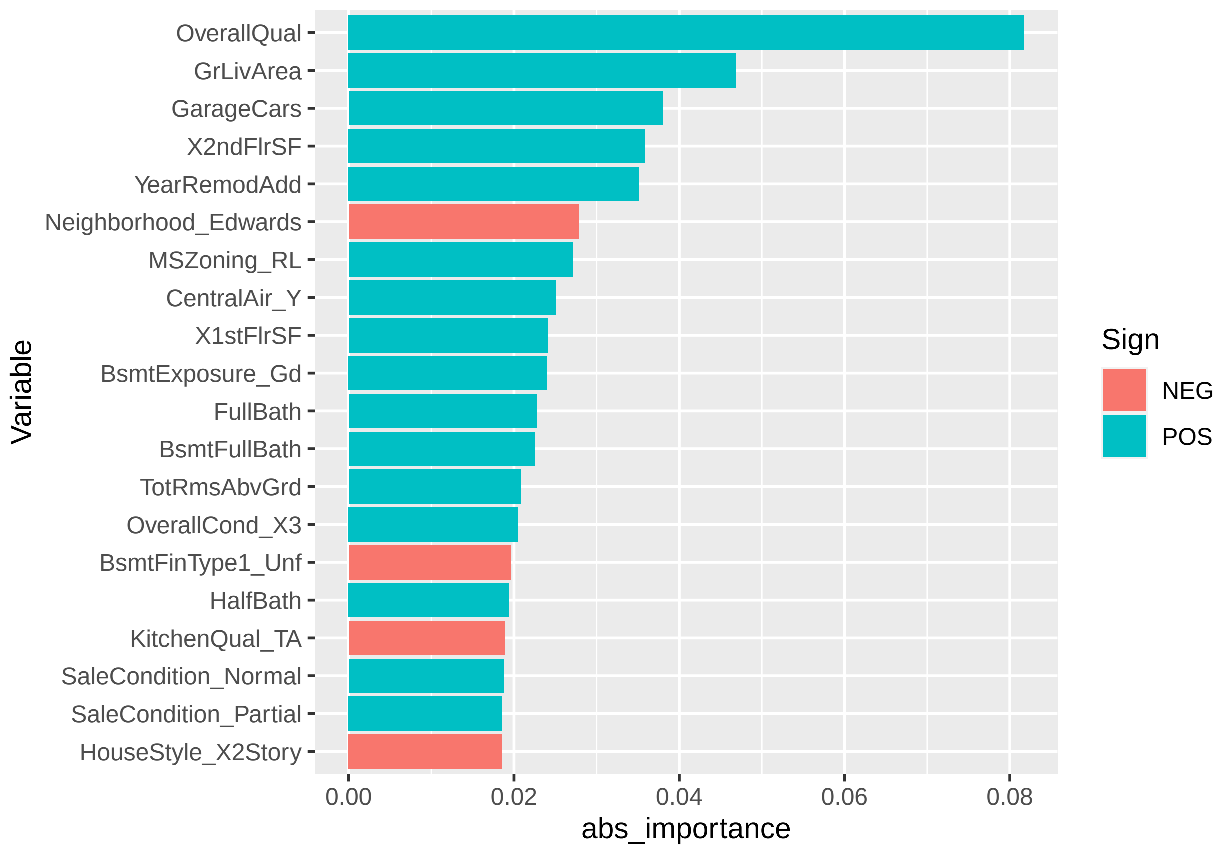
```r
ridge_rec <-
  recipe(log_sale_price ~. , data = house_train)

ridge_mod <-
  linear_reg(penalty = tune(), mixture = tune()) %>%
  set_engine("glmnet")


ridge_grid <- expand.grid(penalty = seq(from = 0.01, to = 0.2, by=0.001), mixture = 0)

folds <- vfold_cv(house_train, v = 10)

doParallel::registerDoParallel()

ridge_res <-
  tune_grid(ridge_rec, model = ridge_mod, resamples = folds ,  grid = ridge_grid)

autoplot(ridge_res)
```



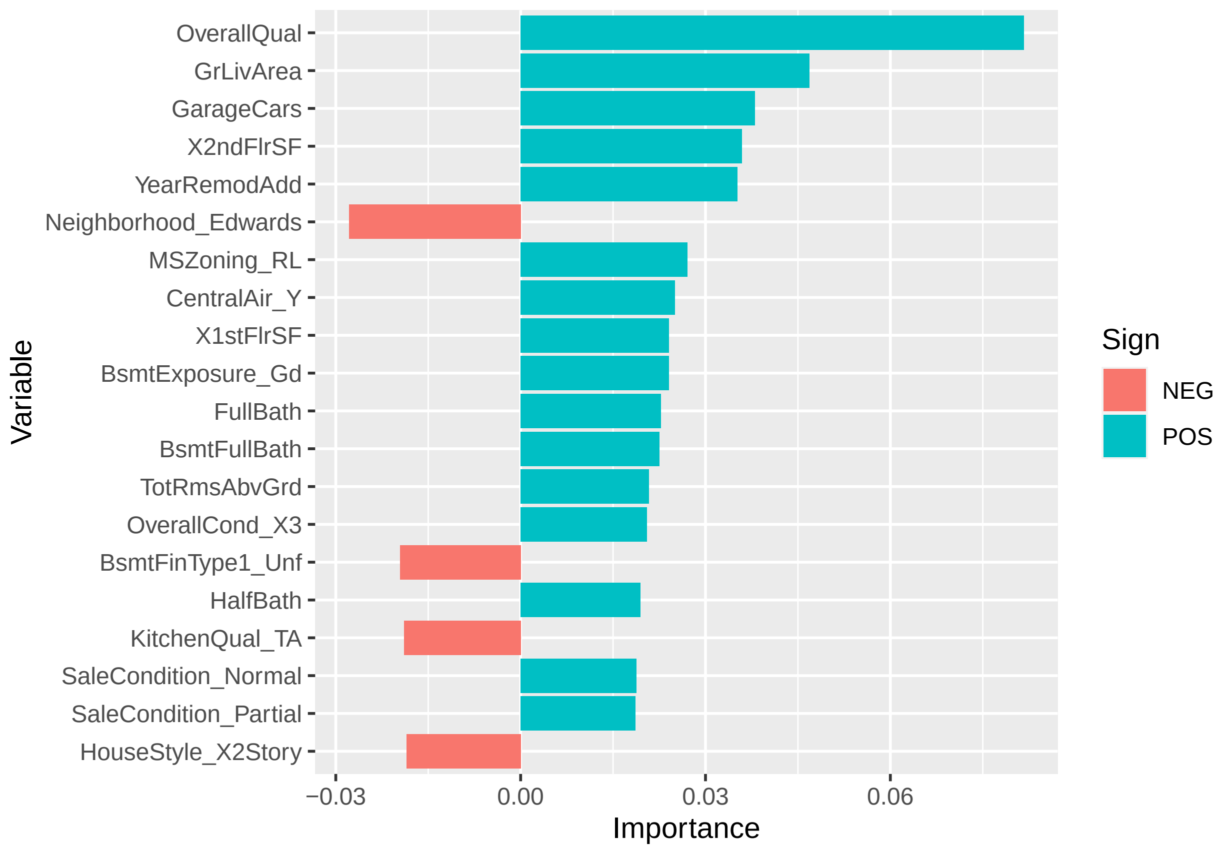

```
final_penalty <- select_best(ridge_res, "rmse", maximize = FALSE)$penalty

#apply best tuning parameter linear model ridge
lm_ridge_model <- linear_reg(penalty = final_penalty , mixture = 0) %>%
  set_mode("regression") %>%
  set_engine("glmnet") %>%
  fit(log_sale_price ~. , data = house_train)
```

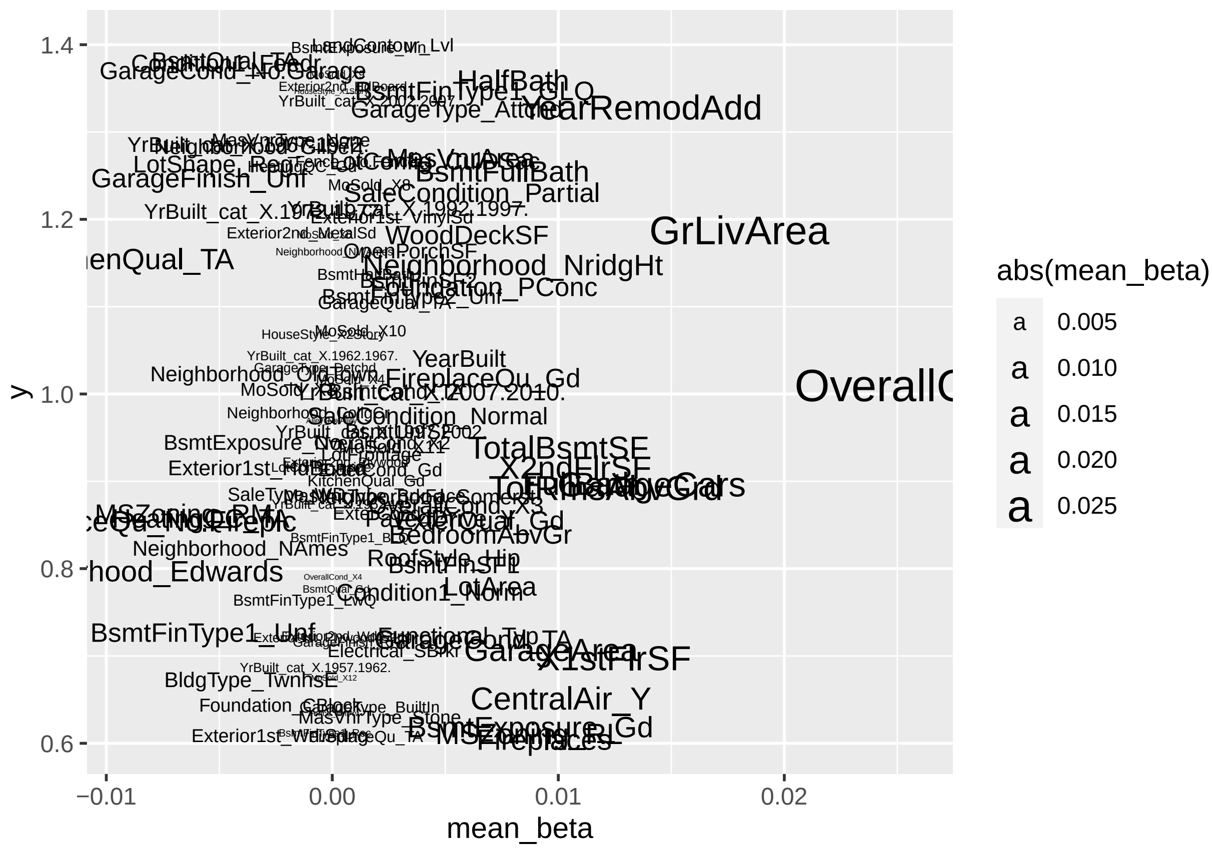
Important variables from ridge model

```
## Selecting by abs_importance
```



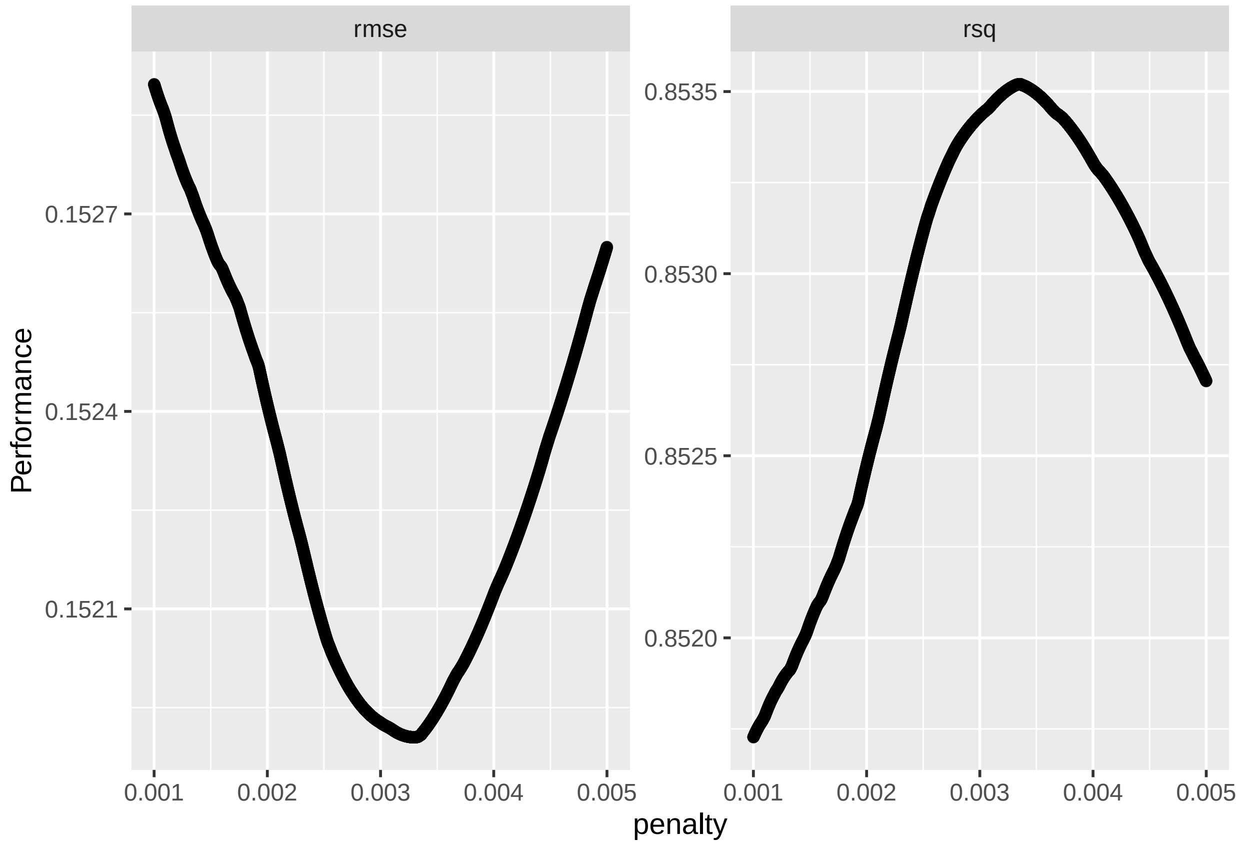

```
## Selecting by abs_importance
```


OverallQual
GrLivArea
GarageCars
X2ndFlrSF
YearRemodAdd
Neighborhood_Edwards
MSZoning_RL
CentralAir_Y
X1stFlrSF
BsmtExposure_Gd
FullBath
BsmtFullBath
TotRmsAbvGrd
OverallCond_X3
BsmtFinType1_Unf
HalfBath
KitchenQual_TA
SaleCondition_Normal
SaleCondition_Partial
HouseStyle_X2Story
Variable
−0.03
0.00
0.03
0.06
Importance
Sign
NEG
POS



linear model lasso

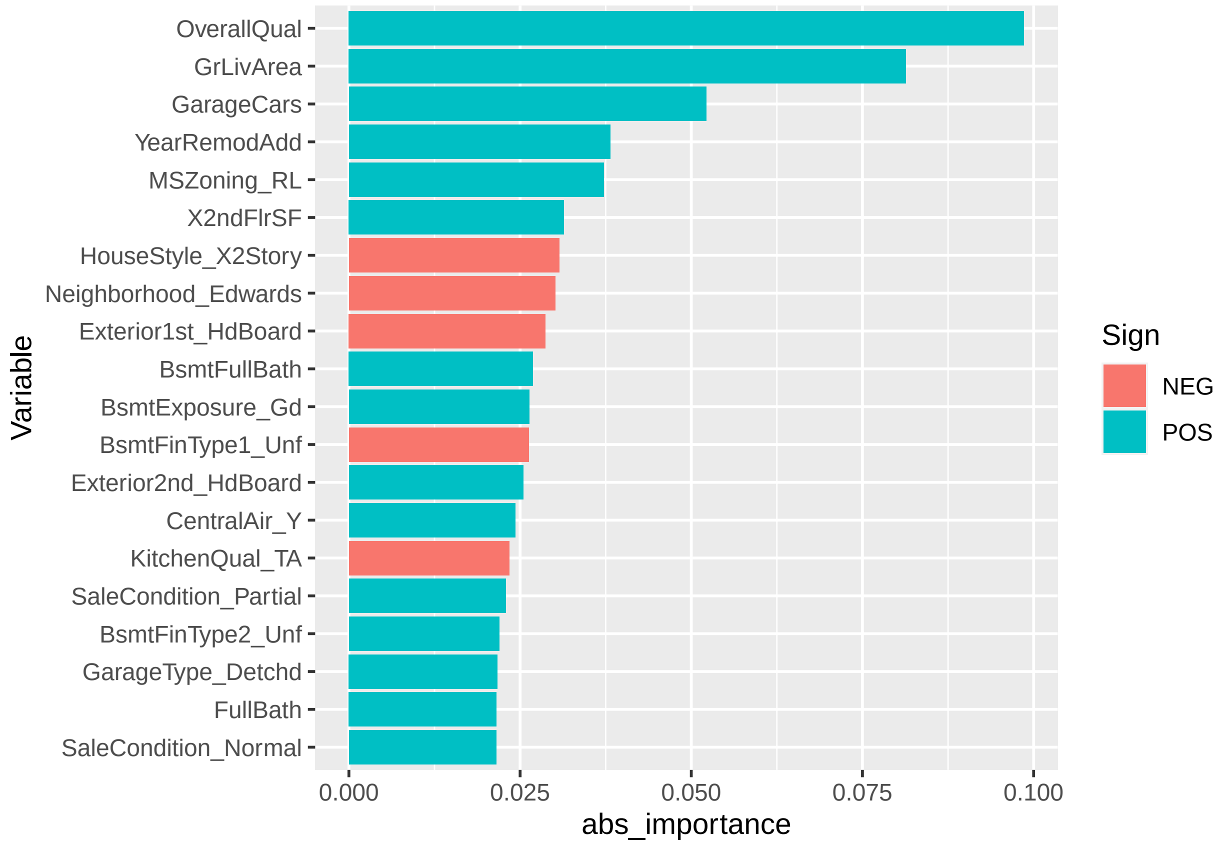
```
lasso_rec <-
  recipe(log_sale_price ~. , data = house_train)

lasso_mod <-
  linear_reg(penalty = tune(), mixture = tune()) %>%
  set_engine("glmnet")

lasso_grid <- expand.grid(penalty = seq(from = 0.001, to = 0.005, by = 0.000001), mixture = 1)

folds_lasso <- vfold_cv(house_train, v = 10)

lasso_res <-
  tune_grid(lasso_rec, model = lasso_mod, resamples = folds_lasso ,  grid = lasso_grid)

lambda_final <-
  select_best(lasso_res, metric = "rmse", maximize = FALSE) %>% select(penalty) %>% unlist

autoplot(lasso_res)
```



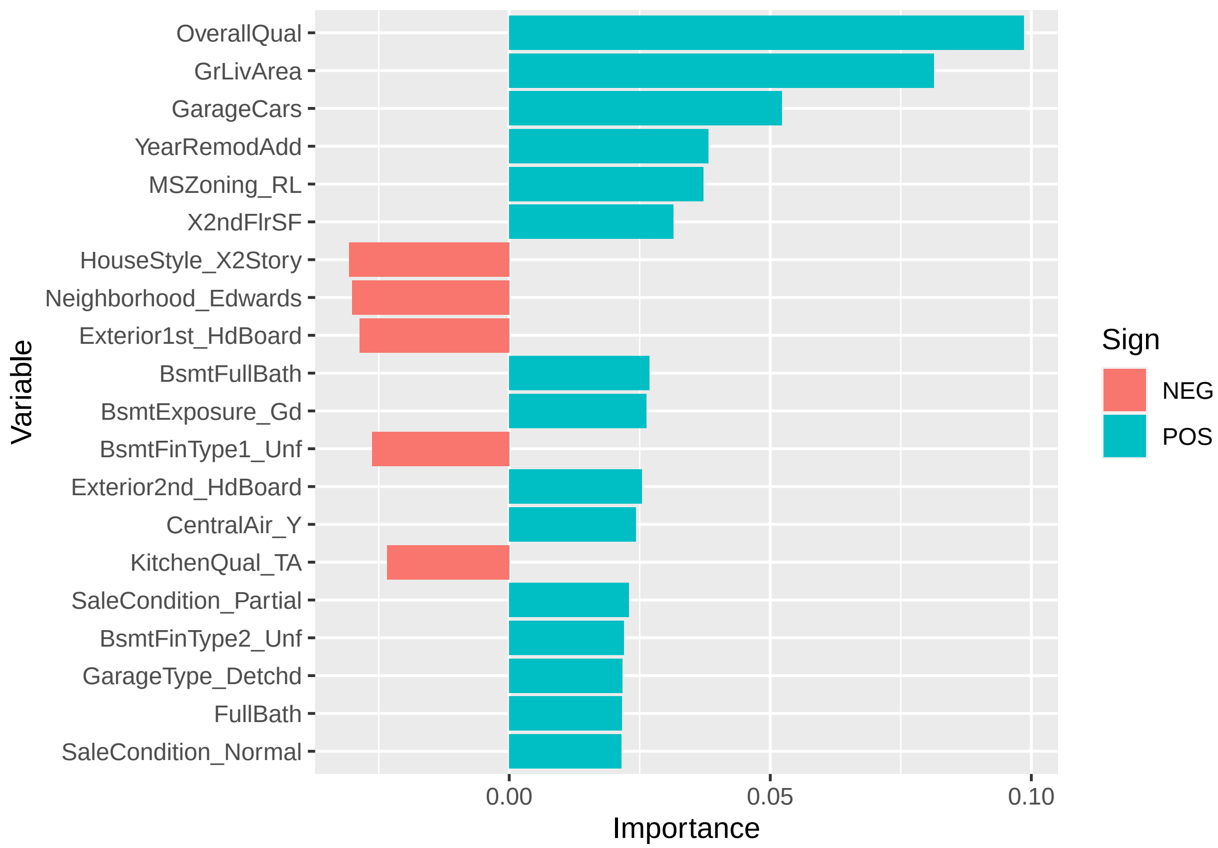

```r
#apply best tuning parameter linear model ridge
lm_lasso_model <- linear_reg(penalty =  lambda_final, mixture = 1) %>%
  set_mode("regression") %>%
  set_engine("glmnet") %>%
  fit(log_sale_price ~. , data = house_train)
```

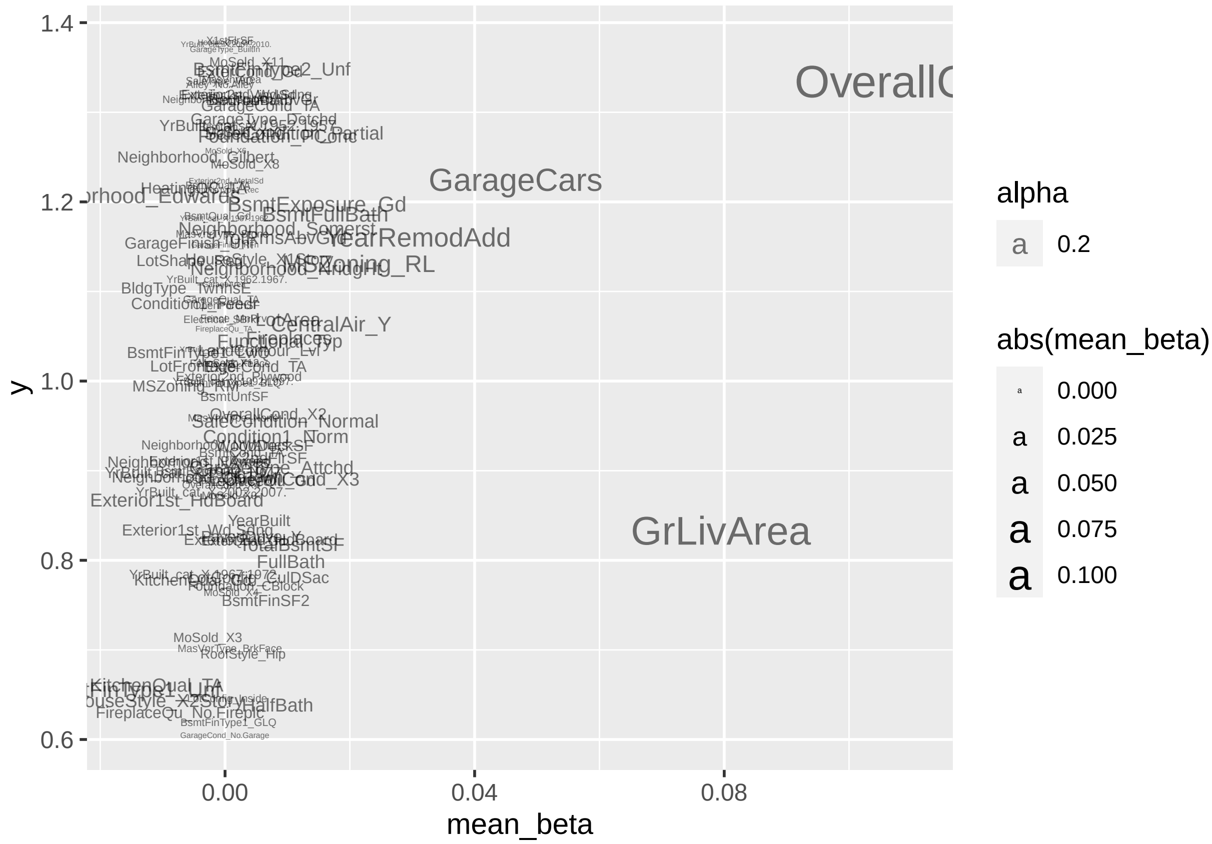
Important variables from lasso model

```
## Selecting by abs_importance
```



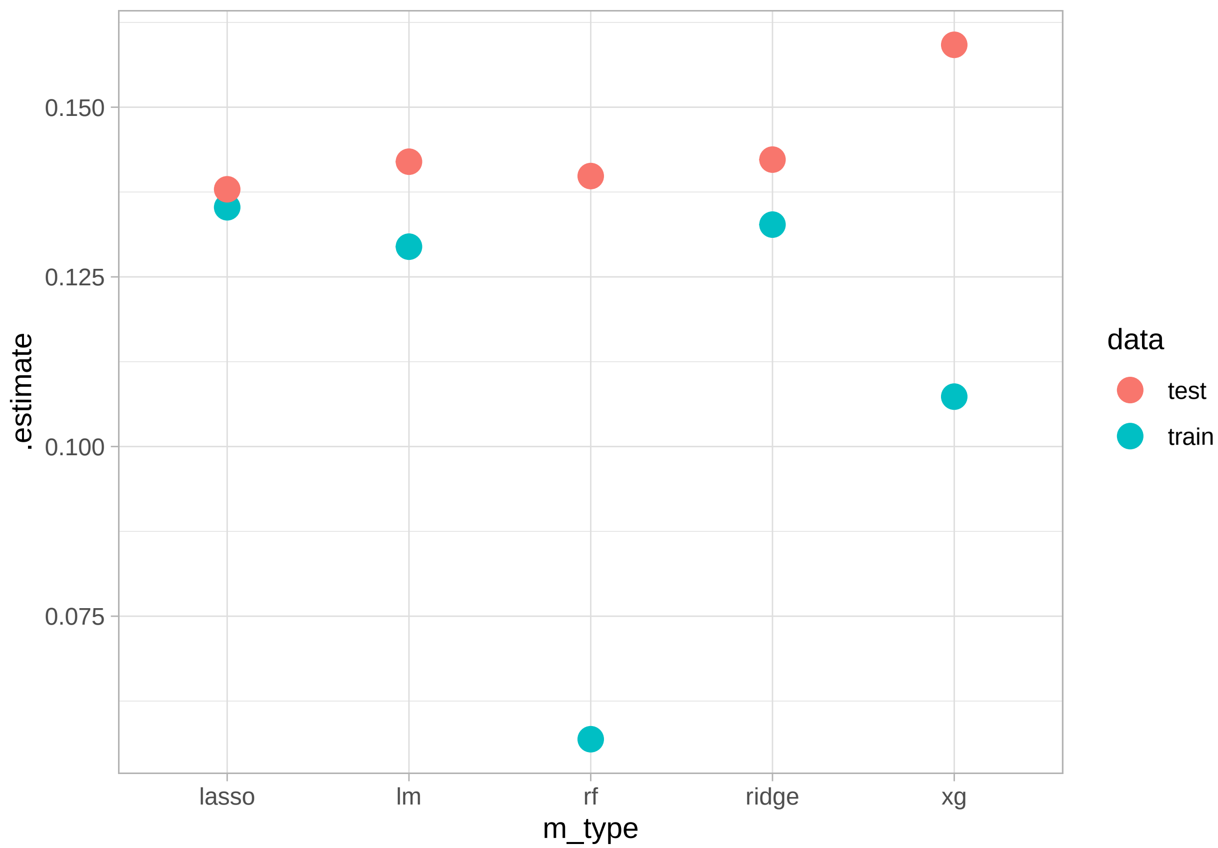

```
## Selecting by abs_importance
```


OverallQual
GrLivArea
GarageCars
YearRemodAdd
MSZoning_RL
X2ndFlrSF
HouseStyle_X2Story
Neighborhood_Edwards
Exterior1st_HdBoard
BsmtFullBath
BsmtExposure_Gd
BsmtFinType1_Unf
Exterior2nd_HdBoard
CentralAir_Y
KitchenQual_TA
SaleCondition_Partial
BsmtFinType2_Unf
GarageType_Detchd
FullBath
SaleCondition_Normal
Variable
0.00
0.05
0.10
Importance
Sign
NEG
POS



Evaluate and consolidate model metrics

```
## # A tibble: 5 x 4
##   .metric m_type  train  test
##   <chr>   <chr>   <dbl> <dbl>
## 1 rmse    lm     0.129  0.142
## 2 rmse    rf     0.0569 0.140
## 3 rmse    xg     0.107  0.159
## 4 rmse    ridge  0.133  0.142
## 5 rmse    lasso  0.135  0.138
```



Select Model

The lasso model gives the best test RMSE value.