Housing Prediction Documentation

Gaurav

28 February 2019

Background

This project involves **Sales Price prediction** for houses in Ames, Iowa.
The data has **79 variables** describing the different aspects of the houses.
The dataset comes from this kaggle competition.

Github Repository

Note: This Rmd File requires the workspace generated by the script. The script requires the dataset either from kaggle or the github repo.

Evaluation Metric The result is evaluated on the basis of **RMSE** between the **logarithm** of the predicted price and actual price.

Introduction

The competition gives us both a training and a test set. The test set gives us an RMSE when uploaded online.
For quick testing, I partitioned the training data into a training subset and validation set with 75% in the training. The true values of SalePrice for test set are not known but prediction performance can be checked by uploading predictions online.

KeySteps: 1. Read in Data and Explore 2. Impute missing Values 3. Choose most relevant columns and engineer New Features 4. Fit models on training subset and validation subset using different approaches 5. Evaluate Results and select promising approaches 6. Fit models on entire training data and predict final outcome 7. Upload to Kaggle and finalize methodolgy

The explanations for the columns are available in a separate data description file. (Available on Kaggle and The Github Rep)

Preprocessing

There are three types of data in the columns, categorical, ordinal and numeric. Their treatment is explained below with examples.

Categorical Data

MSZoning: Identifies the general zoning classification of the sale.

```
A    Agriculture
C    Commercial
FV   Floating Village Residential
I    Industrial
RH   Residential High Density
RL   Residential Low Density
RP   Residential Low Density Park
RM   Residential Medium Density
```

To use this type of data effectively with machine learning algorithms, we will encode it into boolean data. One column will be created for each category. Each column will identify whether that row corresponds to that category.

This will be done for each of the columns in original data and we will finally end up with many columns. Since it is not feasible to manually do this for each variable, it will be achieved using dynamic variable names.

Ordinal Data

ExterQual: Evaluates the quality of the material on the exterior

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

This data has natural ordering and is converted into a column of corresponding numerical values. That is "Po"=0 , "Fa"=1 etc.

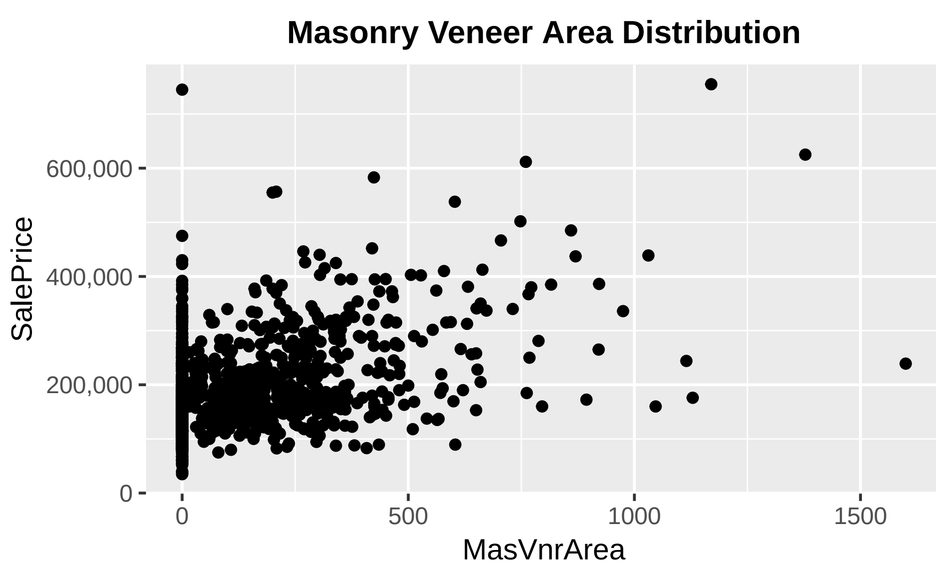
Numeric Data

GrLivArea: Above grade (ground) living area square feet

This type of data is generally used as is. However for neural-networks, these attributes were centered and scaled.

Imputation of Missing Values

I replace some of the missing values with the most common value when one value occurs much more frequently than others.




```
train$MasVnrArea[is.na(train$MasVnrArea)]<-0
train%>%group_by(SaleType)%>%summarize(count=n())
```

```
## # A tibble: 9 x 2
##   SaleType count
##   <fct>    <int>
## 1 COD         43
## 2 Con          2
## 3 ConLD        9
## 4 ConLI        5
## 5 ConLw        5
## 6 CWD          4
## 7 New        122
## 8 Oth          3
## 9 WD        1267
```

```
train$SaleType[is.na(train$SaleType)]<-"WD"
train%>%group_by(Functional)%>%summarize(count=n())
```

```
## # A tibble: 7 x 2
##   Functional count
##   <fct>      <int>
## 1 Maj1          14
## 2 Maj2           5
## 3 Min1          31
## 4 Min2          34
## 5 Mod           15
## 6 Sev            1
## 7 Typ         1360
```

```
train$Functional[is.na(train$Functional)]<-"Typ"
train%>%group_by(Exterior1st)%>%summarize(count=n())
```

```
## # A tibble: 15 x 2
##    Exterior1st count
##    <fct>       <int>
##  1 AsbShng        20
##  2 AsphShn         1
##  3 BrkComm         2
##  4 BrkFace        50
##  5 CBlock          1
##  6 CemntBd        61
##  7 HdBoard       222
##  8 ImStucc         1
##  9 MetalSd       220
## 10 Plywood       108
## 11 Stone           2
## 12 Stucco         25
## 13 VinylSd       515
## 14 Wd Sdng       206
## 15 WdShing        26
```

```
train$Exterior1st[is.na(train$Exterior1st)]<-"VinylSd"
train%>%group_by(MSZoning)%>%summarize(count=n())
```

```
## # A tibble: 5 x 2
##   MSZoning count
##   <fct>    <int>
## 1 C (all)     10
## 2 FV          65
## 3 RH          16
## 4 RL        1151
## 5 RM         218
```

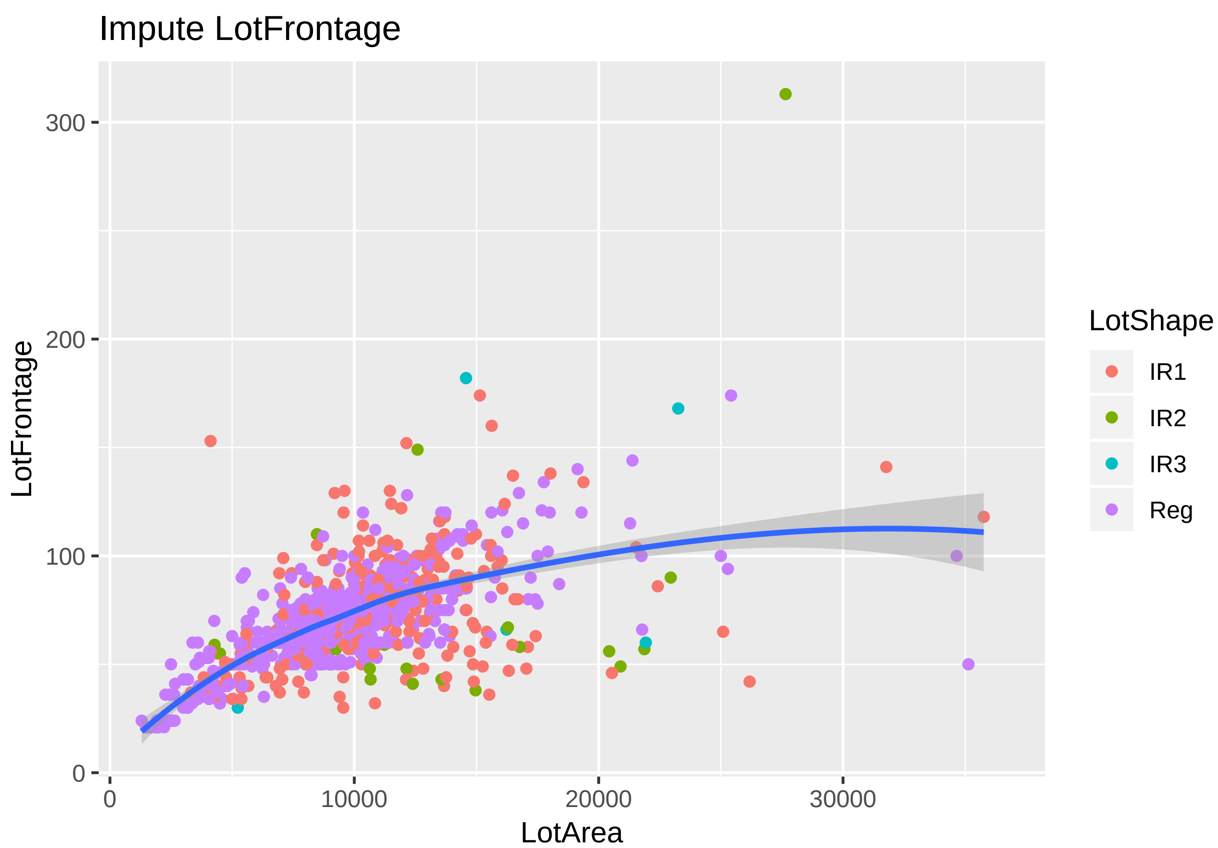
```
train$MSZoning[is.na(train$MSZoning)]<-"RL"
train%>%group_by(Electrical)%>%summarize(count=n())
```

```
## # A tibble: 6 x 2
##   Electrical count
##   <fct>      <int>
## 1 FuseA         94
## 2 FuseF         27
## 3 FuseP          3
## 4 Mix            1
## 5 SBrkr       1334
## 6 <NA>           1
```

```
train[is.na(train$Electrical),"Electrical"]<-"SBrkr"
```

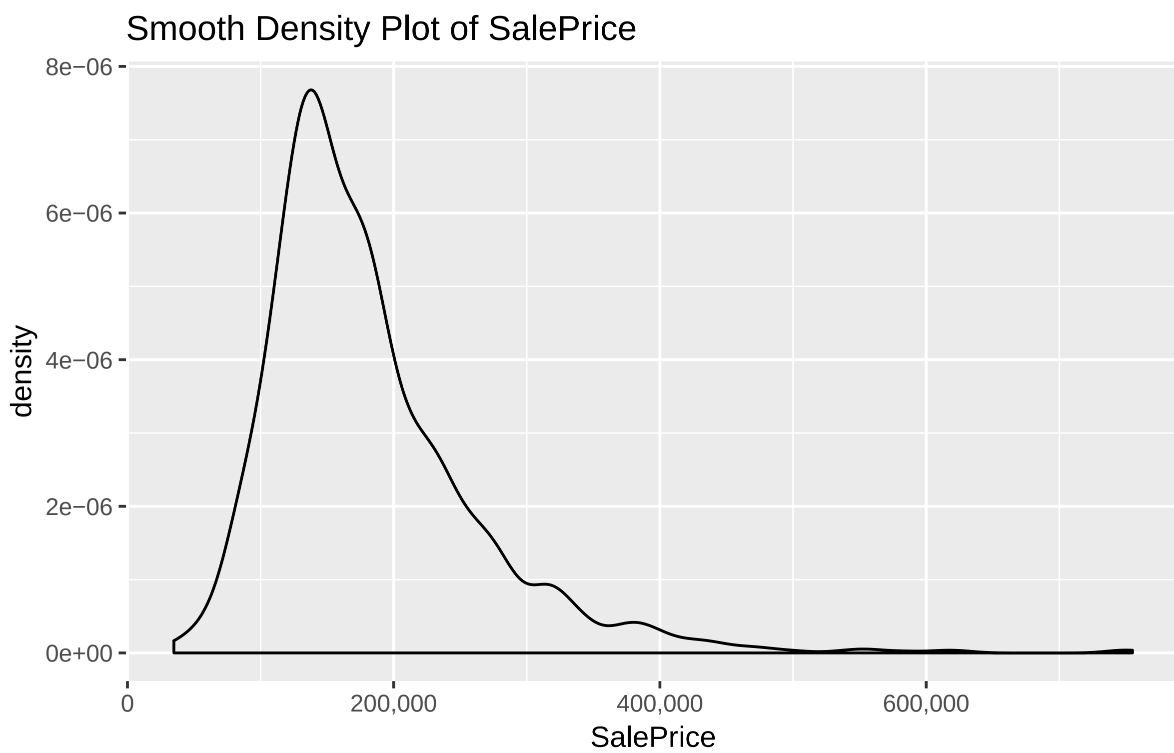
In some cases I suspect values are missing because condition is Not Applicable E.g: Garage Area is missing because Garage does not exist etc.

```
train$GarageArea[is.na(train$GarageArea)]<-0
train$GarageCars[is.na(train$GarageCars)]<-0
train[is.na(train$TotalBsmtSF),c("TotalBsmtSF","BsmtFinSF1","BsmtFinSF2","BsmtUnfSF")]<-0
train[is.na(train$BsmtFullBath),c("BsmtFullBath","BsmtHalfBath")]<-0
```

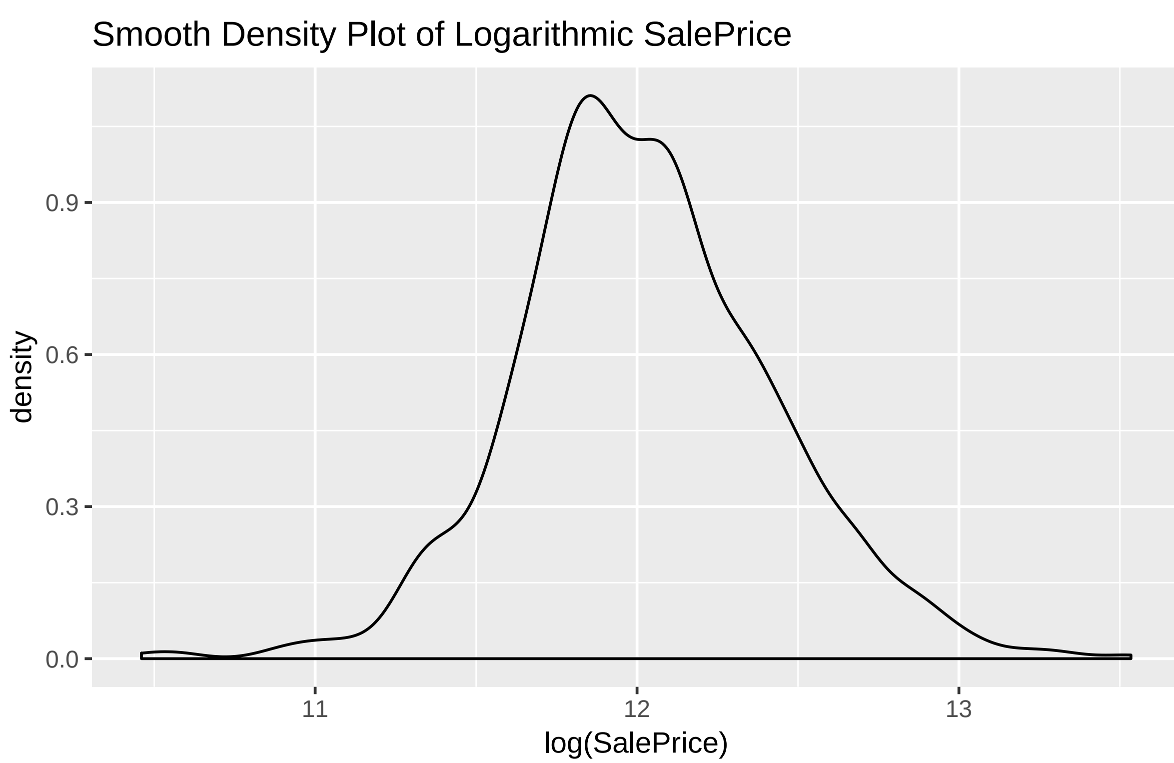


LotFrontage: Linear feet of street connected to property
There are a large number of missing values for LotFrontage.
I therefore fit a curve to estimate LotFrontage from LotArea. These variables are highly correlated. The different LotShapes do not have an obviously different ratio and LotShape was not used to improve the estimate.

Skewness of Target Variable

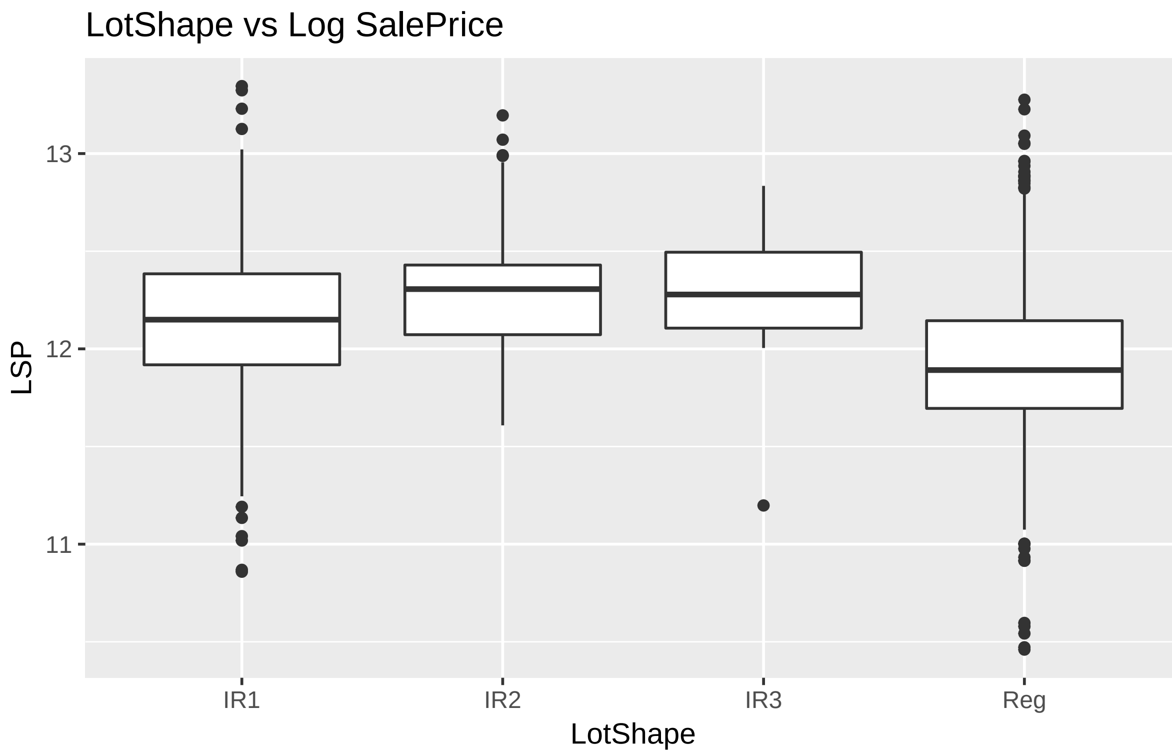


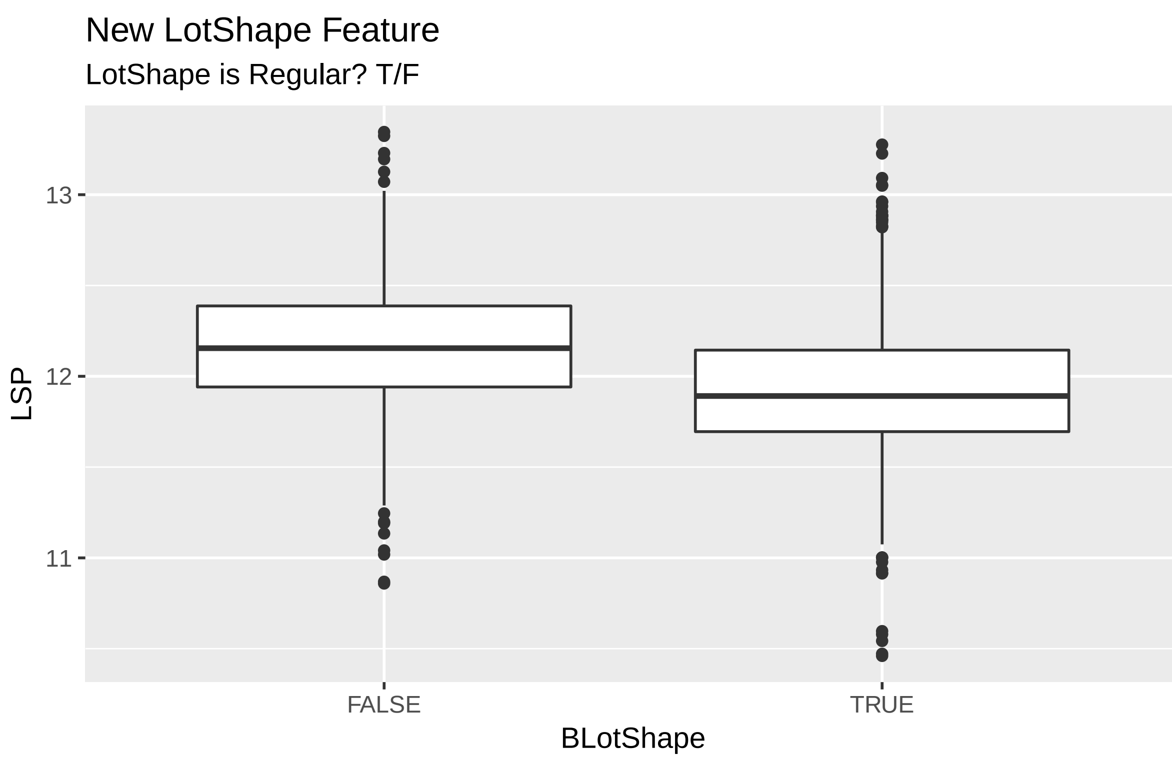
To account for this we take the log of the SalePrice and store it in a variable called LSP

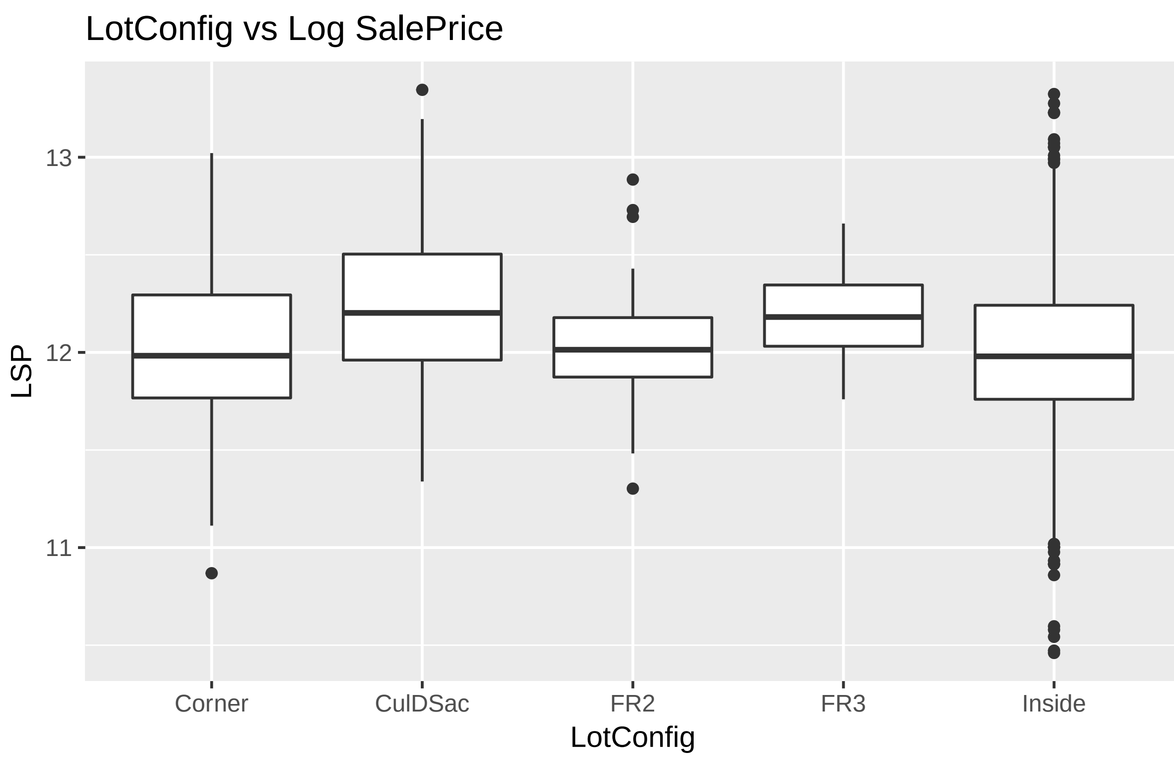


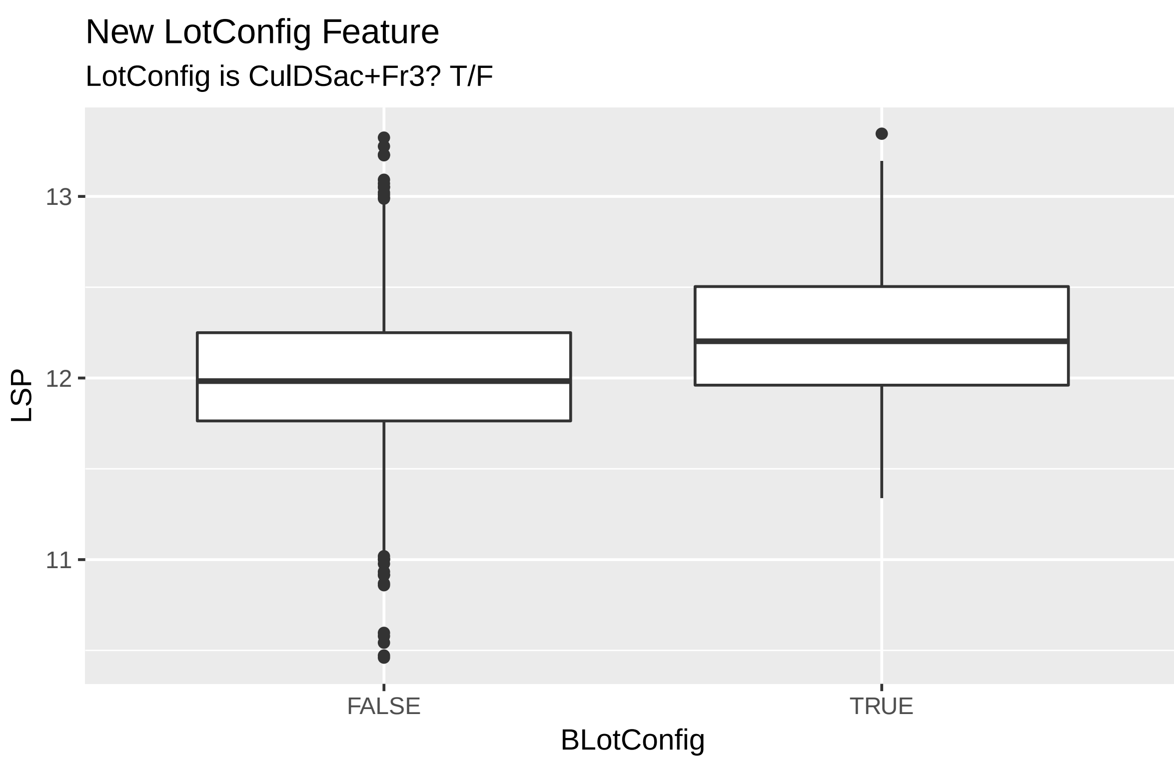
Data Exploration

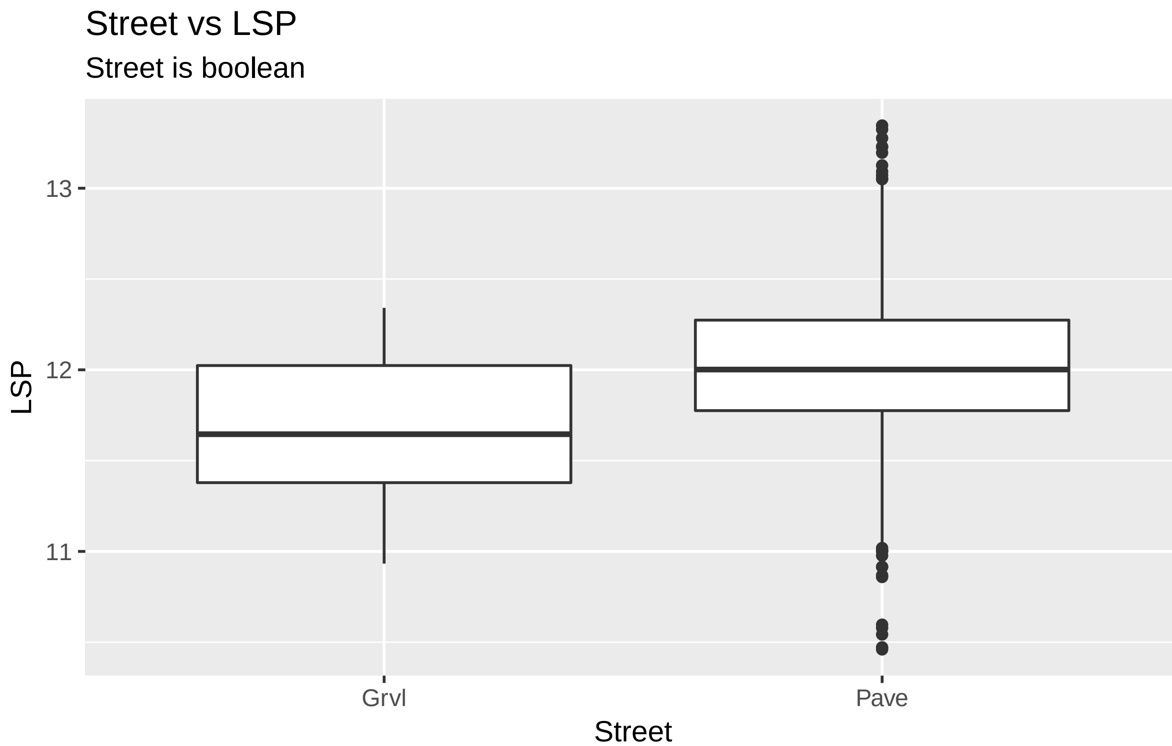
In this section I looked through the columns one by one. I used this to decide which variables could be useful and which were not. This is also the part where I converted the categorical data into separate features.

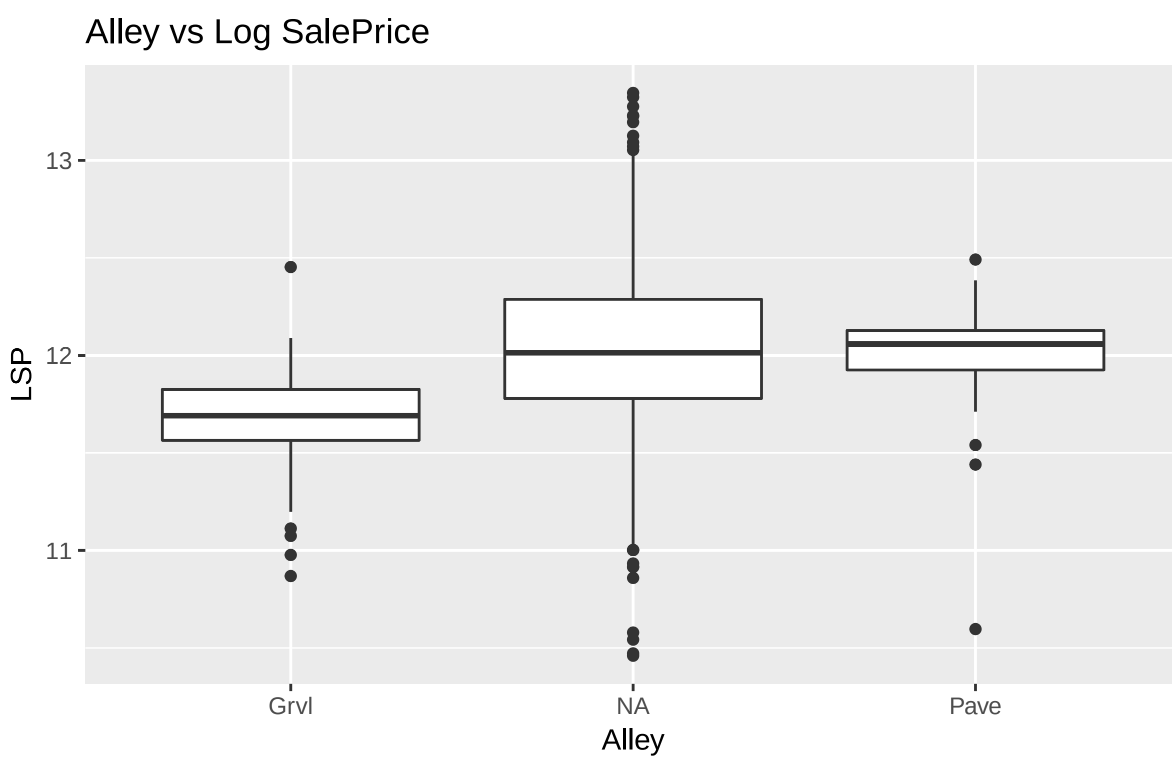


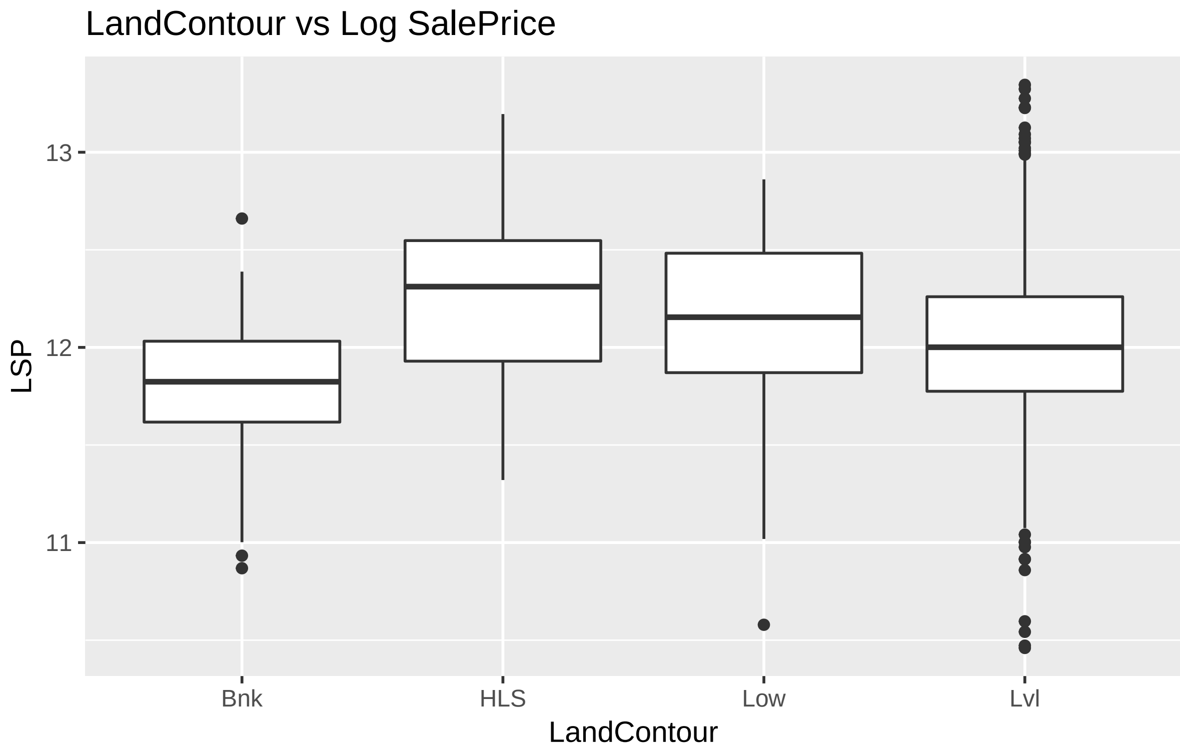


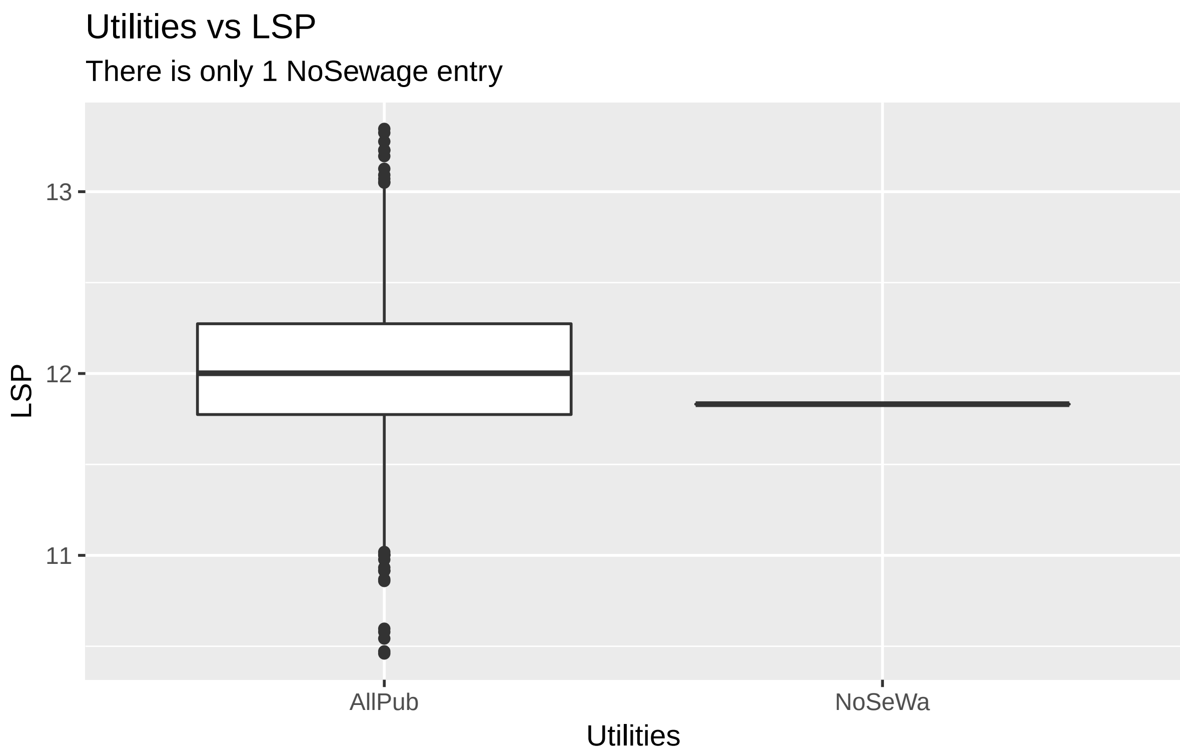

LotConfig vs Log SalePrice
LSP
13
12
11
Corner
CulDSac
FR2
FR3
Inside
LotConfig

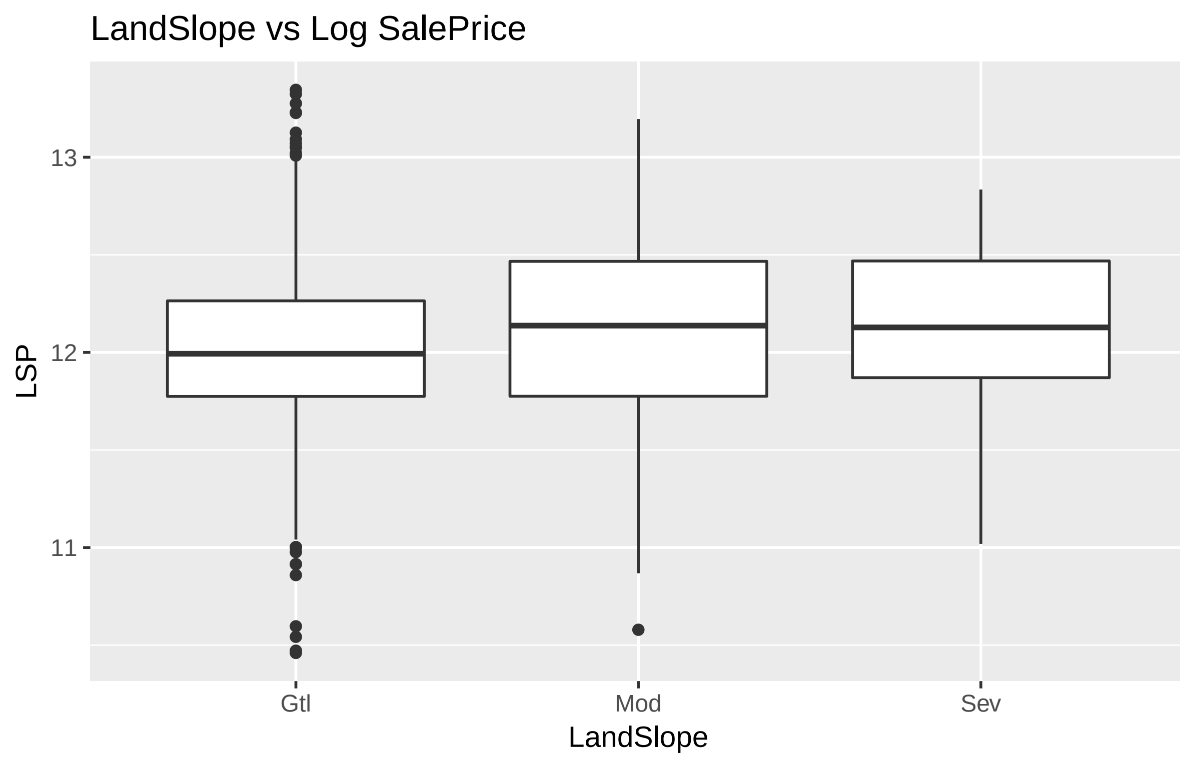

New LotConfig Feature
LotConfig is CulDSac+Fr3? T/F
LSP
13
12
11
FALSE
TRUE
BLotConfig

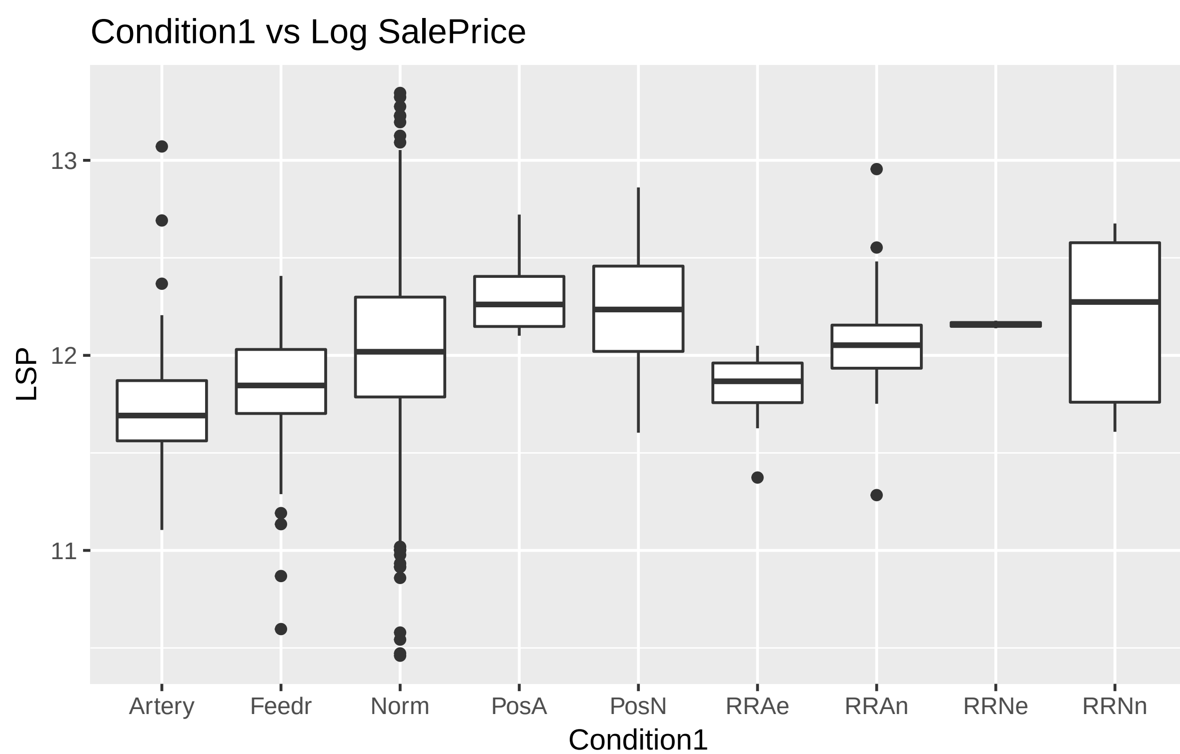

Street vs LSP
Street is boolean
LSP
13
12
11
Grvl
Pave
Street

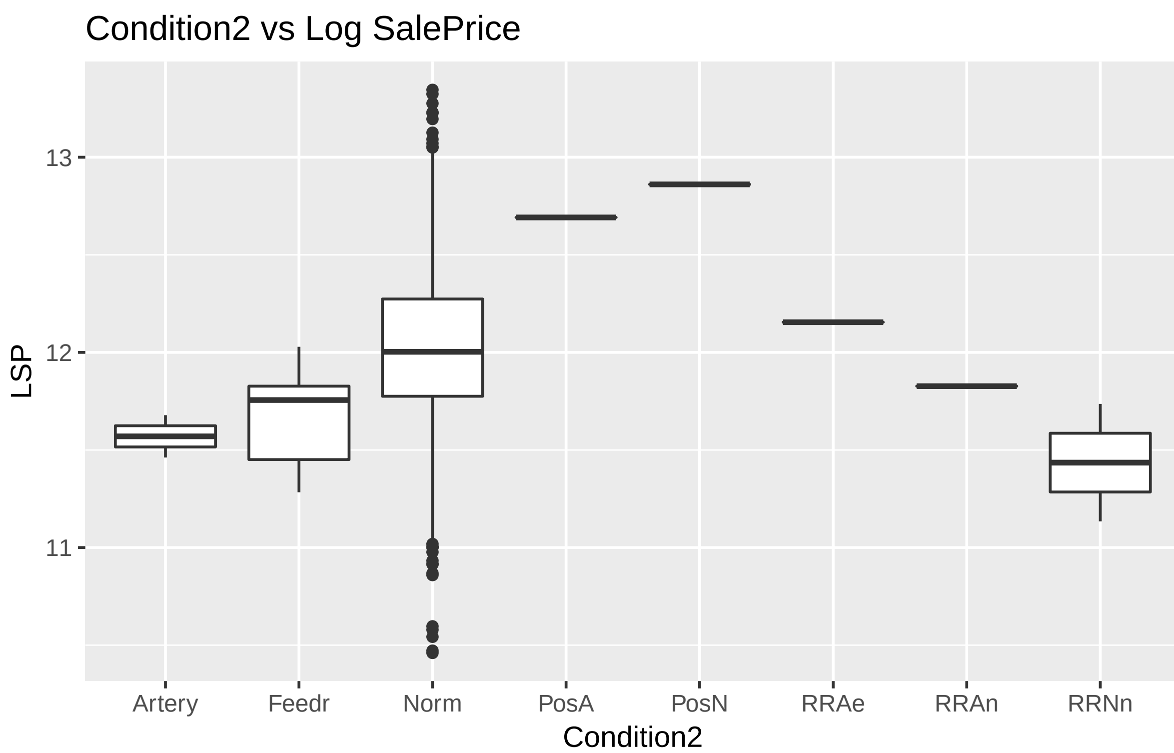

Alley vs Log SalePrice
LSP
13
12
11
Grvl
NA
Pave
Alley

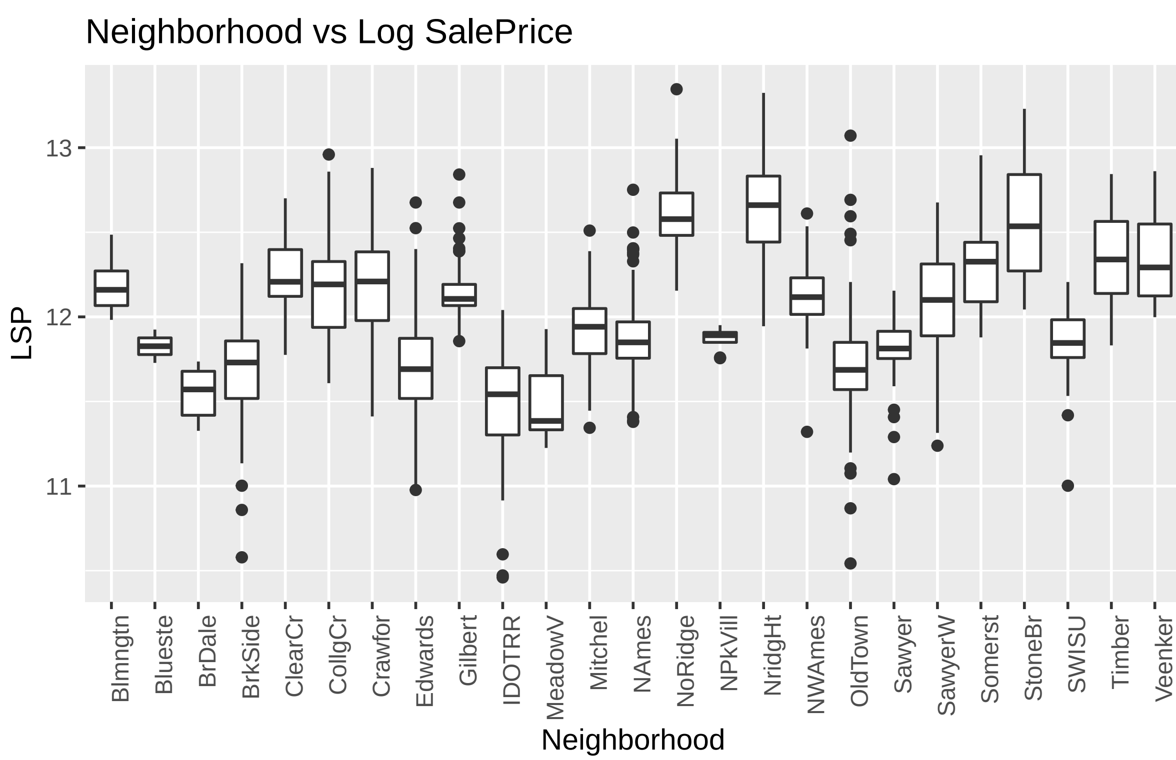

LandContour vs Log SalePrice
LSP
13
12
11
Bnk
HLS
Low
Lvl
LandContour

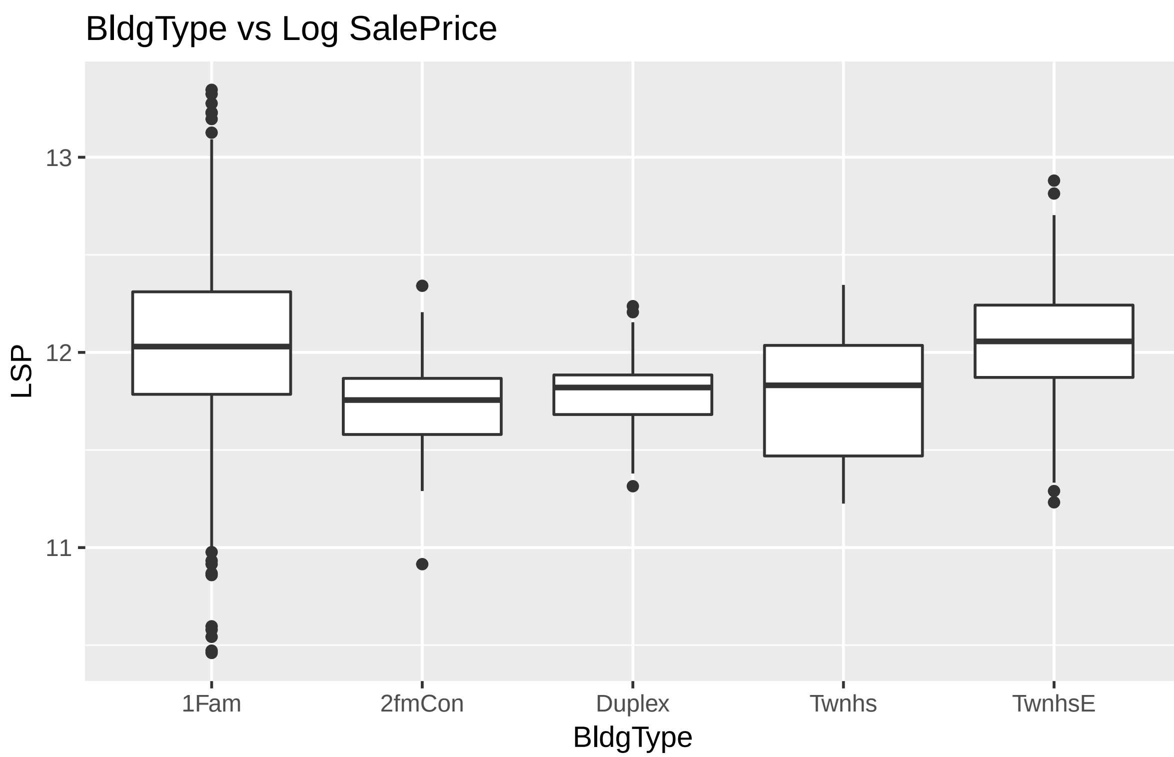

Utilities vs LSP
There is only 1 NoSewage entry
LSP
13
12
11
AllPub
NoSeWa
Utilities

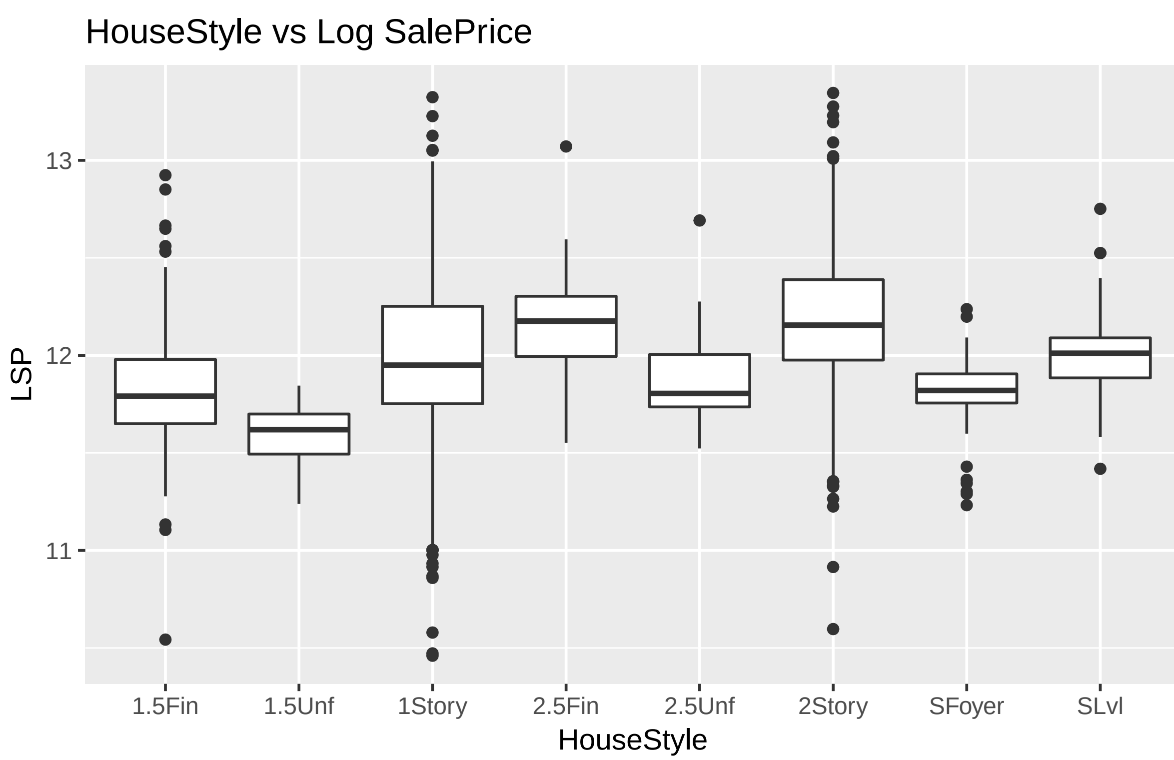

LandSlope vs Log SalePrice
LSP
13
12
11
Gtl
Mod
Sev
LandSlope

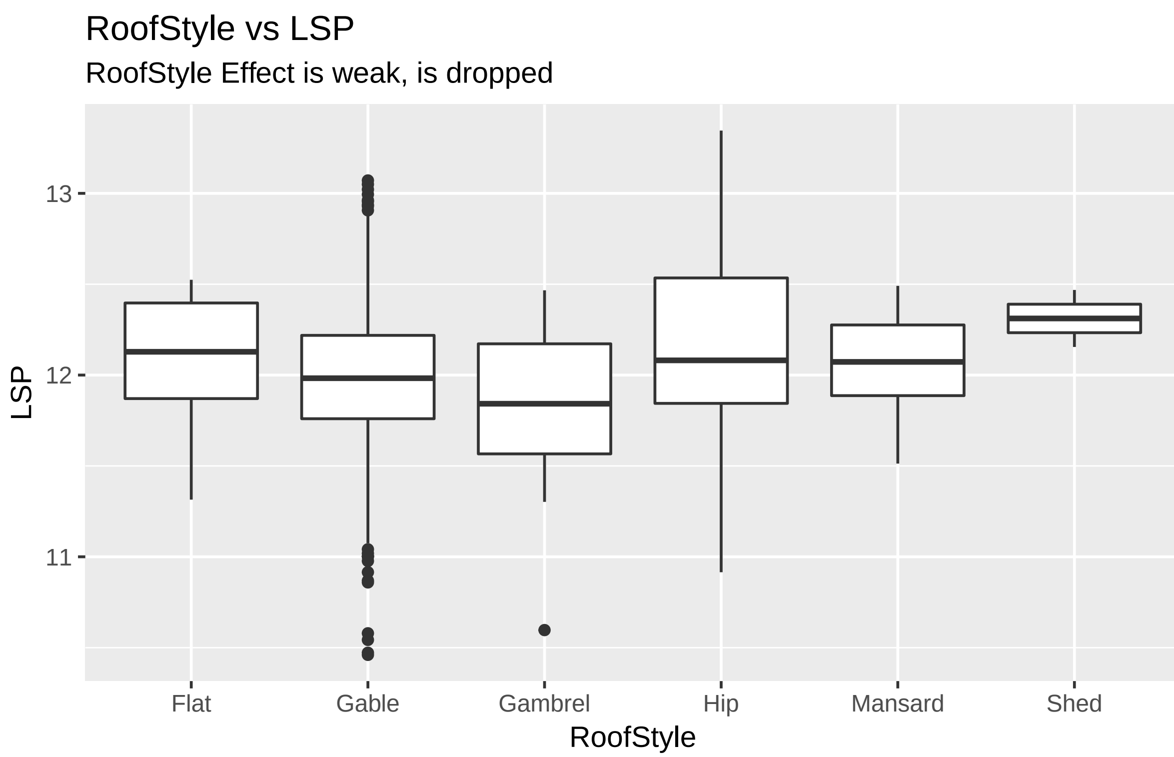

Condition1 vs Log SalePrice
LSP
13
12
11
Artery
Feedr
Norm
PosA
PosN
RRAe
RRAn
RRNe
RRNn
Condition1

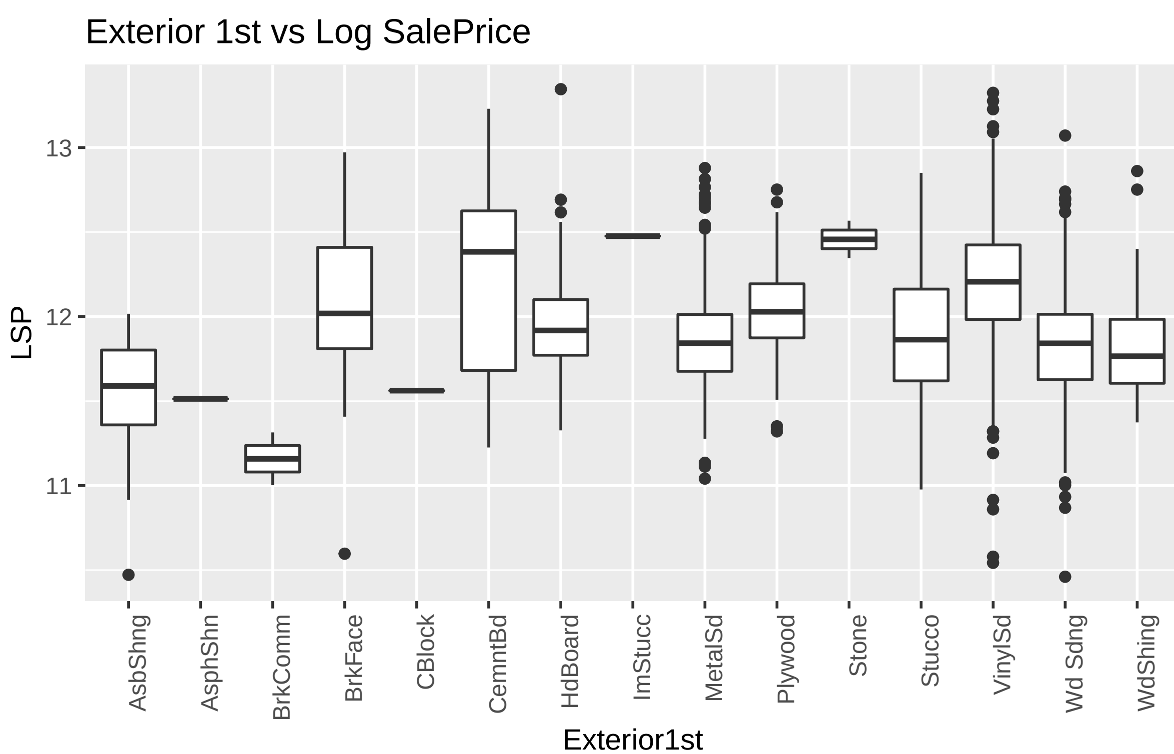

Condition2 vs Log SalePrice
LSP
13
12
11
Artery
Feedr
Norm
PosA
PosN
RRAe
RRAn
RRNn
Condition2

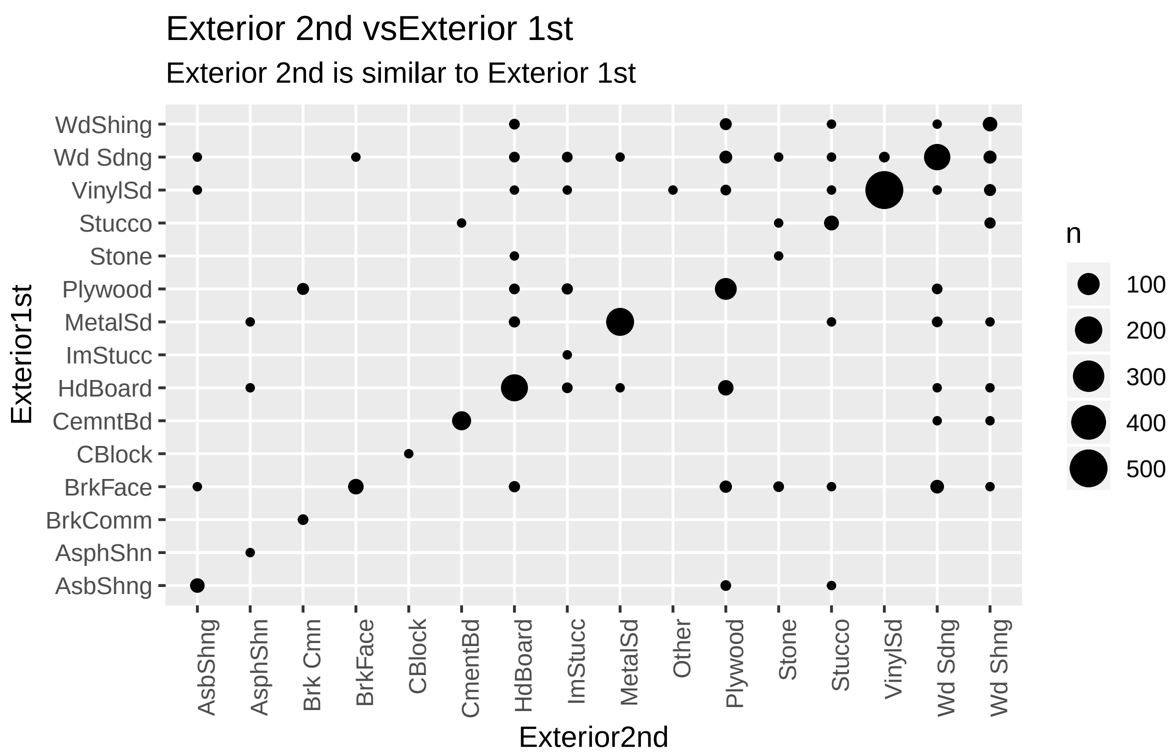

Neighborhood vs Log SalePrice
LSP
13
12
11
Blmngtn
Blueste
BrDale
BrkSide
ClearCr
CollgCr
Crawfor
Edwards
Gilbert
IDOTRR
MeadowV
Mitchel
NAmes
NoRidge
NPkVill
NridgHt
NWAmes
OldTown
Sawyer
SawyerW
Somerst
StoneBr
SWISU
Timber
Veenker
Neighborhood

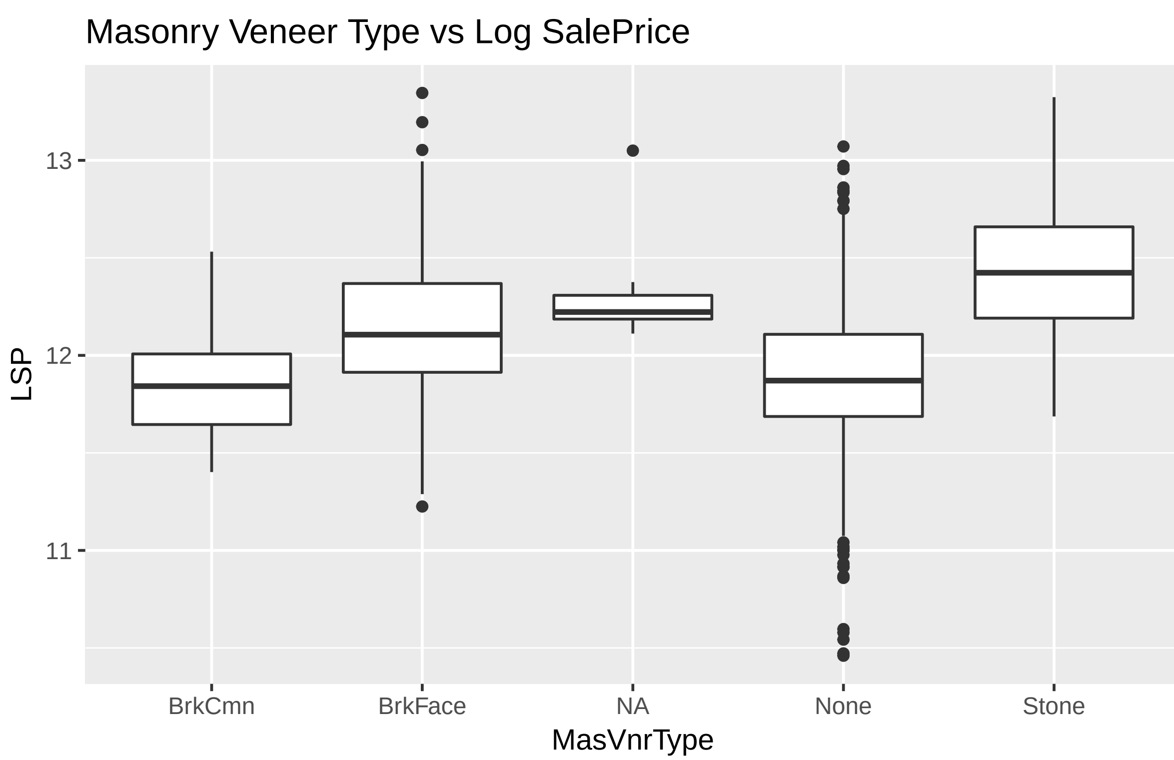

BldgType vs Log SalePrice
LSP
13
12
11
1Fam
2fmCon
Duplex
Twnhs
TwnhsE
BldgType

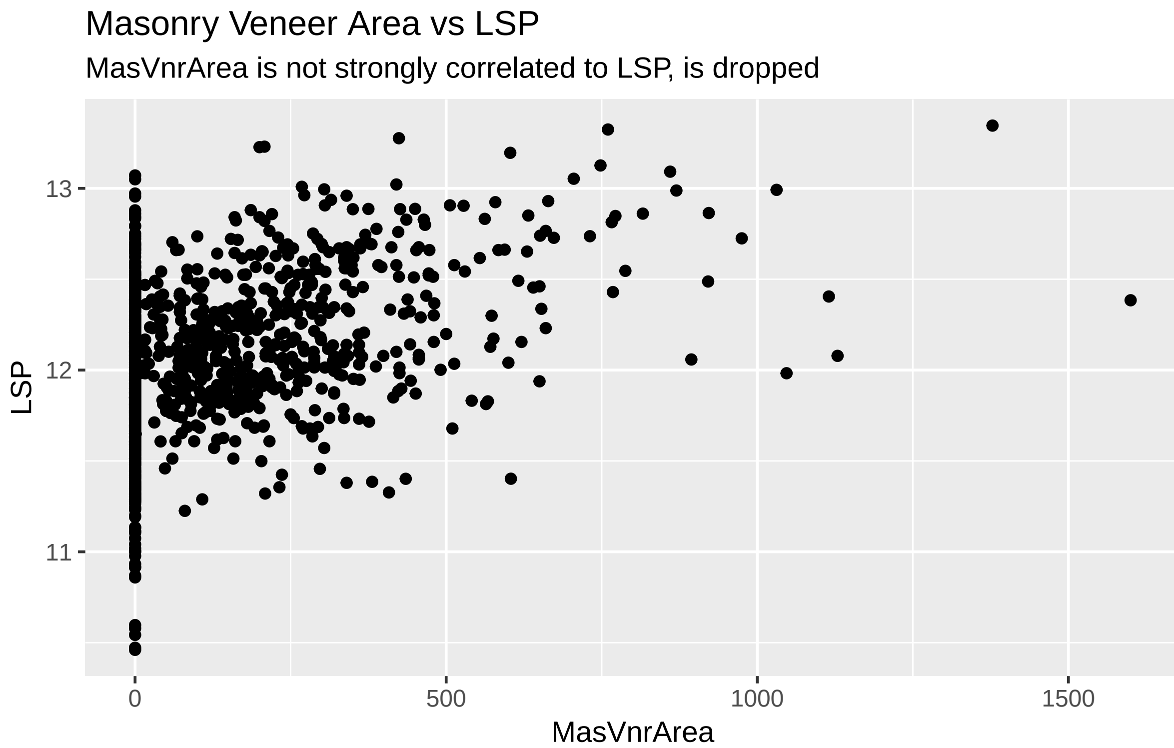

HouseStyle vs Log SalePrice
LSP
13
12
11
1.5Fin
1.5Unf
1Story
2.5Fin
2.5Unf
2Story
SFoyer
SLvl
HouseStyle

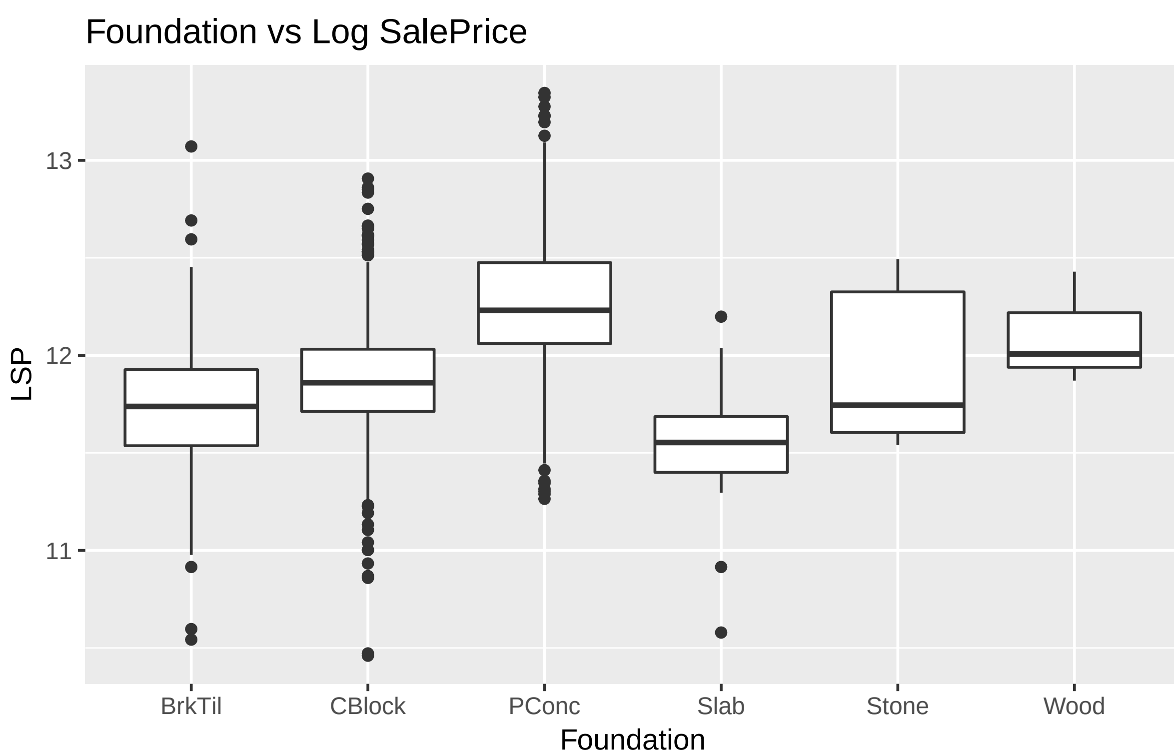

RoofStyle vs LSP
RoofStyle Effect is weak, is dropped
LSP
13
12
11
Flat
Gable
Gambrel
Hip
Mansard
Shed
RoofStyle

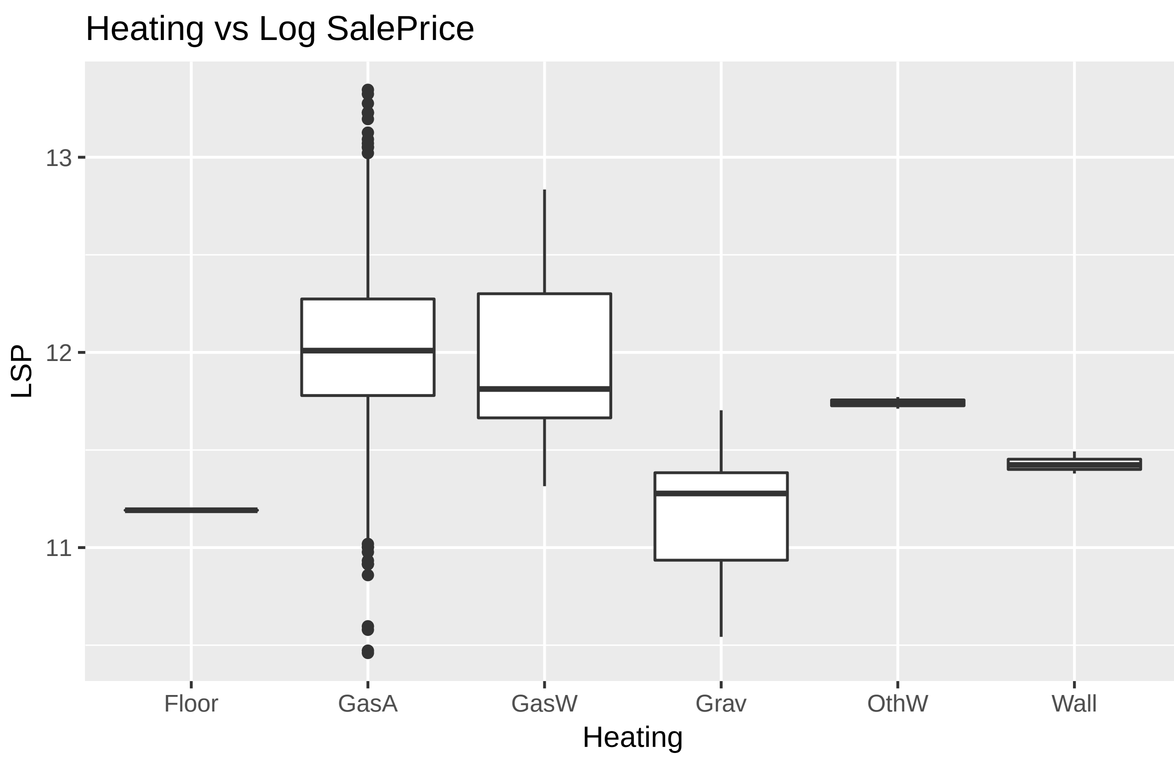

Exterior 1st vs Log SalePrice
LSP
13
12
11
AsbShng
AsphShn
BrkComm
BrkFace
CBlock
CemntBd
HdBoard
ImStucc
MetalSd
Plywood
Stone
Stucco
VinylSd
Wd Sdng
WdShing
Exterior1st

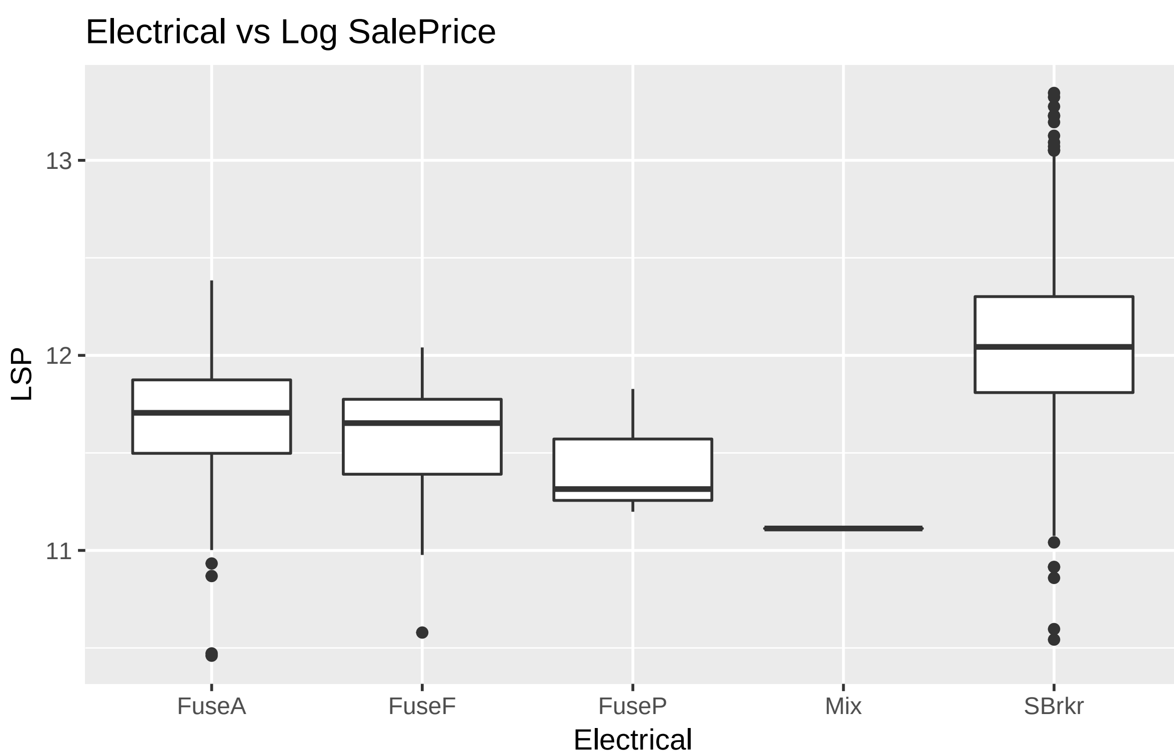

Exterior 2nd vsExterior 1st
Exterior 2nd is similar to Exterior 1st
Exterior1st
WdShing
Wd Sdng
VinylSd
Stucco
Stone
Plywood
MetalSd
ImStucc
HdBoard
CemntBd
CBlock
BrkFace
BrkComm
AsphShn
AsbShng
Exterior2nd
AsbShng
AsphShn
Brk Cmn
BrkFace
CBlock
CmentBd
HdBoard
ImStucc
MetalSd
Other
Plywood
Stone
Stucco
VinylSd
Wd Sdng
Wd Shng
n
100
200
300
400
500

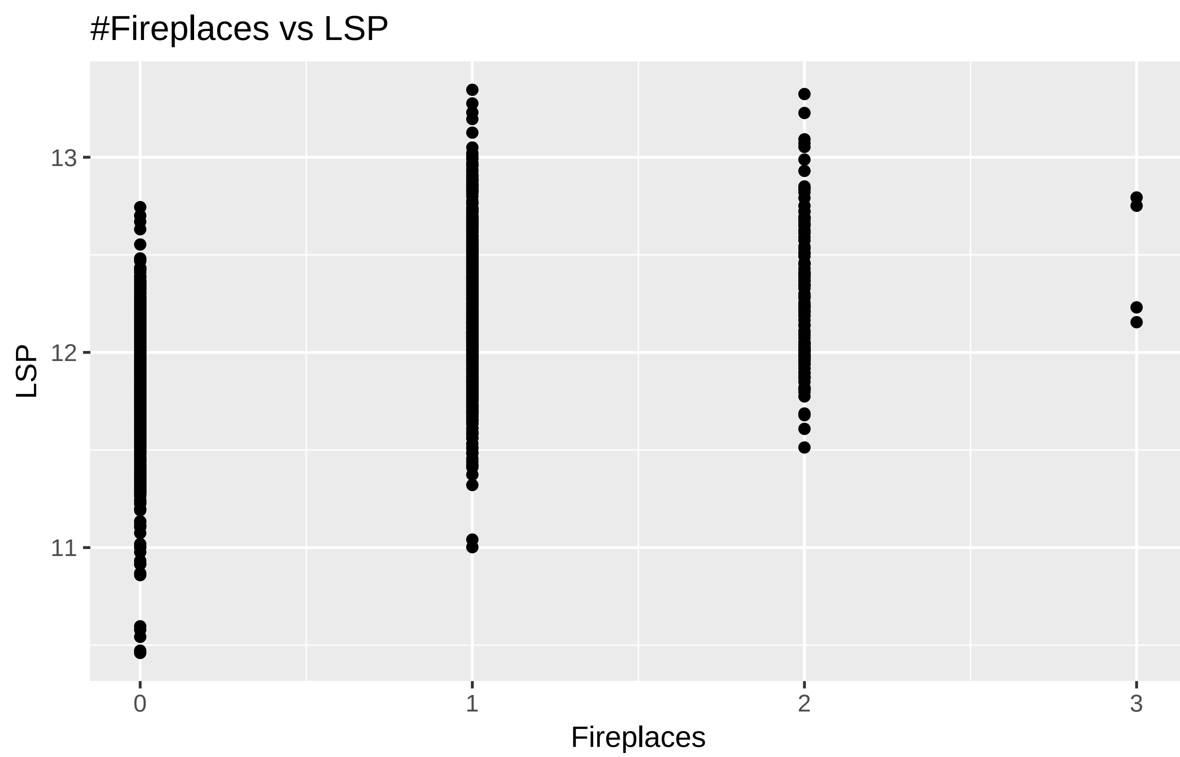

Masonry Veneer Type vs Log SalePrice
LSP
13
12
11
BrkCmn
BrkFace
NA
None
Stone
MasVnrType

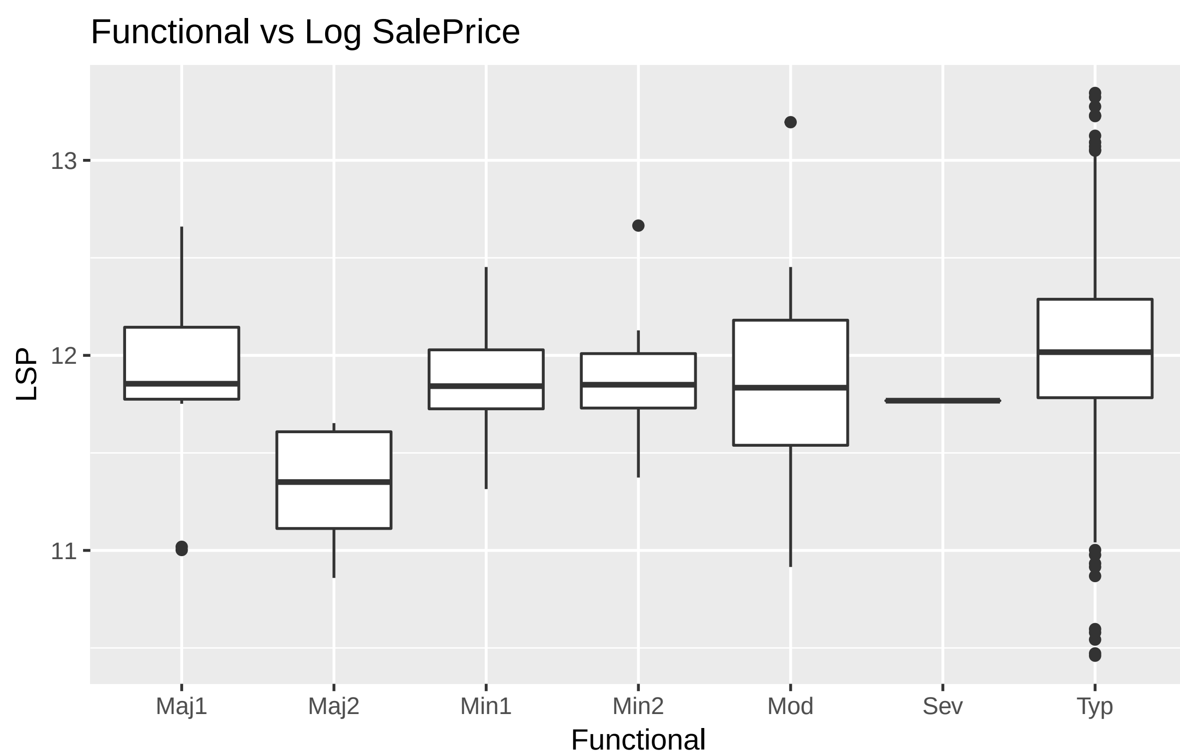

Masonry Veneer Area vs LSP
MasVnrArea is not strongly correlated to LSP, is dropped
LSP
13
12
11
0
500
1000
1500
MasVnrArea

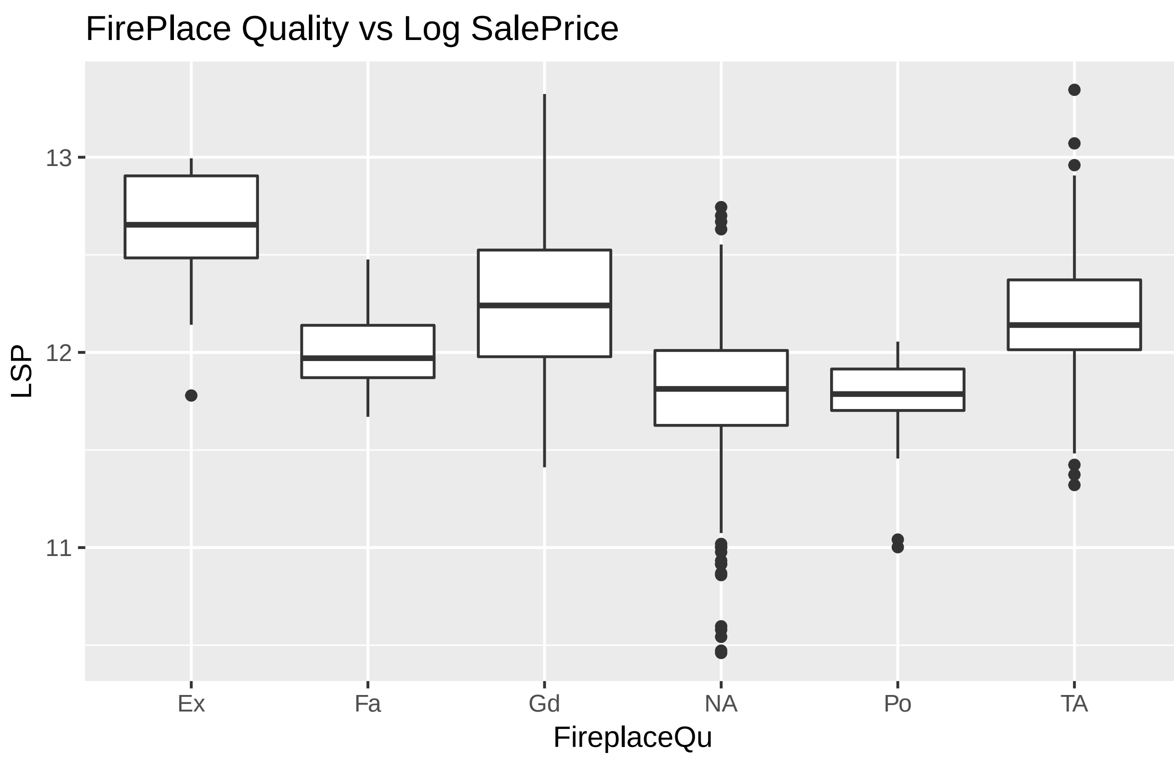

Foundation vs Log SalePrice
LSP
13
12
11
BrkTil
CBlock
PConc
Slab
Stone
Wood
Foundation

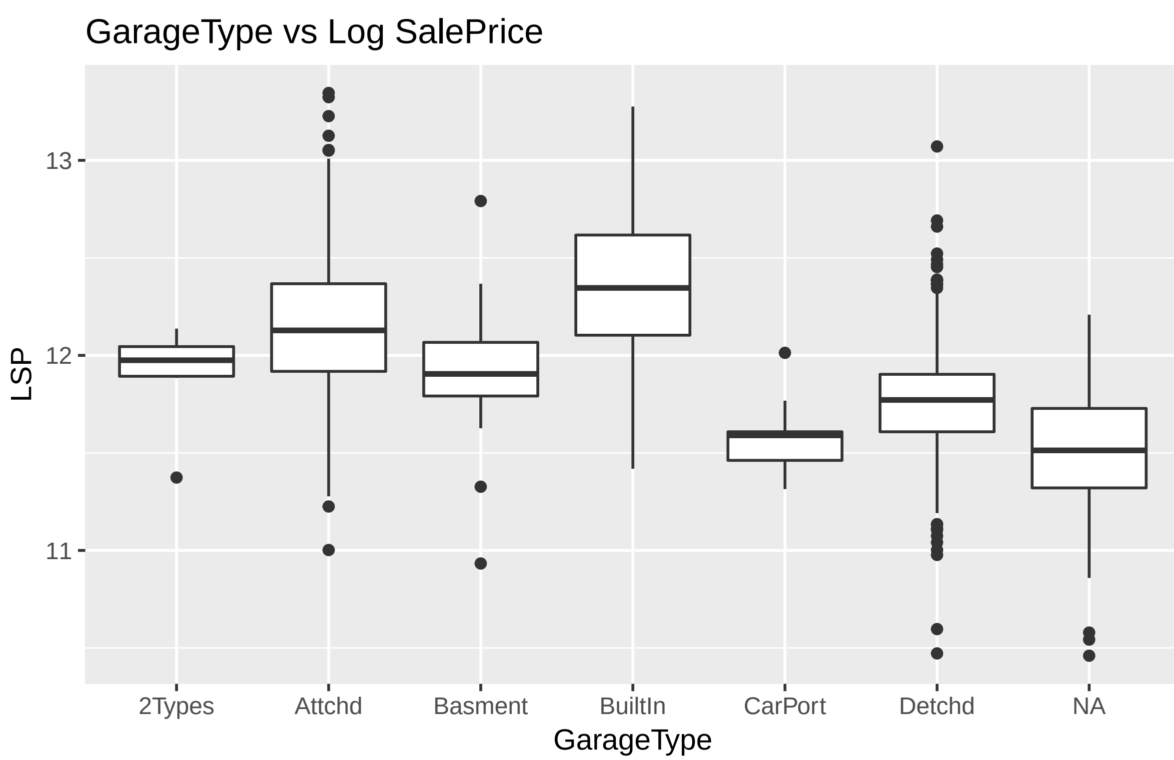

Heating vs Log SalePrice
LSP
13
12
11
Floor
GasA
GasW
Grav
OthW
Wall
Heating


Electrical vs Log SalePrice
LSP
13
12
11
FuseA
FuseF
FuseP
Mix
SBrkr
Electrical

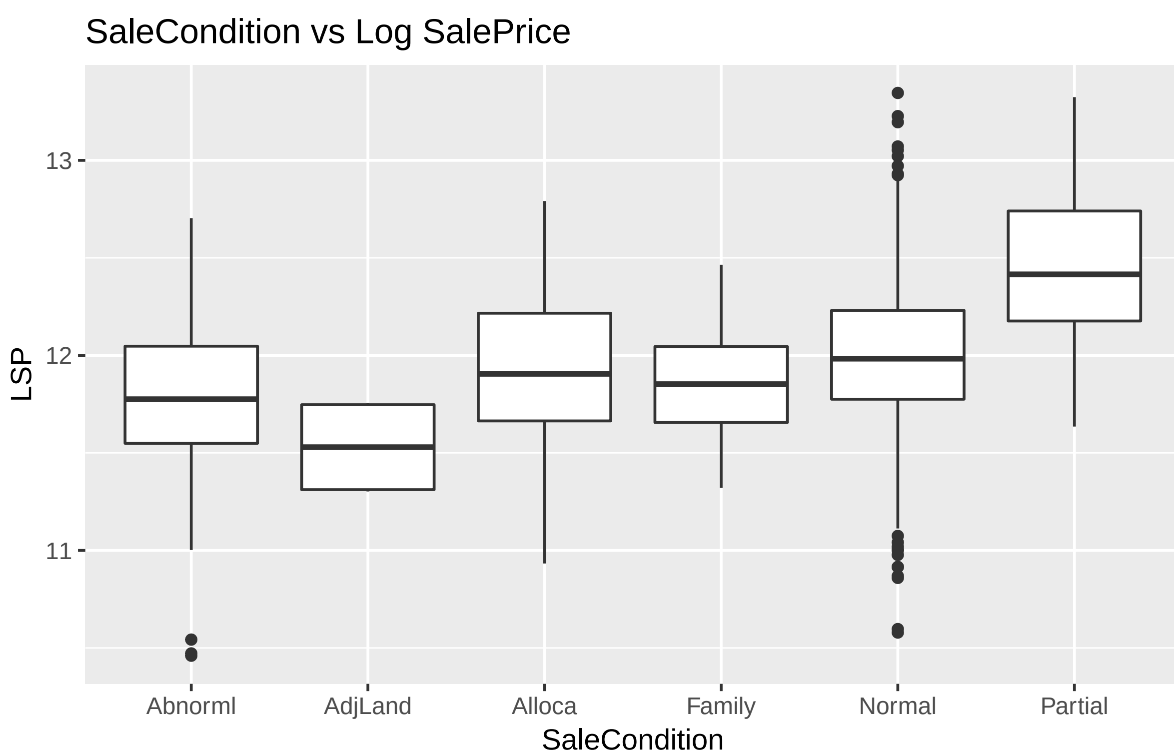

#Fireplaces vs LSP
LSP
13
12
11
0
1
2
3
Fireplaces

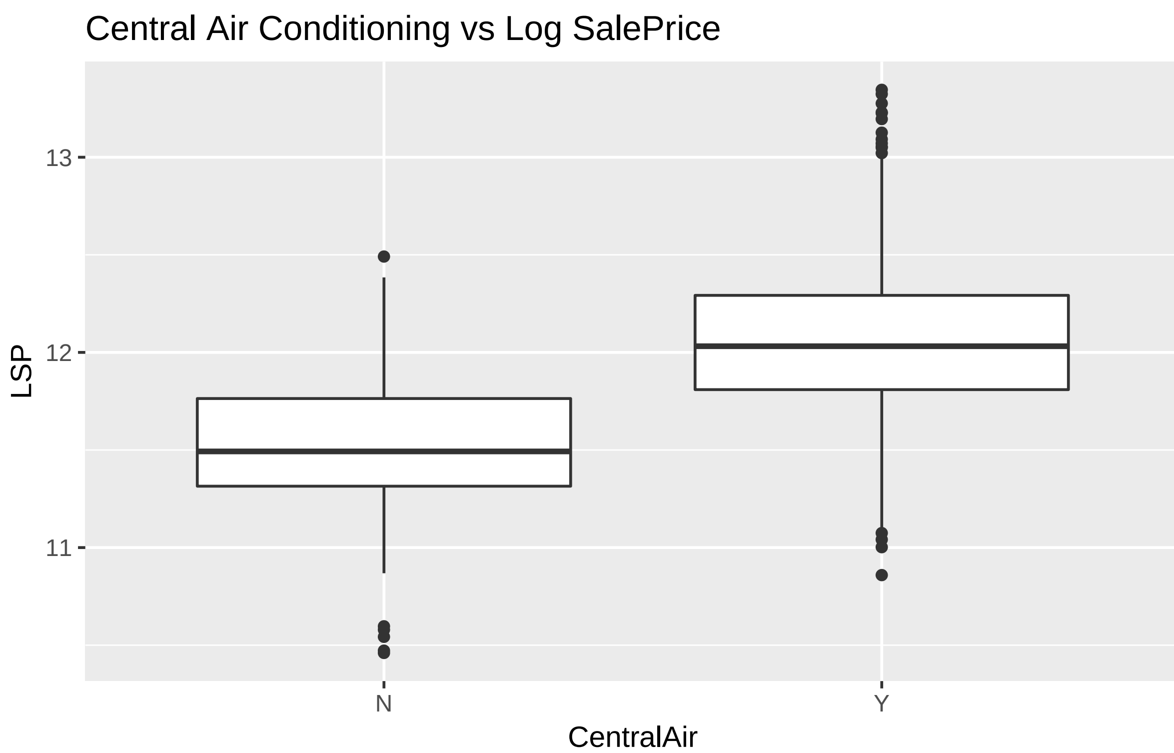

Functional vs Log SalePrice
LSP
13
12
11
Maj1
Maj2
Min1
Min2
Mod
Sev
Typ
Functional

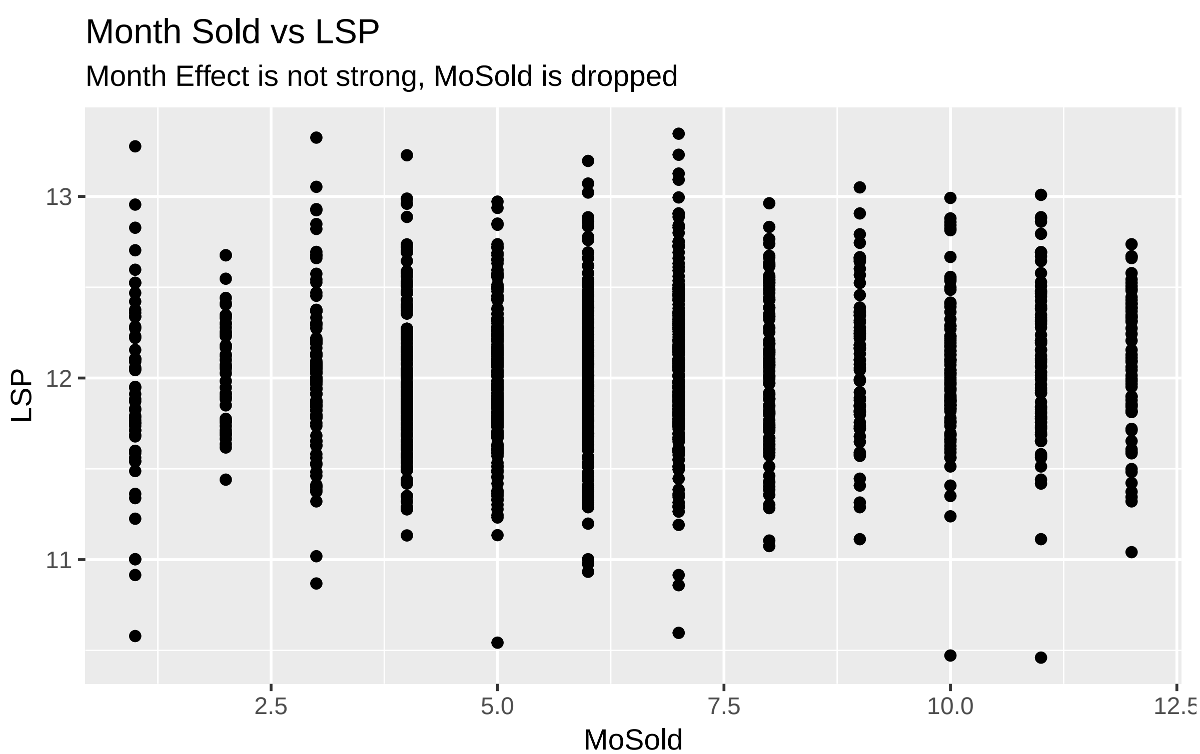

FirePlace Quality vs Log SalePrice
LSP
13
12
11
Ex
Fa
Gd
NA
Po
TA
FireplaceQu


GarageType vs Log SalePrice
LSP
13
12
11
2Types
Attchd
Basment
BuiltIn
CarPort
Detchd
NA
GarageType


SaleType vs Log SalePrice
LSP
13
12
11
COD
Con
ConLD
ConLI
ConLw
CWD
New
Oth
WD
SaleType

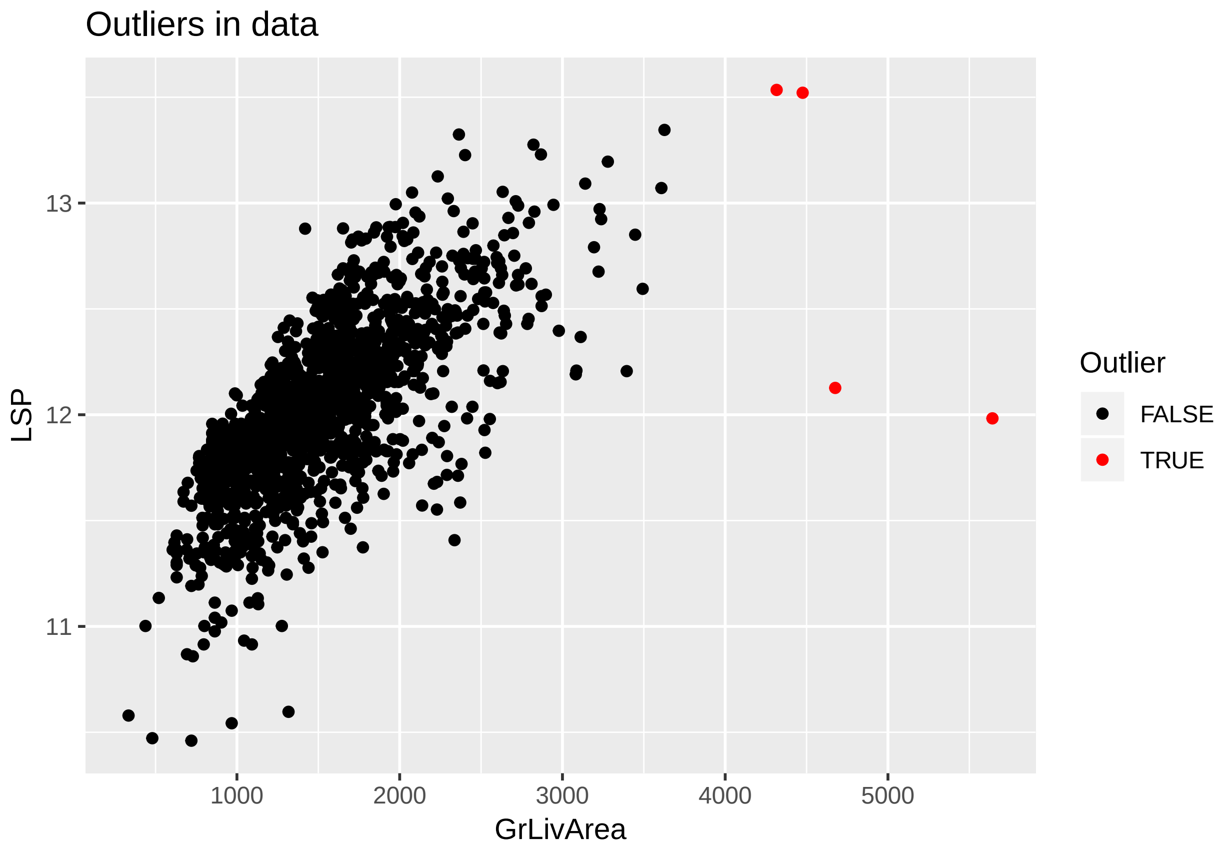

SaleCondition vs Log SalePrice
LSP
13
12
11
Abnorml
AdjLand
Alloca
Family
Normal
Partial
SaleCondition





Outliers

There are a few Outliers in data which may impact the model fit. These 4 values are removed before further analysis. Online documentation by author of this dataset confirmed that these data points may not be representative and should be discarded.



Data Analysis and Results

Several Different Models were fit on training subset.
Of these following Models resulted in promising results:
1. Linear Model RMSE= 0.1188609
2. Random Forest RMSE= 0.1229782
3. GBM RMSE= 0.1156895
4. BRNN RMSE= 0.1184115

Averaging the result of these four models resulted in an incremental improvemnt in result. The RMSE of average prediction for above 4 models is 0.1089506 on validation subset.

The final RMSE on test subset from kaggle is **0.12570**. This score puts the answer at about rank 1500/4100+ teams on Kaggle.

A further improvement in score(small) is possible by replacing BRNN with QRNN. Since QRNN takes significantly longer to train, I have left it out of the report.

Observations and conclusions

The RMSE from validation is slightly lower than that from the test set reported by kaggle. This indicates a slight overfit. In this regard, the Random Forest is closest to the actual value indicating its robustness to overfitting.

The Linear Model was surprisingly effective. In a real world scenario this is probably the best model I would use because it would be easy to interpret and use in a practical scenario. It is possible that on the ground, humans tend to value houses in some approximation of a linear model i.e. paying a fixed price per Sq.Ft depending on quality and locality etc. This may explain why the linear model works well.

For example from the model, it seems that a metal siding is worth about 20% more than wood siding for the Exterior which is covering the house.

From the Tree Fit we can get the variable importance. We can see that the most important factors are the Overall Quality and the Above ground Living Area.