

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 methodology

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.

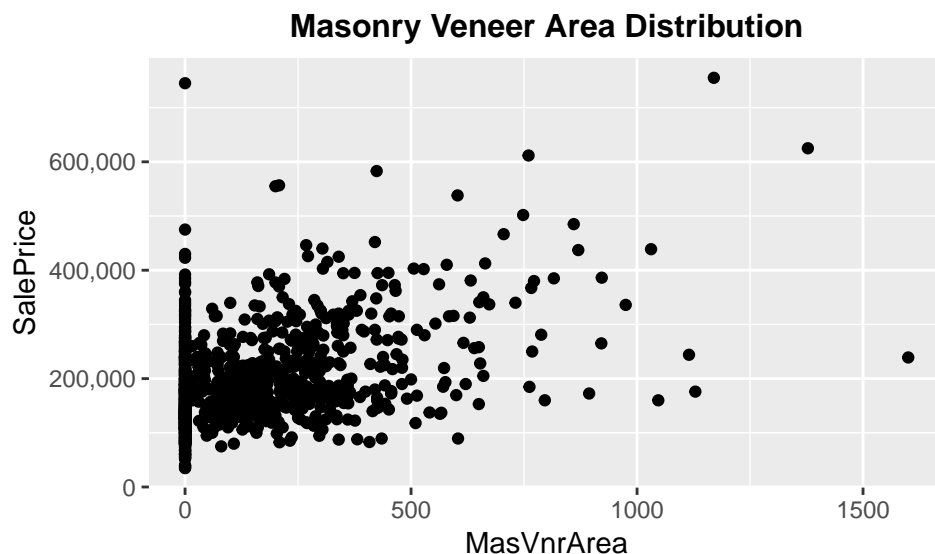
Numeric Data

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
```

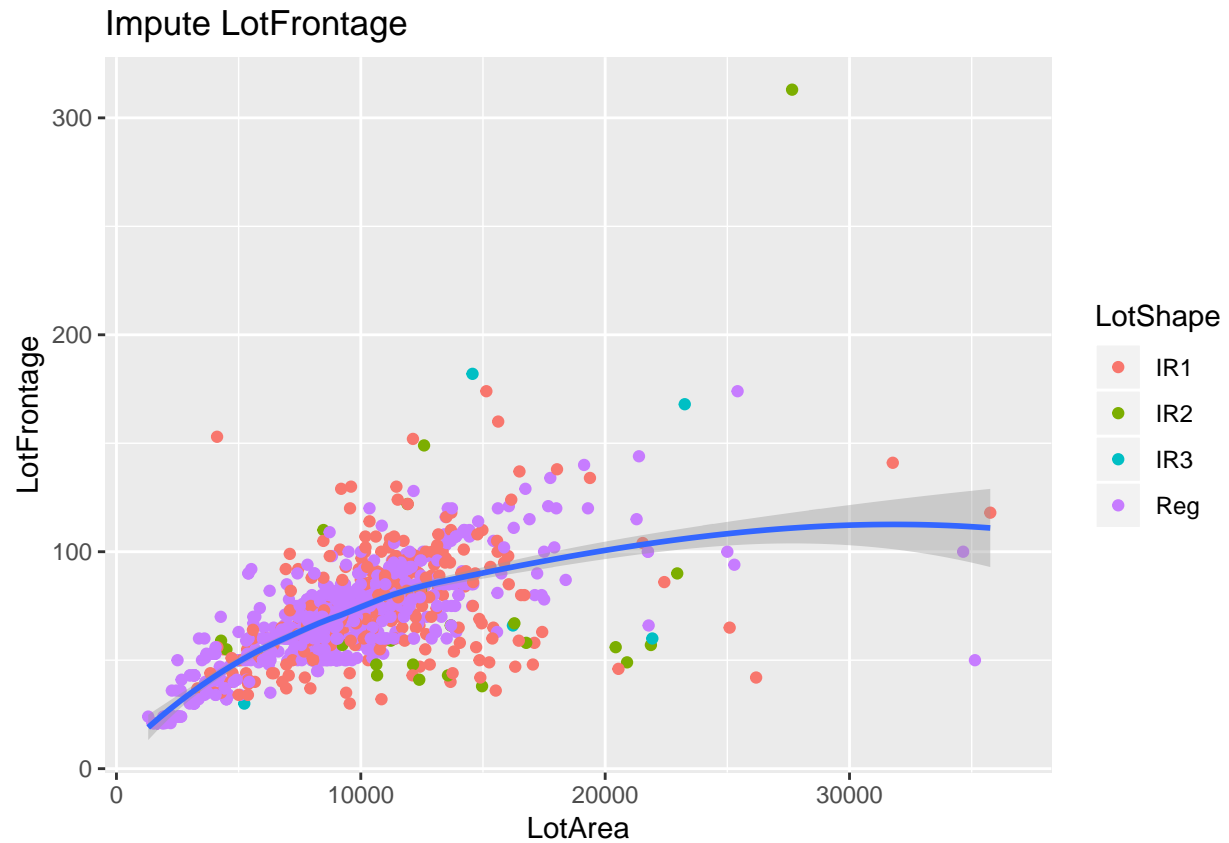
```
train$MSZoning[is.na(train$MSZoning)]<-"RL"
train%>%group_by(Electrical)%>%summarize(count=n())
```

```
## # 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
```

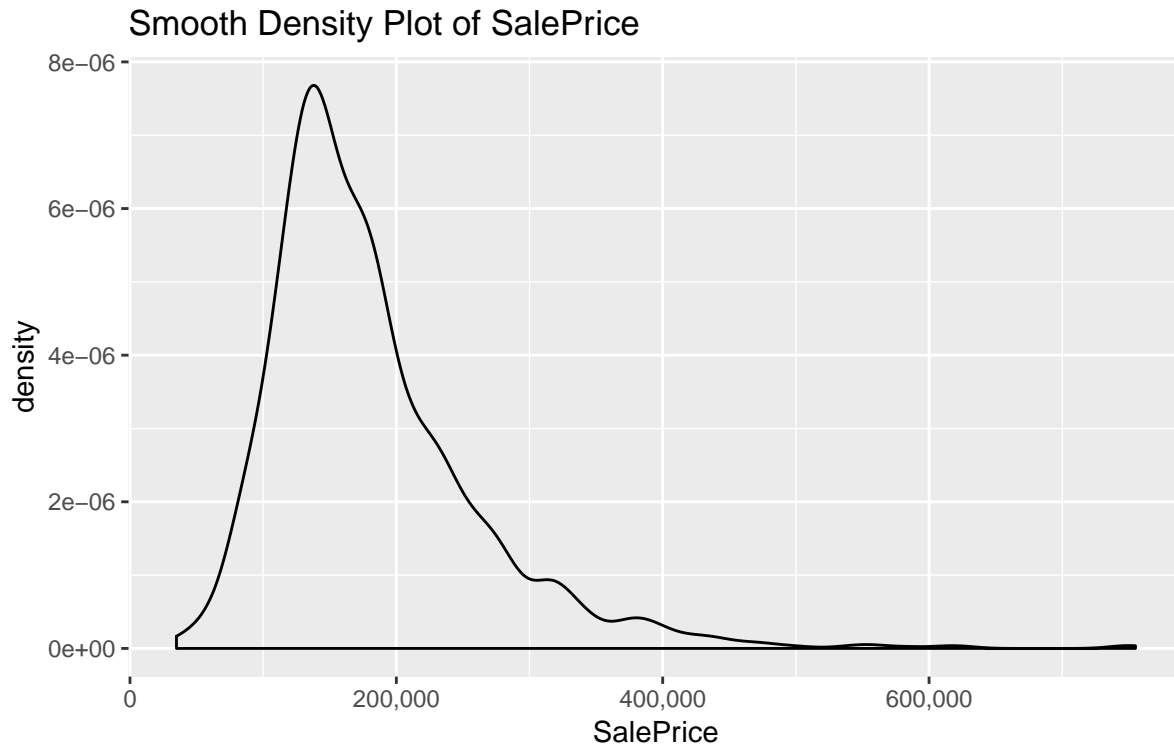


LotFrontage: Linear feet of street connected to property

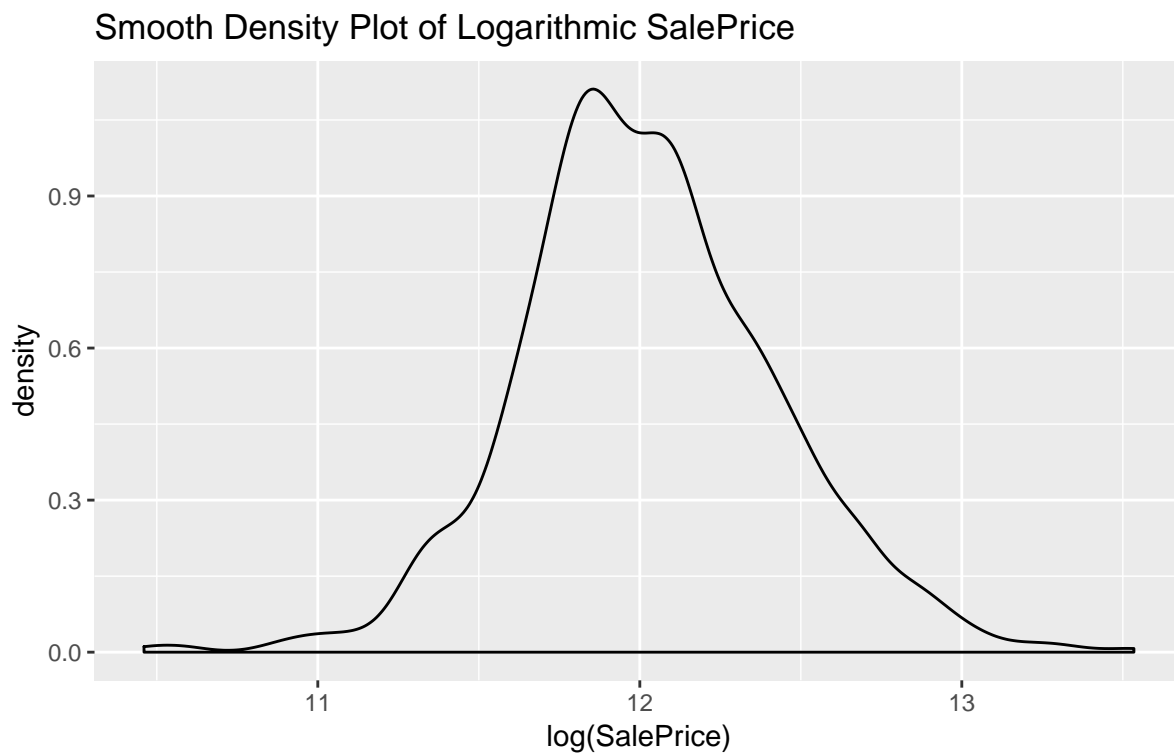
There are a large number of missing values for LotFrontage.

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



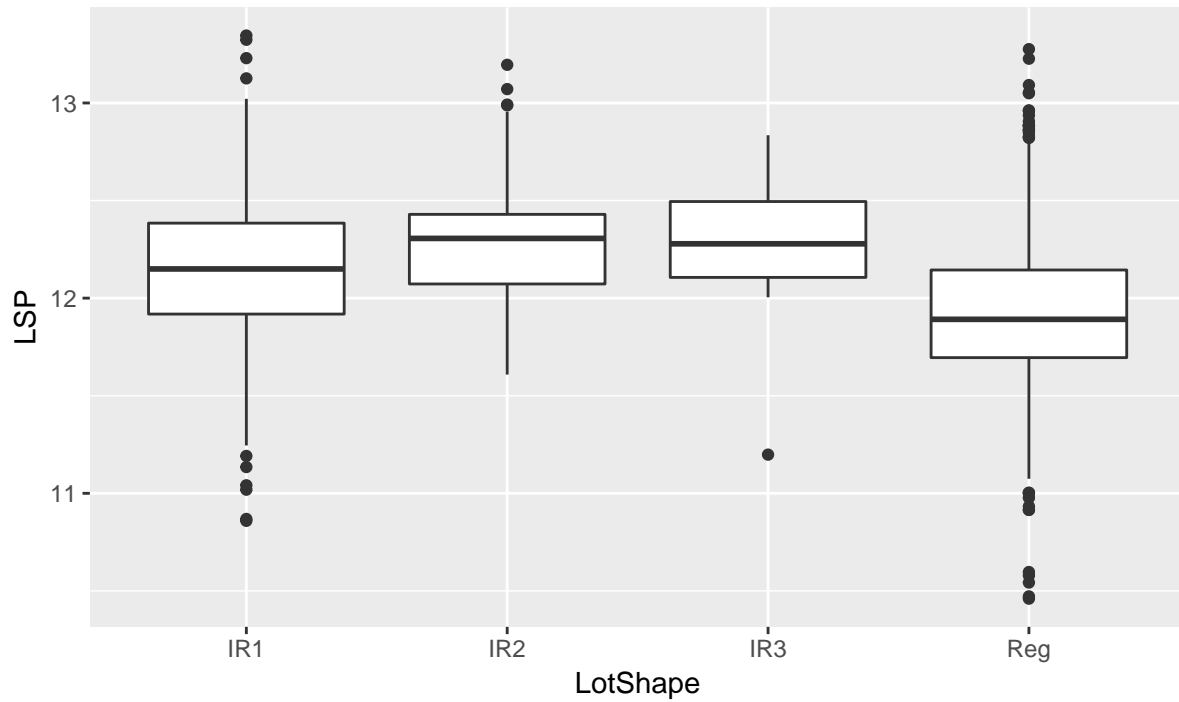
To account for this we take the log of the SalePrice and store it in a variable called LSP



Data Exploration

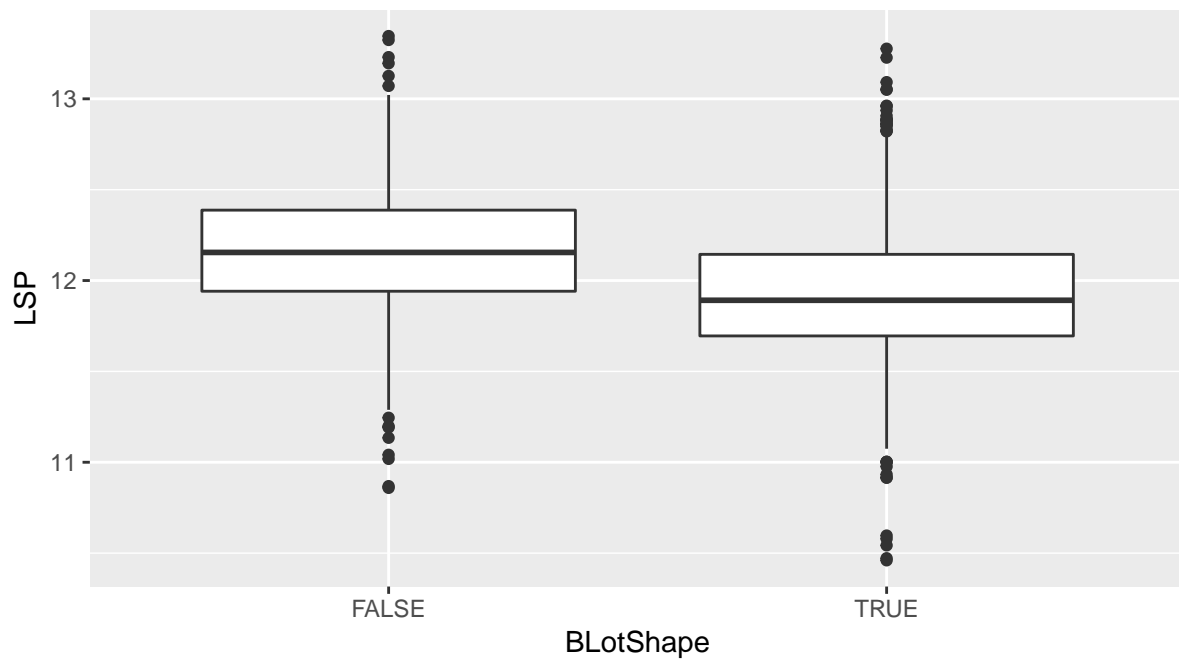
In this section I looked through the columns one by one. I used this to decide which variables could be useful and which were not. This is also the part where I converted the categorical data into separate features.

LotShape vs Log SalePrice

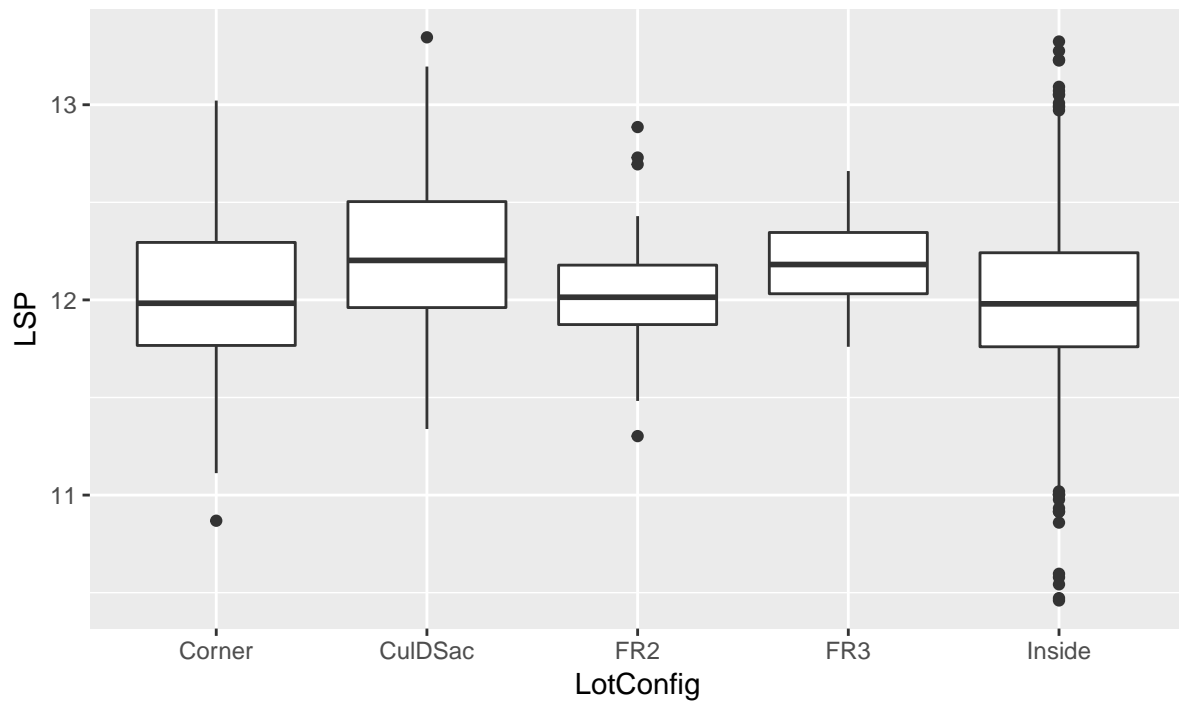


New LotShape Feature

LotShape is Regular? T/F

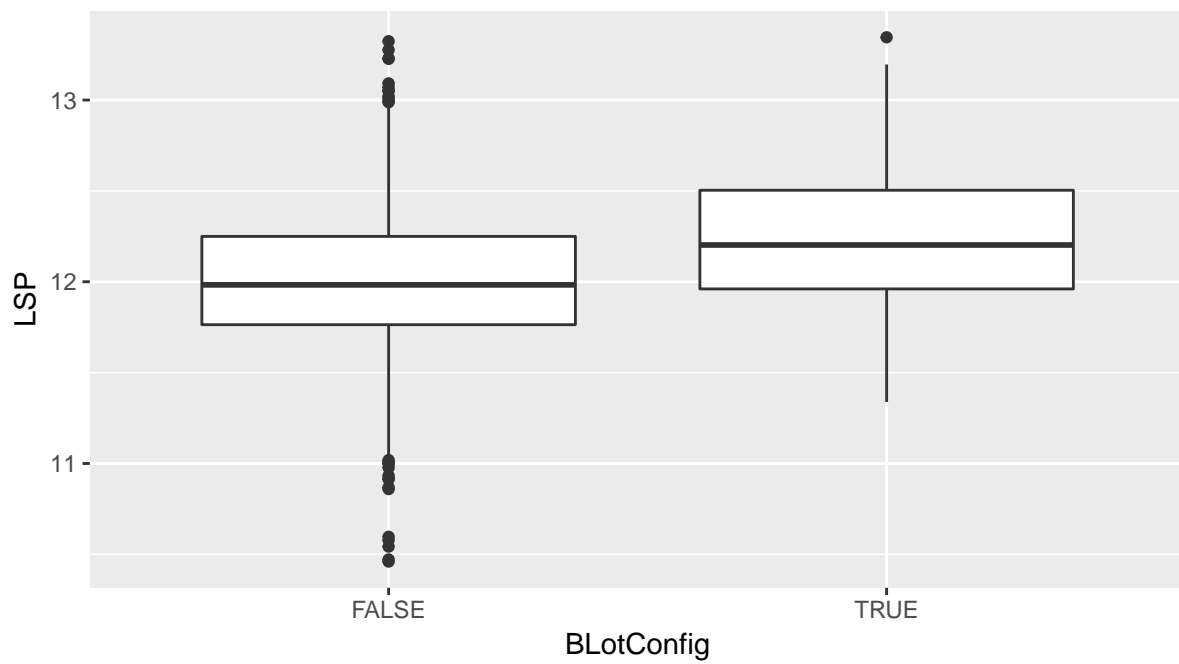


LotConfig vs Log SalePrice

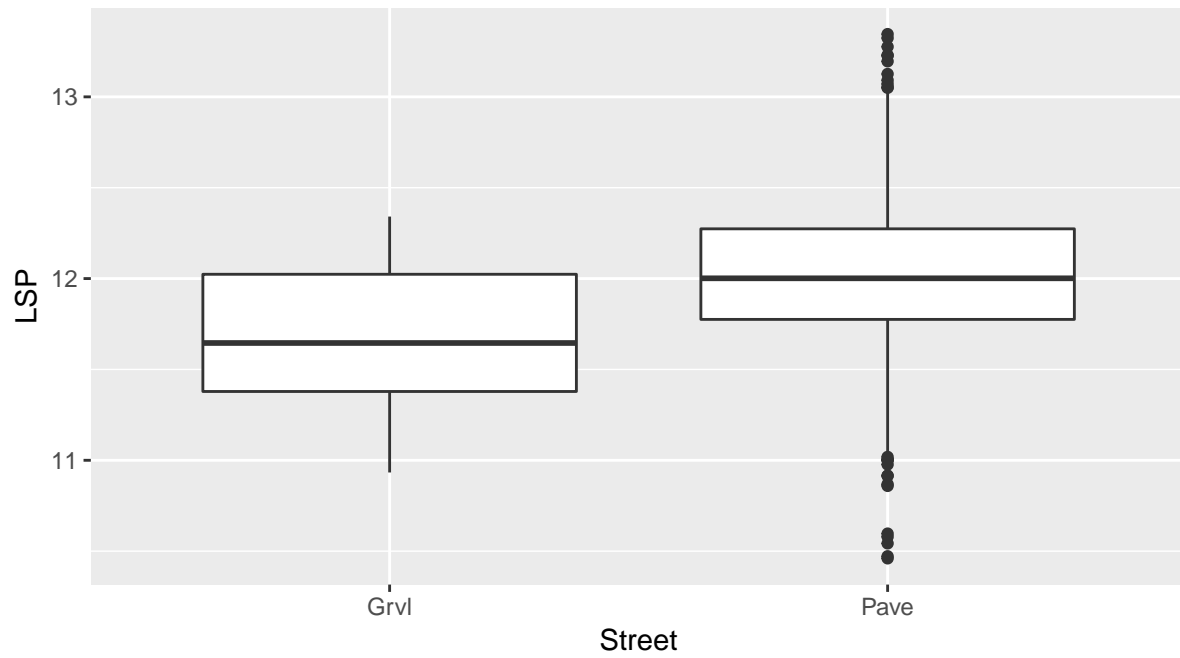


New LotConfig Feature

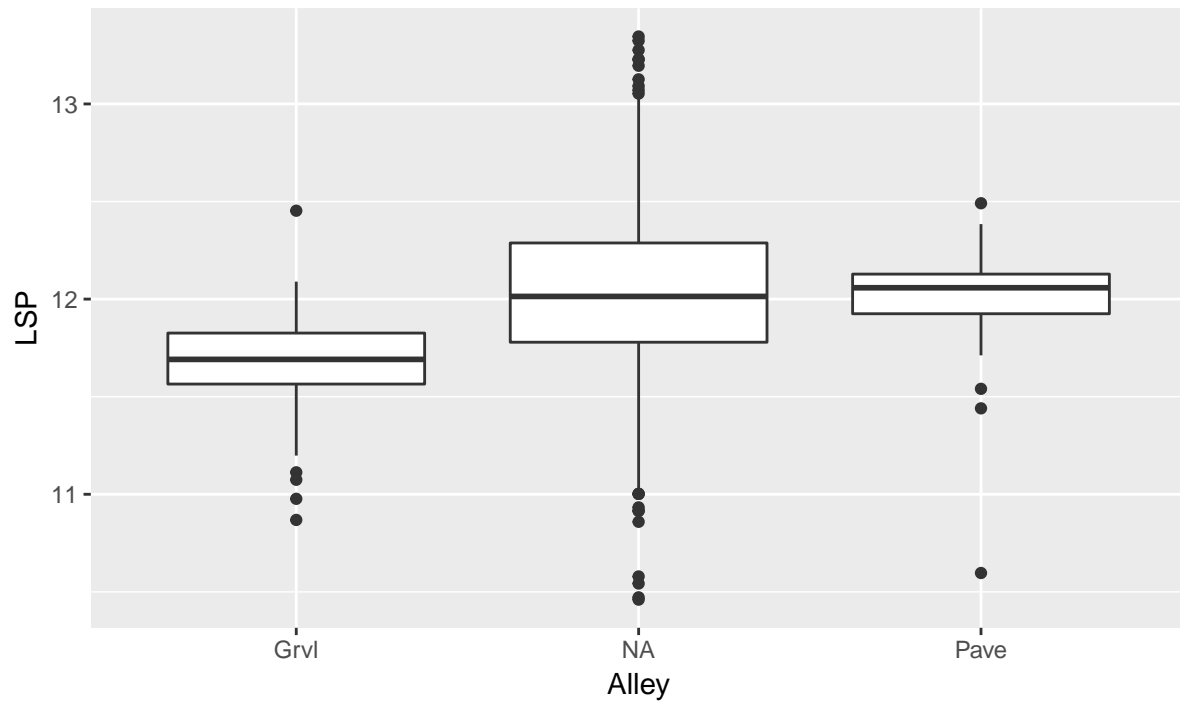
LotConfig is CulDSac+Fr3? T/F



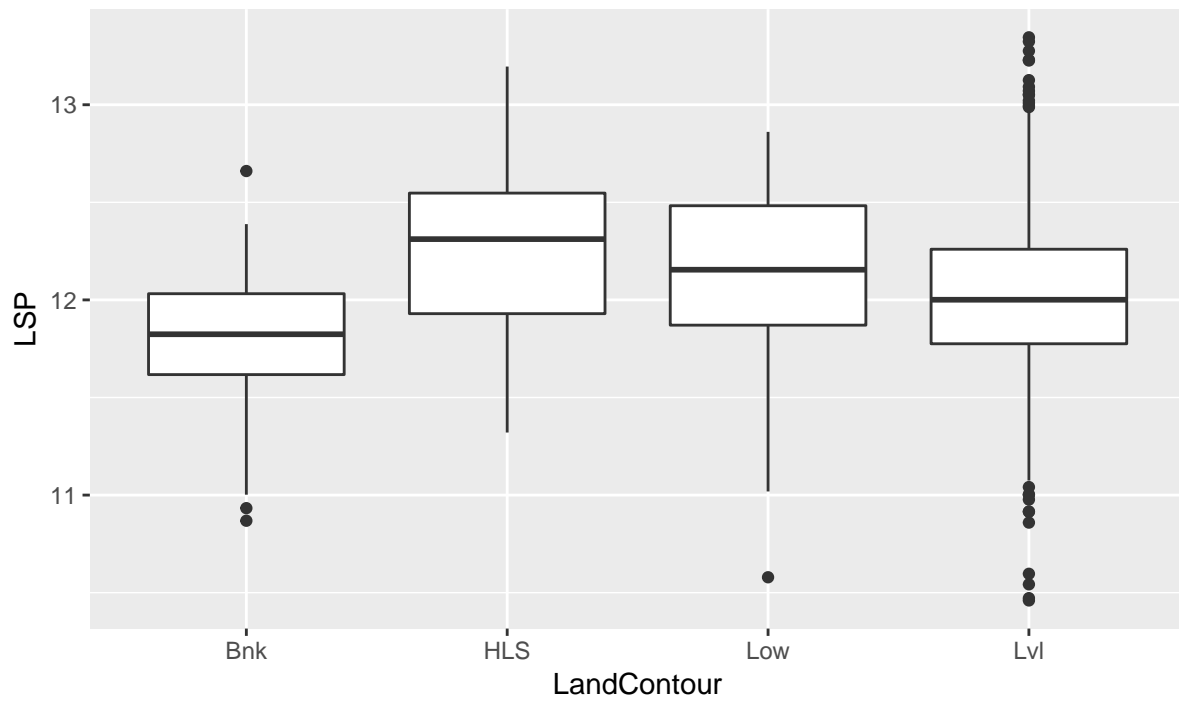
Street vs LSP
Street is boolean



Alley vs Log SalePrice

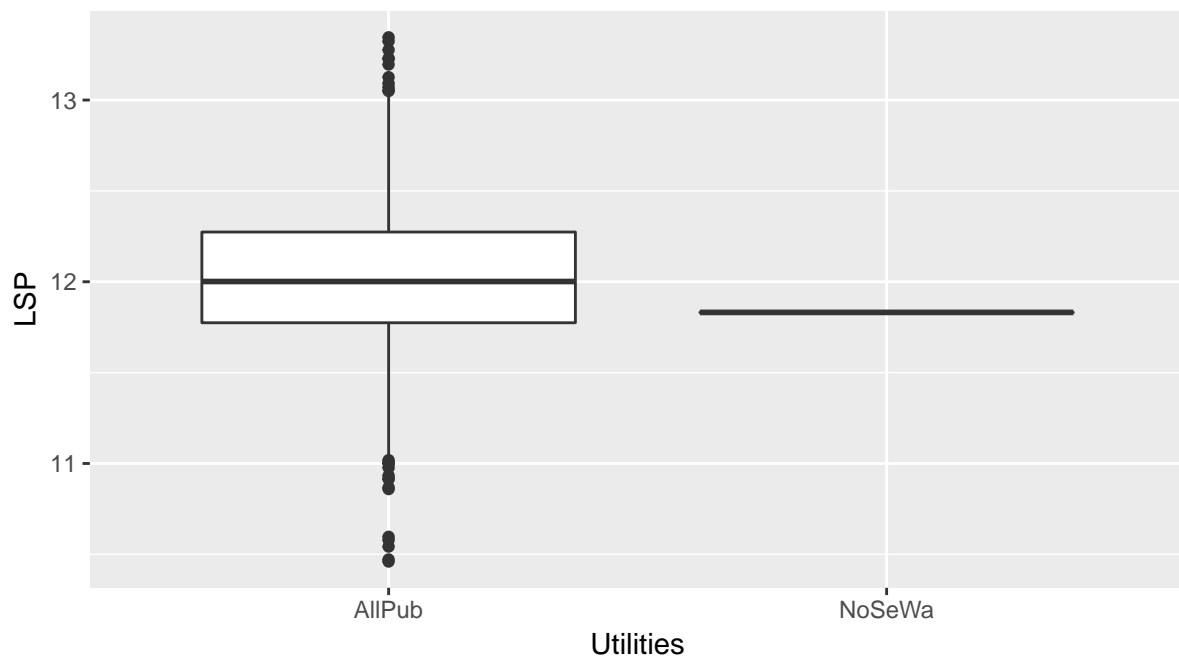


LandContour vs Log SalePrice



Utilities vs LSP

There is only 1 NoSewage entry

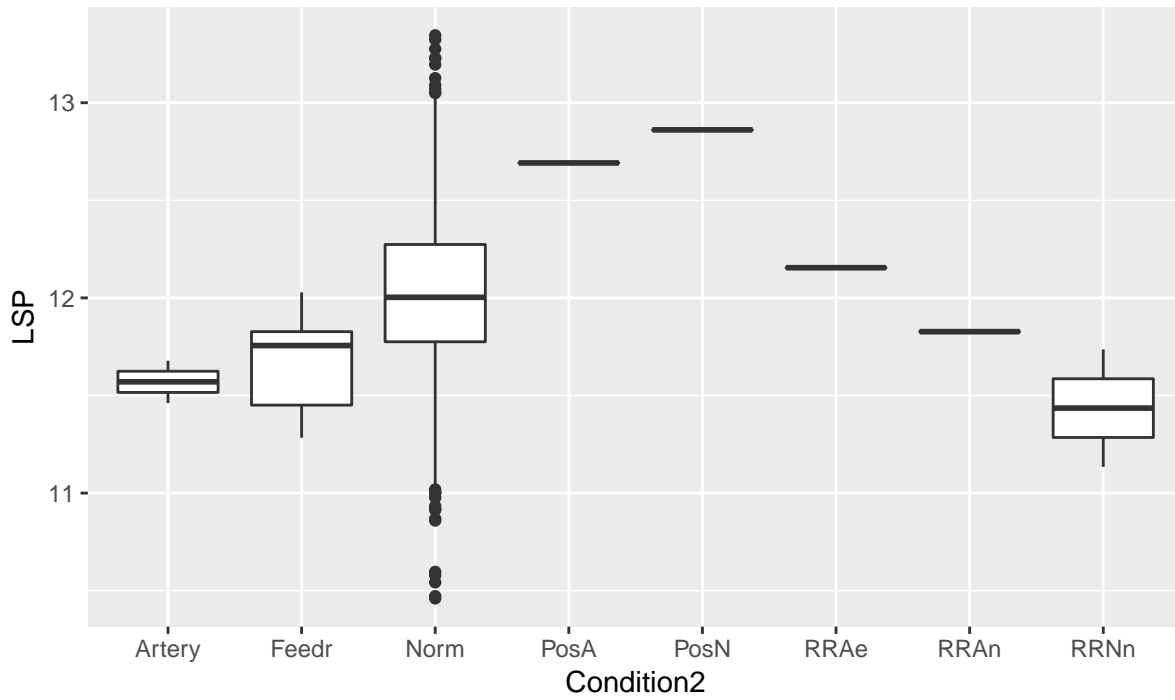


A boxplot comparing the number of children per woman across three countries: Germany, France, and Italy. The y-axis represents the number of children, ranging from 0 to 3. Germany has a median of approximately 1.4, with a box from 1.1 to 1.7 and whiskers from 0.8 to 2.0. France has a median of approximately 1.8, with a box from 1.5 to 2.1 and whiskers from 1.2 to 2.4. Italy has a median of approximately 1.6, with a box from 1.4 to 1.9 and whiskers from 1.1 to 2.1. Outliers are present for Germany (at 0.5, 0.6, 0.7, 2.1, 2.2, 2.3) and France (at 0.4).

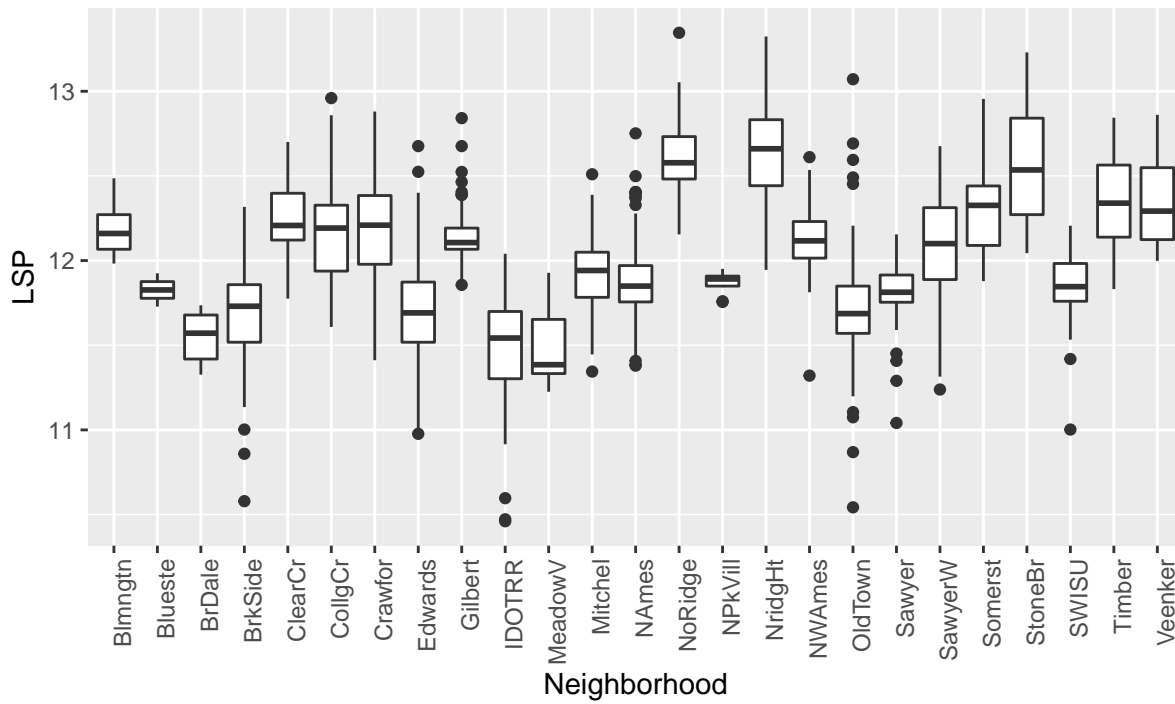
A box plot showing the distribution of LSP (Left Spinal Process) for 10 different categories. The y-axis is labeled 'LSP' and ranges from 11 to 13. The x-axis has 10 categories, each represented by a box plot. The boxes are white with black outlines, and the median is indicated by a thick black horizontal line. Whiskers extend to the minimum and maximum values of the data. Outliers are shown as individual black dots. The categories are ordered by their median LSP value, from lowest to highest.

Category	Min	Q1	Median	Q3	Max	Outliers
1	11.1	11.6	11.7	11.9	12.2	11.3, 11.4, 12.4, 12.7, 13.1
2	11.3	11.7	11.8	12.0	12.4	11.1, 11.2, 11.4, 11.6
3	11.0	11.8	12.0	12.3	13.1	10.9, 11.0, 11.1, 11.2, 11.3, 11.4, 11.5, 11.6, 11.7, 11.8, 11.9, 12.1, 12.2, 12.3, 12.4, 12.5, 12.6, 12.7, 12.8, 12.9, 13.0, 13.1, 13.2, 13.3, 13.4, 13.5, 13.6, 13.7, 13.8, 13.9, 14.0, 14.1, 14.2, 14.3, 14.4, 14.5, 14.6, 14.7, 14.8, 14.9, 15.0, 15.1, 15.2, 15.3, 15.4, 15.5, 15.6, 15.7, 15.8, 15.9, 16.0, 16.1, 16.2, 16.3, 16.4, 16.5, 16.6, 16.7, 16.8, 16.9, 17.0, 17.1, 17.2, 17.3, 17.4, 17.5, 17.6, 17.7, 17.8, 17.9, 18.0, 18.1, 18.2, 18.3, 18.4, 18.5, 18.6, 18.7, 18.8, 18.9, 19.0, 19.1, 19.2, 19.3, 19.4, 19.5, 19.6, 19.7, 19.8, 19.9, 20.0, 20.1, 20.2, 20.3, 20.4, 20.5, 20.6, 20.7, 20.8, 20.9, 21.0, 21.1, 21.2, 21.3, 21.4, 21.5, 21.6, 21.7, 21.8, 21.9, 22.0, 22.1, 22.2, 22.3, 22.4, 22.5, 22.6, 22.7, 22.8, 22.9, 23.0, 23.1, 23.2, 23.3, 23.4, 23.5, 23.6, 23.7, 23.8, 23.9, 24.0, 24.1, 24.2, 24.3, 24.4, 24.5, 24.6, 24.7, 24.8, 24.9, 25.0, 25.1, 25.2, 25.3, 25.4, 25.5, 25.6, 25.7, 25.8, 25.9, 26.0, 26.1, 26.2, 26.3, 26.4, 26.5, 26.6, 26.7, 26.8, 26.9, 27.0, 27.1, 27.2, 27.3, 27.4, 27.5, 27.6, 27.7, 27.8, 27.9, 28.0, 28.1, 28.2, 28.3, 28.4, 28.5, 28.6, 28.7, 28.8, 28.9, 29.0, 29.1, 29.2, 29.3, 29.4, 29.5, 29.6, 29.7, 29.8, 29.9, 30.0, 30.1, 30.2, 30.3, 30.4, 30.5, 30.6, 30.7, 30.8, 30.9, 31.0, 31.1, 31.2, 31.3, 31.4, 31.5, 31.6, 31.7, 31.8, 31.9, 32.0, 32.1, 32.2, 32.3, 32.4, 32.5, 32.6, 32.7, 32.8, 32.9, 33.0, 33.1, 33.2, 33.3, 33.4, 33.5, 33.6, 33.7, 33.8, 33.9, 34.0, 34.1, 34.2, 34.3, 34.4, 34.5, 34.6, 34.7, 34.8, 34.9, 35.0, 35.1, 35.2, 35.3, 35.4, 35.5, 35.6, 35.7, 35.8, 35.9, 36.0, 36.1, 36.2, 36.3, 36.4, 36.5, 36.6, 36.7, 36.8, 36.9, 37.0, 37.1, 37.2, 37.3, 37.4, 37.5, 37.6, 37.7, 37.8, 37.9, 38.0, 38.1, 38.2, 38.3, 38.4, 38.5, 38.6, 38.7, 38.8, 38.9, 39.0, 39.1, 39.2, 39.3, 39.4, 39.5, 39.6, 39.7, 39.8, 39.9, 40.0, 40.1, 40.2, 40.3, 40.4, 40.5, 40.6, 40.7, 40.8, 40.9, 41.0, 41.1, 41.2, 41.3, 41.4, 41.5, 41.6, 41.7, 41.8, 41.9, 42.0, 42.1, 42.2, 42.3, 42.4, 42.5, 42.6, 42.7, 42.8, 42.9, 43.0, 43.1, 43.2, 43.3, 43.4, 43.5, 43.6, 43.7, 43.8, 43.9, 44.0, 44.1, 44.2, 44.3, 44.4, 44.5, 44.6, 44.7, 44.8, 44.9, 45.0, 45.1, 45.2, 45.3, 45.4, 45.5, 45.6, 45.7, 45.8, 45.9, 46.0, 46.1, 46.2, 46.3, 46.4, 46.5, 46.6, 46.7, 46.8, 46.9, 47.0, 47.1, 47.2, 47.3, 47.4, 47.5, 47.6, 47.7, 47.8, 47.9, 48.0, 48.1, 48.2, 48.3, 48.4, 48.5, 48.6, 48.7, 48.8, 48.9, 49.0, 49.1, 49.2, 49.3, 49.4, 49.5, 49.6, 49.7, 49.8, 49.9, 50.0, 50.1, 50.2, 50.3, 50.4, 50.5, 50.6, 50.7, 50.8, 50.9, 51.0, 51.1, 51.2, 51.3, 51.4, 51.5, 51.6, 51.7, 51.8, 51.9, 52.0, 52.1, 52.2, 52.3, 52.4, 52.5, 52.6, 52.7, 52.8, 52.9, 53.0, 53.1, 53.2, 53.3, 53.4, 53.5, 53.6, 53.7, 53.8, 53.9, 54.0, 54.1, 54.2, 54.3, 54.4, 54.5, 54.6, 54.7, 54.8, 54.9, 55.0, 55.1, 55.2, 55.3, 55.4, 55.5, 55.6, 55.7, 55.8, 55.9, 56.0, 56.1, 56.2, 56.3, 56.4, 56.5, 56.6, 56.7, 56.8, 56.9, 57.0, 57.1, 57.2, 57.3, 57.4, 57.5, 57.6, 57.7, 57.8, 57.9, 58.0, 58.1, 58.2, 58.3, 58.4, 58.5, 58.6, 58.7, 58.8, 58.9, 59.0, 59.1, 59.2, 59.3, 59.4, 59.5, 59.6, 59.7, 59.8, 59.9, 60.0, 60.1, 60.2, 60.3, 60.4, 60.5, 60.6, 60.7, 60.8, 60.9, 61.0, 61.1, 61.2, 61.3, 61.4, 61.5, 61.6, 61.7, 61.8, 61.9, 62.0, 62.1, 62.2, 62.3, 62.4, 62.5, 62.6, 62.7, 62.8, 62.9, 63.0, 63.1, 63.2, 63.3, 63.4, 63.5, 63.6, 63.7, 63.8, 63.9, 64.0, 64.1, 64.2, 64.3, 64.4, 64.5, 64.6, 64.7, 64.8, 64.9, 65.0, 65.1, 65.2, 65.3, 65.4, 65.5, 65.6, 65.7, 65.8, 65.9, 66.0, 66.1, 66.2, 66.3, 66.4, 66.5, 66.6, 66.7, 66.8, 66.9, 67.0, 67.1, 67.2, 67.3, 67.4, 67.5, 67.6, 67.7, 67.8, 67.9, 68.0, 68.1, 68.2, 68.3, 68.4, 68.5, 68.6, 68.7, 68.8, 68.9, 69.0, 69.1, 69.2, 69.3, 69.4, 69.5, 69.6, 69.7, 69.8, 69.9, 70.0, 70.1, 70.2, 70.3, 70.4, 70.5, 70.6, 70.7, 70.8, 70.9, 71.0, 71.1, 71.2, 71.3, 71.4, 7

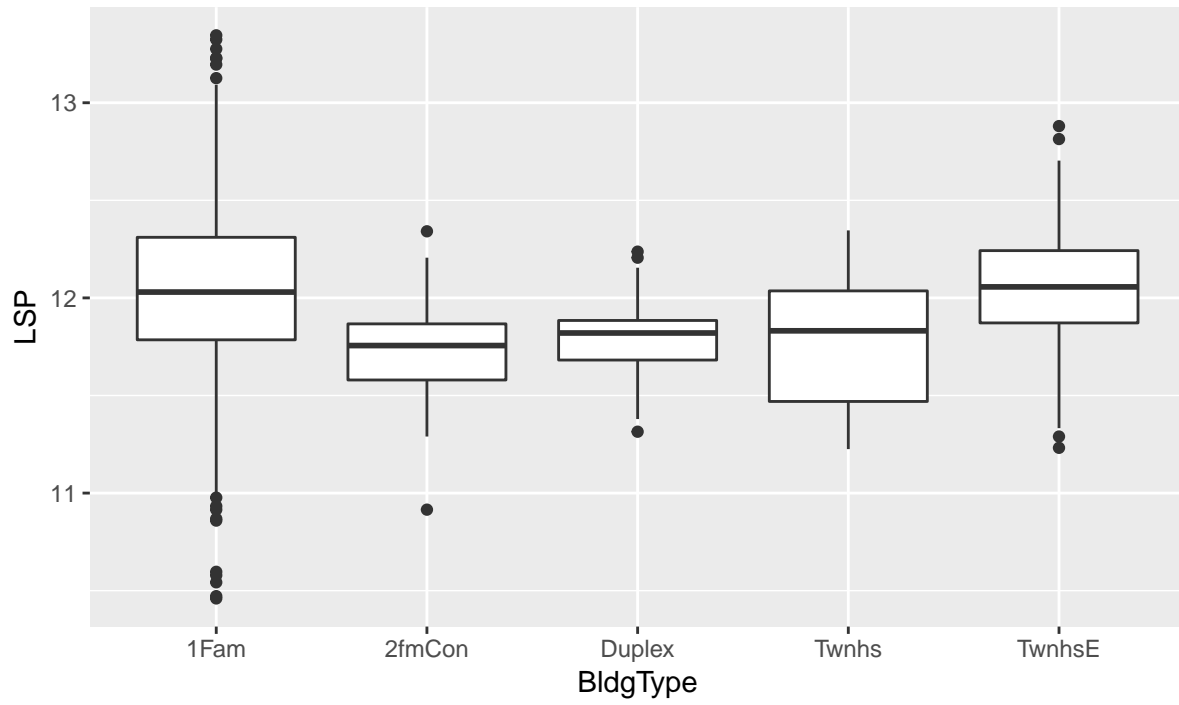
Condition2 vs Log SalePrice



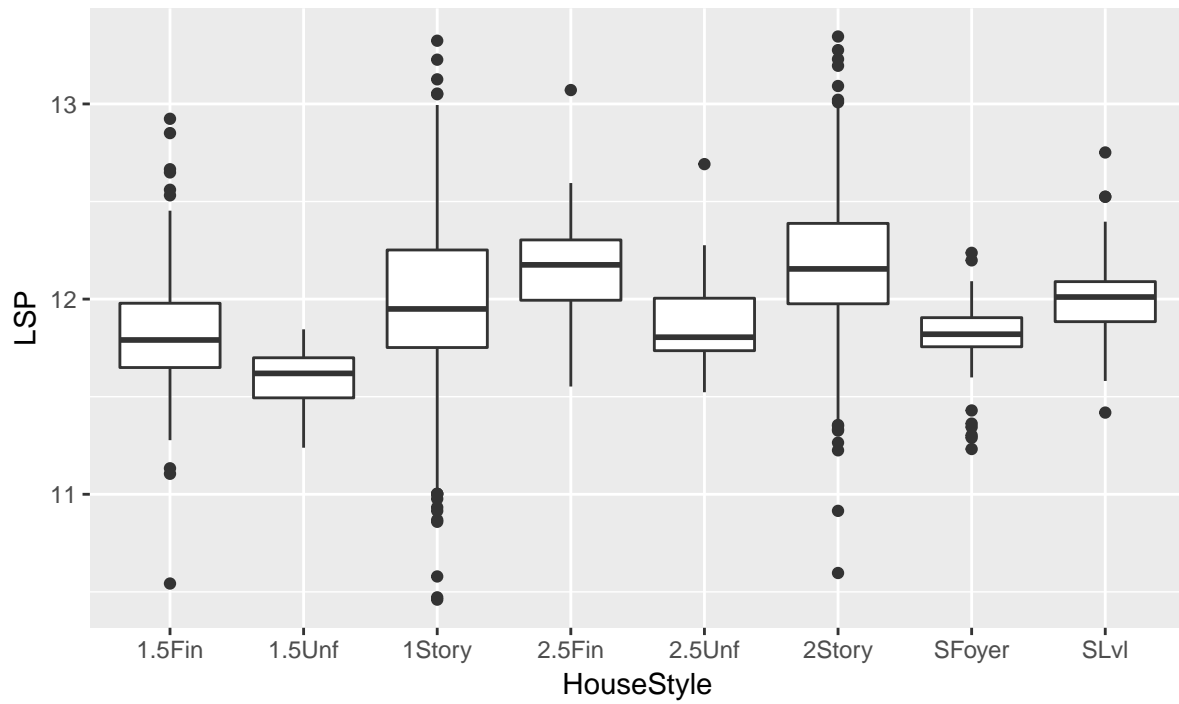
Neighborhood vs Log SalePrice



BldgType vs Log SalePrice

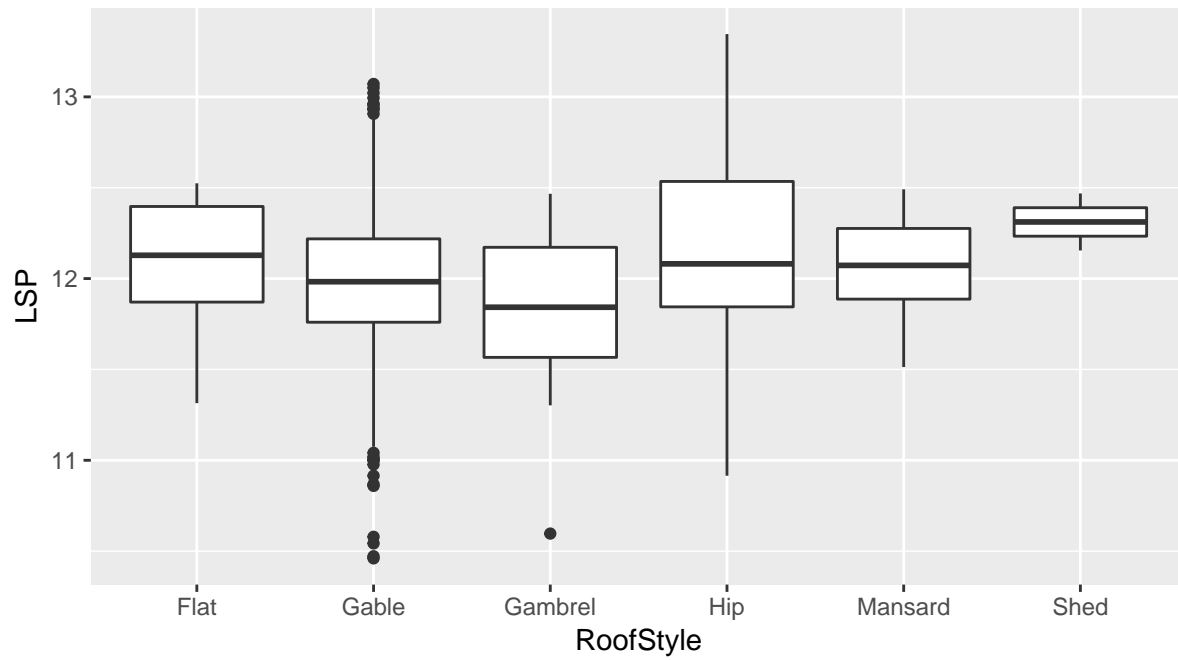


HouseStyle vs Log SalePrice

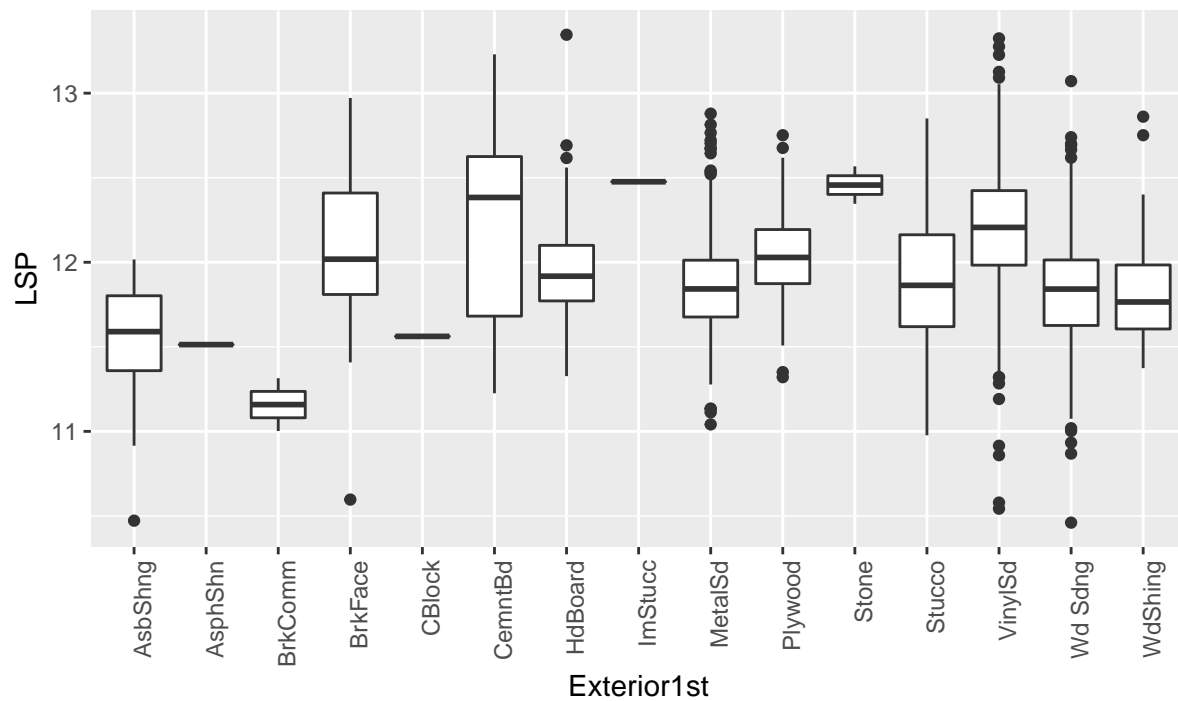


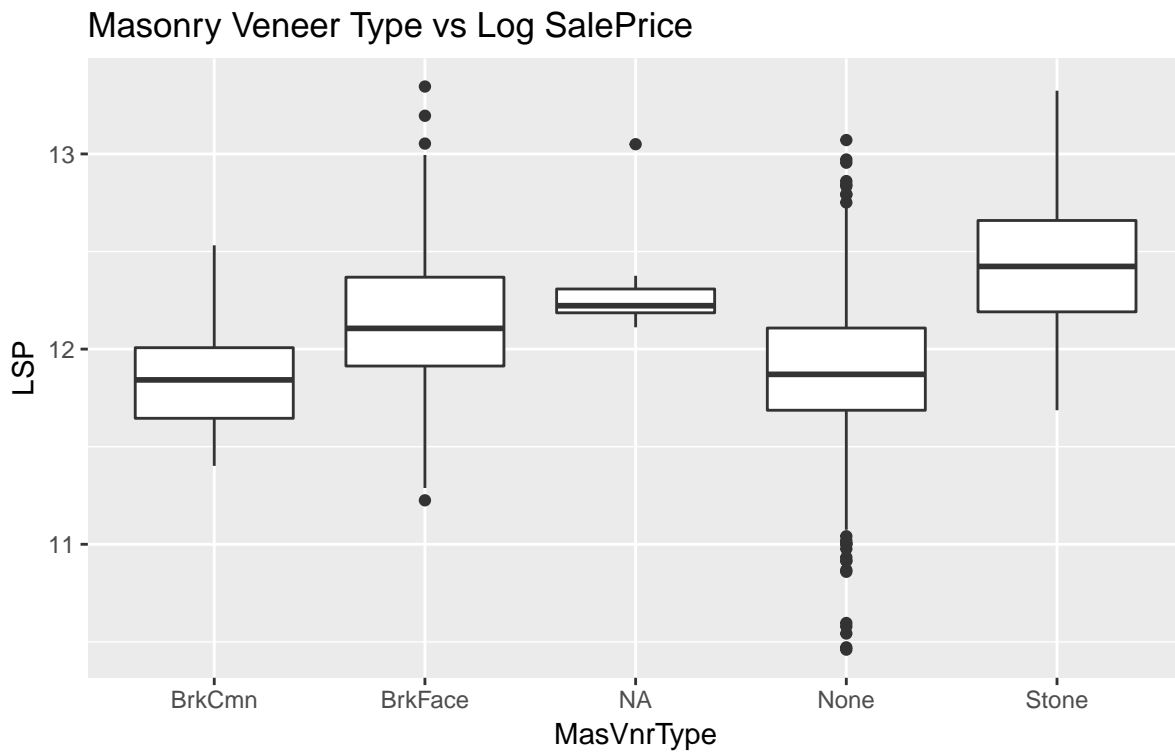
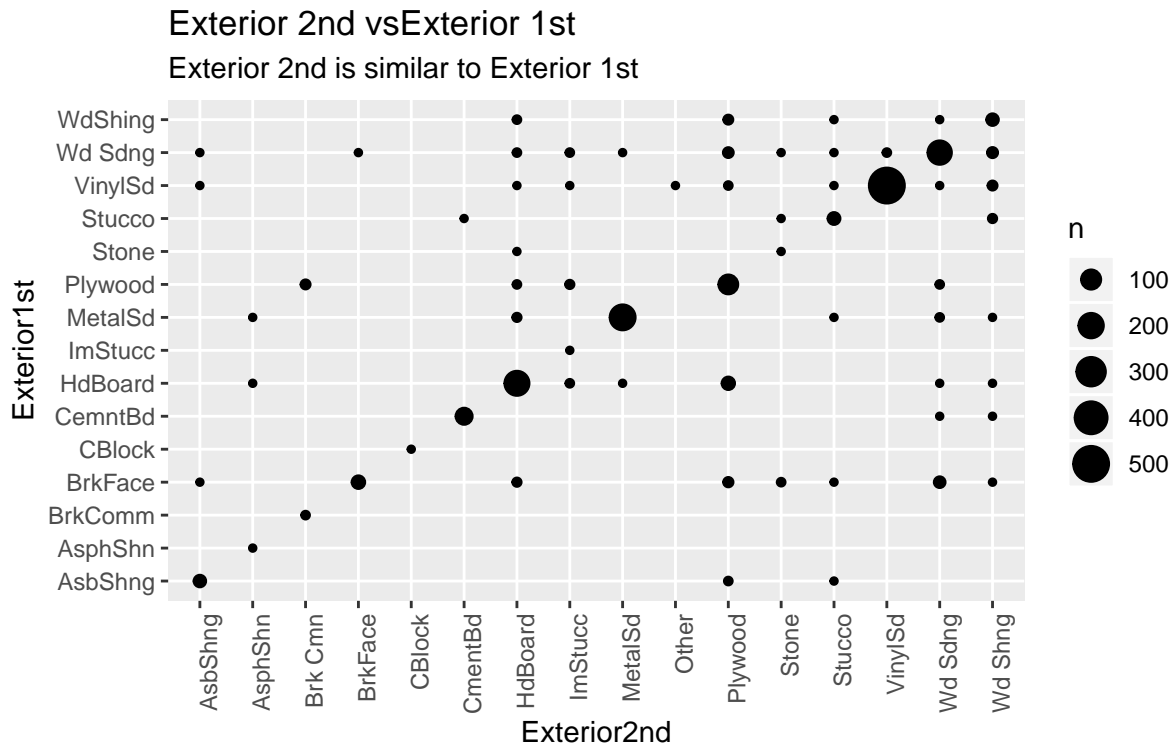
RoofStyle vs LSP

RoofStyle Effect is weak, is dropped



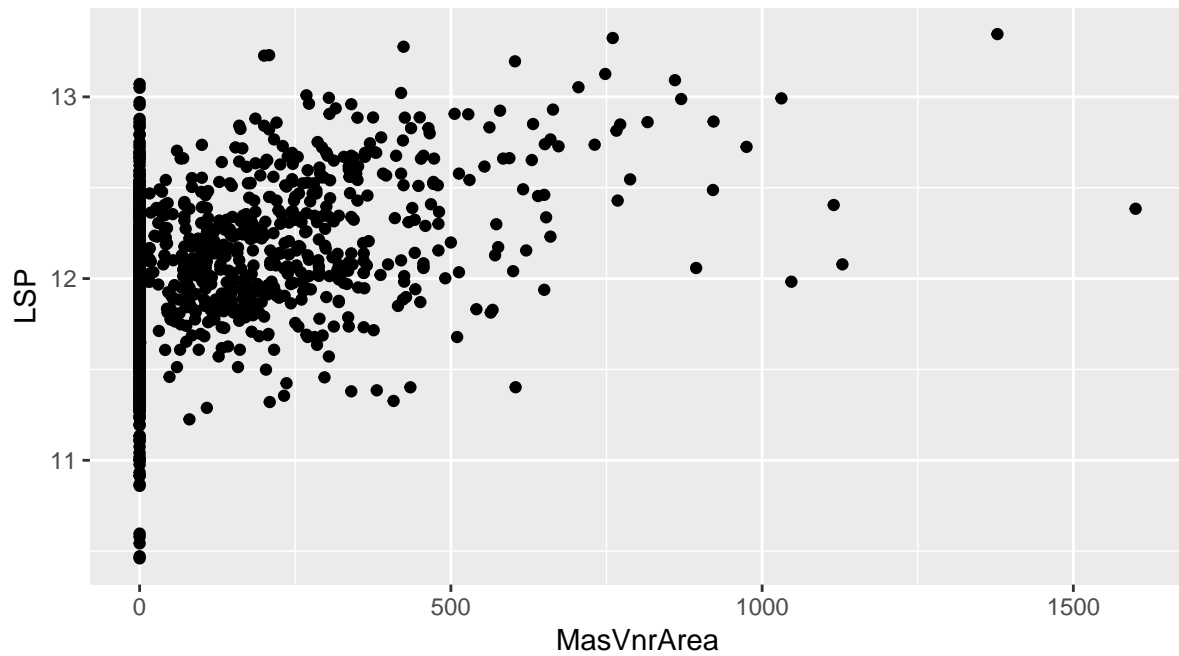
Exterior 1st vs Log SalePrice



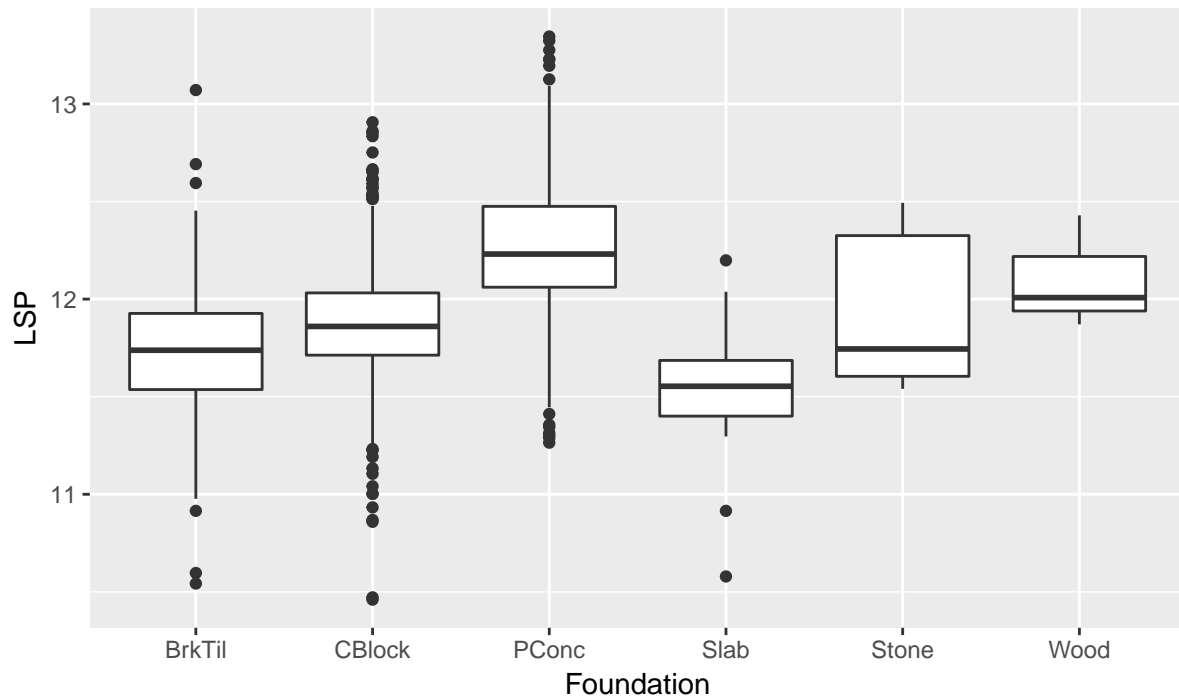


Masonry Veneer Area vs LSP

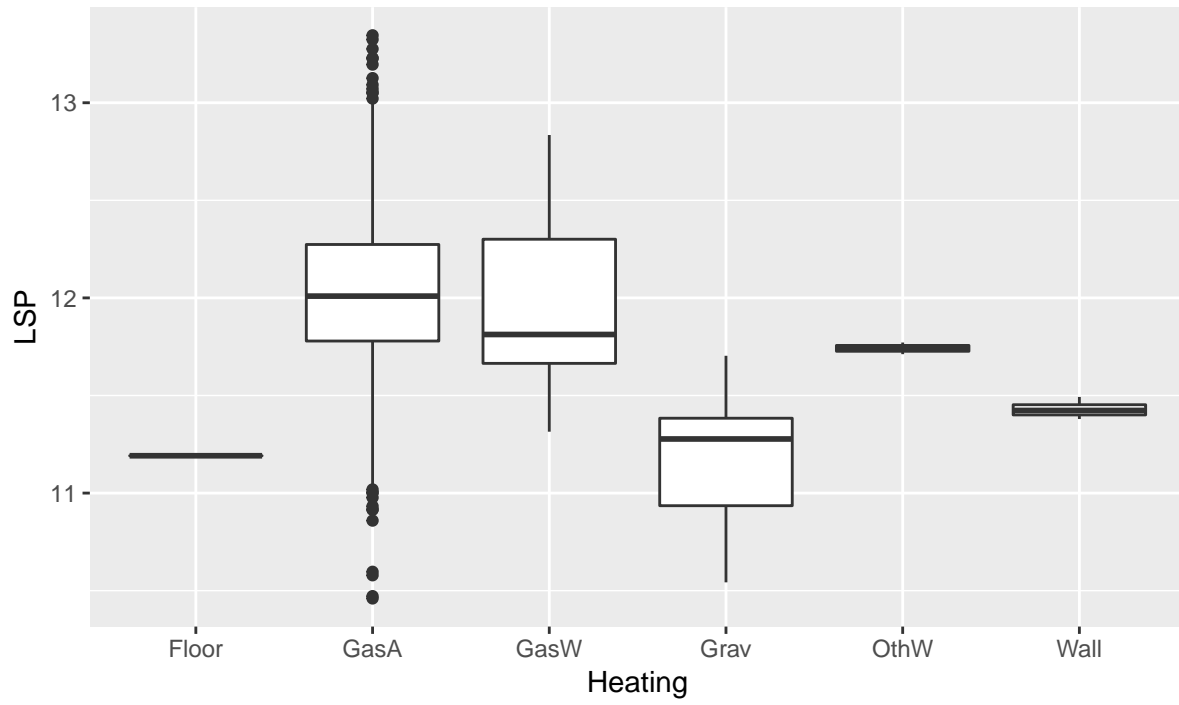
MasVnrArea is not strongly correlated to LSP, is dropped



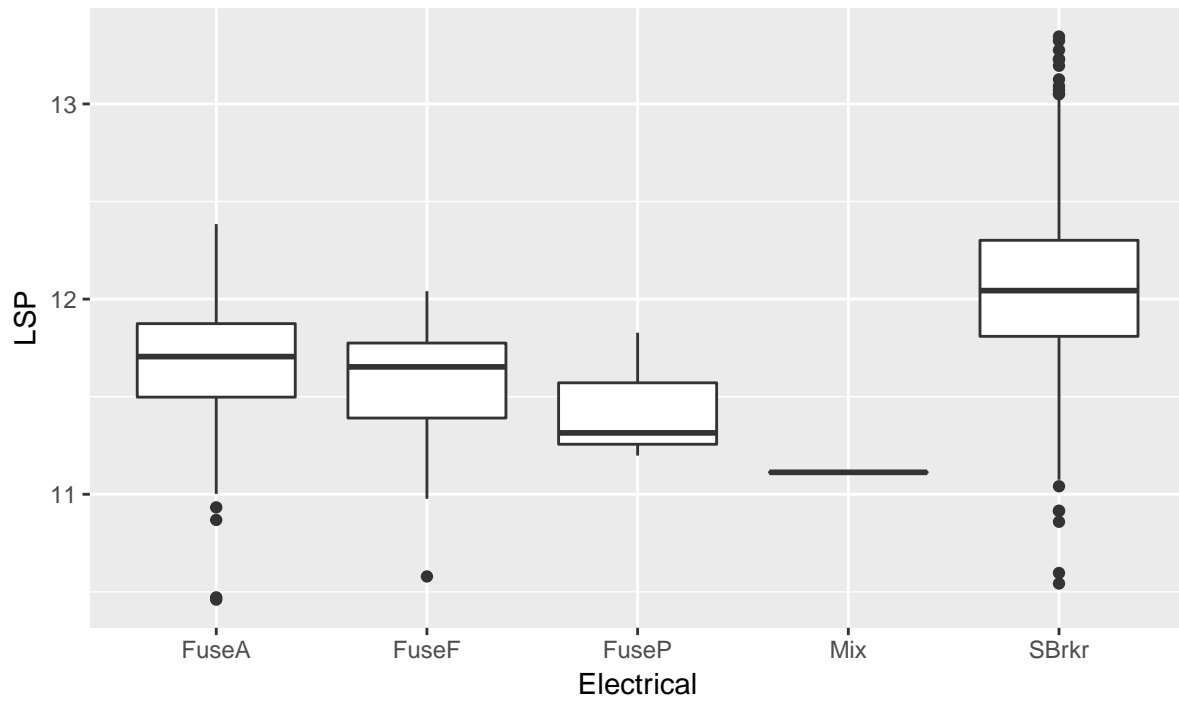
Foundation vs Log SalePrice

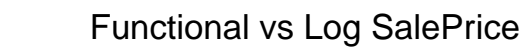


Heating vs Log SalePrice

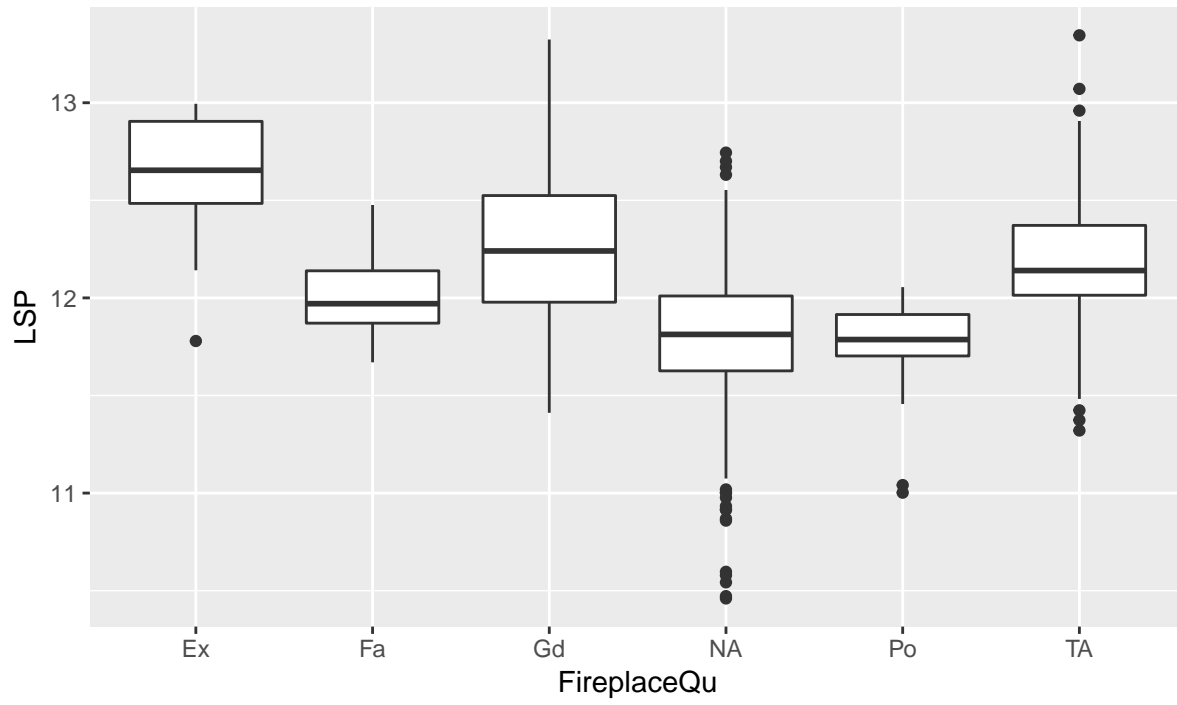


Electrical vs Log SalePrice

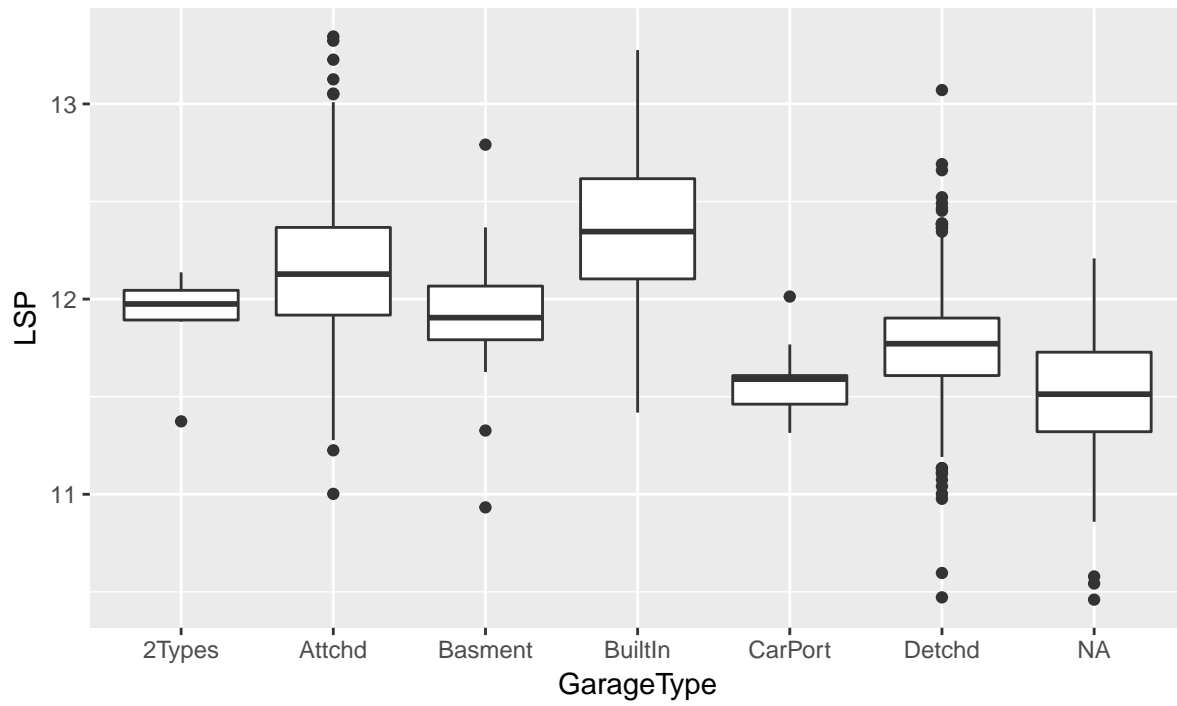




FirePlace Quality vs Log SalePrice



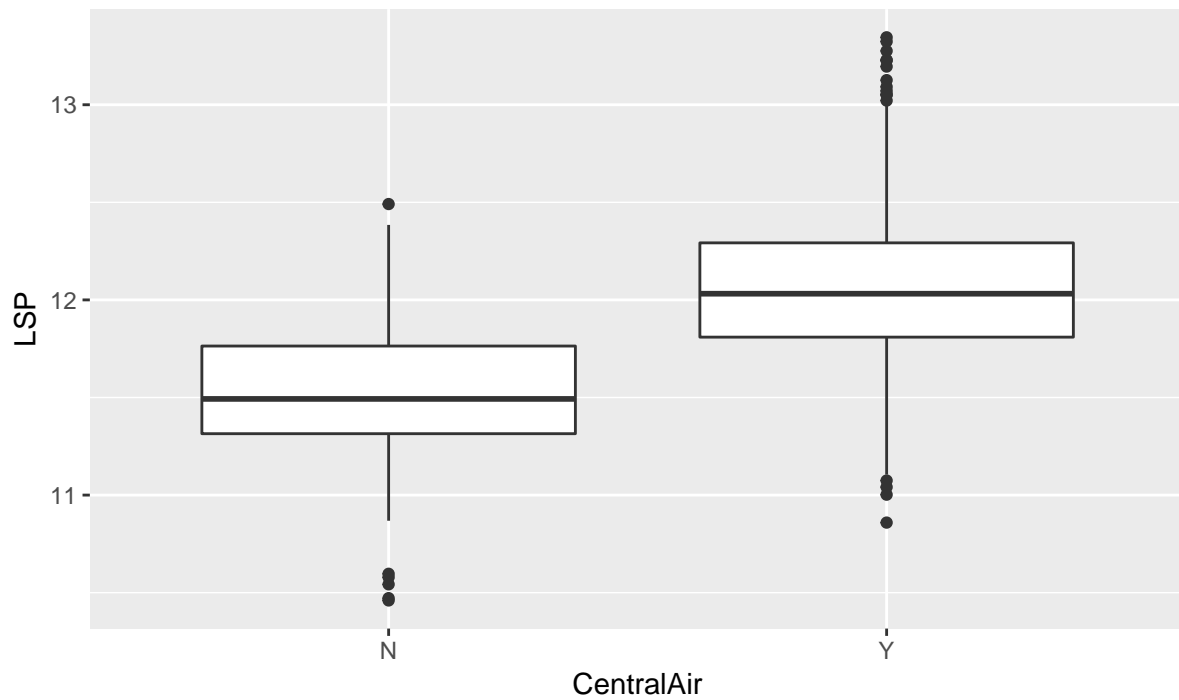
GarageType vs Log SalePrice



A box plot showing the distribution of SaleType across different categories. The x-axis is labeled 'SaleType' and the y-axis represents the value of the variable. The categories on the x-axis are COD, Con, ConLD, ConLI, ConLw, CWD, New, Oth, and WD. The plot shows that the 'WD' category has the highest median and the largest spread, while 'Oth' has the lowest median. The 'Con' category has a relatively narrow distribution with a high median.

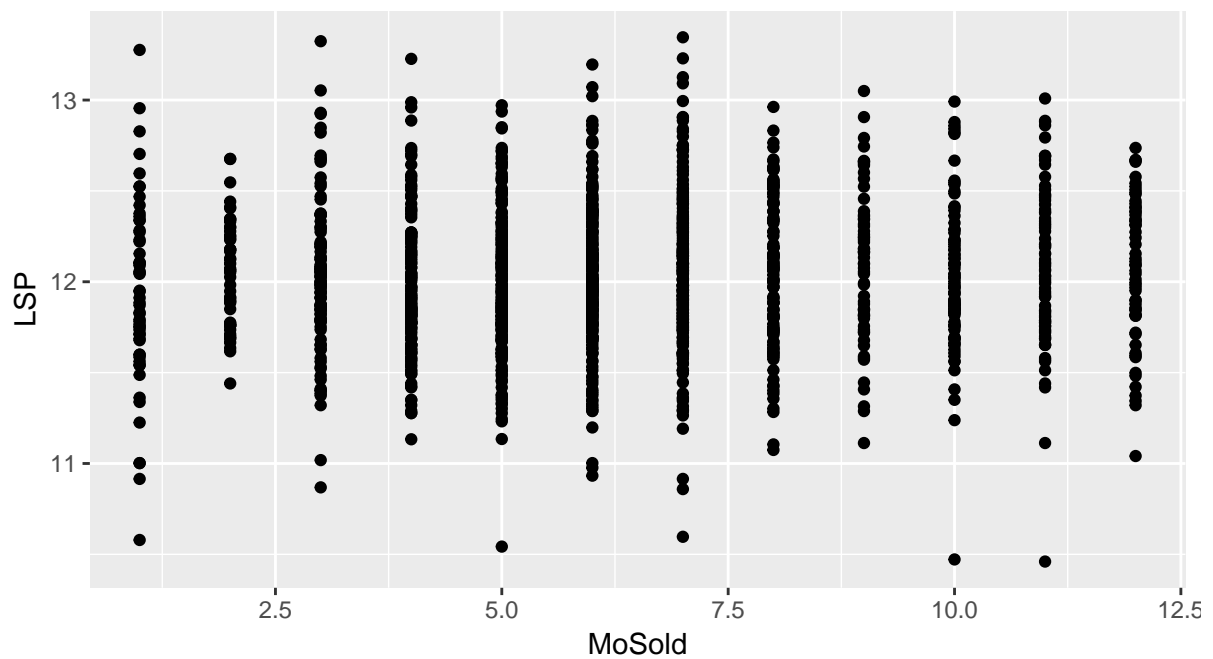
Boxplot showing the distribution of LSP (Y-axis) across different SaleCondition categories (X-axis). The categories are Abnorml, AdjLand, Alloca, Family, Normal, and Partial. The Y-axis ranges from 11 to 13. The boxplots indicate the median, quartiles, and range of LSP values for each condition. The 'Normal' condition shows the highest median LSP, while 'Abnorml' shows the lowest median LSP. Outliers are present for 'Abnorml' and 'Normal'.

Central Air Conditioning vs Log SalePrice



Month Sold vs LSP

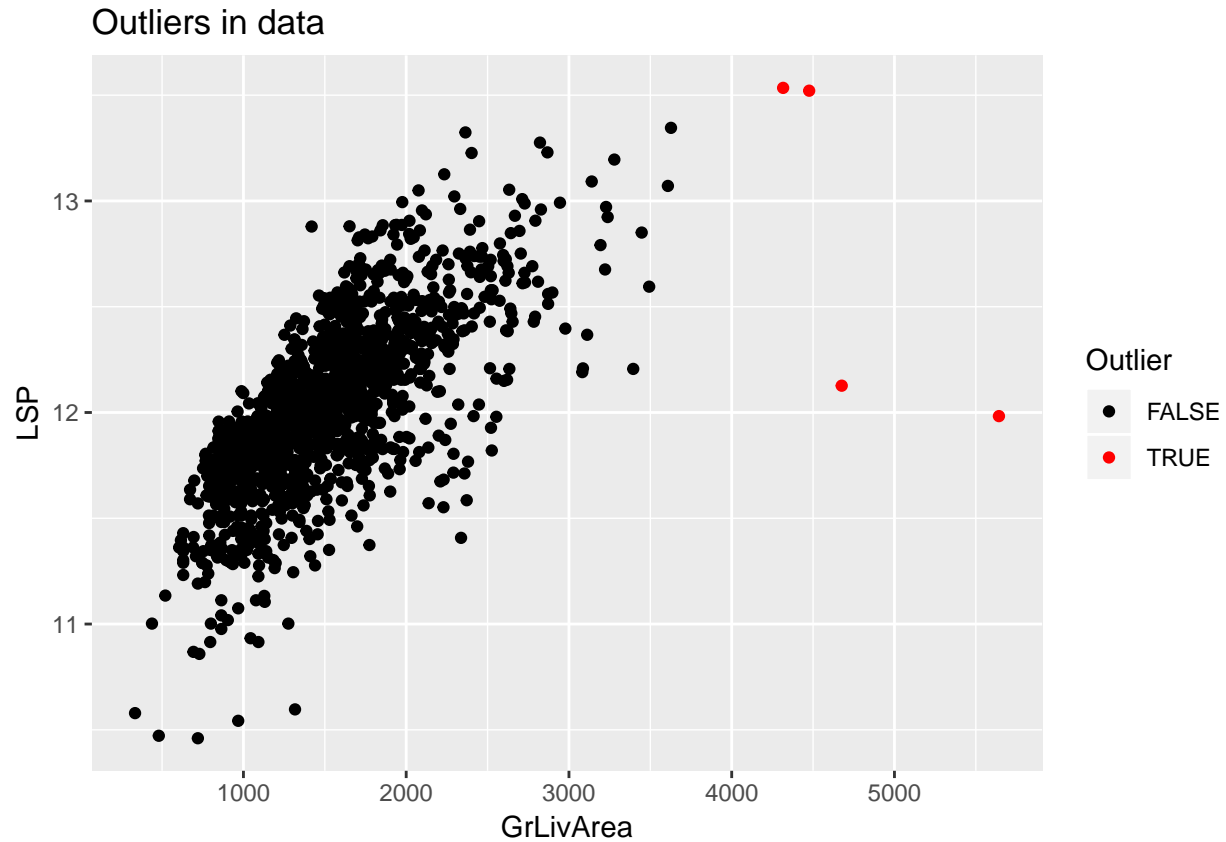
Month Effect is not strong, MoSold is dropped



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 improvement 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.