

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.

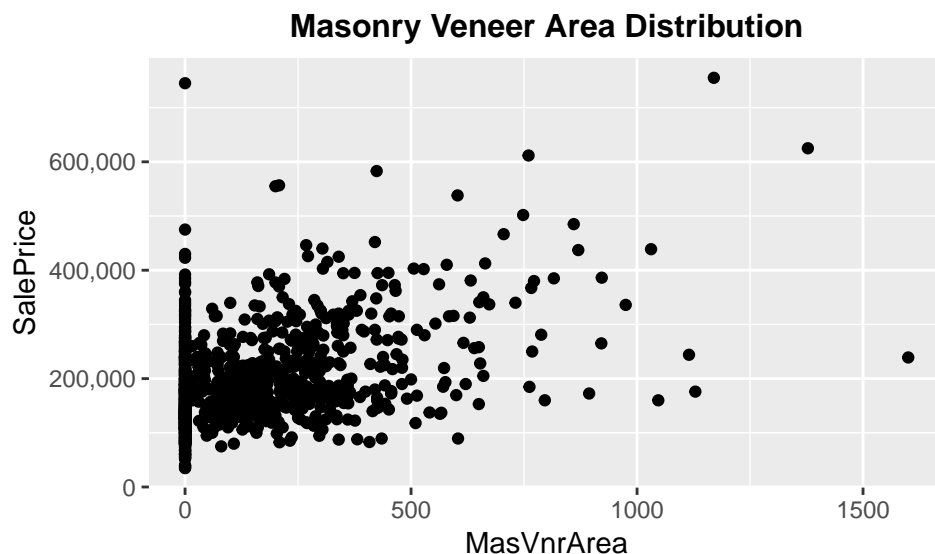
Numeric Data

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
```

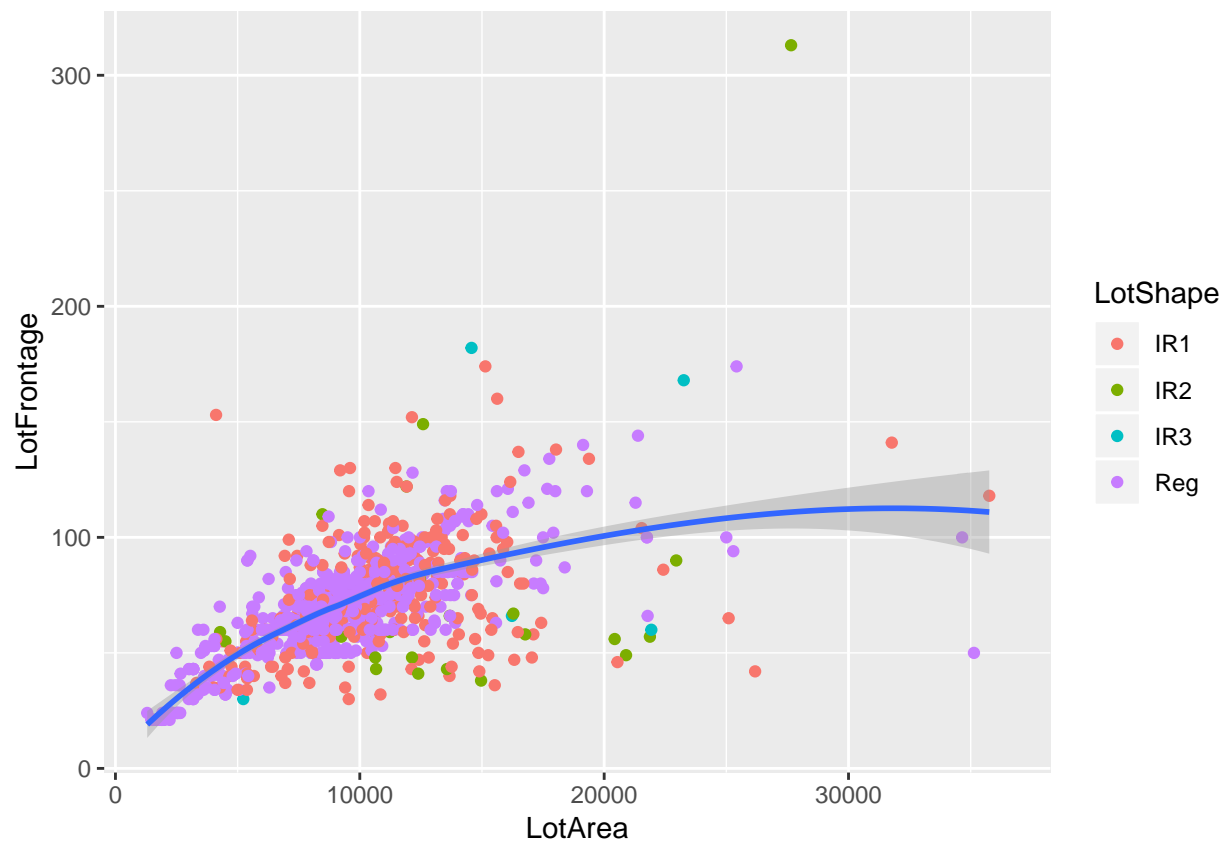
```
train$MSZoning[is.na(train$MSZoning)]<-"RL"
train%>%group_by(Electrical)%>%summarize(count=n())
```

```
## # 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
```

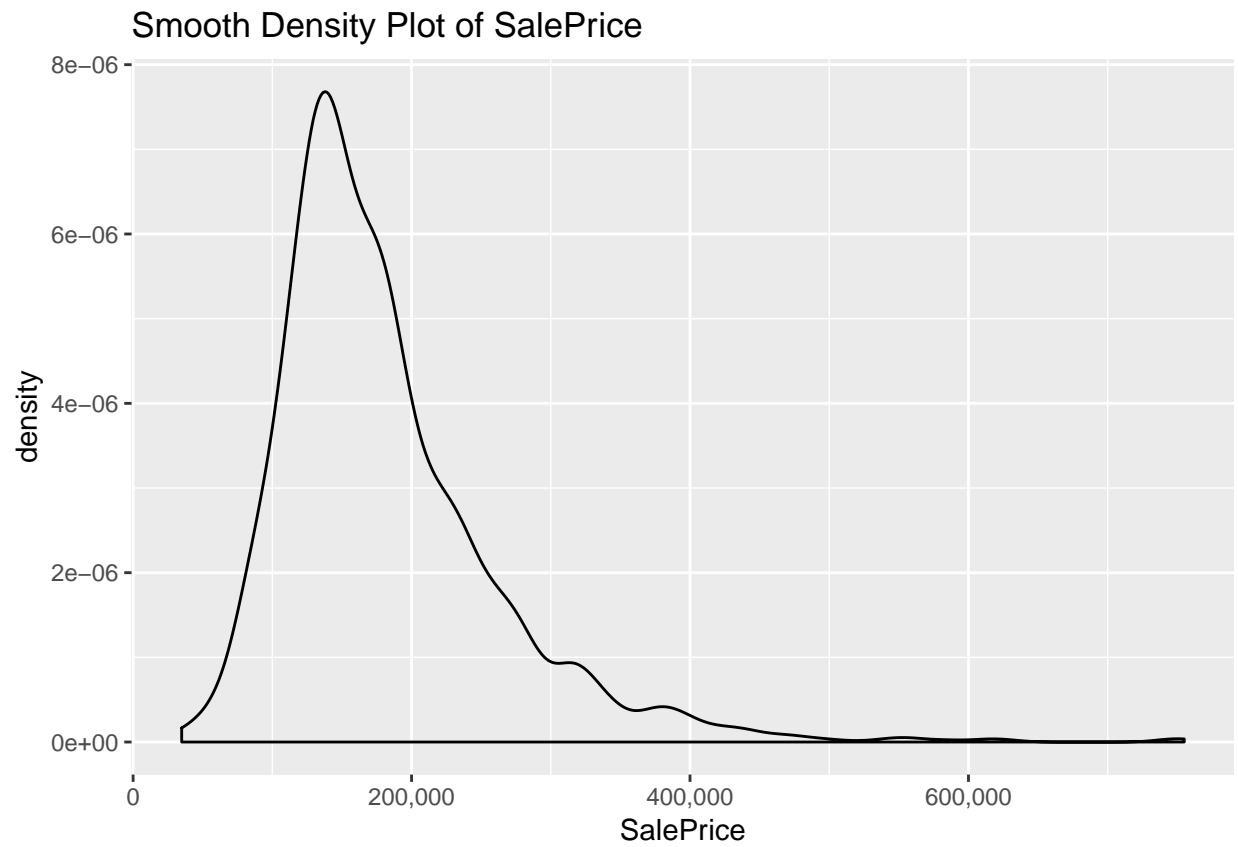


LotFrontage: Linear feet of street connected to property

There are a large number of missing values for LotFrontage.

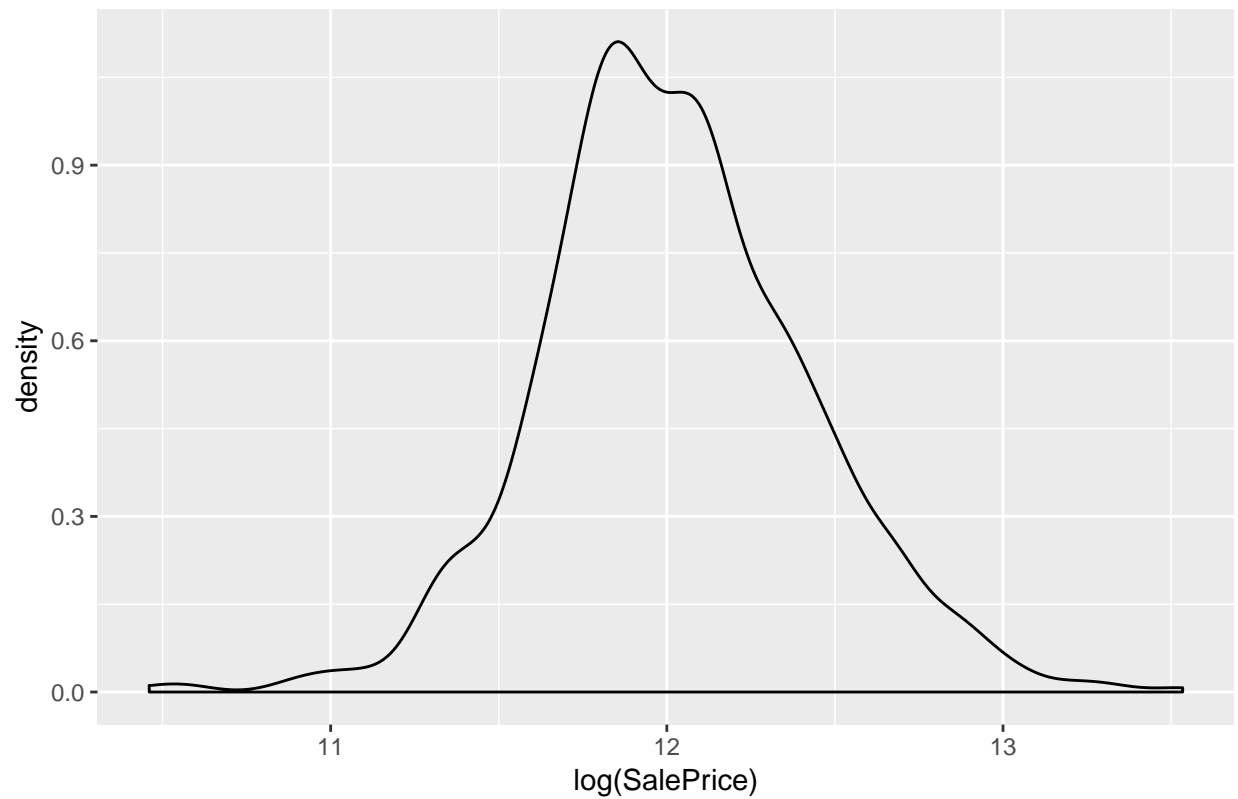
I therefore fit a curve to estimate LotFrontage from LotArea. These variables are highly correlated. The different LotShapes do not have an obviously different ratio and LotShape was not used to improve the estimate.

Skewness of Target Variable



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

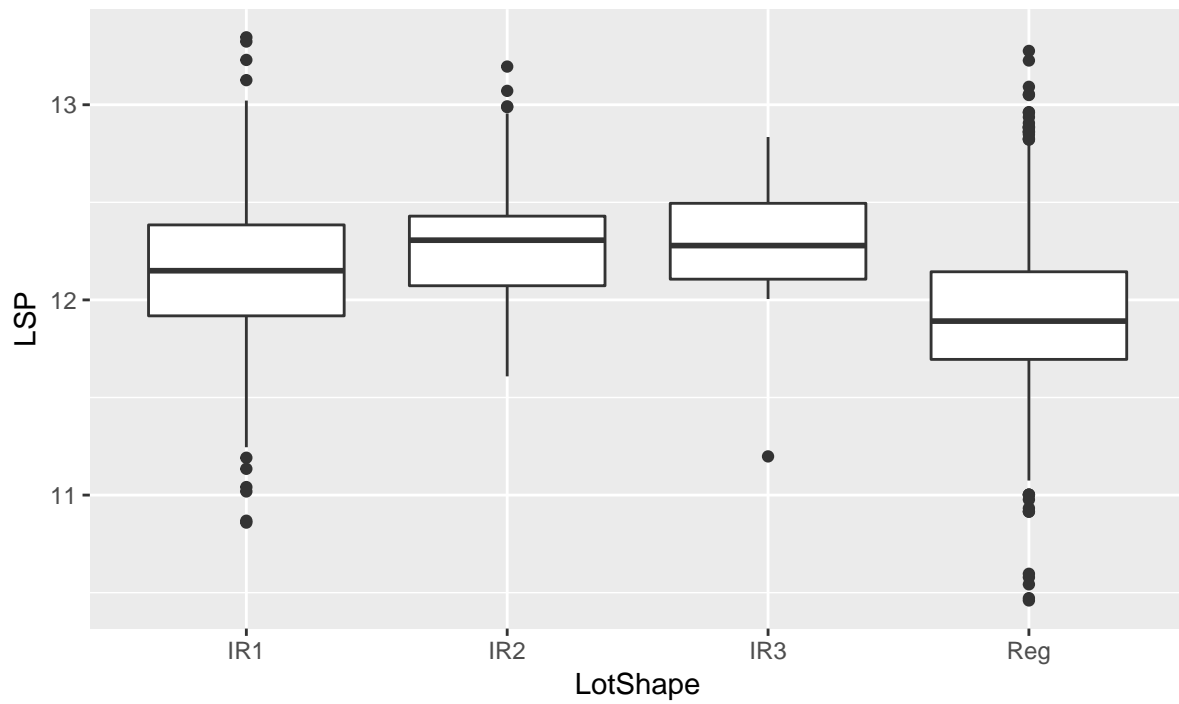
Smooth Density Plot of Logarithmic SalePrice



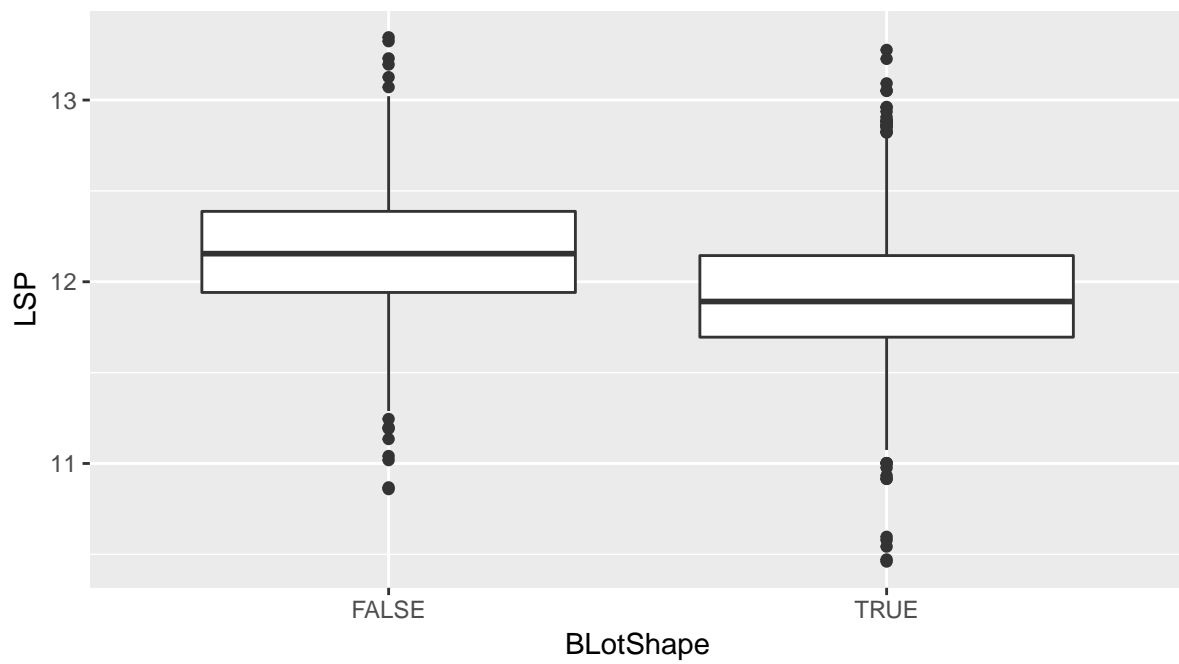
Data Exploration

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.

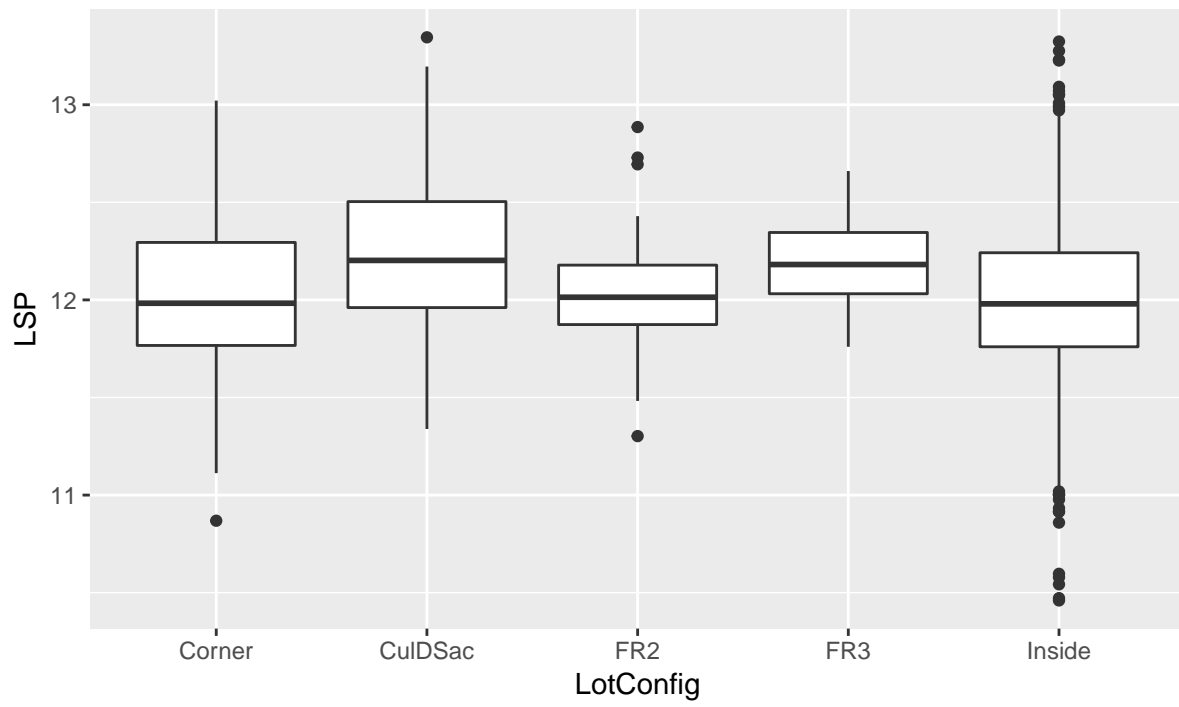
LotShape vs Log SalePrice



New LotShape Feature
LotShape is Regular? T/F

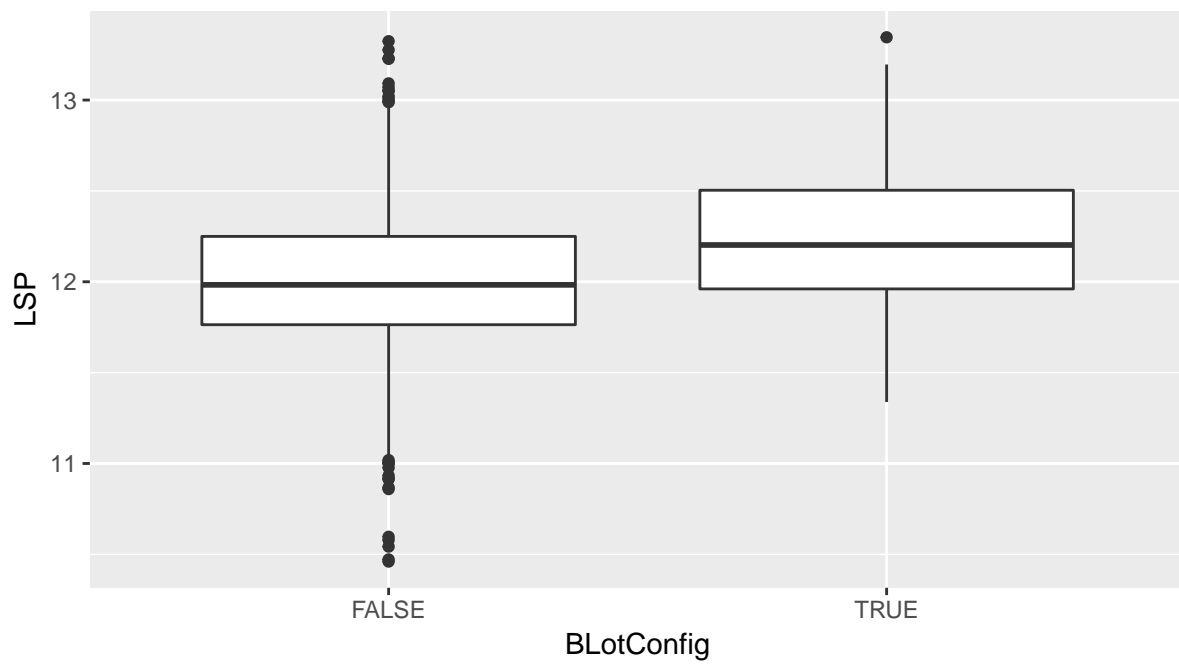


LotConfig vs Log SalePrice



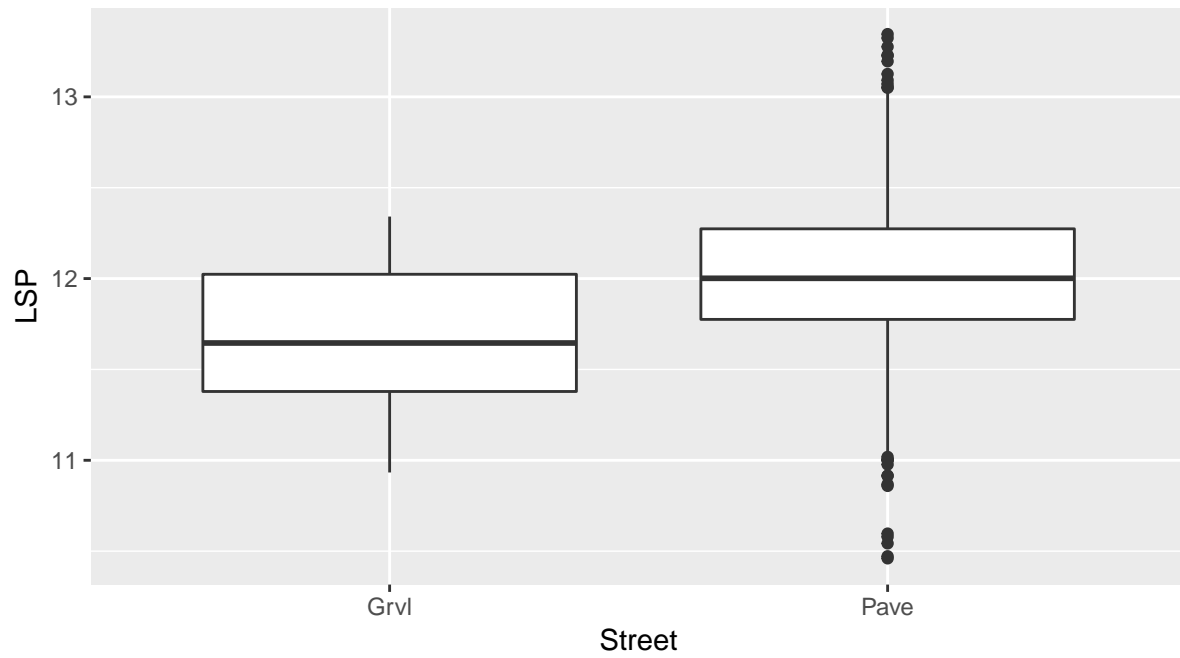
New LotConfig Feature

LotConfig is CulDSac+Fr3? T/F

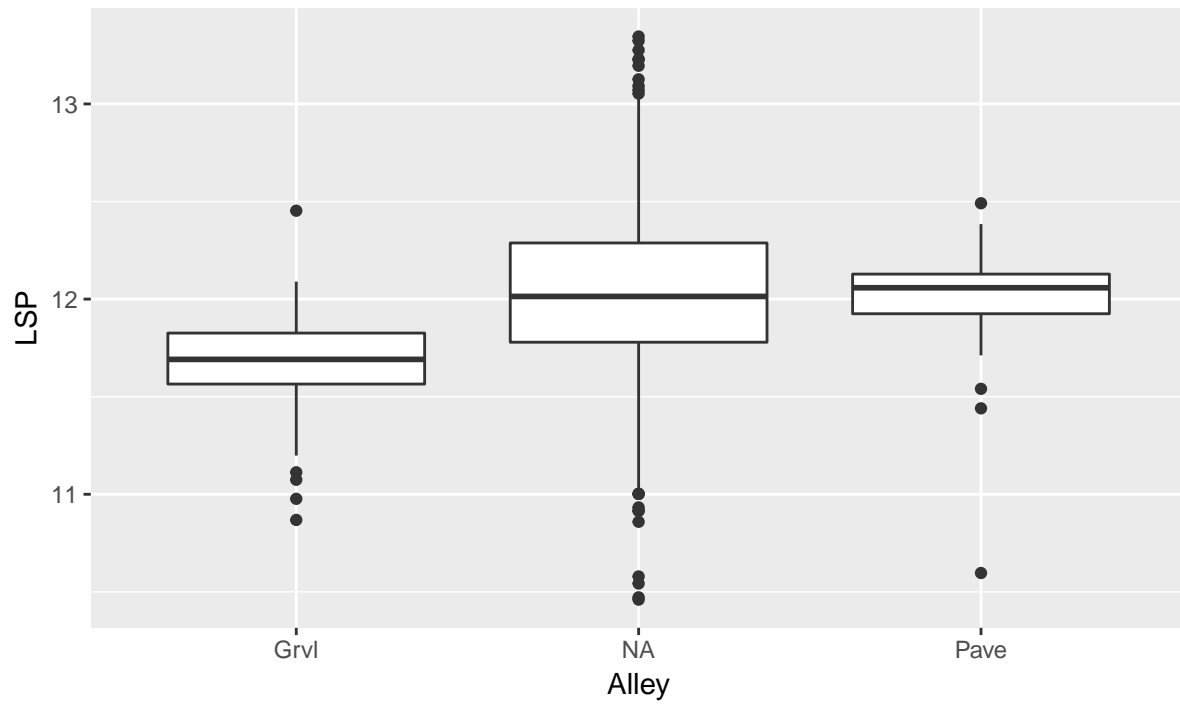


Street vs LSP

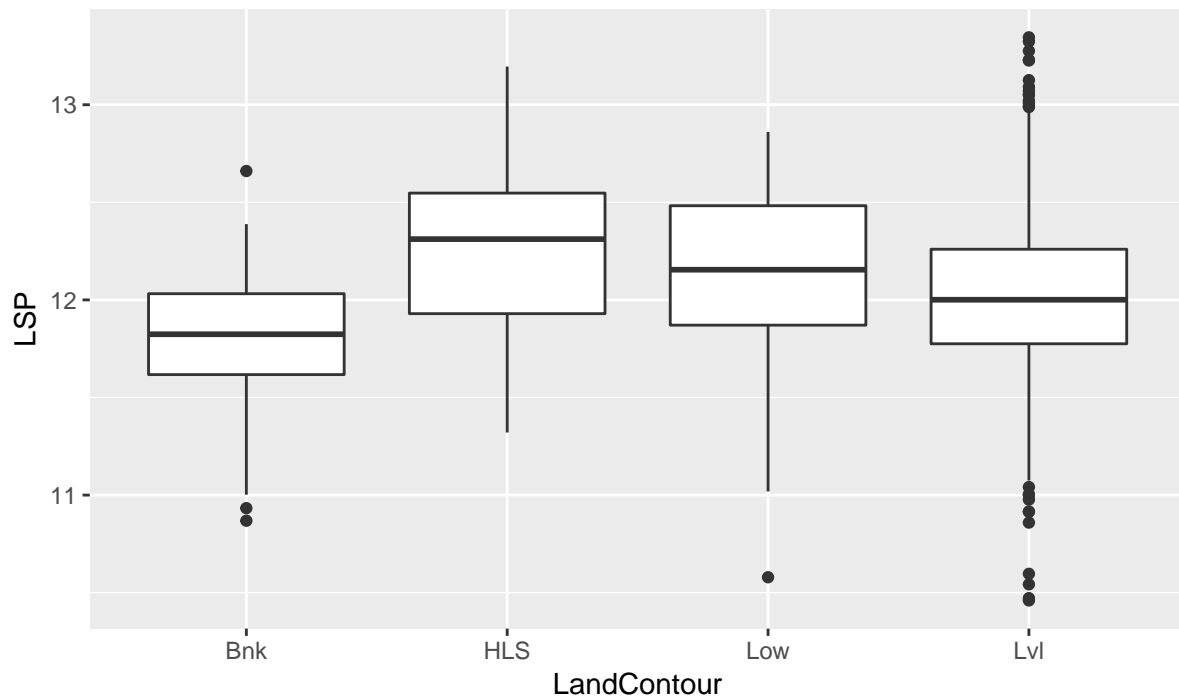
Street is boolean



Alley vs Log SalePrice

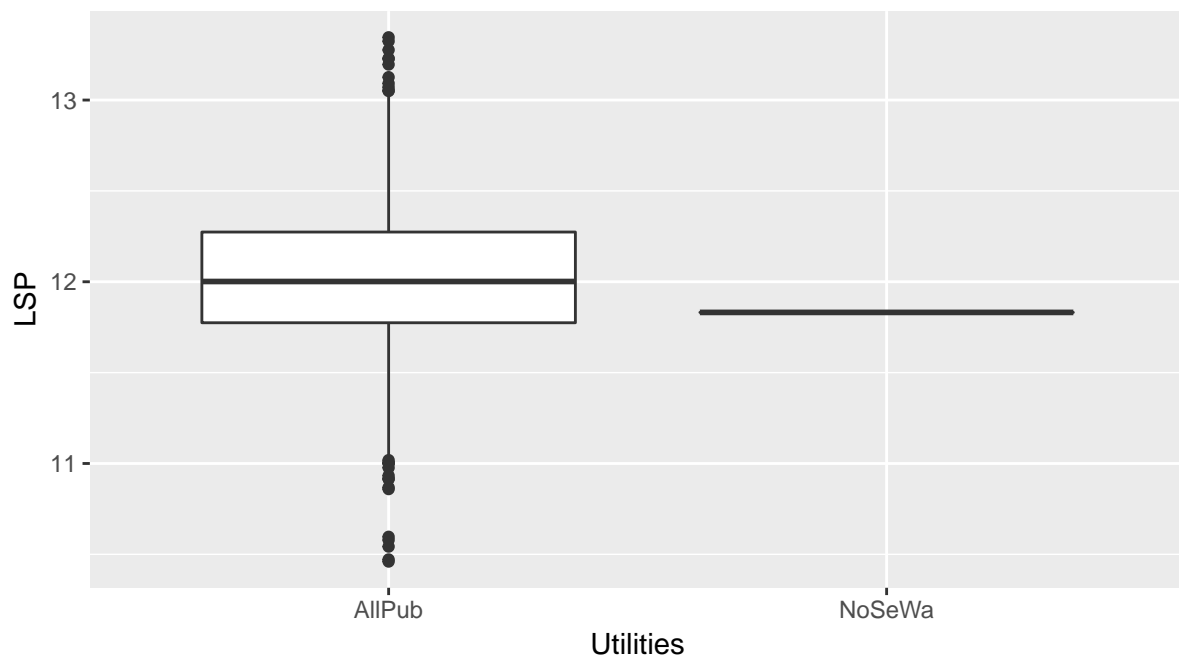


LandContour vs Log SalePrice



Utilities vs LSP

There is only 1 NoSewage entry which may not be enough to represent the effect properly



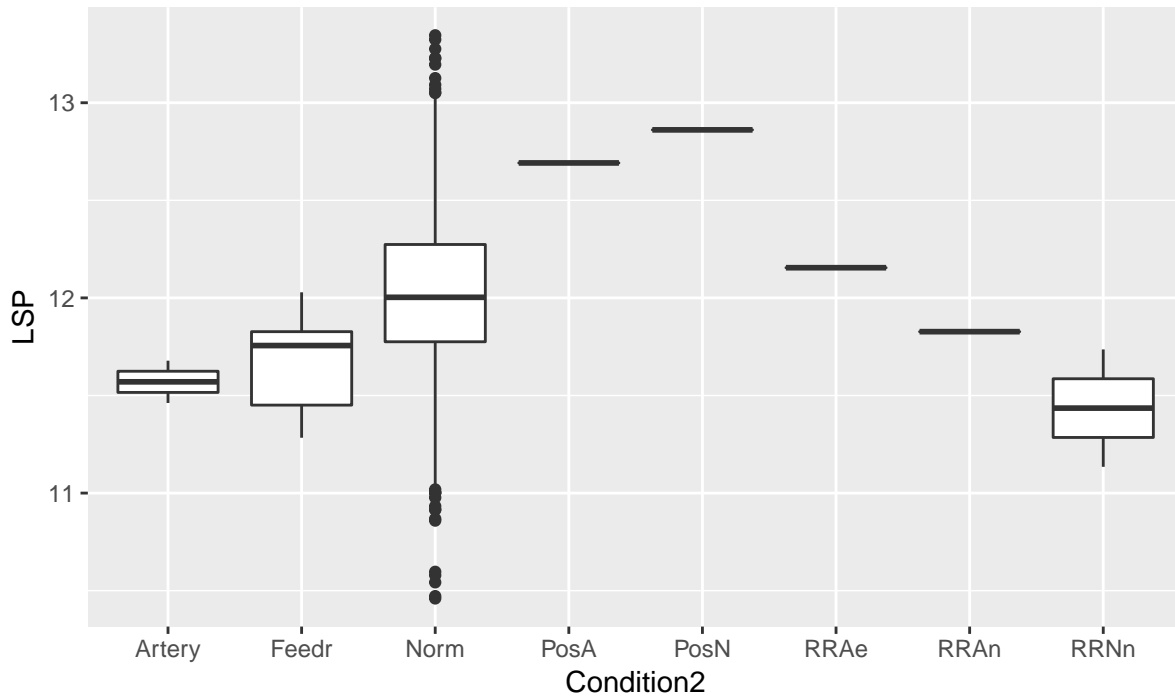
A boxplot comparing the number of children per woman across three countries: Germany, Italy, and the United Kingdom. The y-axis represents the number of children, ranging from 0 to 3. The x-axis lists the countries. Germany shows a median of approximately 1.5, with a box from 1.0 to 2.0 and whiskers extending from 0.5 to 2.5. Italy shows a median of approximately 1.5, with a box from 1.0 to 2.0 and whiskers extending from 0.5 to 2.5. The United Kingdom shows a median of approximately 1.5, with a box from 1.0 to 2.0 and whiskers extending from 0.5 to 2.5. Outliers are present for Germany (at 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0, 10.5, 11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0, 15.5, 16.0, 16.5, 17.0, 17.5, 18.0, 18.5, 19.0, 19.5, 20.0, 20.5, 21.0, 21.5, 22.0, 22.5, 23.0, 23.5, 24.0, 24.5, 25.0, 25.5, 26.0, 26.5, 27.0, 27.5, 28.0, 28.5, 29.0, 29.5, 30.0, 30.5, 31.0, 31.5, 32.0, 32.5, 33.0, 33.5, 34.0, 34.5, 35.0, 35.5, 36.0, 36.5, 37.0, 37.5, 38.0, 38.5, 39.0, 39.5, 40.0, 40.5, 41.0, 41.5, 42.0, 42.5, 43.0, 43.5, 44.0, 44.5, 45.0, 45.5, 46.0, 46.5, 47.0, 47.5, 48.0, 48.5, 49.0, 49.5, 50.0, 50.5, 51.0, 51.5, 52.0, 52.5, 53.0, 53.5, 54.0, 54.5, 55.0, 55.5, 56.0, 56.5, 57.0, 57.5, 58.0, 58.5, 59.0, 59.5, 60.0, 60.5, 61.0, 61.5, 62.0, 62.5, 63.0, 63.5, 64.0, 64.5, 65.0, 65.5, 66.0, 66.5, 67.0, 67.5, 68.0, 68.5, 69.0, 69.5, 70.0, 70.5, 71.0, 71.5, 72.0, 72.5, 73.0, 73.5, 74.0, 74.5, 75.0, 75.5, 76.0, 76.5, 77.0, 77.5, 78.0, 78.5, 79.0, 79.5, 80.0, 80.5, 81.0, 81.5, 82.0, 82.5, 83.0, 83.5, 84.0, 84.5, 85.0, 85.5, 86.0, 86.5, 87.0, 87.5, 88.0, 88.5, 89.0, 89.5, 90.0, 90.5, 91.0, 91.5, 92.0, 92.5, 93.0, 93.5, 94.0, 94.5, 95.0, 95.5, 96.0, 96.5, 97.0, 97.5, 98.0, 98.5, 99.0, 99.5, 100.0, 100.5, 101.0, 101.5, 102.0, 102.5, 103.0, 103.5, 104.0, 104.5, 105.0, 105.5, 106.0, 106.5, 107.0, 107.5, 108.0, 108.5, 109.0, 109.5, 110.0, 110.5, 111.0, 111.5, 112.0, 112.5, 113.0, 113.5, 114.0, 114.5, 115.0, 115.5, 116.0, 116.5, 117.0, 117.5, 118.0, 118.5, 119.0, 119.5, 120.0, 120.5, 121.0, 121.5, 122.0, 122.5, 123.0, 123.5, 124.0, 124.5, 125.0, 125.5, 126.0, 126.5, 127.0, 127.5, 128.0, 128.5, 129.0, 129.5, 130.0, 130.5, 131.0, 131.5, 132.0, 132.5, 133.0, 133.5, 134.0, 134.5, 135.0, 135.5, 136.0, 136.5, 137.0, 137.5, 138.0, 138.5, 139.0, 139.5, 140.0, 140.5, 141.0, 141.5, 142.0, 142.5, 143.0, 143.5, 144.0, 144.5, 145.0, 145.5, 146.0, 146.5, 147.0, 147.5, 148.0, 148.5, 149.0, 149.5, 150.0, 150.5, 151.0, 151.5, 152.0, 152.5, 153.0, 153.5, 154.0, 154.5, 155.0, 155.5, 156.0, 156.5, 157.0, 157.5, 158.0, 158.5, 159.0, 159.5, 160.0, 160.5, 161.0, 161.5, 162.0, 162.5, 163.0, 163.5, 164.0, 164.5, 165.0, 165.5, 166.0, 166.5, 167.0, 167.5, 168.0, 168.5, 169.0, 169.5, 170.0, 170.5, 171.0, 171.5, 172.0, 172.5, 173.0, 173.5, 174.0, 174.5, 175.0, 175.5, 176.0, 176.5, 177.0, 177.5, 178.0, 178.5, 179.0, 179.5, 180.0, 180.5, 181.0, 181.5, 182.0, 182.5, 183.0, 183.5, 184.0, 184.5, 185.0, 185.5, 186.0, 186.5, 187.0, 187.5, 188.0, 188.5, 189.0, 189.5, 190.0, 190.5, 191.0, 191.5, 192.0, 192.5, 193.0, 193.5, 194.0, 194.5, 195.0, 195.5, 196.0, 196.5, 197.0, 197.5, 198.0, 198.5, 199.0, 199.5, 200.0, 200.5, 201.0, 201.5, 202.0, 202.5, 203.0, 203.5, 204.0, 204.5, 205.0, 205.5, 206.0, 206.5, 207.0, 207.5, 208.0, 208.5, 209.0, 209.5, 210.0, 210.5, 211.0, 211.5, 212.0, 212.5, 213.0, 213.5, 214.0, 214.5, 215.0, 215.5, 216.0, 216.5, 217.0, 217.5, 218.0, 218.5, 219.0, 219.5, 220.0, 220.5, 221.0, 221.5, 222.0, 222.5, 223.0, 223.5, 224.0, 224.5, 225.0, 225.5, 226.0, 226.5, 227.0, 227.5, 228.0, 228.5, 229.0, 229.5, 230.0, 230.5, 231.0, 231.5, 232.0, 232.5, 233.0, 233.5, 234.0, 234.5, 235.0, 235.5, 236.0, 236.5, 237.0, 237.5, 238.0, 238.5, 239.0, 239.5, 240.0, 240.5, 241.0, 241.5, 242.0, 242.5, 243.0, 243.5, 244.0, 244.5, 245.0, 245.5, 246.0, 246.5, 247.0, 247.5, 248.0, 248.5, 249.0, 249.5, 250.0, 250.5, 251.0, 251.5, 252.0, 252.5, 253.0, 253.5, 254.0, 254.5, 255.0, 255.5, 256.0, 256.5, 257.0, 257.5, 258.0, 258.5, 259.0, 259.5, 260.0, 260.5, 261.0, 261.5, 262.0, 262.5, 263.0, 263.5, 264.0, 264.5, 265.0, 265.5, 266.0, 266.5, 267.0, 267.5, 268.0, 268.5, 269.0, 269.5, 270.0, 270.5, 271.0, 271.5, 272.0, 272.5, 273.0, 273.5, 274.0, 274.5, 275.0, 275.5, 276.0, 276.5, 277.0, 277.5, 278.0, 278.5, 279.0, 279.5, 280.0, 280.5, 281.0, 281.5, 282.0, 282.5, 283.0, 283.5, 284.0, 284.5, 285.0, 285.5, 286.0, 286.5, 287.0, 287.5, 288.0, 288.5, 289.0, 289.5, 290.0, 290.5, 291.0, 291.5, 292.0, 292.5, 293.0, 293.5, 294.0, 294.5

LandSlope

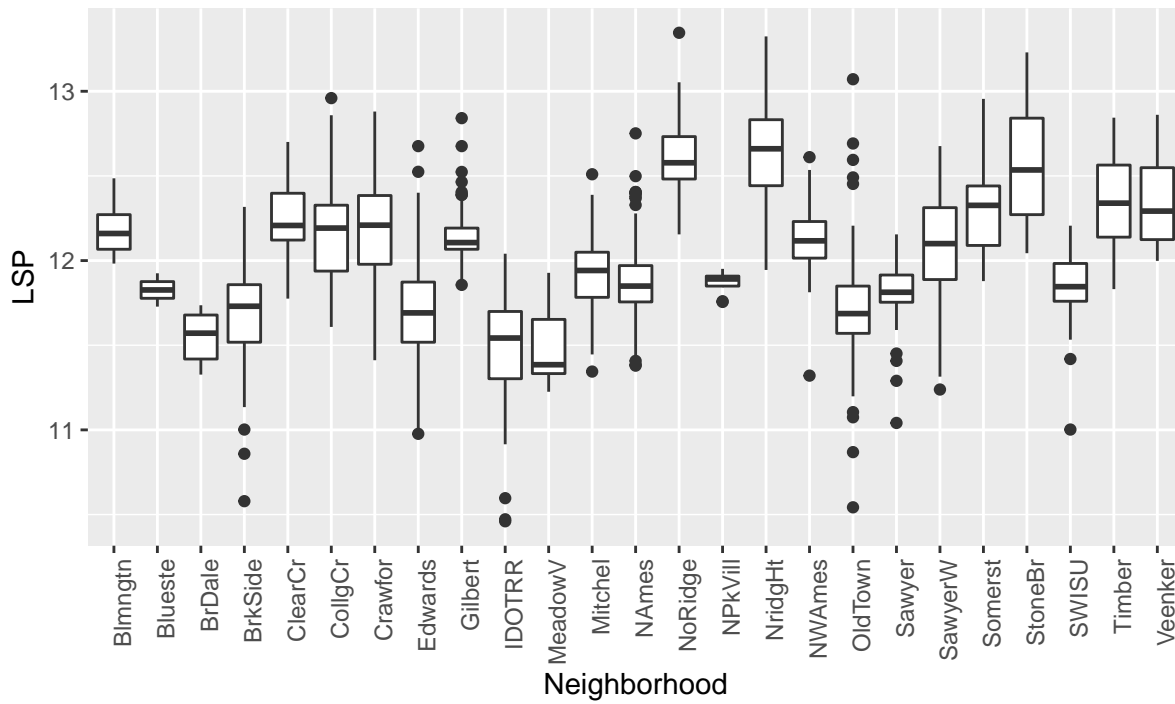
A boxplot showing the distribution of the number of children per woman across ten countries. The y-axis represents the number of children, ranging from 0 to 10. The countries are ordered by their median number of children. The boxplots show the median (horizontal line inside the box), the interquartile range (the box itself), and the range of the data (the whiskers). Outliers are represented by individual dots. The countries are: Niger (median ~2.5), Mali (median ~2.5), Nigeria (median ~2.5), Burkina Faso (median ~2.5), Chad (median ~2.5), Niger (median ~2.5), Mali (median ~2.5), Nigeria (median ~2.5), Burkina Faso (median ~2.5), and Chad (median ~2.5).

Condition1

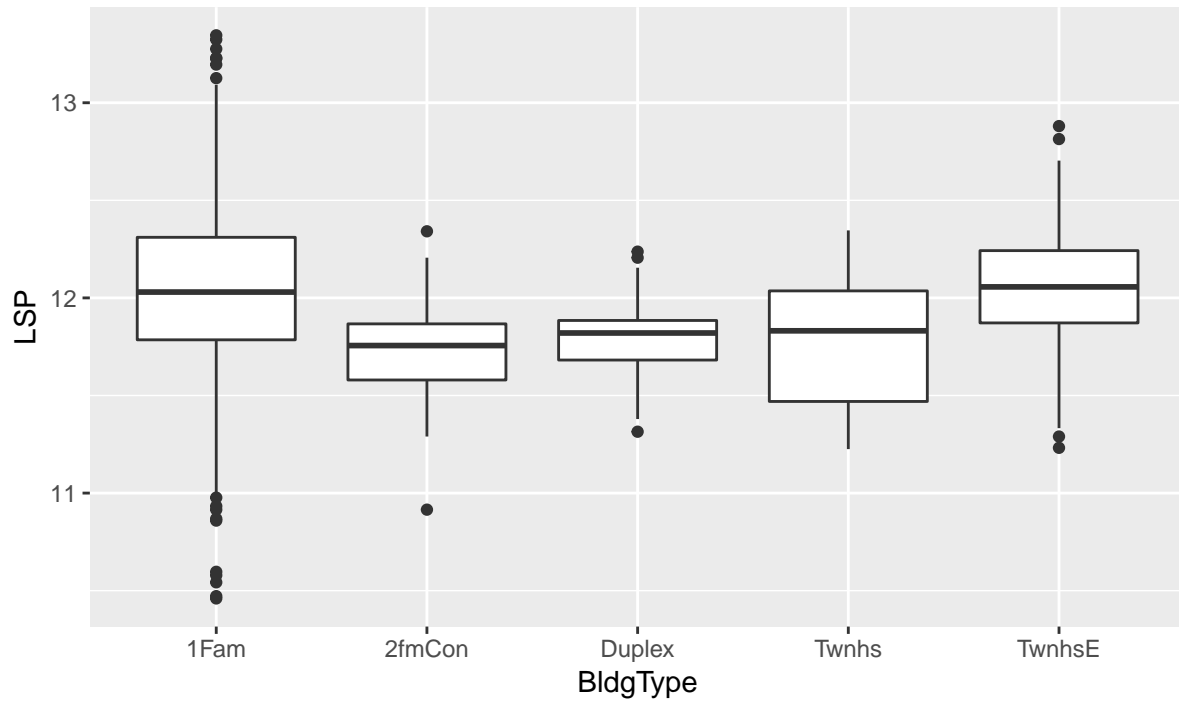
Condition2 vs Log SalePrice



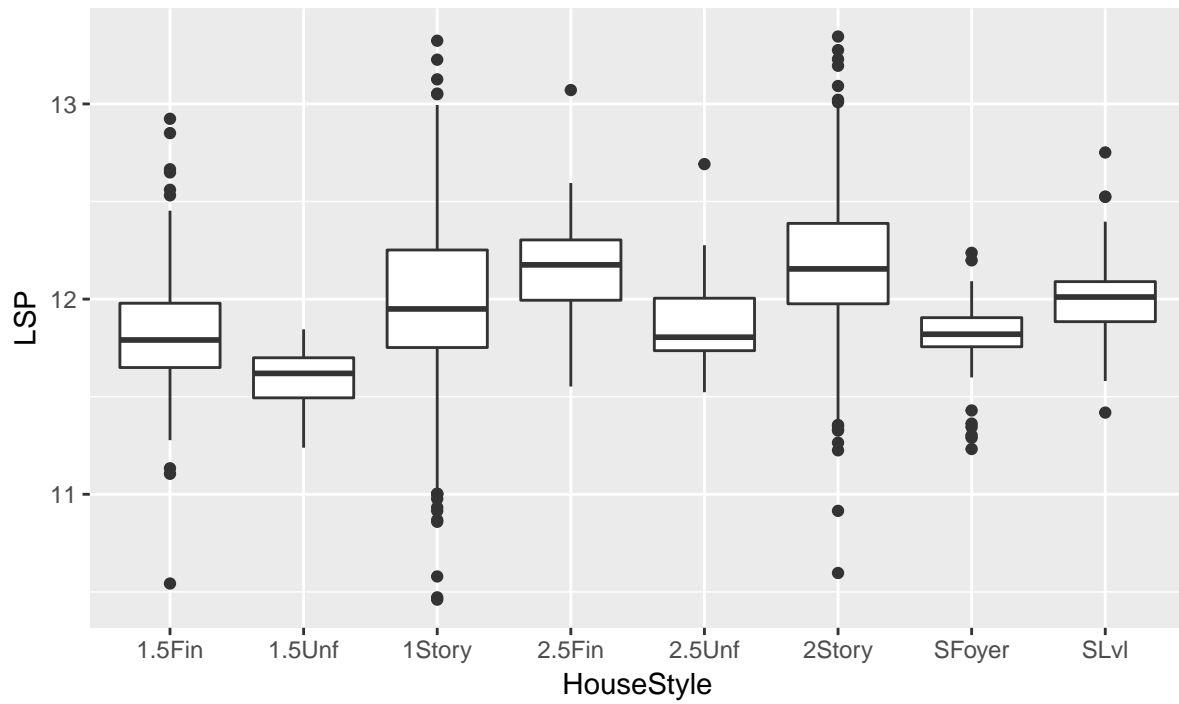
Neighborhood vs Log SalePrice



BldgType vs Log SalePrice

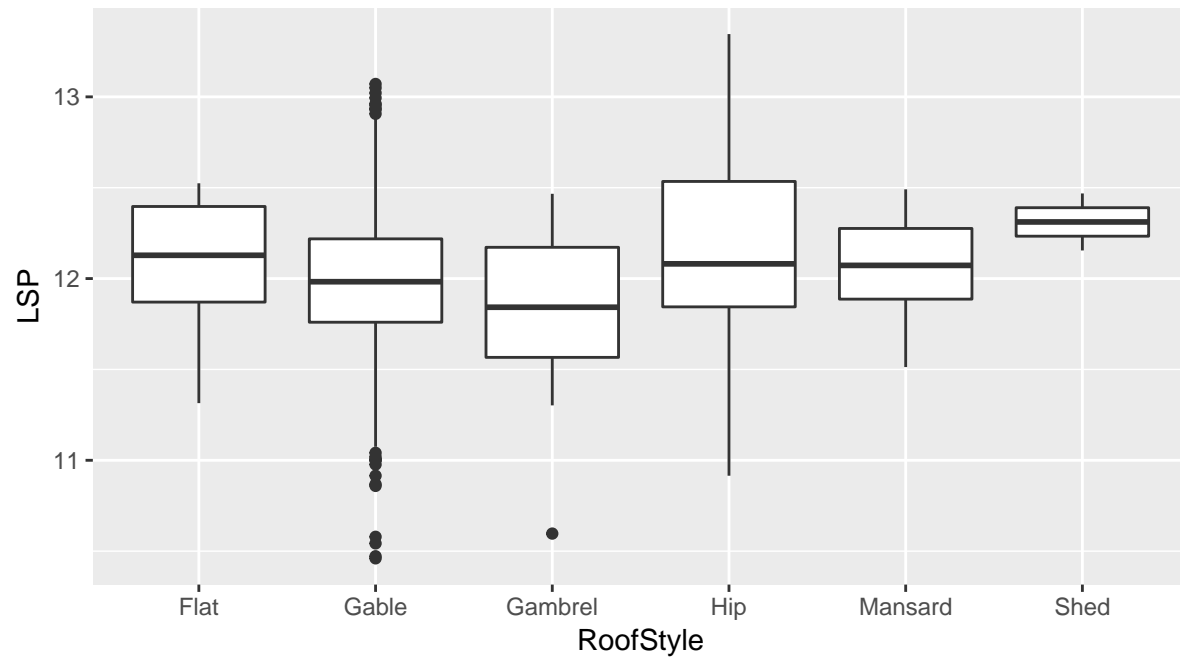


HouseStyle vs Log SalePrice

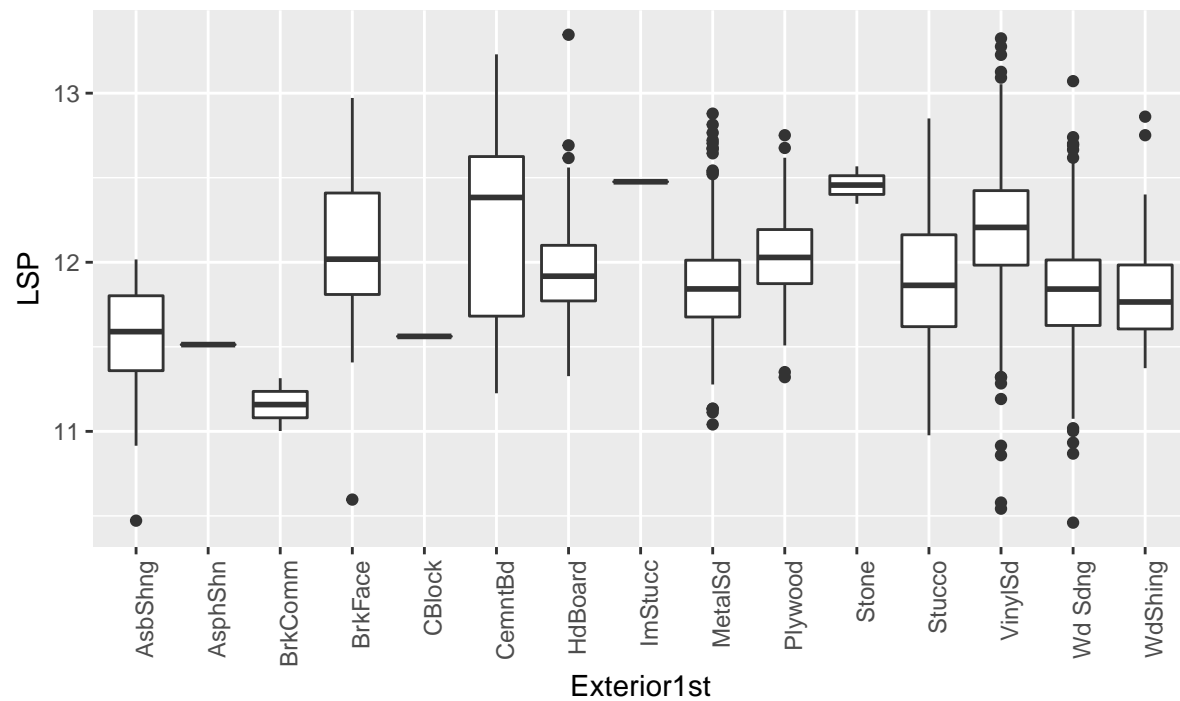


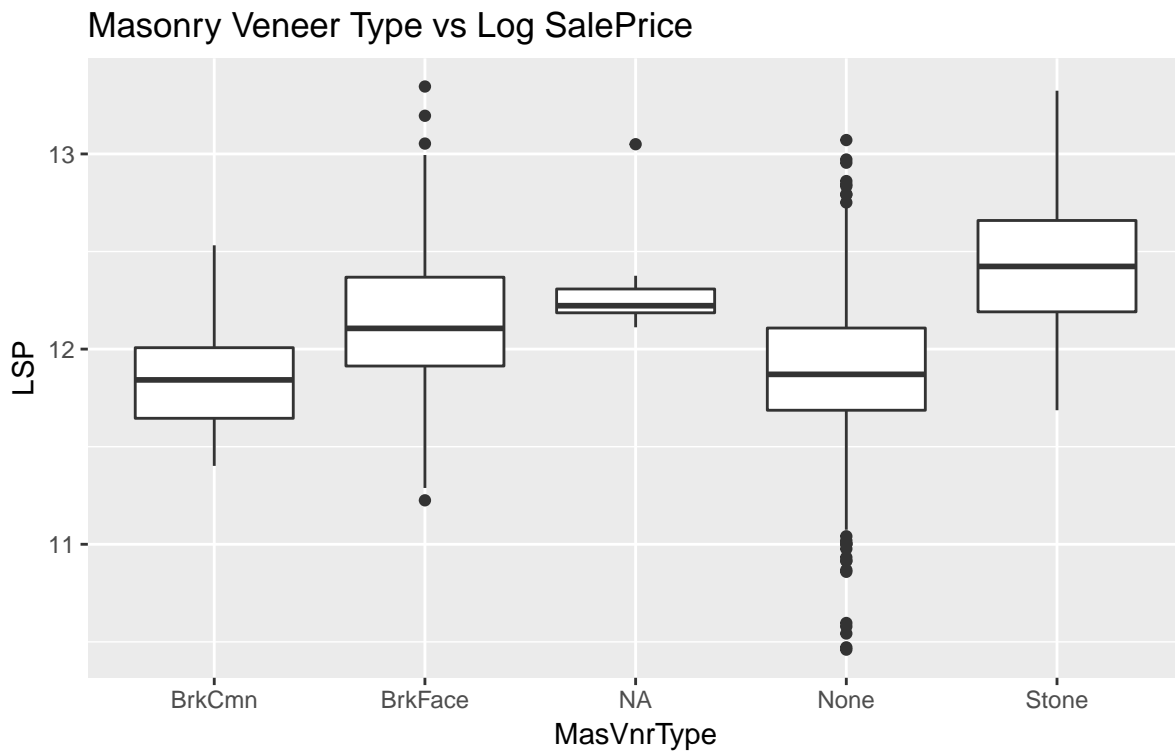
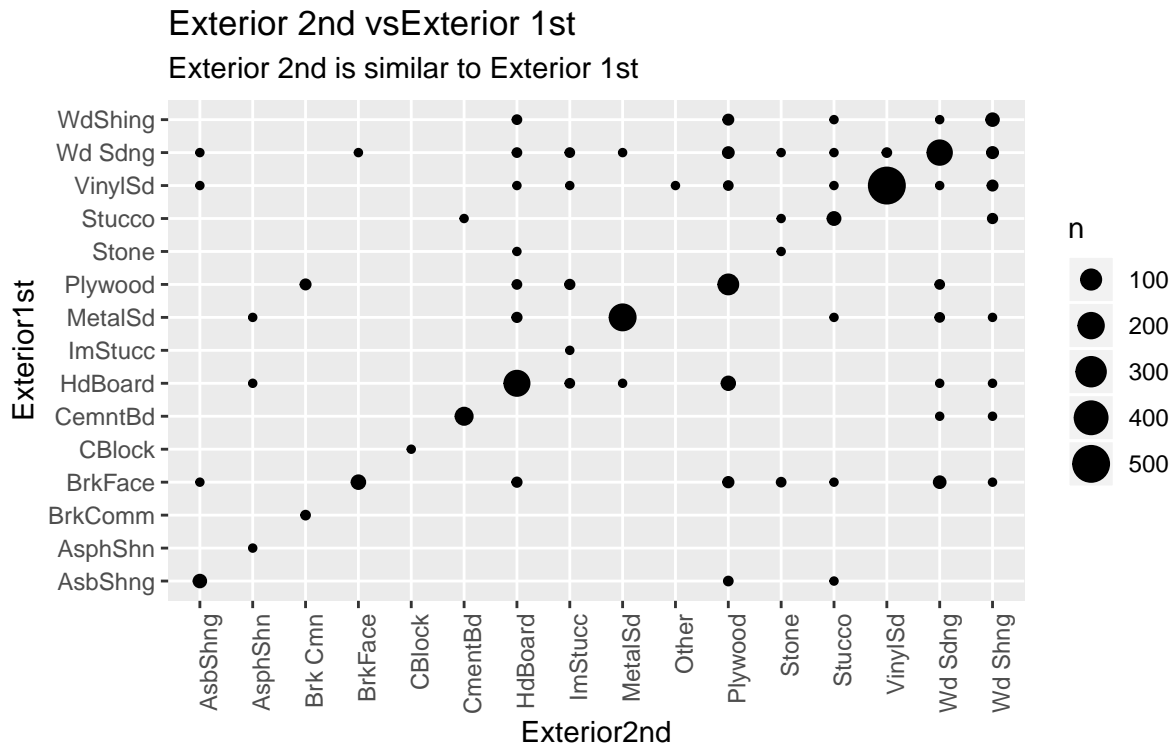
RoofStyle vs LSP

RoofStyle Effect is weak, is dropped



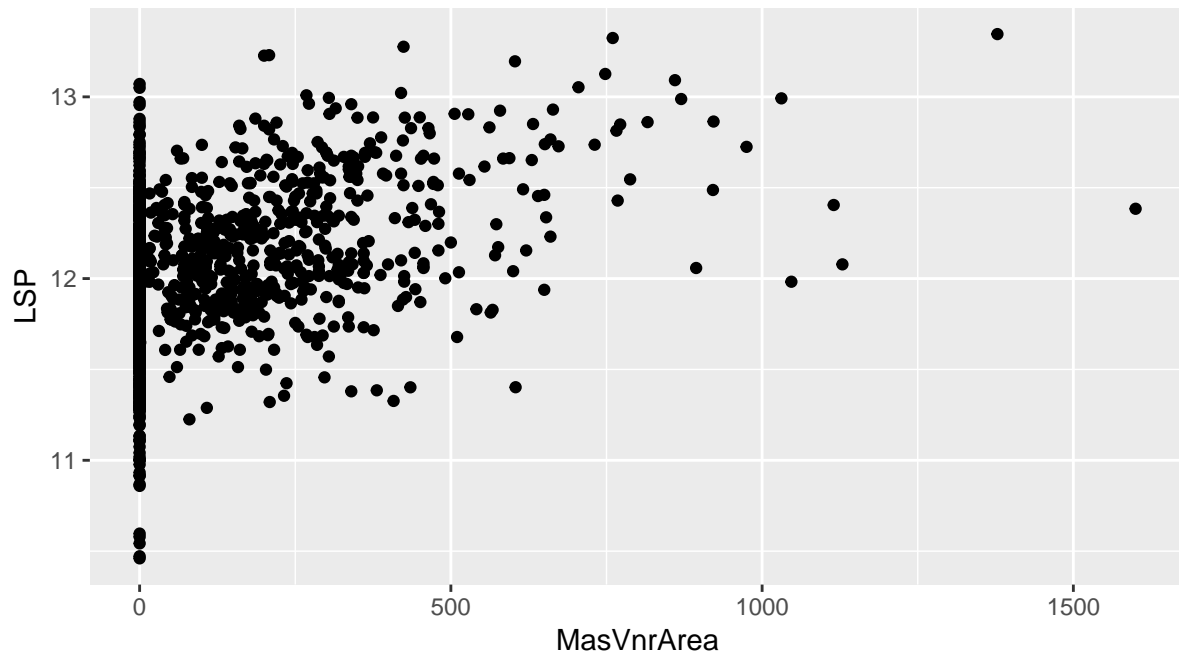
Exterior 1st vs Log SalePrice



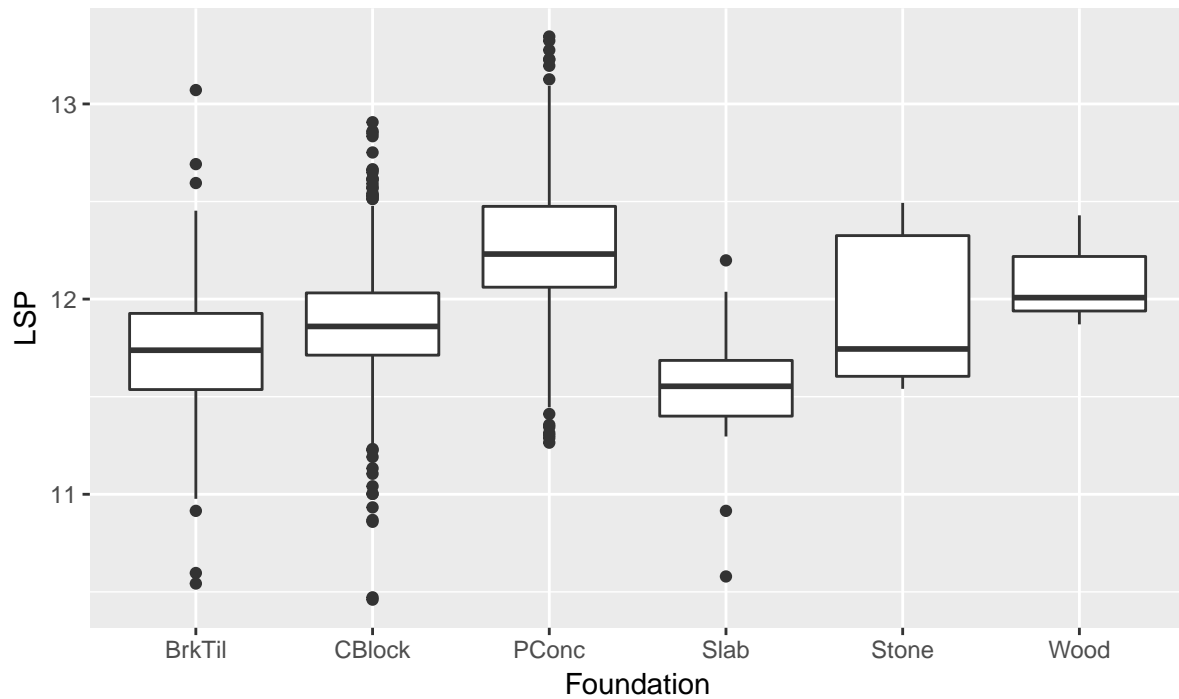


Masonry Veneer Area vs LSP

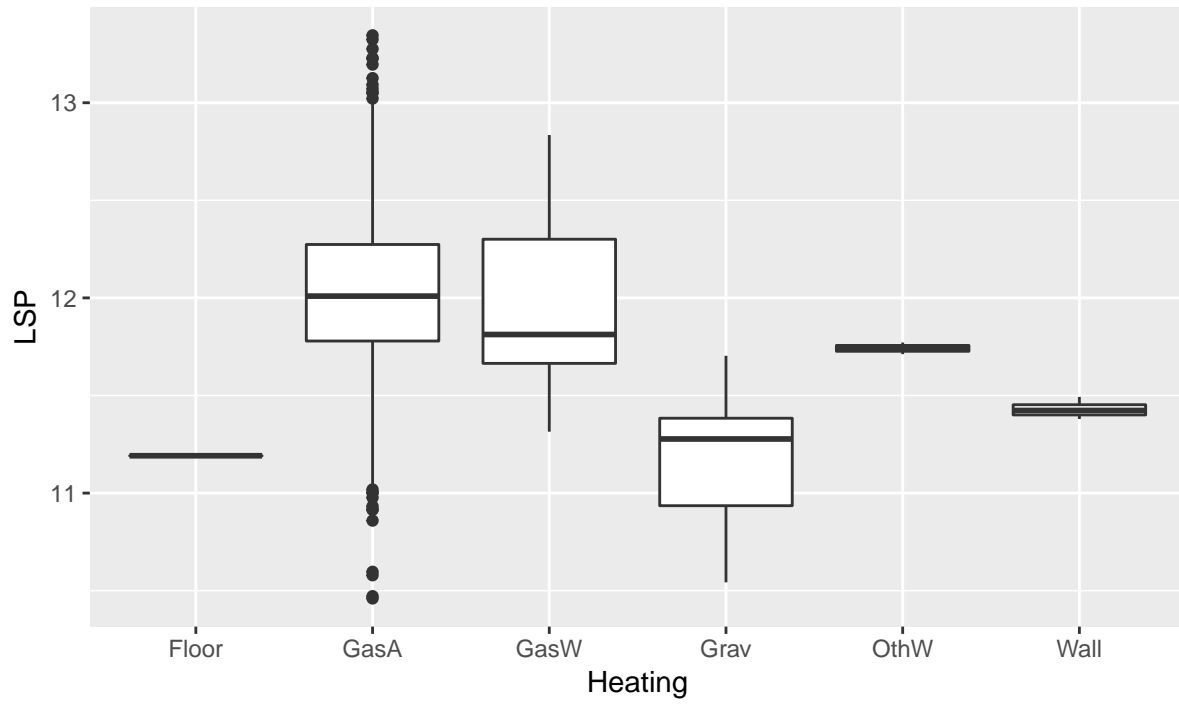
MasVnrArea is not strongly correlated to LSP, is dropped



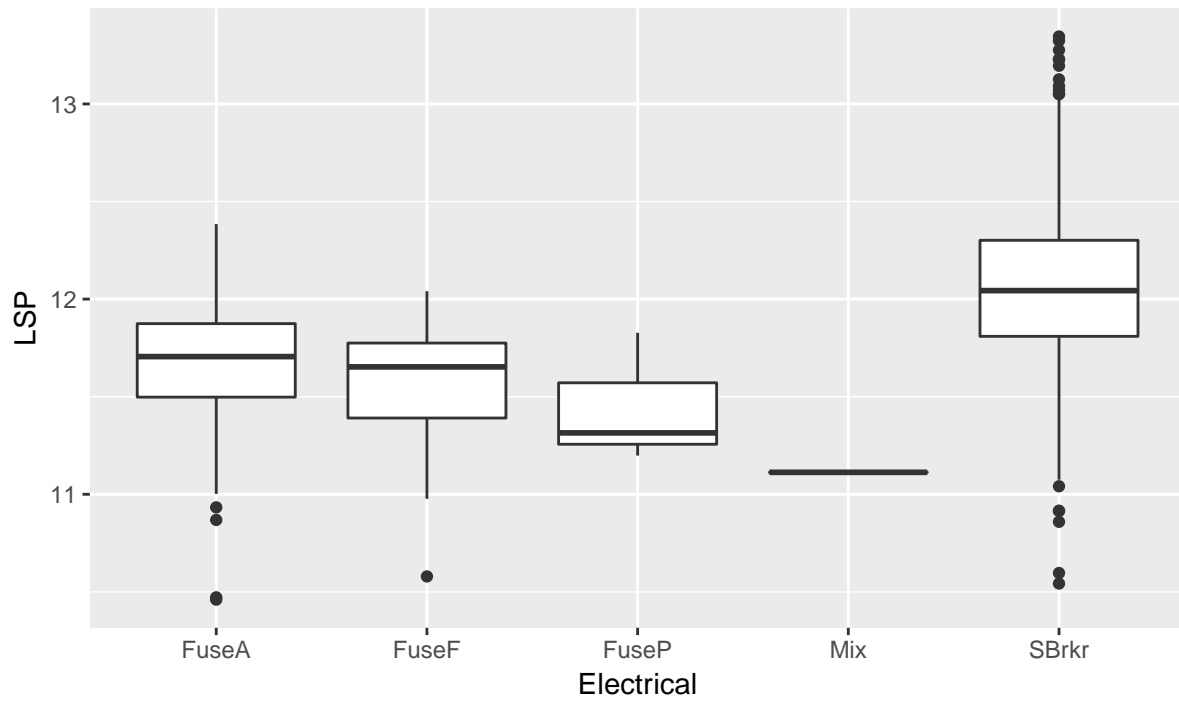
Foundation vs Log SalePrice



Heating vs Log SalePrice



Electrical vs Log SalePrice



The dot plot displays the distribution of LSP values for four categories of fireplaces. The y-axis, labeled 'LSP', has major ticks at 11, 12, and 13. The x-axis, labeled 'Fireplaces', has categories 0, 1, 2, and 3. For 0 fireplaces, there is a dense cluster of points between 10.8 and 12.8, with a few outliers below 11. For 1 fireplace, points are clustered between 11.3 and 13.4, with a few outliers around 11.0. For 2 fireplaces, points are clustered between 11.5 and 13.4. For 3 fireplaces, there are only two points at approximately 12.1 and 12.8.

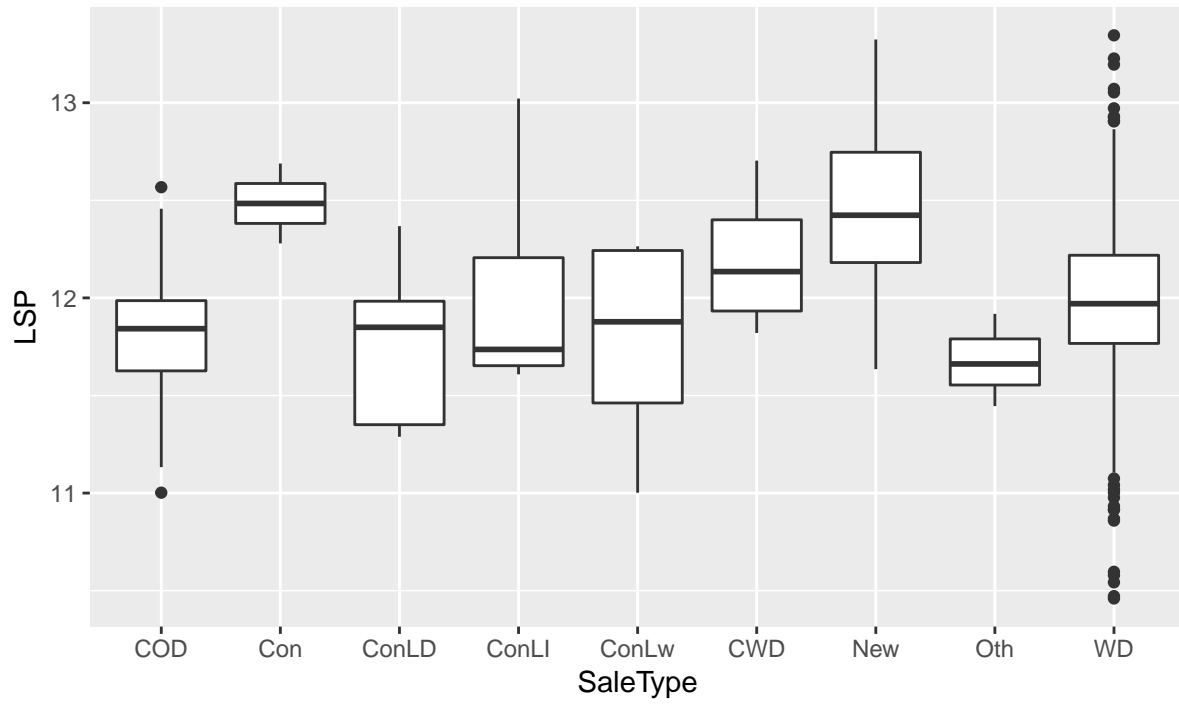
A box plot showing the distribution of LSP (Y-axis, ranging from 10 to 13) across seven functional categories (X-axis: Maj1, Maj2, Min1, Min2, Mod, Sev, Typ). The plot includes individual data points (outliers) for each category. The 'Sev' category shows a very narrow distribution with a single line at approximately 11.75. The 'Typ' category shows the highest range of values, with many outliers above 12.5.

| Functional | Min | Q1 | Median | Q3 | Max | Outliers |
|------------|-------|-------|--------|-------|-------|--|
| Maj1 | 10.8 | 11.8 | 11.85 | 12.15 | 12.7 | 11.0 |
| Maj2 | 10.9 | 11.1 | 11.35 | 11.6 | 11.7 | |
| Min1 | 11.3 | 11.75 | 11.8 | 12.05 | 12.5 | |
| Min2 | 11.4 | 11.75 | 11.85 | 12.0 | 12.1 | 12.7 |
| Mod | 10.9 | 11.55 | 11.85 | 12.15 | 12.5 | 13.2 |
| Sev | 11.75 | 11.75 | 11.75 | 11.75 | 11.75 | |
| Typ | 10.8 | 11.8 | 12.0 | 12.3 | 12.8 | 10.4, 10.5, 10.6, 10.9, 11.0, 11.1, 13.1, 13.2, 13.3, 13.4, 13.5 |

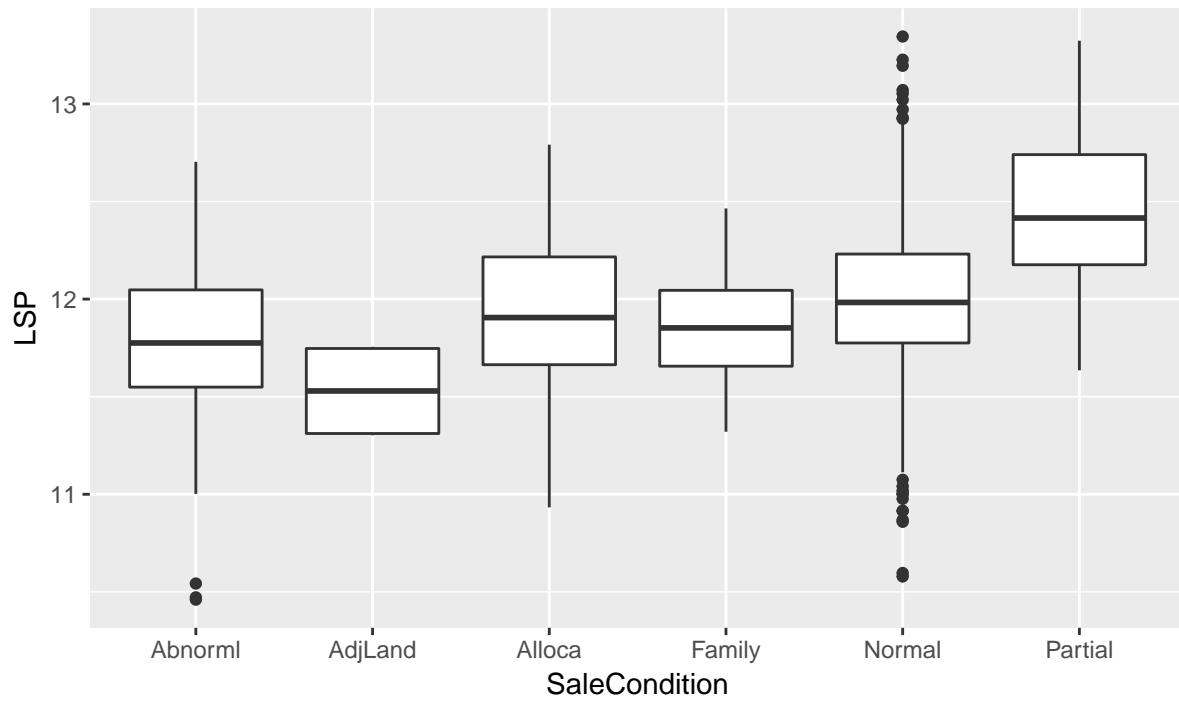
A boxplot showing the distribution of the variable 'FireplaceQu' (Fireplace Quality) across different 'Smoker' categories. The x-axis is labeled 'FireplaceQu' and the y-axis represents the values of the variable. The categories on the x-axis are Ex, Fa, Gd, NA, Po, and TA. The plot shows that the 'TA' (Top of the line) category has the highest median and the most outliers, while the 'Ex' (Excellent) category has a high median and a single outlier. The 'NA' (Not a fireplace) category shows a lower median and more outliers than the other categories.

Boxplot showing the distribution of LSP (Y-axis) across different GarageTypes (X-axis). The Y-axis ranges from 11 to 13. The X-axis categories are 2Types, Attchd, Basment, BuiltIn, CarPort, Detchd, and NA. The plot displays the median, quartiles, and range of LSP values for each GarageType, with individual data points overlaid as dots.

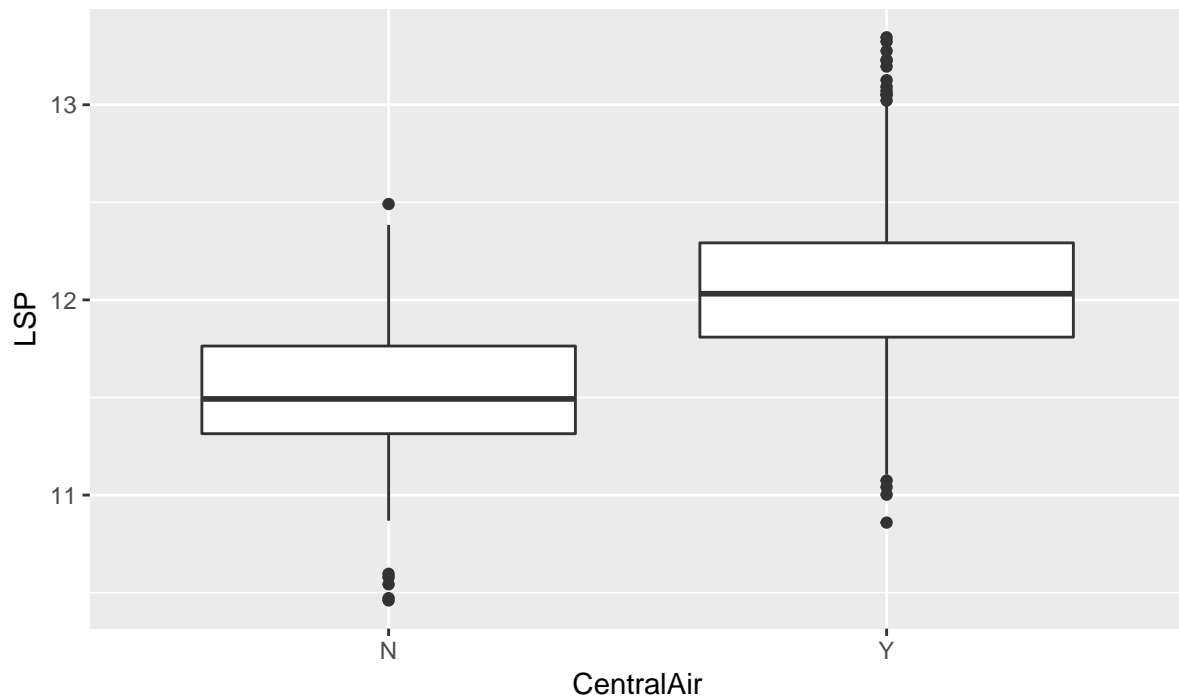
SaleType vs Log SalePrice



SaleCondition vs Log SalePrice

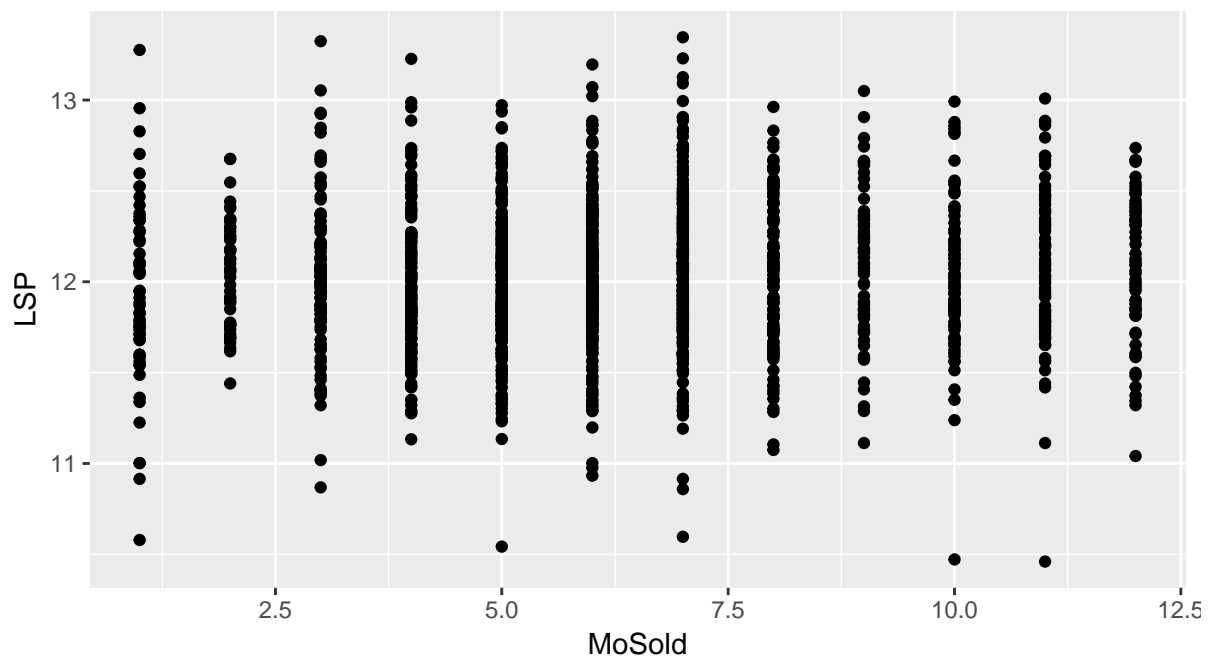


Central Air Conditioning vs Log SalePrice



Month Sold vs LSP

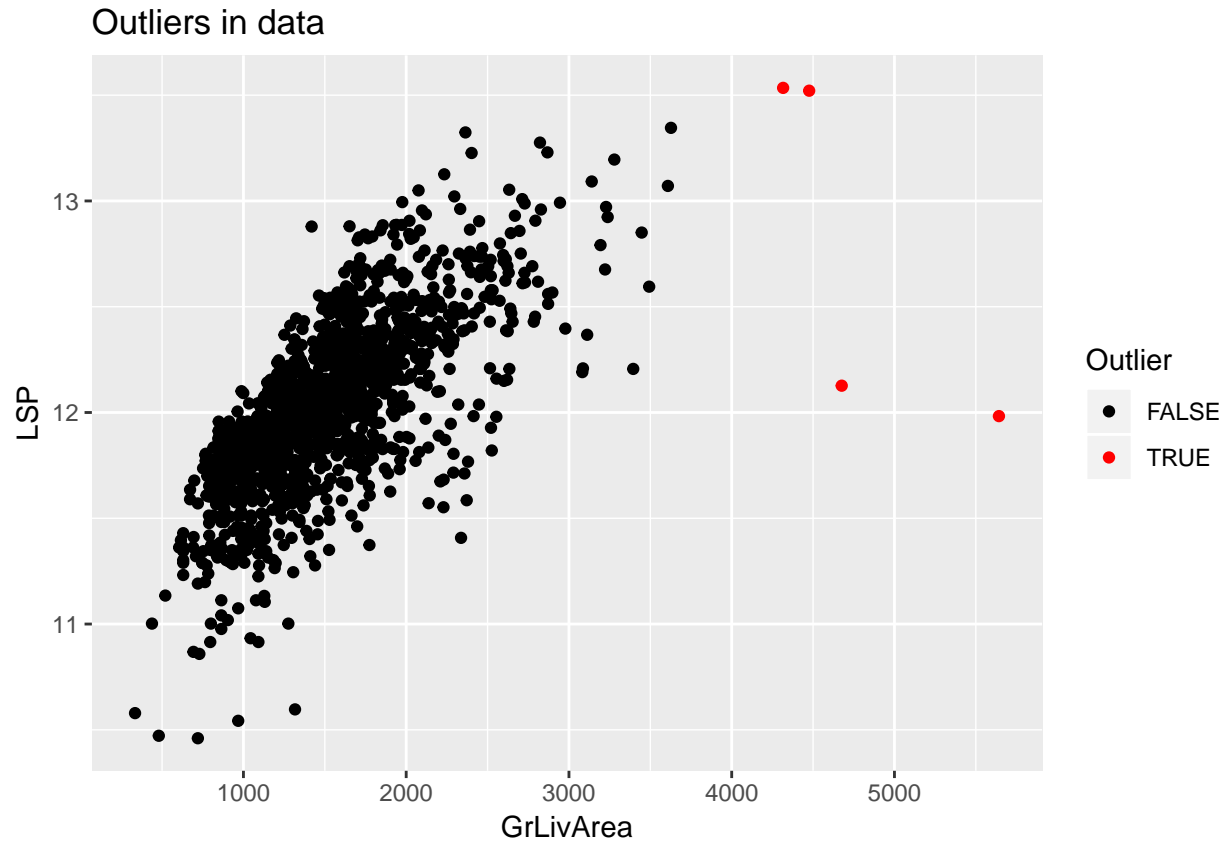
Month Effect is not strong, MoSold is dropped



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.