STA-6543_Final_Project

William Hyltin, Tim Harrison, Holly Milazzo

2024-08-08

Data Cleaning and Observations

Missing values in `LotFrontage`, `MasVnrArea`, `GarageYrBlt`, `Alley`, `MasVnrType`, `BsmtQual`, `BsmtCond`, `BsmtExposure`, `BsmtFinType1`, `BsmtFinType2`, `Electrical`, `FireplaceQu`, `GarageType`, `GarageFinish`, `GarageQual`, `GarageCond`, `PoolQC`, `Fence`, and `MiscFeature`. Some of these missing values may actually be informative, for example several missing values have to do with a basement, so a missing value there may just mean there is no basement in that home. As such we may be able to use logic to impute rather than statistical methods.
`MSSubClass` also comes in as a numeric variable but appears to be a code for a categorical/factor variable. Some other variables are numeric for 1-10 scores, meaning they may be better suited as ordinal categorical variables, but considering the number of other ordinal categorical variable we may only need to address this if it causes any issues.
Note that our test data set does not have the actual sale price of the houses, so post-resample methods will be unavailable.

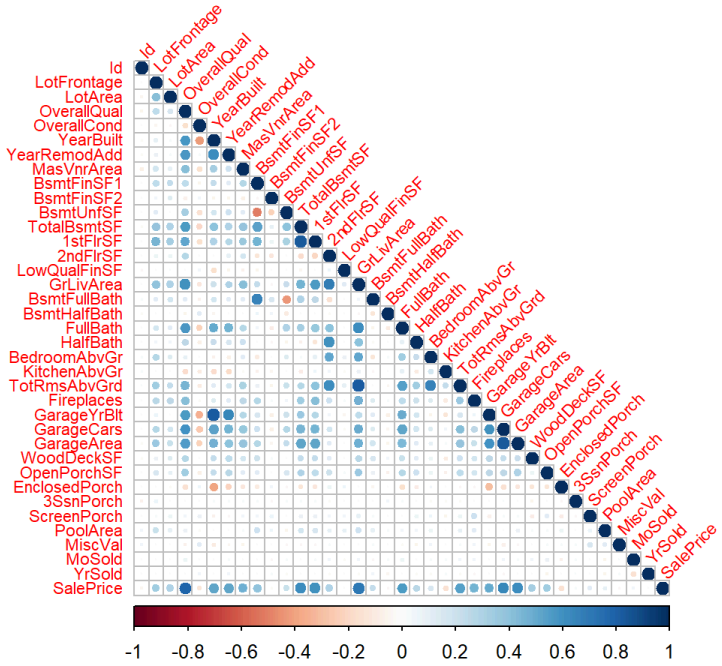
Two records have missing values for some of the basement quality variables where they don't necessarily make sense. The rest of the missing basement variables occur due to there not being a basement. In this case we can still treat these out of place missing variables the same way we do the others, since it is only two records it is unlikely to have a large impact on our models. The same can be said of two other variables, `Electrical` and `MasVnrType`, which have only a small handful of missing values, so would be unlikely to impact a model.

We can't really impute the `GarageYrBlt` logically, i.e. we can't state a year that the garage was built if it doesn't exist, but depending on model performance/ results we may be able to impute that value statistically (mean, median, mode, knn). It would not represent a true year but more a placeholder to represent similar levels of quality across homes.
`LotFrontage` still has missing values, but looking at `LotConfig` suggests these do not appear to intentionally missing. There are records with missing values for `LotFrontage` despite the fact that `LotConfig` states there are two sides with Lot Frontage. The records where `LotConfig` is equal to 'Inside' may indicate there is not Frontage, however the records on the other values are enough to cause some uncertainty. Therefore, it makes the most sense to impute this record as well.
For the two Masonry variables `MasVnrType` and `MasVnrArea`, these are really the only two variables that directly inform each other, so we don't have a way to reasonably impute these two logically. That in mind, there is a value that `MasVnrType` can take for when there is no Masonry Veneer, and in those instances `MasVnrArea` is *usually* 0. Also worth noting that in the instances when the Masonry Variables are missing the `LotConfig` is *usually* 'Inside', which may suggest that properties with an inside lot don't have any Masonry Veneer. Therefore it would make some sense to impute `MasVnrType` and `MasVnrArea` as 'None' and 0 respectively, but again there is enough uncertainty and few enough variables I think statistical imputation methods should be considered first.

Our data should be relatively clean now, and any further imputations can occur within the respective models that we fit.

Exploratory Analysis

Depending on the model that we fit we may have to contend with things like skewness, multicollinearity, near zero variance, and centered and scaling.



Variables Highly Correlated with each other:

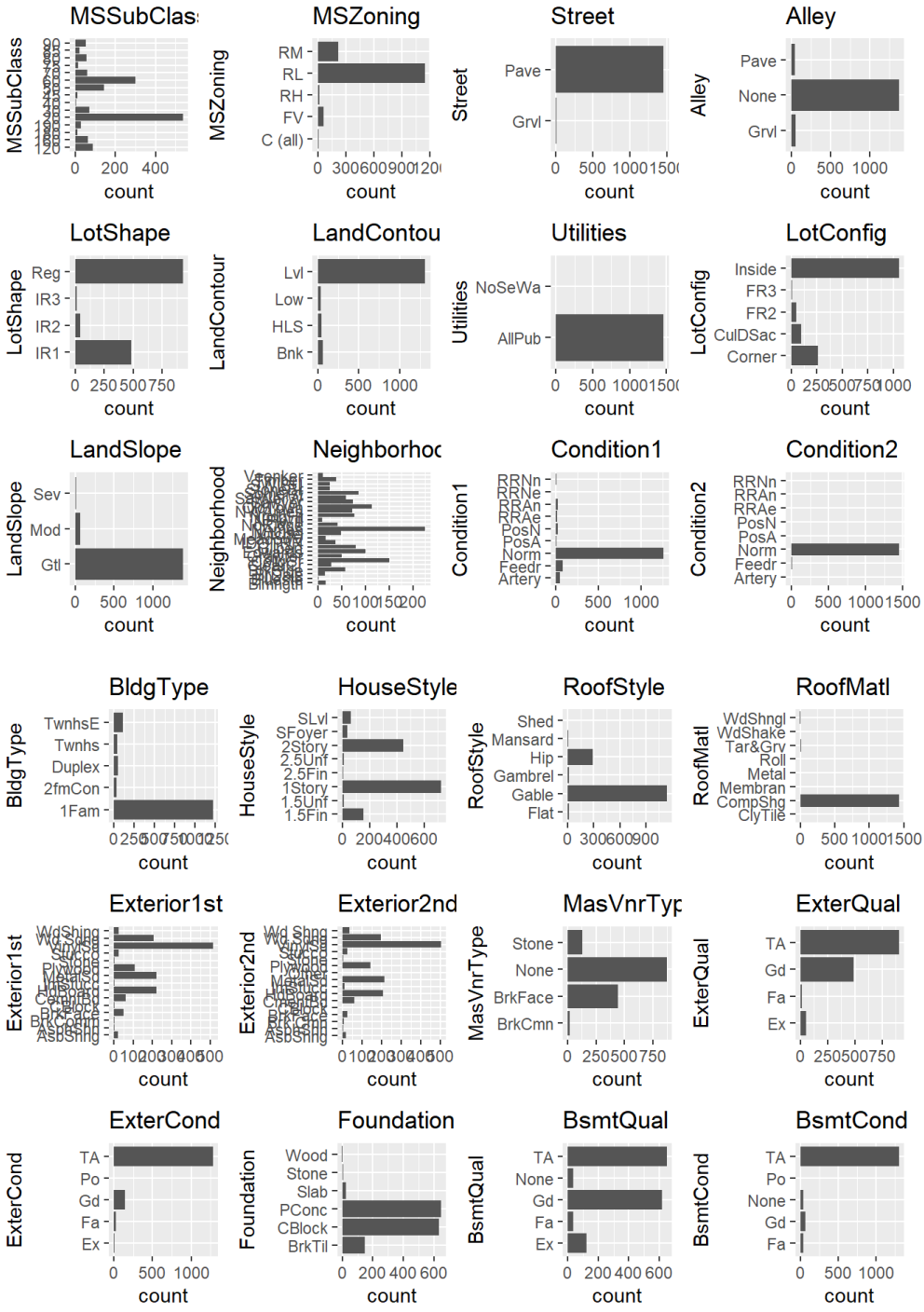
```
##            .
## 1   GrLivArea
## 2  GarageCars
## 3 TotalBsmtSF
## 4   YearBuilt
```

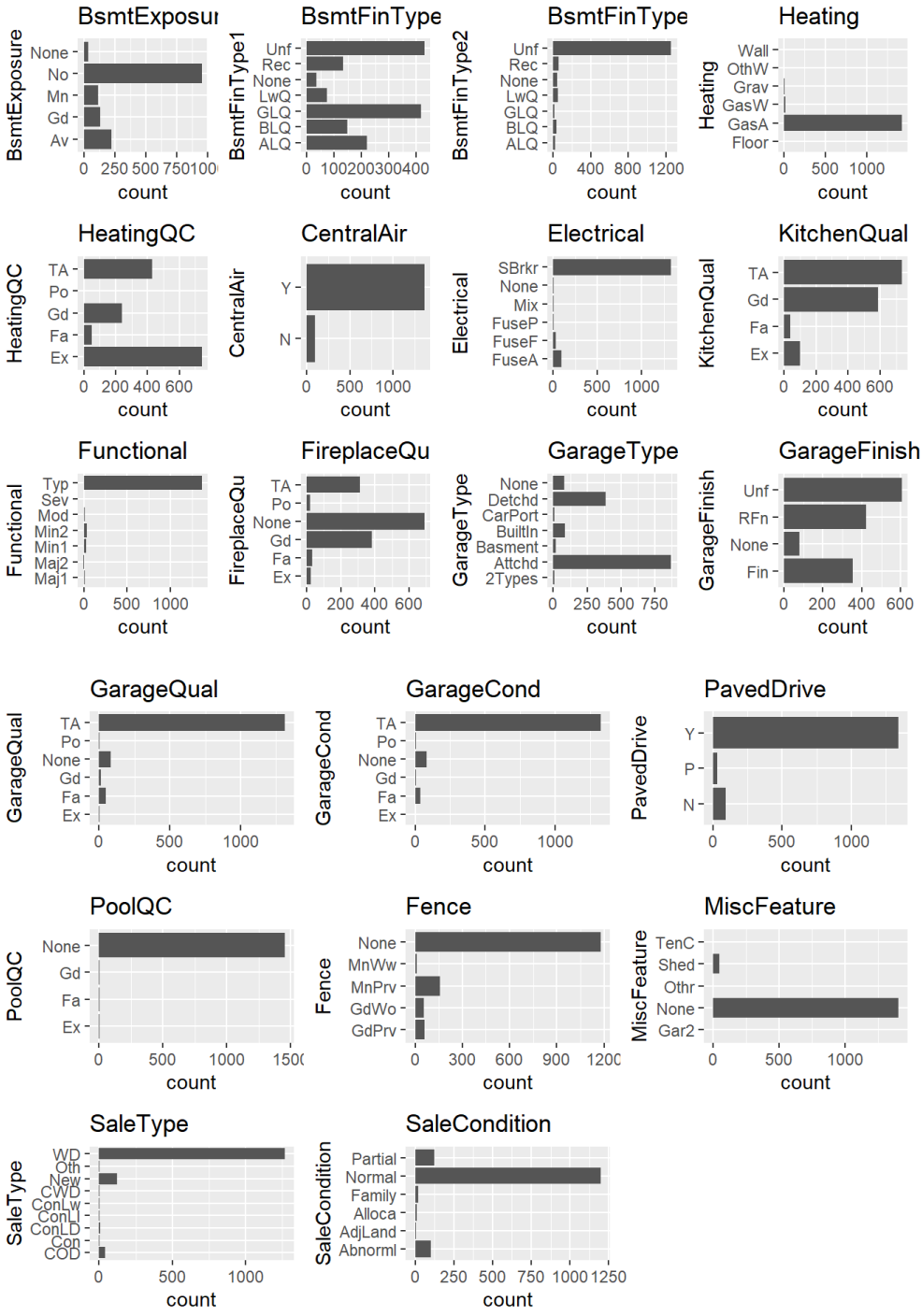
Variables Highly Correlated with Sale Price:

```
##            .
## 1 OverallQual
## 2 TotalBsmtSF
## 3    1stFlrSF
## 4   GrLivArea
## 5  GarageCars
## 6  GarageArea
```

A few variables have relatively high correlation, but for the most most part the variables are pretty independent.

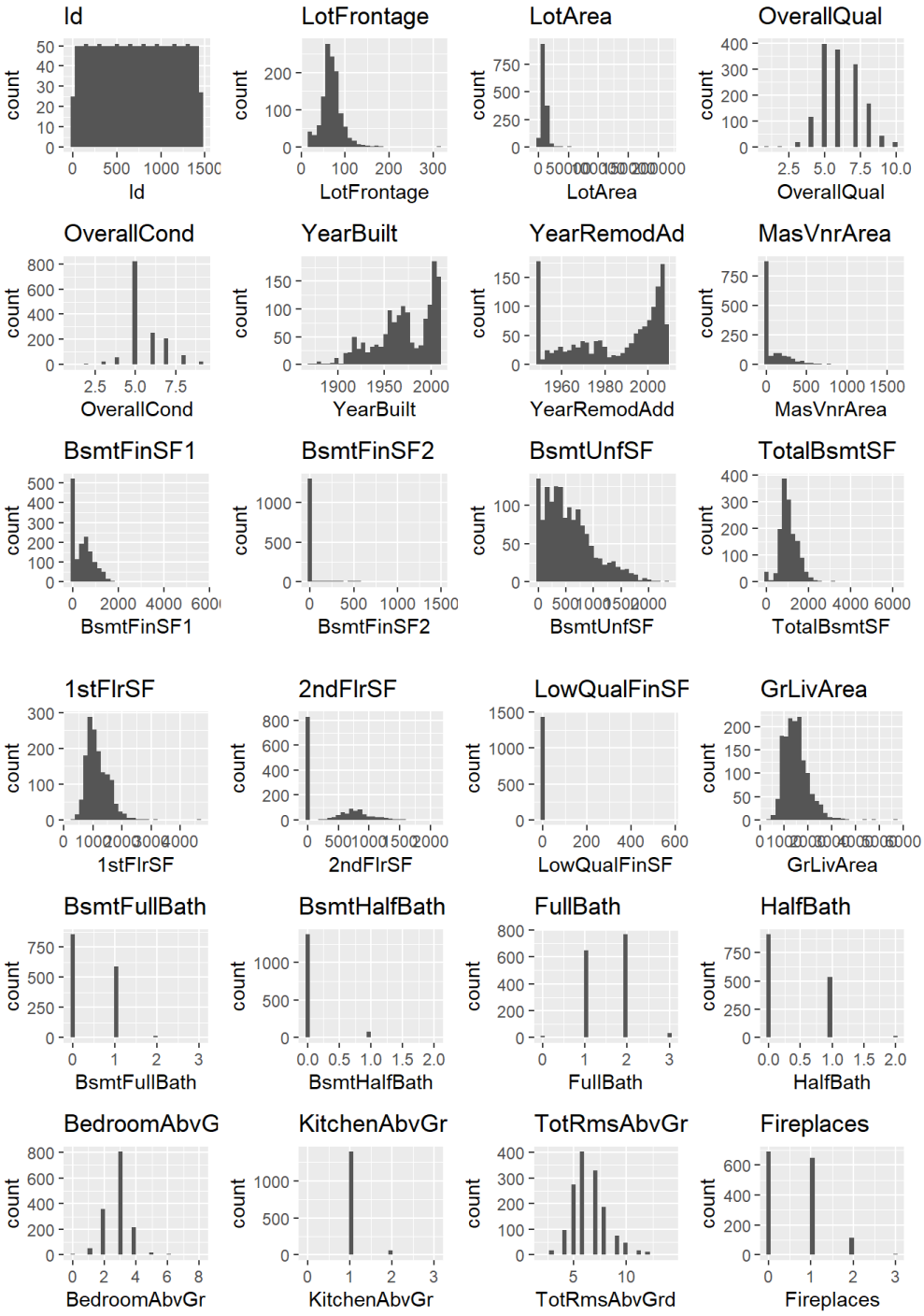
Near Zero Variance Variables:

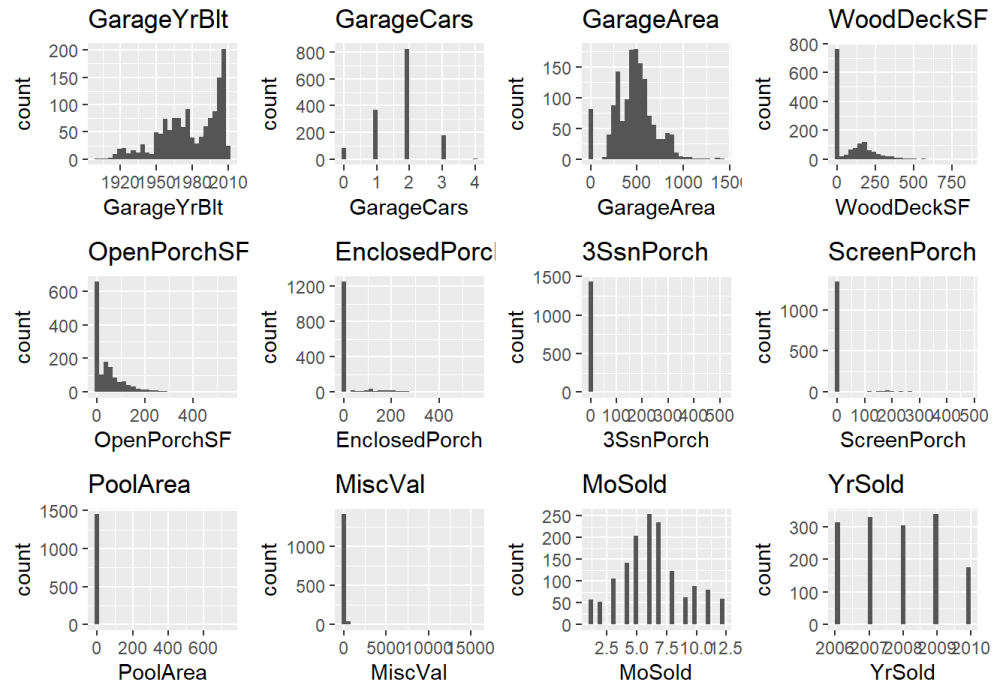

MSSubClas
MSSubClass
count
MSZoning
MSZoning
RM
RL
RH
FV
C (all)
count
Street
Street
Pave
Grvl
count
Alley
Alley
Pave
None
Grvl
count
LotShape
LotShape
Reg
IR3
IR2
IR1
count
LandContou
LandContour
Lvl
Low
HLS
Bnk
count
Utilities
Utilities
NoSeWa
AllPub
count
LotConfig
LotConfig
Inside
FR3
FR2
CulDSac
Corner
count
LandSlope
LandSlope
Sev
Mod
Gtl
count
Neighborhoc
Neighborhood
count
Condition1
Condition1
RRNn
RRNe
RRAn
RRAe
PosN
PosA
Norm
Feedr
Artery
count
Condition2
Condition2
RRNn
RRAn
RRAe
PosN
PosA
Norm
Feedr
Artery
count
BldgType
BldgType
TwnhsE
Twnhs
Duplex
2fmCon
1Fam
count
HouseStyle
HouseStyle
SLvl
SFoyer
2Story
2.5Unf
2.5Fin
1Story
1.5Unf
1.5Fin
count
RoofStyle
RoofStyle
Shed
Mansard
Hip
Gambrel
Gable
Flat
count
RoofMatl
RoofMatl
WdShngl
WdShake
Tar&Grv
Roll
Metal
Membran
CompShg
ClyTile
count
Exterior1st
Exterior1st
count
Exterior2nd
Exterior2nd
count
MasVnrTyp
MasVnrType
Stone
None
BrkFace
BrkCmn
count
ExterQual
ExterQual
TA
Gd
Fa
Ex
count
ExterCond
ExterCond
TA
Po
Gd
Fa
Ex
count
Foundation
Foundation
Wood
Stone
Slab
PConc
CBlock
BrkTil
count
BsmtQual
BsmtQual
TA
None
Gd
Fa
Ex
count
BsmtCond
BsmtCond
TA
Po
None
Gd
Fa
count

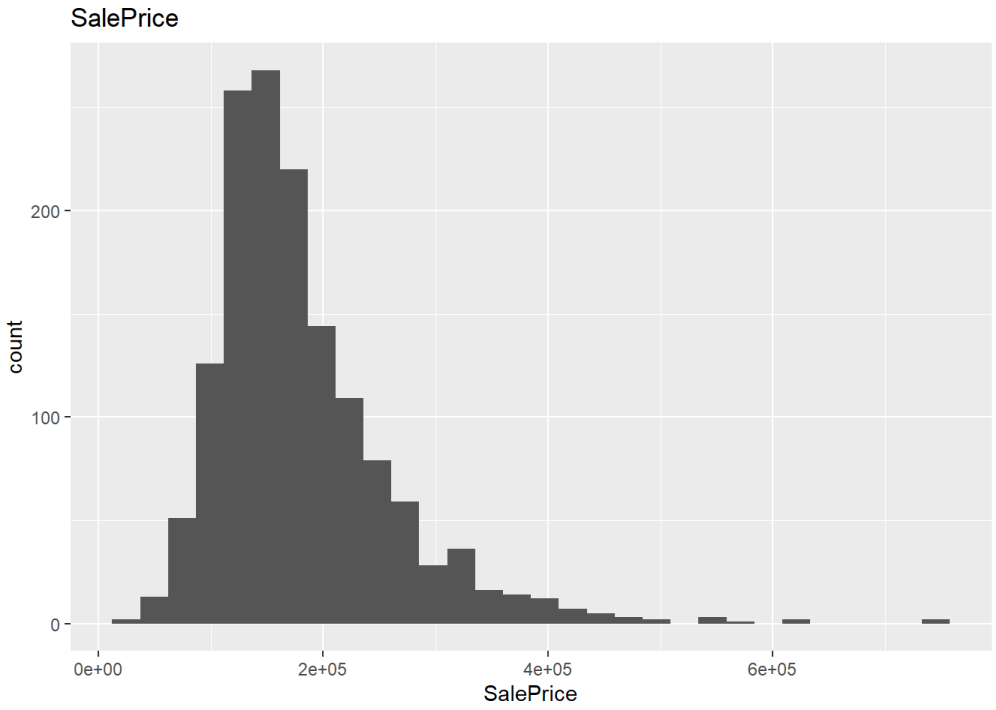

BsmtExposur
BsmtExposure
None
No
Mn
Gd
Av
0 250 500 750100
count
BsmtFinType
BsmtFinType1
Unf
Rec
None
LwQ
GLQ
BLQ
ALQ
0 100200300400
count
BsmtFinType
BsmtFinType2
Unf
Rec
None
LwQ
GLQ
BLQ
ALQ
0 400 800 1200
count
Heating
Heating
Wall
OthW
Grav
GasW
GasA
Floor
0 500 1000
count
HeatingQC
HeatingQC
TA
Po
Gd
Fa
Ex
0 200 400 600
count
CentralAir
CentralAir
Y
N
0 500 1000
count
Electrical
Electrical
SBrkr
None
Mix
FuseP
FuseF
FuseA
0 500 1000
count
KitchenQual
KitchenQual
TA
Gd
Fa
Ex
0 200 400 600
count
Functional
Functional
Typ
Sev
Mod
Min2
Min1
Maj2
Maj1
0 500 1000
count
FireplaceQu
FireplaceQu
TA
Po
None
Gd
Fa
Ex
0 200 400 600
count
GarageType
GarageType
None
Detchd
CarPort
BuiltIn
Basment
Attchd
2Types
0 250 500 750
count
GarageFinish
GarageFinish
Unf
RFn
None
Fin
0 200 400 600
count
GarageQual
GarageQual
TA
Po
None
Gd
Fa
Ex
0 500 1000
count
GarageCond
GarageCond
TA
Po
None
Gd
Fa
Ex
0 500 1000
count
PavedDrive
PavedDrive
Y
P
N
0 500 1000
count
PoolQC
PoolQC
None
Gd
Fa
Ex
0 500 1000 150
count
Fence
Fence
None
MnWw
MnPrv
GdWo
GdPrv
0 300 600 900 120
count
MiscFeature
MiscFeature
TenC
Shed
Othr
None
Gar2
0 500 1000
count
SaleType
SaleType
WD
Oth
New
CWD
ConLw
ConLI
ConLD
Con
COD
0 500 1000
count
SaleCondition
SaleCondition
Partial
Normal
Family
Alloca
AdjLand
Abnorml
0 250 500 750 1000 1250
count

```
##               .
## 1         Street
## 2          Alley
## 3    LandContour
## 4      Utilities
## 5      LandSlope
## 6     Condition2
## 7       RoofMatl
## 8       BsmtCond
## 9   BsmtFinType2
## 10    BsmtFinSF2
## 11       Heating
## 12  LowQualFinSF
## 13  KitchenAbvGr
## 14    Functional
## 15 EnclosedPorch
## 16     3SsnPorch
## 17   ScreenPorch
## 18      PoolArea
## 19        PoolQC
## 20   MiscFeature
## 21       MiscVal
```

We definitely have some variables with Near Zero Variance, so models sensitive to these sorts of variables should have that included in their pre-processing.


Id
LotFrontage
LotArea
OverallQual
OverallCond
YearBuilt
YearRemodAd
MasVnrArea
BsmtFinSF1
BsmtFinSF2
BsmtUnfSF
TotalBsmtSF
1stFlrSF
2ndFlrSF
LowQualFinSF
GrLivArea
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvG
KitchenAbvGr
TotRmsAbvGr
Fireplaces
count





There is definitely some skewness across several of the numeric predictors, so BoxCox transformations will likely be useful for models sensitive to skewness. With the number of categorical variables, it will likely be worth it to create dummy variables to be able to include such variables in certain types of models like Ordinary Least Squares.

Statistical Learning Methods

In our project to predict housing prices, we selected a diverse set of statistical and machine learning methods to ensure robustness and accuracy in our model performance. Each method offers unique benefits suited for different aspects of our data:

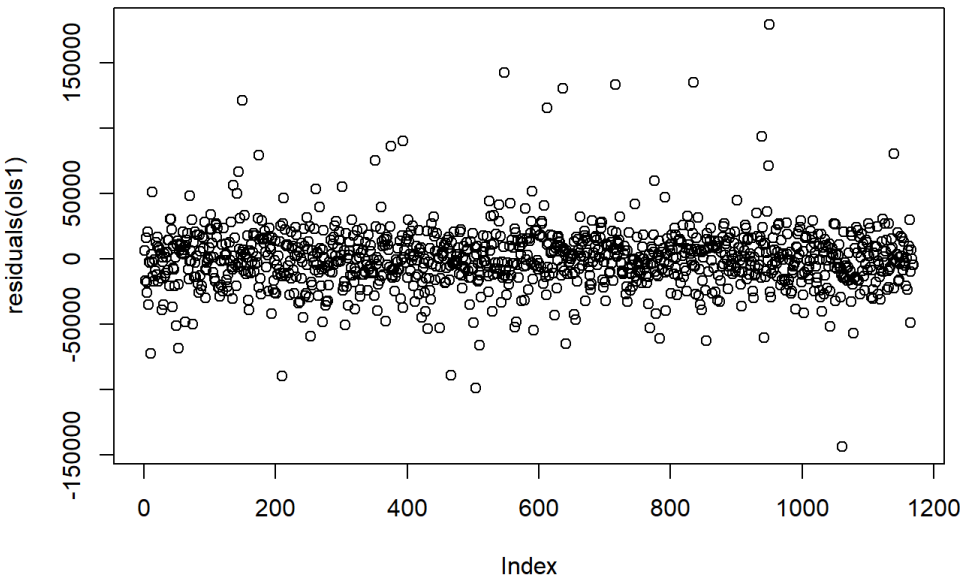
1. **Ordinary Least Squares (OLS)**: We chose OLS for its simplicity and interpretability. It serves as a baseline model, providing an initial understanding of the relationships between features and the target variable.
2. **Random Forest**: Random Forest was chosen because we wanted the advantage of decision trees to handle the number of qualitative predictors in the dataset, wile still improving predictability over regular decision trees and being able to measure variable importance.
3. **Support Vector Machines (SVM)**: SVM is included due to its effectiveness in high-dimensional spaces and its robustness against overfitting, especially in cases where the number of features is greater than the number of observations. SVM's ability to use different kernel functions allows us to model non-linear relationships which is important for accurate house price predictions.

Each of these methods was selected to complement the others, covering a range of assumptions about data distribution and structure. This varied approach ensures that we can tackle the problem of predicting housing prices from multiple angles, enhancing the overall accuracy and reliability of our results.

We will start with OLS, so we will need to create dummy variables to be able to fit the model. Then we will split the train set into a training and test set, since the provided test set does not have values for the response variable.

Test Set Performance:

```
## Linear Regression
##
## 1169 samples
##  231 predictor
##
## Pre-processing: centered (117), scaled (117), Box-Cox transformation
##  (13), nearest neighbor imputation (117), remove (114)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 1050, 1053, 1052, 1052, 1053, 1052, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   26429.13  0.8936074  18160.13
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

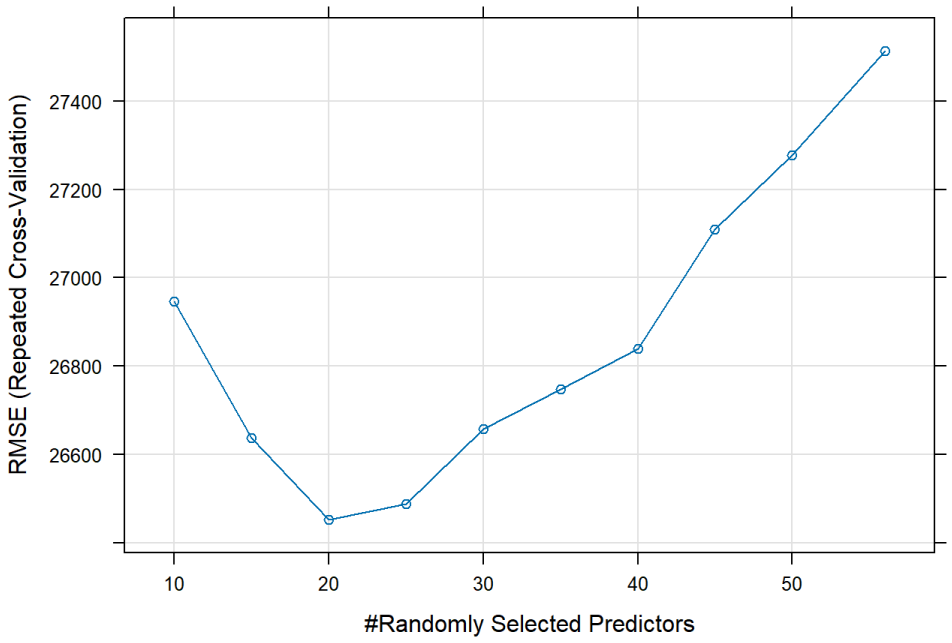


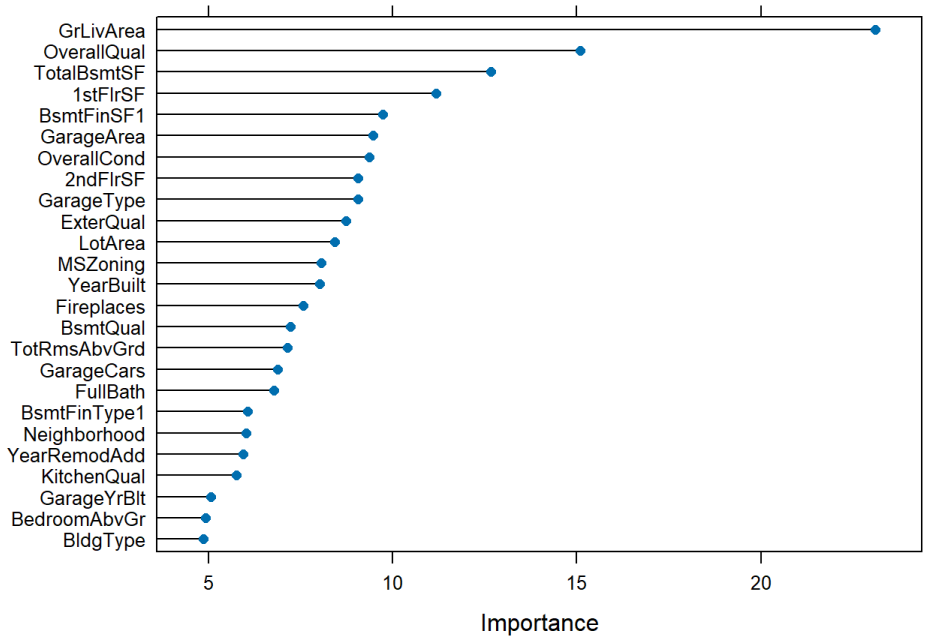

```
##         RMSE     Rsquared          MAE
## 5.436686e+04 6.521157e-01 2.151707e+04
```

Training performance for the OLS model is quite good, but it does not hold up as well against the test set. Still, this sets a baseline for other models to beat.
Next we will fit a Random Forest model. Since Random Forest model is able to handle qualitative predictors, we can use the non-dummy variables this time. This will hopefully reduce some of the predictor noise since there will be fewer models with which to predict.

Test Set Performance:

```
## Random Forest
##
## 1169 samples
##   56 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 1050, 1053, 1052, 1052, 1053, 1052, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##   10    26946.15  0.8993146  17569.39
##   15    26637.27  0.8996650  17418.18
##   20    26451.80  0.8995530  17354.38
##   25    26487.43  0.8980533  17361.80
##   30    26657.41  0.8961114  17405.12
##   35    26748.18  0.8944473  17489.15
##   40    26839.68  0.8930783  17535.74
##   45    27108.49  0.8906860  17707.52
##   50    27276.68  0.8886270  17793.12
##   56    27513.94  0.8856894  17964.96
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 20.
```






```
##         RMSE     Rsquared          MAE
## 4.194982e+04 7.213076e-01 1.872690e+04
```

The Random Forest model looks good, but the performance against the test data suggests there is room for improvement. The Random Forest model does provide us with variable importance, however, and the chart above summarizes these results. We can use these variable importances in our next model, along with the correlation matrix observed earlier to see if we can identify just a few variables of interest. We will then fit an SVM model with a radial basis function using our chosen variables.

The variables that seem reasonable to use moving forward are: OverallQual, GrLivArea, TotalBsmtSF, GarageCars, and FullBath. They don't appear to have issues with missing values, they appeared highly correlated to Sales Price, and there is only slight skewness (which will be addressed) among a few of them.

Variance among these 5 chosen variables was not yet run. Prior to proceeding, It's crucial to check for zero variance in variables before incorporating them into a model because such variables provide no information that can help distinguish between observations in different categories or predict an outcome. Including them can waste computational resources and potentially introduce noise into the model, compromising its accuracy.

Variance of chosen variables:

```
##  OverallQual    GrLivArea  TotalBsmtSF   GarageCars     FullBath
## 1.912679e+00 2.761296e+05 1.924624e+05 5.584797e-01 3.035082e-01
```

The analysis highlights the significance of various predictors in determining housing prices, with no zero variance found among the variables, ensuring their variability contributes meaningfully to the model. ‘GrLivArea’ and ‘TotalBsmtSF’ are the most influential, indicating that larger living areas and basements significantly impact house prices. ‘OverallQual’ is also critical, suggesting that homes of higher quality command higher prices. While ‘GarageCars’ and ‘FullBath’ have smaller impacts, the number of cars a garage holds and the number of full bathrooms still positively influence the price, albeit to a lesser degree.

Data summary

Name	train2[chosen_variables]
Number of rows	1460
Number of columns	5
_____	
Column type frequency:	
numeric	5
_____	
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
OverallQual	0	1	6.10	1.38	1	5.00	6.0	7.00	10	▁▂▇▂▁
GrLivArea	0	1	1515.46	525.48	334	1129.50	1464.0	1776.75	5642	▇▇▁▁▁
TotalBsmtSF	0	1	1057.43	438.71	0	795.75	991.5	1298.25	6110	▅▇▁▁▁
GarageCars	0	1	1.77	0.75	0	1.00	2.0	2.00	4	▁▃▇▁▁
FullBath	0	1	1.57	0.55	0	1.00	2.0	2.00	3	▁▇▁▇▁

```
##   OverallQual       GrLivArea     TotalBsmtSF       GarageCars
##  Min.   : 1.000   Min.   : 334   Min.   :   0.0   Min.   :0.000
##  1st Qu.: 5.000   1st Qu.:1130   1st Qu.: 795.8   1st Qu.:1.000
##  Median : 6.000   Median :1464   Median : 991.5   Median :2.000
##  Mean   : 6.099   Mean   :1515   Mean   :1057.4   Mean   :1.767
##  3rd Qu.: 7.000   3rd Qu.:1777   3rd Qu.:1298.2   3rd Qu.:2.000
##  Max.   :10.000   Max.   :5642   Max.   :6110.0   Max.   :4.000
##     FullBath
##  Min.   :0.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :1.565
##  3rd Qu.:2.000
##  Max.   :3.000
```

It is also essential to address the skewness observed in the ‘GrLivArea’ and ‘TotalBsmtSF’ variables. To rectify this, a Box-Cox transformation will be applied to normalize the distributions. Additionally, any missing values in the data will be replaced with the median value to maintain data integrity and ensure robust model performance.

Here are the transformed datasets, finalized and ready for further analysis.

```
## # A tibble: 6 × 6
##   SalePrice OverallQual GarageCars FullBath GrLivArea_BoxCox TotalBsmtSF_BoxCox
##       <dbl>       <dbl>      <dbl>    <dbl>            <dbl>              <dbl>
## 1    208500           7          2        2             7.44               94.1
## 2    181500           6          2        2             7.14              119.
## 3    223500           7          2        2             7.49               98.4
## 4    140000           7          3        1             7.45               87.2
## 5    250000           8          3        2             7.70              112.
## 6    143000           5          2        1             7.22               90.0
```

Next, we implemented a Support Vector Machine (SVM) model to predict housing prices, recognizing the importance of centering and scaling the data to normalize the feature scales and improve model accuracy. This method was selected due to the uncertain complexity and potential non-linearity of the relationships between the features and the target variable. Properly preparing the data ensures more reliable training outcomes, which is crucial for effectively capturing the underlying patterns and relationships

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 1169 samples
##    5 predictor
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 1052, 1052, 1052, 1053, 1051, 1053, ...
## Resampling results across tuning parameters:
##
##   C     RMSE      Rsquared   MAE
##   0.25  39529.95  0.7721956  22831.54
##   0.50  36276.43  0.8054568  21893.41
##   1.00  34022.17  0.8265475  21389.49
##
## Tuning parameter 'sigma' was held constant at a value of 0.2834467
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were sigma = 0.2834467 and C = 1.
```

Test Set Performance:

```
##         RMSE     Rsquared          MAE
## 30746.717256     0.841221 20888.238215
```

Interpretation of Performance Results for the SVM model:

- The R^2 value of 0.8412 shows our SVM model explains a substantial portion of the variance in house prices.
- The RMSE and MAE values suggest that the model's predictions are reasonably close to the actual values but still have room for improvement. The typical prediction error is in the range of $20,000 to $30,000.We could use some additional model tuning or add additional features to improve the prediction accuracy (such as experimenting with some additional hyper parameter tuning)

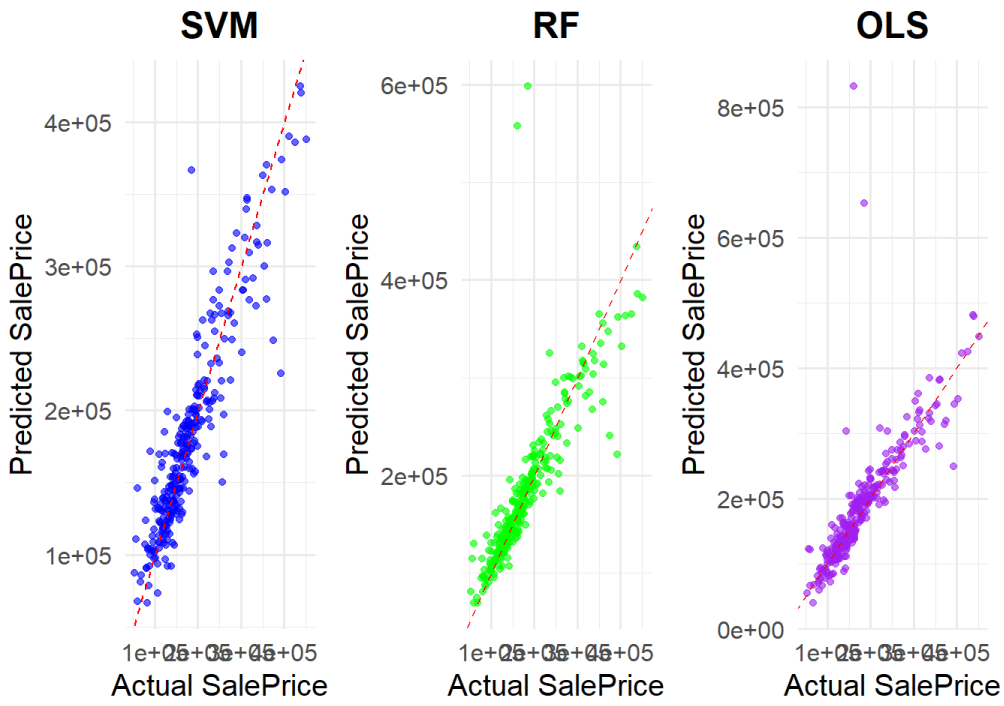
Comparing Test Set performances:

```
##          RMSE  Rsquared      MAE
## OLS  54366.86 0.6521157 21517.07
## RF   41949.82 0.7213076 18726.90
## SVMr 30746.72 0.8412210 20888.24
```

The results from the three modeling approaches—Ordinary Least Squares (OLS), Random Forest (RF), and Support Vector Machine with radial basis function (SVMr)—show varying degrees of predictive accuracy on the housing price data.

The SVMr model provided the highest predictive accuracy with an R2R^2R2 value of 0.841, indicating that it explains about 84.1% of the variance in housing prices, and it achieved the lowest Root Mean Square Error (RMSE) of 30,746.72. Although the SVMr model's Mean Absolute Error (MAE) is relatively close to that of the RF model, its substantially better R2R^2R2 and RMSE scores make it the superior model in terms of overall prediction quality.

The RF model also outperforms the OLS model, with a higher R2R^2R2 value and lower errors, underscoring the effectiveness of more complex algorithms over linear approaches for this dataset



Analysis Results

The analysis process started with a meticulous cleaning of the data, focusing primarily on addressing missing values. For instances where a missing value indicated the absence of a feature (such as a basement or garage), logical imputation was employed. Conversely, for missing values lacking inherent significance, various methods were applied during pre-processing to ensure robustness. The dataset underwent several pre-processing steps, including BoxCox transformations for normalizing distributions, centering and scaling to bring variables to a common scale, removing predictors with near-zero variance to streamline the model, and creating dummy variables necessary for the OLS regression.

The first modeling attempt with Ordinary Least Squares (OLS) served as a baseline, reflecting initial responsiveness of the dependent variable to the predictors. Despite adequate resampling performance, the model exhibited poor generalization to unseen test data, suggesting potential overfitting. To address categorical variables more effectively and gain insights into variable importance, a Random Forest model was subsequently applied. This model yielded improved results across both training and test sets but did not fully optimize the explained variance and error.

A deeper investigation into the importance of variables as suggested by the Random Forest model, alongside correlations identified during data exploration, informed the selection of five key predictors: GrLivArea, OverallQual, TotalBsmtSF, GarageCars, and FullBath. These predictors were chosen for their uniqueness and potential to reduce noise within the model. These variables were then utilized in a Support Vector Machine (SVM) model with a Radial Basis Function kernel, selected for its robustness against outliers. This model demonstrated superior performance and is recommended for its predictive capabilities, significantly outperforming previous models in handling both typical and atypical data points effectively.

In summary, the SVM model emerged as the most effective, recommended for its strong predictive power after thorough examination and optimization of the dataset through various statistical techniques and machine learning models. This iterative approach of refining the model inputs and configurations led to a robust model that leverages the strengths of different statistical techniques to enhance prediction accuracy

Conclusion

The research into predicting housing prices has demonstrated that certain characteristics, notably GrLivArea, OverallQual, TotalBsmtSF, GarageCars, and FullBath, significantly influence the final sale price of homes. These variables were key in driving the performance of an SVM model equipped with a Radial Basis Function, which proved particularly adept at handling data outliers and complexities inherent in housing market data. This model's robustness to outliers ensures that it is capable of delivering reliable predictions across a diverse range of housing attributes.

The study underscores the multifaceted nature of real estate valuation, where both the size and quality of the property play pivotal roles. Moving forward, a deeper tuning of the current models could unlock even higher accuracy. Additionally, incorporating more variables and extending the analysis to include new data could further refine our understanding and prediction of housing prices. The promising results achieved with the SVM model point to the potential of advanced machine learning techniques to supplement traditional real estate appraisal methods, potentially leading to more rapid and cost-effective property assessments.