# Final Project

Akshat Valse

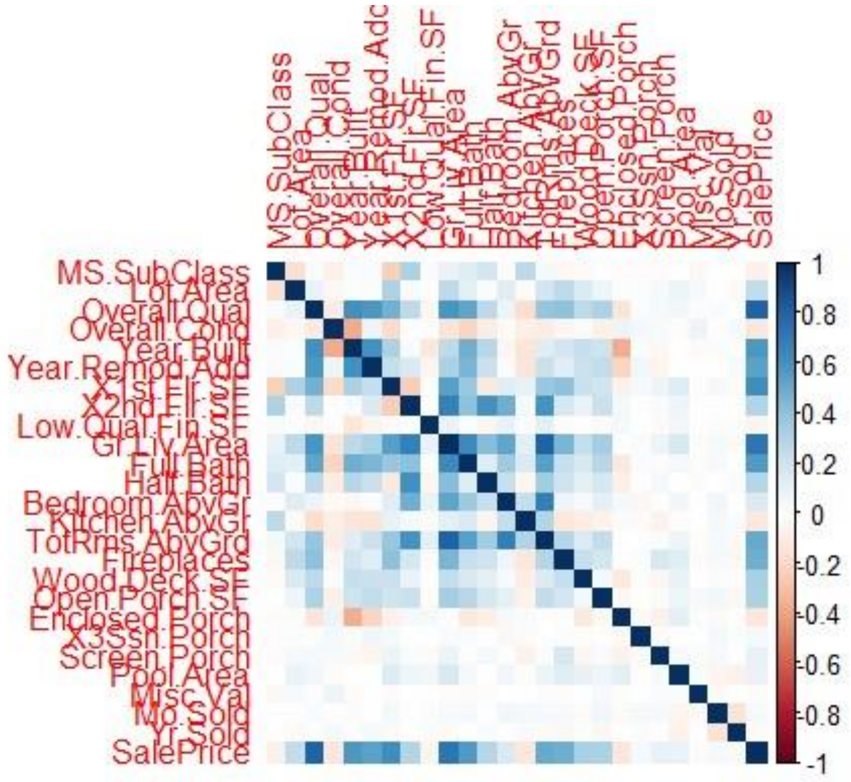
2023-08-15

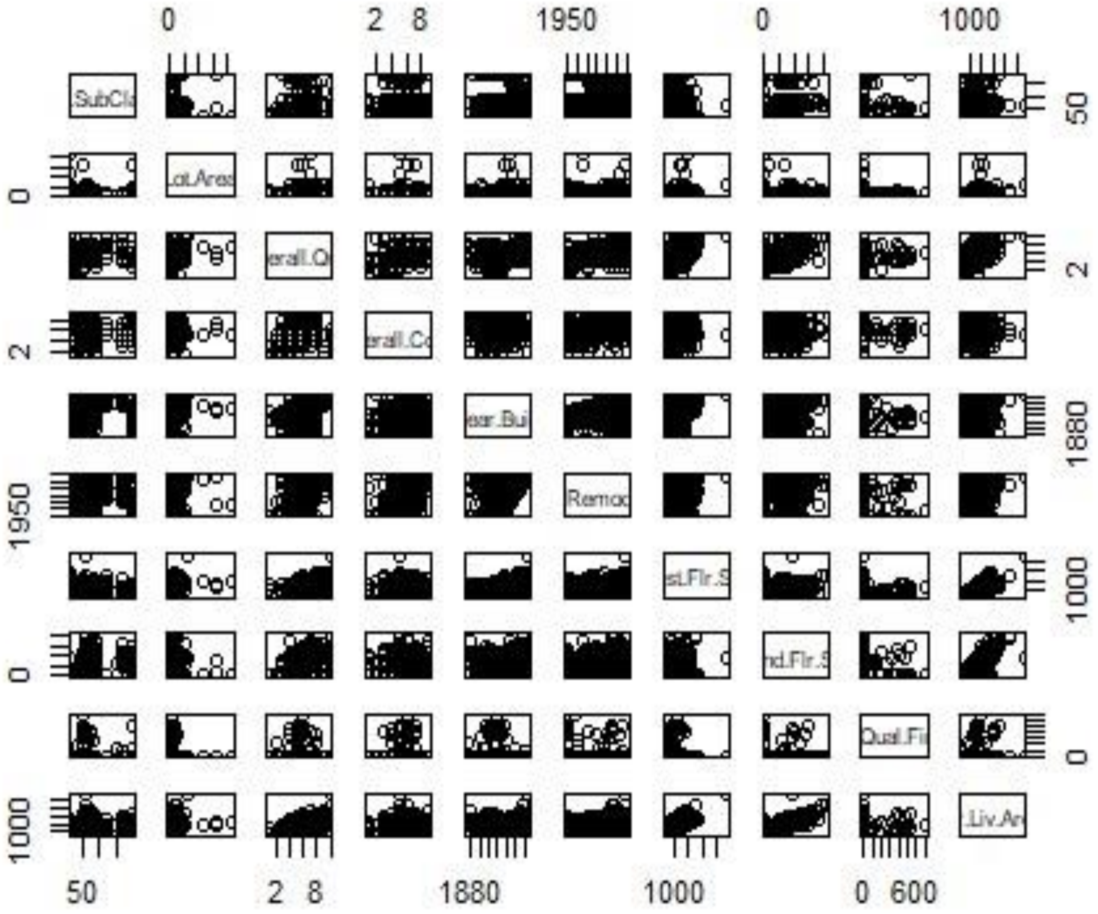
Main Task:

The study - The data comes from the real estate field. Data set contains information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010. I am interesting in how sale price correlates with the all other variables. The goals will be answered using a random forest modeling approach that will be trained on half the data and then testing using the testing data. We will then find the R-squared value for the predictions vs. actual sale price. A high r-squared value would indicate that the model is able to accurately predict sale price based on the other parameters.

This will split the data randomly into a training set and testing set. Ideally I would want to use a 70/30 distribution but the project asks for a 50/50 split. As the total number of rows is 2000, each new split data will be 1000 each. I will also be clearning the data set prior to the split. Overall, this code chunk preprocesses the data by factorizing categorical columns, replacing "None" values with NA, defining and applying functions for replacing NAs based on column type, and splitting the data into training and testing sets.

EXPLORATORY ANALYSIS QUESTIONS –

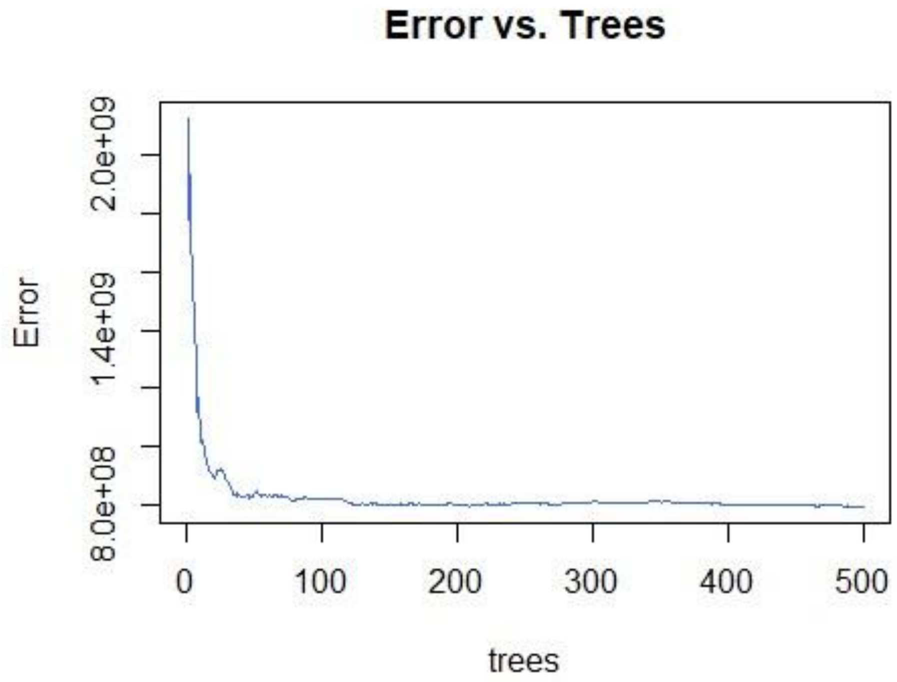


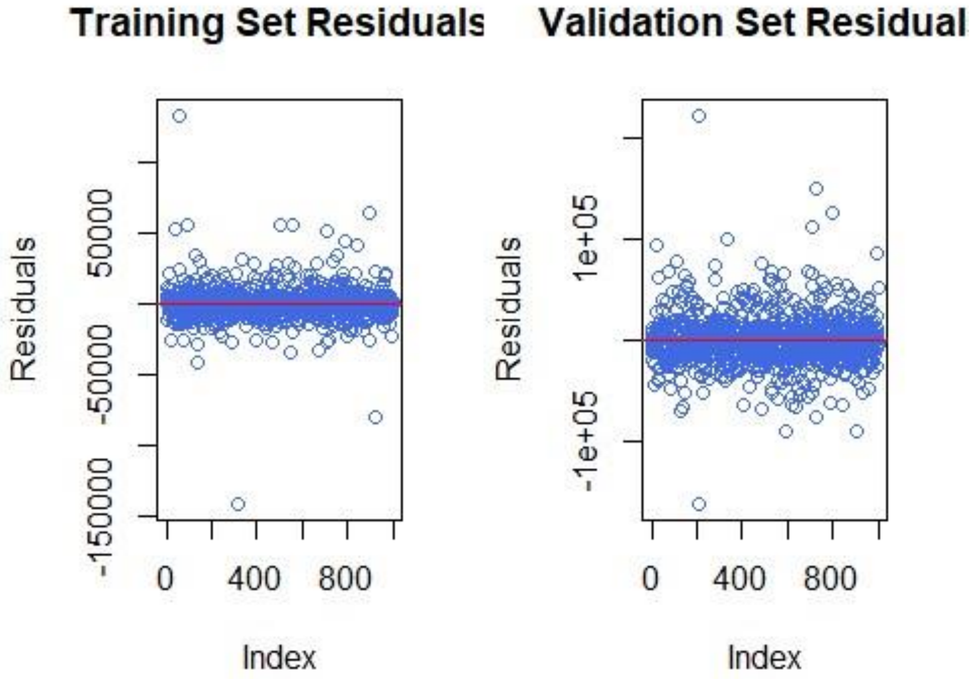


- How are the predictor variables spread out? Are there any noteworthy features to their spread that could be highly influential observations? The zoning of most of the data set is low density residential and as a result our model would best predict houses categorized as such. A lot of categorical variables have a overwhelming majority of a given category which could induce some bias in the model. As for continuous variables, lot area, year built, and garage living area are the three most positively correlated variables with Sale price, therefore those variables may have a greater importance in determining the sale price of a house.

- Are any of the predictor variables highly correlated? Other than the ones mentioned in the last question, we also have correlation in between variables as seen in the correlation matrix in the center top left section of the graph, but some other variables with a high correlation with sale price would include 1st floor square feet and total rooms above ground.

MODELING QUESTIONS –





- Which predictor variables, if any, should be included in the model a priori? I would make sure to include the continuous variables with a high correlation with sale price that we mentioned earlier such as lot area, garage living area, and year built. It is reasonable to assume that they affect sale price as buyers are looking for these items before making a purchase.

- Are there any interactions that should be considered for inclusion in the model? I also considered the interactions between variables in the random forest modeling method. As the approach combines the use of multiple decision trees and as I split the data randomly, the training set would also mimic the rest of the data and along with it the relationships in between variables and how those affect sale price.

- Are there any three way interactions that should be considered? As choosing a sale price of a house requires multiple variables to be considered, I will be considering all possible 3 way interactions in the variables in this approach.

- Are there any interactions that should NOT be considered? It would have little to no effect removing data that contains a lot of missing values or values that are all the same as it does not have an impact in this data set. Doing so however could induce selection bias into the model making the random forest. I would considering doing so for some variables in this data set to save computation time but as that is not a factor I deem important, I will be using the entire data set to train and test my model for prediction.

RESULTS QUESTIONS -

- What is the final regression model for the data? As a random forest is an ensemble of decision trees. Each tree contributes to the overall prediction, and there isn’t a single equation that represents the relationship between the predictors and the target variable. Instead, the random forest algorithm combines the predictions of multiple decision trees to make accurate predictions. Each tree makes its own individual prediction, and the final prediction is often an average (for regression) or a majority vote (for classification) of the individual tree predictions. So, in the context of a random forest model, the “final regression model” is the ensemble of decision trees that collectively make predictions based on the input features.

- Using the standard diagnostic tests, does the model appear to fit the data well? R-squared Value: An R squared value of 0.9 indicates that approximately 90% of the variability in the dependent variable (actual sale price) is explained by the predictor variables included in my model. This is a good indication that my model captures a significant portion of the variation in the target variable. Correlation (r) Value: A correlation value of 0.95 between the predicted and actual values suggests a strong positive linear relationship between my predictions and the actual sale prices. A value close to 1 indicates that my predictions are closely aligned with the actual values. Absolute Mean Error: An absolute mean error of 16425 means that, on average, my predictions deviate from the actual sale prices by approximately $16425. This metric provides insight into the average magnitude of prediction errors. Mean Squared Error (MSE): A mean squared error of 603373819 indicates the average squared difference between your predicted and actual sale prices. Lower MSE values indicate better model performance. Overall, the high R-squared, strong correlation, and relatively low absolute mean error and mean squared error values suggest that my random forest model is performing well in fitting the data and making accurate predictions of sale prices.

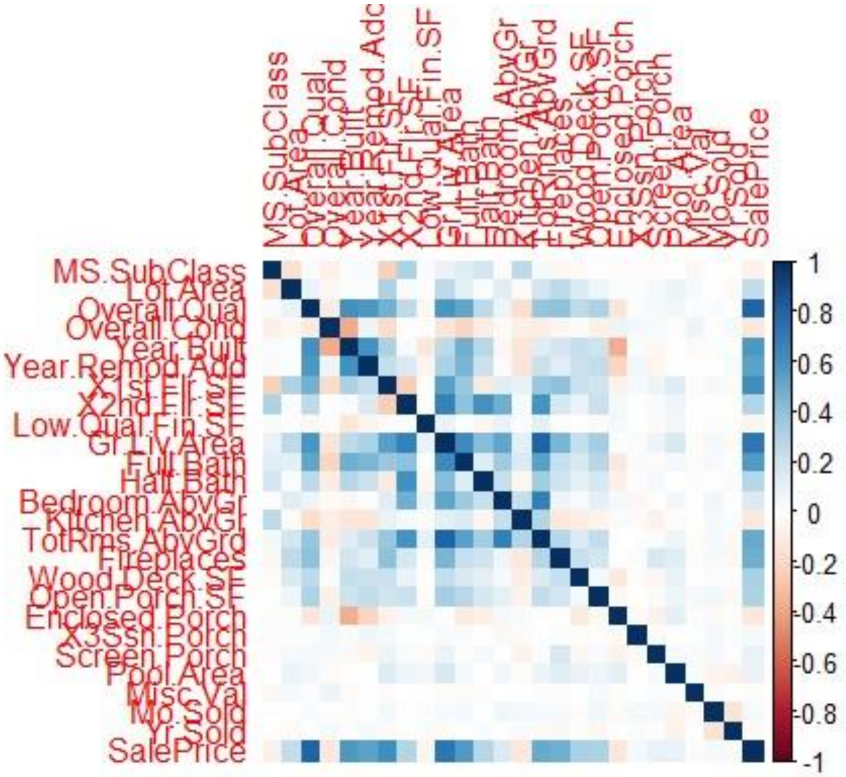
- What is your estimated prediction accuracy for your model? (Evaluated on the validation set). The median error is approximately 16k dollars and the root mean squared error is 24k dollars. This is pretty accurate for the sale price variables as seen with the high R and R squared values.

- Compare the intervals constructed using your final selected model fit to the validation set to the same intervals constructed on the training set. Are they very different? Which do you believe more? As we can see in the graph the testing and training date prediction fits are approximately the same which means we have successfully avoided over fitting to testing data.

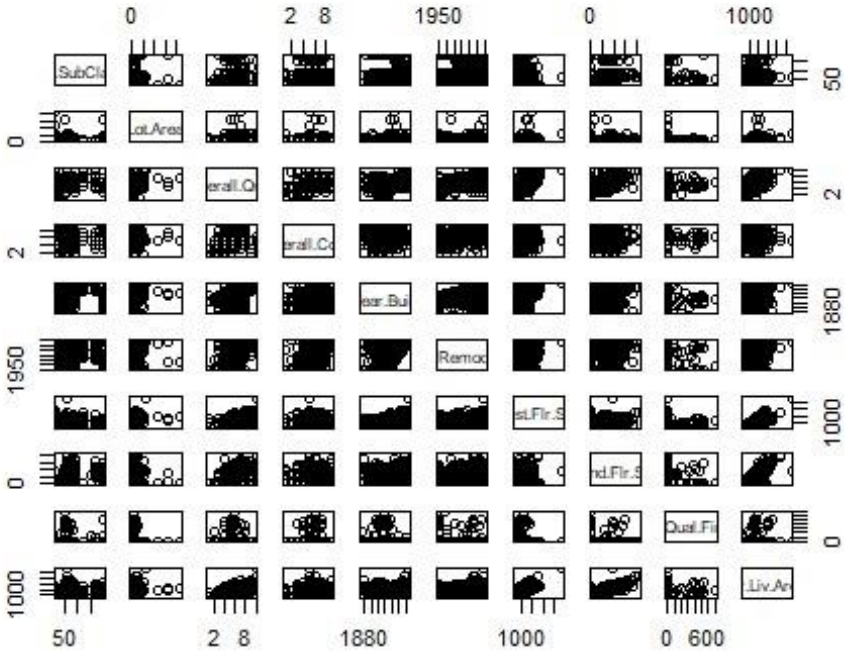
Appendix -

```
# Set seed to ensure
reproducibility set.seed(123);
rnorm(1) ## [1] -0.5604756

# Load required libraries library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine library(caret)

## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'
## The following object is masked from 'package:randomForest': ##
##     margin

## Loading required package:

lattice library(corrplot) ## corrplot

0.92 loaded

library(ggplot2)
library(ggmosaic)

### PREPROCESSING STEP ###
### ALL STUDY QUESTIONS WERE ANSWERED HERE ###

# Create a data set using the CSV file given data <-
read.csv("C:/Users/aksha/Downloads/ames2000_NAfix.csv")

# Preprocess the dataset into a data frame to make it easier to work with
cdata <- data.frame(data)

# Factorize the categorical data
cdata[sapply(cdata, is.character)] <- lapply(cdata[sapply(cdata,
is.character)], as.factor)

# Define a function to calculate the mode of a
vector calculate_mode <- function(x) {   uniq_x <-
unique(x)
  uniq_x[which.max(tabulate(match(x, uniq_x)))]
}

# Define a function to replace NA with desired values based on column
type replace_na_by_type <- function(column) {   if (is.numeric(column)) {
    return(ifelse(is.na(column), mean(column, na.rm = TRUE), column))
  } else if (is.integer(column)) {
    return(ifelse(is.na(column), median(column, na.rm = TRUE), column))
  } else {
    mode_val <- calculate_mode(column)
    return(ifelse(is.na(column), mode_val, column))
}
}

# Apply the function to each column in the cdata data
frame cdata_filled <- cdata %>%
mutate_all(replace_na_by_type)

# Split into training and testing data (1000 data points each) shuffled_data
<- cdata_filled %>% sample_n(size = nrow(.))
```

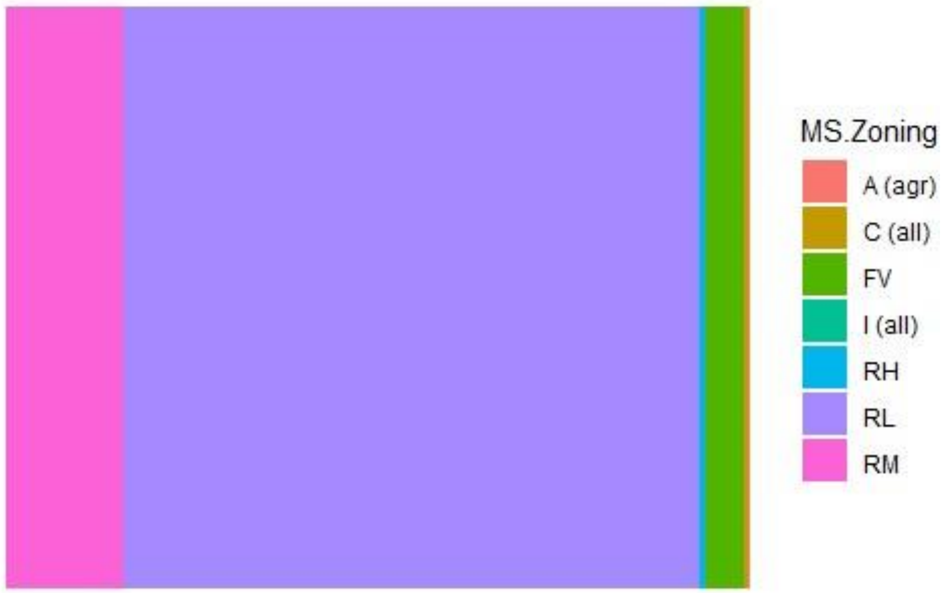
```
split_size <- 1000
train_data <- shuffled_data[1:split_size, ]
test_data <- shuffled_data[(split_size + 1):(split_size * 2), ]

### EXPLORATORY ANALYSIS ### ### ALL EXPLORATORY ANALYSIS QUESTIONS WERE
ANSWERED USING THESE RESULTS ###

# Create a correlation matrix
corrplot(cor(cdata[sapply(cdata, is.numeric)]), method = "color")
```



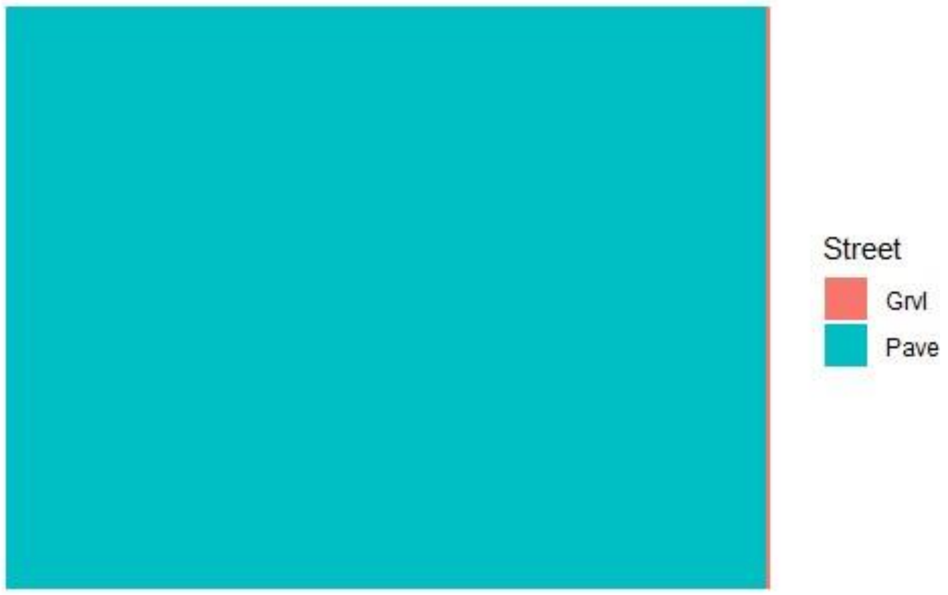
```
# Plot the first 10 numerical variables pairsdata
<- cdata[sapply(cdata, is.numeric)]
pairs(pairsdata[c(1,2,3,4,5,6,7,8,9,10)])
```

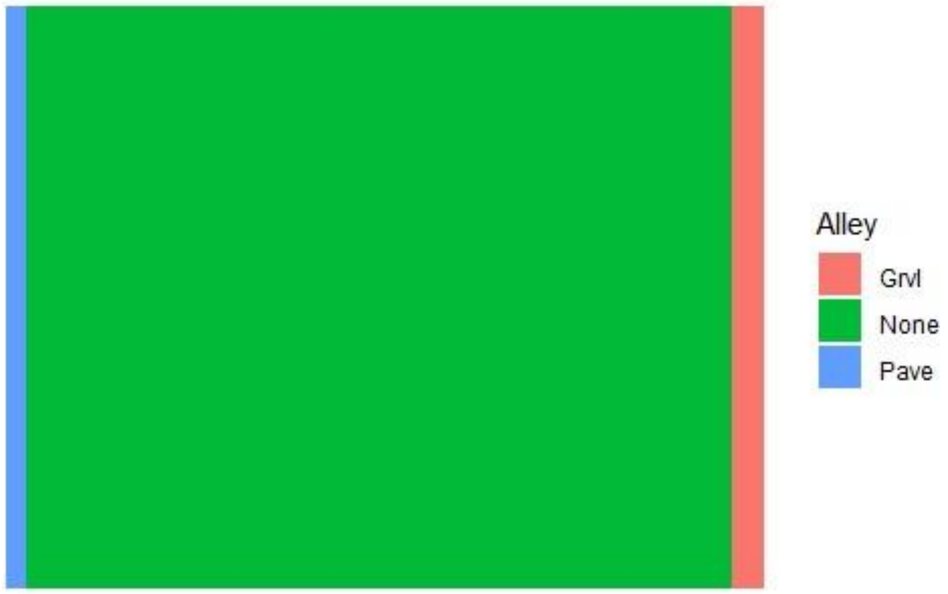


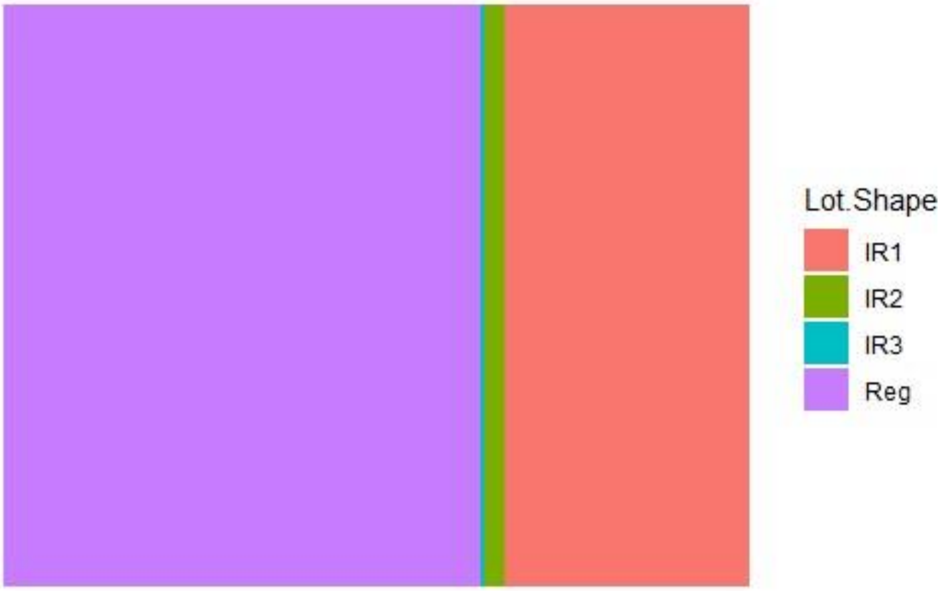
```
# Subset the data frame to include the specified columns
categorical_vars <- colnames(cdata)[sapply(cdata, is.factor)][1:10]
selected_data <- cdata[, categorical_vars]

# Create mosaic plots for each categorical variable for
(var in categorical_vars) {
  mosaic_plot <- ggplot(selected_data, aes(x = "", fill = !!as.symbol(var)))
+
    geom_bar(position = "fill") +
labs(fill = var) +
theme_void() +     coord_flip()

print(mosaic_plot)
}
```

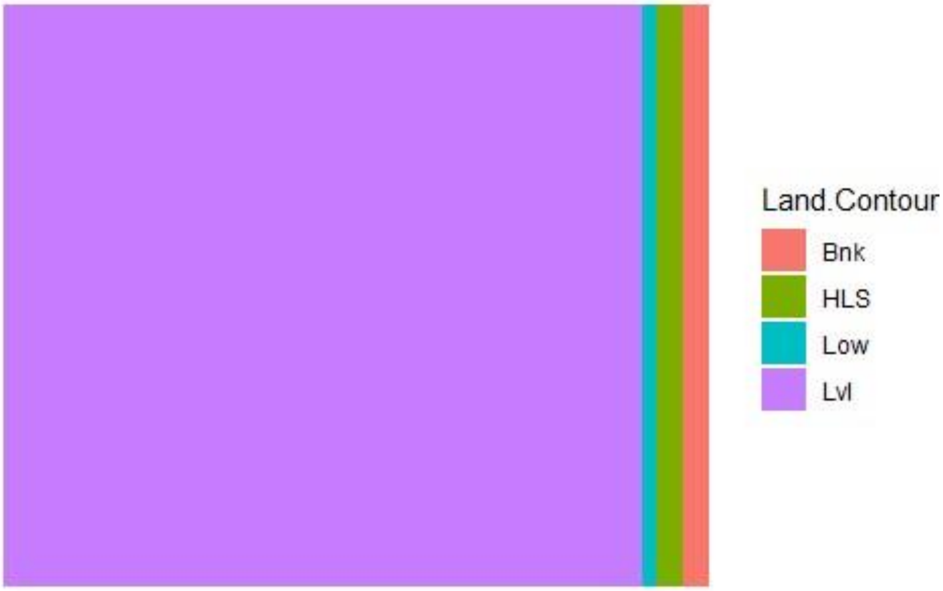

MS.Zoning
A (agr)
C (all)
FV
I (all)
RH
RL
RM

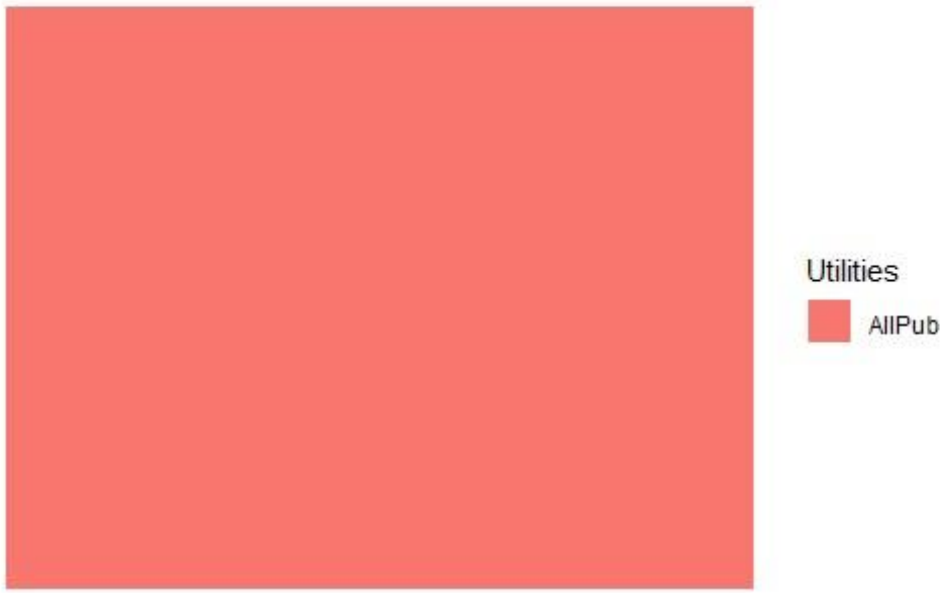
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
118
120
121
124
125
126
128
129
130
133
134
135
136
138
140
141
149
152
153
155
160
168
174
182
195
21
22
24
25
26
28
30
313
32
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
88
89
90
91
92
93
94
95
96
97
98
99
None

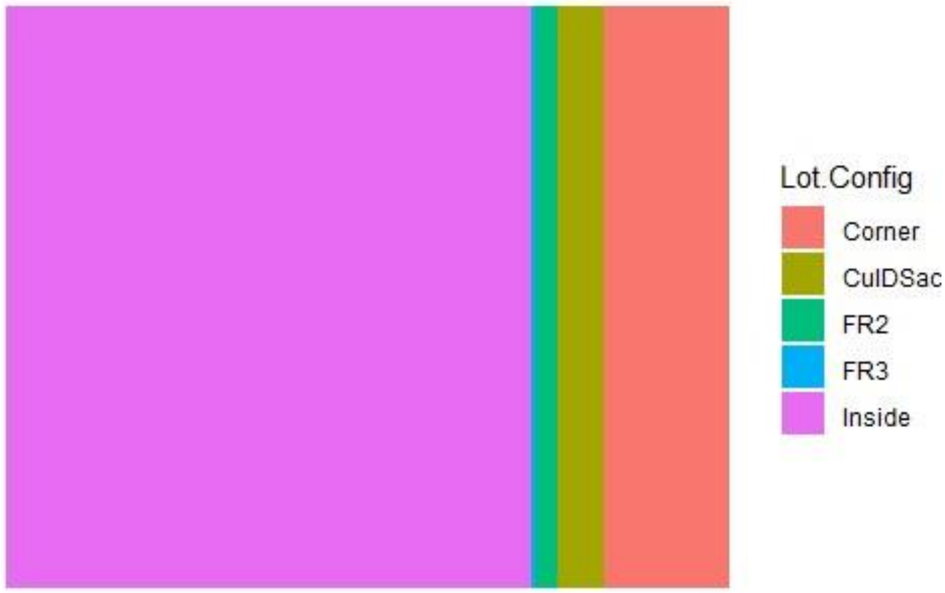

Street
Grvl
Pave

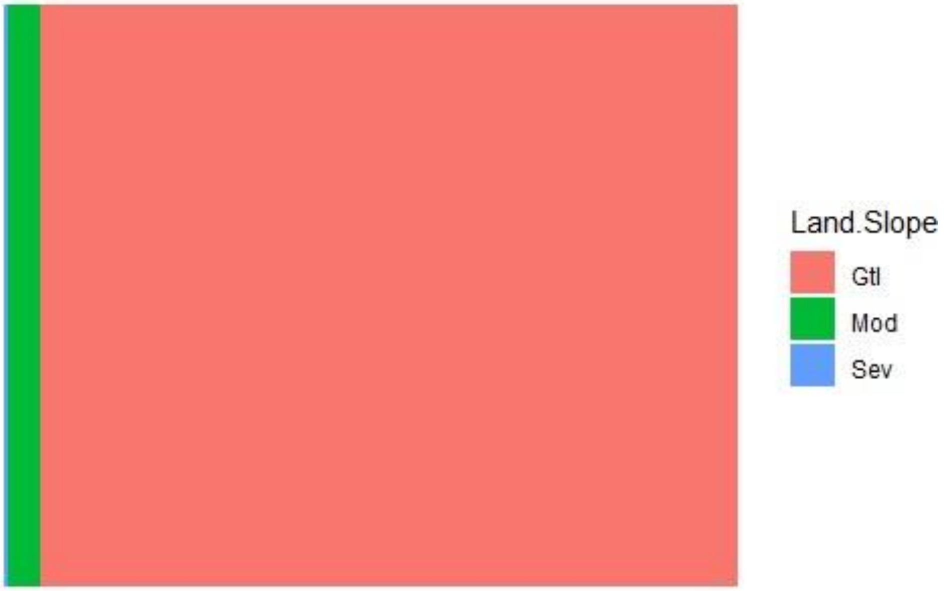

Alley
Grvl
None
Pave

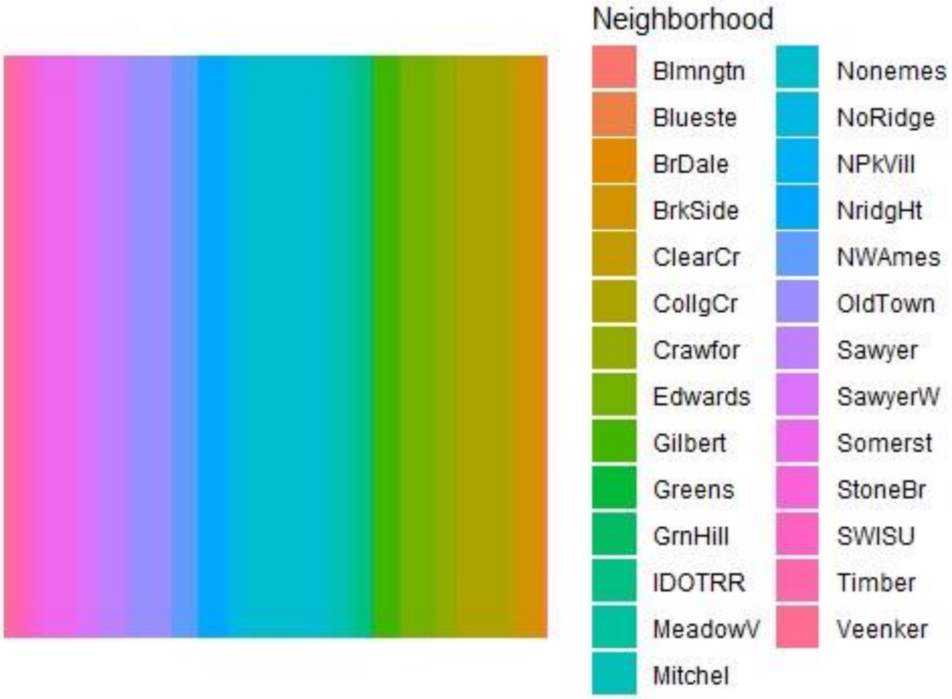

Lot.Shape
IR1
IR2
IR3
Reg


Land.Contour
Bnk
HLS
Low
Lvl


Utilities
AllPub


Lot.Config
Corner
CulDSac
FR2
FR3
Inside

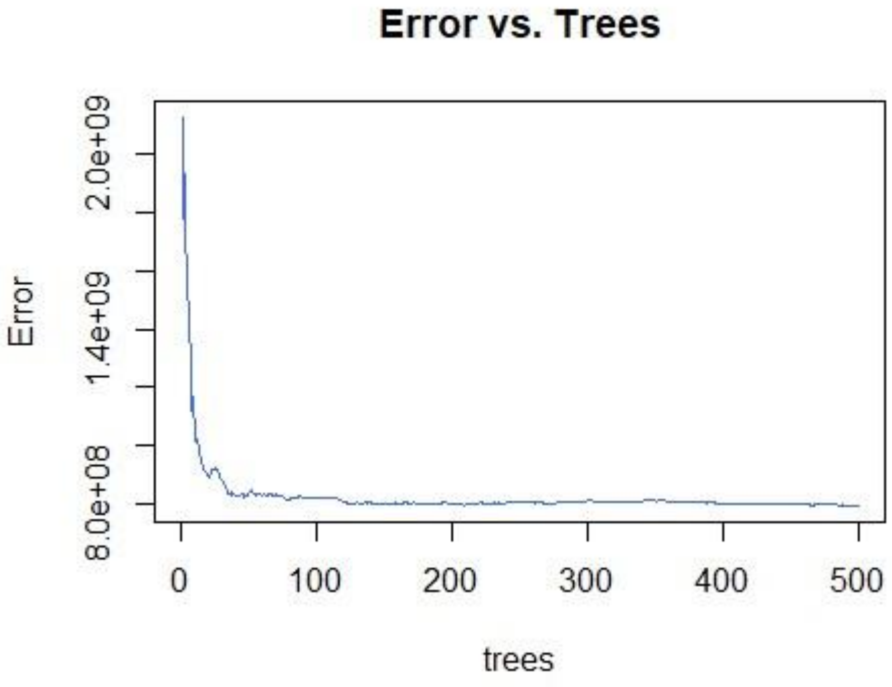

Land.Slope
Gtl
Mod
Sev


Neighborhood
Blmngtn
Blueste
BrDale
BrkSide
ClearCr
CollgCr
Crawfor
Edwards
Gilbert
Greens
GrnHill
IDOTRR
MeadowV
Mitchel
Nonemes
NoRidge
NPkVill
NridgHt
NWAmes
OldTown
Sawyer
SawyerW
Somerst
StoneBr
SWISU
Timber
Veenker

```
### MODELING STEP ###
### ALL MODELING QUESTIONS WERE ANSWERED USING THIS CODE ###

# Train the random forest model rf_model <-
randomForest(SalePrice ~ ., data = train_data)

# Make predictions on the testing set predictions <-
predict(rf_model, newdata = test_data)

### RESULTS STEP ###
### ALL QUESTIONS ABOUT THE RESULTS OF THE MODELING AND MODELING ACCURACY
WERE ANSWERED USING INFORMATION IN THE FOLLOW CODE ###

# Diagnostic Tests
# Calculate the root mean squared error (RMSE)
rmse <- sqrt(mean((predictions - test_data$SalePrice)^2)) print(paste("Root
Mean Squared Error (RMSE):", rmse))
```

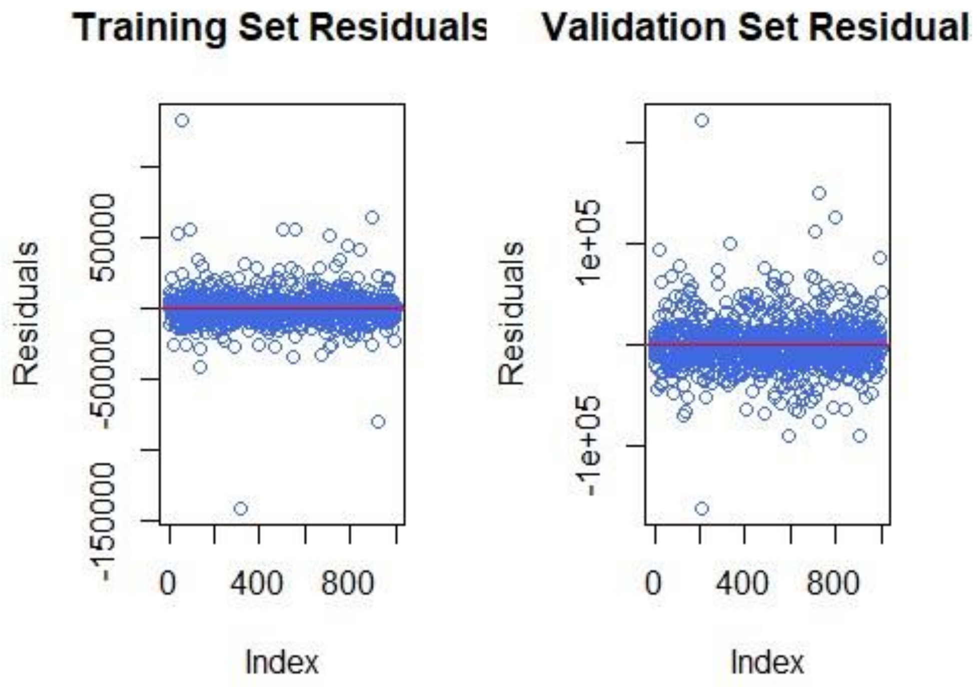
```
## [1] "Root Mean Squared Error (RMSE): 24755.2931695201"
```

```
# Plot error vs number of trees used by the model
plot(rf_model, col = "royalblue", main = "Error vs. Trees")
```



```
# Model Comparison predictions <- predict(rf_model, newdata = test_data)

# Calculate mean absolute error and mean squared error
```

```
MAE <- mean(abs(predictions - test_data$SalePrice))
MSE <- mean((predictions - test_data$SalePrice)^2)
RMSE <- sqrt(mean((predictions - test_data$SalePrice)^2))

# Print MAE and MSE cat("Mean Absolute Error
(MAE):", MAE, "\n") ## Mean Absolute Error (MAE):
16494.55 cat("Mean Squared Error (MSE):", MSE,
"\n") ## Mean Squared Error (MSE): 612824540
cat("Root Mean Squared Error (RMSE):", RMSE, "\n")
## Root Mean Squared Error (RMSE): 24755.29

# Prediction Accuracy
R <- cor(predictions, test_data$SalePrice)
R_squared <- cor(predictions, test_data$SalePrice)^2

# Print R and R-squared cat("R:",
R, "\n") ## R: 0.9506468 cat("R-
squared:", R_squared, "\n")
## R-squared: 0.9037293

# Predict on both training and validation sets train_pred
<- predict(rf_model, newdata = train_data) test_pred <-
predict(rf_model, newdata = test_data)

# Calculate residuals for both sets
train_residuals <- train_data$SalePrice - train_pred test_residuals
<- test_data$SalePrice - test_pred

# Plot the residuals for both sets par(mfrow
= c(1, 2))
plot(train_residuals, main = "Training Set Residuals", ylab = "Residuals",
col = "royalblue") abline(h = 0, col = "red")
plot(test_residuals, main = "Validation Set Residuals", ylab = "Residuals",
col = 'royalblue')



abline(h = 0, col = "red")
```



Acknowledgements –

Stackoverflow - https://stats.stackexchange.com/questions/60157/how-to-get-theconfidence-interval-around-the-variable-importance-generated-by-r

Stackoverflow - https://stackoverflow.com/questions/20637360/convert-all-data-framecharacter-columns-to-factors

Random forest - https://www.projectpro.io/recipes/perform-random-forest-r

GeeksforGeeks - https://www.geeksforgeeks.org/how-to-calculate-the-mode-of-all-rowsor-columns-from-a-dataframe-in-r/

Data - http://web.stanford.edu/class/stats191/data/amesdoc.txt

Data - https://nsamrao.github.io/Boston_Housing/