Kaggle_Home_Prices_Ames_Iowa

```
getwd()
```

```
## [1] "/Users/kevalshah/Google Drive/Research/Kaggle House_Prices"
```

```
train <- read.csv("train.csv", header = TRUE, sep = ",")
test <- read.csv("test.csv", header = TRUE, sep = ",")

# Add sale price new column in test dataset
test["SalePrice"] <- NA

# Let's explore the structure of the data
dim(train)
```

```
## [1] 1460   81
```

```
str(train)
```

```
## 'data.frame':    1460 obs. of  81 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning     : Factor w/ 5 levels "C (all)","FV",..: 4 4 4 4 4 4 4 4 5 4 ...
##  $ LotFrontage  : int  65 80 68 60 84 85 75 NA 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
##  $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Alley        : Factor w/ 2 levels "Grvl","Pave": NA NA NA NA NA NA NA NA NA NA ...
##  $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",..: 4 4 1 1 1 1 4 1 4 4 ...
##  $ LandContour  : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Utilities    : Factor w/ 2 levels "AllPub","NoSeWa": 1 1 1 1 1 1 1 1 1 1 ...
##  $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5 1 5 1 ...
##  $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",..: 6 25 6 7 14 12 21 17 18
4 ...
##  $ Condition1   : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 5 1 1 ...
##  $ Condition2   : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 1 ...
##  $ BldgType     : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 1 1 1 2 ...
##  $ HouseStyle   : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 6 3 6 6 6 1 3 6 1 2 ...
##  $ OverallQual  : int  7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond  : int  5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
##  $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
##  $ RoofStyle    : Factor w/ 6 levels "Flat","Gable",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ RoofMatl     : Factor w/ 8 levels "ClyTile","CompShg",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Exterior1st  : Factor w/ 15 levels "AsbShng","AsphShn",..: 13 9 13 14 13 13 13 7 4
9 ...
##  $ Exterior2nd  : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 16 14 14 14 7 1
6 9 ...
##  $ MasVnrType   : Factor w/ 4 levels "BrkCmn","BrkFace",..: 2 3 2 3 2 3 4 4 3 3 ...
##  $ MasVnrArea   : int  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 4 3 4 3 4 4 4 ...
##  $ ExterCond    : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ Foundation   : Factor w/ 6 levels "BrkTil","CBlock",..: 3 2 3 1 3 6 3 2 1 1 ...
##  $ BsmtQual     : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 3 3 4 3 3 1 3 4 4 ...
##  $ BsmtCond     : Factor w/ 4 levels "Fa","Gd","Po",..: 4 4 4 2 4 4 4 4 4 4 ...
##  $ BsmtExposure : Factor w/ 4 levels "Av","Gd","Mn",..: 4 2 3 4 1 4 1 3 4 4 ...
##  $ BsmtFinType1 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 3 1 3 1 3 3 3 1 6 3 ...
##  $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
##  $ BsmtFinType2 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 6 6 6 6 6 6 6 2 6 6 ...
##  $ BsmtFinSF2   : int  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : int  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
##  $ Heating      : Factor w/ 6 levels "Floor","GasA",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ HeatingQC    : Factor w/ 5 levels "Ex","Fa","Gd",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ CentralAir   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical   : Factor w/ 5 levels "FuseA","FuseF",..: 5 5 5 5 5 5 5 5 2 5 ...
##  $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
##  $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
##  $ BsmtFullBath : int  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : int  0 1 0 0 0 0 0 0 0 0 ...
```

```
##  $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath     : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual  : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 3 3 4 3 4 4 4 ...
##  $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : Factor w/ 7 levels "Maj1","Maj2",..: 7 7 7 7 7 7 7 7 3 7 ...
##  $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu  : Factor w/ 5 levels "Ex","Fa","Gd",..: NA 5 5 3 5 NA 3 5 5 5 ...
##  $ GarageType   : Factor w/ 6 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2 6 2 ...
##  $ GarageYrBlt  : int  2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...
##  $ GarageFinish : Factor w/ 3 levels "Fin","RFn","Unf": 2 2 2 3 2 3 2 2 3 2 ...
##  $ GarageCars   : int  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : int  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 2 3 ...
##  $ GarageCond   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ PavedDrive   : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
##  $ WoodDeckSF   : int  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF  : int  61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch: int  0 0 0 272 0 0 0 228 205 0 ...
##  $ X3SsnPorch   : int  0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : Factor w/ 3 levels "Ex","Fa","Gd": NA NA NA NA NA NA NA NA NA NA
 ...
##  $ Fence        : Factor w/ 4 levels "GdPrv","GdWo",..: NA NA NA NA NA 3 NA NA NA NA
 ...
##  $ MiscFeature  : Factor w/ 4 levels "Gar2","Othr",..: NA NA NA NA NA 3 NA 3 NA NA
 ...
##  $ MiscVal      : int  0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold       : int  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold       : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
##  $ SaleType     : Factor w/ 9 levels "COD","Con","ConLD",..: 9 9 9 9 9 9 9 9 9 9 ...
##  $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 1 5 5 5 5 1 5 ...
##  $ SalePrice    : int  208500 181500 223500 140000 250000 143000 307000 200000 129900
118000 ...
```

The categorical variables are stored as factors in our dataframe.

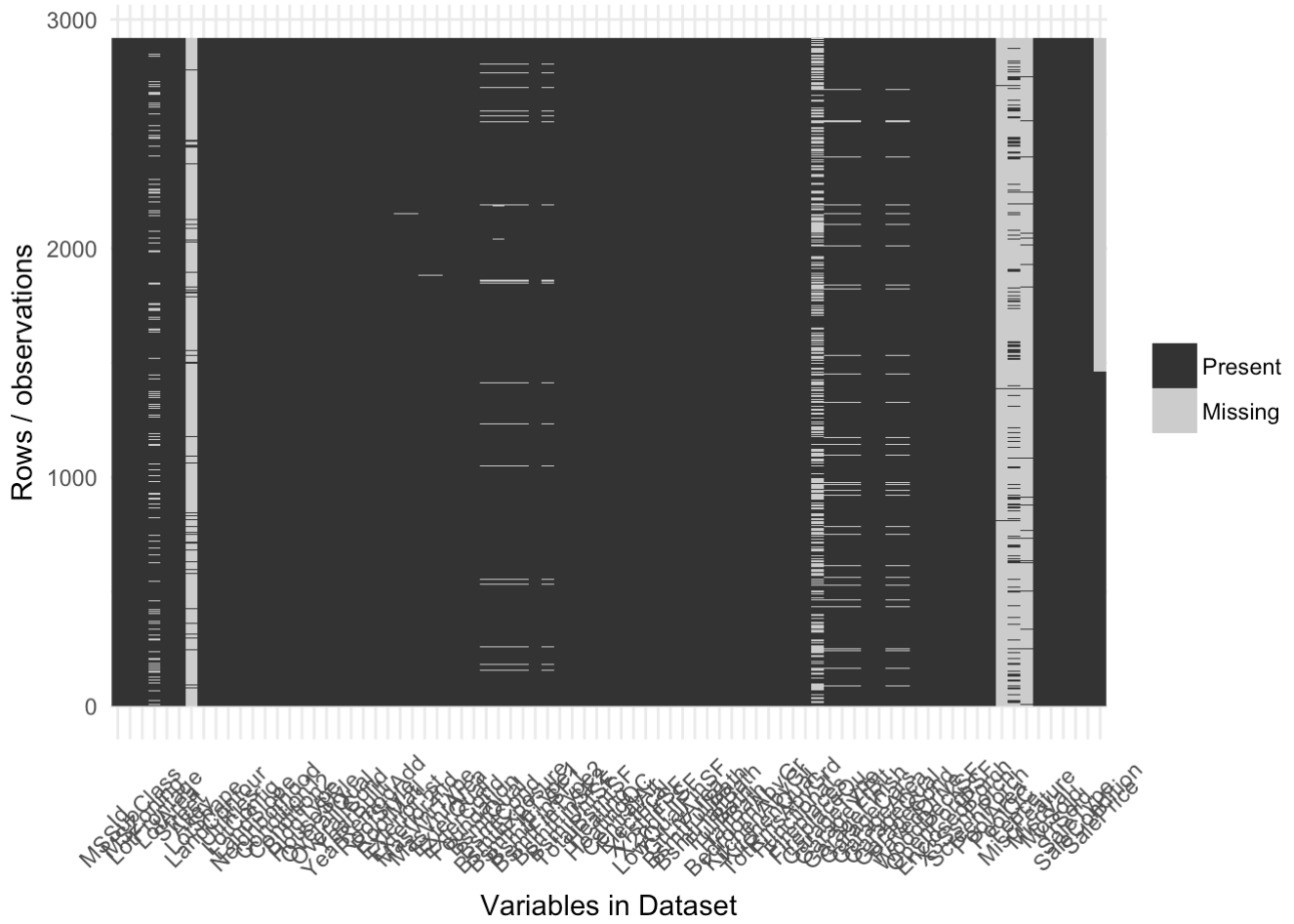
```
# Combine Train and Test datasets
total <- rbind(train, test)

# Visualize missing data using ggplot and a function from neato package in R

library(reshape2)
library(ggplot2)
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
```

```
ggplot_missing <- function(x){

   x %>%
    is.na %>%
    melt %>%
    ggplot(data = .,
           aes(x = Var2,
               y = Var1)) +
    geom_raster(aes(fill = value)) +
    scale_fill_grey(name = "",
                    labels = c("Present","Missing")) +
    theme_minimal() +
    theme(axis.text.x  = element_text(angle=45, vjust=0.5)) +
    labs(x = "Variables in Dataset",
         y = "Rows / observations")
}

ggplot_missing(total)
```




```
# Check for missing values
missing <- colSums(sapply(total, is.na))
missing
```

```
##           Id    MSSubClass      MSZoning   LotFrontage       LotArea
##            0             0             4           486             0
##       Street         Alley      LotShape   LandContour     Utilities
##            0          2721             0             0             2
##    LotConfig     LandSlope  Neighborhood    Condition1    Condition2
##            0             0             0             0             0
##     BldgType    HouseStyle   OverallQual   OverallCond     YearBuilt
##            0             0             0             0             0
## YearRemodAdd     RoofStyle      RoofMatl   Exterior1st   Exterior2nd
##            0             0             0             1             1
##   MasVnrType    MasVnrArea     ExterQual     ExterCond    Foundation
##           24            23             0             0             0
##     BsmtQual      BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1
##           81            82            82            79             1
## BsmtFinType2    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating
##           80             1             1             1             0
##    HeatingQC    CentralAir    Electrical     X1stFlrSF     X2ndFlrSF
##            0             0             1             0             0
## LowQualFinSF     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath
##            0             0             2             2             0
##     HalfBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##            0             0             0             1             0
##   Functional    Fireplaces   FireplaceQu    GarageType   GarageYrBlt
##            2             0          1420           157           159
## GarageFinish    GarageCars    GarageArea    GarageQual    GarageCond
##          159             1             1           159           159
##   PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch    X3SsnPorch
##            0             0             0             0             0
##  ScreenPorch      PoolArea        PoolQC         Fence   MiscFeature
##            0             0          2909          2348          2814
##      MiscVal        MoSold        YrSold      SaleType SaleCondition
##            0             0             0             1             0
##    SalePrice
##         1459
```

```
                                    Data Cleaning Plan
```

Let's look at each missing variables.

LotFrontage: 486 values missing. Linear feet of the street connected to property. Lot frontage, ideally, should correlate with Lot Area. Also, check the lot shape and configuration of missing values.

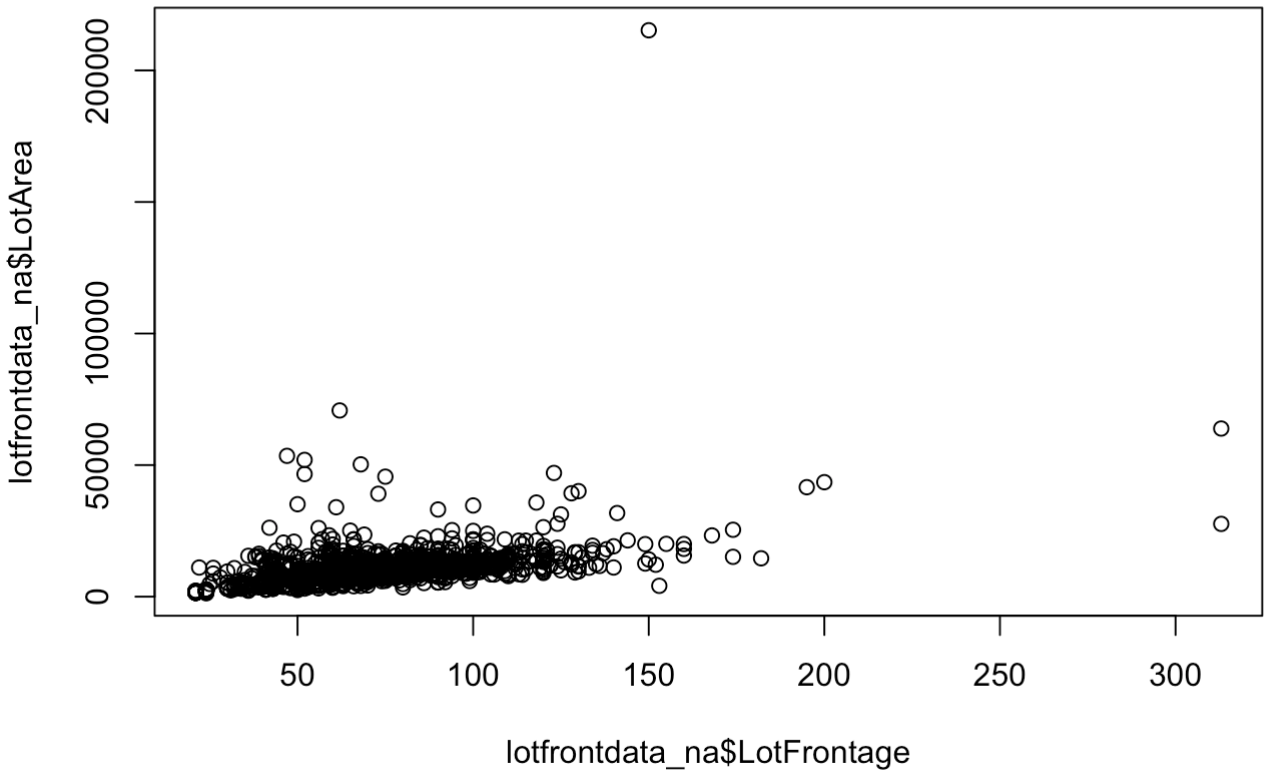
```
lotfront <- c("Id","LotFrontage","LotArea","LotShape","LotConfig")
lotfrontdata <- total[lotfront]
lotfrontdataNA <- lotfrontdata[is.na(lotfrontdata$LotFrontage),]
str(lotfrontdataNA)
```

```
## 'data.frame':    486 obs. of  5 variables:
##  $ Id         : int  8 13 15 17 25 32 43 44 51 65 ...
##  $ LotFrontage: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ LotArea    : int  10382 12968 10920 11241 8246 8544 9180 9200 13869 9375 ...
##  $ LotShape   : Factor w/ 4 levels "IR1","IR2","IR3",..: 1 2 1 1 1 1 1 1 2 4 ...
##  $ LotConfig  : Factor w/ 5 levels "Corner","CulDSac",..: 1 5 1 2 5 2 2 2 1 5 ...
```

```
#hist(lotfrontdataNA[c("Id","LotArea","LotShape","LotConfig")])
summary(lotfrontdataNA)
```

```
##        Id          LotFrontage     LotArea       LotShape    LotConfig
##  Min.   :   8.0   Min.   : NA   Min.   :  1533   IR1:321   Corner :104
##  1st Qu.: 721.2   1st Qu.: NA   1st Qu.:  8125   IR2: 28   CulDSac: 87
##  Median :1364.5   Median : NA   Median : 10452   IR3:  5   FR2    : 20
##  Mean   :1417.2   Mean   :NaN   Mean   : 12380   Reg:132   FR3    :  4
##  3rd Qu.:2142.8   3rd Qu.: NA   3rd Qu.: 12928             Inside :271
##  Max.   :2909.0   Max.   : NA   Max.   :164660
##                   NA's   :486
```

```
lotfrontdata_na <- na.omit(lotfrontdata)

plot(lotfrontdata_na$LotFrontage, lotfrontdata_na$LotArea)
```



```
# We take square root of LotArea to compute correlation with LotFrontage
cor(lotfrontdata_na$LotFrontage, sqrt(lotfrontdata_na$LotArea))
```

```
## [1] 0.647658
```

We see a slightly stronger correlation with Sq. root of Lot Area. However, the correlation is not very strong. We will substitute NAs for LotFrontage with mean value.

```
total$LotFrontage[is.na(total$LotFrontage)] <- round(mean(total$LotFrontage, na.rm = TRU
E))
```

```
                            Categorical Missing Variables.
```

Some homes / properties do not have alley access.

```
total$Alley <- as.character(total$Alley)
total$Alley[is.na(total$Alley)] <- 'None'
total$Alley <- as.factor(total$Alley)
```

MasVnrType: Masonry veneer walls consist of a single non-structural external layer of masonry work, typically brick, backed by an air space. Here NA means that Masonry veneer wall is not existent.

MasVnrType and MasVnrArea have corresponding values of NA. Therefore, we set NA as None and MasVnrArea as 0.

```
total$MasVnrType <- as.character(total$MasVnrType)
total$MasVnrType[is.na(total$MasVnrType)] <- 'None'
total$MasVnrType <- as.factor(total$MasVnrType)

total$MasVnrArea <- as.character(total$MasVnrArea)
total$MasVnrArea[is.na(total$MasVnrArea)] <- '0.0'
total$MasVnrArea <- as.factor(total$MasVnrArea)
```

According to data dictionary, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2, FireplaceQU = NA means that the properties or homes do not have a basement.

```
total$BsmtQual <- as.character(total$BsmtQual)
total$BsmtQual[is.na(total$BsmtQual)] <- 'None'
total$BsmtQual <- as.factor(total$BsmtQual)
```

```
total$BsmtCond <- as.character(total$BsmtCond)
total$BsmtCond[is.na(total$BsmtCond)] <- 'None'
total$BsmtCond <- as.factor(total$BsmtCond)
```

```
total$BsmtExposure <- as.character(total$BsmtExposure)
total$BsmtExposure[is.na(total$BsmtExposure)] <- 'None'
total$BsmtExposure <- as.factor(total$BsmtExposure)
```

```
total$BsmtFinType1 <- as.character(total$BsmtFinType1)
total$BsmtFinType1[is.na(total$BsmtFinType1)] <- 'None'
total$BsmtFinType1 <- as.factor(total$BsmtFinType1)
```

```
total$BsmtFinType2 <- as.character(total$BsmtFinType2)
total$BsmtFinType2[is.na(total$BsmtFinType2)] <- 'None'
total$BsmtFinType2 <- as.factor(total$BsmtFinType2)
```

```
total$FireplaceQu <- as.character(total$FireplaceQu)
total$FireplaceQu[is.na(total$FireplaceQu)] <- 'None'
total$FireplaceQu <- as.factor(total$FireplaceQu)
```

```
total$GarageType <- as.character(total$GarageType )
total$GarageType[is.na(total$GarageType )] <- 'None'
total$GarageType <- as.factor(total$GarageType)
```

```
total$GarageFinish <- as.character(total$GarageFinish )
total$GarageFinish[is.na(total$GarageFinish )] <- 'None'
total$GarageFinish <- as.factor(total$GarageFinish)
```

```
total$GarageQual <- as.character(total$GarageQual )
total$GarageQual[is.na(total$GarageQual )] <- 'None'
total$GarageQual <- as.factor(total$GarageQual)
```

```
total$GarageCond <- as.character(total$GarageCond )
total$GarageCond[is.na(total$GarageCond )] <- 'None'
total$GarageCond <- as.factor(total$GarageCond)
```

```
total$PoolQC <- as.character(total$PoolQC )
total$PoolQC[is.na(total$PoolQC )] <- 'None'
total$PoolQC <- as.factor(total$PoolQC)
```

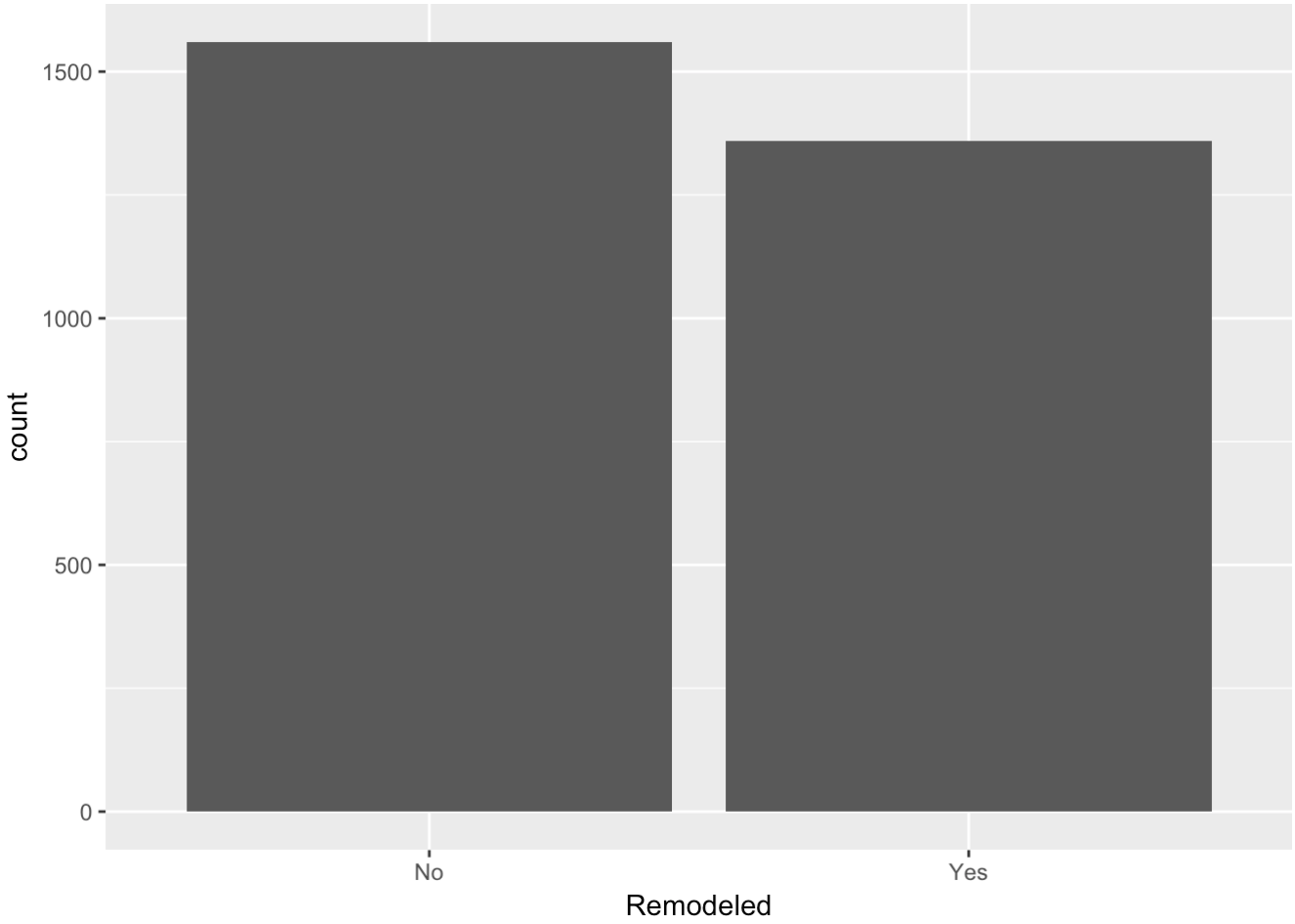
```
total$Fence <- as.character(total$Fence )
total$Fence[is.na(total$Fence )] <- 'None'
total$Fence <- as.factor(total$Fence)
```

```
total$MiscFeature <- as.character(total$MiscFeature )
total$MiscFeature[is.na(total$MiscFeature )] <- 'None'
total$MiscFeature <- as.factor(total$MiscFeature)
```

All missing values have either been imputed or filled with more meaningful values.

Let's explore the variable year built and year remodeled. The data dictionary states that if the year built is different from year remodeled, then the house was remodeled. We will create another column, a binary value/flag for remodeled.

```
total$Remodel_flag <- "Yes"
total[total$YearBuilt==total$YearRemodAdd,]$Remodel_flag <- "No"

# Number of remodeled homes
ggplot(total, aes(x = factor(total$Remodel_flag))) + geom_bar(stat = "count") + xlab("Re
modeled")
```




```
# Percentage of remodeled homes
paste(round(sum(total$Remodel_flag == "Yes")/nrow(total)*100, 2), '%')
```

```
## [1] "46.56 %"
```

Split data into train and test

```
train <- total[1:1460,]
test <- total[1461:2919,]
```

```
                        Exploratory data analysis plan
```

- Exploratory data analysis plan
- Here are the steps that we will take to understand the data and variables to get a better understanding of our data. We will on the explanation of each variable in data dictionary.

- Let's look at each individual variables carefully to understand the meaning of the variable and it's importance
- Analyze the dependent variable sale price.

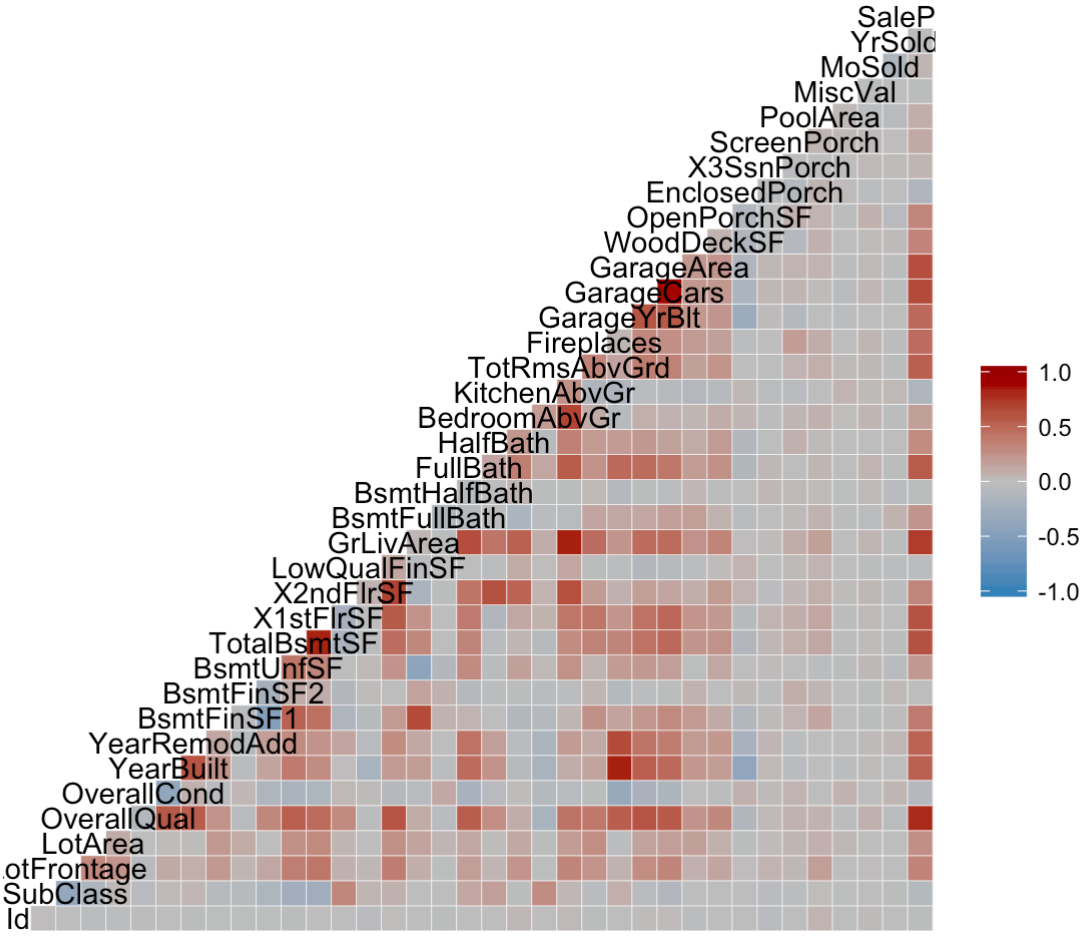
Let's plot the correlation matrix of numeric variables in the dataset

```
train_num <- train[sapply(train,is.numeric)]

#correlations <- cor(na.omit(train_num))
#row_indic <- apply(correlations, 1, function(x) sum(x > 0.3 | x < -0.3) > 1)
#correlations<- correlations[row_indic ,row_indic ]
#corrplot(correlations, method="square")

# Another way to visualize correlation matrix
library(GGally)
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
ggcorr(train_num, low = "steelblue", mid = "grey", high = "darkred")
```



Let's plot our dependent variable sales price

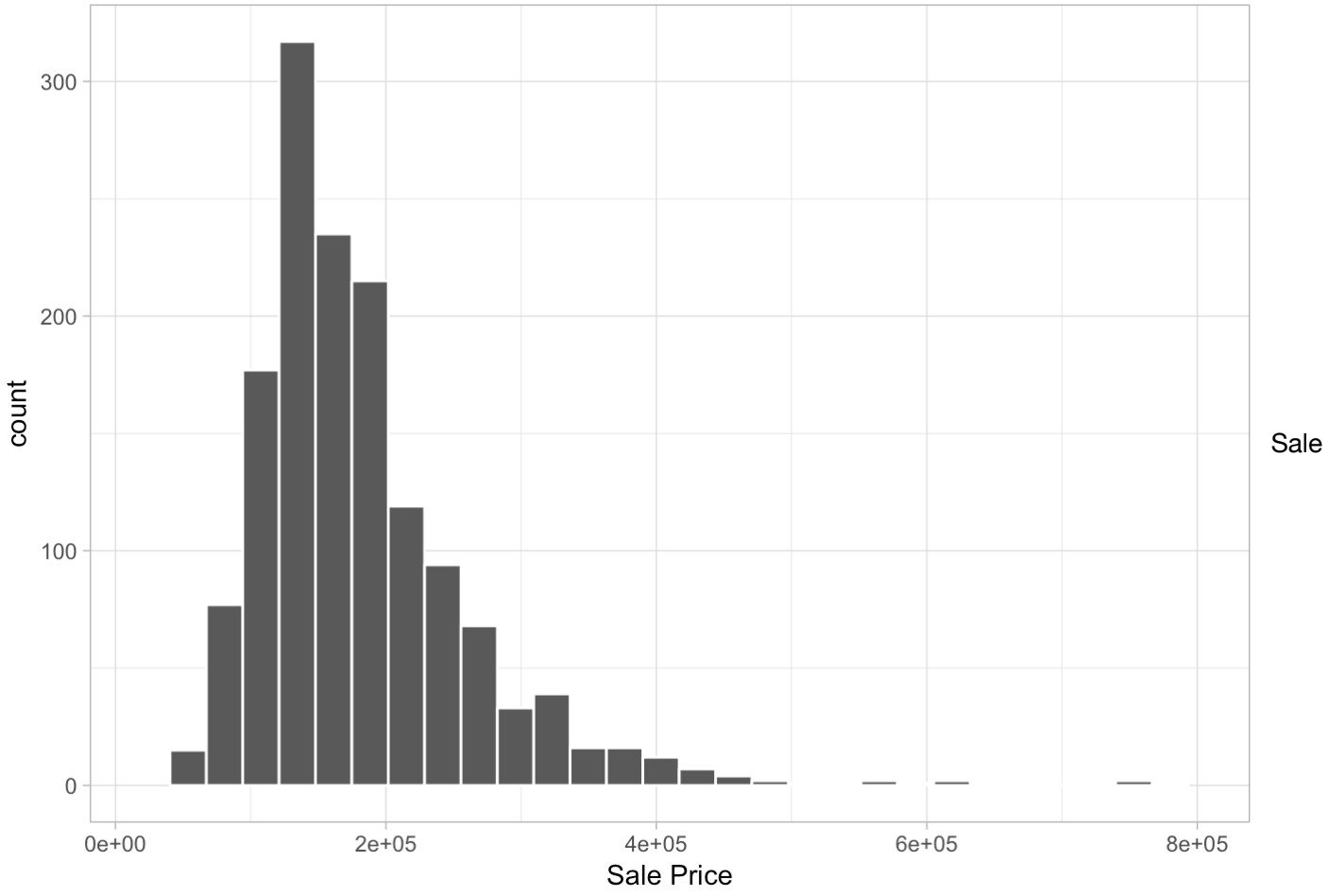
```
summary(train$SalePrice)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   34900  130000  163000  180900  214000  755000
```

```
ggplot(data=train, aes(train$SalePrice)) + geom_histogram(col = "white") + theme_light()
 + xlim(20000, 800000) + xlab("Sale Price")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

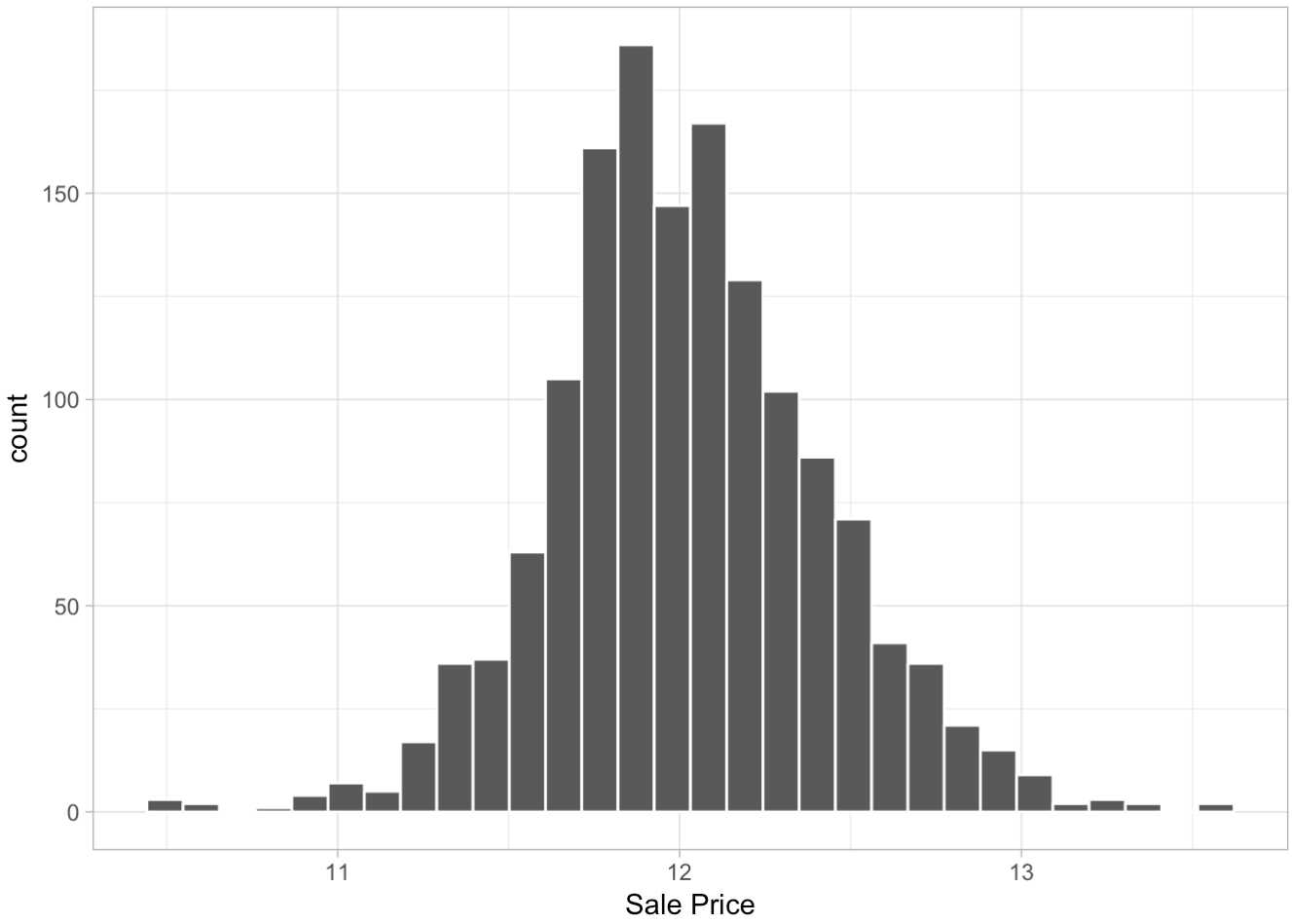
```
## Warning: Removed 1 rows containing missing values (geom_bar).
```



Price appears to be heavily skewed. We will log transform the variable to obtain a normal distribution of our dependent variable. This is to maintain positivity of the sale price variable, in all likelihood, sale price of a home will never be a negative value.

```
ggplot(data=train, aes(log(train$SalePrice))) + geom_histogram(col = "white") + theme_li
ght() + xlab("Sale Price")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


150
100
50
0
count
11
12
13
Sale Price