House Data EDA

Graeme Keleher

July 12, 2019

Load Tidyverse Suite of Packages

```r
library(tidyverse)  #Data wrangling and plotting
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.0     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(scales)     #Formatting axis
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

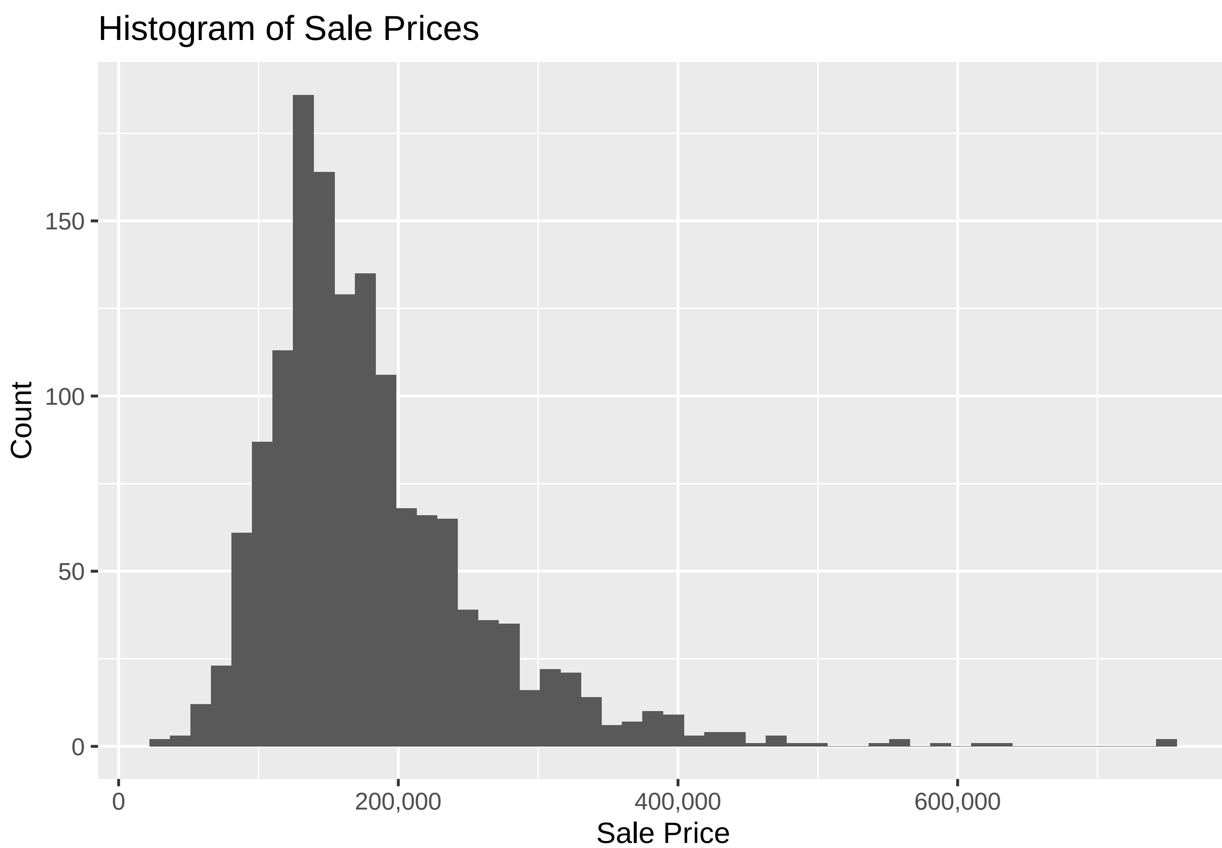
Import Training Data

```r
train <- read_csv("train.csv")
```

Sale Price

Insight: Not a normal distribution! Transformation could be a good idea.

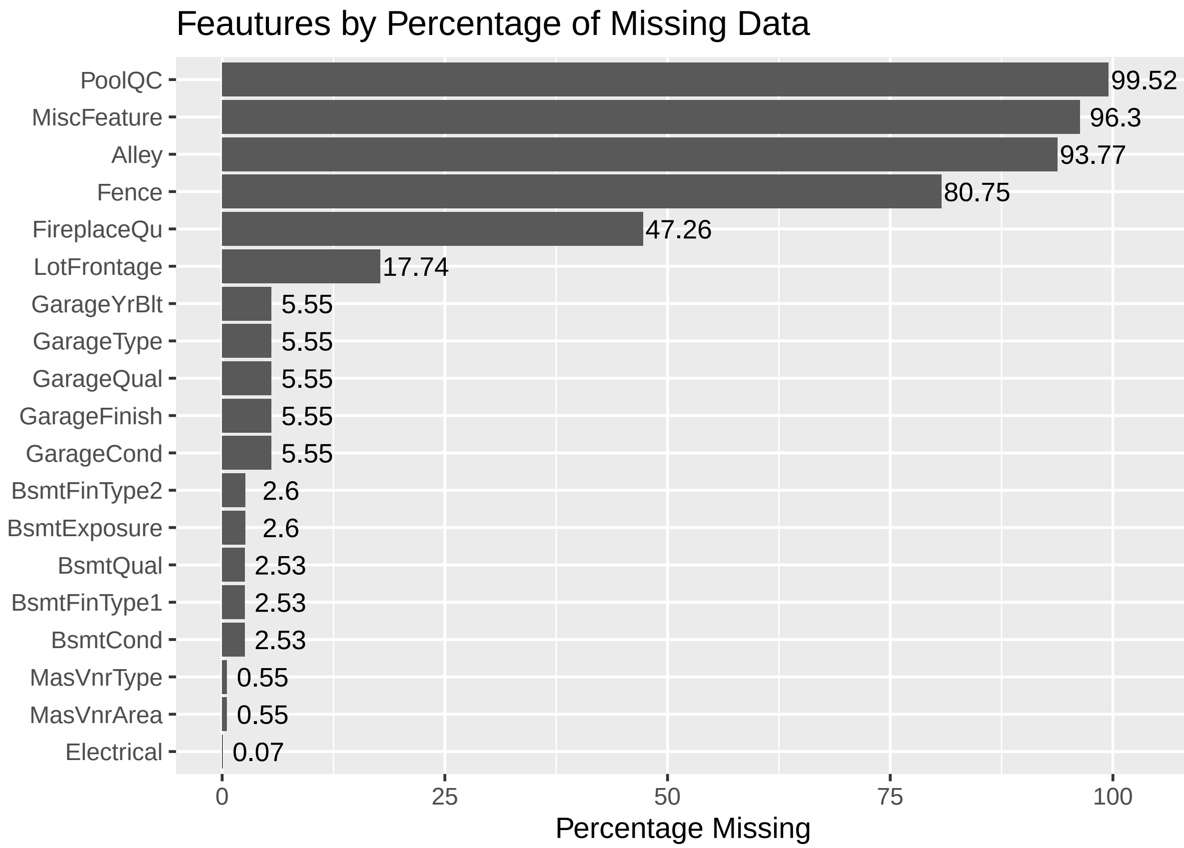
```r
ggplot(train,aes(SalePrice))+
  geom_histogram(bins = 50)+
  labs(title = "Histogram of Sale Prices",
       y = "Count",
       x = "Sale Price")+
  scale_x_continuous(labels = comma)
```



Explore Missing Values

Insight: A substantial number of features have missing data. The percentage missing varies widely. Different stratgies will most likely be required.

```
na_per <- as.data.frame(round(100*colSums(is.na(train))/nrow(train),2))

colnames(na_per) <- "Percent_NA"

na_per %>%
  rownames_to_column("Feature")%>%
  filter(Percent_NA != 0)%>%
  ggplot(aes(reorder(Feature,Percent_NA),Percent_NA))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(title = "Feautures by Percentage of Missing Data",
       y = "Percentage Missing",
       x = "")+
  geom_text(aes(label=Percent_NA),  size=3.5, nudge_y = 4)
```

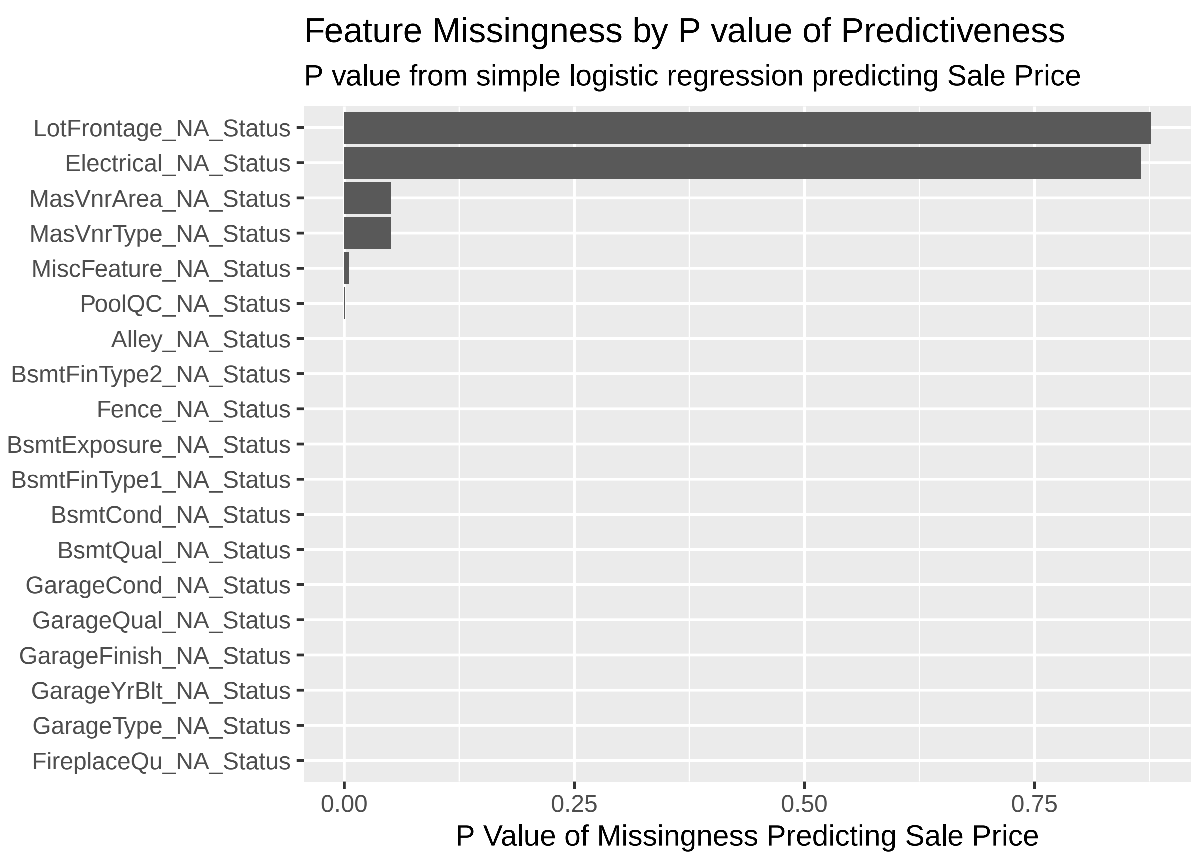


See if any missingness is predictive

Built a simple logistical regression model for the binary 'missingness' variable of each corresponding variable with missing values. It certainly looks like many of them are predictive. That said, some NAs have meaning outlined in the data description file. For example, an NA in the 'pool quality' feature ('PoolQC') means there is no pool.

```
#list of feautures with at least one NA
missing_data_columns <- colnames(train)[colSums(is.na(train)) >0]

train1 <- train
#make new column with this info
for(col in missing_data_columns){
  train1 <- mutate(train1, !!paste0(col,"_NA_Status") := ifelse(is.na(get(col)),1,0))
}

#Filter for Price and NA columns
Na_df <- train1 %>%
  select(SalePrice,contains("_NA_"))


p_vals <- data.frame(Var=character(),
                 P_value=numeric())

for( col in colnames(Na_df)[-1]){
  model <- glm( get(col) ~ SalePrice, data = Na_df, family = "binomial")
```

```
  s <- summary(model)$coefficients[2,4]
  p_vals <- rbind(p_vals,data.frame(Var=col, P_value=s))
}


p_vals %>%
  ggplot(aes(reorder(Var,P_value),P_value))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(title = "Feature Missingness by P value of Predictiveness",
       subtitle = "P value from simple logistic regression predicting Sale Price",
       y = "P Value of Missingness Predicting Sale Price",
       x = "")
```



Baseline Model

Built to

```
#transform character columns to factorogl
col_types <- sapply(train,class)

to_factor <- names(col_types[col_types == "character"])


#For character columns replace na with "NA"
#FOr numeric, repace na with 0
```

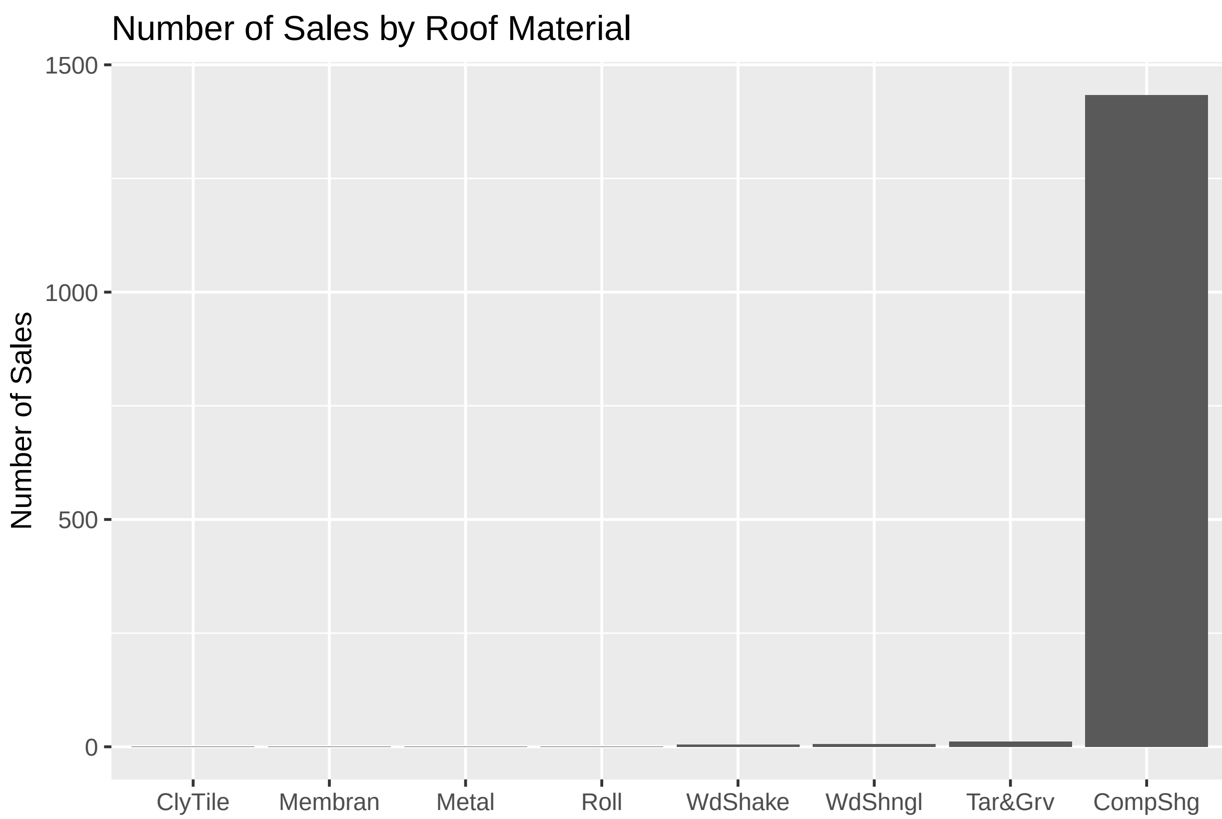
```r
train_f <- train[to_factor]
train_n <- train%>%
  select(-to_factor)

train_f[is.na(train_f)] <- "Data_Missing"
train_n[is.na(train_n)] <- 0

train_processed <- cbind(train_f, train_n)

train_processed[to_factor] <- lapply(train_processed[to_factor], as.factor)


Baseline_Model <- lm(SalePrice ~., data = train_processed)

baseline_c <- data.frame(Feature = row.names(summary(Baseline_Model)$coefficients),
                         summary(Baseline_Model)$coefficients)

baseline_c %>%
  arrange(desc(Estimate))%>%
  slice(1:10)
```

```
##            Feature  Estimate Std..Error   t.value     Pr...t..
## 1  RoofMatlMembran 666423.56   62346.13 10.689093 1.531620e-25
## 2    RoofMatlMetal 635417.18   62015.05 10.246178 1.115054e-23
## 3  RoofMatlWdShngl 625122.19   53294.35 11.729614 3.655448e-30
## 4  RoofMatlTar&Grv 572409.66   56285.81 10.169697 2.303040e-23
## 5  RoofMatlCompShg 570920.35   52425.11 10.890208 2.082251e-26
## 6  RoofMatlWdShake 562871.33   54758.18 10.279219 8.139633e-24
## 7     RoofMatlRoll 558904.59   58148.96  9.611600 3.996258e-21
## 8     GarageQualEx 120084.19   29865.07  4.020891 6.158060e-05
## 9    RoofStyleShed  99369.49   34518.06  2.878769 4.062835e-03
## 10      BsmtCondPo  65699.66   30003.04  2.189766 2.873237e-02
```

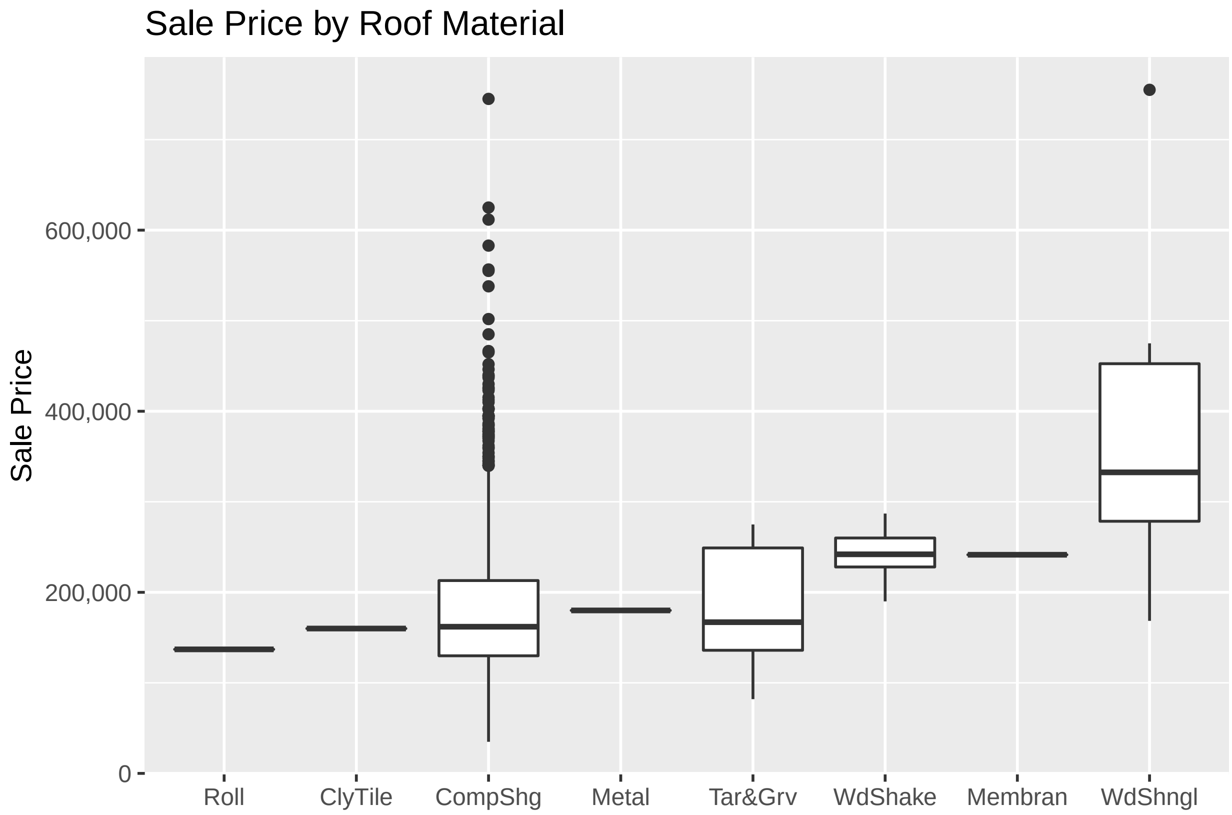
Roof Material

Insight: Wooden Shingles are HIGHLY predictive of a more expensive home. Unfortunatley, these are highly rare

```r
train_processed %>%
  group_by(RoofMatl)%>%
  summarise(class_total = n())%>%
  ggplot(aes(reorder(RoofMatl,class_total), class_total))+
  geom_bar(stat = "identity")+
  labs(title = "Number of Sales by Roof Material",
       y = "Number of Sales",
       x = "")
```




```r
train_processed %>%
  ggplot(aes(reorder(RoofMatl, SalePrice), SalePrice))+
  geom_boxplot()+
  labs(title = "Sale Price by Roof Material",
       y = "Sale Price",
       x = "")+
  scale_y_continuous(labels = comma)
```

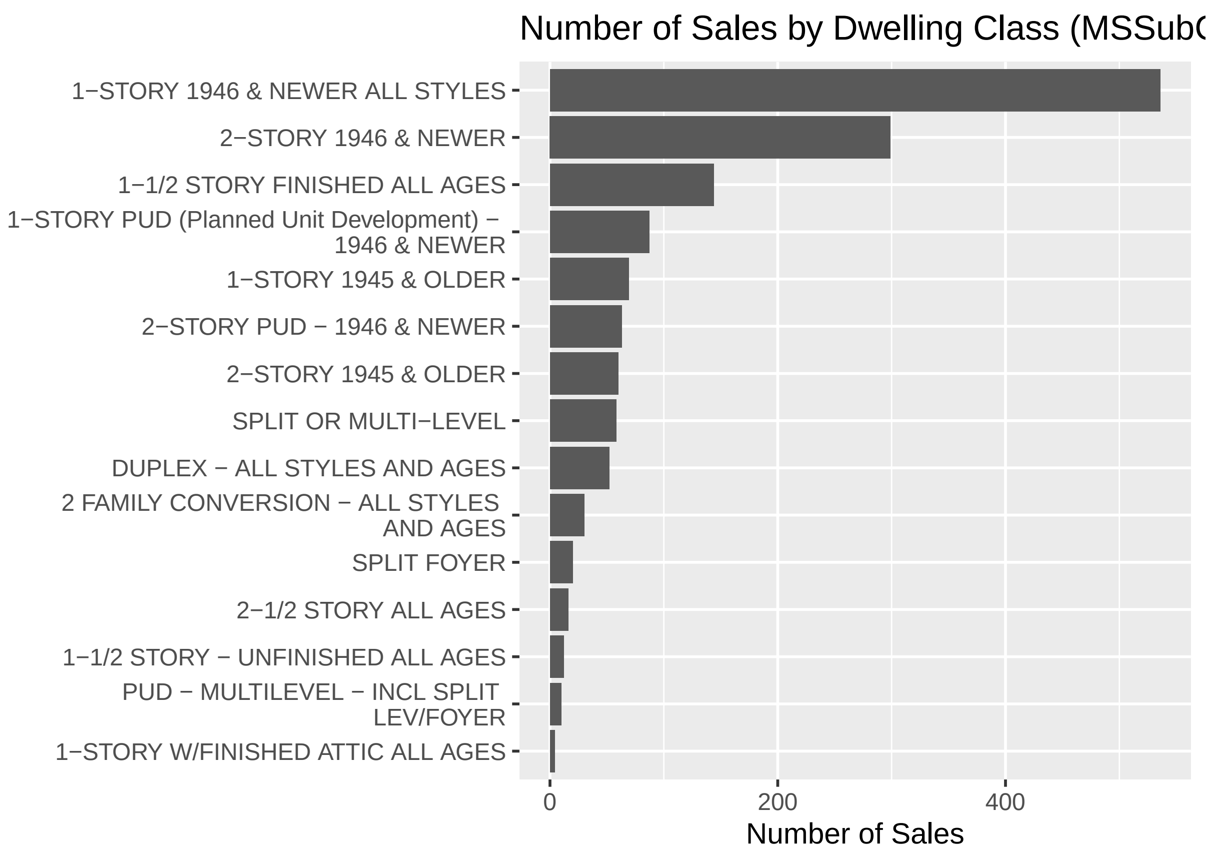


MSSubClass Feature

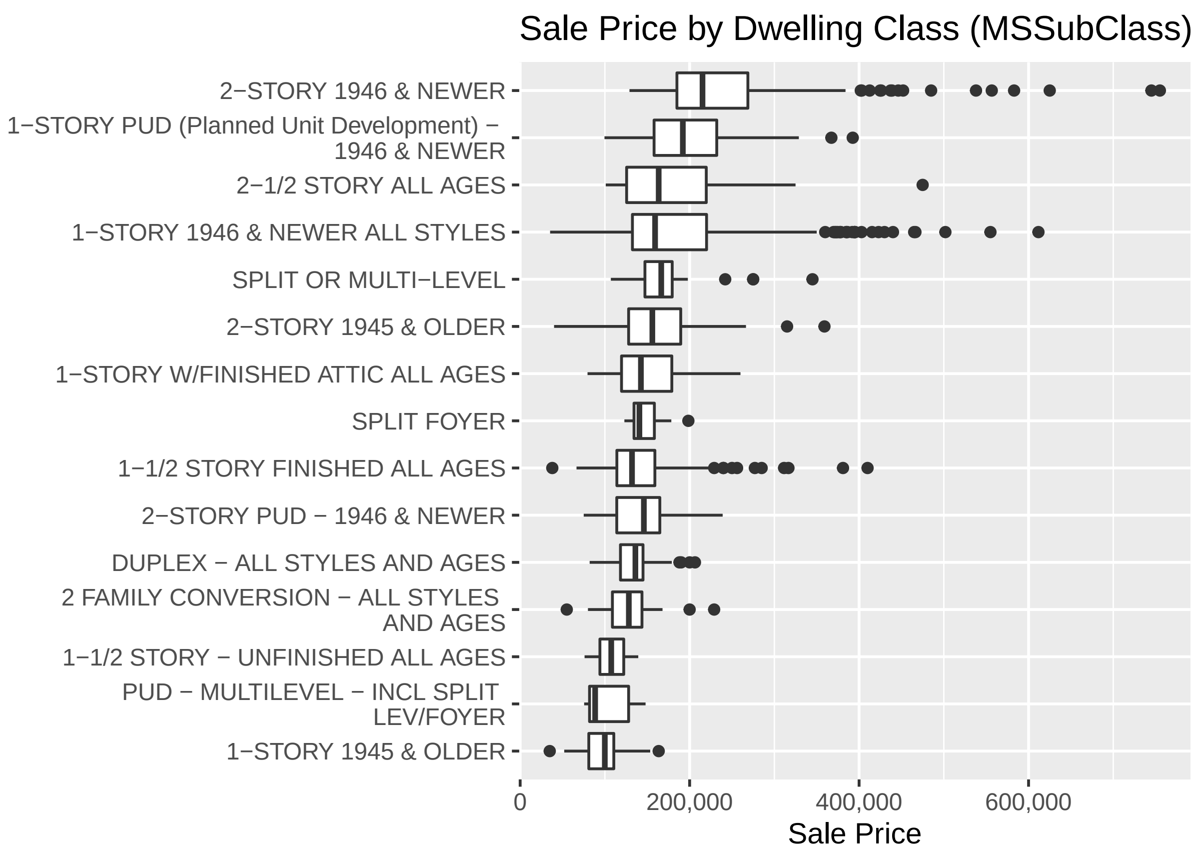
Insight: High class imbalance, looks predictive

```
train$MSSubClass <- as.factor(train$MSSubClass)

levels(train$MSSubClass) <- list("1-STORY 1946 & NEWER ALL STYLES" = "20",
                  "1-STORY 1945 & OLDER" = "30",
                  "1-STORY W/FINISHED ATTIC ALL AGES" = "40",
                  "1-1/2 STORY - UNFINISHED ALL AGES" = "45",
                  "1-1/2 STORY FINISHED ALL AGES" = "50",
                  "2-STORY 1946 & NEWER" = "60",
                  "2-STORY 1945 & OLDER" = "70",
                  "2-1/2 STORY ALL AGES" = "75",
                  "SPLIT OR MULTI-LEVEL" = "80",
                  "SPLIT FOYER" = "85",
                  "DUPLEX - ALL STYLES AND AGES" = "90",
                  "1-STORY PUD (Planned Unit Development) - \n1946 & NEWER" = "120",
                  "1-1/2 STORY PUD - ALL AGES" = "150",
                  "2-STORY PUD - 1946 & NEWER" = "160",
                  "PUD - MULTILEVEL - INCL SPLIT \nLEV/FOYER" = "180",
                  "2 FAMILY CONVERSION - ALL STYLES \nAND AGES" = "190")
```

```r
train %>%
  group_by(MSSubClass)%>%
  summarise(class_total = n())%>%
  ggplot(aes(reorder(MSSubClass,class_total), class_total))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(title = "Number of Sales by Dwelling Class (MSSubClass)",
       y = "Number of Sales",
       x = "")
```




```r
train %>%
  ggplot(aes(reorder(MSSubClass, SalePrice), SalePrice))+
  geom_boxplot()+
  coord_flip()+
  labs(title = "Sale Price by Dwelling Class (MSSubClass)",
       y = "Sale Price",
       x = "")+
  scale_y_continuous(labels = comma)
```

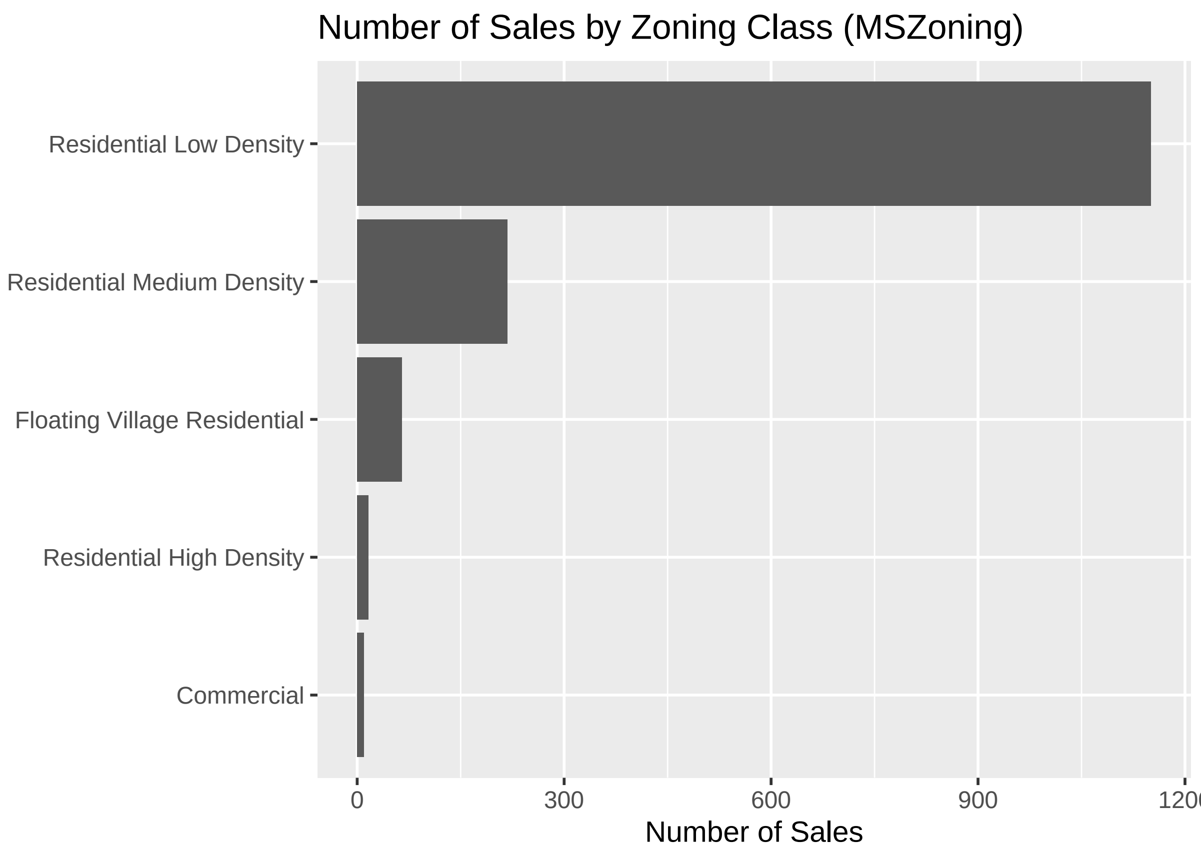


MSZoning

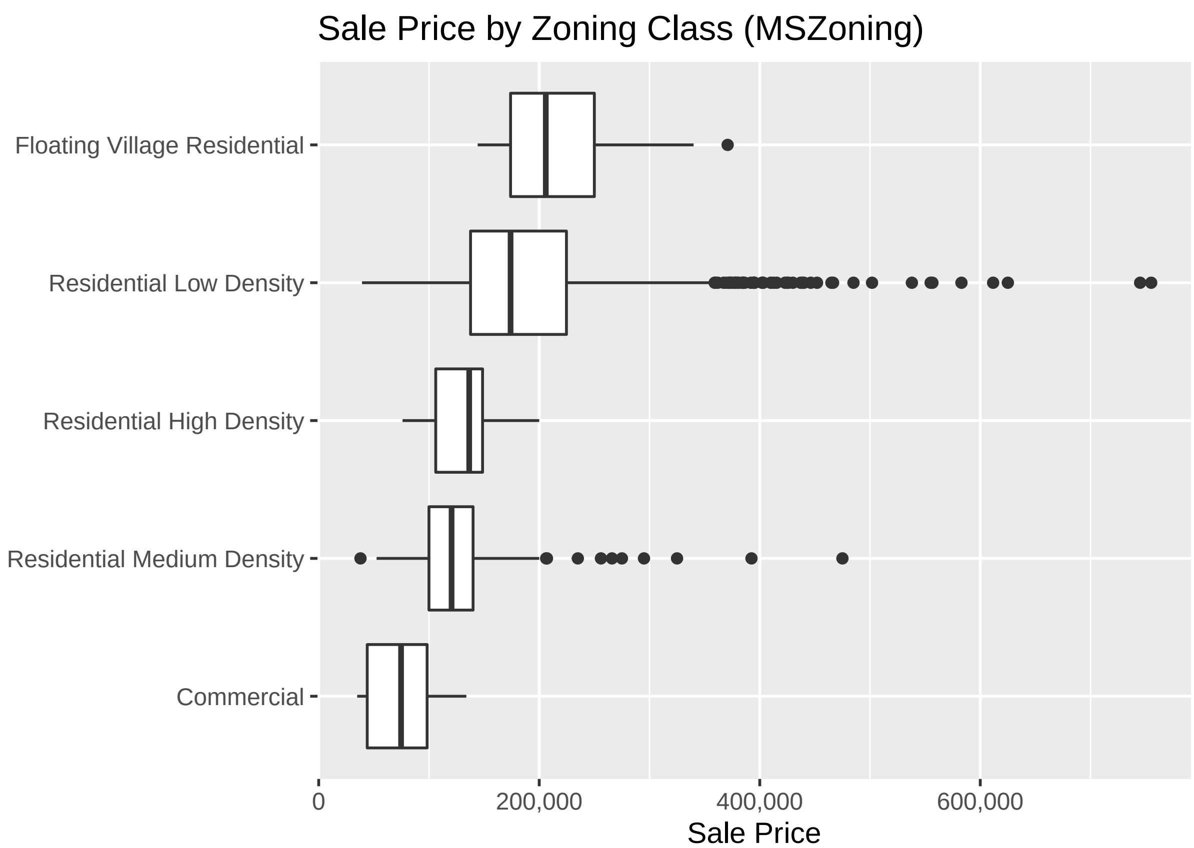
Insight: High class imbalance, looks predictive

```
train$MSZoning <- as.factor(train$MSZoning)

levels(train$MSZoning) <- list("Agriculture" = "A",
                  "Commercial" = "C (all)",
                  "Floating Village Residential" = "FV",
                  "Industrial" = "I",
                  "Residential High Density" = "RH",
                  "Residential Low Density" = "RL",
                  "Residential Low Density Park" = "RP",
                  "Residential Medium Density" = "RM")



train %>%
  group_by(MSZoning)%>%
  summarise(class_total = n())%>%
  ggplot(aes(reorder(MSZoning,class_total), class_total))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(title = "Number of Sales by Zoning Class (MSZoning)",
       y = "Number of Sales",
```

```
       x = "")
```




```
train %>%
  ggplot(aes(reorder(MSZoning, SalePrice), SalePrice))+
  geom_boxplot()+
  coord_flip()+
  labs(title = "Sale Price by Zoning Class (MSZoning)",
       y = "Sale Price",
       x = "")+
  scale_y_continuous(labels = comma)
```



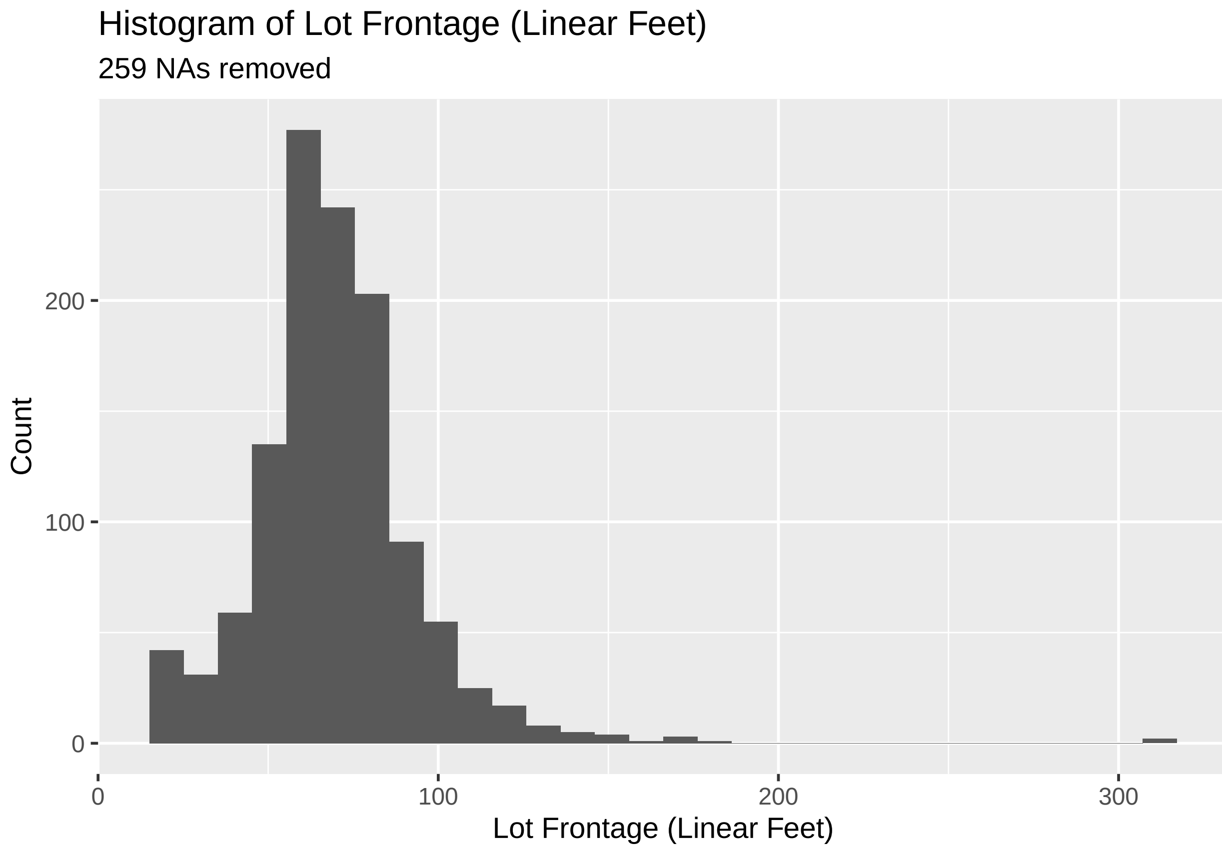
LotFrontage

Insight: Doesn't look very predictive,

```
train %>%
  ggplot(aes(LotFrontage))+
  geom_histogram()+
  labs(title = "Histogram of Lot Frontage (Linear Feet)",
       subtitle = "259 NAs removed",
       y = "Count",
       x = "Lot Frontage (Linear Feet)")
```

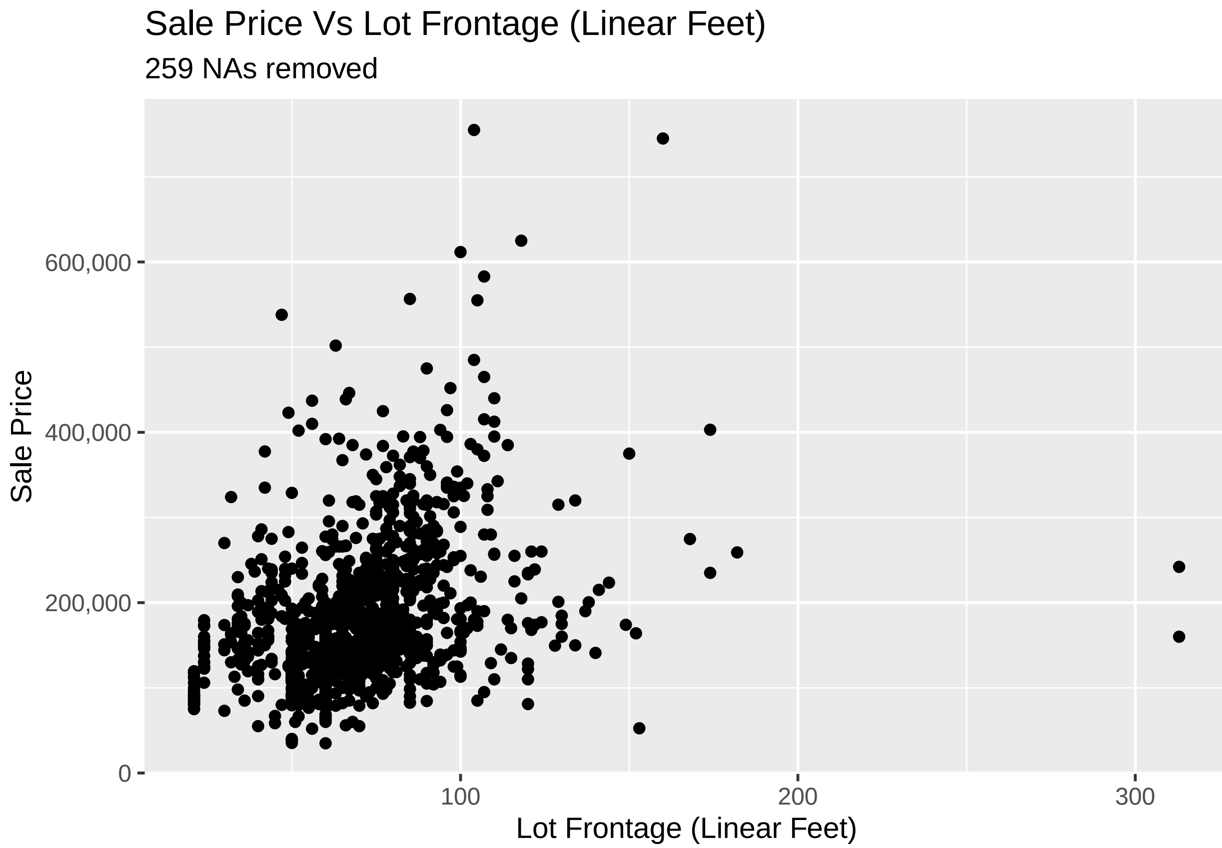
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 259 rows containing non-finite values (stat_bin).
```

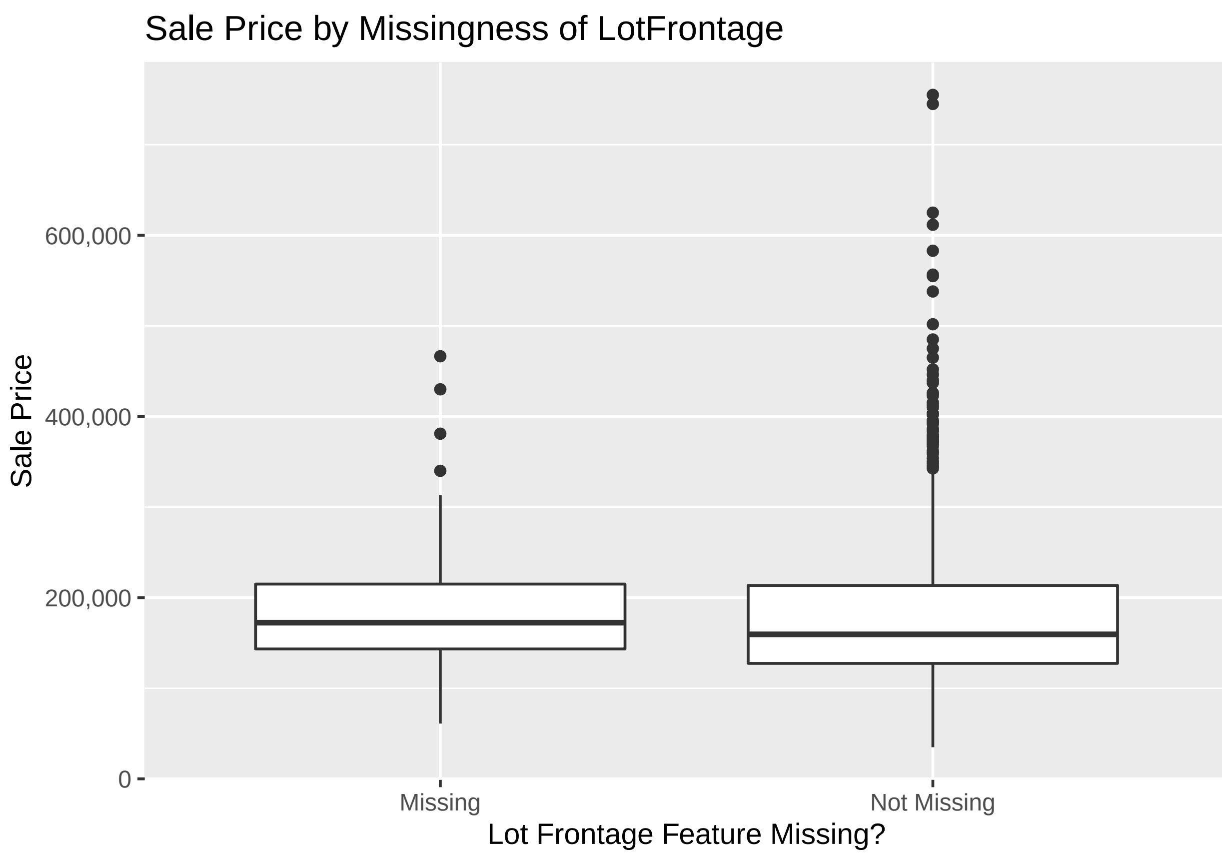



```r
train %>%
  ggplot(aes(LotFrontage, SalePrice))+
  geom_point()+
  labs(title = "Sale Price Vs Lot Frontage (Linear Feet)",
       subtitle = "259 NAs removed",
       y = "Sale Price",
       x = "Lot Frontage (Linear Feet)")+
  scale_y_continuous(labels = comma)
```

```
## Warning: Removed 259 rows containing missing values (geom_point).
```



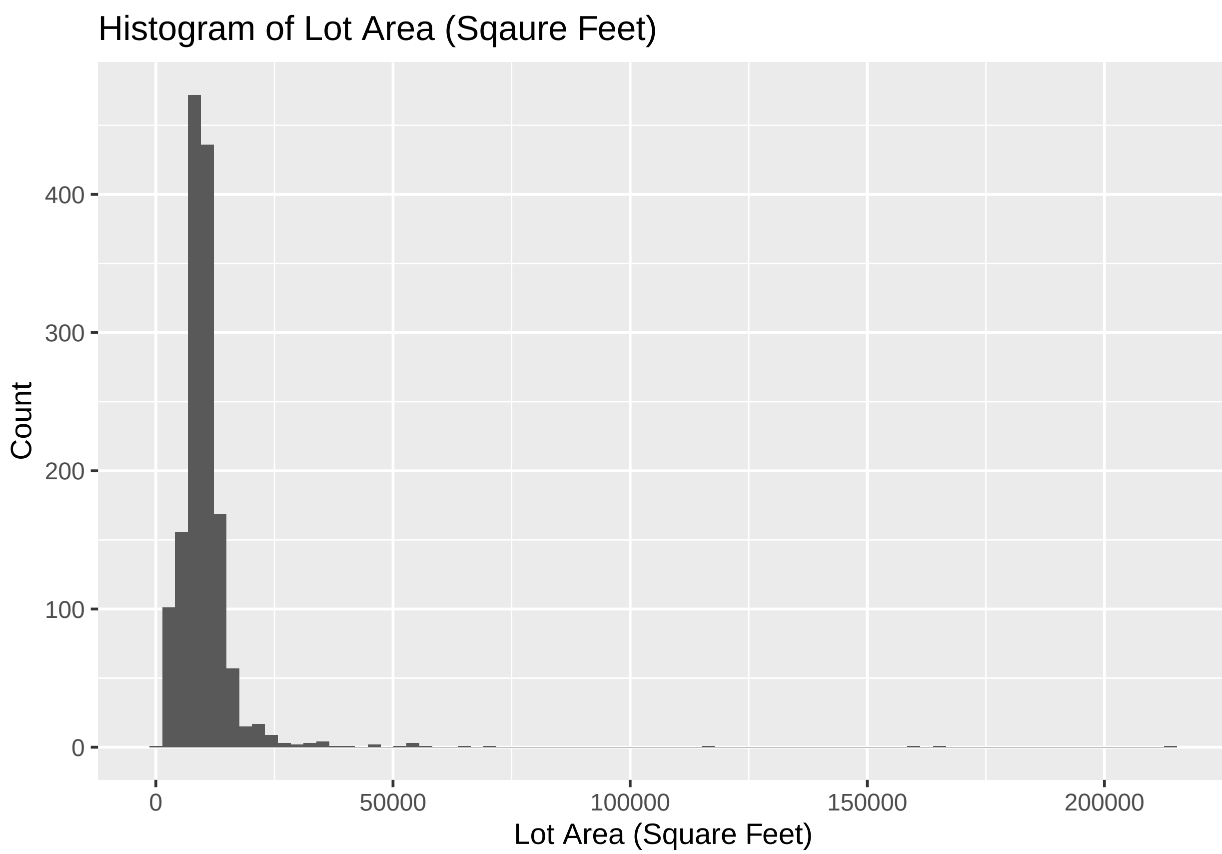

```
train %>%
  mutate(Lot_front_na = ifelse(is.na(LotFrontage),"Missing","Not Missing"))%>%
  ggplot(aes(Lot_front_na, SalePrice))+
  geom_boxplot()+
  labs(title = "Sale Price by Missingness of LotFrontage",
       y = "Sale Price",
       x = "Lot Frontage Feature Missing?")+
  scale_y_continuous(labels = comma)
```



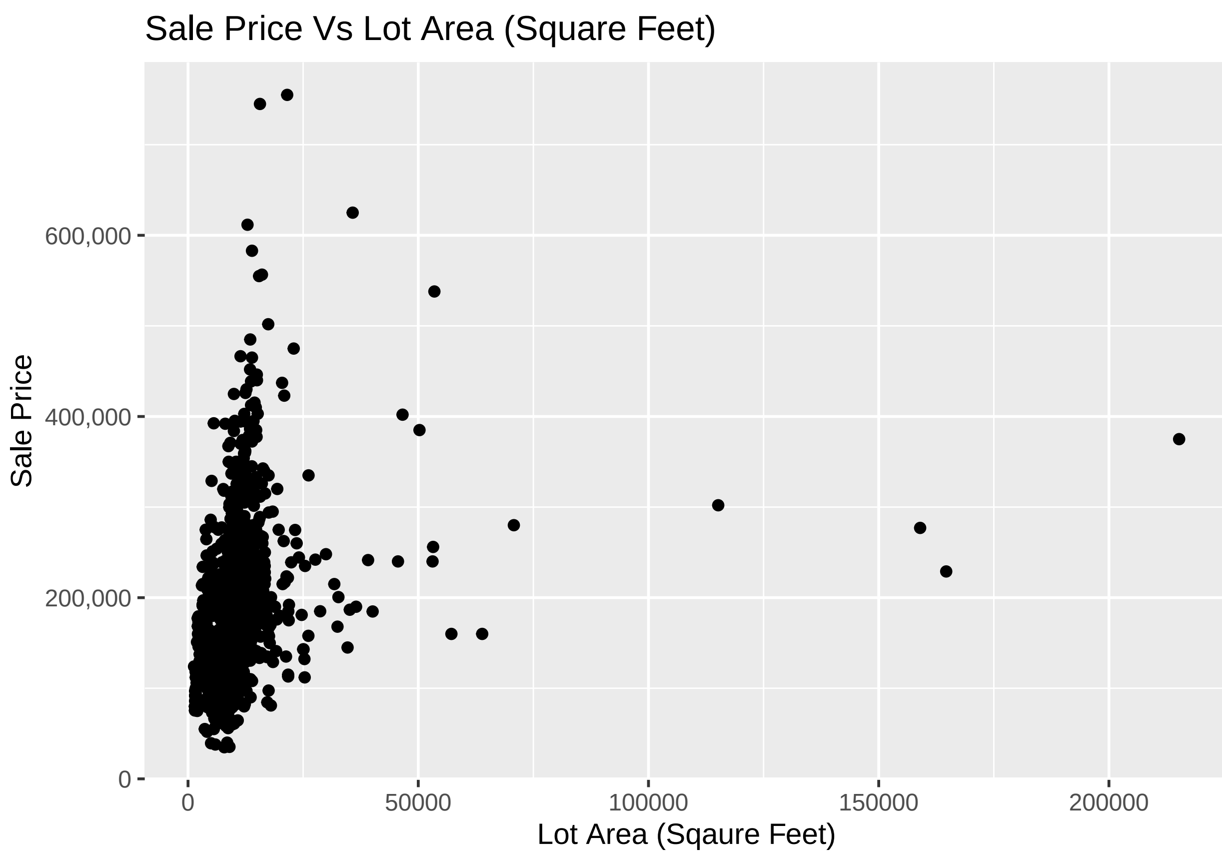
LotArea

Insight: Looks somewhat predictive, OUTLIERS

```
train %>%
  ggplot(aes(LotArea))+
  geom_histogram(bins = 80)+
  labs(title = "Histogram of Lot Area (Sqaure Feet)",
       y = "Count",
       x = "Lot Area (Square Feet)")
```



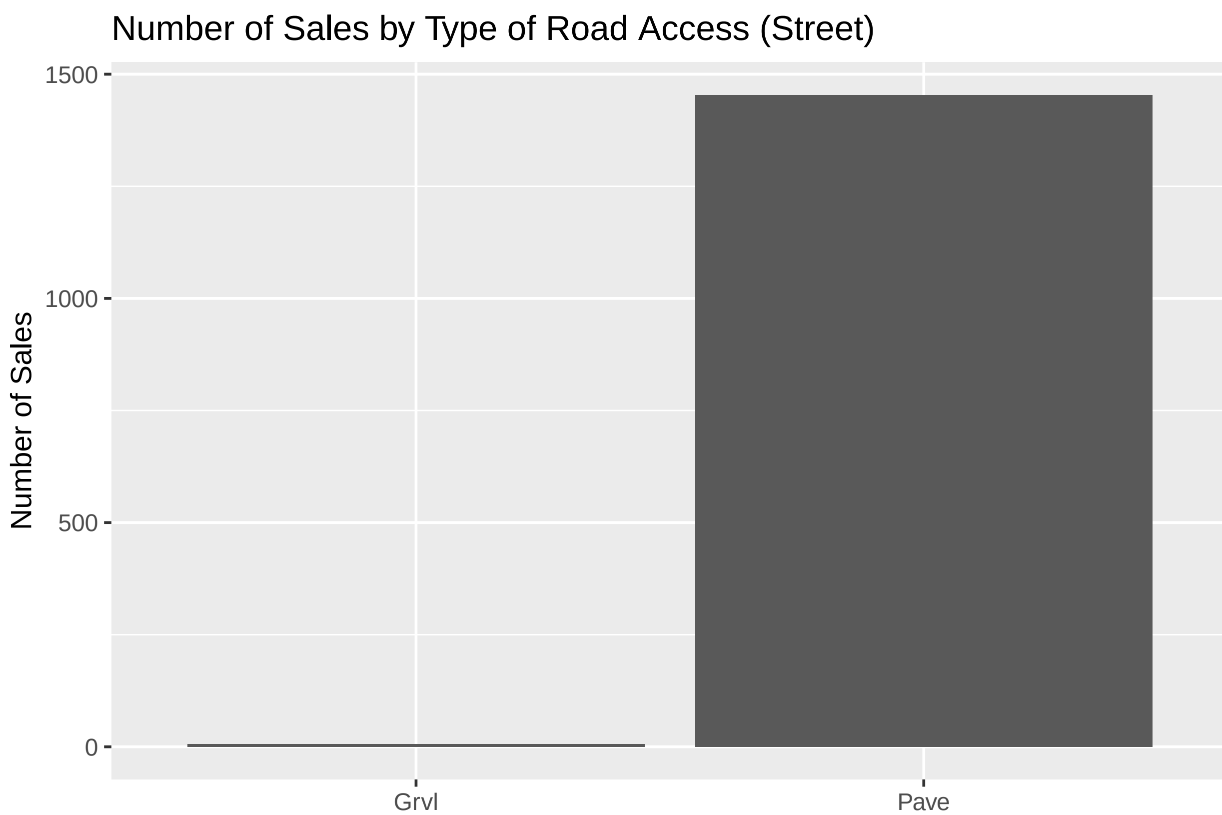

```r
train %>%
  ggplot(aes(LotArea, SalePrice))+
  geom_point()+
  labs(title = "Sale Price Vs Lot Area (Square Feet)",
       y = "Sale Price",
       x = "Lot Area (Sqaure Feet)")+
  scale_y_continuous(labels = comma)
```



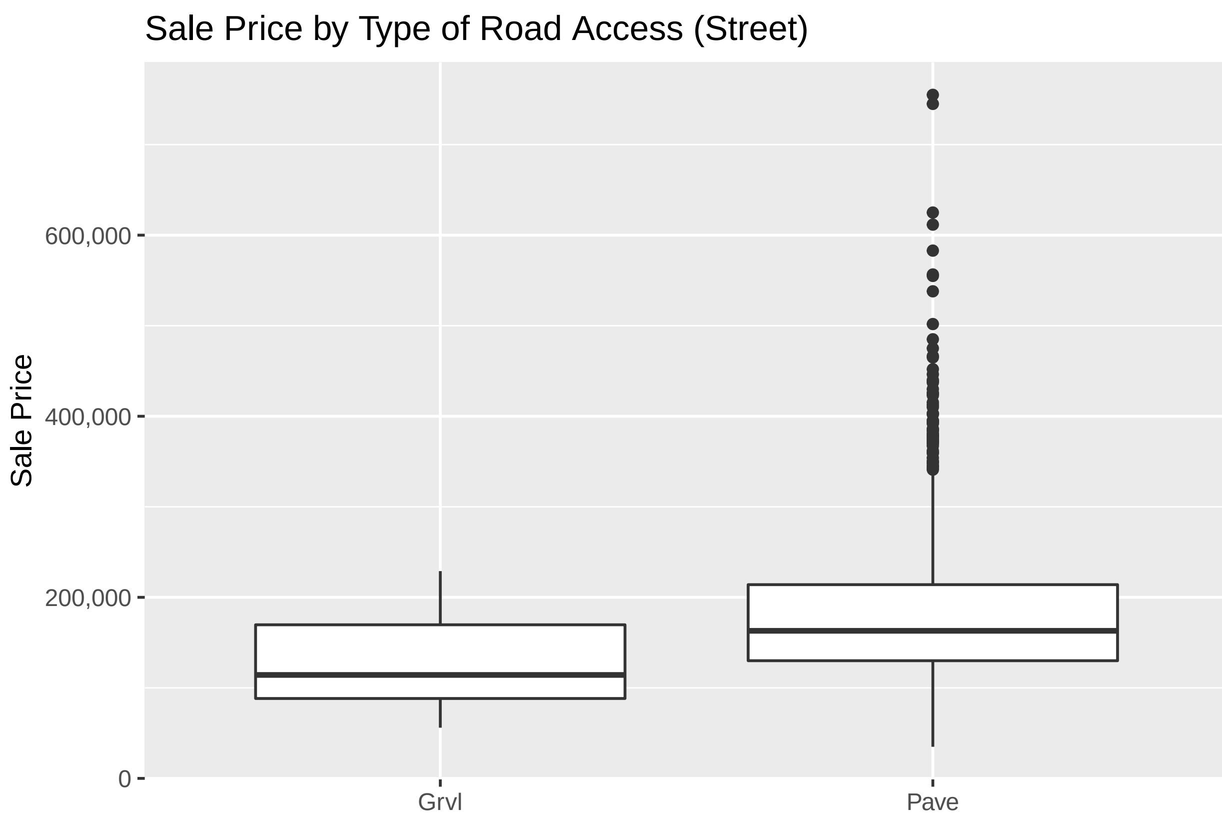
Street

Insight: High class imbalance, possibly predictive

```
train$Street <- as.factor(train$Street)


train %>%
  group_by(Street)%>%
  summarise(class_total = n())%>%
  ggplot(aes(reorder(Street,class_total), class_total))+
  geom_bar(stat = "identity")+
  labs(title = "Number of Sales by Type of Road Access (Street)",
       y = "Number of Sales",
       x = "")
```



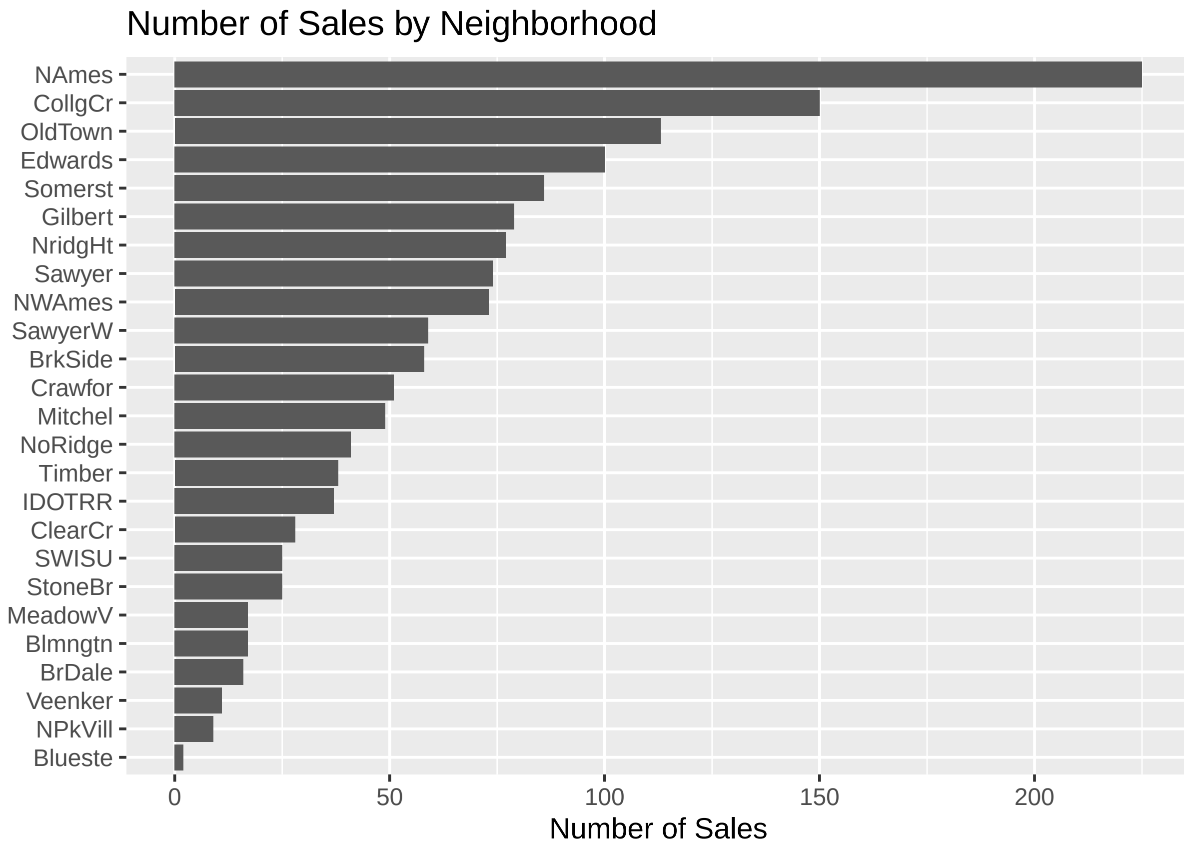

```r
train %>%
  ggplot(aes(reorder(Street, SalePrice), SalePrice))+
  geom_boxplot()+
  labs(title = "Sale Price by Type of Road Access (Street)",
       y = "Sale Price",
       x = "")+
  scale_y_continuous(labels = comma)
```



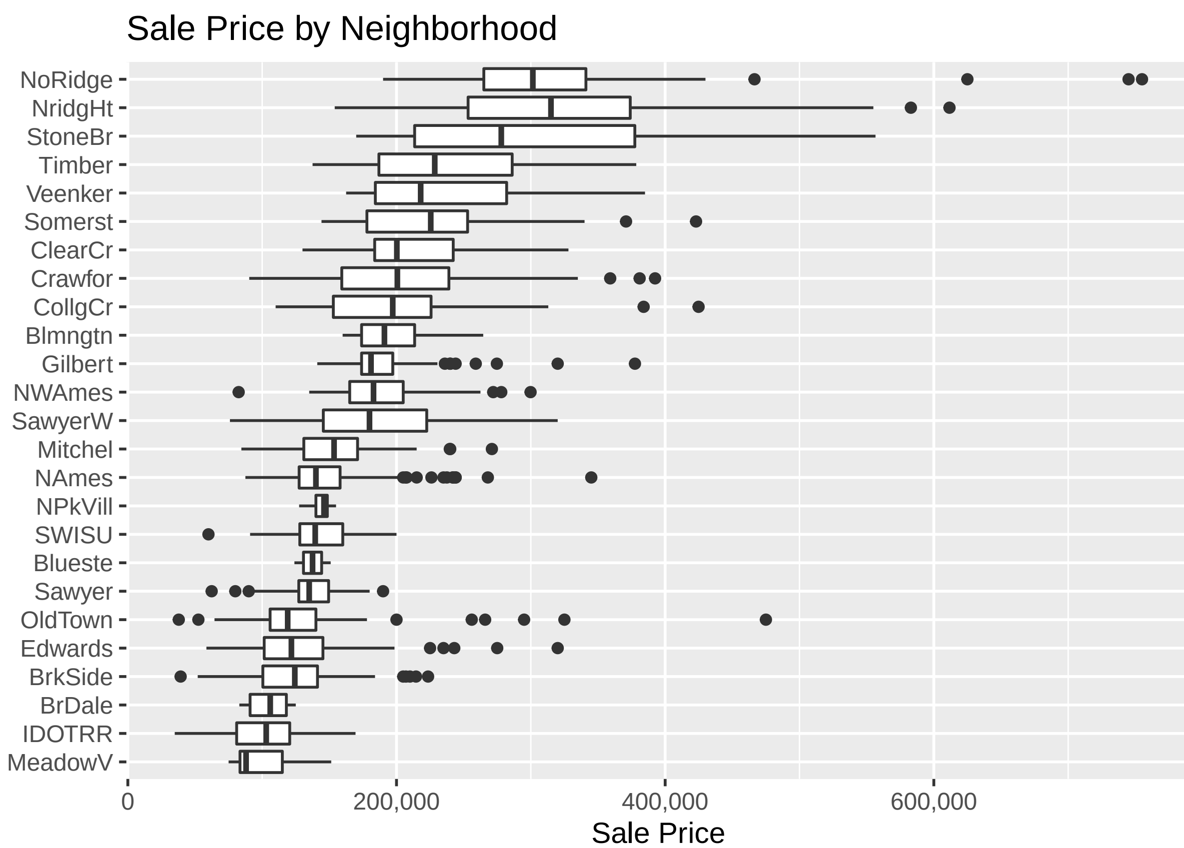
Neighborhood Feature

Insight: High class imbalance, looks predictive

```
train$Neighborhood <- as.factor(train$Neighborhood)



train %>%
  group_by(Neighborhood )%>%
  summarise(class_total = n())%>%
  ggplot(aes(reorder(Neighborhood ,class_total), class_total))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(title = "Number of Sales by Neighborhood ",
       y = "Number of Sales",
       x = "")
```



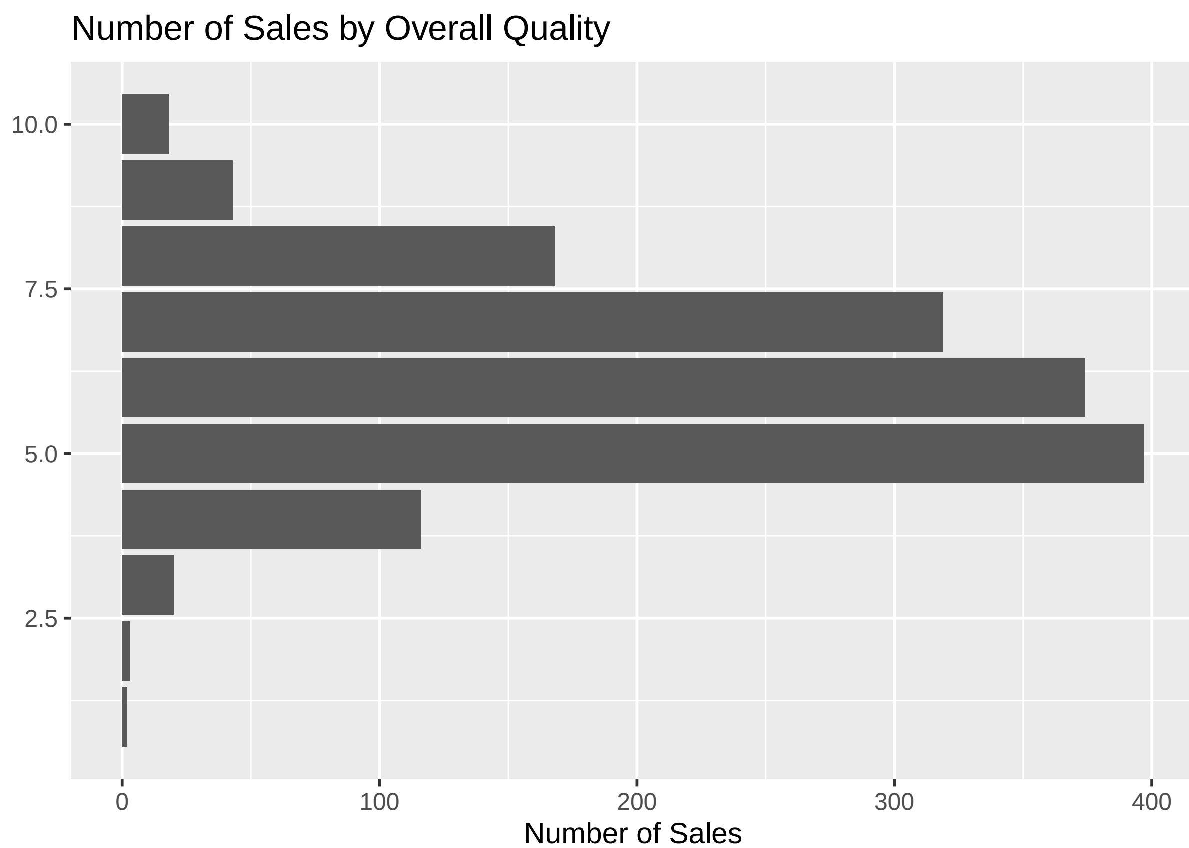

```r
train %>%
  ggplot(aes(reorder(Neighborhood , SalePrice), SalePrice))+
  geom_boxplot()+
  coord_flip()+
  labs(title = "Sale Price by Neighborhood ",
       y = "Sale Price",
       x = "")+
  scale_y_continuous(labels = comma)
```



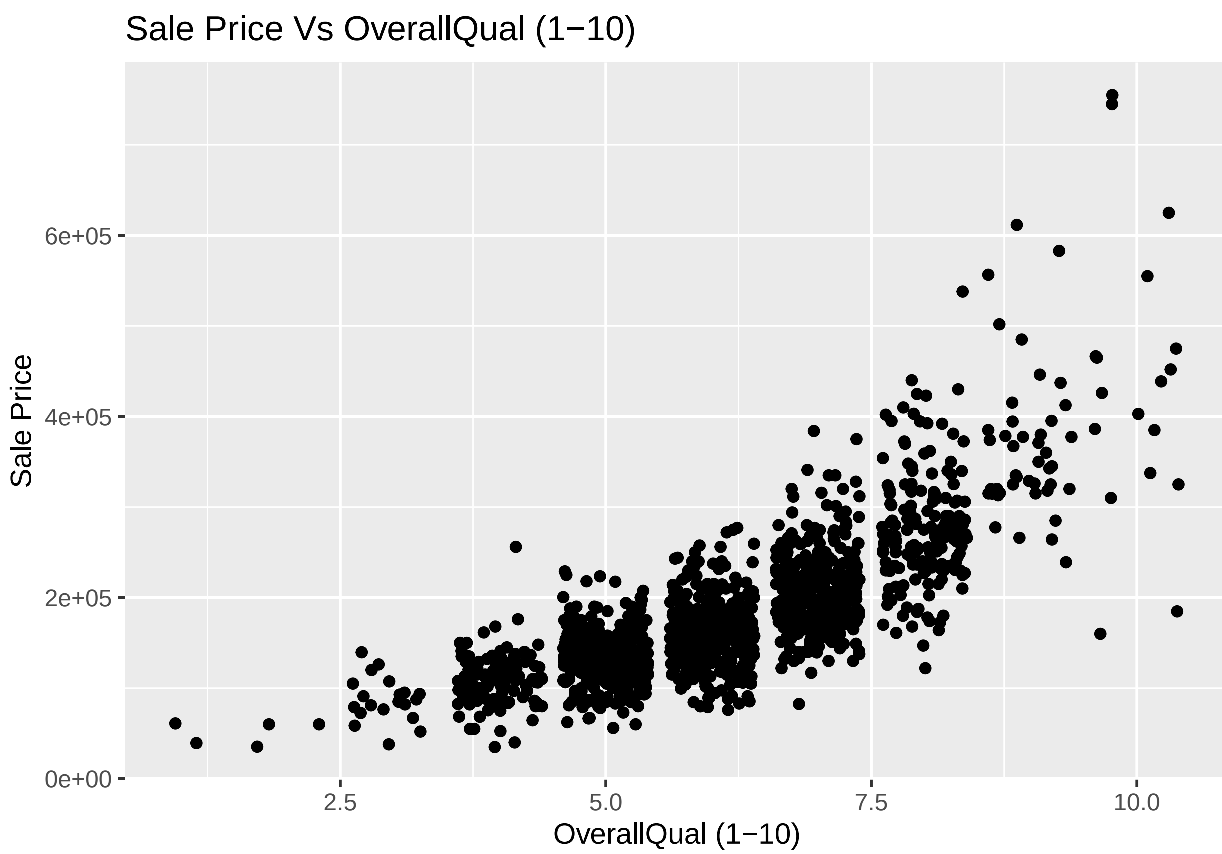
OverallQual

Insight: Looks VERY predictive

```
train %>%
  group_by(OverallQual)%>%
  summarise(class_total = n())%>%
  ggplot(aes(OverallQual, class_total))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(title = "Number of Sales by Overall Quality",
       y = "Number of Sales",
       x = "")
```



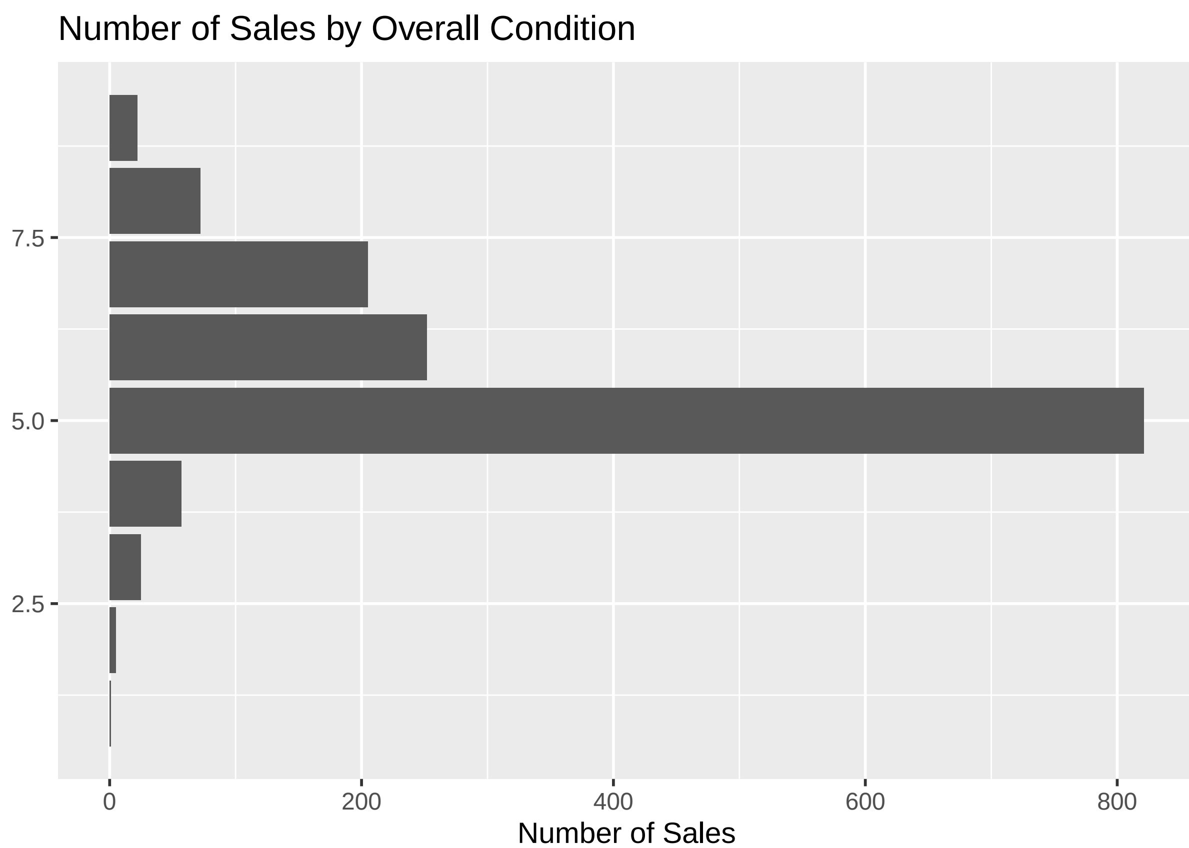

```
train %>%
  ggplot(aes(OverallQual, SalePrice))+
  geom_jitter()+
  labs(title = "Sale Price Vs OverallQual (1-10)",
       y = "Sale Price",
       x = "OverallQual (1-10)")
```



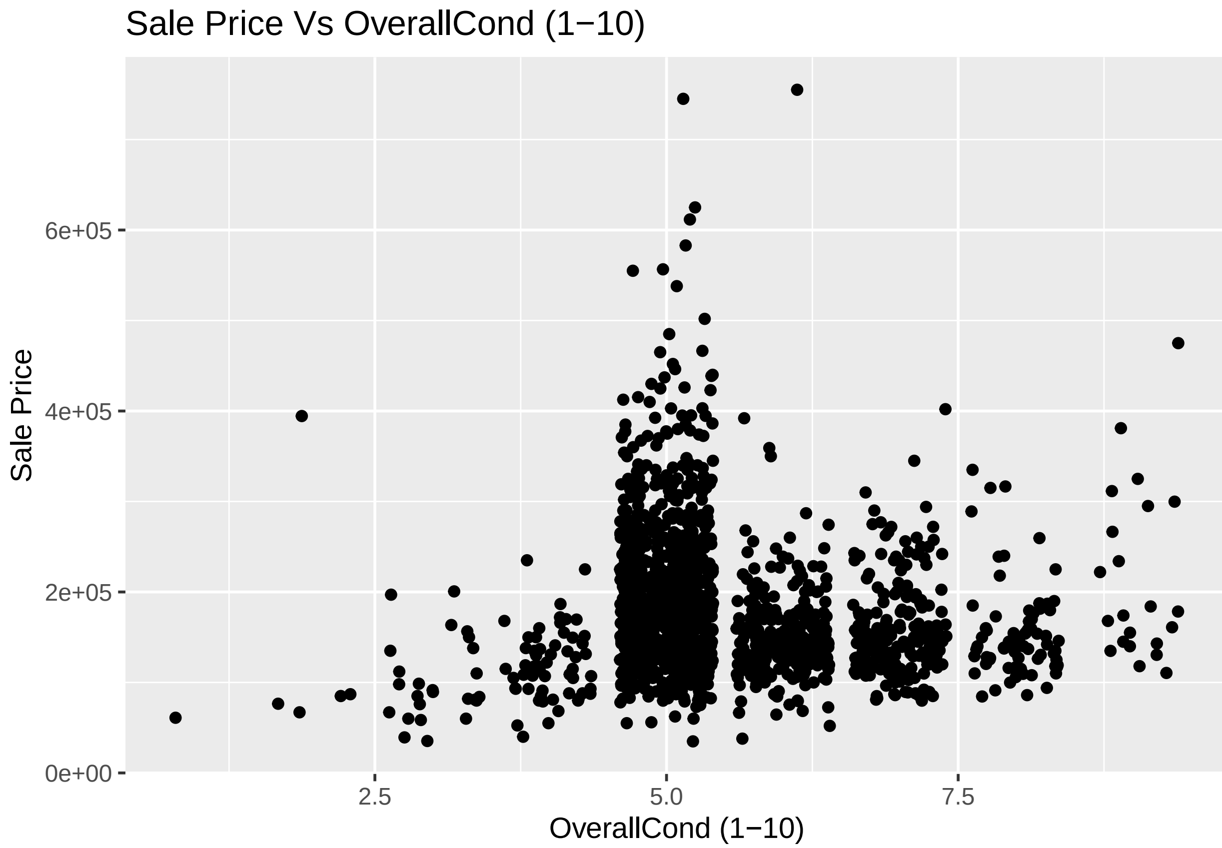
OverallCond

Insight: Doesn;t look very predictive

```
train %>%
  group_by(OverallCond)%>%
  summarise(class_total = n())%>%
  ggplot(aes(OverallCond, class_total))+
  geom_bar(stat = "identity")+
  coord_flip()+
  labs(title = "Number of Sales by Overall Condition",
       y = "Number of Sales",
       x = "")
```




```r
train %>%
  ggplot(aes(OverallCond, SalePrice))+
  geom_jitter()+
  labs(title = "Sale Price Vs OverallCond (1-10)",
       y = "Sale Price",
       x = "OverallCond (1-10)")
```



Sale Price Vs OverallCond (1−10)
Sale Price
6e+05
4e+05
2e+05
0e+00
2.5
5.0
7.5
OverallCond (1−10)