Logistic Regression

Project Team

1. Tejaswini Nutalapati
2. Aditi Bhargava

Loading the Libraries

```
library(ggplot2)
library(readr)
library(xgboost)
library(caret)

## Loading required package: lattice

library(randomForest)

## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

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

library(Matrix)
library(corrplot)

## corrplot 0.84 loaded

library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':
##
##     combine

## The following object is masked from 'package:xgboost':
##
##     slice
```

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

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

library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:randomForest':
##
##     combine

library(cowplot)

list.files("../input")

## character(0)
```

Loading the Data

```
Train<-read.csv("/Users/tejaswininutalapati/Documents/Multivariate Analysis/Project/DataSet/train.csv")
Test<-read.csv("/Users/tejaswininutalapati/Documents/Multivariate Analysis/Project/DataSet/test.csv")

Test$SalePrice <- -1
df <- rbind(Train,Test)
str(df)

## 'data.frame':    2919 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 7
420 ...
##  $ 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 N
A 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
```

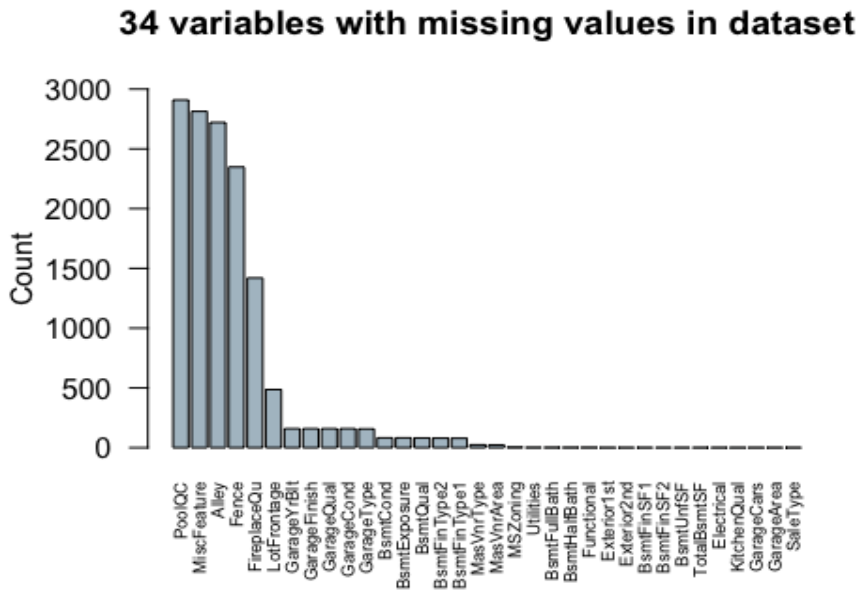
```
4 4 ...
##  $ Utilities    : Factor w/ 2 levels "AllPub","NoSeWa": 1 1 1 1 1 1 1 1 1
1 ...
##  $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5
1 5 1 ...
##  $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1
1 ...
##  $ Neighborhood : Factor w/ 25 levels "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 1
3 13 13 7 4 9 ...
##  $ Exterior2nd  : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 16 1
4 14 14 7 16 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 N
A 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    : num  208500 181500 223500 140000 250000 ...
```

Cleaning the Data and converting to factors

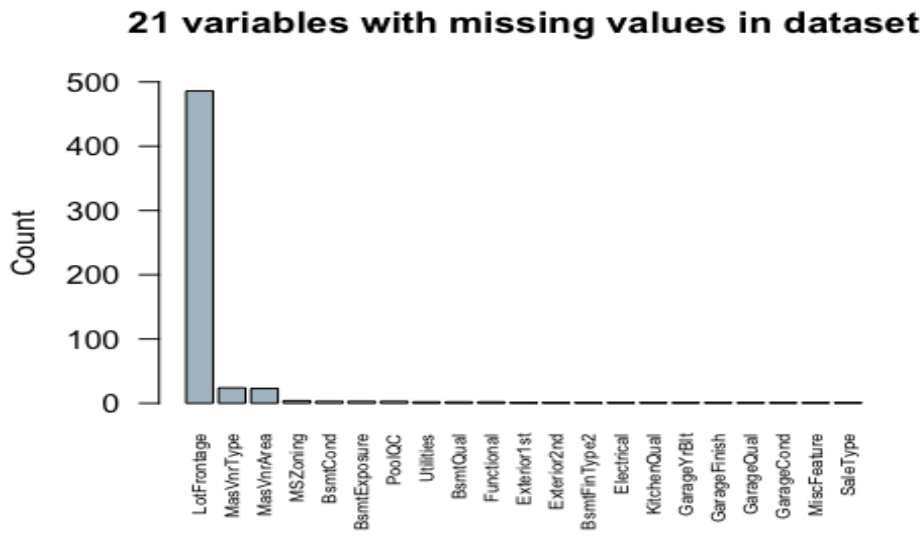
```
#finding how many variables with missing values are in the dataset
options(repr.plot.width=6, repr.plot.height=5)
cMiss = function(x){sum(is.na(x))}
CM <- sort(apply(df,2,cMiss),decreasing=T);
barplot(CM[CM!=0],
        las=2,
        cex.names=0.6,
        ylab="Count",
        ylim=c(0,3000),
        horiz=F,
        col="#AFC0CB",
        main=paste(toString(sum(CM!=0)), "variables with missing values in da
taset"))
```



```r
dfClean <-function(df)
{
  # Pool Variable: If PoolQC = NA and PoolArea = 0 , assign factor NoPool
  df$PoolQC <- as.character(df$PoolQC)
  df$PoolQC[df$PoolArea %in% c(0,NA) & is.na(df$PoolQC)] <- "NoPool"
  df$PoolQC <- as.factor(df$PoolQC)
  # MiscFeature Variable: If MiscFeature = NA and MiscVal = 0, assign factor
None
  df$MiscFeature <- as.character(df$MiscFeature)
  df$MiscFeature[df$MiscVal %in% c(0,NA) & is.na(df$MiscFeature)] <- "None"
  df$MiscFeature <- as.factor(df$MiscFeature)
  # Alley Variable: If Alley = NA, assign factor NoAccess
  df$Alley <- as.character(df$Alley)
  df$Alley[is.na(df$Alley)] <- "NoAccess"
  df$Alley <- as.factor(df$Alley)
  # Fence Variable: If Fence = NA, assign factor NoFence
  df$Fence <- as.character(df$Fence)
  df$Fence[is.na(df$Fence)] <- "NoFence"
  df$Fence <- as.factor(df$Fence)
  # FireplaceQu Variable: If FireplaceQu = NA and Fireplaces = 0 , assign fac
tor NoFirePlace
  df$FireplaceQu <- as.character(df$FireplaceQu)
  df$FireplaceQu[df$Fireplaces %in% c(0,NA) & is.na(df$FireplaceQu)] <- "NoFi
rePlace"
  df$FireplaceQu <- as.factor(df$FireplaceQu)
  # GarageYrBlt Variable: If GarageYrBlt = NA and GarageArea = 0 assign facto
r NoGarage
  df$GarageYrBlt <- as.character(df$GarageYrBlt)
  df$GarageYrBlt[df$GarageArea %in% c(0,NA) & is.na(df$GarageYrBlt)] <- "NoGa
rage"
  df$GarageYrBlt <- as.factor(df$GarageYrBlt)
  # GarageFinish Variable: If GarageFinish = NA and GarageArea = 0 assign fac
tor NoGarage
  df$GarageFinish <- as.character(df$GarageFinish)
  df$GarageFinish[df$GarageArea %in% c(0,NA) & is.na(df$GarageFinish)] <- "No
Garage"
  df$GarageFinish <- as.factor(df$GarageFinish)
  # GarageQual Variable: If GarageQual = NA and GarageArea = 0 assign factor
NoGarage
  df$GarageQual <- as.character(df$GarageQual)
  df$GarageQual[df$GarageArea %in% c(0,NA) & is.na(df$GarageQual)] <- "NoGara
ge"
  df$GarageQual <- as.factor(df$GarageQual)
  # GarageCond Variable: If GarageCond = NA and GarageArea = 0 assign factor
NoGarage
  df$GarageCond <- as.character(df$GarageCond)
  df$GarageCond[df$GarageArea %in% c(0,NA) & is.na(df$GarageCond)] <- "NoGara
ge"
  df$GarageCond <- as.factor(df$GarageCond)
  # GarageType Variable: If GarageType = NA and GarageArea = 0 assign factor
```

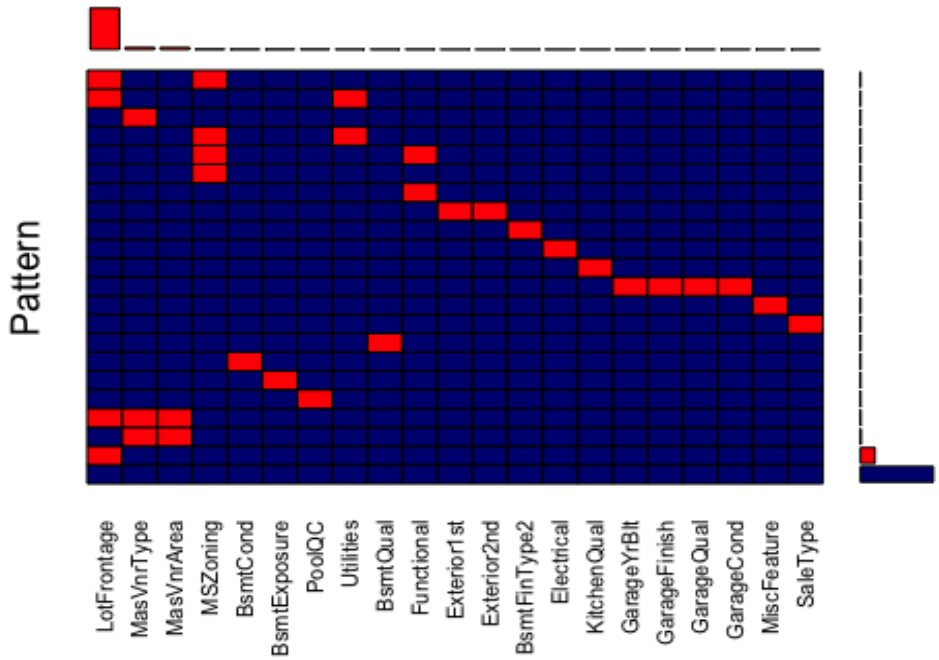
```
NoGarage
  df$GarageType <- as.character(df$GarageType)
  df$GarageType[df$GarageArea %in% c(0,NA) & is.na(df$GarageType)] <- "NoGara
ge"
  df$GarageType <- as.factor(df$GarageType)

  df$GarageArea[is.na(df$GarageArea) & df$GarageCars %in% c(0,NA)] <- 0
  df$GarageCars[is.na(df$GarageCars) & df$GarageArea %in% c(0,NA)] <- 0

  # BsmtFullBath Variable: If BsmtFullBath = NA and TotalBsmtSF = 0 assign 0
  df$BsmtFullBath[df$TotalBsmtSF %in% c(0,NA) & is.na(df$BsmtFullBath)] <- 0
  # BsmtHalfBath Variable: If BsmtHalfBath = NA and TotalBsmtSF = 0 assign 0
  df$BsmtHalfBath[df$TotalBsmtSF %in% c(0,NA) & is.na(df$BsmtHalfBath)] <- 0

  # BsmtFinSF1 Variable: If BsmtFinSF1 = NA and TotalBsmtSF = 0 assign 0
  df$BsmtFinSF1[df$TotalBsmtSF %in% c(0,NA) & is.na(df$BsmtFinSF1)] <- 0
  # BsmtFinSF2 Variable: If BsmtFinSF2 = NA and TotalBsmtSF = 0 assign 0
  df$BsmtFinSF2[df$TotalBsmtSF %in% c(0,NA) & is.na(df$BsmtFinSF2)] <- 0
  # BsmtUnfSF Variable: If BsmtUnfSF = NA and TotalBsmtSF = 0 assign 0
  df$BsmtUnfSF[df$TotalBsmtSF %in% c(0,NA) & is.na(df$BsmtUnfSF)] <- 0
  # TotalBsmtSF Variable: If TotalBsmtSF = NA and TotalBsmtSF = 0 assign 0
  df$TotalBsmtSF[df$TotalBsmtSF %in% c(0,NA) & is.na(df$TotalBsmtSF)] <- 0

  # BsmtQual Variable: If BsmtQual = NA and TotalBsmtSF = 0 assign factor NoB
asement
  df$BsmtQual <- as.character(df$BsmtQual)
  df$BsmtQual[df$TotalBsmtSF %in% c(0,NA) & is.na(df$BsmtQual)] <- "NoBasemen
t"
  df$BsmtQual <- as.factor(df$BsmtQual)
  # BsmtFinType1 Variable: If BsmtFinType1 = NA and TotalBsmtSF = 0 assign fa
ctor NoBasement
  df$BsmtFinType1 <- as.character(df$BsmtFinType1)
  df$BsmtFinType1[df$TotalBsmtSF %in% c(0,NA) & is.na(df$BsmtFinType1)] <- "N
oBasement"
  df$BsmtFinType1 <- as.factor(df$BsmtFinType1)
  # BsmtFinType2 Variable: If BsmtFinType2 = NA and TotalBsmtSF = 0 assign fa
ctor NoBasement
  df$BsmtFinType2 <- as.character(df$BsmtFinType2)
  df$BsmtFinType2[df$TotalBsmtSF %in% c(0,NA) & is.na(df$BsmtFinType2)] <- "N
oBasement"
  df$BsmtFinType2 <- as.factor(df$BsmtFinType2)
  # BsmtExposure Variable: If BsmtExposure = NA and TotalBsmtSF = 0 assign fa
ctor NoBasement
  df$BsmtExposure <- as.character(df$BsmtExposure)
  df$BsmtExposure[df$TotalBsmtSF %in% c(0,NA) & is.na(df$BsmtExposure)] <- "N
oBasement"
  df$BsmtExposure <- as.factor(df$BsmtExposure)
  # BsmtCond Variable: If BsmtCond = NA and TotalBsmtSF = 0 assign factor NoB
asement
```

```
  df$BsmtCond <- as.character(df$BsmtCond)
  df$BsmtCond[df$TotalBsmtSF %in% c(0,NA) & is.na(df$BsmtCond)] <- "NoBasemen
t"
  df$BsmtCond <- as.factor(df$BsmtCond)
  return(df)
}

df <- dfClean(df)

PM <- sort(apply(df,2,cMiss),decreasing=T);
barplot(PM[PM!=0],
        las=2,
        cex.names=0.6,
        ylab="Count",
        ylim=c(0,500),
        horiz=F,
        col="#AFC0CB",
        main=paste(toString(sum(PM!=0)), "variables with missing values in da
taset"))
```

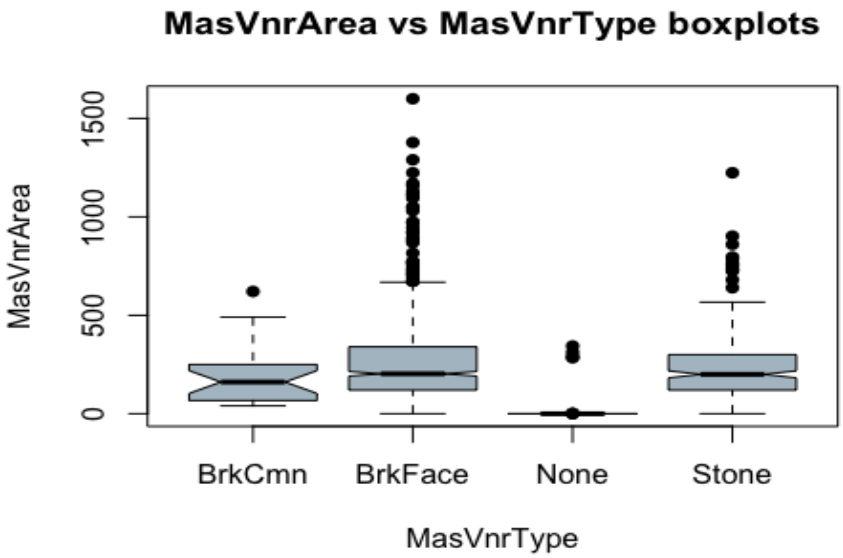



```
#That certainly helped a little bit. Let's see if there's a pattern to the re
maining missing data.

library(VIM);

## Loading required package: colorspace

## Loading required package: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statis
tikat/VIM/issues
```

```
##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep

data = df[, names(PM[PM!=0])];
aggr_plot <- aggr(data,
                  col=c('navyblue','red'),
                  bars=T,
                  numbers=T,
                  combined = T,
                  labels=names(data),
                  cex.axis=.7,
                  gap=3,
                  ylab=c("Pattern"),
                  cex.numbers=0.74)

## Warning in plot.aggr(res, ...): not enough horizontal space to display
## frequencies
```




```
#MasVnrType and MasVnrArea

plot(df[,c("MasVnrType","MasVnrArea")],
     pch=16,
     notch=TRUE,
     main="MasVnrArea vs MasVnrType boxplots",
     col="#AFC0CB")
```



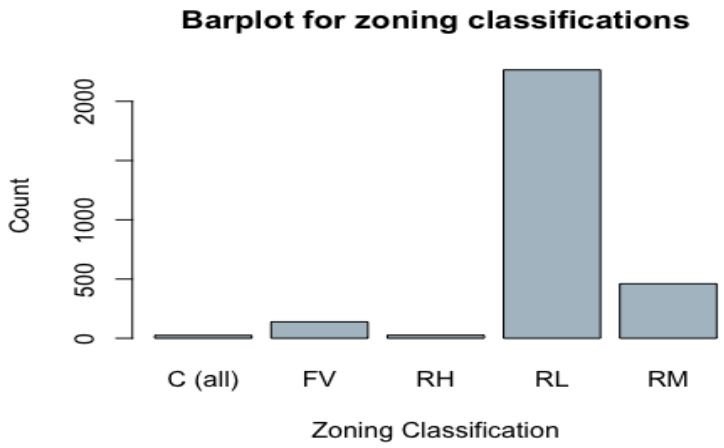

```
df[ (is.na(df$MasVnrType) | is.na(df$MasVnrArea)) ,c("MasVnrType","MasVnrArea
")]
```

```
##      MasVnrType MasVnrArea
## 235        <NA>         NA
## 530        <NA>         NA
## 651        <NA>         NA
## 937        <NA>         NA
## 974        <NA>         NA
## 978        <NA>         NA
## 1244       <NA>         NA
## 1279       <NA>         NA
## 1692       <NA>         NA
## 1707       <NA>         NA
## 1883       <NA>         NA
## 1993       <NA>         NA
## 2005       <NA>         NA
## 2042       <NA>         NA
## 2312       <NA>         NA
## 2326       <NA>         NA
## 2341       <NA>         NA
## 2350       <NA>         NA
## 2369       <NA>         NA
## 2593       <NA>         NA
## 2611       <NA>        198
## 2658       <NA>         NA
## 2687       <NA>         NA
## 2863       <NA>         NA
```

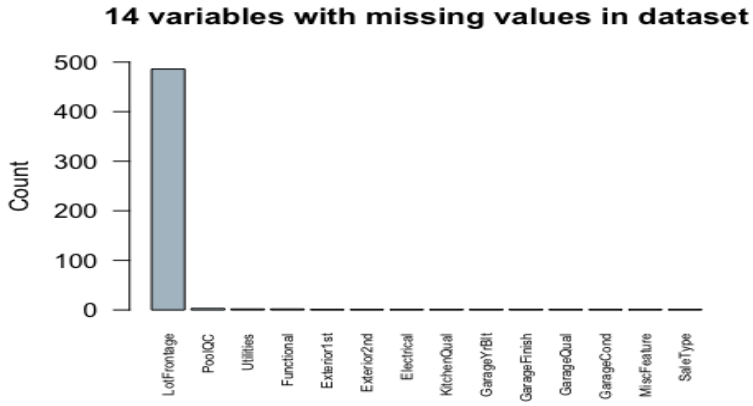
```
summary(df[ !(is.na(df$MasVnrType) | is.na(df$MasVnrArea)) ,c("MasVnrType","M
asVnrArea")])
```

```
##    MasVnrType     MasVnrArea
##  BrkCmn :  25   Min.   :   0.0
##  BrkFace: 879   1st Qu.:   0.0
##  None   :1742   Median :   0.0
##  Stone  : 249   Mean   : 102.2
##                 3rd Qu.: 164.0
##                 Max.   :1600.0
```

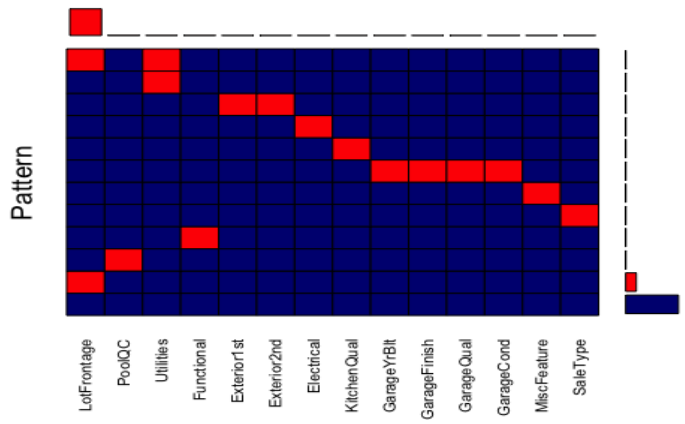
```
df$MasVnrType <- as.character(df$MasVnrType)
df$MasVnrType[is.na(df$MasVnrType)] <- "None"
df$MasVnrType <- as.factor(df$MasVnrType)
df$MasVnrArea[is.na(df$MasVnrArea)] <- 0

#MSZoning
plot(df$MSZoning,
     col="#AFC0CB",
     xlab="Zoning Classification",
     ylab = "Count",
     main = "Barplot for zoning classifications")
```




```
PM <- sort(apply(df,2,cMiss),decreasing=T);
barplot(PM[PM!=0],
        las=2,
        cex.names=0.6,
        ylab="Count",
        ylim=c(0,500),
        horiz=F,
        col="#AFC0CB",
        main=paste(toString(sum(PM!=0)), "variables with missing values in
dataset"))
```

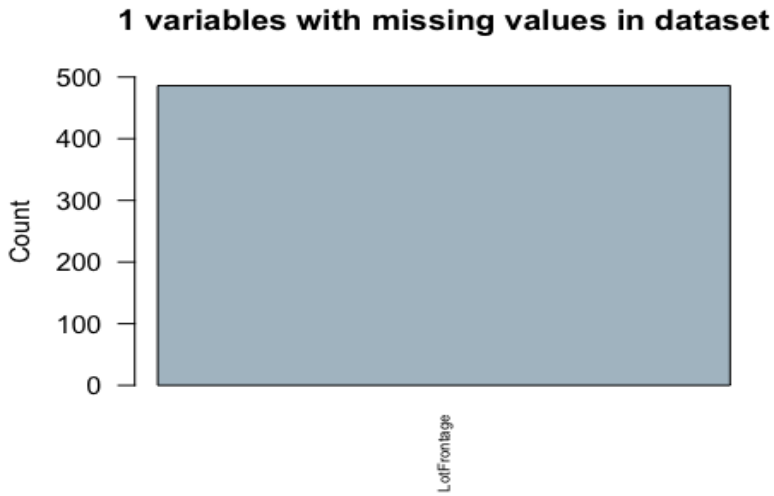



```
data = df[, names(PM[PM!=0])];
aggr_plot <- aggr(data,
                  col=c('navyblue','red'),
                  bars=T,
                  numbers=T,
                  combined = T,
                  labels=names(data),
                  cex.axis=.7,
                  gap=3,
                  ylab=c("Pattern"),
                  cex.numbers=0.74)

## Warning in plot.aggr(res, ...): not enough horizontal space to display
## frequencies
```

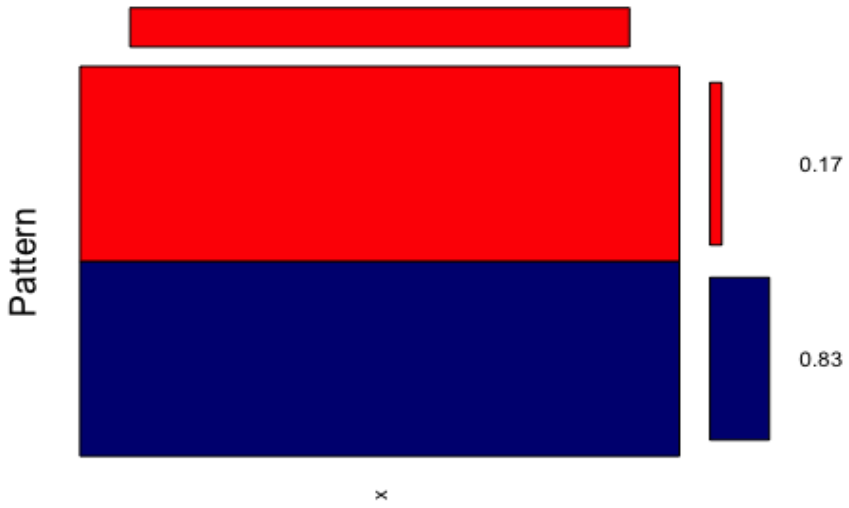



```
#The rest
fillMiss<- function(x)
{
  ux <- unique(x[!is.na(x)])
  x <- as.character(x)
  mode <- ux[which.max(tabulate(match(x[!is.na(x)], ux)))]
  x[is.na(x)] <- as.character(mode)
```

```
  x <- as.factor(x)
  return(x)
}
df[,sapply(df,function(x){!(is.numeric(x))}) ]<-as.data.frame(apply(df[,sappl
y(df,function(x){!(is.numeric(x))}) ],2,fillMiss))

PM <- sort(apply(df,2,cMiss),decreasing=T);
barplot(PM[PM!=0],
        las=2,
        cex.names=0.6,
        ylab="Count",
        ylim=c(0,500),
        horiz=F,
        col="#AFC0CB",
        main=paste(toString(sum(PM!=0)), "variables with missing values in da
taset"))
```

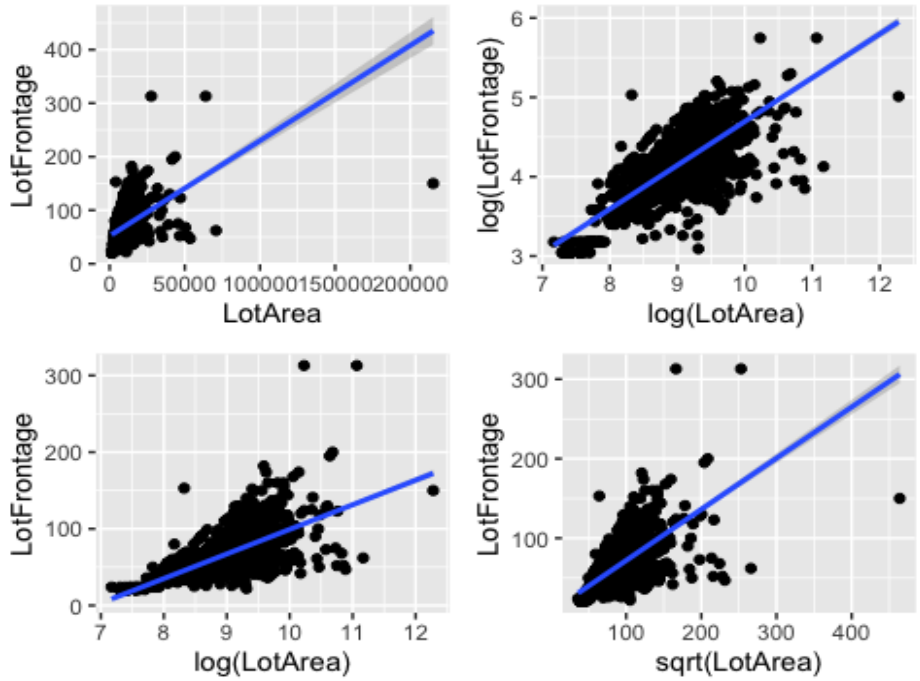



```
data = df[, names(PM[PM!=0])];
aggr_plot <- aggr(data,
                  col=c('navyblue','red'),
                  bars=T,
                  numbers=T,
                  combined = T,
                  labels=names(data),
                  cex.axis=.7,
                  gap=3,
                  ylab=c("Pattern"),
                  cex.numbers=0.74)
```




```
#LotFrontage Imputation

multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)
  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)
  numPlots = length(plots)
  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }
  if (numPlots==1) {
    print(plots[[1]])
  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))
    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))
      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
p1<-ggplot(df, aes(LotArea, LotFrontage)) + geom_point() + geom_smooth(method
= "lm", se = T)
p2<-ggplot(df, aes(log(LotArea), LotFrontage)) + geom_point() + geom_smooth(m
```

```
ethod = "lm", se = T)
p3<-ggplot(df, aes(log(LotArea), log(LotFrontage))) + geom_point() + geom_smo
oth(method = "lm", se = T)
p4<-ggplot(df, aes(sqrt(LotArea), LotFrontage)) + geom_point() + geom_smooth(
method = "lm", se = T)
multiplot(p1, p2, p3, p4, cols=2)
```




```
library(outliers)

##
## Attaching package: 'outliers'

## The following object is masked from 'package:randomForest':
##
##     outlier

chisq.out.test(df$LotArea,opposite=F)

##
##  chi-squared test for outlier
##
## data:  df$LotArea
## X-squared = 676.1, p-value < 2.2e-16
## alternative hypothesis: highest value 215245 is an outlier

chisq.out.test(df$LotFrontage,opposite=F)

##
##  chi-squared test for outlier
##
```

```
## data:  df$LotFrontage
## X-squared = 108.97, p-value < 2.2e-16
## alternative hypothesis: highest value 313 is an outlier
```

```
chisq.out.test(df$LotArea,opposite=T)
```

```
##
##  chi-squared test for outlier
##
## data:  df$LotArea
## X-squared = 1.2643, p-value = 0.2608
## alternative hypothesis: lowest value 1300 is an outlier
```

```
chisq.out.test(df$LotFrontage,opposite=T)
```

```
##
##  chi-squared test for outlier
##
## data:  df$LotFrontage
## X-squared = 4.2817, p-value = 0.03853
## alternative hypothesis: lowest value 21 is an outlier
```

```
grubbs.test(df$LotArea,type=11)
```

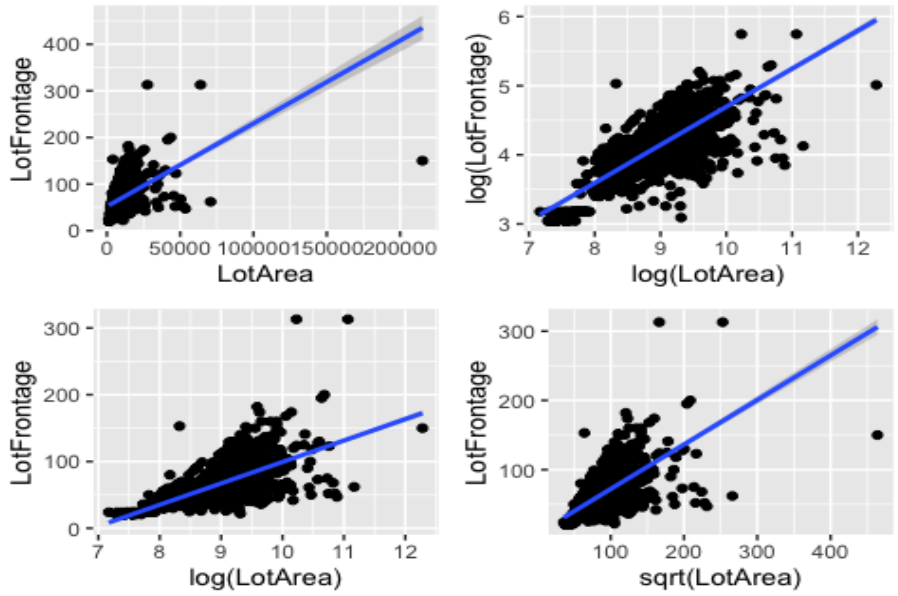
```
##
##  Grubbs test for two opposite outliers
##
## data:  df$LotArea
## G = 27.12630, U = 0.76779, p-value < 2.2e-16
## alternative hypothesis: 1300 and 215245 are outliers
```

```
grubbs.test(df$LotFrontage,type=11)
```

```
##
##  Grubbs test for two opposite outliers
##
## data:  df$LotFrontage
## G = 12.50808, U = 0.95342, p-value < 2.2e-16
## alternative hypothesis: 21 and 313 are outliers
```

```
p1<-ggplot(df  , aes(LotArea, LotFrontage)) + geom_point() + geom_smooth(meth
od = "lm", se = T)
p2<-ggplot(df, aes(log(LotArea), LotFrontage)) + geom_point() + geom_smooth(m
ethod = "lm", se = T)
p3<-ggplot(df, aes(log(LotArea), log(LotFrontage))) + geom_point() + geom_smo
oth(method = "lm", se = T)
p4<-ggplot(df, aes(sqrt(LotArea), LotFrontage)) + geom_point() + geom_smooth(
method = "lm", se = T)
multiplot(p1, p2, p3, p4, cols=2)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
cor(as.numeric(df$LotArea),as.numeric(df$LotFrontage),use="complete.obs")

## [1] 0.4898956

cor(log(as.numeric(df$LotArea)),log(as.numeric(df$LotFrontage)),use="complete
.obs")

## [1] 0.7662858

cor(log(as.numeric(df$LotArea)),as.numeric(df$LotFrontage),use="complete.obs"
)

## [1] 0.6835123

cor(sqrt(as.numeric(df$LotArea)),as.numeric(df$LotFrontage),use="complete.obs
")

## [1] 0.647658

PredModel <- ~-1+log(LotArea)+Street+LotShape+LandContour+LotConfig+LandSlope
+Neighborhood+BldgType
dpredict <- xgb.DMatrix(data = sparse.model.matrix(PredModel,data=df[is.na(df
$LotFrontage),]))
#LotFrontagePredict <- exp(predict(md,dpredict))
#df$LotFrontage[is.na(df$LotFrontage)] <- LotFrontagePredict


str(df)

## 'data.frame':    2919 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 7
420 ...
##  $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ..
.
##  $ Alley        : Factor w/ 3 levels "Grvl","NoAccess",..: 2 2 2 2 2 2 2 2
2 2 ...
##  $ 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 1
3 13 13 7 4 9 ...
##  $ Exterior2nd  : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 16 1
4 14 14 7 16 9 ...
##  $ MasVnrType   : Factor w/ 4 levels "BrkCmn","BrkFace",..: 2 3 2 3 2 3 4
4 3 3 ...
##  $ MasVnrArea   : num  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/ 5 levels "Ex","Fa","Gd",..: 3 3 3 5 3 3 1 3 5
5 ...
##  $ BsmtCond     : Factor w/ 5 levels "Fa","Gd","NoBasement",..: 5 5 5 2 5
5 5 5 5 5 ...
##  $ BsmtExposure : Factor w/ 5 levels "Av","Gd","Mn",..: 4 2 3 4 1 4 1 3 4
4 ...
##  $ BsmtFinType1 : Factor w/ 7 levels "ALQ","BLQ","GLQ",..: 3 1 3 1 3 3 3 1
7 3 ...
##  $ BsmtFinSF1   : num  706 978 486 216 655 ...
##  $ BsmtFinType2 : Factor w/ 7 levels "ALQ","BLQ","GLQ",..: 7 7 7 7 7 7 7 2
7 7 ...
##  $ BsmtFinSF2   : num  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : num  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : num  856 1262 920 756 1145 ...
##  $ 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 : num  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : num  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/ 6 levels "Ex","Fa","Gd",..: 4 6 6 3 6 4 3 6 6
6 ...
##  $ GarageType   : Factor w/ 7 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2
6 2 ...
##  $ GarageYrBlt  : Factor w/ 104 levels "1895","1896",..: 95 68 93 90 92 85
96 65 24 32 ...
##  $ GarageFinish : Factor w/ 4 levels "Fin","NoGarage",..: 3 3 3 4 3 4 3 3
4 3 ...
##  $ GarageCars   : num  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : num  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : Factor w/ 6 levels "Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 2
3 ...
```

```
##  $ GarageCond   : Factor w/ 6 levels "Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 6
6 ...
##  $ PavedDrive   : Factor w/ 3 levels "N","P","Y": 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/ 4 levels "Ex","Fa","Gd",..: 4 4 4 4 4 4 4 4 4
4 ...
##  $ Fence        : Factor w/ 5 levels "GdPrv","GdWo",..: 5 5 5 5 5 3 5 5 5
5 ...
##  $ MiscFeature  : Factor w/ 5 levels "Gar2","None",..: 2 2 2 2 2 4 2 4 2 2
...
##  $ 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    : num  208500 181500 223500 140000 250000 ...
```

Converting the dependent Variable to factors

```
#Converting SalePrice to factors
df$SalePrice <- as.factor(df$SalePrice)
str(df)
```

```
## 'data.frame':    2919 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 7
420 ...
##  $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ..
.
##  $ Alley        : Factor w/ 3 levels "Grvl","NoAccess",..: 2 2 2 2 2 2 2 2
2 2 ...
##  $ 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 1
3 13 13 7 4 9 ...
##  $ Exterior2nd  : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 16 1
4 14 14 7 16 9 ...
##  $ MasVnrType   : Factor w/ 4 levels "BrkCmn","BrkFace",..: 2 3 2 3 2 3 4
4 3 3 ...
##  $ MasVnrArea   : num  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/ 5 levels "Ex","Fa","Gd",..: 3 3 3 5 3 3 1 3 5
5 ...
##  $ BsmtCond     : Factor w/ 5 levels "Fa","Gd","NoBasement",..: 5 5 5 2 5
5 5 5 5 5 ...
##  $ BsmtExposure : Factor w/ 5 levels "Av","Gd","Mn",..: 4 2 3 4 1 4 1 3 4
4 ...
##  $ BsmtFinType1 : Factor w/ 7 levels "ALQ","BLQ","GLQ",..: 3 1 3 1 3 3 3 1
7 3 ...
##  $ BsmtFinSF1   : num  706 978 486 216 655 ...
##  $ BsmtFinType2 : Factor w/ 7 levels "ALQ","BLQ","GLQ",..: 7 7 7 7 7 7 7 2
7 7 ...
##  $ BsmtFinSF2   : num  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : num  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : num  856 1262 920 756 1145 ...
##  $ 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 : num  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : num  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/ 6 levels "Ex","Fa","Gd",..: 4 6 6 3 6 4 3 6 6
6 ...
##  $ GarageType   : Factor w/ 7 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2
6 2 ...
##  $ GarageYrBlt  : Factor w/ 104 levels "1895","1896",..: 95 68 93 90 92 85
96 65 24 32 ...
##  $ GarageFinish : Factor w/ 4 levels "Fin","NoGarage",..: 3 3 3 4 3 4 3 3
4 3 ...
##  $ GarageCars   : num  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : num  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : Factor w/ 6 levels "Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 2
3 ...
##  $ GarageCond   : Factor w/ 6 levels "Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 6
6 ...
##  $ 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/ 4 levels "Ex","Fa","Gd",..: 4 4 4 4 4 4 4 4 4
4 ...
##  $ Fence        : Factor w/ 5 levels "GdPrv","GdWo",..: 5 5 5 5 5 3 5 5 5
5 ...
##  $ MiscFeature  : Factor w/ 5 levels "Gar2","None",..: 2 2 2 2 2 4 2 4 2 2
...
```

```
##  $ 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    : Factor w/ 664 levels "-1","34900","35311",..: 414 341 44
4 196 496 205 575 392 153 115 ...
```

```
xtabs(~ SalePrice + MSZoning , data=df)
```

```
##          MSZoning
## SalePrice C (all)   FV   RH   RL   RM
##    -1          15   74   10 1118  242
##    34900        1    0    0    0    0
##    35311        1    0    0    0    0
##    37900        0    0    0    0    1
##    39300        0    0    0    1    0
##    40000        1    0    0    0    0
##    52000        0    0    0    1    0
##    52500        0    0    0    0    1
##    55000        0    0    0    0    2
##    55993        1    0    0    0    0
.

.
##    466500       0    0    0    1    0
##    475000       0    0    0    0    1
##    485000       0    0    0    1    0
##    501837       0    0    0    1    0
##    538000       0    0    0    1    0
##    555000       0    0    0    1    0
##    556581       0    0    0    1    0
##    582933       0    0    0    1    0
##    611657       0    0    0    1    0
##    625000       0    0    0    1    0
##    745000       0    0    0    1    0
##    755000       0    0    0    1    0
```

```
xtabs(~ SalePrice + Street  , data=df)
```

```
##          Street
## SalePrice Grvl Pave
##    -1        6 1453
##    34900     0    1
##    35311     0    1
##    37900     0    1
##    39300     0    1
```

```
.

.
##    611657    0    1
##    625000    0    1
##    745000    0    1
##    755000    0    1
```

```
xtabs(~ SalePrice + Neighborhood , data=df)
```

```
##           Neighborhood
## SalePrice Blmngtn Blueste BrDale BrkSide ClearCr CollgCr Crawfor Edwards
##    -1          11       8     14       50      16     117       52       94
##    34900        0       0      0        0       0       0        0        0
##    35311        0       0      0        0       0       0        0        0
##    37900        0       0      0        0       0       0        0        0
##    39300        0       0      0        1       0       0        0        0
.

.
##    556581       0       0      0        0       0       0        0        0
##    582933       0       0      0        0       0       0        0        0
##    611657       0       0      0        0       0       0        0        0
##    625000       0       0      0        0       0       0        0        0
##    745000       0       0      0        0       0       0        0        0
##    755000       0       0      0        0       0       0        0        0
##           Neighborhood
## SalePrice Gilbert IDOTRR MeadowV Mitchel NAmes NoRidge NPkVill NridgHt NWA
mes
##    -1          86     56      20      65   218      30      14      89
58
##    34900        0      1       0       0     0       0       0       0
0
##    35311        0      1       0       0     0       0       0       0
0
##    501837       0      0       0       0     0       0       0       0
0

.

.
##    538000       0      0       0       0     0       0       0       0
0
##    555000       0      0       0       0     0       0       0       1
0
##    556581       0      0       0       0     0       0       0       0
0
##    582933       0      0       0       0     0       0       0       1
0
##    611657       0      0       0       0     0       0       0       1
0
```

```
##    625000       0      0       0       0     0       1       0       0
0
##    745000       0      0       0       0     0       1       0       0
0
##    755000       0      0       0       0     0       1       0       0
0
##           Neighborhood
## SalePrice OldTown Sawyer SawyerW Somerst StoneBr SWISU Timber Veenker
##    -1         126     77      66      96      26    23      34      13
##    34900        0      0       0       0       0     0       0       0
##    35311        0      0       0       0       0     0       0       0
##    37900        1      0       0       0       0     0       0       0
##    39300        0      0       0       0       0     0       0       0
##    40000        0      0       0       0       0     0       0       0
##    52000        0      0       0       0       0     0       0       0
.

.
##    538000       0      0       0       0       1     0       0       0
##    555000       0      0       0       0       0     0       0       0
##    556581       0      0       0       0       1     0       0       0
##    582933       0      0       0       0       0     0       0       0
##    611657       0      0       0       0       0     0       0       0
##    625000       0      0       0       0       0     0       0       0
##    745000       0      0       0       0       0     0       0       0
##    755000       0      0       0       0       0     0       0       0

xtabs(~ SalePrice + BldgType , data=df)

##           BldgType
## SalePrice 1Fam 2fmCon Duplex Twnhs TwnhsE
##    -1     1205     31     57    53    113
##    34900     1      0      0     0      0
##    35311     1      0      0     0      0
##    37900     1      0      0     0      0
##    485000    1      0      0     0      0
##    501837    1      0      0     0      0
##    538000    1      0      0     0      0

.

.
##    555000    1      0      0     0      0
##    556581    1      0      0     0      0
##    582933    1      0      0     0      0
##    611657    1      0      0     0      0
##    625000    1      0      0     0      0
##    745000    1      0      0     0      0
##    755000    1      0      0     0      0

xtabs(~ SalePrice + SaleType  , data=df)
```

```
##           SaleType
## SalePrice  COD  Con ConLD ConLI ConLw  CWD  New  Oth   WD
##    -1       44    3    17     4     3    8  117    4 1259
##    34900     0    0     0     0     0    0    0    0    1
##    35311     0    0     0     0     0    0    0    0    1
##    37900     0    0     0     0     0    0    0    0    1
##    39300     0    0     0     0     0    0    0    0    1

.

.

##    485000    0    0     0     0     0    0    1    0    0
##    501837    0    0     0     0     0    0    1    0    0
##    538000    0    0     0     0     0    0    0    0    1
##    555000    0    0     0     0     0    0    0    0    1
##    556581    0    0     0     0     0    0    1    0    0
##    582933    0    0     0     0     0    0    1    0    0
##    611657    0    0     0     0     0    0    1    0    0
##    625000    0    0     0     0     0    0    0    0    1
##    745000    0    0     0     0     0    0    0    0    1
##    755000    0    0     0     0     0    0    0    0    1
```

```
xtabs(~ SalePrice + Street  , data=df)
```

```
##           Street
## SalePrice Grvl Pave
##    -1        6 1453
##    34900     0    1
##    35311     0    1
##    37900     0    1
##    39300     0    1
##    556581    0    1
##    582933    0    1

.

.
##    611657    0    1
##    625000    0    1
##    745000    0    1
##    755000    0    1
```

```
logistic_simple <- glm(SalePrice ~ Street, data=df, family="binomial")
summary(logistic_simple)
```

```
##
## Call:
## glm(formula = SalePrice ~ Street, family = "binomial", data = df)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
```

```
## -1.178  -1.178   1.177   1.177   1.177
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.597e-13  5.774e-01   0.000    1.000
## StreetPave   6.880e-04  5.785e-01   0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4046.6  on 2918  degrees of freedom
## Residual deviance: 4046.6  on 2917  degrees of freedom
## AIC: 4050.6
##
## Number of Fisher Scoring iterations: 2
```

```
ll.null <- logistic_simple$null.deviance/-2
ll.proposed <- logistic_simple$deviance/-2
ll.null
```

```
## [1] -2023.296
```

```
ll.proposed
```

```
## [1] -2023.296
```

```
(ll.null - ll.proposed) / ll.null
```

```
## [1] 3.494721e-10
```

```
1 - pchisq(2*(ll.proposed - ll.null), df=1)
```

```
## [1] 0.9990512
```

```
1 - pchisq((logistic_simple$null.deviance - logistic_simple$deviance), df=1)
```

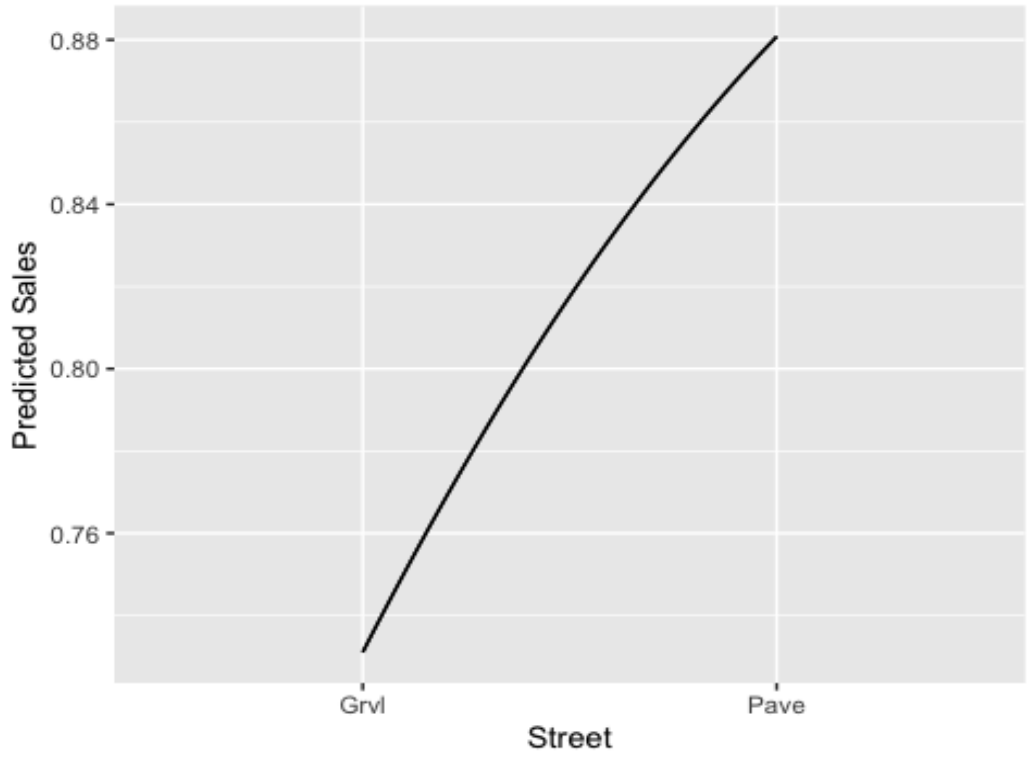
```
## [1] 0.9990512
```

```
predicted.data <- data.frame(probability.of.Street=logistic_simple$fitted.val
ues,SalePrice=df$SalePrice)
predicted.data
```

```
##      probability.of.Street SalePrice
## 1                 0.500172    208500
## 2                 0.500172    181500
## 3                 0.500172    223500
## 4                 0.500172    140000
## 5                 0.500172    250000
## 6                 0.500172    143000
## 7                 0.500172    307000
## 8                 0.500172     2e+05
## 9                 0.500172    129900
## 10                0.500172    118000
```

```
## 11                0.500172    129500
## 12                0.500172    345000
## 13                0.500172    144000
## 14                0.500172    279500
## 15                0.500172    157000

.

.
## 2914              0.500172        -1
## 2915              0.500172        -1
## 2916              0.500172        -1
## 2917              0.500172        -1
## 2918              0.500172        -1
## 2919              0.500172        -1
```

```
ggplot(data=predicted.data, aes(x= df$Street, y=probability.of.Street)) +
  stat_function(fun = function(x) 1/(1 + exp(-x)), geom = "line") +
  xlab("Street") +
  ylab("Predicted Sales")
```




```
xtabs(~ probability.of.Street + SalePrice, data=predicted.data)
```

```
##                       SalePrice
## probability.of.Street   -1 34900 35311 37900 39300 40000 52000 52500 55000
##     0.49999999999981     6     0     0     0     0     0     0     0     0
##     0.500171998622917 1453     1     1     1     1     1     1     1     2
##                       SalePrice
```

```
## probability.of.Street 55993 58500 60000 61000 62383 64500 66500 67000 6840
0
##     0.49999999999981      1     0     0     0     0     0     0     0
0
##     0.500171998622917     0     1     3     1     1     1     1     2
1
##                       SalePrice
## probability.of.Street 68500 72500 73000 75000 75500 76000 76500 78000 7900
0
##     0.49999999999981      0     0     0     0     0     0     0     0
0
##     0.500171998622917     1     1     1     1     1     1     1     1
3
.

.
##                       SalePrice
## probability.of.Street 394432 394617 395000 395192 402000 402861 403000 410
000
##     0.49999999999981       0      0      0      0      0      0      0
0
```

```
logistic <- glm(SalePrice ~ Street + MSZoning, data=df, family="binomial")
summary(logistic)
```

```
##
## Call:
## glm(formula = SalePrice ~ Street + MSZoning, family = "binomial",
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3824  -1.1896   0.9854   1.1653   1.3709
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.2536     0.6326  -0.401    0.688
## StreetPave   -0.1904     0.6078  -0.313    0.754
## MSZoningFV    0.3144     0.4597   0.684    0.494
## MSZoningRH    0.9141     0.5873   1.556    0.120
## MSZoningRL    0.4728     0.4288   1.102    0.270
## MSZoningRM    0.3388     0.4365   0.776    0.438
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4046.6  on 2918  degrees of freedom
## Residual deviance: 4041.8  on 2913  degrees of freedom
## AIC: 4053.8
##
## Number of Fisher Scoring iterations: 3
```

```r
ll.null <- logistic$null.deviance/-2
ll.proposed <- logistic$deviance/-2
(ll.null - ll.proposed) / ll.null
```

```
## [1] 0.00119056
```

```r
1 - pchisq(2*(ll.proposed - ll.null), df=(length(logistic$coefficients)-1))
```

```
## [1] 0.4385299
```

```r
predicted.data <- data.frame(probability.of.SalePrice=logistic$fitted.values,
SalePrice=df$SalePrice)
predicted.data <- predicted.data[order(predicted.data$probability.of.SalePric
e, decreasing=FALSE),]
predicted.data$rank <- 1:nrow(predicted.data)

ggplot(data=predicted.data, aes(x=rank, y=probability.of.SalePrice)) +
stat_function(fun = function(x) 1/(1 + exp(-x)), geom = "line") +
xlab("Index") +
ylab("Predicted Sales")
```