

House Prices Project

Arjuna Anilkumar, A20446963

11/8/2020

Introduction

This project aims to predict the final price of houses using the Ames housing dataset.

Data description

The Ames Housing dataset was compiled by Dean De Cock for use in data science education. It's an alternative to the Boston Housing dataset and is for data scientists looking for a modernized and expanded version of the often cited Boston Housing dataset.

The Ames housing data contains With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa.

Data Processing

Install packages

```
#install.packages(c("Amelia", "purrrr", "tidyR", "ggplot2", "rpart", "plyr", "corrplot", "RColorBrewer", "ggrepel", "Desci
```

Load data

```
df <- read.table("../data/raw/train.csv", sep = ",", header = T)
head(df)

##   Id MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape LandContour
## 1  1          60      RL       65    8450  Pave <NA>    Reg     Lvl
## 2  2          20      RL       80    9600  Pave <NA>    Reg     Lvl
## 3  3          60      RL       68   11250  Pave <NA>   IR1     Lvl
## 4  4          70      RL       60    9550  Pave <NA>   IR1     Lvl
## 5  5          60      RL       84   14260  Pave <NA>   IR1     Lvl
## 6  6          50      RL       85   14115  Pave <NA>   IR1     Lvl
##   Utilities LotConfig LandSlope Neighborhood Condition1 Condition2 BldgType
## 1    AllPub     Inside      Gtl    CollgCr      Norm      Norm    1Fam
## 2    AllPub      FR2       Gtl    Veenker      Feedr      Norm    1Fam
## 3    AllPub     Inside      Gtl    CollgCr      Norm      Norm    1Fam
## 4    AllPub    Corner      Gtl    Crawfor      Norm      Norm    1Fam
## 5    AllPub      FR2       Gtl    NoRidge      Norm      Norm    1Fam
## 6    AllPub     Inside      Gtl    Mitchel      Norm      Norm    1Fam
##   HouseStyle OverallQual OverallCond YearBuilt YearRemodAdd RoofStyle RoofMatl
## 1    2Story         7           5    2003        2003     Gable  CompShg
## 2    1Story         6           8    1976        1976     Gable  CompShg
```

```

## 3 2Story 7 5 2001 2002 Gable CompShg
## 4 2Story 7 5 1915 1970 Gable CompShg
## 5 2Story 8 5 2000 2000 Gable CompShg
## 6 1.5Fin 5 5 1993 1995 Gable CompShg
## Exterior1st Exterior2nd MasVnrType MasVnrArea ExterQual ExterCond Foundation
## 1 VinylSd VinylSd BrkFace 196 Gd TA PConc
## 2 MetalSd MetalSd None 0 TA TA CBlock
## 3 VinylSd VinylSd BrkFace 162 Gd TA PConc
## 4 Wd Sdng Wd Shng None 0 TA TA BrkTil
## 5 VinylSd VinylSd BrkFace 350 Gd TA PConc
## 6 VinylSd VinylSd None 0 TA TA Wood
## BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinSF1 BsmtFinType2
## 1 Gd TA No GLQ 706 Unf
## 2 Gd TA Gd ALQ 978 Unf
## 3 Gd TA Mn GLQ 486 Unf
## 4 TA Gd No ALQ 216 Unf
## 5 Gd TA Av GLQ 655 Unf
## 6 Gd TA No GLQ 732 Unf
## BsmtFinSF2 BsmtUnfSF TotalBsmtSF Heating HeatingQC CentralAir Electrical
## 1 0 150 856 GasA Ex Y SBrkr
## 2 0 284 1262 GasA Ex Y SBrkr
## 3 0 434 920 GasA Ex Y SBrkr
## 4 0 540 756 GasA Gd Y SBrkr
## 5 0 490 1145 GasA Ex Y SBrkr
## 6 0 64 796 GasA Ex Y SBrkr
## X1stFlrSF X2ndFlrSF LowQualFinsF GrLivArea BsmtFullBath BsmtHalfBath FullBath
## 1 856 854 0 1710 1 0 2
## 2 1262 0 0 1262 0 1 2
## 3 920 866 0 1786 1 0 2
## 4 961 756 0 1717 1 0 1
## 5 1145 1053 0 2198 1 0 2
## 6 796 566 0 1362 1 0 1
## HalfBath BedroomAbvGr KitchenAbvGr KitchenQual TotRmsAbvGrd Functional
## 1 1 3 1 Gd 8 Typ
## 2 0 3 1 TA 6 Typ
## 3 1 3 1 Gd 6 Typ
## 4 0 3 1 Gd 7 Typ
## 5 1 4 1 Gd 9 Typ
## 6 1 1 1 TA 5 Typ
## Fireplaces FireplaceQu GarageType GarageYrBlt GarageFinish GarageCars
## 1 0 <NA> Attchd 2003 RFn 2
## 2 1 TA Attchd 1976 RFn 2
## 3 1 TA Attchd 2001 RFn 2
## 4 1 Gd Detchd 1998 Unf 3
## 5 1 TA Attchd 2000 RFn 3
## 6 0 <NA> Attchd 1993 Unf 2
## GarageArea GarageQual GarageCond PavedDrive WoodDeckSF OpenPorchSF
## 1 548 TA TA Y 0 61
## 2 460 TA TA Y 298 0
## 3 608 TA TA Y 0 42
## 4 642 TA TA Y 0 35
## 5 836 TA TA Y 192 84
## 6 480 TA TA Y 40 30
## EnclosedPorch X3SsnPorch ScreenPorch PoolArea PoolQC Fence MiscFeature
## 1 0 0 0 0 <NA> <NA> <NA>
## 2 0 0 0 0 <NA> <NA> <NA>
## 3 0 0 0 0 <NA> <NA> <NA>
## 4 272 0 0 0 <NA> <NA> <NA>
## 5 0 0 0 0 <NA> <NA> <NA>
## 6 0 320 0 0 <NA> MnPrv Shed
## MiscVal MoSold YrSold SaleType SaleCondition SalePrice

```

```

## 1      0      2  2008     WD    Normal  208500
## 2      0      5  2007     WD    Normal  181500
## 3      0      9  2008     WD    Normal  223500
## 4      0      2  2006     WD   Abnorml 140000
## 5      0     12  2008     WD    Normal  250000
## 6    700     10  2009     WD    Normal  143000

combined <- df

combined <- combined[, !names(combined) %in% "Id"]
str(combined)

## 'data.frame': 1460 obs. of  80 variables:
## $ MSSubClass : int  60 20 60 70 60 50 20 60 50 190 ...
## $ MSZoning   : chr  "RL" "RL" "RL" "RL" ...
## $ 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      : chr  "Pave" "Pave" "Pave" "Pave" ...
## $ Alley       : chr  NA NA NA NA ...
## $ LotShape    : chr  "Reg" "Reg" "IR1" "IR1" ...
## $ LandContour : chr  "Lvl" "Lvl" "Lvl" "Lvl" ...
## $ Utilities   : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
## $ LotConfig   : chr  "Inside" "FR2" "Inside" "Corner" ...
## $ LandSlope   : chr  "Gtl" "Gtl" "Gtl" "Gtl" ...
## $ Neighborhood: chr  "CollgCr" "Veenker" "CollgCr" "Crawfor" ...
## $ Condition1  : chr  "Norm" "Feedr" "Norm" "Norm" ...
## $ Condition2  : chr  "Norm" "Norm" "Norm" "Norm" ...
## $ BldgType    : chr  "1Fam" "1Fam" "1Fam" "1Fam" ...
## $ HouseStyle  : chr  "2Story" "1Story" "2Story" "2Story" ...
## $ 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   : chr  "Gable" "Gable" "Gable" "Gable" ...
## $ RoofMatl   : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
## $ Exterior1st : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...
## $ Exterior2nd : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Shng" ...
## $ MasVnrType  : chr  "BrkFace" "None" "BrkFace" "None" ...
## $ MasVnrArea  : int   196 0 162 0 350 0 186 240 0 0 ...
## $ ExterQual   : chr  "Gd" "TA" "Gd" "TA" ...
## $ ExterCond   : chr  "TA" "TA" "TA" "TA" ...
## $ Foundation  : chr  "PConc" "CBlock" "PConc" "BrkTil" ...
## $ BsmtQual    : chr  "Gd" "Gd" "Gd" "TA" ...
## $ BsmtCond    : chr  "TA" "TA" "TA" "Gd" ...
## $ BsmtExposure: chr  "No" "Gd" "Mn" "No" ...
## $ BsmtFinType1: chr  "GLQ" "ALQ" "GLQ" "ALQ" ...
## $ BsmtFinSF1  : int   706 978 486 216 655 732 1369 859 0 851 ...
## $ BsmtFinType2: chr  "Unf" "Unf" "Unf" "Unf" ...
## $ BsmtFinSF2  : int   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     : chr  "GasA" "GasA" "GasA" "GasA" ...
## $ HeatingQC   : chr  "Ex" "Ex" "Ex" "Gd" ...
## $ CentralAir  : chr  "Y" "Y" "Y" "Y" ...
## $ Electrical  : chr  "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
## $ 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 ...
## $ GrLivArea   : int   1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
## $ BsmtFullBath: int   1 0 1 1 1 1 1 0 1 ...
## $ BsmtHalfBath: int   0 1 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   : chr  "Gd" "TA" "Gd" "Gd" ...
## $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
## $ Functional    : chr  "Typ" "Typ" "Typ" "Typ" ...
## $ Fireplaces    : int  0 1 1 1 1 0 1 2 2 2 ...
## $ FireplaceQu   : chr  NA "TA" "TA" "Gd" ...
## $ GarageType    : chr  "Attchd" "Attchd" "Attchd" "Detchd" ...
## $ GarageYrBlt   : int  2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...
## $ GarageFinish  : chr  "RFn" "RFn" "RFn" "Unf" ...
## $ 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    : chr  "TA" "TA" "TA" "TA" ...
## $ GarageCond    : chr  "TA" "TA" "TA" "TA" ...
## $ PavedDrive   : chr  "Y" "Y" "Y" "Y" ...
## $ 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       : chr  NA NA NA NA ...
## $ Fence         : chr  NA NA NA NA ...
## $ MiscFeature   : chr  NA NA 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      : chr  "WD" "WD" "WD" "WD" ...
## $ SaleCondition : chr  "Normal" "Normal" "Normal" "Abnorml" ...
## $ SalePrice     : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...

```

Missing values

```

library(Amelia)

## Warning: package 'Amelia' was built under R version 4.0.3

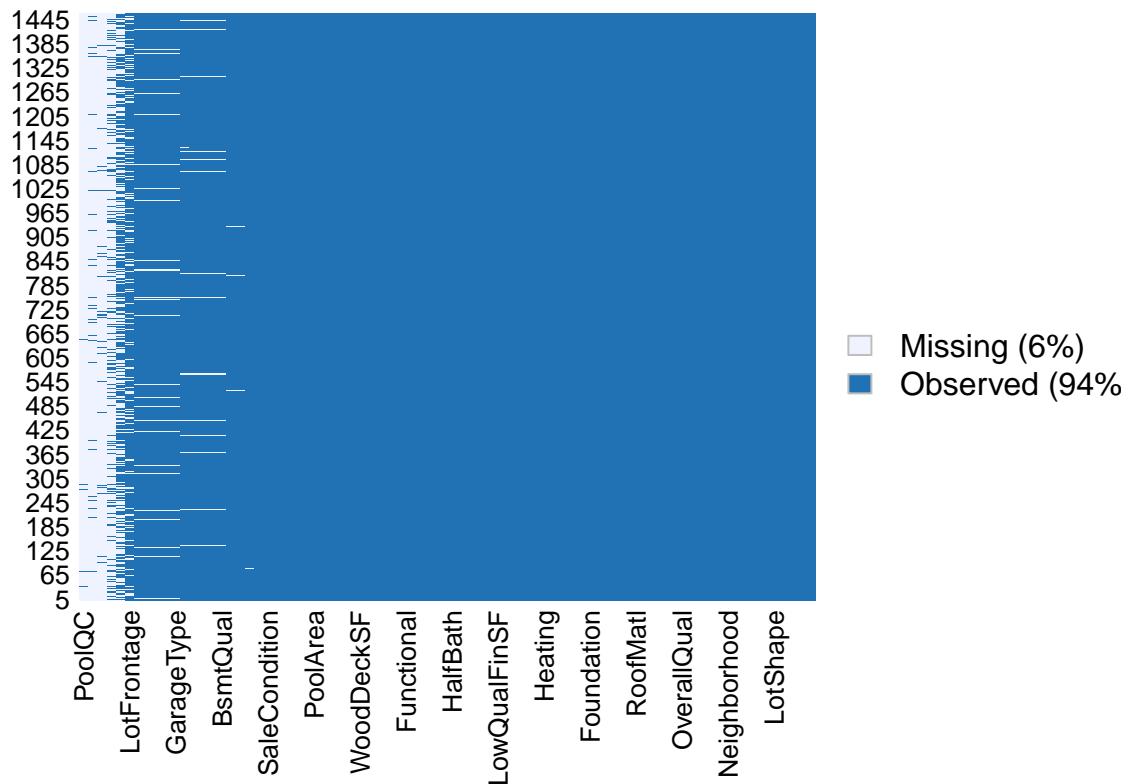
## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.6, built: 2019-11-24)
## ## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

misscounts <- sapply(combined,function(x) sum(is.na(x)))
missmap(combined, main = "Missing values")

```

Missing values



```
sort(colSums(sapply(combined, is.na)), decreasing = T)
```

##	PoolQC	MiscFeature	Alley	Fence	FireplaceQu
##	1453	1406	1369	1179	690
##	LotFrontage	GarageType	GarageYrBlt	GarageFinish	GarageQual
##	259	81	81	81	81
##	GarageCond	BsmtExposure	BsmtFinType2	BsmtQual	BsmtCond
##	81	38	38	37	37
##	BsmtFinType1	MasVnrType	MasVnrArea	Electrical	MSSubClass
##	37	8	8	1	0
##	MSZoning	LotArea	Street	LotShape	LandContour
##	0	0	0	0	0
##	Utilities	LotConfig	LandSlope	Neighborhood	Condition1
##	0	0	0	0	0
##	Condition2	BldgType	HouseStyle	OverallQual	OverallCond
##	0	0	0	0	0
##	YearBuilt	YearRemodAdd	RoofStyle	RoofMatl	Exterior1st
##	0	0	0	0	0
##	Exterior2nd	ExterQual	ExterCond	Foundation	BsmtFinSF1
##	0	0	0	0	0
##	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	Heating	HeatingQC
##	0	0	0	0	0
##	CentralAir	X1stFlrSF	X2ndFlrSF	LowQualFinSF	GrLivArea
##	0	0	0	0	0
##	BsmtFullBath	BsmtHalfBath	FullBath	HalfBath	BedroomAbvGr
##	0	0	0	0	0
##	KitchenAbvGr	KitchenQual	TotRmsAbvGrd	Functional	Fireplaces
##	0	0	0	0	0
##	GarageCars	GarageArea	PavedDrive	WoodDeckSF	OpenPorchSF
##	0	0	0	0	0
##	EnclosedPorch	X3SsnPorch	ScreenPorch	PoolArea	MiscVal
##	0	0	0	0	0
##	MoSold	YrSold	SaleType	SaleCondition	SalePrice

```
##          0          0          0          0
```

pool variables

The PoolQC has the most missing values. Pool area does not have missing values but it is related to PoolQC as it does not make sense to have a pool quality data when there is zero pool area or no pool. Its description from the data description document is.

PoolQC: Pool quality

```
Ex  Excellent
Gd  Good
TA Average/Typical
Fa  Fair
NA  No Pool
```

Since a house with no pool has NA they are not really missing values. we can check with other pool related variables to see if there are any actual missing values in our data.

```
table(is.na(combined$PoolQC))
```

```
##
## FALSE  TRUE
##    7 1453
```

```
table(combined$PoolArea, combined$PoolQC, useNA = 'ifany')
```

```
##
##           Ex   Fa   Gd <NA>
## 0       0   0   0  1453
## 480     0   0   1   0
## 512     1   0   0   0
## 519     0   1   0   0
## 555     1   0   0   0
## 576     0   0   1   0
## 648     0   1   0   0
## 738     0   0   1   0
```

Here we have some actual missing values. We have 13 houses with pool area data but we have only 10 PoolQC data available.

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 4.0.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```
combined[combined$PoolArea==0,]$PoolQC <- "None"
```

```
# convert all NA's in PoolQC to none except for the 3 actual missing values.
```

```
combined[is.na(combined$PoolQC),c("OverallQual","PoolArea")]
```

```
## [1] OverallQual PoolArea
## <0 rows> (or 0-length row.names)
```

```
# imputing the values of poolQC according to overall quality and pool area.

combined[is.na(combined$PoolQC), "PoolQC"] <- c("TA", "Gd", "TA")

# label encoding as the values are ordinal.

quality <- c("None" = 0, 'Po' = 1, 'Fa' = 2, 'TA'=3, 'Gd'=4, 'Ex'=5)

combined$PoolQC <- as.integer(revalue(combined$PoolQC, quality))
```

The following 'from' values were not present in 'x': Po, TA

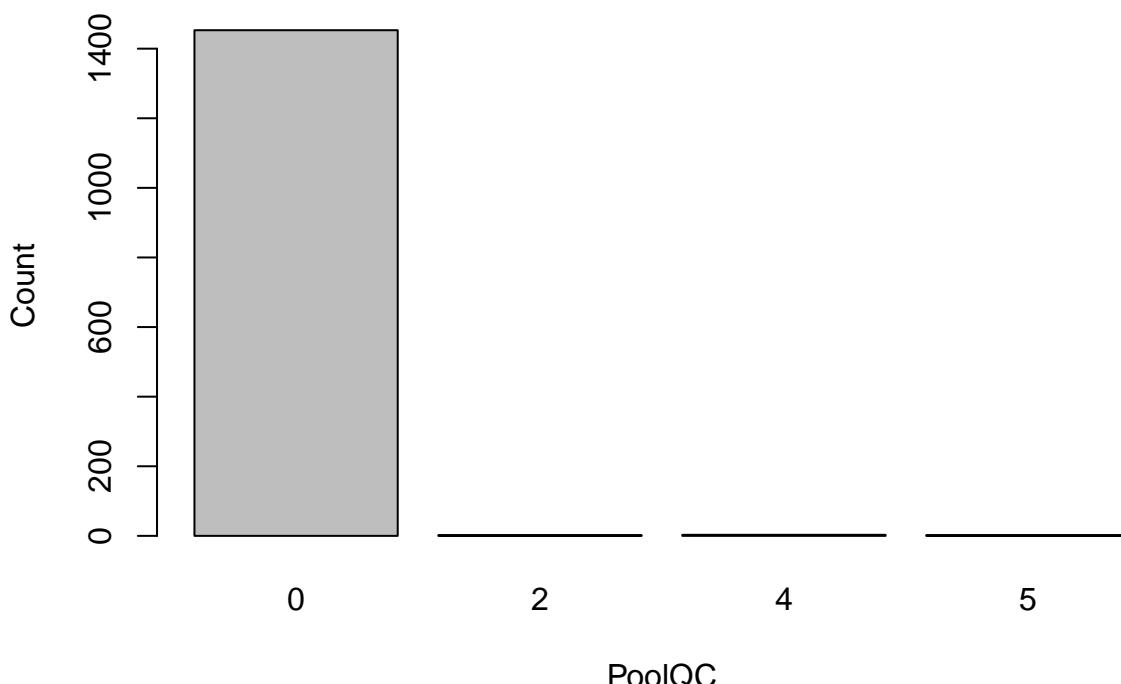
```
table(combined$PoolQC)
```

```
##
##      0      2      4      5
## 1453     2     3     2
```

```
str(combined$PoolQC)
```

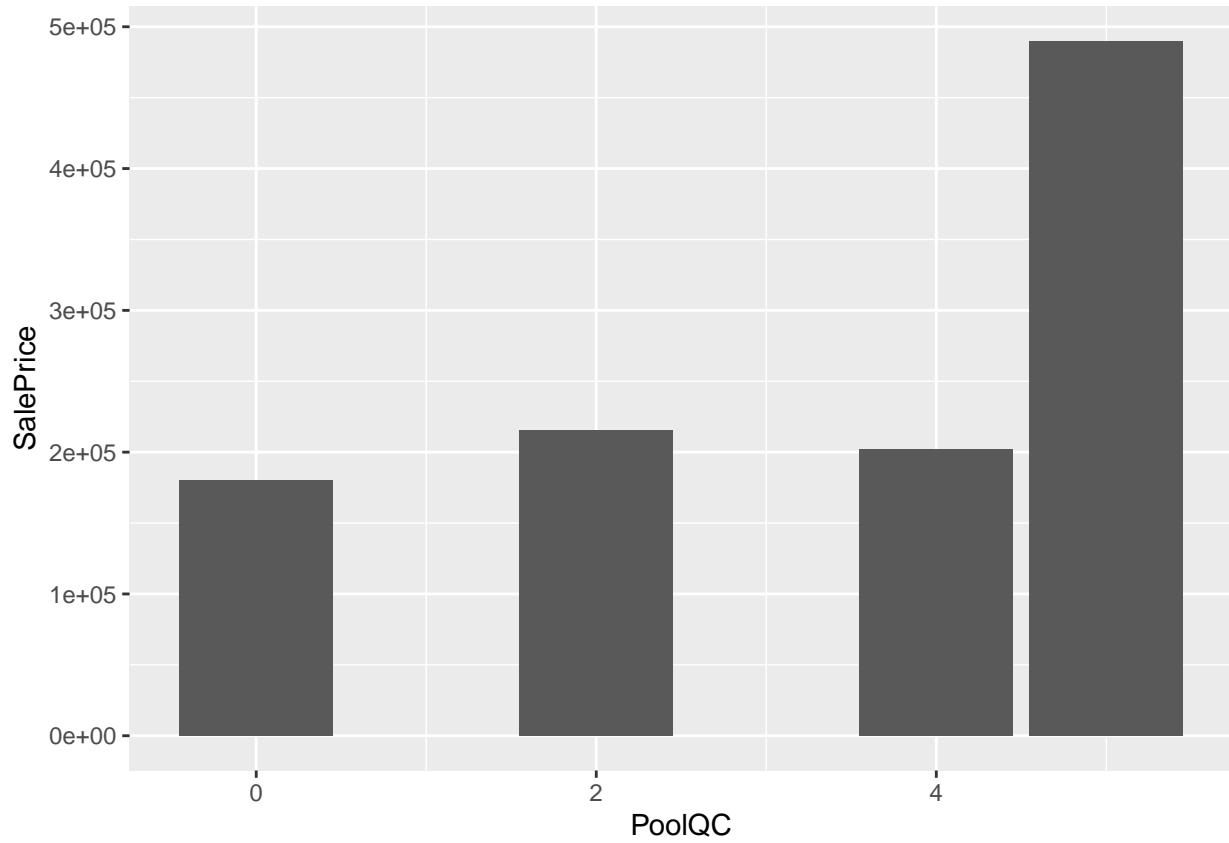
int [1:1460] 0 0 0 0 0 0 0 0 0 ...

```
barplot(table(combined$PoolQC), xlab = "PoolQC", ylab = "Count")
```



```
ggplot(combined, aes(x=PoolQC, y = SalePrice)) + geom_bar(stat = 'summary')
```

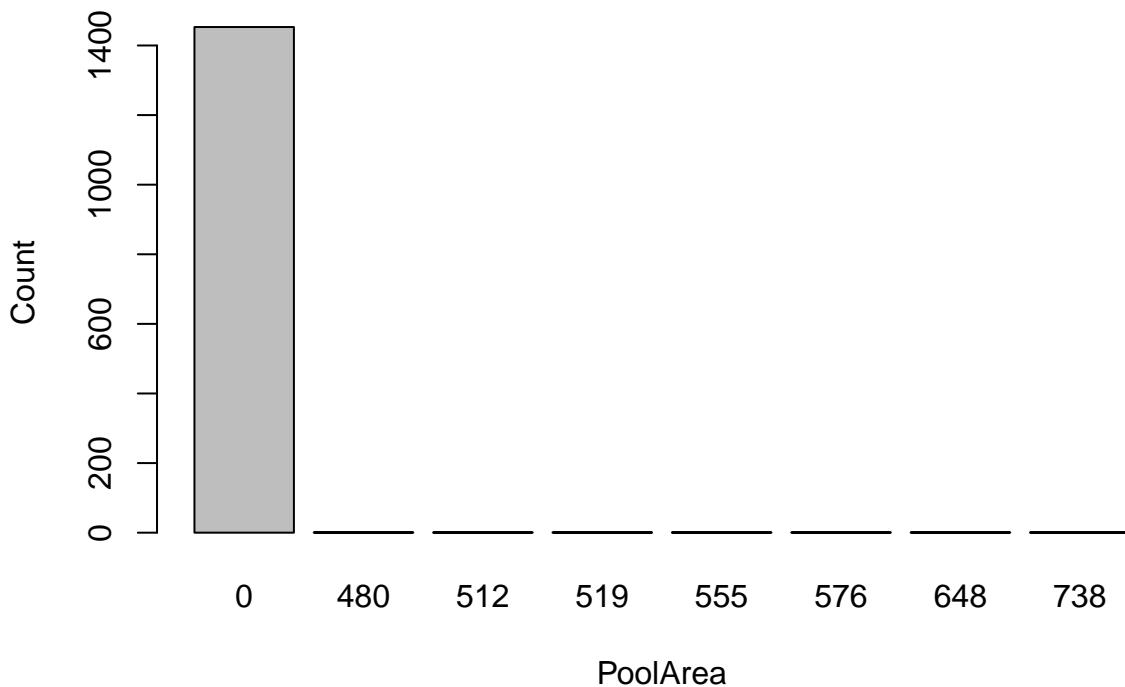
No summary function supplied, defaulting to 'mean_se()'



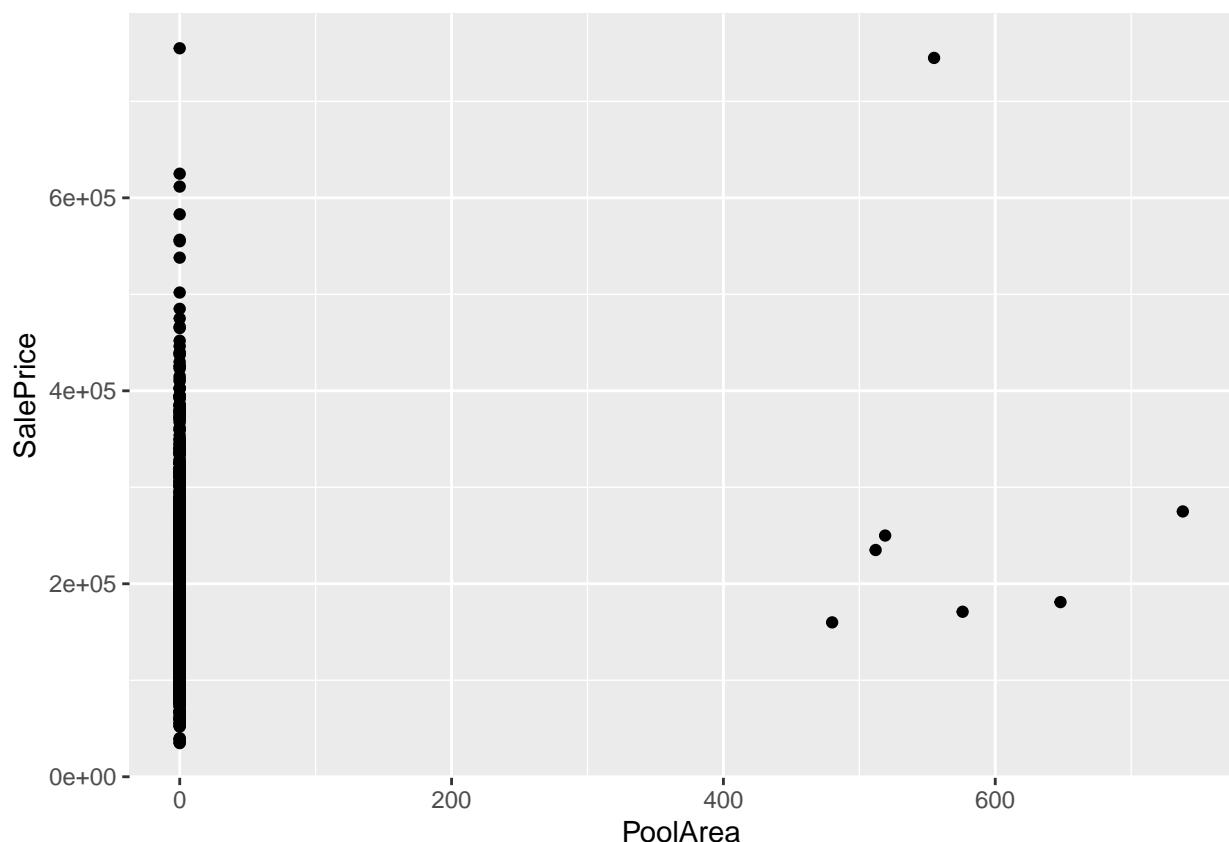
```
table(combined$PoolArea)
```

```
##  
##      0    480    512    519    555    576    648    738  
## 1453     1     1     1     1     1     1     1
```

```
barplot(table(combined$PoolArea), xlab = "PoolArea", ylab = "Count")
```



```
ggplot(combined, aes(x=PoolArea, y = SalePrice)) + geom_point()
```



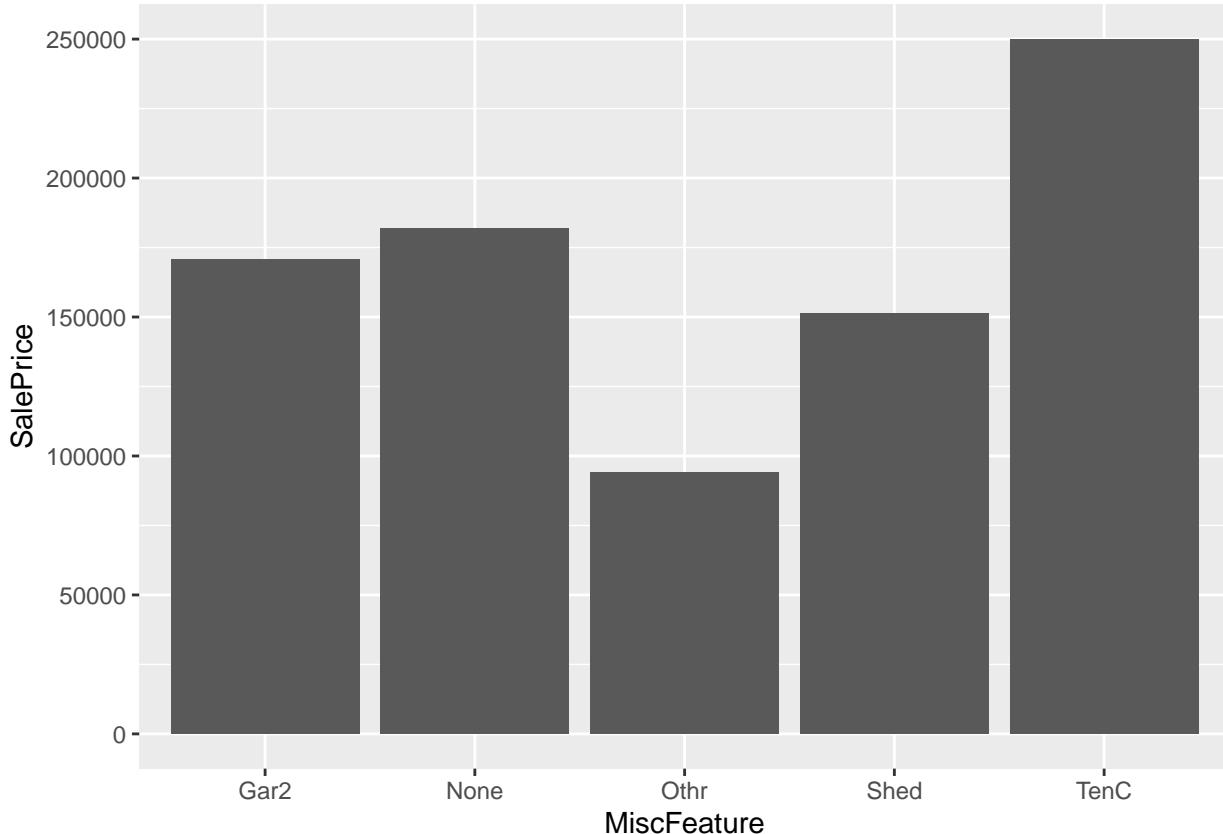
MiscFeature variable

```
table(combined$MiscFeature, useNA = "ifany")
```

```
##  
## Gar2 Othr Shed TenC <NA>  
##    2     2    49      1 1406
```

In MiscFeature variable, there are 1406 missing values that have to be replaced by none.

```
library(ggplot2)  
  
# convert all NA's in MiscFeature to none.  
combined[is.na(combined$MiscFeature), "MiscFeature"] <- "None"  
  
# convert to factor  
combined$MiscFeature <- as.factor(combined$MiscFeature)  
  
ggplot(combined, aes(x=MiscFeature, y = SalePrice)) + geom_bar(stat = 'summary')  
  
## No summary function supplied, defaulting to 'mean_se()'
```



Alley Predictor

```
table(combined$Alley, useNA = "ifany")
```

```
##  
## Grvl Pave <NA>  
##    50    41 1369
```

```

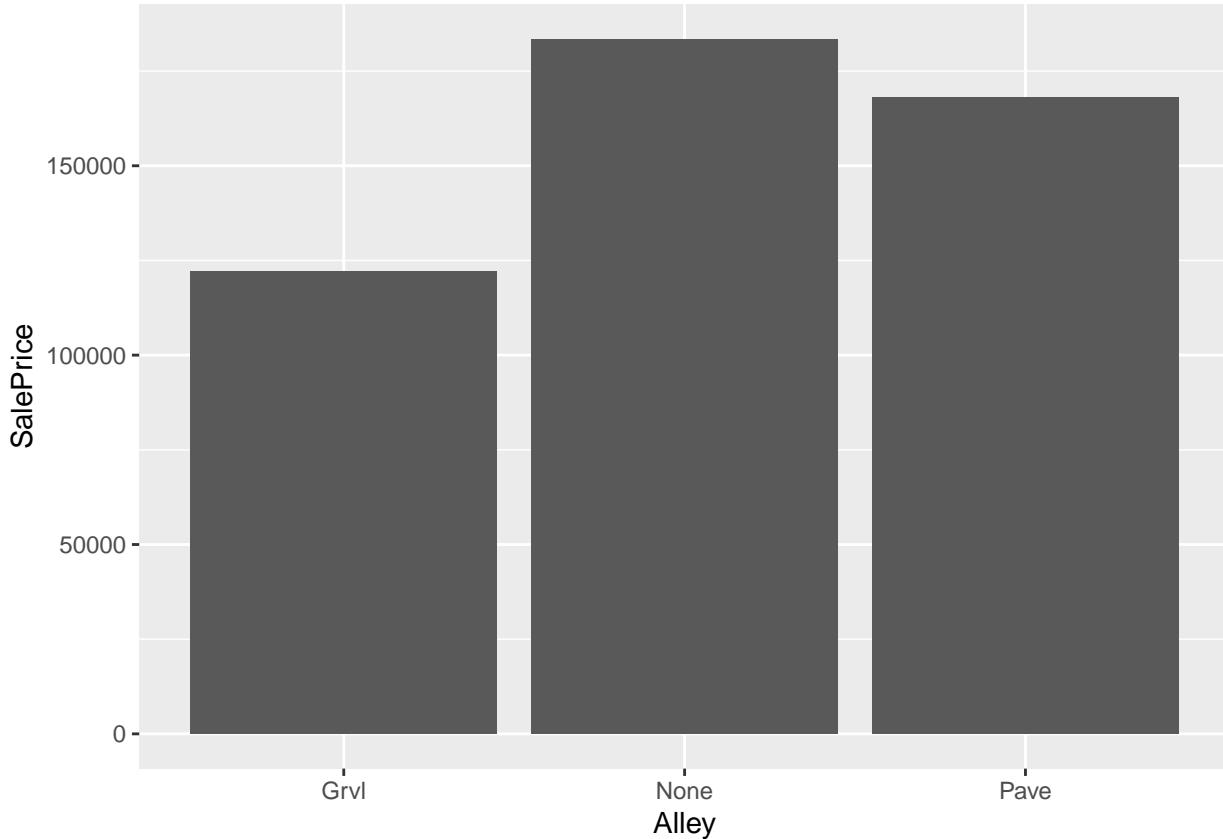
# convert all NA's in Alley to none.
combined[is.na(combined$Alley), "Alley"] <- "None"

# convert to factor
combined$Alley <- as.factor(combined$Alley)

ggplot(combined, aes(x=Alley, y = SalePrice)) + geom_bar(stat = 'summary')

```

No summary function supplied, defaulting to 'mean_se()'



Fence predictor

```

table(combined$Fence, useNA = "ifany")

##
## GdPrv  GdWo  MnPrv  MnWw  <NA>
##    59     54    157     11   1179

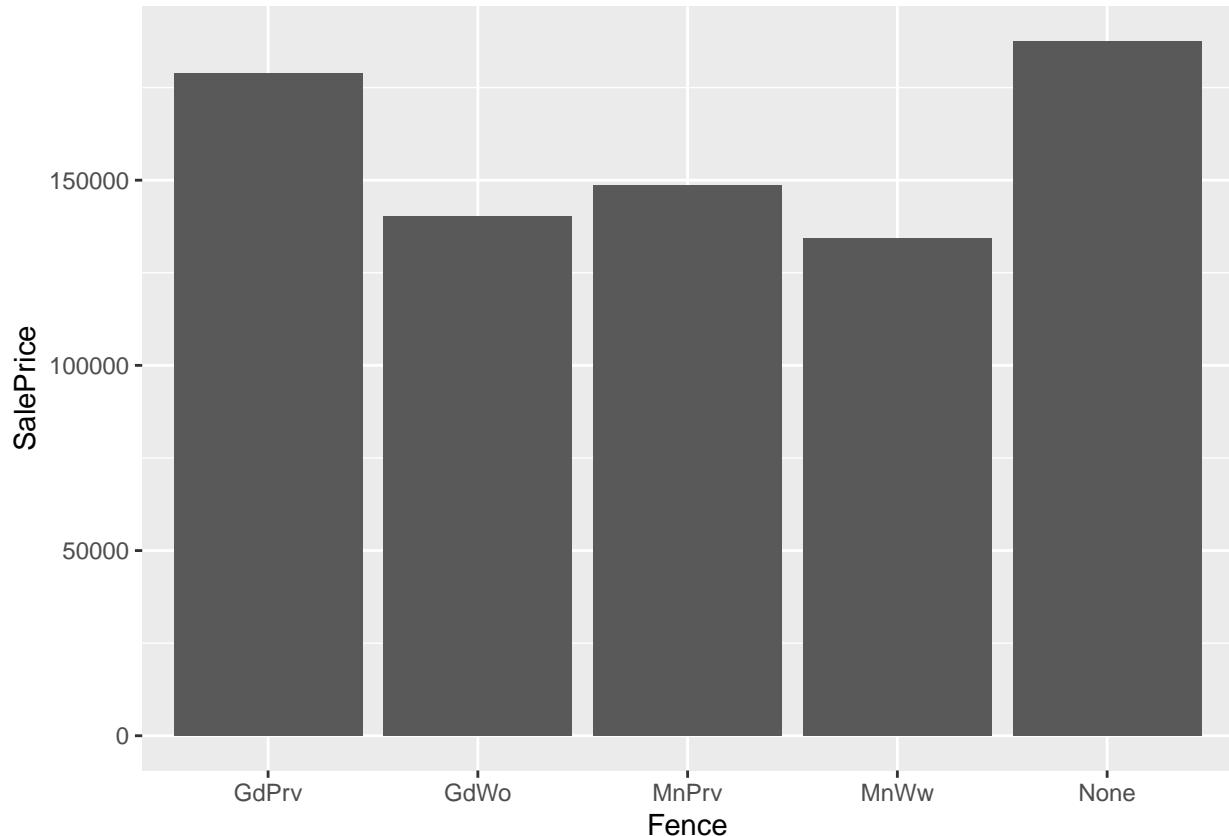
# convert all NA's in Fence to none.
combined[is.na(combined$Fence), "Fence"] <- "None"

# convert to factor
combined$Fence <- as.factor(combined$Fence)

ggplot(combined, aes(x=Fence, y = SalePrice)) + geom_bar(stat = 'summary')

```

No summary function supplied, defaulting to 'mean_se()'



Fireplace variables

Fireplace quality

```
table(combined$FireplaceQu, useNA = "ifany")

##
##   Ex   Fa   Gd   Po   TA <NA>
##   24   33  380   20  313  690

# convert all NA's in FireplaceQu to none.

combined[is.na(combined$FireplaceQu),"FireplaceQu"] <- "None"

# Changing and converting to factor levels from character.

combined$FireplaceQu <- as.integer(revalue(combined$FireplaceQu, quality))

table(combined$FireplaceQu, useNA = "ifany")

##
##   0   1   2   3   4   5
## 690   20   33  313  380   24

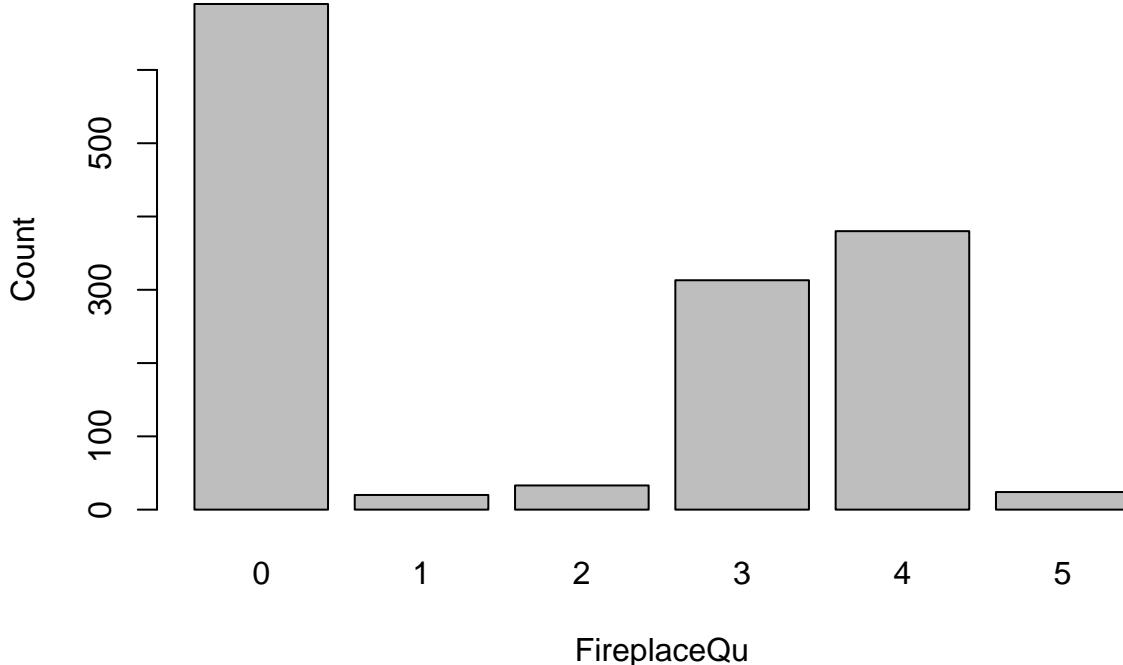
str(combined$FireplaceQu)

##  int [1:1460] 0 3 3 4 3 0 4 3 3 3 ...
```

```
anyNA(combined$FireplaceQu)
```

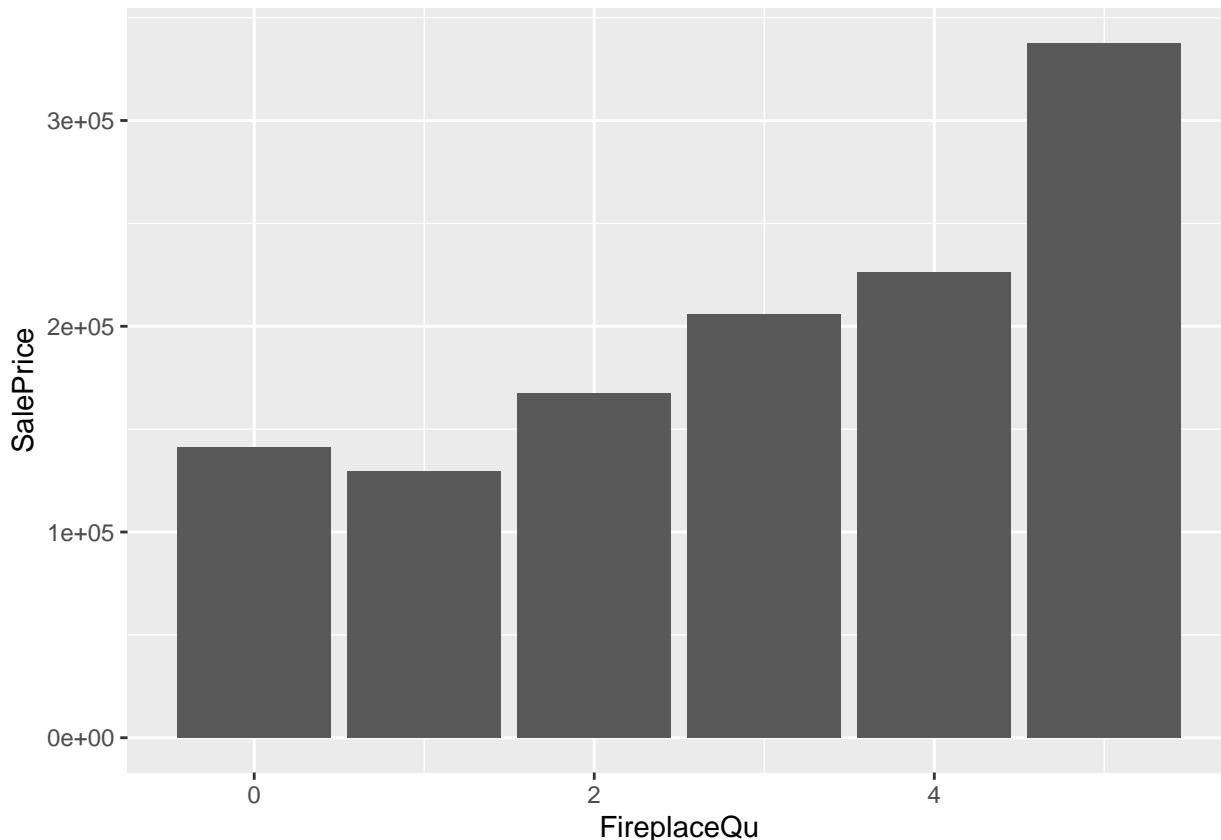
```
## [1] FALSE
```

```
barplot(table(combined$FireplaceQu), xlab = "FireplaceQu", ylab = "Count")
```



```
ggplot(combined, aes(x=FireplaceQu, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



Lot variables

LotFrontage LotShape LotConfig LotArea

```
table(is.na(combined$LotFrontage))
```

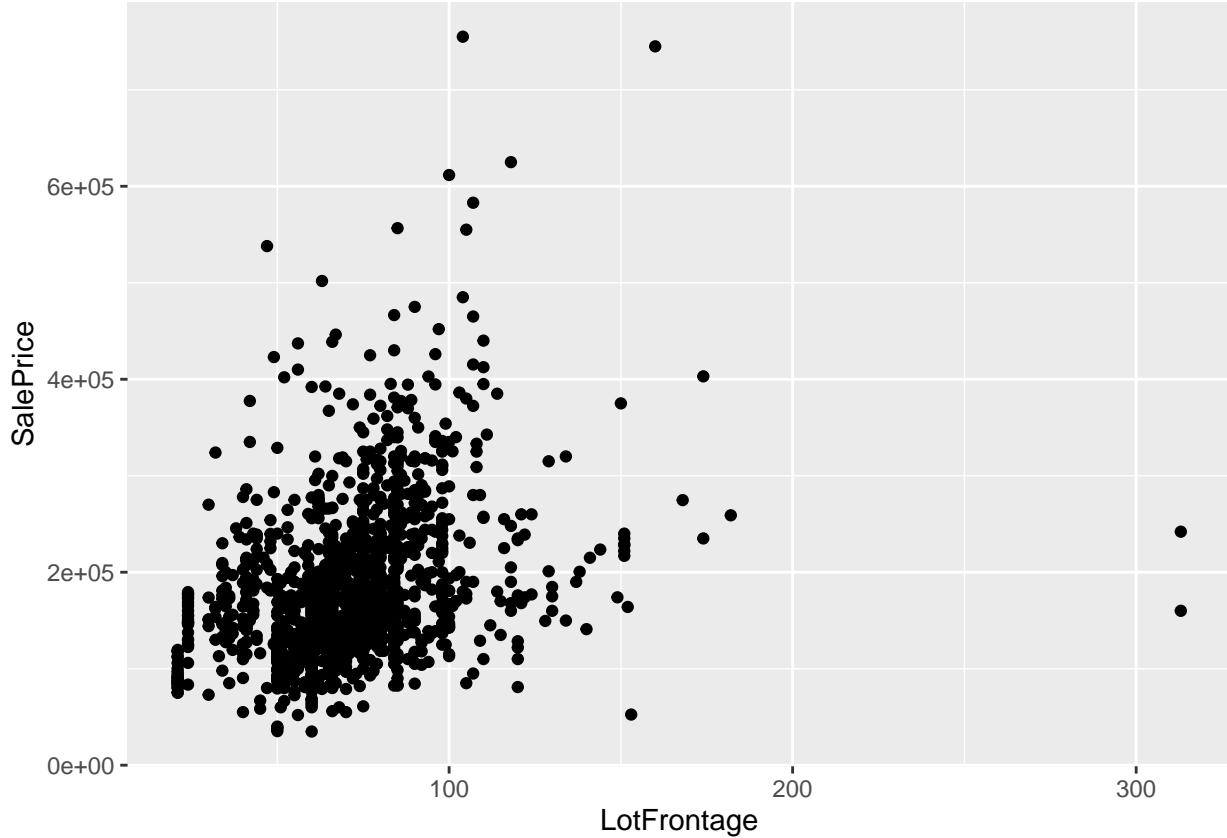
```
##  
## FALSE TRUE  
## 1201 259
```

Here we have 259 missing values which cannot be replaced by none as it is a numerical variable. So we predict using rpart.

<http://r-statistics.co/Missing-Value-Treatment-With-R.html>

```
# predictors that lotfrontage variable might depend on.  
predictors <- c("MSSubClass", "MSZoning", "LotFrontage", "LotArea", "Street", "Alley", "LotShape", "LandContour"  
library(rpart)  
  
## Warning: package 'rpart' was built under R version 4.0.3  
  
mod <- rpart(LotFrontage~, data = combined[!is.na(combined$LotFrontage),predictors], method = "anova", na.action = "na.omit")  
  
pred <- predict(mod, combined[is.na(combined$LotFrontage),predictors])  
pred <- round(pred)  
combined$LotFrontage[is.na(combined$LotFrontage)] <- pred  
anyNA(combined$LotFrontage)  
  
## [1] FALSE
```

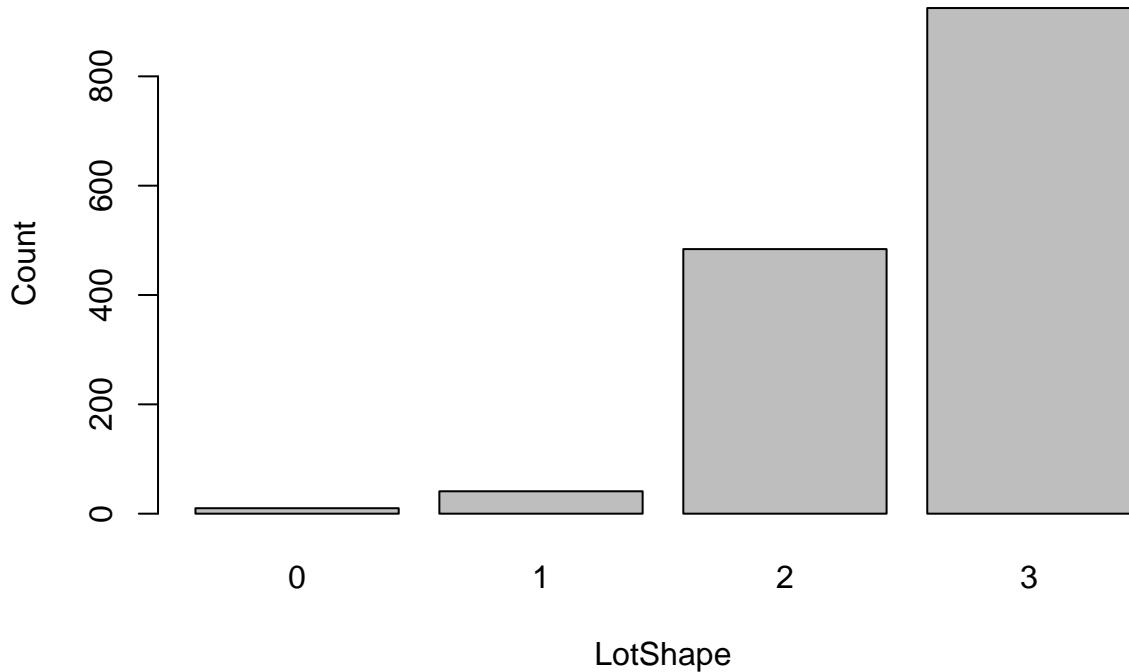
```
ggplot(combined, aes(x=LotFrontage, y = SalePrice)) + geom_point()
```



```
table(combined$LotShape, useNA = "ifany")
```

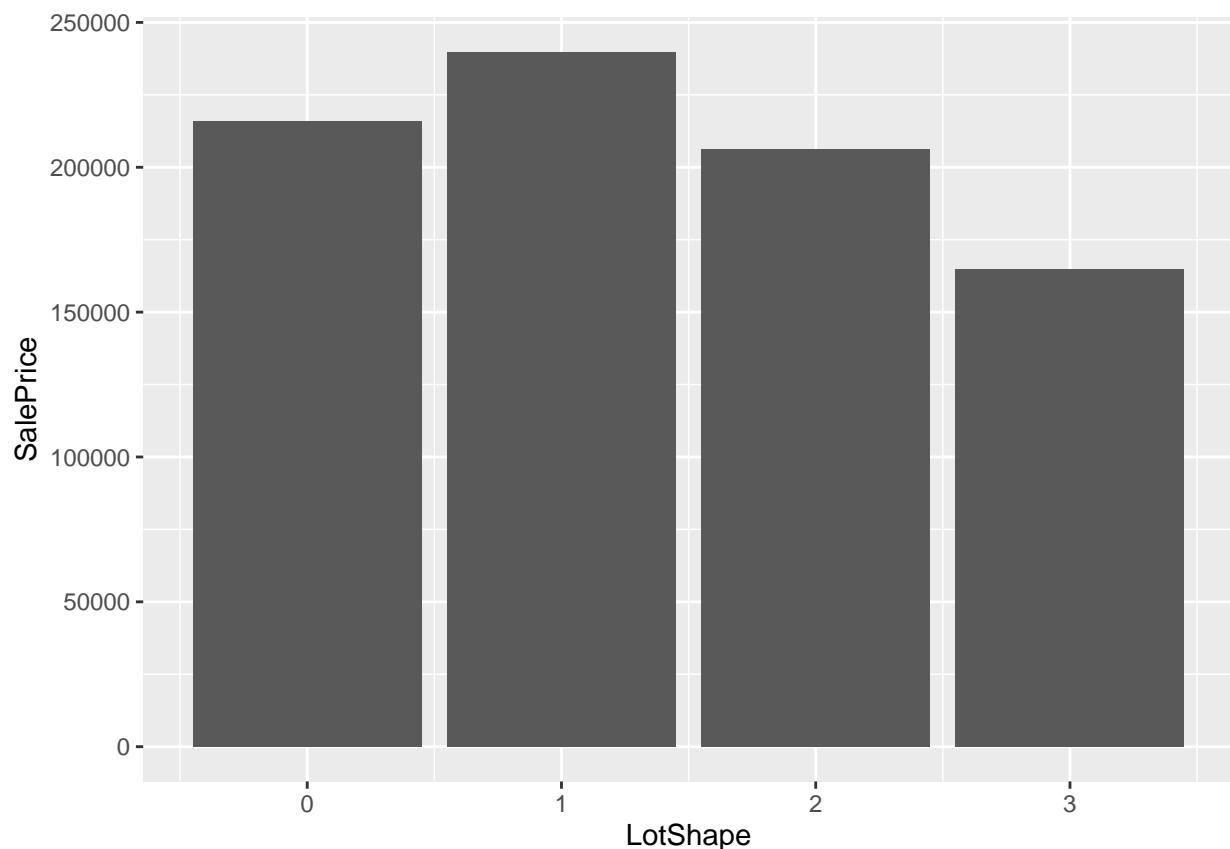
```
##  
## IR1 IR2 IR3 Reg  
## 484 41 10 925
```

```
combined$LotShape <- as.integer(revalue(combined$LotShape, c("IR3"=0 , "IR2"=1 , "IR1"=2 , "Reg"=3 )))  
barplot(table(combined$LotShape), xlab = "LotShape", ylab = "Count")
```



```
ggplot(combined, aes(x=LotShape, y = SalePrice)) + geom_bar(stat = 'summary')
```

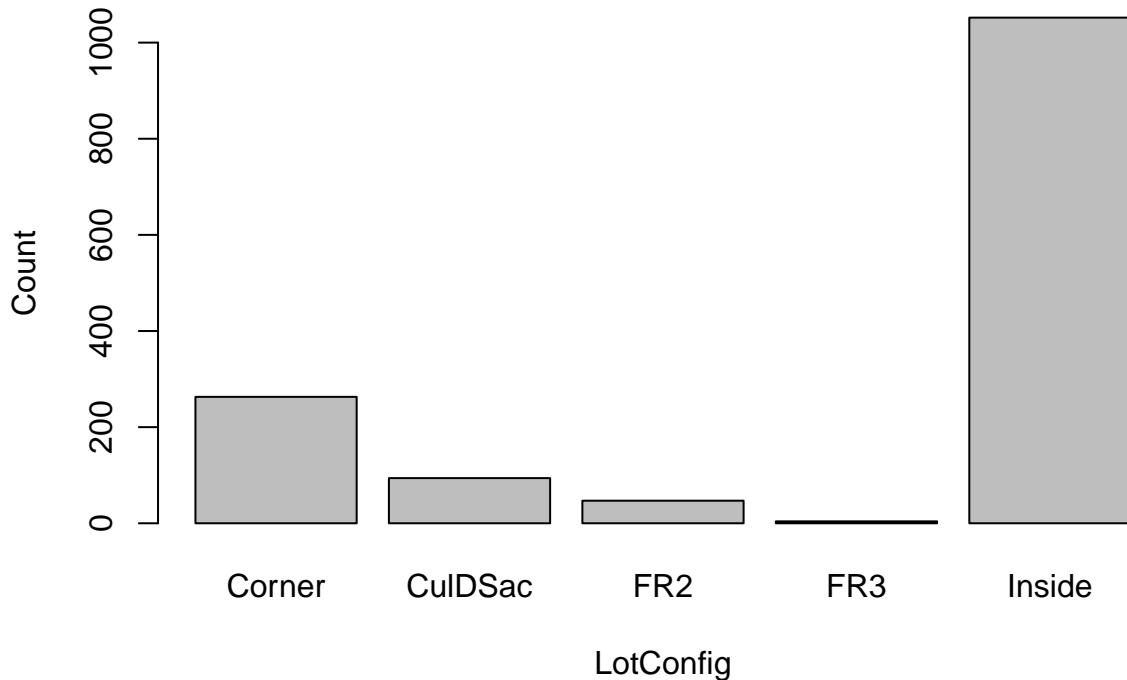
```
## No summary function supplied, defaulting to 'mean_se()'
```



```
table(combined$LotConfig, useNA = "ifany")
```

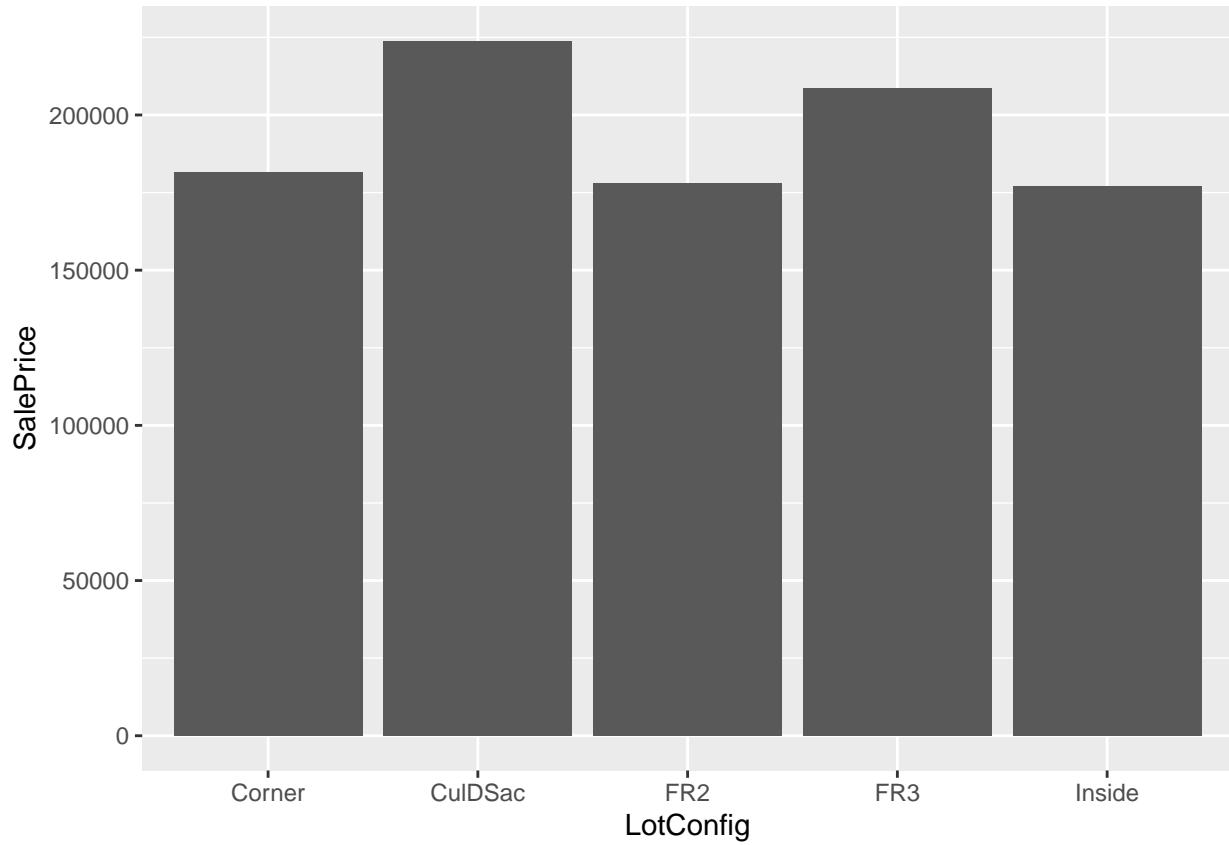
```
##  
##   Corner CulDSac      FR2      FR3 Inside  
##     263      94       47       4    1052
```

```
combined$LotConfig <- as.factor(combined$LotConfig)  
barplot(table(combined$LotConfig), xlab = "LotConfig", ylab = "Count")
```



```
ggplot(combined, aes(x=LotConfig, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



Garage variables

GarageYrBlt GarageType GarageFinish, GarageQual, GarageCond, GarageCars, GarageArea

```
garage <- c("GarageYrBlt", "GarageType", "GarageFinish", "GarageQual", "GarageCond", "GarageCars", "GarageArea")
sort(colSums(sapply(combined[,garage], is.na)), decreasing = T)

##  GarageYrBlt    GarageType GarageFinish    GarageQual    GarageCond    GarageCars
##      81          81          81          81          81          0
##  GarageArea
##      0

combined$GarageYrBlt[is.na(combined$GarageYrBlt)] <- combined$YearBuilt[is.na(combined$GarageYrBlt)]

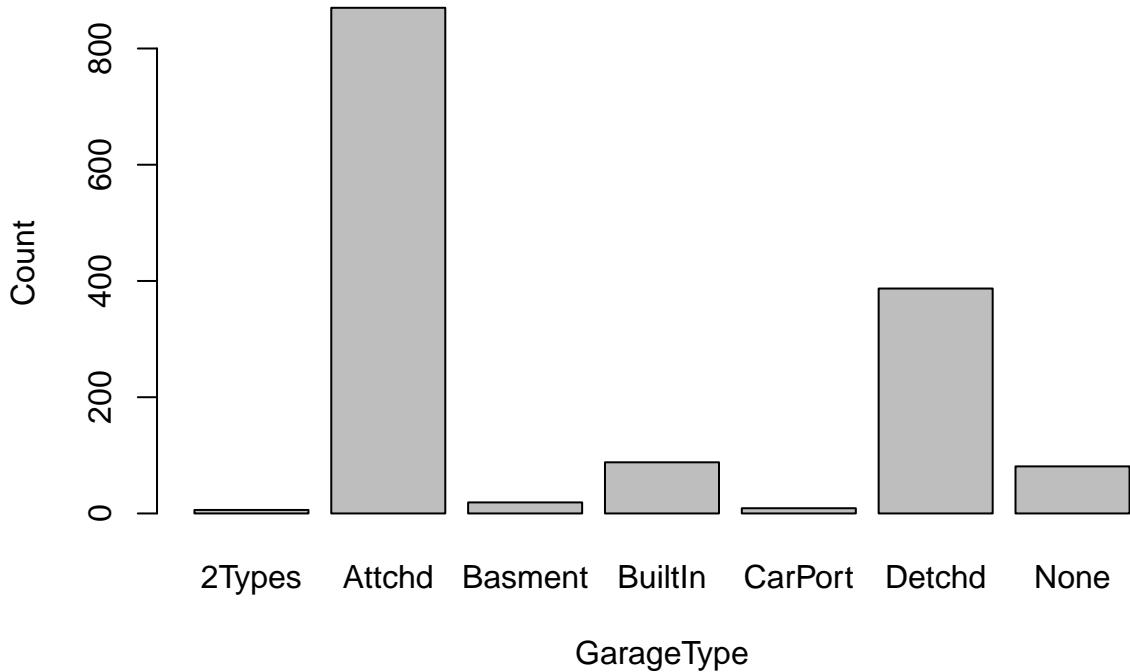
combined$GarageType[is.na(combined$GarageType)] <- "None"
combined$GarageFinish[is.na(combined$GarageFinish)] <- "None"
combined$GarageCond[is.na(combined$GarageCond)] <- "None"
combined$GarageQual[is.na(combined$GarageQual)] <- "None"
sort(colSums(sapply(combined[,garage], is.na)), decreasing = T)

##  GarageYrBlt    GarageType GarageFinish    GarageQual    GarageCond    GarageCars
##      0          0          0          0          0          0
##  GarageArea
##      0

# convert into factor
combined$GarageType <- as.factor(combined$GarageType)
table(combined$GarageType)

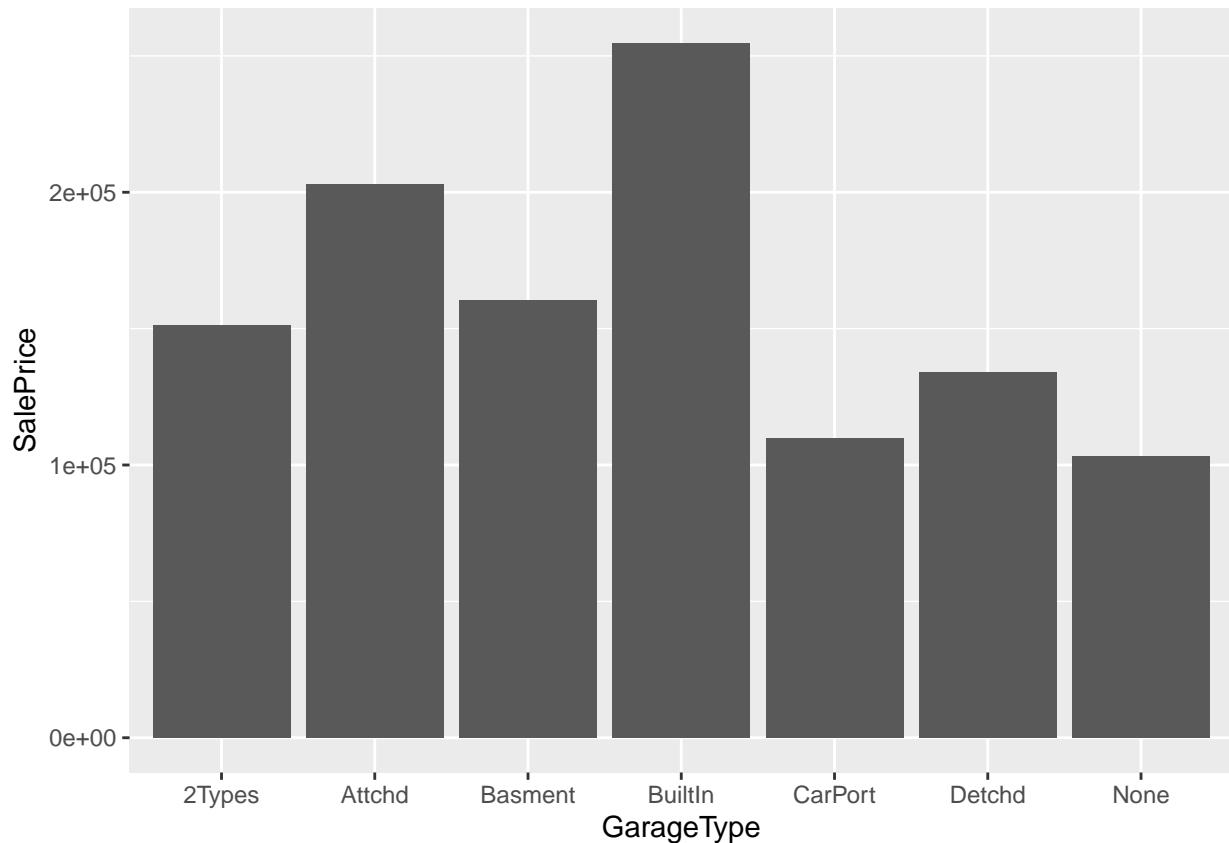
## 
## 2Types Attchd Basement BuiltIn CarPort Detchd None
##     6     870     19     88      9    387    81
```

```
barplot(table(combined$GarageType), xlab = "GarageType", ylab = "Count")
```



```
ggplot(combined, aes(x=GarageType, y = SalePrice)) + geom_bar(stat = 'summary')
```

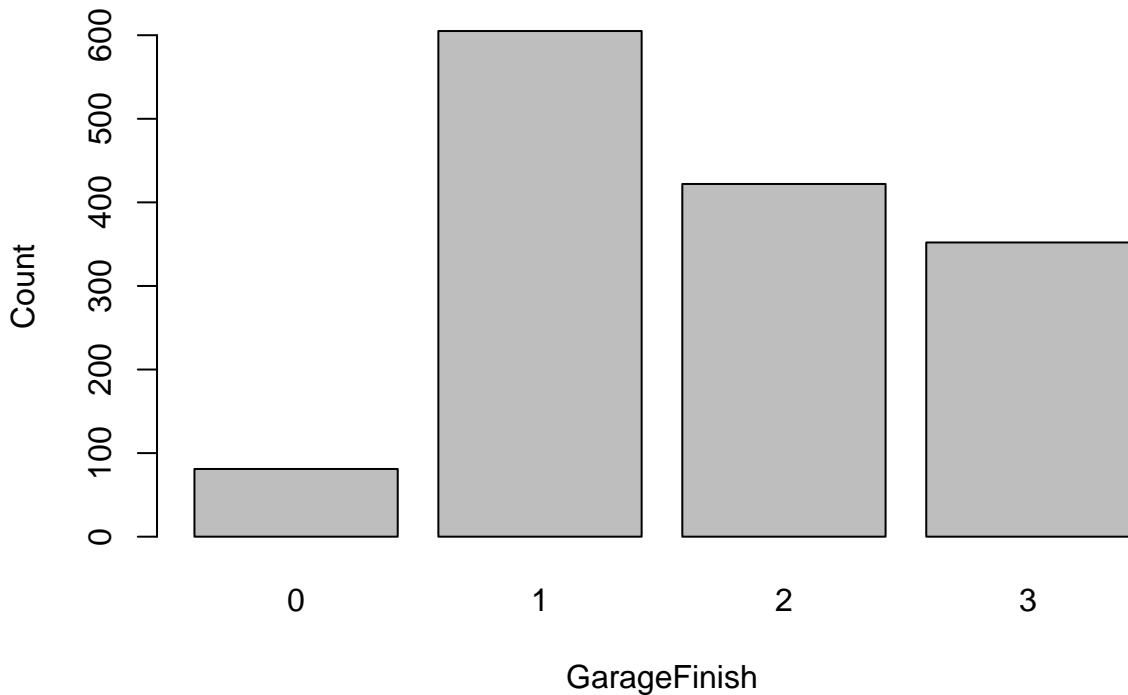
```
## No summary function supplied, defaulting to 'mean_se()'
```



```
Finish <- c('None'=0, 'Unf'=1, 'RFn'=2, 'Fin'=3)
combined$GarageFinish<-as.integer(revalue(combined$GarageFinish, Finish))
table(combined$GarageFinish, useNA = 'ifany')
```

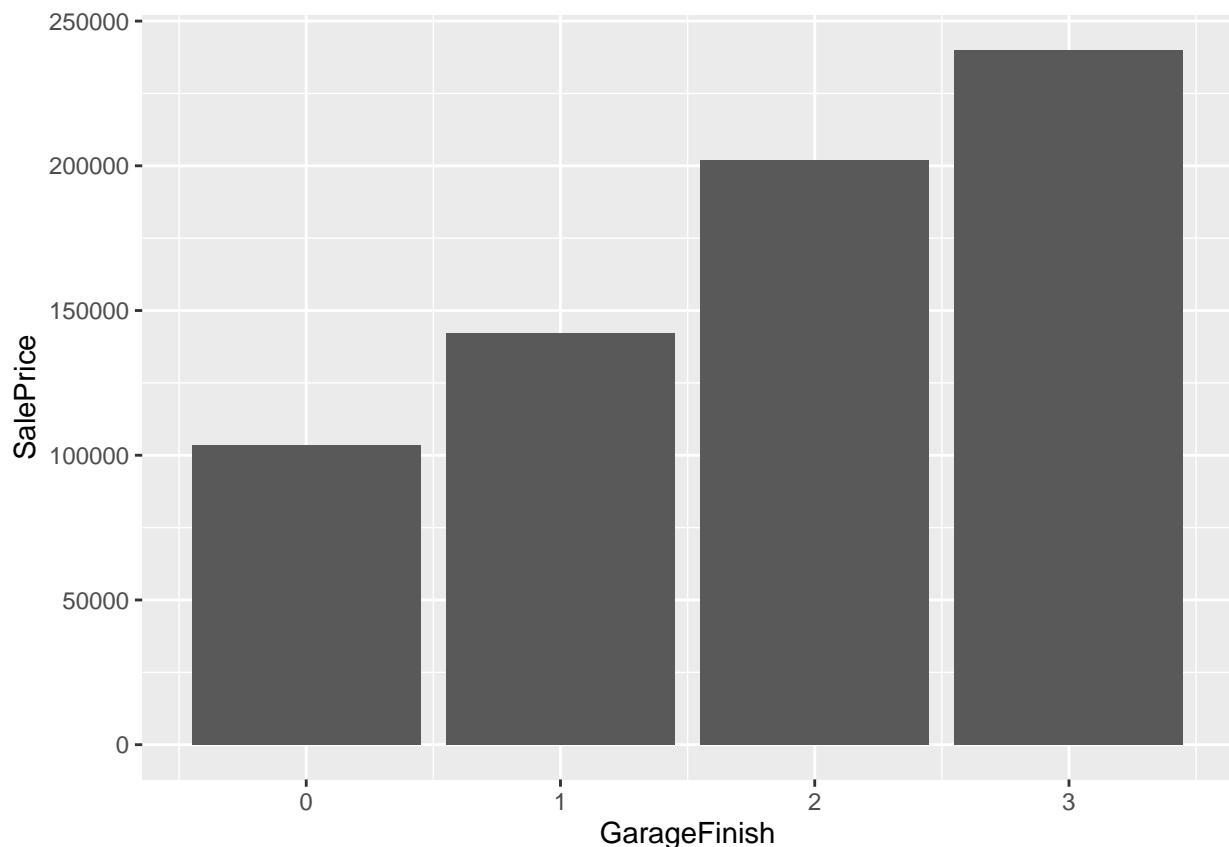
```
##  
##    0    1    2    3  
##   81  605  422  352
```

```
barplot(table(combined$GarageFinish), xlab = "GarageFinish", ylab = "Count")
```



```
ggplot(combined, aes(x=GarageFinish, y = SalePrice)) + geom_bar(stat = 'summary')
```

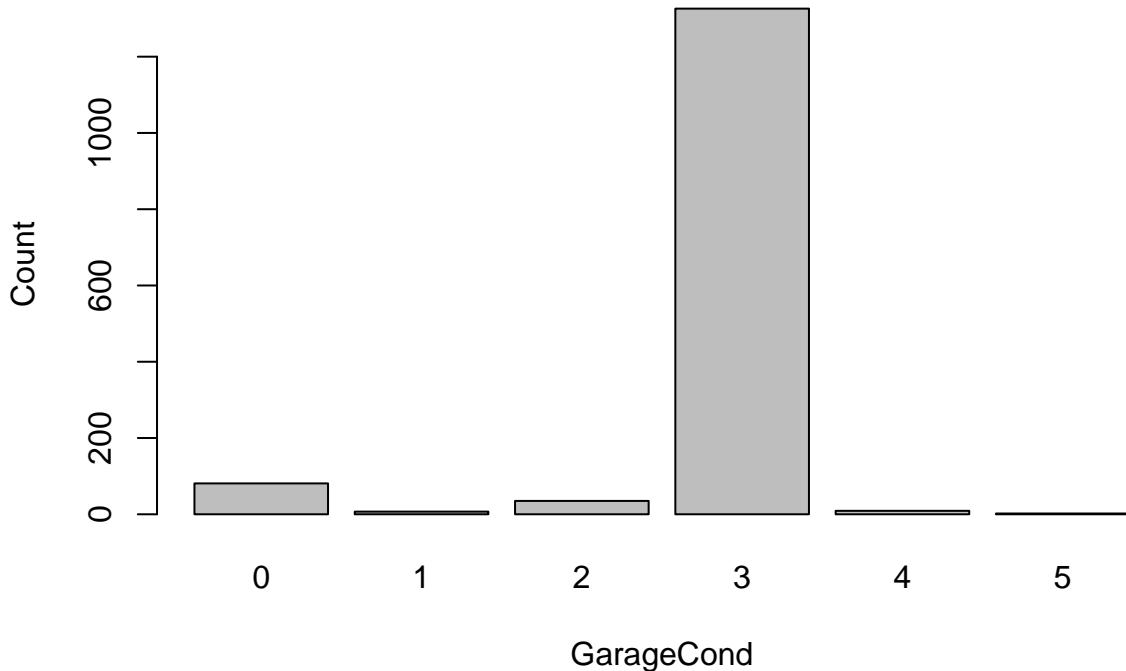
```
## No summary function supplied, defaulting to 'mean_se()'
```



```
combined$GarageCond<-as.integer(revalue(combined$GarageCond, quality))
table(combined$GarageCond, useNA = "ifany")
```

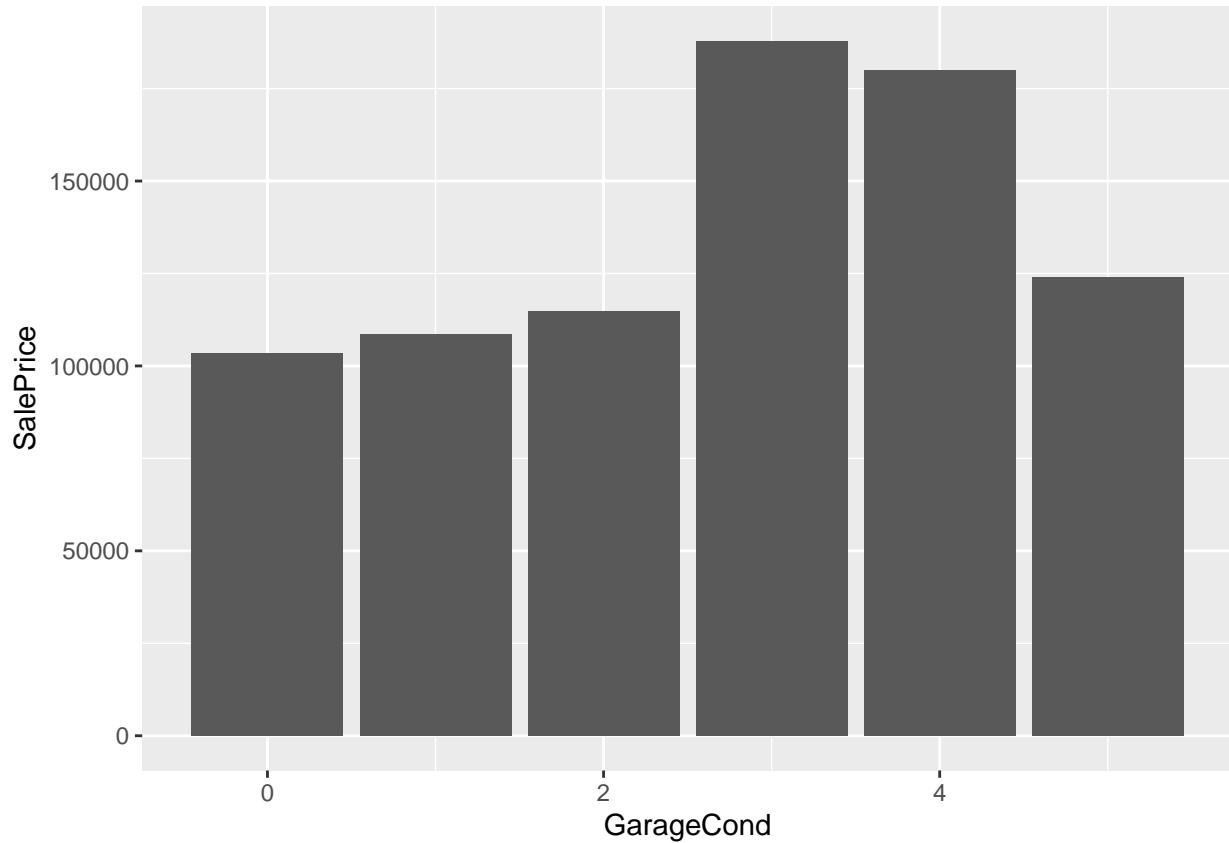
```
##  
##      0      1      2      3      4      5  
##    81     7    35 1326     9     2
```

```
barplot(table(combined$GarageCond), xlab = "GarageCond", ylab = "Count")
```



```
ggplot(combined, aes(x=GarageCond, y = SalePrice)) + geom_bar(stat = 'summary')
```

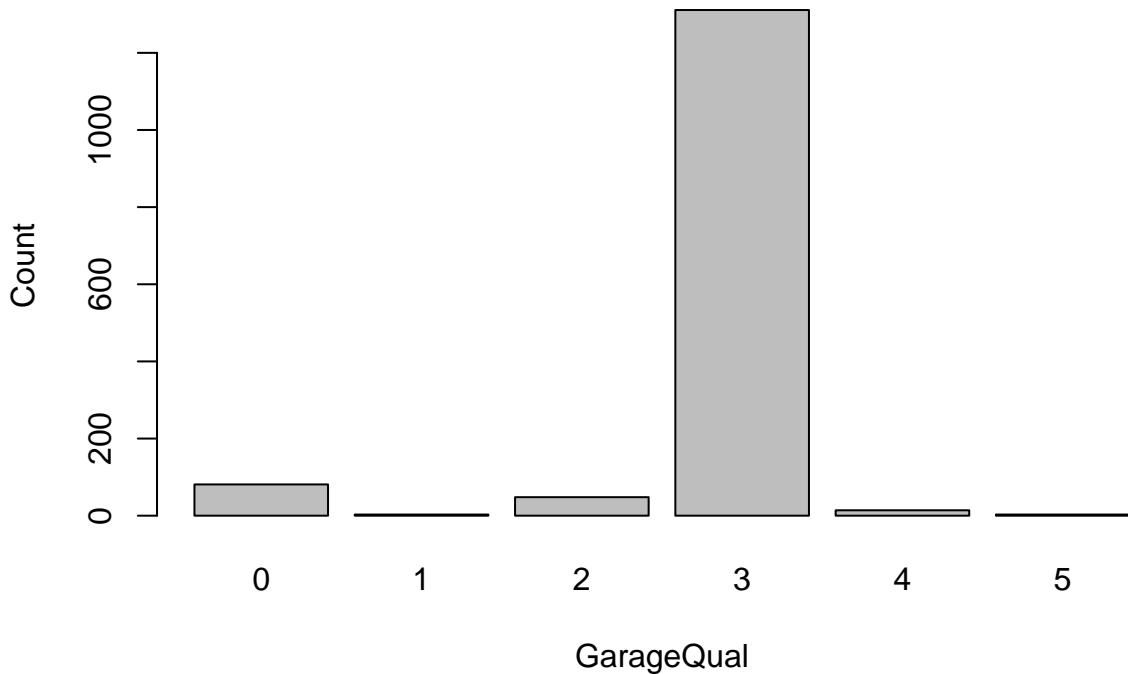
```
## No summary function supplied, defaulting to 'mean_se()'
```



```
combined$GarageQual<-as.integer(revalue(combined$GarageQual, quality))
table(combined$GarageQual, useNA = "ifany")
```

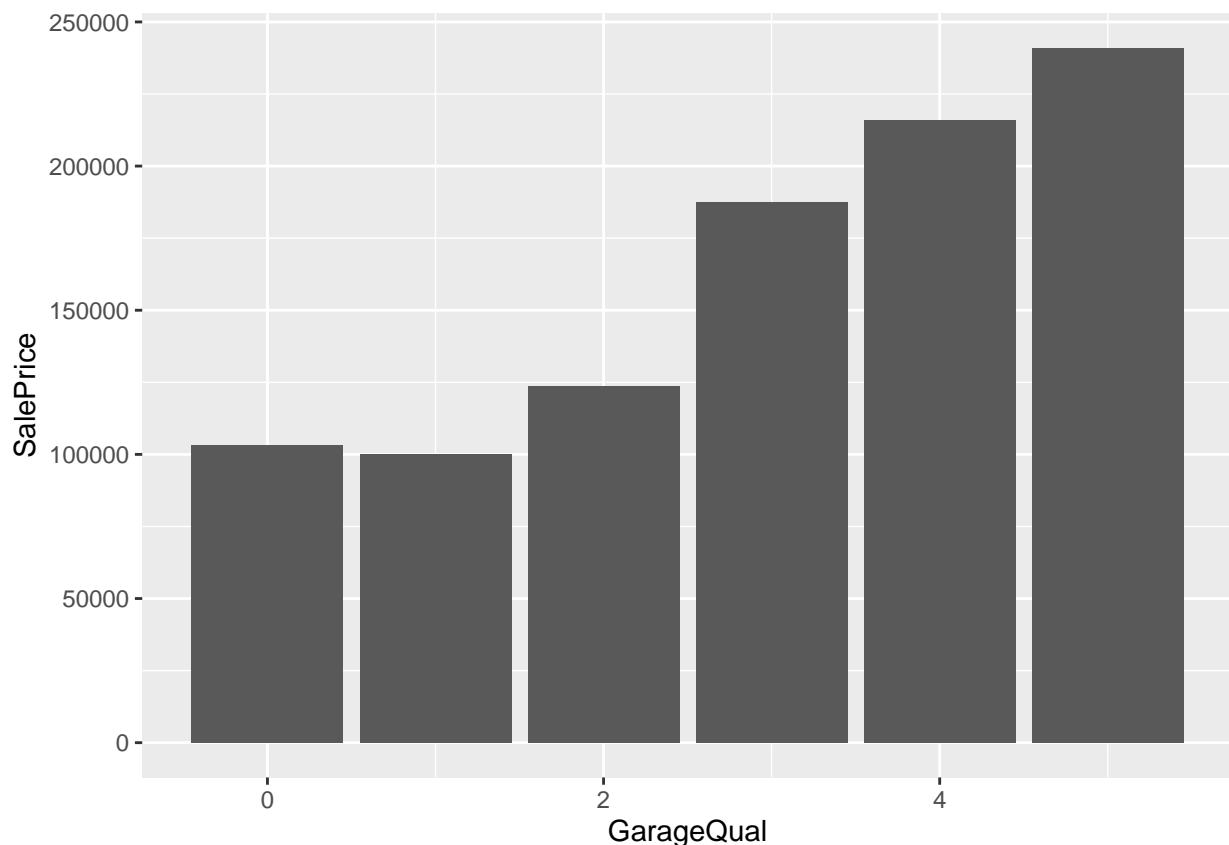
```
##
##      0      1      2      3      4      5
##     81      3     48  1311     14      3
```

```
barplot(table(combined$GarageQual), xlab = "GarageQual", ylab = "Count")
```

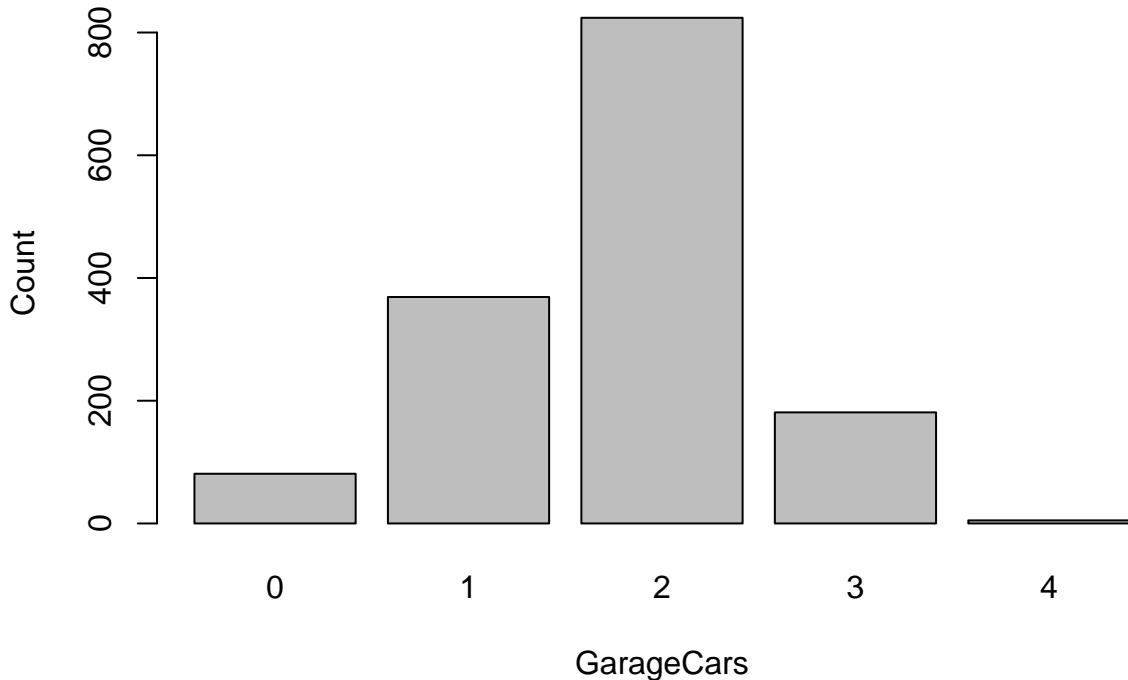


```
ggplot(combined, aes(x=GarageQual, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```

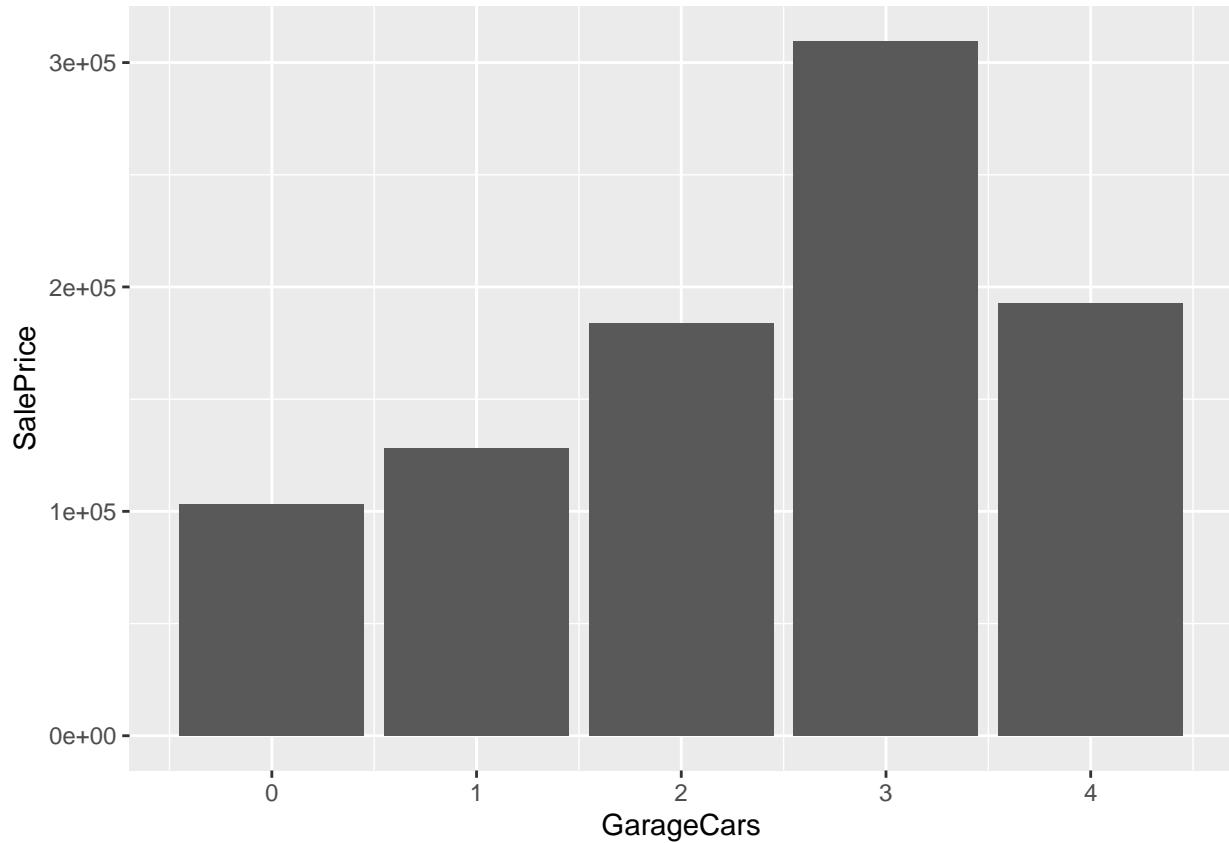


```
barplot(table(combined$GarageCars), xlab = "GarageCars", ylab = "Count")
```

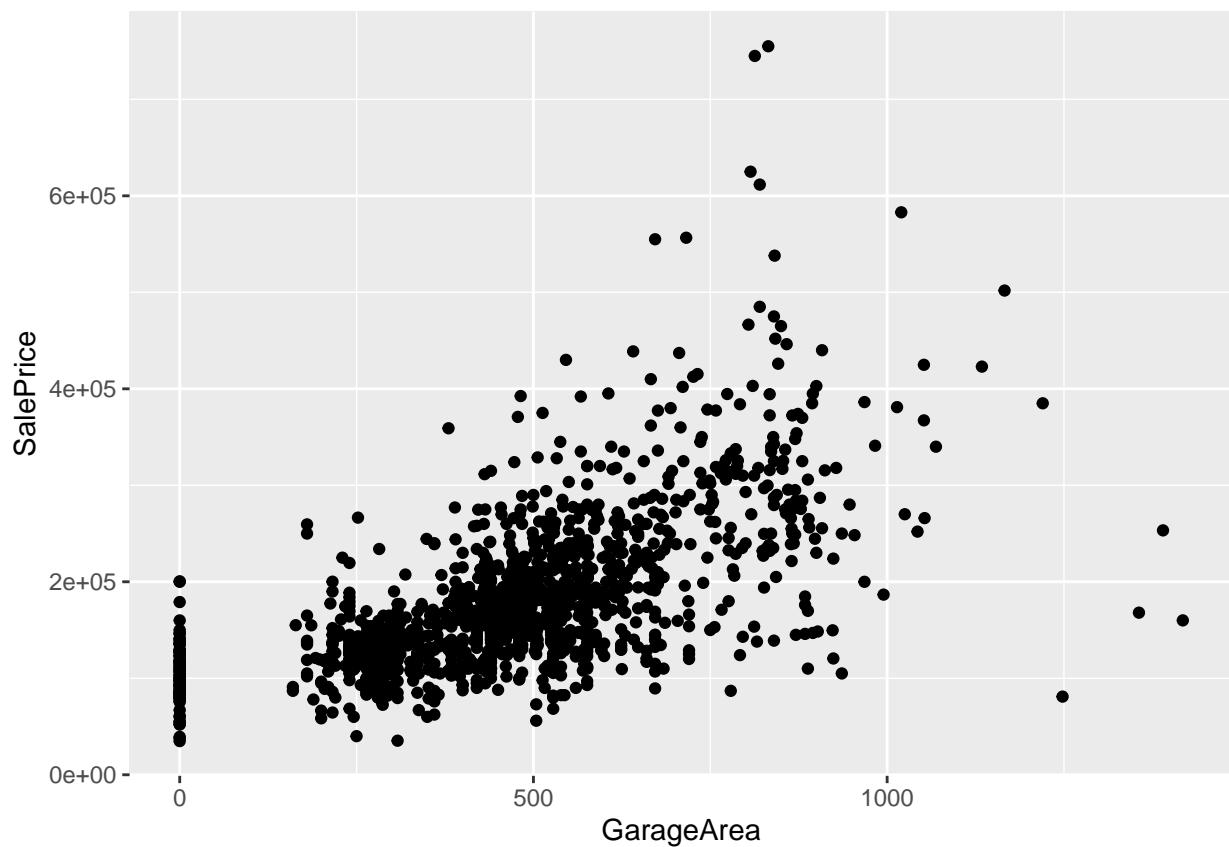


```
ggplot(combined, aes(x=GarageCars, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



```
ggplot(combined, aes(x=GarageArea, y = SalePrice)) + geom_point()
```



Basement variables

there are 11 basement variables

BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2, BsmtFullBath, BsmtHalfBath, BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF

```
basement <- c("BsmtQual", "BsmtCond", "BsmtExposure", "BsmtFinType1", "BsmtFinType2", "BsmtFullBath", "BsmtHalfBath", "  
sort(colSums(sapply(combined[,basement], is.na)), decreasing = T)  
  
## BsmtExposure BsmtFinType2 BsmtQual BsmtCond BsmtFinType1 BsmtFullBath  
## 38 38 37 37 37 0  
## BsmtHalfBath BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF  
## 0 0 0 0 0  
  
x <- which(!is.na(combined$BsmtFinType1) & (is.na(combined$BsmtCond) | is.na(combined$BsmtExposure) | is.na(combined  
combined[x,bsement]  
  
## BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2 BsmtFullBath  
## 333 Gd TA No GLQ <NA> 1  
## 949 Gd TA <NA> Unf Unf 0  
## BsmtHalfBath BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF  
## 333 0 1124 479 1603 3206  
## 949 0 0 0 936 936  
  
# impute mode  
combined[c(949), "BsmtExposure"] <- names(sort(-table(combined$BsmtExposure)))[1]  
combined[c(333), "BsmtFinType2"] <- names(sort(-table(combined$BsmtFinType2)))[1]  
combined[x,bsement]  
  
## BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2 BsmtFullBath  
## 333 Gd TA No GLQ Unf 1  
## 949 Gd TA No Unf Unf 0  
## BsmtHalfBath BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF  
## 333 0 1124 479 1603 3206  
## 949 0 0 0 936 936  
  
anyNA(combined[x,bsement])  
  
## [1] FALSE  
  
sort(colSums(sapply(combined[,bsement], is.na)), decreasing = T)  
  
## BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2 BsmtFullBath  
## 37 37 37 37 37 0  
## BsmtHalfBath BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF  
## 0 0 0 0 0  
  
combined$BsmtQual[is.na(combined$BsmtQual)] <- "None"  
combined$BsmtCond[is.na(combined$BsmtCond)] <- "None"  
combined$BsmtExposure[is.na(combined$BsmtExposure)] <- "None"  
combined$BsmtFinType1[is.na(combined$BsmtFinType1)] <- "None"  
combined$BsmtFinType2[is.na(combined$BsmtFinType2)] <- "None"  
combined$BsmtFullBath[is.na(combined$BsmtFullBath)] <- 0  
combined$BsmtHalfBath[is.na(combined$BsmtHalfBath)] <- 0  
  
sort(colSums(sapply(combined[,bsement], is.na)), decreasing = T)
```

```

##      BsmtQual      BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2 BsmtFullBath
##      0            0            0            0            0            0
## BsmtHalfBath BsmtFinSF1  BsmtFinSF2  BsmtUnfSF TotalBsmtSF
##      0            0            0            0            0

```

```

# label encoding

combined$BsmtQual<-as.integer(revalue(combined$BsmtQual, quality))

```

BsmtQual

The following 'from' values were not present in 'x': Po

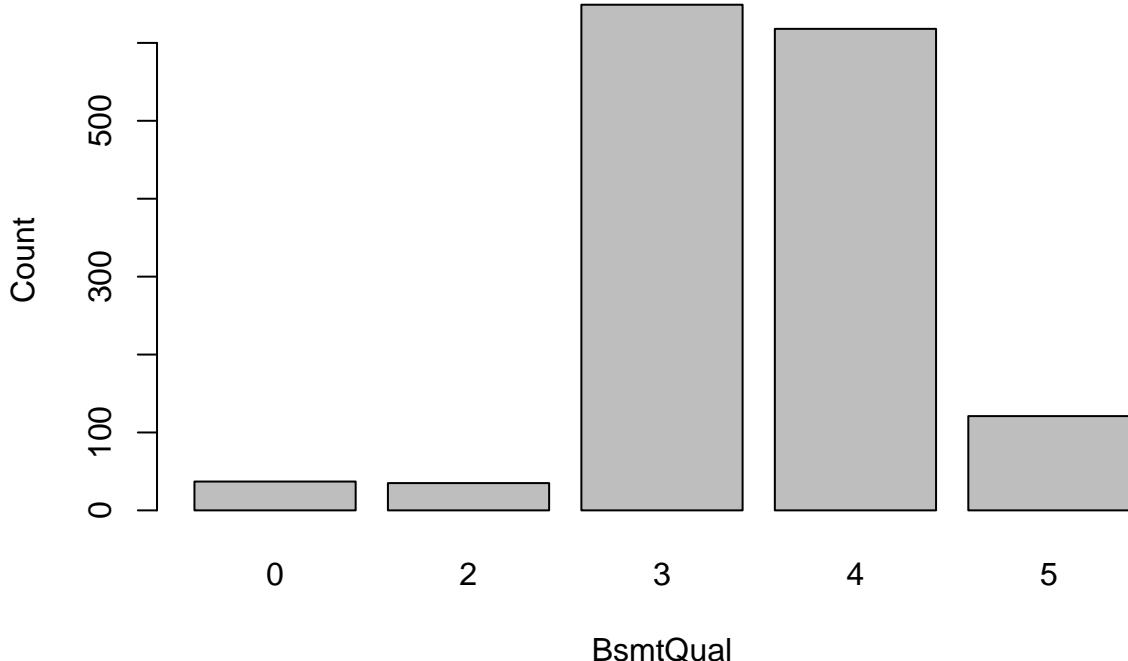
```
table(combined$BsmtQual, useNA = "ifany")
```

```

##
##   0   2   3   4   5
## 37  35 649 618 121

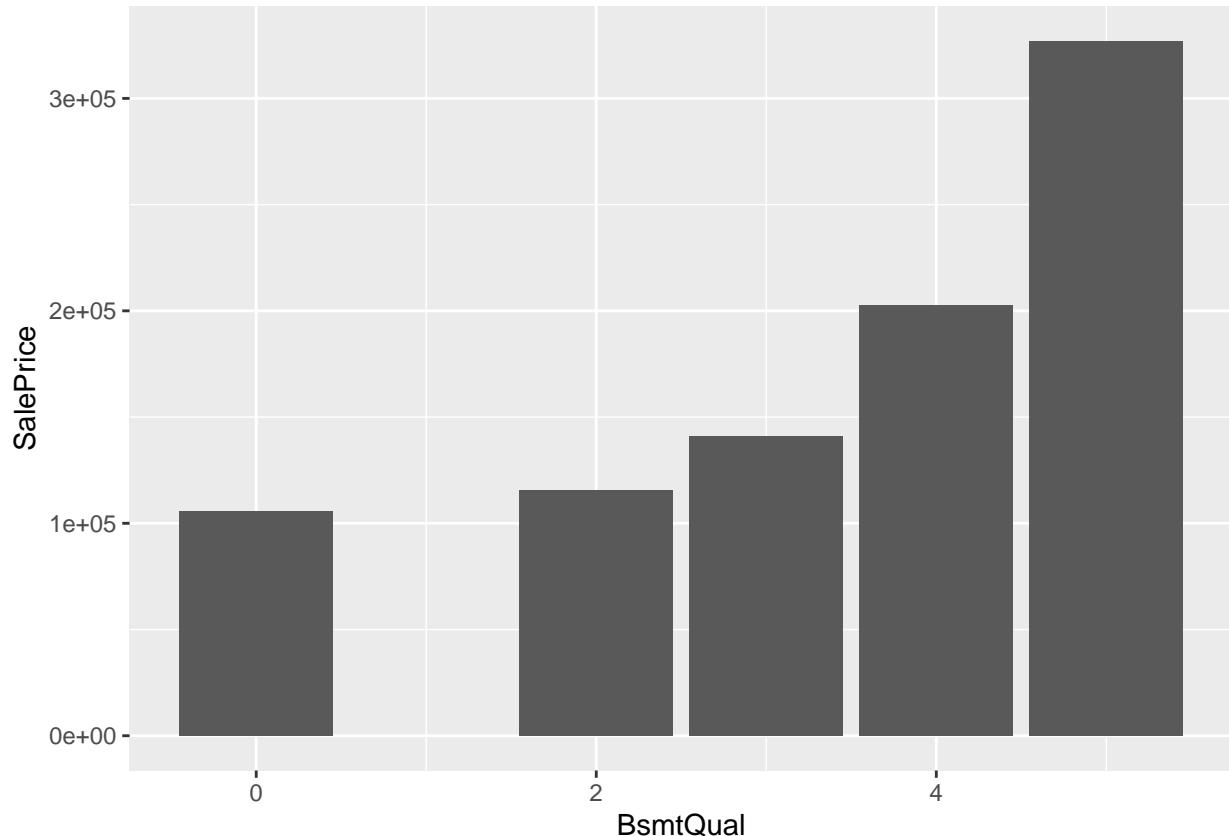
```

```
barplot(table(combined$BsmtQual), xlab = "BsmtQual", ylab = "Count")
```



```
ggplot(combined, aes(x=BsmtQual, y = SalePrice)) + geom_bar(stat = 'summary')
```

No summary function supplied, defaulting to 'mean_se()'



```
combined$BsmtCond<-as.integer(revalue(combined$BsmtCond, quality))
```

BsmtCond

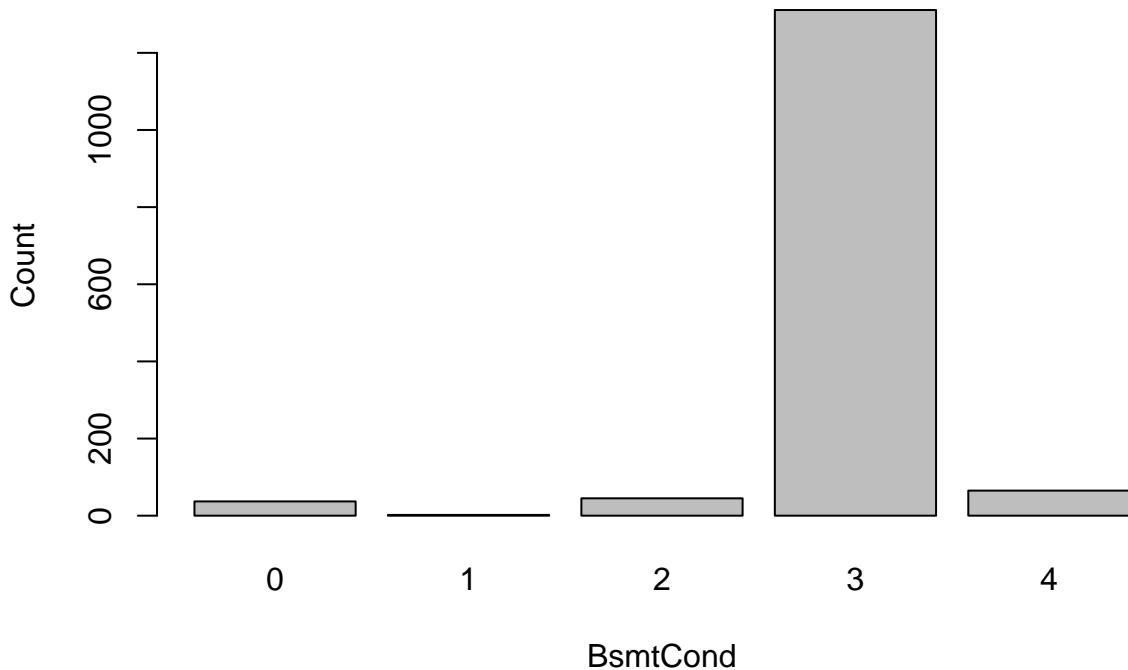
```
## The following 'from' values were not present in 'x': Ex
```

```
table(combined$BsmtCond, useNA = "ifany")
```

```
##
```

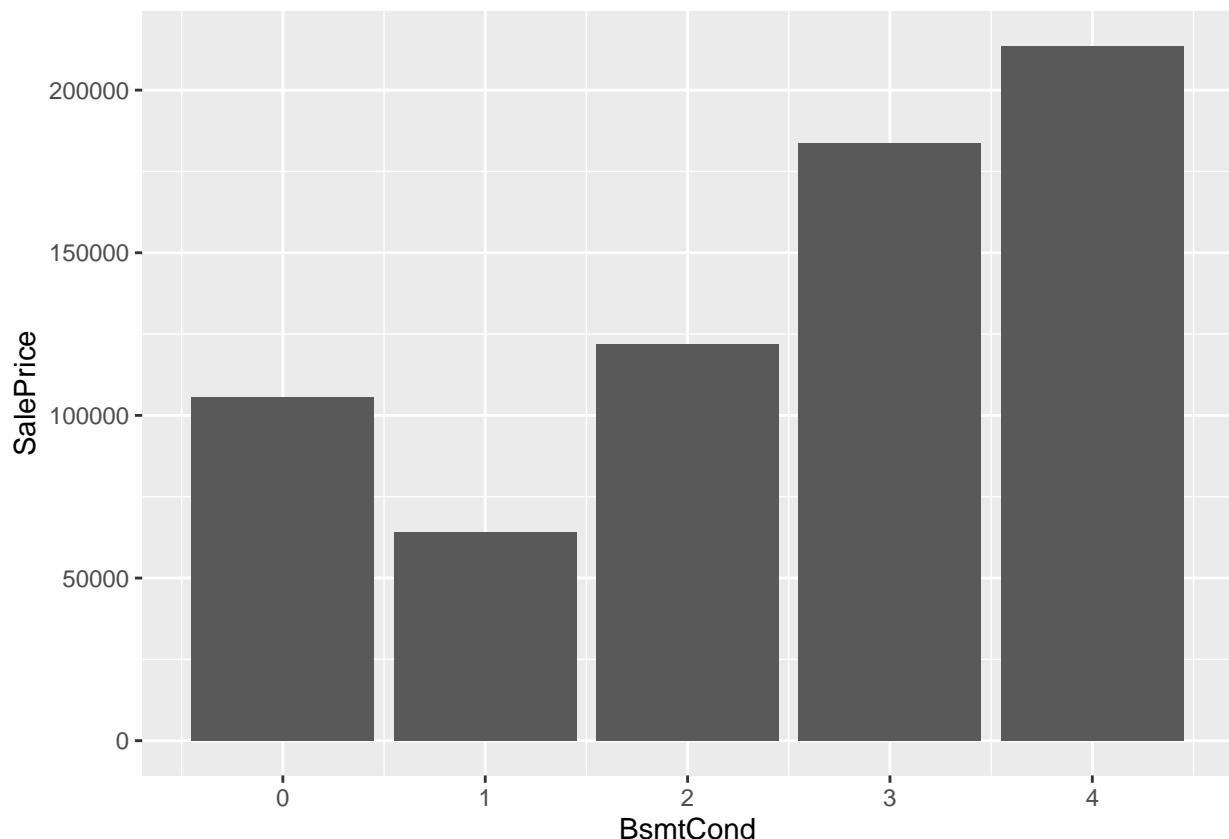
```
##      0      1      2      3      4
##    37     2    45  1311    65
```

```
barplot(table(combined$BsmtCond), xlab = "BsmtCond", ylab = "Count")
```



```
ggplot(combined, aes(x=BsmtCond, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



```

exposure <- c('None'=0, 'No'=1, 'Mn'=2, 'Av'=3, 'Gd'=4)
combined$BsmtExposure<-as.integer(revalue(combined$BsmtExposure, exposure))
table(combined$BsmtExposure, useNA = "ifany")

```

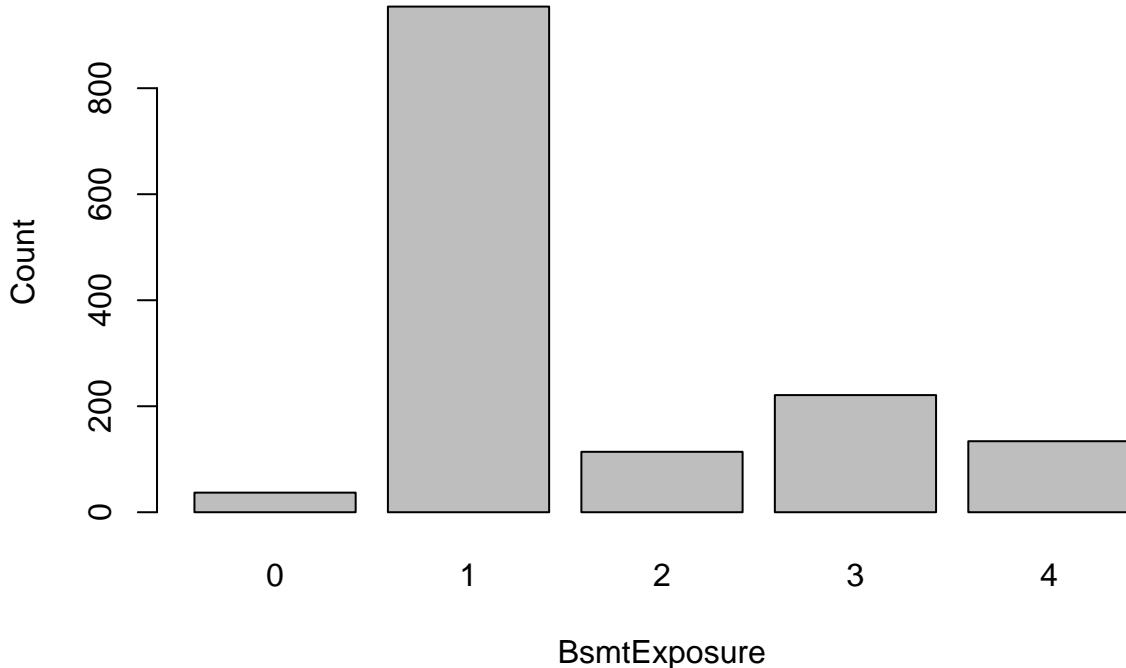
BsmtExposure

```

##
##   0   1   2   3   4
## 37 954 114 221 134

barplot(table(combined$BsmtExposure), xlab = "BsmtExposure", ylab = "Count")

```



```

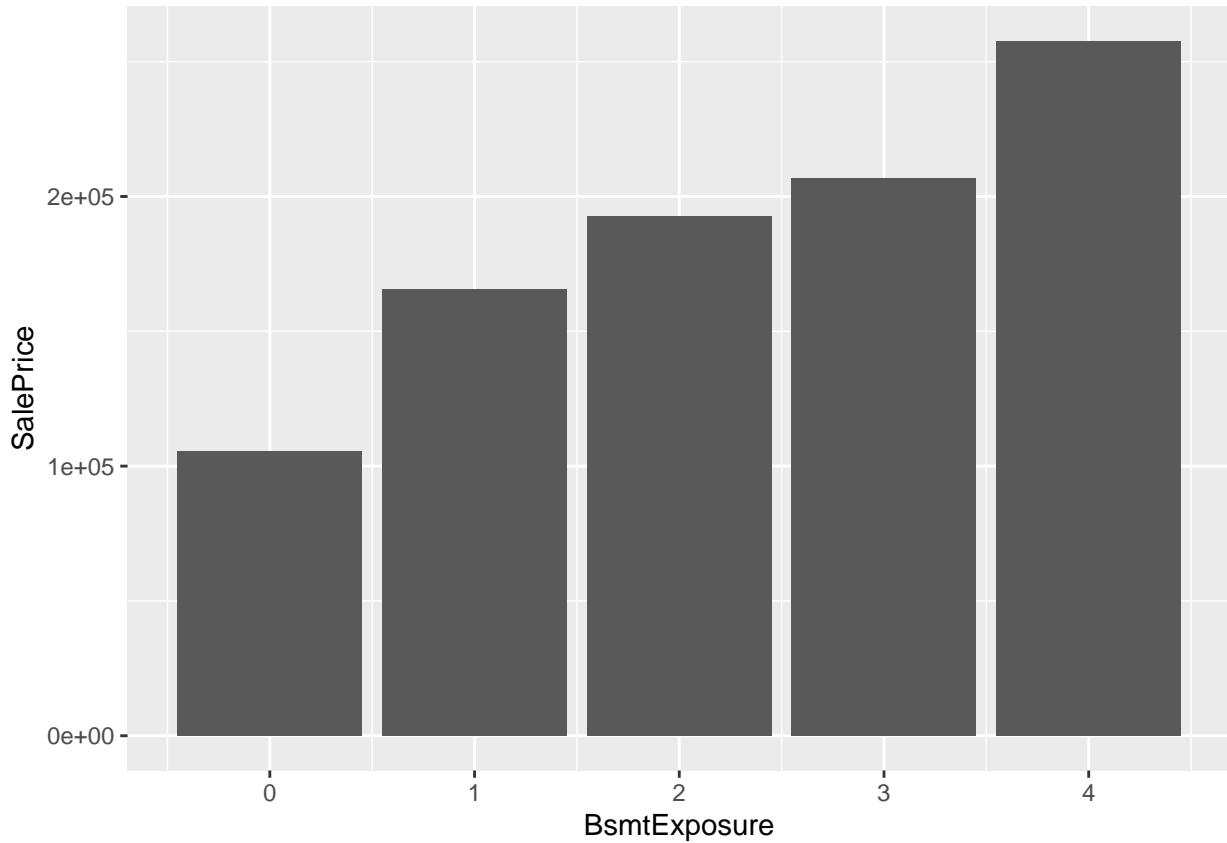
ggplot(combined, aes(x=BsmtExposure, y = SalePrice)) + geom_bar(stat = 'summary')

```

```

## No summary function supplied, defaulting to 'mean_se()'

```

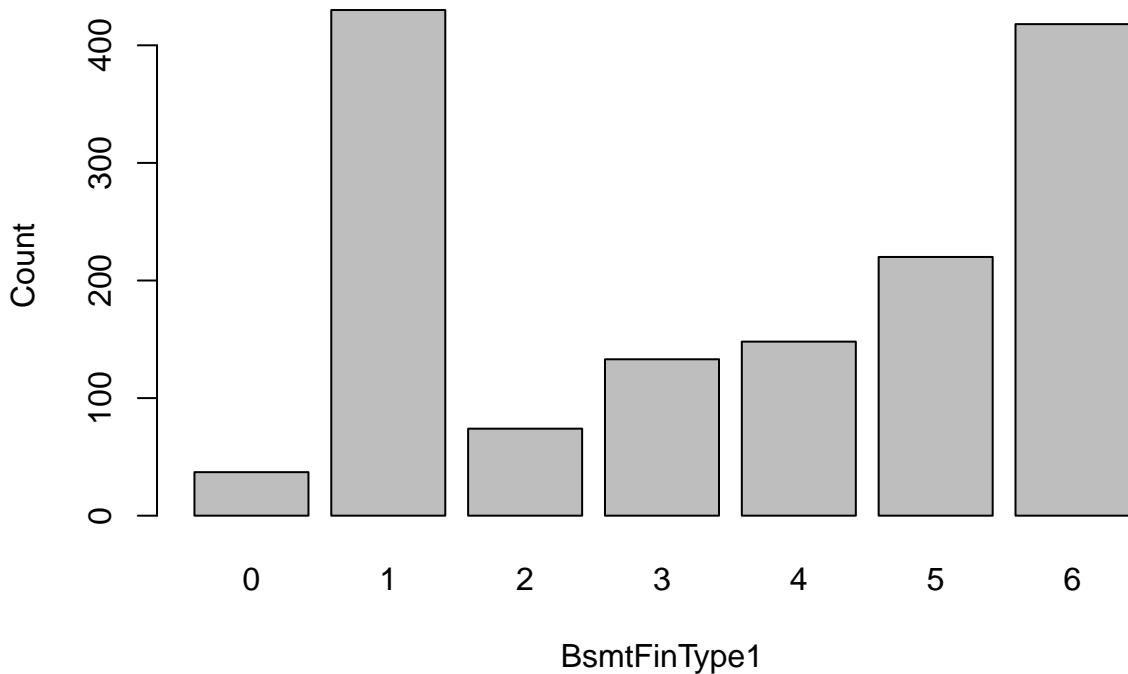


```
rating <- c('None'=0, 'Unf'=1, 'LwQ'=2, 'Rec'=3, 'BLQ'=4, 'ALQ'=5, 'GLQ'=6)
combined$BsmtFinType1<-as.integer(revalue(combined$BsmtFinType1, rating))
table(combined$BsmtFinType1, useNA = "ifany")
```

BsmtFinType1

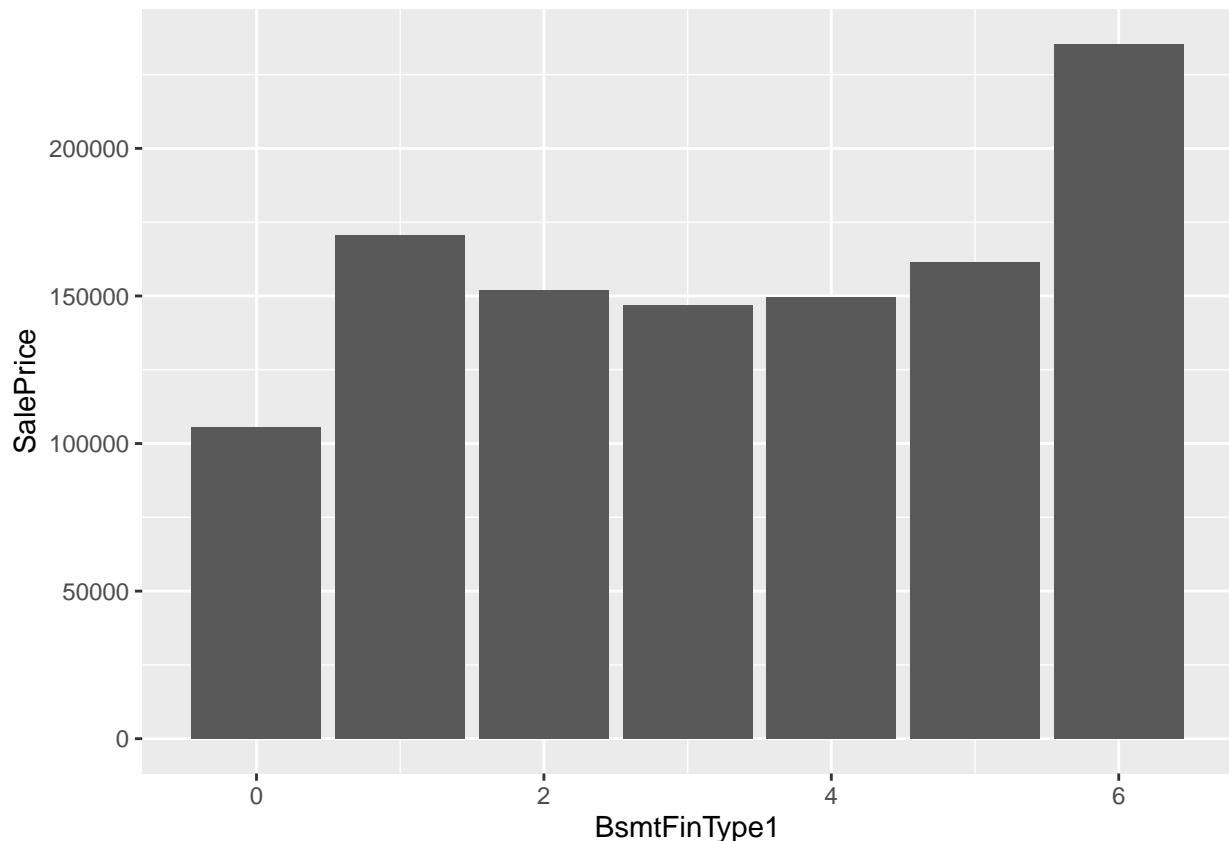
```
##
##    0    1    2    3    4    5    6
##  37  430   74  133  148  220  418

barplot(table(combined$BsmtFinType1), xlab = "BsmtFinType1", ylab = "Count")
```



```
ggplot(combined, aes(x=BsmtFinType1, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```

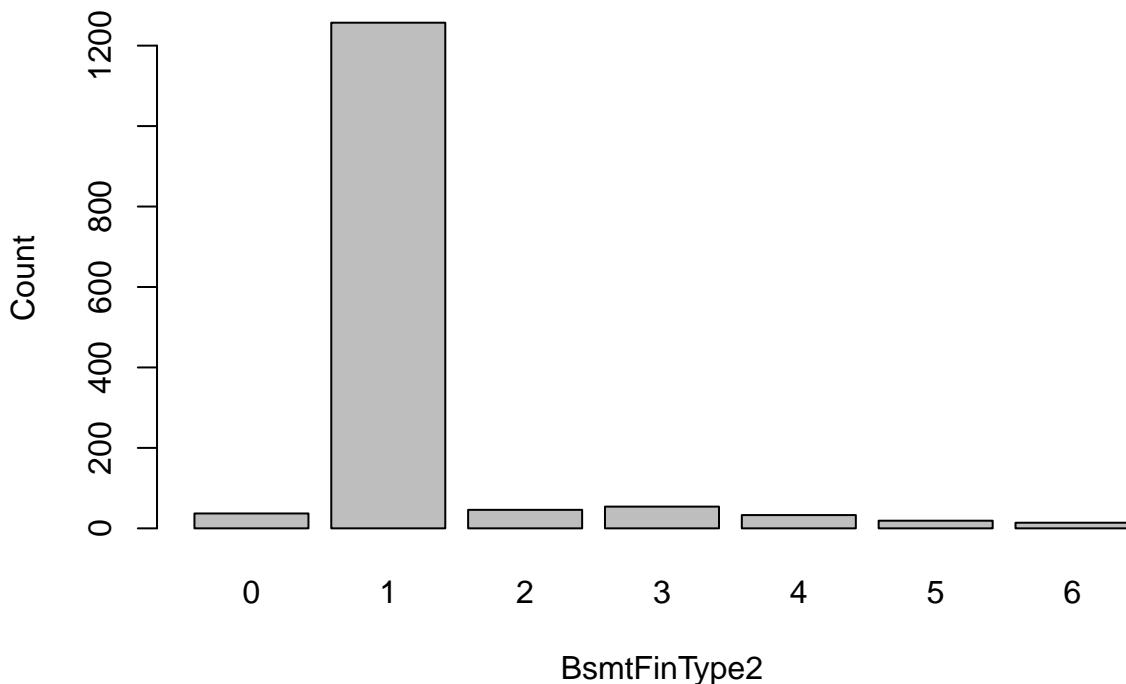


```
combined$BsmtFinType2<-as.integer(revalue(combined$BsmtFinType2, rating))
table(combined$BsmtFinType2, useNA = "ifany")
```

BsmtFinType2

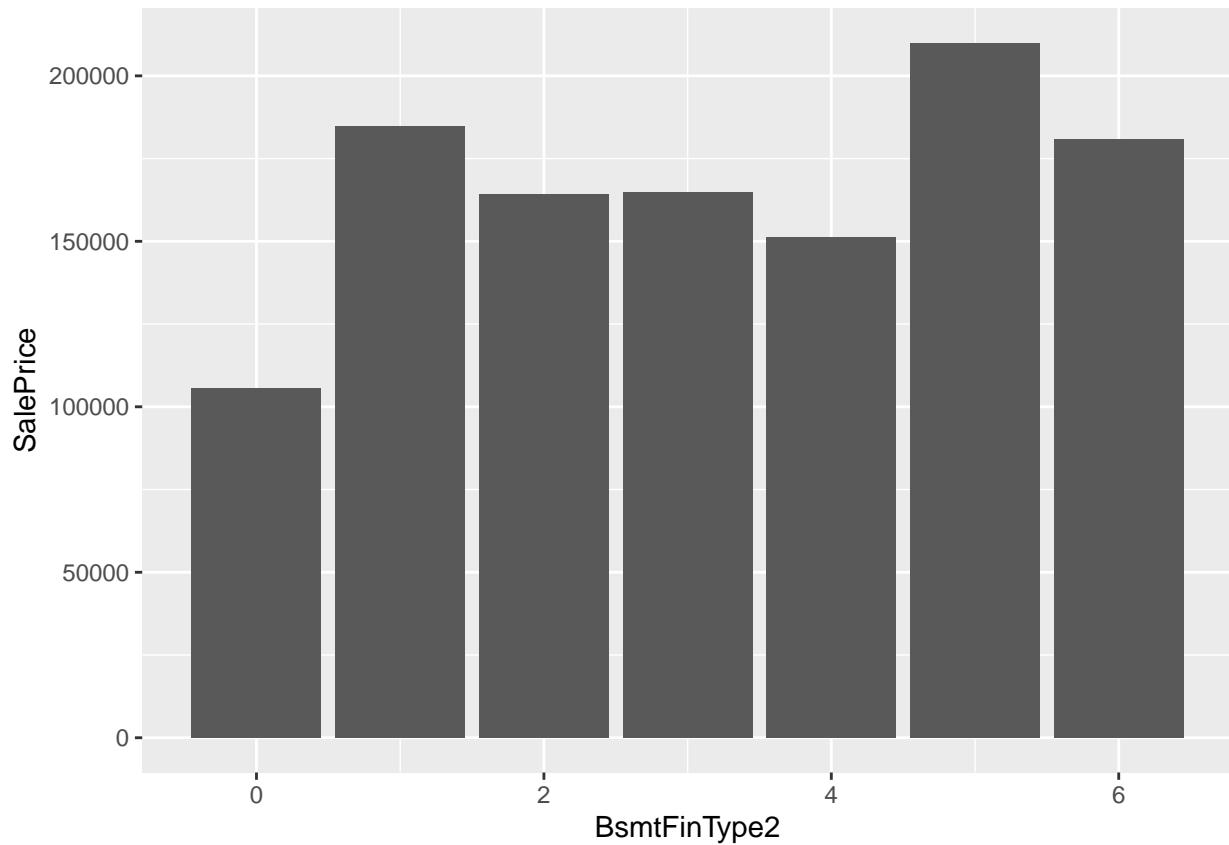
```
##  
##      0      1      2      3      4      5      6  
##    37 1257   46   54   33   19   14
```

```
barplot(table(combined$BsmtFinType2), xlab = "BsmtFinType2", ylab = "Count")
```

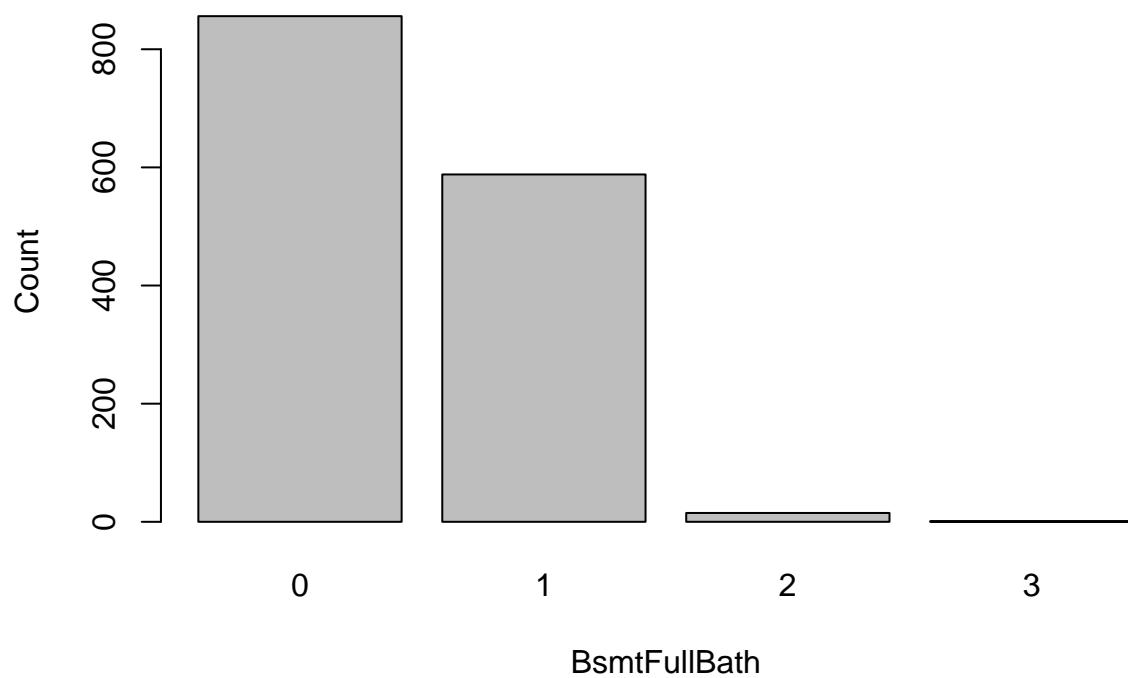


```
ggplot(combined, aes(x=BsmtFinType2, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



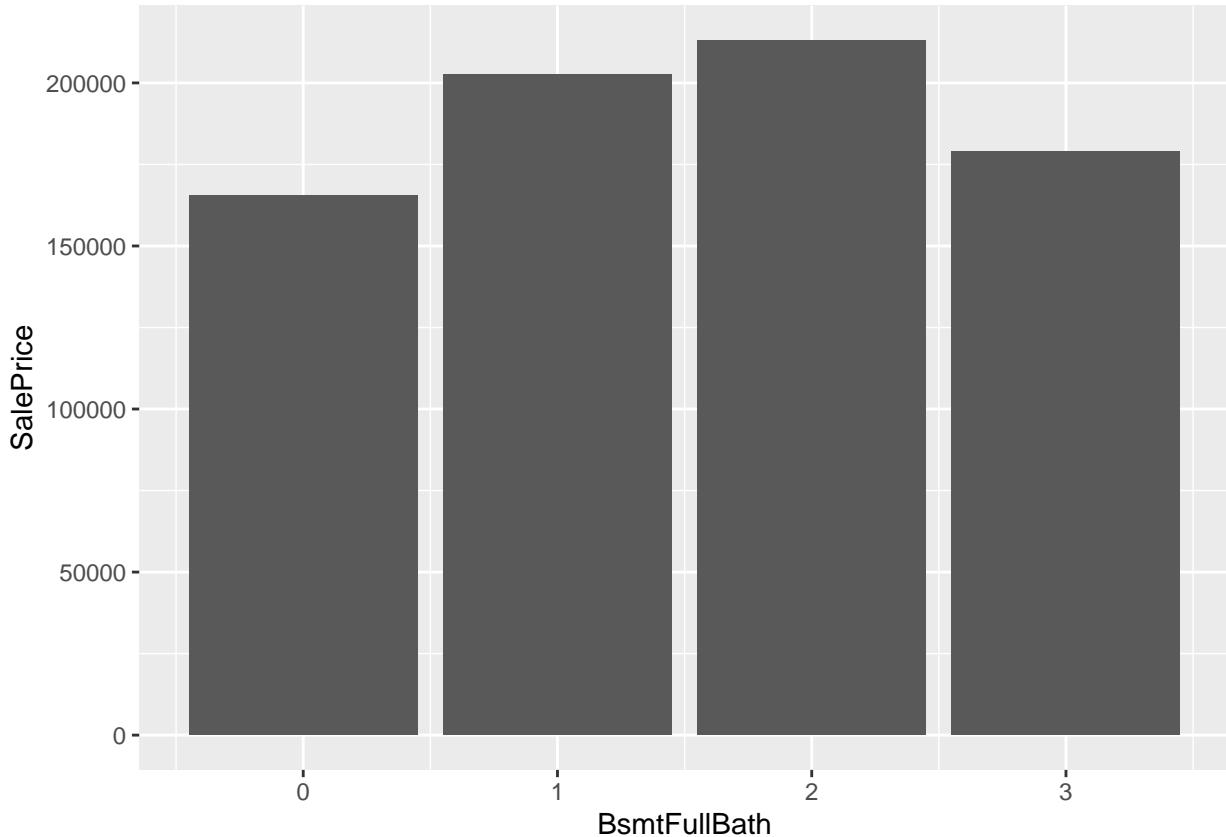
```
barplot(table(combined$BsmtFullBath), xlab = "BsmtFullBath", ylab = "Count")
```



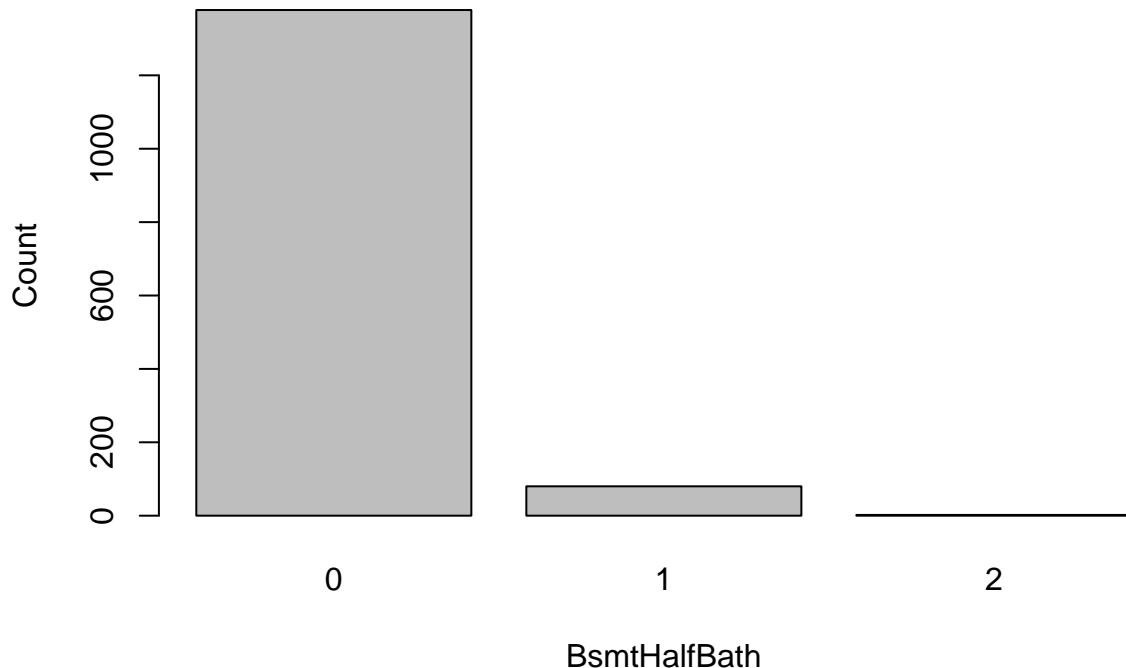
BsmtFullBath

```
ggplot(combined, aes(x=BsmtFullBath, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



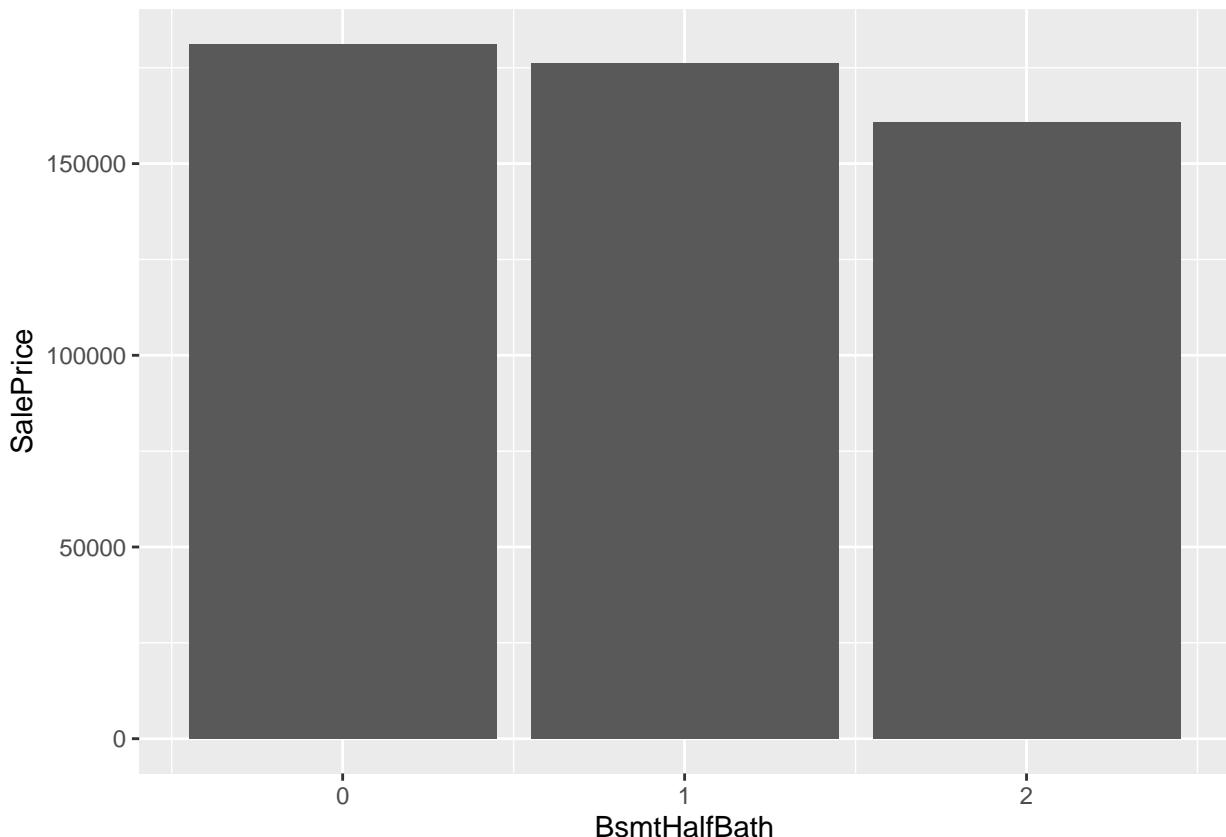
```
barplot(table(combined$BsmtHalfBath), xlab = "BsmtHalfBath", ylab = "Count")
```



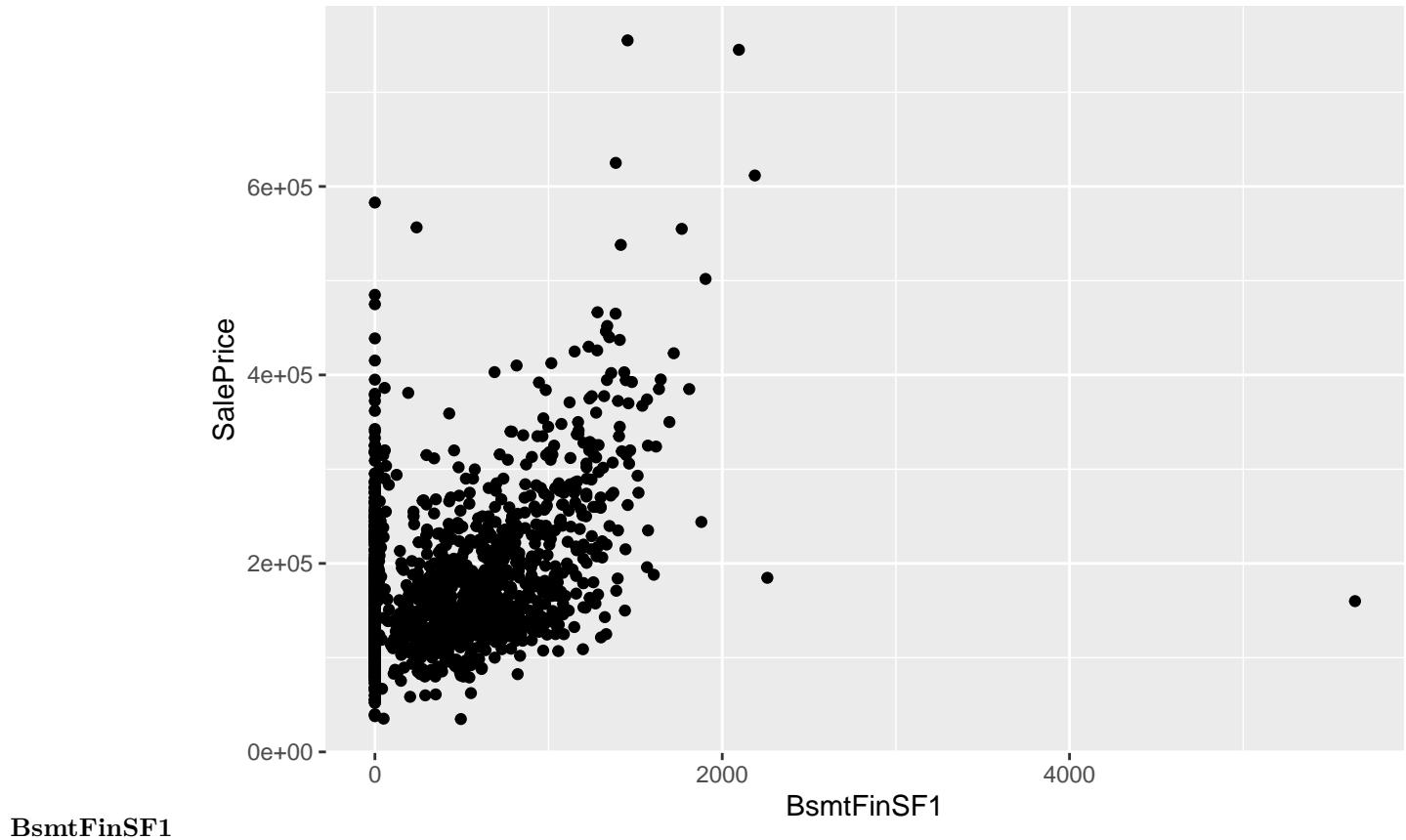
BsmtHalfBath

```
ggplot(combined, aes(x=BsmtHalfBath, y = SalePrice)) + geom_bar(stat = 'summary')
```

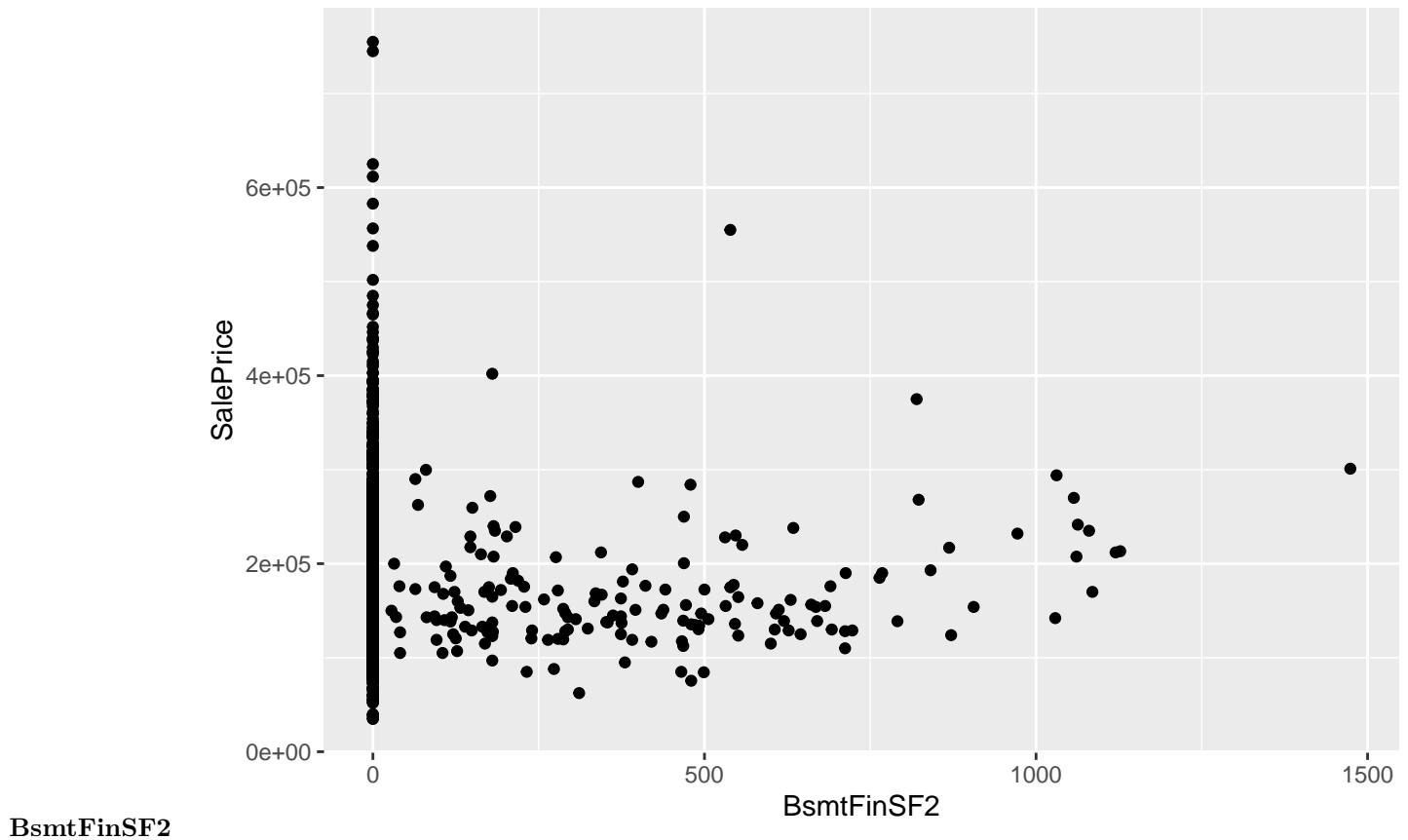
```
## No summary function supplied, defaulting to 'mean_se()'
```



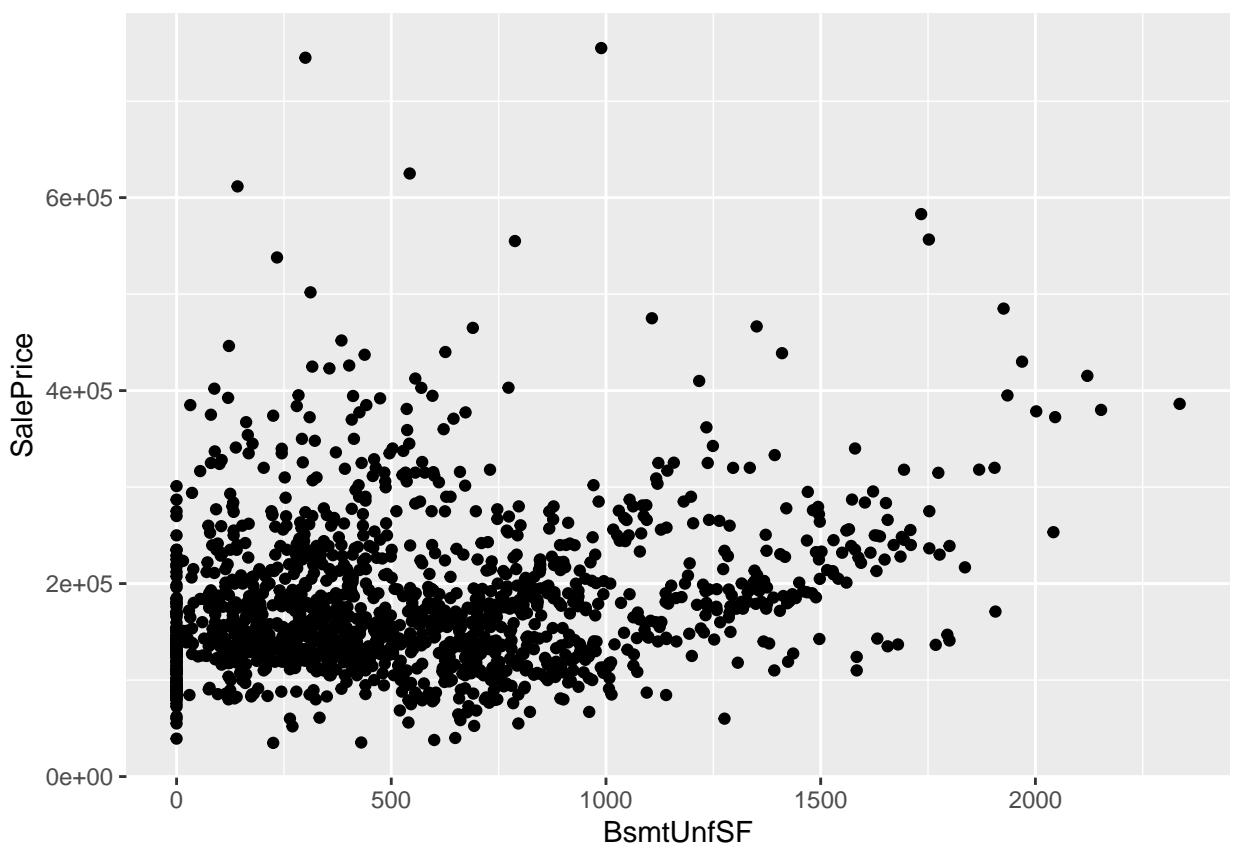
```
ggplot(combined, aes(x=BsmtFinSF1, y = SalePrice)) + geom_point()
```



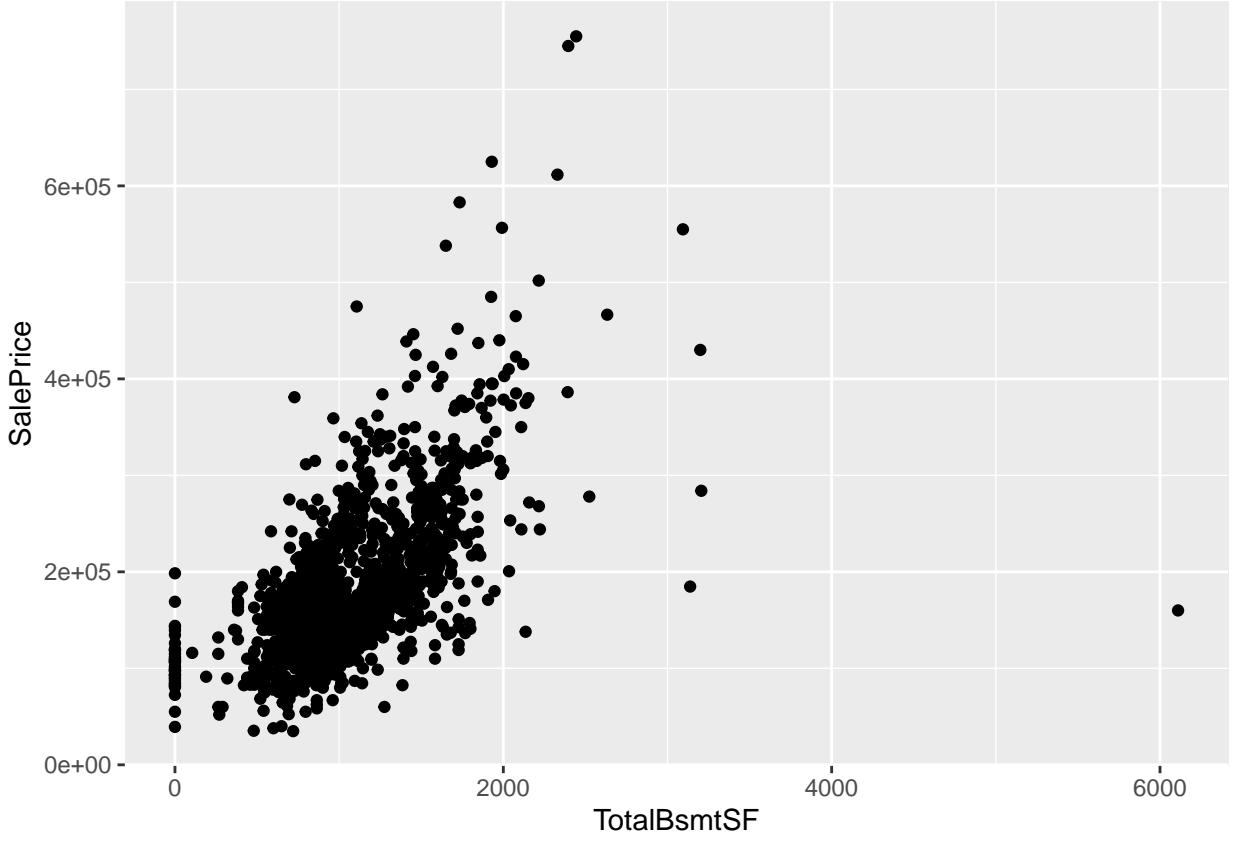
```
ggplot(combined, aes(x=BsmtFinSF2, y = SalePrice)) + geom_point()
```



```
ggplot(combined, aes(x=BsmtUnfSF, y = SalePrice)) + geom_point()
```



```
ggplot(combined, aes(x=TotalBsmtSF, y = SalePrice)) + geom_point()
```



TotalBsmtSF

masonry variables

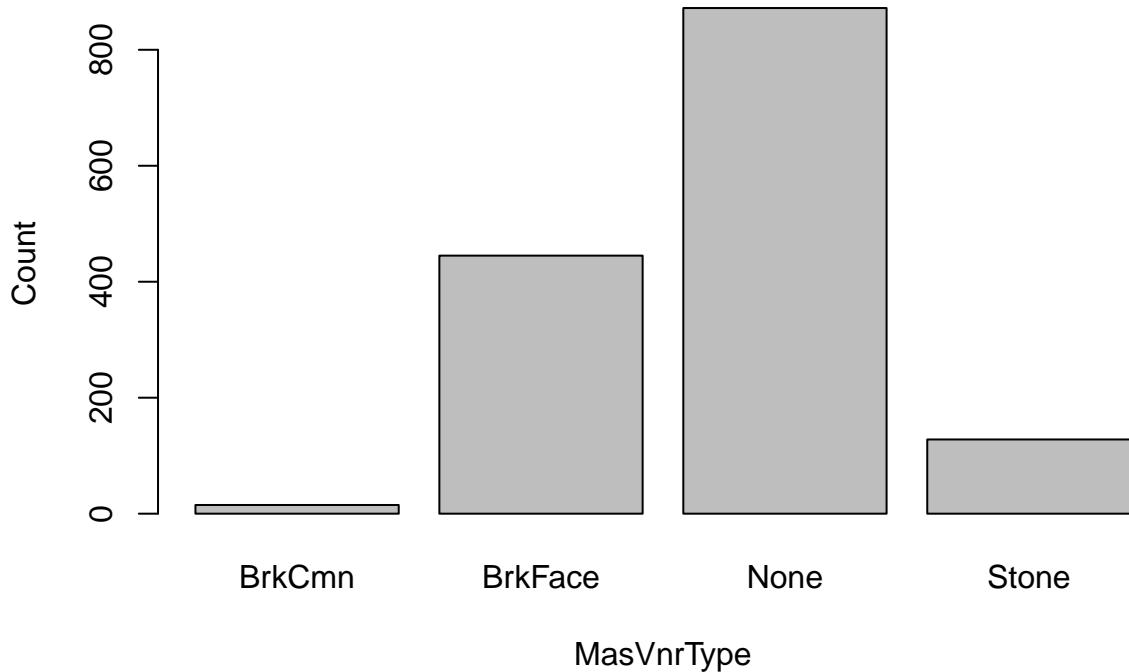
```
table(combined$MasVnrType)
```

```
##  
## BrkCmn BrkFace None Stone  
##     15     445    864   128
```

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

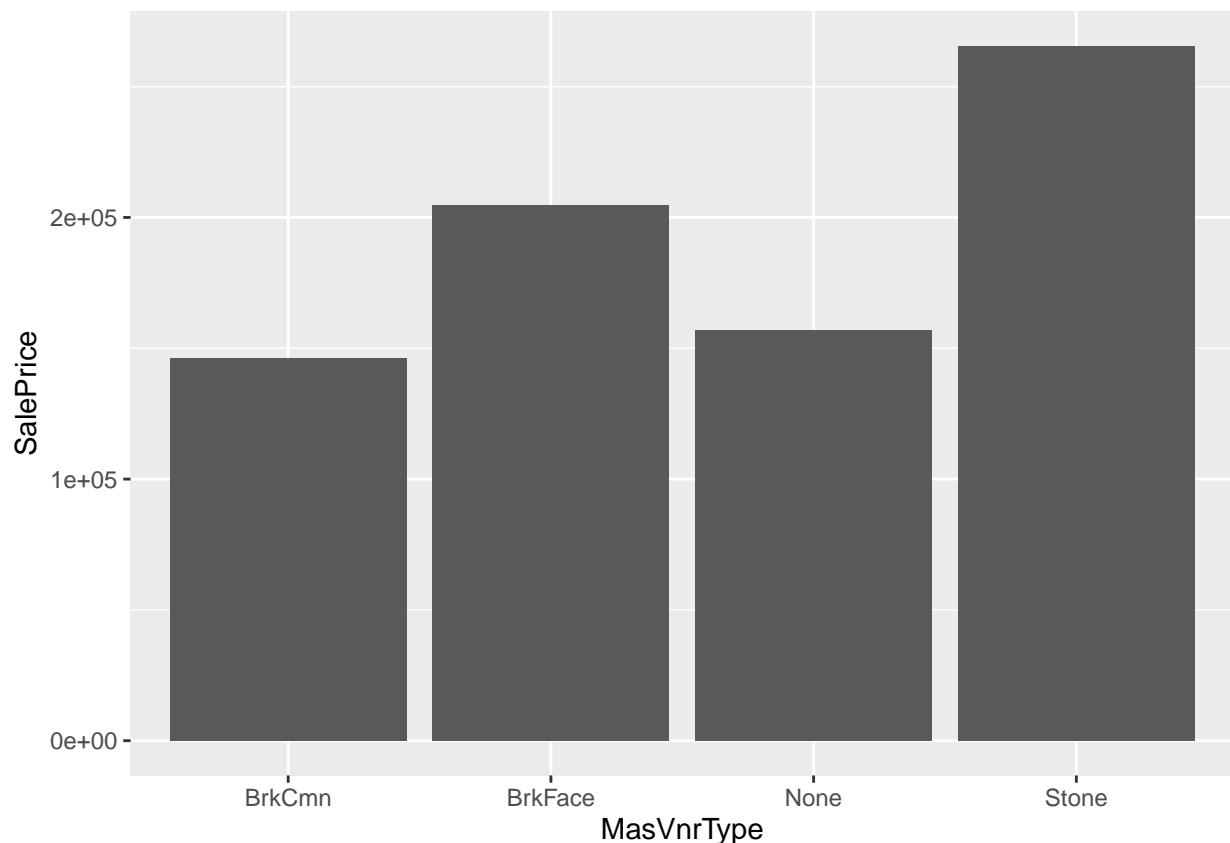
```
##  
## BrkCmn BrkFace None Stone  
##     15     445    872   128
```

```
barplot(table(combined$MasVnrType), xlab = "MasVnrType", ylab = "Count")
```



```
ggplot(combined, aes(x=MasVnrType, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



Electrical variable

```
table(combined$Electrical)

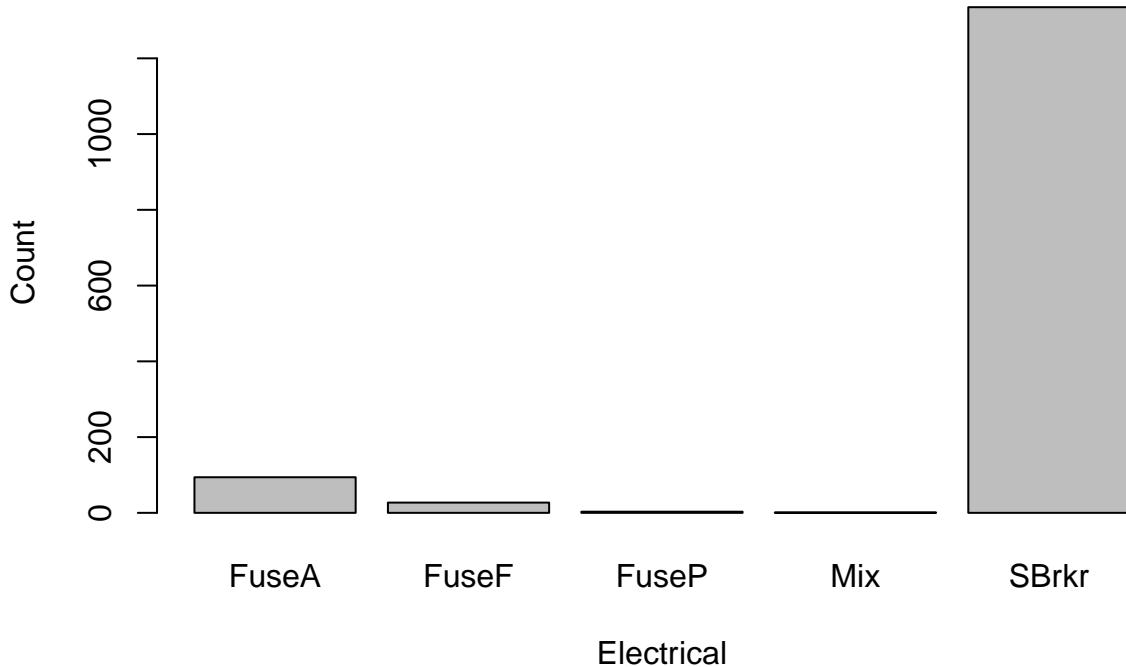
##
## FuseA FuseF FuseP   Mix SBrkr
##    94     27      3     1 1334

combined$Electrical[is.na(combined$Electrical)] <- names(sort(-table(combined$Electrical)))[1]

combined$Electrical <- as.factor(combined$Electrical)
table(combined$Electrical)

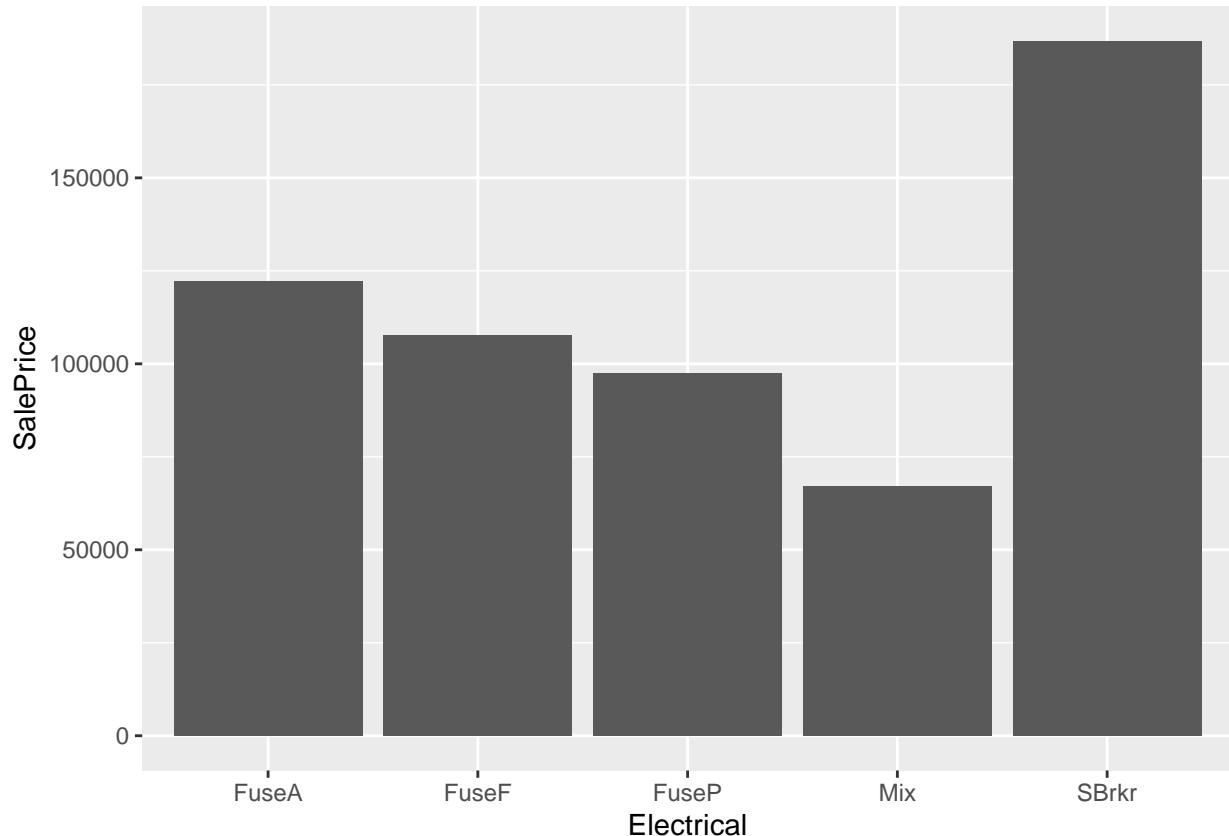
##
## FuseA FuseF FuseP   Mix SBrkr
##    94     27      3     1 1335

barplot(table(combined$Electrical), xlab = "Electrical", ylab = "Count")
```



```
ggplot(combined, aes(x=Electrical, y = SalePrice)) + geom_bar(stat = 'summary')

## No summary function supplied, defaulting to 'mean_se()'
```



MSZoning

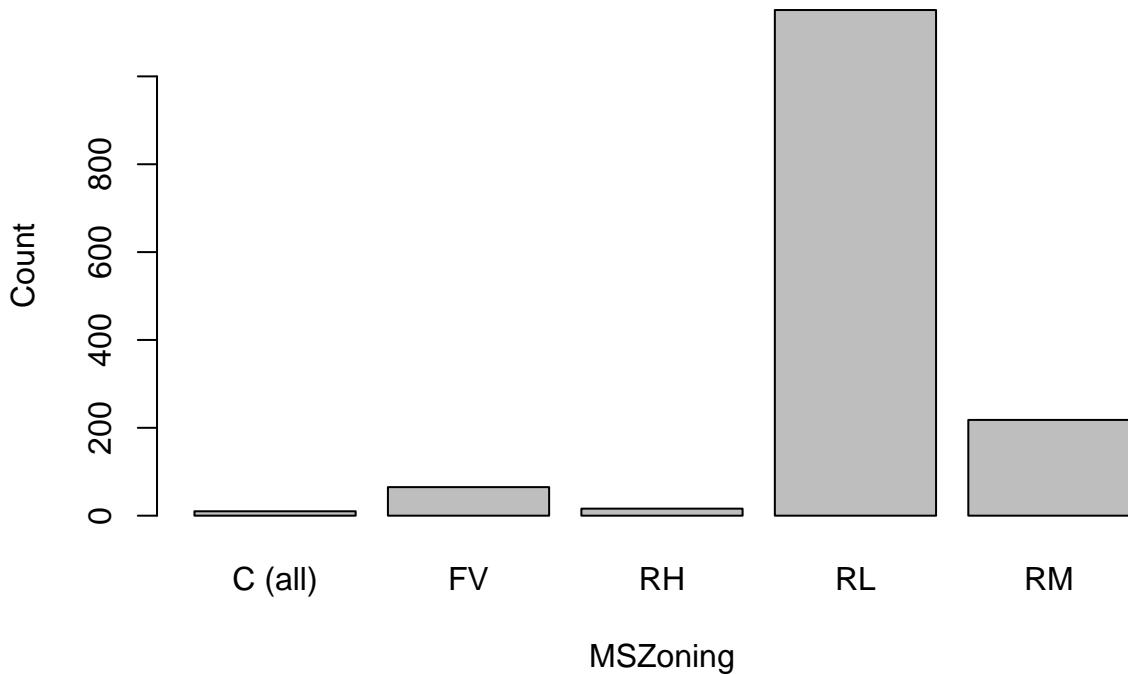
```
table(combined$MSZoning)
```

```
##  
## C (all)      FV       RH       RL       RM  
##      10       65       16     1151      218
```

```
combined$MSZoning <- as.factor(combined$MSZoning)  
table(combined$MSZoning)
```

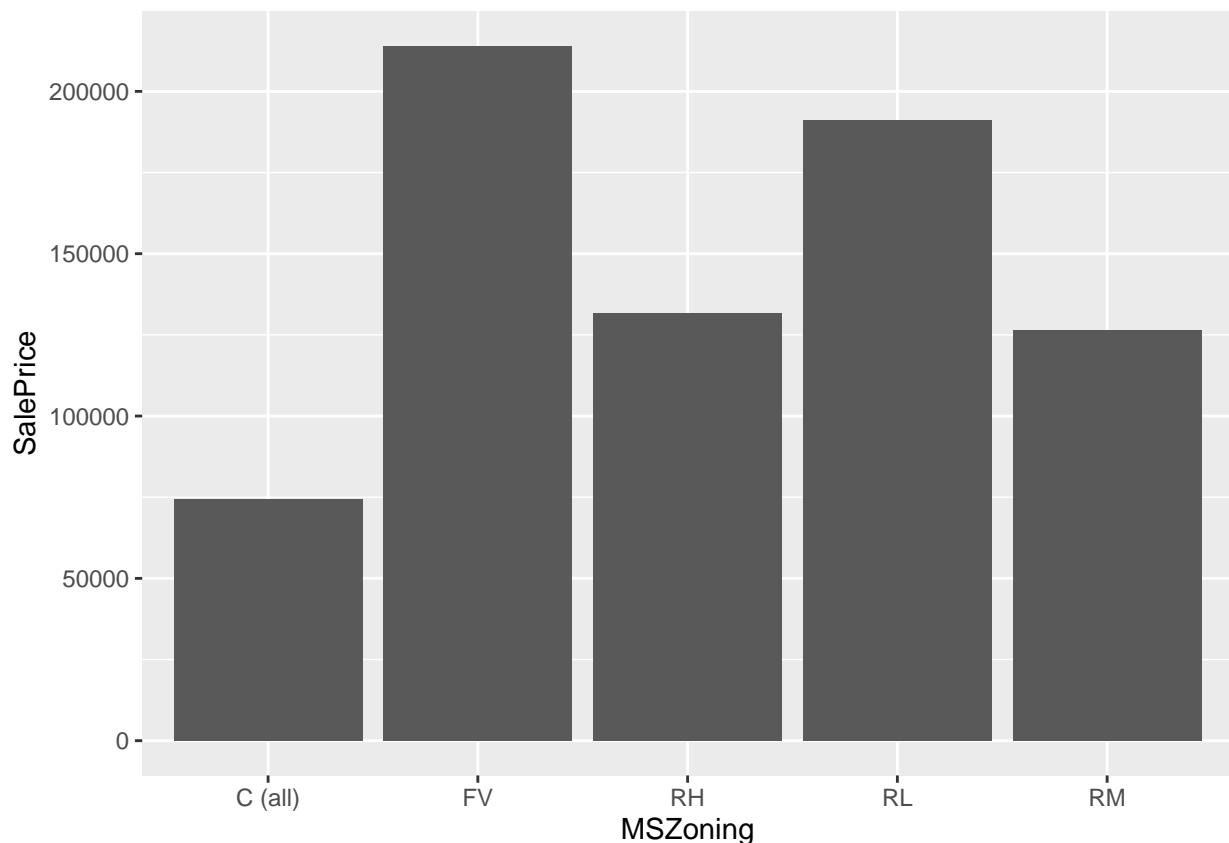
```
##  
## C (all)      FV       RH       RL       RM  
##      10       65       16     1151      218
```

```
barplot(table(combined$MSZoning), xlab = "MSZoning", ylab = "Count")
```



```
ggplot(combined, aes(x=MSZoning, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



Street

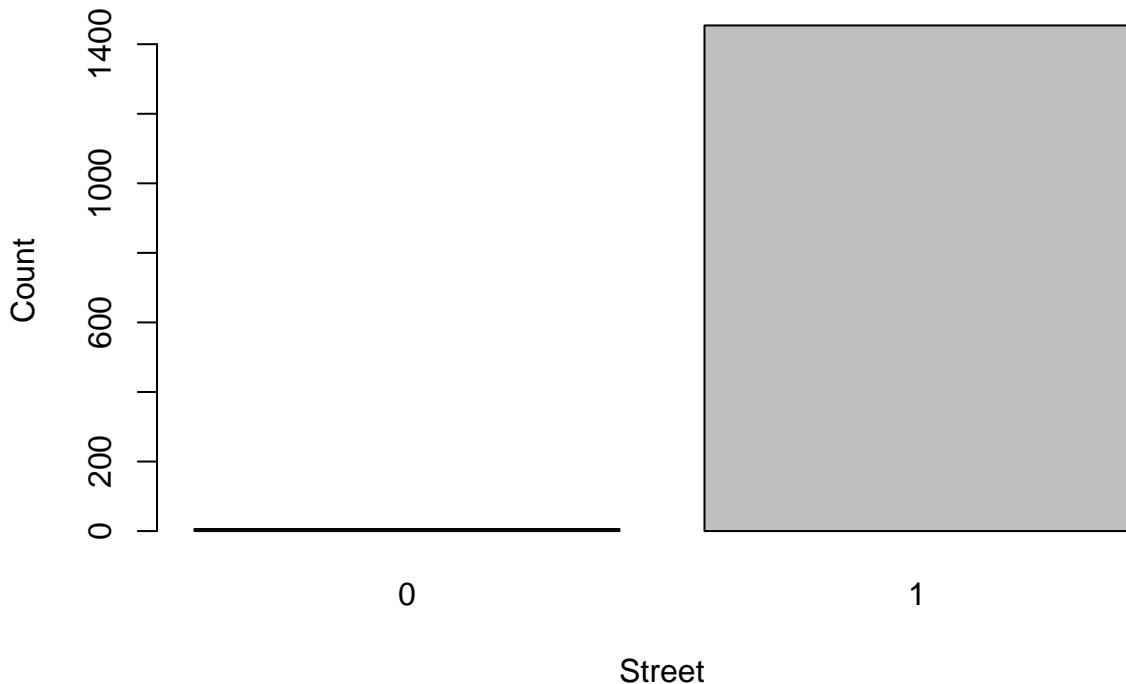
```
table(combined$Street)

##
## Grvl Pave
##      0   1
##      6 1454

combined$Street<-as.integer(revalue(combined$Street, c('Grvl'=0, 'Pave'=1)))
table(combined$Street)

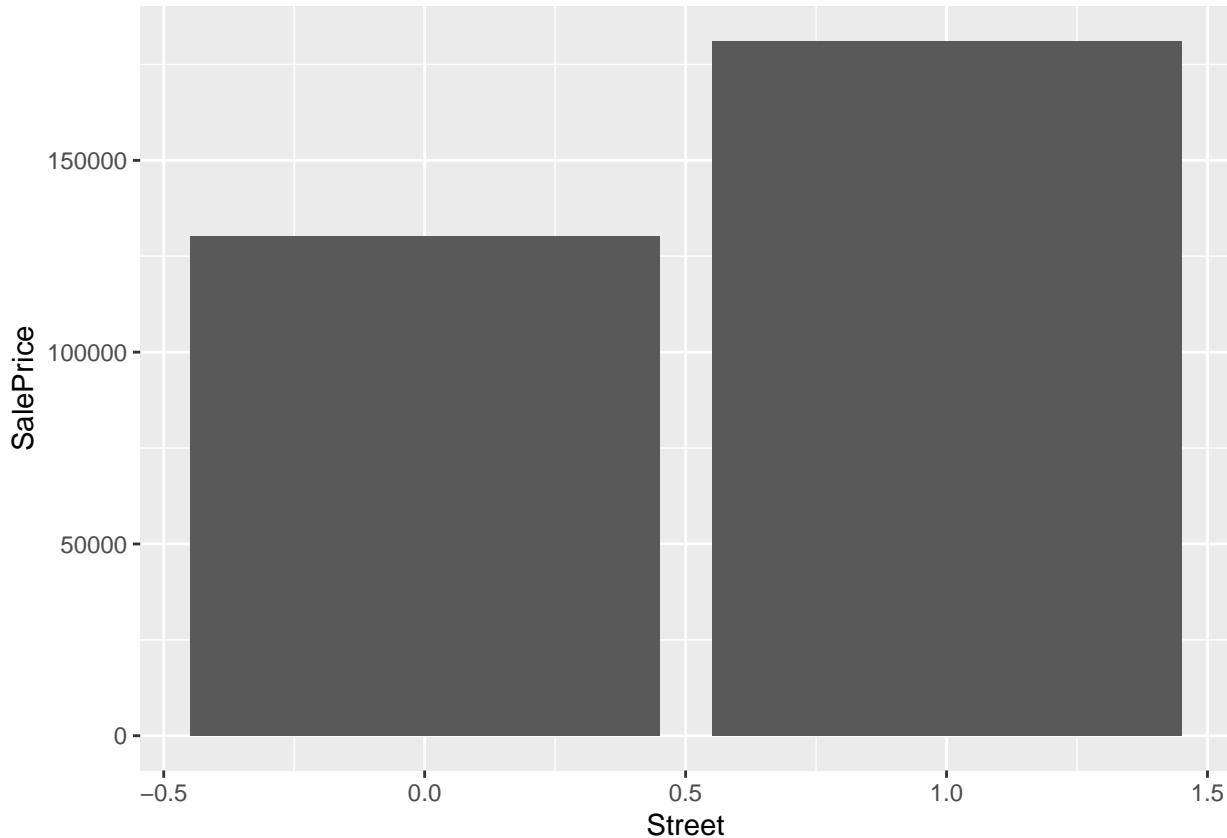
##
##      0      1
##      6 1454

barplot(table(combined$Street), xlab = "Street", ylab = "Count")
```



```
ggplot(combined, aes(x=Street, y = SalePrice)) + geom_bar(stat = 'summary')

## No summary function supplied, defaulting to 'mean_se()'
```



LandContour

Lvl Near Flat/Level Bnk Banked - Quick and significant rise from street grade to building HLS Hillside - Significant slope from side to side Low Depression

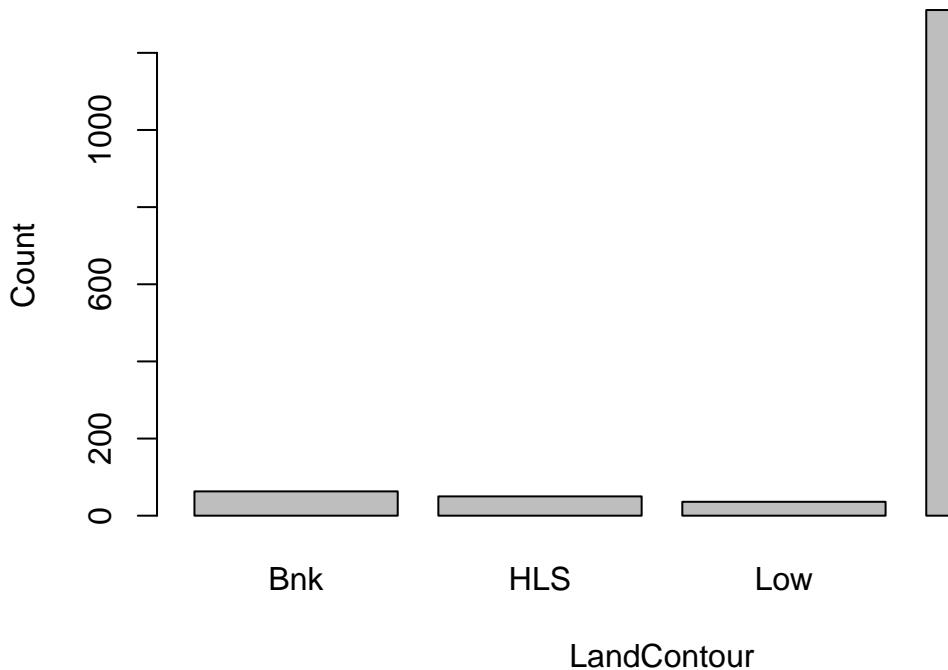
```
table(combined$LandContour)
```

```
##  
##   Bnk   HLS   Low   Lvl  
##   63    50    36  1311
```

```
combined$LandContour <- as.factor(combined$LandContour)  
table(combined$LandContour)
```

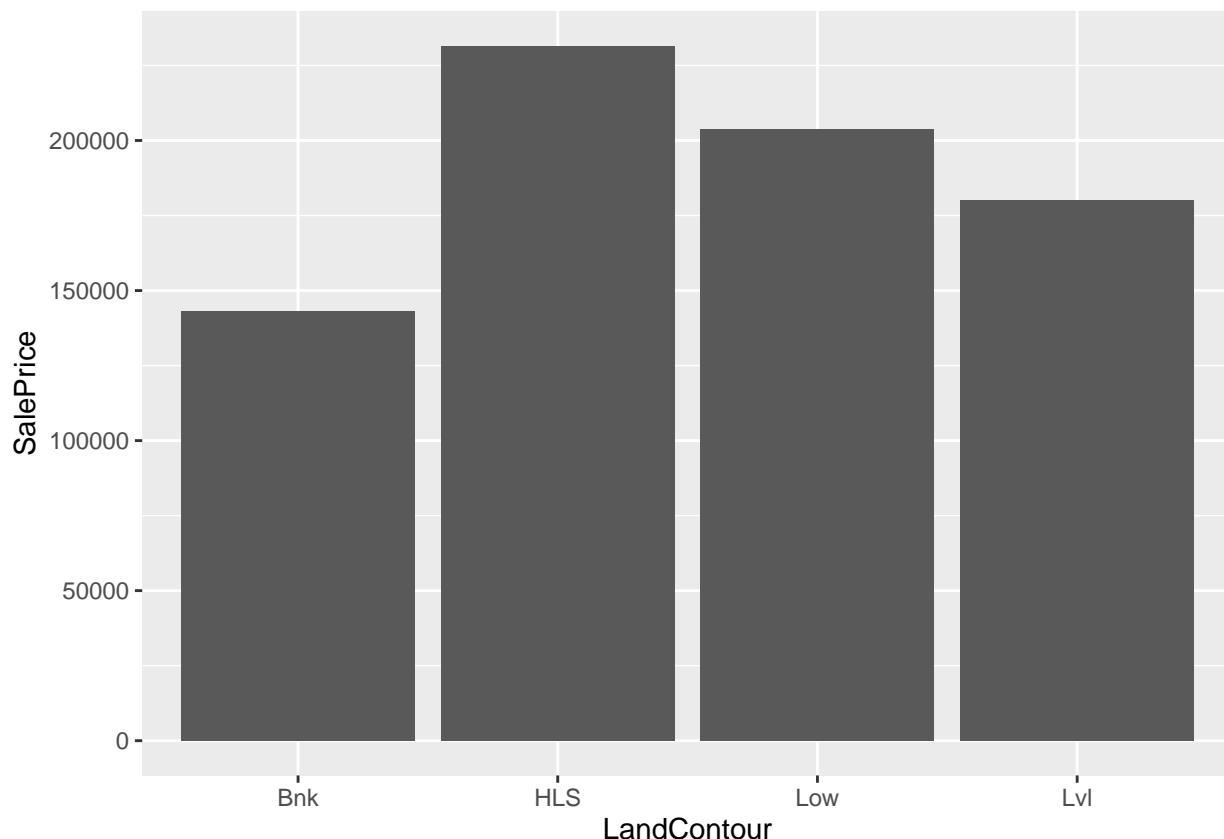
```
##  
##   Bnk   HLS   Low   Lvl  
##   63    50    36  1311
```

```
barplot(table(combined$LandContour), xlab = "LandContour", ylab = "Count")
```



```
ggplot(combined, aes(x=LandContour, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



LandSlope

Gtl Gentle slope Mod Moderate Slope
Sev Severe Slope

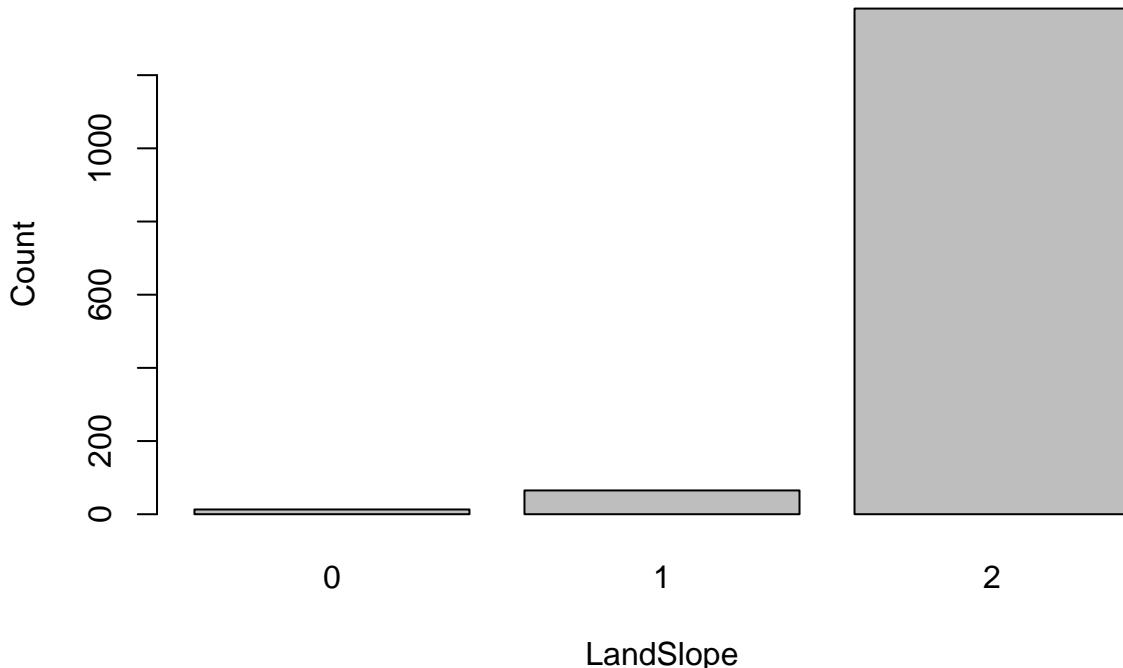
```
table(combined$LandSlope)
```

```
##  
##   Gtl  Mod  Sev  
## 1382    65    13
```

```
combined$LandSlope<-as.integer(revalue(combined$LandSlope, c('Sev'=0, 'Mod'=1, 'Gtl'=2)))  
table(combined$LandSlope)
```

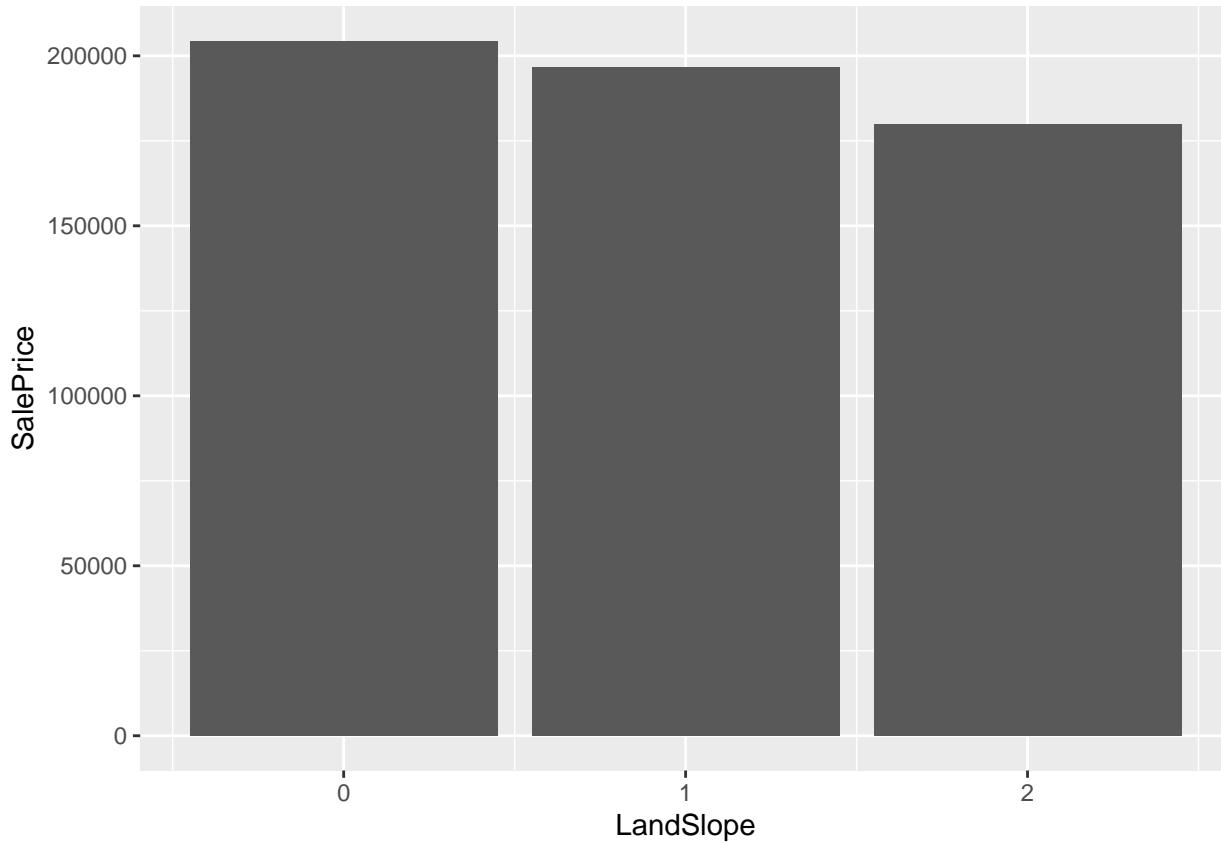
```
##  
##     0     1     2  
##    13    65 1382
```

```
barplot(table(combined$LandSlope), xlab = "LandSlope", ylab = "Count")
```



```
ggplot(combined, aes(x=LandSlope, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



Utilities

```
table(combined$Utilities)

##
## AllPub NoSeWa
##    1459      1

# remove utilities as it does not give any valuable information.
combined <- combined[, !names(combined) %in% "Utilities"]
```

Neighborhood

```
table(combined$Neighborhood)

##
## Blmgtn Blueste BrDale BrkSide ClearCr CollgCr Crawfor Edwards Gilbert IDOTRR
##    17      2     16     58     28    150     51    100     79     37
## MeadowV Mitchel NAmes NoRidge NPkVill NridgHt NWAmes OldTown Sawyer SawyerW
##    17     49    225     41      9     77     73    113     74     59
## Somerst StoneBr SWISU Timber Veenker
##    86     25     25     38     11

combined$Neighborhood <- as.factor(combined$Neighborhood)
table(combined$Neighborhood)
```

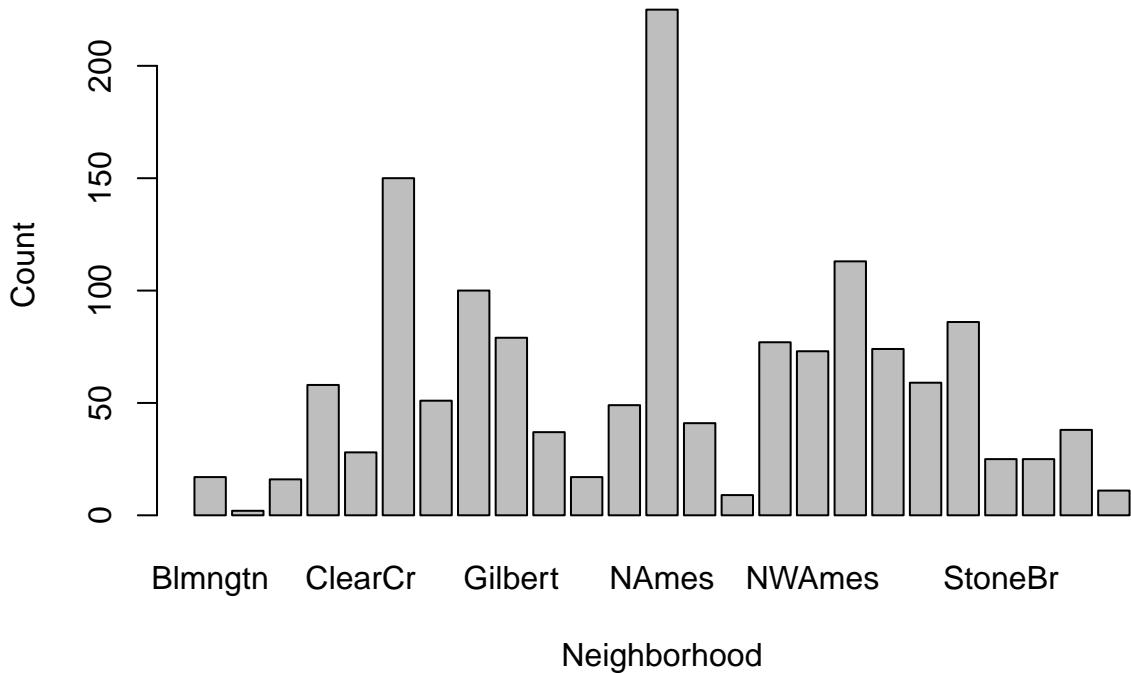
```
##
```

```

## Blmngtn Blueste BrDale BrkSide ClearCr CollgCr Crawfor Edwards Gilbert IDOTRR
##     17      2     16     58     28    150     51    100     79     37
## MeadowV Mitchel NAmes NoRidge NPkVill NridgHt NWAmes OldTown Sawyer SawyerW
##     17     49    225     41      9     77     73   113     74     59
## Somerst StoneBr SWISU Timber Veenker
##     86     25     25     38     11

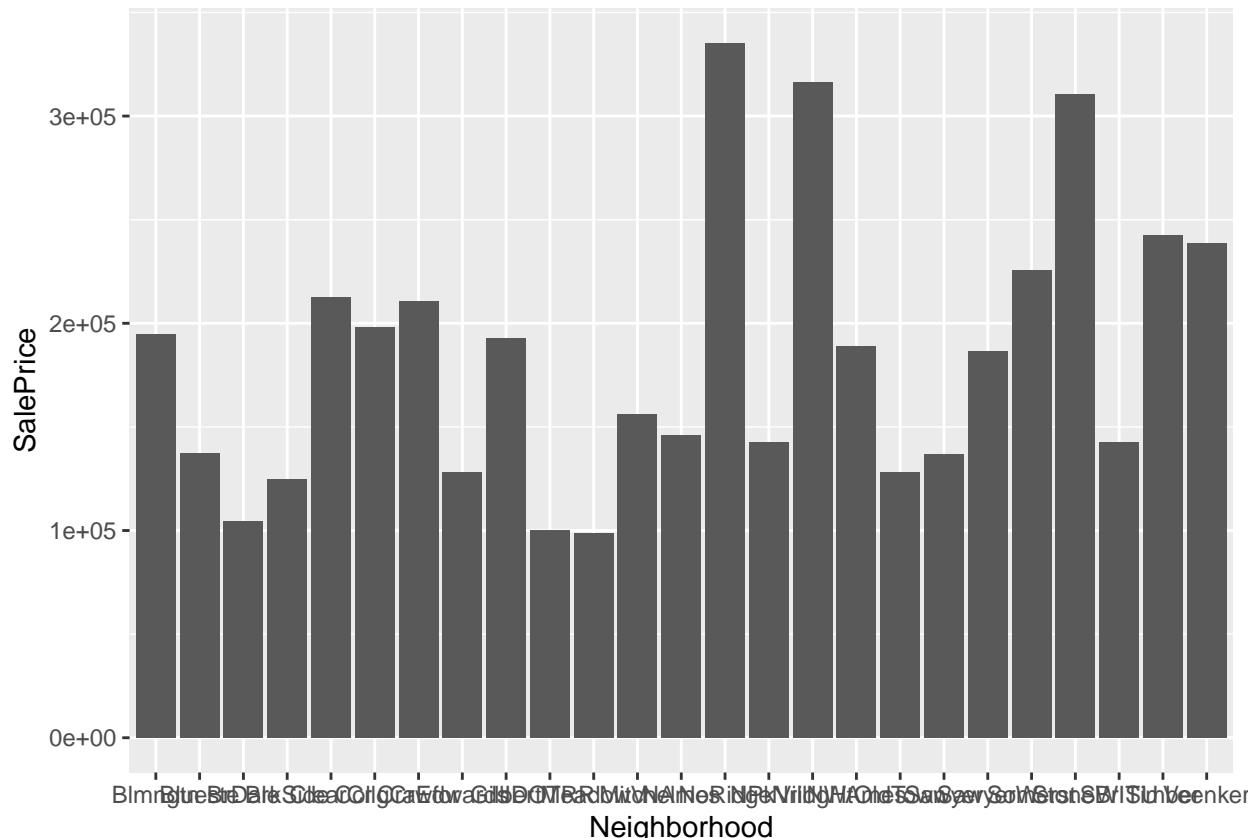
```

```
barplot(table(combined$Neighborhood), xlab = "Neighborhood", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=Neighborhood, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



Condition1 and Condition2

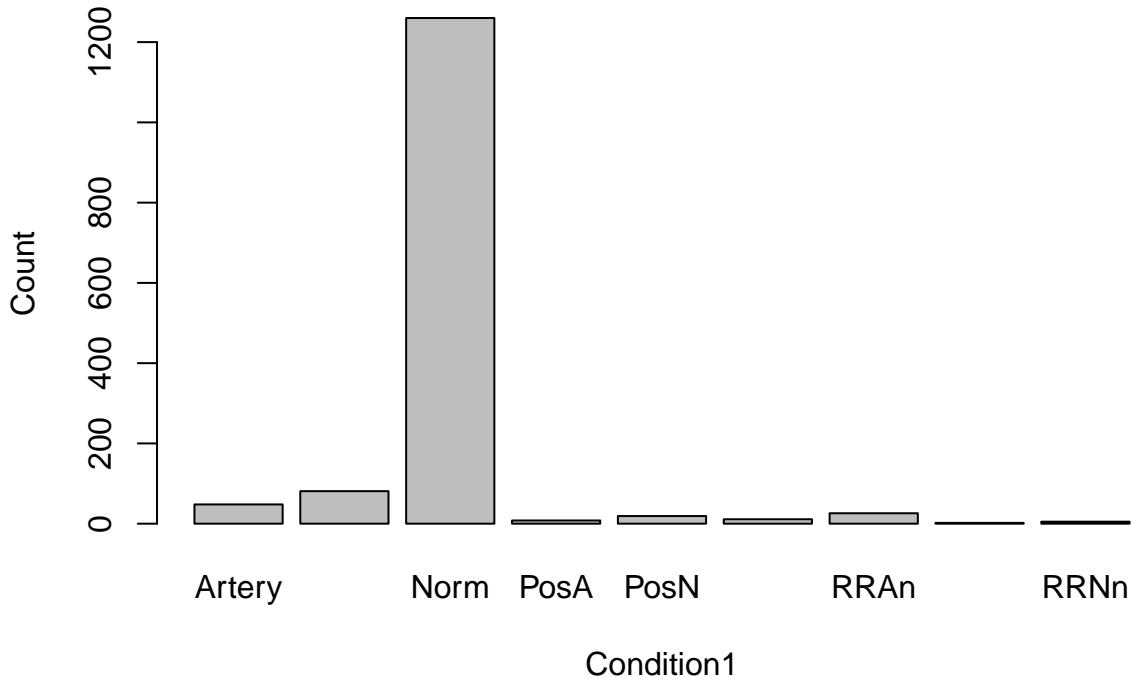
```
table(combined$Condition1)
```

```
## Artery Feedr Norm PosA PosN RRAe RRAn RRNe RRNn
##    48     81 1260     8    19     11     26      2      5
```

```
combined$Condition1 <- as.factor(combined$Condition1)
table(combined$Condition1)
```

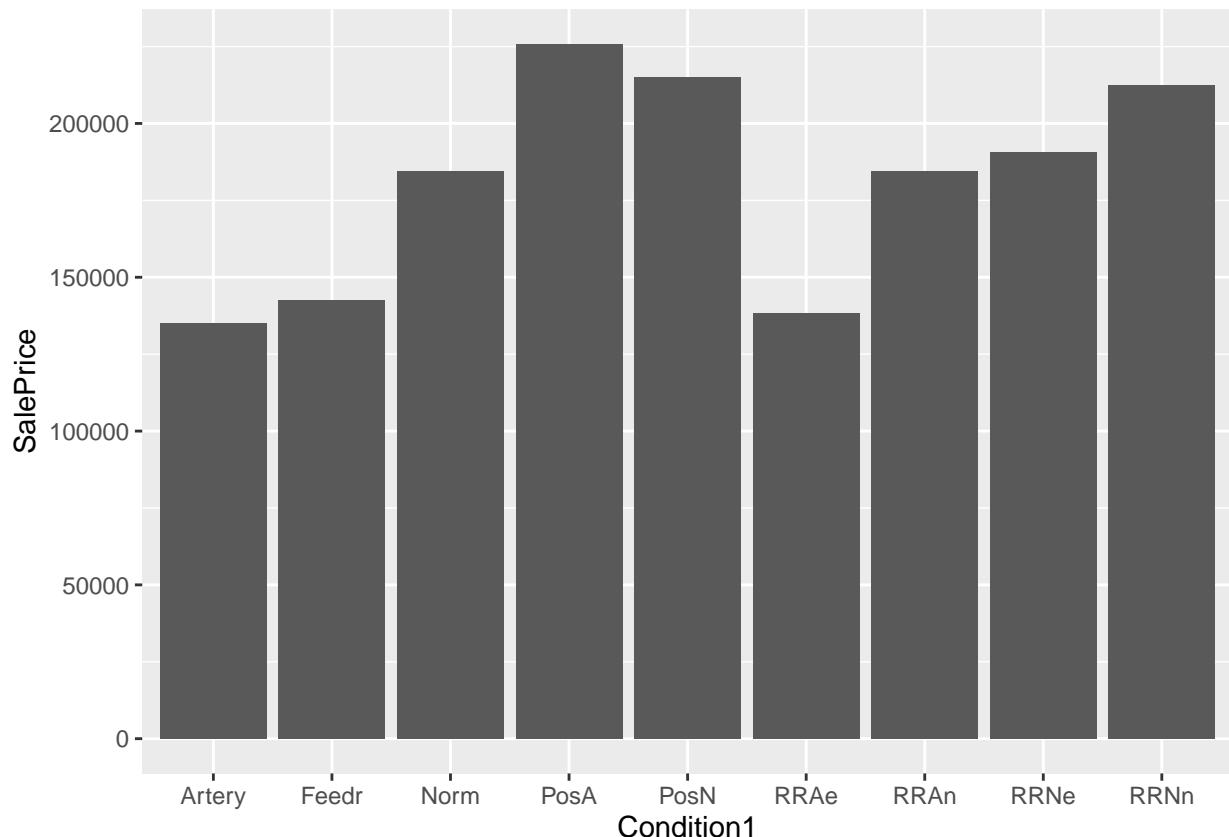
```
## Artery Feedr Norm PosA PosN RRAe RRAn RRNe RRNn
##    48     81 1260     8    19     11     26      2      5
```

```
barplot(table(combined$Condition1), xlab = "Condition1", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=Condition1, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



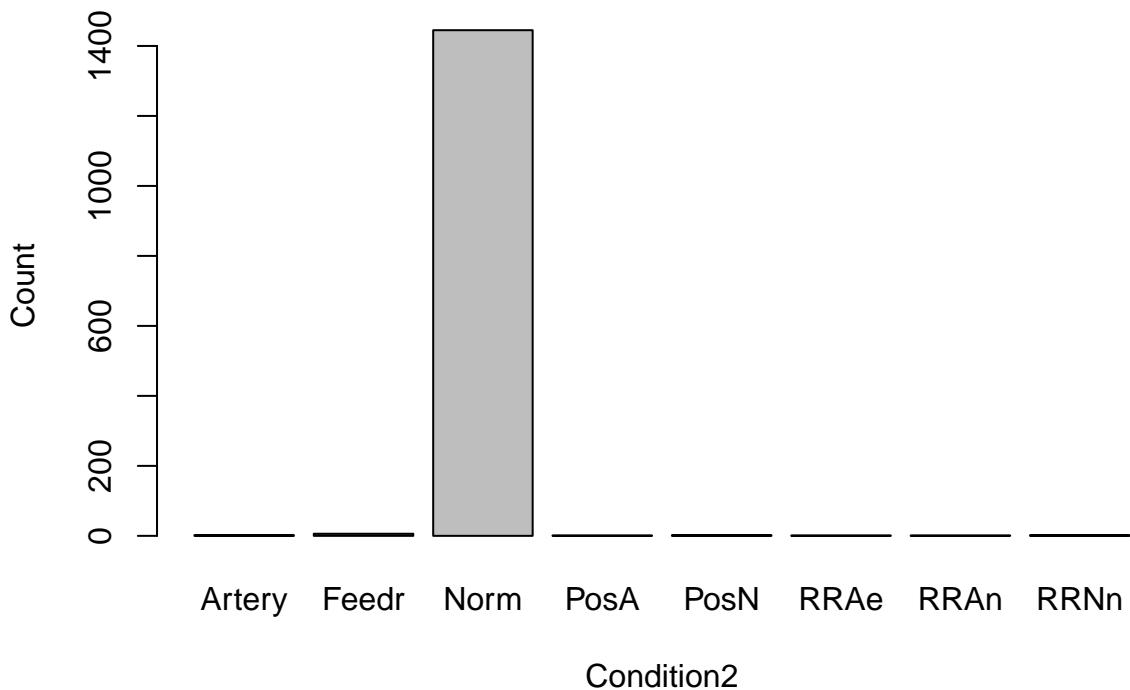
```
table(combined$Condition2)
```

```
##  
## Artery Feedr Norm PosA PosN RRAe RRAn RRNn  
## 2 6 1445 1 2 1 1 2
```

```
combined$Condition2 <- as.factor(combined$Condition2)  
table(combined$Condition2)
```

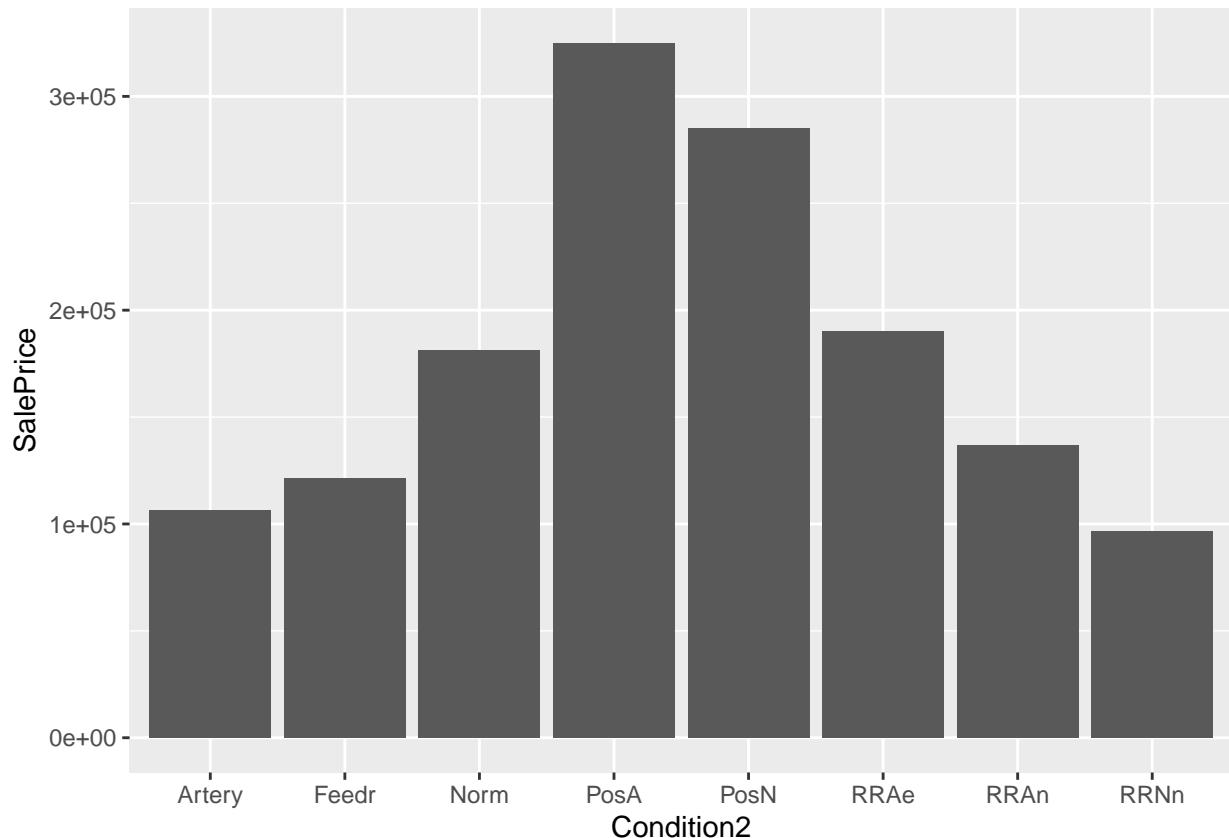
```
##  
## Artery Feedr Norm PosA PosN RRAe RRAn RRNn  
## 2 6 1445 1 2 1 1 2
```

```
barplot(table(combined$Condition2), xlab = "Condition2", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=Condition2, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



BldgType

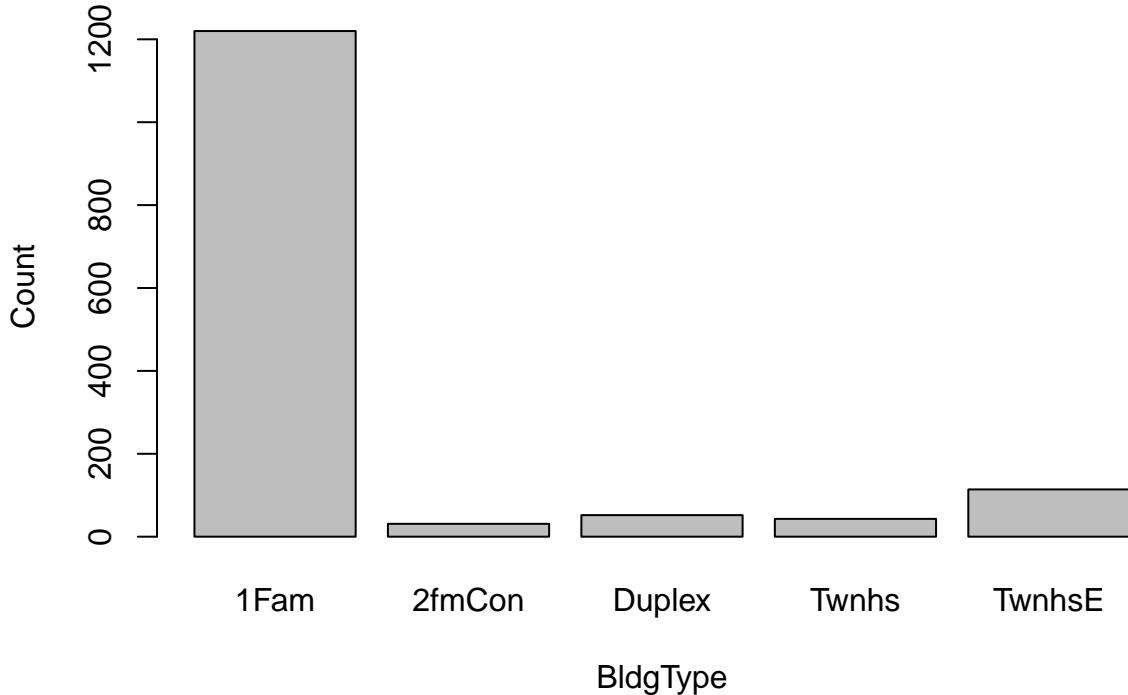
```
table(combined$BldgType)
```

```
##  
## 1Fam 2fmCon Duplex  Twnhs TwnhsE  
## 1220     31      52     43    114
```

```
combined$BldgType <- as.factor(combined$BldgType)  
table(combined$BldgType)
```

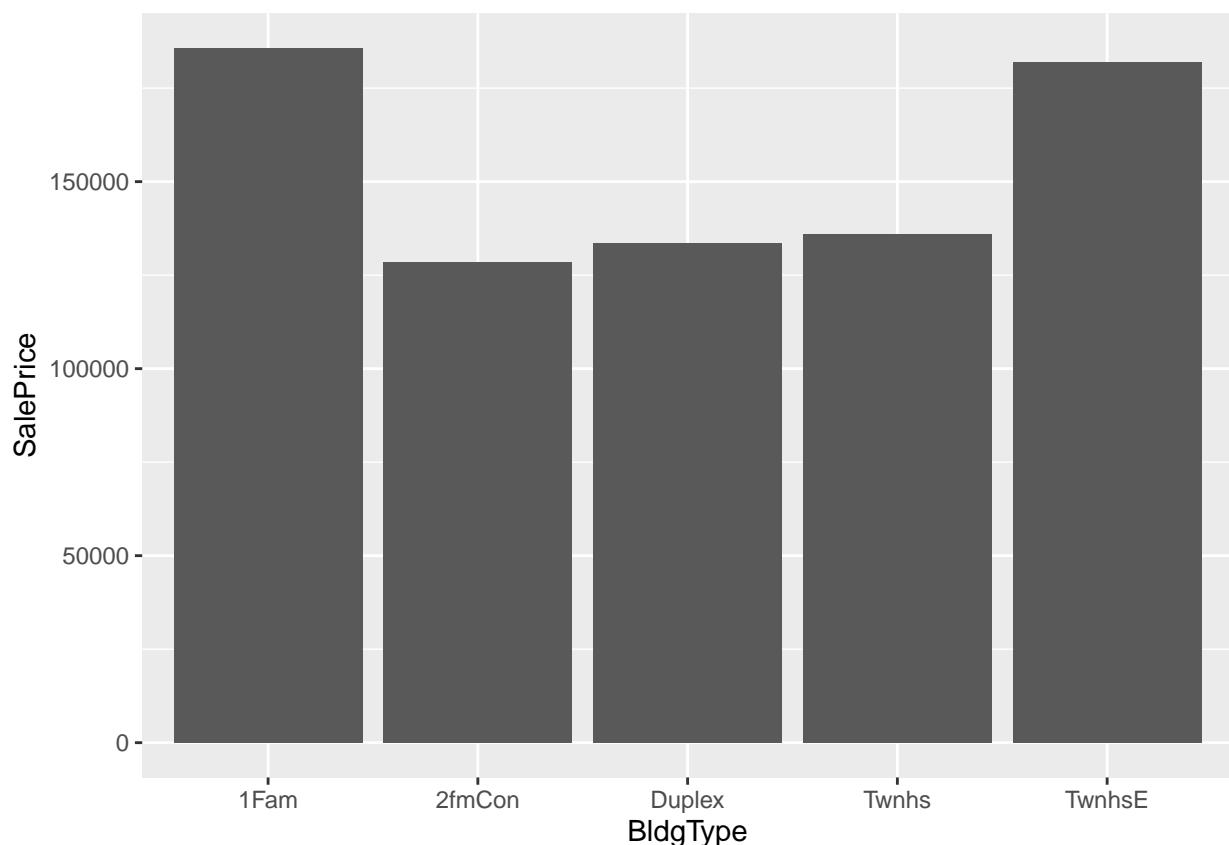
```
##  
## 1Fam 2fmCon Duplex  Twnhs TwnhsE  
## 1220     31      52     43    114
```

```
barplot(table(combined$BldgType), xlab = "BldgType", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=BldgType, y = SalePrice)) + geom_bar(stat = 'summary')

## No summary function supplied, defaulting to 'mean_se()'
```



HouseStyle

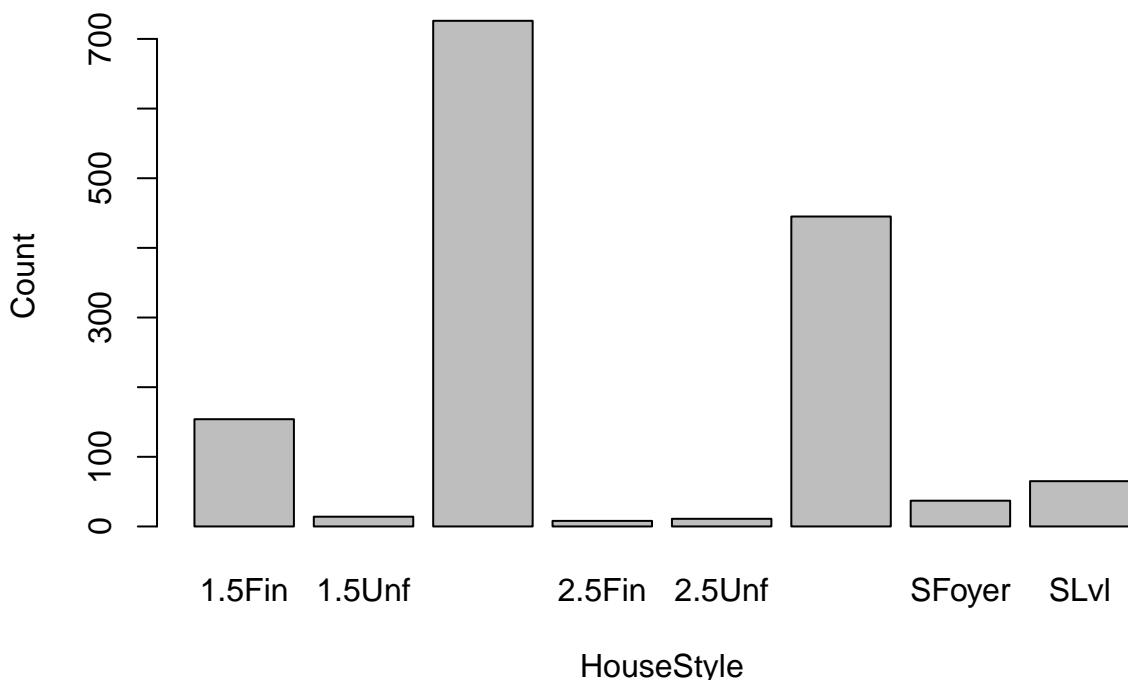
```
table(combined$HouseStyle)
```

```
##  
## 1.5Fin 1.5Unf 1Story 2.5Fin 2.5Unf 2Story SFoyer SLvl  
##    154      14     726      8     11    445      37     65
```

```
combined$HouseStyle <- as.factor(combined$HouseStyle)  
table(combined$HouseStyle)
```

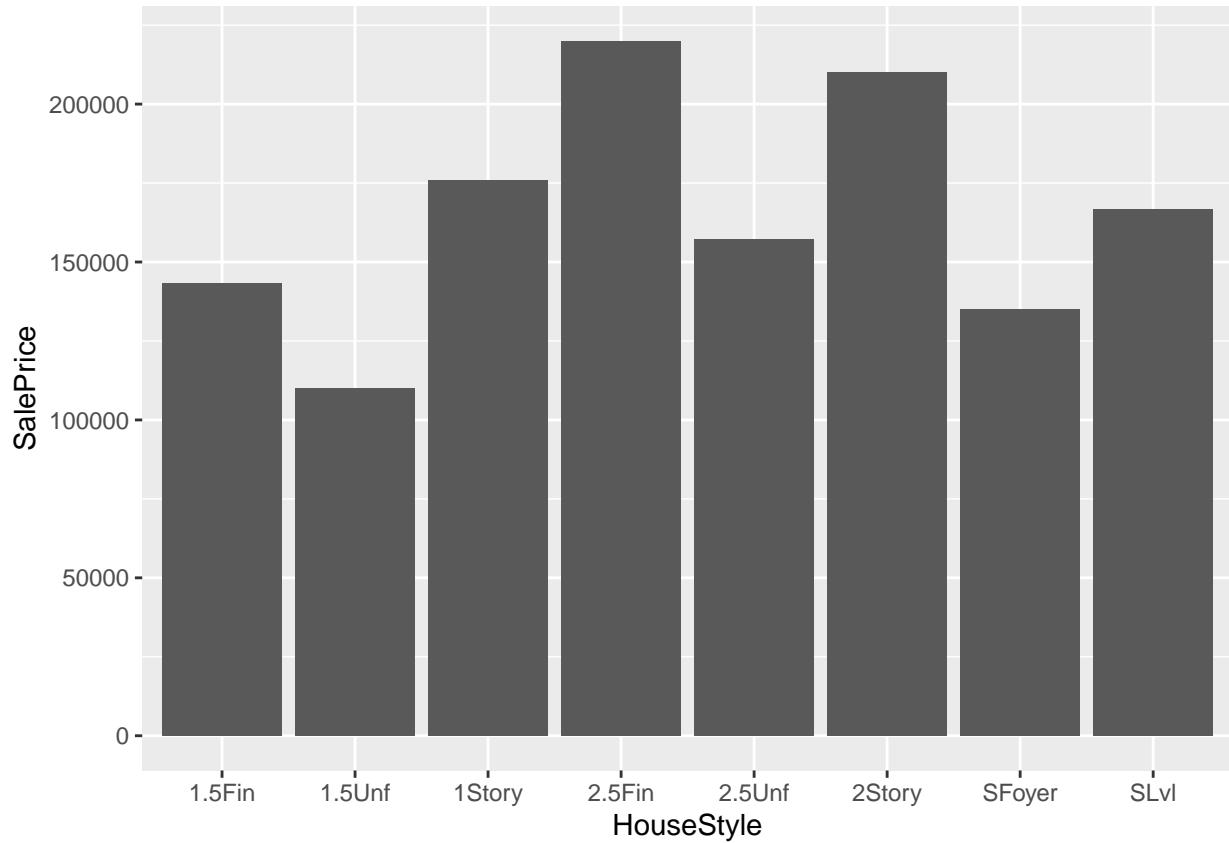
```
##  
## 1.5Fin 1.5Unf 1Story 2.5Fin 2.5Unf 2Story SFoyer SLvl  
##    154      14     726      8     11    445      37     65
```

```
barplot(table(combined$HouseStyle), xlab = "HouseStyle", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=HouseStyle, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



RoofStyle

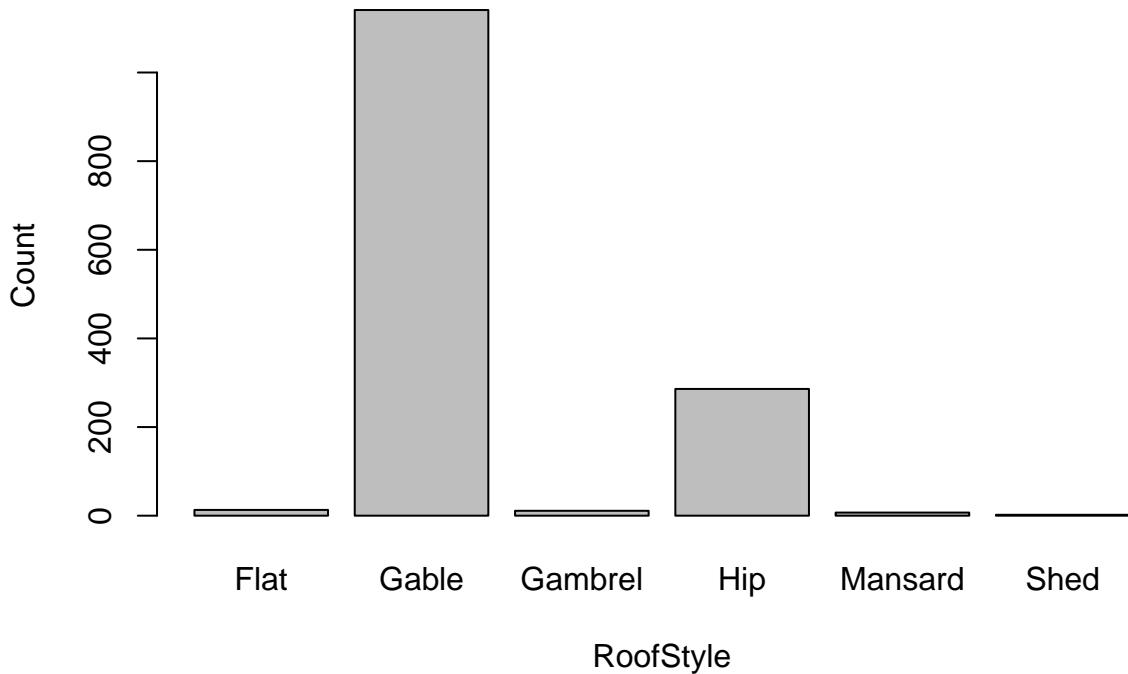
```
table(combined$RoofStyle)
```

```
##  
##      Flat     Gable    Gambrel      Hip    Mansard      Shed  
##      13       1141        11       286         7          2
```

```
combined$RoofStyle <- as.factor(combined$RoofStyle)  
table(combined$RoofStyle)
```

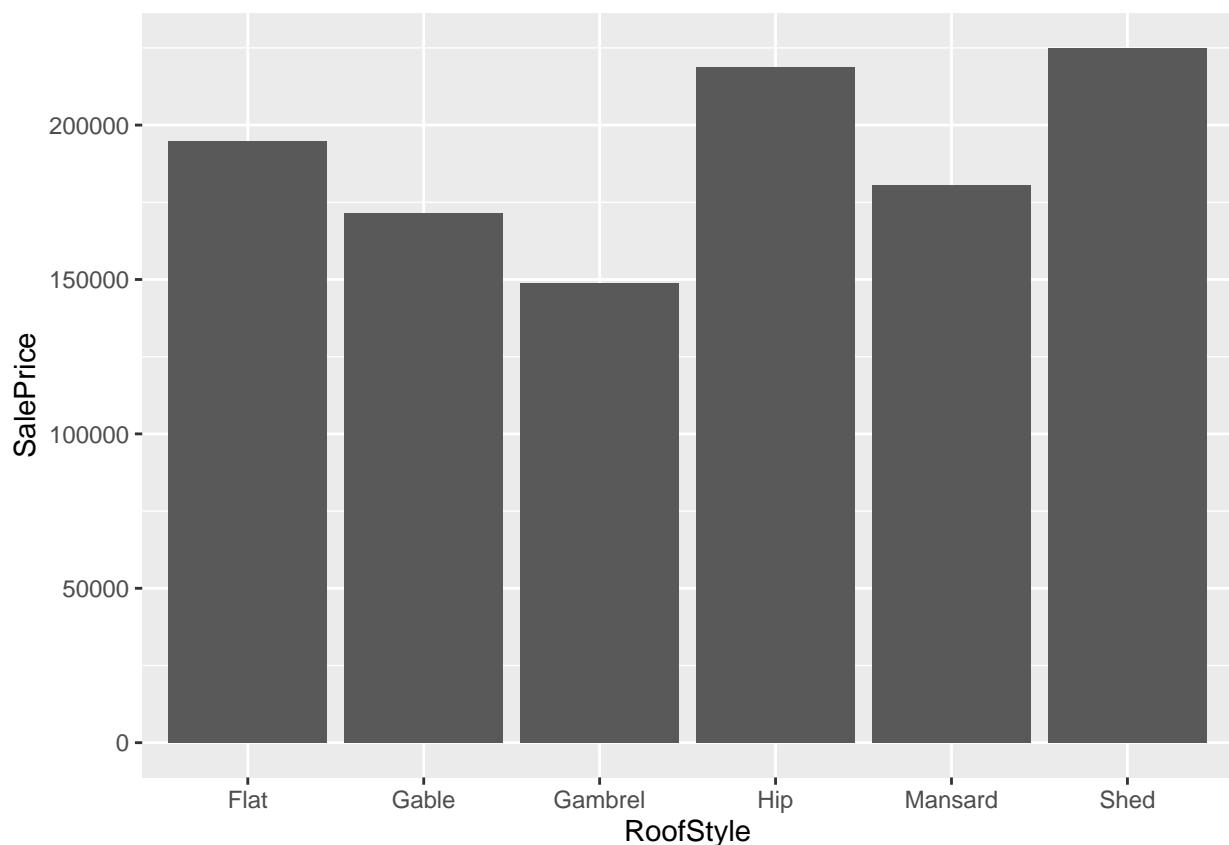
```
##  
##      Flat     Gable    Gambrel      Hip    Mansard      Shed  
##      13       1141        11       286         7          2
```

```
barplot(table(combined$RoofStyle), xlab = "RoofStyle", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=RoofStyle, y = SalePrice)) + geom_bar(stat = 'summary')

## No summary function supplied, defaulting to 'mean_se()'
```



RoofMatl

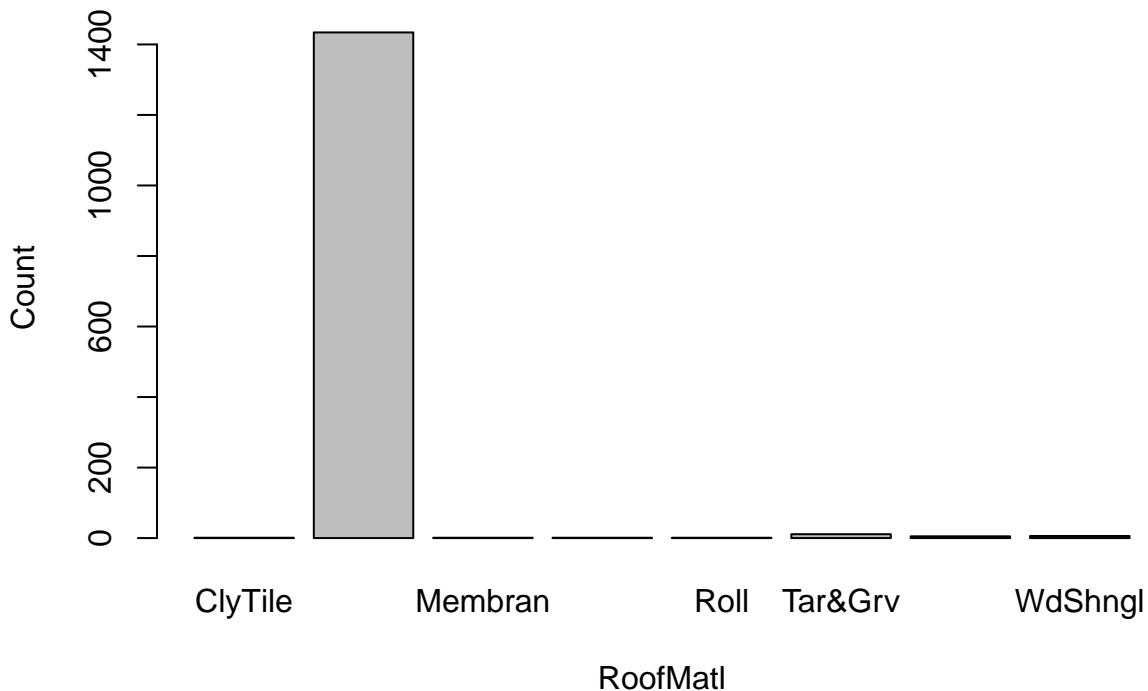
```
table(combined$RoofMatl)
```

```
##  
## ClyTile CompShg Membran Metal Roll Tar&Grv WdShake WdShngl  
##      1     1434      1     1      1     11       5       6
```

```
combined$RoofMatl <- as.factor(combined$RoofMatl)  
table(combined$RoofMatl)
```

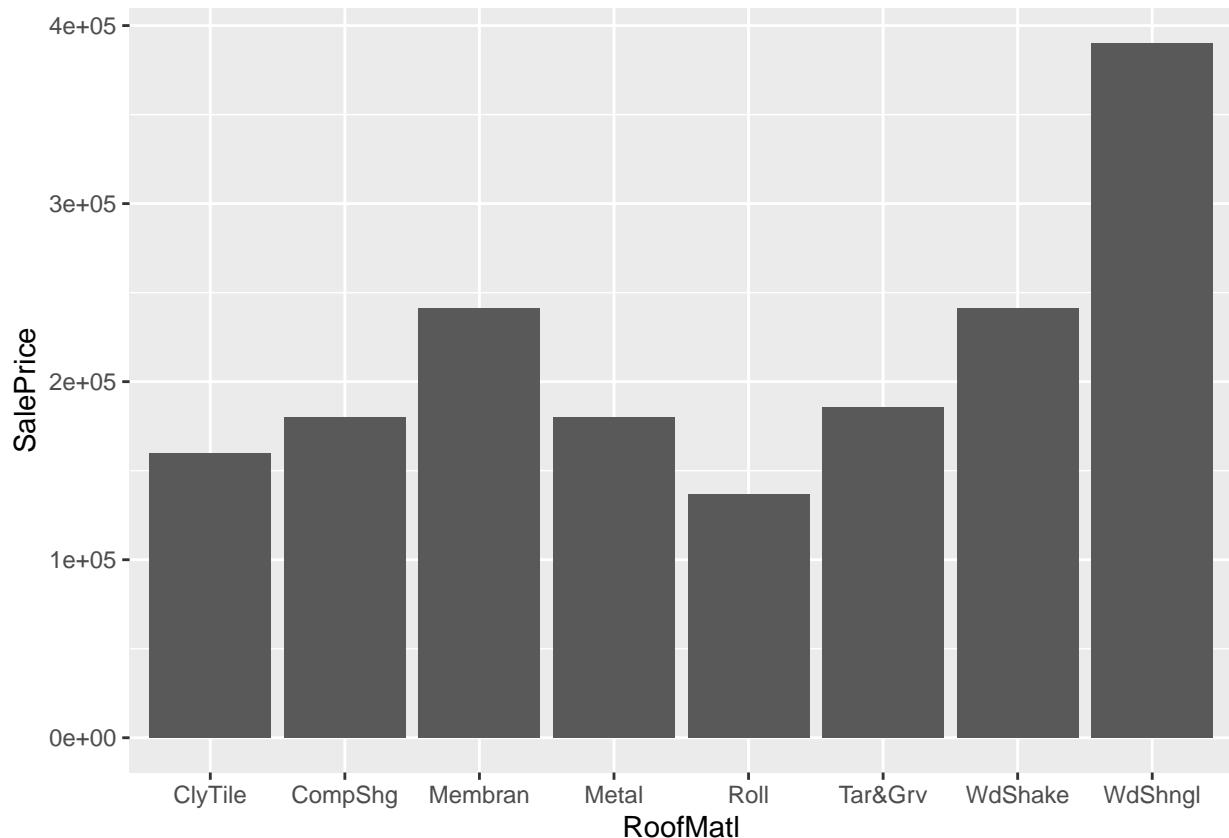
```
##  
## ClyTile CompShg Membran Metal Roll Tar&Grv WdShake WdShngl  
##      1     1434      1     1      1     11       5       6
```

```
barplot(table(combined$RoofMatl), xlab = "RoofMatl", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=RoofMatl, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



Exterior variables

“Exterior1st” “Exterior2nd” “ExterQual” “ExterCond”

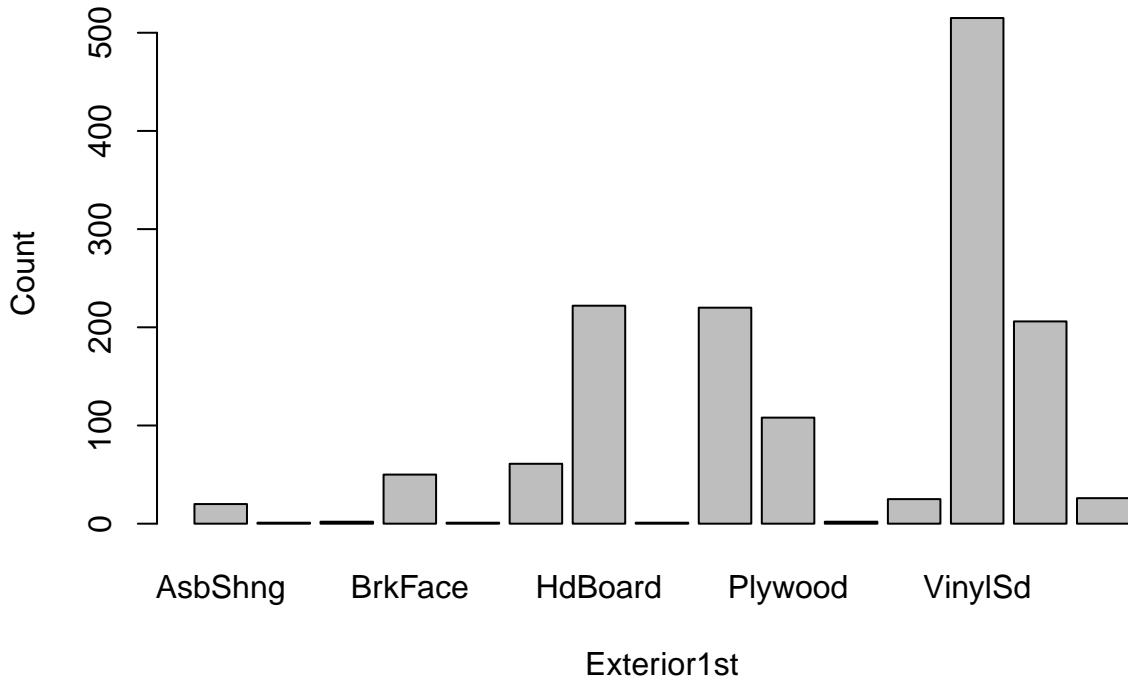
```
table(combined$Exterior1st)
```

```
##
## AsbShng AsphShn BrkComm BrkFace CBlock CemntBd HdBoard ImStucc MetalSd Plywood
##      20      1      2     50      1     61     222      1     220     108
##   Stone  Stucco VinylSd Wd Sdng WdShing
##      2      25     515    206     26
```

```
combined$Exterior1st <- as.factor(combined$Exterior1st)
table(combined$Exterior1st)
```

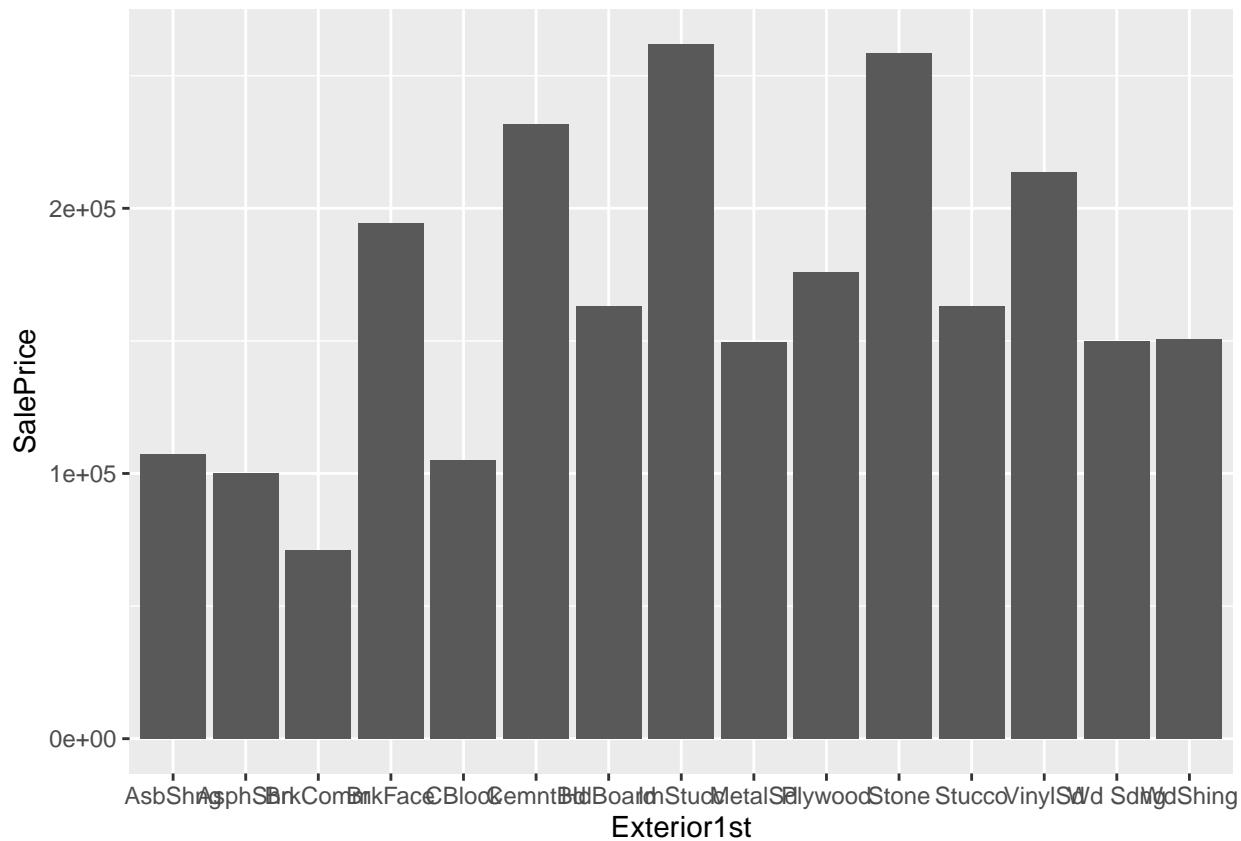
```
##
## AsbShng AsphShn BrkComm BrkFace CBlock CemntBd HdBoard ImStucc MetalSd Plywood
##      20      1      2     50      1     61     222      1     220     108
##   Stone  Stucco VinylSd Wd Sdng WdShing
##      2      25     515    206     26
```

```
barplot(table(combined$Exterior1st), xlab = "Exterior1st", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=Exterior1st, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



```



```

```

combined$Exterior2nd <- as.factor(combined$Exterior2nd)


```

```

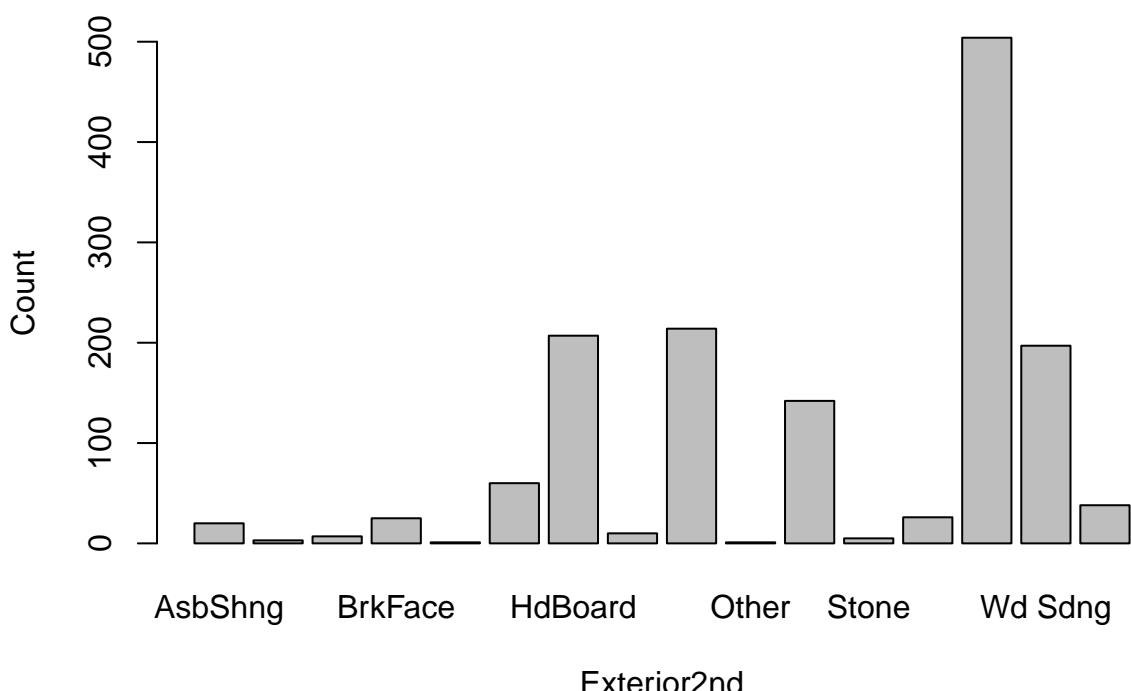
##
## AsbShng AsphShn Brk Cmn BrkFace  CBlock CmentBd HdBoard ImStucc MetalSd  Other
##    20      3     7     25      1     60    207     10     214      1
## Plywood  Stone  Stucco VinylSd Wd Sdng Wd Shng
##    142      5     26    504    197     38

```

```

barplot(table(combined$Exterior2nd), xlab = "Exterior2nd", ylab = "Count")

```

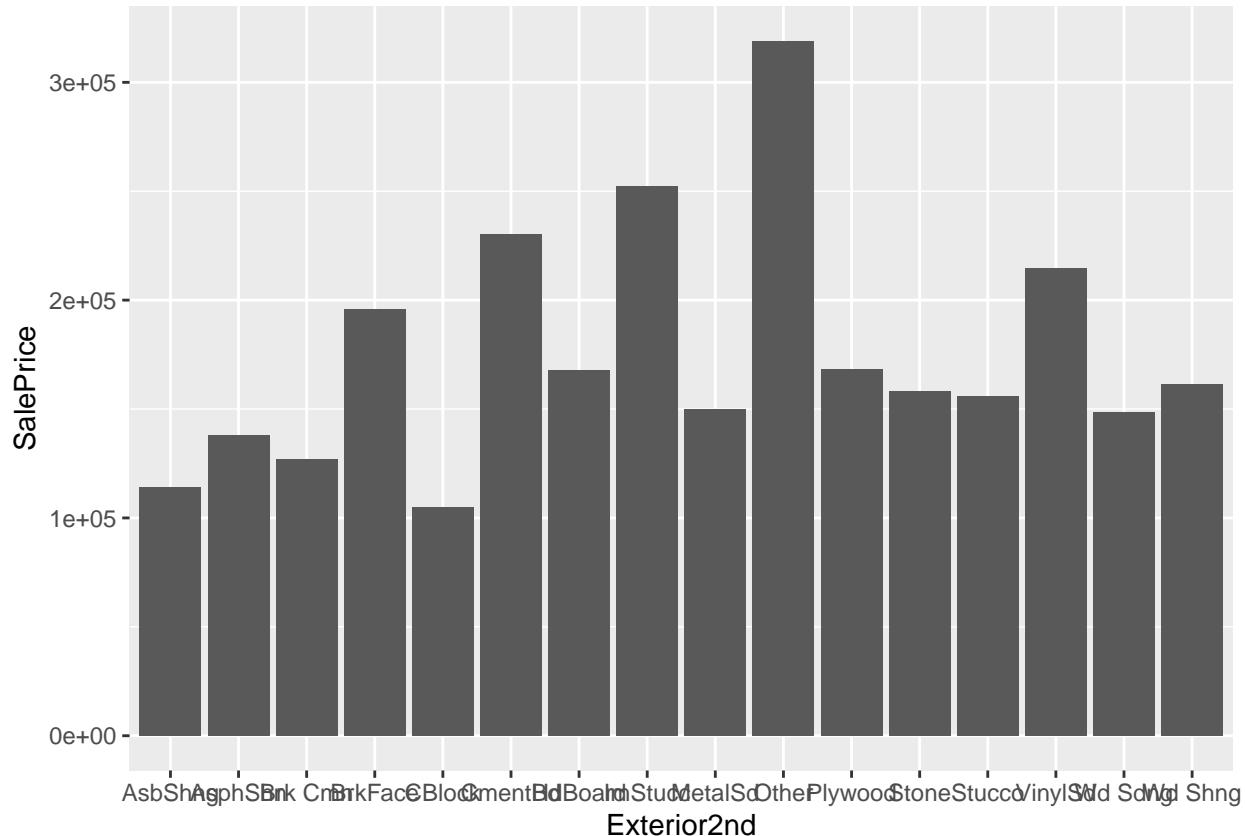


```

ggplot(combined[!is.na(combined$SalePrice),], aes(x=Exterior2nd, y = SalePrice)) + geom_bar(stat = 'summary')

## No summary function supplied, defaulting to 'mean_se()'

```



```
table(combined$ExterQual)
```

```
##
##   Ex   Fa   Gd   TA
##   52   14  488  906
```

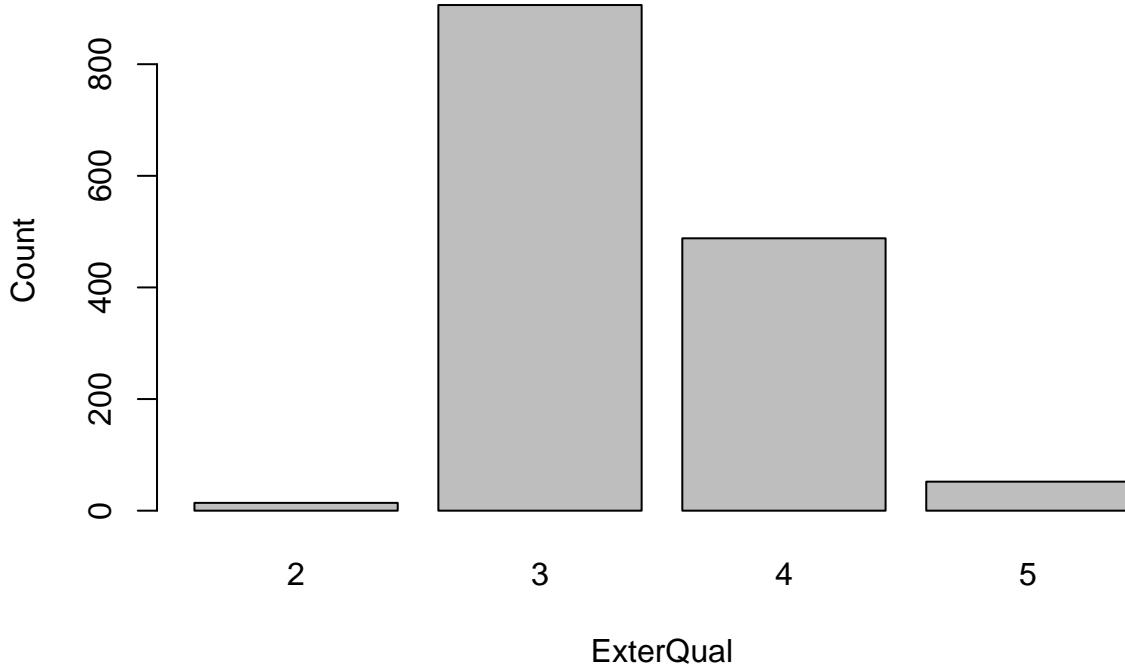
```
combined$ExterQual<-as.integer(revalue(combined$ExterQual, quality))
```

```
## The following 'from' values were not present in 'x': None, Po
```

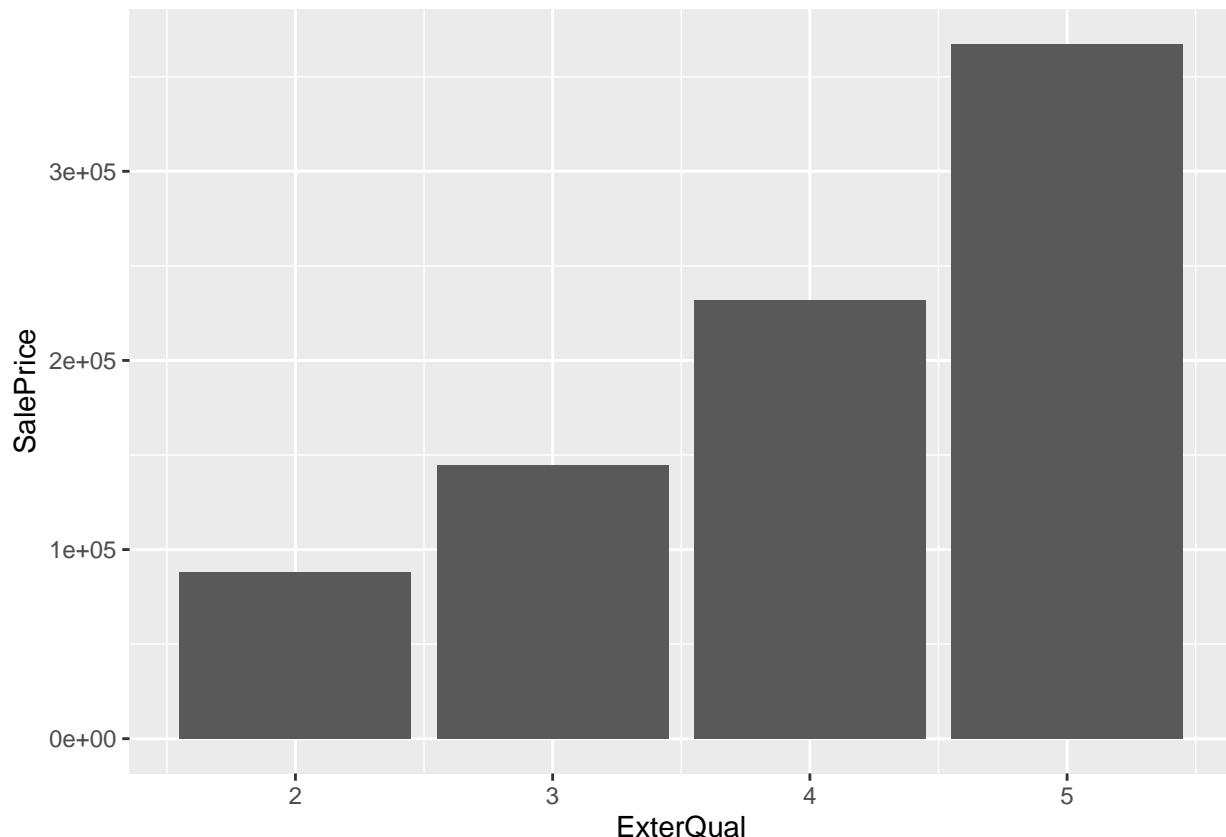
```
table(combined$ExterQual)
```

```
##
##   2    3    4    5
##   14  906  488  52
```

```
barplot(table(combined$ExterQual), xlab = "ExterQual", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=ExterQual, y = SalePrice)) + geom_bar(stat = 'summary')  
## No summary function supplied, defaulting to 'mean_se()'
```



```
table(combined$ExterCond)
```

```
##  
##   Ex   Fa   Gd   Po   TA  
##   3    28  146     1 1282
```

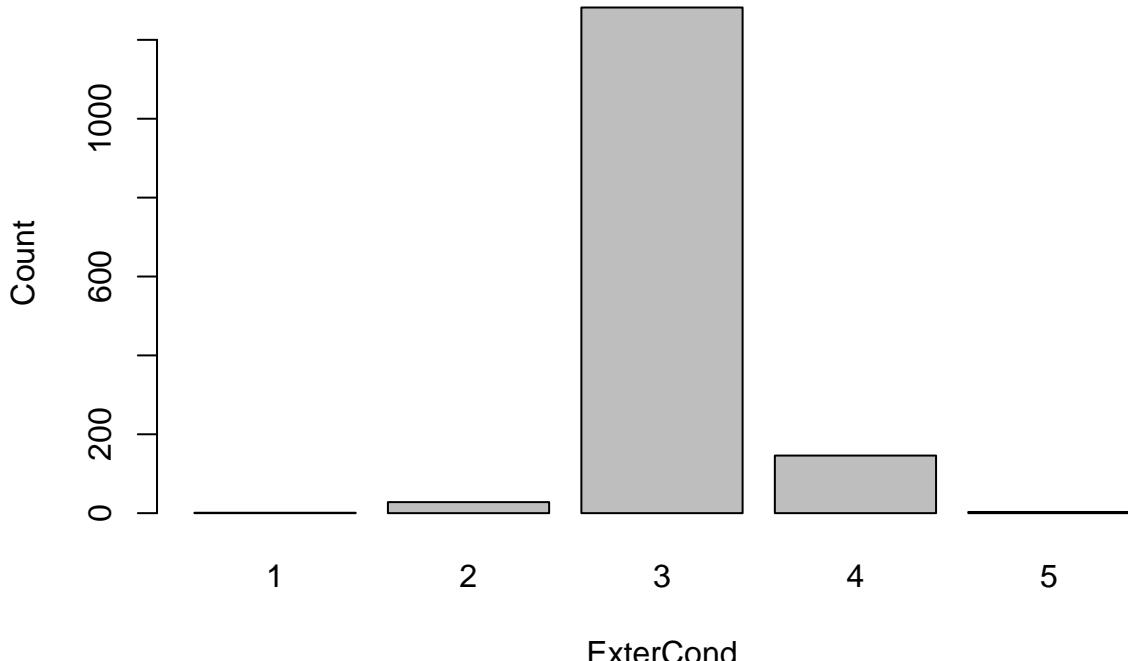
```
combined$ExterCond<-as.integer(revalue(combined$ExterCond, quality))
```

```
## The following 'from' values were not present in 'x': None
```

```
table(combined$ExterCond)
```

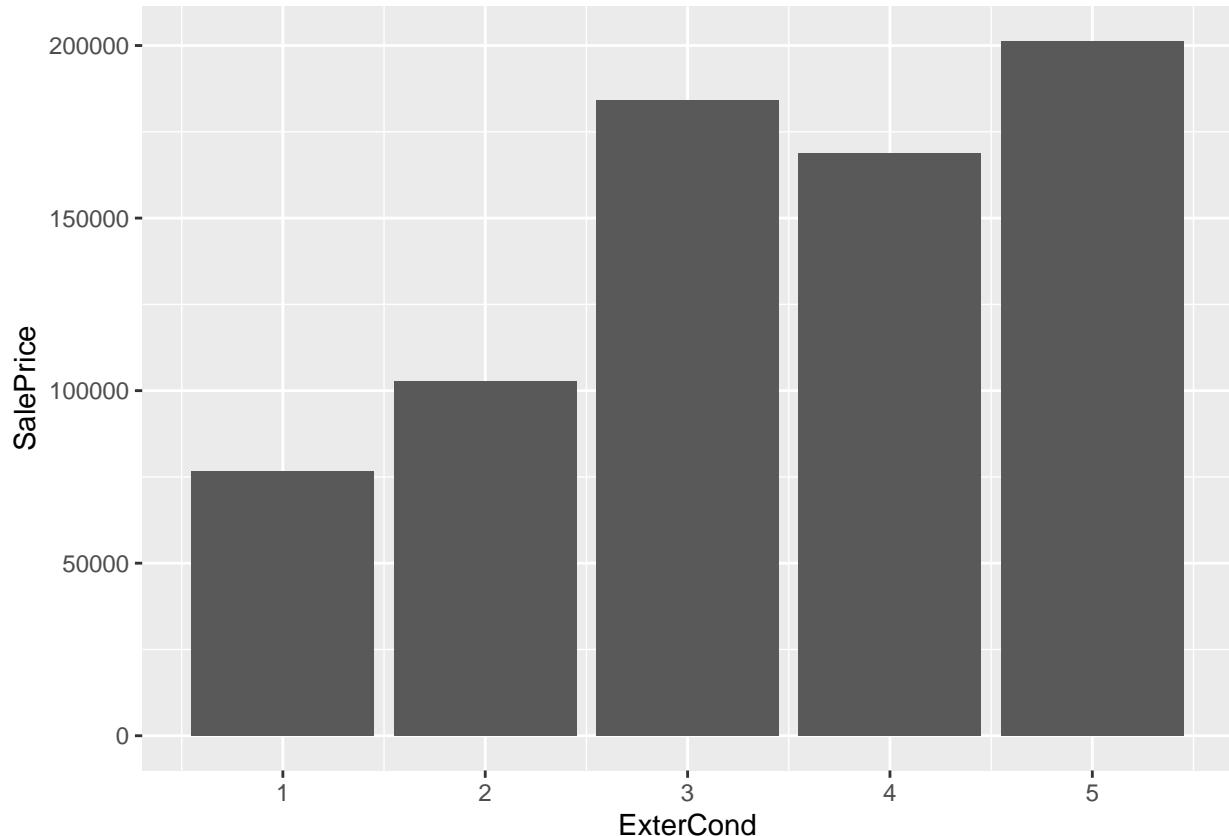
```
##  
##   1    2    3    4    5  
##   1    28  1282  146     3
```

```
barplot(table(combined$ExterCond), xlab = "ExterCond", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=ExterCond, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```

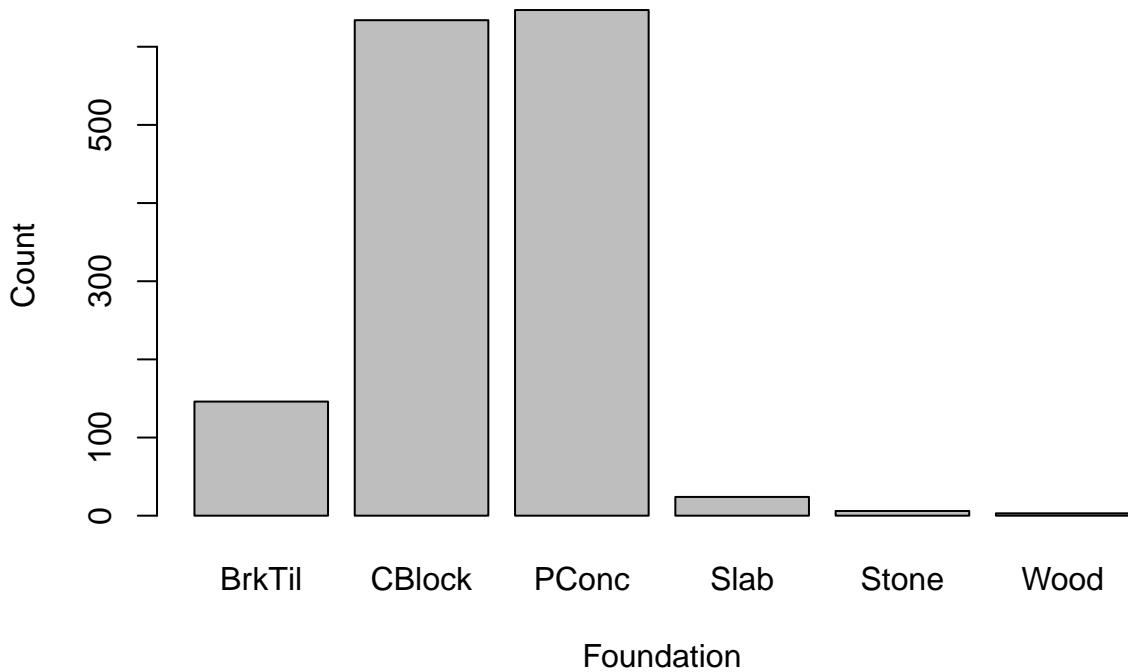


Foundation

```

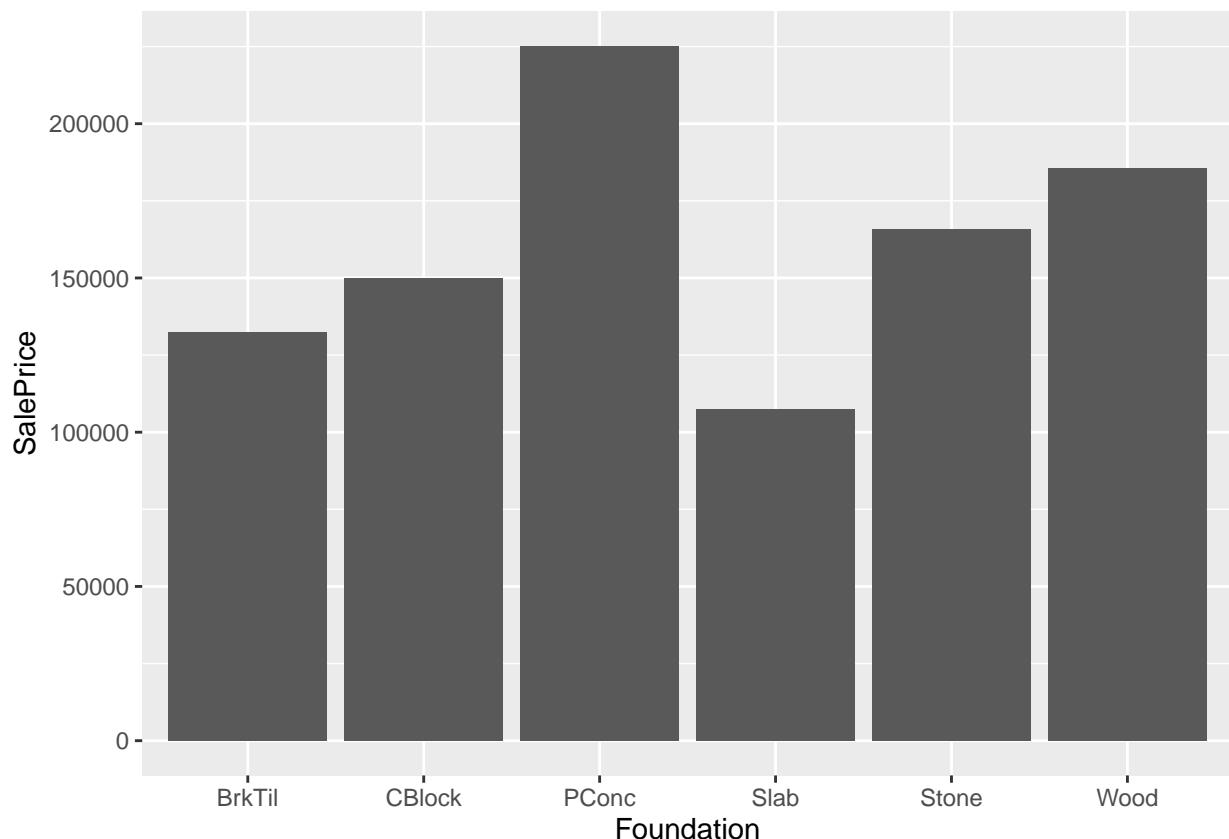


```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=Foundation, y = SalePrice)) + geom_bar(stat = 'summary')

## No summary function supplied, defaulting to 'mean_se()'
```



heating variables

“Heating” “HeatingQC”

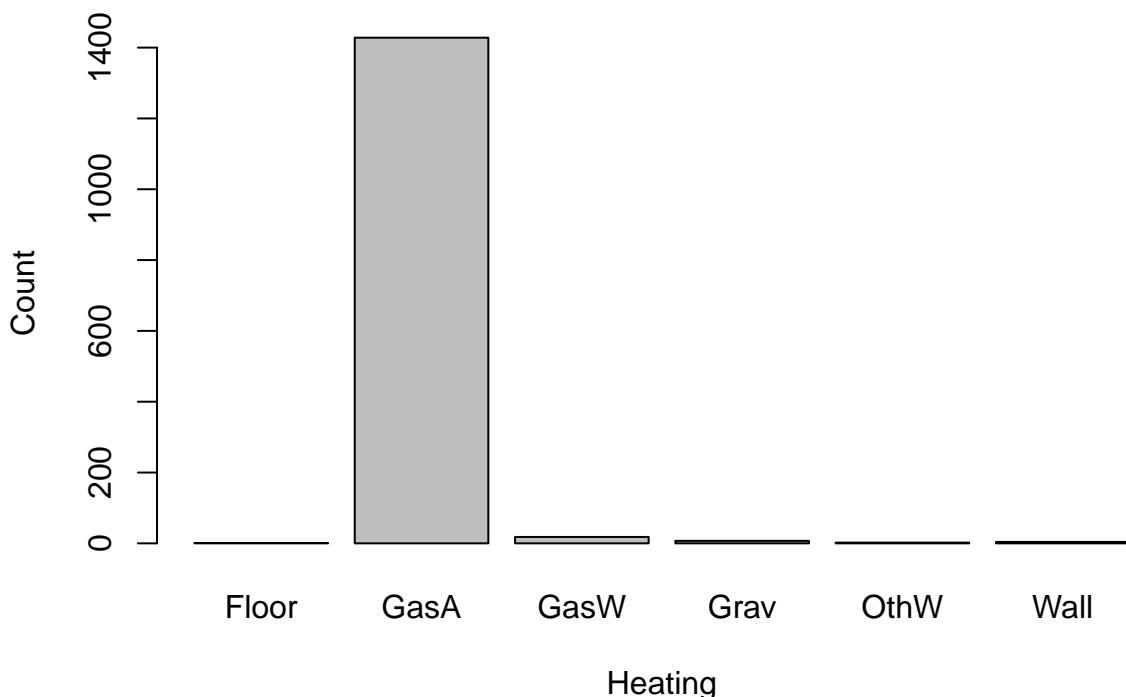
```
table(combined$Heating)
```

```
##  
## Floor GasA GasW Grav OthW Wall  
##    1 1428     18     7     2     4
```

```
combined$Heating <- as.factor(combined$Heating)  
table(combined$Heating)
```

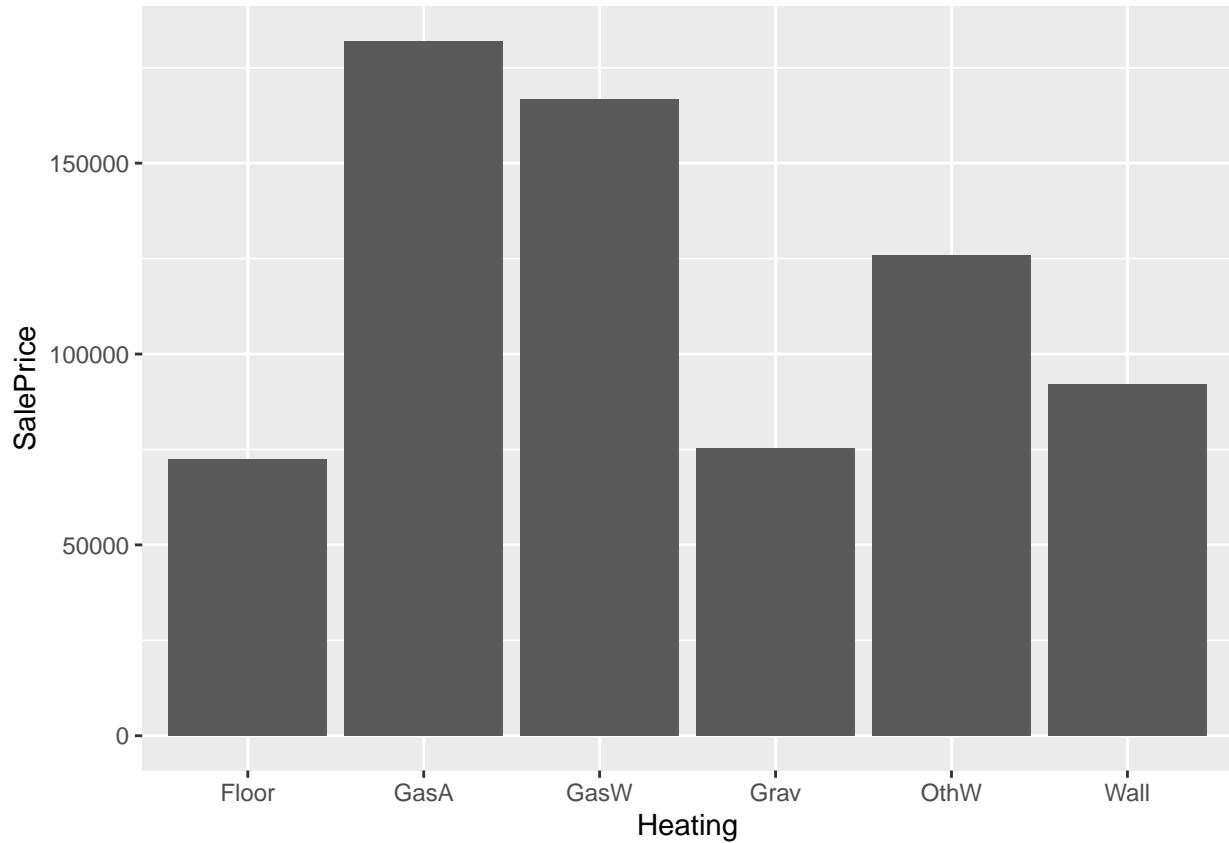
```
##  
## Floor GasA GasW Grav OthW Wall  
##    1 1428     18     7     2     4
```

```
barplot(table(combined$Heating), xlab = "Heating", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=Heating, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



```
table(combined$HeatingQC)
```

```
##
##   Ex   Fa   Gd   Po   TA
## 741   49  241     1  428
```

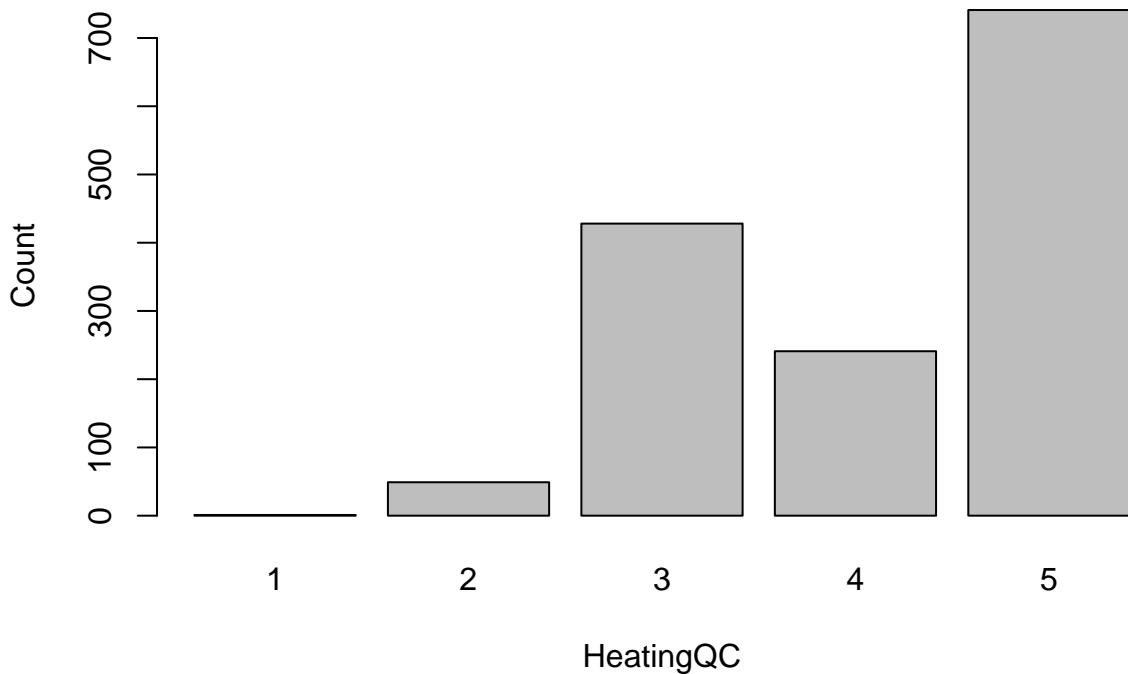
```
combined$HeatingQC<-as.integer(revalue(combined$HeatingQC, quality))
```

```
## The following 'from' values were not present in 'x': None
```

```
table(combined$HeatingQC)
```

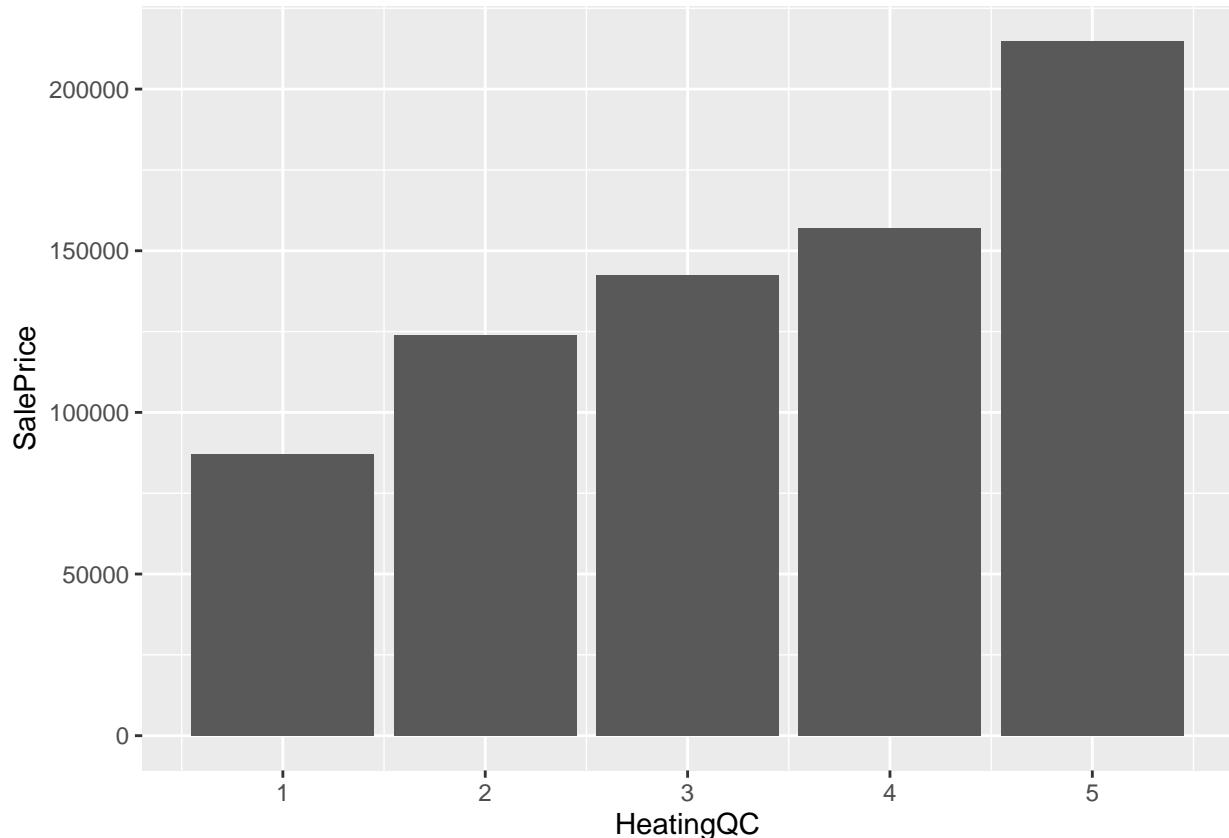
```
##
##   1    2    3    4    5
##   1   49  428  241  741
```

```
barplot(table(combined$HeatingQC), xlab = "HeatingQC", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=HeatingQC, y = SalePrice)) + geom_bar(stat = 'summary')

## No summary function supplied, defaulting to 'mean_se()'
```



CentralAir

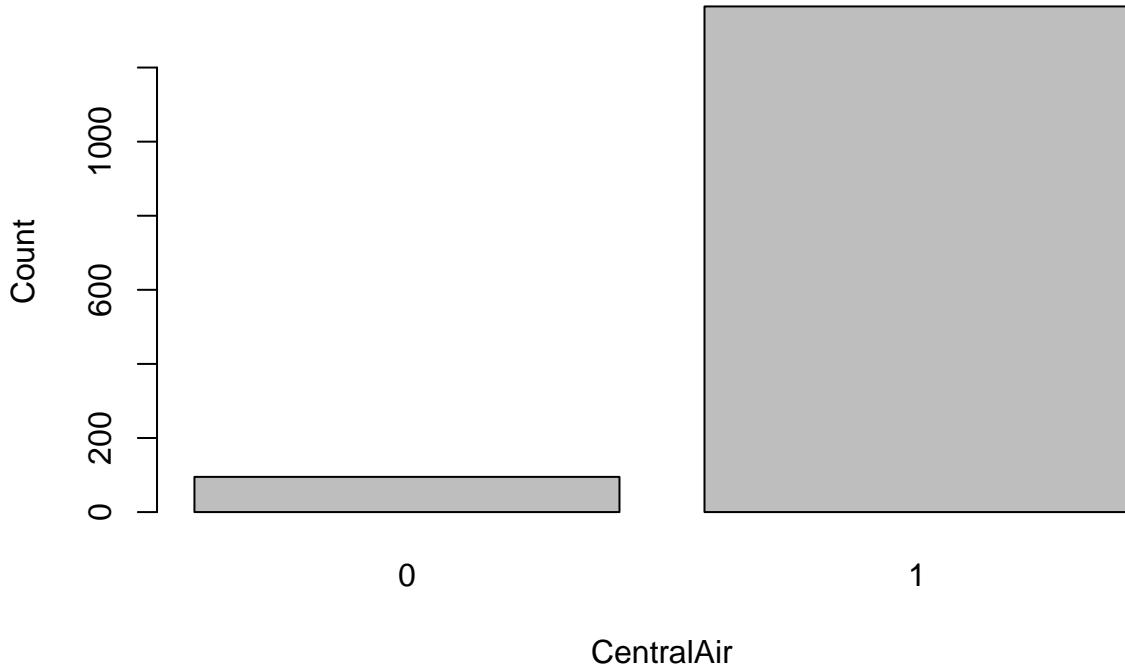
```
table(combined$CentralAir)

##
##      N      Y
##    95 1365

combined$CentralAir<-as.integer(revalue(combined$CentralAir, c('N'=0, 'Y'=1)))
table(combined$CentralAir)

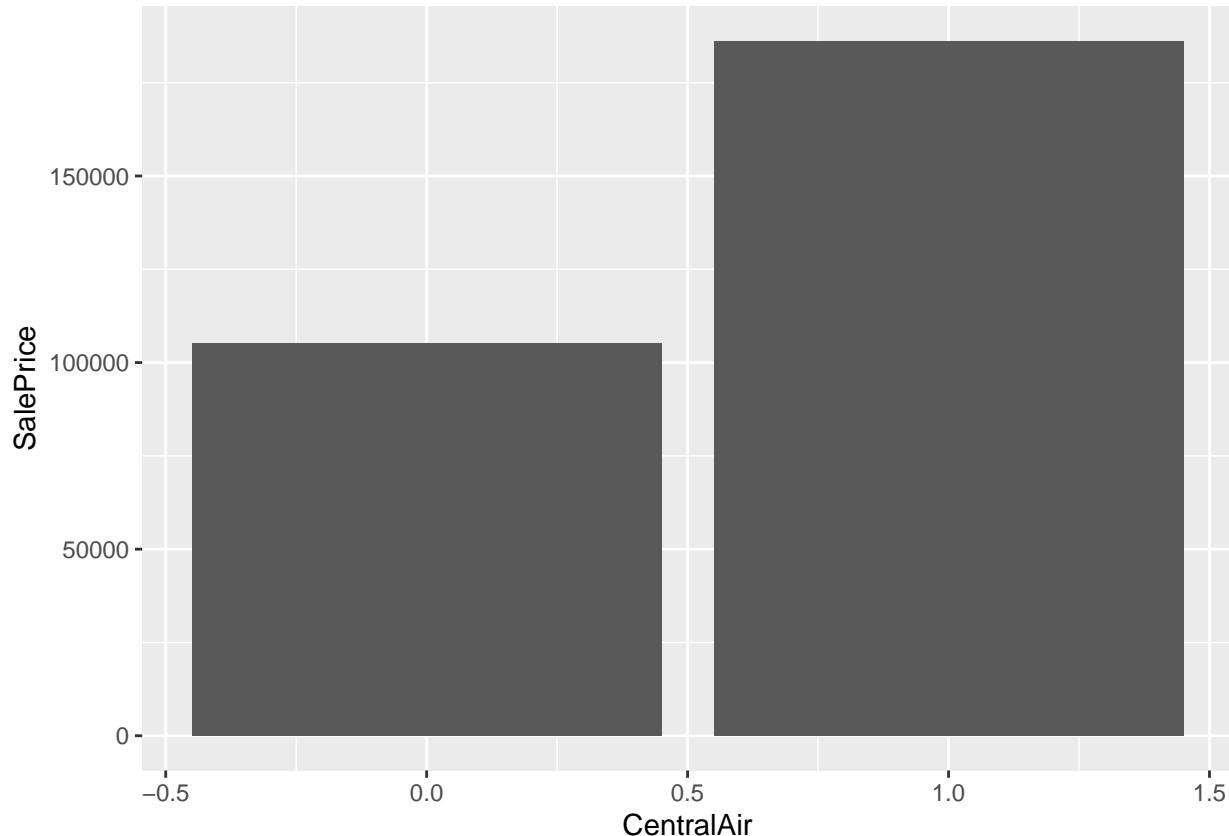
##
##      0      1
##    95 1365

barplot(table(combined$CentralAir), xlab = "CentralAir", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=CentralAir, y = SalePrice)) + geom_bar(stat = 'summary')

## No summary function supplied, defaulting to 'mean_se()'
```

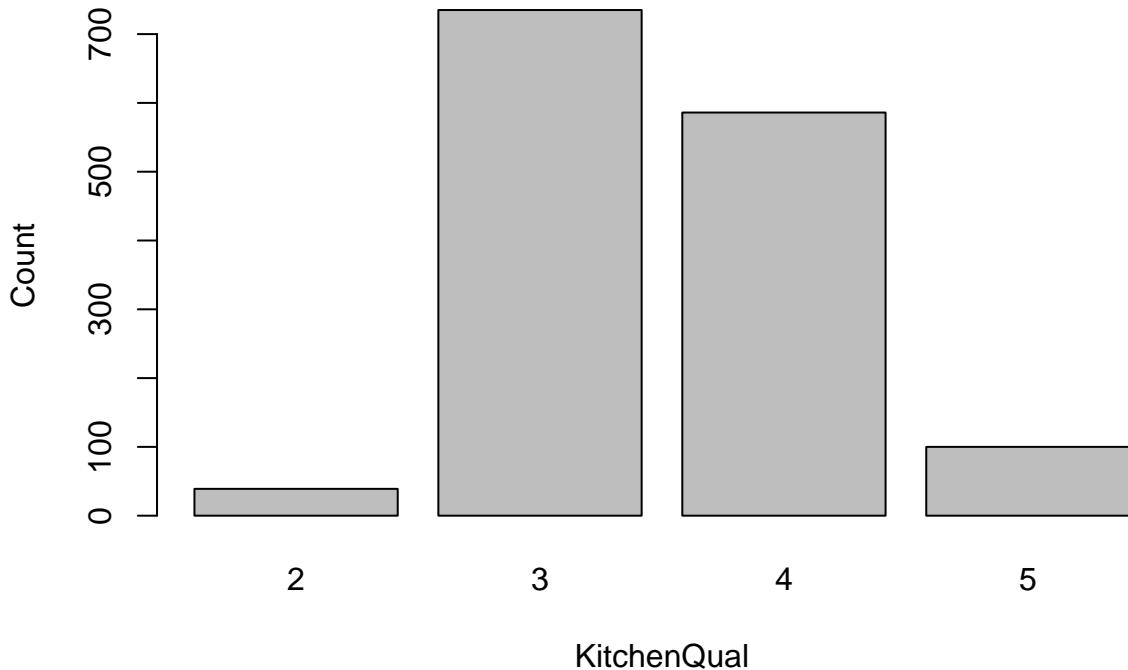


KitchenQual

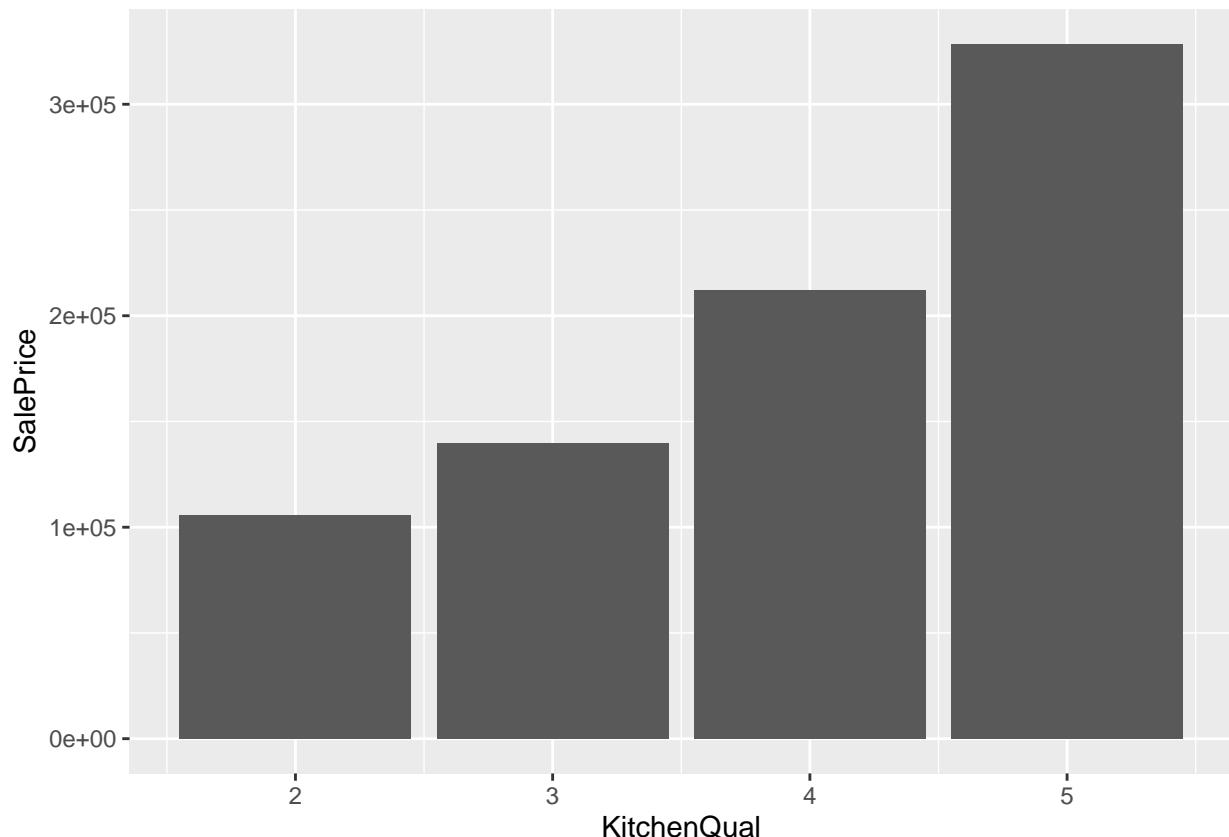
```



```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=KitchenQual, y = SalePrice)) + geom_bar(stat = 'summary')  
## No summary function supplied, defaulting to 'mean_se()'
```



Functional

```
table(combined$Functional)

##
## Maj1 Maj2 Min1 Min2  Mod  Sev  Typ
##   14     5    31    34    15      1 1360

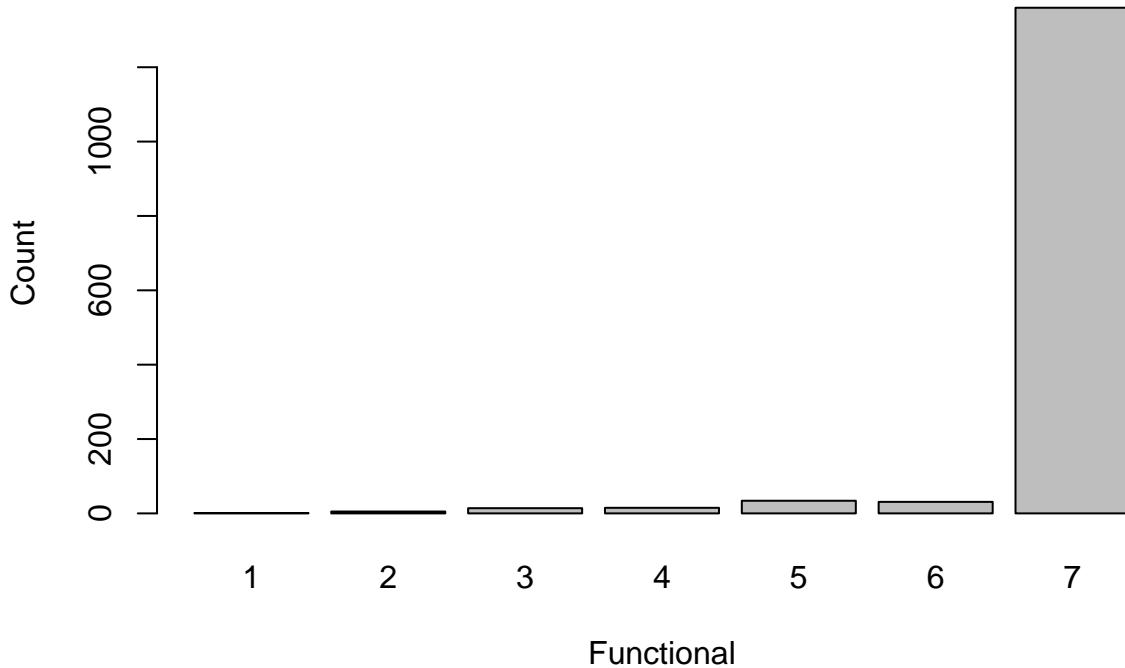
combined$Functional<-as.integer(revalue(combined$Functional, c('Sal'=0, 'Sev'=1, 'Maj2'=2, 'Maj1'=3, 'Mod'=4, 'M

## The following 'from' values were not present in 'x': Sal

table(combined$Functional)

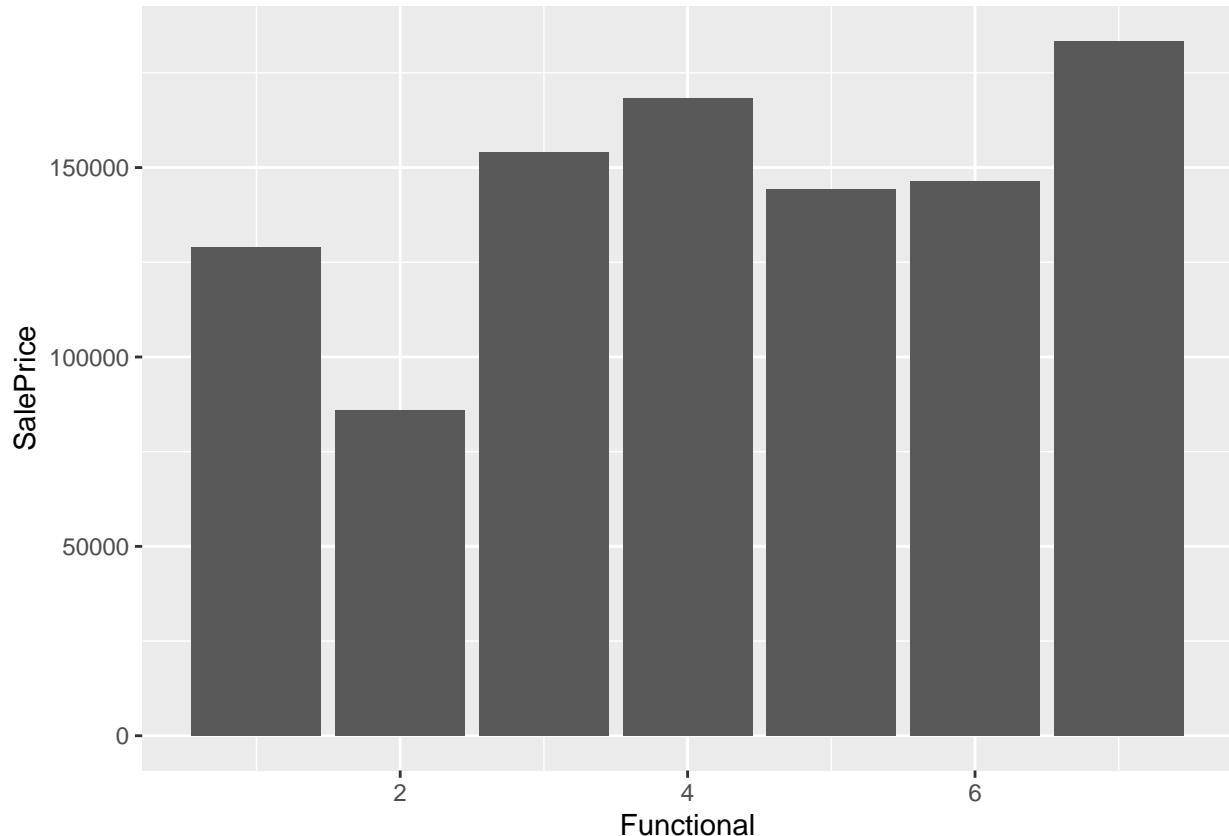
##
##    1     2     3     4     5     6     7
##    1     5    14    15    34    31  1360

barplot(table(combined$Functional), xlab = "Functional", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=Functional, y = SalePrice)) + geom_bar(stat = 'summary')

## No summary function supplied, defaulting to 'mean_se()'
```



PavedDrive

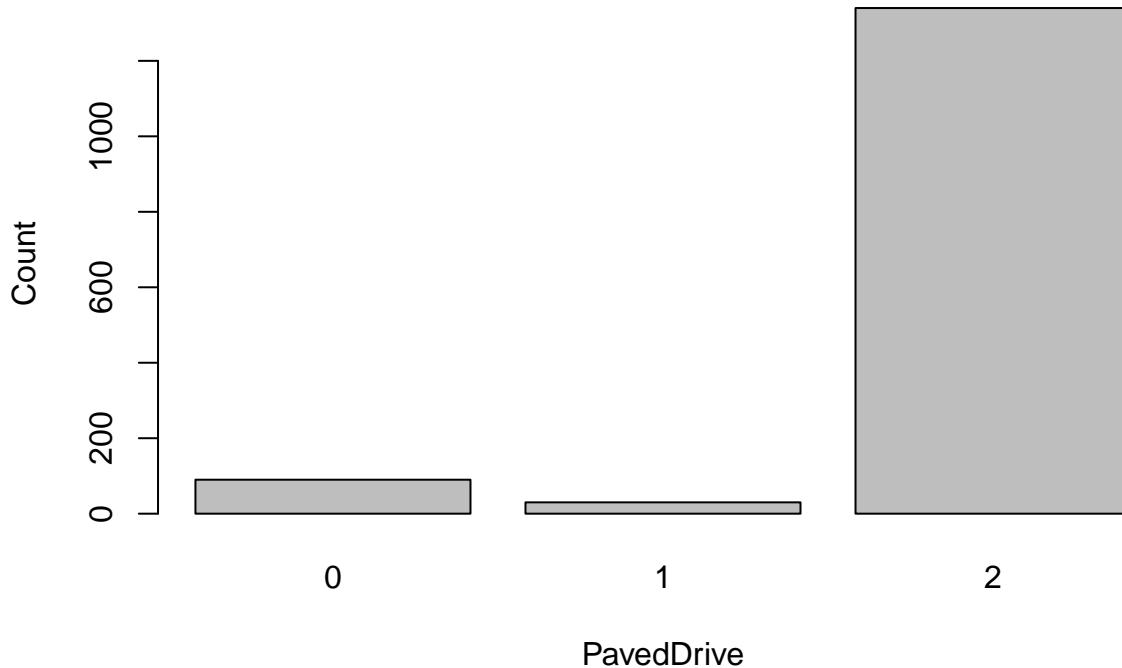
```
table(combined$PavedDrive)
```

```
##
##      N      P      Y
##    90    30  1340
```

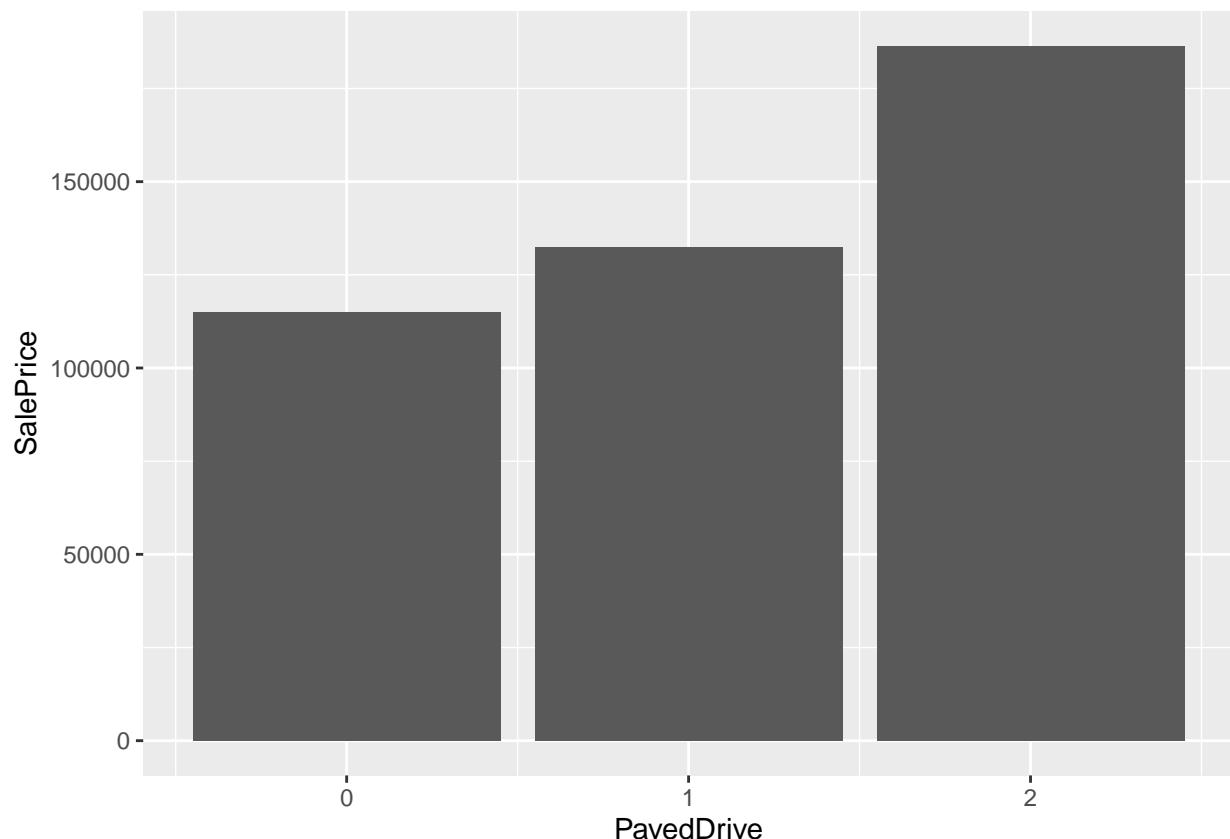
```
combined$PavedDrive<-as.integer(revalue(combined$PavedDrive, c('N'=0, 'P'=1, 'Y'=2)))
table(combined$PavedDrive)
```

```
##
##      0      1      2
##    90    30  1340
```

```
barplot(table(combined$PavedDrive), xlab = "PavedDrive", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=PavedDrive, y = SalePrice)) + geom_bar(stat = 'summary')  
## No summary function supplied, defaulting to 'mean_se()'
```



Sales variables

“SaleType” “SaleCondition”

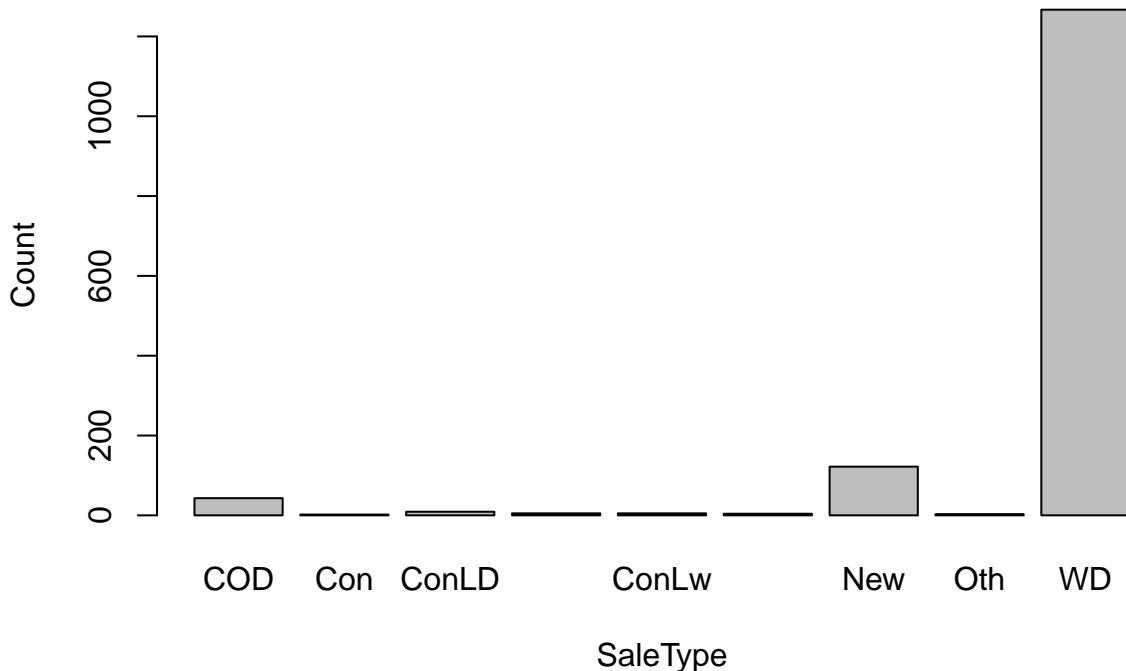
```
table(combined$SaleType)
```

```
##  
##   COD    Con  ConLD ConLI ConLw    CWD    New    Oth    WD  
##   43     2     9     5     5     4    122     3   1267
```

```
combined$SaleType <- as.factor(combined$SaleType)  
table(combined$SaleType)
```

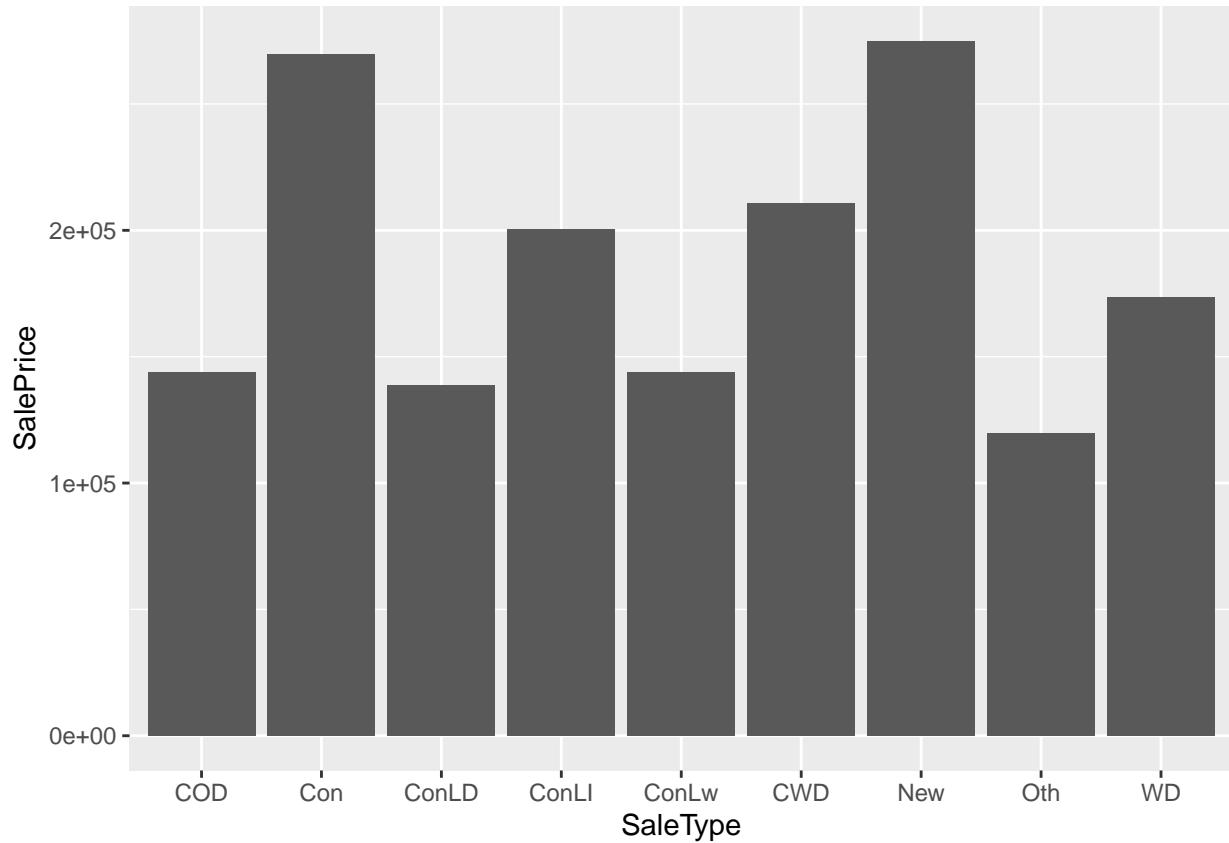
```
##  
##   COD    Con  ConLD ConLI ConLw    CWD    New    Oth    WD  
##   43     2     9     5     5     4    122     3   1267
```

```
barplot(table(combined$SaleType), xlab = "SaleType", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=SaleType, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



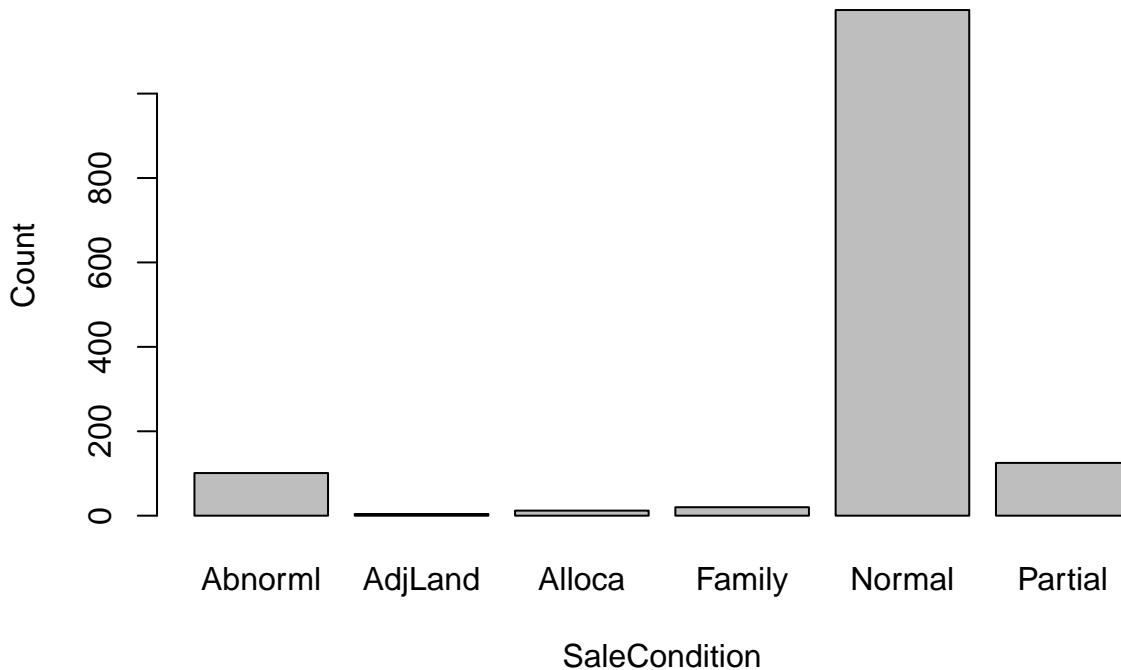
```
table(combined$SaleCondition)
```

```
##
## Abnorml AdjLand Allocat Family Normal Partial
##      101        4       12      20     1198      125
```

```
combined$SaleCondition <- as.factor(combined$SaleCondition)
table(combined$SaleCondition)
```

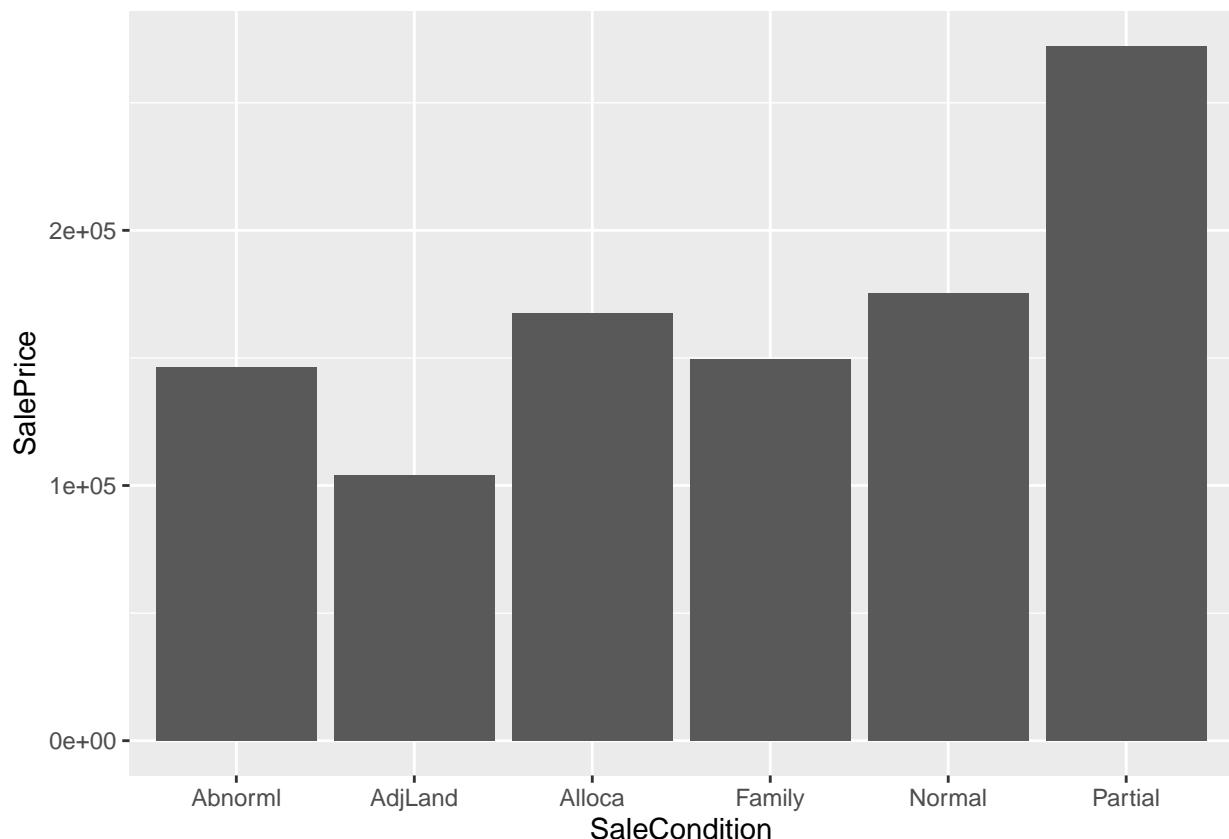
```
##
## Abnorml AdjLand Allocat Family Normal Partial
##      101        4       12      20     1198      125
```

```
barplot(table(combined$SaleCondition), xlab = "SaleCondition", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=SaleCondition, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



Remaining character variables

```
Charcol <- names(combined[, sapply(combined, is.character)])  
Charcol
```

```
## character(0)
```

MSSubClass

```
20 1-STORY 1946 & NEWER ALL STYLES  
30 1-STORY 1945 & OLDER  
40 1-STORY W/FINISHED ATTIC ALL AGES  
45 1-1/2 STORY - UNFINISHED ALL AGES  
50 1-1/2 STORY FINISHED ALL AGES  
60 2-STORY 1946 & NEWER  
70 2-STORY 1945 & OLDER  
75 2-1/2 STORY ALL AGES  
80 SPLIT OR MULTI-LEVEL  
85 SPLIT FOYER  
90 DUPLEX - ALL STYLES AND AGES
```

120 1-STORY PUD (Planned Unit Development) - 1946 & NEWER 150 1-1/2 STORY PUD - ALL AGES 160 2-STORY PUD - 1946 & NEWER 180 PUD - MULTILEVEL - INCL SPLIT LEV/FOYER 190 2 FAMILY CONVERSION - ALL STYLES AND AGES

```
table(combined$MSSubClass)
```

```
##  
## 20 30 40 45 50 60 70 75 80 85 90 120 160 180 190  
## 536 69 4 12 144 299 60 16 58 20 52 87 63 10 30
```

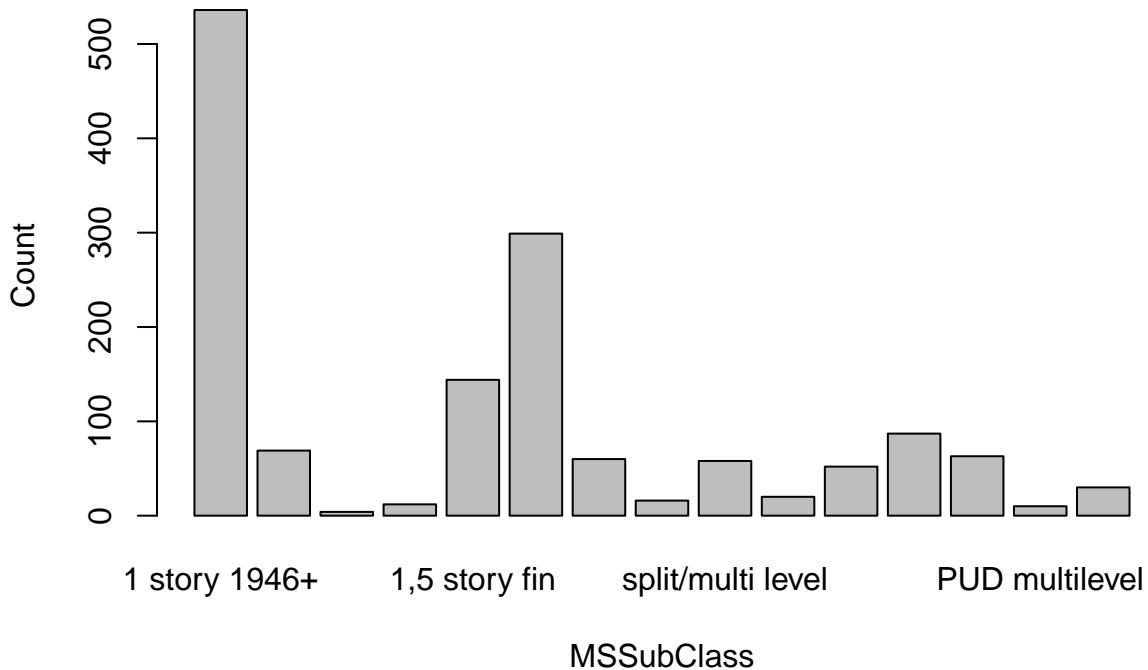
```
combined$MSSubClass <- as.factor(combined$MSSubClass)
```

```
#revalue for better readability
```

```
combined$MSSubClass<-revalue(combined$MSSubClass, c('20'='1 story 1946+', '30'='1 story 1945-', '40'='1 story un'))
```

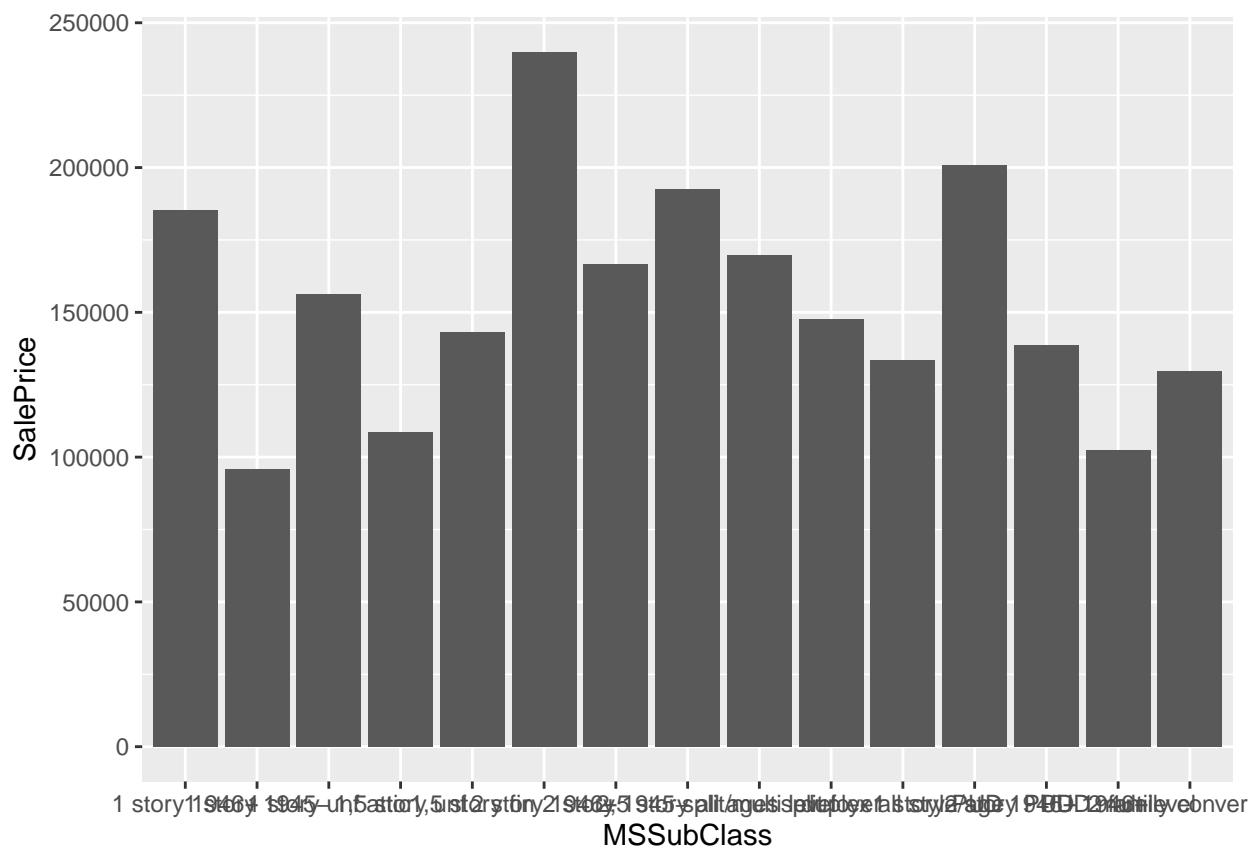
```
## The following 'from' values were not present in 'x': 150
```

```
barplot(table(combined$MSSubClass), xlab = "MSSubClass", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice)], aes(x=MSSubClass, y = SalePrice)) + geom_bar(stat = 'summary')
```

No summary function supplied, defaulting to 'mean_se()'



MoSold

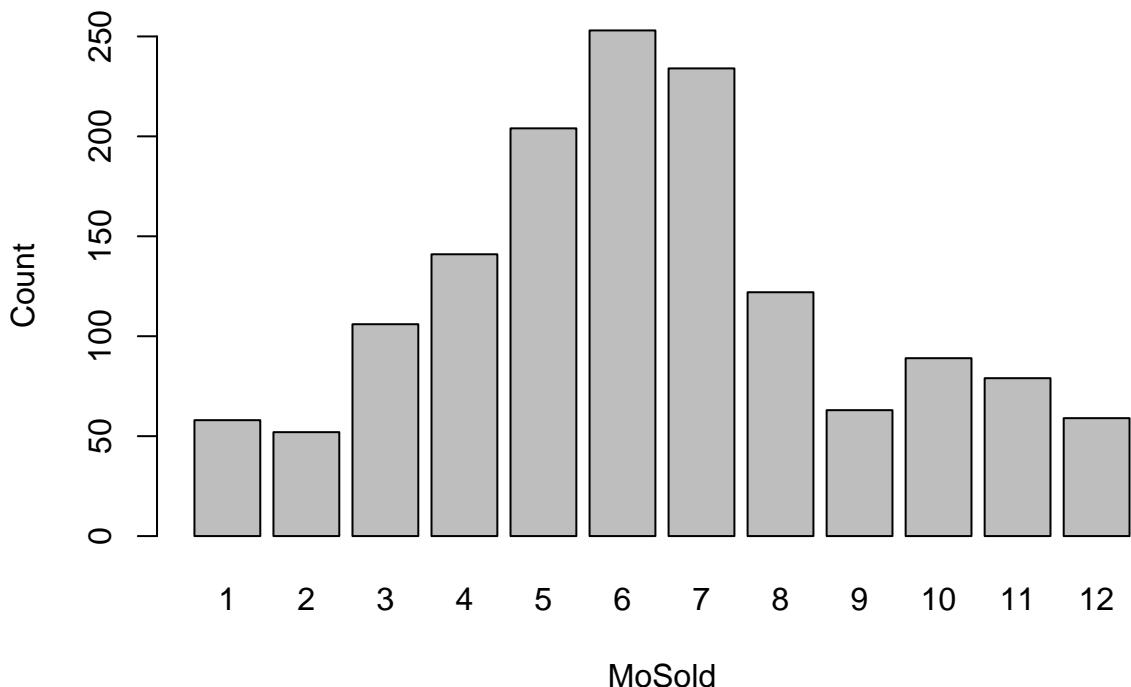
```
table(combined$MoSold)
```

```
##  
##   1   2   3   4   5   6   7   8   9   10  11  12  
##  58  52 106 141 204 253 234 122  63  89  79  59
```

```
combined$MoSold <- as.factor(combined$MoSold)  
table(combined$MoSold)
```

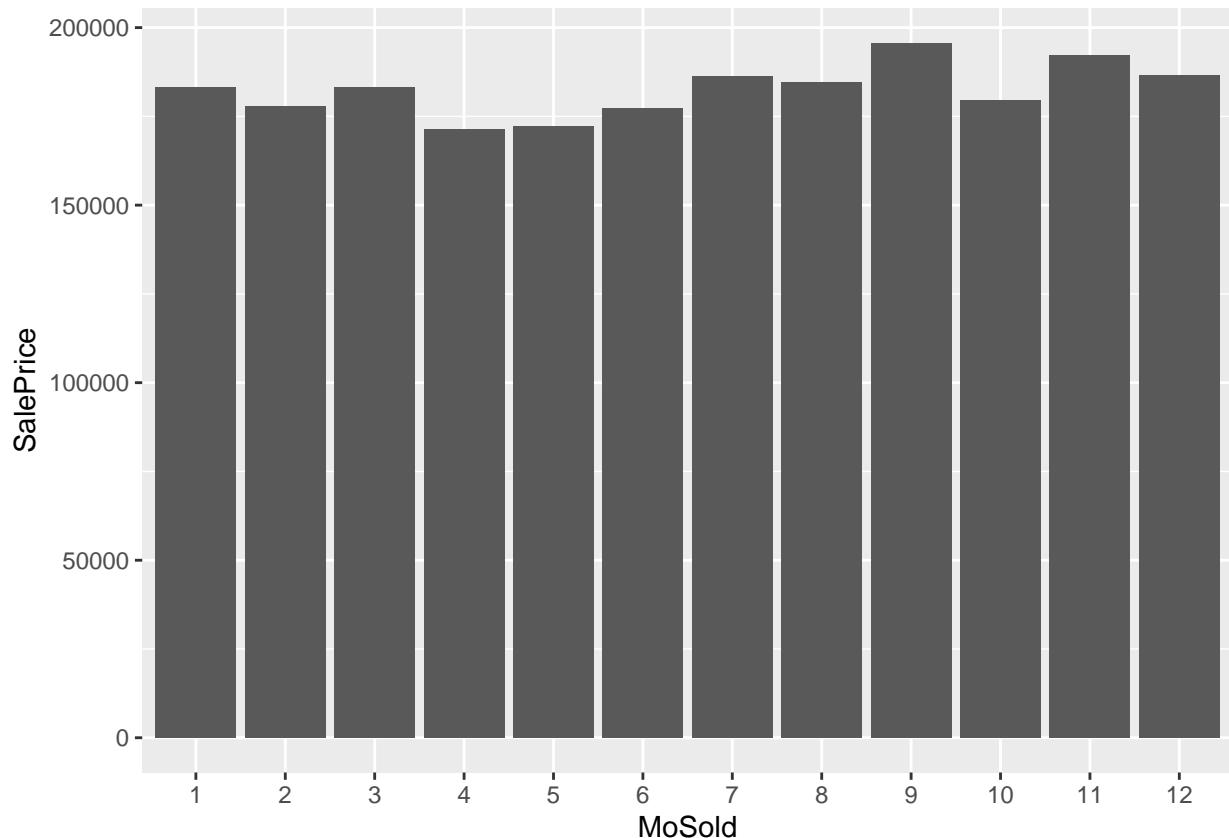
```
##  
##   1   2   3   4   5   6   7   8   9   10  11  12  
##  58  52 106 141 204 253 234 122  63  89  79  59
```

```
barplot(table(combined$MoSold), xlab = "MoSold", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=MoSold, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



YrSold

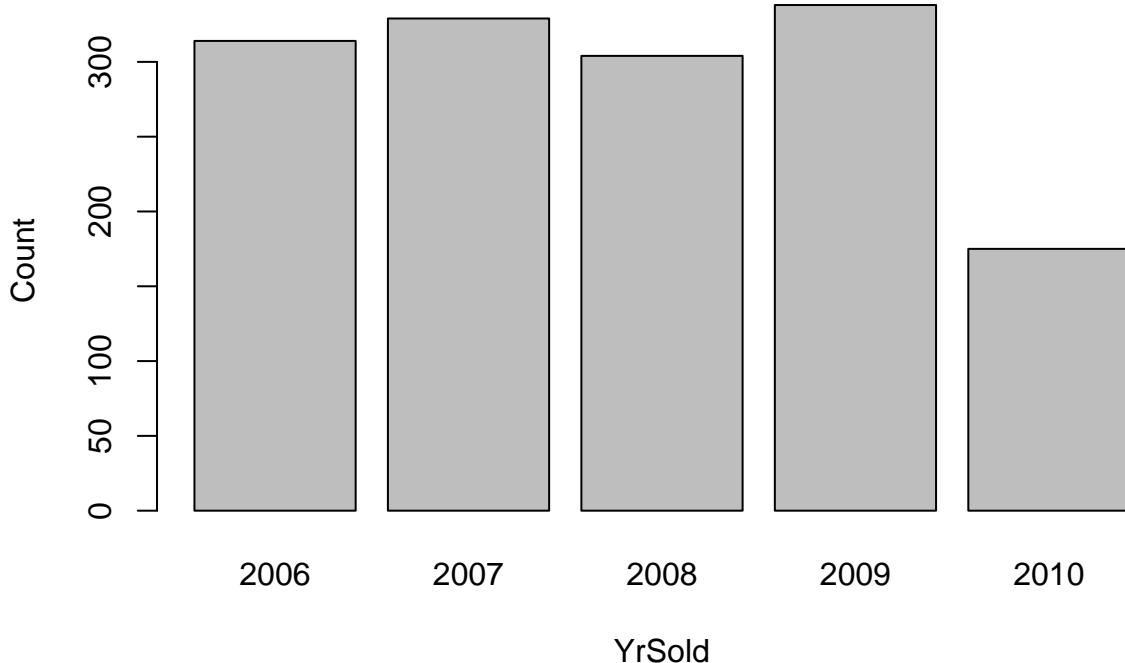
```
table(combined$YrSold)

##
## 2006 2007 2008 2009 2010
## 314 329 304 338 175

combined$YrSold <- as.factor(combined$YrSold)
table(combined$YrSold)

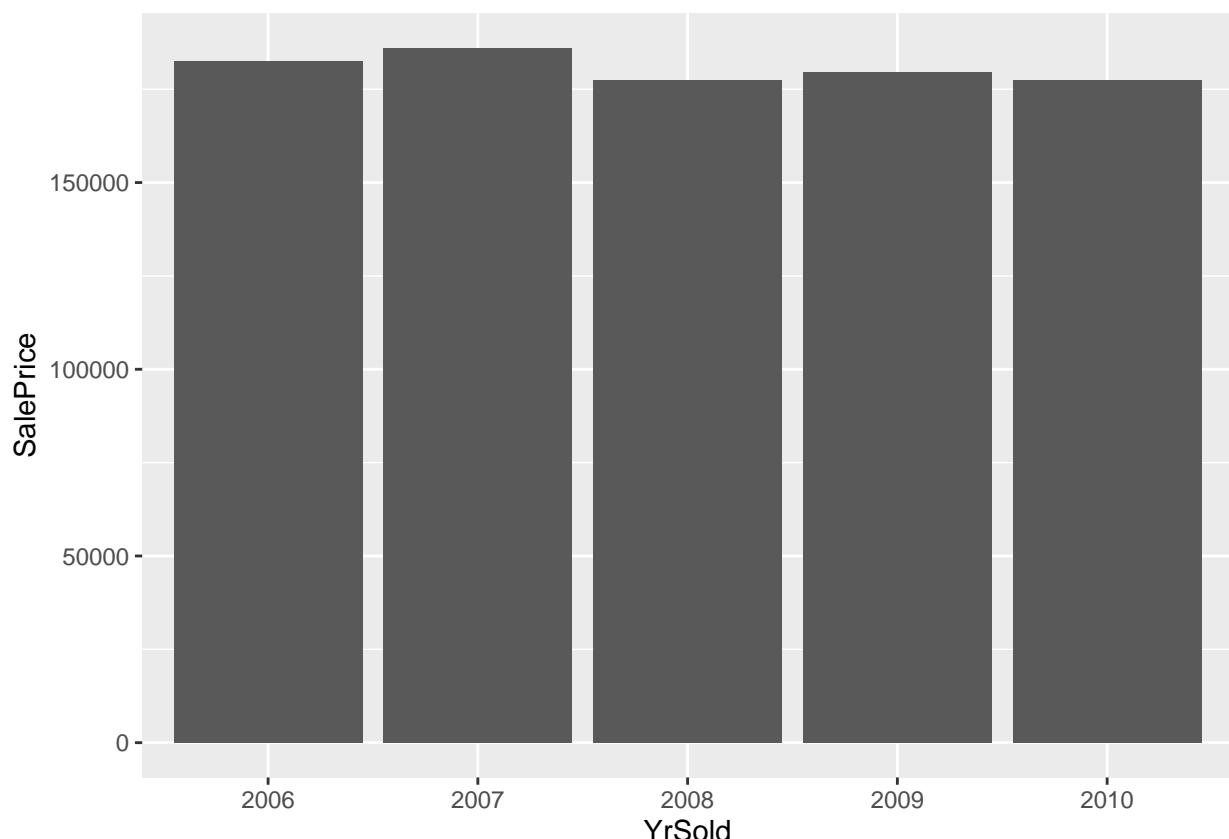
##
## 2006 2007 2008 2009 2010
## 314 329 304 338 175

barplot(table(combined$YrSold), xlab = "YrSold", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=YrSold, y = SalePrice)) + geom_bar(stat = 'summary')

## No summary function supplied, defaulting to 'mean_se()'
```



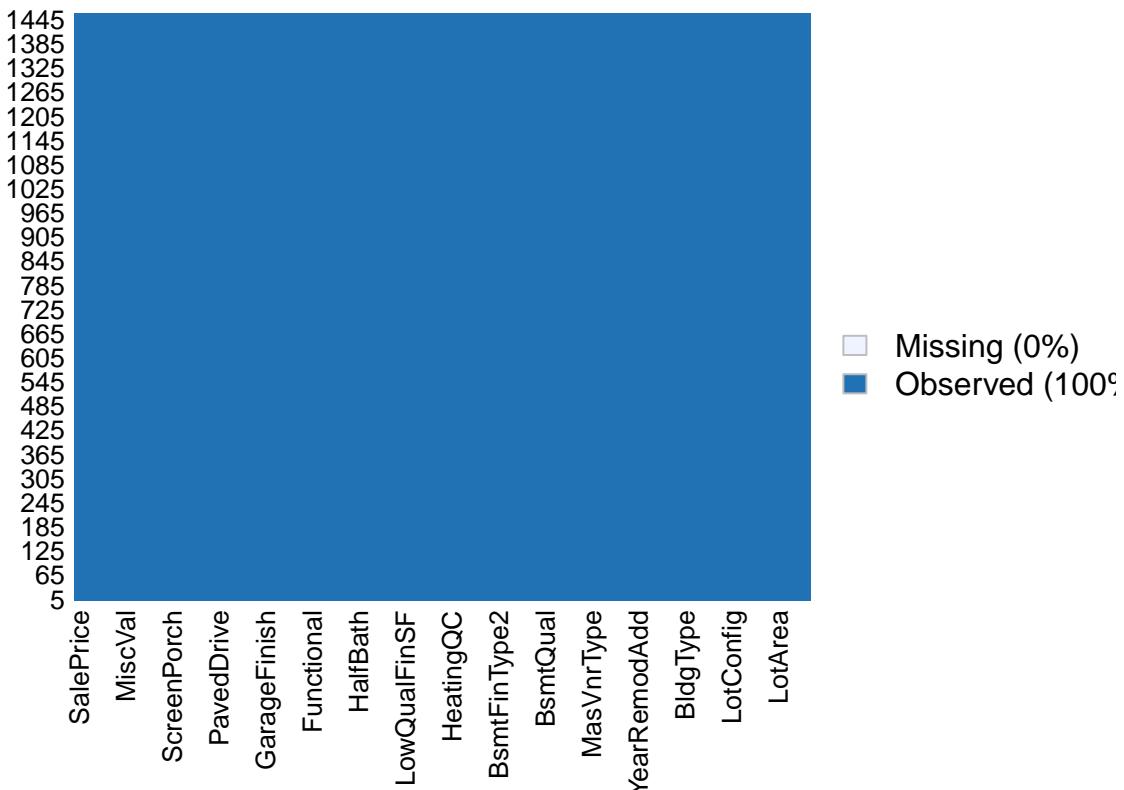
```
sort(colSums(sapply(combined, is.na)), decreasing = T)
```

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

```
misscounts <- sapply(combined, function(x) sum(is.na(x)))
```

```
missmap(combined, main = "Missing values")
```

Missing values



```
anyNA(combined)
```

```
## [1] FALSE
```

As we can see there are no missing values.

```
num_vars <- names(Filter(is.numeric,combined)) #index vector numeric variables
factor_vars <- names(Filter(is.factor,combined))
cat('numeric variables: ', length(num_vars), ' and categorical variables: ',length(factor_vars), '\n')

## numeric variables: 54 and categorical variables: 25

write.csv(combined, "../data/processed/clean_data.csv")
```

EDA

Summarize Datasets

```
# primary dataset
str(combined)

## 'data.frame': 1460 obs. of 79 variables:
## $ MSSubClass : Factor w/ 15 levels "1 story 1946+",...: 6 1 6 7 6 5 1 6 5 15 ...
## $ MSZoning : Factor w/ 5 levels "C (all)", "FV", ...: 4 4 4 4 4 4 4 4 4 5 4 ...
## $ LotFrontage : num 65 80 68 60 84 85 75 75 51 50 ...
## $ LotArea : int 8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
## $ Street : int 1 1 1 1 1 1 1 1 1 ...
## $ Alley : Factor w/ 3 levels "Grvl", "None", ...: 2 2 2 2 2 2 2 2 2 2 ...
```

```

## $ LotShape      : int 3 3 2 2 2 2 3 2 3 3 ...
## $ LandContour   : Factor w/ 4 levels "Bnk","HLS","Low",...: 4 4 4 4 4 4 4 4 4 ...
## $ LotConfig     : Factor w/ 5 levels "Corner","CulDSac",...: 5 3 5 1 3 5 5 1 5 1 ...
## $ LandSlope     : int 2 2 2 2 2 2 2 2 2 2 ...
## $ 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 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 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 ...
## $ RoofMatl      : Factor w/ 8 levels "ClyTile","CompShg",...: 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 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     : int 4 3 4 3 4 3 4 3 3 3 ...
## $ ExterCond     : int 3 3 3 3 3 3 3 3 3 3 ...
## $ Foundation    : Factor w/ 6 levels "BrkTil","CBlock",...: 3 2 3 1 3 6 3 2 1 1 ...
## $ BsmtQual      : int 4 4 4 3 4 4 5 4 3 3 ...
## $ BsmtCond      : int 3 3 3 4 3 3 3 3 3 3 ...
## $ BsmtExposure  : int 1 4 2 1 3 1 3 2 1 1 ...
## $ BsmtFinType1  : int 6 5 6 5 6 6 5 1 6 ...
## $ BsmtFinSF1    : int 706 978 486 216 655 732 1369 859 0 851 ...
## $ BsmtFinType2  : int 1 1 1 1 1 1 4 1 1 ...
## $ BsmtFinSF2    : int 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 ...
## $ HeatingQC     : int 5 5 5 4 5 5 5 4 5 ...
## $ CentralAir    : int 1 1 1 1 1 1 1 1 1 ...
## $ 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 ...
## $ GrLivArea     : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
## $ BsmtFullBath  : num 1 0 1 1 1 1 1 0 1 ...
## $ BsmtHalfBath  : num 0 1 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    : int 4 3 4 4 4 3 4 3 3 3 ...
## $ TotRmsAbvGrd  : int 8 6 6 7 9 5 7 7 8 5 ...
## $ Functional     : int 7 7 7 7 7 7 7 7 6 7 ...
## $ Fireplaces     : int 0 1 1 1 1 0 1 2 2 2 ...
## $ FireplaceQu   : int 0 3 3 4 3 0 4 3 3 3 ...
## $ GarageType     : Factor w/ 7 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   : int 2 2 2 1 2 1 2 2 1 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     : int 3 3 3 3 3 3 3 3 2 4 ...
## $ GarageCond     : int 3 3 3 3 3 3 3 3 3 3 ...
## $ PavedDrive    : int 2 2 2 2 2 2 2 2 2 ...
## $ 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 ...
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 ...
## $ PoolQC : int 0 0 0 0 0 0 0 0 0 ...
## $ 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 : Factor w/ 12 levels "1","2","3","4",...: 2 5 9 2 12 10 8 11 4 1 ...
## $ YrSold : Factor w/ 5 levels "2006","2007",...: 3 2 3 1 3 4 2 4 3 3 ...
## $ SaleType : Factor w/ 9 levels "COD","Con", "ConLD",...: 9 9 9 9 9 9 9 9 9 ...
## $ SaleCondition: Factor w/ 6 levels "Abnrmal","AdjLand",...: 5 5 5 1 5 5 5 5 1 5 ...
## $ SalePrice : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...

```

Numerical variables

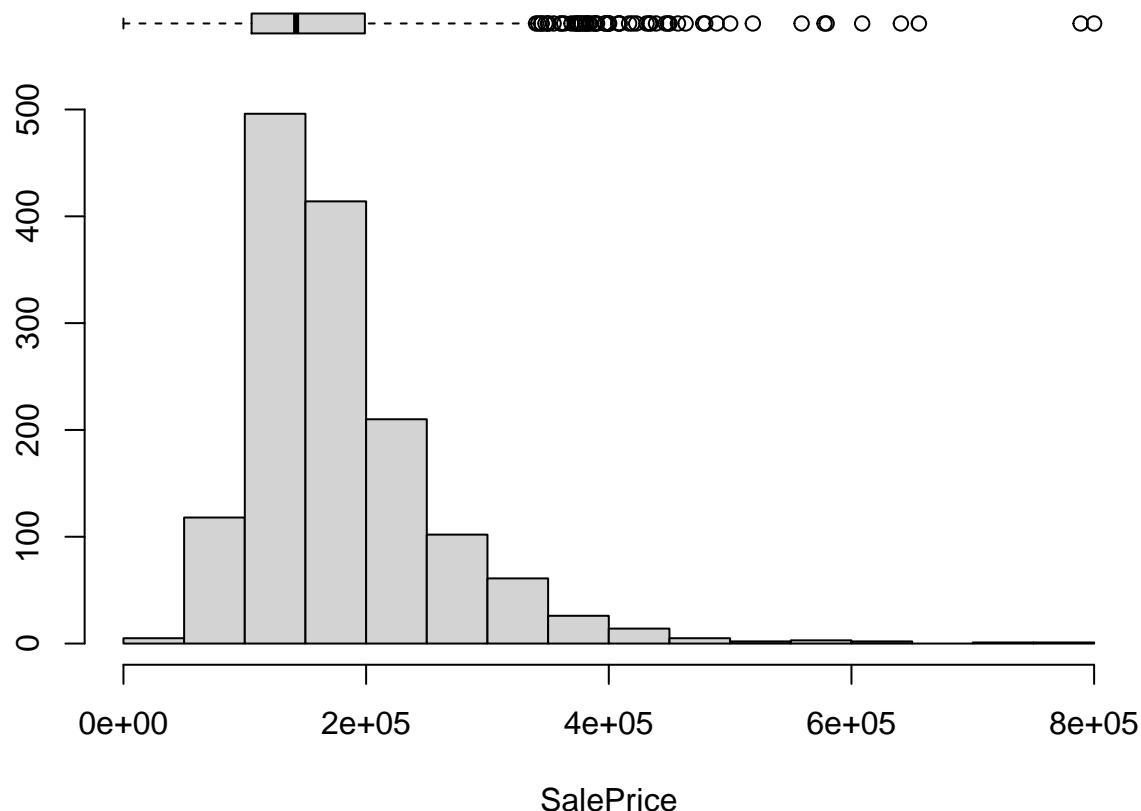
SalePrice variable

```

library(ggplot2)
layout(mat = matrix(c(1,2),2,1, byrow=TRUE), height = c(1,8))
par(mar=c(0, 3.1, 1.1, 2.1))
boxplot(combined$SalePrice , horizontal=TRUE , xaxt="n", frame=F, main=sprintf('Histogram of SalePrice'))
par(mar=c(4, 3.1, 1.1, 2.1))
hist(combined$SalePrice,main=' ', xlab = "SalePrice", ylab = "count")

```

Histogram of SalePrice



SalePrice vs all other numerical variable scatterplots

```

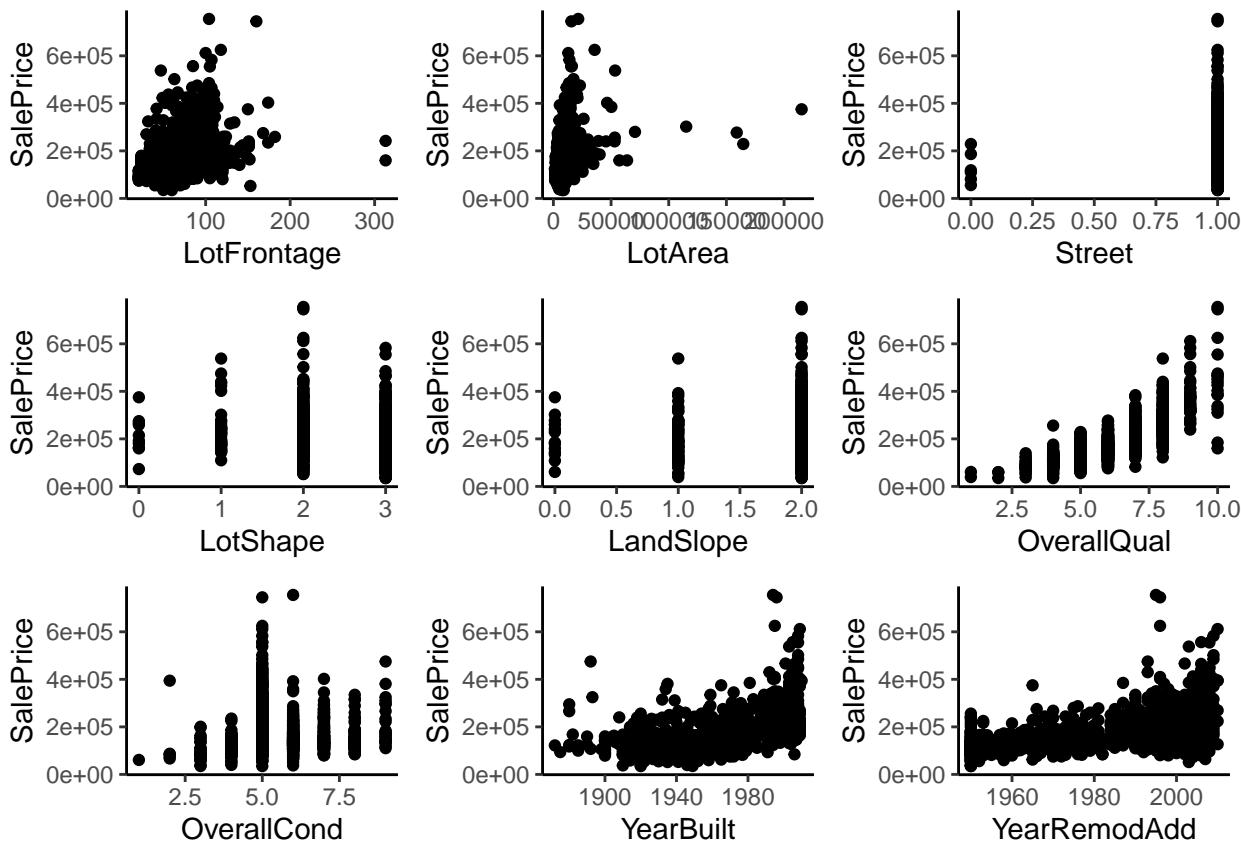
plot = function(variable){
  print(variable)
  ggplot(combined,aes(x = combined[,variable], y = SalePrice)) + geom_point() + theme_classic() + labs(x=variable)
  library(gridExtra)
}

```

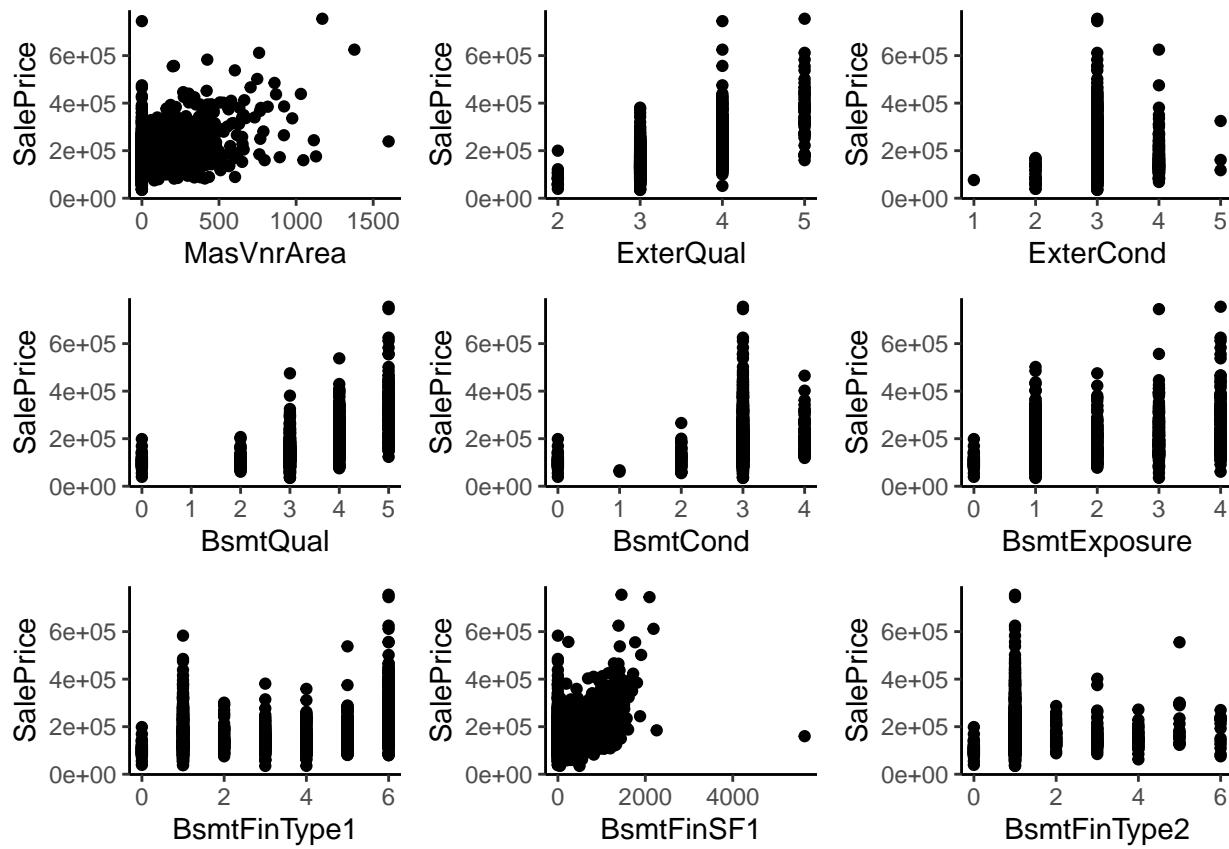
```
## Warning: package 'gridExtra' was built under R version 4.0.3
```

```
p = list()
p <- NULL
val <- 0
d <- combined[,num_vars]
for(j in 1:6){
  for(i in 1:9){
    name = names(d[i+val])
    p[[i]] = plot(as.character(name)))
  val = i+val
  do.call(grid.arrange,p)
  p <- NULL}
```

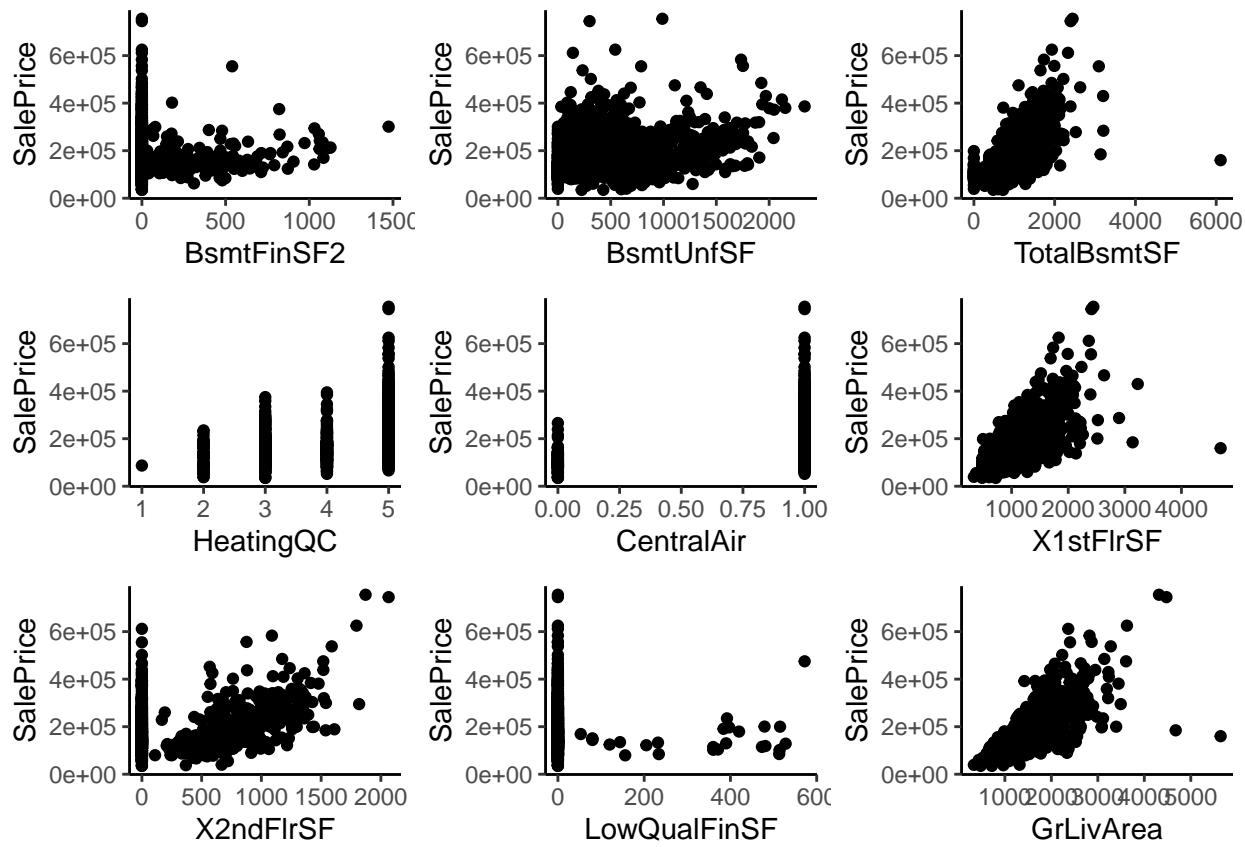
```
## [1] "LotFrontage"
## [1] "LotArea"
## [1] "Street"
## [1] "LotShape"
## [1] "LandSlope"
## [1] "OverallQual"
## [1] "OverallCond"
## [1] "YearBuilt"
## [1] "YearRemodAdd"
```



```
## [1] "MasVnrArea"
## [1] "ExterQual"
## [1] "ExterCond"
## [1] "BsmtQual"
## [1] "BsmtCond"
## [1] "BsmtExposure"
## [1] "BsmtFinType1"
## [1] "BsmtFinSF1"
## [1] "BsmtFinType2"
```



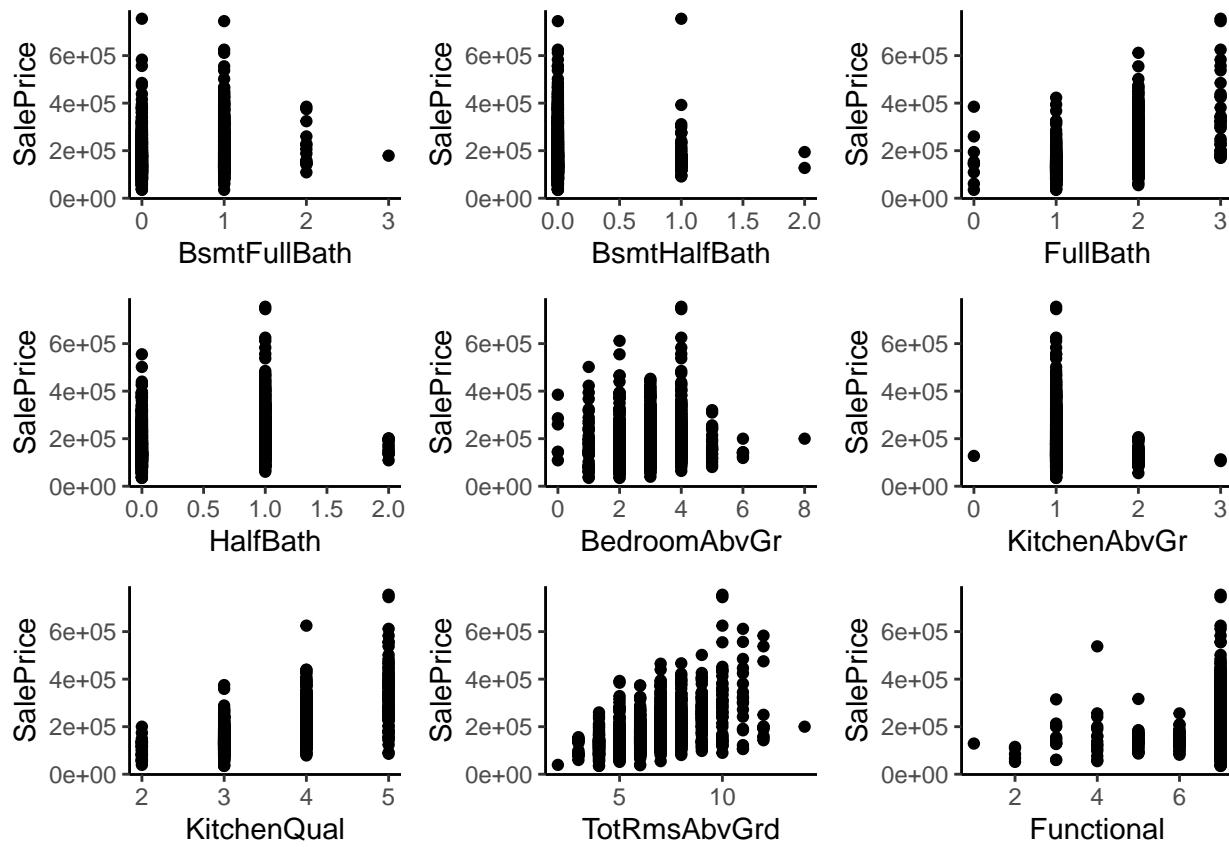
```
## [1] "BsmtFinSF2"
## [1] "BsmtUnfSF"
## [1] "TotalBsmtSF"
## [1] "HeatingQC"
## [1] "CentralAir"
## [1] "X1stFlrSF"
## [1] "X2ndFlrSF"
## [1] "LowQualFinSF"
## [1] "GrLivArea"
```



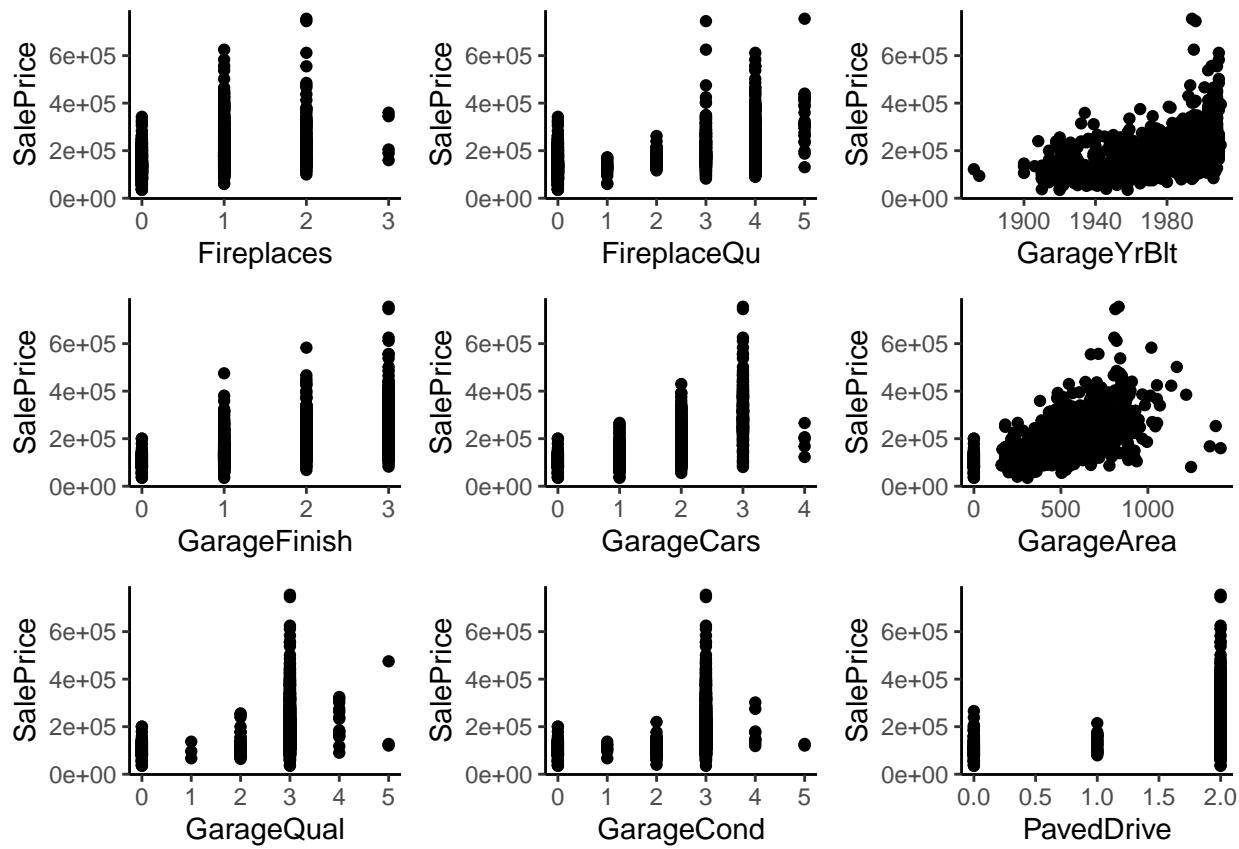
```

## [1] "BsmtFullBath"
## [1] "BsmtHalfBath"
## [1] "FullBath"
## [1] "HalfBath"
## [1] "BedroomAbvGr"
## [1] "KitchenAbvGr"
## [1] "KitchenQual"
## [1] "TotRmsAbvGrd"
## [1] "Functional"

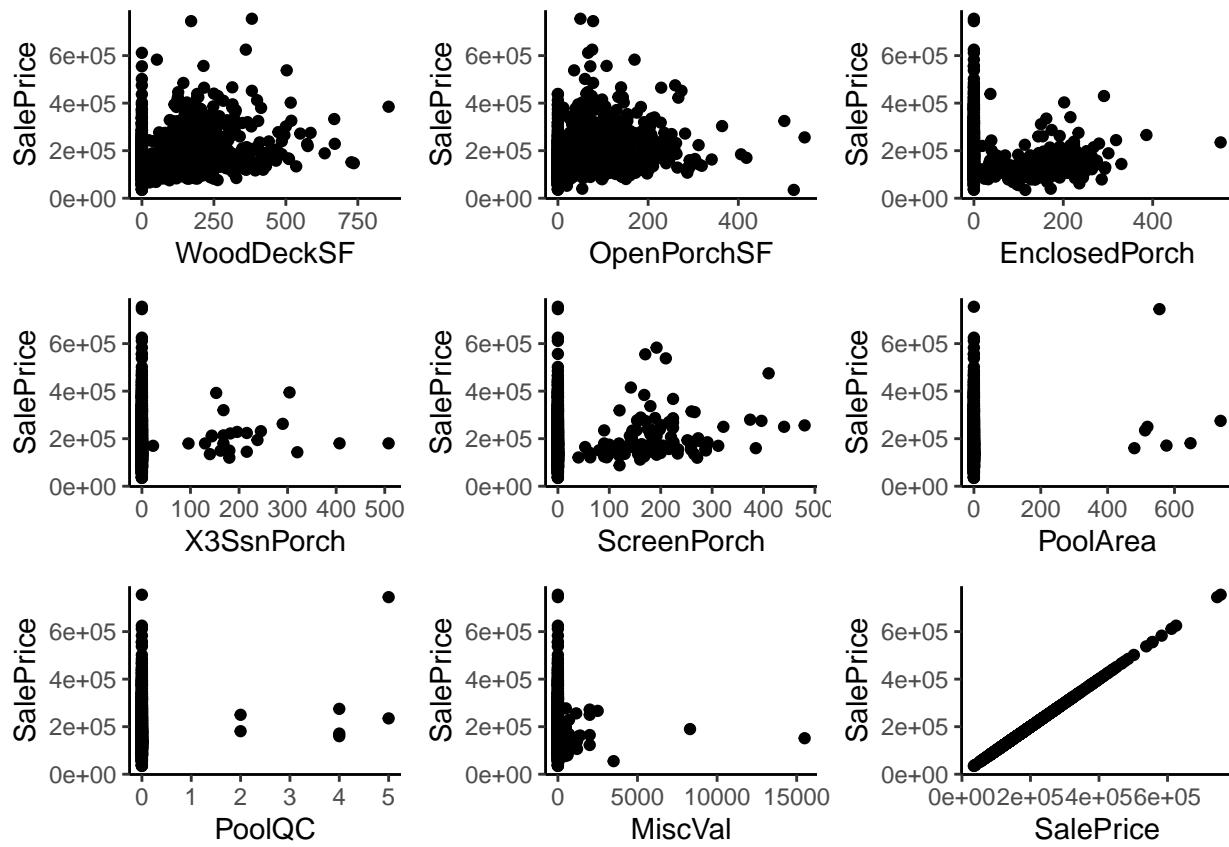
```



```
## [1] "Fireplaces"
## [1] "FireplaceQu"
## [1] "GarageYrBlt"
## [1] "GarageFinish"
## [1] "GarageCars"
## [1] "GarageArea"
## [1] "GarageQual"
## [1] "GarageCond"
## [1] "PavedDrive"
```



```
## [1] "WoodDeckSF"
## [1] "OpenPorchSF"
## [1] "EnclosedPorch"
## [1] "X3SsnPorch"
## [1] "ScreenPorch"
## [1] "PoolArea"
## [1] "PoolQC"
## [1] "MiscVal"
## [1] "SalePrice"
```



all numerical variables density plots

```

library(purrr)

## Warning: package 'purrr' was built under R version 4.0.3

##
## Attaching package: 'purrr'

## The following object is masked from 'package:plyr':
##     compact

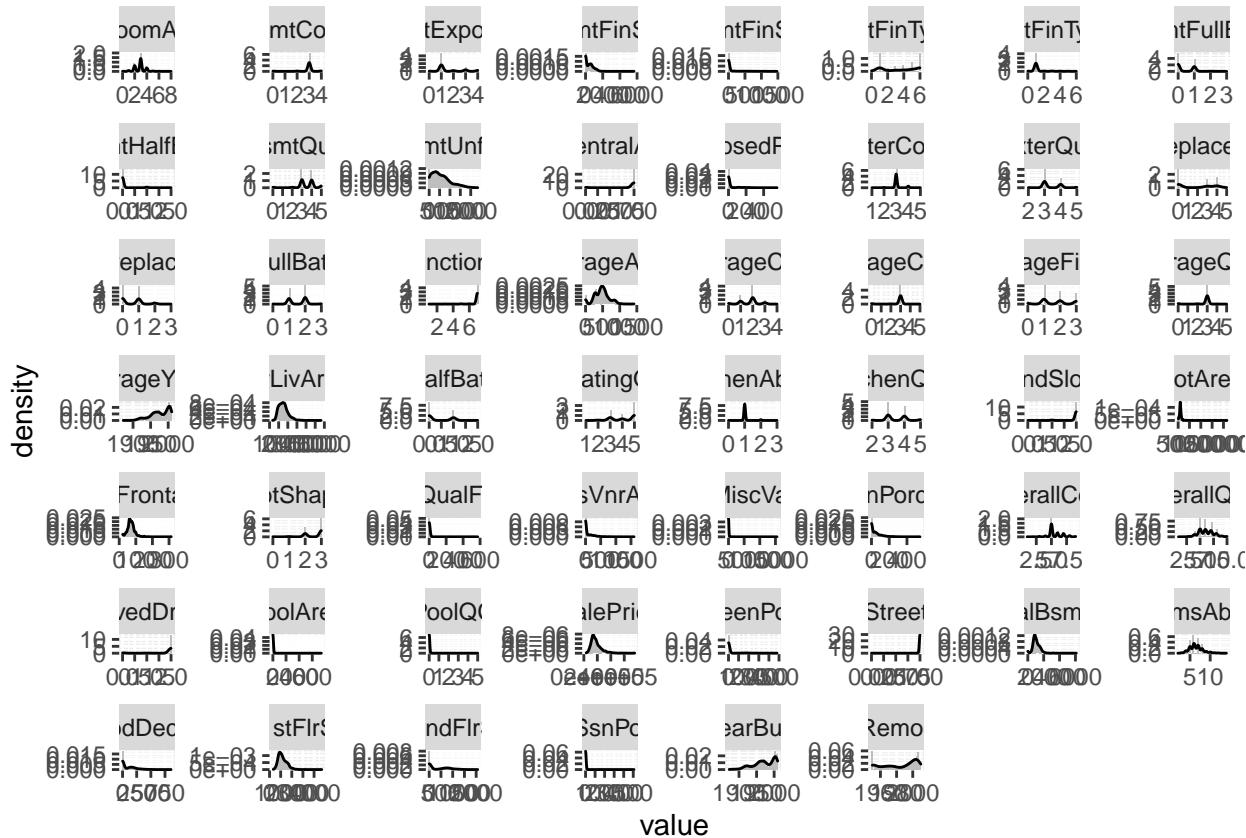
library(tidyr)

## Warning: package 'tidyverse' was built under R version 4.0.3

library(ggplot2)
combined %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram(aes(y=..density..), fill = "grey") +
    geom_density()

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

```



```
ggsave("../Plots/density_numerical_variables.jpg", plot = last_plot(), width = 10, height = 7)
```

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

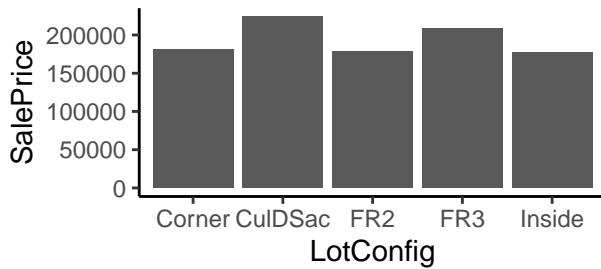
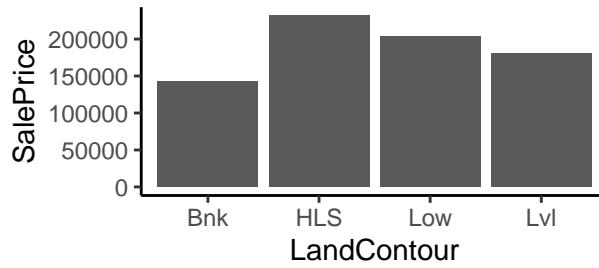
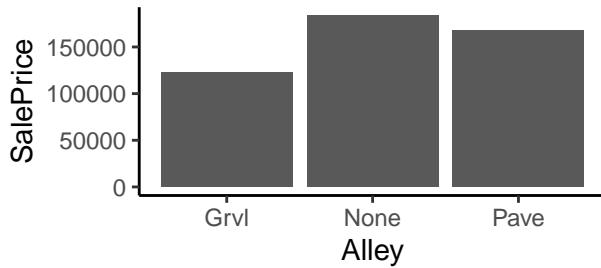
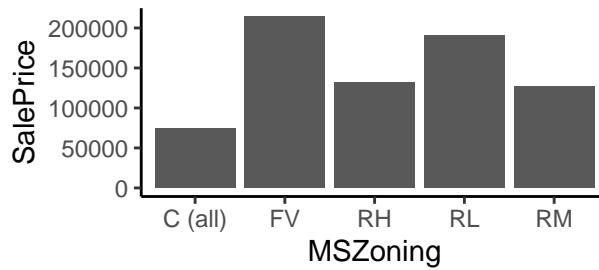
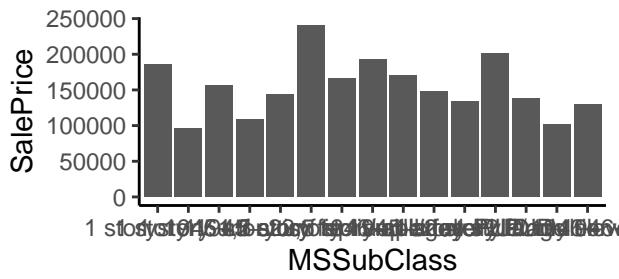
Categorical variables

Histograms for Categorical Variables

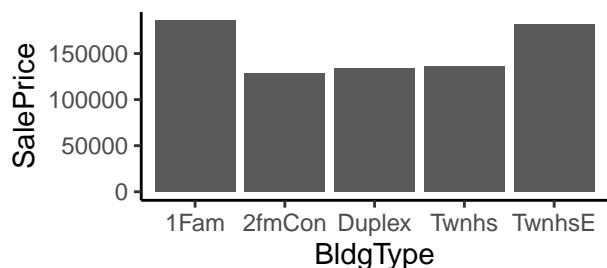
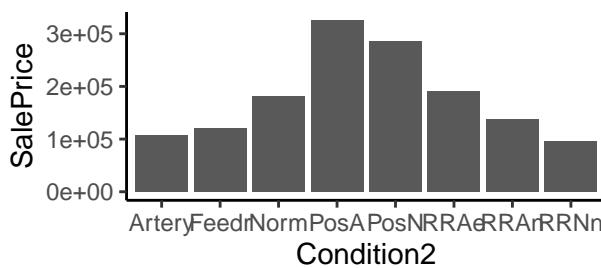
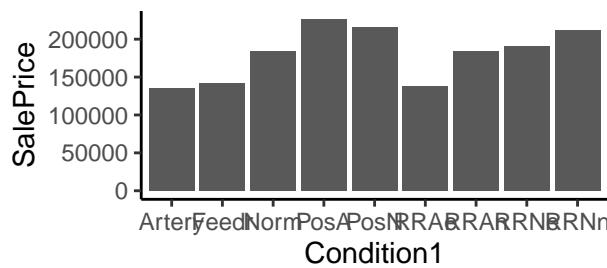
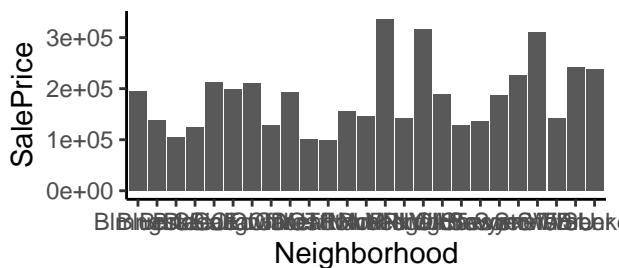
```
plot_factor = function(variable){
  ggplot(combined, aes(x = combined[,variable], y = SalePrice)) + geom_bar(stat = 'summary') + theme_classic() + theme(legend.position = 'none')

library(gridExtra)
p = list()
p <- NULL
val <- 0
d <- combined[,factor_vars]
for(j in 1:5){
  for(i in 1:5){
    name = names(d[i+val])
    p[[i]] = plot_factor((name))
    val = i+val
  }
  do.call(grid.arrange,p)
  p <- NULL}

## No summary function supplied, defaulting to 'mean_se()'
```



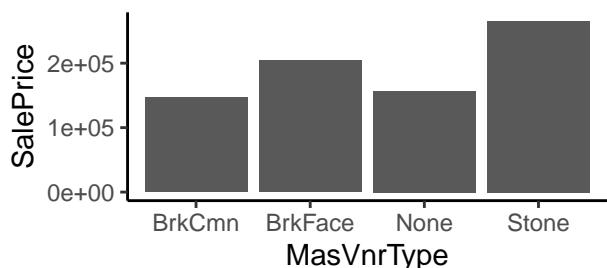
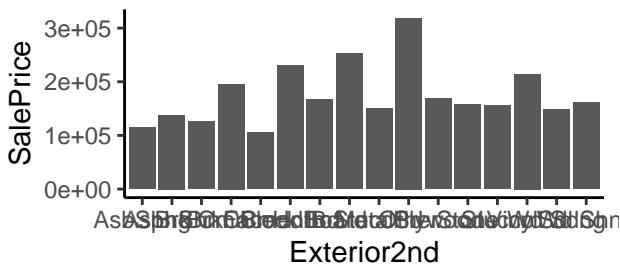
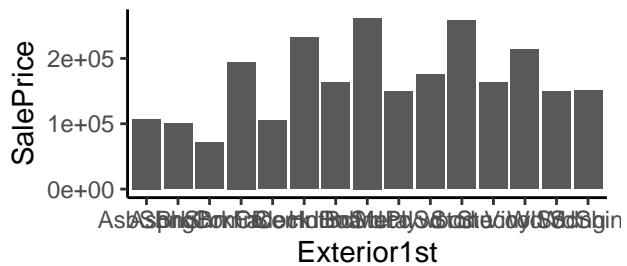
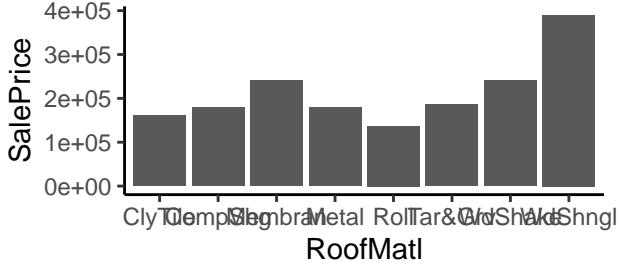
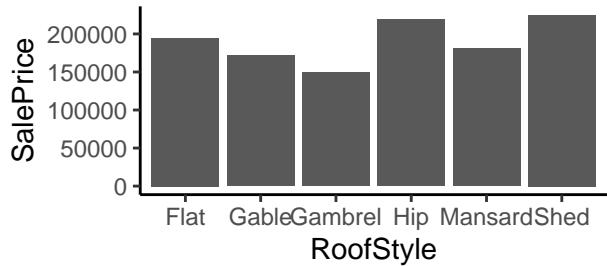
```
## No summary function supplied, defaulting to 'mean_se()'  
## No summary function supplied, defaulting to 'mean_se()'
```



```

## No summary function supplied, defaulting to 'mean_se()'

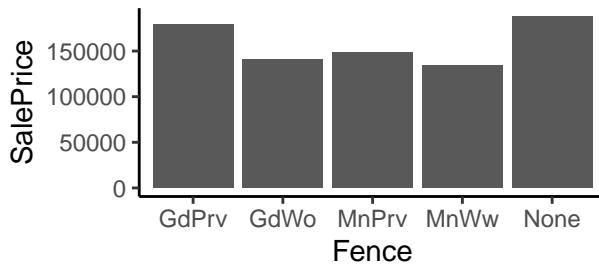
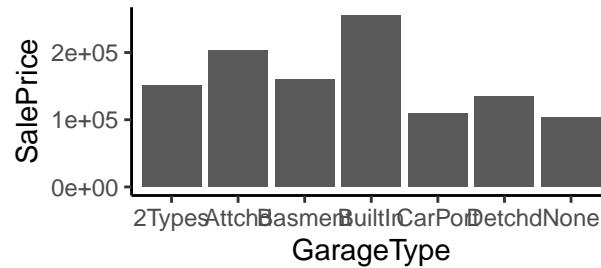
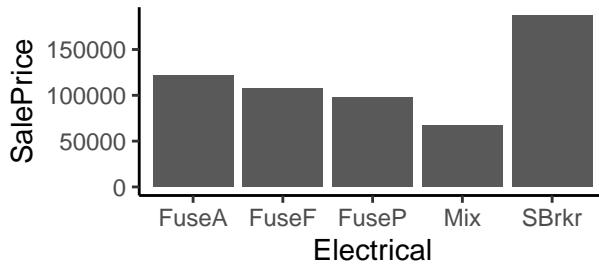
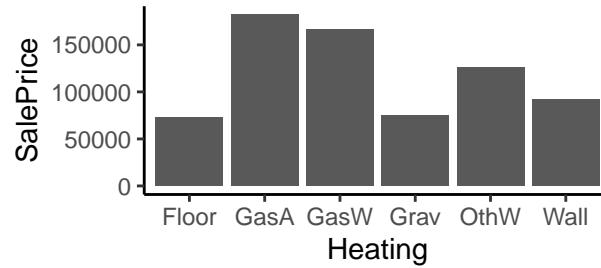
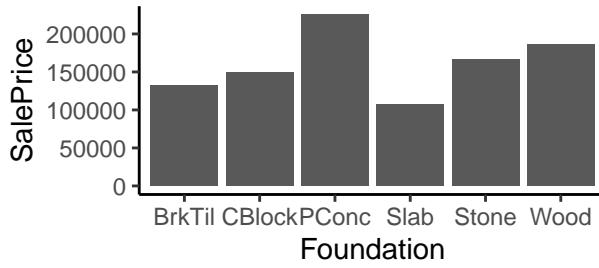
```



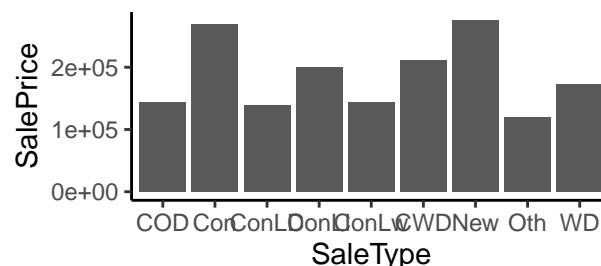
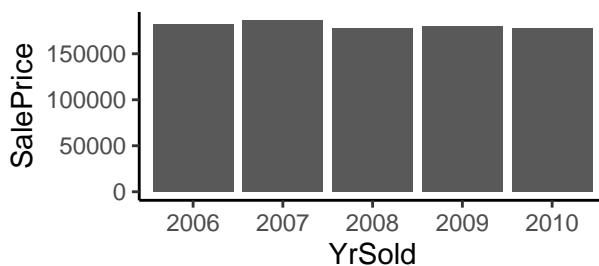
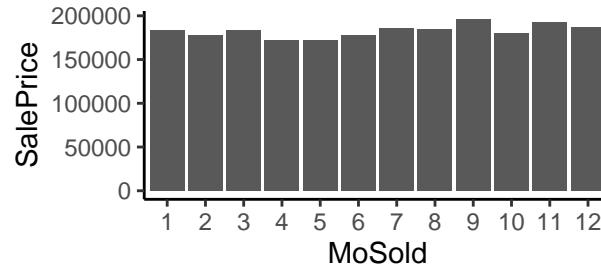
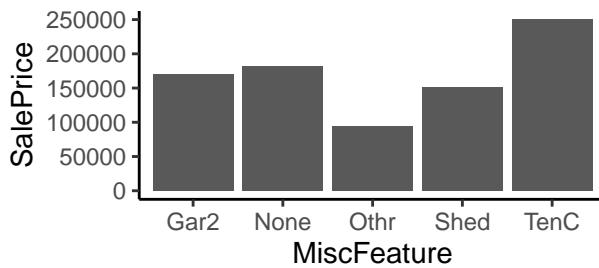
```

## No summary function supplied, defaulting to 'mean_se()'

```



```
## No summary function supplied, defaulting to 'mean_se()'
```



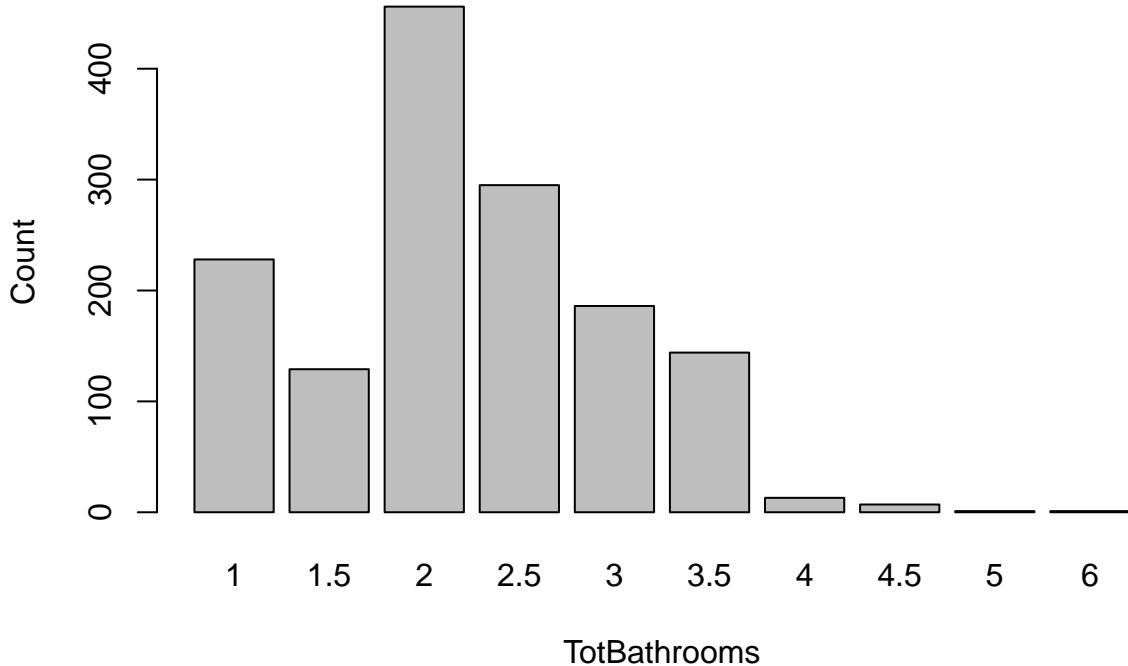
Feature engineering

Total bathrooms

```
combined$TotBathrooms <- combined$FullBath + (combined$HalfBath*0.5) + combined$BsmtFullBath + (combined$BsmtHal
```

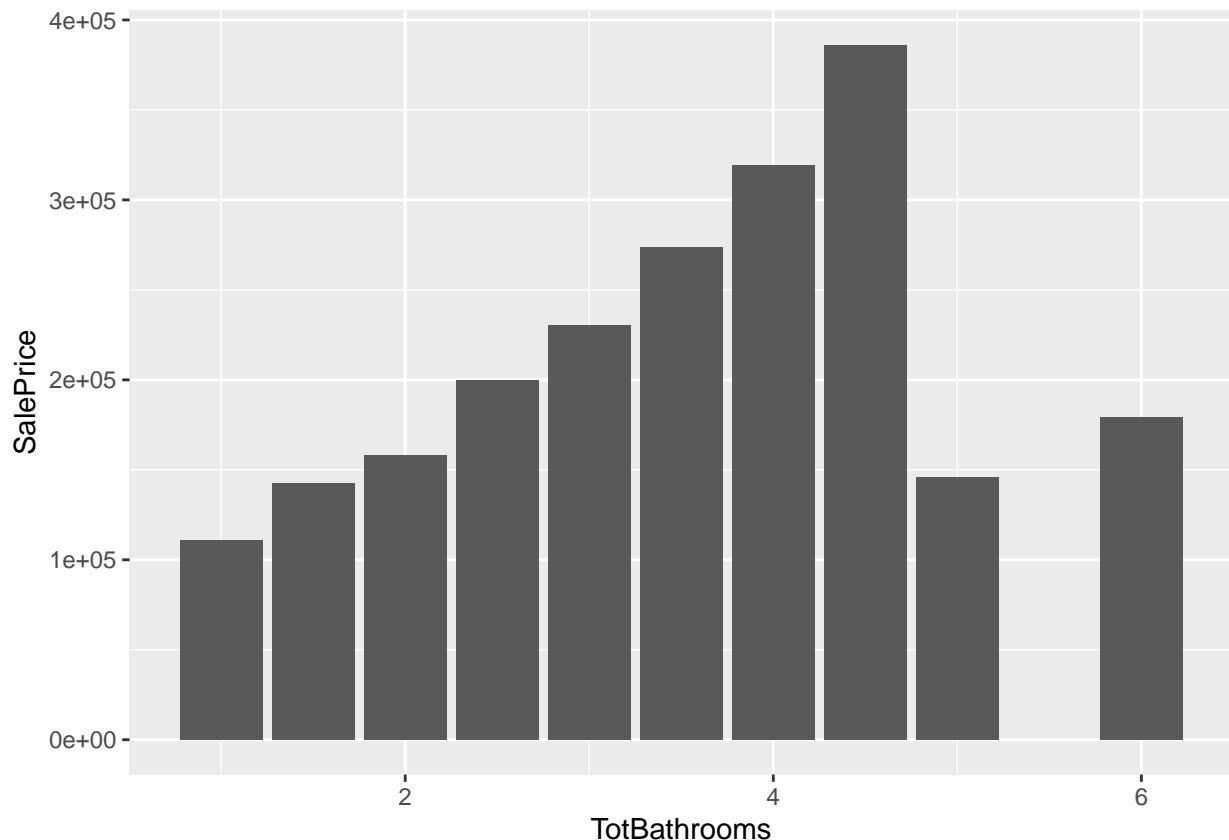
```
##  
##   1 1.5   2 2.5   3 3.5   4 4.5   5   6  
## 228 129 456 295 186 144  13    7    1    1
```

```
barplot(table(combined$TotBathrooms), xlab = "TotBathrooms", ylab = "Count")
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=TotBathrooms, y = SalePrice)) + geom_bar(stat = 'summary')
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



House age and remodelled houses

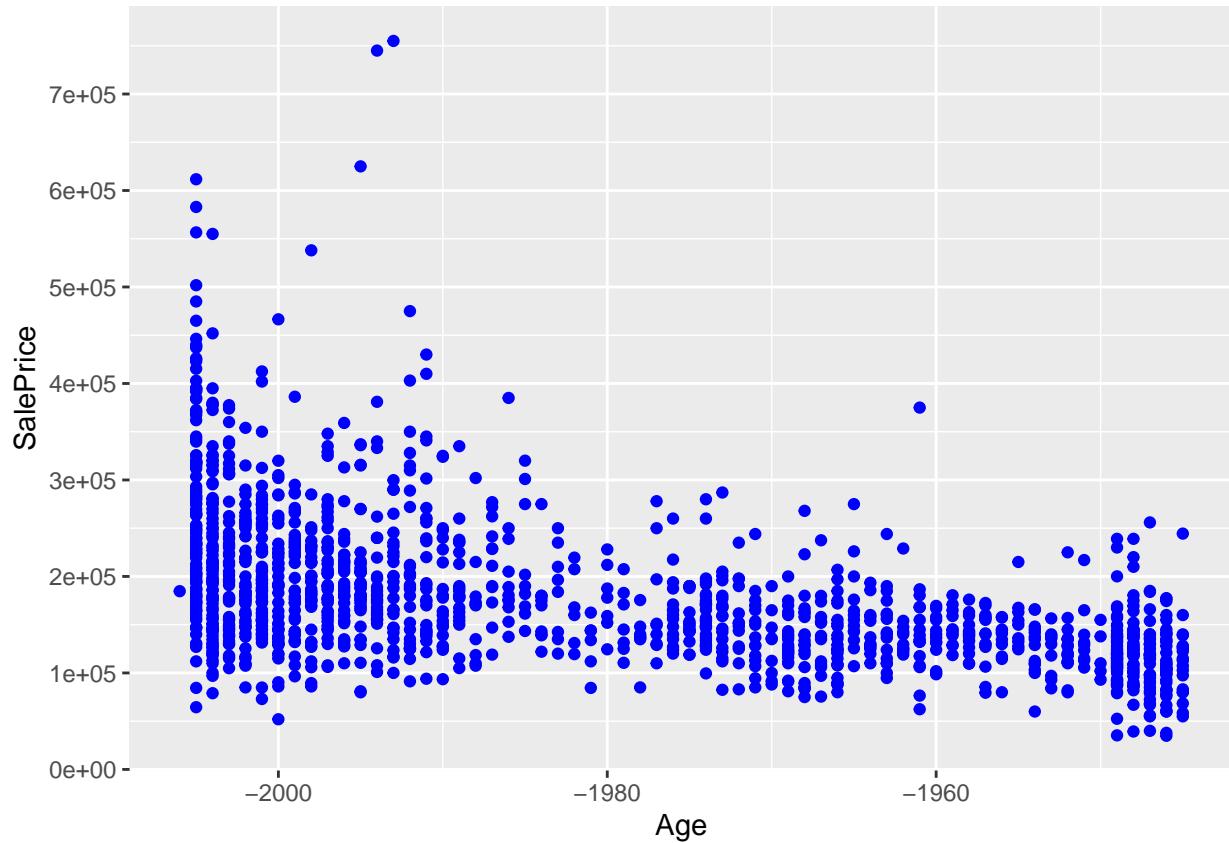
After looking at the density plots and EDA, we found that there is no house age but there is remodeled year and year built variables so adding the age of the house to the data set and whether or not its been remodeled was important as Sale price does depend on how old the house is and whether or not its a remodeled house.

house age plot

```

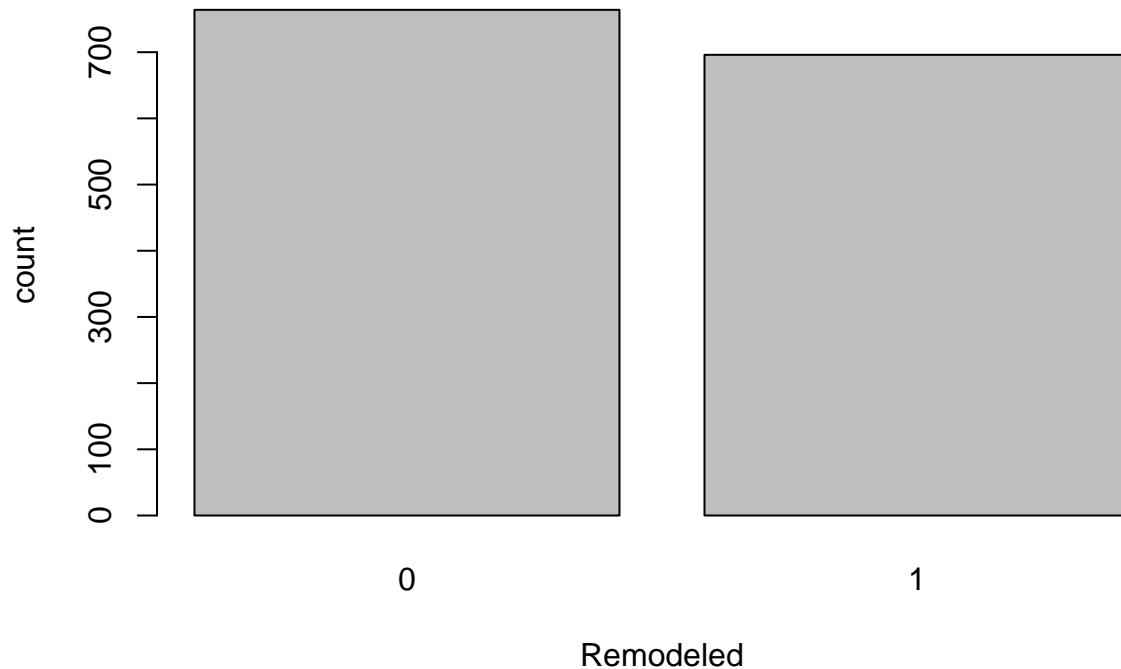
combined$Remod <- ifelse(combined$YearBuilt==combined$YearRemodAdd, 0, 1) #0=No Remodeling, 1=Remodeling
combined$Age <- as.numeric(combined$YrSold)-combined$YearRemodAdd
ggplot(data=combined[!is.na(combined$SalePrice),], aes(x=Age, y=SalePrice))+
  geom_point(col='blue')+
  scale_y_continuous(breaks= seq(0, 800000, by=100000))

```



histogram of remodeled house variable

```
barplot(table(combined$Remod), xlab = "Remodeled", ylab = "count")
```

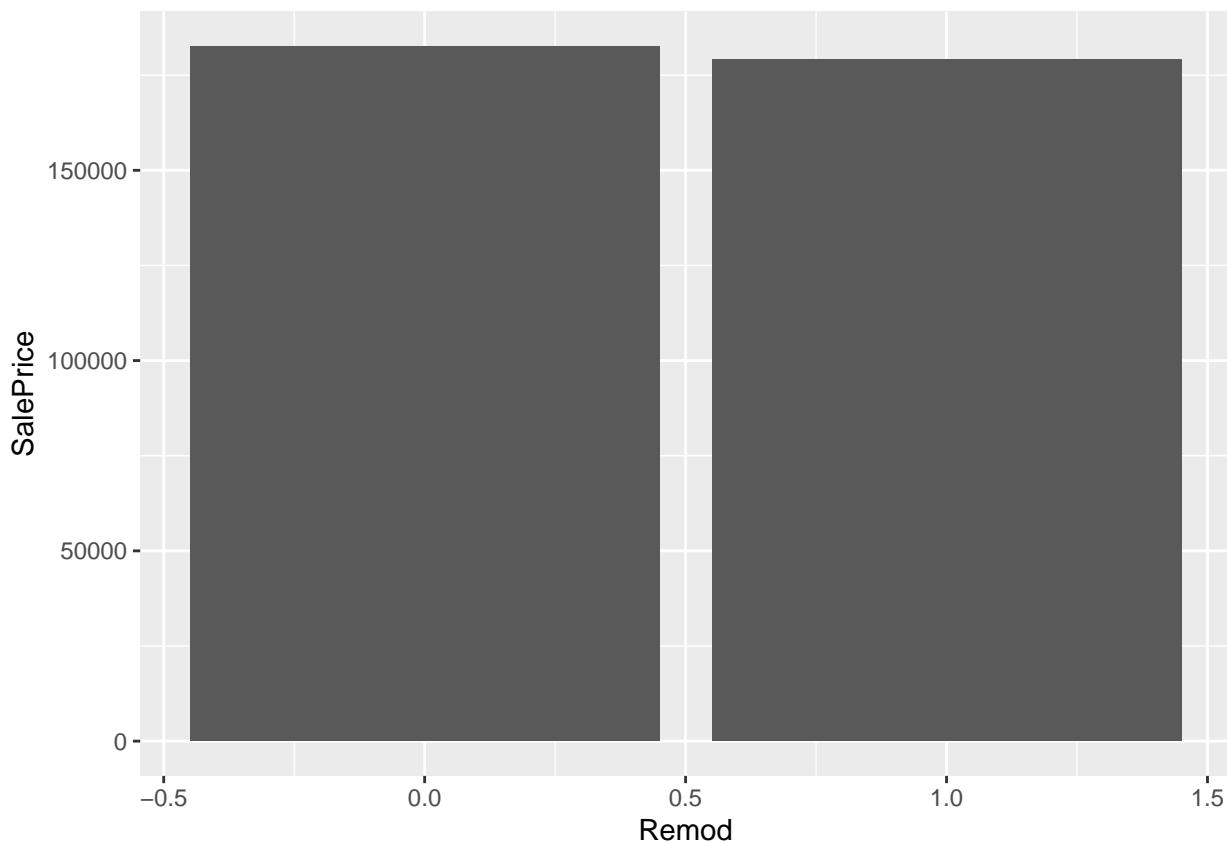


```

ggplot(combined[!is.na(combined$SalePrice),], aes(x=Remod, y = SalePrice)) + geom_bar(stat = 'summary')

## No summary function supplied, defaulting to 'mean_se()'

```



Correlations

correlation matrix

```

num_vars <- names(Filter(is.numeric,combined)) #index vector numeric variables
factor_vars <- names(Filter(is.factor,combined))

corr_matrix <- cor(combined[num_vars],method = "pearson")
correlation_SalePrice <- sort(corr_matrix[, 'SalePrice'],decreasing = TRUE)
correlation_SalePrice

```

	SalePrice	OverallQual	GrLivArea	ExterQual	KitchenQual
##	1.000000000	0.790981601	0.708624478	0.682639242	0.659599721
##	GarageCars	TotBathrooms	GarageArea	TotalBsmtSF	X1stFlrSF
##	0.640409197	0.631731068	0.623431439	0.613580552	0.605852185
##	BsmtQual	FullBath	GarageFinish	TotRmsAbvGrd	YearBuilt
##	0.585207199	0.560663763	0.549246756	0.533723156	0.522897333
##	FireplaceQu	GarageYrBlt	YearRemodAdd	MasVnrArea	Fireplaces
##	0.520437606	0.508043287	0.507100967	0.472614499	0.466928837
##	HeatingQC	BsmtFinSF1	BsmtExposure	LotFrontage	WoodDeckSF
##	0.427648707	0.386419806	0.375044959	0.340871631	0.324413445
##	X2ndFlrSF	OpenPorchSF	BsmtFinType1	HalfBath	GarageQual
##	0.319333803	0.315856227	0.304907873	0.284107676	0.273839074
##	LotArea	GarageCond	CentralAir	PavedDrive	BsmtFullBath

```

## 0.263843354 0.263190784 0.251328164 0.231356952 0.227122233
## BsmtUnfSF BsmtCond BedroomAbvGr PoolQC ScreenPorch
## 0.214479106 0.212607156 0.168213154 0.111695838 0.111446571
## Functional PoolArea X3SsnPorch Street ExterCond
## 0.107618893 0.092403549 0.044583665 0.041035536 0.018899118
## BsmtFinType2 BsmtFinSF2 BsmtHalfBath MiscVal Remod
## -0.004329316 -0.011378121 -0.016844154 -0.021189580 -0.021932600
## LowQualFinSF LandSlope OverallCond EnclosedPorch KitchenAbvGr
## -0.025606130 -0.051152248 -0.077855894 -0.128577958 -0.135907371
## LotShape Age
## -0.267759314 -0.509078738

```

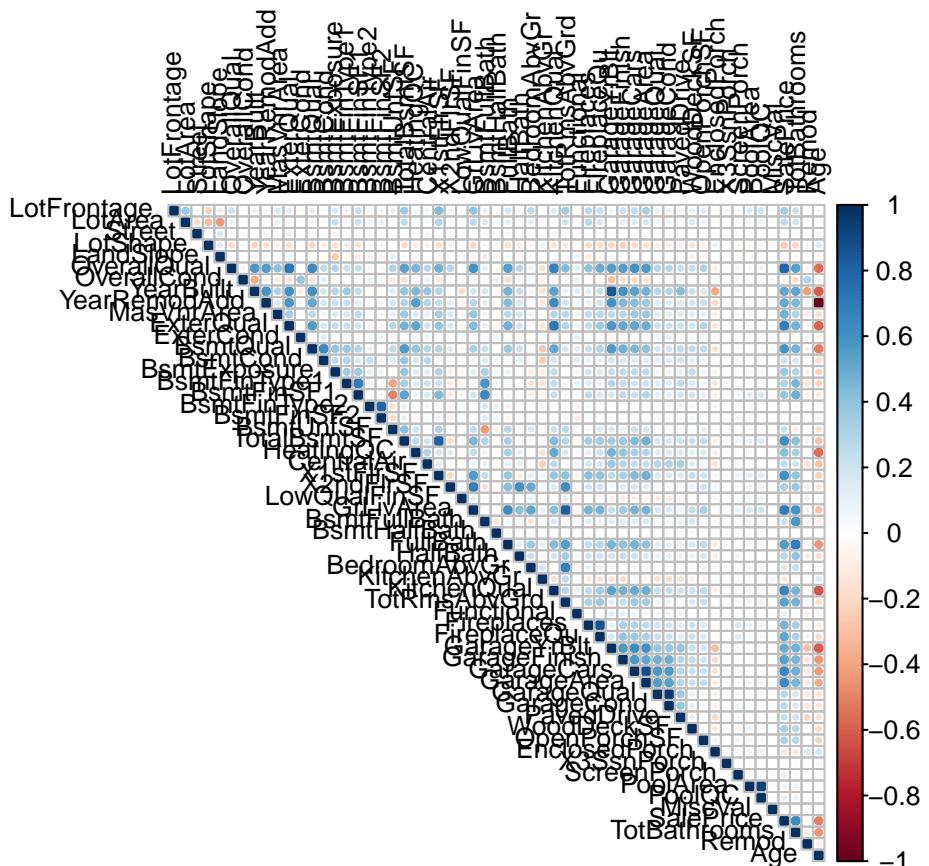
correlation Plot

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.0.3
```

```
## corrplot 0.84 loaded
```

```
corrplot(corr_matrix,type = "upper",tl.col = "black",tl.cex = 0.8)
```



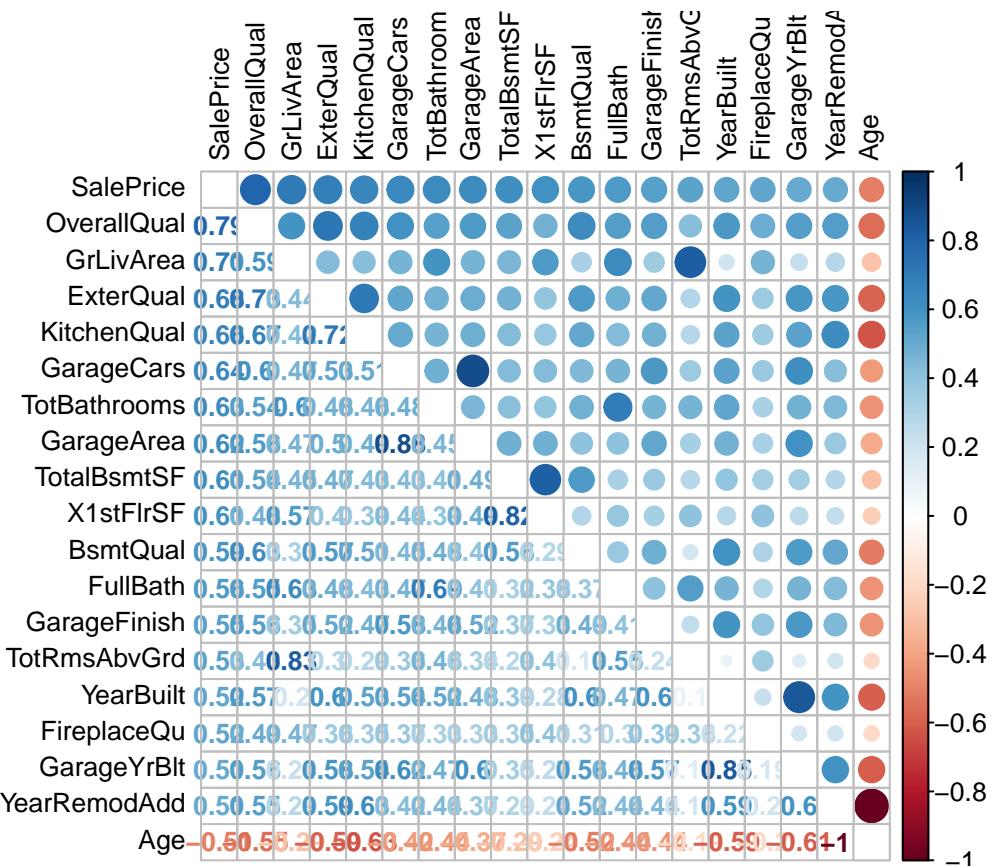
correlation Plot of variables vs SalePrice > 0.5

```

corr_sorted <- as.matrix(sort(corr_matrix[, 'SalePrice'], decreasing = TRUE))
#select only high correlations
Corr_High <- names(which(apply(corr_sorted, 1, function(x) abs(x)>0.5)))
corr_matrix <- corr_matrix[Corr_High, Corr_High]

```

```
corrplot.mixed(corr_matrix, tl.col="black", tl.pos = "lt", tl.cex = 0.8, cl.cex = 0.8, number.cex=0.8)
```



Removing highly correlated variables

```
# removing variables with high correlations between predictor # variables
dropVars <- c('GarageYrBlt', 'GarageArea', 'TotalBsmtSF', 'TotalRmsAbvGrd')
combined <- combined[, !names(combined) %in% dropVars]
```

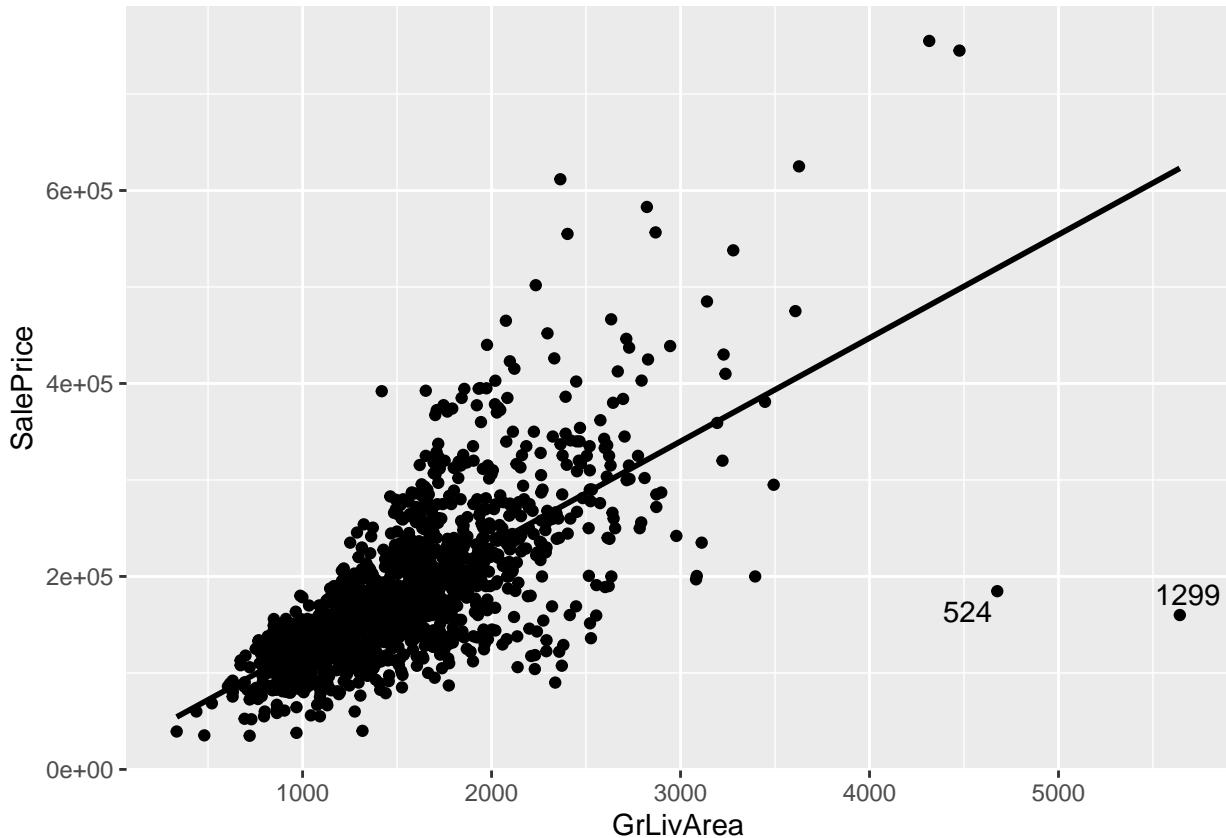
Outlier analysis

Detecting outliers

```
# we look for outliers in the most correlated variable with SalePrice which is GrLivArea.
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 4.0.3
```

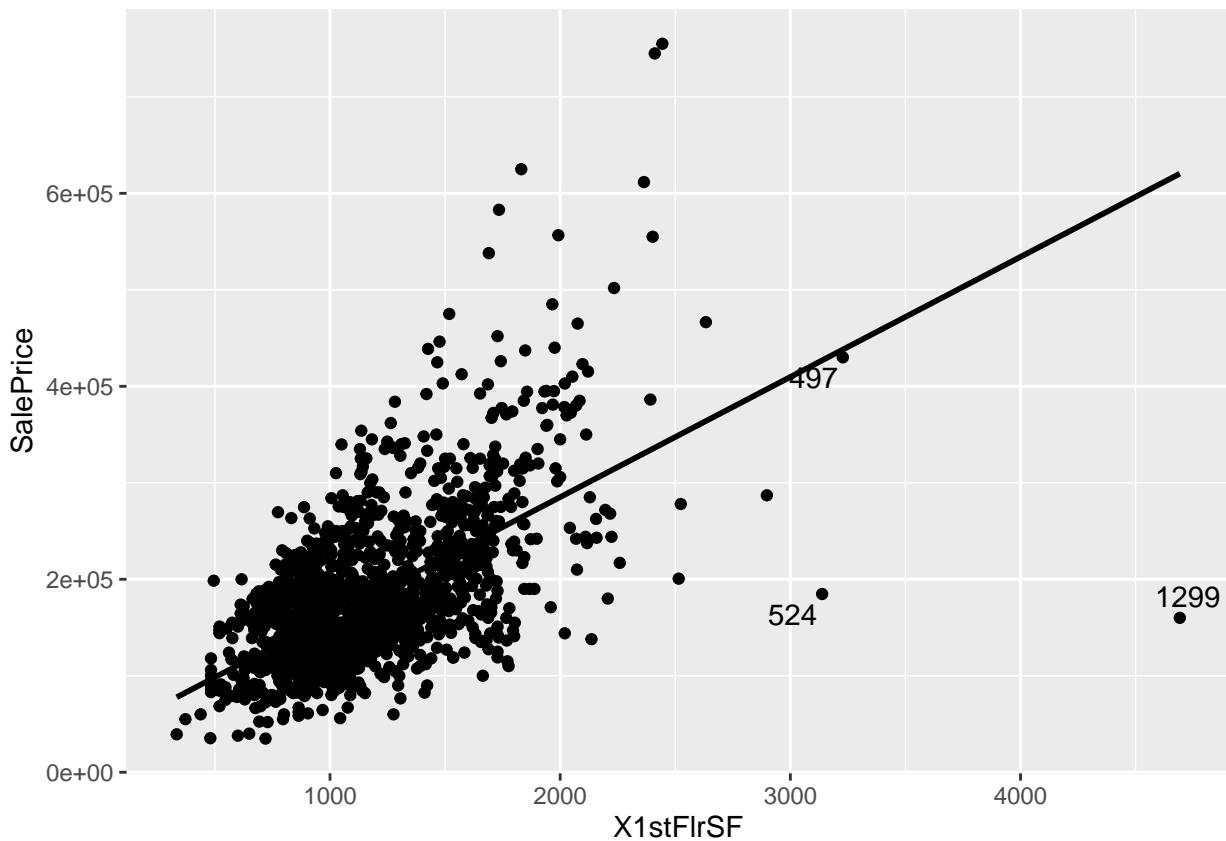
```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=GrLivArea , y = SalePrice)) + geom_point() + geom_smooth(method="loess") + geom_text_repel()
```



```
ggplot(combined[!is.na(combined$SalePrice),], aes(x=X1stFlrSF, y = SalePrice)) + geom_point() + geom_smooth(method=
```

Warning: Use of 'combined\$X1stFlrSF' is discouraged. Use 'X1stFlrSF' instead.

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



```

mod <- lm(SalePrice ~ ., data = combined)
summary(mod)

##
## Call:
## lm(formula = SalePrice ~ ., data = combined)
##
## Residuals:
##    Min     1Q Median     3Q    Max 
## -171545 -11140      0 11142 171545 
##
## Coefficients: (5 not defined because of singularities)
##                                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 -1.464e+06 2.202e+05 -6.649 4.42e-11 ***
## MSSubClass1 story 1945-       1.463e+02 5.029e+03  0.029 0.976804    
## MSSubClass1 story unf attic -1.788e+04 1.844e+04 -0.969 0.332637    
## MSSubClass1,5 story unf        4.020e+03 2.398e+04  0.168 0.866920    
## MSSubClass1,5 story fin        3.591e+03 9.301e+03  0.386 0.699532    
## MSSubClass2 story 1946+        6.490e+03 8.231e+03  0.788 0.430574    
## MSSubClass2 story 1945-       6.880e+03 8.861e+03  0.776 0.437675    
## MSSubClass2,5 story all ages -2.186e+04 1.640e+04 -1.333 0.182697    
## MSSubClasssplit/multi level   -9.647e+02 1.226e+04 -0.079 0.937289    
## MSSubClasssplit foyer          -4.387e+03 1.120e+04 -0.392 0.695304    
## MSSubClassduplex all style/age -5.137e+03 7.644e+03 -0.672 0.501731    
## MSSubClass1 story PUD 1946+    -2.136e+04 1.569e+04 -1.362 0.173444    
## MSSubClass2 story PUD 1946+    -1.592e+04 1.874e+04 -0.850 0.395698    
## MSSubClassPUD multilevel      -1.393e+04 2.080e+04 -0.670 0.503098    
## MSSubClass2 family conversion -3.997e+04 2.747e+04 -1.455 0.145963    
## MSZoningFV                     3.073e+04 1.292e+04  2.379 0.017488 *  
## MSZoningRH                     3.086e+04 1.285e+04  2.402 0.016448 *  
## MSZoningRL                     2.763e+04 1.102e+04  2.507 0.012317 *  
## MSZoningRM                     2.383e+04 1.036e+04  2.300 0.021612 *  
## LotFrontage                    8.897e+01 4.766e+01  1.867 0.062174 .  
## LotArea                         4.585e-01 9.774e-02  4.691 3.02e-06 ***  
## Street                          1.894e+04 1.243e+04  1.524 0.127653    
## AlleyNone                       -2.189e+03 4.502e+03 -0.486 0.626894    
## AlleyPave                      -2.115e+03 6.603e+03 -0.320 0.748780    
## LotShape                        5.863e+02 1.452e+03  0.404 0.686472    
## LandContourHLS                  1.269e+04 5.522e+03  2.298 0.021754 *  
## LandContourLow                  -2.786e+03 6.748e+03 -0.413 0.679738    
## LandContourLvl                  5.021e+03 3.952e+03  1.271 0.204084    
## LotConfigCulDSac                1.021e+04 3.683e+03  2.772 0.005660 **  
## LotConfigFR2                    -3.474e+03 4.308e+03 -0.806 0.420208    
## LotConfigFR3                    -2.034e+04 1.348e+04 -1.508 0.131763    
## LotConfigInside                 9.893e+01 1.942e+03  0.051 0.959375    
## LandSlope                       7.942e+02 3.702e+03  0.215 0.830185    
## NeighborhoodBlueste             2.047e+04 2.109e+04  0.971 0.331886    
## NeighborhoodBrDale              1.455e+04 1.239e+04  1.174 0.240450    
## NeighborhoodBrkSide              3.040e+03 1.004e+04  0.303 0.762103    
## NeighborhoodClearCr             -9.648e+03 9.918e+03 -0.973 0.330825    
## NeighborhoodCollgCr             -1.065e+04 7.735e+03 -1.377 0.168724    
## NeighborhoodCrawfor              1.245e+04 9.257e+03  1.345 0.178935    
## NeighborhoodEdwards              -1.217e+04 8.593e+03 -1.417 0.156831    
## NeighborhoodGilbert              -1.158e+04 8.236e+03 -1.406 0.160066    
## NeighborhoodIDOTRR                -3.573e+03 1.145e+04 -0.312 0.755102    
## NeighborhoodMeadowV              -8.099e+02 1.313e+04 -0.062 0.950838    
## NeighborhoodMitchel              -1.253e+04 8.705e+03 -1.439 0.150337    
## NeighborhoodNAmes                -9.956e+03 8.387e+03 -1.187 0.235414    
## NeighborhoodNoRidge               1.907e+04 9.037e+03  2.110 0.035024 *  
## NeighborhoodNPkVill               2.156e+04 1.487e+04  1.450 0.147207    
## NeighborhoodNridgHt                3.045e+04 7.828e+03  3.890 0.000106 *** 

```

## NeighborhoodNWAmes	-1.385e+04	8.586e+03	-1.614	0.106836
## NeighborhoodOldTown	-5.865e+03	1.033e+04	-0.568	0.570317
## NeighborhoodSawyer	-2.221e+03	8.717e+03	-0.255	0.798959
## NeighborhoodSawyerW	-4.694e+03	8.364e+03	-0.561	0.574790
## NeighborhoodSomerset	-4.056e+02	9.660e+03	-0.042	0.966517
## NeighborhoodStoneBr	4.048e+04	8.928e+03	4.534	6.35e-06 ***
## NeighborhoodSWISU	-5.210e+03	1.038e+04	-0.502	0.615984
## NeighborhoodTimber	-1.116e+04	8.709e+03	-1.282	0.200243
## NeighborhoodVeenker	1.320e+03	1.117e+04	0.118	0.905994
## Condition1Feedr	3.573e+03	5.305e+03	0.674	0.500691
## Condition1Norm	1.403e+04	4.399e+03	3.190	0.001460 **
## Condition1PosA	6.305e+03	1.068e+04	0.590	0.555215
## Condition1PosN	1.095e+04	7.895e+03	1.387	0.165639
## Condition1RRAe	-1.385e+04	9.307e+03	-1.488	0.136976
## Condition1RRAn	1.641e+04	7.338e+03	2.237	0.025487 *
## Condition1RRNe	-7.230e+03	1.879e+04	-0.385	0.700462
## Condition1RRNn	1.176e+04	1.359e+04	0.865	0.387095
## Condition2Feedr	-1.371e+04	2.680e+04	-0.512	0.609012
## Condition2Norm	-1.791e+04	2.378e+04	-0.753	0.451423
## Condition2PosA	3.662e+04	3.683e+04	0.994	0.320324
## Condition2PosN	-2.385e+05	3.095e+04	-7.706	2.66e-14 ***
## Condition2RRAe	-9.697e+04	7.594e+04	-1.277	0.201900
## Condition2RRAn	-1.228e+04	3.498e+04	-0.351	0.725556
## Condition2RRNn	-1.159e+04	3.037e+04	-0.382	0.702837
## BldgType2fmCon	3.002e+04	2.657e+04	1.130	0.258634
## BldgTypeDuplex	NA	NA	NA	NA
## BldgTypeTwnhs	-9.616e+03	1.671e+04	-0.576	0.564963
## BldgTypeTwnhsE	-2.688e+03	1.592e+04	-0.169	0.865949
## HouseStyle1.5Unf	9.961e+03	2.386e+04	0.417	0.676433
## HouseStyle1Story	1.603e+04	9.382e+03	1.709	0.087715 .
## HouseStyle2.5Fin	4.675e+03	1.803e+04	0.259	0.795474
## HouseStyle2.5Unf	9.736e+03	1.685e+04	0.578	0.563574
## HouseStyle2Story	-7.732e+03	8.646e+03	-0.894	0.371364
## HouseStyleSoyer	3.944e+03	1.250e+04	0.315	0.752445
## HouseStyleSLvl	9.322e+03	1.383e+04	0.674	0.500425
## OverallQual	8.662e+03	1.086e+03	7.976	3.43e-15 ***
## OverallCond	5.001e+03	9.163e+02	5.458	5.82e-08 ***
## YearBuilt	3.054e+02	8.725e+01	3.501	0.000481 ***
## YearRemodAdd	-1.548e+01	6.085e+01	-0.254	0.799194
## RoofStyleGable	-1.556e+02	1.945e+04	-0.008	0.993621
## RoofStyleGambrel	-1.960e+02	2.126e+04	-0.009	0.992645
## RoofStyleHip	3.849e+03	1.950e+04	0.197	0.843565
## RoofStyleMansard	9.428e+03	2.269e+04	0.416	0.677799
## RoofStyleShed	6.527e+04	4.013e+04	1.627	0.104070
## RoofMatlCompShg	6.703e+05	3.531e+04	18.983	< 2e-16 ***
## RoofMatlMembran	7.179e+05	4.817e+04	14.905	< 2e-16 ***
## RoofMatlMetal	6.950e+05	4.848e+04	14.335	< 2e-16 ***
## RoofMatlRoll	6.474e+05	4.448e+04	14.554	< 2e-16 ***
## RoofMatlTar&Grv	6.653e+05	4.032e+04	16.502	< 2e-16 ***
## RoofMatlWdShake	6.446e+05	3.876e+04	16.633	< 2e-16 ***
## RoofMatlWdShngl	7.474e+05	3.676e+04	20.331	< 2e-16 ***
## Exterior1stAsphShn	-2.127e+04	3.519e+04	-0.604	0.545685
## Exterior1stBrkComm	-8.461e+03	2.957e+04	-0.286	0.774825
## Exterior1stBrkFace	9.124e+03	1.352e+04	0.675	0.499803
## Exterior1stCBlock	5.726e+03	2.716e+04	0.211	0.833019
## Exterior1stCemntBd	7.069e+02	2.003e+04	0.035	0.971853
## Exterior1stHdBoard	-8.759e+03	1.363e+04	-0.643	0.520556
## Exterior1stImStucc	-4.031e+04	2.987e+04	-1.349	0.177485
## Exterior1stMetalSd	1.483e+03	1.548e+04	0.096	0.923722
## Exterior1stPlywood	-1.307e+04	1.350e+04	-0.968	0.333309
## Exterior1stStone	2.507e+03	2.545e+04	0.099	0.921540

## Exterior1stStucco	4.675e+02	1.480e+04	0.032	0.974811
## Exterior1stVinylSd	-8.154e+03	1.430e+04	-0.570	0.568745
## Exterior1stWd Sdng	-7.918e+03	1.311e+04	-0.604	0.545916
## Exterior1stWdShing	-1.160e+03	1.422e+04	-0.082	0.934988
## Exterior2ndAsphShn	7.105e+03	2.363e+04	0.301	0.763732
## Exterior2ndBrk Cmn	7.667e+03	2.183e+04	0.351	0.725455
## Exterior2ndBrkFace	-2.031e+03	1.383e+04	-0.147	0.883271
## Exterior2ndCBlock	NA	NA	NA	NA
## Exterior2ndCmentBd	7.463e+03	1.966e+04	0.380	0.704369
## Exterior2ndHdBoard	2.595e+03	1.296e+04	0.200	0.841361
## Exterior2ndImStucc	2.227e+04	1.510e+04	1.474	0.140669
## Exterior2ndMetalSd	-2.038e+02	1.500e+04	-0.014	0.989161
## Exterior2ndOther	-3.058e+04	2.912e+04	-1.050	0.293873
## Exterior2ndPlywood	3.249e+03	1.261e+04	0.258	0.796736
## Exterior2ndStone	-9.748e+03	1.806e+04	-0.540	0.589553
## Exterior2ndStucco	-6.368e+03	1.431e+04	-0.445	0.656306
## Exterior2ndVinylSd	5.350e+03	1.370e+04	0.390	0.696327
## Exterior2ndWd Sdng	4.970e+03	1.251e+04	0.397	0.691153
## Exterior2ndWd Shng	-3.373e+03	1.316e+04	-0.256	0.797777
## MasVnrTypeBrkFace	7.268e+03	7.119e+03	1.021	0.307485
## MasVnrTypeNone	1.424e+04	7.151e+03	1.992	0.046602 *
## MasVnrTypeStone	1.359e+04	7.572e+03	1.795	0.072833 .
## MasVnrArea	3.317e+01	6.072e+00	5.463	5.67e-08 ***
## ExterQual	5.161e+03	2.238e+03	2.306	0.021302 *
## ExterCond	-2.525e+03	2.248e+03	-1.123	0.261468
## FoundationCBlock	2.248e+03	3.365e+03	0.668	0.504178
## FoundationPConc	4.117e+03	3.643e+03	1.130	0.258637
## FoundationSlab	1.426e+04	9.681e+03	1.473	0.140929
## FoundationStone	6.633e+03	1.195e+04	0.555	0.578985
## FoundationWood	-2.555e+04	1.586e+04	-1.610	0.107574
## BsmtQual	2.752e+03	1.824e+03	1.509	0.131534
## BsmtCond	-4.608e+03	2.268e+03	-2.032	0.042414 *
## BsmtExposure	5.263e+03	8.991e+02	5.854	6.15e-09 ***
## BsmtFinType1	-9.471e+01	5.397e+02	-0.175	0.860729
## BsmtFinSF1	3.547e+01	5.077e+00	6.987	4.60e-12 ***
## BsmtFinType2	1.062e+02	1.363e+03	0.078	0.937919
## BsmtFinSF2	2.157e+01	8.466e+00	2.548	0.010945 *
## BsmtUnfSF	1.482e+01	4.586e+00	3.230	0.001269 **
## HeatingGasA	-1.267e+03	2.682e+04	-0.047	0.962336
## HeatingGasW	-4.472e+03	2.763e+04	-0.162	0.871426
## HeatingGrav	9.331e+02	2.882e+04	0.032	0.974178
## HeatingOthW	-4.114e+04	3.305e+04	-1.245	0.213382
## HeatingWall	4.010e+03	3.114e+04	0.129	0.897555
## HeatingQC	1.039e+03	1.021e+03	1.018	0.308974
## CentralAir	-3.401e+02	4.132e+03	-0.082	0.934418
## ElectricalFuseF	-2.601e+03	6.131e+03	-0.424	0.671456
## ElectricalFuseP	3.687e+03	1.771e+04	0.208	0.835115
## ElectricalMix	1.224e+04	2.764e+04	0.443	0.658040
## ElectricalSBrkr	-3.426e+03	3.168e+03	-1.081	0.279710
## X1stFlrSF	4.918e+01	5.651e+00	8.702	< 2e-16 ***
## X2ndFlrSF	6.521e+01	5.974e+00	10.916	< 2e-16 ***
## LowQualFinSF	2.407e+01	1.979e+01	1.216	0.224252
## GrLivArea	NA	NA	NA	NA
## BsmtFullBath	1.516e+03	2.085e+03	0.727	0.467210
## BsmtHalfBath	3.817e+02	3.195e+03	0.119	0.904938
## FullBath	3.128e+03	2.332e+03	1.341	0.180162
## HalfBath	2.945e+03	2.225e+03	1.324	0.185852
## BedroomAbvGr	-6.328e+03	1.465e+03	-4.320	1.68e-05 ***
## KitchenAbvGr	-1.446e+04	6.503e+03	-2.224	0.026321 *
## KitchenQual	6.907e+03	1.741e+03	3.968	7.67e-05 ***
## TotRmsAbvGrd	2.828e+03	1.010e+03	2.800	0.005188 **

## Functional	6.108e+03	1.204e+03	5.073	4.52e-07	***
## Fireplaces	5.792e+03	2.332e+03	2.483	0.013150	*
## FireplaceQu	-1.740e+03	8.458e+02	-2.057	0.039922	*
## GarageTypeAttchd	1.169e+04	1.162e+04	1.007	0.314299	
## GarageTypeBasment	1.708e+04	1.355e+04	1.260	0.207759	
## GarageTypeBuiltIn	1.268e+04	1.216e+04	1.043	0.297215	
## GarageTypeCarPort	1.587e+04	1.517e+04	1.047	0.295505	
## GarageTypeDetchd	1.709e+04	1.158e+04	1.476	0.140315	
## GarageTypeNone	4.098e+04	1.630e+04	2.514	0.012053	*
## GarageFinish	2.661e+02	1.291e+03	0.206	0.836677	
## GarageCars	7.823e+03	1.702e+03	4.596	4.74e-06	***
## GarageQual	6.741e+03	3.982e+03	1.693	0.090692	.
## GarageCond	-1.446e+03	4.219e+03	-0.343	0.731824	
## PavedDrive	-2.067e+02	1.758e+03	-0.118	0.906408	
## WoodDeckSF	1.107e+01	6.189e+00	1.788	0.074032	.
## OpenPorchSF	8.935e+00	1.223e+01	0.731	0.465206	
## EnclosedPorch	-4.583e-01	1.325e+01	-0.035	0.972409	
## X3SsnPorch	3.464e+01	2.394e+01	1.447	0.148136	
## ScreenPorch	4.401e+01	1.308e+01	3.365	0.000789	***
## PoolArea	-5.134e+01	5.820e+01	-0.882	0.377873	
## PoolQC	2.578e+04	8.932e+03	2.887	0.003962	**
## FenceGdWo	9.896e+03	5.197e+03	1.904	0.057095	.
## FenceMnPrv	1.192e+04	4.248e+03	2.805	0.005111	**
## FenceMnWw	4.704e+03	8.701e+03	0.541	0.588834	
## FenceNone	9.503e+03	3.855e+03	2.465	0.013833	*
## MiscFeatureNone	8.661e+03	1.051e+05	0.082	0.934329	
## MiscFeatureOthr	3.623e+04	9.510e+04	0.381	0.703271	
## MiscFeatureShed	1.029e+04	1.006e+05	0.102	0.918532	
## MiscFeatureTenC	-6.300e+04	9.832e+04	-0.641	0.521818	
## MiscVal	2.425e-01	6.596e+00	0.037	0.970681	
## MoSold2	-1.152e+04	4.985e+03	-2.312	0.020954	*
## MoSold3	-6.868e+03	4.333e+03	-1.585	0.113227	
## MoSold4	-5.717e+03	4.164e+03	-1.373	0.169940	
## MoSold5	-3.687e+03	3.947e+03	-0.934	0.350409	
## MoSold6	-6.938e+03	3.889e+03	-1.784	0.074637	.
## MoSold7	-5.202e+03	3.953e+03	-1.316	0.188441	
## MoSold8	-1.055e+04	4.214e+03	-2.503	0.012449	*
## MoSold9	-8.938e+03	4.838e+03	-1.848	0.064902	.
## MoSold10	-1.524e+04	4.526e+03	-3.369	0.000779	***
## MoSold11	-8.383e+03	4.590e+03	-1.826	0.068037	.
## MoSold12	-9.555e+03	4.913e+03	-1.945	0.052023	.
## YrSold2007	-1.712e+03	2.098e+03	-0.816	0.414765	
## YrSold2008	6.630e+02	2.182e+03	0.304	0.761317	
## YrSold2009	-1.097e+03	2.100e+03	-0.523	0.601368	
## YrSold2010	-3.274e+02	2.613e+03	-0.125	0.900283	
## SaleTypeCon	2.841e+04	1.882e+04	1.509	0.131464	
## SaleTypeConLD	1.864e+04	1.042e+04	1.789	0.073904	.
## SaleTypeConLI	6.878e+03	1.238e+04	0.555	0.578749	
## SaleTypeConLw	2.404e+03	1.272e+04	0.189	0.850082	
## SaleTypeCWD	1.803e+04	1.370e+04	1.316	0.188501	
## SaleTypeNew	4.111e+04	1.636e+04	2.512	0.012115	*
## SaleTypeOth	1.023e+04	1.560e+04	0.656	0.512209	
## SaleTypeWD	1.885e+03	4.452e+03	0.423	0.672013	
## SaleConditionAdjLand	1.302e+04	1.536e+04	0.848	0.396788	
## SaleConditionAlloca	7.372e+03	9.328e+03	0.790	0.429484	
## SaleConditionFamily	-4.761e+01	6.513e+03	-0.007	0.994169	
## SaleConditionNormal	7.621e+03	3.082e+03	2.473	0.013538	*
## SaleConditionPartial	-1.349e+04	1.574e+04	-0.857	0.391340	
## TotBathrooms	NA	NA	NA	NA	
## Remod	3.629e+03	1.840e+03	1.972	0.048841	*
## Age	NA	NA	NA	NA	

```

## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24350 on 1234 degrees of freedom
## Multiple R-squared: 0.9206, Adjusted R-squared: 0.9061
## F-statistic: 63.56 on 225 and 1234 DF, p-value: < 2.2e-16

cooksdsd <- cooks.distance(mod)

# All outliers
influential <- as.numeric(names(cooksdsd)[(cooksdsd > 10*mean(cooksdsd, na.rm=T))]) # influential row numbers
influential <- na.omit(influential)
influential

## [1] 10 89 94 186 198 524 589 692 770 826 1045 1183 1187 1268 1444
## attr(,"na.action")
## [1] 4 7 8 9 11 13 16 18 19 23 24 26 27 28 29 30 31
## attr(,"class")
## [1] "omit"

```

Deleting outliers

```
combined <- combined[-influential,]
```

Preparing data for modelling

Methods used in: <https://topepo.github.io/caret/pre-processing.html>

```

df <- df[, !names(df) %in% "Id"]
true_num_vars <- names(Filter(is.numeric, df))
num_vars <- names(Filter(is.numeric, combined)) #index vector numeric variables
factor_vars <- names(Filter(is.factor, combined))
cat('numeric variables: ', length(num_vars), ' and categorical variables:', length(factor_vars), '\n')

## numeric variables: 54 and categorical variables: 25

```

Standardizing numerical predictor variables

```

library(caret)

## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice

##
## Attaching package: 'caret'

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

```

```

library(DescTools)

## Warning: package 'DescTools' was built under R version 4.0.3

##
## Attaching package: 'DescTools'

## The following objects are masked from 'package:caret':
## 
##     MAE, RMSE

nvarnames <- true_num_vars[!(true_num_vars %in% c('MSSubClass', 'MoSold', 'YrSold', 'SalePrice', 'OverallQual',
num_df <- combined[, names(combined) %in% nvarnames]
factor_df <- combined[, !(names(combined) %in% nvarnames)]
factor_df <- factor_df[, names(factor_df) != 'SalePrice']

cat('There are', length(num_df), 'numeric variables, and', length(factor_df), 'factor variables')

## There are 28 numeric variables, and 50 factor variables

for(i in 1:ncol(num_df)){
  if (abs(Skew(num_df[,i]))>0.8){
    num_df[,i] <- log(num_df[,i] +1)
  }
}

Penum <- preprocess(num_df, method=c("center", "scale"))
print(Penum)

## Created from 1445 samples and 28 variables
##
## Pre-processing:
##   - centered (28)
##   - ignored (0)
##   - scaled (28)

df_norm <- predict(Penum, num_df)
dim(df_norm)

## [1] 1445 28

DFdummies <- as.data.frame(model.matrix(~.-1, factor_df))
dim(DFdummies)

## [1] 1445 203

ZerocolTest <- which(colSums(DFdummies[1:nrow(combined[!is.na(combined$SalePrice),])],)==0)
colnames(DFdummies[ZerocolTest])

## [1] "Condition2PosN"

DFdummies <- DFdummies[,-ZerocolTest] #removing predictors
fewOnes <- which(colSums(DFdummies[1:nrow(combined[!is.na(combined$SalePrice),])],)<10)
colnames(DFdummies[fewOnes])

```

```

## [1] "MSSubClass1 story unf attic" "LotConfigFR3"
## [3] "NeighborhoodBlueste"          "NeighborhoodNPkVill"
## [5] "Condition1PosA"              "Condition1RRNe"
## [7] "Condition1RRNn"               "Condition2Feedr"
## [9] "Condition2PosA"              "Condition2RRAe"
## [11] "Condition2RRAn"              "Condition2RRNn"
## [13] "HouseStyle2.5Fin"            "RoofStyleMansard"
## [15] "RoofStyleShed"               "RoofMatlMembran"
## [17] "RoofMatlMetal"                "RoofMatlRoll"
## [19] "RoofMatlWdShake"              "RoofMatlWdShngl"
## [21] "Exterior1stAsphShn"           "Exterior1stBrkComm"
## [23] "Exterior1stCBlock"             "Exterior1stImStucc"
## [25] "Exterior1stStone"              "Exterior2ndAsphShn"
## [27] "Exterior2ndBrk Cmn"            "Exterior2ndCBlock"
## [29] "Exterior2ndImStucc"             "Exterior2ndOther"
## [31] "Exterior2ndStone"              "FoundationStone"
## [33] "FoundationWood"                "HeatingGrav"
## [35] "HeatingOthW"                  "HeatingWall"
## [37] "ElectricalFuseP"                "ElectricalMix"
## [39] "GarageTypeCarPort"              "MiscFeature0thr"
## [41] "MiscFeatureTenC"                 "SaleTypeCon"
## [43] "SaleTypeConLD"                  "SaleTypeConLI"
## [45] "SaleTypeConLw"                  "SaleTypeCWD"
## [47] "SaleType0th"                    "SaleConditionAdjLand"
## [49] "Age"

```

```

DFdummies <- DFdummies[, -fewOnes] #removing predictors
dim(DFdummies)

```

```

## [1] 1445 153

```

```

final <- cbind(num_df, DFdummies)

```

Transforming SalePrice variable

```

Skew(combined$SalePrice)

```

```

## [1] 1.536502

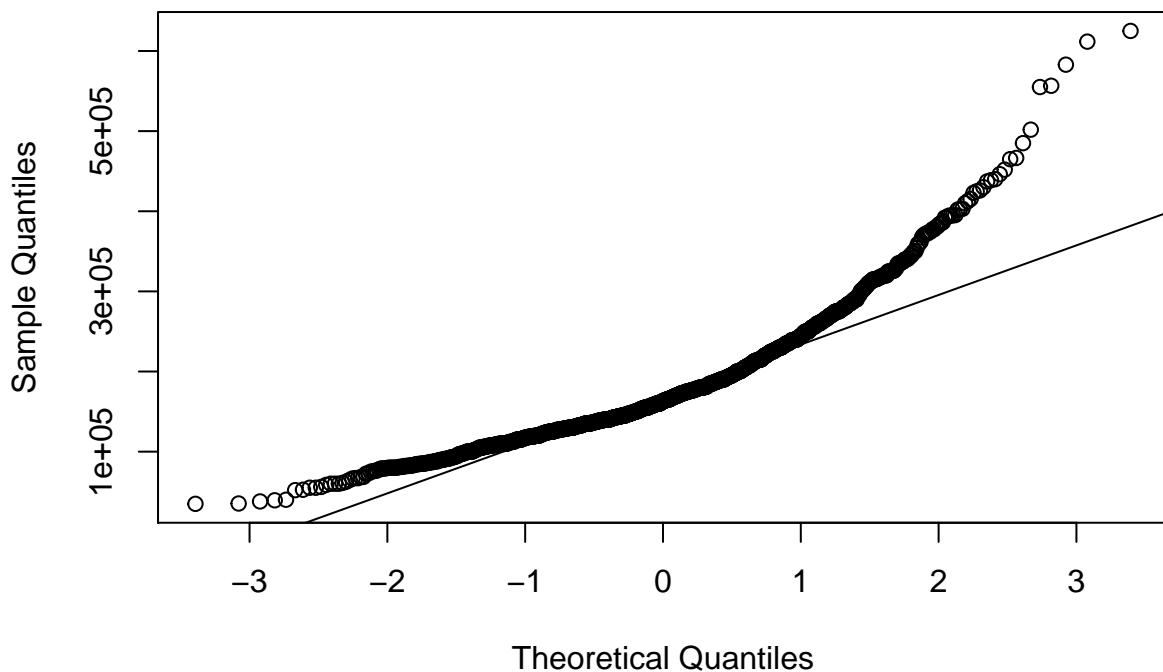
```

```

qqnorm(combined$SalePrice)
qqline(combined$SalePrice)

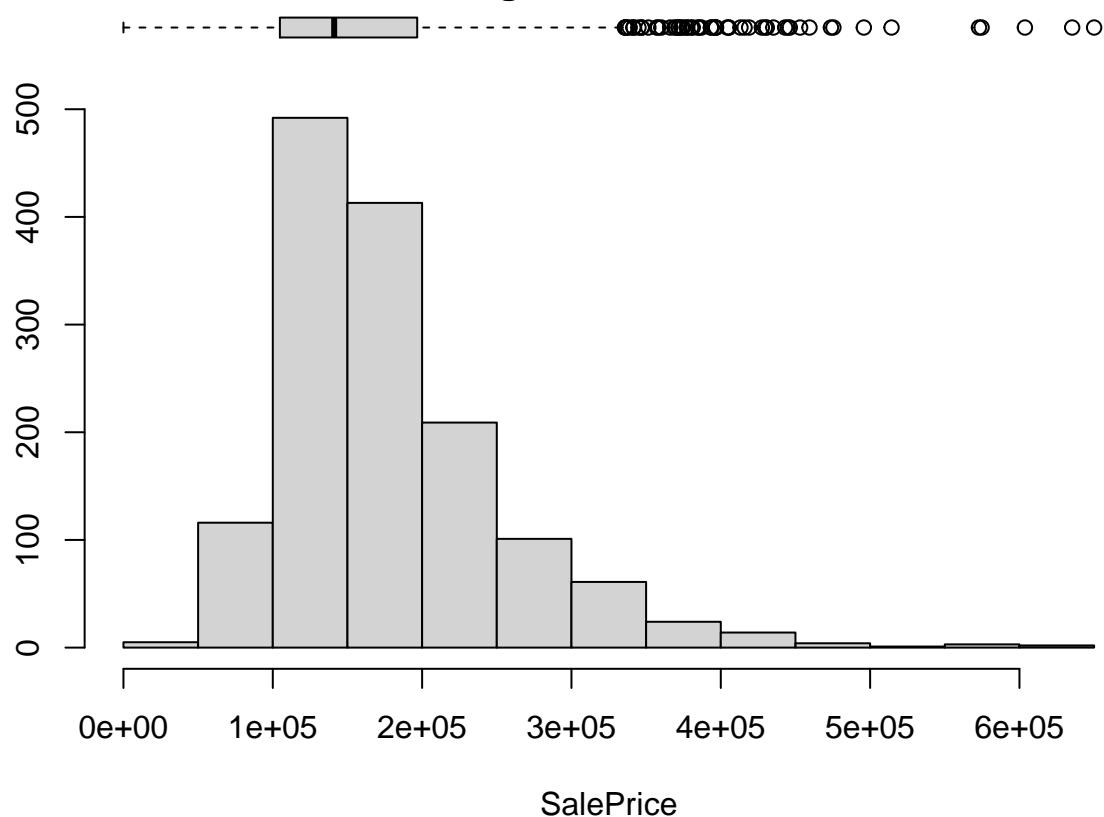
```

Normal Q-Q Plot



```
layout(mat = matrix(c(1,2),2,1, byrow=TRUE), height = c(1,8))
par(mar=c(0, 3.1, 1.1, 2.1))
boxplot(combined$SalePrice , horizontal=TRUE , xaxt="n", frame=F, main=sprintf('Histogram of SalePrice'))
par(mar=c(4, 3.1, 1.1, 2.1))
hist(combined$SalePrice,main=' ', xlab = "SalePrice", ylab = "count")
```

Histogram of SalePrice



```
combined$SalePrice <- log(combined$SalePrice)
```

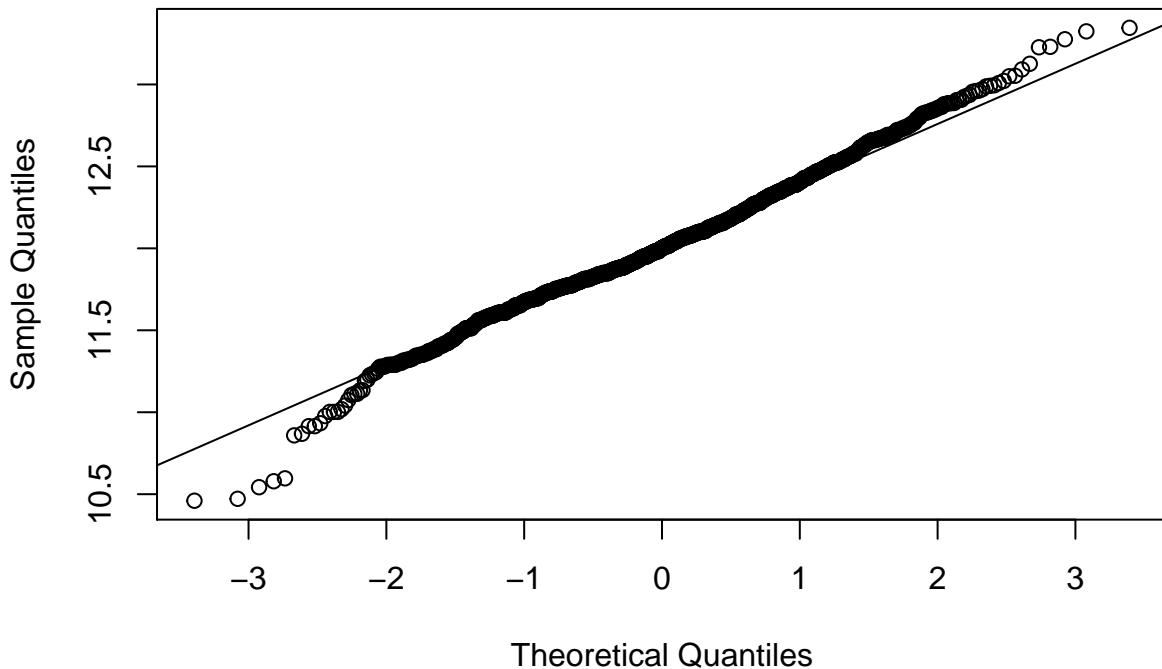
```
Skew(combined$SalePrice)
```

```
## [1] 0.04044969
```

```
qqnorm(combined$SalePrice)
```

```
qqline(combined$SalePrice)
```

Normal Q-Q Plot



```
layout(mat = matrix(c(1,2),2,1, byrow=TRUE), height = c(1,8))
```

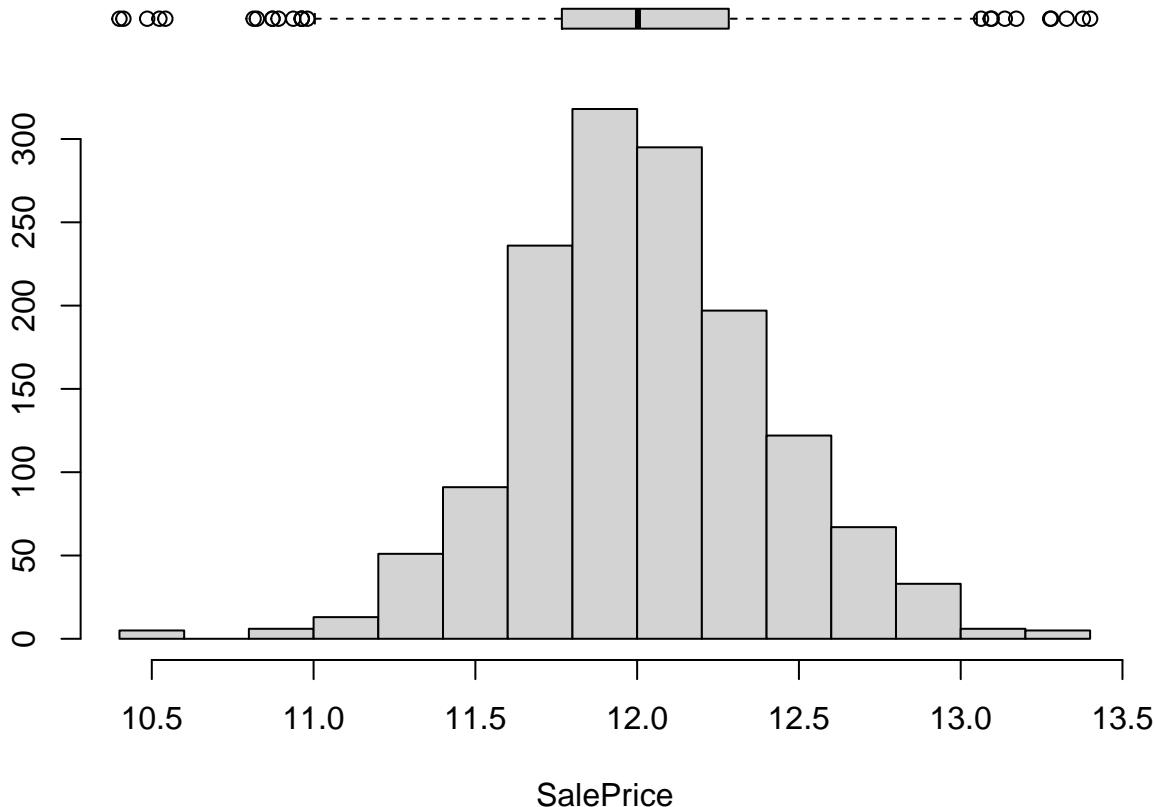
```
par(mar=c(0, 3.1, 1.1, 2.1))
```

```
boxplot(combined$SalePrice , horizontal=TRUE , xaxt="n", frame=F, main=sprintf('Histogram of SalePrice'))
```

```
par(mar=c(4, 3.1, 1.1, 2.1))
```

```
hist(combined$SalePrice,main=' ', xlab = "SalePrice", ylab = "count")
```

Histogram of SalePrice



```
final_df <- final
final_df$SalePrice <- combined$SalePrice
```

Train and test data

```
library(caret)
set.seed(1)
train_rows <- createDataPartition(y=final_df[, 'SalePrice'], list=FALSE, p=.8)
train_dummy <- final_df[train_rows,]
test_dummy <- final_df[-train_rows,]
stopifnot(nrow(train_dummy) + nrow(test_dummy) == nrow(final_df))
write.csv(train_dummy, "../data/processed/train_data_with_dummy.csv")
write.csv(test_dummy, "../data/processed/test_data_with_dummy.csv")

train_rows <- createDataPartition(y=combined[, 'SalePrice'], list=FALSE, p=.8)
train <- combined[train_rows,]
test <- combined[-train_rows,]
stopifnot(nrow(train) + nrow(test) == nrow(combined))
write.csv(train, "../data/processed/train_data.csv")
write.csv(test, "../data/processed/test_data.csv")
```

Modelling

Ridge regression

```
library(caret)
train_dummy <- read.table("../data/processed/train_data_with_dummy.csv", sep = ",", header = T)
head(train_dummy)
```

```

##   X LotFrontage LotArea YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1
## 1 1    4.189655 9.042040    2003      2003  5.283204  6.561031
## 2 2    4.394449 9.169623    1976      1976  0.000000  6.886532
## 3 3    4.234107 9.328212    2001      2002  5.093750  6.188264
## 4 4    4.110874 9.164401    1915      1970  0.000000  5.379897
## 5 6    4.454347 9.555064    1993      1995  0.000000  6.597146
## 6 8    4.330733 9.247925    1973      1973  5.484797  6.756932
##   BsmtFinSF2 BsmtUnfSF X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath
## 1 0.000000  5.017280  6.753438     854        0 7.444833       1
## 2 0.000000  5.652489  7.141245      0        0 7.141245       0
## 3 0.000000  6.075346  6.825460     866        0 7.488294       1
## 4 0.000000  6.293419  6.869014     756        0 7.448916       1
## 5 0.000000  4.174387  6.680855     566        0 7.217443       1
## 6 3.496508  5.379897  7.010312     983        0 7.645398       1
##   BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd
## 1 0.0000000      2      1      3  0.6931472       8
## 2 0.6931472      2      0      3  0.6931472       6
## 3 0.0000000      2      1      3  0.6931472       6
## 4 0.0000000      1      0      3  0.6931472       7
## 5 0.0000000      1      1      1  0.6931472       5
## 6 0.0000000      2      1      3  0.6931472       7
##   Fireplaces GarageCars WoodDeckSF OpenPorchSF EnclosedPorch X3SsnPorch
## 1 0            2 0.000000  4.127134  0.000000  0.000000
## 2 1            2 5.700444  0.000000  0.000000  0.000000
## 3 1            2 0.000000  3.761200  0.000000  0.000000
## 4 1            3 0.000000  3.583519  5.609472  0.000000
## 5 0            2 3.713572  3.433987  0.000000  5.771441
## 6 2            2 5.463832  5.323010  5.433722  0.000000
##   ScreenPorch PoolArea MiscVal MSSubClass1.story.1946. MSSubClass1.story.1945.
## 1 0            0 0.000000          0          0
## 2 0            0 0.000000          1          0
## 3 0            0 0.000000          0          0
## 4 0            0 0.000000          0          0
## 5 0            0 6.552508          0          0
## 6 0            0 5.860786          0          0
##   MSSubClass1.5.story.unf MSSubClass1.5.story.fin MSSubClass2.story.1946.
## 1 0            0          0          1
## 2 0            0          0          0
## 3 0            0          0          1
## 4 0            0          0          0
## 5 0            0          1          0
## 6 0            0          0          1
##   MSSubClass2.story.1945. MSSubClass2.5.story.all.ages
## 1 0            0          0
## 2 0            0          0
## 3 0            0          0
## 4 1            1          0
## 5 0            0          0
## 6 0            0          0
##   MSSubClasssplit.multi.level MSSubClasssplit.foyer
## 1 0            0          0
## 2 0            0          0
## 3 0            0          0
## 4 0            0          0
## 5 0            0          0
## 6 0            0          0
##   MSSubClassduplex.all.style.age MSSubClass1.story.PUD.1946.
## 1 0            0          0
## 2 0            0          0
## 3 0            0          0
## 4 0            0          0

```

```

## 5          0          0
## 6          0          0
##   MSSubClass2.story.PUD.1946. MSSubClassPUD.multilevel
## 1          0          0
## 2          0          0
## 3          0          0
## 4          0          0
## 5          0          0
## 6          0          0
##   MSSubClass2.family.conversion MSZoningFV MSZoningRH MSZoningRL MSZoningRM
## 1          0          0          0          1          0
## 2          0          0          0          1          0
## 3          0          0          0          1          0
## 4          0          0          0          1          0
## 5          0          0          0          1          0
## 6          0          0          0          1          0
##   Street AlleyNone AlleyPave LotShape LandContourHLS LandContourLow
## 1     1       1       0       3       0       0
## 2     1       1       0       3       0       0
## 3     1       1       0       2       0       0
## 4     1       1       0       2       0       0
## 5     1       1       0       2       0       0
## 6     1       1       0       2       0       0
##   LandContourLvl LotConfigCulDSac LotConfigFR2 LotConfigInside LandSlope
## 1     1       0       0       1       2
## 2     1       0       1       0       2
## 3     1       0       0       1       2
## 4     1       0       0       0       2
## 5     1       0       0       1       2
## 6     1       0       0       0       2
##   NeighborhoodBrDale NeighborhoodBrkSide NeighborhoodClearCr
## 1     0       0       0
## 2     0       0       0
## 3     0       0       0
## 4     0       0       0
## 5     0       0       0
## 6     0       0       0
##   NeighborhoodCollgCr NeighborhoodCrawfor NeighborhoodEdwards
## 1     1       0       0
## 2     0       0       0
## 3     1       0       0
## 4     0       1       0
## 5     0       0       0
## 6     0       0       0
##   NeighborhoodGilbert NeighborhoodIDOTRR NeighborhoodMeadowV
## 1     0       0       0
## 2     0       0       0
## 3     0       0       0
## 4     0       0       0
## 5     0       0       0
## 6     0       0       0
##   NeighborhoodMitchel NeighborhoodNAmes NeighborhoodNoRidge NeighborhoodNridgHt
## 1     0       0       0       0
## 2     0       0       0       0
## 3     0       0       0       0
## 4     0       0       0       0
## 5     1       0       0       0
## 6     0       0       0       0
##   NeighborhoodNWAmes NeighborhoodOldTown NeighborhoodSawyer NeighborhoodSawyerW
## 1     0       0       0       0
## 2     0       0       0       0

```

```

## 3          0          0          0          0
## 4          0          0          0          0
## 5          0          0          0          0
## 6          1          0          0          0
## NeighborhoodSomerst NeighborhoodStoneBr NeighborhoodSWISU NeighborhoodTimber
## 1          0          0          0          0
## 2          0          0          0          0
## 3          0          0          0          0
## 4          0          0          0          0
## 5          0          0          0          0
## 6          0          0          0          0
## NeighborhoodVeenerk Condition1Feedr Condition1Norm Condition1PosN
## 1          0          0          1          0
## 2          1          1          0          0
## 3          0          0          1          0
## 4          0          0          1          0
## 5          0          0          1          0
## 6          0          0          0          1
## Condition1RRAe Condition1RRAn Condition2Norm BldgType2fmCon BldgTypeDuplex
## 1          0          0          1          0          0
## 2          0          0          1          0          0
## 3          0          0          1          0          0
## 4          0          0          1          0          0
## 5          0          0          1          0          0
## 6          0          0          1          0          0
## BldgTypeTwnhs BldgTypeTwnhsE HouseStyle1.5Unf HouseStyle1Story
## 1          0          0          0          0
## 2          0          0          0          1
## 3          0          0          0          0
## 4          0          0          0          0
## 5          0          0          0          0
## 6          0          0          0          0
## HouseStyle2.5Unf HouseStyle2Story HouseStyleSFoyer HouseStyleSLvl OverallQual
## 1          0          1          0          0          7
## 2          0          0          0          0          6
## 3          0          1          0          0          7
## 4          0          1          0          0          7
## 5          0          0          0          0          5
## 6          0          1          0          0          7
## OverallCond RoofStyleGable RoofStyleGambrel RoofStyleHip RoofMatlCompShg
## 1          5          1          0          0          1
## 2          8          1          0          0          1
## 3          5          1          0          0          1
## 4          5          1          0          0          1
## 5          5          1          0          0          1
## 6          6          1          0          0          1
## RoofMatlTar.Grv Exterior1stBrkFace Exterior1stCemntBd Exterior1stHdBoard
## 1          0          0          0          0
## 2          0          0          0          0
## 3          0          0          0          0
## 4          0          0          0          0
## 5          0          0          0          0
## 6          0          0          0          1
## Exterior1stMetalSd Exterior1stPlywood Exterior1stStucco Exterior1stVinylSd
## 1          0          0          0          1
## 2          1          0          0          0
## 3          0          0          0          1
## 4          0          0          0          0
## 5          0          0          0          1
## 6          0          0          0          0
## Exterior1stWd.Sdng Exterior1stWdShing Exterior2ndBrkFace Exterior2ndCmentBd

```

## 1	0	0	0	0	0	
## 2	0	0	0	0	0	
## 3	0	0	0	0	0	
## 4	1	0	0	0	0	
## 5	0	0	0	0	0	
## 6	0	0	0	0	0	
## Exterior2ndHdBoard	Exterior2ndMetalSd	Exterior2ndPlywood	Exterior2ndStucco			
## 1	0	0	0	0	0	
## 2	0	1	0	0	0	
## 3	0	0	0	0	0	
## 4	0	0	0	0	0	
## 5	0	0	0	0	0	
## 6	1	0	0	0	0	
## Exterior2ndVinylSd	Exterior2ndWd.Sdng	Exterior2ndWd.Shng	MasVnrTypeBrkFace			
## 1	1	0	0	1		
## 2	0	0	0	0	0	
## 3	1	0	0	1		
## 4	0	0	1	0		
## 5	1	0	0	0	0	
## 6	0	0	0	0	0	
## MasVnrTypeNone	MasVnrTypeStone	ExterQual	ExterCond	FoundationCBlock		
## 1	0	0	4	3	0	
## 2	1	0	3	3	1	
## 3	0	0	4	3	0	
## 4	1	0	3	3	0	
## 5	1	0	3	3	0	
## 6	0	1	3	3	1	
## FoundationPConc	FoundationSlab	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	
## 1	1	0	4	3	1	
## 2	0	0	4	3	4	
## 3	1	0	4	3	2	
## 4	0	0	3	4	1	
## 5	0	0	4	3	1	
## 6	0	0	4	3	2	
## BsmtFinType2	HeatingGasA	HeatingGasW	HeatingQC	CentralAir	ElectricalFuseF	
## 1	1	1	0	5	1	
## 2	1	1	0	5	1	
## 3	1	1	0	5	1	
## 4	1	1	0	4	1	
## 5	1	1	0	5	1	
## 6	4	1	0	5	1	
## ElectricalSBrkr	KitchenQual	Functional	FireplaceQu	GarageTypeAttchd		
## 1	1	4	7	0	1	
## 2	1	3	7	3	1	
## 3	1	4	7	3	1	
## 4	1	4	7	4	0	
## 5	1	3	7	0	1	
## 6	1	3	7	3	1	
## GarageTypeBasment	GarageTypeBuiltIn	GarageTypeDetchd	GarageTypeNone			
## 1	0	0	0	0	0	
## 2	0	0	0	0	0	
## 3	0	0	0	0	0	
## 4	0	0	1	0	0	
## 5	0	0	0	0	0	
## 6	0	0	0	0	0	
## GarageFinish	GarageQual	GarageCond	PavedDrive	PoolQC	FenceGdWo	FenceMnPrv
## 1	2	3	3	2	0	0
## 2	2	3	3	2	0	0
## 3	2	3	3	2	0	0
## 4	1	3	3	2	0	0
## 5	1	3	3	2	0	1

```

## 6          2          3          3          2          0          0          0
##   FenceMnWw FenceNone MiscFeatureNone MiscFeatureShed MoSold2 MoSold3 MoSold4
## 1          0          1          1          0          1          0          0
## 2          0          1          1          0          0          0          0
## 3          0          1          1          0          0          0          0
## 4          0          1          1          0          1          0          0
## 5          0          0          0          1          0          0          0
## 6          0          1          0          1          0          0          0
##   MoSold5 MoSold6 MoSold7 MoSold8 MoSold9 MoSold10 MoSold11 MoSold12 YrSold2007
## 1          0          0          0          0          0          0          0          0
## 2          1          0          0          0          0          0          0          1
## 3          0          0          0          0          1          0          0          0
## 4          0          0          0          0          0          0          0          0
## 5          0          0          0          0          0          1          0          0
## 6          0          0          0          0          0          0          1          0
##   YrSold2008 YrSold2009 YrSold2010 SaleTypeNew SaleTypeWD SaleConditionAlloc
## 1          1          0          0          0          1          0
## 2          0          0          0          0          1          0
## 3          1          0          0          0          1          0
## 4          0          0          0          0          1          0
## 5          0          1          0          0          1          0
## 6          0          1          0          0          1          0
##   SaleConditionFamily SaleConditionNormal SaleConditionPartial TotBathrooms
## 1          0          1          0          3.5
## 2          0          1          0          2.5
## 3          0          1          0          3.5
## 4          0          0          0          2.0
## 5          0          1          0          2.5
## 6          0          1          0          3.5
##   Remod SalePrice
## 1          0    12.24769
## 2          0    12.10901
## 3          1    12.31717
## 4          1    11.84940
## 5          1    11.87060
## 6          0    12.20607

```

```

test_dummy <- read.table("../data/processed/test_data_with_dummy.csv", sep = ",", header = T)
head(test_dummy)

```

```

##   X LotFrontage LotArea YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1
## 1 5    4.442651 9.565284    2000      2000  5.860786  6.486161
## 2 7    4.330733 9.218804    2004      2005  5.231109  7.222566
## 3 21   4.624973 9.562123    2005      2006  5.942799  0.000000
## 4 24   3.806662 8.348775    1976      1976  0.000000  6.734592
## 5 30   4.110874 8.752265    1927      1950  0.000000  0.000000
## 6 31   3.931826 9.047939    1920      1950  0.000000  0.000000
##   BsmtFinSF2 BsmtUnfSF X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath
## 1          0  6.196444  7.044033     1053          0  7.695758          1
## 2          0  5.762051  7.435438       0          0  7.435438          1
## 3          0  7.055313  7.055313     1218          0  7.773594          0
## 4          0  5.303305  6.966967       0          0  6.966967          1
## 5          0  6.255750  6.255750       0          0  6.255750          0
## 6          0  6.476972  6.476972     668          0  7.183871          0
##   BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd
## 1          0        2        1        4    0.6931472         9
## 2          0        2        0        3    0.6931472         7
## 3          0        3        1        4    0.6931472         9
## 4          0        1        0        3    0.6931472         6
## 5          0        1        0        1    0.6931472         4
## 6          0        1        0        3    0.6931472         6

```

```

##  Fireplaces GarageCars WoodDeckSF OpenPorchSF EnclosedPorch X3SsnPorch
## 1      1      3  5.262690   4.442651  0.000000    0
## 2      1      2  5.545177   4.060443  0.000000    0
## 3      1      3  5.484797   5.043425  0.000000    0
## 4      1      2  4.615121   4.709530  0.000000    0
## 5      0      1  3.912023   0.000000  4.477337    0
## 6      0      1  0.000000   4.007333  5.153292    0
##  ScreenPorch PoolArea MiscVal MSSubClass1.story.1946. MSSubClass1.story.1945.
## 1      0      0      0          0          0
## 2      0      0      0          1          0
## 3      0      0      0          0          0
## 4      0      0      0          0          0
## 5      0      0      0          0          1
## 6      0      0      0          0          0
##  MSSubClass1.5.story.unf MSSubClass1.5.story.fin MSSubClass2.story.1946.
## 1          0          0          1
## 2          0          0          0
## 3          0          0          1
## 4          0          0          0
## 5          0          0          0
## 6          0          0          0
##  MSSubClass2.story.1945. MSSubClass2.5.story.all.ages
## 1          0          0
## 2          0          0
## 3          0          0
## 4          0          0
## 5          0          0
## 6          1          0
##  MSSubClasssplit.multi.level MSSubClasssplit.foyer
## 1          0          0
## 2          0          0
## 3          0          0
## 4          0          0
## 5          0          0
## 6          0          0
##  MSSubClassduplex.all.style.age MSSubClass1.story.PUD.1946.
## 1          0          0
## 2          0          0
## 3          0          0
## 4          0          1
## 5          0          0
## 6          0          0
##  MSSubClass2.story.PUD.1946. MSSubClassPUD.multilevel
## 1          0          0
## 2          0          0
## 3          0          0
## 4          0          0
## 5          0          0
## 6          0          0
##  MSSubClass2.family.conversion MSZoningFV MSZoningRH MSZoningRL MSZoningRM
## 1          0          0          0          1          0
## 2          0          0          0          1          0
## 3          0          0          0          1          0
## 4          0          0          0          0          1
## 5          0          0          0          0          1
## 6          0          0          0          0          0
##  Street AlleyNone AlleyPave LotShape LandContourHLS LandContourLow
## 1      1      1      0      2      0      0
## 2      1      1      0      3      0      0
## 3      1      1      0      2      0      0
## 4      1      1      0      3      0      0

```

```

## 5      1      1      0      2      0      0
## 6      1      0      1      3      0      0
##   LandContourLvl LotConfigCulDSac LotConfigFR2 LotConfigInside LandSlope
## 1      1          0          1          0          2
## 2      1          0          0          1          2
## 3      1          0          0          0          2
## 4      1          0          0          1          2
## 5      1          0          0          1          2
## 6      1          0          0          1          2
##   NeighborhoodBrDale NeighborhoodBrkSide NeighborhoodClearCr
## 1      0          0          0
## 2      0          0          0
## 3      0          0          0
## 4      0          0          0
## 5      0          1          0
## 6      0          0          0
##   NeighborhoodCollgCr NeighborhoodCrawfor NeighborhoodEdwards
## 1      0          0          0
## 2      0          0          0
## 3      0          0          0
## 4      0          0          0
## 5      0          0          0
## 6      0          0          0
##   NeighborhoodGilbert NeighborhoodIDOTRR NeighborhoodMeadowV
## 1      0          0          0
## 2      0          0          0
## 3      0          0          0
## 4      0          0          1
## 5      0          0          0
## 6      0          1          0
##   NeighborhoodMitchel NeighborhoodNAmes NeighborhoodNoRidge NeighborhoodNrldgHt
## 1      0          0          1          0
## 2      0          0          0          0
## 3      0          0          0          1
## 4      0          0          0          0
## 5      0          0          0          0
## 6      0          0          0          0
##   NeighborhoodNWAmes NeighborhoodOldTown NeighborhoodSawyer NeighborhoodSawyerW
## 1      0          0          0          0
## 2      0          0          0          0
## 3      0          0          0          0
## 4      0          0          0          0
## 5      0          0          0          0
## 6      0          0          0          0
##   NeighborhoodSomerset NeighborhoodStoneBr NeighborhoodSWISU NeighborhoodTimber
## 1      0          0          0          0
## 2      1          0          0          0
## 3      0          0          0          0
## 4      0          0          0          0
## 5      0          0          0          0
## 6      0          0          0          0
##   NeighborhoodVeenker Condition1Feedr Condition1Norm Condition1PosN
## 1      0          0          1          0
## 2      0          0          1          0
## 3      0          0          1          0
## 4      0          0          1          0
## 5      0          1          0          0
## 6      0          1          0          0
##   Condition1RRAe Condition1RRAn Condition2Norm BldgType2fmCon BldgTypeDuplex
## 1      0          0          1          0          0
## 2      0          0          1          0          0

```

```

## 3          0          0          1          0          0
## 4          0          0          1          0          0
## 5          0          0          0          0          0
## 6          0          0          1          0          0
##   BldgTypeTwnhs BldgTypeTwnhsE HouseStyle1.5Unf HouseStyle1Story
## 1          0          0          0          0
## 2          0          0          0          1
## 3          0          0          0          0
## 4          0          1          0          1
## 5          0          0          0          1
## 6          0          0          0          0
##   HouseStyle2.5Unf HouseStyle2Story HouseStyleSFoyer HouseStyleSLvl OverallQual
## 1          0          1          0          0          8
## 2          0          0          0          0          8
## 3          0          1          0          0          8
## 4          0          0          0          0          5
## 5          0          0          0          0          4
## 6          0          1          0          0          4
##   OverallCond RoofStyleGable RoofStyleGambrel RoofStyleHip RoofMatlCompShg
## 1          5          1          0          0          1
## 2          5          1          0          0          1
## 3          5          1          0          0          1
## 4          7          1          0          0          1
## 5          6          1          0          0          1
## 6          4          0          1          0          1
##   RoofMatlTar.Grv Exterior1stBrkFace Exterior1stCemntBd Exterior1stHdBoard
## 1          0          0          0          0
## 2          0          0          0          0
## 3          0          0          0          0
## 4          0          0          1          0
## 5          0          0          0          0
## 6          0          1          0          0
##   Exterior1stMetalSd Exterior1stPlywood Exterior1stStucco Exterior1stVinylSd
## 1          0          0          0          1
## 2          0          0          0          1
## 3          0          0          0          1
## 4          0          0          0          0
## 5          1          0          0          0
## 6          0          0          0          0
##   Exterior1stWd.Sdng Exterior1stWdShing Exterior2ndBrkFace Exterior2ndCmentBd
## 1          0          0          0          0
## 2          0          0          0          0
## 3          0          0          0          0
## 4          0          0          0          1
## 5          0          0          0          0
## 6          0          0          1          0
##   Exterior2ndHdBoard Exterior2ndMetalSd Exterior2ndPlywood Exterior2ndStucco
## 1          0          0          0          0
## 2          0          0          0          0
## 3          0          0          0          0
## 4          0          0          0          0
## 5          0          1          0          0
## 6          0          0          0          0
##   Exterior2ndVinylSd Exterior2ndWd.Sdng Exterior2ndWd.Shng MasVnrTypeBrkFace
## 1          1          0          0          1
## 2          1          0          0          0
## 3          1          0          0          1
## 4          0          0          0          0
## 5          0          0          0          0
## 6          0          0          0          0
##   MasVnrTypeNone MasVnrTypeStone ExterQual ExterCond FoundationCBlock

```

```

## 1      0      0      4      3      0
## 2      0      1      4      3      0
## 3      0      0      4      3      0
## 4      1      0      3      3      0
## 5      1      0      3      3      0
## 6      1      0      3      2      0
##   FoundationPConc FoundationSlab BsmtQual BsmtCond BsmtExposure BsmtFinType1
## 1      1      0      4      3      3      6
## 2      1      0      5      3      3      6
## 3      1      0      5      3      3      1
## 4      1      0      4      3      1      6
## 5      0      0      3      3      1      1
## 6      0      0      3      3      1      1
##   BsmtFinType2 HeatingGasA HeatingGasW HeatingQC CentralAir ElectricalFuseF
## 1      1      1      0      5      1      0
## 2      1      1      0      5      1      0
## 3      1      1      0      5      1      0
## 4      1      1      0      3      1      0
## 5      1      1      0      2      0      0
## 6      1      1      0      3      0      0
##   ElectricalSBrkr KitchenQual Functional FireplaceQu GarageTypeAttchd
## 1      1      4      7      3      1
## 2      1      4      7      4      1
## 3      1      4      7      4      0
## 4      1      3      7      3      1
## 5      1      2      7      0      0
## 6      1      3      7      0      0
##   GarageTypeBasment GarageTypeBuiltIn GarageTypeDetchd GarageTypeNone
## 1      0      0      0      0
## 2      0      0      0      0
## 3      0      1      0      0
## 4      0      0      0      0
## 5      0      0      1      0
## 6      0      0      1      0
##   GarageFinish GarageQual GarageCond PavedDrive PoolQC FenceGdWo FenceMnPrv
## 1      2      3      3      2      0      0      0
## 2      2      3      3      2      0      0      0
## 3      2      3      3      2      0      0      0
## 4      1      3      3      2      0      0      0
## 5      1      2      3      2      0      0      0
## 6      1      3      2      0      0      0      1
##   FenceMnWw FenceNone MiscFeatureNone MiscFeatureShed MoSold2 MoSold3 MoSold4
## 1      0      1      1      0      0      0      0
## 2      0      1      1      0      0      0      0
## 3      0      1      1      0      0      0      0
## 4      0      1      1      0      0      0      0
## 5      0      1      1      0      0      0      0
## 6      0      0      1      0      0      0      0
##   MoSold5 MoSold6 MoSold7 MoSold8 MoSold9 MoSold10 MoSold11 MoSold12 YrSold2007
## 1      0      0      0      0      0      0      0      1      0
## 2      0      0      0      1      0      0      0      0      1
## 3      0      0      0      0      0      0      1      0      0
## 4      0      1      0      0      0      0      0      0      1
## 5      1      0      0      0      0      0      0      0      0
## 6      0      0      1      0      0      0      0      0      0
##   YrSold2008 YrSold2009 YrSold2010 SaleTypeNew SaleTypeWD SaleConditionAlloca
## 1      1      0      0      0      1      0
## 2      0      0      0      0      1      0
## 3      0      0      0      1      0      0
## 4      0      0      0      0      1      0
## 5      1      0      0      0      1      0

```

```

## 6      1      0      0      0      1      0
## SaleConditionFamily SaleConditionNormal SaleConditionPartial TotBathrooms
## 1          0          1          0      3.5
## 2          0          1          0      3.0
## 3          0          0          1      3.5
## 4          0          1          0      2.0
## 5          0          1          0      1.0
## 6          0          1          0      1.0
## Remod SalePrice
## 1    0 12.42922
## 2    1 12.63460
## 3    1 12.69250
## 4    0 11.77452
## 5    1 11.13459
## 6    1 10.59663

custom<- trainControl(method = "repeatedcv",number = 10,repeats = 5,verboseIter = T)
set.seed(1234)
ridge <- train(y=train_dummy$SalePrice, x=train_dummy[, !names(train_dummy) %in% "SalePrice"], method = 'glmnet',
                ,trControl =custom)

## + Fold01.Rep1: alpha=0, lambda=0.096
## - Fold01.Rep1: alpha=0, lambda=0.096
## + Fold02.Rep1: alpha=0, lambda=0.096
## - Fold02.Rep1: alpha=0, lambda=0.096
## + Fold03.Rep1: alpha=0, lambda=0.096
## - Fold03.Rep1: alpha=0, lambda=0.096
## + Fold04.Rep1: alpha=0, lambda=0.096
## - Fold04.Rep1: alpha=0, lambda=0.096
## + Fold05.Rep1: alpha=0, lambda=0.096
## - Fold05.Rep1: alpha=0, lambda=0.096
## + Fold06.Rep1: alpha=0, lambda=0.096
## - Fold06.Rep1: alpha=0, lambda=0.096
## + Fold07.Rep1: alpha=0, lambda=0.096
## - Fold07.Rep1: alpha=0, lambda=0.096
## + Fold08.Rep1: alpha=0, lambda=0.096
## - Fold08.Rep1: alpha=0, lambda=0.096
## + Fold09.Rep1: alpha=0, lambda=0.096
## - Fold09.Rep1: alpha=0, lambda=0.096
## + Fold10.Rep1: alpha=0, lambda=0.096
## - Fold10.Rep1: alpha=0, lambda=0.096
## + Fold01.Rep2: alpha=0, lambda=0.096
## - Fold01.Rep2: alpha=0, lambda=0.096
## + Fold02.Rep2: alpha=0, lambda=0.096
## - Fold02.Rep2: alpha=0, lambda=0.096
## + Fold03.Rep2: alpha=0, lambda=0.096
## - Fold03.Rep2: alpha=0, lambda=0.096
## + Fold04.Rep2: alpha=0, lambda=0.096
## - Fold04.Rep2: alpha=0, lambda=0.096
## + Fold05.Rep2: alpha=0, lambda=0.096
## - Fold05.Rep2: alpha=0, lambda=0.096
## + Fold06.Rep2: alpha=0, lambda=0.096
## - Fold06.Rep2: alpha=0, lambda=0.096
## + Fold07.Rep2: alpha=0, lambda=0.096
## - Fold07.Rep2: alpha=0, lambda=0.096
## + Fold08.Rep2: alpha=0, lambda=0.096
## - Fold08.Rep2: alpha=0, lambda=0.096
## + Fold09.Rep2: alpha=0, lambda=0.096
## - Fold09.Rep2: alpha=0, lambda=0.096
## + Fold10.Rep2: alpha=0, lambda=0.096
## - Fold10.Rep2: alpha=0, lambda=0.096

```



```

## Selecting tuning parameters
## Fitting alpha = 0, lambda = 0.031 on full training set

ridge$bestTune

##   alpha lambda
## 7      0  0.031

min(ridge$results$RMSE)

## [1] 0.1175073

a <- predict(ridge,test_dummy[,!names(test_dummy) %in% "SalePrice"])
exp(a)

## [1] 308952.58 274681.97 341163.47 137295.61 68945.41 93313.51 142201.23
## [8] 144853.12 131404.61 72436.94 156710.49 288014.52 129754.08 174447.71
## [15] 294858.92 241639.07 242372.93 100018.07 116157.65 117041.38 122783.15
## [22] 206041.41 164101.44 168938.51 184664.50 197030.09 244612.80 109321.40
## [29] 201706.04 144775.71 159420.16 230578.25 231557.46 203276.98 294401.91
## [36] 91484.23 120032.50 192850.23 161641.48 175519.82 114211.12 283469.46
## [43] 324275.49 109876.79 170411.21 246508.12 91021.50 106449.09 348173.41
## [50] 94313.23 125304.79 103429.42 121806.06 138329.21 199850.80 185700.31
## [57] 258296.54 322751.08 169721.18 95349.14 251599.05 240241.96 171834.50
## [64] 172229.14 272502.06 195897.23 239460.47 88882.81 180278.66 190848.93
## [71] 145233.64 135462.13 189170.87 179027.25 173551.09 179027.05 167591.97
## [78] 83511.99 188462.04 166432.26 103048.69 199855.36 141476.12 116076.88
## [85] 201372.17 95954.62 120090.25 135088.22 195496.63 224355.49 117495.35
## [92] 164409.28 259049.00 232649.86 192542.81 133147.94 143699.25 171975.02
## [99] 137258.16 94239.38 119892.34 130289.14 141558.01 167786.68 143062.92
## [106] 400857.34 205573.27 164695.78 238279.66 115228.85 169408.95 114262.78
## [113] 215497.33 127030.57 319013.60 122779.48 139020.20 201411.92 302171.95
## [120] 200971.91 191386.25 290099.19 193914.84 221366.56 167643.21 125199.55
## [127] 111003.54 134051.40 143710.06 84158.87 320052.39 148793.83 87414.20
## [134] 119803.81 167211.88 202674.89 129862.82 288028.95 107279.29 333842.33
## [141] 132432.62 112133.93 176943.32 222648.82 195636.00 298414.01 107046.00
## [148] 131129.35 153174.17 185483.94 264190.23 157271.10 212900.28 90095.00
## [155] 139911.55 312427.93 76857.69 164816.10 210982.15 200130.81 320938.40
## [162] 270135.05 220711.34 156943.71 163913.40 156779.05 166313.95 114699.76
## [169] 211981.18 154795.21 160132.37 190612.47 149096.73 239433.13 217982.89
## [176] 134802.15 140703.02 194945.24 167568.45 247473.74 125226.33 326185.14
## [183] 169885.15 143110.54 128100.82 157739.85 152549.35 125451.11 204765.01
## [190] 161648.73 118750.98 211578.86 135548.11 142338.93 129222.11 174324.02
## [197] 186570.54 81710.13 121337.40 199698.39 137630.83 320020.13 85142.01
## [204] 199158.33 90400.00 119140.25 222474.05 160594.15 164244.45 234050.30
## [211] 208682.80 228509.41 159612.43 105345.94 191397.41 153950.60 191101.53
## [218] 151148.55 91040.46 247419.81 207115.97 87123.02 169677.55 181562.06
## [225] 309737.66 134384.08 229403.29 82557.51 133420.77 175036.89 100735.96
## [232] 189550.67 195548.34 192433.58 191375.16 57911.46 117235.45 112860.59
## [239] 197353.88 120969.38 185175.63 184803.92 232532.11 134774.09 219234.37
## [246] 291417.47 91348.81 125594.92 106780.72 109804.90 210463.80 274283.37
## [253] 132931.92 186706.69 133926.38 242781.93 176219.20 294452.16 116699.50
## [260] 148892.48 296936.02 178715.76 90026.69 217879.90 122282.63 180240.45
## [267] 195961.80 267847.66 194959.23 358134.51 256585.91 246336.90 80810.44
## [274] 155433.35 178603.83 154883.47 127998.60 90765.48 211671.39 145358.74
## [281] 140943.93 75078.19 265323.29 126265.55 103161.22 185265.39 250558.59

```

```

rmse_rf <- sqrt(mean((exp(test_dummy$SalePrice) - exp(a))^2))
rmse_rf

## [1] 22879.93

rss <- sum((exp(a) - exp(test_dummy$SalePrice)) ^ 2) ## residual sum of squares
tss <- sum((exp(test_dummy$SalePrice) - mean(exp(test_dummy$SalePrice))) ^ 2) ## total sum of squares
rsq_rf <- 1 - rss/tss
rsq_rf

## [1] 0.883728

ridgevar <- varImp(ridge)

n <- nrow(train_dummy) + nrow(test_dummy)
k <- ncol(train)
adjr2 <- 1 - ((1 - rsq_rf) * (n - 1)) / (n - k - 1)
adjr2

## [1] 0.8769987

install.packages("randomForest")
http://r-statistics.co/Variable-Selection-and-Importance-With-R.html

```

Random forest model

```

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.0.3

## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

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

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

set.seed(2018)
fit <- randomForest(SalePrice ~ ., data = train, ntree = 500, mtry = 8, importance = TRUE)
fit

##
## Call:
##   randomForest(formula = SalePrice ~ ., data = train, ntree = 500,      mtry = 8, importance = TRUE)
##   Type of random forest: regression
##   Number of trees: 500
##   No. of variables tried at each split: 8
##   Mean of squared residuals: 0.01952665
##   % Var explained: 87.39

```

Variable importance plot

```
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.3

##
## Attaching package: 'dplyr'

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

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

## The following objects are masked from 'package:plyr':
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

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

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

library(ggplot2)
library(randomForest)
library(ranger)

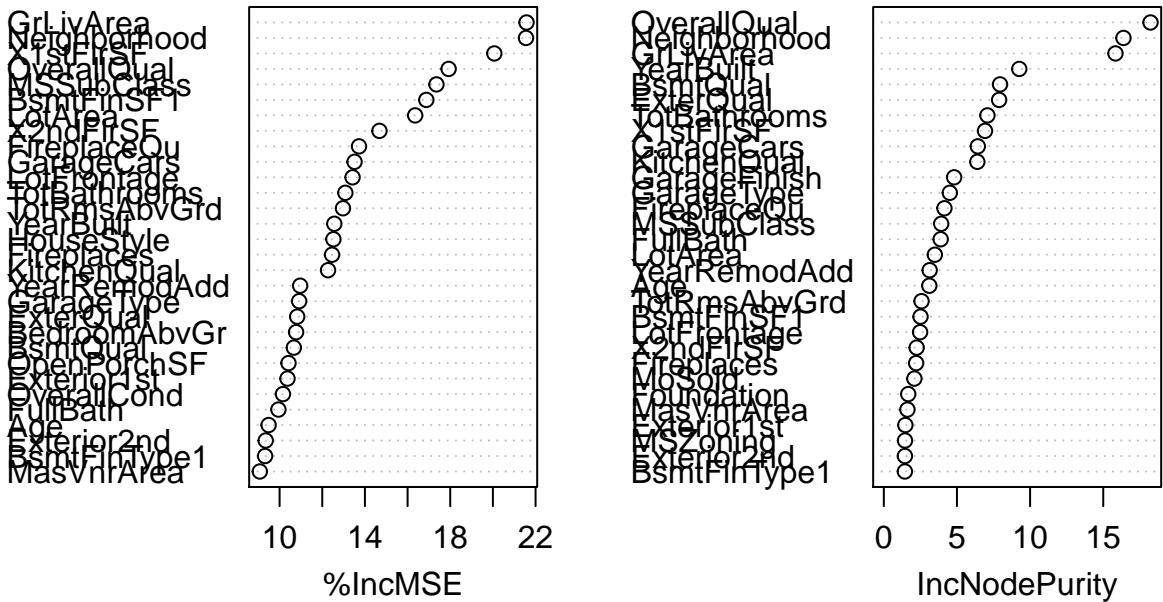
## Warning: package 'ranger' was built under R version 4.0.3

##
## Attaching package: 'ranger'

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

varImpPlot(fit)
```

fit



Top 20 Important variables

```

imp_vars <- (fit$importance)
imp_vars <- sort(imp_vars[,1], decreasing = TRUE)
imp_vars <- imp_vars[1:20]
imp_vars

##      GrLivArea  OverallQual Neighborhood      YearBuilt      X1stFlrSF TotBathrooms
## 0.017205373 0.014642275 0.012387130 0.007975967 0.007202503 0.005999631
##      ExterQual   GarageCars     BsmtQual   KitchenQual    MSSubClass   GarageType
## 0.005161476 0.004812489 0.004806193 0.004433260 0.004050518 0.003559346
##  YearRemodAdd    X2ndFlrSF        Age      FullBath   FireplaceQu      LotArea
## 0.003393891 0.003049048 0.003021403 0.003001277 0.002974967 0.002533384
##  GarageFinish    BsmtFinSF1
## 0.002511131 0.002401321

```

RF model assesment

```

pred_values = exp(predict(fit,test[,!names(test) %in% "SalePrice"]))
actual_values = exp(test$SalePrice)
rmse_rf <- sqrt(mean((actual_values - pred_values)^2))
rmse_rf

## [1] 25086.3

rss <- sum((pred_values - actual_values)^2) ## residual sum of squares
tss <- sum((actual_values - mean(actual_values))^2) ## total sum of squares
rsq_f <- 1 - rss/tss
rsq_f

```

```

## [1] 0.8805027

adjr2 <- 1 - (((1 - rsq_f) * (n - 1)) / (n - k - 1))
adjr2

```

```

## [1] 0.8735868

```

Gradient boosting

for gradient boosting tutorial <https://xgboost.readthedocs.io/en/latest/R-package/xgboostPresentation.html>

Default gradient boosting model

```

label_train <- train_dummy$SalePrice

library(xgboost)

## Warning: package 'xgboost' was built under R version 4.0.3

##
## Attaching package: 'xgboost'

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

set.seed(1)
# train and test data for xgb model
train_xgb <- xgb.DMatrix(data = as.matrix(train_dummy[, !names(train_dummy) %in% "SalePrice"]), label = label_train)
test_xgb <- xgb.DMatrix(data = as.matrix(test_dummy[, !names(test_dummy) %in% "SalePrice"]))
default_param<-list(
  objective = "reg:squarederror",
  booster = "gbtree",
  eta=0.3, #default = 0.3
  gamma=0,
  max_depth=6, #default=6
  min_child_weight=1, #default=1
  subsample=1,
  colsample_bytree=1
)
# cross validation for number of rounds
cv_xgb <- xgb.cv( params = default_param, data = train_xgb, nrounds = 500, nfolds = 5, showsd = T, stratified = T

## [1] train-rmse:8.079474+0.005718    test-rmse:8.079385+0.034986
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [41] train-rmse:0.026288+0.001007    test-rmse:0.138276+0.011188
## Stopping. Best iteration:
## [49] train-rmse:0.020800+0.001580    test-rmse:0.137917+0.011280

```

we got a train-rmse of 0.025476 and a test-rmse of 0.145537.

Tuning hyperparameters


```

## [23:26:05] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:08] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:08] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:08] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:08] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:08] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:08] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:08] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:09] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:09] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo
## [23:26:09] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated in favo

```

```
calibration_model$bestTune
```

```

##      nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 74      50       6 0.1     0             1                 4               1

default_param<-list(
  objective = "reg:squarederror",
  booster = "gbtree",
  eta=0.1, #default = 0.3
  gamma=0,
  max_depth=6, #default=6
  min_child_weight=4, #default=1
  subsample=1,
  colsample_bytree=1
)
# cross validation for number of rounds
cv_xgb <- xgb.cv( params = default_param, data = train_xgb, nrounds = 500, nfolds = 5, showsd = T, stratified = T)

## [1] train-rmse:10.379434+0.003790    test-rmse:10.379425+0.017329
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [41] train-rmse:0.187328+0.000903    test-rmse:0.210624+0.012968
## [81] train-rmse:0.051054+0.002174    test-rmse:0.124570+0.010094
## Stopping. Best iteration:
## [107]   train-rmse:0.039782+0.001496    test-rmse:0.123084+0.010192

```

After tuning the model parameters we got train-rmse: 0.029925 and test-rmse: 0.136163 which is an improvement on the test-rmse when compared to the default training parameter values.

```
model <- xgb.train(data = train_xgb, params=default_param, nrounds = 142)
pred <- exp(predict(model, test_xgb))
```

```

actual <- exp(test_dummy$SalePrice)
rmse <- sqrt(mean((actual - pred)^2))
rmse

## [1] 21069.97

rss <- sum((pred - actual) ^ 2) ## residual sum of squares
tss <- sum((actual - mean(actual)) ^ 2) ## total sum of squares
rsq <- 1 - rss/tss
rsq

## [1] 0.9013962

adjr2 <- 1 - (((1 - rsq) * (n - 1)) / (n - k - 1))
adjr2

## [1] 0.8956894

head(data.frame(actual, pred))

##   actual      pred
## 1 250000 296124.58
## 2 307000 276205.16
## 3 325300 350643.62
## 4 129900 134288.78
## 5  68500  67982.50
## 6  40000  84470.38

```

Results