

final_project

Simran Kota, Yusef Haswarey

Packages Used

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v ggplot2 3.3.5      v purrr  0.3.4
## v tibble  3.1.5      v dplyr  1.0.7
## v tidyr   1.1.4      v stringr 1.4.0
## v readr   2.0.2      v forcats 0.5.1

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

library(randomForest)

## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##   combine
##
## The following object is masked from 'package:ggplot2':
##
##   margin

library(caret)

## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##   lift

library(ggpubr)
```

Research Question

We are attempting to explore the relationship between various features of a house (ex. number of rooms, square footage, year built, etc.) on the future value of the house.

Data Source and Definitions

The USA Housing Dataset from Kaggle provides data regarding sales prices with respect to various houses in the US. It has 81 features describing each record for 1460 records.

```
dat <- read.csv("housing_data.csv")
str(dat)

## 'data.frame':    1460 obs. of  81 variables:
## $ Id             : int  1 2 3 4 5 6 7 8 9 10 ...
## $ MSSubClass     : int  60 20 60 70 60 50 20 60 50 190 ...
## $ MSZoning       : 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 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 0 ...
## $ GrLivArea       : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
## $ BsmtFullBath    : int 1 0 1 1 1 1 1 1 0 1 ...
## $ BsmtHalfBath    : int 0 1 0 0 0 0 0 0 0 0 ...
## $ FullBath        : int 2 2 2 1 2 1 2 2 2 1 ...
## $ HalfBath        : int 1 0 1 0 1 1 0 1 0 0 ...
## $ BedroomAbvGr    : int 3 3 3 3 4 1 3 3 2 2 ...
## $ KitchenAbvGr    : int 1 1 1 1 1 1 1 1 2 2 ...
## $ KitchenQual      : 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 ...
```

We don't care about the Id column, so we will drop it.

```
dat <- dat[,-c(1)]
str(dat)
```

```
## '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 ...
```

While the full variable descriptions can be found in the data_description.txt file in the repository, we have included a summary below:

MSSubClass: type of house, categorical

MSZoning: zone of house, categorical

LotFrontage: Linear feet of street connected to house, continuous

LotArea: Lot size in square feet, continuous

Street: Type of road access to house, categorical

Alley: Type of alley access to house, categorical

LotShape: General shape of property, categorical

LandContour: Flatness of the property, categorical

Utilities: Type of utilities available, categorical

LotConfig: Type of lot, categorical

LandSlope: Slope of property, categorical

Neighborhood: Physical location, categorical

Condition1: Proximity to various conditions, categorical

Condition2: Proximity to various conditions (if more than one is present), categorical

BldgType: Type of home, categorical

HouseStyle: Style of home, categorical

OverallQual: Rating of the overall material and finish of the house, categorical

OverallCond: Rating of the overall condition of the house, categorical

YearBuilt: Original construction date, continuous

YearRemodAdd: Remodel date (same as construction date if no remodeling or additions), continuous

RoofStyle: Type of roof, categorical

RoofMatl: Roof material, categorical
 Exterior1st: Exterior covering on house, categorical
 Exterior2nd: Exterior covering on house (if more than one material), categorical
 MasVnrType: Masonry veneer type, categorical
 MasVnrArea: Masonry veneer area in square feet, categorical
 ExterQual: Evaluates the quality of the material on the exterior, categorical
 ExterCond: Evaluates the present condition of the material on the exterior, categorical
 Foundation: Type of foundation, categorical
 BsmtQual: Evaluates the height of the basement, categorical
 BsmtCond: Evaluates the general condition of the basement, categorical
 BsmtExposure: Refers to walkout or garden level walls, categorical
 BsmtFinType1: Rating of basement finished area, categorical
 BsmtFinSF1: Type 1 finished square feet, continuous
 BsmtFinType2: Rating of basement finished area (if multiple types), categorical
 BsmtFinSF2: Type 2 finished square feet, continuous
 BsmtUnfSF: Unfinished square feet of basement area, continuous
 TotalBsmtSF: Total square feet of basement area, continuous
 Heating: Type of heating, categorical
 HeatingQC: Heating quality and condition, categorical
 CentralAir: Central air conditioning, categorical (binary)
 Electrical: Electrical system, categorical
 1stFlrSF: First Floor square feet, continuous
 2ndFlrSF: Second floor square feet, continuous
 LowQualFinSF: Low quality finished square feet (all floors), continuous
 GrLivArea: Above ground living area square feet, continuous
 BsmtFullBath: Basement full bathrooms, continuous
 BsmtHalfBath: Basement half bathrooms, continuous
 FullBath: Full bathrooms above ground, continuous
 HalfBath: Half baths above ground, continuous
 Bedroom: Bedrooms above ground (does NOT include basement bedrooms), continuous
 Kitchen: Kitchens above grade, continuous
 KitchenQual: Kitchen quality, categorical
 TotRmsAbvGrd: Total rooms above ground (does not include bathrooms), continuous
 Functional: Home functionality, categorical
 Fireplaces: Number of fireplaces, continuous
 FireplaceQu: Fireplace quality, categorical

GarageType: Garage location, categorical
 GarageYrBlt: Year garage was built, continuous
 GarageFinish: Interior finish of the garage, categorical
 GarageCars: Size of garage in car capacity, continuous
 GarageArea: Size of garage in square feet, continuous
 GarageQual: Garage quality, categorical
 GarageCond: Garage condition, categorical
 PavedDrive: Paved driveway, categorical
 WoodDeckSF: Wood deck area in square feet, continuous
 OpenPorchSF: Open porch area in square feet, continuous
 EnclosedPorch: Enclosed porch area in square feet, continuous
 3SsnPorch: Three season porch area in square feet, continuous
 ScreenPorch: Screen porch area in square feet, continuous
 PoolArea: Pool area in square feet, continuous
 PoolQC: Pool quality, categorical
 Fence: Fence quality, categorical
 MiscFeature: Miscellaneous feature not covered in other categories, categorical
 MiscVal: Value of miscellaneous feature, continuous
 MoSold: Month Sold (MM), continuous
 YrSold: Year Sold (YYYY), continuous
 SaleType: Type of sale, categorical
 SaleCondition: Condition of sale, categorical
 SalePrice: Price the house was sold for, continuous

Although some of the variables were automatically mistyped by R, we don't need all the variables for our analysis. We will first drop some variables to reduce the dataset to the data required, then retype the variables as needed.

```
dat <- subset(dat, select=-c(Condition1, Condition2, Exterior2nd, MasVnrType, MasVnrArea, BsmtExposure,
dim(dat)
```

```
## [1] 1460    69
```

We are left with 69 features out of the original 81. We still have 1460 records, however, it is highly likely at least some of these have null values. We will examine these and clean them up next.

```
colSums(is.na(dat))
```

##	MSSubClass	MSZoning	LotFrontage	LotArea	Street
##	0	0	259	0	0
##	Alley	LotShape	LandContour	Utilities	LotConfig
##	1369	0	0	0	0
##	LandSlope	Neighborhood	BldgType	HouseStyle	OverallQual
##	0	0	0	0	0
##	OverallCond	YearBuilt	YearRemodAdd	RoofStyle	RoofMatl
##	0	0	0	0	0

```
## Exterior1st      ExterQual      ExterCond      Foundation      BsmtQual
##           0           0           0           0           37
##      BsmtCond BsmtFinType1 BsmtFinSF1 BsmtUnfSF TotalBsmtSF
##           37           37           0           0           0
##      Heating      HeatingQC      CentralAir      Electrical      X1stFlrSF
##           0           0           0           1           0
##      X2ndFlrSF LowQualFinSF      GrLivArea BsmtFullBath BsmtHalfBath
##           0           0           0           0           0
##      FullBath      HalfBath BedroomAbvGr KitchenAbvGr KitchenQual
##           0           0           0           0           0
## TotRmsAbvGrd      Functional      Fireplaces      FireplaceQu      GarageType
##           0           0           0           690           81
## GarageFinish      GarageCars      GarageArea      GarageQual      GarageCond
##           81           0           0           81           81
##      PavedDrive      WoodDeckSF      OpenPorchSF      EnclosedPorch      X3SsnPorch
##           0           0           0           0           0
##      ScreenPorch      PoolArea      PoolQC      Fence      MiscFeature
##           0           0           1453           1179           1406
##      MiscVal      MoSold      YrSold      SalePrice
##           0           0           0           0
```

```
dat$LotFrontage[is.na(dat$LotFrontage)] <- 0
dat$Alley[is.na(dat$Alley)] <- "None"
dat$BsmtQual[is.na(dat$BsmtQual)] <- "None"
dat$BsmtCond[is.na(dat$BsmtCond)] <- "None"
dat$BsmtFinType1[is.na(dat$BsmtFinType1)] <- "None"
dat$Electrical[is.na(dat$Electrical)] <- "Unknown"
dat$FireplaceQu[is.na(dat$FireplaceQu)] <- "None"
dat$GarageType[is.na(dat$GarageType)] <- "None"
dat$GarageFinish[is.na(dat$GarageFinish)] <- "None"
dat$GarageQual[is.na(dat$GarageQual)] <- "None"
dat$GarageCond[is.na(dat$GarageCond)] <- "None"
dat$PoolQC[is.na(dat$PoolQC)] <- "None"
dat$Fence[is.na(dat$Fence)] <- "None"
dat$MiscFeature[is.na(dat$MiscFeature)] <- "None"
```

Checking our work:

```
colSums(is.na(dat))
```

```
## MSSubClass      MSZoning      LotFrontage      LotArea      Street
##           0           0           0           0           0
##      Alley      LotShape      LandContour      Utilities      LotConfig
##           0           0           0           0           0
##      LandSlope      Neighborhood      BldgType      HouseStyle      OverallQual
##           0           0           0           0           0
## OverallCond      YearBuilt      YearRemodAdd      RoofStyle      RoofMatl
##           0           0           0           0           0
## Exterior1st      ExterQual      ExterCond      Foundation      BsmtQual
##           0           0           0           0           0
##      BsmtCond BsmtFinType1 BsmtFinSF1 BsmtUnfSF TotalBsmtSF
##           0           0           0           0           0
##      Heating      HeatingQC      CentralAir      Electrical      X1stFlrSF
##           0           0           0           0           0
##      X2ndFlrSF LowQualFinSF      GrLivArea BsmtFullBath BsmtHalfBath
##           0           0           0           0           0
```


##	FullBath	HalfBath	BedroomAbvGr	KitchenAbvGr	KitchenQual
##	0	0	0	0	0
##	TotRmsAbvGrd	Functional	Fireplaces	FireplaceQu	GarageType
##	0	0	0	0	0
##	GarageFinish	GarageCars	GarageArea	GarageQual	GarageCond
##	0	0	0	0	0
##	PavedDrive	WoodDeckSF	OpenPorchSF	EnclosedPorch	X3SsnPorch
##	0	0	0	0	0
##	ScreenPorch	PoolArea	PoolQC	Fence	MiscFeature
##	0	0	0	0	0
##	MiscVal	MoSold	YrSold	SalePrice	
##	0	0	0	0	

Now that we have removed all the nulls, we will retype our variables.

```

dat$MSSubClass <- as.factor(dat$MSSubClass)
dat$MSZoning <- as.factor(dat$MSZoning)
dat$Street <- as.factor(dat$Street)
dat$Alley <- as.factor(dat$Alley)
dat$LotShape <- as.factor(dat$LotShape)
dat$LandContour <- as.factor(dat$LandContour)
dat$Utilities <- as.factor(dat$Utilities)
dat$LotConfig <- as.factor(dat$LotConfig)
dat$LandSlope <- as.factor(dat$LandSlope)
dat$Neighborhood <- as.factor(dat$Neighborhood)
dat$BldgType <- as.factor(dat$BldgType)
dat$HouseStyle <- as.factor(dat$HouseStyle)
dat$OverallQual <- as.factor(dat$OverallQual)
dat$OverallCond <- as.factor(dat$OverallCond)
dat$RoofStyle <- as.factor(dat$RoofStyle)
dat$RoofMatl <- as.factor(dat$RoofMatl)
dat$Exterior1st <- as.factor(dat$Exterior1st)
dat$ExterQual <- as.factor(dat$ExterQual)
dat$ExterCond <- as.factor(dat$ExterCond)
dat$Foundation <- as.factor(dat$Foundation)
dat$BsmtQual <- as.factor(dat$BsmtQual)
dat$BsmtCond <- as.factor(dat$BsmtCond)
dat$BsmtFinType1 <- as.factor(dat$BsmtFinType1)
dat$Heating <- as.factor(dat$Heating)
dat$HeatingQC <- as.factor(dat$HeatingQC)
dat$CentralAir <- as.factor(dat$CentralAir)
dat$Electrical <- as.factor(dat$Electrical)
dat$KitchenQual <- as.factor(dat$KitchenQual)
dat$Functional <- as.factor(dat$Functional)
dat$FireplaceQu <- as.factor(dat$FireplaceQu)
dat$GarageType <- as.factor(dat$GarageType)
dat$GarageFinish <- as.factor(dat$GarageFinish)
dat$GarageQual <- as.factor(dat$GarageQual)
dat$GarageCond <- as.factor(dat$GarageCond)
dat$PavedDrive <- as.factor(dat$PavedDrive)
dat$PoolQC <- as.factor(dat$PoolQC)
dat$Fence <- as.factor(dat$Fence)
dat$MiscFeature <- as.factor(dat$MiscFeature)

```

Checking our work:

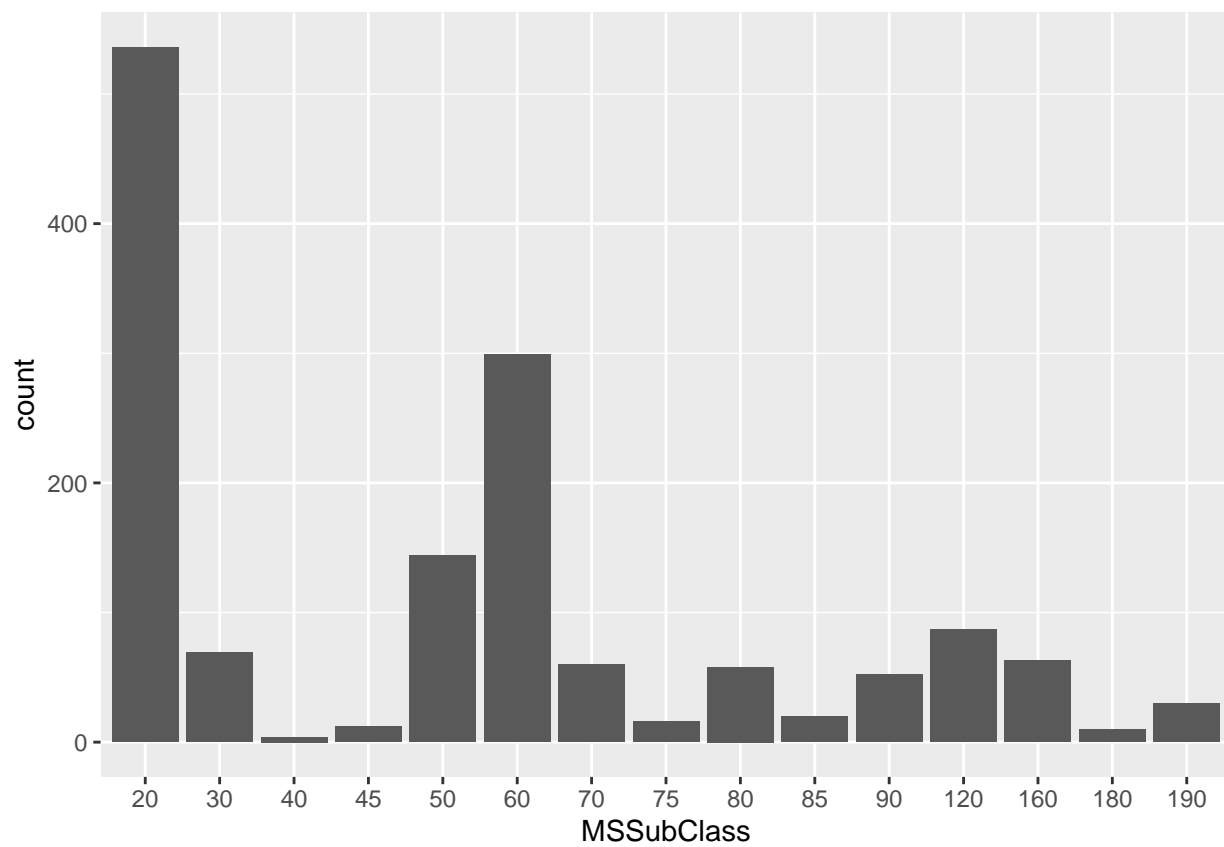
```
str(dat)
```

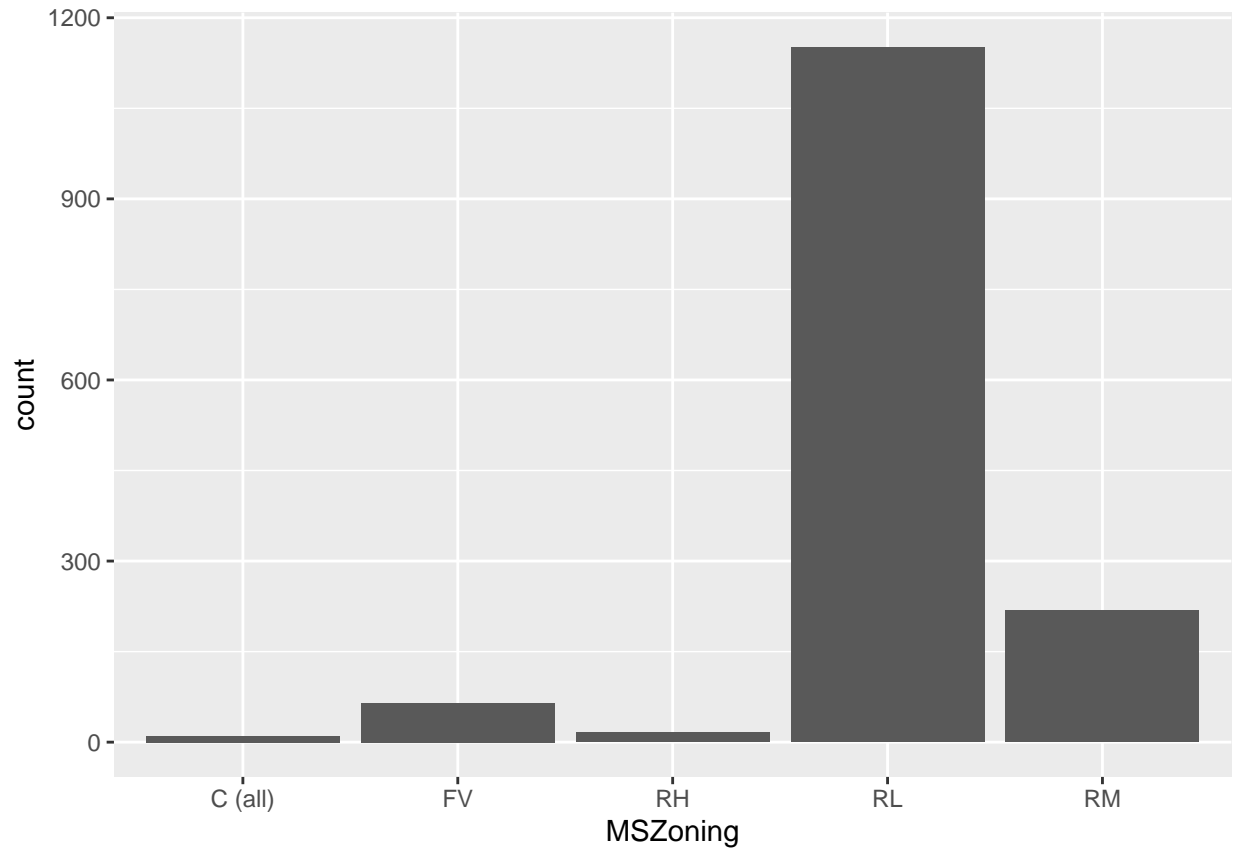
```
## 'data.frame':    1460 obs. of  69 variables:
## $ MSSubClass      : Factor w/ 15 levels "20","30","40",...: 6 1 6 7 6 5 1 6 5 15 ...
## $ MSZoning        : Factor w/ 5 levels "C (all)","FV",...: 4 4 4 4 4 4 4 4 5 4 ...
## $ LotFrontage     : num  65 80 68 60 84 85 75 0 51 50 ...
## $ LotArea         : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
## $ Street          : Factor w/ 2 levels "Grv1","Pave": 2 2 2 2 2 2 2 2 2 2 ...
## $ Alley           : Factor w/ 3 levels "Grv1","None",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ LotShape        : Factor w/ 4 levels "IR1","IR2","IR3",...: 4 4 1 1 1 1 1 4 1 4 4 ...
## $ LandContour     : Factor w/ 4 levels "Bnk","HLS","Low",...: 4 4 4 4 4 4 4 4 4 4 ...
## $ Utilities       : Factor w/ 2 levels "AllPub","NoSeWa": 1 1 1 1 1 1 1 1 1 1 ...
## $ LotConfig       : Factor w/ 5 levels "Corner","CulDSac",...: 5 3 5 1 3 5 5 1 5 1 ...
## $ LandSlope       : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
## $ Neighborhood    : Factor w/ 25 levels "Blmngtn","Blueste",...: 6 25 6 7 14 12 21 17 18 4 ...
## $ 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     : Factor w/ 10 levels "1","2","3","4",...: 7 6 7 7 8 5 8 7 7 5 ...
## $ OverallCond     : Factor w/ 9 levels "1","2","3","4",...: 5 8 5 5 5 5 5 6 5 6 ...
## $ YearBuilt       : int   2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
## $ YearRemodAdd    : int   2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
## $ RoofStyle       : Factor w/ 6 levels "Flat","Gable",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ RoofMatl        : Factor w/ 8 levels "ClyTile","CompShg",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ Exterior1st     : Factor w/ 15 levels "AsbShng","AsphShn",...: 13 9 13 14 13 13 13 7 4 9 ...
## $ ExterQual       : Factor w/ 4 levels "Ex","Fa","Gd",...: 3 4 3 4 3 4 3 4 4 4 ...
## $ ExterCond       : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 5 5 5 5 5 5 5 5 ...
## $ Foundation      : Factor w/ 6 levels "BrkTil","CBlock",...: 3 2 3 1 3 6 3 2 1 1 ...
## $ BsmtQual        : Factor w/ 5 levels "Ex","Fa","Gd",...: 3 3 3 5 3 3 1 3 5 5 ...
## $ BsmtCond        : Factor w/ 5 levels "Fa","Gd","None",...: 5 5 5 2 5 5 5 5 5 5 ...
## $ BsmtFinType1    : Factor w/ 7 levels "ALQ","BLQ","GLQ",...: 3 1 3 1 3 3 3 1 7 3 ...
## $ BsmtFinSF1      : int    706 978 486 216 655 732 1369 859 0 851 ...
## $ BsmtUnfSF       : int    150 284 434 540 490 64 317 216 952 140 ...
## $ TotalBsmtSF     : int    856 1262 920 756 1145 796 1686 1107 952 991 ...
## $ Heating         : Factor w/ 6 levels "Floor","GasA",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ HeatingQC       : Factor w/ 5 levels "Ex","Fa","Gd",...: 1 1 1 3 1 1 1 1 1 3 1 ...
## $ CentralAir      : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
## $ Electrical      : Factor w/ 6 levels "FuseA","FuseF",...: 5 5 5 5 5 5 5 5 5 2 5 ...
## $ X1stFlrSF       : int    856 1262 920 961 1145 796 1694 1107 1022 1077 ...
## $ X2ndFlrSF       : int    854 0 866 756 1053 566 0 983 752 0 ...
## $ LowQualFinSF    : int     0 0 0 0 0 0 0 0 0 0 ...
## $ GrLivArea       : int   1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
## $ BsmtFullBath    : int     1 0 1 1 1 1 1 1 0 1 ...
## $ BsmtHalfBath    : int     0 1 0 0 0 0 0 0 0 0 ...
## $ FullBath        : int     2 2 2 1 2 1 2 2 2 1 ...
## $ HalfBath        : int     1 0 1 0 1 1 0 1 0 0 ...
## $ BedroomAbvGr    : int     3 3 3 3 4 1 3 3 2 2 ...
## $ KitchenAbvGr    : int     1 1 1 1 1 1 1 1 2 2 ...
## $ KitchenQual     : Factor w/ 4 levels "Ex","Fa","Gd",...: 3 4 3 3 3 4 3 4 4 4 ...
## $ TotRmsAbvGrd    : int     8 6 6 7 9 5 7 7 8 5 ...
## $ Functional      : Factor w/ 7 levels "Maj1","Maj2",...: 7 7 7 7 7 7 7 7 3 7 ...
## $ Fireplaces      : int     0 1 1 1 1 0 1 2 2 2 ...
## $ FireplaceQu     : Factor w/ 6 levels "Ex","Fa","Gd",...: 4 6 6 3 6 4 3 6 6 6 ...
## $ GarageType      : Factor w/ 7 levels "2Types","Attchd",...: 2 2 2 6 2 2 2 2 6 2 ...
## $ GarageFinish    : Factor w/ 4 levels "Fin","None","RFn",...: 3 3 3 4 3 4 3 3 4 3 ...
```

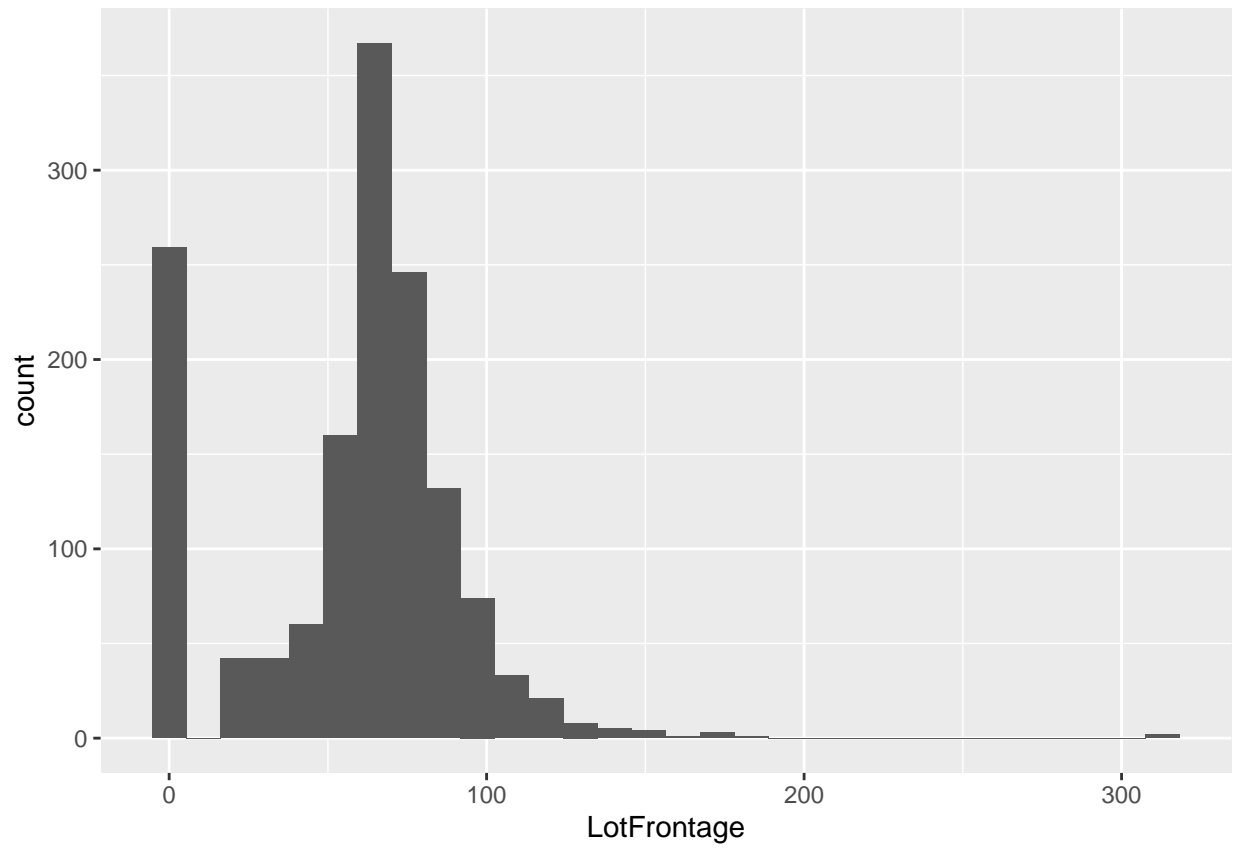
```
## $ GarageCars : int 2 2 2 3 3 2 2 2 1 ...
## $ GarageArea : int 548 460 608 642 836 480 636 484 468 205 ...
## $ GarageQual : Factor w/ 6 levels "Ex","Fa","Gd",...: 6 6 6 6 6 6 6 6 2 3 ...
## $ GarageCond : Factor w/ 6 levels "Ex","Fa","Gd",...: 6 6 6 6 6 6 6 6 6 6 ...
## $ PavedDrive : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
## $ WoodDeckSF : int 0 298 0 0 192 40 255 235 90 0 ...
## $ OpenPorchSF : int 61 0 42 35 84 30 57 204 0 4 ...
## $ EnclosedPorch: int 0 0 0 272 0 0 0 228 205 0 ...
## $ X3SsnPorch : int 0 0 0 0 0 320 0 0 0 0 ...
## $ ScreenPorch : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PoolQC : Factor w/ 4 levels "Ex","Fa","Gd",...: 4 4 4 4 4 4 4 4 4 4 ...
## $ Fence : Factor w/ 5 levels "GdPrv","GdWo",...: 5 5 5 5 5 3 5 5 5 5 ...
## $ MiscFeature : Factor w/ 5 levels "Gar2","None",...: 2 2 2 2 2 4 2 4 2 2 ...
## $ MiscVal : int 0 0 0 0 0 700 0 350 0 0 ...
## $ MoSold : int 2 5 9 2 12 10 8 11 4 1 ...
## $ YrSold : int 2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
## $ SalePrice : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
```

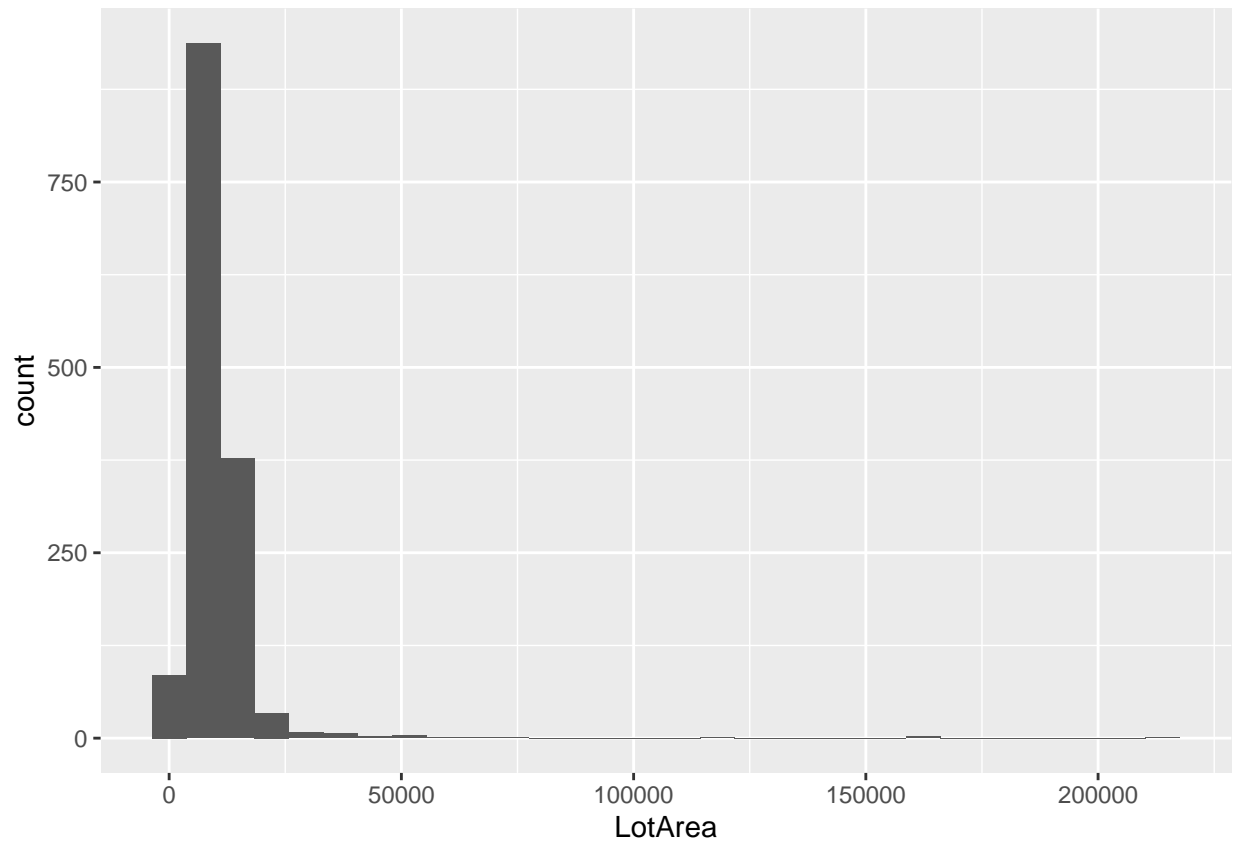
Exploratory Data Analysis

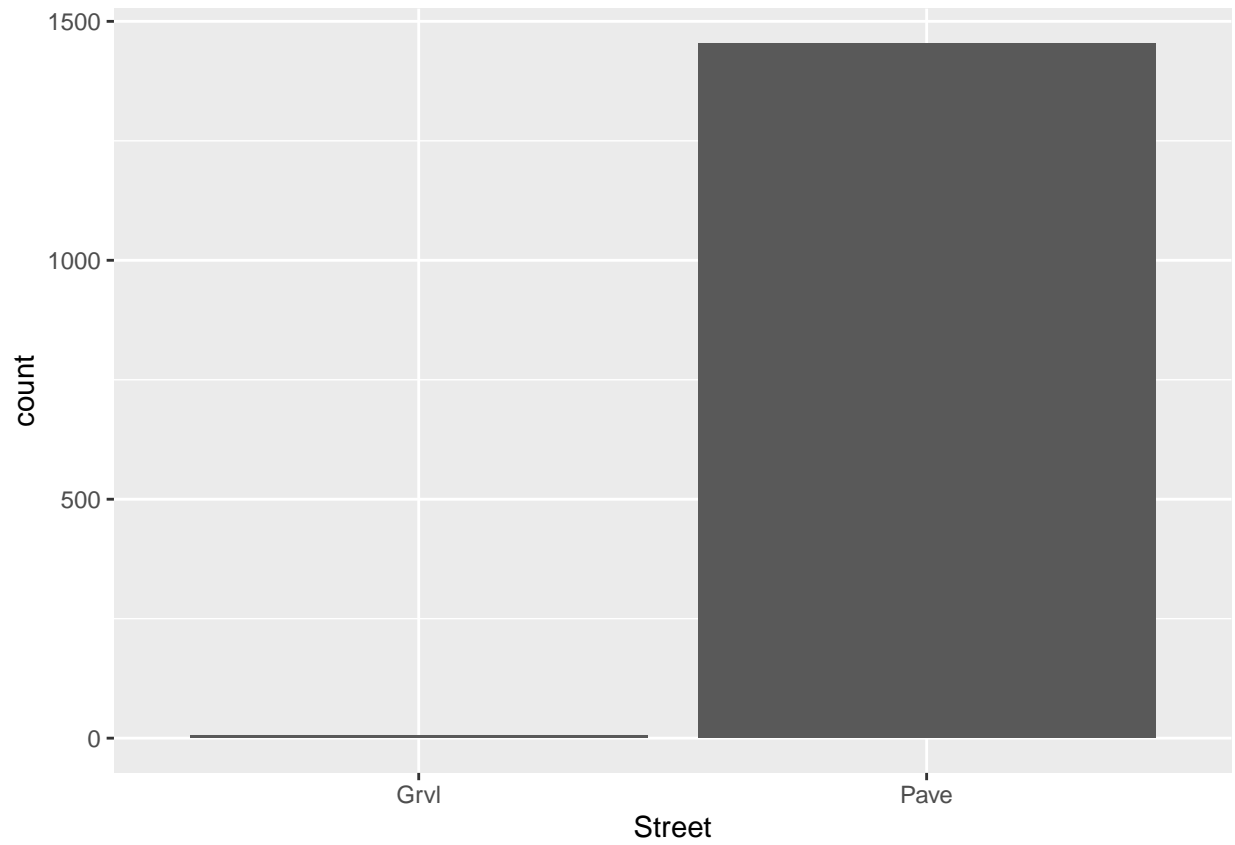
```
for (col in c(names(dat))) {
  g <- ggplot(data=dat, aes_string(x=col))
  if (is.numeric(dat[[col]])) {
    g <- g + geom_histogram(bins=30)
  } else {
    g <- g + geom_bar()
  }
  print(g)
}
```

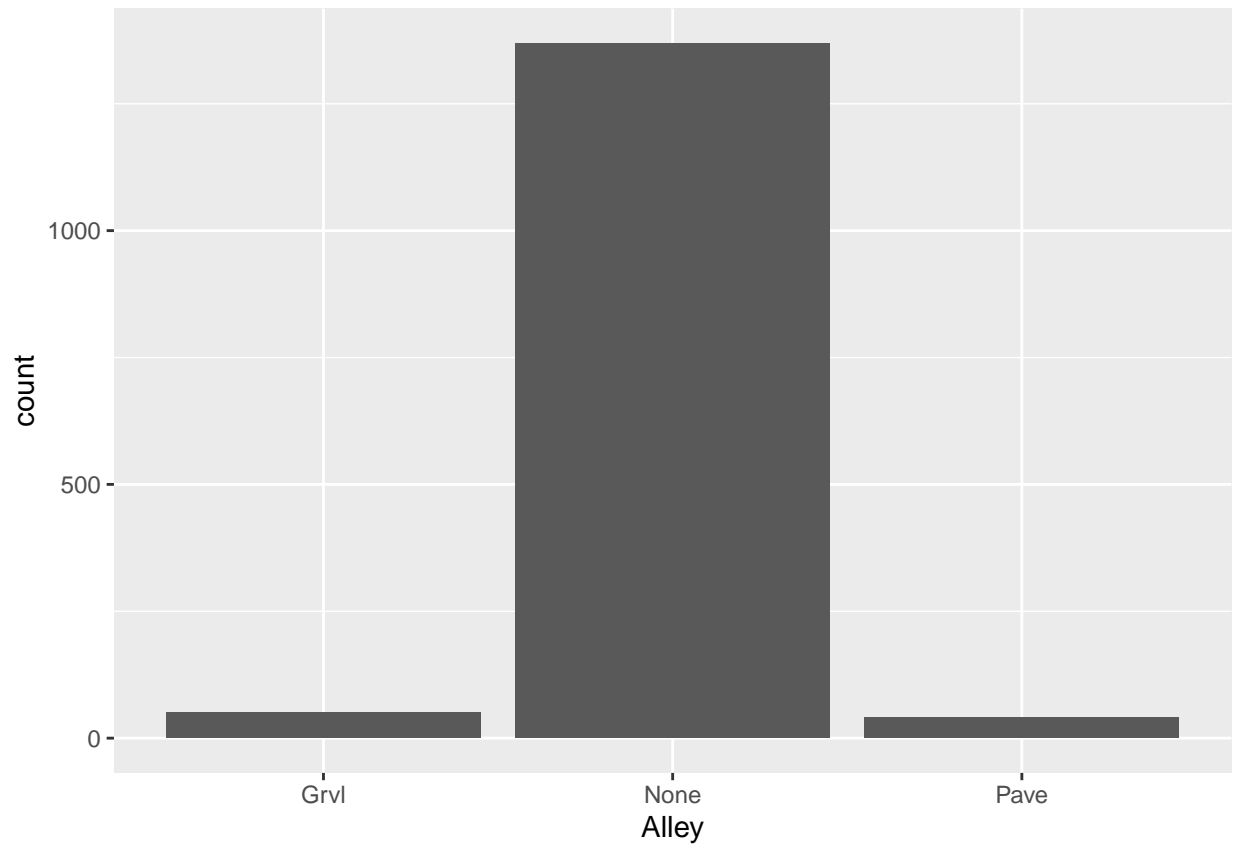


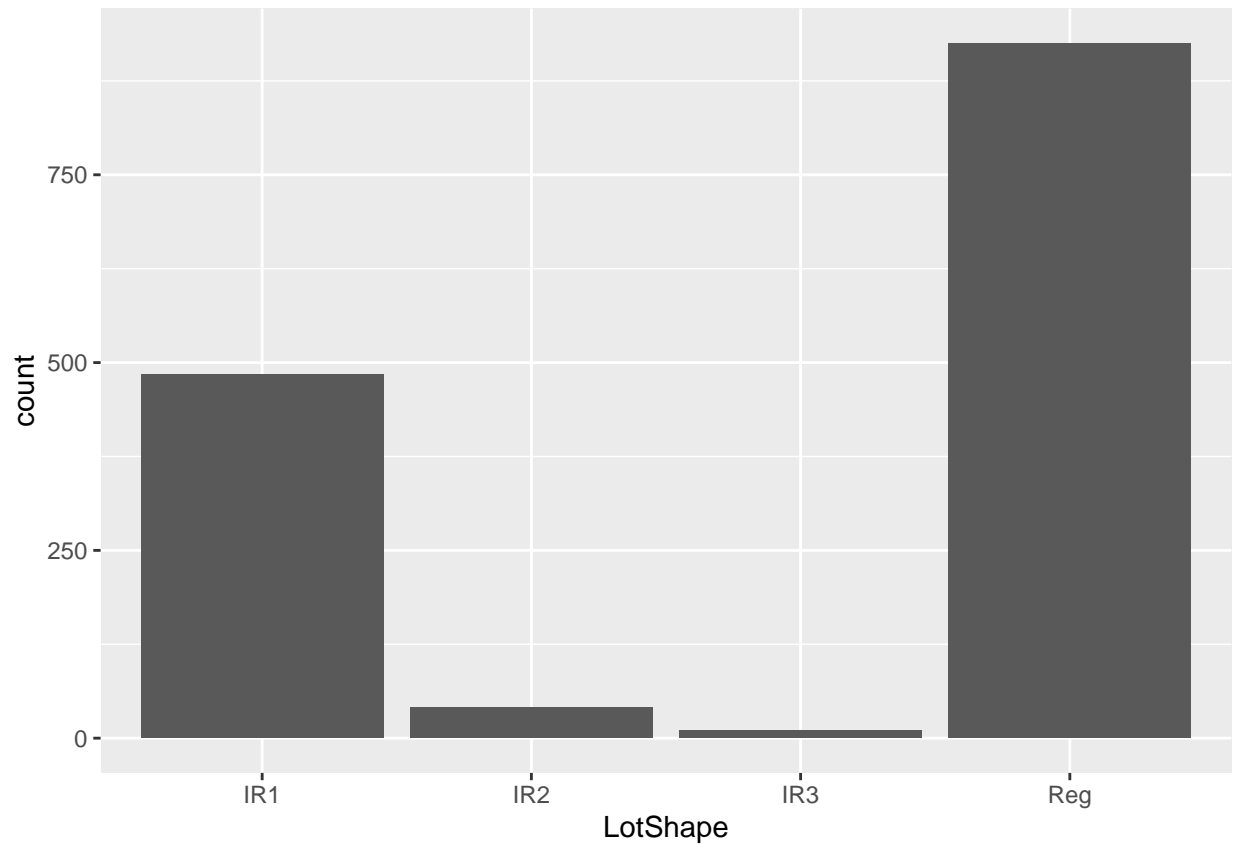


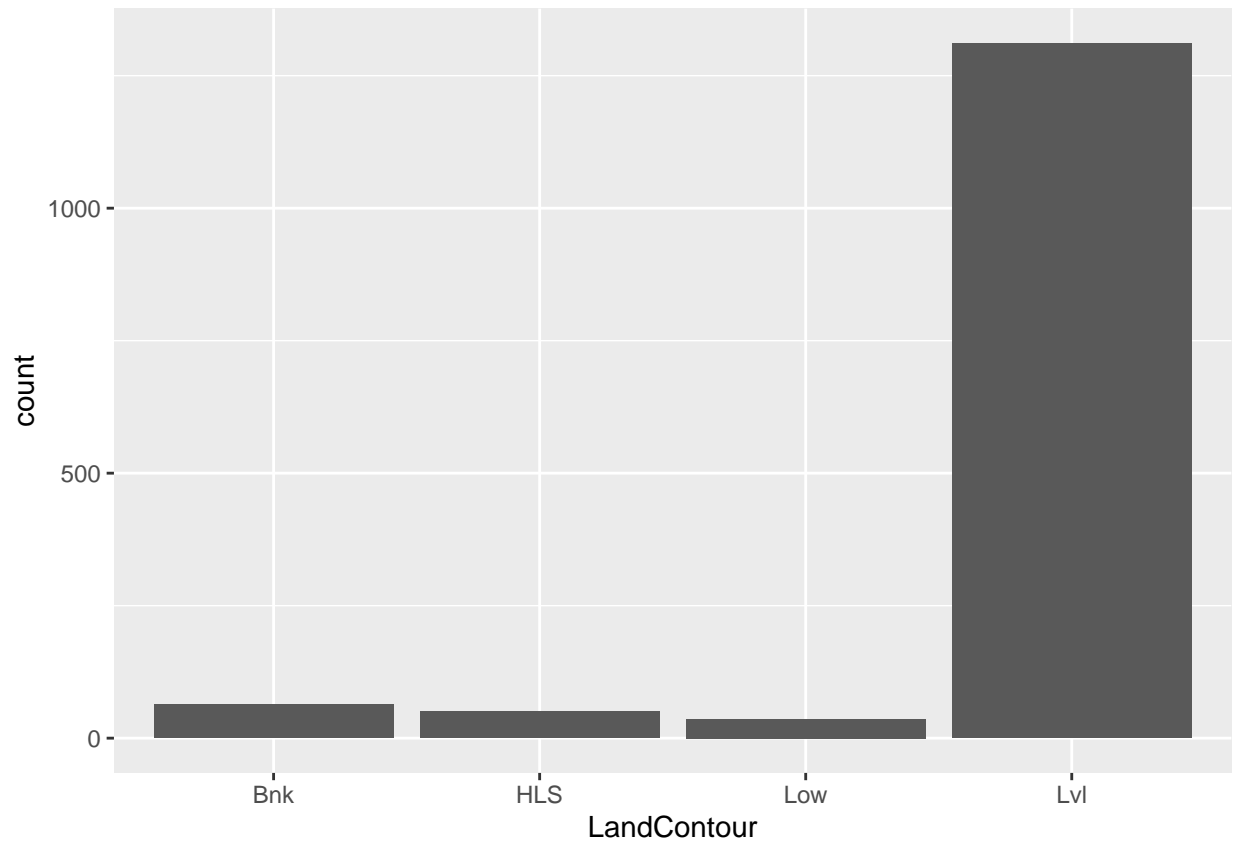


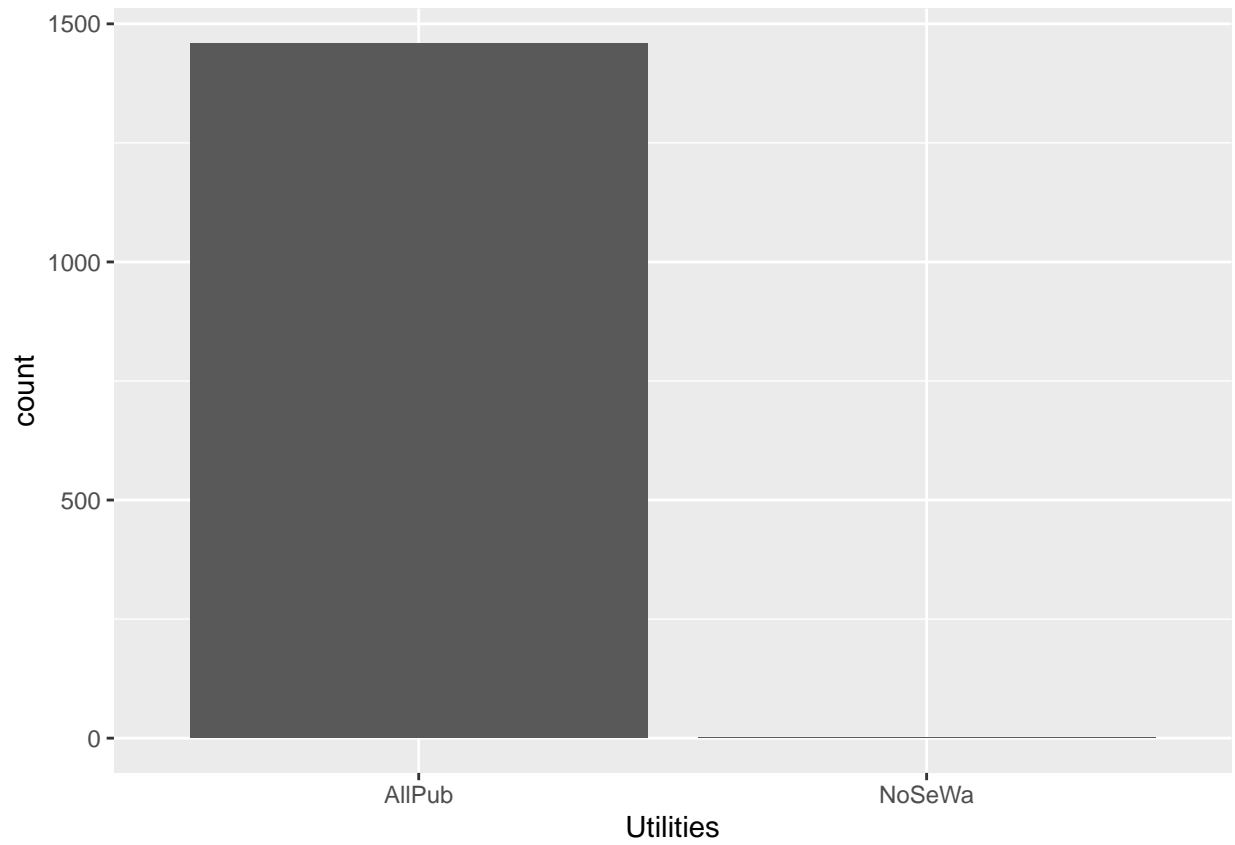


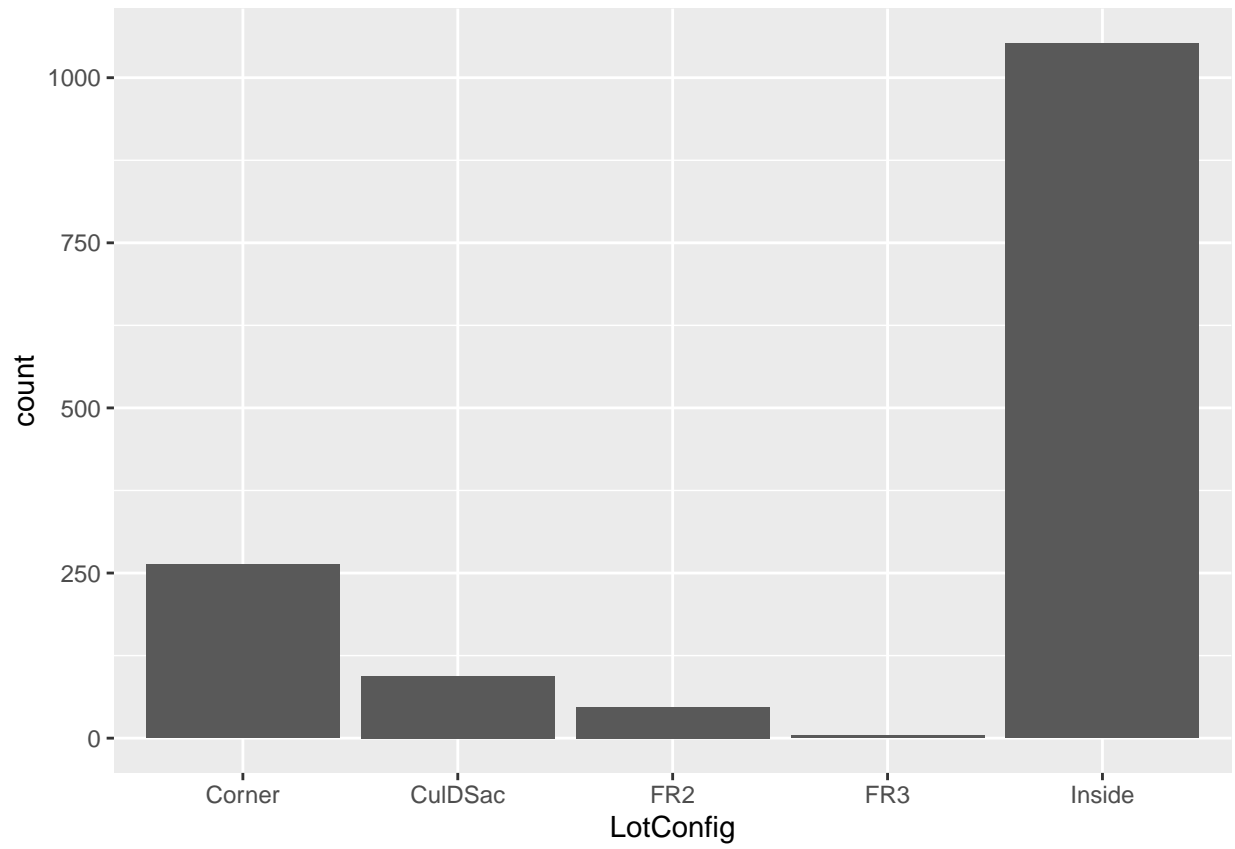


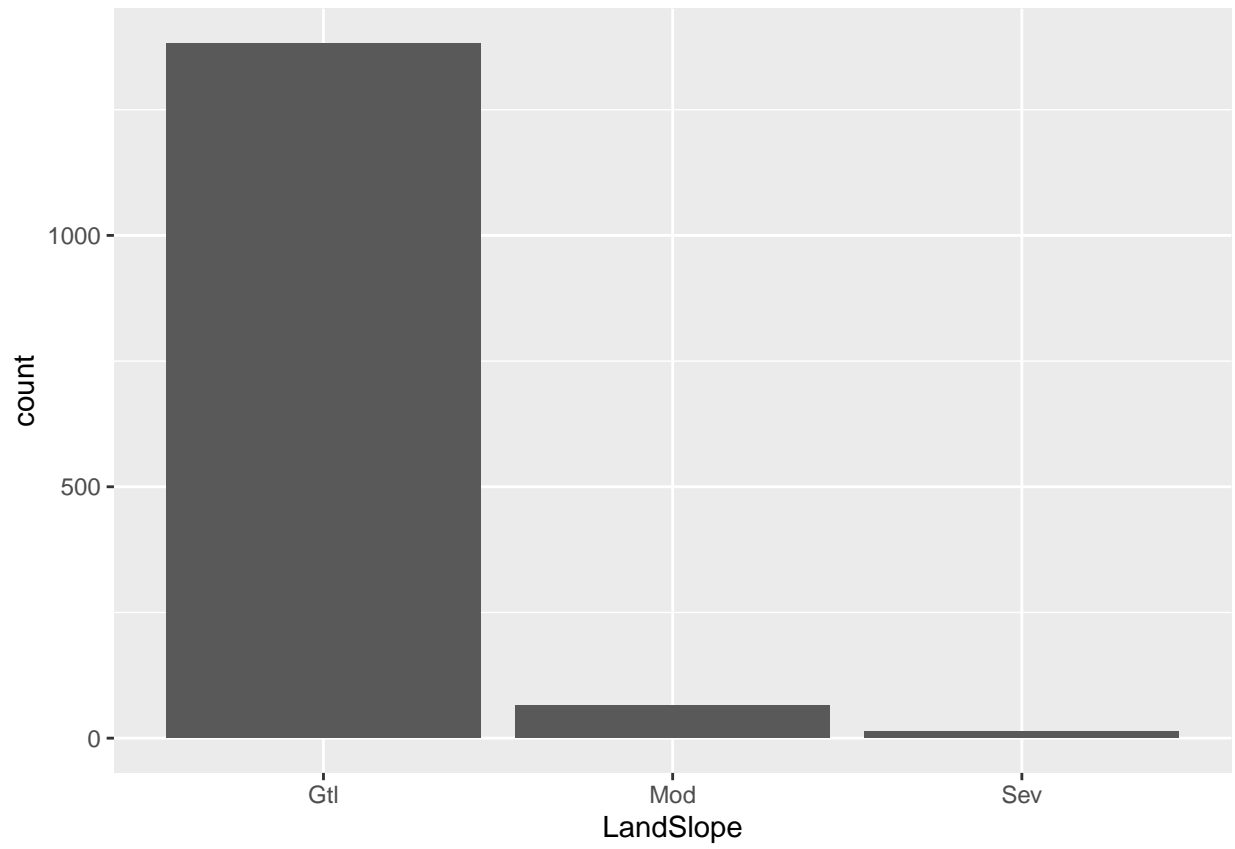


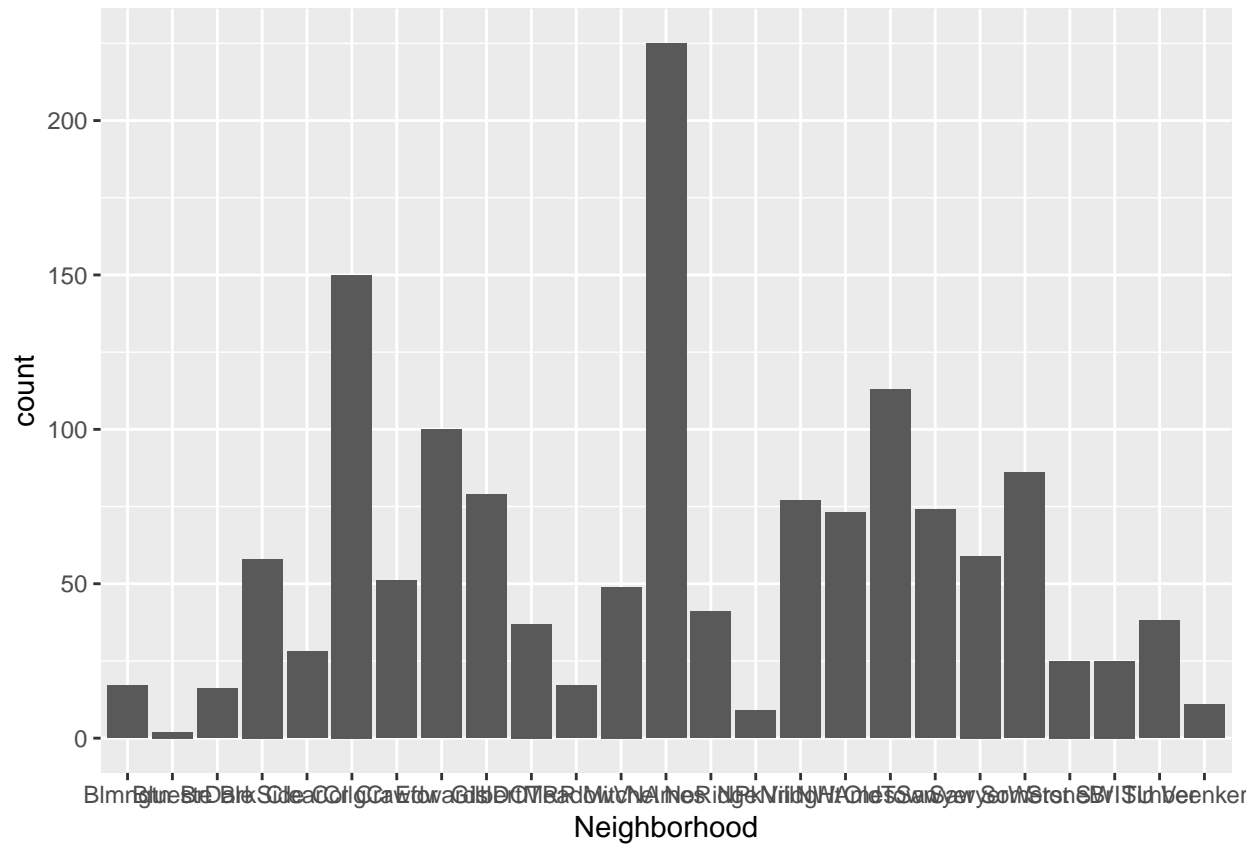


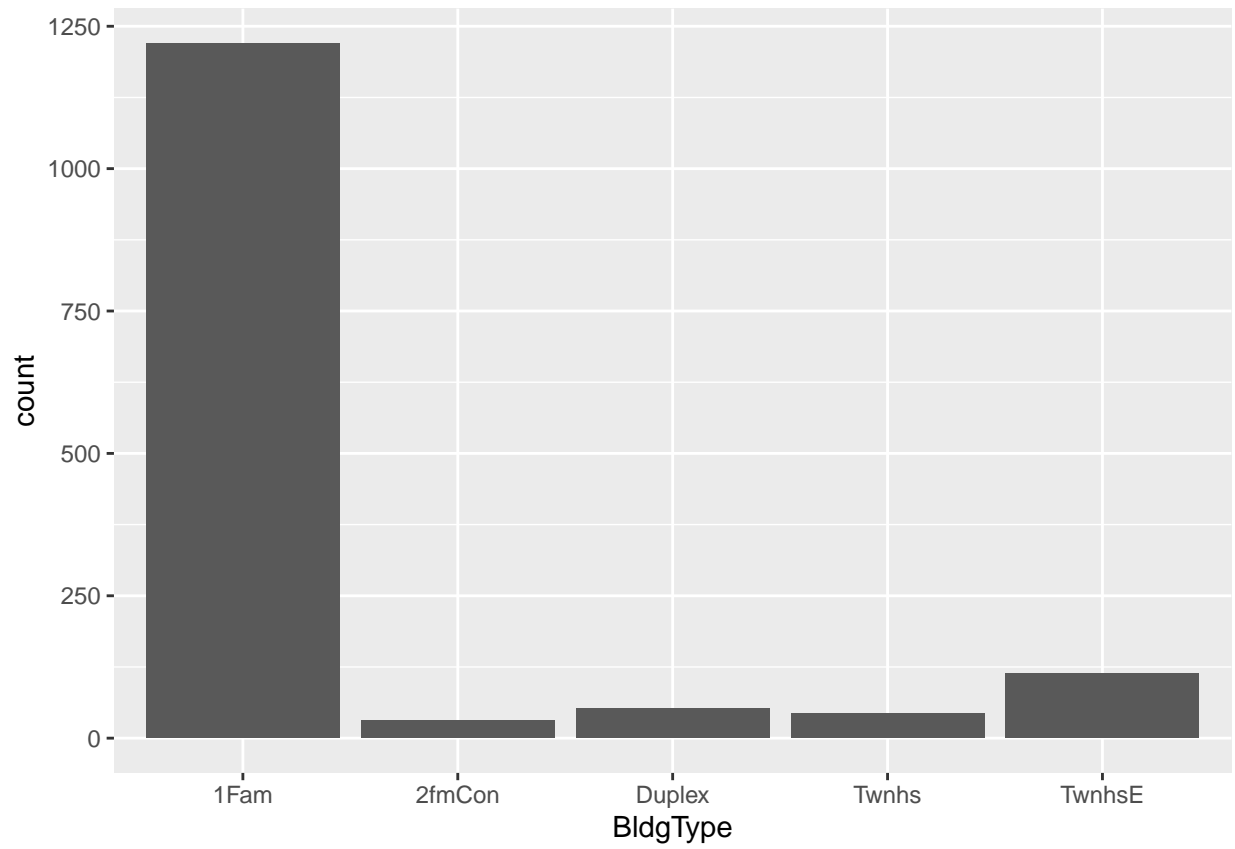


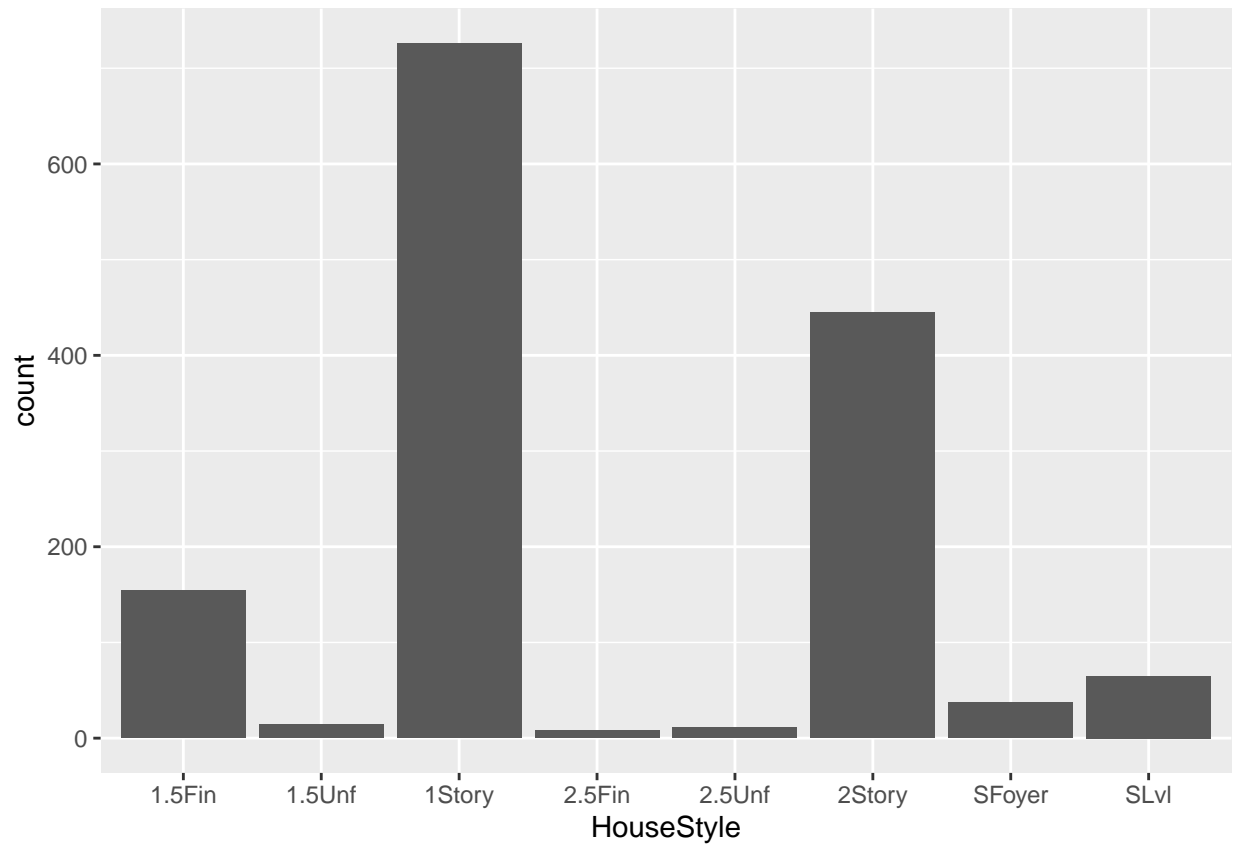


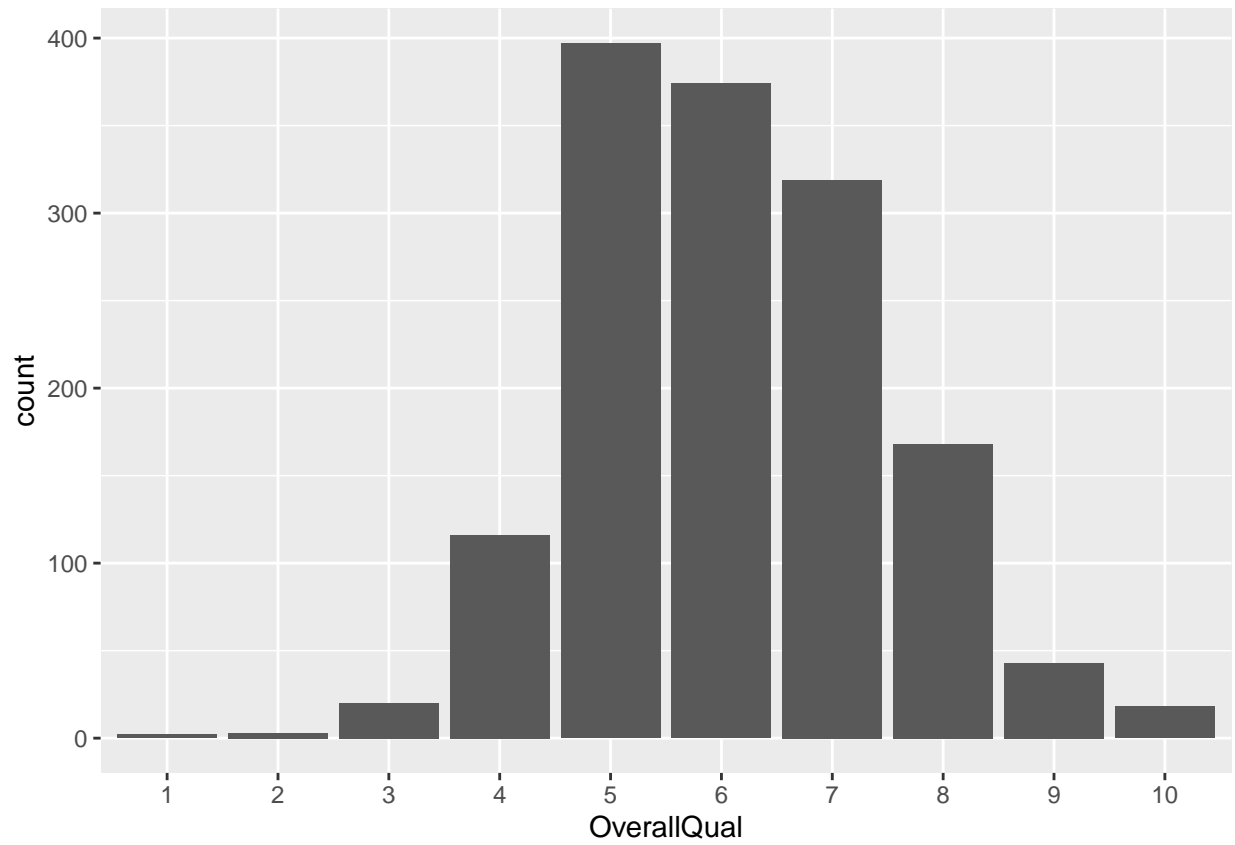


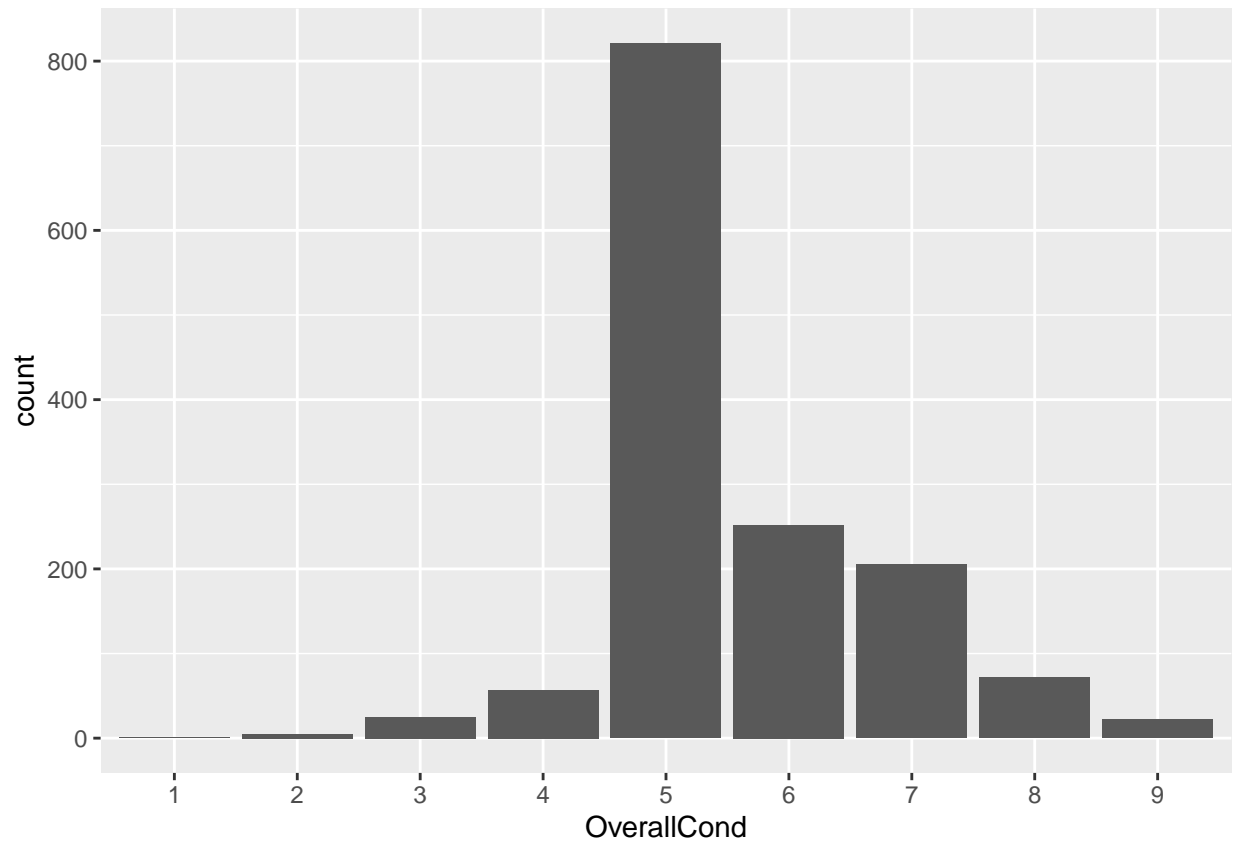


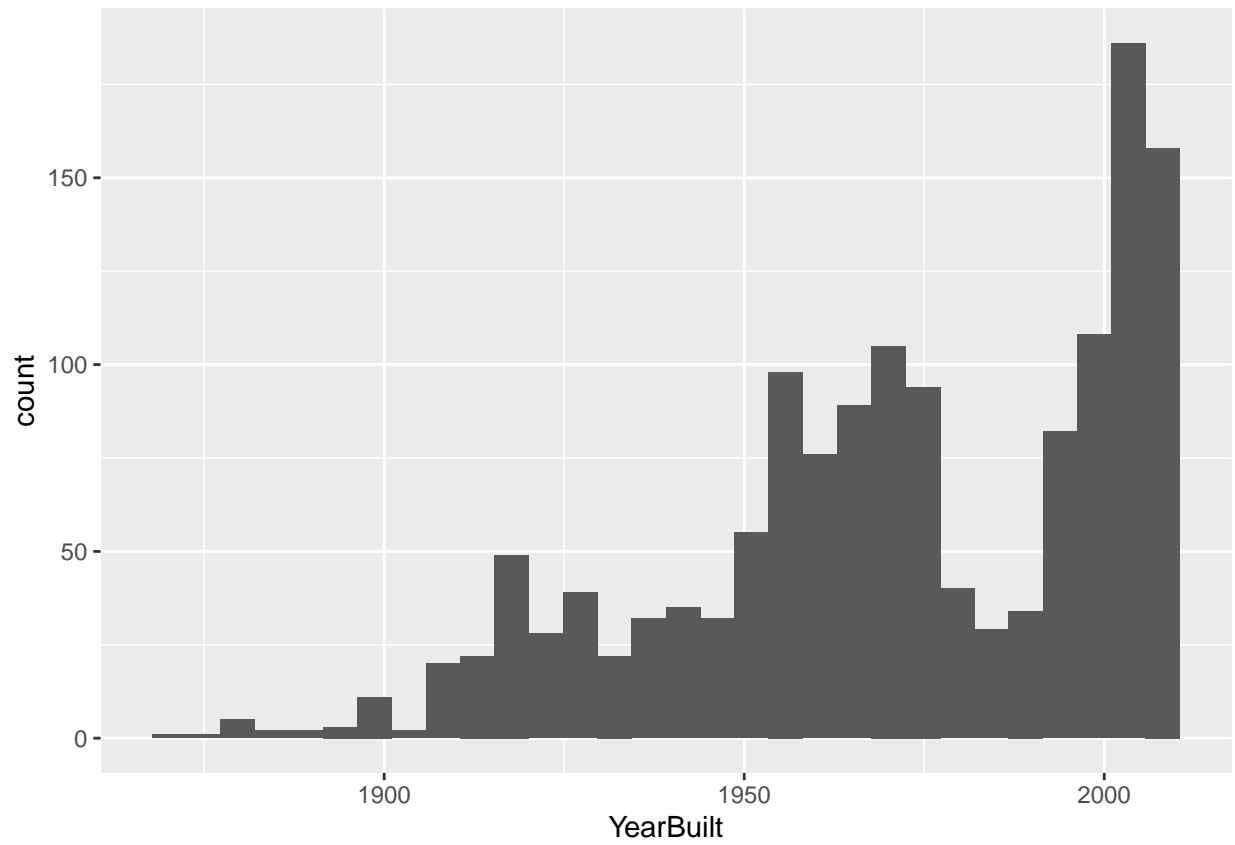


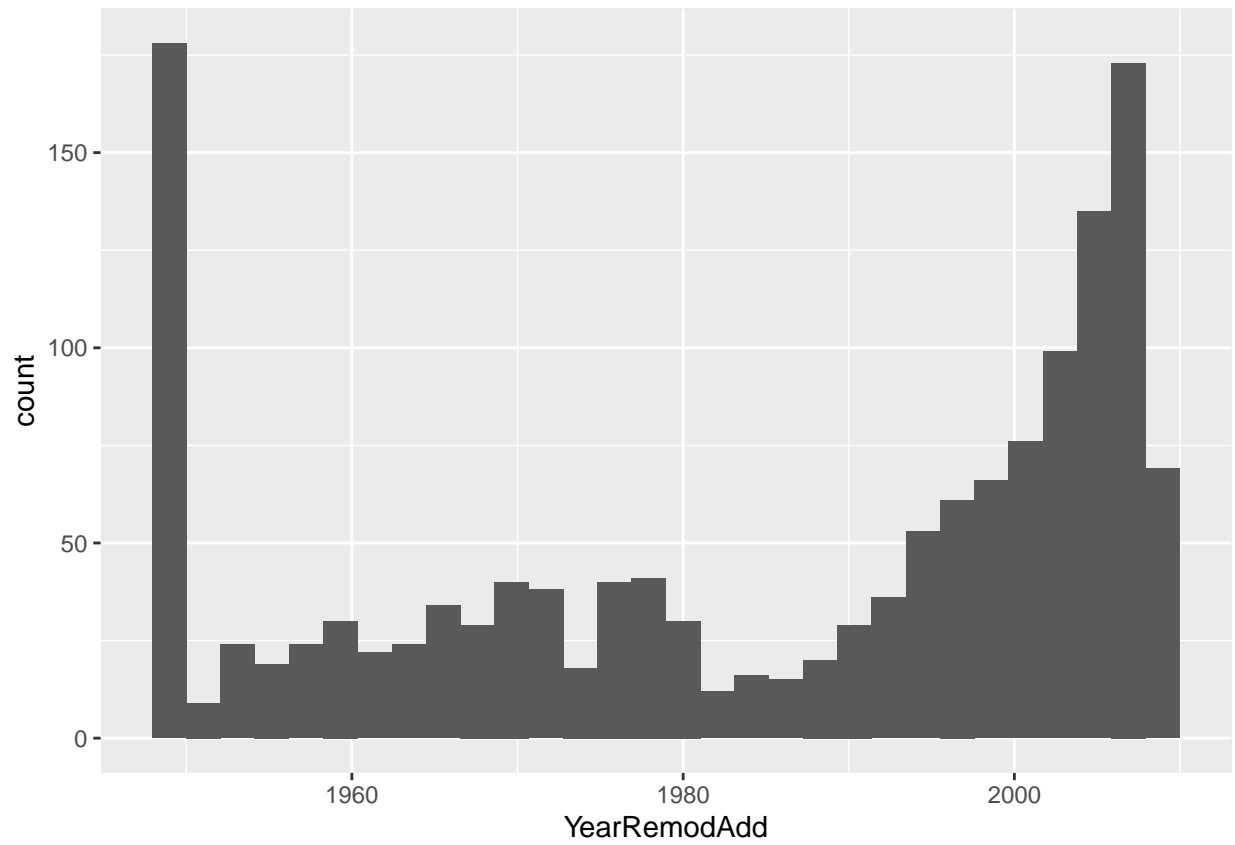


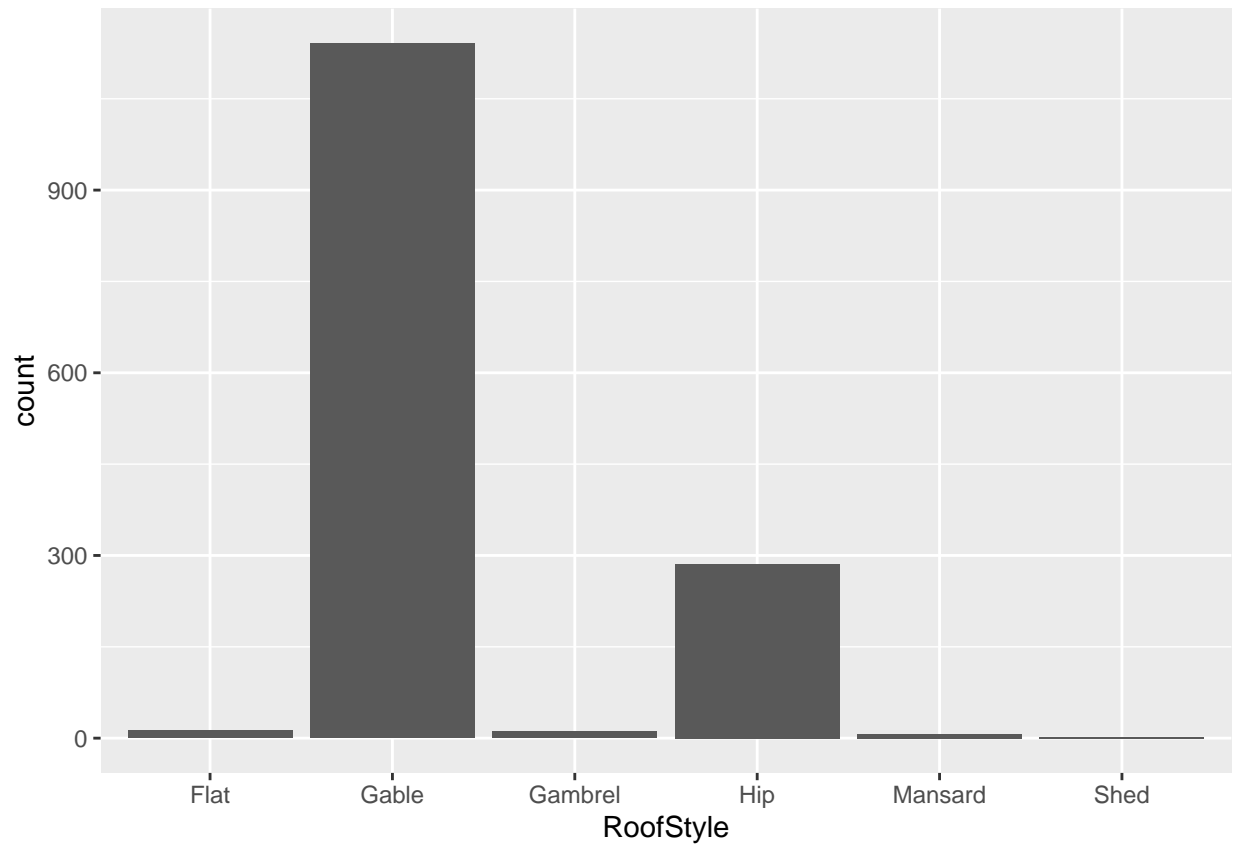


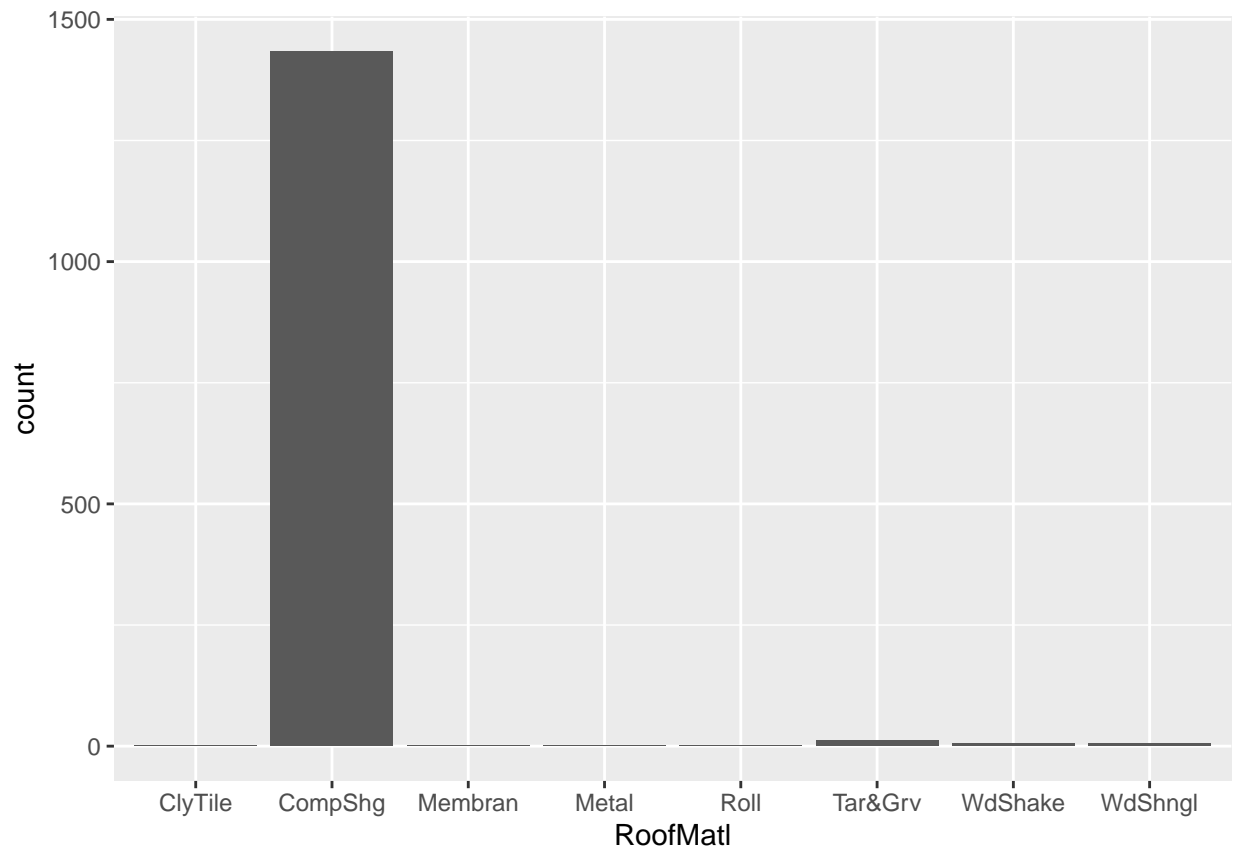


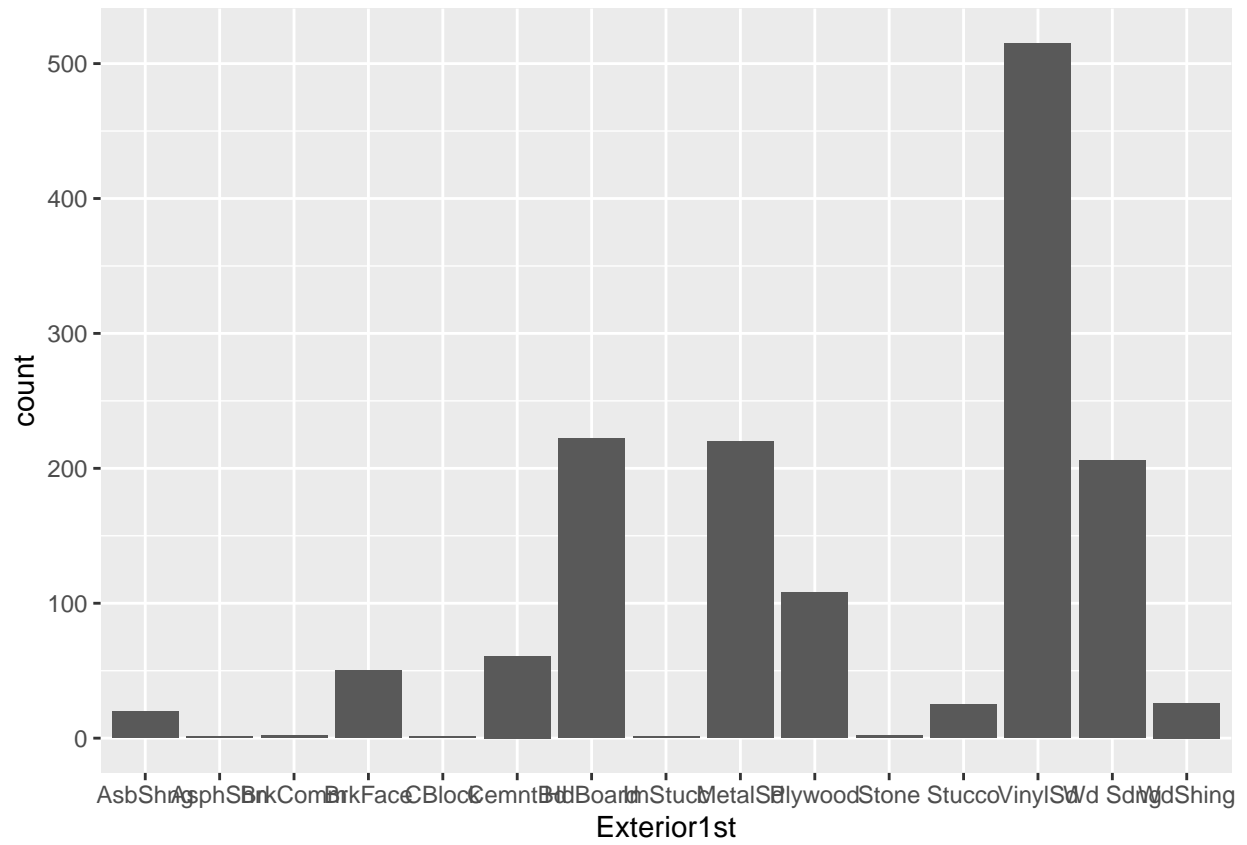


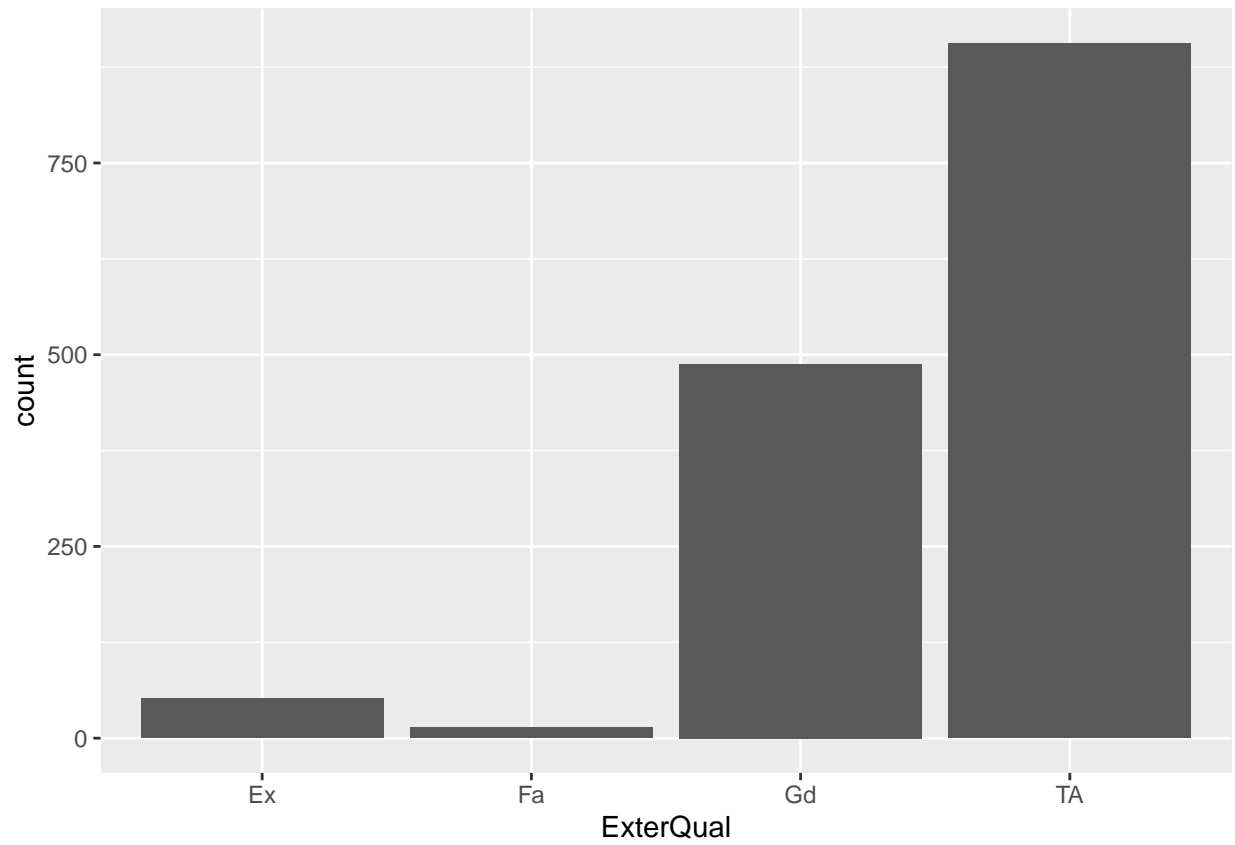


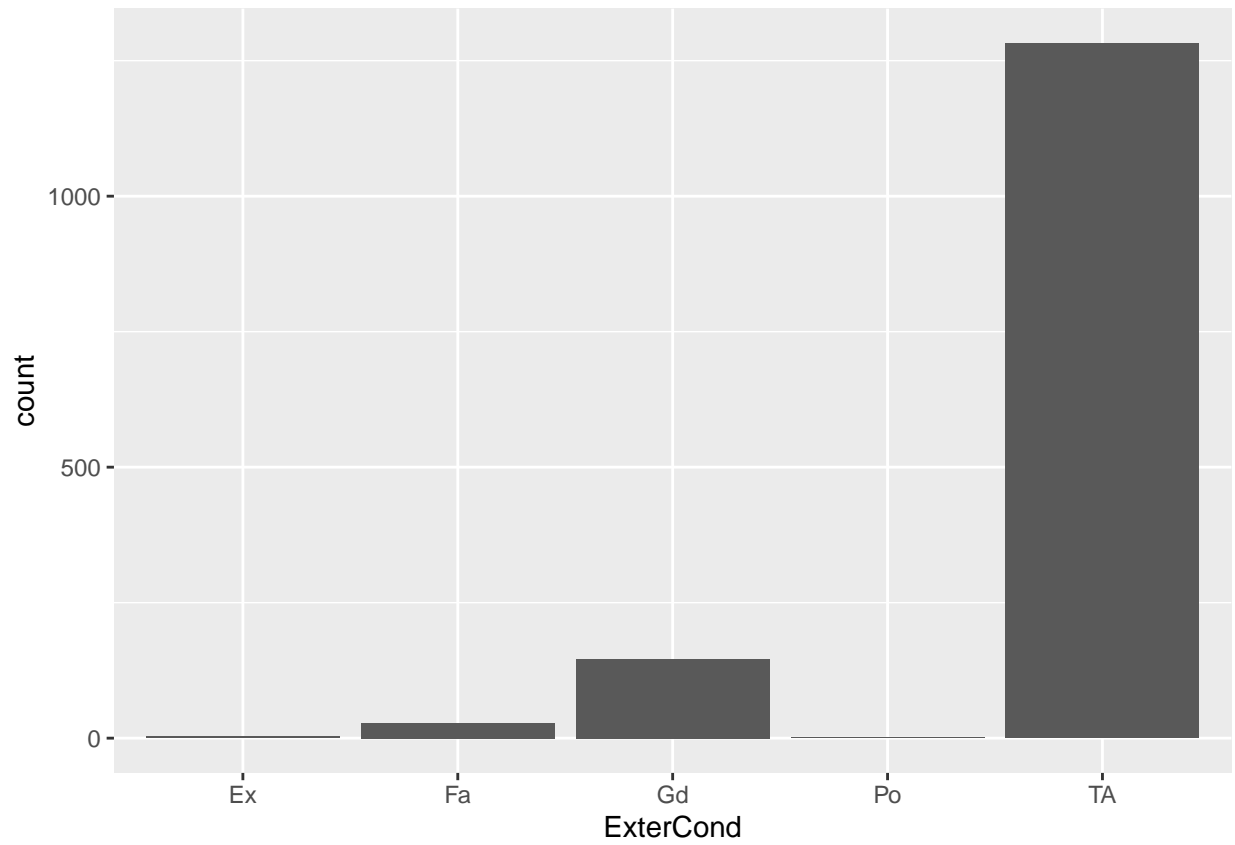


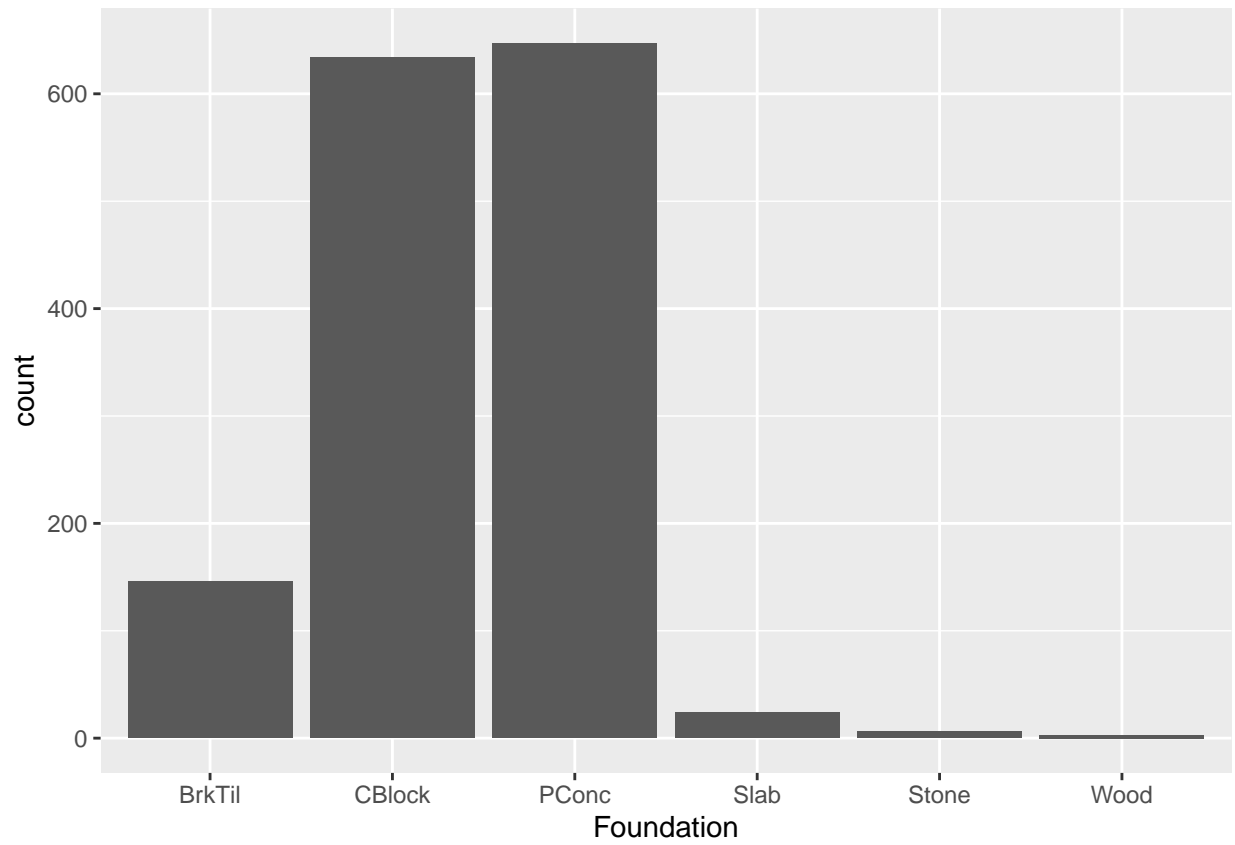


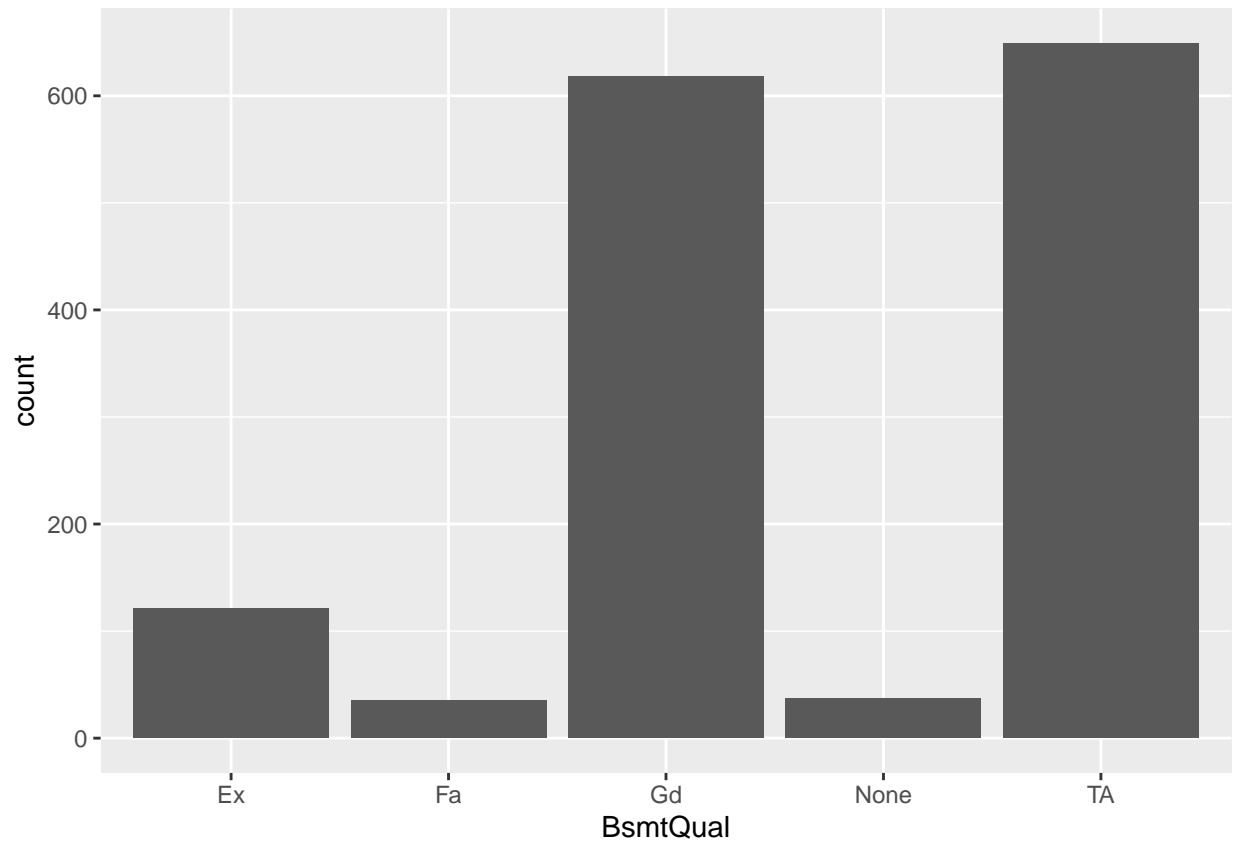


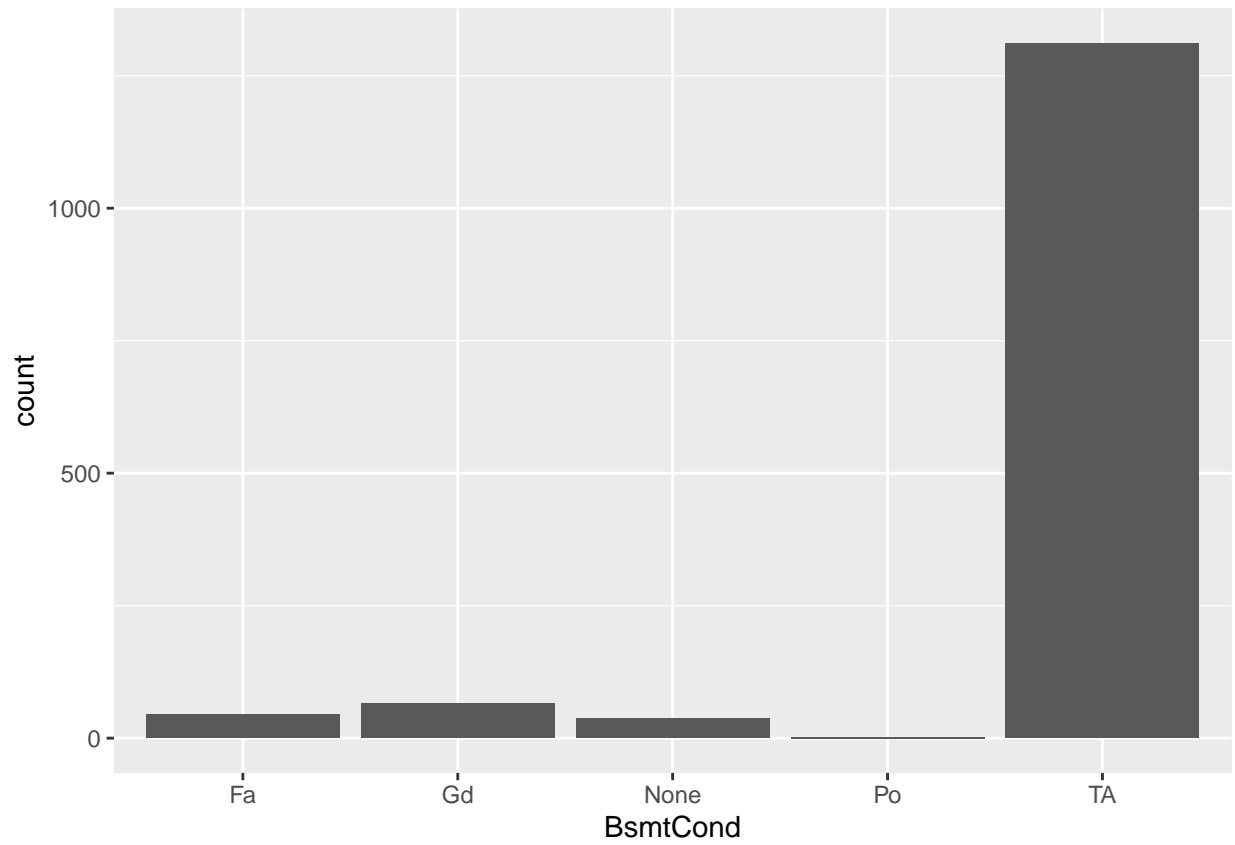


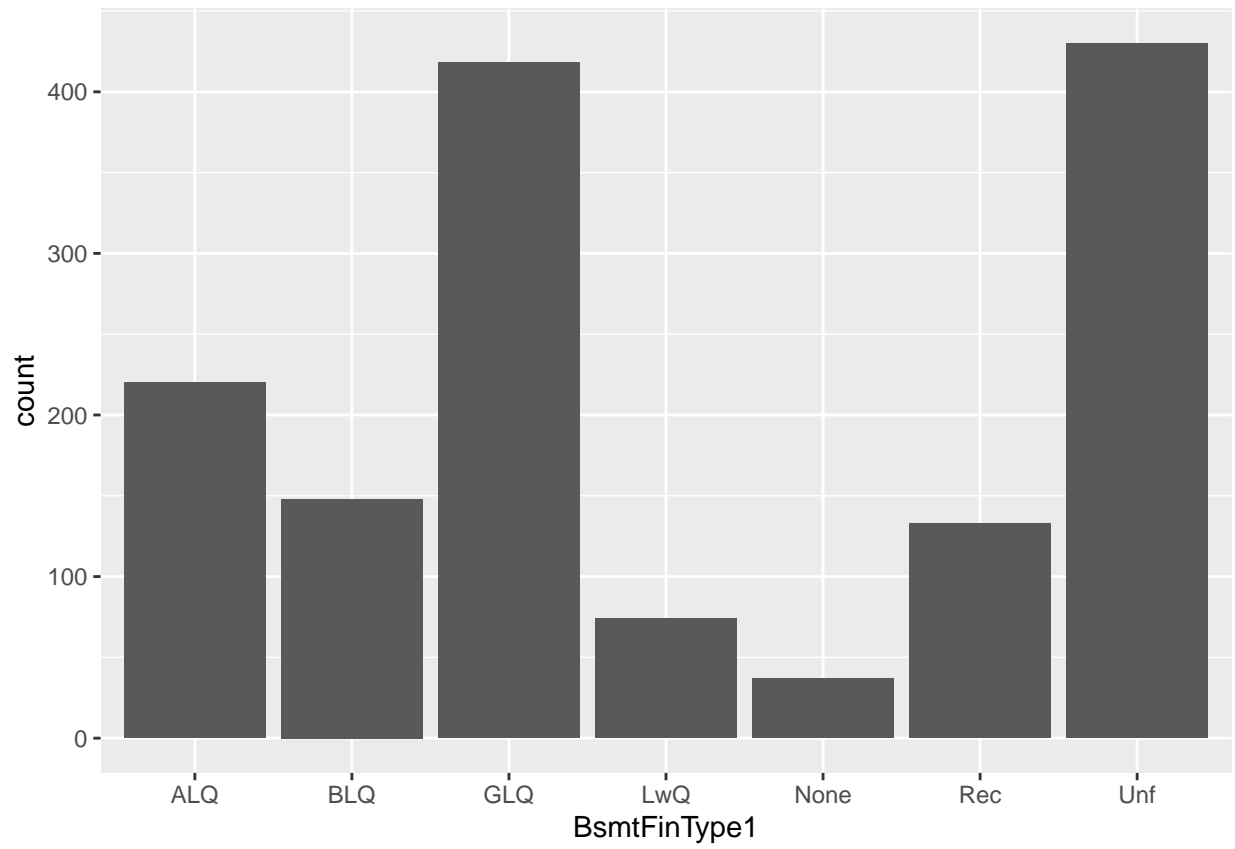


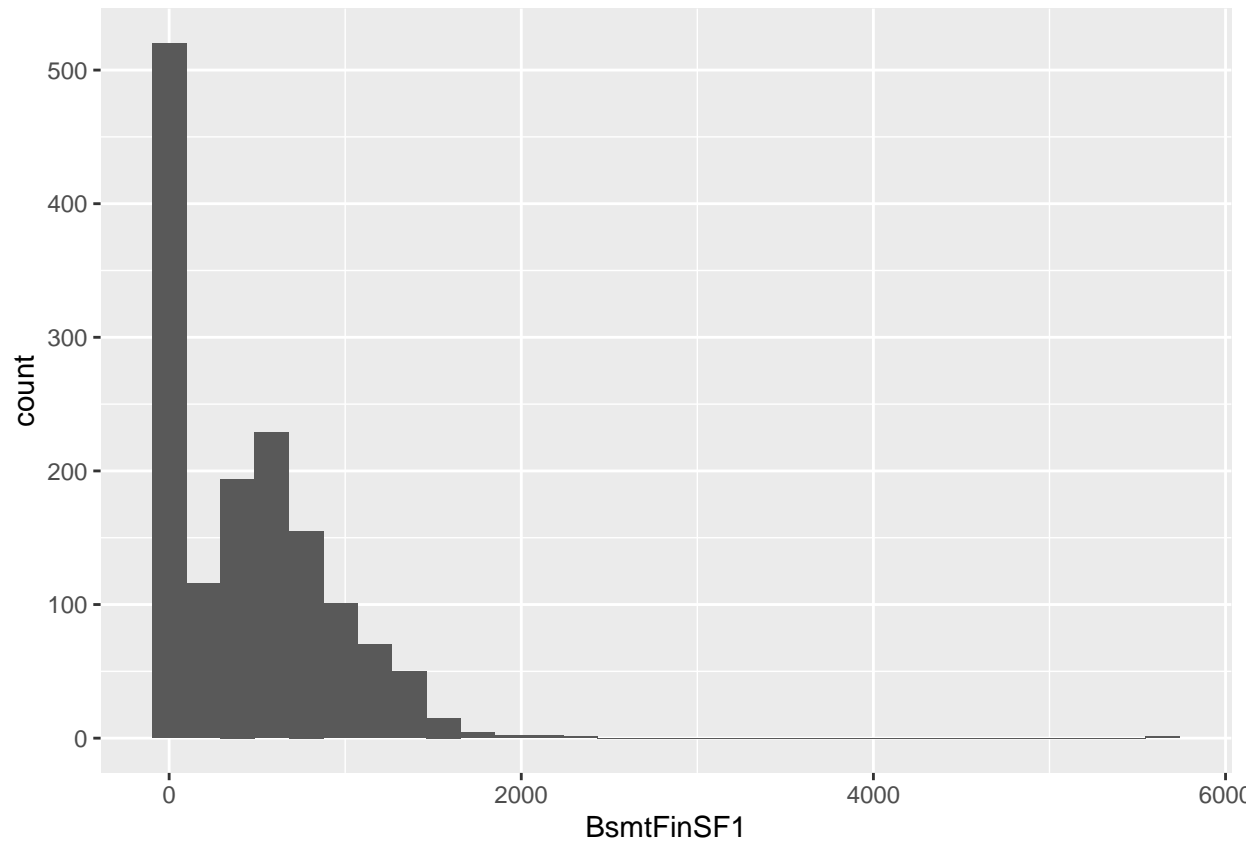


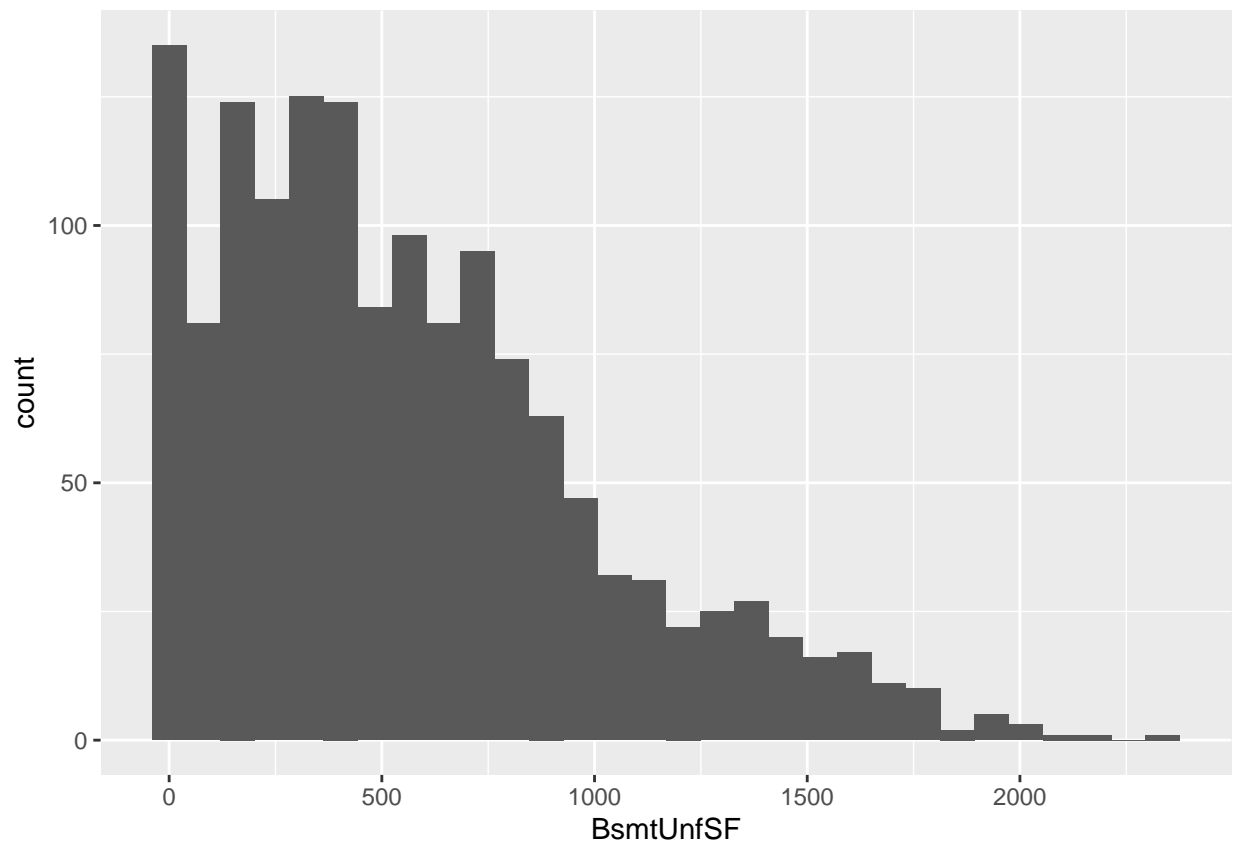


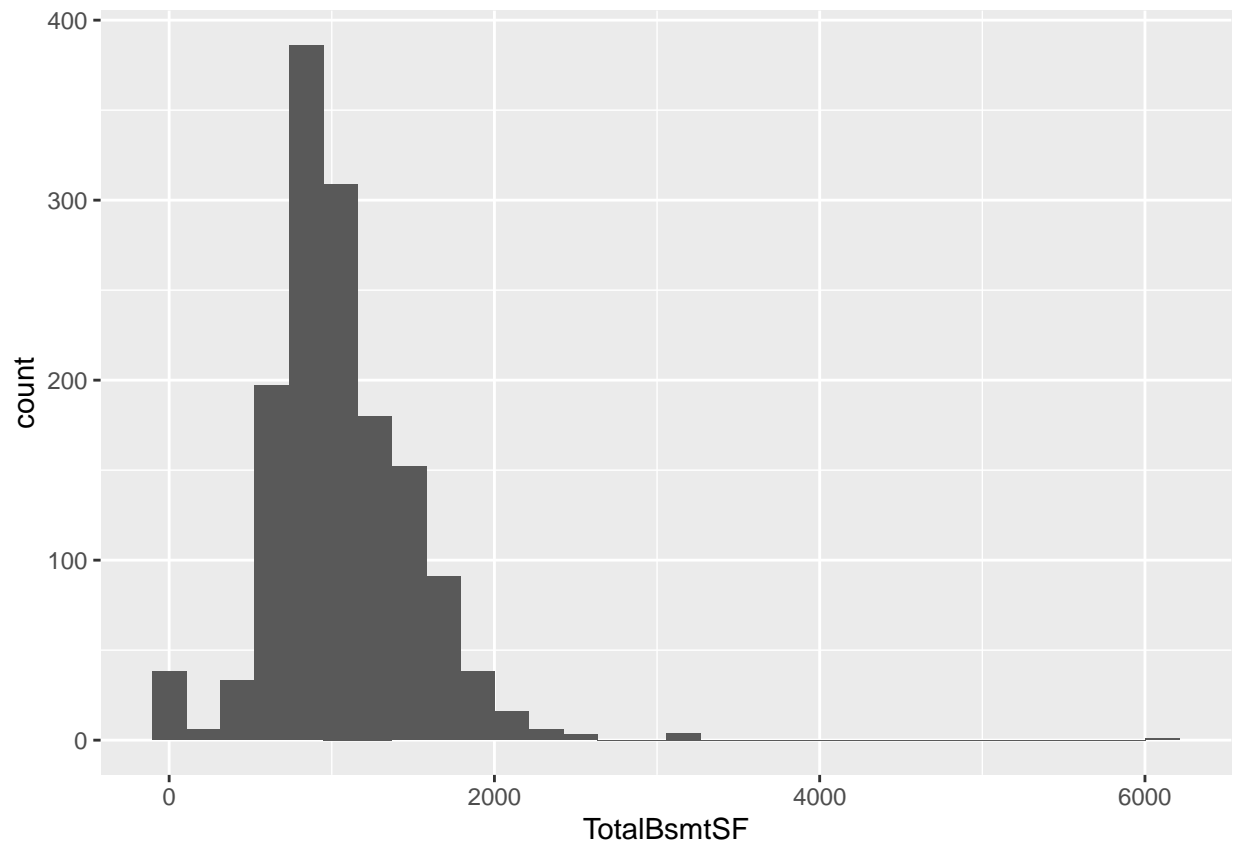


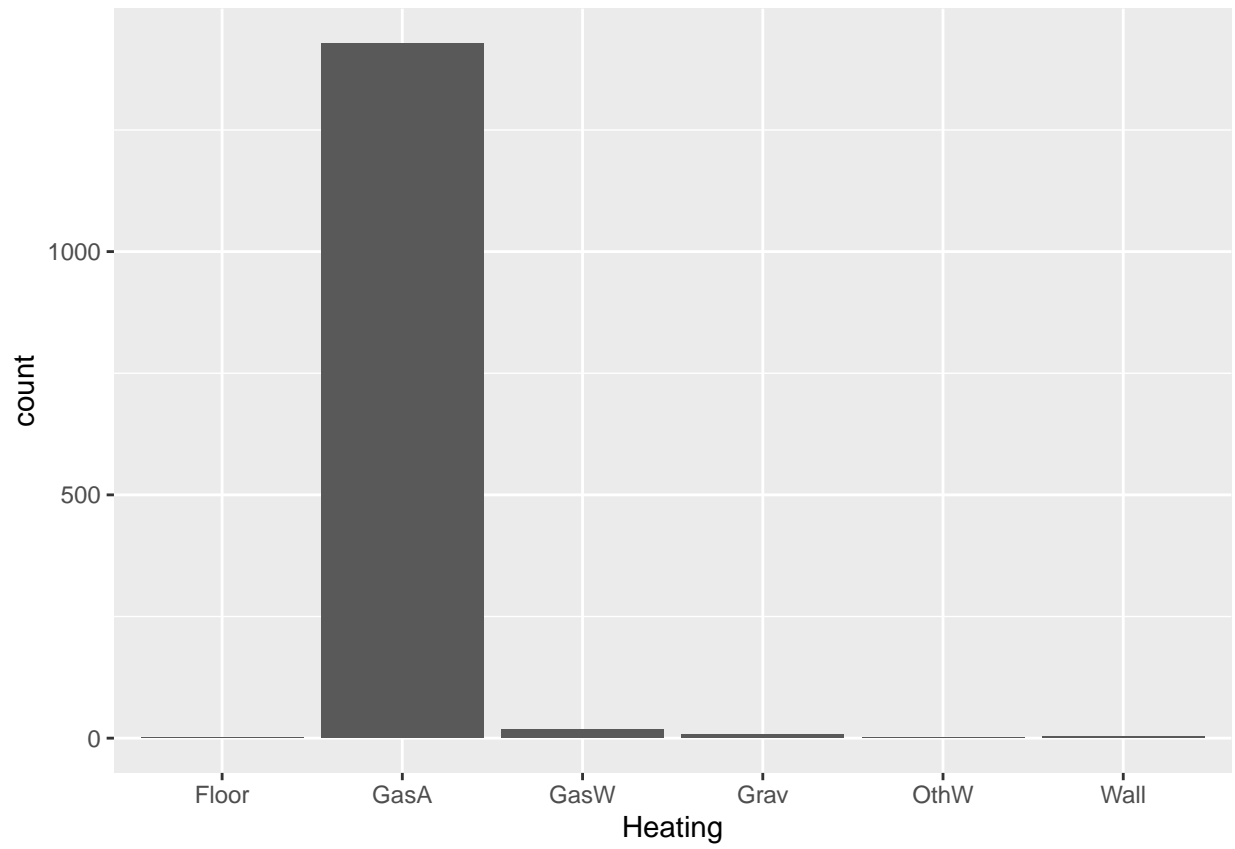


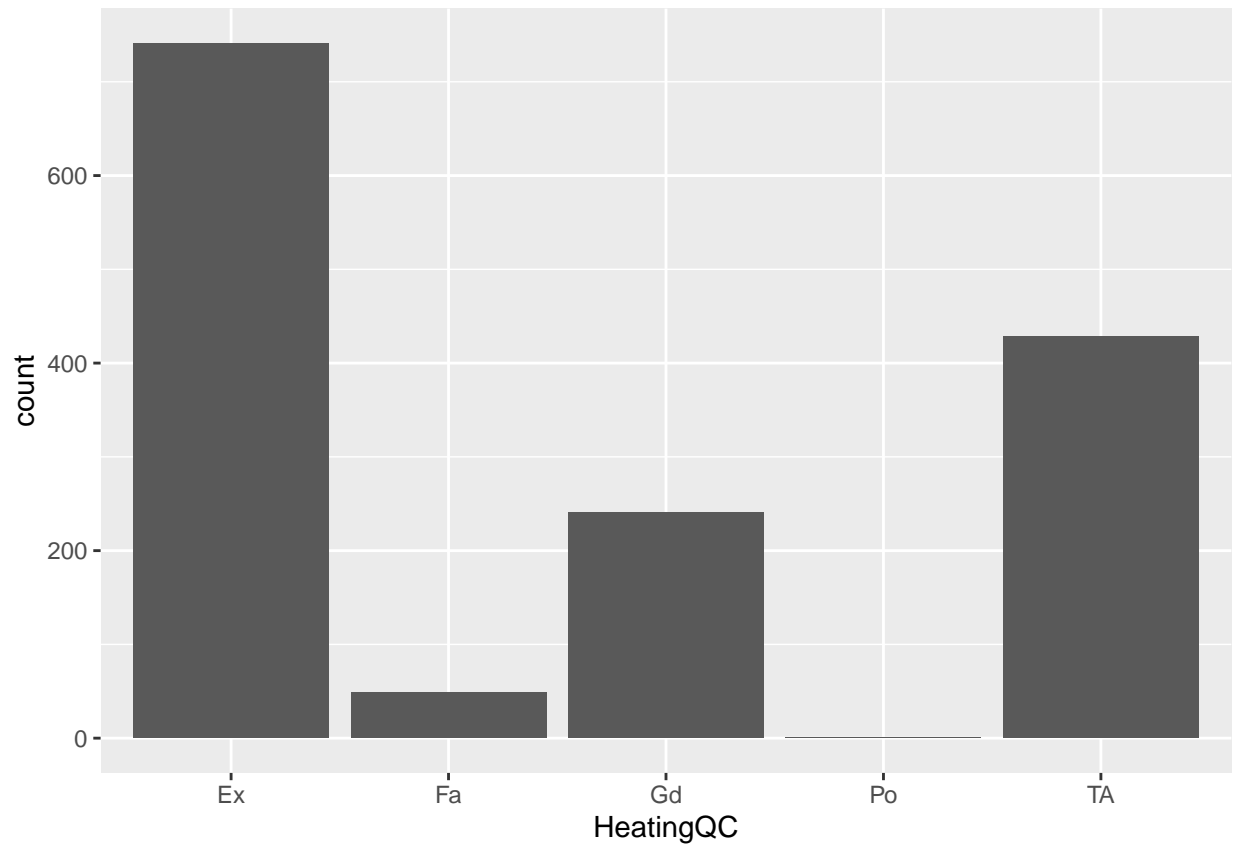


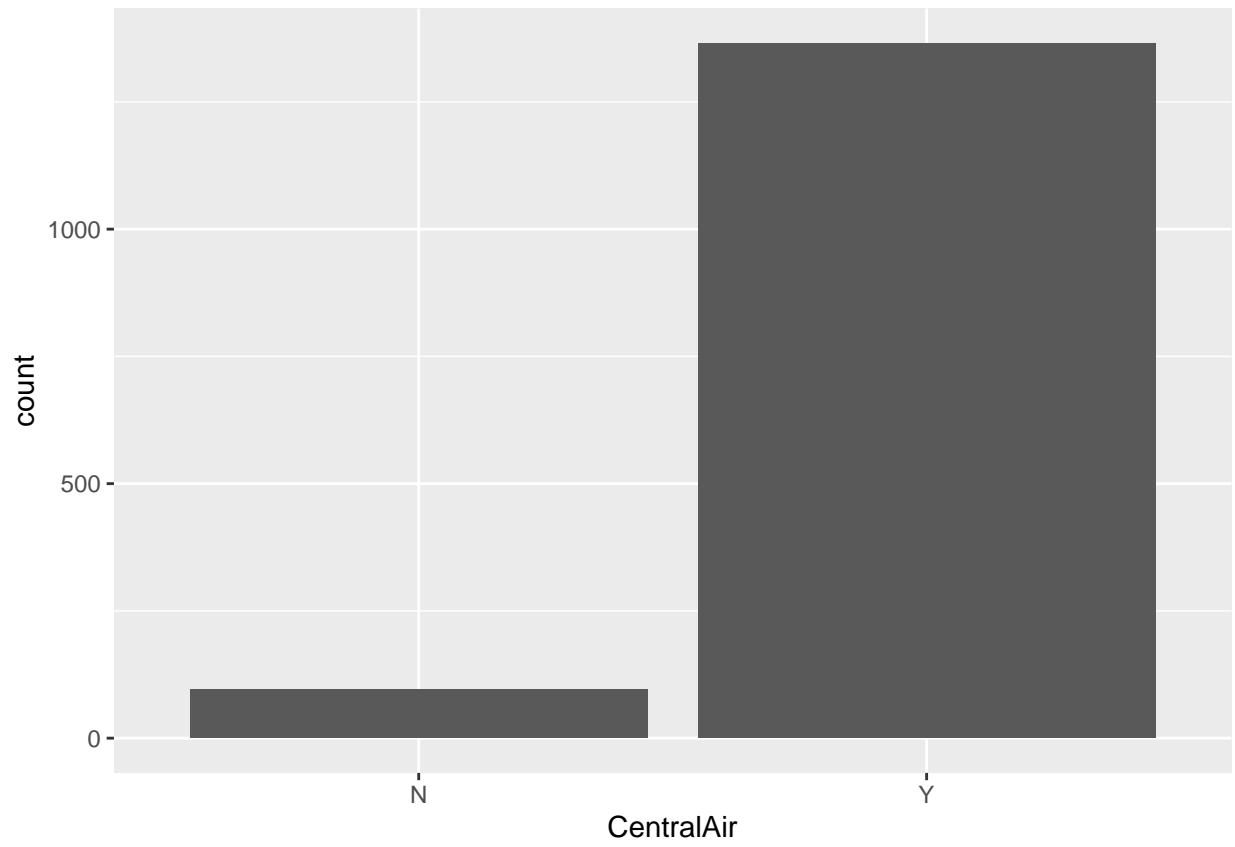


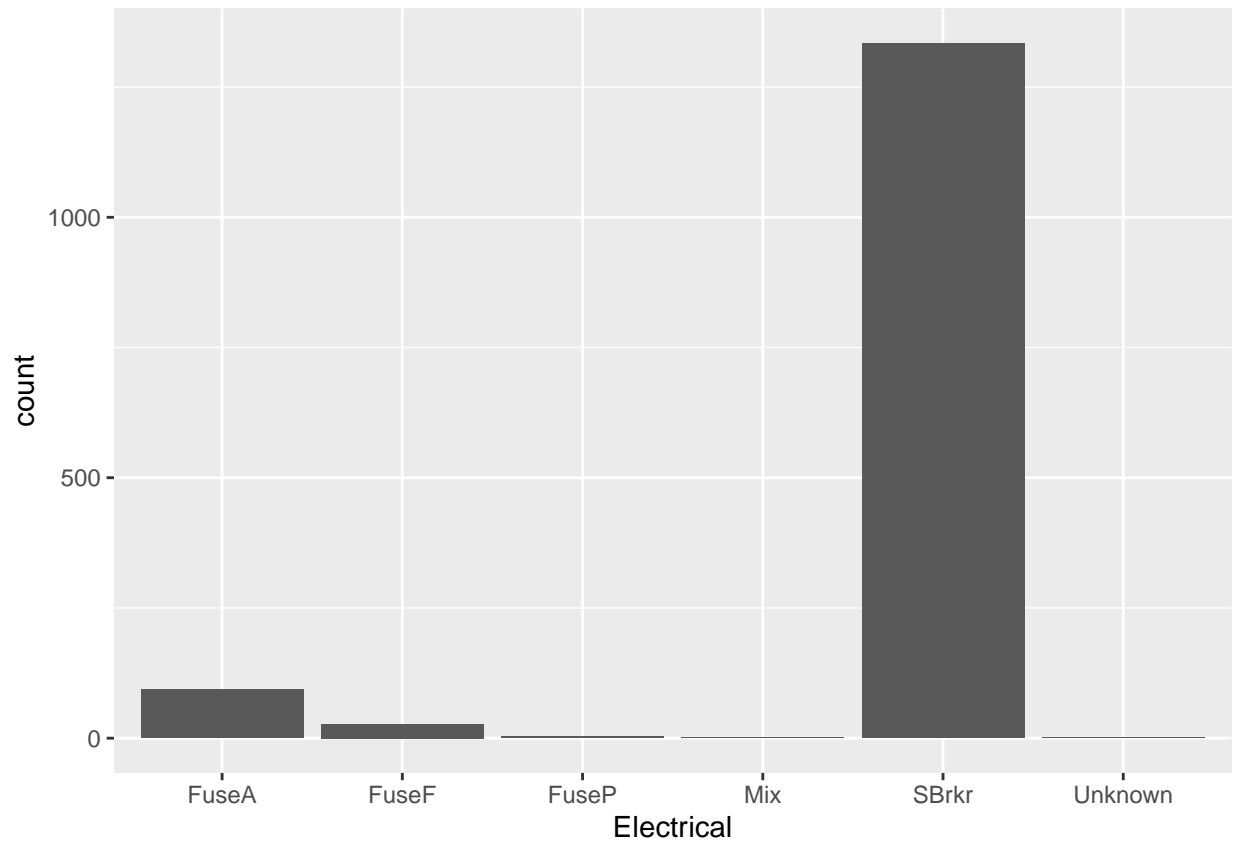


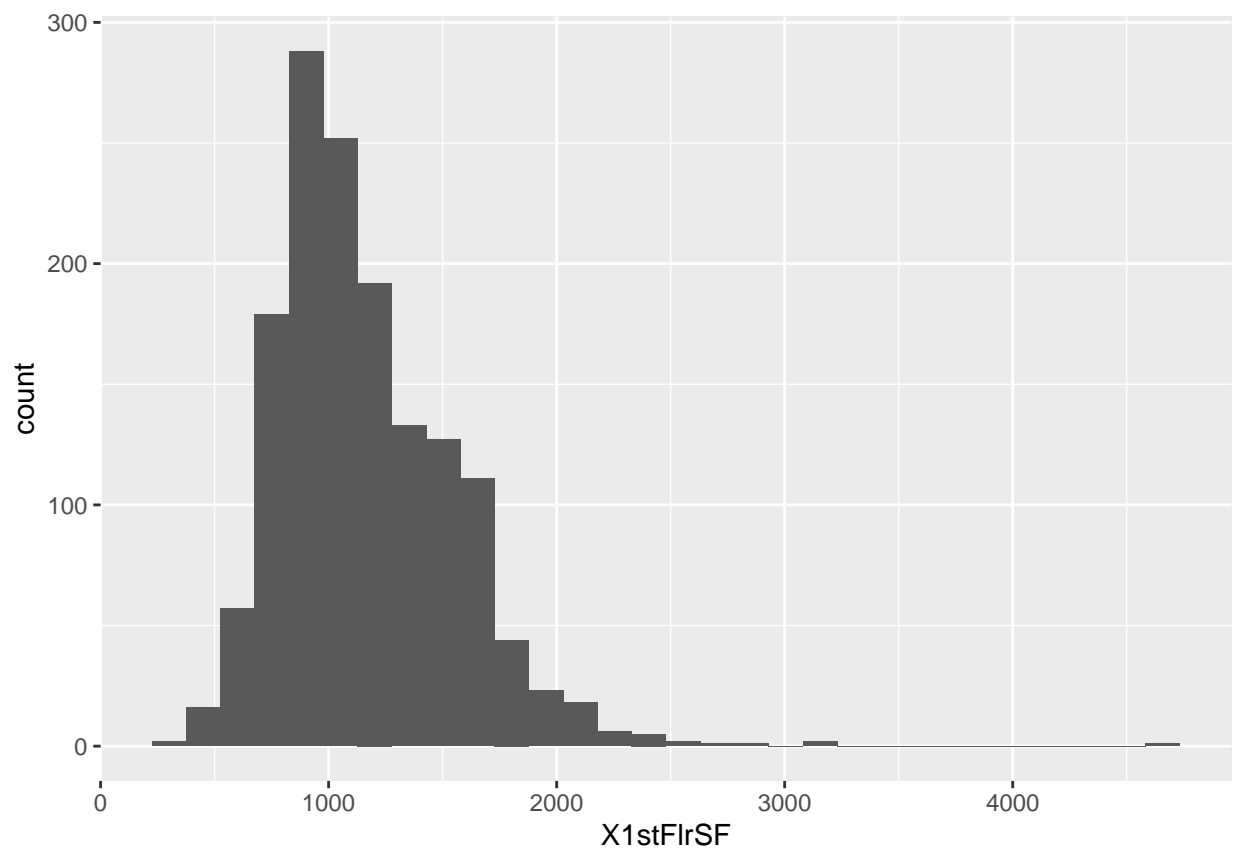


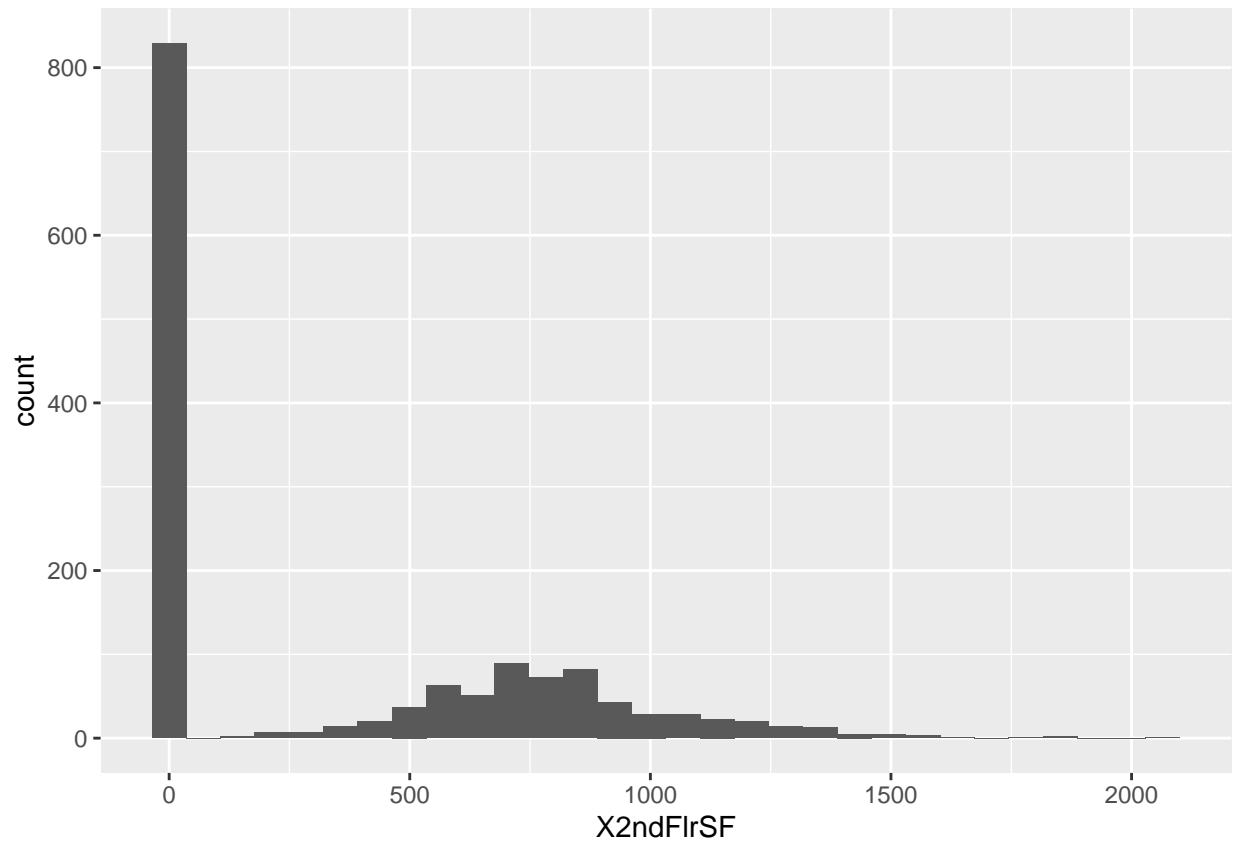


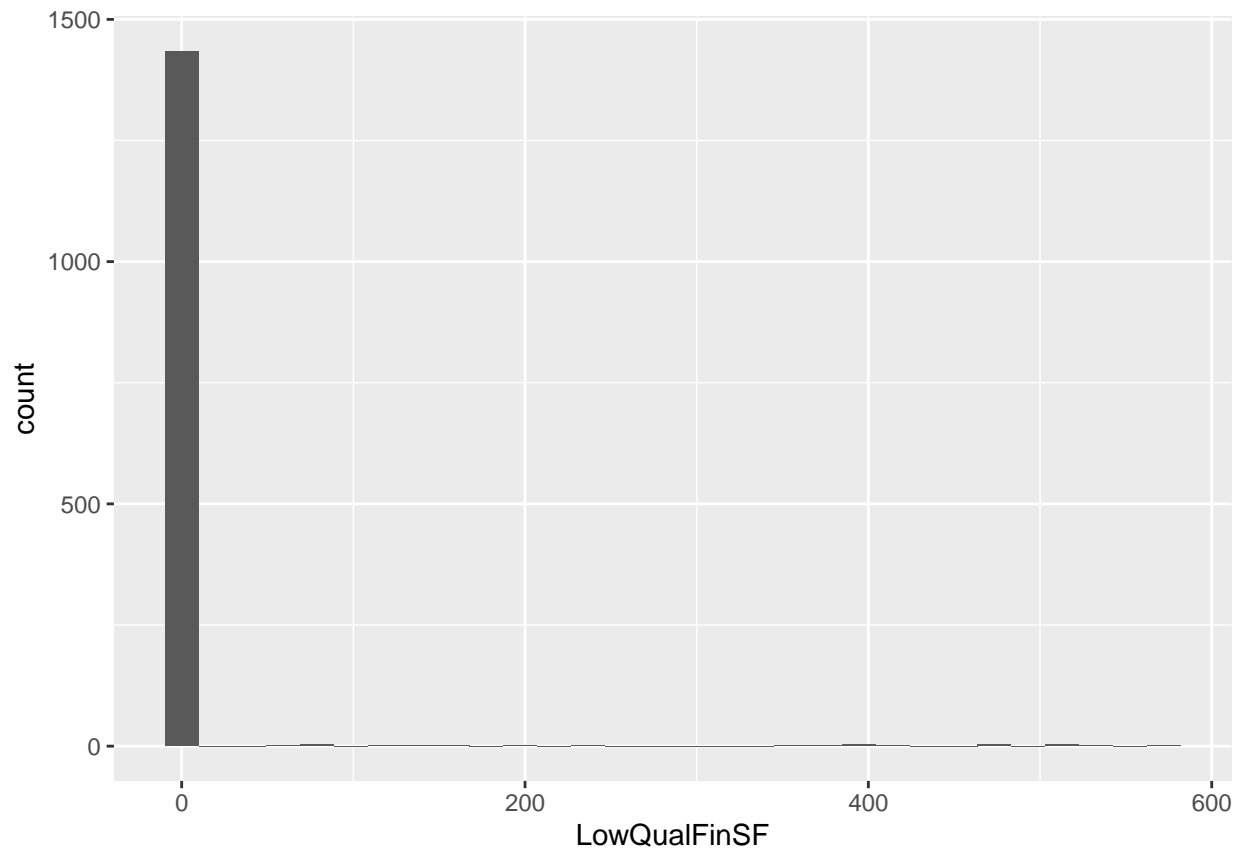


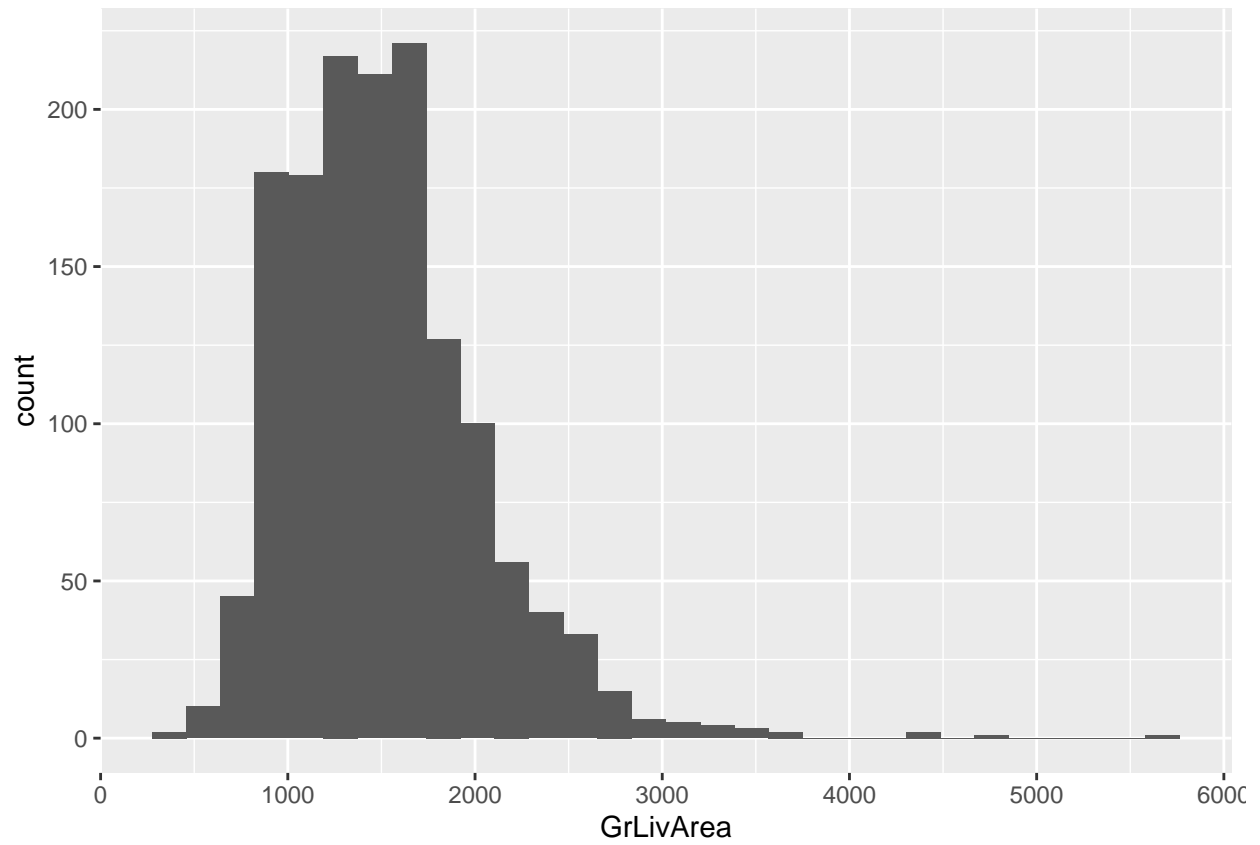


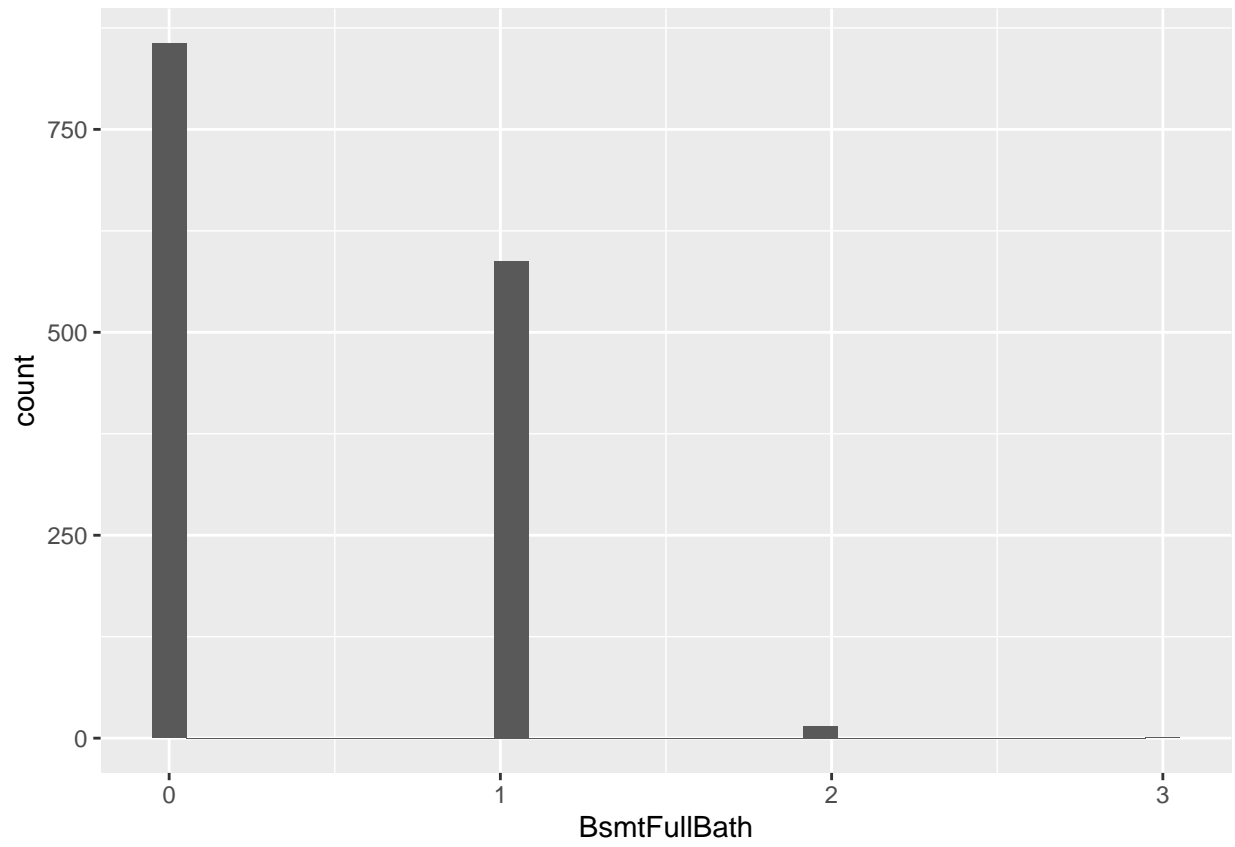


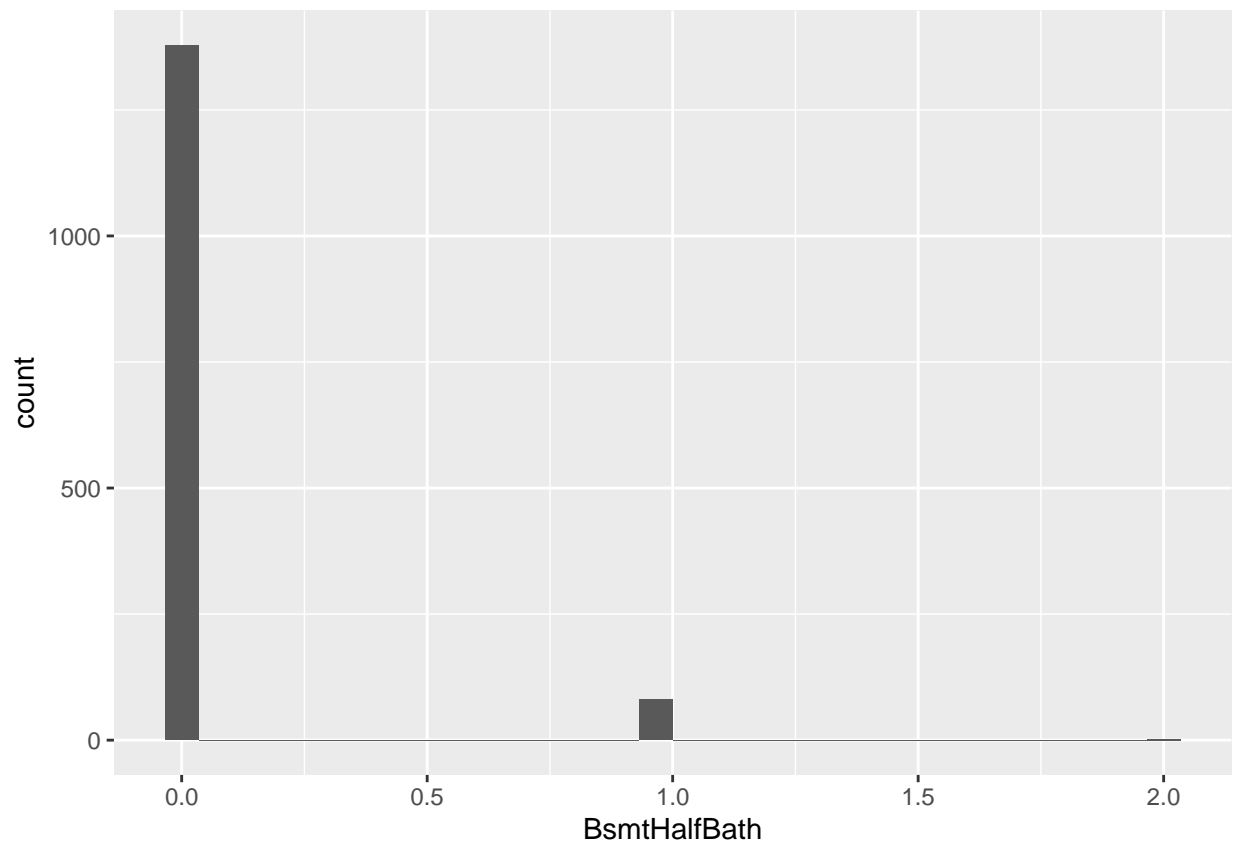


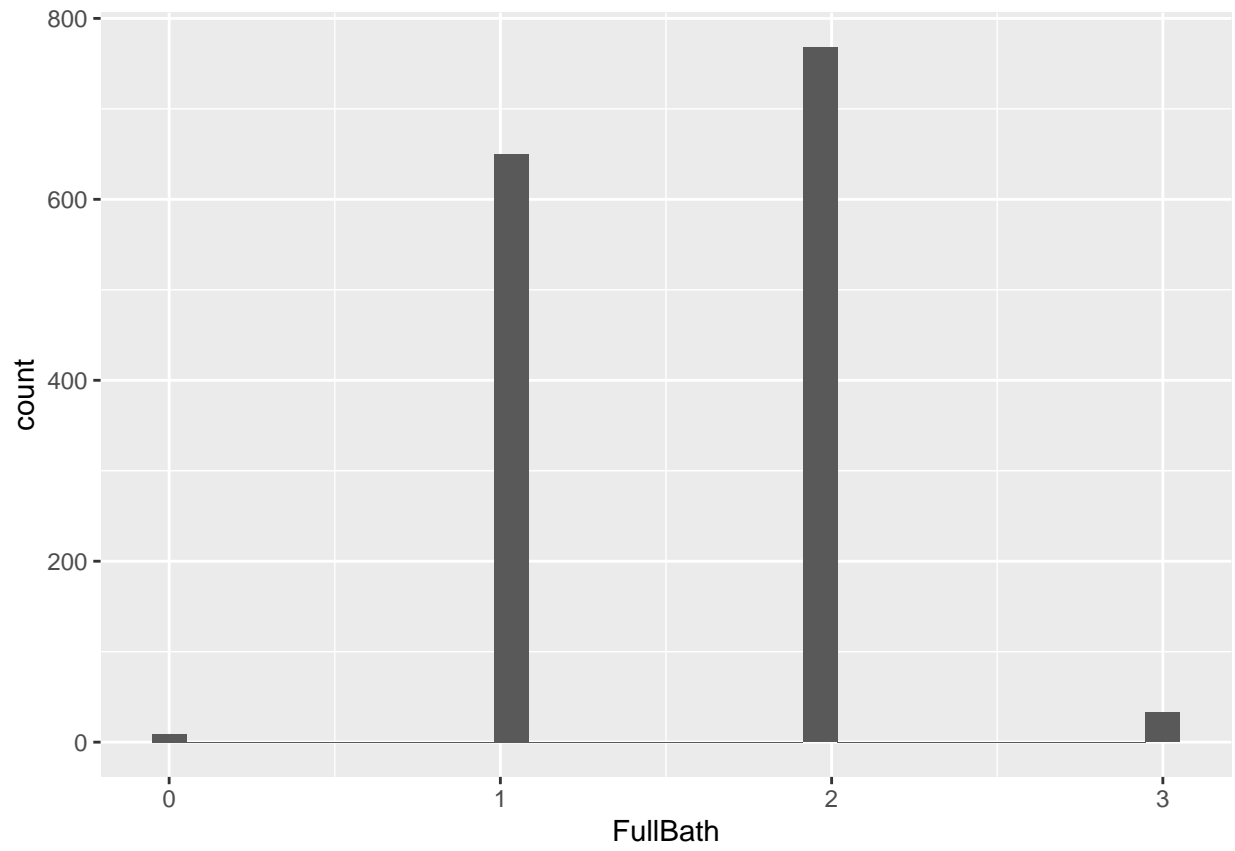


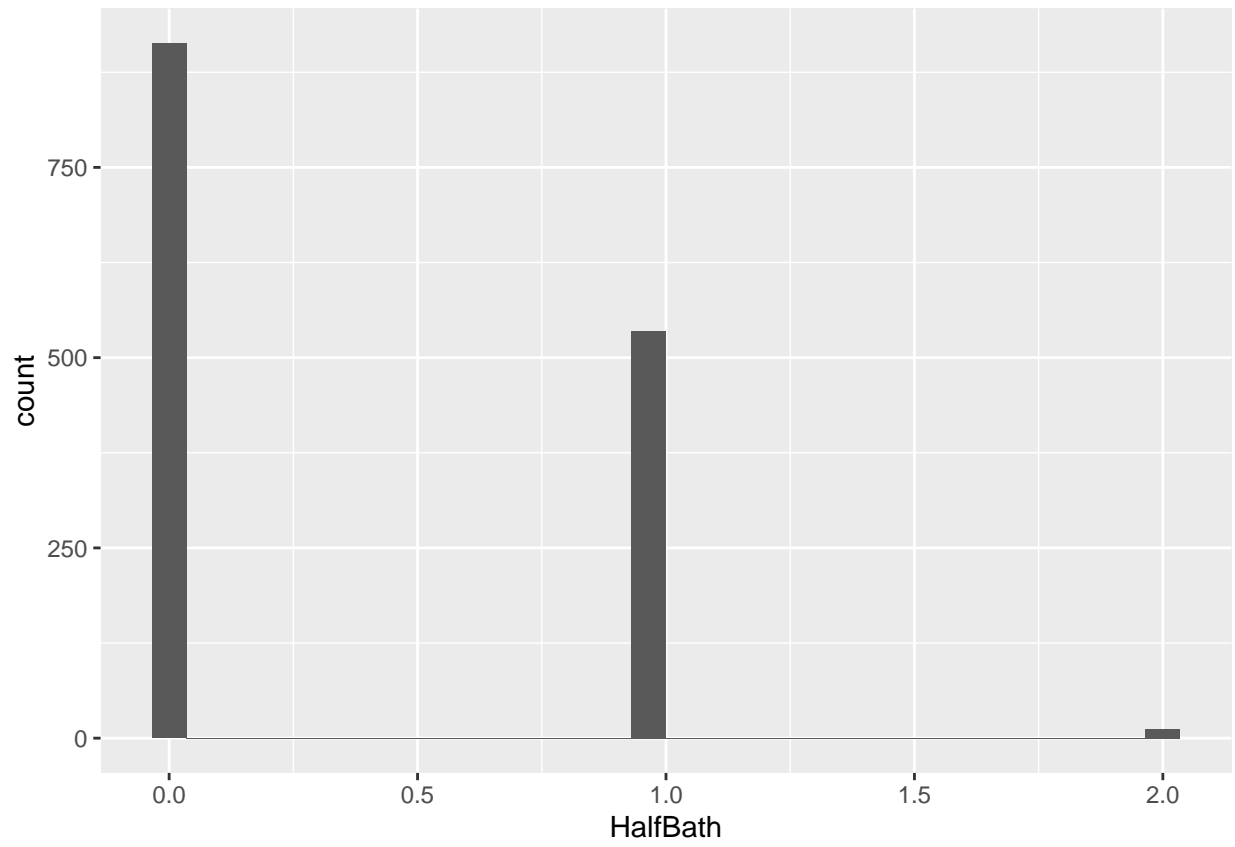


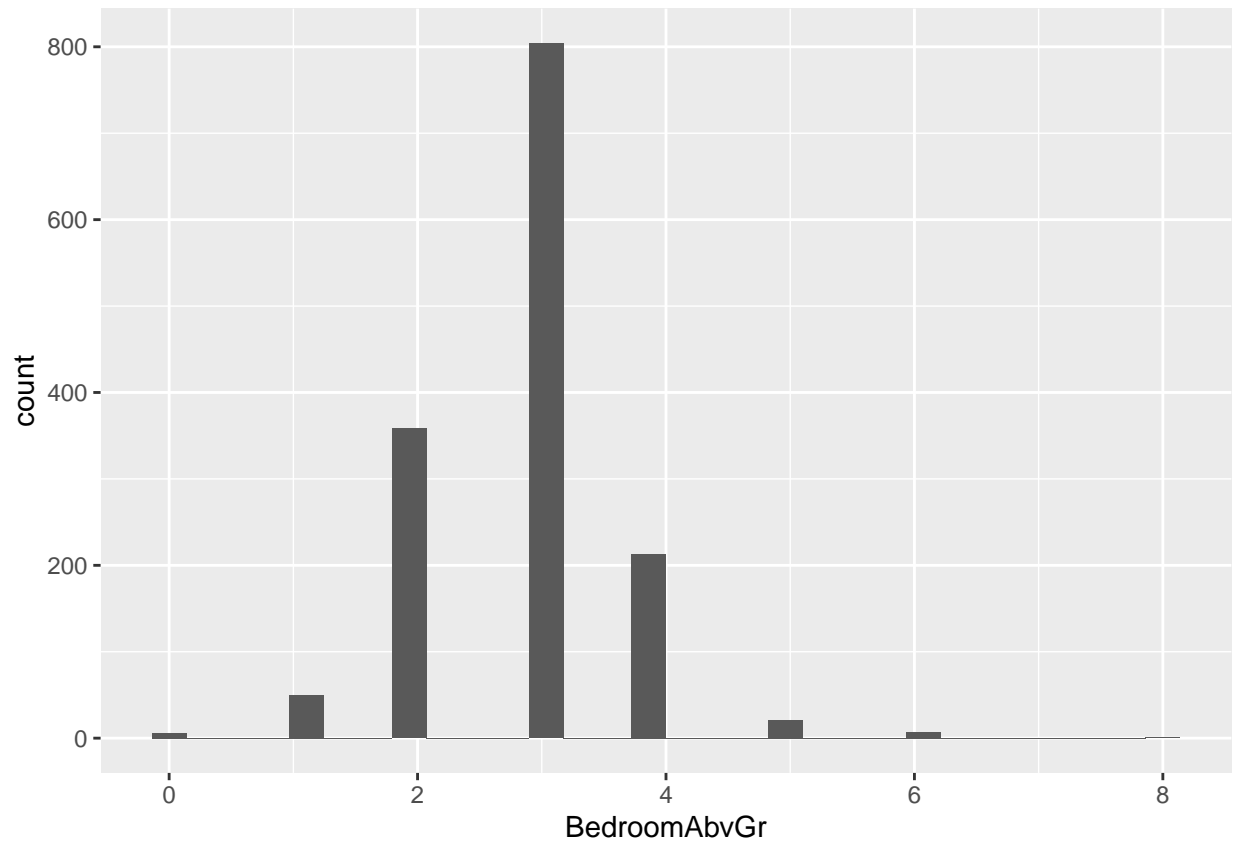


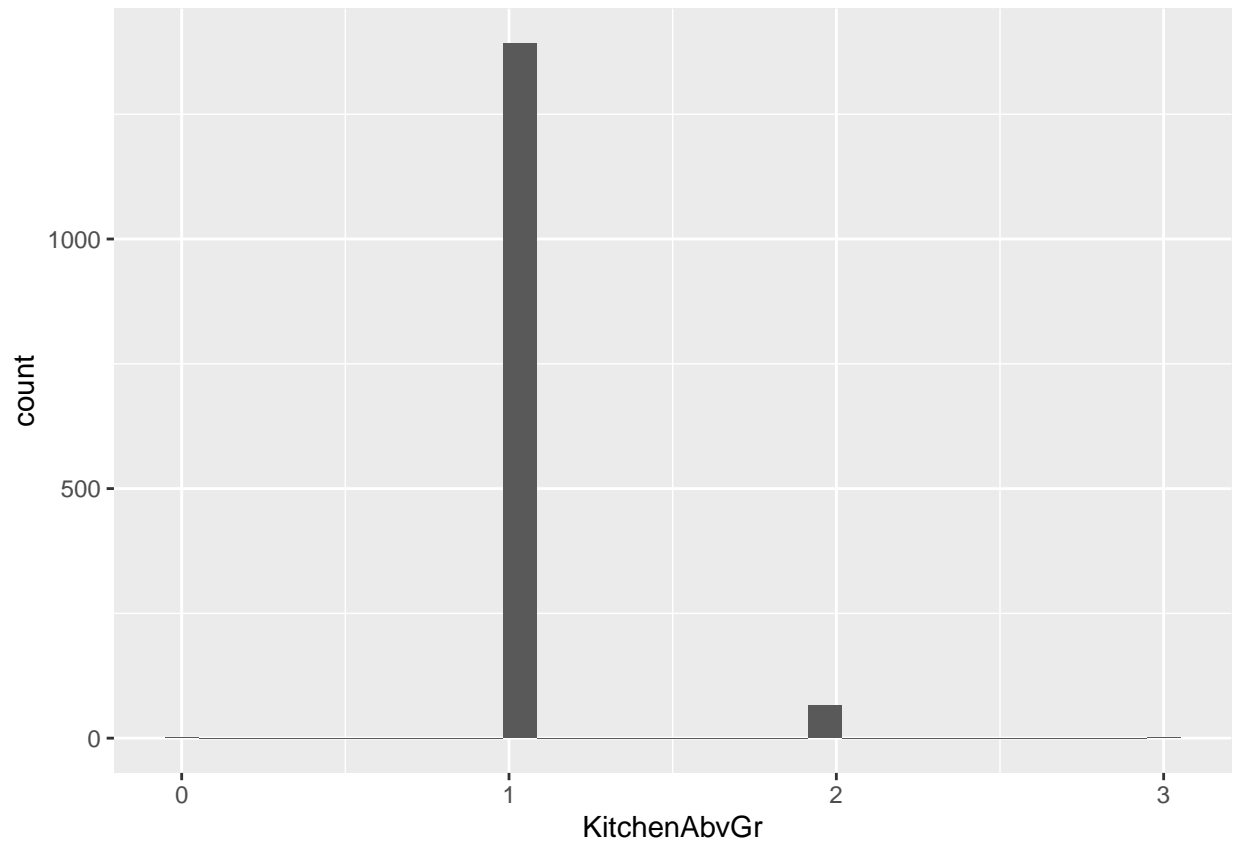


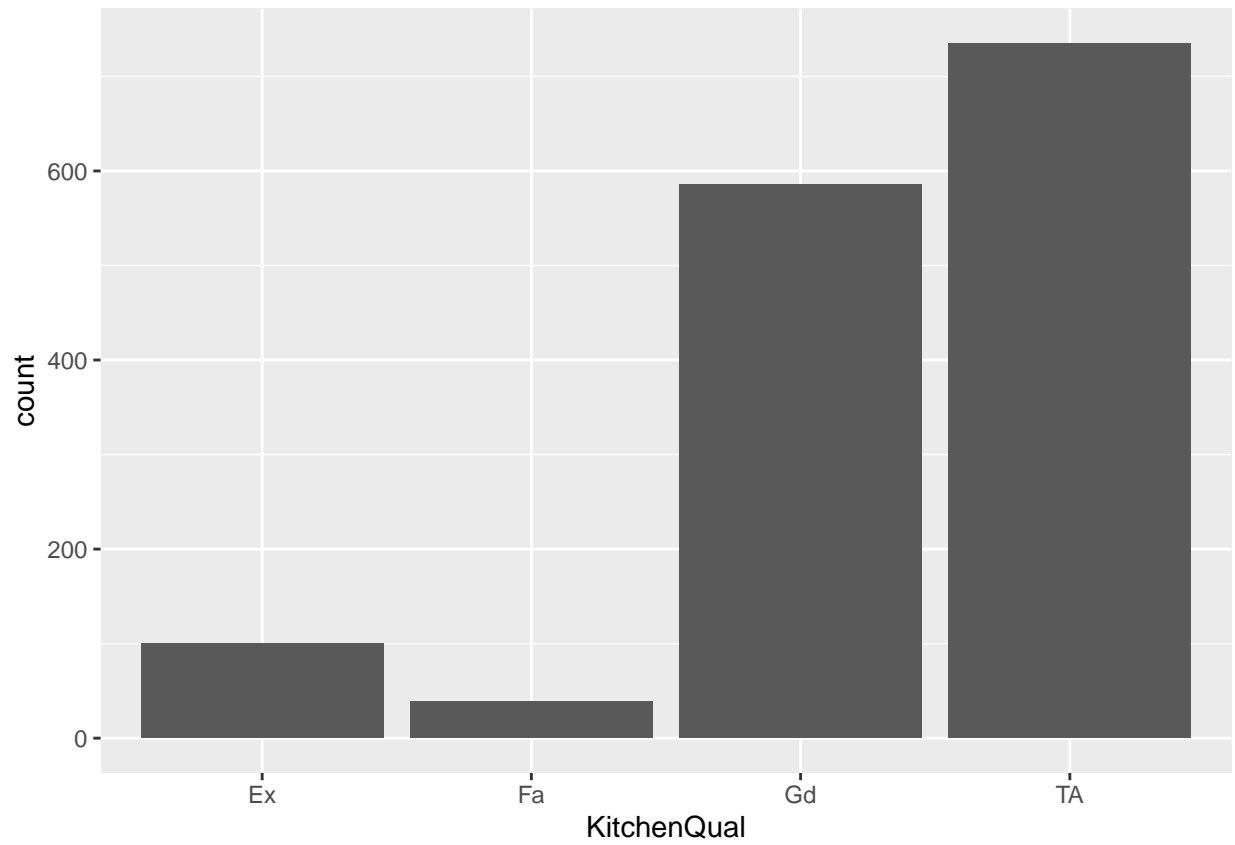


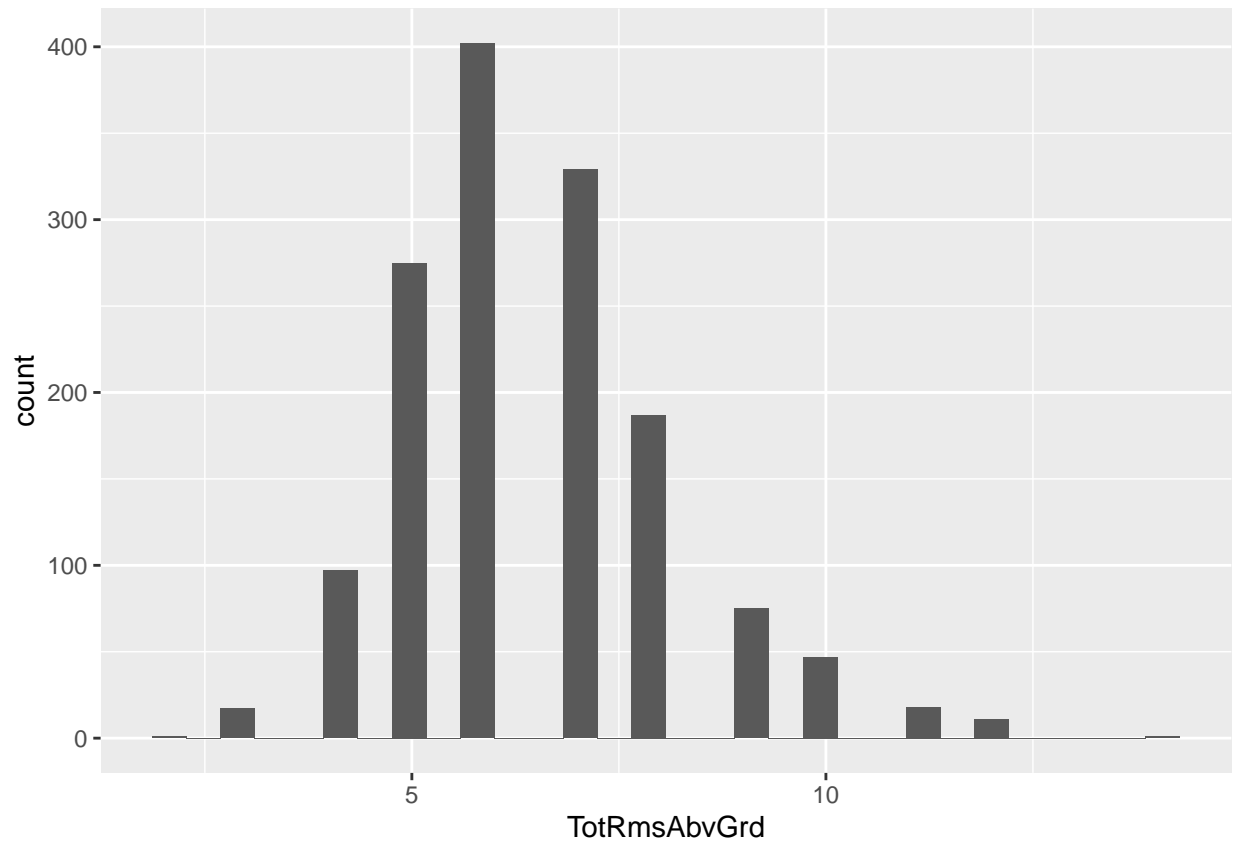


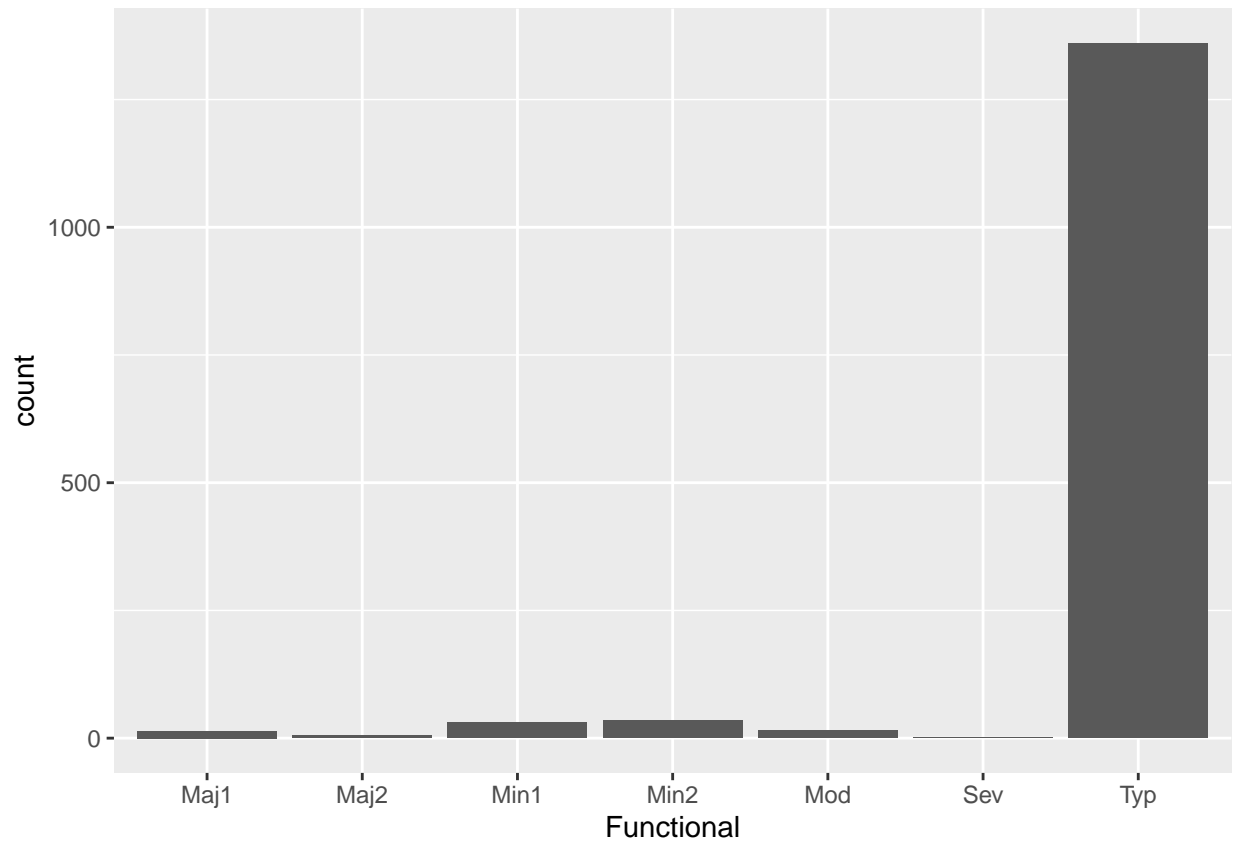


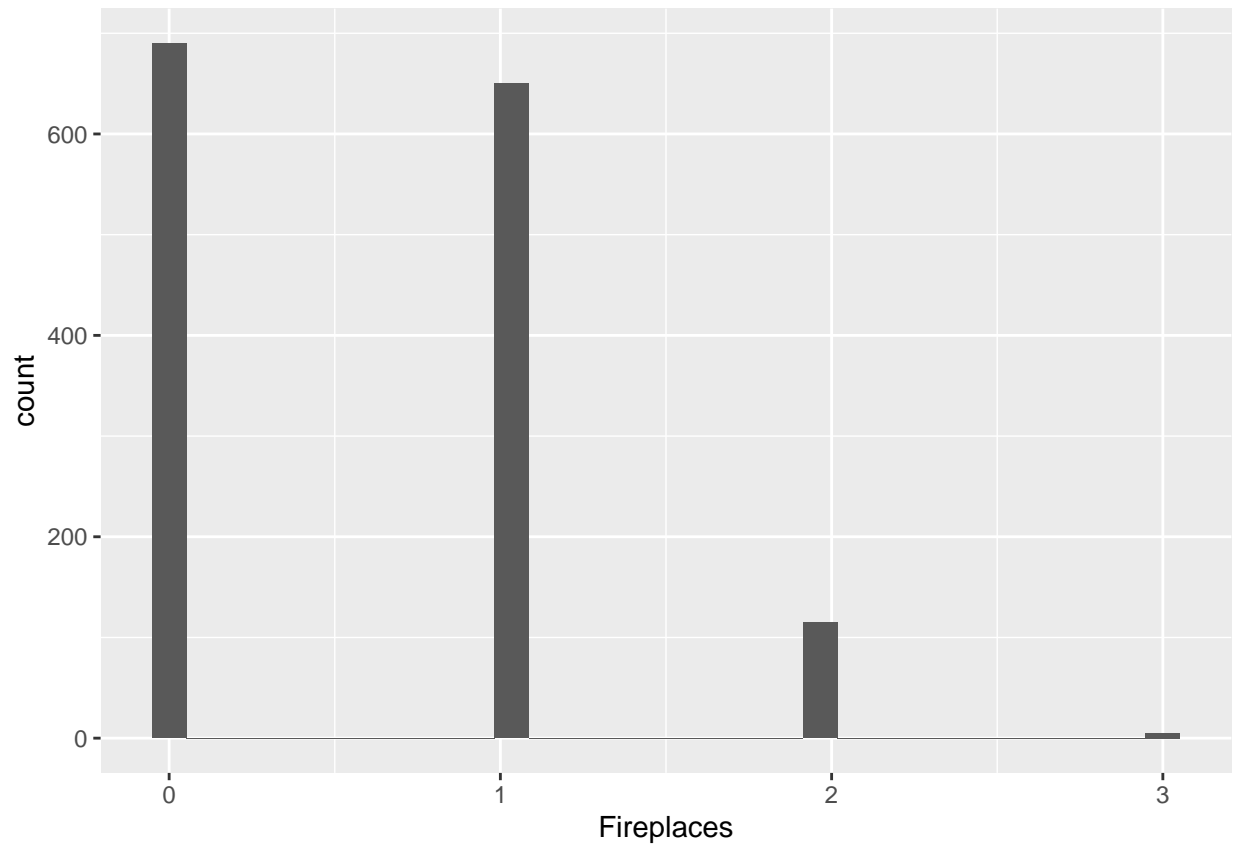


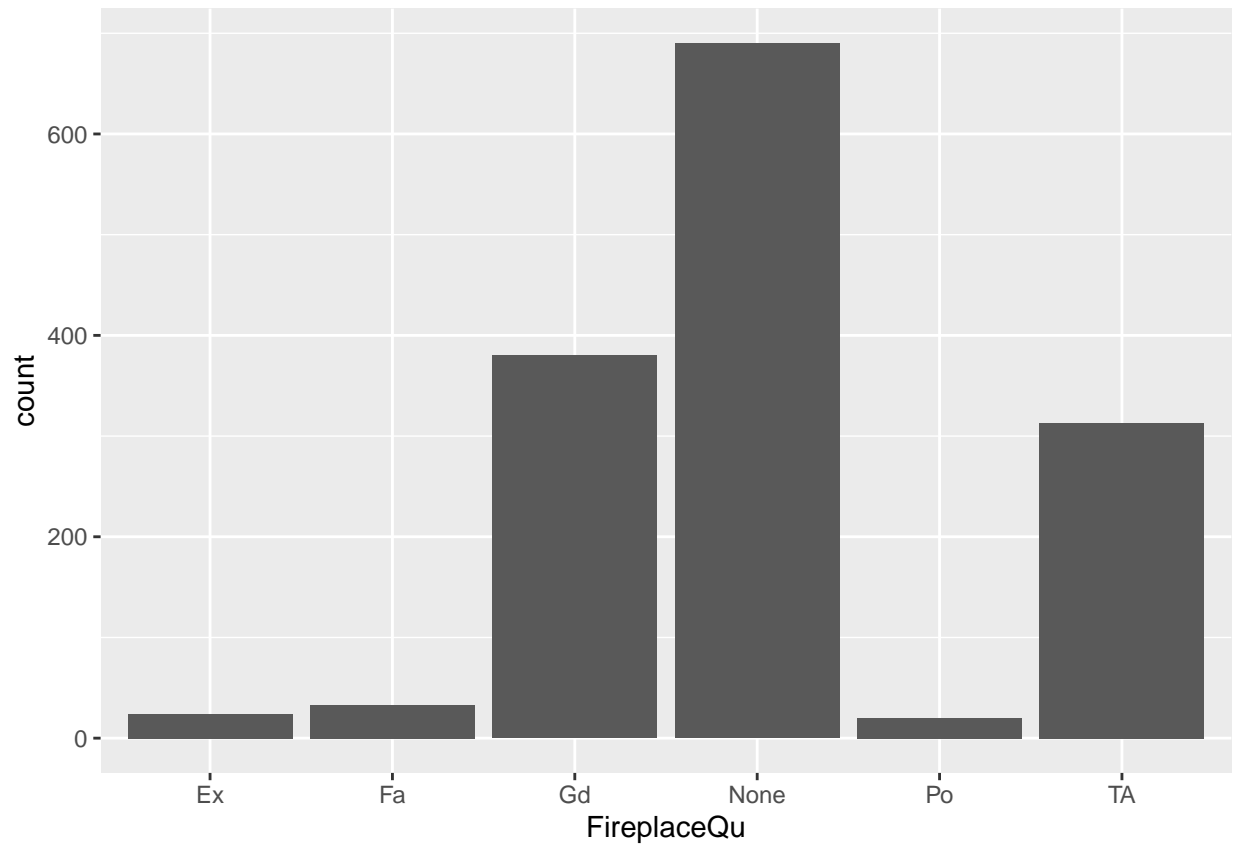


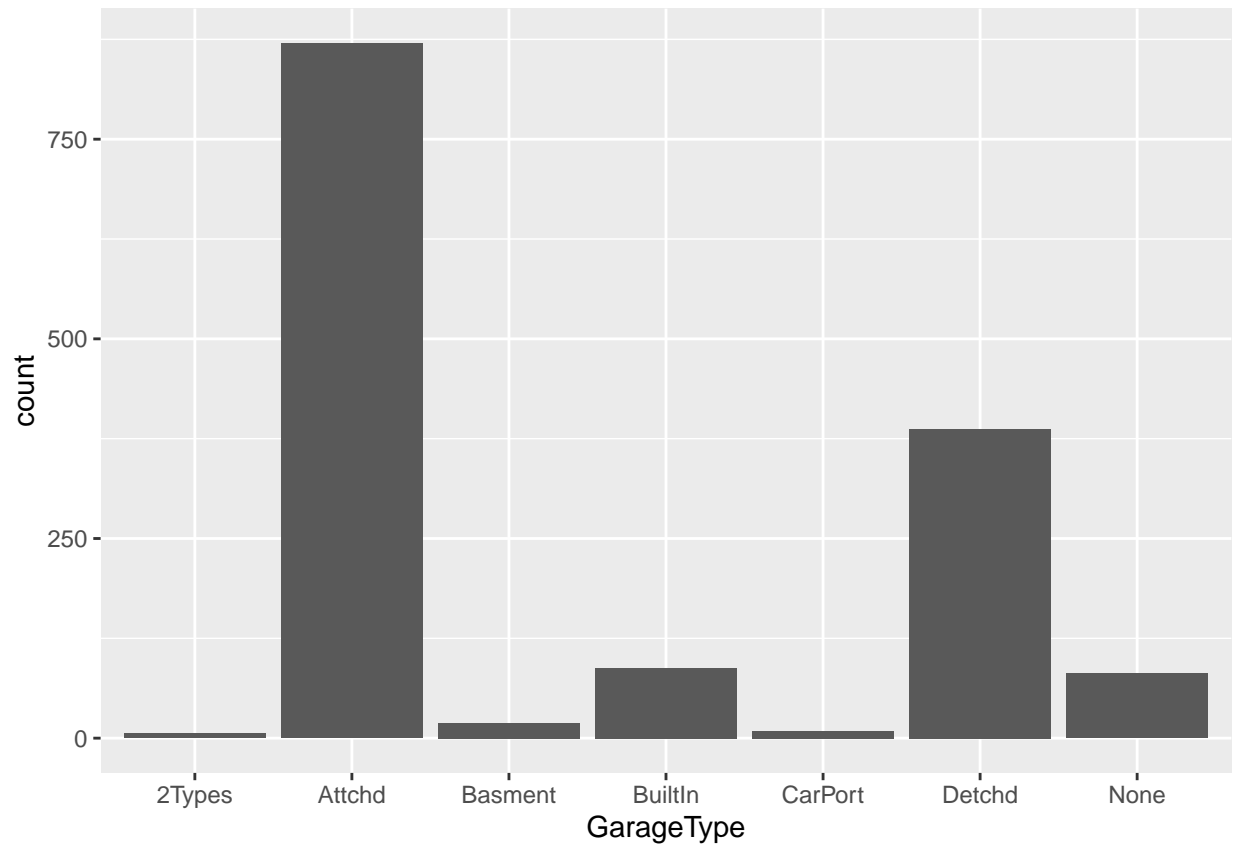


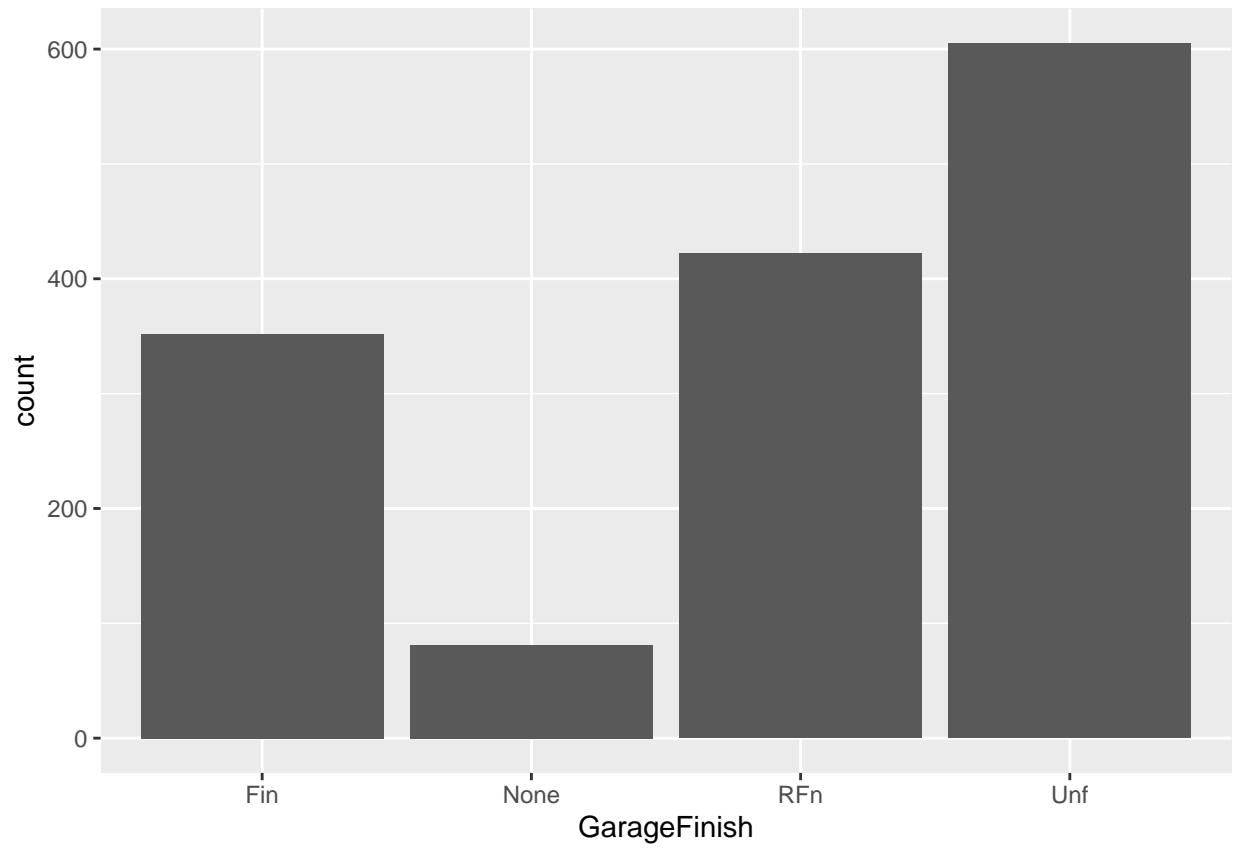


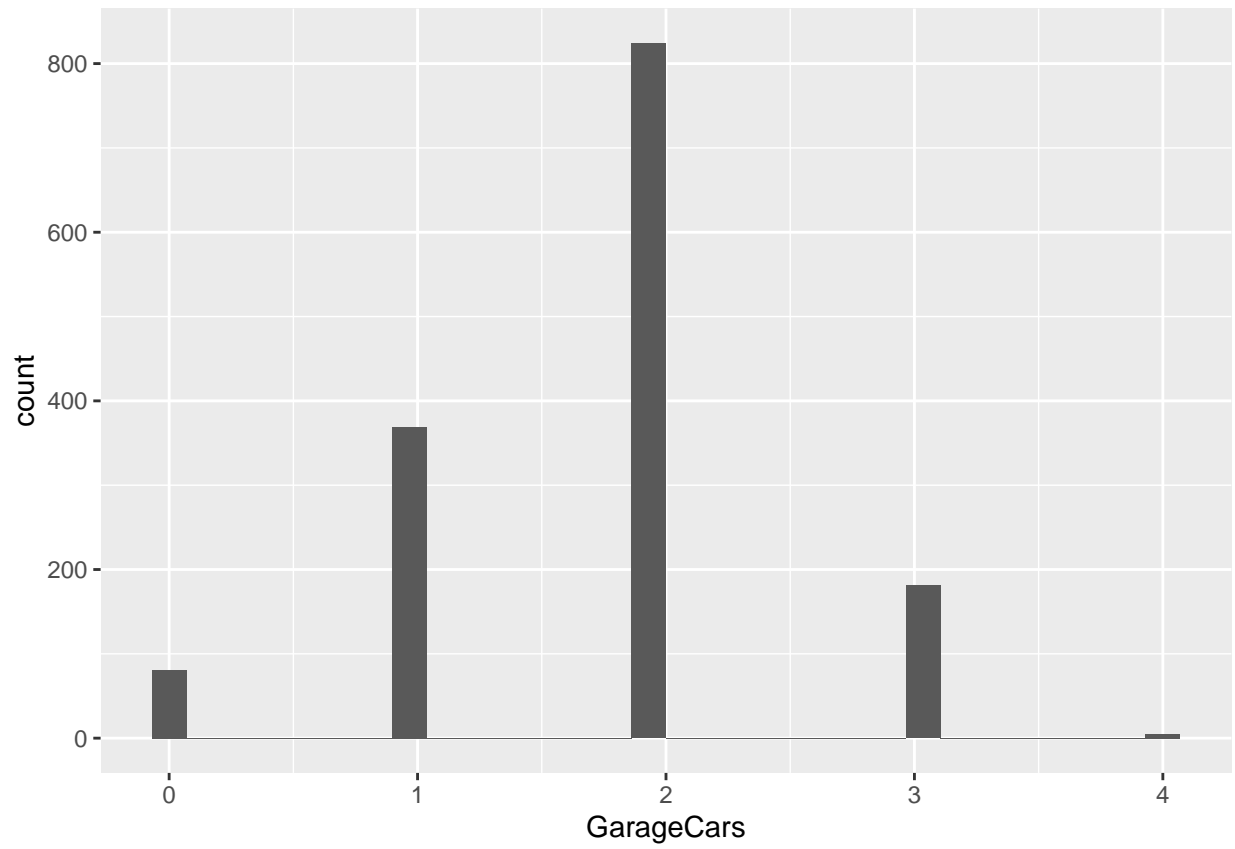


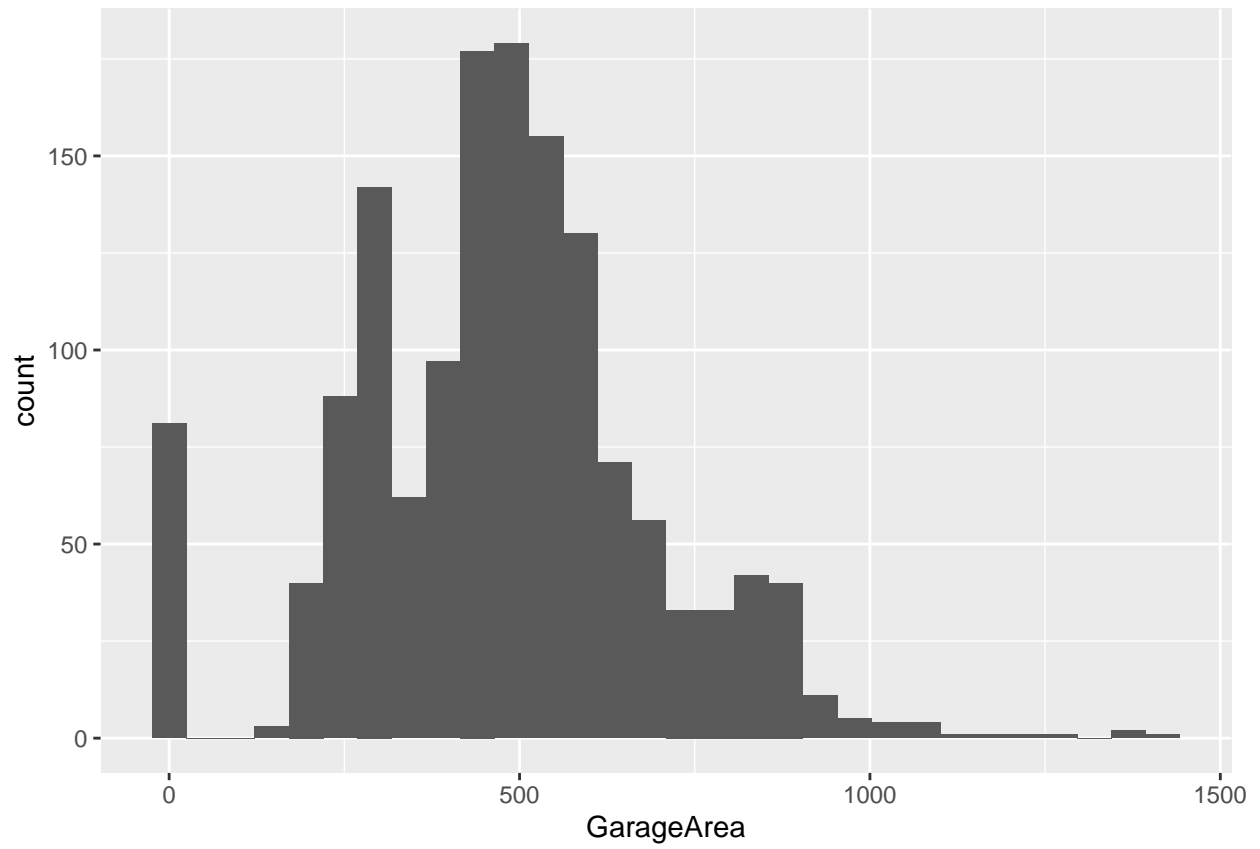


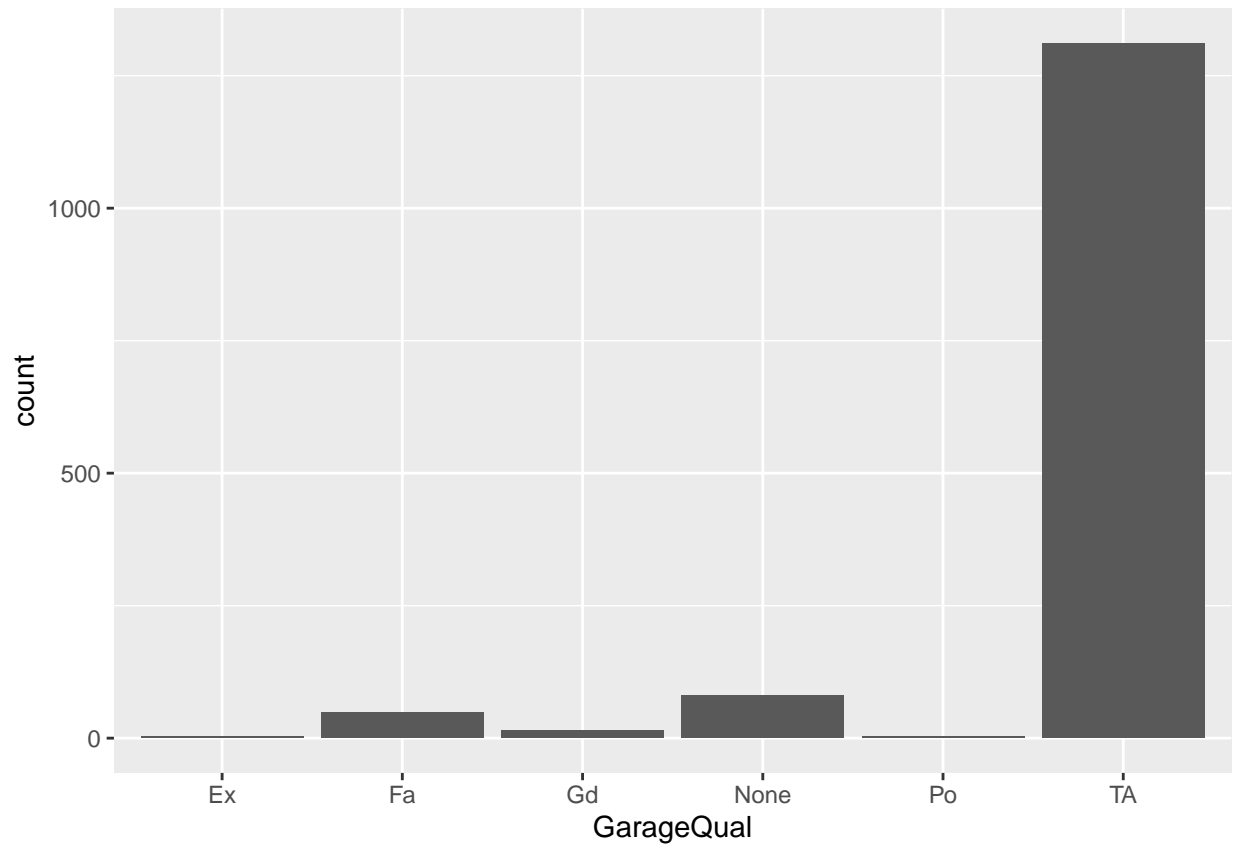


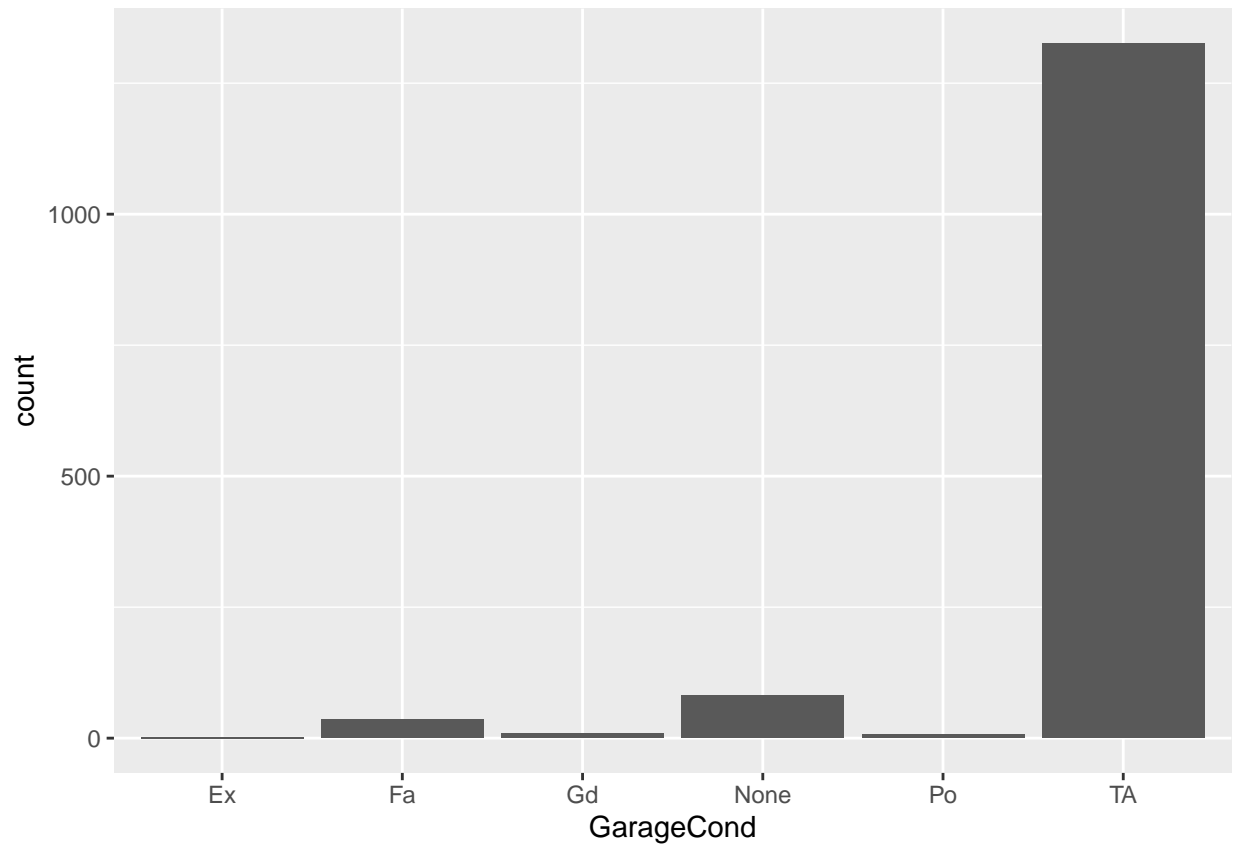


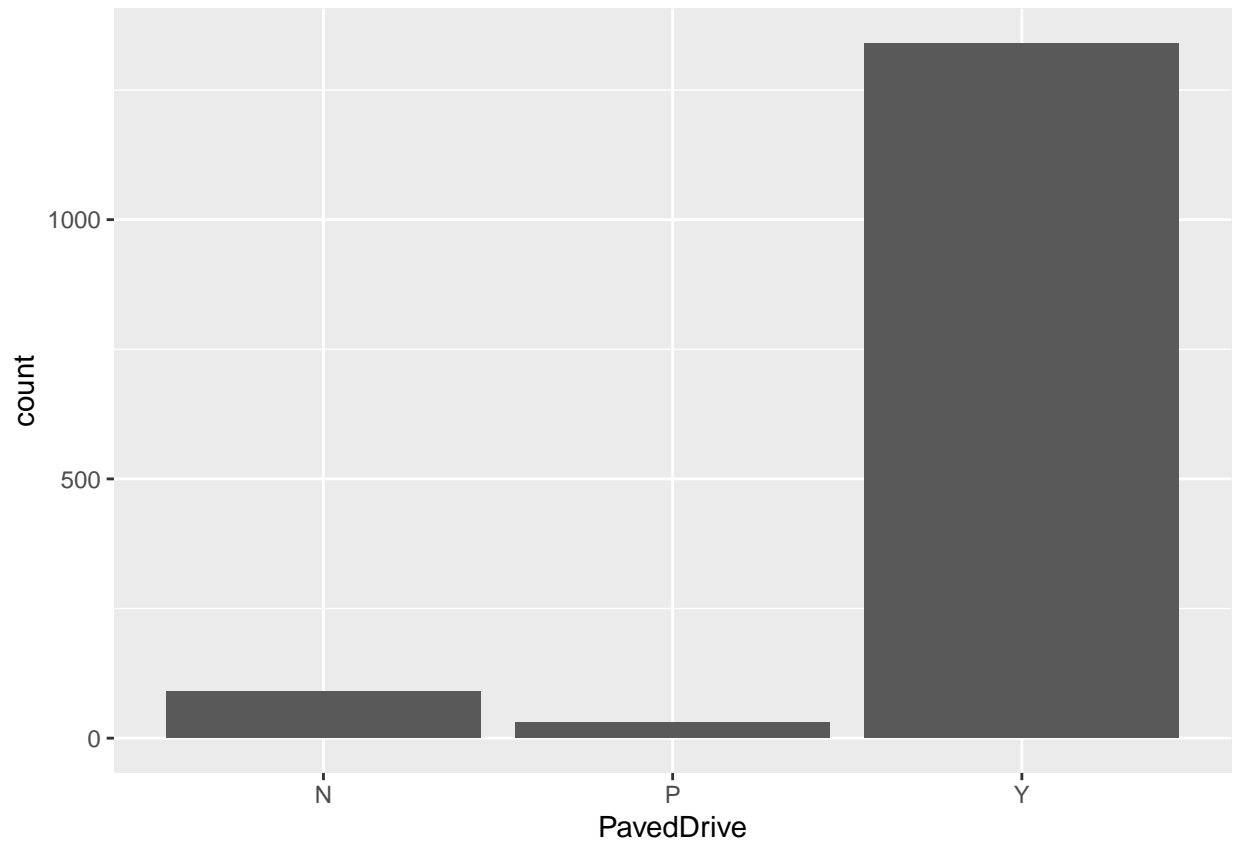


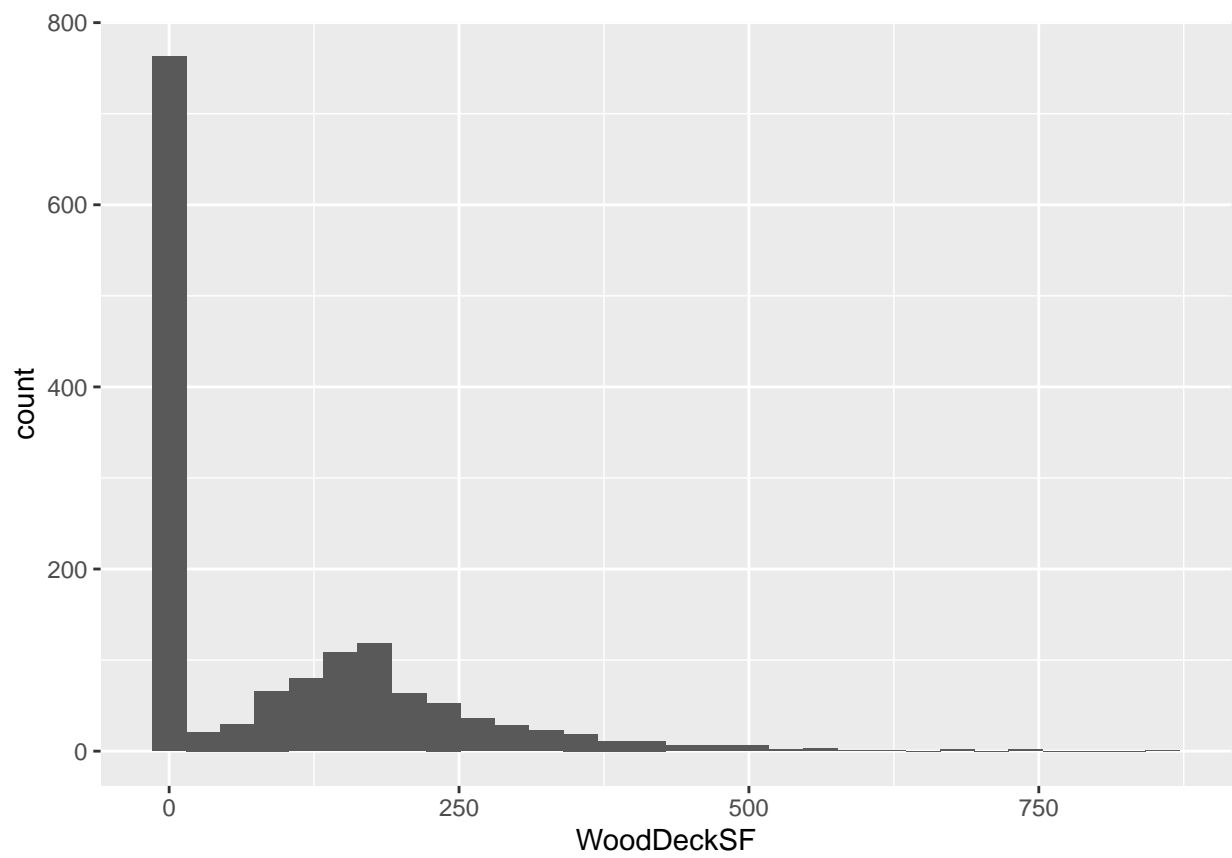


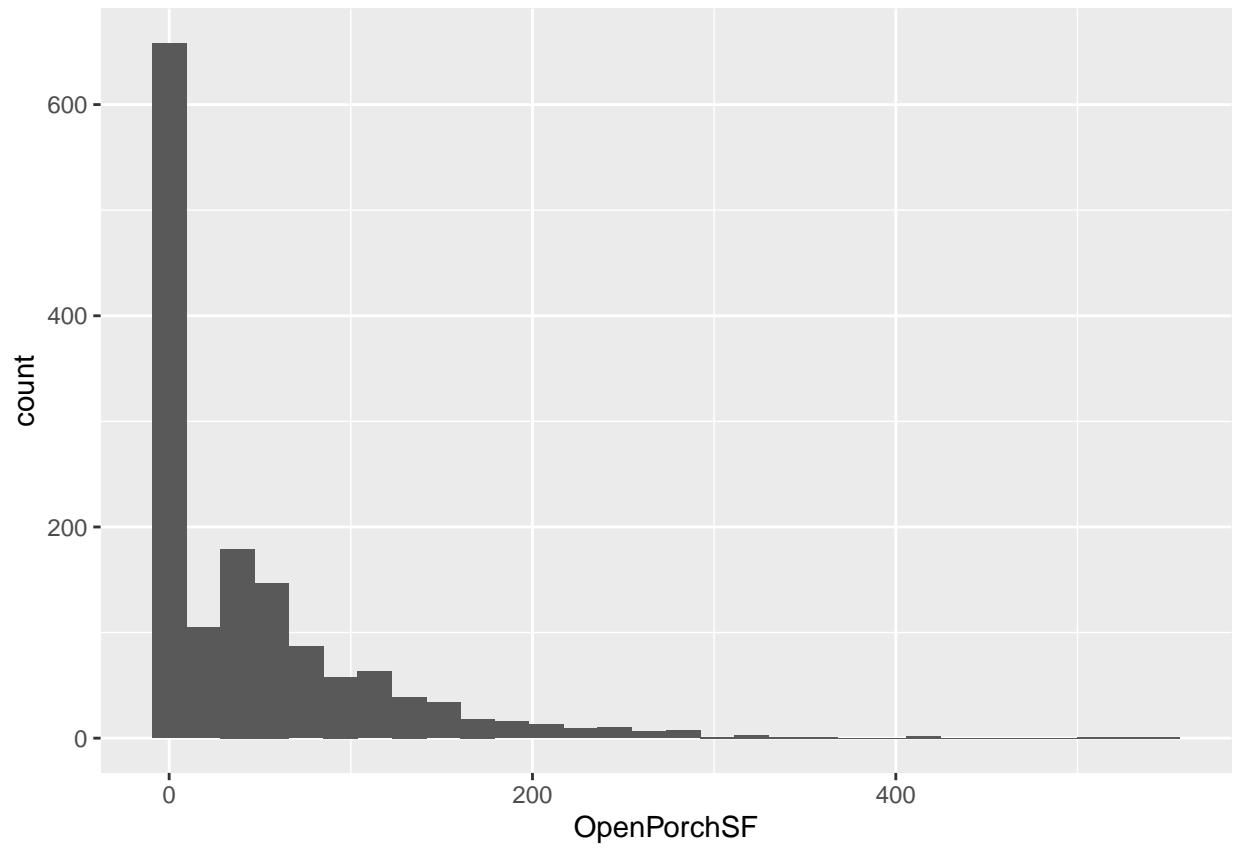


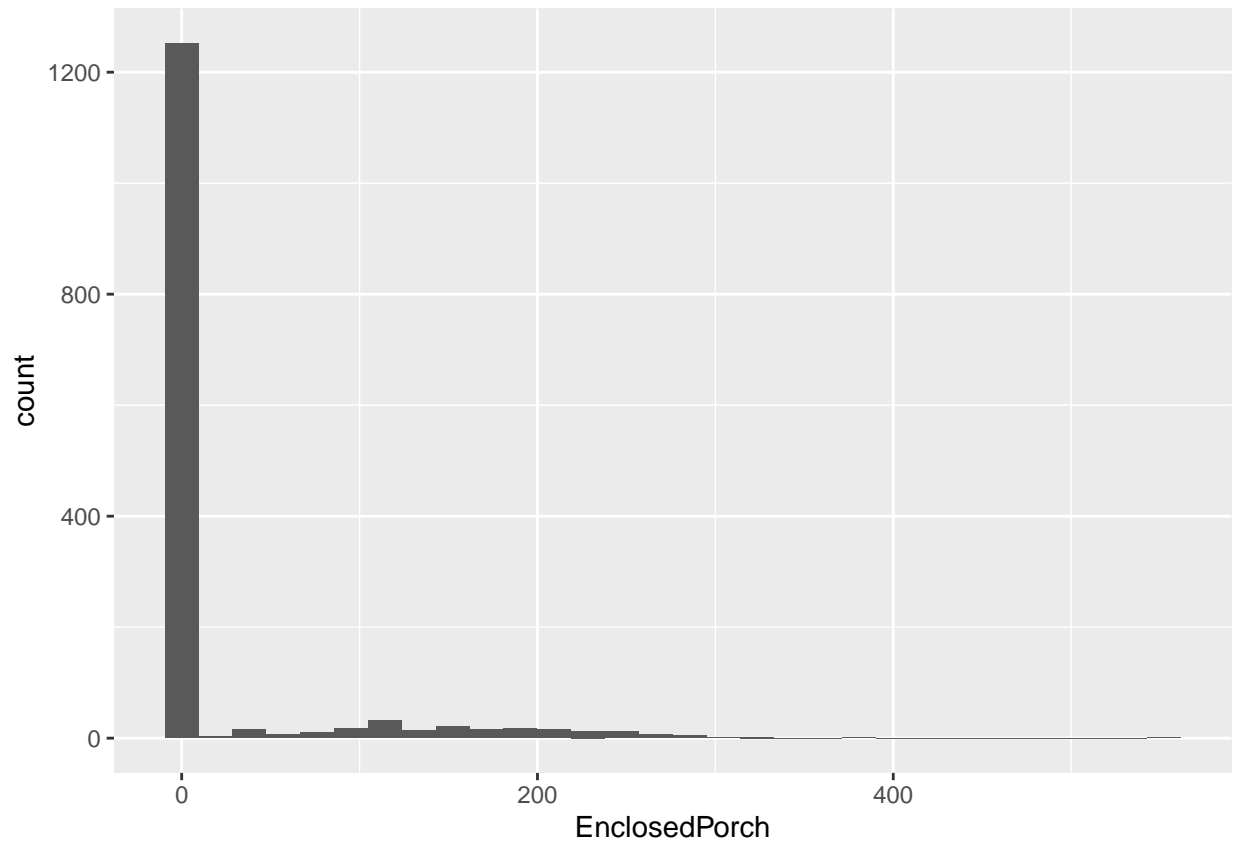


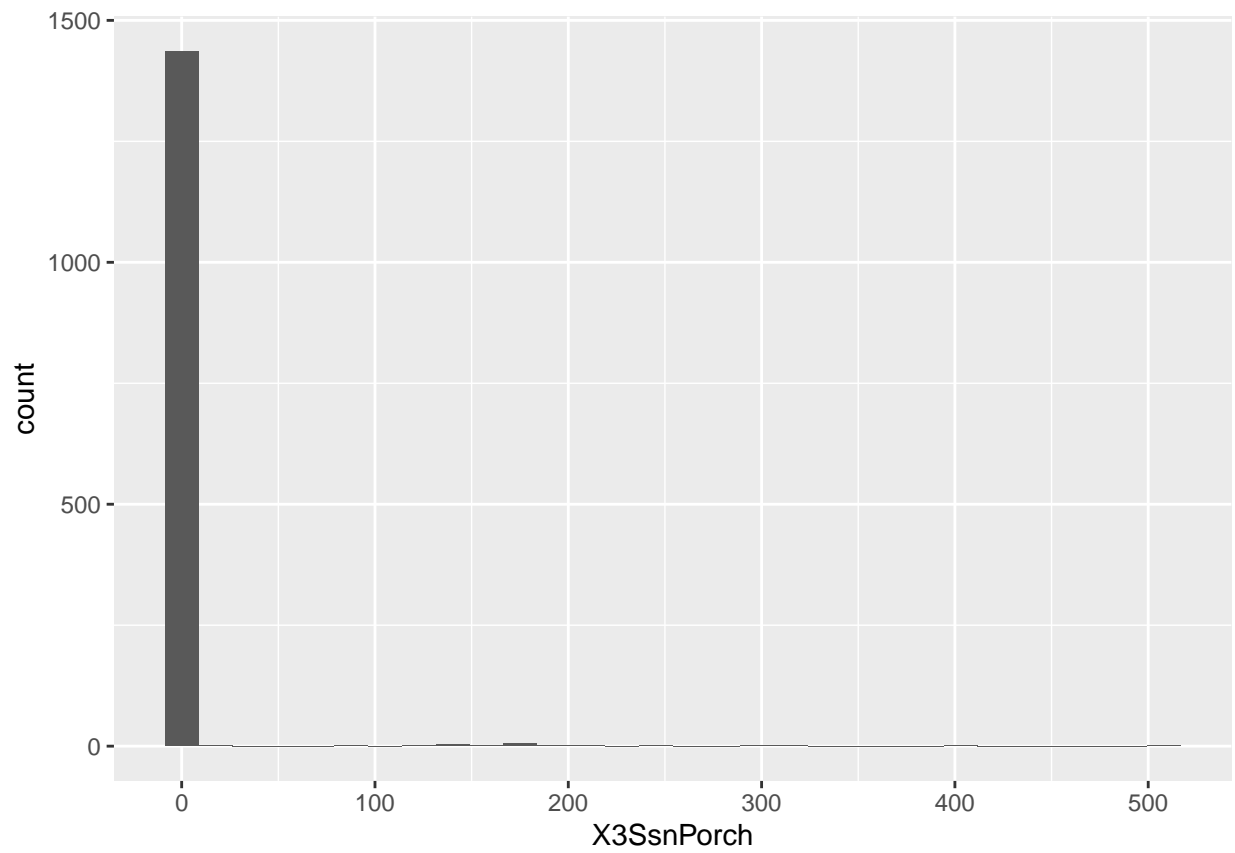


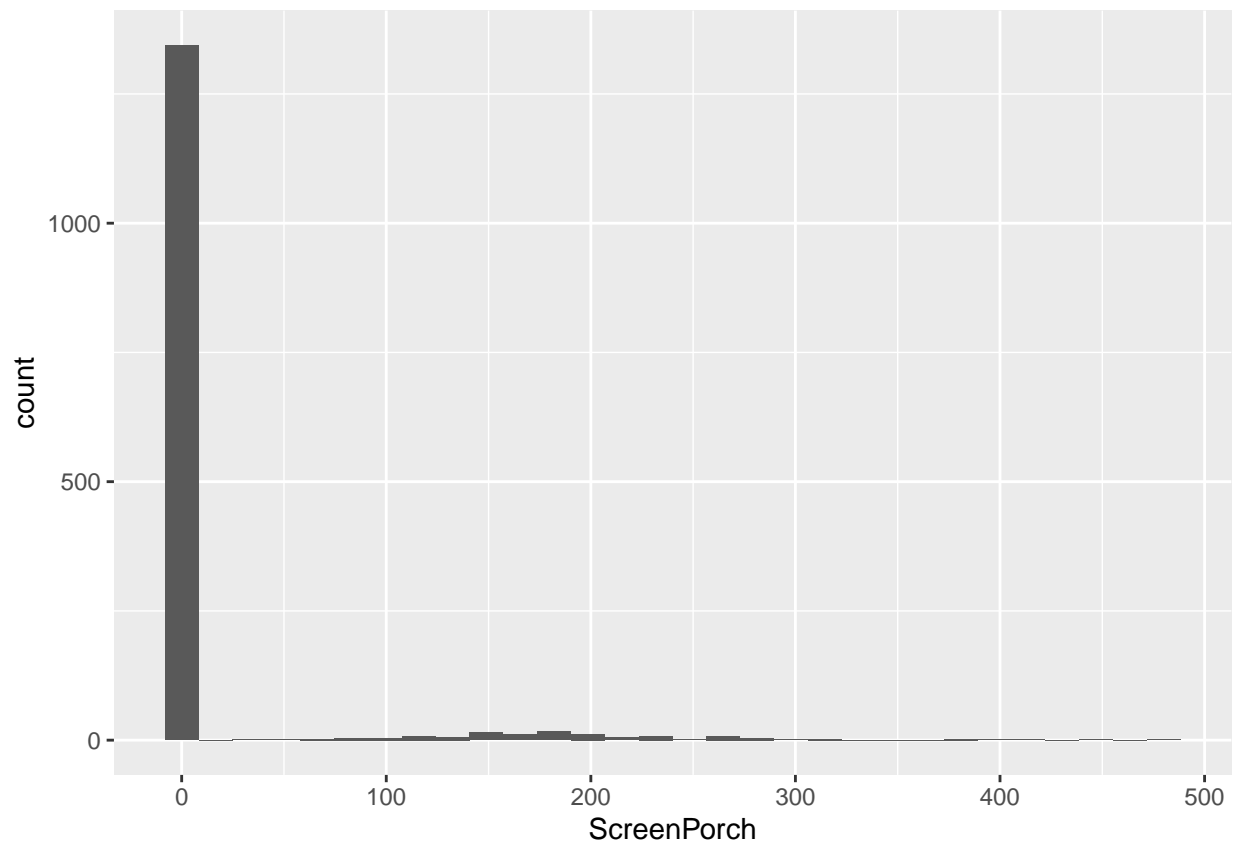


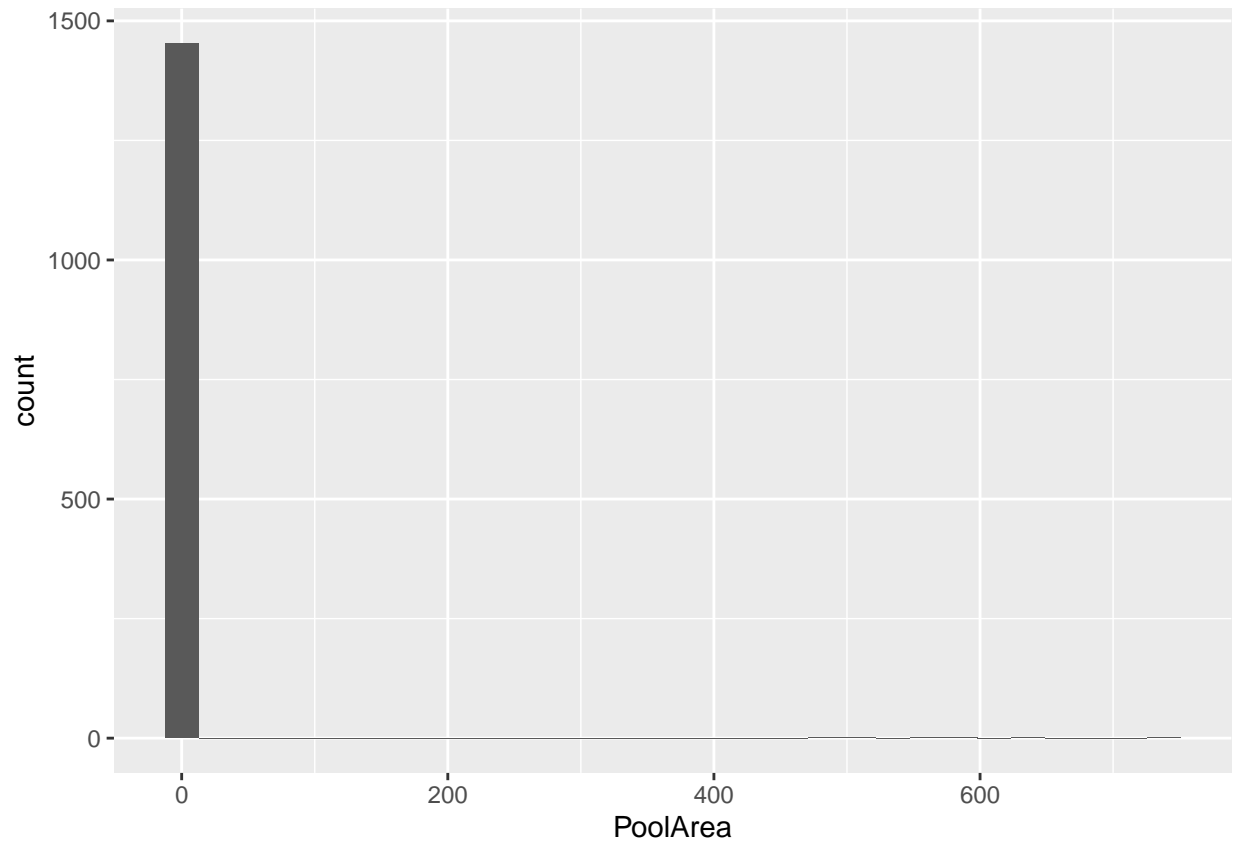


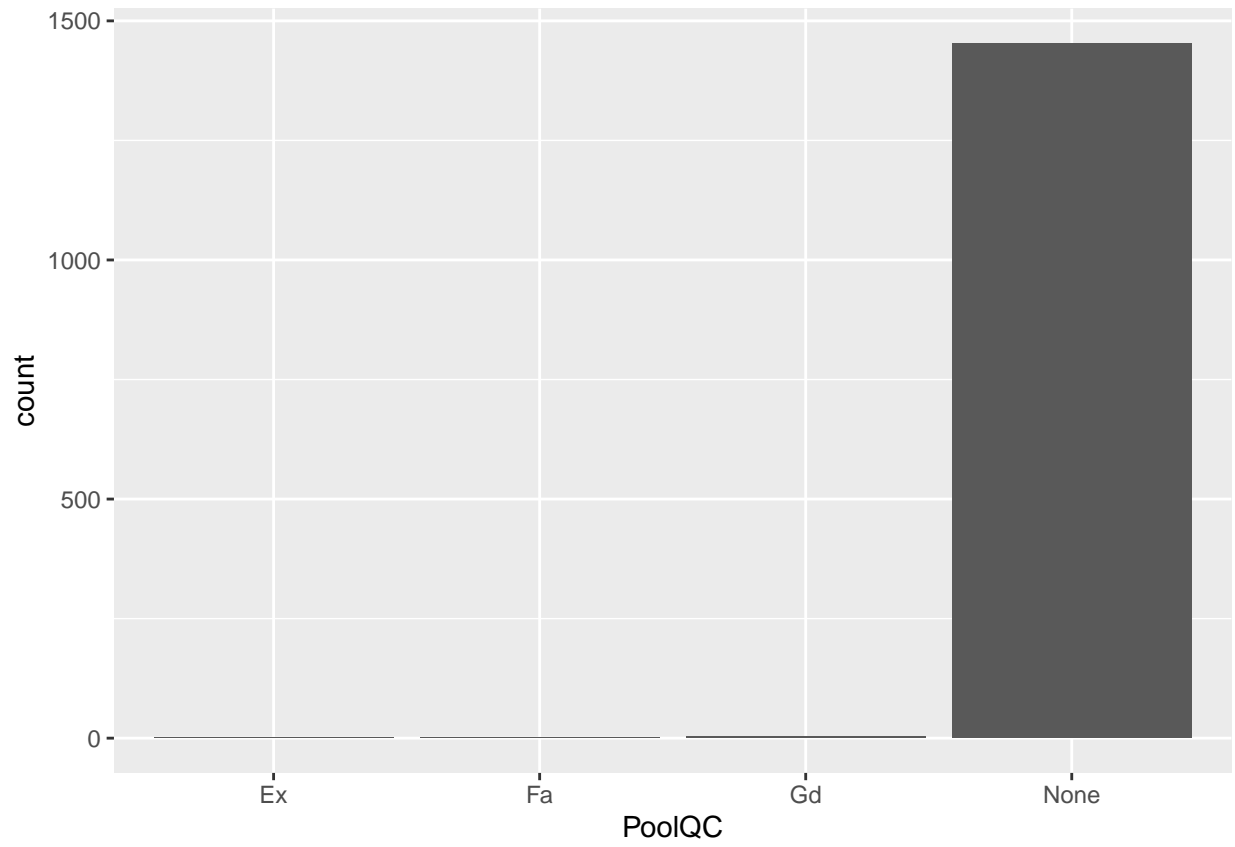


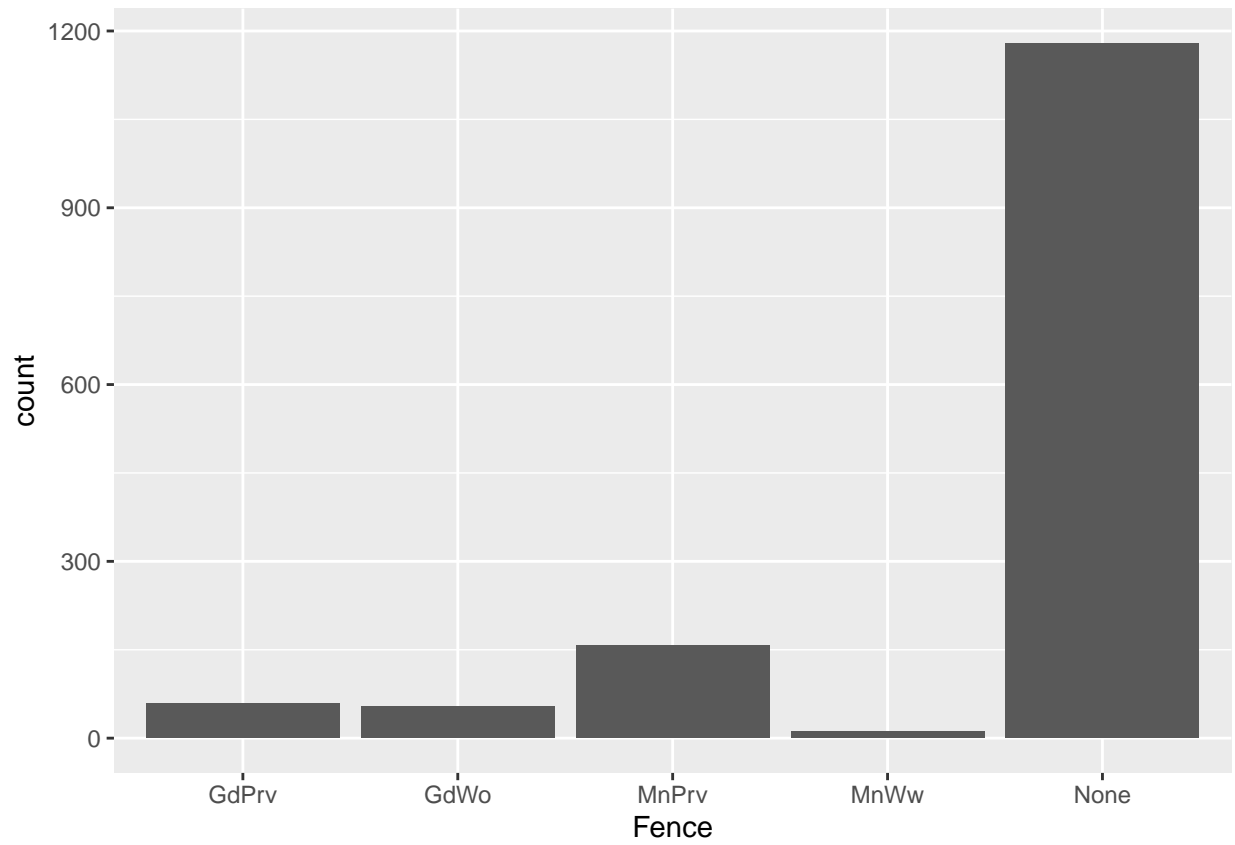


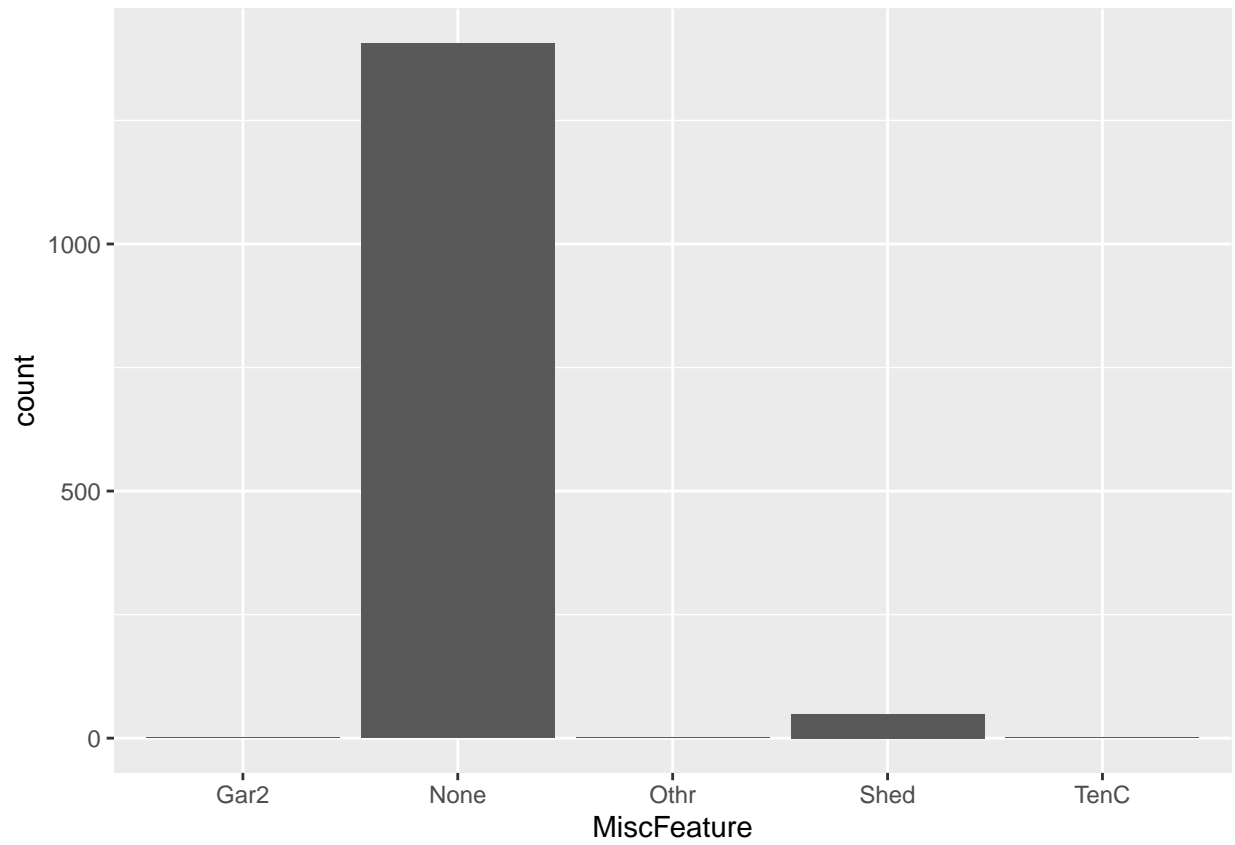


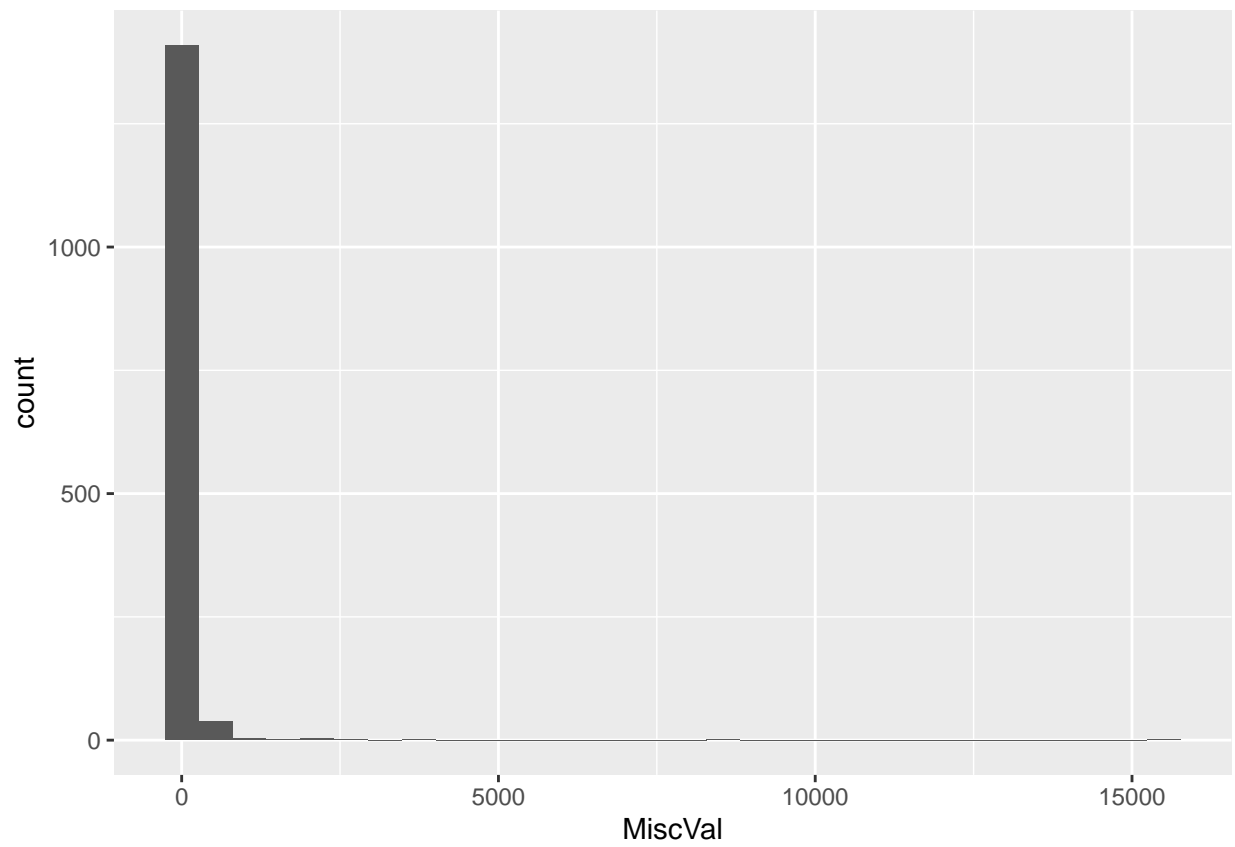


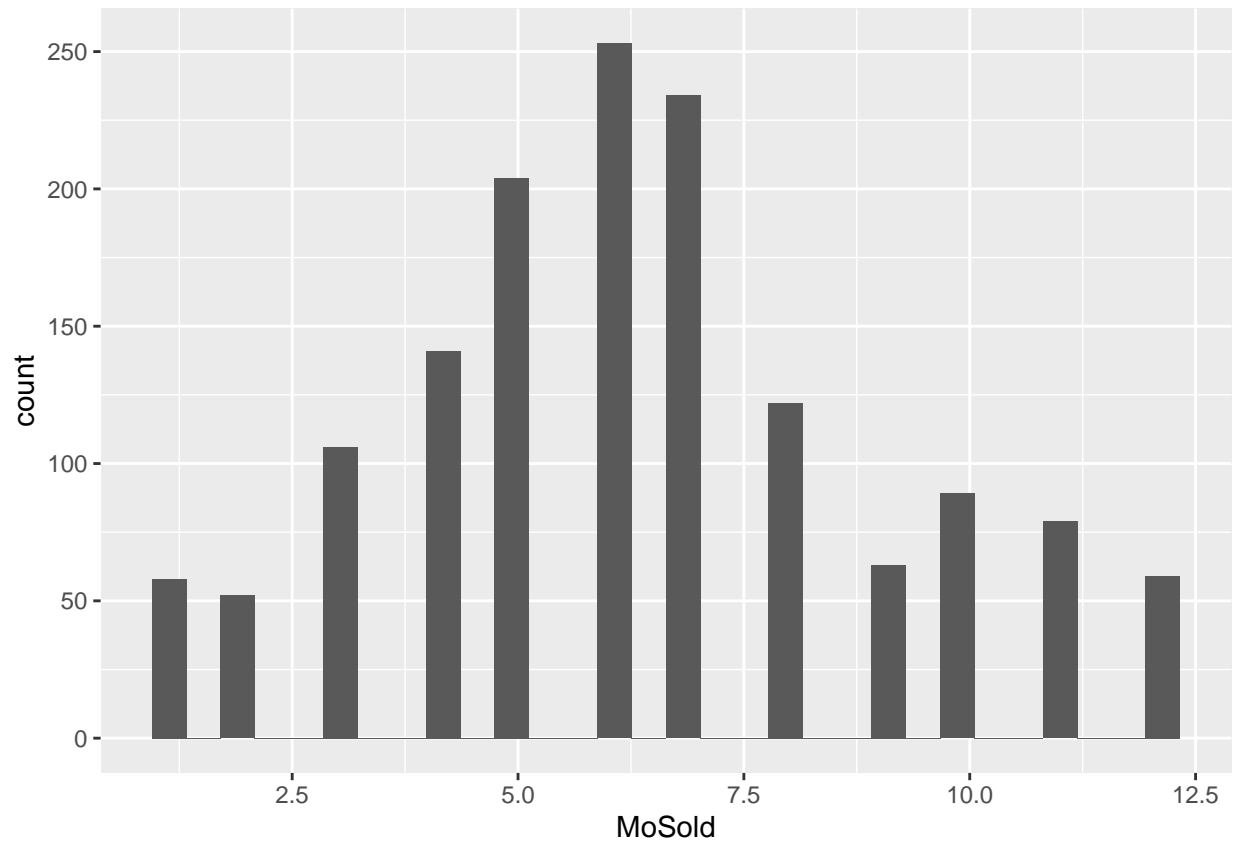


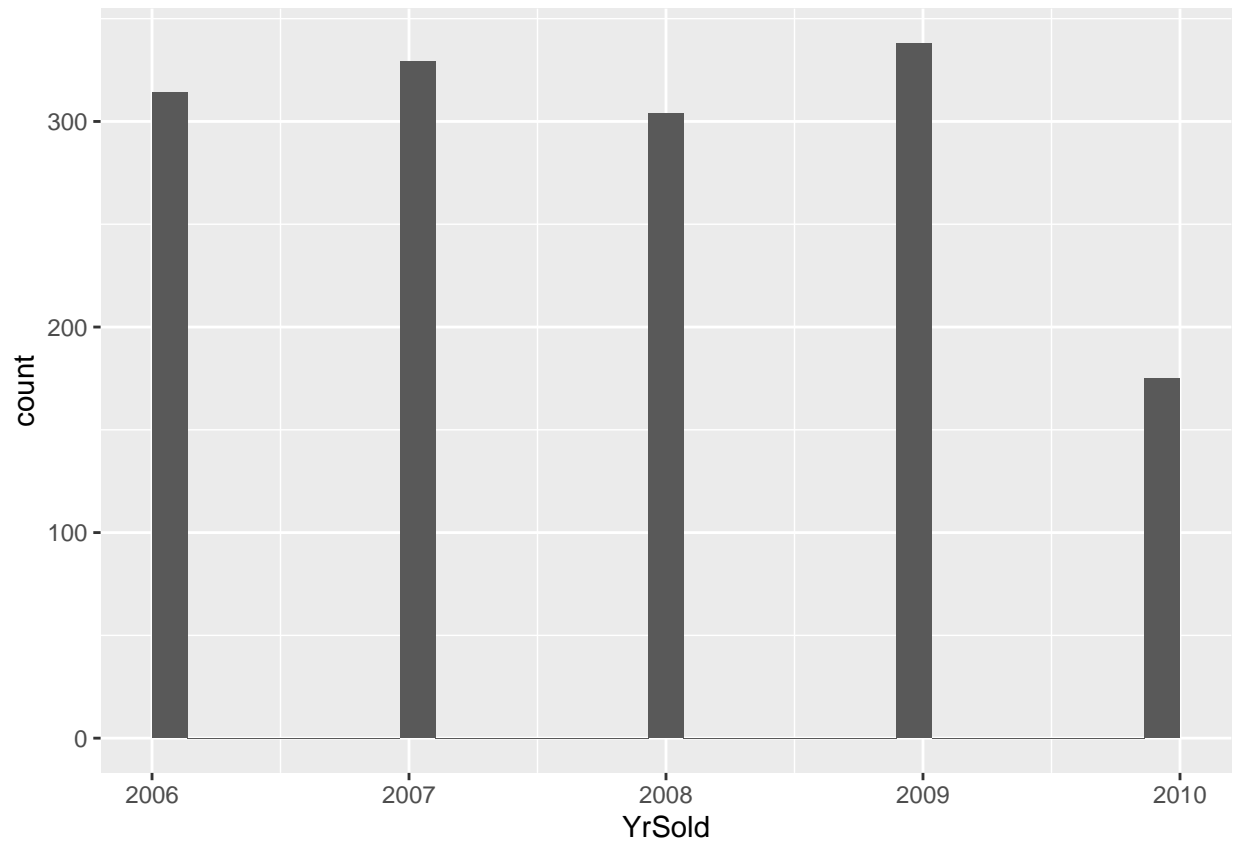


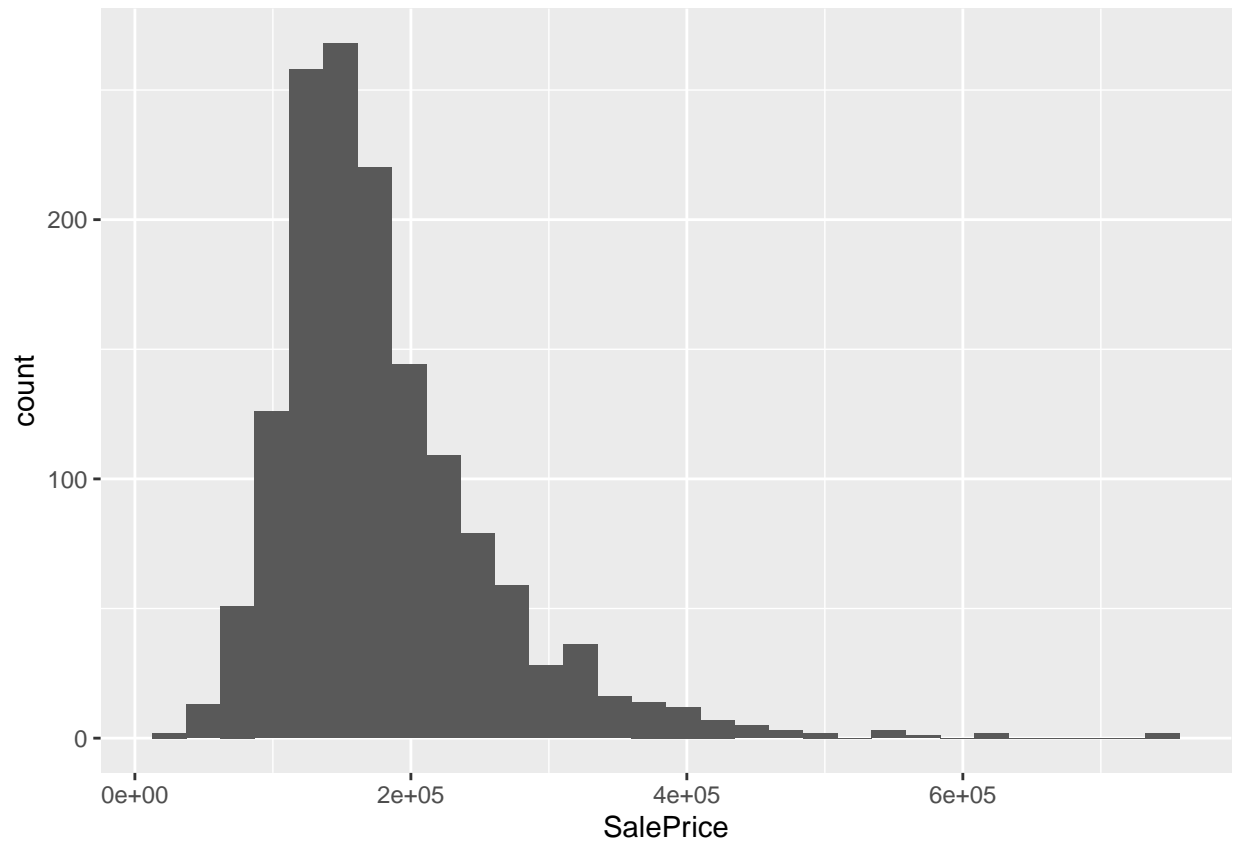












Primary Method - Random Forests

```
mean(dat$SalePrice)

## [1] 180921.2
median(dat$SalePrice)

## [1] 163000
sd(dat$SalePrice)

## [1] 79442.5
set.seed(1234567890)

n <- nrow(dat)

tv.split <- sample(rep(0:1, c(round(n*.3),n-2*round(n*.15))),n)

dat.train <- dat[tv.split==1,]

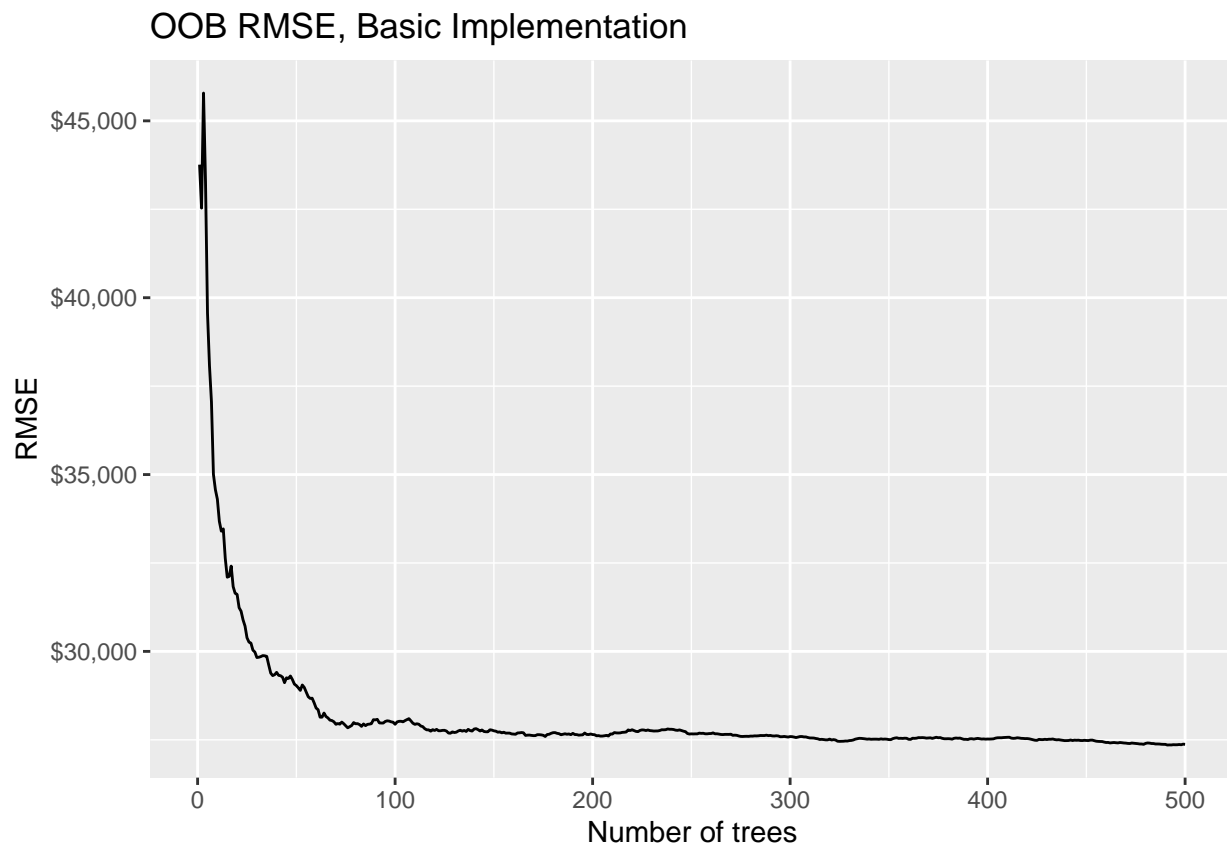
dat.valid <- dat[tv.split==0,]

housing.randomForest <- randomForest(formula = SalePrice ~ ., data = dat)

housing.randomForest
```



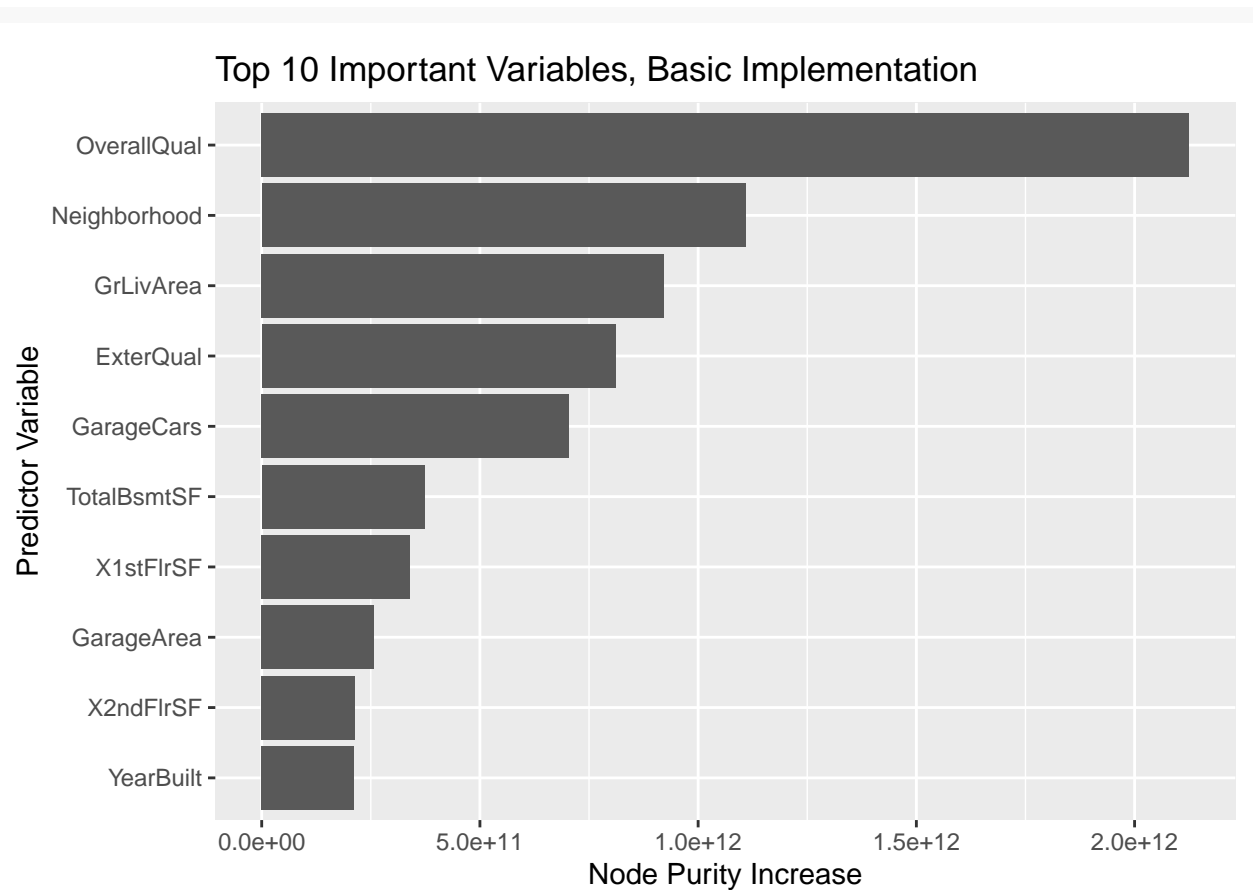
```
##
## Call:
## randomForest(formula = SalePrice ~ ., data = dat)
##           Type of random forest: regression
##           Number of trees: 500
## No. of variables tried at each split: 22
##
##           Mean of squared residuals: 749248979
##           % Var explained: 88.12
tibble(`Out of Bag Error` = sqrt(housing.randomForest$mse),
       ntrees = 1:housing.randomForest$ntree) %>%
  gather(Grouping, RMSE, -ntrees) %>%
  ggplot(aes(ntrees, RMSE)) +
  geom_line() +
  scale_y_continuous(labels = scales::dollar) +
  xlab("Number of trees") + ggtitle("OOB RMSE, Basic Implementation")
```



```
variable.Importance.Basic <- data.frame(housing.randomForest$importance) %>%
  arrange(desc(IncNodePurity)) %>%
  top_n(10)
```

```
## Selecting by IncNodePurity
```

```
(ggplot(data = variable.Importance.Basic,
  aes(x = reorder(row.names(variable.Importance.Basic), IncNodePurity), y = IncNodePurity))
  + geom_col() + coord_flip() + labs(title="Top 10 Important Variables, Basic Implementation",
    x = "Predictor Variable", y = "Node Purity Increase"))
```



```

minErrorTreesBasic <- which.min(housing.randomForest$mse)

minErrorTreesBasic

## [1] 491

minErrorValueBasic <- sqrt(housing.randomForest$mse[which.min(housing.randomForest$mse)])

minErrorValueBasic

## [1] 27352.79

meanErrorValueBasic <- sqrt(mean(housing.randomForest$mse))

meanErrorValueBasic

## [1] 28188.17

x.valid <- dat.valid[setdiff(names(dat.valid), "SalePrice")]

y.valid <- dat.valid$SalePrice

mean(y.valid)

## [1] 183505.8

median(y.valid)

```

```
## [1] 165250
sd(y.valid)

## [1] 81437.67
randomforest.trained <- randomForest(formula = SalePrice ~ ., data = dat.train)

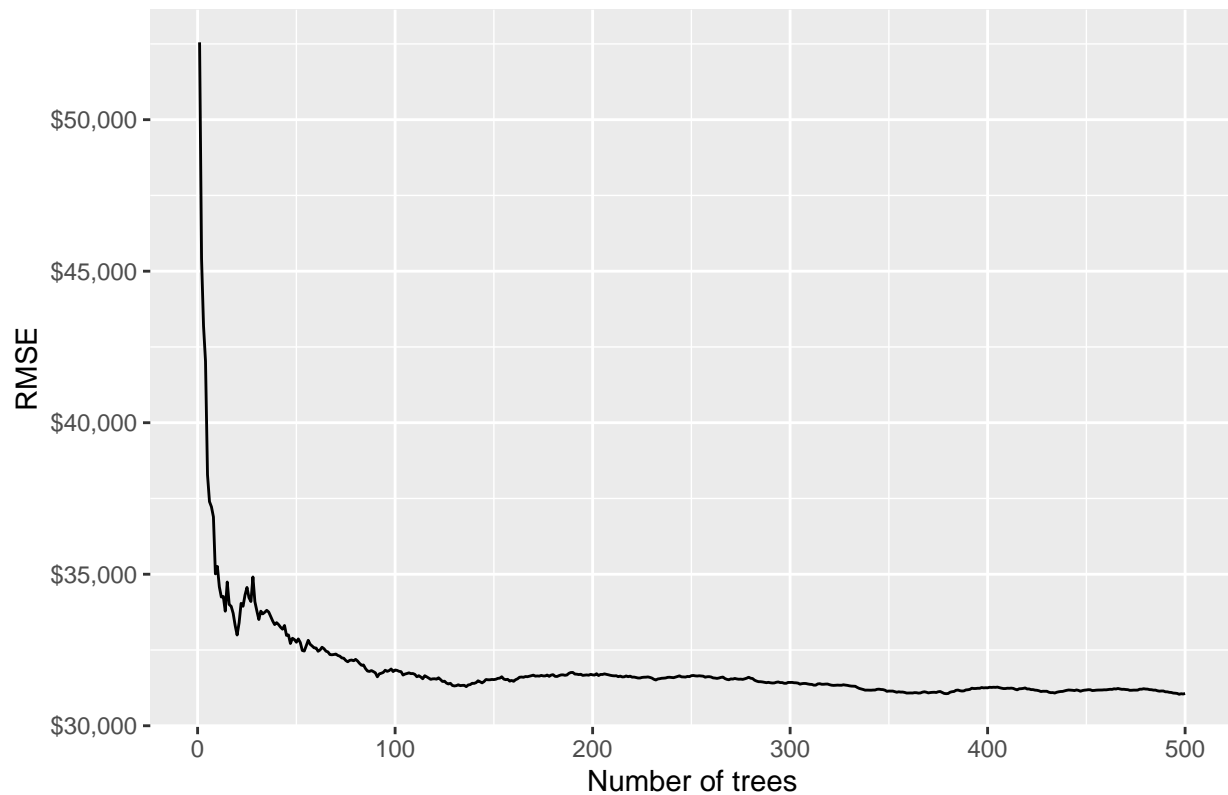
randomforest.trained

##
## Call:
## randomForest(formula = SalePrice ~ ., data = dat.train)
##           Type of random forest: regression
##           Number of trees: 500
## No. of variables tried at each split: 22
##
##           Mean of squared residuals: 964919761
##           % Var explained: 84.36
find.RMSPE <- function(true, predicted, df) {
  return(sqrt(sum((predicted - true)^2)/nrow(df)))
}

predictions.valid <- predict(randomforest.trained, dat.valid)

tibble(`Out of Bag Error` = sqrt(randomforest.trained$mse),
       ntrees = 1:randomforest.trained$ntree) %>%
  gather(Grouping, RMSE, -ntrees) %>%
  ggplot(aes(ntrees, RMSE)) +
  geom_line() +
  scale_y_continuous(labels = scales::dollar) +
  xlab("Number of trees") + ggtitle("OOB RMSE, Manual Validation")
```

OOB RMSE, Manual Validation

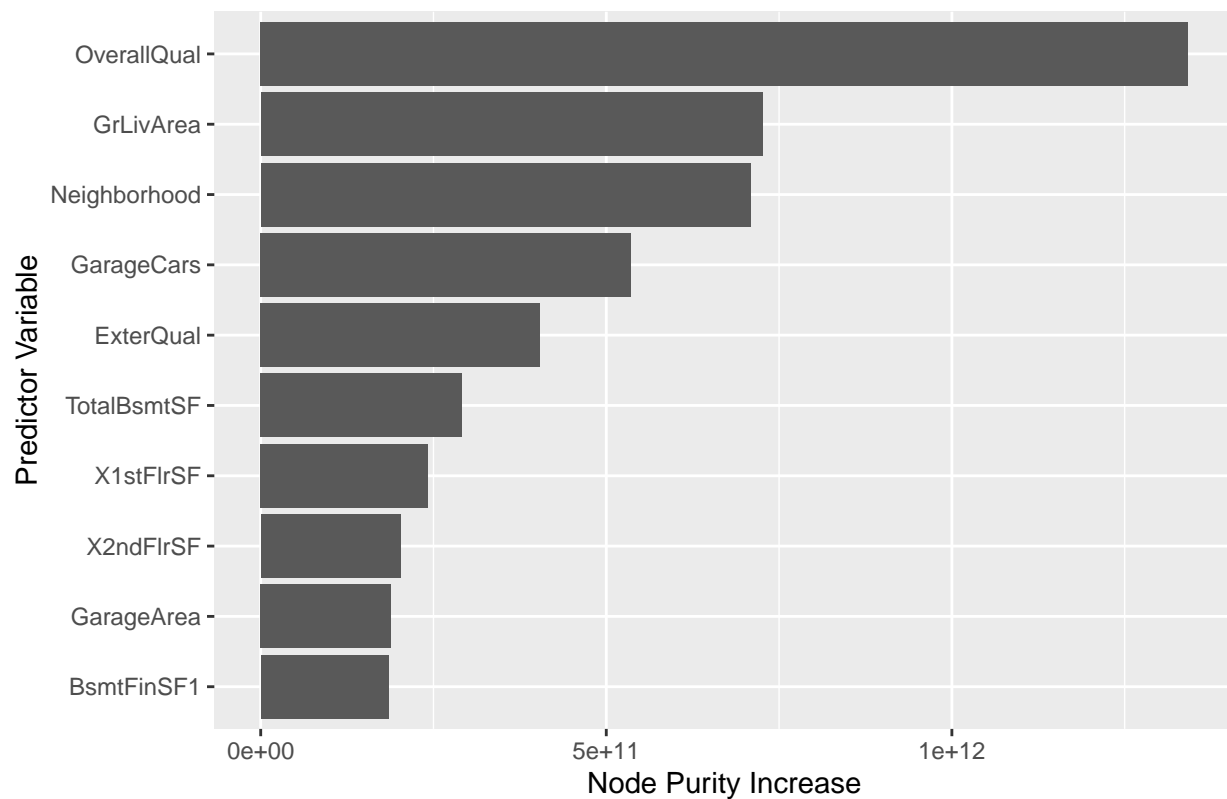


```
variable.Importance.trained <- data.frame(randomforest.trained$importance) %>%  
  arrange(desc(IncNodePurity)) %>%  
  top_n(10)
```

Selecting by IncNodePurity

```
(ggplot(data = variable.Importance.trained,  
  aes(x = reorder(row.names(variable.Importance.trained), IncNodePurity), y = IncNodePurity))  
  + geom_col() + coord_flip() + labs(title="Top 10 Important Variables, Manual Validation",  
    x = "Predictor Variable", y = "Node Purity Increase"))
```

Top 10 Important Variables, Manual Validation



```
minErrorTreesTrained <- which.min(randomforest.trained$mse)
```

```
minErrorTreesTrained
```

```
## [1] 497
```

```
minErrorValueTrained <- sqrt(randomforest.trained$mse[which.min(randomforest.trained$mse)])
```

```
minErrorValueTrained
```

```
## [1] 31038.16
```

```
meanErrorValueTrained <- sqrt(mean(randomforest.trained$mse))
```

```
meanErrorValueTrained
```

```
## [1] 31868.77
```

```
RMSPE <- find.RMSPE(dat.valid$SalePrice, predictions.valid, dat.valid)
```

```
RMSPE
```

```
## [1] 25529.17
```

```
randomforest.validated <- randomForest(formula = SalePrice ~ ., data = dat.train,
                                       xtest = x.valid, ytest = y.valid)
```

```
randomforest.validated
```

```
##
```

```
## Call:
## randomForest(formula = SalePrice ~ ., data = dat.train, xtest = x.valid, ytest = y.valid)
##           Type of random forest: regression
##           Number of trees: 500
## No. of variables tried at each split: 22
##
##           Mean of squared residuals: 952389999
##           % Var explained: 84.56
##           Test set MSE: 660213523
##           % Var explained: 90.04

minErrorTreesOOB <- which.min(randomforest.validated$mse)

minErrorTreesOOB

## [1] 436

minErrorValueOOB <- sqrt(randomforest.validated$mse[which.min(randomforest.validated$mse)])

minErrorValueOOB

## [1] 30792.35

meanErrorValueTrained <- sqrt(mean(randomforest.validated$mse))

meanErrorValueTrained

## [1] 32012.26

minErrorTreesValidation <- which.min(randomforest.validated$test$mse)

minErrorTreesValidation

## [1] 466

minErrorValueValidation <- sqrt(randomforest.validated$test$mse[which.min(randomforest.validated$test$mse)])

minErrorValueValidation

## [1] 25625.55

meanErrorValueValidation <- sqrt(mean(randomforest.validated$test$mse))

meanErrorValueValidation

## [1] 26246.8

tibble(`Out of Bag Error` = sqrt(randomforest.validated$mse),
       `Validation Error` = sqrt(randomforest.validated$test$mse),
       ntrees = 1:randomforest.validated$ntree) %>%
  gather(Grouping, RMSE, -ntrees) %>%
  ggplot(aes(ntrees, RMSE, color = Grouping)) +
  geom_line() +
  scale_y_continuous(labels = scales::dollar) +
  xlab("Number of trees") + ggtitle("OOB and Validation RMSE, Specified Validation Set")
```

OOB and Validation RMSE, Specified Validation Set



```
variable.Importance.validated <- data.frame(randomforest.validated$importance) %>%
  arrange(desc(IncNodePurity)) %>%
  top_n(10)
```

Selecting by IncNodePurity

```
(ggplot(data = variable.Importance.validated,
  aes(x = reorder(row.names(variable.Importance.validated), IncNodePurity), y = IncNodePurity))
  + geom_col() + coord_flip() + labs(title="Top 10 Important Variables, Specified Validation Set"
    x = "Predictor Variable", y = "Node Purity Increase"))
```

Top 10 Important Variables, Specified Validation Set

