House Prices: Advanced Regression Techniques

Table of Contents

[library & Data load step 2](#_Toc32140116)

[Exploring data analysis step 7](#_Toc32140117)

[target variable : SalePrice 7](#_Toc32140118)

[variables imputation 9](#_Toc32140119)

[Pool 10](#_Toc32140120)

[Miscellaneous features 11](#_Toc32140121)

[Alley 12](#_Toc32140122)

[Fence 13](#_Toc32140123)

[Fireplace 14](#_Toc32140124)

[LotFrontage 15](#_Toc32140125)

[Garages 17](#_Toc32140126)

[Basement Variables 20](#_Toc32140127)

[Mansonry veneer 24](#_Toc32140128)

[other variables 26](#_Toc32140129)

[variable engineering 31](#_Toc32140130)

[Remodel Y/N 31](#_Toc32140131)

[New house Y/N 33](#_Toc32140132)

[variable explanation 34](#_Toc32140133)

[correlation among numeric variables 35](#_Toc32140134)

[character variables 44](#_Toc32140135)

[preparing data step 48](#_Toc32140136)

[normalizing 48](#_Toc32140137)

[one-hot encodeing 49](#_Toc32140138)

[combining all variables 50](#_Toc32140139)

[logarithm target variable 50](#_Toc32140140)

[modeling step 52](#_Toc32140141)

[predicting house price in test set (final model) 53](#_Toc32140142)

[lasso 53](#_Toc32140143)

[randomforest 53](#_Toc32140144)

[gradient boost 54](#_Toc32140145)

[final submission step 57](#_Toc32140146)

kaggle address : <https://www.kaggle.com/zizonpingu>

**목적** 2006년부터 2010년간 Ames 지방의 주택 가격 데이터를 바탕으로 주택 가격 예측을 위한 분석.

# library & Data load step

library(tidyverse)  
library(skimr)  
library(plyr)  
library(scales)  
library(knitr)  
library(gridExtra)  
library(Rmisc)  
library(corrplot)  
library(ggrepel)  
library(caret)  
library(randomForest)  
library(gbm)  
library(xgboost)  
library(glmnet)  
library(Matrix)  
library(Metrics)  
library(e1071)

train <- read.csv("~/Github/kaggle\_house\_price/house-prices-advanced-regression-techniques/train.csv", stringsAsFactors = F)  
test <- read.csv("~/Github/kaggle\_house\_price/house-prices-advanced-regression-techniques/test.csv", stringsAsFactors = F)  
names(train)

## [1] "Id" "MSSubClass" "MSZoning" "LotFrontage"   
## [5] "LotArea" "Street" "Alley" "LotShape"   
## [9] "LandContour" "Utilities" "LotConfig" "LandSlope"   
## [13] "Neighborhood" "Condition1" "Condition2" "BldgType"   
## [17] "HouseStyle" "OverallQual" "OverallCond" "YearBuilt"   
## [21] "YearRemodAdd" "RoofStyle" "RoofMatl" "Exterior1st"   
## [25] "Exterior2nd" "MasVnrType" "MasVnrArea" "ExterQual"   
## [29] "ExterCond" "Foundation" "BsmtQual" "BsmtCond"   
## [33] "BsmtExposure" "BsmtFinType1" "BsmtFinSF1" "BsmtFinType2"   
## [37] "BsmtFinSF2" "BsmtUnfSF" "TotalBsmtSF" "Heating"   
## [41] "HeatingQC" "CentralAir" "Electrical" "X1stFlrSF"   
## [45] "X2ndFlrSF" "LowQualFinSF" "GrLivArea" "BsmtFullBath"   
## [49] "BsmtHalfBath" "FullBath" "HalfBath" "BedroomAbvGr"   
## [53] "KitchenAbvGr" "KitchenQual" "TotRmsAbvGrd" "Functional"   
## [57] "Fireplaces" "FireplaceQu" "GarageType" "GarageYrBlt"   
## [61] "GarageFinish" "GarageCars" "GarageArea" "GarageQual"   
## [65] "GarageCond" "PavedDrive" "WoodDeckSF" "OpenPorchSF"   
## [69] "EnclosedPorch" "X3SsnPorch" "ScreenPorch" "PoolArea"   
## [73] "PoolQC" "Fence" "MiscFeature" "MiscVal"   
## [77] "MoSold" "YrSold" "SaleType" "SaleCondition"  
## [81] "SalePrice"

ID 식별자를 따로 분류하고 all에는 ID를 제외한다. (예측에는 필요없기 때문)

testID <- test$Id  
train$Id <- NULL  
test$Id <- NULL

test$SalePrice <- NA  
all <- rbind(train,test)  
names(all)

## [1] "MSSubClass" "MSZoning" "LotFrontage" "LotArea"   
## [5] "Street" "Alley" "LotShape" "LandContour"   
## [9] "Utilities" "LotConfig" "LandSlope" "Neighborhood"   
## [13] "Condition1" "Condition2" "BldgType" "HouseStyle"   
## [17] "OverallQual" "OverallCond" "YearBuilt" "YearRemodAdd"   
## [21] "RoofStyle" "RoofMatl" "Exterior1st" "Exterior2nd"   
## [25] "MasVnrType" "MasVnrArea" "ExterQual" "ExterCond"   
## [29] "Foundation" "BsmtQual" "BsmtCond" "BsmtExposure"   
## [33] "BsmtFinType1" "BsmtFinSF1" "BsmtFinType2" "BsmtFinSF2"   
## [37] "BsmtUnfSF" "TotalBsmtSF" "Heating" "HeatingQC"   
## [41] "CentralAir" "Electrical" "X1stFlrSF" "X2ndFlrSF"   
## [45] "LowQualFinSF" "GrLivArea" "BsmtFullBath" "BsmtHalfBath"   
## [49] "FullBath" "HalfBath" "BedroomAbvGr" "KitchenAbvGr"   
## [53] "KitchenQual" "TotRmsAbvGrd" "Functional" "Fireplaces"   
## [57] "FireplaceQu" "GarageType" "GarageYrBlt" "GarageFinish"   
## [61] "GarageCars" "GarageArea" "GarageQual" "GarageCond"   
## [65] "PavedDrive" "WoodDeckSF" "OpenPorchSF" "EnclosedPorch"  
## [69] "X3SsnPorch" "ScreenPorch" "PoolArea" "PoolQC"   
## [73] "Fence" "MiscFeature" "MiscVal" "MoSold"   
## [77] "YrSold" "SaleType" "SaleCondition" "SalePrice"

skim(all)

Data summary

|  |  |
| --- | --- |
| Name | all |
| Number of rows | 2919 |
| Number of columns | 80 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 43 |
| numeric | 37 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| MSZoning | 4 | 1.00 | 2 | 7 | 0 | 5 | 0 |
| Street | 0 | 1.00 | 4 | 4 | 0 | 2 | 0 |
| Alley | 2721 | 0.07 | 4 | 4 | 0 | 2 | 0 |
| LotShape | 0 | 1.00 | 3 | 3 | 0 | 4 | 0 |
| LandContour | 0 | 1.00 | 3 | 3 | 0 | 4 | 0 |
| Utilities | 2 | 1.00 | 6 | 6 | 0 | 2 | 0 |
| LotConfig | 0 | 1.00 | 3 | 7 | 0 | 5 | 0 |
| LandSlope | 0 | 1.00 | 3 | 3 | 0 | 3 | 0 |
| Neighborhood | 0 | 1.00 | 5 | 7 | 0 | 25 | 0 |
| Condition1 | 0 | 1.00 | 4 | 6 | 0 | 9 | 0 |
| Condition2 | 0 | 1.00 | 4 | 6 | 0 | 8 | 0 |
| BldgType | 0 | 1.00 | 4 | 6 | 0 | 5 | 0 |
| HouseStyle | 0 | 1.00 | 4 | 6 | 0 | 8 | 0 |
| RoofStyle | 0 | 1.00 | 3 | 7 | 0 | 6 | 0 |
| RoofMatl | 0 | 1.00 | 4 | 7 | 0 | 8 | 0 |
| Exterior1st | 1 | 1.00 | 5 | 7 | 0 | 15 | 0 |
| Exterior2nd | 1 | 1.00 | 5 | 7 | 0 | 16 | 0 |
| MasVnrType | 24 | 0.99 | 4 | 7 | 0 | 4 | 0 |
| ExterQual | 0 | 1.00 | 2 | 2 | 0 | 4 | 0 |
| ExterCond | 0 | 1.00 | 2 | 2 | 0 | 5 | 0 |
| Foundation | 0 | 1.00 | 4 | 6 | 0 | 6 | 0 |
| BsmtQual | 81 | 0.97 | 2 | 2 | 0 | 4 | 0 |
| BsmtCond | 82 | 0.97 | 2 | 2 | 0 | 4 | 0 |
| BsmtExposure | 82 | 0.97 | 2 | 2 | 0 | 4 | 0 |
| BsmtFinType1 | 79 | 0.97 | 3 | 3 | 0 | 6 | 0 |
| BsmtFinType2 | 80 | 0.97 | 3 | 3 | 0 | 6 | 0 |
| Heating | 0 | 1.00 | 4 | 5 | 0 | 6 | 0 |
| HeatingQC | 0 | 1.00 | 2 | 2 | 0 | 5 | 0 |
| CentralAir | 0 | 1.00 | 1 | 1 | 0 | 2 | 0 |
| Electrical | 1 | 1.00 | 3 | 5 | 0 | 5 | 0 |
| KitchenQual | 1 | 1.00 | 2 | 2 | 0 | 4 | 0 |
| Functional | 2 | 1.00 | 3 | 4 | 0 | 7 | 0 |
| FireplaceQu | 1420 | 0.51 | 2 | 2 | 0 | 5 | 0 |
| GarageType | 157 | 0.95 | 6 | 7 | 0 | 6 | 0 |
| GarageFinish | 159 | 0.95 | 3 | 3 | 0 | 3 | 0 |
| GarageQual | 159 | 0.95 | 2 | 2 | 0 | 5 | 0 |
| GarageCond | 159 | 0.95 | 2 | 2 | 0 | 5 | 0 |
| PavedDrive | 0 | 1.00 | 1 | 1 | 0 | 3 | 0 |
| PoolQC | 2909 | 0.00 | 2 | 2 | 0 | 3 | 0 |
| Fence | 2348 | 0.20 | 4 | 5 | 0 | 4 | 0 |
| MiscFeature | 2814 | 0.04 | 4 | 4 | 0 | 4 | 0 |
| SaleType | 1 | 1.00 | 2 | 5 | 0 | 9 | 0 |
| SaleCondition | 0 | 1.00 | 6 | 7 | 0 | 6 | 0 |

**Variable type: numeric**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| MSSubClass | 0 | 1.00 | 57.14 | 42.52 | 20 | 20.0 | 50.0 | 70.0 | 190 | ▇▅▂▁▁ |
| LotFrontage | 486 | 0.83 | 69.31 | 23.34 | 21 | 59.0 | 68.0 | 80.0 | 313 | ▇▃▁▁▁ |
| LotArea | 0 | 1.00 | 10168.11 | 7887.00 | 1300 | 7478.0 | 9453.0 | 11570.0 | 215245 | ▇▁▁▁▁ |
| OverallQual | 0 | 1.00 | 6.09 | 1.41 | 1 | 5.0 | 6.0 | 7.0 | 10 | ▁▂▇▅▁ |
| OverallCond | 0 | 1.00 | 5.56 | 1.11 | 1 | 5.0 | 5.0 | 6.0 | 9 | ▁▁▇▅▁ |
| YearBuilt | 0 | 1.00 | 1971.31 | 30.29 | 1872 | 1953.5 | 1973.0 | 2001.0 | 2010 | ▁▂▃▆▇ |
| YearRemodAdd | 0 | 1.00 | 1984.26 | 20.89 | 1950 | 1965.0 | 1993.0 | 2004.0 | 2010 | ▅▂▂▃▇ |
| MasVnrArea | 23 | 0.99 | 102.20 | 179.33 | 0 | 0.0 | 0.0 | 164.0 | 1600 | ▇▁▁▁▁ |
| BsmtFinSF1 | 1 | 1.00 | 441.42 | 455.61 | 0 | 0.0 | 368.5 | 733.0 | 5644 | ▇▁▁▁▁ |
| BsmtFinSF2 | 1 | 1.00 | 49.58 | 169.21 | 0 | 0.0 | 0.0 | 0.0 | 1526 | ▇▁▁▁▁ |
| BsmtUnfSF | 1 | 1.00 | 560.77 | 439.54 | 0 | 220.0 | 467.0 | 805.5 | 2336 | ▇▅▂▁▁ |
| TotalBsmtSF | 1 | 1.00 | 1051.78 | 440.77 | 0 | 793.0 | 989.5 | 1302.0 | 6110 | ▇▃▁▁▁ |
| X1stFlrSF | 0 | 1.00 | 1159.58 | 392.36 | 334 | 876.0 | 1082.0 | 1387.5 | 5095 | ▇▃▁▁▁ |
| X2ndFlrSF | 0 | 1.00 | 336.48 | 428.70 | 0 | 0.0 | 0.0 | 704.0 | 2065 | ▇▃▂▁▁ |
| LowQualFinSF | 0 | 1.00 | 4.69 | 46.40 | 0 | 0.0 | 0.0 | 0.0 | 1064 | ▇▁▁▁▁ |
| GrLivArea | 0 | 1.00 | 1500.76 | 506.05 | 334 | 1126.0 | 1444.0 | 1743.5 | 5642 | ▇▇▁▁▁ |
| BsmtFullBath | 2 | 1.00 | 0.43 | 0.52 | 0 | 0.0 | 0.0 | 1.0 | 3 | ▇▆▁▁▁ |
| BsmtHalfBath | 2 | 1.00 | 0.06 | 0.25 | 0 | 0.0 | 0.0 | 0.0 | 2 | ▇▁▁▁▁ |
| FullBath | 0 | 1.00 | 1.57 | 0.55 | 0 | 1.0 | 2.0 | 2.0 | 4 | ▁▇▇▁▁ |
| HalfBath | 0 | 1.00 | 0.38 | 0.50 | 0 | 0.0 | 0.0 | 1.0 | 2 | ▇▁▅▁▁ |
| BedroomAbvGr | 0 | 1.00 | 2.86 | 0.82 | 0 | 2.0 | 3.0 | 3.0 | 8 | ▁▇▂▁▁ |
| KitchenAbvGr | 0 | 1.00 | 1.04 | 0.21 | 0 | 1.0 | 1.0 | 1.0 | 3 | ▁▇▁▁▁ |
| TotRmsAbvGrd | 0 | 1.00 | 6.45 | 1.57 | 2 | 5.0 | 6.0 | 7.0 | 15 | ▁▇▂▁▁ |
| Fireplaces | 0 | 1.00 | 0.60 | 0.65 | 0 | 0.0 | 1.0 | 1.0 | 4 | ▇▇▁▁▁ |
| GarageYrBlt | 159 | 0.95 | 1978.11 | 25.57 | 1895 | 1960.0 | 1979.0 | 2002.0 | 2207 | ▂▇▁▁▁ |
| GarageCars | 1 | 1.00 | 1.77 | 0.76 | 0 | 1.0 | 2.0 | 2.0 | 5 | ▅▇▂▁▁ |
| GarageArea | 1 | 1.00 | 472.87 | 215.39 | 0 | 320.0 | 480.0 | 576.0 | 1488 | ▃▇▃▁▁ |
| WoodDeckSF | 0 | 1.00 | 93.71 | 126.53 | 0 | 0.0 | 0.0 | 168.0 | 1424 | ▇▁▁▁▁ |
| OpenPorchSF | 0 | 1.00 | 47.49 | 67.58 | 0 | 0.0 | 26.0 | 70.0 | 742 | ▇▁▁▁▁ |
| EnclosedPorch | 0 | 1.00 | 23.10 | 64.24 | 0 | 0.0 | 0.0 | 0.0 | 1012 | ▇▁▁▁▁ |
| X3SsnPorch | 0 | 1.00 | 2.60 | 25.19 | 0 | 0.0 | 0.0 | 0.0 | 508 | ▇▁▁▁▁ |
| ScreenPorch | 0 | 1.00 | 16.06 | 56.18 | 0 | 0.0 | 0.0 | 0.0 | 576 | ▇▁▁▁▁ |
| PoolArea | 0 | 1.00 | 2.25 | 35.66 | 0 | 0.0 | 0.0 | 0.0 | 800 | ▇▁▁▁▁ |
| MiscVal | 0 | 1.00 | 50.83 | 567.40 | 0 | 0.0 | 0.0 | 0.0 | 17000 | ▇▁▁▁▁ |
| MoSold | 0 | 1.00 | 6.21 | 2.71 | 1 | 4.0 | 6.0 | 8.0 | 12 | ▅▆▇▃▃ |
| YrSold | 0 | 1.00 | 2007.79 | 1.31 | 2006 | 2007.0 | 2008.0 | 2009.0 | 2010 | ▇▇▇▇▃ |
| SalePrice | 1459 | 0.50 | 180921.20 | 79442.50 | 34900 | 129975.0 | 163000.0 | 214000.0 | 755000 | ▇▅▁▁▁ |

dim(all)

## [1] 2919 80

# Exploring data analysis step

## target variable : SalePrice

summary(all$SalePrice)

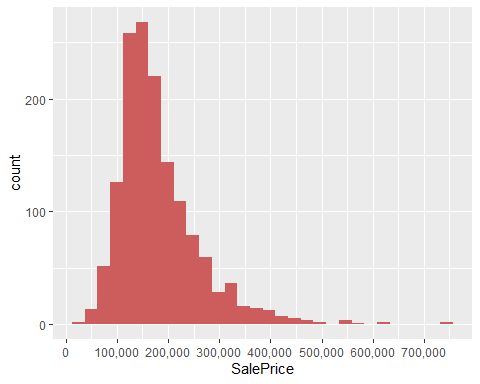
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 34900 129975 163000 180921 214000 755000 1459

skewness(all$SalePrice, na.rm=T)

## [1] 1.879009

ggplot(all[!is.na(all$SalePrice),], aes(x=SalePrice)) +  
 geom\_histogram(fill="indianred") +  
 scale\_x\_continuous(breaks = seq(0,800000, by=100000), labels = scales::comma) #library(scales)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



SalePrice의 분포를 보면, skewness가 1.88로 right-skewed된 분포임을 알 수 있다. 분산 안정화를 위해 log를 취하여 그래프를 그려보면,

summary(log(all$SalePrice))

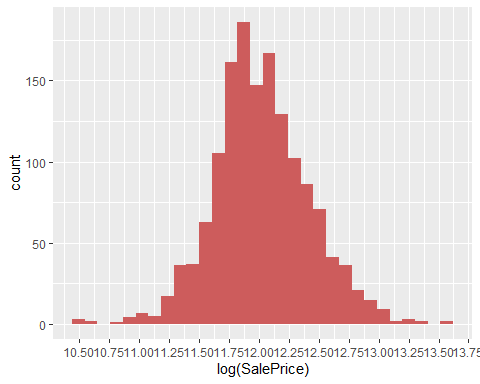
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 10.46 11.78 12.00 12.02 12.27 13.53 1459

skewness(log(all$SalePrice), na.rm=T)

## [1] 0.1210859

ggplot(all[!is.na(all$SalePrice),], aes(x=log(SalePrice))) +  
 geom\_histogram(fill="indianred") +  
 scale\_x\_continuous(breaks = seq(10,15, by=0.25))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



평균이 12.02이고 skewness가 0.12로 skewed된 경향이 감소했음을 알 수 있었다.

## variables imputation

이제 variables (variables)를 살펴보기 전에, 결측값이 존재하는지 아닌지 알아보았다.

NAs <- function(df){  
 aa <- sapply(df, function(x){sum(is.na(x))})  
 return(sort(aa[which(aa>0)], decreasing = T))  
}  
  
NAs(all)

## PoolQC MiscFeature Alley Fence SalePrice FireplaceQu   
## 2909 2814 2721 2348 1459 1420   
## LotFrontage GarageYrBlt GarageFinish GarageQual GarageCond GarageType   
## 486 159 159 159 159 157   
## BsmtCond BsmtExposure BsmtQual BsmtFinType2 BsmtFinType1 MasVnrType   
## 82 82 81 80 79 24   
## MasVnrArea MSZoning Utilities BsmtFullBath BsmtHalfBath Functional   
## 23 4 2 2 2 2   
## Exterior1st Exterior2nd BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF   
## 1 1 1 1 1 1   
## Electrical KitchenQual GarageCars GarageArea SaleType   
## 1 1 1 1 1

관련있는 변수끼리 묶에 결측치 처리 (imputation)을 한다. 참고로 SalePrice의 1459개는 test의 갯수로 예측해야 하는 종속변수이므로 imputation하지 않는다.

#nrow(all[!complete.cases(all),])  
#sum(!complete.cases(all))

결측치가 없는 행의 개수를 출력할 때 complete.cases()를 사용한다.

이제부터 결측이 있는 각 variables를 깊게 살펴보고, 결측값을 보정한다.

### Pool

PoolQC : Pool quality

PoolQC는 전체 2919개 중 10개를 제외한 나머지는 결측으로 처리되어 있다. PoolArea도 0이므로 이를 NA에서 None, 즉 pool이 없다고 바꿔준다.

table(all$PoolArea, all$PoolQC, useNA = "ifany")

##   
## Ex Fa Gd <NA>  
## 0 0 0 0 2906  
## 144 1 0 0 0  
## 228 1 0 0 0  
## 368 0 0 0 1  
## 444 0 0 0 1  
## 480 0 0 1 0  
## 512 1 0 0 0  
## 519 0 1 0 0  
## 555 1 0 0 0  
## 561 0 0 0 1  
## 576 0 0 1 0  
## 648 0 1 0 0  
## 738 0 0 1 0  
## 800 0 0 1 0

all$PoolQC[is.na(all$PoolQC)] <- 'None'

* 에러가 날 경우
* invalid factor level, NA generated

이거는 이미 PoolQC의 class가 factor이며, level이 Ex, Fa, Gd로 설정되있는 상태에서, None 이라는 새로운 level을 넣을 경우 에러로 뜬다. 따라서 처음 데이터를 불러올 때 “stringsAsFactors = F” 설정을 넣어 str class로 범주형 변수를 불러온다. (그 증거로, “stringsAsFactors = F” 없는 경우 str() 함수를 쓸 경우 범주형 변수가 Factor로 설정되는 반면, “stringsAsFactors = F” 있는 경우 범주형 변수가 str 형식으로 설정된다.) 그런 뒤 각 변수의 imputation과정을 거치고 factor로 바꿔준다.

table(all$PoolQC)

##   
## Ex Fa Gd None   
## 4 2 4 2909

PoolArea 에 값이 있으면 PoolQC는 무조건 1 이상이어야 하지만, 0의 값을 갖는 값들이 있었다.

all[all$PoolArea>0 & all$PoolQC=="None", c("PoolArea","PoolQC")]

## PoolArea PoolQC  
## 2421 368 None  
## 2504 444 None  
## 2600 561 None

all$PoolArea[2421] <- 0   
all$PoolArea[2504] <- 0   
all$PoolArea[2600] <- 0

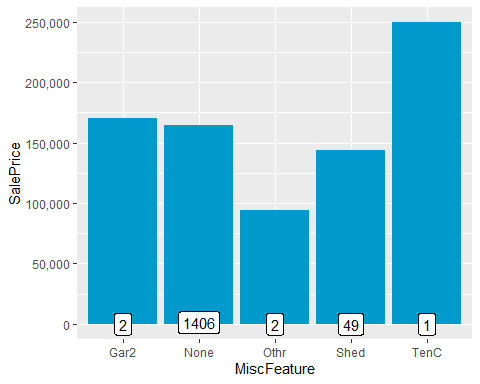
### Miscellaneous features

집에 기타 부대시설이 있는지 나타내는 변수이다. (엘레베이터, 두 번째 차고, 헛간, 테니스장 등등)

table(all$MiscFeature, useNA = "ifany")

##   
## Gar2 Othr Shed TenC <NA>   
## 5 4 95 1 2814

all$MiscFeature[is.na(all$MiscFeature)] <- 'None'  
  
ggplot(all[!is.na(all$SalePrice),], aes(x=MiscFeature, y=SalePrice)) +  
 geom\_bar(stat='summary', fun.y='median', fill='deepskyblue3') +  
 scale\_y\_continuous(breaks = seq(0,300000, by=50000), labels = scales::comma) +  
 geom\_label(stat="count", aes(label=..count.., y=..count..))



부대시설이 있다고 해서 SalePrice에 차이가 극명하게 드러나지 않았다. 하지만 테니스 코트가 있는 집의 경우 SalePrice 값이 현저히 높은것으로 나타났다.

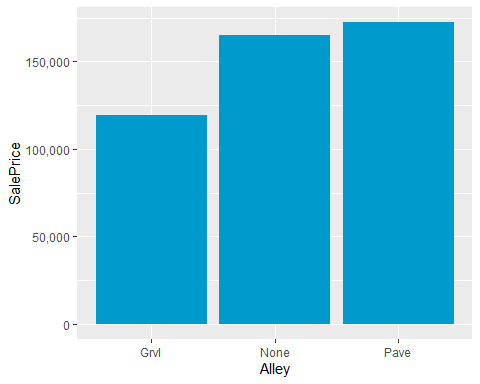
### Alley

Alley는 골목을 나타내는 변수로 대부분이 결측이었다. 결측의 경우 None으로 코딩하였다.

table(all$Alley, useNA = "always")

##   
## Grvl Pave <NA>   
## 120 78 2721

all$Alley[is.na(all$Alley)] <- 'None'  
  
ggplot(all[!is.na(all$SalePrice),], aes(x=Alley, y=SalePrice)) +  
 geom\_bar(stat='summary', fun.y='median', fill='deepskyblue3') +  
 scale\_y\_continuous(breaks=seq(0,200000, by=50000), labels = scales::comma)

 골목이 없는 경우와 포장된(Pave) 경우보다 골목에 자갈이(Grvl)있는 경우의 SalePrice가 더 낮았다.

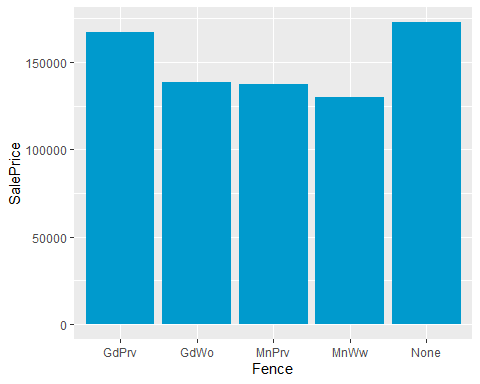
### Fence

table(all$Fence, useNA="always")

##   
## GdPrv GdWo MnPrv MnWw <NA>   
## 118 112 329 12 2348

울타리 퀄리티를 나타내는 변수(Fence)는 NA, NnWw, GdWo, MnPrv, GdPrv 순이다. 울타리가 결측인 값들은 울타리가 없는 집으로 보정하였다.

all$Fence[is.na(all$Fence)] <- 'None'  
  
ggplot(all[!is.na(all$SalePrice),], aes(x=Fence, y=SalePrice)) +  
 geom\_bar(stat='summary', fun.y='median', fill='deepskyblue3')



울타리 퀄리티가 좋을수록 SalePrice가 약간 높아지는 경향이 있지만, None인 변수가 가장 높은 값을 가진 것으로 보아 Fence는 SalePrice에 영향도가 낮을 것이라고 예상하였다.

### Fireplace

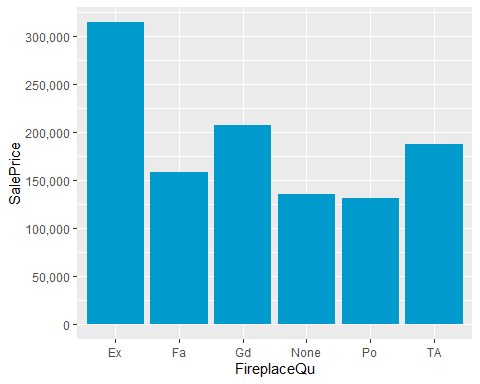
벽난로의 퀄리티를 나타내는 변수로, 결측은 벽난로가 없다고 보정하였다.

table(all$FireplaceQu, useNA="always")

##   
## Ex Fa Gd Po TA <NA>   
## 43 74 744 46 592 1420

all$FireplaceQu[is.na(all$FireplaceQu)] <- 'None'

ggplot(all[!is.na(all$SalePrice),], aes(x=FireplaceQu, y=SalePrice)) +  
 geom\_bar(stat='summary', fun.y='median', fill='deepskyblue3') +  
 scale\_y\_continuous(breaks=seq(0, 400000, by=50000), labels=comma)



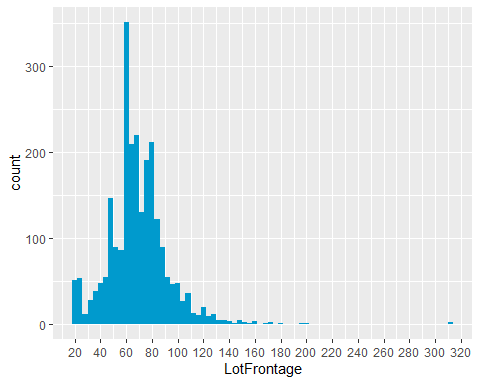
벽난로는 퀄리티가 증가할수록 SalePrice가 증가하는 경향을 가지는 것을 알 수 있었다.

### LotFrontage

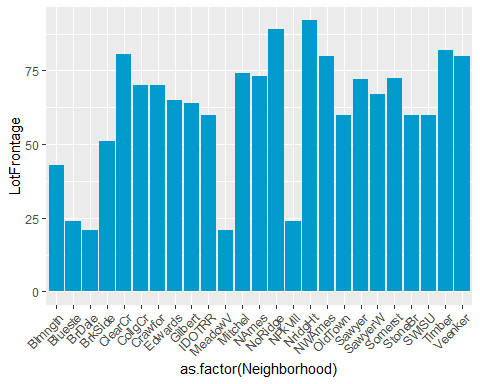
LotFrontage는 집 앞에 연결된 거리의 크기를 나타내는데, 486개의 결측은 근처 주요 건물(Physical location)을 나타내는 Neighborhood 변수와 관련이 있어 보였다. 각 Neighborhood의 범주에 따라 LotFrontage의 중앙값을 살펴보았다. LotFrontage가 조금 right-skewed된 경향이 있어 중앙값을 사용하였다.

ggplot(all, aes(x=LotFrontage)) +  
 geom\_histogram(fill='deepskyblue3', binwidth = 4) +  
 scale\_x\_continuous(breaks=seq(0,320,by=20))

## Warning: Removed 486 rows containing non-finite values (stat\_bin).



ggplot(all[!is.na(all$LotFrontage),], aes(x=as.factor(Neighborhood), y=LotFrontage)) +  
 geom\_bar(stat='summary', fun.y = "median", fill='deepskyblue3') +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



Neighborhood의 범주 중 LotFrontage가 높은 곳의 중앙값은 약 90에 가까웠고 (NridgHt) 낮은 곳의 중앙값은 약 22였다 (BrDale). 따라서 Neighborhood의 각 범주에서의 LotFrontage의 중앙값으로 결측을 보정하였다.

for (i in 1:nrow(all)){  
 if(is.na(all$LotFrontage[i])){  
 all$LotFrontage[i] <- as.integer(median(all$LotFrontage[all$Neighborhood==all$Neighborhood[i]], na.rm=TRUE))   
 }  
}

### Garages

# same to : NAs(all[,names(all) %in% names(all)[grep("Garage", names(all))]])  
NAs(all[,grep("^Garage",names(all))])

## GarageYrBlt GarageFinish GarageQual GarageCond GarageType GarageCars   
## 159 159 159 159 157 1   
## GarageArea   
## 1

Garage에 관련된 변수는 총 7개로 GarageType, GarageYrBlt, GarageFinish, GarageCars, GarageArea, GarageQual, GaragaCond이다. 각각은 다 결측이 있었으므로 imputation을 실행하기로 했다.

grep 함수와 NAs 함수로 Garage로 시작하는 변수의 NA 개수를 반환하게 한다.

**GarageYrBlt**

먼저 GarageYrBlt를 보면, 집이 지어졌을 때와 집이 리모델링했을 때를 나타내는 (각각 YearBuilt, YearRemodAdd) 변수랑 비교해보았다. 만약 리모델링을 하지 않았다면 YearRemodAdd는 YearBuilt와 값이 같다.

nrow(all)

## [1] 2919

c(length(which(all$YearBuilt==all$YearRemodAdd)), length(which(all$YearBuilt==all$YearRemodAdd))/nrow(all))

## [1] 1560.0000000 0.5344296

c(length(which(all$YearBuilt==all$GarageYrBlt)), length(which(all$YearBuilt==all$GarageYrBlt))/nrow(all))

## [1] 2216.0000000 0.7591641

전체 2919중 집이 지어진 연도와 리모델링 연도가 같은 데이터는 1560개로 전체 53%이고 (즉 리모델링하지 않은 집이 53%), 집을 지을때 차고를 같이 지은 집은 2216개로 전체 약 75%였다. 집이 지어질 때 차고도 같이 짓는 경우가 많고, 집을 리모델링 할때 차고도 리모델링 하는 경우는 적다고 판단하여 159개의 GarageYrBlt 결측치는 집을 지을 때 차고도 같이 지었다고, 즉 GarageYrBlt==YearBuilt로 imputation 하였다.

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

**GarageType, GarageFinish, GarageCond, GarageQual, GarageCars, GarageArea**

다음으로, GarageType은 157개의 결측인데, 다른 159개의 결측치와 다른 2개가 무엇인지 데이터 탐색해보았다.

# type, finish, cond, qual  
length(which(is.na(all$GarageType) & is.na(all$GarageFinish) & is.na(all$GarageCond) & is.na(all$GarageQual)))

## [1] 157

all[!is.na(all$GarageType) & is.na(all$GarageFinish),   
 c('GarageCars', 'GarageArea', 'GarageType', 'GarageCond', 'GarageQual', 'GarageFinish')]

## GarageCars GarageArea GarageType GarageCond GarageQual GarageFinish  
## 2127 1 360 Detchd <NA> <NA> <NA>  
## 2577 NA NA Detchd <NA> <NA> <NA>

159개 중 157개는 모두 NA 였다.

2127와 2577번 째 관찰값은 GarageCond, GarageQual, GarageFinish가 결측임에도 불구하고 GarageType이 Detchd로 값이 존재했다. 또한 2577은 GarageCars, GarageArea도 결측이었다. 2127은 차고가 있는 것, 2577은 차고가 없는 것으로 판단하여 2127에 GarageCond, GarageQual, GarageFinish의 mode로 보정하였고, 2577의 GarageType(Detchd)은 NA로 대체하였다.

#2127  
all$GarageCond[2127] <- names(sort(-table(all$GarageCond)))[1]  
all$GarageQual[2127] <- names(sort(-table(all$GarageQual)))[1]  
all$GarageFinish[2127] <- names(sort(-table(all$GarageFinish)))[1]  
  
all[2127, c('GarageYrBlt', 'GarageCars', 'GarageArea', 'GarageType', 'GarageCond', 'GarageQual', 'GarageFinish')]

## GarageYrBlt GarageCars GarageArea GarageType GarageCond GarageQual  
## 2127 1910 1 360 Detchd TA TA  
## GarageFinish  
## 2127 Unf

#2577  
all$GarageCars[2577] <- 0  
all$GarageArea[2577] <- 0  
all$GarageType[2577] <- NA  
  
all[2577, c('GarageYrBlt', 'GarageCars', 'GarageArea', 'GarageType', 'GarageCond', 'GarageQual', 'GarageFinish')]

## GarageYrBlt GarageCars GarageArea GarageType GarageCond GarageQual  
## 2577 1923 0 0 <NA> <NA> <NA>  
## GarageFinish  
## 2577 <NA>

length(which(is.na(all$GarageType) & is.na(all$GarageFinish) & is.na(all$GarageCond) & is.na(all$GarageQual)))

## [1] 158

NAs(all[,grep("^Garage",names(all))])

## GarageType GarageFinish GarageQual GarageCond   
## 158 158 158 158

나머지 157+1개에 대해서는 차고가 없는 것으로 보정하였다. 차고가 없음은 NA에서 No Garage, None 으로 바꿔주었다.

all$GarageType[is.na(all$GarageType)] <- 'No Garage'  
table(all$GarageType, useNA="always")

##   
## 2Types Attchd Basment BuiltIn CarPort Detchd No Garage <NA>   
## 23 1723 36 186 15 778 158 0

all$GarageFinish[is.na(all$GarageFinish)] <- 'None'  
table(all$GarageFinish, useNA="always")

##   
## Fin None RFn Unf <NA>   
## 719 158 811 1231 0

all$GarageQual[is.na(all$GarageQual)] <- 'None'  
table(all$GarageQual, useNA="always")

##   
## Ex Fa Gd None Po TA <NA>   
## 3 124 24 158 5 2605 0

all$GarageCond[is.na(all$GarageCond)] <- 'None'  
table(all$GarageCond, useNA="always")

##   
## Ex Fa Gd None Po TA <NA>   
## 3 74 15 158 14 2655 0

### Basement Variables

NAs(all[,grep("Bsmt",names(all))])

## BsmtCond BsmtExposure BsmtQual BsmtFinType2 BsmtFinType1 BsmtFullBath   
## 82 82 81 80 79 2   
## BsmtHalfBath BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF   
## 2 1 1 1 1

**BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2**

먼저 “BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2” 5개 변수에 79개 NA가 같은 NA를 가지는 행인지 알아보고, 그렇지 않은 행을 추출해보면,

length(which(is.na(all$BsmtQual) & is.na(all$BsmtCond) & is.na(all$BsmtExposure) & is.na(all$BsmtFinType1) & is.na(all$BsmtFinType2)))

## [1] 79

all[!is.na(all$BsmtFinType1) & (is.na(all$BsmtCond)|is.na(all$BsmtQual)|is.na(all$BsmtExposure)|is.na(all$BsmtFinType2)),   
 c('BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2')]

## BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2  
## 333 Gd TA No GLQ <NA>  
## 949 Gd TA <NA> Unf Unf  
## 1488 Gd TA <NA> Unf Unf  
## 2041 Gd <NA> Mn GLQ Rec  
## 2186 TA <NA> No BLQ Unf  
## 2218 <NA> Fa No Unf Unf  
## 2219 <NA> TA No Unf Unf  
## 2349 Gd TA <NA> Unf Unf  
## 2525 TA <NA> Av ALQ Unf

BsmtFinType1의 79개는 “BsmtQual, BsmtCond, BsmtExposurem BsmtFinType1m BsmtFinType2” 모두 NA 이고, 부분적으로 NA가 존재한 9개 관찰값의 각 NA를 각 변수의 최빈값으로 보정해주었다.

#Imputing modes.  
all$BsmtFinType2[333] <- names(sort(-table(all$BsmtFinType2)))[1]  
all$BsmtExposure[c(949, 1488, 2349)] <- names(sort(-table(all$BsmtExposure)))[1]  
all$BsmtCond[c(2041, 2186, 2525)] <- names(sort(-table(all$BsmtCond)))[1]  
all$BsmtQual[c(2218, 2219)] <- names(sort(-table(all$BsmtQual)))[1]

이제 공통적인 79개에 대한 결측치를 보정한다. 79개는 모두 Bsmt에 관련된 변수에 NA였다. 따라서 이 집들은 Bsmt(지하실)이 없는 집이라고 가정하여 결측치 보정을 시행하였다.

all$BsmtQual[is.na(all$BsmtQual)] <- 'None'  
table(all$BsmtQual, useNA = "always")

##   
## Ex Fa Gd None TA <NA>   
## 258 88 1209 79 1285 0

all$BsmtCond[is.na(all$BsmtCond)] <- 'None'  
table(all$BsmtCond, useNA = "always")

##   
## Fa Gd None Po TA <NA>   
## 104 122 79 5 2609 0

all$BsmtExposure[is.na(all$BsmtExposure)] <- 'None'  
table(all$BsmtExposure, useNA = "always")

##   
## Av Gd Mn No None <NA>   
## 418 276 239 1907 79 0

all$BsmtFinType1[is.na(all$BsmtFinType1)] <- 'None'  
table(all$BsmtFinType1, useNA = "always")

##   
## ALQ BLQ GLQ LwQ None Rec Unf <NA>   
## 429 269 849 154 79 288 851 0

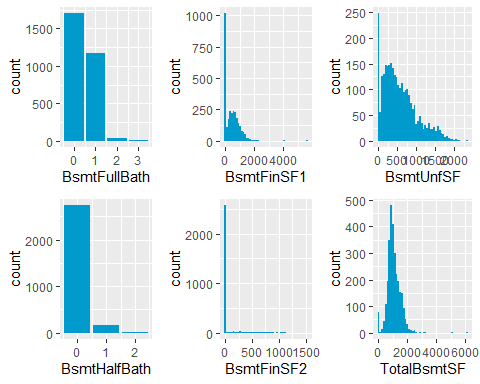
all$BsmtFinType2[is.na(all$BsmtFinType2)] <- 'None'  
table(all$BsmtFinType2, useNA = "always")

##   
## ALQ BLQ GLQ LwQ None Rec Unf <NA>   
## 52 68 34 87 79 105 2494 0

**BsmtFullBath(2), BsmtHalfBath(2), BsmtFinSF1(1), BsmtFinSF2(1), BsmtUnfSF(1), TotalBsmtSF(1)**

이제 “BsmtFullBath(2), BsmtHalfBath(2), BsmtFinSF1(1), BsmtFinSF2(1), BsmtUnfSF(1), TotalBsmtSF(1)” 6개 변수에 대해 살펴보면,

g1 <- ggplot(all, aes(x=BsmtFullBath))+  
 geom\_bar(na.rm=T, fill="deepskyblue3")  
g2 <- ggplot(all, aes(x=BsmtHalfBath))+  
 geom\_bar(na.rm=T, fill="deepskyblue3")  
g3 <- ggplot(all, aes(x=BsmtFinSF1))+  
 geom\_histogram(bins = 50, na.rm=T, fill="deepskyblue3")   
g4 <- ggplot(all, aes(x=BsmtFinSF2))+  
 geom\_histogram(bins = 50, na.rm=T, fill="deepskyblue3")   
g5 <- ggplot(all, aes(x=BsmtUnfSF))+  
 geom\_histogram(bins = 50, na.rm=T, fill="deepskyblue3")   
g6 <- ggplot(all, aes(x=TotalBsmtSF))+  
 geom\_histogram(bins = 50, na.rm=T, fill="deepskyblue3")   
  
multiplot(g1,g2,g3,g4,g5,g6, layout = matrix(c(1,2,3,4,5,6),2,3))



all[is.na(all$BsmtFullBath) | is.na(all$BsmtHalfBath) | is.na(all$BsmtFinSF1) | is.na(all$BsmtFinSF2), c("BsmtFullBath", "BsmtHalfBath","BsmtFinSF1","BsmtFinSF2","BsmtUnfSF","TotalBsmtSF", "BsmtQual", "BsmtCond","BsmtExposure")]

## BsmtFullBath BsmtHalfBath BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF  
## 2121 NA NA NA NA NA NA  
## 2189 NA NA 0 0 0 0  
## BsmtQual BsmtCond BsmtExposure  
## 2121 None None None  
## 2189 None None None

2121번째 관찰값은 Bsmt에 대한 모든 정보가 없었다. 따라서 Bsmt가 없는 집으로 보정하였고, 2189번째 관찰값도 마찬가지로 지하실 넓이변수 등이 모두 0이었으므로 지하실이 없는 집으로 보정하였다.

Bsmt <- c("BsmtFullBath", "BsmtHalfBath","BsmtFinSF1","BsmtFinSF2","BsmtUnfSF","TotalBsmtSF")  
  
for(c in Bsmt){  
 which <- is.na(all[,c])  
 all[,c][which] <- 0  
}  
  
# confirmation  
NAs(all[,grep("Bsmt", names(all))])

## named integer(0)

### Mansonry veneer

석조부분 베니어판을 의미한다. 건물 외벽에 석조 베니어판이 있는 경우.

NAs(all[,grep("MasVnr", names(all))])

## MasVnrType MasVnrArea   
## 24 23

“MasVnrType, MasVnrArea” 두 변수의 결측이 동시에 나타나는 관찰값인지 알아보면,

all[is.na(all$MasVnrType),c("MasVnrType", "MasVnrArea")]

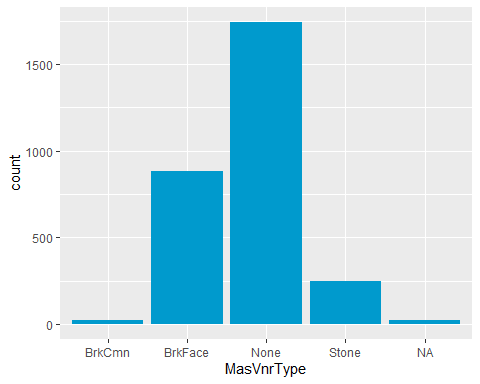
## MasVnrType MasVnrArea  
## 235 <NA> NA  
## 530 <NA> NA  
## 651 <NA> NA  
## 937 <NA> NA  
## 974 <NA> NA  
## 978 <NA> NA  
## 1244 <NA> NA  
## 1279 <NA> NA  
## 1692 <NA> NA  
## 1707 <NA> NA  
## 1883 <NA> NA  
## 1993 <NA> NA  
## 2005 <NA> NA  
## 2042 <NA> NA  
## 2312 <NA> NA  
## 2326 <NA>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

nrow(all[is.na(all$MasVnrType),c("MasVnrType", "MasVnrArea")])

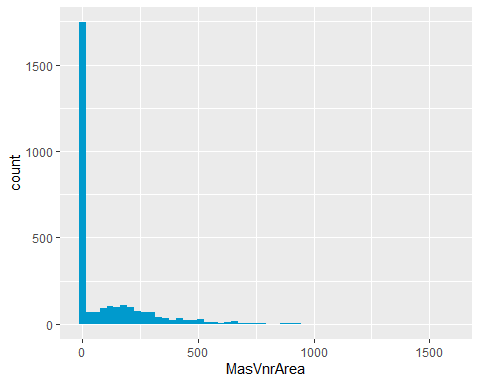
## [1] 24

2611번째 관찰값을 제외하고는 23개 모두 NA값을 가졌다.

ggplot(all, aes(x=MasVnrType)) +  
 geom\_bar(fill="deepskyblue3")



ggplot(all, aes(x=MasVnrArea)) +  
 geom\_histogram(fill="deepskyblue3", na.rm = T, binwidth=30)



table(all$MasVnrType, useNA = "always")

##   
## BrkCmn BrkFace None Stone <NA>   
## 25 879 1742 249 24

summary(all$MasVnrArea)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.0 0.0 0.0 102.2 164.0 1600.0 23

MasVnrType이 없는 None인 빈도가 2919개 중 1742개로 가장 많았고, ManVnrArea또한 0인 빈도가 매우 많았다. 따라서 2611번째를 제외한 23개의 결측에 대해서 None, 0으로 결측치 처리하였고, 2611번째의 ManVnrArea는 198로 평균보다 높은 값을 가졌다. 따라서 MasVnrType 중 None을 제외하고 가장 빈도가 많았던 BrkFace로 결측치 처리하였다.

all$MasVnrType[2611] <- 'BrkFace'  
all$MasVnrType[is.na(all$MasVnrType)] <- 'None'  
all$MasVnrArea[is.na(all$MasVnrArea)] <- 0

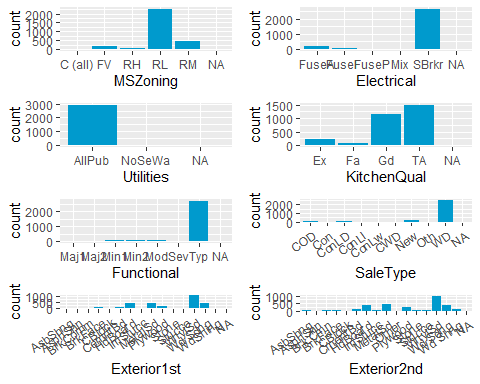
### other variables

NAs(all)

## SalePrice MSZoning Utilities Functional Exterior1st Exterior2nd   
## 1459 4 2 2 1 1   
## Electrical KitchenQual SaleType   
## 1 1 1

각각의 빈도를 살펴보면,

g1 <- ggplot(all, aes(x=MSZoning)) +  
 geom\_bar(fill="deepskyblue3")  
g2 <- ggplot(all, aes(x=Utilities)) +  
 geom\_bar(fill="deepskyblue3")  
g3 <- ggplot(all, aes(x=Functional)) +  
 geom\_bar(fill="deepskyblue3")  
g4 <- ggplot(all, aes(x=Exterior1st)) +  
 geom\_bar(fill="deepskyblue3") + theme(axis.text.x = element\_text(angle = 35, hjust = 1)) +  
 scale\_y\_continuous(breaks=seq(0,1000,by=500))  
g5 <- ggplot(all, aes(x=Exterior2nd)) +  
 geom\_bar(fill="deepskyblue3") + theme(axis.text.x = element\_text(angle = 35, hjust = 1)) +  
 scale\_y\_continuous(breaks=seq(0,1000,by=500))  
g6 <- ggplot(all, aes(x=Electrical)) +  
 geom\_bar(fill="deepskyblue3")  
g7 <- ggplot(all, aes(x=KitchenQual)) +  
 geom\_bar(fill="deepskyblue3")  
g8 <- ggplot(all, aes(x=SaleType)) +  
 geom\_bar(fill="deepskyblue3") + theme(axis.text.x = element\_text(angle = 35, hjust = 1)) +  
 scale\_y\_continuous(breaks=seq(0,2500,by=1000))  
  
multiplot(g1,g2,g3,g4,g6,g7,g8,g5, cols = 2)



table(all$MSZoning)

##   
## C (all) FV RH RL RM   
## 25 139 26 2265 460

table(all$Utilities) # removing

##   
## AllPub NoSeWa   
## 2916 1

table(all$Functional)

##   
## Maj1 Maj2 Min1 Min2 Mod Sev Typ   
## 19 9 65 70 35 2 2717

table(all$Exterior1st)

##   
## AsbShng AsphShn BrkComm BrkFace CBlock CemntBd HdBoard ImStucc MetalSd Plywood   
## 44 2 6 87 2 126 442 1 450 221   
## Stone Stucco VinylSd Wd Sdng WdShing   
## 2 43 1025 411 56

table(all$Exterior2nd)

##   
## AsbShng AsphShn Brk Cmn BrkFace CBlock CmentBd HdBoard ImStucc MetalSd Other   
## 38 4 22 47 3 126 406 15 447 1   
## Plywood Stone Stucco VinylSd Wd Sdng Wd Shng   
## 270 6 47 1014 391 81

table(all$Electrical)

##   
## FuseA FuseF FuseP Mix SBrkr   
## 188 50 8 1 2671

table(all$KitchenQual)

##   
## Ex Fa Gd TA   
## 205 70 1151 1492

table(all$SaleType)

##   
## COD Con ConLD ConLI ConLw CWD New Oth WD   
## 87 5 26 9 8 12 239 7 2525

Utilites의 경우 대부분 ALLPUb (All public utilites - 전기,가스, 수도시설) 이므로 예측변수로 사용하기에 적합하지 않았다. 따라서 제외시켰다.

all$Utilities <- NULL

KitchenQual을 제외한 7개 변수들은 하나의 level에 압도적인 빈도를 가지고 있었다. 따라서 최빈값으로 보정하였고,

all[is.na(all$KitchenQual),c("KitchenQual","KitchenAbvGr")]

## KitchenQual KitchenAbvGr  
## 1556 <NA> 1

names(sort(-table(all[all$KitchenAbvGr==1,"KitchenQual"])))[1]

## [1] "TA"

부엌의 개수도 1개인 데이터 중 KitchenQUal의 최빈값이 TA(Typical/Average)이었으므로 TA로 보정하였다.

all$MSZoning[is.na(all$MSZoning)] <- names(sort(-table(all[,"MSZoning"])))[1]  
all$Electrical[is.na(all$Electrical)] <- names(sort(-table(all[,"Electrical"])))[1]  
all$Functional[is.na(all$Functional)] <- names(sort(-table(all[,"Functional"])))[1]  
all$SaleType[is.na(all$SaleType)] <- names(sort(-table(all[,"SaleType"])))[1]  
all$Exterior1st[is.na(all$Exterior1st)] <- names(sort(-table(all[,"Exterior1st"])))[1]  
all$Exterior2nd[is.na(all$Exterior2nd)] <- names(sort(-table(all[,"Exterior2nd"])))[1]  
  
all[1556,"KitchenQual"] <- "TA"

NAs(all)

## SalePrice   
## 1459

sum(complete.cases(subset(all, select=-c(SalePrice))))

## [1] 2919

which(complete.cases(subset(all,select=-c(SalePrice)))==F) #obs of NAs

## integer(0)

이제 결측치 처리는 완료되었고, data description에 나타난대로 정수로 encoding이 필요한 변수들을 encoding하였다.

# revalue   
Qualities <- c('None'=0, 'Po'=1, "Fa"=2, "TA"=3, "Gd"=4, "Ex"=5)  
Exposure <- c('None'=0, 'No'=1, 'Mn'=2, 'Av'=3, 'Gd'=4)  
FinType <- c('None'=0, 'Unf'=1, 'LwQ'=2, 'Rec'=3, 'BLQ'=4, 'ALQ'=5, 'GLQ'=6)  
  
all$PoolQC <- as.factor(revalue(all$PoolQC, replace=Qualities))

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

all$FireplaceQu <- as.factor(revalue(all$FireplaceQu , replace=Qualities))  
all$LotShape<-as.factor(revalue(all$LotShape, c('IR3'=0, 'IR2'=1, 'IR1'=2, 'Reg'=3)))  
  
#Garage  
all$GarageFinish<-as.factor(revalue(all$GarageFinish, c('None'=0, 'Unf'=1, 'RFn'=2, 'Fin'=3)))  
all$GarageQual<-as.factor(revalue(all$GarageQual, Qualities))  
all$GarageCond<-as.factor(revalue(all$GarageCond, Qualities))  
  
#Basement  
all$BsmtQual<-as.factor(revalue(all$BsmtQual, Qualities))

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

all$BsmtCond<-as.factor(revalue(all$BsmtCond, Qualities))

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

all$BsmtExposure<-as.factor(revalue(all$BsmtExposure, Exposure))  
all$BsmtFinType1<-as.factor(revalue(all$BsmtFinType1, FinType))  
all$BsmtFinType2<-as.factor(revalue(all$BsmtFinType2, FinType))  
  
#others  
all$KitchenQual<-as.factor(revalue(all$KitchenQual, Qualities))

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

all$ExterQual<-as.factor(revalue(all$ExterQual, Qualities))

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

all$ExterCond<-as.factor(revalue(all$ExterCond, Qualities))

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

all$HeatingQC<-as.factor(revalue(all$HeatingQC, Qualities))

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

all$Fence<-as.factor(revalue(all$Fence, c("None"=0, "MnWw"=1, "GdWo"=2, "MnPrv"=3, "GdPrv"=4)))  
all$PavedDrive<-as.factor(revalue(all$PavedDrive, c("N"=0, "P"=1, "Y"=2)))  
all$LandContour <- as.factor(revalue(all$LandContour, c("Low"=0, "HLS"=1, "Bnk"=2, "Lvl"=3)))  
all$LandSlope <- as.factor(revalue(all$LandSlope, c("Sev"=0, "Mod"=1, "Gtl"=2)))

## variable engineering

### Remodel Y/N

집들 중에는 리모델링을 한 집이 있고 안한집이 있지만, 이를 나타내는 변수는 없고, 리모델링을 했다면 했던 연도만 표시하는 변수만 존재했다. 따라서 집 리모델링 유무를 나타내는 변수를 추가해보았다.

all$Remodel <- ifelse(all$YearBuilt==all$YearRemodAdd, 0,1)  
table(all$Remodel, useNA = "always") #all

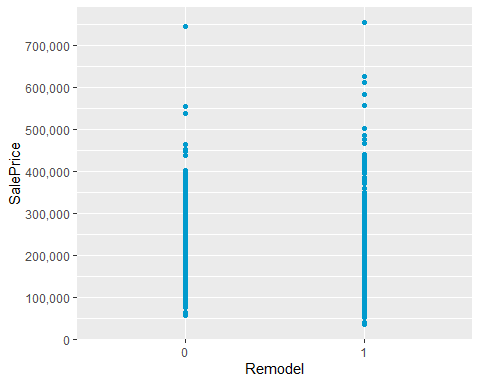
##   
## 0 1 <NA>   
## 1560 1359 0

table(all[!is.na(all$SalePrice),]$Remodel, useNA = "always") #train

##   
## 0 1 <NA>   
## 764 696 0

all$YearRemodAdd <- NULL  
all$Remodel <- as.factor(all$Remodel)

ggplot(data=all[!is.na(all$SalePrice),], aes(x=Remodel, y=SalePrice)) +  
 geom\_point(col="deepskyblue3") + scale\_y\_continuous(breaks=seq(0, 900000, by=100000), labels=comma)



mean(all$SalePrice[!all$Remodel=="0"], na.rm=T)

## [1] 179096.3

mean(all$SalePrice[!all$Remodel=="1"], na.rm=T)

## [1] 182583.7

하지만 리모델링 유무에 따라 SalePrice 평균의 차이가 거의 없이 나타났다. 따라서 바로 제거하였다.

all$Remodel <- NULL

### New house Y/N

all$Newhouse <- ifelse(all$YrSold==all$YearBuilt, 1,0)  
table(all$Newhouse, useNA = "always") #all

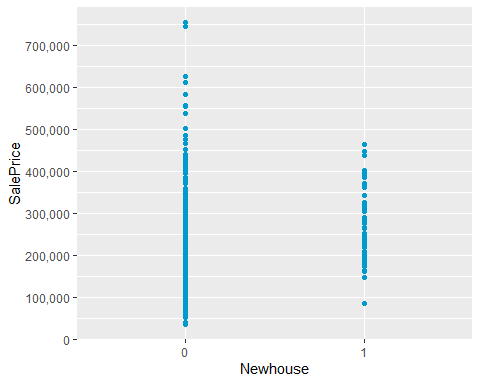
##   
## 0 1 <NA>   
## 2803 116 0

table(all[!is.na(all$SalePrice),]$Newhouse, useNA = "always") #train

##   
## 0 1 <NA>   
## 1396 64 0

all$Newhouse <- as.factor(all$Newhouse)

ggplot(data=all[!is.na(all$SalePrice),], aes(x=Newhouse, y=SalePrice)) +  
 geom\_point(col="deepskyblue3") + scale\_y\_continuous(breaks=seq(0, 900000, by=100000), labels=comma)



mean(all$SalePrice[!all$Newhouse=="0"], na.rm=T)

## [1] 264302.2

mean(all$SalePrice[!all$Newhouse=="1"], na.rm=T)

## [1] 177098.6

지어진 집 바로 팔린 경우, SalePrice의 평균은 그렇지 않은 집보다 오히려 떨어졌다. 하지만 분산은 확실히 작음을 확인할 수 있었다. 이는 all$YrSold 즉, 2006년부터 2010년간 팔린 집들을 대상으로 하다보니, 2006년부터 2010년간 지어진 집은 확실히 전체 기간보다 비슷한 집들이 있을 가능성이 높기 때문이라 판단된다. 이 변수를 모형에 포함시키더라도 각 범주간 SalePrice에 차이를 내기 어렵다고 판단하여 제거하였다.

all$Newhouse <- NULL

## variable explanation

아래 5개 변수는 범주형 변수로, factor로 변형해준다.

all$MSSubClass = as.factor(all$MSSubClass)  
all$OverallQual = as.factor(all$OverallQual)  
all$OverallCond = as.factor(all$OverallCond)  
all$YrSold = as.factor(all$YrSold)  
all$MoSold = as.factor(all$MoSold)

all에 feature class별로 분류하면,

chrvar <- names(all[,!(sapply(all,is.numeric))])  
numvar <- names(all[,sapply(all,is.numeric)])  
sum(length(chrvar),length(numvar))

## [1] 78

chrvar

## [1] "MSSubClass" "MSZoning" "Street" "Alley"   
## [5] "LotShape" "LandContour" "LotConfig" "LandSlope"   
## [9] "Neighborhood" "Condition1" "Condition2" "BldgType"   
## [13] "HouseStyle" "OverallQual" "OverallCond" "RoofStyle"   
## [17] "RoofMatl" "Exterior1st" "Exterior2nd" "MasVnrType"   
## [21] "ExterQual" "ExterCond" "Foundation" "BsmtQual"   
## [25] "BsmtCond" "BsmtExposure" "BsmtFinType1" "BsmtFinType2"   
## [29] "Heating" "HeatingQC" "CentralAir" "Electrical"   
## [33] "KitchenQual" "Functional" "FireplaceQu" "GarageType"   
## [37] "GarageFinish" "GarageQual" "GarageCond" "PavedDrive"   
## [41] "PoolQC" "Fence" "MiscFeature" "MoSold"   
## [45] "YrSold" "SaleType" "SaleCondition"

length(chrvar)

## [1] 47

numvar

## [1] "LotFrontage" "LotArea" "YearBuilt" "MasVnrArea"   
## [5] "BsmtFinSF1" "BsmtFinSF2" "BsmtUnfSF" "TotalBsmtSF"   
## [9] "X1stFlrSF" "X2ndFlrSF" "LowQualFinSF" "GrLivArea"   
## [13] "BsmtFullBath" "BsmtHalfBath" "FullBath" "HalfBath"   
## [17] "BedroomAbvGr" "KitchenAbvGr" "TotRmsAbvGrd" "Fireplaces"   
## [21] "GarageYrBlt" "GarageCars" "GarageArea" "WoodDeckSF"   
## [25] "OpenPorchSF" "EnclosedPorch" "X3SsnPorch" "ScreenPorch"   
## [29] "PoolArea" "MiscVal" "SalePrice"

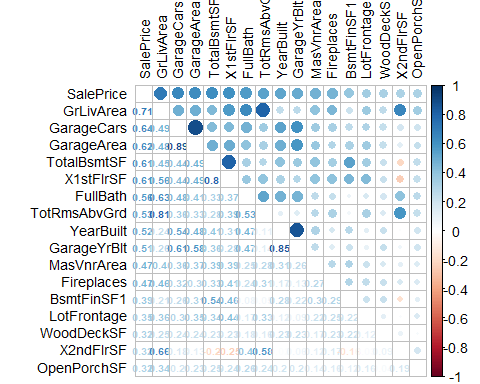
length(numvar)

## [1] 31

### correlation among numeric variables

수치형 변수가 SalePrice와 어느정도 상관관계가 있는지 알아보기 위해 correlation 절대값이 0.3이상인 corrplot을 그려보면,

cor <- cor(all[,numvar], use = "pairwise.complete.obs")  
cor\_sort <- as.matrix(sort(cor[,'SalePrice'], decreasing = TRUE))  
corhigh <- names(which(apply(cor\_sort,1,function(x) abs(x)>0.3)))  
  
cor <- cor[corhigh,corhigh]  
  
corrplot.mixed(cor,tl.col="black", tl.pos = "lt", tl.cex=0.9, cl.cex=0.9, number.cex=.6)

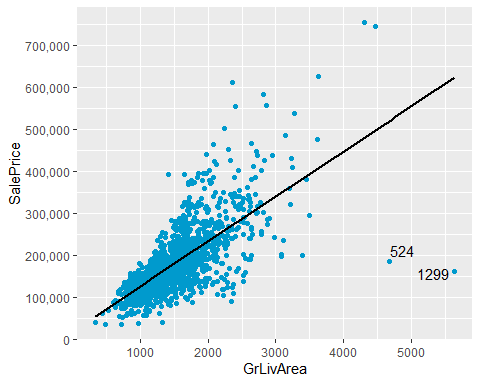


SalePrice에 관련해서 가장 correlation이 높은 변수는 GrLivArea, GarageCars, GarageArea, TotalBsmtSF, X1stFlrSF와 등 순이었다. 그 중 GarageCars와 GarageArea는 0.89로 모든 변수들 중 가장 높은 corr을 가졌고, GaregeYrBlt와 YearBulit도 0.85, GrLivArea와 TotRmsAbvGrd도 0.81, X1stFlrSF와 TotalBsmtSF는 0.8로 corr이 높았다.

GrLivArea : Above ground living area square feet  
GarageCars : Size of garage in car capacity  
GarageArea : Size of garege in square feet  
TotalBsmtSF : Total square feet of basement area  
X1stFlrSF와 : First Floor square feet  
  
GarageYrBlt : Year garage was built  
YearBuilt : Original construction date  
TotRmsAbvGrd : Total rooms above grade (not include bathroom)

**GrLivArea**

ggplot(data=all[!is.na(all$SalePrice),], aes(x=GrLivArea, y=SalePrice))+  
 geom\_point(col='deepskyblue3') + geom\_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +  
 scale\_y\_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +  
 geom\_text\_repel(aes(label = ifelse(all$GrLivArea[!is.na(all$SalePrice)]>4500, rownames(all), '')))



524와 1299번째 관찰값은 선에서 많이 벗어난 것 처럼 보인다. 즉, GrLivArea가 넓음에도 불구하고, SalePrice가 낮은 경우이다. 이 집들의 OverallQual을 본다면,

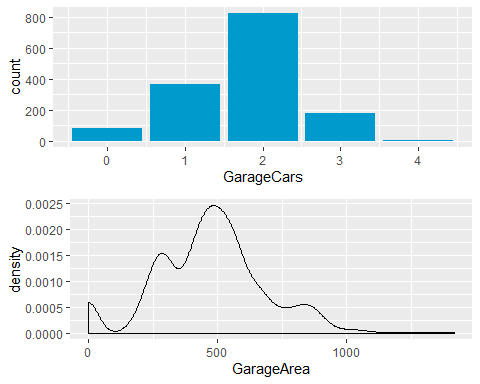
all[c(524, 1299),c("SalePrice", "GrLivArea", "OverallQual")] #train data

## SalePrice GrLivArea OverallQual  
## 524 184750 4676 10  
## 1299 160000 5642 10

전체적인 퀄리티도 10으로 가장 높은 점수를 형성하지만, SalePrice는 낮다.

**GarageCars, GarageArea**

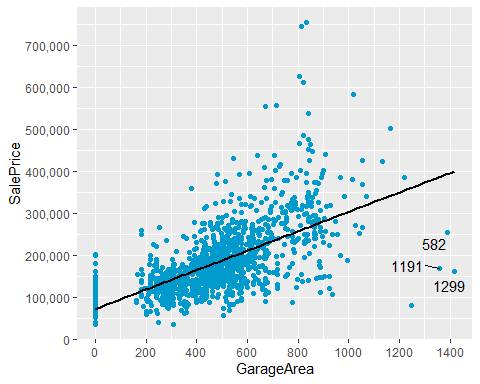
g1 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=GarageCars))+  
 geom\_bar(fill='deepskyblue3', stat="count")   
g2 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=GarageArea))+  
 geom\_density()   
grid.arrange(g1, g2)



cor이 0.89이고 두 그래프 모양이 비슷한 모양을 가졌다. 직관적으로, 차고의 크기가 크면 차고에 수용 가능한 차 개수도 많아질 것이라고 생각하였고, 모델 fitting 과정에서 multicollinearity 발생 우려가 있기 때문에 GarageCars 변수는 분석에서 제외시켰다.

all$GarageCars <- NULL

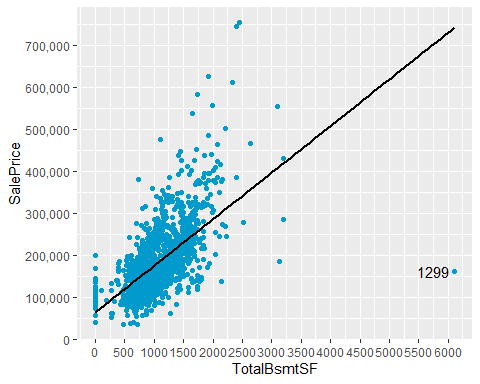
ggplot(data=all[!is.na(all$SalePrice),], aes(x=GarageArea, y=SalePrice))+  
 geom\_point(col="deepskyblue3") + geom\_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +  
 scale\_y\_continuous(breaks=seq(0,800000, by=100000), label=comma) +  
 scale\_x\_continuous(breaks=seq(0,1500, by=200)) +  
 geom\_text\_repel(aes(label = ifelse(all$GarageArea[!is.na(all$SalePrice)]>1300, rownames(all), '')))



GarageArea와 SalePrice간 cor은 0.62였고, 그 중 GarageArea가 높은 값 (GarageArea가>1,300)을 표시하였다.

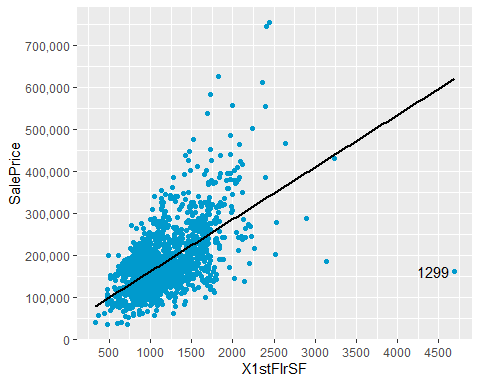
**TotalBsmtSF**

ggplot(data=all[!is.na(all$SalePrice),], aes(x=TotalBsmtSF, y=SalePrice))+  
 geom\_point(col="deepskyblue3") + geom\_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +  
 scale\_y\_continuous(breaks=seq(0,800000, by=100000), label=comma) +  
 scale\_x\_continuous(breaks=seq(0,6000, by=500)) +  
 geom\_text\_repel(aes(label = ifelse(all$TotalBsmtSF[!is.na(all$SalePrice)]>6000 , rownames(all), '')))



**X1stFlrSF**

ggplot(data=all[!is.na(all$SalePrice),], aes(x=X1stFlrSF, y=SalePrice))+  
 geom\_point(col="deepskyblue3") + geom\_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +  
 scale\_y\_continuous(breaks=seq(0,800000, by=100000), label=comma) +  
 scale\_x\_continuous(breaks=seq(0,5000, by=500)) +  
 geom\_text\_repel(aes(label = ifelse(all$X1stFlrSF[!is.na(all$SalePrice)]>4000 , rownames(all), '')))

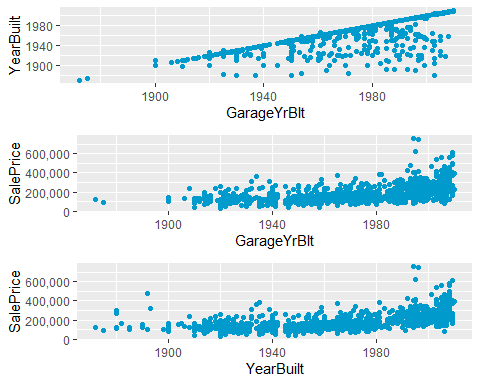


SalePrice와 가장 관련있는 수치형 변수 5가지를 살펴본 결과, 1299번째 값은 모두 중심에서 떨어져 높은 값으로 나타났지만 SalePrice는 상대적으로 낮은 값을 가지는 것을 알 수 있었다. 이를 outlier라고 여겨 데이터에서 제외하였다.

all <- all[-c(1299),]

**GarageYrBlt YearBuilt**: 0.85

g1 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=GarageYrBlt, y=YearBuilt))+  
 geom\_point(col="deepskyblue3")  
g2 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=GarageYrBlt, y=SalePrice))+  
 geom\_point(col="deepskyblue3") + scale\_y\_continuous(label=comma)  
g3 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=YearBuilt, y=SalePrice))+  
 geom\_point(col="deepskyblue3") + scale\_y\_continuous(label=comma)  
  
grid.arrange(g1,g2,g3)

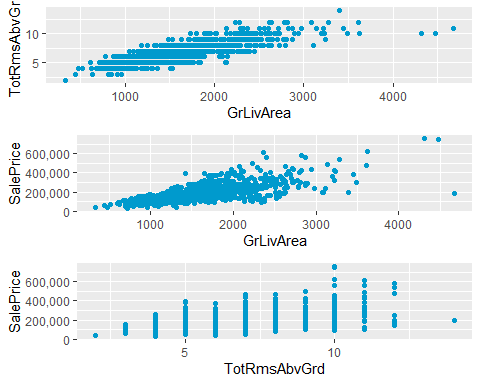


집이 지어질 때 차고가 같이 지어진 경우 두 변수의 값이 같았다. 그래서 SalePrice 대 GarageYrBlt, SalePrice 대 YearBuilt의 모양이 매우 비슷한 것을 알 수 있었다. 또한 차고가 없는 경우 (No Garage)에도 GarageYrBlt 값은 존재하는 경우가 있었다. 따라서 GarageYrBlt 변수를 분석에서 제외시켰다.

all$GarageYrBlt <- NULL

**GrLivArea TotRmsAbvGrd**: 0.81

g1 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=GrLivArea, y=TotRmsAbvGrd))+  
 geom\_point(col="deepskyblue3")   
  
g2 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=GrLivArea, y=SalePrice))+  
 geom\_point(col="deepskyblue3") + scale\_y\_continuous(label=comma)   
  
g3 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=TotRmsAbvGrd, y=SalePrice))+  
 geom\_point(col="deepskyblue3") + scale\_y\_continuous(label=comma)  
  
grid.arrange(g1,g2,g3)

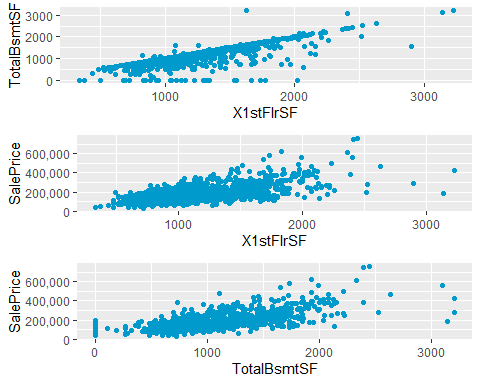


집의 크기(GrLivArea)와 방 개수(TotRmsAbvGrd)에 대한 산점도에서 상관관계가 매우 높은것으로 나타났고, 이에 SalePrice를 예측할 때 다중 공선성이 발생할 수 있을것 같아 SalePrice와의 상관관계가 더 낮은 TotRmsAbvGrd 변수를 제외시켰다.

all$TotRmsAbvGrd <- NULL

**X1stFlrSF TotalBsmtSF**: 0.8

g1 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=X1stFlrSF, y=TotalBsmtSF))+  
 geom\_point(col="deepskyblue3")   
  
g2 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=X1stFlrSF, y=SalePrice))+  
 geom\_point(col="deepskyblue3") + scale\_y\_continuous(label=comma)   
  
g3 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=TotalBsmtSF, y=SalePrice))+  
 geom\_point(col="deepskyblue3") + scale\_y\_continuous(label=comma)  
  
grid.arrange(g1,g2,g3)



X1stFlrSF와 TotalBsmtSF 또한 cor이 0.8로 매우 관련 있었으므로 SalePrice 와의 correlation이 그나마 낮은 TotalBsmtSF를 분석에서 제외시켰다.

all$TotalBsmtSF <- NULL

### character variables

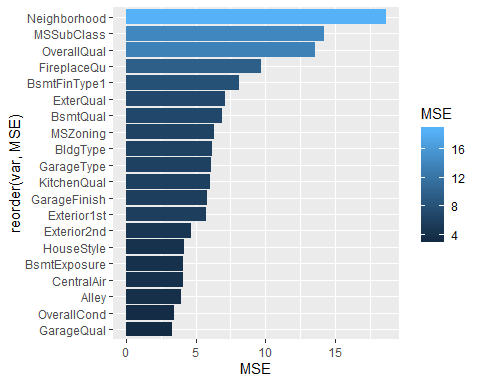
chrvar

## [1] "MSSubClass" "MSZoning" "Street" "Alley"   
## [5] "LotShape" "LandContour" "LotConfig" "LandSlope"   
## [9] "Neighborhood" "Condition1" "Condition2" "BldgType"   
## [13] "HouseStyle" "OverallQual" "OverallCond" "RoofStyle"   
## [17] "RoofMatl" "Exterior1st" "Exterior2nd" "MasVnrType"   
## [21] "ExterQual" "ExterCond" "Foundation" "BsmtQual"   
## [25] "BsmtCond" "BsmtExposure" "BsmtFinType1" "BsmtFinType2"   
## [29] "Heating" "HeatingQC" "CentralAir" "Electrical"   
## [33] "KitchenQual" "Functional" "FireplaceQu" "GarageType"   
## [37] "GarageFinish" "GarageQual" "GarageCond" "PavedDrive"   
## [41] "PoolQC" "Fence" "MiscFeature" "MoSold"   
## [45] "YrSold" "SaleType" "SaleCondition"

chrvar에 있는 변수들과 SalePrice간 관계가 어떤 변수가 가장 높은지 알아보기 위해 rough한 randomforest를 시행하였다. 변수 중요도를 토대로 가장 관계가 있는 변수들이 무엇인지 추려보면,

#roughrf <- train(SalePrice~., data=all[!is.na(all$SalePrice),c(chrvar,"SalePrice")], tuneGrid=grid, method="rf")  
#varImp(roughrf)  
#plot(varImp(roughrf))  
set.seed(555)  
all\_fac=all[!is.na(all$SalePrice),chrvar] %>% mutate\_if(is.character, as.factor)  
roughrf <- randomForest(x=all\_fac, y=all$SalePrice[!is.na(all$SalePrice)], ntree=100, importance=T)  
imp <-importance(roughrf)  
impdf <- data.frame(var=row.names(imp), MSE=imp[,1])  
impdf <- impdf[order(impdf$MSE, decreasing = T),]

ggplot(impdf[1:20,], aes(x=reorder(var, MSE), y=MSE, fill=MSE)) + geom\_bar(stat="identity") +  
 coord\_flip()



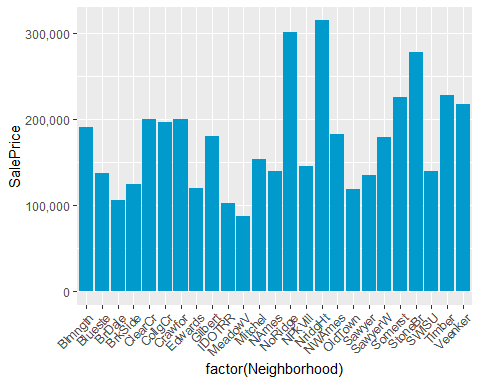
MSEs는 %IncMSE로 전체 MSE0에서 해당 변수가 포함된 트리의 MSE1를 계산해 (MSE1-MSE0)/MSE0\*100을 계산. %IncMSE가 높을수록 col1이 미치는 영향이 크다 > 중요한 변수이다. 나타냄. 변수 중요도를 보면 Neighbrohood, MSSubClass, OverallQual 등 순으로 SalePrice에 영향이 많은 것으로 나타났다.

MSSubClass: Identifies the type of dwelling involved in the sale  
OverallQual: Overall QUality  
Neighborhood: Physical locations within Ames city limits

**Neighborhood**

ggplot(data=all[!is.na(all$SalePrice),], aes(x=factor(Neighborhood), y=SalePrice))+  
 geom\_histogram(fill='deepskyblue3', stat='summary', fun.y='median') +  
 scale\_y\_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +  
 theme(axis.text.x=element\_text(angle=45, hjust=1))

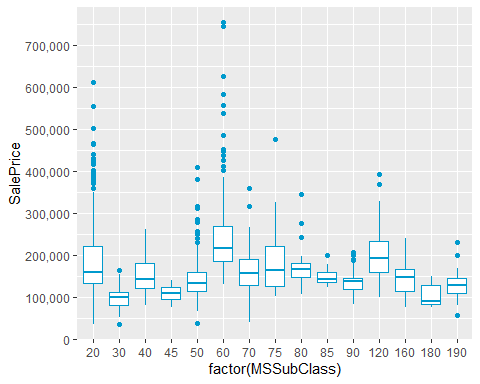
## Warning: Ignoring unknown parameters: binwidth, bins, pad



Neighborhood에 따라 SalePrice 값이 많이 다른 것을 알 수 있었다.

**MSSubClass**

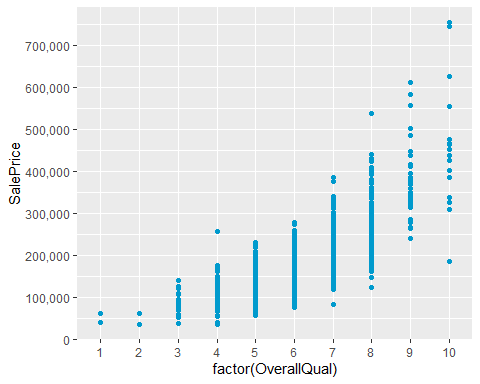
ggplot(data=all[!is.na(all$SalePrice),], aes(x=factor(MSSubClass), y=SalePrice))+  
 geom\_boxplot(col='deepskyblue3') +  
 scale\_y\_continuous(breaks= seq(0, 800000, by=100000), labels = comma)



MSSubClass는 판매와 관련한 주택유형 식별 코드로, 60이 가장 높았다. 60은 2-story 1946 & newer을 나타낸다. 전반적으로 1946년 이상되거나, story가 높을수록 SalePrice 가격이 높았다.

**OverallQual**

ggplot(data=all[!is.na(all$SalePrice),], aes(x=factor(OverallQual), y=SalePrice))+  
 geom\_point(col='deepskyblue3') +  
 scale\_y\_continuous(breaks= seq(0, 800000, by=100000), labels = comma)



Overall Quality가 증가할수록 SalePrice가 점점 증가하는 꼴을 볼 수 있다.

최종적인 데이터는 이상치가 1개 제외되고, 변수가 4개 제외되었다.

dim(all)

## [1] 2918 74

# preparing data step

## normalizing

numvar <- names(all[,sapply(all,is.numeric)])  
Normalizing <- preProcess(all[!is.na(all$SalePrice),numvar[c(which(numvar!="SalePrice"))]], method=c("center", "scale")) # calculating with train data wihtout "SalePrice" variable  
print(Normalizing)

## Created from 1459 samples and 26 variables  
##   
## Pre-processing:  
## - centered (26)  
## - ignored (0)  
## - scaled (26)

normnumerics <- predict(Normalizing, all[, numvar])

## one-hot encodeing

#dummies <- dummyVars(~., all[chrvar])  
#categorical\_1\_hot <- predict(dummies, all[chrvar])  
dummies <- as.data.frame(model.matrix(~.-1, all[,names(all) %in% chrvar]))

데이터에서 관찰값이 없는 level들이거나 빈도가 10 이하인 변수로서 만들어진 predictor들을 찾아 이를 없앤다 (즉, variance가 0인 변수들: 예를 들어 train에서 MSSubClass가 150인 데이터는 train에서 없다. 따라서 train1$MSSubClass150은 모두 0이다.)

#train  
rm1 <- which(colSums(dummies[!is.na(all$SalePrice),])<10)  
colnames(dummies[rm1])

## [1] "MSSubClass40" "MSSubClass150" "LotConfigFR3"   
## [4] "NeighborhoodBlueste" "NeighborhoodNPkVill" "Condition1PosA"   
## [7] "Condition1RRNe" "Condition1RRNn" "Condition2Feedr"   
## [10] "Condition2PosA" "Condition2PosN" "Condition2RRAe"   
## [13] "Condition2RRAn" "Condition2RRNn" "HouseStyle2.5Fin"   
## [16] "OverallQual2" "OverallCond2" "RoofStyleMansard"   
## [19] "RoofStyleShed" "RoofMatlMembran" "RoofMatlMetal"   
## [22] "RoofMatlRoll" "RoofMatlWdShake" "RoofMatlWdShngl"   
## [25] "Exterior1stAsphShn" "Exterior1stBrkComm" "Exterior1stCBlock"   
## [28] "Exterior1stImStucc" "Exterior1stStone" "Exterior2ndAsphShn"   
## [31] "Exterior2ndBrk Cmn" "Exterior2ndCBlock" "Exterior2ndOther"   
## [34] "Exterior2ndStone" "ExterCond5" "FoundationStone"   
## [37] "FoundationWood" "BsmtCond1" "HeatingGrav"   
## [40] "HeatingOthW" "HeatingWall" "ElectricalFuseP"   
## [43] "ElectricalMix" "FunctionalMaj2" "FunctionalSev"   
## [46] "GarageTypeCarPort" "GarageQual1" "GarageQual5"   
## [49] "GarageCond1" "GarageCond4" "GarageCond5"   
## [52] "PoolQC2" "PoolQC4" "PoolQC5"   
## [55] "MiscFeatureOthr" "MiscFeatureTenC" "SaleTypeCon"   
## [58] "SaleTypeConLD" "SaleTypeConLI" "SaleTypeConLw"   
## [61] "SaleTypeCWD" "SaleTypeOth" "SaleConditionAdjLand"

#test  
rm2 <- which(colSums(dummies[is.na(all$SalePrice),])<10)  
colnames(dummies[rm2])

## [1] "MSSubClass40" "MSSubClass45" "MSSubClass75"   
## [4] "MSSubClass150" "MSSubClass180" "NeighborhoodBlueste"   
## [7] "Condition1RRNe" "Condition1RRNn" "Condition2Feedr"   
## [10] "Condition2PosA" "Condition2PosN" "Condition2RRAe"   
## [13] "Condition2RRAn" "Condition2RRNn" "HouseStyle1.5Unf"   
## [16] "HouseStyle2.5Fin" "OverallCond2" "RoofStyleMansard"   
## [19] "RoofStyleShed" "RoofMatlMembran" "RoofMatlMetal"   
## [22] "RoofMatlRoll" "RoofMatlWdShake" "RoofMatlWdShngl"   
## [25] "Exterior1stAsphShn" "Exterior1stBrkComm" "Exterior1stCBlock"   
## [28] "Exterior1stImStucc" "Exterior1stStone" "Exterior2ndAsphShn"   
## [31] "Exterior2ndCBlock" "Exterior2ndImStucc" "Exterior2ndOther"   
## [34] "Exterior2ndStone" "ExterCond5" "FoundationStone"   
## [37] "FoundationWood" "BsmtCond1" "HeatingGasW"   
## [40] "HeatingGrav" "HeatingOthW" "HeatingWall"   
## [43] "ElectricalFuseP" "ElectricalMix" "FunctionalMaj2"   
## [46] "FunctionalSev" "GarageTypeCarPort" "GarageQual1"   
## [49] "GarageQual5" "GarageCond1" "GarageCond4"   
## [52] "GarageCond5" "PoolQC2" "PoolQC4"   
## [55] "PoolQC5" "Fence1" "MiscFeatureOthr"   
## [58] "MiscFeatureTenC" "SaleTypeCon" "SaleTypeConLI"   
## [61] "SaleTypeConLw" "SaleTypeCWD" "SaleTypeOth"   
## [64] "SaleConditionAdjLand"

dummies <- dummies[,-rm1]  
dim(dummies)

## [1] 2918 206

dummies <- dummies[,-rm2]  
dim(dummies)

## [1] 2918 160

## combining all variables

all2 <- cbind(normnumerics,all[,!(names(all) %in% names(normnumerics))]) #for caret::train - gbm   
dim(all2) # no one-hot encoding

## [1] 2918 74

all <- cbind(normnumerics, dummies)  
dim(all)

## [1] 2918 187

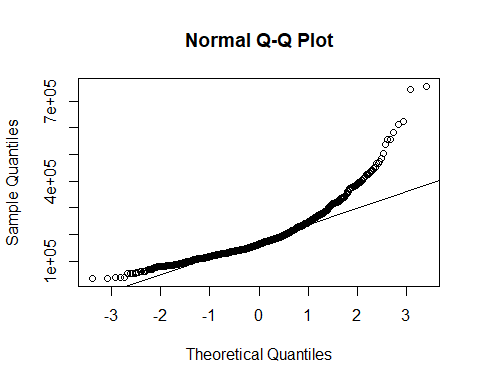
## logarithm target variable

SalePrice는 right-skewed된 분포를 띄고 있었으므로 log변환을 통해 normal하게 만들어주었다. qqplot을 보면 전보다 더 normal하게 바뀌었음을 알 수 있었고 skewness도 0.12로 낮게 나타났다.

skewness(all[!is.na(all$SalePrice),]$SalePrice)

## [1] 1.877969

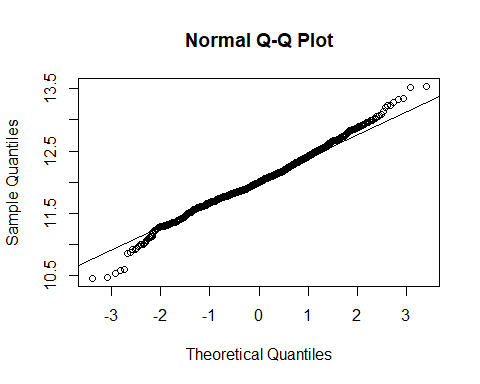
qqnorm(all$SalePrice)  
qqline(all$SalePrice)



#logarithm  
all$SalePrice <- log(all$SalePrice)  
all2$SalePrice <- log(all2$SalePrice)  
  
skewness(all[!is.na(all$SalePrice),]$SalePrice)

## [1] 0.1208349

qqnorm(all$SalePrice)  
qqline(all$SalePrice)



# modeling step

set.seed(555)  
dim(all)

## [1] 2918 187

train1 <- all[!is.na(all$SalePrice),]  
test1 <- all[is.na(all$SalePrice),]  
  
#idx <- createDataPartition(y=train1$SalePrice, p=0.7, list=F)  
#train0 <- train1[idx,]  
#validation0 <- train1[-idx,] #hold-out  
  
dim(train1)

## [1] 1459 187

dim(test1)

## [1] 1459 187

#dim(train0)  
#dim(validation0)

## predicting house price in test set (final model)

### lasso

control <- trainControl(method="cv", number=10)  
lassogrid <- expand.grid(  
 alpha=1,   
 lambda=seq(0.001, 0.1, by=0.0005)  
) # alpha=1; lasso (alpha=0; ridge) / lambda; training rate

# caret::train()  
lasso\_model1 <- train(SalePrice~., data=train1, method = "glmnet", trControl=control, tuneGrid=lassogrid)  
lasso\_model1$bestTune

## alpha lambda  
## 3 1 0.002

preds\_lasso1 <- predict(lasso\_model1, test1)  
preds\_lasso1 <- exp(preds\_lasso1)  
head(preds\_lasso1)

## 1461 1462 1463 1464 1465 1466   
## 127040.4 156405.2 188986.5 203008.6 199912.0 169431.6

# glmnet::cv.glmnet()  
# data.matrix : dataframe to matrix // as.matrix : data.table to matrix  
#lasso\_model <- cv.glmnet(data.matrix(train1[,-which(names(all)=="SalePrice")]), train1[,which(names(all)=="SalePrice")])  
  
#preds\_lasso <- predict(lasso\_model, newx = data.matrix(test1[,-which(names(all)=="SalePrice")]), s="lambda.min")  
#head(exp(preds\_lasso))

Implasso <- varImp(lasso\_model1)  
selected <- length(which(Implasso$importance$Overall!=0))  
notselected <- length(names(train1[,Implasso$importance$Overall==0]))  
  
cat('There are', selected, 'variables in lasso model, and did not selected', notselected, 'variables.')

## There are 123 variables in lasso model, and did not selected 63 variables.

### randomforest

rfgrid <- expand.grid(  
 mtry=c(ncol(train1)/3,sqrt(ncol(train1)))  
)

rf\_model1 <- train(SalePrice~., data=train1, method="rf", trcontrol=control, tuneGrid=rfgrid)  
rf\_model1$bestTune

## mtry  
## 2 62.33333

preds\_rf1 <- predict(rf\_model1, test1)  
preds\_rf1 <- exp(preds\_rf1)  
head(preds\_rf1)

## 1461 1462 1463 1464 1465 1466   
## 126205.6 152472.7 184793.7 187449.4 189149.9 184190.2

varImp(rf\_model1)

## rf variable importance  
##   
## only 20 most important variables shown (out of 186)  
##   
## Overall  
## GrLivArea 100.000  
## YearBuilt 67.000  
## GarageArea 41.418  
## X1stFlrSF 34.615  
## FullBath 28.793  
## KitchenQual3 18.996  
## Fireplaces 12.605  
## BsmtFinSF1 12.079  
## LotArea 10.941  
## BsmtQual5 9.987  
## FoundationPConc 8.921  
## CentralAirY 8.263  
## X2ndFlrSF 8.085  
## LotFrontage 7.111  
## KitchenQual5 4.796  
## GarageTypeAttchd 3.737  
## BsmtUnfSF 3.457  
## MasVnrArea 3.085  
## OpenPorchSF 3.079  
## OverallQual8 2.980

### gradient boost

#train2 <- all2[!is.na(all2$SalePrice),] #NOT ONT-HOT ENCODING  
#test2 <- all2[is.na(all2$SalePrice),]  
#dim(train2)  
#dim(test2)

gbgrid <- expand.grid(shrinkage=c(0.01,0.05, 0.1),  
 interaction.depth=c(2,3),  
 n.minobsinnode=c(5,10),  
 optimaltrees=0,  
 minRMSE=0  
 )

# gb\_model1 <- train(SalePrice~., data=train2, distribution="gaussian", method="gbm", trContol=control, tuneGrid=gbgrid, metric="RMSE", allowParallel=TRUE )

# to find best tune parameters  
system.time(for(i in 1:nrow(gbgrid)) {  
   
 # reproducibility  
 set.seed(555)  
   
 # train model  
 gb\_model <- gbm(SalePrice~., data=train1, n.trees=5000,  
 shrinkage=gbgrid$shrinkage[i],  
 n.minobsinnode=gbgrid$n.minobsinnode[i],  
 interaction.depth=gbgrid$interaction.depth[i], distribution = "gaussian", cv.folds = 5)  
   
 # add min training error and trees to grid  
 gbgrid$optimaltrees[i] <- which.min(gb\_model$cv.error)  
 gbgrid$minRMSE[i] <- sqrt(min(gb\_model$cv.error))  
})

## user system elapsed   
## 331.91 2.36 1091.20

gbgrid %>%  
 arrange(minRMSE) %>%  
 head(10)

## shrinkage interaction.depth n.minobsinnode optimaltrees minRMSE  
## 1 0.05 3 5 2746 0.1303771  
## 2 0.01 3 5 4964 0.1310322  
## 3 0.01 2 5 4967 0.1323442  
## 4 0.05 3 10 526 0.1325364  
## 5 0.01 3 10 2913 0.1326424  
## 6 0.05 2 5 2722 0.1334065  
## 7 0.10 3 10 300 0.1336886  
## 8 0.01 2 10 3223 0.1337009  
## 9 0.10 3 5 1244 0.1340612  
## 10 0.10 2 5 1445 0.1340766

shrinkage는 0.05, n.minobsinnode는 5, interaction.depth는 3, n.trees는 2746일 때 가장 최소 MSE를 가졌다. 따라서 모형을 다시 적합시키면

gb\_model1 <- gbm(SalePrice~., data=train1, n.trees=2746,   
 shrinkage=0.05,  
 n.minobsinnode=5,  
 interaction.depth=3, distribution = "gaussian", cv.folds = 5)  
  
preds\_gb1 <- predict(gb\_model1, test1)

## Using 2484 trees...

preds\_gb1 <- exp(preds\_gb1)  
head(preds\_gb1)

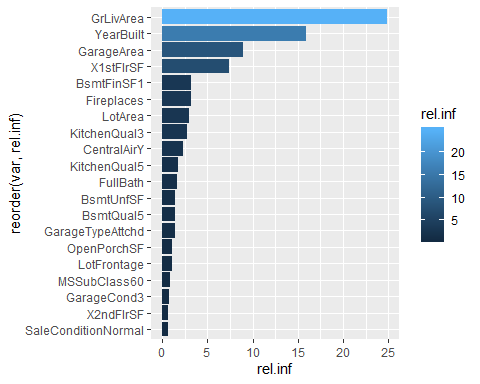
## [1] 127329.2 159898.0 194095.4 199350.1 187201.8 175079.4

가장 성능이 좋았던 gbm의 변수 중요도를 살펴보면

aa<-as.data.frame(summary(gb\_model1, plotit=F))  
head(aa)

## var rel.inf  
## GrLivArea GrLivArea 24.898643  
## YearBuilt YearBuilt 15.858259  
## GarageArea GarageArea 8.980137  
## X1stFlrSF X1stFlrSF 7.344997  
## BsmtFinSF1 BsmtFinSF1 3.186723  
## Fireplaces Fireplaces 3.164498

ggplot(aa[1:20,], aes(x=reorder(var, rel.inf), y=rel.inf, fill=rel.inf)) + geom\_bar(stat="identity") +  
 coord\_flip()



# final submission step

lasso, randomforest, gbm 세 모델의 예측값의 평균을 취해 최종 예측값으로 계산하였다.

preds\_final <- (preds\_lasso1+ preds\_rf1+ preds\_gb1)/3  
sub1 <- data.frame(Id = testID, SalePrice = preds\_final)  
colnames(sub1) <- c("Id","SalePrice")  
head(sub1)

## Id SalePrice  
## 1461 1461 126858.4  
## 1462 1462 156258.6  
## 1463 1463 189291.9  
## 1464 1464 196602.7  
## 1465 1465 192087.9  
## 1466 1466 176233.7

sub2 <- data.frame(Id = testID, SalePrice = preds\_lasso1)  
colnames(sub2) <- c("Id","SalePrice")  
sub3 <- data.frame(Id = testID, SalePrice = preds\_rf1)  
colnames(sub3) <- c("Id","SalePrice")  
sub4 <- data.frame(Id = testID, SalePrice = preds\_gb1)  
colnames(sub4) <- c("Id","SalePrice")

write.csv(sub1, file="submission.csv", row.names = F)  
write.csv(sub2, file="submission2.csv", row.names = F) #LASSO   
write.csv(sub3, file="submission3.csv", row.names = F) #RF   
write.csv(sub4, file="submission4.csv", row.names = F) #GBM

seed : 555 final / lasso / rf / gbm 200207 score : 0.13790 / 0.14084 / 0.15367 / 0.13927