house\_price

Table of Contents

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

# Data load

library(tidyverse)

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

## √ ggplot2 3.2.1 √ purrr 0.3.2  
## √ tibble 2.1.3 √ dplyr 0.8.3  
## √ tidyr 1.0.0 √ stringr 1.4.0  
## √ readr 1.3.1 √ forcats 0.4.0

## -- Conflicts ---------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

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

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

library(scales)

##   
## Attaching package: 'scales'

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

## The following object is masked from 'package:readr':  
##   
## col\_factor

library(knitr)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

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

library(corrplot)

## corrplot 0.84 loaded

library(ggrepel)  
library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

library(xgboost)

##   
## Attaching package: 'xgboost'

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

library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

## Loading required package: foreach

##   
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':  
##   
## accumulate, when

## Loaded glmnet 2.0-18

library(Metrics)

## Warning: package 'Metrics' was built under R version 3.6.2

##   
## Attaching package: 'Metrics'

## The following object is masked from 'package:glmnet':  
##   
## auc

## The following objects are masked from 'package:caret':  
##   
## precision, recall

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)  
  
str(train)

## '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 ...

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"

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

## [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)

trainID <- train[,1]  
testID <- test[,1]  
  
all <- all[,-1]

# target variable

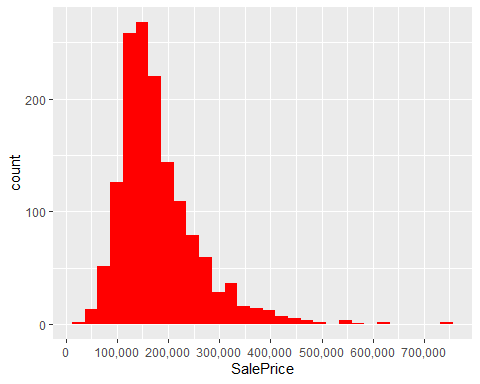
## SalePrice - target variable

summary(all$SalePrice)

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

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

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



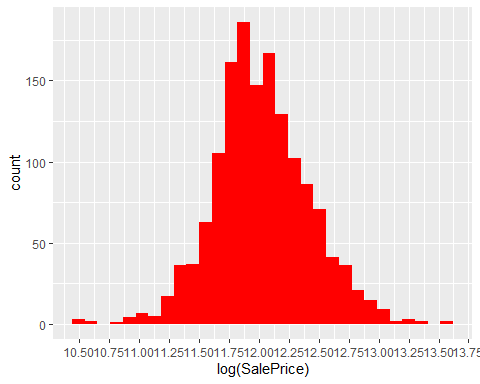
SalePrice의 분포를 보면, right-skewed된 분포임을 알 수 있다. 분산 안정화를 위해 log를 취하여 그래프를 그려보면,

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

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

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



평균이 12.02이고 어느 쪽으로 skewed된 경향이 감소했음을 알 수 있었다.

# Exploring data (Imputation)

## imputation

이제 variables를 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   
## 2909 2814 2721 2348 1459   
## FireplaceQu LotFrontage GarageYrBlt GarageFinish GarageQual   
## 1420 486 159 159 159   
## GarageCond GarageType BsmtCond BsmtExposure BsmtQual   
## 159 157 82 82 81   
## BsmtFinType2 BsmtFinType1 MasVnrType MasVnrArea MSZoning   
## 80 79 24 23 4   
## Utilities BsmtFullBath BsmtHalfBath Functional Exterior1st   
## 2 2 2 2 1   
## Exterior2nd BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF   
## 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),])

## [1] 2919

sum(!complete.cases(all))

## [1] 2919

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

### Pool

PoolQC : Pool quality

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

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

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.

그런 뒤 새로 Factor로 변환시키는데, level을 지정해준다. 순서형 변수이므로 순서로 변환.

library(plyr) #plyr::revalue()  
Qualities <- c('None'=0, 'Po'=1, "Fa"=2, "TA"=3, "Gd"=4, "Ex"=5)  
all$PoolQC <- as.integer(revalue(all$PoolQC, replace=Qualities))

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

table(all$PoolQC)

##   
## 0 2 4 5   
## 2909 2 4 4

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

all[all$PoolArea>0 & all$PoolQC==0, c("PoolArea","PoolQC","OverallQual")]

## PoolArea PoolQC OverallQual  
## 2421 368 0 4  
## 2504 444 0 6  
## 2600 561 0 3

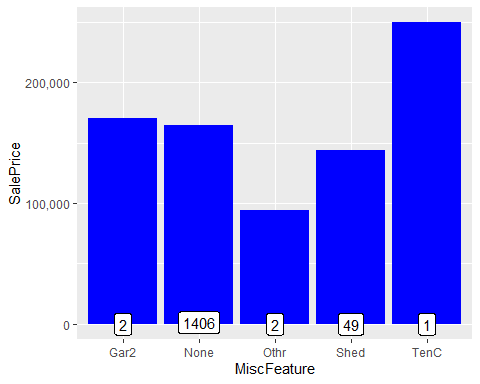
이를 보정해주었다. OverallQual은 집의 전반적인 퀄리티를 나타내는 변수로, 이를 이용해 적절하게 보정하였다.

all$PoolQC[2421] <- 2  
all$PoolQC[2504] <- 3  
all$PoolQC[2600] <- 2

### Miscellaneous

Elev Elevator  
 Gar2 2nd Garage (if not described in garage section)  
 Othr Other  
 Shed Shed (over 100 SF)  
 TenC Tennis Court  
 NA None

all$MiscFeature[is.na(all$MiscFeature)] <- 'None'  
all$MiscFeature <- as.character(all$MiscFeature)  
  
library(scales)  
ggplot(all[!is.na(all$SalePrice),], aes(x=MiscFeature, y=SalePrice)) +  
 geom\_bar(stat='summary', fun.y='median', fill='blue') +  
 scale\_y\_continuous(breaks = seq(0,800000, by=100000), labels = scales::comma) +  
 geom\_label(stat="count", aes(label=..count.., y=..count..))



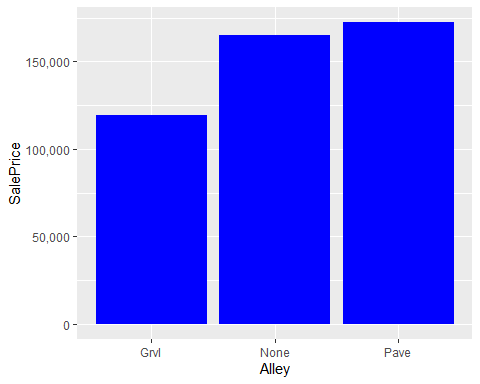
### Alley

Grvl Gravel  
 Pave Paved  
 NA No alley access

all %>% group\_by(Alley) %>% summarize(count = count(Alley))

## count.x count.freq  
## 1 Grvl 120  
## 2 Pave 78  
## 3 <NA> 2721

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



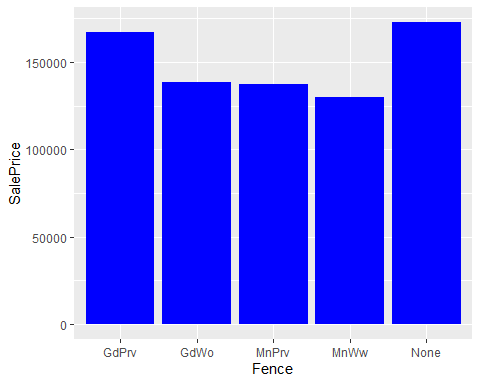
### Fence

GdPrv Good Privacy  
MnPrv Minimum Privacy  
GdWo Good Wood  
MnWw Minimum Wood/Wire  
NA No Fence

all %>% group\_by(Fence) %>% summarise(count(Fence))

## count(Fence).x count(Fence).freq  
## 1 GdPrv 118  
## 2 GdWo 112  
## 3 MnPrv 329  
## 4 MnWw 12  
## 5 <NA> 2348

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



### Fireplace

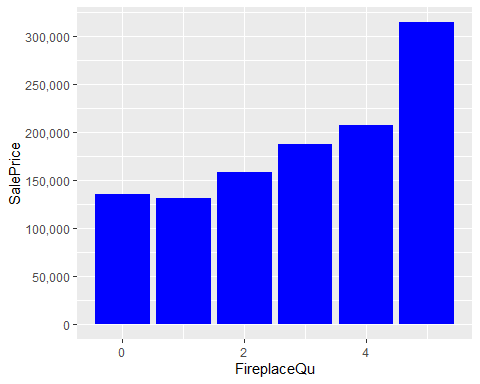
Ex Excellent - Exceptional Masonry Fireplace  
Gd Good - Masonry Fireplace in main level  
TA Average - Prefabricated Fireplace in main living area or Masonry Fireplace in basement  
Fa Fair - Prefabricated Fireplace in basement  
Po Poor - Ben Franklin Stove  
NA No Fireplace

all %>% group\_by(FireplaceQu) %>% summarize(count(FireplaceQu))

## count(FireplaceQu).x count(FireplaceQu).freq  
## 1 Ex 43  
## 2 Fa 74  
## 3 Gd 744  
## 4 Po 46  
## 5 TA 592  
## 6 <NA> 1420

all$FireplaceQu[is.na(all$FireplaceQu)] <- 'None'  
# Qualities None Po Fa TA Gd Ex   
# 0 1 2 3 4 5   
  
all$FireplaceQu <- as.integer(revalue(all$FireplaceQu , replace=Qualities))

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

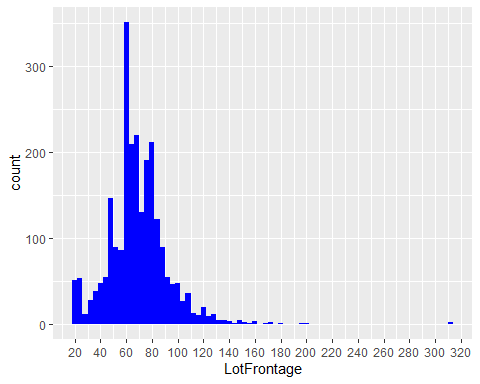


### lot

**LotFrontage : linear feet of street connected to property**

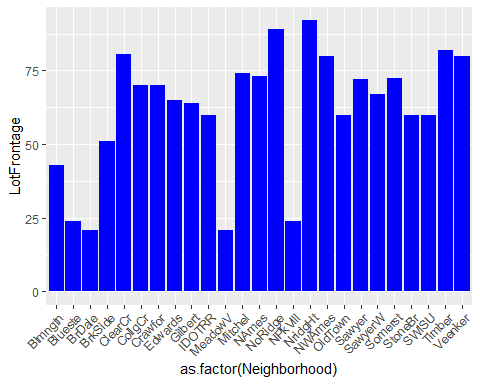
#all %>% group\_by(LotFrontage) %>% summarise(count(LotFrontage))  
  
ggplot(all, aes(x=LotFrontage)) +  
 geom\_histogram(fill='blue', binwidth = 4) +  
 scale\_x\_continuous(breaks=seq(0,320,by=20))

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



486 NAs. The most reasonable imputation seems to take the median per neigborhood.

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



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))   
 }  
}

**LotShape: General shape of property**

No NAs. Values seem ordinal (Regular=best)

Reg Regular   
 IR1 Slightly irregular  
 IR2 Moderately Irregular  
 IR3 Irregular

all$LotShape<-as.integer(revalue(all$LotShape, c('IR3'=0, 'IR2'=1, 'IR1'=2, 'Reg'=3)))  
table(all$LotShape)

##   
## 0 1 2 3   
## 16 76 968 1859

sum(table(all$LotShape))

## [1] 2919

### Garage

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

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

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

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. 따라서 159개의 GarageYrBlt 결측치는 집을 지을 때 차고도 같이 지었다고 imputation 하였다. 즉,

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

**GarageCars, GarageArea**

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

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>

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, GarageFinish를 보정하였고, 2577의 GarageType은 NA로 대체하였다.

#modes.  
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

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

**GarageType**

2Types More than one type of garage  
 Attchd Attached to home  
 Basment Basement Garage  
 BuiltIn Built-In (Garage part of house - typically has room above garage)  
 CarPort Car Port  
 Detchd Detached from home  
 NA No Garage

차고가 없음은 NA에서 No Garage 으로 바꿔주었다.

all$GarageType[is.na(all$GarageType)] <- 'No Garage'  
all$GarageType <- as.character(all$GarageType)  
table(all$GarageType)

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

**GarageFinish: Interior finish of the garage**

Fin Finished  
 RFn Rough Finished   
 Unf Unfinished  
 NA No Garage

차고 인테리어 마감 정도는 순서형(ordinal) 이므로 순서형으로 코딩하였다.

all$GarageFinish[is.na(all$GarageFinish)] <- 'None'  
Finish <- c('None'=0, 'Unf'=1, 'RFn'=2, 'Fin'=3)  
  
all$GarageFinish<-as.integer(revalue(all$GarageFinish, Finish))  
table(all$GarageFinish)

##   
## 0 1 2 3   
## 158 1231 811 719

**GarageQual: Garage quality**

Ex Excellent  
 Gd Good  
 TA Typical/Average  
 Fa Fair  
 Po Poor  
 NA No Garage

all$GarageQual[is.na(all$GarageQual)] <- 'None'  
all$GarageQual<-as.integer(revalue(all$GarageQual, Qualities))  
table(all$GarageQual)

##   
## 0 1 2 3 4 5   
## 158 5 124 2605 24 3

**GarageCond: Garage condition**

Ex Excellent  
 Gd Good  
 TA Typical/Average  
 Fa Fair  
 Po Poor  
 NA No Garage

all$GarageCond[is.na(all$GarageCond)] <- 'None'  
all$GarageCond<-as.integer(revalue(all$GarageCond, Qualities))  
table(all$GarageCond)

##   
## 0 1 2 3 4 5   
## 158 14 74 2655 15 3

### Basement Variables

**11개 변수에 대해결측 처리**

Five of those have 79-82 NAs, six have one or two NAs.

BsmtFinType1이 79개 결측치를 가졌다. 5개 변수가 같은 NA를 가지는 행인지 알아보고, 그렇지 않은 행을 추출해보면,

##### highlight  
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

9개 관찰값의 각 NA를 각 변수의 최빈값으로 보정해주었다.

#Imputing modes.  
##### hightlight  
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.

**BsmtQual: Evaluates the height of the basement**

A variable than can be made ordinal with the Qualities vector.

Ex Excellent (100+ inches)   
 Gd Good (90-99 inches)  
 TA Typical (80-89 inches)  
 Fa Fair (70-79 inches)  
 Po Poor (<70 inches  
 NA No Basement

all$BsmtQual[is.na(all$BsmtQual)] <- 'None'  
all$BsmtQual<-as.integer(revalue(all$BsmtQual, Qualities))  
table(all$BsmtQual)

##   
## 0 2 3 4 5   
## 79 88 1285 1209 258

**BsmtCond: Evaluates the general condition of the basement**

A variable than can be made ordinal with the Qualities vector.

Ex Excellent  
 Gd Good  
 TA Typical - slight dampness allowed  
 Fa Fair - dampness or some cracking or settling  
 Po Poor - Severe cracking, settling, or wetness  
 NA No Basement

all$BsmtCond[is.na(all$BsmtCond)] <- 'None'  
all$BsmtCond<-as.integer(revalue(all$BsmtCond, Qualities))  
table(all$BsmtCond)

##   
## 0 1 2 3 4   
## 79 5 104 2609 122

**BsmtExposure: Refers to walkout or garden level walls**

A variable than can be made ordinal.

Gd Good Exposure  
 Av Average Exposure (split levels or foyers typically score average or above)   
 Mn Mimimum Exposure  
 No No Exposure  
 NA No Basement

all$BsmtExposure[is.na(all$BsmtExposure)] <- 'None'  
Exposure <- c('None'=0, 'No'=1, 'Mn'=2, 'Av'=3, 'Gd'=4)  
  
all$BsmtExposure<-as.integer(revalue(all$BsmtExposure, Exposure))  
table(all$BsmtExposure)

##   
## 0 1 2 3 4   
## 79 1907 239 418 276

**BsmtFinType1: Rating of basement finished area**

A variable than can be made ordinal.

GLQ Good Living Quarters  
 ALQ Average Living Quarters  
 BLQ Below Average Living Quarters   
 Rec Average Rec Room  
 LwQ Low Quality  
 Unf Unfinshed  
 NA No Basement

all$BsmtFinType1[is.na(all$BsmtFinType1)] <- 'None'  
FinType <- c('None'=0, 'Unf'=1, 'LwQ'=2, 'Rec'=3, 'BLQ'=4, 'ALQ'=5, 'GLQ'=6)  
  
all$BsmtFinType1<-as.integer(revalue(all$BsmtFinType1, FinType))  
table(all$BsmtFinType1)

##   
## 0 1 2 3 4 5 6   
## 79 851 154 288 269 429 849

**BsmtFinType2: Rating of basement finished area (if multiple types)**

A variable than can be made ordinal with the FinType vector.

GLQ Good Living Quarters  
 ALQ Average Living Quarters  
 BLQ Below Average Living Quarters   
 Rec Average Rec Room  
 LwQ Low Quality  
 Unf Unfinshed  
 NA No Basement

all$BsmtFinType2[is.na(all$BsmtFinType2)] <- 'None'  
FinType <- c('None'=0, 'Unf'=1, 'LwQ'=2, 'Rec'=3, 'BLQ'=4, 'ALQ'=5, 'GLQ'=6)  
  
all$BsmtFinType2<-as.integer(revalue(all$BsmtFinType2, FinType))  
table(all$BsmtFinType2)

##   
## 0 1 2 3 4 5 6   
## 79 2494 87 105 68 52 34

여기까지 데이터 중 결측이 하나라도 없는 행의 갯수를 알아보면,

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

## [1] 2883

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

## [1] 235 530 651 937 974 978 1244 1279 1380 1556 1692 1707 1883 1916  
## [15] 1946 1993 2005 2042 2121 2152 2189 2217 2251 2312 2326 2341 2350 2369  
## [29] 2474 2490 2593 2611 2658 2687 2863 2905

all <- all[-which(complete.cases(subset(all,select=-c(SalePrice)))==F) ,]

전체 2919개 중 2883개(전체 중 98.77%)이다.

## classifying variable classes and encoding

아래 4개 변수는 char로 변형해준다.

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

### label encoding (ordinal)

순서형 변수들은 level대로 인코딩해준다.

all$ExterQual<-as.integer(revalue(all$ExterQual, c("Po"=0, "Fa"=1, "TA"=2, "Gd"=3, "Ex"=4)))

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

all$ExterCond<-as.integer(revalue(all$ExterCond, c("Po"=0, "Fa"=1, "TA"=2, "Gd"=3, "Ex"=4)))  
all$HeatingQC<-as.integer(revalue(all$HeatingQC, c("Po"=0, "Fa"=1, "TA"=2, "Gd"=3, "Ex"=4)))  
all$KitchenQual<-as.integer(revalue(all$KitchenQual, c("Po"=0, "Fa"=1, "TA"=2, "Gd"=3, "Ex"=4)))

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

all$Fence<-as.integer(revalue(all$Fence, c("None"=0, "MnWw"=1, "GdWo"=2, "MnPrv"=3, "GdPrv"=4)))  
all$PavedDrive<-as.integer(revalue(all$PavedDrive, c("N"=0, "P"=1, "Y"=2)))

all$TotalSF = all$TotalBsmtSF + all$X1stFlrSF + all$X2ndFlrSF  
all <- subset(all, select=-c(TotalBsmtSF, X1stFlrSF, X2ndFlrSF))

all 에는 character, numeric, 그리고 이미 변환한 Factor class로 이루어져 있다. 이를 분류하면,

chrvar <- names(all[,sapply(all,is.character)])  
chrvar

## [1] "MSSubClass" "MSZoning" "Street" "Alley"   
## [5] "LandContour" "Utilities" "LotConfig" "LandSlope"   
## [9] "Neighborhood" "Condition1" "Condition2" "BldgType"   
## [13] "HouseStyle" "OverallQual" "OverallCond" "RoofStyle"   
## [17] "RoofMatl" "Exterior1st" "Exterior2nd" "MasVnrType"   
## [21] "Foundation" "Heating" "CentralAir" "Electrical"   
## [25] "Functional" "GarageType" "MiscFeature" "MoSold"   
## [29] "YrSold" "SaleType" "SaleCondition"

numvar <- names(all[,sapply(all,is.numeric)])  
numvar

## [1] "LotFrontage" "LotArea" "LotShape" "YearBuilt"   
## [5] "YearRemodAdd" "MasVnrArea" "ExterQual" "ExterCond"   
## [9] "BsmtQual" "BsmtCond" "BsmtExposure" "BsmtFinType1"   
## [13] "BsmtFinSF1" "BsmtFinType2" "BsmtFinSF2" "BsmtUnfSF"   
## [17] "HeatingQC" "LowQualFinSF" "GrLivArea" "BsmtFullBath"   
## [21] "BsmtHalfBath" "FullBath" "HalfBath" "BedroomAbvGr"   
## [25] "KitchenAbvGr" "KitchenQual" "TotRmsAbvGrd" "Fireplaces"   
## [29] "FireplaceQu" "GarageYrBlt" "GarageFinish" "GarageCars"   
## [33] "GarageArea" "GarageQual" "GarageCond" "PavedDrive"   
## [37] "WoodDeckSF" "OpenPorchSF" "EnclosedPorch" "X3SsnPorch"   
## [41] "ScreenPorch" "PoolArea" "PoolQC" "Fence"   
## [45] "MiscVal" "SalePrice" "TotalSF"

facvar <- names(all[,sapply(all,is.factor)])  
facvar

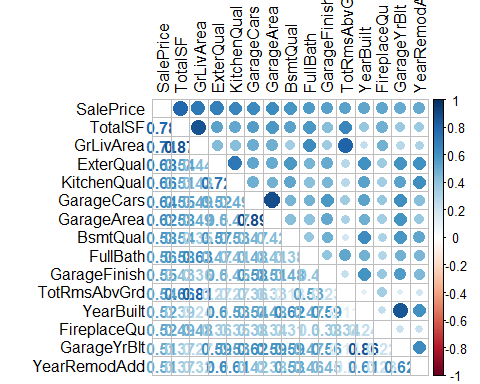
## character(0)

sum(length(chrvar),length(numvar),length(facvar)) #81

## [1] 78

**numeric class** 수치형 변수가 SalePrice와 어느정도 상관관계가 있는지 알아보기 위해 corrplot을 그려보면,

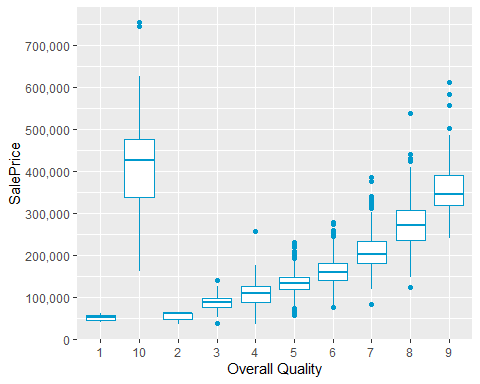
cor <- cor(all[,numvar], use = "pairwise.complete.obs")  
cor\_sort <- as.matrix(sort(cor[,'SalePrice'], decreasing = TRUE))  
corhigh <- names(which(apply(cor\_sort,1,function(x) abs(x)>0.5)))  
  
cor <- cor[corhigh,corhigh]  
  
corrplot.mixed(cor,tl.col="black", tl.pos = "lt")



SalePrice에 관련해서 가장 correlation이 높은 변수는 OverallQual, TotalSF, GrLivArea, GarageCars, ExterQual 등 순이었다. 그 중 GarageCars와 GarageArea는 0.89로 모든 변수들 중 가장 높은 corr을 가졌고, GrLivArea와 TotRmsAbvGrd도 0.81로 corr이 높았다.

SalePrice와 OverallQual 간 plot

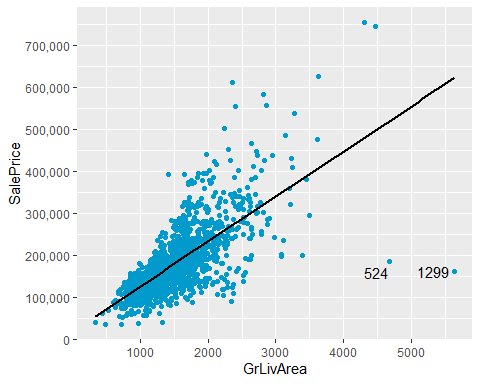
ggplot(data=all[!is.na(all$SalePrice),], aes(x=factor(OverallQual), y=SalePrice))+  
 geom\_boxplot(col='deepskyblue3') + labs(x='Overall Quality') +  
 scale\_y\_continuous(breaks= seq(0, 800000, by=100000), labels = comma)



Overall Quality가 증가할수록 SalePrice가 점점 증가하는 꼴을 볼 수 있다.  
OverallQual=4일 경우 이상치가 존재하는 것 같다.

SalePrice와 GrLivArea 간 plot

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")]

## SalePrice GrLivArea OverallQual  
## 525 315750 2398 7  
## 1307 202500 1368 7

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

chrvar

## [1] "MSSubClass" "MSZoning" "Street" "Alley"   
## [5] "LandContour" "Utilities" "LotConfig" "LandSlope"   
## [9] "Neighborhood" "Condition1" "Condition2" "BldgType"   
## [13] "HouseStyle" "OverallQual" "OverallCond" "RoofStyle"   
## [17] "RoofMatl" "Exterior1st" "Exterior2nd" "MasVnrType"   
## [21] "Foundation" "Heating" "CentralAir" "Electrical"   
## [25] "Functional" "GarageType" "MiscFeature" "MoSold"   
## [29] "YrSold" "SaleType" "SaleCondition"

numvar

## [1] "LotFrontage" "LotArea" "LotShape" "YearBuilt"   
## [5] "YearRemodAdd" "MasVnrArea" "ExterQual" "ExterCond"   
## [9] "BsmtQual" "BsmtCond" "BsmtExposure" "BsmtFinType1"   
## [13] "BsmtFinSF1" "BsmtFinType2" "BsmtFinSF2" "BsmtUnfSF"   
## [17] "HeatingQC" "LowQualFinSF" "GrLivArea" "BsmtFullBath"   
## [21] "BsmtHalfBath" "FullBath" "HalfBath" "BedroomAbvGr"   
## [25] "KitchenAbvGr" "KitchenQual" "TotRmsAbvGrd" "Fireplaces"   
## [29] "FireplaceQu" "GarageYrBlt" "GarageFinish" "GarageCars"   
## [33] "GarageArea" "GarageQual" "GarageCond" "PavedDrive"   
## [37] "WoodDeckSF" "OpenPorchSF" "EnclosedPorch" "X3SsnPorch"   
## [41] "ScreenPorch" "PoolArea" "PoolQC" "Fence"   
## [45] "MiscVal" "SalePrice" "TotalSF"

facvar

## character(0)

sum(length(chrvar),length(numvar),length(facvar)) #81

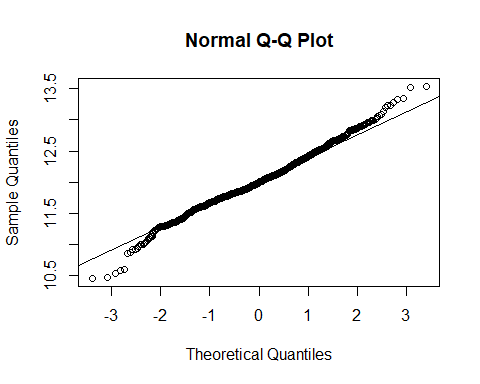
## [1] 78

### one-hot encodeing

dummies <- dummyVars(~., all[chrvar])  
categorical\_1\_hot <- predict(dummies, all[chrvar])  
  
all <- cbind(all[numvar], categorical\_1\_hot)

### log(SalePrice)

all$SalePrice <- log(all$SalePrice + 1)  
  
qqnorm(all$SalePrice)  
qqline(all$SalePrice)



## modeling

set.seed(191102)  
training <- all[!is.na(all$SalePrice),]  
testing <- all[is.na(all$SalePrice),]

### lasso

cv\_lasso = cv.glmnet(as.matrix(subset(training, select = -c(SalePrice))), training[,"SalePrice"])  
  
## Predictions  
preds = data.frame(exp(predict(cv\_lasso, newx = as.matrix(subset(testing, select = -c(SalePrice))),   
 s = "lambda.min")) - 1)