Untitled

김현우,박주연,이주영,이지예,주진영,홍정아

2018년 4월 27일

```
rm(list=ls())
Sys.setlocale('LC_ALL','C')
```

```
## [1] "C"
```

loading packages

```
##
## Attaching package: 'dplyr'
```

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

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

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## corrplot 0.84 loaded
```

```
##
## Attaching package: 'gridExtra'
```

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

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

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

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

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

```
##
## Attaching package: 'xgboost'
```

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

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: lattice
```

loading data

```
train <- read.csv("train.csv")
test <- read.csv("test.csv")
```

data structure

```
str(train) #'data.frame':   15129 obs. of  21 variables:
```

```
## 'data.frame':    15129 obs. of  21 variables:
##  $ price        : num  175003 705000 800000 300000 467000 ...
##  $ bedrooms     : int  3 6 3 2 3 3 3 4 2 3 ...
##  $ bathrooms    : num  1.5 2.75 1.75 1 2 2.5 2 2.25 1.75 2 ...
##  $ sqft_living  : int  1390 2830 1890 1290 1840 2100 2070 1800 1370 2168 ...
##  $ sqft_lot     : int  1882 10579 10292 2482 3432 15120 9000 7200 4495 4000 ...
##  $ floors       : num  2 1 1 2 2 1 1 1 1 1.5 ...
##  $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : int  3 4 4 3 3 4 4 3 4 3 ...
##  $ grade        : int  7 8 8 7 7 8 7 7 8 8 ...
##  $ sqft_above   : int  1390 1430 1890 1290 1840 2100 1450 1230 1370 2168 ...
##  $ sqft_basement: int  0 1400 0 0 0 0 620 570 0 0 ...
##  $ yr_built     : int  2014 1967 1969 2008 2012 1953 1969 1979 1975 1907 ...
##  $ yr_renovated : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ zipcode      : int  98108 98005 98040 98053 98155 98004 98023 98177 98198 98
105 ...
##  $ lat          : num  47.6 47.6 47.5 47.7 47.7 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  1490 2060 2630 1290 1280 3070 1630 2260 1370 1770 ...
##  $ sqft_lot15   : int  2175 10745 10625 2482 7573 16078 7885 7498 4686 4000 ...
##  $ sale_year    : num  2014 2014 2014 2015 2015 ...
##  $ sale_month   : num  12 10 12 4 3 1 12 3 5 9 ...
```

```
summary(train)
```

```
##      price            bedrooms         bathrooms      sqft_living
##  Min.   :  80000   Min.   : 0.000   Min.   :0.000   Min.   :  370
##  1st Qu.: 323800   1st Qu.: 3.000   1st Qu.:1.750   1st Qu.: 1430
##  Median : 450000   Median : 3.000   Median :2.250   Median : 1920
##  Mean   : 540778   Mean   : 3.371   Mean   :2.116   Mean   : 2082
##  3rd Qu.: 648000   3rd Qu.: 4.000   3rd Qu.:2.500   3rd Qu.: 2550
##  Max.   :7700000   Max.   :33.000   Max.   :8.000   Max.   :12050
##     sqft_lot           floors        waterfront           view
##  Min.   :    520   Min.   :1.000   Min.   :0.000000   Min.   :0.0000
##  1st Qu.:   5085   1st Qu.:1.000   1st Qu.:0.000000   1st Qu.:0.0000
##  Median :   7641   Median :1.500   Median :0.000000   Median :0.0000
##  Mean   :  15438   Mean   :1.492   Mean   :0.007601   Mean   :0.2399
##  3rd Qu.:  10800   3rd Qu.:2.000   3rd Qu.:0.000000   3rd Qu.:0.0000
##  Max.   :1164794   Max.   :3.500   Max.   :1.000000   Max.   :4.0000
##    condition          grade          sqft_above   sqft_basement
##  Min.   :1.000   Min.   : 3.000   Min.   : 370   Min.   :   0.0
##  1st Qu.:3.000   1st Qu.: 7.000   1st Qu.:1200   1st Qu.:   0.0
##  Median :3.000   Median : 7.000   Median :1570   Median :   0.0
##  Mean   :3.408   Mean   : 7.661   Mean   :1792   Mean   : 290.5
##  3rd Qu.:4.000   3rd Qu.: 8.000   3rd Qu.:2210   3rd Qu.: 560.0
##  Max.   :5.000   Max.   :13.000   Max.   :8860   Max.   :4820.0
##     yr_built     yr_renovated        zipcode            lat
##  Min.   :1900   Min.   :   0.00   Min.   :98001   Min.   :47.16
##  1st Qu.:1951   1st Qu.:   0.00   1st Qu.:98033   1st Qu.:47.47
##  Median :1975   Median :   0.00   Median :98065   Median :47.57
##  Mean   :1971   Mean   :  85.51   Mean   :98078   Mean   :47.56
##  3rd Qu.:1996   3rd Qu.:   0.00   3rd Qu.:98118   3rd Qu.:47.68
##  Max.   :2015   Max.   :2015.00   Max.   :98199   Max.   :47.78
##       long         sqft_living15   sqft_lot15        sale_year
##  Min.   :-122.5   Min.   : 460   Min.   :   659   Min.   :2014
##  1st Qu.:-122.3   1st Qu.:1490   1st Qu.:  5100   1st Qu.:2014
##  Median :-122.2   Median :1840   Median :  7649   Median :2014
##  Mean   :-122.2   Mean   :1988   Mean   : 12986   Mean   :2014
##  3rd Qu.:-122.1   3rd Qu.:2370   3rd Qu.: 10125   3rd Qu.:2015
##  Max.   :-121.3   Max.   :6210   Max.   :858132   Max.   :2015
##    sale_month
##  Min.   : 1.000
##  1st Qu.: 4.000
##  Median : 7.000
##  Mean   : 6.607
##  3rd Qu.: 9.000
##  Max.   :12.000
```

missing data

```
cat("train missing data...\n")
```

```
## train missing data...
```

```
apply(train,2,function(x) sum(is.na(x))) #NA는 없음.
```

```
##         price      bedrooms     bathrooms   sqft_living      sqft_lot
##             0             0             0             0             0
##        floors    waterfront          view     condition         grade
##             0             0             0             0             0
##    sqft_above sqft_basement      yr_built  yr_renovated       zipcode
##             0             0             0             0             0
##           lat          long sqft_living15    sqft_lot15     sale_year
##             0             0             0             0             0
##    sale_month
##             0
```

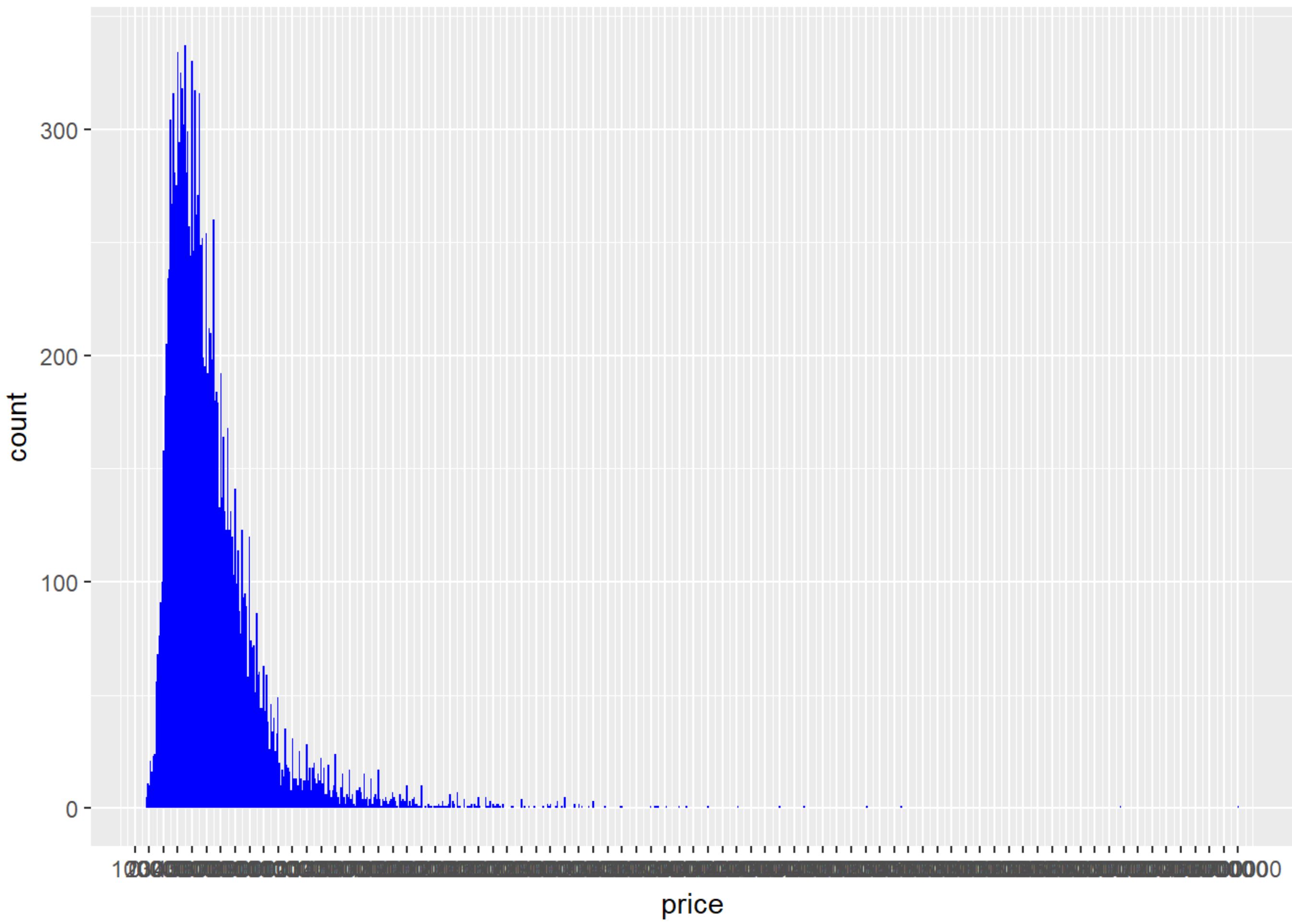
```
cat("\ntest missing data...\n")
```

```
##
## test missing data...
```

```
apply(test,2,function(x) sum(is.na(x))) #NA는 없음.
```

```
##         price      bedrooms     bathrooms   sqft_living      sqft_lot
##             0             0             0             0             0
##        floors    waterfront          view     condition         grade
##             0             0             0             0             0
##    sqft_above sqft_basement      yr_built  yr_renovated       zipcode
##             0             0             0             0             0
##           lat          long sqft_living15    sqft_lot15     sale_year
##             0             0             0             0             0
##    sale_month
##             0
```

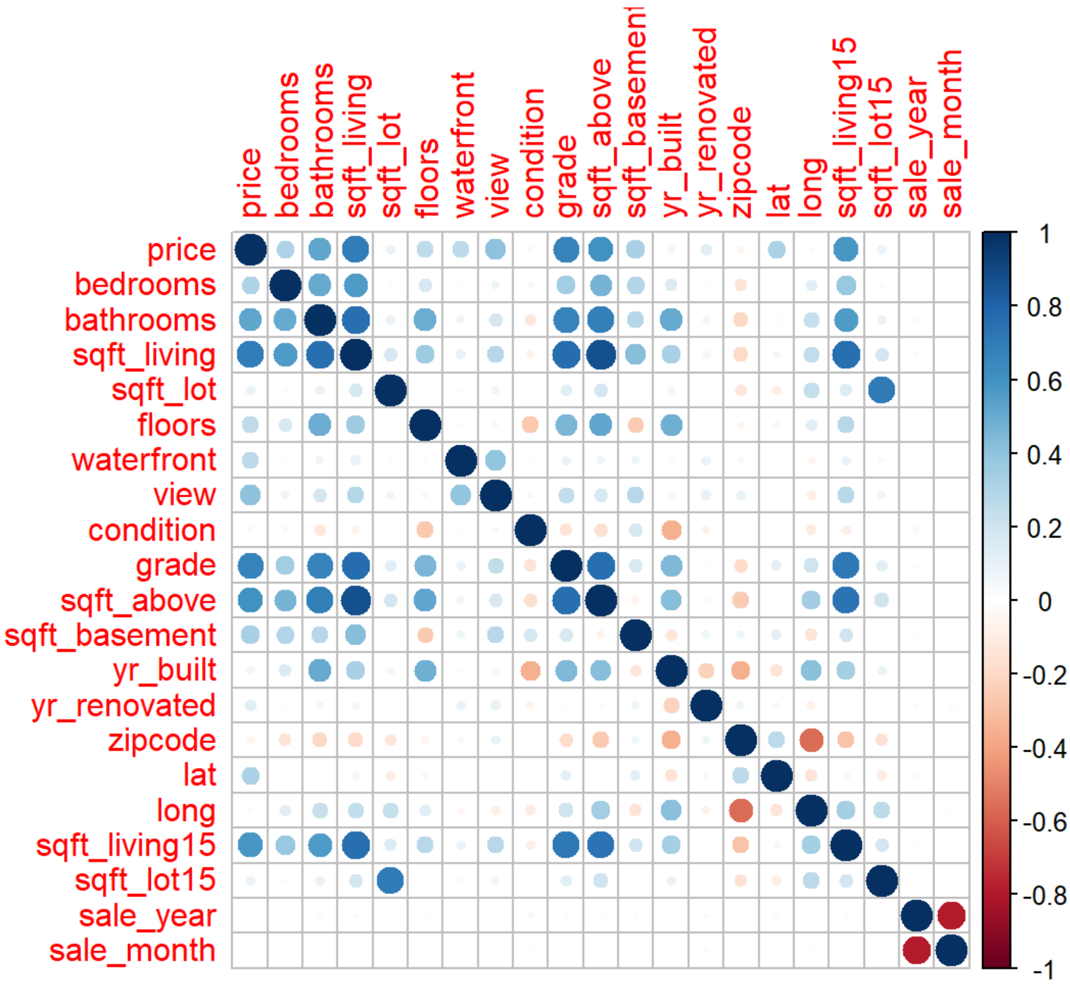
```
ggplot(data=train, aes(x=price)) +
        geom_histogram(fill="blue", binwidth = 10000) +
        scale_x_continuous(breaks= seq(0, 7700000, by=100000))
```




```
summary(train$price)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    80000  323800  450000  540778  648000 7700000
```

```
m <- cor(train)
corrplot(m,method="circle") #method에 따라서 그림이 다름. circle 치면 원형태로 나옴.
```

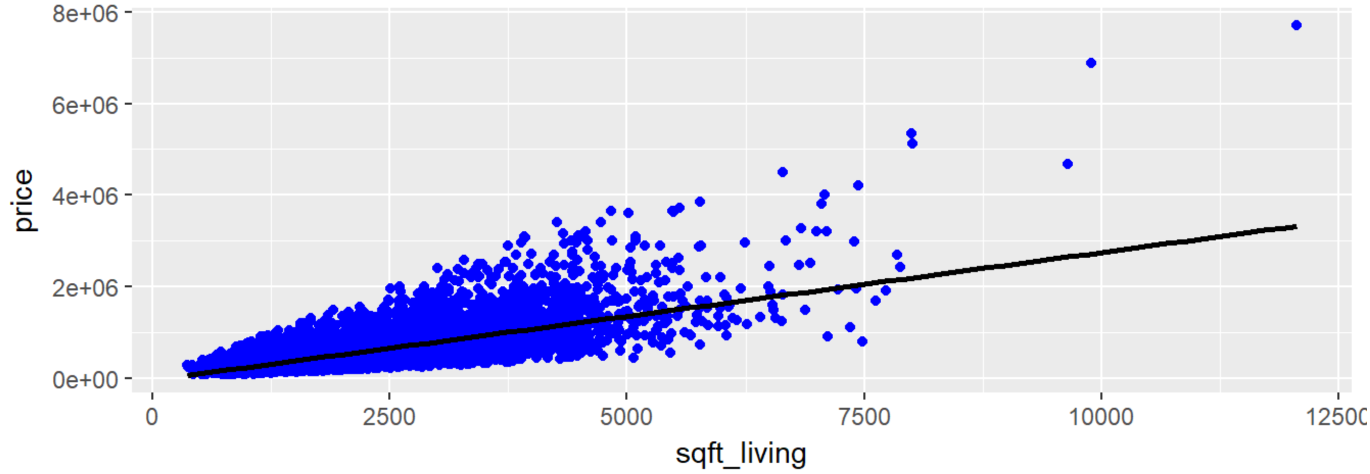


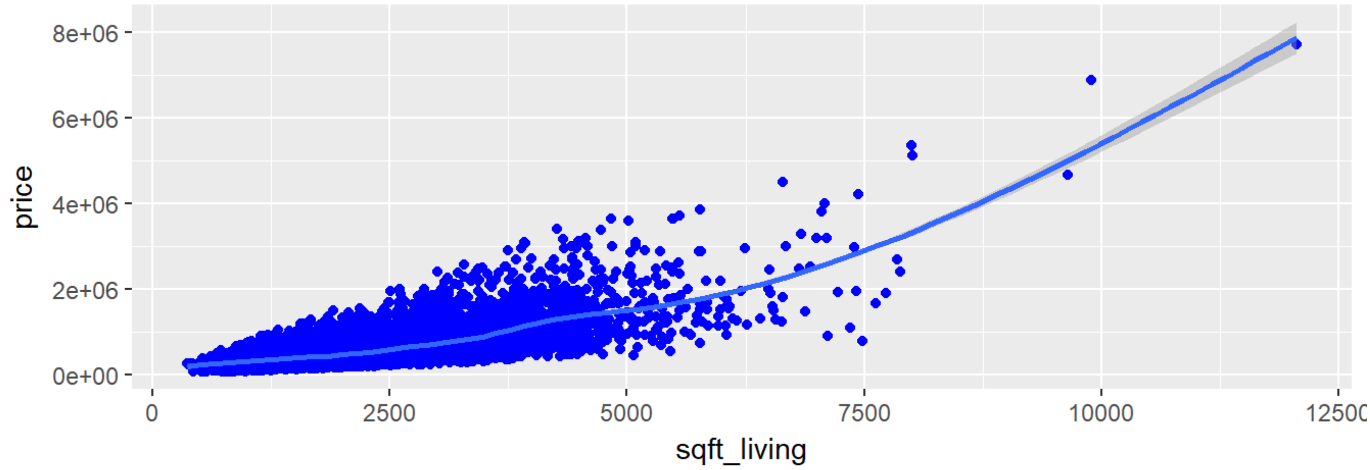
하지만 변수가 많아서 보기가 불편함. 그래서 price와 상관관계가 높은애들만 따로 추출해줄것임.

```
numericVars <- which(sapply(train, is.numeric)) #index vector numeric variables
numericVarNames <- names(numericVars) #saving names vector for use later on
#cat('There are', length(numericVars), 'numeric variables')

train_numVar <- train[, numericVars]
cor_numVar <- cor(train_numVar, use="pairwise.complete.obs") #correlations of train
numeric variables

#sort on decreasing correlations with price
cor_sorted <- as.matrix(sort(cor_numVar[,'price'], decreasing = TRUE))
 #select only high corelations
CorHigh <- names(which(apply(cor_sorted, 1, function(x) abs(x)>0.5)))
cor_numVar <- cor_numVar[CorHigh, CorHigh]

corrplot.mixed(cor_numVar, tl.col="black", tl.pos = "lt")
```


```
p1<- ggplot(data=train, aes(x=sqft_living, y=price))+
        geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black
", aes(group=1)) +                              labs(x='sqft_living')

p2<- ggplot(data=train, aes(x=sqft_living, y=price)) +
        geom_point(col='blue') + geom_smooth() +  labs(x='sqft_living')

grid.arrange(p1,p2,nrow=2)
```

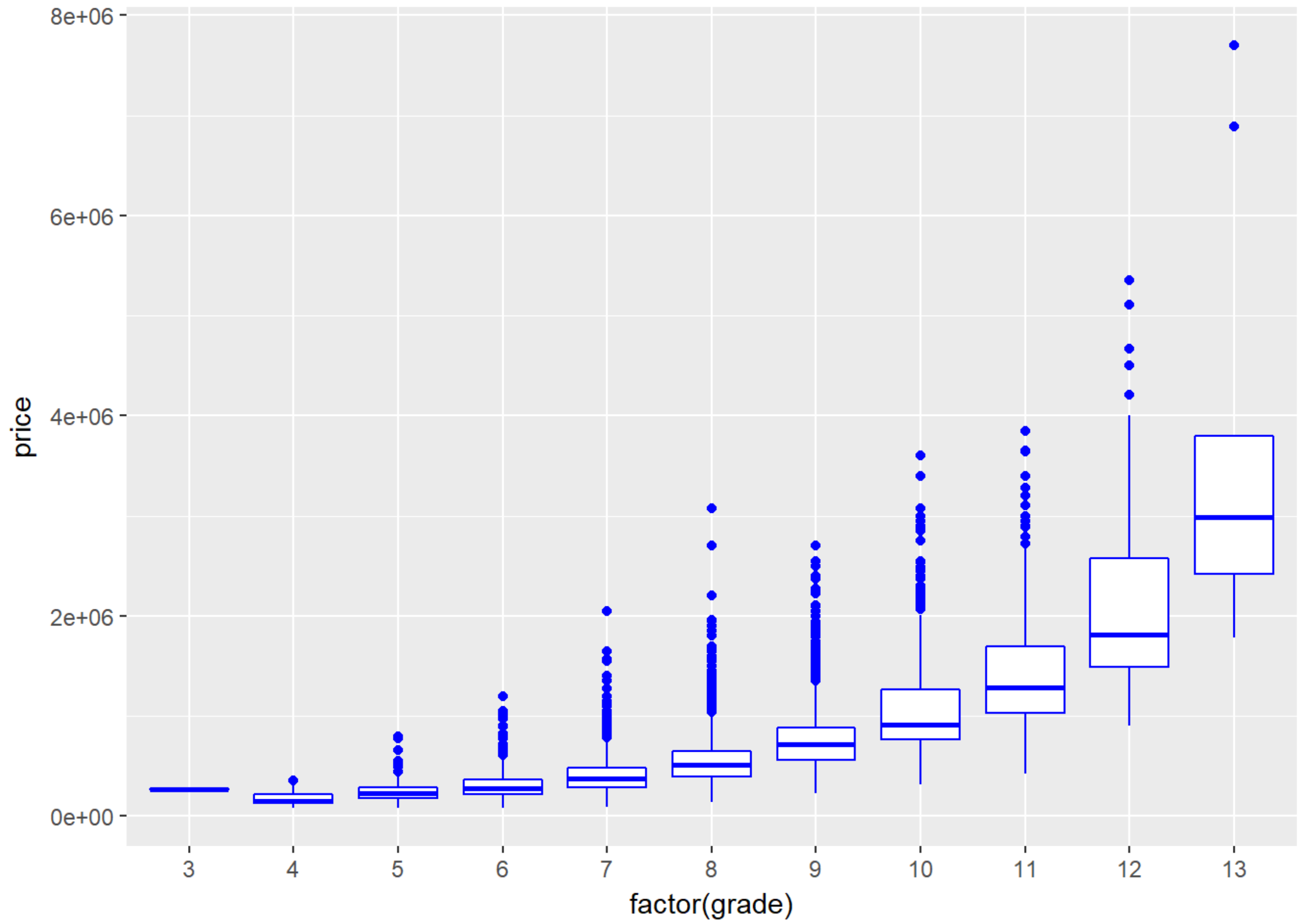
```
## `geom_smooth()` using method = 'gam'
```



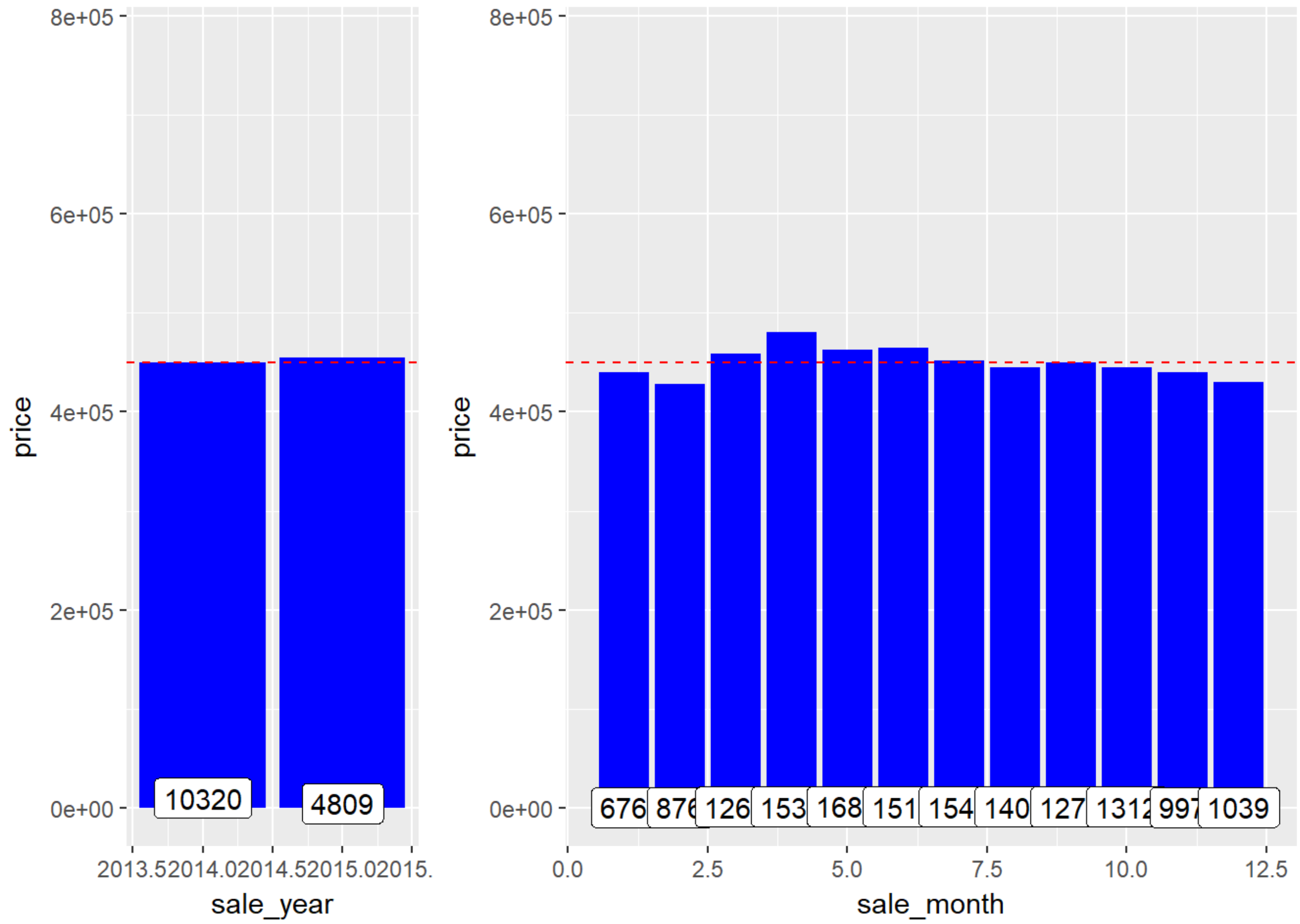


method = gam과 lm이 굉장히 다른 모습을 보여줌.

```
ggplot(data=train, aes(x=factor(grade), y=price)) +
        geom_boxplot(col='blue')
```

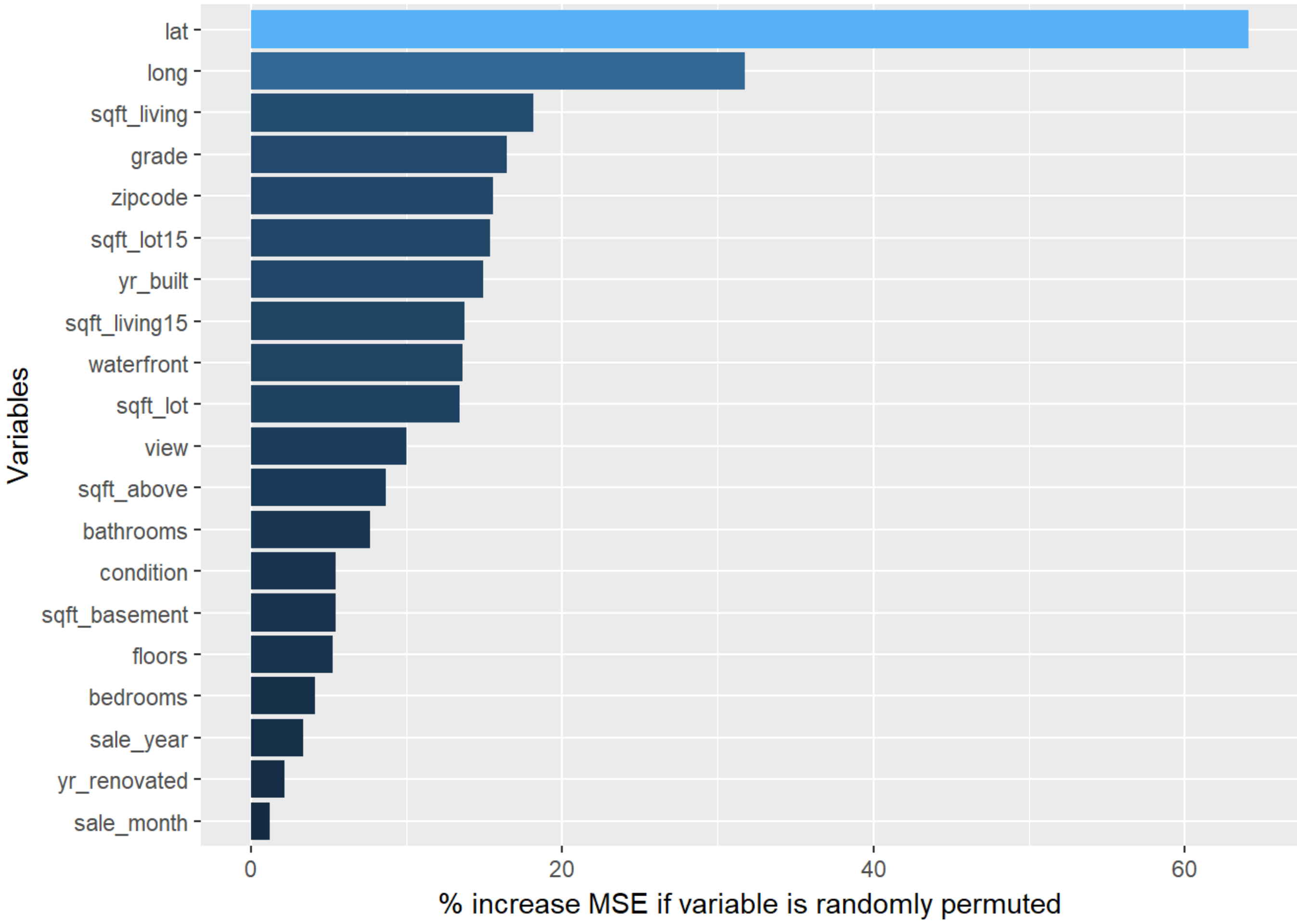



```
ys <- ggplot(train, aes(x=sale_year, y=price)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue')+
        geom_label(stat = "count", aes(label = ..count.., y = ..count..)) +
        coord_cartesian(ylim = c(0, 770000)) +
        geom_hline(yintercept=450000, linetype="dashed", color = "red") #dashed lin
e is median price

ms <- ggplot(train, aes(x=sale_month, y=price)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue')+
        geom_label(stat = "count", aes(label = ..count.., y = ..count..)) +
        coord_cartesian(ylim = c(0, 770000)) +
        geom_hline(yintercept=450000, linetype="dashed", color = "red") #dashed lin
e is median price

grid.arrange(ys, ms, widths=c(1,2))
```

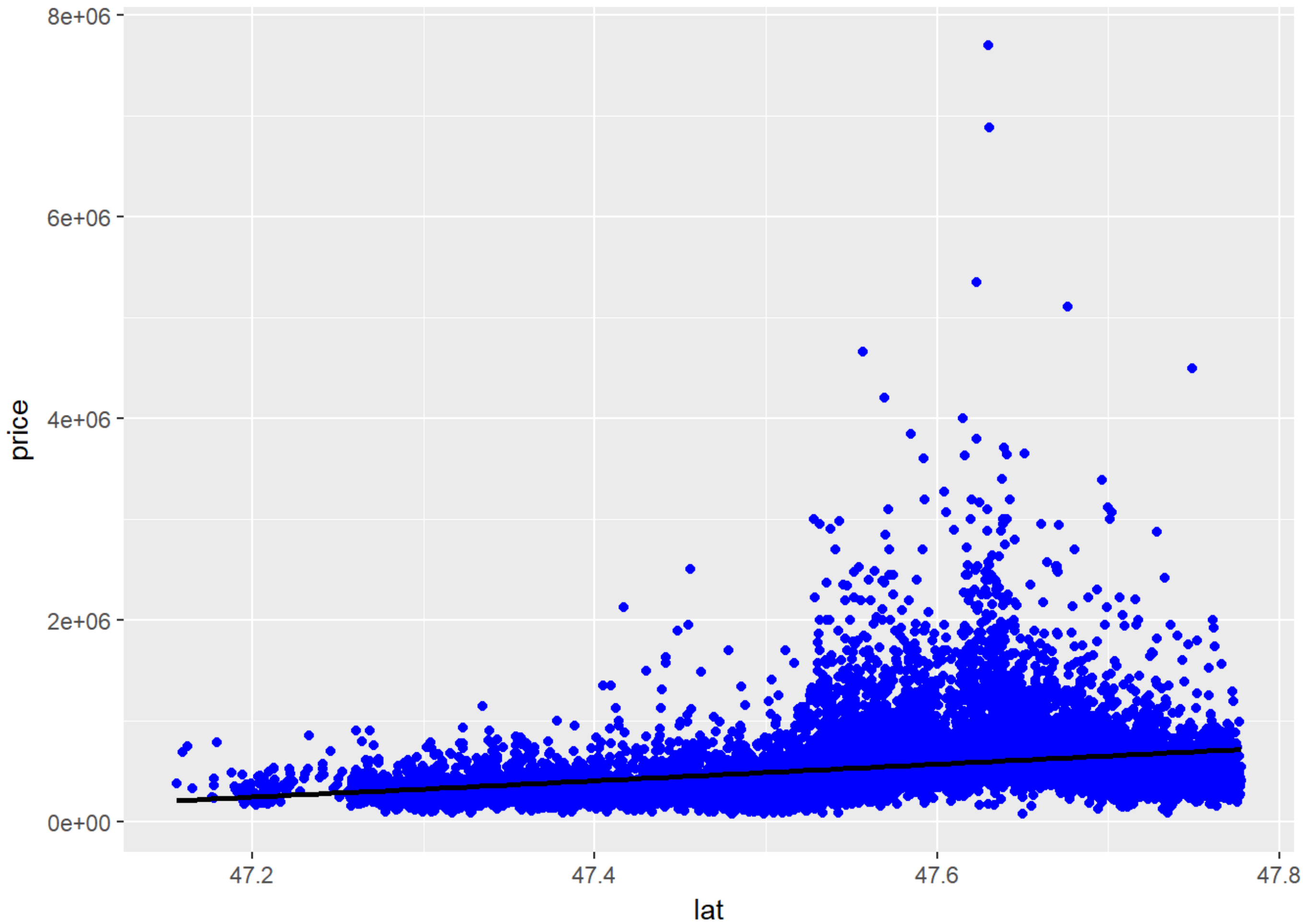


random forest (finding importance variable)

```
set.seed(2018)
quick_RF <- randomForest(x=train[1:15129,2:21], y=train$price, ntree=100,importance
=TRUE)
imp_RF <- importance(quick_RF)
imp_DF <- data.frame(Variables = row.names(imp_RF), MSE = imp_RF[,1])
imp_DF <- imp_DF[order(imp_DF$MSE, decreasing = TRUE),]

ggplot(imp_DF[1:20,], aes(x=reorder(Variables, MSE), y=MSE, fill=MSE)) + geom_bar(s
tat = 'identity') +
  labs(x = 'Variables', y= '% increase MSE if variable is randomly permuted') +
  coord_flip() +
  theme(legend.position="none")
```




```
ggplot(data=train, aes(x=lat, y=price))+
        geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black
", aes(group=1)) +                     labs(x='lat')
```

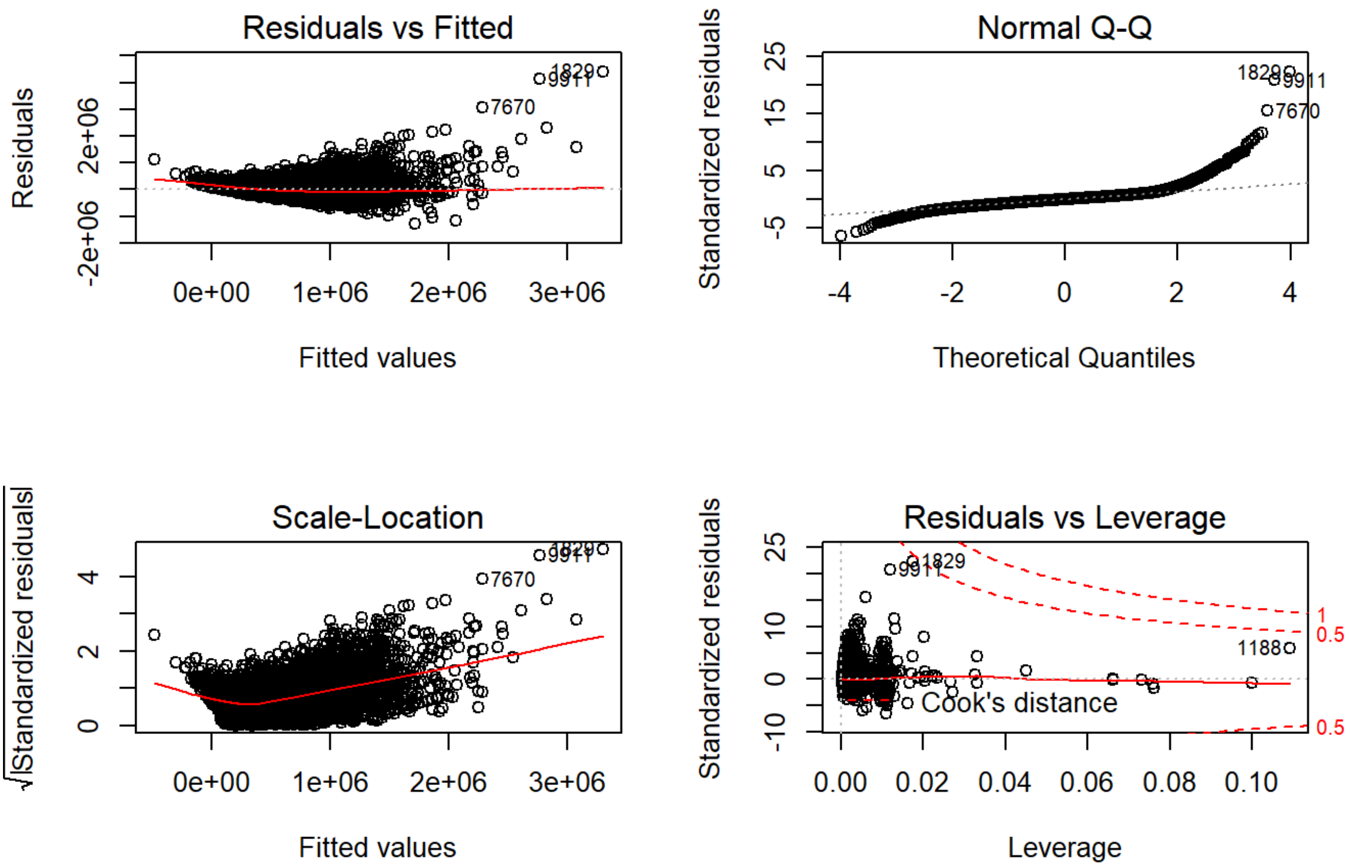



```
        #geom_text_repel(aes(label = ifelse(train$price>6000000, rownames(train), '
')))
```

```
model <- lm(price~.,data = train)
summary(model)
```

```
##
## Call:
## lm(formula = price ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1289272   -98433    -9562    76172  4400147
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.428e+07  1.179e+07  -6.301 3.03e-10 ***
## bedrooms      -3.369e+04  2.203e+03 -15.296  < 2e-16 ***
## bathrooms      4.165e+04  3.863e+03  10.782  < 2e-16 ***
## sqft_living    1.444e+02  5.198e+00  27.778  < 2e-16 ***
## sqft_lot       1.202e-01  5.446e-02   2.207 0.027360 *
## floors         5.802e+03  4.248e+03   1.366 0.172069
## waterfront     5.701e+05  2.039e+04  27.952  < 2e-16 ***
## view           5.602e+04  2.482e+03  22.572  < 2e-16 ***
## condition      2.925e+04  2.802e+03  10.439  < 2e-16 ***
## grade          9.494e+04  2.551e+03  37.217  < 2e-16 ***
## sqft_above     2.731e+01  5.154e+00   5.299 1.18e-07 ***
## sqft_basement         NA         NA      NA       NA
## yr_built      -2.489e+03  8.579e+01 -29.018  < 2e-16 ***
## yr_renovated   2.151e+01  4.304e+00   4.997 5.89e-07 ***
## zipcode       -5.576e+02  3.909e+01 -14.265  < 2e-16 ***
## lat            6.133e+05  1.268e+04  48.360  < 2e-16 ***
## long          -2.092e+05  1.559e+04 -13.422  < 2e-16 ***
## sqft_living15  2.897e+01  4.061e+00   7.134 1.02e-12 ***
## sqft_lot15    -2.804e-01  8.390e-02  -3.342 0.000833 ***
## sale_year      3.893e+04  5.581e+03   6.976 3.16e-12 ***
## sale_month     1.914e+03  8.338e+02   2.296 0.021713 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 198800 on 15109 degrees of freedom
## Multiple R-squared:  0.7017, Adjusted R-squared:  0.7013
## F-statistic:  1870 on 19 and 15109 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(model)
```



- normal qq가 1829,9911,7670 3개에 굉장히 흔들림. + 애초에 직선모양이 아님. - scale-location을 보면 값들의 분포가 일정하지 않은걸 알 수 있음.(그리고 양쪽으로 갈 수록 잔차가 커짐) - Residuals vs leverage를 보면 3값 9911,1829,1188이 예측치와 distance가 많이 멀음.

```
#vif(lm(price~.,data = train))
#Error in vif.default(lm(price ~ ., data = train)) : there are aliased coefficients in the model
#이유는 sqft_basement라는 column이 NA값을 가지고 있어서임.
```

Model 수정

기존의 Adjusted R-squared: 0.7013

- 1. 이상치제거
- 2. sqft_basement제거
- 3. 가정에 부합하게 수저

이상치제거

```
train_1 <- train[-c(1188,1829,7670,9911),]
train_1 <- train_1 %>% filter(price<6000000)
```

sqft_basement제거

```
train_1 <- train_1 %>% select(-sqft_basement)
test_1 <- test %>% select(-sqft_basement)
```

가정에 부합하게 수저

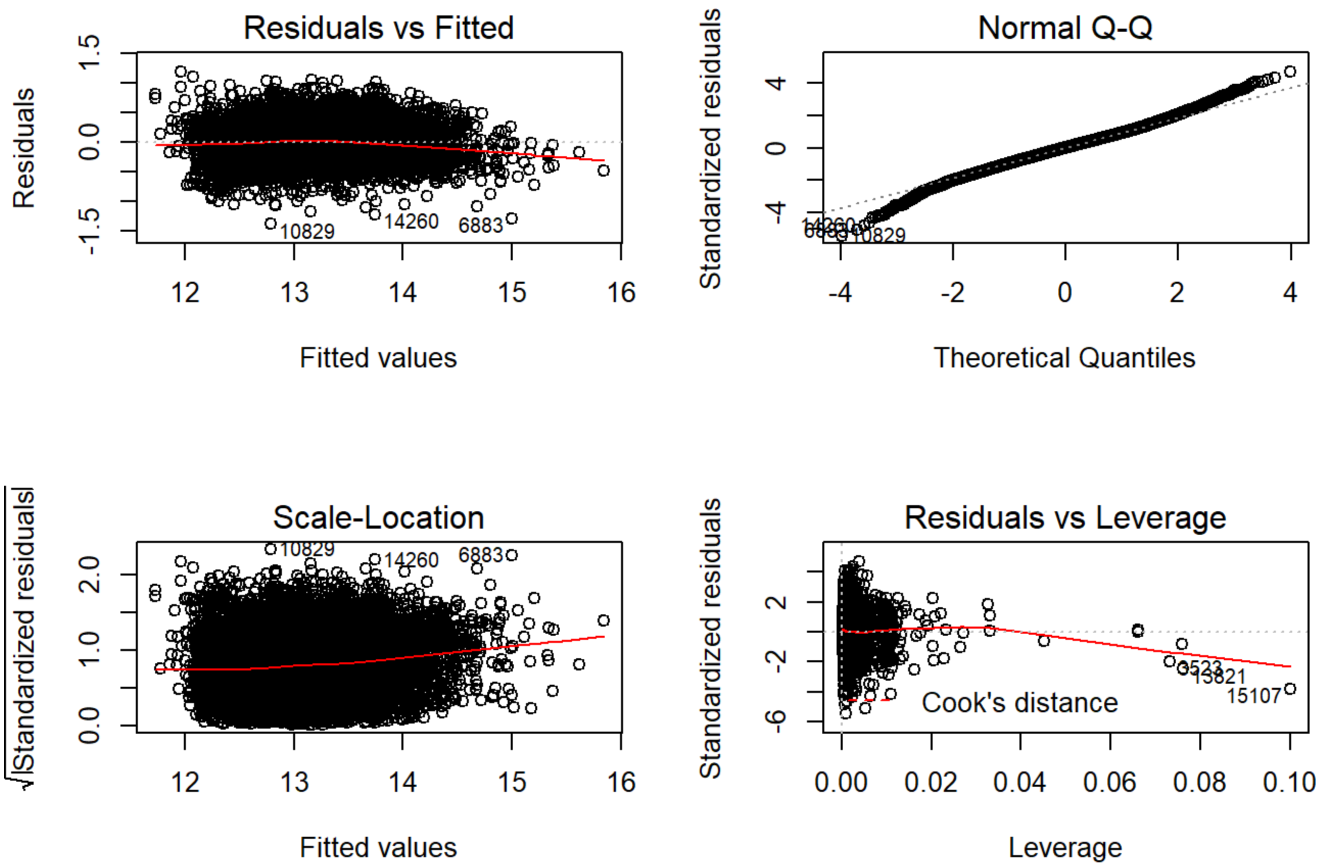
```r
train_1$price <- log(train_1$price + 1)
```

```r
model <- lm(price~.,data = train_1)
summary(model)
```

```
## 
## Call:
## lm(formula = price ~ ., data = train_1)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.38060 -0.15967  0.00275  0.15958  1.18858 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   -1.370e+02  1.500e+01  -9.132  < 2e-16 ***
## bedrooms      -1.475e-02  2.974e-03  -4.959 7.15e-07 ***
## bathrooms      6.427e-02  4.929e-03  13.040  < 2e-16 ***
## sqft_living    1.568e-04  6.656e-06  23.553  < 2e-16 ***
## sqft_lot       3.989e-07  6.930e-08   5.755 8.82e-09 ***
## floors         7.846e-02  5.410e-03  14.503  < 2e-16 ***
## waterfront     3.673e-01  2.596e-02  14.149  < 2e-16 ***
## view           6.339e-02  3.162e-03  20.047  < 2e-16 ***
## condition      6.763e-02  3.565e-03  18.972  < 2e-16 ***
## grade          1.554e-01  3.251e-03  47.796  < 2e-16 ***
## sqft_above    -1.992e-05  6.569e-06  -3.033  0.00242 ** 
## yr_built      -3.257e-03  1.092e-04 -29.821  < 2e-16 ***
## yr_renovated   4.633e-05  5.478e-06   8.458  < 2e-16 ***
## zipcode       -6.750e-04  4.974e-05 -13.570  < 2e-16 ***
## lat            1.417e+00  1.614e-02  87.782  < 2e-16 ***
## long          -1.534e-01  1.984e-02  -7.735 1.10e-14 ***
## sqft_living15  1.048e-04  5.177e-06  20.243  < 2e-16 ***
## sqft_lot15    -6.561e-08  1.068e-07  -0.615  0.53882    
## sale_year      6.669e-02  7.101e-03   9.391  < 2e-16 ***
## sale_month     3.066e-03  1.061e-03   2.890  0.00386 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2529 on 15105 degrees of freedom
## Multiple R-squared:  0.7687, Adjusted R-squared:  0.7684 
## F-statistic:  2642 on 19 and 15105 DF,  p-value: < 2.2e-16
```

R-squared가 0.7013에서 0.7684로 상승한 것을 확인할 수 있음.

```r
par(mfrow = c(2, 2))
plot(model)
```



Normal qq가 좋아져지만, 이젠 아래쪽에서 문제가 좀 있는게 보이고, 나머지는 더 안좋아진것 처럼 보이지만 y값이 달려져서 그렇지 위에보다 좋음.

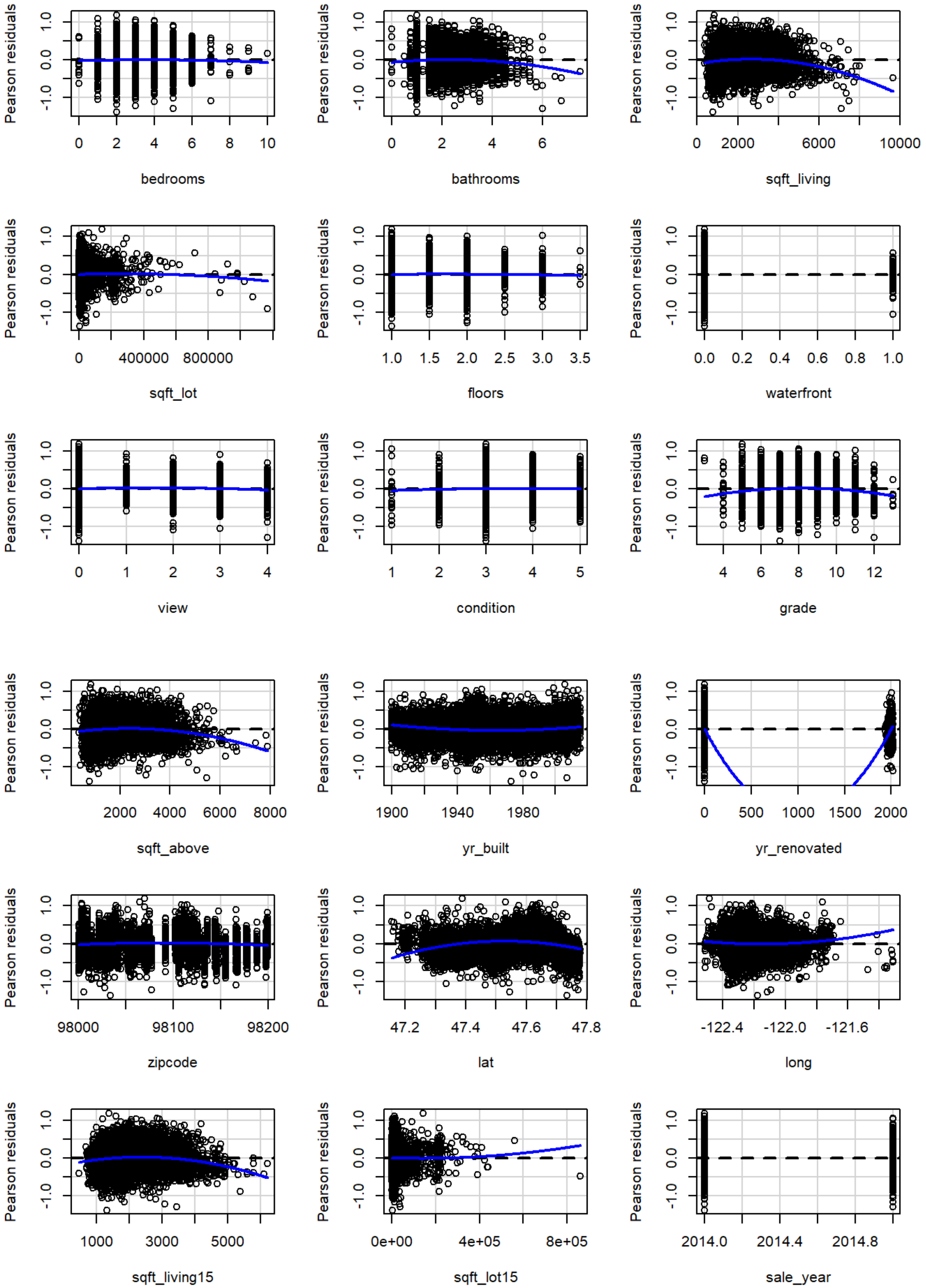
```
vif(lm(price~.,data = train_1)) #보통은 10이상이면 제거해준다고 함 sqft_living이 그나마
큰 상황. 5정도로 보는 시각도 있음.
```

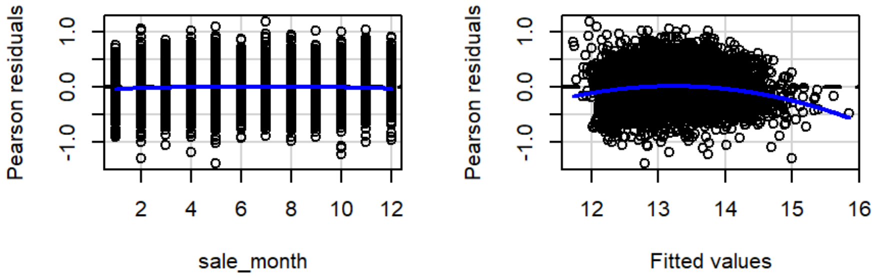
```
##       bedrooms     bathrooms   sqft_living      sqft_lot        floors
##       1.705181      3.356991      8.578401      2.004926      2.003521
##     waterfront          view     condition         grade    sqft_above
##       1.202426      1.425407      1.258053      3.444832      6.862762
##       yr_built  yr_renovated       zipcode           lat          long
##       2.437482      1.157631      1.671640      1.179326      1.840786
## sqft_living15    sqft_lot15     sale_year    sale_month
##       2.982020      2.039123      2.584584      2.576635
```

```
sqrt(vif(lm(price~.,data = train_1))) > sqrt(10)
```

```
##       bedrooms     bathrooms   sqft_living      sqft_lot        floors
##          FALSE         FALSE         FALSE         FALSE         FALSE
##     waterfront          view     condition         grade    sqft_above
##          FALSE         FALSE         FALSE         FALSE         FALSE
##       yr_built  yr_renovated       zipcode           lat          long
##          FALSE         FALSE         FALSE         FALSE         FALSE
## sqft_living15    sqft_lot15     sale_year    sale_month
##          FALSE         FALSE         FALSE         FALSE
```

```
residualPlots(model)
```






```
##                 Test stat Pr(>|Test stat|)
## bedrooms          -1.2137        0.2248660
## bathrooms         -6.8566        7.327e-12 ***
## sqft_living      -13.8171        < 2.2e-16 ***
## sqft_lot          -2.0236        0.0430249 *
## floors            -1.8624        0.0625598 .
## waterfront        -0.6007        0.5480380
## view              -3.4237        0.0006195 ***
## condition         -1.8299        0.0672848 .
## grade             -9.9286        < 2.2e-16 ***
## sqft_above       -11.1345        < 2.2e-16 ***
## yr_built          19.5249        < 2.2e-16 ***
## yr_renovated       6.8858        5.971e-12 ***
## zipcode           -8.0394        9.693e-16 ***
## lat              -34.4195        < 2.2e-16 ***
## long               8.4870        < 2.2e-16 ***
## sqft_living15    -13.4144        < 2.2e-16 ***
## sqft_lot15         1.7422        0.0814853 .
## sale_year         -2.0957        0.0361242 *
## sale_month        -8.0381        9.793e-16 ***
## Tukey test       -16.1485        < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#https://kin.naver.com/qna/detail.nhn?d1id=11&dirId=1113&docId=212884389&qb=Z3ZsbWE=&enc=utf8&section=kin&rank=1&search_sort=0&spq=0&pid=Tx0v6lpySDossvkKYtdssssss6R-200040&sid=iCXpWENrJvIgAyK53tD2zA%3D%3D 이거 보고 좀더 확장시킬 수 있겠다.
```

위 test의 Null은 “Model is additive”라서 이걸 기각하면 문제가 있다는 의미. 마지막 그래프가 잔차인데,

```
assumption <- gvlma::gvlma(model)
summary(assumption)
```

```
##
## Call:
## lm(formula = price ~ ., data = train_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.38060 -0.15967  0.00275  0.15958  1.18858
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.370e+02  1.500e+01  -9.132  < 2e-16 ***
## bedrooms      -1.475e-02  2.974e-03  -4.959 7.15e-07 ***
## bathrooms      6.427e-02  4.929e-03  13.040  < 2e-16 ***
## sqft_living    1.568e-04  6.656e-06  23.553  < 2e-16 ***
## sqft_lot       3.989e-07  6.930e-08   5.755 8.82e-09 ***
## floors         7.846e-02  5.410e-03  14.503  < 2e-16 ***
## waterfront     3.673e-01  2.596e-02  14.149  < 2e-16 ***
## view           6.339e-02  3.162e-03  20.047  < 2e-16 ***
## condition      6.763e-02  3.565e-03  18.972  < 2e-16 ***
## grade          1.554e-01  3.251e-03  47.796  < 2e-16 ***
## sqft_above    -1.992e-05  6.569e-06  -3.033  0.00242 **
## yr_built      -3.257e-03  1.092e-04 -29.821  < 2e-16 ***
## yr_renovated   4.633e-05  5.478e-06   8.458  < 2e-16 ***
## zipcode       -6.750e-04  4.974e-05 -13.570  < 2e-16 ***
## lat            1.417e+00  1.614e-02  87.782  < 2e-16 ***
## long          -1.534e-01  1.984e-02  -7.735 1.10e-14 ***
## sqft_living15  1.048e-04  5.177e-06  20.243  < 2e-16 ***
## sqft_lot15    -6.561e-08  1.068e-07  -0.615  0.53882
## sale_year      6.669e-02  7.101e-03   9.391  < 2e-16 ***
## sale_month     3.066e-03  1.061e-03   2.890  0.00386 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2529 on 15105 degrees of freedom
## Multiple R-squared:  0.7687, Adjusted R-squared:  0.7684
## F-statistic:  2642 on 19 and 15105 DF,  p-value: < 2.2e-16
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma::gvlma(x = model)
##
##                       Value p-value                   Decision
## Global Stat        702.5689  0.0000 Assumptions NOT satisfied!
## Skewness             1.0503  0.3054    Assumptions acceptable.
## Kurtosis           444.4352  0.0000 Assumptions NOT satisfied!
## Link Function      256.7030  0.0000 Assumptions NOT satisfied!
## Heteroscedasticity   0.3803  0.5374    Assumptions acceptable.
```

Global stat와 link function은 linearity 가정이 충족되었는지를 보여주며, 그렇지 않다면(x에대한) data transformation을 하거나 회귀처럼 선형모델이 아닌 비선형 모델을 사용하는 방법이 있다.

Skewness와 Kurtosis는 normality 가정이 충족되었는지를 보여주며, 그렇지 않다면 Y에 대한 data transformation을 해야 할 수 있다.

Heteroscedasticity는 constant variance 가정이 충족되었는지를 보여준다.

우리는 Heteroscedasticity(이분산성)가정과 Skewness(왜도)가 충족되지 않은것을 통해서 어느 가정이 틀렸는지 확인할 수 있다. but gvlma를 이용하면 간편하기는 하지만, statistical testing 기법이 갖는 한계점처럼 유의수준 0.05에서 [가정 충족 || 가정 충족하지 않음]의 경계를 잘라 버리다 보니 융통성이 부족하다는 점이 있다. 선형회귀는 이런 가정 충족에 대해서 비교적 robust 한 편이다 보니 이 결과만 보고 비선형적 모델로 바로 넘어가는 등의 속단은 위험할 수 있다고 생각한다.

참고: 찌니 https://m.blog.naver.com/meunique/221160090068

```
#normality assumption
#shapiro.test(model$residuals)
```

```
#constant variance assumption
#car::ncvTest(model)
```

```
#independent errors assumption
#lmtest::dwtest(model)
```

```
#선형 가정
#car::ceresPlots(model)
```

```
pred <- predict(model, test_1)
pred <- exp(pred)
pred <- ifelse(pred < 0, 0, pred)
```

```
rmsle <- function(pred, act){
  if(sum(pred < 0) > 0)
    stop("예측값에 0보다 작은 값이 존재합니다. 해당 값을 0으로 만들어주세요.")
  if(length(pred) != length(act))
    stop("예측값과 실제값의 벡터 길이가 다릅니다. 예측값을 다시 확인해주세요.")

  len <- length(pred)
  pred <- log(pred + 1)
  act <- log(act + 1)
  msle <- mean((pred - act)^2)
  return(sqrt(msle))
}

cat("[1] Rmsle:" , rmsle(pred, test$price))
```

```
## [1] Rmsle: 0.2475622
```

```
cat("\n[2] Adjusted R-squared:  0.7684")
```

```
##
## [2] Adjusted R-squared:  0.7684
```

Skewness & Heteroscedasticity 를 가정에 맞게 수정해줘야 하는데 이건 다음기회에 … 최종: [1] Rmsle:

0.2488122 [2] Adjusted R-squared: 0.7701

```
#using
#1.bathrooms  sqft_living
train_1 - train %% mutate(bathroomssqft_living = bathroomssqft_living)
test_1 - test %% mutate(bathroomssqft_living = bathroomssqft_living)

model - lm(price~.,data = train_1)
summary(model)
plot(model)
pred - predict(model, test_1)
pred - ifelse(pred  0, 0, pred)
rmsle(pred, test_1$price) # R^2 = 0.736  rmsle = 0.4835
#2.sqft_living  sqft_above

train_2 - train_1 %% mutate(sqft_abovesqft_living = sqft_abovesqft_living)
test_2 - test_1 %% mutate(sqft_abovesqft_living = sqft_abovesqft_living)

model - lm(price~.,data = train_2)
summary(model)
plot(model)
pred - predict(model, test_2)
pred - ifelse(pred  0, 0, pred)
rmsle(pred, test_2$price) # R^2 = 0.739  rmsle = 0.4738

#3.grade  sqft_living
train_3 - train_2 %% mutate(gradesqft_living = gradesqft_living)
test_3 - test_2 %% mutate(gradesqft_living = gradesqft_living)
model - lm(price~.,data = train_3)
summary(model)
plot(model)
pred - predict(model, test_3)
pred - ifelse(pred  0, 0, pred)
rmsle(pred, test_3$price) # R^2 = 0.744  rmsle = 0.3824


m3 - cor(train_3)
corrplot(m3,method=circle) #method에 따라서 그림이 다름. circle 치면 원형태로 나옴.

#4.factor화
train_4_1 - train
test_4_1 - test
train_4_2 - train_3
test_4_2 - test_3

# factor만 제대로 바꿔줘도 error가 0.38까지 줄어듬.
train_4_1[, c(waterfront,view,condition,sale_year,sale_month)] -
  lapply(train[, c(waterfront,view,condition,sale_year,sale_month)], as.factor)
test_4_1[, c(waterfront,view,condition,sale_year,sale_month)] -
  lapply(test[, c(waterfront,view,condition,sale_year,sale_month)], as.factor)

model - lm(price~.,data = train_4_1)
summary(model)
plot(model)
pred - predict(model, test_4_1)
pred - ifelse(pred  0, 0, pred)
```

```
rmsle(pred, test_4_1$price) # R^2 = 0.7038 , error = 0.98

# 우리가 만든 모델
train_4_2[, c(waterfront,view,condition,sale_year,sale_month)] -
  lapply(train[, c(waterfront,view,condition,sale_year,sale_month)], as.factor)
test_4_2[, c(waterfront,view,condition,sale_year,sale_month)] -
  lapply(test[, c(waterfront,view,condition,sale_year,sale_month)], as.factor)

model - lm(price~.,data = train_4_2)
summary(model)
plot(model)
pred - predict(model, test_4_2)
pred - ifelse(pred  0, 0, pred)
rmsle(pred, test_4_1$price) # R^2 = 0.746 , error = 0.408

#5.변수들의 변경.
#ㄱ.yr_renovated
yr_renovated_train - ifelse(train$yr_renovated  0.5, 0, 1)
yr_renovated_test - ifelse(test$yr_renovated  0.5, 0, 1)

train_4_2_1 - train_4_2
train_4_2_1$yr_renovated - yr_renovated_train
test_4_2_1 - test_4_2
test_4_2_1$yr_renovated - yr_renovated_test


model - lm(price~.,data = train_4_2_1)
summary(model)
plot(model)
pred - predict(model, test_4_2_1)
pred - ifelse(pred  0, 0, pred)
rmsle(pred, test_4_2_1$price) # R^2 = 0.746 , error = 0.408


#ㄴ.zipcode
zipcode_train - substr(train$zipcode,1,3)
zipcode_test - substr(test$zipcode,1,3)

train_4_2_2 - train_4_2_1
train_4_2_2$zipcode - zipcode_train
test_4_2_2 - test_4_2_1
test_4_2_2$zipcode - zipcode_test

train_4_2_2$zipcode - as.factor(train_4_2_2$zipcode)
test_4_2_2$zipcode - as.factor(test_4_2_2$zipcode)

model - lm(price~.,data = train_4_2_2)
summary(model)
plot(model)
pred - predict(model, test_4_2_2)
pred - ifelse(pred  0, 0, pred)
rmsle(pred, test_4_2_2$price) # R^2 = 0.7418 (0.746에서 감소) , error = 0.4064

#ㄷ. sqft_basement제거 후 다중공선성 확인
train_4_2_2 - train_4_2_2 %% select(-sqft_basement)
test_4_2_2 - test_4_2_2 %% select(-sqft_basement)
model - lm(price~., data = train_4_2_2)
```

```
model - lm(price~.,data = train_4_2_2)

library(car)

vif-vif(model)
vif



# 10이상인 bathrooms,sqft_living,grade,sqft_above,sale_year 먼저 제거.
train_4_2_3 - train_4_2_2 %% select(-c(bathrooms,sqft_living,grade,sqft_above,sale_
year))
test_4_2_3 - test_4_2_2 %% select(-c(bathrooms,sqft_living,grade,sqft_above,sale_ye
ar))
model - lm(price~.,data = train_4_2_3)
summary(model)
plot(model)
pred - predict(model, test_4_2_3)
pred - ifelse(pred  0, 0, pred)
rmsle(pred, test_4_2_3$price) # R^2 = 0.7234 (0.7418에서 감소) , error = 0.38
vif-vif(model)
vif #많이 깔끔해짐.

studentized - rstudent(model)
table(abs(studentized)3)
outliers - which(abs(studentized)3)
refine_train - train_4_2_3[-outliers, ]

model - lm(price~.,data = refine_train)
summary(model)
plot(model)
pred - predict(model, test_4_2_3)
pred - ifelse(pred  0, 0, pred)
rmsle(pred, test_4_2_3$price) # R^2 = 0.7606 , error = 0.349

# 10이상인 gradesqft_living 제거.
train_4_2_4 - train_4_2_3[,-18]
test_4_2_4 - test_4_2_3[-18]
model - lm(price~.,data = train_4_2_4)
summary(model)
plot(model)
pred - predict(model, test_4_2_4)
pred - ifelse(pred  0, 0, pred)
rmsle(pred, test_4_2_4$price) # R^2 = 0.7042 (0.7418에서 감소) , error = 0.48
vif-vif(model)
vif #많이 깔끔해짐.

# 5. 이상치 제거
studentized - rstudent(model)
table(abs(studentized)3)
outliers - which(abs(studentized)3)
refine_train - train_4_2_4[-outliers, ]

model - lm(price~.,data = refine_train)
summary(model)
plot(model)
pred - predict(model, test_4_2_4)
```

```
pred - ifelse(pred  0, 0, pred)
rmsle(pred, test_4_2_4$price) # R^2 = 0.7286 (0.7042에서 증가) , error = 0.3
```