House Price(P#1)

Hyunpyo Kim

Introduction

This is the price-model of expecting houses' prices. The features to expect prices are age of house, land value, living area size, number of rooms and so on.

```
library(tidyverse)
library(mosaic)
library(foreach)
library(FNN)
library(knitr)
library(kableExtra)
```

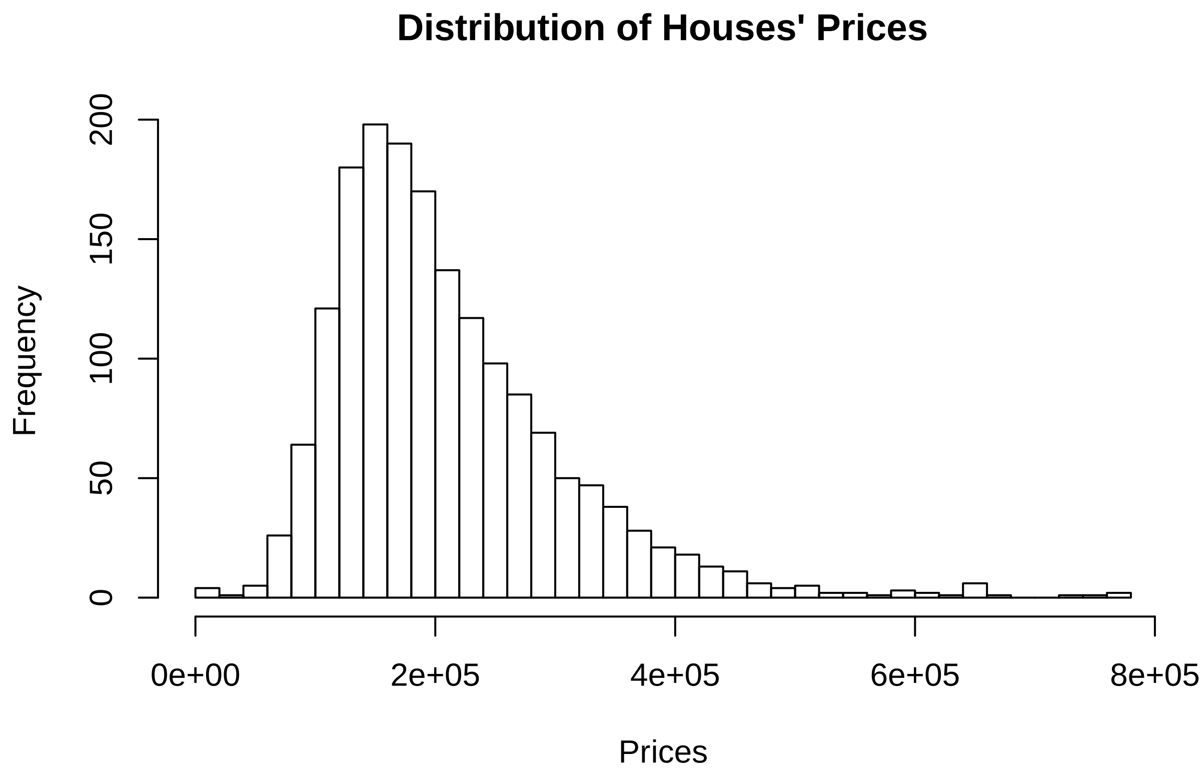
```
data(SaratogaHouses)
head(SaratogaHouses)
```

```
##    price lotSize age landValue livingArea pctCollege bedrooms fireplaces
## 1 132500    0.09  42     50000        906         35        2          1
## 2 181115    0.92   0     22300       1953         51        3          0
## 3 109000    0.19 133      7300       1944         51        4          1
## 4 155000    0.41  13     18700       1944         51        3          1
## 5  86060    0.11   0     15000        840         51        2          0
## 6 120000    0.68  31     14000       1152         22        4          1
##   bathrooms rooms          heating     fuel              sewer waterfront
## 1       1.0     5         electric electric             septic         No
## 2       2.5     6 hot water/steam      gas             septic         No
## 3       1.0     8 hot water/steam      gas public/commercial         No
## 4       1.5     5          hot air      gas             septic         No
## 5       1.0     3          hot air      gas public/commercial         No
## 6       1.0     8          hot air      gas             septic         No
##   newConstruction centralAir
## 1              No         No
## 2              No         No
## 3              No         No
## 4              No         No
## 5             Yes        Yes
## 6              No         No
```

Data analysis

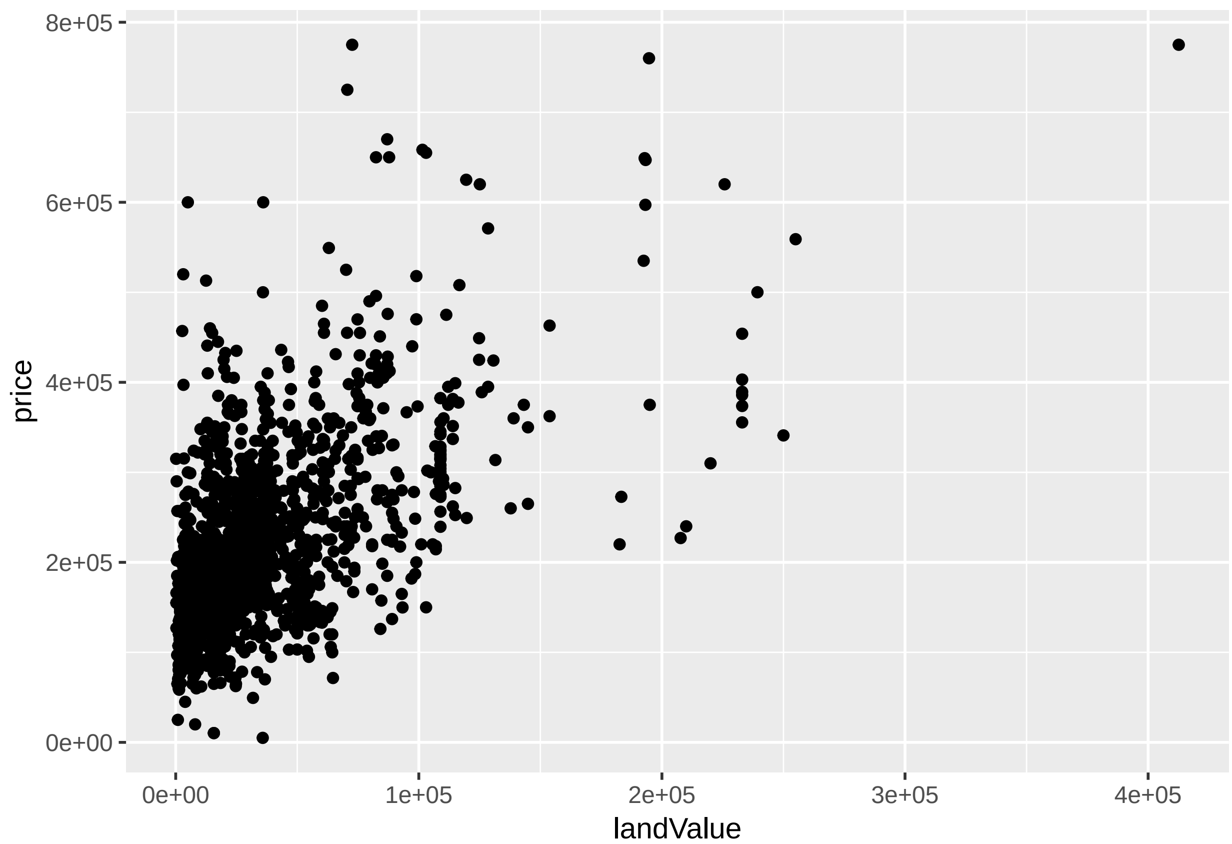
The prices of houses used for modelling are well distributed, though it is right tailed.

```
hist(SaratogaHouses$price, breaks = 50, main = "Distribution of Houses' Prices", xlab = "Prices")
```

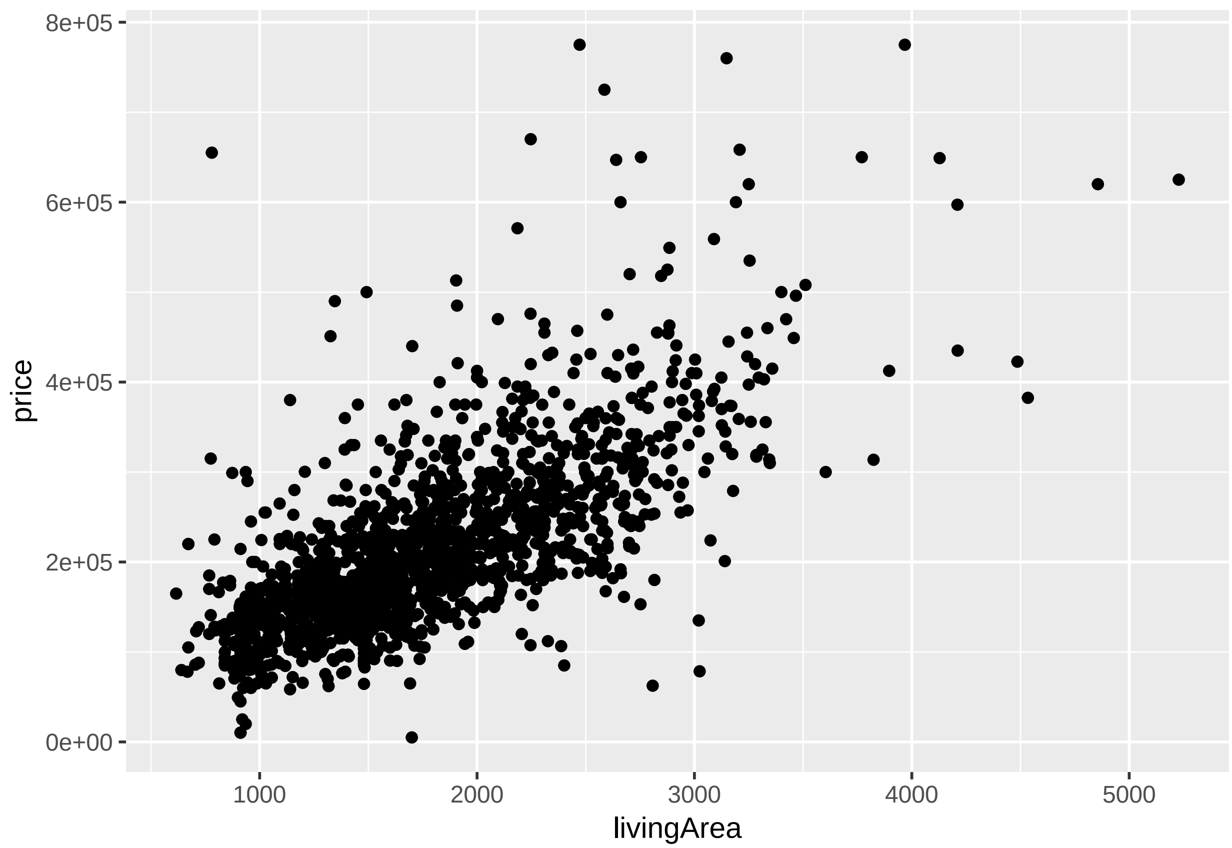


Some feautures can explain the prices well, such as landvalue, living area and number of rooms. However, some(pctCollege) do not explain well.

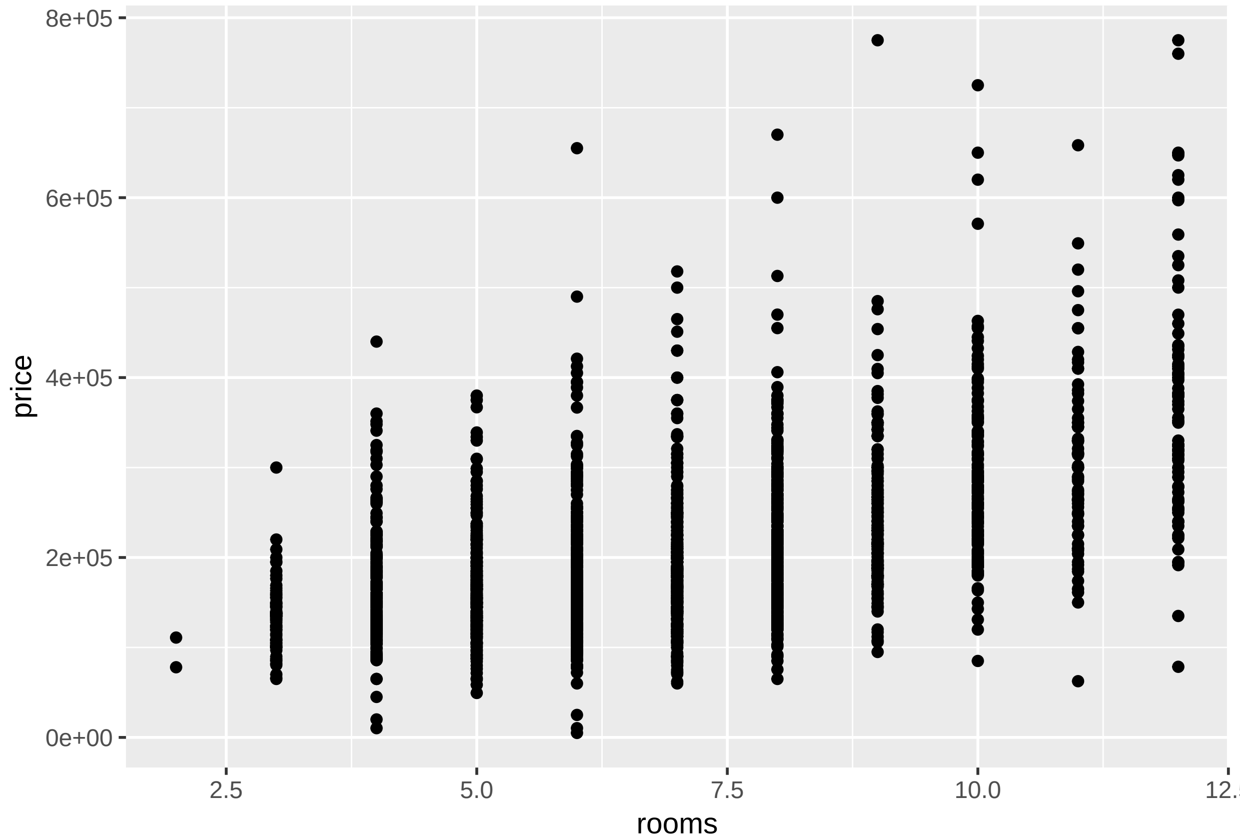
```
ggplot(SaratogaHouses) + geom_point(aes(landValue, price))
```



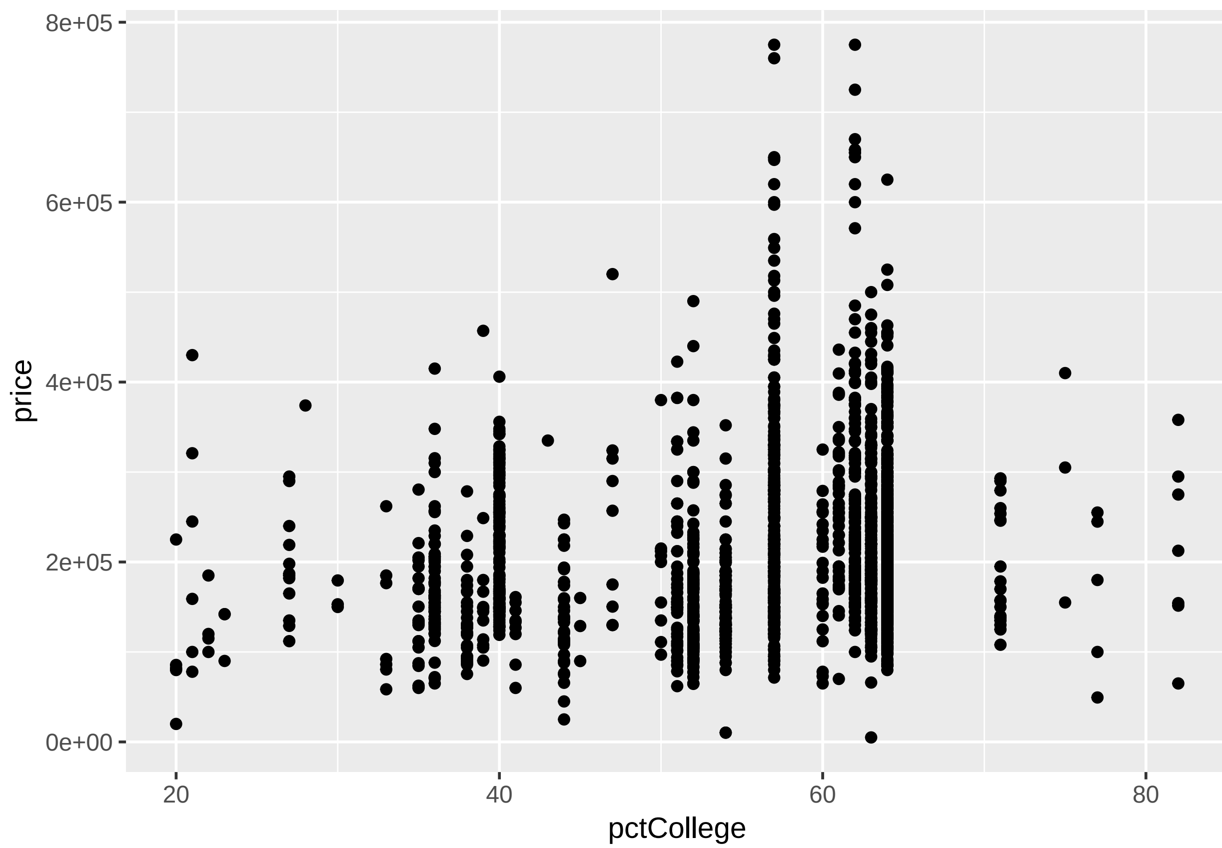

```
ggplot(SaratogaHouses) + geom_point(aes(livingArea, price))
```




```r
ggplot(SaratogaHouses) + geom_point(aes(rooms, price))
```




```r
ggplot(SaratogaHouses) + geom_point(aes(pctCollege, price))
```



Modelling process

The model is trained by 1,382 house-price data(80% of total data), and is verified other 346 house-price data(20% of total data).

```
n = nrow(SaratogaHouses)
n_train = round(0.8*n)
n_test = n - n_train
```

To verify the model, root-mean-square error is used.

```
rmse = function(y, yhat) {
  sqrt( mean( (y - yhat)^2 ) )
}
```

The trained data set and tested data set are sampled randomly. The difference by random sample is moderated by averaging 100 times repeated sampling.

Linear Model

There are two models I benchmark, one is medium model, the other is biggerboom model. First, I skip some features which look unpredictable. Those are age, pctCollege, fireplaces. Then, repeat hand-building a model by changing features and interactions.

```r
rmse_vals = do(100) * {
  ### split train set and test set
  train_cases = sample.int(n, n_train, replace=FALSE)
  test_cases = setdiff(1:n, train_cases)
  saratoga_train = SaratogaHouses[train_cases,]
  saratoga_test = SaratogaHouses[test_cases,]

  ### fitting
  lm_M = lm(price ~ . - sewer - waterfront - landValue - newConstruction, data=saratoga_train)
  lm_B = lm(price ~ lotSize + landValue + waterfront + newConstruction +
              bedrooms*bathrooms + heating + fuel + pctCollege + rooms*bedrooms +
              rooms*bathrooms + rooms*heating + livingArea, data=saratoga_train)
  lm_1 = lm(price ~ lotSize + livingArea * landValue + bedrooms * rooms + bathrooms * rooms +
              heating + waterfront + newConstruction + centralAir,
            data = saratoga_train)

  ### predict
  yhat_M = predict(lm_M, saratoga_test)
  yhat_B = predict(lm_B, saratoga_test)
  yhat_1 = predict(lm_1, saratoga_test)

  ### evaluation
  c(rmse(saratoga_test$price, yhat_M), rmse(saratoga_test$price, yhat_B),
    rmse(saratoga_test$price, yhat_1))
}
```

My final model is

```r
lm_1
```

```
##
## Call:
## lm(formula = price ~ lotSize + livingArea * landValue + bedrooms *
##     rooms + bathrooms * rooms + heating + waterfront + newConstruction +
##     centralAir, data = saratoga_train)
##
## Coefficients:
##            (Intercept)                 lotSize              livingArea
##              9.748e+04               8.382e+03               7.438e+01
##              landValue                bedrooms                   rooms
##              1.121e+00               8.013e+03               1.344e+03
##              bathrooms  heatinghot water/steam         heatingelectric
##             -4.526e+03              -1.302e+04              -6.295e+03
##           waterfrontNo       newConstructionNo            centralAirNo
##             -1.174e+05               4.228e+04              -1.224e+04
##   livingArea:landValue          bedrooms:rooms         rooms:bathrooms
##             -1.090e-04              -1.919e+03               3.918e+03
```

- This model is a little bit better that biggerboom model.

```r
rmse_lpm = data.frame(mediim_Model_BM1 = colMeans(rmse_vals[1]),
                      biggerboom_Model_BM2 = colMeans(rmse_vals[2]),
```

```
                      New_Model = colMeans(rmse_vals[3]))
kable(rmse_lpm) %>% kable_styling("striped")
```

	mediim_Model_BM1	biggerboom_Model_BM2	New_Model
V1	65767.44	57900.53	57769.51

KNN Model

I make KNN Model by using my linear model

First, make feature data sets and result data sets. Make interaction features what I used for my linear model. I can use only numeric values for KNN, transform factor variables to numerical dummies.

```
KNN_X = model.matrix(~lotSize + livingArea * landValue + bedrooms * rooms +
                       bathrooms * rooms + heating + waterfront + newConstruction +
                       centralAir - 1, data = SaratogaHouses)
KNN_y = SaratogaHouses$price
```

We do not know which K will make the best result, so I try some sequence numbers. When trying whole testing numbers, the least rmse occur when K is less than 30. > k_grid = exp(seq(log(2), log(300), length=30)) %>% round %>% unique

```
k_grid = seq(1, 30, by=1)

err_grid = foreach(k = k_grid, .combine='c') %do% {
  out = do(100) * {

    #### split
    train_cases = sample.int(n, n_train, replace=FALSE)
    test_cases = setdiff(1:n, train_cases)
    X_train = KNN_X[train_cases,]
    X_test = KNN_X[test_cases,]
    y_train = KNN_y[train_cases]
    y_test = KNN_y[test_cases]

    #### scale the training set features
    scale_factors = apply(X_train, 2, sd)
    X_train_sc = scale(X_train, scale=scale_factors)
    X_test_sc = scale(X_test, scale=scale_factors)

    # fitting
    knn_try = knn.reg(X_train_sc, X_test_sc, y_train, k=k)

    # errors
    rmse(y_test, knn_try$pred)
  }
  mean(out$result)
}
```

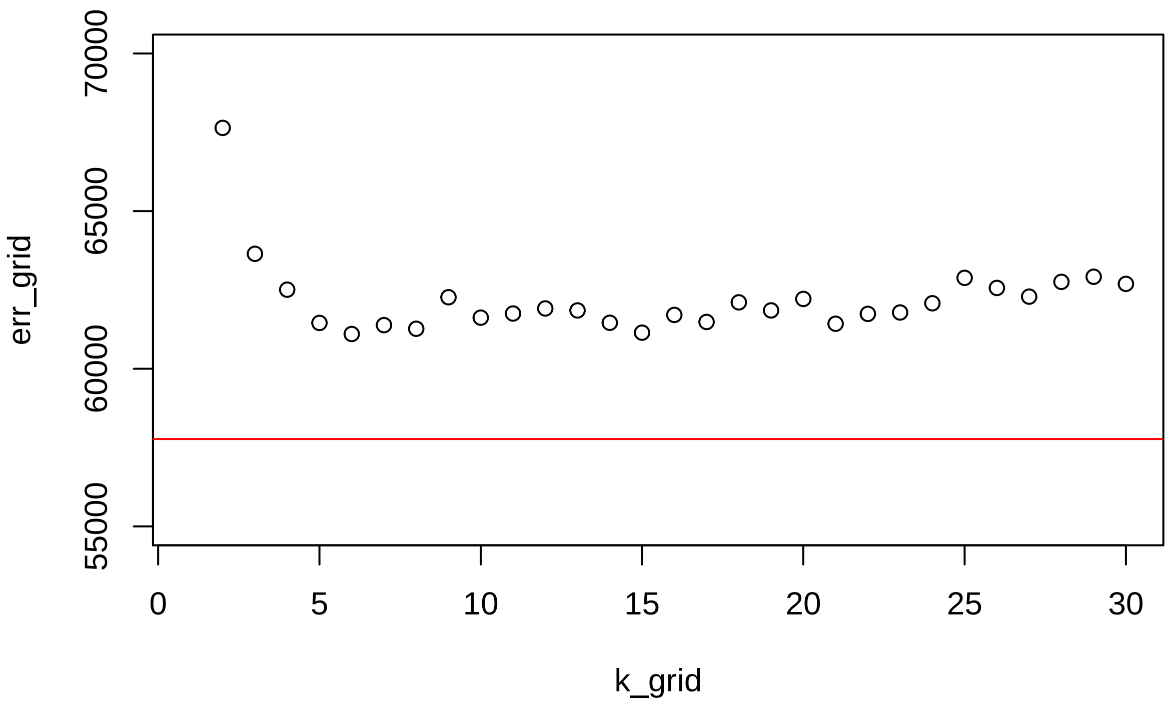
The result is following.

```r
rmse_KNN = data.frame(K_value = c(which(err_grid == min(err_grid))),
                      KNN_RMSE = c(min(err_grid)),
                      LPM_RMSE = colMeans(rmse_vals[3]))
kable(rmse_KNN) %>% kable_styling("striped")
```

	K_value	KNN_RMSE	LPM_RMSE
V3	6	61101.71	57769.51

However, KNN model is not better than linear model. The rmses of FNN are always bigger than the linear model's.

```r
plot(k_grid, err_grid, ylim = c(55000, 70000))
abline(h=colMeans(rmse_vals[3]), col='red')
```



Thus, the best model for expecting houses prices is

```r
lm_1
```

```
##
## Call:
## lm(formula = price ~ lotSize + livingArea * landValue + bedrooms *
##     rooms + bathrooms * rooms + heating + waterfront + newConstruction +
##     centralAir, data = saratoga_train)
```

```
##
## Coefficients:
##          (Intercept)                lotSize             livingArea
##            9.748e+04              8.382e+03              7.438e+01
##            landValue               bedrooms                  rooms
##            1.121e+00              8.013e+03              1.344e+03
##            bathrooms  heatinghot water/steam        heatingelectric
##           -4.526e+03             -1.302e+04             -6.295e+03
##         waterfrontNo     newConstructionNo           centralAirNo
##           -1.174e+05              4.228e+04             -1.224e+04
##   livingArea:landValue         bedrooms:rooms        rooms:bathrooms
##           -1.090e-04             -1.919e+03              3.918e+03
```