Housing Price Analysis using Linear Regression

Bangda Sun

May 14, 2017

Based on the explortary data analysis, we can start from those predictors:

bedrooms, bathrooms, sqftliving, floors, waterfront, view, grade.

So we will start from them.

1. Simple Regression

Our first model is a very basic and simple model, just use one predictor:

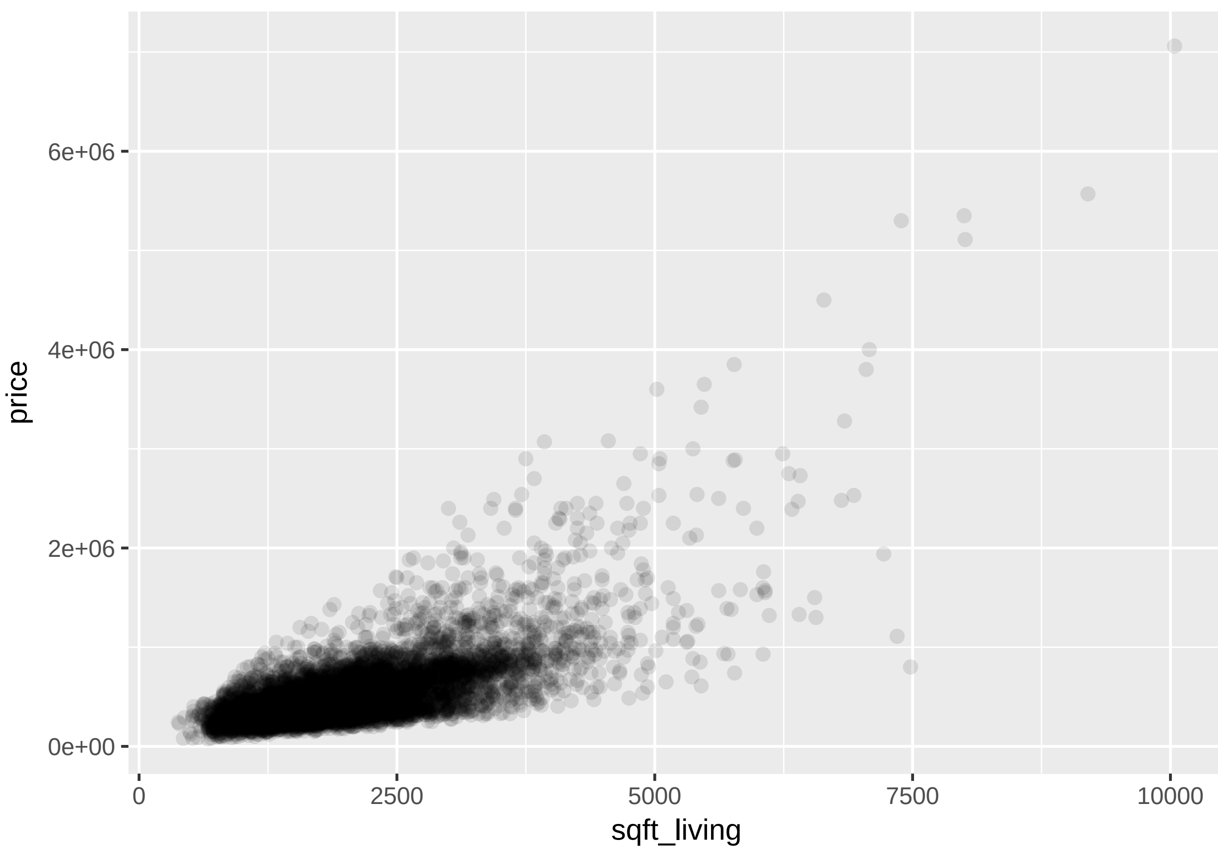
$$\text{price} = \beta_0 + \beta_1 \text{sqft living} + \epsilon.$$

we can plot their relationship

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
ggplot(training) +
  geom_point(mapping = aes(x = sqft_living, y = price), alpha = .1, size = 2)
```



We could say that there exists linear relationship, however in higher level of sqft_living, the variance of observations is higher, they become more sparse. Therefore we can imply that only one predictor sqft_living is not enough.

```r
simple_model1 = lm(price ~ sqft_living, data = training)
summary(simple_model1)
```

```
##
## Call:
## lm(formula = price ~ sqft_living, data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1296881  -143650   -23861   105877  4225597
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -58066.259   7861.478  -7.386 1.68e-13 ***
## sqft_living    288.095      3.482  82.732  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 267200 on 7086 degrees of freedom
## Multiple R-squared:  0.4913, Adjusted R-squared:  0.4913
## F-statistic:  6845 on 1 and 7086 DF,  p-value: < 2.2e-16
```

```r
# Logarithm transform
simple_model2 = lm(I(log(price)) ~ I(log(sqft_living)), data = training)
summary(simple_model2)
```

```
##
## Call:
## lm(formula = I(log(price)) ~ I(log(sqft_living)), data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0833 -0.2959  0.0143  0.2562  1.3171
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.65296    0.08347   79.70   <2e-16 ***
## I(log(sqft_living))  0.84649    0.01105   76.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3928 on 7086 degrees of freedom
## Multiple R-squared:  0.4531, Adjusted R-squared:  0.4531
## F-statistic:  5871 on 1 and 7086 DF,  p-value: < 2.2e-16
```

We tried logarithm transfrom both on response and predictors. And we evaluate them by apply them on testing set:

```r
predOfModel1 = predict(simple_model1, newdata = testing)
rmse1 = sqrt(mean((predOfModel1 - testing$price)^2))
predOfModel2 = predict(simple_model2, newdata = testing)
rmse2 = sqrt(mean((exp(predOfModel2) - testing$price)^2))
```

```r
rmseVec = c(rmse1, rmse2)
rmseVec
```

```
## [1] 261612 277610
```

where RMSE is calculated by

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}.$$

Next we want find the best predictor that will minimize RMSE, we will plug every predictor into the linear regression model,

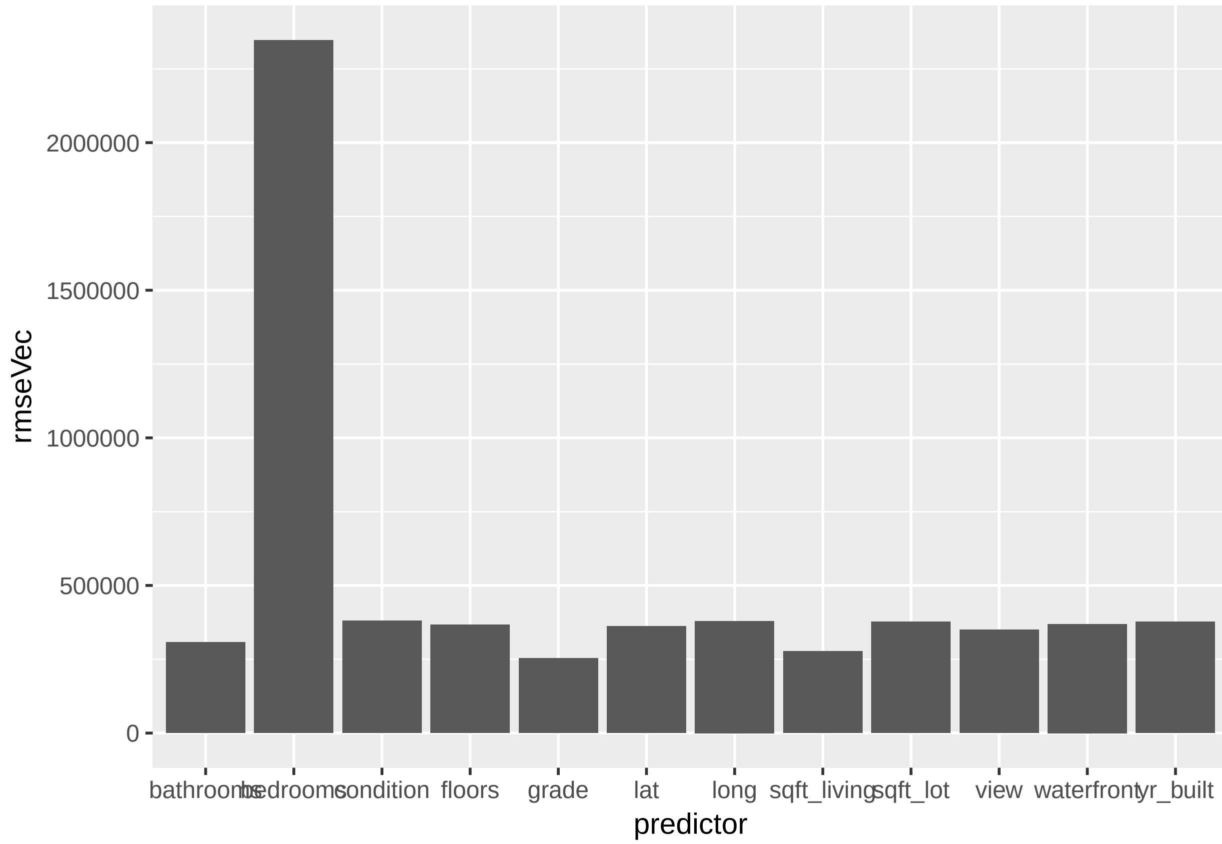
```r
findMinRMSE = function(predictor) {
  # @param predictor
  # @return RMSE of linear regression model
  if (!predictor%in%c("sqft_living", "sqft_lot")) {
    reg   = formula(paste("I(log(price)) ~ ", predictor))
    model = lm(reg, data = training)
    pred  = predict(model, newdata = testing)
    rmse  = sqrt(mean((testing$price - exp(pred))^2))
  }
  else {
    reg   = formula(paste("I(log(price)) ~ ", paste("I(log(", predictor, "))", sep = "")))
    model = lm(reg, data = training)
    pred  = predict(model, newdata = testing)
    rmse  = sqrt(mean((testing$price - exp(pred))^2))
  }
  return(rmse)
}
# Check the function
findMinRMSE(predictor = "sqft_living")
```

```
## [1] 277610
```

```r
# Find the best predictor
predictorName = c("bedrooms", "bathrooms", "sqft_living",
                  "sqft_lot", "floors"    , "waterfront",
                  "view"    , "condition", "grade",
                  "yr_built", "lat"       , "long")
predictorMatrix = matrix(predictorName, nrow = length(predictorName))
rmseVec = apply(predictorMatrix, MARGIN = 1, findMinRMSE)
rmseVec
```

```
##  [1] 2346915.7  307131.1  277610.0  377407.6  366566.5  369560.4  349681.6
##  [8]  380544.8  253440.0  377592.0  361670.1  379686.0
```

```r
rmsedf = data.frame(predictor = predictorName,
                    rmse = rmseVec)
ggplot(rmsedf, mapping = aes(x = predictor, y = rmseVec)) +
  geom_bar(stat = "identity")
```



We can see that use grade will product the smallest RMSE.

2. Multiple Regression

Then we increase the number of predictors, we add bedrooms, bathrooms, grade and waterfront,

```
multiple_model1 = lm(I(log(price)) ~ I(log(sqft_living)) +
                       bedrooms + bathrooms + grade + waterfront,
                     data = training)
summary(multiple_model1)
```

```
##
## Call:
## lm(formula = I(log(price)) ~ I(log(sqft_living)) + bedrooms +
##     bathrooms + grade + waterfront, data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08289 -0.25422 -0.00157  0.23628  1.31014
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          8.048815   0.116598  69.030  < 2e-16 ***
## I(log(sqft_living))  0.483182   0.020175  23.949  < 2e-16 ***
## bedrooms            -0.023574   0.006158  -3.828  0.00013 ***
## bathrooms           -0.008815   0.009070  -0.972  0.33117
```

```
## grade                 0.188826   0.005662  33.349  < 2e-16 ***
## waterfront            0.742457   0.047151  15.747  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3551 on 7082 degrees of freedom
## Multiple R-squared:  0.5534, Adjusted R-squared:  0.5531
## F-statistic:  1755 on 5 and 7082 DF,  p-value: < 2.2e-16
```

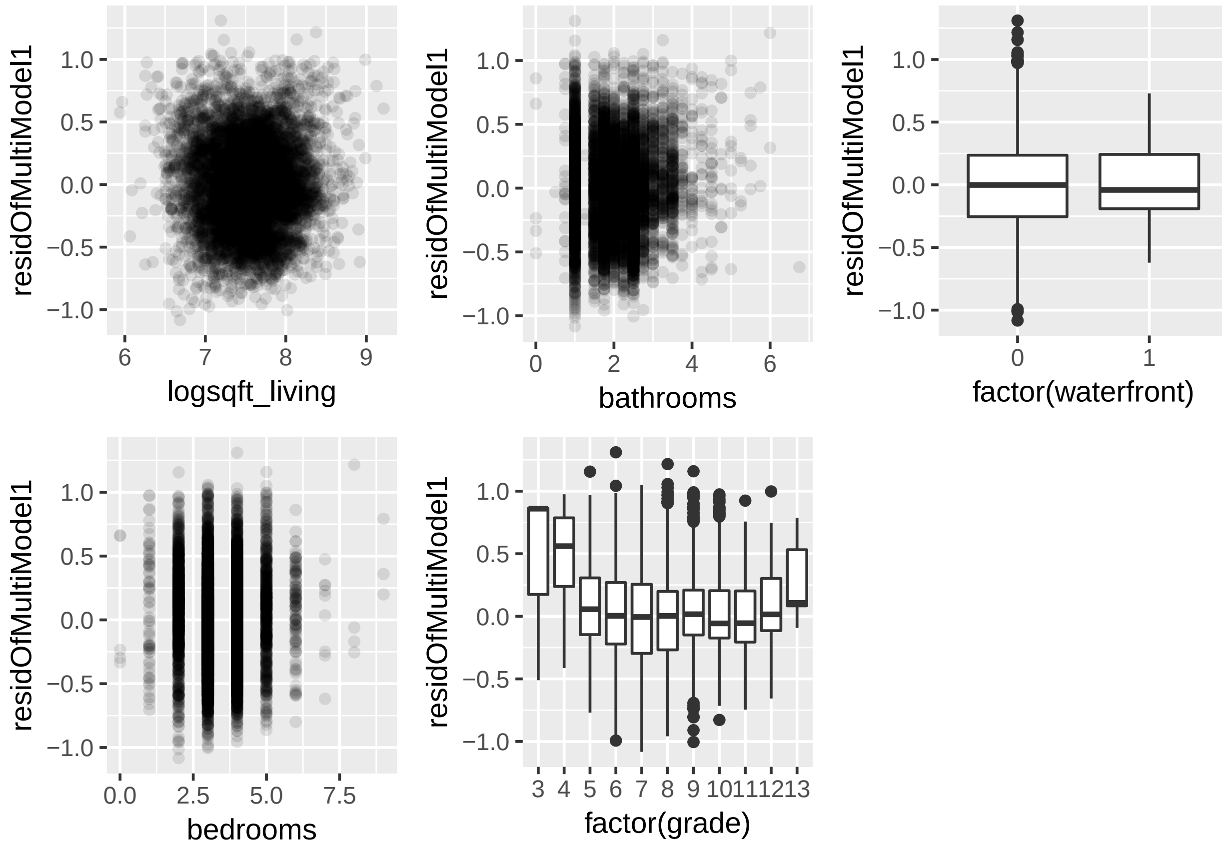
The R^2 has increased, then we evaluate the RMSE of this model,

```r
predOfModel3 = predict(multiple_model1, newdata = testing)
rmse3 = sqrt(mean((testing$price - exp(predOfModel3))^2))
rmse3
```

```
## [1] 233112.2
```

Next we will diagnois the model from the view of residual.

```r
residOfMultiModel1 = residuals(multiple_model1)
residdf = data.frame(residOfMultiModel1 = residOfMultiModel1,
                     logsqft_living     = log(training[, "sqft_living"]),
                     bedrooms           = training[, "bedrooms"],
                     bathrooms          = training[, "bathrooms"],
                     grade              = training[, "grade"],
                     waterfront         = training[, "waterfront"])
p1 = ggplot(residdf) +
  geom_point(mapping = aes(x = logsqft_living, y = residOfMultiModel1), alpha = .1)
p2 = ggplot(residdf) +
  geom_point(mapping = aes(x = bedrooms, y = residOfMultiModel1), alpha = .1)
p3 = ggplot(residdf) +
  geom_point(mapping = aes(x = bathrooms, y = residOfMultiModel1), alpha = .1)
p4 = ggplot(residdf, aes(factor(grade), residOfMultiModel1)) + geom_boxplot()
p5 = ggplot(residdf, aes(factor(waterfront), residOfMultiModel1)) + geom_boxplot()
multiplot(p1, p2, p3, p4, p5, cols = 3)
```



Then we add yr_built and location (lat and long) into the model,

```
multiple_model2 = lm(I(log(price)) ~ I(log(sqft_living)) +
                       bedrooms + bathrooms + grade + waterfront + yr_built + lat + long,
                     data = training)
summary(multiple_model2)
```

```
##
## Call:
## lm(formula = I(log(price)) ~ I(log(sqft_living)) + bedrooms +
##     bathrooms + grade + waterfront + yr_built + lat + long, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3009 -0.1653 -0.0062  0.1590  1.4692
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -3.559e+01  3.386e+00 -10.510  < 2e-16 ***
## I(log(sqft_living))  4.250e-01  1.522e-02  27.918  < 2e-16 ***
## bedrooms            -3.189e-02  4.623e-03  -6.899  5.7e-12 ***
## bathrooms            9.035e-02  7.156e-03  12.626  < 2e-16 ***
## grade                1.957e-01  4.373e-03  44.754  < 2e-16 ***
## waterfront           6.937e-01  3.540e-02  19.593  < 2e-16 ***
## yr_built            -4.912e-03  1.475e-04 -33.312  < 2e-16 ***
## lat                  1.313e+00  2.346e-02  55.969  < 2e-16 ***
## long                 7.314e-02  2.538e-02   2.882  0.00396 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2652 on 7079 degrees of freedom
## Multiple R-squared:  0.751,  Adjusted R-squared:  0.7507
## F-statistic:  2669 on 8 and 7079 DF,  p-value: < 2.2e-16
```

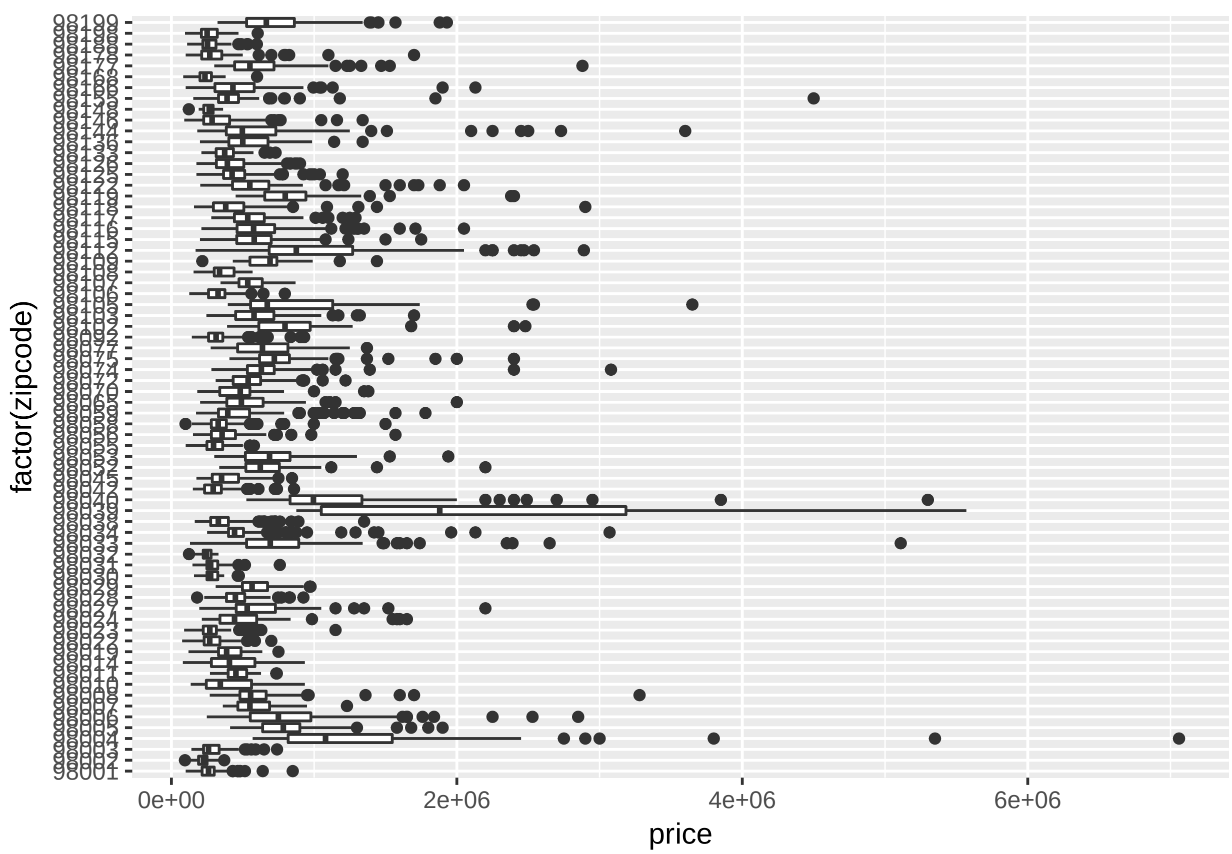
Again we calculate the RMSE of this model,

```
predOfModel4 = predict(multiple_model2, newdata = testing)
rmse4 = sqrt(mean((testing$price - exp(predOfModel4))^2))
rmse4
```

```
## [1] 199695.5
```

From the exploratory data analysis, we find that the zipcode of each house will also affect the price.

```
ggplot(training, aes(factor(zipcode), price)) + geom_boxplot() + coord_flip()
```




```
multiple_model3 = lm(I(log(price)) ~ I(log(sqft_living)) +
                       bedrooms + bathrooms + grade + waterfront + yr_built + lat + long + factor(zipcod
                     data = training)
summary(multiple_model3)
```

```
##
## Call:
## lm(formula = I(log(price)) ~ I(log(sqft_living)) + bedrooms +
##     bathrooms + grade + waterfront + yr_built + lat + long +
##     factor(zipcode), data = training)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24304 -0.10850 -0.00228  0.10812  1.11866
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -1.925e+01  1.296e+01  -1.485 0.137494
## I(log(sqft_living))   4.745e-01  1.158e-02  40.969  < 2e-16 ***
## bedrooms             -1.983e-02  3.520e-03  -5.633 1.84e-08 ***
## bathrooms             4.036e-02  5.484e-03   7.361 2.04e-13 ***
## grade                 1.301e-01  3.495e-03  37.214  < 2e-16 ***
## waterfront            7.416e-01  2.746e-02  27.007  < 2e-16 ***
## yr_built             -2.014e-03  1.267e-04 -15.901  < 2e-16 ***
## lat                   5.421e-01  1.374e-01   3.944 8.09e-05 ***
## long                 -4.564e-02  9.574e-02  -0.477 0.633596
## factor(zipcode)98002 -6.786e-02  3.087e-02  -2.198 0.027966 *
## factor(zipcode)98003 -1.623e-02  2.827e-02  -0.574 0.565857
## factor(zipcode)98004  9.630e-01  5.026e-02  19.161  < 2e-16 ***
## factor(zipcode)98005  5.406e-01  5.289e-02  10.220  < 2e-16 ***
## factor(zipcode)98006  5.032e-01  4.380e-02  11.487  < 2e-16 ***
## factor(zipcode)98007  4.557e-01  5.585e-02   8.160 3.94e-16 ***
## factor(zipcode)98008  4.588e-01  5.283e-02   8.685  < 2e-16 ***
## factor(zipcode)98010  3.172e-01  4.588e-02   6.914 5.11e-12 ***
## factor(zipcode)98011  1.660e-01  6.863e-02   2.419 0.015593 *
## factor(zipcode)98014  1.634e-01  7.568e-02   2.159 0.030894 *
## factor(zipcode)98019  1.128e-01  7.468e-02   1.511 0.130884
## factor(zipcode)98022  1.382e-01  4.162e-02   3.320 0.000905 ***
## factor(zipcode)98023 -7.166e-02  2.553e-02  -2.807 0.005020 **
## factor(zipcode)98024  3.532e-01  6.235e-02   5.665 1.53e-08 ***
## factor(zipcode)98027  3.949e-01  4.553e-02   8.674  < 2e-16 ***
## factor(zipcode)98028  1.443e-01  6.695e-02   2.155 0.031161 *
## factor(zipcode)98029  4.201e-01  5.153e-02   8.152 4.20e-16 ***
## factor(zipcode)98030 -1.187e-02  3.008e-02  -0.395 0.693149
## factor(zipcode)98031 -1.254e-02  3.153e-02  -0.398 0.690900
## factor(zipcode)98032 -1.073e-01  3.734e-02  -2.874 0.004065 **
## factor(zipcode)98033  5.701e-01  5.683e-02  10.031  < 2e-16 ***
## factor(zipcode)98034  2.819e-01  6.131e-02   4.598 4.34e-06 ***
## factor(zipcode)98038  1.455e-01  3.427e-02   4.245 2.22e-05 ***
## factor(zipcode)98039  1.170e+00  7.594e-02  15.402  < 2e-16 ***
## factor(zipcode)98040  7.540e-01  4.423e-02  17.048  < 2e-16 ***
## factor(zipcode)98042  4.323e-02  2.949e-02   1.466 0.142653
## factor(zipcode)98045  2.569e-01  6.270e-02   4.096 4.24e-05 ***
## factor(zipcode)98052  3.987e-01  5.861e-02   6.802 1.11e-11 ***
## factor(zipcode)98053  4.112e-01  6.244e-02   6.586 4.84e-11 ***
## factor(zipcode)98055  2.700e-02  3.540e-02   0.763 0.445649
## factor(zipcode)98056  1.909e-01  3.859e-02   4.947 7.73e-07 ***
## factor(zipcode)98058  7.390e-02  3.291e-02   2.245 0.024778 *
## factor(zipcode)98059  2.508e-01  3.867e-02   6.487 9.36e-11 ***
## factor(zipcode)98065  3.332e-01  5.933e-02   5.616 2.03e-08 ***
## factor(zipcode)98070  2.385e-01  4.324e-02   5.515 3.61e-08 ***
## factor(zipcode)98072  2.427e-01  6.804e-02   3.567 0.000364 ***
## factor(zipcode)98074  3.417e-01  5.542e-02   6.164 7.47e-10 ***
## factor(zipcode)98075  4.078e-01  5.289e-02   7.711 1.42e-14 ***
```

```
## factor(zipcode)98077  2.339e-01  7.057e-02   3.314 0.000923 ***
## factor(zipcode)98092  3.113e-02  2.795e-02   1.114 0.265464
## factor(zipcode)98102  6.179e-01  5.977e-02  10.337  < 2e-16 ***
## factor(zipcode)98103  5.325e-01  5.518e-02   9.650  < 2e-16 ***
## factor(zipcode)98105  6.663e-01  5.681e-02  11.728  < 2e-16 ***
## factor(zipcode)98106  1.578e-01  4.180e-02   3.776 0.000161 ***
## factor(zipcode)98107  5.326e-01  5.808e-02   9.170  < 2e-16 ***
## factor(zipcode)98108  1.814e-01  4.510e-02   4.021 5.86e-05 ***
## factor(zipcode)98109  6.206e-01  6.006e-02  10.333  < 2e-16 ***
## factor(zipcode)98112  7.109e-01  5.186e-02  13.709  < 2e-16 ***
## factor(zipcode)98115  5.106e-01  5.612e-02   9.099  < 2e-16 ***
## factor(zipcode)98116  5.753e-01  4.545e-02  12.658  < 2e-16 ***
## factor(zipcode)98117  5.047e-01  5.677e-02   8.889  < 2e-16 ***
## factor(zipcode)98118  2.741e-01  3.993e-02   6.865 7.21e-12 ***
## factor(zipcode)98119  6.903e-01  5.600e-02  12.326  < 2e-16 ***
## factor(zipcode)98122  4.657e-01  4.941e-02   9.424  < 2e-16 ***
## factor(zipcode)98125  2.913e-01  6.088e-02   4.785 1.75e-06 ***
## factor(zipcode)98126  3.715e-01  4.194e-02   8.860  < 2e-16 ***
## factor(zipcode)98133  1.584e-01  6.300e-02   2.514 0.011964 *
## factor(zipcode)98136  4.822e-01  4.324e-02  11.150  < 2e-16 ***
## factor(zipcode)98144  4.615e-01  4.628e-02   9.973  < 2e-16 ***
## factor(zipcode)98146  9.246e-02  3.882e-02   2.382 0.017253 *
## factor(zipcode)98148 -3.934e-02  5.207e-02  -0.756 0.449963
## factor(zipcode)98155  1.451e-01  6.511e-02   2.229 0.025854 *
## factor(zipcode)98166  2.293e-01  3.395e-02   6.753 1.56e-11 ***
## factor(zipcode)98168 -5.539e-02  3.715e-02  -1.491 0.136040
## factor(zipcode)98177  3.236e-01  6.563e-02   4.931 8.39e-07 ***
## factor(zipcode)98178 -3.243e-03  3.857e-02  -0.084 0.932978
## factor(zipcode)98188 -1.677e-02  3.725e-02  -0.450 0.652525
## factor(zipcode)98198 -1.817e-02  2.908e-02  -0.625 0.532056
## factor(zipcode)98199  5.632e-01  5.424e-02  10.383  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1976 on 7010 degrees of freedom
## Multiple R-squared:  0.8631, Adjusted R-squared:  0.8616
## F-statistic:   574 on 77 and 7010 DF,  p-value: < 2.2e-16
```

and calculate RMSE of this model

```r
predOfModel5 = predict(multiple_model3, newdata = testing)
rmse5 = sqrt(mean((testing$price - exp(predOfModel5))^2))
rmse5
```

```
## [1] 144676.2
```

Still we can improve our model, we can add the interaction terms of the model

```r
multiple_model4 = lm(I(log(price)) ~ I(log(sqft_living)) +
                       bedrooms * bathrooms * grade + floors + waterfront + yr_built +
                       lat + long + factor(view) + factor(zipcode) + factor(condition),
                     data = training)
summary(multiple_model4)
```

```
##
## Call:
## lm(formula = I(log(price)) ~ I(log(sqft_living)) + bedrooms *
```

```
##     bathrooms * grade + floors + waterfront + yr_built + lat +
##     long + factor(view) + factor(zipcode) + factor(condition),
##     data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2519 -0.1025  0.0018  0.1040  1.0393
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -2.171e+01  1.238e+01  -1.753 0.079576 .
## I(log(sqft_living))       4.613e-01  1.137e-02  40.580  < 2e-16 ***
## bedrooms                 -1.752e-01  3.085e-02  -5.679 1.41e-08 ***
## bathrooms                -2.997e-01  5.262e-02  -5.697 1.27e-08 ***
## grade                     1.136e-02  1.479e-02   0.768 0.442425
## floors                    4.003e-03  5.742e-03   0.697 0.485744
## waterfront                4.769e-01  3.323e-02  14.349  < 2e-16 ***
## yr_built                 -1.187e-03  1.339e-04  -8.869  < 2e-16 ***
## lat                       5.296e-01  1.314e-01   4.030 5.63e-05 ***
## long                     -6.378e-02  9.163e-02  -0.696 0.486387
## factor(view)1             1.349e-01  1.782e-02   7.569 4.24e-14 ***
## factor(view)2             1.081e-01  1.107e-02   9.770  < 2e-16 ***
## factor(view)3             2.080e-01  1.516e-02  13.722  < 2e-16 ***
## factor(view)4             3.311e-01  2.586e-02  12.801  < 2e-16 ***
## factor(zipcode)98002     -9.638e-02  2.957e-02  -3.259 0.001124 **
## factor(zipcode)98003     -2.303e-02  2.700e-02  -0.853 0.393742
## factor(zipcode)98004      9.594e-01  4.807e-02  19.960  < 2e-16 ***
## factor(zipcode)98005      5.590e-01  5.060e-02  11.048  < 2e-16 ***
## factor(zipcode)98006      4.690e-01  4.194e-02  11.181  < 2e-16 ***
## factor(zipcode)98007      4.666e-01  5.344e-02   8.731  < 2e-16 ***
## factor(zipcode)98008      4.419e-01  5.062e-02   8.731  < 2e-16 ***
## factor(zipcode)98010      2.868e-01  4.382e-02   6.545 6.39e-11 ***
## factor(zipcode)98011      1.828e-01  6.563e-02   2.786 0.005351 **
## factor(zipcode)98014      1.437e-01  7.236e-02   1.986 0.047032 *
## factor(zipcode)98019      1.282e-01  7.146e-02   1.793 0.072970 .
## factor(zipcode)98022      8.168e-02  3.983e-02   2.051 0.040330 *
## factor(zipcode)98023     -7.782e-02  2.441e-02  -3.188 0.001439 **
## factor(zipcode)98024      3.541e-01  5.961e-02   5.941 2.97e-09 ***
## factor(zipcode)98027      3.940e-01  4.354e-02   9.049  < 2e-16 ***
## factor(zipcode)98028      1.459e-01  6.403e-02   2.279 0.022720 *
## factor(zipcode)98029      4.310e-01  4.939e-02   8.726  < 2e-16 ***
## factor(zipcode)98030     -9.402e-03  2.875e-02  -0.327 0.743615
## factor(zipcode)98031     -1.632e-02  3.017e-02  -0.541 0.588482
## factor(zipcode)98032     -1.263e-01  3.573e-02  -3.534 0.000412 ***
## factor(zipcode)98033      5.512e-01  5.439e-02  10.135  < 2e-16 ***
## factor(zipcode)98034      2.856e-01  5.864e-02   4.870 1.14e-06 ***
## factor(zipcode)98038      1.500e-01  3.280e-02   4.573 4.89e-06 ***
## factor(zipcode)98039      1.150e+00  7.292e-02  15.764  < 2e-16 ***
## factor(zipcode)98040      7.115e-01  4.235e-02  16.802  < 2e-16 ***
## factor(zipcode)98042      3.191e-02  2.822e-02   1.131 0.258196
## factor(zipcode)98045      2.615e-01  6.003e-02   4.357 1.34e-05 ***
## factor(zipcode)98052      4.106e-01  5.609e-02   7.321 2.75e-13 ***
## factor(zipcode)98053      4.333e-01  5.973e-02   7.254 4.47e-13 ***
## factor(zipcode)98055      1.587e-02  3.384e-02   0.469 0.639000
```

```
## factor(zipcode)98056      1.631e-01  3.695e-02   4.415 1.02e-05 ***
## factor(zipcode)98058      7.370e-02  3.147e-02   2.342 0.019225 *
## factor(zipcode)98059      2.400e-01  3.696e-02   6.492 9.03e-11 ***
## factor(zipcode)98065      3.256e-01  5.678e-02   5.734 1.02e-08 ***
## factor(zipcode)98070      1.949e-01  4.154e-02   4.692 2.76e-06 ***
## factor(zipcode)98072      2.595e-01  6.509e-02   3.987 6.76e-05 ***
## factor(zipcode)98074      3.606e-01  5.303e-02   6.801 1.12e-11 ***
## factor(zipcode)98075      4.130e-01  5.056e-02   8.169 3.67e-16 ***
## factor(zipcode)98077      2.474e-01  6.749e-02   3.666 0.000248 ***
## factor(zipcode)98092      2.701e-02  2.669e-02   1.012 0.311679
## factor(zipcode)98102      6.418e-01  5.738e-02  11.186  < 2e-16 ***
## factor(zipcode)98103      5.430e-01  5.290e-02  10.264  < 2e-16 ***
## factor(zipcode)98105      6.708e-01  5.441e-02  12.329  < 2e-16 ***
## factor(zipcode)98106      1.511e-01  3.996e-02   3.782 0.000157 ***
## factor(zipcode)98107      5.496e-01  5.568e-02   9.871  < 2e-16 ***
## factor(zipcode)98108      1.786e-01  4.311e-02   4.142 3.48e-05 ***
## factor(zipcode)98109      6.437e-01  5.756e-02  11.185  < 2e-16 ***
## factor(zipcode)98112      7.366e-01  4.972e-02  14.815  < 2e-16 ***
## factor(zipcode)98115      5.170e-01  5.370e-02   9.626  < 2e-16 ***
## factor(zipcode)98116      5.275e-01  4.367e-02  12.080  < 2e-16 ***
## factor(zipcode)98117      5.057e-01  5.431e-02   9.311  < 2e-16 ***
## factor(zipcode)98118      2.716e-01  3.821e-02   7.109 1.28e-12 ***
## factor(zipcode)98119      6.937e-01  5.371e-02  12.917  < 2e-16 ***
## factor(zipcode)98122      4.845e-01  4.743e-02  10.215  < 2e-16 ***
## factor(zipcode)98125      2.859e-01  5.819e-02   4.913 9.15e-07 ***
## factor(zipcode)98126      3.436e-01  4.014e-02   8.561  < 2e-16 ***
## factor(zipcode)98133      1.515e-01  6.024e-02   2.514 0.011943 *
## factor(zipcode)98136      4.517e-01  4.143e-02  10.903  < 2e-16 ***
## factor(zipcode)98144      4.450e-01  4.432e-02  10.041  < 2e-16 ***
## factor(zipcode)98146      7.597e-02  3.709e-02   2.048 0.040609 *
## factor(zipcode)98148     -2.833e-02  4.971e-02  -0.570 0.568740
## factor(zipcode)98155      1.384e-01  6.221e-02   2.224 0.026174 *
## factor(zipcode)98166      1.944e-01  3.250e-02   5.981 2.33e-09 ***
## factor(zipcode)98168     -6.324e-02  3.552e-02  -1.780 0.075045 .
## factor(zipcode)98177      2.833e-01  6.277e-02   4.513 6.51e-06 ***
## factor(zipcode)98178     -2.562e-02  3.687e-02  -0.695 0.487207
## factor(zipcode)98188     -3.452e-02  3.559e-02  -0.970 0.332119
## factor(zipcode)98198     -4.826e-02  2.786e-02  -1.732 0.083235 .
## factor(zipcode)98199      5.527e-01  5.185e-02  10.659  < 2e-16 ***
## factor(condition)2       -2.484e-02  6.484e-02  -0.383 0.701674
## factor(condition)3        1.587e-01  6.029e-02   2.632 0.008519 **
## factor(condition)4        1.930e-01  6.030e-02   3.201 0.001376 **
## factor(condition)5        2.516e-01  6.063e-02   4.150 3.37e-05 ***
## bedrooms:bathrooms        5.950e-02  1.286e-02   4.627 3.78e-06 ***
## bedrooms:grade            2.201e-02  4.176e-03   5.270 1.40e-07 ***
## bathrooms:grade           4.292e-02  6.566e-03   6.536 6.75e-11 ***
## bedrooms:bathrooms:grade -7.899e-03  1.567e-03  -5.040 4.76e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1885 on 6997 degrees of freedom
## Multiple R-squared:  0.8756, Adjusted R-squared:  0.874
## F-statistic: 547.3 on 90 and 6997 DF,  p-value: < 2.2e-16
```

```r
predOfModel6 = predict(multiple_model4, newdata = testing)
rmse6 = sqrt(mean((testing$price - exp(predOfModel6))^2))
rmse6
```

```
## [1] 134465.1
```