Midterm Project

Rong Li

2020/12/8

Abstract

- **Context**: As house price is related to every one of us, it's especially important to understand what factors may influence it.
- **Methods**: We developed a linear regression model to assess how these house features correlate with house price.
- **Finding**: Our model results found that many of these variables did have a effect on price.
- **Implications**: Adjusting the method for measurement may be a necessary step forward for effectively assessing price.

Introduction

- **Context**. It is common knowledge that house price is affected by many factors. And the house price has attracted much attention. Analyzing the data can give us a general idea of the important factors that influence the house price.
- **Report**. This report will investigate how house price in Seattle can be affected by multiple features of house, including the number of bedrooms, the number of bathrooms, the house area and etc.
  We will use a linear regression model to find the relationship between these features and price.
  We will analyze data downloaded from github, containing house transactions in Seattle from 2014-2015.

Method

Data Cleaning and Selection

Data was downloaded on github and cleaned using stringr package in R. We remove the NAs out of the raw data.
The 'date' column in our data was separated into year/month/day.
We used the transaction year minus the built year to get the house age. If the house has been renovated, we regarded transaction year minus renovated year as house age. Then we calculated the log of house price.
The final dataset included 21613 house transactions across Seattle.

EDA

After checking summary of the data, I wonder several things and get the conclusions by plots: (click here to see summary)
a) What is the most common house? (click here to see the plot)
The most common house is house with one floor and three bedrooms. Knowing this can help the constructors build this kind of houses more.

b) In which month are houses best sold? (click here to see the plot)
Houses are best sold in May. And the turnover is higher in summer than in winter.
c) Where are these houses located in? (click here to see the plot)
These houses are mainly located in Seattle. The latitude ranges from 47.16 to 47.78 and longtitude ranges from -122.5 to -121.3.
d) What factors may influence the house price? (click here to see the plots)
According to the boxplots and scatter plots, the number of bedrooms & the number of bathrooms & square of living area & the number of floors & whether the house is on waterfront & the house view & the house condition $ the house grade & the neibourhoods & house age may all influence the house price.

Model

In order to predict the house price in Seattle, we dicided to use a linear regression model.
The predictors are bedrooms, bathrooms, sqft_living, floors, waterfront, view, condition, grade, zipcode and age. Since the house price is high, we take the logarithm of house price as outcome.
The output is very long, so it is put in appendix. (click in here)

Results

Coefficients & Estimates

From the model summary, R-squared is 0.87 which means the model fits well. The model p value and most coefficient p values are less than the significance level 0.05, so we know we have a statistically significant model.
Bathrooms, sqft_living, grade, age, waterfront and conditon have evidence of a positive effect on house price.
The effect of age is relatively small, with a year increase in age correlating with an average expected increase of 0.00019 in house price on log scale.
On average, when other predictors are remained the same, the house price have this relationship: houses with 1.5 floors $>$ 2 floors $>$ 2.5 floors $>$ 1 floor $>$ 3.5 floors $>$ 3 floors.
On average, when other predictors are remained the same, the house price have this relationship: houses with 4 level view$>$ 3 level view $>$ 1 level view $>$ 2 level view $>$ 0 level view.
On average, when other predictors are remained the same, the houses in area with zipcode of 98039 are the most expensive. And the houses in area with zipcode of 98002 are the cheapest.
Additionally, the number of bedrooms is displayed to have a negative effect on price, with an estimated decrease of 0.002 in price on log scale for one unit of increase in bedrooms.

Multicollinearity & Residual Plots

We noticed that all vifs are less than 5, which indicates the multicollinearity is small.
In figure 1, the two plots look flat and most of the point are evenly distributed on both sides of the red line.

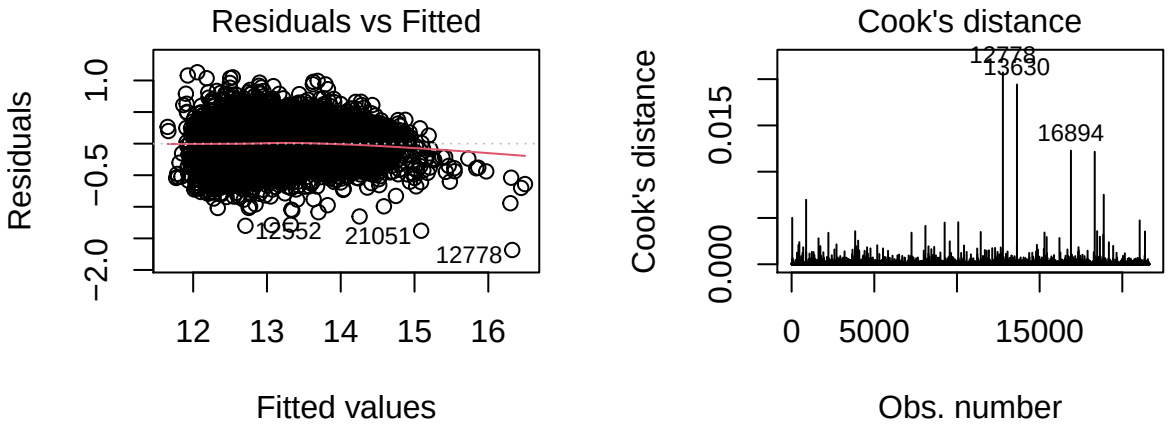


Figure 1: These two pictures show the residuals vs fitted values and the Cook's distance.

Figure 2 shows a Q-Q plot which is lower towards the bottom and higher towards the top. It shows a dataset with "fat tails", meaning that compared to the normal distribution there is more data located at the extremes of the distribution and less data in the center of the distribution. In terms of quantiles this means that the first quantile is much less than the first theoretical quantile and the last quantile is greater than the last theoretical quantile. In brief, the model looks not bad.

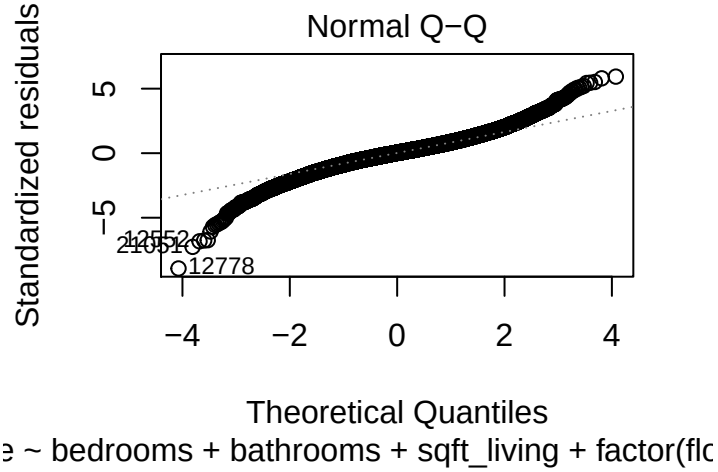


Figure 2: The Q-Q plot shows how similar are the quantiles in my dataset compared to what the quantiles of my dataset would be if my dataset followed the Gaussian (Normal) distribution

Propensity Score Matching

We tried to compare the outcome (house price) of places on waterfronts vs off of them.
One of the easiest way is to look at the coefficient and p value for waterfront in previous regression model. $\beta_{watefront} = 0.44$ and p value is significant on the level of 0.05. We can believe on average, the price of houses on waterfront will be 0.44 higher than off waterfront on the log scale.
Another way is to do propensity score matching. We firstly checked comparison of unmatched samples and calculated the propensity scores from logistic regression. The level of logprice seems to be significantly different in the two groups (on waterfront vs off waterfront). (click in here to see the propensity scores)
Then we matched the two samples using the matchit() function of the MatchIt package. After matching the samples, the size of the off waterfrant sample was reduced to the size of the Waterfront sample (n=163). We checked comparison of matched samples. The levels of the variables

bedrooms/bathrooms/sqft_living/floors/condition/grade/zipcode/age are nearly identical after matching in the two groups(on waterfront vs off waterfront).
We used this subset to run regression model and assessd the treatment effect between comparable groups. Finally, we found a difference between the the two groups. (click in here to see the summary of regression model)
Overall, the waterfront has a positive effect on house price.

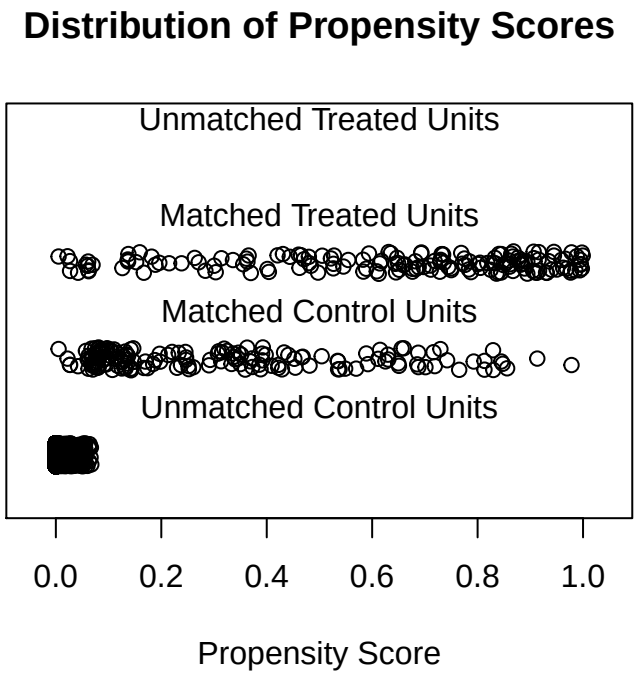


Figure 3: This is the visualization of the propensity scores distribution.

Validation

We fistly tried the validation set approach, spliting the data into two parts: training set and testing set, with the propotion of 8:2.
The model was built on the training dataset. We applied the model to the testing set to predict outcome values. Then we qualified the prediction error as the mean squared difference between the observed and the predicted outcome values.
We got R2 = 0.868, RMSE = 0.192 and MAE = 0.142. The prediction is very accurate. The average prediction error is low.

A disadvantage is that we built a model on a fraction of the data set only, possibly leaving out some interesting information about data, leading to higher bias.
Therefore, we set k = 5 and tried the k-fold cross validation instead. We got R2 = 0.868, RMSE = 0.192 and MAE = 0.141. The prediction is very accurate. The average prediction error is low.
Overall, the prediction of the model is great.

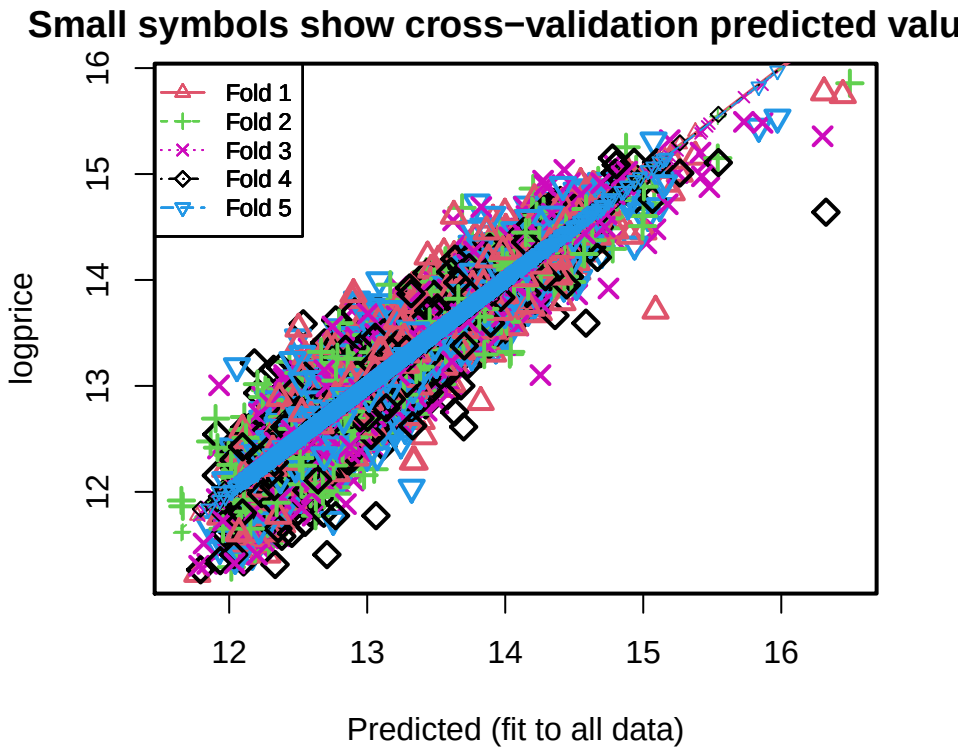


Figure 4: The prediction is very accurate. The average prediction error is low.

Discussion

Many of our model results seem to line up well with literature. The more area of the houses usually comes with larger number of bathrooms, contributing to higher price. The houses on the waterfront are more expensive than off of them. Different neighbourhoods have different price, maybe because of the nearby public facilities such as good schools, commercial center, etc. The better the house condition is, the higher the price. The higher the house grade is, the higher the price.
The positive effect of house age is unexpected, which means the older the house, the higher the price. Usually people tends to buy new houses. It is also strange that the increase of view level doesn't always lead to the increase of house price. It may be important in the future to adjust the method for measuring view level.

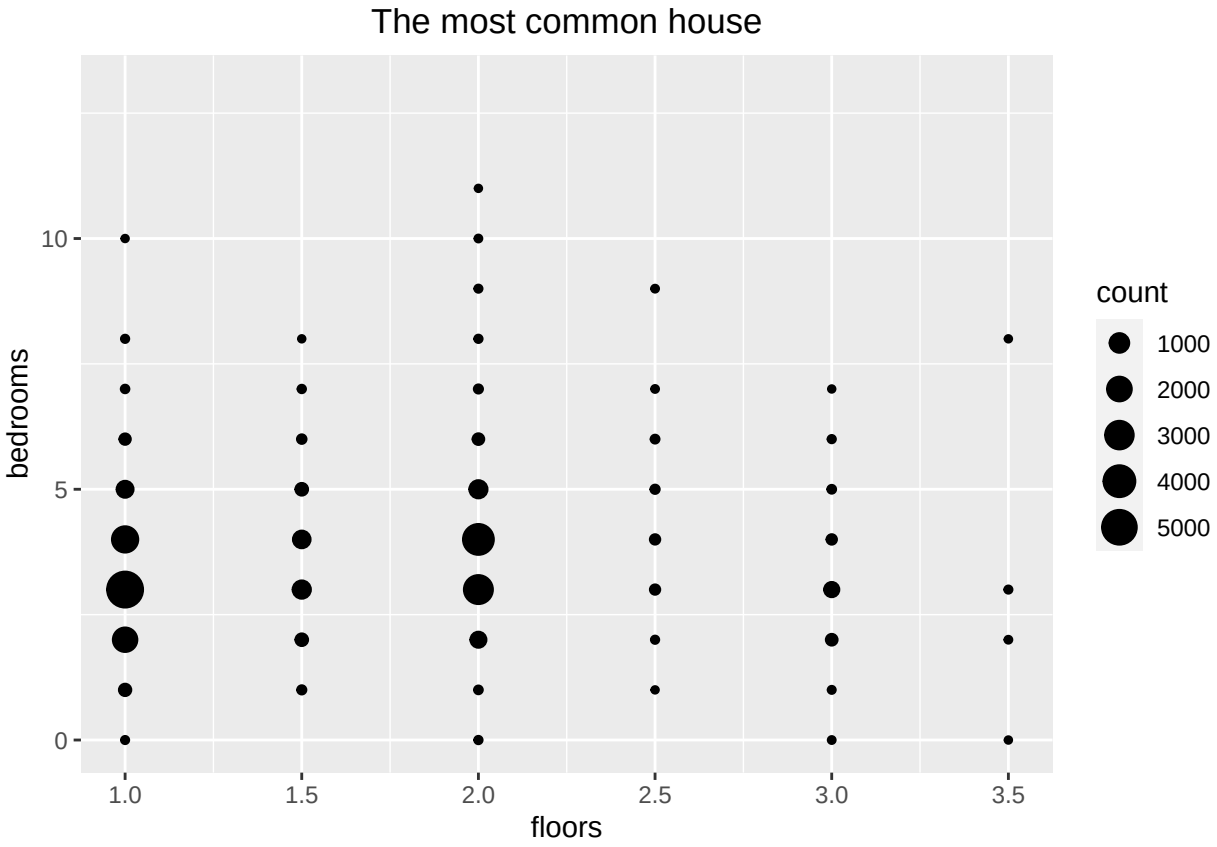
Appendix

Summary of Data

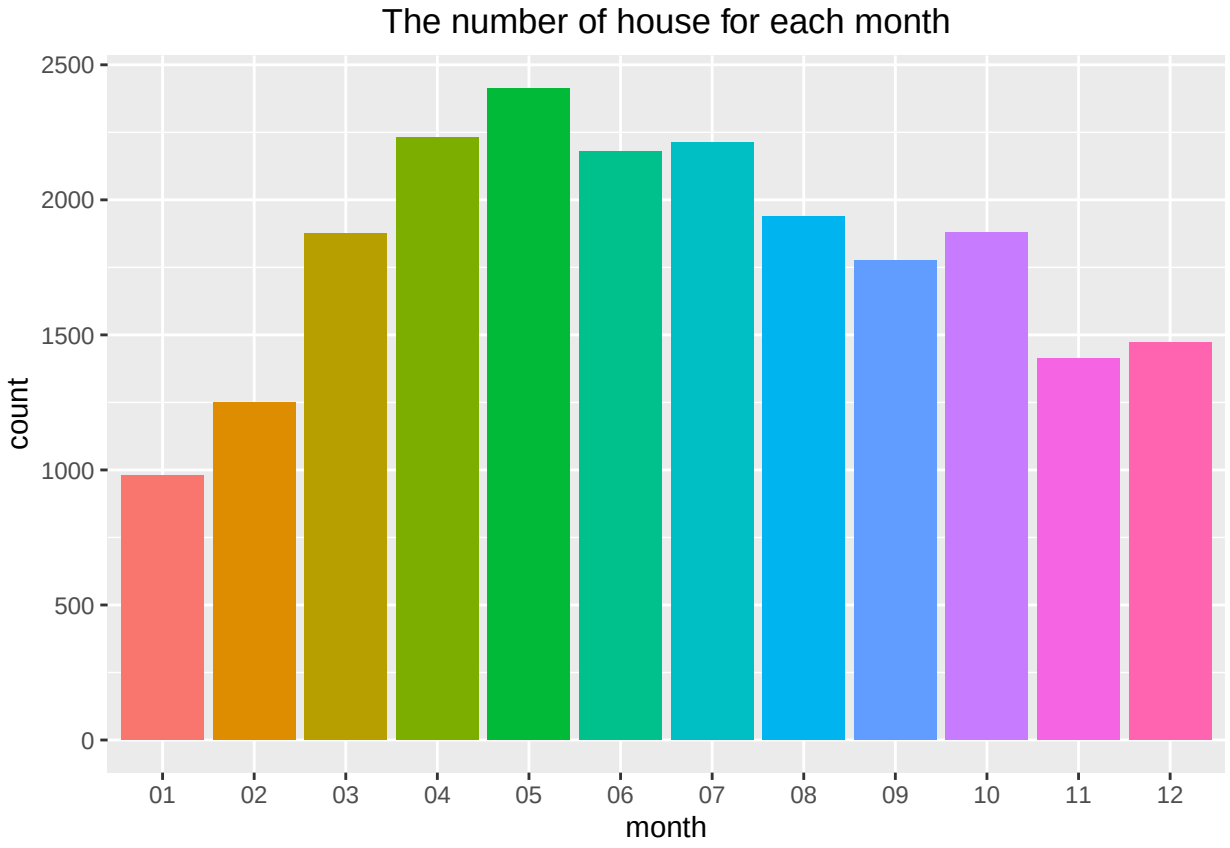
```
##        id                 date               price            bedrooms
##  Min.   :1.000e+06   Length:21613       Min.   :  75000   Min.   : 0.000
##  1st Qu.:2.123e+09   Class :character   1st Qu.: 321950   1st Qu.: 3.000
##  Median :3.905e+09   Mode  :character   Median : 450000   Median : 3.000
##  Mean   :4.580e+09                      Mean   : 540088   Mean   : 3.371
##  3rd Qu.:7.309e+09                      3rd Qu.: 645000   3rd Qu.: 4.000
##  Max.   :9.900e+09                      Max.   :7700000   Max.   :33.000
##    bathrooms      sqft_living       sqft_lot           floors
##  Min.   :0.000   Min.   :  290   Min.   :    520   Min.   :1.000
##  1st Qu.:1.750   1st Qu.: 1427   1st Qu.:   5040   1st Qu.:1.000
##  Median :2.250   Median : 1910   Median :   7618   Median :1.500
##  Mean   :2.115   Mean   : 2080   Mean   :  15107   Mean   :1.494
##  3rd Qu.:2.500   3rd Qu.: 2550   3rd Qu.:  10688   3rd Qu.:2.000
##  Max.   :8.000   Max.   :13540   Max.   :1651359   Max.   :3.500
##    waterfront            view          condition         grade
##  Min.   :0.000000   Min.   :0.0000   Min.   :1.000   Min.   : 1.000
##  1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.: 7.000
##  Median :0.000000   Median :0.0000   Median :3.000   Median : 7.000
##  Mean   :0.007542   Mean   :0.2343   Mean   :3.409   Mean   : 7.657
##  3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:4.000   3rd Qu.: 8.000
##  Max.   :1.000000   Max.   :4.0000   Max.   :5.000   Max.   :13.000
##    sqft_above   sqft_basement       yr_built     yr_renovated
##  Min.   : 290   Min.   :   0.0   Min.   :1900   Min.   :   0.0
##  1st Qu.:1190   1st Qu.:   0.0   1st Qu.:1951   1st Qu.:   0.0
##  Median :1560   Median :   0.0   Median :1975   Median :   0.0
##  Mean   :1788   Mean   : 291.5   Mean   :1971   Mean   :  84.4
##  3rd Qu.:2210   3rd Qu.: 560.0   3rd Qu.:1997   3rd Qu.:   0.0
##  Max.   :9410   Max.   :4820.0   Max.   :2015   Max.   :2015.0
##     zipcode           lat             long        sqft_living15
##  Min.   :98001   Min.   :47.16   Min.   :-122.5   Min.   : 399
##  1st Qu.:98033   1st Qu.:47.47   1st Qu.:-122.3   1st Qu.:1490
##  Median :98065   Median :47.57   Median :-122.2   Median :1840
##  Mean   :98078   Mean   :47.56   Mean   :-122.2   Mean   :1987
##  3rd Qu.:98118   3rd Qu.:47.68   3rd Qu.:-122.1   3rd Qu.:2360
##  Max.   :98199   Max.   :47.78   Max.   :-121.3   Max.   :6210
##    sqft_lot15          year         month               day
##  Min.   :   651   Min.   :2014   Length:21613       Length:21613
##  1st Qu.:  5100   1st Qu.:2014   Class :character   Class :character
##  Median :  7620   Median :2014   Mode  :character   Mode  :character
##  Mean   : 12768   Mean   :2014
##  3rd Qu.: 10083   3rd Qu.:2015
##  Max.   :871200   Max.   :2015
##    yr_latest         age            logprice       Group
##  Min.   :1900   Min.   : -1.00   Min.   :11.23   Mode :logical
##  1st Qu.:1954   1st Qu.: 15.00   1st Qu.:12.68   FALSE:21450
##  Median :1977   Median : 37.00   Median :13.02   TRUE :163
##  Mean   :1973   Mean   : 40.94   Mean   :13.05
##  3rd Qu.:1999   3rd Qu.: 60.00   3rd Qu.:13.38
##  Max.   :2015   Max.   :115.00   Max.   :15.86
```

Plots

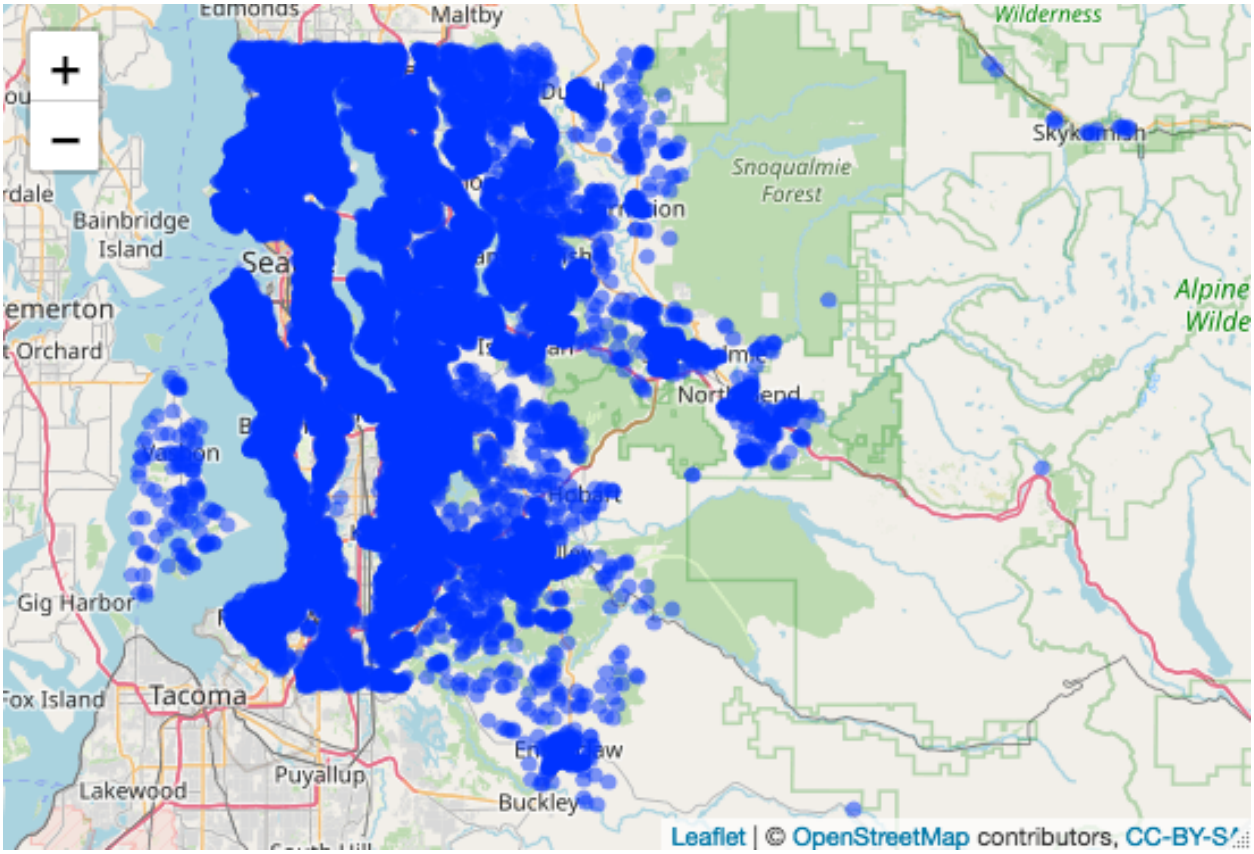
a) What is the most common house:



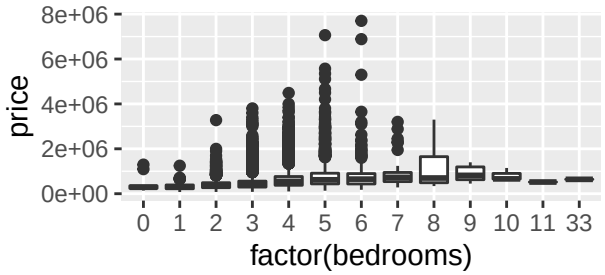
b) In which month are houses best sold?

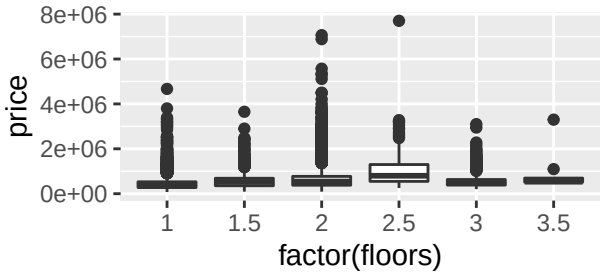


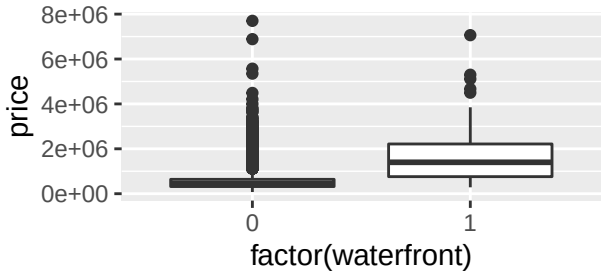
c) Where are these houses located in?

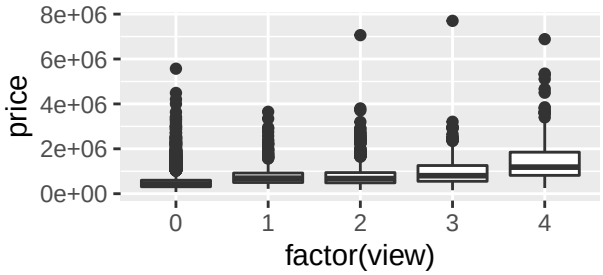


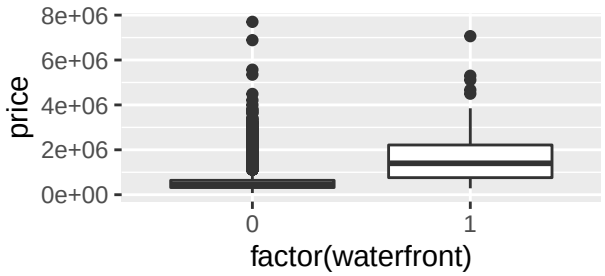
d) What factors may influence the house price?

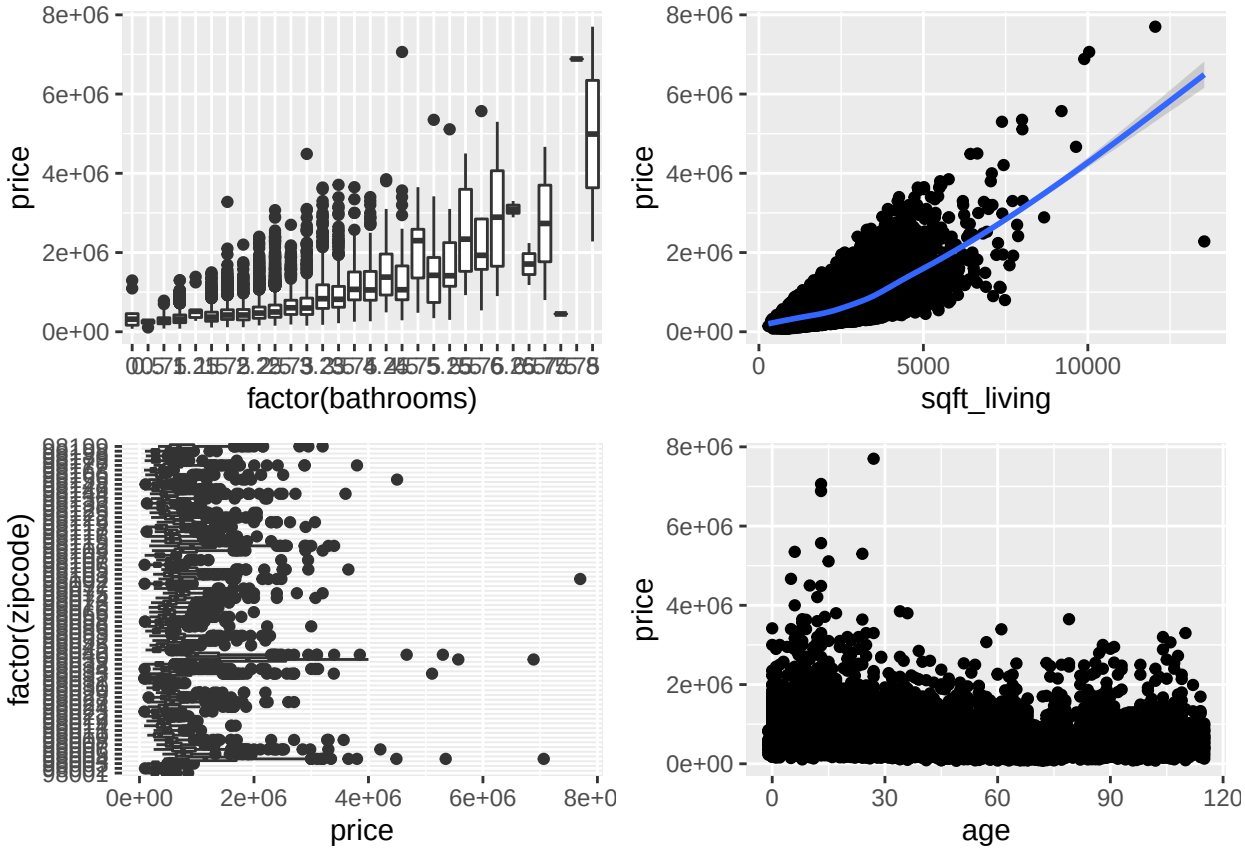












Output of the Linear Regression Model

```
##
## Call:
## lm(formula = logprice ~ bedrooms + bathrooms + sqft_living +
##     factor(floors) + factor(waterfront) + factor(view) + factor(condition) +
##     grade + factor(zipcode) + age, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6844 -0.1014  0.0056  0.1075  1.1303
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.099e+01  3.857e-02 284.850  < 2e-16 ***
## bedrooms               -2.147e-03  1.823e-03  -1.178 0.238784
## bathrooms               2.576e-02  3.099e-03   8.312  < 2e-16 ***
## sqft_living             2.075e-04  2.983e-06  69.568  < 2e-16 ***
## factor(floors)1.5       4.457e-02  5.083e-03   8.768  < 2e-16 ***
## factor(floors)2         6.959e-03  3.810e-03   1.827 0.067770 .
## factor(floors)2.5       5.402e-03  1.562e-02   0.346 0.729467
## factor(floors)3        -1.205e-01  9.330e-03 -12.918  < 2e-16 ***
## factor(floors)3.5      -9.393e-02  6.775e-02  -1.386 0.165630
## factor(waterfront)1     4.391e-01  1.905e-02  23.054  < 2e-16 ***
## factor(view)1           1.356e-01  1.078e-02  12.583  < 2e-16 ***
## factor(view)2           1.224e-01  6.556e-03  18.672  < 2e-16 ***
```

```
## factor(view)3         1.922e-01  8.946e-03  21.480  < 2e-16 ***
## factor(view)4         2.809e-01  1.381e-02  20.347  < 2e-16 ***
## factor(condition)2    1.040e-01  3.788e-02   2.745 0.006061 **
## factor(condition)3    2.171e-01  3.512e-02   6.181 6.47e-10 ***
## factor(condition)4    2.599e-01  3.511e-02   7.401 1.40e-13 ***
## factor(condition)5    3.166e-01  3.533e-02   8.961  < 2e-16 ***
## grade                 1.118e-01  2.021e-03  55.323  < 2e-16 ***
## factor(zipcode)98002 -4.272e-02  1.687e-02  -2.532 0.011362 *
## factor(zipcode)98003  4.381e-03  1.521e-02   0.288 0.773263
## factor(zipcode)98004  1.115e+00  1.487e-02  74.981  < 2e-16 ***
## factor(zipcode)98005  7.508e-01  1.798e-02  41.763  < 2e-16 ***
## factor(zipcode)98006  6.315e-01  1.339e-02  47.153  < 2e-16 ***
## factor(zipcode)98007  6.496e-01  1.902e-02  34.162  < 2e-16 ***
## factor(zipcode)98008  6.411e-01  1.523e-02  42.094  < 2e-16 ***
## factor(zipcode)98010  2.852e-01  2.157e-02  13.218  < 2e-16 ***
## factor(zipcode)98011  4.568e-01  1.697e-02  26.921  < 2e-16 ***
## factor(zipcode)98014  3.506e-01  1.989e-02  17.629  < 2e-16 ***
## factor(zipcode)98019  3.524e-01  1.713e-02  20.571  < 2e-16 ***
## factor(zipcode)98022  8.230e-02  1.608e-02   5.118 3.11e-07 ***
## factor(zipcode)98023 -3.603e-02  1.319e-02  -2.731 0.006318 **
## factor(zipcode)98024  4.942e-01  2.348e-02  21.046  < 2e-16 ***
## factor(zipcode)98027  5.143e-01  1.381e-02  37.254  < 2e-16 ***
## factor(zipcode)98028  4.179e-01  1.515e-02  27.576  < 2e-16 ***
## factor(zipcode)98029  5.816e-01  1.476e-02  39.410  < 2e-16 ***
## factor(zipcode)98030  5.543e-02  1.558e-02   3.557 0.000376 ***
## factor(zipcode)98031  7.304e-02  1.529e-02   4.778 1.78e-06 ***
## factor(zipcode)98032 -5.185e-02  1.982e-02  -2.616 0.008902 **
## factor(zipcode)98033  7.730e-01  1.365e-02  56.622  < 2e-16 ***
## factor(zipcode)98034  5.275e-01  1.296e-02  40.696  < 2e-16 ***
## factor(zipcode)98038  1.865e-01  1.279e-02  14.585  < 2e-16 ***
## factor(zipcode)98039  1.257e+00  2.904e-02  43.298  < 2e-16 ***
## factor(zipcode)98040  8.677e-01  1.543e-02  56.218  < 2e-16 ***
## factor(zipcode)98042  7.009e-02  1.294e-02   5.417 6.12e-08 ***
## factor(zipcode)98045  3.490e-01  1.630e-02  21.406  < 2e-16 ***
## factor(zipcode)98052  6.421e-01  1.287e-02  49.894  < 2e-16 ***
## factor(zipcode)98053  6.127e-01  1.391e-02  44.034  < 2e-16 ***
## factor(zipcode)98055  1.291e-01  1.539e-02   8.386  < 2e-16 ***
## factor(zipcode)98056  3.083e-01  1.382e-02  22.306  < 2e-16 ***
## factor(zipcode)98058  1.679e-01  1.345e-02  12.485  < 2e-16 ***
## factor(zipcode)98059  3.555e-01  1.339e-02  26.545  < 2e-16 ***
## factor(zipcode)98065  4.177e-01  1.486e-02  28.105  < 2e-16 ***
## factor(zipcode)98070  3.622e-01  2.057e-02  17.604  < 2e-16 ***
## factor(zipcode)98072  5.146e-01  1.534e-02  33.545  < 2e-16 ***
## factor(zipcode)98074  5.635e-01  1.367e-02  41.226  < 2e-16 ***
## factor(zipcode)98075  5.830e-01  1.439e-02  40.507  < 2e-16 ***
## factor(zipcode)98077  4.996e-01  1.701e-02  29.363  < 2e-16 ***
## factor(zipcode)98092  4.421e-02  1.431e-02   3.089 0.002009 **
## factor(zipcode)98102  8.987e-01  2.148e-02  41.839  < 2e-16 ***
## factor(zipcode)98103  7.802e-01  1.315e-02  59.321  < 2e-16 ***
## factor(zipcode)98105  9.023e-01  1.642e-02  54.959  < 2e-16 ***
## factor(zipcode)98106  2.769e-01  1.452e-02  19.076  < 2e-16 ***
## factor(zipcode)98107  7.942e-01  1.573e-02  50.485  < 2e-16 ***
## factor(zipcode)98108  3.069e-01  1.728e-02  17.754  < 2e-16 ***
## factor(zipcode)98109  9.439e-01  2.113e-02  44.666  < 2e-16 ***
```

```
## factor(zipcode)98112  1.006e+00  1.575e-02  63.898  < 2e-16 ***
## factor(zipcode)98115  7.742e-01  1.296e-02  59.749  < 2e-16 ***
## factor(zipcode)98116  7.053e-01  1.471e-02  47.953  < 2e-16 ***
## factor(zipcode)98117  7.575e-01  1.313e-02  57.705  < 2e-16 ***
## factor(zipcode)98118  4.148e-01  1.326e-02  31.282  < 2e-16 ***
## factor(zipcode)98119  9.251e-01  1.759e-02  52.605  < 2e-16 ***
## factor(zipcode)98122  7.538e-01  1.529e-02  49.293  < 2e-16 ***
## factor(zipcode)98125  5.418e-01  1.387e-02  39.053  < 2e-16 ***
## factor(zipcode)98126  4.848e-01  1.439e-02  33.684  < 2e-16 ***
## factor(zipcode)98133  4.303e-01  1.328e-02  32.390  < 2e-16 ***
## factor(zipcode)98136  6.305e-01  1.562e-02  40.357  < 2e-16 ***
## factor(zipcode)98144  6.065e-01  1.453e-02  41.739  < 2e-16 ***
## factor(zipcode)98146  2.481e-01  1.514e-02  16.380  < 2e-16 ***
## factor(zipcode)98148  1.495e-01  2.725e-02   5.488 4.12e-08 ***
## factor(zipcode)98155  4.070e-01  1.355e-02  30.036  < 2e-16 ***
## factor(zipcode)98166  2.977e-01  1.571e-02  18.949  < 2e-16 ***
## factor(zipcode)98168  4.881e-02  1.547e-02   3.156 0.001603 **
## factor(zipcode)98177  5.807e-01  1.575e-02  36.858  < 2e-16 ***
## factor(zipcode)98178  1.145e-01  1.559e-02   7.344 2.15e-13 ***
## factor(zipcode)98188  7.406e-02  1.922e-02   3.853 0.000117 ***
## factor(zipcode)98198  4.066e-02  1.524e-02   2.667 0.007650 **
## factor(zipcode)98199  8.111e-01  1.487e-02  54.543  < 2e-16 ***
## age                   1.882e-04  7.761e-05   2.424 0.015340 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1907 on 21524 degrees of freedom
## Multiple R-squared:  0.8694, Adjusted R-squared:  0.8688
## F-statistic:  1628 on 88 and 21524 DF,  p-value: < 2.2e-16
```

Propensity Scores

```
##       pr_score Group
## 1 2.762724e-11 FALSE
## 2 6.930102e-11 FALSE
## 3 1.824723e-11 FALSE
## 4 4.426844e-12 FALSE
## 5 7.802630e-12 FALSE
## 6 2.220446e-16 FALSE
```

```
##
## Call:  glm(formula = Group ~ bedrooms + bathrooms + sqft_living + factor(floors) +
##     factor(view) + factor(condition) + grade + factor(zipcode) +
##     age, family = binomial(), data = df)
##
## Coefficients:
##          (Intercept)              bedrooms             bathrooms
##           -40.666494            -0.853899             0.193352
##          sqft_living     factor(floors)1.5       factor(floors)2
##             0.000478             1.085568             0.909252
##    factor(floors)2.5       factor(floors)3     factor(floors)3.5
##            -0.320988             2.227624             1.876736
##        factor(view)1         factor(view)2         factor(view)3
##            17.973114            19.572254            20.965487
##        factor(view)4    factor(condition)2    factor(condition)3
```

```
##            25.213388             2.363306            -1.244399
##   factor(condition)4   factor(condition)5               grade
##            -1.858671            -0.977046            -0.225754
## factor(zipcode)98002 factor(zipcode)98003 factor(zipcode)98004
##             6.115711             2.727037            17.103930
## factor(zipcode)98005 factor(zipcode)98006 factor(zipcode)98007
##             1.041871            17.301715             4.686872
## factor(zipcode)98008 factor(zipcode)98010 factor(zipcode)98011
##            19.593259             2.571453             0.432639
## factor(zipcode)98014 factor(zipcode)98019 factor(zipcode)98022
##             1.874395             2.647765            -0.203290
## factor(zipcode)98023 factor(zipcode)98024 factor(zipcode)98027
##            20.991720             1.306869            18.845484
## factor(zipcode)98028 factor(zipcode)98029 factor(zipcode)98030
##            18.771882             2.998363             5.277467
## factor(zipcode)98031 factor(zipcode)98032 factor(zipcode)98033
##             5.444155            -1.074021            19.360755
## factor(zipcode)98034 factor(zipcode)98038 factor(zipcode)98039
##            18.770139             1.763491            18.265351
## factor(zipcode)98040 factor(zipcode)98042 factor(zipcode)98045
##            20.373767             3.631585             0.284480
## factor(zipcode)98052 factor(zipcode)98053 factor(zipcode)98055
##            21.655153            -3.079166             2.949061
## factor(zipcode)98056 factor(zipcode)98058 factor(zipcode)98059
##            20.013831             4.804457             3.634657
## factor(zipcode)98065 factor(zipcode)98070 factor(zipcode)98072
##            -1.350658            23.185180             3.476164
## factor(zipcode)98074 factor(zipcode)98075 factor(zipcode)98077
##            19.200617            20.192467             5.683784
## factor(zipcode)98092 factor(zipcode)98102 factor(zipcode)98103
##             0.399243            -3.105688            -1.922390
## factor(zipcode)98105 factor(zipcode)98106 factor(zipcode)98107
##            19.347844             3.160931            -0.977930
## factor(zipcode)98108 factor(zipcode)98109 factor(zipcode)98112
##             2.516113            -1.487773             2.290038
## factor(zipcode)98115 factor(zipcode)98116 factor(zipcode)98117
##            16.446180            15.076135            -0.880416
## factor(zipcode)98118 factor(zipcode)98119 factor(zipcode)98122
##            18.957152            -1.140433            -0.512652
## factor(zipcode)98125 factor(zipcode)98126 factor(zipcode)98133
##            19.831535             0.504042             3.189735
## factor(zipcode)98136 factor(zipcode)98144 factor(zipcode)98146
##            18.433650            17.435363            19.242057
## factor(zipcode)98148 factor(zipcode)98155 factor(zipcode)98166
##            18.646544            20.240304            20.472135
## factor(zipcode)98168 factor(zipcode)98177 factor(zipcode)98178
##             4.591381            15.127824            20.326417
## factor(zipcode)98188 factor(zipcode)98198 factor(zipcode)98199
##             2.473984            19.456352            18.914663
##                  age
##             0.011133
##
## Degrees of Freedom: 21612 Total (i.e. Null);  21525 Residual
## Null Deviance:       1918
```

```
## Residual Deviance: 396.2     AIC: 572.2
```

Regression Model of the Matched data

```
##
## Call:
## lm(formula = logprice ~ bedrooms + bathrooms + sqft_living +
##     factor(floors) + waterfront + factor(view) + factor(condition) +
##     grade + factor(zipcode) + age, data = df.match)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.05212 -0.13835  0.00295  0.14713  0.66069
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.263e+01  2.701e-01  46.746  < 2e-16 ***
## bedrooms              -1.976e-02  1.944e-02  -1.017 0.310071
## bathrooms              5.076e-02  2.541e-02   1.998 0.046725 *
## sqft_living            1.225e-04  2.046e-05   5.985 6.59e-09 ***
## factor(floors)1.5      1.220e-02  4.934e-02   0.247 0.804941
## factor(floors)2       -2.328e-02  3.850e-02  -0.605 0.545994
## factor(floors)2.5      6.842e-02  1.384e-01   0.494 0.621361
## factor(floors)3       -1.455e-01  7.762e-02  -1.875 0.061832 .
## waterfront             4.307e-01  3.166e-02  13.602  < 2e-16 ***
## factor(view)2         -8.322e-02  1.255e-01  -0.663 0.507733
## factor(view)3         -1.444e-01  1.204e-01  -1.199 0.231433
## factor(view)4         -8.985e-02  1.236e-01  -0.727 0.467743
## factor(condition)2     2.886e-01  2.475e-01   1.166 0.244673
## factor(condition)3     4.651e-01  1.907e-01   2.439 0.015353 *
## factor(condition)4     5.476e-01  1.917e-01   2.856 0.004608 **
## factor(condition)5     5.766e-01  1.946e-01   2.964 0.003303 **
## grade                  1.084e-01  1.692e-02   6.406 6.31e-10 ***
## factor(zipcode)98006 -5.883e-01  1.310e-01  -4.492 1.03e-05 ***
## factor(zipcode)98008 -5.133e-01  1.323e-01  -3.881 0.000130 ***
## factor(zipcode)98023 -1.243e+00  1.475e-01  -8.427 1.88e-15 ***
## factor(zipcode)98027 -8.093e-01  1.738e-01  -4.657 4.96e-06 ***
## factor(zipcode)98028 -8.537e-01  1.845e-01  -4.628 5.66e-06 ***
## factor(zipcode)98033 -2.854e-01  1.476e-01  -1.933 0.054214 .
## factor(zipcode)98034 -3.119e-01  1.404e-01  -2.222 0.027109 *
## factor(zipcode)98039 -1.756e-01  2.091e-01  -0.840 0.401657
## factor(zipcode)98040 -3.195e-01  1.257e-01  -2.543 0.011540 *
## factor(zipcode)98052 -7.107e-01  1.459e-01  -4.871 1.86e-06 ***
## factor(zipcode)98056 -5.340e-01  1.717e-01  -3.109 0.002070 **
## factor(zipcode)98070 -1.148e+00  1.286e-01  -8.921  < 2e-16 ***
## factor(zipcode)98074 -5.386e-01  1.431e-01  -3.764 0.000204 ***
## factor(zipcode)98075 -6.101e-01  1.299e-01  -4.696 4.17e-06 ***
## factor(zipcode)98105 -2.428e-01  1.672e-01  -1.452 0.147637
## factor(zipcode)98115 -4.017e-01  1.717e-01  -2.339 0.020025 *
## factor(zipcode)98116 -5.014e-01  1.535e-01  -3.267 0.001222 **
## factor(zipcode)98118 -6.814e-01  1.467e-01  -4.646 5.22e-06 ***
## factor(zipcode)98125 -5.461e-01  1.403e-01  -3.891 0.000125 ***
## factor(zipcode)98136 -5.130e-01  1.429e-01  -3.591 0.000389 ***
## factor(zipcode)98144 -3.832e-01  1.549e-01  -2.474 0.013939 *
```

```
## factor(zipcode)98146 -7.608e-01  1.347e-01  -5.648 3.99e-08 ***
## factor(zipcode)98155 -5.910e-01  1.461e-01  -4.044 6.79e-05 ***
## factor(zipcode)98166 -8.942e-01  1.292e-01  -6.918 3.10e-11 ***
## factor(zipcode)98177 -4.577e-01  1.493e-01  -3.065 0.002386 **
## factor(zipcode)98178 -9.232e-01  1.375e-01  -6.715 1.04e-10 ***
## factor(zipcode)98198 -1.132e+00  1.336e-01  -8.471 1.39e-15 ***
## factor(zipcode)98199 -4.914e-01  1.815e-01  -2.708 0.007194 **
## age                  -8.805e-04  8.528e-04  -1.032 0.302739
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2459 on 280 degrees of freedom
## Multiple R-squared:  0.8819, Adjusted R-squared:  0.8629
## F-statistic: 46.44 on 45 and 280 DF,  p-value: < 2.2e-16
```

Bibliography

- The main packages I use:
  1.Andrew Gelman and Yu-Sung Su (2020). arm: Data Analysis Using Regression and Multilevel/Hierarchical Models. R package version 1.11-2. https://CRAN.R-project.org/package=arm
  2.Daniel E. Ho, Kosuke Imai, Gary King, Elizabeth A. Stuart (2011). MatchIt: Nonparametric Preprocessing for Parametric Causal Inference. Journal of Statistical Software, Vol. 42, No. 8, pp. 1-28. URL https://www.jstatsoft.org/v42/i08/
  3.Max Kuhn (2020). caret: Classification and Regression Training. R package version 6.0-86. https://CRAN.R-project.org/package=caret
  4.H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.
  5.Kazuki Yoshida and Alexander Bartel (2020). tableone: Create 'Table 1' to Describe Baseline Characteristics with or without Propensity Score Weights. R package version 0.12.0. https://CRAN.R-project.org/package=tableone
- The data I use is downloaded from github.

Supplement

This part includes all the code:

```r
# load the date
setwd("/Users/amelia/Documents/mssp/MA678/MA678-Midterm-proposal")
df <- read.table("kc_house_data.txt", sep = ",", header = T)

df$date %<>% str_sub(1, 8)
df$year <- str_sub(df$date, 1, 4)
df$month <- str_sub(df$date, 5, 6)
df$day <- str_sub(df$date, 7, 8)
df$yr_latest <- ifelse(df$yr_built > df$yr_renovated, df$yr_built, df$yr_renovated)
df$year %<>% as.numeric()
df$yr_latest %<>% as.numeric()
df$age <- df$year - df$yr_latest
df$logprice <- log(df$price)
df %<>% na.omit()

df2 <- df %>% dplyr::select(logprice, bedrooms,
              bathrooms, sqft_living, floors, waterfront, view,
              condition, grade, zipcode, lat, long)
```

```r
# EDA
## summary
# summary(df)

## I wonder the most common house
comhouse <- df %>% group_by(bedrooms, floors) %>% summarise(count = sum(price > 0))
p1 <- ggplot(comhouse, aes(x = floors, y = bedrooms, size = count)) +
  geom_point() +
  ylim(0, 13) +
  ggtitle("The most common house") +
  theme(plot.title = element_text(hjust = 0.5))
p1
# As we can see from the picture, the most common house is house with one floor and three bedrooms.

## houses are best sold in which month?
monhouse <- df %>% group_by(month) %>% summarise(count = sum(price > 0))
p2 <- ggplot(monhouse, aes(x = month, y = count, fill = factor(month))) +
  geom_bar(stat = "identity") +
  guides(fill = F)
p2
# houses are best sold in May. And the turnover is higher in summer than in winter.

## I wonder the location of these houses
df%>%leaflet()%>%addTiles()%>%
  addCircleMarkers(lng=~long,lat=~lat, radius = 0.05, fill = F)
## They are in Washington state, and in Seatle.

## I wonder what factors may influence the house price
### will the number of bedrooms influence the house price?
p2 <- ggplot(data = df, mapping = aes(x = factor(bedrooms), price)) +
  geom_boxplot()

### will the number of bathrooms influence the house price?
p3 <- ggplot(data = df, mapping = aes(x = factor(bathrooms), price)) +
  geom_boxplot()

### will the living area influence the house price?
p4 <- ggplot(data = df, mapping = aes(x = sqft_living, price)) +
  geom_point() + geom_smooth()

### will the number of floors influence the house price?
p5 <- ggplot(data = df, mapping = aes(x = factor(floors), price)) +
  geom_boxplot()

### will the waterfront influence the house price?
p6 <- ggplot(data = df, mapping = aes(x = factor(waterfront), price)) +
  geom_boxplot()

### will the view influence the house price?
p7 <- ggplot(data = df, mapping = aes(x = factor(view), price)) +
  geom_boxplot()

### will the waterfront influence the house price?
```

```
p8 <- ggplot(data = df, mapping = aes(x = factor(waterfront), price)) +
  geom_boxplot()

### will the location influence the house price?
p9 <- ggplot(data = df, mapping = aes(x = factor(zipcode), price)) +
  geom_boxplot() +
  coord_flip()

### will the age influence the house price?
p10 <- ggplot(data = df, mapping = aes(x = age, price)) +
  geom_point()

grid.arrange(p2, p5, p6, p7, p8, nrow=3)
grid.arrange(p3, p4, p9, p10, nrow=2)

## Model
# fit linear regession model
fit1 <- lm(logprice ~ bedrooms + bathrooms + sqft_living + factor(floors) + factor(waterfront) + factor
           data = df)
summary(fit1)
vif(fit1)
# all vifs are <5, which indicates the multicollinearity is small.

# residual plot
par(mfrow=c(2, 2))
plot(fit1)


# Propensity Score Matching
df$Group = as.logical(df$waterfront==1)

#Then we compare the distribution of bedrooms/bathrooms/sqft_living/floors/waterfront/view/condition/gr
table1 <- CreateTableOne(vars = c('bedrooms', 'bathrooms', 'sqft_living', 'floors', 'view', 'condition'
                         data = df,
                         factorVars = c('floors', 'view', 'condition', 'zipcode'),
                         strata = 'waterfront')
table1 <- print(table1,
                printToggle = FALSE,
                noSpaces = TRUE)
kable(table1[,1:3],
      align = 'c',
      caption = 'Table 1: Comparison of unmatched samples')
# The level of logprice seems to have no significantly difference in the two groups.


#Propensity scores are from logistic regression, so let's look at how this goes.
scoremodel = glm(Group ~ bedrooms + bathrooms + sqft_living + factor(floors) + factor(view) + factor(co
propenscores = data.frame(pr_score = predict(scoremodel,type="response"),
                          Group= scoremodel$model$Group)
head(propenscores)
scoremodel

propenscores %>%
```

```
  mutate(Group == ifelse(Group==T, "onWaterfront","offWaterfront")) %>%
  ggplot(aes(x = pr_score,fill=Group)) +
  geom_histogram(color="white") +
  facet_wrap(~Group) +
  xlab("Probability of getting treated") +
  theme_bw()

# Matching the sample
#The package MatchIt lets us use propensity score matching more easily and gives us a variety of option
match.it = matchit(Group ~ bedrooms + bathrooms + sqft_living + factor(floors) + factor(view) + factor(
                   data=df, method="nearest")
a <- summary(match.it)

#Here we can see our remaining data post-matching
df.match = match.data(match.it)
head(df.match)
#After matching the samples, the size of the population sample was reduced to the size of the onWaterfr

#We can compare our groups again with our new matched dataset.
table2 <- CreateTableOne(vars = c('bedrooms', 'bathrooms', 'sqft_living', 'floors', 'view', 'condition'
                         data = df.match,
                         factorVars = c('floors', 'view', 'condition', 'zipcode'),
                         strata = 'waterfront')
table2 <- print(table2,
                printToggle = FALSE,
                noSpaces = TRUE)
kable(table2[,1:3],
      align = 'c',
      caption = 'Table 2: Comparison of matched samples')

kable(a$sum.matched[c(1,2,4)], digits = 2, align = 'c',
      caption = 'Table 3: Summary of balance for matched data')

#We can now estimate treatment effects. We can use t-tests or other things, but here I'll use a linear
model = lm(logprice ~ bedrooms + bathrooms + sqft_living + factor(floors) + waterfront + factor(view) +
summary(model)

plot(match.it, type = 'jitter', interactive = FALSE)

# Validation
## create training and testing data
set.seed(1)
trainingrow <- sample(1:nrow(df), 0.8*nrow(df))
training <- df[trainingrow, ]
testing <- df[-trainingrow,]

## fit the testing data
predtest <- predict(fit1, testing)

## calculate the accuracy and error rate
actualpred <- data.frame(cbind(actuals = testing$logprice, predicteds = predtest))
correlationaccuracy <- cor(actualpred)
correlationaccuracy
```

```
data.frame( R2 = R2(predtest, testing$logprice),
            RMSE = RMSE(predtest, testing$logprice),
            MAE = MAE(predtest, testing$logprice))
#head(actualpred)


## k-fold Cross-Validation
par(mfrow=c(1, 1))
cv.lm(df, form.lm=formula(logprice ~ bedrooms + bathrooms + sqft_living + factor(floors) + factor(waterf
      m=5, dots = FALSE, plotit = T, printit = T)

# Define train control for k fold cross validation
train_control <- trainControl(method="cv", number=5)
# Fit Naive Bayes Model
model <- train(logprice ~ bedrooms + bathrooms + sqft_living + factor(floors) + factor(waterfront) + fac
               data=df,
               trControl=train_control,
               method="lm")
# Summarise Results
print(model)
```