Sberbank Russian Housing Market

- **The goal for this project is to predict housing price in Russia**

Step 1: Collecting data

- I upload the housing data to my github account but the data can be found at Kaggle website.
- I modified the data from the original data because I couldn't apply algorithms to the whole data to run because there are over 300 features.

```
rm(list=ls())

house_raw <- read.csv("https://github.com/keosotra/Competition/raw/master/Dat
a/House.csv?raw=true", stringsAsFactors = FALSE)
```

- Removed the ID variable from the data set and inspect all the features.

```
house <- house_raw[-1]

str(house)

## 'data.frame':    30471 obs. of  41 variables:
##  $ price_doc                             : int  5850000 6000000 570000
0 13100000 16331452 9100000 5500000 2000000 5300000 2000000 ...
##  $ life_sq                               : int  27 19 29 50 77 46 14 4
4 27 21 ...
##  $ floor                                 : int  4 3 2 9 4 14 10 5 5 9
...
##  $ max_floor                             : int  NA NA NA NA NA NA NA N
A NA NA ...
##  $ build_year                            : int  NA NA NA NA NA NA NA N
A NA NA ...
##  $ num_room                              : int  NA NA NA NA NA NA NA N
A NA NA ...
##  $ product_type                          : chr  "Investment" "OwnerOcc
upier" "Investment" "OwnerOccupier" ...
##  $ sub_area                              : chr  "Juzhnoe Butovo" "Pose
lenie Vnukovskoe" "Perovo" "Poselenie Voskresenskoe" ...
##  $ raion_popul                           : int  155572 115352 101708 1
78473 108171 43795 57405 155572 96959 142462 ...
##  $ children_preschool                    : int  9576 6880 5879 13087 5
706 2418 2459 9576 6507 9347 ...
##  $ school_education_centers_raion        : int  5 8 7 10 9 3 5 5 6 8 .
..
##  $ healthcare_centers_raion              : int  1 1 1 1 4 0 3 1 4 0 ..
.
##  $ sport_objects_raion                   : int  7 6 5 17 25 7 17 7 7 5
...
```

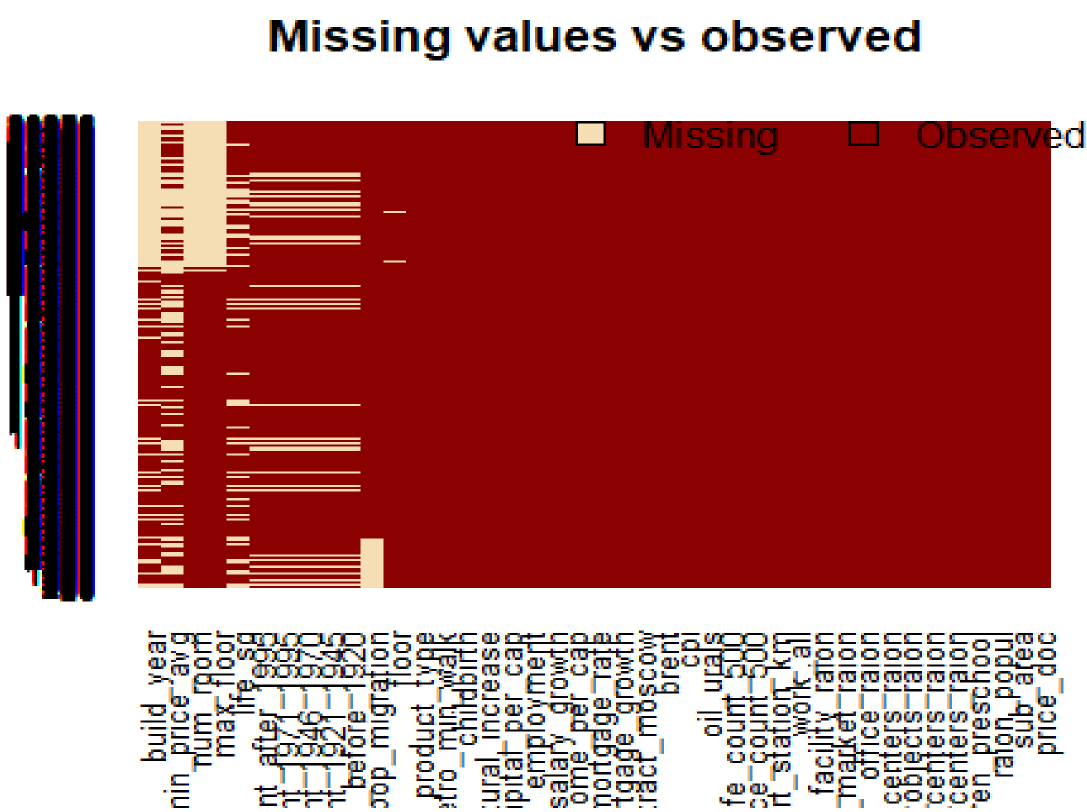
```
##  $ shopping_centers_raion                  : int  16 3 0 11 10 6 6 16 0
3 ...
##  $ office_raion                            : int  1 0 1 4 93 19 9 1 7 3
...
##  $ big_market_raion                        : int  1 1 1 1 1 1 1 1 1 1 ..
.
##  $ detention_facility_raion                : int  1 1 1 1 1 1 2 1 1 1 ..
.
##  $ work_all                                : int  98207 70194 63388 1203
81 68043 29660 35003 98207 59120 85551 ...
##  $ build_count_before_1920                 : int  0 1 1 13 371 0 11 0 1
47 ...
##  $ build_count_1921_1945                   : int  0 1 0 24 114 5 38 0 9
88 ...
##  $ build_count_1946_1970                   : int  0 143 246 40 146 152 9
0 0 290 413 ...
##  $ build_count_1971_1995                   : int  206 84 63 130 62 25 58
206 39 94 ...
##  $ build_count_after_1995                  : int  5 15 20 252 53 6 19 5
51 96 ...
##  $ metro_min_walk                          : num  13.58 7.62 17.35 11.57
8.27 ...
##  $ public_transport_station_km             : num  0.27 0.07 0.33 0.13 0.
07 0.19 0.05 0.25 0.22 0.22 ...
##  $ office_count_500                        : int  0 0 0 0 15 5 3 0 0 1 .
..
##  $ cafe_count_500                          : int  0 5 3 2 48 7 2 4 7 2 .
..
##  $ cafe_sum_500_min_price_avg              : num  NA 860 667 1000 702 ..
.
##  $ oil_urals                               : num  109 109 109 111 111 ..
.
##  $ cpi                                     : num  354 354 354 353 353 ..
.
##  $ brent                                   : num  109 109 111 114 110 ..
.
##  $ average_provision_of_build_contract_moscow: num  6.74 6.74 6.74 6.74 6.
74 6.74 6.74 6.74 6.74 6.74 ...
##  $ mortgage_growth                         : num  1.05 1.05 1.05 1.05 1.
05 1.05 1.05 1.05 1.05 1.05 ...
##  $ mortgage_rate                           : num  11.8 11.8 11.8 11.9 11
.9 ...
##  $ income_per_cap                          : num  42689 42689 42689 4031
1 40311 ...
##  $ salary_growth                           : num  0.17 0.17 0.17 0.17 0.
17 0.17 0.17 0.17 0.17 0.17 ...
##  $ employment                              : num  0.71 0.71 0.71 0.71 0.
71 0.71 0.71 0.71 0.71 0.71 ...
##  $ invest_fixed_capital_per_cap            : num  73976 73976 73976 7397
6 73976 ...
```

```
##  $ pop_natural_increase                      : num  1.1 1.1 1.1 1.1 1.1 1.
1 1.1 1.1 1.1 1.1 ...
##  $ pop_migration                             : num  5.1 5.1 5.1 5.1 5.1 5.
1 5.1 5.1 5.1 5.1 ...
##  $ childbirth                                : num  10.8 10.8 10.8 10.8 10
.8 10.8 10.8 10.8 10.8 10.8 ...
```

Step 2: Exploring and preparing the data

- Using Amelia to make a graph to show the missing values in each features of the data set
- Build year and number of room features have the most missing values in them.

```
library(Amelia)

## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.4, built: 2015-12-05)
## ## Copyright (C) 2005-2017 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

missmap(house, main = "Missing values vs observed")
```

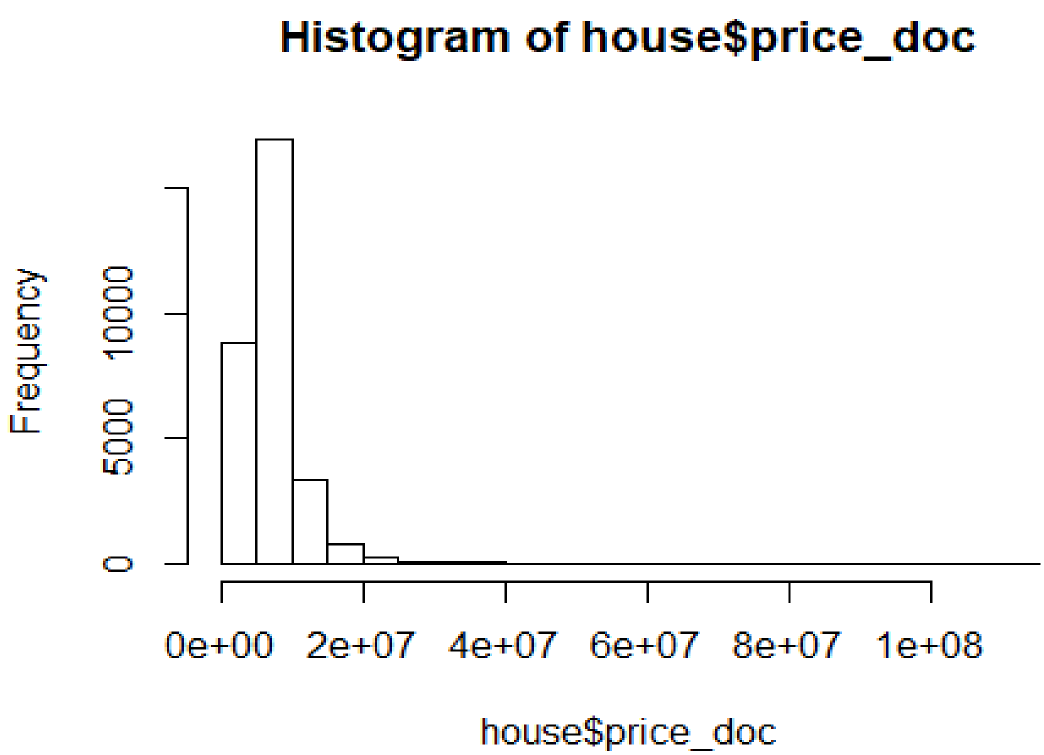


- The mean house price is 7,123,000 Rubles and the median is 6,274,000 Rubles

```
summary(house$price_doc)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##    100000   4740000   6274000   7123000   8300000 111100000
```

- We have a right skewed house price here. Most houses are 2 million Rubles or less.

```
hist(house$price_doc)
```



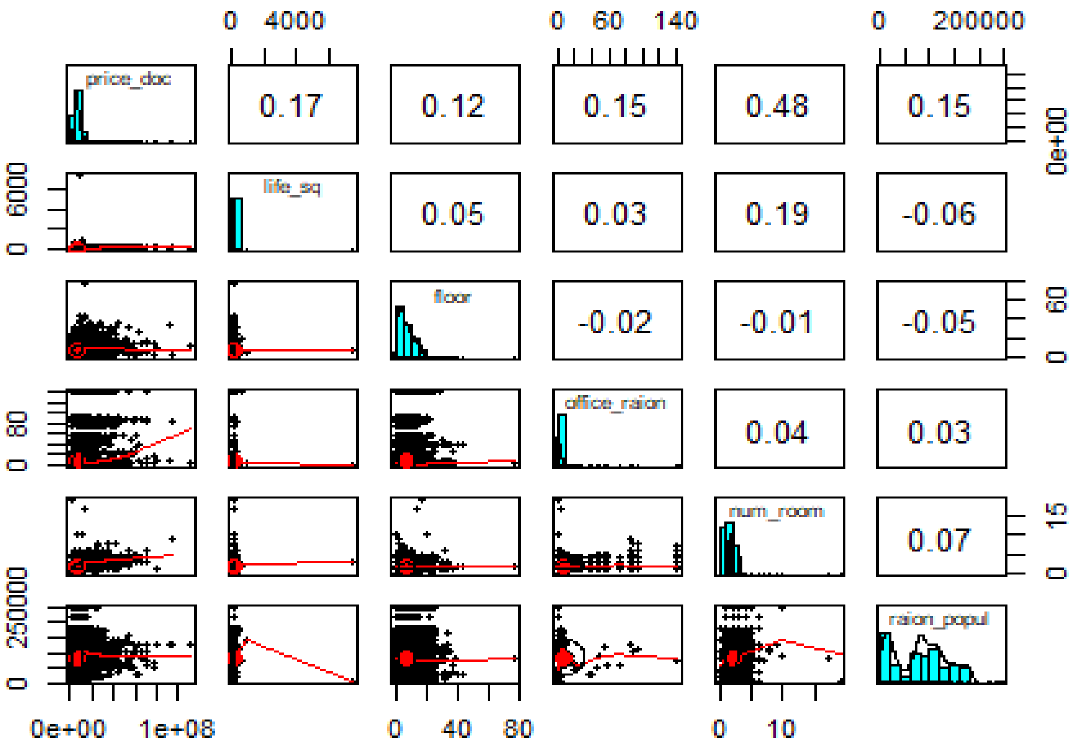
- Here we make a scatter plot matrix of some of the key features here.

- life_sq (living area in square meters) doesn't have any correlation with house price. That's surprising.

- We see some correlation with raion_popul(Number of municipality population) and housing price

- Number of rooms also has some correlation with house prices

```
library(psych)
pairs.panels(house[c("price_doc", "life_sq", "floor", "office_raion", "num_ro
om", "raion_popul")])
```



- We want to look at the which district in Russia has the highest housing price on average. Begovoe district has the highest housing price in Russia which is 9,266,826 Rubles.

```
sub_area <- aggregate(data = house, price_doc ~ sub_area, FUN= mean, na.rm =
TRUE)

sub_area[order(-sub_area$price_doc),]

##                        sub_area  price_doc
## 9                       Begovoe    9266826
## 99        Poselenie Rjazanovskoe    8593104
## 141          Vostochnoe Izmajlovo    8457061
## 121                        Silino    8328731
## 91           Poselenie Kokoshkino    7998453
## 46                       Koptevo    7739712
## 133               Troickij okrug    7704206
## 108                  Presnenskoe    7693557
## 106      Poselenie Voskresenskoe    7605872
## 23                   Dmitrovskoe    7448782
## 71            Nagatino-Sadovniki    7430128
## 16                     Butyrskoe    7385972
## 1                                   7385355
## 143               Zamoskvorech'e    7376009
```

```
## 79                    Obruchevskoe   7352763
## 115                       Severnoe   7280618
## 60              Losinoostrovskoe   7232661
## 38                    Jaroslavskoe   7214119
## 145                    Zjablikovo   7206004
## 47              Kosino-Uhtomskoe   7144493
## 6                            Arbat   7107643
## 49                  Krasnosel'skoe   7085829
## 76              Novo-Peredelkino   7079478
## 13             Birjulevo Zapadnoe   7059588
## 61                  Mar'ina Roshha   7033991
## 62                         Mar'ino   6936741
## 74                      Nekrasovka   6920292
## 43                  Juzhnoportovoe   6904985
## 93         Poselenie Marushkinskoe   6900000
## 21            Chertanovo Severnoe   6894724
## 119             Severnoe Tushino   6852733
## 134            Troparevo-Nikulino   6816179
## 54                        Kuz'minki   6794085
## 132                 Timirjazevskoe   6792249
## 10                 Beskudnikovskoe   6779177
## 127                       Strogino   6758100
## 114                   Savelovskoe   6737808
## 26                  Filevskij Park   6722437
## 56                   Levoberezhnoe   6708672
## 116               Severnoe Butovo   6692311
## 40                  Juzhnoe Butovo   6681357
## 20             Chertanovo Juzhnoe   6667818
## 109          Prospekt Vernadskogo   6640178
## 33                    Horoshevskoe   6624657
## 101         Poselenie Shhapovskoe   6602292
## 135                       Tverskoe   6586396
## 39                        Jasenevo   6578843
## 102          Poselenie Shherbinka   6557359
## 70                      Mozhajskoe   6541518
## 77                     Novogireevo   6523794
## 103           Poselenie Sosenskoe   6510300
## 73                        Nagornoe   6508392
## 125                       Solncevo   6497285
## 128                       Sviblovo   6495168
## 111                     Rjazanskij   6475872
## 146                        Zjuzino   6466473
## 67                          Mitino   6454794
## 105         Poselenie Voronovskoe   6450000
## 2                        Ajeroport   6404689
## 87          Pokrovskoe Streshnevo   6387684
## 42                 Juzhnoe Tushino   6374561
## 48                        Kotlovka   6368208
## 112                      Rostokino   6367314
## 59                    Lomonosovskoe   6354255
```

```
## 117              Severnoe Izmajlovo   6348972
## 58                         Ljublino   6322262
## 97       Poselenie Novofedorovskoe   6320251
## 12            Birjulevo Vostochnoe   6300021
## 22                      Danilovskoe   6296903
## 72               Nagatinskij Zaton   6282377
## 104           Poselenie Vnukovskoe   6267902
## 8                         Basmannoe   6247313
## 88          Poselenie Desjonovskoe   6218631
## 107                  Preobrazhenskoe   6199333
## 129                       Taganskoe   6171939
## 30                      Golovinskoe   6163126
## 37                        Jakimanka   6144016
## 120                        Shhukino   6142038
## 31                        Hamovniki   6139108
## 95            Poselenie Moskovskij   6138587
## 32              Horoshevo-Mnevniki   6136587
## 53                          Kurkino   6122333
## 15                         Brateevo   6118919
## 52                          Kuncevo   6111526
## 64                       Matushkino   6107787
## 130                  Tekstil'shhiki   6105916
## 65                      Meshhanskoe   6105259
## 123                       Sokol'niki   6086076
## 27                   Fili Davydkovo   6074200
## 11                         Bibirevo   6071983
## 29                       Gol'janovo   6067371
## 144              Zapadnoe Degunino   6048557
## 138                      Vojkovskoe   6043588
## 136                       Veshnjaki   6019097
## 35                       Ivanovskoe   6018833
## 36                        Izmajlovo   6016048
## 19          Chertanovo Central'noe   5989869
## 7                     Babushkinskoe   5987227
## 82       Orehovo-Borisovo Severnoe   5986417
## 41              Juzhnoe Medvedkovo   5979380
## 50                         Krjukovo   5959692
## 139                      Vostochnoe   5953399
## 118            Severnoe Medvedkovo   5952239
## 113                          Savelki   5936176
## 34                           Hovrino   5909950
## 84                         Otradnoe   5900970
## 25                     Dorogomilovo   5846723
## 44                         Kapotnja   5837031
## 24                          Donskoe   5811014
## 110                         Ramenki   5806117
## 96            Poselenie Mosrentgen   5800080
## 90              Poselenie Kievskij   5800000
## 17                         Caricyno   5774343
## 14                      Bogorodskoe   5752054
```

```
## 131                    Teplyj Stan   5728087
## 4                     Alekseevskoe   5722655
## 45                        Kon'kovo   5717005
## 57                       Lianozovo   5708022
## 55                       Lefortovo   5706464
## 80            Ochakovo-Matveevskoe   5645195
## 98            Poselenie Pervomajskoe  5642599
## 28                     Gagarinskoe   5637420
## 137                        Vnukovo   5636153
## 81          Orehovo-Borisovo Juzhnoe  5598775
## 142               Vyhino-Zhulebino   5592640
## 78                      Novokosino   5576342
## 83                    Ostankinskoe   5572387
## 63                         Marfino   5561406
## 3                     Akademicheskoe  5552574
## 85                      Pechatniki   5550180
## 69             Moskvorech'e-Saburovo  5541487
## 51                      Krylatskoe   5536540
## 86                          Perovo   5519374
## 5                       Altuf'evskoe  5506579
## 140             Vostochnoe Degunino   5441596
## 66                      Metrogorodok  5429572
## 89          Poselenie Filimonkovskoe  5351575
## 124                 Sokolinaja Gora   5333290
## 122                          Sokol   5285504
## 18                     Cheremushki   5017812
## 126                 Staroe Krjukovo   4792706
## 75                   Nizhegorodskoe   4500011
## 68                   Molzhaninovskoe  4000000
## 94  Poselenie Mihajlovo-Jarcevskoe   3700000
## 100            Poselenie Rogovskoe   3545000
## 92        Poselenie Krasnopahorskoe   2200000
```

Step 3 Training a model on the data

- Here we will apply multiple regressions model. We will use all the features in the data set.

```
set.seed(2)

ins_model <- lm(price_doc ~ life_sq+    floor+  max_floor+  build_year+ num_r
oom+   raion_popul+    children_preschool+ school_education_centers_raion+ he
althcare_centers_raion+   sport_objects_raion+    shopping_centers_raion+ off
ice_raion+   big_market_raion+   detention_facility_raion+   work_all+   buil
d_count_before_1920+    build_count_1921_1945+  build_count_1946_1970+  build
_count_1971_1995+  build_count_after_1995+ metro_min_walk+ public_transport_s
tation_km+    office_count_500+   cafe_count_500+ cafe_sum_500_min_price_avg+
oil_urals+  cpi+    brent+  average_provision_of_build_contract_moscow+ mortg
age_growth+    mortgage_rate+  income_per_cap+ salary_growth, data = house)
```

Step 4 Evaluating model performance

- With our multiple regressions model, our p-value is significant but our adjusted R-squared is 0.4135991 which is quite low.
- I didn't include here, but I did try removing the insignificant features out and the result still didn't improve much.
- In the next step, I will use Random Forests and Neural Network to apply to this data sets to see if there is any improvement.

```
summary(ins_model)

##
## Call:
## lm(formula = price_doc ~ life_sq + floor + max_floor + build_year +
##     num_room + raion_popul + children_preschool + school_education_centers
_raion +
##     healthcare_centers_raion + sport_objects_raion + shopping_centers_raio
n +
##     office_raion + big_market_raion + detention_facility_raion +
##     work_all + build_count_before_1920 + build_count_1921_1945 +
##     build_count_1946_1970 + build_count_1971_1995 + build_count_after_1995
+
##     metro_min_walk + public_transport_station_km + office_count_500 +
##     cafe_count_500 + cafe_sum_500_min_price_avg + oil_urals +
##     cpi + brent + average_provision_of_build_contract_moscow +
##     mortgage_growth + mortgage_rate + income_per_cap + salary_growth,
##     data = house)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -64162559  -1451545    222824   1531969  62565480
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                -2.173e+06  5.684e+06  -0.382
## life_sq                                     9.596e+04  3.142e+03  30.540
## floor                                       4.499e+04  1.074e+04   4.191
## max_floor                                   1.645e+05  9.159e+03  17.959
## build_year                                  2.273e-01  2.180e-01   1.043
## num_room                                    1.681e+06  6.496e+04  25.873
## raion_popul                                 1.028e+02  1.245e+01   8.263
## children_preschool                         -2.502e+02  5.081e+01  -4.925
## school_education_centers_raion              1.488e+05  2.559e+04   5.814
## healthcare_centers_raion                    1.706e+05  3.463e+04   4.925
## sport_objects_raion                         1.125e+05  1.329e+04   8.468
## shopping_centers_raion                     -7.228e+04  1.607e+04  -4.498
## office_raion                               -1.281e+04  5.445e+03  -2.353
## big_market_raion                           -1.024e+05  2.434e+05  -0.421
## detention_facility_raion                   -7.260e+05  1.533e+05  -4.736
## work_all                                   -1.395e+02  1.638e+01  -8.517
```

```
## build_count_before_1920                    3.759e+03  1.769e+03   2.125
## build_count_1921_1945                     -8.028e+02  1.793e+03  -0.448
## build_count_1946_1970                     -3.623e+03  6.174e+02  -5.868
## build_count_1971_1995                     -1.380e+04  1.910e+03  -7.221
## build_count_after_1995                     1.532e+03  8.336e+02   1.838
## metro_min_walk                            -1.296e+04  1.274e+03 -10.169
## public_transport_station_km               -1.652e+06  5.143e+05  -3.212
## office_count_500                           2.739e+05  2.882e+04   9.503
## cafe_count_500                            -3.331e+04  7.517e+03  -4.432
## cafe_sum_500_min_price_avg                 4.810e+02  1.392e+02   3.455
## oil_urals                                 -1.182e+04  1.905e+04  -0.620
## cpi                                        1.793e+04  8.741e+03   2.052
## brent                                      5.581e+02  1.844e+04   0.030
## average_provision_of_build_contract_moscow -7.489e+05  3.311e+05  -2.262
## mortgage_growth                            7.456e+05  5.412e+05   1.378
## mortgage_rate                              3.125e+04  1.945e+05   0.161
## income_per_cap                             1.790e+00  5.848e+00   0.306
## salary_growth                              3.914e+06  6.407e+06   0.611
##                                            Pr(>|t|)
## (Intercept)                                0.702213
## life_sq                                     < 2e-16 ***
## floor                                      2.80e-05 ***
## max_floor                                   < 2e-16 ***
## build_year                                 0.297132
## num_room                                    < 2e-16 ***
## raion_popul                                 < 2e-16 ***
## children_preschool                         8.59e-07 ***
## school_education_centers_raion             6.28e-09 ***
## healthcare_centers_raion                   8.57e-07 ***
## sport_objects_raion                         < 2e-16 ***
## shopping_centers_raion                     6.92e-06 ***
## office_raion                               0.018647 *
## big_market_raion                           0.673976
## detention_facility_raion                   2.21e-06 ***
## work_all                                    < 2e-16 ***
## build_count_before_1920                    0.033571 *
## build_count_1921_1945                      0.654443
## build_count_1946_1970                      4.54e-09 ***
## build_count_1971_1995                      5.53e-13 ***
## build_count_after_1995                     0.066156 .
## metro_min_walk                              < 2e-16 ***
## public_transport_station_km                0.001322 **
## office_count_500                            < 2e-16 ***
## cafe_count_500                             9.44e-06 ***
## cafe_sum_500_min_price_avg                 0.000552 ***
## oil_urals                                  0.535079
## cpi                                        0.040233 *
## brent                                      0.975852
## average_provision_of_build_contract_moscow 0.023718 *
## mortgage_growth                            0.168313
```

```
## mortgage_rate                                      0.872374
## income_per_cap                                     0.759546
## salary_growth                                      0.541219
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4367000 on 9958 degrees of freedom
##   (20479 observations deleted due to missingness)
## Multiple R-squared:  0.4155, Adjusted R-squared:  0.4136
## F-statistic: 214.5 on 33 and 9958 DF,  p-value: < 2.2e-16
```

Step 5 Improving model performance

Improve the prediction with Random Forests

- For Random Forests to work I have to address the missing value in the data set.
- I chose the easy to way to handle the missing values by removing them from the data set and set 80% to training and 20% to testing.

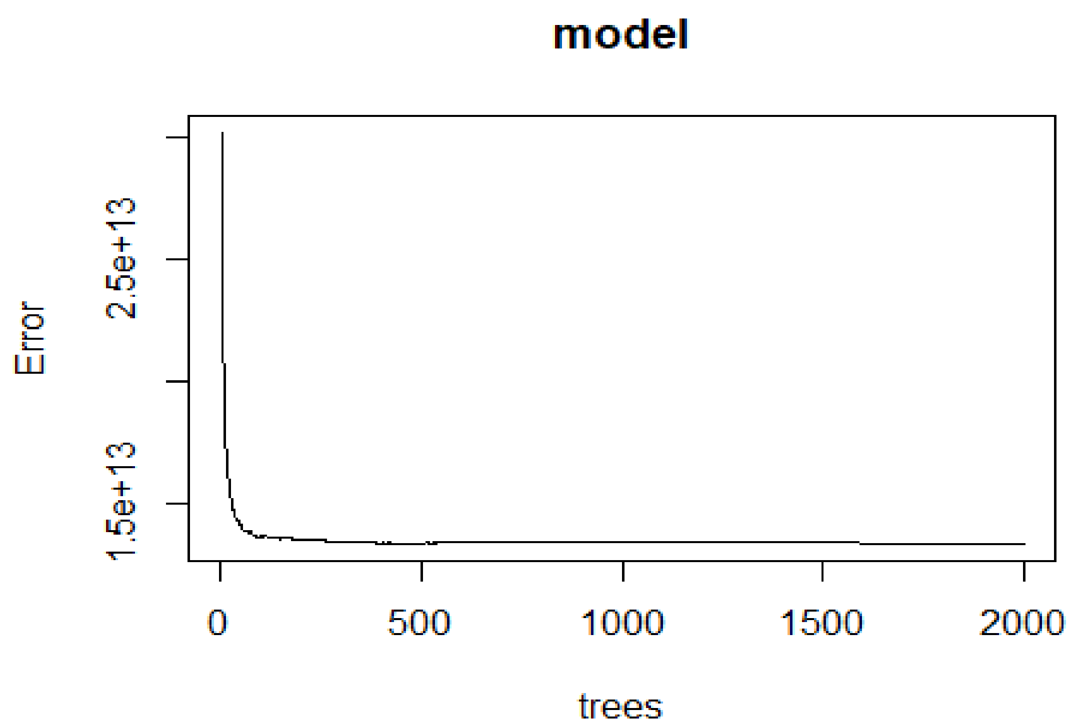
```
house1 <- na.omit(house)
set.seed(12)
samp <- sample(nrow(house1), 0.8 * nrow(house1))
train <- house1[samp, ]
test <- house1[-samp, ]
```

- Now I'll apply the Random Forests to the data set. The model seems to do better without including the factor features sub_area and product_type.

```
library(randomForest)

## randomForest 4.6-12

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:psych':
##
##     outlier

set.seed(12)
model <- randomForest(price_doc ~. -sub_area -product_type, data = train, ntr
ee=2000, mtry=30)
```

- Here we use 2000 trees and 30 splits.
- Type of random forest is regression.
- According to the graph, it seems like we won’t reduce any more error as we increase the trees over 2000 tress.

```
plot(model)
```




```
model

##
## Call:
##  randomForest(formula = price_doc ~ . - sub_area - product_type,      data
= train, ntree = 2000, mtry = 30)
##                Type of random forest: regression
##                      Number of trees: 2000
## No. of variables tried at each split: 30
##
##           Mean of squared residuals: 1.330471e+13
##                     % Var explained: 56.82
```

- Now let's see the correlation with our model to the test data set.
- Our accuracy here is 0.7713 which is not bad and certainly better than the multiple regressions model.

```
pred <- predict(model, newdata = test)

cor(pred, test$price_doc)

## [1] 0.771367
```

Improve the model with Neural Network

- Now we will apply Neural Network on this data set

- First, we need to normalize our data set. In here we exclude the factor features out because we can't normalize the data set with these features in the data set.

```
house2 <- house[-7:-8]
house2 <- as.data.frame(lapply(house2, scale))
```

- To validate that the data set is scaled properly

```
summary(house2$price_doc)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.4690 -0.4985 -0.1775  0.0000  0.2462 21.7500
```

- Again, we remove all the missing value from the data set and then randomize the data set and then set 80% to training and 20% to testing.

```
set.seed(10)
house2 <- na.omit(house2)
samp <- sample(nrow(house2), 0.8 * nrow(house2))
train2 <- house2[samp, ]
test2 <- house2[-samp, ]
```

- Now we are ready to apply the Neural Network algorithm to the data set with logistic activation function.
- We use all the features here except the ones we excluded above.

```
library(neuralnet)
set.seed(5)
model <- neuralnet(formula = price_doc ~
life_sq+    floor+  max_floor+  build_year+ num_room+    raion_popul+    child
ren_preschool+ school_education_centers_raion+ healthcare_centers_raion+    sp
ort_objects_raion+    shopping_centers_raion+ office_raion+    big_market_raio
n+   detention_facility_raion+    work_all+    build_count_before_1920+     buil
d_count_1921_1945+  build_count_1946_1970+  build_count_1971_1995+  build_cou
nt_after_1995+ metro_min_walk+ public_transport_station_km+    office_count_5
00+    cafe_count_500+ cafe_sum_500_min_price_avg+ oil_urals+  cpi+     brent+
average_provision_of_build_contract_moscow+ mortgage_growth+    mortgage_rate
+  income_per_cap+ salary_growth+  employment+ invest_fixed_capital_per_cap+
pop_natural_increase+   pop_migration+  childbirth, data = train2, hidden=1,
threshold = 0.01, stepmax = 1e+05, act.fct = "logistic")
```

- Now we apply the model to the test data set and look at the correlation to see how well our model perform.
- The accuracy is about 69% which is not bad but the Random Forests did better

```
model_results <- compute(model, test2[2:39])

predicted_strength <- model_results$net.result

cor(predicted_strength, test2$price_doc)

##              [,1]
## [1,] 0.6949942337
```

What have I learned?

- I would like to see my accuracy to be above 85%. The best I got was 77.13% from Random Forests. I'm still surprised by how well the Random Forests and Neural Network algorithm perform. They do much better than the traditional multiple regressions models.

- I learned that there are many parameters that we can tweak in the package for Random Forests and Neural Networks. I could spend all day tweaking it around and probably improve my accuracy 3-5% more, I hope.

- I also learned that these algorithms require tremendous processing power. I tried to set hidden node on the Neural Network to 5 and left it overnight and when I woke up, it's still running so I gave up. It only works when I set the hidden node to 1 because more than that would gave me error or take a long time to process. What I can do in the future is reduce my features down and/or maybe learn how to use Amazon computing cloud to process this.

- The hardest parts about this is preparing the data set for it to work with the algorithms and understand which parameters to tweak to improve the model.