Stats2 SBERBANK Housing Price Prediction

Hao Wang, Billy Nayden, Muchigi Kimari

1. Goal 1: Prediction of Individual Property Values

1.1. Introduction

Housing costs demand a significant investment from both consumers and developers, and when it comes to planning a budget, whether personal or corporate, the last thing anyone needs is uncertainty about one of their biggest expenses. Sberbank, Russia's oldest and largest bank, helps their customers by making predictions about realty prices, so renters, developers, and lenders are more confident when they sign a lease or purchase a building. In this project, we developed models which use a broad spectrum of features to predict realty prices.

Our first goal is to build a useful model to predict the price of an individual property given some of the variables in the the data set provided by Sberbank. We will make predictions using the model we developed on 5000 properties. Our goal is to derive a model that minimizes the RMSLE (Root Mean Squared Logarithmic Error).

1.2. Data Description

The data we are used to build the model contained 25,471 observations across 73 variables. Of the 73 variables, 69 variables were continuous or numeric variables, three were categorical or factor variables, and one was our response variable price (**Figure 1.1**). 23 of these variables contained missing values, which we will need to address, should we choose that variable for our model (**Figure 1.2**). All variables containing missing values are continuous. We did all initial data set exploration in R.

1.3. Data Wrangling and Cleaning

We did not need to rename or standardize any of the values of our data set for this particular problem, but went through intense cleaning of missing values, which is described in Section 1.4.

However, we did change three categorical variables in the data set into integers, in order to potentially use them for modeling (**Figure 1.3**).

1.4. Exploratory Data Analysis (EDA)

In order to simplify our variable selection process and analysis, we intuitively divided the data set into nine distinct groups based on their attributes (**Table 1.1**).

We used this grouping to build correlation charts for each of the groups, measured against the response group of price. Thus, we were able to analyze among the distance (**Figures 1.4 and 1.5**), number (**Figure 1.6**), count (**Figure 1.7**), population (**Figure 1.8**), time (**Figure 1.9**), area (**Figure 1.10**), other (**Figure 1.11**), and ID (**Figure 1.12**) groups most closely correlated with price.

This correlation analysis allowed us to select five continuous variables for further analysis: kremlin_km, num_room, X0_17_all, build_year, and full_sq.

1.4.1. Outlier Identification and Handling

The only variable we identified within these five variables with significant outliers is build_year (**Figure 1.13**). However, because there were a relatively limited number of outliers compared to the overall data set, we decided to remove these observations from our model's training set, to satisfy the time constraints of this project, and because we did not feel it would materially affect our results.

1.4.2. Missing Value Identification, Summary, and Possible Imputation

Among the five continuous variables we investigated, two had missing values: num_room and build_year.

We decided to address num_room first. First, we used a correlation plot to determine which variables to use as predictors (**Figure 1.14**). This helped us settle on full_sq and life_sq as predictors. However, only 2,658 of the missing values in num_room can be predicted by modeling, because there are 5,333 missing values in life_sq (**Figure 1.15**).

Using full_sq and life_sq as predictors, we built a k-nearest neighboy (KNN) model to predict the missing values in the num_room data set (**Figure 1.16**). We then ran the model, and produced a new data set with zero missing values for num_room (**Figure 1.17**).

Second we addressed build_year. Because there were no good predictors for build_year, we removed the missing values to ensure we had a clean data set (**Figure 1.18**).

1.4.3. Multicollinearity

We checked multicollinearity in SAS and the plots from the proc sgscatter procedue (**Figure 1.19**) did not demonstrate any evidence of multicollinearity between the variables we selected.

1.4.4. Checking Assumptions (Check assumption on OLS and LASSO specifically)

We checked our assumptions in SAS using proc reg (**Figure 1.20**) and found that the study met all assumptions. The example provided shows the response variable price_doc measured against the exploratory variable full_sq, though we tested all exploratory variables.

Both the plots and the tables provided by the proc reg procedure (**Tables 1.2 and 1.3**) demonstrated the residuals are roughly normally distributed around zero, with a constant standard deviation.

Additionally, the number of observations in this study are large enough to utilize the Central Limit Theorem. The histograms in Figure 1.19 also support this.

1.4.5. Variable Selection

With all of the data wrangling and cleansing, the addressing of missing values and outliers, and the checking of assumptions completed, we finally established a clean data set that is ripe for modeling (**Figure 1.21**).

We ultimately selected the five continuous variables listed above, as well as the three categorical variables product_type, railroad_terminal_raion, and big_market_raion. We will use our SAS modeling to assist with variable selection.

1.5. Modeling

The three models we chose were the LASSO model, the OLS stepwise model, and an OLS stepwise model with interaction variables. We used SAS for all modeling (**Figure 1.22**).

1.5.1. LASSO Model

The LASSO model selected seven of the eight exploratory variables, excluding railroad_terminal_raion. The model is as follows (**Table 1.4**):

Price_doc = 28065501 + 196744(full_sq) - 12920(build_year) - 844309(num_room) 14.755952(X0_17_all) - 166444(kremlin_km) - 315062(product_type) - 821012(big_market_raion)

1.5.2. OLS Stepwise Model

The OLS stepwise model used the same seven variables as the LASSO model. The model is as follows (**Table 1.5**):

Price_doc = 28919545 + 197710(full_sq) - 13351(build_year) - 866209(num_room) + 15.083109(X0_17_all) - 166455(kremlin_km) - 315947(product_type) - 830030(big_market_raion)

1.5.3. OLS Stepwise Model with Interaction Variables

The OLS stepwise model uses the same variables as the above variables, along with an interaction variable of all seven. The model is as follows (**Table 1.6**):

Price_doc = 32356910 + 200778(full_sq) - 15541(build_year) - 822448(num_room) + 27.597161(X0_17_all) - 159224(kremlin_km) - 687817(big_market_raion) - 0.000002418(interaction_variable)

1.5.4. Model Comparison

In order to evaluate the models from SAS we compared them across the following statistics: Root MSE, Dependent Mean, R^2, Adjusted R^2, AIC, AICC, SBC, ASE, and CV Press (**Table 1.7**).

Additionally, we created a linear model in R, and cross-validated with our SAS model (**Figure 1.23**).

1.6. Prediction

Our predictions are included in the attached CSV file in the ZIP folder with this submission.

1.6.1. Selecting the Best Model

Ultimately, we selected the OLS model with the interaction variables because it had a similar Adjusted R^2 to the other two models, but had a lower Root MSE. (Must mention ASE (test) and use this as justification for the best model you select)

2. Goal 2: Prediction of Mean Property Value by Year

2.1. Introduction

Our second goal is to build a useful model to predict the mean price of properties from July 2015 to July 2016. We would also like to provide a table and a plot of these forecasts complete with 95% confidence intervals. This analysis will entail extensive data wrangling, as well as testing for serial correlation in the data to make necessary adjustments to the model if there is evidence of significant serial correlation.

2.2. Data Wrangling

We used R for all data wrangling.

First we selected the two relevant columns from the data set, timestamp and price_doc.

Second, we changed the timestamp class from a factor to a date.

Third, we divided the timestamp into three columns: year, month, and day.

Fourth, we combined year and month into a new column called monthyear.

Fifth, we grouped the data by monthyear and average price.

Finally, we added a new column called MonthNumber (**Figure 2.1**).

2.3. Plot of Time Series

We used R and the package ggplot2, along with the columns price_doc and MonthNumber from our modified data set to graph the time series of the price over number of months (**Figure 2.2**).

2.4. Model of Residual Series

We used R and SAS to make all of the following plots and models.

2.4.1. Linear Regression Model

We used R to make a linear regression model with price_doc as the response variable. The model is as follows (**Figure 2.3**):

Price_doc = 5824652 + 45969(MonthNumber)

2.4.2. Plot of Residual Series

We used R to plot the residuals of this model (**Figure 2.4**).

2.4.3. Autocorrelation Investigation

We used SAS and proc autoreg to investigate autocorrelation in our model (**FIgure 2.5**).

According to the results, the lowest AIC is 1340.68 for our lag(1) model, with a Durbin-Watson value of 2.1502, the highest of the three models, and a p-value of 0.6422. This suggests no signifcant evidence of an autocorrelation effect in the model. However, in the lag(2) model, the p-value is 0.0009, suggesting a significant autocorrelation effect. However, the parameter estimates in model lag(2) are less than those in model lag(1), meaning lag(2) is more stable. Among the three model, lag(1) and lag(2) have lower AIC and SBC than the OLS model (**Table 2.1**).

2.4.4. Forecast of Residuals for Next Year

We used SAS to forecast the residuals of the next year (**Figure 2.6 and Table 2.2**).

2.5. Forecast of Next Year

We used SAS to forecast the next year, and our forecast is in an attached CSV file in the ZIP folder for this project (**Figure 2.7 and Table 2.3**).

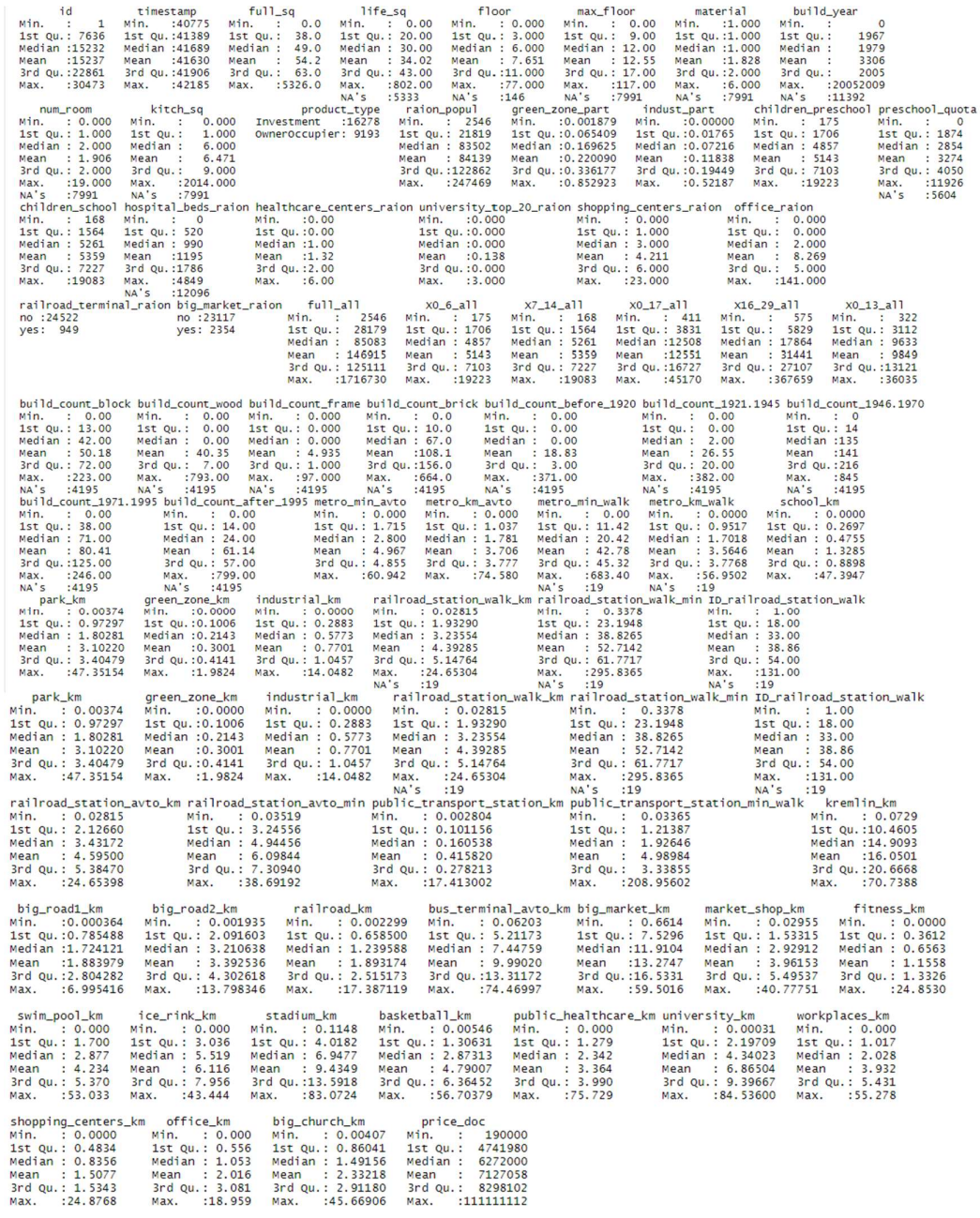
3. Appendix 1: Data Dictionary

Var_ID	Variable	Category	Description
1	id	ID	Transaction id
2	timestamp	Time	Date of transaction
3	full_sq	Area	Total area in square meters, including loggias, balconies and other
4	life_sq	Area	Living area in square meters, excluding loggias, balconies and other
5	floor	Other	For apartments, floor of the building
6	max_floor	Number	Number of floors in the building
7	material	Other	Wall material
8	build_year	Time	Year built
9	num_room	Number	Number of living rooms
10	kitch_sq	Area	Kitchen area
11	product_type	Other	Owner-occupier purchase or investment
12	raion_popul	Number	Number of municipality population. district
13	green_zone_part	Area	Proportion of area of greenery in the total area
14	indust_part	Area	Share of industrial zones in area of the total area
15	children_preschool	Number	Number of pre-school age population
16	preschool_quota	Number	Number of seats in pre-school organizations
17	children_school	Population	Population of school-age children
18	hospital_beds_raion	Number	Number of hospital beds for the district
19	healthcare_centers_raion	Number	Number of healthcare centers in district

20	university_top_20_raion	Number	Number of higher education institutions in the top ten ranking of the
21	shopping_centers_raion	Number	Number of malls and shopping centres in district
22	office_raion	Number	Number of offices in district
23	railroad_terminal_raion	Other	Presence of the railroad terminal in district
24	big_market_raion	Other	Presence of large grocery / wholesale markets
25	full_all	Population	Total population in the municipality
26	0_6_all	Population	Population aged 0-6
27	7_14_all	Population	Population aged 7-14
28	0_17_all	Population	Population aged 0-17
29	16_29_all	Population	Population aged 16-19
30	0_13_all	Population	Population aged 0-13
31	build_count_block	Count	Count of block buildings
32	build_count_wood	Count	Count of wood buildings
33	build_count_frame	Count	Count of frame buildings
34	build_count_brick	Count	Count of brick buildings
35	build_count_before_1920	Count	Count of before_1920 buildings
36	build_count_1921-1945	Count	Count of 1921-1945 buildings
37	build_count_1946-1970	Count	Count of 1946-1970 buildings
38	build_count_1971-1995	Count	Count of 1971-1995 buildings
39	build_count_after_1995	Count	Count of after_1995 buildings
40	metro_min_avto	Time	Time to subway by car, min.
41	metro_km_avto	Distance	Distance to subway by car, km
42	metro_min_walk	Time	Time to metro by foot
43	metro_km_walk	Distance	Distance to the metro, km
44	school_km	Distance	Distance to high school
45	park_km	Distance	Distance to park
46	green_zone_km	Distance	Distance to green zone
47	industrial_km	Distance	Distance to industrial zone
48	railroad_station_walk_km	Distance	Distance to the railroad station (walk)
49	railroad_station_walk_min	Time	Time to the railroad station (walk)
50	ID_railroad_station_walk	ID	Nearest railroad station id (walk)
51	railroad_station_avto_km	Distance	Distance to the railroad station (avto)
52	railroad_station_avto_min	Time	Time to the railroad station (avto)
53	public_transport_station_km	Distance	Distance to the public transport station (walk)
54	public_transport_station_min_	Time	Time to the public transport station (walk)
55	kremlin_km	Distance	Distance to the city center (Kremlin)
56	big_road1_km	Distance	Distance to Nearest major road
57	big_road2_km	Distance	Distance to next distant major road
58	railroad_km	Distance	Distance to the railway / Moscow Central Ring / open areas Underground
59	bus_terminal_avto_km	Distance	Distance to bus terminal (avto)
60	big_market_km	Distance	Distance to grocery / wholesale markets
61	market_shop_km	Distance	Distance to markets and department stores
62	fitness_km	Distance	Distance to fitness
63	swim_pool_km	Distance	Distance to swimming pool
64	ice_rink_km	Distance	Distance to ice palace
65	stadium_km	Distance	Distance to stadium
66	basketball_km	Distance	Distance to the basketball courts
67	public_healthcare_km	Distance	Distance to public healthcare
68	university_km	Distance	Distance to universities
69	workplaces_km	Distance	Distance to workplaces
70	shopping_centers_km	Distance	Distance to shopping centers
71	office_km	Distance	Distance to business centers/ offices
72	big_church_km	Distance	Distance to large church
73	price_doc	Response	sale price (this is the target variable)

4. Appendix 2: Code and Tables

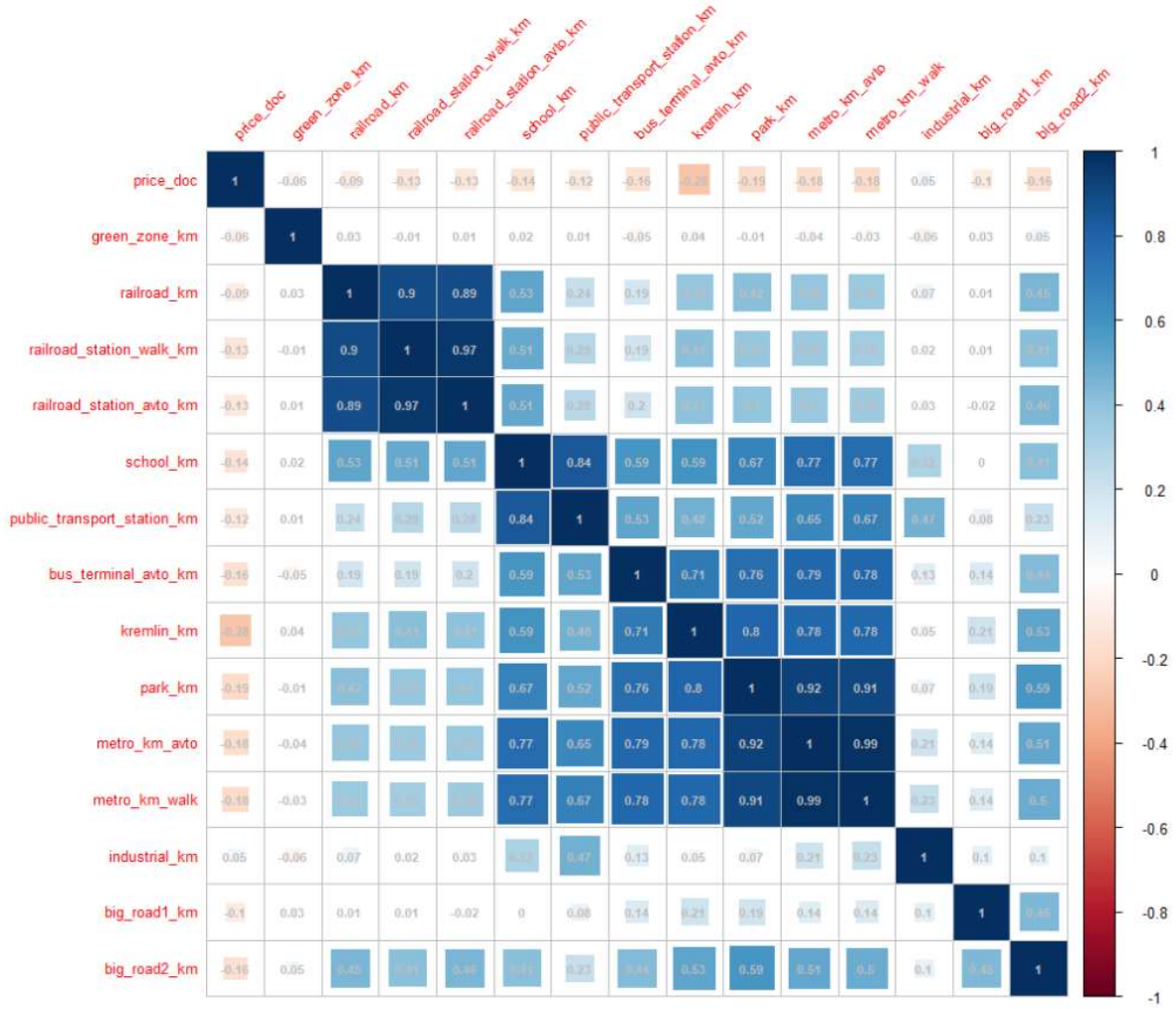
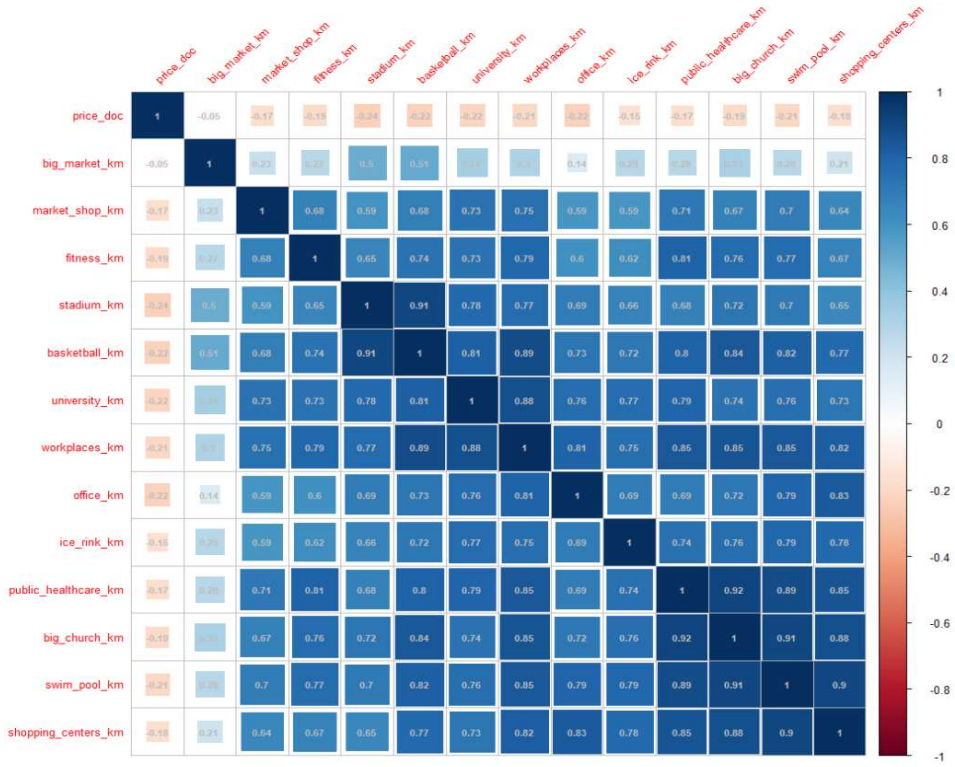
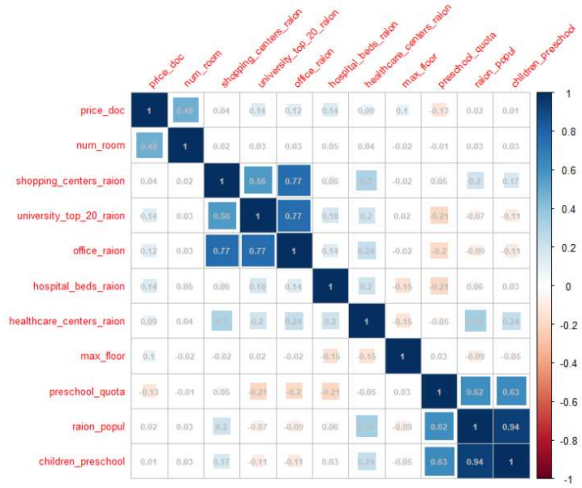
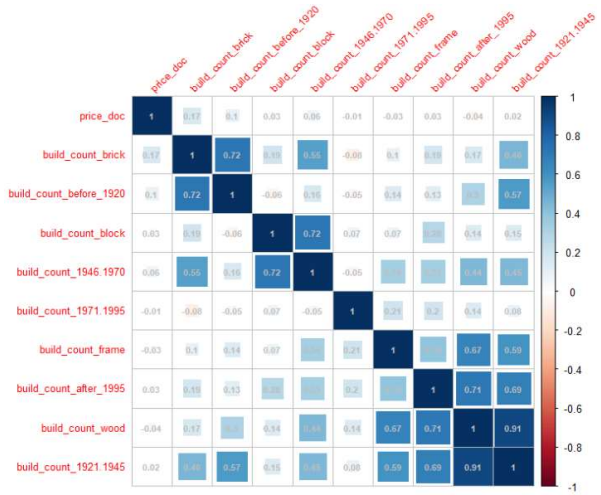
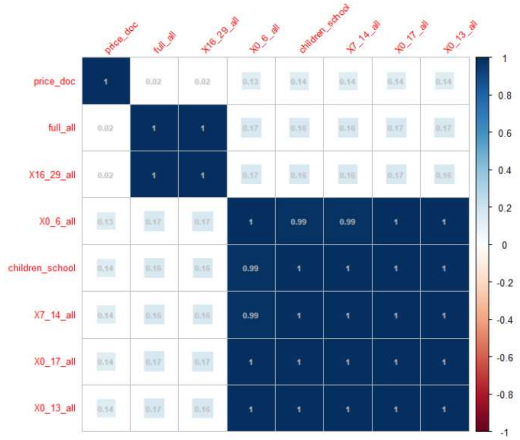
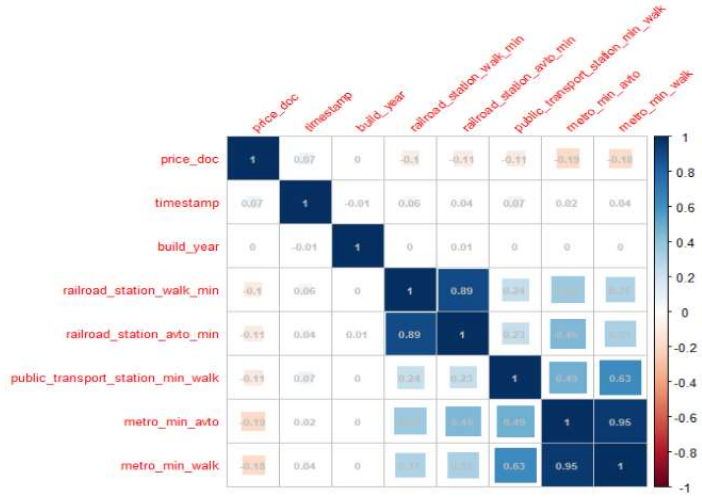
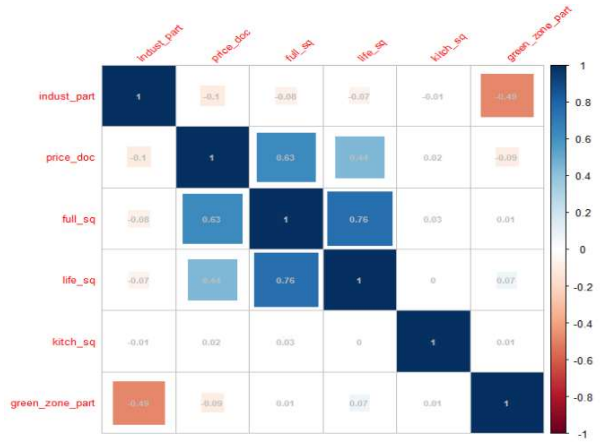
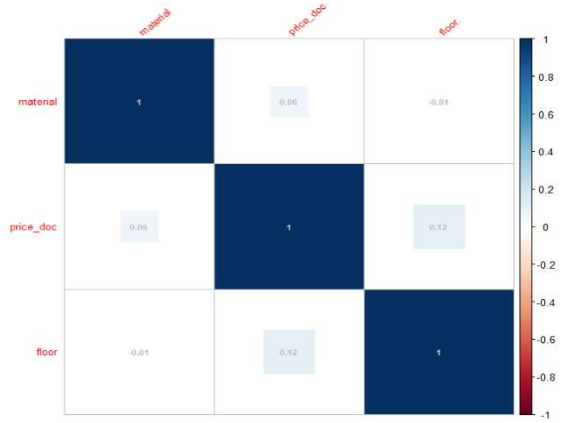
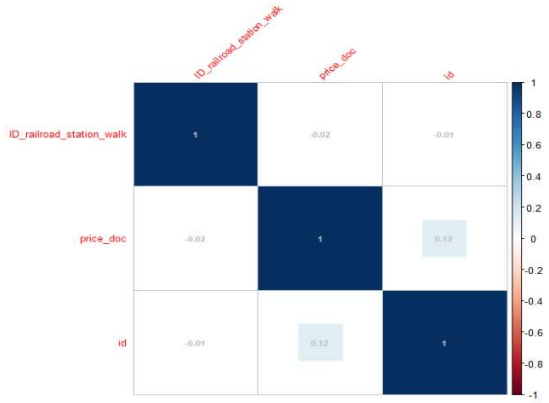
Figure 1.1

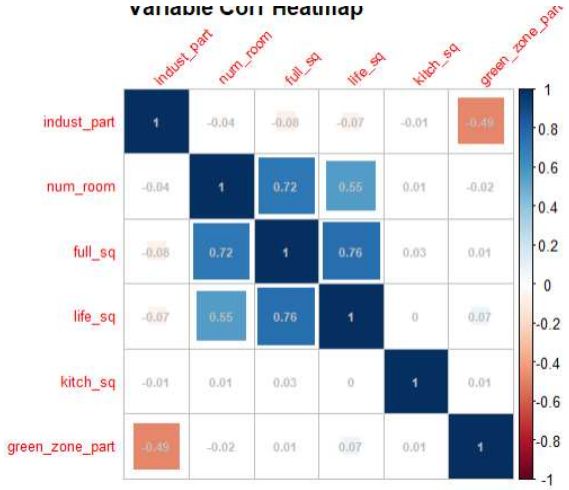
```
# Initial data profiling on house data set
```{r echo=TRUE}
str(HouseDF)
# 25471 obs. of 73 variables.
summary(HouseDF)
```
```

```
'data.frame':   25471 obs. of  73 variables:
 $ id                                : int  1 2 3 4 5 6 7 8 9 10 ...
 $ timestamp                         : int  40775 40778 40782 40787 40791 40792 40794 40795 40796 40799 ...
 $ full_sq                           : int  43 34 43 89 77 67 25 44 42 36 ...
 $ life_sq                           : int  27 19 29 50 77 46 14 44 27 21 ...
 $ floor                             : int  4 3 2 9 4 14 10 5 5 9 ...
 $ max_floor                         : int  NA NA NA NA NA NA NA NA NA NA ...
 $ material                          : int  NA NA NA NA NA NA NA NA NA NA ...
 $ build_year                        : int  NA NA NA NA NA NA NA NA NA NA ...
 $ num_room                          : int  NA NA NA NA NA NA NA NA NA NA ...
 $ kitch_sq                          : int  NA NA NA NA NA NA NA NA NA NA ...
 $ product_type                      : Factor w/ 2 levels "Investment","OwnerOccupier": 1 1 1 1 1 1 1 1 1 1 ...
 $ raion_popul                       : int  155572 115352 101708 178473 108171 43795 57405 155572 96959 142462 ...
 $ green_zone_part                   : num  0.1897 0.3726 0.1126 0.1947 0.0152 ...
 $ indust_part                       : num  0.00007 0.04964 0.11854 0.06975 0.03732 ...
 $ children_preschool                : int  9576 6880 5879 13087 5706 2418 2459 9576 6507 9347 ...
 $ preschool_quota                   : int  5001 3119 1463 6839 3240 852 933 5001 3272 4050 ...
 $ children_school                   : int  10309 7759 6207 13670 6748 2514 2810 10309 6566 9292 ...
 $ hospital_beds_raion               : int  240 229 1183 NA 562 NA 4849 240 1894 2620 ...
 $ healthcare_centers_raion          : int  1 1 1 1 4 0 3 1 4 0 ...
 $ university_top_20_raion           : int  0 0 0 0 2 0 0 0 0 0 ...
 $ 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 ...
 $ railroad_terminal_raion           : Factor w/ 2 levels "no","yes": 1 1 1 1 2 1 1 1 1 1 ...
 $ big_market_raion                  : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ full_all                          : int  86206 76284 101982 21155 28179 19940 85956 86206 78810 78507 ...
 $ X0_6_all                          : int  9576 6880 5879 13087 5706 2418 2459 9576 6507 9347 ...
 $ X7_14_all                         : int  10309 7759 6207 13670 6748 2514 2810 10309 6566 9292 ...
 $ X0_17_all                         : int  23603 17700 14884 32063 15237 5866 6510 23603 15510 22071 ...
 $ X16_29_all                        : int  17508 15164 19401 3292 5164 4851 19445 17508 17662 15929 ...
 $ X0_13_all                         : int  18654 13729 11252 24934 11631 4632 4884 18654 12269 17469 ...
 $ build_count_block                 : int  25 83 59 9 48 24 23 25 101 68 ...
 $ build_count_wood                  : int  0 1 0 51 0 0 1 0 2 204 ...
 $ build_count_frame                 : int  0 0 0 12 0 0 0 0 0 14 ...
 $ build_count_brick                 : int  0 67 206 124 643 147 139 0 216 237 ...
 $ 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 90 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_avto                    : num  2.59 0.937 2.121 1.489 1.257 ...
 $ metro_km_avto                     : num  1.131 0.647 1.638 0.985 0.877 ...
 $ metro_min_walk                    : num  13.58 7.62 17.35 11.57 8.27 ...
 $ metro_km_walk                     : num  1.131 0.635 1.446 0.964 0.689 ...
 $ school_km                         : num  0.178 0.273 0.158 0.236 0.377 ...
 $ park_km                           : num  2.1586 0.5507 0.3748 0.0781 0.2583 ...
 $ green_zone_km                     : num  0.601 0.0653 0.4532 0.1061 0.2362 ...
 $ industrial_km                     : num  1.081 0.966 0.939 0.451 0.393 ...
 $ railroad_station_walk_km          : num  5.42 3.412 1.278 4.291 0.854 ...
 $ railroad_station_walk_min         : num  65 40.9 15.3 51.5 10.2 ...
 $ ID_railroad_station_walk          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ railroad_station_avto_km          : num  5.42 3.64 1.28 3.82 1.6 ...
 $ railroad_station_avto_min         : num  6.91 4.68 1.7 5.27 2.16 ...
 $ public_transport_station_km       : num  0.275 0.0653 0.3288 0.1316 0.0715 ...
 $ public_transport_station_min_walk: num  3.3 0.783 3.945 1.579 0.858 ...
 $ kremlin_km                        : num  15.16 8.7 9.07 19.49 2.58 ...
 $ big_road1_km                      : num  1.422 2.887 0.647 2.678 1.722 ...
$ big_road1_km                      : num  1.422 2.887 0.647 2.678 1.722 ...
$ big_road2_km                      : num  3.83 3.1 2.93 2.78 3.13 ...
$ railroad_km                       : num  1.3052 0.6945 0.7007 1.9993 0.0841 ...
$ bus_terminal_avto_km              : num  24.29 5.71 6.71 6.73 1.42 ...
$ big_market_km                     : num  10.81 6.91 5.75 27.89 9.16 ...
$ market_shop_km                    : num  1.676 3.425 1.375 0.811 1.97 ...
$ fitness_km                        : num  0.486 0.668 0.733 0.623 0.22 ...
$ swim_pool_km                      : num  3.07 2 1.24 1.95 2.54 ...
$ ice_rink_km                       : num  1.11 8.97 1.98 6.48 3.98 ...
$ stadium_km                        : num  8.149 6.127 0.768 7.386 3.611 ...
$ basketball_km                     : num  3.517 1.162 1.953 4.924 0.308 ...
$ public_healthcare_km              : num  0.9747 1.4777 0.0971 2.1637 1.1217 ...
$ university_km                     : num  6.715 1.853 0.841 10.903 0.992 ...
$ workplaces_km                     : num  0.884 0.686 1.51 0.622 0.893 ...
$ shopping_centers_km               : num  0.648 0.519 1.487 0.6 0.429 ...
$ office_km                         : num  0.6372 0.6888 1.543 0.9343 0.0779 ...
$ big_church_km                     : num  0.626 0.968 3.179 1.032 0.379 ...
$ price_doc                         : int  5850000 6000000 5700000 13100000 16331452 9100000 5500000 2000000 5300000 2000000 ...
```

Figure 1.2



```
       id          timestamp        full_sq           life_sq          floor          max_floor       material      build_year
 Min.   :    1   Min.   :40775   Min.   :   0.0   Min.   :  0.00   Min.   : 0.000   Min.   :  0.00   Min.   :1.000   Min.   :       0
 1st Qu.: 7636   1st Qu.:41389   1st Qu.:  38.0   1st Qu.: 20.00   1st Qu.: 3.000   1st Qu.:  9.00   1st Qu.:1.000   1st Qu.:    1967
 Median :15232   Median :41689   Median :  49.0   Median : 30.00   Median : 6.000   Median : 12.00   Median :1.000   Median :    1979
 Mean   :15237   Mean   :41630   Mean   :  54.2   Mean   : 34.02   Mean   : 7.651   Mean   : 12.55   Mean   :1.828   Mean   :    3306
 3rd Qu.:22861   3rd Qu.:41906   3rd Qu.:  63.0   3rd Qu.: 43.00   3rd Qu.:11.000   3rd Qu.: 17.00   3rd Qu.:2.000   3rd Qu.:    2005
 Max.   :30473   Max.   :42185   Max.   :5326.0   Max.   :802.00   Max.   :77.000   Max.   :117.00   Max.   :6.000   Max.   :20052009
                                                  NA's   :5333     NA's   :146      NA's   :7991     NA's   :7991    NA's   :11392
    num_room        kitch_sq             product_type   raion_popul    green_zone_part   indust_part    children_preschool preschool_quota
 Min.   : 0.000   Min.   :   0.000   Investment   :16278   Min.   :  2546   Min.   :0.001879   Min.   :0.00000   Min.   :  175      Min.   :   0
 1st Qu.: 1.000   1st Qu.:   1.000   Owneroccupier: 9193   1st Qu.: 21819   1st Qu.:0.065409   1st Qu.:0.01765   1st Qu.: 1706      1st Qu.: 1874
 Median : 2.000   Median :   6.000                         Median : 83502   Median :0.169625   Median :0.07216   Median : 4857      Median : 2854
 Mean   : 1.906   Mean   :   6.471                         Mean   : 84139   Mean   :0.220090   Mean   :0.11838   Mean   : 5143      Mean   : 3274
 3rd Qu.: 2.000   3rd Qu.:   9.000                         3rd Qu.:122862   3rd Qu.:0.336177   3rd Qu.:0.19449   3rd Qu.: 7103      3rd Qu.: 4050
 Max.   :19.000   Max.   :2014.000                         Max.   :247469   Max.   :0.852923   Max.   :0.52187   Max.   :19223      Max.   :11926
 NA's   :7991     NA's   :7991                                                                                    NA's   :5604
 children_school hospital_beds_raion healthcare_centers_raion university_top_20_raion shopping_centers_raion  office_raion
 Min.   :  168   Min.   :   0        Min.   :0.00             Min.   :0.000            Min.   : 0.000         Min.   :  0.000
 1st Qu.: 1564   1st Qu.: 520        1st Qu.:0.00             1st Qu.:0.000            1st Qu.: 1.000         1st Qu.:  0.000
 Median : 5261   Median : 990        Median :1.00             Median :0.000            Median : 3.000         Median :  2.000
 Mean   : 5359   Mean   :1195        Mean   :1.32             Mean   :0.138            Mean   : 4.211         Mean   :  8.269
 3rd Qu.: 7227   3rd Qu.:1786        3rd Qu.:2.00             3rd Qu.:0.000            3rd Qu.: 6.000         3rd Qu.:  5.000
 Max.   :19083   Max.   :4849        Max.   :6.00             Max.   :3.000            Max.   :23.000         Max.   :141.000
                 NA's   :12096
 railroad_terminal_raion big_market_raion    full_all          X0_6_all       X7_14_all      X0_17_all      X16_29_all     X0_13_all
 no :24522               no :23117        Min.   :   2546   Min.   :  175   Min.   :  168   Min.   :  411   Min.   :  575   Min.   :  322
 yes:  949               yes: 2354        1st Qu.:  28179   1st Qu.: 1706   1st Qu.: 1564   1st Qu.: 3831   1st Qu.: 5829   1st Qu.: 3112
                                          Median :  85083   Median : 4857   Median : 5261   Median :12508   Median :17864   Median : 9633
                                          Mean   : 146915   Mean   : 5143   Mean   : 5359   Mean   :12551   Mean   :31441   Mean   : 9849
                                          3rd Qu.: 125111   3rd Qu.: 7103   3rd Qu.: 7227   3rd Qu.:16727   3rd Qu.:27107   3rd Qu.:13121
                                          Max.   :1716730   Max.   :19223   Max.   :19083   Max.   :45170   Max.   :367659  Max.   :36035

 build_count_block build_count_wood build_count_frame build_count_brick build_count_before_1920 build_count_1921.1945 build_count_1946.1970
 Min.   :  0.00    Min.   :  0.00   Min.   : 0.000    Min.   :  0.0     Min.   :   0.00          Min.   :  0.00         Min.   :   0
 1st Qu.: 13.00    1st Qu.:  0.00   1st Qu.: 0.000    1st Qu.: 10.0     1st Qu.:   0.00          1st Qu.:  0.00         1st Qu.:  14
 Median : 42.00    Median :  0.00   Median : 0.000    Median : 67.0     Median :   0.00          Median :  2.00         Median : 135
 Mean   : 50.18    Mean   : 40.35   Mean   : 4.935    Mean   :108.1     Mean   :  18.83          Mean   : 26.55         Mean   : 141
 3rd Qu.: 72.00    3rd Qu.:  7.00   3rd Qu.: 1.000    3rd Qu.:156.0     3rd Qu.:   3.00          3rd Qu.: 20.00         3rd Qu.: 216
 Max.   :223.00    Max.   :793.00   Max.   :97.000    Max.   :664.0     Max.   :371.00           Max.   :382.00         Max.   : 845
 NA's   :4195      NA's   :4195     NA's   :4195      NA's   :4195      NA's   :4195             NA's   :4195           NA's   :4195
 build_count_1971.1995 build_count_after_1995 metro_min_avto   metro_km_avto    metro_min_walk   metro_km_walk      school_km
 Min.   :  0.00        Min.   :  0.00         Min.   : 0.000   Min.   : 0.000   Min.   :  0.00   Min.   : 0.0000   Min.   : 0.0000
 1st Qu.: 38.00        1st Qu.: 14.00         1st Qu.: 1.715   1st Qu.: 1.037   1st Qu.: 11.42   1st Qu.: 0.9517   1st Qu.: 0.2697
 Median : 71.00        Median : 24.00         Median : 2.800   Median : 1.781   Median : 20.42   Median : 1.7018   Median : 0.4755
 Mean   : 80.41        Mean   : 61.14         Mean   : 4.967   Mean   : 3.706   Mean   : 42.78   Mean   : 3.5646   Mean   : 1.3285
 3rd Qu.:125.00        3rd Qu.: 57.00         3rd Qu.: 4.855   3rd Qu.: 3.777   3rd Qu.: 45.32   3rd Qu.: 3.7768   3rd Qu.: 0.8898
 Max.   :246.00        Max.   :799.00         Max.   :60.942   Max.   :74.580   Max.   :683.40   Max.   :56.9502   Max.   :47.3947
 NA's   :4195          NA's   :4195                                             NA's   :19       NA's   :19
    park_km        green_zone_km     industrial_km     railroad_station_walk_km railroad_station_walk_min ID_railroad_station_walk
 Min.   : 0.00374   Min.   :0.0000   Min.   : 0.0000   Min.   : 0.02815         Min.   :  0.3378          Min.   :  1.00
 1st Qu.: 0.97297   1st Qu.:0.1006   1st Qu.: 0.2883   1st Qu.: 1.93290         1st Qu.: 23.1948          1st Qu.: 18.00
 Median : 1.80281   Median :0.2143   Median : 0.5773   Median : 3.23554         Median : 38.8265          Median : 33.00
 Mean   : 3.10220   Mean   :0.3001   Mean   : 0.7701   Mean   : 4.39285         Mean   : 52.7142          Mean   : 38.86
 3rd Qu.: 3.40479   3rd Qu.:0.4141   3rd Qu.: 1.0457   3rd Qu.: 5.14764         3rd Qu.: 61.7717          3rd Qu.: 54.00
 Max.   :47.35154   Max.   :1.9824   Max.   :14.0482   Max.   :24.65304         Max.   :295.8365          Max.   :131.00
                                                        NA's   :19               NA's   :19                NA's   :19
    park_km        green_zone_km     industrial_km     railroad_station_walk_km railroad_station_walk_min ID_railroad_station_walk
 Min.   : 0.00374   Min.   :0.0000   Min.   : 0.0000   Min.   : 0.02815         Min.   :  0.3378          Min.   :  1.00
 1st Qu.: 0.97297   1st Qu.:0.1006   1st Qu.: 0.2883   1st Qu.: 1.93290         1st Qu.: 23.1948          1st Qu.: 18.00
 Median : 1.80281   Median :0.2143   Median : 0.5773   Median : 3.23554         Median : 38.8265          Median : 33.00
 Mean   : 3.10220   Mean   :0.3001   Mean   : 0.7701   Mean   : 4.39285         Mean   : 52.7142          Mean   : 38.86
 3rd Qu.: 3.40479   3rd Qu.:0.4141   3rd Qu.: 1.0457   3rd Qu.: 5.14764         3rd Qu.: 61.7717          3rd Qu.: 54.00
 Max.   :47.35154   Max.   :1.9824   Max.   :14.0482   Max.   :24.65304         Max.   :295.8365          Max.   :131.00
                                                        NA's   :19               NA's   :19                NA's   :19
 railroad_station_avto_km railroad_station_avto_min public_transport_station_km public_transport_station_min_walk   kremlin_km
 Min.   : 0.02815         Min.   : 0.03519         Min.   : 0.002804           Min.   :  0.03365                 Min.   : 0.0729
 1st Qu.: 2.12660         1st Qu.: 3.24556         1st Qu.: 0.101156           1st Qu.:  1.21387                 1st Qu.:10.4605
 Median : 3.43172         Median : 4.94456         Median : 0.160538           Median :  1.92646                 Median :14.9093
 Mean   : 4.59500         Mean   : 6.09844         Mean   : 0.415820           Mean   :  4.98984                 Mean   :16.0501
 3rd Qu.: 5.38470         3rd Qu.: 7.30940         3rd Qu.: 0.278213           3rd Qu.:  3.33855                 3rd Qu.:20.6668
 Max.   :24.65398         Max.   :38.69192         Max.   :17.413002           Max.   :208.95602                 Max.   :70.7388

  big_road1_km       big_road2_km       railroad_km       bus_terminal_avto_km big_market_km     market_shop_km       fitness_km
 Min.   :0.000364   Min.   : 0.001935   Min.   : 0.002299   Min.   : 0.06203     Min.   : 0.6614   Min.   : 0.02955   Min.   : 0.0000
 1st Qu.:0.785488   1st Qu.: 2.091603   1st Qu.: 0.658500   1st Qu.: 5.21173     1st Qu.: 7.5296   1st Qu.: 1.53315   1st Qu.: 0.3612
 Median :1.724121   Median : 3.210638   Median : 1.239588   Median : 7.44759     Median :11.9104   Median : 2.92912   Median : 0.6563
 Mean   :1.883979   Mean   : 3.392536   Mean   : 1.893174   Mean   : 9.99020     Mean   :13.2747   Mean   : 3.96153   Mean   : 1.1558
 3rd Qu.:2.804282   3rd Qu.: 4.302618   3rd Qu.: 2.515173   3rd Qu.:13.31172     3rd Qu.:16.5331   3rd Qu.: 5.49537   3rd Qu.: 1.3326
 Max.   :6.995416   Max.   :13.798346   Max.   :17.387119   Max.   :74.46997     Max.   :59.5016   Max.   :40.77751   Max.   :24.8530

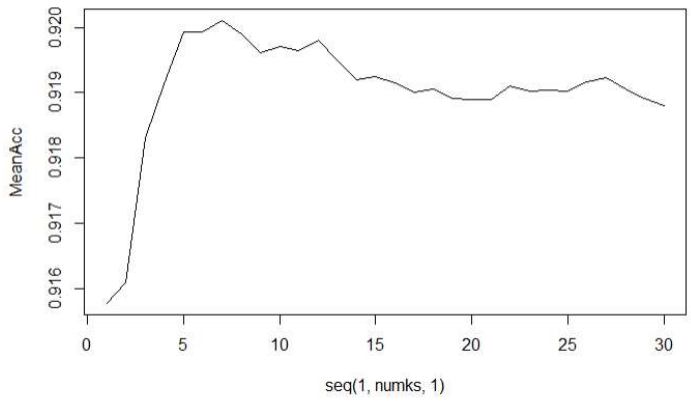
  swim_pool_km      ice_rink_km       stadium_km       basketball_km      public_healthcare_km university_km      workplaces_km
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.1148   Min.   : 0.00546   Min.   : 0.000       Min.   : 0.00031   Min.   :  0.000
 1st Qu.: 1.700   1st Qu.: 3.036   1st Qu.: 4.0182   1st Qu.: 1.30631   1st Qu.: 1.279       1st Qu.: 2.19709   1st Qu.:  1.017
 Median : 2.877   Median : 5.519   Median : 6.9477   Median : 2.87313   Median : 2.342       Median : 4.34023   Median :  2.028
 Mean   : 4.234   Mean   : 6.116   Mean   : 9.4349   Mean   : 4.79007   Mean   : 3.364       Mean   : 6.86504   Mean   :  3.932
 3rd Qu.: 5.370   3rd Qu.: 7.956   3rd Qu.:13.5918   3rd Qu.: 6.36452   3rd Qu.: 3.990       3rd Qu.: 9.39667   3rd Qu.:  5.431
 Max.   :53.033   Max.   :43.444   Max.   :83.0724   Max.   :56.70379   Max.   :75.729       Max.   :84.53600   Max.   :55.278

 shopping_centers_km   office_km       big_church_km        price_doc
 Min.   : 0.0000     Min.   : 0.000   Min.   : 0.00407   Min.   :   190000
 1st Qu.: 0.4834     1st Qu.: 0.556   1st Qu.: 0.86041   1st Qu.:  4741980
 Median : 0.8356     Median : 1.053   Median : 1.49156   Median :  6272000
 Mean   : 1.5077     Mean   : 2.016   Mean   : 2.33218   Mean   :  7127058
 3rd Qu.: 1.5343     3rd Qu.: 3.081   3rd Qu.: 2.91180   3rd Qu.:  8298102
 Max.   :24.8768     Max.   :18.959   Max.   :45.66906   Max.   :111111112
```

Figure 1.3

```
296 ```{r CleanData}
297 ## Build clean data with 5 selected numeric variables and 3 factors
298 cleanDF <- HouseDF2 %>% select(full_sq, build_year, num_room, X0_17_all, kremlin_km,
299                                 product_type, railroad_terminal_raion, big_market_raion, price_doc)
300 ## Change 3 factors to numeric variables for SAS Lasso modeling
301 cleanDF$product_type <- as.numeric(cleanDF$product_type)
302 cleanDF$railroad_terminal_raion <- as.numeric(cleanDF$railroad_terminal_raion)
303 cleanDF$big_market_raion <- as.numeric(cleanDF$big_market_raion)
```

```
> head(cleanDF)
  full_sq build_year num_room X0_17_all kremlin_km  product_type railroad_terminal_raion big_market_raion price_doc
1      11       1907        1     16584   2.109561    Investment                      no                no   2750000
2      53       1980        2     11158  15.345902    Investment                      no                no   9000000
3      77       2014        3      1150  25.735256 OwnerOccupier                      no                no   7011550
4      45       1970        2     11749  20.728839    Investment                      no                no   7100000
5      38       1982        1      9249   8.569880    Investment                      no                no   6450000
6      74       2004        3     11567  13.529297    Investment                      no                no  12100000
```

```
> head(cleanDF)
  full_sq build_year num_room X0_17_all kremlin_km product_type railroad_terminal_raion big_market_raion price_doc
1      11       1907        1     16584   2.109561            1                       1                1   2750000
2      53       1980        2     11158  15.345902            1                       1                1   9000000
3      77       2014        3      1150  25.735256            2                       1                1   7011550
4      45       1970        2     11749  20.728839            1                       1                1   7100000
5      38       1982        1      9249   8.569880            1                       1                1   6450000
6      74       2004        3     11567  13.529297            1                       1                1  12100000
```

```
> str(cleanDF)
'data.frame':   12883 obs. of  9 variables:
 $ full_sq               : int  11 53 77 45 38 74 93 51 77 38 ...
 $ build_year            : int  1907 1980 2014 1970 1982 2004 2013 2003 1957 1986 ...
 $ num_room              : int  1 2 3 2 1 3 3 2 3 1 ...
 $ X0_17_all             : int  16584 11158 1150 11749 9249 11567 1138 17908 10478 14976 ...
 $ kremlin_km            : num  2.11 15.35 25.74 20.73 8.57 ...
 $ product_type          : num  1 1 2 1 1 1 2 1 1 1 ...
 $ railroad_terminal_raion: num  1 1 1 1 1 1 1 1 1 1 ...
 $ big_market_raion      : num  1 1 1 1 1 1 1 1 1 2 ...
 $ price_doc             : int  2750000 9000000 7011550 7100000 6450000 12100000 5427640 7700000 11700000 1600000 ...
```

Figure 1.4


price_doc
green_zone_km
railroad_km
railroad_station_walk_km
railroad_station_avto_km
school_km
public_transport_station_km
bus_terminal_avto_km
kremlin_km
park_km
metro_km_avto
metro_km_walk
industrial_km
big_road1_km
big_road2_km
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

Figure 1.5



Figure 1.6



Figure 1.7



Figure 1.8



Figure 1.9



Figure 1.10



Figure 1.11



Figure 1.12



Figure 1.13

```
> table(HouseDF1$build_year)

   0    1    3  215 1691 1860 1876 1890 1896 1900 1905 1906 1907 1910 1911
 337  198    2    1    1    2    1    4    1    1    1    1    2    4    1
1912 1914 1915 1917 1920 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933
   4    1    4   11    1    3    1    6    8    9    8    5    5    6    7
1934 1935 1936 1937 1938 1939 1940 1941 1943 1946 1947 1948 1949 1950 1951
  12    9    3   10    8    7   10    1    2    2    4    1    2   18   19
1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966
  39   20   30   42   41   98  150  167  287  247  287  275  264  320  292
1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981
 307  310  338  349  301  302  283  297  257  223  223  200  201  192  163
1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996
 160  161  138  153  112  144  126  132  109   78  112   92  132  123  136
1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011
 114  118  107  107  146  173  167  183  143  200  182  196  147  116  133
2012 2013 2014 2015 2016 2017 2018 4965 20052009
 192  321  597  539  273  111    1    1    1
```

```
## Dealing with the missing values in build_year (11392)
```{r}
## Checked which variables have strong correlation with build_year: None
## Delete observations from HouseDF1 for 'build_year' with values: NA's, 0, 1, 3, 215, 1691, 4965, 20052009
## Create clean HouseDF2
HouseDF2 <- HouseDF1 %>% filter(!is.na(build_year) &
                                build_year != 0 &
                                build_year != 1 &
                                build_year != 3 &
                                build_year != 215 &
                                build_year != 1691 &
                                build_year != 4965 &
                                build_year != 20052009)
summary(HouseDF2)
```
```

```
> summary(HouseDF2)
       id              timestamp        full_sq           life_sq          floor          max_floor       material    
 Min.   : 8059   12/16/2014:   98   Min.   :  1.00   Min.   :  0.00   Min.   : 0.000   Min.   : 0.00   Min.   :1.000  
 1st Qu.:14604   12/9/2014 :   91   1st Qu.: 38.00   1st Qu.: 20.00   1st Qu.: 3.000   1st Qu.: 9.00   1st Qu.:1.000  
 Median :19636   6/30/2014 :   83   Median : 47.00   Median : 30.00   Median : 6.000   Median :12.00   Median :1.000  
 Mean   :19764   12/18/2014:   81   Mean   : 52.82   Mean   : 33.04   Mean   : 7.015   Mean   :12.41   Mean   :1.934  
 3rd Qu.:24938   12/2/2014 :   60   3rd Qu.: 62.00   3rd Qu.: 42.00   3rd Qu.:10.000   3rd Qu.:17.00   3rd Qu.:2.000  
 Max.   :30473   9/30/2014 :   57   Max.   :219.00   Max.   :458.00   Max.   :77.000   Max.   :48.00   Max.   :6.000  
                 (Other)   :12413                                                                                      
   build_year      num_room        kitch_sq          product_type    raion_popul     green_zone_part     indust_part      
 Min.   :1860   Min.   :0.000   Min.   :   0.000   Investment   :11260   Min.   :  2546   Min.   :0.001879   Min.   :0.00000  
 1st Qu.:1968   1st Qu.:1.000   1st Qu.:   5.000   OwnerOccupier: 1623   1st Qu.: 72131   1st Qu.:0.065409   1st Qu.:0.03349  
 Median :1980   Median :2.000   Median :   8.000                         Median :101708   Median :0.137846   Median :0.08904  
 Mean   :1984   Mean   :1.941   Mean   :   7.955                         Mean   :101258   Mean   :0.199537   Mean   :0.12338  
 3rd Qu.:2004   3rd Qu.:3.000   3rd Qu.:   9.000                         3rd Qu.:132349   3rd Qu.:0.331319   3rd Qu.:0.19449  
 Max.   :2018   Max.   :5.000   Max.   :2014.000                         Max.   :247469   Max.   :0.852923   Max.   :0.52187  
```

Figure 1.14



Figure 1.15

```
   full_sq      
 Min.   :   0.0  
 1st Qu.:  38.0  
 Median :  49.0  
 Mean   :  54.2  
 3rd Qu.:  63.0  
 Max.   :5326.0  
```

```
    life_sq      
 Min.   :  0.00  
 1st Qu.: 20.00  
 Median : 30.00  
 Mean   : 34.02  
 3rd Qu.: 43.00  
 Max.   :802.00  
 NA's   :5333    
```

```
    num_room      
 Min.   : 0.000  
 1st Qu.: 1.000  
 Median : 2.000  
 Mean   : 1.906  
 3rd Qu.: 2.000  
 Max.   :19.000  
 NA's   :7991    
```

```
> table(nRoomDF$num_room)

   0    1    2    3    4    5    6    8    9   17   19
   6 4988 5396 3257  320   30    5    3    1    1    1
```

Figure 1.16

```
## Use KNN model to predict the missing values of num_room
```{r KNN for num_room, echo=FALSE}
nRoomDF <- HouseDF %>% filter(!is.na(num_room) & !is.na(life_sq) & (num_room=='1' | num_room=='2' | num_room=='3' | num_room =='4' | num_room=='5')) %>%
  select(num_room,full_sq,life_sq)
#Use a 70 - 30 train/test split to use cross validation to
#tune the hyperparameter k
# Loop for many k and the average of many training / test partition
set.seed(1)
iterations = 50
numks = 30
splitPerc = .70
masterAcc = matrix(nrow = iterations, ncol = numks)

for(j in 1:iterations)
{
  trainIndices = sample(1:dim(nRoomDF)[1],round(splitPerc * dim(nRoomDF)[1]))
  train = nRoomDF[trainIndices,]
  test = nRoomDF[-trainIndices,]
  for(i in 1:numks)
  {
    classifications = knn(train[,c(2,3)],test[,c(2,3)],as.factor(train$num_room), prob = TRUE, k = i)
    table(classifications,test$num_room)
    CM = confusionMatrix(table(classifications,test$num_room))
    masterAcc[j,i] = CM$overall[1]
  }
}
MeanAcc = colMeans(masterAcc)
plot(seq(1,numks,1),MeanAcc, type = "l")
which.max(MeanAcc)
max(MeanAcc)
```




```
> which.max(MeanAcc)
[1] 7
> max(MeanAcc)
[1] 0.9201096
```

Figure 1.17

```
```{r NA_num_room, echo=FALSE}
# Firstly build dataframe1 without missing value of num_room
missing_num_room1 <- HouseDF %>% filter(!is.na(num_room) & !is.na(life_sq) & (num_room<=5))
# Secondly build a dataframe2 with missing value of num_room
missing_num_room2 <- HouseDF %>% filter(is.na(num_room) & !is.na(life_sq) & (num_room<=5))
# Thirdly update dataframe2 with all missing value of num_room replaced by predicted value
splitPerc = .70
trainIndices = sample(1:dim(missing_num_room1)[1],round(splitPerc * dim(missing_num_room1)[1]))

train = missing_num_room1[trainIndices,]
test = missing_num_room1[-trainIndices,]

classifications1 = knn(train[,c(3,4)],test[,c(3,4)],as.factor(train$num_room), prob = TRUE, k = 7)
temp <- table(classifications1,as.factor(test$num_room))
CM1 = confusionMatrix(table(classifications1,test$num_room))

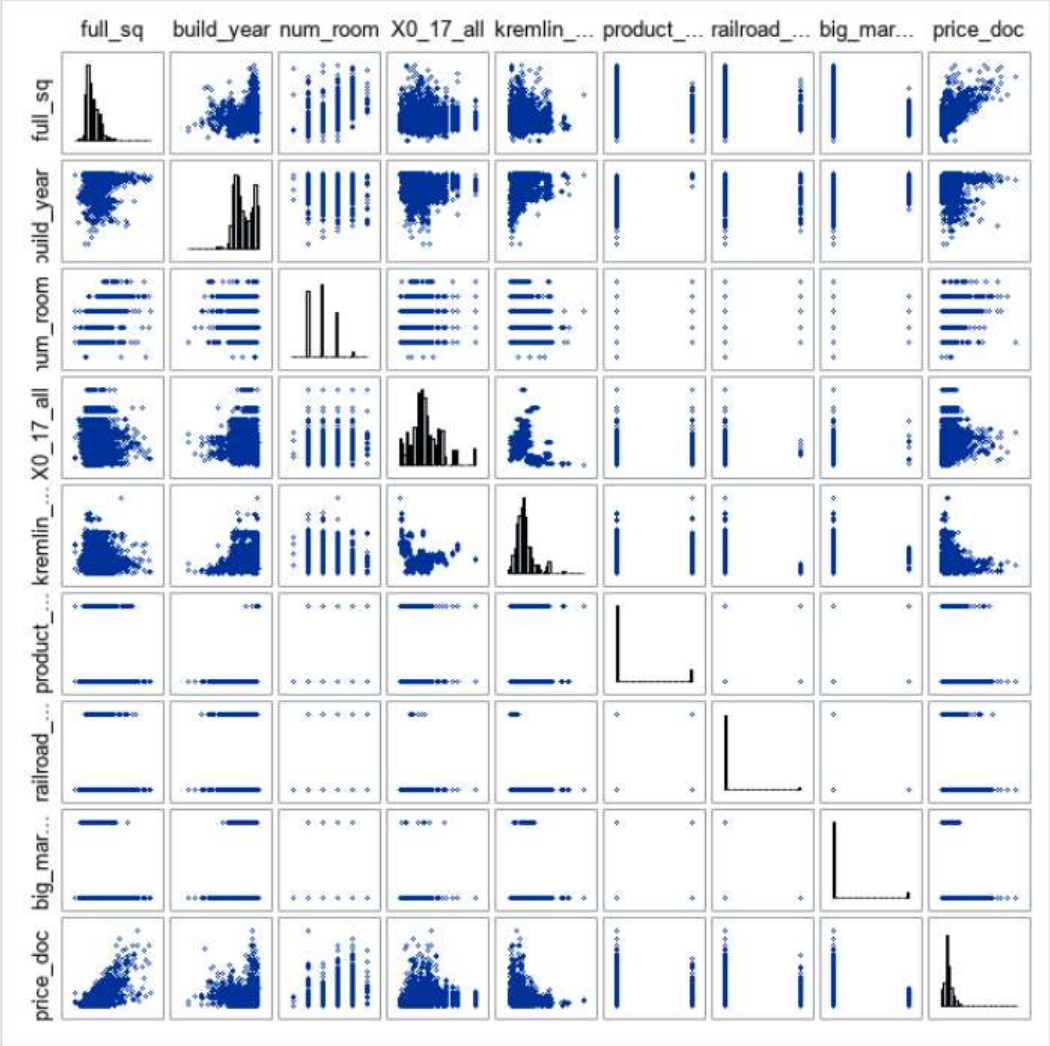
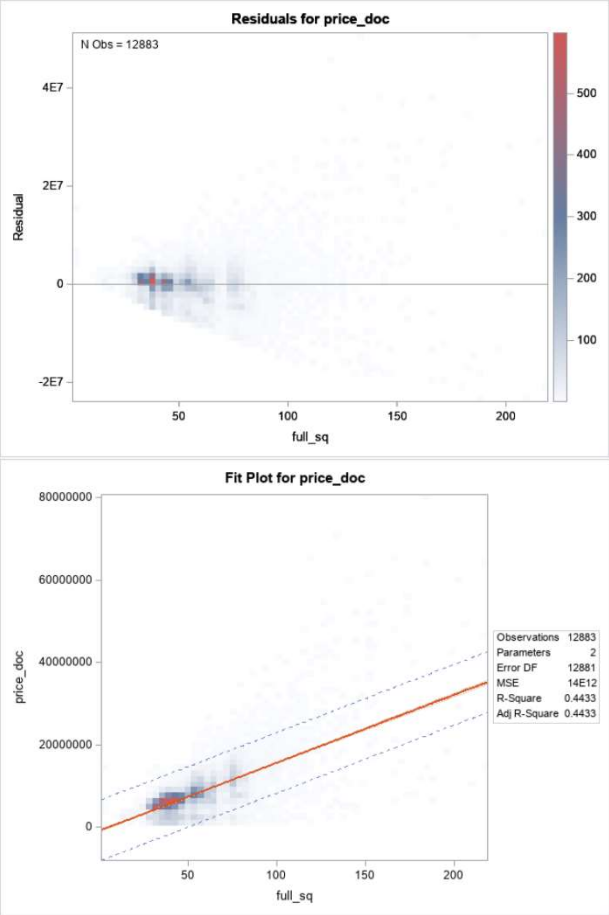
pred_num_room <- knn(train[,c(3,4)],missing_num_room2[,c(3,4)],as.factor(train$num_room), prob = TRUE, k = 7)
updated_missing_num_room2 <- data.frame(missing_num_room2[,1:8],data.frame(as.factor(pred_num_room)),missing_num_room2[,10:73])
colnames(updated_missing_num_room2)[9] = "num_room"
updated_missing_num_room2$num_room <- as.integer(updated_missing_num_room2$num_room)

# Fourthly merge dataframe1 with updated dataframe2
HouseDF1 <- rbind(missing_num_room1, updated_missing_num_room2)
```

```
> str(HouseDF1)
'data.frame':   13997 obs. of  73 variables:
 $ id                            : int  7675 8059 8138 8147 8156 8157 8178 8219 8258 8271 ...
 $ timestamp                     : Factor w/ 1158 levels "1/10/2012","1/10/2013",..: 663 740 755 761 761 761 765 769 775 779 ...
 $ full_sq                       : int  73 11 53 41 77 45 38 58 74 93 ...
 $ life_sq                       : int  36 11 30 37 41 27 20 30 46 93 ...
 $ floor                         : int  17 2 10 13 2 6 15 13 12 3 ...
 $ max_floor                     : int  17 5 16 17 17 9 16 0 24 1 ...
 $ material                      : int  1 2 1 1 6 1 1 1 1 1 ...
 $ build_year                    : int  NA 1907 1980 NA 2014 1970 1982 NA 2004 2013 ...
 $ num_room                      : int  2 1 2 1 3 2 1 2 3 3 ...
 $ kitch_sq                      : int  11 12 8 1 12 6 8 0 9 1 ...
```

```
> summary(HouseDF1)
       id            timestamp        full_sq           life_sq          floor          max_floor        material    
 Min.   : 7675   12/16/2014:  106   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000   Min.   :  0.00   Min.   :1.000  
 1st Qu.:14665   12/9/2014 :   97   1st Qu.: 38.00   1st Qu.: 20.00   1st Qu.: 3.000   1st Qu.:  9.00   1st Qu.:1.000  
 Median :19682   6/30/2014 :   87   Median : 48.00   Median : 30.00   Median : 6.000   Median : 12.00   Median :1.000  
 Mean   :19807   12/18/2014:   85   Mean   : 53.17   Mean   : 33.88   Mean   : 7.105   Mean   : 12.44   Mean   :1.912  
 3rd Qu.:24999   11/25/2014:   64   3rd Qu.: 63.00   3rd Qu.: 43.00   3rd Qu.:10.000   3rd Qu.: 17.00   3rd Qu.:2.000  
 Max.   :30473   12/2/2014 :   64   Max.   :637.00   Max.   :637.00   Max.   :77.000   Max.   :117.00   Max.   :6.000  
                 (Other)   :13494                                                                                       
   build_year       num_room        kitch_sq          product_type    raion_popul      green_zone_part     indust_part      
 Min.   :   0.0   Min.   :0.000   Min.   :   0.000   Investment   :11367   Min.   :  2546   Min.   :0.001879   Min.   :0.00000  
 1st Qu.:1967.0   1st Qu.:1.000   1st Qu.:   5.000   OwnerOccupier: 2630   1st Qu.: 64317   1st Qu.:0.065409   1st Qu.:0.02561  
 Median :1979.0   Median :2.000   Median :   7.000                         Median : 94564   Median :0.141281   Median :0.07578  
 Mean   :3399.1   Mean   :1.928   Mean   :   7.714                         Mean   : 96682   Mean   :0.206120   Mean   :0.12177  
 3rd Qu.:2003.0   3rd Qu.:3.000   3rd Qu.:   9.000                         3rd Qu.:129207   3rd Qu.:0.336177   3rd Qu.:0.19449  
 Max.   :20052009 Max.   :5.000   Max.   :2014.000                         Max.   :247469   Max.   :0.852923   Max.   :0.52187  
 NA's   :572                                                                                                             
```

Figure 1.18

```
## Dealing with the missing values in build_year (11392)
```{r}
## Checked which variables have strong correlation with build_year: None
## Delete observations from HouseDF1 for 'build_year' with values: NA's, 0, 1, 3, 215, 1691, 4965, 20052009
## Create clean HouseDF2
HouseDF2 <- HouseDF1 %>% filter(!is.na(build_year) &
                                build_year != 0 &
                                build_year != 1 &
                                build_year != 3 &
                                build_year != 215 &
                                build_year != 1691 &
                                build_year != 4965 &
                                build_year != 20052009)
summary(HouseDF2)
```
```

```
> summary(HouseDF2)
       id            timestamp        full_sq           life_sq          floor          max_floor        material    
 Min.   : 8059   12/16/2014:   98   Min.   :  1.00   Min.   :  0.00   Min.   : 0.000   Min.   :  0.00   Min.   :1.000  
 1st Qu.:14604   12/9/2014 :   91   1st Qu.: 38.00   1st Qu.: 20.00   1st Qu.: 3.000   1st Qu.:  9.00   1st Qu.:1.000  
 Median :19636   6/30/2014 :   83   Median : 47.00   Median : 30.00   Median : 6.000   Median : 12.00   Median :1.000  
 Mean   :19764   12/18/2014:   81   Mean   : 52.82   Mean   : 33.04   Mean   : 7.015   Mean   : 12.41   Mean   :1.934  
 3rd Qu.:24938   12/2/2014 :   60   3rd Qu.: 62.00   3rd Qu.: 42.00   3rd Qu.:10.000   3rd Qu.: 17.00   3rd Qu.:2.000  
 Max.   :30473   9/30/2014 :   57   Max.   :219.00   Max.   :458.00   Max.   :77.000   Max.   : 48.00   Max.   :6.000  
                 (Other)   :12413                                                                                       
   build_year       num_room        kitch_sq          product_type    raion_popul      green_zone_part     indust_part      
 Min.   :1860   Min.   :0.000   Min.   :   0.000   Investment   :11260   Min.   :  2546   Min.   :0.001879   Min.   :0.00000  
 1st Qu.:1968   1st Qu.:1.000   1st Qu.:   5.000   OwnerOccupier: 1623   1st Qu.: 72131   1st Qu.:0.065409   1st Qu.:0.03349  
 Median :1980   Median :2.000   Median :   8.000                         Median :101708   Median :0.137846   Median :0.08904  
 Mean   :1984   Mean   :1.941   Mean   :   7.955                         Mean   :101258   Mean   :0.199537   Mean   :0.12338  
 3rd Qu.:2004   3rd Qu.:3.000   3rd Qu.:   9.000                         3rd Qu.:132349   3rd Qu.:0.331319   3rd Qu.:0.19449  
 Max.   :2018   Max.   :5.000   Max.   :2014.000                         Max.   :247469   Max.   :0.852923   Max.   :0.52187  
```

```
> str(HouseDF2)
'data.frame':   12883 obs. of  73 variables:
 $ id                            : int  8059 8138 8156 8157 8178 8258 8271 8285 8290 8296 ...
 $ timestamp                     : Factor w/ 1158 levels "1/10/2012","1/10/2013",..: 740 755 761 761 765 775 779 800 873 873 ...
 $ full_sq                       : int  11 53 77 45 38 74 93 51 77 38 ...
 $ life_sq                       : int  11 30 41 27 20 46 93 30 50 19 ...
 $ floor                         : int  2 10 2 6 15 12 3 7 3 17 ...
 $ max_floor                     : int  5 16 17 9 16 24 1 17 5 17 ...
 $ material                      : int  2 1 6 1 1 1 1 1 2 1 ...
 $ build_year                    : int  1907 1980 2014 1970 1982 2004 2013 2003 1957 1986 ...
 $ num_room                      : int  1 2 3 2 1 3 3 2 3 1 ...
```

Figure 1.19

```
proc sgscatter data= HousePrice;
label full_sq = 'full_sq' build_year = 'build_year' num_room = 'num_room'
  X0_17_all = 'X0_17_all' kremlin_km = 'kremlin_km' product_type = 'product_type'
  railroad_terminal_raion = 'railroad_terminal_raion' big_market_raion = 'big_market_raion'
  price_doc = 'price_doc' ;
matrix full_sq build_year num_room X0_17_all kremlin_km product_type
  railroad_terminal_raion big_market_raion price_doc / diagonal= (histogram) ;
run;
```



Figure 1.20

```
/* Checking assumption */
proc reg data= HousePrice all ;
model price_doc = full_sq ;
run;
```



Figure 1.21

```
```{r CleanData}
## Build clean data with 5 selected numeric variables and 3 factors
cleanDF <- HouseDF2 %>% select(full_sq, build_year, num_room, X0_17_all, kremlin_km,
                               product_type, railroad_terminal_raion, big_market_raion, price_doc)
## Change 3 factors to numeric variables for SAS Lasso modeling
cleanDF$product_type <- as.numeric(cleanDF$product_type)
cleanDF$railroad_terminal_raion <- as.numeric(cleanDF$railroad_terminal_raion)
cleanDF$big_market_raion <- as.numeric(cleanDF$big_market_raion)
```

```
> head(cleanDF)
  full_sq build_year num_room X0_17_all kremlin_km  product_type railroad_terminal_raion big_market_raion price_doc
1      11       1907        1     16584   2.109561    Investment                      no               no   2750000
2      53       1980        2     11158  15.345902    Investment                      no               no   9000000
3      77       2014        3      1150  25.735256 OwnerOccupier                      no               no   7011550
4      45       1970        2     11749  20.728839    Investment                      no               no   7100000
5      38       1982        1      9249   8.569880    Investment                      no               no   6450000
6      74       2004        3     11567  13.529297    Investment                      no               no  12100000
```

```
> head(cleanDF)
  full_sq build_year num_room X0_17_all kremlin_km product_type railroad_terminal_raion big_market_raion price_doc
1      11       1907        1     16584   2.109561            1                       1                1   2750000
2      53       1980        2     11158  15.345902            1                       1                1   9000000
3      77       2014        3      1150  25.735256            2                       1                1   7011550
4      45       1970        2     11749  20.728839            1                       1                1   7100000
5      38       1982        1      9249   8.569880            1                       1                1   6450000
6      74       2004        3     11567  13.529297            1                       1                1  12100000
```

```
> str(cleanDF)
'data.frame':   12883 obs. of  9 variables:
 $ full_sq               : int  11 53 77 45 38 74 93 51 77 38 ...
 $ build_year            : int  1907 1980 2014 1970 1982 2004 2013 2003 1957 1986 ...
 $ num_room              : int  1 2 3 2 1 3 3 2 3 1 ...
 $ X0_17_all             : int  16584 11158 1150 11749 9249 11567 1138 17908 10478 14976 ...
 $ kremlin_km            : num  2.11 15.35 25.74 20.73 8.57 ...
 $ product_type          : num  1 1 2 1 1 1 2 1 1 1 ...
 $ railroad_terminal_raion: num  1 1 1 1 1 1 1 1 1 1 ...
 $ big_market_raion      : num  1 1 1 1 1 1 1 1 1 2 ...
 $ price_doc             : int  2750000 9000000 7011550 7100000 6450000 12100000 5427640 7700000 11700000 1600000 ...
```
```

```
> summary(cleanDF)
    full_sq         build_year      num_room       X0_17_all       kremlin_km      product_type   railroad_terminal_raion big_market_raion
 Min.   :  1.00   Min.   :1860   Min.   :0.000   Min.   :  411   Min.   : 0.0729   Min.   :1.000   Min.   :1.000           Min.   :1.000
 1st Qu.: 38.00   1st Qu.:1968   1st Qu.:1.000   1st Qu.: 9514   1st Qu.: 9.5513   1st Qu.:1.000   1st Qu.:1.000           1st Qu.:1.000
 Median : 47.00   Median :1980   Median :2.000   Median :13855   Median :13.2020   Median :1.000   Median :1.000           Median :1.000
 Mean   : 52.82   Mean   :1984   Mean   :1.941   Mean   :14951   Mean   :14.4128   Mean   :1.126   Mean   :1.028           Mean   :1.066
 3rd Qu.: 62.00   3rd Qu.:2004   3rd Qu.:3.000   3rd Qu.:18912   3rd Qu.:16.9334   3rd Qu.:1.000   3rd Qu.:1.000           3rd Qu.:1.000
 Max.   :219.00   Max.   :2018   Max.   :5.000   Max.   :45170   Max.   :70.7388   Max.   :2.000   Max.   :2.000           Max.   :2.000
   price_doc
 Min.   :  500000
 1st Qu.: 5300000
 Median : 6900000
 Mean   : 7780513
 3rd Qu.: 9200000
 Max.   :80777440
```

Figure 1.22

```
/* Variable Selection */
data HousePrice2;
 set HousePrice;
 RandNumber = ranuni(11); run;
data train;
 set HousePrice2;
 if RandNumber <= 1/4 then delete; run;
data test;
 set HousePrice2;
 if RandNumber > 1/4 then delete; run;
/* LASSO model 1 */
proc glmselect data=train testdata = test
              seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);
 model price_doc = full_sq build_year num_room X0_17_all kremlin_km product_type railroad_terminal_raion
   big_market_raion / selection=LASSO( choose=CV stop=CV) CVdetails;
run; quit;
/* OLS model 2 (stepwise)*/
proc glmselect data=train testdata=test
              seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);
model price_doc = full_sq build_year num_room X0_17_all kremlin_km product_type
  big_market_raion / selection=stepwise( choose=CV stop=CV include = 7 ) CVdetails;
run; quit;
```

```
proc glmselect data = house plots=all;
     model price_doc = full_sq build_year num_room X0_17_all kremlin_km product_type railroad_terminal_raion big_market_raion
         full_sq*build_year*num_room*X0_17_all*kremlin_km*product_type*railroad_terminal_raion*big_market_raion
         / SELECTION = STEPWISE (choose=CV stop = cv) CVdetails;
 run;
```

Figure 1.23

```
```{r CV_cleanDF}
## Cross validation on HouseDF2
## With 5 exploratory variables: full_sq, build_year, num_room, X0_17_all, kremlin_km
## Adj.R-squared: 0.5299, p-value: < 2.2e-16
#priceDF2 <- HouseDF2 %>% select(full_sq, build_year, num_room, X0_17_all, kremlin_km, price_doc)
set.seed(8)
splitPerc = .70
trainIndices = sample(1:dim(cleanDF)[1],round(splitPerc * dim(cleanDF)[1]))
train2 = cleanDF[trainIndices,]
test2 = cleanDF[-trainIndices,]
### Step1: Fit model with training dataset
model2_fit <- lm(price_doc ~ full_sq + num_room + X0_17_all + kremlin_km + build_year + product_type
                 + railroad_terminal_raion + big_market_raion , data = cleanDF)
summary(model2_fit)
### Step2: Get predicted value with test set
model2_preds <- predict(model2_fit, test2)
### Step3: Cross validation with RMSLE
RMSLE2 = sqrt(mean((log(test2$price_doc+1) - log(abs(model2_preds)+1))^2))
RMSLE2
### RMSLE2 = 0.5764483
```
```

```
Call:
lm(formula = price_doc ~ full_sq + num_room + X0_17_all + kremlin_km +
    build_year + product_type + railroad_terminal_raion + big_market_raion,
    data = cleanDF)

Residuals:
      Min        1Q    Median        3Q       Max
-27347657   -883467    342337   1401987  49220739

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              3.012e+07  4.248e+06   7.091 1.40e-12 ***
full_sq                  1.986e+05  2.710e+03  73.290  < 2e-16 ***
num_room                -8.952e+05  6.249e+04 -14.326  < 2e-16 ***
X0_17_all                1.370e+01  3.678e+00   3.726 0.000196 ***
kremlin_km              -1.663e+05  4.581e+03 -36.303  < 2e-16 ***
build_year              -1.389e+04  2.197e+03  -6.322 2.67e-10 ***
product_type            -2.701e+05  1.169e+05  -2.311 0.020838 *
railroad_terminal_raion -2.119e+05  1.916e+05  -1.106 0.268829
big_market_raion        -7.662e+05  1.231e+05  -6.226 4.93e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3444000 on 12874 degrees of freedom
Multiple R-squared:  0.5302,    Adjusted R-squared:  0.5299
F-statistic:  1816 on 8 and 12874 DF,  p-value: < 2.2e-16
```

Figure 2.1

```
# How to determine the format code
# %d   day of the month
# %b   abbreviation of the month
# %Y   Year in 4 digits
# %y   Year in 2 digits
# %m   month number
# %B   month full name
# Select 2 variables: timestamp and price_doc
timeDF <- HouseDF %>% select(timestamp, price_doc)
head(timeDF)
# Change timestamp class: factor -> date
timeDF$timestamp <- as.Date(timeDF$timestamp, "%m/%d/%Y")
# Split timestamp to 3 columns in class character
temp <- separate(timeDF, col = timestamp, into = c("year", "month", "day"), sep = "-")
# Combine year and month to 'monthYear'
timeDF1 <- unite(temp, monthYear, year, month, sep = "")
# Group by monthYear and average price
timeDF2 <- timeDF1 %>% group_by(monthYear) %>% dplyr::summarise(AvgPrice=mean(price_doc))
# Add new column MonthNumber
timeDF3 <- data.frame(c(1:dim(timeDF2)[1]), timeDF2)
# Change column name
colnames(timeDF3)[1] = "MonthNumber"
```

```
> head(timeDF3, 10)
   MonthNumber monthYear AvgPrice
1            1    201108  5850000
2            2    201109  6255310
3            3    201110  5667466
4            4    201111  6140269
5            5    201112  5812806
6            6    201201  6967015
7            7    201202  6815771
8            8    201203  6789860
9            9    201204  6720821
10          10    201205  7393949
> summary(timeDF3)
  MonthNumber     monthYear           AvgPrice
 Min.   : 1.0   Length:47          Min.   :5529669
 1st Qu.:12.5   Class :character   1st Qu.:6298967
 Median :24.0   Mode  :character   Median :6922998
 Mean   :24.0                      Mean   :6927908
 3rd Qu.:35.5                      3rd Qu.:7383190
 Max.   :47.0                      Max.   :8501842
> str(timeDF3)
'data.frame':   47 obs. of  3 variables:
 $ MonthNumber: int  1 2 3 4 5 6 7 8 9 10 ...
 $ monthYear  : chr  "201108" "201109" "201110" "201111" ...
 $ AvgPrice   : num  5850000 6255310 5667466 6140269 5812806 ...
```

Figure 2.2

```
394 timeDF3 %>% ggplot(aes(x=MonthNumber, y=AvgPrice)) + geom_point() +
395   xlab("months") + ylab("Average Price") + ggtitle("Time Series of price_doc vs. Months")
```

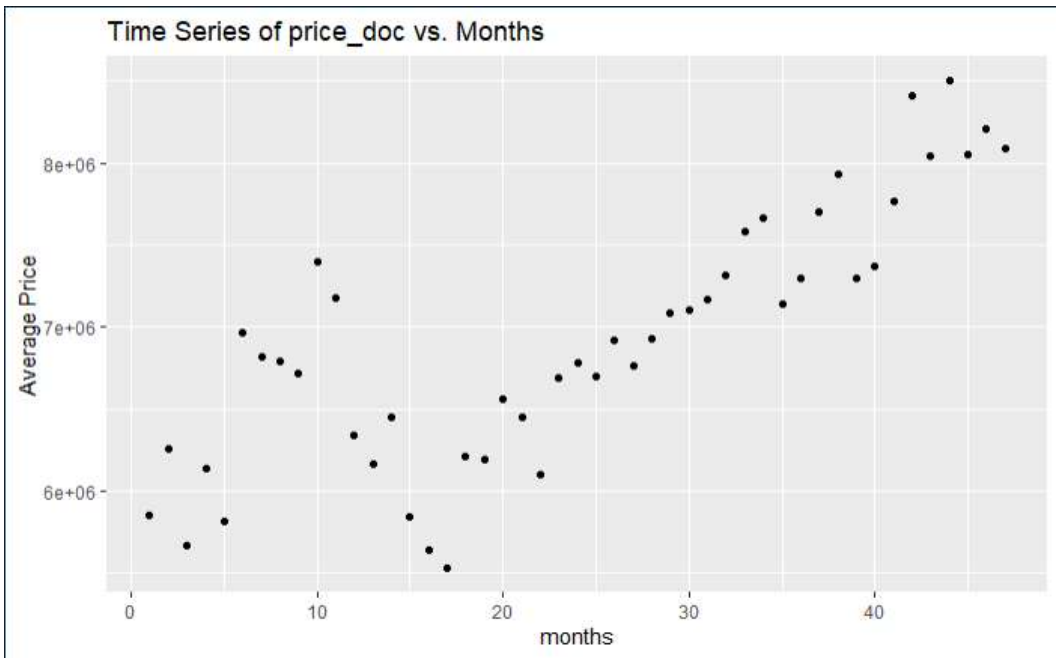


Figure 2.3

```
400 fit <- lm(AvgPrice~MonthNumber, data = timeDF3)
401 summary(fit)
```

```
> summary(fit)

Call:
lm(formula = AvgPrice ~ MonthNumber, data = timeDF3)

Residuals:
     Min       1Q   Median       3Q      Max
-1076456  -283149   -68818   255276  1109607

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5824652     136857   42.56  < 2e-16 ***
MonthNumber    45969       4964    9.26 5.46e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 461700 on 45 degrees of freedom
Multiple R-squared:  0.6558,    Adjusted R-squared:  0.6482
F-statistic: 85.75 on 1 and 45 DF,  p-value: 5.464e-12
```

Figure 2.4

```
# b. Plot residual vs MonthNumber
timeDF4 <- data.frame(timeDF3, as.data.frame(fit$residuals))
colnames(timeDF4)[4] = "Residual"
timeDF4 %>% ggplot(aes(x=MonthNumber, y=Residual)) + geom_line() + geom_point() +
  xlab("Month") + ylab("Residual of Average Price") + ggtitle("Residual vs. Month")
```

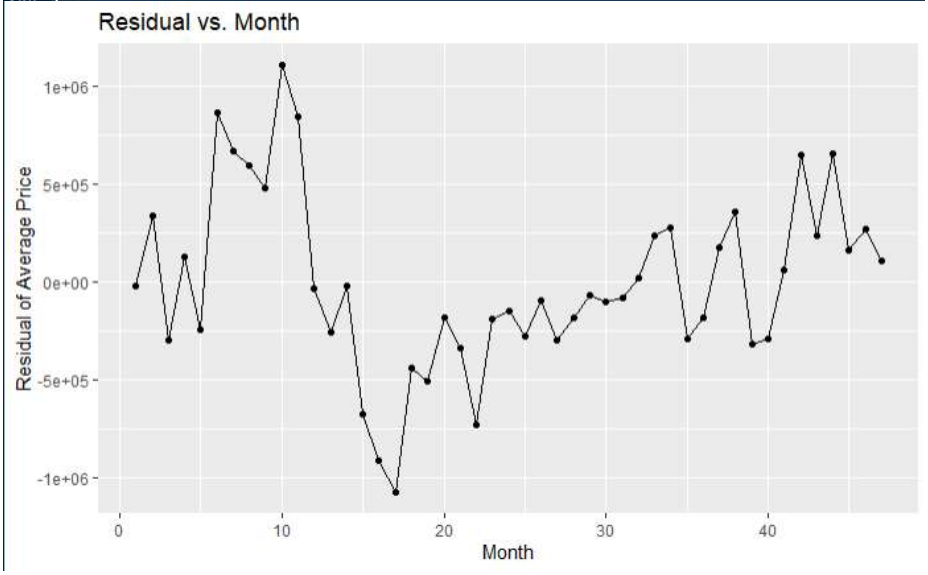


Figure 2.5

```
/* 3. Use autoreg: Ordinary Least Square Regression Model - OLS model */
proc autoreg data=AvgHousePrice;
model Residual = MonthNumber / dwprob; *** no lag;
run;

/* 4. Use autoreg to build model account for lag(1) */
proc autoreg data=AvgHousePrice;
model Residual = MonthNumber / nlag=(1) dwprob; *** lag(1);
run;

/* 5. Use autoreg to build model account for lag(2) */
proc autoreg data=AvgHousePrice;
model Residual = MonthNumber / nlag=(2) dwprob; *** lag(2);
run;
```

Figure 2.6

```
/* 6. Use autoreg to predict 201506-201606 */
data addObsForPred;
input MonthNumber MonthYear @@; cards;
48 201507
49 201508
50 201509
51 201510
52 201511
53 201512
54 201601
55 201602
56 201603
57 201604
58 201605
59 201606
;
run;
data forPred; set AvgHousePrice addObsForPred; run; /* Adding new obs to data set */

/* 6.1 Predict Residual 201506-201606 */
proc autoreg data=forPred;
model Residual = MonthNumber / nlag=(2) dwprob;  /* Prediction model with lag(2) */
output out=preds1 p=prediction1 lcl=lower1 ucl=upper1 pm=trend1; /* Adding new columns */
run;
proc print data=preds1;run;
```

Figure 2.7

```
/* 6.2 Predict AvgPrice 201506-201606 */
proc autoreg data=forPred;
model AvgPrice = MonthNumber / nlag=(2) dwprob;  /* Prediction model with lag(2) */
output out=preds2 p=prediction2 lcl=lower2 ucl=upper2 pm=trend2; /* Adding new columns */
run;
proc print data=preds2;run;
```

Table 1.1

Group Name	Variables
Distance	Green_zone_km, railroad_km, railroad_station_walk_km, railroad_station_avto_km, school_km, public_transport_station_km, bus_terminal_avto_km, kremlin_km, park_km, metro_km_avto, metro_km_walk, industrial_km, bigroad1_km, big_road2_km, big_market_km, market_shop_km, fitness_km, stadium_km, basketball_km, university_km, workplaces_km, office_km, ice_rink_km, public_healthcare_km, big_church_km, swim_pool_km, shopping_centers_km
Number	Num_room, shopping_centers_raion, university_top_20_raion, office_raion, hospital_beds_raion, healthcare_centers_raion, max_floor, preschool_quota, raion_popul, children_preschool
Count	Build_count_brick, build_count_before_1920, build_count_block, build_count_1946.1970, build_count_1971.1995, build_count_frame, build_count_after_1995, build_count_wood, build_count_1921.1945
Population	Full_all, X16_29_all, X0_6_all, children_school, X7_14_all, X0_17_all, X0_13_all
Time	Timestamp, build_year, railroad_station_walk_min, railroad_station_avto_min, public_transport_station_min_walk, metro_min_avto, metro_min_walk
Area	Indust_part, full_sq, life_sq, kitch_sq, green_zone_part
Other	Material, floor
ID	ID_railroad_station_walk, id

Response	Price_doc

Table 1.2

The REG Procedure
Model: MODEL1
Dependent Variable: price_doc

Number of Observations Read	12883
Number of Observations Used	12883

X'X Inverse, Parameter Estimates, and SSE			
Variable	Intercept	full_sq	price_doc
Intercept	0.0006048305	-9.981837E-6	-936202.1445
full_sq	-9.981837E-6	1.8898977E-7	165036.75697
price_doc	-936202.1445	165036.75697	1.8095608E17

Covariance of Estimates		
Variable	Intercept	full_sq
Intercept	8496836842.1	-140227786.7
full_sq	-140227786.7	2654983.879

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	1.441196E17	1.441196E17	10258.9	<.0001
Error	12881	1.809561E17	1.404829E13		
Corrected Total	12882	3.250757E17			

Correlation of Estimates		
Variable	Intercept	full_sq
Intercept	1.0000	-0.9336
full_sq	-0.9336	1.0000

Root MSE	3748105	R-Square	0.4433
Dependent Mean	7780513	Adj R-Sq	0.4433
Coeff Var	48.17298		

Sequential Parameter Estimates	
Intercept	full_sq
7780513	0
-936202	165037

Table 1.3

The REG Procedure
Model: MODEL1
Dependent Variable: price_doc

Heteroscedasticity Consistent Covariance of Estimates		
Variable	Intercept	full_sq
Intercept	48145887315	-987945876.8
full_sq	-987945876.8	20542148.523

Test of First and Second Moment Specification		
DF	Chi-Square	Pr > ChiSq
2	199.14	<.0001

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Heteroscedasticity Consistent			Type I SS	Type II SS	Standardized Estimate	Squared Semi-partial Corr Type I	Squared Partial Corr Type I	Squared Semi-partial Corr Type II	Squared Partial Corr Type II	Tolerance	Variance Inflation	95% Confidence Limits		Heteroscedasticity Consistent 95% Confidence Limits	
						Standard Error	t Value	Pr > |t|													
Intercept	1	-936202	92178	-10.16	<.0001	219422	-4.27	<.0001	7.798903E17	1.449124E15	0					0	0	-1116885	-755519	-1366301	-506103
full_sq	1	165037	1629.41213	101.29	<.0001	4532.34470	36.41	<.0001	1.441196E17	1.441196E17	0.66584	0.44334	0.44334	0.44334	0.44334	1.00000	1.00000	161843	168231	156153	173921

Table 1.4

Parameter Estimates		
Parameter	DF	Estimate
Intercept	1	28065501
full_sq	1	196744
build_year	1	-12920
num_room	1	-844309
X0_17_all	1	14.755952
kremlin_km	1	-166444
product_type	1	-315062
big_market_raion	1	-821012

Table 1.5

Parameter Estimates					
	Parameter	DF	Estimate	Standard Error	t Value
	Intercept	1	28919545	4915664	5.88
*	full_sq	1	197710	3156.723064	62.63
*	build_year	1	-13351	2548.620549	-5.24
*	num_room	1	-866209	72704	-11.91
*	X0_17_all	1	15.083109	4.240004	3.56
*	kremlin_km	1	-166455	5130.106182	-32.45
*	product_type	1	-315947	135979	-2.32
*	big_market_raion	1	-830030	144600	-5.74
* Forced into the model by the INCLUDE= option					

Table 1.6

Parameter Estimates				
Parameter	DF	Estimate	Standard Error	t Value
Intercept	1	32356910	3773420	8.57
full_sq	1	200778	2753.110214	72.93
build_year	1	-15541	1920.473749	-8.09
num_room	1	-822448	65598	-12.54
X0_17_all	1	27.597161	4.501279	6.13
kremlin_km	1	-159224	4687.944557	-33.96
big_market_raion	1	-687817	124462	-5.53
f*b*n*X0*kr*pr*ra*bi	1	-0.000002418	0.000000637	-3.80

Table 1.7

The GLMSELECT Procedure

Data Set	WORK.TRAIN
Test Data Set	WORK.TEST
Dependent Variable	price_doc
Selection Method	LASSO
Stop Criterion	Cross Validation
Choose Criterion	Cross Validation
Cross Validation Method	Random
Cross Validation Fold	5
Effect Hierarchy Enforced	None
Random Number Seed	1

Root MSE	3485628
Dependent Mean	7772367
R-Square	0.5210
Adj R-Sq	0.5206
AIC	303947
AICC	303947
SBC	294238
ASE (Train)	1.213965E13
ASE (Test)	1.097487E13
CV PRESS	1.188483E17

The GLMSELECT Procedure

Data Set	WORK.TRAIN
Test Data Set	WORK.TEST
Dependent Variable	price_doc
Selection Method	Stepwise
Select Criterion	SBC
Stop Criterion	Cross Validation
Choose Criterion	Cross Validation
Cross Validation Method	Random
Cross Validation Fold	5
Number of Included Effects	7
Effect Hierarchy Enforced	None
Random Number Seed	1

Root MSE	3485606
Dependent Mean	7772367
R-Square	0.5210
Adj R-Sq	0.5206
AIC	303947
AICC	303947
SBC	294238
ASE (Train)	1.213949E13
ASE (Test)	1.097476E13
CV PRESS	1.188483E17

Effects: Intercept full_sq build_year num_room X0_17_all kremlin_km big_market_raion f*b*n*X0*kr*pr*ra*bi

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	7	1.724318E17	2.463311E16	2077.72
Error	12875	1.526439E17	1.185584E13	
Corrected Total	12882	3.250757E17		

Root MSE	3443231
Dependent Mean	7780513
R-Square	0.5304
Adj R-Sq	0.5302
AIC	400721
AICC	400721
SBC	387895

Table 2.1

OLS

The AUTOREG Procedure

Ordinary Least Squares Estimates

SSE	9.59068E12	DFE	45
MSE	2.13126E11	Root MSE	461656
SBC	1365.03878	AIC	1361.33848
MAE	351621.453	AICC	1361.61121
MAPE	100	HQC	1362.73093
Durbin-Watson	0.7524	Total R-Square	0.0000

Durbin-Watson Statistics

Order	DW	Pr < DW	Pr > DW
1	0.7524	<.0001	1.0000

Parameter Estimates

Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	2.087E-10	136857	0.00	1.0000
MonthNumber	1	-5.7E-12	4964	-0.00	1.0000

lag(1)

The AUTOREG Procedure

Yule-Walker Estimates

SSE	5.86109E12	DFE	44
MSE	1.33206E11	Root MSE	364975
SBC	1346.23506	AIC	1340.68462
MAE	273459.117	AICC	1341.24276
MAPE	138.3776	HQC	1342.77329
Durbin-Watson	2.1502	Transformed Regression R-Square	0.0001
		Total R-Square	0.3889

Durbin-Watson Statistics

Order	DW	Pr < DW	Pr > DW
1	2.1502	0.6422	0.3578

Parameter Estimates

Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	-9382	263869	-0.04	0.9718
MonthNumber	1	508.4399	9408	0.05	0.9571

lag(2)

The AUTOREG Procedure

Yule-Walker Estimates

SSE	7.49918E12	DFE	44
MSE	1.70436E11	Root MSE	412839
SBC	1357.80708	AIC	1352.25664
MAE	295488.99	AICC	1352.81478
MAPE	122.047492	HQC	1354.34531
Durbin-Watson	1.1810	Transformed Regression R-Square	0.0000
		Total R-Square	0.2181

Durbin-Watson Statistics

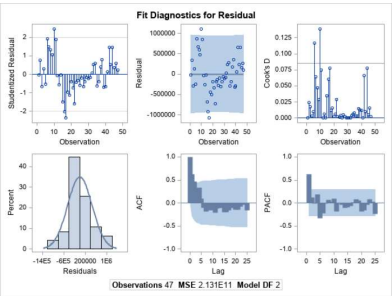
Order	DW	Pr < DW	Pr > DW
1	1.1810	0.0009	0.9991

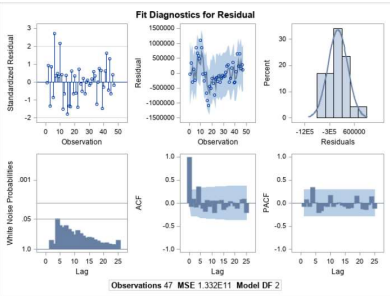
Parameter Estimates

Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	8629	208425	0.04	0.9672
MonthNumber	1	134.2758	7426	0.02	0.9857

Expected Autocorrelations

Lag	Autocorr
0	1.0000
1	0.0000
2	0.4620





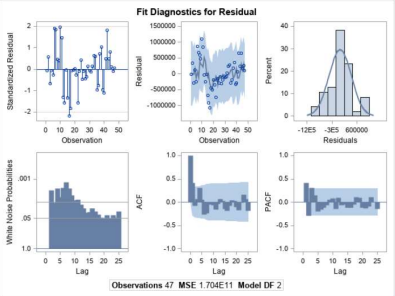


Table 2.2

Obs	prediction1	lower1	upper1	trend1	MonthNumber	monthYear	AvgPrice	Residual
1	8763.05	-1013970.22	1031496.32	8763.05	1	201108	5850000.00	-20621.04
2	8897.32	-1008895.76	1026690.41	8897.32	2	201109	6255310.06	338720.02
3	-4542.46	-859001.21	849916.30	9031.60	3	201110	5667466.10	-295092.93
4	161528.34	-691503.67	1014560.34	9165.88	4	201111	6140269.12	131741.08
46	310462.97	-551201.18	1172127.13	14805.46	46	201505	8210895.89	271669.90
47	83891.28	-779593.58	947376.14	14939.73	47	201506	8091600.95	106405.96
48	133733.21	-731643.47	999109.89	15074.01	48	201507	.	.
49	57461.35	-809877.79	924800.50	15208.29	49	201508	.	.
50	70157.49	-915239.70	1055554.69	15342.56	50	201509	.	.
51	34995.75	-954273.95	1024265.45	15476.84	51	201510	.	.
52	40933.02	-988393.74	1070259.77	15611.11	52	201511	.	.
53	24762.20	-1009707.60	1059232.00	15745.39	53	201512	.	.
54	27577.18	-1025896.03	1081050.40	15879.67	54	201601	.	.
55	20179.28	-1039259.32	1079617.89	16013.94	55	201602	.	.
56	21551.92	-1049857.89	1092961.72	16148.22	56	201603	.	.
57	18206.68	-1059738.53	1096151.89	16282.49	57	201604	.	.
58	18913.02	-1068373.82	1106199.86	16416.77	58	201605	.	.
59	17439.93	-1076818.88	1111698.74	16551.04	59	201606	.	.

Table 2.3

Obs	prediction2	lower2	upper2	trend2	MonthNumber	monthYear	AvgPrice	Residual
1	5879384.09	4856650.82	6902117.36	5879384.09	1	201108	5850000.00	-20621.04
2	5925487.36	4907694.28	6943280.45	5925487.36	2	201109	6255310.06	338720.02
3	5958016.58	5103557.83	6812475.33	5971590.64	3	201110	5667466.10	-295092.93
4	6170056.38	5317024.37	7023088.38	6017693.91	4	201111	6140269.12	131741.08
46	8249688.97	7388024.81	9111353.12	7954031.45	46	201505	8210895.89	271669.90
47	8069086.27	7205601.41	8932571.13	8000134.73	47	201506	8091600.95	106405.96
48	8164897.21	7299520.53	9030273.89	8046238.00	48	201507	.	.
49	8134594.34	7267255.19	9001933.49	8092341.28	49	201508	.	.
50	8193259.48	7207862.29	9178656.68	8138444.55	50	201509	.	.
51	8204066.74	7214797.04	9193336.44	8184547.83	51	201510	.	.
52	8255973.00	7226646.24	9285299.76	8230651.10	52	201511	.	.
53	8285771.19	7251301.39	9320240.99	8276754.37	53	201512	.	.
54	8334555.17	7281081.95	9388028.39	8322857.65	54	201601	.	.
55	8373126.27	7313687.66	9432564.87	8368960.92	55	201602	.	.
56	8420467.90	7349058.10	9491877.70	8415064.20	56	201603	.	.
57	8463091.66	7385146.45	9541036.87	8461167.47	57	201604	.	.
58	8509767.00	7422480.16	9597053.84	8507270.75	58	201605	.	.
59	8554262.91	7460004.10	9648521.72	8553374.02	59	201606	.	.