MSDS 6372 SBERBANK Project 1 - Daniel Serna & Laura Niederlander
October 10, 2018

Goal 1: 8 Pages (*60 pts*)

1. Introduction

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. Although the housing market is relatively stable in Russia, complex interactions between housing features such as number of bedrooms and location are enough to make pricing predictions complicated.

In this project, we are challenged to develop three models which use a broad spectrum of features to predict realty prices. An accurate prediction model will allow Sberbank to provide more certainty to their customers and value to their shareholders.

2. Description of the data

The data includes 292 attributes that include housing, market demographics, industry, transportation, education, religious locations, and recreation facility information to support the housing information. The full data dictionary is below in the appendix.

The data contains information about specific dwellings as well as the surrounding areas.

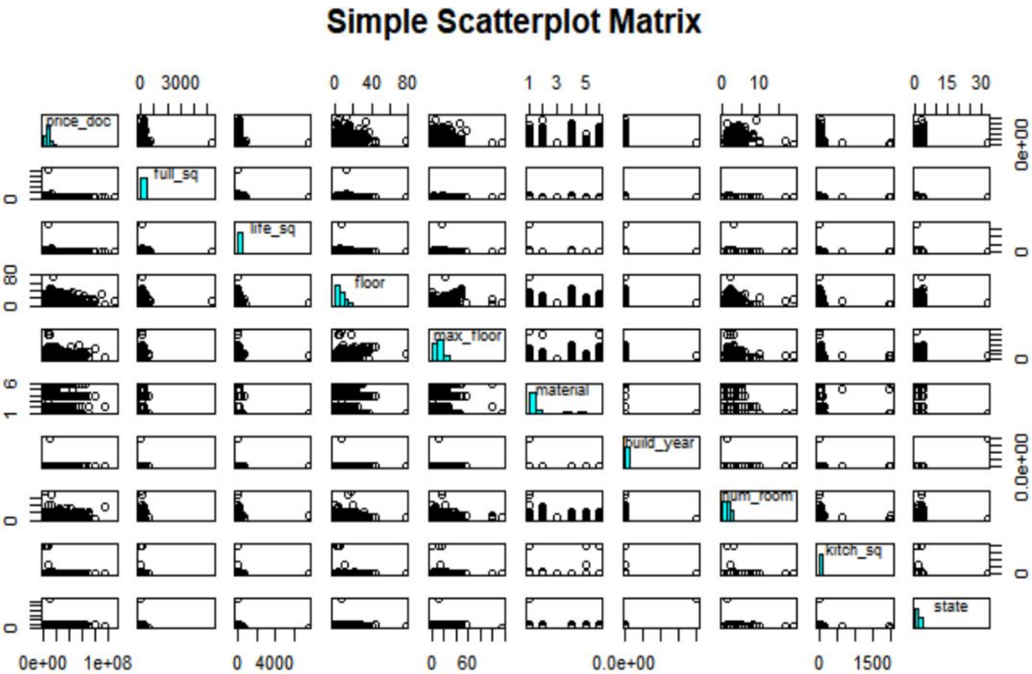
<u>Categories:</u>
Leisure (cafe_count, market_count, green_count, etc.)
Municipal infrastructure (big_road2_km, railroad_km, bus_terminal_avto_km, etc.)
demographics(young_*, work_*

<u>Potential subcategories:</u>
Transportation
Education
Health

3. Data Cleaning / Wrangling (any renaming of variables or standardizing of values.)

All values were imported as integers. Because of the NA value, many of the variable that were integers, were assigned to character format. This format caused issues with plot and reviewing the data that was numeric but assigned the wrong format type.

4. Exploratory Data Analysis (EDA).
(Complete with summary statistics, descriptions, tables and/or plots etc.)



Scatterplot was run on these data on the apartment attributes to review normality and linear relationships. The plot revealed that several attributes with extremely skewed and data to the right and left. We also observed that the price was extremely left skewed.

a. Outlier Identification and Handling

Outliers were discovered in several of the attributes,The following attributes showed a few outliers that were skewing the data; build_Year, Full_Sq, Kitch_SQ and Life_SQ, State. These outliers were removed from the data based on logical deduction. There are outliers in living data that was resolved with removing outliers. It includes:

Keep all records that meet the following criteria:
build_year >= 1691
build_year <= 2018
full_sq <= 5000
kitch_sq <= 1500
life_sq <= 1000
state <= 4
kitch_sq < 600
floor > 0
num_room > 0
max_floor > 0

Total records removed through this process were 905 outliers. Our net dataset has 29,084 observations.

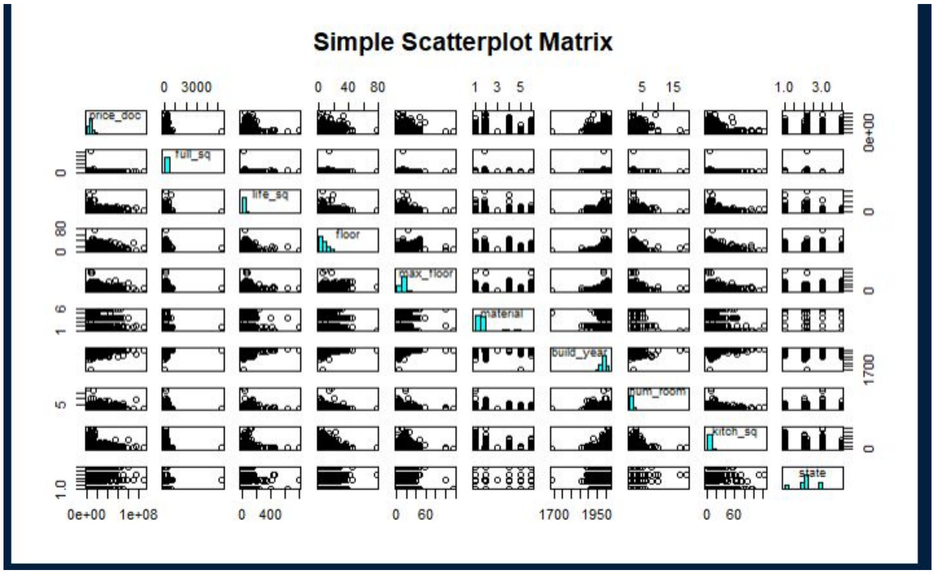
b. Missing value identification, summary and possible imputation (mean, median, regression.) This may also be considered "Data Wrangling".

The dataset included 262,233 NAs across most of the attributes within the dataset.
For the integer attributes, we applied the means value of that attributes to replace the NA values and converted factor data (Yes, No) into "1" and "0."

```
#convert yes/no values to 1/0
culture_objects_top_25 =="yes", 1, 0)
full_all =="yes", 1, 0)
incineration_raion =="yes", 1, 0)
oil_chemistry_raion =="yes", 1, 0)
radiation_raion =="yes", 1, 0)
railroad_terminal_raion =="yes", 1, 0)
big_market_raion =="yes", 1, 0)
nuclear_reactor_raion =="yes", 1, 0)
detention_facility_raion =="yes", 1, 0)
thermal_power_plant_raion =="yes", 1, 0)
water_1line =="yes", 1, 0)
big_road1_1line =="yes", 1, 0)
railroad_1line =="yes", 1, 0)

#convert product_type NAs to Investment
Substitute all NA in the Product_type to "Investment"
Substitue all NA in sub_area to Ajeroport

#apply column mean to NA values
```



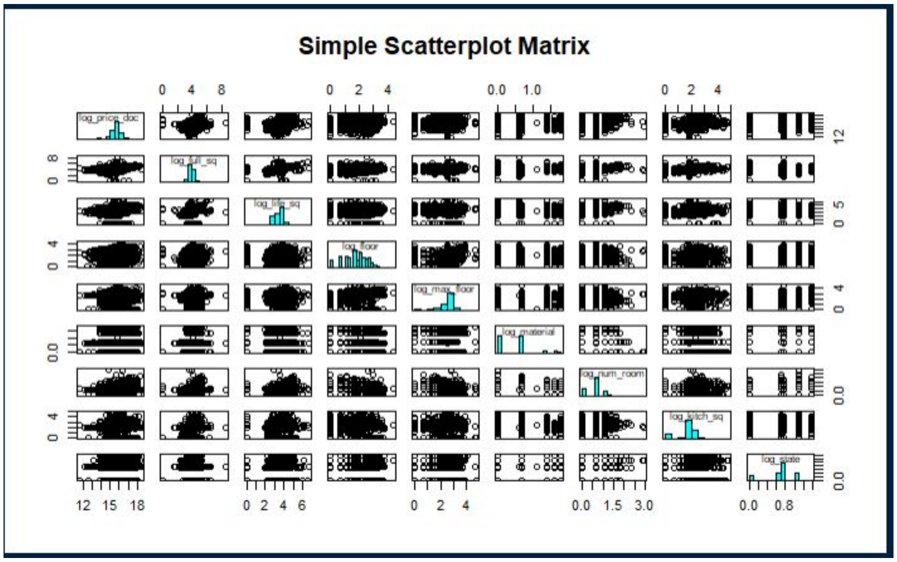
c. Multicollinearity

Based on the scatterplot matrix above, we do not see any evidence of multicollinearity. No variable seems to be predictive of the other variables.

d. Checking assumptions:

Several variables are skewed and will need to be normalized with transformation. It includes the Price_doc, full_sq, max_sq, life_sq, num_room, kitch_sq, The data does needs to be transformed.

logprice = log(trainData$price_doc)
logfull_sq = log(trainData$full_sq)
loglife = log(trainData$life_sq)
logfloor = log(trainData$floor)
logmaxfl = log(trainData$max_floor)
logmaterial = log(trainData$material)
lognumroom = log(trainData$num_room)
logkitsq = log(trainData$kitch_sq)
logstate = log(trainData$state)



e. High level variable selection would be included in the EDA. (Example: There are many potential explanatory variables. Running stepwise variable selection with a high entry and exit threshold will not provide a plausible final model, but may leave you with a smaller set of potential explanatory to work with.)

f. Anything else that might be appropriate in learning about the data before getting started. (Example: You might try interactions between explanatory variables in the EDA.)

We did notice some strange data points such as households with values of 0 for floors or living_area. We treated these as outliers, but it could be helpful to understand what the rationale was behind assigning these values. Perhaps this is representative of something we don't understand and our model could be affected.

5. Modeling

a. Our three models include: Stepwise, Backward, LASSO

Our stepwise model included the following parameters: (See full set in appendix)

Parameter	**DF**	**Estimate**	**Standard Error**	**t Value**
Intercept	1	-51301777	3469574	-14.79
id	1	67.611632	2.390739	28.28
full_sq	1	18573	597.436301	31.09
life_sq	1	61254	1348.118453	45.44

Our backward model included the following parameters:

Parameter	DF	Estimate	Standard Error	t Value
Intercept	1	-52680104	3486352	-15.11
id	1	66.368133	2.39994	27.65
full_sq	1	18523	596.515124	31.05
life_sq	1	61206	1347.65137	45.42

The parameters with a higher absolute value for t-value had a greater significance for predicting the price_doc response. Interestingly the id parameter shows significance even though logically it should have no effect on the price_doc variable. We achieved an r-square value of 0.4524 for stepwise selection and an r-square value of 0.4550 for backward selection.

ii. A model with LASSO estimation and selection.

Our LASSO model contained the following parameters:

Parameter	DF	Estimate
Intercept	1	5222251
life_sq	1	22348
num_room	1	625806
sadovoe_km	1	-3485.263

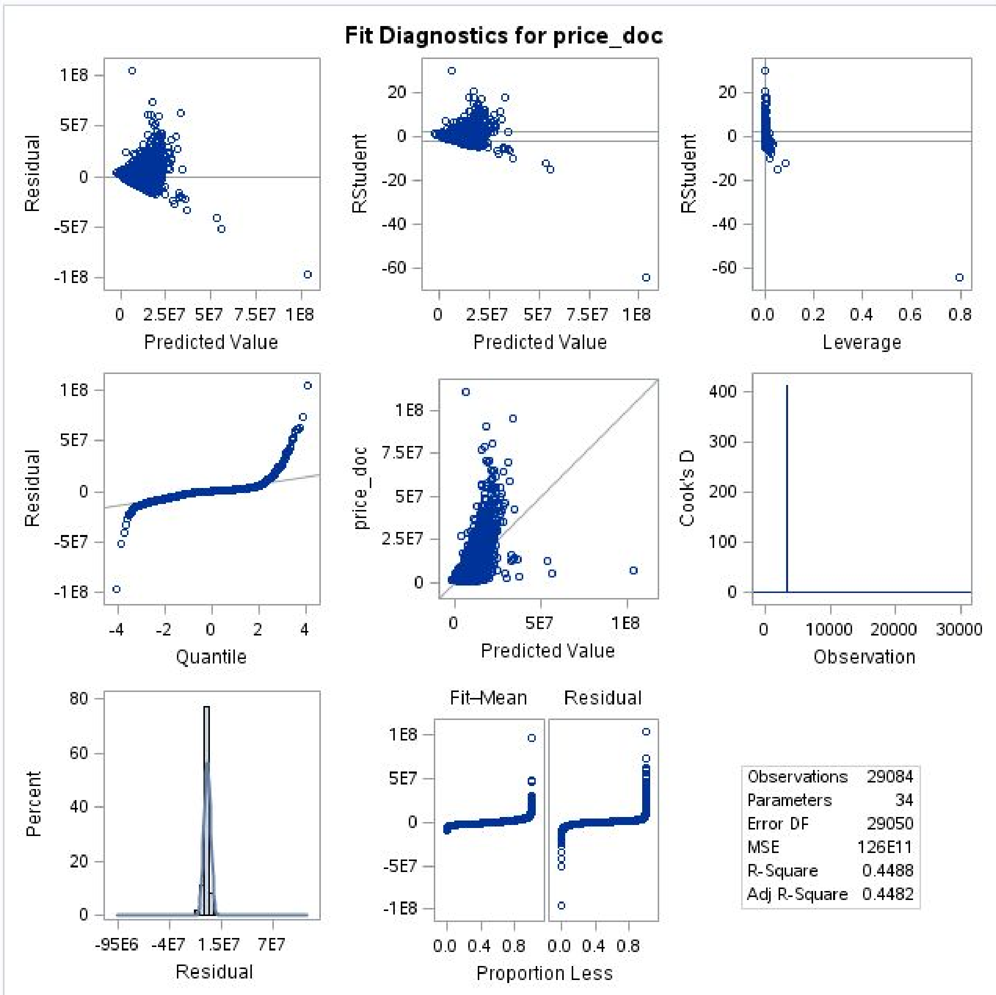
iiii. A model of your choice. This may be using another OLS or LASSO model or custom model, etc.

Our custom model contained the following parameters:

Source	DF	Type I SS	Mean Square	F Value	Pr > F
id	1	1.08E+16	1.08E+16	857.98	<.0001
full_sq	1	7.48E+16	7.48E+16	5937.68	<.0001
life_sq	1	5.63E+16	5.63E+16	4473.77	<.0001

floor	1	3.12E+15	3.12E+15	247.89	<.0001

b. You need to address the assumptions with **respect to the residuals.** (Normally distributed around 0 with constant standard deviation.)



We can see that the our assumptions for residuals is not perfect. We see to have skewness when looking at the q-q plot and our residuals do not appear to be randomly scattered. Our residuals do appear to be normally distributed. With almost 30,000 observations, we will assume our sample size is large enough to proceed with caution.

c. For each model you need to **conduct an internal and external cross validation**.

We do not see any difference in CV Press score with external cross validation techniques.

Analysis of Variance				
Source	DF	Sum of Squares	Mean Square	F Value
Model	53	3.019633E17	5.69742E15	457.32
Error	29030	3.616608E17	1.245817E13	
Corrected Total	29083	6.636241E17		

Root MSE	3529614
Dependent Mean	7128513
R-Square	0.4550
Adj R-Sq	0.4540
AIC	906121
AICC	906122
SBC	877482
CV PRESS	5.49033E17

Cross Validation Details			
	Observations		
Index	Fitted	Left Out	CV PRESS
1	26175	2909	3.62815E16
2	26175	2909	3.51813E16
3	26175	2909	3.69421E16
4	26175	2909	3.26004E16
5	26176	2908	3.25042E16
6	26176	2908	3.31116E16
7	26176	2908	3.07993E16
8	26176	2908	2.33253E17
9	26176	2908	4.39061E16
10	26176	2908	3.44539E16
Total			5.49033E17

d. You should compare the models using the AIC, SBC, Interval k-fold cross validation (you pick k), external cross validation. For external cross validation you will have to subset the train data set into modeling and test data sets.

Table A-C

Table A: Stepwise include 40 attributes within the fit criteria which yielded an R squared value of 0.47. Many of the attributes contributed less than 0.01% to the total model.

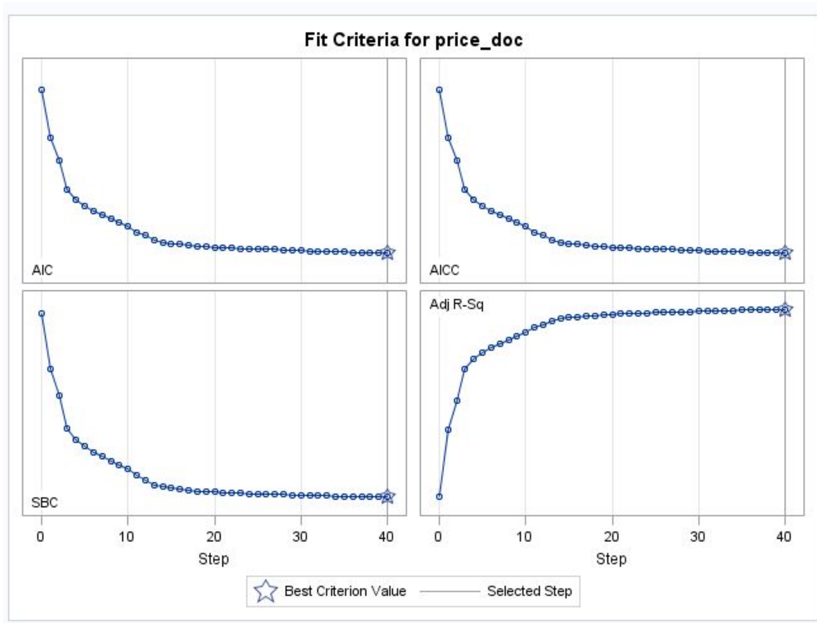


Table B: LASSO - Fit criteria chart shows only 3 steps to reach 0.21 R squared. The limited attributes that contributed to less than the stepwise or backward model fits.

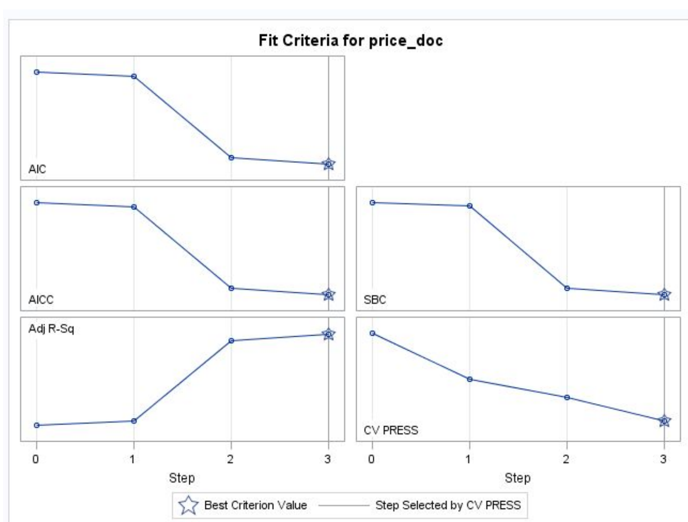
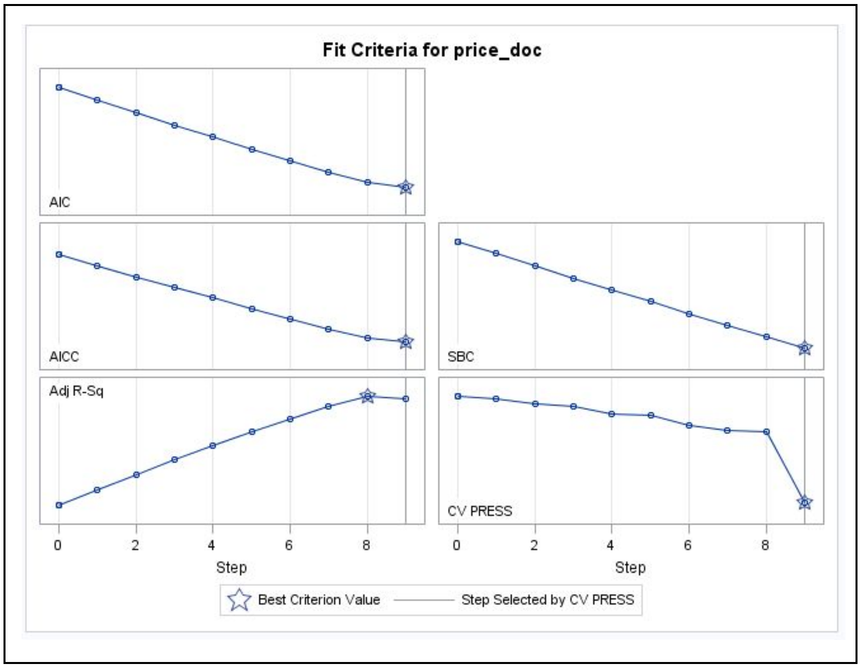


Table C: Backward - The backward fit criteria when through 9 steps to reach the R square value of 0.454.



	Stepwise	Backward	Lasso
AIC	906234	906122	919849
SBC	877488	877482	890796
CV Press		5.49x10^17	4.6006x10^17
Adj R^2	0.4517	.4540	0.1232

6. Prediction

All of our indicators point to backward selection being the best model for us. We have the highest adjusted r^2 with backward selection, the lowest AIC and SBC. We also have the highest CV press score with backward selection.

Goal 2: ≤ 3 Pages (30 pts)
Introduction

In this exercise, we will review the prediction of the pricing based on the month's average sales price. The aggregate of these prices will include the highs and lows of all the variables within the dataset.

Data Wrangling:

Using SAS, the results dataset was subset to the three columns necessary for the time series output. The ID, timestamp and price_doc were created in a new table. The timestamp data was concatenated from the dataset to create Month, Day, and Year. Once these were created, the MonthYear column was created. An additional dataset was created with Month, monthYear, and AvgPrice.

	id	price_doc	timestamp
1	1	5850000	2011-08-20
2	2	6000000	2011-08-23
3	3	5700000	2011-08-27
4	4	13100000	2011-09-01
5	5	16331452	2011-09-05
6	6	9100000	2011-09-06
7	7	5500000	2011-09-08
8	8	2000000	2011-09-09
9	9	5300000	2011-09-10
10	10	2000000	2011-09-13
11	11	[illegible]	[illegible]

Subset of data from the Results Table

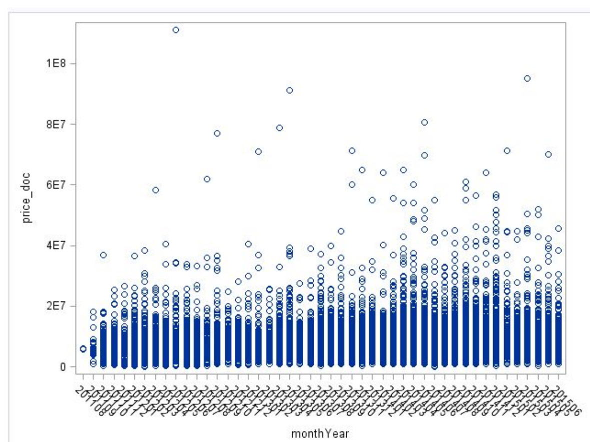
id	price_doc	timestamp	year	month	day	monthYear
1	5850000	2011-08-20	2011	08	20	201108
2	6000000	2011-08-23	2011	08	23	201108
3	5700000	2011-08-27	2011	08	27	201108
4	13100000	2011-09-01	2011	09	01	201109
5	16331452	2011-09-05	2011	09	05	201109
6	9100000	2011-09-06	2011	09	06	201109
7	5500000	2011-09-08	2011	09	08	201109
8	2000000	2011-09-09	2011	09	09	201109
9	5300000	2011-09-10	2011	09	10	201109
10	2000000	2011-09-13	2011	09	13	201109
11	4650000	2011-09-16	2011	09	16	201109
12	4800000	2011-09-16	2011	09	16	201109
13	5100000	2011-09-17	2011	09	17	201109

New subset with timestamp expanded to Month, Day , Year.

Using SAS, the dataset with aggregated based on Month to show all average price sales based on average by month.

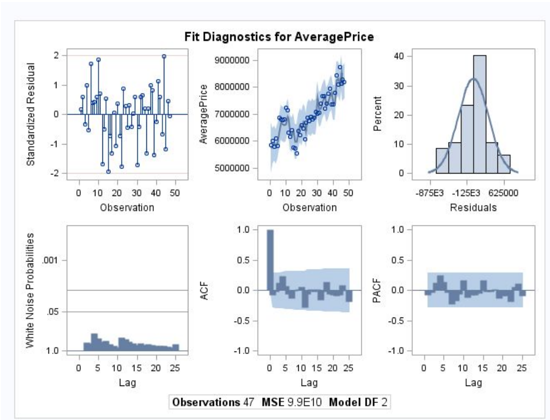
	monthYear	AveragePrice
1	201108	5850000
2	201109	6007329.5385
3	201110	5808421.7324
4	201111	6115835.4054
5	201112	5824305.6318
6	201201	6884006.0467
7	201202	6807455.6484
8	201203	6765833.5158

Average Monthly Pricing

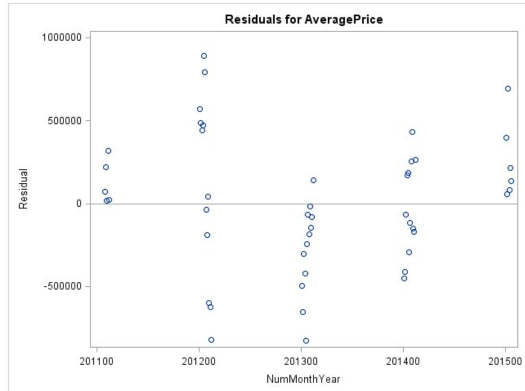


Plot of Price_doc and MonthYear data

Model the residual series



The residual plots indicate that the average prices have many outliers but it does appear to be normally distributed.



This residual plot indicates the wide span of average prices far above and below the zero axes.

Fit a simple linear regression model with price_doc as the response variable and Month_Number as the explanatory variable.

For every one unit spend on realty, the price increases 5,745 per month.

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-1149699006	100180608	-11.48	<.0001
NumMonthYear	1	5745.56219	497.64155	11.55	<.0001

The average price appears to have a positive linear relationship with months/year. The prices fall above and below the confidence levels in all timeframes except in 2015 where the prices are above the confidence intervals. (Table C and D)

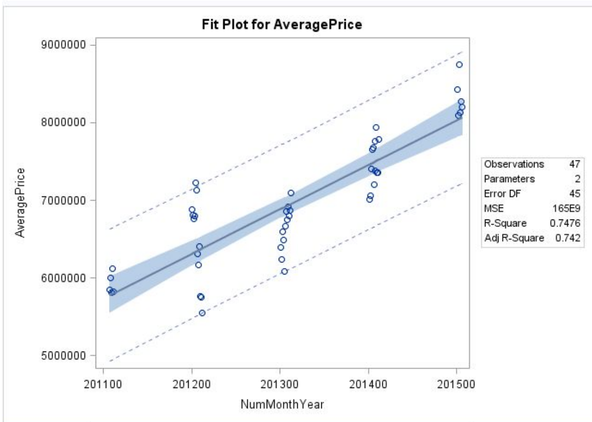


Table C

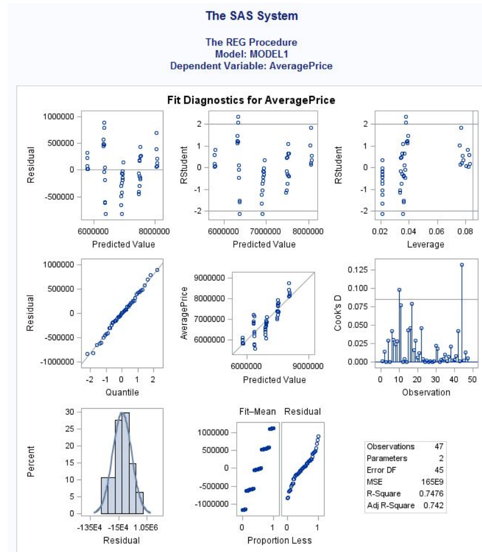


Table D

Table C: Fit Plot for Average Price and Month/Year. Confidence interval is very small and observations are outside the CI but not outside the forecast.
Table D: Residual plots for Average Price and MonthYear. Cook's D indicates all values are within the -2 and 2 range. Single Mode distribution indicates normal distribution. Values appear to be linear.

The autocorrelation structure based on the Yule-Walker estimates shows the Durbin-Watson statistic has a higher AIC and SBC value at 1326.77 and 1332.32 verses the partial autocorrelation with an AIC at 1264.87 and SBC at 1219.57. The Durbin-Watson is at 21.156 with a r squared at 0.85. This would suggest that more of the model is accounted for in the Durbin-Watson statistics than the ANOVA. Table E(below).

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	1	2.196997E13	2.196997E13	133.30
Error	45	7.41669E12	1.648153E11	
Corrected Total	46	2.938666E13		

Root MSE	405975
Dependent Mean	6944398
R-Square	0.7476
Adj R-Sq	0.7420
AIC	1264.87650
AICC	1265.43464
SBC	1219.57679

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value
Intercept	1	-1149699006	100180608	-11.48
NumMonthYear	1	5745.562192	497.641549	11.55

The AUTOREG Procedure

Yule-Walker Estimates

SSE	4.3562E12	DFE	44
MSE	9.90045E10	Root MSE	314650
SBC	1332.32549	AIC	1326.77505
MAE	250045.71	AICC	1327.33319
MAPE	3.65541222	HQC	1328.86372
Durbin-Watson	2.1563	Transformed Regression R-Square	0.5033
		Total R-Square	0.8518

Durbin-Watson Statistics

Order	DW	Pr < DW	Pr > DW
1	2.1563	0.6778	0.3222

5. Predict the Residual for next year (June2015-June 2016)

Conclusion:

Sberbank, Russia's oldest and largest bank requested an analysis and forecast of future realty prices in the housing market. There were many variables to consider and the data was not completely populated. With intelligent and caution, we took great care to review each observation and determine how we could provide the best analysis forecast possible to meet the needs of the client. We were able to determine that over the time series provided the pricing is on the rise overall.

There are many outliers for the data that fall outside the confidence intervals. We have assessed the correlations between the 250 variables provided. Our forecast took into account up to 50 variables that had the most significant impact on the pricing data.

We have provided a forecast based on the overall model fit that was developed using the stepwise, backward, and Lasso procedures. Two of these models, stepwise and backward, accounted for 45% of the variables.

Additionally, we have forecasted the next year between July 2015 through June 2016 based on a 95% confidence interval.

Appendix 1: Data Dictionary 3 pts

Main Attributions	Descriptions

price_doc	**sale price (this is the target variable)**
id	**transaction id**
timestamp	**date of transaction**
full_sq	**total area in square meters, including loggias, balconies and other non-residential areas**
life_sq	**living area in square meters, excluding loggias, balconies and other non-residential areas**
floor	**for apartments, floor of the building**
max_floor	**number of floors in the building**
material	**wall material (1,2,3,4,5,6,NA)**
build_year	**year built**
num_room	**number of living rooms**
kitch_sq	**kitchen area**
state	**apartment condition (1,2,3,4,33,NA)**
product_type	**owner-occupier purchase or investment**
sub_area	**name of the district (string variable)**

Other Attributes used for analysis include:
area_m raion_popul green_zone_part indust_part children_preschool preschool_quota
preschool_education_centers_raion children_school school_quota school_education_centers_raion
school_education_centers_top_20_raion

hospital_beds_raion healthcare_centers_raion
university_top_20_raion sport_objects_raion additional_education_raion culture_objects_top_25
culture_objects_top_25_raion shopping_centers_raion office_raion
thermal_power_plant_raion incineration_raion oil_chemistry_raion
radiation_raion railroad_terminal_raion big_market_raion
nuclear_reactor_raiondetention_facility_raion
ID_metro metro_min_avto metro_km_avto
metro_min_walk metro_km_walk kindergarten_km school_km
park_km green_zone_km industrial_km
water_treatment_km cemetery_km incineration_km railroad_station_walk_km
railroad_station_walk_min ID_railroad_station_walk
railroad_station_avto_km railroad_station_avto_min
ID_railroad_station_avto
public_transport_station_km public_transport_station_min_walk
Water_km water_1line
Mkad_km ttk_km sadovoe_km bulvar_ring_km kremlin_km
big_road1_km ID_big_road1 big_road1_1line big_road2_km
ID_big_road2 railroad_km railroad_1line zd_vokzaly_avto_km
ID_railroad_terminal bus_terminal_avto_km ID_bus_terminal
oil_chemistry_km nuclear_reactor_km radiation_km power_transmission_line_km
thermal_power_plant_km
ts_km big_market_km market_shop_km fitness_km swim_pool_km
ice_rink_km stadium_km basketball_km hospice_morgue_km detention_facility_km

public_healthcare_km university_km workplaces_km shopping_centers_km office_km
additional_education_km preschool_km big_church_km church_synagogue_km mosque_km
theater_km museum_km exhibition_km catering_km

full_all male_f female_f young_all young_male young_female
work_all work_male work_female ekder_all ekder_male ekder_female

0_6_all 0_6_male0_6_female 7_14_all 7_14_male
7_14_female 0_17_all 0_17_male 0_17_female 16_29_all
16_29_male 16_29_female 0_13_all 0_13_male
0_13_female raion_build_count_with_material_info
build_count_before_1920
build_count_1921-1945 build_count_1946-1970 build_count_1971-1995 build_count_after_1995

build_count_block build_count_wood build_count_frame
build_count_brick build_count_monolith build_count_panel
build_count_foam build_count_slag build_count_mix
raion_build_count_with_builddate_info

ID_metro metro_min_avto metro_km_avto
metro_min_walk metro_km_walk kindergarten_km school_km
park_km green_zone_km industrial_km
water_treatment_km cemetery_km incineration_km railroad_station_walk_km
railroad_station_walk_min ID_railroad_station_walk
railroad_station_avto_km railroad_station_avto_min
ID_railroad_station_avto
public_transport_station_km public_transport_station_min_walk
water_km water_1line
mkad_km ttk_km sadovoe_km bulvar_ring_km kremlin_km
big_road1_km ID_big_road1 big_road1_1line big_road2_km
ID_big_road2 railroad_km railroad_1line zd_vokzaly_avto_km
ID_railroad_terminal bus_terminal_avto_km ID_bus_terminal
oil_chemistry_km nuclear_reactor_km radiation_km power_transmission_line_km
thermal_power_plant_km
ts_km big_market_km market_shop_km fitness_km swim_pool_km
ice_rink_km stadium_km basketball_km hospice_morgue_km detention_facility_km
public_healthcare_km university_km workplaces_km shopping_centers_km office_km
additional_education_km preschool_km big_church_km church_synagogue_km mosque_km
theater_km
museum_km exhibition_km catering_km ecology

green_part_500 prom_part_500 office_count_500 office_sqm_500
trc_count_500 trc_sqm_500 cafe_count_500 cafe_sum_500_min_price_avg
cafe_sum_500_max_price_avg
cafe_avg_price_500 cafe_count_500_na_price cafe_count_500_price_500 cafe_count_500_price_1000
cafe_count_500_price_1500 cafe_count_500_price_2500 cafe_count_500_price_4000
cafe_count_500_price_high

big_church_count_500 church_count_500 mosque_count_500 leisure_count_500 sport_count_500
market_count_500

green_part_1000 prom_part_1000 office_count_1000 office_sqm_1000 trc_count_1000
trc_sqm_1000 cafe_count_1000 cafe_sum_1000_min_price_avg cafe_sum_1000_max_price_avg
cafe_avg_price_1000 cafe_count_1000_na_price cafe_count_1000_price_500
cafe_count_1000_price_1000 cafe_count_1000_price_1500 cafe_count_1000_price_2500
cafe_count_1000_price_4000 cafe_count_1000_price_high

big_church_count_1000 church_count_1000 mosque_count_1000 leisure_count_1000
sport_count_1000 market_count_1000

green_part_1500 prom_part_1500 office_count_1500
office_sqm_1500 trc_count_1500 trc_sqm_1500
cafe_count_1500 cafe_sum_1500_min_price_avg cafe_sum_1500_max_price_avg cafe_avg_price_1500

```
cafe_count_1500_na_price  cafe_count_1500_price_500 cafe_count_1500_price_1000
cafe_count_1500_price_1500
cafe_count_1500_price_2500       cafe_count_1500_price_4000       cafe_count_1500_price_high

big_church_count_1500    church_count_1500       mosque_count_1500      leisure_count_1500
sport_count_1500
market_count_1500


green_part_2000   prom_part_2000   office_count_2000 office_sqm_2000
trc_count_2000    trc_sqm_2000     cafe_count_2000   cafe_sum_2000_min_price_avg
cafe_sum_2000_max_price_avg      cafe_avg_price_2000       cafe_count_2000_na_price
cafe_count_2000_price_500
cafe_count_2000_price_1000       cafe_count_2000_price_1500       cafe_count_2000_price_2500
cafe_count_2000_price_4000       cafe_count_2000_price_high

big_church_count_2000    church_count_2000       mosque_count_2000      leisure_count_2000
sport_count_2000 market_count_2000       green_part_3000   prom_part_3000   office_count_3000
office_sqm_3000  trc_count_3000   trc_sqm_3000     cafe_count_3000
cafe_sum_3000_min_price_avg      cafe_sum_3000_max_price_avg      cafe_avg_price_3000
cafe_count_3000_na_price
cafe_count_3000_price_500 cafe_count_3000_price_1000       cafe_count_3000_price_1500
cafe_count_3000_price_2500
cafe_count_3000_price_4000       cafe_count_3000_price_high

big_church_count_3000    church_count_3000       mosque_count_3000      leisure_count_3000
sport_count_3000 market_count_3000       green_part_5000   prom_part_5000   office_count_5000
office_sqm_5000  trc_count_5000   trc_sqm_5000     cafe_count_5000
cafe_sum_5000_min_price_avg      cafe_sum_5000_max_price_avg      cafe_avg_price_5000
cafe_count_5000_na_price
cafe_count_5000_price_500 cafe_count_5000_price_1000       cafe_count_5000_price_1500
cafe_count_5000_price_2500       cafe_count_5000_price_4000       cafe_count_5000_price_high

big_church_count_5000    church_count_5000       mosque_count_5000      leisure_count_5000
sport_count_5000
market_count_5000          price_doc ;
```

Table 1-3. GLM - Custom Model - Included 33 degrees of freedom with all 33 attributes showing signficance at p-value (<.0001). R-Squared was 0.448. Durbin-Watson D was at 1.99 and Press Stat at 5.8.

Tables 1-3 (GLM Custom)

The SAS System

The GLM Procedure

Dependent Variable: price_doc

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	33	2.9783713E17	9.0253676E15	716.78	<.0001
Error	29050	3.6578693E17	1.2591633E13		
Corrected Total	29083	6.6362406E17			

R-Square	Coeff Var	Root MSE	price_doc Mean
0.448804	49.77853	3548469	7128513

Sum of Residuals	-0.048219757
Sum of Squared Residuals	3.6578693E17
Sum of Squared Residuals - Error SS	-5809920
PRESS Statistic	5.8050815E17
First Order Autocorrelation	0.004376473
Durbin-Watson D	1.9912439492

Source	DF	Type I SS	Mean Square	F Value	Pr > F
id	1	1.080338E16	1.080338E16	857.98	<.0001
full_sq	1	7.4765103E16	7.4765103E16	5937.68	<.0001
life_sq	1	5.6332027E16	5.6332027E16	4473.77	<.0001
floor	1	3.1213844E15	3.1213844E15	247.89	<.0001
max_floor	1	4.7637413E14	4.7637413E14	37.83	<.0001
build_year	1	5.1351822E15	5.1351822E15	407.82	<.0001
num_room	1	3.2741915E16	3.2741915E16	2600.29	<.0001
kitch_sq	1	1.1153881E16	1.1153881E16	885.82	<.0001
state	1	3.8655981E15	3.8655981E15	307.00	<.0001
indust_part	1	1.7688002E15	1.7688002E15	140.47	<.0001
children_preschool	1	7.6569888E15	7.6569888E15	608.10	<.0001
preschool_quota	1	3.4097966E16	3.4097966E16	2707.99	<.0001

The GLMSELECT Procedure

LASSO Selection Summary				
Step	Effect Entered	Effect Removed	Number Effects In	CV PRESS
0	Intercept		1	6.63647E17
1	num_room		2	5.55773E17
2	life_sq		3	5.13647E17
3	sadovoe_km		4	4.60062E17*
* Optimal Value of Criterion				

Lasso effects only includes 3 df and stopping at 0.1233 of the R Square value.

Analysis of Variance				
Source	DF	Sum of Squares	Mean Square	F Value
Model	3	8.181186E16	2.727062E16	1363.03
Error	29080	5.818122E17	2.00073E13	
Corrected Total	29083	6.636241E17		

Root MSE	4472952
Dependent Mean	7128513
R-Square	0.1233
Adj R-Sq	0.1232
AIC	919849
AICC	919849
SBC	890796
CV PRESS	4.60062E17

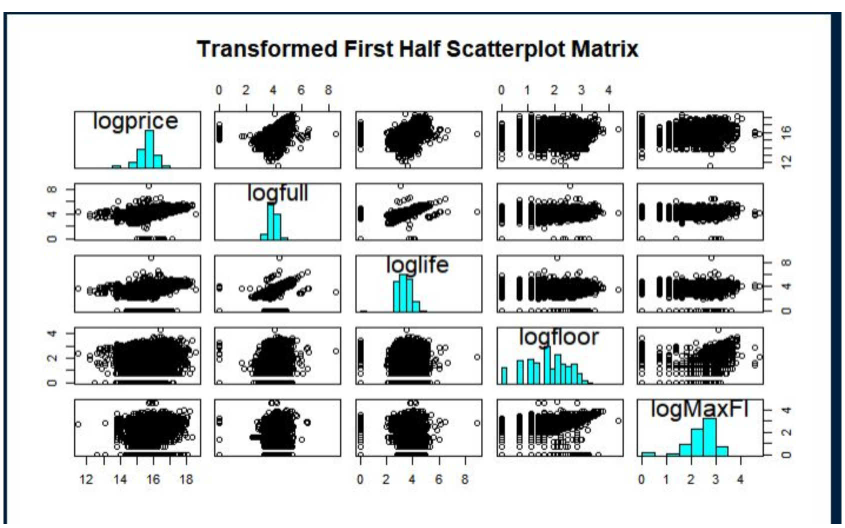
Lasso ANOVA

Analysis of Variance				
Source	DF	Sum of Squares	Mean Square	F Value
Model	53	3.019633E17	5.69742E15	457.32
Error	29030	3.616608E17	1.245817E13	
Corrected Total	29083	6.636241E17		

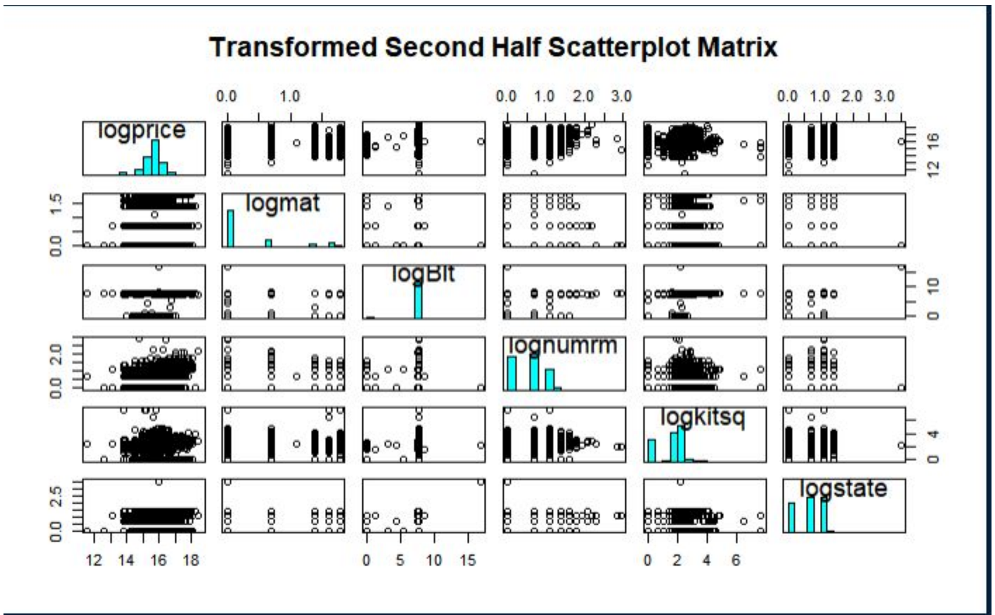
Root MSE	3529614
Dependent Mean	7128513
R-Square	0.4550
Adj R-Sq	0.4540
AIC	906121
AICC	906122
SBC	877482
CV PRESS	5.49033E17

Backward effects included 53 degrees of freedom. Reaching an R-Sq value of 0.455.

Log data scatterplot Matrix on key measures: price, full_sq, life_sq, floor, and max_sq. The data showed a more normal distribution after the log transformation.



Additional fields were logged: Material, built_year, num_room, kitch_sq, state. These were less impactful than the previous charts. These fields were returned to the regular form.



Appendix 2: Code 7pts

Code in R: (Data Wrangling)

```
---
title: "Group Project One"
author: "Daniel Serna and Laura Niederlander"
date: "September 16, 2018"
output: html_document
---

```{r installPackages}
if(!require(tidyverse)) install.packages("tidyverse")
if(!require(sqldf)) install.packages("sqldf")
if(!require(glmnet)) install.packages("glmnet")
if(!require(randomForest)) install.packages("randomForest")
```
```{r importData}
trainData <- read.csv("train.csv")
predictionData <- read.csv("predictionData.csv")
head(trainData)
```

```{r addUtilityFunctions}
panel.hist <- function(x, ...)
{
   usr <- par("usr"); on.exit(par(usr))
```

```
%web_drop_table(data);


FILENAME REFFILE
'/home/dserna0/Code/6372/GroupProject/subsetCleaned.csv';

PROC IMPORT DATAFILE=REFFILE
        DBMS=CSV
        OUT=data;
        GETNAMES=YES;
RUN;

PROC CONTENTS DATA=data; RUN;
```
```

<table>
<tr>
<td>

```
    par(usr = c(usr[1:2], 0, 1.5) )
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks; nB <- length(breaks)
    y <- h$counts; y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}
```

```
```{r removeOutliers}
trainDataCleaned <- trainData
trainDataCleaned <-
trainDataCleaned[which(trainDataCleaned$build_year >= 1691
| is.na(trainDataCleaned$build_year)),]
trainDataCleaned <-
trainDataCleaned[which(trainDataCleaned$build_year <= 2018
| is.na(trainDataCleaned$build_year)),]
trainDataCleaned <-
trainDataCleaned[which(trainDataCleaned$full_sq <= 5000)|
is.na(trainDataCleaned$full_sq),]
trainDataCleaned <-
trainDataCleaned[which(trainDataCleaned$kitch_sq <= 1500|
is.na(trainDataCleaned$kitch_sq)),]
trainDataCleaned <-
trainDataCleaned[which(trainDataCleaned$life_sq <= 1000|
is.na(trainDataCleaned$life_sq)),]
trainDataCleaned <-
trainDataCleaned[which(trainDataCleaned$state <= 4|
is.na(trainDataCleaned$state)),]
trainDataCleaned <-
trainDataCleaned[which(trainDataCleaned$kitch_sq < 600|
is.na(trainDataCleaned$kitch_sq)),]
trainDataCleaned <-
trainDataCleaned[which(trainDataCleaned$floor > 0|
is.na(trainDataCleaned$floor)),]
trainDataCleaned <-
trainDataCleaned[which(trainDataCleaned$num_room > 0 |
is.na(trainDataCleaned$num_room)),]
trainDataCleaned <-
trainDataCleaned[which(trainDataCleaned$max_floor > 0|
is.na(trainDataCleaned$max_floor)),]
```

```{r dataCleanup}

#convert yes/no values to 1/0
trainDataCleaned$culture_objects_top_25 <-
ifelse(trainDataCleaned$culture_objects_top_25 =="yes", 1, 0)
trainDataCleaned$full_all <- ifelse(trainDataCleaned$full_all
=="yes", 1, 0)
trainDataCleaned$incineration_raion <-
ifelse(trainDataCleaned$incineration_raion =="yes", 1, 0)
trainDataCleaned$oil_chemistry_raion <-
ifelse(trainDataCleaned$oil_chemistry_raion =="yes", 1, 0)
trainDataCleaned$radiation_raion <-
ifelse(trainDataCleaned$radiation_raion =="yes", 1, 0)
trainDataCleaned$railroad_terminal_raion <-
ifelse(trainDataCleaned$railroad_terminal_raion =="yes", 1, 0)
trainDataCleaned$big_market_raion <-
ifelse(trainDataCleaned$big_market_raion =="yes", 1, 0)
trainDataCleaned$nuclear_reactor_raion <-
ifelse(trainDataCleaned$nuclear_reactor_raion =="yes", 1, 0)
trainDataCleaned$detention_facility_raion <-
ifelse(trainDataCleaned$detention_facility_raion =="yes", 1, 0)
```

</td>
<td>

```
%web_open_table(data);

%web_drop_table(predictionData);


FILENAME REFFILE
'/home/dserna0/Code/6372/GroupProject/predictionDat
aSubsetCleaned.csv';

PROC IMPORT DATAFILE=REFFILE
        DBMS=CSV
        OUT=predictionData;
        GETNAMES=YES;
RUN;

PROC CONTENTS DATA=predictionData; RUN;


%web_open_table(predictionData);

proc sgscatter data=data;
plot price_doc*full_sq;
run;

proc corr data=data;
run;

proc glm data=data;
model price_doc = id full_sq life_sq floor max_floor
build_year num_room
kitch_sq state indust_part children_preschool
preschool_quota
university_top_20_raion
radiation_raion         build_count_block
build_count_slag kindergarten_km green_zone_km
mkad_km sadovoe_km kremlin_km railroad_km
railroad_1line thermal_power_plant_km
big_market_km office_km mosque_count_3000
green_part_5000
cafe_count_5000 cafe_avg_price_5000
prom_part_2000 office_count_3000
cafe_count_3000 /cli;
run;
```

</td>
</tr>
</table>
```

<table>
<tr>
<td>

```
trainDataCleaned$thermal_power_plant_raion <-
ifelse(trainDataCleaned$thermal_power_plant_raion =="yes",
1, 0)
trainDataCleaned$water_1line <-
ifelse(trainDataCleaned$water_1line =="yes", 1, 0)
trainDataCleaned$big_road1_1line <-
ifelse(trainDataCleaned$big_road1_1line =="yes", 1, 0)
trainDataCleaned$railroad_1line <-
ifelse(trainDataCleaned$railroad_1line =="yes", 1, 0)


#convert product_type NAs to Investment
trainDataCleaned[is.na(trainDataCleaned[,which(names(trainD
ataCleaned) == "product_type")]),
which(names(trainDataCleaned) == "product_type")] <-
"Investment"

#convert sub_area NAs to Ajeroport
trainDataCleaned[is.na(trainDataCleaned[,which(names(trainD
ataCleaned) == "sub_area")]), which(names(trainDataCleaned)
== "sub_area")] <- "Ajeroport"

#exclude not numeric columns for NA cleanup.
columnsToExclude <- names(trainDataCleaned) %in%
c("timestamp", "product_type", "sub_area", "ecology")
subsetCleaned <- trainDataCleaned[!columnsToExclude]

#apply column mean to NA values
for(i in 1:ncol(subsetCleaned)){
  subsetCleaned[is.na(subsetCleaned[,i]), i] <-
mean(subsetCleaned[,i], na.rm = TRUE)
}

#add non numeric columns back in.
subsetCleaned$timestamp <- trainDataCleaned$timestamp
subsetCleaned$product_type <-
trainDataCleaned$product_type
subsetCleaned$sub_area <- trainDataCleaned$sub_area
subsetCleaned$ecology <- trainDataCleaned$ecology
```

```{r removeOutliersPredictionData}
#predictionDataCleaned <- predictionData
#predictionDataCleaned <-
predictionDataCleaned[which(predictionDataCleaned$build_ye
ar >= 1691 | is.na(predictionDataCleaned$build_year)),]
#predictionDataCleaned <-
predictionDataCleaned[which(predictionDataCleaned$build_ye
ar <= 2018 | is.na(predictionDataCleaned$build_year)),]
#predictionDataCleaned <-
predictionDataCleaned[which(predictionDataCleaned$full_sq
<= 5000)| is.na(predictionDataCleaned$full_sq),]
#predictionDataCleaned <-
predictionDataCleaned[which(predictionDataCleaned$kitch_sq
<= 1500| is.na(predictionDataCleaned$kitch_sq)),]
#predictionDataCleaned <-
predictionDataCleaned[which(predictionDataCleaned$life_sq
<= 1000| is.na(predictionDataCleaned$life_sq)),]
#predictionDataCleaned <-
predictionDataCleaned[which(predictionDataCleaned$state <=
4| is.na(predictionDataCleaned$state)),]
#predictionDataCleaned <-
predictionDataCleaned[which(predictionDataCleaned$kitch_sq
```

</td>
<td>

```
/*rsq .45;*/

proc glmselect data=data
        seed=1
plots(stepAxis=number)=(criterionPanel ASEPlot
CRITERIONPANEL);
model price_doc = id full_sq life_sq floor max_floor
build_year num_room
kitch_sq state raion_popul indust_part
children_preschool preschool_quota
hospital_beds_raion healthcare_centers_raion
university_top_20_raion
thermal_power_plant_raion
radiation_raion        railroad_terminal_raion
full_all young_all young_male
work_male build_count_block build_count_frame
build_count_slag metro_min_avto
kindergarten_km green_zone_km
railroad_station_walk_km
railroad_station_walk_min railroad_station_avto_km
railroad_station_avto_min
ID_railroad_station_avto water_km mkad_km
sadovoe_km kremlin_km
railroad_km railroad_1line radiation_km
thermal_power_plant_km
big_market_km hospice_morgue_km
workplaces_km shopping_centers_km office_km
church_synagogue_km
exhibition_km catering_km  church_count_3000
mosque_count_3000 green_part_5000
prom_part_5000
cafe_count_5000 cafe_avg_price_5000
big_church_count_5000 market_count_5000
green_part_2000      prom_part_2000
prom_part_3000 office_count_3000
office_sqm_3000      cafe_count_3000
/ selection=stepwise;
run;


**rsq .12;

Proc glmselect data=data
       seed=1 plots(stepAxis=number)=(criterionPanel
```

</td>
</tr>
</table>

~~~~
< 600| is.na(predictionDataCleaned$kitch_sq)),]
#predictionDataCleaned <-
predictionDataCleaned[which(predictionDataCleaned$floor >
0| is.na(predictionDataCleaned$floor)),]
#predictionDataCleaned <-
predictionDataCleaned[which(predictionDataCleaned$num_ro
om > 0 | is.na(predictionDataCleaned$num_room)),]
#predictionDataCleaned <-
predictionDataCleaned[which(predictionDataCleaned$max_flo
or > 0| is.na(predictionDataCleaned$max_floor)),]
```

```{r dataCleanupPredictionData}
predictionDataCleaned <- predictionData
#convert yes/no values to 1/0
predictionDataCleaned$culture_objects_top_25 <-
ifelse(predictionDataCleaned$culture_objects_top_25 =="yes",
1, 0)
predictionDataCleaned$full_all <-
ifelse(predictionDataCleaned$full_all =="yes", 1, 0)
predictionDataCleaned$incineration_raion <-
ifelse(predictionDataCleaned$incineration_raion =="yes", 1, 0)
predictionDataCleaned$oil_chemistry_raion <-
ifelse(predictionDataCleaned$oil_chemistry_raion =="yes", 1,
0)
predictionDataCleaned$radiation_raion <-
ifelse(predictionDataCleaned$radiation_raion =="yes", 1, 0)
predictionDataCleaned$railroad_terminal_raion <-
ifelse(predictionDataCleaned$railroad_terminal_raion =="yes",
1, 0)
predictionDataCleaned$big_market_raion <-
ifelse(predictionDataCleaned$big_market_raion =="yes", 1, 0)
predictionDataCleaned$nuclear_reactor_raion <-
ifelse(predictionDataCleaned$nuclear_reactor_raion =="yes",
1, 0)
predictionDataCleaned$detention_facility_raion <-
ifelse(predictionDataCleaned$detention_facility_raion =="yes",
1, 0)
predictionDataCleaned$thermal_power_plant_raion <-
ifelse(predictionDataCleaned$thermal_power_plant_raion
=="yes", 1, 0)
predictionDataCleaned$water_1line <-
ifelse(predictionDataCleaned$water_1line =="yes", 1, 0)
predictionDataCleaned$big_road1_1line <-
ifelse(predictionDataCleaned$big_road1_1line =="yes", 1, 0)
predictionDataCleaned$railroad_1line <-
ifelse(predictionDataCleaned$railroad_1line =="yes", 1, 0)


#convert product_type NAs to Investment
predictionDataCleaned[is.na(predictionDataCleaned[,which(na
mes(predictionDataCleaned) == "product_type")]),
which(names(predictionDataCleaned) == "product_type")] <-
"Investment"

#convert sub_area NAs to Ajeroport
predictionDataCleaned[is.na(predictionDataCleaned[,which(na
mes(predictionDataCleaned) == "sub_area")]),
which(names(predictionDataCleaned) == "sub_area")] <-
"Ajeroport"

#exclude not numeric columns for NA cleanup.
columnsToExclude <- names(predictionDataCleaned) %in%
~~~~

~~~~
ASEPlot CRITERIONPANEL);
model price_doc = id full_sq life_sq floor max_floor
build_year num_room
kitch_sq state raion_popul indust_part
children_preschool preschool_quota
hospital_beds_raion healthcare_centers_raion
university_top_20_raion
thermal_power_plant_raion
radiation_raion          railroad_terminal_raion
full_all young_all young_male
work_male build_count_block build_count_frame
build_count_slag metro_min_avto
kindergarten_km green_zone_km
railroad_station_walk_km
railroad_station_walk_min railroad_station_avto_km
railroad_station_avto_min
ID_railroad_station_avto water_km mkad_km
sadovoe_km kremlin_km
railroad_km railroad_1line radiation_km
thermal_power_plant_km
big_market_km hospice_morgue_km
workplaces_km shopping_centers_km office_km
church_synagogue_km
exhibition_km catering_km  church_count_3000
mosque_count_3000 green_part_5000
prom_part_5000
cafe_count_5000 cafe_avg_price_5000
big_church_count_5000 market_count_5000
green_part_2000       prom_part_2000
prom_part_3000 office_count_3000
office_sqm_3000       cafe_count_3000
/ selection=LASSO( choose=CV stop=CV) CVdetails ;
output out=predDataLasso p=predlasso;
run;



/*rsq 0.455;*/
proc glmselect data=data
seed=1 plots(stepAxis=number)=(criterionPanel
ASEPlot CRITERIONPANEL);
model price_doc = id full_sq life_sq floor max_floor
build_year num_room
kitch_sq state raion_popul indust_part
children_preschool preschool_quota
~~~~

<table>
<tr>
<td>

```
c("timestamp", "product_type", "sub_area", "ecology")
predictionDataSubsetCleaned <-
predictionDataCleaned[!columnsToExclude]

#apply column mean to NA values
for(i in 1:ncol(predictionDataSubsetCleaned)){

predictionDataSubsetCleaned[is.na(predictionDataSubsetCleaned[,i]), i] <- mean(predictionDataSubsetCleaned[,i], na.rm =
TRUE)
}

#add non numeric columns back in.
predictionDataSubsetCleaned$timestamp <-
predictionDataCleaned$timestamp
predictionDataSubsetCleaned$product_type <-
predictionDataCleaned$product_type
predictionDataSubsetCleaned$sub_area <-
predictionDataCleaned$sub_area
predictionDataSubsetCleaned$ecology <-
predictionDataCleaned$ecology

write.csv(predictionDataSubsetCleaned,
"predictionDataSubsetCleaned.csv")

```

```{r dataAnalysis1}
pairs(~price_doc+full_sq+life_sq+floor+max_floor+material+build_year+num_room+kitch_sq+state,data=subsetCleaned,
  main="Simple Scatterplot Matrix", diag.panel=panel.hist)
```

```{r logTransform}
subsetCleanedLogged <- subsetCleaned
subsetCleanedLogged$log_price_doc =
log(subsetCleanedLogged$price_doc)
subsetCleanedLogged$log_full_sq =
log(subsetCleanedLogged$full_sq)
subsetCleanedLogged$log_life_sq =
log(subsetCleanedLogged$life_sq)
subsetCleanedLogged$log_floor =
log(subsetCleanedLogged$floor)
subsetCleanedLogged$log_max_floor =
log(subsetCleanedLogged$max_floor)
subsetCleanedLogged$log_material =
log(subsetCleanedLogged$material)
subsetCleanedLogged$log_num_room =
log(subsetCleanedLogged$num_room)
subsetCleanedLogged$log_kitch_sq =
log(subsetCleanedLogged$kitch_sq)
subsetCleanedLogged$log_state =
log(subsetCleanedLogged$state)
```



```{r dataAnalysisLogTransformed}
pairs(~log_price_doc+log_full_sq+log_life_sq+log_floor+log_max_floor+log_material+log_num_room+log_kitch_sq+log_state,
data=subsetCleanedLogged,
  main="Simple Scatterplot Matrix", diag.panel=panel.hist)
```
```

</td>
<td>

```
hospital_beds_raion healthcare_centers_raion
university_top_20_raion
thermal_power_plant_raion
radiation_raion          railroad_terminal_raion
full_all young_all young_male
work_male build_count_block build_count_frame
build_count_slag metro_min_avto
kindergarten_km green_zone_km
railroad_station_walk_km
railroad_station_walk_min railroad_station_avto_km
railroad_station_avto_min
ID_railroad_station_avto water_km mkad_km
sadovoe_km kremlin_km
railroad_km railroad_1line radiation_km
thermal_power_plant_km
big_market_km hospice_morgue_km
workplaces_km shopping_centers_km office_km
church_synagogue_km
exhibition_km catering_km  church_count_3000
mosque_count_3000 green_part_5000
prom_part_5000
cafe_count_5000 cafe_avg_price_5000
big_church_count_5000 market_count_5000
green_part_2000       prom_part_2000
prom_part_3000 office_count_3000
office_sqm_3000       cafe_count_3000
/ selection=backward(choose=CV stop=CV)
cvmethod=split(10) CVdetails;
run;

/*External Cross Validation*/
proc glmselect data=data
seed=1 plots(stepAxis=number)=(criterionPanel
ASEPlot CRITERIONPANEL);
model price_doc = id full_sq life_sq floor max_floor
build_year num_room
kitch_sq state raion_popul indust_part
children_preschool preschool_quota
hospital_beds_raion healthcare_centers_raion
university_top_20_raion
thermal_power_plant_raion
radiation_raion          railroad_terminal_raion
full_all young_all young_male
work_male build_count_block build_count_frame
build_count_slag metro_min_avto
```

</td>
</tr>
</table>
```

```
```{r LASSOAnalyiss}
log_price_doc_columnIndex <-
which(names(subsetCleanedLogged)=="log_price_doc")
x <-
as.matrix(subsetCleanedLogged[,-log_price_doc_columnIndex
]) # Removes class
y <- as.double(as.matrix(subsetCleanedLogged[,
log_price_doc_columnIndex])) # Only class

# Fitting the model (Lasso: Alpha = 1)
set.seed(999)
cv.lasso <- cv.glmnet(x, y, family='binomial', alpha=1,
parallel=TRUE, standardize=TRUE, type.measure='auc')

# Results
plot(cv.lasso)
plot(cv.lasso$glmnet.fit, xvar="lambda", label=TRUE)
cv.lasso$lambda.min
cv.lasso$lambda.1se
coef(cv.lasso, s=cv.lasso$lambda.min)
```

```{r randomForest}
predictors <-
(subsetCleanedLogged[,-log_price_doc_columnIndex]) #
Removes class
response <- (as.matrix(subsetCleanedLogged[,
log_price_doc_columnIndex])) # Only class
fit <- randomForest(x = predictors, y=response,
                    data=subsetCleanedLogged,
                    importance=TRUE,
                    ntree=2000)
```

```{r least angle regression}

library(lars)

lasso <-
lars(x=as.matrix(subsetCleaned$full_sq,subsetCleaned$life_sq
,subsetCleaned$floor,subsetCleaned$max_floor,subsetCleane
d$material,subsetCleaned$build_year,subsetCleaned$num_ro
om,subsetCleaned$kitch_sq,subsetCleaned$state,subsetClea
ned$product_type,subsetCleaned$sub_area,subsetCleaned$a
rea_m,subsetCleaned$raion_popul,subsetCleaned$green_zon
e_part,subsetCleaned$indust_part,subsetCleaned$children_pr
eschool), y=subsetCleanedLogged$log_price_doc, type = "lar",
trace = FALSE, normalize = TRUE)

plot(lasso)


```



```{r randomforest sample}


data(iris)
set.seed(111)
```

```
kindergarten_km green_zone_km
railroad_station_walk_km
railroad_station_walk_min railroad_station_avto_km
railroad_station_avto_min
ID_railroad_station_avto water_km mkad_km
sadovoe_km kremlin_km
railroad_km railroad_1line radiation_km
thermal_power_plant_km
big_market_km hospice_morgue_km
workplaces_km shopping_centers_km office_km
church_synagogue_km
exhibition_km catering_km  church_count_3000
mosque_count_3000 green_part_5000
prom_part_5000
cafe_count_5000 cafe_avg_price_5000
big_church_count_5000 market_count_5000
green_part_2000       prom_part_2000
prom_part_3000 office_count_3000
office_sqm_3000       cafe_count_3000
/ selection=backward(choose=CVEX
stop=CROSSVALIDATE) cvmethod=split(10)
CVdetails;
run;

/*Generate residual plot*/

proc glm data=data pltos=all
PLOTS(MAXPOINTS=40000);
model price_doc = id full_sq life_sq floor max_floor
build_year num_room
kitch_sq state indust_part children_preschool
preschool_quota
university_top_20_raion
radiation_raion          build_count_block
build_count_slag kindergarten_km green_zone_km
mkad_km sadovoe_km kremlin_km railroad_km
railroad_1line thermal_power_plant_km
big_market_km office_km mosque_count_3000
green_part_5000
cafe_count_5000 cafe_avg_price_5000
prom_part_2000 office_count_3000
cafe_count_3000;
run;
```
```

ind <- sample(2, nrow(iris), replace = TRUE, prob=c(0.8, 0.2))
iris.rf <- randomForest(Species ~ ., data=iris[ind == 1,])
iris.pred <- predict(iris.rf, iris[ind == 2,])
table(observed = iris[ind==2, "Species"], predicted = iris.pred)
## Get prediction for all trees. 	
predict(iris.rf, iris[ind == 2,], predict.all=TRUE)
## Proximities. 	
predict(iris.rf, iris[ind == 2,], proximity=TRUE)
## Nodes matrix. 	
str(attr(predict(iris.rf, iris[ind == 2,], nodes=TRUE), "nodes"))
	/*Generate goal 1 output file*/
	data outputData;
	set data predictionData;
	run;
	
	proc glm data = outputData plots = all;
	model price_doc = id full_sq life_sq floor max_floor build_year num_room
	kitch_sq state indust_part children_preschool preschool_quota
	university_top_20_raion
	radiation_raion build_count_block
	build_count_slag kindergarten_km green_zone_km
	mkad_km sadovoe_km kremlin_km railroad_km
	railroad_1line thermal_power_plant_km
	big_market_km office_km mosque_count_3000
	green_part_5000
	cafe_count_5000 cafe_avg_price_5000
	prom_part_2000 office_count_3000
	cafe_count_3000;
	output out = results p = Predict;
	run;
	
	/*predict results;*/
	
	data resultsOutputGoal1;
	set results;
	if Predict > 0 then price_doc = Predict;
	if Predict < 0 then price_doc = 7123035;
	keep id price_doc;
	where id > 30473;
	run;
	
	/**Create subset of timeseries;*/
	data data2;
	set data;
	keep id timestamp price_doc;
	run;
	
	/**convert timestamp to mon, day, year;*/
	
	DATA new;
	set data2;
	year = scan(timestamp,1);
	month = scan(timestamp,2);

	day = scan(timestamp,3);
	monthYear = cats(year,month);
	RUN;
	
	/**convert month from char to number;*/
	data new2;
	set new;
	Num_month = input(month, best5.);
	run;
	
	/**sort;*/
	proc sort data=new2;
	by monthYear;
	run;
	
	/**create average price by month;*/
	data new3; set new2;
	proc means; by monthYear;
	var price_doc;
	output out=price(drop=_type_ _freq_)
	mean=AveragePrice;
	run;
	
	data price2;
	set price;
	NumMonthYear = input(monthYear, best6.);
	NumMonth = _n_;
	run;
	
	/**sort;*/
	proc sort data=price2;
	by NumMonthYear;
	run;
	
	/**plot data;*/
	proc sgscatter data=price2;
	plot AveragePrice*NumMonthYear;
	run;
	
	/*** proc autoreg with priceData below ***/;
	proc autoreg data=price2;
	model AveragePrice = NumMonthYear / nlag =(1)
	dwprob;
	run;

	data yearForecast;
	input numMonth NumMonthYear;
	datalines;
	48 201507
	49 201508
	50 201509
	51 201510
	52 201511
	53 201512
	54 201601
	55 201602
	56 201603
	57 201604
	58 201605
	59 201606
	60 201607
	;
	
	/*Generate goal 2 output file*/
	data outputDataGoal2;
	set price2 yearForecast;
	run;
	
	/*predict results;*/
	proc autoreg data=outputDataGoal2;
	model AveragePrice = NumMonthYear / nlag =(1) dwprob;
	output out = resultsOutputGoal2 p = Predict lcl= lower ucl= upper pm=trend;
	run;
	
	/* generate plot for goal 2 */
	proc glm data=resultsOutputGoal2;
	model AveragePrice = NumMonthYear / cli;
	run;

Appendix 3
Our stepwise model included the following parameters: (See full set in appendix)

Parameter	DF	Estimate	Standard Error	t Value

Intercept	1	-51301777	3469574	-14.79
id	1	67.611632	2.390739	28.28
full_sq	1	18573	597.436301	31.09
life_sq	1	61254	1348.118453	45.44
floor	1	76204	4402.338165	17.31
max_floor	1	27191	4869.835929	5.58
build_year	1	23751	1741.61694	13.64
num_room	1	1412175	34249	41.23
kitch_sq	1	55552	5553.148888	10
state	1	369156	38040	9.7
hospital_beds_raion	1	113.724875	30.844534	3.69
university_top_20_ra	1	797175	74332	10.72
radiation_raion	1	-251732	55274	-4.55
railroad_terminal_ra	1	-896324	185351	-4.84
build_count_block	1	-6589.977571	618.151683	-10.66
build_count_slag	1	15735	2020.088787	7.79
metro_min_avto	1	-61612	8405.11147	-7.33
kindergarten_km	1	148412	19391	7.65

Our backward model included the following parameters:

Parameter	DF	Estimate	Standard Error	t Value
Intercept	1	-52680104	3486352	-15.1

				1
id	1	66.368133	2.39994	27.65
full_sq	1	18523	596.51512 4	31.05
life_sq	1	61206	1347.6513 7	45.42
floor	1	77230	4402.4416 3	17.54
max_floor	1	27135	4877.9598 3	5.56
build_year	1	24460	1747.4021 3	14
num_room	1	1406710	34219	41.11
kitch_sq	1	55175	5566.1495 1	9.91
state	1	381055	38253	9.96
raion_popul	1	7.21824	1.813474	3.98
indust_part	1	-1125896	260705	-4.32
children_preschool	1	-540.515161	86.410708	-6.26
preschool_quota	1	-119.287658	29.086421	-4.1
hospital_beds_raion	1	162.001128	31.916417	5.08
healthcare_centers_r	1	68765	19894	3.46
university_top_20_ra	1	753583	76065	9.91

radiation_raion	1	-403477	61629	-6.55
railroad_terminal_ra	1	-800794	189516	-4.23
young_all	1	375.96868	67.916357	5.54
young_male	1	-302.287405	129.905174	-2.33
build_count_block	1	-6904.52427	677.314363	-10.19
build_count_frame	1	2676.713097	2071.75142	1.29
build_count_slag	1	25803	2787.64425	9.26
metro_min_avto	1	-67978	9316.56772	-7.3
kindergarten_km	1	156781	23975	6.54
green_zone_km	1	-920090	78843	-11.67
railroad_station_avt	1	-31829	15913	-2
ID_railroad_station_	1	4438.967044	765.556399	5.8
mkad_km	1	165531	13650	12.13
sadovoe_km	1	-2081735	95743	-21.74
kremlin_km	1	1943225	96680	20.1
railroad_km	1	225376	29624	7.61
railroad_1line	1	-1067854	133039	-8.03

thermal_power_plant_	1	-105889	14243	-7.43
big_market_km	1	48629	4711.76245	10.32
hospice_morgue_km	1	-41147	18432	-2.23
workplaces_km	1	-45767	14653	-3.12
office_km	1	-142734	20767	-6.87
church_synagogue_km	1	59074	40346	1.46
exhibition_km	1	65625	14358	4.57
catering_km	1	-233166	39697	-5.87
church_count_3000	1	7261.80166	5472.87868	1.33
mosque_count_3000	1	363171	70116	5.18
green_part_5000	1	-24694	3416.01713	-7.23
prom_part_5000	1	-42768	9457.06056	-4.52
cafe_count_5000	1	4311.34533	282.271387	15.27
cafe_avg_price_5000	1	834.830273	172.443007	4.84
market_count_5000	1	-47015	9394.10601	-5
prom_part_2000	1	-41474	4652.92176	-8.91

prom_part_3000	1	31788	7850.69775	4.05
office_count_3000	1	-70944	3387.40502	-20.94
office_sqm_3000	1	-0.139927	0.089244	-1.57
cafe_count_3000	1	13043	605.927388	21.5

Our LASSO model contained the following parameters:

Parameter	DF	Estimate
Intercept	1	5222251
life_sq	1	22348
num_room	1	625806
sadovoe_km	1	-3485.263

iiii. A model of your choice. This may be using another OLS or LASSO model or custom model, etc.

Our custom model contained the following parameters:

Source	DF	Type I SS	Mean Square	F Value	Pr > F
id	1	1.08E+16	1.08E+16	857.98	<.0001
full_sq	1	7.48E+16	7.48E+16	5937.68	<.0001
life_sq	1	5.63E+16	5.63E+16	4473.77	<.0001
floor	1	3.12E+15	3.12E+15	247.89	<.0001
max_floor	1	4.76E+14	4.76E+14	37.83	<.0001
build_year	1	5.14E+15	5.14E+15	407.82	<.0001

num_room	1	3.27E+16	3.27E+16	2600.29	<.0001
kitch_sq	1	1.12E+16	1.12E+16	885.82	<.0001
state	1	3.87E+15	3.87E+15	307	<.0001
indust_part	1	1.77E+15	1.77E+15	140.47	<.0001
children_preschool	1	7.66E+15	7.66E+15	608.1	<.0001
preschool_quota	1	3.41E+16	3.41E+16	2707.99	<.0001
university_top_20_ra	1	3.20E+15	3.20E+15	253.93	<.0001
radiation_raion	1	3.27E+14	3.27E+14	25.96	<.0001
build_count_block	1	4.18E+14	4.18E+14	33.2	<.0001
build_count_slag	1	8.90E+14	8.90E+14	70.65	<.0001
kindergarten_km	1	1.92E+15	1.92E+15	152.6	<.0001
green_zone_km	1	5.59E+14	5.59E+14	44.36	<.0001
mkad_km	1	2.25E+14	2.25E+14	17.89	<.0001
sadovoe_km	1	1.43E+16	1.43E+16	1139.54	<.0001
kremlin_km	1	1.26E+16	1.26E+16	1003.93	<.0001
railroad_km	1	5.83E+13	5.83E+13	4.63	0.0315
railroad_1line	1	3.55E+14	3.55E+14	28.22	<.0001
thermal_power_plant_	1	2.52E+14	2.52E+14	20.03	<.0001
big_market_km	1	2.24E+15	2.24E+15	177.64	<.0001

office_km	1	4.88E+14	4.88E+14	38.8	<.0001
mosque_count_3000	1	1.35E+15	1.35E+15	107.56	<.0001
green_part_5000	1	1.92E+15	1.92E+15	152.65	<.0001
cafe_count_5000	1	2.20E+15	2.20E+15	174.98	<.0001
cafe_avg_price_5000	1	2.05E+14	2.05E+14	16.3	<.0001
prom_part_2000	1	2.53E+15	2.53E+15	200.54	<.0001
office_count_3000	1	3.02E+15	3.02E+15	240.07	<.0001
cafe_count_3000	1	6.77E+15	6.77E+15	537.45	<.0001

Submissions:
What to submit 2 DS in a single zip file:

Prediction from Goal 1. (csv file)
Wrangled data set from Goal 2. (48 rows including title row. / csv file)
Predictions from Goal 2. (csv file)
Line plot of predictions from Goal 2 with 95% confidence intervals. Image or cut and pasted into something like a word doc.
Final paper (No longer than 11 pages without appendix.) LaTex/Word/ etc.

Note: Data Wrangling:

Wrangling = having a long and complicated dispute.
Part of this project is meant to have a significant data wrangling component. As an example, you will more than likely need to work with R or SAS or both to change data from character/string to integer/numeric so your models make the predictions that are required. This is only an example of the data wrangling you will need to conduct. It will help to start early and bring these issues up in live session and/or office hours.
Due Date:
All submissions are due no later than 11:59pm Saturday October 10th.