Contents

[Goal 1 2](#_Toc21283199)

[Introduction 2](#_Toc21283200)

[Description of the data with a table or a reference to a table in an Appendix. 2](#_Toc21283201)

[Data Cleaning / Wrangling (any renaming of variables or standardizing of values.) 3](#_Toc21283202)

[Individual Variable Outlier Identification and Handling 3](#_Toc21283203)

[Checking Assumptions & Model Outlier Identification and Handling 4](#_Toc21283204)

[Variable Selection 5](#_Toc21283205)

[Potential Interactions 6](#_Toc21283206)

[Modeling 6](#_Toc21283207)

[Prediction 7](#_Toc21283208)

[Goal 2 7](#_Toc21283209)

[Introduction 7](#_Toc21283210)

[Data Wrangling 7](#_Toc21283211)

[Serial Correlation Verification 9](#_Toc21283212)

[Time Series Predictions 9](#_Toc21283213)

[Appendix 11](#_Toc21283214)

[Table of missing values 11](#_Toc21283215)

[Variable transformations samples: histograms 11](#_Toc21283216)

[Proportional Imputation Example 12](#_Toc21283217)

[Correlation Matrix Example 12](#_Toc21283218)

[Variance Inflation Factor 12](#_Toc21283219)

[Predictor Variables Interaction Plots 13](#_Toc21283220)

[Data Dictionary 19](#_Toc21283221)

[Model Cross-Validations 21](#_Toc21283222)

[LASSO Regression Using LASSO-Selected Variables – Model 1 21](#_Toc21283223)

[OLS Backward Elimination Using LASSO Variables and OLS – Model 2 21](#_Toc21283224)

[OLS Backward Elimination Using OLS Backward Elimination Variables – Model 3 22](#_Toc21283225)

[Custom Model Using OLS Backward Elimination Selection, Correlation Matrix, & VIF – Custom 22](#_Toc21283226)

[Serial Correlation Verification 23](#_Toc21283227)

[EDA R Code: Imputations & Transformations 24](#_Toc21283228)

[Test Data Cleaning R Code: Preparation for External Cross-Validation 29](#_Toc21283229)

[Goal 1 SAS Code: Modeling & Cross-Validation 33](#_Toc21283230)

[Goal 2 SAS Code: Time-Series Modeling & Predictions 47](#_Toc21283231)

# Goal 1

## Introduction

Linear regression has many applications within the realm of predictive analytics, including the prediction of housing prices. At the request of Sberbank – Russia’s oldest and largest bank – our team was tasked with performing this type of analysis using data provided by the bank. Our team’s first goal in the introductory project for the Applied Statistics: Inference & Modeling course in the Southern Methodist University Master of Science in Data Science (MSDS) program was to build a model employing linear regression techniques we've learned in this course - and the course prior - to date. Our team elected to use SAS as the preferred analysis platform for its analytical power and statistical reliability, as it has been proven stable for decades within the industry. We elected to use R for data wrangling and performing intermediary analysis as R performs quickly and scales easily when transformations are required across large dimensions of data.

The object of our first goal was to apply four distinct predictive models to assess the suitability of our parameter selections for determining the realty prices across a region of Russia relative in distance to the Kremlin – roughly 70 kilometers, or 43 miles. The measures of accuracy were applied in terms of Akaike Information Criterion, Standard Block Metric, k-fold internal Cross Validation measured in terms of the CV PRESS statistic, and external cross validation between competing models. Our approach outlined in this first objective is limited in that we were not permitted to use more advanced algorithms we will be exposed to later in the MSDS program; rather, in conjunction with the aforementioned linear regression techniques.

## Description of the data with a table or a reference to a table in an Appendix.

The Sberbank real estate data set describes the sale of both investment and owner occupier properties from 2011-2015 in the Moscow, Russia metropolitan region. The data provided was comprised of 70 quantitative variables and three qualitative variables. Altogether, our data was split conservatively, with 75% for used for training the models and 25% used for testing. Altogether, these splits were applied to 19,835 observations. While data originally included 25,471 observations, 5,636 were excluded in the analysis and transformation process after being identified as either outliers or possibly the result of data entry or survey sampling error. Within the first objective of the project, we determined sale prices (`price\_doc`) as the response of 69 selected linear (“main effects”) and interactive variables, featuring proportional imputation of missing values and logarithmic and square root transformations of the numeric variables, as well as one-hot encoding for the categorical variables. For reference, an [example of the proportional transformation](#_Proportional_Imputation_Example) is explicated in the appendix.

A [complete list of the variables](#_Data_Dictionary) provided is in the data dictionary section of the appendix. From this list, we excluded from our model the variables timestamp, full\_sq, material, build\_year, raion\_popul, children\_preschool, hospital\_beds\_raion, office\_raion, metro\_min\_avto, metro\_min\_walk, metro\_km\_walk, railroad\_station\_walk\_km, railroad\_station\_walk\_min, railroad\_station\_avto\_min, public\_transport\_station\_min\_walk, and basketball\_km.

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

The variables with missing observations for which we provided imputations were floor of building (`floor`), number of floors in any given building (`max\_floor`), total living area, excluding balconies and other non-residential areas (`life\_sq`), kitchen area (`kitch\_sq`), number of living rooms (`num\_room`), number of seats in pre-school organizations (`preschool\_quota`), walking time to the nearest railroad station (`railroad\_station\_walk\_min`) and the ID of that railroad station (`ID\_railroad\_station\_walk`), walking distance to the nearest railroad (`railroad\_station\_walk\_km`), and shares of local buildings constructed within particular time periods - `build\_count\_before\_1920`, `build\_count\_1921-1945`, `build\_count\_1946-1970`, `build\_count\_1971-1995`, and `build\_count\_after\_1995` - (`metro\_min\_walk`), and (`metro\_km\_walk`). A [table of our missing values and counts](#_Table_of_missing) can be seen in the appendix.

Next, we removed variables for wall material (`material`), number of hospital beds within the district (`hospital\_beds\_raion`, and the year build of buildings (`build\_year`) as these contained too much missing information to provide imputation – we considered imputing these variables, but determined the risk of imputing outweighed the benefits to modeling.

We furthermore removed observations for imputed values of `life\_sq` and `kitch\_sq` where these exceeded the total area (again, designated by `full\_sq`). These imputations – originally calculated as a mean response of their total non-null values proportional to corresponding total square meters – exceeded total square meters and were thus determined to be impractical imputations.

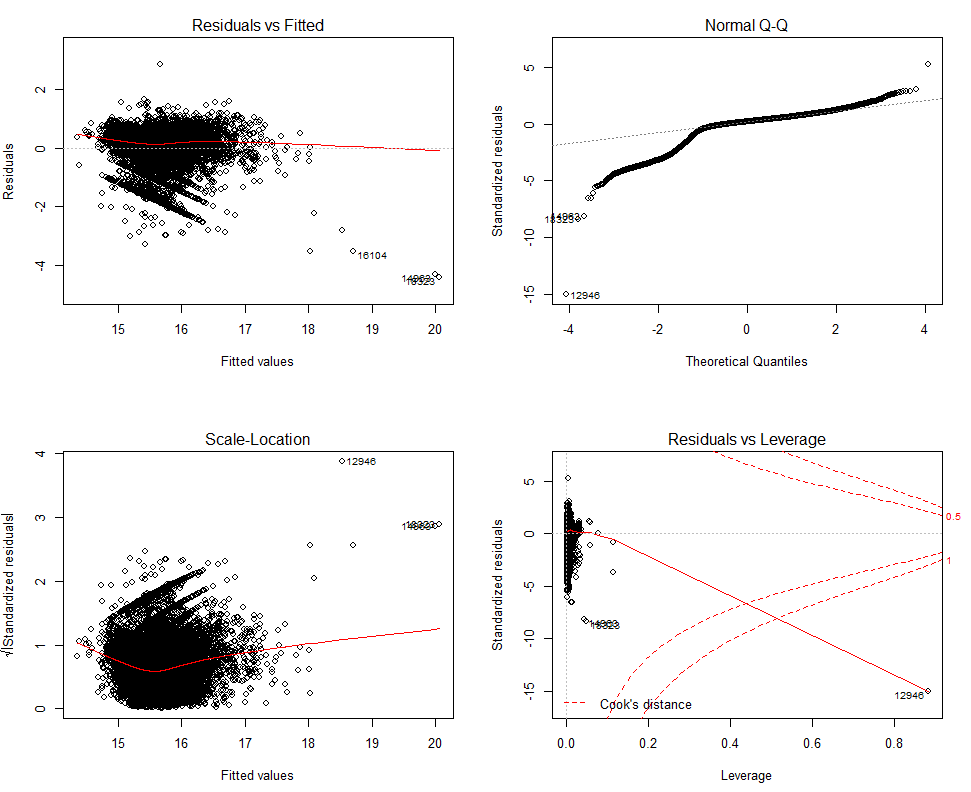
Finally, we provided logarithmic (of varying bases) transformations to `floor` and `price\_doc` and square root transformations of values raised to various exponentials within the predictor features `kitch\_sq`, `preschool\_quota`, and `office\_raion`. We considered further transformations, but determined this would increase the risk of over-fitting once our chosen model was applied to new data, as done for the predictions provided using our file projectionData.csv. This panned to likely be a reasonable decision as evidenced by the very close adjusted R-squared value in our projected data (44%, roughly) compared to our modeling data (46%, roughly). Our approach was to make as much use of the available data as possible, but not to the extent that we would have over-fit modeling in our project. While this is not a guarantee in any statistical project, we feel confident in our efforts herein to mitigate this risk. [Sample histograms](#_Variable_transformations_samples:) for our variables’ transformations can be seen in the appendix.

## Individual Variable Outlier Identification and Handling

Because of the high volume of imputations and transformations to non-imputed variables, we determined, with the assistance of analyzing histograms of the individual predictors, that the best approach would be to run the models without first removing individual observations. Instead, we removed for key variables – based on their significance in predictive power as indicated by the correlation matrix – where the values were extreme. For example, where the kitchen was equal to less than two square meters of the total square meters or where living space was greater than the full square meters available, we removed these observations (2314 observations for `life\_sq` and 68 observations for `kitch\_sq`).

## Checking Assumptions & Model Outlier Identification and Handling

Following the development of our models (more on this later), we analyzed diagnostics plots to determine what further outlying observations should be removed. We performed this analysis in R due to the support R provided; we encountered obstacles where the large dimensions of our data and model prevented us from visualizing these diagnostics in SAS. Although our missing data was handled en masse earlier on in our initial individual predictor variable exploratory data analysis (EDA), we determined the most appropriate method would be to assess the diagnostics following development of our final model in SAS and remove additional outliers following this analysis, as needed. We then reassessed our model after removing outliers to determine if one of our other models was better suited for the task assigned by Sberbank in objective one. The residuals diagnostics plots are output as follows:

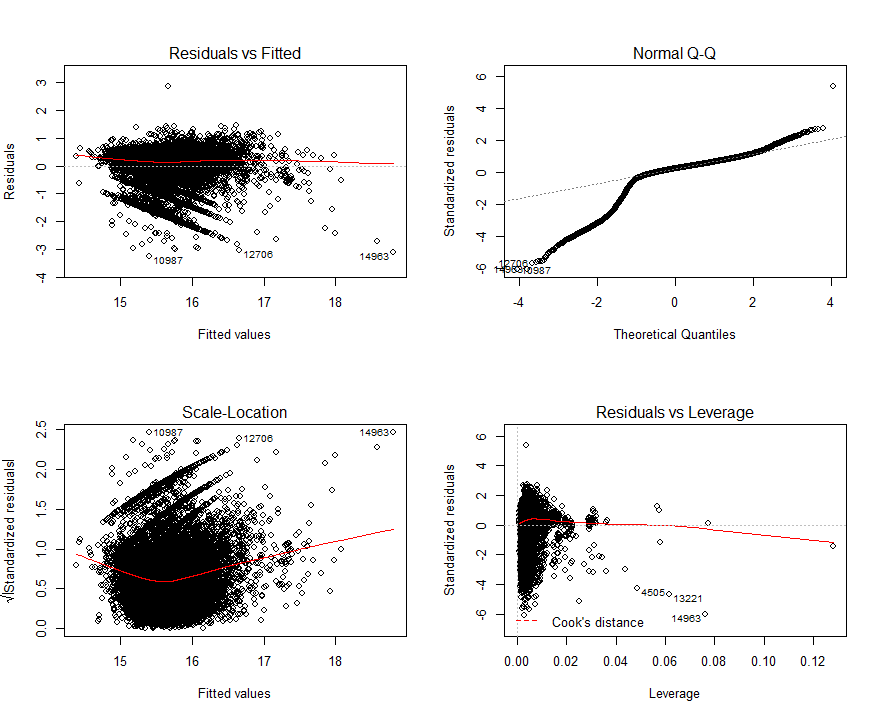


Linear regression diagnostics residuals plots for custom OLS model, prior to outlier removal

As indicated above, it appeared possible that some of our imputations – while necessary to capture a sufficient volume of data relevant to our entire analysis and predictions – resulted in a minor violation of the required non-constant variance assumptions for the regression model, as indicated in our **Residuals** (and Studentized Residuals, which has a higher consideration of standard error) **vs. Fitted** plot above. This could also be the result of sampling error or other phenomena that occurred concurrently with the timeframe relevant to our project’s scope. Even so, there remained a healthy enough scatter among the residuals to assume constant variance in the model.

In analyzing the **QQ plot**, it is notably left-skewed. However, based on the relative adherence to normality of the remaining residuals in the distribution, we feel confident that the central limit theorem – especially with more data gathered over time – would resolve this kurtosis. With that in consideration, it is important to note that this data was gathered over time and we therefore cannot safely assume a high likelihood of independence among the observations.

Finally, in analyzing the **Residuals vs. Leverage**, we identified several outlying observations that required removal from the dataset. Upon removing these, our leverage measurement – especially with respect to Cook’s distance, improved significantly (below plots), which resulted in a 0.3% overall improvement on our final adjusted R-squared metric. While the leverage was improved, the removal of these observations had very little impact on the skewedness and variance within our model.



Linear regression diagnostics residuals plots for custom OLS model, following outlier removal

## Variable Selection

Variable selection was initially managed by the Ordinary Least Squares (OLS) and LASSO regression techniques. Using all variables, excluding wall materials, hospital bed counts by district, and build year variables, which, as previously mentioned, were dropped for excessively high volumes of missing observations that were too risky for imputation. After processing all variables through these methods, Backward Elimination provided the highest adjusted R-squared and lowest AIC and SBC metrics. Consequently, we chose the variables suggested by this method. However, we provided further variable elimination based on direct analysis of the correlation matrix and Variable Inflation Factor (VIF) table’s thresholds. We then considered interactions between linear terms based on our reasonable domain knowledge and reapplied the updated variable set through our OLS and LASSO selection processes again, removing those that were not significant. Following this, we removed interactions based on outputs from the interaction plots (dropping interaction terms where lines were parallel and their 95% confidence intervals did not overlap) and applied OLS and LASSO one more time. A complete list of interaction plots can be viewed in the [appendix](#_Appendix), with a sample of significant interaction immediately below:

|  |  |
| --- | --- |
| **Interaction Plots** | **Interaction Parameter Estimates and Statistics** |
|  |  |

## Potential Interactions

Given the variables present in the data set there may be combinations of variables that would make sense to paired together. Interactions such as school and distance to a school may be of interest to buyers with children; the build material and the year the home was built may be informative as different time periods may have had preferred building materials or availability. By including these types of interactions, a model may have better fit statistics suggesting that it can better describe and predict home pricing values.

## Modeling

In expansion of the modeling methodology, this section will provide statistical output and discussion of results from our modeling. We tested four models, opting to make predictions based on our custom model. The models we developed were OLS Forward Selection, OLS Backward Elimination, OLS Stepwise Regression, and LASSO Regression. Ultimately, the aforementioned combination of OLS Backward Elimination, VIF analysis, and a correlation matrix with visual inspection of interaction plots proved the best. [SAS code](#_Ordinary_Least_Squares) for these models can be found in the appendix.

Overall, the custom model outperformed all other models in terms of AIC, SBC, Internal k-fold Cross Validation, and external cross validation. Because these results changed slightly as testing was repeated across the folds, we considered this and a combination of other metrics, including Mean Square Errors produced from the models as well as the adjusted R-Square values across both internal and external cross validation as this data source diversity would increase chances of representing natural phenomena encountered under practical scenarios. Output tables are in the appendix for this section.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **LASSO Estimates using LASSO Variables**  [Appendix table](#_LASSO_Regression_Using) | **OLS (Backward) Estimates using LASSO Variables**  [Appendix table](#_OLS_Backward_Elimination) | **OLS (Backward) Estimates using OLS Variables**  [Appendix table](#_OLS_Backward_Elimination_1) | **OLS (Backward) Estimates using Custom OLS Variables**  [Appendix table](#_Custom_Model_Using) |
| **AIC (Internal)** | 621818 | 621818 | 620432 | 620750 |
| **AIC (External)** | 470167 | 470167 | 468947 | 469383 |
| **SBC (Internal)** | 602100 | 602100 | 601094 | 601467 |
| **SBC (External)** | 455277 | 455277 | 454424 | 454912 |
| **Internal Cross Validation (5 folds)** | 3.0667x1017 | 3.0698X1017 | 5.1845x1017 | 3.089X1017 |
| **External Cross Validation (5 folds)** | 2.3004x1017 | 2.2927x1017 | 5.2081x1017 | 2.3074x1017 |

## Prediction

After considering information provided by internal and external cross validation, it was found that the custom model had the highest performance. Compared to the other models, the Custom model had the lowest RMSE, indicating that the amount of variation in predicted and observed values are smaller. Although the AIC and SBC values – which suggest better goodness of fit while not being over-fit – were between 0.05% and 0.1% better for the Backward OLS model, the backward OLS model produced the worst of the four models in terms of CV PRESS, suggesting the amount of error from the test and training sets was the worst for this model. However, since the Custom model performed consistently well, and better than all four on the CV PRESS metrics – here, indicating the amount of error from test and training sets was the least, we used the Custom model for predictions. Cross Validation and Model performance statistics can be found in the [appendix](#_Model_Cross-Validations), along with the Extra Sums of Squares outputs.

Data from the test set was cleaned in a similar manner to the training set and appended to the training set. After cleaning the test data there were 5,093 observations. Many of these observations were removed due to absence of values and to maintain consistency within the structures of the training and test sets. The Custom model was then re-run with the training/test set in SAS to allow for predictions with 95% confidence intervals. Please see the attached “SampleSubmission.csv” for the Custom model predictions.

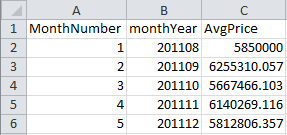
# Goal 2

## Introduction

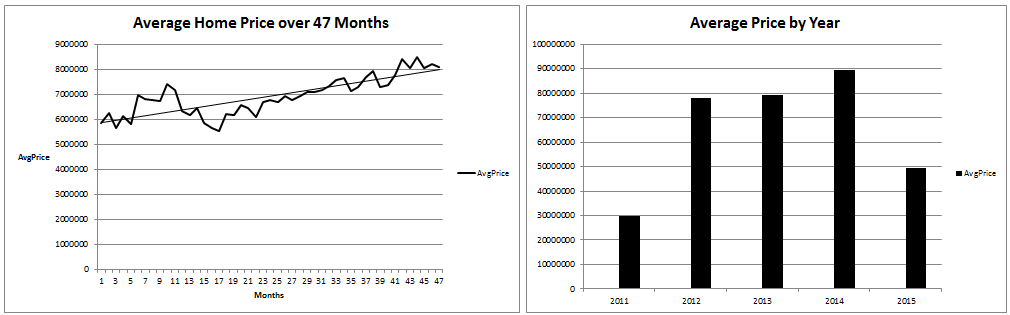
Many factors play a role in housing price and demand. Sberbank, a large bank in Russia provide historical housing data from August 2011 to June 2015 so we could predict the average property price for July 2015 through July 2016. The housing market appears to be stable during this time – although the country’s economy was not, which can make forecasting difficult.

## Data Wrangling

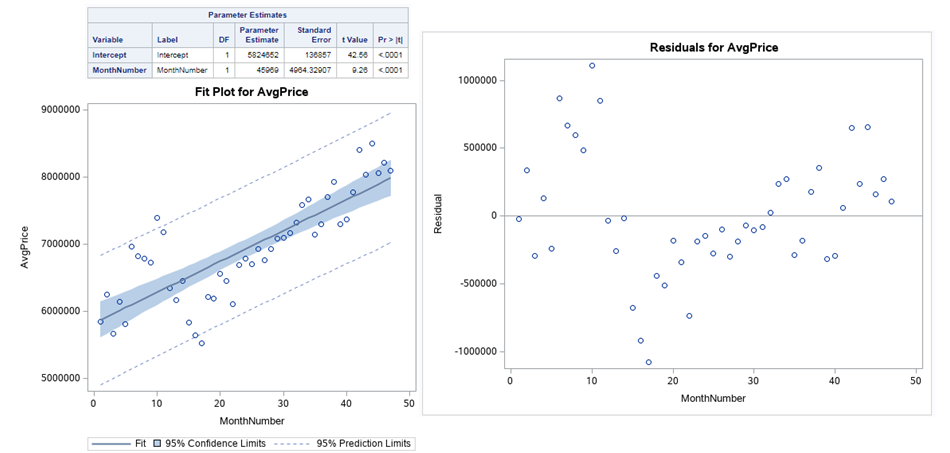
Using the data provided we extracted the timestamp and price values from the data provided by the bank. The timestamp required formatting into a proper “date/time” standard. Our goal was to recognize the average price by month-year and the variation over time. The date field was separated into three additional columns; month, day year. Then we merged the year and month columns together to capture the average price by month-year then this was numbered by in ascending order, example below.



Looking at the data graphically we can see an upward trend of the average home price in the 47 months of data collected, however looking at the average price by year the trend is upward but has a sharp decline in 2015. To explain the trends we analyzed the 71 variables the bank provided and we need to consider Russia’s volatile economy.



Looking at a simple linear regression model: . Can we assume the average home price will trend above or below the regression line for an extended amount of time, suggesting in serially correlated residuals? Upon visual inspection of the residual plot, there evidence the series will run above and below the trend line.



## Serial Correlation Verification

This model has 1 variable, with 47 observations; given this information and the use of a Durban-Watson (DW) Table the lower and upper bounds used for identifying serial correlation are approximately: DWLB ≈ 1.50 and DWUB ≈ 1.59. For a set of data to exhibit serial correlation, a model needs that have a DW statistic that is less than the DW lower bound. If a model’s DW statistic is between the upper and lower bound than the model is inconclusive for serial correlations; if a model’s DW statistic is larger than the DW upper bound than that model does not have, or no longer has, serial correlation.

Upon initial inspection of an AR(0) model, the DW statistic of 0.7524 suggests that there is serial correlation between time observations. The ACF and PACF plots suggest that a lag of 1 is a strong candidate for serial correlation correction. When studying the PACF plot in more detail, the plot suggests that a lag of 5 may also be a candidate for serial correlation correction.

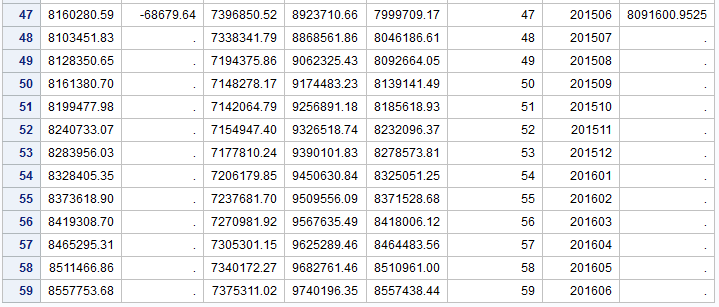
To screen for the correct number of lags to apply for time series, models for AR(1) – AR(5). The main requirement for selecting the best lag for time series requires that the DW statistic be greater than 1.59, and that ACF and PACF plots show that there is no longer serial correlation, AIC, and SBC statistics provide further evidence confirming the performance of a model. It is important to select the correct lag for prediction as it will adjust the standard error appropriately given the correlation between time values. If the incorrect lag is selected prediction estimates may not be as accurate and may also have larger error terms and confidence interval widths.

After running time series models for AR(1) – AR(5), the AR(1) model had a DW statistic of 2.1502 (Pr<DW) = 0.6422; AIC = 1340.68; and SBC = 1346.24. Although the PACF plots in the AR(0) suggested that there was a potential for a lag 5 model, the DW-statistics for AR(2) – AR(5) models all showed DW statistics that were less than the DW lower limit of 1.50. This confirmed that these models all still have serial correlation and are probably not the best models for describing this data. The AR(1) model’s AIC and SBC statistics were also lower than the AR(2) – AR(5) models. This suggests that the AR(1) model is a better fit for the data. Please see the [appendix](#_Serial_Correlation_Verification) for DW, AIC, SBC, ACF, and PACF statistics.

## Time Series Predictions

Predictions were calculated using the AR(1) time series model based on criteria describe in the “Serial Correlation Verification” section. The predictions include an upper and lower bound for the 95% confidence intervals. There are no residuals as residuals cannot be calculated unless there are response values (AvgPrice) present.





To see the full set of data including the training and test data, please see the attached “tsPrediction.csv” file included in the zip file.

# Appendix

## Table of missing values

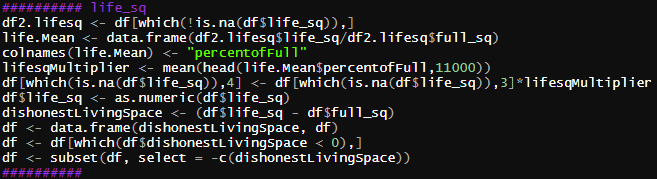
|  |  |  |  |
| --- | --- | --- | --- |
| Feature Name | Count Missing | Feature Name | Count Missing |
| hospital\_beds\_raion | 12,096 | **build\_count\_before\_1920** | 4,195 |
| build\_year | 11,392 | **build\_count\_1921.1945** | 4,195 |
| max\_floor | 7,991 | **build\_count\_1946.1970** | 4,195 |
| material | 7,991 | **build\_count\_1971.1995** | 4,195 |
| num\_room | 7,991 | **build\_count\_after\_1995** | 4,195 |
| kitch\_sq | 7,991 | **floor** | 146 |
| preschool\_quota | 5,604 | **metro\_min\_walk** | 19 |
| life\_sq | 5,333 | **metro\_km\_walk** | 19 |
| build\_count\_block | 4,195 | **railroad\_station\_walk\_km** | 19 |
| build\_count\_wood | 4,195 | **railroad\_station\_walk\_min** | 19 |
| build\_count\_frame | 4,195 | **ID\_railroad\_station\_walk** | 19 |
| build\_count\_brick | 4,195 |  |  |

Reference Table for Missing Values in modelingData.csv

## Variable transformations samples: histograms

|  |  |
| --- | --- |
| **Kitchen Square Meters Log Transformation** | **Office Raion Square Root Transformation** |
|  |  |
|  |  |

## Proportional Imputation Example



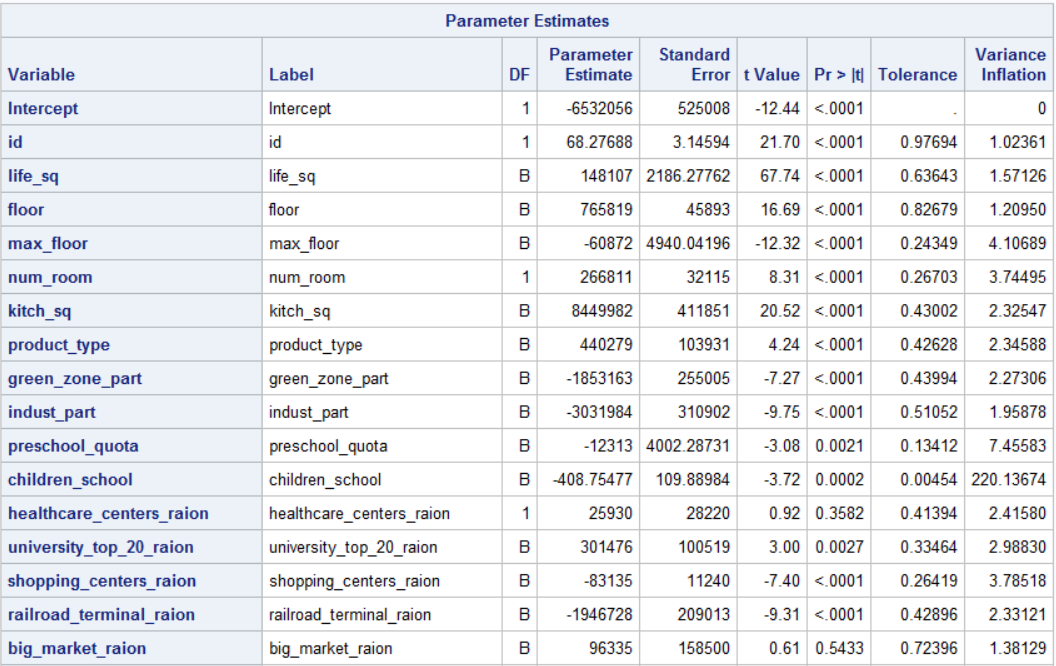
## Correlation Matrix Example

Below is a portion of our correlation matrix used in the custom model. This table is too large to replicate in this document in its entirety. However, the code used to develop this is contained in section [Goal 1 SAS Code: Modeling & Cross-Validation](#_Goal_1_SAS).



## Variance Inflation Factor

Below is a portion of our variance inflation factor scores and table output. As with the correlation matrix, this table is too large to replicate in this document in its entirety. However, the code used to develop this is contained in section [Goal 1 SAS Code: Modeling & Cross-Validation](#_Goal_1_SAS). For our custom model, we applied a threshold cutoff for most variables of a Tolerance value less than 0.1. Terms used in significant interactions that breached this threshold were still used in the model as this provides a degree of healthy error in a model, which in turn prevents over-fitting to the modeling data.



## Predictor Variables Interaction Plots

|  |  |
| --- | --- |
| **Interaction Plots between Predictor Variables** | **Estimates, Errors, Student-t Statistics, p-Values for Predictors** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Data Dictionary

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Attribute Description** | **Data Type - Post Transformation** |
| id | transaction id | integer |
| timestamp | date of transaction | time series |
| full\_sq | total area in square meters, including loggias, balconies and other non-residential areas | integer |
| life\_sq | living area in square meters, excluding loggias, balconies and other non-residential areas | numeric |
| floor | for apartments, floor of the building | numeric |
| max\_floor | number of floors in the building | numeric |
| material | wall material | numeric |
| build\_year | year building was constructed | numeric |
| num\_room | number of living rooms | numeric |
| kitch\_sq | kitchen area | numeric |
| product\_type | owner-occupier purchase or investment | categorical factor with two levels; one-hot encoded |
| raion\_popul | Number of municipality population. district | integer |
| green\_zone\_part | Proportion of area of ​​greenery in the total area | numeric |
| indust\_part | Share of industrial zones in area of ​​the total area | numeric |
| children\_preschool | Number of pre-school age population | integer |
| preschool\_quota | Number of seats in pre-school organizations | numeric |
| children\_school | Population of school-age children | integer |
| hospital\_beds\_raion | Number of hospital beds for the district | integer |
| healthcare\_centers\_raion | Number of healthcare centers in district | integer |
| university\_top\_20\_raion | Number of higher education institutions in the top ten ranking of the Federal rank | integer |
| shopping\_centers\_raion | Number of malls and shopping centers in district | integer |
| office\_raion | Number of malls and shopping centers in district | numeric |
| railroad\_terminal\_raion | Presence of the railroad terminal in district | categorical factor with two levels; one-hot encoded |
| big\_market\_raion | Presence of large grocery / wholesale markets | categorical factor with two levels; one-hot encoded |
| full\_all | Total number of population in the municipality | integer |
| 0\_6\_all | Population count of age 0-6 years | integer |
| 7\_14\_all | Population count of age 7-14 years | integer |
| 0\_17\_all | Population count of age 0-17 years | integer |
| 16\_29\_all | Population count of age 16-19 years | integer |
| 0\_13\_all | Population count of age 0-13 years | integer |
| build\_count\_block | Share of block buildings | integer |
| build\_count\_wood | Share of wood buildings | integer |
| build\_count\_frame | Share of frame buildings | integer |
| build\_count\_brick | Share of brick buildings | integer |
| build\_count\_before\_1920 | Share of buildings constructed before year 1920 | integer |
| build\_count\_1921-1945 | Share of buildings constructed between years 1921 and 1945 | integer |
| build\_count\_1946-1970 | Share of buildings constructed between 1946 and 1970 | integer |
| build\_count\_1971-1995 | Share of buildings constructed between 1971 and 1995 | integer |
| build\_count\_after\_1995 | Share of buildings constructed after 1995 | integer |
| metro\_min\_avto | Time to subway by car, min. | numeric |
| metro\_km\_avto | Distance to subway by car, km | numeric |
| metro\_min\_walk | Time to metro by foot | numeric |
| metro\_km\_walk | Distance to the metro, km | numeric |
| school\_km | Distance to high school | numeric |
| park\_km | Distance to park | numeric |
| green\_zone\_km | Distance to green zone | numeric |
| industrial\_km | Distance to industrial zone | numeric |
| railroad\_station\_walk\_km | Distance to the railroad station (walking) | numeric |
| railroad\_station\_walk\_min | Time to the railroad station (walking) | numeric |
| ID\_railroad\_station\_walk | Nearest railroad station id (walk) | integer |
| railroad\_station\_avto\_km | Distance to the railroad station | numeric |
| railroad\_station\_avto\_min | Time to the railroad station | numeric |
| public\_transport\_station\_km | Distance to the public transport station (walk) | numeric |
| public\_transport\_station\_min\_walk | Time to the public transport station (walk) | numeric |
| kremlin\_km | Distance to the city center (Kremlin) | numeric |
| big\_road1\_km | Distance to Nearest major road | numeric |
| big\_road2\_km | The distance to next distant major road | numeric |
| railroad\_km | Distance to the railway / Moscow Central Ring / open areas Underground | numeric |
| bus\_terminal\_avto\_km | Distance to bus terminal | numeric |
| big\_market\_km | Distance to grocery / wholesale markets | numeric |
| market\_shop\_km | Distance to markets and department stores | numeric |
| fitness\_km | Distance to fitness | numeric |
| swim\_pool\_km | Distance to swimming pool | numeric |
| ice\_rink\_km | Distance to ice palace | numeric |
| stadium\_km | Distance to stadium | numeric |
| basketball\_km | Distance to the basketball courts | numeric |
| public\_healthcare\_km | Distance to public healthcare | numeric |
| university\_km | Distance to universities | numeric |
| workplaces\_km | Distance to workplaces | numeric |
| shopping\_centers\_km | Distance to shopping centers | numeric |
| office\_km | Distance to business centers/ offices | numeric |
| big\_church\_km | Distance to large church | numeric |
| price\_doc | sale price (this is the target variable) | numeric |

## Model Cross-Validations

### LASSO Regression Using LASSO-Selected Variables – Model 1

|  |  |
| --- | --- |
| External Cross Validation Using Test/Train Split | Internal Cross Validation Using Full Data |
|  |  |
|  |  |

### OLS Backward Elimination Using LASSO Variables and OLS – Model 2

|  |  |
| --- | --- |
| External Cross Validation Using Test/Train Split | Internal Cross Validation Using Full Data |
|  |  |
|  |  |

### OLS Backward Elimination Using OLS Backward Elimination Variables – Model 3

|  |  |
| --- | --- |
| External Cross Validation Using Test/Train Split | Internal Cross Validation Using Full Data |
|  |  |
|  |  |

### Custom Model Using OLS Backward Elimination Selection, Correlation Matrix, & VIF – Custom

|  |  |
| --- | --- |
| External Cross Validation Using Test/Train Split | Internal Cross Validation Using Full Data |
|  |  |
|  |  |

## Serial Correlation Verification

|  |  |
| --- | --- |
| AR(0) Time Series Model |  |
|  |  |
|  | |
| AR(1) Time Series Model | |
|  |  |

|  |  |
| --- | --- |
| AR(3) Time Series Model | AR(4) Time Series Model |
|  |  |

|  |  |
| --- | --- |
| AR(4) Time Series Model | AR(5) Time Series Model |
|  |  |

## EDA R Code: Imputations & Transformations

library(pacman)

p\_load(lmtest

,dplyr

,Hmisc

,skimr

,tidyr

,na.tools

,tidyverse

,olsrr

,caret

,multcomp

,ggthemes

,MASS# for OLS

,regclass# for VIF

,stats

,glmnet

,sjPlot

,sjmisc

,ggplot2

,xlsx)

#format dates:

df <- read.csv("./modelingData.csv", header=T, sep=",", strip.white=T, stringsAsFactors = F)

names(df)[36] <- "build\_count\_1921\_1945"

names(df)[37] <- "build\_count\_1946\_1970"

names(df)[38] <- "build\_count\_1971\_1995"

names(df)[54] <- "public\_trans\_station\_time\_walk"

########## Floor

#df$floor <- df$floor %>% replace\_na(0)

df$floor[is.na(df$floor)] <- 0

df$floor <- log(df$floor+1)

##########

##########

#df[which(df$year == 2011),]

########## max floor

df2.maxfl <- df[which(!is.na(df$max\_floor)),]

maxfl.Mean <- data.frame(df2.maxfl$max\_floor/df2.maxfl$full\_sq)

colnames(maxfl.Mean) <- "percentofFloor"

maxflMultiplier <- mean(head(maxfl.Mean$percentofFloor,7000))

df[which(is.na(df$max\_floor)),6] <- df[which(is.na(df$max\_floor)),3]\*maxflMultiplier

##########

#df[which(df$year == 2011),]

########## life sq

df2.lifesq <- df[which(!is.na(df$life\_sq)),]

life.Mean <- data.frame(df2.lifesq$life\_sq/df2.lifesq$full\_sq)

colnames(life.Mean) <- "percentofFull"

lifesqMultiplier <- mean(head(life.Mean$percentofFull,11000))

df[which(is.na(df$life\_sq)),4] <- df[which(is.na(df$life\_sq)),3]\*lifesqMultiplier

df$life\_sq <- as.numeric(df$life\_sq)

dishonestLivingSpace <- (df$life\_sq - df$full\_sq)

df <- data.frame(dishonestLivingSpace, df)

df <- df[which(df$dishonestLivingSpace < 0),] #living space shouldn't be greater than the full sq, considering lofts that have shared external bathrooms

df <- subset(df, select = -c(dishonestLivingSpace)) # remove the counter variable dishonestLivingSpace

##########

#df[which(df$year == 2011),]

########## kithcen sq

df2.kitchsq <- df[which(!is.na(df$kitch\_sq)),]

kitch.Mean <- data.frame(df2.kitchsq$kitch\_sq/df2.kitchsq$full\_sq)

colnames(kitch.Mean) <- "percentofFullkitch"

kitchsqMultiplier <- mean(head(kitch.Mean$percentofFullkitch,7000))

df[which(is.na(df$kitch\_sq)),10] <- df[which(is.na(df$kitch\_sq)),3]\*kitchsqMultiplier

df$kitch\_sq[is.na(df$kitch\_sq)] <- 0

dishonestKitchens <- (df$kitch\_sq - df$full\_sq)

df <- data.frame(dishonestKitchens, df)

df <- df[which(df$dishonestKitchens < -2),] #kitchen space shouldn't be greater than more than 2 meters less than the full sq

df <- subset(df, select = -c(dishonestKitchens)) # remove the counter variable dishonestKitchens

df$kitch\_sq <- sqrt(df$kitch\_sq^1/16+1)

##########

#df[which(df$year == 2011),]

########## num room

df$num\_room <- as.integer(df$num\_room)

df2.numRm <- df[which(!is.na(df$num\_room)),]

numRm.Mean <- as.numeric(df2.numRm$num\_room)/as.numeric(df2.numRm$full\_sq)

#colnames(numRm.Mean) <- "percentofFullrm"

class(numRm.Mean)

#numRm.Mean$percentofFullrm <- as.numeric(percentofFullrm)

numRmMultiplier <- mean(head(numRm.Mean,7000))

df[which(is.na(df$num\_room)),9] <- as.integer(df[which(is.na(df$num\_room)),3])\*numRmMultiplier

##################################################################################################df <- df[which(df$num\_room < 30),]

##########

#df[which(df$year == 2011),]

########## preschool quota

df.preK.no.NA <- df[which(!is.na(df$preschool\_quota)),]

preK.Mean <- data.frame(df.preK.no.NA$preschool\_quota/df.preK.no.NA$X0\_6\_all)

colnames(preK.Mean) <- "percentofFloor"

preKquotaMultiplier <- mean(head(preK.Mean$percentofFloor,5000))

df[which(is.na(df$preschool\_quota)),16] <- df[which(is.na(df$preschool\_quota)),26]\*preKquotaMultiplier

df$preschool\_quota <- sqrt(df$preschool\_quota)

df$preschool\_quota[is.na(df$preschool\_quota)] <- 1

##########

#df[which(df$year == 2011),]

########## build year

##########

#df[which(df$year == 2011),]

########## office raion

df$office\_raion <- sqrt(df$office\_raion^1/10)

##########

#df[which(df$year == 2011),]

########## big market raion

df$big\_market\_raion <- dplyr::recode(df$big\_market\_raion, "no" = 0, "yes"= 1)

##########

########## Product\_type

df$product\_type <- dplyr::recode(df$product\_type, "Investment" = 0, "OwnerOccupier"= 1)

##########

#df[which(df$year == 2011),]

########## railroad terminal raion

df$railroad\_terminal\_raion <- dplyr::recode(df$railroad\_terminal\_raion, "no" = 0, "yes"= 1)

##########

#df[which(df$year == 2011),]

##########

#df <- df %>% mutate(railroad\_station\_walk\_min = if\_else(is.na(railroad\_station\_walk\_min),0,railroad\_station\_walk\_min))

df$railroad\_station\_walk\_min[is.na(df$railroad\_station\_walk\_min)] <- 0

##########

#df[which(df$year == 2011),]

##########

#df <- df %>% mutate(ID\_railroad\_station\_walk = if\_else(is.na(ID\_railroad\_station\_walk),0,ID\_railroad\_station\_walk))

df$ID\_railroad\_station\_walk[is.na(df$ID\_railroad\_station\_walk)] <- 0

##########

#df[which(df$year == 2011),]

##########

#df$railroad\_station\_walk\_km <- df$railroad\_station\_walk\_km %>% replace\_na(0)

df$railroad\_station\_walk\_km[is.na(df$railroad\_station\_walk\_km)] <- 0

##########

#df[which(df$year == 2011),]

##########

df <- df[which(!is.na(df$build\_count\_before\_1920)),] #one fell swoop to take out all NA 'build\_count\_{year range}' rows

##########

#df[which(df$year == 2011),]

##########

#df$metro\_min\_walk <- df$metro\_min\_walk %>% replace\_na(0)

df$metro\_min\_walk[is.na(df$metro\_min\_walk)] <- 0

##########

#df[which(df$year == 2011),]

##########

#df$metro\_km\_walk <- df$metro\_km\_walk %>% replace\_na(0)

df$metro\_km\_walk[is.na(df$metro\_km\_walk)] <- 0

##########

#df[which(df$year == 2011),]

################year\_month <- as.factor(paste0(df$year,df$month))

################df <- data.frame(year\_month,df)

############################## conversion to numeric and factor only for modeling consistency #############################

########## Material, Hospital bed raion

df <- subset(df, select = -c(material, hospital\_beds\_raion, build\_year))

df <- df %>% mutate\_if(is.integer, as.numeric) %>% mutate\_if(is.character, as.factor) %>% data.frame()

#df <- df[-c(2958, 1192, 2998,134, 3156, 1452, 2994, 3151, 98),]

write.csv(df,"cleanData.csv", row.names = F)

df <- read.csv("cleanData.csv", header = T)

######################################################################################################## ignore the below

skim(df)

df2 <- df

df2$life\_sq <- log(df$full\_sq + 1)

df2$basketball\_km <- log(df$basketball\_km + 1)

df2$fitness\_km <- log(df$fitness\_km + 1)

df2$green\_zone\_km <- log(df$green\_zone\_km + 1)

df2$indust\_part <- log(df$indust\_part + 1)

df2$kitch\_sq <- log(df$kitch\_sq + 1)

df2$life\_Sq <- log(df$life\_Sq + 1)

df2$market\_shop\_km <- log(df$market\_shop\_km + 1)

df2$max\_floor <- log(df$max\_floor + 1)

df2$metro\_km\_avto <- log(df$metro\_km\_avto + 1)

df2$metro\_km\_walk <- log(df$metro\_km\_walk + 1)

df2$metro\_min\_avto <- log(df$metro\_min\_avto + 1)

df2$metro\_min\_walk <- log(df$metro\_min\_walk + 1)

df2$num\_room <- log(df$num\_room + 1)

df2$office\_km <- log(df$office\_km + 1)

df2$park\_km <- log(df$park\_km + 1)

df2$price\_doc <- log(df$price\_doc + 1)

df2$public\_healthcare\_km <- log(df$public\_healthcare\_km + 1)

df2$public\_trans\_station\_time\_walk <- log(df$public\_trans\_station\_time\_walk + 1)

df2$public\_transport\_station\_km <- log(df$public\_transport\_station\_km + 1)

df2$railroad\_km <- log(df$railroad\_km + 1)

df2$school\_km <- log(df$school\_km + 1)

df2$shopping\_centers\_km <- log(df$shopping\_centers\_km + 1)

skim(df2)

######################################################################################################## ignore the above

lm.Model <- lm(log(price\_doc) ~ id + life\_sq + floor + max\_floor + num\_room + kitch\_sq + product\_type + green\_zone\_part + indust\_part + preschool\_quota + children\_school + healthcare\_centers\_raion +

university\_top\_20\_raion + shopping\_centers\_raion + railroad\_terminal\_raion + big\_market\_raion + X0\_17\_all + X16\_29\_all + build\_count\_block + build\_count\_wood + build\_count\_frame +

build\_count\_brick + build\_count\_before\_1920 + build\_count\_1921\_1945 + build\_count\_1946\_1970 + build\_count\_1971\_1995 + build\_count\_after\_1995 + metro\_km\_avto + school\_km +

green\_zone\_km + industrial\_km + ID\_railroad\_station\_walk + railroad\_station\_avto\_km + public\_transport\_station\_km + public\_trans\_station\_time\_walk + kremlin\_km + big\_road1\_km +

big\_road2\_km + railroad\_km + bus\_terminal\_avto\_km + big\_market\_km + market\_shop\_km + fitness\_km + swim\_pool\_km + ice\_rink\_km + stadium\_km + public\_healthcare\_km + university\_km +

workplaces\_km + shopping\_centers\_km + office\_km + big\_church\_km + X0\_17\_all\*X16\_29\_all + children\_school\*school\_km + build\_count\_block\*build\_count\_1921\_1945 +

build\_count\_block\*build\_count\_1946\_1970 + build\_count\_block\*build\_count\_1971\_1995 + build\_count\_block\*build\_count\_after\_1995 + build\_count\_wood\*build\_count\_before\_1920 +

build\_count\_wood\*build\_count\_1946\_1970 + build\_count\_wood\*build\_count\_after\_1995 + build\_count\_frame\*build\_count\_before\_1920 + build\_count\_frame\*build\_count\_1921\_1945 +

build\_count\_frame\*build\_count\_1946\_1970 + build\_count\_frame\*build\_count\_after\_1995 + build\_count\_brick\*build\_count\_1946\_1970 + build\_count\_brick\*build\_count\_1971\_1995 +

build\_count\_brick\*build\_count\_after\_1995 + office\_km\*X16\_29\_all, data=df)

# viewing diagnostics residuals:

par(mfrow=c(2,2))

plot(lm.Model)

# Based on the suggested model above from SAS, we dropped outlier observations:

df <- df[-c(12946, 16104,8678, 18323, 8924, 6425, 9728),]

# After updating to remove the outliers, we re-ran the model:

lm.Model <- lm(log(price\_doc) ~ id + life\_sq + floor + max\_floor + num\_room + kitch\_sq + product\_type + green\_zone\_part + indust\_part + preschool\_quota + children\_school + healthcare\_centers\_raion +

university\_top\_20\_raion + shopping\_centers\_raion + railroad\_terminal\_raion + big\_market\_raion + X0\_17\_all + X16\_29\_all + build\_count\_block + build\_count\_wood + build\_count\_frame +

build\_count\_brick + build\_count\_before\_1920 + build\_count\_1921\_1945 + build\_count\_1946\_1970 + build\_count\_1971\_1995 + build\_count\_after\_1995 + metro\_km\_avto + school\_km +

green\_zone\_km + industrial\_km + ID\_railroad\_station\_walk + railroad\_station\_avto\_km + public\_transport\_station\_km + public\_trans\_station\_time\_walk + kremlin\_km + big\_road1\_km +

big\_road2\_km + railroad\_km + bus\_terminal\_avto\_km + big\_market\_km + market\_shop\_km + fitness\_km + swim\_pool\_km + ice\_rink\_km + stadium\_km + public\_healthcare\_km + university\_km +

workplaces\_km + shopping\_centers\_km + office\_km + big\_church\_km + X0\_17\_all\*X16\_29\_all + children\_school\*school\_km + build\_count\_block\*build\_count\_1921\_1945 +

build\_count\_block\*build\_count\_1946\_1970 + build\_count\_block\*build\_count\_1971\_1995 + build\_count\_block\*build\_count\_after\_1995 + build\_count\_wood\*build\_count\_before\_1920 +

build\_count\_wood\*build\_count\_1946\_1970 + build\_count\_wood\*build\_count\_after\_1995 + build\_count\_frame\*build\_count\_before\_1920 + build\_count\_frame\*build\_count\_1921\_1945 +

build\_count\_frame\*build\_count\_1946\_1970 + build\_count\_frame\*build\_count\_after\_1995 + build\_count\_brick\*build\_count\_1946\_1970 + build\_count\_brick\*build\_count\_1971\_1995 +

build\_count\_brick\*build\_count\_after\_1995 + office\_km\*X16\_29\_all, data=df)

# viewing post-outlier removal diagnostics residuals:

par(mfrow=c(2,2))

plot(lm.Model)

# As indicated by the residuals diagnostics, our model is better, with the exception of issues with independence and non-constant

# variance from some of the imputations. This is also causing a left-skew in the QQ plot.

#updating with removed variables

write.csv(df,"cleanData.csv", row.names = F)

## Test Data Cleaning R Code: Preparation for External Cross-Validation

library(pacman)

p\_load(lmtest

,dplyr

,Hmisc

,skimr

,tidyr

,na.tools

,tidyverse

,olsrr

,caret

,multcomp

,ggthemes

,MASS# for OLS

,regclass# for VIF

,stats

,glmnet

,sjPlot

,sjmisc

,ggplot2

,xlsx)

#format dates:

df <- Test

names(df)[35] <- "build\_count\_1921\_1945"

names(df)[36] <- "build\_count\_1946\_1970"

names(df)[37] <- "build\_count\_1971\_1995"

names(df)[53] <- "public\_trans\_station\_time\_walk"

##############df <- df %>% mutate(timestamp = as.Date(timestamp, origin="1899-12-30"))

##############tryFormats = c("%Y-%m-%d", "%Y/%m/%d")

##############timestamp2 <- df$timestamp

##############df <- data.frame(timestamp2, df)

##############df <- df %>% mutate(timestamp = as.Date(timestamp, origin="1899-12-30"))

##############tryFormats = c("%Y-%m-%d", "%Y/%m/%d")

##############df <- df %>% separate(timestamp2, sep="-", into = c("year", "month", "day"))

#names(df$build\_count\_1921.1945)[36] <- "build\_count\_1921\_1945"

#names(df$build\_count\_1946.1970)[37] <- "build\_count\_1946\_1970"

#names(df$build\_count\_1971.1995)[38] <- "build\_count\_1971\_1995"

#names(df$public\_transport\_station\_min\_walk)[54] <- "public\_trans\_station\_time\_walk"

#$colnames(df) <- c('year', 'month', 'day', 'id', 'timestamp', 'full\_sq', 'life\_sq', 'floor', 'max\_floor', 'num\_room', 'kitch\_sq', 'product\_type', 'raion\_popul', 'green\_zone\_part', 'indust\_part', 'children\_preschool', 'preschool\_quota', 'children\_school', 'healthcare\_centers\_raion', 'university\_top\_20\_raion', 'shopping\_centers\_raion', 'office\_raion', 'railroad\_terminal\_raion', 'big\_market\_raion', 'full\_all', 'X0\_6\_all', 'X7\_14\_all', 'X0\_17\_all', 'X16\_29\_all', 'X0\_13\_all', 'build\_count\_block', 'build\_count\_wood', 'build\_count\_frame', 'build\_count\_brick', 'build\_count\_before\_1920', 'build\_count\_1921\_1945', 'build\_count\_1946\_1970', 'build\_count\_1971\_1995', 'build\_count\_after\_1995', 'metro\_min\_avto', 'metro\_km\_avto', 'metro\_min\_walk', 'metro\_km\_walk', 'school\_km', 'park\_km', 'green\_zone\_km', 'industrial\_km', 'railroad\_station\_walk\_km', 'railroad\_station\_walk\_min', 'ID\_railroad\_station\_walk', 'railroad\_station\_avto\_km', 'railroad\_station\_avto\_min', 'public\_transport\_station\_km', 'public\_trans\_station\_time\_walk', 'kremlin\_km', 'big\_road1\_km', 'big\_road2\_km', 'railroad\_km', 'bus\_terminal\_avto\_km', 'big\_market\_km', 'market\_shop\_km', 'fitness\_km', 'swim\_pool\_km', 'ice\_rink\_km', 'stadium\_km', 'basketball\_km', 'public\_healthcare\_km', 'university\_km', 'workplaces\_km', 'shopping\_centers\_km', 'office\_km', 'big\_church\_km', 'price\_doc')

########## Floor

#df$floor <- df$floor %>% replace\_na(0)

df$floor[is.na(df$floor)] <- 0

df$floor <- log(df$floor+1)

##########

#df$product\_type <- as.factor(df$product\_type)

##########

#df[which(df$year == 2011),]

########## max floor

df2.maxfl <- df[which(!is.na(df$max\_floor)),]

maxfl.Mean <- data.frame(df2.maxfl$max\_floor/df2.maxfl$full\_sq)

colnames(maxfl.Mean) <- "percentofFloor"

maxflMultiplier <- mean(head(maxfl.Mean$percentofFloor,7000))

df[which(is.na(df$max\_floor)),7] <- df[which(is.na(df$max\_floor)),4]\*maxflMultiplier

##########

#df[which(df$year == 2011),]

########## life sq

df2.lifesq <- df[which(!is.na(df$life\_sq)),]

life.Mean <- data.frame(df2.lifesq$life\_sq/df2.lifesq$full\_sq)

colnames(life.Mean) <- "percentofFull"

lifesqMultiplier <- mean(head(life.Mean$percentofFull,11000))

df[which(is.na(df$life\_sq)),5] <- df[which(is.na(df$life\_sq)),4]\*lifesqMultiplier

df$life\_sq <- as.numeric(df$life\_sq)

dishonestLivingSpace <- (df$life\_sq - df$full\_sq)

df <- data.frame(dishonestLivingSpace, df)

df <- df[which(df$dishonestLivingSpace < 0),] #living space shouldn't be greater than the full sq, considering lofts that have shared external bathrooms

df <- subset(df, select = -c(dishonestLivingSpace)) # remove the counter variable dishonestLivingSpace

##########

#df[which(df$year == 2011),]

########## kithcen sq

df2.kitchsq <- df[which(!is.na(df$kitch\_sq)),]

kitch.Mean <- data.frame(df2.kitchsq$kitch\_sq/df2.kitchsq$full\_sq)

colnames(kitch.Mean) <- "percentofFullkitch"

kitchsqMultiplier <- mean(head(kitch.Mean$percentofFullkitch,7000))

df[which(is.na(df$kitch\_sq)),9] <- df[which(is.na(df$kitch\_sq)),4]\*kitchsqMultiplier

df$kitch\_sq[is.na(df$kitch\_sq)] <- 0

dishonestKitchens <- (df$kitch\_sq - df$full\_sq)

df <- data.frame(dishonestKitchens, df)

df <- df[which(df$dishonestKitchens < -2),] #kitchen space shouldn't be greater than more than 2 meters less than the full sq

df <- subset(df, select = -c(dishonestKitchens)) # remove the counter variable dishonestKitchens

df$kitch\_sq <- sqrt(df$kitch\_sq^1/16+1)

##########

#df[which(df$year == 2011),]

########## num room

df$num\_room <- as.integer(df$num\_room)

df2.numRm <- df[which(!is.na(df$num\_room)),]

numRm.Mean <- as.numeric(df2.numRm$num\_room)/as.numeric(df2.numRm$full\_sq)

#colnames(numRm.Mean) <- "percentofFullrm"

class(numRm.Mean)

#numRm.Mean$percentofFullrm <- as.numeric(percentofFullrm)

numRmMultiplier <- mean(head(numRm.Mean,7000))

df[which(is.na(df$num\_room)),8] <- as.integer(df[which(is.na(df$num\_room)),4])\*numRmMultiplier

##################################################################################################df <- df[which(df$num\_room < 30),]

##########

#df[which(df$year == 2011),]

########## preschool quota

df.preK.no.NA <- df[which(!is.na(df$preschool\_quota)),]

preK.Mean <- data.frame(df.preK.no.NA$preschool\_quota/df.preK.no.NA$X0\_6\_all)

colnames(preK.Mean) <- "percentofFloor"

preKquotaMultiplier <- mean(head(preK.Mean$percentofFloor,5000))

df[which(is.na(df$preschool\_quota)),15] <- df[which(is.na(df$preschool\_quota)),28]\*preKquotaMultiplier

df$preschool\_quota <- sqrt(df$preschool\_quota)

df$preschool\_quota[is.na(df$preschool\_quota)] <- 1

##########

#df[which(df$year == 2011),]

########## build year

##########

#df[which(df$year == 2011),]

########## office raion

df$office\_raion <- sqrt(df$office\_raion^1/10)

##########

#df[which(df$year == 2011),]

########## big market raion

df$big\_market\_raion <- dplyr::recode(df$big\_market\_raion, "no" = 0, "yes"= 1)

##########

########## Product\_type

df$product\_type <- dplyr::recode(df$product\_type, "Investment" = 0, "OwnerOccupier"= 1)

##########

#df[which(df$year == 2011),]

########## railroad terminal raion

df$railroad\_terminal\_raion <- dplyr::recode(df$railroad\_terminal\_raion, "no" = 0, "yes"= 1)

##########

#df[which(df$year == 2011),]

##########

#df <- df %>% mutate(railroad\_station\_walk\_min = if\_else(is.na(railroad\_station\_walk\_min),0,railroad\_station\_walk\_min))

df$railroad\_station\_walk\_min[is.na(df$railroad\_station\_walk\_min)] <- 0

##########

#df[which(df$year == 2011),]

##########

#df <- df %>% mutate(ID\_railroad\_station\_walk = if\_else(is.na(ID\_railroad\_station\_walk),0,ID\_railroad\_station\_walk))

df$ID\_railroad\_station\_walk[is.na(df$ID\_railroad\_station\_walk)] <- 0

##########

#df[which(df$year == 2011),]

##########

#df$railroad\_station\_walk\_km <- df$railroad\_station\_walk\_km %>% replace\_na(0)

df$railroad\_station\_walk\_km[is.na(df$railroad\_station\_walk\_km)] <- 0

##########

#df[which(df$year == 2011),]

##########

df <- df[which(!is.na(df$build\_count\_before\_1920)),] #one fell swoop to take out all NA 'build\_count\_{year range}' rows

##########

#df[which(df$year == 2011),]

##########

#df$metro\_min\_walk <- df$metro\_min\_walk %>% replace\_na(0)

df$metro\_min\_walk[is.na(df$metro\_min\_walk)] <- 0

##########

#df[which(df$year == 2011),]

##########

#df$metro\_km\_walk <- df$metro\_km\_walk %>% replace\_na(0)

df$metro\_km\_walk[is.na(df$metro\_km\_walk)] <- 0

##########

#df[which(df$year == 2011),]

################year\_month <- as.factor(paste0(df$year,df$month))

################df <- data.frame(year\_month,df)

############################## conversion to numeric and factor only for modeling consistency #############################

########## Material, Hospital bed raion

df <- subset(df, select = -c(material, hospital\_beds\_raion, build\_year))

df <- df %>% mutate\_if(is.integer, as.numeric) %>% mutate\_if(is.character, as.factor) %>% data.frame()

#df <- df[-c(2958, 1192, 2998,134, 3156, 1452, 2994, 3151, 98),]

write.csv(df,"cleanTestData.csv", row.names = F)

## Goal 1 SAS Code: Modeling & Cross-Validation

FILENAME REFFILE 'C:/Users/Pablo/Desktop/KG6372/cleanData.xlsx';

PROC IMPORT DATAFILE=REFFILE

DBMS=XLSX

OUT= DF;

GETNAMES=YES;

run;

proc print data = DF(obs=10);

run;

/\* backward elimination - this is now the best model. It reached 0.4500 with interaction between preschool\_children and preschool\_quota. 0.4618 with more interaction\*/

title "Backward Elimination";

proc glmselect data=DF

testdata=DF

seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model PRICE\_DOC = id timestamp full\_sq life\_sq floor max\_floor num\_room kitch\_sq product\_type raion\_popul green\_zone\_part indust\_part children\_preschool preschool\_quota children\_school healthcare\_centers\_raion university\_top\_20\_raion shopping\_centers\_raion office\_raion railroad\_terminal\_raion big\_market\_raion full\_all X0\_17\_all X16\_29\_all X0\_13\_all build\_count\_block build\_count\_wood build\_count\_frame build\_count\_brick build\_count\_before\_1920 build\_count\_1946\_1970 build\_count\_1971\_1995 build\_count\_after\_1995 metro\_min\_avto metro\_km\_avto metro\_min\_walk school\_km green\_zone\_km industrial\_km railroad\_station\_walk\_km ID\_railroad\_station\_walk railroad\_station\_avto\_km railroad\_station\_avto\_min public\_transport\_station\_km kremlin\_km big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km market\_shop\_km fitness\_km swim\_pool\_km ice\_rink\_km stadium\_km basketball\_km public\_healthcare\_km university\_km workplaces\_km shopping\_centers\_km office\_km big\_church\_km children\_preschool\*preschool\_quota x16\_29\_all\*x0\_6\_all x16\_29\_all\*x0\_13\_all x0\_17\_all\*x0\_13\_all x0\_17\_all\*x16\_29\_all x7\_14\_all\*x0\_17\_all x7\_14\_all\*x0\_17\_all x7\_14\_all\*x0\_17\_all x16\_29\_all\*x0\_13\_all x16\_29\_all\*x0\_17\_all x16\_29\_all\*x7\_14\_all children\_school\*school\_km build\_count\_block\*build\_count\_before\_1920 build\_count\_block\*build\_count\_1921\_1945 build\_count\_block\*build\_count\_1946\_1970 build\_count\_block\*build\_count\_1971\_1995 build\_count\_block\*build\_count\_after\_1995 build\_count\_wood\*build\_count\_before\_1920 build\_count\_wood\*build\_count\_1921\_1945 build\_count\_wood\*build\_count\_1946\_1970 build\_count\_wood\*build\_count\_1971\_1995 build\_count\_wood\*build\_count\_after\_1995 build\_count\_frame\*build\_count\_before\_1920 build\_count\_frame\*build\_count\_1921\_1945 build\_count\_frame\*build\_count\_1946\_1970 build\_count\_frame\*build\_count\_1971\_1995 build\_count\_frame\*build\_count\_after\_1995 build\_count\_brick\*build\_count\_before\_1920 build\_count\_brick\*build\_count\_1921\_1945 build\_count\_brick\*build\_count\_1946\_1970 build\_count\_brick\*build\_count\_1971\_1995 build\_count\_brick\*build\_count\_after\_1995 fitness\_km\*X16\_29\_all swim\_pool\_km\*X0\_6\_all swim\_pool\_km\*X7\_14\_all swim\_pool\_km\*X0\_17\_all swim\_pool\_km\*X0\_13\_all swim\_pool\_km\*X16\_29\_all ice\_rink\_km\*X0\_6\_all ice\_rink\_km\*X7\_14\_all ice\_rink\_km\*X0\_17\_all ice\_rink\_km\*X0\_13\_all ice\_rink\_km\*X16\_29\_all stadium\_km\*X16\_29\_all office\_km\*X16\_29\_all

/ selection=backward( choose=CV stop=CV include = 107) CVdetails;

run;

quit;

title "Forward Selection";

proc glmselect data=DF

testdata=DF

seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model PRICE\_DOC = id life\_sq floor max\_floor kitch\_sq indust\_part kremlin\_km / selection=forward( choose=CV stop=CV ) CVdetails;

run;

quit;

title "Stepwise Regression";

proc glmselect data=DF

testdata=DF

seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model PRICE\_DOC = id timestamp full\_sq life\_sq floor max\_floor num\_room kitch\_sq product\_type raion\_popul green\_zone\_part indust\_part children\_preschool preschool\_quota children\_school healthcare\_centers\_raion university\_top\_20\_raion shopping\_centers\_raion office\_raion railroad\_terminal\_raion big\_market\_raion full\_all X0\_17\_all X16\_29\_all X0\_13\_all build\_count\_block build\_count\_wood build\_count\_frame build\_count\_brick build\_count\_before\_1920 build\_count\_1946\_1970 build\_count\_1971\_1995 build\_count\_after\_1995 metro\_min\_avto metro\_km\_avto metro\_min\_walk school\_km green\_zone\_km industrial\_km railroad\_station\_walk\_km ID\_railroad\_station\_walk railroad\_station\_avto\_km railroad\_station\_avto\_min public\_transport\_station\_km kremlin\_km big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km market\_shop\_km fitness\_km swim\_pool\_km ice\_rink\_km stadium\_km basketball\_km public\_healthcare\_km university\_km workplaces\_km shopping\_centers\_km office\_km big\_church\_km children\_preschool\*preschool\_quota x16\_29\_all\*x0\_6\_all x16\_29\_all\*x0\_13\_all x0\_17\_all\*x0\_13\_all x0\_17\_all\*x16\_29\_all x7\_14\_all\*x0\_17\_all x7\_14\_all\*x0\_17\_all x7\_14\_all\*x0\_17\_all x16\_29\_all\*x0\_13\_all x16\_29\_all\*x0\_17\_all x16\_29\_all\*x7\_14\_all children\_school\*school\_km build\_count\_block\*build\_count\_before\_1920 build\_count\_block\*build\_count\_1921\_1945 build\_count\_block\*build\_count\_1946\_1970 build\_count\_block\*build\_count\_1971\_1995 build\_count\_block\*build\_count\_after\_1995 build\_count\_wood\*build\_count\_before\_1920 build\_count\_wood\*build\_count\_1921\_1945 build\_count\_wood\*build\_count\_1946\_1970 build\_count\_wood\*build\_count\_1971\_1995 build\_count\_wood\*build\_count\_after\_1995 build\_count\_frame\*build\_count\_before\_1920 build\_count\_frame\*build\_count\_1921\_1945 build\_count\_frame\*build\_count\_1946\_1970 build\_count\_frame\*build\_count\_1971\_1995 build\_count\_frame\*build\_count\_after\_1995 build\_count\_brick\*build\_count\_before\_1920 build\_count\_brick\*build\_count\_1921\_1945 build\_count\_brick\*build\_count\_1946\_1970 build\_count\_brick\*build\_count\_1971\_1995 build\_count\_brick\*build\_count\_after\_1995 fitness\_km\*X16\_29\_all swim\_pool\_km\*X0\_6\_all swim\_pool\_km\*X7\_14\_all swim\_pool\_km\*X0\_17\_all swim\_pool\_km\*X0\_13\_all swim\_pool\_km\*X16\_29\_all ice\_rink\_km\*X0\_6\_all ice\_rink\_km\*X7\_14\_all ice\_rink\_km\*X0\_17\_all ice\_rink\_km\*X0\_13\_all ice\_rink\_km\*X16\_29\_all stadium\_km\*X16\_29\_all office\_km\*X16\_29\_all

/ selection=stepwise(choose=CV stop=CV include = 30) CVdetails;

run;

quit;

/\* LASSO selection \*/

title "LASSO Selection";

/\* R-squared is not a great measure for LASSO here \*/

proc glmselect data =DF

seed=1 plots(stepAxis = number)=(criterionPanel ASEPlot CRITERIONPANEL);

model price\_doc = id timestamp full\_sq life\_sq floor max\_floor num\_room kitch\_sq product\_type raion\_popul green\_zone\_part indust\_part children\_preschool preschool\_quota children\_school healthcare\_centers\_raion university\_top\_20\_raion shopping\_centers\_raion office\_raion railroad\_terminal\_raion big\_market\_raion full\_all X0\_6\_all X7\_14\_all X0\_17\_all X16\_29\_all X0\_13\_all build\_count\_block build\_count\_wood build\_count\_frame build\_count\_brick build\_count\_before\_1920 build\_count\_1921\_1945 build\_count\_1946\_1970 build\_count\_1971\_1995 build\_count\_after\_1995 metro\_min\_avto metro\_km\_avto metro\_min\_walk metro\_km\_walk school\_km park\_km green\_zone\_km industrial\_km railroad\_station\_walk\_km railroad\_station\_walk\_min ID\_railroad\_station\_walk railroad\_station\_avto\_km railroad\_station\_avto\_min public\_transport\_station\_km public\_trans\_station\_time\_walk kremlin\_km big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km market\_shop\_km fitness\_km swim\_pool\_km ice\_rink\_km stadium\_km basketball\_km public\_healthcare\_km university\_km workplaces\_km shopping\_centers\_km office\_km big\_church\_km price\_doc

/selection = LASSO(choose = CV stop=CV) CVdetails;

run;

quit;

/\* The start of our custom model (below) uses interaction terms and all variables suggested by our best OLS linear model (from backward) \*/

proc glmselect data=DF

testdata=DF

seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model PRICE\_DOC = id life\_sq floor max\_floor num\_room kitch\_sq product\_type green\_zone\_part indust\_part preschool\_quota children\_school healthcare\_centers\_raion university\_top\_20\_raion shopping\_centers\_raion railroad\_terminal\_raion big\_market\_raion X0\_17\_all X16\_29\_all build\_count\_block build\_count\_wood build\_count\_frame build\_count\_brick build\_count\_before\_1920 build\_count\_1921\_1945 build\_count\_1946\_1970 build\_count\_1971\_1995 build\_count\_after\_1995 metro\_km\_avto school\_km green\_zone\_km industrial\_km ID\_railroad\_station\_walk railroad\_station\_avto\_km public\_transport\_station\_km public\_trans\_station\_time\_walk kremlin\_km big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km market\_shop\_km fitness\_km swim\_pool\_km ice\_rink\_km stadium\_km public\_healthcare\_km university\_km workplaces\_km shopping\_centers\_km office\_km big\_church\_km x16\_29\_all\*x0\_6\_all x16\_29\_all\*x0\_13\_all X0\_17\_all\*X0\_13\_all X0\_17\_all\*X16\_29\_all X0\_17\_all\*X7\_14\_all X16\_29\_all\*X7\_14\_all children\_school\*school\_km build\_count\_block\*build\_count\_before\_1920 build\_count\_block\*build\_count\_1921\_1945 build\_count\_block\*build\_count\_1946\_1970 build\_count\_block\*build\_count\_1971\_1995 build\_count\_block\*build\_count\_after\_1995 build\_count\_wood\*build\_count\_before\_1920 build\_count\_wood\*build\_count\_1921\_1945 build\_count\_wood\*build\_count\_1946\_1970 build\_count\_wood\*build\_count\_1971\_1995 build\_count\_wood\*build\_count\_after\_1995 build\_count\_frame\*build\_count\_before\_1920 build\_count\_frame\*build\_count\_1921\_1945 build\_count\_frame\*build\_count\_1946\_1970 build\_count\_frame\*build\_count\_1971\_1995 build\_count\_frame\*build\_count\_after\_1995 build\_count\_brick\*build\_count\_before\_1920 build\_count\_brick\*build\_count\_1921\_1945 build\_count\_brick\*build\_count\_1946\_1970 build\_count\_brick\*build\_count\_1971\_1995 build\_count\_brick\*build\_count\_after\_1995 X16\_29\_all\*fitness\_km swim\_pool\_km\*X0\_6\_all swim\_pool\_km\*X7\_14\_all X0\_17\_all\*swim\_pool\_km X0\_13\_all\*swim\_pool\_km ice\_rink\_km\*X0\_6\_all ice\_rink\_km\*X7\_14\_all X0\_17\_all\*ice\_rink\_km X0\_13\_all\*ice\_rink\_km office\_km\*X16\_29\_all

/ selection=backward( choose=CV stop=CV include = 40) CVdetails;

run;

quit;

/\* first, a correlation matrix to check collinearity \*/

proc corr data = DF noprob output=OutCorr nomiss

cov;

var price\_doc id timestamp full\_sq life\_sq floor max\_floor num\_room kitch\_sq product\_type raion\_popul green\_zone\_part indust\_part children\_preschool preschool\_quota children\_school healthcare\_centers\_raion university\_top\_20\_raion shopping\_centers\_raion office\_raion railroad\_terminal\_raion big\_market\_raion full\_all X0\_17\_all X16\_29\_all X0\_13\_all build\_count\_block build\_count\_wood build\_count\_frame build\_count\_brick build\_count\_before\_1920 build\_count\_1921\_1945 build\_count\_1946\_1970 build\_count\_1971\_1995 build\_count\_after\_1995 metro\_min\_avto metro\_km\_avto school\_km green\_zone\_km industrial\_km railroad\_station\_walk\_km ID\_railroad\_station\_walk railroad\_station\_avto\_km railroad\_station\_avto\_min public\_transport\_station\_km public\_trans\_station\_time\_walk kremlin\_km big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km market\_shop\_km fitness\_km swim\_pool\_km ice\_rink\_km stadium\_km basketball\_km public\_healthcare\_km university\_km workplaces\_km shopping\_centers\_km office\_km big\_church\_km;

run;

/\* then, a variance importance factor matrix to also check collinearity. We don't want anything that falls below a tolerance of 0.1\*/

proc reg data=df;

model price\_doc = id life\_sq floor max\_floor num\_room kitch\_sq product\_type green\_zone\_part indust\_part preschool\_quota children\_school healthcare\_centers\_raion university\_top\_20\_raion shopping\_centers\_raion railroad\_terminal\_raion big\_market\_raion X0\_17\_all X16\_29\_all build\_count\_block build\_count\_wood build\_count\_frame build\_count\_brick build\_count\_before\_1920 build\_count\_1921\_1945 build\_count\_1946\_1970 build\_count\_1971\_1995 build\_count\_after\_1995 metro\_km\_avto school\_km green\_zone\_km industrial\_km ID\_railroad\_station\_walk railroad\_station\_avto\_km public\_transport\_station\_km public\_trans\_station\_time\_walk kremlin\_km big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km market\_shop\_km fitness\_km swim\_pool\_km ice\_rink\_km stadium\_km public\_healthcare\_km university\_km workplaces\_km shopping\_centers\_km office\_km big\_church\_km / vif tol collin;

run;

/\* start interaction term visuals \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Plotting Interaction Terms Below \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ods graphics on;

proc glm data=DF;

model price\_doc = X0\_17\_all | X16\_29\_all / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=X0\_17\_all sliceby=X16\_29\_all) / clm;

run;

proc glm data=DF;

model price\_doc = children\_school | school\_km / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=children\_school sliceby=school\_km) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_block | build\_count\_1921\_1945 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_block sliceby=build\_count\_1921\_1945) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_block | build\_count\_1946\_1970 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_block sliceby=build\_count\_1946\_1970) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_block | build\_count\_1971\_1995 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_block sliceby=build\_count\_1971\_1995) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_block | build\_count\_after\_1995 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_block sliceby=build\_count\_after\_1995) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_wood | build\_count\_before\_1920 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_wood sliceby=build\_count\_before\_1920) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_wood | build\_count\_1946\_1970 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_wood sliceby=build\_count\_1946\_1970) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_wood | build\_count\_after\_1995 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_wood sliceby=build\_count\_after\_1995) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_frame | build\_count\_before\_1920 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_frame sliceby=build\_count\_before\_1920) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_frame | build\_count\_1921\_1945 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_frame sliceby=build\_count\_1921\_1945) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_frame | build\_count\_1946\_1970 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_frame sliceby=build\_count\_1946\_1970) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_frame | build\_count\_after\_1995 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_frame sliceby=build\_count\_after\_1995) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_brick | build\_count\_1946\_1970 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_brick sliceby=build\_count\_1946\_1970) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_brick | build\_count\_1971\_1995 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_brick sliceby=build\_count\_1971\_1995) / clm;

run;

proc glm data=DF;

model price\_doc = build\_count\_brick | build\_count\_after\_1995 / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=build\_count\_brick sliceby=build\_count\_after\_1995) / clm;

run;

proc glm data=DF;

model price\_doc = office\_km | X16\_29\_all / solution;

ods select ParameterEstimates ContourFit;

store GLMModel;

run;

/\* step two (final step) for plotting interaction terms \*/

proc plm restore=GLMModel noinfo;

effectplot slicefit(x=office\_km sliceby=X16\_29\_all) / clm;

run;

ods graphics off;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Plotting Interaction Terms Above \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* After updating using the suggestion from Backward on our interactions, filtering using correlation matrix and dropping using VIF threshold,

we continued removing interaction terms based on interaction plots. The above plots visualize the terms we left in the model below, which is

our best model. Although we reduced our adjusted R-squared by using the above methods (corr matrix, VIF, interaction plots), we feel this

was because the models were overfit. Below, we are confident in the fit.\*/

ods graphics on;

proc glm data = DF plots(unpack) = ALL;

proc glm data = DF plots(unpack) = (DIAGNOSTICS RESIDUALS);

model price\_doc = id life\_sq floor max\_floor num\_room kitch\_sq product\_type green\_zone\_part indust\_part preschool\_quota children\_school healthcare\_centers\_raion university\_top\_20\_raion shopping\_centers\_raion railroad\_terminal\_raion big\_market\_raion X0\_17\_all X16\_29\_all build\_count\_block build\_count\_wood build\_count\_frame build\_count\_brick build\_count\_before\_1920 build\_count\_1921\_1945 build\_count\_1946\_1970 build\_count\_1971\_1995 build\_count\_after\_1995 metro\_km\_avto school\_km green\_zone\_km industrial\_km ID\_railroad\_station\_walk railroad\_station\_avto\_km public\_transport\_station\_km public\_trans\_station\_time\_walk kremlin\_km big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km market\_shop\_km fitness\_km swim\_pool\_km ice\_rink\_km stadium\_km public\_healthcare\_km university\_km workplaces\_km shopping\_centers\_km office\_km big\_church\_km X0\_17\_all\*X16\_29\_all children\_school\*school\_km build\_count\_block\*build\_count\_1921\_1945 build\_count\_block\*build\_count\_1946\_1970 build\_count\_block\*build\_count\_1971\_1995 build\_count\_block\*build\_count\_after\_1995 build\_count\_wood\*build\_count\_before\_1920 build\_count\_wood\*build\_count\_1946\_1970 build\_count\_wood\*build\_count\_after\_1995 build\_count\_frame\*build\_count\_before\_1920 build\_count\_frame\*build\_count\_1921\_1945 build\_count\_frame\*build\_count\_1946\_1970 build\_count\_frame\*build\_count\_after\_1995 build\_count\_brick\*build\_count\_1946\_1970 build\_count\_brick\*build\_count\_1971\_1995 build\_count\_brick\*build\_count\_after\_1995 office\_km\*X16\_29\_all / cli;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* After updating model using OLS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* correlation matrix \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*and VIF model \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%let inputData = DF;

%let numObs = 19836; \*number of our cleanData observations (19,835) + 1;

%let numVarsLasso = 13;

%let lassoVars = id life\_sq floor kitch\_sq indust\_part office\_raion green\_zone\_km ID\_railroad\_station\_walk kremlin\_km fitness\_km swim\_pool\_km university\_km office\_km;

\*below are the linear variables selected outright by our best performing Ordinary Least Squares method (Backward Elimination);

%let numVarsOLS = 62;

%let OLSVars = id full\_sq life\_sq floor max\_floor num\_room kitch\_sq product\_type raion\_popul green\_zone\_part indust\_part children\_preschool preschool\_quota children\_school healthcare\_centers\_raion university\_top\_20\_raion shopping\_centers\_raion office\_raion railroad\_terminal\_raion big\_market\_raion full\_all X0\_17\_all X16\_29\_all X0\_13\_all build\_count\_block build\_count\_wood build\_count\_frame build\_count\_brick build\_count\_before\_1920 build\_count\_1921\_1945 build\_count\_1946\_1970 build\_count\_1971\_1995 build\_count\_after\_1995 metro\_min\_avto metro\_km\_avto school\_km green\_zone\_km industrial\_km railroad\_station\_walk\_km ID\_railroad\_station\_walk railroad\_station\_avto\_km railroad\_station\_avto\_min public\_transport\_station\_km public\_trans\_station\_time\_walk kremlin\_km big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km market\_shop\_km fitness\_km swim\_pool\_km ice\_rink\_km stadium\_km basketball\_km public\_healthcare\_km university\_km workplaces\_km shopping\_centers\_km office\_km big\_church\_km;

%let numVarsOLSFwd = 7; \* we may want to use these in the future to compare, but for now, we just need four models (same for OLSVarsFwd below);

%let OLSVarsFwd = id life\_sq floor max\_floor kitch\_sq indust\_part kremlin\_km;

/\* below we are setting the variables for the SQL we will eventually use for our custom model \*/

%let numVarsCustom = 69;

%let customOLSVars = id life\_sq floor max\_floor num\_room kitch\_sq product\_type green\_zone\_part indust\_part preschool\_quota children\_school healthcare\_centers\_raion university\_top\_20\_raion shopping\_centers\_raion railroad\_terminal\_raion big\_market\_raion X0\_17\_all X16\_29\_all build\_count\_block build\_count\_wood build\_count\_frame build\_count\_brick build\_count\_before\_1920 build\_count\_1921\_1945 build\_count\_1946\_1970 build\_count\_1971\_1995 build\_count\_after\_1995 metro\_km\_avto school\_km green\_zone\_km industrial\_km ID\_railroad\_station\_walk railroad\_station\_avto\_km public\_transport\_station\_km public\_trans\_station\_time\_walk kremlin\_km big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km market\_shop\_km fitness\_km swim\_pool\_km ice\_rink\_km stadium\_km public\_healthcare\_km university\_km workplaces\_km shopping\_centers\_km office\_km big\_church\_km X0\_17\_all\*X16\_29\_all children\_school\*school\_km build\_count\_block\*build\_count\_1921\_1945 build\_count\_block\*build\_count\_1946\_1970 build\_count\_block\*build\_count\_1971\_1995 build\_count\_block\*build\_count\_after\_1995 build\_count\_wood\*build\_count\_before\_1920 build\_count\_wood\*build\_count\_1946\_1970 build\_count\_wood\*build\_count\_after\_1995 build\_count\_frame\*build\_count\_before\_1920 build\_count\_frame\*build\_count\_1921\_1945 build\_count\_frame\*build\_count\_1946\_1970 build\_count\_frame\*build\_count\_after\_1995 build\_count\_brick\*build\_count\_1946\_1970 build\_count\_brick\*build\_count\_1971\_1995 build\_count\_brick\*build\_count\_after\_1995 office\_km\*X16\_29\_all;

/\* dependent variable is price\_doc \*/

%let depVar = price\_doc;

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\* External Cross-Validation Steps Here \*\*\*\*\*\*\*\*\*/

data DF;

set &inputData;

randNumber = ranuni(11);

if \_n\_ < &numObs;

run;

/\* build training data set for external cross-validation. We will train in 25% blocks, test on 75%. We feel this is reasonable.

75% for training \*/

data dfTrain;

set &inputData;

if randNumber <= 1/4 then delete;

run;

/\* build our test data set for external cross-validation; 25% for testing \*/

data dfTest;

set &inputData;

if randNumber > 1/4 then delete;

run;

ods graphics on;

title "Selection Method LASSO Using LASSO Variables and Cross Validation";

proc glmselect data=dfTrain testdata = dfTest

seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model &depVar = &lassoVars

/ selection=LASSO( choose=CV stop=CV ) CVdetails;

score data=dfTest out=scoredLASSO;

run;

quit;

ods graphics off;

ods graphics on;

title "Selection Method Backward Elimination Using LASSO Variables and OLS";

proc glmselect data=dfTrain testdata = dfTest

plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model &depVar = &lassoVars

/ selection=backward( choose=CV stop=CV include = &numVarsLASSO ) CVdetails;

score data=dfTest out=scoredOLSLASSO;

run;

quit;

ods graphics off;

ods graphics on;

title "Selection Method Backward Elimination Using OLS Variables and OLS";

proc glmselect data=dfTrain testdata = dfTest

plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model &depVar = &OLSVars

/ selection=backward( choose=CV stop=CV include = &numVarsOLS ) CVdetails; /\*default to 5 folds\*/

score data=dfTest out=scoredOLS;

run;

quit;

ods graphics off;

ods graphics on;/\* This sets up the SQL for our custom model cross validation \*/

title "Selection Method Custom Combined Backward Elimination, Corr Matrix, VIF Using OLS Variables";

proc glmselect data=dfTrain testdata = dfTest

plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model &depVar = &customOLSVars

/ selection=backward( choose=CV stop=CV include = &numVarsCustom ) CVdetails;

score data=dfTest out=scoredOLSvarsCustom;

run;

quit;

ods graphics off;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of External Cross-Validation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\* Internal Cross-Validation Steps Here \*\*\*\*\*\*\*\*\*/

ods graphics on;

title "Selection Method LASSO Using LASSO Variables and Cross Validation";

proc glmselect data=df

seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model &depVar = &lassoVars

/ selection=LASSO( choose=CV stop=CV ) CVdetails;

score data=df out=scoredLASSO;

run;

quit;

ods graphics off;

ods graphics on;

title "Selection Method Backward Elimination Using LASSO Variables and OLS";

proc glmselect data=df

plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model &depVar = &lassoVars

/ selection=backward( choose=CV stop=CV include = &numVarsLASSO ) CVdetails;

score data=df out=scoredOLSLASSO;

run;

quit;

ods graphics off;

ods graphics on;

title "Selection Method Backward Elimination Using OLS Variables and OLS";

proc glmselect data=df

plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model &depVar = &OLSVars

/ selection=backward( choose=CV stop=CV include = &numVarsOLS ) CVdetails; /\*default to 5 folds\*/

score data=df out=scoredOLS;

run;

quit;

ods graphics off;

ods graphics on;/\* This sets up the SQL for our custom model cross validation \*/

title "Selection Method Custom Combined Backward Elimination, Corr Matrix, VIF Using OLS Variables";

proc glmselect data=df

plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

model &depVar = &customOLSVars

/ selection=backward( choose=CV stop=CV include = &numVarsCustom ) CVdetails;

score data=df out=scoredOLSvarsCustom;

run;

quit;

ods graphics off;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Internal Cross-Validation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Calculate Sums of Squares from LASSO and OLS outputs \*/

proc sql;

create table fitLasso as

select

count(&depVar) as n

,css(&depVar) as totSS

,sum((&depVar - p\_&depvar)\*\*2) as errSSLasso

from scoredLasso;

create table fitOLSLasso as

select sum((&depVar - p\_&depvar)\*\*2) as errSSOLSLasso /\*THIS IS THE SAME AS BELOW \*/

from scoredOLSLasso;

create table fitOLS as

select sum((&depVar - p\_&depvar)\*\*2) as errSSOLS /\* THIS IS THE SAME AS ABOVE \*/

from scoredOLS;

create table customModel as /\* this is for our custom model, which will be our Regression Model \*/

select sum((&depVar - p\_&depvar)\*\*2) as errSScustomOLS

from scoredOLSvarsCustom;

quit;

run;

/\* Calculate rsq and adjRsq using sums of squares from LASSO and OLS outputs \*/

data allMeasures;

merge fitLasso fitOLSLasso fitOLS customModel;

rsqLasso = (1 - errSSLasso / totSS);

rsqOLSLasso = (1 - errSSOLSLasso / totSS);

rsqOLS = (1 - errSSOLS / totSS);

rsqCustomOLS = (1 - errSScustomOLS / totSS); /\* this is for our custom model \*/

adjRsqLasso = (1 - errSSLasso / totSS)\*((n - 1) / (n - &numVarsLasso - 1));

adjRsqOLSLasso = (1 - errSSOLSLasso / totSS)\*((n - 1) / (n - &numVarsLasso - 1));

adjRsqOLS = (1 - errSSOLS / totSS)\*((n - 1) / (n - &numVarsOLS - 1));

adjRsqCustomOLS = (1 - errSScustomOLS / totSS)\*((n - 1) / (n - &numVarsCustom - 1)); /\* this is for our custom model \*/

run;

title "Goodness of Fit Measures Using Test Data - All Models (LASSO with LASSO Vars, OLS with LASSO Vars, OLS, and Custom OLS)";

proc print data = allMeasures;

run;

/\*Prediction\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

FILENAME REFFILE '/home/u38493344/Applied Stats/Project 1/cleanTestData2.csv';

PROC IMPORT DATAFILE=REFFILE

DBMS=csv

OUT=cleanTest2;

getnames = yes;

RUN;

proc print data = cleanTest2(obs=10);

run;

data forPred1;

set DF cleanTest2;

run;

proc print data = forPred1(obs=10);

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*Best Model - Backwards with Interactions\*/

proc glm data = forPred1 plots = ALL;

class PRODUCT\_TYPE;

model price\_doc = id life\_sq floor max\_floor num\_room kitch\_sq product\_type green\_zone\_part indust\_part preschool\_quota children\_school healthcare\_centers\_raion university\_top\_20\_raion shopping\_centers\_raion railroad\_terminal\_raion big\_market\_raion X0\_17\_all X16\_29\_all build\_count\_block build\_count\_wood build\_count\_frame build\_count\_brick build\_count\_before\_1920 build\_count\_1921\_1945 build\_count\_1946\_1970 build\_count\_1971\_1995 build\_count\_after\_1995 metro\_km\_avto school\_km green\_zone\_km industrial\_km ID\_railroad\_station\_walk railroad\_station\_avto\_km public\_transport\_station\_km public\_trans\_station\_time\_walk kremlin\_km big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km market\_shop\_km fitness\_km swim\_pool\_km ice\_rink\_km stadium\_km public\_healthcare\_km university\_km workplaces\_km shopping\_centers\_km office\_km big\_church\_km X0\_17\_all\*X16\_29\_all children\_school\*school\_km build\_count\_block\*build\_count\_1921\_1945 build\_count\_block\*build\_count\_1946\_1970 build\_count\_block\*build\_count\_1971\_1995 build\_count\_block\*build\_count\_after\_1995 build\_count\_wood\*build\_count\_before\_1920 build\_count\_wood\*build\_count\_1946\_1970 build\_count\_wood\*build\_count\_after\_1995 build\_count\_frame\*build\_count\_before\_1920 build\_count\_frame\*build\_count\_1921\_1945 build\_count\_frame\*build\_count\_1946\_1970 build\_count\_frame\*build\_count\_after\_1995 build\_count\_brick\*build\_count\_1946\_1970 build\_count\_brick\*build\_count\_1971\_1995 build\_count\_brick\*build\_count\_after\_1995 office\_km\*X16\_29\_all;

output out = predsMLR p = prediction lcl = lower ucl = upper;

run;

proc print data = predsMLR(obs=10);

run;

proc export data = predsMLR

dbms = csv outfile="/home/u38493344/Applied Stats/Project 1/mlrPredictions.csv" replace;

run;

## Goal 2 SAS Code: Time-Series Modeling & Predictions

FILENAME REFFILE '/home/u38493344/Applied Stats/Project 1/timeSeriesB.xlsx';

PROC IMPORT DATAFILE=REFFILE

DBMS=xlsx

OUT=ts;

getnames = yes;

RUN;

proc print data = ts;

run;

/\*Goal2 - 4a.\*/

proc reg data=ts;

model AvgPrice = MonthNumber;

output out = residualData residual= residual;

run;

/\*Goal2 - 4b.\*/

proc sgplot data = residualData;

scatter x = MonthNumber y = residual;

series x = MonthNumber y = residual;

xaxis label= "Month";

run;

/\*Goal2 - 4c.(No Lag)\*/

proc autoreg data = ts;

model AvgPrice = monthNumber/dwprob;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*Goal2 - 4c.(AR(1))\*/

proc autoreg data = ts plots = residual;

model AvgPrice = MonthNumber/nlag=(1) dwprob;

output out = residData residual= residual;

run;

proc sgplot data = residData;

scatter x = MonthNumber y = residual;

series x = MonthNumber y = residual;

xaxis label= "Month";

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*Goal2 - 4d. & 5\*/

data predict;

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 ts predict;

run;

proc autoreg data = forPred plots(unpack);

model AvgPrice = MonthNumber/nlag=(1) dwprob;

output out = preds p = prediction lcl = lower ucl = upper pm = trend residual=resid;

run;

proc print data = preds; run;

proc sgplot data = preds;

band x = MonthNumber upper = upper lower = lower;

scatter x = MonthNumber y = prediction;

series x = MonthNumber y = prediction;

series x = MonthNumber y = trend/lineattrs = (color=black);

xaxis label= "Month";

run;

/\*Question 6\*/

proc export data = preds

dbms = csv outfile="/home/u38493344/Applied Stats/Project 1/tsPredictions.csv" replace;

run;