**Goal 1: Prediction of Individual Property Values**

**Introduction**

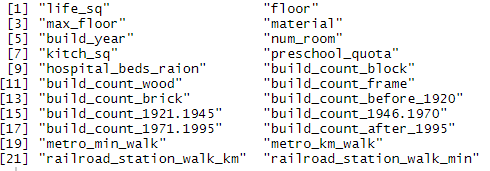
Sberbank, Russia’s oldest and largest banks, wants to help their customers to make predictions about realty prices. We are given a data set that includes 5000 properties which includes the sale price as well as 72 other features about each observation. Our goal is to create a model that can predict the individual property prices with the data set given as accurately as possible.

**Data Cleaning**

So, the data is split into the modeling data and the projection data. The modeling data contains 25471 observations and that is what we will be using to help build our prediction model. In terms of cleaning the data, (1) we modified the timestamp variable so instead of just showing a numerical value we changed it to a date format; and (2) removed two id columns (id and ID\_railroad\_station\_walk) because they do not add any value in the prediction of the prices.

**Exploratory Data Analysis (EDA)**

1. **(Missing Values)** First part of the EDA, we need to deal with missing values. We first need to check if there are missing values and which columns that contain them. The following 22 columns contain missing values:



To deal with the issue of missing values we decided to use imputation. For the columns that are numerical variables, if they are continuous, we substituted the NA value for the mean value of the column and if they were discrete, we substituted them with the rounded mean of the column. For the categorical variables, we substituted the NA value with the mode of the column. After doing that we were left with no NA values in the data set:



1. **(Variable Selection)** We have a total of 70 explanatory variables, so in an effort to cut down as many variables as we can we created a stepwise regression model with all the explanatory variables and the following ends up being the result:

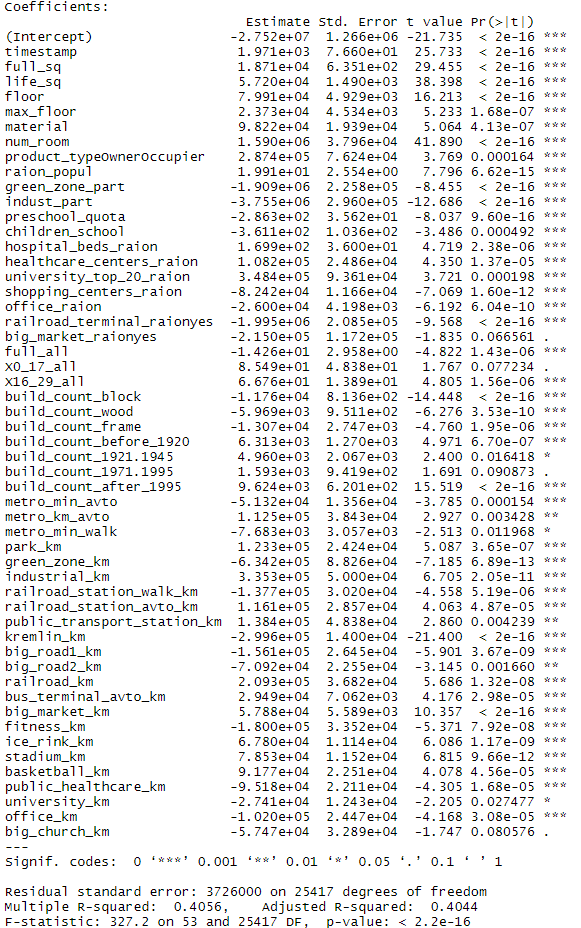


Table 1: results of step wise variable selection as part of EDA

After running the stepwise regression, we are left with the variables above. Since our α = .05 we will remove any of the variables that have p-value above .05, which means we remove big\_market\_raion, X0\_17\_all, build\_count\_1971.1995, and big\_church\_km.

1. **(MultiColinearity)** In this portion of the EDA, we create a data set with columns we have left and brought it over to SAS. We run the proc reg function with data so we can find the VIF of each variable. The below table shows the result:

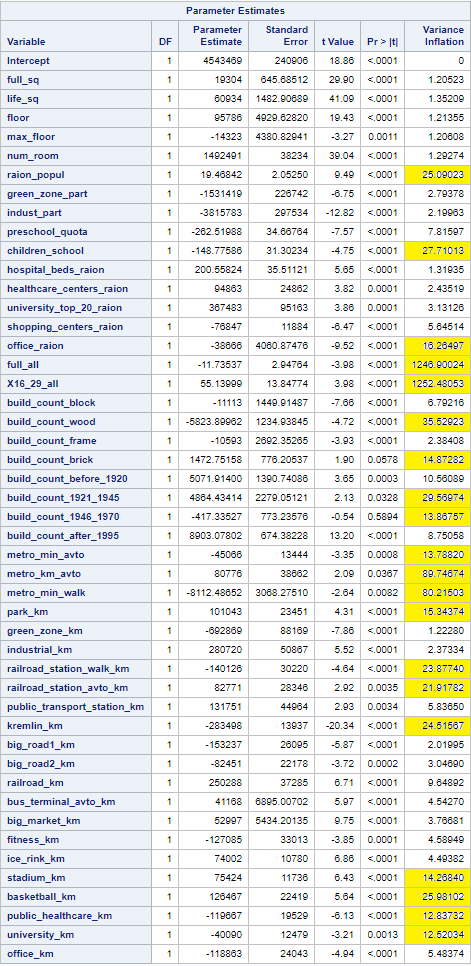


Table 2: Variable inflaction factor (VIF) > 10 indicative of highly correlated variable. Not included further in modeling.

The VIF values that are considerably higher than 10 have been highlighted in yellow letting us know that they have high correlation with another variable. So with that, we remove children\_school, X16\_29\_all, build\_count\_wood, build\_count\_brick, build\_count\_1921\_1945, metro\_km\_avto, metro\_min\_walk, railroad\_station\_walk\_km, kremlin\_km, stadium\_km, basketball\_km, and public\_healthcare\_km

1. (**Checking Assumptions).** We used log transformation on the dependent variable (price\_doc) of our multiple regression. As you can see in the histograms below, the price\_doc distribution was significantly right skewed, as can be expected with continuous random variables which have no limit to their maximum boundary, yet are limited by a minimum boundary of zero. After a log transformation, our dependent variable much more closely resembles a normal distribution.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Table3: Residues and long transformed dependent variable indicating log transformed (left column) is better. .

It is noticeable that the residuals after the transformation no longer exhibit signs of heteroscedasticity and collinearity, complying with assumptions of a linear model.

**Modeling**

**Number of independent variables used is 40** for the modeling, after removing collinear variables from exploratory data analysis.

**Both OLS and LASSO analysis are modeled using following independent variables**

full\_sq life\_sq floor max\_floor num\_room raion\_popul green\_zone\_part indust\_part preschool\_quota hospital\_beds\_raion healthcare\_centers\_raion university\_top\_20\_raion shopping\_centers\_raion office\_raion full\_all build\_count\_block build\_count\_frame build\_count\_before\_1920 build\_count\_1946\_1970 build\_count\_after\_1995 metro\_min\_avto park\_km green\_zone\_km industrial\_km railroad\_station\_avto\_km public\_transport\_station\_km big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km fitness\_km ice\_rink\_km university\_km office\_km

1. **OLS Stepwise method**

Stepwise OLS with external and internal cross validation is run. Refer tables below for summary on fitting estimates and parameter estimates.

27 effects were found to be significant at significance level = 0.05; and factored into the OLS model.

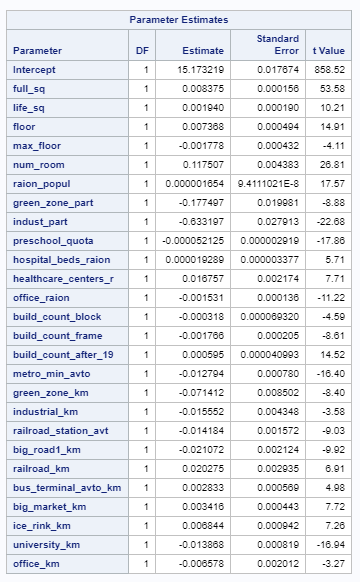
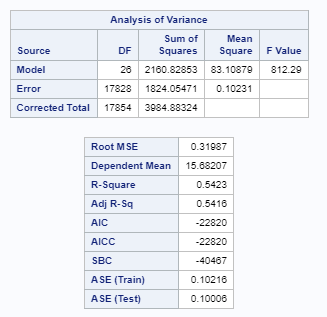
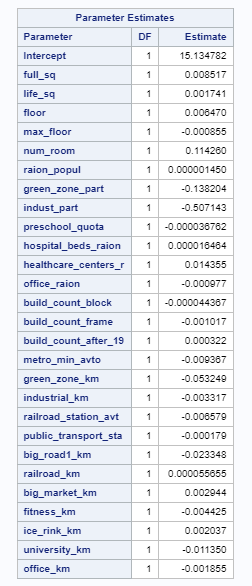


Table 5: model fitting information for OLS

1. **LASSO method**

LASSO with external and internal cross validation is run. Refer tables below for summary on fitting estimates and parameter estimates.

28 effects were factored into the LASSO model.



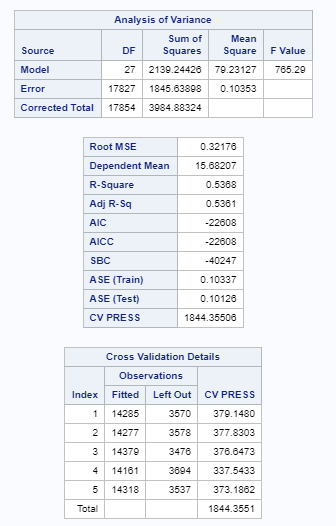


Table 6: model fitting information for LASSO

Using Proc Reg. and above parameters, two variables were removed as they were found to be not significant,

* public\_transport\_station\_km

1. **OLS Backward method**

Backward OLS with external and internal cross validation is run. Refer tables below for summary on fitting estimates and parameter estimates.

29 effects were found to be significant at significance level = 0.05; and factored into the OLS model.

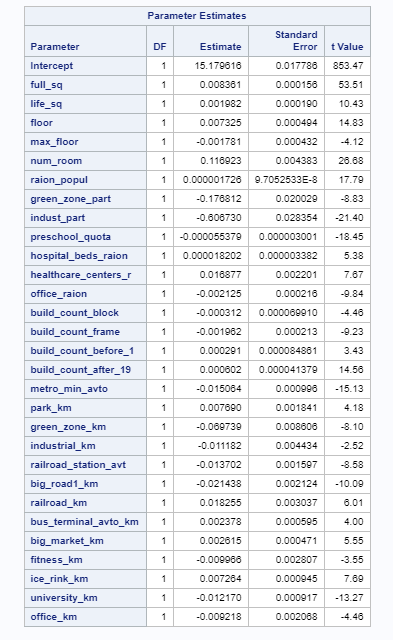
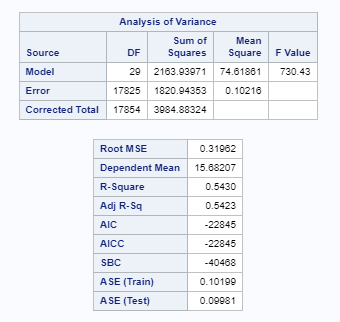


Table 6: model fitting information for Backward OLS

**Prediction**

1. **Which model is best?**

LASSO provides the lowest AIC value (22608) vs Stepwise & Backward OLS (22820 & 22845 respectively.

The adj. R sq for LASSO (0.5361) is slightly lower than Stepwise (0.5416) and Backward OLS (0.5423)

***We go with LASSO that provides smallest AIC value***, and use it for prediction

***RMSLE value calculated using prediction on modeling data is = 0.3212048*** *(Refer spreadsheet rmsle.xlsx as part of submission)*

1. **The predictions can be found in the file submission.csv**

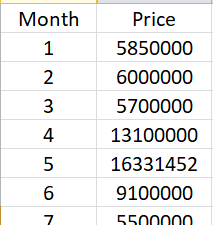
**Goal 2: Prediction of Mean Property Value by Year**

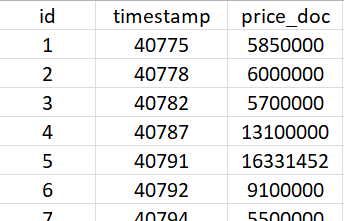
**Introduction (Q1)**

Housing price prediction is a key analysis needed by prospective home buyers, lenders, developers. Multiple features of a house and broad macroeconomic factors can influence property price in a year.

As part of the study, the housing data made available (refer Appendex 1) is used to predict mean price of properties from July 2015 to July 2016

**Data Wrangling (Q2)**

The primary data wrangling component is to transform available house price data into a time series data, i.e.



**R script**

Fig. 2: Time series data set used for further analysis

Fig. 1: Data of interest from complete house price data set

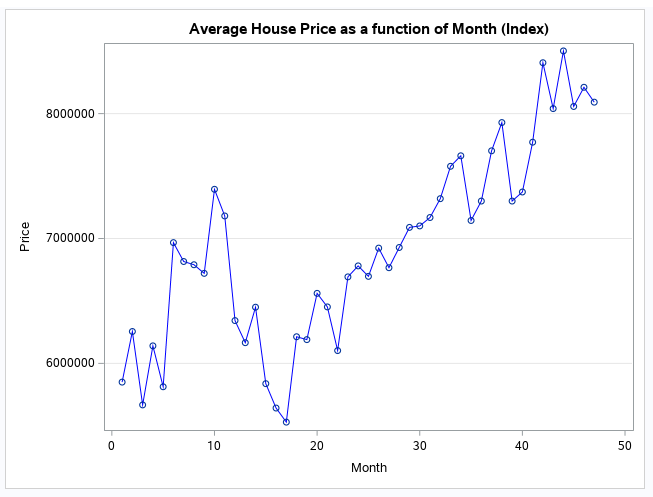
**Housing Price Trend (Q3)**

Figure on the right shows the mean house property trend starting Aug 2011, to June 2015, i.e. 47 months.

Fig. 3: Mean house property value for 47 months

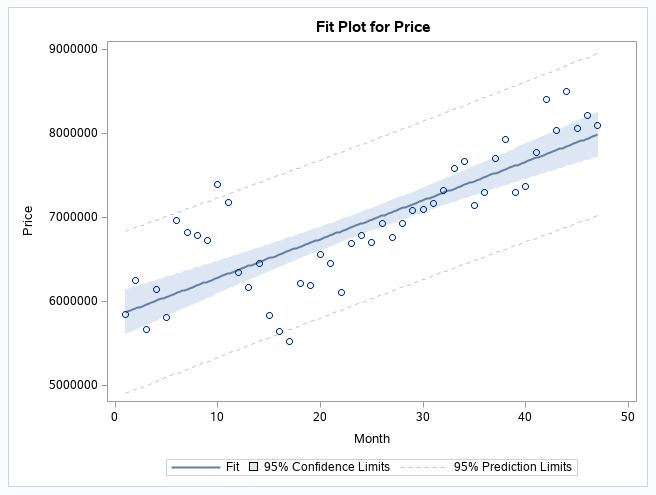
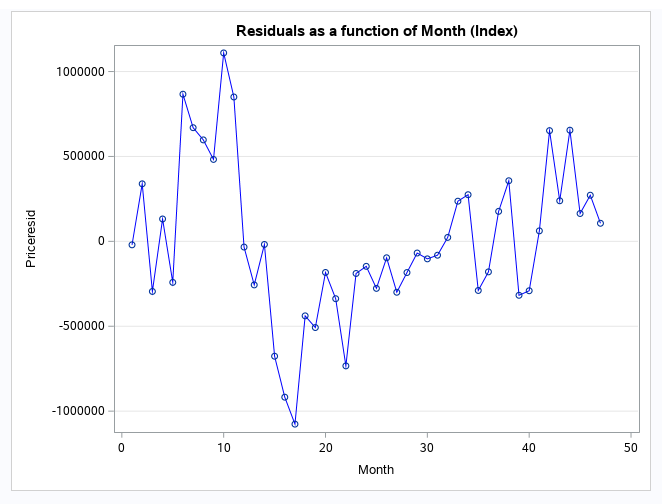
**Residual Modeling and Prediction (Q4)**

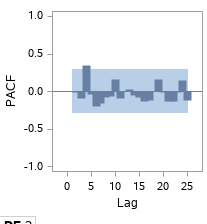
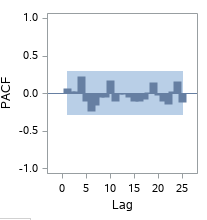
Fig. 5: Residual from OLS fit of mean House Price against the month number (Q4b)

Fig. 4: Ordinary Least Square fit of mean House Price against month number (Q4a)

Simple linear regression model fit is significant with a p value = <0.001 at significance of 0.05. However review of residual plot doesn’t indicate random distribution across independent variable. There is likely a presence of serial (Auto correlation) in the data. (Q4c)

The AIC, SBC and DW statistic based assessment indicates that AR(1) model is sufficient in fixing the autocorrelation component in data, however PACF plot show support for AR(4) model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case | nlag option  (SAS proc autoreg) | DW statistic | AIC | SBC | Inference for autocorrelation AR(1) significant using AIC, SBC and DW statistic |
| 1 | 0 | 0.7524 | 1361.33848 | 1365.03878 | Yes : 0.7524 < 1.48 |
| 2 | 1 | 2.1502 | 1340.68462 | 1346.23506 | No : 2.1502 > 1.57 |
| 3 | 1,2 | 1.9975 | 1341.98697 | 1349.38756 | No : 1.9975 > 1.57 |
| 4 | 1,2,3 | 1.9949 | 1343.9702 | 1353.22094 | No : 1.9949 > 1.57 |
| 5 | 1,2,3,4 | 1.8785 | 1343.96542 | 1355.0663 | No : 1.8785 > 1.57 |



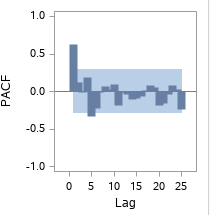
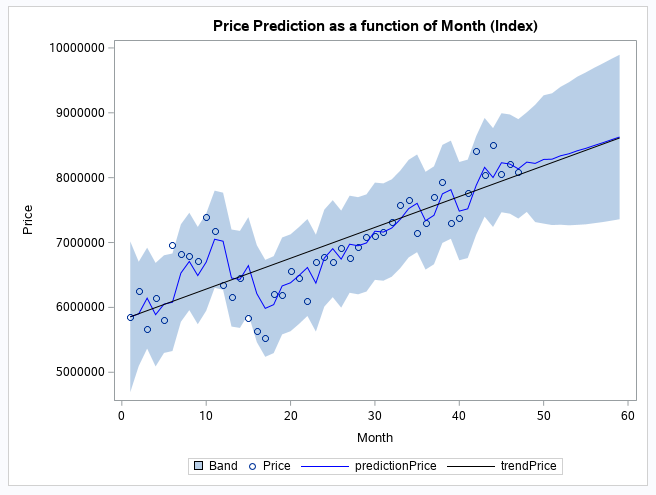


Fig. 5: PACF summary for different lag options (no log on left, to lag = 4 (rightmost figure) lag = 4

Residual forecast for next year is captured in the table on right. (Q4d)

**Housing Price Prediction (Q5)**

House price prediction using proc autoreg (nlag=4). Refer plot and table below.

House price prediction using earlier computed residues and trends is shown in figure on right. Following is observation:

* Price predicted is the same, while the confidence interval is much together when computed using residuals.

**Forecast Submission**

Files submitted alongwith are:

* HousePricePredictionData\_SAS.csv : data directly from proc autoreg SAS procedure
* HousePricePredictionData\_Generated.csv : data generated by summing residual prediction (from SAS proc autoreg) and trend prediction (OLS fitting)

**Appendix 1: Data Dictionary**

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

***Legends***

Variables part of final model

\* Categorical Variable

**Appendix 2: Programming Codes**

**Goal 1**

**R Code:**

library(ggplot2)

library(dplyr)

library(GGally)

library(tidyverse)

library(stringr)

library(MASS)

library(olsrr)

Df = read.csv("C:/Users/Mrinmoy/Documents/School/Applied Statistics/Project 1/modelingData.csv")

sum(is.na(Df))

head(Df)

#Convert timestamp to date type

Df$timestamp <- do.call("c",lapply(Df$timestamp, function(x) as.Date(x, origin = "1899-12-30")))

#Remove id columns

Df$id <- NULL

Df$ID\_railroad\_station\_walk <- NULL

#Checking which columns contain NA values

colnames(Df)[colSums(is.na(Df)) > 0]

#Mode function

getmode <- function(v) {

uniqv <- unique(v)

uniqv[which.max(tabulate(match(v, uniqv)))]

}

#Mode when number of NA values is the greatest

Mode <- function(x) {

ux <- na.omit(unique(x) )

tab <- tabulate(match(x, ux)); ux[tab == max(tab) ]

}

#Replace NA with mean for column

Df$life\_sq[is.na(Df$life\_sq)] <- mean(Df$life\_sq, na.rm = TRUE)

Df$kitch\_sq[is.na(Df$kitch\_sq)] <- round(mean(Df$kitch\_sq, na.rm = TRUE))

Df$preschool\_quota[is.na(Df$preschool\_quota)] <- round(mean(Df$preschool\_quota, na.rm = TRUE))

Df$hospital\_beds\_raion[is.na(Df$hospital\_beds\_raion)] <- round(mean(Df$hospital\_beds\_raion, na.rm = TRUE))

Df$build\_count\_brick[is.na(Df$build\_count\_brick)] <- round(mean(Df$build\_count\_brick, na.rm = TRUE))

Df$build\_count\_wood[is.na(Df$build\_count\_wood)] <- round(mean(Df$build\_count\_wood, na.rm = TRUE))

Df$build\_count\_frame[is.na(Df$build\_count\_frame)] <- round(mean(Df$build\_count\_frame, na.rm = TRUE))

Df$build\_count\_block[is.na(Df$build\_count\_block)] <- round(mean(Df$build\_count\_block, na.rm = TRUE))

Df$build\_count\_before\_1920[is.na(Df$build\_count\_before\_1920)] <- round(mean(Df$build\_count\_before\_1920, na.rm = TRUE))

Df$build\_count\_1921.1945[is.na(Df$build\_count\_1921.1945)] <- round(mean(Df$build\_count\_1921.1945, na.rm = TRUE))

Df$build\_count\_1946.1970[is.na(Df$build\_count\_1946.1970)] <- round(mean(Df$build\_count\_1946.1970, na.rm = TRUE))

Df$build\_count\_1971.1995[is.na(Df$build\_count\_1971.1995)] <- round(mean(Df$build\_count\_1971.1995, na.rm = TRUE))

Df$build\_count\_after\_1995[is.na(Df$build\_count\_after\_1995)] <- round(mean(Df$build\_count\_after\_1995, na.rm = TRUE))

Df$metro\_min\_walk[is.na(Df$metro\_min\_walk)] <- mean(Df$metro\_min\_walk, na.rm = TRUE)

Df$metro\_km\_walk[is.na(Df$metro\_km\_walk)] <- mean(Df$metro\_km\_walk, na.rm = TRUE)

Df$railroad\_station\_walk\_km[is.na(Df$railroad\_station\_walk\_km)] <- mean(Df$railroad\_station\_walk\_km, na.rm = TRUE)

Df$railroad\_station\_walk\_min[is.na(Df$railroad\_station\_walk\_min)] <- mean(Df$railroad\_station\_walk\_min, na.rm = TRUE)

#Replace NA with mode for column

Df$floor[is.na(Df$floor)] <- getmode(Df$floor)

Df$max\_floor[is.na(Df$max\_floor)] <- Mode(Df$max\_floor)

Df$material[is.na(Df$material)] <- getmode(Df$material)

Df$build\_year[is.na(Df$build\_year)] <- Mode(Df$build\_year)

Df$num\_room[is.na(Df$num\_room)] <- Mode(Df$num\_room)

#Checking which columns again for NA values

colnames(Df)[colSums(is.na(Df)) > 0]

#Using step-wise variable selection to decrease number of variables

full.model <- lm(price\_doc~.,data=Df)

step.model <- stepAIC(full.model,direction = "both",trace = FALSE)

summary(step.model)

#Checking assumptions of OLS Stepwise Model

par(mfrow=c(2,2))

plot(step.model)

#Data set created to check multi-colinearity between explanatory variables

Df2 = Df %>% dplyr::select(price\_doc,timestamp,full\_sq,life\_sq,floor,max\_floor,material,num\_room,product\_type,raion\_popul,green\_zone\_part,

indust\_part,preschool\_quota,children\_school,hospital\_beds\_raion,healthcare\_centers\_raion,university\_top\_20\_raion,

shopping\_centers\_raion,office\_raion,railroad\_terminal\_raion,full\_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\_after\_1995,

build\_count\_1921.1945,build\_count\_after\_1995,metro\_min\_avto,metro\_km\_avto,metro\_min\_walk,park\_km,

green\_zone\_km,industrial\_km,railroad\_station\_walk\_km,railroad\_station\_avto\_km,public\_transport\_station\_km,kremlin\_km,

big\_road1\_km,big\_road2\_km,railroad\_km,bus\_terminal\_avto\_km,big\_market\_km,fitness\_km,

ice\_rink\_km,stadium\_km,basketball\_km,public\_healthcare\_km,university\_km,office\_km)

write.csv(Df2,"C:/Users/Mrinmoy/Documents/School/Applied Statistics/Project 1/Df2.csv")

# Data set created after removing correlated explanatory variables

Df3 = Df %>% dplyr::select(price\_doc,timestamp,full\_sq,life\_sq,floor,max\_floor,material,num\_room,product\_type,raion\_popul,green\_zone\_part,

indust\_part,preschool\_quota,hospital\_beds\_raion,healthcare\_centers\_raion,university\_top\_20\_raion,

shopping\_centers\_raion,office\_raion,railroad\_terminal\_raion,full\_all,X16\_29\_all,build\_count\_block,build\_count\_frame,

build\_count\_before\_1920,build\_count\_1946.1970,build\_count\_after\_1995,

build\_count\_after\_1995,metro\_min\_avto,park\_km,

green\_zone\_km,industrial\_km,railroad\_station\_avto\_km,public\_transport\_station\_km,

big\_road1\_km,big\_road2\_km,railroad\_km,bus\_terminal\_avto\_km,big\_market\_km,fitness\_km,

ice\_rink\_km,university\_km,office\_km)

write.csv(Df3,"C:/Users/Mrinmoy/Documents/School/Applied Statistics/Project 1/Df3.csv")

#Clean the projection data as well as remove NA values

projectionData = read.csv("C:/Users/Mrinmoy/Documents/School/Applied Statistics/Project 1/projectionData.csv")

projectionData$timestamp <- do.call("c",lapply(projectionData$timestamp, function(x) as.Date(x, origin = "1899-12-30")))

colnames(projectionData)[colSums(is.na(projectionData)) > 0]

projectionData$ID\_railroad\_station\_walk <- NULL

projectionData$life\_sq[is.na(projectionData$life\_sq)] <- mean(projectionData$life\_sq, na.rm = TRUE)

projectionData$kitch\_sq[is.na(projectionData$kitch\_sq)] <- round(mean(projectionData$kitch\_sq, na.rm = TRUE))

projectionData$preschool\_quota[is.na(projectionData$preschool\_quota)] <- round(mean(projectionData$preschool\_quota, na.rm = TRUE))

projectionData$hospital\_beds\_raion[is.na(projectionData$hospital\_beds\_raion)] <- round(mean(projectionData$hospital\_beds\_raion, na.rm = TRUE))

projectionData$build\_count\_brick[is.na(projectionData$build\_count\_brick)] <- round(mean(projectionData$build\_count\_brick, na.rm = TRUE))

projectionData$build\_count\_wood[is.na(projectionData$build\_count\_wood)] <- round(mean(projectionData$build\_count\_wood, na.rm = TRUE))

projectionData$build\_count\_frame[is.na(projectionData$build\_count\_frame)] <- round(mean(projectionData$build\_count\_frame, na.rm = TRUE))

projectionData$build\_count\_block[is.na(projectionData$build\_count\_block)] <- round(mean(projectionData$build\_count\_block, na.rm = TRUE))

projectionData$build\_count\_before\_1920[is.na(projectionData$build\_count\_before\_1920)] <- round(mean(projectionData$build\_count\_before\_1920, na.rm = TRUE))

projectionData$build\_count\_1921.1945[is.na(projectionData$build\_count\_1921.1945)] <- round(mean(projectionData$build\_count\_1921.1945, na.rm = TRUE))

projectionData$build\_count\_1946.1970[is.na(projectionData$build\_count\_1946.1970)] <- round(mean(projectionData$build\_count\_1946.1970, na.rm = TRUE))

projectionData$build\_count\_1971.1995[is.na(projectionData$build\_count\_1971.1995)] <- round(mean(projectionData$build\_count\_1971.1995, na.rm = TRUE))

projectionData$build\_count\_after\_1995[is.na(projectionData$build\_count\_after\_1995)] <- round(mean(projectionData$build\_count\_after\_1995, na.rm = TRUE))

projectionData$metro\_min\_walk[is.na(projectionData$metro\_min\_walk)] <- mean(projectionData$metro\_min\_walk, na.rm = TRUE)

projectionData$metro\_km\_walk[is.na(projectionData$metro\_km\_walk)] <- mean(projectionData$metro\_km\_walk, na.rm = TRUE)

projectionData$railroad\_station\_walk\_km[is.na(projectionData$railroad\_station\_walk\_km)] <- mean(projectionData$railroad\_station\_walk\_km, na.rm = TRUE)

projectionData$railroad\_station\_walk\_min[is.na(projectionData$railroad\_station\_walk\_min)] <- mean(projectionData$railroad\_station\_walk\_min, na.rm = TRUE)

#Replace NA with mode for column

projectionData$floor[is.na(projectionData$floor)] <- getmode(projectionData$floor)

projectionData$max\_floor[is.na(projectionData$max\_floor)] <- Mode(projectionData$max\_floor)

projectionData$material[is.na(projectionData$material)] <- getmode(projectionData$material)

projectionData$build\_year[is.na(projectionData$build\_year)] <- Mode(projectionData$build\_year)

projectionData$num\_room[is.na(projectionData$num\_room)] <- Mode(projectionData$num\_room)

colnames(projectionData)[colSums(is.na(projectionData)) > 0]

write.csv(projectionData,"C:/Users/Mrinmoy/Documents/School/Applied Statistics/Project 1/projectionData1.csv")

#Calculate prediction with LASSO Model created in SAS using R

df4 <- read.csv("C:/Users/Mrinmoy/Documents/School/Applied Statistics/Project 1/df4.csv", header = T)

LASSOModel <- lm(logprice\_doc~ full\_sq+num\_room+university\_km+office\_km+metro\_min\_avto +

raion\_popul+ fitness\_km+healthcare\_centers\_raion+railroad\_station\_avto\_km+ big\_road1\_km +

life\_sq + indust\_part+floor+hospital\_beds\_raion+green\_zone\_km+ big\_market\_km+

green\_zone\_part+build\_count\_after\_1995+office\_raion+build\_count\_frame+max\_floor+ ice\_rink\_km+

industrial\_km+build\_count\_block+railroad\_km, data = df4)

pred<-predict(LASSOModel,projectionData,se.fit = TRUE)

pred$fit <- exp(pred$fit)

pred$fit

submission <- as.data.frame(pred$fit)

title "Checking for Multi-Colinearity with VIF";

proc import datafile="/home/u41135843/sasuser.v94/Df2.csv"

out= Df2 dbms=csv replace;

getnames=yes;

run;

proc reg data = Df2 PLOTS(MAXPOINTS=NONE) ;

model price\_doc = full\_sq life\_sq floor max\_floor num\_room raion\_popul green\_zone\_part

indust\_part preschool\_quota hospital\_beds\_raion healthcare\_centers\_raion university\_top\_20\_raion

shopping\_centers\_raion office\_raion full\_all build\_count\_block build\_count\_frame

build\_count\_before\_1920 build\_count\_1946\_1970 build\_count\_after\_1995

metro\_min\_avto park\_km

green\_zone\_km industrial\_km railroad\_station\_avto\_km public\_transport\_station\_km

big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km fitness\_km

ice\_rink\_km university\_km office\_km /vif;

run;

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

This program compares LASSO and OLS method variable selection results

using error ss and adjusted RSQ;

It uses coefficients from estimation using 75% of the input data;

It estimates adjusted RSQ using 25% of the input data;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/;

libname xl XLSX '/home/u41977007/DS6372/df4.xlsx';

%let inputDataset = xl.df4;

%let numObs = 23628; \*\*\* number of observations + 1;

%let numVarsLasso = 40; \*\*\* number of variables selected by Lasso;

%let lassoVars = full\_sq life\_sq floor max\_floor num\_room raion\_popul green\_zone\_part

indust\_part preschool\_quota hospital\_beds\_raion healthcare\_centers\_raion university\_top\_20\_raion

shopping\_centers\_raion office\_raion full\_all build\_count\_block build\_count\_frame

build\_count\_before\_1920 build\_count\_1946\_1970 build\_count\_after\_1995

metro\_min\_avto park\_km

green\_zone\_km industrial\_km railroad\_station\_avto\_km public\_transport\_station\_km

big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km fitness\_km

ice\_rink\_km university\_km office\_km ; \*\*\* list of variables from LASSO Selection;

%let numVarsOLS = 40; \*\*\* number of variables selected using OLS methods;

%let OLSVars = full\_sq life\_sq floor max\_floor num\_room raion\_popul green\_zone\_part

indust\_part preschool\_quota hospital\_beds\_raion healthcare\_centers\_raion university\_top\_20\_raion

shopping\_centers\_raion office\_raion full\_all build\_count\_block build\_count\_frame

build\_count\_before\_1920 build\_count\_1946\_1970 build\_count\_after\_1995

metro\_min\_avto park\_km

green\_zone\_km industrial\_km railroad\_station\_avto\_km public\_transport\_station\_km

big\_road1\_km big\_road2\_km railroad\_km bus\_terminal\_avto\_km big\_market\_km fitness\_km

ice\_rink\_km university\_km office\_km; \*\*\* list of variables from selected using OLS methods;

%let depVar = logprice\_doc; \*\*\* dependent (response) variable for models;

data original; set &inputDataset;

data inDat; set &inputDataset; randNumber = ranuni(11); if \_n\_ < &numObs; run;

data train; set inDat; if randNumber <= 1/4 then delete; run;

data test; set inDat; if randNumber > 1/4 then delete; run;

ods graphics on;

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

proc glmselect data=train testdata = test

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

model &depVar = &lassoVars

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

score data=test out=scoredLASSO;

run;

quit;

ods graphics off;

title "Running proc reg to identify significant parameters selected from LASSO";

proc reg data = original;

model &depVar = full\_sq life\_sq floor max\_floor num\_room raion\_popul green\_zone\_part indust\_part preschool\_quota hospital\_beds\_raion healthcare\_centers\_raion office\_raion build\_count\_block build\_count\_frame build\_count\_after\_1995 metro\_min\_avto green\_zone\_km industrial\_km railroad\_station\_avto\_km public\_transport\_station\_km big\_road1\_km railroad\_km big\_market\_km fitness\_km ice\_rink\_km university\_km office\_km;

run;

title "Re-Running proc reg to estimate parameters of only sigificant parameters (selected from LASSO)";

proc reg data = original;

model &depVar = full\_sq life\_sq floor max\_floor num\_room raion\_popul green\_zone\_part indust\_part preschool\_quota hospital\_beds\_raion healthcare\_centers\_raion office\_raion build\_count\_block build\_count\_frame build\_count\_after\_1995 metro\_min\_avto green\_zone\_km industrial\_km railroad\_station\_avto\_km big\_road1\_km railroad\_km big\_market\_km ice\_rink\_km university\_km office\_km;

run;

ods graphics on;

title "Selection Method Stepwise OLS";

proc glmselect data=train testdata = test

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

model &depVar = &lassoVars

/ selection=stepwise( choose=adjrsq stop=adjrsq ) CVdetails;

score data=test out=scoredOLSLasso;

run;

quit;

**Goal 2**

R data wrangling

df <- read.table("modelingData.csv",sep = ",", quote = "", header = TRUE)

df$timestamp = as.Date(df$timestamp, origin = "1899-12-30")

df$year = year(df$timestamp)

df$month = df$month <- format.Date(df$timestamp, "%m")

df$day = day(df$timestamp)

df$ym <- paste(df$year,df$month,sep="")

monthlyPrice <- aggregate(df$price\_doc, by = list(YrMonth = df$ym),FUN = mean)

monthlyPrice$Month <- seq.int(nrow(monthlyPrice))

names(monthlyPrice)[2] <- "Price"

monthlyPrice

write.table(monthlyPrice,file = "timeSeries.csv",sep = ",",col.names = TRUE, row.names = FALSE )

SAS modeling

libname xl XLSX '/home/u41977007/DS6372/timeSeries.xlsx';

proc print data = xl.timeSeries;

run;

\* Printing House Price as a function of Month;

ods graphics on;

title1 "Average House Price as a function of Month (Index)";

proc sgplot data=xl.timeseries;

series X = Month Y = Price /lineattrs = (color=blue) markers;

yaxis grid ;

run;

\* Using OLS model fit and getting residuals;

proc glm data=xl.timeseries PLOTS(UNPACK)=DIAGNOSTICS;

model Price = Month;

output out = dOLSResidues r=Priceresid;

run;

proc print data = dOLSResidues;

run;

\* Plotting Residuals from OLS as a function of Month;

title1 "Residuals as a function of Month (Index)";

proc sgplot data=dOLSResidues;

series X = Month Y = Priceresid /lineattrs = (color=blue) markers;

yaxis grid;

run;

/\* Identifying autocorrelation in the residual\*/

proc autoreg data=dOLSResidues;

model Priceresid = Month / dwprob;

run;

proc autoreg data=dOLSResidues;

model Priceresid = Month / nlag=(1 2 3 4) dwprob;

run;

/\* Preparing month data for which predictions need to be made \*/

data addObs;

input YrMonth Month @@; cards;

201507 48

201508 49

201509 50

201510 51

201511 52

201512 53

201601 54

201602 55

201603 56

201604 57

201605 58

201606 59

;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\* Generating residual prediction \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Preparing month data for which predictions need to be made \*/

data forPredOLSResiduals;

set dOLSResidues addObs;

run;

proc print data = forPredOLSResiduals;

run;

proc autoreg data = forPredOLSResiduals;

model Priceresid = Month / nlag=(1 2 3 4) dwprob;

output out = dPredResiduals p = prediction lcl = lower ucl = upper pm = trend;

run;

proc print data = dPredResiduals; run;

/\* residual predictions by month \*/

title1 "Price Residual Prediction as a function of Month (Index)";

proc sgplot data=dPredResiduals;

band x = Month upper = upper lower = lower;

scatter x = Month y = Priceresid;

series x = Month y = prediction/lineattrs = (color=blue);

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

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\* Generating Trend prediction \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Preparing month data for which predictions need to be made \*/

data forPredOLSMean;

set xl.timeseries addObs;

run;

proc print data = forPredOLSMean;

run;

proc glm data=forPredOLSMean PLOTS(UNPACK)=DIAGNOSTICS;

model Price = Month;

output out = dPredMeanPrice p=Pricehat;

run;

proc print data = dPredMeanPrice;

run;

/\* Superset data containing predicted residual mean values, etc \*/

data dAll;

set dPredMeanPrice;

set dPredResiduals;

varErr = prediction\*\*2;

predPrice = Pricehat + prediction;

clu = predPrice + 1.65 \* sqrt(varErr);

cll = predPrice - 1.65 \* sqrt(varErr);

run;

proc print data = dAll;

run;

/\* Plot for mean price prediction and confidence limits generated from earlier data\*/

proc sgplot data=dAll;

band x = Month upper = clu lower = cll;

scatter x = Month y = Price;

series x = Month y = predPrice/lineattrs = (color=blue);

series x = Month y = pricehat/lineattrs = (color=black);

run;

/\* Using proc autoreg to generated the prediction data on house price\*/

proc autoreg data = forPredOLSMean;

model Price = Month / nlag=(1 2 3 4) dwprob;

output out = dPredsPrice p = predictionPrice lcl = lowerPrice ucl = upperPrice pm = trendPrice;

run;

proc print data = dPredsPrice;

run;

/\* Plotting the predicted house price with confidence interval as a plot.\*/

title1 "Price Prediction as a function of Month (Index)";

proc sgplot data=dPredsPrice;

band x = Month upper = upperPrice lower = lowerPrice;

scatter x = Month y = Price;

series x = Month y = predictionPrice/lineattrs = (color=blue);

series x = Month y = trendPrice/lineattrs = (color=black);

run;