Maynooth University
National University
of Ireland Maynooth

An Overview of Residential Housing Prices in London

Submitted by: Group C

Anandita Ashwath, Kunal Goyal, Lakshya Gazaresen, Mudit Bhargava

Course Title: Case Studies in Data Science and Analytics (NCG612)

Submitted to: Prof. Martin Charlton

All members of the group have read and agreed to the final version of the report.
Signatures:

1. Anandita Ashwath - 19251017
2. Kunal Goyal - 19250776
3. Lakshya Gazaresen - 19251460
4. Mudit Bhargava - 19251439

Index

Abstract 3

Overview 3

Hedonic Pricing 3

Baseline regression & AIC 7

Location co-ordinates Map: 12

Variation by Borough 14

Standardized residual plots 17

Borough coefficient estimates: 19

Partial least Squares: 20

Spatial Modeling & GWR 21

Factors to consider & Evaluation Criteria: 21

Conclusion 22

Appendix 22

References 30

Abstract

The aim of this project is to examine Greater London’s residential housing price in depth. To produce both spatial and non-spatial models, consider the predictor variables of importance and visualize the outputs.

Overview

In this project, house costs are to be analysed and predicted based on various predictor variables which cover several real-life aspects of residential homes. For example, house prices in Barking & Dagenham are very likely to vary from those in Kensington & Chelsea, but statistical analysis or heuristics that include these data are significant using supervised machine learning algorithms which fits the model best. Supervised Learning is a method to use an algorithm to learn mapping function from the input to the output. Prediction of housing prices can be considered a regression problem, since we are concerned with predicting values that can fall within a continuous range of outputs. Regression is a supervised learning method where the output is continuous. We have also applied Random Forrest method to check which model fits the data better. Random forest is a supervised learning method which uses ensemble learning method for classification and regression, if the output variable is continuous. Through the machine learning methods, the relationship between price of houses and various other predictors such as floor area, type of house, number of bedrooms, local population density etc, can be explored. The analysis of house prices is also done spatially. Using the boundaries of the London Boroughs, it is feasible to add the Boroughs codes as dummy variables using spatial join. Using geographically weighted regression (GWR), local estimates of the intercept, variable coefficients and other regression diagnostics are mapped.

Hedonic Pricing

Hedonic pricing is a concept of price estimation based on hedonic price theory, which presumes that a property’s value is the sum of all its attributes. Using regression analysis, hedonic pricing can be applied for execution. Below mentioned equation shows the regression model for determining the house price:

$y = a.x1 + b.x2 + ... + n.xi$

Where, y is the predicted price, and x1, x2, xi are the attributes of a house. While a, b, ... n indicates the correlation coefficients of each variables in the determination of house prices.

On a macro-level, the hedonic price function f, defines the property price P as a function of three types of independent variables: structural, locational and socio-demographic. Generally, hedonic approaches, along with the coefficients that determine the significance of predictors, aim to reduce bias and non-linearity in order to produce a single solution for the intercept term. Spatial Autocorrelation arising from spatially dependent missing variables or temporal externalities and spatial heterogeneity cannot be taken into consideration in hedonic models. Spatial autocorrelation suggests that home in neighbourhood tend to be more similar. The real-estate modelling must use spatial autocorrelation to determine a house’s price at a given location based on nearby or similar houses.

Data Exploration

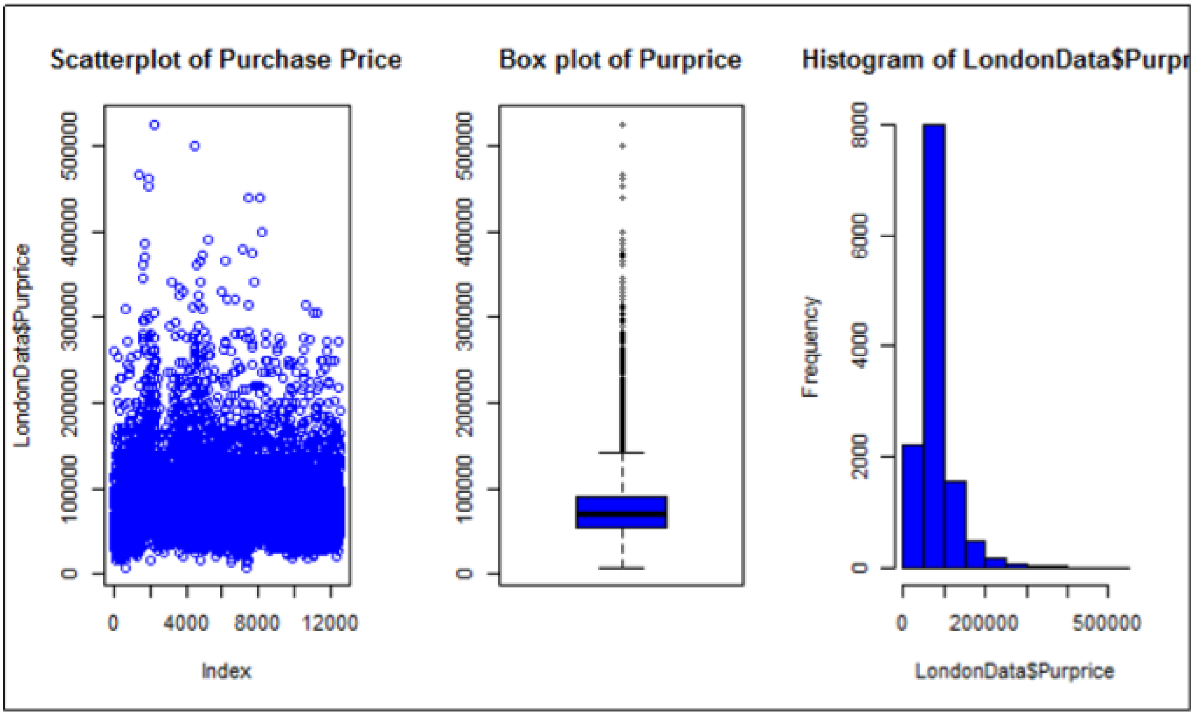
The dataset consists observations for 12,536 residences with 31 predictors, each variables with its description is listed below:

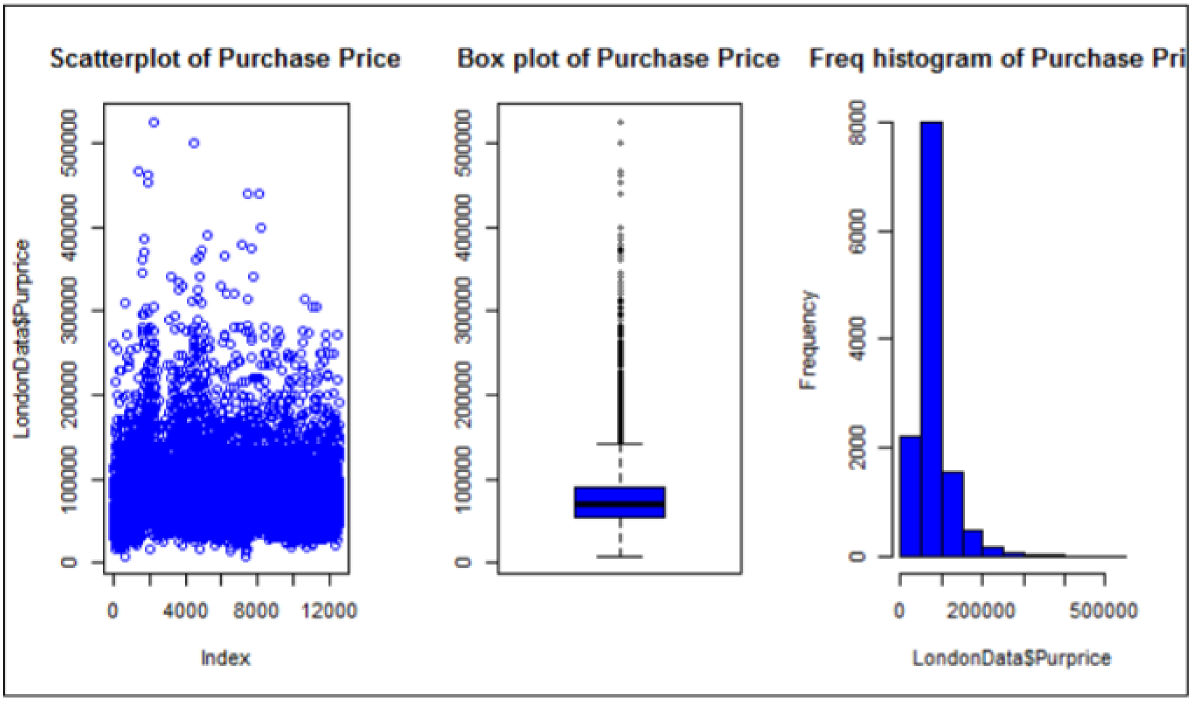
Variables	Description
Easting	Easting in m
Northing	Northing in m
Purprice	Purchase Price in GBP
BldIntWR	Built between 1918 and 1939
BldPostW	Built between 1945 and 1959
Bld60s	Built between 1960 and 1969
Bld70s	Built between 1970 and 1979
Bld80s	Built between 1980 and 1989
TypDetch	Detached property
TypSemiD	Semi-detached property
TypFlat	Flat or apartment
GarSingl	Single Garage
GarDoubl	Double Garage
Tenfree	Leasehold/Freehold indicator
CenHeat	Central heating
BathTwo	Two or more bathrooms
BedTwo	Two bedrooms
BedThree	Three bedrooms
BedFour	Four bedrooms
BedFive	Five bedrooms
NewPropD	New property
FlorArea	Floor area in square meters
NoCarHh	Proportion of households without a car
CarspP	Cars per person in neighborhood
ProfPct	Proportion of Households with Professional Head UnskPct Proportion of Households with Unskilled head
RetiPct	Proportion of residents retired
Saleunem	Not known
Unemploy	Unemployed workers
PopnDnsy	Local population density

We run some basic stats for our data set, and come to the following conclusions based on summary stats & R plots:

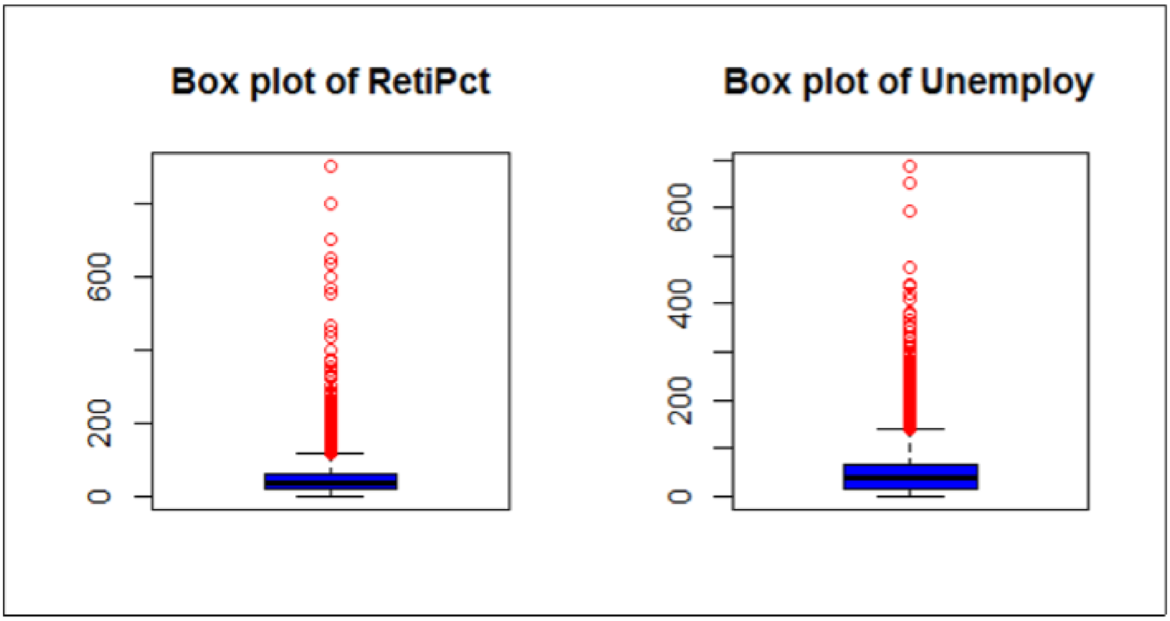
1. Few integer variables are in fact categorical, and some data manipulation function shall be written to represent them as a matrix.
2. Dataset reduction to eliminate outliers may be a prudent approach.

3. There is one value of purprice > 600000, dropping it to avoid outliers.
4. Two columns, RetiPct i.e. Proportion of residents retired and Unemploy i.e Unemployed workers have unusual values.

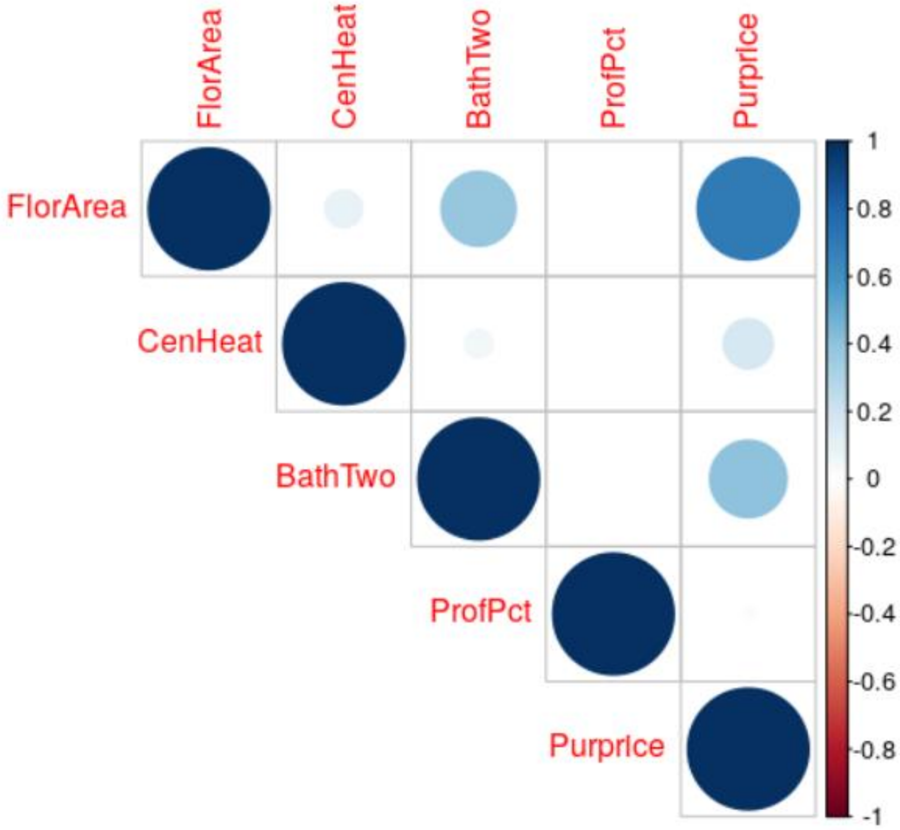




“RetiPct” and “Unempley” are being analysed with plots in the following figure. They are found to be logical outliers, in order to be unbiased, we have decided to omit these variables from the dataset.



Plotting a correlation matrix of continuous input predictors gives us an insight into the data, and that floor price is related with purchase price by a degree of .71



We use a stepwise hedonic procedure for data exploration using linear regression by performing the following

1. Fit the regression of Purchase price with an increasing order of input variables sequentially.
2. Compare the best models with least AIC values.

3. Keep on adding extra variables to the existing best model in order to obtain minimum AICs.
4. By sorting the models based on minimum AIC, generate a combined model with all predictors.

Baseline regression & AIC

The Akaike Information Criterion (AIC) provides a method for assessing the quality of your model through comparison of related models. When comparing the analysis of individual independent variables, standardized coefficients are preferred rather than the unstandardized coefficient as it does not take into account the differences in unit measurements of a variable. The determinants cause 56.41% of the variation in house price in London. The independent variable that contributes in determining the house price is the floor area in total. A house having more than two bathrooms and the age of the property have a positive effect on the sale price. Topnotch houses with higher purchase price would need to have a type Detch design, Double Garage, and at least four bedrooms to reside in. A house with no central heating system might have a negative coefficient which would reduce the value of the property.

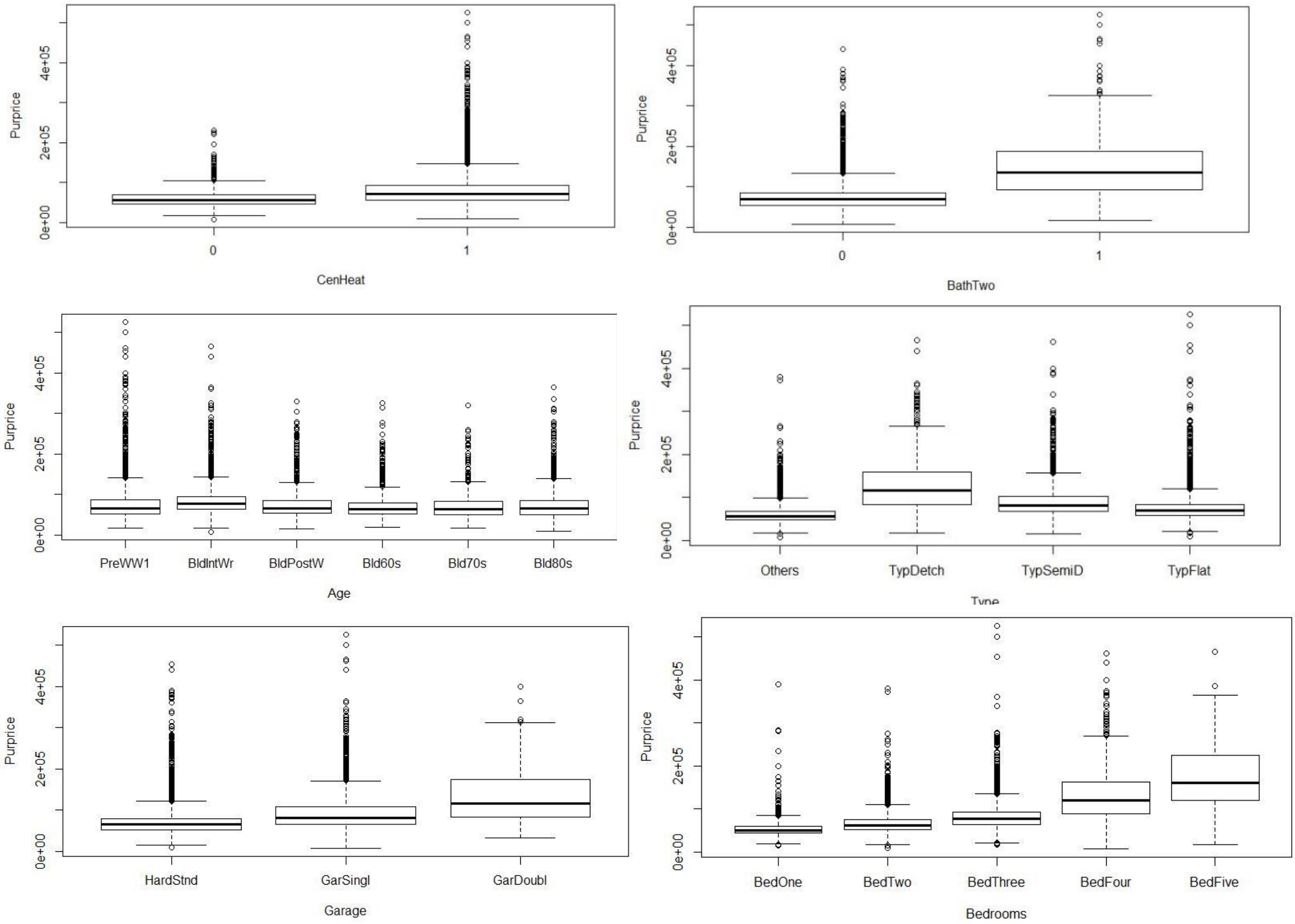
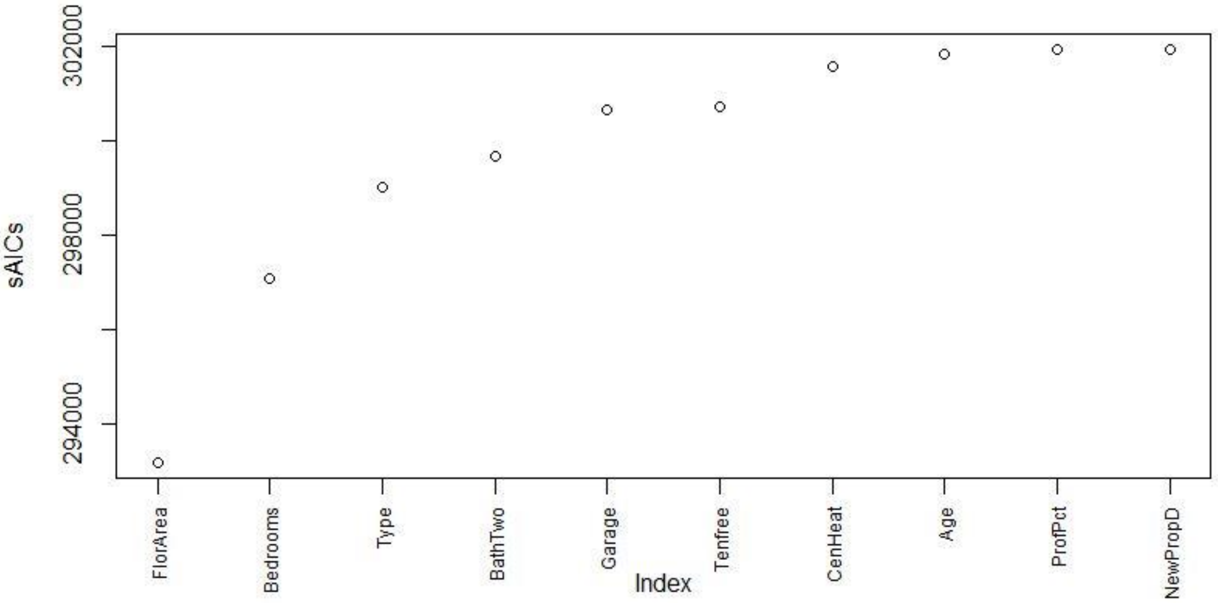


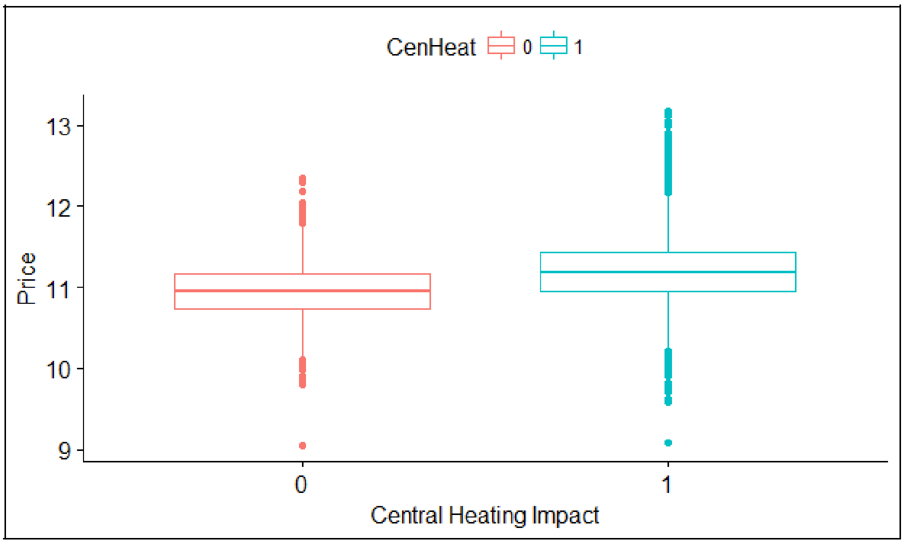
Fig: Boxplots of individual variables regressed on Purchase Price



The above plot shows the plot of sAICs vs predictors, predictor having the lowest AIC value is preferred. We can see from the above plot that Floor Area (FlorArea) has the minimum AIC value. Next, bedrooms, House type and rest of the predictors follow with an increasing AIC value. One predictor model can be obtained by examining the AIC's of all the models. AICs of each predictor model are compared and the one with the least AIC is considered, which in the below table is Floor Area.

Variable	Sorted Cumulative AIC
FlorArea	293198.7
Bedrooms	297086.6
Type	299031.9
BathTwo	299667.9
Garage	300656.4
Tenfree	300712.3
CenHeat	301562.2
Age	301833.0
ProfPct	301929.4
NewPropD	301930.4

Table: AIC values of each predictor variables



The impact of central heating on the price on the house is done through Welch Two Sample T-test. This test is done with the assumption that the variance is not equal among the variables and gives a better performance. The null hypotheses are done based on the mean of the variables with the null hypotheses is taken as the means are equal and its alternate is that the means are not equal or the difference in the mean is not zero. The null hypotheses is rejected on the basis on p-values and we have considered a confidence interval of 95%.

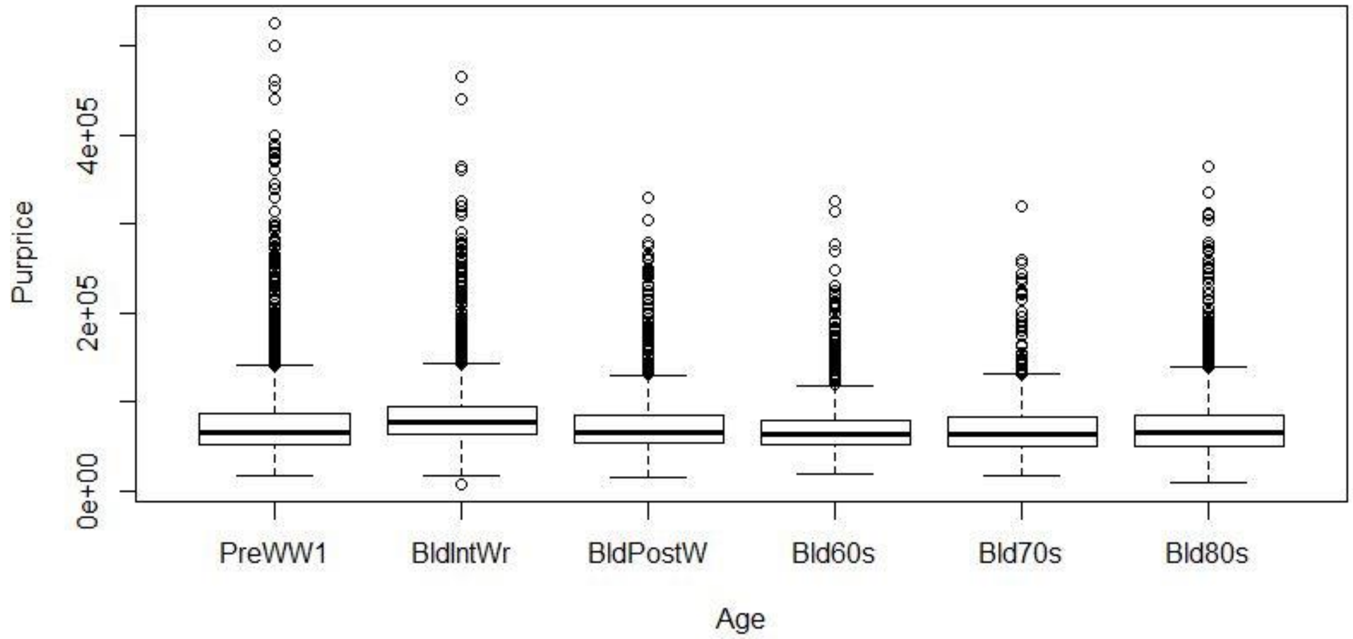
```
Welch Two Sample t-test


Data:  Purprice by CenHeat

t = -27.016, df = 2144.4, p-value < 2.2e-16

Alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

 [-0.2889442, -0.2498352]

sample estimates:

mean in group 0 mean in group 1

      10.95543        11.22481
```

Since the p-value is lessn than 0.05, Null Hypothesis is rejected.

The impact of the age of house or the year in which the house is built on the price of the house is done through pairwise t-test. The pairwise t-test consists of calculating multiple t-test between all possible combination of groups.

```
>>pairwise.t.test(MyData$Purprice, MyData$Age, p.adjust = "bonferroni")

Pairwise comparisons using t tests with pooled SD

data: MyData$Purprice and MyData$Age
```

	PreWW1	BldIntWr	BldPostW	Bld60s	Bld70s
BldIntWr	0.000000000252	-	-	-	-
BldPostW	1.0000	0.0014	-	-	-
Bld60s	0.0203	0.000000000053	0.0623	-	-
Bld70s	0.0119	0.000000000095	0.0361	1.0000	-
Bld80s	1.0000	0.000000030459	1.0000	0.8733	0.5219



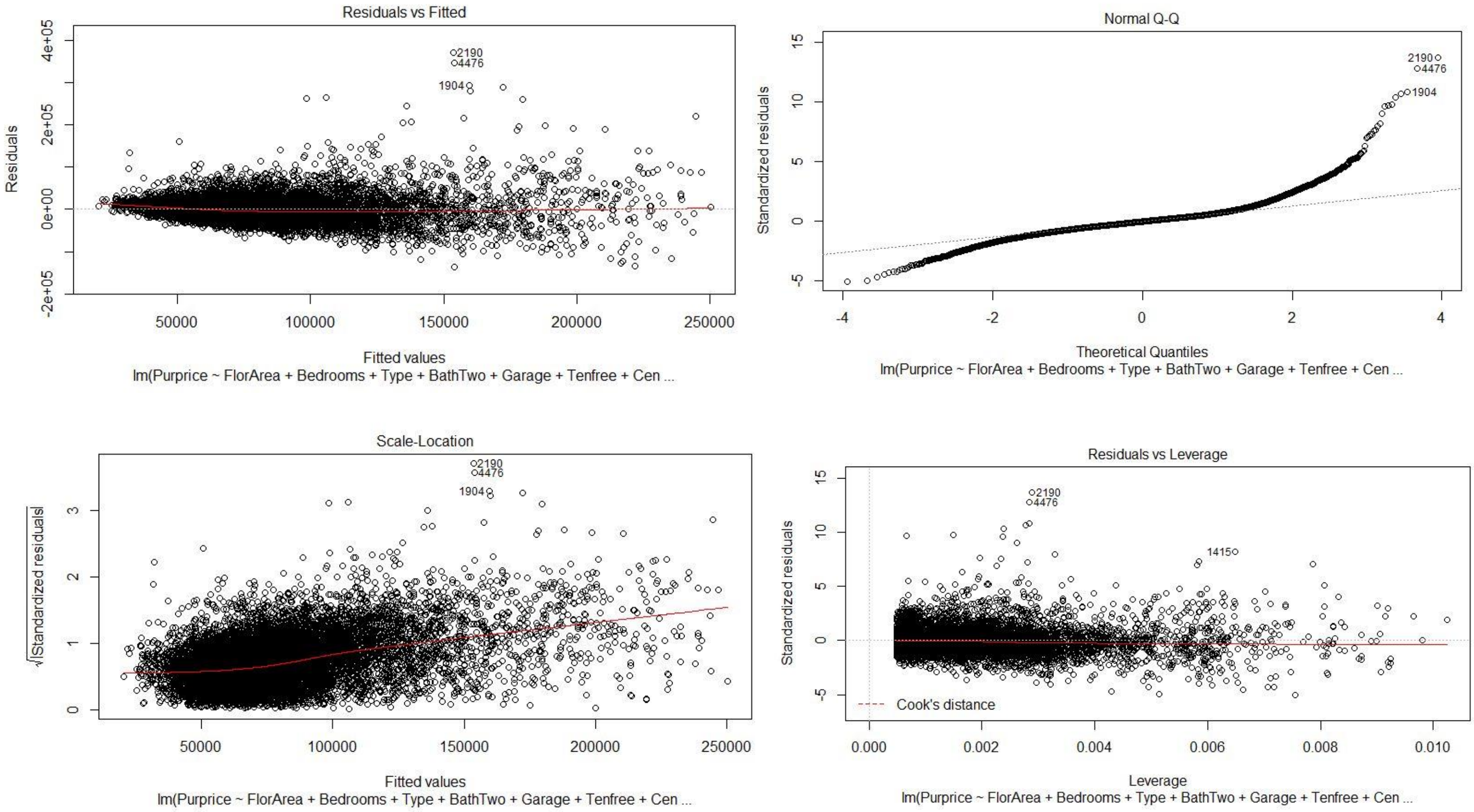
PreWW1(Pre-World War I) and BldIntwr (Built between 1918 and 1939) aged houses and purchase price of the houses is slightly different from the houses built on latter years.

We fit the hedonic generic equation as:

model.9v <- lm(Purprice~FlorArea+Bedrooms+Type+BathTwo+Garage+Tenfree+CenHeat+Age+ProfPct,data= MyDat) summary(model.9v) # adj r^2 ~ .5

All the independent variables have a high t-statistic and low p-values, which proves that the variables are significant with a 99% confidence level. Moreover, the calculated f-value of 854.8 shows that the overall equation is statistically sound. We can conclude from the above and say Null hypothesis can be rejected.

Plot(model.9v)



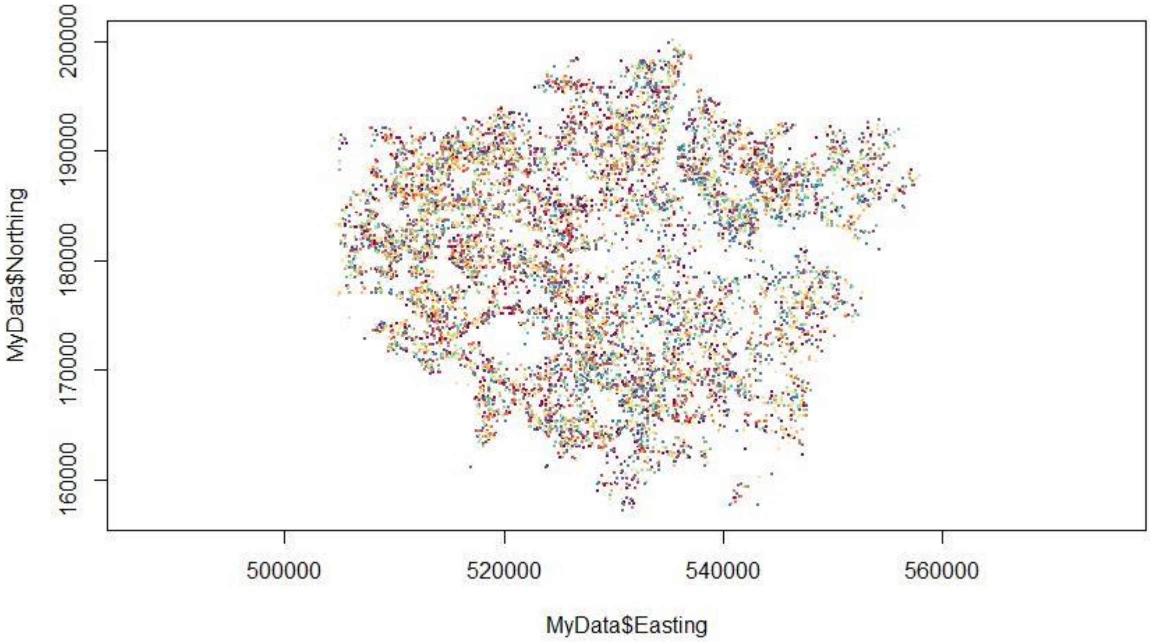
The Residuals versus Fitted Values plot appears to have the data points randomly scattered across the zero line. The Normal QQ plot appears to be distorted at right upper tail which suggests to take log of the output variable. Calculating the MSPE on our dataset with this model gives us an estimation of our accuracy, which is comparatively higher than the standalone models. To check which machine learning model fits the dataset better we applied Linear Regression and Random Forest Regression algorithms. The comparison between models is done by calculating the mean square error of each models. The data is divided into training and test datasets where we have taken 60% of the data as training set and remaining as test set. The machine learning algorithm is first run on the training set and the model is tested on the test dataset and checked if the algorithm is providing proper results. We have taken log of the output variable, "Purprice" as there is a positive skew in the residual plot. When fitting the linear regression, the mean square error is coming around 0.08. When fitting the random forest model, the mean square error for the test dataset is around 0.079 and for the training dataset the error much less around 0.04. From these results it can be stated that random forest fits the house price dataset better than linear regression.

```
Analysis of Variance Table

Response: log(Purprice)
             Df  Sum Sq Mean Sq    F value Pr(>F)    
FlorArea      1 1068.54 1068.54 13021.8363 <2e-16 ***
Bedrooms      4    6.87    1.72    20.9338 <2e-16 ***
Type          3   43.22   14.41   175.5657 <2e-16 ***
BathTwo       1   13.00   13.00   158.3711 <2e-16 ***
Garage        2    9.66    4.83    58.8501 <2e-16 ***
Tenfree       1    7.29    7.29    88.8924 <2e-16 ***
CenHeat       1   35.83   35.83   436.5906 <2e-16 ***
Age           5   20.65    4.13    50.3304 <2e-16 ***
ProfPct       1    0.19    0.19     2.2677 0.1321    
Residuals 12515 1026.95    0.08                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Finding the mean square error of the test dataset using random forest.

```
> mean((pred1 - log(test_data$Purprice))^2)
[1] 0.07919715
```

Location co-ordinates Map:



Simple Map of data spread with price

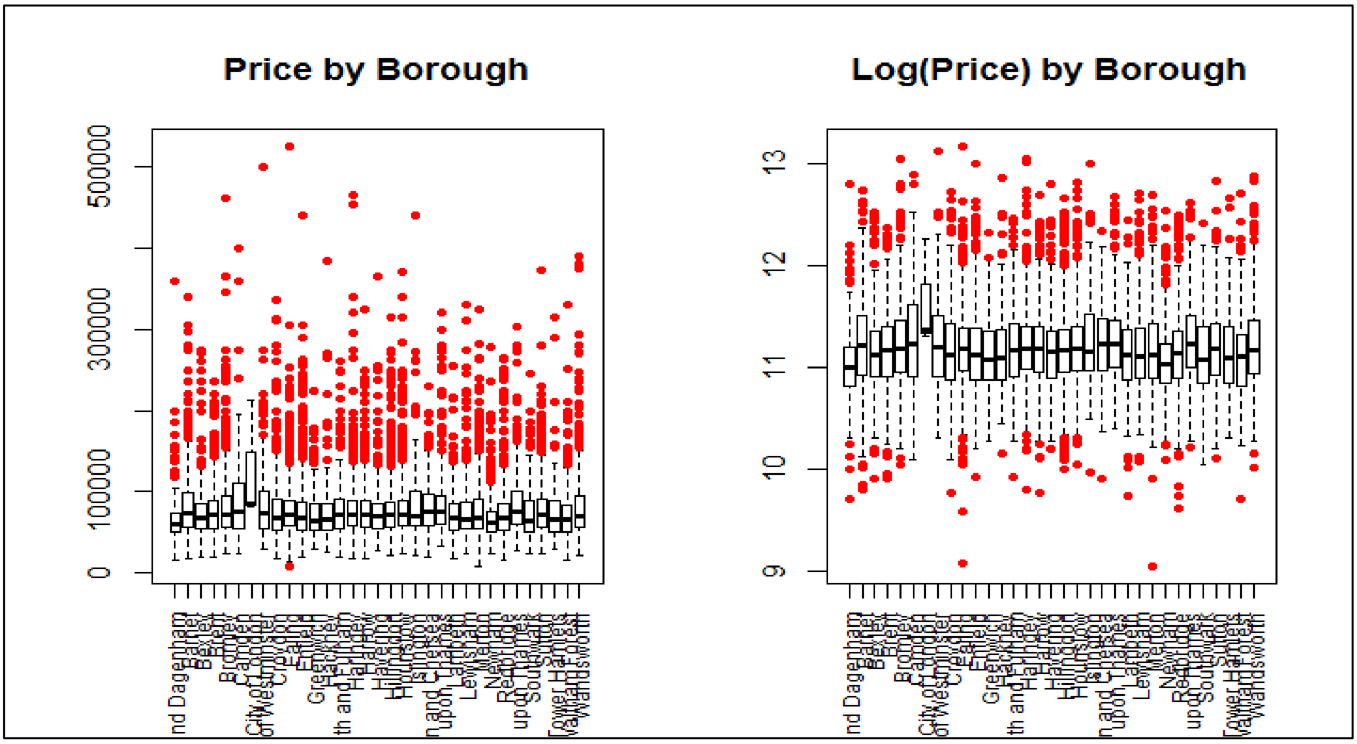
On spatial level, we can search for more extensive level groups/designs on the spot-based lodging prices. The property price is partitioned into 10 quantiles and if each quantile is allotted a shading, and afterward if the price is plotted against the area, the accompanying plot is acquired. It is beyond the realm of imagination to expect to draw a knowledge from the plot because spatial part isn't the main measures for property valuing, there are different factors which influence the price.

```
summary(m.tr1) # lower prices as we move east, slightly lower as we move towards south

        Call:
        lm(formula = Purprice ~ x + y, data = MyData)

        Residuals:
                  Min1QMedian3QMax
             -72863 -24957-100189714443417

        Coefficients:
                             Estimate Std. Error t valuePr(>|t|)
        (Intercept) 165151.00 17668.70 9.347 < 0.0000000000000002 ***
             X            -135.10        30.42      -4.441            0.00000903 ***
             Y            -77.06         40.45      -1.905                0.0568

        Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

        Residual standard error: 41090 on 12532 degrees of freedom



        Multiple R-squared:  0.001854,           Adjusted R-squared:  0.001694
        F-statistic: 11.64 on 2 and 12532 DF, p-value: 0.000008937

        > summary(m.tr2) # lower AIC # higher price as we move west Call:
        lm(formula = Purprice ~ x + y + I(x^2) + I(y^2) + I(x * y), data = MyData)

        Residuals:
          Min           1Q       Median          3Q      Max
         -7392        -24782      -9828          9862   444261

        Coefficients:
                        Estimate              Std. Error       t value          Pr(>|t|)
        (Intercept) -3153110.597        874075.711       -3.607           0.000311 ***
        x                12253.587            2793.148         4.387            0.0000116 ***
        y                  352.539            2915.762         0.121            0.903766
        I(x^2)          -10.739               2.555             -4.203           0.0000266 ***
        I(y^2)             7.372               4.717             1.563            0.118080
        I(x * y)        -5.727                4.323             -1.325           0.185350
        ---
        Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 41050 on 12529 degrees of freedom Multiple
R-squared:  0.004172,                    Adjusted R-squared:  0.003774
F-statistic: 10.5 on 5 and 12529 DF, p-value: 0.0000000004507
```

Variation by Borough

To check the variety of property price as for district, the shape record of the London precincts ought to be stacked and afterward the property price ought to be plotted. To Load the shapefile, readOGR function can be used. Subsequent to perusing the shapefile, the focuses ought to be anticipated on the guide this should be possible by changing over the focuses into same organize reference framework as the shapefile. An over capacity can be utilized to recognize the information focuses in every district and afterward by getting the names of every ward a crate plot can be drawn which gives the mean price in every precinct. The plot got is as underneath. From the plot it is obvious that the middle of all the obtains are not same and each get has high inconstancy/standard deviation. To decrease the standard deviation, log change can be applied on the property price.

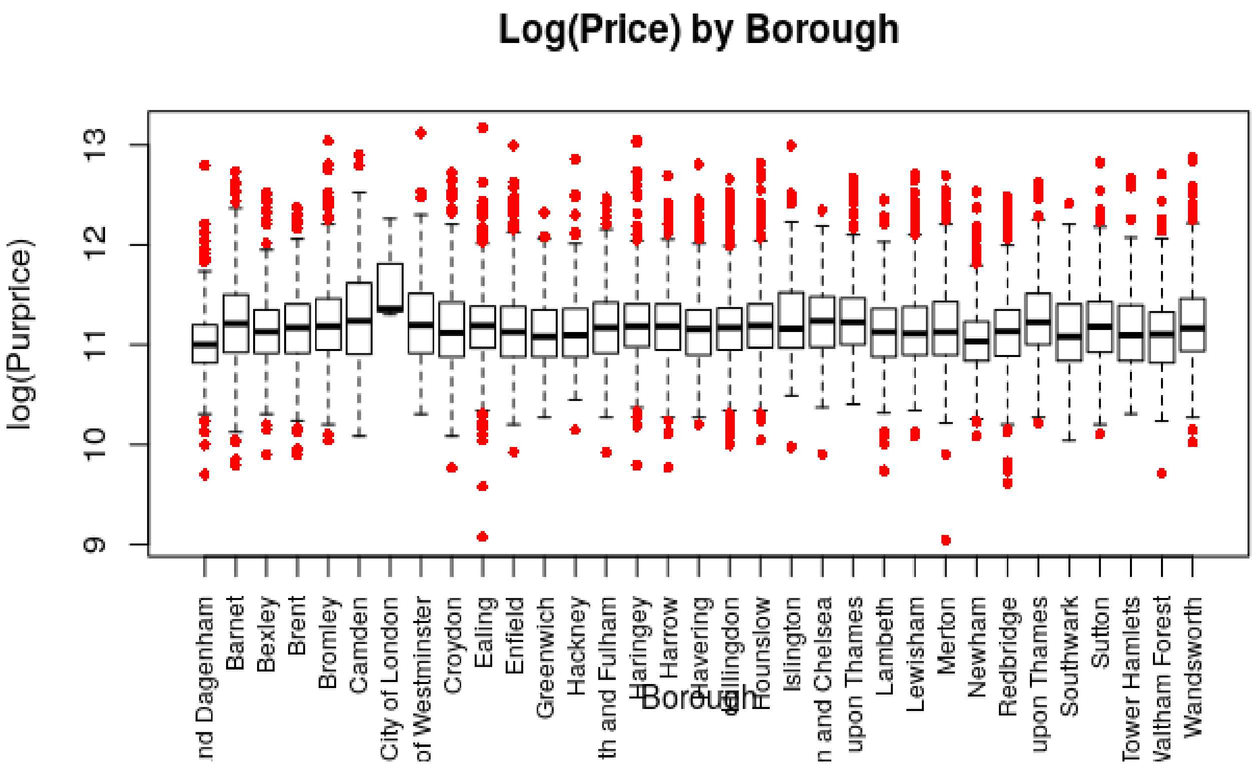


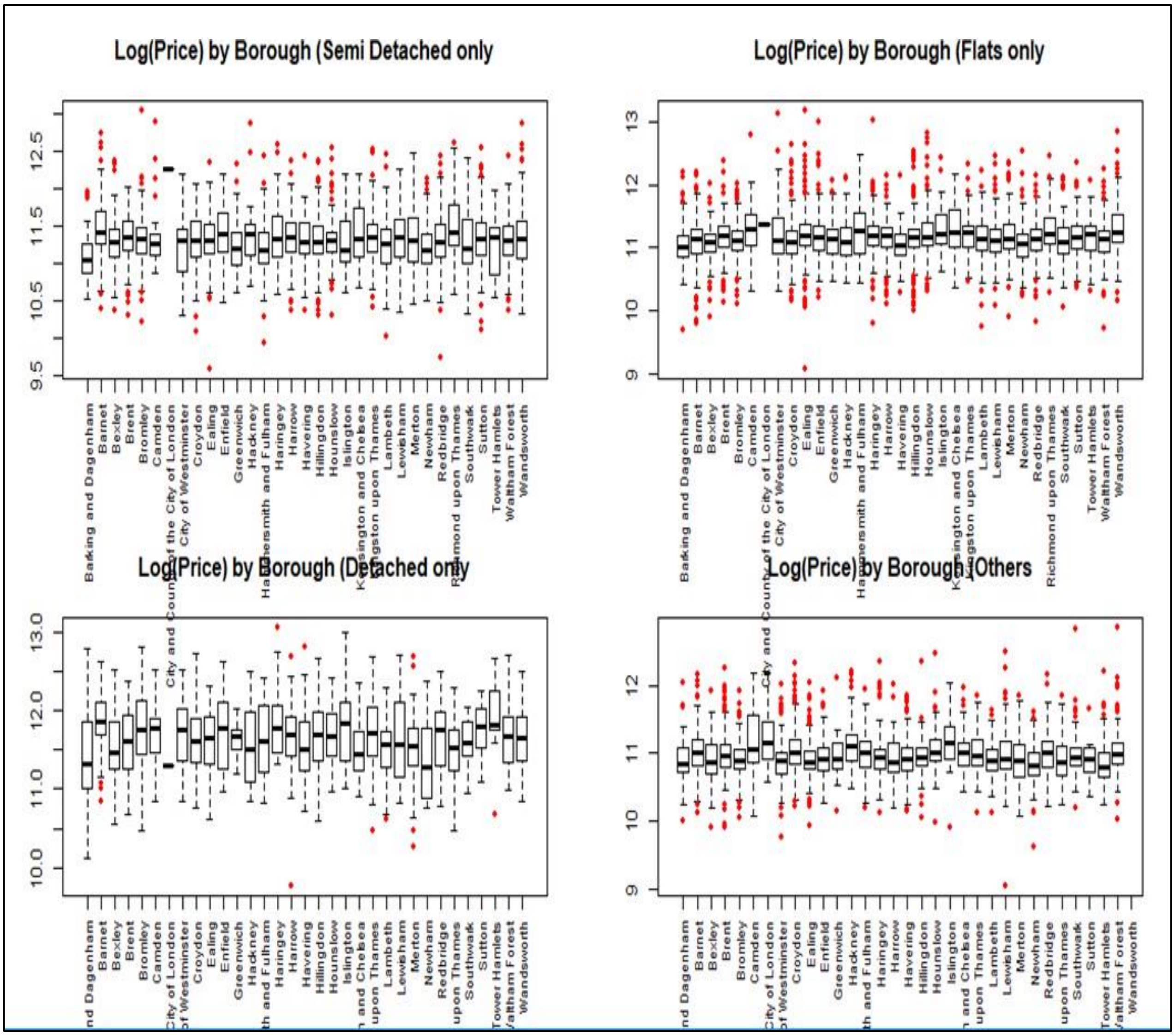
Between the two plots, the log transformed box plot is a superior portrayal to decipher. There are many anomalies in information. The city of London is strikingly price. Also, as indicated by the container plots, "Southwark" and Dagenham is the least expensive spot to purchase a house, anyway there are different areas, for example, Ealing and Merton that show exceptions.

“Hillingdon” has the largest number of houses. “City and County of the City of London” has the least number of houses.

```
thsd <- TukeyHSD(aov(log(Purprice)~Borough,data=MyData[MyData$Type=="TypFla t",]), conf.level = 0.95)$Borough
```

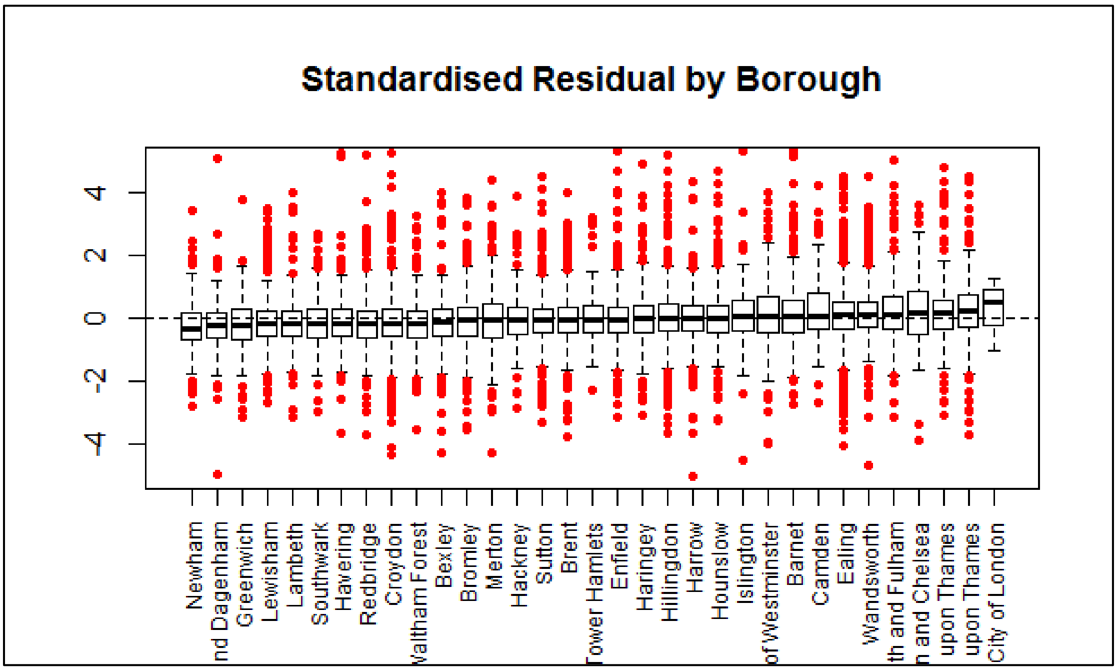
Also, applying Tukeys HSD on these Boroughs, for property of Type Flat it turns out that 32 out of 528 pairs have significant difference and does not have significant difference in their means otherwise.





The log of the house price with respect to Boroughs has been plotted. Areas closer to the downtown area have a high mean/middle, in the semidetached space, and variety in homes from focus is lesser suggesting area reliance. City of London has the most significant expense for semi-disconnected properties. City of Westminster, Chelsea and Tower Hamlets have the most noteworthy variety for semi-confined properties.

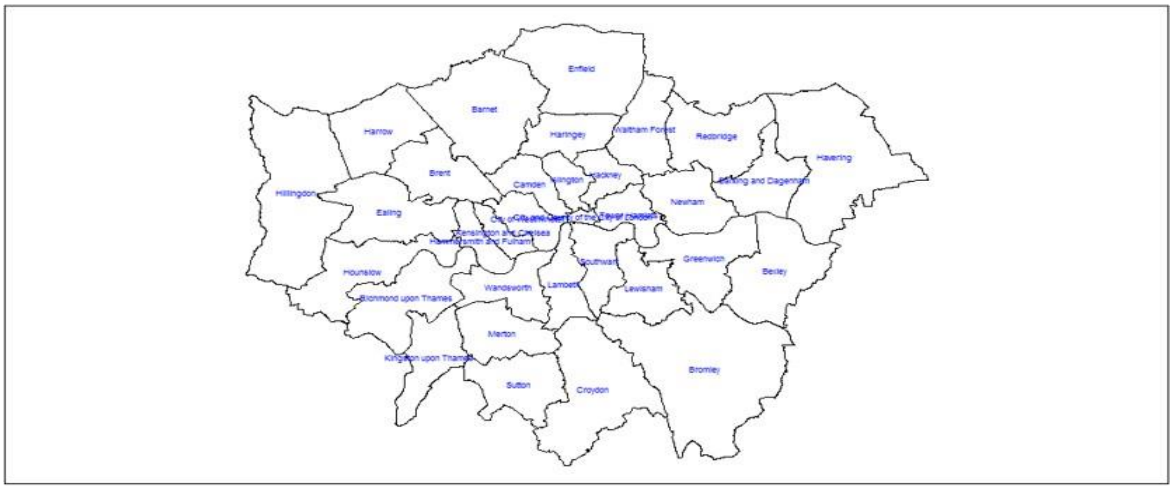
Standardized residual plots



If the residuals got by the model are plotted by every district, the median of residuals are distinctive for every borough. From this it very well may be construed that the difference of price is available in the ward which isn't clarified by current model. In this way, remembering the wards for the model may give a superior R-squared worth. In the plot, negative residuals demonstrate under forecast though positive residuals show over expectation. Presently, arranging with median values.

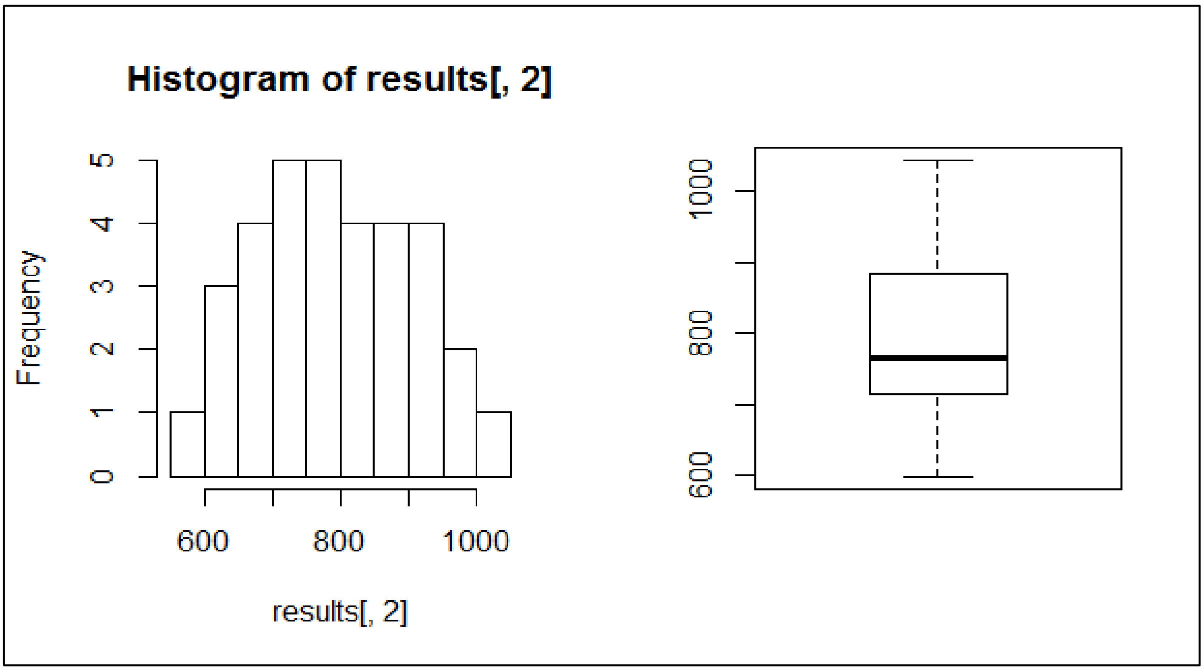
London Borough Models:

A basic map of London with its boroughs outlined can be seen below:



Applying linear regression including Boroughs, the intercepts based on the boroughs can be seen in the table below:

Boroughs	Intercept	FlorArea
Camden	4193.591	912.5144
Tower Hamlets	-22055.905	1042.907
Islington	-9756.782	976.7416
Hackney	-6427.27	888.3199
Haringey	-10165.2	941.118
Newham	8392.221	639.826
Barking and Dagenham	2833.098	714.4698
City and County of the City of London	-8934.581	926.4475
Kingston upon Thames	-7486.608	970.7183
Croydon	2511.36	765.5992
Bromley	-1299.96	838.6432
Hounslow	2331.035	822.3698
Ealing	15196.698	691.0613
Havering	-8434.152	875.8949
Hillingdon	6441.099	774.5079
Harrow	9438.779	737.1701
Brent	20595.103	598.9557
Barnet	-1450.793	885.0956
Lambeth	10948.837	666.9039
Southwark	10211.971	689.2389
Lewisham	-6768.227	860.6584
Greenwich	16655.867	600.0343
Bexley	6120.194	729.3274
Enfield	-1612.182	844.1284
Waltham Forest	9317.256	669.1548
Redbridge	1371.89	757.0958
Sutton	10038.245	730.6204
Richmond upon Thames	13743.097	752.8587
Merton	7064.287	753.2699
Wandsworth	-3590.767	919.3393
Hammersmith and Fulham	14952.08	736.6411
Kensington and Chelsea	24302.279	637.5807
City of Westminster	8260.768	830.7942

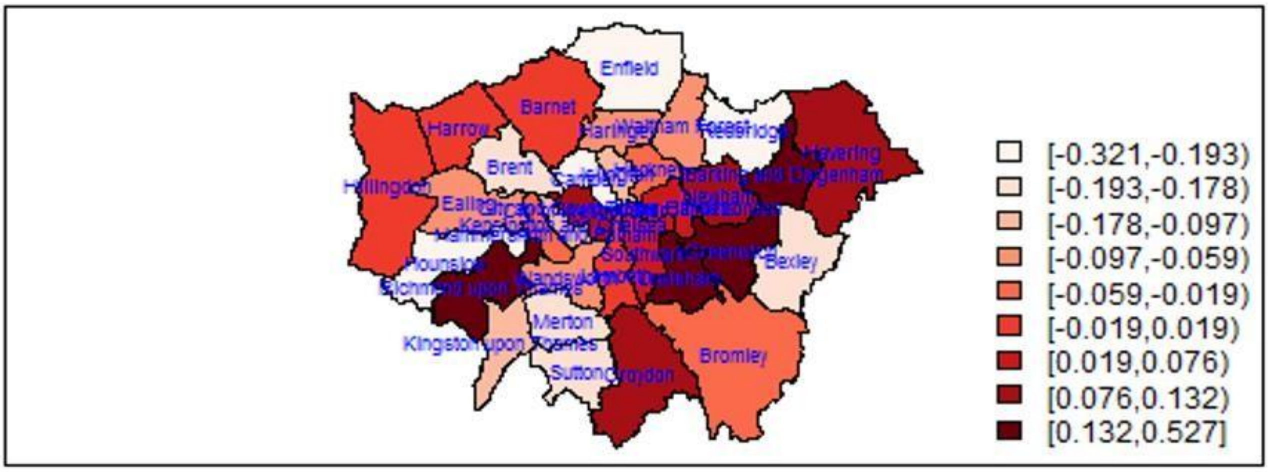


On X-axis is the co-efficient of floor region and on y-axis is the frequency. Thus, it is comprehended that the number occurrences of co-effective for every borough. A large portion of the borough has the co-efficient estimate somewhere in the range of 650 and 950. Most frequent estimate is between 700-800. The median of the co-efficient can be comprehended by the boxplot. The median of the estimate is near to 750.

Borough coefficient estimates:

Plot Borough medians using diff. residuals:

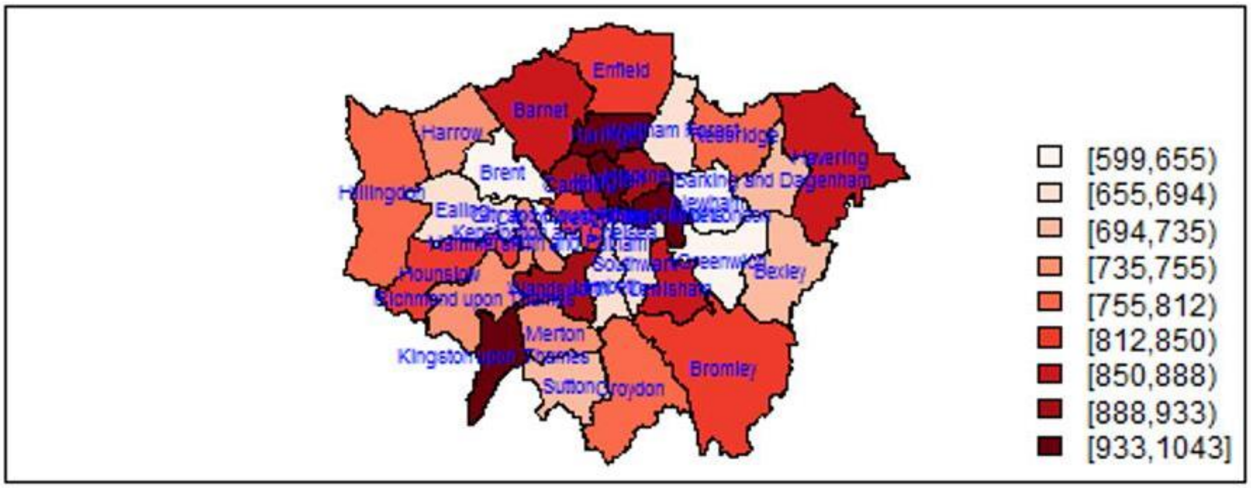
Choropleth Maps can be drawn showing the co-efficient estimate for each borrow and the standard residual ranges of each borrows.

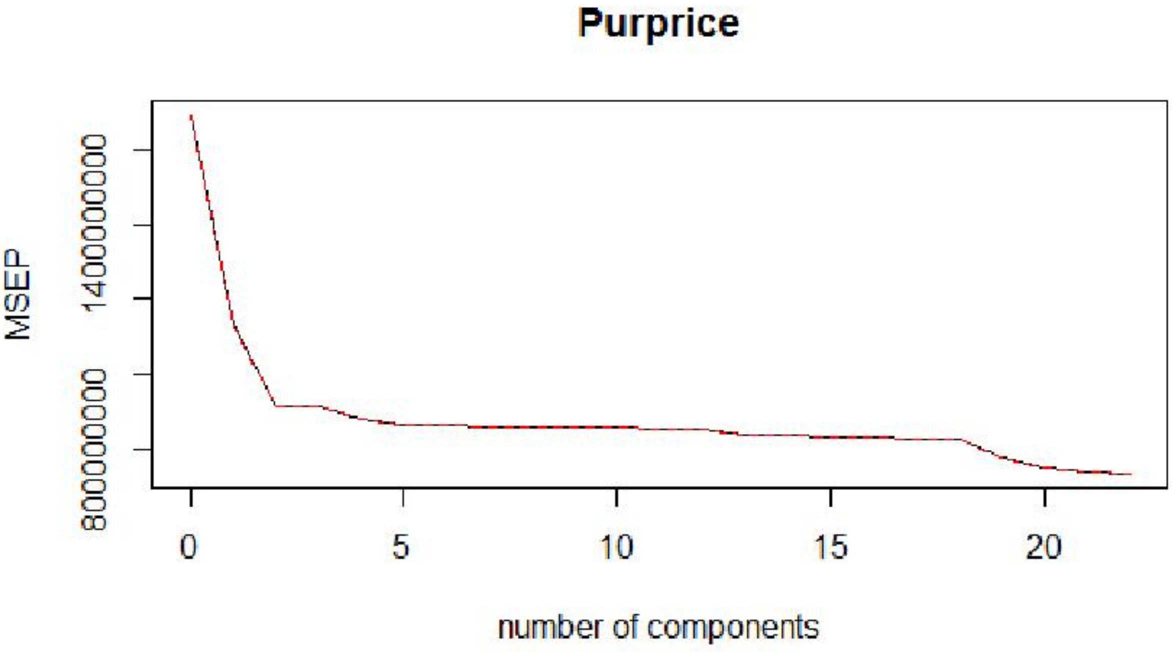


Partial least Squares:

Partial least squares regression (PLS regression) is a statistical method that is like principal components regression; instead of finding hyperplanes of maximum variance between the response and independent variables, it finds a linear regression model by projecting the predicted variables and the observable variables to a new space.

```
library(pls
) MyData1
<-
data.frame(LondonData[,c(2:4,15:17,22,23,26)],Age,Type,Garage,Bedrooms)
pcr.fit <- pcr(Purprice ~., data=MyData1, scale= TRUE, validation = "CV")
summary(pcr.fit)
validationplot(pcr.fit, val.type = "MSEP")
```





Higher number of predictors added (22) improved accuracy explaining 58% of variability in model. Elbow bend at 2,18 predictors.

Spatial Modeling & GWR

Theory

Geographically weighted regression (GWR) is an exploratory technique mainly intended to indicate where non-stationarity is taking place on the map, that is where locally weighted regression coefficients move away from their global values. Its basis is the concern that the fitted coefficient values of a global model, fitted to all the data, may not represent detailed local variations in the data adequately – in this it follows other local regression implementations. It differs, however, in not looking for local variation in ‘data’ space, but by moving a weighted window over the data, estimating one set of coefficient values at every chosen ‘fit’ point. The fit points are very often the points at which observations were made, but do not have to be. If the local coefficients vary in space, it can be taken as an indication of non-stationarity

Factors to consider & Evaluation Criteria:

The basic GWR model assumes the same degree of spatial smoothness for each coefficient, which may not hold true in all contexts. GWR therefore over fits the data and produces a bias. Hence the basic GWR has undergone the following significant revisions:

Generally, GWR models define distances as straight line or Euclidean, while more recent modifications of the distance function adopt non-Euclidean distance metrics to improve the model fit. Second, traditional GWR models use a fixed bandwidth for all variables to estimate the spatial relationship between variables while a revised GWR can use a flexible bandwidth to estimate spatially varying relationships at various geographical scales within one model. There must be focus on diagnostics to check the model fit such as cross-validation (CV) score to derive an optimal kernel bandwidth for GWR regression to reduce model bias. An important measure is the Akaike Information Criterion (AIC) that is traditionally used to account for model parsimony dealing with the trade-off between prediction accuracy and complexity. In GWR, a corrected version of the AIC is used that accounts for sample size and entails fitting bandwidths with different penalty functions. Finally, not all variables in GWR models exhibit non- stationarity in all contexts; hence if all of the independent variables in the GWR exert a spatial influence on the dependent variable can lead to biased estimations. This testing may in some independent variable coefficients being held constant, which are considered global parameters, while some others spatially vary, denoted as local parameters.

Conclusion

Traditionally housing price evaluations have been hedonic models by economists, and it is validated as a right approach in our statistical analysis. Floor area is the most significant, followed by Bedrooms count, structural type and presence of centralized heating. Number of Bathrooms/Bedrooms/Garage are all co-related. When comparing different machine learning algorithms, it was found that random forest fits the dataset better compared to linear regression using OLS method as the mean square error provided by random forest is slightly lesser than the mean square error of linear regression model. From location based spatial visualizations, we can infer the scaled price increase with decreasing distance to center. Distance / Bandwidth estimations in GWR, are parameterized and can be effectively compared using cross validation. Proportion of workers retired and unemployed workers in the neighborhood are two factors which do not have any impact on the price of housing. So, it can be concluded that all the predictors including the spatial component are important in predicting the price of the property.

Appendix

```
# Read the data
LondonData <- read.csv("DataScienceProj.csv",stringsAsFactors=FALSE)
print(dim(LondonData))
head(LondonData)
str(LondonData)
summary(LondonData)
boxplot(LondonData$Purprice)
LondonData <- LondonData[LondonData$Purprice < 600000,]
boxplot(LondonData$Purprice)

### suspicious values for RetiPct, Unemploy

###
### large houses costs more
###
plot(LondonData[,c("FlorArea","Purprice")],pch=16,cex=0.5)
lines(lowess(LondonData[,c("FlorArea","Purprice")]),col="red")

###
### Convert dummies to factors
###  - more convenient for modelling  -
```

```
###
Dummy2Factor <- function(mat,lev1="Level1") {
    mat <- as.matrix(mat)
    factor((mat %*% (1:ncol(mat))) + 1,
        labels = c(lev1, colnames(mat)))
}

Age      <- Dummy2Factor(LondonData[,5:9],"PreWW1")
Type     <- Dummy2Factor(LondonData[,10:12],"Others")
Garage   <- Dummy2Factor(LondonData[,13:14],"HardStnd")
Bedrooms <- Dummy2Factor(LondonData[,18:21],"BedOne")

MyData <- data.frame(LondonData[,c(2:4,15:17,22,23,26)],Age,Type,Garage,Bedrooms)
summary(MyData)

### LondonData is the original
### MyData has the factor-ed versions

### explore
boxplot(Purprice~CenHeat,data=MyData)
boxplot(Purprice~BathTwo,data=MyData)
boxplot(Purprice~Age,data=MyData)
boxplot(Purprice~Type,data=MyData)
boxplot(Purprice~Garage,data=MyData)
boxplot(Purprice~Bedrooms,data=MyData)


### plot simple map
###
library(classInt)
library(RColorBrewer)

nClass = 10
Palette <- rev(brewer.pal(nClass,"Spectral"))
Classes <- classIntervals(MyData$Purprice,nClass,"quantile")
Colours <- findColours(Classes,Palette)
plot(MyData$Easting,MyData$Northing,pch=16,cex=0.25,col=Colours,asp=1)


###
### Geography - look at trends with linear and quadratic trend surfaces
###
x <- MyData$Easting/1000
y <- MyData$Northing/1000
m.tr1 <- lm(Purprice~x+y,data=MyData)
AIC(m.tr1)
m.tr2 <- lm(Purprice~x+y+I(x^2)+I(y^2)+I(x*y),data=MyData)
AIC(m.tr2)
summary(m.tr1) # lower prices as we move east, slightly lower as w move south
summary(m.tr2) # lower AIC # higher price as we move west
stepAIC(m.tr2)
```

```
###
### Explore variation by borough  - first load the data
###
library(rgdal)
library(rgeos)
LB <- readOGR(dsn=".",layer="LondonBoroughs",stringsAsFactors=FALSE)  # Boroughs
LH <- SpatialPointsDataFrame(MyData[,1:2],MyData)                    # Houses
proj4string(LH) <- CRS(proj4string(LB))                              # copy CRS
plot(LB)
points(LH,pch=16,cex=0.5)
box()

###
### Add Brough names to data  - explore by type and borough - we'll need to do an overlay
###
LHLB <- over(LH,LB)   # spatial join: points first, then polygons
dim(LHLB)
head(LHLB)           # data frame has LB attributes in LH order
MyData$Borough <- gsub(" London Boro","",LHLB$NAME)  # get the borough name

boxplot(Purprice~Borough,data=MyData)

Boroughs <- names(table(MyData$Borough))
NB <- length(Boroughs)
boxplot(log(Purprice)~Borough,data=MyData,outpch=16,outcol="red",outcex=0.75,xaxt="n")
axis(1,labels=Boroughs,at=1:NB,cex.axis=0.75,las=2)
title("Log(Price) by Borough")

boxplot(log(Purprice)~Borough,data=MyData[MyData$Type=="TypSemiD",],outpch=16,outcol="red",outcex=0.75
,xaxt="n")
axis(1,labels=Boroughs,at=1:NB,cex.axis=0.75,las=2)
title("Log(Price) by Borough (Semi Detached only")

boxplot(log(Purprice)~Borough,data=MyData[MyData$Type=="TypFlat",],outpch=16,outcol="red",outcex=0.75,xa
xt="n")
axis(1,labels=Boroughs,at=1:NB,cex.axis=0.75,las=2)
title("Log(Price) by Borough (Flats only")

###
### Ordered boxplot
###

b.order <- rank(tapply(MyData$Purprice+runif(nrow(MyData)),MyData$Borough,median))

boxplot(Purprice~Borough,data=MyData,outpch=16,outcol="red",outcex=0.75,xaxt="n",at=b.order,ylim=c(0,50000
0))
axis(1,labels=Boroughs,at=b.order,cex.axis=0.75,las=2)
title("Price by Borough")


boxplot(log(Purprice)~Borough,data=MyData,outpch=16,outcol="red",outcex=0.75,xaxt="n",at=b.order)
axis(1,labels=Boroughs,at=b.order,cex.axis=0.75,las=2)
```

```
title("Log(Price) by Borough")


###
### standardsed residuals -s there a apttern
###
library(MASS)
MyData$stdres.9v <- stdres(model.9v)
boxplot(stdres.9v~Borough,data=MyData,outpch=16,outcol="red",outcex=0.75,xaxt="n")
axis(1,labels=Boroughs,at=1:NB,cex.axis=0.75,las=2)
title("Standardised Residual by Borough")

boxplot(stdres.9v~Borough,data=MyData,outpch=16,outcol="red",outcex=0.75,xaxt="n",ylim=c(-5,5))
axis(1,labels=Boroughs,at=1:NB,cex.axis=0.75,las=2)
title("Standardised Residual by Borough")
abline(h=0,lty=2)

###
### y-yhat negative : overproediction
### y-yhat positive : underprediction
###
b.order.9v <- rank(tapply(MyData$stdres.9v+runif(nrow(MyData))*0.0001,MyData$Borough,median))
boxplot(stdres.9v~Borough,data=MyData,outpch=16,outcol="red",outcex=0.75,xaxt="n",at=b.order.9v,ylim=c(-
5,5))
axis(1,labels=Boroughs,at=b.order.9v,cex.axis=0.75,las=2)
title("Standardised Residual by Borough")
abline(h=0,lty=2)



###
### Map of Boroughs with names
###
head(LB$NAME)
Bname <- gsub(" London Boro","",LB$NAME)
xy <- coordinates(LB)
plot(LB)
text(xy[,1],xy[,2],Bname,col="blue",cex=0.5)
box()
title("London Borough Boundaries")



quickMap <- function(Var,nClass=10){
  require(classInt)
  require(RColorBrewer)
  Classes <- classIntervals(Var,nClass,method="quantile")
  Palette <- brewer.pal(nClass,"Reds")
  Colours <- findColours(Classes,Palette)
  plot(y)
  points(x.sdf2,cex=0.5,pch=16,col=Colours)
  }
```

```
###
 data.frame(Bname,LB$NAME)                  # check ordering of names
 head(MyData)                               # and MyData
 NB <- length(LB)                           # number of boroughs
 results <- matrix(0,NB,2)                  # storage for borough legfel coefficients
 for(i in 1:NB) {
    m.x <- lm(Purprice~FlorArea,data=MyData[MyData$Borough == Bname[i],])
    results[i,] <- coef(m.x)
 }
rownames(results) <- Bname                  # add in names
colnames(results) <- c("Intercept","FlorArea")
print(results)
hist(results[,2])                           # look at FlorArea coefficient
boxplot(results[,2])

###
### borough levels plots with legend
###
quickMap2 <- function(Var,nClass=9,dp=0,plotNames=FALSE){
   require(classInt)
   require(RColorBrewer)
   Classes <- classIntervals(Var,nClass,method="quantile",dataPrecision=dp)
   Palette <- brewer.pal(nClass,"Reds")
   Colours <- findColours(Classes,Palette)
   plot(LB,col=Colours)
   legend("bottomright",
      legend=names(attr(Colours,"table")),
      fill=attr(Colours,"palette"),
      cex=0.75,bty="n")
   box()
   if(plotNames) {
      xy <- coordinates(LB)
      text(xy[,1],xy[,2],Bname,col="blue",cex=0.5)
   }
}

quickMap2(results[,2])                   # without borough names
quickMap2(results[,2],plotNames=TRUE)    # with borough names



###
### and the residuals from the model? Plot the borough medians
###
quickMap2(tapply(MyData$stdres.9v,MyData$Borough,median),plotNames=TRUE,dp=3)


corrplot(cor(MyData[,c("FlorArea","Purprice")]), method="number")
```

##**Applying Random Forest**

```
suppressMessages(library(randomForest))
set.seed(111)
s <- sample(nrow(MyData), round(.6*nrow(MyData)))
train_data<- MyData[s,]
test_data<- MyData[-s,]

rf <- randomForest(log(Purprice)~FlorArea+Bedrooms+Type+BathTwo+Garage+Tenfree+CenHeat+Age+ProfPct,     data=train_data)
##Predicting train error
pred <- predict(rf, train_data)
mean((pred - log(train_data$Purprice))^2)
##Predicting test error
pred1 <- predict(rf, test_data)
mean((pred1 - log(test_data$Purprice))^2)
#Test MSE is 0.079
#Applying linear regression with log(Purprice)
model <- lm(log(Purprice)~FlorArea+Bedrooms+Type+BathTwo+Garage+Tenfree+CenHeat+Age+ProfPct,data=MyData)
summary(model)
anova(model)
#MSE is 0.08
#Random Forest is fitting the data slightly better
```

#**BaseLine Regression and AIC**

```
#Fit models for a single variable and look at AICs - model with *lowest* AIC is closest to unknown 'true' model

AICs <- rep(NA,10)
Models <- vector("list",10)
Vars <- colnames(MyData)[4:13]
for(i in 1:10) {
Models[[i]] <- lm(formula(paste0("Purprice~",Vars[i])),data=MyData)
AICs[i] <- AIC(Models[[i]])
}

print(AICs)
minAIC <- which.min(AICs)
print(AICs[minAIC])

print(Vars[minAIC])
summary(Models[[minAIC]])

# Look at the differences

names(AICs) <- Vars     # add names
```

```
sAICs <- sort(AICs)      # sort into order
print(sAICs)
plot(sAICs,xaxt="n")     # plot
axis(1,labels=names(sAICs),at=1:length(Vars),las=2,cex.axis=.75)

for(i in 2:length(Vars)){ # compute differences
cat(paste(names(sAICs)[i],sAICs[i]-sAICs[i-1],"\n"))
}


#Pairwise comparisons using t tests with pooled SD
pairwise.t.test(MyData$Purprice, MyData$Age, p.adjust = "bonferroni")

#Welch two sample t-test
t.test(

Purprice ~ CenHeat,
data = temp,
alternative = "two.sided",
var.equal = FALSE
)
temp <- MyData[, c(3, 5, 10)]
temp$Purprice <- log(temp$Purprice)

ggboxplot(
temp,
y = "Purprice" ,
x = "CenHeat",
color = "CenHeat",
ylab = "Price",
xlab = "Central Heating Impact"
)

#So the model with the lowest AIC is FlorArea - most variables add a little something
#Residual plots vs Fitted values


model.9v <-
lm(Purprice~FlorArea+Bedrooms+Type+BathTwo+Garage+Tenfree+CenHeat+Age+ProfPct,data=MyDa
ta)
summary(model.9v)      # adj r^2 ~ .56
plot(model.9v)


hist(LondonData$Purprice, col = "blue", main = "Freq histogram of Purchase Price") boxplot(
LondonData$RetiPct,col = "blue", pars = list(outcol = "red"), main = "Box plot of RetiPct" ) boxplot(
LondonData$Unemploy, col = "blue", pars = list(outcol = "red"), main = "Box plot of Unemploy" )
corrplot(cor(MyData[, c("FlorArea", "ProfPct", "Purprice")]), method = "pie")
```

```
#Best Subsets Regression
library(leaps)
allfits <- regsubsets(log(Purprice)~., data=MyData, really.big = T)
summary(allfits)$which
plot(allfits, scale="r2" , col = "yellow",main="Best Subsets Regression")
sum.allfits <- summary(fwfits)
names(sum.allfits)
par(mfrow=c(1,1))
npred <-1:8
plot(npred, sum.allfits$cp, pch=20, xlab="number of predictors")
lines(npred, sum.allfits$cp)
plot(npred, sum.allfits$rsq, pch=20,xlab="number of predictors")
lines(npred, sum.allfits$rsq)

library(pls)
MyData1 <- data.frame(LondonData[,c(2:4,15:17,22,23,26)],Age,Type,Garage,Bedrooms)
pcr.fit <- pcr(Purprice ~., data=MyData1, scale= TRUE, validation = "CV")
summary(pcr.fit)
validationplot(pcr.fit, val.type = "MSEP")

#GWR model
x.gwr<- gwr.basic(Purprice~FlorArea,data=x.sdf2[x.select,],bw=250,adaptive=T,dMat= x.dmat)
x.dmat2 <- gw.dist(coordinates(x.sdf2[x.select,]),coordinates(x.sdf2))
```

References

Binbin Lu, Martin Charlton, A. Stewart Fotheringhama 'Geographically Weighted Regression Using a Non-Euclidean Distance Metric with a Study on London House Price Data'.

Cs229.stanford.edu. (2016). Real Estate Price Prediction with Regression and Classification. [online] Available at: http://cs229.stanford.edu/proj2016/report/WuYu_HousingPrice_report.pdf

Kaggle.com. (2017). Predicting House Prices using R | Kaggle. [online] Available at: https://www.kaggle.com/pradeeptripathi/predicting-house-prices-using-r

Lu, B, Charlton, M, Harris, P & Fotheringham, S 2014, 'Geographically weighted regression with a non-Euclidean distance metric: A case study using hedonic house price data' International Journal of Geographical Information Science