Name: Paula McMahon

Student ID: 17185602

Module: NCG612

Title: London House Price Prediction Project

Date: 28th May 2018

Executive Summary

- In this project, we were tasked with finding the most reliable determinants of property prices for a set of anonymised mortgage records for the Greater London Area.
- To do this, I used a barrage of techniques for data exploration including boxplots, pairs plots, residual analysis, multiple linear regression including interactions, coefficient analysis, Anova tests, splitting data into training and testing sets to produce model test error rates and spatial analysis by borough.
- My results were that a 9-predictor linear model (no interaction) produced the best results. Interactions produced higher values of adjusted R^2 but at the expense of dozens of non-significant predictors. The residuals show a definite spatial pattern between the boroughs. Model coefficient analysis shows a 0.71% proportional increase in average purchase price for a 1 m^2 increase in size.

Project Outline - Aims and Objectives

The "LondonData" dataset dimensions are 12,000+ rows and 31 variables. There is an easting and northing coordinate per row, variables to indicate the age of the property, type of property, whether it has a garage, central heating, two or more bathrooms, number of bedrooms, the floor area in metres squared, and other variables to indicate affluence related to cars and professions. The objective is to find the best group of predictors of property price. This involves choosing a model (or models) whose variables represent the trade-off between predictive power and simplest explanatory power. A commonly used methodology in valuation is known as hedonic modelling. The most accurate way to predict a value is to model it as a function of its attributes. The next section of the report interleaves Methods used, and the Results obtained. Conclusions then follow. For coding, all of my analysis was done in R. The code, with suitable comments, accompanies this report. R-code chunks were run with eval=FALSE so as to ensure the R-Markdown output was as short as possible.

Methods and Results

The methods and results sections for property price prediction are presented jointly in this section under several sub headings.

Data exploration, factor creation and outlier detection

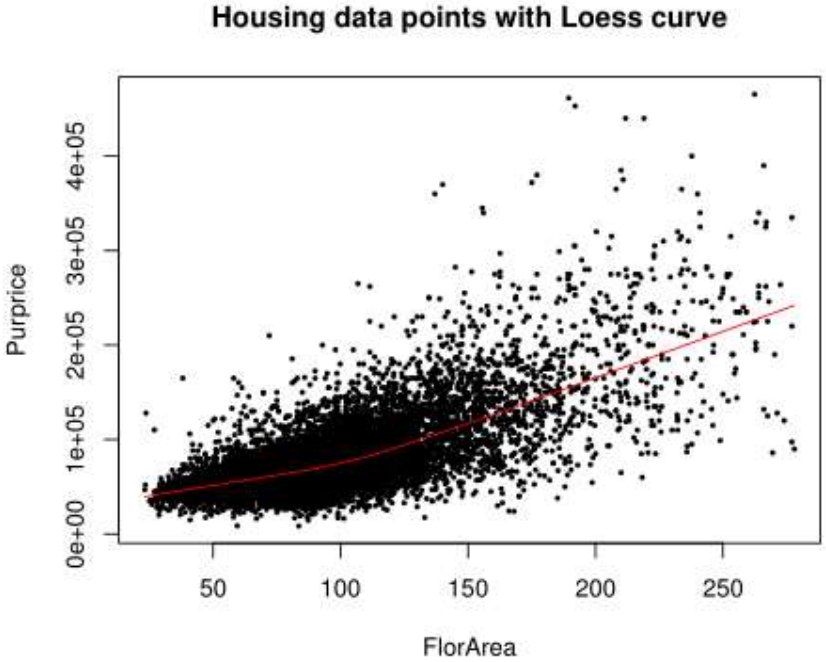
Upon reading in the data, a "summary" command shows, per variable, the minimum, maximum, lower and upper quartiles, median and mean values. This will highlight any anomalous data. From this we can see that:

- The maximum value of *Unemploy* (Proportion of unemployed workers) is 689. This value is highly suspicious and so this variable will be dropped from the analysis.
- The maximum value of *RetiPct* (Proportion of residents retired) is 900. This value is also highly suspicious and so this variable will be dropped from the analysis.
- The maximum value of *Purprice* (Purchase price in GBP) is £850,000. The median and mean values for this variable are £70,000 and £80,000 respectively. Therefore, the maximum value warrants further investigation.

At this point, the dummy variables were converted to factors which is more convenient for modelling. The R code shows the method for doing this.

Below is a boxplot of properties priced under £250,000. A boxplot is constructed by drawing a box between the upper and lower quartiles with a solid line drawn across the box to locate the median. The fences of a boxplot are needed for identifying extreme values in the tails of the distribution: lower fence: Q1 - 1.5*IQR, upper fence: Q3 + 1.5*IQR where IQR is the inter quartile range. The boxplot shows that the median value of a property is £70,000 and the IQR is £35,000.

Below is a plot of the data points of *FlorArea* vs *Purprice* with a superimposed loess line. Prices below £500,000 are only shown. This graph show that, unsurprisingly, larger house cost more. The relationship at the lower ends of the graph is quite gentle and increases more steeply for larger *FlorArea* values.



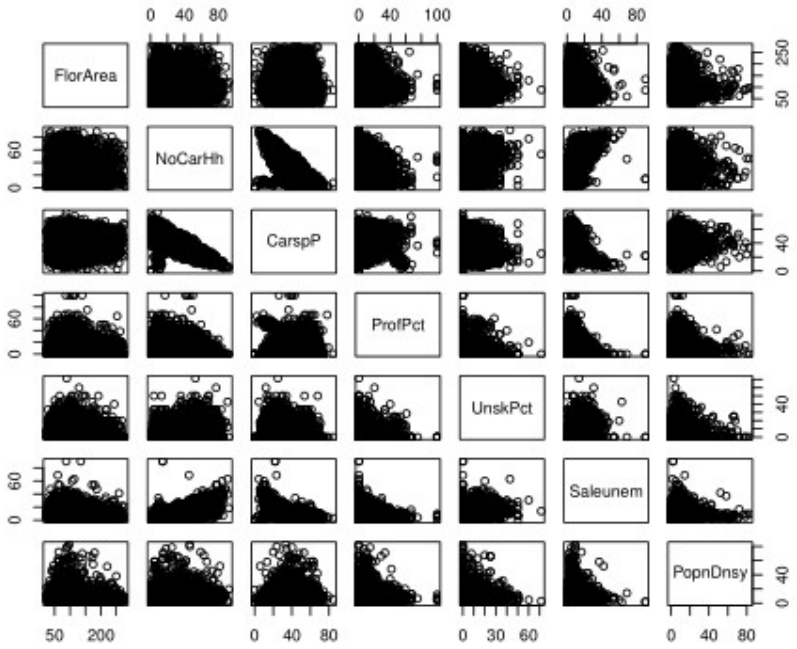
The maximum value of *Purprice* of £850,000 mentioned earlier will now be addressed. It features at row 4800 of the dataset.

Borough	Purprice	Tenfree	CenHeat	BathTwo	FlorArea	Age	Type	Garage	Beds
Lambeth	850,000	No	Yes	Yes	168 sqm	PreW	Bung	No	3

This house is in the *Lambeth* borough. *Lambeth* is the 11th cheapest of the 33 London Boroughs. It is a 168m^2 3-bedroom/2-bathroom bungalow with reasonably average floor area size. The purchase price of £850,000 seems far too high for a property in this area. Perhaps an extra zero was incorrectly recorded for the purchase price or perhaps a historic figure once lived there. It seems like it's an outlier so let's remove it from the analysis.

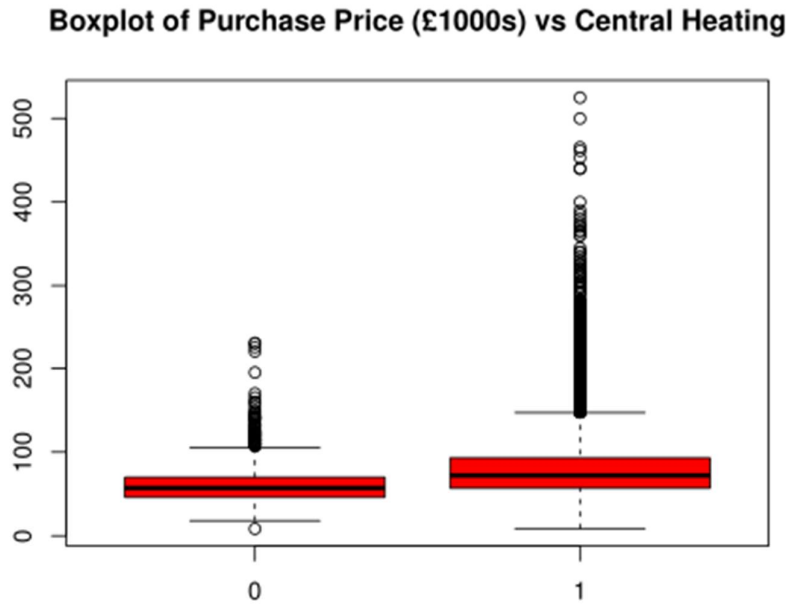
Correlation

Generate a correlation matrix to get an idea of globally correlated variables. This may suggest collinearity. Collinearity is a problem particularly in spatial datasets where, in heterogenous data, some locations may exhibit collinearity while others may not. Therefore, to generate the correlation matrix, use the un-factored data and only use numerical predictors. The strongest correlation is between *CarspP* and *NoCarHh* with a figure of -0.863. Overall, I would say collinearity is not a problem in this dataset.

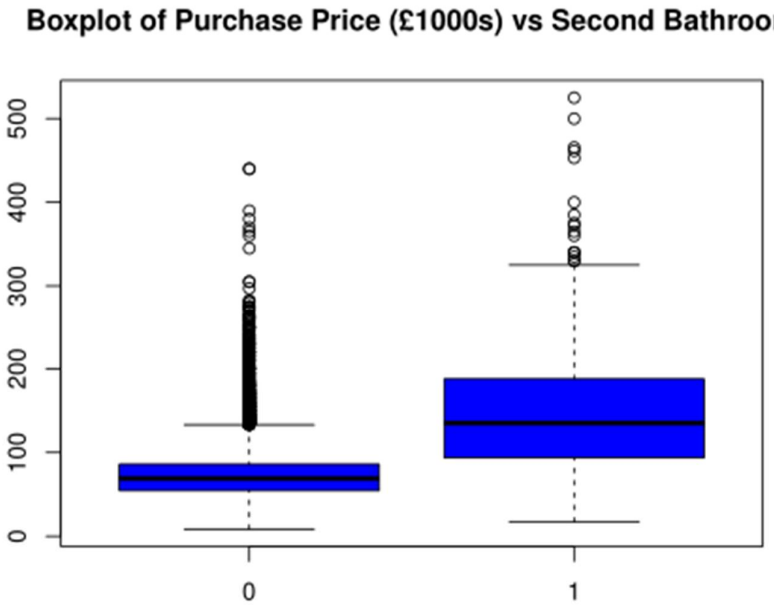


Boxplot analysis

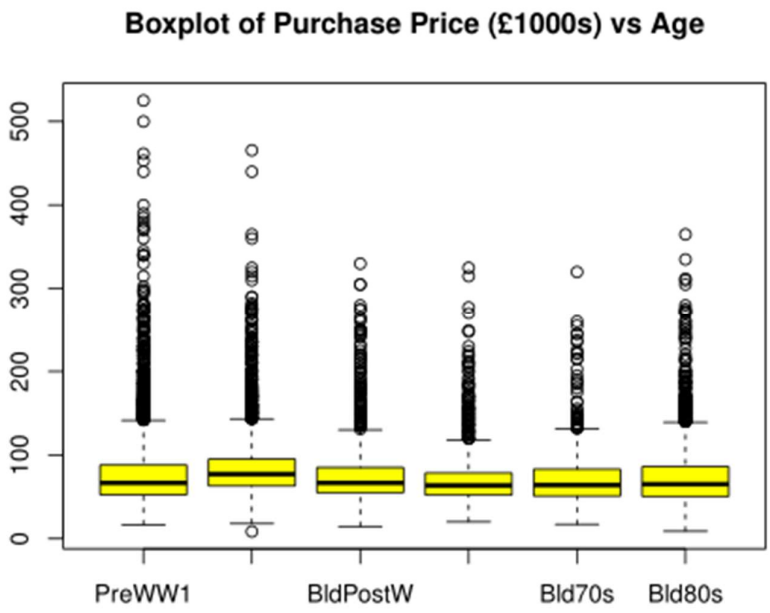
The following series of box plots show *Purprice* versus the factored variables. The first boxplot shows central heating (*CenHeat*). The median is lower for no central heating; therefore, one could infer that central heating increases the value of a house.



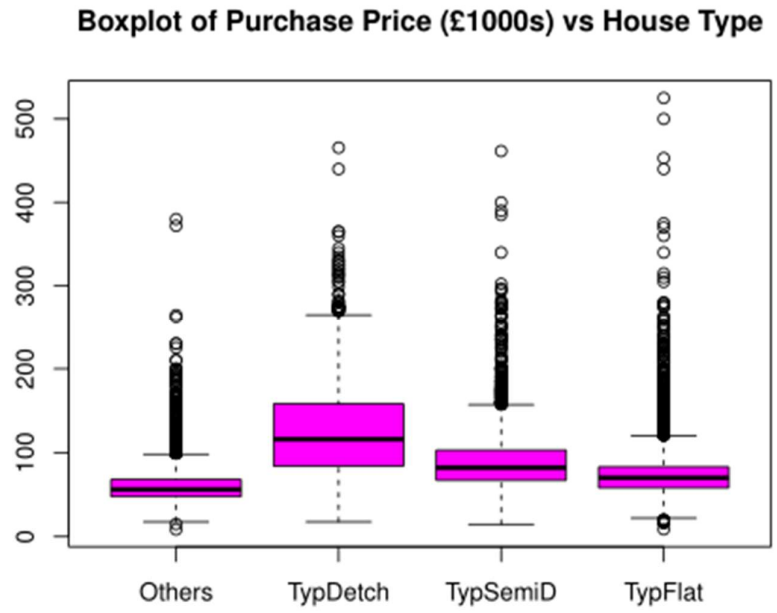
The next boxplot shows the effects of having a second bathroom (*Purprice ~ BathTwo*). There appears to be a significant increase in price for having two bathrooms in the property.



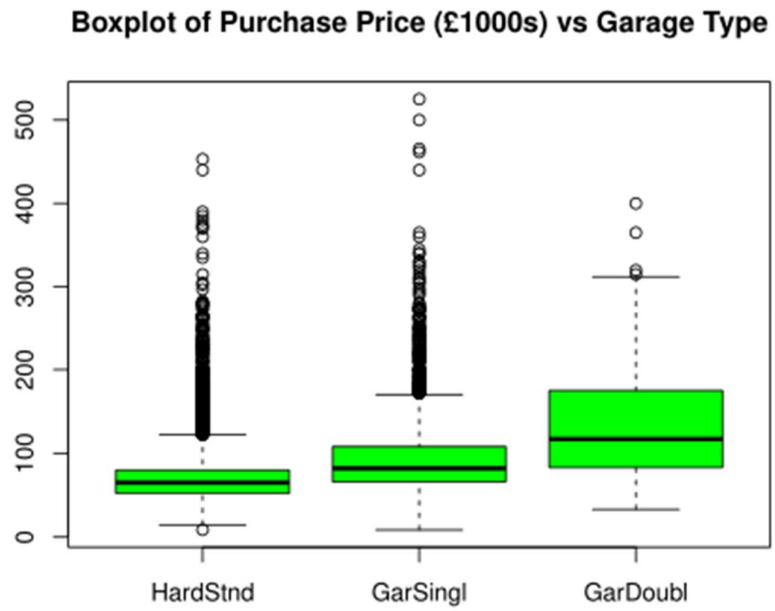
The next boxplot shows whether the age of the property affects the purchase price (*Purprice ~ Age*). There is not really much in the difference here. The properties built between WW1 and WW2 (*BldIntWr*) yield a slightly higher price.



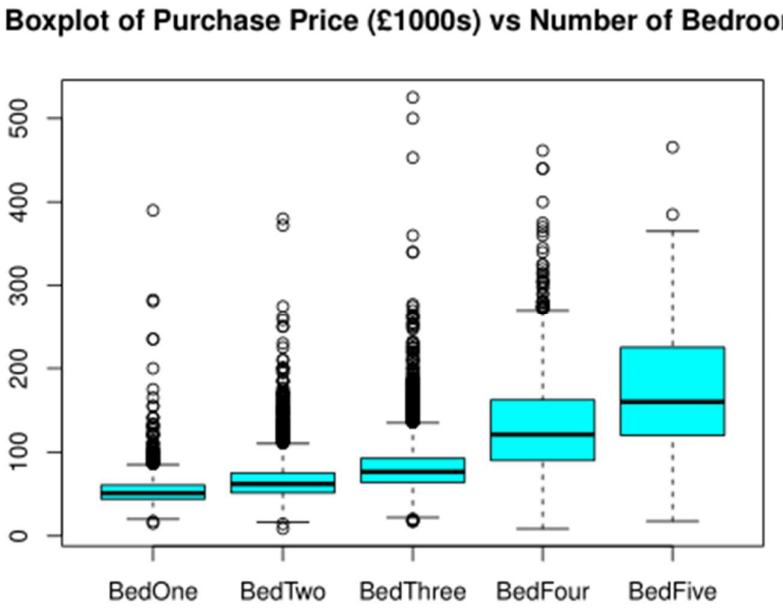
The next boxplot shows how the house type affects purchase price (*Purprice ~ Type*). It can be seen that there is a definite increase in price for having a semi-detached house and an even more significant increase for detached properties. There is real justification for adding the predictor "*Type*" to the model.



The next boxplot shows whether the existence of a garage at the property influences a house's price (*Purprice ~ Garage*). The plot shows that there is a definite additional price gain from having a garage, in particular, a double garage.



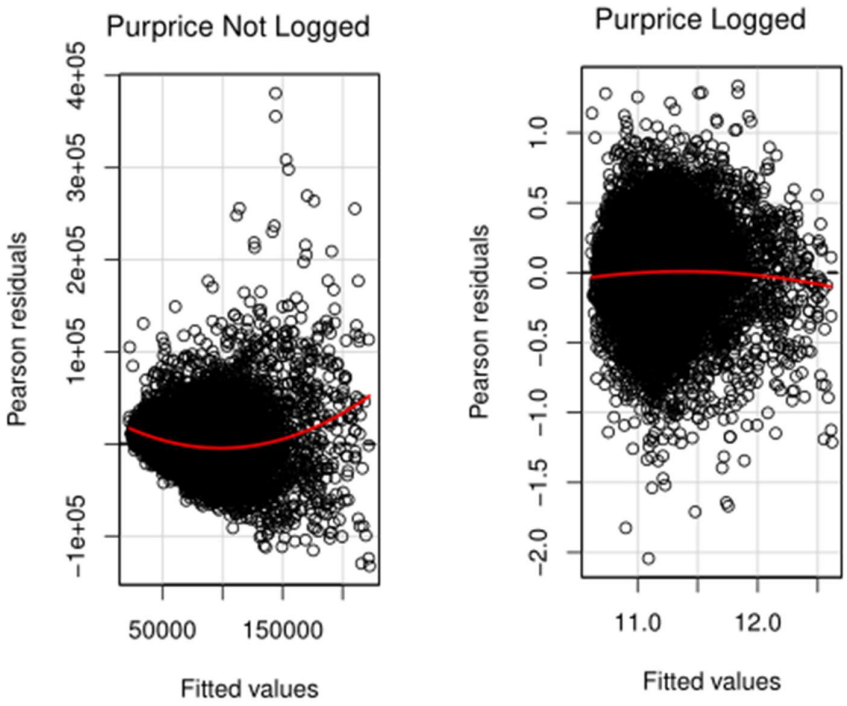
The last boxplot shows the effect that the number of bedrooms has on price (*Purprice ~ Bedrooms*). There is a steady price increase per additional bedroom and a significant gain in price once you go past three bedrooms. "*Bedrooms"* is most definitely an important predictor and should be included in the model.



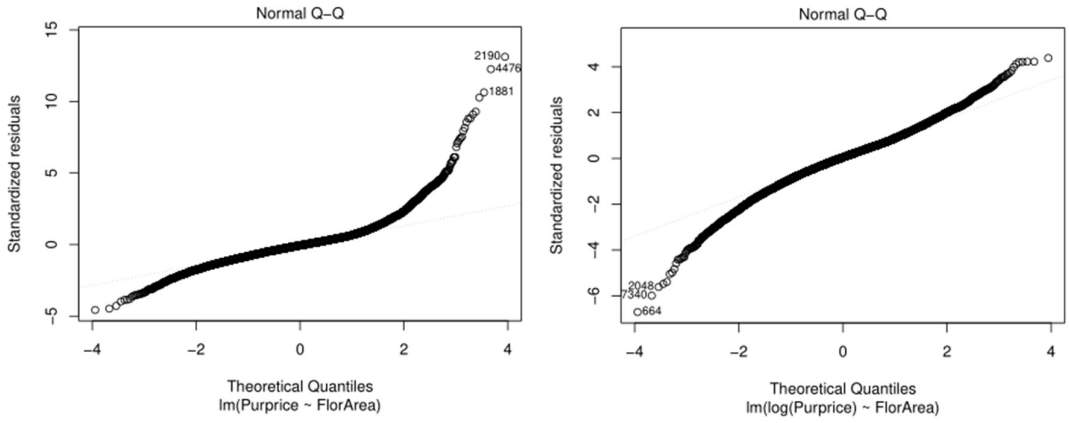
Residuals analysis – should we log Purprice?

Usually with price related data, it is advisable to log the response variable for a couple of reasons. Firstly, to address the skewness towards large values and secondly to show percent change or multiplicative factors. To see if *Purprice* should be logged for the rest of the analysis, let's fit a simple

linear model with one predictor, *FlorArea*. Let's show the residuals vs the fitted values before logging *Purprice* and also after. The plot on the left shows *Purprice* not logged and we see definite curvature in the data and non-constant variance (where the variability of residuals is changing). The logged version on the right shows less curvature.



Now, let's look at the normalised QQ plot of the residuals. A QQ plot is an alternative representation to a histogram. A histogram of the non-logged data would show a skewed right tail. In the images below, the plot on the left is the non-logged version versus the logged version on the right. We can see that the logged version of the residuals appears more normal. Therefore, when we use the log function, our house data appears to come from a Normal distribution as the data quantiles appear similar to those from an N(0,1) distribution. The decision to log *Purprice* is justified.



Finding the best predictors for the model

The strategy to find the best predictors for the model is:

- Fit a linear model with *Purprice* as the response and use the metric of lowest AIC to find the best predictor. (AIC is a goodness-of-fit method. AIC does not tell us about the absolute quality of a model, only the quality relative to other models).
- Add this predictor to the model.
- Record the adjusted R^2 value. R^2 measures the proportion of variance in Y as explained by regression with X. The adjusted R^2 value is a modified version of R^2. The adjusted R^2 increases only if the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected by chance.
- Run Anova tests comparing the newer, fuller model to the preceding one to make sure the p-value is significant i.e. meaning that the fuller model is better. Low p-values are indications of strong evidence against the null hypothesis (H_0: the extra parameters in the fuller model can be set to zero vs H_A: at least one of the extra parameters in the fuller model are not equal to zero). So, p-values less than 0.05 mean the fuller model is the more significant one.
- Repeat these steps as long as there is no evidence that the model has been overfit. (This will be explored in more detail later in the report).

No. of predictors	Predictor Names	Lowest AIC	Adjusted R^2	Anova
1	*FlorArea*	5783.612	0.4787	-
2	*FlorArea+Type*	5288.896	0.4989	2.2e-16 ***
3	*FlorArea+Type+CenHeat*	4840.646	0.5166	2.2e-16 ***
4	*FlorArea+Type+CenHeat+Age*	4596.024	0.5261	2.2e-16 ***
5	*FlorArea+Type+CenHeat+Age+BathTwo*	4451.073	0.5316	2.2e-16 ***
6	*FlorArea+Type+CenHeat+Age+BathTwo+Garage*	4354.956	0.5353	2.2e-16 ***
7	*FlorArea+Type+CenHeat+Age+BathTwo+Garage +Tenfree*	4274.261	0.5383	2.2e-16 ***
8	*FlorArea+Type+CenHeat+Age+BathTwo+Garage +Tenfree+Bedrooms*	4253.403	0.5392	8.518e-06 ***
9	*FlorArea+Type+CenHeat+Age+BathTwo+Garage +Tenfree+Bedrooms+NewPropD*	4248.243	0.5394	0.007502 **

We can see from the above table that the incremental changes to the adjusted R2 values are becoming quite small, so it looks as though we are reaching the point where we are trading off explanatory power for simplicity. The Anova significance values for the last two models are still significant but getting less so. At this point, I decided not to add anymore predictors to this model.

Therefore, the linear model I have chosen is:

```
lm9 <- lm(log(Purprice) ~ FlorArea + Type + CenHeat + Age + BathTwo + Garage + Tenfree + Bedrooms + NewPropD, data=MyData)
```

Looking at predictor interactions

Before we discuss the model coefficients for the 9-predictor model we have decided on, let's see if adding any interactions between the predictors improves the model. The interaction notation *FlorArea*Type* means to fit a separate slope for every *Type* and we effectively obtain a separate

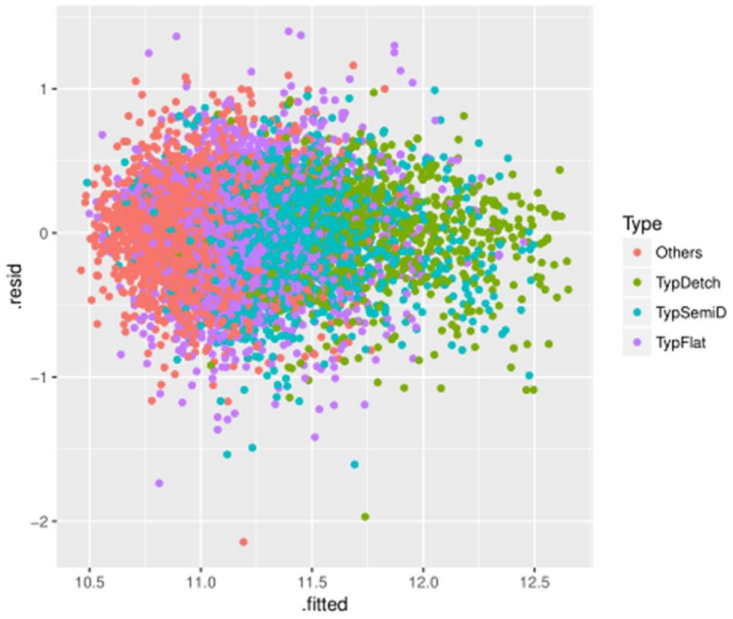
relationship for properties with different types. Generally, you can consider adding interactions that you think might be important based on the dataset knowledge. Let's take the first four predictors – *FlorArea*, *Type*, *CenHeat* and *Age*. Add them to the model plus their interactions like this:

```
lm4int <- lm(log(Purprice) ~ FlorArea+Type+CenHeat+Age+FlorArea*Type*CenHeat*Age, data=MyData)
```

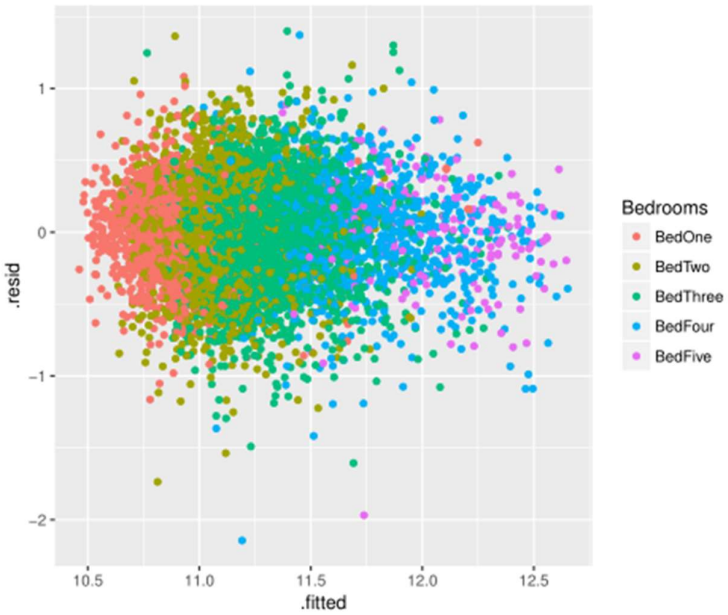
The adjusted R^2 for this model is 0.5439, which is the largest value seen so far. We should ask though, if we use this model, have we sacrificed performance for complexity? The model has almost 100 predictors, most of which are not significant. We could calculate a training error for this model (discussed later) but getting a test error is not possible in the setting I have used as you would need to add every interaction term to a data frame which is too complex for the scope of this project. In further analysis, the adjusted R^2 value increased for 5-predictor + 5-interaction terms and also for the 6-predictor + 6-interaction terms also but for complexity reasons these models are not appropriate. To summarise, interaction terms will not be further considered because the 9 predictors on their own work well enough for our purposes. Sometimes it's best to start simple, and add complexity only as needed.

Plotting the residuals

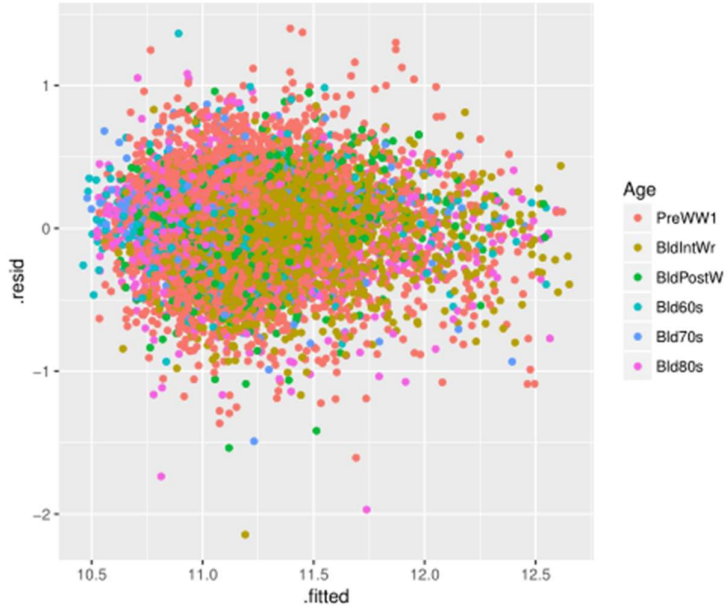
For the 9-predictor model, let's analyse the residuals for this model and how the break down in terms of (i) *House Type*, (ii) *the number of Bedrooms* and (iii) *the Age of the dwelling*. The first plot is residuals by house type. The largest positive residuals seem to be flat or apartment dwellings. These may be the ones in affluent areas. The largest negative residual at the bottom of the plot appears to be a bungalow.



Next is residuals by the number of *Bedrooms*. The largest positive residuals are a mixture of two, three and four bedrooms and we have already established that most of these are flat or apartment dwellings. The largest negative residual is the four-bedroom bungalow.



Lastly, let's look at residuals by *Age*. The largest positive residuals by far relate to PreWW1 dwellings (which are 2,3,4-bedroom apartments/flats) and the largest negative residual is the 4-bedroom bungalow built during the war years.



Interpreting the coefficients

As previously stated, I am using the following 9-predictor model. In R, this looks like:

```
lm9 <- lm(log(Purprice) ~ FlorArea + Type + CenHeat + Age + BathTwo + Garage + Tenfree + Bedrooms + NewPropD, data=MyData)
```

In terms of a linear model equation, this looks like:

$$\log(\text{Purprice}) = \beta_0 + \beta_1\text{FlorArea} + \beta_2\text{Type} + \beta_3\text{CenHeat} + \beta_4\text{Age} + \beta_5\text{BathTwo} + \beta_6\text{Garage} + \beta_7\text{TenFree} + \beta_8\text{Bedrooms} + \beta_9\text{NewPropD} + \epsilon$$

where the β's are the coefficient estimates and ϵ is the error terms for any terms not in the model. The coefficients for this model as reported in R are:

```
Coefficients:
                    Estimate
(Intercept)       10.3525764
FlorArea           0.0063334
TypeTypDetch       0.0053856
TypeTypSemiD      -0.0901536
TypeTypFlat       -0.1649424
CenHeat            0.1715357
AgeBldIntWr        0.0483882
AgeBldPostW       -0.0334688
AgeBld60s         -0.0895198
AgeBld70s         -0.0804697
AgeBld80s         -0.0040174
BathTwo            0.1496116
GarageGarSingl     0.0584977
GarageGarDoubl     0.0819906
Tenfree            0.1276451
BedroomsBedTwo     0.0383324
BedroomsBedThree   0.0385258
BedroomsBedFour    0.0768624
BedroomsBedFive    0.0341153
NewPropD           0.0436939
```

With so many predictors, explaining how each one affects *Purprice* would be overly complex for the purposes of this report so let's just, for the sake of simplicity, take a 3-model predictor.

```
lm3 <- lm(log(Purprice) ~ FlorArea + Type + CenHeat, data=MyData)
```

The linear equation for this model would be:

$$\log(\text{Purprice}) = \beta_0 + \beta_1\text{FlorArea} + \beta_2\text{Type} + \beta_3\text{CenHeat} + \epsilon$$

or to express it as an antilog of Purprice:

$$\text{Purprice} = e^{\beta 0 + \beta 1\text{FlorArea} + \beta 2\text{Type} + \beta 3\text{CenHeat} + \epsilon}$$

The coefficients and confidence intervals for this 3-predictor model as reported in R are:

```
## Coefficients:
##                Estimate
## (Intercept)   1.033e+01
## FlorArea      7.083e-03
## TypeTypDetch  1.730e-01
## TypeTypSemiD  6.834e-02
## TypeTypFlat  -3.088e-02
## CenHeat       1.732e-01
```

```
##                    2.5 %       97.5 %
## (Intercept)  10.313647522 10.352895104
## FlorArea      0.006914238  0.007252471
## TypeTypDetch  0.150350728  0.195626636
## TypeTypSemiD  0.052998179  0.083687369
## TypeTypFlat  -0.044758745 -0.017007296
## CenHeat       0.157312610  0.189027110
```

- Let's interpret the coefficients on the left: for fixed values of all other predictors, floor area (*FlorArea*) impacts positively on the purchase price of a dwelling (*Purprice*).
- For fixed values of all other predictors, a detached house (*TypeTypDetch*) impacts positively and significantly on the purchase price of a dwelling. The same can be said for having a centrally heated house (*CenHeat*).
- For fixed values of all other predictors, a semi-detached house (*TypeTypSemiD*) impacts positively but less significantly on the purchase price than a detached house would.
- For fixed values of all other predictors, a flat or apartment type of dwelling impacts negatively on the purchase price.

- Looking specifically at *FlorArea* as it is the most significant predictor, *FlorArea* changes log(*Purprice*) by 0.007083 for fixed values of other variables.
- *FlorArea* changes *Purprice* by a factor of $e^{0.007083}$ (1.0071) with a 95% confidence interval of $e^{0.0069}$, $e^{0.0073}$ (1.0066, 1.0073)
- This means that *FlorArea* increases *Purprice* by 0.71% with a 95% confidence interval of 0.66% and 0.73%
- The important point to note here is that when using logs, you can never get a negative house price which makes perfect sense for our dataset. When using logs, we are talking about a proportion of an increase (or decrease as the case may be) rather than a fixed increase.
- So, 0.71% is the estimated proportional change in average *Purprice* associated with a 1 m^2 change in *FlorArea* for fixed values of all other predictors.
- The same analysis could be done for *Type* and *CenHeat* as explained above.

Training and Test datasets

When you have many variables without enough data, it is possible that your model overfits to the data by placing an emphasis on unimportant variables. You split the data into training and test sets to be able to obtain a realistic evaluation of your learned model. If you evaluate your learned model with the training data, you obtain an optimistic measure of the goodness of your model. So, you should use a separate testing set to obtain a realistic evaluation of your model. In R, I computed the training and test errors for models using one predictor increasing to nine predictors. I set a seed for reproducibility and split the housing data set into 50% training and 50% test.

The method for calculating each training error (mean squared error) is: $\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$ where N is the number of observations in the training data set, y_i is the logged value of the training *Purprice* value and $\hat{y}_i$ is the predicted value of *Purprice* using the model created on the training data set.

The method for calculating the test error is to create a data frame using the predict function in R on the model created with the training set along with actual values for the predictors from the test data set. Use the same formula as above but this time N is the number of observation in the testing set, y_i is the logged value of the test set *Purprice* value and $\hat{y}_i$ is predicted values from the newly created data frame.

The table below summarises the error rate findings. As we add predictors we can see that the mean squared errors are getting smaller. Also, the test mean squared error we get with a predictor or multiple predictors is larger than the error we get with the training set which is what we would expect. At some point if we have overfit the model, we would expect to see the test error rise, but the training error would continue to fall, the classic sign of overfit data.

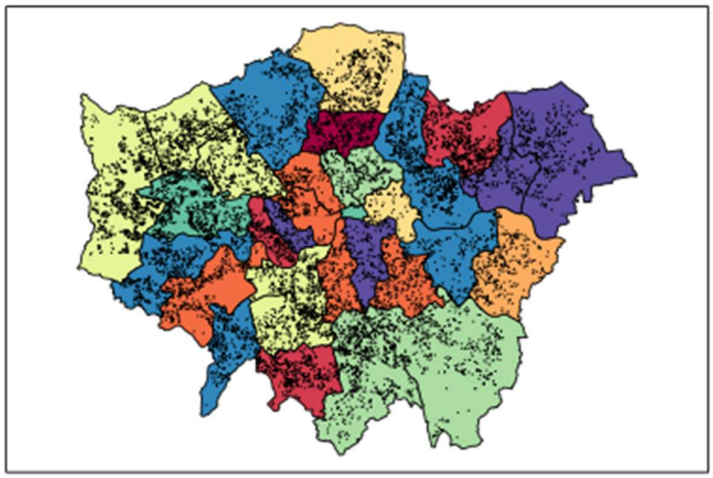
Predictors	**Predictor names**	**Training Error MSE**	**Test Error MSE**
1	*FlorArea*	0.091627	0.09404
2	*FlorArea+Type*	0.0879513	0.090515
3	*FlorArea+Type+CenHeat*	0.0852315	0.086975
4	*FlorArea+Type+CenHeat+Age*	0.0836699	0.085129
5	*FlorArea+Type+CenHeat+Age+BathTwo*	0.0826485	0.084175
6	*FlorArea+Type+CenHeat+Age+BathTwo+Garage*	0.0816611	0.083925
7	*FlorArea+Type+CenHeat+Age+BathTwo+Garage+Tenfree*	0.080669	0.083999
8	*FlorArea+Type+CenHeat+Age+BathTwo+Garage+Tenfree +Bedrooms*	0.0805805	0.083749

9	*FlorArea+Type+CenHeat+Age+BathTwo+Garage+Tenfree +Bedrooms+NewPropD*	0.0805426	0.083694

Plotting London Boundaries and Data

Now we move towards introducing our spatial components, *Easting* and *Northing* into the analysis. We were provided with some files which show the shapes of the 33 boroughs of London. First, let's produce a plot showing the housing data points overlaid with the London Borough boundaries. We use the *readOGR* function from the "rgdal" library to read a shapefile into a suitable Spatial vector object. We then create a SpatialPointsDataFrame of the Easting and Northing housing points. We plot the boroughs, applying some colour to them, then overlay the housing data points producing the plot below.

London Boroughs with housing data points

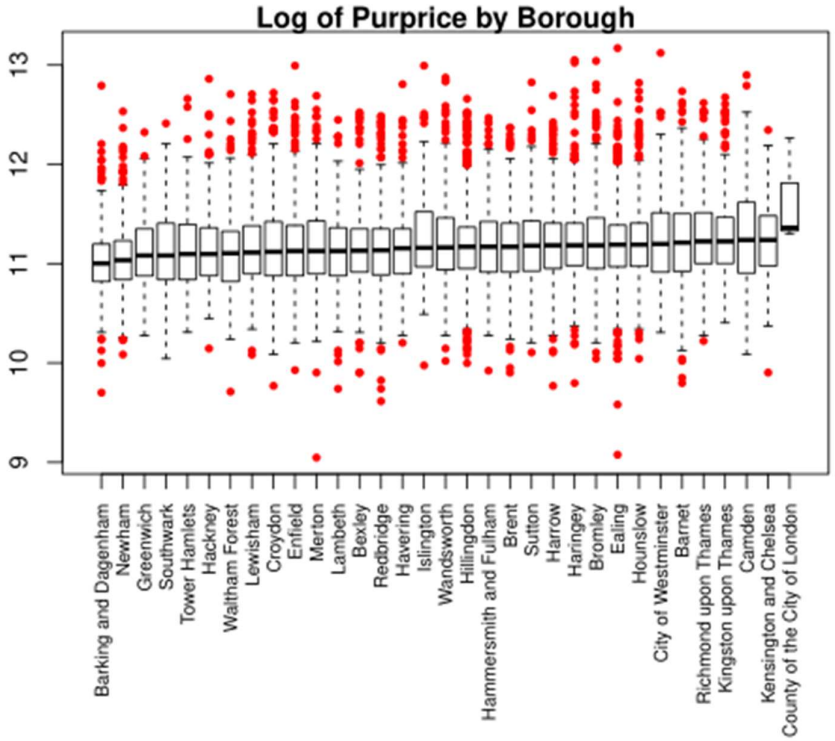


The next map overlays the borough boundaries with their names. Some text manipulation of the borough names is required using the *gsub* function which leaves us with the shortened borough name which we can then plot. Each borough text string is overlaid on its respective borough.

London Borough Names and Boundaries

Spatial Analysis

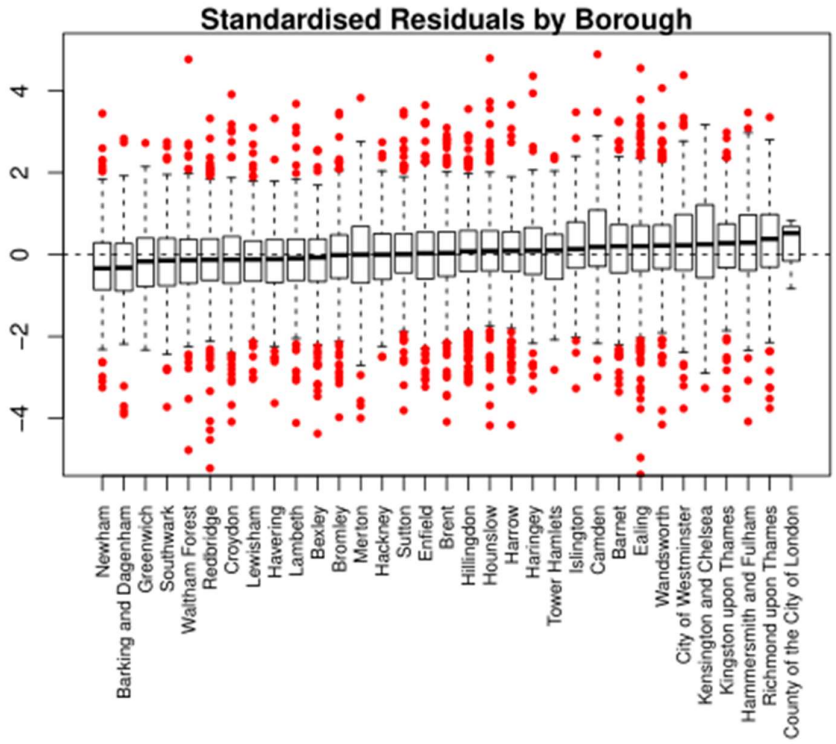
Now, some spatial analysis of the London boroughs can be done. Firstly, make a boxplot of logged *Purprice* by borough. This tells us that the City of London borough is the most expensive and the borough of Barking and Dagenham is the cheapest.



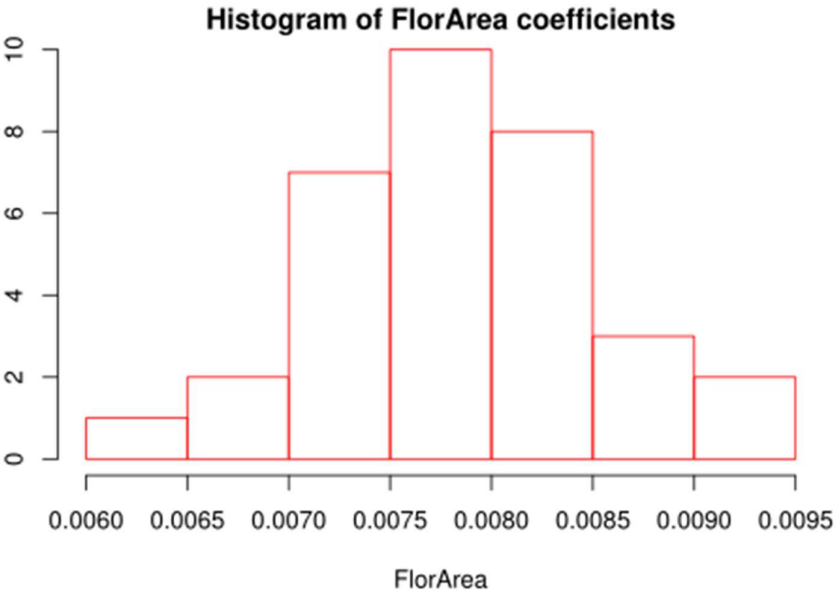
Another interesting plot to view is a boxplot of the standardised residuals, of the chosen model, per borough. Create the model, then filter out the standardised residuals using the R command:

```
MyData$stdres9lm <- stdres(model9lm)
```

Using the vector of standardised residuals above, run the boxplot command per borough adding in the dashed line at the zero axis. This boxplot is shown below. This line shows that there is a definite spatial pattern between the boroughs. Ideally, residual values should be zero meaning that the model fits very well. However, no model is ideal. Low residual values (straddling either side of the zero line) exist in boroughs such as Hackney, Sutton and Enfield. The model fits well here. Large positive or large negative values of residuals mean that the model was not such a good fit. Large negative residuals exist for boroughs like Newham and Barking and Dagenham. This means that property prices are below average for these areas. Large positive residuals exist for boroughs like the City of London and Richmond-Upon-Thames. We know these are affluent areas and so in reality property prices are well above the average predicted by our model. This shows that location is definitely a factor in this data.

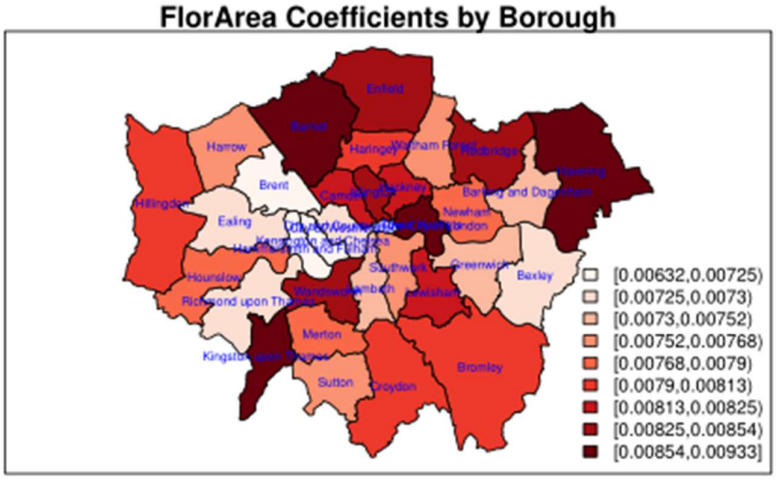


Next, fit a model per borough. Just fit one predictor, *FlorArea* for simplicity. This means we can obtain coefficients on a per borough basis. The histogram below (we could also use a boxplot) shows that the median value of *FlorArea* coefficient is 0.007 – 0.0080 and that the coefficient values for this predictor are quite normally distributed.

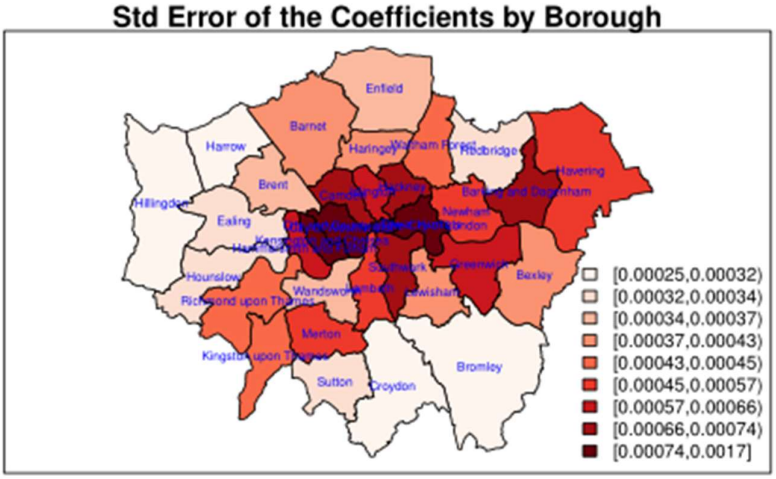


Next, we can produce a choropleth map of model coefficients per borough. A choropleth map is a thematic map in which areas are shaded or patterned in proportion to the measurement of the statistical variable being displayed on the map, in our case *FlorArea* coefficients and standard errors. This first map shows the level of variability in the *FlorArea* predictor. The coefficients describe the

degree of “effect” of the predictor. The lower coefficient estimates can be found in boroughs like Brent, Ealing and Bexley where it would seem that *FlorArea* influences *Purprice* to a less significant degree. The larger coefficient estimates in boroughs like Kingston-Upon-Thames, Barnet and Tower Hamlets means that *FlorArea* has a very large effect on house prices in these areas.



Secondly, we can produce a choropleth map of the standard errors associated with the coefficients. The standard error is an estimate of the standard deviation of the coefficient, the amount it varies across cases. It can be thought of as a measure of the precision with which the regression coefficient is measured. So, the lightly shaded areas on the map correspond to a low standard error meaning that the regression coefficient was precise in these areas. This occurred in areas like Bromley, Croyden, Harrow and Hillingdon. The darker shaded areas on the map correspond to a higher standard error meaning that the regression coefficient was less precise in these areas. This occurred in boroughs like City of London, Tower Hamlets, City of Westminster and Kensington and Chelsea. So, why might *FlorArea* not be a good predictor for these areas? Well, their proximity to the centre of London means that dwellings in these areas are going to be incredibly expensive almost regardless of their size. These areas would have many one or two-bedroom apartments, possibly Georgian in style and these would be completely unrelatable to a one or two-bedroom flat or dwelling on the outskirts of the greater London area. So, to see a map with more dense colours close to the centre and paler colours on the extremities of the map is completely expected here.



Geographically Weighted Regression

Another technique which I did not use for the project, but one which could have justifiably been applied is GWR (Geographically Weighted Regression). This technique is an R library developed by Brunsdon, Charlton et al from the NCG department in Maynooth University. The approach uses a moving window weighting technique, where localized models are used at target locations. At some target location, all neighbouring observations are weighted according to some distance-decay kernel function and then apply the linear model locally to this weighted data. The size of the window over which this localized model might apply is controlled by the bandwidth. Small bandwidths lead to more rapid spatial variation in the results while large bandwidths yield results increasingly close to the universal model solution. A bandwidth can be found using cross-validation techniques. The challenge to using GWR on the London House Price Data is that it is too large, so you would have to take a sample of around 30% of the observations. Additionally, a slight caveat about applying this technique to this dataset is that prices don't stop at the borough boundaries, so this may impact on the inferences from your results.

Conclusions

The objectives set out in this project posed many questions. Let's determine if the analysis done during this project has answered them.

- What are the best predictors of property price? Numerous combinations of predictors would have produced an adequate model. My approach was to add predictors to the model in order of the significance of their low AIC value. I used 9-predictors (*FlorArea*, *Type*, *CenHeat*, *Age*, *BathTwo*, *Garage*, *Tenfree*, *Bedrooms*, *NewPropD*) and stopped at this point for three reasons:
    - The adjusted R^2 value was increasing by very small amounts.
    - The test error rate had not deviated from the training error i.e. the model had not overfit the data.
    - Anova tests confirmed that the fuller model was the more significant one.
- Within the chosen model, do the coefficients seem sensible? I have explained, in the "Interpreting the coefficients" section, albeit with a more simplistic model, how to interpret the coefficients and what impact or effect they have on the overall prediction of house prices.
- Do residuals deviate from zero mean, independent and homoscedastic? Are there large residuals? Why are they unusual? Does location play a role? In this dataset, the residuals definitely deviate from zero mean and the homoscedastic trait of similar variance. A lot of the large positive residuals as seen in the 9-predictor model are related to flat or apartment dwellings. When you look at the residuals plot by *Age*, a lot of these type of residuals date from before WW1. The boxplot of standardised residuals by borough shows that the City of London has the largest cohort of positive residuals. There are a lot of small but very expensive Georgian apartments in this borough, so this would explain the large residuals from a model where *FlorArea* is a key predictor.
- So, overall, location plays a significant role in house price prediction.

References

Gollini, I., Lu, B., Charlton, M., Brunsdon, C., Harris, P. (2015) GWmodel: An R Package for Exploring Spatial Heterogeneity Using Geographically Weighted Models. Journal of Statistical Software, January 2015, Volume 63, Issue 17.

Charlton, M., Brunsdon, C., NCG612/NCG613 course notes, sample R code.