STAC32 Final Project

Yue Gan 1000620606, Liang Liang 1000415525, Zhu Zhu 1000421554, Yu Qian 1000823460, Liang Zhou 1001161574

2017-12-01

# Introduction

Now days, the real estate market has become one of the newly favoured type of investments. People pay attention on the market while selecting a property to either live or to invest. Many of them have already discovered that the location of a property play an important role in setting up the asking price. Many of Toronto neighbourhoods are in such strategic locations for employment, that given the housing shortage, urban intensification, poor transit and roadways, that the condos and homes in them will never see a significant price drop. (<http://gordcollins.com/real-estate/toronto-forecast-2017/>) However, within the same area, there are still some significant price difference that could be discovered through variations of different types of property. In the following report, the selected data <http://www.utsc.utoronto>. ca/~butler/c32/houses.xlsx. was used to look at the relationships between the price and the features of the properties such as the number of bedrooms and bathrooms, and the type pf property for over 50 properties in the Scarborough area. The goal is to create a model that would help people to predict the asking price for a property.

# About the Data

### subjects and groups

subjects and groups Fifty-four of the MLS listing was selected from realtors.ca in this study to form the dataset <http://www.utsc.utoronto>. ca/~butler/c32/houses.xlsx. All the MLS listing is selected from the Scarborough area. The subjects were divided into two groups: variables and different types. Each MLS listing has its corresponding bathrooms, washrooms and property types. Fifty-four subjects were divided into two groups, variable or property type. For variable group, it examines the relationship between the different rooms and the asking price. For the property type group, subjects fall into one of the three different types, townhouse, condo, or house category. The mean for the asking price for this fifty-five properties is 461801.6 and the median is 378900.

house

## # A tibble: 54 x 5  
## mls\_listing asking bedrooms bathrooms type  
## <dbl> <dbl> <dbl> <dbl> <chr>  
## 1 3573407 275000 1.5 1 apartment  
## 2 3580709 434999 3.5 3 townhouse  
## 3 3582848 675000 5.0 2 house  
## 4 3585741 385000 3.5 3 townhouse  
## 5 3599567 529000 3.5 2 apartment  
## 6 3602510 849888 5.0 2 house  
## 7 3607807 139900 3.0 2 apartment  
## 8 3610979 334900 2.0 2 apartment  
## 9 3611464 249000 3.0 2 townhouse  
## 10 3617396 259000 5.0 2 townhouse  
## # ... with 44 more rows

summary(house)

## mls\_listing asking bedrooms bathrooms   
## Min. :3573407 Min. : 95000 Min. :0.000 Min. :1.000   
## 1st Qu.:3623817 1st Qu.:275750 1st Qu.:2.000 1st Qu.:1.000   
## Median :3635660 Median :378900 Median :3.000 Median :2.000   
## Mean :3629677 Mean :461802 Mean :3.083 Mean :1.889   
## 3rd Qu.:3643350 3rd Qu.:698675 3rd Qu.:4.000 3rd Qu.:2.000   
## Max. :3648972 Max. :874900 Max. :6.000 Max. :4.000   
## type   
## Length:54   
## Class :character   
## Mode :character   
##   
##   
##

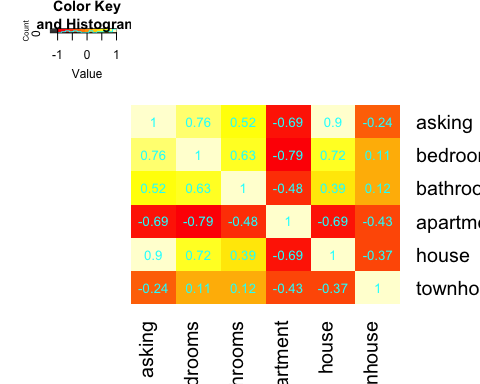
The mean of asking is 461802, bedrooms is 3.083 and bathrooms is 1.889. The median of asking is 378900,bedrooms is 3.000 and bathrooms is 2.000.

## Data Exploration

# Exploring the coorelation of the variables using heatmap  
corr = round(cor(tmp\_house), 2)  
heatmap.2(corr, trace="none", cellnote = corr, Rowv=FALSE, symm=TRUE)

## Warning in heatmap.2(corr, trace = "none", cellnote = corr, Rowv = FALSE, :  
## Discrepancy: Rowv is FALSE, while dendrogram is `both'. Omitting row  
## dendogram.

## Warning in heatmap.2(corr, trace = "none", cellnote = corr, Rowv = FALSE, :  
## Discrepancy: Colv is FALSE, while dendrogram is `column'. Omitting column  
## dendogram.

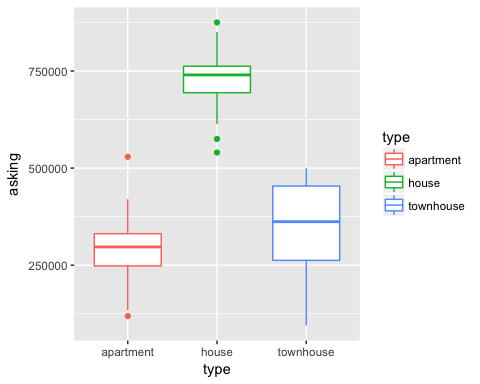


We can see that bedrooms and apartment tends to have a high correlation with asking.

## Checking Relationship between asking, bedrooms, bathrooms  
plot1<-ggpairs(data=house, columns=1:3,mapping = aes(color = "dark green"),axisLabels="show")  
plot1



## Checking Relationship between asking and type   
plot2 <-ggplot(house,aes(x=type,y=asking,colour=type))+geom\_boxplot()  
plot2

 There are two outlires in the apartment.

## Price vs. Bathrooms  
par(mfrow = c(1,3))  
hist(house$bathrooms, breaks = 20, xlab = "", col = "lightsteelblue", main = "Bathrooms")  
plot(density(house$bathrooms), xlab="", col = "steelblue", main="Bathrooms")  
scatter.smooth(house$bathrooms, house$asking, col="steelblue", xlab="", ylab="Price",main="Bathrooms",lpars=list(col="red", lwd=2))

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : pseudoinverse used at 0.985

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : neighborhood radius 1.015

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : There are other near singularities as well. 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : pseudoinverse used at 0.985

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : neighborhood radius 1.015

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : There are other near singularities as well. 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : pseudoinverse used at 0.985

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : neighborhood radius 1.015

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : There are other near singularities as well. 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : pseudoinverse used at 0.985

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : neighborhood radius 1.015

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : reciprocal condition number 0

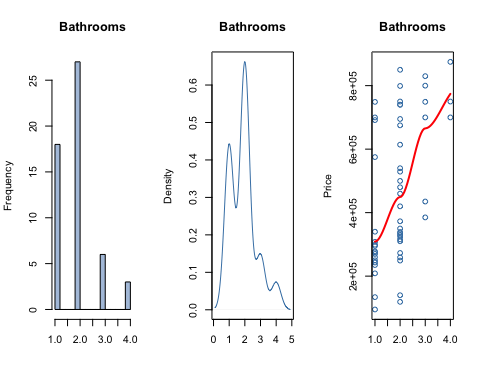
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : There are other near singularities as well. 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : pseudoinverse used at 0.985

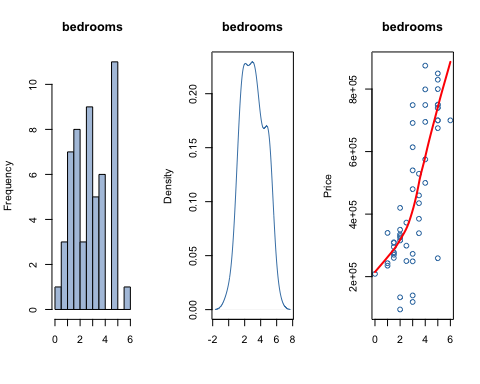
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : neighborhood radius 1.015

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## FALSE, : There are other near singularities as well. 1

 Nice correlation, as # of bahtrooms increases [median of bar plot], price increases as well

## Price vs. Bedrooms   
par(mfrow = c(1,3))  
hist(house$bedrooms, breaks = 20, xlab = "", col = "lightsteelblue", main = "bedrooms")  
plot(density(house$bedrooms), xlab="", col = "steelblue", main="bedrooms")  
scatter.smooth(house$bedrooms, house$asking, col="steelblue", xlab="", ylab="Price",main="bedrooms",lpars=list(col="red", lwd=2))

 Nice correlation, as # of Bedrooms increases [median of bar plot], price increases as well

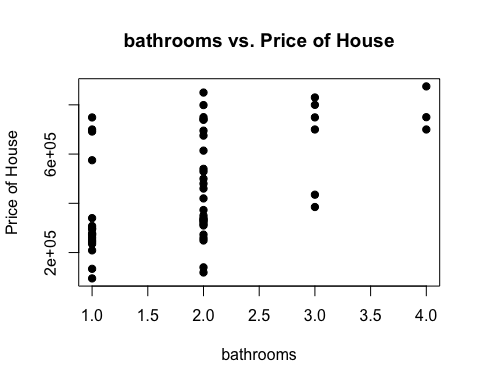
## Bathrooms vs. Bedrooms  
plot5 <- plot(bathrooms~bedrooms, data=house, col=(c("gold","darkgreen")),main="Bathrooms vs. Bedrooms", xlab="Bedrooms", ylab="Bathrooms")



plot5

## NULL

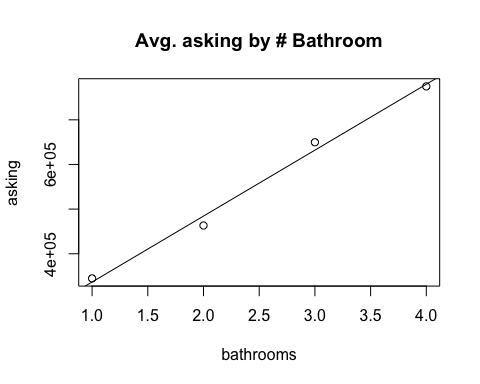
## Plots 1 shows the correlation between each variables and they are:  
# corr between asking vs bathrooms: 0.76  
# corr between asking vs bedrooms: 0.522  
  
## I want to use the predictor bathrooms for predicting house prices.  
  
plot(house$bathrooms,house$asking, main="bathrooms vs. Price of House", xlab="bathrooms", ylab="Price of House", pch=19)



## Ploting average prices in terms of the number of bathrooms and fit a linear model to this graph:  
  
average\_asking\_byBathrooms <-aggregate(asking~bathrooms, FUN=mean, data=house)  
plot(average\_asking\_byBathrooms,main="Avg. asking by # Bathroom")  
lin\_model\_bathroom<-lm(asking~bathrooms,data=average\_asking\_byBathrooms)  
summary(lin\_model\_bathroom)

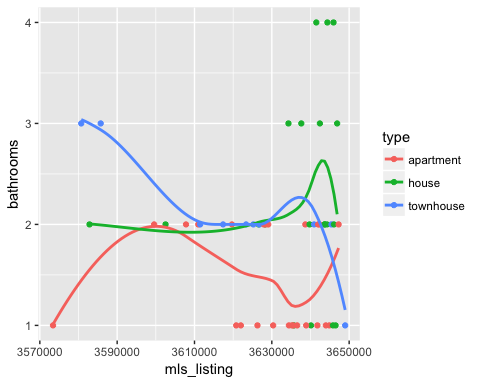
##   
## Call:  
## lm(formula = asking ~ bathrooms, data = average\_asking\_byBathrooms)  
##   
## Residuals:  
## 1 2 3 4   
## 8117 -21055 17758 -4821   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 188937 25216 7.493 0.01735 \*   
## bathrooms 147696 9208 16.041 0.00386 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 20590 on 2 degrees of freedom  
## Multiple R-squared: 0.9923, Adjusted R-squared: 0.9884   
## F-statistic: 257.3 on 1 and 2 DF, p-value: 0.003864

abline(lin\_model\_bathroom)

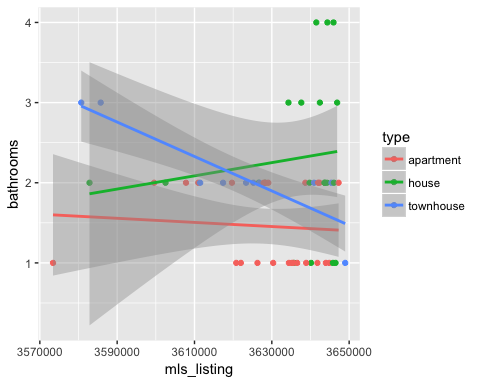


#compare the number of bedrooms between each type  
ggplot(house,aes(x=mls\_listing,y=bathrooms,colour=type))+geom\_point()+geom\_smooth(se=F)

## `geom\_smooth()` using method = 'loess'

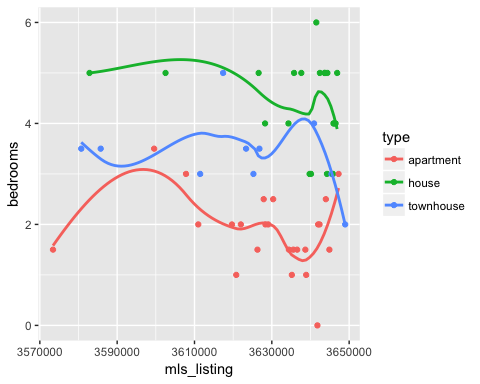


ggplot(house,aes(x=mls\_listing,y=bathrooms,colour=type))+geom\_point()+geom\_smooth(method = "lm")

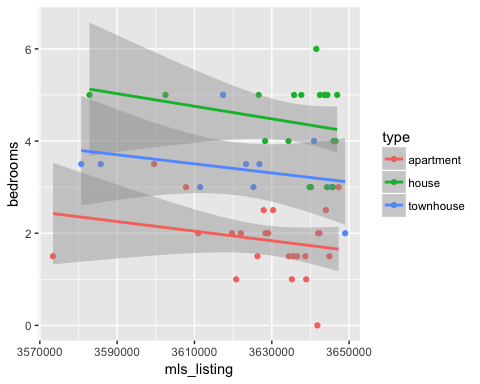


ggplot(house,aes(x=mls\_listing,y=bedrooms,colour=type))+geom\_point()+geom\_smooth(se=F)

## `geom\_smooth()` using method = 'loess'



ggplot(house,aes(x=mls\_listing,y=bedrooms,colour=type))+geom\_point()+geom\_smooth(method = "lm")

 As we see from above, there is no obvious pattern about the number of bathrooms for each kind of type. But clearly, houses normally own more bedrooms than townhouses, and the least number of bedrooms for apartments.

# Methodology

# We want to test whether the median of different house price would have the same median  
  
median\_test(house, asking, type)

## $table  
## above  
## group above below  
## apartment 2 22  
## house 20 0  
## townhouse 5 5  
##   
## $test  
## what value  
## 1 statistic 3.666667e+01  
## 2 df 2.000000e+00  
## 3 P-value 1.091276e-08

#The p-value is smaller than 0.05, therefore we reject the null hypothesis and conclude that the medians are not the same.

# Analysis and results

house

## # A tibble: 54 x 5  
## mls\_listing asking bedrooms bathrooms type  
## <dbl> <dbl> <dbl> <dbl> <chr>  
## 1 3573407 275000 1.5 1 apartment  
## 2 3580709 434999 3.5 3 townhouse  
## 3 3582848 675000 5.0 2 house  
## 4 3585741 385000 3.5 3 townhouse  
## 5 3599567 529000 3.5 2 apartment  
## 6 3602510 849888 5.0 2 house  
## 7 3607807 139900 3.0 2 apartment  
## 8 3610979 334900 2.0 2 apartment  
## 9 3611464 249000 3.0 2 townhouse  
## 10 3617396 259000 5.0 2 townhouse  
## # ... with 44 more rows

full = lm(asking~bedrooms+bathrooms+type, data=house)  
summary(full)

##   
## Call:  
## lm(formula = asking ~ bedrooms + bathrooms + type, data = house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -212389 -43612 10701 44363 189371   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 195990 36444 5.378 2.09e-06 \*\*\*  
## bedrooms 16480 17604 0.936 0.3538   
## bathrooms 42979 19571 2.196 0.0328 \*   
## typehouse 358710 48094 7.459 1.30e-09 \*\*\*  
## typetownhouse 5181 41345 0.125 0.9008   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 89550 on 49 degrees of freedom  
## Multiple R-squared: 0.8552, Adjusted R-squared: 0.8434   
## F-statistic: 72.37 on 4 and 49 DF, p-value: < 2.2e-16

hatvalues(full)

## 1 2 3 4 5 6   
## 0.04991388 0.13539208 0.07556384 0.13539208 0.12393034 0.07556384   
## 7 8 9 10 11 12   
## 0.08105631 0.05327299 0.10502620 0.20594113 0.05327299 0.06389778   
## 13 14 15 16 17 18   
## 0.05525156 0.10127256 0.10502620 0.04991388 0.07556384 0.10127256   
## 19 20 21 22 23 24   
## 0.05557818 0.05327299 0.07991082 0.09102348 0.04991388 0.06389778   
## 25 26 27 28 29 30   
## 0.17648830 0.04991388 0.07015237 0.06836370 0.11287883 0.13208972   
## 31 32 33 34 35 36   
## 0.11684051 0.17581231 0.05327299 0.05327299 0.07015237 0.07556384   
## 37 38 39 40 41 42   
## 0.07556384 0.07991082 0.11287883 0.16025389 0.04991388 0.10502620   
## 43 44 45 46 47 48   
## 0.13208972 0.22198178 0.05557818 0.11564585 0.07015237 0.08105631   
## 49 50 51 52 53 54   
## 0.17060651 0.14983030 0.04991388 0.05327299 0.06389778 0.05750386

2\*(3+1)/54

## [1] 0.1481481

drop1 = lm(asking~bathrooms+type, data=house)  
summary(drop1)

##   
## Call:  
## lm(formula = asking ~ bathrooms + type, data = house)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -198224 -51439 8621 41030 211776   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 213842 31018 6.894 8.79e-09 \*\*\*  
## bathrooms 51691 17195 3.006 0.00413 \*\*   
## typehouse 393333 30705 12.810 < 2e-16 \*\*\*  
## typetownhouse 25066 35427 0.708 0.48251   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 89440 on 50 degrees of freedom  
## Multiple R-squared: 0.8526, Adjusted R-squared: 0.8438   
## F-statistic: 96.44 on 3 and 50 DF, p-value: < 2.2e-16

anova(full,drop1)

## Analysis of Variance Table  
##   
## Model 1: asking ~ bedrooms + bathrooms + type  
## Model 2: asking ~ bathrooms + type  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 49 3.9298e+11   
## 2 50 4.0001e+11 -1 -7028315179 0.8763 0.3538

This is comparing the two models,one with all variables, and the other one with two variables. Those two R-squared are similar. And the most extreme value is 44, with a leverage of 0.22.

tmp\_house

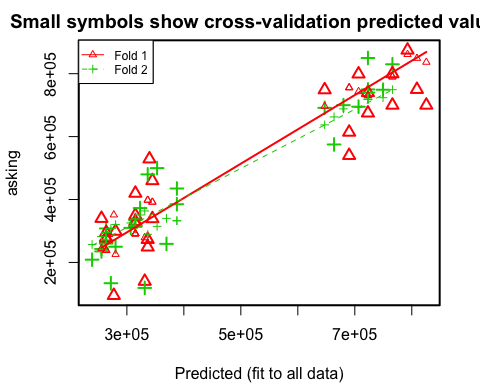
## # A tibble: 54 x 6  
## asking bedrooms bathrooms apartment house townhouse  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 275000 1.5 1 1 0 0  
## 2 434999 3.5 3 0 0 1  
## 3 675000 5.0 2 0 1 0  
## 4 385000 3.5 3 0 0 1  
## 5 529000 3.5 2 1 0 0  
## 6 849888 5.0 2 0 1 0  
## 7 139900 3.0 2 1 0 0  
## 8 334900 2.0 2 1 0 0  
## 9 249000 3.0 2 0 0 1  
## 10 259000 5.0 2 0 0 1  
## # ... with 44 more rows

full = lm(asking~., data=tmp\_house)  
cv.lm(data = as.data.frame(data.matrix(tmp\_house)), full, m=2)

## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from  
## a rank-deficient fit may be misleading  
  
## Warning in predict.lm(subs.lm, newdata = data[rows.out, ]): prediction from  
## a rank-deficient fit may be misleading

## Analysis of Variance Table  
##   
## Response: asking  
## Df Sum Sq Mean Sq F value Pr(>F)   
## bedrooms 1 1.57e+12 1.57e+12 195.50 < 2e-16 \*\*\*  
## bathrooms 1 8.18e+09 8.18e+09 1.02 0.3174   
## apartment 1 6.02e+10 6.02e+10 7.50 0.0086 \*\*   
## house 1 6.85e+11 6.85e+11 85.45 2.6e-12 \*\*\*  
## Residuals 49 3.93e+11 8.02e+09   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Warning in cv.lm(data = as.data.frame(data.matrix(tmp\_house)), full, m = 2):   
##   
## As there is >1 explanatory variable, cross-validation  
## predicted values for a fold are not a linear function  
## of corresponding overall predicted values. Lines that  
## are shown for the different folds are approximate



##   
## fold 1   
## Observations in test set: 27   
## 1 3 5 7 9 11 14 15  
## Predicted 263689 723059 339629 331389 336570 314909 344810 336570  
## cvpred 238169 730820 272548 278728 398008 291089 391828 398008  
## asking 275000 675000 529000 139900 249000 349999 339000 272900  
## CV residual 36831 -55820 256452 -138828 -149008 58910 -52828 -125108  
## 18 19 20 23 29 32 33 34  
## Predicted 344810 706579 314909 263689 690098 825497 314909 314909  
## cvpred 391828 743181 291089 238169 755542 836660 291089 291089  
## asking 459900 799000 329900 295000 614000 699900 325000 419900  
## CV residual 68072 55819 38811 56831 -141542 -136760 33911 128811  
## 35 36 38 39 40 41 43 44  
## Predicted 766038 723059 280170 690098 809017 263689 647119 792537  
## cvpred 789921 730820 225809 755542 849021 238169 696442 861381  
## asking 699900 739900 299000 540000 749900 278000 748800 874900  
## CV residual -90021 9080 73191 -215542 -99121 39831 52358 13519  
## 47 49 53  
## Predicted 766038 277111 255449  
## cvpred 789921 351269 244350  
## asking 799900 95000 339900  
## CV residual 9979 -256269 95550  
##   
## Sum of squares = 3.52e+11 Mean square = 1.3e+10 n = 27   
##   
## fold 2   
## Observations in test set: 27   
## 2 4 6 8 10 12 13 16  
## Predicted 387789 387789 723059 314909 369530 255449 271930 263689  
## cvpred 332504 332504 718673 338231 340097 282285 307701 294993  
## asking 434999 385000 849888 334900 259000 243000 133900 308000  
## CV residual 102495 52496 131215 -3331 -81097 -39285 -173801 13007  
## 17 21 22 24 25 26 27 28 30  
## Predicted 723059 280170 749558 255449 680080 263689 766038 306669 647119  
## cvpred 718673 320409 723787 282285 688142 294993 749203 325523 637311  
## asking 750000 249900 749000 235000 699900 269900 829900 309900 691500  
## CV residual 31327 -70509 25213 -47285 11758 -25093 80697 -15623 54189  
## 31 37 42 45 46 48 50 51  
## Predicted 353050 723059 336570 706579 663600 331389 238969 263689  
## cvpred 314682 718673 289266 693257 662727 363647 256870 294993  
## asking 499900 739900 479900 695000 575000 119000 208800 259900  
## CV residual 185218 21227 190634 1743 -87727 -244647 -48070 -35093  
## 52 54  
## Predicted 314909 323149  
## cvpred 338231 350939  
## asking 315800 372800  
## CV residual -22431 21861  
##   
## Sum of squares = 2.31e+11 Mean square = 8.57e+09 n = 27   
##   
## Overall (Sum over all 27 folds)   
## ms   
## 1.08e+10

We can see that with cross validation, we get an approximately good error result

# Conclusions

The results of the study of the relationship between the asking price of the properties and numbers of bedrooms and bathrooms indicates that additional bedrooms or bathrooms would cause an increasing in the asking price. Among the three types of properties, house is the most expensive which is about 750000 and apartment is the least expensive which is around 350000 around the Scarborough area. Although the asking price could be told by the number of bedrooms and bathrooms the property has, it is still hard to give a precise prediction on the asking price due to many other features.

# Citation

www.kaggle.com/prabhats/linear-regression-on-house-price www.kaggle.com/auygur/step-by-step-house-price-prediction-r-2-0-77/code www.kaggle.com/amitdhakre13/eda-linear-regression-k-fold-cv-adj-r2-0-87