Brooklyn Property Sales Data

Brooklyn Building sales data can be found here: <http://aquarius.tw.rpi.edu/html/DA/>  
Save "rollingsales\_brooklyn.xls"

Let's start by importing the data:

bk <- read.csv("rollingsales\_brooklyn.csv", header = T,   
 na.strings = c(""," "," "," "," "," "," "," "," ",  
 " "," ",  
 " ",  
 " ",  
 "NA"),stringsAsFactors = FALSE) #import data and make white space NA's

Look at the data:

summary(bk) #used to figure out format of fields

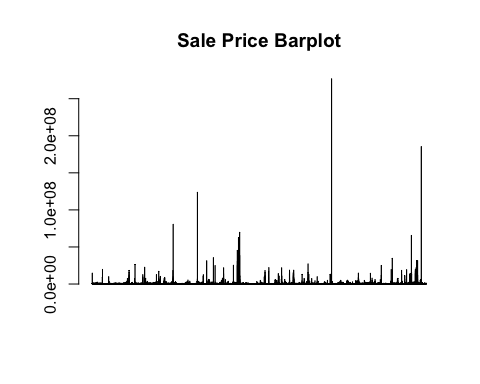
BOROUGH NEIGHBORHOOD BUILDING.CLASS.CATEGORY  
 Min. :3 Length:23373 Length:23373   
 1st Qu.:3 Class :character Class :character   
 Median :3 Mode :character Mode :character   
 Mean :3   
 3rd Qu.:3   
 Max. :3   
 TAX.CLASS.AT.PRESENT BLOCK LOT EASE.MENT   
 Length:23373 Min. : 20 Min. : 1.0 Mode:logical   
 Class :character 1st Qu.:1638 1st Qu.: 22.0 NA's:23373   
 Mode :character Median :3839 Median : 48.0   
 Mean :3984 Mean : 305.4   
 3rd Qu.:6259 3rd Qu.: 142.0   
 Max. :8955 Max. :9039.0   
 BUILDING.CLASS.AT.PRESENT ADDRESS APART.MENT.NUMBER   
 Length:23373 Length:23373 Length:23373   
 Class :character Class :character Class :character   
 Mode :character Mode :character Mode :character   
   
   
   
 ZIP.CODE RESIDENTIAL.UNITS COMMERCIAL.UNITS TOTAL.UNITS   
 Min. : 0 Min. : 0.000 Min. : 0.0000 Min. : 0.00   
 1st Qu.:11209 1st Qu.: 1.000 1st Qu.: 0.0000 1st Qu.: 1.00   
 Median :11218 Median : 1.000 Median : 0.0000 Median : 1.00   
 Mean :11211 Mean : 2.156 Mean : 0.1973 Mean : 2.37   
 3rd Qu.:11230 3rd Qu.: 2.000 3rd Qu.: 0.0000 3rd Qu.: 2.00   
 Max. :11416 Max. :509.000 Max. :222.0000 Max. :509.00   
 LAND.SQUARE.FEET GROSS.SQUARE.FEET YEAR.BUILT   
 Length:23373 Length:23373 Min. : 0   
 Class :character Class :character 1st Qu.:1901   
 Mode :character Mode :character Median :1925   
 Mean :1681   
 3rd Qu.:1950   
 Max. :2013   
 TAX.CLASS.AT.TIME.OF.SALE BUILDING.CLASS.AT.TIME.OF.SALE  
 Min. :1.000 Length:23373   
 1st Qu.:1.000 Class :character   
 Median :1.000 Mode :character   
 Mean :1.705   
 3rd Qu.:2.000   
 Max. :4.000   
 SALE.PRICE SALE.DATE   
 Length:23373 Length:23373   
 Class :character Class :character   
 Mode :character Mode :character

Clean the data:

bk$SALE.PRICE <- gsub("\\$","", bk$SALE.PRICE) #filter out $ signs  
bk$SALE.PRICE <- gsub(",","", bk$SALE.PRICE)   
bk$LAND.SQUARE.FEET <- gsub(",","", bk$LAND.SQUARE.FEET)   
bk$GROSS.SQUARE.FEET <- gsub(",","", bk$GROSS.SQUARE.FEET) #filter out commas  
bk$SALE.PRICE <- as.numeric(bk$SALE.PRICE)  
bk$GROSS.SQUARE.FEET <- as.numeric(bk$GROSS.SQUARE.FEET)  
bk$LAND.SQUARE.FEET <- as.numeric(bk$LAND.SQUARE.FEET) #make fields numeric  
bk$ZIP.CODE <- as.character(bk$ZIP.CODE) #make field character  
bk$NEIGHBORHOOD <- as.factor(bk$NEIGHBORHOOD)   
bk$BUILDING.CLASS.CATEGORY <- as.factor(bk$BUILDING.CLASS.CATEGORY) #make fields factor  
bk <- bk[which(bk$SALE.PRICE > 10000),] #filter out sale prices of less than 10,000  
bk <- subset(bk, select = -c(1)) #drop borough column

We will predict the sale price later but first let's visualize it:

barplot(bk$SALE.PRICE, main="Sale Price Barplot") #bar plot of sale price



Split the data into training and testing sets:

sub <- sample(nrow(bk), floor(nrow(bk) \* 0.90))  
bktrain <- bk[sub, ]  
bktest <- bk[-sub, ] #create train and test data frames

We will attach the train data frame. This allows column names to be referenced without mentioning the train data frame for each column:

attach(bktrain)

Create our a multiple linear regression model to predict the sale price:

bkreg2<- lm(formula = SALE.PRICE ~ LAND.SQUARE.FEET + GROSS.SQUARE.FEET+  
 YEAR.BUILT+ BLOCK, data = bktrain) #multiple linear regression model

Examine how the model performed:

summary(bkreg2)

Call:  
 lm(formula = SALE.PRICE ~ LAND.SQUARE.FEET + GROSS.SQUARE.FEET +   
 YEAR.BUILT + BLOCK, data = bktrain)  
   
 Residuals:  
 Min 1Q Median 3Q Max   
 -32934947 -396204 -146828 91527 184053725   
   
 Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
 (Intercept) 1.131e+06 7.113e+04 15.899 < 2e-16 \*\*\*  
 LAND.SQUARE.FEET 7.157e+00 3.918e-01 18.268 < 2e-16 \*\*\*  
 GROSS.SQUARE.FEET 1.085e+02 9.092e-01 119.317 < 2e-16 \*\*\*  
 YEAR.BUILT -1.878e+02 3.733e+01 -5.031 4.94e-07 \*\*\*  
 BLOCK -6.255e+01 8.305e+00 -7.532 5.34e-14 \*\*\*  
 ---  
 Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
   
 Residual standard error: 2460000 on 12703 degrees of freedom  
 Multiple R-squared: 0.5537, Adjusted R-squared: 0.5536   
 F-statistic: 3940 on 4 and 12703 DF, p-value: < 2.2e-16

The Multiple R-squared of 0.5477 indicates that this model explains 54.77% of the variability of the predicted data around the mean. The results also show that all four independent variables are siginificant at an alpha of .001.

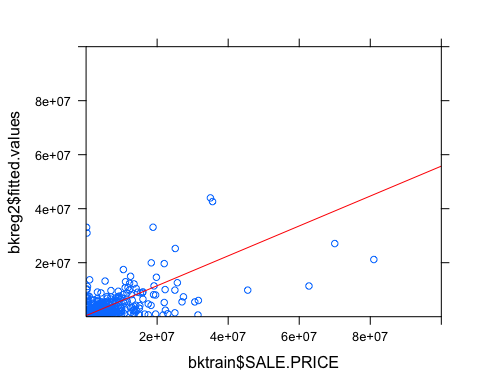
Before we visualize the model we will load a package needed to plot the data:

require("lattice")

Loading required package: lattice

Plot the fitted values by the actual sale price of the train data frame:

xyplot(bkreg2$fitted.values~bktrain$SALE.PRICE , data = bktrain, type = c("p","r"),   
 col.line = "red", xlim = c(1,1.0e+08), ylim = c(1,1.0e+08)) #xy plot



Predict how the model does on the test data:

bkreg2final <- predict(bkreg2, newdata = bktest) #use model on test data

Add a new column onto the test data frame that includes the new predicted values:

bktest$Predict <-c(bkreg2final) #add new column

Hope you enjoyed the read, and thanks for visiting my site!