271 Final

Glenn (Ted) Dunmire, Marlea Gwinn, Julian Phillips

December 7, 2015

```r
#Set Directory

#Ted
setwd("~/Documents/271 Final")

#Marlea
#setwd("C://Users/gwina003/Downloads/Final")

#Julian
#data <- read.csv("//vivica/Documents/MIDS/W271/271-Final/houseValueData.csv")

#Load Relevant Libraries
library(ggplot2)
library(car)
library(reshape2)
library(grid)
library(astsa)
library(forecast)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: timeDate
## This is forecast 6.2
##
##
## Attaching package: 'forecast'
##
## The following object is masked from 'package:astsa':
##
##     gas
```

```r
library(quantmod)
```

```
## Loading required package: xts
## Loading required package: TTR
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
library(fGarch)
```

```
## Loading required package: timeSeries
##
## Attaching package: 'timeSeries'
##
## The following object is masked from 'package:zoo':
##
##     time<-
##
## Loading required package: fBasics
##
##
## Rmetrics Package fBasics
## Analysing Markets and calculating Basic Statistics
## Copyright (C) 2005-2014 Rmetrics Association Zurich
## Educational Software for Financial Engineering and Computational Science
## Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.
## https://www.rmetrics.org --- Mail to: info@rmetrics.org
##
## Attaching package: 'fBasics'
##
## The following object is masked from 'package:TTR':
##
##     volatility
##
## The following object is masked from 'package:astsa':
##
##     nyse
##
## The following object is masked from 'package:car':
##
##     densityPlot
```

```r
library(tseries)
library(gridExtra)
library(scales)
library(plyr)
library(GGally)
library(sandwich)
library(lmtest)
```

Question 1

Analyze each of these variables (as well as a combination of them) very carefully and use them (or a subset of them) to build a model and test hypotheses to address the questions. Also address potential (statistical) issues that may be casued by omitted variables. The philanthropist group hires a think tank to examine the relationship between the house values and neighborhood characteristics. For instance, they are interested in the extent to which houses in neighbhorhood with desirable features command higher values. They are specifically interested in environmental features, such as proximity to water body (i.e. lake, river, or ocean) or air quality.

```r
#Read dataset
data <- read.csv("houseValueData.csv")
data$withWater <- as.factor(data$withWater) #changed to factor based on documentation
```

Let us first begin with some basic examination of the data to see what kinds of variables we have and what their distributions look like. We have turned the withWater variable into a factor based on the documentation, because it is a categorical variable rather than an int.

```r
str(data)
```

```
## 'data.frame':    400 obs. of  11 variables:
##  $ crimeRate_pc     : num  37.6619 0.5783 0.0429 22.5971 0.0664 ...
##  $ nonRetailBusiness: num  0.181 0.0397 0.1504 0.181 0.0405 ...
##  $ withWater        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ageHouse         : num  78.7 67 77.3 89.5 74.4 71.3 68.2 97.3 92.2 96.2 ...
##  $ distanceToCity   : num  2.71 4.12 7.82 1.95 5.54 ...
##  $ distanceToHighway: int  24 5 4 24 5 5 5 5 3 5 ...
##  $ pupilTeacherRatio: num  23.2 16 21.2 23.2 19.6 23.9 22.2 17.7 20.8 17.7 ...
##  $ pctLowIncome     : int  18 9 13 41 8 9 12 18 5 4 ...
##  $ homeValue        : int  245250 1125000 463500 166500 672750 596250 425250 483750 852750 1125000 .
##  $ pollutionIndex   : num  52.9 42.5 31.4 55 36 37 34.9 72.1 33.8 45.5 ...
##  $ nBedRooms        : num  4.2 6.3 4.25 3 4.86 ...
```

```r
sum(is.na(data))
```

```
## [1] 0
```

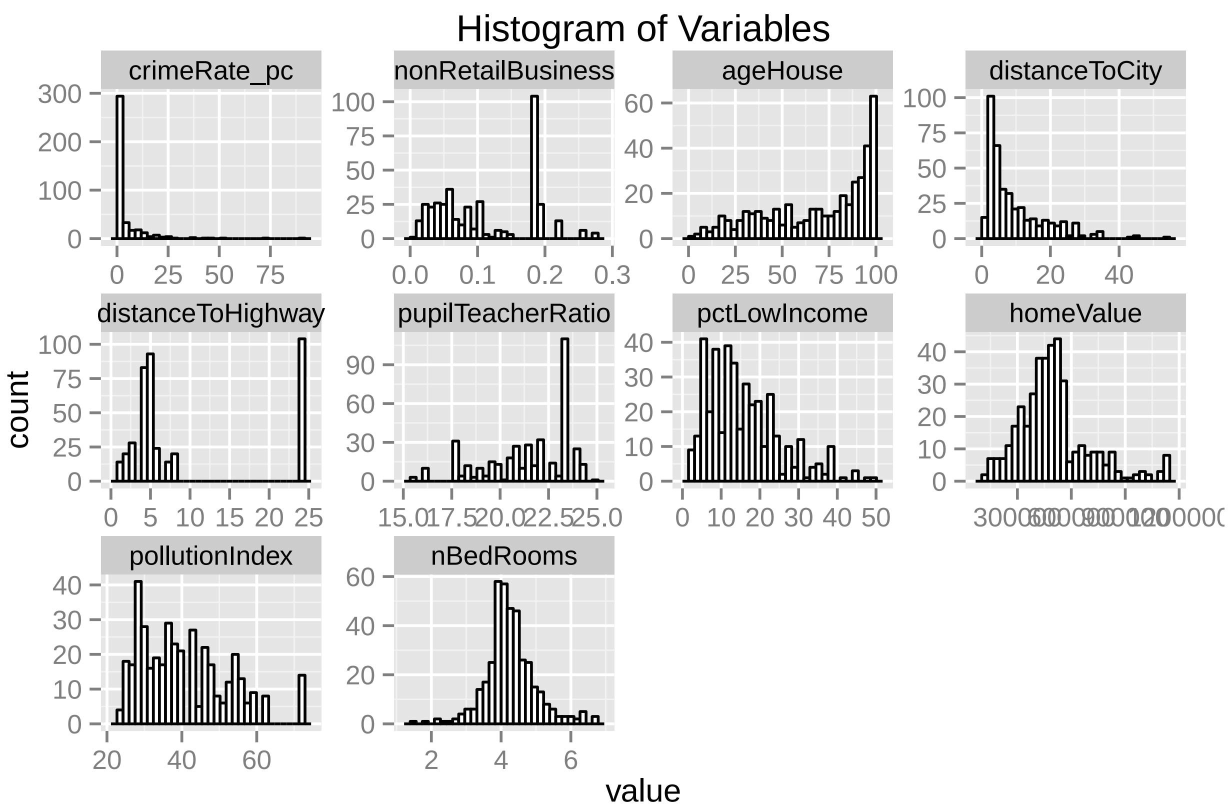
```r
summary(data)
```

```
##   crimeRate_pc      nonRetailBusiness withWater    ageHouse     
##  Min.   : 0.00632   Min.   :0.0074    0:373     Min.   :  2.90  
##  1st Qu.: 0.08260   1st Qu.:0.0513    1: 27     1st Qu.: 45.67  
##  Median : 0.26600   Median :0.0969              Median : 77.95  
##  Mean   : 3.76256   Mean   :0.1115              Mean   : 68.93  
##  3rd Qu.: 3.67481   3rd Qu.:0.1810              3rd Qu.: 94.15  
##  Max.   :88.97620   Max.   :0.2774              Max.   :100.00  
##  distanceToCity   distanceToHighway pupilTeacherRatio  pctLowIncome  
##  Min.   : 1.228   Min.   : 1.000    Min.   :15.60     Min.   : 2.00  
##  1st Qu.: 3.240   1st Qu.: 4.000    1st Qu.:19.90     1st Qu.: 8.00  
##  Median : 6.115   Median : 5.000    Median :21.90     Median :14.00  
##  Mean   : 9.638   Mean   : 9.582    Mean   :21.39     Mean   :15.79  
##  3rd Qu.:13.628   3rd Qu.:24.000    3rd Qu.:23.20     3rd Qu.:21.00  
##  Max.   :54.197   Max.   :24.000    Max.   :25.00     Max.   :49.00  
##    homeValue        pollutionIndex    nBedRooms    
##  Min.   : 112500   Min.   :23.50   Min.   :1.561  
##  1st Qu.: 384188   1st Qu.:29.88   1st Qu.:3.883  
##  Median : 477000   Median :38.80   Median :4.193  
##  Mean   : 499584   Mean   :40.61   Mean   :4.266  
##  3rd Qu.: 558000   3rd Qu.:47.58   3rd Qu.:4.582  
##  Max.   :1125000   Max.   :72.10   Max.   :6.780  
```

Notice we have 400 observations of 11 variables, with no missing values. Here is a histogram with all variables in the same plot.

```r
ggplot(melt(data[,-3]), aes(value)) + geom_histogram(color = "black", fill = "white") + facet_wrap(~vari
```

```
## No id variables; using all as measure variables
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



The following function will be used to get detailed summary statistics for each continuous variable:

```r
ContStat = function(x,y) {
#x must be a vector, not a dataframe
#y is the number of decimal points to round data to
StatLen  = length(x)
StatNA = sum(is.na(x))
StatMean = summary(x)["Mean"]
StatMin  = summary(x)["Min."]
StatMax= summary(x)["Max."]
StatSd  = sd(x)
```

```
StatQuan = quantile(x,c(0.01,0.05,0.1,0.25,0.5,0.75,0.9,0.95,0.99))

rownms =c("N", "#NA's","Mean","Min","Max","Std", "1%","5%","10%","25%","50%","75%","90%","95%","99%")

Stats = c(StatLen,StatNA,StatMean,StatMin,StatMax,StatSd, StatQuan)

ContStatDF = as.data.frame(Stats, row.names=rownms)
ContStatDF = round(ContStatDF,y)
return(ContStatDF)
}
```

The following function will be used to output a histogram and a scatterplot:

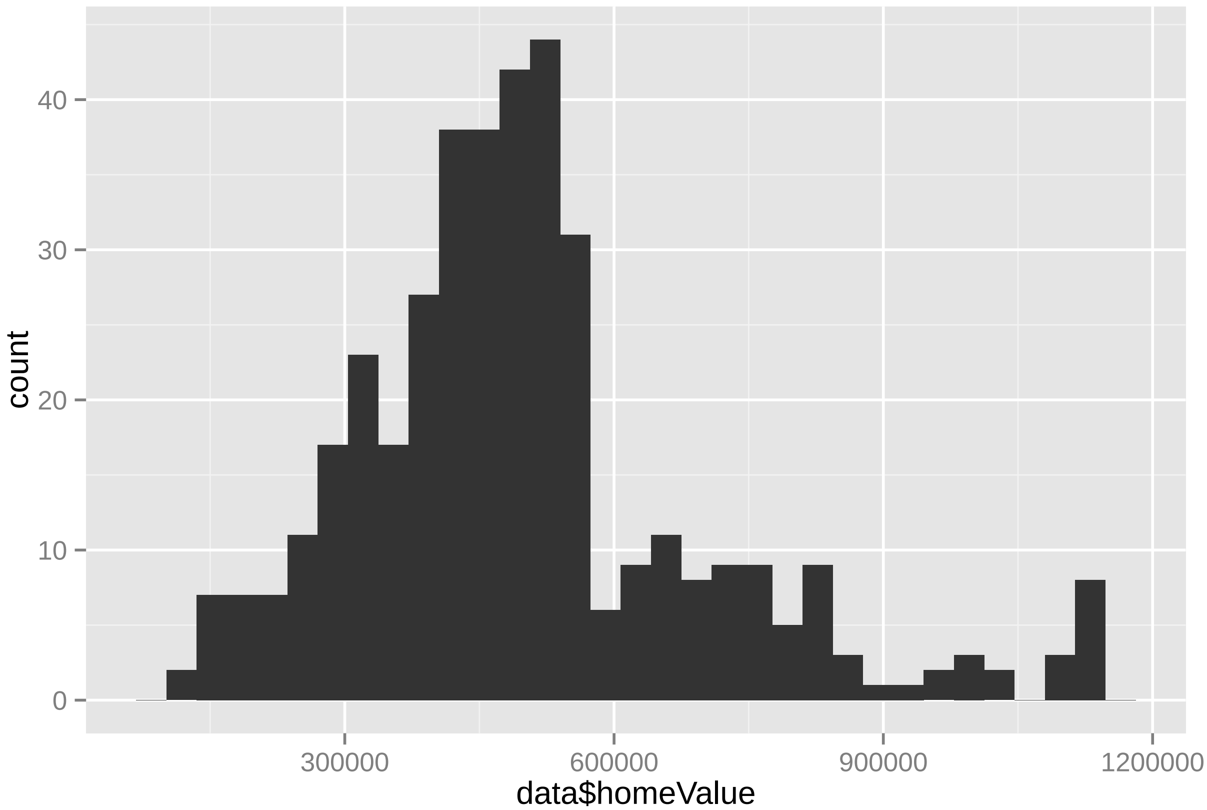
```
Graphs = function(x, y) {
#vect must be a vector, not a dataframe
#y is a string, the name of the variable of interest.  Used for labeling the graphs

subdata = data[,c(x,'homeValue')]
names(subdata)[1] = 'variable'

hist =  ggplot(data=subdata, aes(variable)) + geom_histogram() + ggtitle("Histogram")+ scale_x_continuou

sp = ggplot(data=subdata, aes(x=variable, y=homeValue)) +  geom_point(shape=16) +  ggtitle("Scatterplot

output = grid.arrange(hist, sp, ncol=2,nrow=1, top = textGrob(paste("Histogram and Scatterplot of" , y

return(output)
}
```

Now let's take a closer look at each of the variables and its relationship with the variable of interest, homeValue.

First, homeValue itself. From the attached text file, homeValue is defined as *median price of single-family house in the neighborhood (measured in dollars).*

```
ggplot(data=data, aes(data$homeValue)) + geom_histogram()
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
ContStat(data$homeValue,0)
```

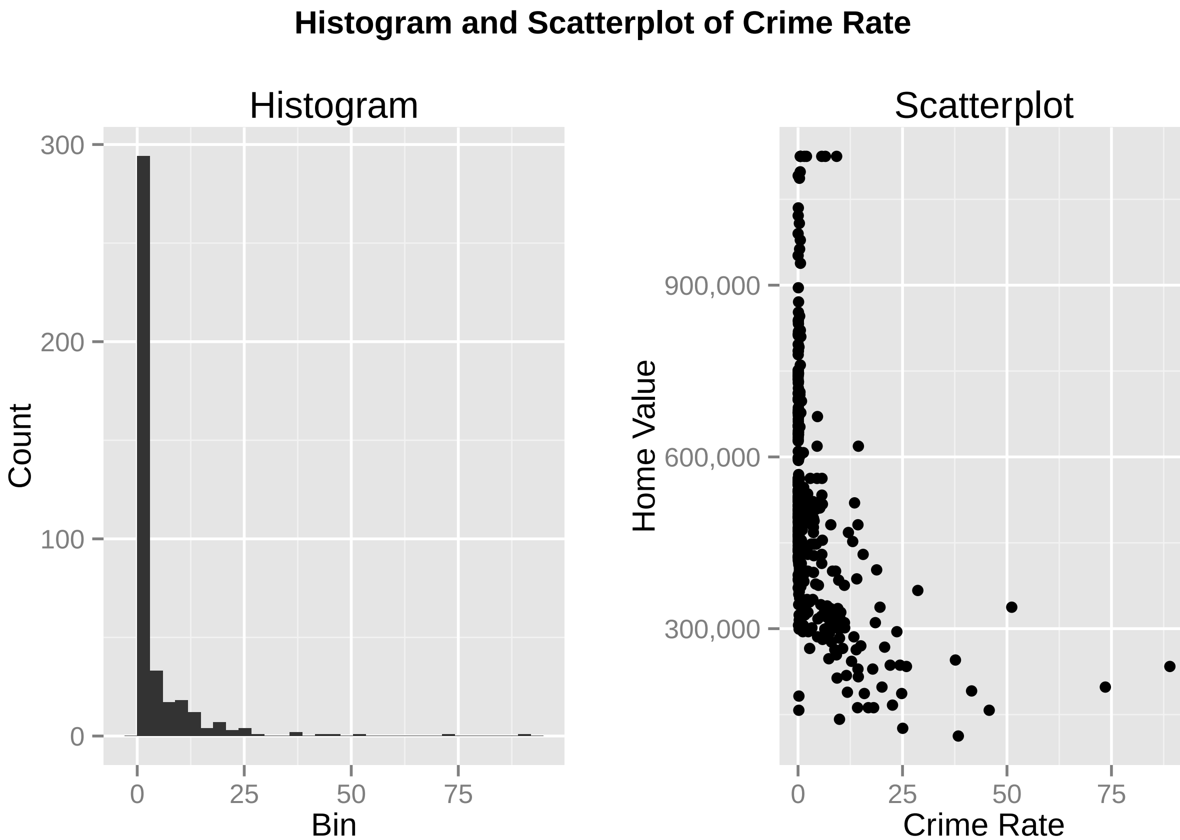
```
##          Stats
## N          400
## #NA's        0
## Mean    499600
## Min     112500
## Max    1125000
## Std     196116
## 1%      157500
## 5%      229500
## 10%     291825
## 25%     384188
## 50%     477000
## 75%     558000
## 90%     749475
## 95%     871987
## 99%    1125000
```

The range of the variable is 112,500 through 1,125,000 There dont appear to be any values that are unreasonable for the homeValue variable. The histogram shows a strong right skew of the variable with many of the values clustered together between the first and third quartile. While this is the target variable of interest, I will also create a log (homeValue) price and use both of them to find the model with the best fit.

Next lets take a look at the crimeRatepc variable which is defined as *crime rate per capita, measured by number of crimes per 1000 residents in neighborhood.*

```r
Graphs('crimeRate_pc', 'Crime Rate')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z     cells    name                grob
## 1 1 (2-2,1-1) arrange      gtable[layout]
## 2 2 (2-2,2-2) arrange      gtable[layout]
## 3 3 (1-1,1-2) arrange text[GRID.text.543]
```

```r
ContStat(data$crimeRate_pc,2)
```

```
##         Stats
## N      400.00
## #NA's    0.00
## Mean     3.76
## Min      0.01
## Max     88.98
## Std      8.87
## 1%       0.01
## 5%       0.03
## 10%      0.04
## 25%      0.08
## 50%      0.27
```

```
## 75%     3.67
## 90%    11.20
## 95%    18.11
## 99%    41.57
```

Crime rate per capita shows a slight negative correlation against Home Value. However, there is an extremely large number of neighborhoods that have a crime rate of zero or close to zero. The scatterplot shows that crime rate is more dispersed around areas of lower home value. That being said, there appears to be a small ceiling in the scatterplot- six points that all seem to have the same home value but with varying crime rates. Lets take a closer look at those points.

```r
subset(data, homeValue>1100000)
```

```
##     crimeRate_pc nonRetailBusiness withWater ageHouse distanceToCity
## 2        0.57834            0.0397         0     67.0       4.116839
## 10       2.01019            0.1958         0     96.2       3.143511
## 18       6.53876            0.1810         1     97.5       1.343007
## 69       5.66998            0.1810         1     96.8       1.629199
## 164      1.51902            0.1958         1     93.9       3.433753
## 172      0.52693            0.0620         0     83.0       5.476381
## 216      0.61154            0.0397         0     86.9       2.563433
## 370      9.23230            0.1810         0    100.0       1.283993
##     distanceToHighway pupilTeacherRatio pctLowIncome homeValue
## 2                   5              16.0            9   1125000
## 10                  5              17.7            4   1125000
## 18                 24              23.2            3   1125000
## 69                 24              23.2            4   1125000
## 164                 5              17.7            4   1125000
## 172                 8              20.4            5   1125000
## 216                 5              16.0            6   1125000
## 370                24              23.2           12   1125000
##     pollutionIndex nBedRooms
## 2             42.5     6.297
## 10            45.5     5.929
## 18            48.1     5.016
## 69            48.1     4.683
## 164           45.5     6.375
## 172           35.4     6.725
## 216           49.7     6.704
## 370           48.1     4.216
```

Unexpectedly there seems to be a maximum limit on the homeValue. These values probably represent that median price being greater than 1125000 which means that the model will not be able to accurately predict points that great since these rows could have a true median homeValue of 1125000 or even ten or fifty times that value. Having identified this ceiling, we should check to see if there is a floor for the minimum home value.

```r
subset(data, homeValue<126000)
```

```
##     crimeRate_pc nonRetailBusiness withWater ageHouse distanceToCity
## 342      38.3518             0.181         0      100       1.891958
##     distanceToHighway pupilTeacherRatio pctLowIncome homeValue
```

```
## 342                 24              23.2           39    112500
##     pollutionIndex nBedRooms
## 342           54.3     3.453
```

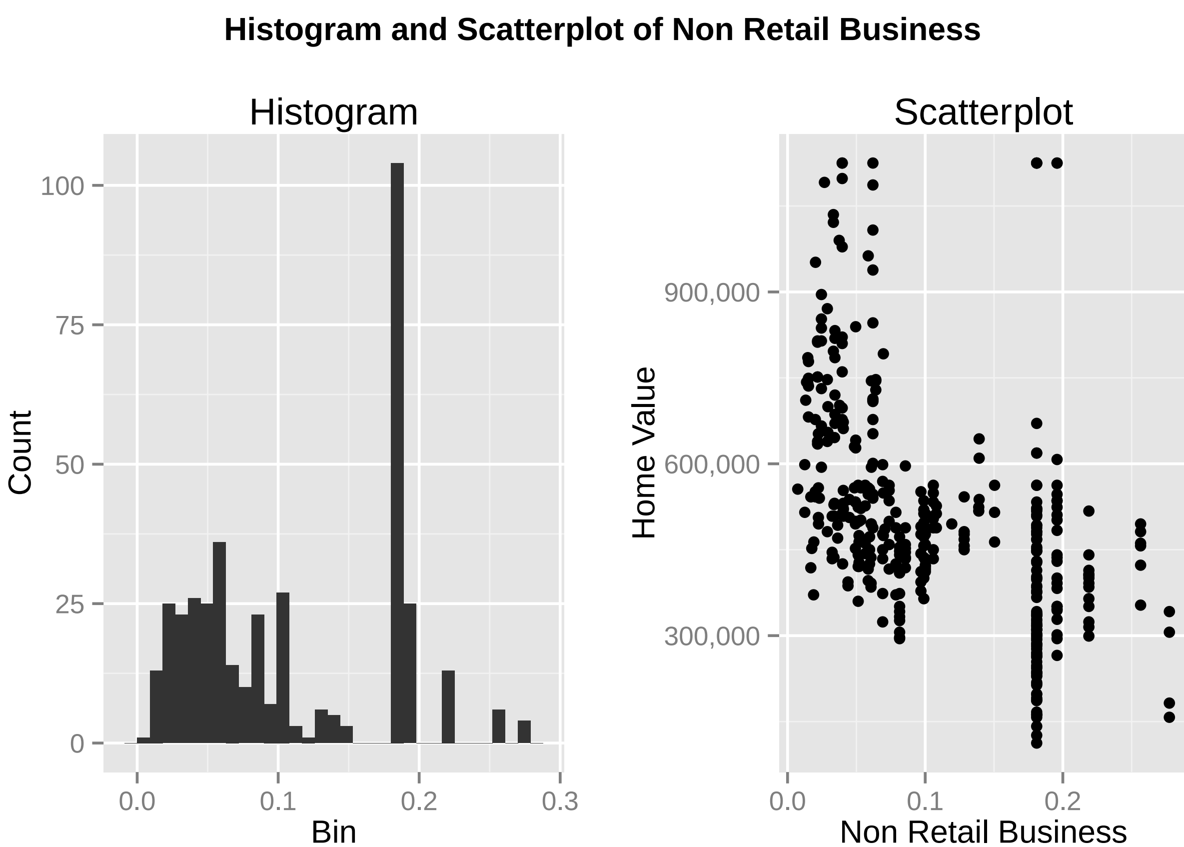
With only one value at the minimum, it seems unlikely that there is a minimum limit to the home value.

A transformation for crime rate may be desired later, but all the other variables will be examined first. Additionally, a decision on the steps to take with the maximum home value will be decided at the end of the variable examination.

Next let's take a look at the nonRetailBusiness variable which is defined as *the proportion of non-retail business acres per neighborhood.*

```
Graphs('nonRetailBusiness', 'Non Retail Business')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z     cells    name                grob
## 1 1 (2-2,1-1) arrange      gtable[layout]
## 2 2 (2-2,2-2) arrange      gtable[layout]
## 3 3 (1-1,1-2) arrange text[GRID.text.629]
```

```r
ContStat(data$nonRetailBusiness,2)
```

```
##        Stats
## N     400.00
## #NA's   0.00
## Mean    0.11
## Min     0.01
## Max     0.28
## Std     0.07
## 1%      0.01
## 5%      0.02
## 10%     0.03
## 25%     0.05
## 50%     0.10
## 75%     0.18
## 90%     0.20
## 95%     0.22
## 99%     0.26
```

The range for non retail business is 0.01 through 0.28. There is also a negative correlation between the percentage of non retail business and the home value. While the data on the left side of the scatterplot seems to be random according to Non Retail Business, on the right side of the scatterplot, they are lining up. Let's take a deeper look. Let's find the most common values for this variable.

```r
freqs = count(data$nonRetailBusiness)
freqs[with(freqs,order(-freq)),]
```

```
##         x freq
## 66 0.1810  104
## 67 0.1958   25
## 54 0.0814   15
## 47 0.0620   14
## 68 0.2189   13
## 30 0.0397   10
## 57 0.0990   10
## 58 0.1001    9
## 55 0.0856    8
## 59 0.1059    8
## 18 0.0246    7
## 32 0.0405    7
## 42 0.0586    7
## 49 0.0691    7
## 56 0.0969    7
## 14 0.0218    6
## 26 0.0344    6
## 38 0.0513    6
## 39 0.0519    6
## 52 0.0738    6
## 62 0.1283    6
## 69 0.2565    6
## 36 0.0493    5
## 6  0.0152    4
```

```
## 20 0.0289    4
## 41 0.0564    4
## 43 0.0596    4
## 53 0.0787    4
## 70 0.2774    4
## 15 0.0224    3
## 22 0.0324    3
## 23 0.0333    3
## 25 0.0341    3
## 37 0.0495    3
## 40 0.0532    3
## 46 0.0609    3
## 48 0.0641    3
## 50 0.0696    3
## 60 0.1081    3
## 64 0.1392    3
## 65 0.1504    3
## 2  0.0125    2
## 5  0.0147    2
## 7  0.0169    2
## 13 0.0203    2
## 21 0.0293    2
## 24 0.0337    2
## 27 0.0364    2
## 33 0.0439    2
## 34 0.0449    2
## 35 0.0486    2
## 44 0.0606    2
## 45 0.0607    2
## 63 0.1389    2
## 1  0.0074    1
## 3  0.0132    1
## 4  0.0138    1
## 8  0.0176    1
## 9  0.0189    1
## 10 0.0191    1
## 11 0.0201    1
## 12 0.0202    1
## 16 0.0225    1
## 17 0.0231    1
## 19 0.0268    1
## 28 0.0375    1
## 29 0.0378    1
## 31 0.0400    1
## 51 0.0707    1
## 61 0.1193    1
```

There are 104 records here (over 25%!) with the same value of 0.1810 for the percentage of non retail business. This is a curious result. Lets examine these records in detail.

```
subset(data, nonRetailBusiness==.181)
```

```
##     crimeRate_pc nonRetailBusiness withWater ageHouse distanceToCity
```

```
## 1       37.66190              0.181          0     78.7         2.705847
## 4       22.59710              0.181          0     89.5         1.950823
## 18       6.53876              0.181          1     97.5         1.343007
## 22      11.81230              0.181          0     76.5         2.547510
## 26      19.60910              0.181          0     97.9         1.552272
## 32       9.33889              0.181          0     95.6         2.954682
## 33       8.05579              0.181          0     95.4         4.139166
## 40       8.64476              0.181          0     92.6         2.541151
## 42       4.64689              0.181          0     67.6         4.423733
## 54       5.44114              0.181          0     98.2         3.937717
## 68      11.16040              0.181          0     94.6         3.339460
## 69       5.66998              0.181          1     96.8         1.629199
## 71       5.66637              0.181          0    100.0         3.043082
## 72      13.52220              0.181          0    100.0         1.934814
## 73       5.87205              0.181          0     96.0         2.286494
## 78      88.97620              0.181          0     91.9         1.745607
## 81      13.07510              0.181          0     56.7         5.263925
## 89      18.49820              0.181          0    100.0         1.228052
## 91       5.73116              0.181          0     77.0         7.120808
## 96       8.20058              0.181          0     80.3         5.131823
## 99      14.23620              0.181          0    100.0         2.066578
## 102     12.04820              0.181          0     87.6         2.913955
## 121     10.67180              0.181          0     94.8         3.002142
## 131      3.77498              0.181          0     84.7         5.407221
## 139      4.26131              0.181          0     81.3         4.357413
## 141      3.16360              0.181          0     48.2         6.006601
## 144      9.51363              0.181          0     94.1         4.321347
## 150      5.20177              0.181          1     83.4         4.965919
## 157     14.33370              0.181          0     88.0         2.913955
## 158      3.67822              0.181          0     96.2         3.286556
## 159     15.57570              0.181          0     71.0         5.518825
## 165      3.56868              0.181          0     75.0         5.482740
## 166      7.02259              0.181          0     95.3         2.733089
## 170     15.02340              0.181          0     97.3         3.279310
## 171      4.22239              0.181          1     89.0         2.803642
## 173     10.23300              0.181          0     96.7         3.455379
## 175     12.80230              0.181          0     96.6         2.782241
## 180     45.74610              0.181          0    100.0         2.246048
## 181      4.54192              0.181          0     88.0         4.382726
## 183      7.83932              0.181          0     65.4         5.686754
## 186     14.43830              0.181          0    100.0         1.843221
## 188     16.81180              0.181          0     98.1         1.764574
## 189      6.28807              0.181          0     96.4         3.207921
## 192      5.29305              0.181          0     82.5         3.448504
## 196      4.66883              0.181          0     87.9         4.557777
## 197      8.98296              0.181          1     97.4         3.333175
## 201     14.33370              0.181          0    100.0         2.099021
## 204      3.84970              0.181          1     91.0         4.346582
## 206      9.32909              0.181          0     98.7         3.690331
## 210      7.99248              0.181          0    100.0         1.981129
## 213     25.04610              0.181          0    100.0         2.097543
## 214     11.10810              0.181          0    100.0         1.292967
## 218     28.65580              0.181          0    100.0         2.098810
## 220      5.82401              0.181          0     64.7         7.166294
```

```
## 226      7.05042             0.181         0     85.1       3.084474
## 233     23.64820             0.181         0     96.2       1.686053
## 234      4.75237             0.181         0     86.5       4.155532
## 235     18.08460             0.181         0    100.0       2.640609
## 237      9.91655             0.181         0     77.8       1.913953
## 242     11.08740             0.181         0    100.0       2.696557
## 245      9.18702             0.181         0    100.0       2.079827
## 248      9.82349             0.181         0     98.8       1.631697
## 249     18.81100             0.181         0    100.0       2.024309
## 252      5.58107             0.181         0     87.9       3.832849
## 254      3.69311             0.181         0     88.4       4.519688
## 255     17.86670             0.181         0    100.0       1.686053
## 257      8.49213             0.181         0     86.1       3.160245
## 275      8.24809             0.181         0     99.3       4.201759
## 276     25.94060             0.181         0     89.1       2.222904
## 278      3.47428             0.181         1     82.9       2.803642
## 290      3.69695             0.181         0     91.4       2.453429
## 294     51.13580             0.181         0    100.0       1.738711
## 296     41.52920             0.181         0     85.4       2.136970
## 302     14.42080             0.181         0     93.3       3.037741
## 305      9.92485             0.181         0     96.6       3.525691
## 312      3.83684             0.181         0     91.1       3.779233
## 320     13.35980             0.181         0     94.7       2.520527
## 321      6.39312             0.181         0     97.4       3.546246
## 325      8.79212             0.181         0     70.6       3.186891
## 327     15.86030             0.181         0     95.4       2.815191
## 332     24.39380             0.181         0    100.0       1.846643
## 336     22.05110             0.181         0     92.4       2.713520
## 337      6.96215             0.181         0     97.0       2.855160
## 340     73.53410             0.181         0    100.0       2.567078
## 341     10.06230             0.181         0     94.3       3.248145
## 342     38.35180             0.181         0    100.0       1.891958
## 344      5.69175             0.181         0     79.8       7.578136
## 347      7.52601             0.181         0     98.3       3.492387
## 352      5.70818             0.181         0     74.9       6.859073
## 356      4.87141             0.181         0     93.6       3.805081
## 357     13.91340             0.181         0     95.0       3.588005
## 359      7.36711             0.181         0     78.1       2.876769
## 364     11.57790             0.181         0     97.0       2.493201
## 365     14.05070             0.181         0    100.0       1.969563
## 366     20.08490             0.181         0     91.2       1.791178
## 368      3.67367             0.181         0     51.9       9.159096
## 370      9.23230             0.181         0    100.0       1.283993
## 376      9.72418             0.181         0     97.2       3.190845
## 378      4.55587             0.181         0     87.9       2.149320
## 382     24.80170             0.181         0     96.0       2.343483
## 383      7.75223             0.181         0     83.7       5.143350
## 387      5.82115             0.181         0     89.9       5.198162
## 389     20.71620             0.181         0    100.0       1.299845
## 393      4.34879             0.181         0     84.0       5.903200
##     distanceToHighway pupilTeacherRatio pctLowIncome homeValue
## 1                  24              23.2           18    245250
## 4                  24              23.2           41    166500
## 18                 24              23.2            3   1125000
```

```
## 22                   24              23.2            29     189000
## 26                   24              23.2            17     337500
## 32                   24              23.2            31     213750
## 33                   24              23.2            23     310500
## 40                   24              23.2            19     310500
## 42                   24              23.2            14     670500
## 54                   24              23.2            22     342000
## 68                   24              23.2            29     301500
## 69                   24              23.2             4    1125000
## 71                   24              23.2            21     414000
## 72                   24              23.2            17     519750
## 73                   24              23.2            24     281250
## 78                   24              23.2            22     234000
## 81                   24              23.2            18     452250
## 89                   24              23.2            49     310500
## 91                   24              23.2             8     562500
## 96                   24              23.2            21     303750
## 99                   24              23.2            26     162000
## 102                  24              23.2            18     468000
## 121                  24              23.2            30     265500
## 131                  24              23.2            21     427500
## 139                  24              23.2            16     508500
## 141                  24              23.2            18     447750
## 144                  24              23.2            24     335250
## 150                  24              23.2            14     510750
## 157                  24              23.2            16     481500
## 158                  24              23.2            12     468000
## 159                  24              23.2            23     429750
## 165                  24              23.2            18     522000
## 166                  24              23.2            20     319500
## 170                  24              23.2            32     270000
## 171                  24              23.2            18     378000
## 173                  24              23.2            23     328500
## 175                  24              23.2            30     243000
## 180                  24              23.2            47     157500
## 181                  24              23.2             9     562500
## 183                  24              23.2            16     481500
## 186                  24              23.2            25     618750
## 188                  24              23.2            39     162000
## 189                  24              23.2            22     335250
## 192                  24              23.2            24     522000
## 196                  24              23.2            24     285750
## 197                  24              23.2            22     400500
## 201                  24              23.2            39     229500
## 204                  24              23.2            16     488250
## 206                  24              23.2            23     317250
## 210                  24              23.2            31     276750
## 213                  24              23.2            34     126000
## 214                  24              23.2            44     310500
## 218                  24              23.2            25     366750
## 220                  24              23.2            13     517500
## 226                  24              23.2            29     301500
## 233                  24              23.2            30     294750
## 234                  24              23.2            23     317250
```

```
## 235                24              23.2           37    162000
## 237                24              23.2           38    141750
## 242                24              23.2           19    375750
## 245                24              23.2           30    254250
## 248                24              23.2           27    299250
## 249                24              23.2           44    402750
## 252                24              23.2           20    321750
## 254                24              23.2           18    398250
## 255                24              23.2           28    229500
## 257                24              23.2           22    326250
## 275                24              23.2           21    400500
## 276                24              23.2           34    234000
## 278                24              23.2            6    492750
## 290                24              23.2           17    492750
## 294                24              23.2           12    337500
## 296                24              23.2           35    191250
## 302                24              23.2           23    216000
## 305                24              23.2           21    283500
## 312                24              23.2           18    447750
## 320                24              23.2           20    285750
## 321                24              23.2           31    299250
## 325                24              23.2           22    263250
## 327                24              23.2           31    186750
## 332                24              23.2           36    236250
## 336                24              23.2           28    236250
## 337                24              23.2           21    339750
## 340                24              23.2           26    198000
## 341                24              23.2           25    317250
## 342                24              23.2           39    112500
## 344                24              23.2           19    429750
## 347                24              23.2           24    292500
## 352                24              23.2            9    533250
## 356                24              23.2           23    375750
## 357                24              23.2           19    263250
## 359                24              23.2           27    247500
## 364                24              23.2           33    218250
## 365                24              23.2           27    387000
## 366                24              23.2           39    198000
## 368                24              23.2           13    477000
## 370                24              23.2           12   1125000
## 376                24              23.2           25    384750
## 378                24              23.2            8    618750
## 382                24              23.2           25    186750
## 383                24              23.2           20    335250
## 387                24              23.2           13    454500
## 389                24              23.2           30    267750
## 393                24              23.2           20    447750
##     pollutionIndex nBedRooms
## 1             52.9     4.202
## 4             55.0     3.000
## 18            48.1     5.016
## 22            56.8     4.824
## 26            52.1     5.313
## 32            52.9     4.380
```

```
## 33            43.4    3.427
## 40            54.3    4.193
## 42            46.4    4.980
## 54            56.3    4.655
## 68            59.0    4.629
## 69            48.1    4.683
## 71            59.0    4.219
## 72            48.1    1.863
## 73            54.3    4.405
## 78            52.1    4.968
## 81            43.0    3.713
## 89            51.8    2.138
## 91            38.2    5.061
## 96            56.3    3.936
## 99            54.3    4.343
## 102           46.4    3.648
## 121           59.0    4.459
## 131           50.5    3.952
## 139           62.0    4.112
## 141           50.5    3.759
## 144           56.3    4.728
## 150           62.0    4.127
## 157           46.4    4.229
## 158           62.0    3.362
## 159           43.0    3.926
## 165           43.0    4.437
## 166           56.8    4.006
## 170           46.4    3.304
## 171           62.0    3.803
## 173           46.4    4.185
## 175           59.0    3.854
## 180           54.3    2.519
## 181           62.0    4.398
## 183           50.5    4.209
## 186           44.7    4.852
## 188           55.0    3.277
## 189           59.0    4.341
## 192           55.0    4.051
## 196           56.3    3.976
## 197           62.0    4.212
## 201           55.0    2.880
## 204           62.0    4.395
## 206           56.3    4.185
## 210           55.0    3.520
## 213           54.3    3.987
## 214           51.8    2.906
## 218           44.7    3.155
## 220           38.2    4.242
## 226           46.4    4.103
## 233           52.1    4.380
## 234           56.3    4.525
## 235           52.9    4.434
## 237           54.3    3.852
## 242           56.8    4.411
```

```
## 245          55.0     3.536
## 248          52.1     4.794
## 249          44.7     2.628
## 252          56.3     4.436
## 254          56.3     4.376
## 255          52.1     4.223
## 257          43.4     4.348
## 275          56.3     5.393
## 276          52.9     3.304
## 278          56.8     6.780
## 290          56.8     2.963
## 294          44.7     3.757
## 296          54.3     3.531
## 302          59.0     4.461
## 305          59.0     4.251
## 312          62.0     4.251
## 320          54.3     3.887
## 321          43.4     4.162
## 325          43.4     3.565
## 327          52.9     3.896
## 332          55.0     2.652
## 336          59.0     3.818
## 337          55.0     3.713
## 340          52.9     3.957
## 341          43.4     4.833
## 342          54.3     3.453
## 344          43.3     4.114
## 347          56.3     4.417
## 352          38.2     4.750
## 356          46.4     4.484
## 357          56.3     4.208
## 359          52.9     4.193
## 364          55.0     3.036
## 365          44.7     4.657
## 366          55.0     2.368
## 368          43.3     4.312
## 370          48.1     4.216
## 376          59.0     4.406
## 378          56.8     1.561
## 382          54.3     3.349
## 383          56.3     4.301
## 387          56.3     4.513
## 389          50.9     2.138
## 393          43.0     4.167
```

Not only do these records have the same value for Non Retail Business, but also for distance to highway and pupil teacher ratio. This could indicate a probelem because 25% of our records have the same value for 3 of 10 variables. It is very likely that these three can be used together for any model.

There was also a high number of records that had a value of 0.1958 for the non retail business variable. Let's take a look at those as well.

```
subset(data, nonRetailBusiness==.1958)
```

```
##     crimeRate_pc nonRetailBusiness withWater ageHouse distanceToCity
```

```
## 8       1.65660          0.1958         0     97.3       2.159562
## 10      2.01019          0.1958         0     96.2       3.143511
## 34      2.30040          0.1958         0     96.1       3.277562
## 43      2.24236          0.1958         0     91.8       4.117927
## 87      1.12658          0.1958         1     88.0       2.142929
## 90      1.34284          0.1958         0    100.0       2.464640
## 97      1.80028          0.1958         0     79.2       4.128541
## 142     3.53501          0.1958         1     82.6       2.438214
## 149     1.49632          0.1958         0    100.0       2.103460
## 161     2.36862          0.1958         0     95.7       1.833771
## 164     1.51902          0.1958         1     93.9       3.433753
## 191     2.44953          0.1958         0     95.2       3.692681
## 198     1.42502          0.1958         0    100.0       2.483967
## 199     2.14918          0.1958         0     98.5       2.170677
## 205     2.92400          0.1958         0     93.0       3.747410
## 262     1.20742          0.1958         0     94.6       4.128541
## 272     1.41385          0.1958         1     96.0       2.446936
## 309     2.44668          0.1958         0     94.0       2.417907
## 323     2.15505          0.1958         0    100.0       1.947124
## 324     3.32105          0.1958         1    100.0       1.562284
## 329     1.27346          0.1958         1     92.6       2.557514
## 334     2.73397          0.1958         0     94.9       1.965851
## 369     2.77974          0.1958         0     97.8       1.608498
## 377     2.31390          0.1958         0     97.3       4.027714
## 390     2.33099          0.1958         0     93.8       1.973898
##     distanceToHighway pupilTeacherRatio pctLowIncome homeValue
## 8                   5              17.7           18    483750
## 10                  5              17.7            4   1125000
## 34                  5              17.7           14    535500
## 43                  5              17.7           14    510750
## 87                  5              17.7           15    344250
## 90                  5              17.7            8    546750
## 97                  5              17.7           15    535500
## 142                 5              17.7           19    351000
## 149                 5              17.7           16    441000
## 161                 5              17.7           38    328500
## 164                 5              17.7            4   1125000
## 191                 5              17.7           14    501750
## 198                 5              17.7            9    524250
## 199                 5              17.7           20    436500
## 205                 5              17.7           12    562500
## 262                 5              17.7           18    391500
## 272                 5              17.7           19    382500
## 309                 5              17.7           20    294750
## 323                 5              17.7           21    351000
## 324                 5              17.7           34    301500
## 329                 5              17.7            6    607500
## 334                 5              17.7           27    346500
## 369                 5              17.7           37    265500
## 377                 5              17.7           15    429750
## 390                 5              17.7           36    400500
##     pollutionIndex nBedRooms
## 8             72.1     4.122
## 10            45.5     5.929
```

```
## 34             45.5     4.319
## 43             45.5     3.854
## 87             72.1     3.012
## 90             45.5     4.066
## 97             45.5     3.877
## 142            72.1     4.152
## 149            72.1     3.404
## 161            72.1     2.926
## 164            45.5     6.375
## 191            45.5     4.402
## 198            72.1     4.510
## 199            72.1     3.709
## 205            45.5     4.101
## 262            45.5     3.875
## 272            72.1     4.129
## 309            72.1     3.272
## 323            72.1     3.628
## 324            72.1     3.403
## 329            45.5     4.250
## 334            72.1     3.597
## 369            72.1     2.903
## 377            45.5     3.880
## 390            72.1     3.186
```

The same issue as above with another 25 sharing the same values.

```
subset(data, nonRetailBusiness==.0814)
```

```
##     crimeRate_pc nonRetailBusiness withWater ageHouse distanceToCity
## 48       1.00245            0.0814         0     87.3      10.083736
## 52       1.38799            0.0814         0     82.0       9.152856
## 64       0.98843            0.0814         0    100.0       9.542017
## 103      0.72580            0.0814         0     69.5       8.453050
## 105      0.75026            0.0814         0     94.1      10.701905
## 113      0.62739            0.0814         0     56.5      11.089802
## 120      0.63796            0.0814         0     84.5      10.945402
## 167      0.85204            0.0814         0     89.2       9.234841
## 240      0.95577            0.0814         0     88.8      10.912060
## 258      0.77299            0.0814         0     94.4      10.917157
## 274      1.25179            0.0814         0     98.1       8.458038
## 314      0.67191            0.0814         0     90.3      11.821981
## 358      1.15172            0.0814         0     95.0       8.419944
## 367      1.23247            0.0814         0     91.7       9.104822
## 398      0.80271            0.0814         0     36.6       8.453050
##     distanceToHighway pupilTeacherRatio pctLowIncome homeValue
## 48                  4                24           15    472500
## 52                  4                24           35    297000
## 64                  4                24           25    326250
## 103                 4                24           14    409500
## 105                 4                24           20    351000
## 113                 4                24           10    447750
## 120                 4                24           13    409500
## 167                 4                24           17    441000
```

```
## 240                 4                24           22     333000
## 258                 4                24           16     414000
## 274                 4                24           27     306000
## 314                 4                24           18     373500
## 358                 4                24           23     294750
## 367                 4                24           24     342000
## 398                 4                24           14     454500
##     pollutionIndex nBedRooms
## 48            38.8     4.674
## 52            38.8     3.950
## 64            38.8     3.813
## 103           38.8     3.727
## 105           38.8     3.924
## 113           38.8     3.834
## 120           38.8     4.096
## 167           38.8     3.965
## 240           38.8     4.047
## 258           38.8     4.495
## 274           38.8     3.570
## 314           38.8     3.813
## 358           38.8     3.701
## 367           38.8     4.142
## 398           38.8     3.456
```

The same issue with another 15 sharing the same values. These 15 also have the same value for pollutionIndex. Looking back, the 25 that had a nonRetailBusiness value of .1958 have only two different values. The original 104 have different values, but I am now wary of a fourth variable.

In fact, when nonRetailBusiness values of 0.0620, 0.2189, 0.0397, 0.0990, 0.1001 and 0.0856 are also looked at, they all have the same values for pollutionindex, distance to highway and pupil teacher ratio. These variables will likely not contribute much together, as they tend to vary together as a group.

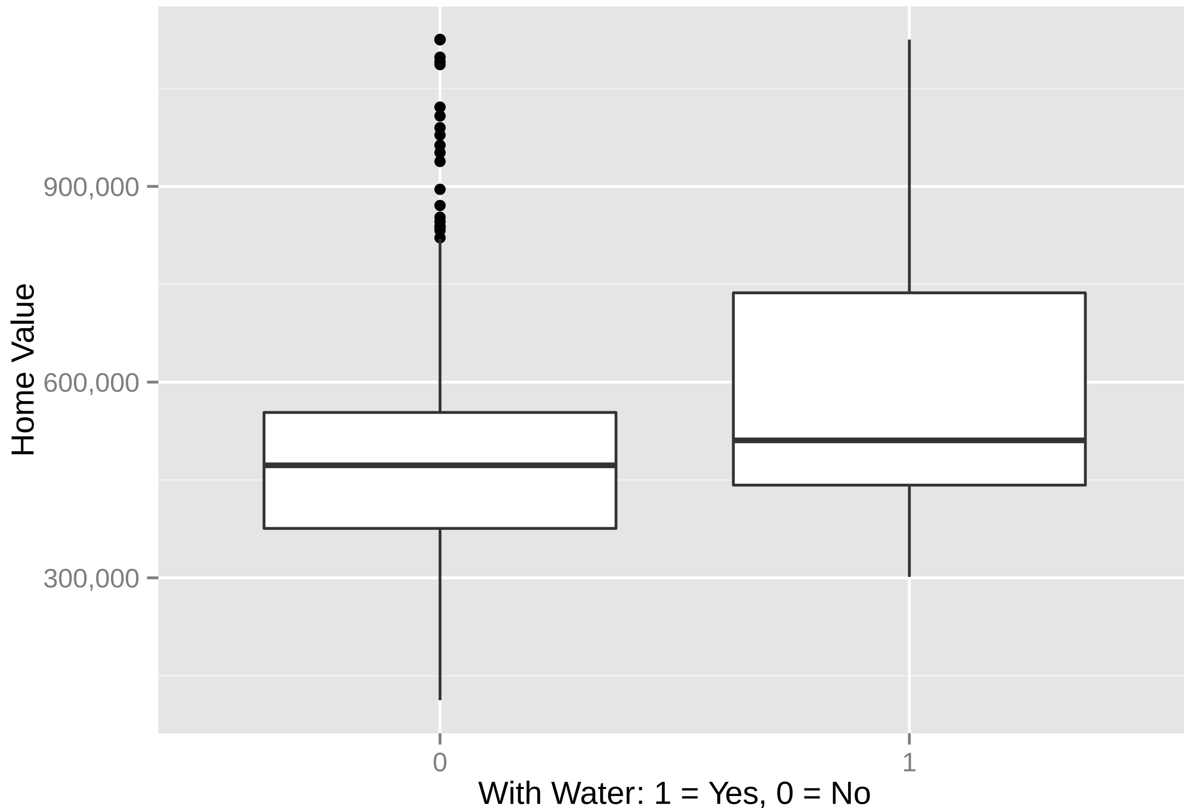
The next variable is withwater which is defined as *whether the neighborhood is within 5 miles of a water body (lake, river, etc); 1 if true and 0 otherwise*

As this is a binary variable, the functions created above are not appropriate.

```r
table(data$withWater)
```

```
##
##   0   1
## 373  27
```

```r
ggplot(data, aes(withWater, homeValue)) + geom_boxplot() +  scale_y_continuous(name = "Home Value", labe
```

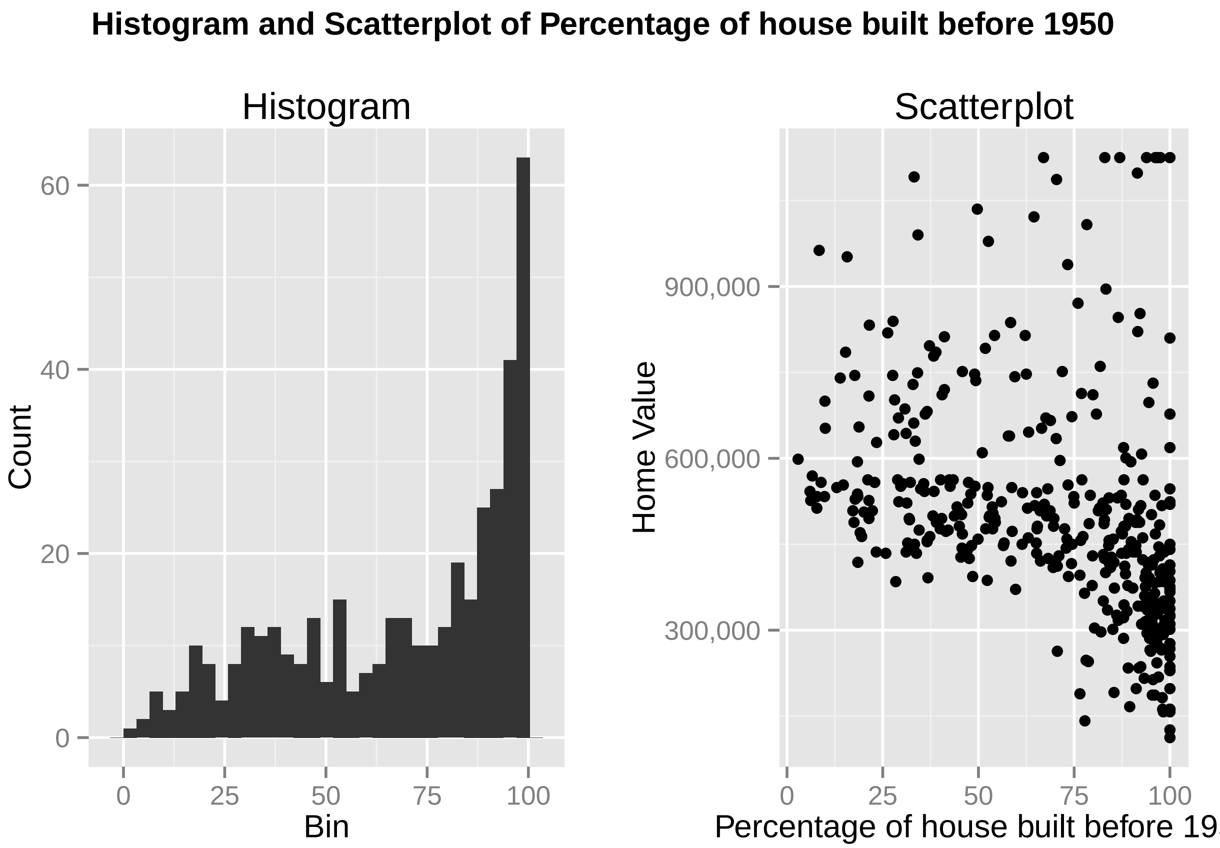


With water tends to have a slightly higher median home value than without water. Neighborhoods without water do tend to see some higher home values, but these are considered outliers that fall outside of the upper whisker.

Now we will examine ageHouse, which is defined as *proportion of houses built before 1950*

```
Graphs('ageHouse', 'Percentage of house built before 1950')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z     cells    name                grob
## 1 1 (2-2,1-1) arrange      gtable[layout]
## 2 2 (2-2,2-2) arrange      gtable[layout]
## 3 3 (1-1,1-2) arrange text[GRID.text.774]
```

```
ContStat(data$ageHouse,1)
```

```
##       Stats
## N     400.0
## #NA's   0.0
## Mean   68.9
## Min     2.9
## Max   100.0
## Std    28.0
## 1%      7.8
## 5%     18.4
## 10%    27.7
## 25%    45.7
## 50%    77.9
## 75%    94.1
## 90%    98.4
## 95%   100.0
## 99%   100.0
```

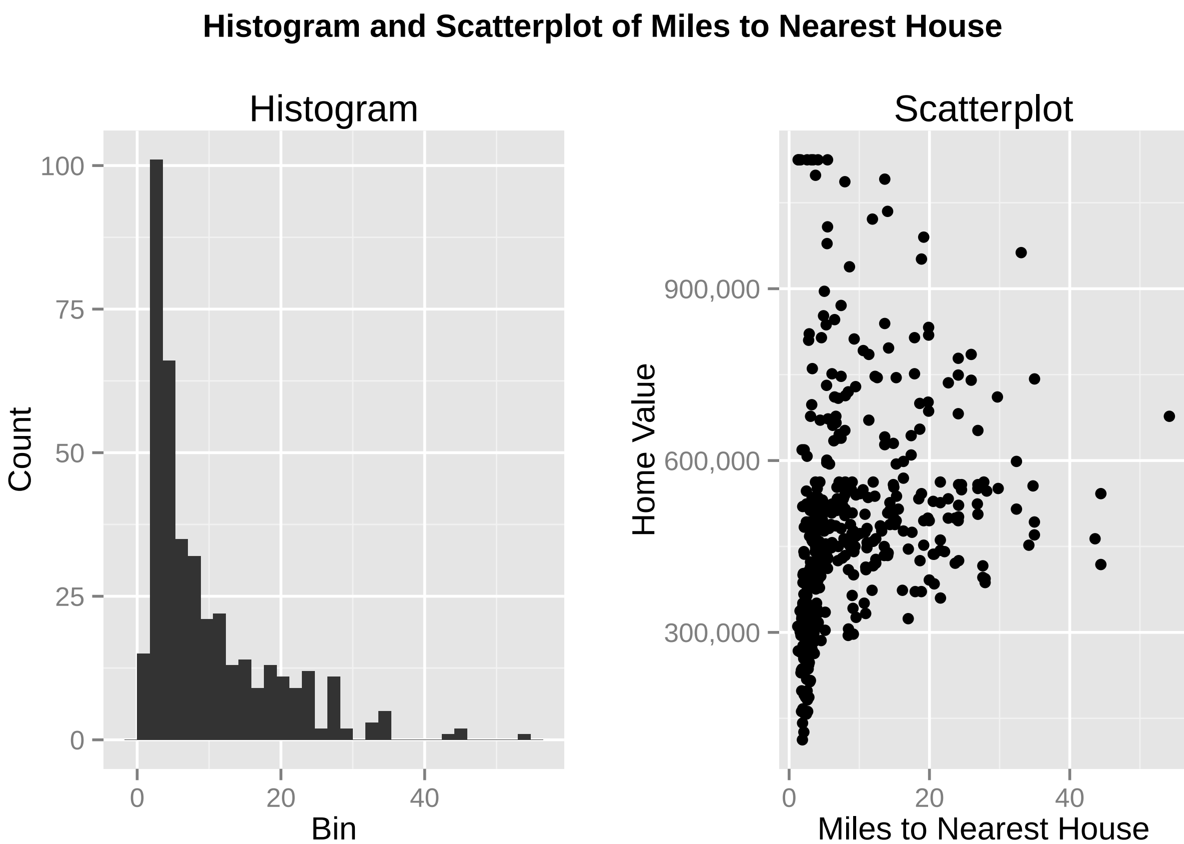
The range here is from 2.9 through 100.0 with a left skew indicating many of these neighborhoods have older homes (built before 1950). With such age buckets, there is ambiguity between neighborhoods built in 1950 and those in 1850. This might account for the larger variation in older neighborhoods home value. Even so, the newer home neighborhoods seem to have higher home values given that there is less of a spread than for older homes. Average age of the home would be a better variable in this case.

---

distanceToCity is next which is *distance to the nearest city (measured in miles)*

```r
Graphs('distanceToCity', 'Miles to Nearest House')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z     cells    name                grob
## 1 1 (2-2,1-1) arrange      gtable[layout]
## 2 2 (2-2,2-2) arrange      gtable[layout]
## 3 3 (1-1,1-2) arrange text[GRID.text.860]
```

```r
ContStat(data$distanceToCity ,1)
```

```
##       Stats
## N     400.0
## #NA's   0.0
## Mean    9.6
```

```
## Min      1.2
## Max     54.2
## Std      8.8
## 1%       1.3
## 5%       1.9
## 10%      2.2
## 25%      3.2
## 50%      6.1
## 75%     13.6
## 90%     22.7
## 95%     26.9
## 99%     35.1
```

Interestingly, the min value here is not 0 which indicates that none of these neighborhoods are actually in the city. The histogram tends to be right skewed, indicating that many neighborhoods are close to the city, while a few are over 40 miles from the city.

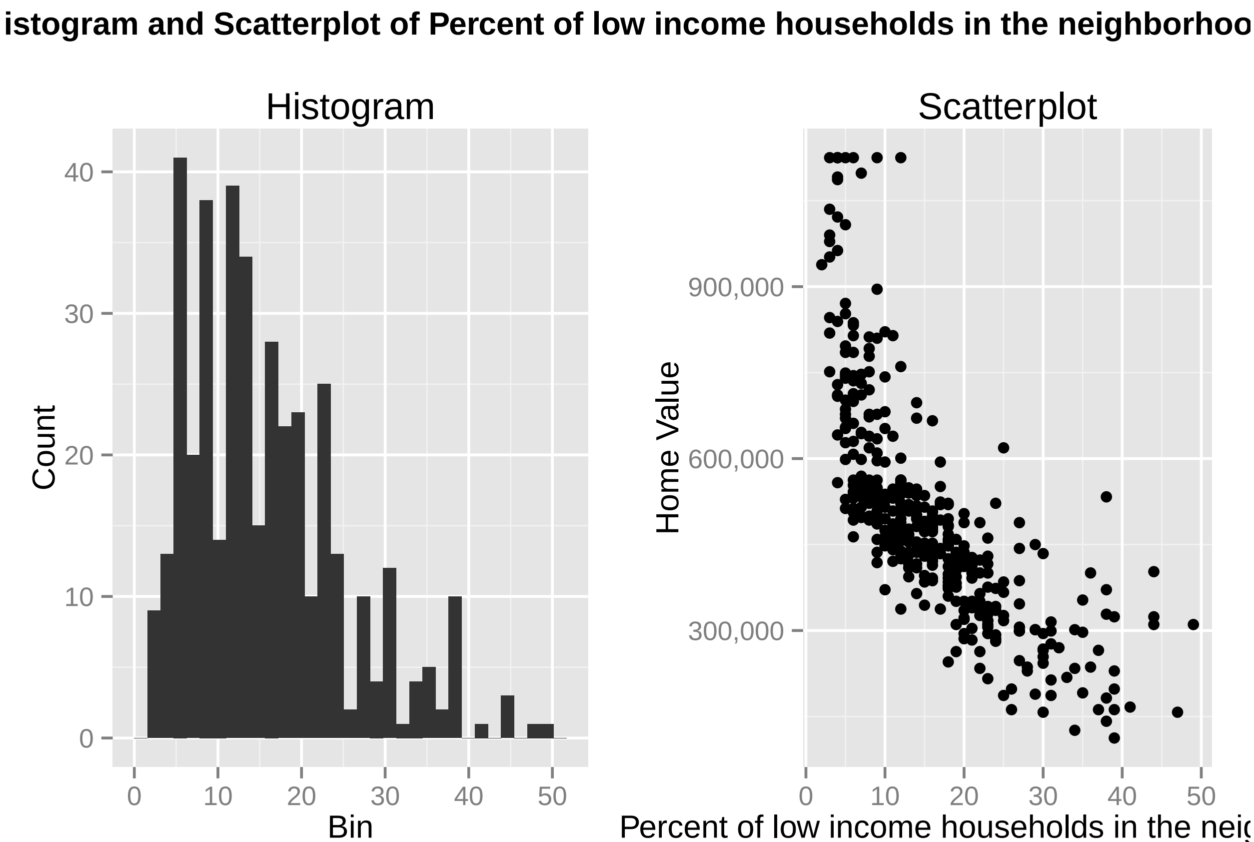
The next two variables in the dataset are distanceToHighway and pupilTeacherRatio. However, as discussed above these two variables will not be used in the model, so further investigation of them is unnecessary.

I think we should probably talk about these... I can write them up tomorrow

The next variable is another percent, pctLowIncome which is *percentage of low income household in the neighborhood*

```
Graphs('pctLowIncome', 'Percent of low income households in the neighborhood')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z     cells    name                grob
## 1 1 (2-2,1-1) arrange      gtable[layout]
## 2 2 (2-2,2-2) arrange      gtable[layout]
## 3 3 (1-1,1-2) arrange text[GRID.text.946]
```

```
ContStat(data$pctLowIncome ,1)
```

```
##        Stats
## N      400.0
## #NA's    0.0
## Mean    15.8
## Min      2.0
## Max     49.0
## Std      9.3
## 1%       3.0
## 5%       4.0
## 10%      5.0
## 25%      8.0
## 50%     14.0
## 75%     21.0
## 90%     29.1
## 95%     35.0
## 99%     44.0
```

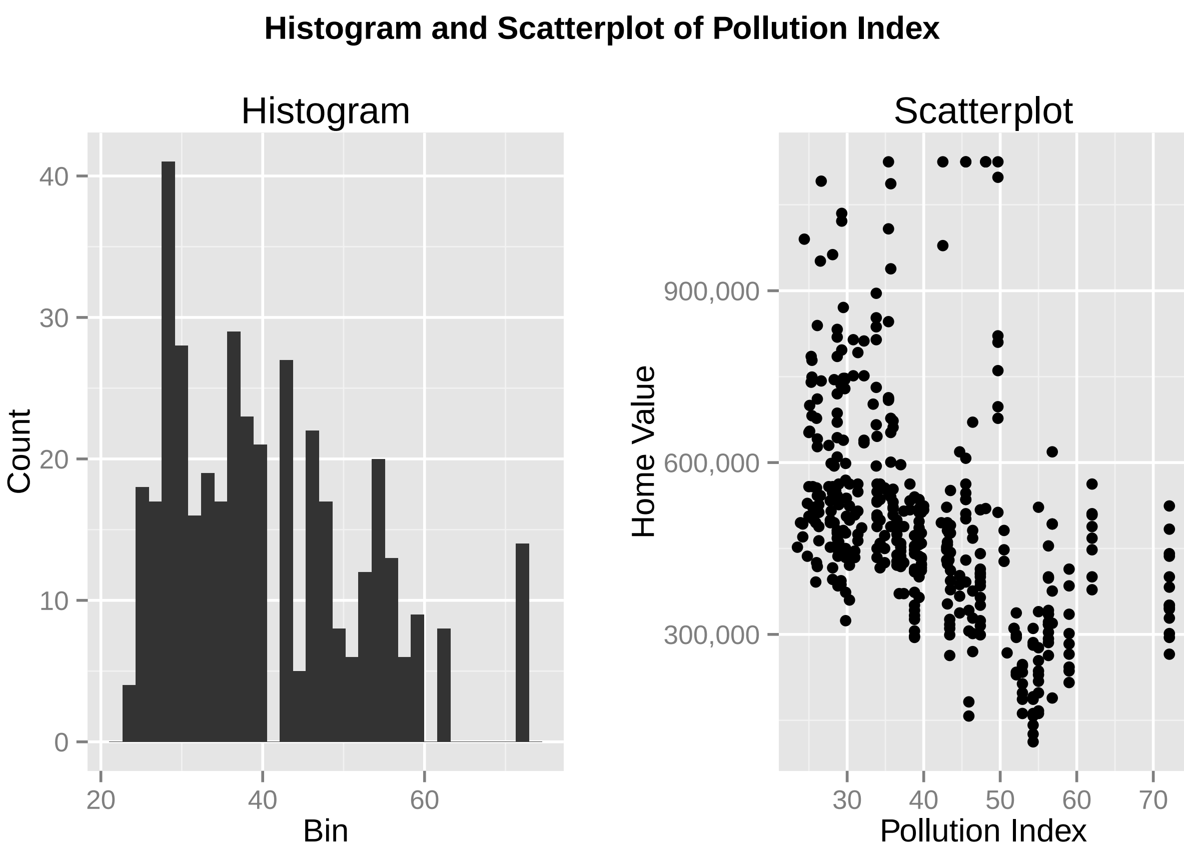
There is a very strong negative correlation on this scatterplot, unsurprisingly. If you have a low income its unlikely that you can afford a house with a high value. This variable is also right skewed, as demonstrated by the histogram.

The next variable is pollutionIndex. Even though it is highly correlated with non retail business, distance to highway and pupil teacher ratio, we will investigate it because the philanthropist group is interested.

Pollutionindex is defined as *scaled between 0 and 100, with 0 being the best and 100 being the worst (i.e. uninhabitable)*

```
Graphs('pollutionIndex', 'Pollution Index')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z     cells    name                grob
## 1 1 (2-2,1-1) arrange      gtable[layout]
## 2 2 (2-2,2-2) arrange      gtable[layout]
## 3 3 (1-1,1-2) arrange text[GRID.text.1032]
```

```
ContStat(data$pollutionIndex ,1)
```

```
##       Stats
```

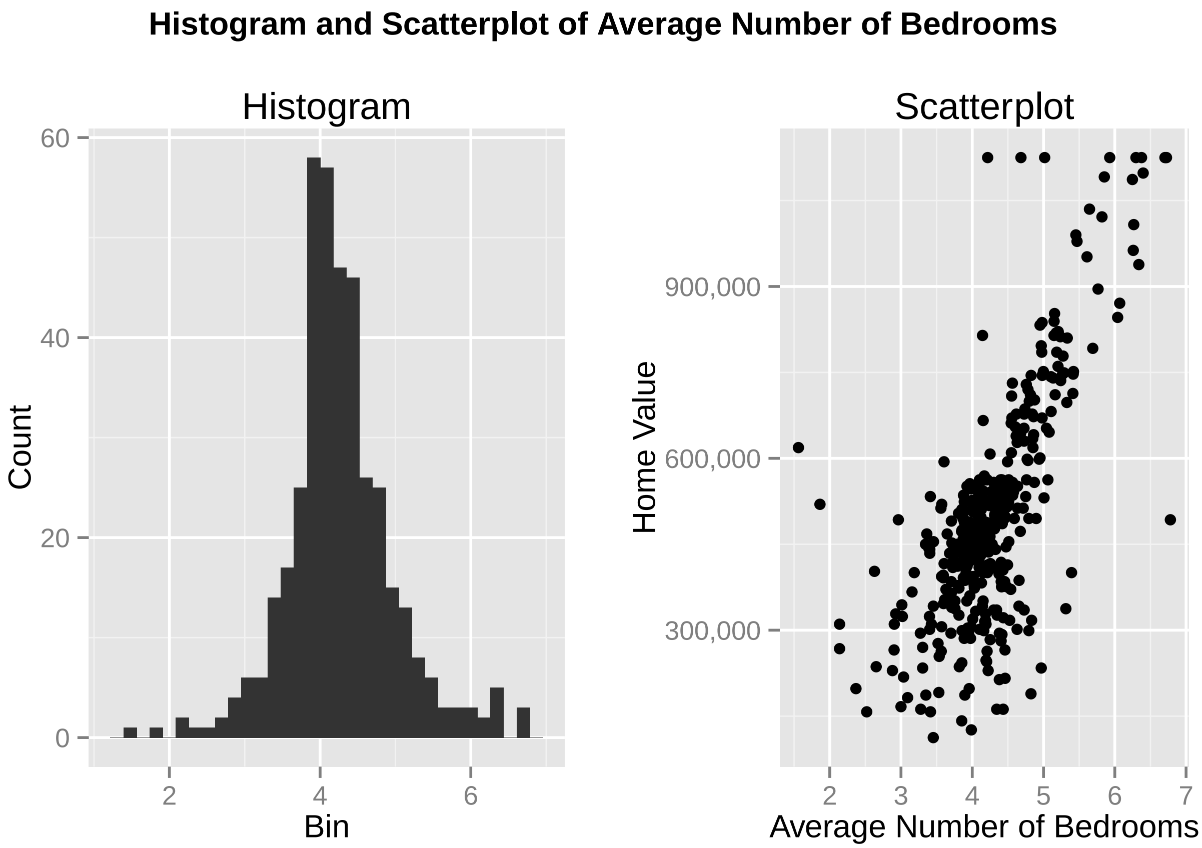
```
## N     400.0
## #NA's   0.0
## Mean   40.6
## Min    23.5
## Max    72.1
## Std    11.8
## 1%     24.4
## 5%     25.9
## 10%    27.6
## 25%    29.9
## 50%    38.8
## 75%    47.6
## 90%    56.3
## 95%    62.0
## 99%    72.1
```

The scatterplot displays multiple segments: high home values and relativey low polution, medium home value and medium pollution, and low home value and high polution. There does seem to be a negative correlation between pollution index and home value, although the scatterplot shows a lot of variation. The histrogram shows a right skew.

The final variable is nBedRooms which is *the average number of bed rooms in the single family houses in the neighborhood*

```r
Graphs('nBedRooms', 'Average Number of Bedrooms')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z     cells    name                grob
## 1 1 (2-2,1-1) arrange       gtable[layout]
## 2 2 (2-2,2-2) arrange       gtable[layout]
## 3 3 (1-1,1-2) arrange text[GRID.text.1118]
```

```
ContStat(data$nBedRooms ,1)
```

```
##        Stats
## N      400.0
## #NA's    0.0
## Mean     4.3
## Min      1.6
## Max      6.8
## Std      0.7
## 1%       2.4
## 5%       3.3
## 10%      3.5
## 25%      3.9
## 50%      4.2
## 75%      4.6
## 90%      5.1
## 95%      5.5
## 99%      6.4
```

Finally! A normally distributed variable. This one is also positively correlated with home value. This will likely be one of the most useful of the prediction variables. It ranges from 1.6 to 6.4, which are reasonable average bedrooms for houses.

From the original dataset, the following decisions were then made.

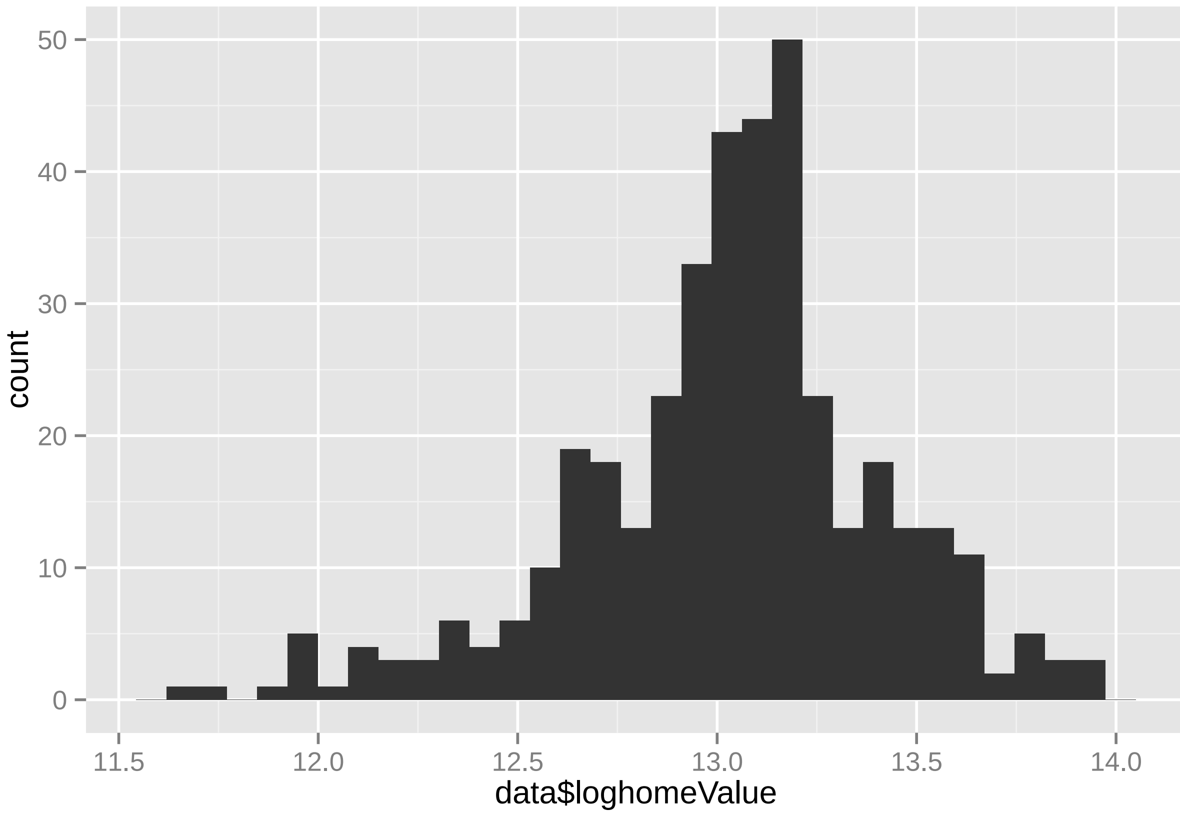
1. Eliminate the variables non retail business, distance to highway and student pupil ratio as they have too much colinearity with each other.

2. While pollutionindex is correlated with the three above, as the group specifically asked about it, it will be kept in the model for now.
3. Create a transformation of home value, log home value, that will be used for fitting the model. Whichever outcome variable performs the best will be used
4. No other transformations will be used at this time. If the model fit is poor, then transformations will be considered.
5. For home value, there are 8 records that are categorical rather than continuous. These are the values that likely mean 1125000 or greater. Because we dont know their true value at all, they will be discarded.

First, we will subset the data and transform some of the variables:

```
#Subset data to remove categorical home values (ceiling)
data = subset(data, homeValue!=1125000)

#Create log home value
data$loghomeValue = log(data$homeValue)

#Examine Transformation
ggplot(data=data, aes(data$loghomeValue)) + geom_histogram()
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```r
ContStat(data$loghomeValue ,1)
```

```
##        Stats
## N      392.0
## #NA's    0.0
## Mean    13.0
## Min     11.6
## Max     13.9
## Std      0.4
## 1%      12.0
## 5%      12.3
## 10%     12.6
## 25%     12.8
## 50%     13.1
## 75%     13.2
## 90%     13.5
## 95%     13.6
## 99%     13.8
```

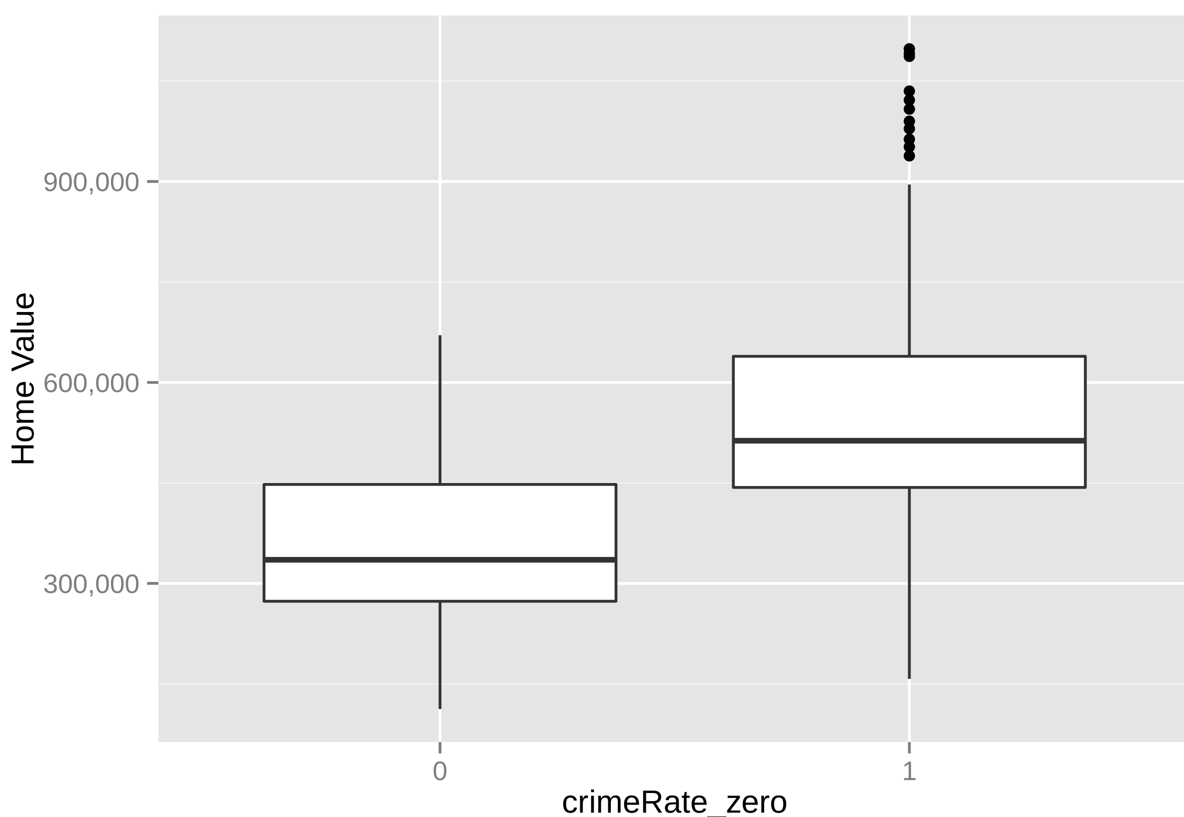
As expected, the log home value transformation has made the histogram more normal, although there is a tail to the left.

We also will create three new binary variables, crimeRate_zero which indicates a very low crime rate and olderneighborhood which indicates if 90% or greater of the houses was built before 1950. FInally we have newerneighborhood which indicates if 25% or less of the houses were built before 1950.

```r
#Transform Variables

data$crimeRate_zero[data$crimeRate_pc < 1.0] <- 1
data$crimeRate_zero[data$crimeRate_pc >= 1.0] <- 0
data$crimeRate_zero <-as.factor(data$crimeRate_zero)

data$olderneighborhood [data$ageHouse >= 90.00] <- 1
data$olderneighborhood [data$ageHouse < 90.00] <- 0
data$olderneighborhood <- as.factor(data$olderneighborhood )

data$newerneighborhood[data$ageHouse <= 10.00] <- 1
data$newerneighborhood[data$ageHouse > 10.00] <- 0
data$newerneighborhood<- as.factor(data$newerneighborhood)

#crimeRate_zero
table(data$crimeRate_zero)
```

```
##
##   0   1
## 131 261
```

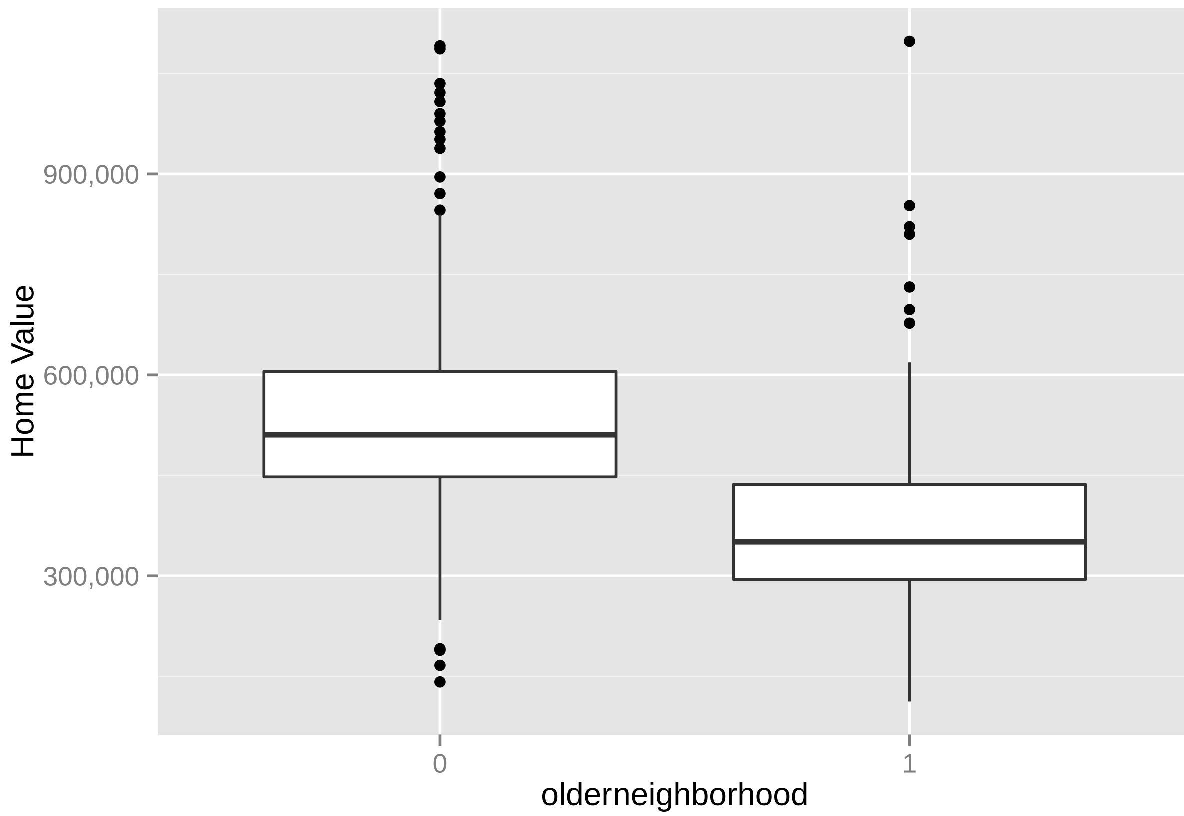
```r
ggplot(data, aes(crimeRate_zero, homeValue)) + geom_boxplot() +  scale_y_continuous(name = "Home Value"
```




```r
#olderneighborhood
table(data$olderneighborhood)
```

```
##
##   0   1
## 263 129
```

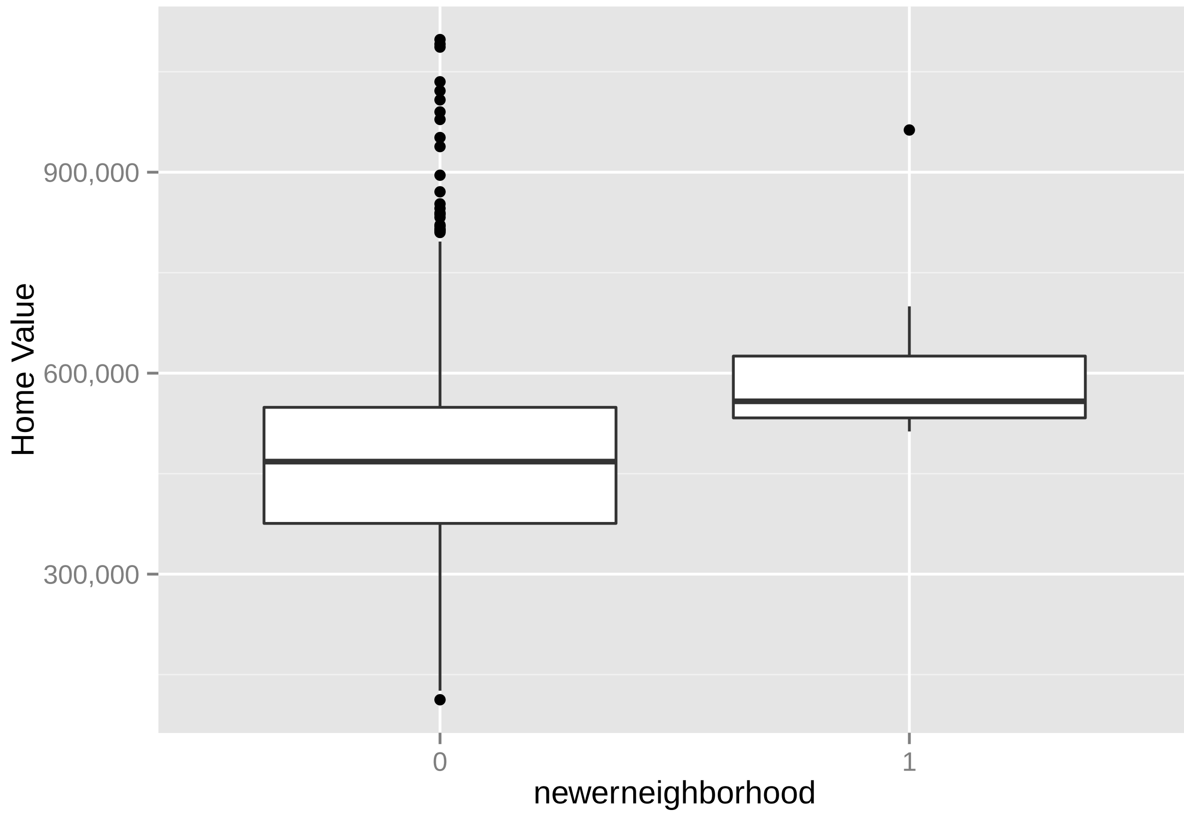
```
ggplot(data, aes(olderneighborhood, homeValue)) + geom_boxplot() +  scale_y_continuous(name = "Home Valu
```




```
#newerneighborhood
table(data$newerneighborhood)
```

```
##
##   0   1
## 381  11
```

```
ggplot(data, aes(newerneighborhood, homeValue)) + geom_boxplot() +  scale_y_continuous(name = "Home Valu
```



All of the transformations box plots look reasonable.

Now we will create two models using the variables identifed above. One will have homevalue as the dependent variable while the other will have the log of home value.

```
#Create Models

lm = lm(homeValue ~ crimeRate_pc+crimeRate_zero+olderneighborhood +newerneighborhood +withWater+ageHous


lmlog = lm(loghomeValue ~ crimeRate_pc+crimeRate_zero+olderneighborhood +newerneighborhood +withWater+ag

#Summarize Models
summary(lm)
```

```
##
## Call:
## lm(formula = homeValue ~ crimeRate_pc + crimeRate_zero + olderneighborhood +
##     newerneighborhood + withWater + ageHouse + distanceToCity +
##     pctLowIncome + pollutionIndex + nBedRooms, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -314268  -63763  -20828   44176  372918
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         356168.7    71459.2   4.984 9.46e-07 ***
```

```
## crimeRate_pc        -2320.3      698.5  -3.322 0.000980 ***
## crimeRate_zero1     20836.3    18790.8   1.109 0.268192
## olderneighborhood1  14575.2    15685.0   0.929 0.353351
## newerneighborhood1  -8511.3    33468.6  -0.254 0.799396
## withWater1          42266.3    21011.5   2.012 0.044969 *
## ageHouse             -735.5      367.1  -2.003 0.045847 *
## distanceToCity      -3432.0      887.1  -3.869 0.000129 ***
## pctLowIncome        -7482.8      928.6  -8.058 1.01e-14 ***
## pollutionIndex      -2067.5      883.7  -2.339 0.019827 *
## nBedRooms           95806.2     9605.2   9.974  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 97630 on 381 degrees of freedom
## Multiple R-squared:  0.7012, Adjusted R-squared:  0.6933
## F-statistic:  89.4 on 10 and 381 DF,  p-value: < 2.2e-16
```

```r
summary(lmlog)
```

```
##
## Call:
## lm(formula = loghomeValue ~ crimeRate_pc + crimeRate_zero + olderneighborhood +
##     newerneighborhood + withWater + ageHouse + distanceToCity +
##     pctLowIncome + pollutionIndex + nBedRooms, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.70652 -0.11549 -0.01647  0.10889  0.79953
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        13.2479547  0.1417015  93.492  < 2e-16 ***
## crimeRate_pc       -0.0093308  0.0013850  -6.737 5.99e-11 ***
## crimeRate_zero1     0.0109161  0.0372616   0.293 0.769713
## olderneighborhood1  0.0147953  0.0311029   0.476 0.634569
## newerneighborhood1  0.0138992  0.0663674   0.209 0.834226
## withWater1          0.1061973  0.0416653   2.549 0.011200 *
## ageHouse           -0.0005916  0.0007280  -0.813 0.416921
## distanceToCity     -0.0064739  0.0017591  -3.680 0.000267 ***
## pctLowIncome       -0.0217021  0.0018414 -11.786  < 2e-16 ***
## pollutionIndex     -0.0054667  0.0017524  -3.119 0.001950 **
## nBedRooms           0.1106651  0.0190468   5.810 1.32e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1936 on 381 degrees of freedom
## Multiple R-squared:  0.747,  Adjusted R-squared:  0.7404
## F-statistic: 112.5 on 10 and 381 DF,  p-value: < 2.2e-16
```

None of the newly created variables are significant. Let's remove them and take another look.

```r
#Create Models

lm = lm(homeValue ~ crimeRate_pc+withWater+ageHouse+distanceToCity+pctLowIncome+pollutionIndex+nBedRooms


lmlog = lm(loghomeValue ~ crimeRate_pc+withWater+ageHouse+distanceToCity+pctLowIncome+pollutionIndex+nBe

#Summarize Models
summary(lm)
```

```
##
## Call:
## lm(formula = homeValue ~ crimeRate_pc + withWater + ageHouse +
##     distanceToCity + pctLowIncome + pollutionIndex + nBedRooms,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -326175  -62589  -20390   44101  358626
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    385514.8    57645.6   6.688 8.02e-11 ***
## crimeRate_pc    -2586.2      641.3  -4.033 6.64e-05 ***
## withWater1      42448.3    20963.3   2.025 0.043570 *
## ageHouse         -565.8      305.4  -1.853 0.064686 .
## distanceToCity  -3425.9      867.3  -3.950 9.29e-05 ***
## pctLowIncome    -7472.5      903.3  -8.272 2.18e-15 ***
## pollutionIndex  -2565.0      694.8  -3.692 0.000255 ***
## nBedRooms       95425.1     9550.4   9.992  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 97520 on 384 degrees of freedom
## Multiple R-squared:  0.6995, Adjusted R-squared:  0.694
## F-statistic: 127.7 on 7 and 384 DF,  p-value: < 2.2e-16
```

```r
summary(lmlog)
```

```
##
## Call:
## lm(formula = loghomeValue ~ crimeRate_pc + withWater + ageHouse +
##     distanceToCity + pctLowIncome + pollutionIndex + nBedRooms,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69694 -0.11228 -0.01758  0.10709  0.79841
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    13.2586595  0.1140522 116.251  < 2e-16 ***
```

```
## crimeRate_pc   -0.0094543  0.0012687  -7.452 6.15e-13 ***
## withWater1      0.1056555  0.0414760   2.547 0.011242 *
## ageHouse       -0.0005112  0.0006042  -0.846 0.398039
## distanceToCity -0.0064698  0.0017159  -3.770 0.000189 ***
## pctLowIncome   -0.0215567  0.0017872 -12.062  < 2e-16 ***
## pollutionIndex -0.0056440  0.0013747  -4.106 4.93e-05 ***
## nBedRooms       0.1110501  0.0188956   5.877 9.07e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1929 on 384 degrees of freedom
## Multiple R-squared:  0.7467, Adjusted R-squared:  0.7421
## F-statistic: 161.8 on 7 and 384 DF,  p-value: < 2.2e-16
```

In both models, withwater and ageHouse are far less significant than the other 5 variables. I will remove them and recreate the models.

```
lm = lm(homeValue ~ crimeRate_pc+distanceToCity+pctLowIncome+pollutionIndex+nBedRooms, data=data)
lmlog = lm(loghomeValue ~ crimeRate_pc+distanceToCity+pctLowIncome+pollutionIndex+nBedRooms, data=data)

#Summarize Models
summary(lm)
```

```
##
## Call:
## lm(formula = homeValue ~ crimeRate_pc + distanceToCity + pctLowIncome +
##     pollutionIndex + nBedRooms, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -287459  -65855  -22211   51161  343691
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    373812.2    57897.0   6.457 3.22e-10 ***
## crimeRate_pc    -2667.8      641.8  -4.157 3.98e-05 ***
## distanceToCity  -2841.3      805.4  -3.528  0.00047 ***
## pctLowIncome    -8122.3      847.6  -9.582  < 2e-16 ***
## pollutionIndex  -2842.2      650.3  -4.370 1.60e-05 ***
## nBedRooms       93481.4     9452.4   9.890  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 98230 on 386 degrees of freedom
## Multiple R-squared:  0.6936, Adjusted R-squared:  0.6896
## F-statistic: 174.7 on 5 and 386 DF,  p-value: < 2.2e-16
```

```
summary(lmlog)
```

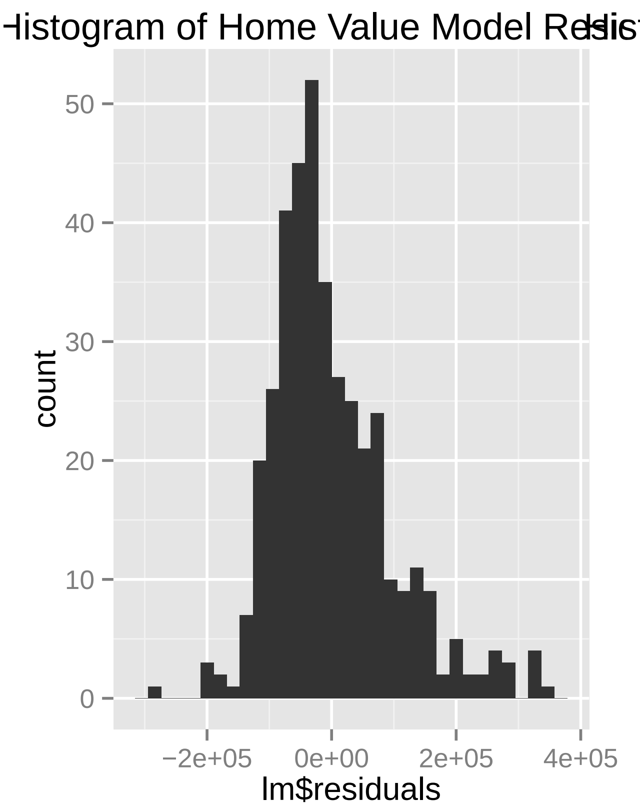
```
##
## Call:
## lm(formula = loghomeValue ~ crimeRate_pc + distanceToCity + pctLowIncome +
```

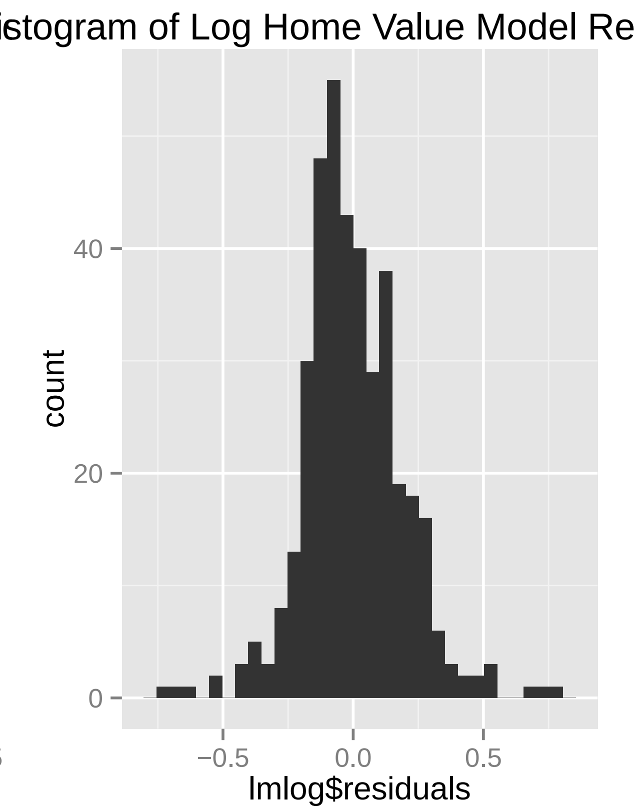
```
##     pollutionIndex + nBedRooms, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.70710 -0.11638 -0.01863  0.11227  0.80215
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    13.235932   0.114497 115.600  < 2e-16 ***
## crimeRate_pc   -0.009740   0.001269  -7.673 1.38e-13 ***
## distanceToCity -0.005999   0.001593  -3.766 0.000192 ***
## pctLowIncome   -0.022211   0.001676 -13.250  < 2e-16 ***
## pollutionIndex -0.005624   0.001286  -4.373 1.58e-05 ***
## nBedRooms       0.111126   0.018693   5.945 6.20e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1943 on 386 degrees of freedom
## Multiple R-squared:  0.742,  Adjusted R-squared:  0.7386
## F-statistic:   222 on 5 and 386 DF,  p-value: < 2.2e-16
```

The r squared values did not change which is unsurprising since the two variables removed were not adding a whole lot of new information to the model. Lets take a look at histograms of the residuals.

```
lmresid = ggplot(data=lm, aes(lm$residuals)) + geom_histogram() +  ggtitle("Histogram of Home Value Mode
lmlogresid =ggplot(data=lmlog, aes(lmlog$residuals)) + geom_histogram() +  ggtitle("Histogram of Log Hom

grid.arrange(lmresid, lmlogresid, ncol=2,nrow=1)
```

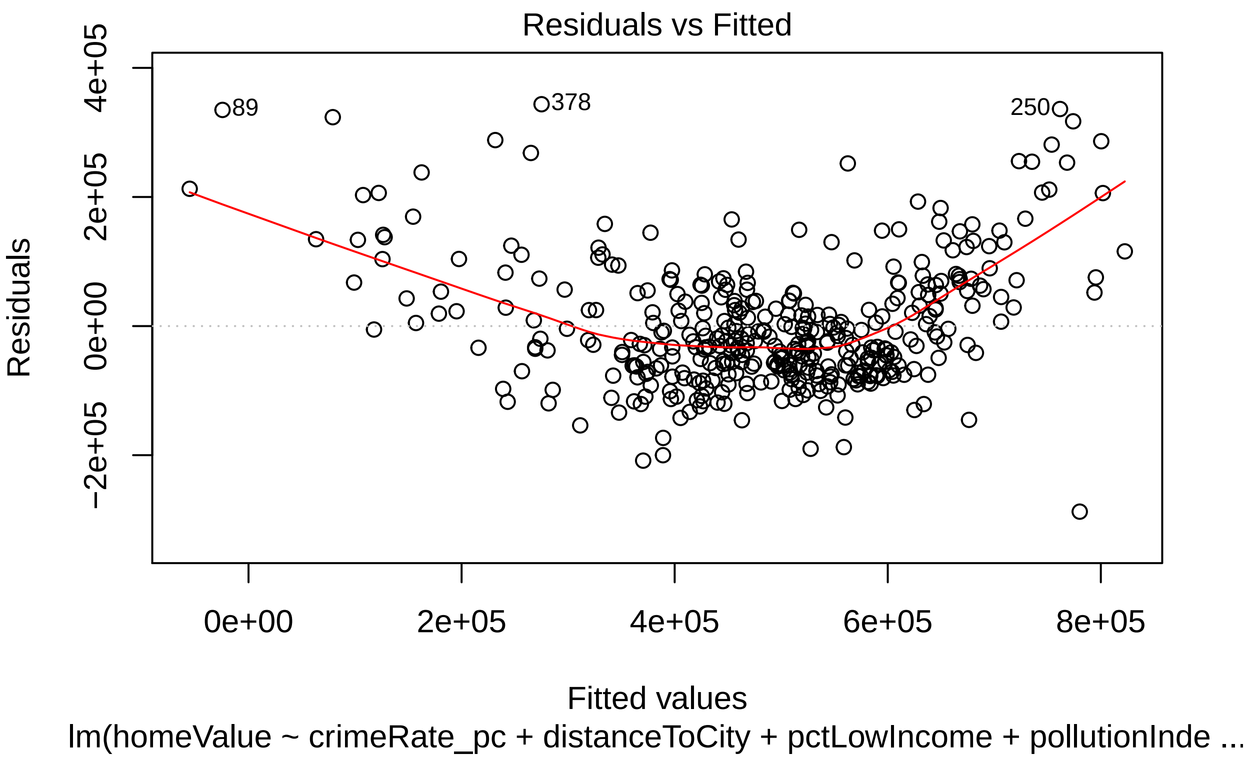
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

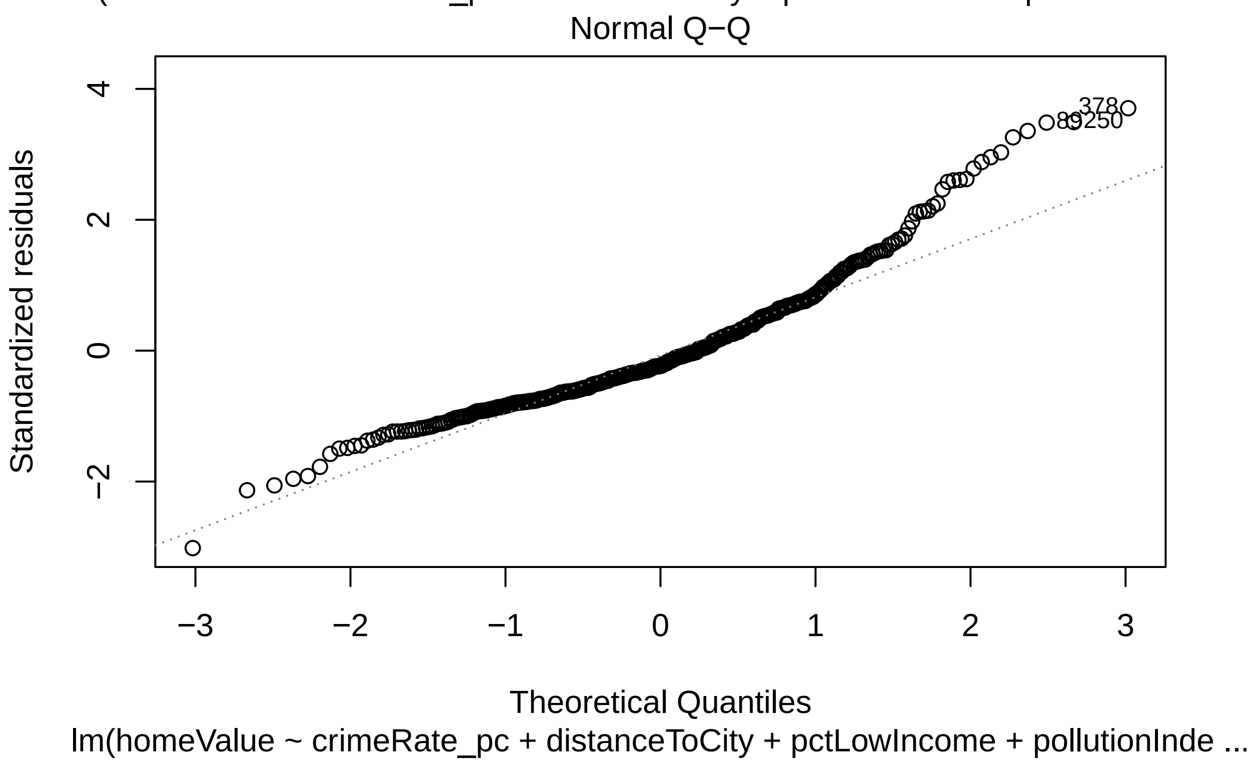


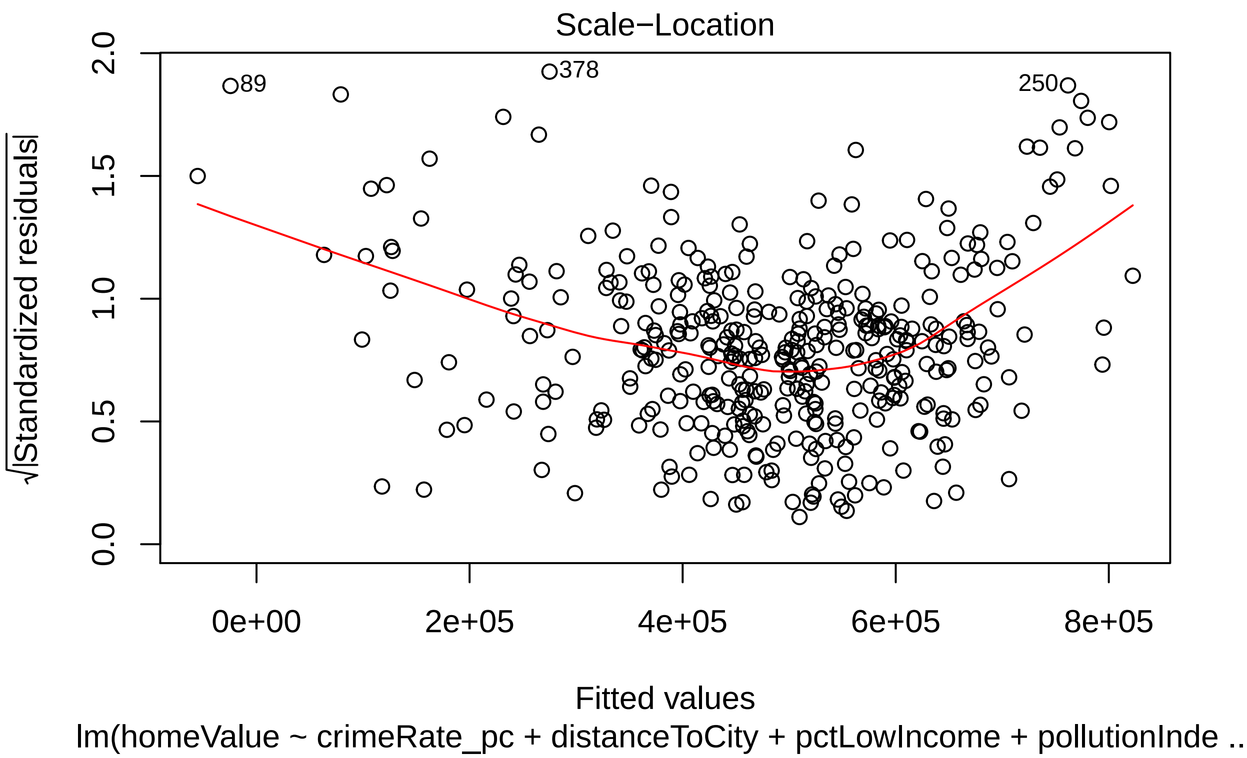


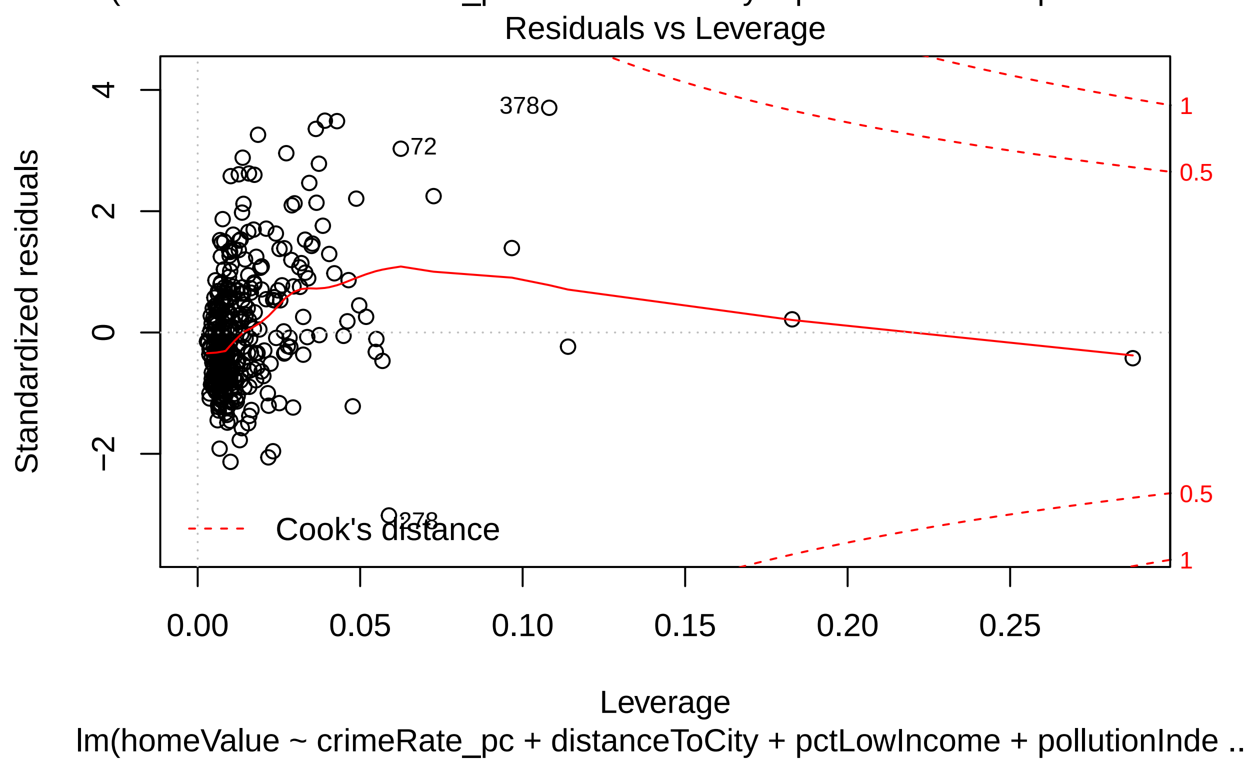
Both sets of residuals are fairly normal, although the log home value residuals are more normal. That in addition to its higher r squared score makes it the favorite thus far. However, lets take a look at the residual disagnostic plots for them before any final decision or addition or transformation of variables is undertaken.

```
plot(lm)
```

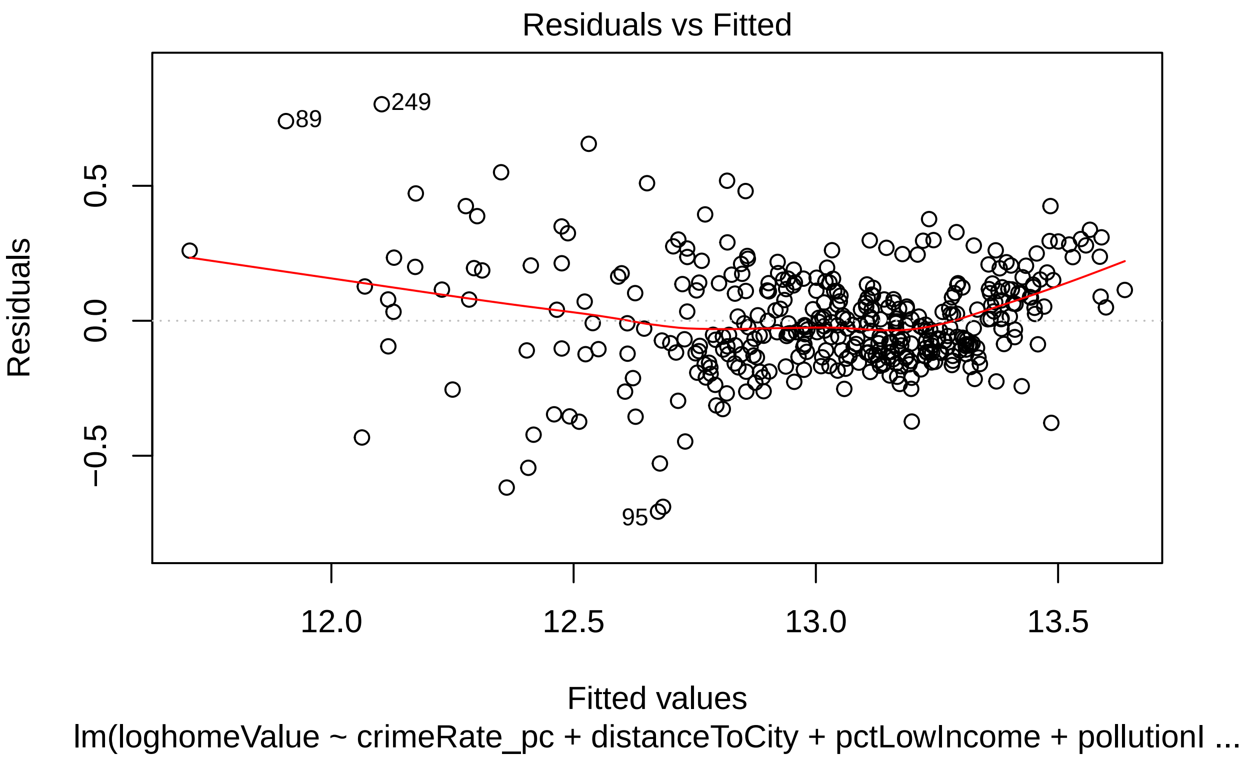

Residuals vs Fitted
Residuals
Fitted values
lm(homeValue ~ crimeRate_pc + distanceToCity + pctLowIncome + pollutionInde ...
4e+05
2e+05
0e+00
−2e+05
0e+00
2e+05
4e+05
6e+05
8e+05
89
378
250

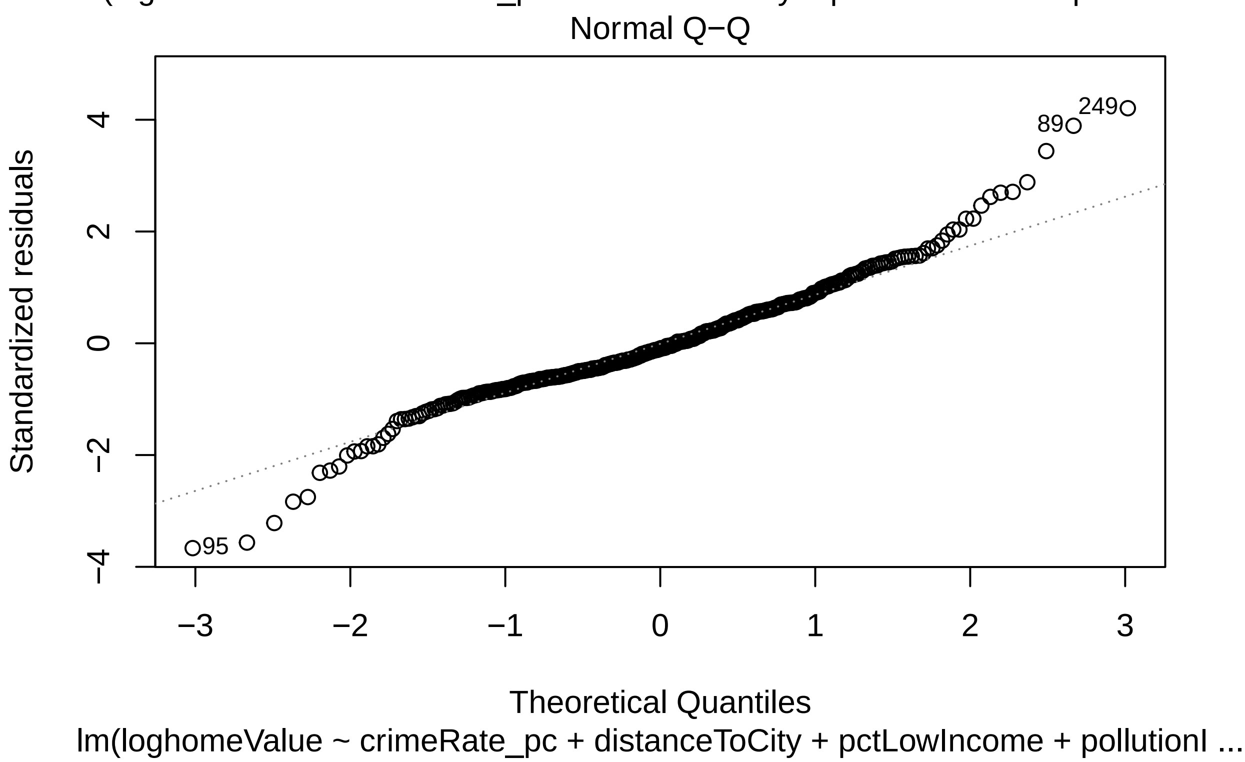

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(homeValue ~ crimeRate_pc + distanceToCity + pctLowIncome + pollutionInde ...
4
2
0
−2
−3
−2
−1
0
1
2
3
378
89
250

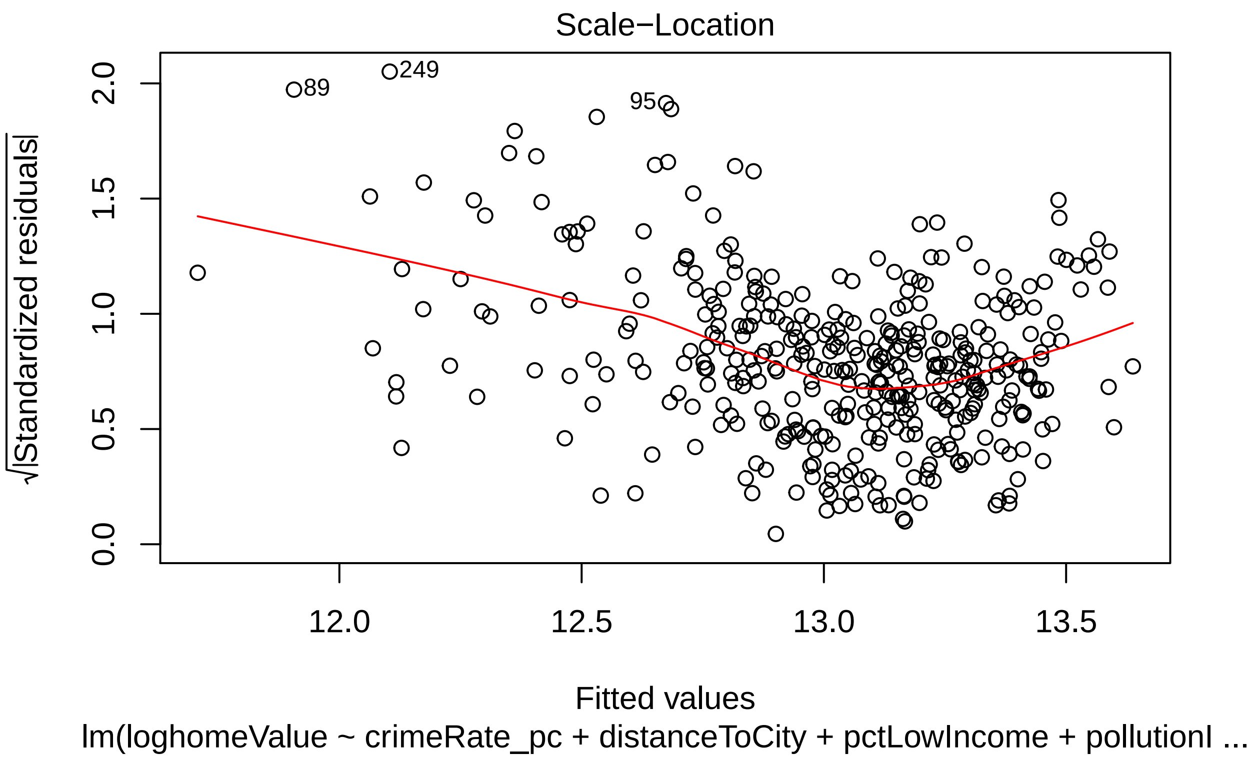

Scale−Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
89
378
250
0e+00
2e+05
4e+05
6e+05
8e+05
Fitted values
lm(homeValue ~ crimeRate_pc + distanceToCity + pctLowIncome + pollutionInde ...

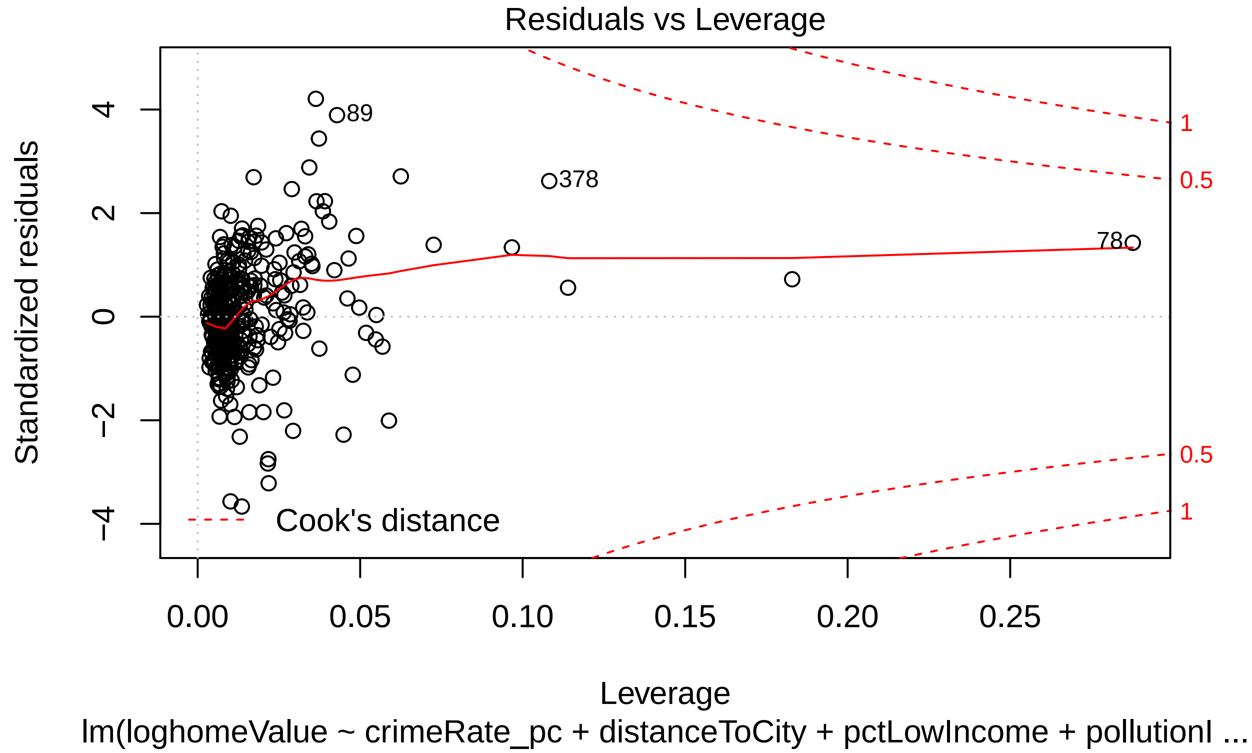

Residuals vs Leverage
Standardized residuals
4
2
0
−2
378
72
278
1
0.5
0.5
1
Cook's distance
0.00
0.05
0.10
0.15
0.20
0.25
Leverage
lm(homeValue ~ crimeRate_pc + distanceToCity + pctLowIncome + pollutionInde ...

```
plot(lmlog)
```


Residuals vs Fitted
Residuals
Fitted values
lm(loghomeValue ~ crimeRate_pc + distanceToCity + pctLowIncome + pollutionI ...
0.5
0.0
−0.5
12.0
12.5
13.0
13.5
89
249
95


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(loghomeValue ~ crimeRate_pc + distanceToCity + pctLowIncome + pollutionI ...
4
2
0
−2
−4
−3
−2
−1
0
1
2
3
249
89
95





Both models show evidence of heteroscedasticity in their residuals vs fitted plots. The log home value model is worse in this sense than the normal one. Both Q-Q plots show that the residuals are pretty normally distributed.

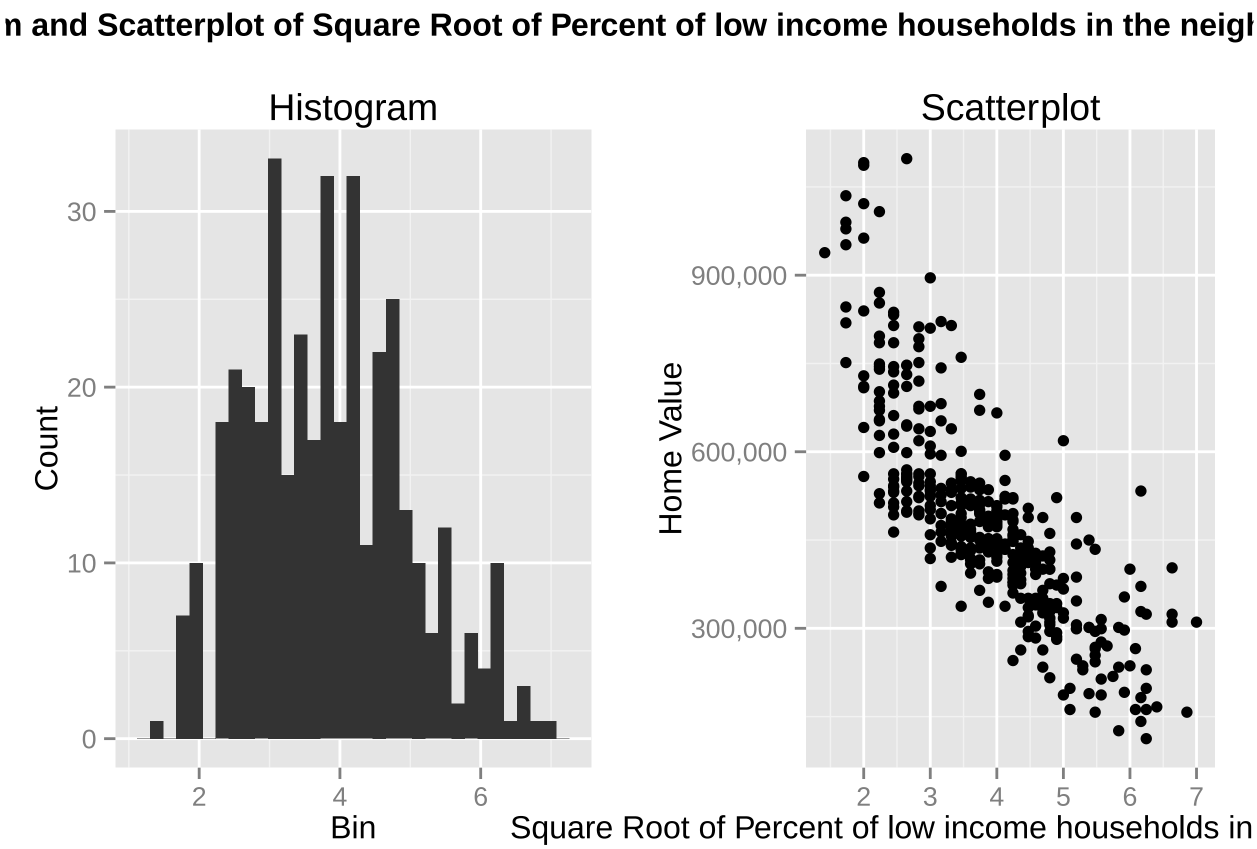
Both scale-location plots also indicate some heteroscedasticity. Finally, the leverage plot indicates that while there are points with a large amount of leverage, they are within our bounds.

At this point we need to either transform variables or add interaction terms.

From the original variable analysis, we know that crimerate_pc and pctLowIncome are skewed to the left. Let's take a look at their distributions when square rooted.

```
data$sqrtpctIncome = sqrt(data$pctLowIncome)
Graphs('sqrtpctIncome', 'Square Root of Percent of low income households in the neighborhood')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



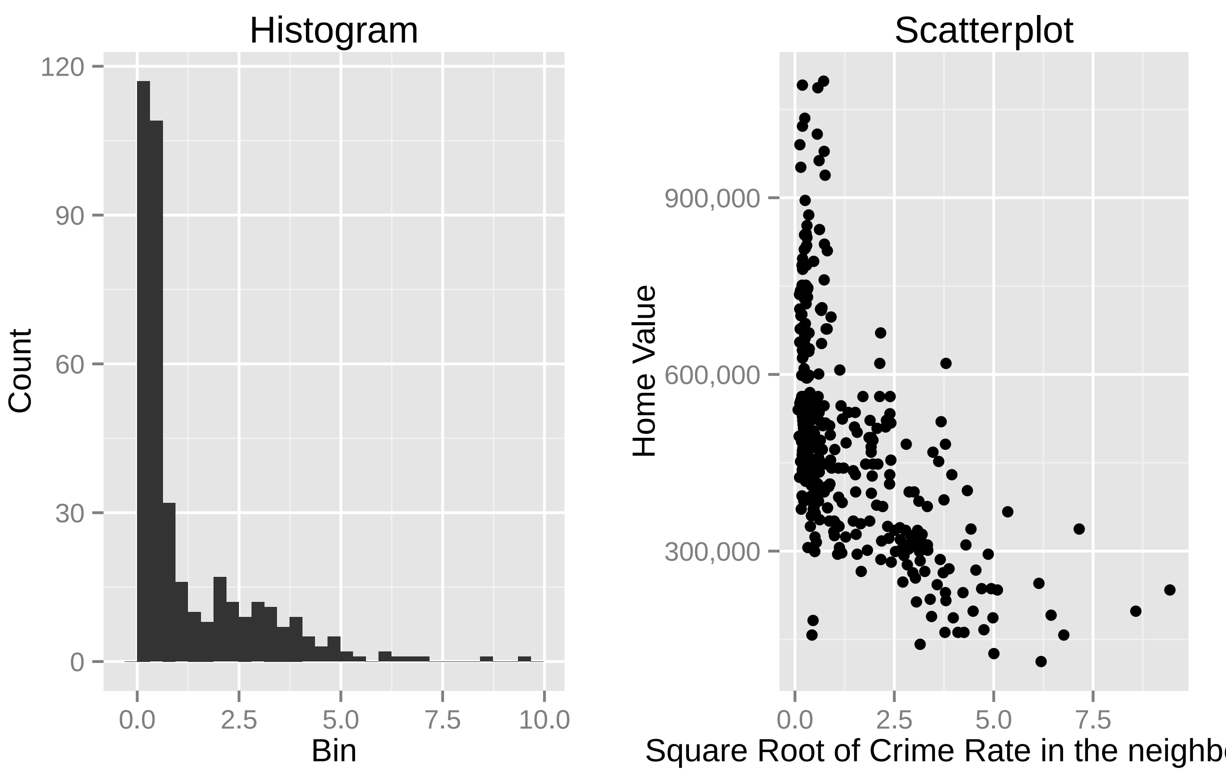

```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z     cells    name              grob
## 1 1 (2-2,1-1) arrange      gtable[layout]
## 2 2 (2-2,2-2) arrange      gtable[layout]
## 3 3 (1-1,1-2) arrange text[GRID.text.1511]
```

The histogram looks far more normal and the scatterplot was not affected negatively which is a great sign.

```
data$sqrtcrimeRate_pc = sqrt(data$crimeRate_pc)
Graphs('sqrtcrimeRate_pc', 'Square Root of Crime Rate in the neighborhood')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

Histogram and Scatterplot of Square Root of Crime Rate in the neighborhood




```
## TableGrob (2 x 2) "arrange": 3 grobs
##   z     cells    name                grob
## 1 1 (2-2,1-1) arrange       gtable[layout]
## 2 2 (2-2,2-2) arrange       gtable[layout]
## 3 3 (1-1,1-2) arrange text[GRID.text.1597]
```

With such a strong left skew, even the square root here does not make the data any more normal in the histogram. However, model performance may have improved. I also want to take another look at withWater as that was a variable of interest.

```
lm = lm(homeValue ~ sqrtcrimeRate_pc+distanceToCity+sqrtpctIncome +pollutionIndex+nBedRooms+withWater, d
lmlog = lm(loghomeValue ~ (crimeRate_pc+distanceToCity+sqrtpctIncome +pollutionIndex+nBedRooms+withWate

#Summarize Models
summary(lm)
```

```
##
## Call:
## lm(formula = homeValue ~ sqrtcrimeRate_pc + distanceToCity +
##     sqrtpctIncome + pollutionIndex + nBedRooms + withWater, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -317297  -60459  -19415   43616  339654
##
## Coefficients:
```

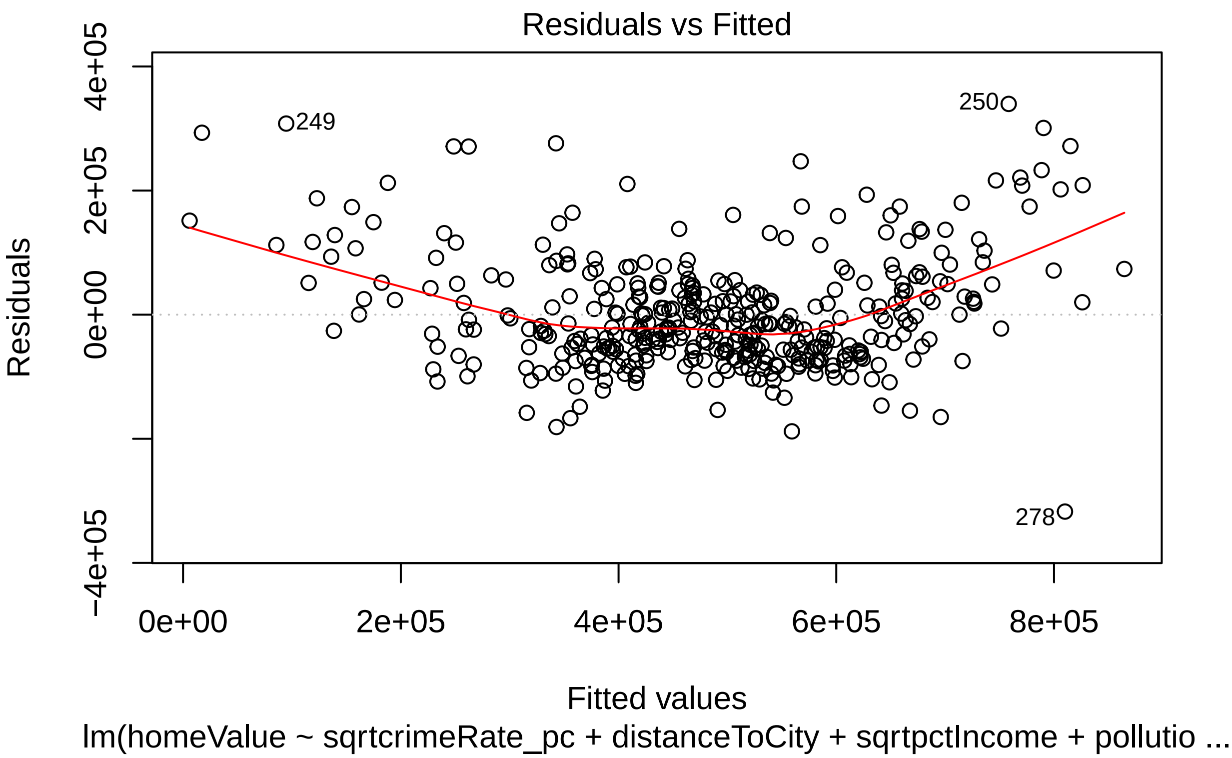
```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      608450.6    64303.6   9.462  < 2e-16 ***
## sqrtcrimeRate_pc -17887.7     4300.7  -4.159 3.94e-05 ***
## distanceToCity    -3303.7      753.8  -4.383 1.51e-05 ***
## sqrtpctIncome    -82162.7     6890.9 -11.923  < 2e-16 ***
## pollutionIndex    -1880.9      649.4  -2.896  0.00399 **
## nBedRooms         75974.2     9138.2   8.314 1.61e-15 ***
## withWater1        37186.6    19734.9   1.884  0.06028 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 91510 on 385 degrees of freedom
## Multiple R-squared:  0.7347, Adjusted R-squared:  0.7306
## F-statistic: 177.7 on 6 and 385 DF,  p-value: < 2.2e-16
```

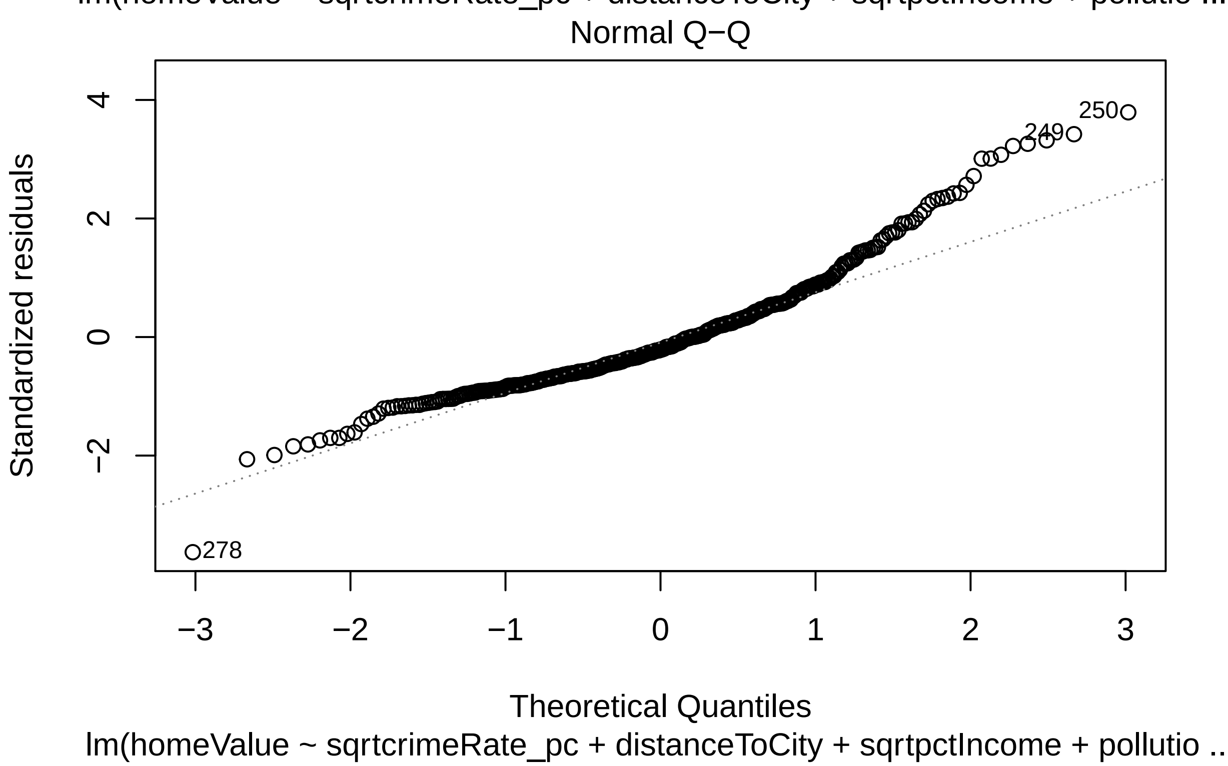
```r
summary(lmlog)
```

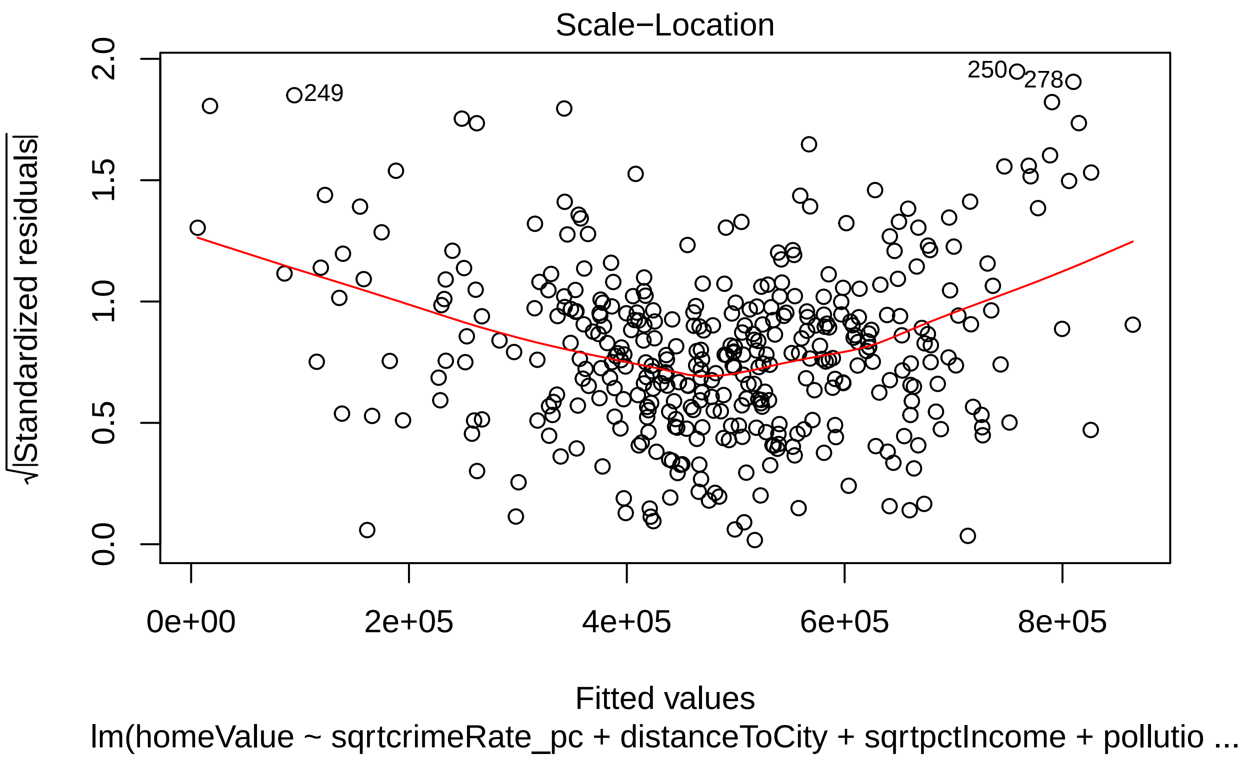
```
##
## Call:
## lm(formula = loghomeValue ~ (crimeRate_pc + distanceToCity +
##     sqrtpctIncome + pollutionIndex + nBedRooms + withWater),
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.71007 -0.10849 -0.01186  0.10663  0.70129
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    13.762408   0.127673 107.794  < 2e-16 ***
## crimeRate_pc   -0.009785   0.001205  -8.119 6.43e-15 ***
## distanceToCity -0.006769   0.001529  -4.427 1.25e-05 ***
## sqrtpctIncome  -0.202586   0.013626 -14.867  < 2e-16 ***
## pollutionIndex -0.005239   0.001248  -4.198 3.35e-05 ***
## nBedRooms       0.082998   0.018460   4.496 9.17e-06 ***
## withWater1      0.099690   0.039909   2.498   0.0129 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1856 on 385 degrees of freedom
## Multiple R-squared:  0.765,  Adjusted R-squared:  0.7614
## F-statistic: 208.9 on 6 and 385 DF,  p-value: < 2.2e-16
```

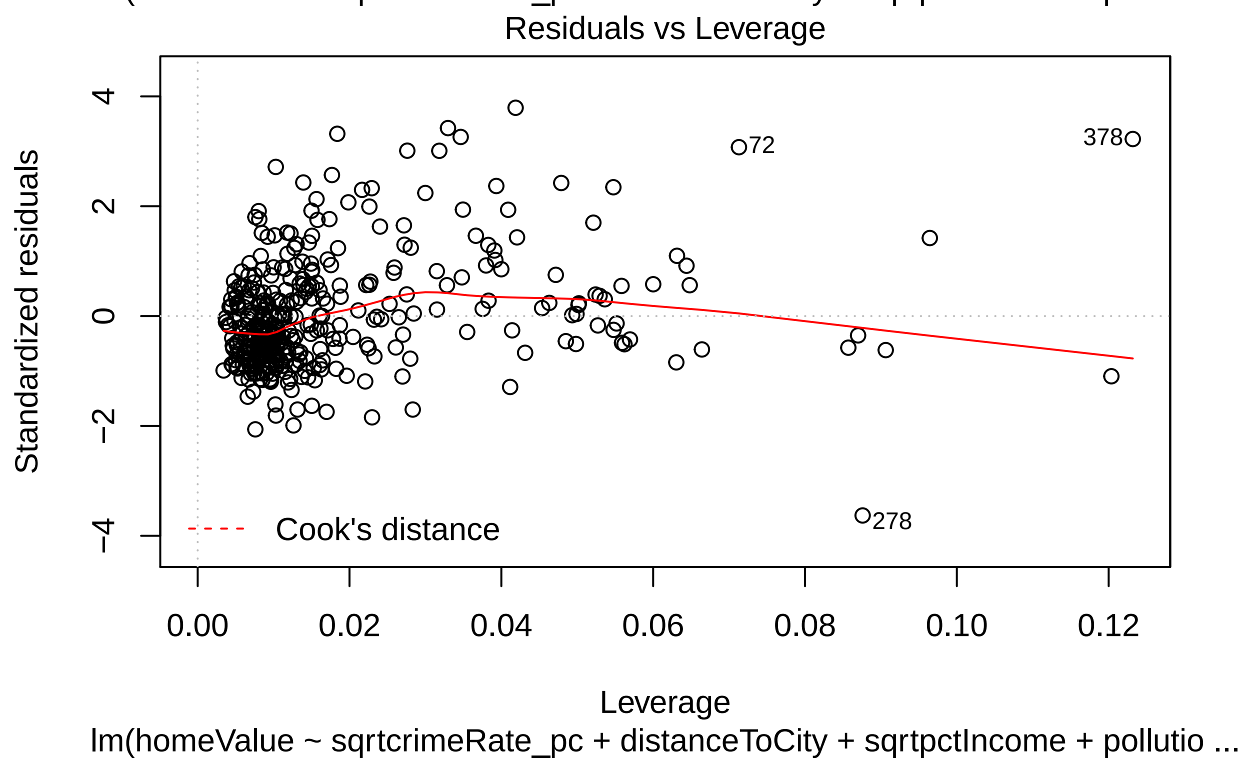
The fit indicated by the r squared value is slightly better. Interestingly, pollutionIndex is becoming far less significant. We will keep it in the model for now and look at the residual plots.

```r
plot(lm)
```

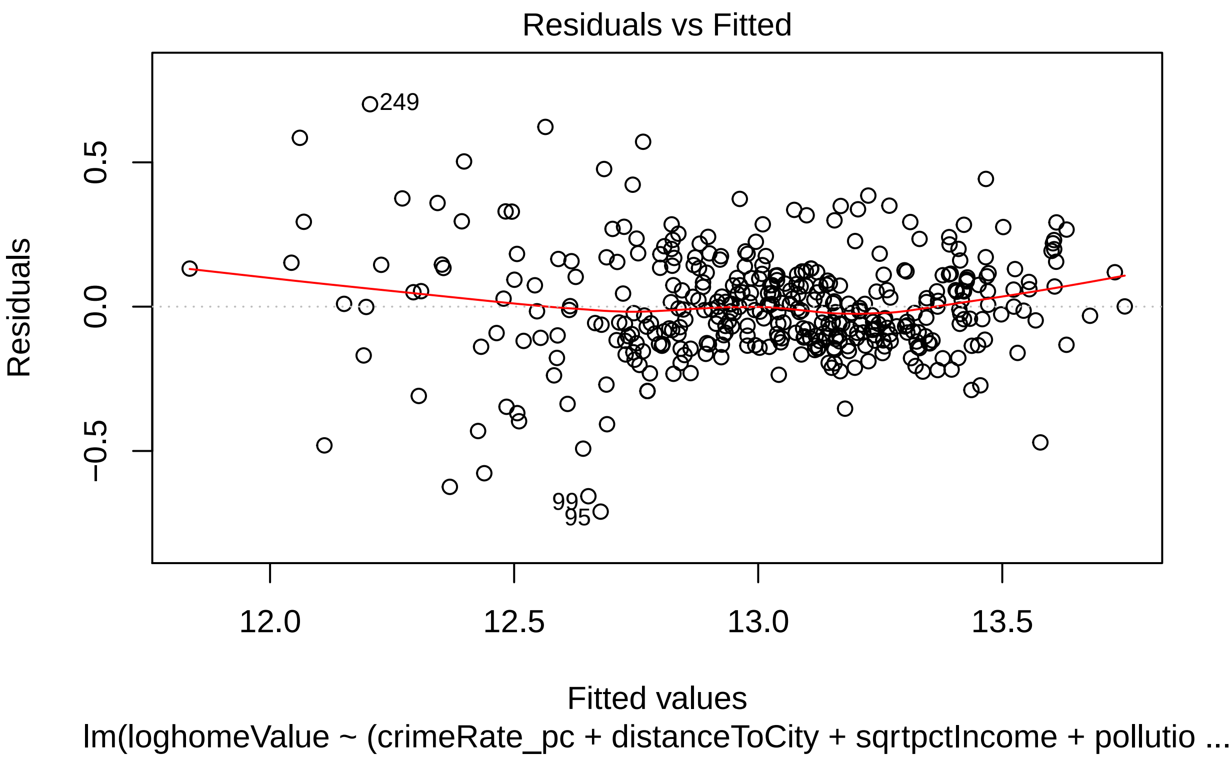

Residuals vs Fitted
Residuals
Fitted values
lm(homeValue ~ sqrtcrimeRate_pc + distanceToCity + sqrtpctIncome + pollutio ...
249
250
278
−4e+05
0e+00
2e+05
4e+05
0e+00
2e+05
4e+05
6e+05
8e+05

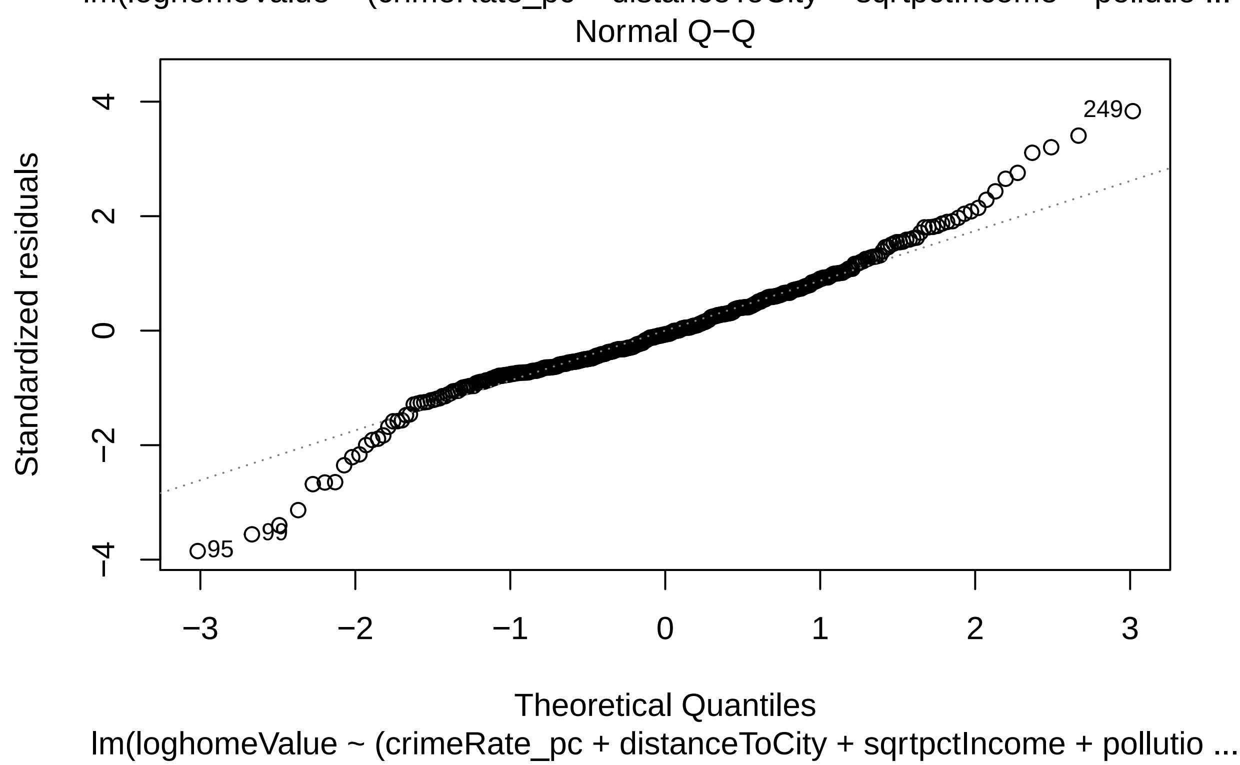

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(homeValue ~ sqrtcrimeRate_pc + distanceToCity + sqrtpctIncome + pollutio ...
250
249
278
−2
0
2
4
−3
−2
−1
0
1
2
3

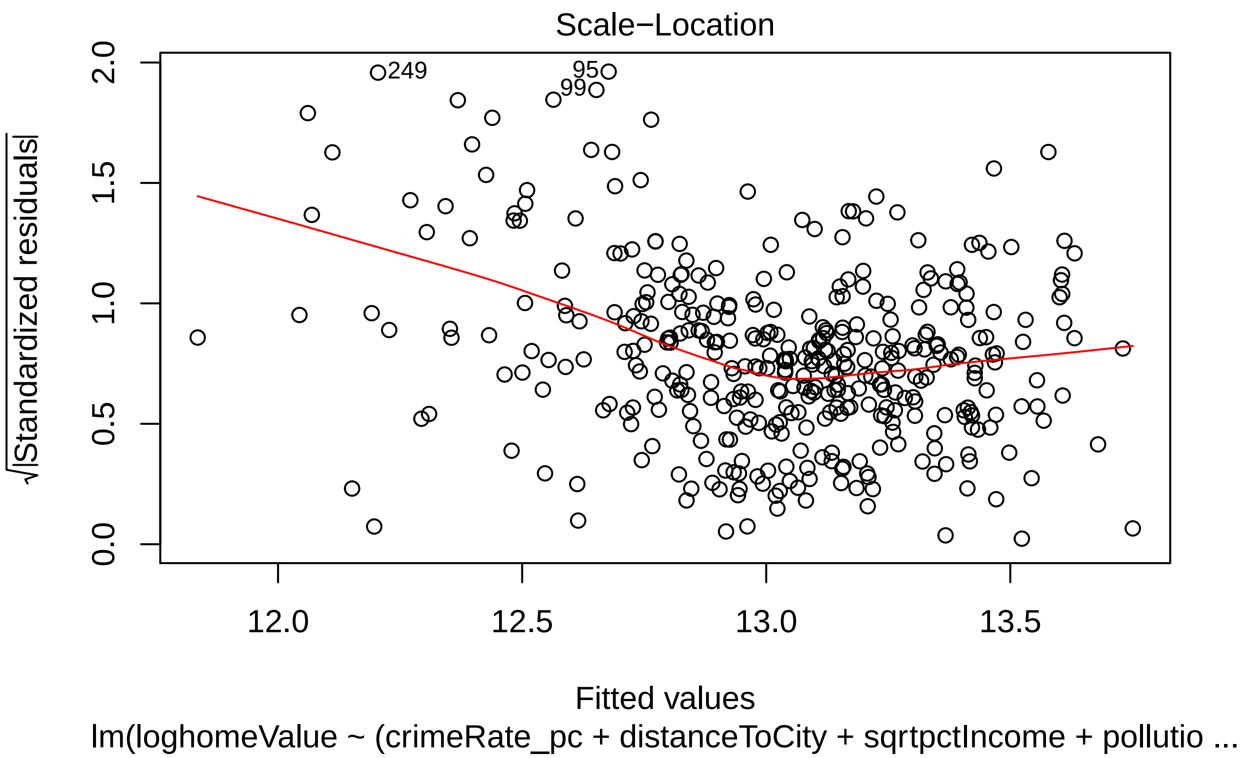

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
249
250
278
0e+00
2e+05
4e+05
6e+05
8e+05
Fitted values
lm(homeValue ~ sqrtcrimeRate_pc + distanceToCity + sqrtpctIncome + pollutio ...

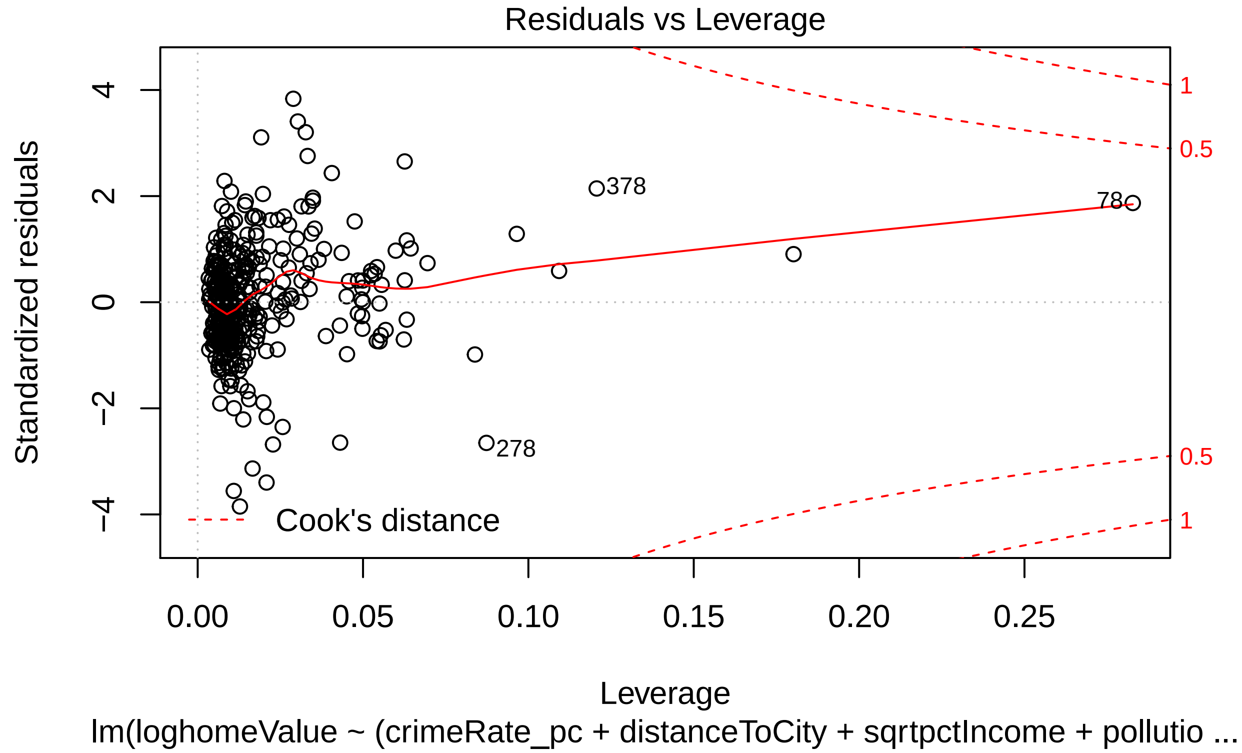

Residuals vs Leverage
Standardized residuals
−4
−2
0
2
4
72
378
278
Cook's distance
0.00
0.02
0.04
0.06
0.08
0.10
0.12
Leverage
lm(homeValue ~ sqrtcrimeRate_pc + distanceToCity + sqrtpctIncome + pollutio ...

```
plot(lmlog)
```


Residuals vs Fitted
249
99
95
0.5
0.0
−0.5
Residuals
12.0
12.5
13.0
13.5
Fitted values
lm(loghomeValue ~ (crimeRate_pc + distanceToCity + sqrtpctIncome + pollutio ...


Normal Q−Q
249
99
95
4
2
0
−2
−4
Standardized residuals
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(loghomeValue ~ (crimeRate_pc + distanceToCity + sqrtpctIncome + pollutio ...





There still seems to be heteroscedasticity. However, no amount of other transformation helped. Other transformations that were considered included, rounding the number of bedrooms to the nearest whole number, taking the log of distancetoCity and other binary considerations.

The log home value model is the one we will select to answer the questions of the group. However, due to the heteroscedasticity, we will need to ensure we are using robust standard errors.

```r
lmlog$newse<-vcovHC(lmlog)
coeftest(lmlog,lmlog$newse)
```

```
##
## t test of coefficients:
##
##                   Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)    13.7624078  0.1889175  72.8488 < 2.2e-16 ***
## crimeRate_pc   -0.0097852  0.0018920  -5.1720 3.733e-07 ***
## distanceToCity -0.0067690  0.0014107  -4.7983 2.293e-06 ***
## sqrtpctIncome  -0.2025856  0.0197261 -10.2699 < 2.2e-16 ***
## pollutionIndex -0.0052387  0.0014306  -3.6619 0.0002853 ***
## nBedRooms       0.0829979  0.0292939   2.8333 0.0048497 **
## withWater1      0.0996904  0.0333663   2.9878 0.0029903 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Specifically, the group wanted to know how environmental features affect the value of a home. There are two variables in our model that address this, the binary withWater variable and the pollution index.

Because we are using a log scale for home Value, we have to interpret this as follows.

The neighborhood being within 5 miles of water increases the value of the home 8.3% versus not being in that proximitiy.

For every one unit increase in the pollutionIndex as it is calculated, the value of the home descreases by 0.5%.

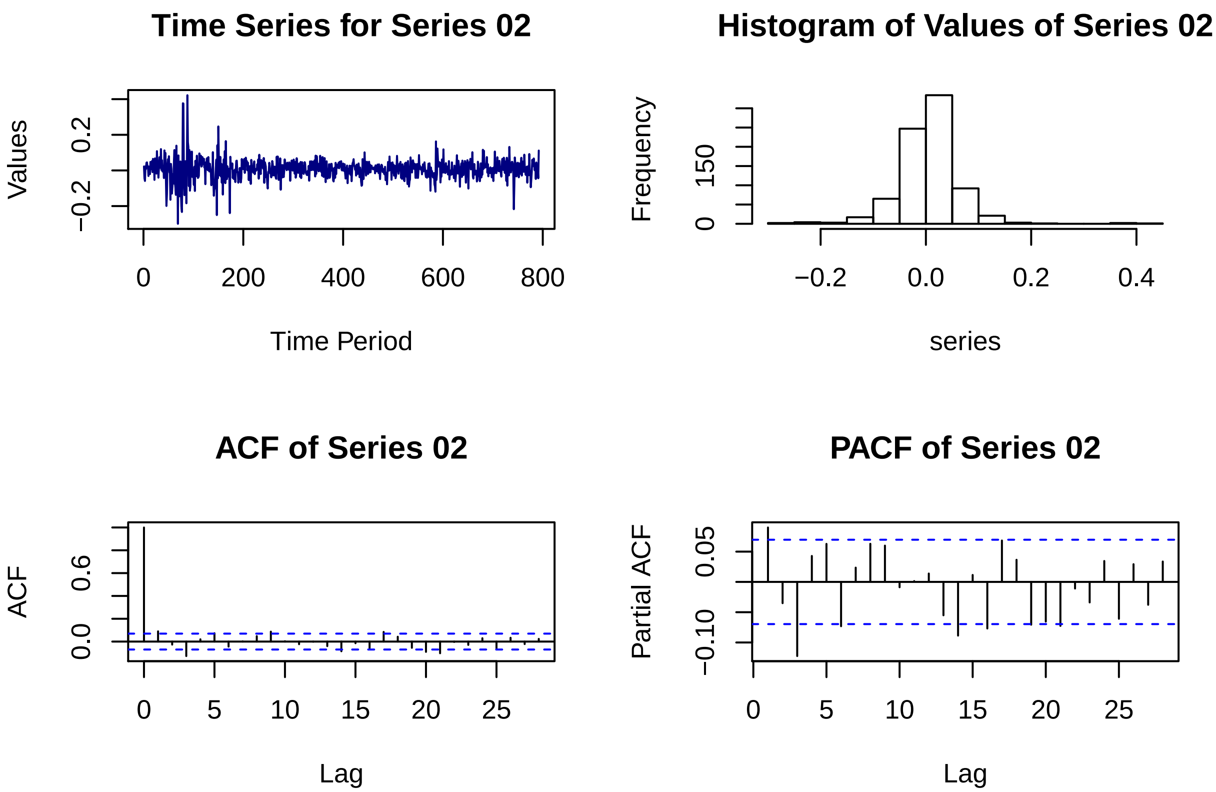
The other variable that positively increases the value of the home is the number of bedrooms. For each additional bedroom, the value of the home increases 10%.

sqrt of home value

Question 2

Build a time-series model for the series in series02.txt and use it to perform a 24-step ahead forecast.

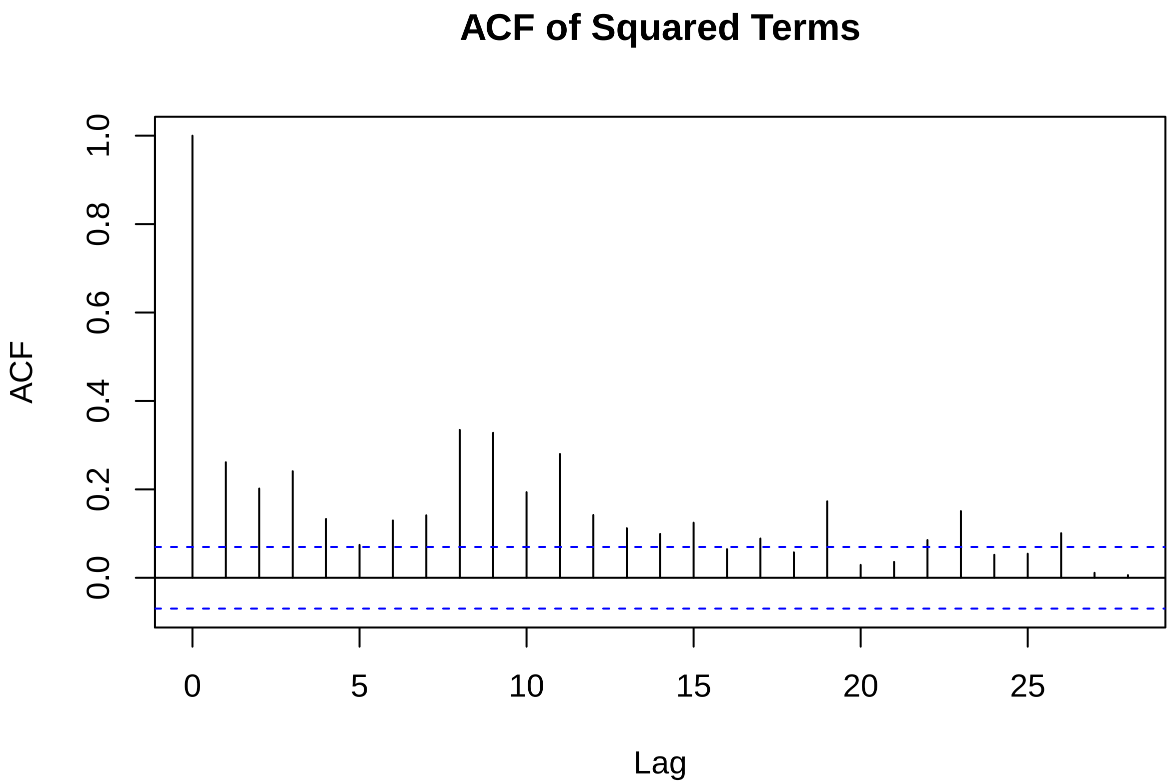
```r
series <- read.table("series02.txt")
series <- ts(series$V1)

#Visualize the data
par(mfrow = c(2,2))
plot.ts(series, col = "navy", xlab = "Time Period", ylab = "Values", main = "Time Series for Series 02")
hist(series, main = "Histogram of Values of Series 02")
acf(series, main = "ACF of Series 02")
pacf(series, main = "PACF of Series 02")
```



Notice the general structure of the series. There seems to be a long run average, where the values are fluctuating around a central axis but with with a major series of spikes in the beginning signaling serious volatility. There does not seem to be seasonality or a trend. The ACF interestingly shows a sharp drop after the 0 lag, but slightly statistically significant lags throughout the series. The PACF also shows slight significance at several lags after the most significant at what looks like the 3rd lag.

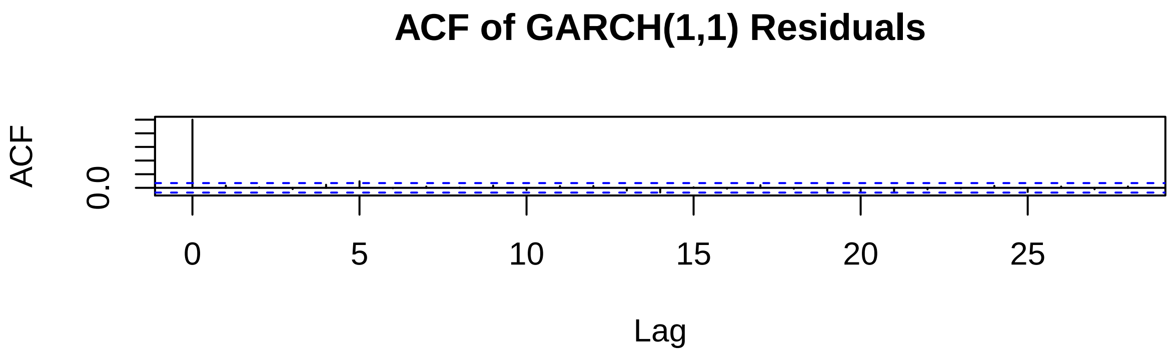
We suspect there is non-constant variance present in this series, so we will plot a correlogram of the squared values of a mean adjusted version of this series (adjusted so the mean is zero).

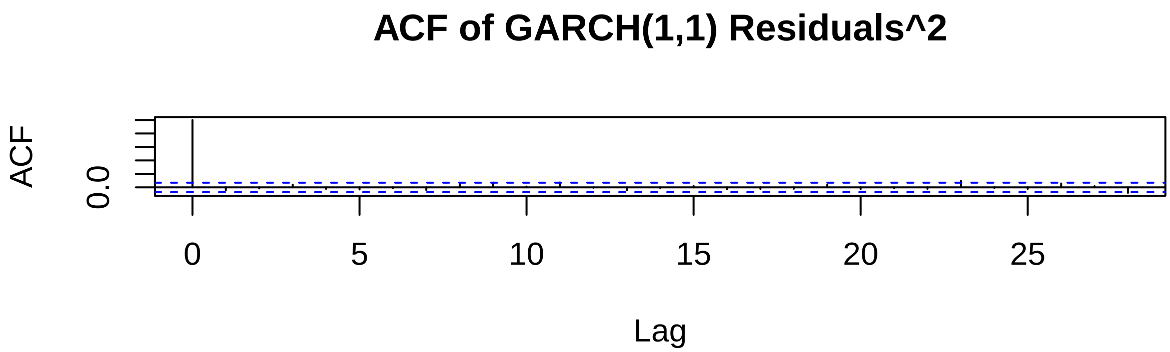
```
par(mfrow = c(1,1))
acf((series - mean(series))^2, main = "ACF of Squared Terms")
```



The square values that are plotted are equivalent to the variance. What the statistically significant values indicate is that there is serial correlation, meaning conditional heteroskedasticity. In plain English, this means that the variance is not constant throughout the series, rather the variance depends on what window of time we are looking at. This violates a core assumption of stationarity, meaning we will have to use a non-stationary model to fit this data.

```
garch1 <- garch(series, trace = F)
#garch1 <- garchFit(~garch(1,1), include.mean = FALSE, data = series, trace = FALSE)
#g.fit <- garchFit(~garch(1,1),data = resids)
resids <- residuals(garch1)[-1] #first value is NA

#Currently give different values

par(mfrow = c(2,1))
acf(resids, main = "ACF of GARCH(1,1) Residuals")
acf(resids^2, main = "ACF of GARCH(1,1) Residuals^2")
```






```
Box.test(resids, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resids
## X-squared = 0.88996, df = 1, p-value = 0.3455
```

```
garch1
```

```
##
## Call:
## garch(x = series, trace = F)
##
## Coefficient(s):
##        a0         a1         b1
## 7.785e-05  1.155e-01  8.617e-01
```

```
t(confint(garch1))
```

```
##                  a0         a1        b1
## 2.5 %  2.886309e-05 0.07712201 0.8231187
## 97.5 % 1.268441e-04 0.15385823 0.9002210
```

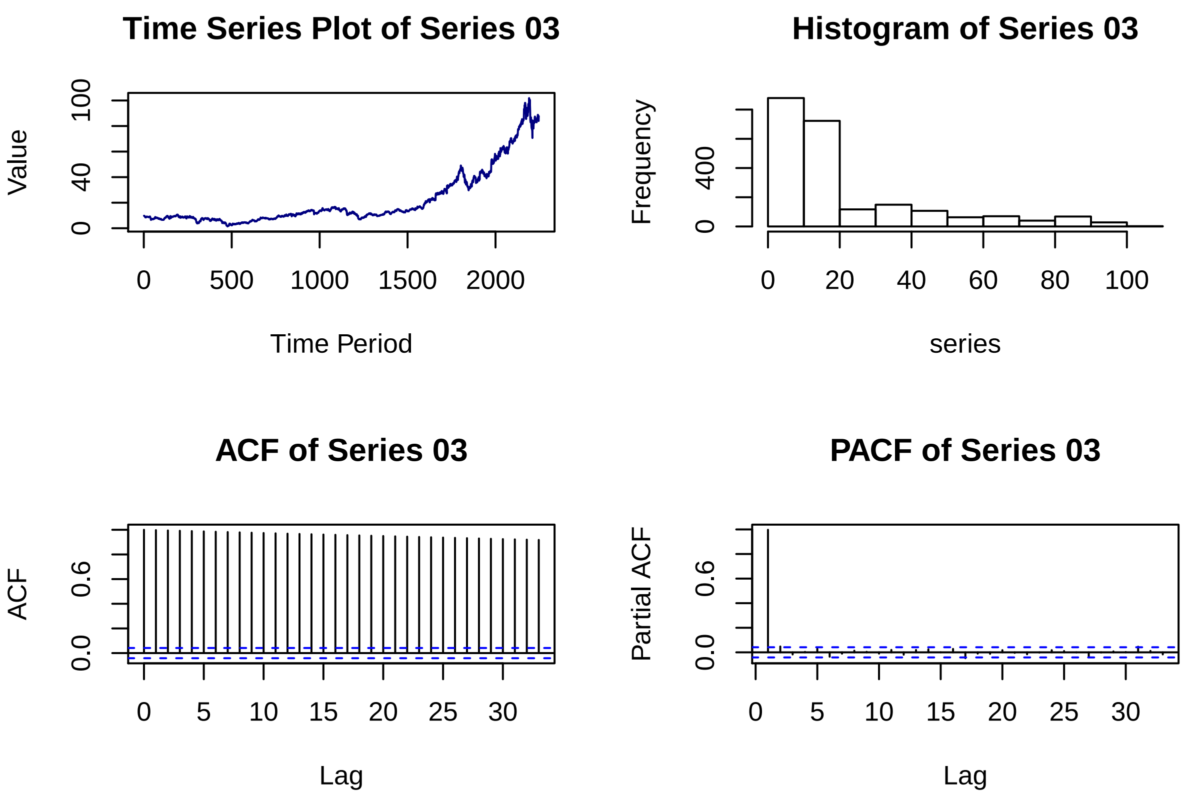
Notice that with the ACF of both the series residuals and squared residuals there is no autocorrelation. This suggests the residuals are behaving like white noise and thus the model is a good fit. Examining the model itself 0 is not contained in the 95% confidence intervals for the coefficients, meaning that the coefficients are satatistically significant at the 95% level.

Forecasting

Question 3

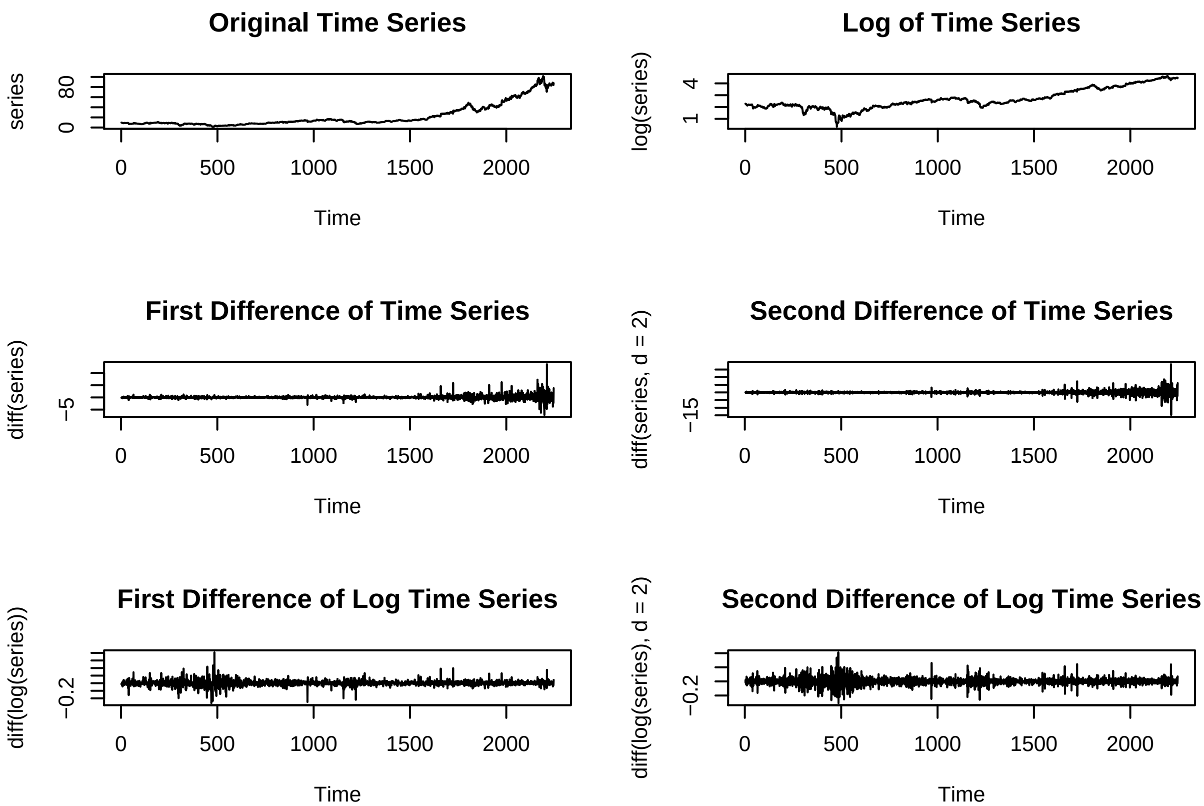
Build a time-series model for the series in series03.csv and use it to perform a 24-step ahead forecast

```
#load data and do preliminary visualization
series <- read.csv("series03.csv")
series <- ts(series$X9.88)

par(mfrow = c(2,2))
plot.ts(series, xlab = "Time Period", ylab = "Value", main = "Time Series Plot of Series 03", col = "na
hist(series, main = "Histogram of Series 03")
acf(series, main = "ACF of Series 03")
pacf(series, main = "PACF of Series 03")
```



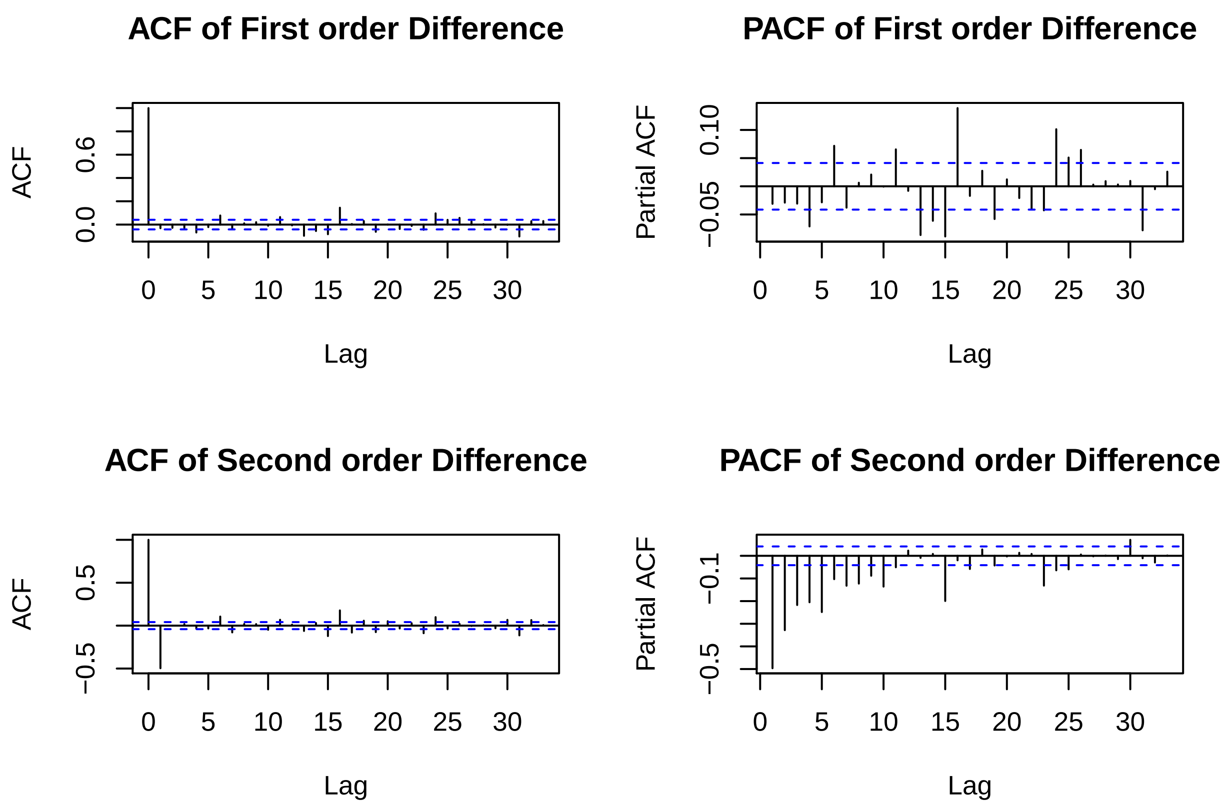
Notice from the time series plot that there is significant trend going on, long term upwards. The ACF shows significance through all past lags while the PACF is only siginficant for the first lag. There does not seem to be any seasonality. This looks like the realization of a random walk with drift process.

```
#plot different time series to suggest differencing
par(mfrow = c(3, 2))
plot.ts(series, main = "Original Time Series")
plot.ts(log(series), main = "Log of Time Series")
plot.ts(diff(series), main = "First Difference of Time Series")
plot.ts(diff(series, d = 2), main = "Second Difference of Time Series")
plot.ts(diff(log(series)), main = "First Difference of Log Time Series")
plot.ts(diff(log(series), d = 2), main = "Second Difference of Log Time Series")
```

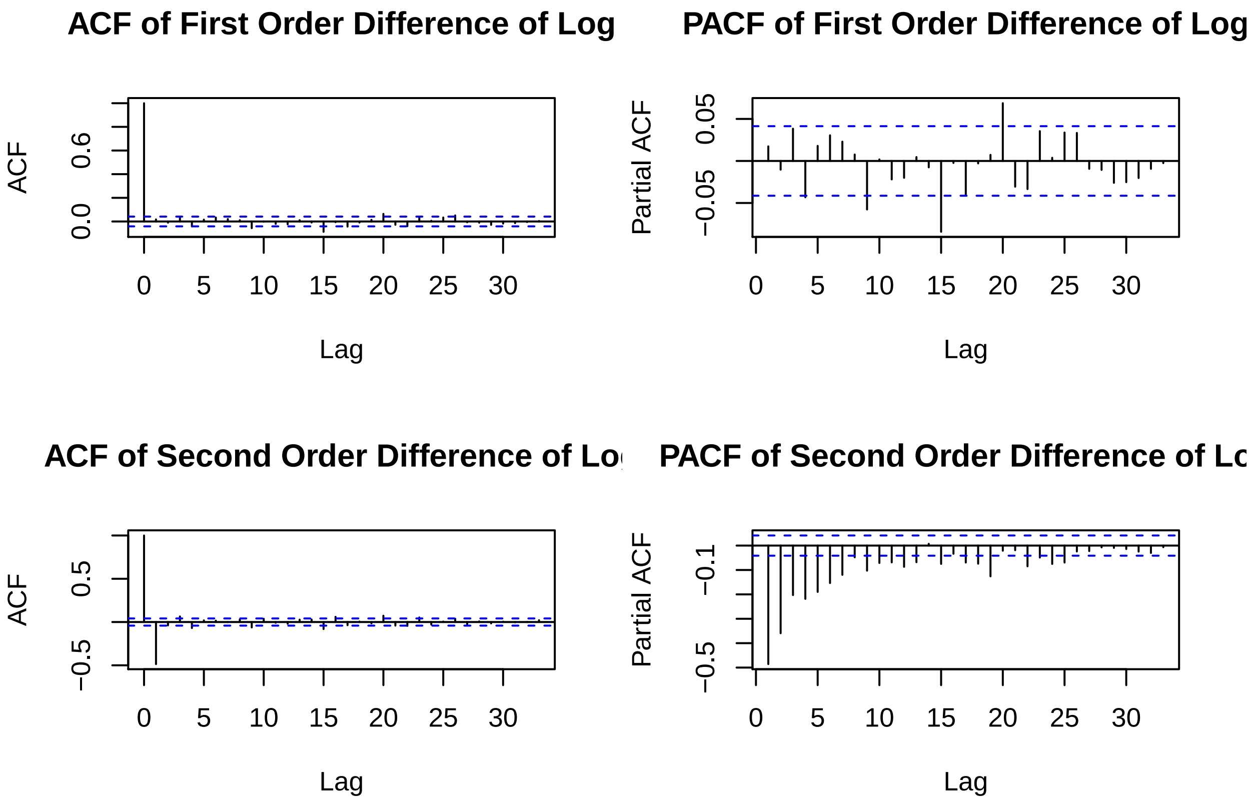


It is clear from the original time series plot that the series is not stationary. Before proceeding to build a model we must render the series as stationary.

```
par(mfrow = c(2,2))
acf(diff(series), main = "ACF of First order Difference")
pacf(diff(series), main = "PACF of First order Difference")
acf(diff(series, d= 2), main = "ACF of Second order Difference")
pacf(diff(series, d = 2), main = "PACF of Second order Difference")
```




```r
acf(diff(log(series)), main = "ACF of First Order Difference of Log")
pacf(diff(log(series)), main = "PACF of First Order Difference of Log")
acf(diff(log(series), d = 2), main = "ACF of Second Order Difference of Log")
pacf(diff(log(series), d = 2), main = "PACF of Second Order Difference of Log")
```



From examining these plots, it seems as though the second order difference provides the best transformation into white noise. In both cases the ACF shows a sharp cut off (suggesting an MA term) while the PACF gradually declines. The first order difference shows a lot of volatility in the PACF, suggesting correlations that are not easily captured. Between the second order difference and the second order difference of the log, the second order difference of the log seems to look more like white noise. There are fewer significant autocorrelations (which might be due to sampling) in the second order difference of the log and it decays more smoothly. Therefore, we will use the second order difference of the log to estimate the model.

```
get.best.arima <- function(x.ts, maxord = c(1,1,1))
{
  best.aic <- 1e8
  n <- length(x.ts)
  for (p in 0:maxord[1]) for (d in 0:maxord[2]) for (q in 0:maxord[3])
  {
    fit <- arima(x.ts, order = c(p, d, q), method = "ML")
    fit.aic <- -2 * fit$loglik + (log(n) + 1) * length(fit$coef)
    if (fit.aic < best.aic)
    {
      best.aic <- fit.aic
      best.fit <- fit
      best.model <- c(p, d, q)
    }
  }
  list(best.aic, best.fit, best.model)
}

auto.arima(log(series), allowdrift = FALSE)
```

```
## Series: log(series)
```

```
## ARIMA(0,1,0)
##
## sigma^2 estimated as 0.001456:  log likelihood=4146.46
## AIC=-8290.91   AICc=-8290.91   BIC=-8285.2
```

```
mod <- auto.arima(log(series), d = 2)
```

```
## Warning in auto.arima(log(series), d = 2): Unable to fit final model using
## maximum likelihood. AIC value approximated
```

```
mod
```

```
## Series: log(series)
## ARIMA(2,2,1)
##
## Coefficients:
##          ar1      ar2      ma1
##       0.0139  -0.0120  -0.9886
## s.e.  0.0212   0.0213   0.0030
##
## sigma^2 estimated as 0.001476:  log likelihood=4129.66
## AIC=-8238.81   AICc=-8238.79   BIC=-8215.94
```

```
t(confint(mod))
```

```
##                ar1         ar2        ma1
## 2.5 %  -0.02770088 -0.05363241 -0.9944319
## 97.5 %  0.05550080  0.02970417 -0.9828398
```

Here we try using the auto.arima() function to find the best best. When using the auto.arima() function it suggests the first order difference of the log series. However, we saw above that this was not the best model examining the ACF and PACF so we instead specificed the order of differencing to be 2. When doing this, the suggested model is an ARIMA(2, 2, 1) model. However, examining the confidence intervals, we find that the 2 AR terms contain 0 in their confidence interval. That means we will fail to reject the null hypothesis these coefficients are 0. The MA term however does not contain 0 in its confidence interval and therefore we can reject the null hypothesis. Therefore, we will construct an ARIMA(0, 2, 1) model.

```
model <- arima(log(series), order = c(0, 2, 1))
model
```

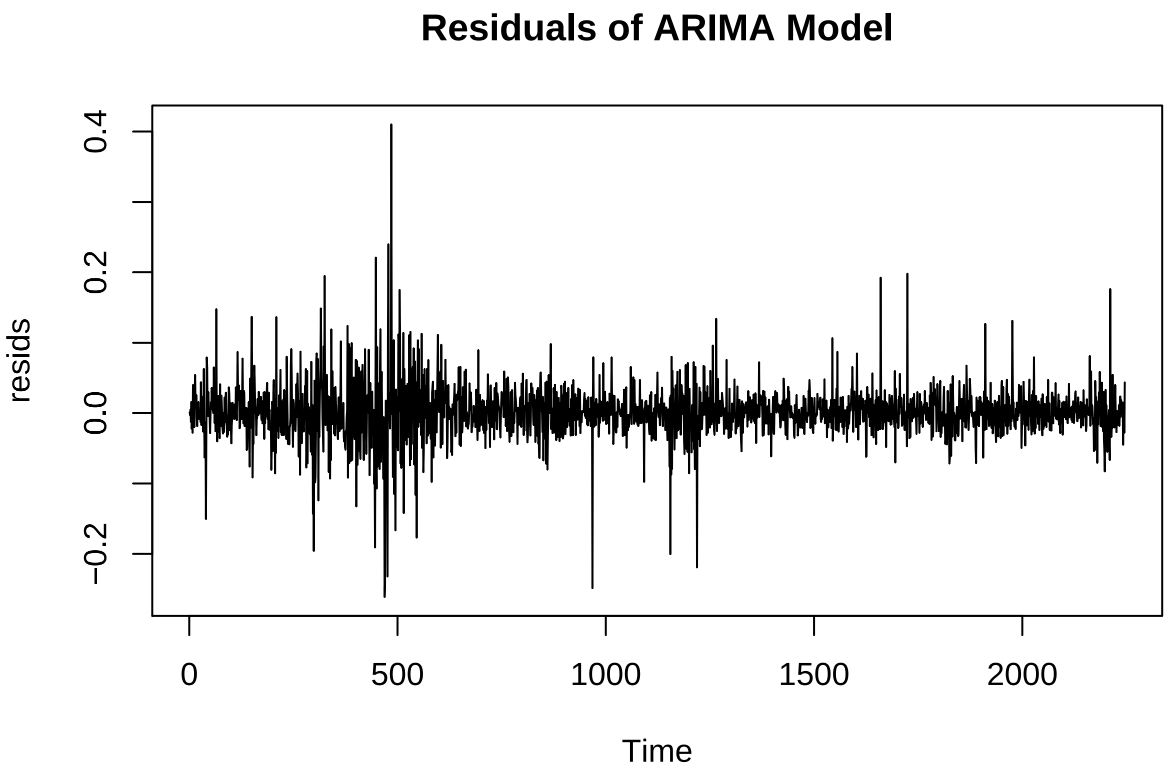
```
##
## Call:
## arima(x = log(series), order = c(0, 2, 1))
##
## Coefficients:
##           ma1
##       -0.9997
## s.e.   0.0026
##
## sigma^2 estimated as 0.001456:  log likelihood = 4141,  aic = -8278
```

```r
t(confint(model))
```

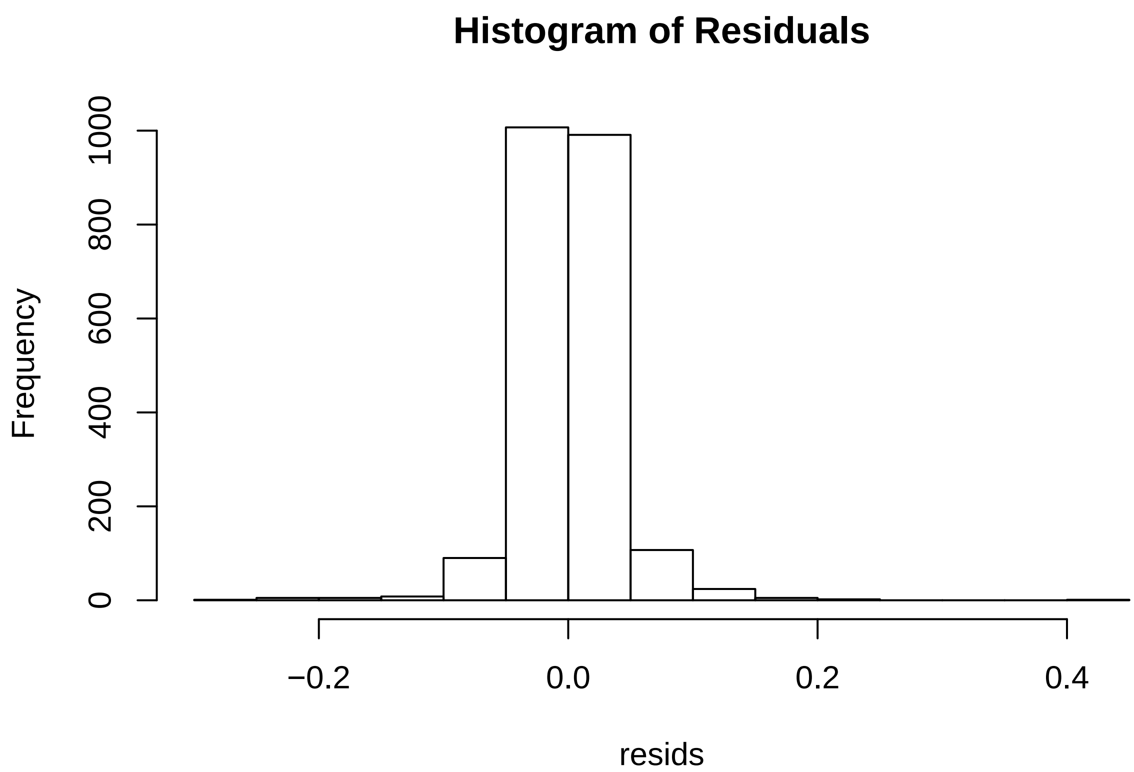
```
##               ma1
## 2.5 %  -1.0048193
## 97.5 % -0.9944951
```

0 is not contained in the confidence interval so this coefficient is statistically significant.

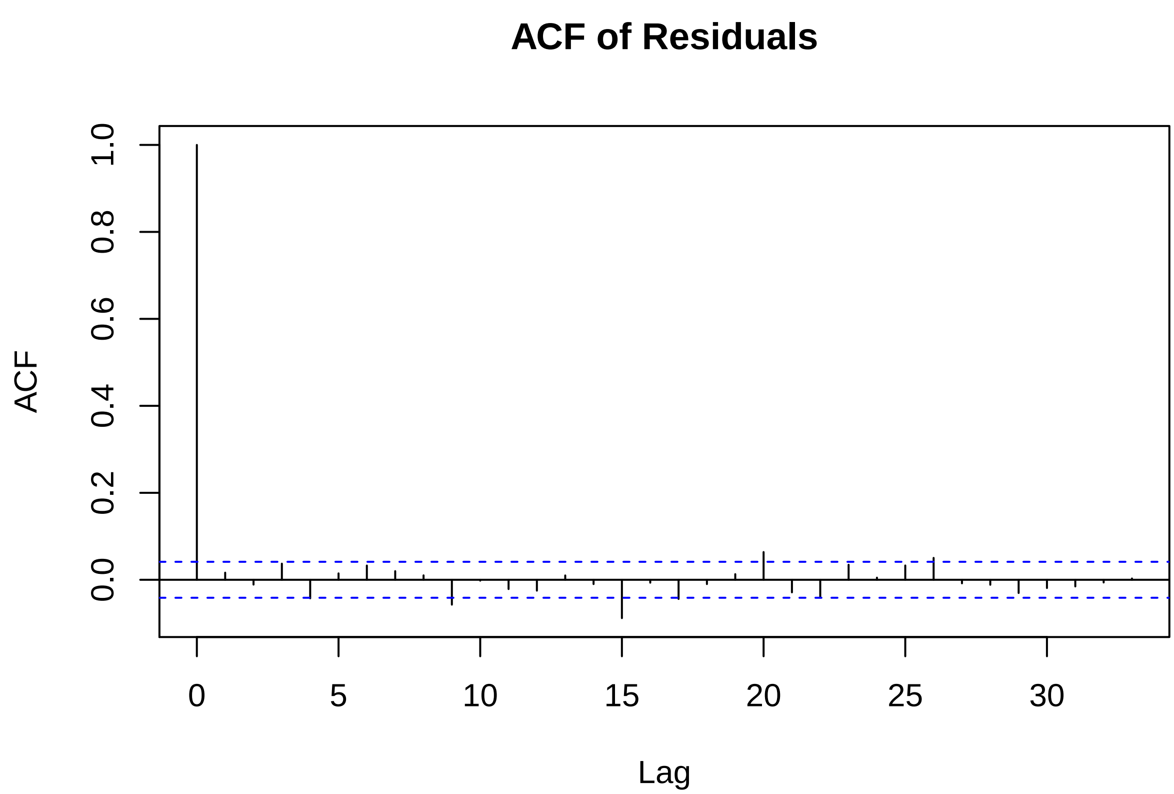
```r
#diagnostic plots of residuals
resids <- model$residuals
plot.ts(resids, main = "Residuals of ARIMA Model")
```



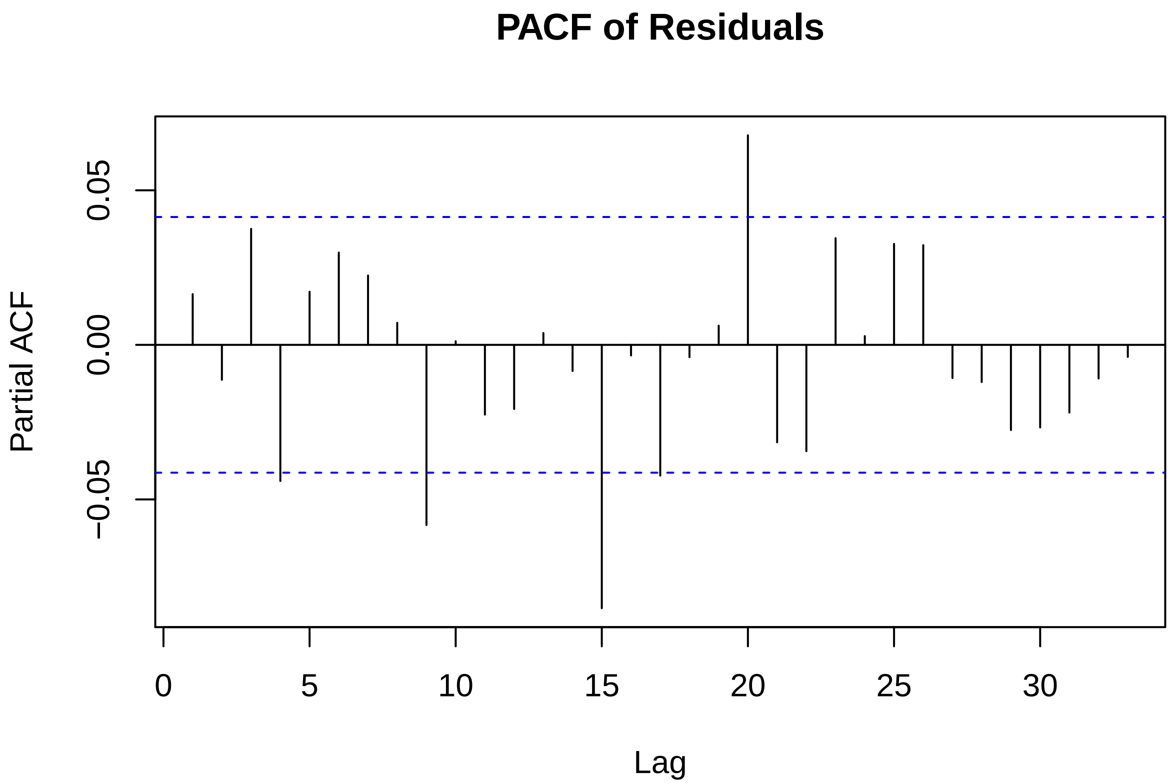

```r
hist(resids, main = "Histogram of Residuals")
```




```
acf(resids, main = "ACF of Residuals")
```

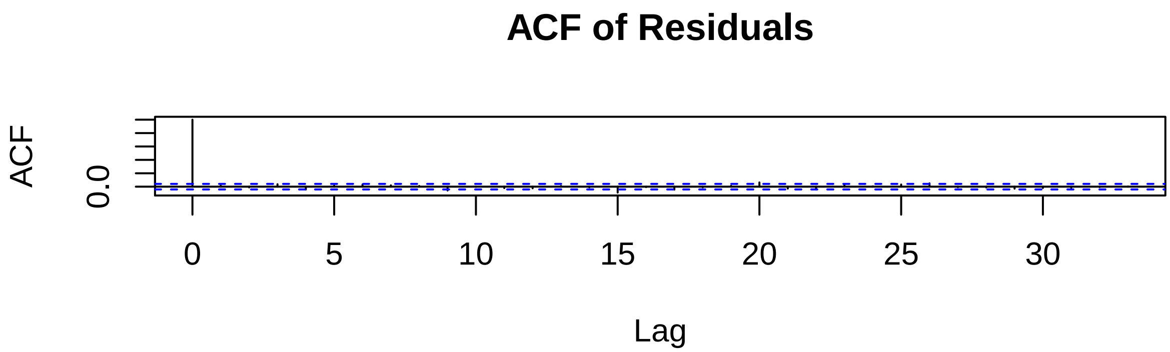


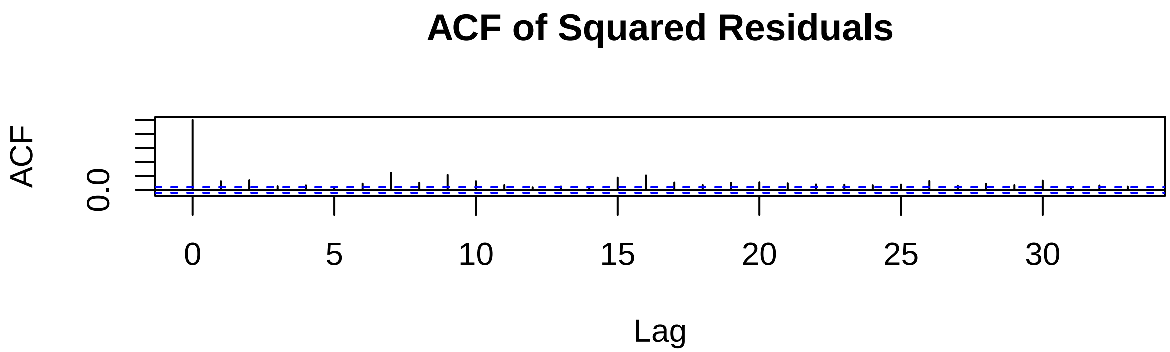
```r
pacf(resids, main = "PACF of Residuals")
```



These residual diagnostics suggest a reasonably good approximation of white noise. The ACF and PACF however do show quite a bit of volatility, so we will examine the squared residuals because we suspect there is non-constant variance.

```r
par(mfrow = c(2,1))
acf(resids, main = "ACF of Residuals")
acf(resids^2, main = "ACF of Squared Residuals")
```



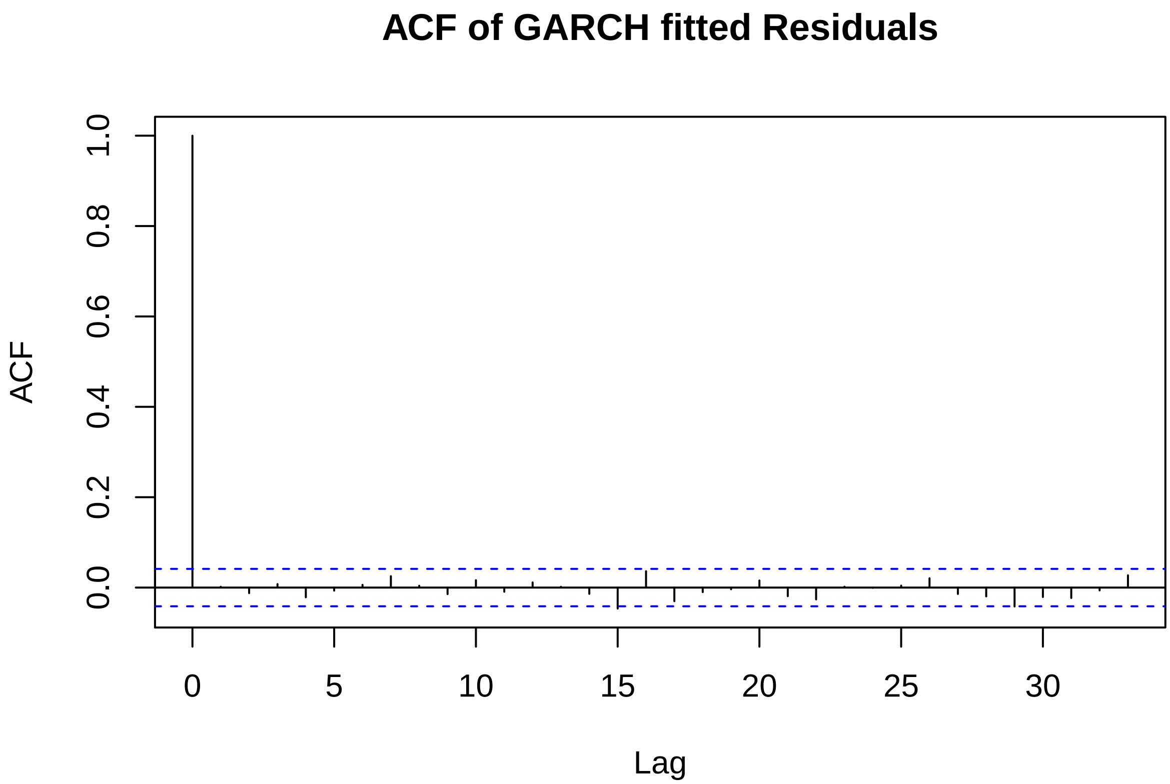


As we had suspected, the squared residuals show statistically significant terms at different intervals. Clearly, this suggests there is non-constant variance. Therefore, we will fit a GARCH model to the residuals.

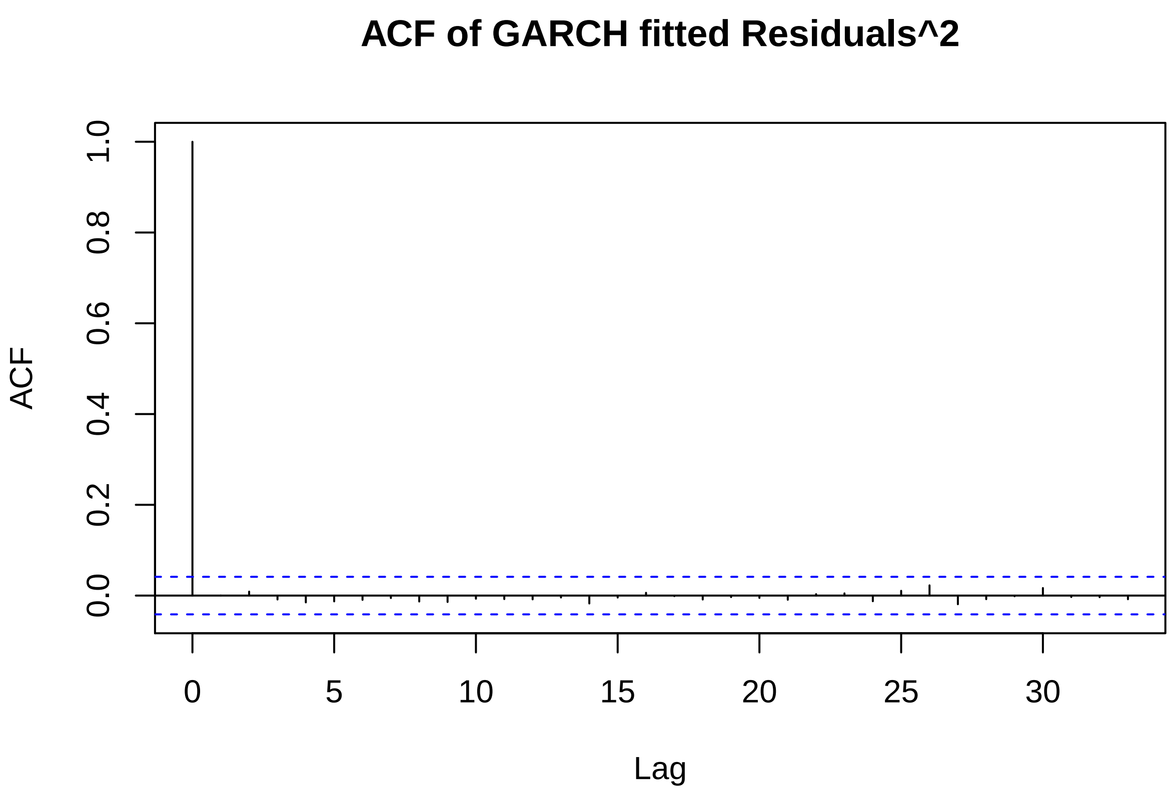
```
resid.garch = garch(resids, trace = FALSE)
t(confint(resid.garch))
```

```
##                   a0         a1        b1
## 2.5 %  4.554540e-05 0.06872135 0.8591772
## 97.5 % 6.374866e-05 0.09501795 0.8952889
```

```
acf(resid.garch$residuals[-1], main = "ACF of GARCH fitted Residuals")
```




```r
acf(resid.garch$residuals[-1]^2, main = "ACF of GARCH fitted Residuals^2")
```



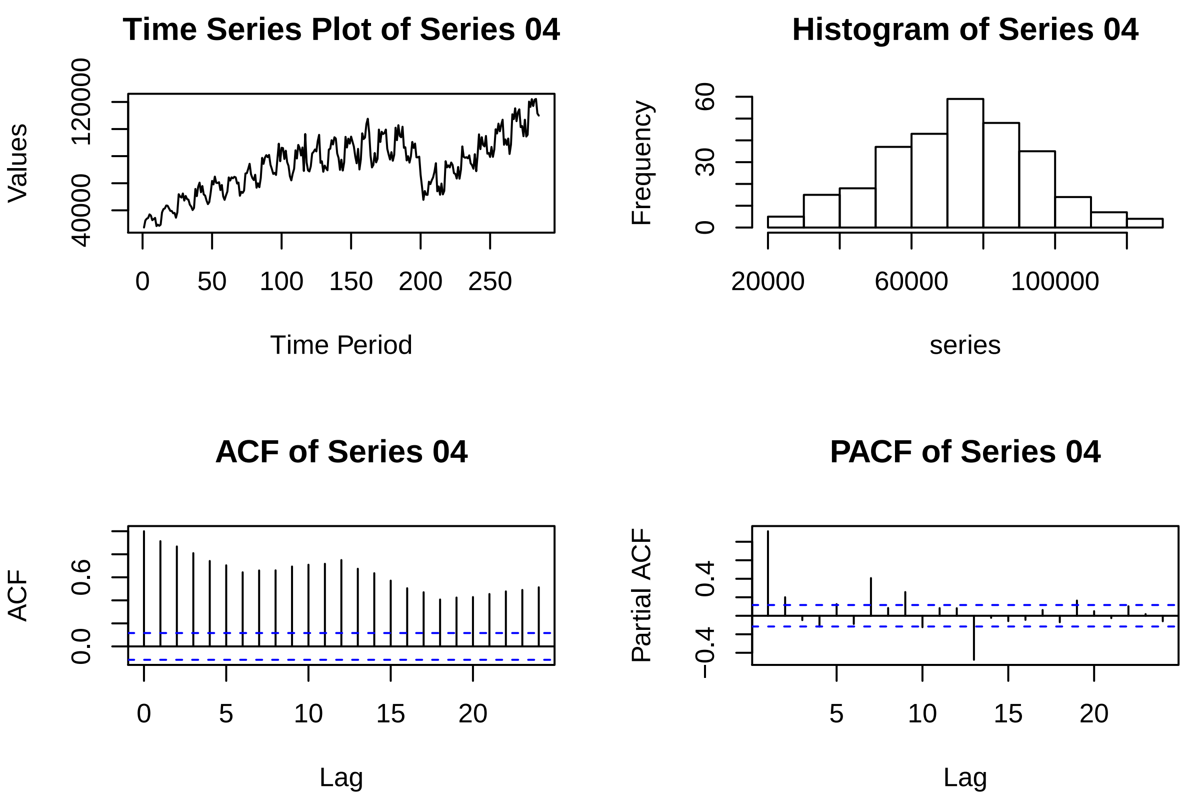
The GARCH model shows statistically significant coefficients at the 95% level, because 0 is not contained in the confidence intervals. The ACFs of both the GARCH residuals and residuals squared show no significant values, meaning a good fit to the residuals.

Forecast?

Question 4

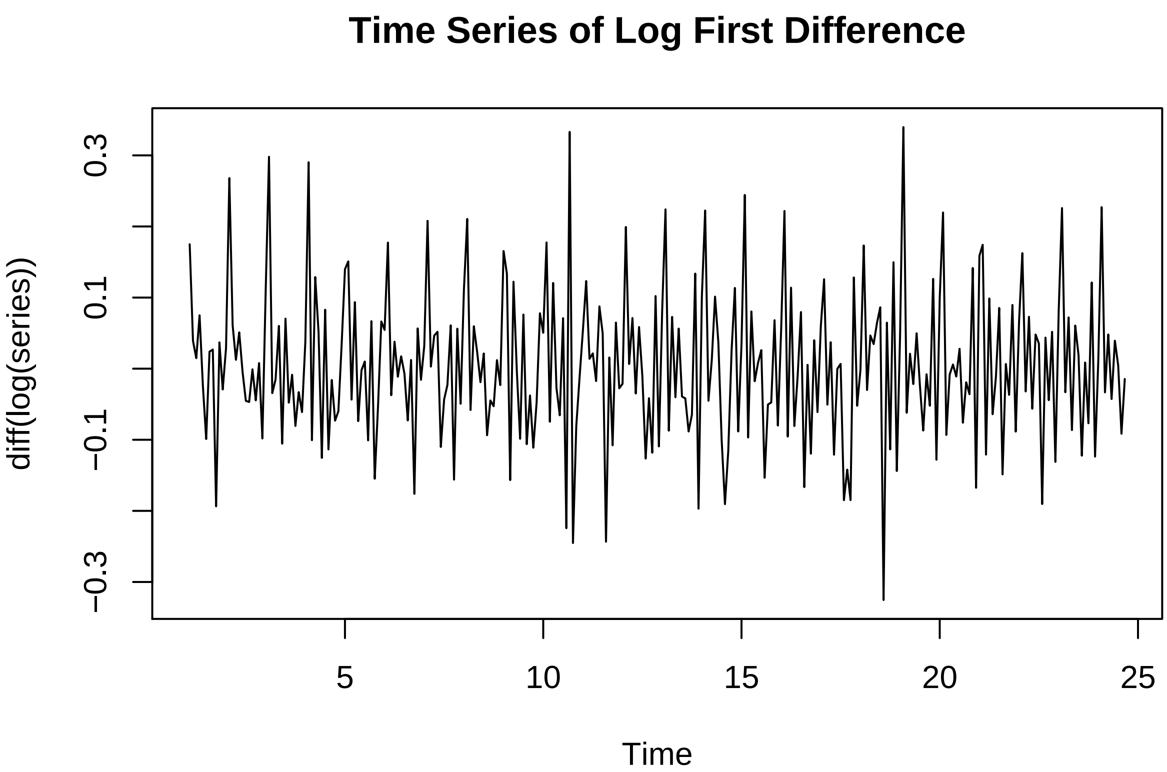
Build a time-series model for the series in series04.csv and use it to perform a 24-step ahead forecast. Possible models include AR, MA, ARMA, ARIMA, Seasonal ARIMA, GARCH, ARIMA-GARCH, or Seasonal ARIMA-GARCH models. Note that the original series may need to be transformed before it be modelled.

```
#import data and run basic visualizations
series <- read.csv("series04.csv")
series <- ts(series$X25182)

par(mfrow = c(2,2))
plot.ts(series, xlab = "Time Period", ylab = "Values", main = "Time Series Plot of Series 04")
hist(series, main = "Histogram of Series 04")
acf(series, main = "ACF of Series 04")
pacf(series, main = "PACF of Series 04")
```

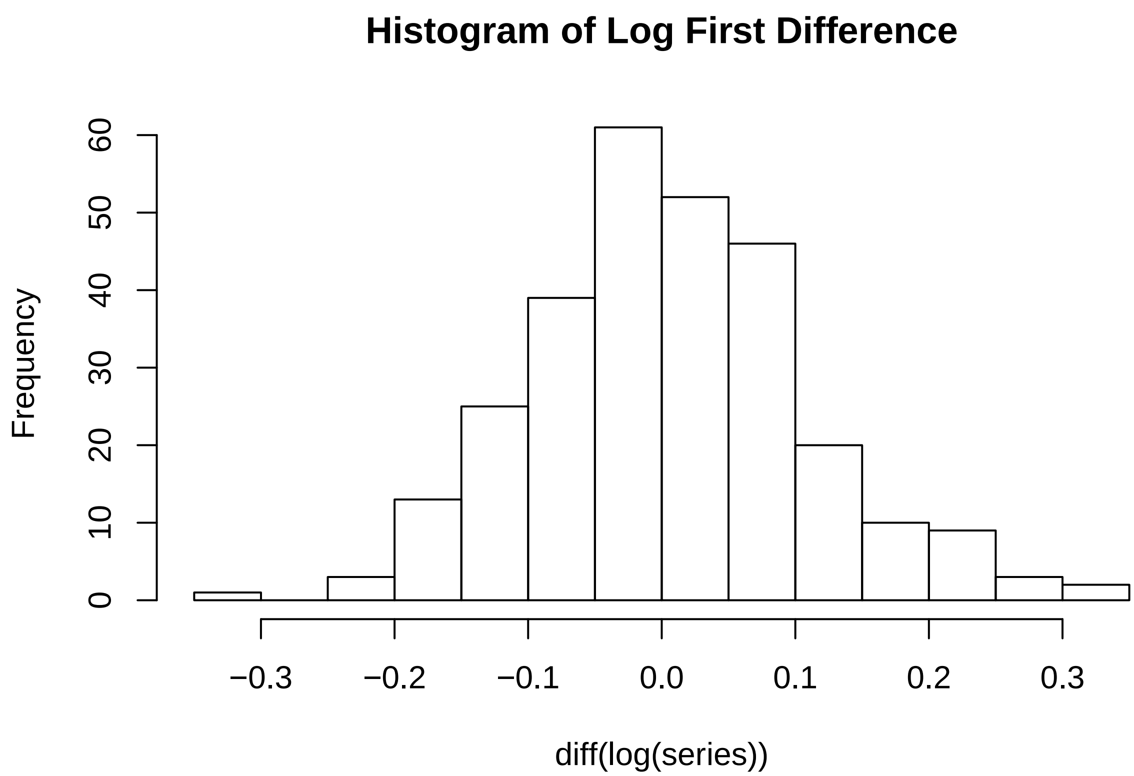


From the time series plot it should be obvious that there is seasonality in this series, suggesting seasonal lag terms will be needed. The series shows a general upwards trend, and we would argue this series is definitely not stationary. The ACF show statistically significant lags persisting but at different heights, further suggesting non-stationarity and seasonality.

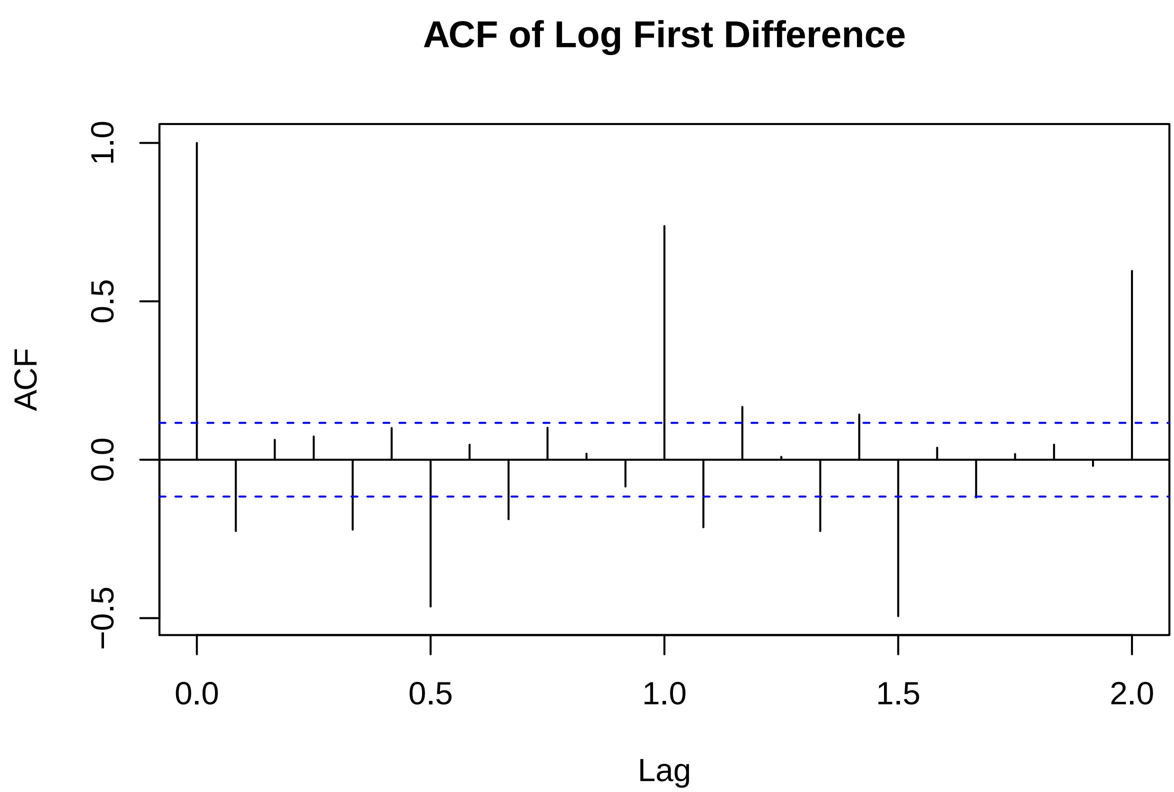
```r
#Transform series to make stationary
series <- read.csv("series04.csv")
series <- ts(series$X25182, frequency = 12)

plot.ts(diff(log(series)), main = "Time Series of Log First Difference")
```



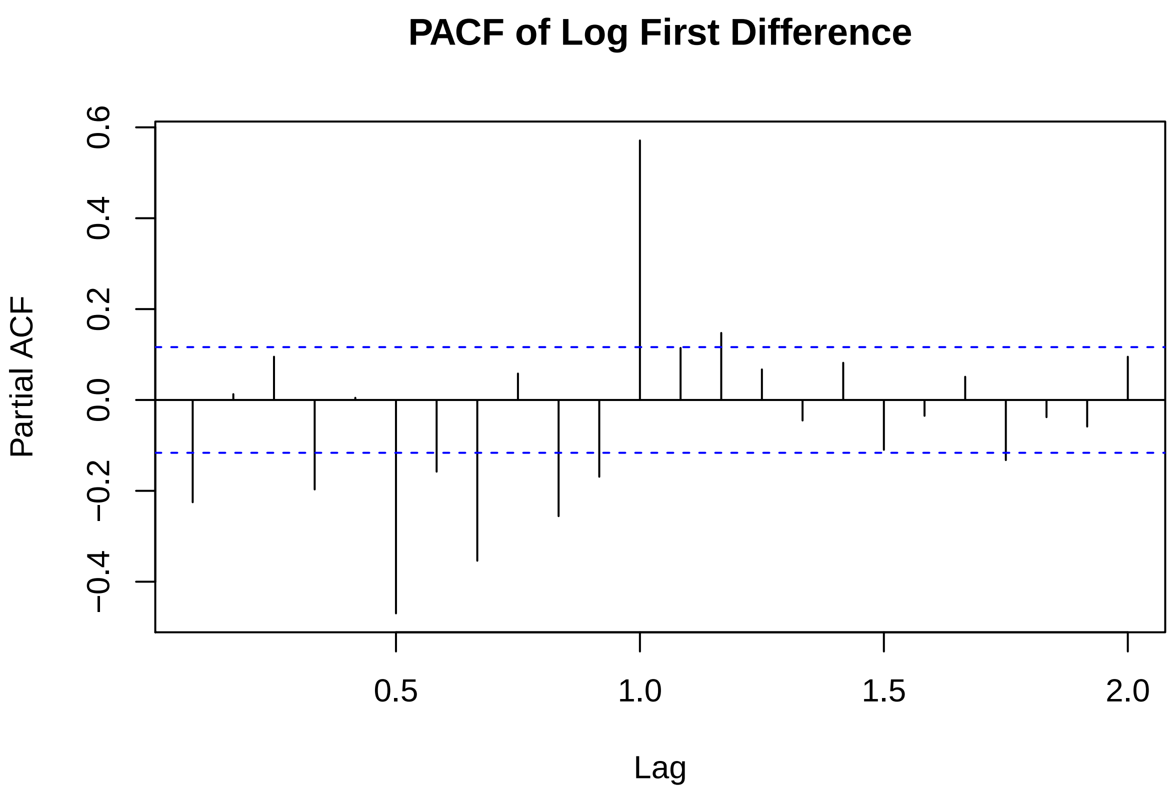

```r
hist(diff(log(series)), main = "Histogram of Log First Difference")
```




```
acf(diff(log(series)), main = "ACF of Log First Difference")
```



```
pacf(diff(log(series)), main = "PACF of Log First Difference")
```



We are reimporting the series and setting the frequency to 12. We suspect the seasonality occurs on a monthly basis and counted 24 trough to peak cycles, indicating a seasonal period of 12. As is generally good practice we will take the log of the series and take the first difference to render the series more stationary. The time series plot of the differenced series resembles white noise. However, the ACF shows regular significance suggesting seasonal terms will be needed there. The PACF also shows seasonality although somewhat less as it decreases eventually. Both plots also show significance at the first lag suggesting non-seasonal terms will also be needed.

```
#function to find the best arima model. Credit to Cowpertwait and Metcalfe
get.best.arima.seas <- function(x.ts, maxord = c(1,1,1,1,1,1)) {
  best.aic <- 1e8
  n <- length(x.ts)
  for (p in 0:maxord[1]) for(d in 0:maxord[2]) for(q in 0:maxord[3])
    for (P in 0:maxord[4]) for(D in 0:maxord[5]) for(Q in maxord[6])
    {
      fit <- arima(x.ts, order = c(p, d, q), seas = list(order = c(P,D,Q), 12), method = "CSS")
      fit.aic <- -2 * fit$loglik + (log(n) + 1) * length(fit$coef)
      if (fit.aic < best.aic)
      {
        best.aic <- fit.aic
        best.fit <- fit
        best.model <- c(p, d, q, P, D, Q)
      }
    }
  list(best.aic, best.fit, best.model)
}
```

```r
get.best.arima.seas(log(series), maxord = rep(3, 6))
```

```
## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1
```

```
## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## Warning in arima(x.ts, order = c(p, d, q), seas = list(order = c(P, D,
## Q), : possible convergence problem: optim gave code = 1

## [[1]]
## [1] -811.9272
##
## [[2]]
##
## Call:
## arima(x = x.ts, order = c(p, d, q), seasonal = list(order = c(P, D, Q), 12),
##     method = "CSS")
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##           ar1     ar2     ar3     ma1     ma2      ma3    sar1    sar2
##       -0.1864  0.1633  0.9602  0.8195  0.4838  -0.4573  0.6859  0.2926
## s.e.   0.0059  0.0021  0.0098  0.0081     NaN      NaN  0.0671  0.0702
##          sma1     sma2    sma3  intercept
##       -0.3415  -0.3488  0.0524    11.1926
## s.e.   0.0862   0.0749  0.0606     2.0218
##
## sigma^2 estimated as 0.002562:  part log likelihood = 445.88
##
## [[3]]
## [1] 3 0 3 2 0 3
```

```
auto.arima(log(series), d = 1, D = 1) #note specified the use of seasonality
```

```
## Series: log(series)
## ARIMA(2,1,0)(2,1,2)[12]
##
## Coefficients:
##           ar1      ar2    sar1     sar2     sma1    sma2
##       -0.3788  -0.2638  0.5919  -0.2884  -1.2183  0.4125
## s.e.   0.0620   0.0601  0.2195   0.0892   0.2195  0.1862
##
## sigma^2 estimated as 0.003037:  log likelihood=396.79
## AIC=-779.57   AICc=-779.15   BIC=-754.33
```

```
#get.best -> (3, 0, 3) (2, 0, 3) [12]
#auto -> (2, 1, 0) (2, 1, 2) [12]

mod <- auto.arima(log(series), d = 1, D = 1)
mod2 <- arima(log(series), order = c(3, 0, 3), seasonal = list(order = c(2, 0, 3), 12))
mod
```

```
## Series: log(series)
## ARIMA(2,1,0)(2,1,2)[12]
##
## Coefficients:
##           ar1      ar2    sar1     sar2     sma1    sma2
##       -0.3788  -0.2638  0.5919  -0.2884  -1.2183  0.4125
## s.e.   0.0620   0.0601  0.2195   0.0892   0.2195  0.1862
##
## sigma^2 estimated as 0.003037:  log likelihood=396.79
## AIC=-779.57   AICc=-779.15   BIC=-754.33
```

```
mod2
```

```
##
## Call:
## arima(x = log(series), order = c(3, 0, 3), seasonal = list(order = c(2, 0, 3),
##     12))
##
```

```
## Coefficients:
##           ar1     ar2     ar3     ma1     ma2      ma3    sar1    sar2
##       -0.1859  0.1848  0.9720  0.8376  0.4928  -0.4074  0.8325  0.1519
## s.e.   0.0131  0.0126  0.0124  0.0598  0.0755   0.0587  0.6131  0.6077
##          sma1     sma2    sma3  intercept
##       -0.5152  -0.2522  0.0995    11.1842
## s.e.   0.6123   0.3912  0.1020     1.5779
##
## sigma^2 estimated as 0.002691:  log likelihood = 421.43,  aic = -816.86
```

We utilized both the auto.arima() function and the get.best.arima.seas() function (from the time series textbook) to acquire suggested model fits. However, we know the model should include a first difference and a seasonal difference from our previous investigation. Otherwise the model will not be stationary, and we will be unable to fit a model to it. The auto.arima() model's AIC is slightly higher -779.5714051 versus -816.8591981, but we believe that the model suggested by auto arima will better satisfy our assumptions. Therefore, we will investigate this model going forward.

```r
t(confint(mod))
```

```
##                ar1        ar2      sar1       sar2       sma1       sma2
## 2.5 %  -0.5003736 -0.3816227 0.1616603 -0.4632530 -1.6484367 0.04750577
## 97.5 % -0.2572684 -0.1459414 1.0222092 -0.1134823 -0.7882052 0.77746222
```

```r
#comparison model is (2, 1, 0)(2, 1, 2)[12] with AIC -779.5714
mod3 <- arima(log(series), order = c(3, 1, 0), seasonal = list(order = c(2, 1, 2), 12))
mod4 <- arima(log(series), order = c(2, 1, 1), seasonal = list(order = c(2, 1, 2), 12))
mod5 <- arima(log(series), order = c(2, 1, 0), seasonal = list(order = c(3, 1, 2), 12))
mod6 <- arima(log(series), order = c(2, 1, 0), seasonal = list(order = c(2, 1, 3), 12))
AIC(mod3)
```

```
## [1] -778.6494
```

```r
AIC(mod4)
```

```
## [1] -779.2913
```

```r
AIC(mod5)
```

```
## [1] -775.8237
```

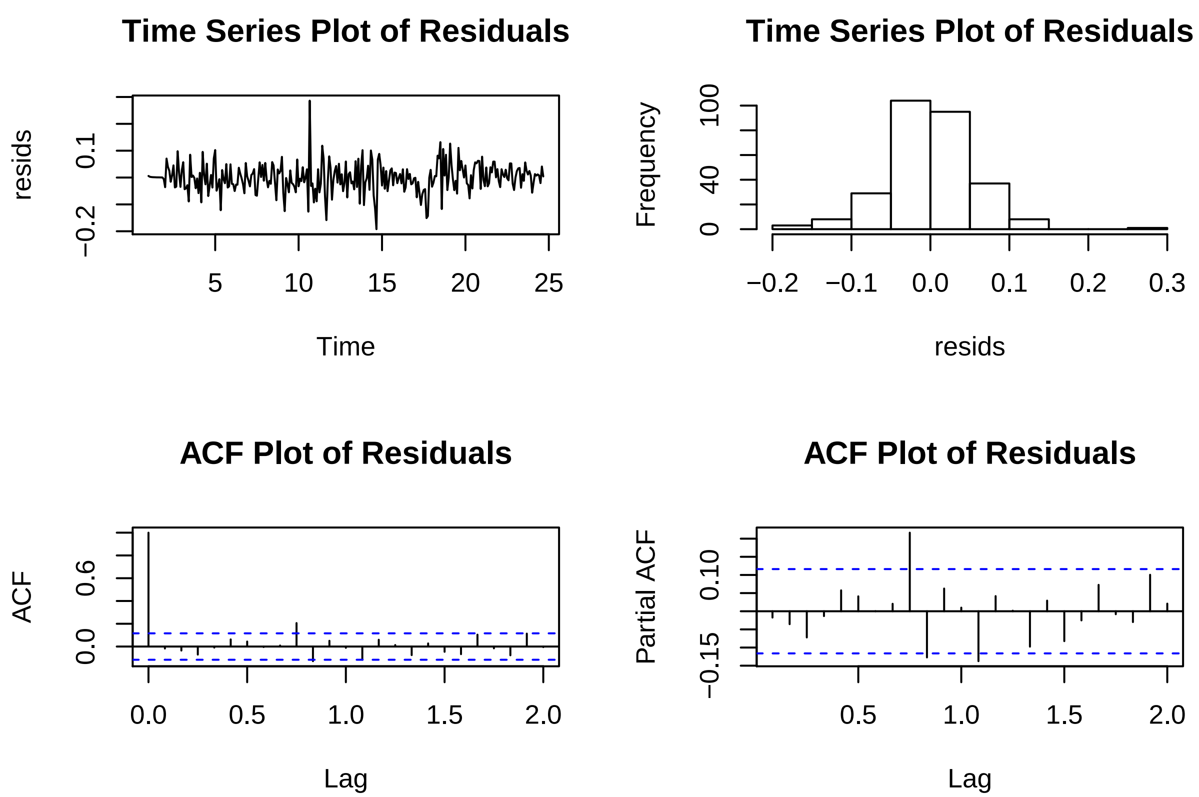
```r
AIC(mod6)
```

```
## [1] -774.8587
```

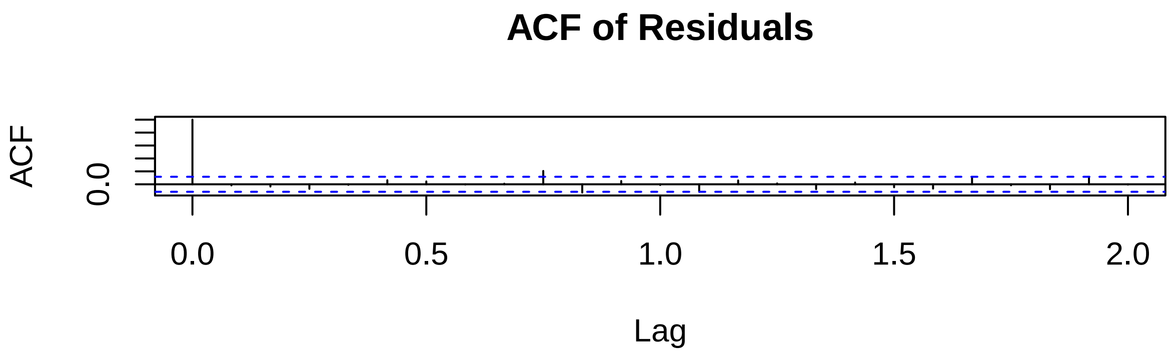
Note that 0 is not contained in the confidence intervals of any of the terms for our model, which is currently ARIMA(2, 1, 0)(2, 1, 2)[12]. This means that we reject the null hypothesis and conclude that the evidence supports the alternative hypothesis that our model coefficients are different from 0. Further, above we have deliberately attempted to overfit our data by providing additional parameters. In all cases the AIC increases, suggesting that these models do not do a better job of explaining our data simply. In general, one wants a model that minimizes the AIC.

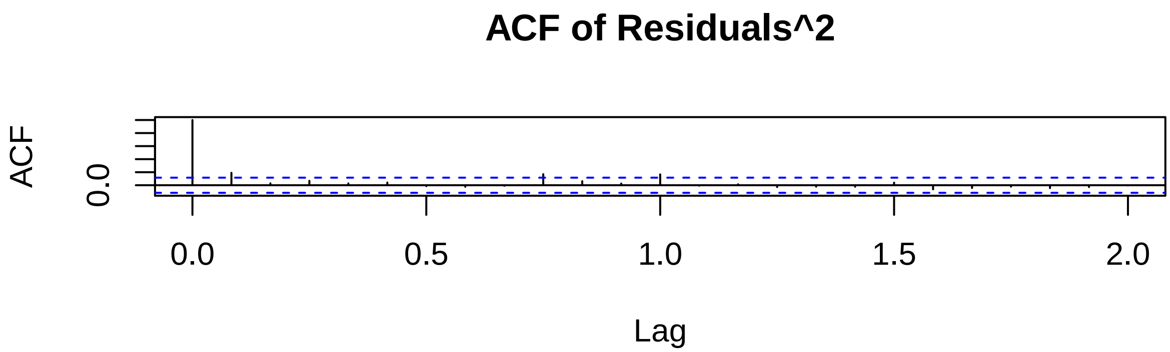
Therefore, we will continue with residual diagnostics for our chosen model

```
par(mfrow = c(2,2))
resids <- mod$residuals
plot.ts(resids, main = "Time Series Plot of Residuals")
hist(resids, main = "Time Series Plot of Residuals")
acf(resids, main = "ACF Plot of Residuals")
pacf(resids, main = "ACF Plot of Residuals")
```




```
par(mfrow = c(2, 1))
acf(resids, main = "ACF of Residuals")
acf(resids^2, main = "ACF of Residuals^2")
```






```
Box.test(resids, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resids
## X-squared = 0.087567, df = 1, p-value = 0.7673
```

Overall, the residuals appear to largely resemble white noise. The time series plot looks fairly like white noise, with no obvious patterns suggesting seasonality or a trend. There is one large spike, which we would need to know more about this data to properly try to account for. This was noted in the time series plot of the original data. The ACF shows one significant term at around 3/4 and the PACF shows two significant terms around the same area. However, there are no highly significant terms in the early part of the model (beyond the expected term of the ACF) and there is no repeating pattern of terms that are significant. Further, the residuals fail to reject the null hypothesis of the Ljung-Box test, meaning that the evidence suggests the observations are independent. There are some terms that are significant in the residual squared ACF, but none are highly significant and as there are only three, this does not represent a large enough number to suggest our residuals are behaving other than white noise.

We will note here that the residuals do not perfectly resemble white noise. There are still several lags showing statistical significant, which is not what we would want to see. However, we believe that we have fit the best possible model with the available information that we have. We would like to know more about the data and sampling methods to be able to fit the most appropriate possible model. We do however believe that we have satistified the conditions of stationarity and residuals behaving as white noise sufficiently to be able to forecast.

Check on Box test, is that what it says?

Appendix

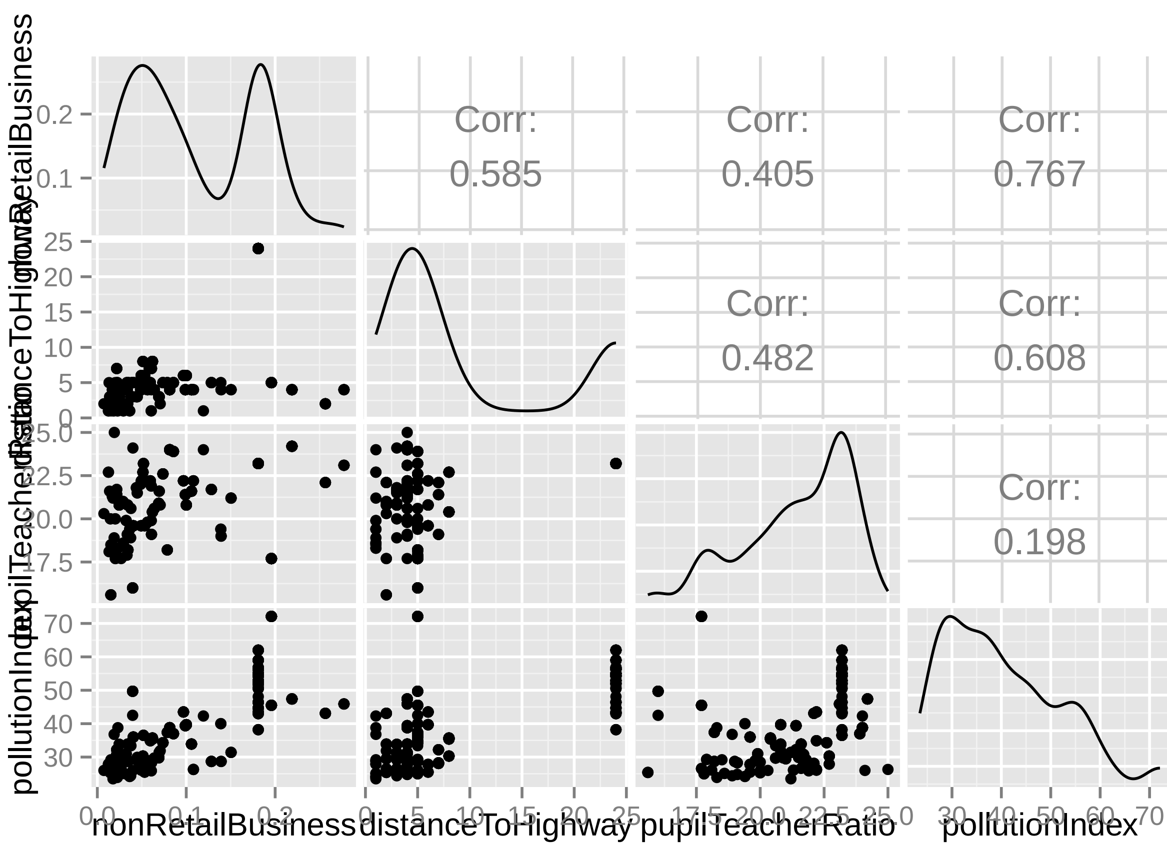
probably can be ultimately deleted.

I did the following to see if the issue was the variables or the records before I found the scope of the problem. Tossing all the records is too much, but I didnt want to delete this yet.

With the identification of variables that seem to strongly correlate, I want to do a couple of scatterplot matrices.

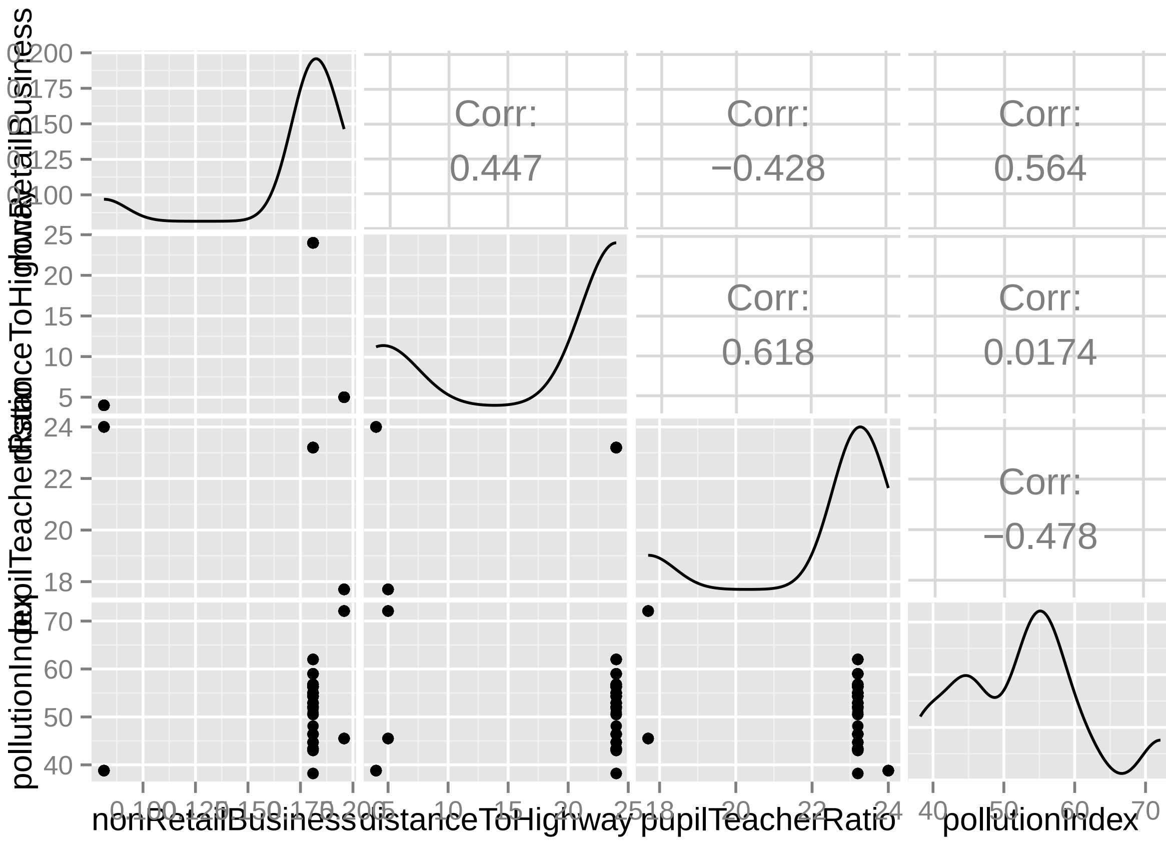
I will start with the first four identified.

```
data3 = data[,c("nonRetailBusiness","distanceToHighway","pupilTeacherRatio","pollutionIndex")]
ggpairs(data3)
```



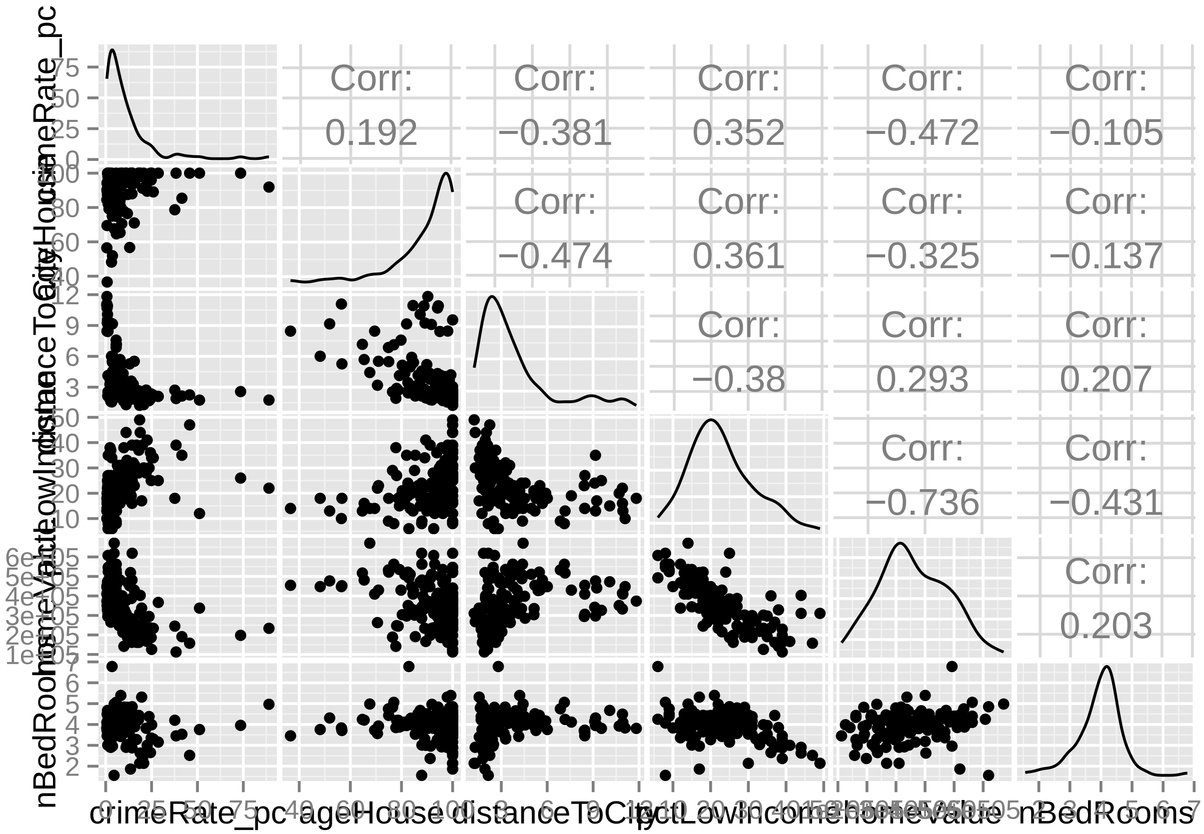
Surprisingly, there is not a super strong correlation between the variables (except between Non Retail Business and pollutionindex). Perhaps the problem is with those records then and not the variables themselves. Another subset will be created will just those 144 records first identified and anoother scatterplot matrix created.

```r
data2 = data[,c("nonRetailBusiness","distanceToHighway","pupilTeacherRatio","pollutionIndex")]
data3 = subset(data2, nonRetailBusiness==.181|nonRetailBusiness==.1958|nonRetailBusiness==.0814 )
ggpairs(data3)
```



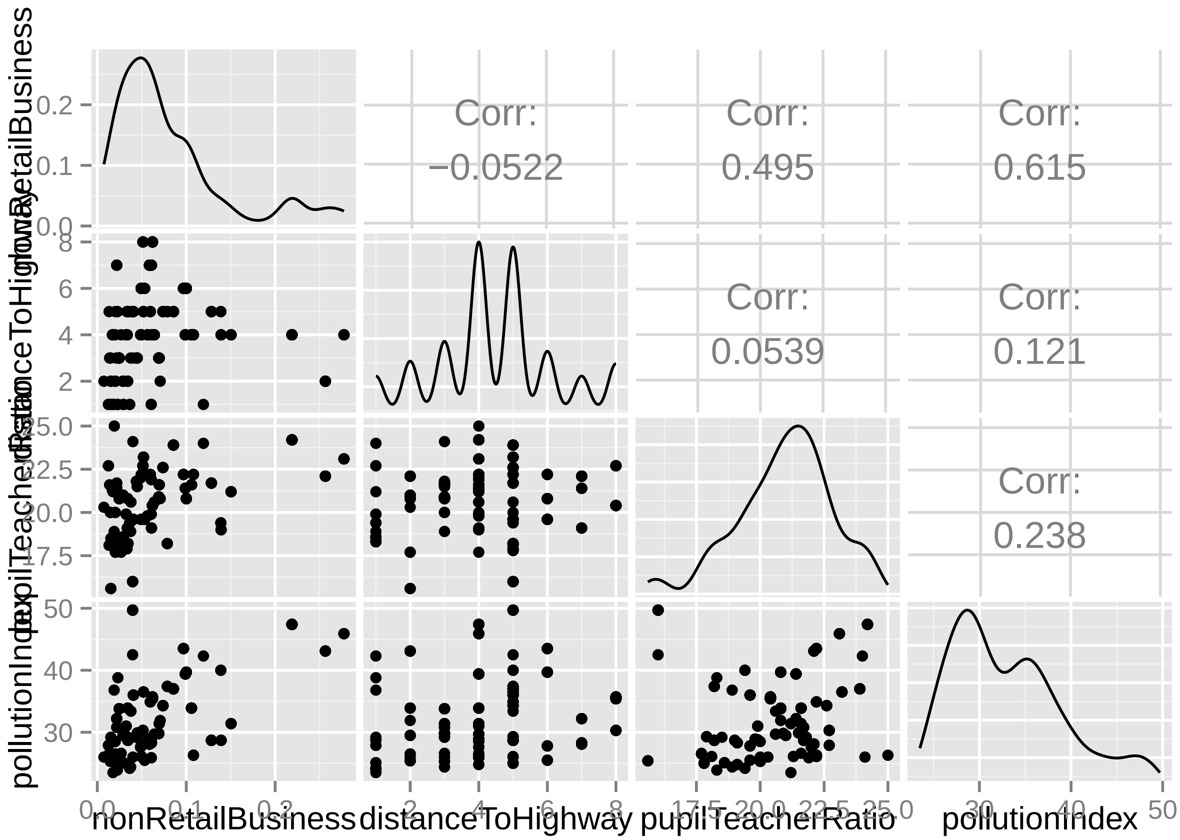
There is far too much colinearity with these records for comfort. I want to examine these 144 records in a scatterplot matrix with the other variables selected.

```r
data2 = subset(data, nonRetailBusiness==.181|nonRetailBusiness==.1958|nonRetailBusiness==.0814 )
data3 = data2[,c("crimeRate_pc","ageHouse","distanceToCity","pctLowIncome","homeValue","nBedRooms")]
ggpairs(data3)
```

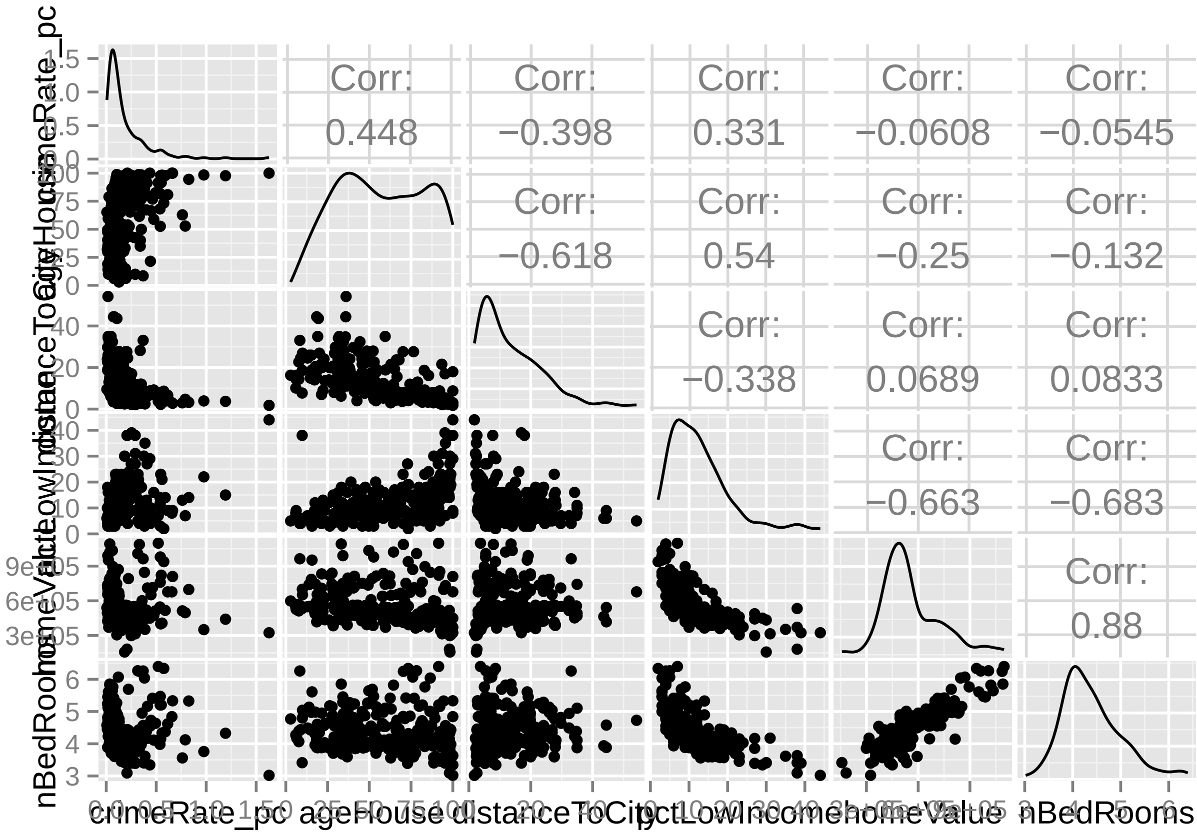


The matrix here causes me no concern. I am still unsure of whether or records or the variables are the problem here, so I will use the other 256 records and do the same two matrices.

```
data2 = data[,c("nonRetailBusiness","distanceToHighway","pupilTeacherRatio","pollutionIndex")]
data3 = subset(data2, nonRetailBusiness!=.181&nonRetailBusiness!=.1958&nonRetailBusiness!=.0814 )
ggpairs(data3)
```




```
data2 = subset(data, nonRetailBusiness!=.181&nonRetailBusiness!=.1958&nonRetailBusiness!=.0814 )
data3 = data2[,c("crimeRate_pc","ageHouse","distanceToCity","pctLowIncome","homeValue","nBedRooms")]
ggpairs(data3)
```



After all the examination of the variables and records, I have decided that the problem is with those 144 records. I will create a new subset of the remaining 256 and continue to use all variables.

```
data = subset(data, nonRetailBusiness!=.181&nonRetailBusiness!=.1958&nonRetailBusiness!=.0814 )
```
