PHASE 1 MSA – R ANALYSIS

Yatai Tian

Executive Summary

The dataset provided by NZMSA contains variables to predict the capital value of a home in New Zealand. We have added two columns to work with our investigation, specifically the depreciation index and population of the given SA1 area.

Our analysis includes 1051 observations, each observation with 17 variables. Our response variable is "CV" which is a positive numeric number with the rest of the variables to be regarded as explanatory. Further investigation showed some explanatory variables were removed from the analysis such as 'Address'. This is because every house has a different address, and this wouldn't be beneficial in predicting the cost of a house. Moreover, the SA1 unit area was also removed because it overlapped with the explanatory variable 'Suburbs'.

So what variables are significant in determining house CV then? We have evidence that the number of bathrooms, the NZ depreciation index, the number of 30-39-year-olds and the number of 50-59-year-olds living in an SA1 unit area all impact the CV of a house in New Zealand. We can further classify the relationship between CV and other explanatory variables to be the following; bathrooms is positively cubic, NZ depreciation score is negatively linear, the number of 30-39 years in an SA1 unit area is the first half of a positive quadratic graph, and the number of 50-59 years in an SA1 unit area is the first half of a negative quadratic graph.

After exploring different distributions from summary statistics, we have fitted a model that best fits our CV data set. This allows us to predict the capital value of a given house with its variance using certain variables.

Initial Data Exploration

We start by reading the data in and storing it in a variable called 'finalHouse.df'. The first few rows of the data are checked so we can see what we are working with and so far, everything seems appropriate. We then check the type of each variable to confirm it is correct. For example, the land area will not be a "chr" variable but a "num" variable. We will further subset our data and call it 'finalHouseRCHAR' which removes all the categorical data in our data set. This is done as categorical variables such as "Address" has too many levels to work with right now.

```
file <- read.csv(file = "final_house.csv", header = TRUE)
finalHouse.df <- subset(file, select = -c(X))
finalHouse.df[c(1,5,8,9,10,11,12,13,14,16)] <- lapply(finalHouse.df[c(1,5,8,9
```

```
,10,11,12,13,14,16)], as.numeric)
head(finalHouse.df)
```

```
##   Bedrooms Bathrooms                                    Address Land.area
## 1        5         3  106 Lawrence Crescent Hill Park, Auckland       714
## 2        5         3             8 Corsica Way Karaka, Auckland       564
## 3        6         4     243 Harbourside Drive Karaka, Auckland       626
## 4        2         1  2/30 Hardington Street Onehunga, Auckland        65
## 5        3         1     59 Israel Avenue Clover Park, Auckland       601
## 6        3         1 14 Tainui Terrace Mangere Bridge, Auckland       100
##        CV  Latitude Longitude     SA1 X0.19.years X20.29.years X30.39.year
s
## 1  960000 -37.01292  174.9041 7009770          48           27           2
4
## 2 1250000 -37.06367  174.9229 7009991          42           18           1
2
## 3 1250000 -37.06358  174.9240 7009991          42           18           1
2
## 4  740000 -36.91300  174.7874 7007871          42            6           2
1
## 5  630000 -36.97904  174.8926 7008902          93           27           3
3
## 6 1050000 -36.94393  174.7805 7007917          63           15           2
4
##   X40.49.years X50.59.years X60..years        Suburbs Population NZDep2018
## 1           21           24         21       Manurewa        174         6
## 2           21           15         30         Karaka        129         1
## 3           21           15         30         Karaka        129         1
## 4           21           12         15       Onehunga        120         2
## 5           30           21         33    Clover Park        231         9
## 6           33           30         39 Mangere Bridge        195         4
```

```
str(finalHouse.df)
```

```
## 'data.frame':    1048 obs. of  17 variables:
##  $ Bedrooms    : num  5 5 6 2 3 3 3 3 3 4 ...
##  $ Bathrooms   : num  3 3 4 1 1 1 1 2 2 2 ...
##  $ Address     : chr  "106 Lawrence Crescent Hill Park, Auckland" "8 Corsi
ca Way Karaka, Auckland" "243 Harbourside Drive Karaka, Auckland" "2/30 Hardi
ngton Street Onehunga, Auckland" ...
##  $ Land.area   : num  714 564 626 65 601 ...
##  $ CV          : num  960000 1250000 1250000 740000 630000 ...
##  $ Latitude    : num  -37 -37.1 -37.1 -36.9 -37 ...
##  $ Longitude   : num  175 175 175 175 175 ...
##  $ SA1         : num  7009770 7009991 7009991 7007871 7008902 ...
##  $ X0.19.years : num  48 42 42 42 93 63 33 36 45 30 ...
##  $ X20.29.years: num  27 18 18 6 27 15 12 33 27 27 ...
##  $ X30.39.years: num  24 12 12 21 33 24 18 39 15 36 ...
##  $ X40.49.years: num  21 21 21 21 30 33 12 21 12 15 ...
##  $ X50.59.years: num  24 15 15 12 21 30 15 12 12 24 ...
```

```
##  $ X60..years  : num  21 30 30 15 33 39 9 24 12 12 ...
##  $ Suburbs     : chr  "Manurewa" "Karaka" "Karaka" "Onehunga" ...
##  $ Population  : num  174 129 129 120 231 195 102 162 126 141 ...
##  $ NZDep2018   : num  6 1 1 2 9 4 4 4 10 6 ...

finalHouseRCHAR.df <- subset(finalHouse.df, select = -c(Address, Suburbs))
```

Let's split the data to create a training and testing set. We use set.seed for reproduction purposes.

```
smp_size <- floor(0.75 * nrow(finalHouseRCHAR.df))
set.seed(123)
train_ind <- sample(seq_len(nrow(finalHouseRCHAR.df)), size = smp_size)
train.df <- finalHouseRCHAR.df[train_ind, ]
test.df <- finalHouseRCHAR.df[-train_ind, ]
train2.df <- finalHouse.df[train_ind, ]
test2.df <- finalHouse.df[-train_ind, ]
```

Correlation and Relationships

Let's check our correlation coefficients of all our explanatory variables.

```
round(cor(train.df[, -4]), 2)

##                Bedrooms Bathrooms Land.area Latitude Longitude   SA1 X0.19.y
ears
## Bedrooms           1.00      0.70      0.10     0.00      0.05  0.05
0.02
## Bathrooms          0.70      1.00      0.07     0.07      0.09 -0.02       -
0.06
## Land.area          0.10      0.07      1.00     0.14     -0.06 -0.11       -
0.07
## Latitude           0.00      0.07      0.14     1.00     -0.36 -0.87       -
0.27
## Longitude          0.05      0.09     -0.06    -0.36      1.00  0.56
0.10
## SA1                0.05     -0.02     -0.11    -0.87      0.56  1.00
0.30
## X0.19.years        0.02     -0.06     -0.07    -0.27      0.10  0.30
1.00
## X20.29.years      -0.06     -0.06     -0.11    -0.15     -0.02  0.14
0.43
## X30.39.years      -0.08     -0.05     -0.14    -0.17      0.02  0.17
0.64
## X40.49.years       0.01      0.00     -0.02    -0.10      0.03  0.10
0.72
## X50.59.years       0.07      0.09      0.11    -0.04      0.13  0.09
0.47
## X60..years        -0.03      0.02      0.03     0.07      0.10  0.01
0.08
```

```
## Population      -0.03     -0.03     -0.06    -0.17      0.09  0.21
0.81
## NZDep2018       -0.22     -0.32     -0.11    -0.20     -0.01  0.19
0.14
##              X20.29.years X30.39.years X40.49.years X50.59.years X60..year
s
## Bedrooms             -0.06        -0.08         0.01         0.07      -0.0
3
## Bathrooms            -0.06        -0.05         0.00         0.09       0.0
2
## Land.area            -0.11        -0.14        -0.02         0.11       0.0
3
## Latitude             -0.15        -0.17        -0.10        -0.04       0.0
7
## Longitude            -0.02         0.02         0.03         0.13       0.1
0
## SA1                   0.14         0.17         0.10         0.09       0.0
1
## X0.19.years           0.43         0.64         0.72         0.47       0.0
8
## X20.29.years          1.00         0.73         0.29         0.15      -0.0
4
## X30.39.years          0.73         1.00         0.57         0.26       0.0
7
## X40.49.years          0.29         0.57         1.00         0.56       0.2
0
## X50.59.years          0.15         0.26         0.56         1.00       0.3
4
## X60..years           -0.04         0.07         0.20         0.34       1.0
0
## Population            0.66         0.81         0.76         0.59       0.4
3
## NZDep2018             0.24         0.11        -0.23        -0.28      -0.1
6
##              Population NZDep2018
## Bedrooms          -0.03     -0.22
## Bathrooms         -0.03     -0.32
## Land.area         -0.06     -0.11
## Latitude          -0.17     -0.20
## Longitude          0.09     -0.01
## SA1                0.21      0.19
## X0.19.years        0.81      0.14
## X20.29.years       0.66      0.24
## X30.39.years       0.81      0.11
## X40.49.years       0.76     -0.23
## X50.59.years       0.59     -0.28
## X60..years         0.43     -0.16
## Population         1.00      0.03
## NZDep2018          0.03      1.00
```

```r
#pairs(train.df.df, pch=19,col=rgb(0,0,1,.4)) better in Python
```

We can see all our high correlation coefficients relate to the ages of people in each of the SA1 unit areas. 20-29-year-olds and 30-39-year-olds are related at a correlation coefficient of 0.73 which could be expected as there is relatively the same amount of 20-29-year-olds compared to 30-39-year-olds, only dropping as we get older. Population and 30-39-year-olds have our highest correlation coefficient at 0.81 while comparing population and 40-49-year-olds are right behind at a correlation coefficient of 0.76. Since latitude and longitude depict what the suburb a house is situated in, I will drop longitude and latitude for suburb as it is more effective at determining CV. This is due to different areas of school zones have a heavier impact on affecting capital value than how South a house is.

Let us check multicollinearity now by calculating the variance inflation factors.

```r
round(diag(solve(cor(train.df[, -4]))), 2)
```

```
##     Bedrooms    Bathrooms     Land.area     Latitude    Longitude            
SA1 
##         2.06         2.14          1.07         4.76         1.68            6
.15 
##   X0.19.years X20.29.years X30.39.years X40.49.years X50.59.years   X60..ye
ars 
##        25.24        18.55         15.35         8.14         5.31          21
.14 
##   Population    NZDep2018 
##       196.49         1.70
```

We take values above five for some type of multicollinearity and values above ten to be considered as serious multicollinearity. We can see the population has very significant multicollinearity with some other variables, followed by 0-19 years. This is a good indication that we can use another variable instead of population and 0-19 years.

Analysis of Data

Let's fit this into R. Looking at suburbs we will check out the 1% significant suburbs.

```r
fullC.lm <- lm(CV ~. - Address - Latitude - Longitude - SA1, data = finalHous
e.df)
print(summary(fullC.lm))
```

```
## 
## Call:
## lm(formula = CV ~ . - Address - Latitude - Longitude - SA1, data = finalHo
use.df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2215382  -255209   -16434   182975 15293268 
##                                             Pr(>|t|)
```

```
## (Intercept)                                   0.534677
## Bedrooms                                      0.111430
## Bathrooms                                     1.82e-12 ***
## Land.area                                     1.04e-08 ***
## X0.19.years                                   0.031918 *
## X20.29.years                                  0.217997
## X30.39.years                                  0.512196
## X40.49.years                                  0.640265
## X50.59.years                                  0.017875 *
## X60..years                                    0.100295
## SuburbsEpsom                                  0.005444 **
## SuburbsHerne Bay                              0.001671 **
## SuburbsOkura                                  0.000141 ***
## SuburbsPohuehue                               2.23e-05 ***
## SuburbsRemuera                                0.003436 **
## SuburbsSaint Marys Bay                        0.003251 **
## SuburbsSt Heliers                             0.008677 **
## SuburbsWestmere                               0.009889 **
## Population                                    0.131587
## NZDep2018                                     0.021058 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 944400 on 848 degrees of freedom
## Multiple R-squared:  0.485,   Adjusted R-squared:  0.3642
## F-statistic: 4.014 on 199 and 848 DF,  p-value: < 2.2e-16

## Coefficients:
##                                                 Estimate Std. Error t value
## (Intercept)                                   -6.241e+05  1.005e+06  -0.621
## Bedrooms                                      -6.544e+04  4.107e+04  -1.593
## Bathrooms                                      3.513e+05  4.910e+04   7.154
## Land.area                                      1.609e+02  2.783e+01   5.782
## X0.19.years                                    1.524e+04  7.092e+03   2.149
## X20.29.years                                   8.728e+03  7.080e+03   1.233
## X30.39.years                                   4.832e+03  7.368e+03   0.656
## X40.49.years                                  -4.063e+03  8.691e+03  -0.467
## X50.59.years                                   1.820e+04  7.669e+03   2.373
## X60..years                                     1.107e+04  6.727e+03   1.645

## Population                                    -9.889e+03  6.552e+03  -1.509
## NZDep2018                                     -4.315e+04  1.867e+04  -2.311

(newdataEpsom <- nrow(finalHouse.df[which(finalHouse.df$Suburbs == "Epsom"),]
)) #23

(newdataHerneBay <- nrow(finalHouse.df[which(finalHouse.df$Suburbs == "Herne
Bay"),])) #5

 (newdataOkura <- nrow(finalHouse.df[which(finalHouse.df$Suburbs == "Okura"),
])) #1
```

```r
(newdataPohuehue <- nrow(finalHouse.df[which(finalHouse.df$Suburbs == "Pohueh
ue"),])) #1

(newdataRemuera <- nrow(finalHouse.df[which(finalHouse.df$Suburbs == "Remuera
"),])) #61

(newdataSaintMarysBay <- nrow(finalHouse.df[which(finalHouse.df$Suburbs == "S
aint Marys Bay"),])) #5

(newdataStHeliers <- nrow(finalHouse.df[which(finalHouse.df$Suburbs == "St He
liers"),])) #29

(newdataWestmere <- nrow(finalHouse.df[which(finalHouse.df$Suburbs == "Westme
re"),])) #1
```

We can see that Epsom, Herne Bay, Okura, Pohuehue, Remuera, Saint Marys Bay, St Heliers and Westmere fall in the 1% significance range. Checking the number of entry points we have in each significant suburb, we see that there is only one entry for Okura, Pohuehue and Westmere. Furthermore, there are only 5 entries for Herne Bay and Saint Marys Bay. There is certainly not enough data for these suburbs to imply that all houses in that subrub has an increased or decrease price range from the average suburb.

The 5% significant variables in this regression includes the number of bathrooms, land area, number of 0-19-year-olds and 50-59-year-olds in the suburb, suburb area and NZ depreciation score. Interestingly, the number of bedrooms was not significant. However, bedrooms and bathrooms were correlated at a value of 0.71 so there is some evidence that each impacts the other. It is also interesting to note that the number of bedrooms had a negative value, indicating the more bedrooms the cheaper the house. This must be further investigated in.

What happens if we remove the variable suburb? I fitted another linear model and found out this time land area is now insignificant. To make this model better I will decrease the variable with the greatest P-value and run the summary again, repeating until all my variables are 5% significant. This process included the removal of bedrooms, population, >60 years, 0-19 years, 40-49 years and then finally, land area.

```r
full.lm <- lm(CV ~. - SA1 - Latitude - Longitude, data = train.df)
print(summary(full.lm))

##
## Call:
## lm(formula = CV ~ . - SA1 - Latitude - Longitude, data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1793947  -476540  -113692   222241  9449492
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1330342.04  189660.02   7.014 5.04e-12 ***
```

```
## Bedrooms         2664.65   43426.98   0.061   0.9511
## Bathrooms      345216.64   50044.54   6.898 1.10e-11 ***
## Land.area          31.32      19.46   1.610   0.1078
## X0.19.years      6754.10    6567.06   1.028   0.3040
## X20.29.years     7658.99    6599.37   1.161   0.2462
## X30.39.years   -13849.56    7167.27  -1.932   0.0537 .
## X40.49.years    -5117.92    8436.11  -0.607   0.5442
## X50.59.years    13093.71    7217.54   1.814   0.0700 .
## X60..years       4595.26    6374.38   0.721   0.4712
## Population      -3101.24    6307.25  -0.492   0.6231
## NZDep2018     -118705.32   14716.42  -8.066 2.76e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 930400 on 774 degrees of freedom
## Multiple R-squared:  0.3184, Adjusted R-squared:  0.3087
## F-statistic: 32.87 on 11 and 774 DF,  p-value: < 2.2e-16
```

```
full.lm1 <- lm(CV ~. - SA1 - Latitude - Longitude - Bedrooms, data = train.df
)
#print(summary(full.lm1))
full.lm2 <- lm(CV ~. - SA1 - Latitude - Longitude - Bedrooms - Population, da
ta = train.df)
#print(summary(full.lm2))
full.lm3 <- lm(CV ~. - SA1 - Latitude - Longitude - Bedrooms - Population - X
60..years, data = train.df)
#print(summary(full.lm3))
full.lm4 <- lm(CV ~. - SA1 - Latitude - Longitude - Bedrooms - Population - X
60..years - X0.19.years, data = train.df)
#print(summary(full.lm4))
full.lm5 <- lm(CV ~. - SA1 - Latitude - Longitude - Bedrooms - Population - X
60..years - X0.19.years - X40.49.years, data = train.df)
#print(summary(full.lm5))
full.lm6 <- lm(CV ~. - SA1 - Latitude - Longitude - Bedrooms - Population - X
60..years - X0.19.years - X40.49.years - Land.area , data = train.df)
print(summary(full.lm6))
## Call:
## lm(formula = CV ~ . - SA1 - Latitude - Longitude - Bedrooms -
##     Population - X60..years - X0.19.years - X40.49.years - Land.area,
##     data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1791443  -478251  -112562   224286  9578876
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1279323     148732   8.602  < 2e-16 ***
## Bathrooms       353593      36345   9.729  < 2e-16 ***
## X20.29.years      4691       2343   2.002 0.045597 *
```

```
## X30.39.years   -17332       2785  -6.222 7.98e-10 ***
## X50.59.years    12076       3440   3.511 0.000473 ***
## NZDep2018     -108720      13042  -8.336 3.45e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 930900 on 780 degrees of freedom
## Multiple R-squared:  0.3124, Adjusted R-squared:  0.308
## F-statistic: 70.87 on 5 and 780 DF,  p-value: < 2.2e-16
```

After all this, we have manually calculated our best model. This includes the explanatory variables bathrooms, number of 20-29, 30-39, 50-59-year-old people in each SA1 unit area, and NZ depreciation score. There can be lots of inaccuracy that has occurred while fitting this model and we can do better by letting the computer decide the best model for us. We want to find what variables to include to not underfit or overfit our model. To do this, we use the help of "MuMIn" package in R. This allows us to choose a model resulting in the smallest mean square prediction error. Here, I will not be adding our variable suburb because running dredge with these many variables takes an exponential amount of time. However, I will add it afterwards to our top models as it proved significant in our earlier tests.

```
library("MuMIn")
finalHouse.lmodel <- lm(CV ~. - SA1 - Latitude - Longitude, data = train.df,
na.action = "na.fail")
#summary(finalHouse.lmodel)
finalHouse.ldredge <- dredge(finalHouse.lmodel)

## Fixed term is "(Intercept)"

print(round(finalHouse.ldredge[1:10, ]),2)

##      (Intercept) Bathrooms Bedrooms Land.area NZDep2018 Population X0.19.y
ears
## 718      1252584    351913       NA        31   -107807         NA
NA
## 926      1288046    349559       NA        31   -113406       2499
NA
## 714      1279323    353593       NA        NA   -108720         NA
NA
## 922      1316824    351134       NA        NA   -114369       2494
NA
## 1006     1353790    346075       NA        31   -119280         NA
3499
## 1950     1320811    347053       NA        31   -118746       3778
NA
## 1438     1376560    347471       NA        33   -125535       5123
NA
## 990      1287587    348741       NA        31   -114550       1938
NA
## 414      1344620    351034       NA        34   -119781       3679
```

```
NA
## 974      1298854    349148       NA        31   -110134          NA
NA
##      X20.29.years X30.39.years X40.49.years X50.59.years X60..years df log
Lik
## 718          4646       -16802           NA        11395         NA  8 -11
914
## 926            NA       -16871       -10111         8141         NA  9 -11
913
## 714          4691       -17332           NA        12076         NA  7 -11
915
## 922            NA       -17298       -10274         8907         NA  8 -11
914
## 1006         4388       -16679        -8075        11472         NA 10 -11
912
## 1950           NA       -19795       -12515         6941      -2231 10 -11
912
## 1438           NA       -22862       -12811           NA      -2918  9 -11
913
## 990          2745       -18230        -8105         8964         NA 10 -11
912
## 414            NA       -19555        -9560           NA         NA  8 -11
914
## 974          4272       -15348        -3840        12983         NA  9 -11
913
##       AICc delta weight
## 718  23844     0      0
## 926  23844     0      0
## 714  23844     1      0
## 922  23844     1      0
## 1006 23845     1      0
## 1950 23845     1      0
## 1438 23845     1      0
## 990  23845     1      0
## 414  23845     1      0
## 974  23845     1      0
```

```r
finalHouse.ldredgeBIC <- dredge(finalHouse.lmodel, rank = "BIC")
```

```
## Fixed term is "(Intercept)"
```

```r
print(round(finalHouse.ldredgeBIC[1:10, ]),2)
```

```
##     (Intercept) Bathrooms Bedrooms Land.area NZDep2018 Population X0.19.y
ears
## 650     1268045    356758       NA        NA   -102644          NA
NA
## 154     1294955    359385       NA        NA   -112039        2571
NA
## 714     1279323    353593       NA        NA   -108720          NA
NA
```

```
## 654      1241016    355022       NA        31   -101777         NA
NA
## 410      1382865    352955       NA        NA   -121534       3797
NA
## 158      1257264    357349       NA        35   -110383       2467
NA
## 906      1334396    352452       NA        NA   -106611         NA
NA
## 666      1228502    358053       NA        NA   -104832       1289
NA
## 138      1555843    356872       NA        NA   -116327         NA
NA
## 1674     1249757    357577       NA        NA   -102037         NA
NA
##       X20.29.years X30.39.years X40.49.years X50.59.years X60..years df log
Lik
## 650             NA       -13331           NA        12139         NA  6 -11
917
## 154             NA       -19799           NA           NA         NA  6 -11
918
## 714           4691       -17332           NA        12076         NA  7 -11
915
## 654             NA       -12832           NA        11446         NA  7 -11
916
## 410             NA       -20318        -9685           NA         NA  7 -11
916
## 158             NA       -19031           NA           NA         NA  7 -11
916
## 906             NA       -11714        -5440        14370         NA  7 -11
916
## 666             NA       -16959           NA         8291         NA  7 -11
916
## 138             NA       -11236           NA           NA         NA  5 -11
923
## 1674            NA       -13315           NA        11439        961  7 -11
917
##        BIC delta weight
## 650  23874     0      0
## 154  23876     2      0
## 714  23877     3      0
## 654  23878     4      0
## 410  23879     4      0
## 158  23879     5      0
## 906  23879     5      0
## 666  23879     5      0
## 138  23880     6      0
## 1674 23880     6      0
```

Our first 10 rows of our AIC model have a difference of AICc value smaller or equal to 2 so we cannot fully determine that one model is better than the other. Diving a bit more into these models we see that bathrooms, NZDep2018, and the number of 30-39-year-olds are all included in our first 10 models. Using BIC as subsetting we find that the top 10 rows all include bathrooms, NZDep2018, and the number of 30-39-year-olds in the suburb. These variables are the ones exactly in the AICc model too. I like picking a model that covers the top 10 in both AICc and BIC. This is model 714. Let's extract that model out and add suburbs to see if anything changes.

```
finalHouse.lmodel2 <- lm(CV ~ Bathrooms + NZDep2018 + X20.29.years + X30.39.y
ears + X50.59.years + Suburbs, data = train2.df, na.action = "na.fail")
#summary(finalHouse.lmodel2)
finalHouse.ldredge2 <- dredge(finalHouse.lmodel2)

## Fixed term is "(Intercept)"

finalHouse.ldredge2BIC <- dredge(finalHouse.lmodel2, rank = "BIC")

## Fixed term is "(Intercept)"

print(finalHouse.ldredge2[1:10, ])

## Global model call: lm(formula = CV ~ Bathrooms + NZDep2018 + X20.29.years
+ X30.39.years +
##     X50.59.years + Suburbs, data = train2.df, na.action = "na.fail")
## ---
## Model selection table
##      (Int)    Bth     NZD X20.29.yrs X30.39.yrs X50.59.yrs df    logLik
AICc
## 60 1279000 353600 -108700     4691.0     -17330      12080  7 -11915.02 23
844.2
## 52 1268000 356800 -102600                -13330      12140  6 -11917.04 23
846.2
## 28 1566000 353700 -122400     4765.0     -15310             6 -11921.18 23
854.5
## 20 1556000 356900 -116300                -11240             5 -11923.23 23
856.5
## 44 1176000 363200 -105100    -5766.0                  7654  6 -11934.06 23
880.2
## 12 1373000 362500 -114500    -4910.0                        5 -11936.54 23
883.1
## 36 1144000 360800 -118400                             4911  5 -11940.01 23
890.1
## 4  1281000 360600 -123400                                   4 -11941.07 23
890.2
## 50  446300 445900                        -16170      20650  5 -11948.56 23
907.2
## 58  445200 446000             146.7      -16300      20660  6 -11948.56 23
909.2
##    delta weight
## 60  0.00  0.726
## 52  1.99  0.268
## 28 10.29  0.004
```

```
## 20 12.35  0.002
## 44 36.04  0.000
## 12 38.96  0.000
## 36 45.90  0.000
## 4  46.00  0.000
## 50 63.02  0.000
## 58 65.04  0.000
## Models ranked by AICc(x)
```

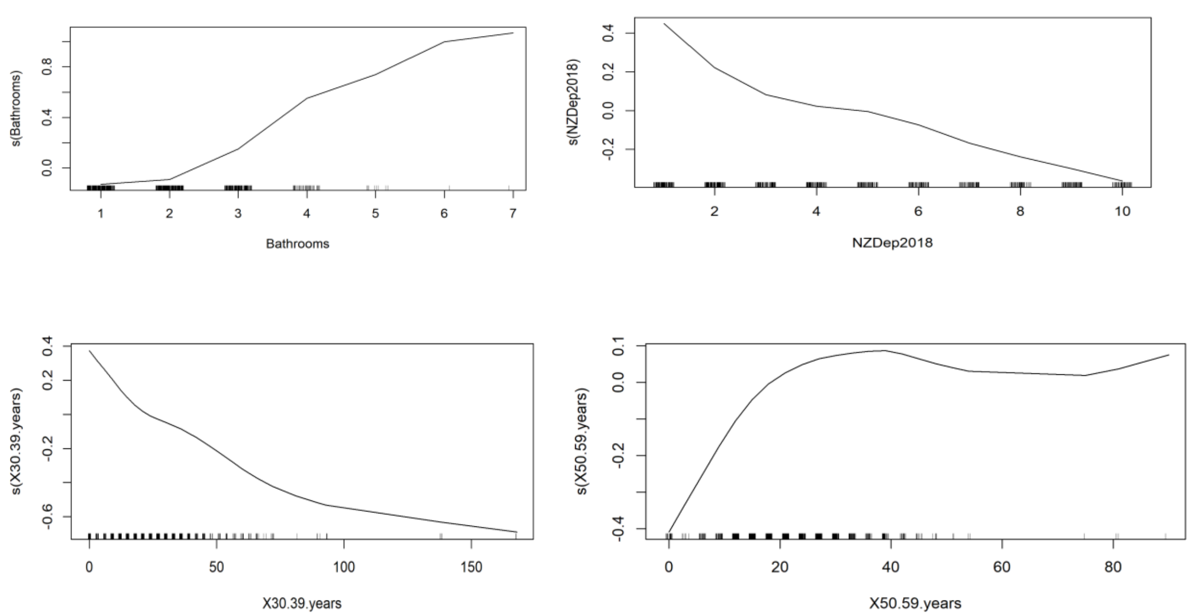
```r
print(finalHouse.ldredge2BIC[1:10, ])
```

```
## Global model call: lm(formula = CV ~ Bathrooms + NZDep2018 + X20.29.years
+ X30.39.years +
##     X50.59.years + Suburbs, data = train2.df, na.action = "na.fail")
## ---
## Model selection table
##      (Int)    Bth     NZD X20.29.yrs X30.39.yrs X50.59.yrs df    logLik
BIC
## 52 1268000 356800 -102600                -13330      12140  6 -11917.04 23
874.1
## 60 1279000 353600 -108700     4691.0     -17330      12080  7 -11915.02 23
876.7
## 20 1556000 356900 -116300                -11240             5 -11923.23 23
879.8
## 28 1566000 353700 -122400     4765.0     -15310             6 -11921.18 23
882.4
## 12 1373000 362500 -114500    -4910.0                        5 -11936.54 23
906.4
## 44 1176000 363200 -105100    -5766.0                 7654  6 -11934.06 23
908.1
## 4  1281000 360600 -123400                                   4 -11941.07 23
908.8
## 36 1144000 360800 -118400                            4911  5 -11940.01 23
913.3
## 50  446300 445900                        -16170      20650  5 -11948.56 23
930.5
## 58  445200 446000             146.7     -16300      20660  6 -11948.56 23
937.1
##    delta weight
## 52  0.00  0.746
## 60  2.64  0.200
## 20  5.72  0.043
## 28  8.29  0.012
## 12 32.33  0.000
## 44 34.05  0.000
## 4  34.73  0.000
## 36 39.27  0.000
## 50 56.39  0.000
## 58 63.05  0.000
## Models ranked by BIC(x)
```

Very interestingly, our top 10 models don't include suburbs as a significant variable anymore. Furthermore, all top 10 models in AICc overlap with BIC. As a statistician, I will pick one of the models with low AICc and low BIC with fewer variables. In this case, it is model 52. Comparing model 52 with my previous model that was done manually, we can easily see we have only removed the number of 20-29-year-olds in the SA1 unit area. In fact, model 60 was our manually completed model. Let's extract model 52 out and fit a GAM using all regressors to explore ways of improving our model. Using the VGAM package we have…

Building a Model

```
library("VGAM")

## Loading required package: stats4

## Loading required package: splines

##
## Attaching package: 'VGAM'

## The following object is masked from 'package:MuMIn':
##
##     AICc

bestModel.lm <- get.models(finalHouse.ldredge2BIC, 1)[[1]]
bestModel.gam <- vgam(CV~s(Bathrooms) + s(NZDep2018) + s(X30.39.years) + s(X5
0.59.years), family = poissonff, data = train.df)
plot(bestModel.gam)
```



Looking at the plots, bathrooms seems to be modelled as positive cubic plot, NZDep2018 seems like a negative linear plot, X30.39.years is somewhat of the left half of a positive quadratic curve and X50.59.years could be modelled as a positive cubic too. Let's see if adding this would change anything in our new dredge model.

```
finalHouse.lmodel3 <- lm(CV ~ Bathrooms + I(Bathrooms^2) + I(Bathrooms^3) + N
ZDep2018 + X30.39.years + I(X30.39.years^2) + X50.59.years + I(X50.59.years^2
) + I(X50.59.years^3), data = train.df, na.action = "na.fail")
finalHouse.ldredge3 <- dredge(finalHouse.lmodel3)

## Fixed term is "(Intercept)"

print(finalHouse.ldredge3[1:10, ])

## Global model call: lm(formula = CV ~ Bathrooms + I(Bathrooms^2) + I(Bathro
oms^3) +
##     NZDep2018 + X30.39.years + I(X30.39.years^2) + X50.59.years +
##     I(X50.59.years^2) + I(X50.59.years^3), data = train.df, na.action = "n
a.fail")
## ---
## Model selection table
##       (Int)      Bth  Bth^2  Bth^3     NZD X30.39.yrs X30.39.yrs^2 X50.59.
yrs
## 256 2531000 -1205000 474800 -37150 -110300     -23140       113.10      33
750
## 512 2369000 -1191000 470800 -36780 -110500     -23410       111.30      55
550
## 384 2628000 -1200000 472500 -36930 -110400     -22340       104.70      20
660
## 128 2685000 -1150000 454800 -35220 -111200     -19610        66.69      12
690
## 508 1753000  -318200 135100        -108800     -23250       106.90      54
990
## 252 1916000  -323500 135600        -108600     -22970       108.70      32
280
## 480 2259000 -1129000 452600 -35150 -115200     -12590                   46
910
## 224 2432000 -1143000 456500 -35530 -115100     -12120                   23
410
## 96  2552000 -1125000 448800 -34740 -114500     -12730                   13
020
## 448 2809000 -1183000 465900 -36310 -112700     -20830        91.89
##     X50.59.yrs^2 X50.59.yrs^3 df    logLik    AICc delta weight
## 256       -388.8              10 -11893.49 23807.3  0.00  0.387
## 512      -1148.0        6.804 11 -11892.50 23807.3  0.09  0.371
## 384                    -2.800 10 -11894.60 23809.5  2.23  0.127
## 128                            9 -11897.09 23812.4  5.17  0.029
## 508      -1158.0        7.083 10 -11896.40 23813.1  5.83  0.021
## 252       -366.4               9 -11897.46 23813.1  5.89  0.020
## 480      -1006.0        7.282 10 -11896.87 23814.0  6.77  0.013
```

```
## 224       -189.7              9 -11897.99 23814.2  6.95  0.012
## 96                            8 -11899.07 23814.3  7.06  0.011
## 448        541.4       -6.541 10 -11897.40 23815.1  7.82  0.008
## Models ranked by AICc(x)
```

```
first12 <- 12; out = rep(0, first12)
for (i in 1:first12) {
  preds = predict(get.models(finalHouse.ldredge3, i)[[1]], newdata = test.df,
type = "response")
  out[i] = mean((preds - test.df$CV)^2)
}
round(out, 2)
```

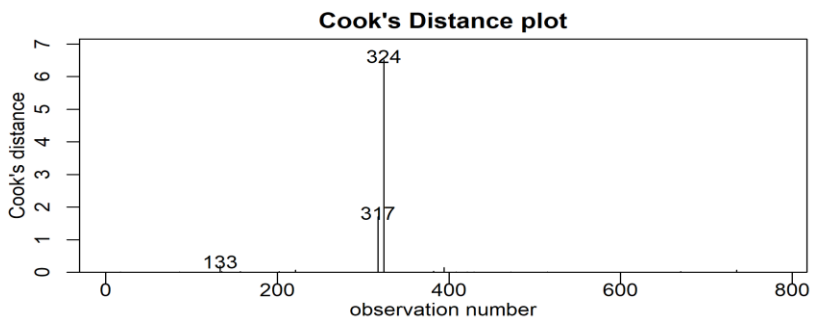
```
##  [1] 1.632585e+12 1.630458e+12 1.633332e+12 1.632165e+12 1.730677e+12
##  [6] 1.734161e+12 1.649148e+12 1.652180e+12 1.649482e+12 1.632417e+12
## [11] 1.652086e+12 1.735272e+12
```

Here, we can see that adding on these polynomials improves our AICc score and increases log-likelihood from our first dredge function. Calculating the MSPE for ou first 12 models on our test data, we have the full model (model 2) giving the lowest MSPE but all models result in a small certain range. This means all models have a similar error rate. Let's use the top (first) model and check some assumptions we have used.

```
bestModelPoly.lm <- get.models(finalHouse.ldredge3, 1)[[1]]
summary(bestModelPoly.lm)
```

```
##
## Call:
## lm(formula = CV ~ Bathrooms + I(Bathrooms^2) + I(Bathrooms^3) +
##     NZDep2018 + X30.39.years + I(X30.39.years^2) + X50.59.years +
##     I(X50.59.years^2) + 1, data = train.df, na.action = "na.fail")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2250236  -468093   -90143   254923  9164870
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2531441.3   306885.0   8.249 6.82e-16 ***
## Bathrooms          -1204784.3   337906.8  -3.565 0.000385 ***
## I(Bathrooms^2)       474776.9   123103.0   3.857 0.000124 ***
## I(Bathrooms^3)       -37154.5    13224.7  -2.809 0.005087 **
## NZDep2018           -110261.9    12631.5  -8.729  < 2e-16 ***
## X30.39.years         -23138.6     4162.1  -5.559 3.72e-08 ***
## I(X30.39.years^2)       113.1       37.8   2.992 0.002864 **
## X50.59.years          33754.4     8553.3   3.946 8.65e-05 ***
## I(X50.59.years^2)      -388.8      145.2  -2.678 0.007571 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 907500 on 777 degrees of freedom
```

```
## Multiple R-squared:  0.3491, Adjusted R-squared:  0.3424
## F-statistic: 52.08 on 8 and 777 DF,  p-value: < 2.2e-16

cooks20x(bestModelPoly.lm)
```

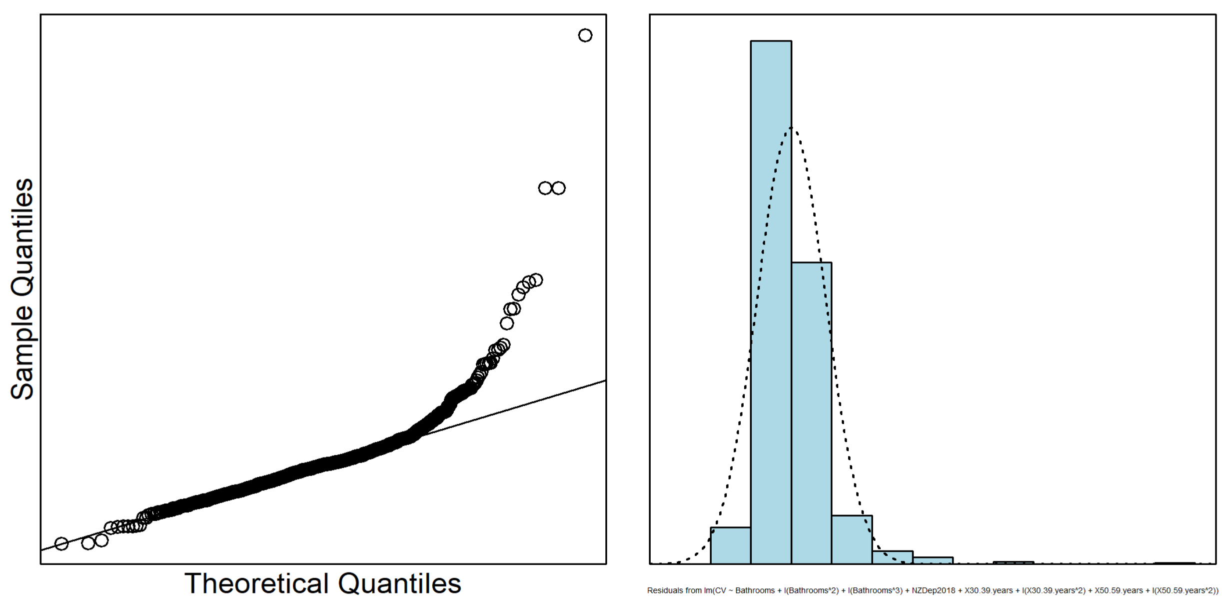


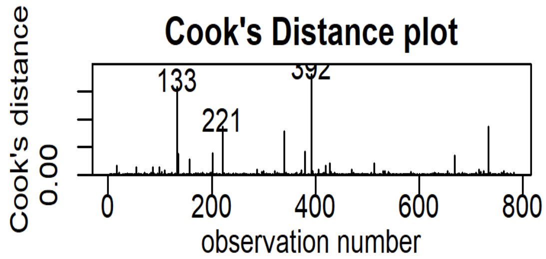

```
train.df[c(317,324),]

##      Bedrooms Bathrooms Land.area       CV  Latitude Longitude     SA1
## 1037        5         6      3565 12500000 -36.87555  174.7954 7005840
## 893         6         7       507  1875000 -36.87632  174.7612 7005400
##      X0.19.years X20.29.years X30.39.years X40.49.years X50.59.years X60..
years
## 1037          42            9            6           27           30
36
## 893           33           75           42           24           33
36
##      Population NZDep2018
## 1037        150         1
## 893         237         4
```

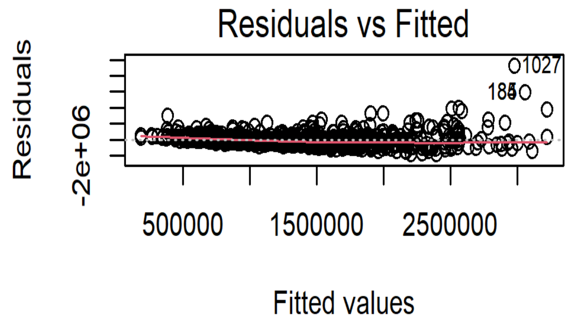
From the summary, all our variables are significant at the 5% level. This is good so we keep all of them. The model, however, only explains 35% of the variability of house prices. We should remove outliers first, so we take Cook's distance greater than 0.4 and remove them from our data set. This is row 317 and 324.

```
bestModelPoly.lm2 <- lm(CV ~ Bathrooms + I(Bathrooms^2) + I(Bathrooms^3) + NZ
Dep2018 + X30.39.years + I(X30.39.years^2) + X50.59.years + I(X50.59.years^2)
, data = train.df[-c(317, 324), ])
summary(bestModelPoly.lm2)

##
## Call:
## lm(formula = CV ~ Bathrooms + I(Bathrooms^2) + I(Bathrooms^3) +
##     NZDep2018 + X30.39.years + I(X30.39.years^2) + X50.59.years +
##     I(X50.59.years^2), data = train.df[-c(317, 324), ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1821620  -464465   -92460   254022  9274906
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.008e+06  3.769e+05   7.981 5.23e-15 ***
## Bathrooms          -2.015e+06  4.874e+05  -4.134 3.95e-05 ***
## I(Bathrooms^2)      8.733e+05  2.032e+05   4.298 1.94e-05 ***
## I(Bathrooms^3)     -9.465e+04  2.541e+04  -3.726 0.000209 ***
## NZDep2018          -1.069e+05  1.192e+04  -8.970  < 2e-16 ***
## X30.39.years       -2.141e+04  3.942e+03  -5.432 7.45e-08 ***
## I(X30.39.years^2)   1.006e+02  3.577e+01   2.813 0.005029 **
## X50.59.years        2.965e+04  8.086e+03   3.666 0.000263 ***
## I(X50.59.years^2) -3.372e+02  1.371e+02  -2.460 0.014107 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 856200 on 775 degrees of freedom
## Multiple R-squared:  0.339,  Adjusted R-squared:  0.3322
## F-statistic: 49.69 on 8 and 775 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
cooks20x(bestModelPoly.lm2)
plot(bestModelPoly.lm2, which=1)
normcheck(bestModelPoly.lm2)
```







Normality plot for CV is highly left-skewed. This means our model of which we assumed a normal distribution is not valid so we must tend to a different distribution. Let's try fitting a Poisson model as we are dealing with house prices which is a discrete value and CV is greater than 0.

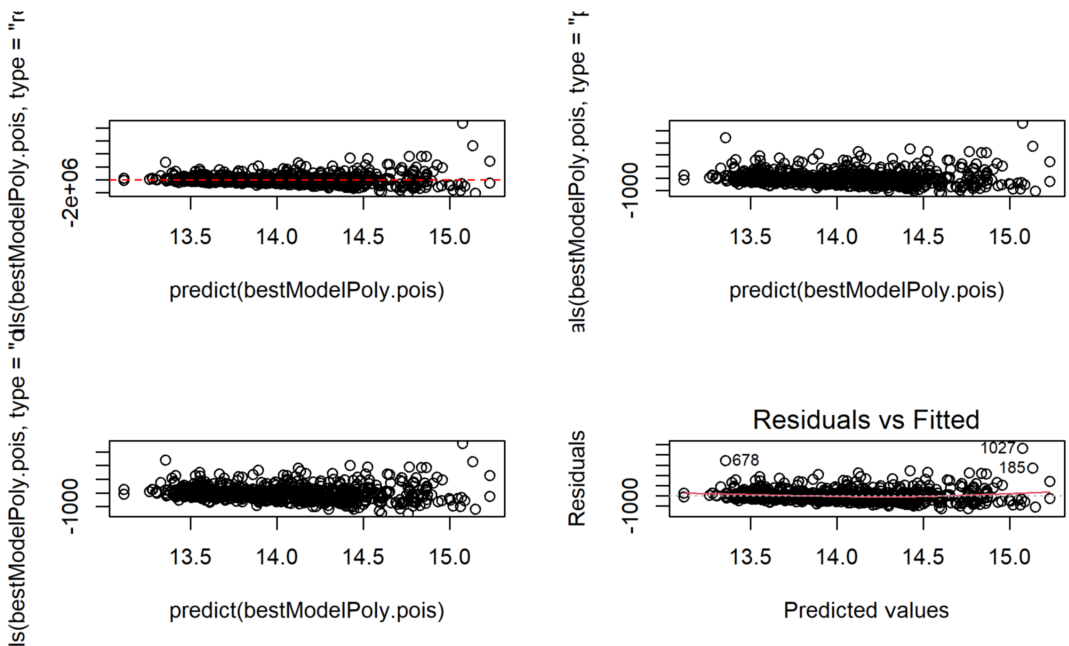
```
bestModelPoly.pois <- glm(CV ~ Bathrooms + I(Bathrooms^2) + I(Bathrooms^3) +
NZDep2018 + X30.39.years + I(X30.39.years^2) + X50.59.years + I(X50.59.years^
2), family = poisson, data = train.df[-c(317, 324), ])
summary(bestModelPoly.pois)

##
## Call:
## glm(formula = CV ~ Bathrooms + I(Bathrooms^2) + I(Bathrooms^3) +
##     NZDep2018 + X30.39.years + I(X30.39.years^2) + X50.59.years +
##     I(X50.59.years^2), family = poisson, data = train.df[-c(317,
##     324), ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1511.8   -360.0   -117.9    204.9   3614.9
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.508e+01  3.822e-04   39448   <2e-16 ***
## Bathrooms         -1.140e+00  4.673e-04   -2439   <2e-16 ***
## I(Bathrooms^2)     5.102e-01  1.873e-04    2724   <2e-16 ***
## I(Bathrooms^3)    -5.691e-02  2.273e-05   -2504   <2e-16 ***
## NZDep2018         -8.309e-02  1.273e-05   -6525   <2e-16 ***
## X30.39.years      -1.388e-02  4.099e-06   -3385   <2e-16 ***
## I(X30.39.years^2)  5.714e-05  4.013e-08    1424   <2e-16 ***
## X50.59.years       1.793e-02  8.424e-06    2129   <2e-16 ***
## I(X50.59.years^2) -2.014e-04  1.399e-07   -1440   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
```

```
##
##     Null deviance: 435975539  on 783  degrees of freedom
## Residual deviance: 234930637  on 775  degrees of freedom
## AIC: 234943043
##
## Number of Fisher Scoring iterations: 4
```

```
1 - pchisq(bestModelPoly.pois$deviance, bestModelPoly.pois$df.residual)
```

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

```
par(mfrow=c(2,2))
plot(predict(bestModelPoly.pois),residuals(bestModelPoly.pois, type="response
"))
abline(h=0,lty="dashed", col="red")
## Pearson residuals
plot(predict(bestModelPoly.pois),residuals(bestModelPoly.pois,type="pearson")
)
## Deviance residuals
plot(predict(bestModelPoly.pois),residuals(bestModelPoly.pois,type="deviance"
))
plot(bestModelPoly.pois,which=1)
```

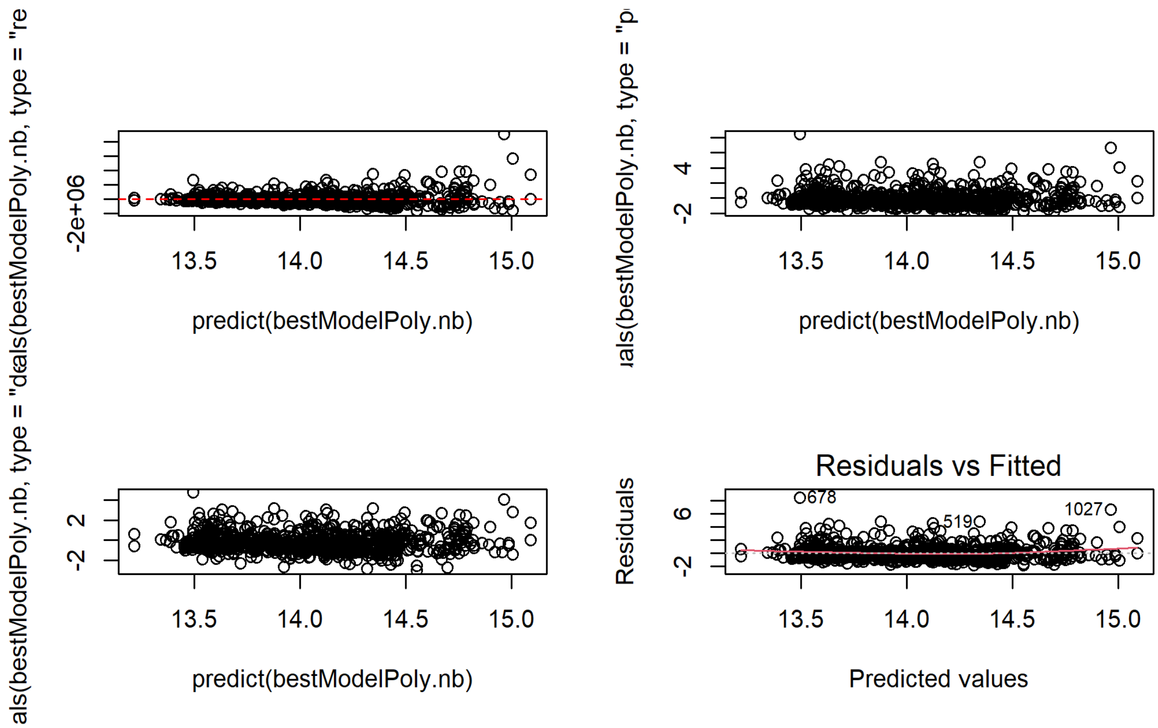


The Pearson and deviance residuals show that the variance in our data is not correctly captured by the model. The deviance highly suggests a lack-of-fit from our pchisq test (p=0). Instead of going to quasi-Poisson, let us jump straight to a negative binomial model. I have done this because we are not capturing our variance correctly.

```
library("MASS")
bestModelPoly.nb <- glm.nb(CV ~ Bathrooms + I(Bathrooms^2) + I(Bathrooms^3) +
NZDep2018 + X30.39.years + I(X30.39.years^2) + X50.59.years + I(X50.59.years^
```

```r
2), data = train.df[-c(317, 324), ])
summary(bestModelPoly.nb)

##
## Call:
## glm.nb(formula = CV ~ Bathrooms + I(Bathrooms^2) + I(Bathrooms^3) +
##     NZDep2018 + X30.39.years + I(X30.39.years^2) + X50.59.years +
##     I(X50.59.years^2), data = train.df[-c(317, 324), ], init.theta = 5.346
355275,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0040  -0.7884  -0.3060   0.3491   4.7589
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.477e+01  1.904e-01  77.589  < 2e-16 ***
## Bathrooms         -7.395e-01  2.462e-01  -3.004 0.002666 **
## I(Bathrooms^2)     3.558e-01  1.026e-01   3.467 0.000526 ***
## I(Bathrooms^3)    -3.993e-02  1.283e-02  -3.112 0.001860 **
## NZDep2018         -7.851e-02  6.023e-03 -13.035  < 2e-16 ***
## X30.39.years      -1.014e-02  1.991e-03  -5.090 3.57e-07 ***
## I(X30.39.years^2)  3.569e-05  1.807e-05   1.975 0.048215 *
## X50.59.years       1.326e-02  4.085e-03   3.246 0.001170 **
## I(X50.59.years^2) -1.406e-04  6.924e-05  -2.030 0.042312 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(5.3464) family taken to be 1)
##
##     Null deviance: 1491.68  on 783  degrees of freedom
## Residual deviance:  808.36  on 775  degrees of freedom
## AIC: 22873
##
## Number of Fisher Scoring iterations: 1
##
##
##              Theta:  5.346
##          Std. Err.:  0.262
##
##  2 x log-likelihood:  -22853.091

par(mfrow=c(2,2))
plot(predict(bestModelPoly.nb),residuals(bestModelPoly.nb, type="response"))
abline(h=0,lty="dashed", col="red")
## Pearson residuals
plot(predict(bestModelPoly.nb),residuals(bestModelPoly.nb,type="pearson"))
## Deviance residuals
```

```r
plot(predict(bestModelPoly.nb),residuals(bestModelPoly.nb,type="deviance"))
plot(bestModelPoly.nb,which=1)
```




```r
1-pchisq(bestModelPoly.nb$deviance,bestModelPoly.nb$df.residual)

## [1] 0.197038

detach("package:MuMIn")
detach("package:s20x")
detach("package:VGAM")
detach("package:MASS")
```

The Pearson and deviance residuals here look a lot better than our Poisson model. There is no evidence to suggest that these residuals do not come from a Normal(0,1) distribution. Residuals seem just about centred at 1 with approximately constant variance. There is no evidence to suggest a lack of fit for this negative binomial model (p=0.20). Using a log-link function our final model would be… (next page)

$$\log(\mu_i) = \beta_0 + \beta_1 \cdot Bathrooms_i + \beta_2 \cdot Bathrooms_i^2 + \beta_3 \cdot Bathrooms_i^3 + \beta_4 \cdot NZDep2018_i + \beta_5 \cdot X30.39.years_i + \beta_6 \cdot X30.39.years_i^2 + \beta_7 \cdot X50.59.years_i + \beta_8 \cdot X50.59.years_i^2$$

Where $\beta_0, \beta_1, \ldots, \beta_8$ takes on values from the coefficient summaries. $Bathrooms_i$ is the number of bathrooms in the i^{th} house, $NZDep2018_i$ is the depreciation score for the i^{th} house, $X30.39.years_i$ is the number of 30-39-year-olds living in the SA1 unit area and $50.59.years_i$ is the number of 30-39-year-olds living in the SA1 unit area based on the 2018 census. Moreover,

$$Y_i \sim NegBin(\mu_i, \theta)$$

for the i^{th} house. We estimate $\hat{\theta} = 5.3464$ from the summary. Moreover, we have assumed $E(Y) = \mu$ and $Var(Y) = \mu + \frac{\mu^2}{\theta}$. A quadratic relationship between the mean and the variance appears to be suitable to model our data.

Conclusion

A negative binomial regression model is the best fit for predicting CV houses in New Zealand. The significant explanatory variables include the number of bathrooms, the NZ depreciation score, and the number of 30-39 and 50-59-year old's in each SA1 unit area.

Our final model does not include the suburb, but former models showed significance in the house suburb. This could be investigated further into as school zones may depict higher house prices. We can check this by subsetting a new variable that includes the school zones of each house.

It is interesting to note the positive cubic relationship between CV and bathrooms. House prices tend to increase at the greatest rate up to 4 bathrooms while increasing less as we go up to 7. I would have expected prices to increase exponentially because 7 bathrooms would fit a mansion. The NZ depreciation score was negatively linear which would be expected as if a house price decreases annually at a greater rate, the house would be cheaper as it does not hold it's capital value for long. 30-39 year old's tend to go out to work, so the more 30-39-year-olds in an SA1 unit area may depict more apartments and busy streets. Maybe rural places with less 'work' areas would hold less 30-39 year old's in the area so house prices are expensive because it is quieter. This is related to the number of 50-59-year-olds as prices increase the more 50-59-year-olds are situated in the SA1 unit area. There could be less busy roads and houses in the outer rural areas are more expensive but quiet for the elderly.