Saratoga House Prices

Bao Doquang, Dhwanit Agarwal, Akksay Singh and Shristi Singh

March 13, 2020

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.2
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## <U+2713> ggplot2 3.2.1     <U+2713> purrr   0.3.3
## <U+2713> tibble  2.1.3     <U+2713> dplyr   0.8.4
## <U+2713> tidyr   1.0.0     <U+2713> stringr 1.4.0
## <U+2713> readr   1.3.1     <U+2713> forcats 0.4.0
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(mosaic)
```

```
## Warning: package 'mosaic' was built under R version 3.6.2
```

```
## Loading required package: lattice
```

```
## Loading required package: ggformula
```

```
## Loading required package: ggstance
```

```
## Warning: package 'ggstance' was built under R version 3.6.2
```

```
##
## Attaching package: 'ggstance'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```
## Loading required package: mosaicData
```

```
## Warning: package 'mosaicData' was built under R version 3.6.2
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:purrr':
##
##     cross
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library(FNN)
```

```
## Warning: package 'FNN' was built under R version 3.6.2
```

```
library(foreach)
```

```
## Warning: package 'foreach' was built under R version 3.6.3
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
data(SaratogaHouses)

summary(SaratogaHouses)
```

```
##      price            lotSize             age           landValue
##  Min.   :  5000   Min.   : 0.0000   Min.   :  0.00   Min.   :   200
##  1st Qu.:145000   1st Qu.: 0.1700   1st Qu.: 13.00   1st Qu.: 15100
##  Median :189900   Median : 0.3700   Median : 19.00   Median : 25000
##  Mean   :211967   Mean   : 0.5002   Mean   : 27.92   Mean   : 34557
##  3rd Qu.:259000   3rd Qu.: 0.5400   3rd Qu.: 34.00   3rd Qu.: 40200
##  Max.   :775000   Max.   :12.2000   Max.   :225.00   Max.   :412600
##    livingArea     pctCollege       bedrooms       fireplaces       bathrooms
##  Min.   : 616   Min.   :20.00   Min.   :1.000   Min.   :0.0000   Min.   :0.0
##  1st Qu.:1300   1st Qu.:52.00   1st Qu.:3.000   1st Qu.:0.0000   1st Qu.:1.5
##  Median :1634   Median :57.00   Median :3.000   Median :1.0000   Median :2.0
##  Mean   :1755   Mean   :55.57   Mean   :3.155   Mean   :0.6019   Mean   :1.9
##  3rd Qu.:2138   3rd Qu.:64.00   3rd Qu.:4.000   3rd Qu.:1.0000   3rd Qu.:2.5
##  Max.   :5228   Max.   :82.00   Max.   :7.000   Max.   :4.0000   Max.   :4.5
##      rooms                   heating           fuel
##  Min.   : 2.000   hot air        :1121   gas     :1197
##  1st Qu.: 5.000   hot water/steam: 302   electric: 315
##  Median : 7.000   electric       : 305   oil     : 216
##  Mean   : 7.042
##  3rd Qu.: 8.250
##  Max.   :12.000
##                sewer      waterfront newConstruction centralAir
##  septic           : 503   Yes:  15   Yes:  81        Yes: 635
##  public/commercial:1213   No :1713   No :1647        No :1093
##  none             :  12
##
##
##
```

```
#Defining models

# Baseline model
lm_small = lm(price ~ bedrooms + bathrooms + lotSize, data=SaratogaHouses)

# 11 main effects
lm_medium = lm(price ~ lotSize + age + livingArea + pctCollege + bedrooms +
                 fireplaces + bathrooms + rooms + heating + fuel + centralAir, data=SaratogaHouses)

# Sometimes it's easier to name the variables we want to leave out
# The command below yields exactly the same model.
# the dot (.) means "all variables not named"
# the minus (-) means "exclude this variable"
lm_medium2 = lm(price ~ . - sewer - waterfront - landValue - newConstruction, data=SaratogaHouses)

coef(lm_medium)
```

```
##            (Intercept)                lotSize                    age
##            28627.73165             9350.45188               47.54722
##             livingArea             pctCollege               bedrooms
##               91.86974              296.50809           -15630.71950
##             fireplaces              bathrooms                  rooms
##              985.06117            22006.97108             3259.11923
## heatinghot water/steam        heatingelectric           fuelelectric
##            -9429.79463            -3609.98574           -12094.12195
##               fueloil           centralAirNo
```

```
##            -8873.13971           -17112.81908
```

```
coef(lm_medium2)
```

```
##            (Intercept)                lotSize                    age
##            28627.73165             9350.45188               47.54722
##             livingArea             pctCollege               bedrooms
##               91.86974              296.50809           -15630.71950
##             fireplaces              bathrooms                  rooms
##              985.06117            22006.97108             3259.11923
## heatinghot water/steam       heatingelectric           fuelelectric
##            -9429.79463            -3609.98574           -12094.12195
##                fueloil          centralAirNo
##            -8873.13971           -17112.81908
```

```
# All interactions
# the ()^2 says "include all pairwise interactions"
lm_big = lm(price ~ (. - sewer - waterfront - landValue - newConstruction)^2, data=SaratogaHouses)


####
# Compare out-of-sample predictive performance
####

# Split into training and testing sets
n = nrow(SaratogaHouses)
n_train = round(0.8*n)  # round to nearest integer
n_test = n - n_train
train_cases = sample.int(n, n_train, replace=FALSE)
test_cases = setdiff(1:n, train_cases)
saratoga_train = SaratogaHouses[train_cases,]
saratoga_test = SaratogaHouses[test_cases,]

# Fit to the training data
lm1 = lm(price ~ lotSize + bedrooms + bathrooms, data=saratoga_train)
lm2 = lm(price ~ . - sewer - waterfront - landValue - newConstruction, data=saratoga_train)
lm3 = lm(price ~ (. - sewer - waterfront - landValue - newConstruction)^2, data=saratoga_train)

# Predictions out of sample
yhat_test1 = predict(lm1, saratoga_test)
yhat_test2 = predict(lm2, saratoga_test)
yhat_test3 = predict(lm3, saratoga_test)

rmse = function(y, yhat) {
  sqrt( mean( (y - yhat)^2 ) )
}

# Root mean-squared prediction error
rmse(saratoga_test$price, yhat_test1)
```

```
## [1] 77469.29
```

```
rmse(saratoga_test$price, yhat_test2)
```

```
## [1] 62680.03
```

```
rmse(saratoga_test$price, yhat_test3)
```

```
## [1] 61345.99
```

```
# easy averaging over train/test splits
library(mosaic)

n_train = round(0.8*n)  # round to nearest integer
n_test = n - n_train

rmse_vals = do(100)*{

  # re-split into train and test cases with the same sample sizes
  train_cases = sample.int(n, n_train, replace=FALSE)
  test_cases = setdiff(1:n, train_cases)
  saratoga_train = SaratogaHouses[train_cases,]
  saratoga_test = SaratogaHouses[test_cases,]

  # Fit to the training data
  lm1 = lm(price ~ lotSize + bedrooms + bathrooms, data=saratoga_train)
  lm2 = lm(price ~ . - sewer - waterfront - landValue - newConstruction, data=saratoga_train)
  lm3 = lm(price ~ (. - sewer - waterfront - landValue - newConstruction)^2, data=saratoga_train)

  lm_dominate = lm(price ~ lotSize + age + livingArea + pctCollege +
                     bedrooms + fireplaces + bathrooms + rooms + heating + fuel +
                     centralAir + lotSize:heating + livingArea:rooms + newConstruction + livingArea:new

  # Predictions out of sample
  yhat_test1 = predict(lm1, saratoga_test)
  yhat_test2 = predict(lm2, saratoga_test)
  yhat_test3 = predict(lm3, saratoga_test)
  yhat_test4 = predict(lm_dominate, saratoga_test)

  c(rmse(saratoga_test$price, yhat_test1),
    rmse(saratoga_test$price, yhat_test2),
    rmse(saratoga_test$price, yhat_test3),
    rmse(saratoga_test$price, yhat_test4))
}
```

```
## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
```

```
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
```

```
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(lm3, saratoga_test): prediction from a rank-deficient fit
## may be misleading
```

```
rmse_vals
```

```
##          V1       V2        V3       V4
## 1   80789.29 72238.17  73989.03 72297.95
## 2   80337.74 71883.97  70938.05 71068.77
## 3   76217.72 66187.70  65919.10 65330.60
## 4   72381.52 62365.80  61901.99 62312.96
## 5   74002.85 62414.80  66222.01 62713.66
## 6   68546.90 59763.52  62134.87 59280.95
## 7   73131.57 64152.25  69514.25 65368.12
## 8   76230.09 66635.97  72833.72 68349.21
## 9   82846.95 72827.97  83132.72 71896.43
## 10  80870.91 70372.35  68430.94 69497.17
## 11  72315.02 58492.26  58089.38 57576.94
## 12  70743.11 63757.10  63646.71 64115.40
## 13  71930.16 62206.59  75571.46 62246.38
## 14  68785.99 61398.00  59323.45 60706.34
```

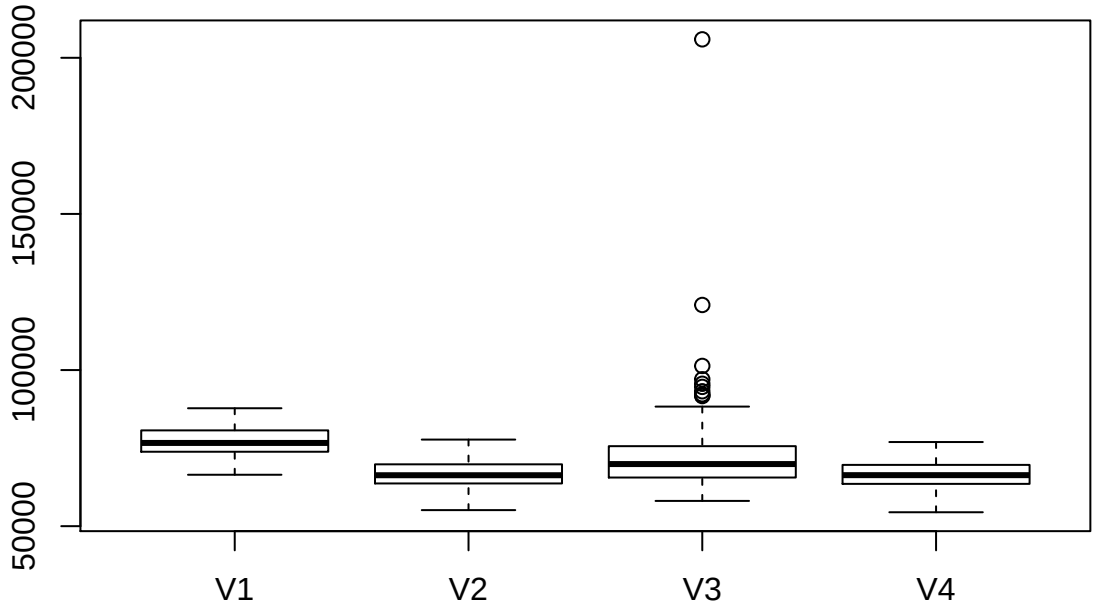
```
## 15  81065.17 68986.04  70316.62 67769.65
## 16  76790.77 67178.12  72046.62 67089.89
## 17  70832.32 60691.84  59463.49 60021.92
## 18  80956.79 71859.56  91864.93 70819.91
## 19  70161.60 62667.16  66085.99 62287.40
## 20  80364.29 68122.11  75285.38 68708.93
## 21  83753.30 74877.72  72773.84 74514.03
## 22  73608.51 64297.90  64112.98 63532.65
## 23  77193.49 69880.40  76345.96 69794.04
## 24  68727.14 61108.87  61683.75 60736.92
## 25  86800.64 77734.86  76648.10 76883.57
## 26  74148.11 64251.49  79805.97 64411.44
## 27  80570.28 69491.63  71008.63 68598.75
## 28  72059.20 61912.03  97047.14 61645.80
## 29  71025.69 62100.53  69053.94 61992.60
## 30  82310.55 71122.25  70270.12 71744.76
## 31  79668.25 67563.21  68212.83 67925.25
## 32  78966.36 68185.11  70027.76 67400.28
## 33  74300.98 66217.61  63459.83 65561.46
## 34  80179.41 71515.83  72803.39 70987.65
## 35  82319.15 75281.04  76252.66 74433.72
## 36  70025.16 55142.12  61604.53 54462.12
## 37  76161.16 62690.87  67436.09 62356.93
## 38  73848.83 66283.54  65975.77 66695.86
## 39  68778.02 59509.36  59822.72 59263.09
## 40  73837.26 62843.38  62103.59 62085.62
## 41  75303.31 64625.79  66480.83 64141.04
## 42  82696.70 73411.94  91656.98 72863.51
## 43  80157.89 66965.79  66766.95 65786.05
## 44  76376.10 72281.05  95603.17 71421.46
## 45  72722.67 64755.53  65263.80 64655.07
## 46  72544.67 60773.29  60977.20 61411.14
## 47  76263.94 64499.38  69822.48 63564.07
## 48  70653.50 57467.68  59116.60 57088.36
## 49  78784.81 69091.71  75843.04 70100.41
## 50  78089.81 65886.72  65547.73 66083.73
## 51  74724.52 65198.57  62717.81 65053.74
## 52  75110.13 65270.69  69039.68 65543.35
## 53  83860.61 75161.11 101320.88 76807.23
## 54  84933.57 77216.91  76917.97 76949.21
## 55  71496.68 64440.69  94605.56 65700.43
## 56  76679.80 65288.35  62041.66 65162.70
## 57  81427.44 72119.00  72969.80 71778.25
## 58  73836.31 61751.50  61544.29 61154.88
## 59  81068.98 68322.86  71474.80 67825.35
## 60  79042.54 69557.18  68281.10 69239.64
## 61  71198.22 63615.98  86049.69 65066.85
## 62  80155.30 69820.25  71971.49 69610.80
## 63  76351.68 67132.88 120840.86 66855.67
## 64  83766.32 68707.16  80882.64 67729.86
## 65  74686.85 66985.18  83331.24 66516.75
## 66  78700.66 68433.00  67285.23 67611.77
## 67  84848.55 73543.20  73386.26 73930.07
## 68  78659.37 67833.45  71218.51 67052.79
```

```
## 69  80727.92 69260.29  75677.81 69637.51
## 70  81332.50 75564.14  73859.19 74667.60
## 71  84488.33 72121.32  69968.91 72123.77
## 72  80231.69 70916.64  88293.65 70296.12
## 73  75237.02 62291.54  59962.78 61706.97
## 74  76107.24 64936.07  68920.07 64653.71
## 75  87765.51 77122.86  76372.57 76412.57
## 76  74129.66 63814.69  93201.88 63300.62
## 77  78109.54 69790.52  68345.68 68933.66
## 78  76804.59 66334.42  64666.82 66404.23
## 79  73281.74 63256.84  65602.25 62723.51
## 80  83785.55 70537.97  71774.54 70045.55
## 81  79584.09 68706.77  71571.53 68090.86
## 82  83698.35 68296.77  71088.89 67287.44
## 83  66469.28 55345.57  59551.94 54519.75
## 84  67995.27 60378.47  61025.23 59719.63
## 85  80634.26 69783.83  72089.17 69498.81
## 86  74184.86 64418.94  72898.18 64394.68
## 87  77914.39 68108.25 205909.72 67314.46
## 88  84780.70 71762.57  76570.48 73294.09
## 89  75351.39 67234.65  67149.99 66277.49
## 90  74191.92 65968.76  65230.05 65263.28
## 91  81590.02 70482.00  71789.78 69799.64
## 92  79236.56 65484.65  65930.96 64517.12
## 93  76913.06 65530.00  92236.05 65130.22
## 94  74155.97 63955.80  76057.94 64595.34
## 95  80872.50 65059.01  67275.85 64474.49
## 96  76295.75 63358.42  68100.57 64091.97
## 97  75844.71 69514.11  68129.33 69041.74
## 98  78229.12 64653.59  69417.71 67188.61
## 99  68228.52 60645.93  62986.83 60894.06
## 100 76635.18 64298.88  68252.18 64245.70
```

```
colMeans(rmse_vals)
```

```
##       V1       V2       V3       V4
## 76965.30 66753.02 73160.27 66517.35
```

```
boxplot(rmse_vals)
```




```
#########################################################################################################################

str(SaratogaHouses)
```

```
## 'data.frame':    1728 obs. of  16 variables:
##  $ price          : int  132500 181115 109000 155000 86060 120000 153000 170000 90000 122900 ...
##  $ lotSize        : num  0.09 0.92 0.19 0.41 0.11 0.68 0.4 1.21 0.83 1.94 ...
##  $ age            : int  42 0 133 13 0 31 33 23 36 4 ...
##  $ landValue      : int  50000 22300 7300 18700 15000 14000 23300 14600 22200 21200 ...
##  $ livingArea     : int  906 1953 1944 1944 840 1152 2752 1662 1632 1416 ...
##  $ pctCollege     : int  35 51 51 51 51 22 51 35 51 44 ...
##  $ bedrooms       : int  2 3 4 3 2 4 4 4 3 3 ...
##  $ fireplaces     : int  1 0 1 1 0 1 1 1 0 0 ...
##  $ bathrooms      : num  1 2.5 1 1.5 1 1 1.5 1.5 1.5 1.5 ...
##  $ rooms          : int  5 6 8 5 3 8 8 9 8 6 ...
##  $ heating        : Factor w/ 3 levels "hot air","hot water/steam",..: 3 2 2 1 1 1 2 1 3 1 ...
##  $ fuel           : Factor w/ 3 levels "gas","electric",..: 2 1 1 1 1 1 3 3 2 1 ...
##  $ sewer          : Factor w/ 3 levels "septic","public/commercial",..: 1 1 2 1 2 1 1 1 1 3 ...
##  $ waterfront     : Factor w/ 2 levels "Yes","No": 2 2 2 2 2 2 2 2 2 2 ...
##  $ newConstruction: Factor w/ 2 levels "Yes","No": 2 2 2 2 1 2 2 2 2 2 ...
##  $ centralAir     : Factor w/ 2 levels "Yes","No": 2 2 2 2 1 2 2 2 2 2 ...
```

```
# New bariables for "hand-built" model

SaratogaHouses$NewBuilt <- ifelse(SaratogaHouses$age == 0, 1,0)
SaratogaHouses$NewBuilt
```

```
##    [1] 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##   [38] 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 0
##   [75] 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
##  [112] 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
##  [149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0
##  [186] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [223] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0
##  [260] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [297] 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [334] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
##  [371] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [408] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [445] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
##  [482] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
##  [519] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0
##  [556] 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
##  [593] 0 0 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 1 0 0 0 0
##  [630] 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
##  [667] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [704] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [741] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0
##  [778] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
##  [815] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [852] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
##  [889] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [926] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [963] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
## [1000] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1037] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1074] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1111] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1148] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
## [1185] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
## [1222] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1259] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1296] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1333] 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1370] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
## [1407] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1444] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
## [1481] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1518] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1555] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
## [1592] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1629] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1666] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [1703] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
```

```
ConstructionCost <- SaratogaHouses$price - SaratogaHouses$landValue
head(ConstructionCost)
```

```
## [1]  82500 158815 101700 136300  71060 106000
```

```
HeatingElectric <- SaratogaHouses[grep("electric", SaratogaHouses$heating), ]
head(HeatingElectric)
```

```
##     price lotSize age landValue livingArea pctCollege bedrooms fireplaces
## 1  132500    0.09  42     50000        906         35        2          1
## 9   90000    0.83  36     22200       1632         51        3          0
## 13  85860    8.97  13      4800        704         41        2          0
## 21 112000    1.00  12      8600       1056         35        3          0
## 22 104900    0.43  21      5600       1600         39        3          0
## 25  90400    0.36  16      5200       1600         39        3          0
##    bathrooms rooms  heating     fuel            sewer waterfront
## 1        1.0     5 electric electric           septic         No
## 9        1.5     8 electric electric           septic         No
## 13       1.0     4 electric electric           septic         No
## 21       1.0     7 electric electric           septic         No
## 22       1.5     4 electric electric public/commercial         No
## 25       1.5     4 electric electric public/commercial         No
##    newConstruction centralAir NewBuilt
## 1               No         No        0
## 9               No         No        0
## 13              No         No        0
## 21              No         No        0
## 22              No         No        0
## 25              No         No        0
```

```
HeatingSteam <- SaratogaHouses[grep("hot water/steam", SaratogaHouses$heating), ]
head(HeatingSteam)
```

```
##     price lotSize age landValue livingArea pctCollege bedrooms fireplaces
## 2  181115    0.92   0     22300       1953         51        3          0
## 3  109000    0.19 133      7300       1944         51        4          1
## 7  153000    0.40  33     23300       2752         51        4          1
## 14  97000    0.11 153      3100       1383         57        3          0
## 16  89900    0.00  88      2500        936         57        3          0
## 19  60000    0.21  82      8500        924         35        2          0
##    bathrooms rooms         heating fuel            sewer waterfront
## 2        2.5     6 hot water/steam  gas           septic         No
## 3        1.0     8 hot water/steam  gas public/commercial         No
## 7        1.5     8 hot water/steam  oil           septic         No
## 14       2.0     5 hot water/steam  gas public/commercial         No
## 16       1.0     4 hot water/steam  gas public/commercial         No
## 19       1.0     6 hot water/steam  oil           septic         No
##    newConstruction centralAir NewBuilt
## 2               No         No        1
## 3               No         No        0
## 7               No         No        0
## 14              No         No        0
## 16              No         No        0
## 19              No         No        0
```

```
HeatingHotAir <- SaratogaHouses[grep("hot air", SaratogaHouses$heating), ]
head(HeatingHotAir)
```

```
##     price lotSize age landValue livingArea pctCollege bedrooms fireplaces
## 4  155000    0.41  13     18700       1944         51        3          1
## 5   86060    0.11   0     15000        840         51        2          0
## 6  120000    0.68  31     14000       1152         22        4          1
## 8  170000    1.21  23     14600       1662         35        4          1
```

```
## 10 122900    1.94   4     21200       1416         44        3          0
## 11 325000    2.29 123     12600       2894         51        7          0
##    bathrooms rooms heating fuel             sewer waterfront newConstruction
## 4        1.5     5 hot air  gas            septic         No              No
## 5        1.0     3 hot air  gas public/commercial         No             Yes
## 6        1.0     8 hot air  gas            septic         No              No
## 8        1.5     9 hot air  oil            septic         No              No
## 10       1.5     6 hot air  gas              none         No              No
## 11       1.0    12 hot air  oil            septic         No              No
##    centralAir NewBuilt
## 4          No        0
## 5         Yes        1
## 6          No        0
## 8          No        0
## 10         No        0
## 11         No        0
```

```
FuelOil <- SaratogaHouses[grep("oil", SaratogaHouses$fuel), ]
head(FuelOil)
```

```
##     price lotSize age landValue livingArea pctCollege bedrooms fireplaces
## 7  153000    0.40  33     23300       2752         51        4          1
## 8  170000    1.21  23     14600       1662         35        4          1
## 11 325000    2.29 123     12600       2894         51        7          0
## 15 127000    0.14   9       300       1300         41        3          0
## 17 155000    0.13   9       300       1300         41        3          0
## 19  60000    0.21  82      8500        924         35        2          0
##    bathrooms rooms         heating fuel  sewer waterfront newConstruction
## 7        1.5     8 hot water/steam  oil septic         No              No
## 8        1.5     9         hot air  oil septic         No              No
## 11       1.0    12         hot air  oil septic         No              No
## 15       1.5     8         hot air  oil septic         No              No
## 17       1.5     7         hot air  oil septic         No              No
## 19       1.0     6 hot water/steam  oil septic         No              No
##    centralAir NewBuilt
## 7          No        0
## 8          No        0
## 11         No        0
## 15         No        0
## 17         No        0
## 19         No        0
```

```
FuelGas <- SaratogaHouses[grep("gas", SaratogaHouses$fuel), ]
head(FuelGas)
```

```
##     price lotSize age landValue livingArea pctCollege bedrooms fireplaces
## 2  181115    0.92   0     22300       1953         51        3          0
## 3  109000    0.19 133      7300       1944         51        4          1
## 4  155000    0.41  13     18700       1944         51        3          1
## 5   86060    0.11   0     15000        840         51        2          0
## 6  120000    0.68  31     14000       1152         22        4          1
## 10 122900    1.94   4     21200       1416         44        3          0
##    bathrooms rooms         heating fuel             sewer waterfront
## 2        2.5     6 hot water/steam  gas            septic         No
## 3        1.0     8 hot water/steam  gas public/commercial         No
## 4        1.5     5         hot air  gas            septic         No
```

```
## 5        1.0     3         hot air  gas public/commercial         No
## 6        1.0     8         hot air  gas            septic         No
## 10       1.5     6         hot air  gas              none         No
##    newConstruction centralAir NewBuilt
## 2               No         No        1
## 3               No         No        0
## 4               No         No        0
## 5              Yes        Yes        1
## 6               No         No        0
## 10              No         No        0
```

```
FuelElectric <- SaratogaHouses[grep("electric", SaratogaHouses$fuel), ]
head(FuelElectric)
```

```
##     price lotSize age landValue livingArea pctCollege bedrooms fireplaces
## 1  132500    0.09  42     50000        906         35        2          1
## 9   90000    0.83  36     22200       1632         51        3          0
## 13  85860    8.97  13      4800        704         41        2          0
## 21 112000    1.00  12      8600       1056         35        3          0
## 22 104900    0.43  21      5600       1600         39        3          0
## 25  90400    0.36  16      5200       1600         39        3          0
##    bathrooms rooms  heating     fuel             sewer waterfront
## 1        1.0     5 electric electric            septic         No
## 9        1.5     8 electric electric            septic         No
## 13       1.0     4 electric electric            septic         No
## 21       1.0     7 electric electric            septic         No
## 22       1.5     4 electric electric public/commercial         No
## 25       1.5     4 electric electric public/commercial         No
##    newConstruction centralAir NewBuilt
## 1               No         No        0
## 9               No         No        0
## 13              No         No        0
## 21              No         No        0
## 22              No         No        0
## 25              No         No        0
```

```
#Defining the models
#Base model

BaseModel = lm(price ~ lotSize + age + livingArea + pctCollege + bedrooms + fireplaces +
               heating + bathrooms + rooms + fuel + centralAir + NewBuilt, data = SaratogaHouses)

#Hand Built Model
HandBuiltModel = lm(price ~ lotSize + pctCollege  + heating + bathrooms + bedrooms
            + rooms + fuel + centralAir + NewBuilt + landValue + NewBuilt*lotSize
            + centralAir*heating + pctCollege*age + landValue*fuel + heating*bedrooms
            , data = SaratogaHouses)

#Define only the numerics of the train-test data sets
N = nrow(SaratogaHouses)
train = round(0.8*N)
test = (N-train)
#Define the futions
rmse = function(y, yhat) {
  sqrt( mean( (y - yhat)^2 ) )
```

```
}

#Rmse iterations
rmse1 <- NULL
rmse2 <- NULL

for (i in seq(1:200)){
  #Picking data up for training and testing
  train_cases = sample.int(N, train, replace=FALSE)
  test_cases = setdiff(1:N, train_cases)

  #Define the train-test data sets (for all X's and Y)
  saratoga_train = SaratogaHouses[train_cases,]
  saratoga_test = SaratogaHouses[test_cases,]

  #Training
  #Base Model
  lm1 = lm(price ~ lotSize + age + livingArea + pctCollege + bedrooms + fireplaces +
             heating + bathrooms + rooms + fuel + centralAir + NewBuilt , data=saratoga_train)
  #Hand-built Model
  lm2 = lm(price ~ lotSize + pctCollege  + heating  + bathrooms + bedrooms
           + rooms + fuel + centralAir + NewBuilt + landValue + NewBuilt*lotSize
           + centralAir*heating + pctCollege*age + landValue*fuel + heating*bedrooms
           , data=saratoga_train)

  #Testing
  yhat_test1 = predict(lm1, saratoga_test)
  yhat_test2 = predict(lm2, saratoga_test)

  #Run it on the actual and the predicted values
  rmse1[i]= rmse(saratoga_test$price, yhat_test1)
  rmse2[i]= rmse(saratoga_test$price, yhat_test2)
}

mean(rmse1)
```

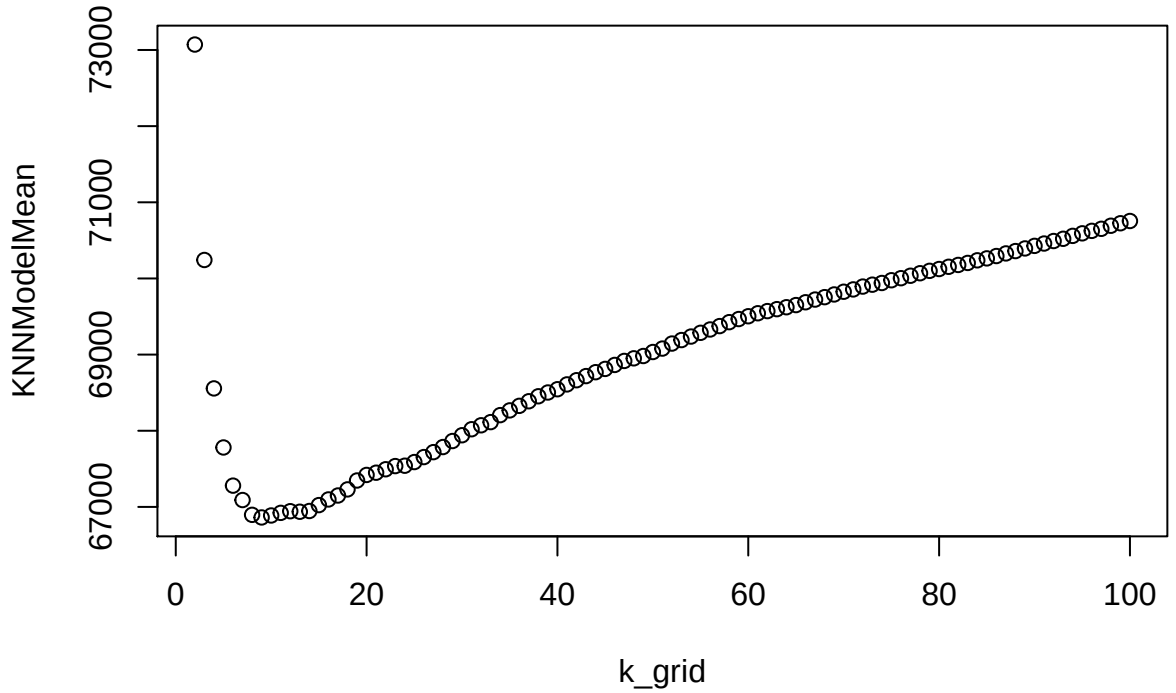
```
## [1] 66578.26
```

```
mean(rmse2)
```

```
## [1] 63829.28
```

```
# K-Nearest Neighbors Model

#Defining train-test sets for the hand-built regression model
KNNModel = do(100)*{
  N = nrow(SaratogaHouses)
  train = round(0.8*N)
  test = (N-train)

  train_cases = sample.int(N, train, replace=FALSE)
  test_cases = setdiff(1:N, train_cases)

  saratoga_train = SaratogaHouses[train_cases,]
  saratoga_test = SaratogaHouses[test_cases,]
```

```
  Xtrain = model.matrix(~ lotSize + pctCollege  + heating  + bathrooms + bedrooms
                        + rooms + fuel + centralAir + NewBuilt + landValue - 1, data=saratoga_train)
  Xtest = model.matrix(~ lotSize + pctCollege  + heating  + bathrooms + bedrooms
                       + rooms + fuel + centralAir + NewBuilt + landValue - 1, data=saratoga_test)
  Ytrain = saratoga_train$price
  Ytest = saratoga_test$price

  #Scaling the features (Standardization)
  scale_train = apply(Xtrain, 2, sd)
  Xtilde_train = scale(Xtrain, scale = scale_train)
  Xtilde_test = scale(Xtest, scale = scale_train)

  #The for loop
    k_grid = seq(2,100)
  rmse_grid = foreach(K = k_grid, .combine='c') %do% {
    KNNModel = knn.reg(Xtilde_train, Xtilde_test, Ytrain, k=K)
    rmse(Ytest, KNNModel$pred)
  }
}
KNNModelMean = colMeans(KNNModel)

#Plotting
plot(k_grid, KNNModelMean)
abline(h=rmse(Ytest, yhat_test2))
```



```
#We conclude that variables giving the same data that is completely captured by another variable can be
#Additionally, we have found that newer houses are bigger and are correlated with an increase in pricin
```