Regression Analysis of Energy Efficiency in Residential Buildings

Daniel Yang, Sharon Yang

December 04, 2018

Abstract

Among the interests of producers and tenants regarding residential buildings are that of energy efficiency - the amount of energy required to heat or cool a building unit to an acceptable level. This performance detail is relevant throughout the sales process, from meeting building regulations to long term utility pricing. To explore the factors that contribute to energy efficiency (which industry refers to as 'heating load' and 'cooling load'), we use data from Xifara and Tsanas [*Energy and Buildings*, Vol. 49, pp. 560-567, 2012]; namely, data involving compactness, surface area, wall area, roof area, height, orientation, and glazing area. We suspect these variables all affect energy efficiency. By constructing several regression models and employing a variety of analysis methods (Variable Selection, LASSO, Re-sampling, Regression Trees and Random Forest), we find that a log-transformed model including compactness, surface area, wall area, height, and glazing area to offer the most with respect to the aforementioned criteria, and that compactness and surface area are the most significant factors. These variables should therefore be first considered by prospective builders and tenants. We relax some regression assumptions and other variables (particularly that residential buildings are constructed in Greece) are unstated but controlled - limiting our conclusions to areas with similar climate.

Contents

1 Introduction

1.1 Background

Energy efficiency has always been a point of interest for consumers and producers. The associated cost of owning, renting, or supplying a residential building all involve power, and so increasing energy efficiency would translate to non-trivial savings, among other conscious decisions such as minimizing carbon footprint. Having established a suitable motive for studying this topic, an appropriate data set is considered. In their paper, Xifara and Tsanas (2012) simulates residential buildings with various properties, and reports the estimated heating and cooling loads for each simulation. The heating and cooling loads refers to the amount of heat energy added or removed to maintain an acceptable temperature respectively. A lower heating or cooling load is therefore an indication of a more energy efficient building.

The goal of this report is focused on establishing the relationships of the predictors with respect to heating load in the data set. The validity of the relationships are then assessed via the predictive power of the models that are based on those relationships. A variety of different methods are constructed and the findings are reported. Since the emphasis is on the more physical properties of a residential building, plausible effects on energy efficiency such as temperature and building materials are controlled. Additionally, the simulated buildings are based in Greece, perhaps limiting the inferential scope of the findings to similar climates. Nonetheless, the results generally agree with statistical and scientific literature. This offers consumers and producers a relatively accurate, if imprecise, rule of thumb in estimating energy efficiency in a residential building.

1.2 Variables

As previously mentioned, heating and cooling loads describe the amount of heat energy necessary to maintain reasonable indoor temperatures. While not explicitly detailed in the data set, these values appear to be measured in British thermal units, or BTU. Relative compactness refers to the ratios of a considered shape of the building to the reference shape, since to compare the surface to volume ratio ratio of different shapes, the volumes must be equal. A paper by Geletka and Sedlakova (2012) succinctly expresses this quantity as

$$RC = \frac{(V/A)_{building}}{(V/A)_{ref}}$$

Where $\frac{V}{A}$ refers to the ratio of volume to surface area. Since relative compactness is a ratio (of ratios), it does not have a unit. Surface area, wall area, roof area, are fairly self explanatory, referring to the areas that define and enclose the building, and specific areas that are known as ‘walls’ and ‘roofs’ respectively. These variables are measured in m^2. Overall height measures the length of the building and is measured in m. Orientation refers to the cardinal direction of which the building is facing. Glazing area and glazing area distribution are two related but distinct variables; glazing area refers to the % of glaze covering a floor area, whilst glazing area distribution describes which cardinal direction the glaze is concentrated in (about 55%). For reference, glazing in this context refers to “A covering of transparent or translucent material (typically glass or plastic) used for admitting light.” (Glossary of Energy Terms, 2016)(“Glossary of Energy Terms,” n.d.) A table is constructed denoting the variable types:

Table 1: Variable types within the data set

Variable name	Variable type
Heating Load	Continuous
Cooling Load	Continuous
Relative Compactness	Continuous
Surface Area	Continuous
Wall Area	Continuous
Roof Area	Continuous
Overall Height	Ordinal
Orientation	Categorical
Glazing Area	Ordinal
Glazing Area Distribution	Categorical

Some justification and explanations may be required. Orientation is categorical and is encoded to take 4 different values: 2, 3, 4, and 5. Although the amount of values matches the amount of directions, it is not mentioned in the data set description which number associates to which cardinal direction. Overall height may be rightfully interpreted as continuous, however in this data set only takes upon 2 values; it may better suit the needs of analysis for it to be classified as ordinal. Glazing area takes on values of 0, 0.1, 0.25, and 0.4, which indicate percentages. While not continuous nor discrete, there is certainly an ordering to it. Glazing area distribution is clearly categorical. The corresponding directions to the values 0, 1, 2, 3, 4, and 5 are uniform (glazing area evenly distributed), north, east, south, and west.

1.3 The Dataset

The data set will be briefly described here. Below is the constructed tibble of the *energy.csv* datafile used in this report:

```
## # A tibble: 768 x 10
##       RC    SA    WA    RA    OH     O    GA   GAD    HL    CL
##  * <dbl> <dbl> <dbl> <dbl> <dbl> <int> <dbl> <int> <dbl> <dbl>
##  1  0.98  514.  294   110.     7     2     0     0  15.6  21.3
##  2  0.98  514.  294   110.     7     3     0     0  15.6  21.3
##  3  0.98  514.  294   110.     7     4     0     0  15.6  21.3
##  4  0.98  514.  294   110.     7     5     0     0  15.6  21.3
##  5  0.9   564.  318.  122.     7     2     0     0  20.8  28.3
##  6  0.9   564.  318.  122.     7     3     0     0  21.5  25.4
##  7  0.9   564.  318.  122.     7     4     0     0  20.7  25.2
##  8  0.9   564.  318.  122.     7     5     0     0  19.7  29.6
##  9  0.86  588   294   147      7     2     0     0  19.5  27.3
## 10  0.86  588   294   147      7     3     0     0  20.0  22.0
## # ... with 758 more rows
```

This tibble slightly differs from the original data set as it omits missing entries and abbreviates the names of the variables. These abbreviations logically match the variables described earlier. Note that the variable data type and sample size are also included in the tibble.

As previously mentioned, orientation and glazing area distributions are categorical variables unsuited for quantitative interpretation. Overall height and glazing area, while quantitative in nature, more closely relates to ordinal variables as height dichotomizes the data set into 2 distinct groups and glazing area is reported in percentages. To address these problems, they are factored in R as dummy variables (with height and glazing area being ordered):

```r
OH.F = factor(energy$OH, ordered = T)
GA.F = factor(energy$GA, ordered = T)
GAD.F = as.factor(energy$GAD)
O.F = as.factor(energy$O)
```

Our updated data set can now be represented as

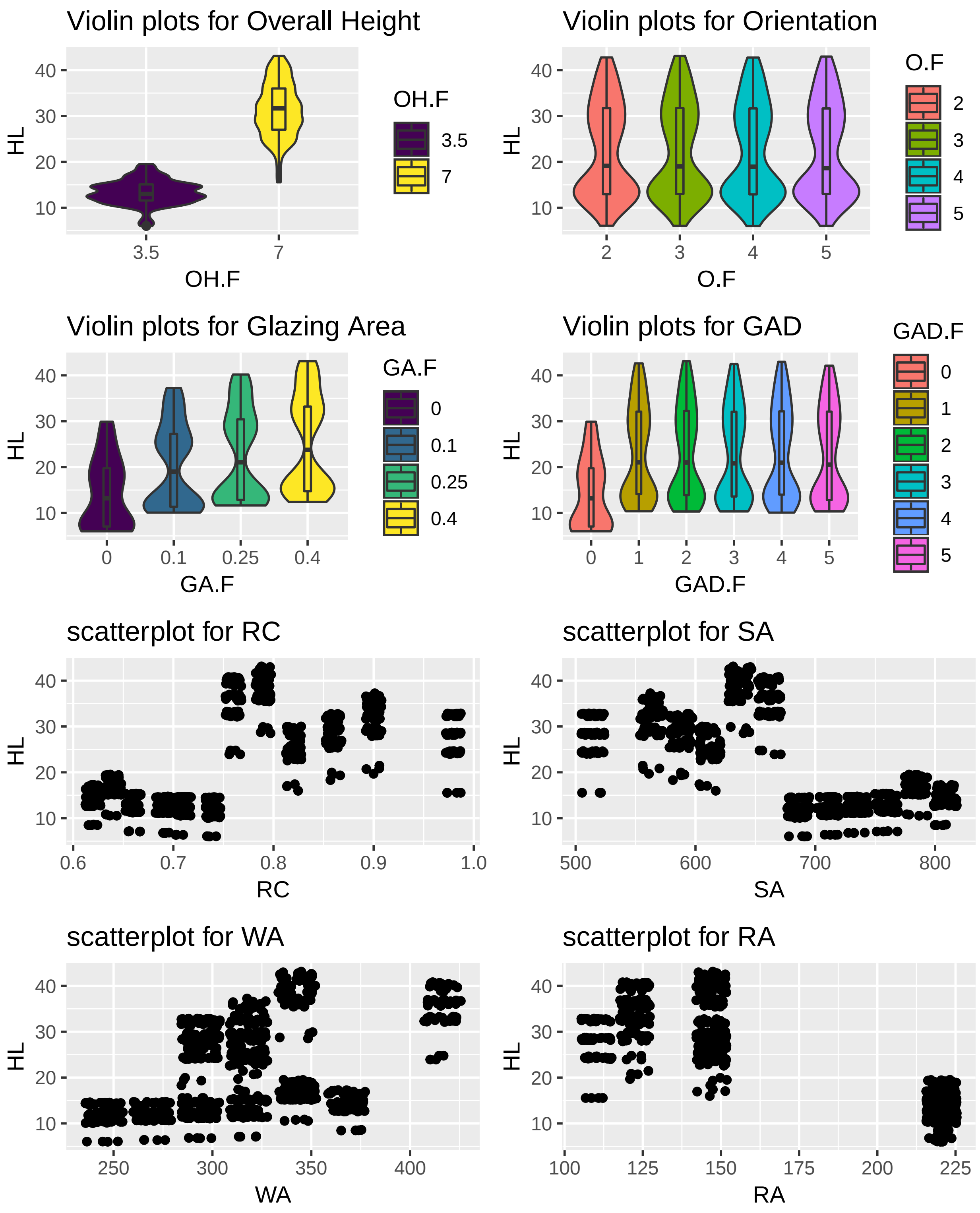
```
## # A tibble: 768 x 9
##       RC    SA    WA    RA    HL OH.F  GA.F  GAD.F O.F
##    <dbl> <dbl> <dbl> <dbl> <dbl> <ord> <ord> <fct> <fct>
##  1  0.98  514.  294   110.  15.6 7     0     0     2
##  2  0.98  514.  294   110.  15.6 7     0     0     3
##  3  0.98  514.  294   110.  15.6 7     0     0     4
##  4  0.98  514.  294   110.  15.6 7     0     0     5
##  5  0.9   564.  318.  122.  20.8 7     0     0     2
##  6  0.9   564.  318.  122.  21.5 7     0     0     3
##  7  0.9   564.  318.  122.  20.7 7     0     0     4
##  8  0.9   564.  318.  122.  19.7 7     0     0     5
##  9  0.86  588   294   147   19.5 7     0     0     2
## 10  0.86  588   294   147   20.0 7     0     0     3
## # ... with 758 more rows
```

As *CL* is not considered in the project, it is removed.

2 Methods

2.1 Visualising data

The data set is first visualised to get some preliminary sense of how the predictors are distributed. Using the ggplot2 package in R, the relevant plots are constructed on the next page:



Violin plots were chosen for discrete predictors and scatter plots for continuous predictors. From the visualisations, it is observed that among the factored variables most of the violin plots are bimodally distributed, although the embedded boxplots indicate symmetry. Additionally, there's a clear upwards trend in the glazing area plot, which suggests a positive correlation between glazing area percentages that cover a residential building and heating load. Since orientation and most of the glazing area distribution violin plots maintain the same median, it raises questions to be verified if they are significant. Among the continuous predictors, the scatterplots seem to be clearly separated into two distinct groups, with marginal linear relations. The similarities in the scatterplot patterns also suggest high collinearity, although to be certain variance-inflation factors should be computed. Finally, the median and distribution of the violin plot for overall height makes it evident that a higher heat load corresponds to height. With only two levels, it may be worthwhile to investigate overall height as the potential reason behind the bimodal distributions and scatterplot groups.

2.2 Multiple Linear Regression

2.2.1 Model 1

It is befitting for the first model to be the simplest. This model contains all possible predictors.

```r
model.1 = lm(HL ~ ., data = energy)
summary(model.1)
```

```
##
## Call:
## lm(formula = HL ~ ., data = energy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1238 -1.4333 -0.1815  1.2628  7.3917
##
## Coefficients: (2 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.094e+02  1.695e+01   6.452 1.98e-10 ***
## RC          -6.477e+01  9.837e+00  -6.585 8.55e-11 ***
## SA          -8.729e-02  1.632e-02  -5.347 1.19e-07 ***
## WA           6.081e-02  6.356e-03   9.569  < 2e-16 ***
## RA                  NA         NA      NA       NA
## OH.F.L       1.032e+01  7.997e-01  12.905  < 2e-16 ***
## GA.F.L       7.892e+00  3.339e-01  23.635  < 2e-16 ***
## GA.F.Q      -1.627e+00  2.766e-01  -5.884 6.03e-09 ***
## GA.F.C       8.419e-01  2.038e-01   4.131 4.02e-05 ***
## GAD.F1       3.452e-01  3.306e-01   1.044    0.297
## GAD.F2       2.535e-01  3.306e-01   0.767    0.443
## GAD.F3       5.556e-04  3.306e-01   0.002    0.999
## GAD.F4       2.058e-01  3.306e-01   0.622    0.534
## GAD.F5              NA         NA      NA       NA
## O.F3         6.781e-02  2.863e-01   0.237    0.813
## O.F4        -5.297e-02  2.863e-01  -0.185    0.853
## O.F5        -3.750e-02  2.863e-01  -0.131    0.896
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.805 on 753 degrees of freedom
## Multiple R-squared:  0.9241, Adjusted R-squared:  0.9227
## F-statistic:   655 on 14 and 753 DF,  p-value: < 2.2e-16
```

An immediate problem is presented from the summary: The variables *RA* and *GAD.F5* do not have defined slopes. This is evidence of perfect multicollinearity - both *RA* and *GAD.F5* can be perfectly predicted by another variable. This is a critical issue, since an over-determined model does not have unique estimates.

2.2.2 Model 2

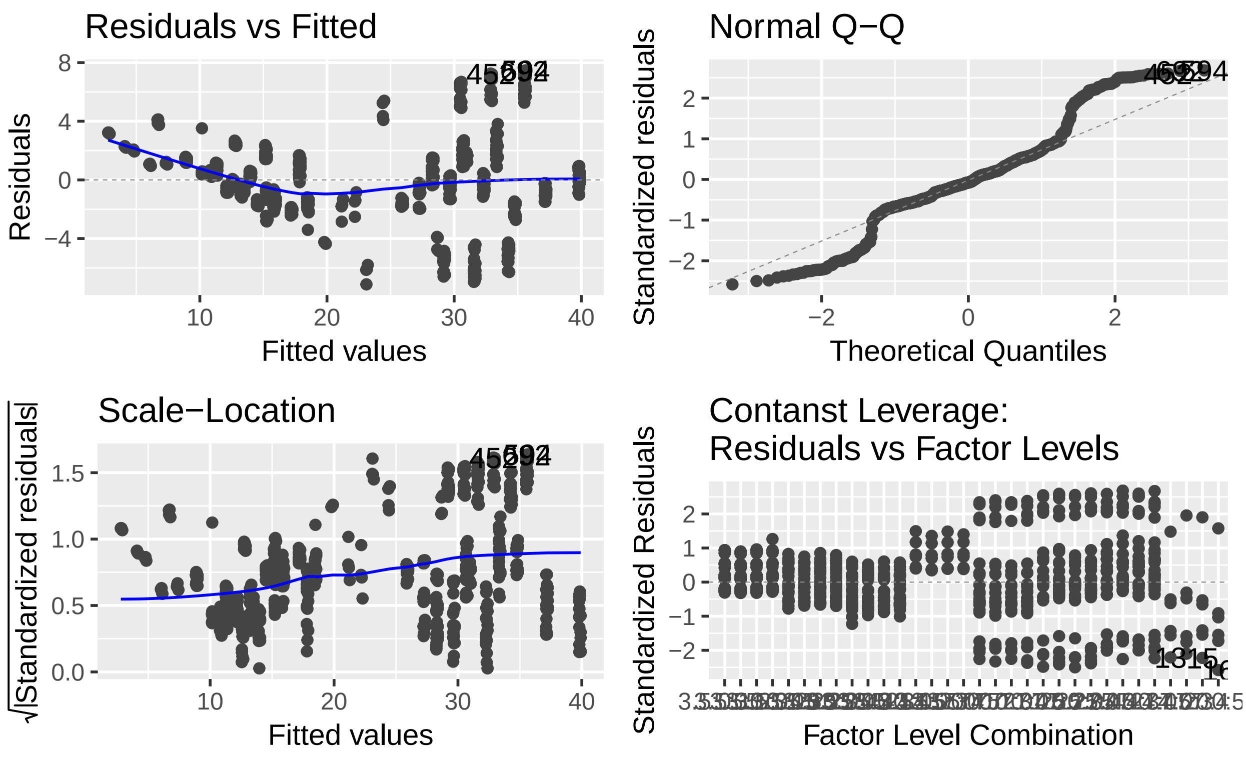
The second model drops the offending predictors:

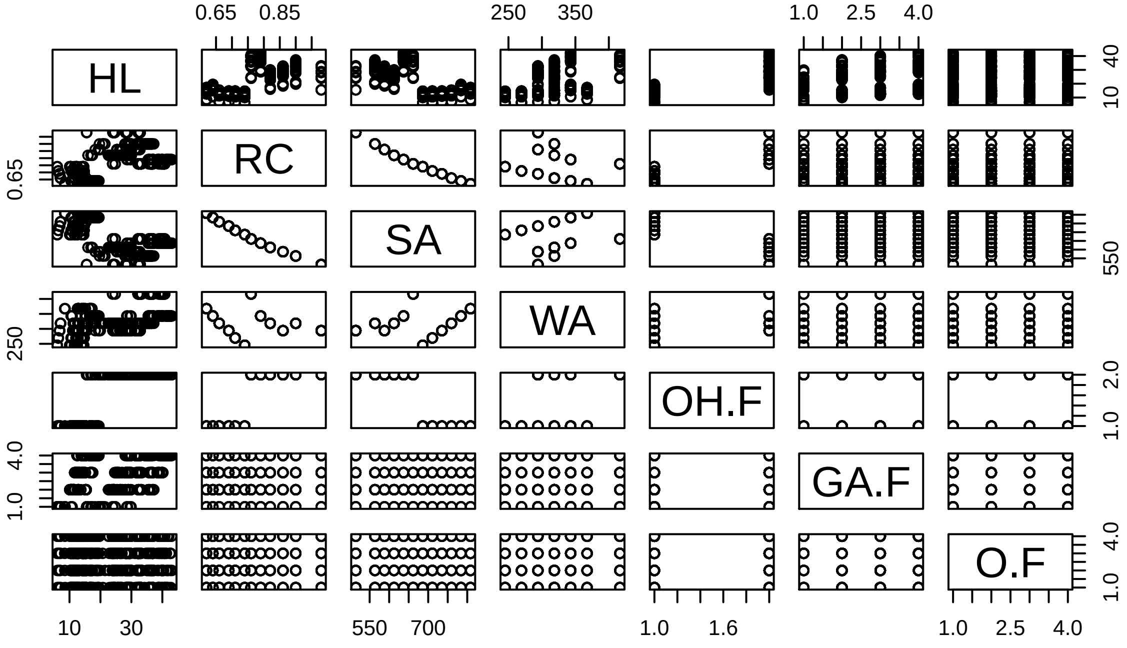
```r
model.2 = lm(HL ~ RC + SA + WA + OH.F + GA.F + O.F, data = energy)
summary(model.2)
```

```
##
## Call:
## lm(formula = HL ~ RC + SA + WA + OH.F + GA.F + O.F, data = energy)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.124 -1.457 -0.210  1.352  7.471
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 109.471397  16.923204   6.469 1.77e-10 ***
## RC          -64.773991   9.822379  -6.595 8.00e-11 ***
## SA           -0.087290   0.016300  -5.355 1.14e-07 ***
## WA            0.060813   0.006346   9.583  < 2e-16 ***
## OH.F.L       10.320072   0.798512  12.924  < 2e-16 ***
## GA.F.L        7.999617   0.302556  26.440  < 2e-16 ***
## GA.F.Q       -1.707979   0.255706  -6.679 4.64e-11 ***
## GA.F.C        0.877871   0.198069   4.432 1.07e-05 ***
## O.F3          0.067812   0.285888   0.237    0.813
## O.F4         -0.052969   0.285888  -0.185    0.853
## O.F5         -0.037500   0.285888  -0.131    0.896
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.801 on 757 degrees of freedom
## Multiple R-squared:  0.9239, Adjusted R-squared:  0.9229
## F-statistic: 919.5 on 10 and 757 DF,  p-value: < 2.2e-16
```

No errors are present in the summary. Observe that the median is quite close to 0, and only orientation is statistically insignificant at $\alpha = 0.05$ (Glazing area distribution is also found to be statistically insignificant). Adjusted R^2 gives 0.9229, or 92.29% of the variation in heating load is explained in this model. Since no other problem is apparent at this stage, the diagnostic plots[1], pairs plot, and variance-inflation factors are considered:

[1] The apparent mispelling error "Contanst Leverage" is an error created by the ggplot package; see https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_lm.html for an example.





```
##              GVIF Df GVIF^(1/(2*Df))
## RC    105.524054  1       10.272490
## SA    201.531134  1       14.196166
## WA      7.492984  1        2.737332
## OH.F   31.205474  1        5.586186
## GA.F    1.000000  3        1.000000
## O.F     1.000000  3        1.000000
```

It is almost depressingly apparent that this model is inadequate in satisfying any of the linear regression assumptions. From the residuals vs. fitted plot a funnel shape pattern emerges, indicating heteroskedasticity (nonequal variances in the response at different levels). The QQ plot is not linear, meaning the errors are not normally distributed. From the pairs plot there are no linear relationship between *HL* and any of the continuous predictors. Finally the variance-inflation factor gives off some spectacularly awful results, with only *GA.F* and *O.F* being below the value of 5; there is extreme collinearity among the rest of the predictors.

2.2.3 Model 3

To improve upon the model, the Box-Cox power transformation is considered:

```
summary(powerTransform(cbind(HL,RC,SA,WA)~1,data= energy))
```

```
## bcPower Transformations to Multinormality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## HL    0.2433        0.33       0.1376       0.3491
## RC   -1.5373       -1.54      -1.8490      -1.2256
## SA    1.6944        2.00       1.3717       2.0170
## WA   -0.5711       -0.50      -0.8882      -0.2540
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                  LRT df       pval
## LR test, lambda = (0 0 0 0) 196.497  4 < 2.22e-16
##
## Likelihood ratio test that no transformations are needed
##                                   LRT df       pval
## LR test, lambda = (1 1 1 1) 2792.464  4 < 2.22e-16
```

```
bcHL = (energy$HL)^0.33
bcRC = (energy$RC)^-1.54
bcSA = (energy$SA)^2
bcWA = (energy$WA)^-0.5

model.3 = lm(bcHL ~ bcRC + bcSA + bcWA + energy$OH.F + energy$GA.F + energy$O.F)
summary(model.3)
```

```
##
## Call:
## lm(formula = bcHL ~ bcRC + bcSA + bcWA + energy$OH.F + energy$GA.F +
##     energy$O.F)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23180 -0.05663 -0.00296  0.05177  0.31754
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.677e+00  2.115e-01  17.386  < 2e-16 ***
## bcRC          -7.434e-01  4.333e-01  -1.715   0.0867 .
## bcSA           2.467e-06  1.268e-06   1.945   0.0521 .
## bcWA          -1.775e+01  2.600e+00  -6.826 1.80e-11 ***
## energy$OH.F.L  5.796e-01  3.024e-02  19.169  < 2e-16 ***
## energy$GA.F.L  3.737e-01  1.030e-02  36.286  < 2e-16 ***
## energy$GA.F.Q -1.026e-01  8.705e-03 -11.787  < 2e-16 ***
## energy$GA.F.C  4.843e-02  6.743e-03   7.183 1.63e-12 ***
## energy$O.F3    2.630e-03  9.732e-03   0.270   0.7871    
## energy$O.F4   -2.532e-03  9.732e-03  -0.260   0.7948    
## energy$O.F5   -1.037e-03  9.732e-03  -0.107   0.9152    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.09536 on 757 degrees of freedom
## Multiple R-squared:  0.9498, Adjusted R-squared:  0.9492 
## F-statistic:  1433 on 10 and 757 DF,  p-value: < 2.2e-16
```

2.2.4 Model 4

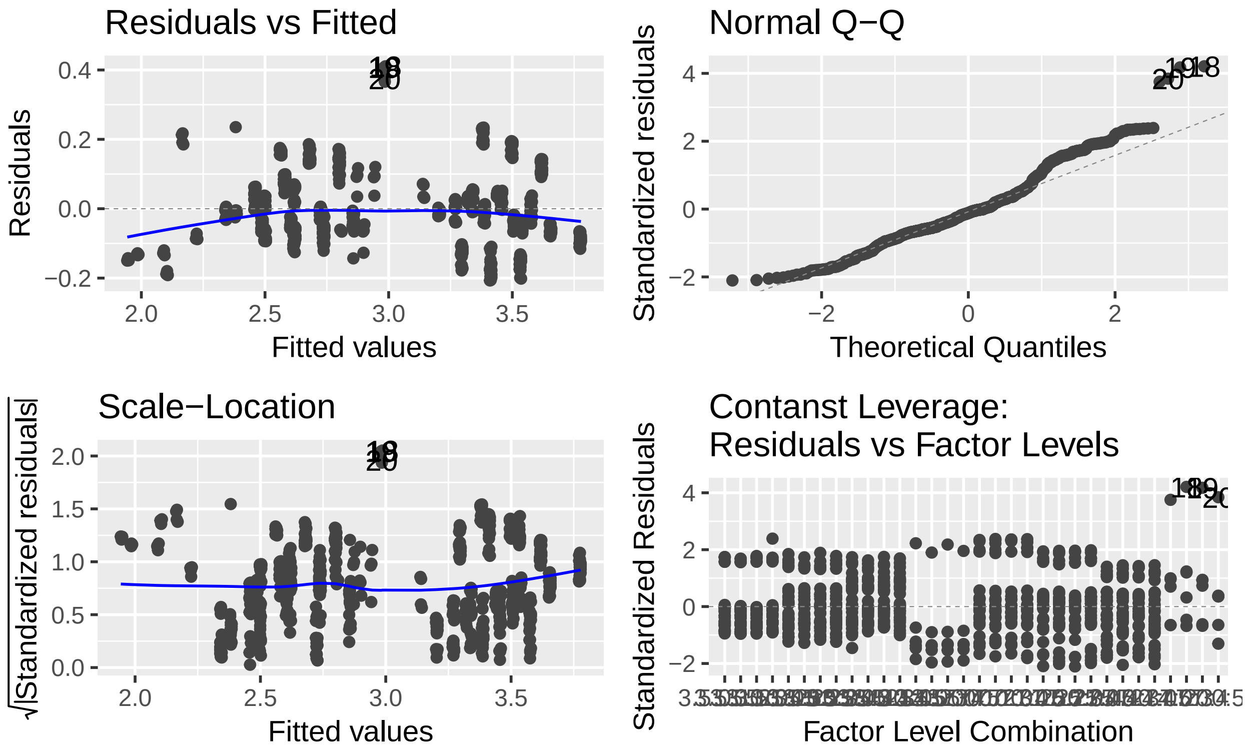
From the summary the estimated Box-Cox transformation has revealed an unintended consequence: the once statistically significant variables RC and SA are now statistically insignificant, likely due to the nature of their high collinearity. While it's possible that variable selection can correct this issue, the model would still suffer from lack of interpretability. The logarithm transformation is instead considered; in addition to being a logical transformation, it also reasonably easy to interpret.

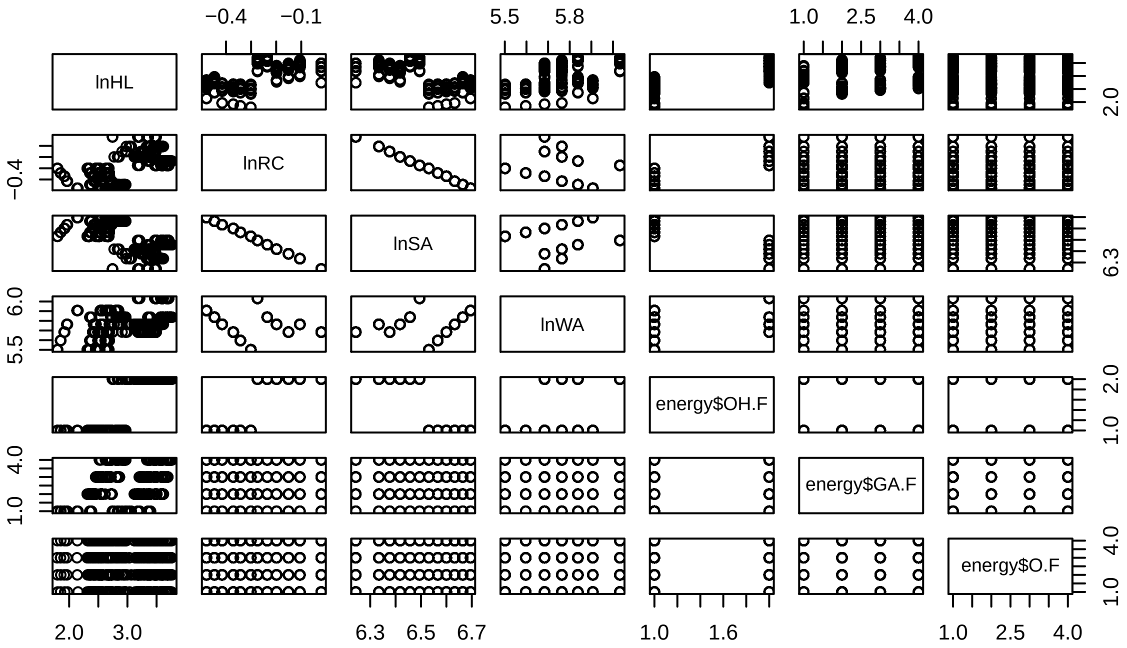
```
lnHL = log(energy$HL)
lnRC = log(energy$RC)
lnSA = log(energy$SA)
lnWA = log(energy$WA)

model.4 = lm(lnHL ~ lnRC + lnSA + lnWA + energy$OH.F + energy$GA.F + energy$O.F)
summary(model.4)
```

```
## 
## Call:
## lm(formula = lnHL ~ lnRC + lnSA + lnWA + energy$OH.F + energy$GA.F + 
##     energy$O.F)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.20685 -0.06252 -0.01139  0.04745  0.41055 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   -4.965e+01  7.215e+00  -6.881 1.25e-11 ***
## lnRC           7.375e+00  1.126e+00   6.547 1.08e-10 ***
## lnSA           7.709e+00  1.140e+00   6.761 2.74e-11 ***
## lnWA           7.792e-01  5.730e-02  13.598  < 2e-16 ***
## energy$OH.F.L  6.119e-01  1.871e-02  32.697  < 2e-16 ***
## energy$GA.F.L  4.510e-01  1.069e-02  42.184  < 2e-16 ***
## energy$GA.F.Q -1.383e-01  9.036e-03 -15.307  < 2e-16 ***
## energy$GA.F.C  6.333e-02  6.999e-03   9.048  < 2e-16 ***
## energy$O.F3    2.822e-03  1.010e-02   0.279    0.780    
## energy$O.F4   -3.064e-03  1.010e-02  -0.303    0.762    
## energy$O.F5   -8.344e-04  1.010e-02  -0.083    0.934    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.09898 on 757 degrees of freedom
## Multiple R-squared:  0.9574, Adjusted R-squared:  0.9569
## F-statistic:  1703 on 10 and 757 DF,  p-value: < 2.2e-16
```

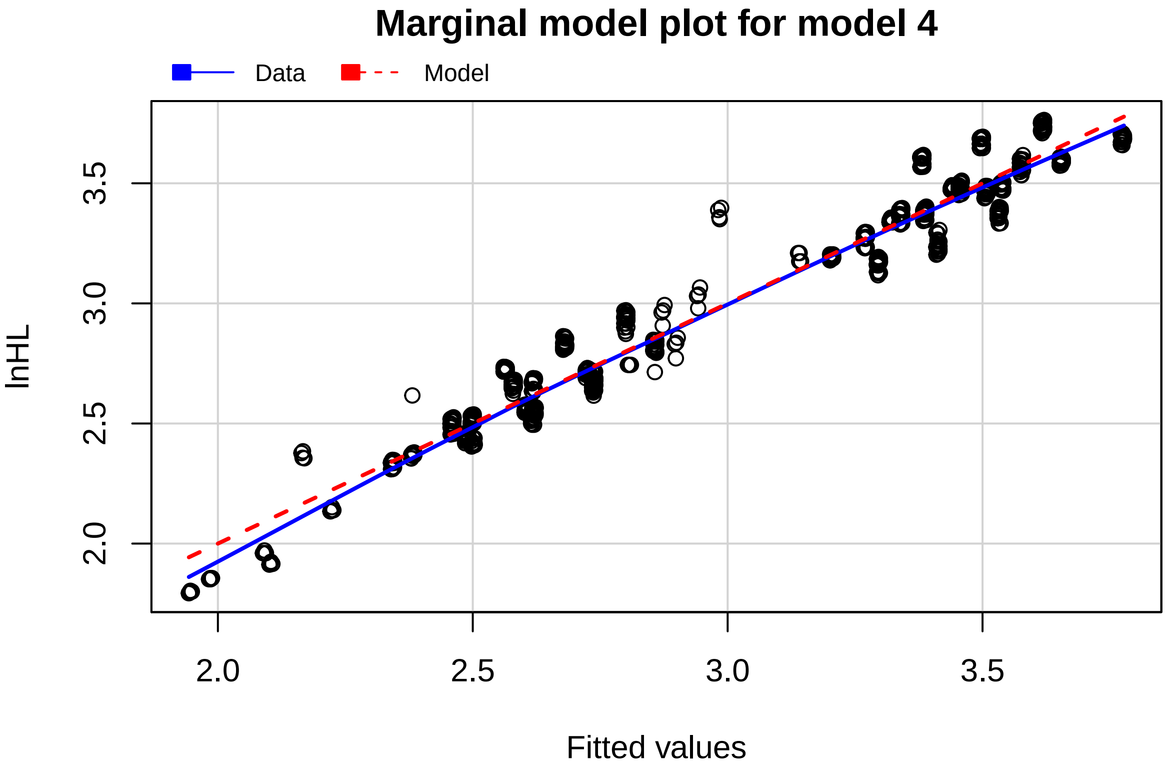
Statistical significance is preserved and no errors are present in the summary. A lower median is observed in addition to a higher adjusted R^2 in comparison to model 2. Proceeding with the diagnostics,





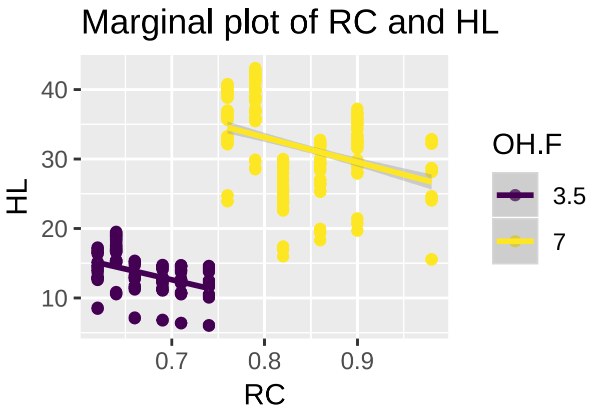
```
##                       GVIF Df GVIF^(1/(2*Df))
## lnRC          1832.926698  1       42.812693
## lnSA          1828.556406  1       42.761623
## lnWA             4.683857  1        2.164222
## energy$OH.F     13.723439  1        3.704516
## energy$GA.F      1.000000  3        1.000000
## energy$O.F       1.000000  3        1.000000
```

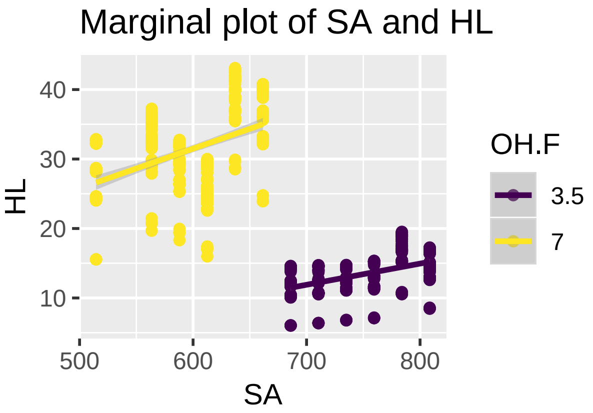
While the residuals and QQ plots look noticeably better, the VIF has inflated to an order of magnitude higher than the untransformed model. However, note that neither *lnRC* nor *lnSA* are considered insignificant, so a common consequence with high collinearity appears to have been avoided. Unfortunately, it does not appear that the transformation resolved the linearity issues. The marginal model plot is now considered:

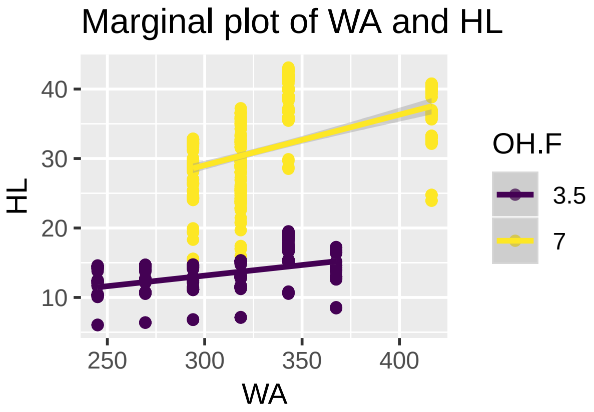


The model fits the data reasonably well; insofar as prediction goes, this model would be adequate for the task. From these results this model is tenatively selected as the most valid model thus far.

Recall from the violin plots that larger overall height corresponded to a much larger heating load. To check if there is a need to explore second order terms (e.g. interaction), marginal scatterplots are created and assessed.







From the marginal scatterplots it is evident that the levels of overall height makes a difference in heating load. However, note that the slopes for each level appears to be the same. Since the relationship between the variables modelled in the marginal plots do not differ for different levels of *OH.F*, it stands to reason that the inclusion of interaction terms would only marginally improve the model. **Model 4 is therefore chosen for continued analysis.**

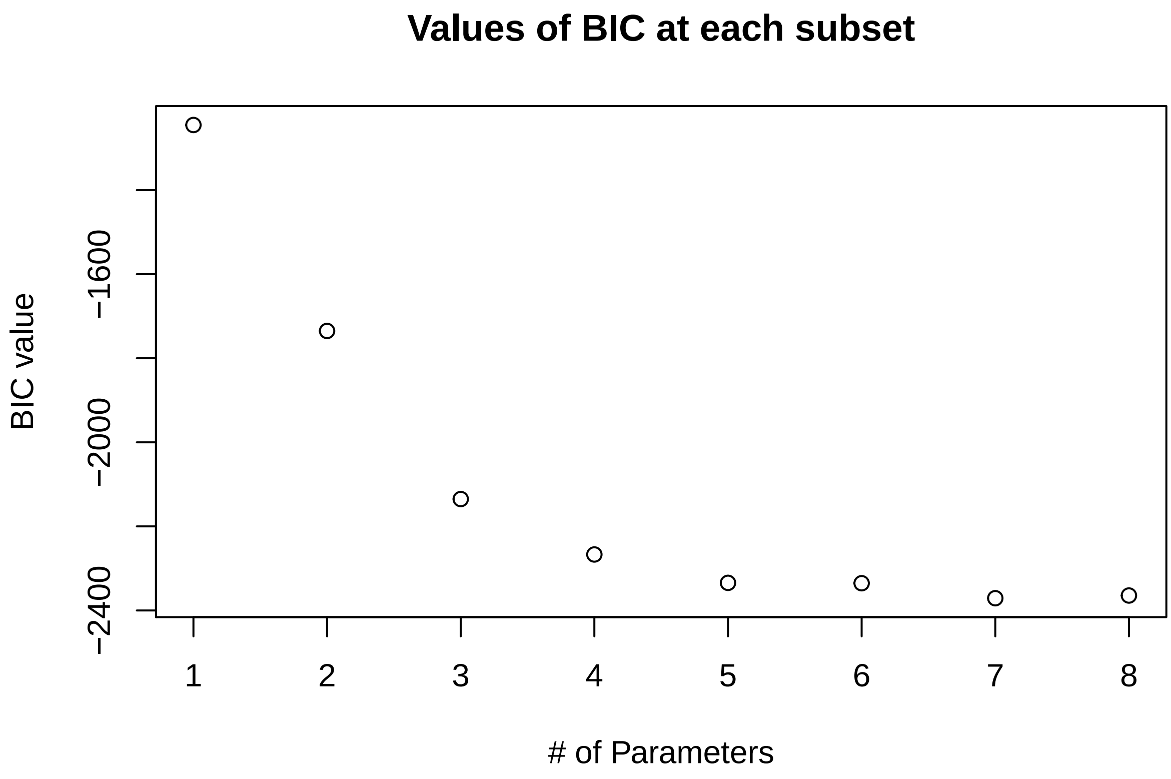
2.3 Variable Selection

It is natural to check if it's possible that a simpler model exists that would still have the same strength of interpretation as a more complicated one. As this dataset has perfect collinearity issues, a reduction in variables featured in a model would be ideal. Both all possible subsets and stepwise subsets are considered:

2.3.1 All possible subsets

A summary, plot of the BIC values and corresponding adjusted R^2 is produced for all possible subsets of model 4.

```
## Subset selection object
## Call: regsubsets.formula(lnHL ~ lnRC + lnSA + lnWA + OH.F + GA.F +
##     O.F, data = energy)
## 10 Variables  (and intercept)
##         Forced in Forced out
## lnRC        FALSE      FALSE
## lnSA        FALSE      FALSE
## lnWA        FALSE      FALSE
## OH.F.L      FALSE      FALSE
## GA.F.L      FALSE      FALSE
## GA.F.Q     FALSE      FALSE
## GA.F.C     FALSE      FALSE
## O.F3       FALSE      FALSE
## O.F4       FALSE      FALSE
## O.F5       FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          lnRC lnSA lnWA OH.F.L GA.F.L GA.F.Q GA.F.C O.F3 O.F4 O.F5
## 1  ( 1 ) " "  " "  " "  "*"    " "    " "    " "    " "  " "  " "
## 2  ( 1 ) " "  " "  " "  "*"    "*"    " "    " "    " "  " "  " "
## 3  ( 1 ) " "  " "  "*"  "*"    "*"    " "    " "    " "  " "  " "
## 4  ( 1 ) " "  " "  "*"  "*"    "*"    "*"    " "    " "  " "  " "
## 5  ( 1 ) " "  " "  "*"  "*"    "*"    "*"    "*"    " "  " "  " "
## 6  ( 1 ) " "  "*"  "*"  "*"    "*"    "*"    "*"    " "  " "  " "
## 7  ( 1 ) "*"  "*"  "*"  "*"    "*"    "*"    "*"    " "  " "  " "
## 8  ( 1 ) "*"  "*"  "*"  "*"    "*"    "*"    "*"    "*"  " "  " "
```



```
##      # of parameters adjusted R squared
## [1,]               1          0.8055054
## [2,]               2          0.8979740
## [3,]               3          0.9398328
## [4,]               4          0.9496918
## [5,]               5          0.9542594
## [6,]               6          0.9546484
## [7,]               7          0.9570215
## [8,]               8          0.9569790
```

The subset containing 7 variables is the best model, as it has the lowest BIC and highest adjusted R^2. To compare, the stepwise method is used:

2.3.2 Stepwise subsets

```
backAIC = step(model.4, direction = "backward")
```

```
## Start:  AIC=-3541.52
## lnHL ~ lnRC + lnSA + lnWA + energy$OH.F + energy$GA.F + energy$O.F
##
##                Df Sum of Sq     RSS     AIC
## - energy$O.F    3    0.0034  7.4205 -3547.2
## <none>                       7.4171 -3541.5
## - lnRC          1    0.4200  7.8371 -3501.2
## - lnSA          1    0.4479  7.8650 -3498.5
## - lnWA          1    1.8118  9.2289 -3375.7
## - energy$OH.F   1   10.4752 17.8923 -2867.2
## - energy$GA.F   3   18.6392 26.0563 -2582.6
##
## Step:  AIC=-3547.17
## lnHL ~ lnRC + lnSA + lnWA + energy$OH.F + energy$GA.F
##
##                Df Sum of Sq     RSS     AIC
## <none>                       7.4205 -3547.2
## - lnRC          1    0.4200  7.8405 -3506.9
## - lnSA          1    0.4479  7.8684 -3504.2
## - lnWA          1    1.8118  9.2323 -3381.4
## - energy$OH.F   1   10.4752 17.8957 -2873.1
## - energy$GA.F   3   18.6392 26.0597 -2588.4
```

```
backBIC = step(model.4, direction = "backward", k = log(nrow(energy)))
```

```
## Start:  AIC=-3490.44
## lnHL ~ lnRC + lnSA + lnWA + energy$OH.F + energy$GA.F + energy$O.F
##
##                Df Sum of Sq     RSS     AIC
## - energy$O.F    3    0.0034  7.4205 -3510.0
## <none>                       7.4171 -3490.4
## - lnRC          1    0.4200  7.8371 -3454.8
## - lnSA          1    0.4479  7.8650 -3452.1
## - lnWA          1    1.8118  9.2289 -3329.2
## - energy$OH.F   1   10.4752 17.8923 -2820.8
## - energy$GA.F   3   18.6392 26.0563 -2545.4
##
## Step:  AIC=-3510.02
## lnHL ~ lnRC + lnSA + lnWA + energy$OH.F + energy$GA.F
##
##                Df Sum of Sq     RSS     AIC
## <none>                       7.4205 -3510.0
## - lnRC          1    0.4200  7.8405 -3474.4
## - lnSA          1    0.4479  7.8684 -3471.7
## - lnWA          1    1.8118  9.2323 -3348.9
## - energy$OH.F   1   10.4752 17.8957 -2840.6
## - energy$GA.F   3   18.6392 26.0597 -2565.2
```

All methods agree on this new model being the best. Interestingly, despite the high collinearity present in the model, no reduction is advised. The only recommendation was to drop orientation, which was not significant in the first place. Nevertheless, the updated model is now

```
model.5 = lm(lnHL ~ lnRC + lnSA + lnWA + OH.F + GA.F, data = energy)
```

2.4 Model Prediction

To assess the predictive power of the model, the resampling method is used. Since the model involved transformations, it is necessary to use transformed data.

```
logEnergy = as_tibble(cbind(lnHL, lnRC, lnSA, lnWA))
logEnergy = logEnergy %>% mutate(energy$OH.F, energy$GA.F)
names(logEnergy) = c("lnHL", "lnRC", "lnSA", "lnWA", "OH.F", "GA.F")
logEnergy
```

```
## # A tibble: 768 x 6
##     lnHL    lnRC  lnSA  lnWA OH.F  GA.F
##    <dbl>   <dbl> <dbl> <dbl> <ord> <ord>
##  1  2.74 -0.0202  6.24  5.68 7     0
##  2  2.74 -0.0202  6.24  5.68 7     0
##  3  2.74 -0.0202  6.24  5.68 7     0
##  4  2.74 -0.0202  6.24  5.68 7     0
##  5  3.04 -0.105   6.33  5.76 7     0
##  6  3.07 -0.105   6.33  5.76 7     0
##  7  3.03 -0.105   6.33  5.76 7     0
##  8  2.98 -0.105   6.33  5.76 7     0
##  9  2.97 -0.151   6.38  5.68 7     0
## 10  2.99 -0.151   6.38  5.68 7     0
## # ... with 758 more rows
```

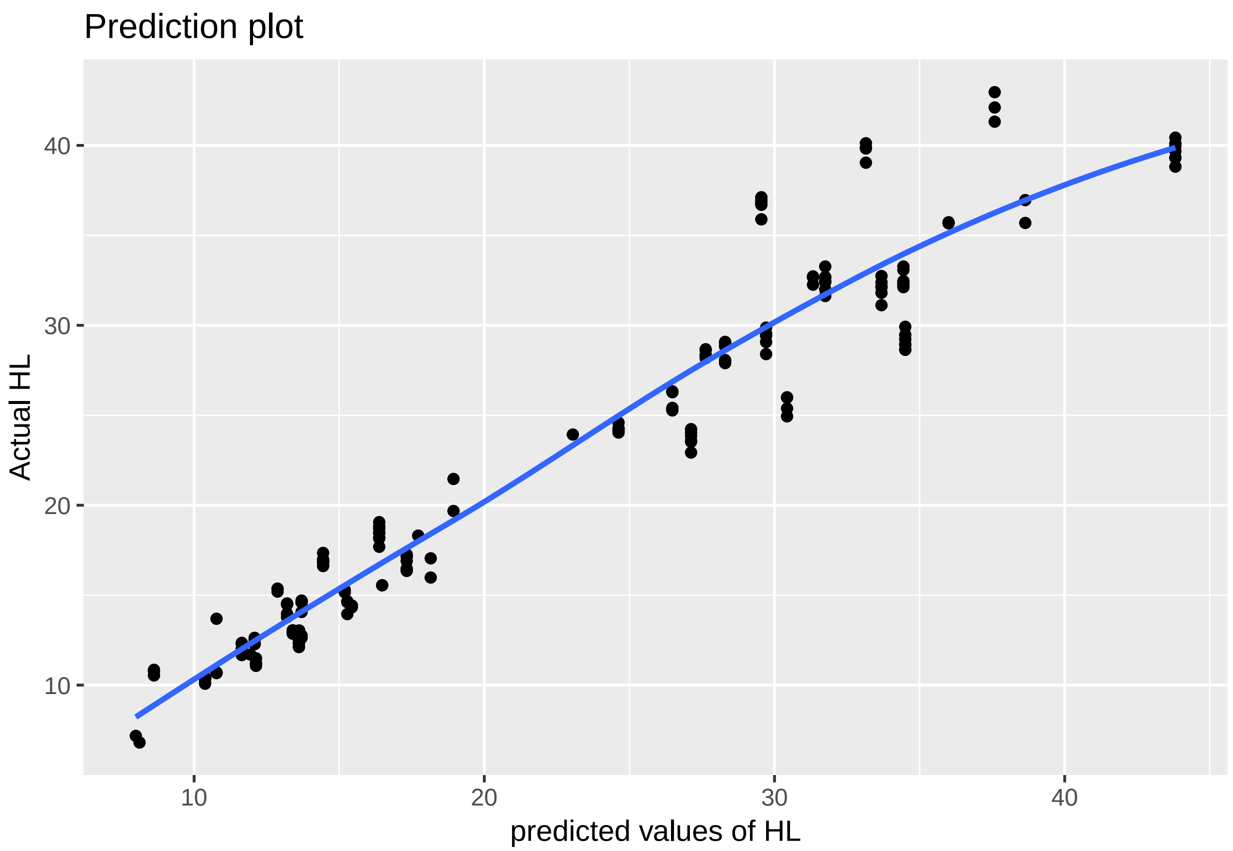
2.4.1 Resampling

The dataset is split into a testing set and training set. The proportions chosen are 0.75 and 0.25 for the training set and the testing set, respectively.

```
set.seed(0)
trainingIndex = sample(1:nrow(logEnergy), 0.75*nrow(logEnergy))
train = logEnergy[trainingIndex,]
test = logEnergy[-trainingIndex,]
trainedModel.5 = lm(lnHL ~ lnRC + lnSA + lnWA + OH.F + GA.F, data = train)
predictModel.5 = predict(trainedModel.5, test)
```

The prediction plot is then visualised:

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The plot seems to fit fairly well. Checking the correlation and MSE:

```
cor(actualPrediction)
```

```
##                 predictModel.5      lnHL
## predictModel.5      1.0000000 0.9746233
## lnHL                0.9746233 1.0000000
```

```
mean((predictModel.5-test$lnHL)^2)
```

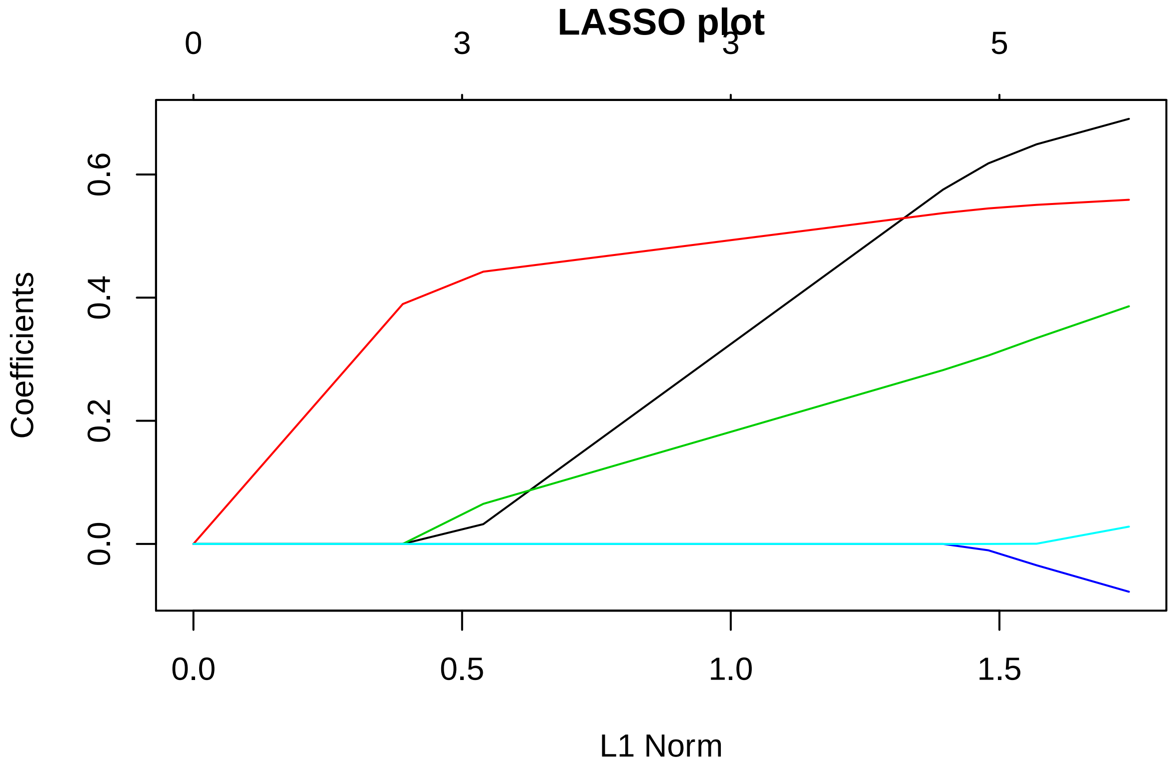
```
## [1] 0.01041878
```

With high correlation and minimal MSE, model 5 has exceptional predictive power.

2.5 LASSO

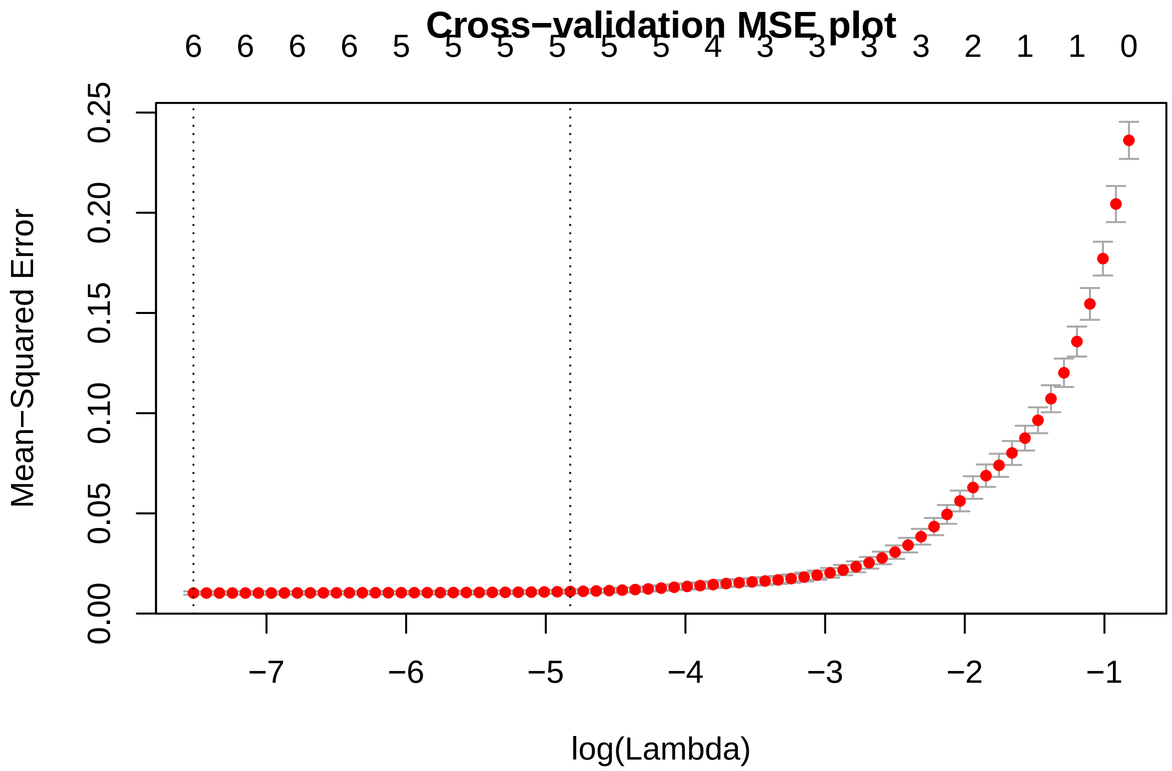
The LASSO method is conducted as an attempt to create a better model. To employ the LASSO method, a model matrix is created, with the intercept removed.

```
set.seed(0)
x = model.matrix(lnHL ~ ., data = logEnergy)[,-1]
y = logEnergy$lnHL
lambda = 10^seq(10,-2,length = 100)
lasso.mod = glmnet(x[trainingIndex, ],y[trainingIndex],alpha = 1, lambda = lambda, family = "gaussian")
plot(lasso.mod, main = "LASSO plot")
```



The LASSO plot shows complete divergence at 5 variables, which suggests that all 5 variables are important. Performing cross validation, the obtained plot and values are:

```
cv.out = cv.glmnet(x[trainingIndex,],y[trainingIndex], alpha = 1)
plot(cv.out, main = "Cross-validation MSE plot")
```



```
bestLambda = cv.out$lambda.min
lambdaSE = cv.out$lambda.1se
bestLambda
```

```
## [1] 0.0005405139
```

```
lambdaSE
```

```
## [1] 0.008026459
```

```
mean((predict(lasso.mod, s = bestLambda, newx = x[-trainingIndex,])-y[-trainingIndex])^2)
```

```
## [1] 0.01136974
```

```
mean((predict(lasso.mod, s = lambdaSE, newx = x[-trainingIndex,])-y[-trainingIndex])^2)
```

```
## [1] 0.01136974
```

```
mean((predict(lasso.mod, s = 0, newx = x[-trainingIndex,])-y[-trainingIndex])^2)
```

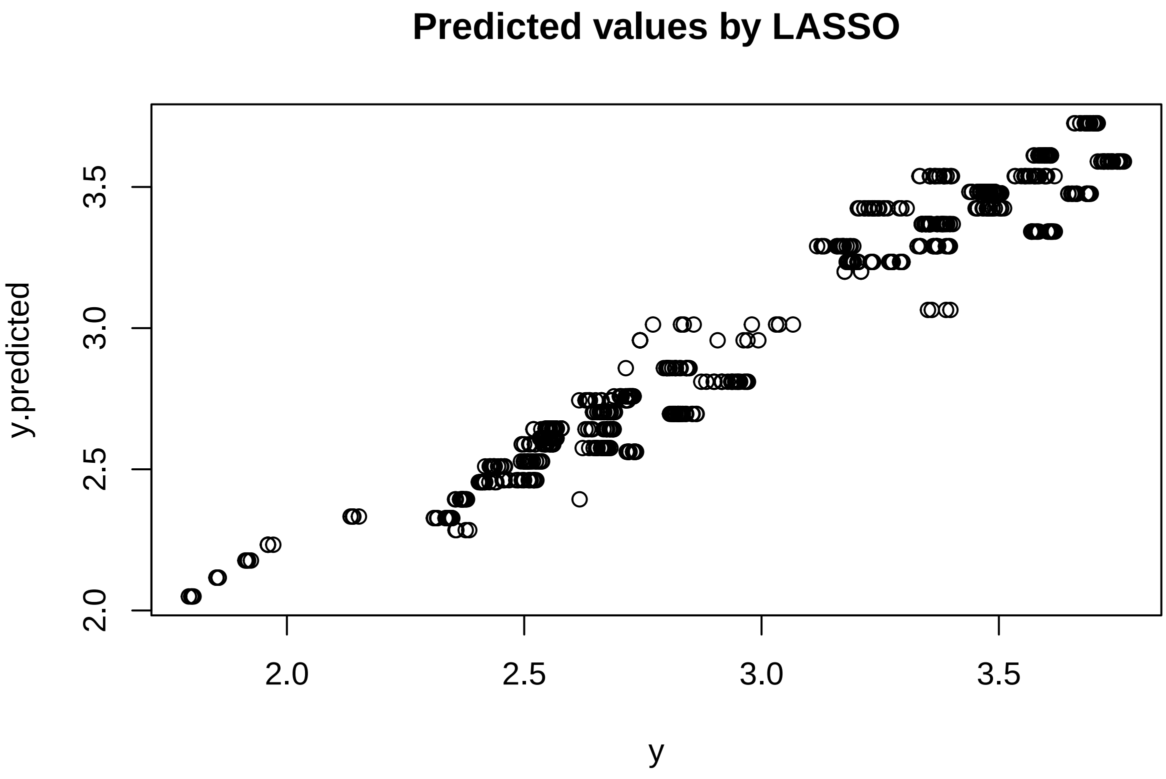
```
## [1] 0.01136974
```

Interestingly, all λ values produce the same mean. Since all λ values are close to 0, the ordinary least squares approach is still the most ideal for carrying out the analysis. For completion's sake, the prediction plot is generated:

```
out = glmnet(x,y, alpha = 1, lambda = lambda, family = "gaussian")
lasso.coeff = predict(out, type = "coefficients", s = 0)
lasso.coeff
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept) -1.09649744
## lnRC         .
## lnSA         .
## lnWA         0.69799008
## OH.F.L       0.55149352
## GA.F.L       0.38257784
## GA.F.Q      -0.08170683
## GA.F.C       0.02750815
```

```
y.predicted = predict(out, s = 0, newx = x)
plot(y,y.predicted, main = "Predicted values by LASSO")
```



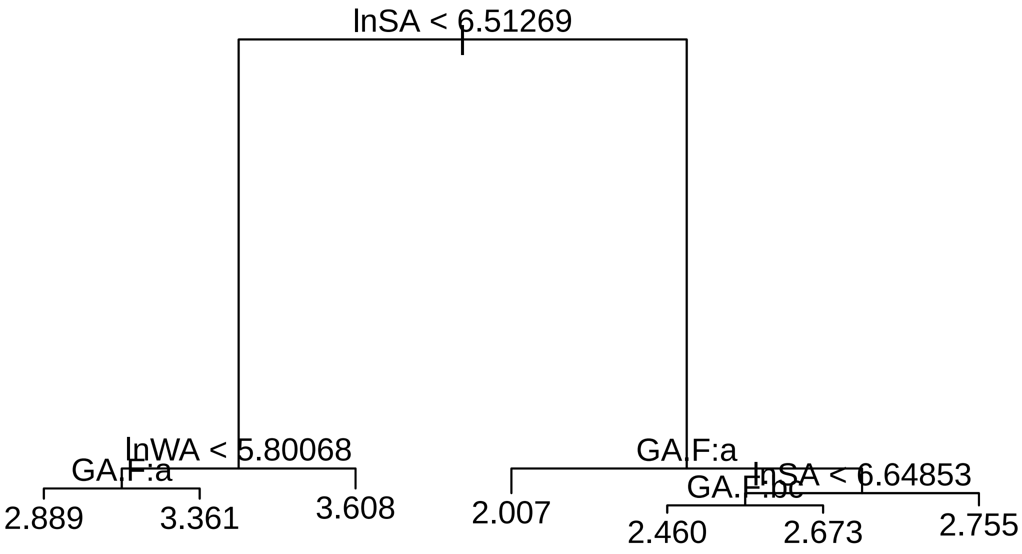
The model created by LASSO also seems to have reasonable predictive power.

2.6 Regression Tree

A regression tree is created to generate another model. In the context of this dataset, since the predictors are not linearly distributed, the regression tree may handle the non-linearity better than the other methods.

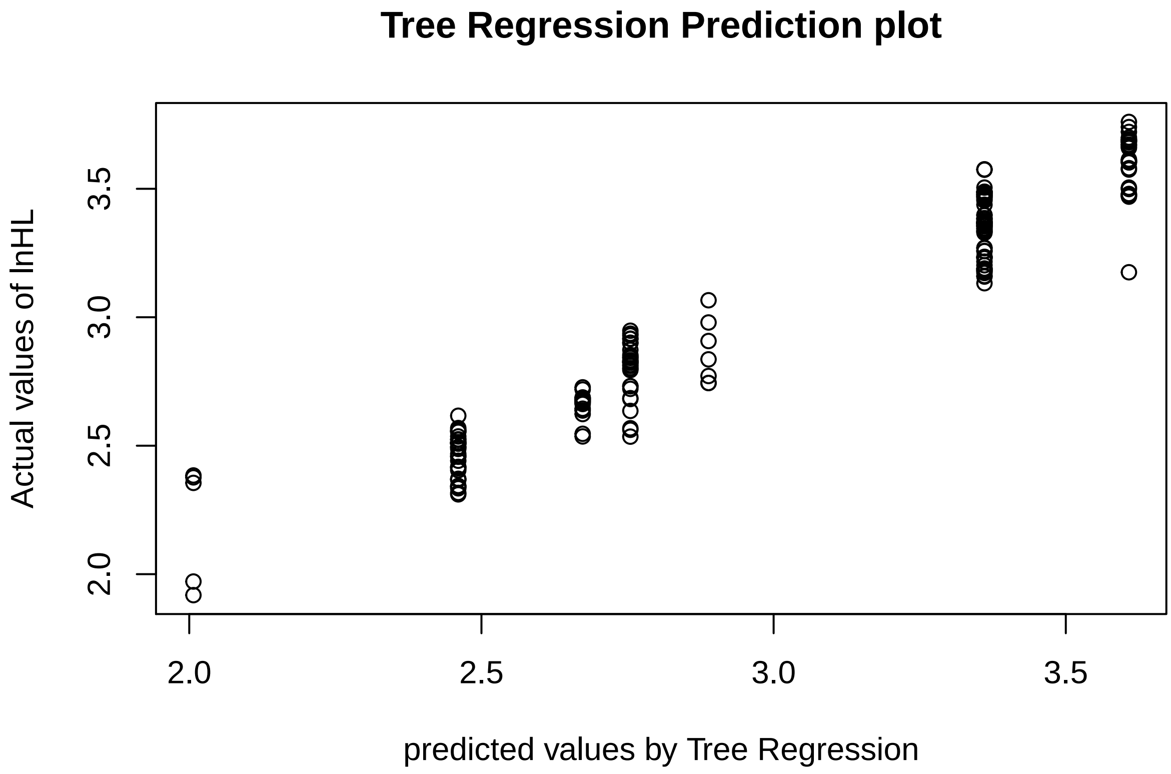
```
##
## Regression tree:
## tree(formula = lnHL ~ ., data = logEnergy)
## Variables actually used in tree construction:
## [1] "lnSA" "lnWA" "GA.F"
## Number of terminal nodes:  7
## Residual mean deviance:  0.01247 = 9.488 / 761
## Distribution of residuals:
```

```
##       Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## -0.432900 -0.073300  0.001628  0.000000  0.080300  0.377000
```



Note that since the data values are logged, the interpretation for each value would have to be exponentiated. Additionally, the model yields some surprising and non-intuitive results. For example, the regression tree classifies that, for values of HL which are $\sim e^{2.007}$, SA is over $e^{6.51269}$ and GA to be factor 0. In other words, a larger surface area corresponds to lower heating loads, contradicting analysis conducted earlier. A prediction plot is generated from this model to ascertain the predictive validity.

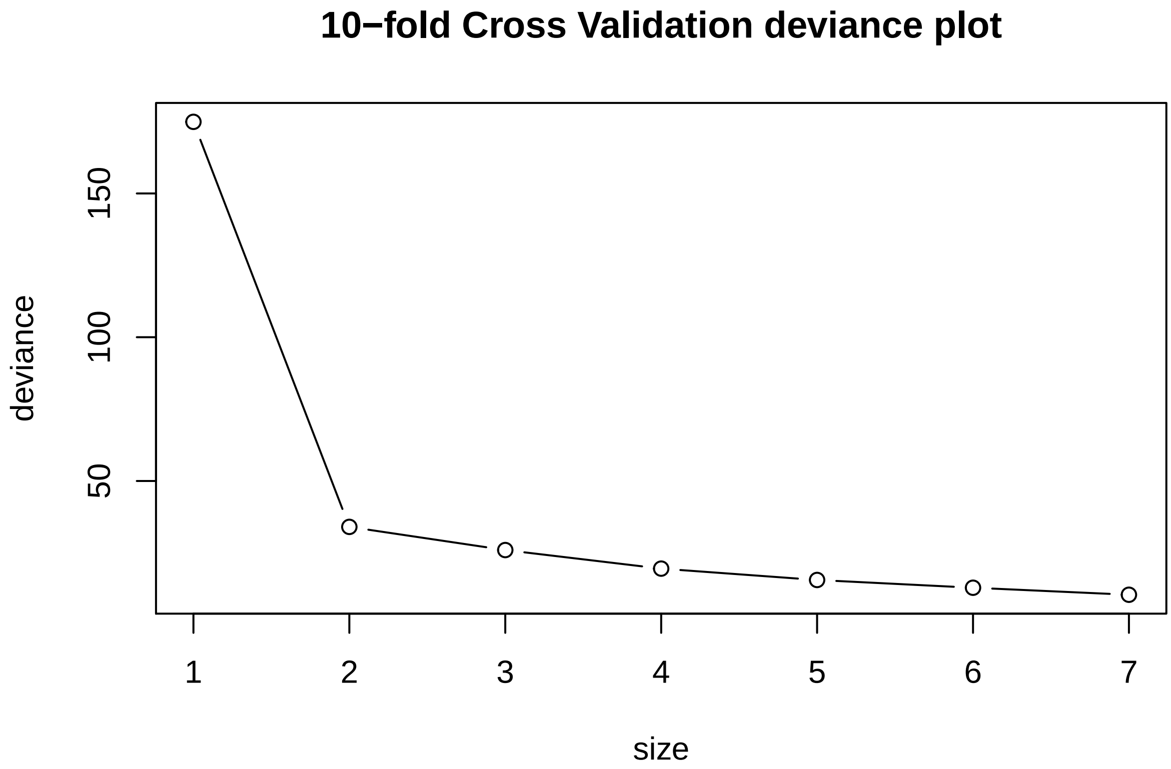
```
tree.predict = predict(tree.data, newdata = logEnergy[-trainingIndex,])
tree.test = as.matrix(logEnergy[-trainingIndex,"lnHL"])
plot(tree.predict, tree.test, main = "Tree Regression Prediction plot", xlab = "predicted values by Tre
```



```
tree.mse = mean((tree.predict-tree.test)^2)
tree.mse
```

```
## [1] 0.013054
```

The regression tree prediction plot and MSE is similar to previous analyses. Regardless, this model should be treated with deep skepticism - it is clear that larger surface area is associated with larger heating loads, and the logarithm function preserve monotonicity, so these results are nonsensical. Cross validation is plotted to check deviance, and the regression tree will undergo pruning to see if inference can be simplified.

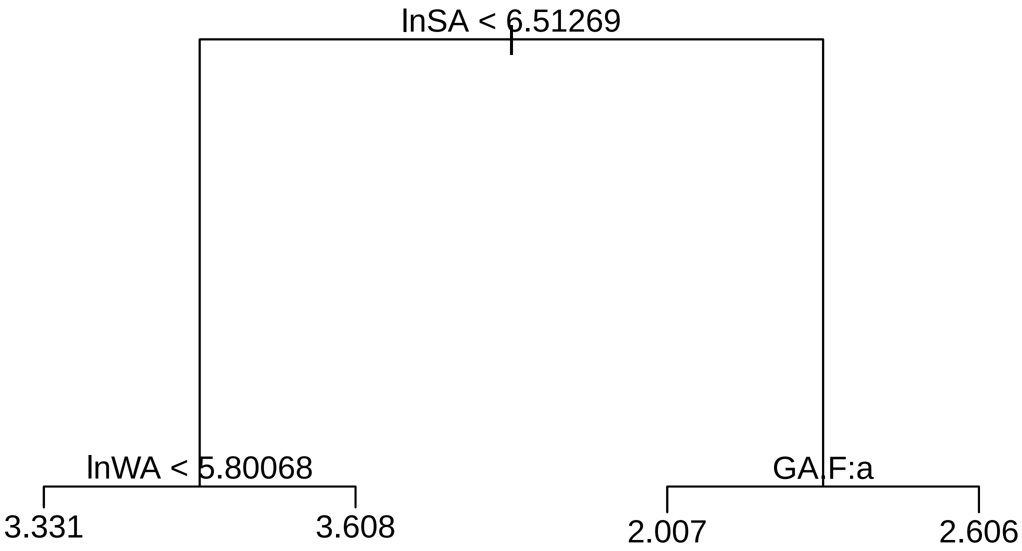


The deviance sharply decreases with the inclusion of 2 predictors and tapers off with the inclusion of each additional predictor. Since the deviance reduction is marginal, the model is pruned at size equal to 4. Pruning the tree,

```
tree.prune = prune.tree(tree.data, best = 4)
summary(tree.prune)
```

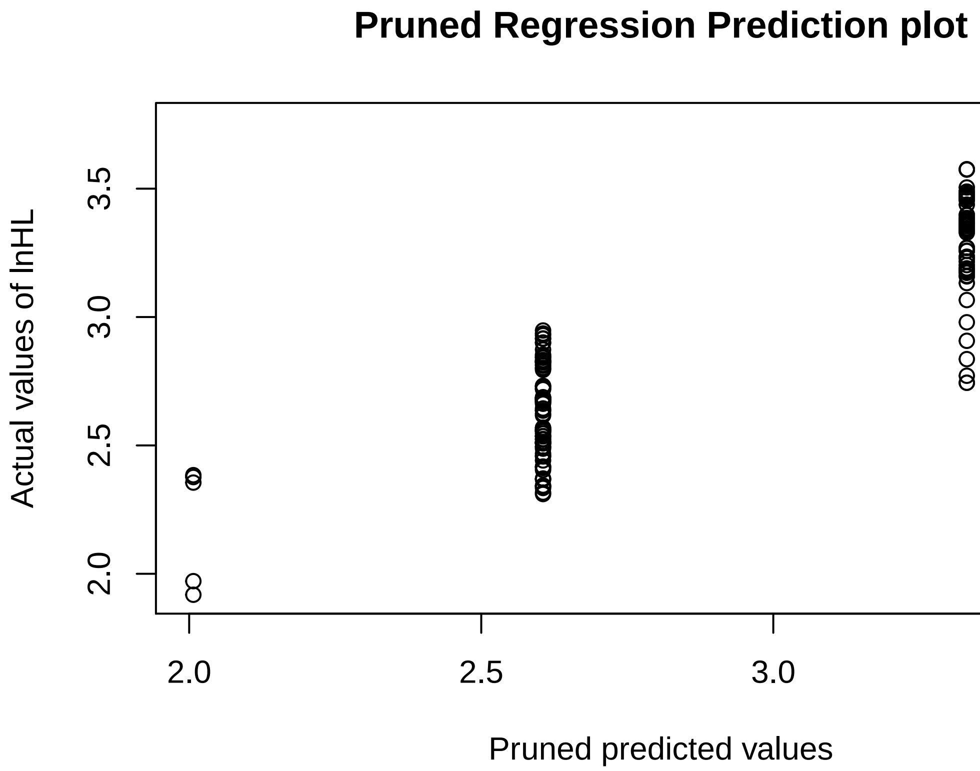
```
##
## Regression tree:
## snip.tree(tree = tree.data, nodes = c(4L, 7L))
## Variables actually used in tree construction:
## [1] "lnSA" "lnWA" "GA.F"
## Number of terminal nodes:  4
## Residual mean deviance:  0.02519 = 19.25 / 764
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.5872 -0.1043  0.0166  0.0000  0.1128  0.3770
```

```
plot(tree.prune)
text(tree.prune)
```



A simpler regression tree plot is produce. Evaluating the prediction plot,

```
predict.prune = predict(tree.prune, newdata = logEnergy[-trainingIndex,])
plot(predict.prune, tree.test, main = "Pruned Regression Prediction plot", xlab = "Pruned predicted valu
```



```
mean((predict.prune - tree.test)^2)
```

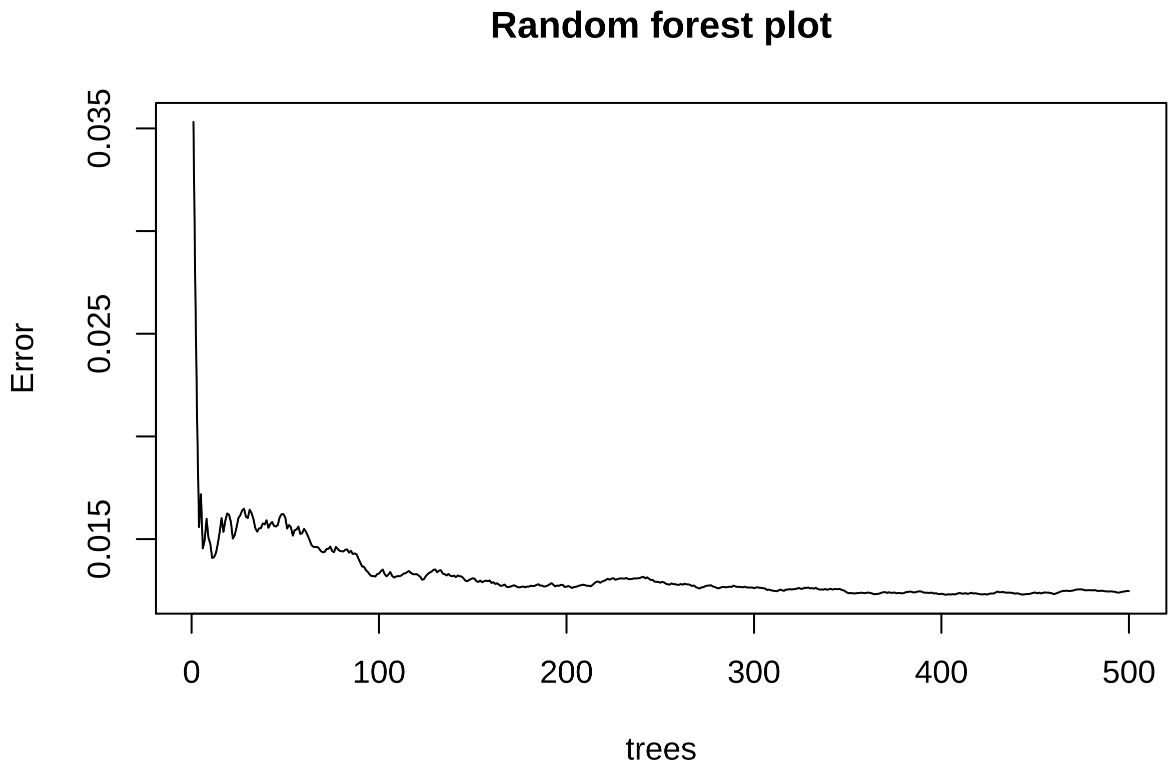
```
## [1] 0.03108109
```

The pruned regression tree appears to have produced worse predictions and has a higher MSE than the original tree regression. In light of the already contentious findings from the original regression tree model, the pruned model is not particularly reliable or outstanding.

2.7 Random Forest

A random forest is a method introduced to alleviate some of the problems that regression trees run into. Regression trees are highly sensitive to small changes in the data. Since the regression tree requires the dataset to be partitioned into a training and testing set, a different training set would yield vastly different results. A random forest is designed by sampling from the data set repeatedly (this is known as 'bootstrapping') to create a unique regression tree from each sampling, and then averaging the results. This corrects the overfitting tendency from regression trees while improving predictive power, at the cost of some bias and interpretation. As the number of trees increases, the lesser the error the random forest has. As observed,

```
rf = randomForest(lnHL ~ ., data = logEnergy, importance = T)
plot(rf, main = "Random forest plot")
```



```
rf
```

```
##
## Call:
##  randomForest(formula = lnHL ~ ., data = logEnergy, importance = T)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 0.01246527
##                     % Var explained: 94.51
```

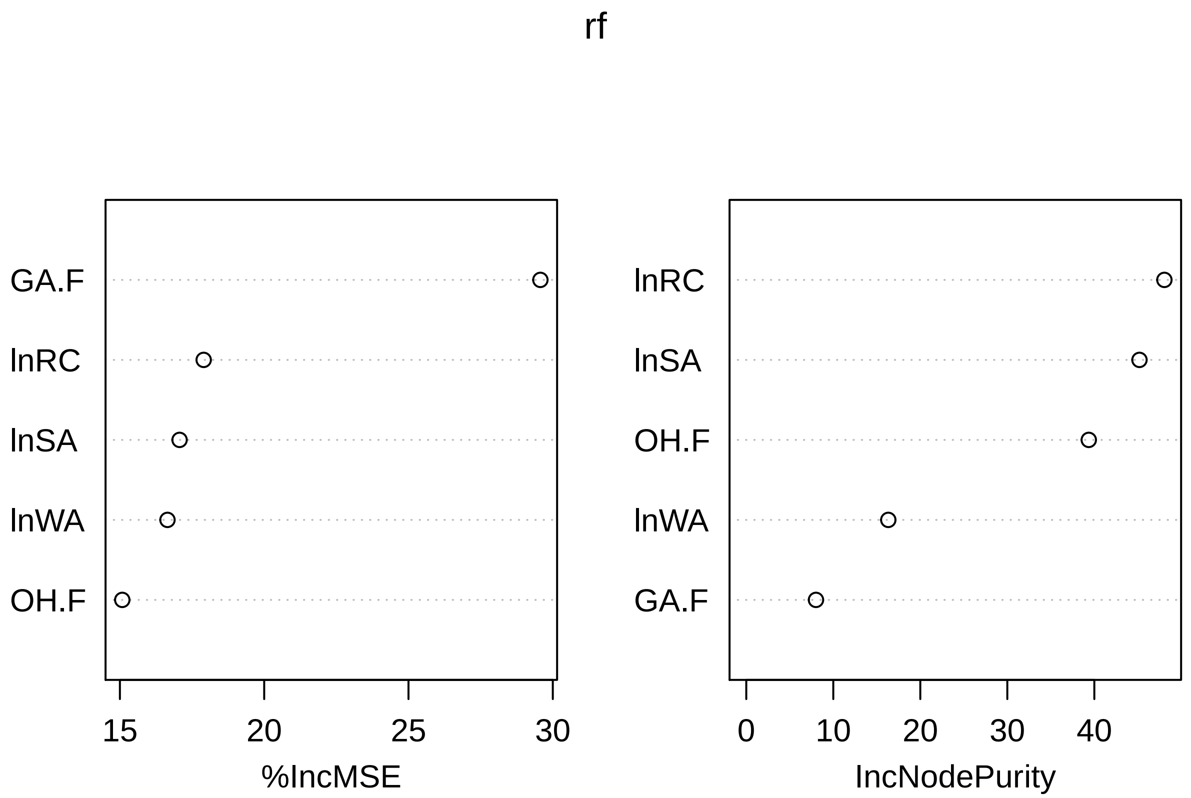
A comparison in predictive power between the random forest and regression tree is found:

```
set.seed(0)
rf.predict = randomForest(lnHL ~ ., data = train)
rf.mse = mean((predict(rf.predict, test)-tree.test)^2)
c(rf = rf.mse, tree = tree.mse)
```

```
##        rf      tree
## 0.0114497 0.0130540
```

The random forest marginally beats the regression tree in predictive power.

Variable importance (in the context of model prediction) can also be described through random forests.



The random forest plot ranks *lnSA* and *lnRC*' to be the most important variables, while *GA.F* ranks among the least. It is still meaningful to note that high collinearity may have skewed the analysis. Since the random forest is difficult to interpret, no additional attempt has been made.

3 Results

The model that seems most adequate to address quantifiable statements with regards to heating load is model 5:

```
summary(model.5)
```

```
##
## Call:
## lm(formula = lnHL ~ lnRC + lnSA + lnWA + OH.F + GA.F, data = energy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20964 -0.06220 -0.01125  0.04782  0.41365
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -49.648500   7.202571  -6.893 1.15e-11 ***
## lnRC          7.375110   1.124438   6.559 1.00e-10 ***
## lnSA          7.709043   1.138197   6.773 2.53e-11 ***
## lnWA          0.779225   0.057202  13.622  < 2e-16 ***
## OH.F.L        0.611852   0.018680  32.754  < 2e-16 ***
## GA.F.L        0.451019   0.010673  42.258  < 2e-16 ***
## GA.F.Q       -0.138313   0.009020 -15.334  < 2e-16 ***
## GA.F.C        0.063330   0.006987   9.064  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09881 on 760 degrees of freedom
## Multiple R-squared:  0.9574, Adjusted R-squared:  0.957
## F-statistic:  2441 on 7 and 760 DF,  p-value: < 2.2e-16
```

The interpretation of each coefficient are as follows: On average, and holding all other predictors constant, per doubling of RC, SA, and WA, corresponds to a $2^{7.375}$, $2^{7.709}$, and $2^{0.779}$ times increase in HL, respectively. The ordinal factors are somewhat more difficult to interpret, but an attempt is given. The average change in HL as overall height goes from 3 to 7.5 is $e^{0.612}$, holding all predictors constant. For glazing area, at the baseline of 0%, the average change in HL as glazing area increases to 10%, 25%, and 40% is $e^{0.451}$, $e^{-0.138}$, and $e^{0.063}$, respectively. The reported adjusted R squared value is 0.957, meaning 95.7% of the variation in $lnHL$ can be explained in this model. A low median value is reported.

Variable selection and model prediction were investigated to improve upon and test this model. In this respect, removing the non-significant variable $O.F$ was recommended and carried out. LASSO, regression tree, and random forests were introduced to identify possible different models. Prediction plots for all methods (with the exception of the pruned regression tree) produced very similar results. While the LASSO corroborates with the original model as being the best (since $\lambda = 0$), the regression tree provides an intriguing but useless alternative. The random forest identifies $lnSA$ and $lnRC$ to be the most important variables and $GA.F$ to be the least.

4 Discussion

4.1 Model Interpretation

In modelling the variables that affect heating load, an individual relationship is found for each predictor. It can be observed that the orientation of the building and where the glaze area is distributed has no bearing on the heating load of the building. Relative compactness, surface area, wall area, and overall height all were positively correlated with larger heating load; this makes conceptual sense, as a larger building would naturally require additional energy to maintain acceptable temperature. Roof area was not considered significant in the model, but unlike orientation or glazing area distribution it can be perfectly computed with a combination of the previous variables mentioned. That means the non-significance is due to its perfect collinearity with some of the predictors, not that it had no relationship with heating load in of itself. Glazing area has the most surprising result - the relationship between the amount of glaze and heating load does not seem static. Compared at the 0% baseline, while the inclusion of glazing area increases heating load at 10% and 40%, it actually decreases at 20%, since $e^{-0.138} < 1$. This discrepancy is not so easily explainable; the sample size is reasonably large and the dataset was controlled from other outside factors. Additionally, as the dataset was simulated, the proportion of samples having different factors are the same. It could be possible that the relationship between glazing area and heating loads really do change depending on the amount of glaze, but that seems unlikely and would require deeper analysis.

It seems natural to compare these results with the paper (Xifara and Tsanas, 2012) that this report is based on to ascertain the validity of the findings. Unfortunately, few properties are identical; outside of what was considered significant and the signs of the correlation, the actual quantitative estimates are all different. However, it is important to mention that the methodology carried out in the paper are in many ways different from the methods conducted here. The linear regression in the paper was applied through a method known as 'Iteratively reweighted least squares', and the classification random forest actually generated coefficients for each predictor. Furthermore, the inferential scope of this project is to establish the relationships of each variable with respect to heating load, which has been modestly accomplished.

A final point to discuss is how this ties into real life energy performance as a whole. The coefficient of determination and MSE all point to very promising results, but these results are after controlling for all other possible variables that could affect energy efficiency in simulated data. While it is still relatively clear that there is an association, the actual difference that physical properties makes may be over pronounced and would be lost in the noise of real world data. This is not entirely implausible, as col-linearity issues persist and mask the true estimated effect of each variable.

4.2 Limitations

George Box was recorded in saying that "all models are wrong, but some are useful". His profound wisdom will be apparent in this section.

4.2.1 Basic Assumptions

The four assumptions that linear regression hinges on were all violated with varying degrees of severity (log transformation did resolve heteroskedasticity and non-normal errors). The most pressing concerns are that the continuous predictors (relative compactness, surface area, wall area, roof area) are not linearly distributed and the staggeringly high vif values indicating serious multicollinearity. These problems could not be remedied with any of the possible solutions, which perhaps illustrate the greater point that linear regression was not suitable as a method for this data set in the first place. Regardless, the models examined all have substantial predictive power, so although it would be fair criticism to claim that the true relationships between the predictors and heating load have not been found, there are still some uses obtained in trying.

4.2.2 Mis-classified variable significance

The random forest generated in this project reported the most significant variables as surface area followed closely by relative compactness. These results does not match the results of the random forest obtained by Xifara and Tsanas. They report the most significant variable to be glazing area, for which they point out agrees with the existing literature regarding energy performance of buildings. In fact, it can be considered one of the primary goals of their work is to illustrate the advantages of non-linear methods, particularly in the case of multicollinear non-linear data. This makes the discrepancy of the random forest concerning, but since the scope is focused primarily on linear models, an ad-hoc solution did not feel necessary.

4.3 Improvements

Xifara and Tsanas(2012) recommend machine learning methods to handle this data set. This would drastically improve results, particularly since machine learning methods do not require the linearity assumption to be upheld. A very limited machine learning approach was actually implemented in this report; namely, the random forest. The random forest only made contributions to determining variable significance and as a proof-of-concept, since the technical depth required for full analysis was outside of the scope of this project.

Another possible direction (without involving machine learning) is multilevel regression. Multilevel (or hierarchical) regression involves modelling of parameters at more than one level. This would allow inference on variations between groups, rather than on a per-individual basis. Gelman and Hill (2006) recommends this approach for hierarchical resembling data. An example of how this would be implemented is by specifying the overall height as the group of interest, and proceeding with the multilevel regression. Due to the technical sophistication (this modelling mostly requires Bayesian methods) and the limited scope of the project, it was not chosen to be done, despite being perhaps the most logical approach.

5 Conclusion

In modelling the factors that affect energy efficiency, it is found that relative compactness, surface area, wall area, and overall height all correspond to higher heating loads. Orientation, glazing area distribution, and roof area are not found to be significant, and glazing area has mixed results. While not complete in satisfying the assumptions of a linear model, the model predictions are promising and model selection was cross examined in a variety of ways. Some shortcomings are discussed and model improvements are considered via machine learning and multilevel regression. For practical uses, builders and prospective tenants should prioritise smaller sizes of the building dimensions for a more energy efficient design.

6 Appendix

Below are the packages used for this project:

```r
library(tidyverse)
library(car)
library(leaps)
library(glmnet)
library(tree)
library(randomForest)
library(reshape2)
library(ggfortify)
library(GGally)
library(gridExtra)
```

As this was written using R markdown, the actual R code involved in creating all results are scaffolded in the .rmd file.

References

Geletka, Vladimir, and Anna Sedláková. 2012. "Shape of Buildings and Energy Consumption." *Czasopismo Techniczne. Budownictwo* 109.

Gelman, Andrew, and Jennifer Hill. 2006. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge university press.

"Glossary of Energy Terms." n.d. https://definedterm.com/a/definition/264536.

Tsanas, Athanasios, and Angeliki Xifara. 2012. "Accurate Quantitative Estimation of Energy Performance of Residential Buildings Using Statistical Machine Learning Tools." *Energy and Buildings* 49. Elsevier: 560–67.