Statistical Analysis on Real Estate Valuation Dataset

STAT444 Final Project

August 10, 2019

Prepared by:

Shuby Sharma 20680025

Congxiao Jin 20608285

Ruiqi Wang 20566688

STAT 444 Final Project - Real Estate Valuation Dataset

Shubham, Jin, Ruiqi

7/21/2019

Motivation and introduction of the problem:

House prices have always been a hot topic, and while the global housing markets have been steadily climbing up, we want to have an idea on what is affecting the housing price the most. In this project we chose to specifically focus on the Taiwan housing market, we studied and analayzed the real estate valuation data set, which consists of 6 key factors on the housing price in New Taipei City, by fitting three regularization models, as well as using smoothing spline, random forest and graident boosting to help guide us determing which factors play the most important role in Taiwan house prices, and later on to discuss if the conclusion we reach could be generalized and apply to a bigger region.

Data:

We used a data set from the UCI Machine Learning Repository, it consists of 6 explanatory variables, and one response variable. The market historical data set of real estate valuation were collected from Sindian Dist., New Taipei City, Taiwan. The areal estate valuations is a regression problem.

Attribute Information:

The inputs are as follows:

X1 - the transaction date (for example, 2013.250=2013 March, 2013.500=2013 June, etc.)

X2 - the house age (unit: year)

X3 - the distance to the nearest MRT station (unit: meter)

X4 - the number of convenience stores in the living circle on foot (integer)

X5 - the geographic coordinate, latitude. (unit: degree)

X6 - the geographic coordinate, longitude. (unit: degree)

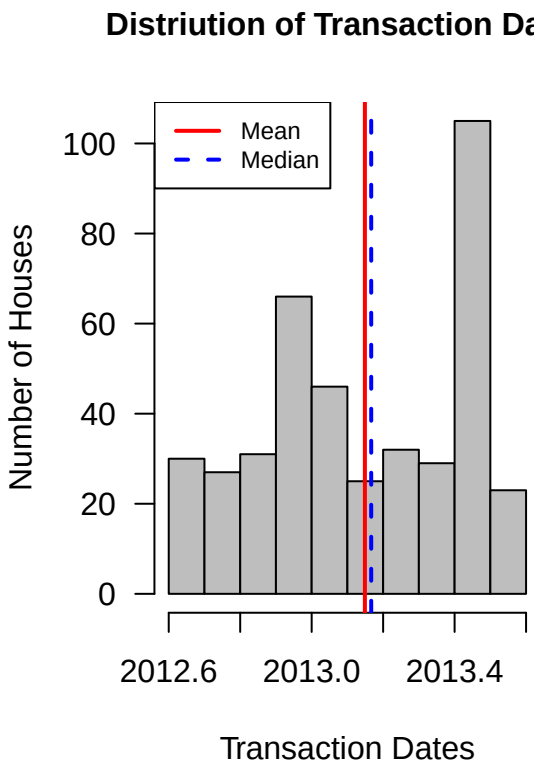
The output is as follow:

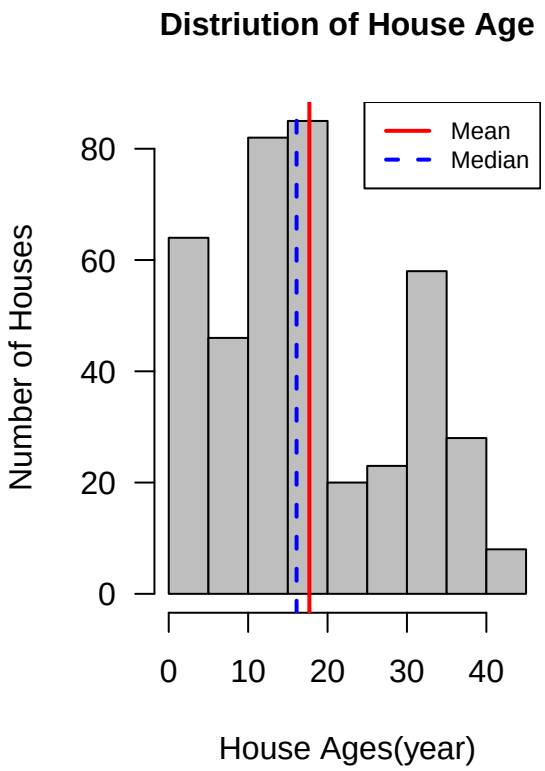
Y - house price of unit area (10000 New Taiwan Dollar/Ping, where Ping is a local unit, 1 Ping = 3.3 meter squared)

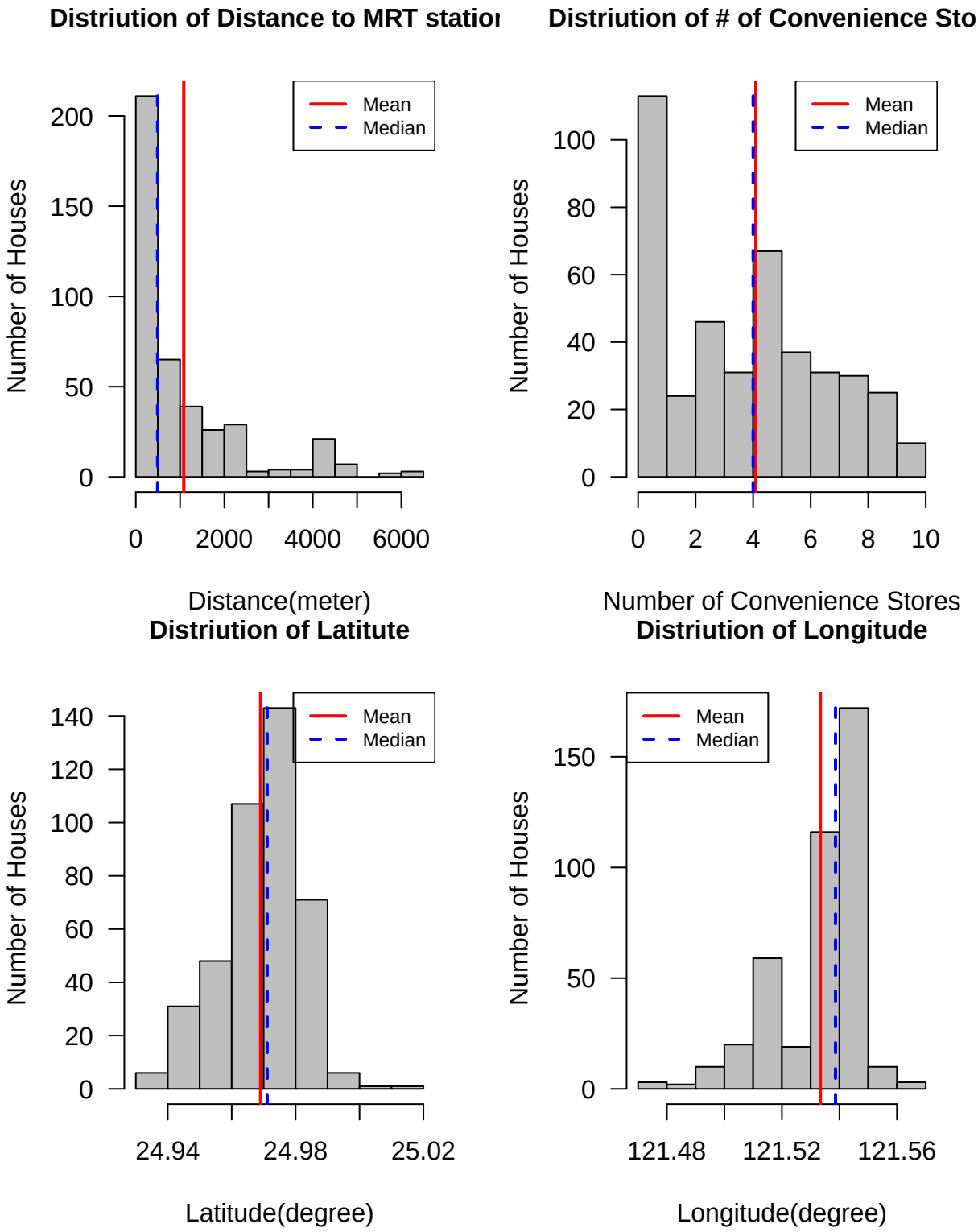
An overview of the Real Estate Valutaion data

```
##        No               X1              X2               X3
##  Min.   :  1.0   Min.   :2013   Min.   : 0.000   Min.   :  23.38
##  1st Qu.:104.2   1st Qu.:2013   1st Qu.: 9.025   1st Qu.: 289.32
##  Median :207.5   Median :2013   Median :16.100   Median : 492.23
##  Mean   :207.5   Mean   :2013   Mean   :17.713   Mean   :1083.89
##  3rd Qu.:310.8   3rd Qu.:2013   3rd Qu.:28.150   3rd Qu.:1454.28
##  Max.   :414.0   Max.   :2014   Max.   :43.800   Max.   :6488.02
##        X4               X5              X6               Y
##  Min.   : 0.000   Min.   :24.93   Min.   :121.5   Min.   :  7.60
##  1st Qu.: 1.000   1st Qu.:24.96   1st Qu.:121.5   1st Qu.: 27.70
##  Median : 4.000   Median :24.97   Median :121.5   Median : 38.45
##  Mean   : 4.094   Mean   :24.97   Mean   :121.5   Mean   : 37.98
##  3rd Qu.: 6.000   3rd Qu.:24.98   3rd Qu.:121.5   3rd Qu.: 46.60
##  Max.   :10.000   Max.   :25.01   Max.   :121.6   Max.   :117.50
```

Distributions of input variables



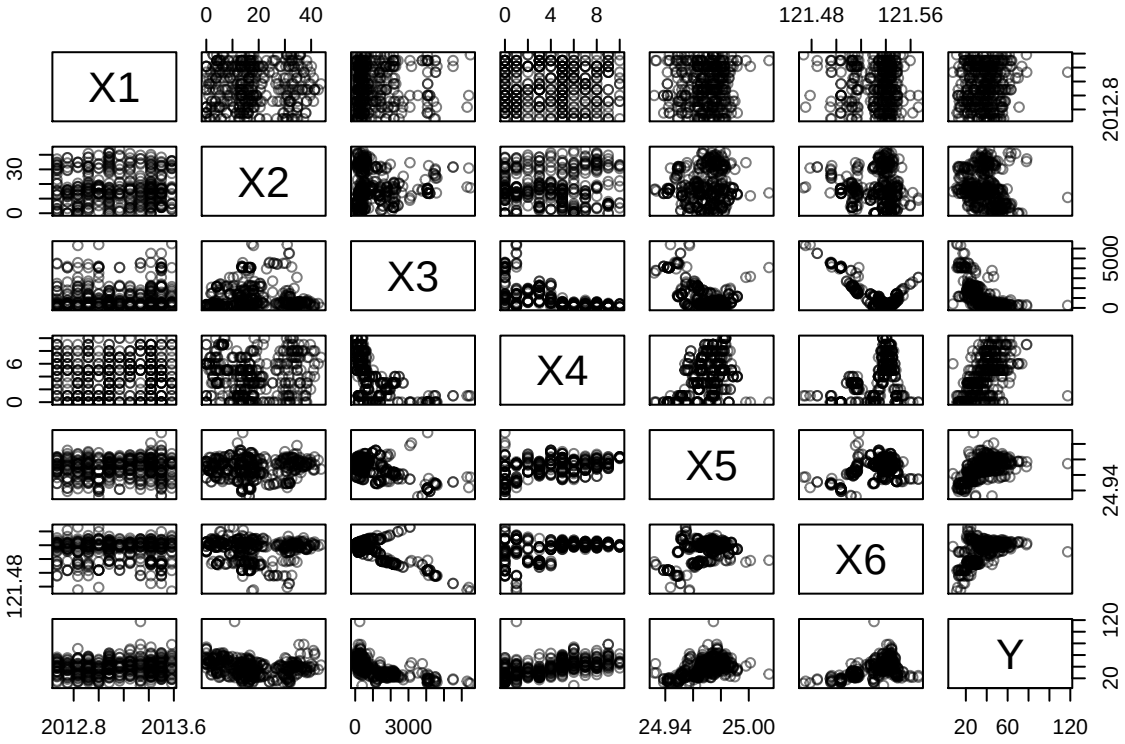




From the overveiw of the data, we see that the average house price per unit area is 379800 new Taiwan dollar/Ping which is approximately 4.8k cad/square meter, 450 cad/sqft. Most transactions happened in 2013 and the average house age is 18 years. Most houses have around 4 convenience stores nearby and is around 1km to the closet MRT station. One thing worth notice is according to the mean longtitude and mean latitude, we find that the majority of the houses are located around Zhonghe District, it lies south-west of New Taipei City, with a total area of 7.836 sq mi and over 410k population, which is a relatively high

population density.

Scatterplot matrix of all the attributes



We notice an obvious pattern when X3 is plotted against X5 and X6, we will look more into the interaction effects between them later.

Regularization:

Here we will fit optimal LASSO, elastic net, and Ridge regression models. The data is randomly split 70/30 for training/test sets by random uniform selection. We begin with the optimal LASSO model for the training set. Next we fit the optimal LASSO model for the training set and compare how well the predictions of both models agree via scatterplot. We repeat this for the elastic net and ridge regression models and constrast each model fit. We analyze the predicative accuracy of each model using the MSPE of the training set.

Important points

- For all three regression models assumptions are the same as least squares regression except the assumption of normality does not have to be validated.
- These models all use a form of regression called regularization. The approach is to constrain or shrink the coefficent estimates, and reduce complex models to avoid the risk of overfitting.
- This decreases the variance but is contingent on added bias. The goal is to find a bias-variance-tradeoff that minimizes the total error which we will determine using cross validation

LASSO Regression:

- Able to perform model selection as it contrains certain coefficents to zero
- If variables are highly correlated, LASSO chooses one and shrinks the others to zero
- Tends to work well with smaller amount of significant variables

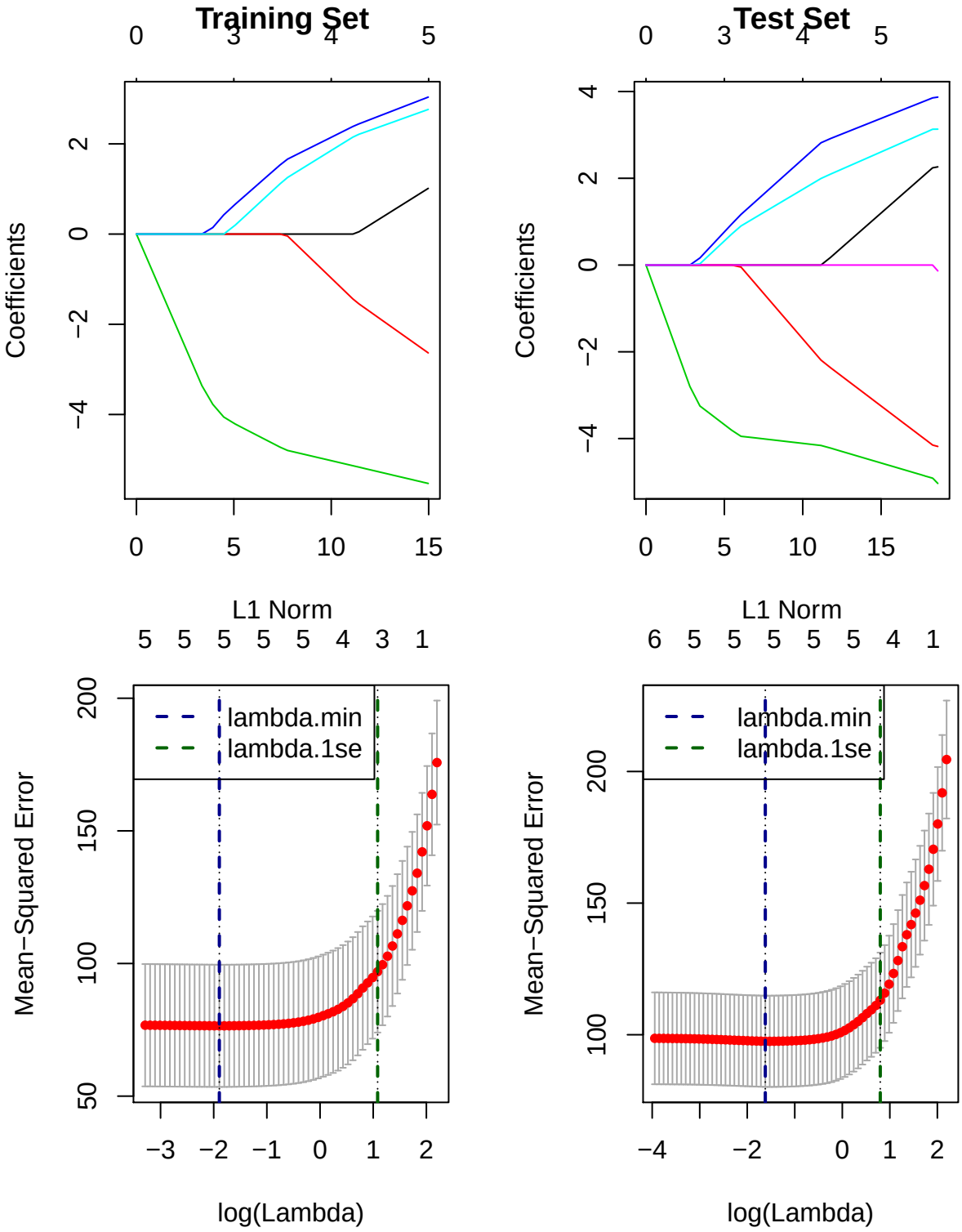
Ridge Regression:

- Ridge regression shrinks the value of coefficients but does not set them to zero, and thus does not perform variable selection
- Coefficents of correlated variables are similar
- Tends to work well for many large variables of similar values

Elastic Net Regression:

- Mixture of LASSO & Ridge
- Constrains coefficents more than ridge but less than LASSO
- Can perform model selection

LASSO (alpha=1) - Plot of Coefficent Paths with Scaled Paramaters & CV-MSPE:



Training set

- Coefficents at minimum lambda

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -1.188129e+04
## X1           3.296473e+00
## X2          -2.201835e-01
## X3          -4.328111e-03
## X4           1.016329e+00
## X5           2.117323e+02
## X6           .
```

- Coefficents at optimal lambda using +1se rule (more sparse)

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -1.841946e+03
## X1           .
## X2           .
## X3          -3.657757e-03
## X4           4.746149e-01
## X5           7.533660e+01
## X6           .
```

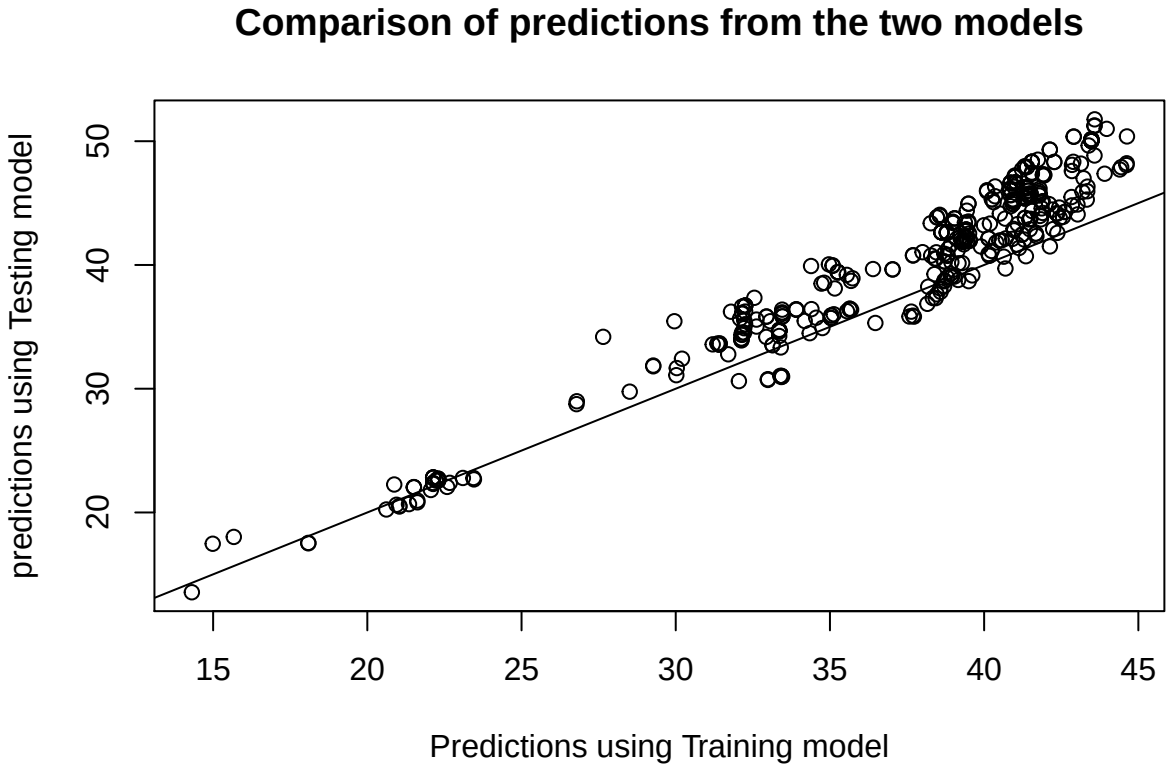
Test set

- Coefficents at minimum lambda

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -2.085035e+04
## X1           7.076311e+00
## X2          -3.546661e-01
## X3          -3.956159e-03
## X4           1.273216e+00
## X5           2.663090e+02
## X6           .
```

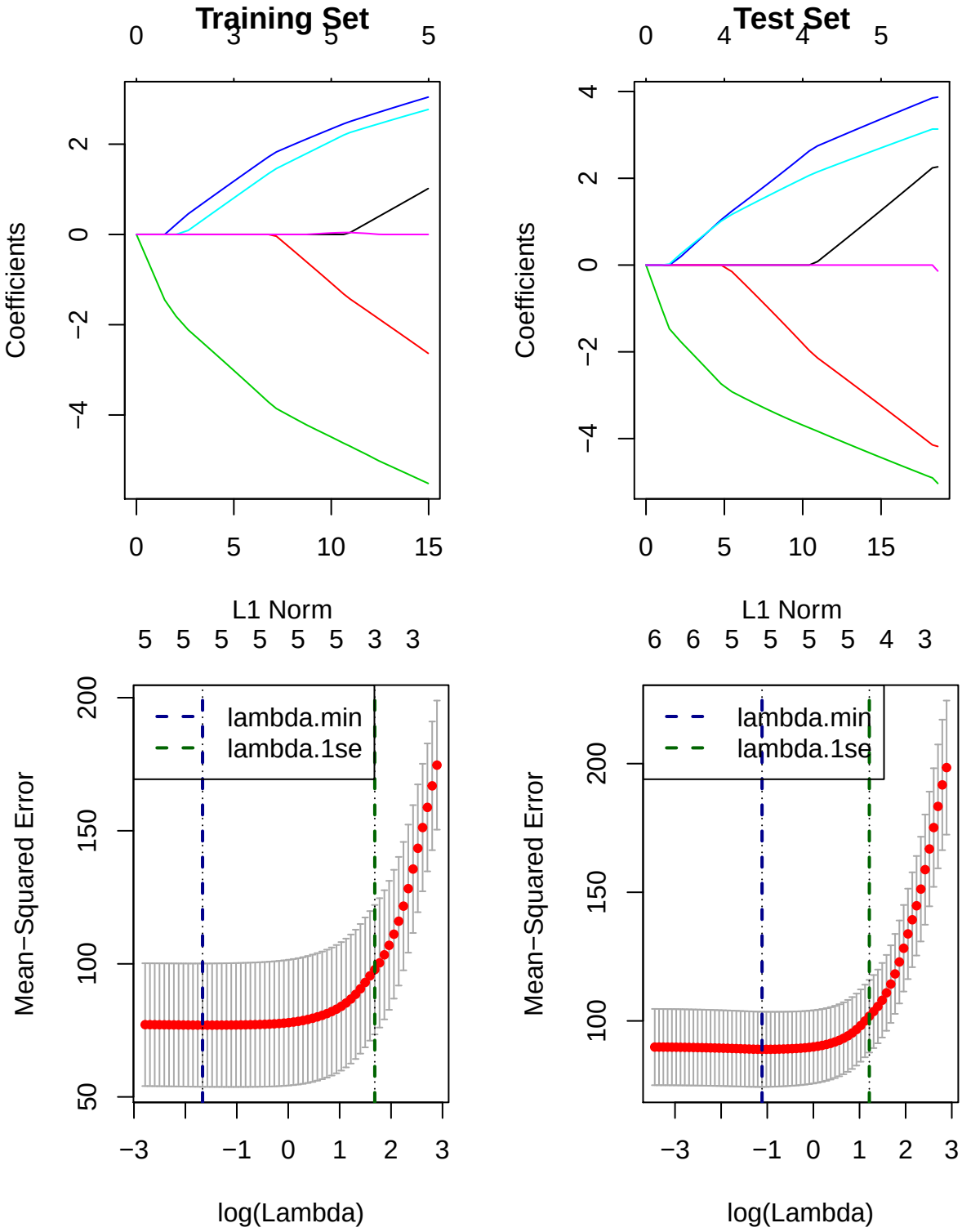
- Coefficents at optimal lambda using +1se rule (more sparse)

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -2.999248e+03
## X1           .
## X2          -9.014144e-02
## X3          -3.285228e-03
## X4           6.446526e-01
## X5           1.218247e+02
## X6           .
```



Comment: The plot of coefficients and CV-MSPE are very similar for both the training & test sets. The LASSO model using the +1SE rule to choose lambda seems like a better fit than lambda min since we are simplifying the model for small reduction in predicative accuracy. We can see this as the training model using +1SE has only 3 variables instead of 5 for the the lambda min model, and only sacrifices a small decrease in MSPE (shown in CV-MSPE plot). The training and test tests both conclude that X6 should be removed from the minimum lambda LASSO models. For the +1SE models, the training set removed variables X1,X2 and x6 whereas the test set removed variable X1,X6. From the scatterplot, we can see that the correlation between the predictors for both models is fairly linear which suggests that both models seem to agree pretty well.

Elastic Net (alpha=0.5) - Plot of Coefficent Paths with Scaled Paramaters & CV-MSPE:



Training set

- Coefficents at minimum lambda

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -1.243913e+04
## X1           3.531634e+00
## X2          -2.258531e-01
## X3          -4.321162e-03
## X4           1.032392e+00
## X5           2.151150e+02
## X6           .
```

- Coefficents at optimal lambda using +1se rule (more sparse)

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -4.102547e+03
## X1           .
## X2          -8.640186e-02
## X3          -3.497694e-03
## X4           7.849729e-01
## X5           1.575743e+02
## X6           1.706143e+00
```

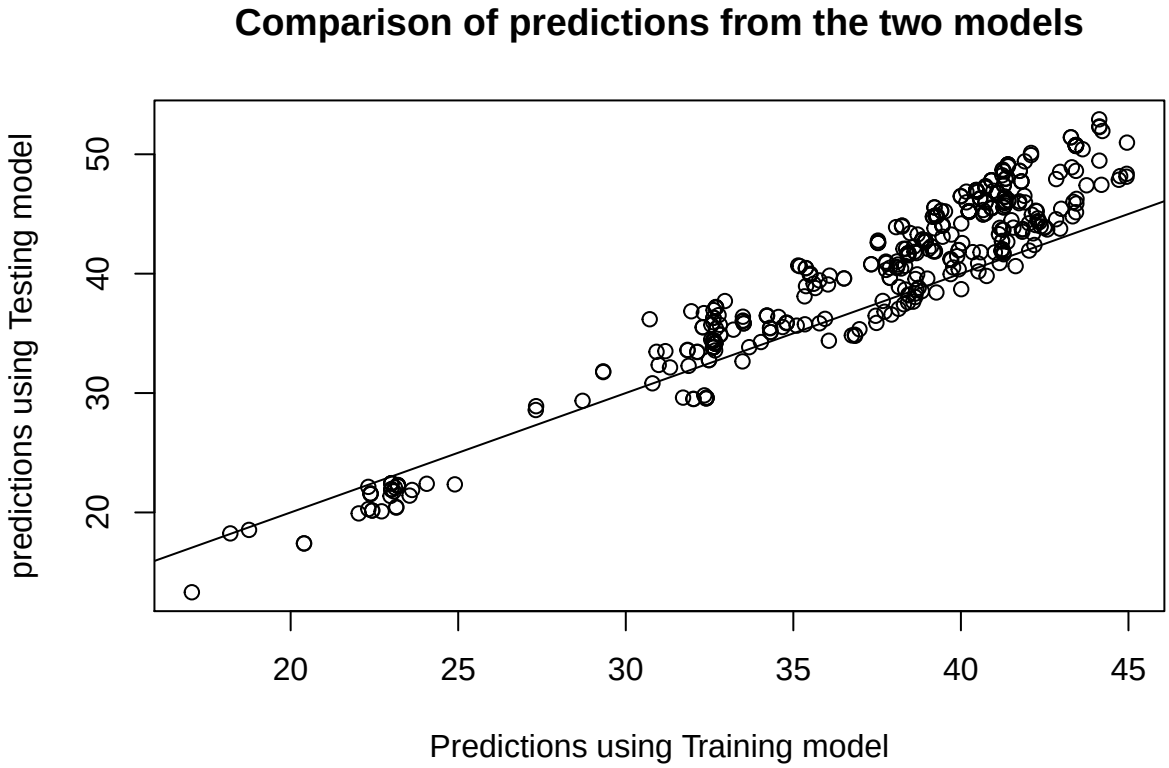
Test set

- Coefficents at minimum lambda

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -2.147496e+04
## X1           7.335478e+00
## X2          -3.599690e-01
## X3          -3.964835e-03
## X4           1.283598e+00
## X5           2.704307e+02
## X6           .
```

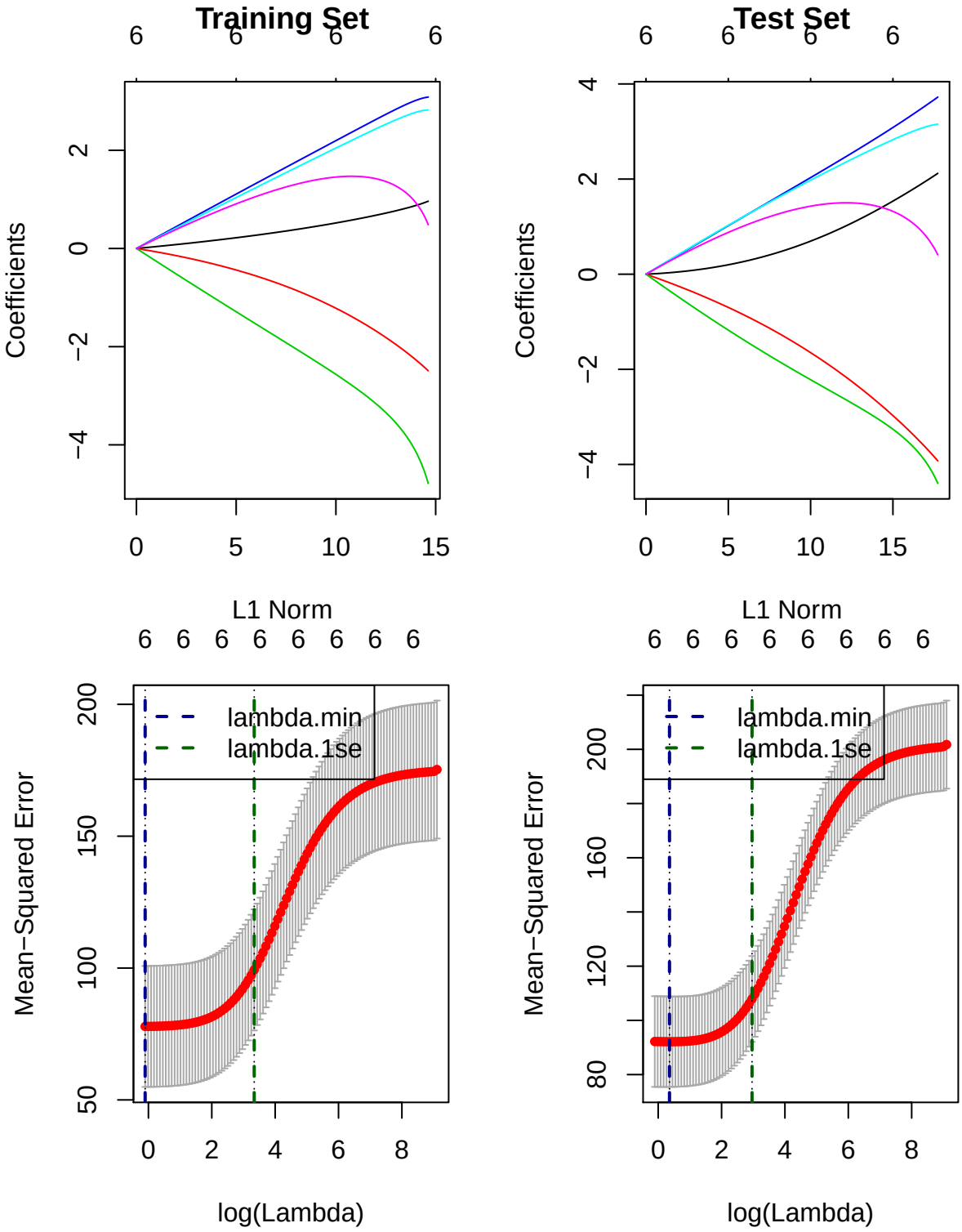
- Coefficents at optimal lambda using +1se rule (more sparse)

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -7.050678e+03
## X1           1.081099e+00
## X2          -2.080411e-01
## X3          -3.216355e-03
## X4           9.653137e-01
## X5           1.969389e+02
## X6           .
```



Comment: The plot of coefficients and CV-MSPE are very similar for both the training & test sets. The Elastic Net model using the +1SE rule to choose lambda is a better fit than lambda min since we are simplifying the model for small reduction in predicative accuracy. The training and test tests for both lambda min and lambda +1SE models include all parameters, where the test set seems to have predictor variables that are more extreme (higher magnitude). From the scatterplot, we can see that the correlation between the predictors for both models is linear (more than LASSO) which suggests that both models agree well with one another.

Ridge Regression (alpha=1) - Plot of Coefficent Paths with Scaled Paramaters & CV-MSPE:



Training set

- Coefficents at minimum lambda

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -1.629160e+04
## X1           3.499402e+00
## X2          -2.181434e-01
## X3          -3.774220e-03
## X4           1.054448e+00
## X5           2.209149e+02
## X6           3.103433e+01
```

- Coefficents at optimal lambda using +1se rule (more sparse)

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -1.988470e+04
## X1           3.029613e+00
## X2          -1.882103e-01
## X3          -3.067210e-03
## X4           1.011035e+00
## X5           2.133795e+02
## X6           6.991952e+01
```

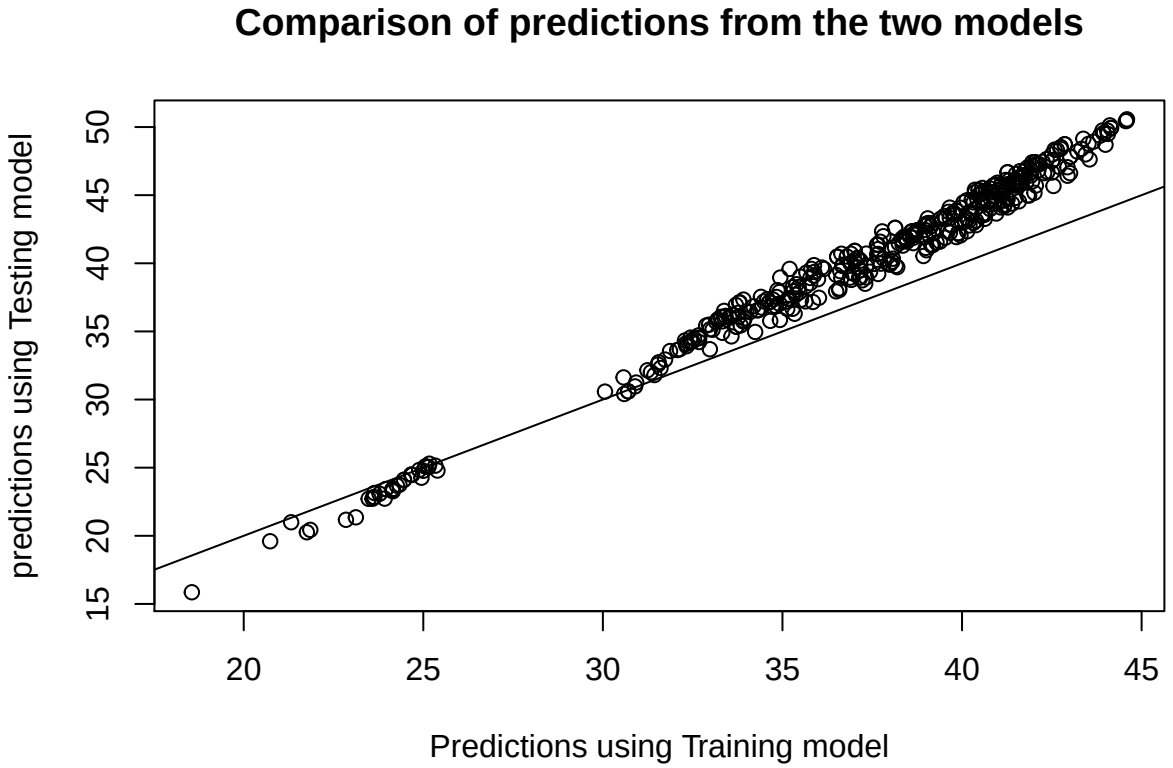
Test set

- Coefficents at minimum lambda

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -2.452737e+04
## X1           7.122323e+00
## X2          -3.451640e-01
## X3          -3.574549e-03
## X4           1.249915e+00
## X5           2.743471e+02
## X6           2.783781e+01
```

- Coefficents at optimal lambda using +1se rule (more sparse)

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) -2.741012e+04
## X1           6.072456e+00
## X2          -2.993886e-01
## X3          -2.973053e-03
## X4           1.136420e+00
## X5           2.624906e+02
## X6           7.137671e+01
```



Comment: The plot of coefficients and CV-MSPE are very similar for both the training & test sets. The Ridge regression model using the +1SE rule to choose lambda is a better fit than lambda min since again, we are simplifying the model for small reduction in predicative accuracy. The training/test tests for both lambda min and lambda +1SE models include all parameters where the test set seems to have more extreme predictor coefficients (similar to Elastic Net). From the scatterplot, we can see that the correlation between the predictors for both models is very linear (more than Elastic Net & LASSO) which suggests that both models agree very well with each other.

Comparison of LASSO, Elastic Net & Ridge

- Assessing Model accuracy with MSPE

```
##         Model Alpha lambda.1se             Optimal_Model     MSPE
## 1       LASSO   1.0   2.953736    Y ~ X3+X4+X5           126.1695
## 2 Elastic_Net   0.5   5.382669 Y ~ X1+X2+X3+X4+X5        128.4081
## 3       Ridge   0.0  28.194843 Y ~ X1+X2+X3+X4+X5+X6     127.9746
```

Comments

We can see that all three models are similar in terms of their prediction errors with LASSO having the highest MSPE (126.17), Ridge having the lowest MSPE (122.82). The LASSO model has 3 variables, Elastic Net has 5 and Ridge Regression contains the full model.

Overall, LASSO seems to provide the most optimal model as it works well with a smaller amount of significant variables, which is the case here as the full model consists of 6 total variables. We can see that the LASSO provides the most sparse/simple fit as it contains the least number variables (half the full model) in exchange for a small decrease in MSPE. Note that the LASSO eliminates unsignificant variables and handles multicollinearity by penalizing highly correlated variables by only keeping one. In this case, X1 (transaction data), X6 (age) was removed as they were considered to be insignificant and X6 (longitude) was removed as it is correlated with X5 (latitude).

The overall optimal model between the 3 methods is

$$Y = -1841.9460 - 0.0037X_3 + 0.4746X_4 + 75.3366X_5$$

which include the variables X3: distance to the nearest MRT station, X4: number of convenience stores in the living circle, and X5: latitude coordinate

Smoothing method - Smoothing spline (5-fold CV)

1. No Interactions(GAM):

Shown below is the summary of the model and AIC and MSE of each fold

```
## Loading required package: nlme
## This is mgcv 1.8-24. For overview type 'help("mgcv-package")'.
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Y ~ s(X1) + s(X2) + s(X3) + s(X4) + s(X5) + s(X6)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   37.536      0.416   90.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##         edf Ref.df      F  p-value
## s(X1) 1.000  1.000 18.613 2.14e-05 ***
## s(X2) 2.913  3.627 13.393 3.28e-09 ***
## s(X3) 4.364  5.254 10.637 1.18e-09 ***
## s(X4) 1.000  1.000  0.504    0.478
## s(X5) 6.324  7.400  7.375 1.77e-08 ***
## s(X6) 3.034  3.855  0.628    0.688
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.689   Deviance explained = 70.7%
## GCV = 60.734  Scale est. = 57.12     n = 330
## [1] 2220.957 2282.670 2312.175 2305.593 2292.428
## [1] 100.85320  68.17734  41.60264  45.92523  49.12581
```

2. Interaction effects between the distance to the nearest MRT station(X3) and longitude of the house(X6):

Shown below is the model and the AIC and MSE of each fold

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Y ~ s(X1) + s(X2) + s(X3) + s(X4) + s(X5) + s(X6) + s(X3, X6)
##
## Parametric coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.5361     0.4011   93.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf  Ref.df      F  p-value
## s(X1)     1.0000  1.0000 15.816 8.69e-05 ***
## s(X2)     5.7683  6.9243  7.241 6.53e-08 ***
## s(X3)     0.9999  0.9999 12.295 0.000522 ***
## s(X4)     1.0000  1.0000  2.814 0.094473 .
## s(X5)     1.1407  1.2648 32.431 4.47e-07 ***
## s(X6)     1.0000  1.0000  0.040 0.842180
## s(X3,X6) 16.1855 23.0000  5.819  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 75/82
## R-sq.(adj) =  0.711   Deviance explained = 73.5%
## GCV = 58.038  Scale est. = 53.097    n = 330
```

```
## [1] 2218.163 2276.581 2312.175 2305.593 2276.123
```

```
## [1] 96.60481 66.70653 41.60264 45.92523 52.55143
```

From the summary of the model, we can tell that the interaction effect of the distance to the nearest station(X3) and the longtitude of the house is significant, which makes sense since wether there is a nearby MRT station or not highly depends on the location of the house.

3. Interaction effects between the distance to the nearest MRT station(X3), latitude and longitude of the

house(X6):

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Y ~ s(X1) + s(X2) + s(X3) + s(X4) + s(X5) + s(X6) + ti(X3, X5,
##     X6)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   36.389      1.153   31.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                edf Ref.df      F  p-value
## s(X1)        1.000  1.000 18.001 2.92e-05 ***
## s(X2)        2.152  2.681 14.007 1.03e-07 ***
## s(X3)        2.374  2.843 13.601 2.50e-05 ***
## s(X4)        2.530  3.114  1.411   0.2568
```

```
## s(X5)         7.957  8.493  5.143 3.28e-06 ***
## s(X6)         1.000  1.000  0.001   0.9697
## ti(X3,X5,X6) 10.774 13.073  2.058   0.0168 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.707   Deviance explained = 73.2%
## GCV = 58.909  Scale est. = 53.77     n = 330
```

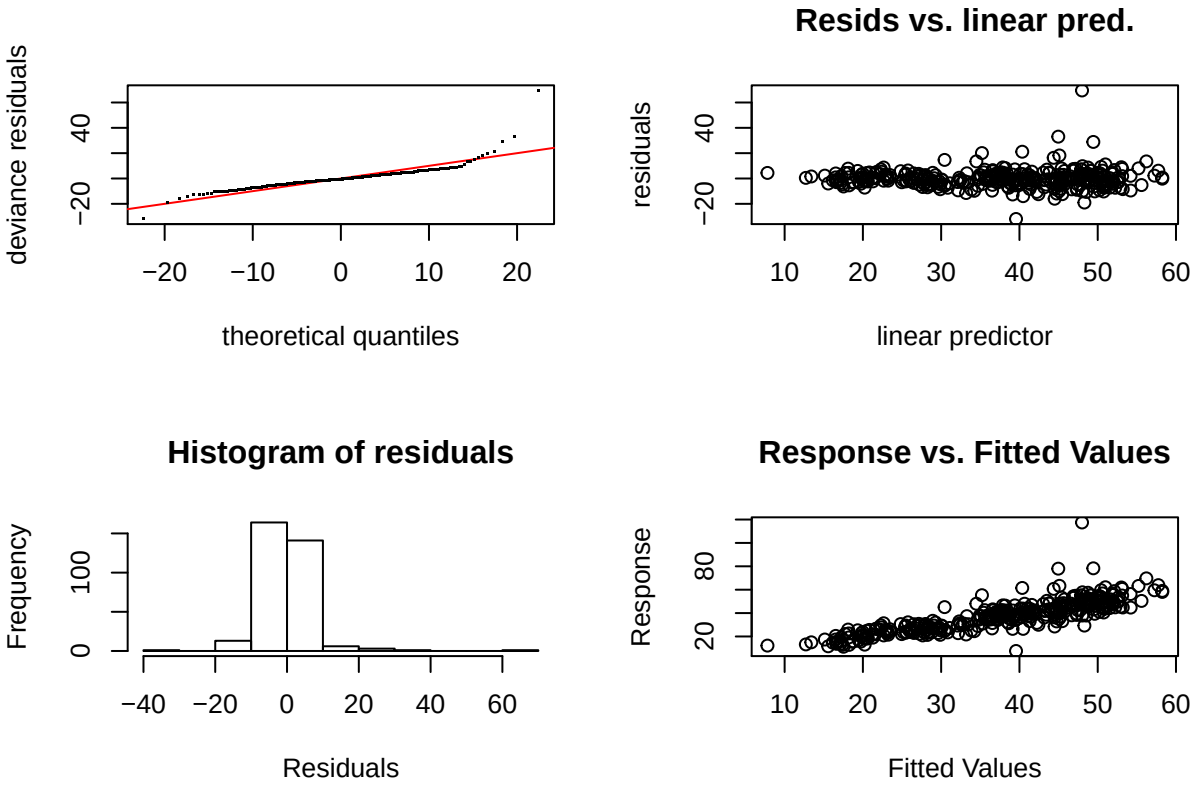
```
## [1] 2191.615 2272.286 2299.571 2296.344 2280.911
```

```
## [1] 102.77982  69.93951  50.17443  42.54973  46.61192
```

From the summary of the model including the three way interaction between distance to the closest MRT station, the latitude and longtitude of the house, the interaction effect does not seem as important as the two way interaction effect.

Compare the Residuals

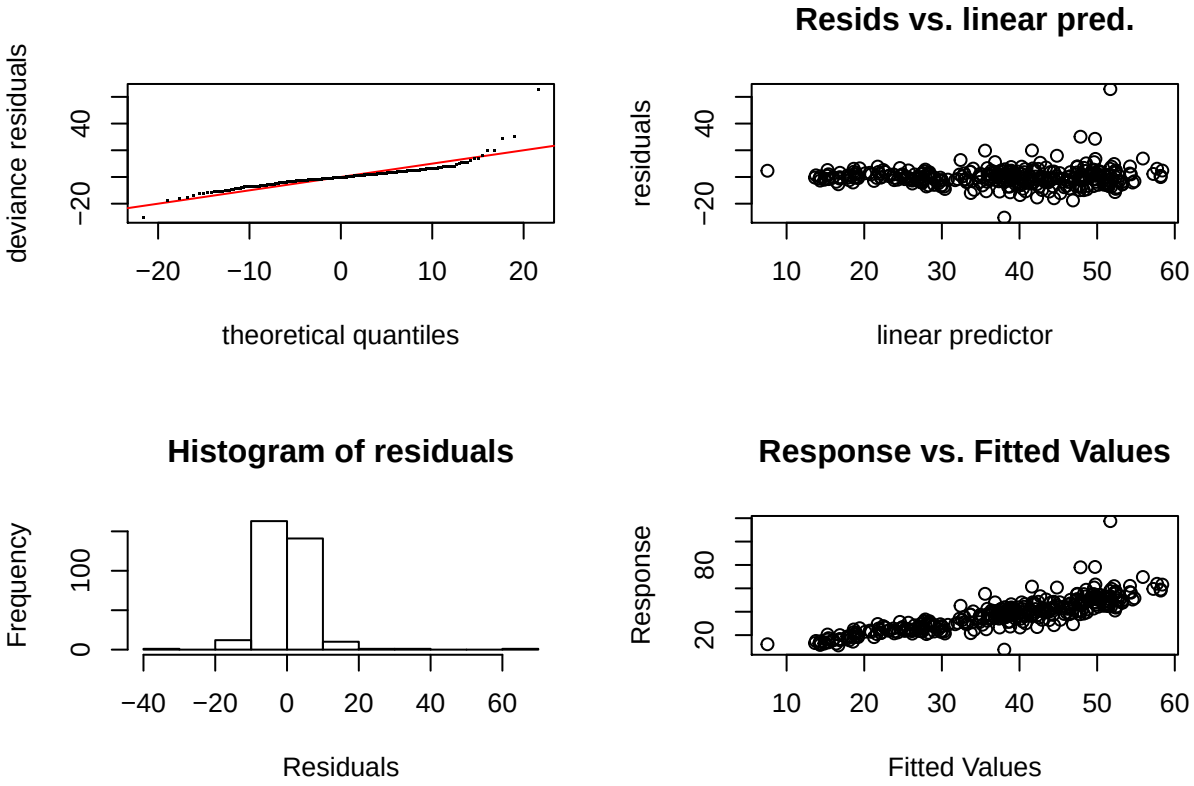
1. No Interaction




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 29 iterations.
## The RMS GCV score gradient at convergence was 1.496395e-06 .
## The Hessian was positive definite.
## Model rank =  55 / 55
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
```

```
## indicate that k is too low, especially if edf is close to k'.
##
##         k'  edf k-index p-value
## s(X1) 9.00 1.00    1.00    0.43
## s(X2) 9.00 2.91    0.98    0.39
## s(X3) 9.00 4.36    0.82  <2e-16 ***
## s(X4) 9.00 1.00    1.09    0.95
## s(X5) 9.00 6.32    0.99    0.38
## s(X6) 9.00 3.03    0.96    0.18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

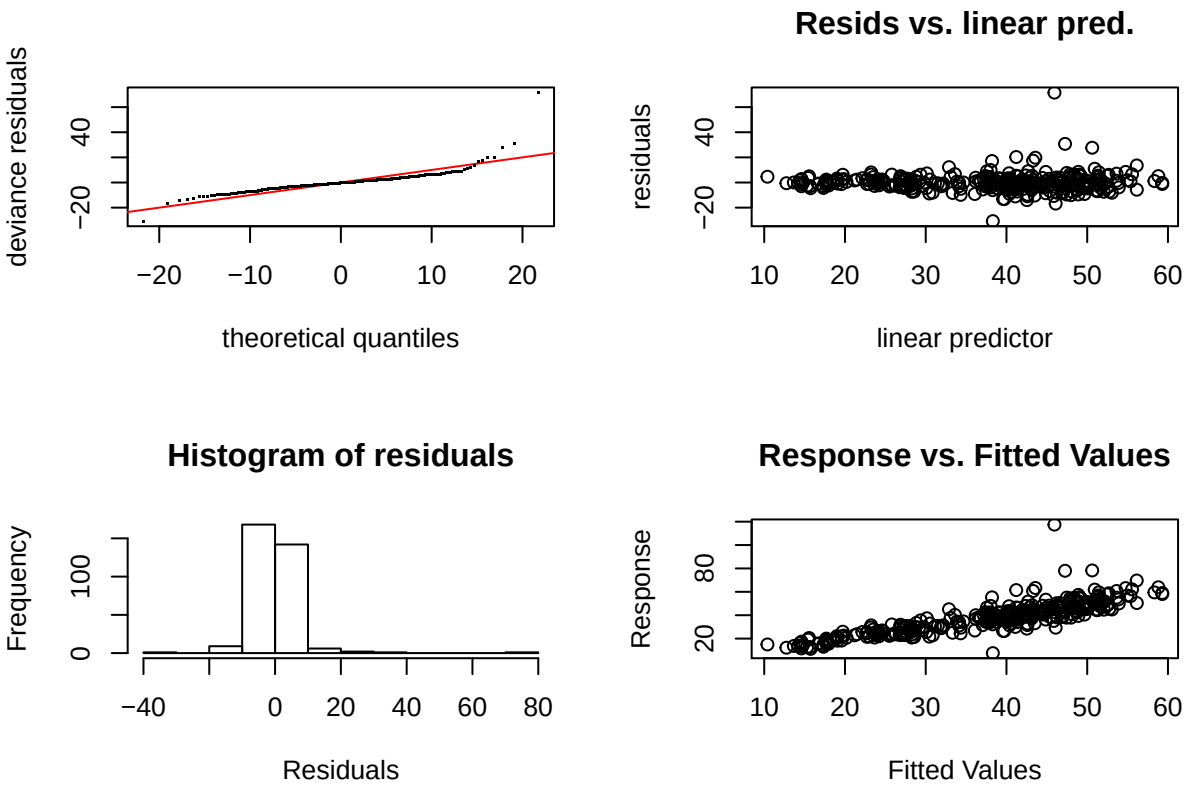
2. Interaction effects between the distance to the nearest MRT station(X3) and longitude of the house(X6):




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 20 iterations.
## The RMS GCV score gradient at convergence was 2.425168e-06 .
## The Hessian was positive definite.
## Model rank =  75 / 82
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##             k'   edf k-index p-value
## s(X1)     9.00  1.00    1.00    0.54
## s(X2)     9.00  5.77    0.98    0.34
```

```
## s(X3)     9.00  1.00    0.91    0.05 *
## s(X4)     9.00  1.00    1.09    0.96
## s(X5)     9.00  1.14    1.03    0.68
## s(X6)     9.00  1.00    0.99    0.43
## s(X3,X6) 27.00 16.18    1.04    0.66
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3. Interaction effects between the distance to the nearest MRT station(X3), latitude and longitude of the house(X6):




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 15 iterations.
## The RMS GCV score gradient at convergence was 1.955033e-06 .
## The Hessian was positive definite.
## Model rank =  119 / 119
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                k'   edf k-index p-value
## s(X1)        9.00  1.00    1.00   0.450
## s(X2)        9.00  2.15    1.02   0.550
## s(X3)        9.00  2.37    0.86   0.025 *
## s(X4)        9.00  2.53    1.10   0.975
## s(X5)        9.00  7.96    1.03   0.670
## s(X6)        9.00  1.00    1.01   0.505
```

```
## ti(X3,X5,X6) 64.00 10.77    0.91   0.235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Compare the AIC & MSE

	MSE	AIC
model1	61.13684	2282.765
model2	60.67813	2277.727
model3	62.41108	2268.145

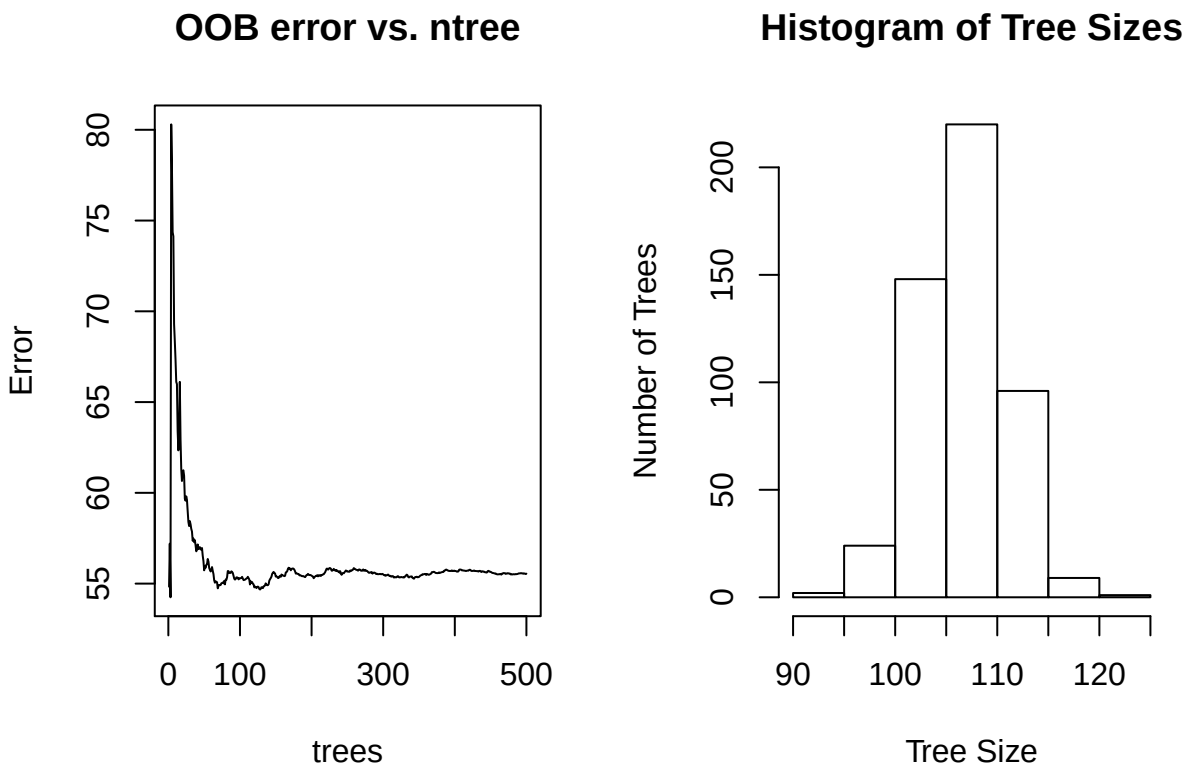
While there seem to be no big difference between the residual plots of all three models, and since all the model have the same mean AIC, we pick the model with the smallest MSE, which is model2 that considers the interaction effects between the distance to the nearest MRT station(X3) and longitude of the house(X6).

Random Forest(m=p/3):

Shown below is the function call

```
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
##
## Call:
##  randomForest(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6, data = RE[-folds[[i]],      ], mtry = 2, nt
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 55.54676
##                     % Var explained: 69.68
```

Plot OOB error vs. number of trees and histogram of tree sizes



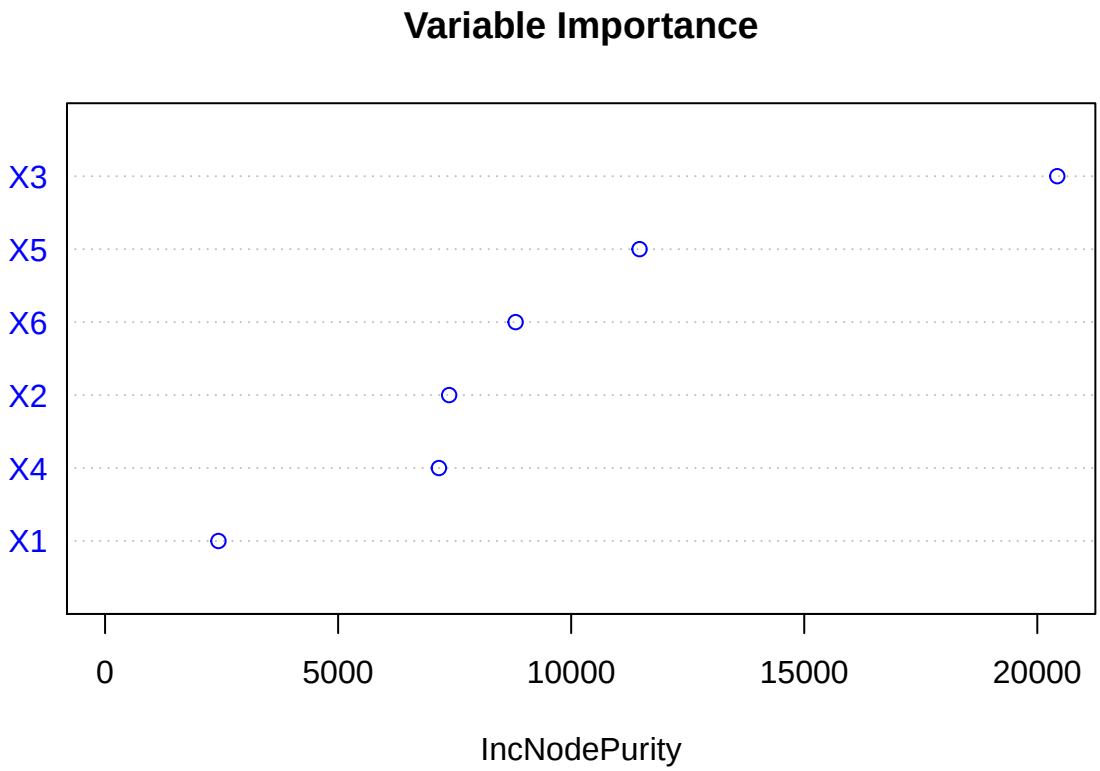
The average MSE of 5 folds

```
## [1] "The average MSE:53.197"
```

The Random Forest model has a smaller overall MSE compare to the smoothing spline models.

Variable Importance

	IncNodePurity
X1	2432.422
X2	7382.219
X3	20428.380
X4	7162.587
X5	11466.435
X6	8807.521



From the variable importance plot we can tell that distance to the nearest MRT station (X3) is the most important variable in the model, followed by the latitude(X5) and longtitude(X6) of the house, which means the geographical locaation of the house is more important compare to the house age(X2) and the number of convenience stores in the living circle of the house(X4). The transaction date(X1), has the smallest impact on the house price which makes sense since the dataset only recorded data over a period of less than one year.

Boosting - Gradient Boosting

Again we use 5-fold CV

```
## Stochastic Gradient Boosting
##
## 414 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 331, 332, 331, 330, 332
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  RMSE      Rsquared   MAE
##   1                   50      7.615315  0.6946961  5.254969
##   1                  100      7.520602  0.6970835  5.174350
##   1                  150      7.449194  0.7044337  5.150694
##   2                   50      7.313082  0.7141454  4.988658
##   2                  100      7.412922  0.7054831  5.025514
##   2                  150      7.380668  0.7075623  4.960212
##   3                   50      7.331455  0.7122314  4.956300
##   3                  100      7.391674  0.7081756  4.950987
##   3                  150      7.463843  0.7030555  5.021127
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were n.trees = 50, interaction.depth
##  = 2, shrinkage = 0.1 and n.minobsinnode = 10.
```

We use tuneGrid to test out different combinations of tuning parameters:

- n.trees = 100, 150, 200
- interaction.depth = 5, 6, 7
- shrinkage = 0.01 0.05 0.1
- n.minobsinnode hold constant at 10

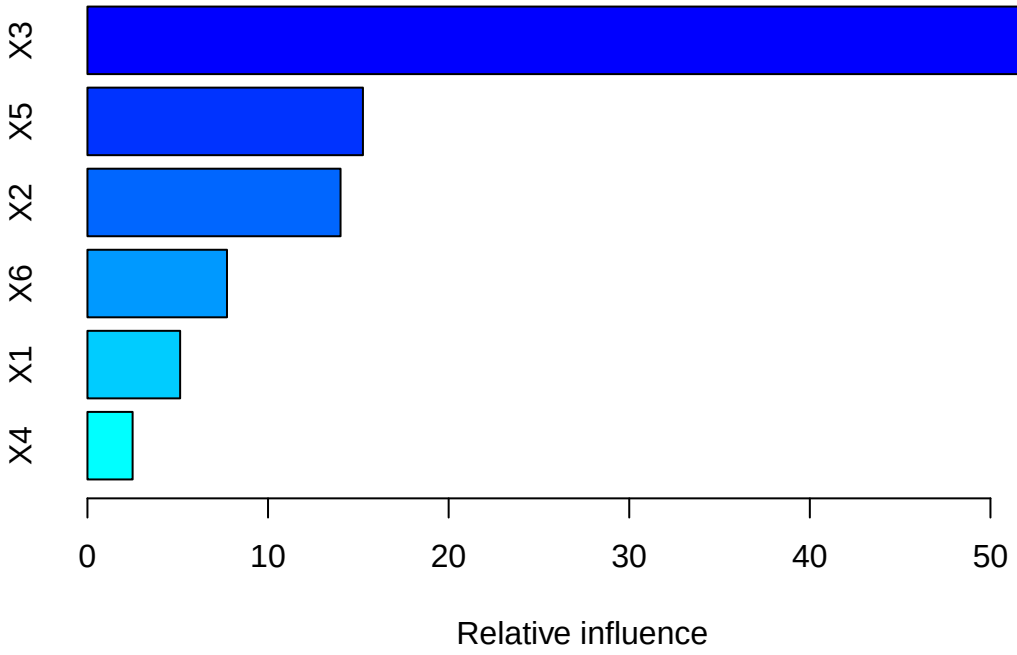
```
## Stochastic Gradient Boosting
##
## 414 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 331, 331, 331, 332, 331
## Resampling results across tuning parameters:
##
##   shrinkage  interaction.depth  n.trees  RMSE      Rsquared   MAE
##   0.01       5                  100      8.651474  0.7080854  6.148848
##   0.01       5                  150      7.856225  0.7148870  5.406006
##   0.01       5                  200      7.510220  0.7176154  5.073174
##   0.01       6                  100      8.593074  0.7096690  6.121617
```

```
##   0.01       6                  150      7.821098  0.7148497  5.390095
##   0.01       6                  200      7.487477  0.7181604  5.057055
##   0.01       7                  100      8.545634  0.7108166  6.062477
##   0.01       7                  150      7.795922  0.7143753  5.342505
##   0.01       7                  200      7.454337  0.7184104  5.001497
##   0.05       5                  100      7.393949  0.7118122  4.940093
##   0.05       5                  150      7.471290  0.7050037  4.936214
##   0.05       5                  200      7.554115  0.6988430  4.986143
##   0.05       6                  100      7.316382  0.7160043  4.809984
##   0.05       6                  150      7.418476  0.7079822  4.859016
##   0.05       6                  200      7.482185  0.7038638  4.910161
##   0.05       7                  100      7.258537  0.7205179  4.789504
##   0.05       7                  150      7.309913  0.7151852  4.761810
##   0.05       7                  200      7.402233  0.7090324  4.822341
##   0.10       5                  100      7.413451  0.7076242  4.978283
##   0.10       5                  150      7.546806  0.6980749  5.098110
##   0.10       5                  200      7.567540  0.6962322  5.093395
##   0.10       6                  100      7.432457  0.7070287  4.931466
##   0.10       6                  150      7.549687  0.6980481  5.053200
##   0.10       6                  200      7.721434  0.6867973  5.179177
##   0.10       7                  100      7.438766  0.7050040  4.921212
##   0.10       7                  150      7.541266  0.6979999  5.014285
##   0.10       7                  200      7.724637  0.6853624  5.117933
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were n.trees = 100,
##  interaction.depth = 7, shrinkage = 0.05 and n.minobsinnode = 10.
```

The optimal set of tuning parameters is:

- n.trees=150, interaction.depth = 6, shrinkage=0.05 and n.minobsinnode = 10

Rerun the model with optimal tuning parameters




```
##    var   rel.inf
## X3  X3 55.367733
## X5  X5 15.255181
## X2  X2 14.014930
## X6  X6  7.727632
## X1  X1  5.133362
## X4  X4  2.501162
```

We see that while distance to the nearest MRT station (X3) remains to be the most influencial variable in the gradient boosting model, all the other varialbe importances are different from the random forest model. The second important variable becomes house age(X2), and followed by the latitude(X5) and longtitude(X6) of the house, number of convenience stores in the living circle of the house(X4) becomes the least important variable in this model.

The overall MSE of 5 folds

```
## [1] "The average MSE:56.073"
```

The overall MSE generated from the gradient boosting model is slightly bigger than the one from random forest model, but still slightly smaller than the smoothing spline models.

Statistical Conclusion

Model MSEs

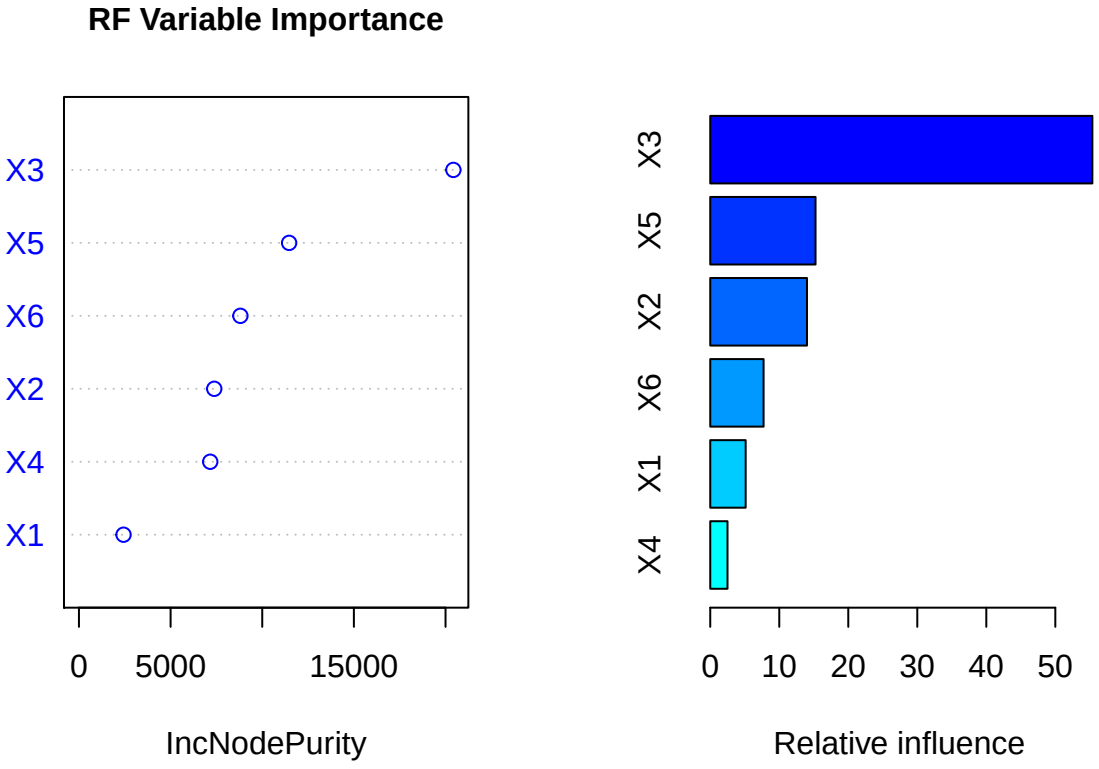
To decide on which model is our best candidate, we compare the MSE of each model, note: for smoothing spline we used model2 which is the best one among the three models we fitted above.

	MSE
LASSO	89.50584
Elasticnet	90.84076
Ridge Regression	126.88729
Smoothing Spline	60.67813
Random Forest	53.19688
Gradient Boosting	56.07304

From the MSE table above, among all the models, gradient boosting has the smallest MSE, therefore the gradient boosting model is the best candidate for fitting this dataset.

Variable Importance

Since our motivation is to find which factors affect the housing price the most, we also want to compare the variable importance from the random forest and the gradient boosting model.




```
##    var   rel.inf
## X3  X3 55.367733
## X5  X5 15.255181
## X2  X2 14.014930
```

```
## X6  X6  7.727632
## X1  X1  5.133362
## X4  X4  2.501162
```

By comparing the variable importance graphs, there is only a slight difference between two models, both models show that the distance to the nearest MRT station is the most crucial factor, and the latitude is the second most important, the house age and the longitude are right after, and the transaction date and the number of convenience stores around the house have the smallest impact on the house prices.

Conclusion

Even though we chose gradient boosting model as the best candidate, it does not necessarily mean it is the true model for our dataset, but all the model fitting does provide a valuable insight on the question we are trying to answer, which is what factor amongst the six has the biggest influence on Taiwan house price. However since the dataset we have is relatively small with only 414 instances, and was only collected from one district in Taiwan, the conclusion we drew from it can somehow be biased, also the dataset only consists of data collected from mid 2012 to mid 2013, which is a really short period of time, the importance of all the factors may change in the following years. Overall, the conclusion we reach applies to the New Taipei City and considering the population base is big enough, it can be applied to the whole Taiwan, but due to the bias that might exist, it cannot be generalized and applied to other cities or a bigger region, which lead us to the next part, what can we do in future work to strength the model.

Future Work

Since we want a model that can give us less biased result and provide more accurate prediction, we need a bigger datset with more instances as well as taking more detailed and even financial factors into account, such as number of bedrooms in the house, the interest rate while purchaing the house, etc. And the data need to be collected over a longer period of time. In the process of model fitting, we can achieve a better result by putting more weight on the more important factors. We can also fit more models to see if any outperforms the gradient boosting model, for example the genralized linear model and logistic regression, etc.

Contribution

Congxiao Jin: Introduction, Smoothing Method, Random Forest, Boosting, Conclusion

Shuby Sharma: Regularization-LASSO, Ridge Regression, Elastic Net Regression