EAS508- REGRESSION PROJECT 1

Team Members:

Alifiya Aziz Batterywala (50539267)

Karthika Rajan Nair (50510469)

Meghna Ramesan Aalingil (50496345)

Prachi Vijay Patil (50539237)

HEALTH DATA SET

HEALTH DATASET DESCRIPTION:

X1 = death rate per 1000 residents

X2 = doctor availability per 100,000 residents

X3 = hospital availability per 100,000 residents

X4 = annual per capita income in thousands of dollars

X5 = population density people per square mile

MODEL 1 - The Lasso

```
library (glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.3.2
```

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

```
## Loaded glmnet 4.1-8
```

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.3.2
```

```
library(leaps)
library ( MASS )
library(car)
```

```
## Loading required package: carData
```

```r
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.3.2
```

```r
library(boot)
```

```
##
## Attaching package: 'boot'

## The following object is masked from 'package:car':
##
##     logit
```

```r
library(mgcv)
```

```
## Loading required package: nlme

## This is mgcv 1.8-42. For overview type 'help("mgcv-package")'.
```

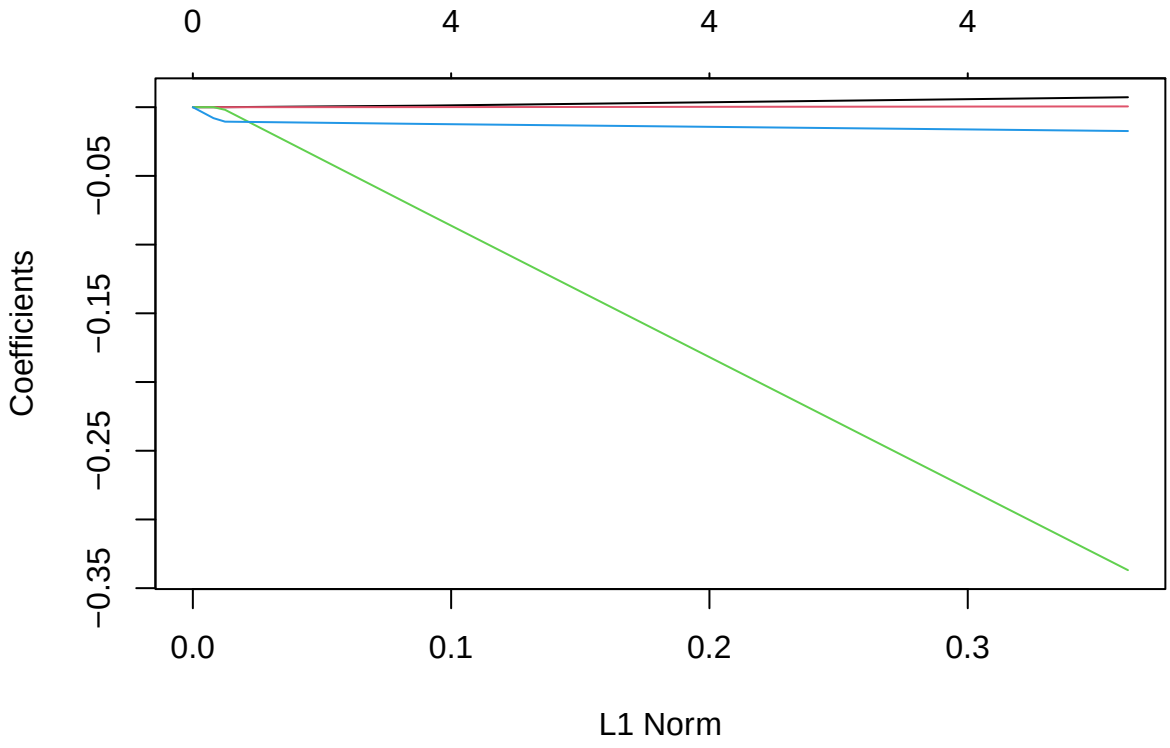
```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.2

## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:boot':
##
##     melanoma

##
## Attaching package: 'caret'

## The following objects are masked from 'package:Metrics':
##
##     precision, recall
```

```r
library(boot)
Health <- read_excel("Health.xlsx")
healthdata = Health
```

```
set.seed(1)
# Fit a lasso model in order to predict X1 (Death rate) on the Health data.
x <- model.matrix (X1 ~ ., healthdata)[, -1]
y <- healthdata$X1
grid <- 10 ^ seq (10, -2, length = 100)

train <- sample (1: nrow (x), nrow (x) / 1.25)
test <- (-train)
y.test <- y[test]
lasso.mod <- glmnet (x[train , ], y[train], alpha = 1, lambda = grid)
plot (lasso.mod)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```




```
# Evaluate using K-Fold Cross-Validation
# 10 fold cross validation- select best lambda value and find MSE
cv.out <- cv.glmnet(x[train , ], y[train], alpha = 1, nfolds = 10)

bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.002503384
```

```
lasso.pred <- predict (lasso.mod, s = bestlam, newx = x[test,])

mse <- mean ((lasso.pred - y.test)^2)
mse
```

```
## [1] 2.819071
```

```
out <- glmnet (x, y, alpha = 1, lambda = grid)
lasso.coef <- predict (out , type = "coefficients", s = bestlam)[1:5, ]
lasso.coef
```

```
##   (Intercept)            X2            X3            X4            X5
## 12.1679669583  0.0070158133  0.0005595802 -0.3160027833 -0.0092649168
```

X4 is the only significant coefficient (-0.3160027833)

```
# Evaluate using LOOCV
# Select best lambda value and find MSE
set.seed(1)
cv.out <- cv.glmnet(x[train , ], y[train], alpha = 1, nfolds =
                      nrow(healthdata[train,]))
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.3466902
```

```
lasso.pred <- predict (lasso.mod, s = bestlam, newx = x[test,])

mse <- mean ((lasso.pred - y.test)^2)
mse
```

```
## [1] 1.725168
```

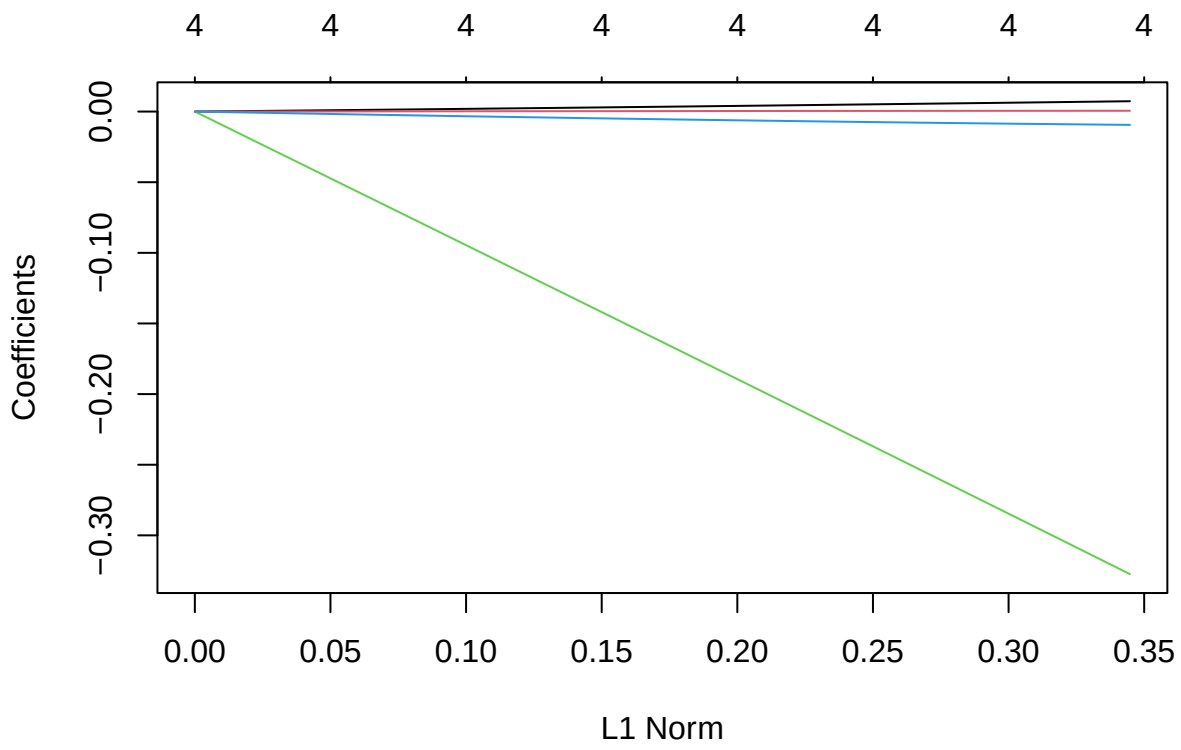
```
out <- glmnet (x, y, alpha = 1, lambda = grid)
lasso.coef <- predict (out , type = "coefficients", s = bestlam)[1:5, ]
lasso.coef
```

```
##  (Intercept)           X2           X3           X4           X5
##  9.567190119  0.000000000  0.000000000  0.000000000 -0.002363758
```

X5 is only significant coefficient (-0.002363758)

MODEL 2 - Ridge Regression

```
set.seed(1)
x <- model.matrix (X1 ~ ., healthdata)[, -1]
y <- healthdata$X1
grid <- 10 ^ seq (10, -2, length = 100)

# Perform Ridge Regression
ridge.mod <- glmnet (x, y, alpha = 0, lambda = grid)
plot (ridge.mod)
```




```
# Evaluate using K-Fold Cross-Validation
# 10 fold cross validation- select best lambda value and find MSE
cv.out <- cv.glmnet(x[train , ], y[train], alpha = 0, nfolds = 10)

bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.8197454
```

```
ridge.pred <- predict (ridge.mod, s = bestlam, newx = x[test,])
mse <- mean ((ridge.pred - y.test)^2)
mse
```

```
## [1] 1.549471
```

```
out <- glmnet (x, y, alpha = 0, lambda = grid)
ridge.coef <- predict (out , type = "coefficients", s = bestlam)[1:5, ]
ridge.coef
```

```
##   (Intercept)            X2            X3            X4            X5
## 11.1693701174  0.0042316271  0.0004003988 -0.1988953373 -0.0064567400
```

From the coefficient values, we see that X4 is weakly related to the output and the rest of the coefficients are negligible.

```
# Evaluate using LOOCV
# Select best lambda value and find MSE
set.seed(1)
cv.out <- cv.glmnet(x[train , ], y[train], alpha = 0, nfolds = nrow(healthdata))
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.8996696
```

```
ridge.pred <- predict (ridge.mod, s = bestlam, newx = x[test,])

mse <- mean ((ridge.pred - y.test)^2)
mse
```

```
## [1] 1.539418
```

```
out <- glmnet (x, y, alpha = 0, lambda = grid)
ridge.coef <- predict (out , type = "coefficients", s = bestlam)[1:5, ]
ridge.coef
```

```
##   (Intercept)            X2            X3            X4            X5
## 11.1052256261  0.0040632881  0.0003875297 -0.1915659431 -0.0062568266
```

LOOCV also suggests that all coefficients except X4 are negligible and X4 has a very weak negative relationship with the output.

MODEL 3 - Multiple Linear Regression

```
# We are using 3 different multiple linear regression models and evaluating whic
# h one fits the model well.
set.seed(1)
model <- lm(X1 ~ X2 + X3 + X4 + X5, data = healthdata)
summary(model)
```

```
##
## Call:
## lm(formula = X1 ~ X2 + X3 + X4 + X5, data = healthdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6404 -0.7904  0.3053  0.9164  2.7906
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.2662552  2.0201467   6.072 1.95e-07 ***
## X2           0.0073916  0.0069336   1.066   0.2917
## X3           0.0005837  0.0007219   0.809   0.4228
## X4          -0.3302302  0.2345518  -1.408   0.1656
## X5          -0.0094629  0.0048868  -1.936   0.0587 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.601 on 48 degrees of freedom
## Multiple R-squared:  0.1437, Adjusted R-squared:  0.07235
## F-statistic: 2.014 on 4 and 48 DF,  p-value: 0.1075
```

```r
model1 <- lm(X1 ~ X4 + X5, data = healthdata)
summary(model1)
```

```
##
## Call:
## lm(formula = X1 ~ X4 + X5, data = healthdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9334 -1.0161  0.0936  1.0659  2.5673
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.339452   1.992835   6.192  1.1e-07 ***
## X4          -0.214183   0.209616  -1.022   0.3118
## X5          -0.009154   0.004778  -1.916   0.0611 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.612 on 50 degrees of freedom
## Multiple R-squared:  0.09594,    Adjusted R-squared:  0.05978
## F-statistic: 2.653 on 2 and 50 DF,  p-value: 0.08033
```

```r
model2 <- lm(X1 ~ X4 + I(X5^2) + X5, data = healthdata)
summary(model2)
```

```
##
## Call:
## lm(formula = X1 ~ X4 + I(X5^2) + X5, data = healthdata)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -5.7317 -0.8186  0.1385  1.1268  2.4883
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.414e+01  2.194e+00   6.443 4.85e-08 ***
## X4          -2.291e-01  2.053e-01  -1.116   0.2699
## I(X5^2)      1.012e-04  5.657e-05   1.788   0.0800 .
## X5          -3.732e-02  1.643e-02  -2.271   0.0276 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.578 on 49 degrees of freedom
## Multiple R-squared:  0.1513, Adjusted R-squared:  0.09936
## F-statistic: 2.912 on 3 and 49 DF,  p-value: 0.0436
```

```
#P-Values
p_values <- summary(model)$coefficients[, "Pr(>|t|)"]
p_values
```

```
##  (Intercept)           X2           X3           X4           X5
## 1.946569e-07 2.917340e-01 4.227533e-01 1.655984e-01 5.871691e-02
```

X5 has p-value less than 0.05 so the null hypothesis that the coefficient of X5 is zero can be rejected

```
set.seed(1)
p_values1 <- summary(model1)$coefficients[, "Pr(>|t|)"]
p_values1
```

```
##  (Intercept)           X4           X5
## 1.100475e-07 3.118005e-01 6.111971e-02
```

X5 has p-value close to 0.05 so the null hypothesis that the coefficient of X5 is zero can be rejected

```
set.seed(1)
p_values2 <- summary(model2)$coefficients[, "Pr(>|t|)"]
p_values2
```

```
##  (Intercept)           X4      I(X5^2)           X5
## 4.847752e-08 2.699160e-01 7.995584e-02 2.756492e-02
```

X5 has p-value close to 0.05 so the null hypothesis that the coefficient of X5 is zero can be rejected. Overall it shows in every model that X5 is a significant parameter

```
#Adjusted R-squared
set.seed(1)
adjusted_r_squared <- summary(model)$adj.r.squared
adjusted_r_squared
```

```
## [1] 0.07234595
```

R-squared value of 0.07234595 indicates that the predictors in your model explain about 7.23% of the variability in the response variable.

```
set.seed(1)
adjusted_r_squared1 <- summary(model1)$adj.r.squared
adjusted_r_squared1
```

```
## [1] 0.05978092
```

R-squared value of 0.05978092 indicates that the predictors in your model explain about 5.97% of the variability in the response variable.

```
set.seed(1)
adjusted_r_squared2 <- summary(model2)$adj.r.squared
adjusted_r_squared2
```

```
## [1] 0.09935601
```

R-squared value of 0.09935601 indicates that the predictors in your model explain about 9.93% of the variability in the response variable

```
#Residual Sum of Squares(RSS)
set.seed(1)
rss <- sum(residuals(model)^2)
rss
```

```
## [1] 123.074
```

```
rss1 <- sum(residuals(model1)^2)
rss1
```

```
## [1] 129.9386
```

```
rss2 <- sum(residuals(model2)^2)
rss2
```

```
## [1] 121.9799
```

RSS is minimum for model 1 but it is not significantly lesser than the other two models hence we consider more factors for judging.

```
#Mean Squared Error(MSE)
set.seed(1)
mse <- mean(residuals(model)^2)
mse
```

```
## [1] 2.322151
```

```r
mse1 <- mean(residuals(model1)^2)
mse1
```

```
## [1] 2.451671
```

```r
mse2 <- mean(residuals(model2)^2)
mse2
```

```
## [1] 2.301507
```

Model 1 is giving the least mean square error but it is not significantly lesser than the other two models hence it can't be the only factor in consideration

```r
#F-value/ANOVA
set.seed(1)
anova_table <- anova(model)
f_value <- anova_table$"F value"[1]
f_value
```

```
## [1] 0.7512286
```

```r
anova_table1 <- anova(model1)
f_value1 <- anova_table1$"F value"[1]
f_value1
```

```
## [1] 1.636036
```

```r
anova_table2 <- anova(model2)
f_value2 <- anova_table2$"F value"[1]
f_value2
```

```
## [1] 1.707924
```

```r
#AIC and BIC
aic <- AIC(model)
aic
```

```
## [1] 207.0597
```

```r
aic1 <- AIC(model1)
aic1
```

```
## [1] 205.9363
```

```r
aic2 <- AIC(model2)
aic2
```

```
## [1] 204.5864
```

AIC has the least value for model X1 = X4 + I(X5^2) + X5 The least value of AIC suggests the best fitting model

```r
set.seed(1)
bic <- BIC(model)
bic
```

```
## [1] 218.8814
```

```r
bic1 <- BIC(model1)
bic1
```

```
## [1] 213.8174
```

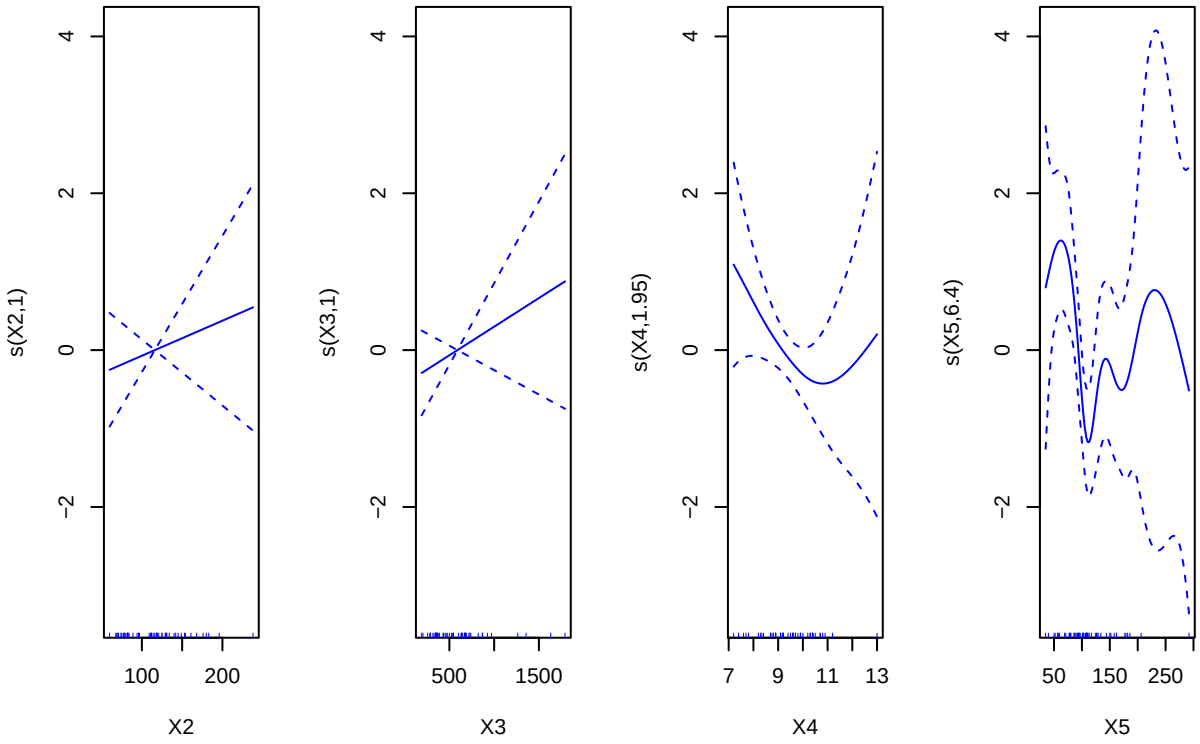
```r
bic2 <- BIC(model2)
bic2
```

```
## [1] 214.4378
```

BIC has the least value for model X1 = X4 + X5 The least value of BIC suggests the best fitting model From AIC and BIC, it is evident that X5 has the strongest influence on the model.

MODEL 4 - GAMs

```r
set.seed(1)
gam.m1 <- gam(X1 ~ s(X2) + s(X3) + s(X4) + s(X5), data = healthdata)
par(mfrow = c(1, 4))
plot(gam.m1, se = TRUE, col = "blue")
```




```
summary(gam.m1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## X1 ~ s(X2) + s(X3) + s(X4) + s(X5)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.306      0.194   47.97   <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(X2) 1.000  1.000 0.481  0.4919
## s(X3) 1.000  1.000 1.164  0.2868
## s(X4) 1.946  2.449 1.491  0.1983
## s(X5) 6.395  7.507 2.614  0.0218 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.278   Deviance explained = 42.2%
## GCV = 2.5379  Scale est. = 1.9948    n = 53
```

```r
pred.m1 <- predict(gam.m1, newdata = healthdata)
rss.m1 <- sum((healthdata$X1 - pred.m1)^2)
rss.m1
```

```
## [1] 83.10062
```

```r
mse.m1 <- mean((healthdata$X1 - pred.m1)^2)
mse.m1
```

```
## [1] 1.567936
```

```r
gam.m2 <- gam(X1 ~ s(X2) + s(X3) + te(X4, X5), data = healthdata)
par(mfrow = c(1, 4))
plot(gam.m2, se = TRUE, col = "blue")
summary(gam.m2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## X1 ~ s(X2) + s(X3) + te(X4, X5)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.3057     0.1659   56.09   <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(X2)     1.000  1.000 0.302    0.586
## s(X3)     1.000  1.000 0.955    0.334
## te(X4,X5) 8.648  8.947 6.183 3.97e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.472   Deviance explained =   58%
## GCV = 1.8701  Scale est. = 1.4591    n = 53
```

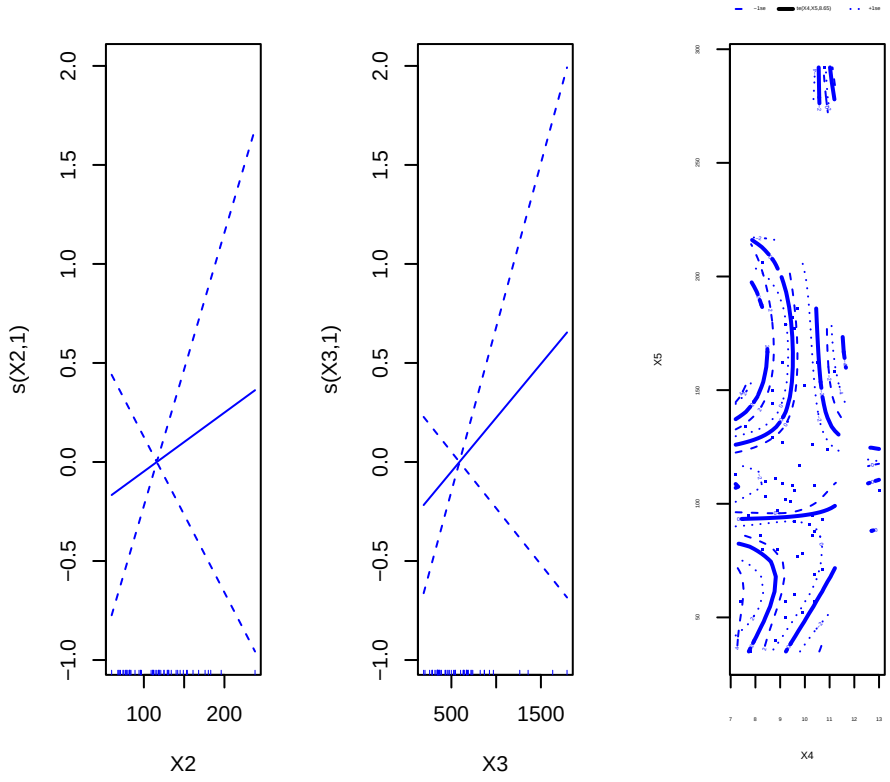
```r
pred.m2 <- predict(gam.m2, newdata = healthdata)
rss.m2 <- sum((healthdata$X1 - pred.m2)^2)
rss.m2
```

```
## [1] 60.33665
```

```r
mse.m2 <- mean((healthdata$X1 - pred.m2)^2)
mse.m2
```

```
## [1] 1.138427
```

```r
gam.m3 <- gam(X1 ~  te(X2, X3) + te(X4, X5), data = healthdata)
par(mfrow = c(1, 4))
```




```r
plot(gam.m3, se = TRUE, col = "blue")
summary(gam.m3)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## X1 ~ te(X2, X3) + te(X4, X5)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.306      0.168    55.4   <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
## te(X2,X3) 3.000  3.000 0.571    0.637
## te(X4,X5) 8.636  8.944 5.911 6.06e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## R-sq.(adj) =  0.459   Deviance explained =   58%
## GCV = 1.9635  Scale est. = 1.4954    n = 53
```

```
pred.m3 <- predict(gam.m3, newdata = healthdata)
rss.m3 <- sum((healthdata$X1 - pred.m3)^2)
rss.m3
```

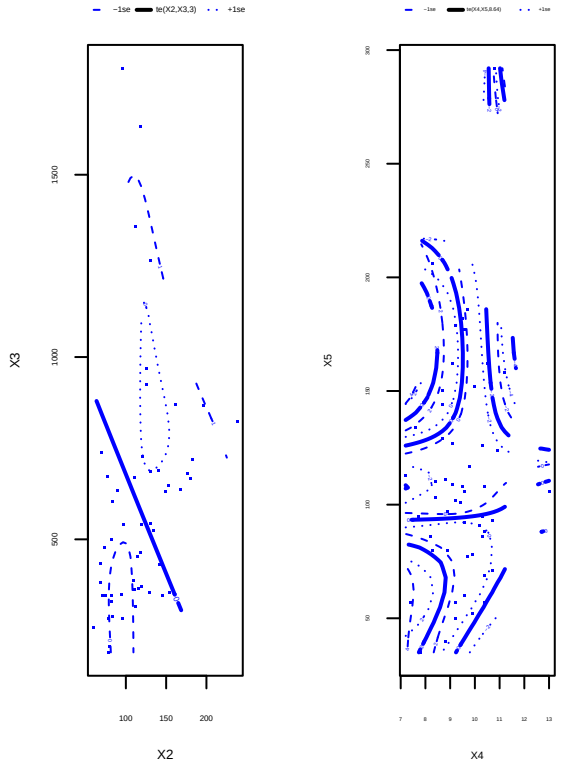
```
## [1] 60.35785
```

```
mse.m3 <- mean((healthdata$X1 - pred.m3)^2)
mse.m3
```

```
## [1] 1.138827
```

```
anova(gam.m1, gam.m2, gam.m3, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: X1 ~ s(X2) + s(X3) + s(X4) + s(X5)
## Model 2: X1 ~ s(X2) + s(X3) + te(X4, X5)
## Model 3: X1 ~ te(X2, X3) + te(X4, X5)
##   Resid. Df Resid. Dev       Df Deviance F Pr(>F)
## 1    40.044     83.101
## 2    41.053     60.337 -1.00830  22.7640
## 3    40.056     60.358  0.99723  -0.0212
```



GAM-Model 2 has a significantly lower deviance compared to Model 1 (Deviance = 22.7640, $p < 0.05$). GAM-Model 3 has a slightly lower deviance compared to Model 2, but the difference is not statistically significant ($p > 0.05$).

Conclusion: GAM-sModel 2 provides a better fit than Model 1. Model 3, which includes tensor product smooth terms, does not show a significant improvement over Model 2. Therefore, Model 2 might be the preferred model among the three based on the provided results.

MODEL 5 - Polynomial Regression

```
set.seed(1)
X <- healthdata[, -1]
Y <- healthdata$X1
attach(healthdata)
train <- sample(1:nrow(healthdata),0.9*nrow(healthdata))
test <- (-train)
healthdata_train <- healthdata[train,]
healthdata_test <- healthdata[-train,]

degree <- 2
fit <- lm(X1 ~ poly(X2, degree) + poly(X3, degree) + poly(X4, degree) + poly(X5,
degree), data = healthdata_train)
summary(fit)
```

```
##
## Call:
## lm(formula = X1 ~ poly(X2, degree) + poly(X3, degree) + poly(X4,
##     degree) + poly(X5, degree), data = healthdata_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1668 -0.5336  0.2675  0.7577  2.6044
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)          9.3617     0.2339  40.016   <2e-16 ***
## poly(X2, degree)1    0.8609     2.2790   0.378   0.7077
## poly(X2, degree)2    0.2237     1.6558   0.135   0.8933
## poly(X3, degree)1    1.6250     1.8071   0.899   0.3742
## poly(X3, degree)2   -1.7217     1.9887  -0.866   0.3921
## poly(X4, degree)1   -1.9610     1.8744  -1.046   0.3021
## poly(X4, degree)2    1.3229     1.6588   0.797   0.4301
## poly(X5, degree)1   -4.3323     1.7042  -2.542   0.0152 *
## poly(X5, degree)2    1.9133     1.6748   1.142   0.2604
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.604 on 38 degrees of freedom
## Multiple R-squared:  0.2681, Adjusted R-squared:  0.114
## F-statistic:  1.74 on 8 and 38 DF,  p-value: 0.1207
```

Since the Adjusted R-squared is 0.114 so this model may not be the best fit. p-value is 0.1207 which is

greater than 0.05 which indicates one of the predictor variable which is X5 is significantly related to response variable.

```
set.seed(1)
coef(fit)
```

```
##       (Intercept) poly(X2, degree)1 poly(X2, degree)2 poly(X3, degree)1
##         9.3617022         0.8608878         0.2236735         1.6250190
## poly(X3, degree)2 poly(X4, degree)1 poly(X4, degree)2 poly(X5, degree)1
##        -1.7216922        -1.9610040         1.3228581        -4.3323245
## poly(X5, degree)2
##         1.9133321
```

```
pred =  predict(fit, newdata = healthdata_test)
length(pred)
```

```
## [1] 6
```

```
mse <- mean((Y[test]-pred)^2)
mse
```

```
## [1] 2.866654
```

MODEL 6 - Model selection

```
set.seed(1)
attach(healthdata)
```

```
## The following objects are masked from healthdata (pos = 3):
##
##     X1, X2, X3, X4, X5
```

```
# Fit model for all the variables and check summary
regfit.full <- regsubsets(X1~., healthdata)
reg.summary <- summary(regfit.full)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(X1 ~ ., healthdata)
## 4 Variables  (and intercept)
##    Forced in Forced out
## X2     FALSE      FALSE
## X3     FALSE      FALSE
## X4     FALSE      FALSE
## X5     FALSE      FALSE
## 1 subsets of each size up to 4
## Selection Algorithm: exhaustive
##          X2  X3  X4  X5
## 1  ( 1 ) " " " " " " "*"
## 2  ( 1 ) " " "*" " " "*"
## 3  ( 1 ) "*" " " "*" "*"
## 4  ( 1 ) "*" "*" "*" "*"
```

The summary suggests that a single variable model has only X5 and a 2 variable model has X3 and X5.

```
which.max(reg.summary$adjr2)
```

```
## [1] 3
```

```
which.min(reg.summary$rss)
```

```
## [1] 4
```

```
which.min(reg.summary$cp)
```

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

```
which.min(reg.summary$bic)
```

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

The adjusted R2 value is maximum for a 3 variable model and the RSS is minimum for a model which includes all 4 variables. The statistics BIC and Cp suggests that a one variable model with X5 is the best.

Since all 4 statistics select different number of coefficients (3,4,1), we are fitting the model with 3 predictors which has the best value of R2.

```
num_of_coefficients <- which.max(reg.summary$adjr2)
coef(regfit.full,num_of_coefficients)
```

```
##  (Intercept)           X2           X4           X5
## 12.565899606  0.009284091 -0.359139896 -0.008579780
```

This selects X2, X4 and X5 as the 3 significant predictors

```
# Do forward and backward selection on the model
set.seed(1)
regfit.fwd <- regsubsets(X1~., data = healthdata, method="forward")
reg.summary <- summary(regfit.fwd)
which.max(reg.summary$adjr2)
```

```
## [1] 4
```

```
which.min(reg.summary$rss)
```

```
## [1] 4
```

```
which.min(reg.summary$cp)
```

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

```
which.min(reg.summary$bic)
```

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

Here the R2 value is maximum for a model fit with all 4 variables.

```
set.seed(1)
regfit.bwd <- regsubsets(X1~., data = healthdata, method = "backward")
reg.summary <- summary(regfit.bwd)
which.max(reg.summary$adjr2)
```

```
## [1] 3
```

```
which.min(reg.summary$rss)
```

```
## [1] 4
```

```
which.min(reg.summary$cp)
```

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

```
which.min(reg.summary$bic)
```

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

```
num_of_coefficients <- which.max(reg.summary$adjr2)
coef(regfit.bwd,num_of_coefficients)
```

```
##  (Intercept)           X2           X4           X5
## 12.565899606  0.009284091 -0.359139896 -0.008579780
```

This selects X2, X4 and X5 as the 3 significant predictors For a 3 variable model, X4 seems to be the most significant predictor with a negative relationship with the output.

```
# Split into training and test set
set.seed(1)
train <- sample(1:nrow(healthdata),0.7*nrow(healthdata))
test <- (-train)
Health_train <- healthdata[train,]
Health_test <- healthdata[-train,]
# Since best subset and backward selection gave the same set of predictors (X2,
# X4 and X5),use it to calculate cross validation errors with 5 fold and 10 fold
regfit.best <- regsubsets(X1~X2+X4+X5,data = healthdata[train,])
test.mat<-model.matrix (X1~X2+X4+X5, data = healthdata[test,])
val.errors <- rep (NA, num_of_coefficients)
for (i in 1:num_of_coefficients) {
  coefi <- coef(regfit.best , id = i)
  pred <- test.mat[,names(coefi)] %*% coefi
  val.errors[i] <- mean((healthdata$X1[test] - pred)^2)
}
val.errors
```

```
## [1] 2.229449 2.635484 2.472440
```

```
num_of_predictors_mse <- which.min(val.errors)
num_of_predictors_mse
```

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

MSE value is similar among the 3 combinations and is minimum for a 1 variable model- 2.229449

```
coef(regfit.best , num_of_predictors_mse)
```

```
## (Intercept)          X5
## 11.27684387 -0.01838455
```

The validation set approach shows that X5 is more significant in predicting the response; coef of X5= -0.01838455, but has a very weak relationship with the output.

```
# Predict function to predict the output X1 for k fold

predict.regsubsets <- function(object,newdata,id,...){
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form,newdata)
  coefi <- coef(object,id=id)
  xvars <- names(coefi)
  mat[,xvars]%*%coefi
}

# Model evaluation with 10 fold
k <- 10
n <- nrow(Health)
set.seed(1)
folds <- sample(rep(1:k, length = n))
cv.errors <- matrix(NA, k, num_of_coefficients,dimnames = list (NULL ,
                        paste (1:num_of_coefficients)))

for  (j in 1:k){
  best.fit <- regsubsets(X1~X2+X4+X5,data=healthdata[folds!=j,])
  for (i in 1:num_of_coefficients){
    pred <- predict(regfit.best,healthdata[folds==j,],id=i)
    cv.errors[j,i] <- mean((healthdata$X1[folds==j]-pred)^2)
  }
}
cv.errors
```

```
##              1        2        3
##  [1,] 3.114718 3.1782002 3.0406899
##  [2,] 1.455923 1.7781520 1.6531366
##  [3,] 2.677260 1.8664970 2.0407692
##  [4,] 4.894759 5.0326880 4.7032321
##  [5,] 3.080389 3.0047698 2.5416998
##  [6,] 1.080337 1.2540277 0.9249597
##  [7,] 1.032552 0.8738002 0.6627030
```

```
##  [8,] 1.397938 1.2541326 1.1262798
##  [9,] 7.070137 7.9269354 7.6624625
## [10,] 1.031733 1.0073374 1.5494431
```

```
# Display the cv errors matrix and the mean of errors
mean.cv.errors <- apply(cv.errors , 2, mean)
mean.cv.errors
```

```
##        1        2        3
## 2.683575 2.717654 2.590538
```

```
min(mean.cv.errors)
```

```
## [1] 2.590538
```

The minimum of the mean errors and it shows that the 3 variable model has the least cv error.

```
# Model evaluation with 5 fold
k <- 5
n <- nrow(healthdata)
set.seed(1)
folds <- sample(rep(1:k, length = n))
cv.errors <- matrix(NA, k, num_of_coefficients,dimnames = list (NULL ,
                        paste (1:num_of_coefficients)))

for  (j in 1:k){
  best.fit <- regsubsets(X1~.,data=healthdata[folds!=j,])
  for (i in 1:num_of_coefficients){
    pred <- predict(regfit.best,healthdata[folds==j,],id=i)
    cv.errors[j,i] <- mean((healthdata$X1[folds==j]-pred)^2)
  }
}
cv.errors
```

```
##             1        2        3
## [1,] 2.189999 2.303576 2.078994
## [2,] 1.263482 1.367083 1.202940
## [3,] 2.095750 1.588150 1.625092
## [4,] 5.982448 6.479812 6.182847
## [5,] 2.056061 2.006054 2.045571
```

```
mean.cv.errors <- apply (cv.errors , 2, mean)
mean.cv.errors
```

```
##        1        2        3
## 2.717548 2.748935 2.627089
```

```
min(mean.cv.errors)
```

```
## [1] 2.627089
```

The display of minimum of the mean errors and it shows that the 3 variable model has the least cv error.

FINAL CONCLUSION (Health Dataset)

Conclusion and Best Model Recommendation Consistency Across Models: X5 frequently emerges as a significant predictor across multiple models, especially in Lasso, Multiple Linear Regression, and the 3-variable model selection.

Predictive Accuracy: Ridge Regression shows the lowest MSE, especially in the LOOCV, indicating strong predictive accuracy.

GAMs Performance: GAM Model 2 (s(X2) + s(X3) + te(X4, X5)) shows a very strong adjusted R-squared value, suggesting good explanatory power, along with a low MSE.

Best Model: Considering both predictive accuracy and explanatory power, GAM Model 2 appears to be the best model. It balances a low MSE with a relatively high adjusted R-squared. While Ridge Regression has a slightly lower MSE, the better explanatory power of GAM Model 2 (adjusted R-squared of 0.47) makes it more favorable, especially in a health context where understanding the relationship between variables can be as crucial as prediction accuracy.

Final Thoughts: The choice of the best model can also depend on the specific context of the health data and the practical implications of the predictors involved. If the primary goal is prediction, Ridge Regression could be more appropriate. However, for a balance of understanding and predicting the outcomes, GAM Model 2 is recommended.

REAL ESTATE VALUATION

REAL ESTATE DATASET DESCRIPTION:

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)

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

```
RealEstate_Data <- read_excel("RealEstate.xlsx")
attach(RealEstate_Data)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     Y
```

```
## The following objects are masked from healthdata (pos = 3):
##
##     X1, X2, X3, X4, X5
```

```
## The following objects are masked from healthdata (pos = 4):
##
##     X1, X2, X3, X4, X5
```

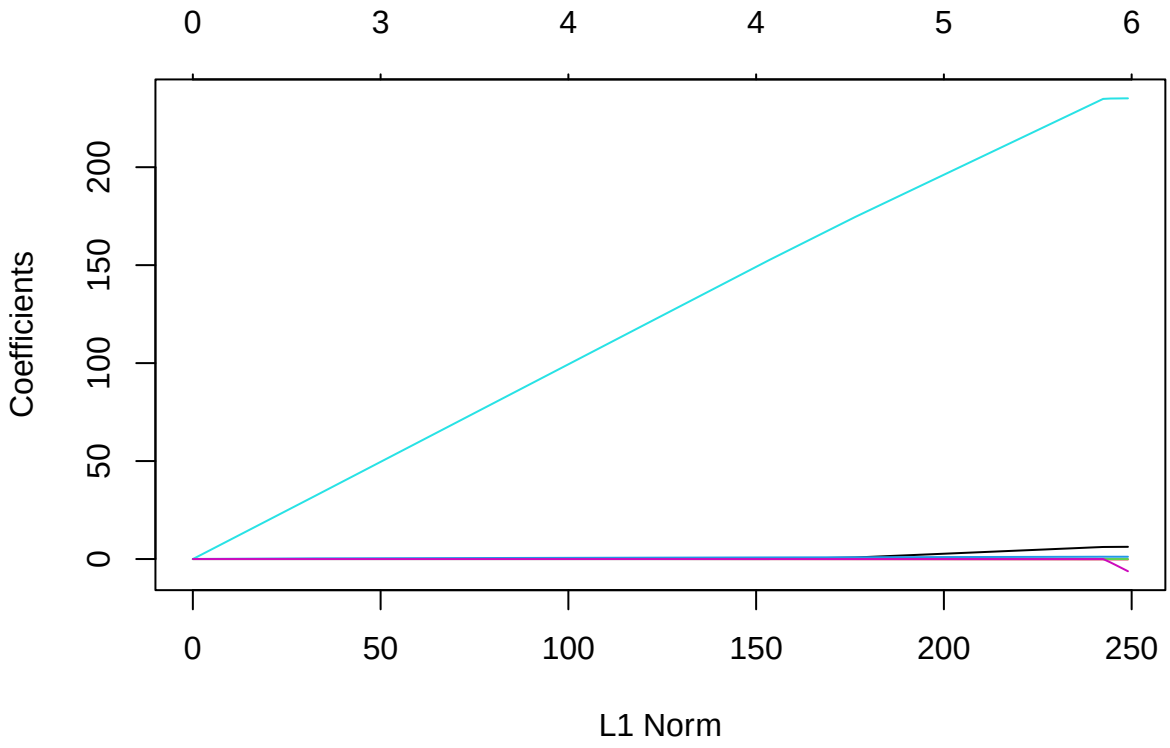
```r
# Remove the index column
RealEstate_Data$No <- NULL
head(RealEstate_Data)
```

```
## # A tibble: 6 x 7
##      X1    X2     X3    X4    X5    X6     Y
##   <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2013.  32     84.9    10  25.0  122.  37.9
## 2 2013.  19.5  307.      9  25.0  122.  42.2
## 3 2014.  13.3  562.      5  25.0  122.  47.3
## 4 2014.  13.3  562.      5  25.0  122.  54.8
## 5 2013.   5    391.      5  25.0  122.  43.1
## 6 2013.   7.1 2175.      3  25.0  122.  32.1
```

MODEL 1 - The Lasso

```r
set.seed (1)
x <- model.matrix (Y ~ ., RealEstate_Data)[, -1]
y <- RealEstate_Data$Y
grid <- 10 ^ seq (10, -2, length = 100)

# Perform Lasso Regression
train <- sample (1: nrow (x), nrow (x) / 1.25)
test <- (-train)
y.test <- y[test]
lasso.mod <- glmnet (x[train,], y[train], alpha = 1, lambda = grid)
plot (lasso.mod)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```




```
# Evaluate using K-Fold Cross-Validation
# 10 fold cross validation- select best lambda value and find MSE
cv.out <- cv.glmnet(x[train , ], y[train], alpha = 1, nfolds = 10)

bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.1289491
```

```
lasso.pred <- predict (lasso.mod, s = bestlam, newx = x[test,])

mse <- mean ((lasso.pred - y.test)^2)
mse
```

```
## [1] 57.48314
```

```
out <- glmnet (x, y, alpha = 0, lambda = grid)
lasso.coef <- predict (out , type = "coefficients", s = bestlam)
lasso.coef
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept) -1.538176e+04
## X1           5.078100e+00
```

```
## X2          -2.670597e-01
## X3          -4.350165e-03
## X4           1.134376e+00
## X5           2.269310e+02
## X6          -3.823479e+00
```

X1, X4, X5 and X6 are significant in terms of co-efficient values. There is a positive association of X4 with the dependent variable and a relatively large positive effect of X5.

```
# Evaluate using LOOCV
# Select best lambda value and find MSE
set.seed (1)
cv.out <- cv.glmnet(x[train , ], y[train], alpha = 1,
                    nfolds = nrow(RealEstate_Data))
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.05581899
```

```
lasso.pred <- predict (lasso.mod, s = bestlam, newx = x[test,])

mse <- mean ((lasso.pred - y.test)^2)
mse
```

```
## [1] 57.72051
```

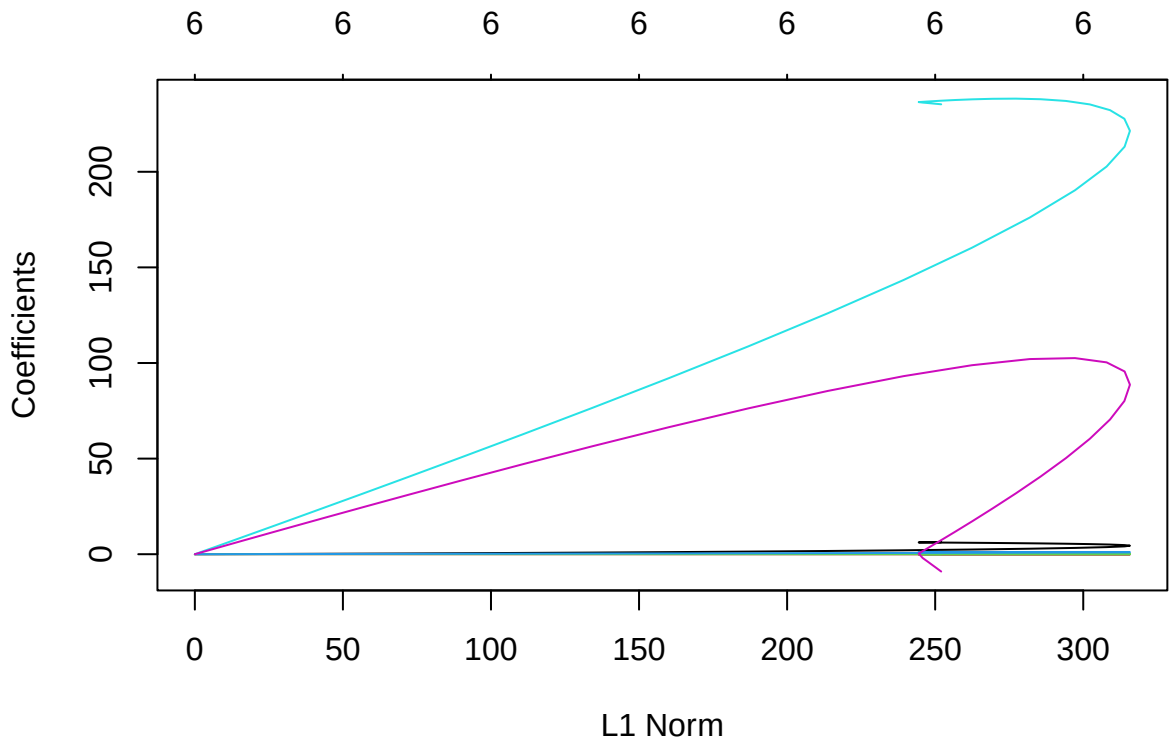
```
out <- glmnet (x, y, alpha = 1, lambda = grid)
lasso.coef <- predict (out , type = "coefficients", s = bestlam)
lasso.coef
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept) -1.549776e+04
## X1           4.936970e+00
## X2          -2.641313e-01
## X3          -4.338239e-03
## X4           1.125162e+00
## X5           2.243440e+02
## X6           .
```

X4 and X5 shows relatively positive effect and is significant. X6 is indicating that its coefficient has been shrunk to zero. This means X6 is not contributing to the model and can be considered as not having a significant relationship with the dependent variable in this specific model context.

MODEL 2 - Ridge Regression

```
set.seed (1)
x <- model.matrix (Y ~ ., RealEstate_Data)[, -1]
y <- RealEstate_Data$Y
grid <- 10 ^ seq (10, -2, length = 100)

# Perform Ridge Regression
train <- sample (1: nrow (x), nrow (x) / 1.25)
test <- (-train)
y.test <- y[test]
ridge.mod <- glmnet (x[train , ], y[train], alpha = 0, lambda = grid)
plot (ridge.mod)
```




```
# Evaluate using K-Fold Cross-Validation
# 10 fold cross validation- select best lambda value and find MSE
cv.out <- cv.glmnet(x[train , ], y[train], alpha = 0, nfolds = 10)

bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.9311168
```

```
ridge.pred <- predict (ridge.mod, s = bestlam, newx = x[test,])

mse <- mean ((ridge.pred - y.test)^2)
mse
```

```
## [1] 57.49682
```

```
out <- glmnet (x, y, alpha = 0, lambda = grid)
ridge.coef <- predict (out , type = "coefficients", s = bestlam)
ridge.coef
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept) -1.909331e+04
## X1           4.712895e+00
## X2          -2.515495e-01
## X3          -3.776267e-03
## X4           1.123716e+00
## X5           2.301140e+02
## X6           3.210440e+01
```

X1, X5 and X6 indicates a substantial increase in the dependent variable signifying a strong positive relationship with the dependent variable.

```
# Evaluate using LOOCV
# Select best lambda value and find MSE
cv.out <- cv.glmnet(x[train , ], y[train], alpha = 0,
                    nfolds = nrow(RealEstate_Data[train,]))
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.9311168
```

```
ridge.pred <- predict (ridge.mod, s = bestlam, newx = x[test,])
mse <- mean ((ridge.pred - y.test)^2)
mse
```

```
## [1] 57.49682
```

```
out <- glmnet (x, y, alpha = 0, lambda = grid)
ridge.coef <- predict (out , type = "coefficients", s = bestlam)
ridge.coef
```

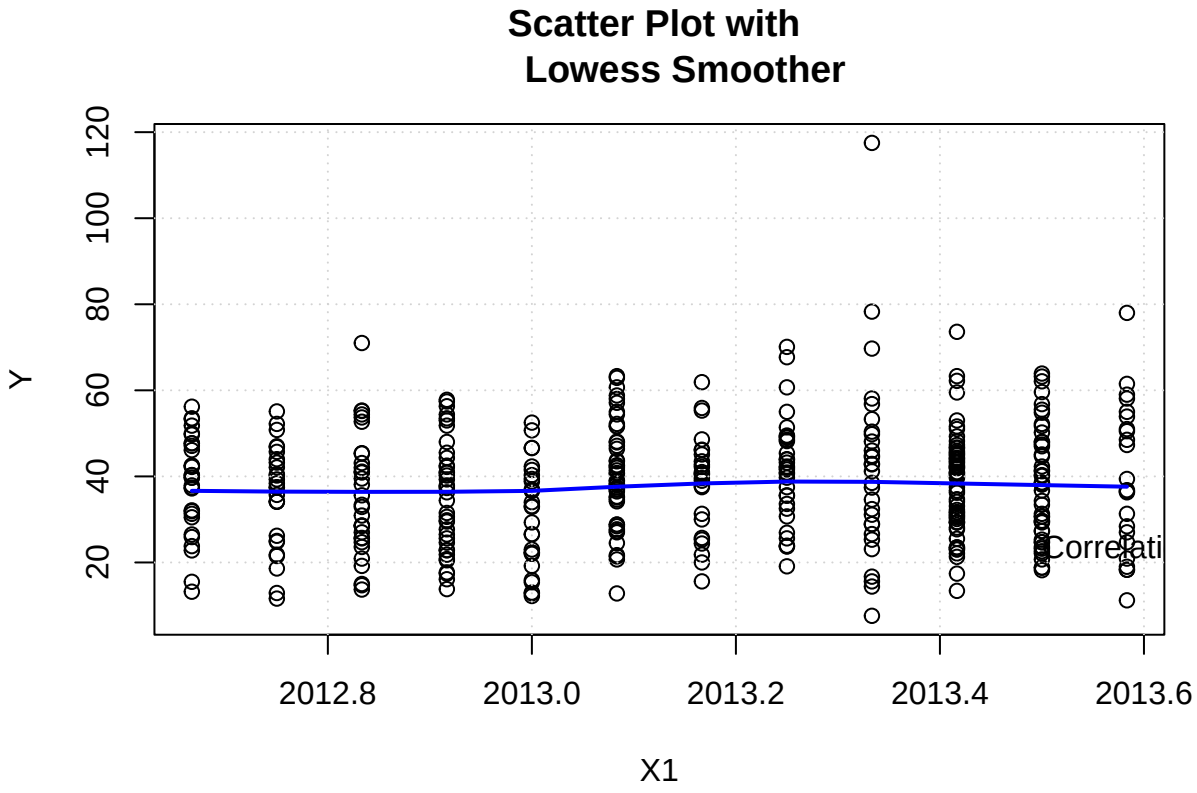
```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept) -1.909331e+04
## X1           4.712895e+00
## X2          -2.515495e-01
## X3          -3.776267e-03
## X4           1.123716e+00
## X5           2.301140e+02
## X6           3.210440e+01
```

X1, X4, X5 and X6 are significantly influencing the output as the per the coefficient values.

```
# Correlation
plot(RealEstate_Data$X1, RealEstate_Data$Y, main = "Scatter Plot with
     Lowess Smoother", xlab = "X1", ylab = "Y")
lines(lowess(RealEstate_Data$X1, RealEstate_Data$Y), col = "blue", lwd = 2)
cor_value <- cor(RealEstate_Data$X1, RealEstate_Data$Y)
cor_value
```

```
## [1] 0.08752927
```

```
text(quantile(RealEstate_Data$X1, 0.9), quantile(RealEstate_Data$Y, 0.1),
     paste("Correlation =", round(cor_value, 2)), adj = c(0, 0))
grid()
```

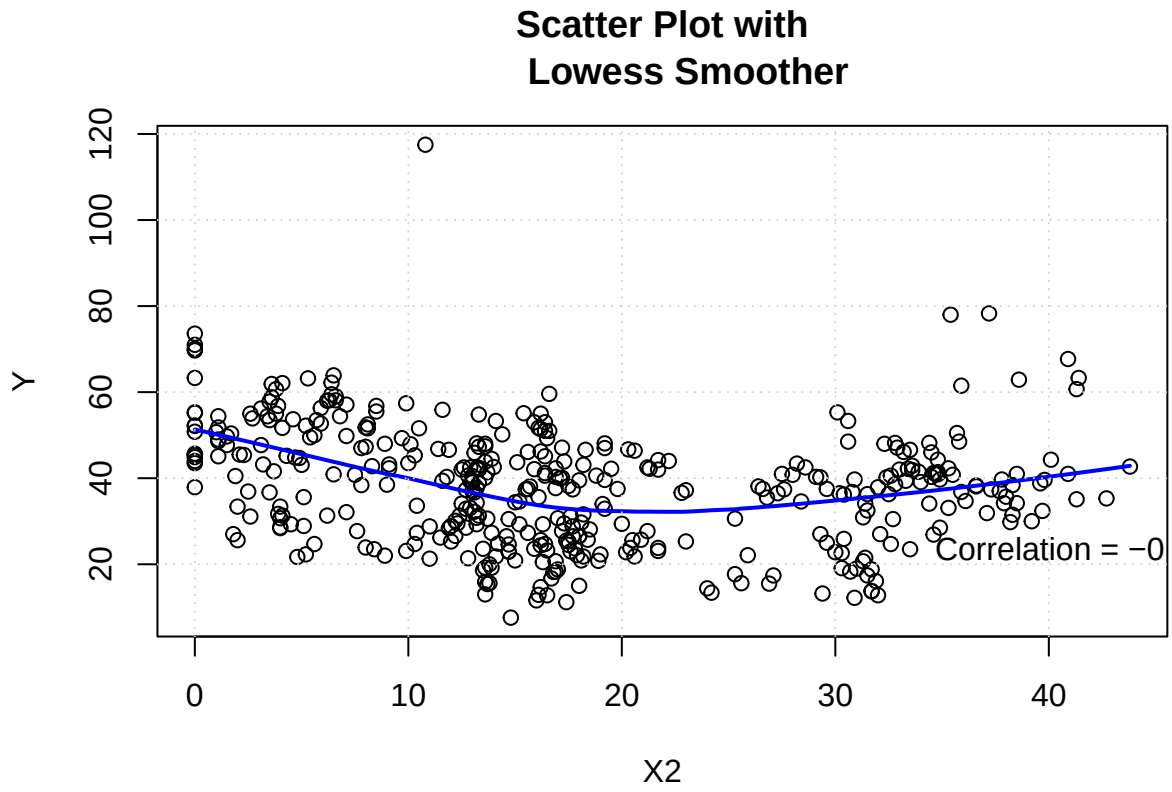


Correlation of Y and X2 = 0.0875

```
plot(RealEstate_Data$X2, RealEstate_Data$Y, main = "Scatter Plot with
     Lowess Smoother", xlab = "X2", ylab = "Y")
lines(lowess(RealEstate_Data$X2, RealEstate_Data$Y), col = "blue", lwd = 2)
cor_value <- cor(RealEstate_Data$X2, RealEstate_Data$Y)
cor_value
```

```
## [1] -0.210567
```

```
text(quantile(RealEstate_Data$X2, 0.9), quantile(RealEstate_Data$Y, 0.1)
     , paste("Correlation =", round(cor_value, 2)), adj = c(0, 0))
grid()
```




```
cor_value
```

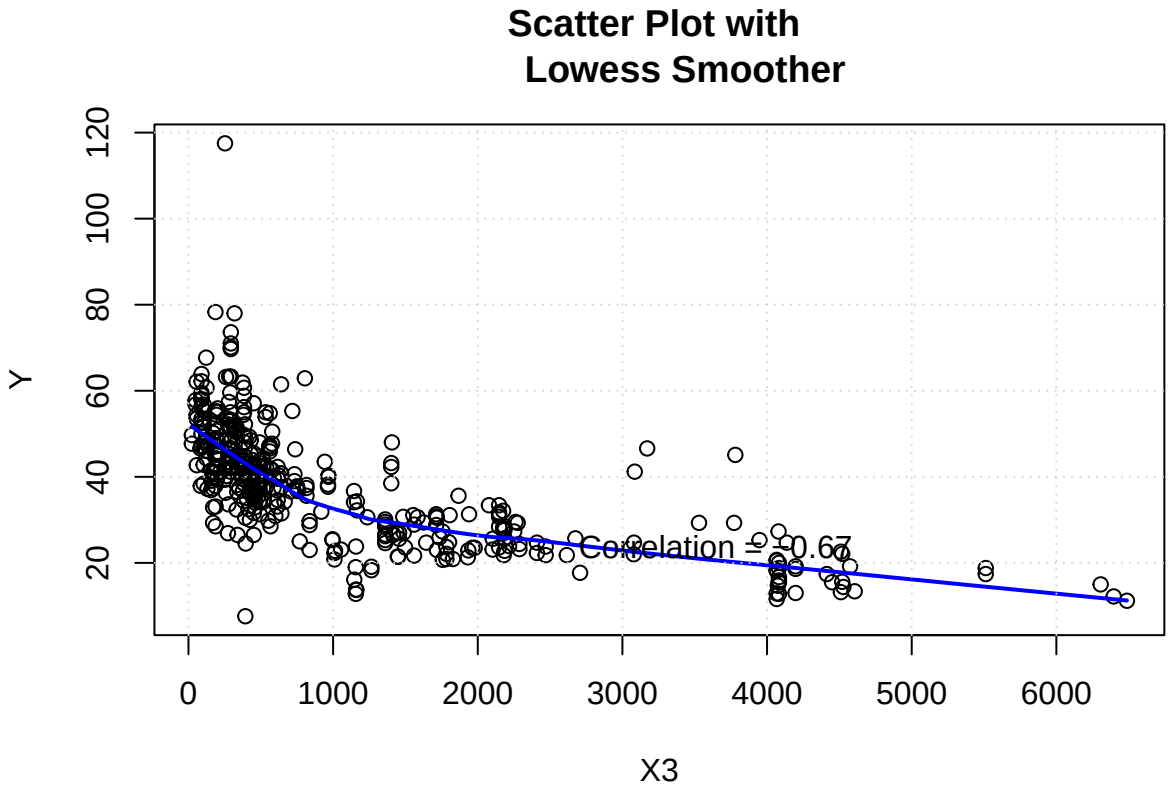
```
## [1] -0.210567
```

Correlation of Y and X2 = -0.21

```
plot(RealEstate_Data$X3, RealEstate_Data$Y, main = "Scatter Plot with
     Lowess Smoother", xlab = "X3", ylab = "Y")
lines(lowess(RealEstate_Data$X3, RealEstate_Data$Y), col = "blue", lwd = 2)
cor_value <- cor(RealEstate_Data$X3, RealEstate_Data$Y)
cor_value
```

```
## [1] -0.6736129
```

```
text(quantile(RealEstate_Data$X3, 0.9), quantile(RealEstate_Data$Y, 0.1),
     paste("Correlation =", round(cor_value, 2)), adj = c(0, 0))
grid()
```

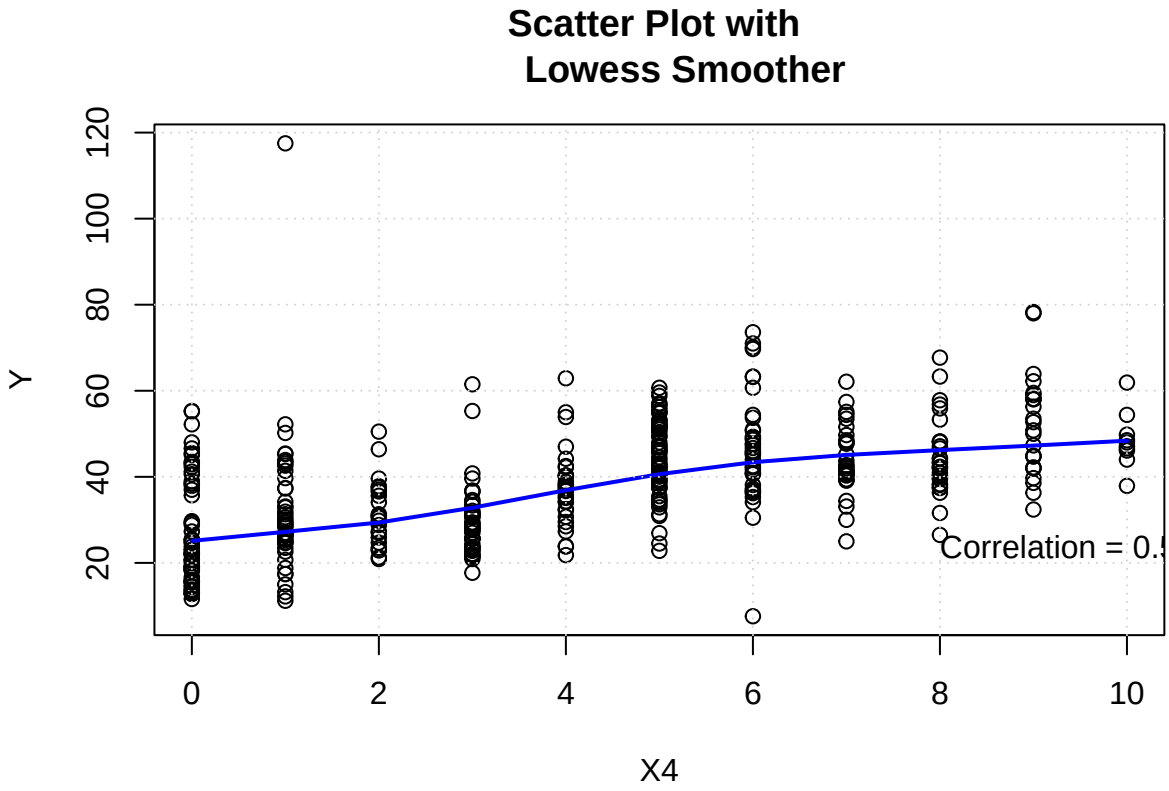


Correlation of Y and X3 = -0.67

```
plot(RealEstate_Data$X4, RealEstate_Data$Y, main = "Scatter Plot with
     Lowess Smoother", xlab = "X4", ylab = "Y")
lines(lowess(RealEstate_Data$X4, RealEstate_Data$Y), col = "blue", lwd = 2)
cor_value <- cor(RealEstate_Data$X4, RealEstate_Data$Y)
cor_value
```

```
## [1] 0.5710049
```

```
text(quantile(RealEstate_Data$X4, 0.9), quantile(RealEstate_Data$Y, 0.1),
     paste("Correlation =", round(cor_value, 2)), adj = c(0, 0))
grid()
```

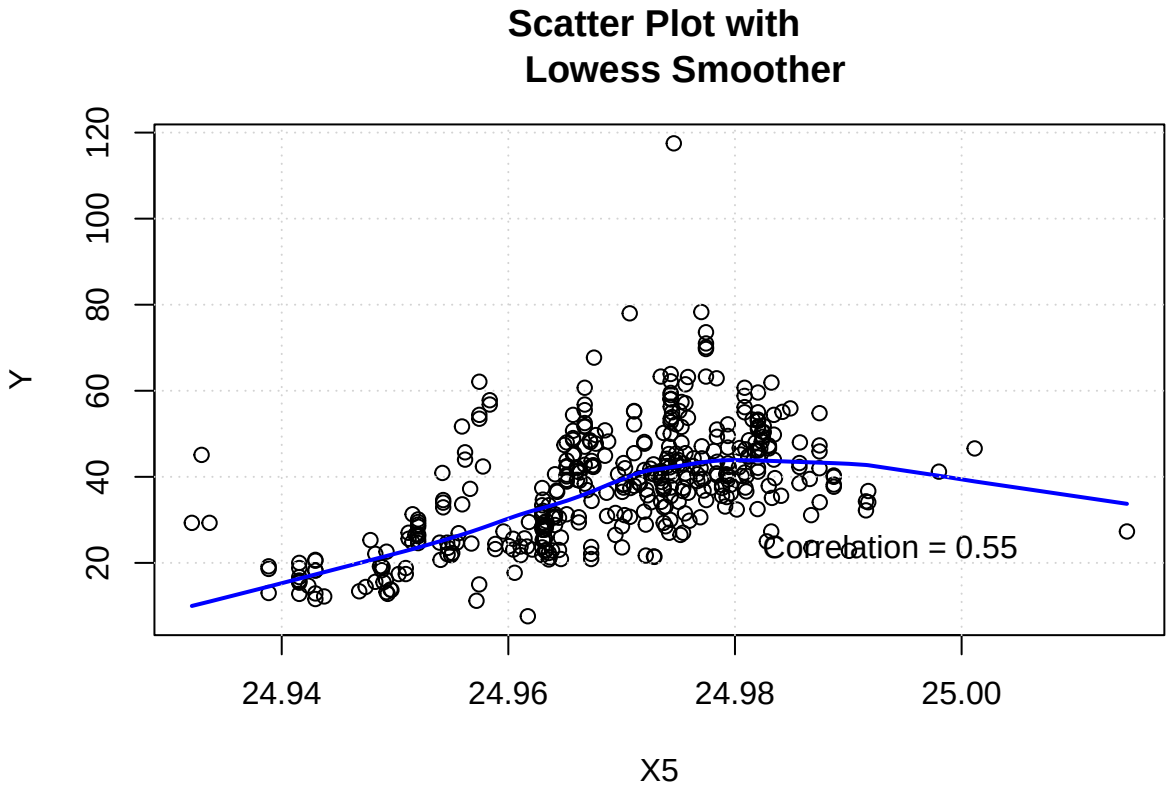


Correlation of Y and X4 = 0.57

```
plot(RealEstate_Data$X5, RealEstate_Data$Y, main = "Scatter Plot with
    Lowess Smoother", xlab = "X5", ylab = "Y")
lines(lowess(RealEstate_Data$X5, RealEstate_Data$Y), col = "blue", lwd = 2)
cor_value <- cor(RealEstate_Data$X5, RealEstate_Data$Y)
cor_value
```

```
## [1] 0.5463067
```

```
text(quantile(RealEstate_Data$X5, 0.9), quantile(RealEstate_Data$Y, 0.1),
    paste("Correlation =", round(cor_value, 2)), adj = c(0, 0))
grid()
```



Correlation of Y and X5 = 0.546

MODEL 3 - Multiple Linear Regression

```
set.seed(1)
model <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6, data = RealEstate_Data)
summary(model)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6, data = RealEstate_Data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.667  -5.412  -0.967   4.217  75.190
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.444e+04  6.775e+03  -2.132  0.03364 *
## X1           5.149e+00  1.557e+00   3.307  0.00103 **
## X2          -2.697e-01  3.853e-02  -7.000 1.06e-11 ***
## X3          -4.488e-03  7.180e-04  -6.250 1.04e-09 ***
## X4           1.133e+00  1.882e-01   6.023 3.83e-09 ***
## X5           2.255e+02  4.457e+01   5.059 6.38e-07 ***
```

```
## X6          -1.243e+01  4.858e+01  -0.256  0.79820
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.858 on 407 degrees of freedom
## Multiple R-squared:  0.5824, Adjusted R-squared:  0.5762
## F-statistic:  94.6 on 6 and 407 DF,  p-value: < 2.2e-16
```

```
adjusted_r_squared <- summary(model)$adj.r.squared
adjusted_r_squared
```

```
## [1] 0.5762286
```

R-squared value of 0.5762286 indicates that the predictors in your model explain about 57.62% of the variability in the response variable

```
set.seed(1)
model1 <- lm(Y ~ X2 + X3 + X4, data = RealEstate_Data)
summary(model1)
```

```
##
## Call:
## lm(formula = Y ~ X2 + X3 + X4, data = RealEstate_Data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.304  -5.430  -1.738   4.325  77.315
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.977286   1.384542  31.041  < 2e-16 ***
## X2          -0.252856   0.040105  -6.305 7.47e-10 ***
## X3          -0.005379   0.000453 -11.874  < 2e-16 ***
## X4           1.297443   0.194290   6.678 7.91e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.251 on 410 degrees of freedom
## Multiple R-squared:  0.5411, Adjusted R-squared:  0.5377
## F-statistic: 161.1 on 3 and 410 DF,  p-value: < 2.2e-16
```

```
adjusted_r_squared1 <- summary(model1)$adj.r.squared
adjusted_r_squared1
```

```
## [1] 0.5377052
```

R-squared value of 0.5377052 indicates that the predictors in your model explain about 53.77% of the variability in the response variable.

```r
set.seed(1)
model2 <- lm(Y ~ X2 + I(X6^2) + I(X4^2) + I(X5^2), data = RealEstate_Data)
summary(model2)
```

```
##
## Call:
## lm(formula = Y ~ X2 + I(X6^2) + I(X4^2) + I(X5^2), data = RealEstate_Data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.194  -5.447  -1.070   4.170  79.979
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.767e+04  2.014e+03  -8.773  < 2e-16 ***
## X2          -2.880e-01  4.086e-02  -7.049 7.73e-12 ***
## I(X6^2)      8.819e-01  1.432e-01   6.159 1.76e-09 ***
## I(X4^2)      1.604e-01  1.956e-02   8.197 3.21e-15 ***
## I(X5^2)      7.510e+00  8.452e-01   8.885  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.382 on 409 degrees of freedom
## Multiple R-squared:  0.5292, Adjusted R-squared:  0.5246
## F-statistic: 114.9 on 4 and 409 DF,  p-value: < 2.2e-16
```

```r
adjusted_r_squared2 <- summary(model2)$adj.r.squared
adjusted_r_squared2
```

```
## [1] 0.5245754
```

R-squared value of 0.5245754 indicates that the predictors in your model explain about 52.45% of the variability in the response variable

```r
set.seed(1)
model3 <- lm(Y ~ X2 + X3 + I(X6^2) + I(X4^2) + I(X5^2), data = RealEstate_Data)
summary(model3)
```

```
##
## Call:
## lm(formula = Y ~ X2 + X3 + I(X6^2) + I(X4^2) + I(X5^2), data = RealEstate_Data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.791  -5.087  -1.301   3.873  75.362
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.549e+03  3.090e+03  -0.501    0.616
## X2          -2.766e-01  3.890e-02  -7.110 5.23e-12 ***
## X3          -4.635e-03  6.971e-04  -6.649 9.48e-11 ***
```

```
## I(X6^2)     -1.003e-01  2.009e-01  -0.499    0.618
## I(X4^2)      1.249e-01  1.936e-02   6.451 3.14e-10 ***
## I(X5^2)      4.932e+00  8.924e-01   5.527 5.83e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.922 on 408 degrees of freedom
## Multiple R-squared:  0.5752, Adjusted R-squared:   0.57
## F-statistic: 110.5 on 5 and 408 DF,  p-value: < 2.2e-16
```

```
adjusted_r_squared3 <- summary(model3)$adj.r.squared
adjusted_r_squared3
```

```
## [1] 0.5700064
```

R-squared value of 0.5700064 indicates that the predictors in your model explain about 57% of the variability in the response variable

```
# P-Value
p_values <- summary(model)$coefficients[, "Pr(>|t|)"]
p_values
```

```
##  (Intercept)           X1           X2           X3           X4           X5
## 3.364344e-02 1.025782e-03 1.063915e-11 1.037344e-09 3.826895e-09 6.382166e-07
##           X6
## 7.982028e-01
```

The small p-values for X1, X2, X3, X4, and X5 suggest that these predictors are likely significant in predicting Y X1 has higher p value which suggests it does not have a significant impact on Y

```
p_values1 <- summary(model1)$coefficients[, "Pr(>|t|)"]
p_values1
```

```
##   (Intercept)            X2            X3            X4
## 1.085576e-109  7.470473e-10  3.764064e-28  7.908452e-11
```

The small p-values for X2, X3, X4 suggest that there is strong evidence to reject the null hypothesis for each corresponding coefficient

```
p_values2 <- summary(model2)$coefficients[, "Pr(>|t|)"]
p_values2
```

```
##  (Intercept)           X2      I(X6^2)      I(X4^2)      I(X5^2)
## 4.737889e-17 7.733627e-12 1.755300e-09 3.205417e-15 2.043878e-17
```

X2, I(X6^2), I(X4^2), I(X5^2) have very low p-value

```
p_values3 <- summary(model3)$coefficients[, "Pr(>|t|)"]
p_values3
```

```
##  (Intercept)           X2           X3      I(X6^2)      I(X4^2)      I(X5^2)
## 6.163358e-01 5.229740e-12 9.476893e-11 6.179729e-01 3.144951e-10 5.834516e-08
```

X2, X3, I(X4^2), and I(X5^2) have very small p-values, indicating strong evidence against the null hypothesis that their corresponding coefficients are zero

```
# Residual Sum of Squares(RSS)
rss <- sum(residuals(model)^2)
rss
```

```
## [1] 31931.41
```

```
rss1 <- sum(residuals(model1)^2)
rss1
```

```
## [1] 35090.93
```

```
rss2 <- sum(residuals(model2)^2)
rss2
```

```
## [1] 35999.55
```

```
rss3 <- sum(residuals(model3)^2)
rss3
```

```
## [1] 32479.87
```

RSS is minimum for model 1.

```
#Mean Squared Error(MSE)
mse <- mean(residuals(model)^2)
mse
```

```
## [1] 77.12902
```

```
mse1 <- mean(residuals(model1)^2)
mse1
```

```
## [1] 84.76071
```

```
mse2 <- mean(residuals(model2)^2)
mse2
```

```
## [1] 86.95543
```

```
mse3 <- mean(residuals(model3)^2)
mse3
```

```
## [1] 78.45378
```

MSE is minimum for model 1 hence Y = X1 + X2 + X3 + X4 + X5 + X6 is the most efficient model.

```r
# F-value/ANOVA
anova_table <- anova(model)
f_value <- anova_table$"F value"[1]
f_value
```

```
## [1] 7.466636
```

```r
anova_table1 <- anova(model1)
f_value1 <- anova_table1$"F value"[1]
f_value1
```

```
## [1] 39.61064
```

```r
anova_table2 <- anova(model2)
f_value2 <- anova_table2$"F value"[1]
f_value2
```

```
## [1] 38.51671
```

```r
anova_table3 <- anova(model3)
f_value3 <- anova_table3$"F value"[1]
f_value3
```

```
## [1] 42.5862
```

```r
# AIC and BIC
aic <- AIC(model)
aic
```

```
## [1] 2989.91
```

```r
aic1 <- AIC(model1)
aic1
```

```
## [1] 3022.972
```

```r
aic2 <- AIC(model2)
aic2
```

```
## [1] 3035.555
```

```r
aic3 <- AIC(model3)
aic3
```

```
## [1] 2994.96
```

AIC has the least value for model Y = X1 + X2 + X3 + X4 + X5 + X6 followed by Y = X2 + X3 + I(X6^2) + I(X4^2) + I(X5^2) The least value of AIC suggests the best fitting model

```
bic <- BIC(model)
bic
```

```
## [1] 3022.117
```

```
bic1 <- BIC(model1)
bic1
```

```
## [1] 3043.101
```

```
bic2 <- BIC(model2)
bic2
```

```
## [1] 3059.71
```

```
bic3 <- BIC(model3)
bic3
```

```
## [1] 3023.141
```

BIC has the least value for model Y = X1 + X2 + X3 + X4 + X5 + X6 and almost similar for every other model. The least value of BIC suggests the best fitting model. In this case, there is no significant difference to predict the best model. From AIC and BIC, it is evident that X2, X3, X4, X5, X6 have significant impact on Y.

MODEL 4 - Polynomial Regression

```
set.seed(3)
X <- RealEstate_Data[, -1]
Y <- RealEstate_Data$Y
attach(RealEstate_Data)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     Y
```

```
## The following objects are masked from RealEstate_Data (pos = 3):
##
##     X1, X2, X3, X4, X5, X6, Y
```

```
## The following objects are masked from healthdata (pos = 4):
##
##     X1, X2, X3, X4, X5
```

```
## The following objects are masked from healthdata (pos = 5):
##
##     X1, X2, X3, X4, X5
```

```r
train <- sample(1:nrow(RealEstate_Data),0.9*nrow(RealEstate_Data))
test <- (-train)
Realestate_train <- RealEstate_Data[train,]
Realestate_test <- RealEstate_Data[-train,]

degree <- 2
fit <- lm(Y ~ poly(X1, degree) + poly(X2, degree) + poly(X3, degree) +
poly(X4, degree) + poly(X5, degree) + poly(X6, degree), data = Realestate_train)
summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ poly(X1, degree) + poly(X2, degree) + poly(X3,
##     degree) + poly(X4, degree) + poly(X5, degree) + poly(X6,
##     degree), data = Realestate_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.929  -4.355  -0.243   3.522  31.331
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          38.1255     0.3586 106.303  < 2e-16 ***
## poly(X1, degree)1    24.1221     6.9805   3.456 0.000615 ***
## poly(X1, degree)2     2.6822     6.9744   0.385 0.700776
## poly(X2, degree)1   -63.4447     7.1773  -8.840  < 2e-16 ***
## poly(X2, degree)2    47.2315     7.5854   6.227 1.33e-09 ***
## poly(X3, degree)1   121.0465    77.1798   1.568 0.117676
## poly(X3, degree)2   118.9971    23.1570   5.139 4.55e-07 ***
## poly(X4, degree)1    47.6457     9.6657   4.929 1.26e-06 ***
## poly(X4, degree)2    -2.2811     7.6414  -0.299 0.765478
## poly(X5, degree)1    75.4873    13.2034   5.717 2.28e-08 ***
## poly(X5, degree)2   -38.2531    20.3700  -1.878 0.061204 .
## poly(X6, degree)1   171.5359    54.0100   3.176 0.001622 **
## poly(X6, degree)2 -125.6961    41.1187  -3.057 0.002404 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.917 on 359 degrees of freedom
## Multiple R-squared:  0.7326, Adjusted R-squared:  0.7237
## F-statistic: 81.97 on 12 and 359 DF,  p-value: < 2.2e-16
```

Adjusted R-squared is 0.7237 , indicating that the model explains 72.37% of the variability in the response data around its mean. The F-statistic is extremely high, and the associated p-value is very low (< 2.2e-16), indicating the model is statistically significant.

```r
coef(fit)
```

```
##       (Intercept) poly(X1, degree)1 poly(X1, degree)2 poly(X2, degree)1
##         38.125538         24.122119          2.682204        -63.444655
## poly(X2, degree)2 poly(X3, degree)1 poly(X3, degree)2 poly(X4, degree)1
##         47.231488        121.046486        118.997059         47.645662
```

```
## poly(X4, degree)2 poly(X5, degree)1 poly(X5, degree)2 poly(X6, degree)1
##         -2.281116         75.487348        -38.253129        171.535911
## poly(X6, degree)2
##       -125.696056
```

```
set.seed(1)
pred =  predict(fit, newdata = Realestate_test)
mse <- mean((Y[test] - pred)^2)
mse
```

```
## [1] 193.2498
```

MODEL 5 - Model selection

```
set.seed(1)
# To verify that there are no missing values
sum (is.na(RealEstate_Data))
```

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

```
# Fit model for all the variables and check summary
regfit.full <- regsubsets(Y~., RealEstate_Data)
reg.summary <- summary(regfit.full)
# The summary suggests that a single variable model has X3
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., RealEstate_Data)
## 6 Variables  (and intercept)
##    Forced in Forced out
## X1     FALSE      FALSE
## X2     FALSE      FALSE
## X3     FALSE      FALSE
## X4     FALSE      FALSE
## X5     FALSE      FALSE
## X6     FALSE      FALSE
## 1 subsets of each size up to 6
## Selection Algorithm: exhaustive
##          X1  X2  X3  X4  X5  X6
## 1  ( 1 ) " " " " "*" " " " " " "
## 2  ( 1 ) " " " " "*" "*" " " " "
## 3  ( 1 ) " " "*" "*" "*" " " " "
## 4  ( 1 ) " " "*" "*" "*" "*" " "
## 5  ( 1 ) "*" "*" "*" "*" "*" " "
## 6  ( 1 ) "*" "*" "*" "*" "*" "*"
```

```
num_of_coefficients <- which.max(reg.summary$adjr2)
num_of_coefficients
```

```
## [1] 5
```

```
which.min(reg.summary$rss)
```

```
## [1] 6
```

```
which.min(reg.summary$cp)
```

```
## [1] 5
```

```
which.min(reg.summary$bic)
```

```
## [1] 5
```

The adjusted R2, Cp and BIC indicates that a 5 variable model is the best one.

```
# Display the coefficients for the best subset model
coef(regfit.full,num_of_coefficients)
```

```
##   (Intercept)            X1            X2            X3            X4
## -1.596480e+04  5.137555e+00 -2.693805e-01 -4.353338e-03  1.136193e+00
##            X5
##  2.268794e+02
```

This selects X1, X2, X3, X4 and X5 as the 5 significant predictors

```
# Do forward and backward selection on the model
regfit.fwd <- regsubsets(Y~., data = RealEstate_Data, method="forward")
reg.summary <- summary(regfit.fwd)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = RealEstate_Data, method = "forward")
## 6 Variables  (and intercept)
##    Forced in Forced out
## X1     FALSE      FALSE
## X2     FALSE      FALSE
## X3     FALSE      FALSE
## X4     FALSE      FALSE
## X5     FALSE      FALSE
## X6     FALSE      FALSE
## 1 subsets of each size up to 6
## Selection Algorithm: forward
##          X1  X2  X3  X4  X5  X6
## 1  ( 1 ) " " " " " " "*" " " " "
## 2  ( 1 ) " " " " " " "*" "*" " "
## 3  ( 1 ) " " " " "*" "*" "*" " "
## 4  ( 1 ) " " " " "*" "*" "*" "*"
## 5  ( 1 ) "*" " " "*" "*" "*" "*"
## 6  ( 1 ) "*" "*" "*" "*" "*" "*"
```

```
num_of_coefficients <- which.max(reg.summary$adjr2)
num_of_coefficients
```

```
## [1] 5
```

```
which.min(reg.summary$rss)
```

```
## [1] 6
```

```
which.min(reg.summary$cp)
```

```
## [1] 5
```

```
which.min(reg.summary$bic)
```

```
## [1] 5
```

```
coef(regfit.fwd,num_of_coefficients)
```

```
##   (Intercept)            X1            X2            X3            X4
## -1.596480e+04  5.137555e+00 -2.693805e-01 -4.353338e-03  1.136193e+00
##            X5
##  2.268794e+02
```

This selects X1, X2, X3, X4 and X5 as the 5 significant predictors

```
regfit.bwd <- regsubsets(Y~., data = RealEstate_Data, method = "backward")
reg.summary <- summary(regfit.bwd)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = RealEstate_Data, method = "backward")
## 6 Variables  (and intercept)
##    Forced in Forced out
## X1     FALSE      FALSE
## X2     FALSE      FALSE
## X3     FALSE      FALSE
## X4     FALSE      FALSE
## X5     FALSE      FALSE
## X6     FALSE      FALSE
## 1 subsets of each size up to 6
## Selection Algorithm: backward
##          X1  X2  X3  X4  X5  X6
## 1  ( 1 ) " " " " " " "*" " " " "
## 2  ( 1 ) " " " " " " "*" "*" " "
## 3  ( 1 ) " " " " "*" "*" "*" " "
## 4  ( 1 ) " " " " "*" "*" "*" "*"
## 5  ( 1 ) "*" " " "*" "*" "*" "*"
## 6  ( 1 ) "*" "*" "*" "*" "*" "*"
```

```
num_of_coefficients <- which.max(reg.summary$adjr2)
num_of_coefficients
```

```
## [1] 5
```

```
which.min(reg.summary$rss)
```

```
## [1] 6
```

```
which.min(reg.summary$cp)
```

```
## [1] 5
```

```
which.min(reg.summary$bic)
```

```
## [1] 5
```

```
coef(regfit.bwd,num_of_coefficients)
```

```
##   (Intercept)            X1            X2            X3            X4
## -1.596480e+04  5.137555e+00 -2.693805e-01 -4.353338e-03  1.136193e+00
##            X5
##  2.268794e+02
```

This selects X1, X2, X3, X4 and X5 as the 5 significant predictors. For a 5 variable model, X5 seems to be the most significant predictor with a high coefficient.

```
# Split into training and test set
train <- sample(1:nrow(RealEstate_Data),0.7*nrow(RealEstate_Data))
test <- (-train)
RealEstate_train <- RealEstate_Data[train,]
RealEstate_test <- RealEstate_Data[-train,]
```

Since all the selection methods gave the same set of predictors (X1,X2,X3,X4 and X5),use it to calculate cross validation errors with 5 fold and 10 fold

```
set.seed(1)
regfit.best <- regsubsets(Y~X1+X2+X3+X4+X5,data = RealEstate_Data[train,])
test.mat<-model.matrix (Y~X1+X2+X3+X4+X5, data = RealEstate_Data[test,])
val.errors <- rep (NA, num_of_coefficients)
for (i in 1:num_of_coefficients) {
  coefi <- coef(regfit.best , id = i)
  pred <- test.mat[,names(coefi)] %*% coefi
  val.errors[i] <- mean((RealEstate_Data$Y[test] - pred)^2)
}
val.errors
```

```
## [1] 72.68456 70.94079 69.23575 53.56831 52.54730
```

```
num_of_predictors_mse <- which.min(val.errors)
num_of_predictors_mse
```

```
## [1] 5
```

MSE value is similar among the combinations and is minimum for a 5 variable model- 52.54730

```
set.seed(1)
coef(regfit.best , num_of_predictors_mse)
```

```
##   (Intercept)            X1            X2            X3            X4
## -1.715441e+04  5.576749e+00 -2.294846e-01 -4.376704e-03  1.200042e+00
##            X5
##  2.390939e+02
```

The validation set approach also shows that X5 is more significant in predicting the response coef of X5= 2.390939e+02

```
# Predict function to predict the output Y for k fold
set.seed(1)
predict.regsubsets <- function(object,newdata,id,...){
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form,newdata)
  coefi <- coef(object,id=id)
  xvars <- names(coefi)
  mat[,xvars]%*%coefi
}

# Model evaluation with 10 fold
k <- 10
n <- nrow(RealEstate_Data)
set.seed(1)
folds <- sample(rep(1:k, length = n))
cv.errors <- matrix(NA, k, num_of_coefficients,dimnames =
                      list (NULL , paste (1:num_of_coefficients)))

for  (j in 1:k){
  best.fit <- regsubsets(Y~X2+X4+X5,data=RealEstate_Data[folds!=j,])
  for (i in 1:num_of_coefficients){
    pred <- predict(regfit.best,RealEstate_Data[folds==j,],id=i)
    cv.errors[j,i] <- mean((RealEstate_Data$Y[folds==j]-pred)^2)
  }
}
cv.errors
```

```
##              1         2         3         4         5
##  [1,]  88.53811  88.79466  82.25306  71.70445  69.54232
##  [2,] 115.40784  94.47440 101.90847 101.29554  96.41677
##  [3,]  91.20828  86.31220  75.35041  72.68155  78.88796
##  [4,]  82.89198  78.93201  73.47685  61.08185  60.55321
##  [5,] 120.53864  94.36598  90.57265  82.82523  82.18138
```

```
##  [6,] 221.35756 229.42540 219.46718 209.12770 202.40357
##  [7,]  77.06155  73.34831  75.57638  68.31252  59.10441
##  [8,]  76.21849  56.04214  48.27085  33.49088  37.58946
##  [9,]  57.32834  57.83551  48.60492  34.94580  34.84980
## [10,]  81.85130  76.26987  74.91232  62.72289  55.74982
```

```
# Display the cv errors matrix
mean.cv.errors <- apply(cv.errors , 2, mean)
mean.cv.errors
```

```
##         1         2         3         4         5
## 101.24021  93.58005  89.03931  79.81884  77.72787
```

```
min(mean.cv.errors)
```

```
## [1] 77.72787
```

The display of minimum of the mean errors and it shows that the 5 variable model has the least cv error (77.72787)

```
# Model evaluation with 5 fold
set.seed(1)
k <- 5
n <- nrow(RealEstate_Data)
set.seed(1)
folds <- sample(rep(1:k, length = n))
cv.errors <- matrix(NA, k, num_of_coefficients,dimnames =
                      list (NULL , paste (1:num_of_coefficients)))

for  (j in 1:k){
  best.fit <- regsubsets(Y~.,data=RealEstate_Data[folds!=j,])
  for (i in 1:num_of_coefficients){
    pred <- predict(regfit.best,RealEstate_Data[folds==j,],id=i)
    cv.errors[j,i] <- mean((RealEstate_Data$Y[folds==j]-pred)^2)
  }
}
cv.errors
```

```
##              1         2         3         4         5
## [1,] 154.14772 158.26286 150.03353 139.58822 135.17257
## [2,]  96.46569  84.03862  88.90106  85.00272  77.98536
## [3,]  83.80369  71.35952  61.97376  53.32230  58.48749
## [4,]  70.26415  68.51085  61.19071  48.17127  47.85634
## [5,] 101.19497  85.31793  82.74249  72.77406  68.96560
```

```
mean.cv.errors <- apply (cv.errors , 2, mean)
mean.cv.errors
```

```
##         1         2         3         4         5
## 101.17524  93.49795  88.96831  79.77172  77.69347
```

```
min(mean.cv.errors)
```

```
## [1] 77.69347
```

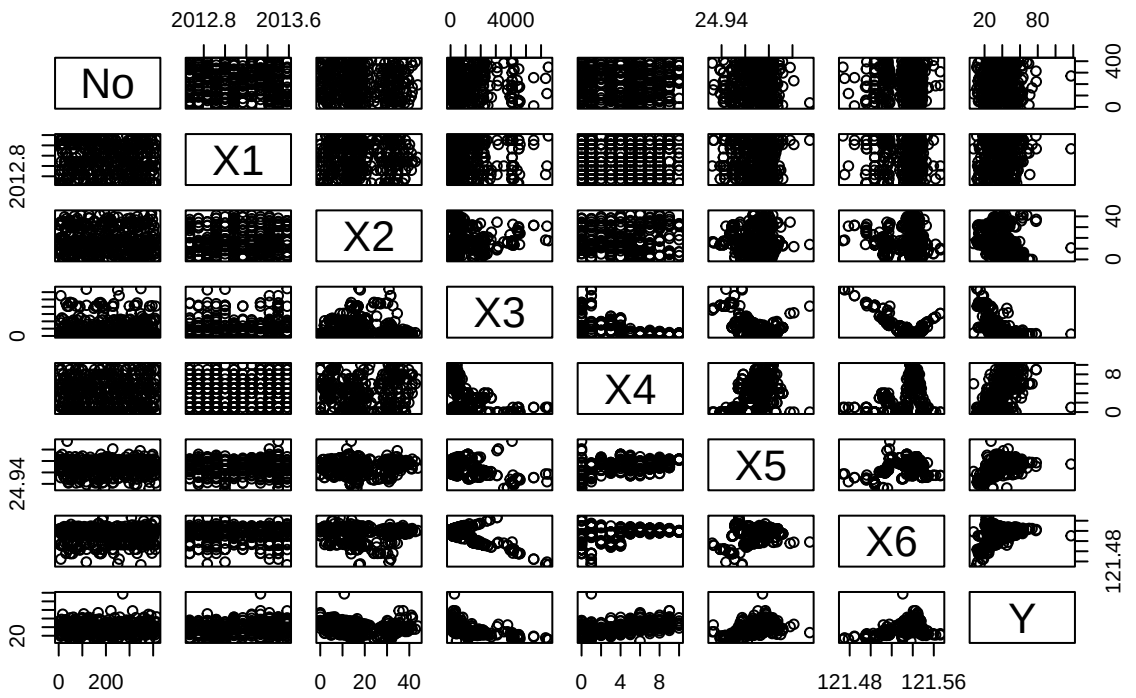
Display the minimum of the mean errors and it shows that the 3 variable model has the least cv error (77.69347)

MODEL 6 - GAMs

```
library(readxl)
real_estate <- read_excel("RealEstate.xlsx")
summary(real_estate)
```

```
##        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
```

```
pairs(real_estate[, sapply(real_estate, is.numeric)], main = "Pairs Plot")
```




```
# Correlation matrix
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.2
```

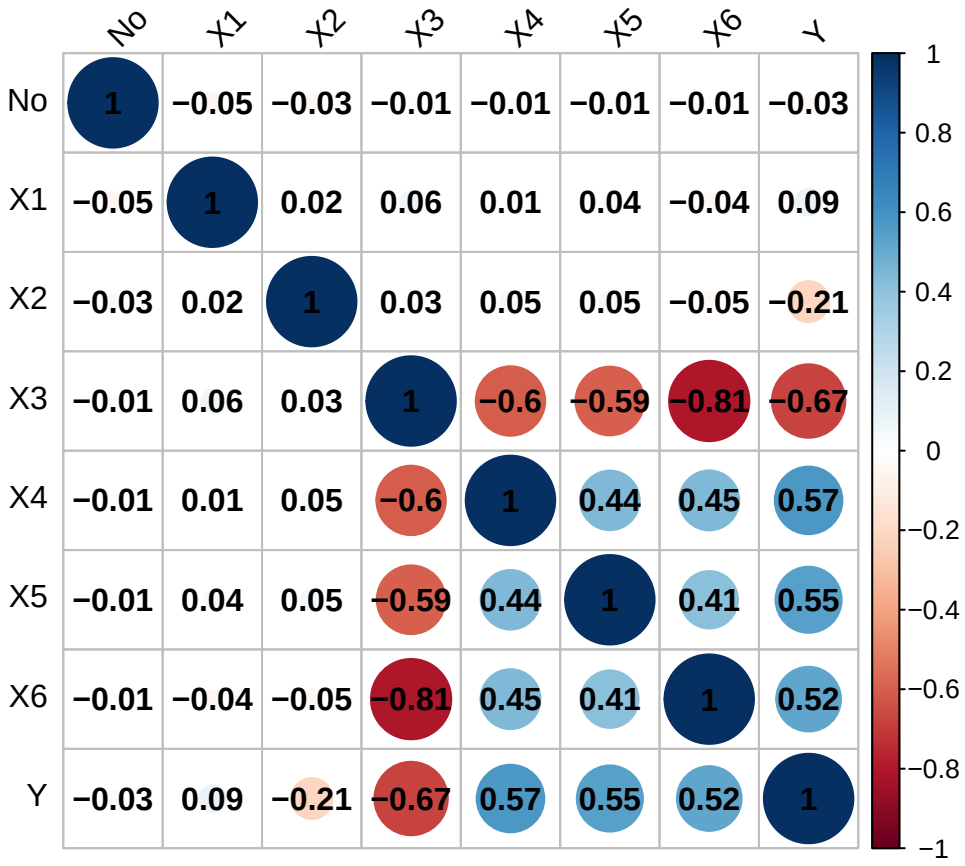
```
## corrplot 0.92 loaded
```

```
cor_matrix <- cor(real_estate)
cor_matrix
```

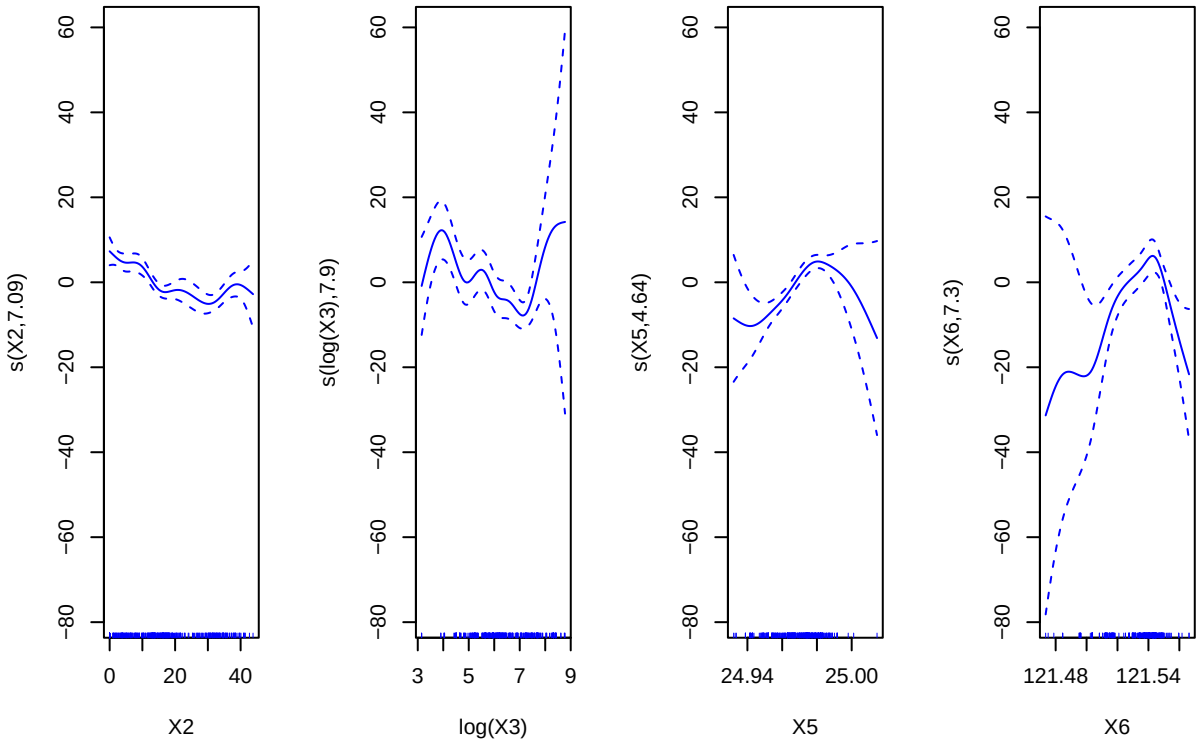
```
##             No           X1          X2          X3           X4          X5
## No  1.00000000 -0.048634447 -0.03280811 -0.01357349 -0.012698946 -0.01010966
## X1 -0.04863445  1.000000000  0.01754234  0.06088009  0.009544199  0.03501631
## X2 -0.03280811  0.017542341  1.00000000  0.02562205  0.049592513  0.05441990
## X3 -0.01357349  0.060880095  0.02562205  1.00000000 -0.602519145 -0.59106657
## X4 -0.01269895  0.009544199  0.04959251 -0.60251914  1.000000000  0.44414331
## X5 -0.01010966  0.035016305  0.05441990 -0.59106657  0.444143306  1.00000000
## X6 -0.01105928 -0.041065078 -0.04852005 -0.80631677  0.449099007  0.41292394
## Y  -0.02858717  0.087529272 -0.21056705 -0.67361286  0.571004911  0.54630665
##             X6           Y
## No -0.01105928 -0.02858717
## X1 -0.04106508  0.08752927
## X2 -0.04852005 -0.21056705
## X3 -0.80631677 -0.67361286
```

```
## X4  0.44909901  0.57100491
## X5  0.41292394  0.54630665
## X6  1.00000000  0.52328651
## Y   0.52328651  1.00000000
```

```
corrplot(cor_matrix, method = "circle", tl.col = "black",
         tl.srt = 45, addCoef.col = "black")
```




```
#GAMs
library(mgcv)
library(caret)
library(boot)
real_estate <- as.data.frame(real_estate)
# Removing the "No" column
real_estate <- real_estate[, -1]
# Rename columns
colnames(real_estate) <- c("X1", "X2", "X3", "X4", "X5", "X6", "Y")

gam.m1 <- gam(Y ~ s(X2) + s(log(X3)) + s(X5) + s(X6), data = real_estate)
par(mfrow = c(1, 4))
plot(gam.m1, se = TRUE, col = "blue")
```




```
summary(gam.m1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Y ~ s(X2) + s(log(X3)) + s(X5) + s(X6)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.9802     0.3642   104.3   <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(X2)       7.092  8.145  9.630  <2e-16 ***
## s(log(X3)) 7.897  8.429  7.053  <2e-16 ***
## s(X5)       4.644  5.738 11.322  <2e-16 ***
## s(X6)       7.303  8.257  2.389  0.0187 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.703   Deviance explained = 72.3%
## GCV = 58.898  Scale est. = 54.924    n = 414
```

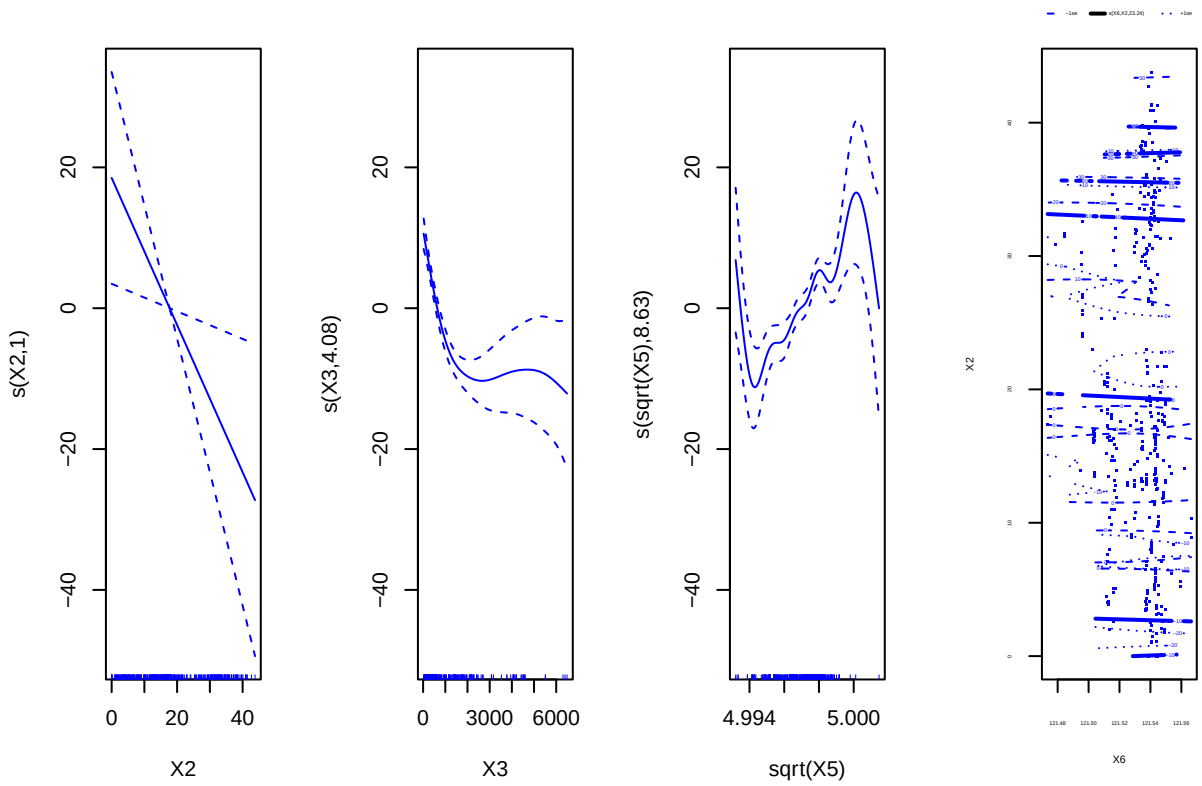
```r
pred.m1 <- predict(gam.m1, newdata = real_estate)
rss.m1 <- sum((real_estate$Y - pred.m1)^2)
rss.m1
```

```
## [1] 21204.04
```

```r
mse.m1 <- mean((real_estate$Y - pred.m1)^2)
mse.m1
```

```
## [1] 51.21749
```

```r
gam.m2 <- gam(Y ~ s(X2) + s(X3) + s(sqrt(X5)) + s(X6, X2), data = real_estate)
par(mfrow = c(1, 4))
plot(gam.m2, se = TRUE, col = "blue")
```




```r
summary(gam.m2)
```

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

```
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.9802     0.3602   105.4   <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(X2)        1.000  1.000  6.056   0.0143 *
## s(X3)        4.080  4.991 29.814  < 2e-16 ***
## s(sqrt(X5))  8.632  8.946 10.065  < 2e-16 ***
## s(X6,X2)    23.240 26.411  2.748 1.38e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.71   Deviance explained = 73.6%
## GCV = 59.136  Scale est. = 53.715    n = 414
```

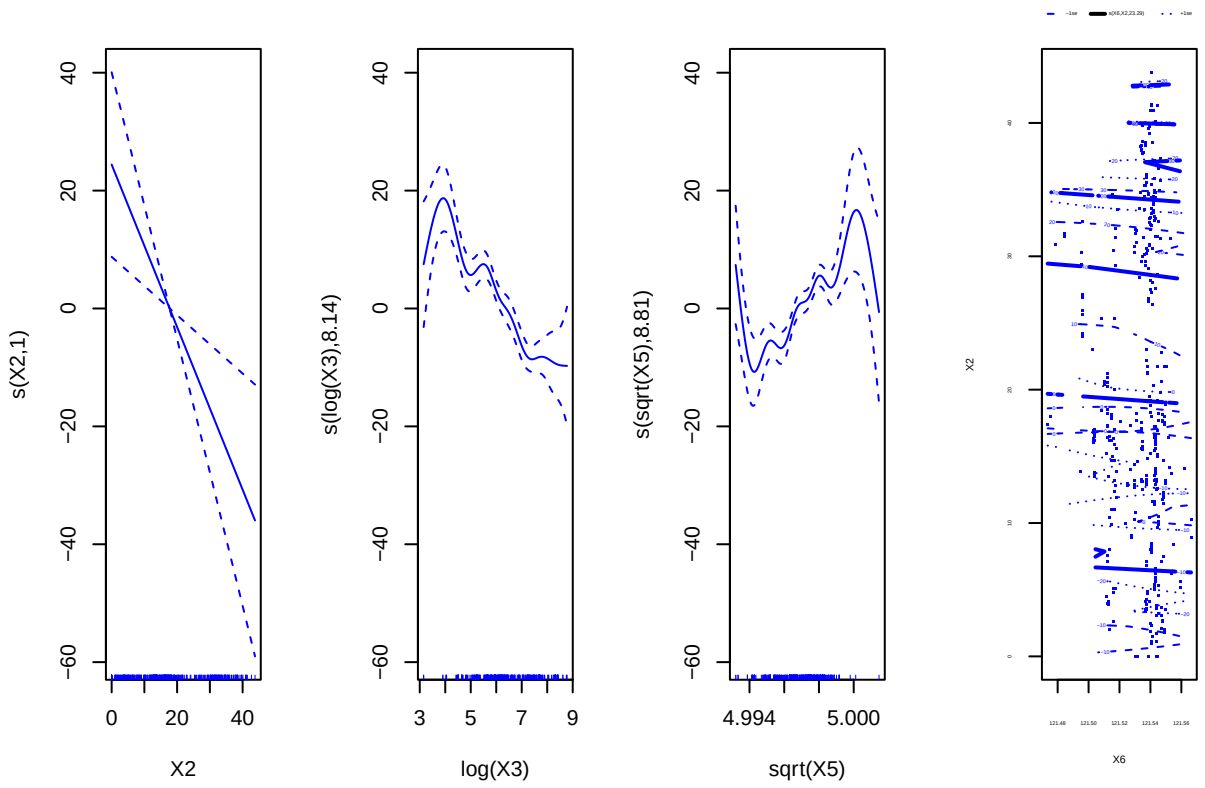
```
pred.m2 <- predict(gam.m2, newdata = real_estate)
rss.m2 <- sum((real_estate$Y - pred.m2)^2)
rss.m2
```

```
## [1] 20199.45
```

```
mse.m2 <- mean((real_estate$Y - pred.m2)^2)
mse.m2
```

```
## [1] 48.79095
```

```
gam.m3 <- gam(Y ~ s(X2) + s(log(X3)) + s(sqrt(X5)) +
                s(X6, X2), data = real_estate)
par(mfrow = c(1, 4))
plot(gam.m3, se = TRUE, col = "blue")
```




```
summary(gam.m3)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Y ~ s(X2) + s(log(X3)) + s(sqrt(X5)) + s(X6, X2)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.9802     0.3553   106.9   <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(X2)        1.000  1.000  9.733 0.00195 **
## s(log(X3))   8.137  8.768 19.678 < 2e-16 ***
## s(sqrt(X5))  8.811  8.984 10.993 < 2e-16 ***
## s(X6,X2)    23.295 26.450  2.623 3.3e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.718   Deviance explained = 74.6%
## GCV =   58.2  Scale est. = 52.261    n = 414
```

```r
pred.m3 <- predict(gam.m3, newdata = real_estate)
rss.m3 <- sum((real_estate$Y - pred.m3)^2)
rss.m3
```

```
## [1] 19428.54
```

```r
mse.m3 <- mean((real_estate$Y - pred.m3)^2)
mse.m3
```

```
## [1] 46.92883
```

```r
anova(gam.m1, gam.m2, gam.m3, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ s(X2) + s(log(X3)) + s(X5) + s(X6)
## Model 2: Y ~ s(X2) + s(X3) + s(sqrt(X5)) + s(X6, X2)
## Model 3: Y ~ s(X2) + s(log(X3)) + s(sqrt(X5)) + s(X6, X2)
##   Resid. Df Resid. Dev      Df Deviance      F   Pr(>F)
## 1    382.43      21204
## 2    371.65      20200 10.7786  1004.59 1.7834 0.056628 .
## 3    367.80      19429  3.8547   770.92 3.8268 0.005173 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

FINAL CONCLUSION (REALESTATE)

Conclusion and Best Model Recommendation:

GAM Model 3 (s(X2) + s(log(X3)) + s(sqrt(X5)) + s(X6, X2)) shows the best performance among the GAM models, with the lowest MSE, suggesting it might be an effective choice, especially if the goal includes understanding the relationships in the data as well as prediction.

Ridge Regression and Lasso Regression perform similarly in terms of MSE. Lasso identifies X4 and X5 as significant, while Ridge suggests X1, X4, X5, and X6 are significant.

The 5 Variable Model in Model Selection achieves a good balance with a relatively low MSE, but its cross-validation error is higher compared to other models.

Multiple Linear Regression and Polynomial Regression show higher MSEs, indicating they might not be as effective in this scenario.

Best Model: Considering both predictive accuracy (low MSE) and model complexity, GAM Model 3 seems to be the most suitable choice. It not only provides the lowest MSE but also might offer valuable insights into the data's structure and relationships, which can be crucial in real estate market analysis.

Final Thoughts: The choice of the best model also depends on specific needs, such as the trade-off between model complexity and interpretability, and the particular use case in the real estate context. The GAM Model 3 offers a good balance and is recommended based on the given data and results.