Assignment 2

Mitheysh Asokan, Jason (Eungjoo) Kim

Question 1: Nonlinear Regression

1.1 Process your data

Dataset chosen: diamonds

Choose the input and output variables.

```
diamonds <- read.csv('diamonds.csv')
diamonds <- select(diamonds,carat:price)

dim(diamonds)
```

```
## [1] 5000    7
```

```
head(diamonds)
```

```
##   carat       cut color clarity depth table price
## 1  0.77   Premium     E     SI1  60.4    58  2975
## 2  1.51      Fair     F      I1  67.8    59  3734
## 3  0.71   Premium     D     SI1  61.7    56  2863
## 4  0.90 Very Good     H     SI1  62.3    63  3387
## 5  1.00      Fair     I     SI1  67.9    62  2856
## 6  0.92 Very Good     J     SI1  62.6    58  3170
```

Remove all NAs from your data

```
diamonds <- na.omit(diamonds)

dim(diamonds)
```

```
## [1] 5000    7
```

Downsample if your data is larger than 5000 rows:

```
diamonds <- diamonds[diamonds$price > 2500,]

dim(diamonds)
```

```
## [1] 4460    7
```

If there are numeric variables that were supposed to be categorical, convert them to categorical

```
diamonds$cut <- as.factor(diamonds$cut)
diamonds$color <- as.factor(diamonds$color)
diamonds$clarity <- as.factor(diamonds$clarity)
```

1.2 train/Test Split

Split your data into train and test using 80/20 ratio. Print number of observations each dataset contains.

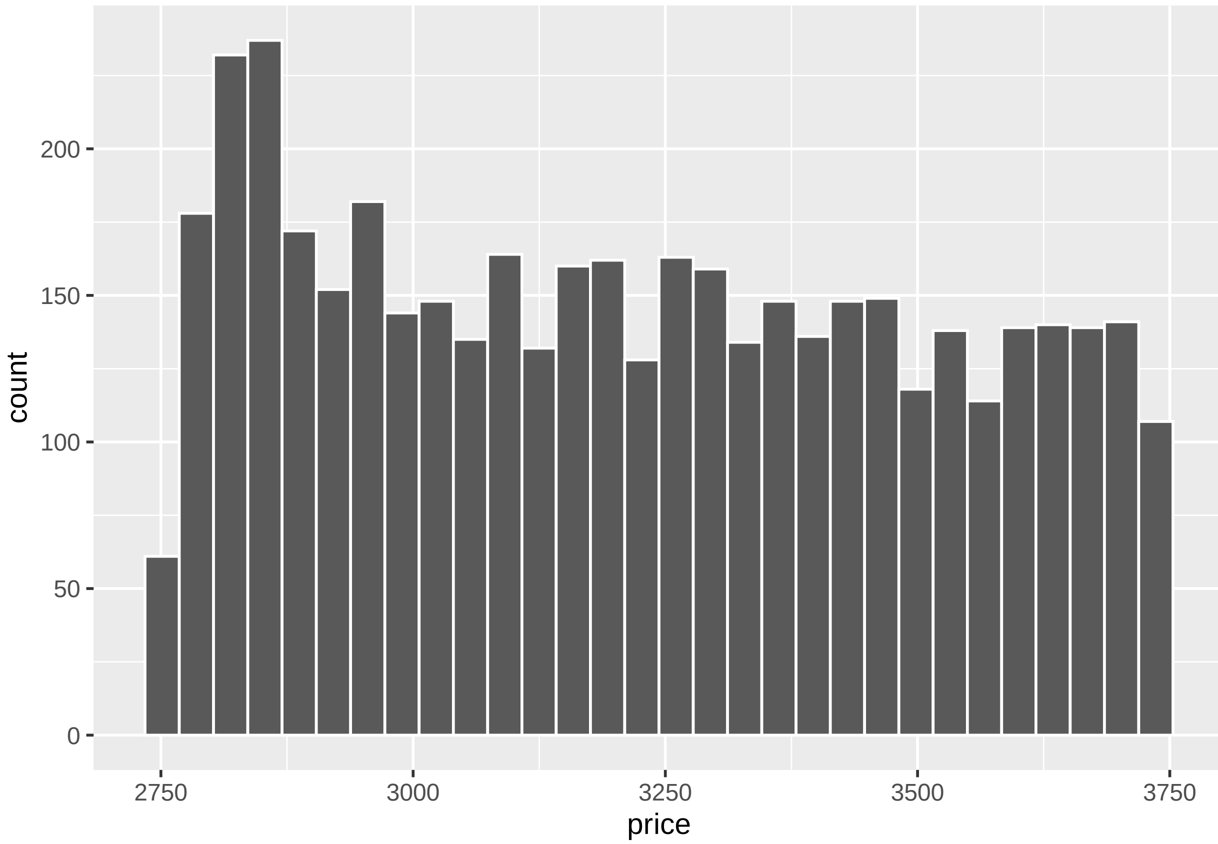
```
set.seed(100)

train_size <- floor(0.8 * nrow(diamonds))
train_inds <- sample(1:nrow(diamonds), size = train_size)
test_inds  <- setdiff(1:nrow(diamonds), train_inds)

train <- diamonds[ train_inds , ]
test  <- diamonds[ test_inds , ]

cat('train size:', nrow(train), '\ntest size:', nrow(test))
```

```
## train size: 3568
## test size: 892
```

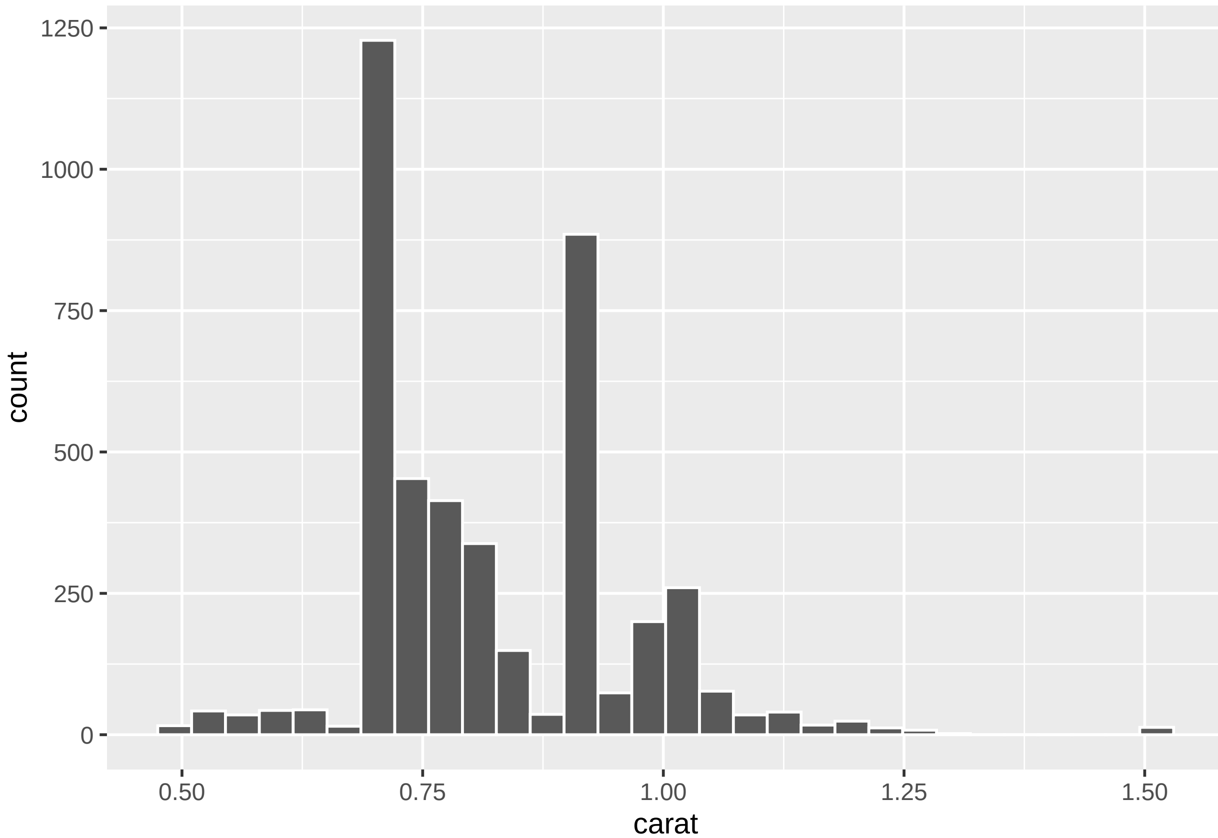
1.3 Visualize the data

Draw histogram of one of the numeric input variable and the output variable you selected.

```
ggplot(data = diamonds) +
  geom_histogram(aes(x = price), color = 'white')
```

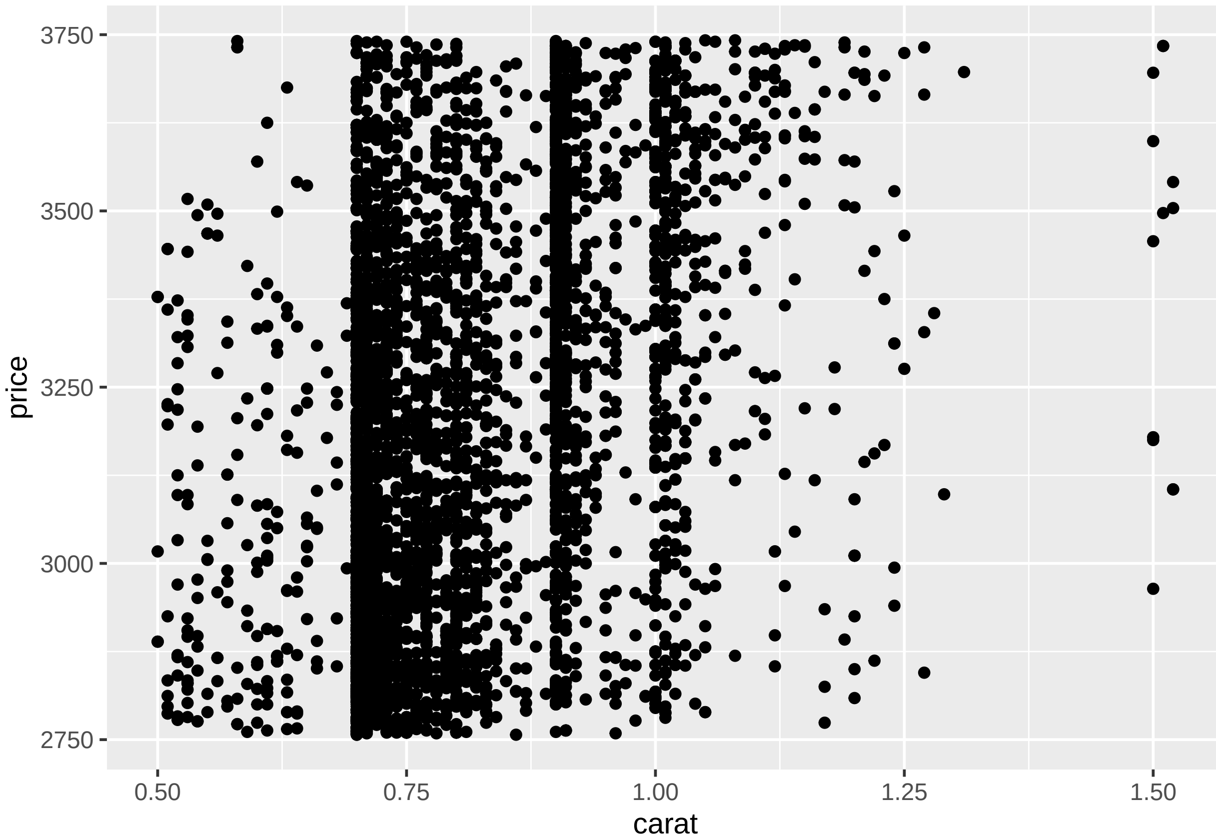



```
ggplot(data = diamonds) +
  geom_histogram(aes(x = carat), color = 'white')
```



Draw scatter plot between one of your numeric inputs and the output variable.

```
ggplot(data = diamonds) +
  geom_point(aes(x = carat, y = price))
```



Discuss whether the plot indicate a relation, if it is linear, if there are outliers?

The plot definitely hints at some minor relations. According to the diagram, the price of the diamond is increasing with respect to the carat. This is a true statement when compared to real diamond markets, where the carat weight of the diamond tends to partial influence the price of the diamond.

It is hard to definitely claim if the relationship is linear, but an upward sloping linear trend can be slightly observed.

Based on the data used, there doesn't seem to be any outliers.

1.4 Fit 4 models

One simple linear regression,

```
fit1 <- lm(price ~ carat, data = train)
```

One multilinear regression with all the variables you selected.

```
fit2 <- lm(price ~ . , data = train)
```

One polynomial regression using one input variable and one output variable.

```
fit3 <- lm(price ~ poly(carat,2), data = train)
```

One Locally Weighted Regression using one input numeric input variable and one output variable.

```r
fit4 <- loess(price ~ depth, data = train)
```

Calculate each model's RMSE on the train set. Which one performed the best and which did worse? Rank the models based on their training error.

Best: Model2

Worst: Model4

Ranking: Model2, Model3, Model1, Model4

```r
sigma(fit1)
```

```
## [1] 269.3641
```

```r
sigma(fit2)
```

```
## [1] 234.5879
```

```r
sigma(fit3)
```

```
## [1] 268.3738
```

```r
pred4 <- predict(fit4)
RMSE(pred4, train$price)
```

```
## [1] 291.9991
```

Calculate each model's RMSE on the test set. Which one performed the best and which did worse? Rank the models based on their test error.

Best: Model2

Worst: Model4

Ranking: Model2, Model3, Model1, Model4

```r
pred1 <- predict(fit1, newdata=test)
pred2 <- predict(fit2, newdata=test)
pred3 <- predict(fit3, newdata=test)

RMSE(pred1, test$price)
```

```
## [1] 265.4409
```

```r
RMSE(pred2, test$price)
```

```
## [1] 233.0535
```

```r
RMSE(pred3, test$price)
```

```
## [1] 265.2131
```

```r
RMSE(pred4, test$price)
```

```
## [1] 286.1072
```

Did the order of models change when ranked using training and test error?

No.

1.5. Cross Validation

Fit the 4 models but train using the cross validation.

```r
train.control <- trainControl(method = 'cv', number = 10)

model1 <- train(price ~ carat, data = diamonds, method = 'lm', trControl = train.control)
model2 <- train(price ~ ., data = diamonds, method = 'lm', trControl = train.control)
model3 <- train(price ~ poly(carat,2), data = diamonds, method = 'lm', trControl = train.control)
model4 <- train(price ~ carat, data = diamonds, method = 'gamLoess', trControl = train.control)

pred1 <- predict(model1, newdata=test)
pred2 <- predict(model2, newdata=test)
pred3 <- predict(model3, newdata=test)
pred4 <- predict(model4, newdata=test)

RMSE(pred1, test$price)
```

```
## [1] 265.0814
```

```r
RMSE(pred2, test$price)
```

```
## [1] 231.9981
```

```r
RMSE(pred3, test$price)
```

```
## [1] 264.7938
```

```r
RMSE(pred4, test$price)
```

```
## [1] 261.1916
```

What is the test error of each resulting model?

Best: Model2

Worst: Model1

Ranking: Model2, Model4, Model3, Model1

```
pred1 <- predict(model1, newdata=test)
pred2 <- predict(model2, newdata=test)
pred3 <- predict(model3, newdata=test)
pred4 <- predict(model4, newdata=test)

RMSE(pred1, test$price)
```

```
## [1] 265.0814
```

```
RMSE(pred2, test$price)
```

```
## [1] 231.9981
```

```
RMSE(pred3, test$price)
```

```
## [1] 264.7938
```

```
RMSE(pred4, test$price)
```

```
## [1] 261.1916
```

Did the order of models' test performances change when trained using cross validation?

Yes.

1.6. Shrinkage

Fit the first three models (exclude locally weighted model) using ridge regression.

```
x1 <- model.matrix(price ~ carat, data = diamonds)
x2 <- model.matrix(price ~ ., data = diamonds)
x3 <- model.matrix(price ~ poly(carat,2), data = diamonds)

y <- diamonds$price

fit1 <- glmnet(x1,y,alpha=0)
fit2 <- glmnet(x2,y,alpha=0)
fit3 <- glmnet(x3,y,alpha=0)
```

Calculate RMSE loss on test set.

```r
pred1 <- predict(fit1, newx=x1, newdata=test)
pred2 <- predict(fit2, newx=x2, newdata=test)
pred3 <- predict(fit3, newx=x3, newdata=test)

RMSE(pred1, test$price)
```

```
## [1] 290.3445
```

```r
RMSE(pred2, test$price)
```

```
## [1] 291.6528
```

```r
RMSE(pred3, test$price)
```

```
## [1] 290.5159
```

Fit the first three models (exclude locally weighted model) using lasso regression.

```r
fit4 <- glmnet(x1,y,alpha=1)
fit5 <- glmnet(x2,y,alpha=1)
fit6 <- glmnet(x3,y,alpha=1)
```

Calculate RMSE loss on test set.

```r
pred4 <- predict(fit4, newx=x1, newdata=test)
pred5 <- predict(fit5, newx=x2, newdata=test)
pred6 <- predict(fit6, newx=x3, newdata=test)

RMSE(pred4, test$price)
```

```
## [1] 299.5081
```

```r
RMSE(pred5, test$price)
```

```
## [1] 316.14
```

```r
RMSE(pred6, test$price)
```

```
## [1] 300.0939
```

Which model yielded the minimum test loss? Rank the 6 models.

Minimum Test Loss: Model1

Rankings: Model1, Model3, Model2, Model4, Model6, Model5

Question 2 Text Classification

2.2 Model Fitting

Initial Data Read and Setup

```
health <- read.csv("mental_health.csv")[,-1]

## 80% of the sample size
smp_size <- floor(0.80 * nrow(health))

## set the seed to make your partition reproducible
set.seed(123)
train_ind <- sample(seq_len(nrow(health)), size = smp_size)

train <- health[train_ind, ]
test <- health[-train_ind, ]

x   <- model.matrix(IsMentalHealthRelated ~ .,train)
y   <- train$IsMentalHealthRelated
```

Determining the optimal labmda values for L1 and L2 regularization

```
cv.fit  <- cv.glmnet(x,y,alpha=1, family="binomial", nfolds = 10)
cv.fit$lambda.min # 0.002455181

cv.fit  <- cv.glmnet(x,y,alpha=0, family="binomial", nfolds = 10)
cv.fit$lambda.min # 0.02822862
```

Fitting the Models

```
fit.logreg <- glmnet(x,y,family="binomial")
fit.l1 <- glmnet(x,y,alpha=1,family="binomial", lambda = 0.002455181)
fit.l2 <- glmnet(x,y,alpha=0,family="binomial", lambda = 0.02822862)
```

2.3 Performance Comparison

```
# Model w.o Regularization
newdata_x   <- model.matrix(IsMentalHealthRelated ~ .,test)
probs <- predict(fit.logreg,newdata_x,type = "response")
preds <- ifelse(probs >= 0.5, 1, 0)
target <- ifelse(test$IsMentalHealthRelated == "Yes", 1, 0)
acc1 <- mean(preds == target)

# Model with L1 Regularization
newdata_x   <- model.matrix(IsMentalHealthRelated ~ .,test)
probs <- predict(fit.l1,newdata_x,type = "response")
preds <- ifelse(probs >= 0.5, 1, 0)
```

```r
target <- ifelse(test$IsMentalHealthRelated == "Yes", 1, 0)
acc2 <- mean(preds == target)

# Model with L2 Regularization
newdata_x   <- model.matrix(IsMentalHealthRelated ~ .,test)
probs <- predict(fit.l2,newdata_x,type = "response")
preds <- ifelse(probs >= 0.5, 1, 0)
target <- ifelse(test$IsMentalHealthRelated == "Yes", 1, 0)
acc3 <- mean(preds == target)
```

Accuracy of logistic regression without regularization (acc1) is 41.2% Accuracy of logistic regression with l1 regularization (acc2) is 46.2% Accuracy of logistic regression with l2 regularization (acc3) is 46.5%

We see higher accuracy when applying regularization to the logistic regression as overfitting is minimized.

2.4 Interpretation of Models

L1 Regularization Sorted Results

```r
sort(coef(fit.l1)[,1])
```

```
##       fitness       workout        muscle         squat      workouts
## -12.354251851 -11.081349370 -10.346110217  -8.418041231  -8.072865923
##     time.week      shoulder         sugar           gym        weight
##  -7.128948542  -6.659329604  -6.426007410  -6.369614081  -6.228810353
##       protein      amp.x200b          size      strength           leg
##  -6.134612107  -6.008464620  -5.902251443  -5.549271926  -5.457451669
##      calories     deadlifts       however          legs      recently
##  -5.438329384  -5.430929660  -5.425298746  -5.400463048  -5.279800636
##          grip         bench   suggestions            hi      exercise
##  -5.221457651  -5.104493314  -5.061805117  -4.989918883  -4.946729657
##           bar       ampnbsp          lift         hello           arm
##  -4.652525955  -4.623989541  -4.502337087  -4.305443741  -4.113147404
##          bulk         stand         short          type      progress
##  -4.106652402  -4.097218993  -4.088156191  -4.074681689  -3.996053085
##          reps       example     days.week       routine        decide
##  -3.968410164  -3.934931568  -3.802909215  -3.744804283  -3.582204984
##       stretch           rep         carbs         lower         chest
##  -3.565385452  -3.458567932  -3.307777245  -3.294993032  -3.292411832
##           fit           old          curl         level           lbs
##  -3.136401760  -3.092310894  -3.089527452  -3.085732581  -3.061907642
##          turn       include           run      question         since
##  -3.059515335  -3.051907519  -2.954047357  -2.952201647  -2.942327990
##         weigh      body.fat        wonder          diet         press
##  -2.914526642  -2.861138441  -2.846878393  -2.829738642  -2.785347106
##         check          post          form           use          goal
##  -2.783356390  -2.722932889  -2.722072614  -2.676732239  -2.667601391
##       advance           tip       calorie           fat         cycle
##  -2.631461965  -2.571471196  -2.501612840  -2.496903323  -2.494589153
```

```
##      minutes         mass       follow        order       notice
## -2.483322086 -2.473792722 -2.427793623 -2.420304102 -2.419738362
##         test          buy       result        train          eat
## -2.402412292 -2.355806272 -2.351067264 -2.258837226 -2.229885950
##         male         game        split       reddit          ask
## -2.222498302 -2.208326258 -2.202881268 -2.126824397 -2.049928729
##          set          hey         look          pay         food
## -2.032512376 -2.012399072 -2.002635548 -2.002593528 -1.953175722
##          ago   appreciate         burn          abs         head
## -1.952784604 -1.940469405 -1.926257879 -1.879243375 -1.865813133
##     beginner        heavy        later    basically        front
## -1.861612727 -1.853808401 -1.826778873 -1.798047381 -1.789693294
##        small         edit         pull        drink      machine
## -1.770609558 -1.750379493 -1.745500692 -1.743661154 -1.737512868
##       decent        shape        would          big        whole
## -1.702688795 -1.701215845 -1.610034472 -1.605806103 -1.601485348
##         lean     increase         real         gain          ppl
## -1.599147154 -1.584606774 -1.548211113 -1.530198324 -1.520678739
##          may    different       please        watch         kind
## -1.507387436 -1.478373951 -1.471372616 -1.456393785 -1.439135682
##          one       cardio        enjoy        water          fix
## -1.429199990 -1.395246645 -1.372771397 -1.370023231 -1.346904718
##         push          guy       answer       family        point
## -1.322811815 -1.304544827 -1.284379365 -1.231829634 -1.200934527
##        heart          top         drop         year   especially
## -1.200302374 -1.194337071 -1.178085510 -1.176776951 -1.146654543
##          new         base    currently    every.day        track
## -1.122454779 -1.118893031 -1.117306020 -1.117085093 -1.106599484
##       little        goals       couple         home        stick
## -1.100717807 -1.091416778 -1.084575563 -1.071550500 -1.033536213
##         slow       matter       become          amp        never
## -1.031483297 -1.021712881 -1.020866874 -0.995117088 -0.991873117
##         call       anyone       months          see        could
## -0.974916188 -0.962642252 -0.955719615 -0.946659187 -0.921004443
##       others         full          two        close       search
## -0.908371905 -0.908022707 -0.897457882 -0.893109810 -0.891779035
##       number          sit        years    make.sure         tell
## -0.857291453 -0.849591267 -0.826268440 -0.814440679 -0.812071556
##         know      suppose       figure         hear           us
## -0.796608765 -0.725941572 -0.709781096 -0.699294062 -0.693127605
##        light         show        begin         body         seem
## -0.685283816 -0.675074716 -0.671937491 -0.668560097 -0.645503120
##         also         hold         free    deadlift       second
## -0.621224590 -0.612504495 -0.610530509 -0.571136751 -0.523508995
##          say         idea      nothing       either         true
## -0.516654706 -0.505421491 -0.488726755 -0.480100371 -0.473411668
##     anything         face       around      similar      general
## -0.414952445 -0.387388437 -0.326223564 -0.312788030 -0.297272038
##         side         come     personal      instead     together
## -0.289878565 -0.275863464 -0.274193403 -0.269041905 -0.267634538
```

```
##          cut         plan       friend        power       rather
## -0.256961917 -0.232865957 -0.208216541 -0.194618697 -0.183590358
##         rest      program        cause        night         walk
## -0.179655293 -0.155674371 -0.154016314 -0.150771097 -0.150584004
##   difference        every      problem         long         days
## -0.146905910 -0.112892533 -0.103705668 -0.098465873 -0.087890102
##        think         ever        share         meal        x200b
## -0.082760785 -0.078483372 -0.075737267 -0.070821363 -0.059029242
##           im          add        quite         fast         high
## -0.058831310 -0.054909335 -0.052599148 -0.049541371 -0.045861646
##        place  (Intercept)         able       advice       almost
## -0.028242767  0.000000000  0.000000000  0.000000000  0.000000000
##       always       amount      another         away         back
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##      barbell  bench.press         best         bite        bring
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##        build         case       change           co      comment
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##   completely     consider     continue      current        daily
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##         dont     dumbbell          end       energy       enough
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##          etc   everything   experience         fact         fall
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##          far    feel.like         felt      finally        first
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##      forward          fun     get.back         give       go.gym
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##          god        great        group         half         hand
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##         hang         hate       health      healthy          hit
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##         hour        hours    important      improve      incline
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##     interest           kg         last        leave          let
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##         like   literally         live    look.like         lose
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##  lose.weight          low         main     maintain         make
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##         mean         meet         mind       minute         miss
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##        money        month      morning         move         much
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##         need        next.       normal    obviously        often
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##          ohp         ones         open         pain         past
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##          per         play        pound       pretty          put
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
```

```
##       raise       reach        read      really   recommend
## 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
##    remember       right         row   seem.like       sense
## 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
##      single   situation        sort       start        step
## 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
##        stop      strong    struggle       stuff      summer
## 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
##       super     support        sure       thing    thoughts
## 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
##      thread       three        tire       today         try
## 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
##  understand       upper         ups     usually      volume
## 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
##        want         way       weeks     without        work
## 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
##  would.like       write       wrong         yet      monday
## 0.000000000 0.000000000 0.000000000 0.000000000 0.004807194
##      person    whatever       drive    possible        stay
## 0.022662881 0.023021167 0.026123760 0.034590735 0.036533713
##         due        pick    everyone       guess       least
## 0.050421486 0.119932069 0.128443415 0.133792494 0.139552907
##         lot        fail       thank   something        wake
## 0.140957211 0.160711007 0.170879979 0.172700488 0.176148885
##        love         day      parent       else.     average
## 0.223902986 0.228868047 0.229341312 0.230413195 0.234053066
##      happen        keep        okay     deficit      break.
## 0.242759550 0.242859801 0.247633789 0.256677160 0.266087846
##        find          gt     go.back      things       brain
## 0.312150096 0.328999511 0.334348615 0.339758534 0.342778656
##      reason        shit        hope        pass       world
## 0.357369167 0.359171847 0.372642686 0.388308663 0.392203080
##     believe         hop        even         get        many
## 0.393769076 0.416215837 0.424073766 0.427180912 0.465120720
##       spend        time        good    honestly         man
## 0.466371690 0.468759330 0.480214730 0.486600959 0.509257628
##       learn         bad          go       sleep        self
## 0.523957929 0.569232329 0.578327048 0.615873391 0.616767391
##        part         lol       sorry     anymore        talk
## 0.628719772 0.649292852 0.663603491 0.666489896 0.680261174
##       alone        soon     realize     someone      finish
## 0.708090869 0.713747561 0.719709236 0.729087207 0.739685944
##       worth        care        take        hard        well
## 0.767134760 0.785896981 0.874227260 0.900007809 0.915145188
##      people        hurt       issue       still     weekend
## 0.958353516 0.976959237 1.004398496 1.013695046 1.036955969
##     already        loss      intake      better   sometimes
## 1.038902625 1.044927929 1.107931093 1.115010308 1.136753763
## (Intercept)      switch     outside        nice        fuck
## 1.160292736 1.164638257 1.194930179 1.216041400 1.254125770
```

```
##         manage          fam         week      friends     schedule
##    1.272084055  1.281584898  1.302974609  1.313574772  1.398444168
##           less        might        maybe       though     probably
##    1.491117397  1.532271956  1.543124196  1.585236621  1.596030217
##          grade         fine         help     problems         wish
##    1.619598922  1.667720059  1.714469405  1.760626461  1.776317624
##         stress       mostly        sound         wait        class
##    1.815167266  1.890394287  1.919894253  1.955811290  2.046484567
##         course   definitely         deal       school         easy
##    2.064597657  2.250697873  2.369649828  2.376032640  2.395469352
##         mental      service         yeah         life        focus
##    2.526623165  2.573168603  2.690094308  2.706072225  2.707690149
##          co.op          job         feel        worry     actually
##    2.721728574  2.732410061  2.740582430  2.748090178  2.847616425
##           mark        happy   depression        study      anxiety
##    2.871445616  2.897102868  3.018362888  3.165072456  3.398525503
##     university           op mental.health      counsel         term
##    4.300543070  4.853024202  5.297897049  7.540543493  8.488821996
```

L2 Regularization Sorted Results

```
sort(coef(fit.l2)[,1])
```

```
##        fitness       workout     time.week         sugar      workouts
## -6.3014707830 -5.4593245458 -5.3907605360 -5.0449027019 -5.0117839575
##         muscle          size       protein           gym   suggestions
## -4.7535524403 -4.6229089769 -4.5830338468 -4.5170406906 -4.4163510026
##          squat       however       ampnbsp      shoulder          grip
## -4.2632485446 -4.2565864415 -4.2558792518 -4.2513361023 -4.2024917523
##       recently     deadlifts            hi        weight      calories
## -4.1365604904 -4.0496252728 -4.0021824886 -3.8783614977 -3.8646123477
##      days.week      strength          legs      exercise           leg
## -3.8616079275 -3.8536111864 -3.8040357812 -3.7714697435 -3.7696023398
##          stand       stretch           bar           arm           rep
## -3.6231350720 -3.6143401633 -3.5782710649 -3.5486706916 -3.5325493816
##           type          lift        decide         hello          mass
## -3.4943489520 -3.4633282686 -3.4573845069 -3.4562654186 -3.3980406169
##       body.fat          reps         short       example      progress
## -3.3896543341 -3.3738794985 -3.3015885049 -3.2644568747 -3.2508317263
##          lower           fit         bench         carbs       calorie
## -3.1980895897 -3.1917009848 -3.1588447709 -3.1511989796 -3.1466832228
##        include           old          bulk         weigh          diet
## -2.9883356798 -2.9538931278 -2.9407167563 -2.8632212614 -2.8415604156
##           curl       routine       advance         level        notice
## -2.8266446652 -2.8098365778 -2.8082323920 -2.7872577119 -2.7688744179
##       beginner           lbs      question         cycle          form
## -2.7376660526 -2.7309542108 -2.7286355515 -2.7239326468 -2.6900593347
##           turn          test          goal           tip           run
## -2.6874504599 -2.6681027804 -2.6123325832 -2.5913526405 -2.5832785452
##          chest         check         press         split           use
```

```
## -2.5649723038 -2.5339071540 -2.5245788985 -2.5096469092 -2.5088416010
##        wonder        cardio      amp.x200b         x200b          male
## -2.5015438426 -2.4852081855 -2.4481133604 -2.4459417485 -2.4211693208
##         since          gain         heavy       minutes          burn
## -2.3998485202 -2.3906678848 -2.3847395134 -2.3738800720 -2.3700901014
##           buy         order       barbell         train          post
## -2.3682336246 -2.3232732742 -2.2962736413 -2.2806561123 -2.2734276761
##       incline           fat           abs            kg        follow
## -2.2631789217 -2.2476981939 -2.2238688590 -2.2210621338 -2.2081229191
##        decent           ppl     currently        result      increase
## -2.1781731094 -2.1646217141 -2.1550552676 -2.1404977984 -2.1309367418
##     basically           set          head         front           ago
## -2.1279917367 -2.1013519339 -2.0987142358 -2.0910250075 -2.0797401085
##           eat          game           ohp        search         shape
## -2.0734762811 -2.0605329653 -2.0605306021 -2.0571531106 -2.0540743788
##           hey          food         track         goals    appreciate
## -2.0216992869 -2.0053265371 -1.9970756913 -1.9939759399 -1.9858502756
##          lean        volume          pull        reddit         drink
## -1.9770308742 -1.9702952382 -1.9331387309 -1.8905943463 -1.8786237442
##      deadlift         later          edit   bench.press           ask
## -1.8554603755 -1.8535066661 -1.8522178116 -1.8241789159 -1.8214385125
##       machine          push          base          look           amp
## -1.8183740252 -1.8092000088 -1.8074569394 -1.7788381810 -1.7735109377
##         small           big           fix           pay          slow
## -1.7564971389 -1.7363387957 -1.6911225948 -1.6751644191 -1.6129393455
##         watch         whole        answer        months           may
## -1.6040596537 -1.5881230489 -1.5661991940 -1.5610263910 -1.5509699650
##           top         would    especially     every.day          drop
## -1.5494912950 -1.5403228895 -1.5281303904 -1.5198718685 -1.5156526173
##     different         water        please           guy           sit
## -1.5010105562 -1.4880260975 -1.4788425447 -1.4712847841 -1.4664717293
##         enjoy         stick          real      dumbbell          kind
## -1.4581149298 -1.4478483445 -1.4450518817 -1.4402668684 -1.3924495910
##        become        couple          home          body         point
## -1.3735533094 -1.3640851575 -1.3462769295 -1.3432407409 -1.3385542300
##         heart         raise           one        family        matter
## -1.3015915889 -1.2941468223 -1.2769364513 -1.2721076828 -1.2667983485
##       suppose       general          main          year           new
## -1.2609676224 -1.2536713121 -1.2529716616 -1.2256000658 -1.2191246357
##          full        anyone         could        number         light
## -1.2189031064 -1.2179701434 -1.2153615270 -1.1782856273 -1.1682462729
##         begin        others        little         never     make.sure
## -1.1566075700 -1.1437175842 -1.1433127602 -1.1349773227 -1.1286300002
##          pain          call          meal        figure         years
## -1.0791929232 -1.0754358276 -1.0646768997 -1.0610789840 -1.0513912199
##    difference        either   lose.weight         pound         close
## -1.0504952863 -1.0474596311 -1.0251766142 -1.0144517363 -0.9996131329
##           two       similar          hear          idea           hit
## -0.9833580196 -0.9811250740 -0.9759943746 -0.9654921772 -0.9612358262
##          know          fast           see           low          seem
```

```
## -0.9506899616 -0.9342330954 -0.9264628706 -0.9222024216 -0.9186336213
##           per         power            us          tell          plan
## -0.9169076874 -0.9039178663 -0.8856684530 -0.8741376075 -0.8715255352
##           ups      personal      together          show        around
## -0.8585051754 -0.8319110421 -0.8191004899 -0.8023572507 -0.7974800759
##           add          face       program        amount           cut
## -0.7936648499 -0.7915434469 -0.7908067958 -0.7889954097 -0.7712574147
##          rest        rather       healthy          hold        second
## -0.7658136415 -0.7602637462 -0.7541856916 -0.7538080690 -0.7530264552
##          hand       instead          free          true       nothing
## -0.7469195476 -0.7396082549 -0.7395681241 -0.7395433711 -0.7275410116
##      anything          also           say          days         month
## -0.7158678375 -0.7036591877 -0.7028048503 -0.6884196819 -0.6571548452
##          high         night         quite          walk          side
## -0.6466803675 -0.6403574269 -0.6166421653 -0.5962342408 -0.5936595498
##         cause     look.like        go.gym       problem         start
## -0.5783978878 -0.5723968256 -0.5303486561 -0.5254974686 -0.5200104115
##    would.like         share         often          come    understand
## -0.5146483056 -0.5071829679 -0.5048011838 -0.4789851833 -0.4728202802
##          back          long         every          fact         stuff
## -0.4592600295 -0.4493039096 -0.4427077564 -0.4314551436 -0.4277899003
##         leave     important            im       forward         place
## -0.4256644820 -0.3724287808 -0.3720929709 -0.3702913961 -0.3677470210
##        friend           fun          play       without          give
## -0.3659192227 -0.3434494767 -0.3407481778 -0.3332064197 -0.3283706553
##          ever          bite           etc          mean          live
## -0.3275178761 -0.3259173846 -0.3094338519 -0.2747407858 -0.2643439340
##          step          need      interest        change          work
## -0.2632583140 -0.2622954327 -0.2478457416 -0.2406913130 -0.2378244614
##      continue          last          tire         think    completely
## -0.2354165764 -0.2223880324 -0.2122659367 -0.2119235611 -0.2118188707
##          case        strong        advice        summer          best
## -0.1871601569 -0.1858805813 -0.1823680123 -0.1808550128 -0.1449830359
##       another       morning          open        thread           put
## -0.1381024221 -0.1246519909 -0.1167554289 -0.1130437680 -0.1103333589
##         next.         three         daily        minute           god
## -0.1051011477 -0.1036731307 -0.0929778481 -0.0894601329 -0.0887551013
##           far          sure         weeks         today          want
## -0.0746106501 -0.0717217300 -0.0706163064 -0.0653950727 -0.0612031088
##       improve     recommend           row          lose          make
## -0.0600066602 -0.0585648976 -0.0515403603 -0.0506839602 -0.0432488731
##         money       current        pretty         group         build
## -0.0425457281 -0.0337760666 -0.0275205692 -0.0253887553 -0.0246559607
##          dont         wrong         great      remember         upper
## -0.0235842852 -0.0152511862 -0.0052094966 -0.0011225858 -0.0000970257
##   (Intercept)        energy         hours        always        almost
##  0.0000000000  0.0022712655  0.0057550657  0.0261856247  0.0304709907
##       comment          stop          like          felt          away
##  0.0324729899  0.0437330020  0.0453016502  0.0579237096  0.0632236582
##         first          read        normal          hang        really
```

```
##  0.0642196169  0.0697895259  0.0711766525  0.0743387941  0.1035663548
##          hour           yet        enough         bring         thing
##  0.1164666823  0.1242566544  0.1330497532  0.1382497719  0.1631085457
##         write       finally          mind          sort    everything
##  0.1695388598  0.1750566587  0.1801362911  0.1867877472  0.1975231928
##           try         right    experience         reach      maintain
##  0.1979492470  0.2102529126  0.2276948635  0.2292622010  0.2324051524
##          much     something         thank          meet           let
##  0.2328485791  0.2386720826  0.2455340001  0.2469905436  0.2542378888
##      consider      thoughts          able          miss     seem.like
##  0.2656783919  0.2656996509  0.2947650800  0.2956335507  0.3127569669
##          past        single           way          wake       support
##  0.3174303580  0.3436362580  0.3585757676  0.3649272606  0.3911839207
##          hate           lot       usually          move     situation
##  0.3997730957  0.3999853568  0.4100216426  0.4208929109  0.4332445021
##          half           day           get          okay        monday
##  0.4394150730  0.4451928874  0.4532616090  0.4694802616  0.4714869436
##         guess          fall            gt          keep      struggle
##  0.4737704708  0.4745002434  0.4826246343  0.4910096012  0.4942136461
##          find        reason          ones      everyone         drive
##  0.5172957480  0.5257408282  0.5265598633  0.5286657268  0.5350509453
##            go          time     obviously         else.         sense
##  0.5380207244  0.5397320897  0.5415618674  0.5529689981  0.5533127897
##        happen          love           end        person          pick
##  0.5654757568  0.5702339547  0.5735635274  0.5774587103  0.5830783195
##         super         least          shit          stay           due
##  0.6111703208  0.6116686664  0.6148487091  0.6280520008  0.6370447426
##     literally        things      possible          good      whatever
##  0.6498253061  0.6502362904  0.6543679225  0.6607218023  0.6747143153
##         world          hope     feel.like      get.back         learn
##  0.7018738551  0.7076926957  0.7206927416  0.7229094532  0.7366096083
##          fail        break.           man          even           bad
##  0.7435336267  0.7471763316  0.7516934250  0.7517675787  0.7579481375
##         sleep       believe          many          pass          take
##  0.7612240469  0.7915650429  0.8480278279  0.8514831328  0.8634834079
##       someone      honestly          loss         spend           lol
##  0.8650163568  0.8689320482  0.8811917534  0.9084973672  0.9265911080
##          hurt          care       anymore        people         sorry
##  0.9658197519  0.9779960681  0.9840452814  0.9990499741  1.0144683180
##          talk         still        health         brain        better
##  1.0414485406  1.0612475835  1.0627656922  1.0824914579  1.0990777183
##          part      schedule       realize          well        parent
##  1.1174095192  1.1255724770  1.1474443332  1.1550225068  1.1553719232
##          soon   (Intercept)          week         alone       go.back
##  1.1574129082  1.1585548046  1.1698438678  1.2029682755  1.2164210714
##           hop       deficit          self         issue         worth
##  1.2171870641  1.2413615823  1.2549401078  1.2663851229  1.2791783849
##        finish          fuck       average          hard        switch
##  1.2997577538  1.3052414067  1.3073922271  1.3500243612  1.3621990222
##     sometimes       weekend       already          nice       friends
```

```
## 1.3660445688 1.3811130417 1.4101361745 1.4120526443 1.4316537875
##         help          fam        maybe       though        might
## 1.4635274172 1.4727370497 1.5200270214 1.5943520246 1.6070810691
##     probably      outside           co       intake         fine
## 1.6116643160 1.6241323565 1.6305263342 1.6478898471 1.6722751547
##         less        grade        sound         wait       mostly
## 1.7214579396 1.8014224312 1.8110982759 1.8125570700 1.8414860624
##       manage         wish         feel        class       stress
## 1.8822457193 1.8860032602 1.9013634563 1.9468354446 1.9879342797
##     problems       course        focus       school     actually
## 2.0872402391 2.0884777594 2.2201899739 2.2249273692 2.2605014606
##         easy         deal        happy   definitely         life
## 2.2644391575 2.3062968006 2.3074405361 2.3243714814 2.3288887314
##         yeah        worry        study          job       mental
## 2.3321555406 2.3456753000 2.3768767365 2.3960567090 2.5645584171
##         mark        co.op   depression      anxiety      service
## 2.6168562773 2.6184329447 2.6540591390 2.8974208726 2.9186257280
## mental.health          op   university      counsel         term
## 2.9382126015 3.0315360843 3.6288453948 4.0957587758 4.6581584734
```

Words that are related to school and mental health, such as: "depression, term, mental-health, anxiety, co-op" seemed to show the highest coefficient estimations.The words that are related to fitness, such as: "fitness, workout, muscle, protein, weight" show lowest coefficient estimations.

L1 regularization method tends to approximate many coefficients when the values are higher than two significant digits to zero. The L1 regularization enforces sparsity; the less important coefficients given the model are zeroed out. This methodology essentially removes the corresponding feature from the model, hence optimizing RAM usage as well as reducing noise in the model.

L2 regularization method tends to shrink all the coefficients and doesn't zero any. This is because we define the regularization in L2 as the sum of the squares of all the feature weights. In this context, weights close to zero have little effect on the model complexity, while outlier weights can have a significant impact. Therefore, we are not concerned with zeroing any of the coefficients.

Question 3 Subset Selection

Forward Selection

Data Initialization and setting up the variables

```
ames        <- AmesHousing::make_ames()
numericVars <- ames %>% summarise_all(is.numeric) %>% unlist()
ames        <- ames[, numericVars]
NumCols     <- ncol(ames)

res <- regsubsets(Sale_Price ~ .,data=ames, method = "forward",  nvmax=NumCols)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 1 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
## Warning in rval$lopt[] <- rval$vorder[rval$lopt]: number of items to replace is
## not a multiple of replacement length
```

```
smm <- summary(res)
smm$rss
```

```
##  [1] 9.354907e+12 6.372705e+12 5.372622e+12 5.000405e+12 4.711132e+12
##  [6] 4.509022e+12 4.366282e+12 4.216771e+12 4.101703e+12 4.014448e+12
## [11] 3.952959e+12 3.910112e+12 3.877808e+12 3.852701e+12 3.829707e+12
## [16] 3.808074e+12 3.788825e+12 3.772223e+12 3.759006e+12 3.747105e+12
## [21] 3.736053e+12 3.725905e+12 3.716953e+12 3.708821e+12 3.704526e+12
## [26] 3.703314e+12 3.702500e+12 3.701952e+12 3.701714e+12 3.701525e+12
## [31] 3.701381e+12 3.701365e+12 3.701352e+12
```

```
min_rss <- which.min(smm$rss)
min_bic <- which.min(smm$bic)

min_rss
```

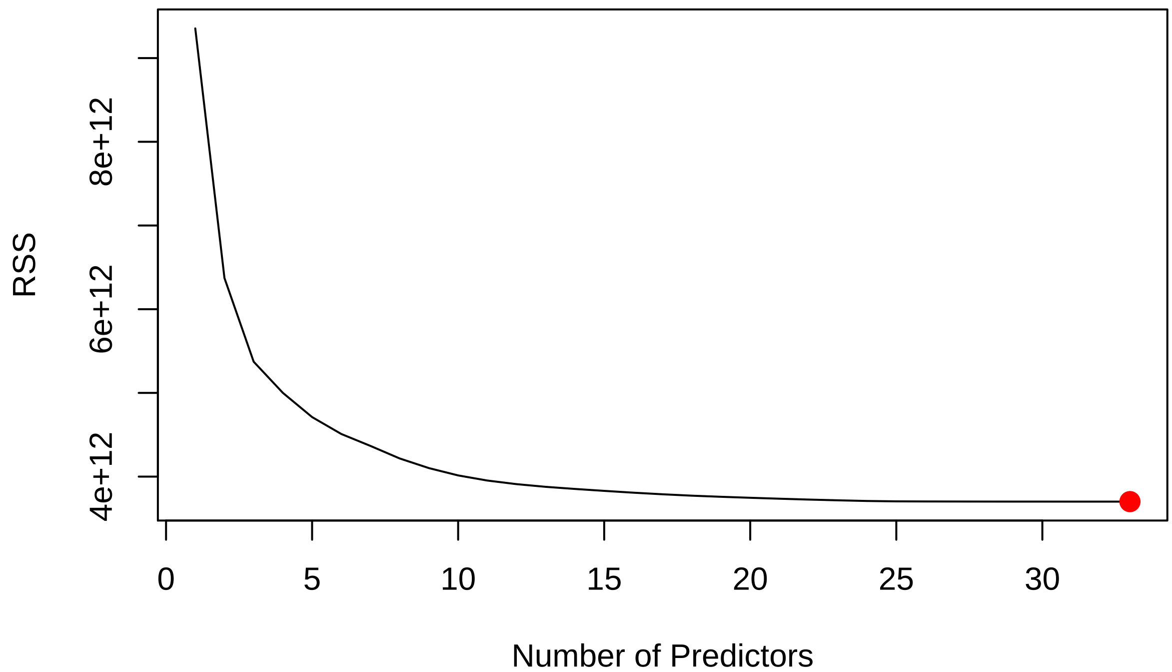
```
## [1] 33
```

```
min_bic
```

```
## [1] 21
```

RSS Plot (Forward Selection) Plotting the RSS of each Model (Forward Selection)

```r
plot(smm$rss,xlab="Number of Predictors", ylab="RSS", type='l')
points(min_rss, smm$rss[min_rss], col="red", cex=2, pch=20)
```



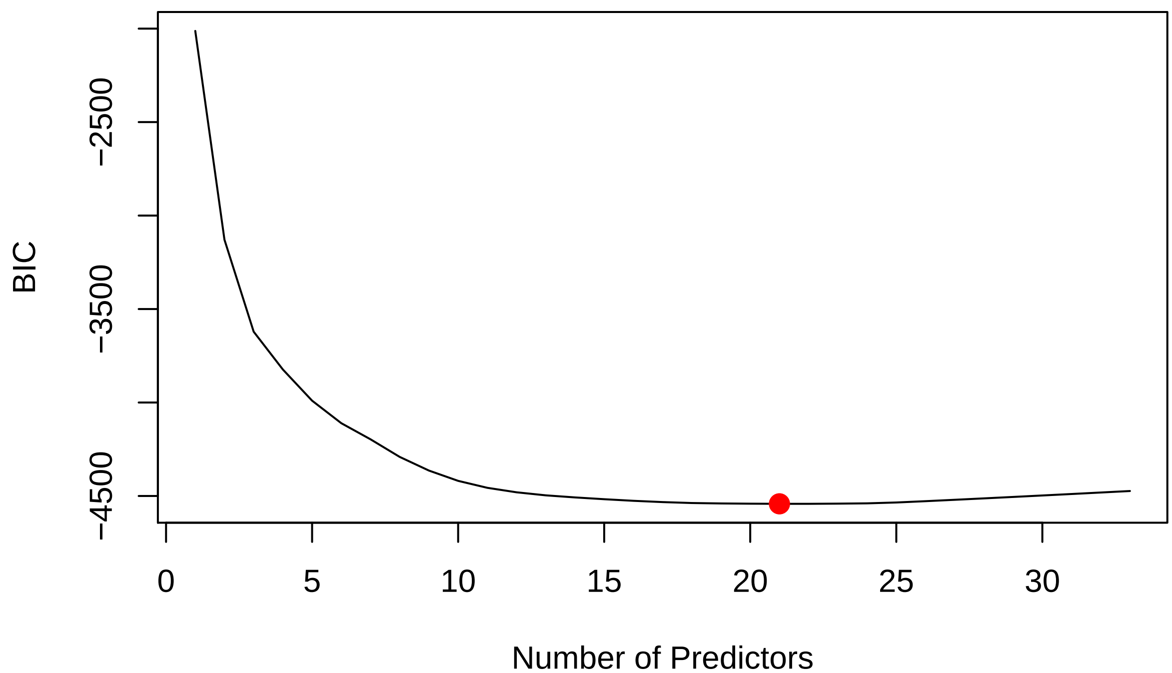
The best model, that is, a model that produces the least RSS is the model that uses 33 predictors. The coefficients of these are as follows:

```r
coef(res, min_rss)
```

```
##       (Intercept)       Lot_Frontage           Lot_Area         Year_Built
##     -1.142977e+07       8.737532e+01       3.141331e-01       3.845931e+02
##    Year_Remod_Add       Mas_Vnr_Area       BsmtFin_SF_1       BsmtFin_SF_2
##      5.129858e+02       3.794721e+01       3.002994e+02      -1.338433e+01
##       Bsmt_Unf_SF      Total_Bsmt_SF      First_Flr_SF    Low_Qual_Fin_SF
##     -1.337146e+01       3.759189e+01       3.554565e-01      -4.417005e+01
##    Bsmt_Full_Bath     Bsmt_Half_Bath          Full_Bath          Half_Bath
##      6.504458e+03      -1.883312e+03       1.949198e+03      -3.471763e+03
##     Bedroom_AbvGr       Kitchen_AbvGr      TotRms_AbvGrd         Fireplaces
##     -1.034286e+04      -3.360632e+04       4.068734e+03       7.084818e+03
##       Garage_Cars        Garage_Area       Wood_Deck_SF      Open_Porch_SF
##      7.737977e+03       2.082670e+01       2.430170e+01      -4.100172e+00
##    Enclosed_Porch Three_season_porch       Screen_Porch          Pool_Area
##      2.974408e+01       8.723251e+00       6.200042e+01      -6.447100e+01
##          Misc_Val            Mo_Sold          Year_Sold          Longitude
##     -9.497111e+00       2.762025e+01      -9.346976e+02      -1.299076e+04
##          Latitude        Gr_Liv_Area
##      2.464128e+05       6.324190e+01
```

BIC Plot (Forward Selection) Plotting the BIC of each Model (Forward Selection)

```
plot(smm$bic,xlab="Number of Predictors", ylab="BIC", type='l')
points(min_bic, smm$bic[min_bic], col="red", cex=2, pch=20)
```



The best model, that is, a model that produces the least BIC is the model that uses 21 predictors. The coefficients of these are as follows:

```
coef(res, min_bic)
```

```
##    (Intercept)   Lot_Frontage       Lot_Area     Year_Built Year_Remod_Add
##  -1.804094e+06   9.403297e+01   2.439368e-01   3.616190e+02   5.689112e+02
##   Mas_Vnr_Area   BsmtFin_SF_2    Bsmt_Unf_SF  Total_Bsmt_SF Bsmt_Full_Bath
##   4.363806e+01  -1.280552e+01  -1.309842e+01   4.126980e+01   6.192556e+03
## Bsmt_Half_Bath   Kitchen_AbvGr  TotRms_AbvGrd     Fireplaces    Garage_Cars
##  -4.186852e+03  -3.385257e+04   5.606576e+02   9.867642e+03   1.004416e+04
##    Garage_Area   Wood_Deck_SF  Open_Porch_SF      Pool_Area       Misc_Val
##   2.165199e+01   1.963979e+01   1.895785e+00  -5.499532e+01  -9.029755e+00
##        Mo_Sold    Gr_Liv_Area
##   9.536313e+01   5.928065e+01
```

Backward Selection

Data Initialization and setting up the variables

```r
ames        <- AmesHousing::make_ames()
numericVars <- ames %>% summarise_all(is.numeric) %>% unlist()
ames        <- ames[, numericVars]
NumCols     <- ncol(ames)

res <- regsubsets(Sale_Price ~ .,data=ames, method = "backward",  nvmax=NumCols)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 1 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
## Warning in rval$lopt[] <- rval$vorder[rval$lopt]: number of items to replace is
## not a multiple of replacement length
```

```r
smm <- summary(res)
smm$rss
```

```
##  [1] 1.146822e+13 7.869601e+12 5.693659e+12 5.277521e+12 4.915896e+12
##  [6] 4.671204e+12 4.481447e+12 4.314304e+12 4.179940e+12 4.092241e+12
## [11] 4.002680e+12 3.946271e+12 3.902998e+12 3.867500e+12 3.842522e+12
## [16] 3.819776e+12 3.796777e+12 3.777550e+12 3.763157e+12 3.750030e+12
## [21] 3.738591e+12 3.727819e+12 3.718299e+12 3.711271e+12 3.707002e+12
## [26] 3.704526e+12 3.703298e+12 3.702482e+12 3.701936e+12 3.701699e+12
## [31] 3.701509e+12 3.701367e+12 3.701352e+12
```

```r
min_rss <- which.min(smm$rss)
min_bic <- which.min(smm$bic)

min_rss
```

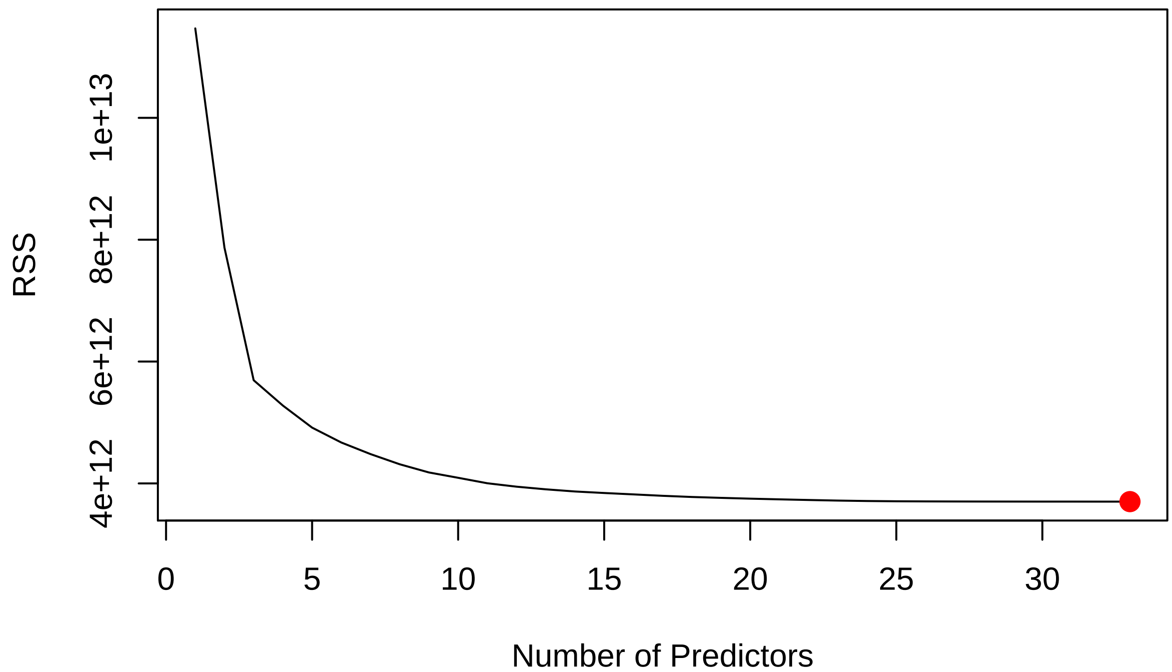
```
## [1] 33
```

```r
min_bic
```

```
## [1] 22
```

RSS Plot (Backward Selection) Plotting the RSS of each Model (Backward Selection)

```r
plot(smm$rss,xlab="Number of Predictors", ylab="RSS", type='l')
points(min_rss, smm$rss[min_rss], col="red", cex=2, pch=20)
```



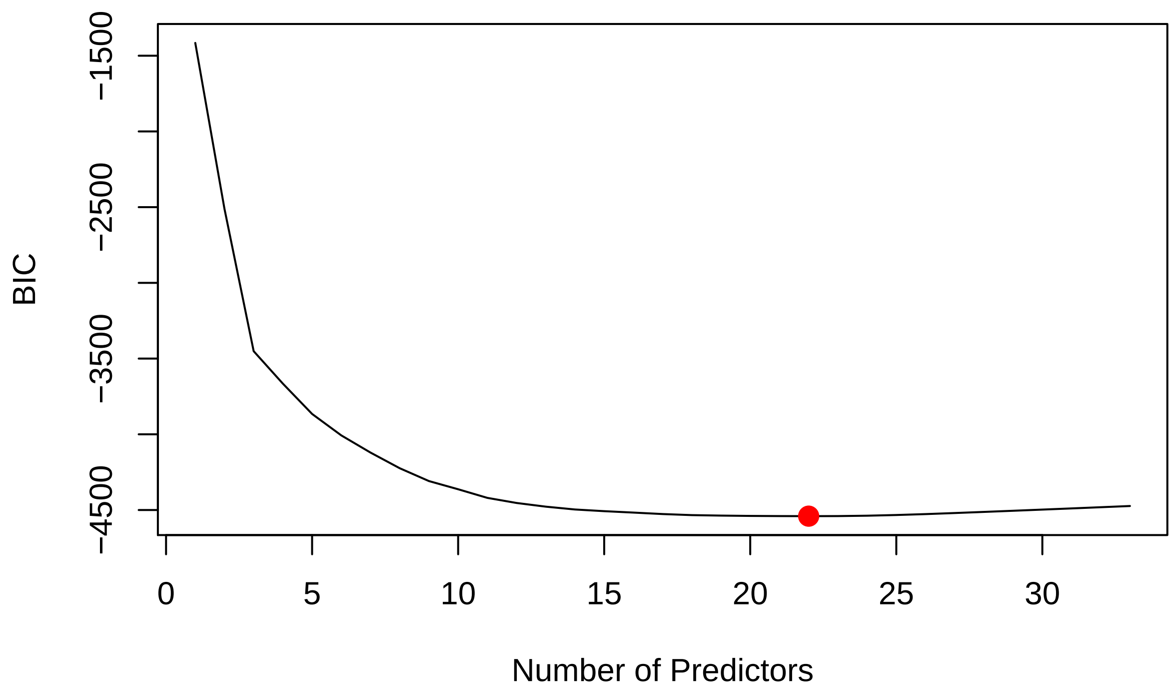
The best model, that is, a model that produces the least RSS is the model that uses 33 predictors. The coefficients of these are as follows:

```
coef(res, min_rss)
```

```
##       (Intercept)       Lot_Frontage           Lot_Area         Year_Built
##     -1.170805e+07       8.688692e+01       3.250816e-01       3.915167e+02
##    Year_Remod_Add       Mas_Vnr_Area        BsmtFin_SF_1       BsmtFin_SF_2
##      5.250215e+02       3.754647e+01       1.414811e+02      -1.391134e+01
##       Bsmt_Unf_SF      Total_Bsmt_SF       First_Flr_SF      Second_Flr_SF
##     -1.797736e+01       4.219896e+01       6.308277e+01       6.342274e+01
##   Low_Qual_Fin_SF     Bsmt_Half_Bath          Full_Bath          Half_Bath
##      1.994256e+01      -4.985513e+03       1.170822e+03      -3.889125e+03
##      Bedroom_AbvGr      Kitchen_AbvGr      TotRms_AbvGrd         Fireplaces
##     -1.045933e+04      -3.204082e+04       4.031002e+03       7.123055e+03
##       Garage_Cars        Garage_Area       Wood_Deck_SF      Open_Porch_SF
##      8.075298e+03       1.987748e+01       2.550571e+01      -2.347879e+00
##    Enclosed_Porch Three_season_porch       Screen_Porch          Pool_Area
##      3.067302e+01       9.134332e+00       6.239160e+01      -6.435958e+01
##          Misc_Val            Mo_Sold          Year_Sold          Longitude
##     -9.835393e+00       4.225967e+01      -8.848423e+02      -1.570146e+04
##          Latitude        Gr_Liv_Area
##      2.437618e+05       0.000000e+00
```

BIC Plot (Backward Selection) Plotting the BIC of each Model (Backward Selection)

```r
plot(smm$bic,xlab="Number of Predictors", ylab="BIC", type='l')
points(min_bic, smm$bic[min_bic], col="red", cex=2, pch=20)
```



The best model, that is, a model that produces the least BIC is the model that uses 22 predictors. The coefficients of these are as follows:

```r
coef(res, min_bic)
```

```
##    (Intercept)   Lot_Frontage       Lot_Area     Year_Built Year_Remod_Add
##  -1.816554e+06   8.988071e+01   2.252629e-01   3.568215e+02   5.800933e+02
##   Mas_Vnr_Area   BsmtFin_SF_2    Bsmt_Unf_SF  Total_Bsmt_SF   First_Flr_SF
##   4.228330e+01  -1.357431e+01  -1.785513e+01   4.275903e+01   4.067518e+01
##  Second_Flr_SF Bsmt_Half_Bath  Kitchen_AbvGr  TotRms_AbvGrd     Fireplaces
##   3.517944e+01  -7.244059e+03  -3.430550e+04   6.485669e+02   9.556181e+03
##    Garage_Cars    Garage_Area   Wood_Deck_SF  Open_Porch_SF      Pool_Area
##   1.024918e+04   2.015845e+01   2.092499e+01   3.072430e+00  -5.526520e+01
##       Misc_Val        Mo_Sold    Gr_Liv_Area
##  -9.437835e+00   9.540143e+01   2.300514e+01
```