Team 2 - Homework Two

Assignment 2: KJ 7.2; KJ 7.5

Bethany Poulin

November 1, 2019

```
knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE, comment = NA)
```

Dependencies

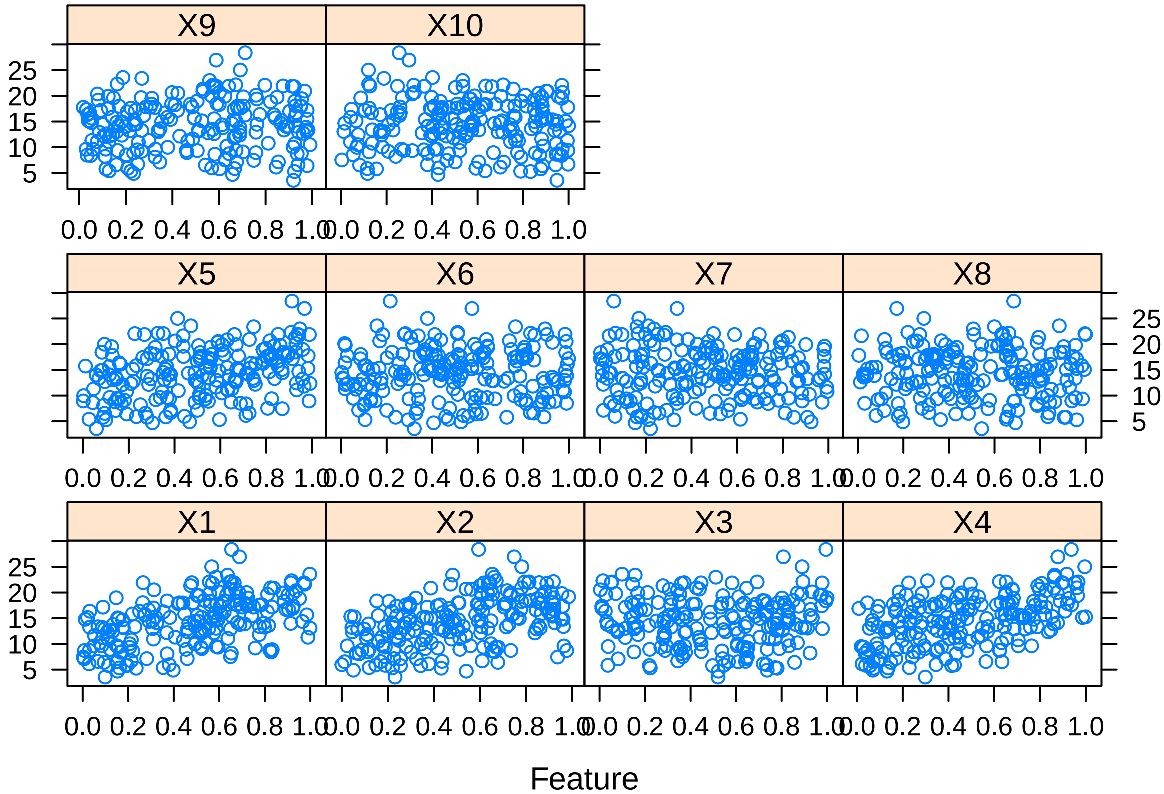
```
# predictive modeling
libraries('mlbench', 'caret')

# Formatting Libraries
libraries('default', 'knitr', 'kableExtra')

# Plotting Libraries
libraries('ggplot2', 'grid', 'ggfortify')
```

Kuhn & Johnson 7.2

> Friedman (1991) introduced several benchmark data sets create by simulation. One of these simulations used the following nonlinear equation to create data: $y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + N(0, \sigma^2)$; where the x values are random variables uniformly distributed between $[0, 1]$ (there are also 5 other non-informative variables also created in the simulation).

The package `mlbench` contains a function called `mlbench.friedman1` that simulates these data:

```
set.seed(200)
trainingData <- mlbench.friedman1(200, sd = 1)

## We convert the 'x' data from a matrix to a data frame
## One reason is that this will give the columns names.

trainingData$x <- data.frame(trainingData$x)

## Look at the data using
featurePlot(trainingData$x, trainingData$y)
```




```r
## or other methods.

## This creates a list with a vector 'y' and a matrix
## of predictors 'x'. Also simulate a large test set to
## estimate the true error rate with good precision:

testData <- mlbench.friedman1(5000, sd = 1)
testData$x <- data.frame(testData$x)
```

(a) Tune several models on these data. For example:

KNN

```r
knnTuned_72 <- train(x = trainingData$x,
                     y = trainingData$y,
                     method = "knn",
                     preProc = c("center", "scale"),
                     tuneLength = 10)
knnTuned_72
```

```
k-Nearest Neighbors

200 samples
 10 predictor

Pre-processing: centered (10), scaled (10)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 200, 200, 200, 200, 200, 200, ...
Resampling results across tuning parameters:
```

```
  k   RMSE      Rsquared   MAE     
   5  3.466085  0.5121775  2.816838
   7  3.349428  0.5452823  2.727410
   9  3.264276  0.5785990  2.660026
  11  3.214216  0.6024244  2.603767
  13  3.196510  0.6176570  2.591935
  15  3.184173  0.6305506  2.577482
  17  3.183130  0.6425367  2.567787
  19  3.198752  0.6483184  2.592683
  21  3.188993  0.6611428  2.588787
  23  3.200458  0.6638353  2.604529

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 17.
```

```r
knnPred_72 <- predict(knnTuned_72, newdata = testData$x)
postResample(pred = knnPred_72, obs = testData$y)
```

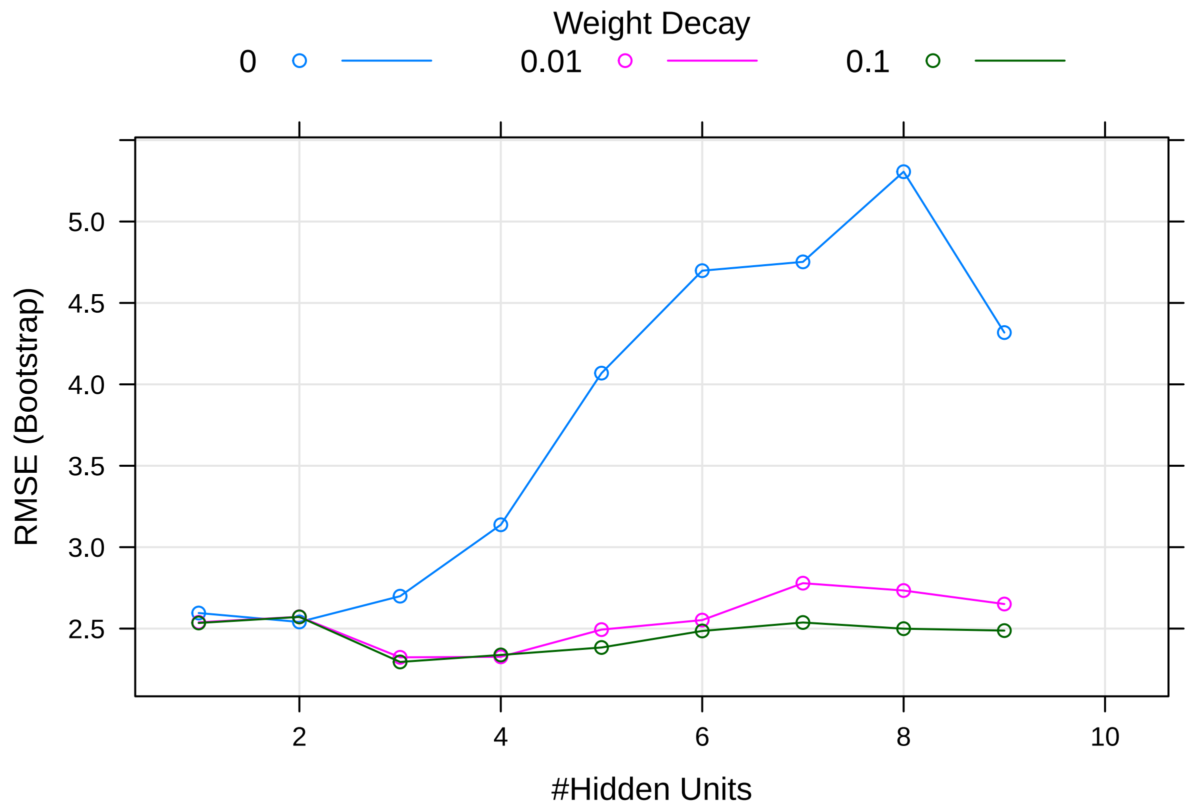
```
     RMSE  Rsquared       MAE 
3.2040595 0.6819919 2.5683461 
```

```r
varImp(knnTuned_72)
```

```
loess r-squared variable importance

     Overall
X4  100.0000
X1   95.5047
X2   89.6186
X5   45.2170
X3   29.9330
X9    6.3299
X10   5.5182
X8    3.2527
X6    0.8884
X7    0.0000
```

Neural Net

```r
nnetGrid_72 <- expand.grid(.decay = c(0, 0.01, .1),
                        .size = c(1:10),
                        .bag = FALSE)
set.seed(100)
nnetTune_72 <- train(trainingData$x,  trainingData$y,
                 method = "avNNet",
                 tuneGrid = nnetGrid_72,
                 preProc = c("center", "scale"),
                 linout = TRUE,
                 trace = FALSE,
                 MaxNWts = 10 * (ncol(trainingData$x) + 1) + 5 + 1,
                 maxit = 500)


plot(nnetTune_72)
```



Minimum RMSE from Grid: 2.2948782

```r
print(nnetTune_72)
```

```
Model Averaged Neural Network

200 samples
 10 predictor

Pre-processing: centered (10), scaled (10)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 200, 200, 200, 200, 200, 200, ...
Resampling results across tuning parameters:

  decay  size  RMSE      Rsquared   MAE
  0.00    1    2.595448  0.7409360  2.039665
  0.00    2    2.540665  0.7465990  2.004844
  0.00    3    2.698990  0.7467573  2.006059
  0.00    4    3.137447  0.6727711  2.237839
  0.00    5    4.068883  0.5620574  2.733698
  0.00    6    4.698198  0.5123062  3.084489
  0.00    7    4.752135  0.4771393  3.167245
  0.00    8    5.306007  0.4536278  3.249912
  0.00    9    4.317862  0.5733699  2.716157
  0.00   10         NaN        NaN       NaN
  0.01    1    2.538554  0.7499244  1.988860
  0.01    2    2.572420  0.7422020  2.032978
  0.01    3    2.323519  0.7875812  1.844155
  0.01    4    2.326440  0.7860554  1.854140
```

```
  0.01    5    2.493551  0.7596522  1.993350
  0.01    6    2.552077  0.7507789  2.035192
  0.01    7    2.778792  0.7114431  2.205708
  0.01    8    2.733604  0.7217070  2.165101
  0.01    9    2.650741  0.7312514  2.112637
  0.01   10         NaN        NaN       NaN
  0.10    1    2.534358  0.7496485  1.980592
  0.10    2    2.571641  0.7410121  2.042609
  0.10    3    2.294878  0.7900421  1.829704
  0.10    4    2.338091  0.7856999  1.857482
  0.10    5    2.383784  0.7772791  1.887853
  0.10    6    2.485410  0.7616765  1.974917
  0.10    7    2.537340  0.7563561  2.000942
  0.10    8    2.499243  0.7553886  1.989252
  0.10    9    2.487433  0.7604963  1.981012
  0.10   10         NaN        NaN       NaN

Tuning parameter 'bag' was held constant at a value of FALSE
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were size = 3, decay = 0.1 and bag
 = FALSE.
```

```r
nnetFit_72 <- nnet(trainingData$x,
                trainingData$y,
                size = 3,
                decay = 0.1,
                linout = TRUE,
                trace = FALSE,
                maxit = 500,
                MaxNWts = 5 * (ncol(trainingData$x) + 1) + 5 + 1)

nnetPred_72 <- predict(nnetFit_72, newdata = testData$x)
postResample(pred = nnetPred_72, obs = testData$y)
```

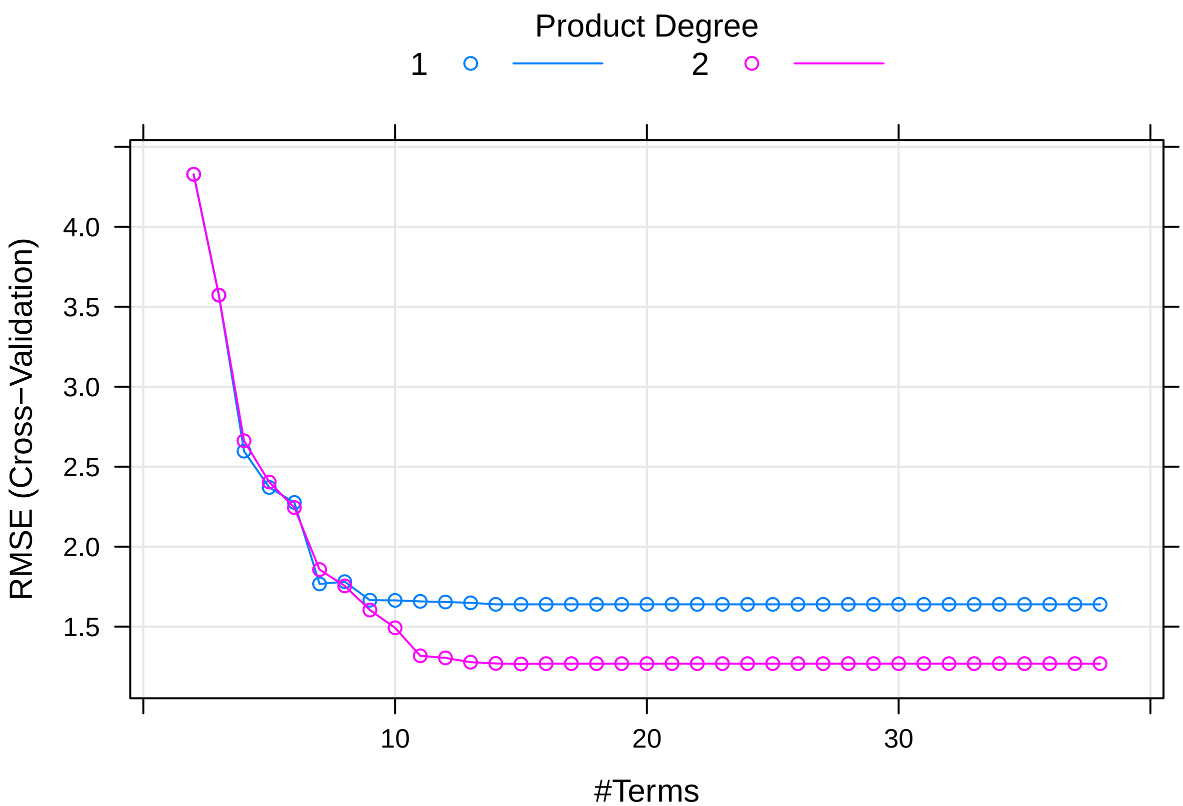
```
     RMSE  Rsquared       MAE
2.4016742 0.7686684 1.8163326
```

```r
varImp(nnetTune_72)
```

```
loess r-squared variable importance

     Overall
X4  100.0000
X1   95.5047
X2   89.6186
X5   45.2170
X3   29.9330
X9    6.3299
X10   5.5182
X8    3.2527
X6    0.8884
X7    0.0000
```

MARS

```
marsGrid_72 <- expand.grid(.degree = 1:2, .nprune = 2:38)
# Fix the seed so that the results can be reproduced
set.seed(100)
marsTuned_72 <- train(trainingData$x,
                   trainingData$y,
                   method = "earth",
                   tuneGrid = marsGrid_72,
                   trControl = trainControl(method = "cv"))

plot(marsTuned_72)
```




```
marsTuned_72
```

```
Multivariate Adaptive Regression Spline

200 samples
 10 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 180, 180, 180, 180, 180, 180, ...
Resampling results across tuning parameters:

  degree  nprune  RMSE      Rsquared   MAE
  1        2      4.327937  0.2544880  3.600474
  1        3      3.572450  0.4912720  2.895811
  1        4      2.596841  0.7183600  2.106341
```

1	5	2.370161	0.7659777	1.918669
1	6	2.276141	0.7881481	1.810001
1	7	1.766728	0.8751831	1.390215
1	8	1.780946	0.8723243	1.401345
1	9	1.665091	0.8819775	1.325515
1	10	1.663804	0.8821283	1.327657
1	11	1.657738	0.8822967	1.331730
1	12	1.653784	0.8827903	1.331504
1	13	1.648496	0.8823663	1.316407
1	14	1.639073	0.8841742	1.312833
1	15	1.639073	0.8841742	1.312833
1	16	1.639073	0.8841742	1.312833
1	17	1.639073	0.8841742	1.312833
1	18	1.639073	0.8841742	1.312833
1	19	1.639073	0.8841742	1.312833
1	20	1.639073	0.8841742	1.312833
1	21	1.639073	0.8841742	1.312833
1	22	1.639073	0.8841742	1.312833
1	23	1.639073	0.8841742	1.312833
1	24	1.639073	0.8841742	1.312833
1	25	1.639073	0.8841742	1.312833
1	26	1.639073	0.8841742	1.312833
1	27	1.639073	0.8841742	1.312833
1	28	1.639073	0.8841742	1.312833
1	29	1.639073	0.8841742	1.312833
1	30	1.639073	0.8841742	1.312833
1	31	1.639073	0.8841742	1.312833
1	32	1.639073	0.8841742	1.312833
1	33	1.639073	0.8841742	1.312833
1	34	1.639073	0.8841742	1.312833
1	35	1.639073	0.8841742	1.312833
1	36	1.639073	0.8841742	1.312833
1	37	1.639073	0.8841742	1.312833
1	38	1.639073	0.8841742	1.312833
2	2	4.327937	0.2544880	3.600474
2	3	3.572450	0.4912720	2.895811
2	4	2.661826	0.7070510	2.173471
2	5	2.404015	0.7578971	1.975387
2	6	2.243927	0.7914805	1.783072
2	7	1.856336	0.8605482	1.435682
2	8	1.754607	0.8763186	1.396841
2	9	1.603578	0.8938666	1.261361
2	10	1.492421	0.9084998	1.168700
2	11	1.317350	0.9292504	1.033926
2	12	1.304327	0.9320133	1.019108
2	13	1.277510	0.9323681	1.002927
2	14	1.269626	0.9350024	1.003346
2	15	1.266217	0.9359400	1.013893
2	16	1.268470	0.9354868	1.011414
2	17	1.268470	0.9354868	1.011414
2	18	1.268470	0.9354868	1.011414
2	19	1.268470	0.9354868	1.011414
2	20	1.268470	0.9354868	1.011414
2	21	1.268470	0.9354868	1.011414

```
  2       22      1.268470  0.9354868  1.011414
  2       23      1.268470  0.9354868  1.011414
  2       24      1.268470  0.9354868  1.011414
  2       25      1.268470  0.9354868  1.011414
  2       26      1.268470  0.9354868  1.011414
  2       27      1.268470  0.9354868  1.011414
  2       28      1.268470  0.9354868  1.011414
  2       29      1.268470  0.9354868  1.011414
  2       30      1.268470  0.9354868  1.011414
  2       31      1.268470  0.9354868  1.011414
  2       32      1.268470  0.9354868  1.011414
  2       33      1.268470  0.9354868  1.011414
  2       34      1.268470  0.9354868  1.011414
  2       35      1.268470  0.9354868  1.011414
  2       36      1.268470  0.9354868  1.011414
  2       37      1.268470  0.9354868  1.011414
  2       38      1.268470  0.9354868  1.011414

RMSE was used to select the optimal model using the smallest value.
The final values used for the model were nprune = 15 and degree = 2.
```

Minimum RMSE: 1.2662173

```r
marsPred_72 <- predict(marsTuned_72, newdata = testData$x)
postResample(pred = marsPred_72, obs = testData$y)
```

```
     RMSE  Rsquared       MAE 
1.1589948 0.9460418 0.9250230 
```

```r
varImp(marsTuned_72)
```

```
earth variable importance

    Overall
X1   100.00
X4    85.14
X2    69.24
X5    49.31
X3    40.00
X9     0.00
X6     0.00
X8     0.00
X7     0.00
X10    0.00
```

Support Vector Machine

```r
svmRTuned_72 <- train(trainingData$x,
                   trainingData$y,
                   method = "svmRadial",
                   preProc = c("center", "scale"),
                   tuneLength = 14,
                   trControl = trainControl(method = "cv"))

svmRTuned_72
```

```
Support Vector Machines with Radial Basis Function Kernel

200 samples
 10 predictor

Pre-processing: centered (10), scaled (10)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 180, 180, 180, 180, 180, 180, ...
Resampling results across tuning parameters:

  C        RMSE      Rsquared   MAE
     0.25  2.536604  0.7865906  2.035796
     0.50  2.262783  0.8033031  1.800168
     1.00  2.087501  0.8225671  1.636606
     2.00  1.973976  0.8359125  1.540666
     4.00  1.890687  0.8494370  1.489167
     8.00  1.837229  0.8573247  1.465234
    16.00  1.830431  0.8587775  1.459156
    32.00  1.830431  0.8587775  1.459156
    64.00  1.830431  0.8587775  1.459156
   128.00  1.830431  0.8587775  1.459156
   256.00  1.830431  0.8587775  1.459156
   512.00  1.830431  0.8587775  1.459156
  1024.00  1.830431  0.8587775  1.459156
  2048.00  1.830431  0.8587775  1.459156

Tuning parameter 'sigma' was held constant at a value of 0.06450665
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were sigma = 0.06450665 and C = 16.
```

```r
svmRTuned_72$finalModel
```

```
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 16

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.064506652354808

Number of Support Vectors : 151

Objective Function Value : -71.0731
Training error : 0.008521
```

```r
svmPred_72 <- predict(svmRTuned_72, newdata = testData$x)
postResample(pred = svmPred_72, obs = testData$y)
```

```
     RMSE  Rsquared       MAE
2.0772741 0.8250955 1.5779991
```

```r
##  RMSE  Rsquared       MAE
##  2.0421971 0.8424968 1.5994147
varImp(svmRTuned_72)
```

```
loess r-squared variable importance
```

```
     Overall
X4  100.0000
X1   95.5047
X2   89.6186
X5   45.2170
X3   29.9330
X9    6.3299
X10   5.5182
X8    3.2527
X6    0.8884
X7    0.0000
```

(b) Which models appear to give the best performance?

Model Analysis

KKN

```r
postResample(pred = knnPred_72, obs = testData$y)
```

```
     RMSE  Rsquared       MAE
3.2040595 0.6819919 2.5683461
```

Neural Network

```r
postResample(pred = nnetPred_72, obs = testData$y)
```

```
     RMSE  Rsquared       MAE
2.4016742 0.7686684 1.8163326
```

MARS

```r
postResample(pred = marsPred_72, obs = testData$y)
```

```
     RMSE  Rsquared       MAE
1.1589948 0.9460418 0.9250230
```

SVM

```r
postResample(pred = svmPred_72, obs = testData$y)
```

```
     RMSE  Rsquared       MAE
2.0772741 0.8250955 1.5779991
```

Based on the RMSE and R-Squared, the Multiple Adaptive Regression Splines is the best with the RMSE almost have as small as the next best model, the Support Vector Machine Regression and the r-squared is about 95% suggesting that the MARS model explains a fair bit more of the outcome variable.

b. Does MARS select the informative predictors (those named X1-X5)?

MARS importance:

```r
varImp(marsTuned_72)
```

```
earth variable importance

    Overall
X1   100.00
```

```
X4    85.14
X2    69.24
X5    49.31
X3    40.00
X10    0.00
X6     0.00
X9     0.00
X7     0.00
X8     0.00
```

As well as this being the most predictive of the optimized and tuned models, the MARS also ranks `X1-X5` the most important ordered, with `X6-X10` not contributing at all to the variable importance.

It is very likely that lack of contribution alloted to the `X6-X10` variables which bolster the R-Squared and RMSE performance and noise from these variables did not reduce the predictive strength of this model as it does in small quantities in the other three models.

7.5

```r
data('ChemicalManufacturingProcess')
# Total NA Values
na_table<- table(is.na(ChemicalManufacturingProcess))
total_na<-sapply(ChemicalManufacturingProcess[2:57], function(x) sum(is.na(x)))
na_table<-sapply(ChemicalManufacturingProcess, function(x) table(is.na(x)))
total_na<- data.frame(sort(total_na, decreasing = TRUE))
total_na<- cbind(Variable = rownames(total_na), total_na)
rownames(total_na) <- 1:nrow(total_na)
colnames(total_na)<-  c("Variable", "Count")
total_na<-cbind(total_na[1:28,],total_na[29:56,])
hist_yield <-ggplot(ChemicalManufacturingProcess, aes(x = Yield))+
    geom_histogram(colour ='black', fill = 'violetred4') +
    ggtitle('Distribution of Yield Chemical Manufacturing Process Data')
imputed_data = data.frame(impute.knn(as.matrix(ChemicalManufacturingProcess),
                                     k =10,
                                     rowmax =.30,
                                     colmax =.85,
                                     rng.seed =1942)$data)
head(imputed_data)
```

```
  Yield BiologicalMaterial01 BiologicalMaterial02 BiologicalMaterial03
1 38.00                 6.25                49.58                56.97
2 42.44                 8.01                60.97                67.48
3 42.03                 8.01                60.97                67.48
4 41.42                 8.01                60.97                67.48
5 42.49                 7.47                63.33                72.25
6 43.57                 6.12                58.36                65.31
  BiologicalMaterial04 BiologicalMaterial05 BiologicalMaterial06
1                12.74                19.51                43.73
2                14.65                19.36                53.14
3                14.65                19.36                53.14
4                14.65                19.36                53.14
5                14.02                17.91                54.66
6                15.17                21.79                51.23
  BiologicalMaterial07 BiologicalMaterial08 BiologicalMaterial09
1                  100                16.66                11.44
```

```
2                  100                 19.04                 12.55
3                  100                 19.04                 12.55
4                  100                 19.04                 12.55
5                  100                 18.22                 12.80
6                  100                 18.30                 12.13
  BiologicalMaterial10 BiologicalMaterial11 BiologicalMaterial12
1                 3.46               138.09                18.83
2                 3.46               153.67                21.05
3                 3.46               153.67                21.05
4                 3.46               153.67                21.05
5                 3.05               147.61                21.05
6                 3.78               151.88                20.76
  ManufacturingProcess01 ManufacturingProcess02 ManufacturingProcess03
1                   9.18                   6.19               1.558000
2                   0.00                   0.00               1.543333
3                   0.00                   0.00               1.542857
4                   0.00                   0.00               1.542857
5                  10.70                   0.00               1.545000
6                  12.00                   0.00               1.550000
  ManufacturingProcess04 ManufacturingProcess05 ManufacturingProcess06
1                  927.8                1023.64                  207.7
2                  917.0                1032.20                  210.0
3                  912.0                1003.60                  207.1
4                  911.0                1014.60                  213.3
5                  918.0                1027.50                  205.7
6                  924.0                1016.80                  208.9
  ManufacturingProcess07 ManufacturingProcess08 ManufacturingProcess09
1                  177.4                  177.2                  43.00
2                  177.0                  178.0                  46.57
3                  178.0                  178.0                  45.07
4                  177.0                  177.0                  44.92
5                  178.0                  178.0                  44.96
6                  178.0                  178.0                  45.32
  ManufacturingProcess10 ManufacturingProcess11 ManufacturingProcess12
1               9.212500               9.837500                      0
2               9.414286               9.614286                      0
3               9.271429               9.357143                      0
4               9.171429               9.414286                      0
5               8.755556               9.566667                      0
6               8.911111               9.811111                      0
  ManufacturingProcess13 ManufacturingProcess14 ManufacturingProcess15
1                   35.5                   4898                   6108
2                   34.0                   4869                   6095
3                   34.8                   4878                   6087
4                   34.8                   4897                   6102
5                   34.6                   4992                   6233
6                   34.0                   4985                   6222
  ManufacturingProcess16 ManufacturingProcess17 ManufacturingProcess18
1                   4682                   35.5                   4865
2                   4617                   34.0                   4867
3                   4617                   34.8                   4877
4                   4635                   34.8                   4872
5                   4733                   33.9                   4886
6                   4786                   33.4                   4862
```

```
  ManufacturingProcess19 ManufacturingProcess20 ManufacturingProcess21
1                   6049                   4665                    0.0
2                   6097                   4621                    0.0
3                   6078                   4621                    0.0
4                   6073                   4611                    0.0
5                   6102                   4659                   -0.7
6                   6115                   4696                   -0.6
  ManufacturingProcess22 ManufacturingProcess23 ManufacturingProcess24
1                    5.5                    1.9                    6.9
2                    3.0                    0.0                    3.0
3                    4.0                    1.0                    4.0
4                    5.0                    2.0                    5.0
5                    8.0                    4.0                   18.0
6                    9.0                    1.0                    1.0
  ManufacturingProcess25 ManufacturingProcess26 ManufacturingProcess27
1                   4873                   6074                   4685
2                   4869                   6107                   4630
3                   4897                   6116                   4637
4                   4892                   6111                   4630
5                   4930                   6151                   4684
6                   4871                   6128                   4687
  ManufacturingProcess28 ManufacturingProcess29 ManufacturingProcess30
1                   10.7                   21.0                    9.9
2                   11.2                   21.4                    9.9
3                   11.1                   21.3                    9.4
4                   11.1                   21.3                    9.4
5                   11.3                   21.6                    9.0
6                   11.4                   21.7                   10.1
  ManufacturingProcess31 ManufacturingProcess32 ManufacturingProcess33
1                   69.1                    156                     66
2                   68.7                    169                     66
3                   69.3                    173                     66
4                   69.3                    171                     68
5                   69.4                    171                     70
6                   68.2                    173                     70
  ManufacturingProcess34 ManufacturingProcess35 ManufacturingProcess36
1                    2.4                    486                  0.019
2                    2.6                    508                  0.019
3                    2.6                    509                  0.018
4                    2.5                    496                  0.018
5                    2.5                    468                  0.017
6                    2.5                    490                  0.018
  ManufacturingProcess37 ManufacturingProcess38 ManufacturingProcess39
1                    0.5                      3                    7.2
2                    2.0                      2                    7.2
3                    0.7                      2                    7.2
4                    1.2                      2                    7.2
5                    0.2                      2                    7.3
6                    0.4                      2                    7.2
  ManufacturingProcess40 ManufacturingProcess41 ManufacturingProcess42
1                    0.0                   0.00                   11.6
2                    0.1                   0.15                   11.1
3                    0.0                   0.00                   12.0
4                    0.0                   0.00                   10.6
```

```
5                     0.0                   0.00                  11.0
6                     0.0                   0.00                  11.5
  ManufacturingProcess43 ManufacturingProcess44 ManufacturingProcess45
1                     3.0                    1.8                   2.4
2                     0.9                    1.9                   2.2
3                     1.0                    1.8                   2.3
4                     1.1                    1.8                   2.1
5                     1.1                    1.7                   2.1
6                     2.2                    1.8                   2.0
```

(a) Which nonlinear regression model gives the optimal resampling and test set performance?

KNN

```
set.seed(1492)   #  set seed to ensure you always have same random numbers generated
sample = sample.split(imputed_data, SplitRatio = 0.80) # splits the data in the ratio mentioned in Spli
trainingData =subset(imputed_data,sample ==TRUE) # creates a training dataset named train1 with rows wh
testData=subset(imputed_data, sample==FALSE)
x_train <-trainingData[,2:58]
x_test <- testData[,2:58]
knnFit_75 <- train(x = x_train,
                  y = trainingData$Yield,
                  method = "knn",
                  preProc = c("center", "scale"),
                  tuneLength = 10)

knnFit_75
```

```
k-Nearest Neighbors

140 samples
 57 predictor

Pre-processing: centered (57), scaled (57)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 140, 140, 140, 140, 140, 140, ...
Resampling results across tuning parameters:

  k   RMSE      Rsquared   MAE
   5  1.490422  0.3723029  1.168537
   7  1.460890  0.3913409  1.149121
   9  1.463084  0.3958924  1.154529
  11  1.448818  0.4064650  1.153667
  13  1.463806  0.3938749  1.164496
  15  1.471282  0.3890392  1.174310
  17  1.474635  0.3909431  1.174061
  19  1.477009  0.3926099  1.175082
  21  1.480839  0.3958046  1.176151
  23  1.484486  0.3958470  1.183146

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 11.
```

```r
knnPred_75 <- predict(knnFit_75, newdata = x_test)
postResample(pred = knnPred_75, obs = testData$Yield)
```

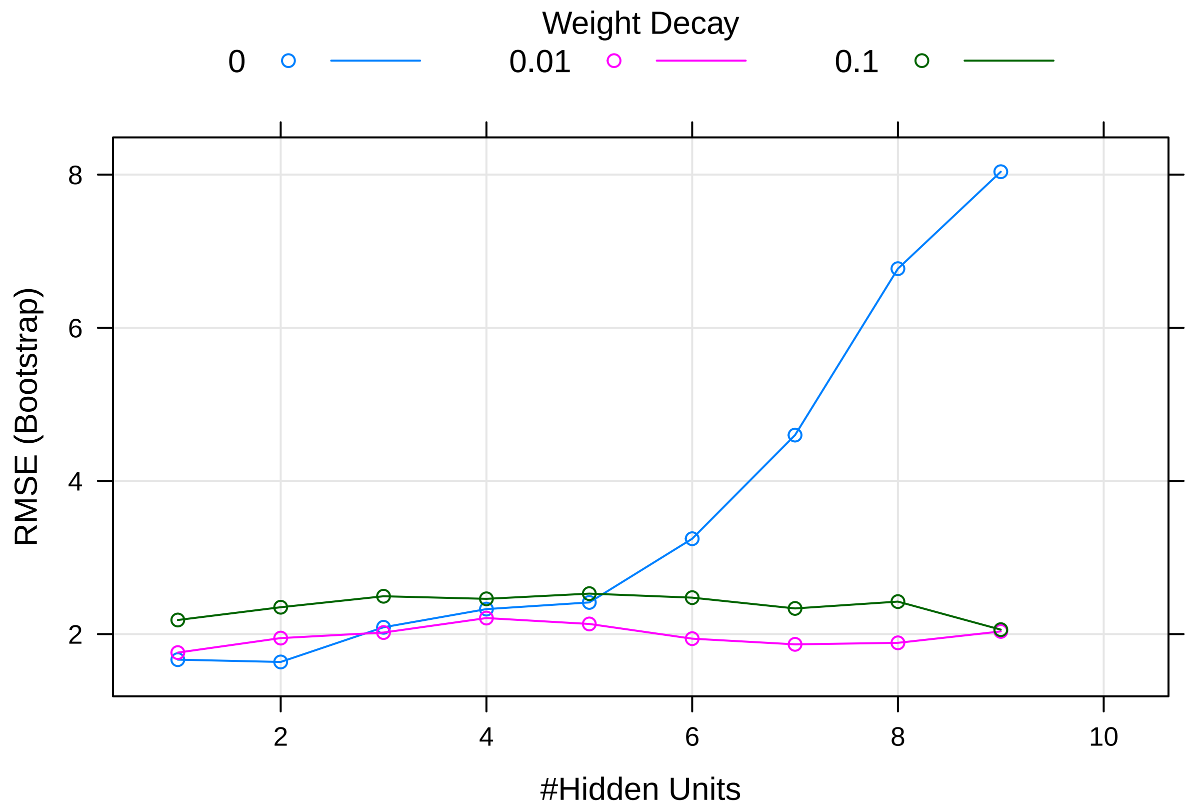
```
     RMSE  Rsquared       MAE 
1.1382855 0.6303354 0.9761111 
```

```r
varImp(knnFit_75)
```

```
loess r-squared variable importance

  only 20 most important variables shown (out of 57)

                       Overall
ManufacturingProcess32  100.00
ManufacturingProcess13   80.04
BiologicalMaterial06     79.67
ManufacturingProcess17   78.78
BiologicalMaterial03     70.18
ManufacturingProcess36   69.17
ManufacturingProcess09   60.62
BiologicalMaterial02     60.16
BiologicalMaterial12     53.67
ManufacturingProcess31   51.71
ManufacturingProcess33   51.30
ManufacturingProcess29   48.47
ManufacturingProcess06   46.76
BiologicalMaterial04     42.96
BiologicalMaterial11     40.95
ManufacturingProcess02   34.47
BiologicalMaterial01     33.22
BiologicalMaterial09     33.19
BiologicalMaterial08     32.55
ManufacturingProcess11   32.49
```

Neural Network

```r
nnetGrid_75 <- expand.grid(.decay = c(0, 0.01, .1),
                        .size = c(1:10),
                        .bag = FALSE)
set.seed(100)
nnetTune_75 <- train(x = x_train,  trainingData$Yield,
                  method = "avNNet",
                  tuneGrid = nnetGrid_75,
                  preProc = c("center", "scale"),
                  linout = TRUE,
                  trace = FALSE,
                  MaxNWts = 10 * (ncol(x_train) + 1) + 5 + 1,
                  maxit = 500)
plot(nnetTune_75)
```




```
min(nnetTune_75$results$RMSE, na.rm = TRUE)
```

```
[1] 1.636137
```

```
#         decay  size  RMSE      Rsquared   MAE
# Chosen  0.10    3    2.332714  0.7655842  1.808308



nnetFit_75 <- nnet(x = x_train,
                   trainingData$Yield,
                   size = 3,
                   decay = 0.1,
                   linout = TRUE,
                   trace = FALSE,
                   maxit = 500,
                   MaxNWts = 5 * (ncol(x_train) + 1) + 5 + 1)
nnetFit_75
```

```
a 57-3-1 network with 178 weights
options were - linear output units  decay=0.1
```

```
nnetPred_75 <- predict(nnetFit_75, newdata = x_test)
postResample(pred = nnetPred_75, obs = testData$Yield)
```

```
     RMSE  Rsquared       MAE
2.1814738 0.2613588 1.6469824
```

```
varImp(nnetFit_75)
```

```
       Overall
```

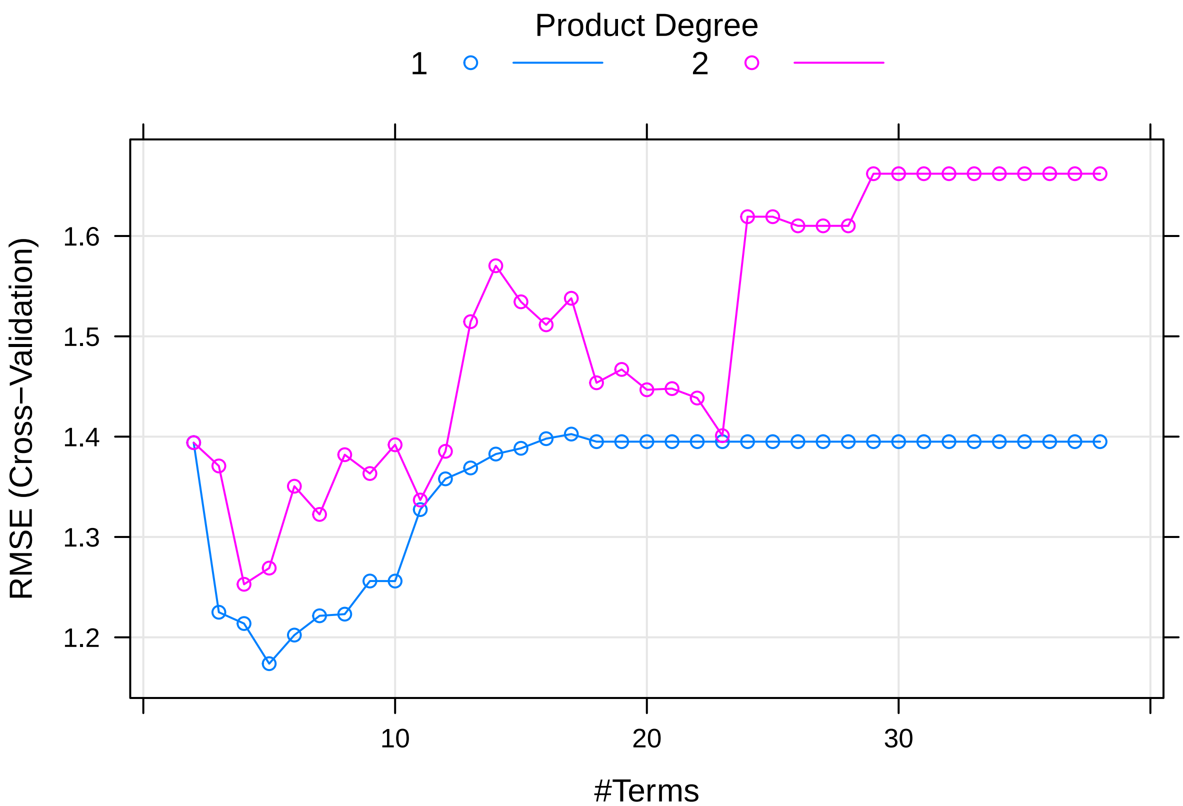
```
X1  4.36650476
X2  0.53219770
X3  0.68777555
X4  1.07541772
X5  0.54541904
X6  0.54358106
X7  3.10799691
X8  1.20928574
X9  3.16642022
X10 2.79400979
X11 0.37165273
X12 2.19095442
X13 0.46256953
X14 0.59068255
X15 0.19050264
X16 0.71035413
X17 0.93414101
X18 0.39104015
X19 1.62311491
X20 1.97808750
X21 1.07315358
X22 2.49632937
X23 2.92130430
X24 4.94306678
X25 0.64581902
X26 1.58709567
X27 3.75140543
X28 0.55278083
X29 0.76125377
X30 1.52901560
X31 3.25262866
X32 3.19648192
X33 0.54860958
X34 0.27970471
X35 1.25194720
X36 0.66233144
X37 4.41778532
X38 3.17016456
X39 2.69809567
X40 0.69025903
X41 4.26589434
X42 1.36976079
X43 1.14696334
X44 1.40168384
X45 1.75947493
X46 3.84105549
X47 1.52715888
X48 0.03451682
X49 1.27752395
X50 2.55382216
X51 0.38730581
X52 3.98754345
X53 3.65753004
X54 0.99242060
```

```
X55 0.39972062
X56 1.16254009
X57 2.33414434
```

MARS Model

```r
# Define the candidate models to test
marsGrid_75 <- expand.grid(.degree = 1:2, .nprune = 2:38)
# Fix the seed so that the results can be reproduced
set.seed(100)
marsTuned_75 <- train(x = x_train,
                      trainingData$Yield,
                      method = "earth",
                      tuneGrid = marsGrid_75,
                      trControl = trainControl(method = "cv"))
```

```r
plot(marsTuned_75)
```




```r
min(marsTuned_75$results$RMSE, na.rm = TRUE) # 1.181011
```

```
[1] 1.173717
```

```r
##   degree   nprune   RMSE       Rsquared    MAE
##    2        14       1.181011   0.9428116   0.9653660
marsPred_75 <- predict(marsTuned_75, newdata = x_test)
postResample(pred = marsPred_75, obs = testData$Yield)
```

```
     RMSE  Rsquared       MAE
1.0748296 0.6300397 0.9186156
```

```r
varImp(marsTuned_75)
```

```
earth variable importance

  only 20 most important variables shown (out of 57)

                      Overall
ManufacturingProcess32  100.00
ManufacturingProcess09   57.33
ManufacturingProcess13   22.71
BiologicalMaterial08      0.00
ManufacturingProcess22    0.00
ManufacturingProcess30    0.00
ManufacturingProcess24    0.00
ManufacturingProcess40    0.00
ManufacturingProcess43    0.00
ManufacturingProcess16    0.00
ManufacturingProcess21    0.00
ManufacturingProcess15    0.00
BiologicalMaterial06      0.00
BiologicalMaterial12      0.00
ManufacturingProcess08    0.00
ManufacturingProcess01    0.00
ManufacturingProcess07    0.00
ManufacturingProcess03    0.00
ManufacturingProcess38    0.00
ManufacturingProcess45    0.00
```

SVM

```r
svmRTuned_75 <- train(x = x_train,
                   trainingData$Yield,
                   method = "svmRadial",
                   preProc = c("center", "scale"),
                   tuneLength = 14,
                   trControl = trainControl(method = "cv"))

svmRTuned_75
```

```
Support Vector Machines with Radial Basis Function Kernel

140 samples
 57 predictor

Pre-processing: centered (57), scaled (57)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 126, 127, 124, 125, 125, 126, ...
Resampling results across tuning parameters:

  C        RMSE      Rsquared   MAE
     0.25  1.461709  0.5235456  1.1808494
     0.50  1.327649  0.5744012  1.0791040
     1.00  1.218655  0.6295296  0.9893988
     2.00  1.139970  0.6656411  0.9149801
```

```
     4.00  1.119078  0.6679289  0.8800893
     8.00  1.105754  0.6725463  0.8655476
    16.00  1.106299  0.6721960  0.8667174
    32.00  1.106299  0.6721960  0.8667174
    64.00  1.106299  0.6721960  0.8667174
   128.00  1.106299  0.6721960  0.8667174
   256.00  1.106299  0.6721960  0.8667174
   512.00  1.106299  0.6721960  0.8667174
  1024.00  1.106299  0.6721960  0.8667174
  2048.00  1.106299  0.6721960  0.8667174

Tuning parameter 'sigma' was held constant at a value of 0.01452438
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were sigma = 0.01452438 and C = 8.
```

```r
svmRTuned_75$finalModel
```

```
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 8

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.0145243761370095

Number of Support Vectors : 125

Objective Function Value : -83.9244
Training error : 0.009711
```

```r
svmPred_75 <- predict(svmRTuned_75, newdata = x_test)
postResample(pred = svmPred_75, obs = testData$Yield)
```

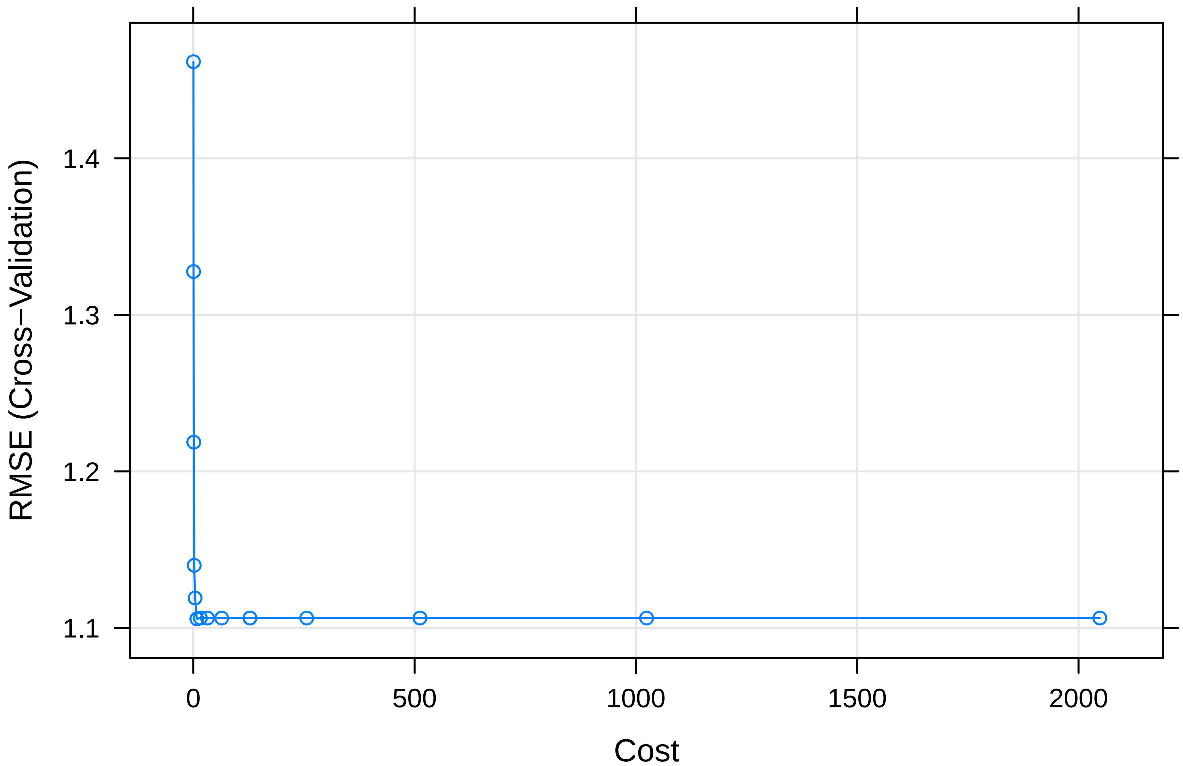
```
     RMSE  Rsquared       MAE
0.9868486 0.6751334 0.7751097
```

```r
##  RMSE  Rsquared       MAE
##  2.0421971 0.8424968 1.5994147
varImp(svmRTuned_75)
```

```
loess r-squared variable importance

  only 20 most important variables shown (out of 57)

                       Overall
ManufacturingProcess32  100.00
ManufacturingProcess13   80.04
BiologicalMaterial06     79.67
ManufacturingProcess17   78.78
BiologicalMaterial03     70.18
ManufacturingProcess36   69.17
ManufacturingProcess09   60.62
BiologicalMaterial02     60.16
BiologicalMaterial12     53.67
ManufacturingProcess31   51.71
ManufacturingProcess33   51.30
```

```
ManufacturingProcess29   48.47
ManufacturingProcess06   46.76
BiologicalMaterial04     42.96
BiologicalMaterial11     40.95
ManufacturingProcess02   34.47
BiologicalMaterial01     33.22
BiologicalMaterial09     33.19
BiologicalMaterial08     32.55
ManufacturingProcess11   32.49
```

```
plot(svmRTuned_75)
```




```
postResample(pred = knnPred_75, obs = testData$Yield)
```

```
     RMSE  Rsquared       MAE
1.1382855 0.6303354 0.9761111
```

```
postResample(pred = marsPred_75, obs = testData$Yield)
```

```
     RMSE  Rsquared       MAE
1.0748296 0.6300397 0.9186156
```

```
postResample(pred = nnetPred_75, obs = testData$Yield)
```

```
     RMSE  Rsquared       MAE
2.1814738 0.2613588 1.6469824
```

```
postResample(pred = svmPred_75, obs = testData$Yield)
```

```
     RMSE  Rsquared       MAE
0.9868486 0.6751334 0.7751097
```

The most effecitive model with this data is the Support Vector Machine Regression model. with an RMSE of .98 and an R-Squared of around .675. Although the R-Squared is not supremely impressive with the Biolgoical and Process variables explaining approximately 67% of the `Yeild`, that rSqured is 4% higher than the KNN, and MARS model, and about 41% heigher than the Neural Net model.

(b) Which predictors are most important in the optimal nonlinear regression model?

```r
varImp(svmRTuned_75)
```

```
loess r-squared variable importance

  only 20 most important variables shown (out of 57)

                       Overall
ManufacturingProcess32  100.00
ManufacturingProcess13   80.04
BiologicalMaterial06     79.67
ManufacturingProcess17   78.78
BiologicalMaterial03     70.18
ManufacturingProcess36   69.17
ManufacturingProcess09   60.62
BiologicalMaterial02     60.16
BiologicalMaterial12     53.67
ManufacturingProcess31   51.71
ManufacturingProcess33   51.30
ManufacturingProcess29   48.47
ManufacturingProcess06   46.76
BiologicalMaterial04     42.96
BiologicalMaterial11     40.95
ManufacturingProcess02   34.47
BiologicalMaterial01     33.22
BiologicalMaterial09     33.19
BiologicalMaterial08     32.55
ManufacturingProcess11   32.49
```

The Five Most Predictive Variables for this model were

ManufacturingProcess32
ManufacturingProcess13
BiologicalMaterial06
ManufacturingProcess17
BiologicalMaterial03

(b)Do either the biological or process variables dominate the list?

`ManufacturingProcess` Variables make ups six of the top ten variables are and three of the top four. Although they are only slightly more represented in the top 10, they definitely have greater overall contributions to this SVM model.

(b)How do the top ten important predictors compare to the top ten predictors from the optimal linear model?

Partial Least Squares Model

```
     RMSE  Rsquared       MAE
1.2918866 0.4853202 1.0457651
```

Variable Least Squares Importance

```
                             Overall
BiologicalMaterial01   0.060172069
BiologicalMaterial02   0.071017293
BiologicalMaterial03   0.068170221
BiologicalMaterial04   0.058091647
BiologicalMaterial05   0.033815243
BiologicalMaterial06   0.068069105
BiologicalMaterial07   0.038148945
BiologicalMaterial08   0.067984593
BiologicalMaterial09   0.029204647
BiologicalMaterial10   0.049496288
BiologicalMaterial11   0.063190066
BiologicalMaterial12   0.063448475
ManufacturingProcess01 0.024209688
ManufacturingProcess02 0.044220983
ManufacturingProcess03 0.009280345
ManufacturingProcess04 0.055220345
ManufacturingProcess05 0.032567859
ManufacturingProcess06 0.063124077
ManufacturingProcess07 0.011020534
ManufacturingProcess08 0.007480644
ManufacturingProcess09 0.096983144
ManufacturingProcess10 0.027139654
ManufacturingProcess11 0.058444770
ManufacturingProcess12 0.068764169
ManufacturingProcess13 0.097992018
ManufacturingProcess14 0.023134706
ManufacturingProcess15 0.045515801
ManufacturingProcess16 0.008751461
ManufacturingProcess17 0.097242427
ManufacturingProcess18 0.018461094
ManufacturingProcess19 0.039873453
ManufacturingProcess20 0.018291082
ManufacturingProcess21 0.032613128
ManufacturingProcess22 0.010859770
ManufacturingProcess23 0.016405204
ManufacturingProcess24 0.030550294
ManufacturingProcess25 0.010455066
ManufacturingProcess26 0.013000812
ManufacturingProcess27 0.013625637
ManufacturingProcess28 0.058006953
ManufacturingProcess29 0.031313341
ManufacturingProcess30 0.035501659
ManufacturingProcess31 0.019554533
ManufacturingProcess32 0.129672096
ManufacturingProcess33 0.081265980
ManufacturingProcess34 0.049590514
ManufacturingProcess35 0.028618269
ManufacturingProcess36 0.096528526
ManufacturingProcess37 0.041723487
ManufacturingProcess38 0.015949136
ManufacturingProcess39 0.021590742
ManufacturingProcess40 0.024385825
ManufacturingProcess41 0.027051444
```

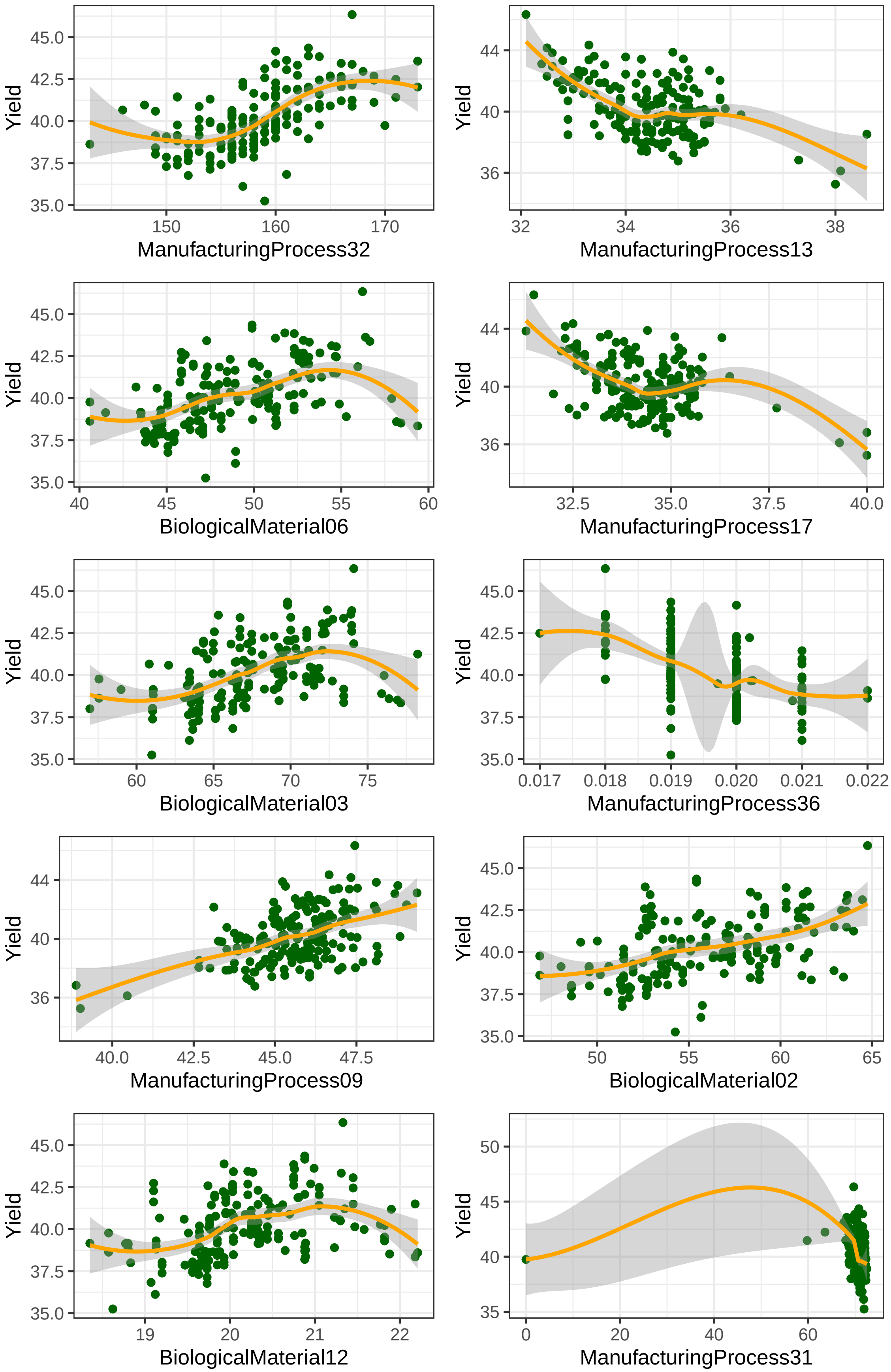
```
ManufacturingProcess42 0.015037021
ManufacturingProcess43 0.030021698
ManufacturingProcess44 0.026596036
ManufacturingProcess45 0.020119319
```

For the 41-variable partial least squares model the 10 most important variables were all `BiologicalMaterials`, where as in this more predictive SVM model, the top three are `ManufacturingProcess` variables and the blend of variables are more complex than in the partial least squares model.

As you can see from the importance values in the lists, the most imporant PLS variables are less important than the least important from the non-linear regression models in chapter seven.

(c) Explore the relationships between the top predictors and the response for the predictors that are unique to the optimal nonlinear regression model. Do these plots reveal intuition about the biological or process predictors and their relationship with yield?

```r
variables <-c( 'Yield',
'ManufacturingProcess32',
'ManufacturingProcess13' ,
'BiologicalMaterial06' ,
'ManufacturingProcess17',
'BiologicalMaterial03' ,
'ManufacturingProcess36',
'ManufacturingProcess09',
'BiologicalMaterial02',
'BiologicalMaterial12',
'ManufacturingProcess31' )

importants<-varImp(svmRTuned_75)
important <- importants$importance %>%
  rownames_to_column() %>%
  arrange(desc(Overall)) %>%
  head(10)

cols_importants <- imputed_data[,important$rowname]
cols_importants$Yield <- imputed_data$Yield
library(gridExtra)

plot_chems <- function(col){
  ggplot(cols_importants) +
  geom_point(aes_string(x = col, y='Yield'), color = 'darkgreen') +
    geom_smooth(aes_string(x = col, y = "Yield"), color = 'orange')+
  theme_bw()
}


plots <-lapply(colnames(cols_importants)[1:length(cols_importants) - 1], plot_chems)


grid.arrange(grobs = plots, ncol = 2, nrow = 6)
```


Yield
ManufacturingProcess32
Yield
ManufacturingProcess13
Yield
BiologicalMaterial06
Yield
ManufacturingProcess17
Yield
BiologicalMaterial03
Yield
ManufacturingProcess36
Yield
ManufacturingProcess09
Yield
BiologicalMaterial02
Yield
BiologicalMaterial12
Yield
ManufacturingProcess31

In looking at the top 12 most influential variables of the non-linear models, there are some pretty clear differences in the data which might explain both the overall poor performance of the linear models as well as the improved significance of Process-Based variables in the non-linear models.

Of the `ManfuacturingProcess` variables, only 32 & 09 were even remotely linear, and 09 could arguably b considered a cluster with a few outliers that leverage it to seem linear. the rest are either tight clusters or discrete values which predict an array of possible Yields, which is directly opposed the the definition of linearly separable data.

On the other hand, the biological variable are more continuous and some approximate linear distributions when plotted against the `Yield`. However, when you look closely at these plots, you can see that the smoothed line for each of them is curved, often sinusoidal or semi-sinusoidal. This likely explains the weakness of even this variables in the importance analysis, as only `BiolgocialMaterial02` (from the non-linear model) shows a full linear relationship and it is the second most predictive variable in linear model. I would assume that the most important `BiologicalMaterial01` would also be generally linear to `Yield`.