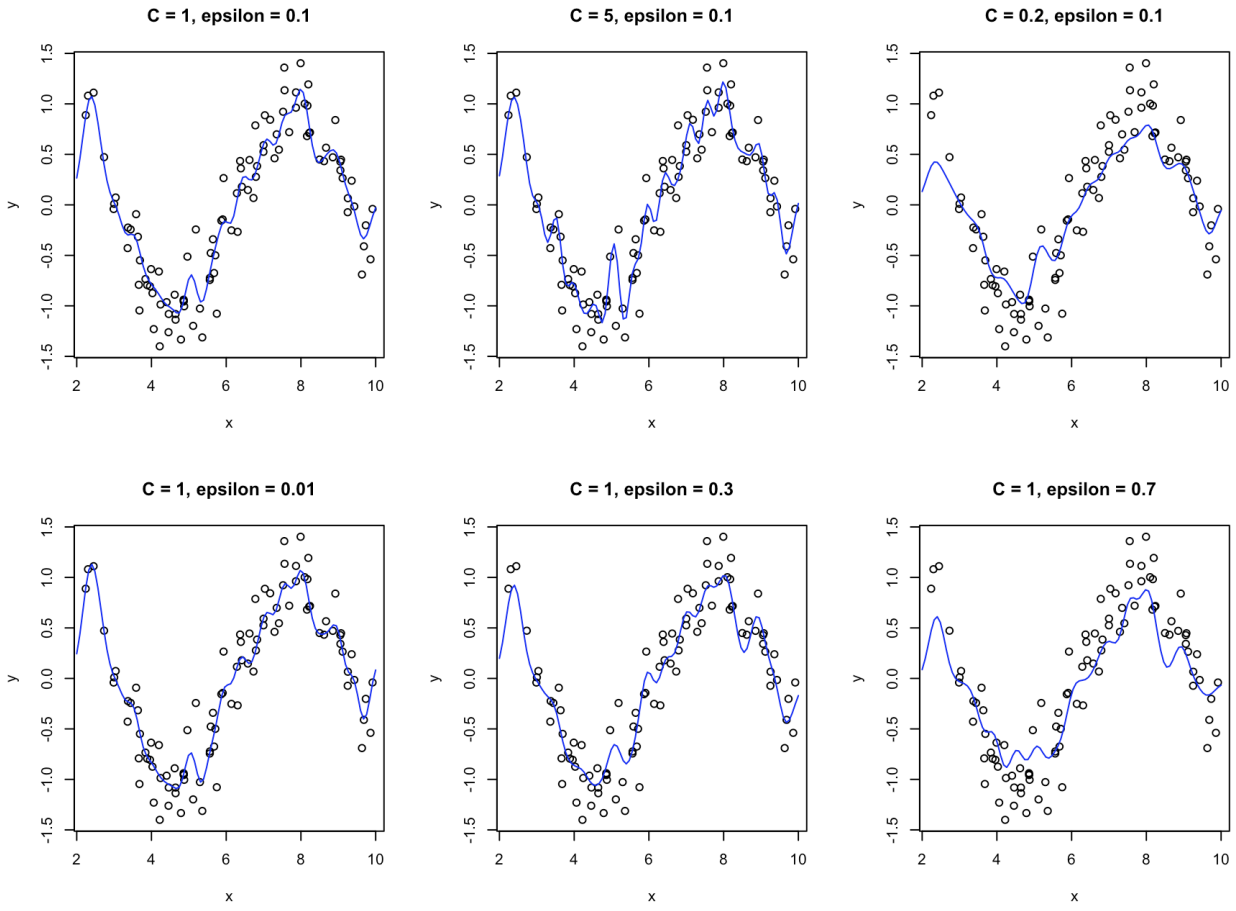
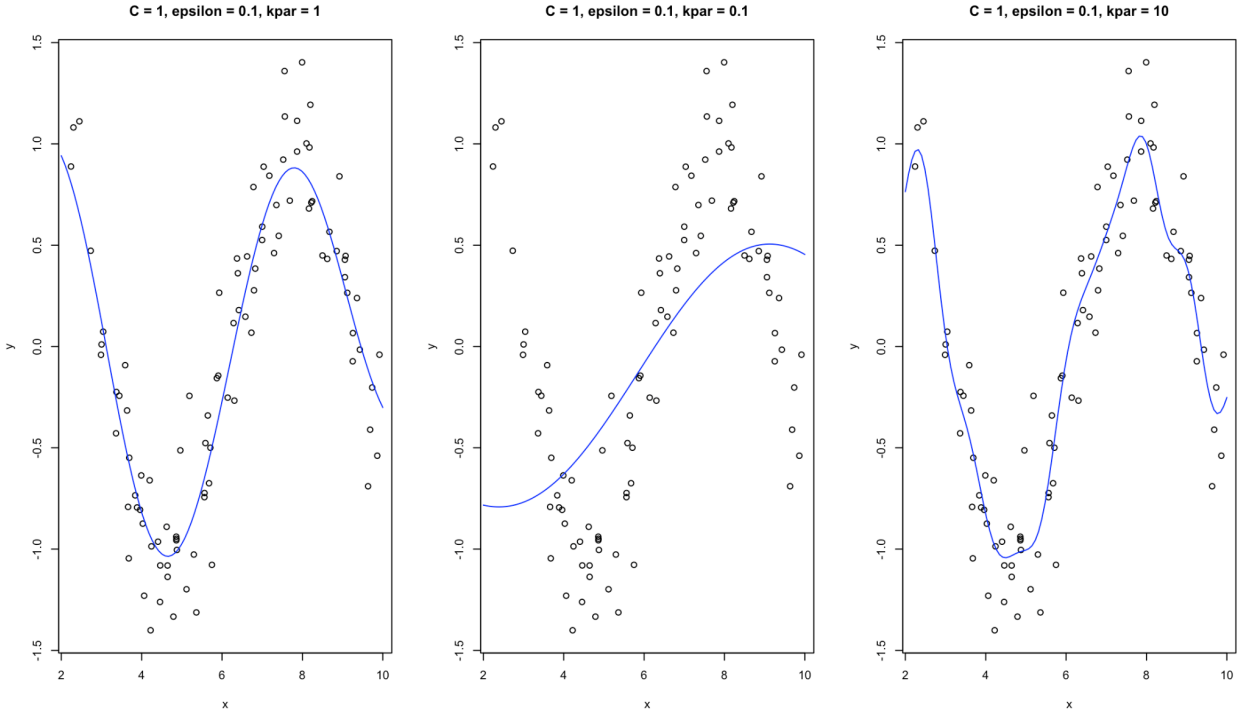
Week 5 Homework

Problem 7.1a Fit different models using a radial basis function and different values of the cost (the C parameter) and ε. Plot the fitted curve.



Problem 7.1b The σ parameter can be adjusted using the kpar argument, such as kpar = list(sigma = 1). Try different values of σ to understand how this parameter changes the model fit. How do the cost, ε, and σ values affect the model?



- It is shown that the cost parameter is affecting how stiff or flexible the model is. As the cost is increased, the fitted line tends to be more noisy thus resulting in a more overly fit model.
- It is shown that the epsilon parameter is also affecting how stiff or flexible the model fit is, but to a less drastic degree than the cost parameter. As epsilon is increased, the fitted line tends to be less noisy thus resulting in a more under fit model.
- It is shown that the sigma parameters is also affecting the flexibility of the model fit, but to a less drastic degree than both the cost and sigma parameters. As sigma is increased, the fitted line tends to be more noisy thus resulting in a more overly fit model.

Problem 7.2a Consider KNN and MARS, which model appears to give the best performance?

- KNN Model

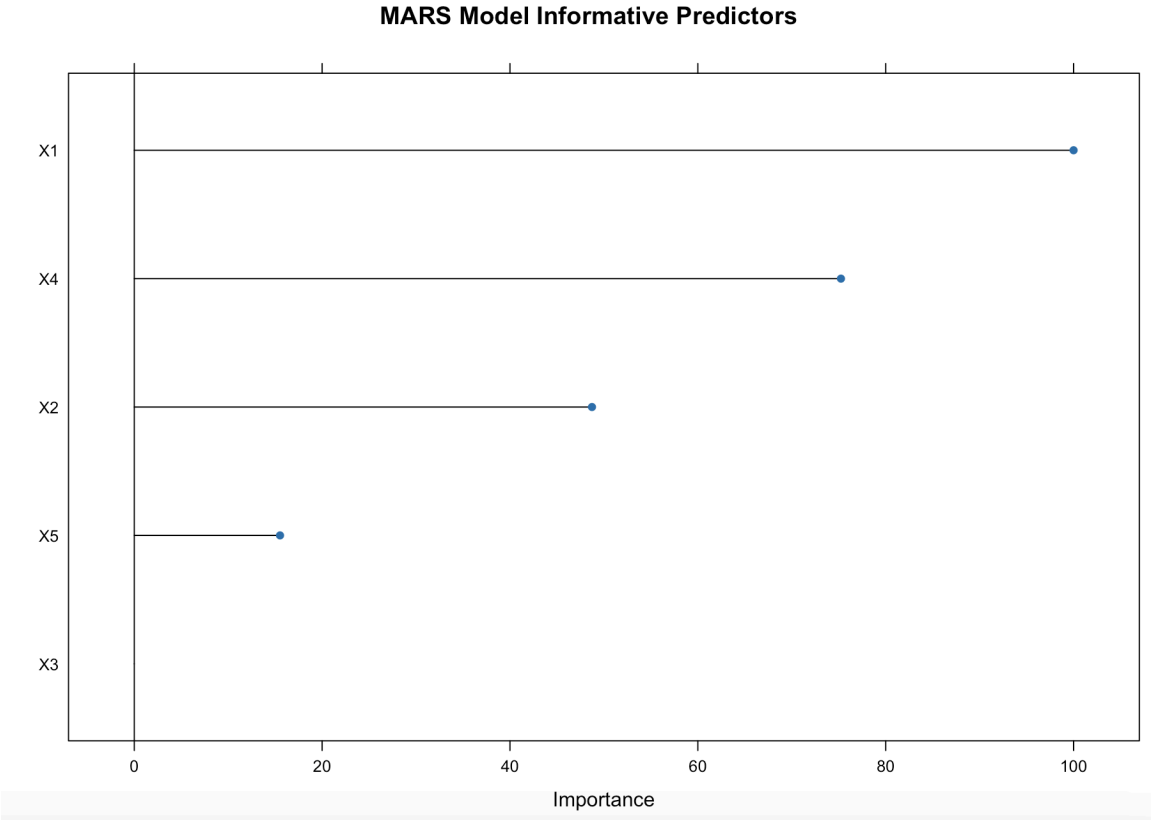
```
k-Nearest Neighbors

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  3.569425  0.4967799  2.910695
   7  3.456938  0.5295171  2.810768
   9  3.360125  0.5683839  2.734616
  11  3.305196  0.5962254  2.687646
  13  3.294275  0.6130595  2.670403
  15  3.275696  0.6306881  2.648741
  17  3.273680  0.6423614  2.650513
  19  3.275232  0.6543585  2.641627
  21  3.278413  0.6633100  2.643754
  23  3.301181  0.6662048  2.667520

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

- MARS Model

```
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
  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

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.
```

- The MARS model performed better in the cross validation resampling approach. It showed a significantly lower RMSE and a higher R squared for the optimal model deduced from training than the KNN model.

Problem 7.2b Does MARS select the informative predictors (those named X1-X5)?

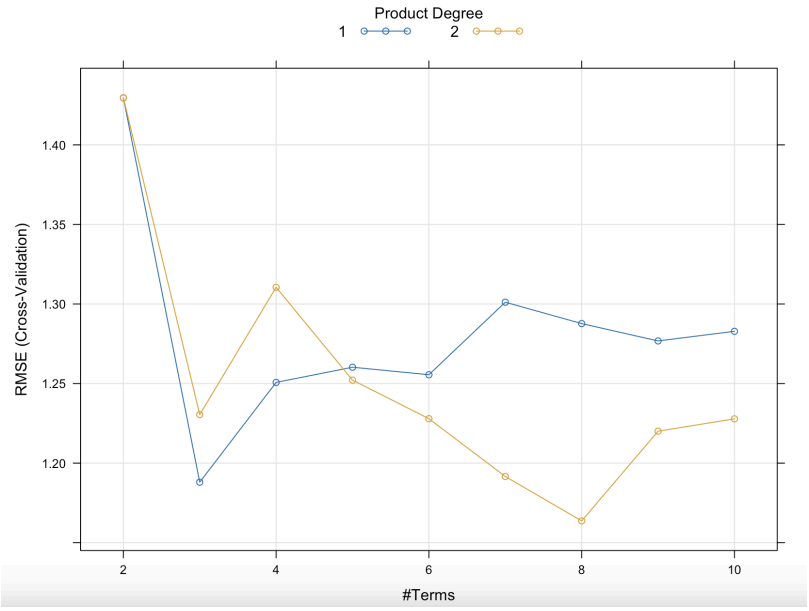


- The Mars model selected predictors X1-X5 as the most important predictors, however the predictor X3 is shown not to contribute very much.

7.5a Which nonlinear regression model gives the optimal resampling and test set performance? (Please put the table of the summary statistics for each tuning parameter and the figure for the tuning parameter, then use a table to summarize the RMSE and the best tuning parameter values for all models)

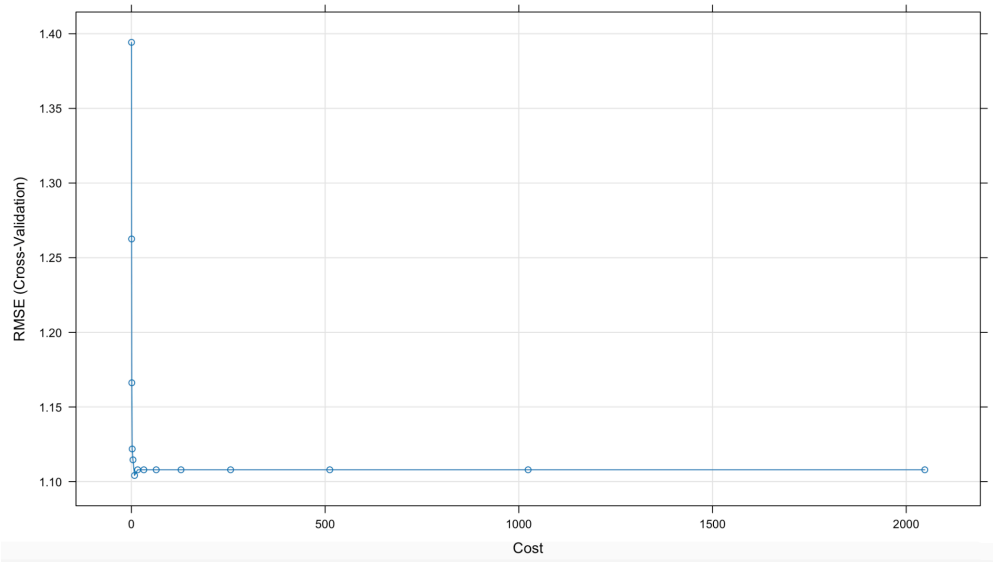
- MARS Model:

```
Multivariate Adaptive Regression Spline

144 samples
 57 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 129, 130, 130, 130, 130, 130, ...
Resampling results across tuning parameters:

  degree  nprune  RMSE      Rsquared   MAE
  1        2      1.429556  0.4139077  1.1186889
  1        3      1.188024  0.5909543  0.9708303
  1        4      1.250731  0.5575319  0.9758262
  1        5      1.260261  0.5529275  1.0045965
  1        6      1.255542  0.5592966  1.0092761
  1        7      1.301125  0.5377737  1.0311511
  1        8      1.287694  0.5392004  1.0236782
  1        9      1.276788  0.5548110  0.9964154
  1       10      1.282836  0.5500369  1.0142896
  2        2      1.429556  0.4139077  1.1186889
  2        3      1.230477  0.5740757  0.9932185
  2        4      1.310504  0.5420260  1.0397017
  2        5      1.252175  0.5619478  0.9814648
  2        6      1.227914  0.5775375  0.9789457
  2        7      1.191610  0.6083045  0.9519472
  2        8      1.163684  0.6275667  0.9309502
  2        9      1.220070  0.6057522  0.9880062
  2       10      1.227821  0.6107432  0.9611237

RMSE was used to select the optimal model using the smallest value.
The final values used for the model were nprune = 8 and degree = 2.
```



- Radial Support Vector Machine:

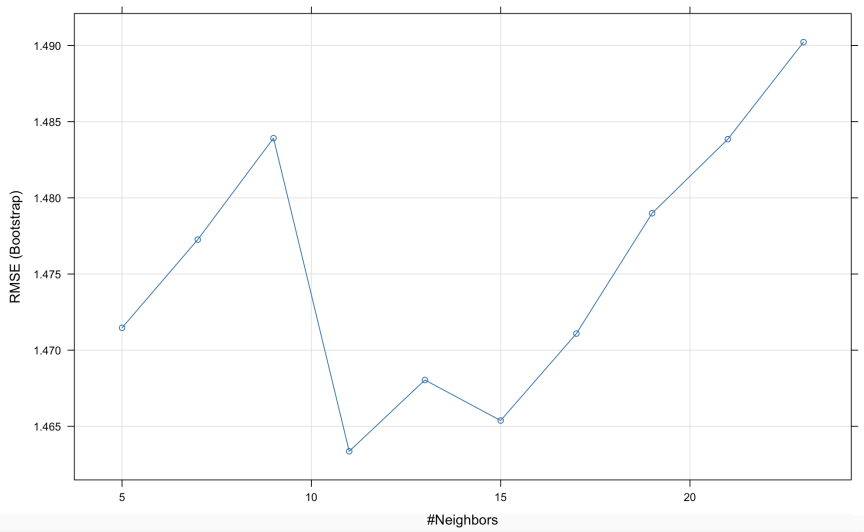
```
Support Vector Machines with Radial Basis Function Kernel

144 samples
 57 predictor

Pre-processing: centered (57), scaled (57)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 129, 130, 130, 130, 130, 130, ...
Resampling results across tuning parameters:

  C        RMSE      Rsquared   MAE
     0.25  1.394262  0.5141933  1.1455893
     0.50  1.262544  0.5659327  1.0361630
     1.00  1.166208  0.6082486  0.9612703
     2.00  1.121870  0.6340404  0.9097590
     4.00  1.114613  0.6382909  0.8909902
     8.00  1.104133  0.6487791  0.8835416
    16.00  1.107931  0.6484868  0.8885898
    32.00  1.107931  0.6484868  0.8885898
    64.00  1.107931  0.6484868  0.8885898
   128.00  1.107931  0.6484868  0.8885898
   256.00  1.107931  0.6484868  0.8885898
   512.00  1.107931  0.6484868  0.8885898
  1024.00  1.107931  0.6484868  0.8885898
  2048.00  1.107931  0.6484868  0.8885898

Tuning parameter 'sigma' was held constant at a value of 0.01407434
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were sigma = 0.01407434 and C = 8.
```



-

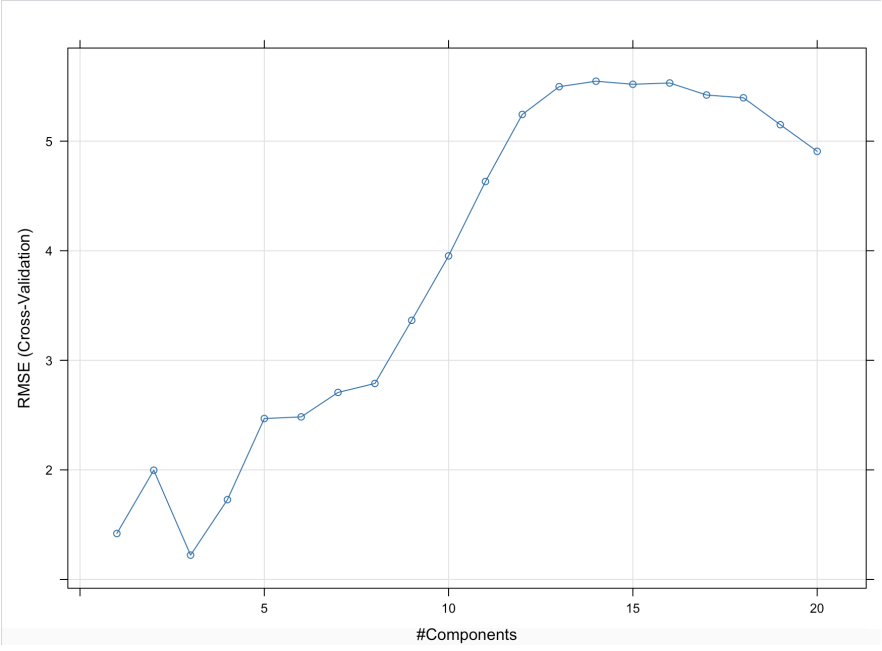
- K Nearest Neighbors (KNN):

```
k-Nearest Neighbors

144 samples
 57 predictor

Pre-processing: centered (57), scaled (57)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
Resampling results across tuning parameters:

  k   RMSE      Rsquared   MAE
   5  1.471468  0.3778904  1.179587
   7  1.477259  0.3724717  1.198329
   9  1.483918  0.3661788  1.212592
  11  1.463363  0.3820165  1.200061
  13  1.468042  0.3843021  1.206155
  15  1.465381  0.3908357  1.201536
  17  1.471089  0.3896014  1.206109
  19  1.478989  0.3863659  1.211427
  21  1.483851  0.3869820  1.215335
  23  1.490223  0.3821859  1.220561

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



- Partial Least Squares Regression Model

```
Partial Least Squares

144 samples
 57 predictor

Pre-processing: centered (57), scaled (57)
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 115, 116, 115, 116, 114
Resampling results across tuning parameters:

  ncomp  RMSE      Rsquared   MAE
   1     1.419432  0.4269294  1.1495254
   2     1.996798  0.3749835  1.2036994
   3     1.222357  0.5685897  0.9979326
   4     1.728261  0.4526376  1.1593988
   5     2.468671  0.3798426  1.3255159
   6     2.483699  0.3917499  1.3153000
   7     2.706696  0.3826221  1.3670381
   8     2.788652  0.3882168  1.3689829
   9     3.364899  0.3338295  1.4932421
  10     3.953010  0.2935103  1.6185159
  11     4.631643  0.2734378  1.7593869
  12     5.243850  0.2635558  1.8873704
  13     5.497431  0.2566988  1.9375334
  14     5.547109  0.2548997  1.9522387
  15     5.519178  0.2549265  1.9417842
  16     5.530887  0.2546334  1.9405085
  17     5.421166  0.2515745  1.9290337
  18     5.395905  0.2502569  1.9435262
  19     5.150379  0.2499816  1.9162764
  20     4.907347  0.2505897  1.8884910

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was ncomp = 3.
```



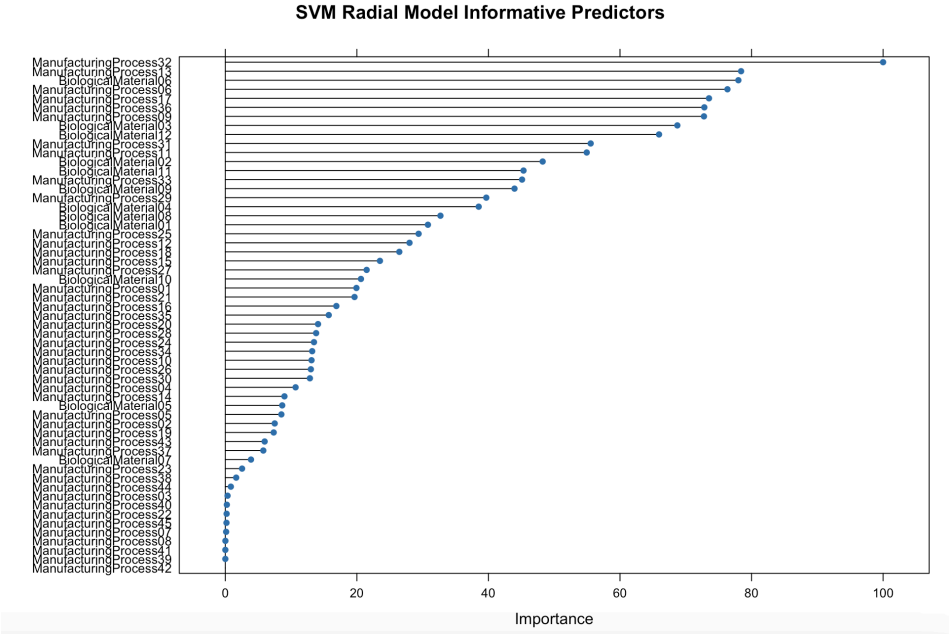
- Table Summary of all Models tested:

Model	Tuning Parameter	RMSE
MARS	Degree = 2, Terms = 8	1.163694
Radial Support Vector Machine	Cost = 8	1.104133
K Nearest Neighbor	Neighbors = 11	1.463363
Partial Least Squares (Linear)	Components = 3	1.222357
Multiple Linear Regression (Linear)	N/A	3.282545
Lasso (Linear)	Lambda = 0.06210526	1.144857

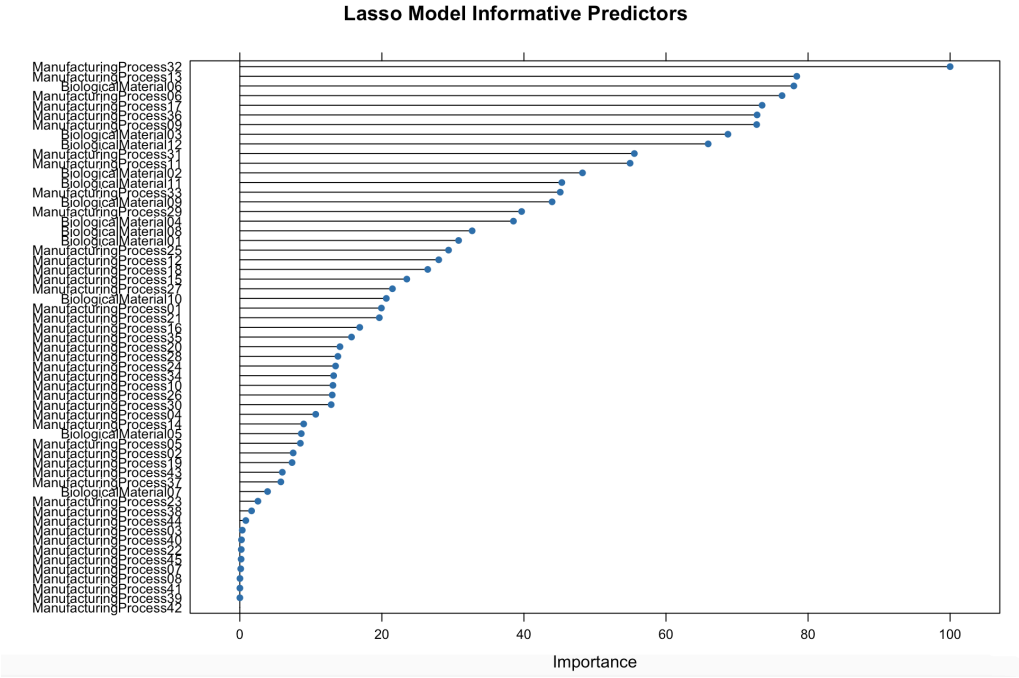
- Out of the Nonlinear model approaches tested here, the Radial Support Vector Machine performed the best with respect to RMSE. It had the Lowest RMSE of all of the models tested.
- Out of the Linear model approaches tested here, the Lasso model performed the best with respect to RMSE. It had the Lowest RMSE of all of the models tested.

7.5b Which predictors are most important in the optimal nonlinear regression model?

- Optimal Nonlinear Regression Model (Radial SVM)



- Optimal Linear Regression Model (Lasso Model)



7.5c Do either the biological or process variables dominate the list? How do the top ten important predictors compare to the top ten predictors from the optimal linear model?

- The majority of the top 10 predictors are Process variables rather than Biological variables for the optimal nonlinear model, although there are some important Biological predictors such as BiologicalMaterial06. The optimal linear model was the Lasso model. Its predictor importance ranking looks identical to the Radial Support Vector Machine model.

R Code

```
#Problem 7.1 Simulate a single predictor and a nonlinear relationship, such as a sin wave shown in Fig. 7.7, and
#investigate the relationship between the cost, ε, and kernel parameters for a support vector machine model
par(mfrow = c(1,3))
set.seed(100)
x <- runif(100, min = 2, max = 10)
y <- sin(x) + rnorm(length(x)) * .25
sinData <- data.frame(x = x, y = y)
plot(x, y, main = 'C = 1, epsilon = 0.1, kpar = 10')
## Create a grid of x values to use for prediction
dataGrid <- data.frame(x = seq(2, 10, length = 100))

# 7.1a Fit different models using a radial basis function and different values of the cost (the C parameter) and
# ε. Plot the fitted curve. For example
library(kernlab)
rbfSVM <- ksvm(x = x, y = y, data = sinData, kernel ="rbfdot", kpar = list(sigma = 10), C = 1, epsilon = 0.1)
modelPrediction <- predict(rbfSVM, newdata = dataGrid)
## This is a matrix with one column. We can plot the
## model predictions by adding points to the previous plot
points(x = dataGrid$x, y = modelPrediction[,1], type = "l", col = "blue")

#7.2 Friedman (1991) introduced several benchmark data sets create by simulation.
# One of these simulations used the following nonlinear equation to create data
# 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
library(mlbench)
library(caret)
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)
## 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)

#Tune several models on these data. For example:
knnModel <- train(x = trainingData$x,y = trainingData$y,method = "knn",preProc = c("center",
"scale"),tuneLength = 10)
knnModel
knnPred <- predict(knnModel, newdata = testData$x)
## The function 'postResample' can be used to get the test set
## performance values
postResample(pred = knnPred, obs = testData$y)

#7.2a Consider KNN and MARS, which model appears to give the best performance?
marsGrid <- expand.grid(.degree = 1:2, .nprune = 2:50)
# Fix the seed so that the results can be reproduced
set.seed(100)
marsTuned <- train(trainingData$x, trainingData$y, method = "earth",tuneGrid = marsGrid,
trControl = trainControl(method = "cv"))
marsTuned
marsPred <- predict(marsTuned, testData$x)
postResample(pred = marsPred, obs = testData$y)

#7.2b Does MARS select the informative predictors (those named X1-X5)?
plot(varImp(marsTuned), main = 'MARS Model Informative Predictors')

# 7.5 Exercise 6.3 describes data for a chemical manufacturing process. Use the same data
imputation, data
# splitting, and pre-processing steps as before and train several nonlinear regression models.
library(AppliedPredictiveModeling)
library(VIM)
data(ChemicalManufacturingProcess)
chem <- kNN(ChemicalManufacturingProcess, imp_var = FALSE)
trainRows <- createDataPartition(chem$Yield, p = .80, list = FALSE)
chem.predictors <- chem[,-1]
train.pred <- chem.predictors[trainRows,]
test.pred <- chem.predictors[-trainRows,]
Yield <- chem[,1]
train.resp <- Yield[trainRows]
test.resp <- Yield[-trainRows]

# 7.5a Which nonlinear regression model gives the optimal resampling and test set
#performance? (Please put the table of the summary statistics for each tuning parameter and
#the figure for the tuning parameter, then use a table to summarize the RMSE and the best
#tuning parameter values for all models)
```

```
### MARS Model
marsGrid <- expand.grid(.degree = 1:2, .nprune = 2:10)
set.seed(100)
marsTuned <- train(train.pred, train.resp, method = "earth",tuneGrid = marsGrid, trControl =
trainControl(method = "cv"))
marsTuned
plot(marsTuned)
marsPred <- predict(marsTuned, test.pred)
postResample(pred = marsPred, obs = test.resp)

### KNN Model
set.seed(100)
knnModel <- train(x = train.pred,y = train.resp,method = "knn",preProc = c("center",
"scale"),tuneLength = 10)
knnModel
plot(knnModel)
knnPred <- predict(knnModel, newdata = test.pred)
postResample(pred = knnPred, obs = test.resp)

### Radial Support Vector Machine
set.seed(100)
svmRTuned <- train(train.pred, train.resp, method = "svmRadial", preProc = c("center", "scale"),
tuneLength = 14,trControl = trainControl(method = "cv"))
svmRTuned
plot(svmRTuned)
svmRPred <- predict(svmRTuned, newdata = test.pred)
postResample(pred = svmRPred, obs = test.resp)

### Partial Least Squares Regression Model
set.seed(100)
plsTune <- train(train.pred, train.resp, method = "pls", tuneLength = 20, trControl = ctrl, preProc
= c("center", "scale"))
plsTune
plot(plsTune)
plsPred <- predict(plsTune, newdata = test.pred)
postResample(pred = plsPred, obs = test.resp)

### Multiple Linear Regression
ctrl <- trainControl(method = "cv", number = 5)
set.seed(100)
lmRegFit <- train(train.pred, train.resp, method = "lm", trControl = ctrl, preProc = c("center",
"scale"))
lmRegFit
```

```
# Lasso Penalized Model
lassoGrid=expand.grid(.fraction=seq(0.01,1,length = 20))
set.seed(100)
lassoTune=train(train.pred,train.resp,method="lasso",tuneGrid=lassoGrid,trControl=ctrl)
lassoTune
plot(lassoTune)

# 7.5b Which predictors are most important in the optimal nonlinear regression model?
plot(varImp(svmRTuned), main = 'SVM Radial Model Informative Predictors')

# 7.5c Do either the biological or process variables dominate the list? How do the top ten
#important predictors compare to the top ten predictors from the optimal linear model?
plot(varImp(lassoTune), main = 'Lasso Model Informative Predictors')
```