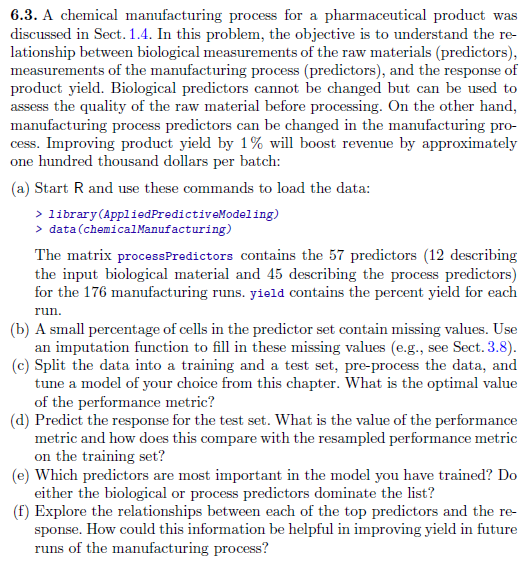
DATA 624 Spring 2019: Homework-2

Ahmed Sajjad, Harpreet Shoker, Jagruti Solao, Chad Smith, Todd Weigel

library(dplyr)  
library(ggplot2)  
library(pander)  
library(tibble)  
library(purrr)  
library(tidyr)  
library(stringr)  
library(readr)  
library(forcats)  
library(earth)  
library(Formula)  
library(plotmo)  
library(plotrix)  
library(TeachingDemos)  
library(rpart)  
library("rpart.plot")  
library(mlbench)  
library(caret)  
library(mice)  
library(corrplot)  
library(olsrr)  
library(nnet)  
library(kernlab)  
library(rminer)  
library(AppliedPredictiveModeling)  
library(DMwR)  
library(party)  
library(ipred)  
library(randomForest)  
library(gbm)  
library(Cubist)  
library(party)



#### a) Start R and use the following commands to load the data. The matrix ‘processPredictors’ contains 57 predictors (12 describing the input biological material and 45 describing the process predictors) for the 176 manufacturing runs. ‘yield’ contains the percent yield for each run.

library(AppliedPredictiveModeling)  
data(ChemicalManufacturingProcess)  
processPredictors <- ChemicalManufacturingProcess

#### b) A small percentage of cells in the redictor set contain missing values. Use an imputation function to fill in these missing values

The summary for this dataset shows that many columns have 5 or less missing values. The column ‘ManufacturingProcess03’ had the most missing values with 15.

summary(processPredictors)

## Yield BiologicalMaterial01 BiologicalMaterial02  
## Min. :35.25 Min. :4.580 Min. :46.87   
## 1st Qu.:38.75 1st Qu.:5.978 1st Qu.:52.68   
## Median :39.97 Median :6.305 Median :55.09   
## Mean :40.18 Mean :6.411 Mean :55.69   
## 3rd Qu.:41.48 3rd Qu.:6.870 3rd Qu.:58.74   
## Max. :46.34 Max. :8.810 Max. :64.75   
##   
## BiologicalMaterial03 BiologicalMaterial04 BiologicalMaterial05  
## Min. :56.97 Min. : 9.38 Min. :13.24   
## 1st Qu.:64.98 1st Qu.:11.24 1st Qu.:17.23   
## Median :67.22 Median :12.10 Median :18.49   
## Mean :67.70 Mean :12.35 Mean :18.60   
## 3rd Qu.:70.43 3rd Qu.:13.22 3rd Qu.:19.90   
## Max. :78.25 Max. :23.09 Max. :24.85   
##   
## BiologicalMaterial06 BiologicalMaterial07 BiologicalMaterial08  
## Min. :40.60 Min. :100.0 Min. :15.88   
## 1st Qu.:46.05 1st Qu.:100.0 1st Qu.:17.06   
## Median :48.46 Median :100.0 Median :17.51   
## Mean :48.91 Mean :100.0 Mean :17.49   
## 3rd Qu.:51.34 3rd Qu.:100.0 3rd Qu.:17.88   
## Max. :59.38 Max. :100.8 Max. :19.14   
##   
## BiologicalMaterial09 BiologicalMaterial10 BiologicalMaterial11  
## Min. :11.44 Min. :1.770 Min. :135.8   
## 1st Qu.:12.60 1st Qu.:2.460 1st Qu.:143.8   
## Median :12.84 Median :2.710 Median :146.1   
## Mean :12.85 Mean :2.801 Mean :147.0   
## 3rd Qu.:13.13 3rd Qu.:2.990 3rd Qu.:149.6   
## Max. :14.08 Max. :6.870 Max. :158.7   
##   
## BiologicalMaterial12 ManufacturingProcess01 ManufacturingProcess02  
## Min. :18.35 Min. : 0.00 Min. : 0.00   
## 1st Qu.:19.73 1st Qu.:10.80 1st Qu.:19.30   
## Median :20.12 Median :11.40 Median :21.00   
## Mean :20.20 Mean :11.21 Mean :16.68   
## 3rd Qu.:20.75 3rd Qu.:12.15 3rd Qu.:21.50   
## Max. :22.21 Max. :14.10 Max. :22.50   
## NA's :1 NA's :3   
## ManufacturingProcess03 ManufacturingProcess04 ManufacturingProcess05  
## Min. :1.47 Min. :911.0 Min. : 923.0   
## 1st Qu.:1.53 1st Qu.:928.0 1st Qu.: 986.8   
## Median :1.54 Median :934.0 Median : 999.2   
## Mean :1.54 Mean :931.9 Mean :1001.7   
## 3rd Qu.:1.55 3rd Qu.:936.0 3rd Qu.:1008.9   
## Max. :1.60 Max. :946.0 Max. :1175.3   
## NA's :15 NA's :1 NA's :1   
## ManufacturingProcess06 ManufacturingProcess07 ManufacturingProcess08  
## Min. :203.0 Min. :177.0 Min. :177.0   
## 1st Qu.:205.7 1st Qu.:177.0 1st Qu.:177.0   
## Median :206.8 Median :177.0 Median :178.0   
## Mean :207.4 Mean :177.5 Mean :177.6   
## 3rd Qu.:208.7 3rd Qu.:178.0 3rd Qu.:178.0   
## Max. :227.4 Max. :178.0 Max. :178.0   
## NA's :2 NA's :1 NA's :1   
## ManufacturingProcess09 ManufacturingProcess10 ManufacturingProcess11  
## Min. :38.89 Min. : 7.500 Min. : 7.500   
## 1st Qu.:44.89 1st Qu.: 8.700 1st Qu.: 9.000   
## Median :45.73 Median : 9.100 Median : 9.400   
## Mean :45.66 Mean : 9.179 Mean : 9.386   
## 3rd Qu.:46.52 3rd Qu.: 9.550 3rd Qu.: 9.900   
## Max. :49.36 Max. :11.600 Max. :11.500   
## NA's :9 NA's :10   
## ManufacturingProcess12 ManufacturingProcess13 ManufacturingProcess14  
## Min. : 0.0 Min. :32.10 Min. :4701   
## 1st Qu.: 0.0 1st Qu.:33.90 1st Qu.:4828   
## Median : 0.0 Median :34.60 Median :4856   
## Mean : 857.8 Mean :34.51 Mean :4854   
## 3rd Qu.: 0.0 3rd Qu.:35.20 3rd Qu.:4882   
## Max. :4549.0 Max. :38.60 Max. :5055   
## NA's :1 NA's :1   
## ManufacturingProcess15 ManufacturingProcess16 ManufacturingProcess17  
## Min. :5904 Min. : 0 Min. :31.30   
## 1st Qu.:6010 1st Qu.:4561 1st Qu.:33.50   
## Median :6032 Median :4588 Median :34.40   
## Mean :6039 Mean :4566 Mean :34.34   
## 3rd Qu.:6061 3rd Qu.:4619 3rd Qu.:35.10   
## Max. :6233 Max. :4852 Max. :40.00   
##   
## ManufacturingProcess18 ManufacturingProcess19 ManufacturingProcess20  
## Min. : 0 Min. :5890 Min. : 0   
## 1st Qu.:4813 1st Qu.:6001 1st Qu.:4553   
## Median :4835 Median :6022 Median :4582   
## Mean :4810 Mean :6028 Mean :4556   
## 3rd Qu.:4862 3rd Qu.:6050 3rd Qu.:4610   
## Max. :4971 Max. :6146 Max. :4759   
##   
## ManufacturingProcess21 ManufacturingProcess22 ManufacturingProcess23  
## Min. :-1.8000 Min. : 0.000 Min. :0.000   
## 1st Qu.:-0.6000 1st Qu.: 3.000 1st Qu.:2.000   
## Median :-0.3000 Median : 5.000 Median :3.000   
## Mean :-0.1642 Mean : 5.406 Mean :3.017   
## 3rd Qu.: 0.0000 3rd Qu.: 8.000 3rd Qu.:4.000   
## Max. : 3.6000 Max. :12.000 Max. :6.000   
## NA's :1 NA's :1   
## ManufacturingProcess24 ManufacturingProcess25 ManufacturingProcess26  
## Min. : 0.000 Min. : 0 Min. : 0   
## 1st Qu.: 4.000 1st Qu.:4832 1st Qu.:6020   
## Median : 8.000 Median :4855 Median :6047   
## Mean : 8.834 Mean :4828 Mean :6016   
## 3rd Qu.:14.000 3rd Qu.:4877 3rd Qu.:6070   
## Max. :23.000 Max. :4990 Max. :6161   
## NA's :1 NA's :5 NA's :5   
## ManufacturingProcess27 ManufacturingProcess28 ManufacturingProcess29  
## Min. : 0 Min. : 0.000 Min. : 0.00   
## 1st Qu.:4560 1st Qu.: 0.000 1st Qu.:19.70   
## Median :4587 Median :10.400 Median :19.90   
## Mean :4563 Mean : 6.592 Mean :20.01   
## 3rd Qu.:4609 3rd Qu.:10.750 3rd Qu.:20.40   
## Max. :4710 Max. :11.500 Max. :22.00   
## NA's :5 NA's :5 NA's :5   
## ManufacturingProcess30 ManufacturingProcess31 ManufacturingProcess32  
## Min. : 0.000 Min. : 0.00 Min. :143.0   
## 1st Qu.: 8.800 1st Qu.:70.10 1st Qu.:155.0   
## Median : 9.100 Median :70.80 Median :158.0   
## Mean : 9.161 Mean :70.18 Mean :158.5   
## 3rd Qu.: 9.700 3rd Qu.:71.40 3rd Qu.:162.0   
## Max. :11.200 Max. :72.50 Max. :173.0   
## NA's :5 NA's :5   
## ManufacturingProcess33 ManufacturingProcess34 ManufacturingProcess35  
## Min. :56.00 Min. :2.300 Min. :463.0   
## 1st Qu.:62.00 1st Qu.:2.500 1st Qu.:490.0   
## Median :64.00 Median :2.500 Median :495.0   
## Mean :63.54 Mean :2.494 Mean :495.6   
## 3rd Qu.:65.00 3rd Qu.:2.500 3rd Qu.:501.5   
## Max. :70.00 Max. :2.600 Max. :522.0   
## NA's :5 NA's :5 NA's :5   
## ManufacturingProcess36 ManufacturingProcess37 ManufacturingProcess38  
## Min. :0.01700 Min. :0.000 Min. :0.000   
## 1st Qu.:0.01900 1st Qu.:0.700 1st Qu.:2.000   
## Median :0.02000 Median :1.000 Median :3.000   
## Mean :0.01957 Mean :1.014 Mean :2.534   
## 3rd Qu.:0.02000 3rd Qu.:1.300 3rd Qu.:3.000   
## Max. :0.02200 Max. :2.300 Max. :3.000   
## NA's :5   
## ManufacturingProcess39 ManufacturingProcess40 ManufacturingProcess41  
## Min. :0.000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:7.100 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :7.200 Median :0.00000 Median :0.00000   
## Mean :6.851 Mean :0.01771 Mean :0.02371   
## 3rd Qu.:7.300 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :7.500 Max. :0.10000 Max. :0.20000   
## NA's :1 NA's :1   
## ManufacturingProcess42 ManufacturingProcess43 ManufacturingProcess44  
## Min. : 0.00 Min. : 0.0000 Min. :0.000   
## 1st Qu.:11.40 1st Qu.: 0.6000 1st Qu.:1.800   
## Median :11.60 Median : 0.8000 Median :1.900   
## Mean :11.21 Mean : 0.9119 Mean :1.805   
## 3rd Qu.:11.70 3rd Qu.: 1.0250 3rd Qu.:1.900   
## Max. :12.10 Max. :11.0000 Max. :2.100   
##   
## ManufacturingProcess45  
## Min. :0.000   
## 1st Qu.:2.100   
## Median :2.200   
## Mean :2.138   
## 3rd Qu.:2.300   
## Max. :2.600   
##

Using the MICE package, the ‘predictive mean matching’ method was used to impute missing values on the dataset.  
It ran through 50 iterations.

Summary of the completed dataset shows all missing values have been imputed.

summary(completeData)

## Yield BiologicalMaterial01 BiologicalMaterial02  
## Min. :35.25 Min. :4.580 Min. :46.87   
## 1st Qu.:38.75 1st Qu.:5.978 1st Qu.:52.68   
## Median :39.97 Median :6.305 Median :55.09   
## Mean :40.18 Mean :6.411 Mean :55.69   
## 3rd Qu.:41.48 3rd Qu.:6.870 3rd Qu.:58.74   
## Max. :46.34 Max. :8.810 Max. :64.75   
## BiologicalMaterial03 BiologicalMaterial04 BiologicalMaterial05  
## Min. :56.97 Min. : 9.38 Min. :13.24   
## 1st Qu.:64.98 1st Qu.:11.24 1st Qu.:17.23   
## Median :67.22 Median :12.10 Median :18.49   
## Mean :67.70 Mean :12.35 Mean :18.60   
## 3rd Qu.:70.43 3rd Qu.:13.22 3rd Qu.:19.90   
## Max. :78.25 Max. :23.09 Max. :24.85   
## BiologicalMaterial06 BiologicalMaterial07 BiologicalMaterial08  
## Min. :40.60 Min. :100.0 Min. :15.88   
## 1st Qu.:46.05 1st Qu.:100.0 1st Qu.:17.06   
## Median :48.46 Median :100.0 Median :17.51   
## Mean :48.91 Mean :100.0 Mean :17.49   
## 3rd Qu.:51.34 3rd Qu.:100.0 3rd Qu.:17.88   
## Max. :59.38 Max. :100.8 Max. :19.14   
## BiologicalMaterial09 BiologicalMaterial10 BiologicalMaterial11  
## Min. :11.44 Min. :1.770 Min. :135.8   
## 1st Qu.:12.60 1st Qu.:2.460 1st Qu.:143.8   
## Median :12.84 Median :2.710 Median :146.1   
## Mean :12.85 Mean :2.801 Mean :147.0   
## 3rd Qu.:13.13 3rd Qu.:2.990 3rd Qu.:149.6   
## Max. :14.08 Max. :6.870 Max. :158.7   
## BiologicalMaterial12 ManufacturingProcess01 ManufacturingProcess02  
## Min. :18.35 Min. : 0.00 Min. : 0.00   
## 1st Qu.:19.73 1st Qu.:10.80 1st Qu.:19.23   
## Median :20.12 Median :11.40 Median :21.00   
## Mean :20.20 Mean :11.21 Mean :16.64   
## 3rd Qu.:20.75 3rd Qu.:12.20 3rd Qu.:21.50   
## Max. :22.21 Max. :14.10 Max. :22.50   
## ManufacturingProcess03 ManufacturingProcess04 ManufacturingProcess05  
## Min. :1.47 Min. :911.0 Min. : 923.0   
## 1st Qu.:1.53 1st Qu.:927.8 1st Qu.: 986.8   
## Median :1.54 Median :934.0 Median : 999.4   
## Mean :1.54 Mean :931.8 Mean :1002.0   
## 3rd Qu.:1.55 3rd Qu.:936.0 3rd Qu.:1009.2   
## Max. :1.60 Max. :946.0 Max. :1175.3   
## ManufacturingProcess06 ManufacturingProcess07 ManufacturingProcess08  
## Min. :203.0 Min. :177.0 Min. :177.0   
## 1st Qu.:205.7 1st Qu.:177.0 1st Qu.:177.0   
## Median :206.8 Median :177.0 Median :178.0   
## Mean :207.4 Mean :177.5 Mean :177.6   
## 3rd Qu.:208.7 3rd Qu.:178.0 3rd Qu.:178.0   
## Max. :227.4 Max. :178.0 Max. :178.0   
## ManufacturingProcess09 ManufacturingProcess10 ManufacturingProcess11  
## Min. :38.89 Min. : 7.500 Min. : 7.500   
## 1st Qu.:44.89 1st Qu.: 8.700 1st Qu.: 9.000   
## Median :45.73 Median : 9.100 Median : 9.400   
## Mean :45.66 Mean : 9.188 Mean : 9.379   
## 3rd Qu.:46.52 3rd Qu.: 9.525 3rd Qu.: 9.900   
## Max. :49.36 Max. :11.600 Max. :11.500   
## ManufacturingProcess12 ManufacturingProcess13 ManufacturingProcess14  
## Min. : 0.0 Min. :32.10 Min. :4701   
## 1st Qu.: 0.0 1st Qu.:33.90 1st Qu.:4827   
## Median : 0.0 Median :34.60 Median :4856   
## Mean : 852.9 Mean :34.51 Mean :4853   
## 3rd Qu.: 0.0 3rd Qu.:35.20 3rd Qu.:4882   
## Max. :4549.0 Max. :38.60 Max. :5055   
## ManufacturingProcess15 ManufacturingProcess16 ManufacturingProcess17  
## Min. :5904 Min. : 0 Min. :31.30   
## 1st Qu.:6010 1st Qu.:4561 1st Qu.:33.50   
## Median :6032 Median :4588 Median :34.40   
## Mean :6039 Mean :4566 Mean :34.34   
## 3rd Qu.:6061 3rd Qu.:4619 3rd Qu.:35.10   
## Max. :6233 Max. :4852 Max. :40.00   
## ManufacturingProcess18 ManufacturingProcess19 ManufacturingProcess20  
## Min. : 0 Min. :5890 Min. : 0   
## 1st Qu.:4813 1st Qu.:6001 1st Qu.:4553   
## Median :4835 Median :6022 Median :4582   
## Mean :4810 Mean :6028 Mean :4556   
## 3rd Qu.:4862 3rd Qu.:6050 3rd Qu.:4610   
## Max. :4971 Max. :6146 Max. :4759   
## ManufacturingProcess21 ManufacturingProcess22 ManufacturingProcess23  
## Min. :-1.8000 Min. : 0.000 Min. :0.000   
## 1st Qu.:-0.6000 1st Qu.: 3.000 1st Qu.:2.000   
## Median :-0.3000 Median : 5.000 Median :3.000   
## Mean :-0.1642 Mean : 5.398 Mean :3.017   
## 3rd Qu.: 0.0000 3rd Qu.: 8.000 3rd Qu.:4.000   
## Max. : 3.6000 Max. :12.000 Max. :6.000   
## ManufacturingProcess24 ManufacturingProcess25 ManufacturingProcess26  
## Min. : 0.000 Min. : 0 Min. : 0   
## 1st Qu.: 4.000 1st Qu.:4834 1st Qu.:6021   
## Median : 8.000 Median :4856 Median :6047   
## Mean : 8.841 Mean :4832 Mean :6017   
## 3rd Qu.:14.000 3rd Qu.:4882 3rd Qu.:6073   
## Max. :23.000 Max. :4990 Max. :6161   
## ManufacturingProcess27 ManufacturingProcess28 ManufacturingProcess29  
## Min. : 0 Min. : 0.000 Min. : 0.0   
## 1st Qu.:4563 1st Qu.: 0.000 1st Qu.:19.7   
## Median :4588 Median :10.400 Median :19.9   
## Mean :4566 Mean : 6.405 Mean :20.0   
## 3rd Qu.:4614 3rd Qu.:10.700 3rd Qu.:20.4   
## Max. :4710 Max. :11.500 Max. :22.0   
## ManufacturingProcess30 ManufacturingProcess31 ManufacturingProcess32  
## Min. : 0.000 Min. : 0.00 Min. :143.0   
## 1st Qu.: 8.800 1st Qu.:70.10 1st Qu.:155.0   
## Median : 9.200 Median :70.80 Median :158.0   
## Mean : 9.188 Mean :70.19 Mean :158.5   
## 3rd Qu.: 9.700 3rd Qu.:71.40 3rd Qu.:162.0   
## Max. :11.200 Max. :72.50 Max. :173.0   
## ManufacturingProcess33 ManufacturingProcess34 ManufacturingProcess35  
## Min. :56.00 Min. :2.300 Min. :463   
## 1st Qu.:62.00 1st Qu.:2.500 1st Qu.:490   
## Median :64.00 Median :2.500 Median :496   
## Mean :63.48 Mean :2.496 Mean :496   
## 3rd Qu.:65.00 3rd Qu.:2.500 3rd Qu.:502   
## Max. :70.00 Max. :2.600 Max. :522   
## ManufacturingProcess36 ManufacturingProcess37 ManufacturingProcess38  
## Min. :0.01700 Min. :0.000 Min. :0.000   
## 1st Qu.:0.01900 1st Qu.:0.700 1st Qu.:2.000   
## Median :0.01900 Median :1.000 Median :3.000   
## Mean :0.01953 Mean :1.014 Mean :2.534   
## 3rd Qu.:0.02000 3rd Qu.:1.300 3rd Qu.:3.000   
## Max. :0.02200 Max. :2.300 Max. :3.000   
## ManufacturingProcess39 ManufacturingProcess40 ManufacturingProcess41  
## Min. :0.000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:7.100 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :7.200 Median :0.00000 Median :0.00000   
## Mean :6.851 Mean :0.01761 Mean :0.02358   
## 3rd Qu.:7.300 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :7.500 Max. :0.10000 Max. :0.20000   
## ManufacturingProcess42 ManufacturingProcess43 ManufacturingProcess44  
## Min. : 0.00 Min. : 0.0000 Min. :0.000   
## 1st Qu.:11.40 1st Qu.: 0.6000 1st Qu.:1.800   
## Median :11.60 Median : 0.8000 Median :1.900   
## Mean :11.21 Mean : 0.9119 Mean :1.805   
## 3rd Qu.:11.70 3rd Qu.: 1.0250 3rd Qu.:1.900   
## Max. :12.10 Max. :11.0000 Max. :2.100   
## ManufacturingProcess45  
## Min. :0.000   
## 1st Qu.:2.100   
## Median :2.200   
## Mean :2.138   
## 3rd Qu.:2.300   
## Max. :2.600

#### c) Split the data into a training and a test set, pre-process the data, and tune a model of your choice from this chapter. What is the optimal value of the performance metric?

#Smallest negative value is -1.8 so we'll add 2 to the column to make it positive for log tranformations  
completeData$ManufacturingProcess21 <- completeData$ManufacturingProcess21 + 2  
## 75% of the sample size  
smp\_size <- floor(0.75 \* nrow(completeData))  
## set the seed to make your partition reproducible  
set.seed(123)  
train\_ind <- sample(seq\_len(nrow(completeData)), size = smp\_size)  
train <- completeData[train\_ind, ]  
test <- completeData[-train\_ind, ]

Pre-Process and transform the Data

y <- train$Yield  
train.trim <- subset(train, select = -c(Yield))  
#The preProcess function returns a list of elements, such as vectors of standard deviations  
# and vector of means since scale and center were reqeuested.  
prepro\_set <- preProcess(train.trim, method = c('scale', 'center'))  
print(prepro\_set)

## Created from 132 samples and 57 variables  
##   
## Pre-processing:  
## - centered (57)  
## - ignored (0)  
## - scaled (57)

#The predict function uses the list of elements to produce a dataframe of pre-processed variables  
transformed\_train <- predict(prepro\_set, train.trim)

Building the Model using all predictor variables.

transformed\_train$Yield <- y  
model1 <- lm(Yield ~., data = transformed\_train)  
summary(model1)

##   
## Call:  
## lm(formula = Yield ~ ., data = transformed\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.79908 -0.48221 -0.06976 0.48632 2.16631   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 40.159697 0.092172 435.705 < 2e-16 \*\*\*  
## BiologicalMaterial01 0.202017 0.314215 0.643 0.52223   
## BiologicalMaterial02 -0.191507 0.621809 -0.308 0.75895   
## BiologicalMaterial03 3.012612 1.275181 2.362 0.02075 \*   
## BiologicalMaterial04 -0.176316 1.278741 -0.138 0.89070   
## BiologicalMaterial05 0.476217 0.255182 1.866 0.06593 .   
## BiologicalMaterial06 -3.086532 1.633032 -1.890 0.06261 .   
## BiologicalMaterial07 -0.161147 0.139102 -1.158 0.25034   
## BiologicalMaterial08 1.015059 0.615428 1.649 0.10326   
## BiologicalMaterial09 -1.448267 0.780764 -1.855 0.06754 .   
## BiologicalMaterial10 -0.335153 1.133504 -0.296 0.76829   
## BiologicalMaterial11 -0.442061 0.493837 -0.895 0.37357   
## BiologicalMaterial12 0.669643 0.614159 1.090 0.27905   
## ManufacturingProcess01 0.379582 0.226297 1.677 0.09763 .   
## ManufacturingProcess02 -0.579465 0.647839 -0.894 0.37394   
## ManufacturingProcess03 0.006602 0.155555 0.042 0.96626   
## ManufacturingProcess04 0.164460 0.257571 0.639 0.52509   
## ManufacturingProcess05 0.089941 0.147984 0.608 0.54517   
## ManufacturingProcess06 0.259113 0.179747 1.442 0.15359   
## ManufacturingProcess07 -0.086602 0.133636 -0.648 0.51894   
## ManufacturingProcess08 -0.162234 0.162686 -0.997 0.32186   
## ManufacturingProcess09 0.570596 0.370413 1.540 0.12766   
## ManufacturingProcess10 -0.869564 0.636866 -1.365 0.17621   
## ManufacturingProcess11 0.108475 0.721971 0.150 0.88097   
## ManufacturingProcess12 0.076218 0.258847 0.294 0.76923   
## ManufacturingProcess13 -0.381742 0.463222 -0.824 0.41249   
## ManufacturingProcess14 -0.555068 0.767378 -0.723 0.47173   
## ManufacturingProcess15 0.033174 0.688203 0.048 0.96168   
## ManufacturingProcess16 -0.109327 0.212231 -0.515 0.60798   
## ManufacturingProcess17 -0.065016 0.476017 -0.137 0.89173   
## ManufacturingProcess18 2.805363 2.301616 1.219 0.22671   
## ManufacturingProcess19 -0.600517 0.465998 -1.289 0.20148   
## ManufacturingProcess20 -2.821287 2.291912 -1.231 0.22218   
## ManufacturingProcess21 NA NA NA NA   
## ManufacturingProcess22 -0.331512 0.199268 -1.664 0.10036   
## ManufacturingProcess23 0.175541 0.176049 0.997 0.32191   
## ManufacturingProcess24 -0.431112 0.180848 -2.384 0.01966 \*   
## ManufacturingProcess25 -4.132329 6.914666 -0.598 0.55189   
## ManufacturingProcess26 3.655591 6.777841 0.539 0.59125   
## ManufacturingProcess27 -2.060077 3.699300 -0.557 0.57927   
## ManufacturingProcess28 -0.544470 0.220619 -2.468 0.01587 \*   
## ManufacturingProcess29 4.032978 2.210150 1.825 0.07202 .   
## ManufacturingProcess30 -1.300066 0.863825 -1.505 0.13652   
## ManufacturingProcess31 -0.168381 0.860092 -0.196 0.84532   
## ManufacturingProcess32 1.427597 0.453306 3.149 0.00235 \*\*   
## ManufacturingProcess33 -0.892563 0.402023 -2.220 0.02943 \*   
## ManufacturingProcess34 -0.044068 0.190832 -0.231 0.81800   
## ManufacturingProcess35 0.005478 0.236853 0.023 0.98161   
## ManufacturingProcess36 0.148635 0.320840 0.463 0.64451   
## ManufacturingProcess37 -0.407466 0.179050 -2.276 0.02572 \*   
## ManufacturingProcess38 -0.073788 0.188152 -0.392 0.69604   
## ManufacturingProcess39 -0.003817 0.273699 -0.014 0.98891   
## ManufacturingProcess40 0.011921 0.341555 0.035 0.97225   
## ManufacturingProcess41 0.104779 0.345256 0.303 0.76236   
## ManufacturingProcess42 0.420653 0.623317 0.675 0.50184   
## ManufacturingProcess43 0.139169 0.188633 0.738 0.46295   
## ManufacturingProcess44 -0.852203 0.518993 -1.642 0.10477   
## ManufacturingProcess45 0.387369 0.290189 1.335 0.18595   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.059 on 75 degrees of freedom  
## Multiple R-squared: 0.8118, Adjusted R-squared: 0.6713   
## F-statistic: 5.778 on 56 and 75 DF, p-value: 2.971e-12

The R-Squared value is at 0.8072 which tells us that the model is a decent fit for the data. The Adjusted R-Squared value decreases to 0.6632 which also tells us that many of predictors do not improve the model as much as they should. None of the predictors show to be significant to the model since most have high p-values.

*Tuning the Model*

Using backward stepwise regression. This takes all the variables and removes them based on their p-value. The variables left have the most significant impact on the model.

model.tune <- lm(Yield ~., data = transformed\_train)  
k <- ols\_step\_backward\_p(model.tune)

## Backward Elimination Method   
## ---------------------------  
##   
## Candidate Terms:   
##   
## 1 . BiologicalMaterial01   
## 2 . BiologicalMaterial02   
## 3 . BiologicalMaterial03   
## 4 . BiologicalMaterial04   
## 5 . BiologicalMaterial05   
## 6 . BiologicalMaterial06   
## 7 . BiologicalMaterial07   
## 8 . BiologicalMaterial08   
## 9 . BiologicalMaterial09   
## 10 . BiologicalMaterial10   
## 11 . BiologicalMaterial11   
## 12 . BiologicalMaterial12   
## 13 . ManufacturingProcess01   
## 14 . ManufacturingProcess02   
## 15 . ManufacturingProcess03   
## 16 . ManufacturingProcess04   
## 17 . ManufacturingProcess05   
## 18 . ManufacturingProcess06   
## 19 . ManufacturingProcess07   
## 20 . ManufacturingProcess08   
## 21 . ManufacturingProcess09   
## 22 . ManufacturingProcess10   
## 23 . ManufacturingProcess11   
## 24 . ManufacturingProcess12   
## 25 . ManufacturingProcess13   
## 26 . ManufacturingProcess14   
## 27 . ManufacturingProcess15   
## 28 . ManufacturingProcess16   
## 29 . ManufacturingProcess17   
## 30 . ManufacturingProcess18   
## 31 . ManufacturingProcess19   
## 32 . ManufacturingProcess20   
## 33 . ManufacturingProcess21   
## 34 . ManufacturingProcess22   
## 35 . ManufacturingProcess23   
## 36 . ManufacturingProcess24   
## 37 . ManufacturingProcess25   
## 38 . ManufacturingProcess26   
## 39 . ManufacturingProcess27   
## 40 . ManufacturingProcess28   
## 41 . ManufacturingProcess29   
## 42 . ManufacturingProcess30   
## 43 . ManufacturingProcess31   
## 44 . ManufacturingProcess32   
## 45 . ManufacturingProcess33   
## 46 . ManufacturingProcess34   
## 47 . ManufacturingProcess35   
## 48 . ManufacturingProcess36   
## 49 . ManufacturingProcess37   
## 50 . ManufacturingProcess38   
## 51 . ManufacturingProcess39   
## 52 . ManufacturingProcess40   
## 53 . ManufacturingProcess41   
## 54 . ManufacturingProcess42   
## 55 . ManufacturingProcess43   
## 56 . ManufacturingProcess44   
## 57 . ManufacturingProcess45   
##   
## We are eliminating variables based on p value...  
##   
## Variables Removed:

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess39

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess35

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess40

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess15

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess03

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - BiologicalMaterial04

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess11

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess31

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess34

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess12

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - BiologicalMaterial02

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess38

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess27

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess26

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - BiologicalMaterial11

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess36

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess07

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess05

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess23

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess04

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

## - ManufacturingProcess42

## Note: model has aliased coefficients  
## sums of squares computed by model comparison

##   
## No more variables satisfy the condition of p value = 0.3  
##   
##   
## Final Model Output   
## ------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Model Summary   
## ---------------------------------------------------------------  
## R 0.894 RMSE 0.969   
## R-Squared 0.798 Coef. Var 2.412   
## Adj. R-Squared 0.725 MSE 0.938   
## Pred R-Squared -10.530 MAE 0.646   
## ---------------------------------------------------------------  
## RMSE: Root Mean Square Error   
## MSE: Mean Square Error   
## MAE: Mean Absolute Error   
##   
## ANOVA   
## --------------------------------------------------------------------  
## Sum of   
## Squares DF Mean Square F Sig.   
## --------------------------------------------------------------------  
## Regression 356.865 35 10.196 10.866 0.0000   
## Residual 90.080 96 0.938   
## Total 446.945 131   
## --------------------------------------------------------------------  
##   
## Parameter Estimates   
## ----------------------------------------------------------------------------------------------------  
## model Beta Std. Error Std. Beta t Sig lower upper   
## ----------------------------------------------------------------------------------------------------  
## (Intercept) 40.160 0.084 476.320 0.000 39.992 40.327   
## BiologicalMaterial01 0.268 0.235 0.145 1.141 0.257 -0.198 0.735   
## BiologicalMaterial03 3.149 0.778 1.705 4.049 0.000 1.605 4.692   
## BiologicalMaterial05 0.408 0.183 0.221 2.231 0.028 0.045 0.772   
## BiologicalMaterial06 -3.472 0.944 -1.880 -3.680 0.000 -5.345 -1.599   
## BiologicalMaterial07 -0.133 0.105 -0.072 -1.272 0.207 -0.341 0.075   
## BiologicalMaterial08 0.751 0.403 0.407 1.864 0.065 -0.049 1.551   
## BiologicalMaterial09 -1.478 0.428 -0.800 -3.449 0.001 -2.328 -0.627   
## BiologicalMaterial10 -0.523 0.198 -0.283 -2.641 0.010 -0.916 -0.130   
## BiologicalMaterial12 0.471 0.362 0.255 1.302 0.196 -0.247 1.190   
## ManufacturingProcess01 0.475 0.139 0.257 3.414 0.001 0.199 0.751   
## ManufacturingProcess02 -0.364 0.340 -0.197 -1.068 0.288 -1.039 0.312   
## ManufacturingProcess06 0.353 0.142 0.191 2.484 0.015 0.071 0.636   
## ManufacturingProcess08 -0.151 0.125 -0.082 -1.207 0.231 -0.400 0.098   
## ManufacturingProcess09 0.696 0.256 0.377 2.723 0.008 0.189 1.203   
## ManufacturingProcess10 -0.936 0.532 -0.507 -1.760 0.082 -1.991 0.119   
## ManufacturingProcess13 -0.386 0.361 -0.209 -1.070 0.287 -1.101 0.330   
## ManufacturingProcess14 -0.537 0.470 -0.291 -1.143 0.256 -1.469 0.395   
## ManufacturingProcess16 -0.128 0.122 -0.070 -1.057 0.293 -0.370 0.113   
## ManufacturingProcess17 0.166 0.357 0.090 0.464 0.644 -0.543 0.874   
## ManufacturingProcess18 3.330 1.412 1.803 2.358 0.020 0.527 6.134   
## ManufacturingProcess19 -0.622 0.253 -0.337 -2.457 0.016 -1.124 -0.119   
## ManufacturingProcess20 -3.397 1.413 -1.839 -2.405 0.018 -6.201 -0.593   
## ManufacturingProcess21 NA 0.106 -0.113 -1.979 0.051 NA NA   
## ManufacturingProcess22 -0.209 0.117 -0.172 -2.721 0.008 -0.419 0.001   
## ManufacturingProcess24 -0.318 1.320 -2.065 -2.890 0.005 -0.550 -0.086   
## ManufacturingProcess25 -3.814 0.170 -0.284 -3.088 0.003 -6.434 -1.194   
## ManufacturingProcess28 -0.525 1.410 2.706 3.543 0.001 -0.863 -0.188   
## ManufacturingProcess29 4.998 0.469 -0.526 -2.070 0.041 2.198 7.798   
## ManufacturingProcess30 -0.971 0.242 0.690 5.276 0.000 -1.902 -0.040   
## ManufacturingProcess32 1.275 0.227 -0.480 -3.907 0.000 0.795 1.754   
## ManufacturingProcess33 -0.886 0.138 -0.198 -2.653 0.009 -1.337 -0.436   
## ManufacturingProcess37 -0.366 0.100 0.062 1.143 0.256 -0.640 -0.092   
## ManufacturingProcess41 0.114 0.146 0.100 1.266 0.209 -0.084 0.312   
## ManufacturingProcess43 0.185 0.274 -0.314 -2.118 0.037 -0.105 0.474   
## ManufacturingProcess44 -0.580 0.216 0.259 2.208 0.030 -1.124 -0.037   
## ManufacturingProcess45 0.478 NA 0.145 NA NA 0.048 0.907   
## ----------------------------------------------------------------------------------------------------

x <- k$removed  
print(x)

## [1] "ManufacturingProcess39" "ManufacturingProcess35"  
## [3] "ManufacturingProcess40" "ManufacturingProcess15"  
## [5] "ManufacturingProcess03" "BiologicalMaterial04"   
## [7] "ManufacturingProcess11" "ManufacturingProcess31"  
## [9] "ManufacturingProcess34" "ManufacturingProcess12"  
## [11] "BiologicalMaterial02" "ManufacturingProcess38"  
## [13] "ManufacturingProcess27" "ManufacturingProcess26"  
## [15] "BiologicalMaterial11" "ManufacturingProcess36"  
## [17] "ManufacturingProcess07" "ManufacturingProcess05"  
## [19] "ManufacturingProcess23" "ManufacturingProcess04"  
## [21] "ManufacturingProcess42"

The stepwise model removed all variables with a low p-value leaving. A total of 21 variables were removed, the majority of them ManufacturingProcess variables. The R-Squared value was almost the exact same at 0.796 with an Adjusted R-Square value at 0.722. With less variables without losing accuracy, I would use this as the primary model.

#### d) Predict the response for the test set. What is the value of the performance metric and how does this compare with the resampled performance metric on the training set?

y\_test <- test$Yield  
test.tune <- subset(test, select = -c(Yield))  
#Drop columns that were removed in the previous backward stepwise model  
test.tune <- test.tune[, !(names(test.tune) %in% x)]  
#The preProcess function returns a list of elements, such as vectors of standard deviations  
# and vector of means since scale and center were reqeuested.  
test\_set <- preProcess(test.tune, method = c('scale', 'center'))

## Warning in preProcess.default(test.tune, method = c("scale", "center")):  
## These variables have zero variances: BiologicalMaterial07

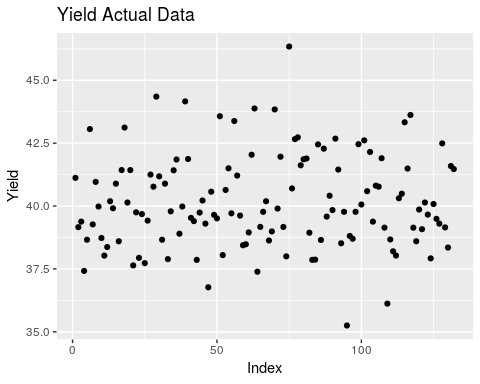
print(test\_set)

## Created from 44 samples and 36 variables  
##   
## Pre-processing:  
## - centered (36)  
## - ignored (0)  
## - scaled (36)

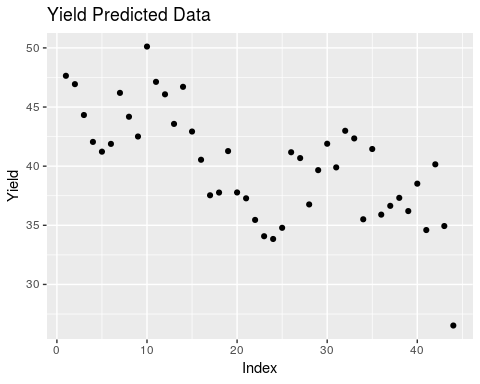
#The predict function uses the list of elements to produce a dataframe of pre-processed variables  
transformed\_test <- predict(test\_set, test.tune)  
pred <- predict(k$model, transformed\_test)

## Warning in predict.lm(k$model, transformed\_test): prediction from a rank-  
## deficient fit may be misleading

qplot(seq\_along(transformed\_train$Yield), transformed\_train$Yield, ylab = 'Yield', xlab = 'Index', main = 'Yield Actual Data', geom = 'point')



qplot(seq\_along(pred), pred, ylab = 'Yield', xlab = 'Index', main = 'Yield Predicted Data', geom = 'point')



RSME for predicted values

RMSE(pred, y\_test)

## [1] 3.929993

The performance metric was worse for the predicted data than the training data. Using the RSME metric, the training data had a 0.945. The testing data showed a 4.11 with the plots clearly showing how big the variance is between actual and predicted data.

#### e) Which predictors are most important in the model you have trained? Do either the biological or process predictors dominate the list?

The original dataset had 12 biological predictors and 45 process predictors. The backwards stepwise regression function dropped 3 biological and 15 process predictors, or 25% and 33% respectively. Process predictors dominate the list but the stepwise model calculated the a higher percentage of those were insignificant to the model.

#### f) Explore the relationships between each of the top predictors and the response. How could this information be helpful in improving yield in future runs of the manufacturing process?

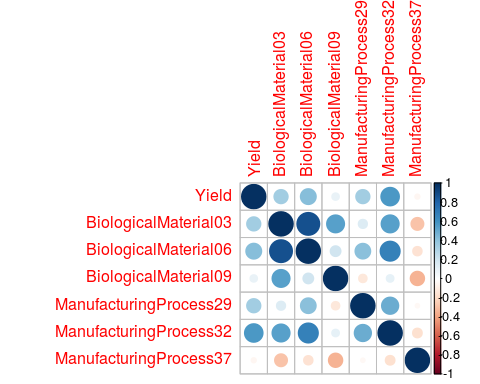
summary(k$model)

##   
## Call:  
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),   
## data = l)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.99990 -0.49908 -0.05963 0.53714 2.01401   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 40.15970 0.08431 476.320 < 2e-16 \*\*\*  
## BiologicalMaterial01 0.26844 0.23520 1.141 0.256567   
## BiologicalMaterial03 3.14853 0.77761 4.049 0.000104 \*\*\*  
## BiologicalMaterial05 0.40845 0.18304 2.231 0.027977 \*   
## BiologicalMaterial06 -3.47233 0.94363 -3.680 0.000385 \*\*\*  
## BiologicalMaterial07 -0.13300 0.10459 -1.272 0.206585   
## BiologicalMaterial08 0.75127 0.40303 1.864 0.065372 .   
## BiologicalMaterial09 -1.47753 0.42835 -3.449 0.000836 \*\*\*  
## BiologicalMaterial10 -0.52309 0.19809 -2.641 0.009659 \*\*   
## BiologicalMaterial12 0.47127 0.36195 1.302 0.196028   
## ManufacturingProcess01 0.47464 0.13901 3.414 0.000938 \*\*\*  
## ManufacturingProcess02 -0.36358 0.34042 -1.068 0.288184   
## ManufacturingProcess06 0.35350 0.14231 2.484 0.014726 \*   
## ManufacturingProcess08 -0.15123 0.12535 -1.207 0.230574   
## ManufacturingProcess09 0.69598 0.25558 2.723 0.007683 \*\*   
## ManufacturingProcess10 -0.93593 0.53164 -1.760 0.081511 .   
## ManufacturingProcess13 -0.38580 0.36054 -1.070 0.287274   
## ManufacturingProcess14 -0.53668 0.46961 -1.143 0.255957   
## ManufacturingProcess16 -0.12847 0.12156 -1.057 0.293219   
## ManufacturingProcess17 0.16564 0.35692 0.464 0.643631   
## ManufacturingProcess18 3.33040 1.41227 2.358 0.020392 \*   
## ManufacturingProcess19 -0.62179 0.25306 -2.457 0.015803 \*   
## ManufacturingProcess20 -3.39713 1.41268 -2.405 0.018102 \*   
## ManufacturingProcess21 NA NA NA NA   
## ManufacturingProcess22 -0.20935 0.10580 -1.979 0.050725 .   
## ManufacturingProcess24 -0.31790 0.11683 -2.721 0.007727 \*\*   
## ManufacturingProcess25 -3.81401 1.31978 -2.890 0.004765 \*\*   
## ManufacturingProcess28 -0.52520 0.17010 -3.088 0.002638 \*\*   
## ManufacturingProcess29 4.99774 1.41049 3.543 0.000612 \*\*\*  
## ManufacturingProcess30 -0.97097 0.46901 -2.070 0.041116 \*   
## ManufacturingProcess32 1.27482 0.24162 5.276 8.16e-07 \*\*\*  
## ManufacturingProcess33 -0.88641 0.22688 -3.907 0.000174 \*\*\*  
## ManufacturingProcess37 -0.36587 0.13792 -2.653 0.009341 \*\*   
## ManufacturingProcess41 0.11387 0.09959 1.143 0.255721   
## ManufacturingProcess43 0.18461 0.14581 1.266 0.208534   
## ManufacturingProcess44 -0.58023 0.27389 -2.118 0.036718 \*   
## ManufacturingProcess45 0.47762 0.21632 2.208 0.029631 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9687 on 96 degrees of freedom  
## Multiple R-squared: 0.7985, Adjusted R-squared: 0.725   
## F-statistic: 10.87 on 35 and 96 DF, p-value: < 2.2e-16

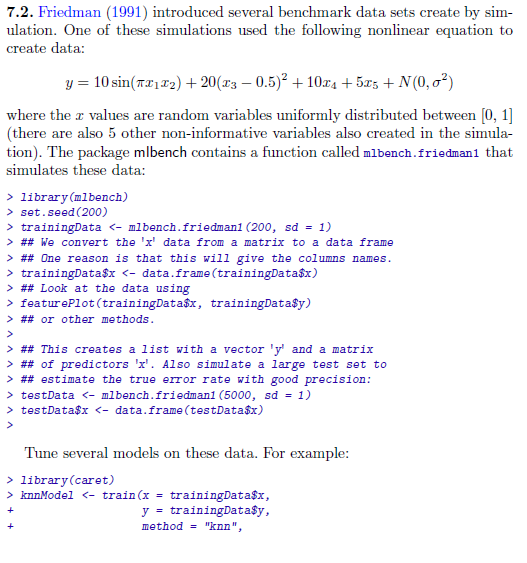
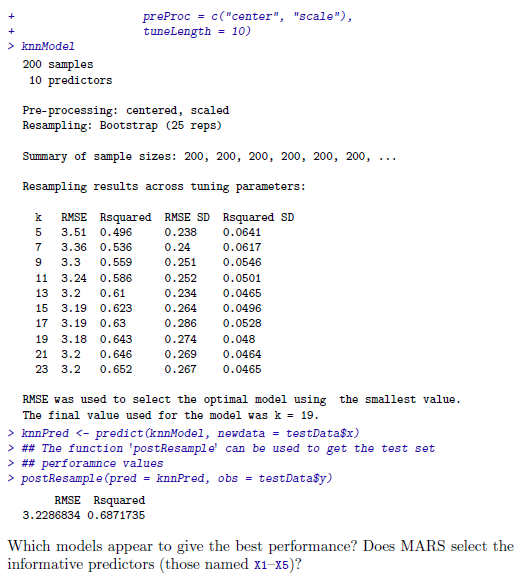
Top predictor variables in this model is BiologicalMaterial03, BiologicalMaterial06, BiologicalMaterial09, ManufacturingProcess29, ManufacturingProcess32, and ManufacturingProcess37 based on their p-values.

These predictors were slightly significant with the model of all variables though they became very significant when the stepwise function dropped insignificant variables.

top <- subset(test, select = c(Yield, BiologicalMaterial03, BiologicalMaterial06, BiologicalMaterial09,  
 ManufacturingProcess29, ManufacturingProcess32, ManufacturingProcess37))  
m <- cor(top)  
corrplot(m, method = 'circle')



From the correlation plot we see the BiologicalMaterial03 and 06 are highly correlated and may be redundant in the model. For the Manufacturing Process predictors, #32 seems to be the most correlated with the Yield. For future runs, more attention could be made to this process to help improve the yield.

library(mlbench)  
set.seed(200)  
# training data set  
trainingData = mlbench.friedman1(200, sd = 1)  
trainingData$x = data.frame(trainingData$x)  
# test data set  
testData = mlbench.friedman1(5000, sd = 1)  
testData$x = data.frame(testData$x)

We are going to investigate the relationship between the predictors X1-X10 and the response y in the training data:

head(trainingData$x, 5)

## X1 X2 X3 X4 X5 X6  
## 1 0.5337724 0.6478064 0.85078526 0.18159957 0.92903976 0.36179060  
## 2 0.5837650 0.4381528 0.67272659 0.66924914 0.16379784 0.45305931  
## 3 0.5895783 0.5879065 0.40967108 0.33812728 0.89409334 0.02681911  
## 4 0.6910399 0.2259548 0.03335447 0.06691274 0.63744519 0.52500637  
## 5 0.6673315 0.8188985 0.71676079 0.80324287 0.08306864 0.22344157  
## X7 X8 X9 X10  
## 1 0.8266609 0.4214081 0.59111440 0.5886216  
## 2 0.6489601 0.8446239 0.92819306 0.7584008  
## 3 0.1785614 0.3495908 0.01759542 0.4441185  
## 4 0.5133614 0.7970260 0.68986918 0.4450716  
## 5 0.6644906 0.9038919 0.39696995 0.5500808

head(trainingData$y, 5)

## [1] 18.46398 16.09836 17.76165 13.78730 18.42984

head(testData$x, 5)

## X1 X2 X3 X4 X5 X6  
## 1 0.49578735 0.2611653 0.8098616 0.823182309 0.8217010 0.32193361  
## 2 0.40777740 0.7163803 0.9635523 0.505653024 0.8802850 0.57453040  
## 3 0.49910226 0.7146126 0.6810968 0.003843726 0.4983548 0.06028523  
## 4 0.19563672 0.3689804 0.3779541 0.385691399 0.2787616 0.55466124  
## 5 0.02280348 0.7460249 0.3907981 0.873982576 0.1967682 0.17619075  
## X7 X8 X9 X10  
## 1 0.0544078 0.5185030 0.39143353 0.7389365041  
## 2 0.4551802 0.9806052 0.66632676 0.0005895968  
## 3 0.8925606 0.9750831 0.08564886 0.5922129985  
## 4 0.3972387 0.8403877 0.09039455 0.1622669764  
## 5 0.5066592 0.5561729 0.37898130 0.6500855859

head(testData$y, 5)

## [1] 17.518866 20.868802 12.817534 5.094464 10.793634

trainingData\_df1 = mutate(trainingData$x, y=trainingData$y)  
trainingData\_df2 = gather(trainingData\_df1, var, x, X1:X10)  
trainingData\_df2 = mutate(trainingData\_df2, var = forcats::fct\_relevel(factor(var), "X10", after = Inf))  
head(trainingData\_df1)

## X1 X2 X3 X4 X5 X6  
## 1 0.5337724 0.6478064 0.85078526 0.18159957 0.92903976 0.36179060  
## 2 0.5837650 0.4381528 0.67272659 0.66924914 0.16379784 0.45305931  
## 3 0.5895783 0.5879065 0.40967108 0.33812728 0.89409334 0.02681911  
## 4 0.6910399 0.2259548 0.03335447 0.06691274 0.63744519 0.52500637  
## 5 0.6673315 0.8188985 0.71676079 0.80324287 0.08306864 0.22344157  
## 6 0.8392937 0.3862983 0.64618857 0.86105431 0.63038947 0.43703891  
## X7 X8 X9 X10 y  
## 1 0.8266609 0.4214081 0.59111440 0.5886216 18.46398  
## 2 0.6489601 0.8446239 0.92819306 0.7584008 16.09836  
## 3 0.1785614 0.3495908 0.01759542 0.4441185 17.76165  
## 4 0.5133614 0.7970260 0.68986918 0.4450716 13.78730  
## 5 0.6644906 0.9038919 0.39696995 0.5500808 18.42984  
## 6 0.3360117 0.6489177 0.53116033 0.9066182 20.85817

head(trainingData\_df2)

## y var x  
## 1 18.46398 X1 0.5337724  
## 2 16.09836 X1 0.5837650  
## 3 17.76165 X1 0.5895783  
## 4 13.78730 X1 0.6910399  
## 5 18.42984 X1 0.6673315  
## 6 20.85817 X1 0.8392937

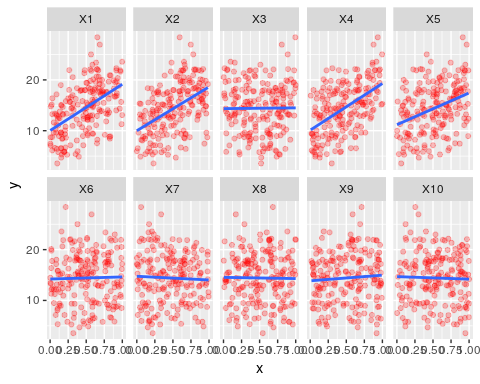
trainingData\_df2 <- trainingData\_df2[c("var", "x", "y")]  
head(trainingData\_df2)

## var x y  
## 1 X1 0.5337724 18.46398  
## 2 X1 0.5837650 16.09836  
## 3 X1 0.5895783 17.76165  
## 4 X1 0.6910399 13.78730  
## 5 X1 0.6673315 18.42984  
## 6 X1 0.8392937 20.85817

tail(trainingData\_df2)

## var x y  
## 1995 X10 0.93012621 15.33752  
## 1996 X10 0.60616776 17.17179  
## 1997 X10 0.07350651 10.49663  
## 1998 X10 0.69847823 14.79381  
## 1999 X10 0.39542251 17.33376  
## 2000 X10 0.21985817 17.15359

ggplot(trainingData\_df2, aes(x, y)) + geom\_point(color='red', alpha = 0.25) +   
 stat\_smooth(method = "glm", se = FALSE) + facet\_wrap(~ var, nrow = 2)



From the plot, we notice that the variables X1, X2, X4, & X5 show positive correlations with the response y. The remaining variables appear to show near-zero correlation to the response.

## Models

We are going to test this with three models: 1) K nearest neighbors model 2) MARS model  
3) SVM model

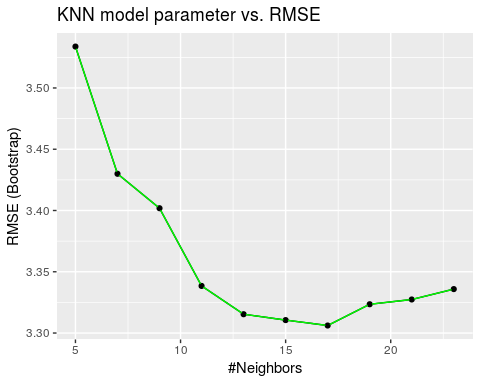
### K Nearest Neighbors Model

The K nearest neighbors model is used as the first model, with predictors centered and scaled:

library(caret)  
set.seed(100)  
fried\_knn = train(trainingData$x, trainingData$y, method = "knn", preProc = c("center", "scale"), tuneLength = 10)

The resampled RMSE of the model with various tuning parameters is shown below:

ggplot(fried\_knn, aes(x=k, y=RMSE)) + geom\_line(color="green") +   
 geom\_point(size = 1) + labs(title = "KNN model parameter vs. RMSE")



The optimal RMSE of this model occurs at = , which yields an RMSE of ‘r round(min(fried\_knn$results), 3)’.

### MARS Model

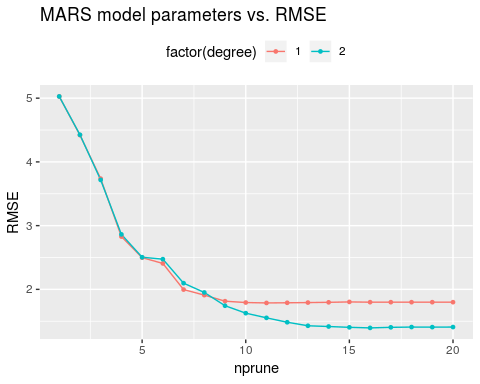
MARS model is used as the second model. It is used with the same pre-processing as the KNN model above, for first- and second-degree products and pruning parameter from 1-20:

set.seed(100)  
fried\_mars = train(trainingData$x, trainingData$y, method = "earth", preProcess = c("center", "scale"), tuneGrid = expand.grid(degree = 1:2, nprune = 1:20))

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =  
## trainInfo, : There were missing values in resampled performance measures.

The tuning profile of this model is shown below:

ggplot(fried\_mars$results, aes(x = nprune, y = RMSE, col = factor(degree))) + geom\_line() +   
 geom\_point(size = 1) + labs(title = "MARS model parameters vs. RMSE") + theme(legend.position = "top")



The optimal RMSE of ‘r round(min(fried\_marsresultsRMSE), 3)’ is obtained with a second-degree model with 16 parameters. The importance of the predictors is shown below:

‘r pander::pander(varImp(fried\_mars)[[“importance”]])’

We notice that the five informative predictors X1-X5 are the only predictors selected by the model.

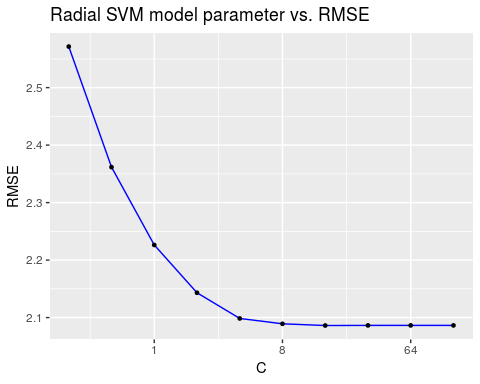
### SVM Model

SVM Model is used as the third model.

set.seed(100)  
fried\_svm = train(trainingData$x, trainingData$y, method = "svmRadial", preProc = c("center", "scale"), tuneLength = 10)

The tuning profile of the model is shown below:

ggplot(fried\_svm$results, aes(x = C, y = RMSE)) + geom\_line(color="blue") +   
 geom\_point(size = 1) + scale\_x\_continuous(trans = "log2") + labs(title = "Radial SVM model parameter vs. RMSE")



##### The optimal RMSE of ‘r round(min(fried\_svmresultsRMSE), 3)’ achieved with a cost parameter = and a shape parameter results. The RMSE of the model levels out above = .

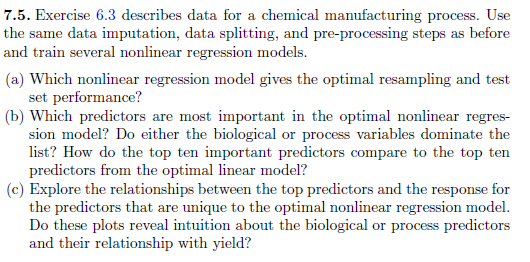
## Model Performance

Each of the models is used to predict for the test set, and their performance is measured:

fried\_knn\_prediction = predict(fried\_knn, testData$x)  
fried\_mars\_prediction = predict(fried\_mars, testData$x)  
fried\_svm\_prediction = predict(fried\_svm, testData$x)  
fried\_knn\_performance = defaultSummary(data.frame(obs = testData$y, pred = fried\_knn\_prediction))  
fried\_mars\_performance = defaultSummary(data.frame(obs = testData$y, pred = fried\_mars\_prediction[, 1]))  
fried\_svm\_performance = defaultSummary(data.frame(obs = testData$y, pred = fried\_svm\_prediction))  
pander(data.frame(RMSE = c(fried\_knn\_performance["RMSE"],   
 fried\_mars\_performance["RMSE"],   
 fried\_svm\_performance["RMSE"]),  
 row.names = c("KNN", "MARS", "SVM")))

|  |  |
| --- | --- |
|  | RMSE |
| **KNN** | 3.204 |
| **MARS** | 1.279 |
| **SVM** | 2.049 |

We notice that the most accurate model is the **MARS model** based on the resampled RMSE and performance against the test set.



#### a) Which nonlinear regression model gives the optimal resampling and test set performance.

#Get testing data and scale and center it  
y\_test <- test$Yield  
test.tune <- subset(test, select = -c(Yield))  
#The preProcess function returns a list of elements, such as vectors of standard deviations  
# and vector of means since scale and center were reqeuested.  
test\_set <- preProcess(test.tune, method = c('scale', 'center'))

## Warning in preProcess.default(test.tune, method = c("scale", "center")):  
## These variables have zero variances: BiologicalMaterial07

#The predict function uses the list of elements to produce a dataframe of pre-processed variables  
transformed\_test <- predict(test\_set, test.tune)

Neural Network Regression Model

y <- transformed\_train$Yield  
X <- subset(transformed\_train, select = -c(Yield))  
mod1 <- nnet(X, y, size = 10, linout = T, maxit = 500)

## # weights: 591  
## initial value 210370.031589   
## iter 10 value 3442.604934  
## iter 20 value 1986.106363  
## iter 30 value 619.894034  
## iter 40 value 301.185402  
## iter 50 value 205.140739  
## iter 60 value 124.728149  
## iter 70 value 56.017926  
## iter 80 value 17.152529  
## iter 90 value 8.506966  
## iter 100 value 6.048400  
## iter 110 value 4.738300  
## iter 120 value 3.532649  
## iter 130 value 2.229928  
## iter 140 value 1.767995  
## iter 150 value 1.566008  
## iter 160 value 1.450439  
## iter 170 value 1.408429  
## iter 180 value 1.402825  
## iter 190 value 1.377750  
## iter 200 value 1.242641  
## iter 210 value 1.009396  
## iter 220 value 0.945108  
## iter 230 value 0.944216  
## iter 240 value 0.944185  
## iter 250 value 0.940777  
## iter 260 value 0.940442  
## iter 270 value 0.940269  
## iter 280 value 0.939876  
## iter 290 value 0.938300  
## iter 300 value 0.922342  
## iter 310 value 0.915994  
## iter 320 value 0.903931  
## iter 330 value 0.877043  
## iter 340 value 0.835758  
## iter 350 value 0.782630  
## iter 360 value 0.717245  
## iter 370 value 0.706316  
## iter 380 value 0.673582  
## iter 390 value 0.338192  
## iter 400 value 0.022919  
## final value 0.000093   
## converged

pred1 <- predict(mod1, transformed\_test)

Support Vector Machines

mod2 <- train(X, y, method = 'svmRadial', tuneLength = 14, trControl = trainControl(method = 'cv'))  
mod2

## Support Vector Machines with Radial Basis Function Kernel   
##   
## 132 samples  
## 57 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 120, 118, 118, 119, 118, 120, ...   
## Resampling results across tuning parameters:  
##   
## C RMSE Rsquared MAE   
## 0.25 1.440908 0.4601284 1.1465883  
## 0.50 1.340948 0.4939208 1.0738737  
## 1.00 1.251971 0.5401044 0.9978593  
## 2.00 1.214127 0.5554844 0.9755667  
## 4.00 1.185436 0.5686885 0.9648934  
## 8.00 1.172149 0.5818170 0.9591794  
## 16.00 1.167814 0.5847865 0.9555041  
## 32.00 1.167814 0.5847865 0.9555041  
## 64.00 1.167814 0.5847865 0.9555041  
## 128.00 1.167814 0.5847865 0.9555041  
## 256.00 1.167814 0.5847865 0.9555041  
## 512.00 1.167814 0.5847865 0.9555041  
## 1024.00 1.167814 0.5847865 0.9555041  
## 2048.00 1.167814 0.5847865 0.9555041  
##   
## Tuning parameter 'sigma' was held constant at a value of 0.01389369  
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were sigma = 0.01389369 and C = 16.

pred2 <- predict(mod2$finalModel, transformed\_test)

K-Nearest Neighbors

mod3 <- train(X, y, method = 'knn', tuneGrid = data.frame(.k = 1:20),  
 trControl = trainControl(method = 'cv'))  
mod3

## k-Nearest Neighbors   
##   
## 132 samples  
## 57 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 119, 120, 118, 119, 118, 120, ...   
## Resampling results across tuning parameters:  
##   
## k RMSE Rsquared MAE   
## 1 1.421008 0.4917431 1.0547658  
## 2 1.223401 0.5823022 0.9850824  
## 3 1.264447 0.5931384 1.0279990  
## 4 1.277701 0.5895970 1.0271169  
## 5 1.296011 0.5611784 1.0670241  
## 6 1.298115 0.5678576 1.0528464  
## 7 1.316132 0.5646015 1.0821496  
## 8 1.316209 0.5595367 1.0753555  
## 9 1.315688 0.5662971 1.0615701  
## 10 1.320108 0.5619975 1.0652563  
## 11 1.349801 0.5383212 1.0930365  
## 12 1.371064 0.5331499 1.1087910  
## 13 1.377632 0.5302188 1.1173328  
## 14 1.384684 0.5207308 1.1185131  
## 15 1.391713 0.5248225 1.1196964  
## 16 1.405045 0.5157922 1.1231406  
## 17 1.413948 0.5001480 1.1210621  
## 18 1.420607 0.4966248 1.1309912  
## 19 1.424013 0.4900618 1.1328919  
## 20 1.435354 0.4796768 1.1488892  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was k = 2.

pred3 <- predict(mod3$finalModel, transformed\_test)

Comparing RMSE for the three models:

one <- RMSE(pred1, y\_test)  
print(paste0('Neural Network RMSE: ', one))

## [1] "Neural Network RMSE: 10.9544641077595"

two <- RMSE(pred2, y\_test)  
print(paste0('Support Vector Machines RMSE: ', two))

## [1] "Support Vector Machines RMSE: 1.06500637104131"

three <- RMSE(pred3, y\_test)  
print(paste0('k-Nearest Neighbors RMSE: ', three))

## [1] "k-Nearest Neighbors RMSE: 1.41080972881147"

The Support Vector Machines model has the lowest RMSE giving us the best model.

#### b) Which predictors are most important in the optimal nonlinear regression model? Do either 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?

varImp(mod2)

## loess r-squared variable importance  
##   
## only 20 most important variables shown (out of 57)  
##   
## Overall  
## ManufacturingProcess32 100.00  
## ManufacturingProcess13 94.05  
## BiologicalMaterial06 77.39  
## BiologicalMaterial03 75.78  
## ManufacturingProcess17 70.86  
## BiologicalMaterial02 65.86  
## ManufacturingProcess09 60.99  
## ManufacturingProcess36 58.09  
## BiologicalMaterial12 57.04  
## ManufacturingProcess33 51.83  
## ManufacturingProcess06 51.82  
## BiologicalMaterial11 50.21  
## ManufacturingProcess31 46.57  
## ManufacturingProcess29 43.09  
## BiologicalMaterial04 41.94  
## BiologicalMaterial09 38.50  
## BiologicalMaterial08 38.19  
## ManufacturingProcess11 31.73  
## BiologicalMaterial01 31.09  
## ManufacturingProcess12 27.97

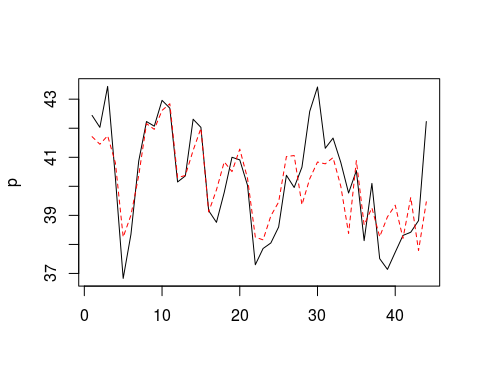
The top predictor variables from the SVM model are shown above. From that list, compared with the top linear regression model variables, ‘ManufacturingProcess32’, ‘BiologicalMaterial03’, and ‘BiologicalMaterial06’ were the only variables that were in both lists.

With the SVM model, there were 6 Manufacturing variables and 4 Biological variables in the top ten list. Considering there are many more Manufacturing variables than Biological variables overall, a higher percentage of Biological variables that were included may mean more significance in that context.

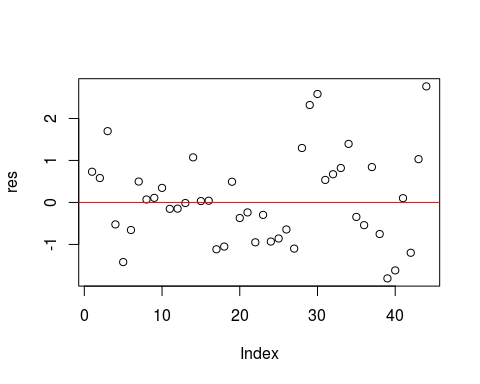
#### c) Explore the relationships between the top predictors and the response for the predictors that are unique to the optimal nonlinear regression model. Do these plots reveal intuition about the biological or process predictors and their relationship with yield?

Below, the actual data is in Black while the predicted data is in Red. We can see that the accuracy is very good and doesn’t seem to have much variance in most of the graph.

p <- data.frame('actual' = y\_test, 'pred' = pred2)  
matplot(p, type = 'l', pch = 1)

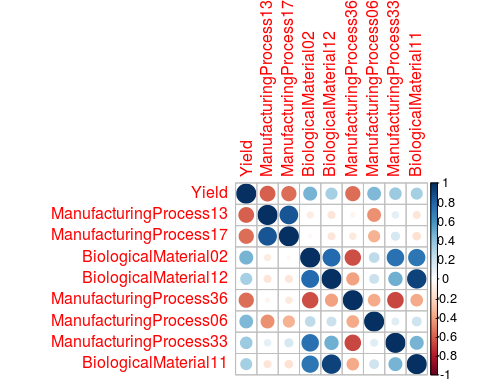


#Calculate residuals  
res <- y\_test-pred2  
plot(res)  
abline(h=0, col='red')

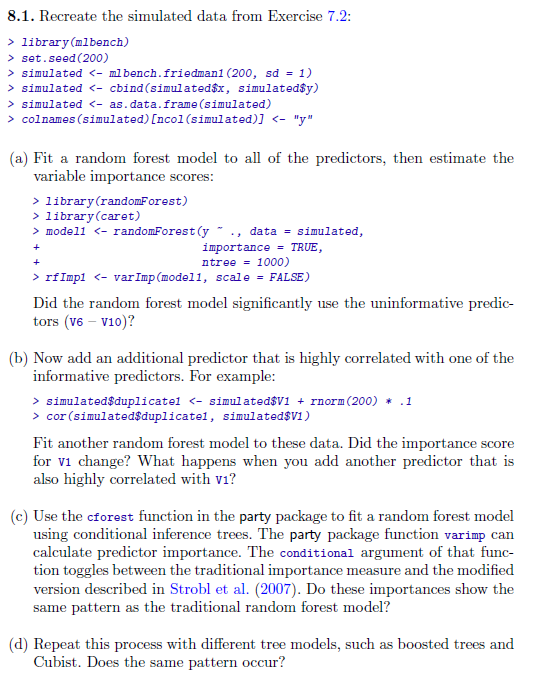
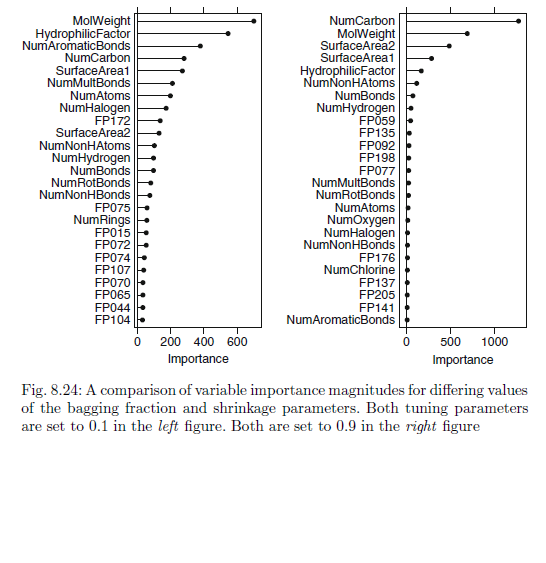


The residuals for this model seem to have a uniform variance though the variance may get greater towards the end. There doesn’t seem to be any noticeable skew.

top <- subset(test, select = c(Yield, ManufacturingProcess13, ManufacturingProcess17, BiologicalMaterial02,  
 BiologicalMaterial12, ManufacturingProcess36, ManufacturingProcess06,  
 ManufacturingProcess33, BiologicalMaterial11))  
m <- cor(top)  
corrplot(m, method = 'circle')



From these predictors we can see that many more variables are highly correlated. BioMaterial 11 and 12 are highly correlated as well as ManufProcess 13 and 17. An intuitive thought would be to look out for predictors that have negative correlation to Yield. ManufProcess13, 17, and 36 should be reformed or analyzed to understand why they are negatively correlated to Yield.

## Problem 8.1 from Max Kuhn, Kjell Johnson

####Setup

set.seed(200)  
simulated <- mlbench.friedman1(200, sd =1)  
simulated <- cbind(simulated$x, simulated$y)  
simulated <- as.data.frame(simulated)  
colnames(simulated)[ncol(simulated)] <- "y"

###8.1A Fit a random forest to model to all predictors and estimate the importance scores of the variables.

model1 <- randomForest(y ~ . , data= simulated, importance = TRUE, ntree = 1000)  
rfImp1 <- varImp(model1, scale = FALSE)  
rfImp1

## Overall  
## V1 8.732235404  
## V2 6.415369387  
## V3 0.763591825  
## V4 7.615118809  
## V5 2.023524577  
## V6 0.165111172  
## V7 -0.005961659  
## V8 -0.166362581  
## V9 -0.095292651  
## V10 -0.074944788

Examining the predictors, we see that variable importance for variables 6-10, show that these variables do not contribute much. Variable 6 has minimal predictive ability but 7-10 have very small importance values and are negative.

###8.1B

Now we will add another highly correlated predictor variable, and see how the model changes predictive weights. First we add in a variable that is a slight variation on V1, and we see about 94% correlation.

simulatedAddCorVar <- simulated  
simulatedAddCorVar$duplicate1 <- simulatedAddCorVar$V1 + rnorm(200) \* .1  
cor(simulatedAddCorVar$duplicate1, simulatedAddCorVar$V1)

## [1] 0.9460206

Let’s refit a model with the random forest model. When we do that we see that the importance of V1 decreases with the new near “duplicative” predictor we added. Basically adding a well correlated variable decreases the importance of the first variable.

model1 <- randomForest(y ~ . , data= simulatedAddCorVar, importance = TRUE, ntree = 1000)  
rfImp1 <- varImp(model1, scale = FALSE)  
rfImp1

## Overall  
## V1 5.69119973  
## V2 6.06896061  
## V3 0.62970218  
## V4 7.04752238  
## V5 1.87238438  
## V6 0.13569065  
## V7 -0.01345645  
## V8 -0.04370565  
## V9 0.00840438  
## V10 0.02894814  
## duplicate1 4.28331581

###8.1C Fit a different model, cforest, to the data, and examine the predictive values of the variables with this model. We will check the variable importance with the varimp conditional flag set to true and false and compare the differences. The conditional toggle will switch between weighting importance using traditional measures and those described by Strobl et al here: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-307

We see that the predictive values (with either toggle of the conditional flag), while not the same do closely mirror what we saw with the original random forest model, in that the relative importance of each predictor to the others has not changd (e.g., V3 in each model is less important the the other variables from V1-V5). We also see with the conditional flag set, the predictive values of each of the first 5 variables are all closer together.

model1 <- cforest(y ~ . , data= simulated, controls = cforest\_unbiased(ntree = 1000))  
cfImp1 <- varimp(model1)  
cfImp1

## V1 V2 V3 V4 V5 V6   
## 8.90081100 6.75099581 0.02443900 8.18768625 1.97955106 -0.01082758   
## V7 V8 V9 V10   
## 0.04196924 -0.02263576 -0.04148320 -0.04539002

cfImp1 <- varimp(model1, conditional = TRUE)  
cfImp1

## V1 V2 V3 V4 V5   
## 5.466680142 5.141732988 0.004600345 6.778369754 1.225878045   
## V6 V7 V8 V9 V10   
## -0.017944692 -0.019609289 -0.012222203 -0.012511252 -0.017915846

###8.1D

We will repeat the above process with some other models: “bagged” tree, “boosted” tree model, as well as the Cubist model.

First the bagged tree model. Here we see that variable importance is noticeably different from the prior models. V6-V10 are still do not have as great an importance, but their relative importance is higher. And the importance between variables V1-V5 does not vary nearly as much as some of the other models.

baggedTree <- bagging(y ~ ., data = simulated)  
bagImp1 <- varImp(baggedTree)  
bagImp1

## Overall  
## V1 1.7095009  
## V10 0.6917125  
## V2 2.3712173  
## V3 1.2458586  
## V4 2.7209707  
## V5 2.4329853  
## V6 0.8634778  
## V7 1.0400758  
## V8 0.6162473  
## V9 0.6134991

We also tried the bagged model with the additional correlated variable. It’s weighting of variable importance, unlike prior models did not change much with the addition of the correlated variable.

baggedTree <- baggedTree <- bagging(y ~ ., data = simulatedAddCorVar)  
bagImp1 <- varImp(baggedTree)  
bagImp1

## Overall  
## duplicate1 1.7650418  
## V1 1.8127686  
## V10 0.6603158  
## V2 2.1997814  
## V3 1.1348823  
## V4 2.5883058  
## V5 2.0678788  
## V6 0.7816333  
## V7 0.7845731  
## V8 0.3889693  
## V9 0.5025792

The Boosted Trees model. Here we see that this model suggests only using 4 variables as predictors (V1, V2, V4, V5).

boostTreeModel <- gbm.fit(simulated[,1:10], simulated[,11], distribution = "gaussian", verbose = FALSE)  
boostImp <- varImp(boostTreeModel, numTrees = 100)  
boostImp

## Overall  
## V1 39942.3385  
## V2 14208.0689  
## V3 0.0000  
## V4 14743.0683  
## V5 454.6016  
## V6 0.0000  
## V7 0.0000  
## V8 0.0000  
## V9 0.0000  
## V10 0.0000

If we add in the additional correlated variable, we see that it reduces the predictive value of many of the variables as before.

boostTreeModel <- gbm.fit(simulatedAddCorVar[,c(1:10,12)], simulatedAddCorVar[,11], distribution = "gaussian", verbose = FALSE)  
boostImp <- varImp(boostTreeModel, numTrees = 100)  
boostImp

## Overall  
## V1 33762.093  
## V2 13612.944  
## V3 0.000  
## V4 7731.723  
## V5 0.000  
## V6 0.000  
## V7 0.000  
## V8 0.000  
## V9 0.000  
## V10 0.000  
## duplicate1 17850.179

Lastly now the Cubist model.

It gives predictive values for four variables only (V1,V2,V4,V5) and interestingly, the predictive value for each of them is equally weighted.

library(Cubist)  
cubistMod <- cubist(simulated[,1:10], simulated[,11])  
varImp(cubistMod)

## Overall  
## V1 50  
## V2 50  
## V4 50  
## V5 50  
## V3 0  
## V6 0  
## V7 0  
## V8 0  
## V9 0  
## V10 0

If we add in the additional correlated variable to the cubst model, it does not affect the predictive value of the four variables that were predicive, it just adds in the correlated variable at the same predictive level.

library(Cubist)  
cubistMod <- cubist(simulatedAddCorVar[,c(1:10,12)], simulatedAddCorVar[,11],)  
varImp(cubistMod)

## Overall  
## V1 50  
## V2 50  
## V4 50  
## V5 50  
## duplicate1 50  
## V3 0  
## V6 0  
## V7 0  
## V8 0  
## V9 0  
## V10 0

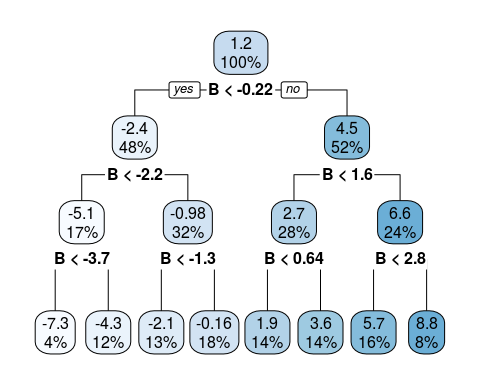
### 8.2) Use a simulation to show tree bias with different granularities.

Single regression trees are more diluted in result with single bias.Predictors with a higher number of distinct values (lower variance) are choosen over more granular (higher variance) predictors. data consisting of both informative and noise variables, and the noise variables have many more splits than the informative variables. There is therefore a high probability that the noise variables will be chosen to split the top nodes of the tree.

library(rpart)  
library("rpart.plot")  
library(mlbench)  
library(caret)

set.seed(500)  
A <- rep(1:2, each=1000)  
z <- A + rnorm(200, mean=0, sd=4)  
  
set.seed(500)  
B <- rnorm(200, mean=0, sd=2)  
simData <- data.frame(z=z, A=A, B=B)  
  
set.seed(624)  
fit <- rpart(z ~ ., data = simData)

rpart.plot(fit)



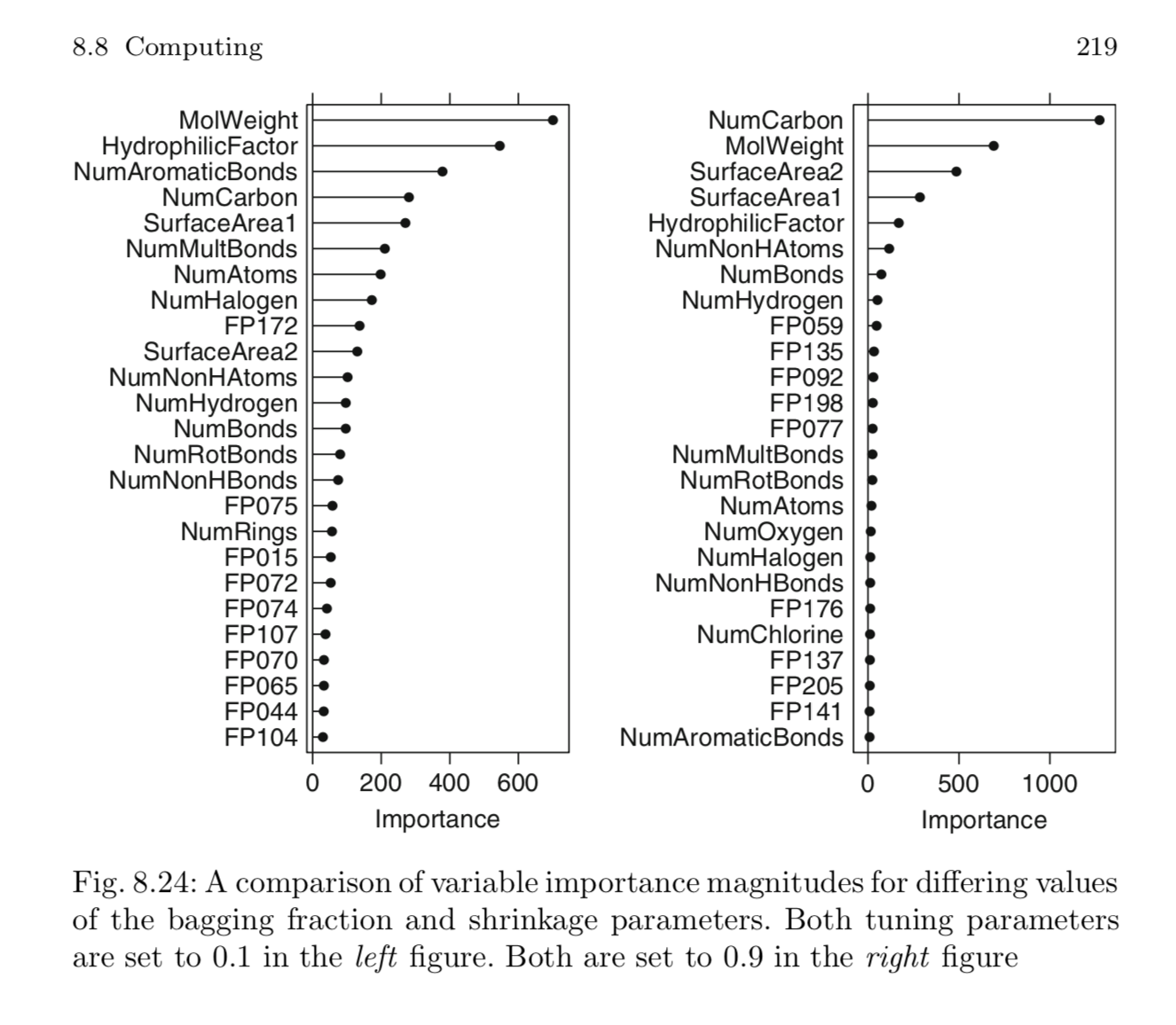
varImp(fit)

## Overall  
## A 0.6443238  
## B 4.5502846

In the simulation below,a predictor ‘A’ is created.Response variable Z is created that is equal to addition of ‘A’ and Gaussian variable with μ=0,σ=2 and a higher variance Gaussian variable B with μ=0,σ=4 is created. These variables are then placed in a data frame and modeled using the rpart() function which makes splits. The varImp() function is used to calculate the variable importance for the model.

The selection bias of Single Regression Trees is evident from this example. The lower variance predictor ‘A’ which is related to ‘z’ is found to be less important than the higher variance ‘B’ predictor which is completely unrelated to ‘z’.The importance of the unrelated B predictor is not just more,it is almost more thant four times.

#### 8.3)



Figure

#### (a) Why does the model on the right focus its importance on just the first few of predictors, whereas the model on the left spreads importance across more predictors?

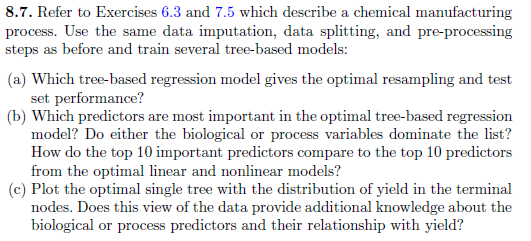
Lower learning rate of 0.1 in the left model results in fewer predictors than the model on the right which has a 0.9 learning rate. As randomness increases,there is increase in number of predictors.Therefore, the lower bagging fraction of 0.1 in the left model results in fewer predictors than the more deterministic model on the right which selects 0.9 of the training set observations to propose the next tree in the expansion.

#### (b) Which model do you think would be more predictive of other samples?

Model with leraning rate of 0.9 is more prone to overfitting therefore model on the left with learning rate of 0.1 is more predictive of other samples.

#### (c) How would increasing interaction depth affect the slope of predictor im-portance for either model in Fig. 8.24?

1. The positive impact of interaction depth on RMSE is proportional to the learning rate and number of trees.
2. An increase in interaction depth will increase the number of predictors and RMSE. Variable importance will be spread across more predictors and the Iterations graph will have a higher intercept that increases the negative slope.



8.7) Refer to Exercises 6.3 and 7.5 which describe a chemical manufacturing process. Use the same data imputation, data splitting, and pre-processing steps as before and train several tree-based models:

data("ChemicalManufacturingProcess")  
processPredictors <- ChemicalManufacturingProcess

summary(processPredictors)

## Yield BiologicalMaterial01 BiologicalMaterial02  
## Min. :35.25 Min. :4.580 Min. :46.87   
## 1st Qu.:38.75 1st Qu.:5.978 1st Qu.:52.68   
## Median :39.97 Median :6.305 Median :55.09   
## Mean :40.18 Mean :6.411 Mean :55.69   
## 3rd Qu.:41.48 3rd Qu.:6.870 3rd Qu.:58.74   
## Max. :46.34 Max. :8.810 Max. :64.75   
##   
## BiologicalMaterial03 BiologicalMaterial04 BiologicalMaterial05  
## Min. :56.97 Min. : 9.38 Min. :13.24   
## 1st Qu.:64.98 1st Qu.:11.24 1st Qu.:17.23   
## Median :67.22 Median :12.10 Median :18.49   
## Mean :67.70 Mean :12.35 Mean :18.60   
## 3rd Qu.:70.43 3rd Qu.:13.22 3rd Qu.:19.90   
## Max. :78.25 Max. :23.09 Max. :24.85   
##   
## BiologicalMaterial06 BiologicalMaterial07 BiologicalMaterial08  
## Min. :40.60 Min. :100.0 Min. :15.88   
## 1st Qu.:46.05 1st Qu.:100.0 1st Qu.:17.06   
## Median :48.46 Median :100.0 Median :17.51   
## Mean :48.91 Mean :100.0 Mean :17.49   
## 3rd Qu.:51.34 3rd Qu.:100.0 3rd Qu.:17.88   
## Max. :59.38 Max. :100.8 Max. :19.14   
##   
## BiologicalMaterial09 BiologicalMaterial10 BiologicalMaterial11  
## Min. :11.44 Min. :1.770 Min. :135.8   
## 1st Qu.:12.60 1st Qu.:2.460 1st Qu.:143.8   
## Median :12.84 Median :2.710 Median :146.1   
## Mean :12.85 Mean :2.801 Mean :147.0   
## 3rd Qu.:13.13 3rd Qu.:2.990 3rd Qu.:149.6   
## Max. :14.08 Max. :6.870 Max. :158.7   
##   
## BiologicalMaterial12 ManufacturingProcess01 ManufacturingProcess02  
## Min. :18.35 Min. : 0.00 Min. : 0.00   
## 1st Qu.:19.73 1st Qu.:10.80 1st Qu.:19.30   
## Median :20.12 Median :11.40 Median :21.00   
## Mean :20.20 Mean :11.21 Mean :16.68   
## 3rd Qu.:20.75 3rd Qu.:12.15 3rd Qu.:21.50   
## Max. :22.21 Max. :14.10 Max. :22.50   
## NA's :1 NA's :3   
## ManufacturingProcess03 ManufacturingProcess04 ManufacturingProcess05  
## Min. :1.47 Min. :911.0 Min. : 923.0   
## 1st Qu.:1.53 1st Qu.:928.0 1st Qu.: 986.8   
## Median :1.54 Median :934.0 Median : 999.2   
## Mean :1.54 Mean :931.9 Mean :1001.7   
## 3rd Qu.:1.55 3rd Qu.:936.0 3rd Qu.:1008.9   
## Max. :1.60 Max. :946.0 Max. :1175.3   
## NA's :15 NA's :1 NA's :1   
## ManufacturingProcess06 ManufacturingProcess07 ManufacturingProcess08  
## Min. :203.0 Min. :177.0 Min. :177.0   
## 1st Qu.:205.7 1st Qu.:177.0 1st Qu.:177.0   
## Median :206.8 Median :177.0 Median :178.0   
## Mean :207.4 Mean :177.5 Mean :177.6   
## 3rd Qu.:208.7 3rd Qu.:178.0 3rd Qu.:178.0   
## Max. :227.4 Max. :178.0 Max. :178.0   
## NA's :2 NA's :1 NA's :1   
## ManufacturingProcess09 ManufacturingProcess10 ManufacturingProcess11  
## Min. :38.89 Min. : 7.500 Min. : 7.500   
## 1st Qu.:44.89 1st Qu.: 8.700 1st Qu.: 9.000   
## Median :45.73 Median : 9.100 Median : 9.400   
## Mean :45.66 Mean : 9.179 Mean : 9.386   
## 3rd Qu.:46.52 3rd Qu.: 9.550 3rd Qu.: 9.900   
## Max. :49.36 Max. :11.600 Max. :11.500   
## NA's :9 NA's :10   
## ManufacturingProcess12 ManufacturingProcess13 ManufacturingProcess14  
## Min. : 0.0 Min. :32.10 Min. :4701   
## 1st Qu.: 0.0 1st Qu.:33.90 1st Qu.:4828   
## Median : 0.0 Median :34.60 Median :4856   
## Mean : 857.8 Mean :34.51 Mean :4854   
## 3rd Qu.: 0.0 3rd Qu.:35.20 3rd Qu.:4882   
## Max. :4549.0 Max. :38.60 Max. :5055   
## NA's :1 NA's :1   
## ManufacturingProcess15 ManufacturingProcess16 ManufacturingProcess17  
## Min. :5904 Min. : 0 Min. :31.30   
## 1st Qu.:6010 1st Qu.:4561 1st Qu.:33.50   
## Median :6032 Median :4588 Median :34.40   
## Mean :6039 Mean :4566 Mean :34.34   
## 3rd Qu.:6061 3rd Qu.:4619 3rd Qu.:35.10   
## Max. :6233 Max. :4852 Max. :40.00   
##   
## ManufacturingProcess18 ManufacturingProcess19 ManufacturingProcess20  
## Min. : 0 Min. :5890 Min. : 0   
## 1st Qu.:4813 1st Qu.:6001 1st Qu.:4553   
## Median :4835 Median :6022 Median :4582   
## Mean :4810 Mean :6028 Mean :4556   
## 3rd Qu.:4862 3rd Qu.:6050 3rd Qu.:4610   
## Max. :4971 Max. :6146 Max. :4759   
##   
## ManufacturingProcess21 ManufacturingProcess22 ManufacturingProcess23  
## Min. :-1.8000 Min. : 0.000 Min. :0.000   
## 1st Qu.:-0.6000 1st Qu.: 3.000 1st Qu.:2.000   
## Median :-0.3000 Median : 5.000 Median :3.000   
## Mean :-0.1642 Mean : 5.406 Mean :3.017   
## 3rd Qu.: 0.0000 3rd Qu.: 8.000 3rd Qu.:4.000   
## Max. : 3.6000 Max. :12.000 Max. :6.000   
## NA's :1 NA's :1   
## ManufacturingProcess24 ManufacturingProcess25 ManufacturingProcess26  
## Min. : 0.000 Min. : 0 Min. : 0   
## 1st Qu.: 4.000 1st Qu.:4832 1st Qu.:6020   
## Median : 8.000 Median :4855 Median :6047   
## Mean : 8.834 Mean :4828 Mean :6016   
## 3rd Qu.:14.000 3rd Qu.:4877 3rd Qu.:6070   
## Max. :23.000 Max. :4990 Max. :6161   
## NA's :1 NA's :5 NA's :5   
## ManufacturingProcess27 ManufacturingProcess28 ManufacturingProcess29  
## Min. : 0 Min. : 0.000 Min. : 0.00   
## 1st Qu.:4560 1st Qu.: 0.000 1st Qu.:19.70   
## Median :4587 Median :10.400 Median :19.90   
## Mean :4563 Mean : 6.592 Mean :20.01   
## 3rd Qu.:4609 3rd Qu.:10.750 3rd Qu.:20.40   
## Max. :4710 Max. :11.500 Max. :22.00   
## NA's :5 NA's :5 NA's :5   
## ManufacturingProcess30 ManufacturingProcess31 ManufacturingProcess32  
## Min. : 0.000 Min. : 0.00 Min. :143.0   
## 1st Qu.: 8.800 1st Qu.:70.10 1st Qu.:155.0   
## Median : 9.100 Median :70.80 Median :158.0   
## Mean : 9.161 Mean :70.18 Mean :158.5   
## 3rd Qu.: 9.700 3rd Qu.:71.40 3rd Qu.:162.0   
## Max. :11.200 Max. :72.50 Max. :173.0   
## NA's :5 NA's :5   
## ManufacturingProcess33 ManufacturingProcess34 ManufacturingProcess35  
## Min. :56.00 Min. :2.300 Min. :463.0   
## 1st Qu.:62.00 1st Qu.:2.500 1st Qu.:490.0   
## Median :64.00 Median :2.500 Median :495.0   
## Mean :63.54 Mean :2.494 Mean :495.6   
## 3rd Qu.:65.00 3rd Qu.:2.500 3rd Qu.:501.5   
## Max. :70.00 Max. :2.600 Max. :522.0   
## NA's :5 NA's :5 NA's :5   
## ManufacturingProcess36 ManufacturingProcess37 ManufacturingProcess38  
## Min. :0.01700 Min. :0.000 Min. :0.000   
## 1st Qu.:0.01900 1st Qu.:0.700 1st Qu.:2.000   
## Median :0.02000 Median :1.000 Median :3.000   
## Mean :0.01957 Mean :1.014 Mean :2.534   
## 3rd Qu.:0.02000 3rd Qu.:1.300 3rd Qu.:3.000   
## Max. :0.02200 Max. :2.300 Max. :3.000   
## NA's :5   
## ManufacturingProcess39 ManufacturingProcess40 ManufacturingProcess41  
## Min. :0.000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:7.100 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :7.200 Median :0.00000 Median :0.00000   
## Mean :6.851 Mean :0.01771 Mean :0.02371   
## 3rd Qu.:7.300 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :7.500 Max. :0.10000 Max. :0.20000   
## NA's :1 NA's :1   
## ManufacturingProcess42 ManufacturingProcess43 ManufacturingProcess44  
## Min. : 0.00 Min. : 0.0000 Min. :0.000   
## 1st Qu.:11.40 1st Qu.: 0.6000 1st Qu.:1.800   
## Median :11.60 Median : 0.8000 Median :1.900   
## Mean :11.21 Mean : 0.9119 Mean :1.805   
## 3rd Qu.:11.70 3rd Qu.: 1.0250 3rd Qu.:1.900   
## Max. :12.10 Max. :11.0000 Max. :2.100   
##   
## ManufacturingProcess45  
## Min. :0.000   
## 1st Qu.:2.100   
## Median :2.200   
## Mean :2.138   
## 3rd Qu.:2.300   
## Max. :2.600   
##

A small percentage of cells in the redictor set contain missing values. Use an knnimputation function to fill in these missing values

The summary for this dataset shows that many columns have 5 or less missing values. The column ManufacturingProcess03 had the most missing values with 15.

# imputing null values  
completeData <- knnImputation(ChemicalManufacturingProcess[, 1:57], k = 3, meth = "weighAvg")  
# Standardizing and scaling the predictors  
completeData[,2:(ncol(completeData))] <- scale(completeData[,2:(ncol(completeData))])  
# Splitting the data into training and testing data sets.  
set.seed(1)  
data\_training <- createDataPartition(completeData$Yield, p = 0.80, list=FALSE)  
training <- completeData[ data\_training,]  
testing <- completeData[-data\_training,]  
X\_train <- training[,2:(length(training))]  
Y\_train <- training$Yield  
X\_test <- testing[,2:(length(testing))]  
Y\_test <- testing$Yield

data is now imputed, cleaned, pre-processed and splitted.

head(completeData)

## Yield BiologicalMaterial01 BiologicalMaterial02 BiologicalMaterial03  
## 1 38.00 -0.2261036 -1.5140979 -2.68303622  
## 2 42.44 2.2391498 1.3089960 -0.05623504  
## 3 42.03 2.2391498 1.3089960 -0.05623504  
## 4 41.42 2.2391498 1.3089960 -0.05623504  
## 5 42.49 1.4827653 1.8939391 1.13594780  
## 6 43.57 -0.4081962 0.6620886 -0.59859075  
## BiologicalMaterial04 BiologicalMaterial05 BiologicalMaterial06  
## 1 0.2201765 0.4941942 -1.3828880  
## 2 1.2964386 0.4128555 1.1290767  
## 3 1.2964386 0.4128555 1.1290767  
## 4 1.2964386 0.4128555 1.1290767  
## 5 0.9414412 -0.3734185 1.5348350  
## 6 1.5894524 1.7305423 0.6192092  
## BiologicalMaterial07 BiologicalMaterial08 BiologicalMaterial09  
## 1 -0.1313107 -1.233131 -3.3962895  
## 2 -0.1313107 2.282619 -0.7227225  
## 3 -0.1313107 2.282619 -0.7227225  
## 4 -0.1313107 2.282619 -0.7227225  
## 5 -0.1313107 1.071310 -0.1205678  
## 6 -0.1313107 1.189487 -1.7343424  
## BiologicalMaterial10 BiologicalMaterial11 BiologicalMaterial12  
## 1 1.1005296 -1.838655 -1.7709224  
## 2 1.1005296 1.393395 1.0989855  
## 3 1.1005296 1.393395 1.0989855  
## 4 1.1005296 1.393395 1.0989855  
## 5 0.4162193 0.136256 1.0989855  
## 6 1.6346255 1.022062 0.7240877  
## ManufacturingProcess01 ManufacturingProcess02 ManufacturingProcess03  
## 1 0.2006342 0.5551856 0.3045192  
## 2 -6.1677821 -1.9906511 0.5708010  
## 3 -6.1677821 -1.9906511 0.1938106  
## 4 -6.1677821 -1.9906511 0.5427779  
## 5 -0.2803476 -1.9906511 0.4262861  
## 6 0.4349481 -1.9906511 0.4262861  
## ManufacturingProcess04 ManufacturingProcess05 ManufacturingProcess06  
## 1 0.2349609 -0.43413255 -0.4693269  
## 2 -2.3747305 1.00413856 0.9730078  
## 3 -3.1737741 0.06509023 -0.1046607  
## 4 -3.3335828 0.42626267 2.1993201  
## 5 -2.2149218 0.84981943 -0.6249144  
## 6 -1.2560694 0.49849715 0.5642370  
## ManufacturingProcess07 ManufacturingProcess08 ManufacturingProcess09  
## 1 -0.4256500 -0.5761169 -1.7201524  
## 2 -0.9578363 0.8991659 0.5883746  
## 3 1.0427233 0.8991659 -0.3815947  
## 4 -0.9578363 -1.1108074 -0.4785917  
## 5 1.0427233 0.8991659 -0.4527258  
## 6 1.0427233 0.8991659 -0.2199332  
## ManufacturingProcess10 ManufacturingProcess11 ManufacturingProcess12  
## 1 -0.29888818 -0.3232328 -0.4790178  
## 2 0.48941917 1.1432407 -0.4790178  
## 3 0.19393012 0.2640909 -0.4790178  
## 4 0.09437624 1.0545956 -0.4790178  
## 5 -0.37534711 0.5336292 -0.4790178  
## 6 -0.32443045 0.5765004 -0.4790178  
## ManufacturingProcess13 ManufacturingProcess14 ManufacturingProcess15  
## 1 0.97711512 0.8141279 1.1846438  
## 2 -0.50030980 0.2811695 0.9617071  
## 3 0.28765016 0.4465704 0.8245152  
## 4 0.28765016 0.7957500 1.0817499  
## 5 0.09066017 2.5416480 3.3282665  
## 6 -0.50030980 2.4130029 3.1396277  
## ManufacturingProcess16 ManufacturingProcess17 ManufacturingProcess18  
## 1 0.3303945 0.9263296 0.1505348  
## 2 0.1455765 -0.2753953 0.1559773  
## 3 0.1455765 0.3655246 0.1831898  
## 4 0.1967569 0.3655246 0.1695836  
## 5 0.4754056 -0.3555103 0.2076811  
## 6 0.6261033 -0.7560852 0.1423710  
## ManufacturingProcess19 ManufacturingProcess20 ManufacturingProcess21  
## 1 0.4563798 0.3109942 0.2109804  
## 2 1.5095063 0.1849230 0.2109804  
## 3 1.0926437 0.1849230 0.2109804  
## 4 0.9829430 0.1562704 0.2109804  
## 5 1.6192070 0.2938027 -0.6884239  
## 6 1.9044287 0.3998171 -0.5599376  
## ManufacturingProcess22 ManufacturingProcess23 ManufacturingProcess24  
## 1 0.3855865 0.7815967 0.4717009  
## 2 -0.7262673 -1.8212444 -1.0109484  
## 3 -0.4252906 -1.2190925 -0.8381332  
## 4 -0.1243139 -0.6169407 -0.6653181  
## 5 0.7786162 0.5873630 1.5812789  
## 6 1.0795929 -1.2190925 -1.3565787  
## ManufacturingProcess25 ManufacturingProcess26 ManufacturingProcess27  
## 1 0.1216354 0.1274268 0.3500191  
## 2 0.1107691 0.1994510 0.1923811  
## 3 0.1868337 0.2190939 0.2124441  
## 4 0.1732507 0.2081812 0.1923811  
## 5 0.2764812 0.2954832 0.3471529  
## 6 0.1162023 0.2452845 0.3557513  
## ManufacturingProcess28 ManufacturingProcess29 ManufacturingProcess30  
## 1 0.8103308 0.6071537 0.7568194  
## 2 0.9052417 0.8509759 0.7568194  
## 3 0.8862595 0.7900203 0.2379678  
## 4 0.8862595 0.7900203 0.2379678  
## 5 0.9242238 0.9728869 -0.1771134  
## 6 0.9432060 1.0338425 0.9643601  
## ManufacturingProcess31 ManufacturingProcess32 ManufacturingProcess33  
## 1 -0.2010964 -0.4568829 0.9999809  
## 2 -0.2741327 1.9517531 0.9999809  
## 3 -0.1645783 2.6928719 0.9999809  
## 4 -0.1645783 2.3223125 1.8158588  
## 5 -0.1463192 2.3223125 2.6317367  
## 6 -0.3654280 2.6928719 2.6317367  
## ManufacturingProcess34 ManufacturingProcess35 ManufacturingProcess36  
## 1 -1.7011967 -0.87382216 -0.6515638  
## 2 1.9880597 1.17282515 -0.6515638  
## 3 1.9880597 1.26585457 -1.8089817  
## 4 0.1434315 0.05647207 -1.8089817  
## 5 0.1434315 -2.54835178 -2.9663997  
## 6 0.1434315 -0.50170447 -1.8089817  
## ManufacturingProcess37 ManufacturingProcess38 ManufacturingProcess39  
## 1 -1.1540243 0.7174727 0.2317270  
## 2 2.2161351 -0.8224687 0.2317270  
## 3 -0.7046697 -0.8224687 0.2317270  
## 4 0.4187168 -0.8224687 0.2317270  
## 5 -1.8280562 -0.8224687 0.2981503  
## 6 -1.3787016 -0.8224687 0.2317270  
## ManufacturingProcess40 ManufacturingProcess41 ManufacturingProcess42  
## 1 -0.4610622 -0.4390981 0.20279570  
## 2 2.1565813 2.3542004 -0.05472265  
## 3 -0.4610622 -0.4390981 0.40881037  
## 4 -0.4610622 -0.4390981 -0.31224099  
## 5 -0.4610622 -0.4390981 -0.10622632  
## 6 -0.4610622 -0.4390981 0.15129203  
## ManufacturingProcess43 ManufacturingProcess44  
## 1 2.40564734 -0.01588055  
## 2 -0.01374656 0.29467248  
## 3 0.10146268 -0.01588055  
## 4 0.21667191 -0.01588055  
## 5 0.21667191 -0.32643359  
## 6 1.48397347 -0.01588055

We will be using RMSE from the test set performance to determine the most optimal tree-based regression model.

A. Which tree-based regression model gives the optimal resampling and test set performance?

We are using here several tree-based regression models single tree,bagged trees,random forest,boosted trees and cubist. Given that we are comparing different trees, we will be utilizing the RMSE values from the test set performance to choose the most optimal model.

Single Tree

set.seed(1)  
rpartTune <- train(X\_train, Y\_train, method = "rpart2",  
 tuneLength = 10,  
 trControl = trainControl(method = "cv"))  
rpartTune

## CART   
##   
## 144 samples  
## 56 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 128, 130, 130, 131, 129, 130, ...   
## Resampling results across tuning parameters:  
##   
## maxdepth RMSE Rsquared MAE   
## 1 1.308352 0.5130147 1.0569095  
## 2 1.352595 0.5029501 1.1032161  
## 3 1.330961 0.5208352 1.0847034  
## 4 1.295385 0.5340336 1.0397327  
## 5 1.310353 0.5120659 1.0278080  
## 6 1.324231 0.4993690 1.0023312  
## 7 1.321880 0.5002141 0.9965908  
## 8 1.314831 0.5017745 1.0038798  
## 9 1.315171 0.5110080 0.9942259  
## 10 1.287015 0.5330928 0.9692818  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was maxdepth = 10.

y\_pred <- predict(rpartTune, X\_test)  
RMSE(y\_pred, Y\_test)

## [1] 1.746993

Bagged Trees

set.seed(1)  
train\_df <- cbind(X\_train, Y\_train)  
baggedTree <- bagging(Y\_train ~ ., data = train\_df)  
baggedTree

##   
## Bagging regression trees with 25 bootstrap replications   
##   
## Call: bagging.data.frame(formula = Y\_train ~ ., data = train\_df)

summary(baggedTree)

## Length Class Mode   
## y 144 -none- numeric  
## X 56 data.frame list   
## mtrees 25 -none- list   
## OOB 1 -none- logical  
## comb 1 -none- logical  
## call 3 -none- call

# Checking RMSE for bagged tree based model  
y\_pred <- predict(baggedTree, X\_test)  
RMSE(y\_pred, Y\_test)

## [1] 1.574695

Random Forest

rfModel <- randomForest(X\_train, Y\_train,  
 importance = TRUE,  
 ntress = 1000)  
rfModel

##   
## Call:  
## randomForest(x = X\_train, y = Y\_train, importance = TRUE, ntress = 1000)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 18  
##   
## Mean of squared residuals: 1.326004  
## % Var explained: 60.6

# Checking RMSE for random forest based model  
y\_pred <- predict(rfModel, X\_test)  
RMSE(y\_pred, Y\_test)

## [1] 1.229119

Boosted Trees

gbmModel <- gbm.fit(X\_train, Y\_train, distribution = "gaussian")

## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 3.3618 -nan 0.0010 0.0031  
## 2 3.3588 -nan 0.0010 0.0032  
## 3 3.3558 -nan 0.0010 0.0032  
## 4 3.3526 -nan 0.0010 0.0031  
## 5 3.3499 -nan 0.0010 0.0030  
## 6 3.3470 -nan 0.0010 0.0032  
## 7 3.3436 -nan 0.0010 0.0029  
## 8 3.3410 -nan 0.0010 0.0024  
## 9 3.3376 -nan 0.0010 0.0031  
## 10 3.3343 -nan 0.0010 0.0031  
## 20 3.3054 -nan 0.0010 0.0031  
## 40 3.2471 -nan 0.0010 0.0029  
## 60 3.1888 -nan 0.0010 0.0029  
## 80 3.1324 -nan 0.0010 0.0023  
## 100 3.0824 -nan 0.0010 0.0025

gbmModel

## A gradient boosted model with gaussian loss function.  
## 100 iterations were performed.  
## There were 56 predictors of which 4 had non-zero influence.

# Checking RMSE for boosted tree based model  
y\_pred <- predict(gbmModel, X\_test, n.trees=100)  
RMSE(y\_pred, Y\_test)

## [1] 1.82493

Cubist

cubistMod <- cubist(X\_train, Y\_train)  
cubistMod

##   
## Call:  
## cubist.default(x = X\_train, y = Y\_train)  
##   
## Number of samples: 144   
## Number of predictors: 56   
##   
## Number of committees: 1   
## Number of rules: 1

y\_pred <- predict(cubistMod, X\_test)  
RMSE(y\_pred, Y\_test)

## [1] 2.043035

It appears that the random forest had the smallest RMSE, so we will choose the random forest as the most optimal model.

B. Which predictors are most important in the optimal tree-based regression model? Do either the biological or process variables dominate the list? How do the top 10 important predictors compare to the top 10 predictors from the optimal linear and nonlinear models?

We will be using the random forest model.

a <- varImp(rfModel)  
a$Variables <- rownames(a)  
rownames(a) <- 1:nrow(a)  
head(a[order(a$Overall, decreasing = TRUE),],10)

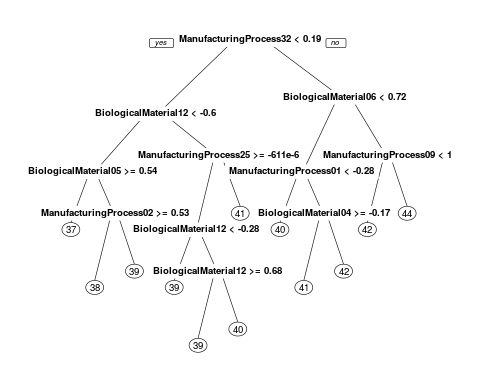
## Overall Variables  
## 44 25.356936 ManufacturingProcess32  
## 6 10.925491 BiologicalMaterial06  
## 2 7.781620 BiologicalMaterial02  
## 29 7.706733 ManufacturingProcess17  
## 12 7.685157 BiologicalMaterial12  
## 25 7.661669 ManufacturingProcess13  
## 43 7.596638 ManufacturingProcess31  
## 48 7.495015 ManufacturingProcess36  
## 3 7.480981 BiologicalMaterial03  
## 21 6.850197 ManufacturingProcess09

There are 6 Biological Material and 6 Manufacturing Processes in the top 10 and ManufacturingProcess32 and BiologicalMaterial06 were on the top.

C. Plot the optimal single tree with the distribution of yield in the terminal nodes. Does this view of the data provide additional knowledge about the biological or process predictors and their relationship with yield?

rpartTree <- rpart(Y\_train ~ ., data = train\_df)  
#summary(rpartTree)

prp(rpartTree)

 The single tree seems to confirm the importance of ManufacturingProcess32 which then breaks down into different branches incluindg BiologicalMaterial12 and BiologicalMaterial06. The Biological materials and manufacturing processes are mixed in the top 10 for variable importance.we can say that manufacturer processes have a larger impacting role towards yield then biological factors but Biological materials are the initial nodes and are also influential.