HW7

Team 4

25/10/2019

### Excersise 6.3

A chemical manufacturing process for a pharmaceutical product was discussed in Sect.1.4. In this problem, the objective is to understand the relationship between biological measurements of the raw materials (predictors), measurements of the manufacturing process (predictors), and the response of product yield. Biological predictors cannot be changed but can be used to assess the quality of the raw material before processing. On the other hand, manufacturing process predictors can be changed in the manufacturing process. Improving product yield by 1% will boost revenue by approximately one hundred thousand dollars per batch:

1. Start R and use these commands to load the data:

library(AppliedPredictiveModeling)

data(chemicalManufacturing)

The matrix processPredictors contains the 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.

1. A small percentage of cells in the predictor set contain missing values. Use an imputation function to fill in these missing values (e.g., see Sect. 3.8).
2. 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?
3. 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?
4. Which predictors are most important in the model you have trained? Do either the biological or process predictors dominate the list?
5. 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?

### Approach

Let’s explore the data set. It contains 57 predictors, 1 response variable (yield) and 176 samples (manufacturing runs). Data set contains 1% missing values. These missing values will be replaced using knn impute method during pre-processing step. Near-zero variance predictors will be eliminated using “nzv”. Response variable looks nearly normally distributed, predictors variables will be centered and scalled.

The model of our choice will be PLS model. Partial Least Squares (PLS) regression identifies new principal components that not only summarizes the original predictors, but also that are related to the outcome. These components are then used to fit the regression model. So, compared to PCR, PLS uses a dimension reduction strategy that is supervised by the outcome. PLS is convenient for data with highly-correlated predictors. 10 fold cross validation will be used to assess the accuracy of each model on training set and then accuracy will be re-assessed on the test set.

library(vip)  
library(RANN)  
library(AppliedPredictiveModeling)  
library(dplyr)  
library(tidyr)  
library(e1071)  
library(MASS)  
library(caret)  
library(corrplot)  
library(ggplot2)

Data Exploration

ls(pattern = "^ChemicalManufacturing")

## character(0)

data(ChemicalManufacturingProcess)  
  
# checking data structure  
str(ChemicalManufacturingProcess)

## 'data.frame': 176 obs. of 58 variables:  
## $ Yield : num 38 42.4 42 41.4 42.5 ...  
## $ BiologicalMaterial01 : num 6.25 8.01 8.01 8.01 7.47 6.12 7.48 6.94 6.94 6.94 ...  
## $ BiologicalMaterial02 : num 49.6 61 61 61 63.3 ...  
## $ BiologicalMaterial03 : num 57 67.5 67.5 67.5 72.2 ...  
## $ BiologicalMaterial04 : num 12.7 14.7 14.7 14.7 14 ...  
## $ BiologicalMaterial05 : num 19.5 19.4 19.4 19.4 17.9 ...  
## $ BiologicalMaterial06 : num 43.7 53.1 53.1 53.1 54.7 ...  
## $ BiologicalMaterial07 : num 100 100 100 100 100 100 100 100 100 100 ...  
## $ BiologicalMaterial08 : num 16.7 19 19 19 18.2 ...  
## $ BiologicalMaterial09 : num 11.4 12.6 12.6 12.6 12.8 ...  
## $ BiologicalMaterial10 : num 3.46 3.46 3.46 3.46 3.05 3.78 3.04 3.85 3.85 3.85 ...  
## $ BiologicalMaterial11 : num 138 154 154 154 148 ...  
## $ BiologicalMaterial12 : num 18.8 21.1 21.1 21.1 21.1 ...  
## $ ManufacturingProcess01: num NA 0 0 0 10.7 12 11.5 12 12 12 ...  
## $ ManufacturingProcess02: num NA 0 0 0 0 0 0 0 0 0 ...  
## $ ManufacturingProcess03: num NA NA NA NA NA NA 1.56 1.55 1.56 1.55 ...  
## $ ManufacturingProcess04: num NA 917 912 911 918 924 933 929 928 938 ...  
## $ ManufacturingProcess05: num NA 1032 1004 1015 1028 ...  
## $ ManufacturingProcess06: num NA 210 207 213 206 ...  
## $ ManufacturingProcess07: num NA 177 178 177 178 178 177 178 177 177 ...  
## $ ManufacturingProcess08: num NA 178 178 177 178 178 178 178 177 177 ...  
## $ ManufacturingProcess09: num 43 46.6 45.1 44.9 45 ...  
## $ ManufacturingProcess10: num NA NA NA NA NA NA 11.6 10.2 9.7 10.1 ...  
## $ ManufacturingProcess11: num NA NA NA NA NA NA 11.5 11.3 11.1 10.2 ...  
## $ ManufacturingProcess12: num NA 0 0 0 0 0 0 0 0 0 ...  
## $ ManufacturingProcess13: num 35.5 34 34.8 34.8 34.6 34 32.4 33.6 33.9 34.3 ...  
## $ ManufacturingProcess14: num 4898 4869 4878 4897 4992 ...  
## $ ManufacturingProcess15: num 6108 6095 6087 6102 6233 ...  
## $ ManufacturingProcess16: num 4682 4617 4617 4635 4733 ...  
## $ ManufacturingProcess17: num 35.5 34 34.8 34.8 33.9 33.4 33.8 33.6 33.9 35.3 ...  
## $ ManufacturingProcess18: num 4865 4867 4877 4872 4886 ...  
## $ ManufacturingProcess19: num 6049 6097 6078 6073 6102 ...  
## $ ManufacturingProcess20: num 4665 4621 4621 4611 4659 ...  
## $ ManufacturingProcess21: num 0 0 0 0 -0.7 -0.6 1.4 0 0 1 ...  
## $ ManufacturingProcess22: num NA 3 4 5 8 9 1 2 3 4 ...  
## $ ManufacturingProcess23: num NA 0 1 2 4 1 1 2 3 1 ...  
## $ ManufacturingProcess24: num NA 3 4 5 18 1 1 2 3 4 ...  
## $ ManufacturingProcess25: num 4873 4869 4897 4892 4930 ...  
## $ ManufacturingProcess26: num 6074 6107 6116 6111 6151 ...  
## $ ManufacturingProcess27: num 4685 4630 4637 4630 4684 ...  
## $ ManufacturingProcess28: num 10.7 11.2 11.1 11.1 11.3 11.4 11.2 11.1 11.3 11.4 ...  
## $ ManufacturingProcess29: num 21 21.4 21.3 21.3 21.6 21.7 21.2 21.2 21.5 21.7 ...  
## $ ManufacturingProcess30: num 9.9 9.9 9.4 9.4 9 10.1 11.2 10.9 10.5 9.8 ...  
## $ ManufacturingProcess31: num 69.1 68.7 69.3 69.3 69.4 68.2 67.6 67.9 68 68.5 ...  
## $ ManufacturingProcess32: num 156 169 173 171 171 173 159 161 160 164 ...  
## $ ManufacturingProcess33: num 66 66 66 68 70 70 65 65 65 66 ...  
## $ ManufacturingProcess34: num 2.4 2.6 2.6 2.5 2.5 2.5 2.5 2.5 2.5 2.5 ...  
## $ ManufacturingProcess35: num 486 508 509 496 468 490 475 478 491 488 ...  
## $ ManufacturingProcess36: num 0.019 0.019 0.018 0.018 0.017 0.018 0.019 0.019 0.019 0.019 ...  
## $ ManufacturingProcess37: num 0.5 2 0.7 1.2 0.2 0.4 0.8 1 1.2 1.8 ...  
## $ ManufacturingProcess38: num 3 2 2 2 2 2 2 2 3 3 ...  
## $ ManufacturingProcess39: num 7.2 7.2 7.2 7.2 7.3 7.2 7.3 7.3 7.4 7.1 ...  
## $ ManufacturingProcess40: num NA 0.1 0 0 0 0 0 0 0 0 ...  
## $ ManufacturingProcess41: num NA 0.15 0 0 0 0 0 0 0 0 ...  
## $ ManufacturingProcess42: num 11.6 11.1 12 10.6 11 11.5 11.7 11.4 11.4 11.3 ...  
## $ ManufacturingProcess43: num 3 0.9 1 1.1 1.1 2.2 0.7 0.8 0.9 0.8 ...  
## $ ManufacturingProcess44: num 1.8 1.9 1.8 1.8 1.7 1.8 2 2 1.9 1.9 ...  
## $ ManufacturingProcess45: num 2.4 2.2 2.3 2.1 2.1 2 2.2 2.2 2.1 2.4 ...

# checking data dimentions  
dim(ChemicalManufacturingProcess)

## [1] 176 58

Checking missing values

# total % of missing values in a data frame  
(sum(is.na(ChemicalManufacturingProcess))/(176\*58))\*100

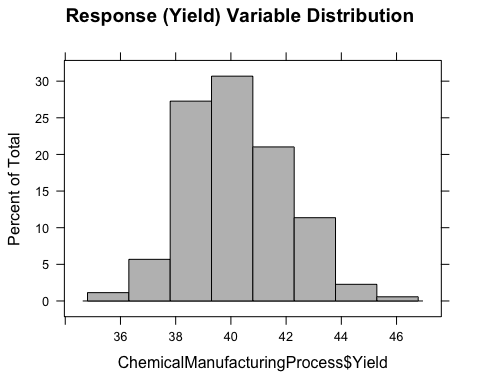
## [1] 1.038401

# missing values distribution by variables  
is\_na <- sort(colSums(is.na(ChemicalManufacturingProcess)), decreasing = TRUE)  
is\_na[is\_na > 0]

## ManufacturingProcess03 ManufacturingProcess11 ManufacturingProcess10   
## 15 10 9   
## ManufacturingProcess25 ManufacturingProcess26 ManufacturingProcess27   
## 5 5 5   
## ManufacturingProcess28 ManufacturingProcess29 ManufacturingProcess30   
## 5 5 5   
## ManufacturingProcess31 ManufacturingProcess33 ManufacturingProcess34   
## 5 5 5   
## ManufacturingProcess35 ManufacturingProcess36 ManufacturingProcess02   
## 5 5 3   
## ManufacturingProcess06 ManufacturingProcess01 ManufacturingProcess04   
## 2 1 1   
## ManufacturingProcess05 ManufacturingProcess07 ManufacturingProcess08   
## 1 1 1   
## ManufacturingProcess12 ManufacturingProcess14 ManufacturingProcess22   
## 1 1 1   
## ManufacturingProcess23 ManufacturingProcess24 ManufacturingProcess40   
## 1 1 1   
## ManufacturingProcess41   
## 1

Checking response variable distribution

histogram(ChemicalManufacturingProcess$Yield, col="grey", main = "Response (Yield) Variable Distribution")



Pre-process the data as described above

pp <- preProcess(ChemicalManufacturingProcess, method = c("center","scale", "knnImpute", "nzv"))  
# summary of the variables transformations  
pp$method

## $center  
## [1] "Yield" "BiologicalMaterial01"   
## [3] "BiologicalMaterial02" "BiologicalMaterial03"   
## [5] "BiologicalMaterial04" "BiologicalMaterial05"   
## [7] "BiologicalMaterial06" "BiologicalMaterial08"   
## [9] "BiologicalMaterial09" "BiologicalMaterial10"   
## [11] "BiologicalMaterial11" "BiologicalMaterial12"   
## [13] "ManufacturingProcess01" "ManufacturingProcess02"  
## [15] "ManufacturingProcess03" "ManufacturingProcess04"  
## [17] "ManufacturingProcess05" "ManufacturingProcess06"  
## [19] "ManufacturingProcess07" "ManufacturingProcess08"  
## [21] "ManufacturingProcess09" "ManufacturingProcess10"  
## [23] "ManufacturingProcess11" "ManufacturingProcess12"  
## [25] "ManufacturingProcess13" "ManufacturingProcess14"  
## [27] "ManufacturingProcess15" "ManufacturingProcess16"  
## [29] "ManufacturingProcess17" "ManufacturingProcess18"  
## [31] "ManufacturingProcess19" "ManufacturingProcess20"  
## [33] "ManufacturingProcess21" "ManufacturingProcess22"  
## [35] "ManufacturingProcess23" "ManufacturingProcess24"  
## [37] "ManufacturingProcess25" "ManufacturingProcess26"  
## [39] "ManufacturingProcess27" "ManufacturingProcess28"  
## [41] "ManufacturingProcess29" "ManufacturingProcess30"  
## [43] "ManufacturingProcess31" "ManufacturingProcess32"  
## [45] "ManufacturingProcess33" "ManufacturingProcess34"  
## [47] "ManufacturingProcess35" "ManufacturingProcess36"  
## [49] "ManufacturingProcess37" "ManufacturingProcess38"  
## [51] "ManufacturingProcess39" "ManufacturingProcess40"  
## [53] "ManufacturingProcess41" "ManufacturingProcess42"  
## [55] "ManufacturingProcess43" "ManufacturingProcess44"  
## [57] "ManufacturingProcess45"  
##   
## $scale  
## [1] "Yield" "BiologicalMaterial01"   
## [3] "BiologicalMaterial02" "BiologicalMaterial03"   
## [5] "BiologicalMaterial04" "BiologicalMaterial05"   
## [7] "BiologicalMaterial06" "BiologicalMaterial08"   
## [9] "BiologicalMaterial09" "BiologicalMaterial10"   
## [11] "BiologicalMaterial11" "BiologicalMaterial12"   
## [13] "ManufacturingProcess01" "ManufacturingProcess02"  
## [15] "ManufacturingProcess03" "ManufacturingProcess04"  
## [17] "ManufacturingProcess05" "ManufacturingProcess06"  
## [19] "ManufacturingProcess07" "ManufacturingProcess08"  
## [21] "ManufacturingProcess09" "ManufacturingProcess10"  
## [23] "ManufacturingProcess11" "ManufacturingProcess12"  
## [25] "ManufacturingProcess13" "ManufacturingProcess14"  
## [27] "ManufacturingProcess15" "ManufacturingProcess16"  
## [29] "ManufacturingProcess17" "ManufacturingProcess18"  
## [31] "ManufacturingProcess19" "ManufacturingProcess20"  
## [33] "ManufacturingProcess21" "ManufacturingProcess22"  
## [35] "ManufacturingProcess23" "ManufacturingProcess24"  
## [37] "ManufacturingProcess25" "ManufacturingProcess26"  
## [39] "ManufacturingProcess27" "ManufacturingProcess28"  
## [41] "ManufacturingProcess29" "ManufacturingProcess30"  
## [43] "ManufacturingProcess31" "ManufacturingProcess32"  
## [45] "ManufacturingProcess33" "ManufacturingProcess34"  
## [47] "ManufacturingProcess35" "ManufacturingProcess36"  
## [49] "ManufacturingProcess37" "ManufacturingProcess38"  
## [51] "ManufacturingProcess39" "ManufacturingProcess40"  
## [53] "ManufacturingProcess41" "ManufacturingProcess42"  
## [55] "ManufacturingProcess43" "ManufacturingProcess44"  
## [57] "ManufacturingProcess45"  
##   
## $knnImpute  
## [1] "Yield" "BiologicalMaterial01"   
## [3] "BiologicalMaterial02" "BiologicalMaterial03"   
## [5] "BiologicalMaterial04" "BiologicalMaterial05"   
## [7] "BiologicalMaterial06" "BiologicalMaterial08"   
## [9] "BiologicalMaterial09" "BiologicalMaterial10"   
## [11] "BiologicalMaterial11" "BiologicalMaterial12"   
## [13] "ManufacturingProcess01" "ManufacturingProcess02"  
## [15] "ManufacturingProcess03" "ManufacturingProcess04"  
## [17] "ManufacturingProcess05" "ManufacturingProcess06"  
## [19] "ManufacturingProcess07" "ManufacturingProcess08"  
## [21] "ManufacturingProcess09" "ManufacturingProcess10"  
## [23] "ManufacturingProcess11" "ManufacturingProcess12"  
## [25] "ManufacturingProcess13" "ManufacturingProcess14"  
## [27] "ManufacturingProcess15" "ManufacturingProcess16"  
## [29] "ManufacturingProcess17" "ManufacturingProcess18"  
## [31] "ManufacturingProcess19" "ManufacturingProcess20"  
## [33] "ManufacturingProcess21" "ManufacturingProcess22"  
## [35] "ManufacturingProcess23" "ManufacturingProcess24"  
## [37] "ManufacturingProcess25" "ManufacturingProcess26"  
## [39] "ManufacturingProcess27" "ManufacturingProcess28"  
## [41] "ManufacturingProcess29" "ManufacturingProcess30"  
## [43] "ManufacturingProcess31" "ManufacturingProcess32"  
## [45] "ManufacturingProcess33" "ManufacturingProcess34"  
## [47] "ManufacturingProcess35" "ManufacturingProcess36"  
## [49] "ManufacturingProcess37" "ManufacturingProcess38"  
## [51] "ManufacturingProcess39" "ManufacturingProcess40"  
## [53] "ManufacturingProcess41" "ManufacturingProcess42"  
## [55] "ManufacturingProcess43" "ManufacturingProcess44"  
## [57] "ManufacturingProcess45"  
##   
## $ignore  
## character(0)  
##   
## $remove  
## [1] "BiologicalMaterial07"

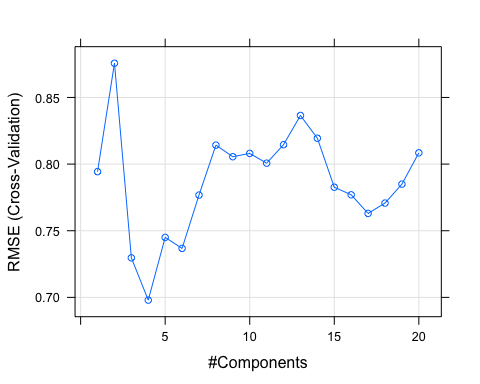
# application of the transformations on the training set  
ChemicalManufacturingProcess<- predict(pp, ChemicalManufacturingProcess)

Splitting data on train and test

set.seed(123)  
training.samples <- ChemicalManufacturingProcess$Yield %>%  
createDataPartition(p = 0.8, list = FALSE)  
train.data <- ChemicalManufacturingProcess[training.samples, ]  
test.data <- ChemicalManufacturingProcess[-training.samples, ]

Building PLS model

model <- train(Yield~., data = train.data, method = "pls", scale = TRUE, trControl = trainControl("cv", number = 10), tuneLength = 20)  
# Plot model RMSE vs different values of components  
plot(model)



# best tuning parameters  
model$bestTune

## ncomp  
## 4 4

# model summary  
model

## Partial Least Squares   
##   
## 144 samples  
## 56 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 130, 128, 129, 130, 130, 129, ...   
## Resampling results across tuning parameters:  
##   
## ncomp RMSE Rsquared MAE   
## 1 0.7943255 0.4304338 0.6277772  
## 2 0.8756506 0.4918526 0.6179463  
## 3 0.7296670 0.5804953 0.5710554  
## 4 0.6979392 0.5881865 0.5570296  
## 5 0.7450023 0.5838615 0.5673552  
## 6 0.7367736 0.5788768 0.5728364  
## 7 0.7767460 0.5730017 0.5933019  
## 8 0.8142302 0.5628303 0.6145876  
## 9 0.8055336 0.5632577 0.6137303  
## 10 0.8080164 0.5582931 0.6126512  
## 11 0.8006887 0.5447062 0.6150225  
## 12 0.8145774 0.5330077 0.6212473  
## 13 0.8364713 0.5219486 0.6265942  
## 14 0.8194040 0.5248474 0.6235994  
## 15 0.7826547 0.5319358 0.6155874  
## 16 0.7769589 0.5275952 0.6217911  
## 17 0.7629862 0.5323295 0.6170801  
## 18 0.7707632 0.5289435 0.6215907  
## 19 0.7849680 0.5209794 0.6284032  
## 20 0.8084534 0.5136889 0.6377438  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was ncomp = 4.

# final model summary  
summary(model$finalModel)

## Data: X dimension: 144 56   
## Y dimension: 144 1  
## Fit method: oscorespls  
## Number of components considered: 4  
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps  
## X 19.36 30.06 37.16 44.20  
## .outcome 48.99 62.72 70.26 72.87

# making predictions on the test set  
predictions <- model %>% predict(test.data)  
  
# model performance metrics on test set  
data.frame(  
 RMSE = caret::RMSE(predictions, test.data$Yield),  
 Rsquare = caret::R2(predictions, test.data$Yield)  
)

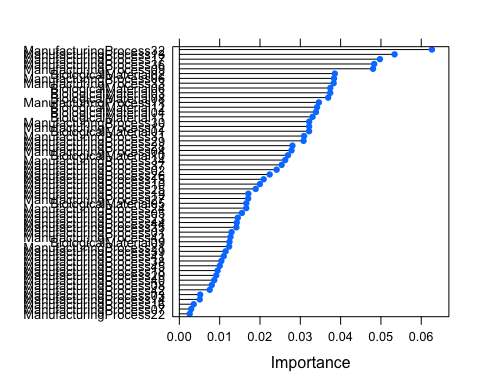
## RMSE Rsquare  
## 1 1.326151 0.1806916

The optimal number of principal components included in the PLS model is 4. This captures 44% of the variation in the predictors and 72% of the variation in the outcome variable (Yield).

RMSE is higher on the test set compare to the performance on the training set. This is expected as model usually performs better on the set which it is trained on. At the same it can be a sign of overfitting.

Finding the most important variables.

importance <- varImp(model, scale=FALSE)  
plot(importance)



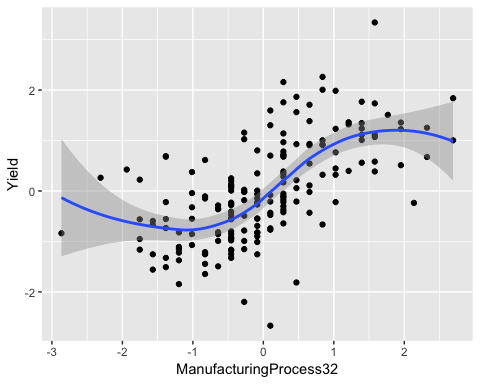
importance

## pls variable importance  
##   
## only 20 most important variables shown (out of 56)  
##   
## Overall  
## ManufacturingProcess32 0.06265  
## ManufacturingProcess13 0.05338  
## ManufacturingProcess17 0.04975  
## ManufacturingProcess36 0.04833  
## ManufacturingProcess09 0.04804  
## BiologicalMaterial02 0.03855  
## ManufacturingProcess06 0.03836  
## ManufacturingProcess33 0.03827  
## BiologicalMaterial06 0.03744  
## BiologicalMaterial03 0.03742  
## BiologicalMaterial08 0.03691  
## ManufacturingProcess11 0.03461  
## BiologicalMaterial12 0.03416  
## BiologicalMaterial04 0.03389  
## BiologicalMaterial11 0.03303  
## ManufacturingProcess30 0.03221  
## ManufacturingProcess12 0.03219  
## BiologicalMaterial01 0.03217  
## ManufacturingProcess31 0.03093  
## ManufacturingProcess29 0.03080

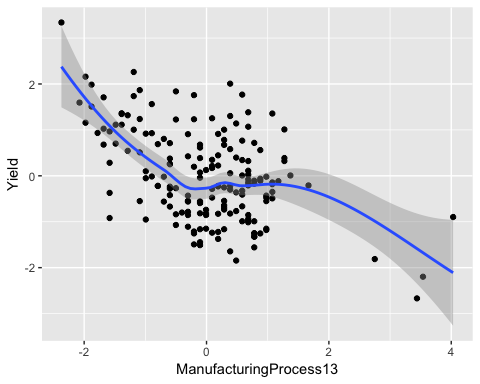
7 out 10 most important variables are Manufacturing Process variables, which indicates the dominant importance of Manufacturing Process over Biological Material for the response variable (Yield).

Let’s look at the relationships of the top 10 most important variables and Yield.

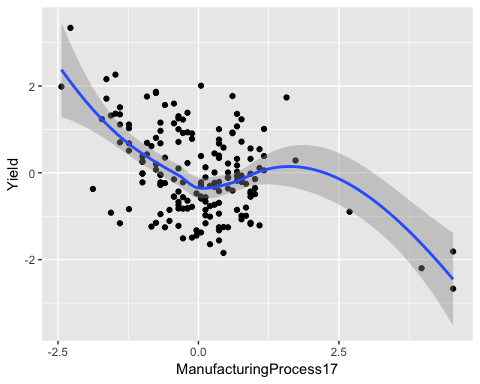
ggplot(ChemicalManufacturingProcess, aes(ManufacturingProcess32,Yield)) + geom\_point() + geom\_smooth()



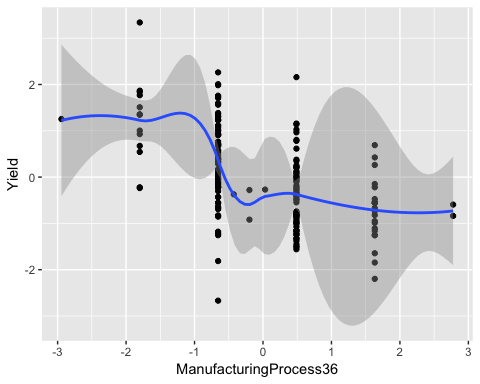
ggplot(ChemicalManufacturingProcess, aes(ManufacturingProcess13,Yield)) + geom\_point() + geom\_smooth()



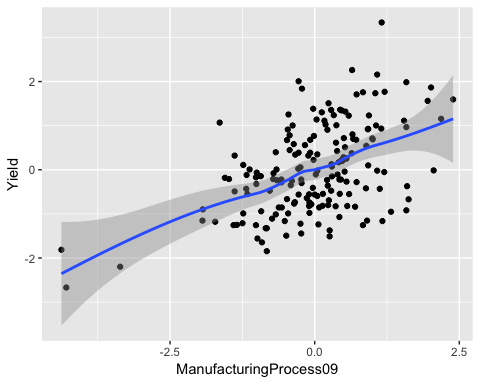
ggplot(ChemicalManufacturingProcess, aes(ManufacturingProcess17,Yield)) + geom\_point() + geom\_smooth()



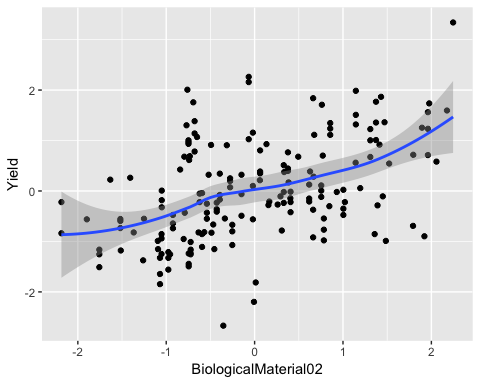
ggplot(ChemicalManufacturingProcess, aes(ManufacturingProcess36,Yield)) + geom\_point() + geom\_smooth()



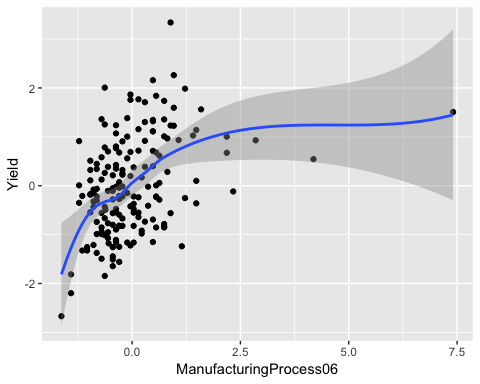
ggplot(ChemicalManufacturingProcess, aes(ManufacturingProcess09,Yield)) + geom\_point() + geom\_smooth()



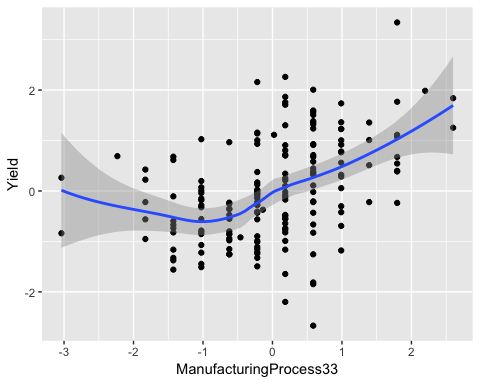
ggplot(ChemicalManufacturingProcess, aes(BiologicalMaterial02,Yield)) + geom\_point() + geom\_smooth()



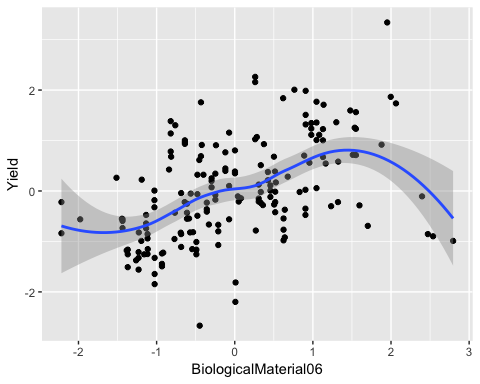
ggplot(ChemicalManufacturingProcess, aes(ManufacturingProcess06,Yield)) + geom\_point() + geom\_smooth()



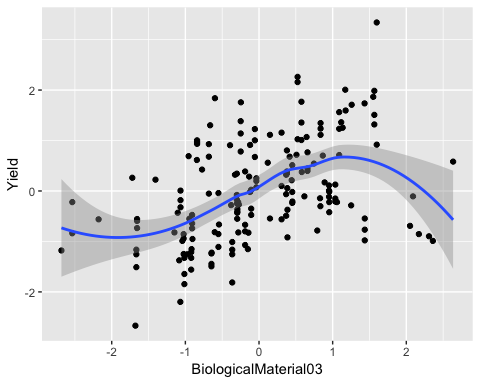
ggplot(ChemicalManufacturingProcess, aes(ManufacturingProcess33,Yield)) + geom\_point() + geom\_smooth()



ggplot(ChemicalManufacturingProcess, aes(BiologicalMaterial06,Yield)) + geom\_point() + geom\_smooth()



ggplot(ChemicalManufacturingProcess, aes(BiologicalMaterial03,Yield)) + geom\_point() + geom\_smooth()



The examination of the relationships of top 10 predictor variables with the response variable allow us to find the following:

ManufacturingProcess32 and ManufacturingProcess09 positively correlate with Yield, whereas ManufacturingProcess17,36, 13 have negative relationships with Yield. That means we can improve yield by having higher (or lower) values of these processes if manufacturing processes can be controlled. Biological materials (02, 06, 03) have positive correlation with the Yield.