Homework #2

Cheryl Bowersox

May 20, 2018

### KH 6.3

Using the manufacturing data set, understand the relationship between biological and manufacturing predictors with the yield levels. Biological cannot be changed but can provide insight into the quality of material used. Manufacturing data can be changed.  
What factors can be adjusted to boost yield and increase revenue?

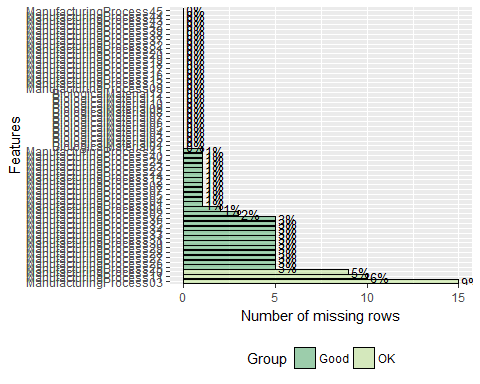
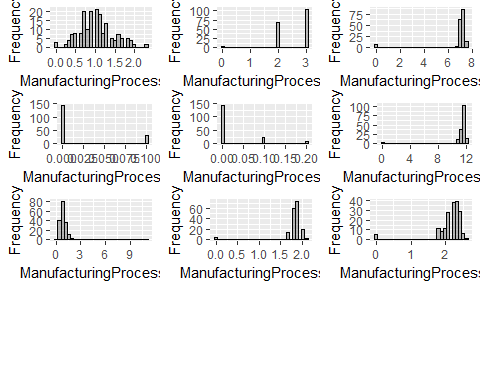
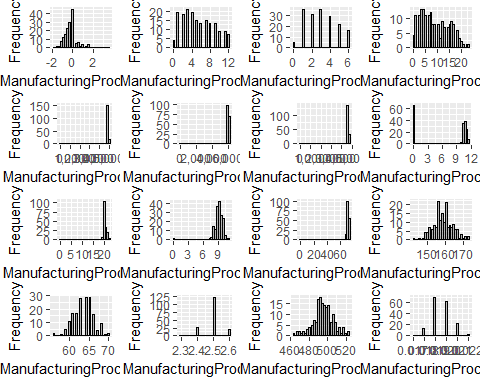
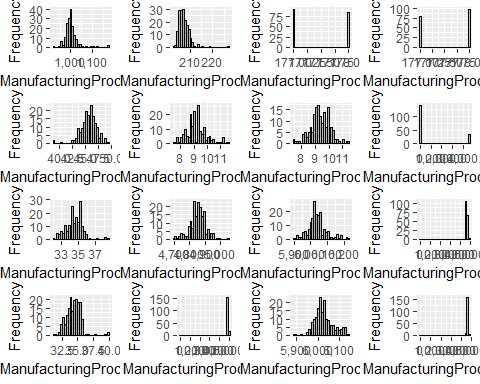
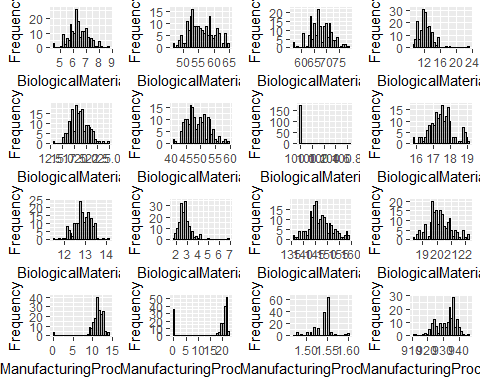
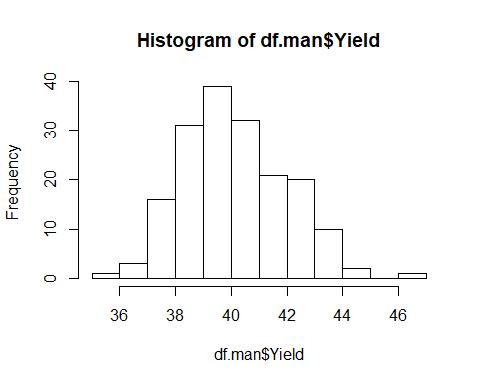
## Exploring Manurfacturing Data

The manufacturing data set includes 58 variables and 176 observations. This data set is numeric, although some variables appear to be binomial may stand for categorical qualities, such as ManufacturingProcess12.

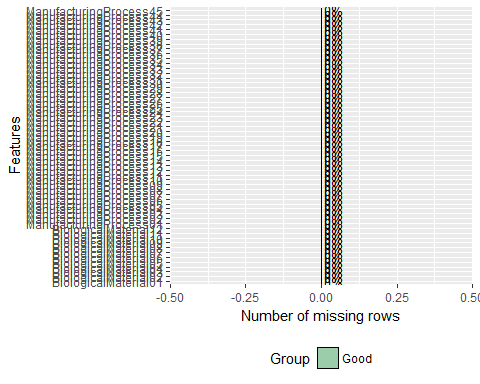
There are 12 biological predictors, and 45 manufacturing data. The data contains missing manufacturing values, with the Manufacturing.Process03 missing the most at 9%.

The data is clean, there are no missing values and no obvious outliers in the data. The biological predictors tend to be normally distributed, while the manufacturing values have some

## 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   
##



**(b)**  
Impute missing values using the preprocess function in the caret package. This uses K-nearest neighbor to estimate the missing values.



**(C) Splitting and Processesing** 80% of the data set(143 observations) was split out to train a model, while the remaining 20% (33 observations) were held back for testing.  
Additional pre- processing was applied to scale and center the data, as well as BoxCox transformations. The BoxCox transformations will help address skewness found in some of the predictor variables.

The transformations, centering, and scaling help normalize the variables.

In addition to transformations, it is important to review the data for co-linearity and check for an variances close to zero. The selected model does no handle near-zero variance well, and variables with this attribute will be dropped from analysis.

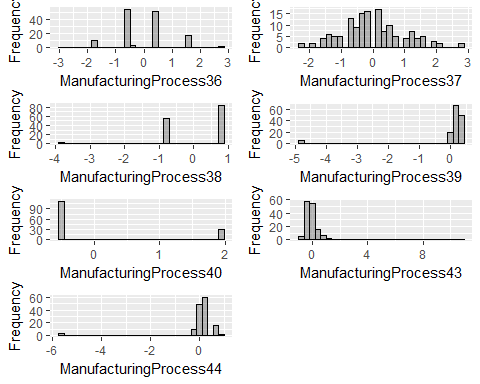
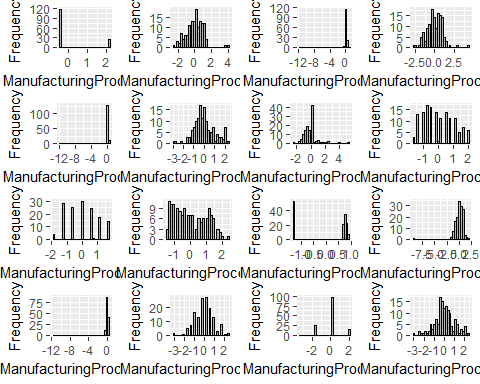
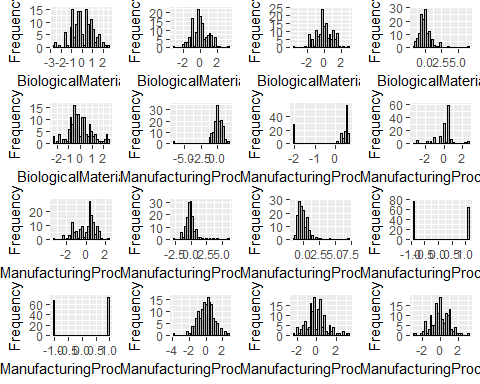
#review for high correlations  
  
dfcorr<- cor(df.man2, use="pairwise.complete" )  
#rows with correlation >.2  
highCorr <- findCorrelation(dfcorr,cutoff = .8, exact = FALSE, names = TRUE)  
highCorr1 <- findCorrelation(dfcorr,cutoff = .8, exact = FALSE, names = FALSE) #return index  
  
df.man3 <- df.man2[,-highCorr1]  
  
vars <- sapply(df.man3, function(x) sd(x))  
  
#near zeros  
  
var.zeros <- nearZeroVar(df.man3, names = TRUE)  
var.zeros1 <- nearZeroVar(df.man3)  
  
df.man4 <- df.man3[,-var.zeros1]  
  
# Splitting DAta  
  
set.seed(42)  
# Training Rows created  
trnRows <- createDataPartition(df.man4[,1],p =.8,list = FALSE, times = 1)  
  
#select training and testing subsets - for this example can use just the first sample created  
trndf.man <- df.man4[trnRows,]  
trndf.Yield <- df.Yield[trnRows,]  
  
testdf.man <- df.man4[-trnRows,]  
testdf.Yield <- df.Yield[-trnRows,]  
  
nrow(trndf.man)

## [1] 143

nrow(testdf.man)

## [1] 33

#PreProcess to scale and apply Box-Cox transformations   
  
proc.man2 <- preProcess(trndf.man,method = c("scale", "center", "BoxCox"))  
#apply to traning and test data  
  
trndf.man2 <- predict(proc.man2, trndf.man)  
testdf.man2 <- predict(proc.man2, testdf.man)  
  
DataExplorer::plot\_histogram(trndf.man2)



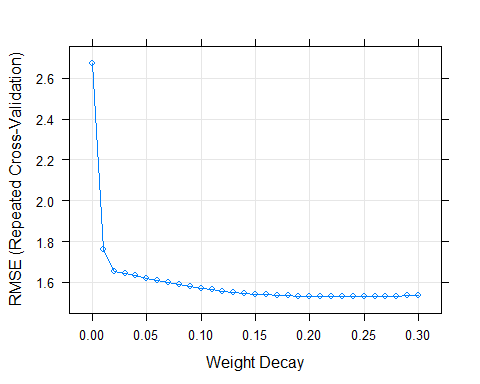
The following 17 variables show a correlation of > .75 with others and are removed from the data set. BiologicalMaterial06, BiologicalMaterial12, ManufacturingProcess14, ManufacturingProcess15, ManufacturingProcess20, ManufacturingProcess26, ManufacturingProcess27, ManufacturingProcess29, ManufacturingProcess41, ManufacturingProcess45, BiologicalMaterial02, BiologicalMaterial01, BiologicalMaterial04, BiologicalMaterial08, ManufacturingProcess25, ManufacturingProcess32, ManufacturingProcess42

A single variable, BiologicalMaterial07 was found with internal variance close to zero. This was also dropped from the data set.

ctrlSamp1 = trainControl(method = "repeatedcv", number = 10)# classProbs = TRUE)  
  
  
  
ridg.grid <- data.frame(.lambda = seq(0, .3,.01))  
  
set.seed(45)  
  
modelridg <- train(x=trndf.man2,y=trndf.Yield,  
 method = "ridge",  
 trControl = ctrlSamp1,  
 tuneGrid = ridg.grid)  
(modelridg)

## Ridge Regression   
##   
## 143 samples  
## 39 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 1 times)   
## Summary of sample sizes: 129, 129, 130, 128, 130, 129, ...   
## Resampling results across tuning parameters:  
##   
## lambda RMSE Rsquared MAE   
## 0.00 2.675517 0.3851392 1.491999  
## 0.01 1.760193 0.4681446 1.206320  
## 0.02 1.653571 0.4872867 1.179107  
## 0.03 1.642127 0.4801953 1.178790  
## 0.04 1.631857 0.4794582 1.175071  
## 0.05 1.620263 0.4814982 1.170568  
## 0.06 1.608543 0.4845886 1.165832  
## 0.07 1.597421 0.4880448 1.161395  
## 0.08 1.587212 0.4915649 1.157208  
## 0.09 1.578015 0.4950051 1.153380  
## 0.10 1.569827 0.4982949 1.150221  
## 0.11 1.562600 0.5014007 1.147342  
## 0.12 1.556267 0.5043087 1.144729  
## 0.13 1.550756 0.5070159 1.142367  
## 0.14 1.545999 0.5095256 1.140237  
## 0.15 1.541930 0.5118447 1.138319  
## 0.16 1.538491 0.5139818 1.136595  
## 0.17 1.535629 0.5159465 1.135333  
## 0.18 1.533295 0.5177488 1.134264  
## 0.19 1.531448 0.5193986 1.133334  
## 0.20 1.530050 0.5209058 1.132531  
## 0.21 1.529065 0.5222799 1.131847  
## 0.22 1.528465 0.5235297 1.131270  
## 0.23 1.528221 0.5246640 1.130793  
## 0.24 1.528308 0.5256908 1.130483  
## 0.25 1.528705 0.5266177 1.130980  
## 0.26 1.529390 0.5274520 1.132754  
## 0.27 1.530345 0.5282003 1.134799  
## 0.28 1.531552 0.5288688 1.136976  
## 0.29 1.532998 0.5294635 1.139183  
## 0.30 1.534666 0.5299898 1.141417  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was lambda = 0.23.

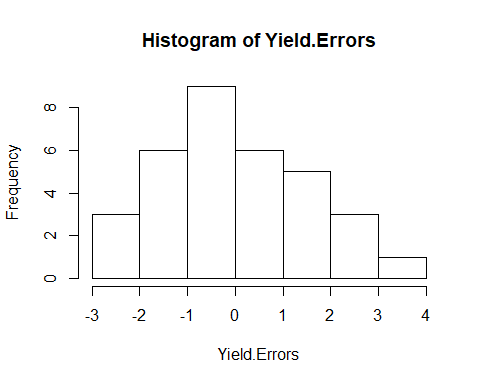
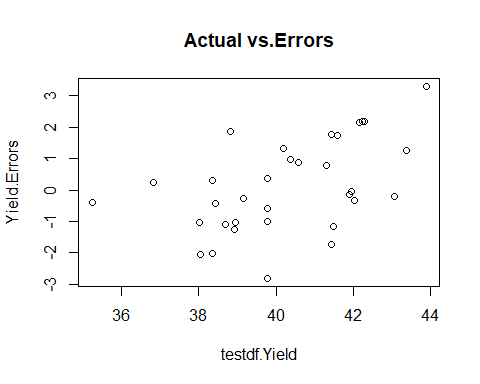
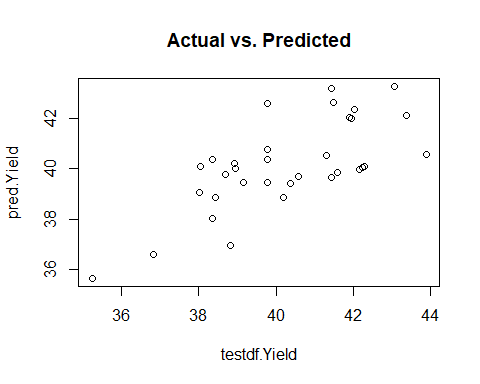
(plot(modelridg))



bl <- modelridg$bestTune[1,1]  
  
mod.rmse <- modelridg$results[modelridg$results$lambda == bl,2]  
mod.rsqr <- modelridg$results[modelridg$results$lambda == bl,3]

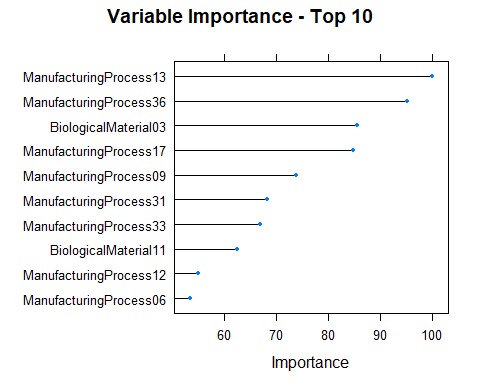
**(D) Ridge Regression Model**  A ridge regression model was trained on the transformed data set, with 10-fold validation.  
The tuning parameter, lambda, was evaluated from .001 to .3, and was found to produce the best MSE and value when lambda = 0.23. The RMSE found for this best model was 1.528221 , with an Rsquared value of 0.524664.

**(D) Prediciton** Predication based on the ridge model with lambda = .1 were made for the Yield using the test data and compared against the actual observed Yield values.  
.

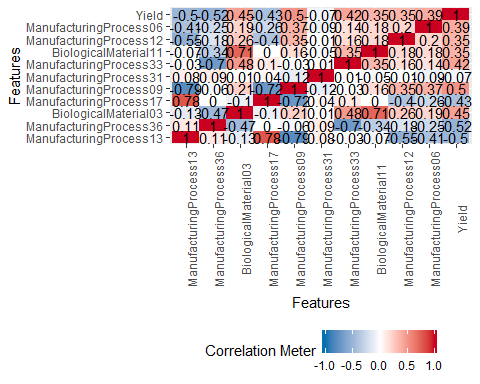
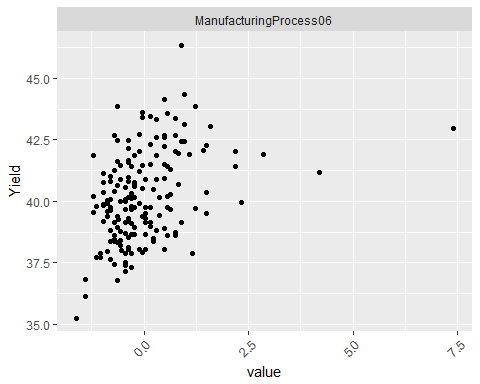
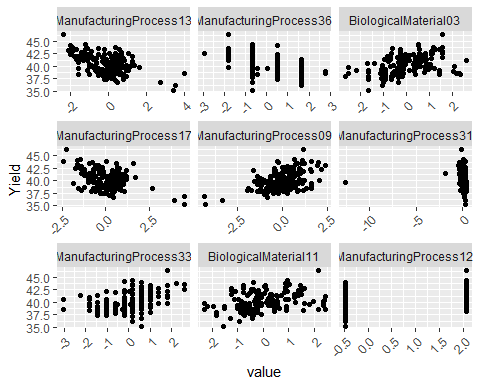


The predictions had a lower RMSE, of 1.4399067, but a much higher Rsquared value of 0.7053064 This may indicate the model well fitted to the actual scenario, and that the higher RMSE during training is caused by outliers or other unusual features included in the training data, but re-sampling avoided over-fitting to those data points.

**(E) Variable Imprortance** The top 10 variables are a mixture of Biological and Manufacturing, but the Manufacturing process variables dominate the model, and make up 8 of the top 10 in importance. This is good news for our manufacturer, as it the Biological are fixed, but the manufacturing process is flexible and these levers identified below can influence the Yield.

 **(F) Variable Relationships**

With the top drivers given by our model, the relationship between these particular predictors and the Yield is more important to understand in depth. An analysis of the relationships reveal Manufacturing Process 13, 6, and 7 are negatively correlated with the Yield, which means if these values can be driven lower the resulting Yield may increase. The Biological Materials 03 and Manufacturing Process 09 are also influential variables but have a strong positive correlation with the Yield. There is quite a bit of correlation among variables, with process 13 very highly related to other variables. This implies only one lever should be adjusted at a time, and the best place to start would be trying to decrease process 13.



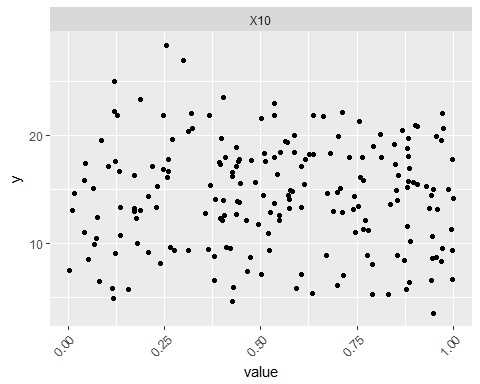
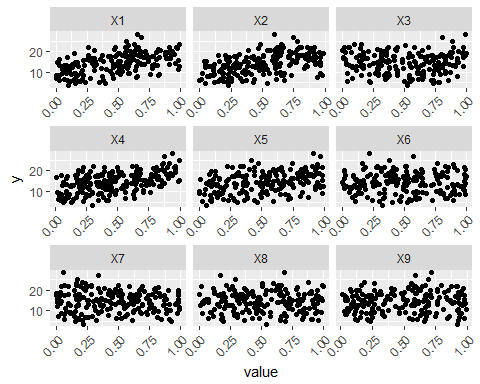
## K.J 7.2

Create several models using the Friedman generated data and evaluate prediction accuracy and variable importance.

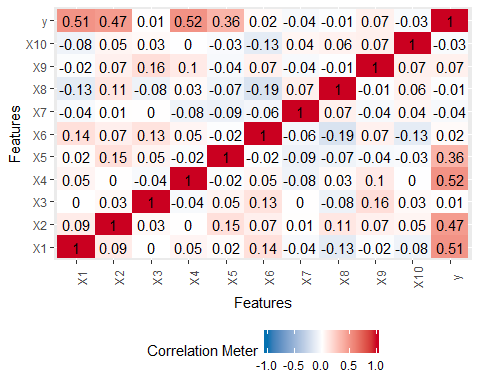
library(mlbench)  
set.seed(200)  
  
#training data  
trndata <- mlbench.friedman1(200, sd = 1)  
  
trn.df <- data.frame(trndata$x)  
trn.df$y <- trndata$y  
  
  
#test data  
testdf <- mlbench.friedman1(5000, sd = 1)  
  
test.df <- data.frame(testdf$x)  
test.df$y <- testdf$y  
  
#review training data  
str(trn.df)

## 'data.frame': 200 obs. of 11 variables:  
## $ X1 : num 0.534 0.584 0.59 0.691 0.667 ...  
## $ X2 : num 0.648 0.438 0.588 0.226 0.819 ...  
## $ X3 : num 0.8508 0.6727 0.4097 0.0334 0.7168 ...  
## $ X4 : num 0.1816 0.6692 0.3381 0.0669 0.8032 ...  
## $ X5 : num 0.929 0.1638 0.8941 0.6374 0.0831 ...  
## $ X6 : num 0.3618 0.4531 0.0268 0.525 0.2234 ...  
## $ X7 : num 0.827 0.649 0.179 0.513 0.664 ...  
## $ X8 : num 0.421 0.845 0.35 0.797 0.904 ...  
## $ X9 : num 0.5911 0.9282 0.0176 0.6899 0.397 ...  
## $ X10: num 0.589 0.758 0.444 0.445 0.55 ...  
## $ y : num 18.5 16.1 17.8 13.8 18.4 ...

plot\_scatterplot(data=trn.df, by = "y")



plot\_correlation(trn.df, type = c("all"),  
 maxcat = 20)



Initial analysis of the training data shows some limited linear relationship between x4 and the y value, however the variables x1, x2, and x3 show additional non-linear curved relationships.

Several models were tuned to this data, a glm model, knn model, and an SVM model, and the resulting variable importance compared to the results of the MARS run using the ‘earth’ model.

## Generalized Linear Model   
##   
## 200 samples  
## 10 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 4 times)   
## Summary of sample sizes: 180, 180, 180, 180, 180, 180, ...   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 2.426757 0.774816 1.932749

## k-Nearest Neighbors   
##   
## 200 samples  
## 10 predictor  
##   
## Pre-processing: centered (10), scaled (10)   
## Resampling: Cross-Validated (10 fold, repeated 4 times)   
## Summary of sample sizes: 180, 180, 180, 180, 180, 180, ...   
## Resampling results across tuning parameters:  
##   
## k RMSE Rsquared MAE   
## 1 4.239267 0.3846485 3.496360  
## 2 3.460409 0.5349682 2.847362  
## 3 3.336335 0.5566463 2.745877  
## 4 3.225266 0.5885398 2.645595  
## 5 3.166818 0.6123234 2.622512  
## 6 3.136655 0.6257864 2.578383  
## 7 3.092965 0.6440738 2.539632  
## 8 3.093366 0.6536051 2.522604  
## 9 3.062449 0.6714501 2.476036  
## 10 3.066986 0.6795993 2.495475  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was k = 9.

## Support Vector Machines with Radial Basis Function Kernel   
##   
## 200 samples  
## 10 predictor  
##   
## Pre-processing: centered (10), scaled (10)   
## Resampling: Cross-Validated (10 fold, repeated 4 times)   
## Summary of sample sizes: 180, 180, 180, 180, 180, 180, ...   
## Resampling results across tuning parameters:  
##   
## C RMSE Rsquared MAE   
## 0.25 2.488407 0.8051956 1.989127  
## 0.50 2.241277 0.8201648 1.775611  
## 1.00 2.074568 0.8389337 1.634643  
## 2.00 1.937784 0.8586459 1.522829  
## 4.00 1.847685 0.8700445 1.463304  
## 8.00 1.827858 0.8730906 1.457398  
## 16.00 1.818358 0.8750138 1.458677  
## 32.00 1.817895 0.8751099 1.457463  
## 64.00 1.817895 0.8751099 1.457463  
## 128.00 1.817895 0.8751099 1.457463  
## 256.00 1.817895 0.8751099 1.457463  
## 512.00 1.817895 0.8751099 1.457463  
## 1024.00 1.817895 0.8751099 1.457463  
## 2048.00 1.817895 0.8751099 1.457463  
##   
## Tuning parameter 'sigma' was held constant at a value of 0.05442019  
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were sigma = 0.05442019 and C = 32.

## Loading required package: earth

## Loading required package: plotmo

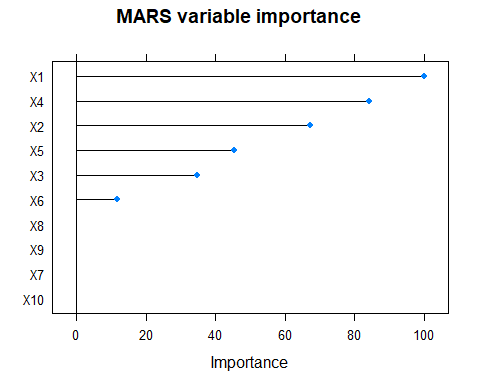
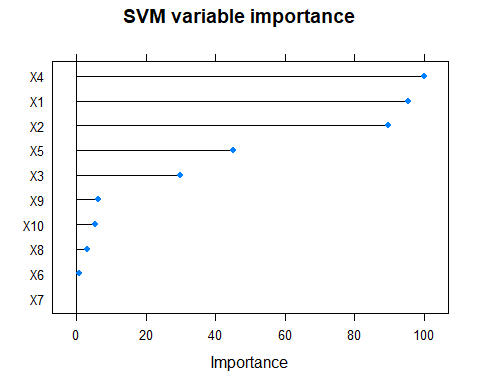
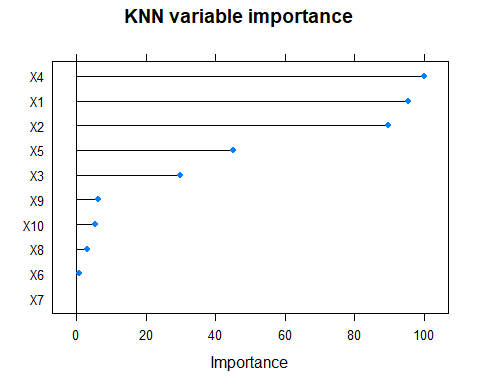
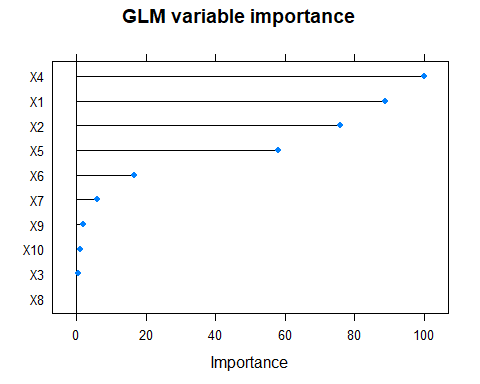
## Loading required package: plotrix

## Loading required package: TeachingDemos

## Multivariate Adaptive Regression Spline   
##   
## 200 samples  
## 10 predictor  
##   
## Pre-processing: centered (10), scaled (10)   
## Resampling: Cross-Validated (10 fold, repeated 4 times)   
## Summary of sample sizes: 180, 180, 180, 180, 180, 180, ...   
## Resampling results across tuning parameters:  
##   
## nprune RMSE Rsquared MAE   
## 2 4.415835 0.2324315 3.658213  
## 8 1.712590 0.8836443 1.342736  
## 15 1.627378 0.8937940 1.270309  
##   
## Tuning parameter 'degree' was held constant at a value of 1  
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were nprune = 15 and degree = 1.

Comparing the variable importance of the four models shows that MARS model correctly identified the most relevant variables, X4,X1,X2,and X3. The SVM model placed greater importance on X4, and lesser on x1-x3 but followed the same pattern of importance. The KNN model was very similar in importance ranking to the SVM, but the GLM model was more interesting. The GLM gave the most importance to X4 as the others, but X2, x1, and X5 are the next most important, with X3 showing no importance in this view.

Comparing the accuracy measures of the models shows that the Mars model had both the lowest RMSE and the highest Rsquared value, making it a better model. The SVM model also did well, and considered the correct variables with the same level of important, but over-emphasized the X4 contribution. Despite the similar importance ranking, the KNN model had a much higher RMSE and lower Rsquared. The GLM model performed better than the KNN, even though it had an unusual variable importance pattern.



## RMSE Rsquared MAE  
## glm 2.697068 0.7084666 2.060054  
## knn 3.117232 0.6556622 2.489991  
## svm 2.056483 0.8285460 1.563646  
## mars 1.813647 0.8677298 1.391184

## KJ 7.5

Revisit the problem 6.3 and evaluate non-linear models using similar pre-processing. Compare the models.

Using the same pre-processed data used in problem 6.3 above, three non-linear models were created, MARS, SVM, and KNN models.

## k-Nearest Neighbors   
##   
## 143 samples  
## 39 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 4 times)   
## Summary of sample sizes: 129, 129, 130, 127, 129, 128, ...   
## Resampling results across tuning parameters:  
##   
## k RMSE Rsquared MAE   
## 1 1.650573 0.3249774 1.3110292  
## 2 1.317650 0.4946330 1.0400719  
## 3 1.251096 0.5424076 0.9888885  
## 4 1.242347 0.5516421 0.9811704  
## 5 1.256210 0.5457640 1.0014107  
## 6 1.234354 0.5663789 0.9994399  
## 7 1.245296 0.5667917 1.0222128  
## 8 1.236446 0.5706592 1.0146242  
## 9 1.245293 0.5661490 1.0319599  
## 10 1.255701 0.5631368 1.0407625  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was k = 6.

## Support Vector Machines with Radial Basis Function Kernel   
##   
## 143 samples  
## 39 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 4 times)   
## Summary of sample sizes: 128, 129, 130, 128, 128, 127, ...   
## Resampling results across tuning parameters:  
##   
## C RMSE Rsquared MAE   
## 0.25 1.351600 0.5628553 1.1180796  
## 0.50 1.227275 0.5928008 1.0100285  
## 1.00 1.133083 0.6356418 0.9147838  
## 2.00 1.116423 0.6329880 0.8758501  
## 4.00 1.144085 0.6111783 0.8786246  
## 8.00 1.169804 0.5909104 0.8986092  
## 16.00 1.181552 0.5822775 0.9093287  
## 32.00 1.181552 0.5822775 0.9093287  
## 64.00 1.181552 0.5822775 0.9093287  
## 128.00 1.181552 0.5822775 0.9093287  
## 256.00 1.181552 0.5822775 0.9093287  
## 512.00 1.181552 0.5822775 0.9093287  
## 1024.00 1.181552 0.5822775 0.9093287  
## 2048.00 1.181552 0.5822775 0.9093287  
##   
## Tuning parameter 'sigma' was held constant at a value of 0.01936116  
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were sigma = 0.01936116 and C = 2.

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

## Multivariate Adaptive Regression Spline   
##   
## 143 samples  
## 39 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 4 times)   
## Summary of sample sizes: 129, 128, 130, 129, 127, 129, ...   
## Resampling results across tuning parameters:  
##   
## nprune RMSE Rsquared MAE   
## 2 1.591795 0.2739579 1.288793  
## 9 1.544204 0.5101615 1.125859  
## 17 1.550797 0.5249873 1.101475  
##   
## Tuning parameter 'degree' was held constant at a value of 1  
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were nprune = 9 and degree = 1.

## Variable Ridge Mars KNN SVM  
## 1 ManufacturingProcess13 100.00000 66.16956 100.00000 100.00000  
## 2 ManufacturingProcess36 95.23189 100.00000 95.23189 95.23189  
## 3 BiologicalMaterial03 85.64490 55.73967 85.64490 85.64490  
## 4 ManufacturingProcess17 84.91709 NA 84.91709 84.91709  
## 5 ManufacturingProcess09 73.93419 NA 73.93419 73.93419  
## 6 ManufacturingProcess31 68.39801 NA 68.39801 68.39801  
## 7 ManufacturingProcess33 67.08852 NA 67.08852 67.08852  
## 8 BiologicalMaterial11 62.52503 0.00000 62.52503 62.52503  
## 9 ManufacturingProcess12 55.05922 NA 55.05922 55.05922  
## 10 ManufacturingProcess06 53.62140 NA 53.62140 53.62140  
## 11 ManufacturingProcess30 NA 55.73967 NA NA  
## 12 ManufacturingProcess39 NA 38.96028 NA NA  
## 13 ManufacturingProcess05 NA 10.21650 NA NA  
## 14 BiologicalMaterial05 NA 0.00000 NA NA  
## 15 BiologicalMaterial09 NA 0.00000 NA NA  
## 16 BiologicalMaterial10 NA 0.00000 NA NA

## RMSE Rsquared MAE  
## ridge 1.439907 0.4974571 1.183939  
## knn 1.575981 0.3492612 1.246061  
## svm 1.488523 0.4259091 1.211356  
## mars 1.563489 0.3603892 1.274098

Comparing the three new models with the prior ridge model shows that the ridge model has a better RMSE and Rsquared value.

Reviewing the top 10 variables for each model show most had similiar results. Most found ManufacturingProcess13 as the most important variable, with MARS model showing this as second most important. all models selected only two biological variables, BiologicalMaterial03 and BiologicalMaterial11, within the top 10. It is interesting to note the MARS model identified a much different set of lower-ranked manufacturing process variables when building the model, and only returned 6 variables of any importance to the model.

The Biological Material predictors are ranked similarly across the models.

## KJ 8.1

Recreate simulated data from 7.2, fit random forest model and compare

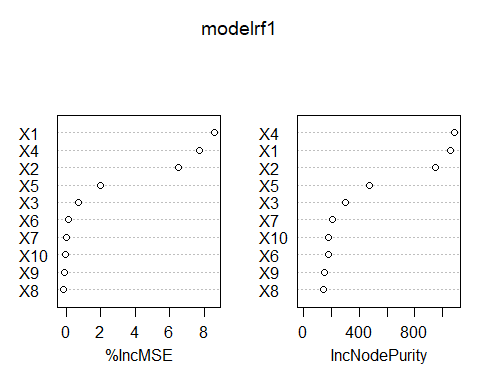
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

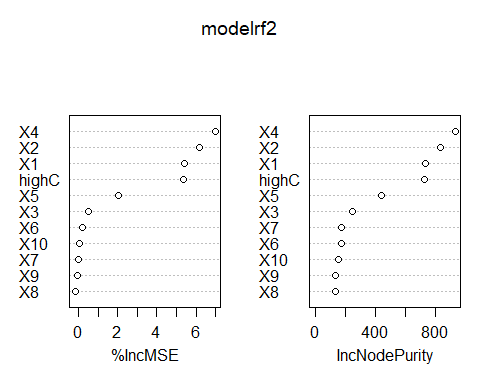
## The following object is masked from 'package:ggplot2':  
##   
## margin

 **(a) Random Forest Variable IMportnace** The random forest model gave importance to variables x1 - x5, but did not identify the remaining variables is important.

**(b) Add a predictor** A predictor, highC, is added that is highly correlated with variable 1 and evaluate changed performance.

The new variable has a high overall importance in the new random forest model but both it and Variable 1 having similar importance are now less important than variable 2. The importance of variable 1 has decreased from 8.7 before the new addition, to 5.4 after the new variable is added. This is interesting because the new variable is not adding any new information to the model, but the importance now is spread between the two variables.

## [1] 0.9993539



**(c) Conditional Inference Tree** Create a random forest model with conditional inference trees and evaluate importance comparison.

The third model, creating using cforest from the party package and using the original data set has different importance values when compared to the first random forest model. It has placed a higher importance on X1, X2, and X5, and decreased the importance of X3.

The fourth model, created using the cforest function and the modified data is different from the results found on same data using the randomForest model. The importance variables are ranked similarly, but with less importance to X1 and highC, and more importance given to X4. Although it is different than the randomForest model (model 2) it contains the same problem of identifying a linearly correlated variable as highly important.

The fifth and sixth model importance are the fourth and third models but with importance calculated using the conditional method. This resulted in all variables showing a significant less important than the other views, but their relative ranking shows different patterns than seen in the non-conditional views. For the fifth it now has x2, x4, then x1 as the top three in importance, while the six view has X2, X4 and highC replacing X1 in the top 3.

library(party)

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: sandwich

#Model old + New data using cforest function in party library  
  
trn.dfold <- trn.df[,1:11]  
  
set.seed(867)  
modelrf3<- cforest(y ~ ., data = trn.dfold, controls = cforest\_control(ntree = 1000))  
  
set.seed(53)  
modelrf4 <- cforest(y ~ ., data = trn.df, controls = cforest\_control(ntree = 1000))  
  
vars.imp.rf3 <- varimp(modelrf3)  
vars.imp.rf4 <- varimp(modelrf4)  
vars.imp.rf3C <- varimp(modelrf3, conditional = TRUE)  
vars.imp.rf4C <- varimp(modelrf4, conditional = TRUE)  
  
#consol data  
d1 <- data.frame(cbind(Var = row.names(var.imp.rf1), "rf1" = var.imp.rf1[,1]))  
d2 <- data.frame(cbind(Var = row.names(var.imp.rf2), "rf2" = var.imp.rf2[,1]))  
d3 <- data.frame(cbind(Var = names(vars.imp.rf3), "rf3" = vars.imp.rf3))  
d4 <- data.frame(cbind(Var = names(vars.imp.rf4), "rf4"= vars.imp.rf4))  
d5 <- data.frame(cbind(Var = names(vars.imp.rf3C), "rf3.Cond" = vars.imp.rf3C))  
d6 <- data.frame(cbind(Var = names(vars.imp.rf4C), "rf4.Cond"= vars.imp.rf4C))  
  
#unite!  
comp.vars.rf <- suppressWarnings(full\_join(d1, d2, by='Var') %>%  
 full\_join(.,d3 , by='Var')%>%  
 full\_join(.,d4 , by = 'Var')%>%full\_join(.,d5, by = 'Var')%>%  
 full\_join(.,d6, by = 'Var'))  
  
#write.csv(comp.vars.rf, "comprf.csv")

**(d) Additional models**  
Using a gbm and a cubist model generates similar results

The cubist model gave the exact same results when run with default or conditional importance measures. 1 and 3, calculated on the old data, give all importance to variables x1-x6, and assign none to the later variables. The importance placed on X6 was surprising. In the model using the new predictor, the importance of X1 was reduced and the new predictor given greater importance, while rest of the variables remain largely unchanged with this addition.

The gmb model identified X4 as the most important variable for both models and regardless of condition value.X1 comes next, and this model does a better job identifying that the new variable, highC, is not as important. highC is given an importance of 11, it does not as dramatically decrease the X1 variable in importance, or change magnitude or order of the other variables very , and does not decrease the most important variable, X4, at all.

## Loading required package: survival

##   
## Attaching package: 'survival'

## The following object is masked from 'package:caret':  
##   
## cluster

## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.3

## [1] "Cube Variable Importance"

## Var cube1 cube2 cube2.cond cube3.cond  
## 1 X1 71 60 71 60  
## 2 X3 44.5 43 44.5 43  
## 3 X2 57.5 63.5 57.5 63.5  
## 4 X4 48.5 50 48.5 50  
## 5 X5 35 34 35 34  
## 6 X6 15.5 9.5 15.5 9.5  
## 7 X7 0 1 0 1  
## 8 X8 0 7 0 7  
## 9 X9 0 0 0 0  
## 10 X10 0 0 0 0  
## 11 highC <NA> 30.5 <NA> 30.5

## [1] "GBM Variable Importance"

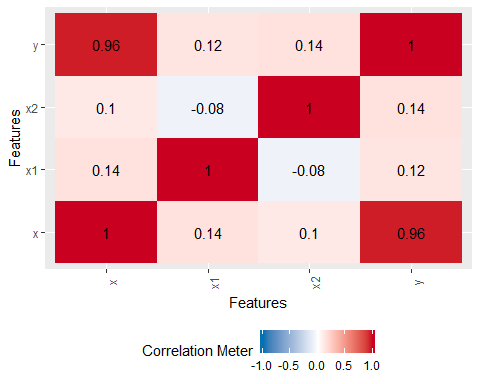
## Var gbm1 gbm2 gbm.cond.x  
## 1 X1 95.5862188857496 57.1883447491172 95.5862188857496  
## 2 X2 72.2999883195159 73.02881208328 72.2999883195159  
## 3 X3 26.9497847529522 26.2303432524158 26.9497847529522  
## 4 X4 100 100 100  
## 5 X5 44.0103623953548 44.3753214094923 44.0103623953548  
## 6 X6 1.360881123677 1.04724056210573 1.360881123677  
## 7 X7 1.66390051466439 1.29021446601346 1.66390051466439  
## 8 X8 0.424534263419646 0.388792256056952 0.424534263419646  
## 9 X9 0 0 0  
## 10 X10 0.0223224195753524 0.125998251228834 0.0223224195753524  
## 11 highC <NA> 36.6958270287506 <NA>  
## gbm.cond.y  
## 1 57.1883447491172  
## 2 73.02881208328  
## 3 26.2303432524158  
## 4 100  
## 5 44.3753214094923  
## 6 1.04724056210573  
## 7 1.29021446601346  
## 8 0.388792256056952  
## 9 0  
## 10 0.125998251228834  
## 11 36.6958270287506

## KJ 8.2

Using a simulation to show tree bias with different granularity.

First a data set was created, with a related x predictor and y response, then two additional unrelated variables, x2, and x3, were added to the data. A tree model was run on this data set and variable importance analyzed.

X2 and X3 have no relationship to X1 or Y, but each is selected for the model.



## 8.6 Permiablity

Review permeability problem from problem 6.2 and train several tree models, evaluate the results. Permeability is a data set with 165 responses. The fingerprints data contains 1,107 predictors for those 165 responses. There are 719 predictors with almost no variance which will not work will with a tree model and are removed, resulting in 388 remaining predictors.

80% of the data was split to create a training /test set.

data("permeability")  
  
df.perm <- permeability  
summary(df.perm)

## permeability   
## Min. : 0.06   
## 1st Qu.: 1.55   
## Median : 4.91   
## Mean :12.24   
## 3rd Qu.:15.47   
## Max. :55.60

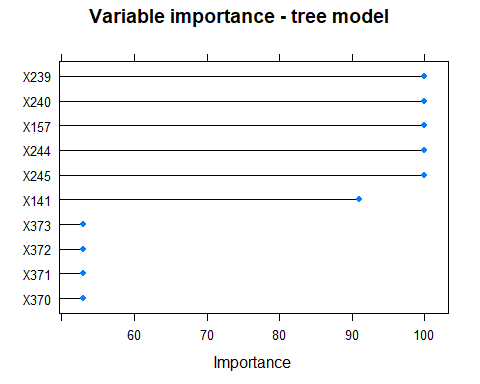
#remove zeros  
zees <- nearZeroVar(fingerprints)  
df.fing <- as.data.frame(fingerprints[,-zees])  
#remove zeros  
dim(df.fing)

## [1] 165 388

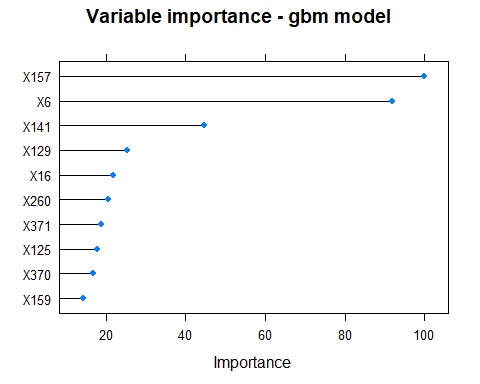
#spliting into training and test sets  
  
set.seed(42)  
# Training Rows created  
trnRows <- createDataPartition(df.perm,p =.8,list = FALSE, times = 1)  
  
#select training and testing subsets - for this example can use just the first sample created  
trn.df.fing <- df.fing[trnRows[,1],]  
trn.df.perm <- df.perm[trnRows[,1],]  
  
test.df.fing <- df.fing[-trnRows[,1],]  
test.df.perm <- df.perm[-trnRows[,1],]

Three tree-based models were used to train and evaluate the models:  
a single tree, a boosted tree (gbm), and a random forest tree. The gbm and random forest model have very similar importance for their top 10 predictors, with top value for both being X6, X157, X129 while the single tree model has different values such as X141, X6, and X240. All three appreciate the X6 inputs but have different outputs.

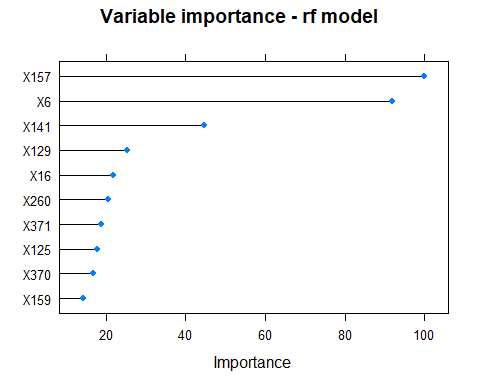
#models  
#train test df  
library(rpart)  
modeltree1 <- train(x = trn.df.fing, y=trn.df.perm,   
 method='rpart2',   
 trControl= trainControl(method = "cv"),  
 tuneLength = 10)  
   
  
modelgbm2 <- train(x = trn.df.fing, y=trn.df.perm,   
 method='gbm',   
 trControl=ctrlSamp1,  
 verbose = FALSE)  
 #metric = "ROC")  
  
modelrf5 <- train(x = trn.df.fing, y=trn.df.perm,   
 method='rf',   
 importance = TRUE,  
 ntrees = 1000)  
  
vars.imp.tr1 <- varImp(modeltree1, top=10)  
vars.imp.g2 <- varImp(modelgbm2, top =10)  
vars.imp.rf5 <-varImp(modelgbm2, top=10)  
  
plot(vars.imp.tr1, top=10,main="Variable importance - tree model")



plot(vars.imp.g2, top=10,main="Variable importance - gbm model")

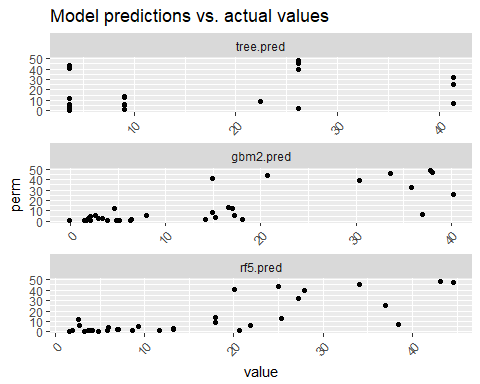


plot(vars.imp.rf5,top=10, main="Variable importance - rf model")



Of the models created for this exercise, I would choose the random forest model. It is computational more intensive,but has the best Rsquared value and lowest RMSE, although the MAE is only slight better than the single tree model. The computation time, however to compute the random forest model may be consideration, and the lab may be willing to trade speed for accuracy.

## RMSE Rsquared MAE  
## tree 14.84985 0.2565413 9.745898  
## gbm 10.75331 0.5807727 7.690129  
## rf 10.78960 0.6012688 8.344001

 ##8.7 Continuing Chemical Manufacturing Process

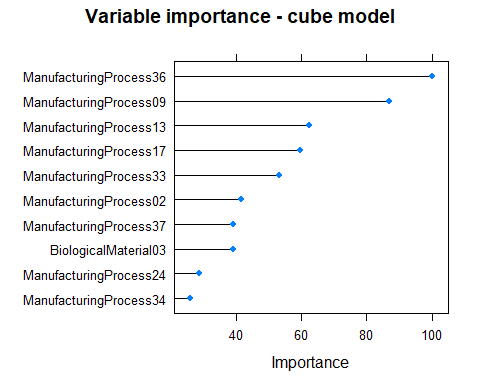
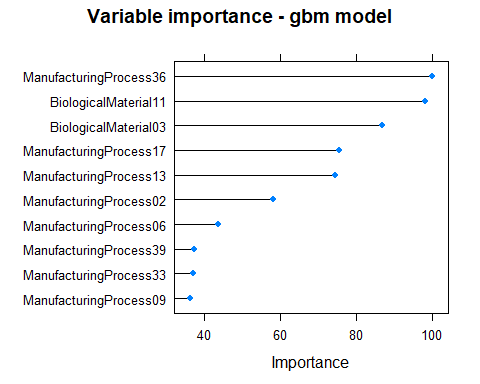
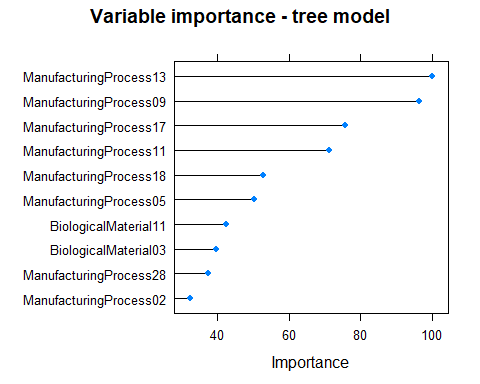
Using same data processing as in problem 6.2 and 7.5, build and evaluate several tree models. Compare these results to prior results using other methods.

To compare, I created a tee model, a gbm model, and a cube model using the preprocessed data. These models were then compared, and the accuracy and rsquared evaluated against results from prior exercises mode results.

The variable importance in these three models are different from other models. Most interesting is the gbm model, which is the only model that lists a biological element as the most important predictor. The best model, cube, has a different variable as the most important, ManufacturingProcess09, which did not show up in the top 3 for any of the linear or non-linear models, although was in position 2 on the tree model.

After reviewing the combined metrics of the linear, non-linear, and tree models, the best RMSE and Rsquared value are resulting from the cube model.

## CART   
##   
## 143 samples  
## 39 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 130, 128, 129, 129, 128, 130, ...   
## Resampling results across tuning parameters:  
##   
## maxdepth RMSE Rsquared MAE   
## 1 1.732330 0.1477952 1.3686816  
## 2 1.602040 0.2510433 1.2829519  
## 3 1.549117 0.2976783 1.2203910  
## 4 1.516699 0.3360986 1.2243426  
## 5 1.425564 0.4024188 1.1404817  
## 6 1.340267 0.4576561 1.0668649  
## 7 1.304195 0.4934119 0.9974559  
## 8 1.277185 0.5067224 0.9783256  
## 9 1.304367 0.5075696 0.9953958  
## 10 1.305224 0.5059311 0.9998822  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was maxdepth = 8.



## RMSE Rsquared MAE Models  
## 1 1.323798 0.5628049 1.054396 cube  
## 2 1.439907 0.4974571 1.183939 ridge  
## 3 1.439907 0.4974571 1.183939 ridge.1  
## 4 1.460953 0.4554322 1.178201 gbm  
## 5 1.488523 0.4259091 1.211356 svm.1  
## 6 1.563489 0.3603892 1.274098 mars.1  
## 7 1.575981 0.3492612 1.246061 knn.1  
## 8 1.813647 0.8677298 1.391184 mars  
## 9 2.002414 0.1444735 1.544838 tree  
## 10 2.056483 0.8285460 1.563646 svm  
## 11 2.697068 0.7084666 2.060054 glm  
## 12 3.117232 0.6556622 2.489991 knn

