DATA624 Group 3

**TEAM:**

## **Kuiete Tchoupou Alain**

## **Littlejohn Jeffrey**

## **Malhotra Samriti**

## **Mishra Rajwant**

## **Ng Jimmy**

Homework Assignment 2

Table of Contents

[Kuhn & Johnson 7 2](#__RefHeading___Toc6528_1911604043)

[HW\_KJ\_7.2 2](#__RefHeading___Toc6530_1911604043)

[Tune several models on these data. For example: 3](#__RefHeading___Toc6275_1911604043)

[MARS 5](#__RefHeading___Toc6277_1911604043)

[SVM 6](#__RefHeading___Toc6279_1911604043)

[NNET 7](#__RefHeading___Toc6281_1911604043)

[Which models appear to give the best performance? Does MARS select the informative predictors (those named X1-X5)? 8](#__RefHeading___Toc6283_1911604043)

[HW\_KJ\_7.5 11](#__RefHeading___Toc6532_1911604043)

[(a). Which nonlinear regression model gives the optimal resampling and test set performance? 12](#__RefHeading___Toc6287_1911604043)

[(b). Which predictors are most important in the optimal nonlinear regression model? Do either the biological or process variables dominate the list? How do the top ten important predictors compare to the top ten predictors from the optimal linear model? 13](#__RefHeading___Toc6289_1911604043)

[(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? 14](#__RefHeading___Toc6291_1911604043)

[Kuhn & Johnson 8 18](#__RefHeading___Toc6534_1911604043)

[HW\_KJ\_8.1 18](#__RefHeading___Toc6536_1911604043)

[8.1. Recreate the simulated data from Exercise 7.2: 19](#__RefHeading___Toc6295_1911604043)

[(a) Fit a random forest model to all of the predictors, then estimate the variable importance scores: 19](#__RefHeading___Toc6297_1911604043)

[Did the random forest model significantly use the uninformative predictors (V6 – V10)? 19](#__RefHeading___Toc6299_1911604043)

[(b) Now add an additional predictor that is highly correlated with one of the informative predictors. For example: 20](#__RefHeading___Toc6301_1911604043)

[Fit another random forest model to these data. Did the importance score for V1 change? What happens when you add another predictor that is also highly correlated with V1? 20](#__RefHeading___Toc6303_1911604043)

[(c) Use the cforest function in the party package to fit a random forest model using conditional inference trees. The party package function varimp can calculate predictor importance. The conditional argument of that function toggles between the traditional importance measure and the modified version described in Strobl et al. (2007). Do these importances show the same pattern as the traditional random forest model? 21](#__RefHeading___Toc6305_1911604043)

[(d) Repeat this process with different tree models, such as boosted trees and Cubist. Does the same pattern occur? 24](#__RefHeading___Toc6307_1911604043)

[HW\_KJ\_8.2 25](#__RefHeading___Toc6538_1911604043)

[8.2. Use a simulation to show tree bias with different granularities. 25](#__RefHeading___Toc6309_1911604043)

[HW\_KJ\_8.3 26](#__RefHeading___Toc6540_1911604043)

[8.3. In stochastic gradient boosting the bagging fraction and learning rate will govern the construction of the trees as they are guided by the gradient. Although the optimal values of these parameters should be obtained through the tuning process, it is helpful to understand how the magnitudes of these parameters affect magnitudes of variable importance. Figure 8.24 provides the variable importance plots for boosting using two extreme values for the bagging fraction (0.1 and 0.9) and the learning rate (0.1 and 0.9) for the solubility data. The left-hand plot has both parameters set to 0.1, and the right-hand plot has both set to 0.9: 26](#__RefHeading___Toc6311_1911604043)

[(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? 27](#__RefHeading___Toc6313_1911604043)

[(b) Which model do you think would be more predictive of other samples? 27](#__RefHeading___Toc6542_1911604043)

[(c) How would increasing interaction depth affect the slope of predictor importance for either model in Fig. 8.24 27](#__RefHeading___Toc6317_1911604043)

# Kuhn & Johnson 7

## HW\_KJ\_7.2

|  |
| --- |
| Friedman (1991) introduced several benchmark data sets create by simulation. One of these simulations used the following nonlinear equation to create data: *y*=10sin(*π**x*1*x*2)+20(*x*3−0.5)2+10*x*4+5*x*5+*N*(0,*σ*2); where the *x* values are random variables uniformly distributed between [0,1] (there are also 5 other non-informative variables also created in the simulation).  The package mlbench contains a function called mlbench.friedman1 that simulates these data: |
| **library**(mlbench)  set.seed(200)  trainingData <- mlbench.friedman1(200, sd = 1)  *## We convert the 'x' data from a matrix to a data frame*  *## One reason is that this will give the columns names.*  trainingData$x <- data.frame(trainingData$x)  *## Look at the data using*  featurePlot(trainingData$x, trainingData$y) |
|  |
| **Tune several models on these data. For example:** **library**(caret)  knnModel <- train(x = trainingData$x,  y = trainingData$y,  method = "knn",  preProc = c("center", "scale"),  tuneLength = 10)  knnModel  *## k-Nearest Neighbors*  *##*  *## 200 samples*  *## 10 predictor*  *##*  *## Pre-processing: centered (10), scaled (10)*  *## Resampling: Bootstrapped (25 reps)*  *## Summary of sample sizes: 200, 200, 200, 200, 200, 200, ...*  *## Resampling results across tuning parameters:*  *##*  *## k RMSE Rsquared MAE*  *## 5 3.466085 0.5121775 2.816838*  *## 7 3.349428 0.5452823 2.727410*  *## 9 3.264276 0.5785990 2.660026*  *## 11 3.214216 0.6024244 2.603767*  *## 13 3.196510 0.6176570 2.591935*  *## 15 3.184173 0.6305506 2.577482*  *## 17 3.183130 0.6425367 2.567787*  *## 19 3.198752 0.6483184 2.592683*  *## 21 3.188993 0.6611428 2.588787*  *## 23 3.200458 0.6638353 2.604529*  *##*  *## RMSE was used to select the optimal model using the smallest value.*  *## The final value used for the model was k = 17.*  knnPred <- predict(knnModel, newdata = testData$x)  *## The function 'postResample' can be used to get the test set*  *## performance values*  postResample(pred = knnPred, obs = testData$y)  *## RMSE Rsquared MAE*  *## 3.2040595 0.6819919 2.5683461* |
| **MARS** arsGrid <- expand.grid(degree = 1:2, nprune = seq(2, 14,by = 2))  marsModel <- train(x = trainingData$x,  y = trainingData$y,  method = 'earth',  tuneGrid = marsGrid,  trControl = trainControl(method = "cv"))  ggplot(marsModel) + labs(title = "MARS Cross-Validated RMSE Profile") + theme\_classic() |
|  |
| **SVM** svmModel <- train(x = trainingData$x,  y = trainingData$y,  method = 'svmRadial',  tuneLength = 14,  trControl = trainControl(method = "cv"))  ggplot(svmModel) + labs(title = "SVM Cross-Validated RMSE Profile") + theme\_gray() |
|  |
| **NNET** nnetGrid <- expand.grid(.size = c(1:10), .decay = c(0, 0.01, .1))  nnetModel <- train(trainingData$x,  trainingData$y,  method = "nnet",  tuneGrid = nnetGrid,  trControl = trainControl(method="cv"),  preProc = c("center", "scale"),  linout = TRUE,  trace = FALSE,  MaxNWts = 10 \* (ncol(trainingData$x) + 1) + 10 + 1,  maxit = 500)  ggplot(nnetModel) + labs(title = "NNET Cross-Validated RMSE Profile") + theme\_linedraw() |
|  |
| **Which models appear to give the best performance? Does MARS select the informative predictors (those named X1-X5)?** |
| As displayed in the summary table, MARS appears to be supreme comparing to KNN, SVM and NNET in terms of RMSE, R squared and MAE on the test set. The variable importance plot for MARS indicates that the variables X1 through 5 are the most important. X1 appears to be the most important variable. MARS appears to have successfully selected the most informative predictors and outperformed the other models in this exercise.  *# Prediction*  *# knnPred is already given to us from the textbook*  marsPred <- predict(marsModel, newdata = testData$x)  svmPred <- predict(svmModel, newdata = testData$x)  nnetPred <- predict(nnetModel, newdata = testData$x)  *# Performance*  knnPerformance <- postResample(pred = knnPred, obs = testData$y)  marsPerformance <- postResample(pred = marsPred, obs = testData$y)  svmPerformance <- postResample(pred = svmPred, obs = testData$y)  nnetPerformance <- postResample(pred = nnetPred, obs = testData$y)  *# Summary*  rbind(  *# KNN*  "KNN Train" = c("RMSE" = max(knnModel$results$RMSE),  "RSquared" = max(knnModel$results$RMSE),  "MAE" = max(knnModel$results$RMSE)),  "KNN Test" = knnPerformance,  *# MARS*  "MARS Train" = c("RMSE" = max(marsModel$results$RMSE),  "RSquared" = max(marsModel$results$Rsquared),  "MAE" = max(marsModel$results$MAE)),  "MARS Test" = marsPerformance,  *# SVM*  "SVM Train" = c(max(svmModel$results$RMSE),  max(svmModel$results$Rsquared),  max(svmModel$results$MAE)),  "SVM Test" = svmPerformance,  *# NNET*  "NNET Train" = c(max(nnetModel$results$RMSE),  max(nnetModel$results$Rsquared),  max(nnetModel$results$MAE)),  "NNET Test" = nnetPerformance) %>%  kable(caption = "Model Performance", digits = 3) %>%  kable\_styling() |
|  |





|  |
| --- |
| *# Variable Importance*  marsImportance <- caret::varImp(marsModel)  marsImportance$importance %>%  as.data.frame() %>%  tibble::rownames\_to\_column() %>%  dplyr::mutate(name = forcats::fct\_inorder(rowname)) %>%  arrange(desc(Overall)) %>%  head(10) %>%  ggplot(., aes(x = reorder(name, Overall), y = Overall)) +  geom\_point() +  geom\_segment(aes(x = name, xend = name, y = 0, yend = Overall)) +  ggtitle("Variable Importance: MARS for Simulated Data Set") +  labs(x = "Variable", y = "Importance") +  coord\_flip() +  theme\_minimal() |
|  |
|  |

## HW\_KJ\_7.5

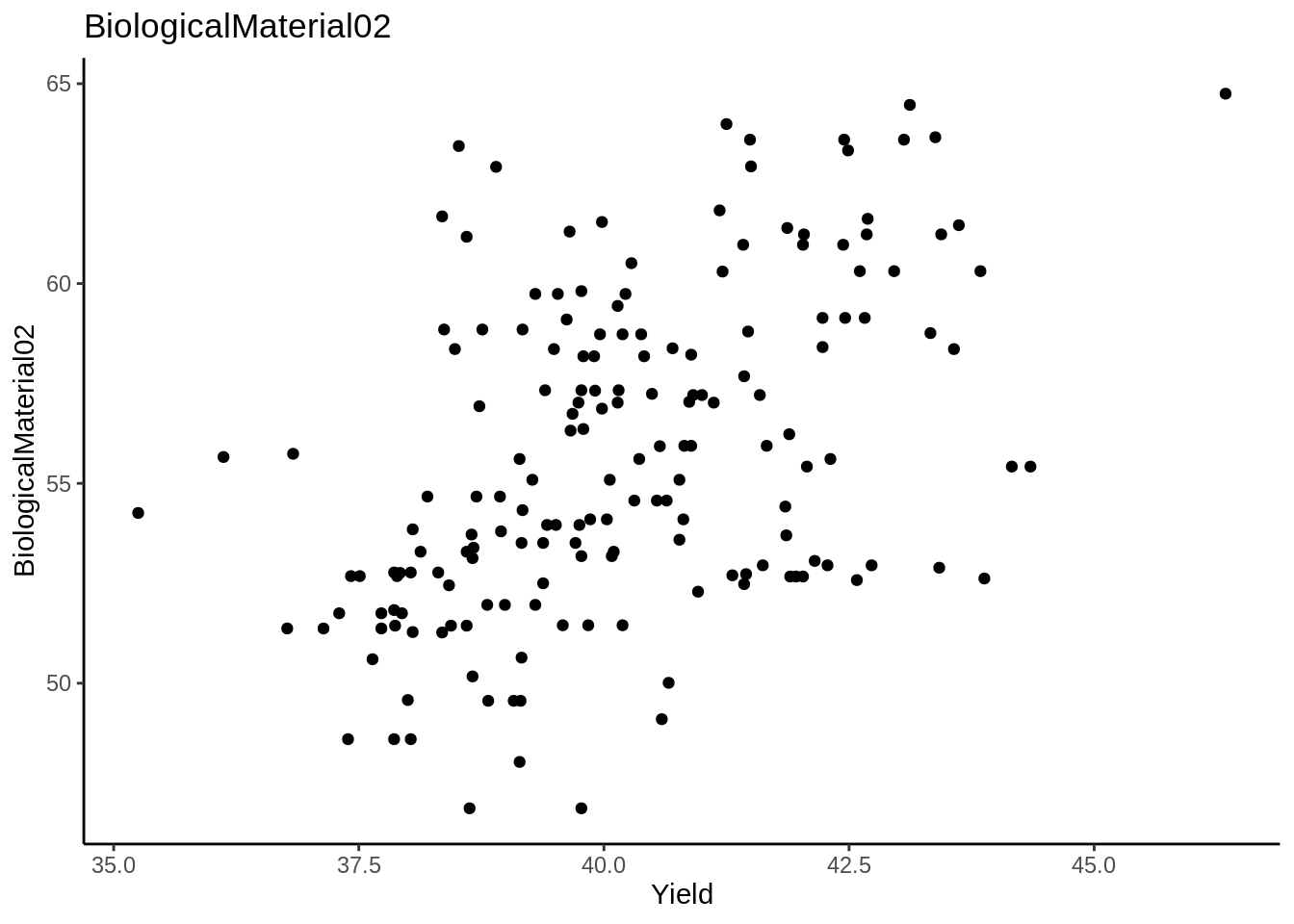
|  |
| --- |
| Exercise 6.3 describes data for a chemical manufacturing process. Use the same data imputation, data splitting, and preprocessing steps as before and train several nonlinear regression models. **(a). Which nonlinear regression model gives the optimal resampling and test set performance?** *# data prep - using the same methodology from exercise 6.3*  data(ChemicalManufacturingProcess)  impu\_data <- kNN(ChemicalManufacturingProcess, imp\_var = FALSE)  n <- nrow(impu\_data)  i.training <- sort(sample(n,round(n\*0.8)))  L.training <- impu\_data[i.training,]  L.test <- impu\_data[-i.training,]  *# check if any missing left*  **if**(!any(colSums(is.na(impu\_data)) %>% unlist >0)){print("there is no more missing after successful imputation")}  *## [1] "there is no more missing after successful imputation"*  *# dim*  sapply(list(impu\_data, L.training, L.test), dim) %>%  as.data.frame %>%  cbind(dim = c("# of row", "# of column")) %>%  dplyr::select(dim, imput\_data = V1, L.training = V2, L.test = V3)  *## dim imput\_data L.training L.test*  *## 1 # of row 176 141 35*  *## 2 # of column 58 58 58*  *# KNN*  knnModel2 <- train(Yield~.,  data = L.training,  method = "knn",  preProc = c("center", "scale"),  tuneLength = 10)  *# MARS*  marsGrid2 <- expand.grid(degree = 1:2, nprune = 1:20)  marsModel2 <- train(Yield~.,  data = L.training,  method = "earth",  tuneGrid = marsGrid2,  trControl = trainControl(method = "cv"))  *# SVM*  svmModel2 <- train(Yield~.,  data = L.training,  method = "svmRadial",  preProc = c("center", "scale"),  tuneLength = 14,  trControl = trainControl(method = "cv"))  *# NNET*  nnetGrid2 <- expand.grid(decay = c(0, 0.01, .1), size = c(1:10))  nnetModel2 <- train(Yield~.,  data = L.training,  method = "nnet",  tuneGrid = nnetGrid2,  trControl = trainControl(method = "cv"),  preProc = c("center", "scale"),  linout = TRUE,  trace = FALSE,  maxit = 500)  *# Prediction*  knnPred2 <- predict(knnModel2, newdata = L.test)  marsPred2 <- predict(marsModel2, newdata = L.test)  svmPred2 <- predict(svmModel2, newdata = L.test)  nnetPred2 <- predict(nnetModel2, newdata = L.test)  *# Performance*  knnPerformance2 <- postResample(pred = knnPred2, obs = L.test$Yield)  marsPerformance2 <- postResample(pred = marsPred2, obs = L.test$Yield)  svmPerformance2 <- postResample(pred = svmPred2, obs = L.test$Yield)  nnetPerformance2 <- postResample(pred = nnetPred2, obs = L.test$Yield)  *# Summary*  rbind(  *# KNN*  "KNN Train" = c("RMSE" = max(knnModel2$results$RMSE),  "RSquared" = max(knnModel2$results$Rsquared),  "MAE" = max(knnModel2$results$MAE)),  "KNN Test" = knnPerformance2,  *# MARS*  "MARS Train" = c("RMSE" = max(marsModel2$results$RMSE),  "RSquared" = max(marsModel2$results$Rsquared),  "MAE" = max(marsModel2$results$MAE)),  "MARS Test" = marsPerformance2,  *# SVM*  "SVM Train" = c(max(svmModel2$results$RMSE),  max(svmModel2$results$Rsquared),  max(svmModel2$results$MAE)),  "SVM Test" = svmPerformance2,  *# NNET*  "NNET Train" = c(max(nnetModel2$results$RMSE),  max(nnetModel2$results$Rsquared),  max(nnetModel2$results$MAE)),  "NNET Test" = nnetPerformance2) %>%  kable(caption = "Model Performance on ChemicalManufacturing Data", digits = 3) %>%  kable\_styling() |



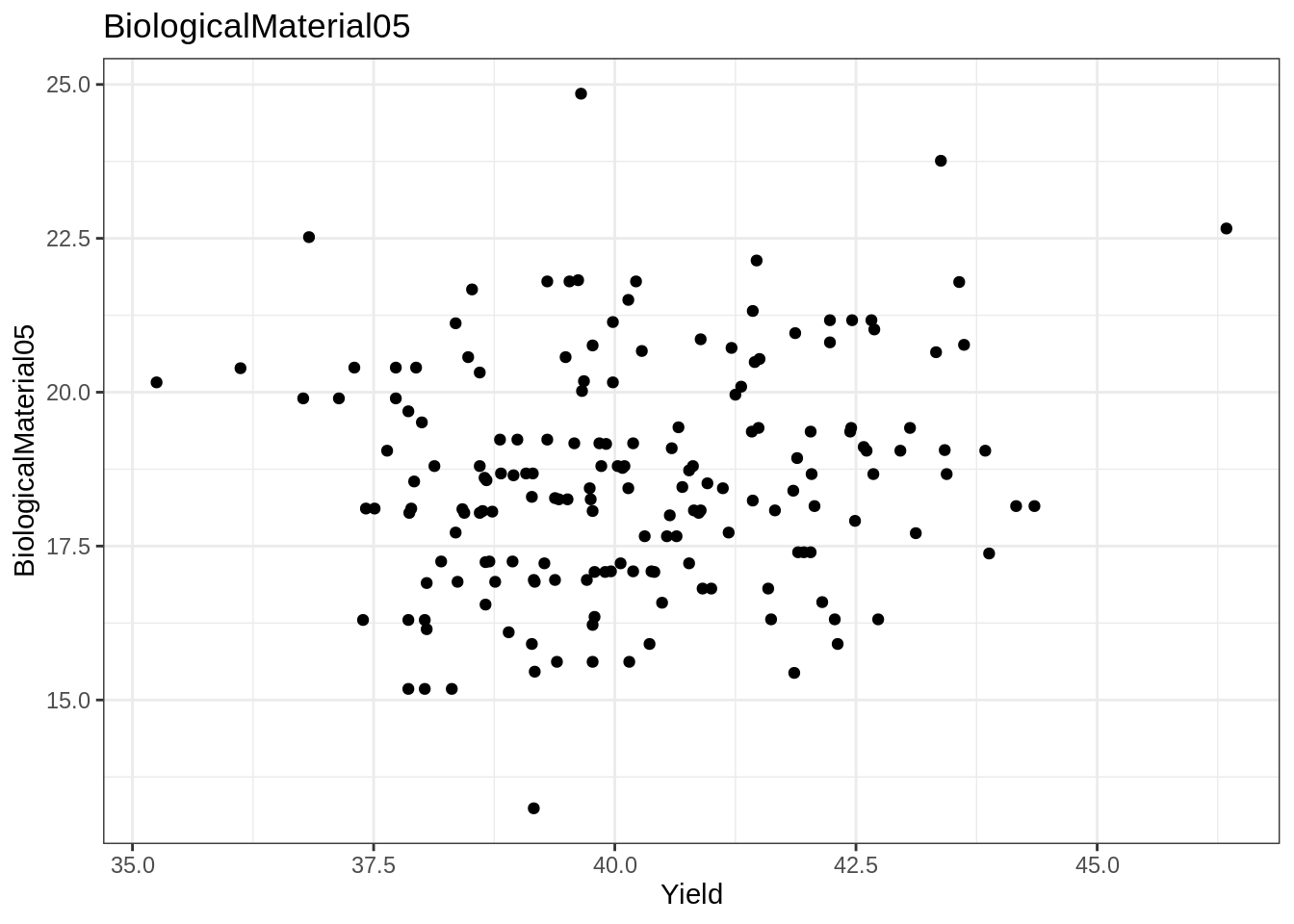
|  |
| --- |
| **(b). Which predictors are most important in the optimal nonlinear regression model? Do either the biological or process variables dominate the list? How do the top ten important predictors compare to the top ten predictors from the optimal linear model?** |
| svmImportance2 <- caret::varImp(svmModel2)  svmImportance2$importance %>%  as.data.frame() %>%  tibble::rownames\_to\_column() %>%  dplyr::mutate(name = forcats::fct\_inorder(rowname)) %>%  arrange(desc(Overall)) %>%  head(10) %>%  ggplot(., aes(x = reorder(name, Overall), y = Overall)) +  geom\_point() +  geom\_segment(aes(x = name, xend = name, y = 0, yend = Overall)) +  ggtitle("Variable Importance: SVM for ChemicalManufacturing Data") +  labs(x = "Variable", y = "Importance") +  coord\_flip() +  theme\_minimal() |
|  |
| **(c). Explore the relationships between the top predictors and the response for the predictors that are unique to the optimal nonlinear regression model. Do these plots reveal intuition about the biological or process predictors and their relationship with yield?** In general, the SVM model pointed to the same direction, conclusion as the linear model (Elastic net) as both indicated that the ManufacturingProcess variables seemed to be more important, e.g. ManufacturingProcess32, 09 tops the list. However, the SVM found that several biological variables important that were not found in the linear model and vice versa, e.g. BiologicalMaterial02 was found in SVM only, whereas BiologicalMaterial05 was found in the linear model only. It seemed that both variables positively correlated with Yield, but the relationship between BiologicalMaterial02 and Yield seemed to be much more obvious and stronger than 05.  features <- svmImportance2$importance %>%  as.data.frame() %>%  tibble::rownames\_to\_column() %>%  dplyr::mutate(name = forcats::fct\_inorder(rowname)) %>%  arrange(desc(Overall)) %>%  head(10) %>%  .$rowname %>%  as.vector  subset <- impu\_data %>%  dplyr::select(all\_of(c("Yield", features)))  corSubset <- cor(subset) %>%  as.matrix %>%  as.data.frame %>%  tibble::rownames\_to\_column(., "variable") %>%  arrange(desc(Yield)) %>%  dplyr::select(variable, Yield)  corSubset %>%  kable(caption = "Correlation") %>%  kable\_styling() |



|  |
| --- |
| *# BiologicalMaterial02*  impu\_data %>%  dplyr::select(Yield, BiologicalMaterial02) %>%  ggplot(., aes(Yield, BiologicalMaterial02)) +  geom\_point() +  ggtitle("BiologicalMaterial02") +  theme\_classic() |



|  |
| --- |
| *# BiologicalMaterial05*  impu\_data %>%  dplyr::select(Yield, BiologicalMaterial05) %>%  ggplot(., aes(Yield, BiologicalMaterial05)) +  geom\_point() +  ggtitle("BiologicalMaterial05") +  theme\_bw() |



# Kuhn & Johnson 8

## HW\_KJ\_8.1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **library**(AppliedPredictiveModeling) 8.1. Recreate the simulated data from Exercise 7.2: **library**(mlbench)  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" (a) Fit a random forest model to all of the predictors, then estimate the variable importance scores: **library**(randomForest)  ## randomForest 4.6-14  ## Type rfNews() to see new features/changes/bug fixes.  **library**(caret)  rfmodel1 <- randomForest(y ~ ., data = simulated,  importance = TRUE,  ntree = 1000)  rfImp1 <- varImp(rfmodel1, 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 Did the random forest model significantly use the uninformative predictors (V6 – V10)? The importance values of predictors (V6 - V10) are either very low and positive or very low and negative. This means that the random forest model really did not use those variables.  varImpPlot(rfmodel1) | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (b) Now add an additional predictor that is highly correlated with one of the informative predictors. For example: simulated2 <- simulated  simulated2$duplicate1 <- simulated2$V1 + rnorm(200) \* .1  cor(simulated2$duplicate1, simulated2$V1)  ## [1] 0.9460206 Fit another random forest model to these data. Did the importance score for V1 change? What happens when you add another predictor that is also highly correlated with V1? rfmodel2 <- randomForest(y ~ ., data = simulated2,  importance = TRUE,  ntree = 1000)  rfImp2 <- varImp(rfmodel2, scale = FALSE)  rfImp2  ## 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  The variable V1 importance value has been affected, It importance value has decreased. The modele chose randomly which variable to use among the two highly correlated variable.There are two variables with the same type of information. This introduces instability in the model coefficients. | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (c) Use the cforest function in the party package to fit a random forest model using conditional inference trees. The party package function varimp can calculate predictor importance. The conditional argument of that function toggles between the traditional importance measure and the modified version described in Strobl et al. (2007). Do these importances show the same pattern as the traditional random forest model? **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  bagCtrl <- cforest\_control(mtry = ncol(simulated) - 1)  baggedTree <- cforest(y ~ ., data = simulated)  **library**(caret)  crfImp <- varimp(baggedTree, conditional = TRUE)  crfImp  ## V1 V2 V3 V4 V5 V6  ## 5.471457361 5.166826657 0.020994281 6.689072245 1.256076719 0.004925215  ## V7 V8 V9 V10  ## -0.008184439 -0.009141850 -0.012594617 -0.013200299  crf.importance <- data.frame(var = names(crfImp), y =crfImp)  **library**(dplyr)  ##  ## Attaching package: 'dplyr'  ## The following object is masked from 'package:randomForest':  ##  ## combine  ## The following objects are masked from 'package:stats':  ##  ## filter, lag  ## The following objects are masked from 'package:base':  ##  ## intersect, setdiff, setequal, union  crf.imp <- crf.importance %>% arrange(desc(y))  crf.imp  ## var y  ## V4 V4 6.689072245  ## V1 V1 5.471457361  ## V2 V2 5.166826657  ## V5 V5 1.256076719  ## V3 V3 0.020994281  ## V6 V6 0.004925215  ## V7 V7 -0.008184439  ## V8 V8 -0.009141850  ## V9 V9 -0.012594617  ## V10 V10 -0.013200299  crf.imp %>% dplyr::mutate(var = reorder(var, y)) %>%  ggplot(aes(x = var, y= y)) +  geom\_point(show.legend = FALSE) +  labs(x = "Important Variables", y = NULL) +  coord\_flip() | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (d) Repeat this process with different tree models, such as boosted trees and Cubist. Does the same pattern occur? Boosted trees models  **library**(gbm)  ## Loaded gbm 2.1.5  gbmModel <- gbm(y ~ ., data = simulated, distribution = "gaussian")  gbmImp <- varImp(gbmModel, numTrees = 100, scale = FALSE)  gbmImp  ## Overall  ## V1 4071.65288  ## V2 3550.71836  ## V3 1156.48989  ## V4 4416.45306  ## V5 1791.01067  ## V6 81.00046  ## V7 0.00000  ## V8 0.00000  ## V9 0.00000  ## V10 0.00000  The most important predictors are V1, V2, V3, V4, V5 The V6 predictor is less important than the first four V7, V8, V9, and V10 have not been used by the model  summary(gbmModel) | |  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

## HW\_KJ\_8.2

|  |
| --- |
| 8.2. Use a simulation to show tree bias with different granularities. Generate random variables Predictors with different granularities Generate also the rsponse variable collinear to some of them  X1 <- sample(c(3, 5), 100, replace = TRUE)  X2 <- sample(c(3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41), 100, replace = TRUE)  X3 <- sample(10:1000/10, 100)  y <- 7\*X1 + 3\*X2 + rnorm(100)  df <- data.frame(X1, X2, X3, y)  head(df)  ## X1 X2 X3 y  ## 1 5 3 80.1 44.95411  ## 2 3 19 54.6 76.88111  ## 3 3 11 45.0 51.92110  ## 4 5 7 54.5 55.32196  ## 5 3 5 51.2 34.74864  ## 6 3 19 66.0 76.96352  Fit a model and Compare the predictors importance  rfBias <- randomForest(y ~ ., data = df,  importance = TRUE,  ntree = 100)  varImportance <- varImp(rfBias, scale = FALSE)  varImportance  ## Overall  ## X1 125.24888  ## X2 1536.94859  ## X3 6.56273  The predictor X2 with more unique values is more used by the random forest model than the predictor X2 with less unique values. But the variable X3 with more distinct values that is non correlated and has no relation with the response variable was also used by the model. |

## HW\_KJ\_8.3

|  |
| --- |
| 8.3. In stochastic gradient boosting the bagging fraction and learning rate will govern the construction of the trees as they are guided by the gradient. Although the optimal values of these parameters should be obtained through the tuning process, it is helpful to understand how the magnitudes of these parameters affect magnitudes of variable importance. Figure 8.24 provides the variable importance plots for boosting using two extreme values for the bagging fraction (0.1 and 0.9) and the learning rate (0.1 and 0.9) for the solubility data. The left-hand plot has both parameters set to 0.1, and the right-hand plot has both set to 0.9:(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? The gradient boosting model employs the greedy strategy of choosing the optimal weak learner at each stage. Ridgeway (2007) suggests that small values of the learning parameter (< 0.01) work best. High values of learning rate will result in increasing the contribution of few variables at each stage of the process and then shrinking the number of important variables.  Learning rate controls the fraction of the predictions of each tree being added. A higher learning rate means that larger fraction of each tree’s predictions are added to the final prediction. This effectively means that a higher learning rate increases the dependent / correlation structure. More of the same predictors will be selected among the trees. This is why the right-hand plot has its importance focus on just the first few of the predictors, and look very steep.  Bagging fraction is the fraction of data being used in each iteration of the trees. When you have a small bagging fraction, say 0.1, on each iteration just 10% of the full data is randomly sampled. So each tree may be built using very different dataset. Since the dataset are very different, the trees will be splitting very differently from each other. On the contrast, when you have large bagging fraction, say 0.9, essentially on each iteration the trees are seeing the same dataset - they will likely split similarly. This means that larger bagging fraction increases the dependent / correlated structure in the boosting trees. Therefore, the right-hand plot with a larger bagging fraction has its importance focus on just the first few of the predictors. (b) Which model do you think would be more predictive of other samples? Greedy models are less likely to select the optimal global model and are prone to over-fitting. Stochastic models reduce prediction variance. Therefore, the less greedy model on the left (with a 0.1 learning rate) that is also more random (due to only selecting 0.1 of the training set observations to propose the next tree in the expansion) would be more predictive of other samples.  Learning rate and bagging fraction are important parameters to control the overfitting of the gradient boosting model that requires tuning. A smaller learning rate and bagging fraction leads to better generalization ability over unseen samples. If I have to guess, the model with 0.1 learning rate and bagging fraction will be more predictive of out of bag samples. However, since this invovles a trade off between bias-variance, I can only confirm using cross validation or a test set. (c) How would increasing interaction depth affect the slope of predictor importance for either model in Fig. 8.24 The author comment stated that the larger value of shrinkage has an impact on reducting RMSE for all choises of tree depth and number of trees.  Figure 8.20 presents the cross-validated RMSE results for boosted trees across tuning parameters of tree depth (1–7), number of trees (100–1,000), and shrinkage (0.01 or 0.1); the bagging fraction in this illustration was fixed at 0.5. When examining this figure, the larger value of shrinkage (right-hand plot) has an impact on reducing RMSE for all choices of tree depth and number of trees. Also, RMSE decreases as tree depth increases when shrinkage is 0.01. The same pattern holds true for RMSE when shrinkage is 0.1 and the number of trees is less than 300. Using the one-standard-error rule, the optimal boosted tree has depth 3 with 400 trees and shrinkage of 0.1. |