DATA 624: Group 2 Homework

# Dependencies

The following packages were used in R for completion of our homework assignments:

#Textbook Packages  
library(fpp2)  
library(AppliedPredictiveModeling)  
library(mlbench)  
  
#Graphing  
library(ggplot2)  
library(gridExtra)  
  
#Math  
library(caret)  
library(forecast)  
library(randomForest)  
  
#Formatting  
library(knitr)

# Assignment 1

Week 1: HA #2.1; HA #2.3

## 2.1: Use the help function to explore what the series gold, woolyrnq and gas represent.

#help("gold")  
#help("woolyrnq")  
#help("gas")

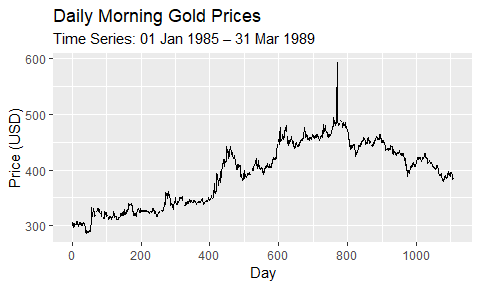
Per the help function, the gold series represents “daily morning gold prices in US dollars. 1 January 1985 – 31 March 1989”; the woolyrnq series represents “quarterly production of woollen yarn in Australia: tonnes. Mar 1965 – Sep 1994”; and, the gas series represents “Australian monthly gas production: 1956–1995.”

### (a). Use autoplot() to plot each of these in separate plots.

The time plots below were generated from the autoplot function.

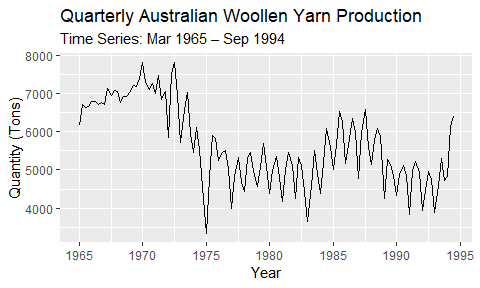
The gold plot follows a general incrementral trend until a large spike around the indexed date of appromately 775. At which point, the value of gold begins to decrease over the remaining time captured in this series.

autoplot(gold) +   
 labs(title = "Daily Morning Gold Prices",   
 subtitle = "Time Series: 01 Jan 1985 – 31 Mar 1989",   
 x = "Day",   
 y = "Price (USD)")



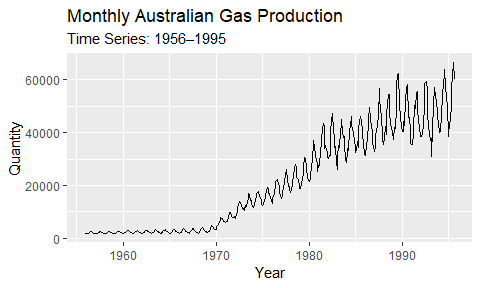
The woolyrnq plot follows a decremental trend. There is a large, downward spike at the start of 1975 worth noting.

autoplot(woolyrnq) +  
 labs(title = "Quarterly Australian Woollen Yarn Production",   
 subtitle = "Time Series: Mar 1965 – Sep 1994",   
 x = "Year",   
 y = "Quantity (Tons)")



Lastly, the gas plot shows monthly changes to the Australian gas production. The seasonal pattern of this series appears to decrease, while the overall trend in production increases.

autoplot(gas) +  
 labs(title = "Monthly Australian Gas Production",   
 subtitle = "Time Series: 1956–1995",   
 x = "Year",   
 y = "Quantity")



### (b). What is the frequency of each series? Hint: apply the frequency() function.

The frequency of timeseries shows us the number of observations within a singular seasonal pattern.

frequency(gold)

## [1] 1

frequency(woolyrnq)

## [1] 4

frequency(gas)

## [1] 12

Our output from the frequency function informs us that gold is an annual time series, woolyrnq is a quarterly series, and gas is a monthly series.

### (c). Use which.max() to spot the outlier in the gold series. Which observation was it?

The which.max() function returns an index value of the maximum value within a series. Using this function, the maximum value for gold is shown below:

which.max(gold)

## [1] 770

This number aligns with the observed spike in gold price from the time series plot in part (a).

## 2.3: Download some monthly Australian retail data from the book website. These represent retail sales in various categories for different Australian states, and are stored in a MS-Excel file.

### (a). You can read the data into R with the following script:

retaildata <- readxl::read\_excel("data/retail.xlsx", skip=1)

*The second argument (skip=1) is required because the Excel sheet has two header rows.*

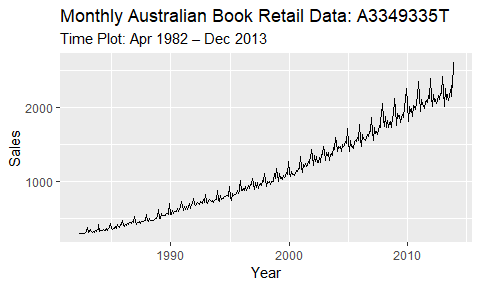
### (b). Select one of the time series as follows (but replace the column name with your own chosen column):

myts <- ts(retaildata[,"A3349335T"], frequency=12, start=c(1982,4))

### (c). Explore your chosen retail time series using the following functions: autoplot(), ggseasonplot(), ggsubseriesplot(), gglagplot(), ggAcf().

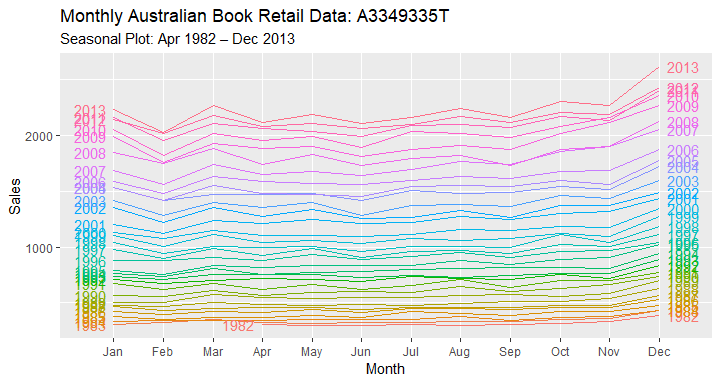
The output from the autoplot function shows a general, incremental trend over the observed period; however, there appears to be small, decremental trends within individual seasonal pattern.

autoplot(myts) +   
 labs(title = "Monthly Australian Book Retail Data: A3349335T",   
 subtitle = "Time Plot: Apr 1982 – Dec 2013",  
 x = "Year",   
 y = "Sales")



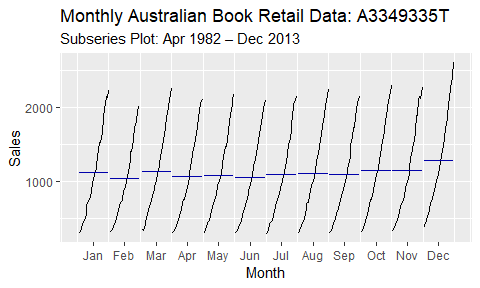
Seaonal plots shows the observed data plotted against each season within our time series. The seasons are overlapped, which allows for us to view underlying seasonal patterns and the years in which these trends occur. Our seaonal plot below shows that book sales tend to generally decrease in from Jan-Feb, Mar-Apr, May-Jun, Aug-Sep, and Oct-Nov. Sales generally appear to increase seasonally between the other observed months.

ggseasonplot(myts,year.labels=TRUE, year.labels.left=TRUE) +   
 labs(title = "Monthly Australian Book Retail Data: A3349335T",   
 subtitle = "Seasonal Plot: Apr 1982 – Dec 2013",   
 x = "Month",   
 y = "Sales")



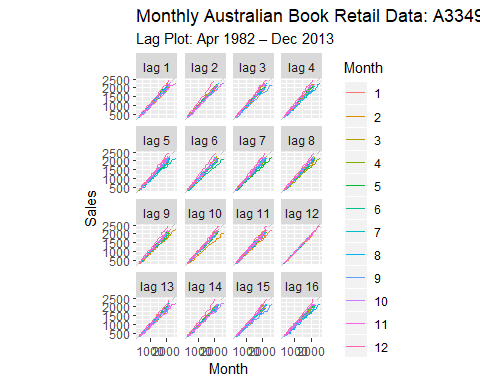
The subseries plot provides a different way to view underlying seasonal patterns by separating each season in a year into an individual plot. The horizonal line shows the mean value of each month. The plot below reveals that, on average, the month of Feburary holds the lowest sales and the month of December contains the highest sales throughout all observed time periods.

ggsubseriesplot(myts) +  
 labs(title = "Monthly Australian Book Retail Data: A3349335T",   
 subtitle = "Subseries Plot: Apr 1982 – Dec 2013",   
 x = "Month",   
 y = "Sales")



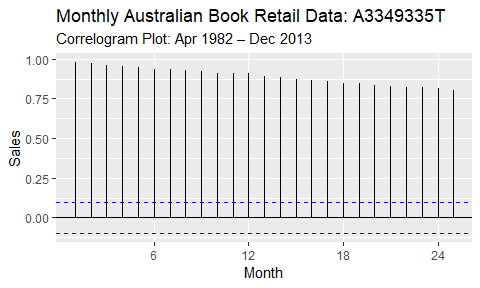
Lag plots is used to examine the correlation between the X and Y axis over a fixed period of time using a scatterplot. Per the text, “each graph shows plotted against for different values of ”. Our lag plot below shows an overall positive relationship at each lag, which indicates a strong seaonal relationship in our data.

gglagplot(myts)+  
 labs(title = "Monthly Australian Book Retail Data: A3349335T",   
 subtitle = "Lag Plot: Apr 1982 – Dec 2013",   
 x = "Month",   
 y = "Sales")



Lastly, our autocorrelation plot (also referred to as correlogram plot) measures the linear relationship between the time series’ lagged values. We used an autocorrelation function (ACF) below (ggAcf) to examine this relationship within our selected variable from the book retail data. From the graph, we can tell that is our highest lag and is our smallest lag. We can tell our data is trended because our plot depicts positive values that decrease slowly as our lags increase.

ggAcf(myts) +  
 labs(title = "Monthly Australian Book Retail Data: A3349335T",   
 subtitle = "Correlogram Plot: Apr 1982 – Dec 2013",   
 x = "Month",   
 y = "Sales")



# Assignment 2

Week 2: HA #6.2

## 6.2: The plastics data set consists of the monthly sales (in thousands) of product A for a plastics manufacturer for five years.

### (a). Plot the time series of sales of product A. Can you identify seasonal fluctuations and/or a trend-cycle?

#code

### (b). Use a classical multiplicative decomposition to calculate the trend-cycle and seasonal indices.

#code

### (c). Do the results support the graphical interpretation from part a?

#code

### (d). Compute and plot the seasonally adjusted data.

#code

### (e). Change one observation to be an outlier (e.g., add 500 to one observation), and recompute the seasonally adjusted data. What is the effect of the outlier?

#code

### (f). Does it make any difference if the outlier is near the end rather than in the middle of the time series?

#code

# Assignment 3

Week 3: KJ #3.1; KJ #3.2

## 3.1: The UC Irvine Machine Learning Repository contains a data set related to glass identiﬁcation. The data consist of 214 glass samples labeled as one of seven class categories. There are nine predictors, including the refractive index and percentages of eight elements: Na, Mg, Al, Si, K, Ca, Ba, and Fe. The data can be accessed via:

data(Glass)  
str(Glass)

FALSE 'data.frame': 214 obs. of 10 variables:  
FALSE $ RI : num 1.52 1.52 1.52 1.52 1.52 ...  
FALSE $ Na : num 13.6 13.9 13.5 13.2 13.3 ...  
FALSE $ Mg : num 4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...  
FALSE $ Al : num 1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...  
FALSE $ Si : num 71.8 72.7 73 72.6 73.1 ...  
FALSE $ K : num 0.06 0.48 0.39 0.57 0.55 0.64 0.58 0.57 0.56 0.57 ...  
FALSE $ Ca : num 8.75 7.83 7.78 8.22 8.07 8.07 8.17 8.24 8.3 8.4 ...  
FALSE $ Ba : num 0 0 0 0 0 0 0 0 0 0 ...  
FALSE $ Fe : num 0 0 0 0 0 0.26 0 0 0 0.11 ...  
FALSE $ Type: Factor w/ 6 levels "1","2","3","5",..: 1 1 1 1 1 1 1 1 1 1 ...

### (a). Using visualizations, explore the predictor variables to understand their distributions as well as the relationships between predictors.

#code

### (b). Do there appear to be any outliers in the data? Are any predictors skewed?

#code

### (c). Are there any relevant transformations of one or more predictors that might improve the classification model?

#code

## 3.2: The soybean data can also be found at the UC Irvine Machine Learning Repository. Data were collected to predict disease in 683 soybeans. The 35 predictors are mostly categorical and include information on the environmental conditions (e.g., temperature, precipitation) and plant conditions (e.g., left spots, mold growth). The outcome labels consist of 19 distinct classes. The data can be loaded via:

data(Soybean)

### (a). Investigate the frequency distributions for the categorical predictors. Are any of the distributions degenerate in the ways discussed earlier in this chapter?

#code

### (b). Roughly 18% of the data are missing. Are there particular predictors that are more likely to be missing? Is the pattern of missing data related to the classes?

#code

### (c). Develop a strategy for handling missing data, either by eliminating predictors or imputation.

#code

# Assignment 4

Week 4: HA #7.1; HA #7.3

## 7.1: Consider the pigs series — the number of pigs slaughtered in Victoria each month.

### (a). Use the ses() function in R to find the optimal values of and , and generate forecasts for the next four months.

#code

### (b). Compute a 95% prediction interval for the first forecast using where is the standard deviation of the residuals. Compare your interval with the interval produced by R.

#code

## 7.3: Modify your function from the previous exercise to return the sum of squared errors rather than the forecast of the next observation. Then use the optim() function to find the optimal values of and . Do you get the same values as the ses() function?

#code

# Assignment 5

Week 5: HA #7.5; HA #7.6; HA #7.10

## 7.5: Data set books contains the daily sales of paperback and hardcover books at the same store. The task is to forecast the next four days’ sales for paperback and hardcover books.

### (a). Plot the series and discuss the main features of the data.

#code

### (b). Use the ses() function to forecast each series, and plot the forecasts.

#code

### (c). Compute the RMSE values for the training data in each case.

#code

## 7.6: Continuation of exercise 7.5.

### (a). Now apply Holt’s linear method to the paperback and hardback series and compute four-day forecasts in each case.

#code

### (b). Compare the RMSE measures of Holt’s method for the two series to those of simple exponential smoothing in the previous question. (Remember that Holt’s method is using one more parameter than SES.) Discuss the merits of the two forecasting methods for these data sets.

#code

### (c). Compare the forecasts for the two series using both methods. Which do you think is best?

#code

### (d). Calculate a 95% prediction interval for the first forecast for each series, using the RMSE values and assuming normal errors. Compare your intervals with those produced using ses and holt.

#code

## 7.10: For this exercise use data set ukcars, the quarterly UK passenger vehicle production data from 1977Q1–2005Q1.

### (a). Plot the data and describe the main features of the series.

#code

### (b). Decompose the series using STL and obtain the seasonally adjusted data.

#code

### (c). Forecast the next two years of the series using an additive damped trend method applied to the seasonally adjusted data. (This can be done in one step using stlf() with arguments etsmodel="AAN", damped=TRUE.)

#code

### (d). Forecast the next two years of the series using Holt’s linear method applied to the seasonally adjusted data (as before but with damped=FALSE).

#code

### (e). Now use ets() to choose a seasonal model for the data.

#code

### (f). Compare the RMSE of the ETS model with the RMSE of the models you obtained using STL decompositions. Which gives the better in-sample fits?

#code

### (g). Compare the forecasts from the three approaches? Which seems most reasonable?

#code

### (h). Check the residuals of your preferred model.

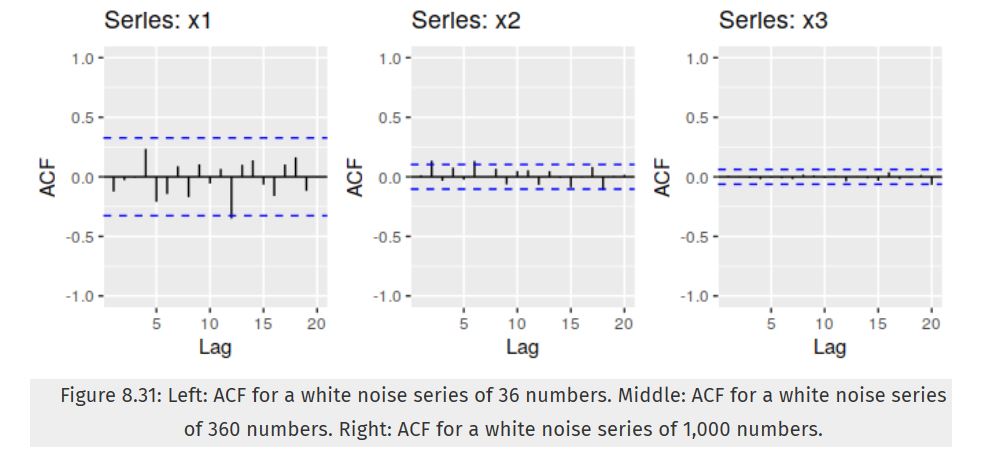
#code

# Assignment 6

Week 6-7: HA #8.1; HA #8.2; HA #8.6; HA #8.8

## 8.1: Figure 8.31 shows the ACFs for 36 random numbers, 360 random numbers and 1,000 random numbers.

### (a). Explain the differences among these figures. Do they all indicate that the data are white noise?



### (b). Why are the critical values at different distances from the mean of zero? Why are the autocorrelations different in each figure when they each refer to white noise?

#code

## 8.2: A classic example of a non-stationary series is the daily closing IBM stock price series (data set ibmclose). Use R to plot the daily closing prices for IBM stock and the ACF and PACF. Explain how each plot shows that the series is non-stationary and should be differenced.

#code

## 8.6: Use R to simulate and plot some data from simple ARIMA models.

### (a). Use the following R code to generate data from an AR(1) model with and . The process starts with .

y <- ts(numeric(100))  
e <- rnorm(100)  
for(i in 2:100)  
 y[i] <- 0.6\*y[i-1] + e[i]

### (b). Produce a time plot for the series. How does the plot change as you change ?

#code

### (c). Write your own code to generate data from an MA(1) model with and .

#code

### (d).Produce a time plot for the series. How does the plot change as you change ?

#code

### (e). Generate data from an ARMA(1,1) model with , , and .

#code

### (f). Generate data from an AR(2) model with , , and . (Note that these parameters will give a non-stationary series.)

#code

### (g). Graph the latter two series and compare them.

#code

## 8.8: Consider austa, the total international visitors to Australia (in millions) for the period 1980-2015.

### (a). Use auto.arima() to find an appropriate ARIMA model. What model was selected. Check that the residuals look like white noise. Plot forecasts for the next 10 periods.

#code

### (b). Plot forecasts from an ARIMA(0,1,1) model with no drift and compare these to part a. Remove the MA term and plot again.

#code

### (c). Plot forecasts from an ARIMA(2,1,3) model with drift. Remove the constant and see what happens.

#code

### (d). Plot forecasts from an ARIMA(0,0,1) model with a constant. Remove the MA term and plot again.

#code

### (e). Plot forecasts from an ARIMA(0,2,1) model with no constant.

#code

# Assignment 7

Week 8: KJ #6.3

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

### (a) Start R and use these commands to load the data:

data(chemicalManufacturing)

## Warning in data(chemicalManufacturing): data set 'chemicalManufacturing'  
## not found

The matrix processPredictors contains the 57 predictors (12 describing the input biological material and 45 describing the process predictors) for the 176 manufacturing runs. yield contains the percent yield for each run.

### (b) A small percentage of cells in the predictor set contain missing values. Use an imputation function to ﬁll in these missing values (e.g., see Sect. 3.8).

#code

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

#code

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

#code

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

#code

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

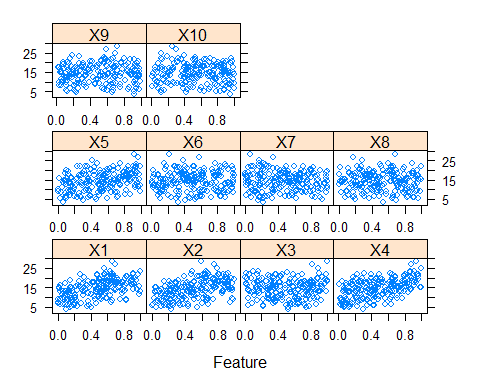
#code

# Assignment 8

Week 9: KJ #7.2; KJ #7.5

## 7.2: Friedman (1991) introduced several benchmark data sets create by simulation. One of these simulations used the following nonlinear equation to create data: ; where the values are random variables uniformly distributed between (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:

set.seed(200)   
trainingData <- mlbench.friedman1(200, sd = 1)  
  
## We convert the 'x' data from a matrix to a data frame   
## One reason is that this will give the columns names.  
  
trainingData$x <- data.frame(trainingData$x)   
  
## Look at the data using   
featurePlot(trainingData$x, trainingData$y)



## or other methods.   
  
## This creates a list with a vector 'y' and a matrix   
## of predictors 'x'. Also simulate a large test set to   
## estimate the true error rate with good precision:   
  
testData <- mlbench.friedman1(5000, sd = 1)  
testData$x <- data.frame(testData$x)

### (a). Tune several models on these data. For example:

knnModel <- train(x = trainingData$x,  
 y = trainingData$y,   
 method = "knn",  
 preProc = c("center", "scale"),   
 tuneLength = 10)   
knnModel

## 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 3.565620 0.4887976 2.886629  
## 7 3.422420 0.5300524 2.752964  
## 9 3.368072 0.5536927 2.715310  
## 11 3.323010 0.5779056 2.669375  
## 13 3.275835 0.6030846 2.628663  
## 15 3.261864 0.6163510 2.621192  
## 17 3.261973 0.6267032 2.616956  
## 19 3.286299 0.6281075 2.640585  
## 21 3.280950 0.6390386 2.643807  
## 23 3.292397 0.6440392 2.656080  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was k = 15.

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   
## 3.1750657 0.6785946 2.5443169

Model 1:

#code

Model 2:

#code

Model 3:

#code

### (b). Which models appear to give the best performance? Does MARS select the informative predictors (those named X1–X5)?

#code

# Assignment 9

Week 10-11: KJ #8.1; KJ #8.2; KJ #8.3; KJ #8.7

## 8.1: Recreate the simulated data from Exercise 7.2:

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. Did the random forest model signiﬁcantly use the uninformative predictors (V6 – V10)?

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

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

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

## [1] 0.9460206

#code

### (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?

#code

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

#code

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

#code

## 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?

#code

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

#code

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

#code

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

### (a). Which tree-based regression model gives the optimal resampling and test set performance?

#code

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

#code

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

#code